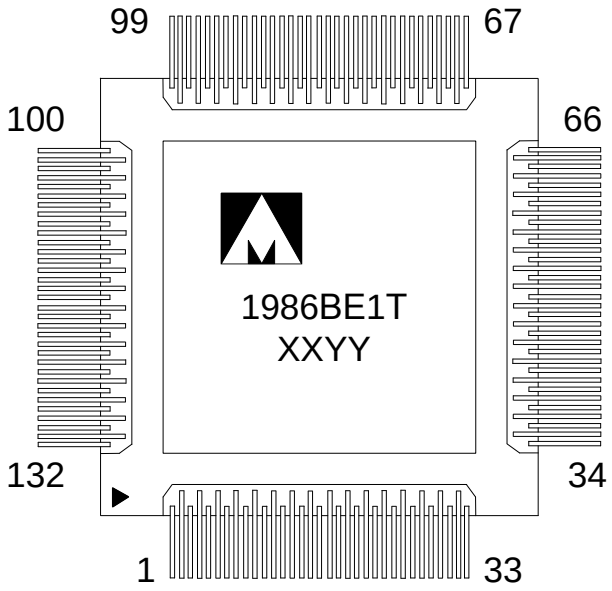
32-х разрядный контроллер для авиационного применения
1986BE1T, К1986BE1T



XX – год выпуска
YY – неделя выпуска

Основные параметры микросхемы

- Тактовая частота до 144* МГц
- 62 однословных инструкции
- 32 х 32 битный аппаратный умножитель, за 3 такта
- Поддержка прямого, косвенного и относительного режимов адресации
- Аппаратная поддержка интерфейса CAN
- 8 канальный 12 разрядный АЦП
- 2 блока 12 разрядных ЦАП
- Уменьшенное до 14 мс время запуска микроконтроллера
- Диапазон напряжения питания 3,0…3,6 В
- Температурный диапазон:

Обозначение	Диапазон
1986BE1T	минус 60 – 125 °С
К1986BE1T	минус 60 – 125 °С
К1986BE1TК	0 – 70 °С

*- начиная с ревизии 3 тактовая частота до 144 МГц (140 МГц для 1 и 2 ревизий)

Тип корпуса:

- 132-х выводной металлокерамический 4229.132-3

Основные возможности

Ядро:

- 32-битное RISC ядро, тактовая частота до 144 МГц (с ревизии 3), производительность 0.8 DMIPS/MHz при нулевой задержке памяти;
- умножение 32x32 за три цикла

Память:

- встроенная энергонезависимая память программ FLASH типа размером 128 Кбайт
- встроенное ОЗУ размером 48 Кбайт
- контроллер внешней системной шины с поддержкой микросхем памяти СОЗУ, ПЗУ, NAND Flash.

Питание и тактовая частота:

- внешнее питания 2,2…3,6 В
- встроенный регулятор напряжения на 1,8 В для питания ядра
- встроенные схемы контроля питания
- встроенный домен управления батарейным питанием
- встроенный подстраиваемый RC генератор 8 МГц
- встроенный подстраиваемый RC генератор 40 КГц
- внешний осциллятор 2…16 МГц
- внешний осциллятор 20…30 МГц (с ревизии 2)
- внешний осциллятор 32 КГц
- встроенный умножитель тактовой частоты PLL для ядра
- встроенный умножитель тактовой частоты PLL для контроллера USB

Режим пониженного энергопотребления:

- батарейный домен с часами реального времени и регистрами аварийного сохранения

Аналоговые модули:

- 12-ти разрядный АЦП (до 8 каналов) с амплитудой измеряемых сигналов от 0 до 3,6 В
- двухканальный 12-ти разрядный ЦАП
- 32-х разрядный ШИМ (до 8 каналов)
- температурный датчик

Периферия:

- контроллер прямого доступа в память с функциями передачи Периферия-Память, Память-Память
- два контроллера CAN интерфейса
- цифровой контроллер интерфейса по ГОСТ 18977-79
- два цифровых контроллера интерфейса по ГОСТ Р52070-2003
- цифровой контроллер интерфейса Ethernet 10/100 и PHY Transceiver
- контроллер USB интерфейса с функциями работы Device и Host
- контроллеры интерфейсов 2xUART, 3xSPI
- до 96 пользовательских линий ввода/вывода

Режим отладки:

- последовательные отладочные интерфейсы SWD и JTAG

Содержание

Введение

Микроконтроллер для авиационного применения является микроконтроллером со встроенной Flash-памятью программ и построен на базе высокопроизводительного процессорного RISC ядра. Микроконтроллер работает на тактовой частоте до 144 МГц и содержит 128 Кбайт Flash-памяти программ и 48 Кбайт ОЗУ. Периферия включает в себя контроллер USB интерфейса со встроенным аналоговым приемопередатчиком и со скоростью передачи 12 Мбит/с (Full Speed) и 1.5 Мбит/с (Low Speed), стандартные интерфейсы UART и SPI, авиационные интерфейсы по ГОСТ 18977-79 и ГОСТ Р52070-2003, цифровой интерфейс Ethernet со скоростью передачи 10/100 Мбит, интерфейсом MII и со встроенным аналоговым приемопередатчиком, контроллер внешней системной шины позволяющий работать с внешними микросхемами статического ОЗУ и ПЗУ, NAND Flash памятью и другими периферийными устройствами. Микроконтроллер содержит четыре 32-ти разрядных таймера с 4 каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Также микроконтроллер содержит системный 24-х разрядный таймер и два сторожевых таймера. Микроконтроллер содержит 12-ти разрядный высокоскоростной (до 512 Квыб/с) АЦП с возможностью оцифровки информации с 8 каналов, встроенного датчика температуры и опорного напряжения. Два 12-ти разрядных ЦАП.

Встроенные RC генераторы HSI (8 МГц) и LSI (40 кГц), внешние генераторы HSE (2…16 МГц) и LSE (32 кГц) и две схемы умножения тактовой частоты PLL для ядра и USB интерфейса позволяют гибко настраивать скорость работы микроконтроллера.

Архитектура системы памяти за счет матрицы системных шин позволяет минимизировать возможные конфликты при работе системы и повысить общую производительность. Контроллер DMA позволяет ускорить обмен информацией между ОЗУ и периферией без участи процессорного ядра.

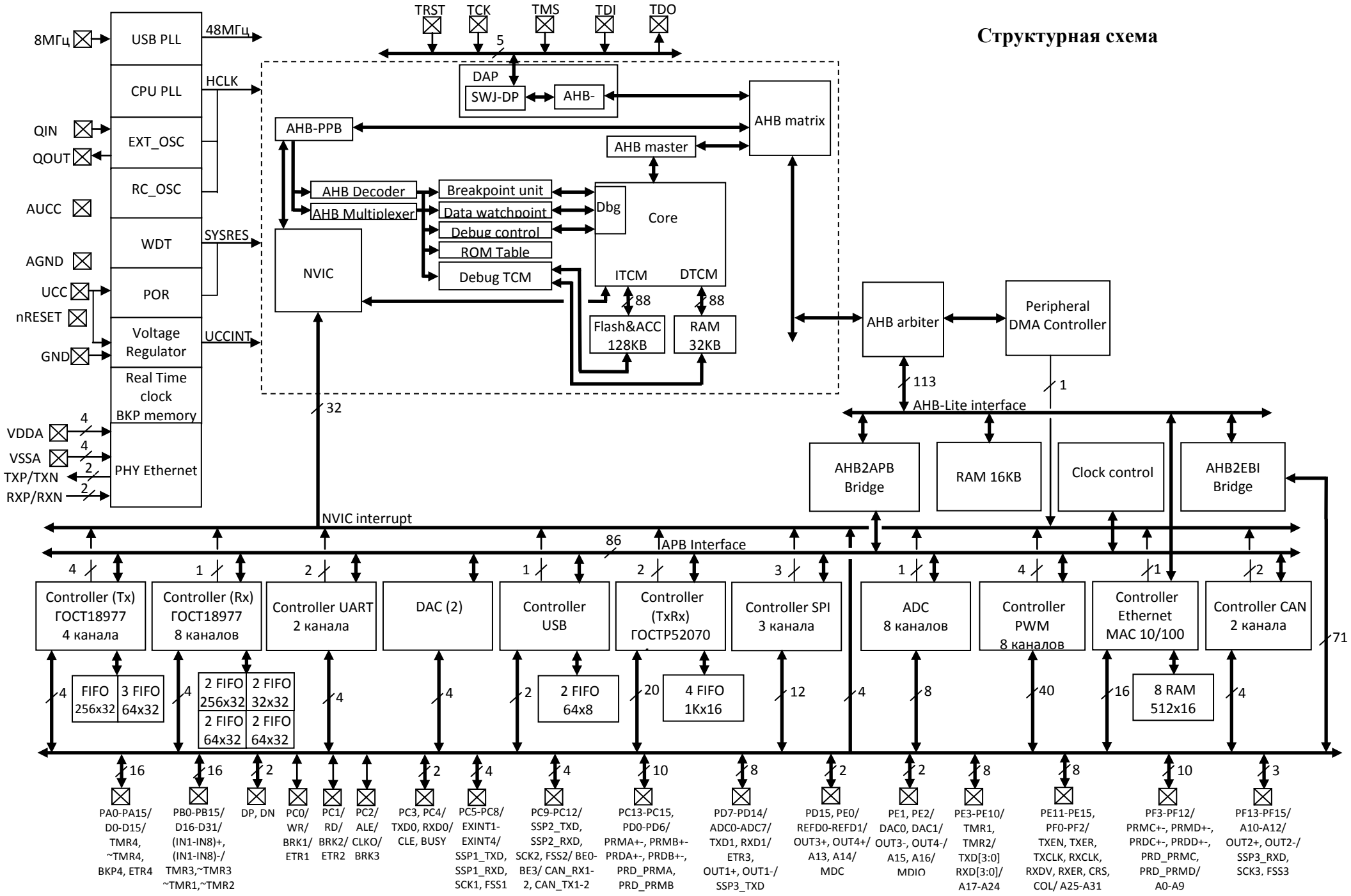
Встроенный регулятор для формирования питания внутренний цифровой части формирует напряжения 1,8В и не требует дополнительных внешних элементов. Таким образом, для работы микроконтроллера достаточно одного внешнего напряжения питания в диапазоне от 2,2 до 3,6В. Также в микроконтроллерах реализован батарейный домен, работающий от внешней батареи при отсутствии основного питания. В батарейном домене могут быть сохранены специальные флаги, а также работают часы реального времени. Встроенный детектор напряжения питания может отслеживать уровень внешнего основного питания и уровень напряжения питания на батарее. Аппаратные схемы сброса по просадке питания позволяют исключить сбойную работу микросхемы при выходе уровня напряжения питания за допустимые пределы.

Основные характеристики

В зависимости от корпуса, в котором выпускается микросхема, изменяются функциональные возможности микроконтроллеров, но при этом объем памяти программ и ОЗУ остается одинаковым.

Корпус	132 вывода					
Ядро	32 разрядное RISC					
ПЗУ	128 Кбайт Flash					
ОЗУ	48 Кбайт					
Питание	2,0…3,6 В					
Частота	144 МГц					
Температура	Минус 60….+125С					
USER IO	96					
USB	Device и Host FS (до 12 Мбит/с) встроенный PHY					
UART	2					
CAN	2					
ГОСТ Р52070-2003	2 осн. и 2 рез					
ГОСТ 18977-79	8 пр. и 4 пер.					
Ethernet 10/100 MII+PHY	1					
SPI	3					
ADC 12 разрядов 512 Квыб/с	8 каналов					
DAC 12 разрядов	2					
Внешняя шина	32 разряда					

Структурная схема



Описание выводов

Дополнительные функции вывода				Вывод порта	Корпус
Аналог.	Основ.	Альтер.	Переопр.		4229.132-3
Порт A					
-	D0	EXTINT1	ETR1	PA0 / MODE[0]	127
-	D1	EXTINT2	ETR2	PA1 / MODE[1]	126
-	D2	EXTINT3	ETR3	PA2 / MODE[2]	125
-	D3	EXTINT4	BRK1	PA3	124
-	D4	BRK2	FRX	PA4	123
-	D5	BRK3	FSD	PA5	122
-	D6	TMR4_CH1	FXEN	PA6	121
-	D7	TMR4_CH1N	FTX	PA7	120
-	D8	TMR4_CH2	PRMC+	PA8	119
-	D9	TMR4_CH2N	PRMC-	PA9	118
-	D10	TMR4_CH3	PRMD+	PA10	115
-	D11	TMR4_CH3N	PRMD-	PA11	114
-	D12	TMR4_CH4	PRDC+	PA12	113
-	D13	TMR4_CH4N	PRDC-	PA13	112
-	D14	BRK4	PRDD+	PA14	111
-	D15	ETR4	PRDD-	PA15	110
Порт B					
-	D16	IN1+	TMR3_CH1	PB0	109
-	D17	IN1-	TMR3_CH1N	PB1	108
-	D18	IN2+	TMR3_CH2	PB2	107
-	D19	IN2-	TMR3_CH2N	PB3	106
-	D20	IN3+	TMR3_CH3	PB4	105
-	D21	IN3-	TMR3_CH3N	PB5	104
-	D22	IN4+	TMR3_CH4	PB6	103
-	D23	IN4-	TMR3_CH4N	PB7	102
-	D24	IN5+	TMR1_CH1N	PB8	101
-	D25	IN5-	TMR2_CH1N	PB9	100
-	D26	IN6+	TMR1_CH2N	PB10	99
-	D27	IN6-	TMR2_CH2N	PB11	98
-	D28	IN7+	TMR1_CH3N	PB12	97
-	D29	IN7-	TMR2_CH3N	PB13	96
-	D30	IN8+	TMR1_CH4N	PB14	95
-	D31	IN8-	TMR2_CH4N	PB15	94
Порт C					
-	nWR	ETR1	BRK1	PC0	89
-	nRD	ETR2	BRK2	PC1	90
-	ALE	CLKO	BRK3	PC2	91
-	UART_TXD1	CLE	SIR_OUT1	PC3	92
-	UART_RXD1	BUSY	SIR_IN1	PC4	93
-	EXTINT1	SSP1_TXD	SSP1_RXD	PC5	56
-	EXTINT2	SSP1_RXD	SSP1_TXD	PC6	57
-	EXTINT3	SSP1_SCK	FXEN	PC7	58
-	EXTINT4	SSP1_FSS	FTX	PC8	59
-	SSP2_TXD	BE0	CAN_RX1	PC9	60
-	SSP2_RXD	BE1	CAN_TX1	PC10	61
-	SSP2_SCK	BE2	CAN_RX2	PC11	62
-	SSP2_FSS	BE3	CAN_TX2	PC12	63
-	PRMA+	A30	UART_TXD2	PC13	67
-	PRMA-	A31	UART_RXD2	PC14	68
-	PRMB+	BUSY	TMR2_CH1	PC15	69
Порт D					
-	PRMB-	ALE	A16	PD0	72
-	PRDA+	CLE	A15	PD1	73
-	PRDA-	SSP1_TXD	A14	PD2	74
-	PRDB+	SSP1_RXD	A13	PD3	75
-	PRDB-	SSP1_SCK	A7	PD4	76
-	PRD_PRMA	SSP1_FSS	A6	PD5	77
-	PRD_PRMB	nUART2RI	A5	PD6	78
ADC0_REF+	SSP2_TXD	nUART2DCD	A4	PD7	41
ADC1_REF-	SSP2_RXD	nUART2DTR	A3	PD8	42
ADC2	SSP2_SCK	nUART2DSR	A2	PD9	43
ADC3	SSP2_FSS	nUART2RTS	A1	PD10	44

ADC4	A0	nUART2CTS	FRX	PD11	45
ADC5	SSP3_TXD	ETR3	SSP3_RXD	PD12	46
ADC6	UART_TXD2	OUT1+	SIR_OUT2	PD13	47
ADC7	UART_RXD2	OUT1-	SIR_IN2	PD14	48
REFD0	OUT3+	A13	FSD	PD15	49
Порт E					
REFD1	OUT4+	A14	MDC	PE0	50
DAC0	OUT3-	A15	nUART2RI	PE1	51
DAC1	OUT4-	A16	MDIO	PE2	52
-	TMR1_CH1	A17	TXD[0]	PE3	16
-	TMR1_CH2	A18	TXD[1]	PE4	17
-	TMR1_CH3	A19	TXD[2]	PE5	18
OSC_IN32	TMR1_CH4	A20	TXD[3]	PE6	34
OSC_OUT32	TMR2_CH1	A21	RXD[0]	PE7	35
-	TMR2_CH2	A22	RXD[1]	PE8	19
-	TMR2_CH3	A23	RXD[2]	PE9	20
-	TMR2_CH4	A24	RXD[3]	PE10	21
-	CAN_RX1	A25	TXEN	PE11	22
-	CAN_TX1	A26	TXER	PE12	23
-	CAN_RX2	A27	TXCLK	PE13	24
-	CAN_TX2	A28	RXCLK	PE14	25
-	PRD_PRMD	A29	RXDV	PE15	26
Порт F					
OSC_IN25	PRD_PRMA	READY	RXER	PF0	27
OSC_OUT25	PRD_PRMB	A30	CRS	PF1	28
-	READY /PRD_PRMC	A31	COL	PF2	29
-	PRMC+	A0	TMR1_CH1	PF3	79
-	PRMC-	A1	TMR1_CH2	PF4	80
-	PRMD+	A2	TMR1_CH3	PF5	81
-	PRMD-	A3	TMR1_CH4	PF6	82
-	PRDC+	A4	OUT4+	PF7	83
-	PRDC-	A5	OUT4-	PF8	84
-	PRDD+	A6	OUT3+	PF9	85
-	PRDD-	A7	OUT3-	PF10	86
-	PRD_PRMC	A8	OUT2+	PF11	87
-	PRD_PRMD	A9	OUT2-	PF12	88
-	OUT2+	A10	SSP3_FSS	PF13	64
-	OUT2-	A11	SSP3_SCK	PF14	65
-	SSP3_RXD	A12	SSP3_TXD	PF15	66
Системное управление					
-	nRESET	-	-	Сигнал внешнего сброса	55
-	OSC_IN	-	-	Вход генератора HSE	53
-	OSC_OUT	-	-	Выход генератора HSE	54
-	WAKEUP	-	-	Сигнал внешнего выхода из режима Standby	36
-	ITCMLAEN	-	-	Сигнал выбора памяти программ: 1- внутренняя; 0- внешняя.	15
USB интерфейс					
-	DP	-	-	Шина USB D+	37
-	DN	-	-	Шина USB D-	38
PHY Ethernet интерфейс					
-	TXP	-	-	Дифференциальный выход передатчика на трансформаторы	3
-	TXN	-	-		2
-	RXP	-	-	Дифференциальный вход приёмника с транформатора	7
	RXN	-	-		6
-	EXRES1	-	-	Вход для подключения опорного резистора 12,4 кОм 1% на VSS2A	10
Питание					
-	Ucc	-	-	Питание 2,0…3,6 В	31,71,117
-	AUcc	-	-	Аналоговое питание АЦП, ЦАП, PLL 2,4…3,6 В (должно совпадать с Ucc)	40

	VDD1A-VDD4A			Аналоговое питание PHY 3,0…3,6 В	5,8,12,14
-	BUcc	-	-	Батарейное питание 1.8…3,6 В	32
-	GND	-	-	Общий	30,70,116
-	AGND	-	-	Общий АЦП, ЦАП, PLL	39
-	VSS1A-VSS4A	-	-	Общий PHY	1,4,9,11,13
JTAG интерфейс					
-	TMS	-	-		128
-	TDI	-	-		129
-	TDO	-	-		130
-	TCK	-	-		131
-	TRST	-	-		132
-	JTAGEN	-	-	Вход разрешения отладочного TAP интерфейса на выводах JTAG	33

Примечание:
1. При использовании внешнего Ethernet PHY, посредством портов PE[15:2], PE[0] и PF[2:0], внутренний Ethernet PHY должен находиться в состоянии сброса (см. описание Блока PHY).
2. При использовании режима работы с ITCMLAEN=0 по умолчанию порты A и B настраиваются как двунаправленная 32-х разрядная шина данных, порт C[2:0] как сигналы CLKO, nRD, nWR, а порты PF[15:3], PD[15], PE[2:0] как шина адреса A[16:0]. Программно можно расширить адресную шину до требуемого количества разрядов.
3. Функции выводов выделенные красным цветом доступны с ревизии 3.

Описание выводов режима Stand Alone

Вывод порта		**Корпус**
Порт A		**4229.132-3**
DATA0	Двунаправленная шина данных	127
DATA1		126
DATA2		125
DATA3		124
DATA4		123
DATA5		122
DATA6		121
DATA7		120
DATA8		119
DATA9		118
DATA10		115
DATA11		114
DATA12		113
DATA13		112
DATA14		111
DATA15		110
Порт B		
ADDR0	Шина адреса	109
ADDR1		108
ADDR2		107
ADDR3		106
ADDR4		105
ADDR5		104
ADDR6		103
ADDR7		102
ADDR8		101
ADDR9		100
ADDR10		99
ADDR11		98
nBE_ETH0	Сигнал byte enable0 для Ethernet контроллера	97
nBE_ETH1	Сигнал byte enable1 для	96

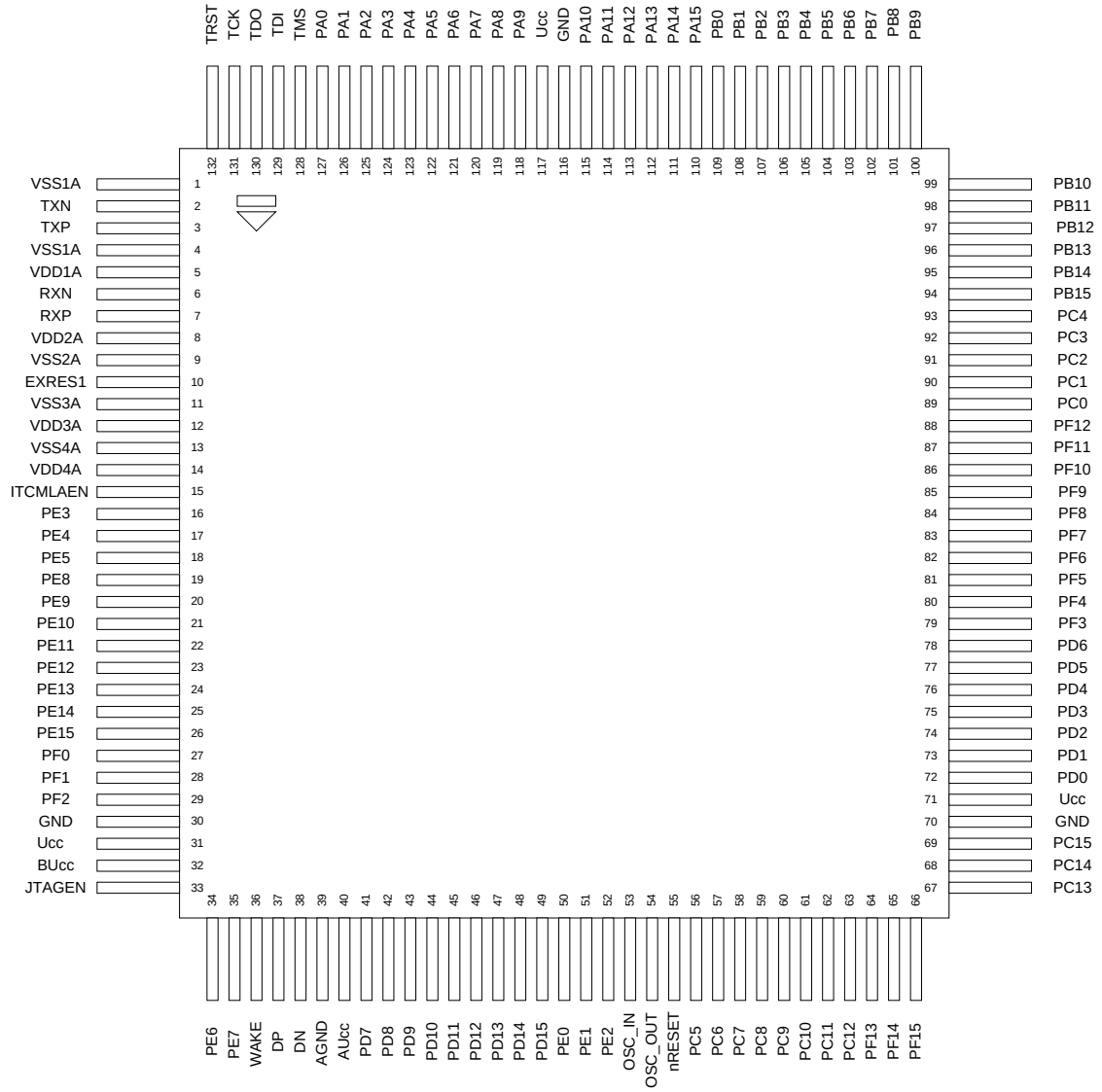
	Ethernet контроллера	
nCE1	Сигнал chip enable 1 для выбора области регистров Ethernet контроллера или выбора первого контроллера ГОСТ Р52070-2003	95
nCE2	Сигнал chip enable 2 для выбора области данных Ethernet контроллера или выбора второго контроллера ГОСТ Р52070-2003	94
Порт C		
nWE	Сигнал записи данных	89
nOE	Сигнал чтения данных	90
RDY_ETH	Сигнал готовности при записи/чтении регитстров и памяти Ethernet контроллера	91
ADDR12	Старший разряд шины адреса	92
DATA16	Двунаправленная шина данных только для контроллеров ГОСТ Р52070-2033	93
DATA17		56
DATA18		57
FXEN / INT1	Сигнал разрешения оптического выхода PHY или выход прерывания от первого контроллера ГОСТ Р52070-2003	58
FTX /INT2	Сигнал оптического выхода PHY или выход прерывания от второго контроллера ГОСТ Р52070-2003	59
-		60
-		61
-		62
-		63
PRMA+	Дифференциальный вход приёмника основного канала первого контроллера ГОСТ Р52070-2003	67
PRMA-		68
PRMB+	Дифференциальный вход приёмника резервного канала первого контроллера ГОСТ Р52070-2003	69
Порт D		
PRMB-	Дифференциальный вход приёмника резервного канала первого контроллера ГОСТ Р52070-2003	72
PRDA+	Дифференциальный выход передатчика основного канала первого контроллера ГОСТ Р52070-2003	73
PRDA-		74
PRDB+	Дифференциальный выход передатчика резервного канала первого контроллера ГОСТ Р52070-2003	75
PRDB-		76
PRD_PRMA	Сигнал разрешения передачи основного канала первого контроллера ГОСТ Р52070-2003	77
PRD_PRMB	Сигнал разрешения передачи резервного канала первого	78

	контроллера ГОСТ Р52070-2003	
		41
		42
		43
		44
FRX	Сигнал оптического входа PHY	45
FXEN	Сигнал разрешения оптического выхода PHY(с ревизии 3)	46
FTX	Сигнал оптического выхода PHY(с ревизии 3)	47
		48
FSD	Сигнал валидности данных на оптическом входе PHY	49
Порт E		
MDC	Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY	50
ETH_INT	Выход прерывания от Ethernet контроллера	51
MDIO	Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY. При этом внутренний PHY должен находиться в состоянии сброса	52
TXD[0]		16
TXD[1]		17
TXD[2]		18
TXD[3]		34
RXD[0]		35
RXD[1]		19
RXD[2]		20
RXD[3]		21
TXEN		22
TXER		23
TXCLK		24
RXCLK		25
RXDV		26
Порт F		
RXER	Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY	27
CRS		28
COL		29
PRMC+	Дифференциальный вход приёмника основного канала второго контроллера ГОСТ Р52070-2003	79
PRMC-		80
PRMD+	Дифференциальный вход приёмника резервного канала второго контроллера ГОСТ Р52070-2003	81
PRMD-		82
PRDC+	Дифференциальный выход передатчика основного канала второго контроллера ГОСТ Р52070-2003	83
PRDC-		84
PRDD+	Дифференциальный выход передатчика резервного канала второго контроллера ГОСТ Р52070-2003	85
PRDD-		86
PRD_PRMC	Сигнал разрешения передачи основного канала второго контроллера ГОСТ Р52070-2003	87
PRD_PRMD	Сигнал разрешения передачи резервного канала второго контроллера ГОСТ Р52070-2003	88
-		64
-		65

-		66
Системное управление		
nRESET	Сигнал внешнего сброса	55
OSC_IN	Вход генератора HSE	53
OSC_OUT	Выход генератора HSE	54
WAKEUP	Сигнал внешнего выхода из режима Standby	36
ITCMLAEN	Сигнал выбора доступа к контроллеру Eternet, ГОСТ Р52070-2003	15
PHY Ethernet интерфейс		
TXP	Дифференциальный выход передатчика на трансформаторы	3
TXN		2
RXP	Дифференциальный вход приёмника с транформатора	7
RXN		6
EXRES1	Вход для подключения опорного резистора 12,4 кОм 1% на VSS2A	10
Питание		
Ucc	Питание 2,0…3,6 В	31,71,117
AUcc	Аналоговое питание АЦП, ЦАП, PLL 2,4…3,6 В (должно совпадать с Ucc)	40
VDD1A-VDD4A	Аналоговое питание PHY 3,0…3,6 В	5,8,12,14
BUcc	Батарейное питание 1.8…3,6 В	32
GND	Общий	30,70,116
AGND	Общий	39
VSS1A-VSS4A	Общий	1,4,9,11,13

Примечание: максимальная частота работы SRAM интерфейса не более 50 МГц.

Диаграмма расположения выводов в корпусе



132-х выводной металлокерамический корпус 4229.132-3

Питание микросхемы

Микроконтроллер имеет несколько типов выводов питания

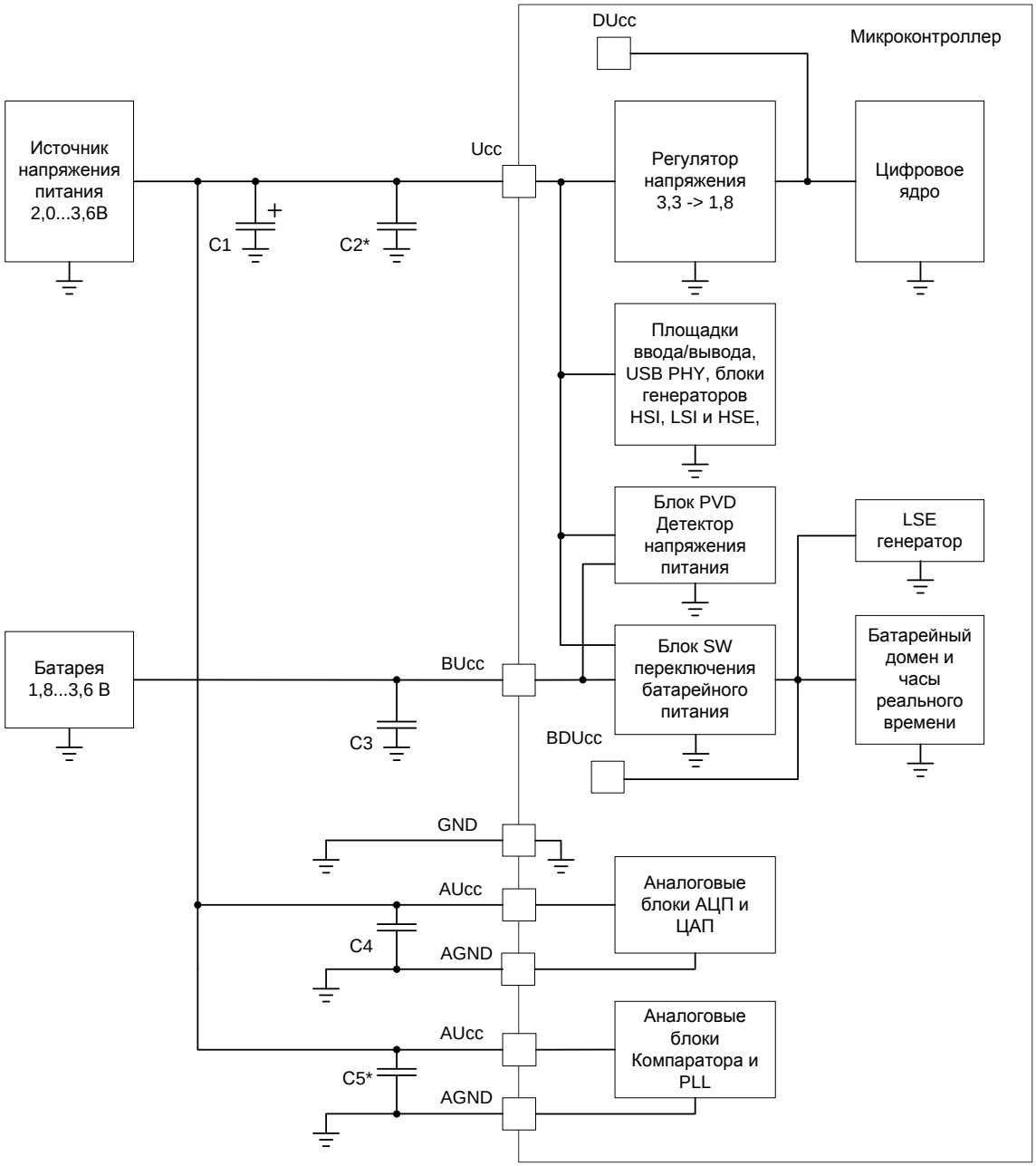
- **Ucc выводы:** основное питание микросхемы, включает питание пользовательских выводов, встроенного регулятора напряжения, USB PHY и генераторов. Входное напряжение должно быть в пределах от 2,2 до 3,6 В. Если используется интерфейс USB, то входное напряжение должно быть в пределах от 3,0 до 3,6 В. Если используется АЦП или ЦАП, то входное напряжение должно быть в пределах от 2,4 до 3,6 В;

- **BUcc вывод:** питание батарейного домена используется при отсутствии основного питания Ucc для питания батарейного домена и LSE генератора. Переключение с основного питания на батарейное происходит автоматически при снижении уровня Ucc ниже 2,0В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 4 мс после превышения уровнем Ucc 2,0В. Входное напряжение должно быть в пределах от 1,8 до 3,6 В. Если в системе не требуется батарейного питания вывод BUcc должен быть объединен с Ucc;

- **AUcc вывод:** питание аналоговых блоков АЦП, ЦАП, компаратора и PLL выведено на отдельные выводы для уменьшения помех создаваемых работой других блоков. На данные выводы должно подаваться напряжения с того же источника что и Ucc, но при этом на печатной плате должны быть применены меры по снижению наводки помех. Для корректной работы АЦП входное напряжение должно быть в пределах от 2,4 до 3,6 В. Если входное напряжение будет в пределах от 2,2 до 2,4В, то корректная работа АЦП не гарантируется;

- **GND выводы:** общие выводы цифровой части микросхемы;

- **AGND вывод:** общий вывод аналоговой части микросхемы. Данный вывод должен соединяться с GND, но при этом на печатной плате должны быть применены меры по снижению наводки от помех.



Примечание:

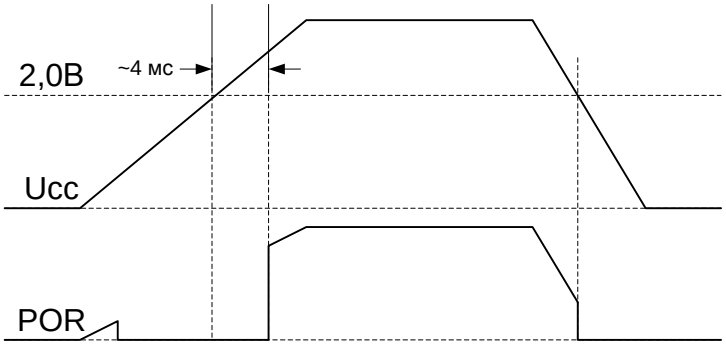
* - конденсаторы должны быть установлены у каждого вывода питания

1. Конденсатор C1 = 22 мкФ, C2 = C3 = C4 = C5 = 0,1 мкФ;
2. Если не используется батарейное питание, то вывод BUcc должен быть объединен с Ucc;
3. Если используется интерфейс USB, то напряжение питания Ucc должно быть в пределах от 3,0 до 3,6 В;
4. Если используется АЦП или ЦАП, то напряжение питания Ucc (AUcc) должно быть в пределах от 2,4 до 3,6 В;

Микроконтроллер имеет встроенный детектор напряжения питания, подробнее смотри раздел «Детектор напряжения питания».

Схема сброса при включении и выключении основного питания.

При включении питания вырабатывается внутренний сигнал сброса POR для цифровой части, питание Ucc нарастает и, пока оно не превысило уровень 2,0 В, сигнал сброса POR удерживается; после превышения данного уровня сигнал POR выдается еще на протяжении ~ 4 мс для того, чтобы гарантировано установилось напряжение питания, после чего сигнал POR снимается, и схема может начать работать.

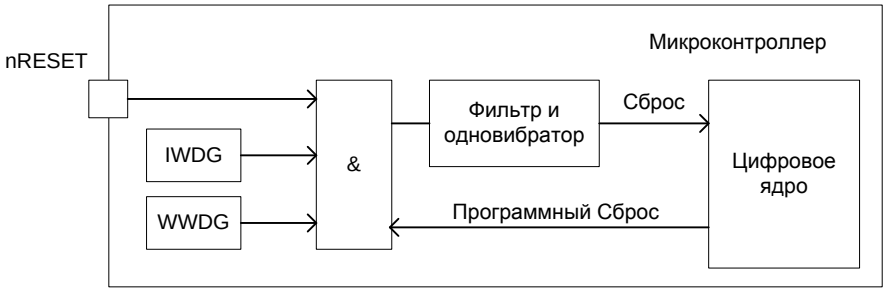


При снижении напряжения питания Ucc ниже уровня 2,0 В сигнал POR вырабатывается без задержки.

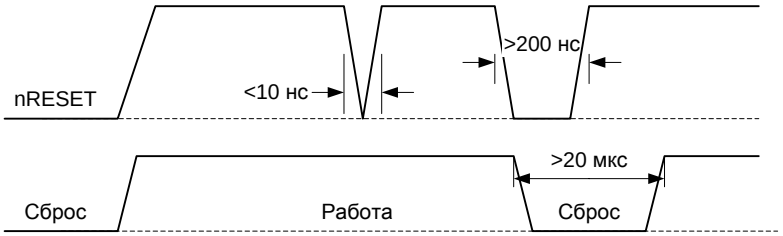
Сигнал POR также служит для переключения питания батарейного домена между BUcc и Ucc.

При включении основного напряжения питания Ucc автоматически включается встроенный регулятор напряжения для формирования напряжения DUcc питания цифрового ядра. В ходе работы микроконтроллера встроенный регулятор может быть отключен.

Начальная установка микроконтроллера может быть произведена внешним сигналом сброса nRESET, или внутренними сигналами сброса сторожевых таймеров, или программным сбросом. При этом сигнал nRESET формируется специальной схемой сброса, содержащий фильтр «иголок» и одновибратор для увеличения длительности этого сигнала.

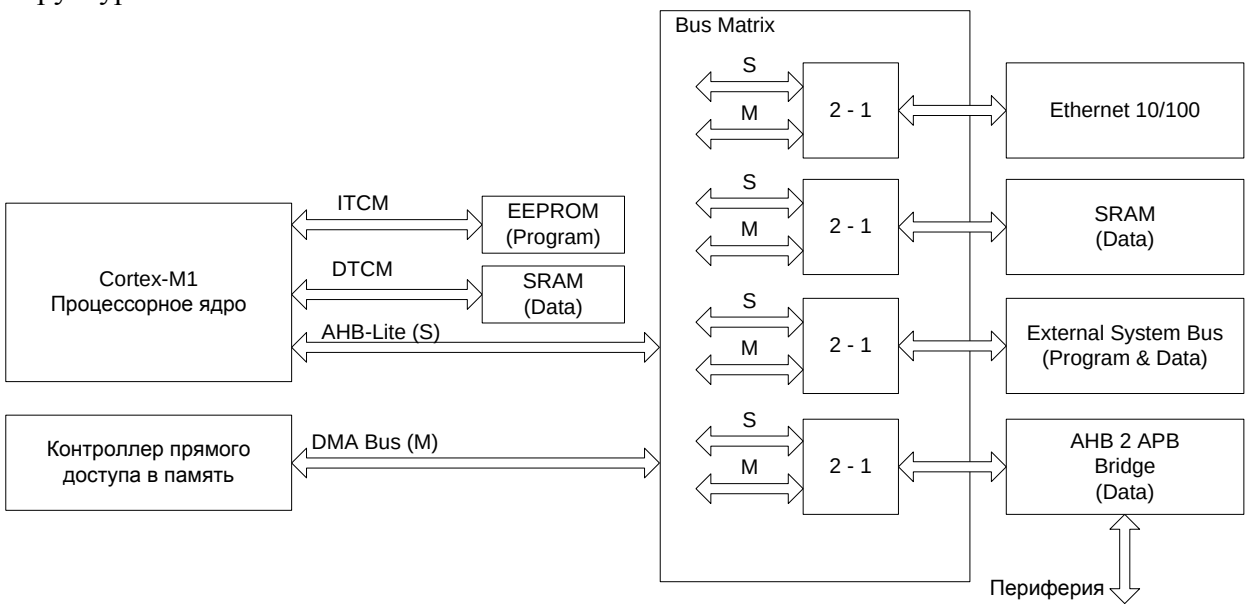


При подаче на вход nRESET импульсов сброса длительностью менее 10 нс они отфильтровываются и не приводят к сбросу процессора. Если длительность импульса больше 200 нс, вырабатывается сигнал сброса. При этом длительность сформированного сигнала сброса будет не менее 20 мкс.



Организация памяти

Структурная схема



Процессорное ядро имеет три системные шины:

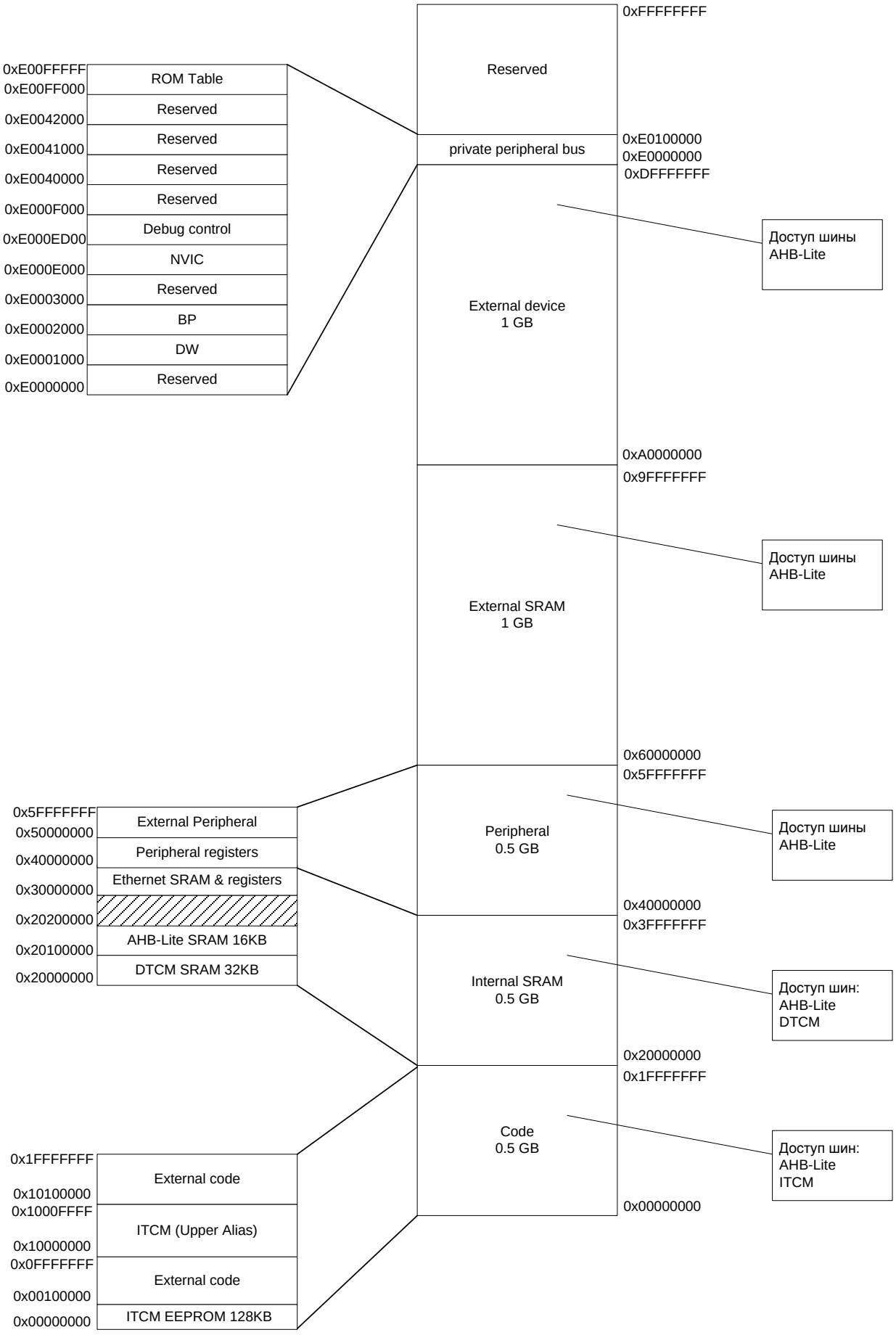
ITCM Bus – шина выборки инструкций и данных, расположенный в коде программы;

DTCM Bus – шина выборки данных, расположенный в области ОЗУ;

AHB-Lite – шина выборки инструкций и данных с внешнего адресного пространства.

Также в микроконтроллере реализован контроллер прямого доступа в память (DMA) осуществляющий выборку через шину DMA Bus.

Все адресное пространство микроконтроллера единое и имеет максимальный объем 4 Гбайта. В данное адресное пространство отображаются различные модули памяти и периферии.


0xFFFFFFFF
Reserved
0xE0100000
private peripheral bus
0xE0000000
0xDFFFFFFF
External device
1 GB
Доступ шины
AHB-Lite
0xA0000000
0x9FFFFFFF
External SRAM
1 GB
Доступ шины
AHB-Lite
0x60000000
0x5FFFFFFF
Peripheral
0.5 GB
Доступ шины
AHB-Lite
0x40000000
0x3FFFFFFF
Internal SRAM
0.5 GB
Доступ шин:
AHB-Lite
DTCM
0x20000000
0x1FFFFFFF
Code
0.5 GB
Доступ шин:
AHB-Lite
ITCM
0x00000000
0xE00FFFFF
0xE00FF000
ROM Table
Reserved
0xE0042000
Reserved
0xE0041000
Reserved
0xE0040000
Reserved
0xE000F000
Debug control
0xE000ED00
NVIC
0xE000E000
Reserved
0xE0003000
BP
0xE0002000
DW
0xE0001000
Reserved
0xE0000000
0x5FFFFFFF
0x50000000
External Peripheral
0x40000000
Peripheral registers
0x30000000
Ethernet SRAM & registers
0x20200000
0x20100000
AHB-Lite SRAM 16KB
0x20000000
DTCM SRAM 32KB
0x1FFFFFFF
0x10100000
External code
0x1000FFFF
0x10000000
ITCM (Upper Alias)
0x0FFFFFFF
0x00100000
External code
0x00000000
ITCM EEPROM 128KB

Секция Code

Область ITCM EEPROM:

Основная область энергонезависимой памяти программы, доступной для перепрограммирования пользователем. Память предназначена для хранения основной рабочей программы на кристалле.

Область External code:

Область отображения шины AHB-Lite в адресное пространство области программы. Предназначена для хранения кода программ во внешних микросхемах памяти, подсоединенных к внешней системной шине.

Секция Internal SRAM

Область DTCM SRAM:

Основная область ОЗУ, предназначенная для хранения данных программы. В данной области так же располагается стек (stack) и «куча» (heap) программы. Адресные диапазоны стека и «кучи» задаются пользователем при написании программы.

Область AHB-Lite SRAM:

Область отображения шины AHB-Lite в адресное пространство области данных. Предназначена для хранения данных в памяти микросхемы подсоединенной к шине AHB-Lite.

Область Ethernet SRAM & registers:

Область отображения шины AHB-Lite в адресное пространство регистров и памяти контроллера Ehernet. Предназначена для хранения данных в памяти микросхемы подсоединенной к шине AHB-Lite.

Секция Peripheral

Область Peripheral registers:

Область отображения регистров периферии в общее адресное пространство памяти.

Область External Peripheral:

Область отображения внешней системной шины в адресное пространство области периферии. Предназначена для хранения данных во внешних микросхемах памяти или работы с периферийными устройствами, подсоединенными к внешней системной шине.

Секция External SRAM и External device

Область отображения шины AHB-Lite в адресное пространство области внешней памяти и внешних устройств. Предназначена для хранения данных во внешних микросхемах памяти или работы с внешними устройствами, подсоединенными к внешней системной шине.

Секция Private Peripheral Bus

Предназначена для отображения системных регистров ядра и системной периферии.

Блок BUS MATRIX

Блок BUS MATRIX предназначен для переключения шин AHB-Lite и DMA Bus между различными областями памяти. Переключение производится автоматически на основании адреса запроса каждой конкретной шины. Если адреса запросов не пересекаются, то они могут быть выполнены одновременно. Если адреса запросов пересекаются, то они выполняются в порядке приоритета. Приоритеты обращений заданы аппаратно. Наивысшим приоритетом обладает запрос по шине AHB-Lite, а наименьшим приоритетом обладает запрос DMA Bus. Если два запроса пришли одновременно, то выполняется запрос с большим приоритетом, а запрос с более низким приоритетом задерживается до окончания обработки запроса с более высоким приоритетом. При переключении между шинами возникает дополнительная задержка в один цикл. Если запросы идут непосредственно друг за другом, то дополнительных задержек не возникает.

Память EEPROM

Память области EEPROM реализована в виде перепрограммируемой энергонезависимой памяти. Скорость доступа к памяти EEPROM – порядка 40 нс. При работе микроконтроллера на скорости до 100 МГц, скорость доступа к памяти может составлять до 5 циклов системной частоты. При последовательной выборке за счет упреждающего чтения задержка может быть сокращена до 1 цикла системной частоты. Более подробная информация о работе контроллера EEPROM памяти программ представлена в разделе контроллер EERPOM.

Память SRAM

Память области SRAM реализована в виде блока статической памяти. Скорость доступа к памяти SRAM – 1 цикл системной частоты.

Регионы памяти, типы и атрибуты

Отображение памяти разбивает все адресное пространство на регионы. Каждый регион имеет определенный тип памяти, а некоторые регионы имеют дополнительные атрибуты. Тип памяти и атрибуты определяют поведение системы при доступе к этим регионам.

Normal

Процессор может переопределить последовательность обращений для большей эффективности или проведения спекулятивного считывания

Device

Процессор сохраняет последовательность обращений по отношению к другим обращениям в области Device и Strongly-ordered памяти

Strongly-ordered

Процессор сохраняет последовательность обращений по отношению ко всем обращениям. Отличие последовательности для Device и Strongly-Ordered памяти

означает, что система памяти может буферизировать запись в Device, но никогда не буферизирует запись в Strongly-ordered память.

Shareable

Для shareable регионов система памяти обеспечивает синхронизацию между различными мастерами на шине, например, DMA и само ядро. Strongly-ordered память всегда shareable. При наличии работы нескольких мастеров в не shareable памяти, программное обеспечение должно отслеживать когерентность данных различных мастеров.

Execution Never (XN)

Область памяти, из которой не могут извлекаться инструкции. Любая попытка извлечь инструкцию из XN региона приведет к исключению Hard fault.

Последовательность обращений в память:

Для большинства обращений в память, выполняемых инструкциями доступа, система памяти не гарантирует последовательность их выполнения в соответствии с последовательностью выполнения этих инструкций, за исключением, когда эта последовательность может повлиять на последовательность инструкций. Обычно, когда выполнение программы требует последовательно выполнить два обращения в память, то программно должна быть выполнена барьерная инструкция между инструкциями обращения. Смотри раздел «Программное определение последовательности доступа в память».

Однако арбитр памяти гарантирует однозначную последовательность доступа в регионы памяти Device и Strongly-ordered. Для двух инструкций доступа в память A1 и A2, если A1 выполняется перед A2 в коде программы, последовательность обращений двух инструкций будет такой, как показано в таблице ниже.

A2 / A1	Доступ в Normal	Доступ в Device Не shareable	Доступ в Device shareable	Strongly-ordered
Доступ в Normal	-	-	-	-
Доступ в Device Не shareable	-	<	-	<
Доступ в Device shareable	-	-	<	<
Strongly-ordered	-	<	<	<

Где « - » означает что система памяти не гарантирует последовательность выполнения обращений, а « < » означает что обращение инструкции A1 всегда будет выполнено перед инструкций A2.

Поведение обращений в память

Поведение обращений описано в таблице

Адресный диапазон	Регион памяти	Тип памяти	XN	Описание
0x00000000-0x000FFFFF	Code, ITCM, Lower Alias	Normal	-	Область памяти шины ITCM для кода программы и данных, если ITCMLAEN установлен в единицу. Если ITCMLAEN равен нулю, то область памяти внешней системной шины.
0x00100000-0x0FFFFFFF	Code, external	Normal	-	Область памяти внешней системной шины для кода программы и данных.
0x10000000-0x1000FFFF	Code, ITCM, Upper Alias	Normal	-	Область памяти шины ITCM для кода программы и данных, если ITCMUAEN установлен в единицу. Если ITCMUAEN равен нулю, то область памяти внешней системной шины.
0x10010000-0x1FFFFFFF	Code, external	Normal	-	Область памяти внешней системной шины для кода программы и данных.
0x20000000-0x200FFFFF	SRAM, DTCM	Normal	XN	Область памяти шины DTCM для данных. Доступ за инструкцией в эту область приводит к ошибке.
0x20100000-0x3FFFFFFF	SRAM	Normal		Область памяти внешней системной шины для кода программы и данных.
0x40000000-0x5FFFFFFF	Peripheral	Device	XN	Область памяти внешней системной шины для данных. Доступ за инструкцией в эту область приводит к ошибке
0x60000000-0x9FFFFFFF	External SRAM	Normal	-	Область памяти внешней системной шины для кода программы и данных.
0xA0000000-0xDFFFFFFF	External Device	Device	XN	Область памяти внешней системной шины для внешних устройств. Доступ за инструкцией в эту область приводит к ошибке
0xE0000000-0xE00FFFFF	Private Peripheral Bus	Strongly-ordered	XN	Этот регион содержит регистры NVIC, system timer и регистры блока управления ядра
0xE0100000-0xFFFFFFFF	Зарезервировано	-	XN	Зарезервировано

Программное определение последовательности доступов в память

Последовательность инструкций в потоке программы не всегда гарантирует последовательность соответствующих обращений в память, это происходить потому, что:

- процессор может изменить последовательность обращений для увеличения производительности, но при этом не изменяется общее поведение программы;
- процессор имеет несколько шин обращений в память;
- память или устройства могут иметь различные скорости доступа;
- некоторые обращения в память буферизируются или выполняются спекулятивно.

Этот раздел описывает, как гарантировать при необходимости корректную последовательность обращений в память. Если последовательность обращений в память критично, программное обеспечение должно содержать инструкции барьерной синхронизации для задания корректной последовательности. Процессор предлагает следующие инструкции барьерной синхронизации:

DMB

Инструкция Data Memory Barrier (DMB) позволяет быть уверенным, что выполняемая инструкция транзакции в память будет завершена до следующей транзакции в память. Смотри описание инструкции DMB.

DSB

Инструкция Data Synchronization Barrier позволяет быть уверенным, что выполняемая инструкция транзакции в память будет завершена до начала выполнения следующей инструкции. Смотри описание инструкции DSB.

ISB

Инструкция Instruction Synchronization Barrier позволяет быть уверенным, что эффект всех выполняемых транзакций в память полностью соответствует инструкциям. Смотри описание инструкции ISB.

Инструкции барьерной синхронизации используются, например, при:

- Само модифицируемый код. Если программа содержит само модифицируемый код, используйте ISB инструкцию сразу после модификации кода программы. Это необходимо для уверенности, что после этого будет выполняться уже модифицированный код.
- Переключение карты памяти. Если система содержит механизм переключения карты памяти, то используйте инструкцию DSB после переключения карты памяти в программе. Это гарантирует, что дальнейшее выполнение инструкций будет идти с новой картой памяти.
- Динамическое изменение приоритетов исключений. Когда приоритеты исключений изменяются при обработке исключительной ситуации, используйте DSB инструкцию после изменения. Это гарантирует, что изменение произойдет при завершении DSB инструкции.
- Использование семафоров или много-мастерная система. Если система содержит несколько мастеров, например другой процессор, то оба процессора должны использовать DMB инструкции после инструкций работы с семафорами. Это гарантирует, что другой мастер будет видеть транзакции в той последовательности, как они выполняются.

При обращениях в память Strongly-ordered, такую как системный блок управления ядра (NVIC, System Timer и так далее) не требуется использовать DMB инструкции.

Базовые адреса процессора

Адрес	Размер	Блок	Примечание
Память программ			
0x0000_0000		EEPROM BOOT ROM	Область Flash памяти программ с пользовательской программой
0x0010_0000		External code	Область доступа к внешней системной шине
Память данных			
0x2000_0000		DTCM SRAM	Область внутреннего ОЗУ
0x2010_0000		AHB-Lite SRAM	Область внутреннего ОЗУ
0x3000_0000		Ethernet	Область внутреннего ОЗУ и регистров
Периферия			
0x4000_0000		CAN1	Регистры контроллера интерфейса CAN1
0x4000_8000		CAN2	Регистры контроллера интерфейса CAN2
0x4001_0000		USB	Регистры контроллера интерфейса USB
0x4001_8000		EEPROM_CNTRL	Регистры контроллера Flash памяти программ
0x4002_0000		RST_CLK	Регистры контроллера сигналов тактовой частоты
0x4002_8000		DMA	Регистры контроллера прямого доступа в память
0x4003_0000		UART1	Регистры контроллера интерфейса UART1
0x4003_8000		UART2	Регистры контроллера интерфейса UART2
0x4004_0000		SPI1	Регистры контроллера интерфейса SSP1
0x4004_8000		MIL-STD-1553B1	Регистры контроллера интерфейса MIL-STD-1553B канал 1
0x4005_0000		MIL-STD-1553B2	Регистры контроллера интерфейса MIL-STD-1553B канал 2
0x4005_8000		POWER	Регистры детектора напряжения питания
0x4006_0000		WWDT	Регистры контроллера сторожевого таймера WWDT
0x4006_8000		IWDT	Регистры контроллера сторожевого таймера IWDT
0x4007_0000		TIMER1	Регистры управления Таймер 1
0x4007_8000		TIMER2	Регистры управления Таймер 2
0x4008_0000		TIMER3	Регистры управления Таймер 3
0x4008_8000		ADC	Регистры управления АЦП
0x4009_0000		DAC	Регистры управления ЦАП
0x4009_8000		TIMER4	Регистры управления Таймер 4
0x400A_0000		SPI2	Регистры контроллера интерфейса SSP2
0x400A_8000		PORTA	Регистры управления порта A
0x400B_0000		PORTB	Регистры управления порта B
0x400B_8000		PORTC	Регистры управления порта C
0x400C_0000		PORTD	Регистры управления порта D
0x400C_8000		PORTE	Регистры управления порта E
0x400D_0000		ARINC429R	Регистры контроллера интерфейса приёмников ARINC429
0x400D_8000		BKP	Регистры доступа и управления батарейным доменом

0x400E_0000		ARINC429T	Регистры контроллера интерфейса передатчиков ARINC429
0x400E_8000		PORTF	Регистры управления порта F
0x400F_0000		EXT_BUS_CNTRL	Область доступа к внешней системной шине
0x400F_8000		SPI3	Регистры контроллера интерфейса SSP3
0x5000_0000		External peripheral	Область доступа к внешней системной шине
Внешняя системная шина			
0x6000_0000		External SRAM	Область доступа к внешней системной шине
0xA000_0000		External device	Область доступа к внешней системной шине
Private Peripheral Bus			
0xE000_0000			Системные регистры процессора

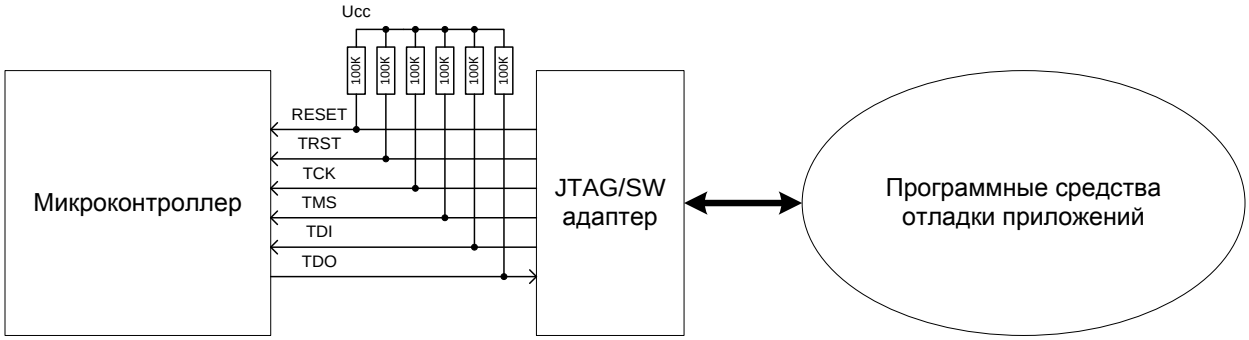
Загрузочное ПЗУ и режимы работы микроконтроллера

После включения питания и снятия внутренних (POR) и внешних (RESET) сигналов сброса, микроконтроллер начинает выполнять программу из загрузочной области ПЗУ BOOT ROM. В загрузочной программе микроконтроллер определяет, в каком из режимов он будет функционировать, и переходит в этот режим. Режим функционирования определяется внешними выводами MODE[2:0] (PA[2:0]). Также устанавливается бит FPOR в регистре BKP_REG_0E, который может быть сброшен только при отключении основного питания Ucc. После перезапуска микроконтроллера уровни на выводах MODE[2:0] не влияют на режим функционирования микроконтроллера, если установлен бит FPOR. В пользовательской программе выводы PA[2:0] могут использоваться пользователем.

MODE[2:0]	Режим	Стартовый адрес/ таблица векторов прерываний	Описание
000	Микроконтроллер с режимом отладки	0x0000_0000	Процессор начинает выполнять программу из внутренней FLASH памяти программ. При этом разрешается работа отладочного интерфейса JTAG Сигнал выбора памяти программ: ITCMLAEN=1- внутренняя память. ITCMLAEN=0- внешняя память.
001	Режим Stand alone1	0x0000_0000	Процессор конфигурирует прямой доступ к контроллеру Ethernet с помощью внешней системной шины и переходит в режим сброса. Частота задаётся внешним генератором HSE, умноженная на 11 с помощью PLL. Адрес – {PC[3],PB[11:0]} Данные – PA[15:0] Byte enable – PB[13:12] Chip enable – PB[15:14] Write enable – PC[0] Output enable – PC[1] Вход ITCMLAEN=1.
010	Режим Stand alone2	0x0000_0000	Процессор конфигурирует прямой доступ к контроллеру интерфейса ГОСТ Р52070-2003 с помощью внешней системной шины и переходит в режим сброса. Частота задаётся внешним генератором HSE, умноженная на 11 с помощью PLL. Адрес – {PC[3],PB[11:0]} Данные – {PC[6:4],PA[15:0]} Chip enable – PB[15:14] Write enable – PC[0] Output enable – PC[1]

			Вход ITCMLAEN=0.
011	Режим Stand alone3	0x0000_0000	Процессор конфигурирует прямой доступ к контроллерам Ethernet и интерфейса ГОСТ Р52070-2003 с помощью внешней системной шины и переходит в режим сброса. Частота задаётся внешним генератором HSE, умноженная на 11 с помощью PLL. Вход ITCMLAEN=1 – доступ к контроллеру Ethernet. Вход ITCMLAEN=0 – доступ к контроллеру интерфейса ГОСТ Р52070-2003.
100-110	UART загрузчик	0x0000_0000	Микроконтроллер через интерфейс UART1 на выводах PC[4:3] получает код программы в ОЗУ для исполнения
111	Зарезервировано	-	-

При работе в режиме отладки разрешается работа отладочного интерфейса JTAG/SW. При этом к микроконтроллеру может быть подключен JTAG/SW адаптер, с помощью которого программные средства разработки позволяют работать с микроконтроллером в отладочном режиме.



В отладочном режиме можно:
- стирать, записывать, считывать внутреннюю FLASH память программ
- считывать и записывать содержимое ОЗУ, периферии
- выполнять программу в пошаговом режиме
- запускать программу в нормальном режиме
- останавливать программу по точкам остановки
- просматривать переменные выполняемой программы
- проводить трассировку хода выполнения программного обеспечения

В зависимости от режима работы выводы интерфейса JTAG/SW переопределяются на различные выводы микроконтроллера, представленные в таблице

Вывод JTAG/SW	Вывод микроконтроллера	Описание
JTAG		
TRST	TRST	В качестве выводов интерфейса используются выделенные выводы микросхемы.
TCK	TCK	
TMS	TMS	
TDI	TDI	
TDO	TDO	
JTAG_EN	JTAG_EN	Необходимо доопределить в логический “ноль”.
SW		
SWCLKTCK	TCK	Вход тактовой частоты
SWDITMS	TMS	Двунаправленные данные

UART загрузчик

В режиме UART загрузчика используют один и тот же периферийный модуль UART1, один и тот же протокол обмена (см. таблицу ниже).

Режим	Rx	Tx
100b	PC4	PC3
101b	PC4	PC3
110b	PC4	PC3

Данные режимы предоставляют достаточный набор операций, необходимых для записи в ОЗУ какой-либо программы (в частности программатора Flash-памяти), верификации ее и запуска на выполнение. Кроме того, существует возможность задания внешним устройством скорости обмена. Помимо доступа к ОЗУ может быть осуществлен доступ и к другим адресным диапазонам (EERPOM, ROM, Периферия)

В качестве источника тактовой частоты UART1 используется внутренний RC-генератор HSI с частотой 8 МГц. Так как имеется разброс значений частоты HSI, то требуется этап подбора значения делителя частоты UART1 для синхронизации с внешним устройством.

Параметры связи по UART

Для связи по UART выбраны следующие параметры канала связи:
Начальная скорость – 9600 бод
Количетво бит данных – 8
Четность – нет
Количество Stop бит – 1
Загрузчик не использует FIFO UART1.
Загрузчик всегда выступает в качестве Slave, а внешнее устройство, подающее команды – в качестве Master.
Данные передаются младшим битом вперед.

Протокол обмена по UART

После синхронизации с Master’ом загрузчик переходит в диспетчер команд.
Таким образом, Master’у доступны следующие команды:

Команды UART загрузчика

Команда	Код	ASCII Символ	Описание
CMD_SYNC	0x00		Пустая команда. Загрузчик ее принимает, но ничего по ней не делает.
CMD_CR	0x0D		Выдача приглашения Master-у.
CMD_BAUD	0x42	'B'	Установка скорости обмена.
CMD_LOAD	0x4C	'L'	Загрузка массива байт.
CMD_VFY	0x59	'Y'	Выдача массива байт.
CMD_RUN	0x52	'R'	Запуск программы на выполнение.

Синхронизация с внешним устройством

Начальные условия
На этапе синхронизации с внешним устройством (Master'ом) вывод Rx используется как вход. Master постоянно посылает в канал синхросимвол 0x00. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ощибки обмена. Как только загрузчик настроил скорость, он переходит в диспетчер команд и выдает приглашение (3 байта 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>'),) Master'у.
Master завершает выдачу синхросимволов и теперь может подавать команды согласно протоколу обмена.

Команда CMD_SYNC

Пустая команда.
Загрузчик (Slave) ее принимает, но ничего по ней не делает. Код команды соответствует символу синхронизации.

Команда CMD_SYNC

Код команды	CMD_SYNC = 0x00
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master: выдает код команды CMD_SYNC.	Slave: если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды.

Команда CMD_CR

Выдача приглашения Master'у

Команда CMD_CR

Код команды	CMD_CR = 0x0D
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master: выдает код команды CMD_CR.	Slave: если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD

	и завершает обработку текущей команды. Выдает код команды CMD_CR. Выдает код 0x0A. Выдает код 0x3E (ASCII символ '>')

Команда CMD_BAUD

Установка скорости обмена

Команда CMD_BAUD

Код команды	CMD_BAUD = 0x42
ASCII символ, соответствующий коду команды	'B'
Количество параметров команды	1
Параметр	Новое значение скорости обмена [бод].
Формат команды:	
Master: выдает код команды CMD_BAUD	Slave: если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master: выдает параметр	Slave: если параметр принят с ошибками, то выдает код ошибки ERR_CHN или ERR_BAUD и завершает обработку текущей команды. Выдает код команды CMD_BAUD. Устанавливает новое значение скорости обмена.

Команда CMD_LOAD

Загрузка массива байт в память микроконтроллера

Команда CMD_LOAD

Код команды	CMD_LOAD = 0x4C
ASCII символ, соответствующий коду команды	'L'
Количество параметров команды	2
Параметр 1.	Адрес памяти приемника данных.
Параметр 2.	Размер массива в байтах
Формат команды:	
Master: выдает код команды CMD_LOAD	Slave: если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды.
Master: выдает параметр 1.	Slave: если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды.
Master: выдает параметр 2.	Slave: если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей

	команды. Выдает код команды CMD_LOAD.
Master: выдает массив байт младшим байтом вперед.	Slave: принимает массив байт. Если хотя бы один байт принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончанию принятия массива выдает код ответа REPLY_OK = 0x4B ('K').

Команда CMD_VFY

Выдача массива байт из памяти микроконтроллера

Команда CMD_VFY

Код команды	CMD_VFY = 0x59
ASCII символ, соответствующий коду команды	'Y'
Количество параметров команды	2
Параметр 1	Адрес памяти источника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master: выдает код команды CMD_VFY	Slave: если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: выдает параметр 1	Slave: если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды
Master: выдает параметр 2	Slave: если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_VFY. Выдает массив байт младшим байтом вперед. По окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K')

Команда CMD_RUN

Запуск программы на выполнение.

Команда CMD_RUN

Код команды	CMD_RUN = 0x52
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр.	Адрес таблицы векторов загруженной программы
Формат команды:	
Master: выдает код команды CMD_RUN.	Slave: если команда принята с ошибками, то

	выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: выдает параметр.	Slave: если параметр принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_RUN. Устанавливает значение MSP и PC согласно таблице векторов (NVIC не перепрограммируется) и, таким образом, Slave завершает свое выполнение

Прием параметров команды

Параметры команд – это 4-х байтные числа.
Параметры передаются младшим байтом вперед.
В качестве значения параметра запрещено использовать число 0xFFFFFFFF.
Если при приеме параметра обнаружена аппаратная ошибка (UART установил в '1' какой-либо из флагов ошибки), то прием параметров не прекращается.
Анализ всех видов ошибок, связанных с передачей параметров, загрузчик производит только после принятия всех параметров команды.

Сообщения об ошибках

Сообщения об ошибках – это 2-х байтные последовательности символов. Первый символ всегда 0x45 ('E'). Второй символ определяет тип ошибки.
После выдачи сообщения об ошибке загрузчик переходит в режим ожидания следующей команды, поэтому Master после получения такого сообщения должен прекратить передачу байт, относящихся к текущей команде.
После принятия сообщения об ошибке Master должен подавать команду CMD_CR до тех пор, пока не получит корректный ответ, соответствующий этой команде.
Возможны следующие сообщения об ошибках: ERR_CHN, ERR_CMD, ERR_BAUD.

Ошибка ERR_CHN

Аппаратная ошибка UART.
Код ошибки 0x69 ('i').
Выдается диспетчером команд, если UART установил в '1' один из аппаратных флагов ошибки при приеме очередного байта.

Ошибка ERR_CMD

Принята неизвестная команда.
Код ошибки 0x63 ('c').
Выдается диспетчером команд, если принят неизвестный код команды.

Ошибка ERR_BAUD

Принята неизвестная команда.
Код ошибки 0x62 ('b').
Выдается диспетчером команд, если по принятому от Master'а значению скорости обмена невозможно вычислить корректное значение делителя частоты UART.

Контроллер FLASH памяти программ.

Микроконтроллер содержит встроенную Flash память программ с объемом 128 Кбайт основной памяти программ и 4 Кбайта информационной памяти. В обычном режиме (бит CON = 0, регистр EEPROM_CMD) доступна основная память программ через системную шину ITCM для выборки инструкций и данных кода программы. В режиме программирования (бит CON=1, регистр EEPROM_CMD) основная и информационная память доступны как периферийное устройство и могут быть использованы для нужд разработчика приложения. В режиме программирования программный код должен выполняться из области шины AHB-Lite или ОЗУ. Выполнение программного кода из Flash памяти программ в режиме программирования невозможно.

Работа Flash памяти программ в обычном режиме

Скорость доступа во Flash память ограничена и составляет порядка 40 нс, в результате выдача новых значений из Flash памяти может происходить с частотой не более 25 МГц. Для того, что бы процессорное ядро могло получать новые инструкции на больших частотах, в микроконтроллере реализуется Flash память с физической организацией 32К на 128 разрядов. Таким образом, за 40 нс из Flash памяти извлекается 16 байт, в которых может быть закодировано от 4 до 8 инструкций процессора. И пока ядро выполняет эти инструкции, из памяти извлекается следующая порция данных. Таким образом, тактовая частота может превышать частоты извлечения данных из памяти в несколько раз при линейном выполнении программы. При возникновении переходов, в выполненяемой программе, когда из памяти программ не выбраны нужные инструкции, возникает пауза в несколько тактов процессора для того, чтобы данные успели считаться из Flash. Число тактов паузы зависит от тактовой частоты процессора, так при работе с частотой ниже 25 МГц пауза не требуется, так как Flash память успевает выдать новые данные за один такт, при частоте от 25 до 50 МГц требуется один такт паузы и так далее. Число тактов паузы задается в регистре EEPROM_CMD битами Delay[2:0]. В таблице приведены характеристики необходимой паузы для работы Flash памяти программ.

Delay[2:0]	Тактов паузы	Тактовая частота	Примечание
0x00	0	До 25 МГц	
0x01	1	До 50 МГц	
0x02	2	До 75 МГц	
0x03	3	До 100 МГц	
0x04	4	До 125 МГц	Установлено по умолчанию после сброса
0x05	5	До 150 МГц	Работа с частотой более 144 МГц не гарантируется
0x06	6	До 175 МГц	
0x07	7	До 200 МГц	

Число тактов паузы устанавливается до момента повышения тактовой частоты или после снижения тактовой частоты.

Работа Flash памяти программ в режиме программирования

В режиме программирования Flash память программ не может выдавать инструкции и данные процессору, поэтому перевод памяти в режим программирования (установка бита CON = 1) возможен только программой, исполняемой из внешней памяти или ОЗУ. Перед переводом памяти в режим программирования необходимо в регистр EERPOM_KEY записать кобинацию 0x8AAA5551.

В режиме программирования возможны следующие операции как с основной (бит IFREN = 0, регистр EEPROM_CON), так и с информационной (бит IFREN = 1) памятью:

- стирание всей памяти
- стирание страницы памяти размером 4 Кбайт
- запись 32-х битного слова в память
- чтение 32-х битного слова из памяти

Bank 31 1К x 128 16K x 8	0x0001_FFFC … 0x0001_F00C	0x0001_FFF8 … 0x0001_F008	0x0001_FFF4 … 0x0001_F004	0x0001_FFF0 … 0x0001_F000
	…	…	…	…
Bank 1 1К x 128 16K x 8	0x0000_1FFC … 0x0000_100C	0x0000_1FF8 … 0x0000_1008	0x0000_1FF4 … 0x0000_1004	0x0000_1FF0 … 0x0000_1000
Bank 0 1К x 128 16K x 8	0x0000_0FFC … 0x0000_001C 0x0000_000C	0x0000_0FF8 … 0x0000_0018 0x0000_0008	0x0000_0FF4 … 0x0000_0014 0x0000_0004	0x0000_0FF0 … 0x0000_0010 0x0000_0000
	Sector_A 1K x 32 4K x 8	Sector_A 1K x 32 4K x 8	Sector_A 1K x 32 4K x 8	Sector_A 1K x 32 4K x 8

Основная память

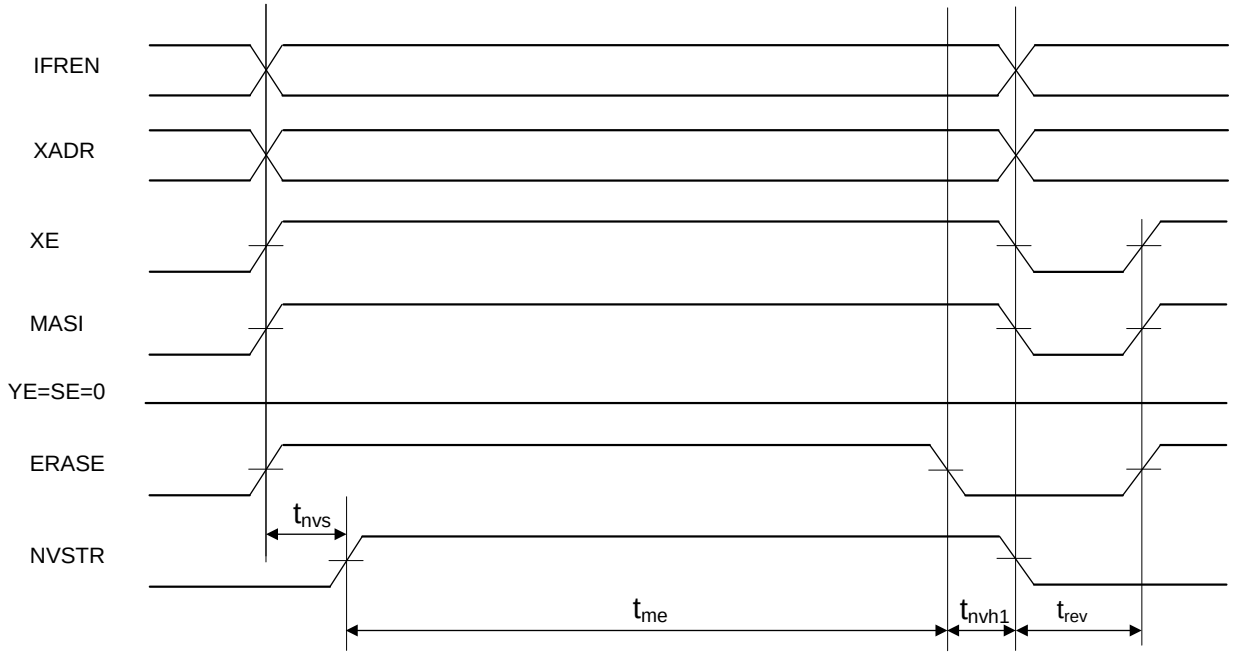
Bank 0 1К x 128 16K x 8	0x0000_0FFC … 0x0000_001C 0x0000_000C	0x0000_0FF8 … 0x0000_0018 0x0000_0008	0x0000_0FF4 … 0x0000_0014 0x0000_0004	0x0000_0FF0 … 0x0000_0010 0x0000_0000
	Sector_A 1K x 32 4K x 8	Sector_A 1K x 32 4K x 8	Sector_A 1K x 32 4K x 8	Sector_A 1K x 32 4K x 8

Информационная память

Структура памяти FLASH

Стирание всей памяти

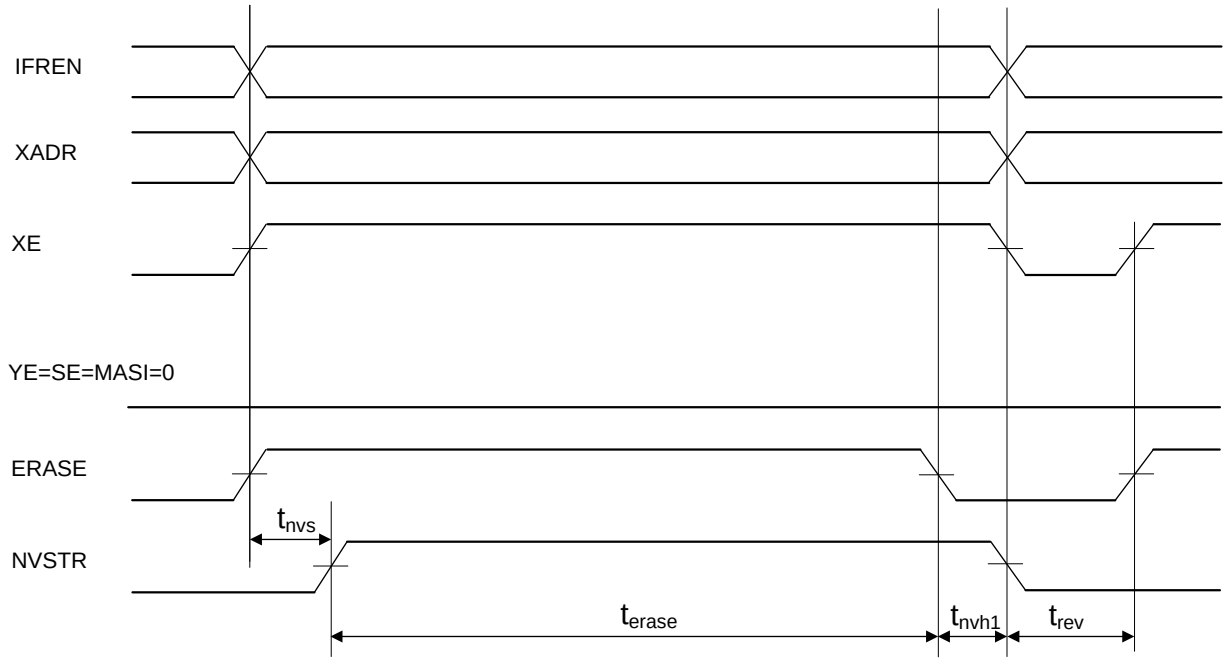
Стирание памяти возможно только в режиме программирования. Для стирания всей памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить биты XE, MAS1 и ERASE в единицу, и спустя время tnvs = 5 мкс установить бит NVSTR в единицу. Полное стирание памяти длиться время tme = 40 мс. Спустя это время необходимо очистить бит ERASE, и спустя время tnvh1 = 100 мкс очистить биты XE, MAS1 и NVSTR. Последующие операции с память можно выполнять спустя время trcv = 1 мкс. Временная диаграмма стирания памяти представлена на Временная диаграмма стирания памяти. При стирании всей информационной памяти также стирается и основная память.



Временная диаграмма стирания памяти

Стирание страницы памяти размером 4 Кбайта

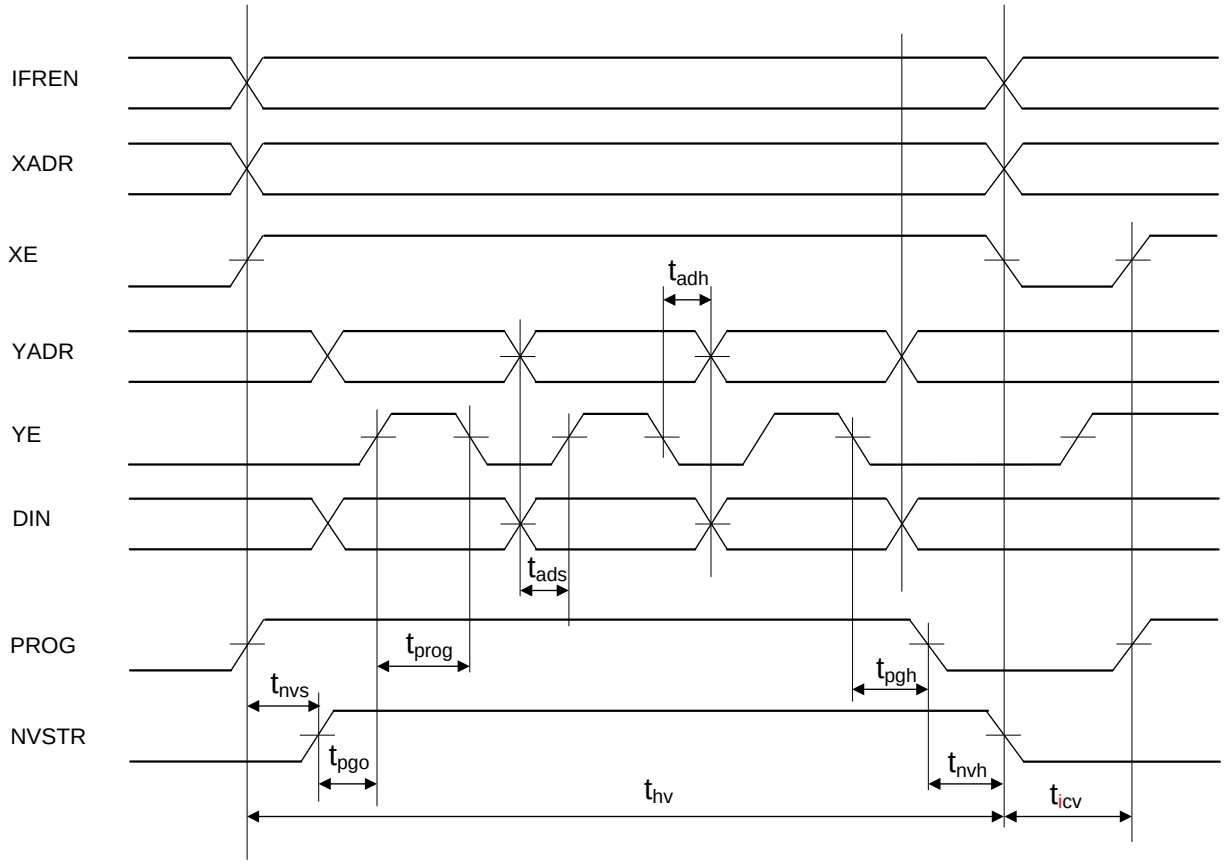
Стирание страницы памяти возможно только в режиме программирования. Для стирания страницы памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес стираемой страницы в регистре EEPROM_ADR и установить биты XE и ERASE в единицу, и спустя время tnvs = 5 мкс установить бит NVSTR в единицу. Стирание страницы памяти длиться время terase = 40 мс. Спустя это время необходимо очистить бит ERASE, и спустя время tnvh = 5 мкс очистить биты XE и NVSTR. Последующие операции с память можно выполнять спустя время trcv = 1 мкс. Временная диаграмма стирания страницы памяти представлена на Временная диаграмма стирания страницы памяти.



Временная диаграмма стирания страницы памяти

Запись 32-х битного слова в память

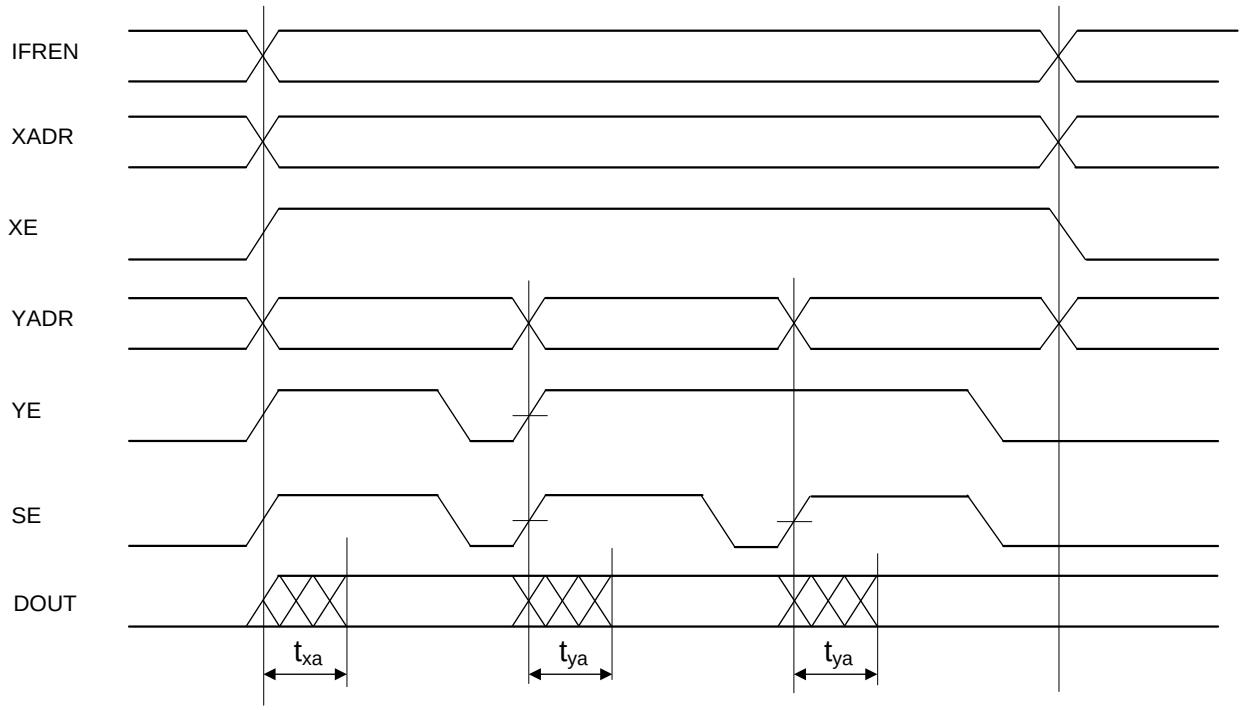
Запись в память возможна только в режиме программирования. Для записи в память надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес, по которому производится запись в регистре EEPROM_ADR, в регистр EEPROM_DI записать записываемое в память слово, установить биты XE и PROG в единицу, и спустя время tnvs = 5 мкс установить бит NVSTR в единицу. Спустя время tpgs = 10 мкс установить бит YE в единицу. Запись в память длиться время tprog = 40 мкс. Спустя это время необходимо очистить бит YE, и спустя время tadh = 20 нс установить новый адрес и значение для записи в другую ячейку памяти. И спустя tadh = 20 нс установить YE в единицу и записать следующую слово. Если запись больше не требуется, то спустя время tpgh = 20 нс после очистки бита YE необходимо очистить бит PROG и спустя время tnvh = 5 мкс очистить биты XE и NVSTR. Последующие операции с память можно выполнять спустя время trcv = 1 мкс. Временная диаграмма записи памяти представлена на Временная диаграмма записи памяти.



Временная диаграмма записи памяти

Чтение 32-х битного слова из памяти

В обычном режиме работы для чтения доступна только основная память. Для этого необходимо просто считать требуемый адрес памяти. В режиме программирования для чтения доступна и основная и информационная память. Для чтения из памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес, из которого необходимо считать данные в регистре EEPROM_ADR, установить биты XE, YE и SE в единицу, и спустя время txa = 30 нс из регистра EEPROM_DO можно считать данные. Если необходимо считать следующее слово, то в регистр EEPROM_ADR необходимо записать новый адрес и спустя время txa = 30 нс из регистра EEPROM_DO можно считать следующие данные. Если чтение больше не требуется, то можно очистить все биты управления. Временная диаграмма чтения памяти представлена на Временная диаграмма чтения памяти.



Временная диаграмма чтения памяти

Flash память программ поддерживает до 20 000 тысяч циклов перезаписи. Нельзя повторять циклы стирания – записи и стирания – стирания одной ячейки памяти с периодом менее 4 мс.

Описание регистров управления контроллера Flash памяти программ

Базовый Адрес	Название	Описание
0x4001_8000	EEPROM_CNTRL	Регистры контроллера Flash памяти программ
Смещение		
0x00	EEPROM_CMD	Регистр управления EEPROM памяти
0x04	EEPROM_ADR	Регистр адреса
0x08	EEPROM_DI	Регистр данных на запись
0x0C	EEPROM_DO	Регистр считанных данных
0x10	EEPROM_KEY	Регистр ключа

EEPROM_CMD

Номер	9	8	7	6	5…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	100	0	0	0
	IFREN	SE	YE	XE	Delay[2:0]	RD	WR	CON

Номер	31			14	13	12	11	10
Доступ	U			U	R/W	R/W	R/W	R/W
Сброс	0			0	0	0	0	0
					NVSTR	PROG	MAS1	ERASE

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..14	-	Зарезервировано
13	NVSTR	Операции записи или стирания 0 – при чтении 1- при записи или стирании
12	PROG	Записать данные по ADR[16:2] из регистра EERPOM_DI 0 – нет записи 1 – есть запись
11	MAS1	Стереть весь блок, при ERASE =1 0 – нет стирания 1 – стирание
10	ERASE	Стереть строку с адресом ADR[16:9], ADR[8:0] значения не имеет 0 – нет стирания 1 – стирание
9	IFREN	Работа с блоком информации 0 – основная память 1 – информационный блок
8	SE	Усилитель считывания 0 – не включен 1 – включен
7	YE	Выдача адреса ADR[8:2] 0 – не разрешено 1 – разрешено
6	XE	Выдача адреса ADR[16:9] 0 – не разрешено 1 - разрешено
5…3	Delay[2:0]	Задержка памяти программ при чтении в циклах (в рабочем режиме) 000 – 0 цикл 001 – 1 цикл … 111 – 7 циклов
2	RD	Чтение из память EERPOM (в режиме программирования)

		0 – нет чтения 1 – есть чтение
1	WR	Запись в память EERPOM (в режиме программирования) 0 – нет записи 1 – есть запись
0	CON	Переключение контроллера памяти EEPROM на регистровое управление, не может производится при исполнении программы из области EERPOM 0 – управление EERPOM от ядра, рабочий режим 1 – управление от регистров, режим программирования

EEPROM_ADR

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
		ADR [31:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	ADR[31:0]	Адрес обращения в память ADR[1:0] – не имеет значения, минимально адресуемая ячейка 32 бита

EEPROM_DI

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
		DATA [31:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	DATA[31:0]	Данные для записи в EERPOM

EEPROM_DO

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
		DATA [31:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	DATA[31:0]	Данные, считанные из EERPOM

EEPROM_KEY

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
	KEY [31:0]	

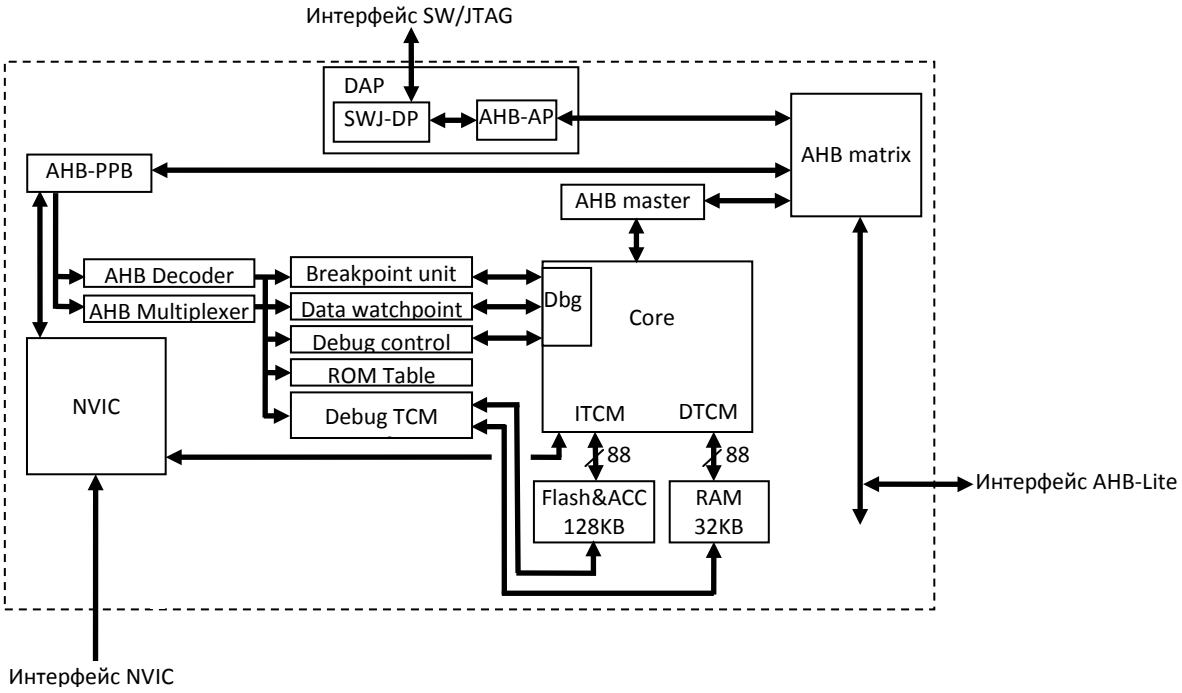
№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	KEY[31:0]	Ключ для разрешения доступа к Flash памяти через регистровый доступ. Перед переводом памяти в режим программирования необходимо в регистр EERPOM_KEY записать комбинацию 0x8AAA5551.

Процессорное ядро

Описание процессора и периферии ядра:

- Процессорное ядро, с минимизированным количеством вентилей, обладающее следующими характеристиками:
  - Содержит в своём составе 3-х уровневый конвейер.
  - Набор инструкций архитектуры ARM v6-M, включающий 32 битные Thumb-2 инструкции, такие как BL, MRS, MSR, ISB, DSB и DMB.
  - Возможность запуска операционной системы и доступные для этого режима работы SVC инструкции, групповой регистр указателя стека и интегрированный системный таймер.
  - Системная модель исключительных ситуаций.
  - Режимы Handler и Thread.
  - Два указателя стека.
  - Возможность работы только в режиме Thumb.
  - Отсутствие аппаратной поддержки не выровненного доступа.
  - Содержит 13x32 разрядных регистра общего назначения, link регистр (LR), счётчик команд (PC), программный регистр статуса xPSR, и два групповых регистра указателя стека (SP).
- Контроллер прерываний NVIC. Контроллер интегрирован в процессор для уменьшения задержек в процессе прерываний.
  - Поддержка до 32 внешних прерываний.
  - Два бита приоритета, обеспечивающие четырёхуровневый приоритет прерываний.
  - Состояние процессора автоматически сохраняется при входе в прерывание и восстанавливается при выходе, что не вызывает потерь на выполнение инструкций.
- Интерфейс памяти ITCM, DTCM, а также внешний интерфейс AHB-Lite.
  TCM интерфейс не поддерживает тактов ожидания, поэтому при тактовой частоте ядра выше 25 МГц, акселератор Flash памяти выключает тактовую частоту ядра на необходимое количество тактов.
- Полный набор отладочных модулей:
  - Полный доступ в режиме останова ко всей памяти и регистрам.
  - Отладочный порт DAP.
  - Модуль точек останова BPU.
  - Модуль наблюдения данных DW.
- 32-х разрядный аппаратный умножитель.

Структурная схема процессора



Периферийными блоками ядра является:

- контроллер прерываний NVIC

Реализует высокоскоростную обработку прерываний.

- Bus master

Обеспечивает два интерфейса. Один связывает внутренние Private Peripheral Bus (PPB) сигналы с шиной AHB PPB. Второй интерфейс связывает сигналы внешней шины с AHB портом.

- AHB Private Peripheral Bus (AHB-PPB).

Обеспечивает доступ к контроллеру NVIC и компонентам модулей отладки.

-AHB decoder

Дешифрирует адреса AHB шины для выработки сигналов выбора для периферии системы отладки.

- AHB multiplexer

Объединяет все ответы ведомых для отладочных блоков.

-AHB matrix

Выполняет функцию арбитража между процессором и отладочной системой при доступе к внутренней PPB и внешнему интерфейсу AHB-Lite.

- DAP

Процессор содержит AHB-Access Port (AHB-AP). AHB-AP преобразует выходы от внешних DP компонентов в интерфейс AHB-Lite. AHB-AP master имеет наивысший приоритет в AHB matrix.

Serial-Wire JTAG Debug Port (SWJ-DP) это комбинация JTAG порта и Serial Wire порта, а также механизма, позволяющего переключаться между Serial Wire и JTAG.

- Debug TCM интерфейс

Обеспечивает отладочный интерфейс для доступа к ITCM или DTCM. Только один TCM может быть доступен в любой момент времени.

- Breakpoint Unit

Содержит в своём составе компаратор 4-х адресов инструкций. Можно сконфигурировать каждый компаратор адреса инструкции для выполнения останова

программы с использованием аппаратной точки останова. Каждый компаратор может сравнивать адрес выбираемой инструкции с установленным адресом. Если адрес совпал, то BPU обеспечивает останов процессора в момент выполнения инструкции, вызвавшей совпадение. Точки останова поддерживаются только в области кода карты памяти.

-Data Watchpoint unit

Содержит в своём составе два компаратора адреса. Можно сконфигурировать компараторы для сравнения адреса инструкции или адреса данных. Поддерживается также маскирование компараторов.

Watchpoint частично точно. Это означает, что останов ядра происходит после выполнения следующей инструкции, после той, адрес которой вызвал совпадение компаратора.

-Debug control

Обеспечивает доступ к управляющим регистрам отладки через PPB для останова и пуска процессора. Помимо этого обеспечивается доступ к регистрам процессора, когда он остановлен.

- ROM table

Разрешает стандартным отладочным средствам распознать процессор и доступную периферию отладки, а также определить адреса, необходимые для доступа к этой периферии.

Программная модель

Процессор обеспечивает облегчённую версию Thumb-2, это все инструкции определённые в архитектуре ARM v6-M. Процессор не поддерживает выполнение ARM инструкций.

Процессор не поддерживает различий между режимами User и Privileged. Процессор всегда в режиме Privileged.

Процессор может функционировать в режимах:

- *Thread* режим

Используется для исполнения приложений, процессор находится в этом режиме сразу после сброса

- *Handler* режим

Используется для обработки исключений. После обработки процессор переходит в Thread режим.

Процессор может функционировать в одном из состояний:

-*Thumb state*

Это нормальное исполнение Thumb и Thumb-2 инструкций с 16-битными и 32-битными выровненными по полуслову данными.

- *Debug state*

Это состояние, при котором ядро остановлено.

Стек

По окончанию сброса весь код использует main стек. Обработчик прерываний, такой как SVCall, может переключить стек, который отображался в Thread режиме, из main в process, модификацией значения EXC_RETURN при выходе. Все прерывания продолжают использовать main стек. Указатель стека, R13, совмещённый регистр

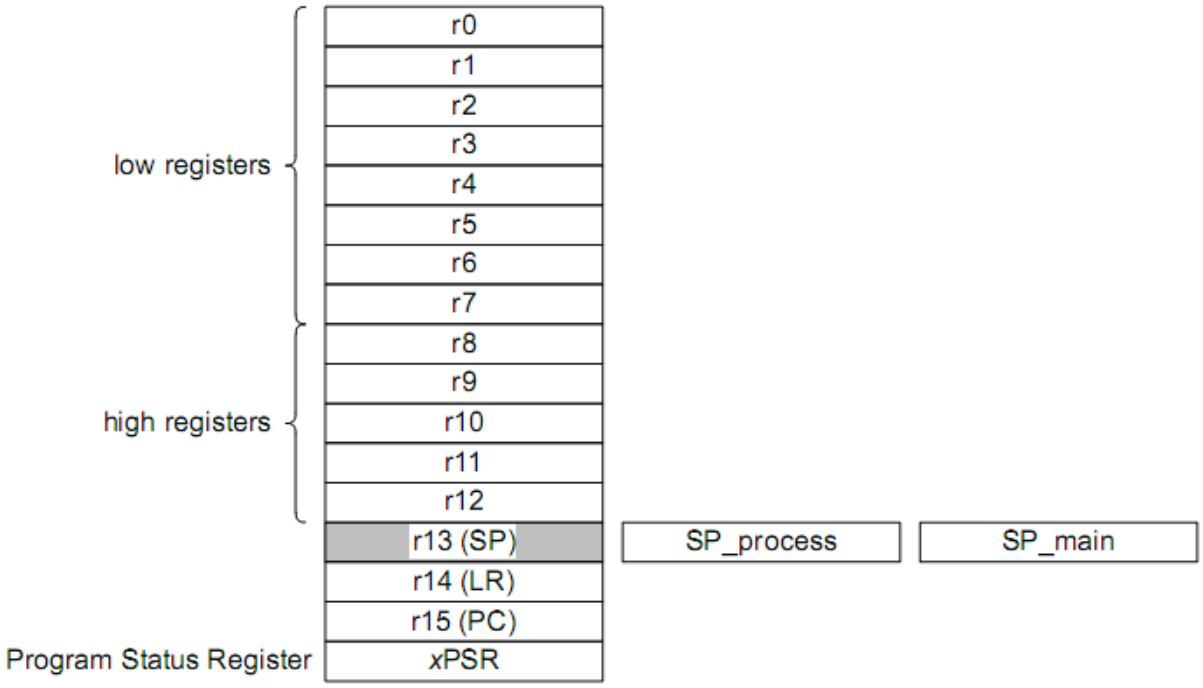
переключается между main и process стеком. Только один стек, process или main, виден посредством регистра R13 в данный момент времени.

Также возможно переключение между стеками main и process в Thread режиме записью в Special-Purpose Control регистр инструкцией MSR.

Регистры ядра

Процессор содержит следующие 32-х разрядные регистры:

- 13 регистров общего назначения, R0-R12
- Указатель стека (SP, R13) и объединенные регистры, SP_process и SP_main
- Link регистр (LR, R14)
- Счётчик команд (PC, R15)
- Программный регистр состояния, xPSR



Регистры общего назначения R0-R12

Low registers или R0-R7 доступны для все инструкций, которые определены для работы с регистрами общего назначения.
High registers или R8-R12 не доступны для 16 разрядных инструкций.

Указать стека SP R13

Регистр R13 используется как указатель стека. Запись в биты [1:0] этого регистра игнорируется, так как он автоматически выровнен по границе слова (четырёх байт). Биты SP[1:0] могут быть очищены инструкцией SBZP. В режиме Handler всегда используется SP_main, а в режиме Thread может быть использован либо SP_main, либо SP_process.

Регистр связи LR R14

Регистр R14 это регистр связи для подпрограмм. LR содержит адрес возврата для PC после выполнения инструкций перехода. Регистр используется для сохранения информации об адресе возврата при уходе на обработку прерываний, вызовах функций и обработке исключений. Во всех остальных случаях регистр может быть использован как регистр общего назначения.

Счетчик команд PC R15

Program Counter это регистр R15. Он содержит адрес текущей инструкции. Бит 0 всегда 0, так как все инструкции выровнены по границе полуслов. При сбросе процессор считывает в этот регистр вектор сброса, который расположен по адресу 0x00000004.

Программный регистр состояния PSR

Регистр Program Status Register (PSR) объединяет:

- Application Program Status Register (APSR)
- Interrupt Program Status Register (IPSR)
- Execution Program Status Register (EPSR)

Эти регистры разделяют различные битовые поля в 32-х разрядном PSR. Описание регистров приведено ниже. Доступ к этим регистрам может быть как индивидуальный, так и комбинированный к двум или всем трем разом, используя имена регистров как аргументы инструкций MSR или MRS. Например:

- читать все регистры, используя PSR с MRS инструкцией
- записать только в APSR используя APSR с MSR инструкцией

Комбинация PSR и их атрибуты

Регистр	Тип	Комбинация
XPSR	RW (1),(2)	APSR, EPSR и IPSR
IEPSR	RO	EPSR и IPSR
IAPSR	RW(1)	APSR и IPSR
EAPSR	RW(2)	APSR и EPSR

1. игнорируется запись в IPSR биты
2. при чтении EPSR битов читаются нули, и запись в них игнорируется

Подробнее в описании инструкции «MRS» и «MSR»

Регистр APSR содержит текущие флаги состояния выполнения предыдущей инструкции.

APSR

Номер	31	30	29	28	27	26		0
Доступ								
Сброс								
	N	Z	C	V	-	-	-	-

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31	N	**Negative** 0 – результат операции положительный, нулевой, больше чем, равен. 1 – результат операции отрицательный или меньше чем

30	Z	**Zero:** 0 – результат операции не нулевой 1 – результат операции нулевой
29	C	**Carry:** 0 – при суммировании не было переноса, при вычитании не было заема 1 – при суммировании был перенос, при вычитании был заем
28	V	**Overflow:** 0 – в результате операции не было переполнения 1 – в результате операции было переполнение
27…0	-	Зарезервировано

Регистр IPSR содержит номер типа исключения для текущего обработчика прерывания.

IPSR

Номер	31					6	5		0
Доступ									
Сброс									
	-				-	ISR_NUMBER			

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…6	-	Зарезервировано
5…0	ISR_NUMBER	**Номер текущего исключения** 0 – Thread режим 2 – NMI 3 – Hard Fault 11 – SVCall 14 – PendSV 15 – SysTick 16 – IRQ0 … 47 – IRQ31

Регистр EPSR содержит бит состояния Thumb инструкции.

EPSR

Номер	31					25	24	23		0
Доступ										
Сброс										
	-	-	-	-	T	-	-	-		

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..25	-	Зарезервировано

24	T	Этот бит устанавливается в соответствии с вектором сброса, когда процессор выходит из состояния reset. Выполнение инструкции очистки T-бита регистра EPSR приводит к возникновению аппаратной ошибки Hard Fault. Это позволяет быть уверенным, что переключение в ARM состояние, не приведет к непредсказуемым последствиям.
23..0	-	Зарезервировано

Пока процессор не в режиме отладки, попытка читать EPSR, используя MSR инструкцию, всегда возвращает ноль, а попытка записать EPSR, используя MSR напрямую, игнорируется.

Сохранение xPSR бит

При входе в прерывание процессор сохраняет сгруппированные данные из трёх регистров в стек. Бит 9, помещённого в стек, xPSR содержит статус активного SP, когда начался процесс обработки прерывания.

Priority Mask регистр

Регистр PRIMASK используется для повышения приоритета.

PRIMASK

Номер	31							1	0
Доступ									
Сброс									
	-	-	-	-	-	-	-		PRIMASK

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…1	-	Зарезервировано
0	PRIMASK	0 – не влияет 1 – увеличивает приоритет исполнения до 0

Для доступа к регистру применяются инструкции MSR и MRS, а также инструкция CPS для установки или очистки бита PRIMASK.

Контрольный регистр специального назначения

Регистр определяет текущий указатель стека.
CONTROL

Номер	31							2	1	0
Доступ										
Сброс										

-	-	-	-	-	-	Active Stack Pointer	-

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…8	-	Зарезервировано
1	Active Stack Pointer	0 – SP_main используется, как текущий указатель стека 1 – Для Thread режима, SP_process используется, как текущий указатель стека[a]
0	-	Зарезервировано

a. Попытка установить этот бит в режиме Handler игнорируется.

Типы данных

Процессор поддерживает следующие типы данных:

- 32-битное слово (words)
- 16-битное полуслово (half words)
- 8-битный байт (bytes)

Процессор может иметь доступ ко всем регионам памяти, включая code регион, со всеми типами данных. Для поддержки этого, система, включая память, должна поддерживать запись полуслов и байт без изменения соседних байт в слове. Процессор манипулирует всеми данными в little-endian режиме. Доступ в память инструкций и Private Peripheral Bus (PPB) всегда в little-endian режиме.

Система команд

В процессоре реализована версия системы команд Thumb. Поддерживаемые команды представлены в таблице 13-13.

В таблице используются следующие обозначения:

- в угловых скобках <> записываются альтернативные формы представления операндов;
- в фигурных скобках {} указываются необязательные операнды;
- информация в столбце "операнды" может быть неполной;
- второй операнд Op2 может быть либо регистром, либо константой;
- большинство команд могут содержать суффикс кода условного выполнения.

Более подробная информация представлена в детальном описании команд.

Таблица 13-13. Система команд процессора

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
ADC, ADCS	{Rd,} Rn, Op2	Сложение с переносом	N,Z,C,V	
ADD, ADDS	{Rd,} Rn, Op2	Сложение	N,Z,C,V	
ADR	Rd, label	Загрузка адреса, заданного относительно счетчика команд	-	
AND, ANDS	{Rd,} Rn, Op2	Логическое И	N,Z,C	
ASR, ASRS	Rd, Rn, Op2	Арифметический сдвиг вправо	N,Z,C	
B	label	Переход	-	
BIC, BICS	{Rd,} Rn, Op 2	Сброс битов по маске	N,Z,C	
BKPT	#imm8	Точка останова	-	
BL	label	Переход со связью	-	
BLX	Rm	Косвенный переход со связью	-	
BX	Rm	Косвенный переход	-	
CMN, CMNS	Rn, Op2	Сравнить с противоположным знаком	N,Z,C,V	
CMP, CMPS	Rn, Op2	Сравнить	N,Z,C,V	
CPSID	iflags	Изменить состояние процес-сора, запретить прерывания	-	
CPSIE	iflags	Изменить состояние процес-сора, разрешить прерывания	-	
CPY	Rd, Op2	Загрузка	N,Z,C	
DMB	-	Барьер синхронизации доступа к памяти данных	-	
DSB	-	Барьер синхронизации доступа к памяти данных	-	
EOR, EORS	{Rd,} Rn, Op2	Исключающее ИЛИ	N,Z,C	
ISB	-	Барьер синхронизации доступа к инструкциям	-	

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
LDM	Rn!, reglist	Загрузка множества регистров, инкремент после доступа	-	
LDMFD, LDMIA	Rn!, reglist	Загрузка множества регистров, инкремент после доступа	-	
LDR	Rt, [Rn, #offset]	Загрузка слова в регистр	-	
LDRB	Rt, [Rn, #offset]	Загрузка байта в регистр	-	
LDRH	Rt, [Rn, #offset]	Загрузка полуслова в регистр	-	
LDRSB	Rt, [Rn, #offset]	Загрузка в регистр байта со знаком	-	
LDRSH	Rt, [Rn, #offset]	Загрузка в регистр полуслова со знаком	-	
LSL, LSLS	Rd, Rm, <Rs|#n>	Логический сдвиг влево	N,Z,C	
LSR, LSRS	Rd, Rm, <Rs|#n>	Логический сдвиг вправо	N,Z,C	
MOV, MOVS	Rd, Op2	Загрузка	N,Z,C	
MRS	Rd, spec_reg	Считать специальный регистр в регистр общего назначения	-	
MSR	spec_reg, Rm	Записать регистр общего назначения в специальный регистр	N,Z,C,V	
MUL, MULS	{Rd,} Rn, Rm	Умножение, 32-разрядный результат	N,Z	
MVN, MVNS	Rd, Op2	Загрузка инверсного значения	N,Z,C	
NEG	{Rd,} Rm	Инвертирование	N,Z,C,V	
NOP	-	Нет операции	-	
ORR, ORRS	{Rd,} Rn, Op2	Логическое ИЛИ	N,Z,C	
POP	reglist	Извлечь регистры из стека	-	
PUSH	reglist	Занести регистры в стек	-	
REV	Rd, Rn	Изменить на обратный порядок байтов в слове	-	
REV16	Rd, Rn	Изменить на обратный порядок байтов в полусловах	-	
REVSH	Rd, Rn	Изменить на обратный порядок байт в младшем полуслове, произвести распространение знакового бита в старшее полуслово	-	
ROR, RORS	Rd, Rm, <Rs|#n>	Циклический сдвиг вправо	N,Z,C	
RSB, RSBS	{Rd,} Rn, Op2	Вычитание с противоположным порядком аргументов	N,Z,C,V	
SBC, SBCS	{Rd,} Rn, Op2	Вычитание с учетом переноса	N,Z,C,V	
SEV	-	Установить признак события	-	
STM	Rn!, reglist	Сохранение множества	-	

Мнемокод команды	Операнды	Краткое описание	Флаги	Страница
		регистров, инкремент после доступа		
STMEA	Rn!, reglist	Сохранение множества регистров, инкремент перед доступом	-	
STMIA	Rn!, reglist	Сохранение множества регистров, инкремент после доступа	-	
STR	Rt, [Rn, #offset]	Сохранение регистра	-	
STRB	Rt, [Rn, #offset]	Сохранение регистра, байт	-	
STRH	Rt, [Rn, #offset]	Сохранение регистра, полуслово	-	
SUB, SUBS	{Rd,} Rn, Op2	Вычитание	N,Z,C,V	
SVC	#imm	Вызов супервизора	-	
SXTB	{Rd,}Rm{,ROR#n}	Преобразовать байт со знаком в слово	-	
SXTH	{Rd,}Rm{,ROR#n}	Преобразовать полуслово со знаком в слово	-	
TST	Rn, Op2	Проверка значения битов по маске	N,Z,C	
UXTB	{Rd,}Rm{,ROR#n}	Преобразовать байт без знака в слово	-	
UXTH	{Rd,}Rm{,ROR#n}	Преобразовать полуслово без знака в слово	-	
YIELD	-	Инструкция hint для аппаратного обеспечения при многопотоковых задачах	-	

Встроенные функции

Стандарт ANSI языка С не обеспечивает непосредственного доступа к некоторым инструкциям процессора. В разделе описаны встроенные (intrinsic) функции, которые указывают компилятору на необходимость генерации соответствующих инструкций. В случае если используемый компилятор не поддерживает ту или иную встроенную функцию, рекомендуется включить в текст программы ассемблерную вставку с необходимой инструкцией.

В CMSIS предусмотрены следующие встроенные функции, расширяющие возможности стандарта ANSI C.

Таблица13-14. Встроенные функции CMSIS, позволяющие генерировать некоторые инструкции процессора

Мнемокод команды процессора	Описание встроенной функции
CPSIE I	void __enable_irq(void)
CPSID I	void __disable_irq(void)
CPSIE F	void __enable_fault_irq(void)
CPSID F	void __disable_fault_irq(void)
ISB	void __ISB(void)
DSB	void __DSB(void)
DMB	void __DMB(void)
REV	uint32_t __REV(uint32_t int value)
REV16	uint32_t __REV16(uint32_t int value)
REVSH	uint32_t __REVSH(uint32_t int value)
SEV	void __SEV(void)

Кроме того, CMSIS также обеспечивает возможность чтения и записи специальных регистров процессора, доступных с помощью команд MRS и MSR.

Таблица 13-15. Встроенные функции CMSIS для доступа к специальным регистрам процессора

Наименование специального регистра	Режим доступа	Описание встроенной функции
PRIMASK	Чтение	uint32_t __get_PRIMASK (void)
	Запись	void __set_PRIMASK (uint32_t value)
CONTROL	Чтение	uint32_t __get_CONTROL (void)
	Запись	void __set_CONTROL (uint32_t value)
MSP	Чтение	uint32_t __get_MSP (void)
	Запись	void __set_MSP (uint32_t TopOfMainStack)
PSP	Чтение	uint32_t __get_PSP (void)
	Запись	void __set_PSP (uint32_t TopOfProcStack)

Описание инструкций

В разделе представлена подробная информация об инструкциях процессора:

- операнды;
- ограничения на использование счетчика команд PC и указателя стека SP;
- формат второго операнда;
- операции сдвига;
- выравнивание адресов;
- выражения с участием счетчика команд;
- условное исполнение.

Операнды

В качестве операнда инструкции может выступать регистр, константа, либо другой параметр, специфичный для конкретной команды. Процессор применяет инструкцию к операндам и, как правило, сохраняет результат в регистре-получателе. В случае если формат команды предусматривает спецификацию регистра-получателя, он, как правило, указывается непосредственно перед операндами.

Операнды в некоторых инструкциях допускают гибкий формат представления, то есть могут быть как регистром, так и константой. Подробнее см. “Формат второго операнда”.

Ограничения на использование PC и SP

Многие инструкции не позволяют использовать регистры счетчика команд (PC) и указателя стека (SP) в качестве регистра-получателя. Подробная информация содержится в описании конкретных инструкций.

Бит [0] адреса, загружаемого в PC с помощью одной из команд BX, BLX, LDM, LDR или POP должен быть равен 1, так как этот бит указывает на требуемый набор команд, а процессор поддерживает только инструкции из набора Thumb.

Формат второго операнда

Большинство команд обработки данных поддерживают гибкий формат задания второго операнда. Далее в описании синтаксиса инструкций процессора такой операнд будет обозначаться как Operand2. При этом в качестве операнда может выступать:

- константа;
- регистр с необязательным параметром сдвига.

Константа

Данный тип второго операнда задается в формате:

#constant

где constant может быть:

- любой константой, которая может быть получена путем сдвига восьмиразрядного числа влево на любое количество разрядов в пределах 32-разрядного слова;
- любая константа в виде 0x00XY00XY;
- любая константа в виде 0xXY00XY00;
- любая константа в виде 0xXYXYXYXY.

во всех вышеописанных случаях X и Y представляют шестнадцатеричные цифры.

Кроме того в небольшом количестве инструкций constant может принимать более широкий диапазон значений. Подробности изложены в описании соответствующих инструкций.

При использовании константного операнда Operand2 в командах MOVS, MVNS, ANDS, ORRS, EORS и TST в случае, если константа больше 255 и может быть получена путем сдвига восьмиразрядного числа значение бита [31] константы влияет на значение флага переноса. Для всех остальных значений Operand2 изменения флага переноса не происходит.

Замена инструкций

В случае если пользователь указывает константу, не удовлетворяющую требованиям, ассемблер может сгенерировать код с использованием другой инструкции, обеспечивающей необходимую функциональность.

Например, команда CMP Rd, #0xFFFFFFFE может быть преобразована в эквивалентную команду CMN Rd, #0x2.

Регистр с необязательным параметром сдвига

В данном случае операнд Operand2 задается в форме:

Rm {, shift}

где:
Rm - регистр, содержащий данные для второго операнда инструкции;
shift - необязательный параметр, определяющий сдвиг данных регистра Rm. Он может принимать одно из следующих значений:

ASR #n - арифметический сдвиг вправо на n бит, 1 <= n <= 32.
LSL #n - логический сдвиг влево на n бит, 1 <= n <= 31.
LSR #n - логический сдвиг вправо на n бит, 1 <= n <= 32.
ROR #n - циклический сдвиг вправо на n бит, 1 <= n <= 31.

Случай, если сдвиг не указан, эквивалентен заданию сдвига LSL #0. При этом в качестве операнда используется непосредственно значение регистра Rm без каких-либо дополнительных преобразований.

При указании параметра сдвига в качестве операнда используется преобразованное соответствующим образом 32-разрядное значение регистра Rm, однако содержимое самого регистра Rm не меняется.

Использование операнда со сдвигом в некоторых инструкциях влияет на значение флага переноса. Более подробно действие операций сдвига и их влияние на флаг переноса рассмотрено в разделе "Операции сдвига".

Операции сдвига

Операции сдвига переносят значение битов содержимого регистра влево или вправо на заданное количество позиций - длина сдвига. Сдвиг может выполняться:

- непосредственно с помощью инструкций ASR, LSR, LSL и ROR, при этом результат сдвига заносится в регистр-получатель;
- во время вычисления значения второго операнда Operand2 команд, при этом результат сдвига используется как один из операндов инструкции.

Допустимая длина сдвига зависит от типа сдвига и инструкции, в которой он был применен. В случае, если этот параметр равен 0, фактически сдвиг не производится. Операции сдвига регистра влияют на значение флага переноса, за исключением случая, когда длина сдвига равна 0. Различные варианты сдвига и их влияние на флаг переноса описаны в следующем подразделе (Rm - сдвигаемый регистр, n - длина сдвига).

ASR

Арифметический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. Бит [31] исходного значения регистра записывается в n крайних слева бит результата. См. иллюстрацию на рис. 13-4.

Операцию ASR # n можно использовать для деления значения регистра Rm на 2^n, с округлением результата в меньшую сторону (в направлении минус бесконечности).

При использовании инструкции ASRS, а также в случае, если сдвиг ASR #n используется при вычислении второго операнда команд MOVS, MVNS, ANDS, ORRS, EORS или TST флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита [n-1] регистра Rm.

В случае, если n >= 32, все биты результата устанавливаются в значение бита [31] регистра Rm. Если при этом операция влияет на флаг переноса, то значение этого флага устанавливается равным значению бита [31] регистра Rm.

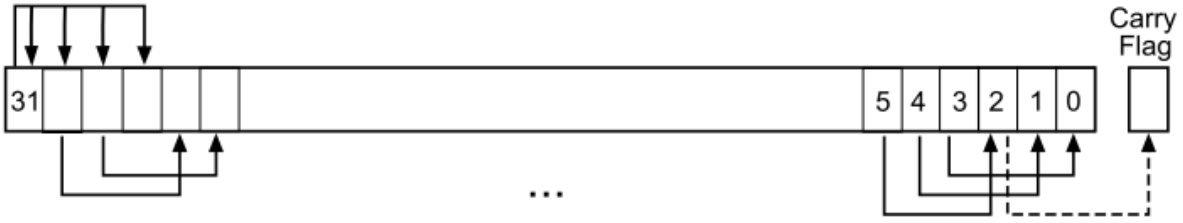


Рис. 13-14. Инструкция ASR #3

LSR

Логический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. При этом в n крайних слева бит результата записывается 0. См. иллюстрацию на рис. 13-5.

Операцию LSR # n можно использовать для деления значения регистра Rm на 2^n, в случае, если значение интерпретируется как целое число без знака.

При использовании инструкции LSRS, а также в случае, если сдвиг LSR #n используется при вычислении второго операнда команд MOVS, MVNS, ANDS, ORRS, EORS или

TST флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита [n-1] регистра Rm.

В случае, если n >= 32, все биты результата устанавливаются в 0. Если n >= 33 и операция влияет на флаг переноса, то значение этого флага устанавливается равным 0.

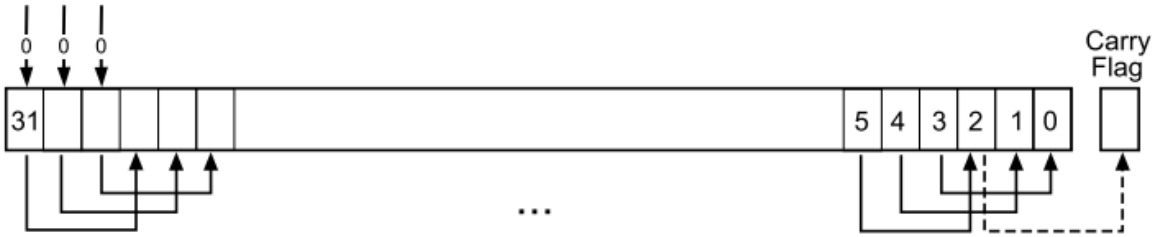


Рис. 13-15 Инструкция LSR # 3

LSL

Логический сдвиг влево на n бит переносит крайние справа 32-n бит регистра Rm влево на n позиций, то есть на место крайних слева 32-n. При этом в n крайних слева бит результата записывается 0. См. иллюстрацию на рис. 13-6.

Операцию LSL # n можно использовать для умножения значения регистра Rm на 2^n, в случае, если значение интерпретируется как целое число без знака, либо целое число со знаком, записанное в дополнительном коде. Переполнение при выполнении умножения не диагностируется.

При использовании инструкции LSLS, а также в случае, если сдвиг LSL #n используется при вычислении второго операнда команд MOVS, MVNS, ANDS, ORRS, EORS или TST флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита [32-n] регистра Rm. Инструкция LSL #0 не влияет на значение флага переноса.

В случае, если n >= 32, все биты результата устанавливаются в 0. Если n >= 33 и операция влияет на флаг переноса, то значение этого флага устанавливается равным 0.

Рис. 13-16. Инструкция LSL # 3

ROR

Циклический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. При этом n крайних справа разрядов регистра переносятся в крайние n слева разрядов результата. См. иллюстрацию на рис. 13-7.

При использовании инструкции RORS, а также в случае, если сдвиг ROR #n используется при вычислении второго операнда команд MOVS, MVNS, ANDS, ORRS, EORS

или TST флаг переноса принимает значение последнего сдвинутого бита, то есть бита [n-1] регистра Rm.

В случае, если n = 32, результат совпадает с исходным значением регистра. Если n = 32 и операция влияет на флаг переноса, то значение этого флага устанавливается равным биту [31] регистра Rm.

Операция циклического сдвига ROR с параметром, большим 32, эквивалентна циклическому сдвигу с параметром n-32.

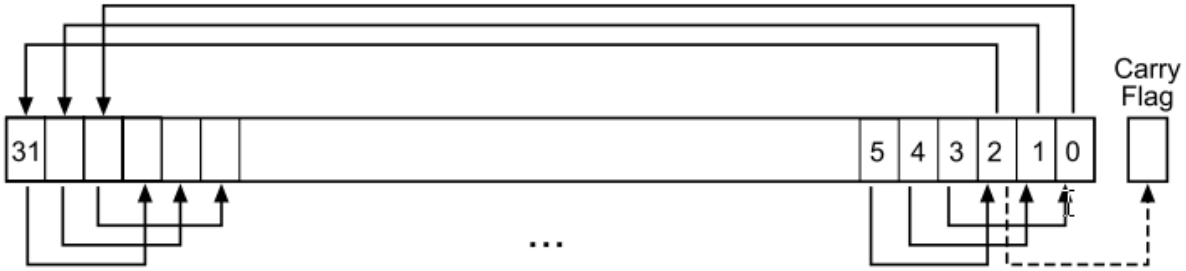


Рис. 13-17. Инструкция ROR # 3

Выравнивание адресов

Под доступом по выровненным адресам понимаются операции, в которых чтение и запись слов, двойных слов, и более длинных последовательностей слов осуществляется по адресам, выровненным по границе слова, а доступ к полусловам осуществляется по адресам, выровненным по границе полуслова. Чтение и запись байтов гарантированно является выровненным.

Процессор поддерживает доступ по невыровненным адресам только для следующих инструкций:

- LDR;
- LDRH;
- LDRSH;
- STR;
- STRH.

Все остальные инструкции при попытке доступа по невыровненному адресу генерируют исключение (Hard fault). Более подробно данный вопрос рассмотрен в разделе "Обработка отказов".

Невыровненный доступ к данным, как правило, осуществляется медленнее, чем выровненный. Кроме того, некоторые области адресного пространства могут не поддерживать доступ по невыровненному адресу. В связи с этим ARM рекомендует программистам обеспечивать необходимое выравнивание данных. Для того, чтобы избежать ситуаций, в которых невыровненный доступ осуществляется непреднамеренно, следует установить в 1 бит UNALIGN_TRP регистра конфигурации и управления CCR, что приведет к формированию процессором исключительной ситуации в данной ситуации (см. "Регистр конфигурации и управления").

Адресация относительно счетчика команд PC

В системе команд предусмотрена адресация команды или области данных в виде суммы значения счетчика команд PC плюс/минус численное смещение. Смещение

вычисляется ассемблером автоматически исходя из адреса метки и текущего адреса. В случае если смещение слишком велико, диагностируется ошибка.

- для инструкций B, BL текущий адрес определяется как адрес этой инструкции плюс 4 байта;
- для всех остальных инструкций текущий адрес определяется как адрес инструкции плюс 4 байта, при этом бит [1] результата должен быть установлен в 0 для обеспечения выравнивания адреса по границе слова.
- ассемблер может поддерживать расширенные варианты синтаксиса для адресации относительно PC, например "метка плюс/минус число" или выражения типа [PC, #number].

Условное исполнение

Большая часть команд обработки данных способна изменять значения флагов в регистре состояния прикладной программы (APSR) в зависимости от результата выполнения.

Некоторые команды влияют на все флаги, некоторые только на часть. В случае, если инструкция не меняет значение данного флага, сохраняется его старое значение. Более подробно влияние на флаги рассмотрено в описании конкретных инструкций.

Возможность исполнения или неисполнения инструкции, в зависимости от значения флагов, сформированных ранее, может быть достигнута либо за счет использования условных переходов, либо путем добавления суффикса условия исполнения к инструкции. В таблице 13-16 представлен список суффиксов, которые можно добавить к инструкции для того, чтобы сделать ее условной.

При наличии одного из указанных суффиксов процессор проверяет значение флагов на соответствие заданному условию. Если условие не выполняется, то инструкция:

- не исполняется;
- не записывает значение операции в регистр-получатель;
- не влияет на флаги;
- не генерирует исключений.

Процессорное ядро поддерживает только одну инструкцию условного перехода B<c> (Branch), где <c> один из суффиксов условного исполнения.

Ниже в разделе рассматриваются:

- флаги условий;
- суффиксы условного исполнения.

Флаги условий

Регистр состояния прикладной программы APSR содержит следующие флаги:

- N=1 в случае, если результат операции меньше нуля, 0 в противном случае.
- Z=1 в случае, если результат равен нулю, 0 в противном случае.
- C=1 в случае, если при выполнении операции возник перенос, 0 в противном случае.
- V=1 в случае, если при выполнении операции возникло переполнение, 0 в противном случае.

Перенос возникает в следующих случаях:

- результат сложения оказался больше или равен 2^32;
- результат вычитания больше или равен нулю;
- в результате работы внутренней логики процессора при операциях загрузки данных и логических операций.

Переполнение возникает в случае, если результат сложения, вычитания или сравнения больше или равен 2^31, либо меньше -2^31.

Большая часть инструкций меняют значение флагов только в случае, если у них указан суффикс S. Подробную информацию см. в описании конкретных команд.

Суффиксы условного исполнения

В мнемокодах команд, допускающих условное исполнение, предусмотрена возможность указания необязательного кода условия. В описании синтаксиса это обозначается как {cond}.

Если код условия указан, инструкция выполняется только при удовлетворении соответствующему условию флагов регистра APSR. Используемые коды представлены в таблице 13-16. Там же указаны соответствующие логические выражения для значений флагов.

Условные команды рекомендуется использовать для снижения количества ветвлений в программе.

Таблица 13-16 Суффиксы условного исполнения

Суффикс	Флаги	Значение
EQ	Z = 1	Равенство
NE	Z = 0	Неравенство
CS или HS	C = 1	Больше или равно, беззнаковое сравнение
CC или LO	C = 0	Меньше, беззнаковое сравнение
MI	N = 1	Меньше нуля
PL	N = 0	Больше или равно нулю
VS	V = 1	Переполнение
VC	V = 0	Нет переполнения
HI	C = 1 and Z = 0	Больше, беззнаковое сравнение
LS	C = 0 or Z = 1	Меньше или равно, беззнаковое сравнение
GE	N = V	Больше или равно, знаковое сравнение
LT	N != V	Меньше, знаковое сравнение
GT	Z = 0 and N = V	Больше, знаковое сравнение
LE	Z = 1 and N != V	Меньше или равно, знаковое сравнение
AL	1	Безусловное исполнение.

Команды доступа к памяти

Обобщенные данные о командах доступа к памяти представлены в таблице 13-17:

Таблица 13-17. Команды доступа к памяти

Мнемокод	Краткое описание	Страница
ADR	Загрузка адреса, заданного относительно счетчика команд	
LDM{mode}	Загрузка множества регистров	
LDR{type}	Загрузка регистра, непосредственно указанное смещение	
LDR{type}	Загрузка регистра, смещение в регистре	
LDR	Загрузка регистра по относительному адресу	
POP	Извлечение регистров из стека	
PUSH	Загрузка регистров в стек	
STM{mode}	Сохранение множества регистров	
STR{type}	Сохранение регистра, непосредственно указанное смещение	
STR{type}	Сохранение регистра, смещение в регистре	

ADR

Загрузка адреса, заданного относительно счетчика команд.

Синтаксис

ADR Rd, label

где:
Rd - регистр-получатель.
label - относительный адрес, см. "Адресация относительно счетчика команд".

Описание

Инструкция ADR вычисляет адрес доступа к памяти путем сложения текущего значения счетчика команд PC и непосредственно заданного смещения, после чего записывает результат в регистр-получатель.

Благодаря использованию относительно адресации код команды не зависит от ее размещения в физической памяти.

При формировании с помощью команды ADR адреса перехода для команд BX или BLX программисту необходимо убедиться, что бит [0] формируемого адреса установлен в 1.

Значения смещения относительно PC должны находиться в пределах 0...1020.

Ограничения

В качестве регистра Rd нельзя использовать указатель стека SP и счетчик команд PC.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
ADR R1, TextMessage     ; Загрузить адрес позиции, указанный
                        ; меткой TextMessage, в регистр R1
```

LDR и STR, непосредственно заданное смещение

Загрузка или сохранение регистра в режиме адресации со смещением, пре-индексированием или пост-индексированием.

Синтаксис

```
op{type} Rt, [Rn {, #offset}]      ; адресация со смещением
op{type} Rt, [Rn, #offset]!        ; пре-индексирование
op{type} Rt, [Rn], #offset         ; пост-индексирование
opD Rt, Rt2, [Rn {, #offset}]      ; адресация со смещением, двойное слово
opD Rt, Rt2, [Rn, #offset]!        ; пре-индексирование, двойное слово
opD Rt, Rt2, [Rn], #offset         ; пост-индексирование, двойное слово
```

где:
op - один из кодов операций:

- LDR загрузить регистр.
- STR сохранить регистр.

type - один из суффиксов размера данных:

- B - байт без знака, при загрузке старшие байты устанавливаются в нуль.
- SB - байт со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- H - беззнаковое полуслово, при загрузке старшие байты устанавливаются в нуль.
- SH - полуслово со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- без суффикса - 32-разрядное слово.

Rt - регистр, в который должна производиться загрузка или значение которого должно быть сохранено.
Rn - регистр, содержащий базовый адрес памяти.
offset - смещение относительно базового адреса Rn. В случае, если смещение не указано, оно подразумевается равным нулю.
Rt2 - дополнительный регистр, предназначенный для двухсловных операций чтения или записи.

Описание

LDR - загружает один или два регистра значением из памяти.
STR - сохраняет значение одного или двух регистров в память.

Инструкции с непосредственно заданным смещением могут функционировать в одном из следующих режимов адресации:

Адресация со смещением

Значение смещения добавляется к или вычитается из содержимого регистра Rn. Результат используется в качества адреса чтения или записи. Значение регистра Rn остается неизменным.

Синтаксис задания данного режима: [Rn, #offset].

Адресация с пре-индексированием

Значение смещения добавляется к или вычитается из содержимого регистра Rn. Результат используется в качества адреса чтения или записи, а также записывается обратно в регистр Rn.

Синтаксис задания данного режима: [Rn, #offset]! .

Адресация с пост-индексированием

Содержимое регистра Rn используется в качества адреса чтения или записи. Значение смещения добавляется к или вычитается из содержимого регистра Rn, после чего записывается обратно в регистр Rn.

Синтаксис задания данного режима: [Rn], #offset .

Загружаемое или сохраняемое значение может быть байтом, полусловом, словом или двойным словом. Байты и полуслова могут интерпретироваться как числа со знаком или без знака. См. “Выравнивание адресов”.

В таблице 13-18 показаны диапазоны значений смещения для различных форм адресации.

Table 13-18. Диапазон значений смещения

Тип инструкции	Смещение	Пре-индексирование	пост-индексирование
Слово, полуслово, байт	От 0 до 124	от 0 до 124	от 0 до 124
Двойное слово	Значения, кратные 4, в диапазоне от 0 до 1020		

Ограничения

Для команд загрузки регистров:

- использовать в качестве Rt регистры PC и SP можно только в командах загрузки слова;
- при загрузке двойных слов регистры Rt и Rt2 не должны совпадать
- в режимах адресации с пре- и пост-индексированием регистр Rn не должен совпадать с регистрами Rt или Rt2.

В случае, если в команде загрузки слова в качестве регистра Rt используется счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1;
- передача управления происходит по адресу, соответствующему значению бита [0] в 0;

Для команд сохранения регистров:

- использовать в качестве Rt регистры SP можно только в командах записи слова;
- в качестве регистров Rt и Rn нельзя использовать счетчик команд PC;
- в режимах адресации с пре- и пост-индексированием регистр Rn не должен совпадать с регистрами Rt или Rt2.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
LDR R8, [R10]              ; Загрузка регистра R8 из ячейки по адресу,
                           ; содержащемуся в R10.
STR R2, [R9,#const-struc]  ; const-struc - выражение с постоянным значением,
                           ; лежащим в диапазоне 0-124.
STRH R3, [R4], #4          ; Записать содержимое R3, интерпретируемое как
                           ; полуслово, по адресу, содержащемуся в R4, после чего
                           ; увеличить R4 на 4
```

LDR и STR, смещение задано в регистре

Загрузка или сохранение регистра в режиме адресации со смещением, заданным в регистре.

Синтаксис

op{type} Rt, [Rn, Rm {, LSL #n}]

где:
op - один из кодов операций:

- LDR загрузить регистр.
- STR сохранить регистр.

type - один из суффиксов размера данных:

- B - байт без знака, при загрузке старшие байты устанавливаются в нуль.
- SB - байт со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- H - беззнаковое полуслово, при загрузке старшие байты устанавливаются в нуль.
- SH - полуслово со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- без суффикса - 32-разрядное слово.

Rt - регистр, в который должна производиться загрузка или значение которого должно быть сохранено.
Rn - регистр, содержащий базовый адрес памяти.
Rm - регистр, содержащий смещение относительно базового адреса.
LSL #n - необязательный параметр сдвига, в диапазоне от 0 до 3.

Описание

LDR - загружает регистра значением из памяти.
STR - сохраняет значение регистра в памяти.

Адрес области памяти, в которую будет производиться обращение, вычисляется на основании значения базового адреса в регистре Rn и смещения. Смещение определяется значением регистра Rm и параметром сдвига влево значения этого регистра.

Считываемое или записываемое значение может иметь размер байта, полуслова или слова. При загрузке данных из памяти байты и полуслова могут интерпретироваться либо как числа со знаком, либо как беззнаковые. См. “Выравнивание адресов”.

Ограничения

Для данных команд:

- Rn не может быть счетчиком команд PC;
- Rm не может быть SP или PC;
- использовать в качестве Rt регистр SP можно только в командах чтения и записи слова;

- использовать в качестве Rt регистр PC можно только в командах чтения слова;

В случае, если в команде загрузки слова в качестве регистра Rt используется счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу;

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
STR R0, [R5, R1]      ; Записать значение R0 по адресу, равному сумме R5 и R1
LDRSB R0, [R5, R1, LSL #1]
                      ; Считать байт по адресу, равному сумме R5 и R1,
                      ; умноженному на два, распространить значение знакового
                      ; бита на старшие значащие байты слова, загрузить результат
                      ; в регистр R0
STR R0, [R1, R2, LSL #2]
                      ; Сохранить значение регистра R0 по адресу, равному
R1+4*R2
```

LDR, адресация относительно счетчика команд PC

Загрузка регистра из памяти.

Синтаксис

LDR{type}Rt, label

где:
type - один из суффиксов размера данных:

- B - байт без знака, при загрузке старшие байты устанавливаются в нуль.
- SB - байт со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- H - беззнаковое полуслово, при загрузке старшие байты устанавливаются в нуль.
- SH - полуслово со знаком, при загрузке происходит распространение знакового бита в старшие байты (только LDR).
- без суффикса - 32-разрядное слово.

Rt - регистр, в который должна производиться загрузка.
Rt2 - второй регистр, в который должна производиться загрузка.
label - относительный адрес, см. "Адресация относительно счетчика команд".

Описание

LDR - загружает регистра значением из памяти с адресом, заданным в виде метки, относительно счетчика команд PC.

Считываемое значение может иметь размер байта, полуслова или слова. При загрузке данных из памяти байты и полуслова могут интерпретироваться либо как числа со знаком, либо как беззнаковые. См. “Выравнивание адресов”.

Метка должна располагаться на ограниченном расстоянии от текущей инструкции. В таблице 13-19 показаны возможные значения смещений между меткой данных и текущим значением счетчика команд.

Table 13-19. Диапазон значений смещения

Тип инструкции	Диапазон значений смещения
Слово, полуслово со знаком или без знака, байт со знаком или без знака	от 0 до 124
Двойное слово	от 0 до 1020

Ограничения

В данной инструкции:

- использовать в качестве Rt регистры PC или SP можно только в командах чтения слова;
- нельзя использовать в качестве Rt2 регистры PC и SP;
- при загрузке двойных слов регистры Rt и Rt2 не должны совпадать.

В случае, если в команде загрузки слова в качестве регистра Rt используется счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу;

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
LDR R0, LookUpTable     ; Загрузить R0 словом данных по адресу
                        ; с меткой LookUpTable
LDRSB R7, localdata     ; Загрузить байт данных по адресу с меткой localdata,
                        ; распространить значение знакового бита в старшие
                        ; байты слова данных, сохранить результат в R7
```

LDM и STM

Загрузка или сохранение множества регистров.

Синтаксис

op{addr_mode} Rn!, reglist

где:
op - один из кодов операций:

- LDM загрузить множество регистров.
- STM сохранить множество регистров.

addr_mode - один из режимов адресации:

- IA - с увеличением адреса после каждого доступа. Этот режим используется по умолчанию.
- EA - с увеличением адреса после каждого доступа. (только для STM)
- FD - с увеличением адреса после каждого доступа. (только для LDM)

Rn - регистр, содержащий базовый адрес памяти.

! - обязательный суффикс обратной записи значения базового регистра. В случае, если он присутствует в команде, последний адрес, по которому осуществлялся доступ, будет записан обратно в регистр Rn.

reglist - заключенный в фигурные скобки список из одного или нескольких регистров, которые должны быть записаны или считаны. В списке можно указывать диапазон номеров регистров. Начальный и конечный регистр диапазона разделены знаком "-". Элементы списка (отдельные регистры или диапазоны) разделяются запятыми. См. "Примеры".

Мнемокод LDMFD и LDMIA - это псевдокоманды LDM. Использование команды LDMFD обусловлено извлечением данных из полного нисходящего стека, с указателем на последний загруженный элемент (Full Descending stack).

Мнемокод STMEA и STMIA - это псевдокоманы STM. Использование команды STMEA обусловлено сохранением данных в пустой восходящий стек, с указателем на последнюю свободную ячейку (Empty Ascending stack).

Описание

Инструкции LDM осуществляют загрузку регистров из списка reglist значениями слов данных из памяти с базовым адресом, содержащимся в регистре Rn.

Инструкции STM осуществляют сохранение слов данных, содержащихся в регистрах из списка reglist, в память с базовым адресом, содержащимся в регистре Rn.

Команды LDM, LDMIA, LDMFD, STM, STMIA и STMEA для доступа используют адреса памяти в интервале от Rn до Rn+4*(n-1), где n - количество регистров в списке reglist. Доступ осуществляется в порядке увеличения номера регистра, при этом регистр с наименьшим номером соответствует наименьшему адресу памяти, а регистр с наибольшим номером - наибольшему адресу. Значение Rn+4*(n-1) записывается обратно в регистр Rn.

Ограничения

В описываемых в разделе командах:

- в качестве регистра Rn нельзя использовать счетчик команд PC;
- список регистров reglist не может содержать указатель стека SP;
- в любой инструкции STM в списке регистров reglist нельзя указывать PC;
- в любой инструкции LDM в reglist нельзя указывать одновременно PC и LR;

В случае, если инструкция LDM содержит в списке reglist счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу;

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
LDMFD R8!,{R0,R2,R9} ; LDMFD - синоним LDM
STMEA R1!,{R3-R6,R11,R12}; STMEA – синоним STM
```

Примеры неправильного использования

```
STM R5!,{R5,R4,R9}      ; Сохраненное значение R5 является непредсказуемым
LDM R2!, {}             ; Список должен содержать хотя бы один регистр
```

PUSH и POP

Загружает или считывает регистры в стек или из стека, растущего вниз, с указателем на последний загруженный элемент (full-descending stack).

Синтаксис

PUSH reglist
POP reglist

где:
reglist - заключенный в фигурные скобки список из одного или нескольких регистров, которые должны быть записаны или считаны. В списке можно указывать диапазон номеров регистров. Начальный и конечный регистр диапазона разделены знаком "-". Элементы списка (отдельные регистры или диапазоны) разделяются запятыми.

Команды PUSH и POP являются синонимами команд STM и LDM в которых базовый адрес памяти содержится в регистре указателя стека SP, а режим записи обратной записи значения базового регистра включен.

Мнемокоды PUSH и POP являются предпочтительными вариантами записи.

Описание

PUSH - сохраняет регистры в стеке в порядке уменьшения номеров регистров, при этом регистр с большим номером сохраняется в память с большим значением адреса.

POP - восстанавливает значения регистров из стека в порядке увеличения номеров регистров, при этом регистр с меньшим номеров считывается из памяти с меньшим значением адресом.

Ограничения

В данных инструкциях:

- список регистров reglist не должен содержать указатель стека SP;
- в инструкции PUSH список регистров не должен содержать счетчик команд PC;
- в инструкции POP список регистров не должен одновременно содержать регистры PC и LR.

В случае, если инструкция POP содержит в списке reglist счетчик команд PC:

- бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу;

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
PUSH {R0,R4-R7}
PUSH {R2,LR}
POP {R0,R10,PC}
```

Инструкции обработки данных

В таблице 13-20 представлены инструкции обработки данных:

Таблица 13-20. Команды обработки данных

Мнемокод	Краткое описание	Страница
ADC	Сложение с учетом переноса	
ADD	Сложение	
AND	Логическое И	
ASR	Арифметический сдвиг вправо	
BIC	Сброс битов по маске	
CMN	Сравнить с противоположным знаком	
CMP	Сравнить	
EOR	Исключающее ИЛИ	
LSL	Логический сдвиг влево	
LSR	Логический сдвиг вправо	
MOV	Загрузка	
MVN	Загрузка инверсного значения	
ORR	Логическое ИЛИ	
REV	Изменить на обратный порядок байтов в слове	
REV16	Изменить на обратный порядок байтов в полусловах	
REVSH	Изменить на обратный порядок байт в младшем полуслове, произвести распространение знакового бита в старшее полуслово	
ROR	Циклический сдвиг вправо	
RSB	Вычитание с противоположным порядком аргументов	
SBC	Вычитание с учетом переноса	
SUB	Вычитание	
TST	Проверка значения битов по маске	

ADD, ADC, SUB, SBC и RSB

Сложение, сложение с переносом, вычитание, вычитание с переносом, вычитание с противоположным порядком аргументов.

Синтаксис

```
op{S} {Rd,} Rn, Operand2
op {Rd,} Rn, #imm8  ; только для команд ADD, SUB и RSB
```

где:
op - один из кодов операции:

- ADD - сложение.
- ADC - сложение с учетом переноса.
- SUB - вычитание.
- SBC - вычитание с учетом переноса.
- RSB - вычитание с противоположным порядком аргументов.

S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. “Условное исполнение”.

Rd - регистр-получатель результата. В случае, если регистр Rd не указан, результат записывается в Rn.

Rn - регистр, содержащий значение первого операнда.

Operand2 - второй операнд. См. “Формат второго операнда”.

imm8 - любое число в диапазоне от 0 до 1020 (Thumb) или 0-508 (Thumb-2) (для RSB только 0)

Описание

Команда ADD складывает значение Operand2 или imm8 со значением регистра Rn.

Команда ADC складывает вместе значения Rn и Operand2, а также флага переноса.

Команда SUB вычитает значение Operand2 или imm8 из значения регистра Rn.

Команда SBC вычитает значение Operand2 из значения регистра Rn. Если флаг переноса не установлен, результат дополнительно уменьшается на единицу.

Команда RSB вычитает значение регистра Rn из значения Operand2. Этот вариант команды полезен, так как существует широкий выбор вариантов построения Operand2.

Инструкции ADC и SBC полезны при реализации вычислений с повышенной разрядностью, см. “Арифметика с повышенной разрядностью ”.

См. также описание команды ADR.
Ограничения

Для рассматриваемых инструкций:

- в качестве Operand2 нельзя использовать SP или PC;
- использовать SP в качестве регистра Rd допустимо только в командах ADD и SUB, со следующими дополнительными ограничениями:
  - в качестве Rn также должен использоваться SP;
  - сдвиг в Operand2 должен быть не более 3 бит в режиме LSL;
- указатель стека SP может использоваться в качестве Rn только в командах ADD и SUB;
- счетчик команд PC может использоваться в качестве Rd только в команде:
  ADD PC, PC, Rm
  причем:
  - не допускается использование суффикса S;
  - в качестве Rm не допускается использовать PC и SP;
  - если инструкция условная, то она должна быть последней в IT-блоке.
- не считая команды
  ADD PC, PC, Rm
  в качестве регистра Rn можно использовать счетчик команд PC только в инструкциях ADD и SUB с дополнительными ограничениями:
  - не допускается использование суффикса S;
  - второй операнд должен находится в интервале от 0 до 1020.
  - при использовании PC в операциях сложения или вычитания биты [1:0] счетчика команд округляются до 0b00 перед выполнением операции, обеспечивая выравнивание адреса по границе слова.
  - при необходимости сформировать адрес инструкции, необходимо скорректировать значение смещения относительно PC. ARM рекомендует использовать вместо инструкцию ADR, так как в этом ассемблер автоматически сгенерирует правильное смещение.
  - в случае, если PC используется в качестве Rd в команде
    ADD PC, PC, Rm
    бит[0] значения, записываемого в PC, будет проигнорирован, передача управления будет осуществляться по адресу, соответствующему нулевому значению этого бита.

Флаги

В случае, если в команде указан суффикс S, процессор устанавливает флаги N, Z, C и V в соответствии с результатом выполнения операции.

Примеры

```
ADD R2, R1, R3
SUBS R8, R6, #240   ; установить флаги по результату операции вычитания
RSB R4, R4, #0      ; вычесть содержимое регистра R4 из 0
```

Арифметика с повышенной разрядностью

64-разрядное сложение

Следующий пример показывает, как осуществить сложение 64-разрядного целого числа, записанного в паре регистров R2 и R3, с другим 64-разрядным числом, записанным в паре регистров R0 и R1, после чего записывает результат в пару регистров R4 и R5.

```
ADDS R4, R0, R2     ; сложить младшие значащие слова
ADC R5, R1, R3      ; сложить старшие значащие слова с учетом переноса
```

96-разрядное вычитание

Данные с повышенной разрядностью не обязательно содержать в смежных регистрах. В примере, приведенном ниже, показан фрагмент кода, осуществляющий вычитание 96-разрядного целого числа, записанного в регистрах R9, R1 и R11, из другого числа, содержащегося в R6, R2 и R8. Результат записывается в регистрах R6, R9 и R2.

```
SUBS R6, R6, R9     ; вычитание младших значащих слов
SBCS R9, R2, R1     ; вычитание средних значащих слов с переносом
SBC R2, R8, R11     ; вычитание старших значащих слов с переносом
```

AND, ORR, EOR, BIC

Логические операции И, ИЛИ, Исключающее ИЛИ, сброс битов по маске.

Синтаксис

op{S}{Rd,} Rn, Operand2

где:
op - один из кодов операции:

- AND - логическое И.
- ORR - логическое ИЛИ.
- EOR - логическое Исключающее ИЛИ.
- BIC - сброс битов по маске.

S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. “Условное исполнение”.

Rd - регистр-получатель результата.
Rn - регистр, содержащий значение первого операнда.
Operand2 - второй операнд. См. “Формат второго операнда”.

Описание

Инструкции AND, ORR и EOR осуществляют, соответственно, логические операции И, ИЛИ и исключающего ИЛИ между аргументами, содержащимися в регистре Rn и вторым операндом Operand2.

Инструкция BIC выполняет операцию логического И между аргументами, содержащимися в регистре Rn и инверсным значением второго операнда Operand2.

Ограничения

Не допускается использованием указателя стека SP и счетчика команд PC.

Флаги

В случае, если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- может изменить флаг C в ходе вычисления значения второго операнда, см. “Формат второго операнда”;
- не влияет на значение флага V.

Примеры

```
AND R9, R2,#0xFF00
ANDS R9, R8, #0x19
EORS R7, R11, #0x18181818
BIC R0, R1, #0xab
```

ASR, LSL, LSR, ROR

Арифметический сдвиг вправо, логический сдвиг влево, логический сдвиг вправо, циклический сдвиг вправо и циклический сдвиг вправо с переносом.

Синтаксис

op{S} Rd, Rm, Rs
op{S} Rd, Rm, #n

где:
op - один из кодов операции:

- ASR - арифметический сдвиг вправо.
- LSL - логический сдвиг влево.
- LSR - логический сдвиг вправо.
- ROR - циклический сдвиг вправо.

S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. “Условное исполнение”.

Rd - регистр-получатель результата.
Rm - регистр, значение которого должно быть подвергнуто сдвигу.
Rs - регистр, содержащий параметр сдвига. Процессор анализирует только младший значащий байт регистра, таким образом, параметр сдвига может принимать значения от 0 до 255.
n - параметр сдвига. Диапазон допустимых значений параметра зависит от инструкции:

- ASR - от 1 до 32;
- LSL - от 0 до 31;
- LSR - от 1 до 32;
- ROR - от 1 до 31.

Команду
LSL{S} Rd, Rm, #0
рекомендуется записывать в формате
MOV{S} Rd, Rm.

Описание

Команда ASR, LSL, LSR и ROR сдвигает биты регистра Rm влево или вправо на заданное количество позиций, определяемое константой n или содержимым регистра Rs.

Во всех указанных инструкциях результат записывается в регистр Rd, при этом содержание регистра Rm остается неизменным. Детальное описание операций сдвига представлено в разделе “Операции сдвига”.

Ограничения

Не допускается использованием указателя стека SP и счетчика команд PC.

Флаги

В случае, если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- флаг C устанавливается в значение последнего сдвинутого бита, за исключением случая параметра сдвига, равного нулю. См. “Операции сдвига”.

Примеры

```
ASR R7, R8, #9      ; Арифметический сдвиг вправо на 9 бит
LSLS R1, R2, #3     ; Логический сдвиг влево на 3 бита с установкой флагов
LSR R4, R5, #6      ; Логический сдвиг вправо на 6 бит
ROR R4, R5, R6      ; Циклический сдвиг вправо на количество бит, указанное
                    ; в младшем байте регистра R6
```

CMP и CMN

Сравнение и сравнение с противоположным знаком.

Синтаксис

CMP Rn, Operand2
CMN Rn, Operand2

где:

Rm - регистр, содержащий первый операнд.
Operand2 - второй операнд. См. “Формат второго операнда”.

Описание

Данные инструкции осуществляют сравнение значений регистра и второго операнда. По результатам сравнения устанавливаются соответствующие флаги, однако сам результат в регистр не записывается.

Команда CMP вычитает из регистра Rn значение второго операнда Operand2. Она аналогична инструкции SUBS, за исключением того, что не сохраняет результат вычитания.

Команда CMN складывает значения регистра Rn и второго операнда Operand2. Она аналогичная инструкции ADDS, за исключением того, что не сохраняет результат вычитания.

Ограничения

В данных инструкциях:

- не допускается использованием PC;
- в качестве второго операнда Operand2 нельзя использовать SP.

Флаги

Процессор устанавливает флаги N, Z, C и V в соответствии с результатом сравнения.

Примеры

CMP R2, R9
CMN R0, #6400

MOV и MVN

Загрузка в регистр прямого или инверсного значения второго операнда.

Синтаксис

```
MOV{S} Rd, Operand2
MOV Rd, #imm8
MVN{S} Rd, Operand2
```

где:
S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. “Условное исполнение”.

Rd - регистр-получатель результата.
Operand2 - второй операнд. См. “Формат второго операнда”.
imm8 - любое значение в диапазоне от 0 до 255.

Инструкция MVN считывает значение второго операнда Operand2, производит его побитную инверсию, после чего помещает результат в регистр Rd.

Ограничения

Регистры SP и PC допускается использовать в исключительно совместно с инструкцией MOV, при следующих ограничениях:

- второй операнд должен быть регистром без указания параметра сдвига;
- суффикс S не должен быть указан.

В случае, если в качестве Rd используется счетчик команд PC:

- бит [0] значения, загружаемого в PC, игнорируется;
- передача управления осуществляется по адресу, соответствующему загруженному значению с битом [0], принудительно установленным в 0.

Флаги

В случае, если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- может изменить флаг C в ходе вычисления значения второго операнда, см. “Формат второго операнда”;
- не влияет на значение флага V.

Примеры

```
MOVS R11, #0x000B ;Записать значение 0x000B в R11, флаги устанавливаются
MOVS R10, R12     ; Записать регистр R12 в R10, флаги устанавливаются
MOVS R3, #23      ; Записать значение 23 в R3
MOVS R8, SP       ; Записать значение указателя стека в регистр R8
MVNS R2, #0xF     ; Записать значение 0xFFFFFFF0 (инверсия значения 0x0F)
                  ; в регистр R2, установить флаги
```

REV, REV16, REVSH

Изменение порядка битов или байтов в слове.

Синтаксис

op Rd, Rn

где:
op - один из кодов операции:

- REV - изменить на обратный порядок байтов в слове;
- REV16 - изменить на обратный порядок байтов в полусловах;
- REVSH - изменить на обратный порядок байт в младшем полуслове, произвести распространение знакового бита в старшее полуслово.

Rd - регистр-получатель результата.
Rn - регистр, содержащий операнд.

Описание

Инструкции предназначены для изменения формата представления (endianness) данных:

- REV - преобразует 32-разрядное число в формате big-endian в число в формате little-endian и наоборот.
- REV16 - преобразует 32-разрядное число в формате big-endian в число в формате little-endian и наоборот.
- REVSH - выполняет одно из следующих преобразований:
  - 16-разрядное число со знаком в формате big-endian в 32-разрядное число со знаком в формате little-endian;
  - 16-разрядное число со знаком в формате little-endian в 32-bit 32-разрядное число со знаком в формате big-endian.

Ограничения

Нельзя использовать указатель стека SP и счетчик команд PC.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
REV R3, R7      ; Обратить порядок следования байтов в R7, записать в R3
REV16 R0, R0    ; Обратить порядок байтов в каждом 16-битном полуслове R0
REVSH R0, R5    ; Обратить полуслово со знаком
REVHS R3, R7    ; Обратить порядок при условии "больше или равно" (HS)
```

TST

Проверить значение битов по маске, проверить равенство.

Синтаксис

TST Rn, Operand2

где:

Rn - регистр, содержащий первый операнд.
Operand2 - второй операнд. См. “Формат второго операнда”.

Описание

Данные инструкции позволяют проверить значение регистра с учетом значения второго операнда Operand2. По результату устанавливаются флаги, сам результат не сохраняется.

Команда TST выполняет побитовую операцию логического И между значениями Rn и Operand2. Она совпадает с инструкцией ANDS, за исключением того, что не сохраняет результат.

Ограничения

Нельзя использовать указатель стека SP и счетчик команд PC.

Флаги

В случае если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- может изменить флаг C в ходе вычисления значения второго операнда, см. “Формат второго операнда”;
- не влияет на значение флага V.

Примеры

```
TST R0, R1          ; Побитовое И между R0 и R1
                    ; устанавливаются флаги, результат не сохраняется
```

Инструкция умножения

В таблице 13-21 представлена информация о команде умножения:

Таблица 13-21. Инструкции умножения и деления

Мнемокод	Краткое описание	Страница
MUL	Умножение, 32-разрядный результат	

MUL

Умножение или умножение с накоплением (сложением, вычитанием) с использованием 32-разрядных операндов и выдающее 32-разрядный результат.

Синтаксис

```
MUL{S} {Rd,} Rn, Rm        ; Умножение
```

где:
S - необязательный суффикс. Если он указан, результат выполнения операции приводит к установке соответствующих флагов, см. “Условное исполнение”.

Rd - регистр-получатель результата. Если регистр Rd не указан, то в качестве получателя используется регистр Rn.
Rn, Rm - регистры, содержащий значения первого и второго сомножителей.
Ra - регистр, содержащий значение, к которому должно быть прибавлено или вычтено произведение.

Описание

Команда MUL выполняет перемножение значений, содержащихся в регистрах Rn и Rm, после чего сохраняет 32 младших значащих бита произведения в Rd.

Результат выполнения операций не зависит от того, используются ли в качестве операндов числа со знаком или без знака.

Ограничения

Нельзя использовать указатель стека SP и счетчик команд PC.
В случае, если инструкция MUL используется с суффиксом установки флагов S:

- регистры Rd, Rn и Rm должны находиться в диапазоне от R0 до R7;
- регистр Rd должен быть тем же, что и Rm;
- не допускается использование суффикса условного исполнения cond.

Флаги

В случае, если в команде указан суффикс S, процессор:

- устанавливает флаги N и Z в соответствии с результатом выполнения операции;
- не влияет на значение флагов C и V.

Примеры

```
MUL R10, R2, R5          ; R10 = R2 x R5
MULS R0, R2, R2          ; R0 = R2 x R2, установить флаги
```

Команды работы с битовыми полями

В таблице 13-22 показаны инструкции, позволяющие манипулировать последовательностями смежных бит данных в регистрах или битовых полях:

Таблица 13-22. Инструкции упаковки и распаковки данных

Мнемокод команд	Краткое описание	Страница
SXTB	Преобразовать байт со знаком в слово	
SXTH	Преобразовать полуслово со знаком в слово	
UXTB	Преобразовать байт без знака в слово	
UXTH	Преобразовать полуслово без знака в слово	

SXT и UXT

Преобразование байта или полуслова в слово с распространением знакового бита или нулей в старшие значащие разряды.

Синтаксис

SXT*extend* Rd, Rm
UXT*extend* Rd, Rm

где:
Суффикс *extend* может принимать одно из следующих значений:

- B - преобразование 8-битного числа в 32-битное.
- H - преобразование 16-битного числа в 32-битное.

Rd - регистр-получатель результата.
Rm - регистр-источник данных.

Описание

Команда SXTB младшие восемь бит [7:0] регистра Rm, преобразует в 32-разрядное число со знаком путем копирования знакового разряда [7] в биты [31:8], сохраняет результат в регистре Rd.

Команда UXTB младшие восемь бит [7:0] регистра Rm, преобразует в 32-разрядное число без знака путем копирования нуля в биты [31:8], сохраняет результат в регистре Rd.

Команда SXTH младшие шестнадцать бит [15:0] регистра Rm, преобразует в 32-разрядное число со знаком путем копирования знакового разряда [15] в биты [31:16], сохраняет результат в регистре Rd.

Команда UXTH младшие шестнадацать бит [15:0] регистра Rm, преобразует в 32-разрядное число без знака путем копирования нуля в биты [31:16], сохраняет результат в регистре Rd.

Ограничения

Нельзя использовать указатель стека SP и счетчик команд PC.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
SXTH R4, R6          ; младшее полуслово из R6,
                     ; преобразовать в 32-разрядное
                     ; число с распространением знака, записать в R4
UXTB R3, R10         ; младший байт из R10, преобразовать в
                     ; 32-разрядное число, старшие байты заполнить нулями
                     ; записать результат в R3
```

Инструкции передачи управления

В таблице 13-23 представлен список инструкций передачи управления.

Таблица 13-23. Инструкции передачи управления

Мнемокод команды	Краткое описание	Страница
B	Переход	
BL	Переход со связью	
BLX	Косвенный переход со связью	
BX	Косвенный переход	

B, BL, BX и BLX

Команды ветвления.

Синтаксис

B {cond} label
BL label
BX Rm
BLX Rm

где:

B - переход по непосредственно заданному адресу.
BL - переход со связью по непосредственно заданному адресу.
BX - косвенный переход по адресу, заданному значением регистра.
BLX - косвенный переход со связью.

cond - необязательный код условия, см. “Условное исполнение”.
label - относительный адрес, см. "Адресация относительно счетчика команд".

Rm - регистр, содержащий адрес, на который необходимо передать управления. Бит [0] этого регистра должен быть установлен в 1, однако передача управления будет выполнена по адресу, соответствующему значению бита [0], равному 0.

Описание

Все рассматриваемые в данном разделе инструкции осуществляют передачу управления на адрес, заданный меткой либо содержащийся в регистре Rm. кроме того:

- команды BL и BLX записывают адрес следующей инструкции в регистр связи LR (R14).
- команды BX и BLX формируют отказ (Hard fault) в случае, если bit[0] регистра Rm равен 0.

В таблице 13-24 представлен диапазон адресуемых переходов для различных команд ветвления.

Таблица 13-24. Диапазон адресуемых переходов для команд ветвления

Инструкция	Диапазон адресации
B {cond} label	от -1 МБайт до +1 МБайт относительно текущей позиции
BL label	от -16 МБайт до +16 МБайт относительно текущей позиции
BX Rm	любое значение, записанное в регистре
BLX Rm	любое значение, записанное в регистре

13.16.1.3 Ограничения

- в команде BLX не допускается использование регистра PC;
- в командах BX и BLX, бит [0] регистра Rm должен быть установлен в 1, при этом передача управления будет, тем не менее, осуществлена по адресу, соответствующему значению бита [0], равного 0;
- B {cond} - единственная условно исполняемая команда
- команды BLX и BX выполняется только в режиме Thumb. При попытке изменить режим при выполнении инструкции возникает исключение Hard Fault

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
B loopA      ; передача управления на метку loopA
BLE ng       ; условная передача управления на метку ng
BEQ target   ; условный переход на метку target
BL funC      ; переход со связью (вызов функции) funC, адрес возврата будет
             ; записан в регистре LR
BX LR        ; возврат из функции
BLX R0       ; переход со связью (вызов функции) по адресу, записанному в R0
```

Прочие инструкции

В таблице 13-25 представлен список не рассмотренных в предыдущих разделах инструкций процессора:

Таблица 13-25. Прочие инструкции

Мнемокод команды	Краткое описание	Страница
BKPT	Точка останова	
CPSID	Изменить состояние процессора, запретить прерывания	
CPSIE	Изменить состояние процессора, разрешить прерывания	
CPY	Аналогична MOV	
DMB	Барьер синхронизации доступа к памяти данных	
DSB	Барьер синхронизации доступа к памяти данных	
ISB	Барьер синхронизации доступа к инструкциям	
MRS	Загрузка из специального регистра в регистр общего назначения	
MSR	Записать регистр общего назначения в специальный регистр	
NOP	Нет операции	
SEV	Установить признак события	
SVC	Вызов супервизора	
YIELD	Инструкция hint. Применяется в мультипотоковых приложениях	

CPS

Изменить состояние процессора.

Синтаксис

CPS*effect* iflags

где:
effect - один из возможных суффиксов:

- IE - сбрасывает специальный регистр в 0.
- ID - устанавливает специальный регистр в 1.

iflags - последовательность флагов:

- i - сбрасывает или устанавливает регистр PRIMASK.

Описание

Команда CPS позволяет изменить значение специального регистра PRIMASK.

Ограничения

- команда CPS доступна только из привилегированного приложения, при вызове из непривилегированного приложения она игнорируется;

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
CPSID i     ; Запретить прерывания и конфигурируемые обработчики отказов
CPSIE i     ; Разрешить прерывания и конфигурируемые обработчики отказов
```

DMB

Барьер синхронизации доступа к памяти данных.

Синтаксис

DMB

Описание

Команда DMB выполняет функцию барьерной синхронизации доступа к памяти данных. Она гарантирует, что все явные (explicit) операции доступа к памяти, которые были инициированы перед выполнением инструкции DMB, будут завершены до того, как начнется выполнение любой операции доступа к памяти после этой инструкции.

Команда DMB не влияет на очередность и порядок выполнения инструкций, не выполняющих доступа к памяти.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
DMB   ; Барьер синхронизации доступа к памяти данных
```

DSB

Барьер синхронизации доступа к памяти данных.

Синтаксис

DSB

Описание

Инструкция DSB выполняет функцию барьерной синхронизации доступа к памяти данных. Команды, которые будут следовать, в порядке выполнения, после DSB, не начнут исполняться до ее завершения. Инструкция DSB завершает свою работу после того, как будут выполнены все инициированные перед ней явные (explicit) операции доступа к памяти.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
DSB    ; Data Synchronisation Barrier
```

ISB

Барьер синхронизации доступа к инструкциям.

Синтаксис

ISB

Описание

Команда ISB выполняет функцию барьерной синхронизации выполнения команд. Она осуществляет сброс конвейера инструкций процессора, гарантируя таким образом, что все команды, расположенные после инструкции ISB, по окончании ее исполнения будут загружены в конвейер повторно.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
ISB     ; Барьер синхронизации доступа к инструкциям
```

MRS

Считать содержимое специального регистра в регистр общего назначения.

Синтаксис

MRS Rd, spec_reg

Rd - регистр-получатель результата.
spec_reg - один из специальных регистров: APSR, IPSR, EPSR, IEPSR, IAPSR, EAPSR, PSR, MSP, PSP, PRIMASK или CONTROL.

Описание

Инструкции MRS совместно с MSR используются для чтения-модификации-записи элементов PSR, например, для сброса флага Q.

В коде, отвечающем за переключение процессов, необходимо обеспечить сохранение состояния приостановленного процесса, и восстановление состояния активизированного процесса. Необходимой составной частью сохраняемой (восстанавливаемой) информации является значение регистра PSR. При этом на этапе сохранения состояния используется команда MRS, а на этапе восстановления - команда MSR.

См. также описание инструкции MSR.

Ограничения

В качестве регистра-получателя Rd нельзя использовать SP или PC.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

MRS R0, PRIMASK ; Считать значение PRIMASK и записать значение в R0

MSR

Записать регистр общего назначения в специальный регистр.

Синтаксис

MSR spec_reg, Rn

Rn - регистр-источник данных.
spec_reg - один из специальных регистров: APSR, IPSR, EPSR, IEPSR, IAPSR, EAPSR, PSR, MSP, PSP, PRIMASK или CONTROL.

Описание

Доступ к специальным регистрам в команде MSR различен для привилегированных и непривилегированных приложений. Непривилегированному приложению доступен только регистр APSR. При этом попытки записи в нераспределенные биты, а также в EPSR игнорируются.

Привилегированное приложение имеет доступ ко всем специальным регистрам.

См. также описание инструкции MRS.

Ограничения

В качестве регистра-источника данных Rn нельзя использовать SP или PC.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
MSR CONTROL, R1          ; Записать значение регистра R1 в регистр CONTROL
```

NOP

Нет операции.

Синтаксис

NOP

Описание

Инструкция NOP ничего не делает. В частности, эта инструкция в некоторых случаях может быть автоматически исключена из конвейера команд, и таким образом, выполнена за ноль тактов. Команду NOP рекомендуется использовать для заполнения, например, с целью разместить очередную инструкцию по адресу, выровненному по 64-битной границе.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
NOP    ; нет операции
```

SEV

Установить признак события.

Синтаксис

SEV

Описание

Инструкция SEV используется для передачи информации о событии всем процессорам в составе многопроцессорной системы. Кроме того, он устанавливает собственный регистр события в 1.

См. также “Управление электропитанием”.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

```
SEV     ; Послать признак события
```

SVC

Вызов супервизора.

Синтаксис

SVC #imm8

где:
imm8 - константное выражение, целое число в диапазоне от 0 до 255 (8-битное число).

Описание

Инструкция SVC вызывает формирование исключения SVC. Параметр imm8 игнорируется процессором. При необходимости он может быть получен обработчиком исключения для определения запрошенного приложением варианта обслуживания.

Флаги

Данная инструкция не влияет на состояние флагов.

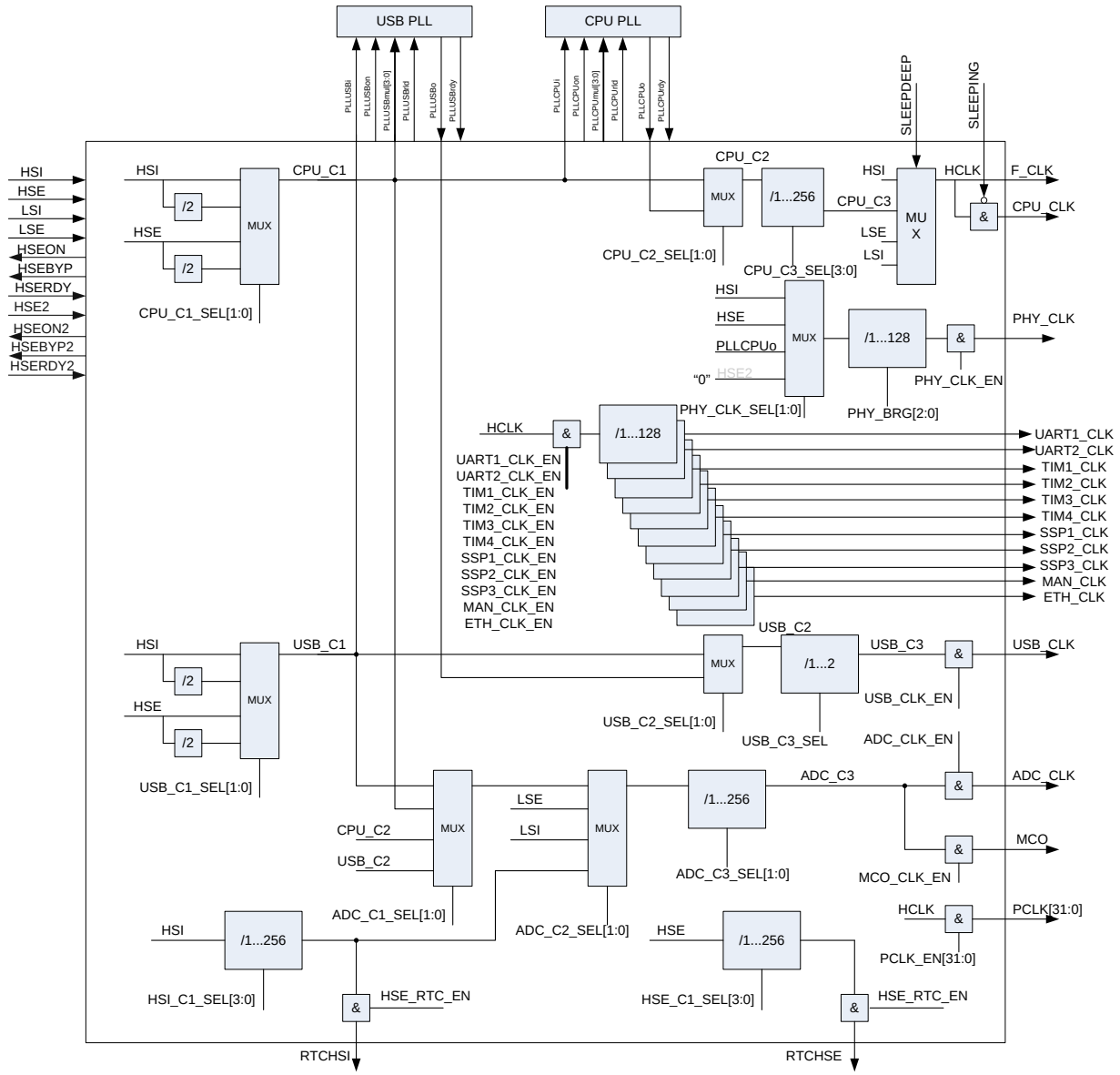
Примеры

```
SVC 0x32      ; Вызов супервизора
              ; обработчик SVC может извлечь параметр по сохраненному в стеке,
              ; адресу PC приложения.
```

Сигналы тактовой частоты.

Микроконтроллер имеет два встроенных генератора и два внешних осциллятора. А так же специализированный блок формирования тактовой синхронизации микроконтроллера.

Структурная схема



Встроенный RC Генератор HSI

Генератор HSI вырабатывает тактовую частоту 8 МГц. Генератор автоматически запускается при появлении питания Ucc и при выходе в нормальный режим работы вырабатывает сигнал HSIRDY в регистре батарейного домена BKP_REG_0F. Первоначально процессорное ядро запускается на тактовой частоте HSI. При дальнейшей работе генератор HSI может быть отключен при помощи сигнала HSION в регистре

BKP_REG_0F. Так же генератор может быть подстроен при помощи сигнала HSITRIM в регистре BKP_REG_0F.

Встроенный RC генератор LSI

Генератор LSI вырабатывает тактовую частоту 40КГц. Генератор автоматически запускается при появлении питания Ucc и при выходе в нормальный режим работы вырабатывает сигнал LSIRDY в регистре BKP_REG_0F. Первоначально тактовая частота генератор LSI используется для формирования дополнительной задержки tpor. При дальнейшей работе генератор LSI может быть отключен при помощи сигнала LSION в регистре BKP_REG_0F.

Внешний осциллятор HSE

Осциллятор HSE предназначен для выработки тактовой частоты 2…16 МГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания Ucc и сигнала разрешения HSEON в регистре HS_CONTROL. При выходе в нормальный режим работы вырабатывает сигнал HSERDY в регистре CLOCK_STATUS. Также осциллятор может работать в режиме HSEBYP, когда входная тактовая частота с входа OSC_IN проходит напрямую на выход HSE, выход OSC_OUT находится в этом режиме в третьем состоянии.

Внешний осциллятор LSE

Осциллятор LSE предназначен для выработки тактовой частоты 32 КГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания BDUcc и сигнала разрешения LSEON в регистре BKP_REG_0F . При выходе в нормальный режим работы вырабатывает сигнал LSERDY в регистре BKP_REG_0F. Так же осциллятор может работать в режиме LSEBYP, когда входная тактовая частота с входа OSC_IN32 проходит напрямую на выход LSE. Выход OSC_OUT32 находится в этом режиме в третьем состоянии. Так как генератор LSE питается от напряжения питания BDUcc и его регистр управления BKP_REG_0F расположен в батарейном домене, то генератор может продолжать работать при пропадании основного питания Ucc. Генератор LSE используется для работы часов реального времени.

Встроенный блок умножения системной тактовой частоты

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 16, задаваемый на входе PLLCPUMUL[3:0] в регистре PLL_CONTROL. Входная частота блока умножителя должна быть в диапазоне 2…16МГц выходная до 144 МГц. При выходе блока умножителя тактовой частоты в расчетный режим вырабатывается сигнал PLLCPURDY в регистре CLOCK_STATUS. Блок включается с помощью сигнала PLLCPUON в регистре PLL_CONTROL. Выходная частота используется как основная частота процессора и периферии.

Встроенный блок умножения тактовой частоты для контроллера USB

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 16, задаваемый на входе PLLUSBMUL[3:0] в регистре PLL_CONTROL. Входная частота блока умножителя должна быть в диапазоне 2…16МГц выходная должна составлять 48 МГц. При выходе блока умножителя тактовой частоты в расчетный режим вырабатывается сигнал PLLUSBRDY в регистре CLOCK_STATUS.

Блок включается с помощью сигнала PLLUSBON в регистре PLL_CONTROL. Выходная частота используется как основная частота протокольной части USB интерфейса.

Управление тактовыми частотами ведется через периферийный блок RST_CLK. При включении питания микроконтроллер запускается на частоте HSI генератора. Выдача тактовых сигналов синхронизации для всех периферийных блоков кроме RST_CLK отключена. Для начала работы с нужным периферийным блоком необходимо включить его тактовую частоту в регистре PER_CLOCK. Некоторые контроллеры интерфейсов (UART, CAN, USB, Таймеры) могут работать на частотах отличных от частоты процессорного ядра, поэтому в соответствующих регистрах (UART_CLOCK, CAN_CLOCK, USB_CLOCK, TIM_CLOCK) могут быть заданы их скорости работы. Для изменения тактовой частоты ядра можно перейти на другой генератор и/или воспользоваться блоком умножения тактовой частоты. Для корректной смены тактовой частоты сначала должны быть сформированы необходимые тактовые частоты и за тем осуществлено переключение на них на соответствующих мультиплексорах управляемых регистрами CPU_CLOCK и USB_CLOCK.

Описание регистров блока контроллера тактовой частоты

Базовый Адрес	Название	Описание
0x4002_0000	RST_CLK	Контроллер тактовой частоты
Смещение		
0x00	CLOCK_STATUS	Регистр состояния блока управления тактовой частотой
0x04	PLL_CONTROL	Регистр управления блоками умножения частоты
0x08	HS_CONTROL	Регистр управления высокочастотным генератором и осциллятором
0x0C	CPU_CLOCK	Регистр управления тактовой частотой процессорного ядра
0x10	USB_CLOCK	Регистр управления тактовой частотой контроллера USB
0x14	ADC_MCO_CLOCK	Регистр управления тактовой частотой АЦП
0x18	RTC_CLOCK	Регистр управления формированием высокочастотных тактовых сигналов блока RTC
0x1C	PER_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x20	CAN_CLOCK	Регистр управления тактовой частотой CAN

0x24	TIM_CLOCK	Регистр управления тактовой частотой TIMER
0x28	UART_CLOCK	Регистр управления тактовой частотой UART
0x2C	SSP_CLOCK	Регистр управления тактовой частотой SSP
0x34	ETH_CLOCK	Регистр управления тактовой частотой Ethernet и ГОСТР52070-2003

CLOCK_STATUS

Номер	31			4	3	2	1	0
Доступ	U			U	RO	RO	RO	RO
Сброс	0			0	0	0	0	0
	-	-	-	-	HSE RDY2	HSE RDY	PLL CPU RDY	PLL USB RDY

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3	HSE RDY2	Флаг выхода в рабочий режим осциллятора HSE2 с ревизии 2 0 – осциллятор не запущен или не стабилен 1 – осциллятор запущен и стабилен
2	HSE RDY	Флаг выхода в рабочий режим осциллятора HSE 0 – осциллятор не запущен или не стабилен 1 – осциллятор запущен и стабилен
1	PLL CPU RDY	Флаг выхода в рабочий режим CPU PLL 0 – PLL не запущена или не стабильна 1 – PLL запущена и стабильна
0	PLL USB RDY	Флаг выхода в рабочий режим USB PLL 0 – PLL не запущена или не стабильна 1 – PLL запущена и стабильна

PLL_CONTROL

Номер	31	12	11…8	7…4	3	2	1	0
Доступ	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0000	0000	0	0	0	0
	-	-	PLL CPU MUL[3:0]	PLL USB MUL[3:0]	PLL CPU PLD	PLL CPU ON	PLL USB RLD	PLL USB ON

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11...8	PLL CPU MUL[3:0]	Коэффициент умножения для CPU PLL: PLLCPUo = PLLCPUi x (PLLCPUMUL+1)
7..4	PLL USB MUL[3:0]	Коэффициент умножения для USB PLL: PLLUSBo = PLLUSBi x (PLLUSBMUL+1)
3	PLL CPU PLD	Бит перезапуска PLL При смене коэффициента умножения в рабочем режиме необходимо задать равным 1, а после этого сбросить в ноль

2	PLL CPU ON	Бит включения PLL 0 – PLL выключена 1 – PLL включена
1	PLL USB RLD	Бит перезапуска PLL При смене коэффициента умножения в рабочем режиме необходимо задать равным 1, а после этого сбросить в ноль
0	PLL USB ON	Бит включения PLL 0 – PLL выключена 1 – PLL включена

HS_CONTROL

Номер	31			4	3	2	1	0
Доступ	U			U	R/W	R/W	R/W	R/W
Сброс	0			0	0	0	0	0
	-	-	-	-	HSE BYP2	HSE ON2	HSE BYP	HSE ON

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3	HSE BYP2	Бит управления HSE2 осциллятором с ревизии2 0 – режим осциллятора 1 – режим внешнего генератора
2	HSE ON2	Бит управления HSE2 осциллятором с ревизии2 0 – выключен 1 – включен
1	HSE BYP	Бит управления HSE осциллятором 0 – режим осциллятора 1 – режим внешнего генератора
0	HSE ON	Бит управления HSE осциллятором 0 – выключен 1 – включен

CPU_CLOCK

Номер	31			10	9…8	7…4	3	2	1…0
Доступ	U			U	R/W	R/W	U	R/W	R/W
Сброс	0			0	00	0000	0	0	00
	-	-	-	HCLK SEL[1:0]	CPU C3 SEL[3:0]	-	CPU C2 SEL	CPU C1 SEL[1:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..10	-	Зарезервировано
9…8	HCLK SEL[1:0]	Биты выбора источника для HCLK 00 – HSI 01 – CPU_C3 10 – LSE 11 – LSI
7…4	CPU C3 SEL[3:0]	Биты выбора делителя для CPU_C3 0xxx – CPU_C3 = CPU_C2; 1000 - CPU_C3 = CPU_C2 / 2; 1001 - CPU_C3 = CPU_C2 / 4; 1010 - CPU_C3 = CPU_C2 / 8; …

		1111 - CPU_C3 = CPU_C2 / 256;
3	-	Зарезервировано
2	CPU C2 SEL	Биты выбора источника для CPU_C2 0 – CPU_C1 1 – PLLCPUo
1…0	CPU C1 SEL[1:0]	Биты выбора источника для CPU_C1 00 – HSI 01 – HSI/2 10 – HSE 11 – HSE/2

USB_CLOCK

Номер	31	8	7…5	4	3	2	1…0
Доступ	U	R/W	U	R/W	U	R/W	R/W
Сброс	0	0	000	0	0	0	00
	-	USB_CLK_EN	-	USB_C3_SEL	-	USB_C2_SEL	USB_C1_SEL[1:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..10	-	Зарезервировано
8	USB_CLK_EN	Бит разрешения тактирования USB 0 – нет тактовой частоты 1 – есть тактовая частота
7…5	-	Зарезервировано
4	USB_C3_SEL	Биты выбора делителя для USB_C3 USB_C3 = USB_C2/(USB_C3_SEL+1);
3	-	Зарезервировано
2	USB_C2_SEL	Биты выбора источника для USB_C2 0 – USB_C1 1 – PLLUSBo
1…0	USB_C1_SEL[1:0]	Биты выбора источника для USB_C1 00 – HSI 01 – HSI/2 10 – HSE 11 – HSE/2

ADC_MCO_CLOCK

Номер	31…14	13	12	11…8	7…6	5…4	3…2	1…0
Доступ	U	R/W	U	R/W	U	R/W	U	R/W
Сброс	0	0	0	0000	00	00	00	00
	-	ADC CLK EN	-	ADC C3 SEL[3:0]	-	ADC C2 SEL[1:0]	-	ADC C1 SEL[1:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..14	-	Зарезервировано
13	ADC CLK EN	Бит разрешения выдачи тактовой частоты ADC CLK 0 – запрещен 1 – разрешен
12	-	Зарезервировано
11…8	ADC C3 SEL[3:0]	Биты выбора делителя для ADC_C3 0xxx – ADC_C3 = ADC_C2; 1000 - ADC_C3 = ADC_C2 / 2; 1001 - ADC_C3 = ADC_C2 / 4; 1010 - ADC_C3 = ADC_C2 / 8; … 1111 - ADC_C3 = ADC_C2 / 256;
7..6	-	Зарезервировано
5…4	ADC C2 SEL[1:0]	Биты выбора источника для ADC_C1 00 – LSE 01 – LSI 10 – ADC_C1 11 – HSI_C1
3…2	-	Зарезервировано
1…0	ADC C1 SEL[1:0]	Биты выбора источника для ADC_C1 00 – CPU_C1 01 – USB_C1 10 – CPU_C2 11 – USB_C2

RTC_CLOCK

Номер	31…14			10	9	8	7…4	3…0
Доступ	U			U	R/W	R/W	R/W	R/W
Сброс	0			0	0	0	0000	0000
	-			-	HSI RTC EN	HSE RTC EN	HSI SEL[1:0]	HSE SEL[1:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..10	-	Зарезервировано
9	HSI RTC EN	Бит разрешения HSI RTC 0 – запрещен 1 – разрешен
8	HSE RTC EN	Бит разрешения HSE RTC 0 – запрещен 1 – разрешен
7…4	HSI SEL[3:0]	Биты выбора делителя для HSI_C1 0xxx –HSI_C1 = HSI; 1000 - HSI_C1 = HSI / 2; 1001 - HSI_C1 = HSI / 4; 1010 - HSI_C1 = HSI / 8; … 1111 - HSI_C1 = HSI / 256;
3…0	HSE SEL[3:0]	Биты выбора делителя для HSE_C1 0xxx –HSE_C1 = HSE; 1000 - HSE_C1 = HSE / 2; 1001 - HSE_C1 = HSE / 4; 1010 - HSE_C1 = HSE / 8; … 1111 - HSE_C1 = HSE / 256;

PER_CLOCK

Номер	31		0
Доступ	R/W		R/W
Сброс	0		0
	PCLK_EN[31:0]		

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…0	PCLK EN[31:0]	Биты разрешения тактирования периферийных блоков 0 – запрещено 1 – разрешено PCLK[0] – CAN1

		PCLK[1] – CAN2 PCLK[2] – USB PCLK[3] – EEPROM_CNTRL PCLK[4] – RST_CLK PCLK[5] – DMA PCLK[6] – UART1 PCLK[7] – UART2 PCLK[8] – SPI1 PCLK[9] – MIL-STD-1553B1 PCLK[10] – MIL-STD-1553B2 PCLK[11] – POWER PCLK[12] – WWDT PCLK[13] – IWDT PCLK[14] – TIMER1 PCLK[15] – TIMER2 PCLK[16] – TIMER3 PCLK[17] – ADC PCLK[18] – DAC PCLK[19] – TIMER4 PCLK[20] – SPI2 PCLK[21] – PORTA PCLK[22] – PORTB PCLK[23] – PORTC PCLK[24] – PORTD PCLK[25] – PORTE PCLK[26] – ARINC429R PCLK[27] – BKP PCLK[28] – ARINC429T PCLK[29] – PORTF PCLK[30] – EXT_BUS_CNTRL PCLK[31] – SPI3

CAN_CLOCK

Номер	31	26	25	24	23…16	15…8	7…0
Доступ	U	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	0	00000000	00000000
	-	-	-	CAN2 CLK EN	CAN1 CLK EN	-	CAN2 BRG [7:0]	CAN1 BRG [7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..26	-	Зарезервировано
25	CAN2 CLK EN	Разрешение тактовой частоты на CAN2 0 – нет частоты 1 – есть частота
24	CAN1 CLK	Разрешение тактовой частоты на CAN1

	EN	0 – нет частоты 1 – есть частота
23..16	-	Зарезервировано
15…8	CAN2 BRG [7:0]	Делитель тактовой частоты CAN2 xxxxx000 – CAN2_CLK == HCLK xxxxx001 – CAN2_CLK == HCLK/2 xxxxx010 – CAN2_CLK == HCLK/4 … xxxxx111 – CAN2_CLK == HCLK/128
7…0	CAN1 BRG [7:0]	Делитель тактовой частоты CAN1 xxxxx000 – CAN1_CLK == HCLK xxxxx001 – CAN1_CLK == HCLK/2 xxxxx010 – CAN1_CLK == HCLK/4 … xxxxx111 – CAN1_CLK == HCLK/128

TIM_CLOCK

Номер	31	26	25	24	23…16	15…8	7…0
Доступ	U	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	0	00000000	00000000
	-	-	TIM3 CLK EN	TIM2 CLK EN	TIM1 CLK EN	TIM3 BRG [7:0]	TIM2 BRG [7:0]	TIM1 BRG [7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..27	-	Зарезервировано
26	TIM3 CLK EN	Разрешение тактовой частоты на TIM3 0 – нет частоты 1 – есть частота
25	TIM2 CLK EN	Разрешение тактовой частоты на TIM2 0 – нет частоты 1 – есть частота
24	TIM1 CLK EN	Разрешение тактовой частоты на TIM1 0 – нет частоты 1 – есть частота
23..16	TIM3 BRG [7:0]	Делитель тактовой частоты TIM3 xxxxx000 – TIM3_CLK == HCLK xxxxx001 – TIM3_CLK == HCLK/2 xxxxx010 – TIM3_CLK == HCLK/4 … xxxxx111 – TIM3_CLK == HCLK/128
15…8	TIM2 BRG [7:0]	Делитель тактовой частоты TIM2 xxxxx000 – TIM2_CLK == HCLK xxxxx001 – TIM2_CLK == HCLK/2 xxxxx010 – TIM2_CLK == HCLK/4 … xxxxx111 – TIM2_CLK == HCLK/128
7…0	TIM1 BRG [7:0]	Делитель тактовой частоты TIM1 xxxxx000 – TIM1_CLK == HCLK xxxxx001 – TIM1_CLK == HCLK/2 xxxxx010 – TIM1_CLK == HCLK/4 … xxxxx111 – TIM1_CLK == HCLK/128

UART_CLOCK

Номер	31	26	25	24	23…16	15…8	7…0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	00000000	00000000
	-	-	TIM4 CLK EN	UART2 CLK EN	UART 1 CLK EN	TIM4 BRG [7:0]	UART 2 BRG [7:0]	UART 1 BRG [7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..27	-	Зарезервировано
26	TIM4 CLK EN	Разрешение тактовой частоты на TIM4 0 – нет частоты 1 – есть частота
25	UART2 CLK EN	Разрешение тактовой частоты на UART2 0 – нет частоты 1 – есть частота
24	UART1 CLK EN	Разрешение тактовой частоты на UART 1 0 – нет частоты 1 – есть частота
23..16	TIM4 BRG [7:0]	Делитель тактовой частоты TIM4 xxxxx000 – TIM4_CLK == HCLK xxxxx001 – TIM4_CLK == HCLK/2 xxxxx010 – TIM4_CLK == HCLK/4 … xxxxx111 – TIM4_CLK == HCLK/128
15…8	UART2 BRG [7:0]	Делитель тактовой частоты UART 2 xxxxx000 – UART 2_CLK == HCLK xxxxx001 – UART 2_CLK == HCLK/2 xxxxx010 – UART 2_CLK == HCLK/4 … xxxxx111 – UART 2_CLK == HCLK/128
7…0	UART1 BRG [7:0]	Делитель тактовой частоты UART 1 xxxxx000 – UART 1_CLK == HCLK xxxxx001 – UART 1_CLK == HCLK/2 xxxxx010 – UART 1_CLK == HCLK/4 … xxxxx111 – UART 1_CLK == HCLK/128

SSP_CLOCK

Номер	31	26	25	24	23…16	15…8	7…0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	00000000	00000000	00000000
	-	-	SSP3 CLK EN	SSP2 CLK EN	SSP 1 CLK EN	SSP 3 BRG [7:0]	SSP 2 BRG [7:0]	SSP 1 BRG [7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..27	-	Зарезервировано
26	SSP3 CLK EN	Разрешение тактовой частоты на SSP 3 0 – нет частоты 1 – есть частота
25	SSP2 CLK EN	Разрешение тактовой частоты на SSP 2 0 – нет частоты 1 – есть частота
24	SSP1 CLK EN	Разрешение тактовой частоты на SSP 1 0 – нет частоты 1 – есть частота
23..16	SSP3 BRG [7:0]	Делитель тактовой частоты SSP 3 xxxxx000 – SSP 3_CLK == HCLK xxxxx001 – SSP 3_CLK == HCLK/2 xxxxx010 – SSP 3_CLK == HCLK/4 … xxxxx111 – SSP 3_CLK == HCLK/128
15…8	SSP2 BRG [7:0]	Делитель тактовой частоты SSP 2 xxxxx000 – SSP 2_CLK == HCLK xxxxx001 – SSP 2_CLK == HCLK/2 xxxxx010 – SSP 2_CLK == HCLK/4 … xxxxx111 – SSP 2_CLK == HCLK/128
7…0	SSP1 BRG [7:0]	Делитель тактовой частоты SSP 1 xxxxx000 – SSP 1_CLK == HCLK xxxxx001 – SSP 1_CLK == HCLK/2 xxxxx010 – SSP 1_CLK == HCLK/4 … xxxxx111 – SSP 1_CLK == HCLK/128

ETH_CLOCK

Номер	29..28	27	26	25	24	23…16	15…8	7…0
Доступ	R/W		R/W		R/W	R/W	R/W	R/W
Сброс	00	0	0	0	0	00000000	00000000	00000000
	PHY CLK SEL	PHY CLK EN	SLEEP	MAN CLK EN	ETH CLK EN	PHY BRG [7:0]	MAN BRG [7:0]	ETH BRG [7:0]

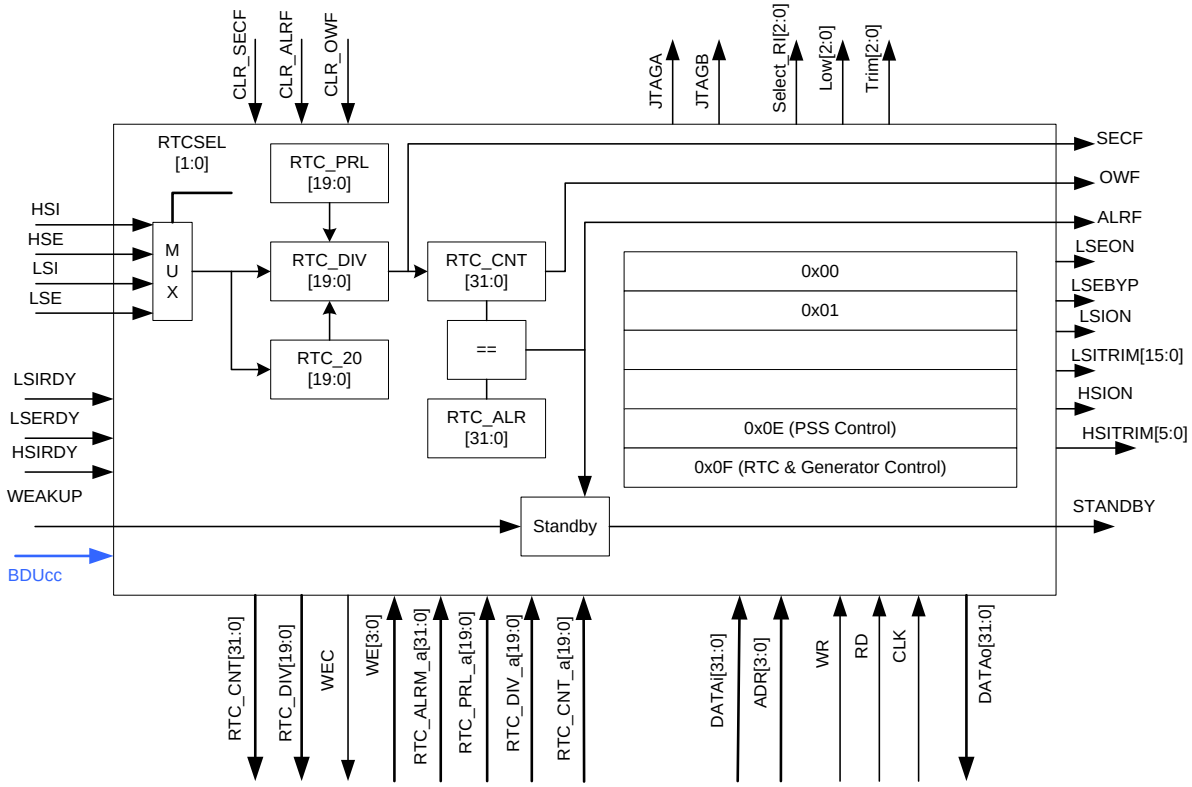
№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…30	Зарезервировано	
29..28	PHY_CLK_SEL [1:0]	Биты выбора источника частоты для Ethernet PHY 00 – HSI 01 – HSE 10 – PLLCPUo 11 – HSE2 с ревизии 2
27	PHY CLK EN	Разрешение тактовой частоты Ethernet PHY 0 – нет частоты 1 – есть частота
26	SLEEP	Перевод ядра контроллера в режим пониженного электропотребления 0- рабочий режим 1-режим пониженного электропотребления В этом режиме тактовая частота поступает только на выбранные периферийные блоки, прерывание от которых возобновляет подачу тактовой частоты на ядро.
25	MAN CLK EN	Разрешение тактовой частоты на контроллер ГОСТР52070-2003 0 – нет частоты 1 – есть частота
24	ETH CLK EN	Разрешение тактовой частоты на Ethernet MAC 0 – нет частоты 1 – есть частота
23..16	PHY BRG [7:0]	Делитель тактовой частоты PHY xxxxx000 – PHY_CLK == PHY1_CLK xxxxx001 – PHY_CLK == PHY1_CLK/2 xxxxx010 – PHY_CLK == PHY1_CLK/4 xxxxx011 – PHY_CLK == PHY1_CLK/8 … xxxxx111 – PHY_CLK == PHY1_CLK/128
15…8	MAN BRG [7:0]	Делитель тактовой частоты контроллера ГОСТР52070-2003 xxxxx000 – MAN_CLK == HCLK xxxxx001 – MAN_CLK == HCLK/2 xxxxx010 – MAN_CLK == HCLK/4 …

		xxxxx111 – MAN_CLK == HCLK/128
7…0	ETH BRG [7:0]	Всегда задавать 0

Батарейный домен и часы реального времени.

Блок батарейного домена предназначен для обеспечения функций часов реального времени и сохранения некоторого набора пользовательских данных при отключении основного источника питания. При снижении питания Ucc в блоке SW происходит автоматическое переключение питания BDUcc с Ucc на BUcc. Если на BUcc имеется отдельный источник питания (батарейка), то батарейный домен остается включенным и может выполнять свои функции.

Структурная схема:



Часы реального времени

Часы реального времени позволяют организовать механизм отсчета времени в кристалле, в том числе при отключении основного источника питания. Включение часов реального времени осуществляется битом RTCEN. В качестве источника тактовой частоты часов реального времени может выступать генератор LSI, осциллятор LSE, HSE, HSI с дополнительным делителем до 256 (HSE и HSI формируются в блоке управления тактовыми частотами и могут быть выбраны только при наличии питания DUcc, LSI может быть выбран при наличии питания Ucc, LSE может быть выбран при наличии Ucc или BUcc). Выбор между источниками осуществляется битами RTCSEL. При возможном отключении основного источника питания Ucc в качестве источника тактовой частоты должен использоваться осциллятор LSE, так как он также имеет питание BDUcc. Биты управления осциллятором LSE расположены в батарейном домене и, таким образом, при отключении основного питания они не сбрасываются. При этом при первоначальном включении эти биты так же не определены и могут принять любое значение.

Для калибровки тактовой частоты используются биты CAL[6:0]. Значение CAL определяет, какое число тактов из 2^20 будет замаскировано. Таким образом, с помощью битов CAL производится замедление хода часов. Изменение значения битов CAL может быть осуществлено в ходе работы часов реального времени.

Регистр RTC_DIV выступает в роли 20-ти битного предварительного делителя входной тактовой частоты, таким образом, чтобы на его выходе была тактовая частота в 1 Гц. Для задания коэффициента деления регистра RTC_DIV используется регистр RTC_PRL.

Регистр RTC_CNT предназначен для отсчета времени в секундах. И работает на выходной частоте делителя RTC_DIV. Регистр RTC_ALR предназначен для задания времени, при совпадении с которым вырабатывается флаг прерывания и пробуждения процессора. Таким образом, бит STANBY, отключающий внутренний регулятор напряжения автоматически сбрасывается при совпадении RTC_CNT и RTC_ALR.

Бит STANDBY так же может быть сброшен с помощью вывода WAKEUP.

Регистры аварийного сохранения

Батарейный домен имеет 16 встроенных 32-х разрядных регистров аварийного сохранения. 16-тый регистр служит для хранения битов управления батарейным доменом, оставшиеся 15 регистров могут быть использованы разработчиком программы.

Описание регистров блока батарейного домена

Базовый Адрес	Название	Описание
0x400D_8000	BKP	Контроллер батарейного домена и часов реального времени.
Смещение		
0x00	BKP_REG_00	Регистр аварийного сохранения 0
…		
0x38	BKP_REG_0E	Регистр аварийного сохранения 14
0x3C	BKP_REG_0F	Регистр аварийного сохранения 15 и управления блоками RTC, LSE, LSI и HSI
0x40	RTC_CNT	Регистр основного счетчика часов реального времени
0x44	RTC_DIV	Регистр предварительного делителя основного счетчика
0x48	RTC_PRL	Регистр основания счета предварительного делителя
0x4C	RTC_ALRM	Регистр значения для сравнения основного счетчика и выработки сигнала ALRF

0x50	RTC_CS	Регистр управления и состояния флагов часов реального времени

BKP_REG_00

BKP_REG_01

BKP_REG_02

BKP_REG_03

BKP_REG_04

BKP_REG_05

BKP_REG_06

BKP_REG_07

BKP_REG_08

BKP_REG_09

BKP_REG_0A

BKP_REG_0B

BKP_REG_0C

BKP_REG_0D

Регистры BKP_REG_[0D…00]

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
	BKP REG[31:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…0	BKP REG[31:0]	Регистр аварийного сохранения

BKP_REG_0E

Номер	15	14	13..12	11	10…8	7	6	5…3	2…0
Доступ	R/W	U	R/W	R/W	R/W	U	R/W	R/W	R/W
Сброс	0	0	00	0	000	0	0	000	000
	ilimen	-	Trim [4:3]	FPOR	Trim[2:0]	-	Stand_ Alone	SelectRI [2:0]	LOW [2:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15	ilimen	Бит разрешения защиты ограничения регулятора по току 150 мА 1 – разрешено 0 - запрещено
14	-	Зарезервировано
13..12	Trim[4:3]	Коэффициент настройки опорного напряжения регулятора 00-1.8 В 01-1.6 В 10-1.4 В 11-1.2 В
11	FPOR	Флаг срабатывания POR Устанавливается в 1 загрузочным ПЗУ после сброса по питанию, при сбросе по питанию устанавливается в 0. Служит для анализа загрузочным ПЗУ, что сейчас идет выполнение программы после системного или программного сброса, либо после сброса по питанию.
10…8	Trim[2:0]	Коэффициент настройки опорного напряжения встроенного регулятора напряжения DUcc. С помощью Trim осуществляется подстройка напряжения DUcc 000 – DUcc + 0,10В – значение по умолчанию. 001 – DUcc + 0,06В 010 – DUcc + 0,04В 011 – DUcc + 0,01В 100 – DUcc – 0,01В 101 – DUcc – 0,04В 110 – DUcc – 0,06В 111 – DUcc – 0,10В
7	-	Зарезервировано
6	Stand_Alone	1 – выбор режима StandAlone 0 – обычный режим работы
5…3	SelectRI[2:0]	Выбор дополнительной нагрузки для регулятора 1.8В 000 – ~6 КОм (дополнительный ток потребления 300 мкА) 001 – ~270 КОм (дополнительный ток потребления 6,6 мкА) 010 – ~90 КОм (дополнительный ток потребления 20 мкА) 011 – ~24 КОм (дополнительный ток потребления 80 мкА) 100 – ~900 КОм (собственное потребление 2 мкА) 101 – ~2 КОм (дополнительный ток потребления 900 мкА) 110 – ~400 Ом (дополнительный ток потребления 4,4 мА) 111 – ~100 Ом (дополнительный ток потребления 19 мА)
2…0	LOW[2:0]	Выбор режима работы регулятора 1.8В Значение LOW должно совпадать с значением SelectRI и выставляться в зависимости от тактовой частоты микроконтроллера 000 – Частота до 10 МГц 001 – Частота до 200 КГц 010 – Частота до 500 КГц

		011 – Частота до 1 МГц 100 – При выключении всех генераторов 101 – Частота до 40 МГц 110 – Частота до 80 МГц 111 – Частота более 80 МГц

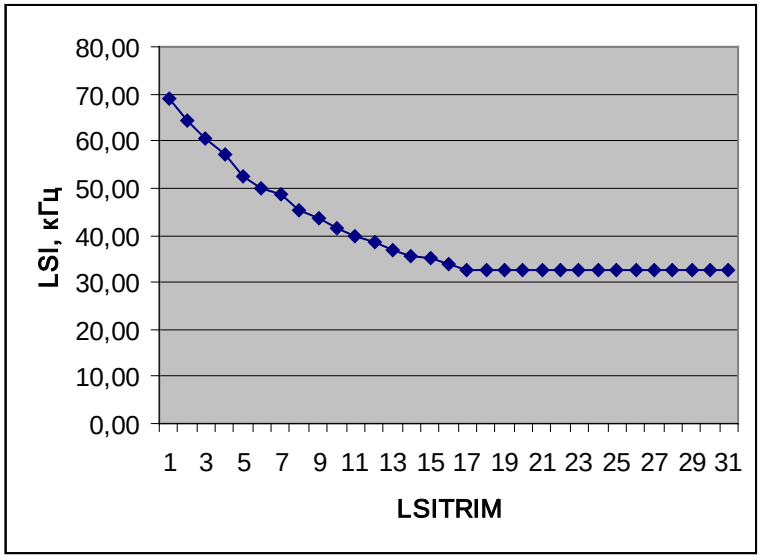
BKP_REG_0F

Номер	15	14	13	12…5	4	3..2	1	0
Доступ	R/W	U	RO	R/W	R/W	R/W	R/W	R/W
Сброс	1	0	0	0000000	0	00	0	0
	LSI ON	-	LSE RDY	CAL[7:0]	RTC EN	RTC SEL[1:0]	LSE BYP	LSE ON

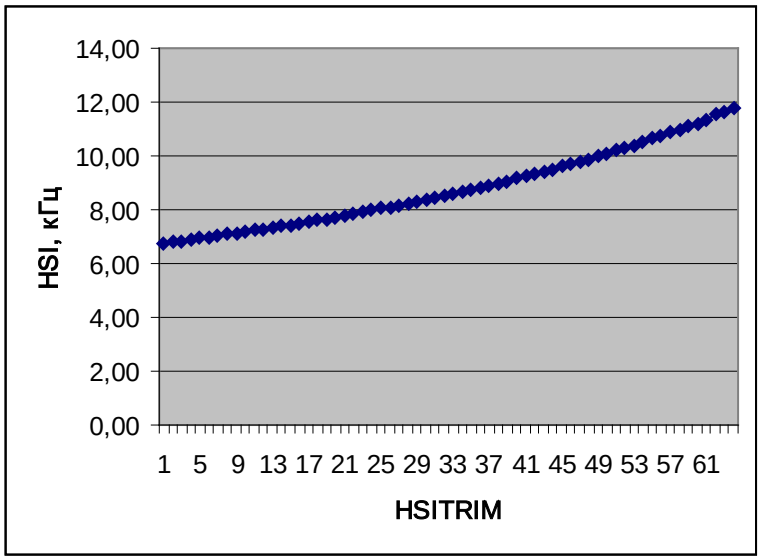
Номер	31	30	29…24	23	22	21	20…16
Доступ	R/W	R/W	R/W	R/W	R/W	RO	R/W
Сброс	0	0	0000	0	0	0	0000
	RTC RESET	STANDBY	HSI TRIM [5:0]	HSI RDY	HSI ON	LSI RDY	LSI TRIM [4:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31	RTC RESET	Сброс часов реального времени 0 – часы не сбрасываются 1 – часы сбрасываются
30	STANDBY	Режим отключения регулятора DUcc на 1.8В 0 – регулятор включен и выдает напряжение Запись 1 – выключение регулятора Триггер сбрасывается по событию ALRF или по низкому уровню на выводе WAKEUP.
29..24	HSI TRIM[5:0]	Коэффициент подстройки частоты генератора HSI Смотри диаграмму зависимости на Зависимость частоты HSI от значения HSITRIM
23	HSI RDY	Флаг выхода генератора HSI в рабочий режим 0 – генератор не запущен или не вышел в режим 1 – генератор работает в рабочем режиме
22	HSI ON	Бит управления генератором HIS 0 – генератор выключен 1 – генератор включен
21	LSI RDY	Флаг выхода генератора LSI в рабочий режим 0 – генератор не запущен или не вышел в режим 1 – генератор работает в рабочем режиме
20..16	LSI TRIM[4:0]	Коэффициент подстройки частоты генератора LSI Смотри диаграмму зависимости на Зависимость частоты LSI от значения LSITRIM
15	LSI ON	Бит управления генератором LSI 0 – генератор выключен 1 – генератор включен
14	-	Зарзервировано
13	LSE RDY	Флаг выхода генератора LSE в рабочий режим 0 – генератор не запущен или не вышел в режим 1 – генератор работает в рабочем режиме

12..5	CAL[7:0]	Коэффициент подстройки тактовой частоты часов реального времени, из каждых 2^20 тактов будет замаскировано CAL тактов. 00000000 – 0 тактов 00000001 – 1 такт …. 11111111 – 256 тактов Таким образом, при частоте 32768.00000 Гц при CAL = 0 тактов, частота = 32768.00000 Гц при CAL = 1 такт, частота = 32767,96875 Гц; … при CAL = 256 тактов, частота = 32760.00000 Гц;
4	RTC EN	Бит разрешения работы часов реального времени 0 – работа запрещена 1 – работа разрешена
3…2	RTC SEL[1:0]	Биты выбора источника тактовой синхронизации часов реального времени 00 – LSI 01 – LSE 10 – HSIRTC (формируется в блоке CLKRST) 11 – HSERTC (формируется в блоке CLKRST)
1	LSE BYP	Бит управления генератором LSE 0 – режим осциллятора 1 – режим работы на проход (внешний генератор)
0	LSE ON	Флаг выхода генератора LSI в рабочий режим 0 – генератор не запущен или не вышел в режим 1 – генератор работает в рабочем режиме



Зависимость частоты LSI от значения LSITRIM



Зависимость частоты HSI от значения HSITRIM

RTC_CNT

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

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	RTC CNT[31:0]	Значение основного счетчика часов реального времени

RTC_DIV

Номер	31						20	19…0
Доступ	U						U	R/W
Сброс	0						0	0
	-	-	-	-	-	-	-	RTC DIV [19:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	RTC DIV [19:0]	Значение счетчика предварительного делителя часов реального времени

RTC_PRL

Номер	31						20	19…0
Доступ	U						U	R/W
Сброс	0						0	0
	-	-	-	-	-	-	-	RTC PRL [19:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	RTC PRL [19:0]	Значение основания для счета счетчика предварительного делителя часов реального времени

RTC_ALRM

Номер	31							0
Доступ	R/W							R/W
Сброс	0							0
	RTC ALRM[31:0]							

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	RTC ALRM[31:0]	Значения для сравнения основного счетчика и выработки сигнала ALRF

RTC_CS

Номер	31	6	5	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0	0
	-	WEC	ALRF_IE	SECF_IE	OWF_IE	ALRF	SECF	OWF

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
30…7	-	Зарезервировано
6	WEC	Запись завершена 0 – можно записывать в регистры RTC 1 – идет запись в регистры RTC, запись в регистры запрещена.
5	ALRF_IE	Флаг разрешения прерывания по совпадению основного счетчики и регистра RTC_ALRM 0 – нет совпадения 1 – есть совпадение
4	SECF_IE	Флаг разрешения прерывания по разрешению счета основного счетчика от счетчика предварительного деления 0 – нет разрешения счета 1 – разрешение счета
3	OWF_IE	Флаг разрешения прерывания по переполнения основного счетчика RTC_CNT 0 – нет переполнения 1 – было переполнение
2	ALRF	Флаг совпадения основного счетчики и регистра RTC_ALRM 0 – нет совпадения 1 – есть совпадение
1	SECF	Флаг разрешения счета основного счетчика от счетчика предварительного деления 0 – нет разрешения счета 1 – разрешение счета
0	OWF	Флаг переполнения основного счетчика RTC_CNT 0 – нет переполнения 1 – было переполнение

Порты ввода/вывода.

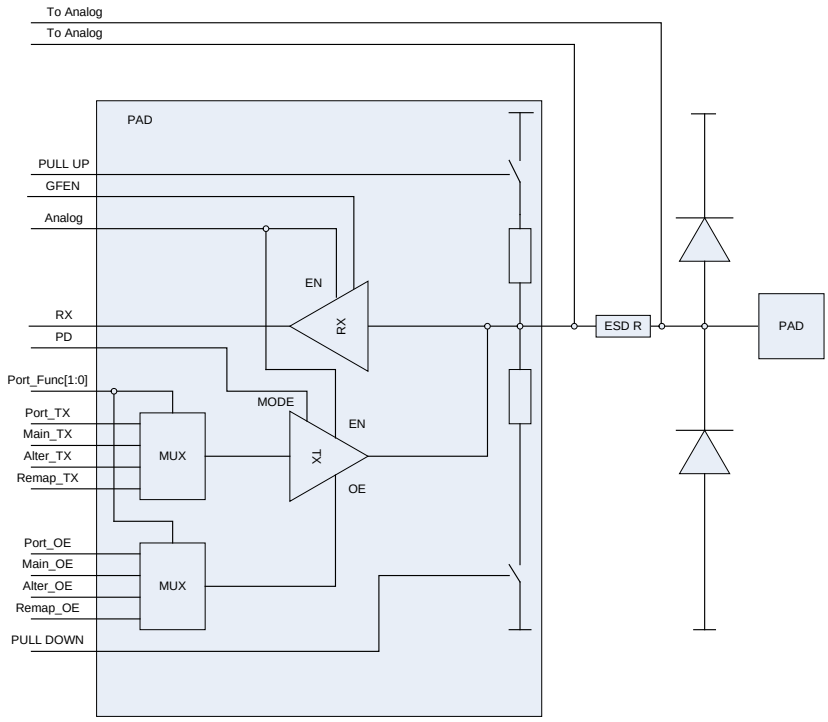
Микроконтроллер имеет 6 портов ввода/вывода. Порты 16-ти разрядные и их выводы мультиплексируются между различными функциональными блоками, управление для каждого вывода отдельное. Для того, что бы выводы порта перешли под управление того или иного периферийного блока необходимо задать для нужных выводов выполняемую функцию и настройки.

Вывод	Аналоговая функция ANALOG_EN=0		Цифровая функция Порт IO MODE[1:0]=00 ANALOG_EN=1	Основная MODE[1:0]=01 ANALOG_EN=1		Альтернативная MODE[1:0]=10 ANALOG_EN=1		Переопределенная MODE[1:0]=11 ANALOG_EN=1	
Порт A									
PA0	-		PA0	D0	1	EXTINT1	8	ETR1	3
PA1	-		PA1	D1		EXTINT2		ETR2	13
PA2	-		PA2	D2		EXTINT3		ETR3	10
PA3	-		PA3	D3		EXTINT4		BRK1	3
PA4	-		PA4	D4		BRK2	13	FRX	15
PA5	-		PA5	D5		BRK3	10	FSD	
PA6	-		PA6	D6		TMR4_CH1	16	FXEN	
PA7	-		PA7	D7		TMR4_CH1N		FTX	
PA8	-		PA8	D8		TMR4_CH2		PRMC+	4
PA9	-		PA9	D9		TMR4_CH2N		PRMC-	
PA10	-		PA10	D10		TMR4_CH3		PRMD+	
PA11	-		PA11	D11		TMR4_CH3N		PRMD-	
PA12	-		PA12	D12		TMR4_CH4		PRDC+	
PA13	-		PA13	D13		TMR4_CH4N		PRDC-	
PA14	-		PA14	D14		BRK4		PRDD+	
PA15	-		PA15	D15		ETR4		PRDD-	
Порт B									
PB0	-		PB0	D16	1	IN1+	2	TMR3_CH1	10
PB1	-		PB1	D17		IN1-		TMR3_CH1N	
PB2	-		PB2	D18		IN2+		TMR3_CH2	
PB3	-		PB3	D19		IN2-		TMR3_CH2N	
PB4	-		PB4	D20		IN3+		TMR3_CH3	
PB5	-		PB5	D21		IN3-		TMR3_CH3N	
PB6	-		PB6	D22		IN4+		TMR3_CH4	
PB7	-		PB7	D23		IN4-		TMR3_CH4N	
PB8	-		PB8	D24		IN5+		TMR1_CH1N	3
PB9	-		PB9	D25		IN5-		TMR2_CH1N	13
PB10	-		PB10	D26		IN6+		TMR1_CH2N	3
PB11	-		PB11	D27		IN6-		TMR2_CH2N	13
PB12	-		PB12	D28		IN7+		TMR1_CH3N	3
PB13	-		PB13	D29		IN7-		TMR2_CH3N	13
PB14	-		PB14	D30		IN8+		TMR1_CH4N	3
PB15	-		PB15	D31		IN8-		TMR2_CH4N	13
Порт C									
PC0	-		PC0	nWR	1	ETR1	3	BRK1	3
PC1	-		PC1	nRD		ETR2	13	BRK2	13
PC2	-		PC2	ALE		CLKO	1	BRK3	10
PC3	-		PC3	UART_TXD1	9	CLE		SIROUT0	9
PC4	-		PC4	UART_RXD1		BUSY		SIRIN0	
PC5	-		PC5	EXTINT1	8	SSP1_TXD	14	SSP1_RXD	14
PC6	-		PC6	EXTINT2		SSP1_RXD		SSP1_TXD	
PC7	-		PC7	EXTINT3		SSP1_SCK		FXEN	15
PC8	-		PC8	EXTINT4		SSP1_FSS		FTX	
PC9	-		PC9	SSP2_TXD	11	BE0	1	CAN_RX1	17
PC10	-		PC10	SSP2_RXD		BE1		CAN_TX1	
PC11	-		PC11	SSP2_SCK		BE2		CAN_RX2	18
PC12	-		PC12	SSP2_FSS		BE3		CAN_TX2	
PC13	-		PC13	PRMA+	4	A30	1	UART_TXD2	12
PC14	-		PC14	PRMA-		A31		UART_RXD2	
PC15	-		PC15	PRMB+		BUSY		TMR2_CH1	13

Порт D									
PD0	-		PD0	PRMB-	4	ALE	1	A16	1
PD1	-		PD1	PRDA+		CLE		A15	
PD2	-		PD2	PRDA-		SSP1_TXD	14	A14	
PD3	-		PD3	PRDB+		SSP1_RXD		A13	
PD4	-		PD4	PRDB-		SSP1_SCK		A7	
PD5	-		PD5	PRD_PRMA		SSP1_FSS		A6	
PD6	-		PD6	PRD_PRMB		nUART2RI	12	A5	
PD7	ADC0_REF+	5	PD7	SSP2_TXD	11	nUART2DCD		A4	
PD8	ADC1_REF-		PD8	SSP2_RXD		nUART2DTR		A3	
PD9	ADC2		PD9	SSP2_SCK		nUART2DSR		A2	
PD10	ADC3		PD10	SSP2_FSS		nUART2RTS		A1	
PD11	ADC4		PD11	A0	1	nUART2CTS		FRX	15
PD12	ADC5		PD12	SSP3_TXD	19	ETR3	10	SSP3_RXD	19
PD13	ADC6		PD13	UART_TXD2	12	OUT1+	2	SIROUT1	12
PD14	ADC7		PD14	UART_RXD2		OUT1-		SIRIN1	
PD15	DAC1_REF	6	PD15	OUT3+	2	A13	1	FSD	15
Порт E									
PE0	DAC2_REF	6	PE0	OUT4+	2	A14	1	MDC	15
PE1	DAC1_OUT		PE1	OUT3-		A15		nUART2RI	12
PE2	DAC2_OUT		PE2	OUT4-		A16		MDIO	15
PE3	-		PE3	TMR1_CH1	3	A17		TXD[0]	
PE4	-		PE4	TMR1_CH2		A18		TXD[1]	
PE5	-		PE5	TMR1_CH3		A19		TXD[2]	
PE6	OSC_IN32	7	PE6	TMR1_CH4		A20		TXD[3]	
PE7	OSC_OUT32		PE7	TMR2_CH1	13	A21		RXD[0]	
PE8	-		PE8	TMR2_CH2		A22		RXD[1]	
PE9	-		PE9	TMR2_CH3		A23		RXD[2]	
PE10	-		PE10	TMR2_CH4		A24		RXD[3]	
PE11	-		PE11	CAN_RX1	17	A25		TXEN	
PE12	-		PE12	CAN_TX1		A26		TXER	
PE13	-		PE13	CAN_RX2	18	A27		TXCLK	
PE14	-		PE14	CAN_TX2		A28		RXCLK	
PE15	-		PE15	PRD_PRMD	4	A29		RXDV	
Порт F									
PF0	OSC_IN25		PF0	PRD_PRMA	4	READY	1	RXER	15
PF1	OSC_OUT25		PF1	PRD_PRMB		A30		CRS	
PF2	-		PF2	READY/PRD_PRMC	1/4	A31		COL	
PF3	-		PF3	PRMC+	4	A0		TMR1_CH1	3
PF4	-		PF4	PRMC-		A1		TMR1_CH2	
PF5	-		PF5	PRMD+		A2		TMR1_CH3	
PF6	-		PF6	PRMD-		A3		TMR1_CH4	
PF7	-		PF7	PRDC+		A4		OUT4+	2
PF8	-		PF8	PRDC-		A5		OUT4-	
PF9	-		PF9	PRDD+		A6		OUT3+	
PF10	-		PF10	PRDD-		A7		OUT3-	
PF11	-		PF11	PRD_PRMC		A8		OUT2+	
PF12	-		PF12	PRD_PRMD		A9		OUT2-	
PF13	-		PF13	OUT2+	2	A10		SSP3_FSS	19
PF14	-		PF14	OUT2-		A11		SSP3_SCK	
PF15	-		PF15	SSP3_RXD	19	A12		SSP3_TXD	

Примечание:
1 - выводы управляются системной шиной EXT_BUS
2 - выводы управляются контроллером интерфейса по ГОСТ 18977-79
3 - выводы управляются Таймером 1
4 - выводы управляются контроллером интерфейса по ГОСТ Р52070-2003
5 - выводы используются АЦП
6 - выводы используются ЦАП
7- выводы используются генератором LSE
8 - выводы используются контроллером прерываний
9 - выводы управляются контроллером интерфейса UART1
10 - выводы управляются Таймером 3
11 - выводы управляются контроллером интерфейса SSP2

12 - выводы управляются контроллером интерфейса UART2
13 - выводы управляются Таймером 2
14 - выводы управляются контроллером интерфейса SSP1
15- выводы управляются контроллером интерфейса Ethernet 10/100
16 - выводы управляются Таймером 4
17 - выводы управляются контроллером интерфейса CAN1
18 - выводы управляются контроллером интерфейса CAN2
19 - выводы управляются контроллером интерфейса SSP3



Описание регистров портов ввода/вывода

Базовый Адрес	Название	Описание
0x400A_8000	GPIO1	Порт A
0x400B_0000	GPIO2	Порт B
0x400B_8000	GPIO3	Порт C
0x400C_0000	GPIO4	Порт D
0x400C_8000	GPIO5	Порт E
0x400E_8000	GPIO6	Порт F
Смещение		
0x00	PORT_RXTX[15:0]	Данные порта
0x04	PORT_OE[15:0]	Направление порта
0x08	PORT_FUNC[31:0]	Режим работы порта
0x0C	PORT_ANALOG[15:0]	Аналоговый режим работы порта
0x10	PORT_PULL[31:0]	Подтяжка порта
0x14	PORT_PD[31:0]	Режим работы выходного драйвера
0x18	PORT_PWR[31:0]	Режим мощности

		передатчика
0x1C	PORT_GFEN[15:0]	Режим работы входного фильтра
0x20	PORT_SETTX[15:0]	Регистр SET_TX записью 1 устанавливает 1 в регистре PORT_RXTX
0x24	PORT_CLRTX[15:0]	Регистр CLR_TX записью 1 устанавливает 0 в регистре RXTX
0x28	PORT_RDTX	Регистр позволяет читать то, что записано в выходной регистр порта

PORTx_RXTX

Номер	31					15	0
Доступ	U					R/W	R/W
Сброс	0					0	0
	-						PORT RXTX[15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15…0	PORT RXTX[15:0]	Режим работы контроллера Данные для выдачи на выводы порта и для чтения.

PORTx_OE

Номер	31					15	0
Доступ	U					R/W	R/W
Сброс	0					0	0
	-						PORT OE[15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15…0	PORT OE[15:0]	Режим работы контроллера Направление передачи данных на выводах порта 1 – выход 0 – вход

PORTx_FUNC

Номер	31	30		3	2	1	0
Доступ	R/W	R/W		R/W	R/W	R/W	R/W
Сброс	0	0		0	0	0	0

MODE15[1:0]			MODE1[1:0]	MODE0[1:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	MODEx	Аналогично MODE0 для остальных битов порта
1…0	MODE0[1:0]	Режим работы вывода порта 00 – порт 01 – основная функция 10 – альтернативная функция 11 – переопределенная функция

PORTx_ANALOG

Номер	31	16	15	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-			ANALOG EN[15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16		
15…0	ANALOG EN[15:0]	Режим работы контроллера 0 – аналоговый 1 – цифровой

PORTx_PULL

Номер	31	16	15	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
		PULL UP[15:0]		PULL DOWN[15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	PULL UP15:0]	Режим работы контроллера Разрешение подтяжки вверх 0 – подтяжка в питание выключена 1 – подтяжка в питание включена (есть подтяжка)
15…0	PULL DOWN[15:0]	Режим работы контроллера Разрешение подтяжки вниз 0 – подтяжка в ноль выключена 1 – подтяжка в ноль включена (есть подтяжка)

PORTx_PD

Номер	31	16	15	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
		PORT SHM[15:0]		PORT PD[15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	PORT SHM[15:0]	Режим работы контроллера Режим работы входа 0 – триггер Шмита выключен гистерезис 200 мВ. 1 – триггер Шмита включен гистерезис 400 мВ.
15…0	PORT PD[15:0]	Режим работы контроллера Режим работы выхода 0 – управляемый драйвер 1 – открытый сток

PORTx_PWR

Номер	31	30			3	2	1	0
Доступ	R/W	R/W			R/W	R/W	R/W	R/W
Сброс	0	0			0	0	0	0
	PWR15[1:0]				PWR1[1:0]		PWR0[1:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	PWRx	Аналогично PWR0 для остальных битов порта
1…0	PWR0[1:0]	Режим работы вывода порта 00 – зарезервировано 01 – медленный фронт 10 – быстрый фронт 11 – максимально быстрый фронт

PORTx_GFEN

Номер	31	16	15	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
		-		GFEN[15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16		
15…0	GFEN[15:0]	Режим работы входного фильтра 0 – фильтр выключен

		1 – фильтр включен

Детектор напряжения питания

Блок детектора напряжения питания PVD предназначен для контроля питания Ucc и BUcc при работе микроконтроллера. Блок PVD позволяет сравнивать внешние уровни напряжения с внутренними опорными уровнями и в случае превышения или снижение ниже опорного уровня выработать сигнал или прерывание для программной обработки.
Уровень опорного напряжения для сравнения с Ucc задается битами PLS[2:0] в регистре PVDCS, для сравнения с BUcc задается битами PLBS[1:0] в регистре PVDCS. В соответствии с уровнями напряжения формируются флаги PVD и PBVD. Данные флаги выставляются при возникновении события и сбрасываются программно.

Параметр	Не менее	Типовое	Не более
Входное напряжение, Ucc, В	2,0	-	3,6
Входное напряжение, BUcc, В	1,8	-	3,6
Уровень срабатывания PVD от Ucc, при PLS = “000”, В		2,0	
Уровень срабатывания PVD от Ucc, при PLS = “001”, В		2,2	
Уровень срабатывания PVD от Ucc, при PLS = “010”, В		2,4	
Уровень срабатывания PVD от Ucc, при PLS = “011”, В		2,6	
Уровень срабатывания PVD от Ucc, при PLS = “100”, В		2,8	
Уровень срабатывания PVD от Ucc, при PLS = “101”, В		3,0	
Уровень срабатывания PVD от Ucc, при PLS = “110”, В		3,2	
Уровень срабатывания PVD от Ucc, при PLS = “111”, В		3,4	
Уровень срабатывания PBVD от BUcc, при PBLS = “00”, В		1,8	
Уровень срабатывания PBVD от BUcc, при PBLS = “01”, В		2,2	
Уровень срабатывания PBVD от BUcc, при PBLS = “10”, В		2,6	
Уровень срабатывания PBVD от BUcc, при PBLS = “11”, В		3,0	

Описание регистров блока PVD

Базовый Адрес	Название	Описание
0x4005_8000	POWER	Датчик подсистемы питания
Смещение		
0x00	PVDCS [12:0]	Регистр управления и состояния датчика питания

PVDCS

Номер	9	8	7	6	5…3	2…1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	000	00	0
	IEPVD	IEPVBD	PVD	PVBD	PLS [2:0]	PBLS [1:0]	PVD EN

Номер	31				12	11	10
Доступ	U				R/W	R/W	R/W
Сброс	0				0	0	0
	-	-	-	-	PVDBEN	INV	INVB

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
12	PVDBEN	Бит разрешения работы блока датчика напряжения питания BUcc 0 – датчик отключен 1 – датчик включен
11	INV	Флаг инверсии выхода от датчика PVD 0 – нет инверсии 1 – есть инверсия Если флаг не инвертируется, то флаг выставляется при превышении заданного уровня, если инвертируется, то при снижении ниже заданного уровня
10	INVB	Флаг инверсии выхода от датчика PVBD 0 – нет инверсии 1 – есть инверсия Если флаг не инвертируется, то флаг выставляется при превышении заданного уровня, если инвертируется, то при снижении ниже заданного уровня
9	IEPVD	Флаг разрешения прерывания от датчика PVD 0 – прерывание запрещено 1 – прерывание разрешено Очищается записью 0, если при очистке, датчик продолжает выдавать сигнал, то флаг не будет очищен.
8	IEPVBD	Флаг разрешения прерывания от датчика PVBD 0 – прерывание запрещено 1 – прерывание разрешено Очищается записью 0, если при очистке, датчик продолжает выдавать сигнал, то флаг не будет очищен.
7	PVD	Результат сравнения напряжения основного питания 0 – напряжение питания меньше чем уровень задаваемый PLS 1 – напряжение питания больше чем уровень задаваемый PLS

6	PVBD	Результат сравнения напряжения батарейного питания 0 – напряжение питания меньше чем уровень задаваемый PBLS 1 – напряжение питания больше чем уровень задаваемый PBLS
5…3	PLS[2:0]	Уровень напряжения для сравнения с напряжением основного питания 000 – 2,0В 001 – 2,2В 010 – 2,4В 011 – 2,6В 100 – 2,8В 101 – 3,0В 110 – 3,2В 111 – 3,4В
2…1	PBLS[1:0]	Уровень напряжения для сравнения с напряжением батарейного питания 00 – 1,8В 01 – 2,2В 10 – 2,6В 11 – 3,0В
0	PVDEN	Бит разрешения работы блока датчика напряжения питания Ucc 0 – датчик отключен 1 – датчик включен

Внешняя системная шина

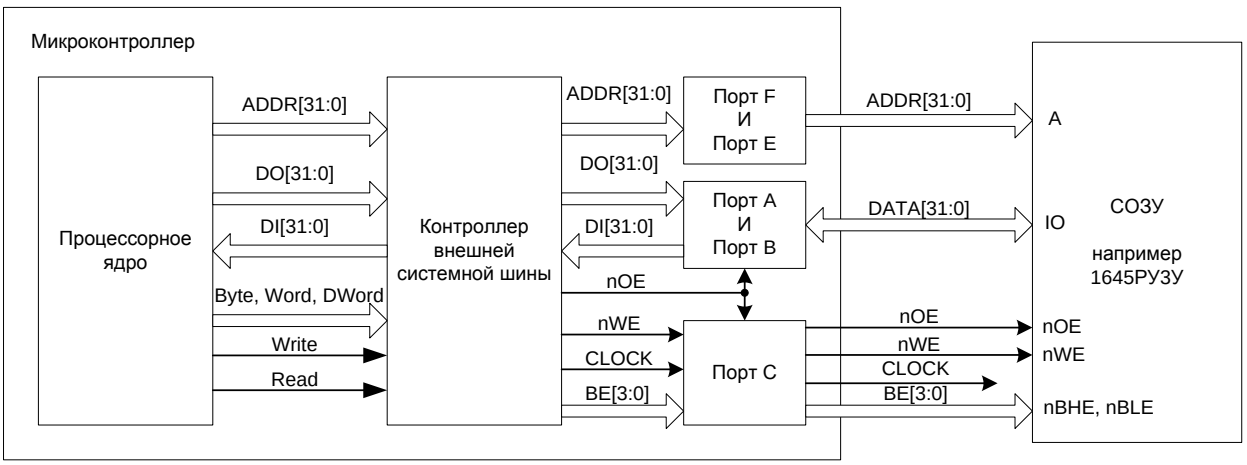
Внешняя системная шина позволяет работать с внешними микросхемами памяти и периферийными устройствами. Области адресного пространства микроконтроллера отведены для работы с внешней системной шиной.

Адресный диапазон	Размер	Описание
0x0010_0000 – 0x1FFF_FFFF ITCMLAEN=1 0x0000_0000 – 0x1FFF_FFFF ITCMLAEN=0	256 Мбайт	Область памяти секции Code отображаемая на внешнюю системную шину с доступом через AHB-Lite шину
0x5000_0000 – 0xDFFF_FFFF	2 Гбайт	Область памяти секции Peripheral и External SRAM отображаемая на внешнюю системную шину с доступом через шину AHB-Lite . К этой области имеет доступ DMA контроллер

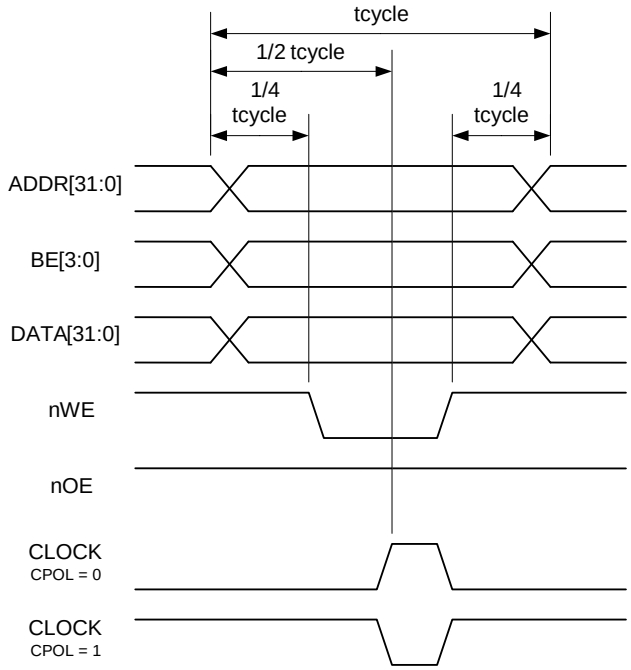
Контроллер внешней системной шины во всех режимах не формирует сигналов выборки чипа CE. При работе с внешними статическими ОЗУ, ПЗУ и периферийными устройствами в качестве сигнала выборки чипа можно использовать старшие линии шины адреса, не используемые для непосредственной адресации, либо использовать программно управляемые выводы портов для формирования сигналов CE.

Работа с внешними статическими ОЗУ, ПЗУ и периферийными устройствами

Для работы контроллера внешней системной шины с внешними микросхемами статического ОЗУ, ПЗУ или внешними периферийными устройствами необходимо задать режим работы через регистр EXT_BUS_CONROL. Бит RAM разрешает работу с внешними ОЗУ, бит ROM разрешает только чтение внешних ОЗУ или ПЗУ. В зависимость от скорости работы ядра микроконтроллера и внешних устройств необходимо задать времена транзакции на внешней системной шине через биты WAIT_STATE[3:0]. После этого все обращения в область памяти отображаемой на внешнюю системную шину будут транслироваться на выводы внешней системной шины ADDR, DATA и сигналы управления OE, WE, BE[3:0] и сигнал синхронизации CLOCK.

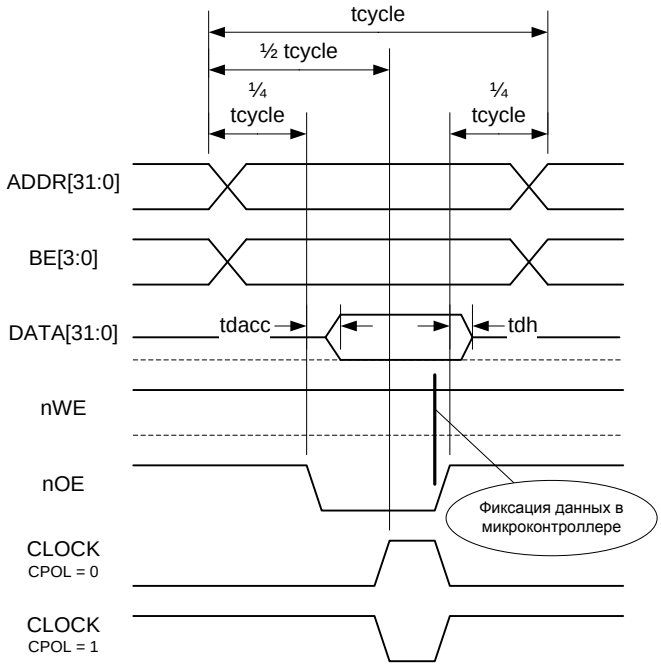


Обмен по внешней системной шине
Диаграмма записи представлена на рисунке



Время цикла записи tcycle задается битами WAIT_STATE[3:0]. Активный уровень сигналов nWE, nOE, BE[3:0] низкий. Если сигнал CLOCK не требуется, он может не использоваться.

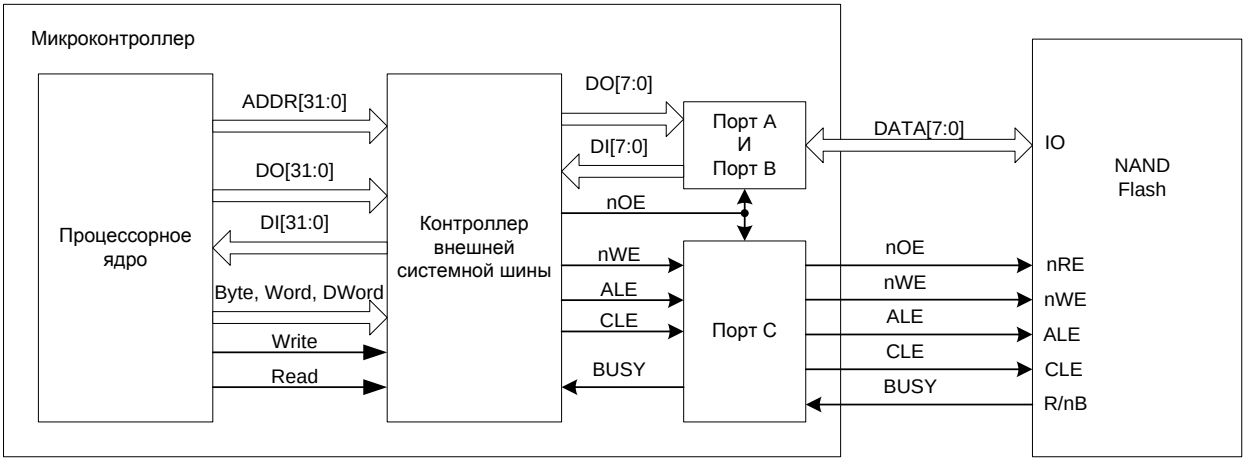
Диаграмма чтения представлена на рисунке



При чтении по внешней системной шине необходимо выбрать такую длительность времени tcycle, что бы выполнялось время скорости доступа к памяти. Время tdh для микроконтроллера равно нулю.

Работа с внешней NAND Flash памятью.

Для работы контроллера внешней системной шины с внешними NAND Flash микросхемами памяти необходимо задать режим работы через регистр EXT_BUS_CONROL. Бит NAND разрешает работу с внешними NAND Flash микросхемами. В зависимость от скорости работы ядра микроконтроллера и внешних устройств необходимо задать времена выполнения различных этапов работы NAND Flash памяти через регистр NAND_CYCLES. После этого обращения в область памяти отображаемой на внешнюю системную шину будут перекодироваться в командные, адресные и обмена данными циклы обращения с NAND Flash через выводы внешней системной шины DATA[7:0], ALE, CLE, BUSY1 и BUSY2.



Контроллер имеет сигнал BUSY для подключения соответствующего вывода NAND Flash. Сигналы BUSY от различных NAND Flash объединяются по логическому И на выводе контроллера и формируют общий сигнал BUSY. При работе с NAND Flash памятью тип выполняемой операции кодируется адресом обращения, а данные и адрес передаются данными при записи и чтении памяти. Формат кодирования адреса обращения представлен в таблице

Адрес обращения	Фаза команды	Фаза данных
ADDR[31:24]	Не имеет значения, но должно попадать в адресные диапазоны внешней системной шины 0x10…0x1F 0x30…0x3F 0x50…0xCF	
ADDR[23:21]	ADR_CYCLES[2:0] 000 – 0 циклов 001 – 1 цикл … 111 – 7 циклов	Не имеет значения
ADDR[20]	Выполнение завершающей команды 0 – не выполнять 1 – выполнять	
ADDR[19]	Всегда 0	Всегда 1
ADDR[18:11]	Код завершающей команды	

	ECMD[7:0] 0x10/0x11 Page Program 0xD0 Block Erase	
ADDR[10:3]	Код начальной команды SCMD[7:0] 0x00/0x01 – Read1 0x50 – Read2 0x90 – Read ID 0xFF – Reset 0x80 – Page Program 0x60 – Block Erase 0x70 – Read Status	Не имеет значения
ADDR[2:0]	Не имеет значения	

Более подробная информация о командах NAND Flash памяти представлена в документации на этот тип микросхем

Пример работы с NAND Flash памятью.

```
// =====================================================================
// Инициализация контроллера внешней системной шины для работы с NAND Flash
// =====================================================================

NAND_CYCLES = 0x02A63466;
// время t_rr = 2 цикла HCLK или 20 нс при частоте HCLK 100 МГц
// время t_alea = 10 циклов
// время t_whr = 6 циклов
// время t_wp = 3 цикла
// время t_rea = 4 цикла
// время t_wc = 6 циклов
// время t_rc = 6 циклов

EXT_BUS_CONTROL = 0x00000004;
// NAND = 1;

// =====================================================================
// Чтение ID микросхемы
// =====================================================================

unsigned char IDH;
unsigned char IDL;

// Фаза команды
*((volatile unsigned char *) (0x77200480)) = 0x00;
// ADR_CYCLE = 1
// SCMD = 0x90 (READ)
// Address 1 cycle = 0x00

// Фаза данных
IDL = *((volatile unsigned char *)(0x77080000));
IDH = *((volatile unsigned char *)(0x77080000));


// =====================================================================
// Стирание блока памяти
// =====================================================================
```

```
// Фаза команды
*((volatile unsigned char *)(0x70768300))=0x11;
*((volatile unsigned char *)(0x70768301))=0x22;
*((volatile unsigned char *)(0x70768302))=0x33;
// ADR_CYCLE = 3
// выполнять завершающую команду
// ECMD= 0xD0
// SCMD = 0x60
// Address 1 cycle = 0x11
// Address 2 cycle = 0x22
// Address 1 cycle = 0x33
while (EXT_BUS_CONTROL!=0x080 ) {};
// Ждем R/nB

// Фаза команды
*((volatile unsigned char *)(0x70000380+addon))=0x00;
// ADR_CYCLE = 0
// SCMD = 0x70
// Фаза данных
IDL = *((volatile unsigned char *)(0x77080000));
If (IDL & 0x01==0x01) Error ();
// Если бит IO0==1 то стирание не выполнено

// ========================================================================
// Запись страницы
// ========================================================================

// Фаза команды
*((volatile unsigned char *)(0x70800400))=0x11;
*((volatile unsigned char *)(0x70800400))=0x22;
*((volatile unsigned char *)(0x70800400))=0x33;
*((volatile unsigned char *)(0x70800400))=0x44;
// ADR_CYCLE = 4
// SCMD = 0x80

// Фаза данных
*((volatile unsigned char *)(0x70088000+addon))=0xBB;
*((volatile unsigned char *)(0x70088000+addon))=0xCC;
*((volatile unsigned char *)(0x70088000+addon))=0xDD;
// не выполнять завершающую команду
// ECMD= 0x10
…
*((volatile unsigned char *)(0x70188000+addon))=0xEE;
// не выполнять завершающую команду
// ECMD= 0x10
// Дынные 0 – 0xBB, 1 – 0xCC,… N – 0xEE
// N от 1 до 528
while (EXT_BUS_CONTROL!=0x080 ) {};
// Ждем R/nB

// Фаза команды
*((volatile unsigned char *)(0x70000380+addon))=0x00;
// ADR_CYCLE = 0
// SCMD = 0x70
// Фаза данных
IDL = *((volatile unsigned char *)(0x77080000));
If (IDL & 0x01==0x01) Error ();
// Если бит IO0==1 то запись не выполнена
```

```
// ===================================================================
// Чтение страницы
// ===================================================================

// Фаза команды
*((volatile unsigned char *)(0x70800000))=0x11;
*((volatile unsigned char *)(0x70800000))=0x22;
*((volatile unsigned char *)(0x70800000))=0x33;
*((volatile unsigned char *)(0x70800000))=0x44;
// ADR_CYCLE = 4
// SCMD = 0x00
while (EXT_BUS_CONTROL!=0x080 ) {};
// Ждем R/nB

// Фаза данных
IDL=*((volatile unsigned char *)(0x70080000));
IDH=*((volatile unsigned char *)(0x70080000));
If (IDL != 0xBB || IDH != 0xCC)  Error ();
// Если считали не то что записали то ощибка
```

Описание регистров блока контроллера внешней системной шины

Базовый Адрес	Название	Описание
0x400F_0000	EXT_BUS	Датчик подсистемы питания
Смещение		
0x50	NAND_CYCLES	Регистр управления работой с NAND_Flash
0x54	EXT_BUS_CONTROL	Регистр управления внешней системной шиной
0x58	RAM_Cycles1	Регистр индивидуальной настройки параметров обмена с RAM для адресного пространства 0x10000000-0x1FFFFFFF с ревизии 2
0x5C	RAM_Cycles2	Регистр индивидуальной настройки параметров обмена с RAM для адресного пространства 0x50000000-0x5FFFFFFF с ревизии 2
0x60	RAM_Cycles3	Регистр индивидуальной настройки параметров обмена с RAM для адресного пространства 0x60000000-0x6FFFFFFF с ревизии 2
0x64	RAM_Cycles4	Регистр индивидуальной настройки параметров обмена с RAM для адресного пространства 0x70000000-0xDFFFFFFF с ревизии 2

EXT_BUS_CONTROL

Номер	7	6	5	4	3	2	1	0
Доступ	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	1	0	0	0		0	0	1
	BUSY	LOW16	LOW8	ENDIAN	CPOL	NAND	RAM	ROM

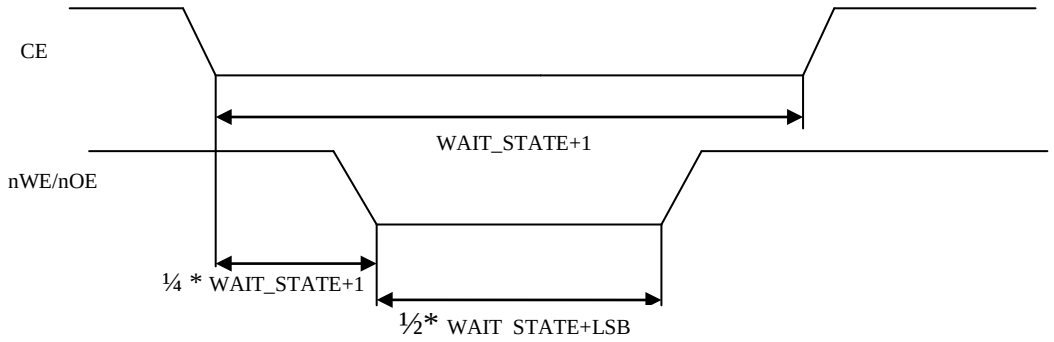
Номер	15	14	13	12	11	10	9	8
Доступ	R/W	R/W	R/W	R/W	U	U	U	U
Сброс	1	1	1	1				
	WAIT_STATE[3:0]				-			

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
15..12	WAIT STATE[3:0]	Количество тактов шины AHB, необходимых для стандартного цикла записи/чтения. Сигналы OE/WE устанавливаются в момент времени ¼ WAIT_STATE, снимаются в момент времени ¾ WAIT_STATE.
11..8	-	Зарезервировано
7	BUSY	Сигнал занятости NAND Flash памяти. 1 – операция завершена 0 – операция не завершена
6	LOW16	Выравнивание данных по 16 младшим разрядам внешней системной шины с ревизии 2 1- данные записываются и читаются всегда с 16 младших разрядов данных 0- обычный режим работы шины С ревизии 3 запись и чтение 32 разрядных данных при LOW16=1 происходит автоматически за два обращения на внешнюю шину.
5	LOW8	Выравнивание данных по 8 младшим разрядам внешней системной шины с ревизии 2 1- данные записываются и читаются всегда с 8 младших разрядов данных 0- обычный режим работы шины С ревизии 3 запись и чтение 32 разрядных данных при LOW8=1 происходит автоматически за четыре обращения на внешнюю шину.
4	ENDIAN	Всегда записывать ноль
3	CPOL	Бит задания полярности сигнала CLOCK 0 – положительная полярность 1 – отрицательная полярность
2	NAND	Бит глобального разрешения памяти NAND 1 – выбрана NAND 0 – память NAND не выбрана Одновременная установка нескольких бит 3..0 недопустима, в этом случае запрещается работа со всей памятью

1	RAM	Бит глобального разрешения памяти RAM 1 – выбрана RAM 0 – память RAM не выбрана
0	ROM	Бит глобального разрешения памяти ROM 1 – выбрана ROM 0 – память ROM не выбрана

Длительность фаз обращения в тактах процессора

WAIT_STATE	Предустановка Адреса и данных перед сигналом WE или OE	Длительность WE или OE	Удержание Адреса и данных после сигнала WE или OE
0	1	1	0
1	1	1	1
2	1	1	1
3	2	1	1
4	2	2	1
5	3	2	1
6	3	2	2
7	4	2	2
8	4	3	2
9	5	3	2
10	5	3	3
11	6	3	3
12	6	4	3
13	7	4	3
14	7	4	4
15	8	4	4



NAND_CYCLES

Номер	31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Срос		0	0	0	0	0	0	0
	-	t_rr	t_alea	t_whr	t_wp	t_rea	t_wc	t_rc

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28		Зарезервировано
27..24	t_rr[3:0]	время от снятия busy до операции чтения 0000 – 0 HCLK циклов 0001 – 1 HCLK цикл …. 1111 – 15 HCLK циклов Типовое значение для памяти NAND Flash составляет 20 нс
23..20	t_alea[3:0]	время доступа к регистрам ID аналогично t_rr Типовое значение для памяти NAND Flash составляет 100 нс
19..16	t_whr[3:0]	время доступа к регистру статуса аналогично t_rr Типовое значение для памяти NAND Flash составляет 60 нс
15..12	t_wp[3:0]	время доступа по записи аналогично t_rr Типовое значение для памяти NAND Flash составляет 25 нс
11..8	t_rea[3:0]	время доступа по чтению аналогично t_rr Типовое значение для памяти NAND Flash составляет 35 нс
7..4	t_wc[3:0]	время цикла записи аналогично t_rr Типовое значение для памяти NAND Flash составляет 60 нс
3..0	t_rc[3:0]	время цикла чтения аналогично t_rr Типовое значение для памяти NAND Flash составляет 60 нс

RAM_CYCLESx

Номер	15	14	13-11	10-8	7-1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W
Срос		0	0	0	0	0
	-	USE_READY	WS_HOLD	WS_SETUP	WS_ACTIVE	ENABLE_TUNE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..15		Зарезервировано
14	USE_READY	Разрешение опроса внешнего сигнала READY на выводе PF[2], динамически определяющего аппаратные состояния ожидания в цикле обмена по внешней системной шине. 1 – опрашивается 0 – не опрашивается Опрос сигнала READY производится на последнем такте фазы ACTIVE, если READY находится в активном состоянии – осуществляется переход к фазе HOLD и завершене цикла обмена, в противном случае производится повторение опроса на каждом последующем такте пока количество тактов ожидания не превысит 256. После этого обмен завершается.
13..11	WS_HOLD	Время удержания сигналов nWE/nOE, выраженное в количестве тактов системной частоты в диапазоне от 1 до 8 плюс один такт
10..8	WS_SETUP	Время предустановки сигналов nWE/nOE в цикле записи/чтения, выраженное в количестве тактов системной частоты в диапазоне от 1 до 8 плюс один такт
7..1	WS_ACTIVE	Длительность низкого уровня сигналов nWE/nOE в цикле записи/чтения, выраженное в количестве тактов системной частоты в диапазоне от 1 до 128 плюс один такт
0	ENABLE_TUNE	Разрешение настройки параметров обмена соответствующего диапазона адресов 1- разрешена 0 – запрещена

Режим Stand Alone

Данный режим предназначен для прямого доступа к контроллерам интерфейса Ethernet и интерфейса по ГОСТ Р52070-2003. При этом ядро и все остальные блоки находятся в состоянии сброса за исключением генератора тактовой частоты, который обеспечивает тактирование контроллеров. Для увеличения частоты работы контроллеров можно использовать PLL, предварительно настроив коэффициент умножения и включив блок. Для осуществления перехода в этот режим можно воспользоваться функциями загрузочного ПЗУ, описанного выше, но только для режима Stand Alone 2, установив при этом внешний осциллятор 8 МГц на входы OSC_IN и OSC_OUT. Для режимов Stand Alone 1 и Stand Alone 3 необходимо использовать режим загрузочного ПЗУ микроконтроллер с режимом отладки и провести инициализацию этих режимов программно, как описано ниже.

Режим Stand Alone 1 (доступ только к контроллеру Ethernet):

```
RST_CLK->PLL_CONTROL=0x304;
RST_CLK->HS_CONTROL= 3;
RST_CLK->CPU_CLOCK=0x00000107;
BKP->REG_0E &= ~(1<<7);
BKP->REG_0E |= 1<<6;
RST_CLK->ETH_CLOCK=0x19000000;
```

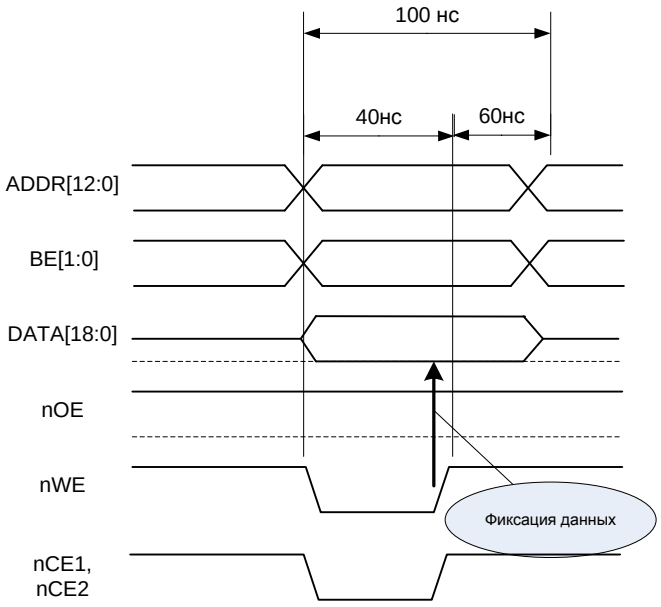
Режим Stand Alone 2 (доступ только к контроллеру по ГОСТ Р52070-2003):

```
RST_CLK->PLL_CONTROL=0x904;
RST_CLK->HS_CONTROL=1 или 3; (3 – генератор; 1 – осциллятор)
RST_CLK->CPU_CLOCK=0x00000106;
RST_CLK->PER_CLOCK|=1<<4 | 1<<9| 1<<10 | 1<<27;
BKP->REG_0E &= ~(1<<7);
BKP->REG_0E |= 1<<6;
RST_CLK->ETH_CLOCK=0x02000000;
```

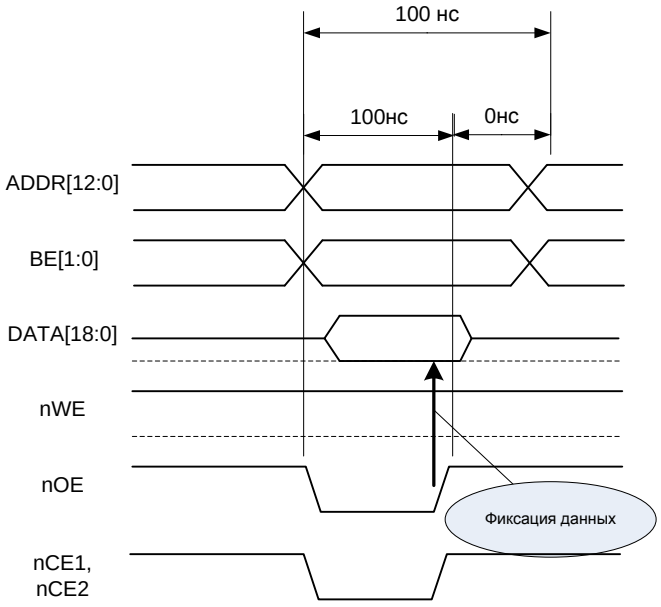
Режим Stand Alone 3 (доступ к контроллерам ГОСТ Р52070-2003 и Ethernet):

```
RST_CLK->PLL_CONTROL=0x304;
RST_CLK->HS_CONTROL=3;
RST_CLK->CPU_CLOCK=0x00000107;
RST_CLK->PER_CLOCK|=1<<4 | 1<<9| 1<<10 | 1<<27;
BKP->REG_0E &= ~(1<<7);
BKP->REG_0E |= 1<<6;
RST_CLK->ETH_CLOCK=0x1B000000;
```

В режимах Stand Alone 1 и 3 внешняя частота на HSE блок должна быть 25 МГц и подаваться с внешнего генератора. В режиме Stand Alone 2 внешняя частота на HSE блок должна быть кратна 8 МГц и подаваться либо с генератора, либо с осциллятора. После перехода в режим Stand Alone обмен данными с контроллером осуществляется посредством параллельного асинхронного интерфейса. Временная диаграмма циклов работы асинхронного интерфейса приведена на рисунке. Для приведённых временных диаграмм тактовая частота контроллера составляет 50 МГц. Если частоту увеличить в два раза, то все временные параметры на диаграмме можно уменьшить в два раза. Дальнейшее увеличение частоты не даст прироста пропускной способности, так как для параллельного асинхронного интерфейса она ограничена частотой 50 МГц.



Временная диаграмма цикла записи



Временная диаграмма цикла чтения

Контроллер интерфейса USB.

Контролер USB реализует функции контроллера функционального устройства (Device) и управляющего устройства (Host) в соответствии со спецификацией USB 2.0.

Контроллер USB поддерживает: Full Speed (12 Мбит/с) и Low Speed (1.5 Мбит/с) режимы работы, контроль ошибок с помощью циклического избыточного кода (CRC), NRZI код приема/передачи, управляющие (Control), сплошные (Bulk), изохронные (Isochronous) передачи и передача по прерываниям (Interrupt). Также поддерживается конфигурирование USB Device от 1-й до 4-х оконечных точек, автоматическая отправка SOF пакетов, вычисление оставшегося во фрейме времени. Каждая оконечная точка USB Device имеет собственный буфер FIFO размером 64 байта. USB Host поддерживает до 16 оконечных точек. USB Host имеет буфер FIFO размером 64 байта.

Инициализация контроллера при включении

При включении питания в первую очередь должны быть заданы параметры тактового сигнала блока USB. Параметры задаются в блоке «Сигналы сброса и тактовой частоты». Источником тактового сигнала для блока USB может быть либо встроенный высокочастотный генератор (HSI) или внешний осциллятор (HSE). Блок USB функционирует на частоте 48 МГц. Требуемая частота может быть получена умножением частоты одного из двух генераторов до требуемого значения. Умножение выполняется встроенным блоком PLLUSB.
Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 16, задаваемый в поле PLLUSBMUL регистра PLL_CONTROL. Входная частота блока умножителя должна быть в диапазоне 2…16МГц выходная должна составлять 48 МГц. При выходе блока умножителя тактовой частоты в расчетный режим вырабатывается сигнал PLLRDY. Блок включается с помощью сигнала PLLUSBON. Выходная частота используется как основная частота протокольной части USB интерфейса.
Для задания тактовой частоты блока необходимо соблюдать следующий порядок работы. Установить бит разрешения тактирования блока (бит 2 регистра PER_CLOCK). В регистре USB_CLOCK установить бит USBCLKEN, задать источник тактового сигнала в полях USBC1SEL и USBC2SEL. Установить бит PLLUSBON и задать коэффициент умножения в поле PLLUSBMUL регистра PLL_CONTROL, если используется USBPLL.
После подачи тактового сигнала на блок USB необходимо выполнить сброс контроллера. Сброс выполняется установкой бита RESET_CORE в регистре USB_HSCR. Сигнал сброса необходимо удерживать как минимум 10 циклов тактовой частоты. После этого могут быть заданы параметры шины USB (скорость, полярность, наличие подтяжек).

Задание параметров шины USB и события подключения/отключения

Контроллер USB может быть сконфигурирован как USB Host или как USB Device. Конфигурация задается битом CORE_MODE в регистре USB_HSCR (0 – режим Device, 1

– режим Host). Прием/передача через физический интерфейс USB разрешается установкой бит EN_RX и EN_TX в этом же регистре. В режиме приема имеется возможность отключить передатчик в целях экономии потребления (EN_TX=0). Отключение всего блока в целом осуществляется при EN_RX=0.
В режиме Device параметры шины задаются в регистре USB_SC. Скорость задается битом SCFSR (0 – 1.5 Мбит/с, 1 – 12 Мбит/с), полярность битом SCFSP (0 – Low speed, 1 – Full speed) этого регистра.
В режиме Host параметры шины задаются в регистре USB_TXLC. Скорость задается битом FSLR (0 - 1.5 Мбит/с, 1 – 12 Мбит/с), полярность битом FSPL (0 – Low speed, 1 – Full speed) этого регистра.
В режиме Host контроллер автоматически определяет подключение или отключение устройства к шине. Бит CONEV регистра USB_HIS устанавливается в 1 при возникновении одного из событий.

Задание адреса и инициализация оконечных точек

Функциональный адрес устройства USB задается в регистре USB_SA.
Для инициализации конечной точки в первую очередь необходимо установить бит глобального разрешения всех оконечных точек (SCGEN = 1 в регистре USB_SC). Биты EPEN в регистрах USB_SEPx.CTRL должны быть установлены, чтобы разрешить соответствующую оконечную точку. Если предполагается использовать изохронной тип передачи оконечной точки, то необходимо установить бит EPISOEN в соответствующем регистре USB_SEPx.CTRL.

Транзакция IN (Usb Device)

Если на шине появляется IN пакет и адрес совпадает с заданным в регистре USB_SA, бит SCTDONE регистра USB_SIS устанавливается в 1.
Если оконечная точка не готова (бит EPRDY = 0 в регистре USB_SEPx.CTRL), то контроллер отправляет NAK пакет (рис 1а). Бит NAKSENT регистра USB_SEPx.STS устанавливается в 1.
Если оконечная точка готова и установлен бит EPSSTALL в регистре USB_SEPx.CTRL, то контроллер отправляет STALL пакет (рис 1б). Бит SCSTALLSENT регистра USB_SEPx.STS устанавливается в 1.
Если оконечная точка готова (рис 1в), биты SCTTYPE[1:0] в регистре USB_SEPx.TS устанавливаются в значение 1 для конечной точки с номером, содержащимся в поле пакета. Контроллер может передавать пакет данных. Пакет данных формируется записью в регистр USB_EPx.TXFD побайтно в FIFO оконечной точки. Запись 1 в USB_EPx.TXFC сбрасывает указатель FIFO передачи в 0. Максимальный размер передаваемого пакета составляет 64 байт. Попытка записи более 64 байт подряд приведет к переполнению FIFO. Перед началом формирования очередного пакета необходимо выполнять сброс указателя FIFO.
Если в ответ на переданные данные хост отправляет ACK пакет, то бит SCACKRXED в регистре USB_SEPx.STS устанавливается в 1. Для отправки следующего пакета необходимо инвертировать бит EPDATASEQ в регистре USB_SEPx.CTRL, чтобы соблюдалась очередность оправки пакетов DATA0, DATA1.
После окончания транзакции бит SCTDONE регистра USB_SIS должен быть очищен записью 1.

Транзакция SETUP/OUT (Usb Device)

Если на шине появляется SETUP/OUT пакет, и адрес совпадает с заданным в регистре USB_SA, и оконечная точка готова (бит EP_READY = 1 в регистре ENDPOINTx_CONTROL), то бит SCTDONE регистра USB_SIS устанавливается в 1.
Если оконечная точка не готова (бит EPRDY = 0 в регистре USB_SEPx.CTRL), то контроллер отправляет NAK пакет (рис. 2а). Бит NAKSENT регистра USB_SEPx.ST устанавливается в 1.
Если оконечная точка готова и установлен бит EPSSTALL в регистре USB_SEPx.CTRL, то контроллер отправляет STALL пакет (рис. 2б). Бит SCSTALLSENT регистра USB_SEPx.STS устанавливается в 1.
Если оконечная точка готова (рис. 2в) и на шине был пакет SETUP, то биты SCTTYPE[1:0] в регистре USB_SEPx.TS устанавливаются в значение 00 для конечной точки с номером, содержащимся в поле пакета. Если пакет OUT, то значение SCTTYPE[1:0] = 2.
Когда на шине появляется DATA0/DATA1 пакет, то данные начинают записываться побайтно в FIFO приема соответствующей оконечной точки. После записи каждого байта увеличивается на единицу счетчик принятых байт. Принятые байты считываются через регистр USB_EPx.RXFD. Количество принятых байт содержится в регистре USB_EPx.RXFDC. После приема очередного пакета необходимо выполнять сброс указателя FIFO приема записью 1 в регистр USB_EPx.RXFC.
После окончания транзакции бит SCTDONE регистра USB_SIS должен быть очищен записью 1.

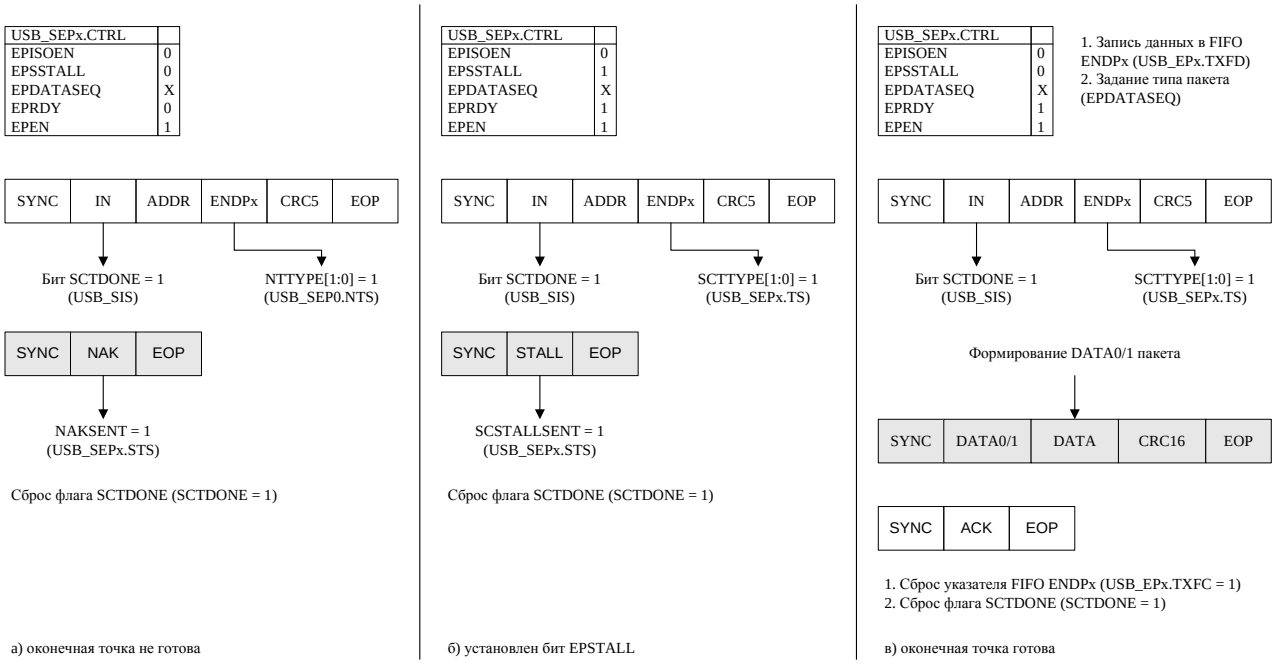


Рисунок 1

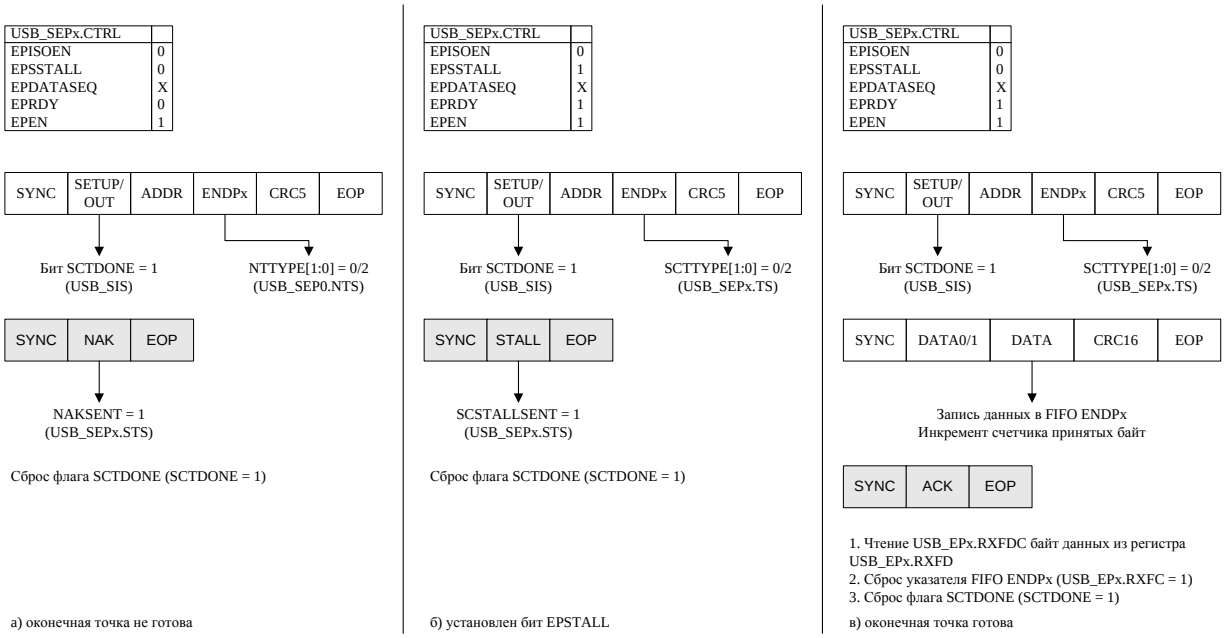


Рисунок 2

Транзакция SETUP/OUT (Usb Host)

Для начала транзакции должны быть заданы адрес устройства (регистр USB_HTXA), оконечная точка (регистр USB_HTXE) и тип token пакета (регистр USB_HTXT). Данные записываются побайтно в регистр USB_HTXFD. Максимальный размер передаваемого пакета составляет 64 байт. Попытка записи более 64 байт подряд приведет к переполнению FIFO. Запись 1 в USB_HTXFDC сбрасывает указатель FIFO передачи в 0. Перед началом формирования очередного пакета необходимо выполнять сброс указателя FIFO. Транзакция запускается при установке бита TREQ регистра USB_HTXC. Host отправляет пакет Setup/Out и пакет данных.
После окончания транзакции бит TDONE = 1 (регистр USB_HIS). Этот бит перед началом каждой транзакцией должен быть очищен записью 1. PID принятого пакета записывается в регистре USB_HRXP.
Если в ответ получен пакет NAK (рис 3а), то бит NAKRXED = 1 (регистр USB_HRXS).
Если в ответ получен пакет STALL (рис 3б), то бит STALLRXED = 1 (регистр USB_HRXS).
Если в ответ получен пакет ACK (рис 3в), то бит ACKRXED = 1 (регистр USB_HRXS).

Транзакция IN (Usb Host)

Для начала транзакции должны быть заданы адрес устройства (регистр USB_HTXA), оконечная точка (регистр USB_HTXE) и тип token пакета (регистр USB_HTXT). Транзакция запускается при установке бита TREQ регистра USB_HTXC. Host отправляет IN пакет.
После окончания транзакции бит TDONE = 1 (регистр USB_HIS). Этот бит перед началом каждой транзакцией должен быть очищен записью 1. PID принятого пакета записывается в регистре USB_HRXP.
Если в ответ получен пакет NAK (рис 4а), то бит NAKRXED = 1 (регистр USB_HRXS).
Если в ответ получен пакет STALL (рис 4б), то бит STALLRXED = 1 (регистр USB_HRXS).

Если приходит DATA0/DATA1 пакет (рис 4в), то данные начинают записываться побайтно в FIFO приема. После записи каждого байта увеличивается на единицу счетчик принятых байт. Принятые байты считываются через регистр USB_HRXFD. Количество принятых байт содержится в регистре USB_HRXFDC. После приема очередного пакета необходимо выполнять сброс указателя FIFO приема записью 1 в регистр USB_HRXFC. Бит DATASEQ регистра USB_HRXS отображает тип принятого пакета данных (0 – DATA0, 1 – DATA1).

Отправка SOF пакетов и отсчет времени (Usb Host)

Для того чтобы контроллер автоматически отправлял SOF пакеты на Full speed необходимо установить SOFEN в регистре USB_HTXSE. Если FSPL = 1 (регистр USB_TXLC), то SOF будет автоматически отсылаться каждые 1 мс. Если FSPL = 0, то автоматически будет отправляться EOP каждые 1 мс.

После отправки SOF пакета бит SOFS = 1 (регистр USB_HIS). Этот бит должен быть очищен записью 1.

Контроллер ведет счет времени во фрейме таймером. Таймер увеличивается на частоте 48 МГц и имеет 48000 тактов в 1 мс фрейме. Старший байт таймера содержится в регистре USB_HSTM. Этот регистр может быть использован для вычисления времени, оставшегося во фрейме.

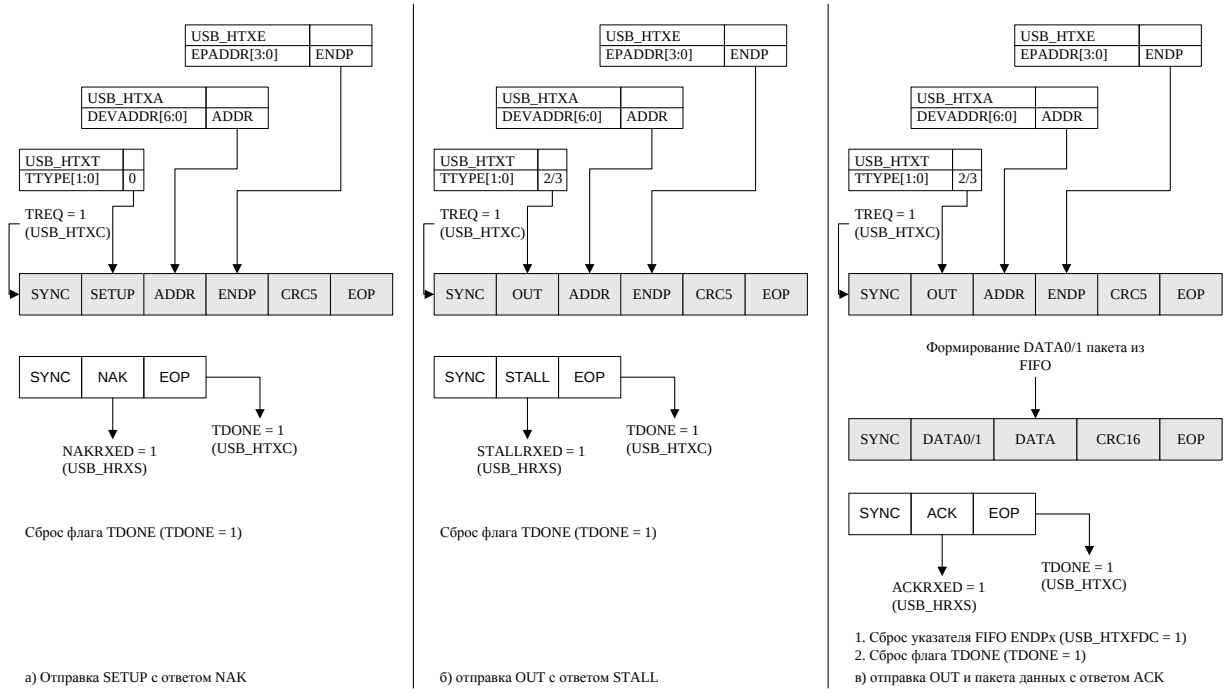


Рисунок 3

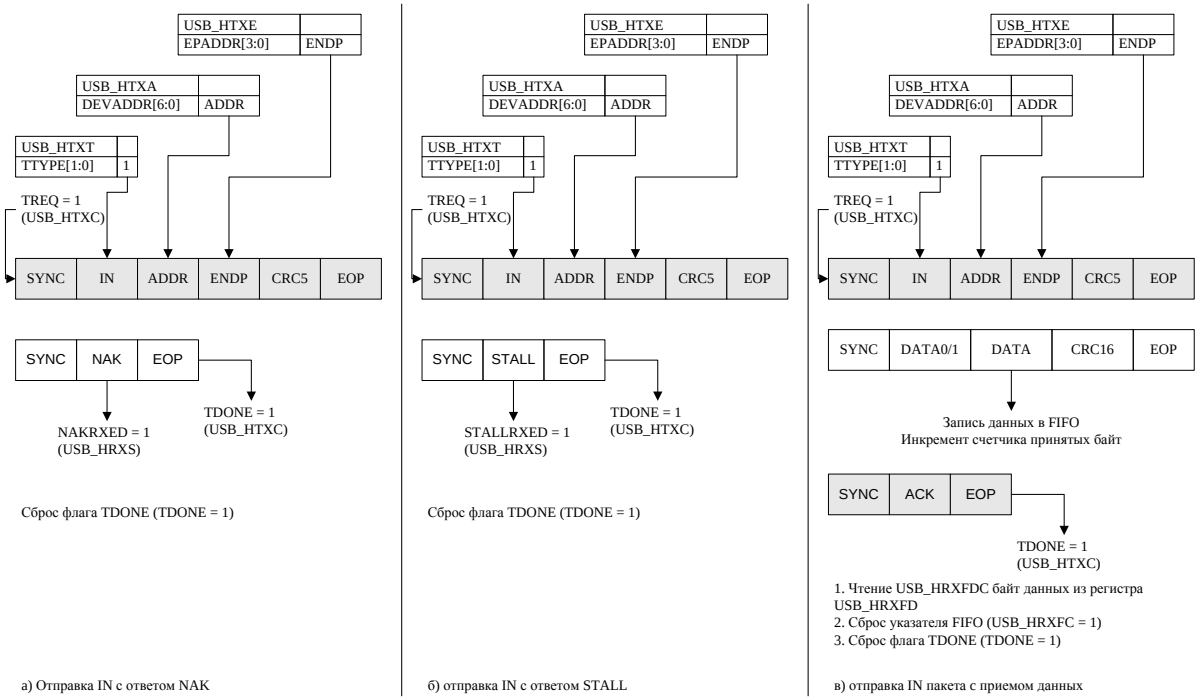


Рисунок 4

Описание регистров управление контроллером USB интерфейса

Базовый Адрес	Название	Описание
0x4001_0000	USB	Контроллер USB интерфейса
Смещение		
0x380	USB_HSCR	Общее управление для контроллера USB интерфейса
0x384	USB_HSVR	Версия аппаратного контроллера USB интерфейса
	Контроллер HOST	
0x00	USB_HTXC	Регистр управления передачей пакетов со стороны хоста
0x04	USB_HTXT	Регистр задания типа передаваемых пакетов со стороны хоста
0x08	USB_TXLC	Регистр управления линиями шины USB
0x0C	USB_HTXSE	Регистр управление автоматической отправки SOF
0x10	USB_HTXA	Регистр задания адреса устройства для

		отправки пакета
0x14	USB_HTXE	Регистр задания номера оконечной точки для отправки пакета
0x18, 0x1C	USB_HFN_L USB_HFN_H	Регистр задания номера фрейма для отправки SOF
0x20	USB_HIS	Регистр флагов событий контроллера хост.
0x24	USB_HIM	Регистра флагов разрешения прерываний по событиям контролера хоста
0x28	USB_HRXS	Регистр состояния очереди приема данных хоста
0x2C	USB_HRXP	Регистр отображения PID принятого пакета
0x30	USB_HRXA	Регистр отображения адреса устройства, от которого принят пакет.
0x34	USB_HRXE	Регистр отображения номер оконечной точки, от которой принят пакет.
0x38	USB_HRXCS	Регистр отображения состояния подсоединения устройства
0x3C	USB_HSTM	Регистр расчета времени фрейма
0x80	USB_HRXFD	Данные очереди приема
0x88, 0x8C	USB_HRXFDC_L USB_HRXFDC_H	Число принятых данных в очереди
0x90	USB_HRXFC	Управление очередью приема
0xC0	USB_HTXFD	Данные для передачи
0xD0	USB_HTXFDC	Управление очередью передачи
	Контроллер SLAVE	
		Управление очередью

0x100 0x110 0x120 0x130	USB_SEP0.CTRL USB_SEP1.CTRL USB_SEP2.CTRL USB_SEP3.CTRL	нулевой оконечной точки
0x104 0x114 0x124 0x134	USB_SEP0.STS USB_SEP1.STS USB_SEP2.STS USB_SEP3.STS	Состояние оконечной точки
0x108 0x118 0x128 0x138	USB_SEP0.TS USB_SEP1.TS USB_SEP2.TS USB_SEP3.TS	Состояние типа передачи оконечной точки
0x10C 0x11C 0x12C 0x13C	USB_SEP0.NTS USB_SEP1.NTS USB_SEP2.NTS USB_SEP3.NTS	Состояние передачи NAK оконечной точки
0x140	USB_SC	Управление контроллеров SLAVE
0x144	USB_SLS	Отображение состояния линий USB шины
0x148	USB_SIS	Флаги событий контроллера SLAVE
0x14C	USB_SIM	Флаги разрешения прерываний от контроллера SLAVE
0x150	USB_SA	Функциональный адрес контроллера
0x154, 0x158	USB_SFN_L USB_SFN_H	Номер фрейма
0x180 0x200 0x280 0x300	USB_EP0.RXFD USB_EP1.RXFD USB_EP2.RXFD USB_EP3.RXFD	Принятые данные оконечной точки
0x188, 0x18C 0x208, 0x20C 0x288, 0x28C 0x308, 0x30C	USB_EP0.RXFDC_L USB_EP0.RXFDC_H USB_EP1.RXFDC_L USB_EP1.RXFDC_H USB_EP2.RXFDC_L USB_EP2.RXFDC_H USB_EP3.RXFDC_L USB_EP3.RXFDC_H	Число данных в оконечной точке
		Управление очередью

0x190 0x210 0x290 0x310	USB_EP0.RXFC USB_EP1.RXFC USB_EP2.RXFC USB_EP3.RXFC	приема оконечной точки
0x1C0 0x240 0x2C0 0x340	USB_EP0.TXFD USB_EP1.TXFD USB_EP2.TXFD USB_EP3.TXFD	Данные для передачи через оконечную точку
0x1D0 0x250 0x2D0 0x350	USB_EP0.TXFC USB_EP1.TXFC USB_EP2.TXFC USB_EP3.TXFC	Управление очередью передачи оконечной точки

USB_HSCR

Номер	31…8	7	6	5	4	3	2	1	0
Доступ	U	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0		0	0	0	0	0	0
	-	D-PULL DOWN	D-PULL UP	D+ PULL DOWN	D+ PULL UP	EN RX	EN TX	RESET CORE	HOST MODE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7	D- PULLDOWN	Управление встроенной подтяжкой линии D- 0 – нет подтяжки вниз 1 – есть подтяжка вниз
6	D- PULLUP	Управление встроенной подтяжкой линии D- 0 – нет подтяжки вверх 1 – есть подтяжка вверх
5	D+ PULLDOWN	Управление встроенной подтяжкой линии D+ 0 – нет подтяжки вниз 1 – есть подтяжка вниз
4	D+ PULLUP	Управление встроенной подтяжкой линии D+ 0 – нет подтяжки вверх 1 – есть подтяжка вверх
3	EN_RX	Разрешение работы приемника USB 0 – запрещен 1 – разрешен Может использоваться в энергосберегающих целях
2	EN_TX	Разрешение работы передатчика USB 0 – запрещен 1 – разрешен Может использоваться в энергосберегающих целях
1	RESET_CORE	Программный сброс контроллера 1 – сброс контроллера (удерживать минимум 10 циклов USBCLK) 0 – рабочий режим
0	HOST_MODE	Режим работы контроллера 1 – режим HOST 0 – режим Device

USB_HSVR

Номер	31	7	4	3	0
Доступ	U	RO	RO	RO	RO
Сброс	0	0	0	0	0
	-	-	REVISION		VERSION

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7…4	REISION	Номер Ревизии
3…0	VERSION	Номер Версии

Регистры HOST режима

USB_HTXC

Номер	31				3	2	1	0
Доступ	U				R/W	R/W	R/W	R/W
Сброс	0				0	0	0	0
	-				ISOEN	PREEN	SOFS	TREQ

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3	ISOEN	Флаг разрешения изохронного режима 1 – разрешение изохронного режима, ACK не посылается и не принимается. Необходимо, что бы TRANS_TYPE_REG был установлен в IN_TRANS или OUTDATA0_TRANS. Изохронный режим не применим ни с какими другими типами передачи. 0 – запрещение изохронного режима
2	PREEN	Флаг разрешения приамбулы 1 – разрешение преамбулы. Должна быть установлена только когда host подсоединен к low speed устройству через хаб. Преамбула – это токен перед всеми пакетами передачи и передается на full speed независимо от состояния FULL_SPEED_LINE_RATE_BIT.
1	SOFS	Флаг задания синхронизации передачи с SOF 1 – синхронизировать передачу с окончанием SOF. Передача будет запущена сразу за передачей SOF 0 – передача не синхронизирована
0	TREQ	Флаг запроса передачи данных 1 – запрос разрешения передачи данных, автоматически сбрасывается после передачи 0 – запрещена передача

USB_HTXT

Номер	31						2	1	0
Доступ	U						U	R/W	R/W
Сброс	0						0	0	0
	-						-	TTYPE	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
1…0	TTYPE	Тип передачи 00 – setup_trans 01 – in_trans 10 – outdata0_trans 01 – outdata1_trans

USB_HTXLC

Номер	31		4	3	2	1	0
Доступ	U		R/W	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0
	-		FSLR	FSLP	DC	TXLS[1:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..5	-	Зарезервировано
4	FSLR	1 – 12 Мбит в сек. 0 – 1.5 Мбит в сек
3	FSPL	1 – FULL SPEED полярность шины USB 0 – LOW SPEED полярность шины USB Если host работает с full speed устройтсвом, full speed полярность должна быть установлена. Если работа ведется с low speed устройством на прямую, то должна быть установлена low speed полярность, если работа ведется с low speed через хаб, то должна быть установлена full speed полярность.
2	DC	Режим управления линиями шины USB 1 – разрешение прямого управления состоянием линий USB шины. 0 – нормальный режим работы
1…0	TXLC[1:0]	Если установлен бит DIRECT_CONTROL_BIT, то отображается состояние шины USB. TX_LINE_STATE[0] = D- TX_LINE_STATE[1] = D+

USB_HTXSE

Номер	31						1	0
Доступ	U						U	R/W
Сброс	0						0	0
	-						-	SOFEN

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..1	-	Зарезервировано
0	SOFEN	1 – Если FULL_SPEED_LINE_POLARITY_BIT установлен, то SOF будет автоматически отсылаться каждые 1 мс. SOF отправляется на full speed не зависимо от состояния FULL_SPEED_LINE_RATE_BIT. Если FULL_SPEED_LINE_POLARITY_BIT не установлен, то автоматически будет отправляться EOP каждые 1 мс. Это необходимо при работе с low speed устройством напрямую (не через хаб) 0 – запрет автоматической отправки SOF/EOP и позволяет подсоединенным устройствам перейти в suspend режим.

USB_HTXA

Номер	31					6	0
Доступ	U					R/W	R/W
Сброс	0					0	0
	-						DEVADDR[6:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..7	-	Зарезервировано
6…0	DEVADDR[6:0]	USB Device address. Адрес устройства для обращения

USB_HTXE

Номер	31					3	0
Доступ	U					R/W	R/W
Сброс	0					0	0
	-						EPADDR[3:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3…0	EPADDR[3:0]	Endpoint address. Номер оконченной точки устройства для обращения

USB_HFN

Номер	31					10	0
Доступ	U					R/W	R/W
Сброс	0					0	0
	-						FNUM[10:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
10…0	FNUM[10:0]	Номер фрейма

USB_HIS

Номер	31			3	2	1	0
Доступ	U			R/W	R/W	R/W	R/W
Сброс	0			0	0	0	0
	-			SOFS	CONEV	RESUME	TDONE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3	SOFS	1 – автоматически устанавливается, когда SOF был отправлен. Должен быть очищен записью 1. 0 – не было SOF
2	CONEV	1 – автоматически устанавливается, когда происходит подсоединение или отсоединение. Должно быть очищено записью 1. 0 – события не было
1	RESUME	1 – автоматически устанавливается, когда возникает состояние повтора. Должен быть очищен записью 1. 0 – не было повтора.
0	TDONE	1 – автоматически устанавливается, когда передача закончена. Должен быть очищен записью 1. 0 – передача не закончена или ее нет.

USB_HIM

Номер	31		4	3	2	1	0
Доступ	U		U	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0
	-		-	SOFSIE	CONEVIE	RESUMEIE	TDONEIE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3	SOFIE	1 – разрешение выработки прерывания при окончании передачи 0 – запрещение выработки прерывания
2	CONEVIE	1 – разрешение выработки прерывания при повторе передачи 0 – запрещение выработки прерывания
1	RESUMEIE	1 – разрешение выработки прерывания при подсоединении или отсоединении 0 – запрещение выработки прерывания
0	TDONEIE	1 – разрешение выработки прерывания при передаче SOF 0 – запрещение выработки прерывания

USB_HRXS

Номер	31..8	7	6	5	4	3	2	1	0
Доступ	U	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	0
	-	DATASEQ	ACK RXED	STALL RXED	NAK RXED	RX TO	RXOF	BSERR	CRCER

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7	DATASEQ	Если последняя транзакция была типа IN_TRANS, этот бит указывает номер последнего принятого пакета. DATA0 = 0, DATA1 = 1.
6	ACK RXED	1 – получен ACK 0 – не получен ACK
5	STALL RXED	1 – получен STALL 0 – не получен STALL
4	NAK RXED	1 – получен NAK от устройства 0 – не получен NAK
3	RXTO	1 – превышение времени ожидания ответа от устройства 0 – нет превышения времени
2	RXOF	1 – обнаружена ошибка переполнения FIFO при приеме пакета 0 – не было переполнения
1	BSERR	1 – обнаружена ошибка stuff при последней передаче 0 – ошибки stuff не было
0	CRCERR	1 – обнаружена ошибка CRC при последней передаче 0 – ошибки CRC не было

USB_HRXP

Номер	31	3	0
Доступ	U	R/W	R/W
Сброс	0	0	0
	-		RPID[3:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3…0	RPID[3:0]	Packet identifier последнего принятого пакета

USB_HRXA

Номер	31					6	1	0
Доступ	U					R/W	R/W	R/W
Сброс	0					0	0	0
	-							RADDR[6:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
6…0	RADDR[6:0]	Адрес устройства от которого принят последний пакет.

USB_HRXE

Номер	31					3	1	0
Доступ	U					R/W	R/W	R/W
Сброс	0					0	0	0
	-							RXENDP[3:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4	-	Зарезервировано
3…0	RXENDP[3:0]	Номер оконечной точки от которой принят последний пакет.

USB_HRXCS

Номер	31						2	1	0
Доступ	U						U	R/W	R/W
Сброс	0						0	0	0
	-						-	RXLS[1:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
1…0	RXLS[1:0]	Состояние линий шины USB: DISCONNECT = 0 LOW_SPEED_CONNECT = 1 FULL_SPEED_CONNECT = 2

USB_HSTM

Номер	31						7	0
Доступ	U						R/W	R/W
Сброс	0						0	0
	-							HSTM[7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7…0	HSTM[7:0]	Старший байт SOF таймера, используемого для передачи SOF. Таймер увеличивается на частоте 48MHz и имеет 48000 тактов в 1 мс фрейме. Этот регистр может быть использован для вычисления времени, оставшегося во фрейме

USB_HRXFD

Номер	31						7	0
Доступ	U						R/W	R/W
Сброс	0						0	0
	-							RX FIFO DATA[7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7…0	RX FIFO	Если последняя транзакция была IN_TRANS, то в буфере содержатся принятые данные, и они могут быть считаны

	DATA[7:0]	через этот регистр.

USB_HRXDC

Номер	31						15	0
Доступ	U						R/W	R/W
Сброс	0						0	0
	-							FIFO DATA COUNT[15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16	-	Зарезервировано
15…0	FIFO DATA COUNT[15:0]	счетчик байтов, записанных в буфер

USB_HRXFC

Номер	31						1	0
Доступ	U						U	R/W
Сброс	0						0	0
	-							FIFO FORCE EMPTY

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..1	-	Зарезервировано
0	FIFO FORCE EMPTY	Запись 1 принудительно сбрасывает указатель FIFO

USB_HTXFD

Номер	31						7	1	0
Доступ	U						R/W	R/W	R/W
Сброс	0						0	0	0
	-								TX FIFO DATA[7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано

7…0	TX FIFO DATA[7:0]	При запросах передачи OUTDATA0_TRANS или OUTDATA1_TRANS, через данный регистр должны быть загружены данные для отправки

USB_HTXFC

Номер	31							1	0
Доступ	U							U	R/W
Сброс	0							0	0
	-								FIFO FORCE EMPTY

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..1	-	Зарезервировано
0	FIFO FORCE EMPTY	Запись 1 принудительно сбрасывает указатель FIFO

USB Slave (Device)

USB_SEP0.CTRL

USB_SEP1.CTRL

USB_SEP2.CTRL

USB_SEP3.CTRL

Номер	31			4	3	2	1	0
Доступ	U			R/W	R/W	R/W	R/W	R/W
Сброс	0			0	0	0	0	0
	-			EPISOEN	EPSSTALL	EPDATASEQ	EPRDY	EPEN

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
4	EPISOEN	1 – изохронный режим передачи 0 – не изохронный режим передачи В изохронном режиме не отсылаются какие-либо подтверждения передачи
3	EPSSTALL	1 – если точка разрешена, готова и не в изохронном режиме, то на запрос хоста будет отвечен STALL 0 – не отвечать STALL
2	EPDATASEQ	1 – отвечать на IN запрос от хоста с DATA1 0 – отвечать на IN запрос от хоста с DATA0
1	EPRDY	1 – оконечная точка готова 0 – оконечная точка не готова или закончила передачу Если точка разрешена и готова, то она может ответить на инициализированную хостом передачу. Бит автоматически сбрасывается в 0 после успешного окончания передачи
0	EPEN	1 – оконечная точка разрешена. 0 – оконечная точка запрещена Если точка запрещена, она не отвечает на транзакции. Если точка разрешена, но не готова и не находится в изохронном режиме, то она отвечает NAK

USB_SEP0.STS

USB_SEP1.STS

USB_SEP2.STS

USB_SEP3.STS

Номер	31…8	7	6	5	4	3	2	1	0
Доступ	U	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	0
	-	SC DATA SEQ	SC ACK RXED	SC STALL SENT	NAK SENT	SC RXTO	SC RXOF	SC BS ERR	SC CRC ERR

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7	SC DATA SEQ	Если предыдущий тип передачи был OUT_TRANS то этот бит определяет тип принятого пакета DATA0 = 0, DATA1= 1.
6	SC ACK RXED	1 – получен ACK от хоста на переданные данные 0 – нет подтверждения
5	SC STALL SENT	1 – обозначает отправку STALL 0 – не было STALL
4	SC NAK SENT	1 – обозначает отправку NAK ответа 0 – не было NAK
3	SC RXTO	1 – обозначает возникновение ошибки времени ожидания ответа от хоста 0 – нет ошибки
2	SC RXOF	1 – обозначает возникновение переполнения буфера FIFO при приеме последнего пакета 0 – нет переполнения
1	SC BS ERR	1 – обозначает возникновение STUFF ошибки в последней передаче 0 – нет ошибки
0	SC CRC ERR	1 – обозначает возникновение CRC ошибки в последней передаче 0 – нет ошибки

USB_SEP0.TS

USB_SEP1.TS

USB_SEP2.TS

USB_SEP3.TS

Номер	31					2	1	0
Доступ	U					U	R/W	R/W
Сброс	0					0	0	0
	-					-	SCTTYPE[1:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
	SCTTYPE[1:0]	Отображает тип последней передачи, перед тем как ENDPOINT_READY_BIT был изменен с 1 на 0. SC_SETUP_TRANS = 0 SC_IN_TRANS = 1 SC_OUTDATA_TRANS = 2

USB_SEP0.NTS

USB_SEP0.NTS

USB_SEP0.NTS

USB_SEP0.NTS

Номер	31						2	1	0
Доступ	U						U	R/W	R/W
Сброс	0						0	0	0
	-							NTTYPE[1:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
1…0	NTTYPE[1:0]	Тип последней передачи, в результате которой на хост был послан NAK SC_SETUP_TRANS = 0 SC_IN_TRANS = 1 SC_OUTDATA_TRANS = 2

USB_SC

Номер	31		5	4	3	2	1	0
Доступ	U		R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0	0
	-		SCFSR	SCFSP	SCDC	SCTXLS [1:0]		SCGEN

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..6	-	Зарезервировано
5	SCFSR	Флаг управления скоростью работы 1 – 12 Мбит/с 0 – 1.5 Мбит/с
4	SCFSP	Флаг выбора полярности линий USB шины 1 – FULL SPEED 0 – LOW SPEED
3	SCDC	Флаг прямого управления линиями USB шины 1 – разрешено прямое управление 0 – запрещено прямое управление
2…1	SCTXL[1:0]	Если установлен бит SC_DIRECT_CONTROL_BIT, то через SC_TX_LINE_STATE осуществляется прямое управление состоянием линий USB шины SC_TX_LINE_STATE [1] = D+ SC_TX_LINE_STATE [0] = D-
0	SCGEN	1 – разрешение для работы с разрешенных оконечных точек 0 – все оконечные точки запрещены

USB_SLS

Номер	31					2	1	0
Доступ	U					U	R/W	R/W
Сброс	0					0	0	0
	-					-	SCRXLS[1:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
1…0	SCRXLS[1:0]	Отображает состояние подсоединения на шине USB RESET = 0 LOW_SPEED_CONNECT = 1 FULL_SPEED_CONNECT = 2

USB_SIS

Номер	31		4	3	2	1	0
Доступ	U		R/W	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0
	-		SC NAK SENT	SC SOF REC	SC RESET EV	SC RESUME	SC TDONE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..5	-	Зарезервировано
4	SC NAK SENT	Устанавливается в 1, когда отвечен NAK. Очищается записью 1
3	SC SOF REC	Устанавливается в 1, когда принят пакет SOF. Очищается записью 1
2	SC RESET EV	Устанавливается в 1, когда обнаруживается состояние сброса на шине USB. Очищается записью 1
1	SC RESUME	Устанавливается в 1, когда обнаруживается состояние повтора. Очищается записью 1
0	SC TDONE	Устанавливается в 1 после успешного выполнения передачи. Очищается записью 1

USB_SIM

Номер	31	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	-	SC NAK SENT IE	SC SOF RECIE	SC RESET EVIE	SC RESUME IE	SC TDONE IE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..6	-	Зарезервировано
4	SC NAK SENT IE	Бит разрешения прерывания при отправке NAK 1 – разрешено прерывание 0 – запрещено прерывание
3	SC SOF RECIE	Бит разрешения прерывания при приеме SOF 1 – разрешено прерывание 0 – запрещено прерывание
2	SC RESET EVIE	Бит разрешения прерывания при состоянии сброса на шине 1 – разрешено прерывание 0 – запрещено прерывание
1	SC RESUME IE	Бит разрешения прерывания при состоянии повтора 1 – разрешено прерывание 0 – запрещено прерывание
0	SC TDONE IE	Бит разрешения прерывания при окончании передачи 1 – разрешено прерывание 0 – запрещено прерывание

USB_SA

Номер	31						6	0
Доступ	U						R/W	R/W
Сброс	0						0	0
	-							SDEVADDR[6:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
6…0	SDEVADDR [6:0]	Функциональный адрес USB Device

USB_SFN

Номер	31						10	0
Доступ	U						R/W	R/W
Сброс	0						0	0
	-							FRAME NUM [10:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..11	-	Зарезервировано
10…0	FRAME NUM [10:0]	Номер фрейма принятый в последнем SOF

USB_SEP0.RXFD

USB_SEP1.RXFD

USB_SEP2.RXFD

USB_SEP3.RXFD

Номер	31					7	0
Доступ	U					R/W	R/W
Сброс	0					0	0
	-						RX FIFO DATA[7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7…0	RX FIFO DATA[7:0]	После приема OUTDATA_TRANS или SETUP_TRANS пакета, принятые данные читаются из регистра RX_FIFO_DATA

USB_SEP0.RXFDC

USB_SEP1.RXFDC

USB_SEP2.RXFDC

USB_SEP3.RXFDC

Номер	31						15	0
Доступ	U						R/W	R/W
Сброс	0						0	0
	-							FIFO DATA COUNT [15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…16	-	Зарезервировано
15…0	FIFO DATA COUNT [15:0]	Отображает число байт, записанных в буфер FIFO

USB_SEP0.RXFC

USB_SEP1.RXFC

USB_SEP2.RXFC

USB_SEP3.RXFC

Номер	31						1	0
Доступ	U						U	R/W
Сброс	0						0	0
	-							FIFO FORCE EMPTY

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..1	-	Зарезервировано
0	FIFO FORCE EMPTY	Запись 1 очищает указатель буфера FIFO

USB_SEP0.TXFD

USB_SEP1.TXFD

USB_SEP2.TXFD

USB_SEP3.TXFD

Номер	31						7	0
Доступ	U						R/W	R/W
Сброс	0						0	0
	-							TX FIFO DATA[7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2	-	Зарезервировано
	TX FIFO DATA [7:0]	Перед приемом IN_TRANS в буфер FIFO записываются данные для отправки

USB_SEP0.TXFDC

USB_SEP1.TXFDC

USB_SEP2.TXFDC

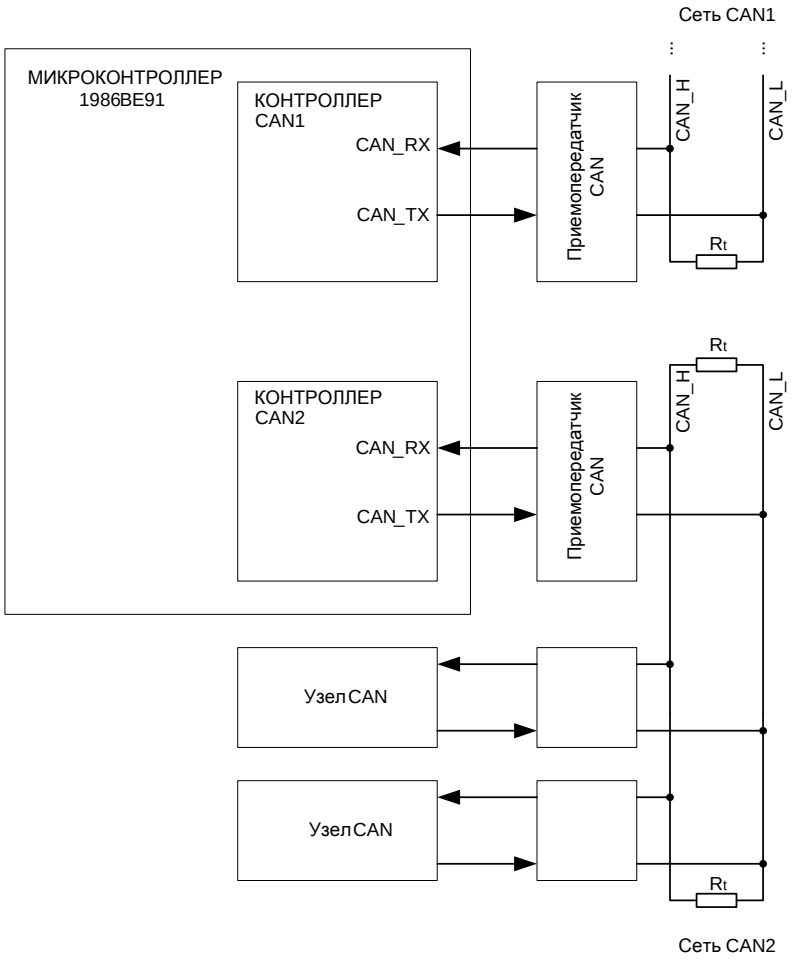
USB_SEP3.TXFDC

Номер	31						1	0
Доступ	U						U	R/W
Сброс	0						0	0
	-							FIFO FORCE EMPTY

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..1	-	Зарезервировано
0	FIFO FORCE EMPTY	Запись 1 очищает указатель буфера FIFO

Контроллер CAN интерфейса

В микроконтроллере реализовано два независимых цифровых контроллера интерфейса CAN. Они являются полнофункциональными CAN-узлами, отвечающими требованиям к активным и пассивным устройствам CAN 2.0A и 2.0B и поддерживающими передачу данных на скорости до 1 Мбит/сек.



Структурная блок – схема организации сети CAN

Интерфейс CAN позволяет обмениваться сообщениями в сети равноправных устройств. При передаче сообщения в сети CAN все узлы сети получают это сообщение. В сообщении передается уникальный идентификатор узла и данные. Все сообщения в протоколе CAN довольно короткие и могут содержать не более восьми байтов данных. При возникновении коллизий (одновременная передача сообщений различными узлами) при передаче идентификатора происходит арбитраж, и узел с большим номером идентификатора уступает сеть узлу с меньшим номером идентификатора.

Особенности

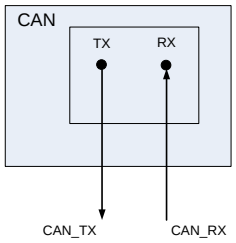
- Поддержка CAN протокола версии CAN 2.0 A и B
- Скорость передачи до 1 Мбит/с
- 32 буфера приема/передачи
- Поддержка приоритетов сообщений
- 32 фильтра приема
- Маскирование прерываний

Режимы работы

CAN-контроллер поддерживает несколько режимов работы: нормальный режим для приема и передачи пакетов сообщений, режим работы только на прием, режим самотестирования и режим инициализации для задания параметров связи.

- Режим нормальной передачи (регистр CAN_STATUS: ROM = 0, STM = 0)

Выводы CAN_TX и CAN_RX подключены к шине.

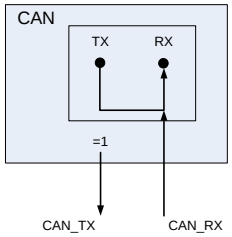


Режим нормальной передачи

В этом режиме можно установить флаги разрешения приема своих пакетов и разрешения подтверждения своих пакетов посылкой ACK (регистр CAN_CONTROL поля SAP и ROP).

- Режим работы только на прием - Receive Only Mode (регистр CAN_STATUS: ROM = 1, STM = 0)

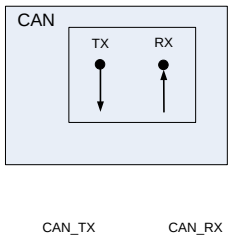
Контроллер CAN интерфейса принимает, но не посылает никакой информации, т.е. линия TX всегда в «1», но внутри контроллера все управляющие сигналы проходят.



Режим работы только на прием - Receive Only Mode

- Режим самотестирования - Self Test Mode (регистр CAN_STATUS: STM = 1, ROM = 0)

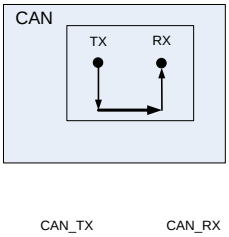
Выводы CAN_TX и CAN_RX отключены, вся передаваемая информация видна только внутри контроллера.



Режим самотестирования - Self Test Mode

Для успешного приема своих сообщений необходимо установить флаги разрешения приема своих пакетов и разрешения подтверждения своих пакетов посылкой ACK (регистр CAN_CONTROL поля SAP и ROP). В этом режиме передаваемые сообщения

сразу же принимаются в приемный буфер. Режим самотестирования полезен в период отладки кода программы.



Режим инициализации для задания параметров связи

Еще одна важная функция CAN-контроллера – фильтрация получаемых сообщений. Поскольку CAN является широковещательной шиной, каждое переданное сообщение принимается всеми узлами шины. В CAN-шине любой разумной степени сложности передается достаточно большое число сообщений. Задачей каждого подключенного к CAN-узлу ЦПУ является реагирование на CAN-сообщения. Таким образом, чтобы избавить CAN-контроллер от проблемы приема в буфер нежелательных сообщений, необходима их фильтрация. У CAN-контроллера микроконтроллеров 1986BE имеется 32 регистра фильтров и 32 регистра масок, которые можно использовать для блокировки всех CAN-сообщений, кроме избранных сообщений или групп сообщений.

Типы пакетов сообщений

Информация на шине представлена в виде фиксированных сообщений различной, но ограниченной длины. Когда шина свободна, любой подключенный узел может начать передавать новое сообщение. При передаче информации с помощью протокола CAN используется четыре типа пакетов.

- Пакет удаленного запроса данных передается узлом, чтобы запросить передачу пакета данных с тем же самым идентификатором.
- Пакет ошибки передается любым узлом при обнаружении ошибочного состояния на шине. Пакет ошибки передается сразу же после обнаружения ошибки и накладывается на передаваемый пакет так, чтобы испортить его окончательно. Таким образом, если один из узлов обнаружил ошибку, он усиливает ошибку для того, чтобы ее обнаружили и другие узлы.
- Пакет перегрузки используется для обеспечения дополнительной задержки между предшествующим и последующим кадрами данных или кадрами удаленного запроса данных. Он передается в редких случаях, подробнее можно прочесть в стандарте ISO 11898-1. Контроллер CAN интерфейса отсылает пакет перегрузки в соответствии со стандартом.
- Основными пакетами на шине CAN являются пакеты данных. Пакет данных передает данные от передатчика приемнику. Пакеты могут быть стандартными и расширенными. Отличие пакетов заключается в размере полей идентификатора. Пакеты с 11 разрядным идентификатором – называются стандартными пакетами, пакеты, содержащие 29 разрядные идентификаторы, называются расширенными пакетами. При передаче идентификационной информации происходит автоматический арбитраж на шине CAN таким образом, чтобы пакет с меньшим значением поля ID остался на шине. На шине не допускается наличие двух или более узлов с одним и тем же идентификатором. Размер передаваемых данных кодируется в поле DLC и может составлять от 0 до 8 байт. После передачи поля

данных контроллер автоматически передает рассчитанное значение CRC. Если хотя бы один из узлов принял пакет, то он выставляет ACK подтверждение на шине, если хотя бы один из узлов обнаружит ошибку, то на шину будет выставлен пакет ошибки. Таким образом, обеспечивается гарантированность доставки сообщений.

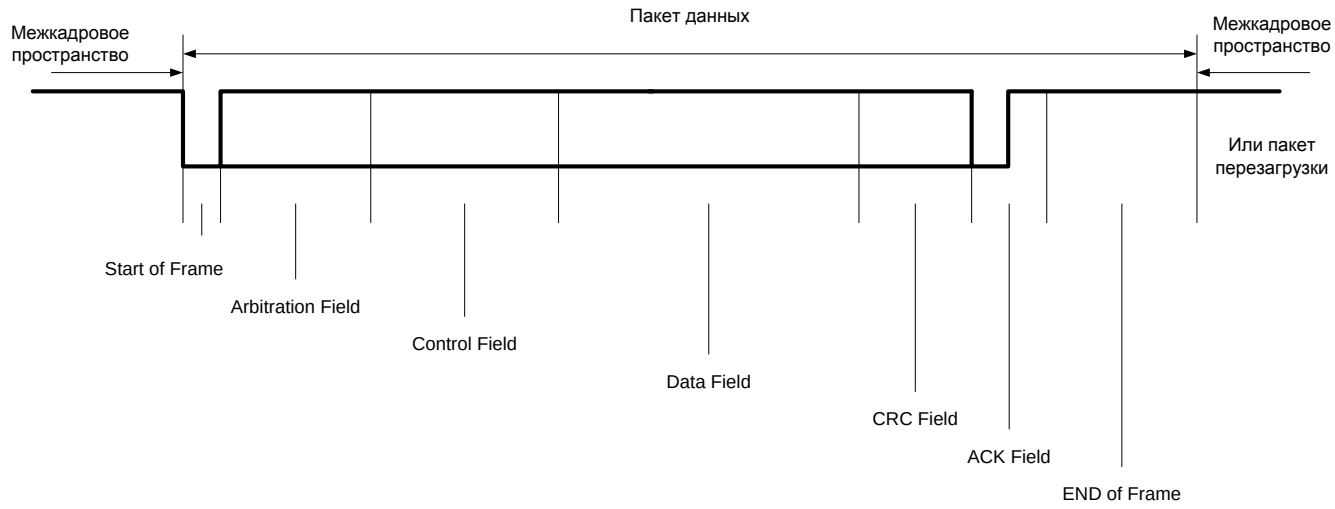
Пакеты данных и пакеты удаленного запроса данных отделяются от предшествующих пакетов межкадровым пространством.

Структура пакета данных (Data Frame)

Пакет данных состоит из 7 различных полей:

- "начало пакета" (SOF-start of frame),
- "поле арбитража" (arbitration field),
- "поле контроля" (control field),
- "поле данных" (data field),
- "поле CRC" (CRC field),
- "поле подтверждения" (ACK field),
- "конец пакета" (end of frame).

Поле данных может иметь нулевую длину.



Пакет сообщения CAN

В терминах протокола CAN логическая единица называется рецессивным битом, а логический ноль называется доминантным битом. Во всех случаях доминантный бит будет затирать рецессивный. То есть, если несколько узлов выставят на шину рецессивный бит, а один – доминантный, то обратно всеми узлами будет считан доминантный бит.

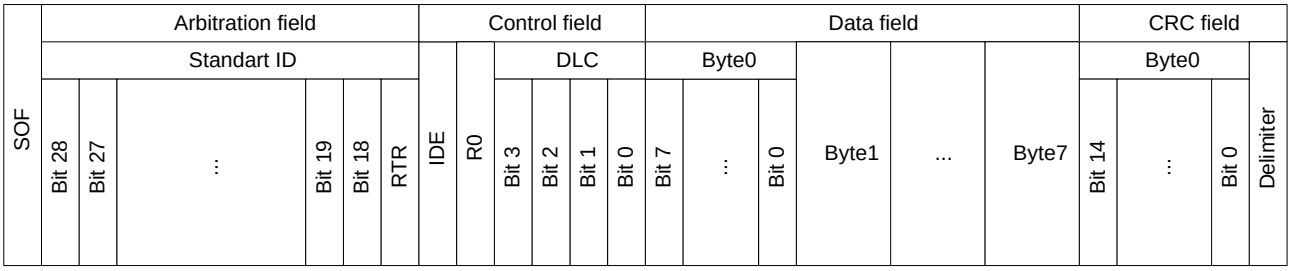
Начало пакета (Start of frame)

Начало пакета отмечает начало пакета данных или пакета удаленного запроса данных. Это поле состоит из одиночного доминантного бита. Узлу разрешено начать передачу, когда шина свободна. Все узлы должны синхронизироваться по фронту, вызванному передачей поля «начало пакета» узла, начавшего передачу первым.

Поле арбитража (Arbitration field)

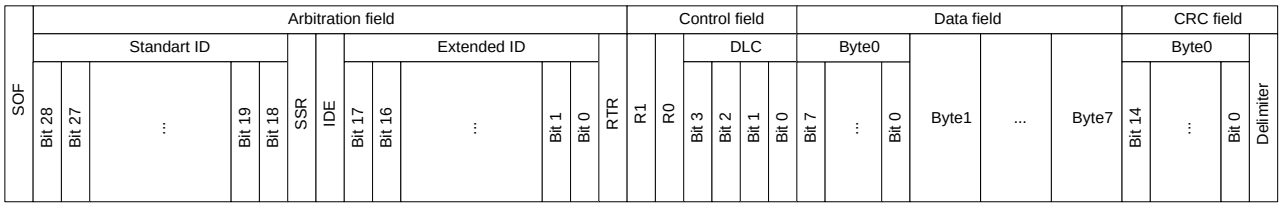
Формат поля арбитража отличается для стандартного и расширенного форматов:

– в стандартном формате поле арбитража состоит из 11 разрядного идентификатора и RTR-бита.



Структура стандартного пакета данных

– в расширенном формате поле арбитража состоит из 29 разрядного идентификатора, SRR-бита, IDE-бита и RTR-бита.



Структура расширенного пакета данных

Идентификатор

Идентификатор - стандартный формат. Длина идентификатора - 11 бит и соответствует Standart ID в расширенном формате. Эти биты передаются в порядке Bit28 … Bit18. Самый младший бит - Bit18. 7 старших битов (Bit28 - Bit 22) не должны быть все единичными битами.

Идентификатор - расширенный формат. В отличие от стандартного идентификатора, расширенный идентификатор состоит из 29 бит. Его формат содержит две секции:

– Standart ID - 11 бит
– Extended ID - 18 бит

Standart ID состоит из 11 бит. Эта секция передается в порядке от Bit28 … Bit18. Это эквивалентно формату стандартного идентификатора. Standart ID определяет базовый приоритет расширенного пакета.

Extended ID состоит из 18 бит. Эта секция передается в порядке от Bit17 до Bit0. В стандартном пакете идентификатор сопровождается RTR битом.

Бит RTR

Бит запроса удаленной передачи. В пакетах данных RTR бит должен быть передан нулевым уровнем. Внутри пакета удаленного запроса данных RTR бит должен быть единичным. В расширенном пакете сначала передается Standart ID, с последующими битами IDE и SRR. Extended ID передается после SRR бита.

Бит SRR (расширенный формат)

Заменитель бита удаленного запроса. SRR - единичный бит. Он передается в расширенных пакетах в позиции RTR бита. Таким образом, он заменяет RTR - бит стандартного пакета.

Следовательно, при одновременной передаче стандартного пакета и расширенного пакета, Standart ID которого совпадает с идентификатором стандартного пакета, стандартный пакет преобладает над расширенным пакетом.

Бит IDE (расширенный формат)

Бит расширения идентификатора

IDE бит принадлежит:

- Полю арбитража для расширенного формата
- Полю управления для стандартного формата

IDE бит в стандартном формате передается нулевым уровнем, в расширенном формате IDE бит – единичный уровень.

Поле управления (Control field)

Поле управления состоит из шести битов. Формат поля управления отличается для стандартного и расширенного формата.

Пакеты в стандартном формате включают: код длины данных (DLC), бит IDE, который передается нулевым уровнем (см. выше), и зарезервированный бит r0.

Пакеты в расширенном формате включают код длины данных и два зарезервированных бита r1 и r0. Зарезервированные биты должны быть посланы нулевым уровнем, но приемники принимают единичные и нулевые уровни биты во всех комбинациях.

Код длины данных (Data length code)

Число байт в поле данных обозначается кодом длины данных. Этот код длины данных, размером 4 бита, передается внутри поля управления. Допустимое число байт данных: {0,1, ..., 7,8}. Другие величины использоваться не могут.

Поле данных (Data field)

Поле данных состоит из данных, которые будут переданы внутри пакета данных. Оно может содержать от 0 до 8 байт, каждый содержит 8 бит, которые передаются, начиная со старшего значащего бита.

Поле CRC (CRC field)

Содержит последовательность CRC и CRC - разделитель. При вычислении 15 битного CRC кода используется последовательность бит, состоящая из полей: "начало пакета", "поле арбитража", "управляющее поле", "поле данных" (если есть). Последовательность CRC сопровождается разделителем CRC, который состоит из одного единичного бита.

Поле подтверждения (ACK field)

Поле подтверждения имеет длину два бита и содержит: "область подтверждения" и разделитель подтверждения. В поле подтверждения передающий узел посылает два бита с единичным уровнем. Приемник, который получил сообщение правильно (CRC

соответствует), сообщает об этом передатчику, посылая бит с нулевым уровнем в течение приема поля "область подтверждения".

Конец пакета (End of frame)

Каждый пакет данных и пакет удаленного запроса данных ограничен последовательностью флагов, состоящей из семи единичных бит.

Структура пакета удаленного запроса данных (Remote frame)

Узел, действующий как приемник некоторых данных, может инициировать передачу соответствующих данных узлами-источниками, посылая пакет удаленного запроса данных. Пакет удаленного запроса данных существует и в стандартном формате, и в расширенном формате. В обоих случаях он состоит из шести битовых полей:

- "начало пакета" (Start of frame),
- "поле арбитража" (Arbitration field),
- "управляющее поле" (Control field),
- "поле CRC" (CRC - field),
- "поле подтверждения" (ACK field),
- "конец пакета" (End of frame).

В отличие от обычного пакета данных, RTR бит пакета удаленного запроса данных - единичный. В этом пакете отсутствует поле данных. При этом значение кода длины данных может принимать любое значение в пределах допустимого диапазона от 0 до 8. Значение кода длины данных соответствует коду длины данных кадра данных. RTR бит указывает, является ли переданный кадр кадром данных.

Арбитраж на шине

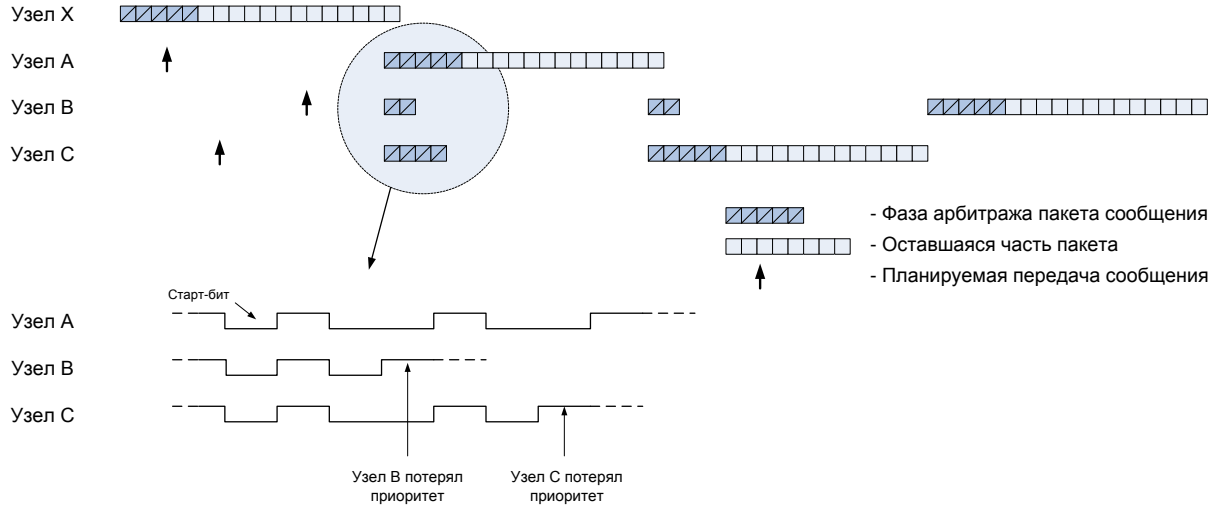
Арбитраж сообщений гарантирует, что наиболее важное сообщение захватит шину и будет передано без задержки. Затем будут переданы приостановленные сообщения согласно их приоритетам (сообщение с наименьшим идентификатором передается первым).

Если планируется передача сообщения, и шина свободна, то сообщение будет передано и сможет быть принято любым заинтересованным в нем узлом. Если передача сообщения запланирована, а шина активна, то прежде чем приступить к передаче сообщения, необходимо дождаться освобождения шины. Если запланирована передача нескольких сообщений, то при освобождении шины они начнут передаваться одновременно, синхронизируясь по признаку начала пакета. В этом случае на шине начнется процесс арбитража, задача которого – определить, какое именно из сообщений захватит шину и будет передано.

Арбитраж сообщений на шине CAN осуществляется методом, который называется «неразрушающий побитовый арбитраж».

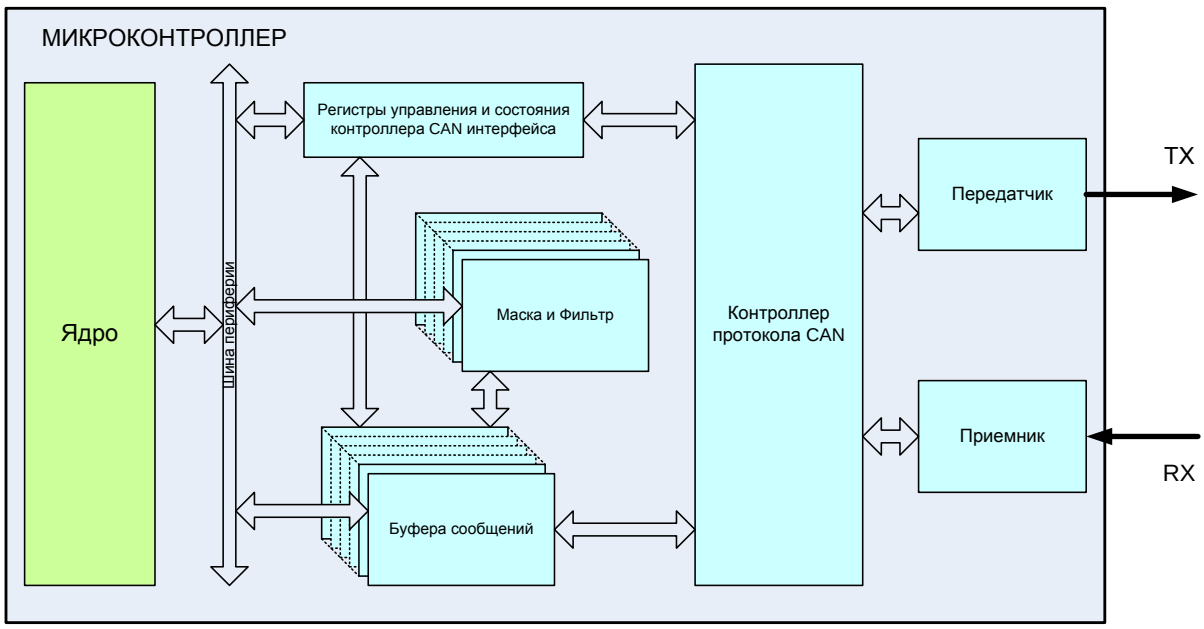
На рисунке изображены три сообщения, ожидающие передачи. После освобождения шины и синхронизации пакетов сообщений по старт-биту на шину начинают выдаваться все три идентификатора. При передаче первых двух битов все три узла выставляют на шину одинаковые логические уровни и соответственно считывают те же значения, поэтому они все продолжают передачу. Однако при передаче третьего бита узлы A и C выставляют на шину доминантный бит, а узел B выставляет рецессивный бит, но при этом считывает с шины доминантный. В результате узел B освобождает шину и начинает следить за ее состоянием. Узлы A и C продолжают передачу, пока ситуация не

повторится; теперь узел С выдаёт рецессивный бит, а узел А - доминантный. При этом узел С прекращает передачу и начинает следить за состоянием шины. С этого момента шина захватывается узлом А. После передачи сообщения узлом А узлы В и С начинают передачу, причем узел С захватит шину и передает свое сообщение. Если бы узлу А снова надо было передавать сообщение, он снова захватил бы шину. Таким образом, первым на шине CAN передается сообщение с наименьшим идентификатором.



Арбитраж на шине CAN

В случае «проигрыша» арбитража в регистре статуса контроллера CAN будет установлен флаг ID_LOWER.



Структурная блок-схема контроллера CAN

Инициализация

Перед началом работы с контроллерами CAN в первую очередь должны быть заданы параметры их тактового сигнала. Параметры задаются в блоке «Сигналы тактовой частоты».

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 0 для CAN1, бит 1 для CAN2 регистра PER_CLOCK). В регистре CAN_CLOCK установить бит CANyCLKEN, чтобы разрешить тактовую частоту для определенного контроллера CAN, задать коэффициент деления тактовой частоты HCLK для каждого CAN контроллера.

После подачи тактового сигнала на контроллер CAN можно приступать к работе с ним.

Для работы контроллера шины CAN он должен быть настроен на соответствующую скорость шины CAN. Для этого должны быть заданы соответствующим образом поля SB, SJW, SEG2, SEG1, PSEG и BRP в регистре CAN_BITTMNG. После этого должны быть заданы работающие буфера сообщений путем задания битов EN (разрешение работы) RXTXn (1 – прием, 0 - передача) в регистре BUF_xx_CON. После этого должен быть выдан общий сигнал разрешения работы контроллера через задание бита CANEN в регистре CONTROL. После этого контроллер CAN начинает работу.

Передача сообщений

Для передачи сообщения необходимо в разрешенный для работы и конфигурируемый на передачу буфер записать сообщение для передачи (задать значения регистрам CAN_BUF[x].ID, CAN_BUF[x].DLC, CAN_BUF[x].DATAL и CAN_BUF[x].DATAH), после чего установить бит TX_REQ. После установки этого бита сообщение будет поставлено в очередь на отправку. После отправки сообщения бит TX_REQ будет автоматически сброшен. Если в нескольких буферах есть сообщения на отправку, то порядок отправки определяется по полю PRIOR_0. Если у сообщения выставлен бит PRIOR_0, то оно отправляется в первую очередь. Если есть несколько сообщений с одинаковым приоритетом, то порядок отправки определяется порядковым номером буфера, буфер с меньшим порядковым номером имеет больший приоритет. Значение полей ID для выбора порядка отправки в рамках контроллера CAN (одного узла) значения не имеет. По ID выбирается приоритет между различными узлами.

Передача сообщений по Remote Transmit Request (RTR)

Для автоматической отправки сообщения по запросу Remote Transmit Request необходимо задать режим маскирования для данного буфера таким образом, чтобы он принимал только сообщения от устройства, которое может выслать RTR запрос. В регистре INT_TX разрешить генерацию прерывания при отправке для соответствующего буфера. В регистре управления этим буфером (BUFF_CON[x]) проверить, что флаг TX_REQ = 0, задать приоритет отправляемого сообщения PRIOR_0, установить разрешение ответа при приеме RTR в буфер (RTR_EN=1), задать RX_TX = 0 для разрешения отправки сообщения и задать EN = 1 для разрешения работы буфера. В регистре идентификации задать необходимые SID и EID, в регистре BUF_xx_DLC указать формат пакета (расширенный или стандартный) и указать длину передаваемых данных в поле DLC. В регистрах данных CAN_BUF[x].DATAL и CAN_BUF[x].DATAH задать необходимые для отправки данные. Далее можно переходить к выполнению остальной части программы с отправкой CAN сообщений. Отправка сообщения буфером будет произведена по RTR запросу, удовлетворяющему механизму фильтрации для принимаемых сообщений, который выбран для данного буфера.

Прием сообщений

Для приема сообщений необходимо иметь свободные и разрешенные для работы буфера, сконфигурированные на прием сообщений. При этом если по шине CAN будут передаваться сообщения от других узлов, они будут сохраняться в этих буферах.

Автоматическая фильтрация принимаемых сообщений

Для уменьшения затрат процессорного ядра на обработку принимаемых сообщений, контроллер CAN интерфейса может автоматически фильтровать принимаемые сообщения. Для каждого буфера могут быть заданы маска (CAN_BUF_FILTER[x].MASK) и фильтр (CAN_BUF_FILTER[x].FILTER) таким образом, что в этот буфер будут приниматься только те сообщения, для которых выполняется условие:

```
ID & CAN_BUF_FILTER[x].MASK == CAN_BUF_FILTER[x].FILTER
```

Если принимаемое сообщение не может быть помещено ни в один из буферов, то оно будет проигнорировано. Если сообщение может быть принято более чем одним буфером, то оно будет помещено в буфер с меньшим порядковым номером. При инициализации после включения питания или сброса CAN_BUF_FILTER[x].MASK и CAN_BUF_FILTER[x].FILTER для всех буферов имеют произвольное значение, таким образом, необходимо перед началом работы их проинициализировать. Для приема всех сообщений без фильтрации необходимо задать им нулевое значение. Специального бита для включения или выключения фильтрации нет.

Перезапись принятых сообщений

В буфере может быть включено разрешение перезаписи принятого сообщения. Если принимаемое сообщение не может быть сохранено в свободный буфер, то оно может быть сохранено в буфер с ранее полученным сообщением, если для него выставлен бит OVER_EN. При этом выставляется флаг OVER_WR. Таким образом, если у буфера разрешена перезапись принятых сообщений, после прочтения сообщения необходимо проверить флаг OVER_WR. Если он выставлен в 1, то необходимо сбросить OVER_WR (не сбрасывая флаг RX_FULL), затем еще раз прочесть сообщение, после чего снова проверить флаг OVER_WR и, если он не выставлен повторно, то сбросить флаг RX_FULL. И считанное значение считать корректным.

Прибегать к помощи механизма перезаписи принятых сообщений можно только в случае, когда допустима потеря сообщений, работа с перезаписью сообщений не гарантирует прием всех сообщений, а только позволяет принять сообщение корректно, так как момент чтения сообщения может совпасть с моментом сохранения нового сообщения. При этом первая часть считанного процессорным ядром сообщения будет от первого сообщения, вторая от второго. Если же между сбросом флага OVER_WR, чтением сообщения и при следующей проверке OVER_WR он оказался не выставлен, это означает, что в момент чтения сообщения из буфера в него не сохранялось новое сообщение.

Задание скорости передачи и момента семплирования

Все узлы шины CAN должны работать на одной скорости. Протокол CAN использует кодирование без возврата в ноль (NRZ). Также при передаче не передаются тактовые сигналы. Таким образом, приемники должны засинхронизоваться под тактовый сигнал передатчика. Поскольку все узлы имеют свои индивидуальные тактовые генераторы, все приемники имеют специальный блок синхронизации DPLL.

Максимальная скорость передачи CAN 1 Мбит/сек. Время битового интервала Nominal Bit Time определяется как

```
TBIT = 1/Скорость передачи
```

Блок DPLL разбивает битовый интервал на интервалы Time Quanta (TQ). Битовый интервал состоит из 4 частей:

- Synchronization Segment (Sync_Seg)
- Propagation Time Segment (PSEG)
- Phase Buffer Segment 1 (SEG1)
- Phase Buffer Segment 2 (SEG2)

По определению Nominal Bit Time программируется длительностью от 8 до 25 TQ. В этом случае

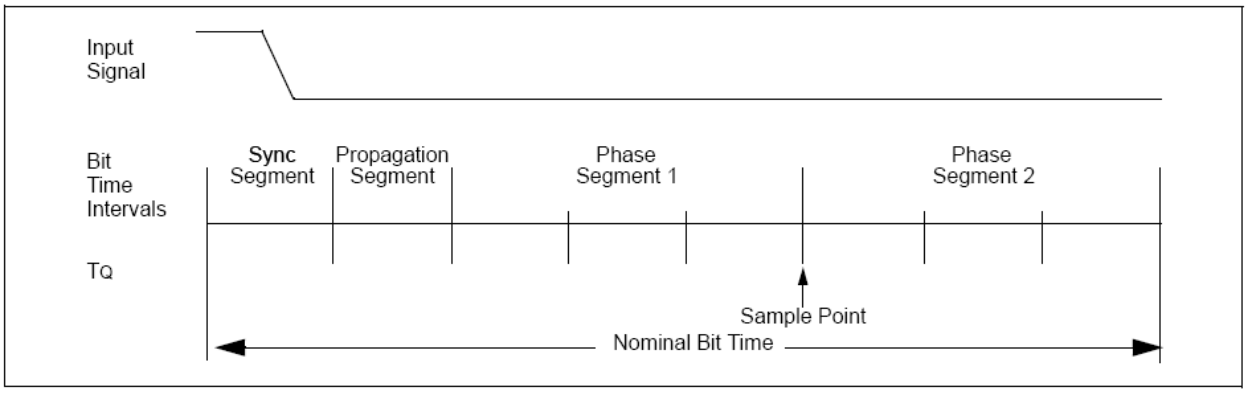
```
Nominal Bit Time = TQ * (Sync_Seg + PSEG + SEG1 + SEG2)
```

Время TQ фиксировано и определяется периодом генератора и программируемым прескалером BRP со значением от 1 до 65536:

```
TQ (µs) = ((BRP+1))/CLK (MHz)
```

или

```
TQ (µs) = ((BRP+1)) * Tclk (µs)
```



Структура битового интервала

Synchronization Segment

Эта часть битового интервала, в которой должно происходить переключение сигнала. Длительность этого интервала 1 TQ. Если переключение происходит в этой области, то приемник засинхронизирован с передатчиком.

Propagation Time Segment

Эта часть предназначена, чтобы компенсировать физические задержки времени распространения сигнала в шине и внутренние задержки в узлах. Длительность этого интервала может быть запрограммирована от 1 до 8 TQ

Phase Buffer Segments

Эти интервалы предназначены для более точной установки точки семплирования, которая располагается между ними. Длительности этих интервалов могут быть запрограммированы между 1 и 8 TQ.

Синхронизация

При обнаружении фронта принимаемого сигнала этот момент принимается как граница между битовыми интервалами; в зависимости от того, на какой интервал приходится фронт, DPLL выполняет различного рода действия по подсинхронизации данных.

Hard Synchronization

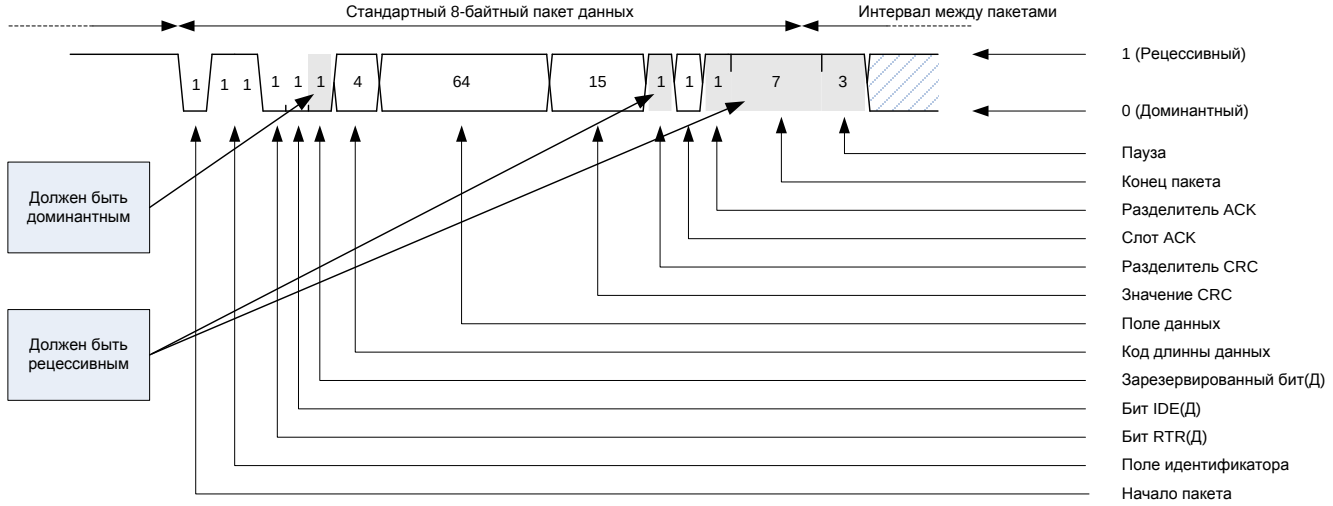
Жесткая синхронизация выполняется однократно во время начала приема сообщения. Независимо от того, в каком состоянии находился DPLL при возникновении фронта, он переводится в Sync_Seg.

Resynchronization

Если фронт принимаемого сигнала отклоняется от Sync_Seg, длительность Phase Segment 1 может быть увеличена, а Phase Segment 2 уменьшена, чтобы в следующий раз фронт прошел в нужном месте. Величина изменения Phase Segment 1 и Phase Segment 2 варьируется в зависимости от значения отклонения фронта, но не превышает значения Synchronization Jump Width (SJW).

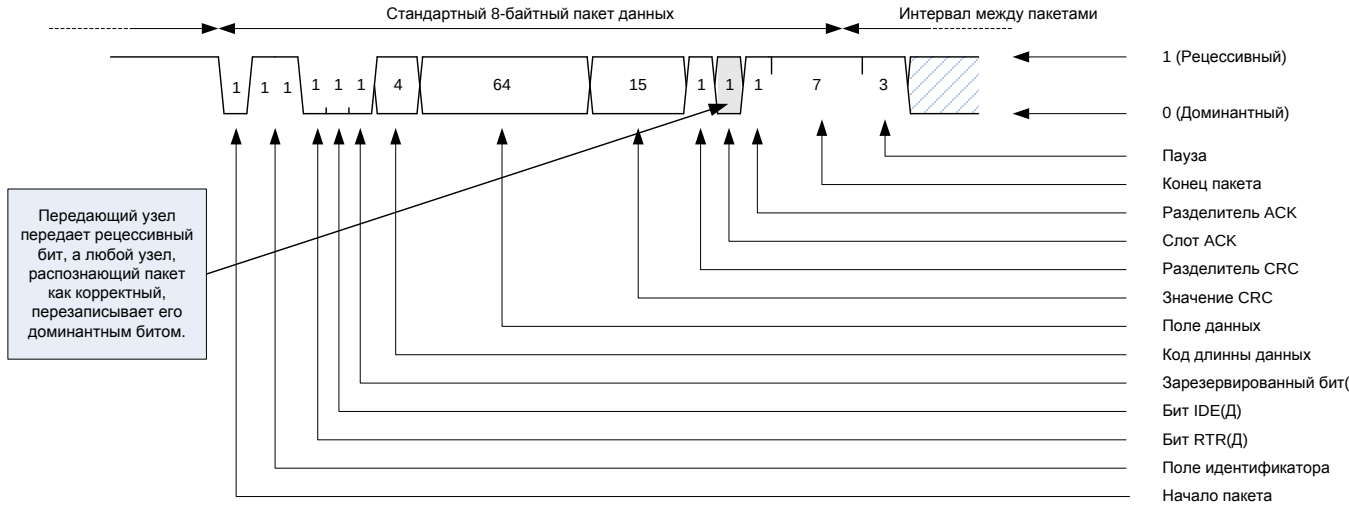
Обработка ошибок

В спецификации протокола CAN определено пять методов ограничения распространения ошибок, реализованных на аппаратном уровне. При обнаружении любой ошибки передающее устройство повторяет посылку пакета, поэтому ядру не нужно вмешиваться до тех пор, пока не возникнет грубая ошибка. Предусмотрено три метода обнаружения ошибок на уровне пакетов (контроль формата, CRC и подтверждение) и два метода на уровне битов (контроль битов и битстаффинг). Для реализации этих методов используется несколько полей, добавляемых к основному сообщению. При приеме осуществляется проверка, все ли поля присутствуют в сообщении. Если нет, то сообщение игнорируется, генерируется кадр ошибки и в регистре статуса контроллера STATUS устанавливается флаг ошибки формата пакета FRAME_ERR.



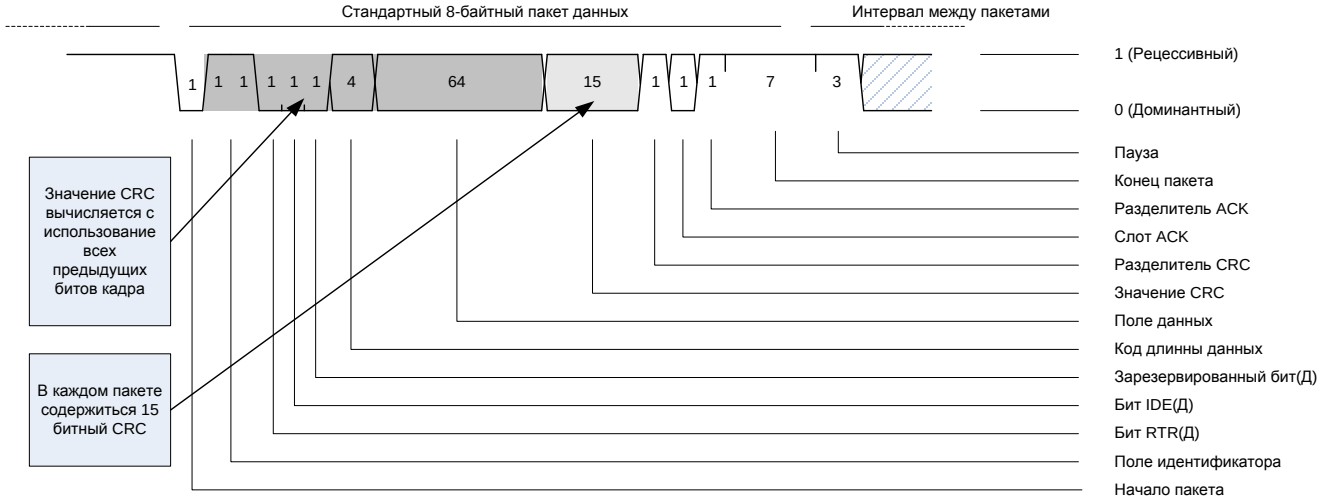
Контроль формата пакета

Каждое сообщение должно подтверждаться вставкой доминантного бита в поле подтверждения. Если подтверждения нет, передающий узел будет передавать сообщение до тех пор, пока не получит подтверждение, при этом в регистре статуса контроллера STATUS будет установлен флаг ошибки подтверждения ACK_ERR.



Контроль подтверждения

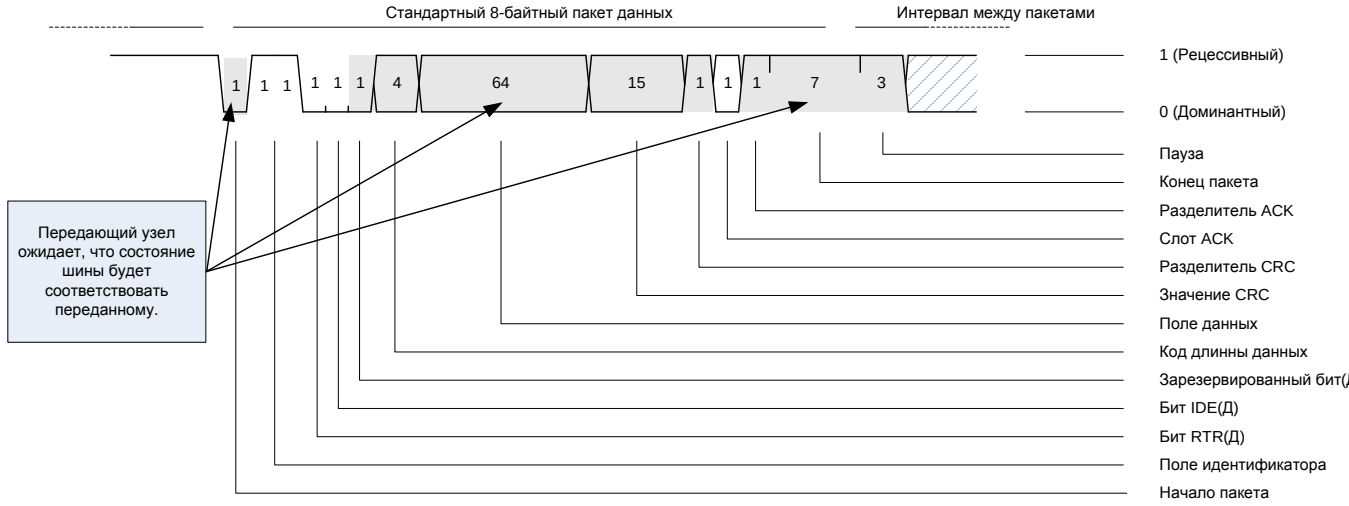
Пакет сообщения CAN содержит 15-битовое значение CRC, которое автоматически генерируется передатчиком и проверяется приемником. С помощью этого кода можно обнаружить и исправить ошибку в 4-х битах сообщения от начала кадра до начала поля CRC. Если CRC неверен и сообщение игнорируется, то передается кадр ошибки, и в регистре статуса контроллера STATUS будет установлен флаг ошибки контрольной суммы пакета CRC_ERR.



Контроль CRC

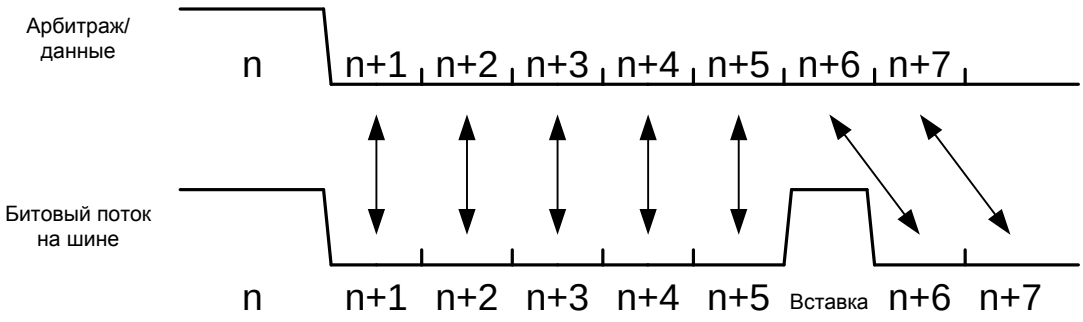
После того, как узел выиграет арбитраж, он начинает передачу своего сообщения по шине. Как и во время арбитража, CAN-контроллер считывает обратно каждый бит, выдаваемый им на шину. Поскольку узел уже выиграл арбитраж, больше никто не должен передавать данные на шине, поэтому значение каждого выданного на шину бита должно соответствовать значению, считанному обратно с шины. Если считано неверное значение, передатчик генерирует кадр ошибки, в регистре статуса контроллера STATUS

устанавливается флаг ошибки передаваемых битов пакета BIT_ERR, и сообщение снова ставится в очередь. Это сообщение будет послано в следующем слоте сообщений, однако, при этом оно должно пройти через процесс арбитража с другими запланированными сообщениями.



Контроль передаваемых битов

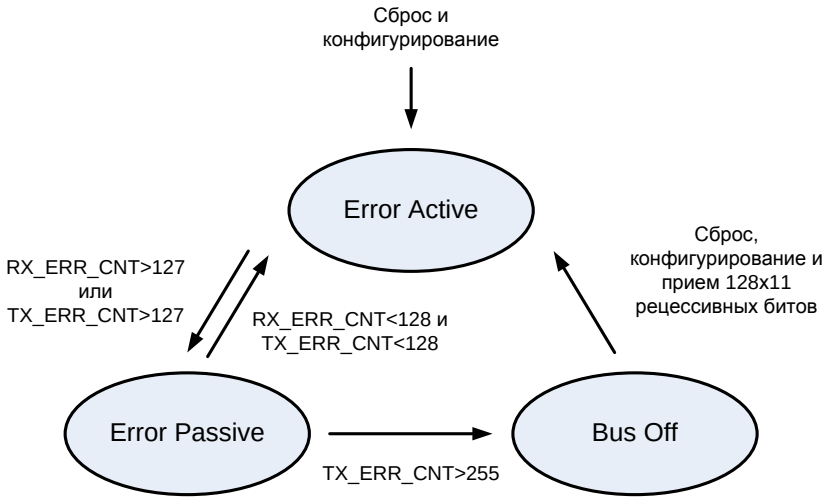
На уровне битов в протоколе CAN реализован также метод вставки бита (битстаффинг). После каждой последовательности из пяти доминантных битов вставляется рецессивный бит; если рецессивный бит не обнаружен, в регистре статуса устанавливается флаг ошибки вставленных битов пакета BIT_STUF_ERR. Этот метод позволяет предотвратить появление на шине постоянных уровней и обеспечивает наличие в потоке битов достаточного количества переходов, используемых для повторной синхронизации. Кадр ошибки в протоколе CAN представляет собой простую последовательность из шести доминантных битов. Это позволяет любому контроллеру CAN формировать на шине сообщение об ошибке сразу после ее обнаружения, не дожидаясь конца сообщения.



Битстаффинг

В каждом CAN контроллере имеется два счетчика. Этими счетчиками являются счетчик ошибок приема (регистр STATUS, поле RX_ERR_CNT) и счетчик ошибок передачи (регистр STATUS, поле TX_ERR_CNT). Изменение состояния этих счетчиков происходит при приеме или передаче кадра ошибки. Когда любой счетчик достигает значения 128, контроллер CAN переходит в режим «error passive». В этом режиме он продолжает отзываться на кадры ошибки, однако при генерации кадра ошибки он вместо доминантных битов выставляет на шину рецессивные. Если счетчик ошибок передачи

достигает значения 255, то контроллер CAN переходит в режим «bus-off» и больше не принимает участия в обмене по шине. Для возобновления обмена необходимо вмешательство процессора, который повторно инициализирует контроллер и подключает его обратно к шине. Текущий статус состояния контроллера можно посмотреть в регистре статуса контроллера STATUS. При успешном приеме/передаче кадров в режиме «error passive» счетчики ошибок RX_ERR_CNT и TX_ERR_CNT декрементируются, и модуль может перейти в режим «error active».



Счетчики ошибок

Контроллер CAN имеет несколько механизмов обнаружения ошибок. Во-первых, из регистра состояния контроллера CAN_STATUS можно считать текущее состояние счетчиков ошибок приема и передачи. Также в этом регистре содержится флаг превышения счетчиками ошибок порогового значения ERROR_OVER. Это значение произвольно и записывается в регистр CAN_OVER. Как и регистры синхронизации, регистр CAN_OVER можно изменять только при нахождении контроллера в состоянии сброса.

Прерывания

В контроллере CAN в качестве источников прерывания выступают буфера сообщений. Генерируемые прерывания делятся на три группы:

- Прерывания передачи (по одному для каждого буфера)
- Прерывания приема (по одному для каждого буфера)
- Прерывания ошибки

При возникновении какого-либо прерывания и наличии сигналов разрешения этих прерываний, буфер вырабатывает прерывание. Контроллер CAN объединяет прерывания приема, передачи и ошибки в каждом буфере и вырабатывает прерывание, отображаемое в регистре прерываний периферии. Если прерывание разрешено в регистре прерываний периферии, процессор выполняет переход на обработчик прерываний. Обработчик прерываний должен выполнить действия по обработке прерывания и снять его выставление. Прерывание передачи/приема для каждого буфера может быть замаскировано путем установления соответствующего бита в регистрах

CAN_INT_TX/CAN_INT_RX. Также есть возможность группового маскирования прерываний по приему, по передаче и по ошибке (см. регистр CAN_INT_EN).

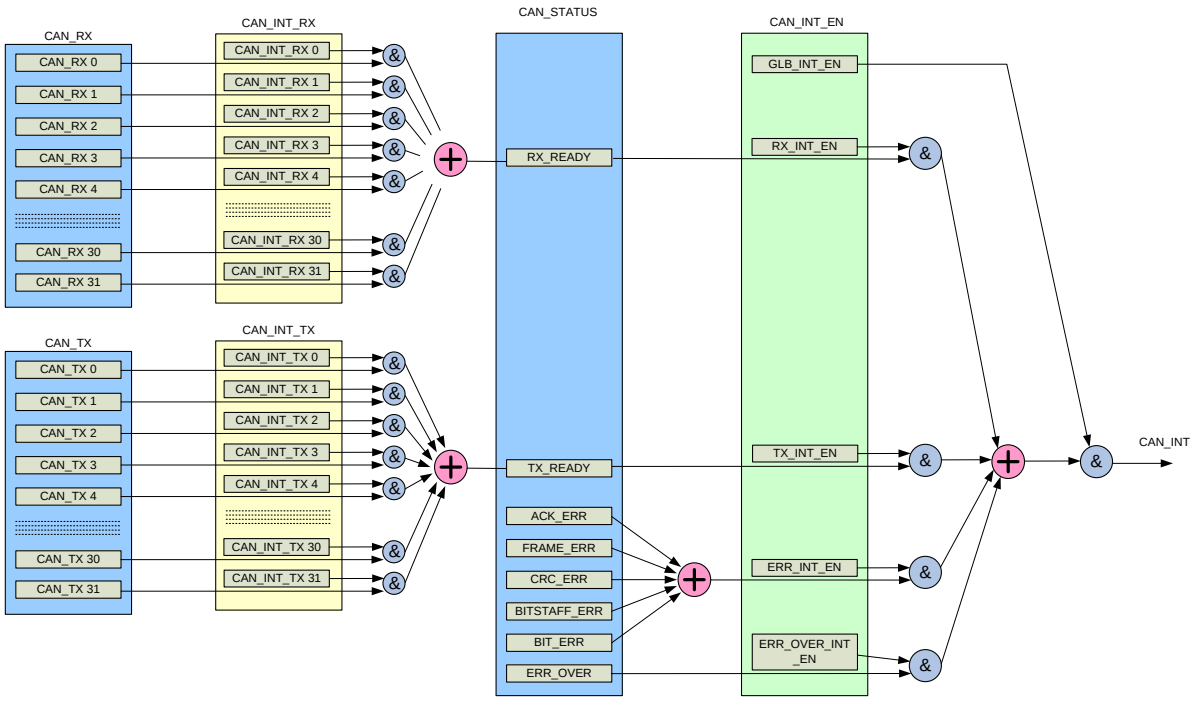


Схема формирования прерывания блока CAN

Описание регистров контроллера CAN

Базовый Адрес	Название	Описание
0x4000_0000	CAN1	Контроллер интерфейса CAN1
0x4000_8000	CAN2	Контроллер интерфейса CAN2
Смещение		
0x00	CAN_CONTROL	Регистр управление контроллером CAN
0x04	CAN_STATUS	Регистр состояния контроллера CAN
0x08	CAN_BITTMNG	Регистр задания скорости работы
0x10	CAN_INT_EN	Регистр разрешения прерываний контроллера
0x1C	CAN_OVER	Регистр границы счетчика ошибок
0x20	CAN_RXID	Регистр принятого ID сообщения
0x24	CAN_RXDLC	Регистр принятого DLC сообщения
0x28	CAN_RXDATAL	Регистр принятых данных
0x2C	CAN_RXDATAH	Регистр принятых данных
0x30	CAN_TXID	Регистр передаваемого ID сообщения

0x34	CAN_TXDLC	Регистр передаваемого DLC сообщения
0x38	CAN_DATAL	Регистр передаваемых данных
0x3C	CAN_DATAH	Регистр передаваемых данных
0x40	CAN_BUF_01_CON	Регистр управления буфером 01
	…	
0xBC	CAN_BUF_32_CON	Регистр управления буфером 32
0xC0	CAN_INT_RX	Флаги разрешения прерываний от приемных буферов
0xC4	CAN_RX	Флаги RX_FULL от приемных буферов
0xC8	CAN_INT_TX	Флаги разрешения прерываний от передающих буферов
0xCC	CAN_TX	Флаги ~TX_REQ от передающих буферов
0x200	BUF_01_ID	ID сообщения буфера 01
0x204	BUF_01_DLC	DLC сообщения буфера 01
0x208	BUF_01_DATAL	Данные сообщения буфера 01
0x20C	BUF_01_DATAH	Данные сообщения буфера 01
0x210	BUF_02_ID	ID сообщения буфера 02
	…	
0x4FC	BUF_32_DATAH	Данные сообщения буфера 32
0x500	BUF_01_MASK	Маска для приема сообщения в буфер 01
0x504	BUF_01_FILTER	Фильтр для приема сообщения в буфер 01
0x508	BUF_02_MASK	Маска для приема сообщения в буфер 02
	…	
0x5FC	BUF_32_FILTER	Фильтр для приема сообщения в буфер 32

CANx_CONTROL

Регистр управления контроллером

Номер	31	5	4	3	2	1	0
Доступ	U	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
	-	-	ROP	SAP	STM	ROM	CAN EN

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..5	-	Зарезервировано
4	ROP	Receive own packets 1 – контроллер принимает собственные пакеты 0 – контроллер принимает только чужие пакеты
3	SAP	Send ACK on own packets 1 – контроллер подтверждает прием собственных пакетов 0 – контроллер подтверждает прием только чужих пакетов
2	STM	Self Test Mode 1 – контроллер работает в режиме самотестирования 0 – контроллер работает в нормальном режиме
1	ROM	Read Only Mode 1 – контроллер работает только на прием 0 – контроллер работает в нормальном режиме
0	CAN_EN	Режим работы контроллера CAN 1 – разрешение работы 0 – сброс

CANx_STATUS

Регистр состояния контроллера

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO
Сброс	0	0	0	0	0	0	0	0
	ACK ERR	FRAME ERR	CRC ERR	BIT STUFF ERR	BIT ERR	ERROR OVER	TX READY	RX READY

Номер	15	14	13	12	11	10	9	8
Доступ	R/W	R/W	R/W	RO	RO	RO	RO	R/W
Сброс	0	0	0	0	0	0	0	0
	-	-	-	TX ERR CNT8	RX ERR CNT8	ERR STATUS[1:0]		ID LOWER

Номер	31	24	24	16
Доступ	RO	RO	RO	RO
Сброс	0	0	0	0
	TX ERR CNT [7:0]		RX ERR CNT [7:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..24	TX ERR CNT [7:0]	Счетчик ошибок передатчика TEC, биты [7:0] TEC > 127, ERROR PASSIVE
23..16	RX ERR CNT [7:0]	Счетчик ошибок приемника REC, биты [7:0] REC > 127, ERROR PASSIVE
15..13	-	
12	TX ERR CNT8	Счетчик ошибок передатчика TEC, бит 8 0 – TEC менее 255 1 – TEC более 255
11	RX ERR CNT8	Счетчик ошибок приемника REC, бит 8 0 – REC менее 255 1 – REC более 255
10…9	ERR STATUS[1:0]	Статус состояния контроллера CAN 00 – ERROR ACTIVE, при возникновении ошибки отсылается флаг активной ошибки 01 – ERROR PASSIVE, при возникновении ошибки отсылается флаг пассивной ошибки 1x – BUS OFF, ожидается восстановление шины
8	ID LOWER	Флаг «проигрыша» арбитража 0 – при передаче не было проигрыша арбитража 1 – при передаче был проигран арбитраж
7	ACK ERR	Флаг ошибки подтверждения приема 0 – нет ошибки

		1 – есть ошибка
6	FRAME ERR	Флаг ошибки формата пакета 0 – нет ошибки 1 – есть ошибка
5	CRC ERR	Флаг ошибки контрольной суммы пакета 0 – нет ошибки 1 – есть ошибка
4	BIT STUFF ERR	Флаг ошибки вставленных битов пакета 0 – нет ошибки 1 – есть ошибка
3	BIT ERR	Флаг ошибки передаваемых битов пакета 0 – нет ошибки 1 – есть ошибка
2	ERROR OVER	Флаг превышения TEC и REC уровня заданного ERROR_MAX 0 – ERROR_MAX < TEC и REC 1 – ERROR_MAX ≥ TEC или REC
1	TX READY	Флаг наличия буферов для отправки 0 – нет буферов готовых для отправки сообщений 1 – есть буфер готовый для отправки сообщений
0	RX READY	Флаг наличия принятых сообщений 0 – нет буферов с принятыми сообщениями 1 – есть буфер с принятым сообщением

CANx_BITTMNG

Регистр задания скорости работы

Номер	31		27	26…25	24…22	21…19	18…16	15…0
Доступ	U		R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0	0
			SB	SJW [1:0]	SEG2 [2:0]	SEG1 [2:0]	PSEG [2:0]	BRP [15:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28	-	Зарезервировано
27	SB	Семплирование: 0 – однократное 1 – трехкратное с мажоритарным контролем
26…25	SJW [1:0]	Значение размера фазы SJW 11 = Synchronization jump width time = 4 x TQ 10 = Synchronization jump width time = 3 x TQ 01 = Synchronization jump width time = 2 x TQ 00 = Synchronization jump width time = 1 x TQ SJW – это максимальное значение, на которое происходит подстройка приема и передачи при работе на шине CAN. Приемник подстраивается на значение ошибки, но не более чем SJW.
24…22	SEG2 [2:0]	Значение размера фазы SEG2 111 = Phase Segment 2 time = 8 x TQ 110 = Phase Segment 2 time = 7 x TQ 101 = Phase Segment 2 time = 6 x TQ 100 = Phase Segment 2 time = 5 x TQ 011 = Phase Segment 2 time = 4 x TQ 010 = Phase Segment 2 time = 3 x TQ 001 = Phase Segment 2 time = 2 x TQ 000 = Phase Segment 2 time = 1 x TQ SEG2 – это время, используемое для сокращения битового интервала при подстройке.
21…19	SEG1 [2:0]	Значение размера фазы SEG1 111 = Phase Segment 1 time = 8 x TQ 110 = Phase Segment 1 time = 7 x TQ 101 = Phase Segment 1 time = 6 x TQ 100 = Phase Segment 1 time = 5 x TQ 011 = Phase Segment 1 time = 4 x TQ 010 = Phase Segment 1 time = 3 x TQ 001 = Phase Segment 1 time = 2 x TQ 000 = Phase Segment 1 time = 1 x TQ SEG1 – это время, используемое для увеличения битового интервала при подстройке.
18…16	PSEG[2:0]	Значение размера фазы PSEG 111 = Propagation time = 8 x TQ 110 = Propagation time = 7 x TQ 101 = Propagation time = 6 x TQ

		100 = Propagation time = 5 x TQ 011 = Propagation time = 4 x TQ 010 = Propagation time = 3 x TQ 001 = Propagation time = 2 x TQ 000 = Propagation time = 1 x TQ PSEG - это время, компенсирующее задержку распространения сигналов в шине CAN
15…0	BRP [15:0]	Предделитель системной частоты CLK = PCLK/(BRP + 1); TQ(us) = (BRP + 1)/CLK(MHz)

CANx_INT_EN

Регистр разрешения прерываний

Номер	31		5	4	3	2	1	0
Доступ	U		U	U	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0	0
	-		-	ERR OVER INT EN	ERR INT EN	TX INT EN	RX INT EN	GLB INT EN

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..5	-	Зарезервировано
4	ERR OVER INT EN	Флаг разрешения прерывания по превышению TEC или REC допустимого значения в ERROR_MAX 0 – запрещено прерывание 1 – разрешено прерывание
3	ERR INT EN	Флаг разрешения прерывания по возникновению ошибки 0 – запрещено прерывание 1 – разрешено прерывание
2	TX INT EN	Флаг разрешения прерывания по возможности передачи 0 – запрещено прерывание 1 – разрешено прерывание
1	RX INT EN	Флаг разрешения прерывания по приему сообщений 0 – запрещено прерывание 1 – разрешено прерывание
0	GLB INT EN	Общий флаг разрешения прерывания блока CAN 0 – запрещено прерывание 1 – разрешено прерывание

CANx_OVER

Регистр границы счета ошибок

Номер	31	9	8	7	2	1	0
Доступ	U	U	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0

-		-	-	ERROR_MAX[7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7…0	ERROR MAX [7:0]	Регистр границы счетчика ошибок Допустимое значение счетчиков ошибок TEC и REC, при превышении которого вырабатывается флаг ERROR_OVER

CANx_BUF_xx_CON

Регистр управления буфером

Номер	31…8	7	6	5	4	3	2	1	0
Доступ	U	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	0

-	OVER WR	RX FULL	TX REQ	PRIOR 0	RTR EN	OVER EN	RX TXn	EN

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..8	-	Зарезервировано
7	OVER_WR	Флаг перезаписи принятого сообщения 0 – не было перезаписи 1 – была перезапись принятого сообщения
6	RX_FULL	Флаг готовности приема 0 – нет принятого сообщения 1 – принятое сообщение в буфере
5	TX_REQ	Запрос на отправку сообщения 0 – нет запроса или отправлено 1 – запрос на отправку
4	PRIOR_0	Приоритет при отправке 0 – нет приоритета 1 – приоритет
3	RTR_EN	Режим ответа на RTR 1 – ответить при приеме RTR в буфер 0 – не отвечать при приеме RTR
2	OVER_EN	Разрешение перезаписи принятого сообщения 1 – разрешена перезапись сообщения 0 – не разрешена перезапись
1	RX_TXn	Режим работы буфера 1 – на прием

		0 – не передачу
0	EN	Разрешение работы буфера 1 – работает 0 – отключен

CANx_INT_RX

Регистр разрешения прерываний от приемных буферов

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
	CAN_INT_RX[31:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	CAN_INT_RX [31:0]	Флаги разрешения прерываний от буферов по приму сообщений CAN_INT_RX[0] – для первого буфера CAN_INT_RX[1] – для второго буфера и так далее

CANx_RX

Регистр флагов RX_FULL от приемных буферов

Номер	31	0
Доступ	RO	RO
Сброс	0	0
	CAN_RX[31:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	CAN_RX[31:0]	Флаги RX_FULL разрешенных на прием буферов CAN_RX[0] – флаг RX_FULL от первого буфера CAN_RX[1] – флаг RX_FULL от второго буфера и так далее, доступны только на чтение

CANx_INT_TX

Регистр разрешения прерываний от передающих буферов

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
	CAN_INT_TX[31:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	CAN_INT_TX [31:0]	Флаги разрешения прерываний от буферов по передаче сообщений CAN_INT_TX[0] – для первого буфера CAN_INT_TX[1] – для второго буфера и так далее

CANx_TX

Регистр флагов ~TX_REQ от передающих буферов

Номер	31	0
Доступ	RO	RO
Сброс	0	0
	CAN_TX[31:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	CAN_TX[31:0]	Флаги ~TX_REQ разрешенных на передачу буферов CAN_TX[0] – флаг ~TX_REQ от первого буфера CAN_TX[1] – флаг ~TX_REQ от второго буфера и так далее, доступны только на чтение

CANx_RXID

CANx_TXID

CANx_BUF_xx_ID

Регистры ID сообщения:

Номер	31					29	28…18	17…0
Доступ	U					U	R/W	R/W
Сброс	0					0	0	0
	-					-	SID [10:0]	EID [17:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..29	-	Зарезервировано
28…18	SID [10:0]	Поле SID Для стандартного и расширенного пакетов CAN Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже.
17…0	EID [17:0]	Поле EID Для расширенных пакетов CAN Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже.

CANx_RXDLC

CANx_TXDLC

CANx_BUF_xx_DLC

Регистры DLC сообщения:

Номер	31…13	12	11	10	9	8	7…4	3…0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	-	IDE	SSR	R0	R1	RTR	-	DLC [3:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..13	-	Зарезервировано
12	IDE	Поле IDE Поле обозначающее формат пакета 1 – расширенный пакет 0 – стандартный пакет
11	SSR	Поле SSR, расширенного формата Всегда должен быть “1”
10	R0	Поле R0 Всегда должен быть “0”
9	R1	Поле R1, расширенного формата Всегда должен быть “1”
8	RTR	Поле RTR, запроса обратного ответа 0 – нет запроса 1 – есть запрос Если узел получил пакет с запросом обратного ответа, он должен ответить.
7…4	-	Зарезервировано
3…0	DLC[3:0]	Поле DLC, длина передаваемых данных в пакете 0000 – нет данных 0001 – 1 байт 0010 – 2 байт 0011 – 3 байт 0100 – 4 байт 0101 – 5 байт 0110 – 6 байт 0111 – 7 байт 1000 – 8 байт 1xxx – недопустимо.

CANx_RXDATAL

CANx_TXDATAL

CANx_BUF_xx_DATAL

Регистры данных сообщения:

Номер	31…24	23…16	15…8	7…0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	DB3[7:0]	DB2[7:0]	DB1[7:0]	DB0[7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…24	DB3[7:0]	Поле DB3, четвертый байт, передаваемый в пакете
23…16	DB2[7:0]	Поле DB2, третий байт, передаваемый в пакете
15…8	DB1[7:0]	Поле DB1, второй байт, передаваемый в пакете
7…0	DB0[7:0]	Поле DB0, первый байт, передаваемый в пакете

CANx_RXDATAH

CANx_TXDATAH

CANx_BUF_xx_DATAH

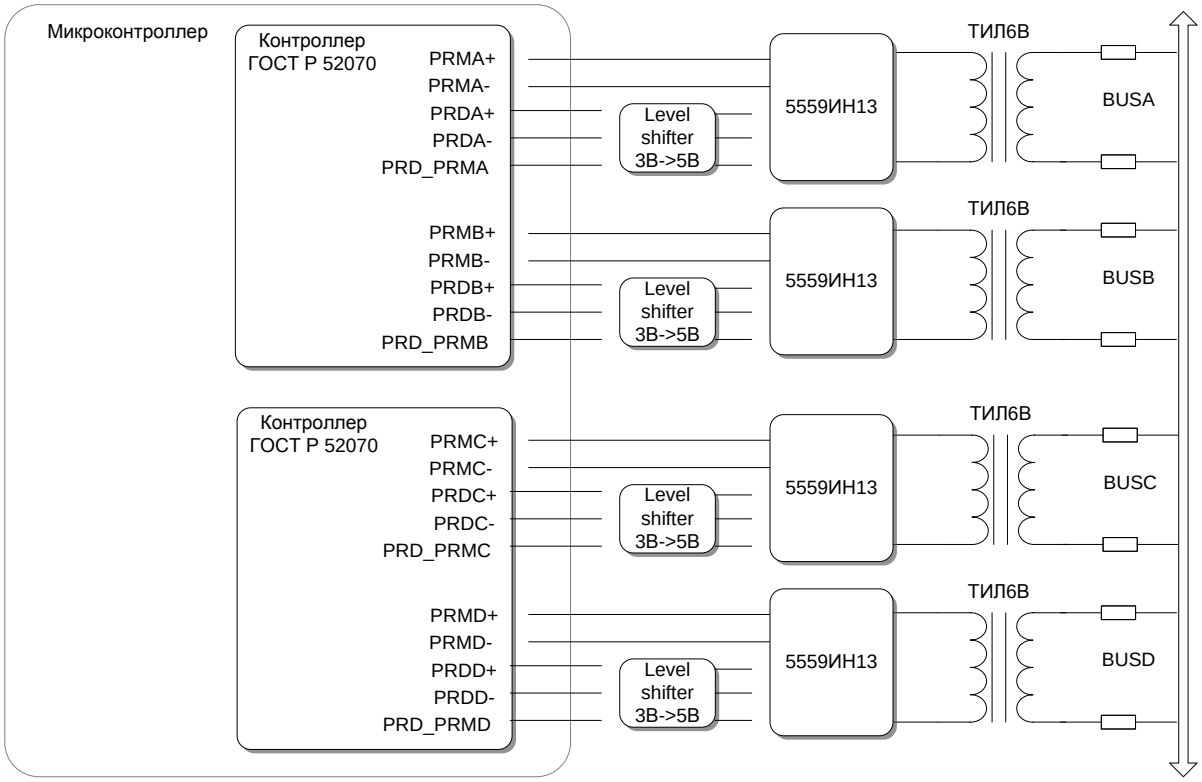
Регистры данных сообщения:

Номер	31…24	23…16	15…8	7…0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	DB7[7:0]	DB6[7:0]	DB5[7:0]	DB4[7:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…24	DB7[7:0]	Поле DB7, восьмой байт, передаваемый в пакете
23…16	DB6[7:0]	Поле DB6, седьмой байт, передаваемый в пакете
15…8	DB5[7:0]	Поле DB5, шестой байт, передаваемый в пакете
7…0	DB4[7:0]	Поле DB4, пятый байт, передаваемый в пакете

Контроллер интерфейса по ГОСТ Р52070-2003

В микроконтроллере имеется два независимых контроллера интерфейса 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 двухпортовая память, в которой данные хранятся в области памяти соответствующей подадресу командного слова. В каждом подадресе 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.



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 1Мбит/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

Режимы работы

Контроллер поддерживает три режима работы: контроллера шина (КШ), оконечного устройства (ОУ) и неадресуемого монитора (М).

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.

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.

Монитор

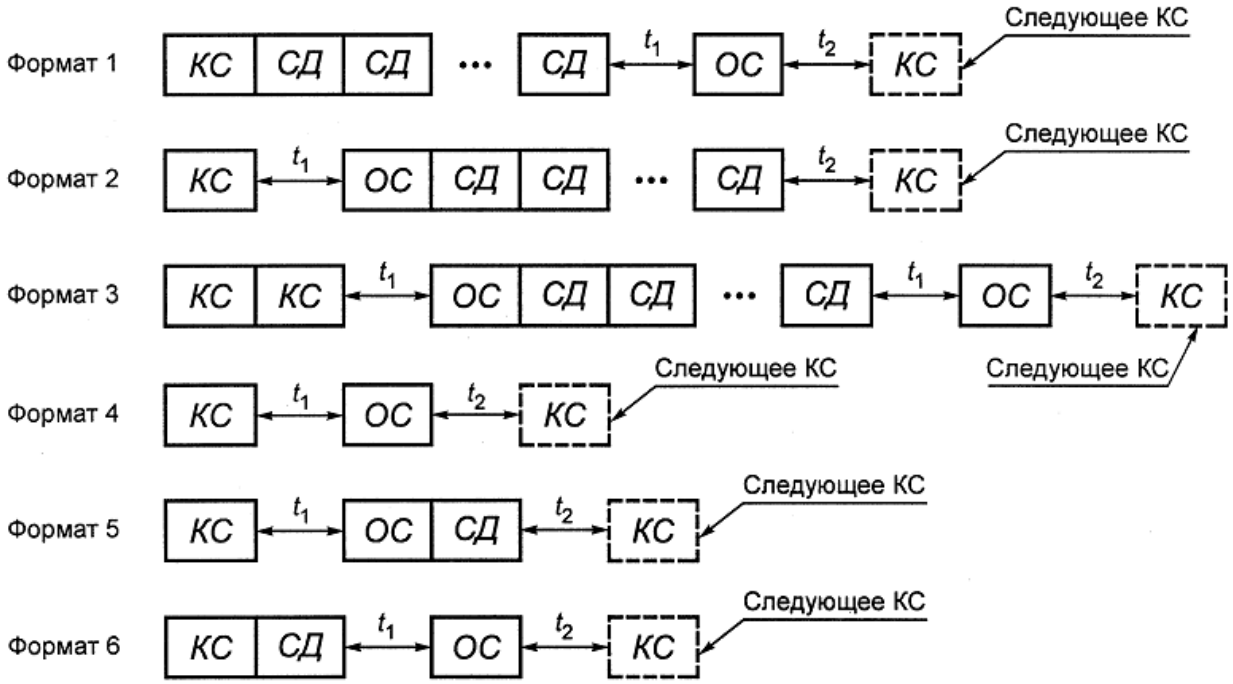
В этом режиме осуществляется прослушивание магистрали и отбор необходимой информации для проведения: технического обслуживания, регистрации эксплуатационных параметров, анализа решаемых задач или обеспечения информацией резервного КШ. Монитор пассивно прослушивает выбранную шину и захватывает весь трафик 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.

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.



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

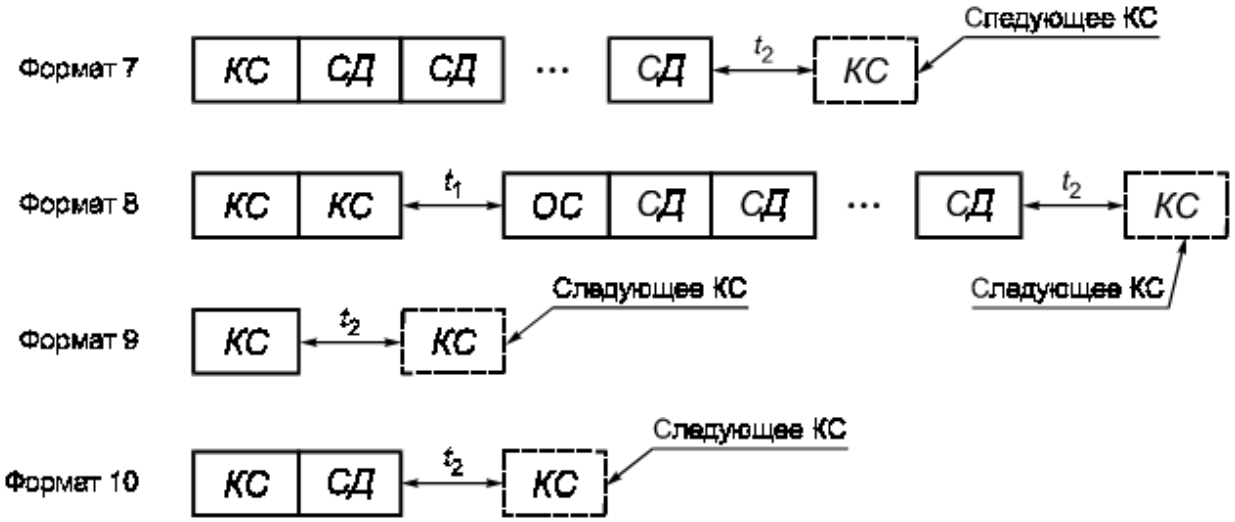
Формат 6 – передача КУ и СД оконечному устройству

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

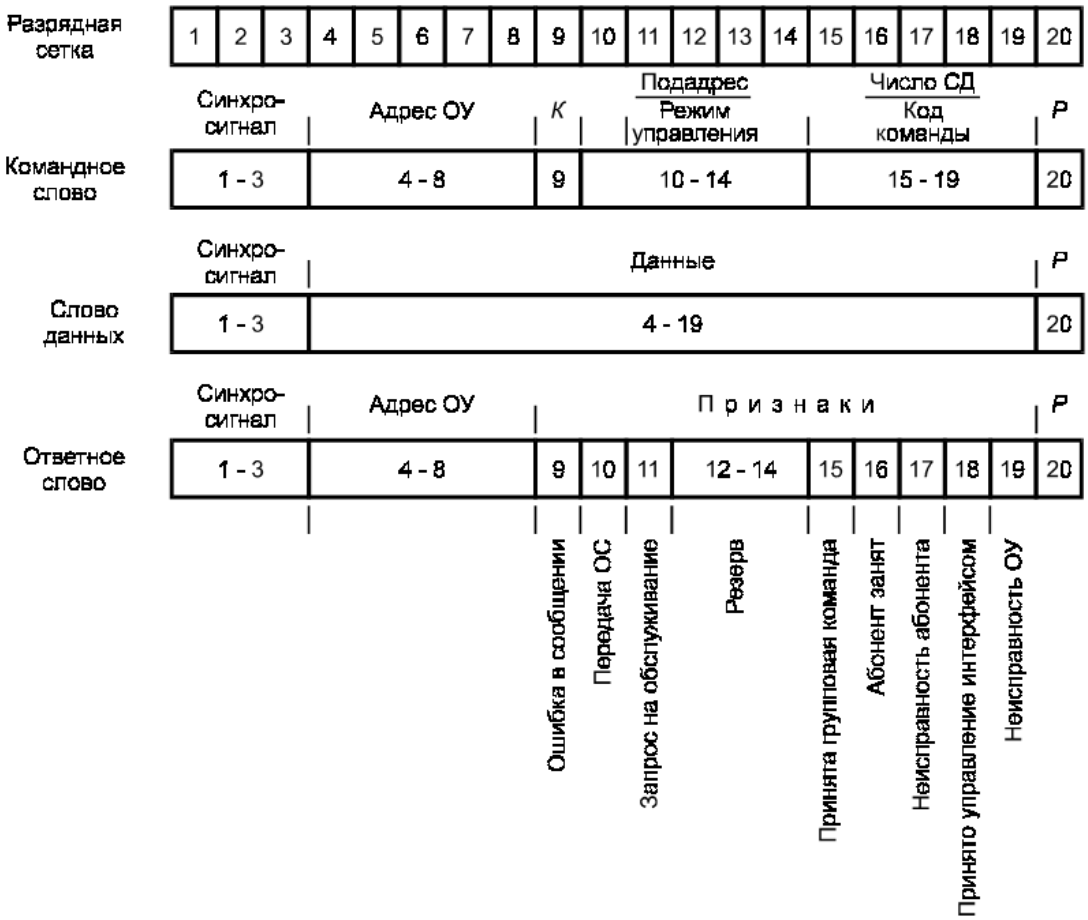
Формат 9 – передача групповой команды управления
Формат 10 – передача групповой команды управления со словом данных



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.

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.



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 вторая 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.

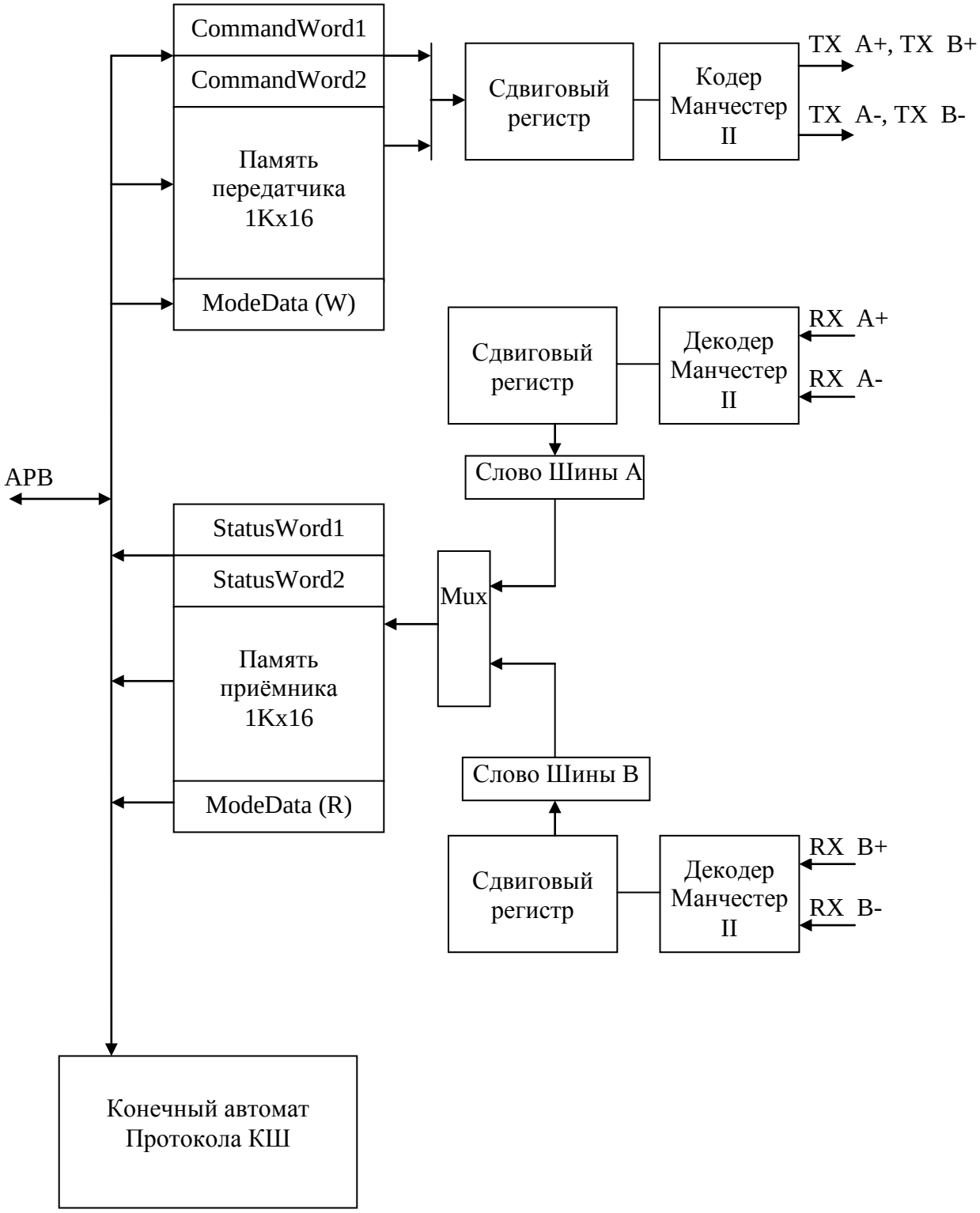
Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала отрицательная, а вторая положительная.

Поле данных содержит передаваемые данные, а разряд контроля по чётности формируется так же, как в командном слове.

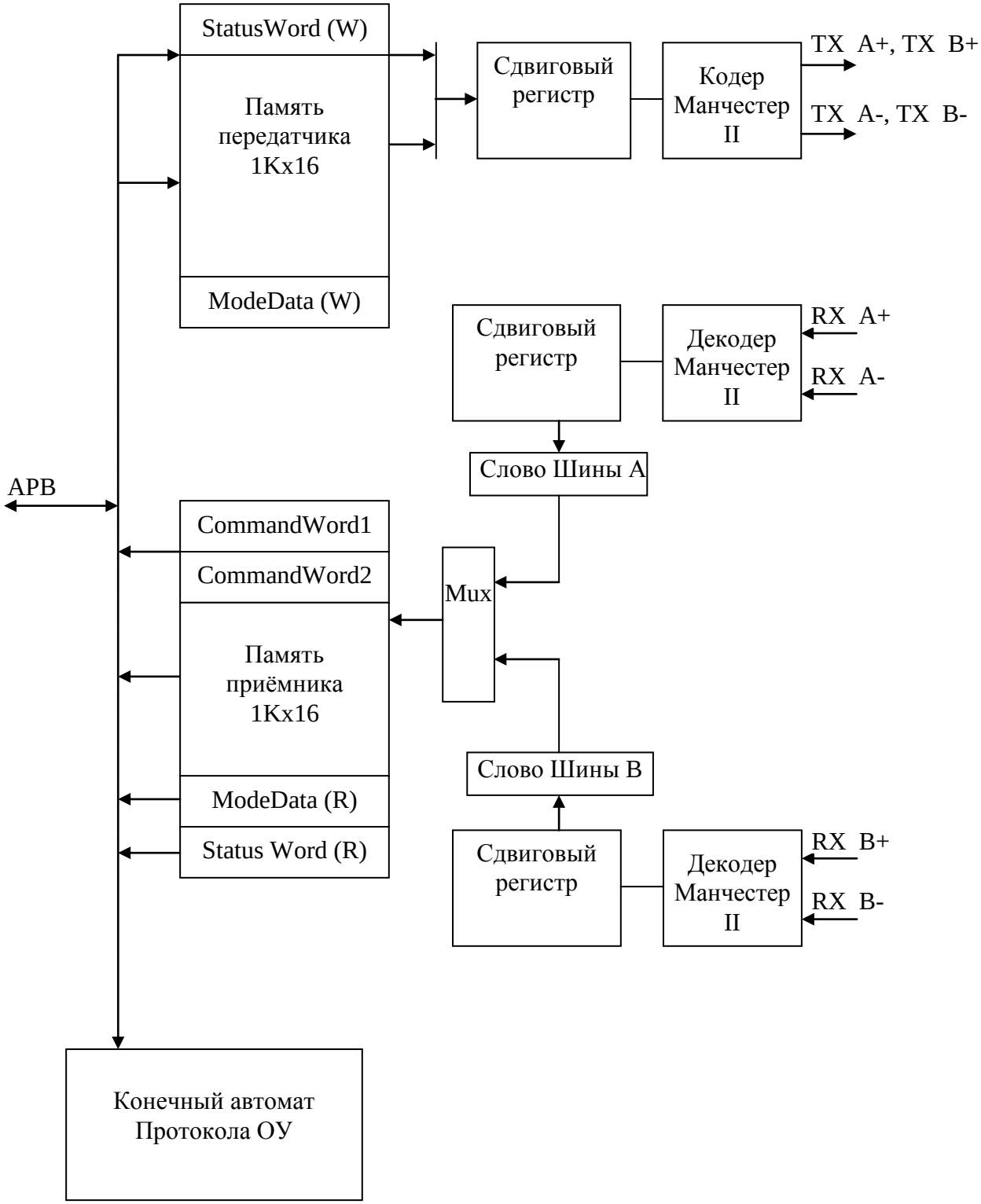
Ответное слово содержит:
- синхросигнал;
- поле «Адрес ОУ»;
- поле разрядов признаков состояния: ошибка в сообщении, передача ОС, запрос на обслуживание, принята групповая команда, абонент занят, неисправность абонента, принято управление интерфейсом, неисправность ОУ;
- разряд контроля по четности.

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.

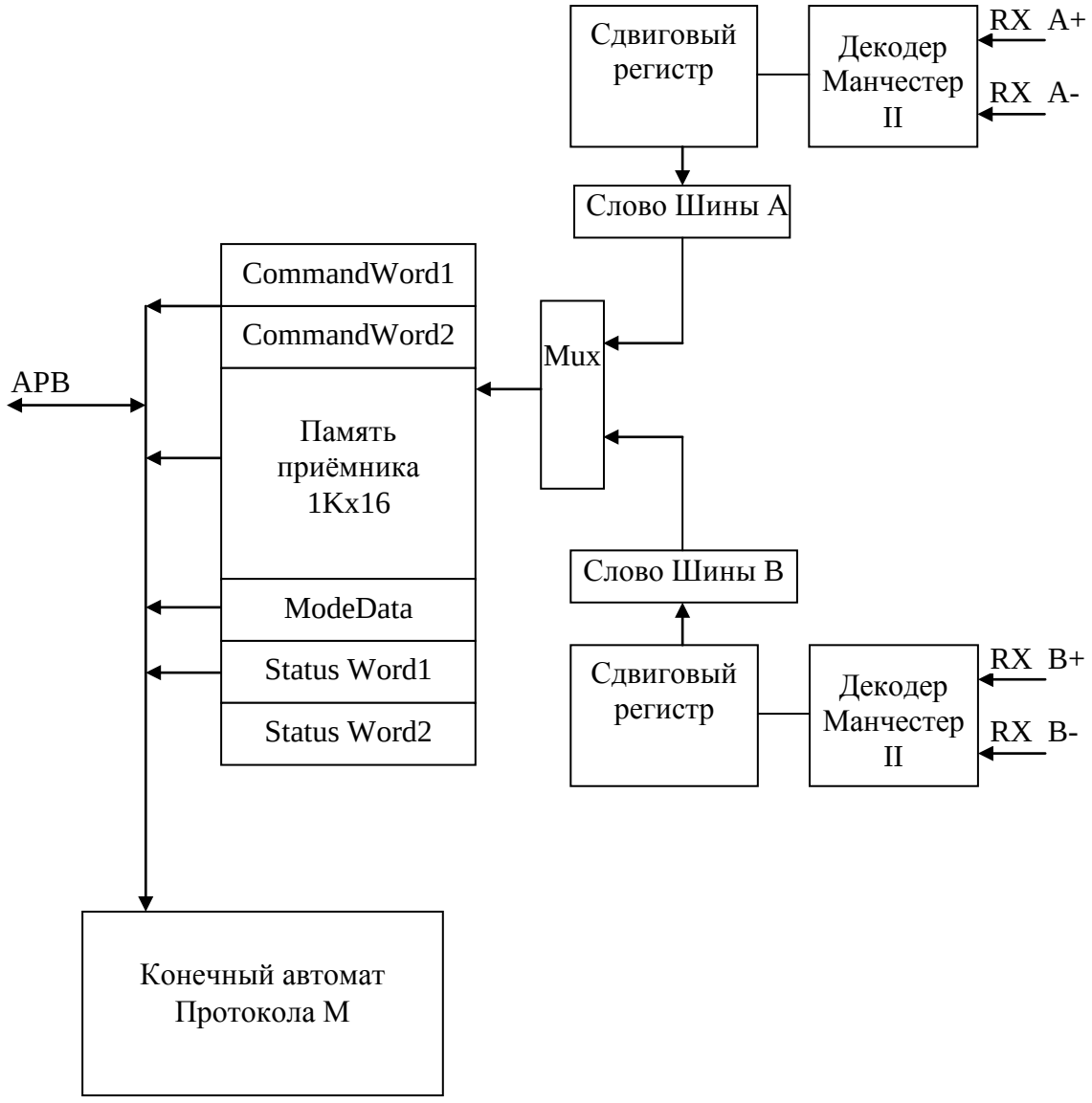
Структурная схема в режиме КШ



Структурная схема в режиме ОУ



Структурная схема в режиме М



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. Одновременная установка двух этих бит запрещена.

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 40МГц/DIV=1МГц.

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-х 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.

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 нужно 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 командно 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 либо установкой битом MR, либо инициированием новой транзакции битом BCSTART.

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.

Описание регистров контроллера ГОСТ Р52070-2003

Базовый Адрес	Название	Описание
0x4004_8000	MIL-STD-1553B1	Контроллер интерфейса 1553 канал 1
0x4005_0000	MIL-STD-1553B2	Контроллер интерфейса 1553 канал 2
Смещение		
0x000-0xFFC	DATA	Память принимаемых/передаваемых СД
0x1000	CONTROL	Регистр управление контроллером
0x1004	STATUS	Регистр состояния контроллера

0x1008	ERROR	Регистр ошибок контроллера
0x100C	CommandWord1	Регистр командного слова 1
0x1010	CommandWord2	Регистр командного слова 2
0x1014	ModeData	Слово данных команды управления
0x1018	StatusWord1	Регистр ответного слова 1
0x101C	StatusWord2	Регистр ответного слова 2
0x1020	INTEN	Регистр разрешения прерываний
0x1024	MSG	Регистр декодирования сообщений

CONTROL

Регистр управления

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
Значение после сброса	0	0	0	0	0	0	0	0
	RTA1	RTA0	TRB	TRA	RTMODE	BCMODE	BCSTART	MR

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
Значение после сброса	0	0	0	0	0	0	0	0
	DIV4	DIV3	DIV2	DIV1	DIV0	RTA4	RTA3	RTA2

Номер	31	22	21	20	19	18	17	16
Доступ*	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса			0	0	0	0	0	0
	-		AUTOTUNE	ENFILTER	INVTR	RERR	DIV6	DIV5

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	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 3)** 0 – автоподстройка разрешена 1- автоподстройка запрещена
20	ENFILTER	**Бит разрешения фильтрации потока NRZ (с ревизии 3)** 1 - фильтрация разрешена 0 - фильрация запрещена В случае применения драйверов с неккоректной скважностью и длительностью импульсов NRZ кода,

		таких как 5559ИН67Т, необходимо устанавливать бит в единицу для корректного приёма. В этом случае контроль длительностей импульсов NRZ не осуществляется.
19	INVTR	**Разрешение инверсии сигналов (с ревизии 3)** PRD_PRMA, PRD_PRMB, PRD_PRMC, PRD_PRMD 1- инверсия 0 - прямой выход
18	RERR	**Сброс ошибок в режиме ОУ** 1 – ошибки могут быть сброшены только битом MR 0 – сброс ошибок происходит автоматически, после установки бита IDLE
17..11	DIV6-DIV0	**Делитель частоты MAN_CLK до 1 МГц** Содержит значение, на которое необходимо поделить частоту MAN_CLK, чтобы получить 1 МГц. Частота MAN_CLK обязательно должна быть не более 120 МГц и кратна 8. Если MAN_CLK не кратна 8, то DIV[6:3]=(MAN_CLK/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

STATUS

Регистр состояния

Номер	7	6	5	4	3	2	1	0
Доступ*	U	U	U	U	RO	RO	RO	RO
Значение после сброса					0	0	0	1
	-	RCVB	RCVA	ERR	VALMESS	RFLAGN	IDLE	

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	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

ERROR

Регистр ошибок

Номер	6	5	4	3	2	1	0
Доступ *	RO	RO	RO	RO	RO	RO	RO
Значение после сброса	0	0	0	0	0	0	0

PROERR	CONERR	GAPERR	CSYCERR/ SEQERR	DSYCERR/ SYNCERR	MANERR	NORCV

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..7		Зарезервировано
6	PROERR	**Ошибка в протоколе.** 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 и М (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
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

CommandWord1

Регистр команды 1

Номер	15	11	10	9	5	4	0
Доступ *	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0

Адрес ОУ	Приём/ Передача	Подадрес / Режим управления	Число СД / Код команды

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	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.

CommandWord2

Регистр команды 2

Номер	15	11	10	9	5	4	0
Доступ *	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0		0

Адрес ОУ	1	Подадрес / Режим управления	Число СД / Код команды

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	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.

ModeData

Слово данных команды управления

Номер	15	0
Доступ *	R/W	R/W
Значение после сброса	0	0

Слово данных команды управления

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	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

StatusWord1

Ответное слово 1

Номер	15	10	9	8	4	3	2	1	0
Доступ *	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Значение после сброса	0	0	0	0	0	0	0	0	0

Адрес ОУ	Ошибка в сообщ.	Пер.ОС	Запр. на обсл.	-	Прин. ГК	Абон зан.	Неисп. абон.	Прин упр. инт.	Неисп ОУ

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	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

Примечание: В режиме ОУ регистр по записи содержит предназначенное для передачи КШ ответное слово, а по чтению содержит ОС, полученное от передающего ОУ в транзакции ОУ-ОУ. В режиме КШ и М регистр доступен только на чтение и содержит принятое от ОУ ответное слово.

StatusWord2

Ответное слово 2

Номер	15	10	9	8	4	3	2	1	0
Доступ *	RO	RO	RO	RO	RO	RO	RO		RO
Значение после сброса	0	0	0	0	0	0	0	0	0

Адрес ОУ	Ошибка в сообщ.	Пер. ОС	Запр. на обсл.	-	Прин. ГК	Абон зан.	Неисп. абон.	Прин упр.	Неисп ОУ

								инт.	

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	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

Примечание: В режиме КШ и М регистр доступен только на чтение и содержит ОС передающего ОУ в транзакции ОУ-ОУ. В режиме ОУ регистр не используется.

INTEN

Регистр разрешения прерываний

Номер	7	6	5	4	3	2	1	0
Доступ*	U	U	U	U	R/W	R/W	R/W	R/W
Значение после сброса					0	0	0	0

-	ERRIE	VALMESSIE	RFLAGNIE	IDLEIE

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	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

MSG

Регистр декодирования сообщений

Номер	15	14	13		0
Доступ*	U	U	RO		RO
Значение после сброса			0		0

-	-	Код сообщения

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

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.

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 К=1 нет данных, не групповая	RTA	1	00000	11111 0XXXX				
0400 Код 0-15 К=1 нет данных, групповая	11111	1	00000	11111 0XXXX				
2420 Код 16-31 К=1 с данными, не групповая	RTA	1	00000	11111 1XXXX				
0040 Код 16-31 К=0 с данными, не групповая	RTA	0	00000	11111 1XXXX				
0800 Код 16-31 К=0 с данными, групповая	1111	0	00000	11111 1XXXX				

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.

Таймеры общего назначения

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 четырех таймеров микроконтроллера содержит 32-битный счетчик, 16-битный предделитель частоты и 4-канальный блок захвата/сравнения. Их можно синхронизировать системной синхронизацией, 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;

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 восьмью регистрами совпадения. Кроме того, в микроконтроллере 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 на рисунке

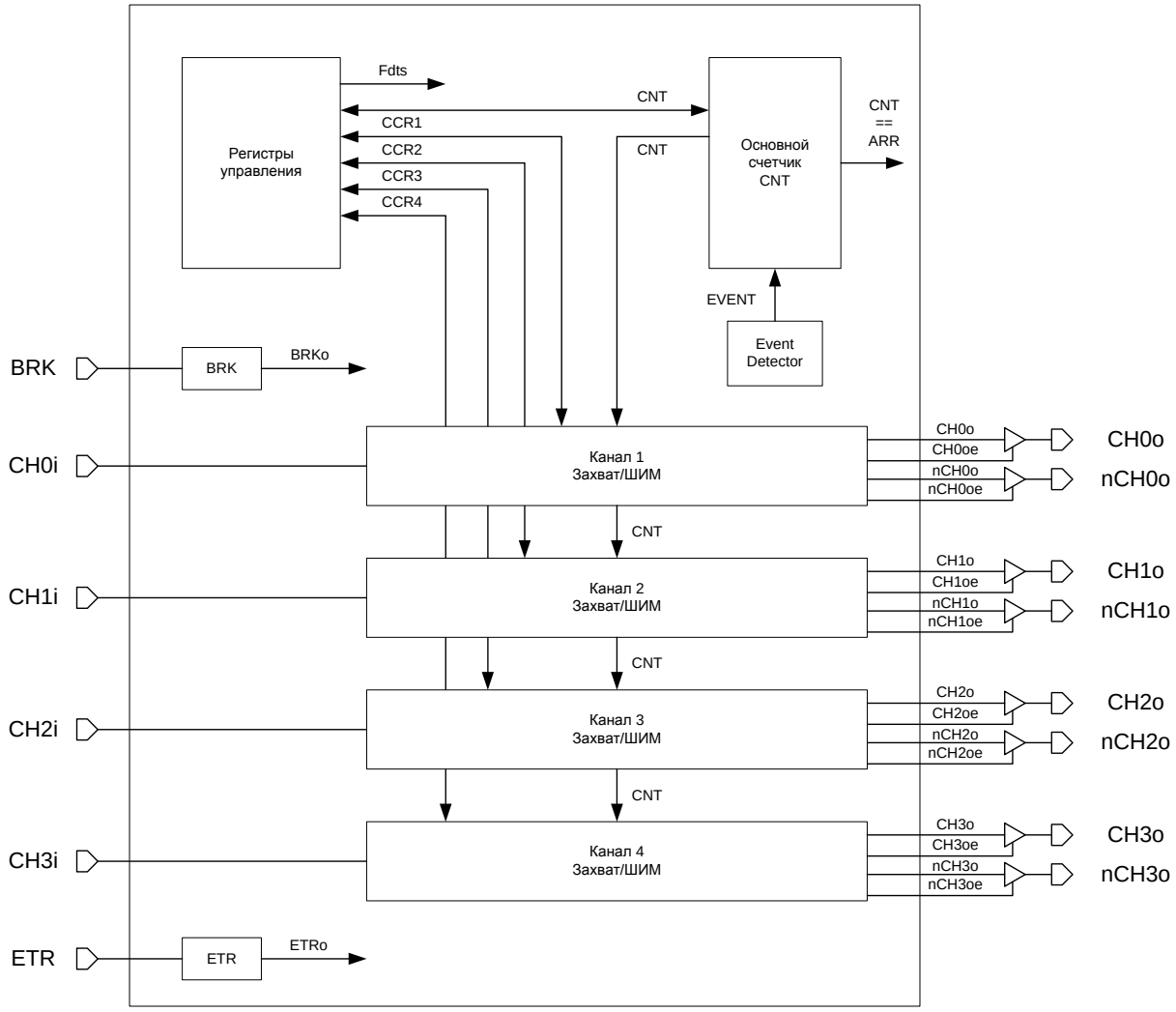


Рисунок Структурная схема таймера

Таймер содержит основной 32-х 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.

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 14 для таймера 1, бит 15 для таймера 2, бит 16 для таймера3, бит 19 для таймера 4 регистра PER_CLOCK). В регистре TIM_CLOCK установить бит TIMyCLKEN, чтобы разрешить тактирование определенного таймера, задать коэффициент деления тактовой частоты HCLK для каждого таймера.
После подачи тактового сигнала на блок таймера можно приступать к работе с ним.

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 (TIMx_CNT);
- делитель частоты при счете основного счетчика (TIMx_PSC);
- основание счета основного счетчика (TIMx_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 – TIMx_CNT;
- значение предварительного делителя счетчика – TIMx_PSG, при этом основной счетчик будет считать на частоте CLK= TIMx_CLK/(PSG+1);
- значение основания счета для основного счетчика TIMx_ARR;
- режим работы счетчика TIMx_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 запроса DMA, которые могут быть замаскированны.

Режимы счета

Счет вверх: CNT_MODE = 00, DIR = 0

```
TIMx->TIMx_CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000004;//Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000;//Предделитель частоты
TIMx->TIMx_ARR = 0x00000013;//Основание счета
```

//Разрешение работы таймера.
TIMx->TIMx_CNTRL = 0x00000001;//Счет вверх по TIM_CLK.

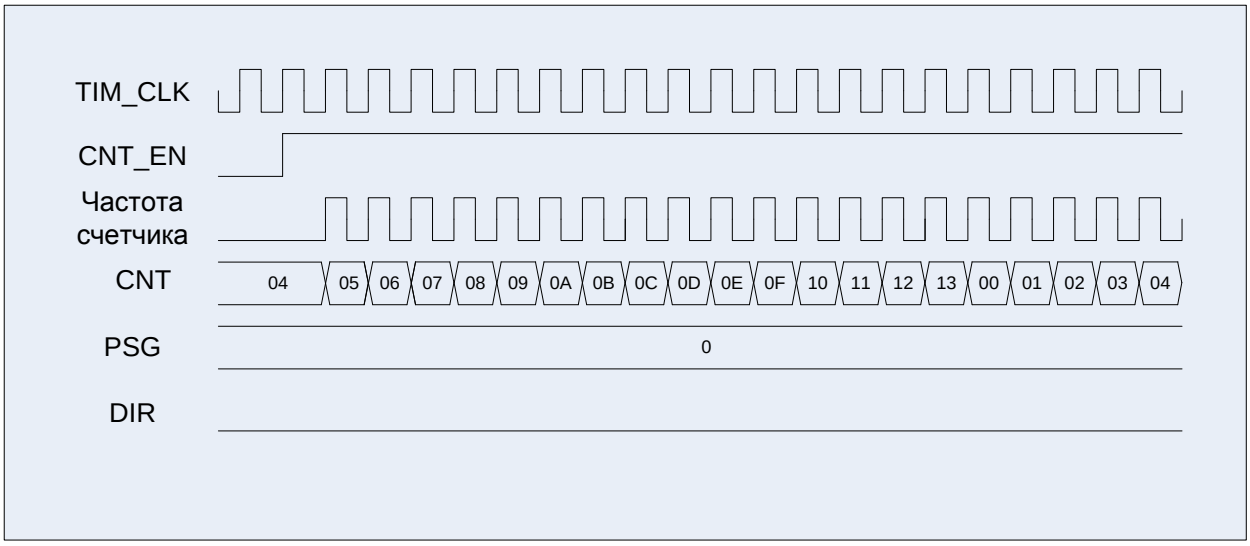


Рисунок Диаграммы работы таймера, счет вверх от 0 до 0x0013, стартовое значение 0x0004

Счет вниз: CNT_MODE = 00, DIR = 1
TIMx->TIMx_CNTRL = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000004;//Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000;//Предделитель частоты
TIMx->TIMx_ARR = 0x00000013;//Основание счета

//Разрешение работы таймера.
TIMx->TIMx_CNTRL = 0x00000009;//Счет вниз по TIM_CLK.

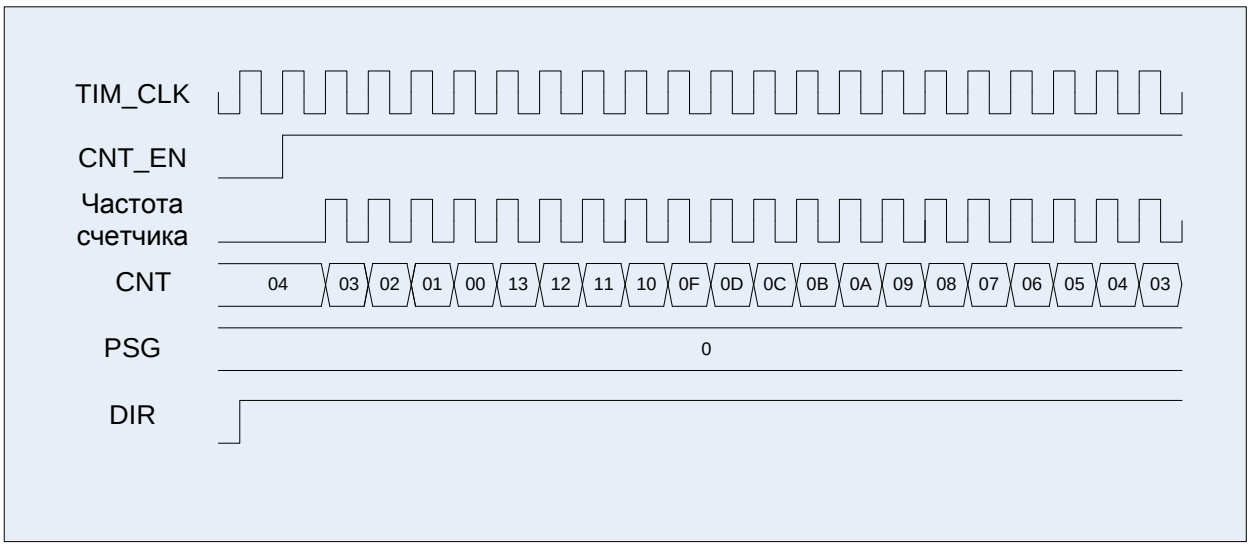


Рисунок Диаграммы работы таймера, счет вниз от 0x0013 до 0, стартовое значение 0x0004

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

TIMx->TIMx_CNTRL = 0x00000000;//Режим инициализации таймера

```
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000004;//Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000;//Предделитель частоты
TIMx->TIMx_ARR = 0x00000013;//Основание счета

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

Рисунок Диаграммы работы таймера, счет вверх/вниз, сначала вверх.

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

```
TIMx->TIMx_CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000004;//Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000;//Предделитель частоты
TIMx->TIMx_ARR = 0x00000013;//Основание счета

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

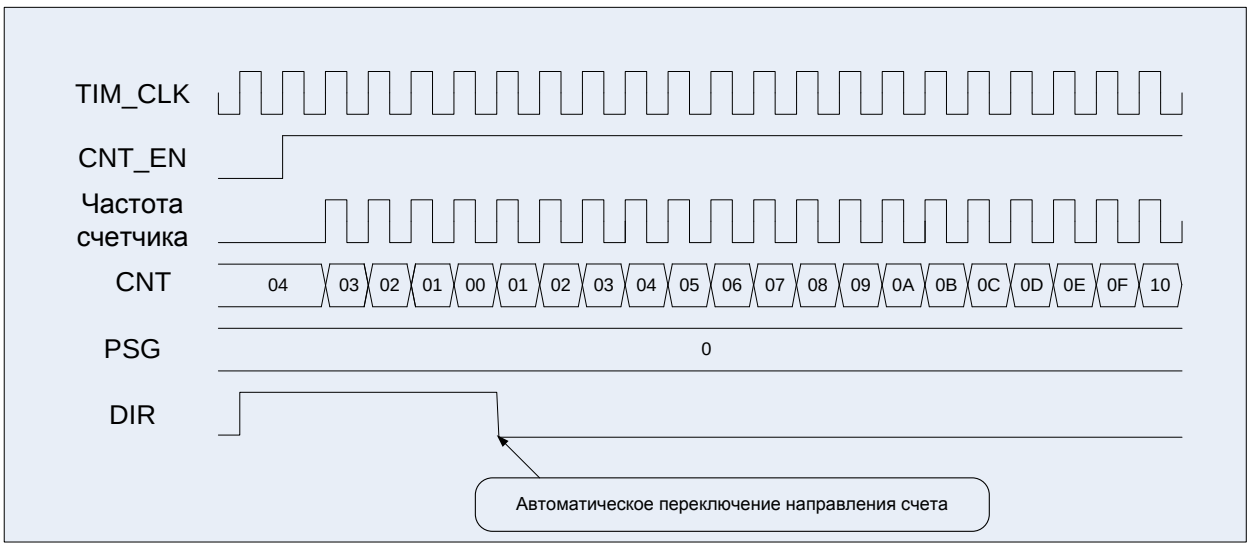


Рисунок Диаграммы работы таймера, счет вверх/вниз, сначала вниз.

Источник 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 линиях TxCHO данного счетчика.
Внешний тактовый сигнал режим 3: События на входе ETR данного счетчика.

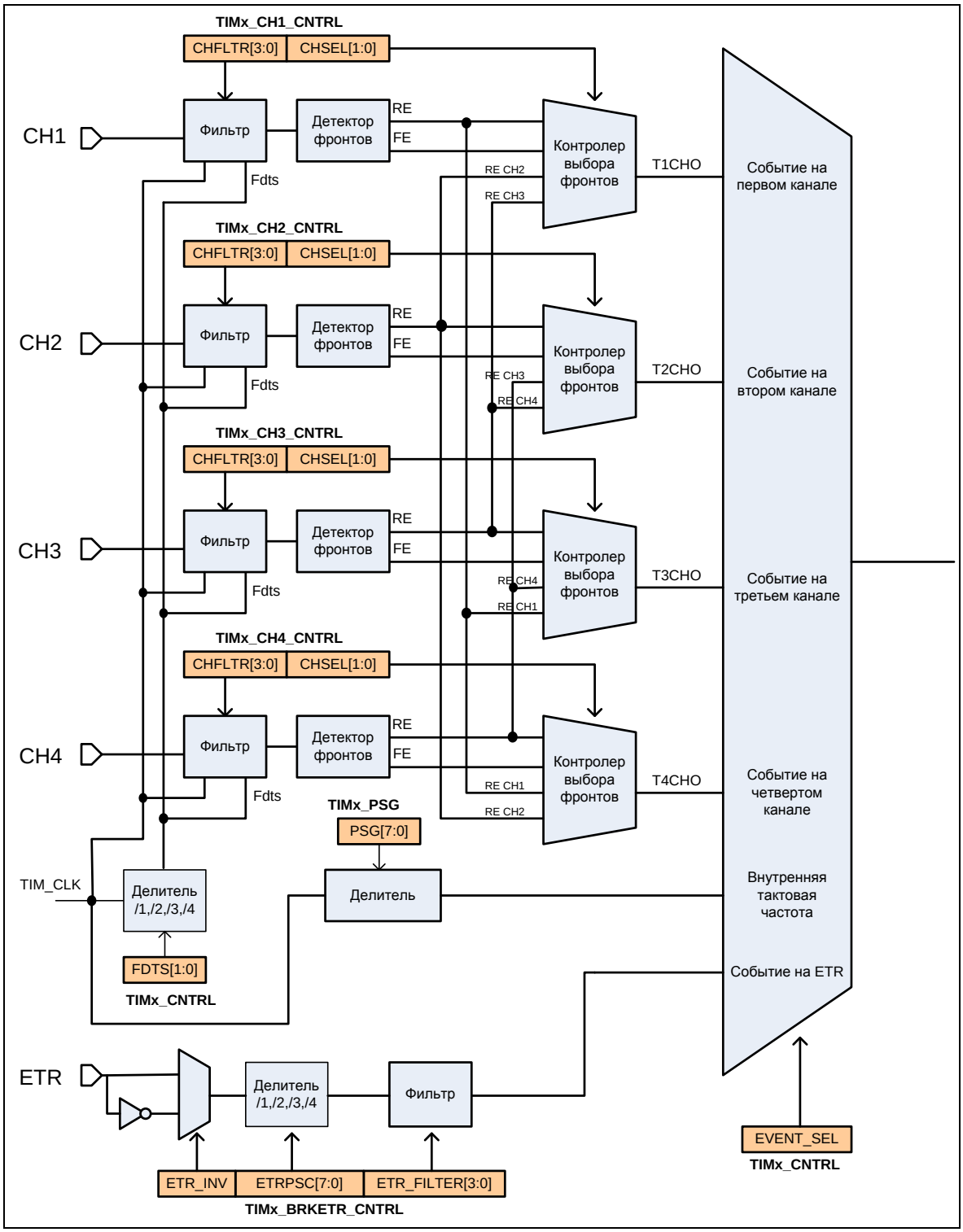


Рисунок Структурная схема формирования события для счета

Внутренний тактовый сигнал (TIM_CLK)

Этот режим выбирается, когда CNT_MODE = 0x, EVENT_SEL = 0000. Для запуска этого режима необходимо задать начальное значение основного счетчика, значение предварительного делителя основного счетчика, основание счета для основного счетчика и задать режим работы в регистре TIMx_CNTRL. Значения регистров TIMx_CNT, TIMx_PSG и TIMx_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 основание счета (TIMx_ARR) может вступить в силу мгновенно после записи его в регистр при условии установленного поля ARRB_EN = 1 (регистр TIMx_CNTRL). Если значение предварительного делителя основного счетчика неравно 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 (см. рис.), DIR = 1 – счетчик считает вниз (см. рис.). Если CNT_MODE = 00, то направление счета определяется полем DIR, если CNT_MODE = 01, счетчик считает вверх/вниз с автоматическим изменением DIR (см. рис.).

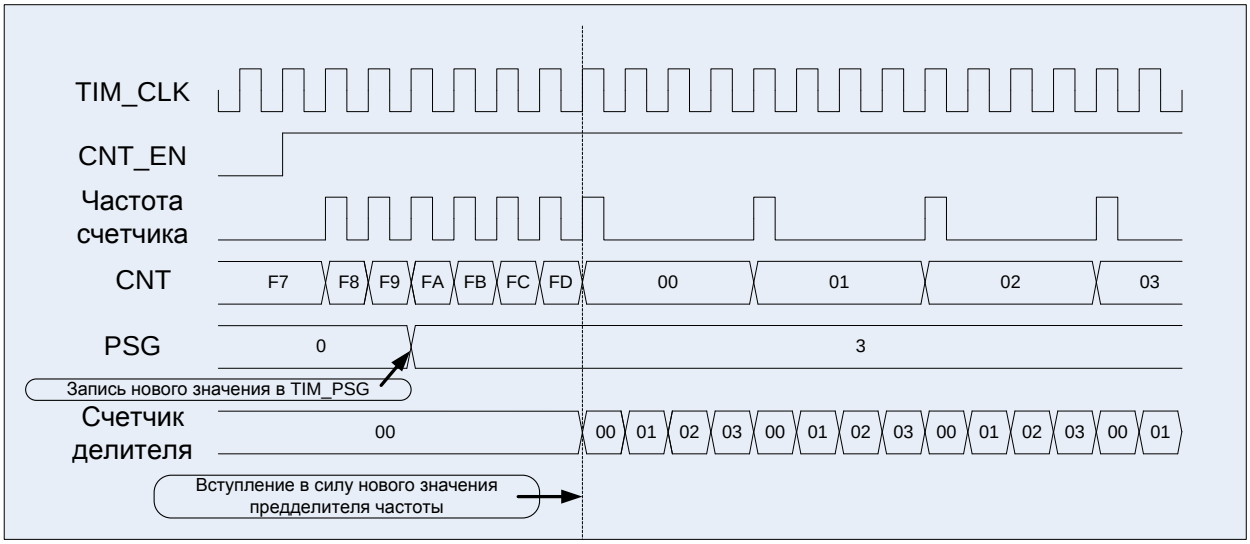


Рисунок Диаграммы работы счетчика: счет вверх (CNT_MODE = 00, EVENT_SEL = 0000, DIR = 0)

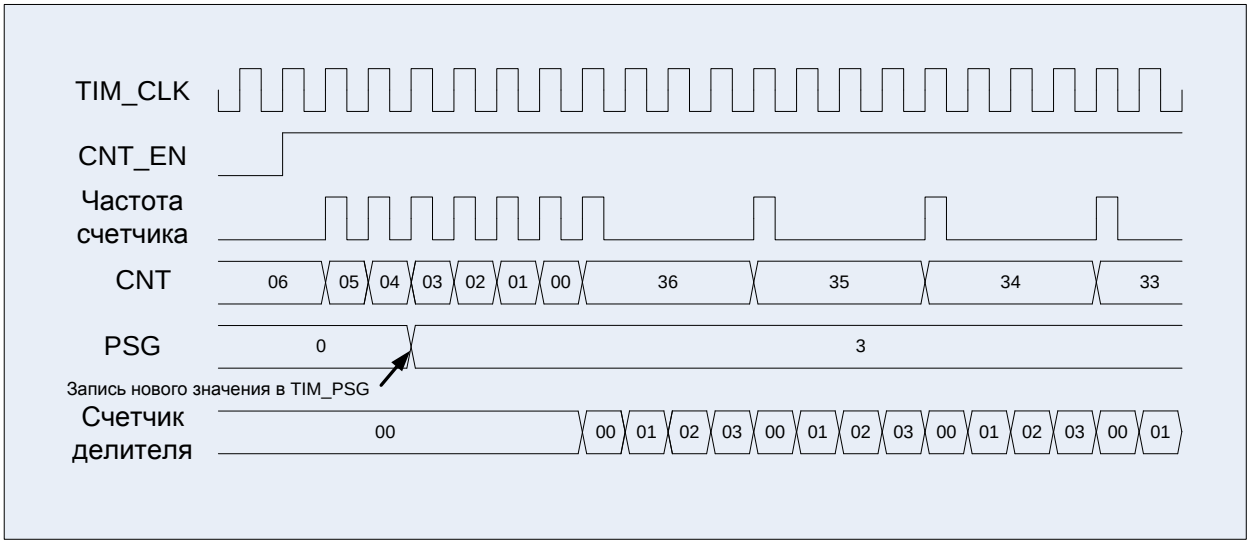


Рисунок Диаграммы работы счетчика: счет вниз
(CNT_MODE = 00, EVENT_SEL = 0000, DIR = 1).

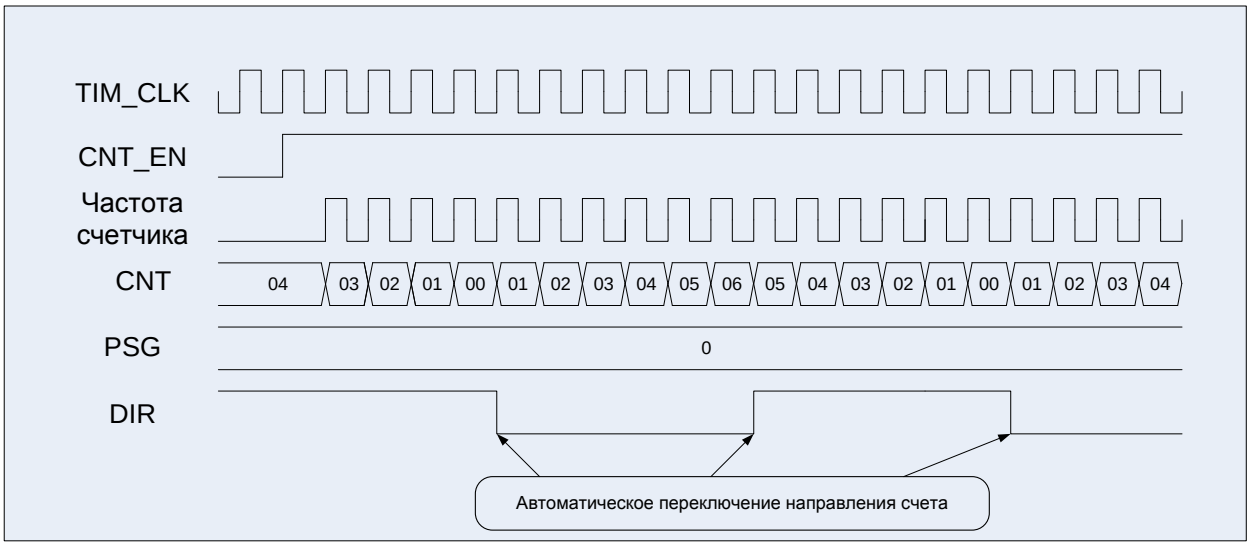


Рисунок Диаграммы работы счетчика: счет вниз/вверх
(CNT_MODE = 01, EVENT_SEL = 0000, DIR = 1).

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.

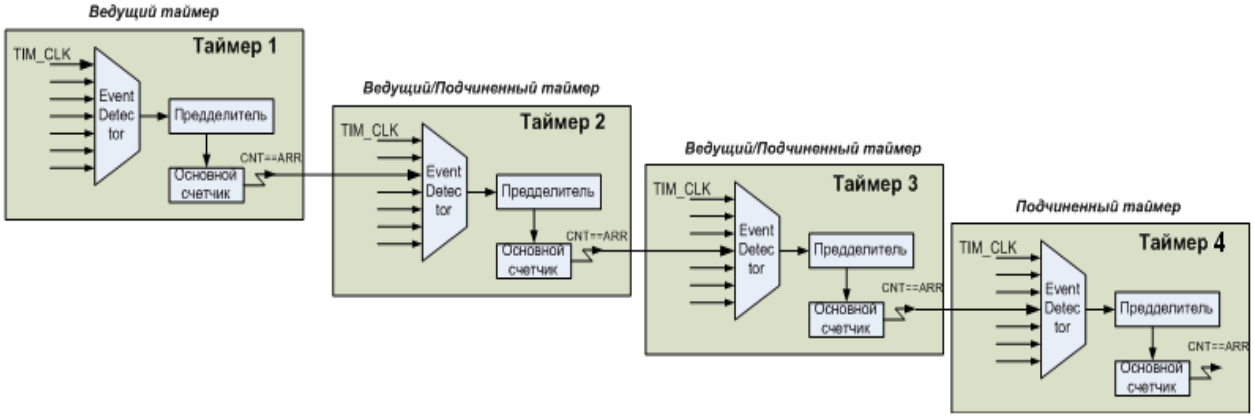


Рисунок Пример каскадного соединения таймеров

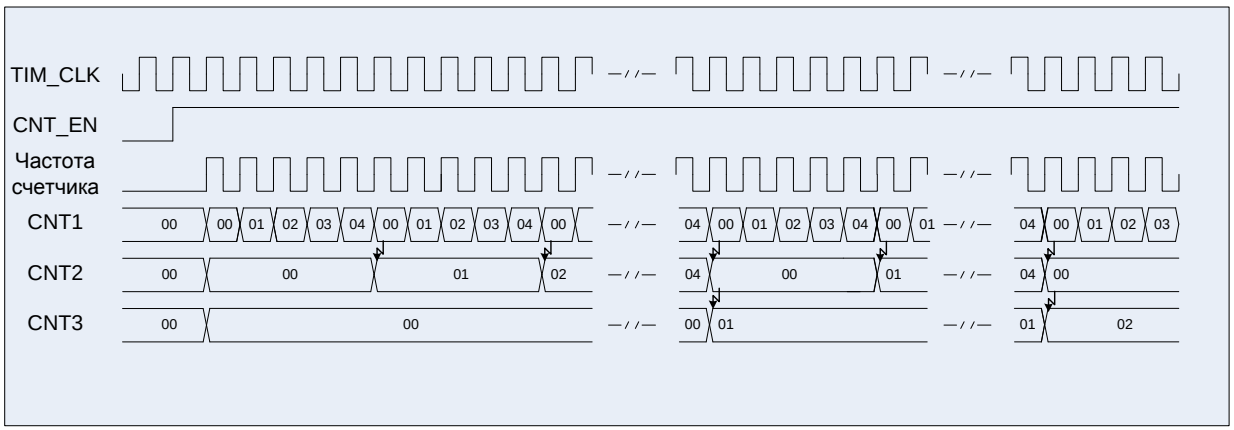


Рисунок Диаграммы работы трех таймеров в каскаде

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;

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

Этот режим выбирается, когда EVENT_SEL = 01xx в регистре TIMx_CNTRL. Счетчик может считать по положительному фронту или по отрицательному фронту на выбранном входе или по положительному фронту на других каналах (см. рис.). 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 TIMx_CNTRL в поле FDTS.

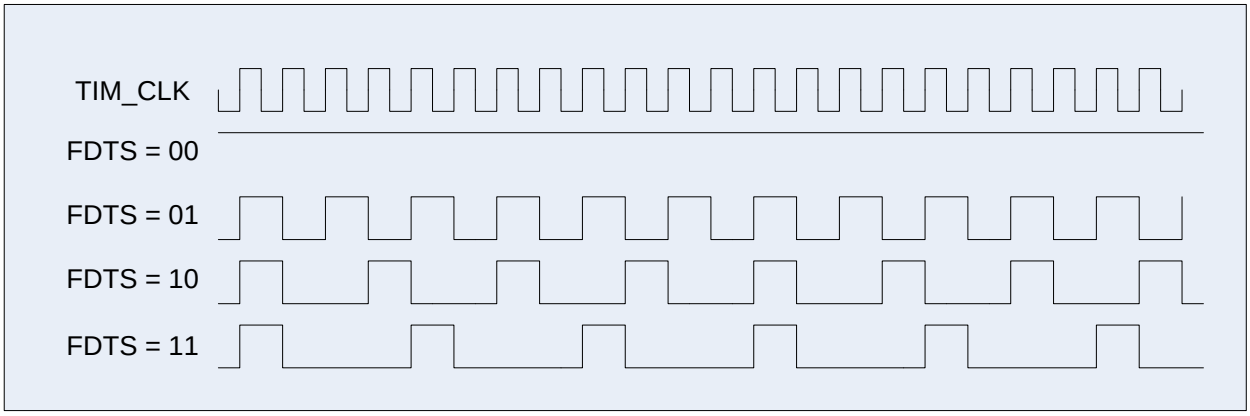


Рисунок Диаграммы возможных частот семплирования данных (FDTS)

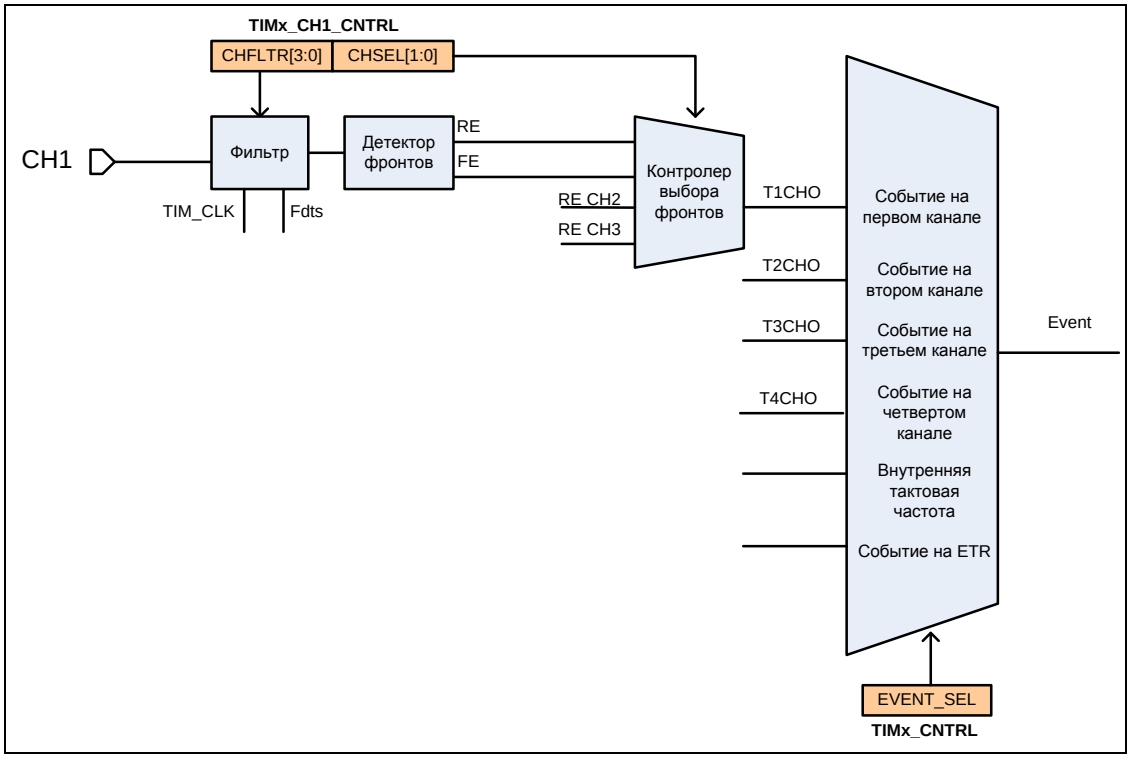


Рисунок Тактирование с входа первого канала

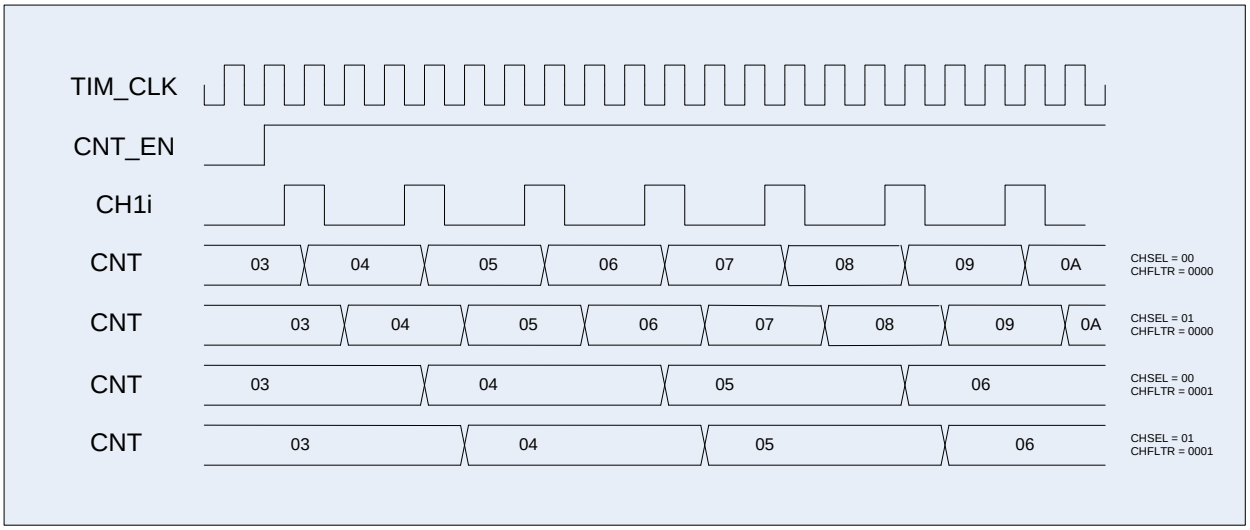


Рисунок Диаграмма внешнего тактирования с разными вариантами фильтра

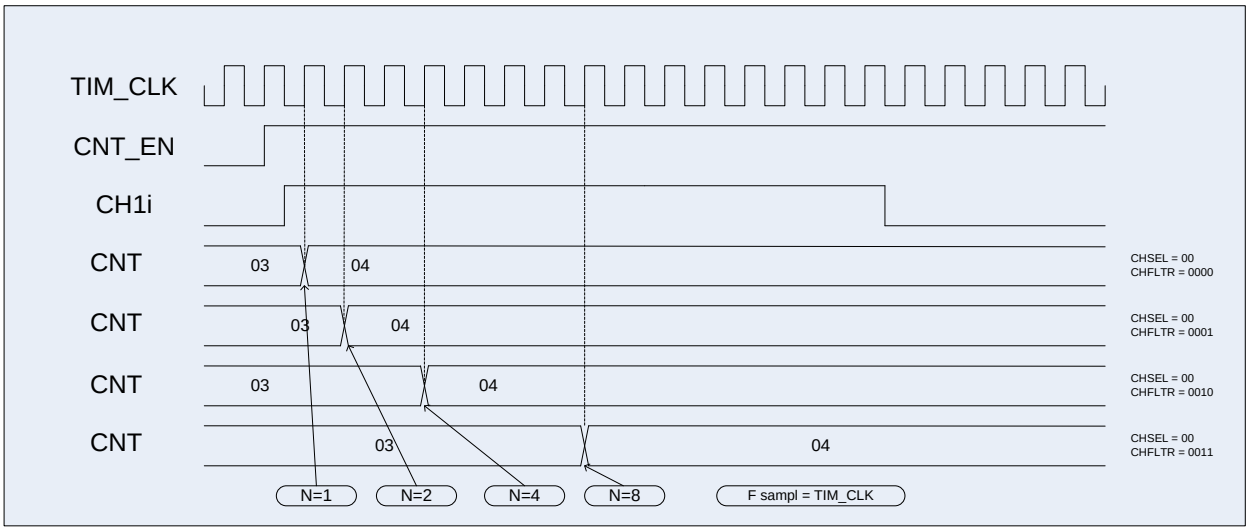


Рисунок Диаграмма внешнего тактирования с разными вариантами фильтра

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

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

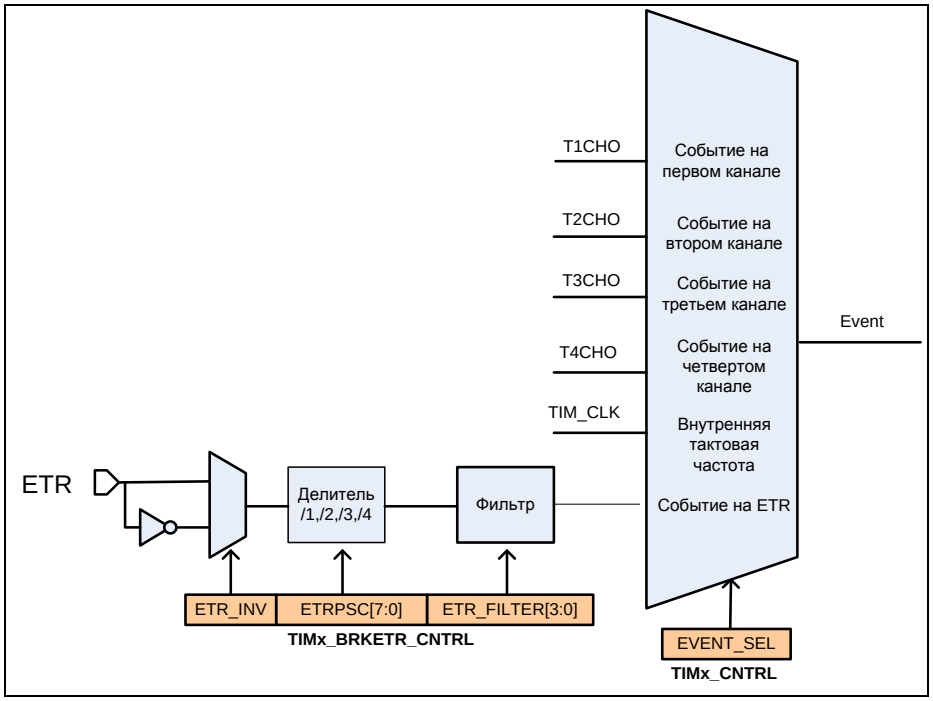


Рисунок Схема тактирования сигналом с входа ETR

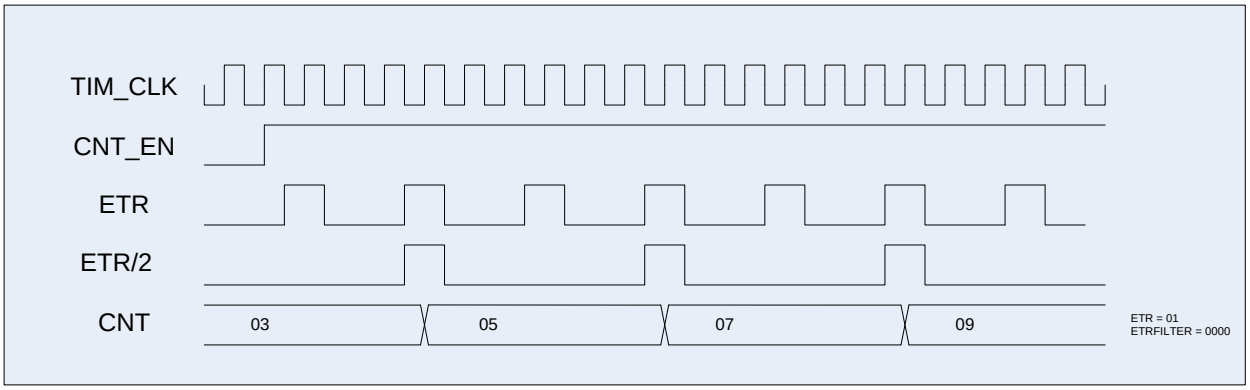


Рисунок Диаграмма тактирования сигналом с входа ETR.

Режим захвата

Структурная схема блока Захвата представлена на рис

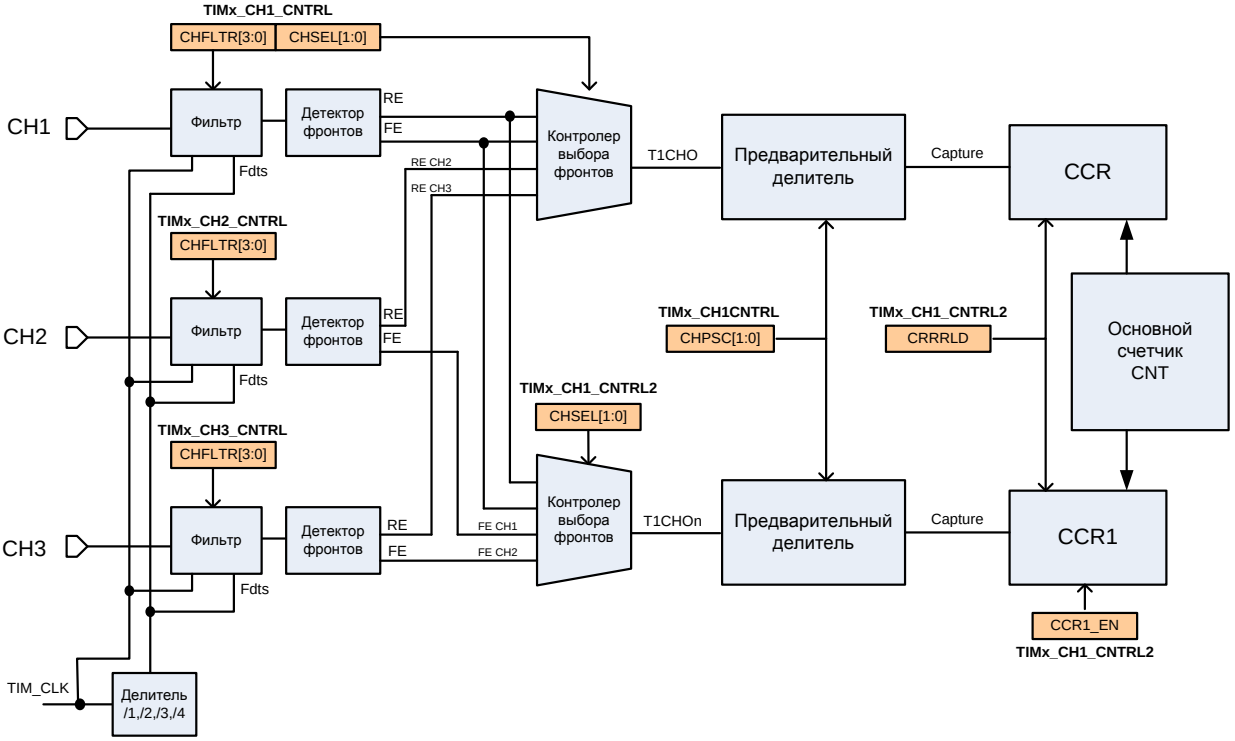


Рисунок Структурная схема блока захвата на примере канала 1.

Для включения режима захвата для определенного канала необходимо в регистре управления каналом TIMx_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.

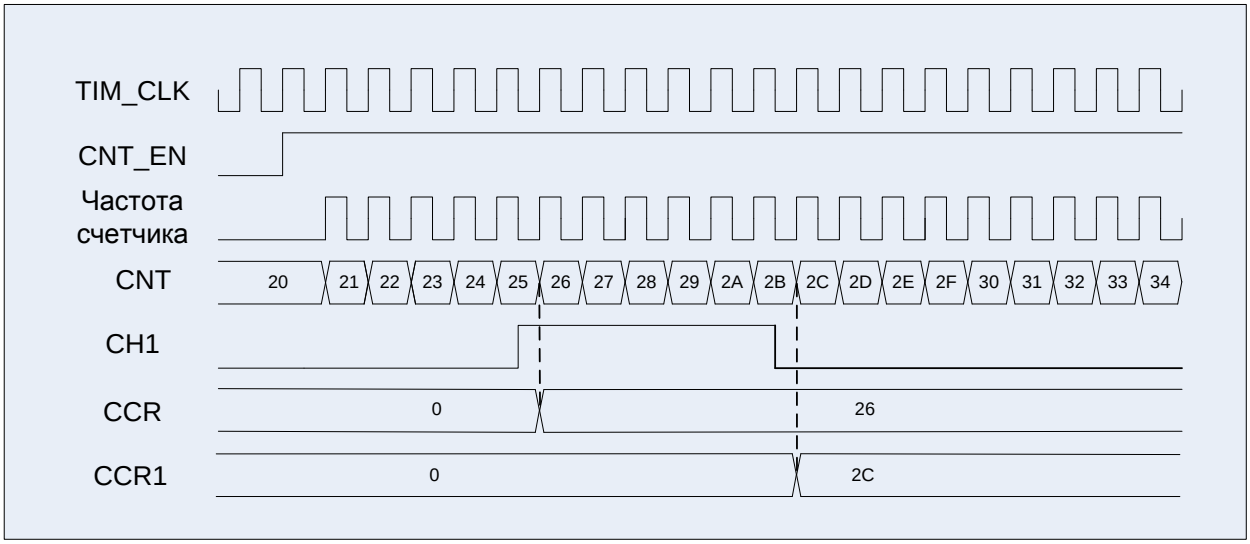


Рисунок Диаграмма захвата события с входа первого канала.

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 TIMx_IE можно разрешить выработку прерываний по событию захвата на определенном канале, а в регистре TIMx_DMA_RE можно разрешить формирование запросов DMA.

Режим ШИМ

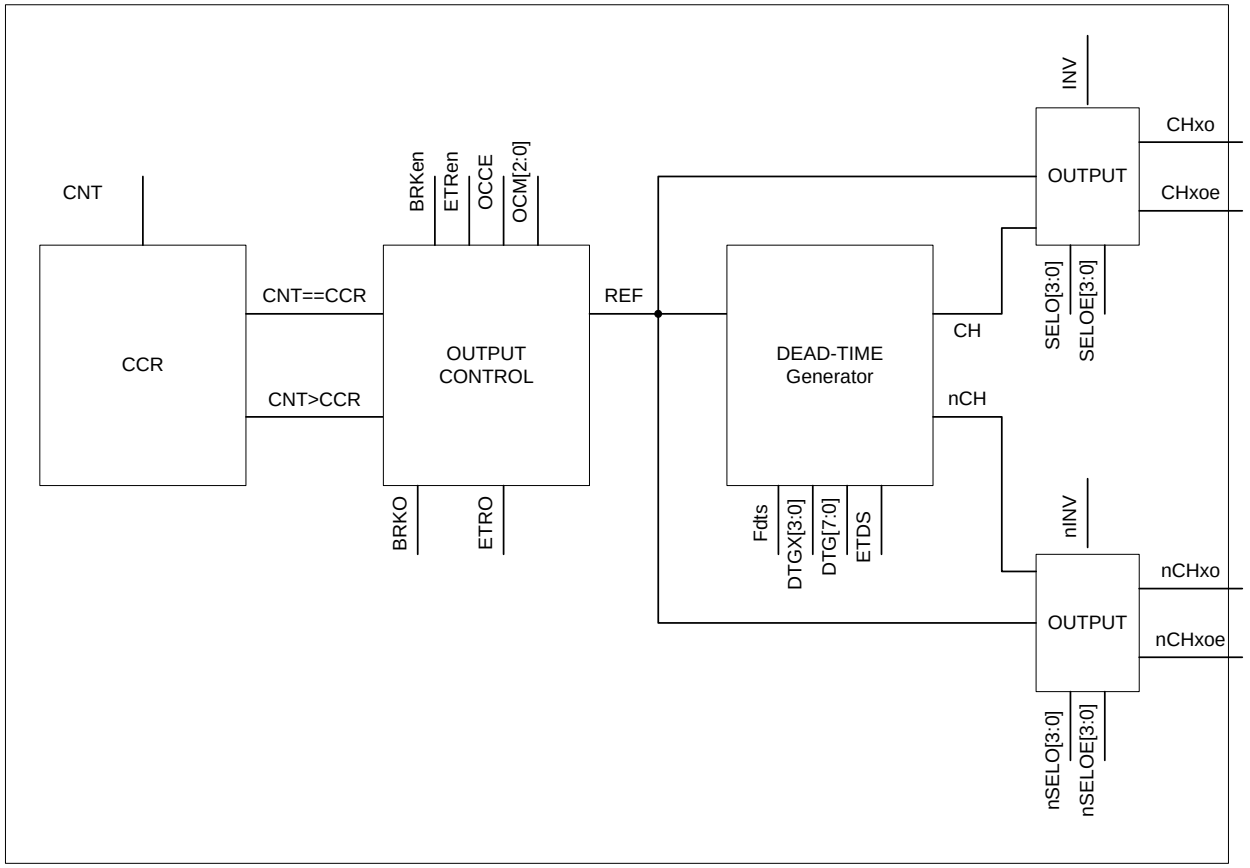


Рисунок Структурная схема блока сравнения.

Для включения режима сравнения для определенного канала необходимо в регистре управления каналом TIMx_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.

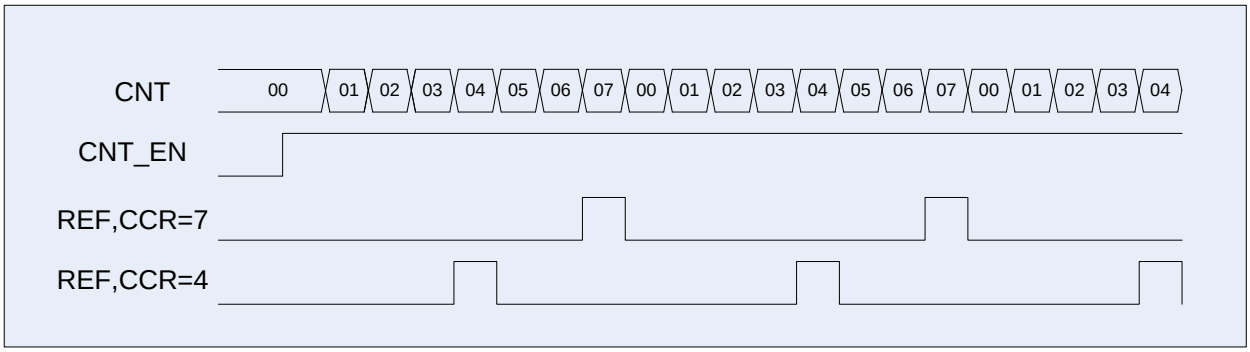


Рисунок Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

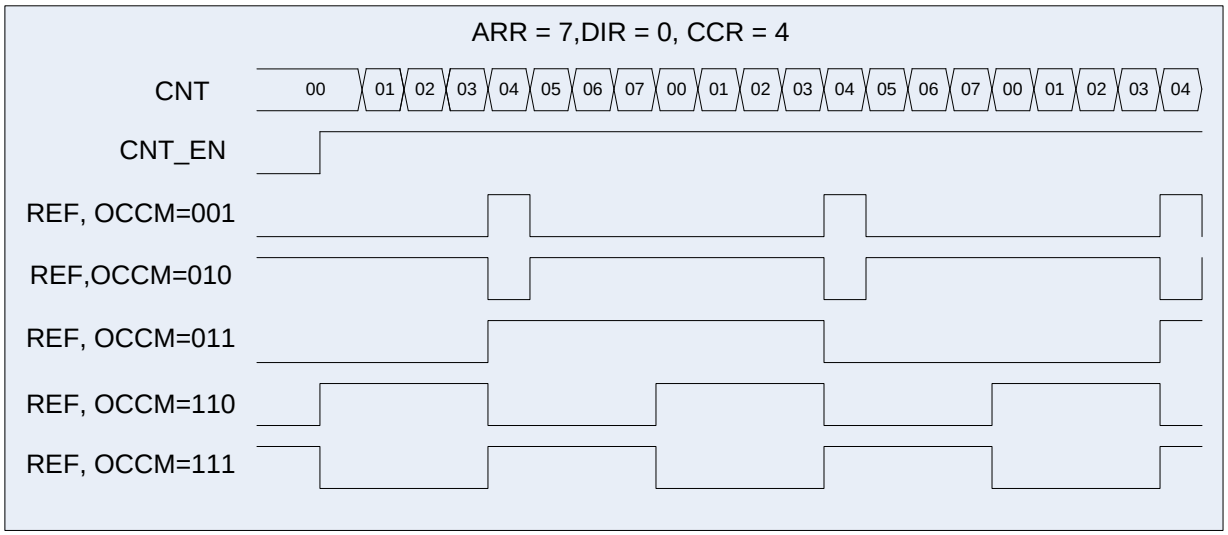


Рисунок Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

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

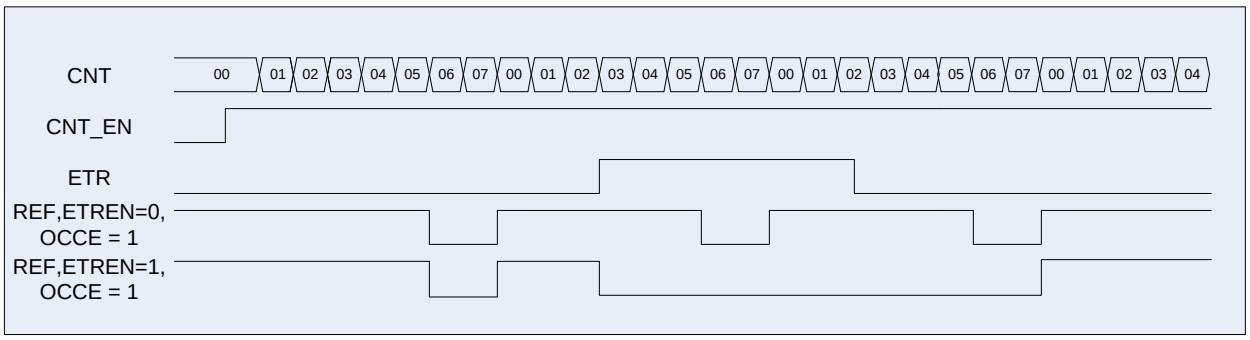


Рисунок Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 0.

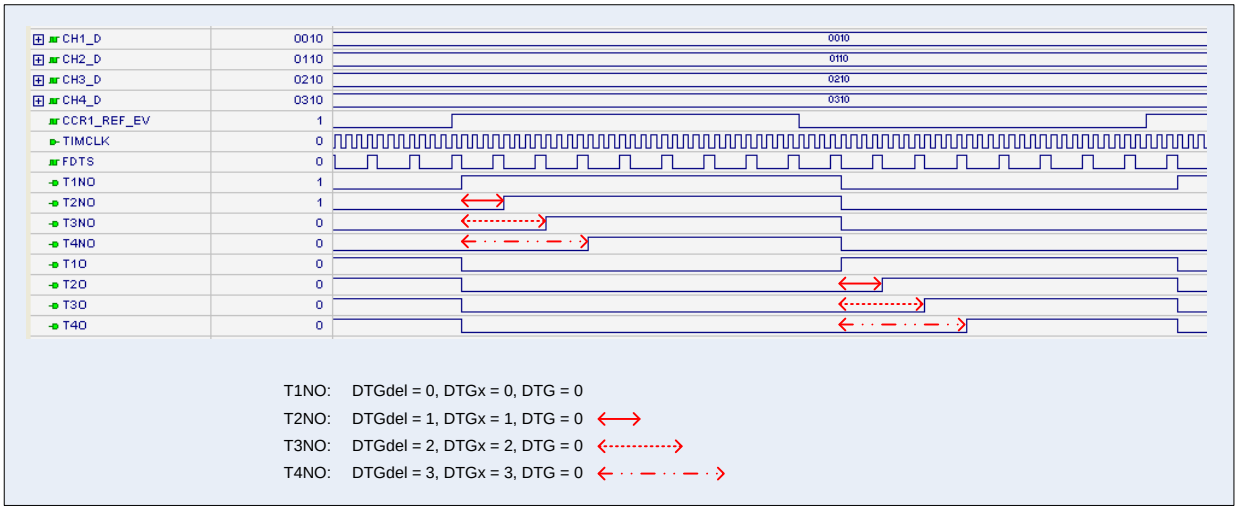


Рисунок Диаграмма работы схемы DTG

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 TIMx_CHy_CNTRL поле OCCM) будет формироваться сигнал соответствующей формы.

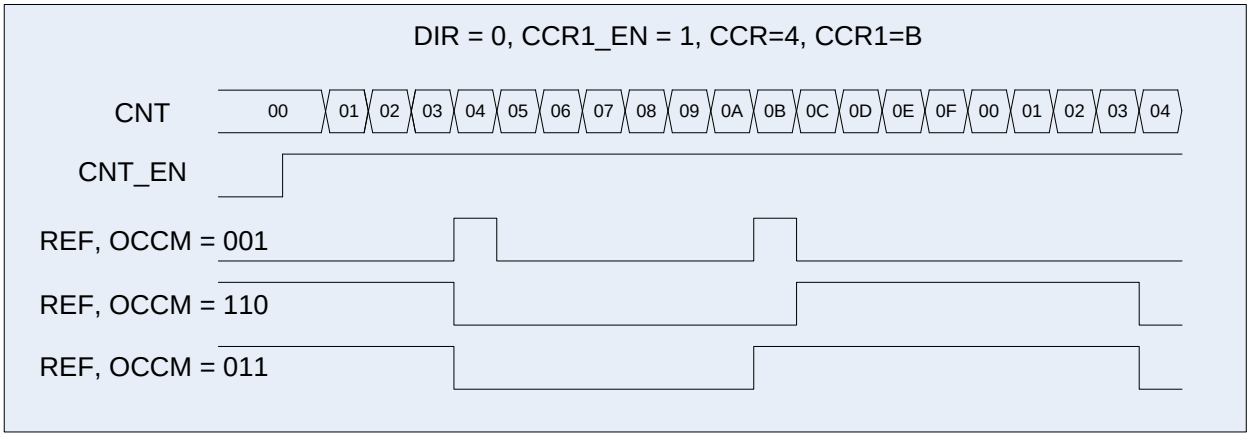


Рисунок Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 1.

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.

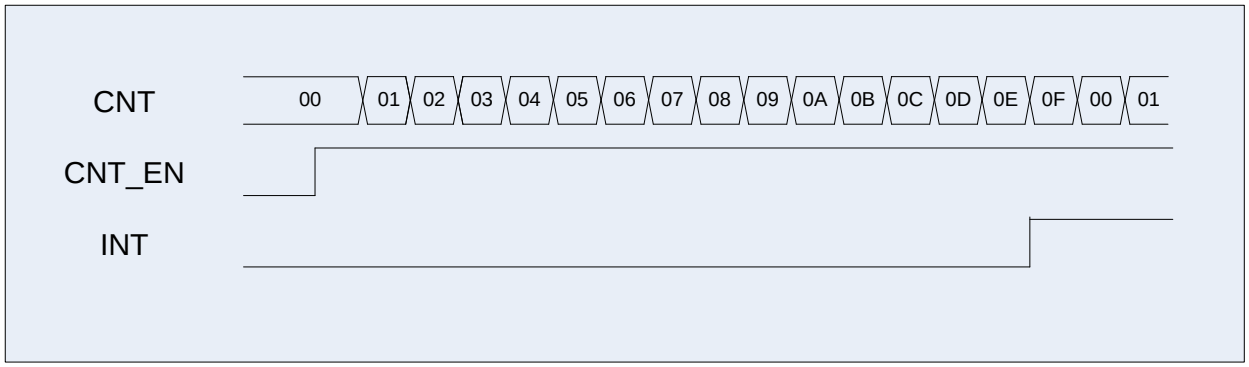
Примеры

Обычный счетчик

```
RST_CLK->PER_CLOCK = 0xFFFFFFFF;
RST_CLK->TIM_CLOCK = 0x07000000;
TIMx->TIMx_CNTRL  = 0x00000000;
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000000;//Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000;//Предделитель частоты
TIMx->TIMx_ARR = 0x0000000F;//Основание счета

TIMx->TIMx_IE = 0x00000002;//Разрешение генерировать прерывание при CNT = ARR

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



Режим захвата

```
RST_CLK->PER_CLOCK = 0xFFFFFFFF;//Разрешение тактовой частоты таймеров
RST_CLK->TIM_CLOCK = 0x07000000;//Включение тактовой частоты таймеров
TIMx->TIMx_CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000000;//Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000;//Предделитель частоты
TIMx->TIMx_ARR = 0x000000FF;//Основание счета

TIMx->TIMx_IE = 0x00001E00;//Разрешение генерировать прерывание
            //по переднему фронту на выходе CAP по всем каналам
//Режим работы каналов - захват
TIMx->TIMx_CHy_CNTRL[0] = 0x00008000;
TIMx->TIMx_CHy_CNTRL[1] = 0x00008002;
TIMx->TIMx_CHy_CNTRL[2] = 0x00008001;
TIMx->TIMx_CHy_CNTRL[3] = 0x00008003;

//Режим работы выхода канала – канал на выход не работает
TIMx->TIMx_CHy_CNTRL1[0]= 0x00000000;
TIMx->TIMx_CHy_CNTRL1[1]= 0x00000000;
TIMx->TIMx_CHy_CNTRL1[2]= 0x00000000;
TIMx->TIMx_CHy_CNTRL1[3]= 0x00000000;

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

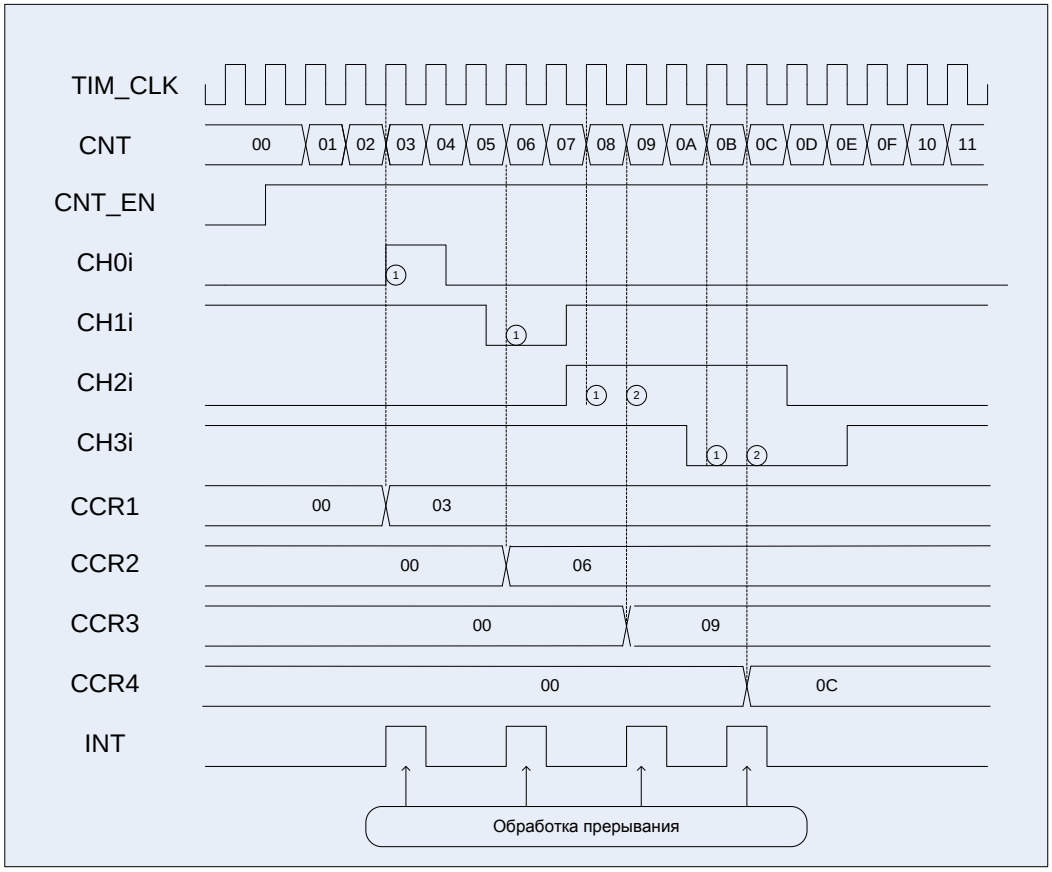


Рисунок Диаграммы примера работы в режиме захвата

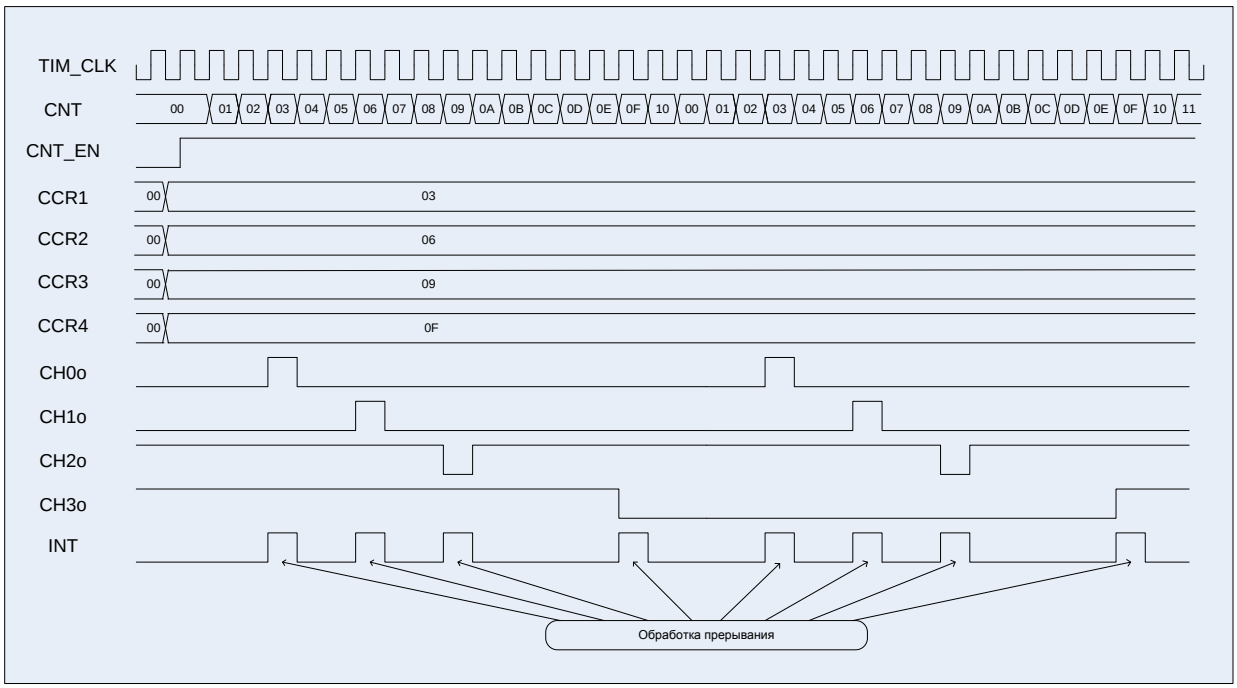
Режим ШИМ

```
RST_CLK->PER_CLOCK = 0xFFFFFFFF;//Разрешение тактовой частоты таймеров
RST_CLK->TIM_CLOCK = 0x07000000;//Включение тактовой частоты таймеров
TIMx->TIMx_CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000000;//Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000;//Предделитель частоты
TIMx->TIMx_ARR = 0x00000010;//Основание счета

TIMx->TIMx_IE = 0x000001E0;//Разрешение генерировать прерывание
            //по переднему фронту на выходе REF по всем каналам
//Режим работы каналов - ШИМ
TIMx->TIMx_CHy_CNTRL[0] = 0x00000200;
TIMx->TIMx_CHy_CNTRL[1] = 0x00000200;
TIMx->TIMx_CHy_CNTRL[2] = 0x00000400;
TIMx->TIMx_CHy_CNTRL[3] = 0x00000600;

//Режим работы выхода канала – канал на выход не работает
TIMx->TIMx_CHy_CNTRL1[0]= 0x00000099;
TIMx->TIMx_CHy_CNTRL1[1]= 0x00000099;
TIMx->TIMx_CHy_CNTRL1[2]= 0x00000099;
TIMx->TIMx_CHy_CNTRL1[3]= 0x00000099;

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


TIM_CLK
CNT
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11
CNT_EN
CCR1
00
03
CCR2
00
06
CCR3
00
09
CCR4
00
0F
CH0o
CH1o
CH2o
CH3o
INT
Обработка прерывания

Описание регистров блока таймера

Базовые адреса и смещения регистров управления.

Адрес	Название	Описание
0x4007_0000	Timer1	Контроллер Timer1
0x4007_8000	Timer2	Контроллер Timer2
0x4008_0000	Timer3	Контроллер Timer3
0x4009_8000	Timer4	Контроллер Timer4
Смещение		
0x00	TIMx_CNT[31:0]	Основной счетчик таймера
0x04	TIMx_PSG[15:0]	Делитель частоты при счете основного счетчика
0x08	TIMx_ARR[31:0]	Основание счета основного счетчика
0x0C	TIMx_CNTRL[11:0]	Регистр управления основного счетчика
0x10	TIMx_CCR1[31:0]	Регистр сравнения, захвата для 1 канала таймера
0x14	TIMx_CCR2[31:0]	Регистр сравнения, захвата для 2 канала таймера
0x18	TIMx_CCR3[31:0]	Регистр сравнения, захвата для 3 канала таймера
0x1C	TIMx_CCR4[31:0]	Регистр сравнения, захвата для 4 канала таймера
0x20	TIMx_CH1_CNTRL[15:0]	Регистр управления для 1 канала таймера
0x24	TIMx_CH2_CNTRL[15:0]	Регистр управления для 2 канала таймера
0x28	TIMx_CH3_CNTRL[15:0]	Регистр управления для 3 канала таймера
0x2C	TIMx_CH4_CNTRL[15:0]	Регистр управления для 4 канала таймера
0x30	TIMx_CH1_CNTRL1[15:0]	Регистр управления 1 для 1 канала таймера
0x34	TIMx_CH2_CNTRL1[15:0]	Регистр управления 1 для 2 канала таймера
0x38	TIMx_CH3_CNTRL1[15:0]	Регистр управления 1 для 3 канала таймера
0x3C	TIMx_CH4_CNTRL1[15:0]	Регистр управления 1 для 4 канала таймера
0x40	TIMx_CH1_DTG[15:0]	Регистр управления DTG для 1 канала таймера
0x44	TIMx_CH2_DTG[15:0]	Регистр управления DTG для 2 канала таймера
0x48	TIMx_CH3_DTG[15:0]	Регистр управления DTG для 3 канала таймера
0x4C	TIMx_CH4_DTG[15:0]	Регистр управления DTG для 4 канала таймера
0x50	TIMx_BRKETR_CNTRL[15:0]	Регистр управления входом BRK и ETR
0x54	TIMx_STATUS[15:0]	Регситр статуса таймера
0x58	TIMx_IE[15:0]	Регистр разрешения прерывания таймера
0x5C	TIMx_DMA_RE[15:0]	Регистр разрешения запросов DMA от прерываний таймера
0x60	TIMx_CH1_CNTRL2[15:0]	Регистр управления 2 для 1 канала таймера
0x64	TIMx_CH2_CNTRL2[15:0]	Регистр управления 2 для 2 канала таймера
0x68	TIMx_CH3_CNTRL2[15:0]	Регистр управления 2 для 3 канала таймера
0x6C	TIMx_CH4_CNTRL2[15:0]	Регистр управления 2 для 4 канала таймера
0x70	TIMx_CCR11[31:0]	Регистр сравнения, захвата 1 для 1 канала таймера

0x74	TIMx_CCR21[31:0]	Регистр сравнения, захвата 1 для 2 канала таймера
0x78	TIMx_CCR31[31:0]	Регистр сравнения, захвата 1 для 3 канала таймера
0x7C	TIMx_CCR41[31:0]	Регистр сравнения, захвата 1 для 4 канала таймера
0x80	TIMx_DMA_RE1[15:0]	Регистр разрешения запросов DMA от прерываний канала 1 таймера
0x84	TIMx_DMA_RE2[15:0]	Регистр разрешения запросов DMA от прерываний канала 2 таймера
0x88	TIMx_DMA_RE3[15:0]	Регистр разрешения запросов DMA от прерываний канала 3 таймера
0x8C	TIMx_DMA_RE4[15:0]	Регистр разрешения запросов DMA от прерываний канала 4 таймера

TIMx_CNT

Основной счетчик таймера

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

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

TIMx_PSG

Делитель частоты при счете основного счетчика

Номер	31	15	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
			PSG[15:0]

№	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)

TIMx_ARR

Основание счета основного счетчика

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

№	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]

TIMx_CNTRL

Регистр управления основного счетчика

Номер	31..10	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

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..11	-	Зарезервировано
11..8	EVENT_SEL [3:0]	Биты выбора источника событий 0000 – всегда “0” 0001 – CNT == ARR в таймере 1 0010 – CNT == ARR в таймере 2 0011 – CNT == ARR в таймере 3 0100 – событие на первом канале 0101 – событие на втором канале 0110 – событие на третьем канале 0111 – событие на четвертом канале 1000 – событие переднего фронта ETR 1001 - событие заднего фронта ETR 1010 - CNT == ARR в таймере 4
7..6	CNT_MODE [1:0]	Режим счета основного счетчика 00 – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при PSG = 0 01 – счетчик вверх/вниз с автоматическим изменением DIR при PSG = 0 10 – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при EVENT = 1 11 – счетчик вверх/вниз с автоматическим изменением DIR при EVENT = 1
5..4	FDTS[1:0]	Частота семплирования данных FDTS 00 – каждый TIM_CLK 01 – каждый второй TIM_CLK 10 – каждый третий TIM_CLK 11 – 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

TIMx_CCRy

Регистр сравнения, захвата для ‘y’ канала таймера

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

№	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

TIMx_CCRy1

Регистр сравнения, захвата 1 для ‘y’ канала таймера

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

№	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

TIMx_CHy_CNTRL

Регистр управления для ‘y’ канала таймера

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]	CHFLTR [3:0]

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

№	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 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

TIMx_CHy_CNTRL1

Регистр управления 1 для ‘y’ канала таймера

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]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..13	-	Зарезервировано
12	NINV	Режим выходной инверсии 0 – выход не инвертируется 1 – выход инвертируется
11..10	NSELO[1:0]	Режим работы выхода 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 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	-	
4	INV	Режим выходной инверсии 0 – выход не инвертируется 1 – выход инвертируется
3…2	SELO[1:0]	Режим работы выхода 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 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

TIMx_CHy_CNTRL2

Регистр управления 2 для ‘y’ канала таймера

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

№	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

TIMx_CHy_DTG

Регистр управления DTG

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

№	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

TIMx_BRKETR_CNTRL

Регистр управления входом BRK и ETR

Номер	31			8	7…4	3…2	1	0
Доступ	U			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

№	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
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

TIMx_STATUS

Регистр статуса таймера

Номер	16…13	12…9	8…5	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	CCR CAP1 EVENT [3:0]	CCR REF EVENT [3:0]	CCR CAP EVENT [3:0]	BRK EVENT	ETR FE EVENT	ETR RE EVENT	CNT ARR EVENT	CNT ZERO EVENT

№	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
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.

TIMx_IE

Регистр разрешения прерывания таймера

Номер	16…13	12…9	8…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	0
	CCR CAP1 EVENT IE [3:0]	CCR REF EVENT IE [3:0]	CCR CAP EVENT IE [3:0]	BRK EVENT IE	ETR FE EVENT IE	ETR RE EVENT IE	CNT ARR EVENT IE	CNT ZERO EVENT 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	Флаг разрешения прерывания по событию совпадения CNT и ARR

	EVENT IE	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

TIMx_DMA_RE

Регистр разрешения запросов DMA от прерываний таймера

Номер	16…13	12…9	8…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	0
	CCR CAP1 EVENT RE [3:0]	CCR REF EVENT RE [3:0]	CCR CAP EVENT RE [3:0]	BRK EVENT RE	ETR FE EVENT RE	ETR RE EVENT RE	CNT ARR EVENT RE	CNT ZERO EVENT 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
1	CNT ARR	Флаг разрешения запроса DMA по событию совпадения CNT и ARR

	EVENT RE	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

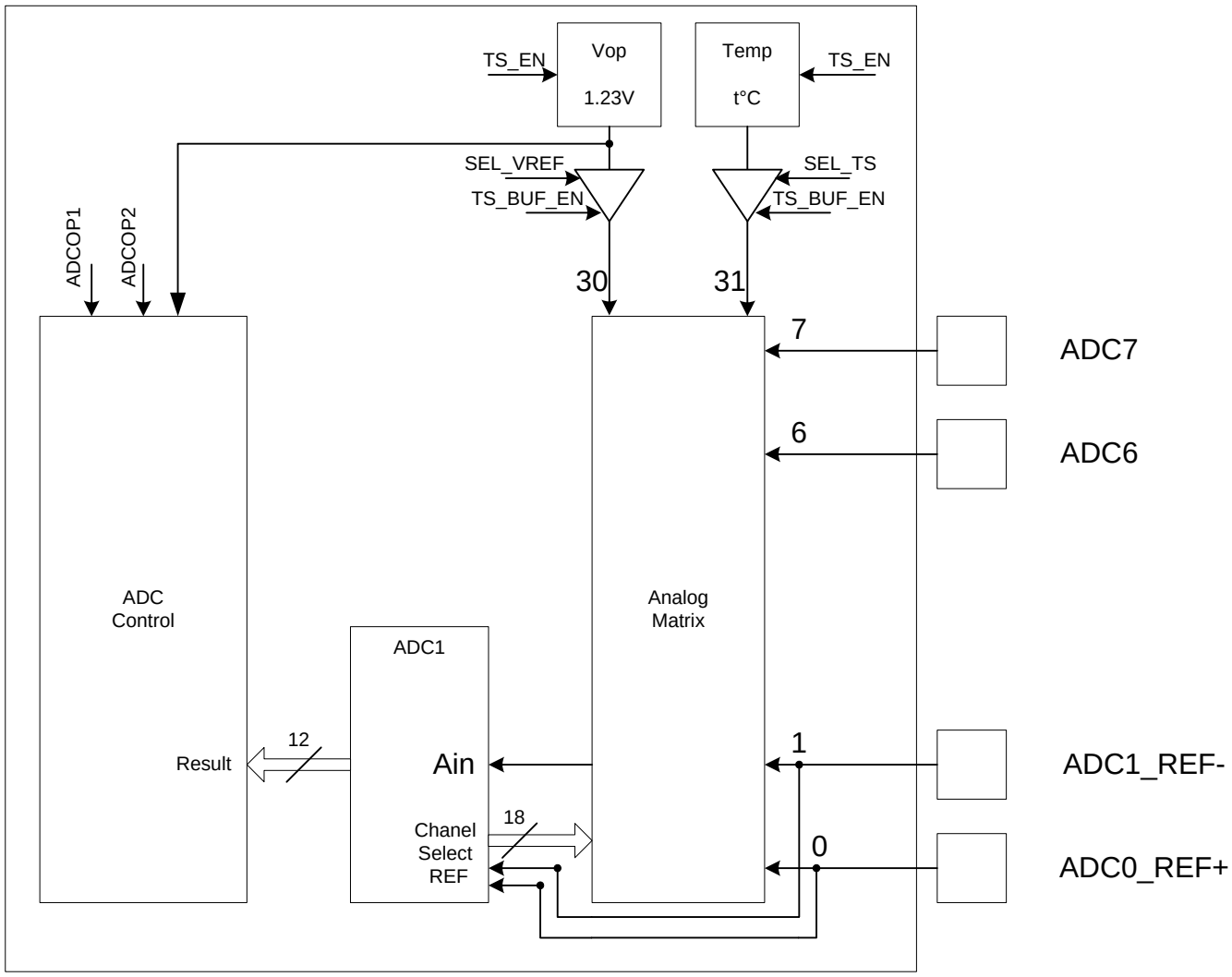
Контроллер АЦП (ревизия 1).

В микроконтроллере реализован 12-ти разрадный АЦП. С помощью АЦП можно оцифровать сигнал с 8 внешних аналоговых выводов порта D и двух внутренних каналов, на которые выводится датчик температуры и источник опорного напряжения. Скорость выборки сотстовляет до 500 тысяч преобразований в секунду.

Контроллер АЦП повзоляет:

- оцифровать один из 8 внешних каналов;
- оцифровать значение встроенного датчика температуры;
- оцифровать значение встроенного источника опорного напряжения;
- осуществить автоматический опрос заданных каналов;
- выработать прерывание при выходе оцифрованного значения за заданные пределы.

Для осуществления преобразования требуется 28 тактов синхросигнала CLK. В качестве синхросигнала может выступать частота процессора CPU_CLK либо частота ADC_CLK формируемая в блоке «Сигналы тактовой частоты». Выбор частоты осуществляется с помощью бита Cfg_REG_CLKS. В контроллере АЦП частота может быть поделена с помощью битов Cfg_REG_DIVCLK[3:0]. Максимальная частота CLK не может превышать 14 МГЦ.



Для включения АЦП необходимо установить бит Cfg_REG_ADON. Для снижения тока потребления вместо собственного источника опорного напряжения в АЦП может использоваться источник датчика температуры. Для этого необходимо включить блок датчика тепературы, и источник опорного напряжения, установив бит TS_EN в 1. После включения можно использовать источник опорного напряжения датчика температуры для АЦП вместо его собственного. Для этого необходимо установить биты ADC1_OP в единицу. Для преобразования необходимо, чтобы выводы, используемые АЦП в порте D, были сконфигурированы как аналоговые и были отключены какие либо внутренние подтяжки.

Преобразование внешнего канала

В регистре ADC1_CFG в битах Cfg_REG_CHS[4:0] необходимо задать соответствующий выводу номер канала. Преобразование может осуществляться при опорном напряжении внутреннем (бит Cfg_M_REF = 0) и внешнем (бит Cfg_M_REF = 1), в этом случае опорное напряжение берется с выводов ADC0_REF+ и ADC1_REF-. Биты Cfg_REG_CHCH, Cfg_REG_RNGC, Cfg_REG_SAMPLE, TS_BUF_EN, SEL_VREF, SEL_TS и Cfg_Sync_Conver должны быть сброшены.

Для начала перобразования необходимо записать 1 в бит Cfg_REG_GO.
После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования.
После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан, и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Фдаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.

Последовательное преобразование нескольких каналов

Для автоматического последовательного преобразования нескольких каналов или одного канала в регистре ADC1_CHSEL необходимо установить единицы в битах соответствующих необходимым для преобразования каналам. Преобразование может осуществляться при опорном напряжении внутреннем (бит Cfg_M_REF = 0) или внешнем (Cfg_M_REF = 1), в этом случае опора берется с выводов ADC0_REF+ и ADC1_REF-. Биты, Cfg_REG_RNGC, TS_BUF_EN, SEL_VREF, SEL_TS и Cfg_Sync_Conver должены быть сброшены, а Cfg_REG_SAMPLE и Cfg_REG_CHCH должны быть установлены. С помощью битов Delay_GO можно задать паузу между преобразованиями при переборе каналов. Для начала перобразования необходимо записать 1 в бит Cfg_REG_GO.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования. После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан, и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Фдаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.

Для последовательного преобразования одного и того же канала можно в регистре ADC1_CHSEL выбрать только один канал и установить бит Cfg_REG_CHCH в 1, либо установить номер канала в битах Cfg_REG_CHS[4:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для

данного канала. Последовательное преобразование значения датчика температуры и источника опорного напряжения могут выполняться только в режиме последовательного преобразования одного канала.

Преобразование с контролем границ

При необходимости отслеживать нахождение оцифрованных значений в допустимых пределах можно задать нижнюю и верхнюю допустимые границы в регистрах ADC1_L_LEVEL и ADC1_H_LEVEL. При этом если установлен бит Cfg_REG_RNGC, то в случае если результат преобразования выходит за границы выставляется флаг Flg_REG_ AWOIFEN. А в регистре результата будет полученное значение.

Датчик опорного напряжения

С помощью АЦП можно осуществить преобразования источника опорного напряжения. Для этого необходимо включить блок датчика тепературы и источник опорного напряжения, установив бит TS_EN в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1_OP в единицу. Для выбора источника опорного напряжения в качестве источника для преобразования необходимо в битах Cfg_REG_CHS установить значение 30 канала. Установить биты TS_BUF_EN и SEL_VREF. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать 1 в бит Cfg_REG_GO.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования.
После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан, и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Флаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.
Для последовательного преобразования только источника опорного напряжения можно в регистре ADC1_CHSEL выбрать только 30 канал и установить бит Cfg_REG_CHCH в 1, либо установить номер 30-го канала в битах Cfg_REG_CHS[4:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должны быть так же установлены биты TS_BUF_EN и SEL_VREF.

Датчик температуры

С помощью первого АЦП можно осуществить преобразования датчика опорного напряжения. Для этого необходимо включить блок датчика тепературы и источник опорного напряжения установив бит TS_EN в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1_OP в единицу. Для выбора датчика тепературы в качестве источника для преобразования необходимо в битах Cfg_REG_CHS установить значение 31 канала. Установить биты TS_BUF_EN и SEL_TS. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать 1 в бит Cfg_REG_GO.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования.
После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан, и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Флаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.

Для последовательного преобразования только датчика тепературы можно в регистре ADC1_CHSEL выбрать только 31 канал и установить бит Cfg_REG_CHCH в 1, либо установить номер 31-го канала в битах Cfg_REG_CHS[4:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должны быть так же установлены биты TS_BUF_EN и SEL_TS.

Время заряда внутренней емкости

Процесс преобразования состоит из двух этапов: сначала происходит заряд внутренней емкости до уровня внешнего сигнала, и затем происходит преобразование уровня заряда внутренней емкости в цифровой вид. Таким образом, для точного преобразования внешнего сигнала в цифровой вид, за время первого этапа внутренняя емкость должна зарядиться до уровня внешнего сигнала. Это время определяется соотноршением номинальной внутренней емкости, входным сопротивлением тракта АЦП и выходным сопротивлением источника сигнала. Приведенная ниже формула позволяет определить максимальное выходное сопротивление источника R_{AIN} для обеспечения качественного преобразования:

$$R_{AIN} < (T_S/(f_{C\ ADC}*C_{ADC}*\ln(2^N)))-R_{ADC}$$

где: T_S - время заряда внутренней емкости в тактах;
$f_{C\ ADC}$ - рабочая частота АЦП;
C_{ADC} - внутренняя емкость АЦП (~15-20пФ);
N - требуемая точность в разрядах;
R_{ADC} - входное сопротивление тракта АЦП (~500 Ом).

Если необходимо обеспечить преобразование с точностью 12 разрядов ± 1/4 LSB, то N = 14. Если необходимо обеспечить преобразование с точностью 10 разрядов ± 1 LSB, то N=10. Время заряда T_S = определяется битами DelayGo[2:0] и схемой самого АЦП и представлена в таблице. Время зарядки внутренней емкости задается битами DelayGo[2:0] определяется в тактах CPU_CLK, независимо от того на какой частоте ADC_CLK или CPU_CLK идет само преобразование.

Время заряда внутренней емкости АЦП и время преобразования

DelayGo[2:0]	Дополнительная задержка перед началом преобразования	Общее время T_S заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	1 x CPU_CLK	4 x CLK + 1 x CPU_CLK	28 x CLK + 1 x CPU_CLK
001	2 x CPU_CLK	4 x CLK + 2 x CPU_CLK	28 x CLK + 2 x CPU_CLK
010	3 x CPU_CLK	4 x CLK + 3 x CPU_CLK	28 x CLK + 3 x CPU_CLK
011	4 x CPU_CLK	4 x CLK + 4 x CPU_CLK	28 x CLK + 4 x CPU_CLK
100	5 x CPU_CLK	4 x CLK + 5 x CPU_CLK	28 x CLK + 5 x CPU_CLK
101	6 x CPU_CLK	4 x CLK + 6 x CPU_CLK	28 x CLK + 6 x CPU_CLK
110	7 x CPU_CLK	4 x CLK + 7 x CPU_CLK	28 x CLK + 7 x CPU_CLK
111	8 x CPU_CLK	4 x CLK + 8 x CPU_CLK	28 x CLK + 8 x CPU_CLK

Помимо точности определяемой временем зарядки внутренней емкости АЦП точность преобразования имеет ошибки связанные с технологическими разбросами схемы и шумами и определяемые параметрами E_{DLADC}, E_{ILADC} и E_{OFFADC}.

Для корректного задание режимов работы АЦП, настройки в регистре ADC1_CFG необходимо сделать до задания бита Go, иначе новая конфигурация будет действовать со следующего преобразования.

Описание регистров блока контроллера АЦП

Базовый Адрес	Название	Описание
0x4008_8000	ADC	Контроллер ADC
Смещение		
0x00	ADC1_CFG	Регистр управления ADC
0x04	ADC2_CFG	Регистр управления ADC
0x08	ADC1_H_LEVEL	Регистр верхней границы ADC
0x10	ADC1_L_LEVEL	Регистр нижней границы ADC
0x18	ADC1_RESULT	Регистр результата ADC
0x20	ADC1_STATUS	Регистр статуса ADC
0x28	ADC1_CHSEL	Регистр выбора каналов перебора ADC

ADCx_CFG

Номер	11	10	9	8…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	0
	Cfg REG M_REF	Cfg REG RNGC	Cfg REG CHCH	Cfg REG CHS [4:0]	Cfg REG SAMPLE	Cfg REG CLKS	Cfg REG GO	Cfg REG ADON

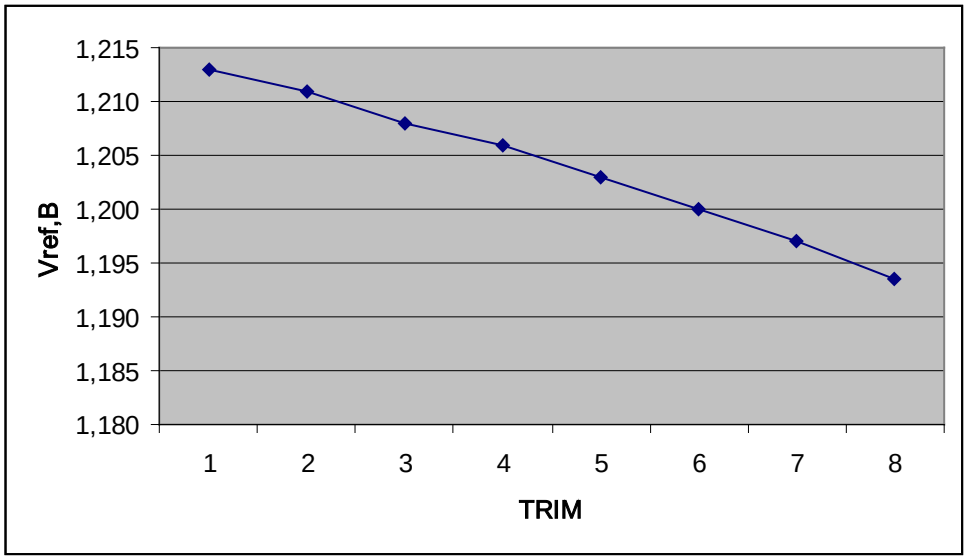
Номер	27…25	24…21	20	19	18	17	16	15…12
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	Delay Go [2:0]	TR[3:0]	SEL VREF	SEL TS	TS_BUF EN	TS_EN / ADC1 OP	Cfg Sync Conver	Cfg REG DIVCLK [3:0]

Номер	31…28
Доступ	R/W
Сброс	0
	Delay ADC [3:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28	Delay ADC [3:0]	Задержка между началом преобразования ADC1 при последовательном переборе, либо работе на один канал. 0000 – 0 тактов CLK 0001 – 1 такт CLK … 1111 – 15 тактов CLK
27..25	Delay Go [2:0]	Задержка перед началом следующего преобразования после завершения предыдущего при последовательном переборе каналов 000 – 0 тактов CLK 001 – 1 такт CLK … 111 – 7 тактов CLK
24..21	TR[3:0]	Подстройка опорного напряжения Смотри диаграмму на Зависимость источника опорного напряжения от подстройки
20	SEL VREF	Выбор для оцифровки источника опорного напряжения на 1.23В 0 – не выбран 1 – выбран Должен использоваться совместно с выбором канала Cfg_REG_CHS = 30.
19	SEL TS	Выбор для оцифровки датчика температуры 0 – не выбран 1 – выбран

		Должен использоваться совместно с выбором канала Cfg_REG_CHS = 31.
18	TS BUF EN	**В регистре ADC1_CFG.** Включение выходного усилителя для датчика температуры и источника опорного напряжения 0 – выключен 1 – включен Используется при TS_EN = 1. Для уменьшения тока потребления.
17	TS EN	**В регистре ADC1_CFG.** Включение датчика температуры и источника опорного напряжения 0 – выключен 1 – включен При включении датчика температуры и источника опорного напряжения выходной сигнал стабилизируется в течение времени Tstb.
17	ADC1 OP	**В регистре ADC2_CFG.** Выбор источника опорного напряжения 1.23В 0 – внутренний (не точный) 1 – от датчика температуры (точный)
16	Cfg Sync Conver	Записывать всегда ноль
15..12	Cfg REG DIVCLK [3:0]	Выбор коэффициента деления входной частоты 0000 – CLK 0001 – CLK/2 0010 – CLK/4 0011 – CLK/8 … 1111 – CLK/32768
11	Cfg M_REF	Выбор источника опорных напряжений 0 – внутренне опорное напряжение (от AUdd и AUss) 1 – внешнее опорное напряжение (от Uref+ и Uref-)
10	Cfg REG RNGC	Разрешение автоматического контролирования уровней 1 – Разрешено, выработка прерывания при выходе за диапазон в регистрах границы обработки 0 – не разрешено
9	Cfg REG CHCH	Выбор переключения каналов 1 – переключение включено (перебираются каналы, выбранные в регистре выбора канала) 0 – используется только выбранный канал
8…4	Cfg REG CHS [4:0]	Выбор аналогового канала, по которому поступает сигнал для преобразования 00000 – 0 канал 00001 – 1 канал … 11111 – 31 канал
3	Cfg REG SAMPLE	Выбор способа запуска АЦП. 1 – последовательное преобразование, автоматический запуск после завершения предыдущего преобразования 0 – одиночное преобразование
2	Cfg	Выбор источника синхросигнала CLK работы ADC

	REG CLKS	1 – ACLK 0 – PCLK
1	Cfg REG GO	Начало преобразования Запись “1” начинает процесс преобразования, сбрасывается автоматически
0	Cfg REG ADON	Включение АЦП 1 – включено 0 – выключено



Зависимость источника опорного напряжения от подстройки

ADC1_H_LEVEL

Номер	31	12	11…0
Доступ	U	U	R/W
Сброс	0	0	0
	-	-	REG H LEVEL [11:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11…0	REG H LEVEL [11:0]	Верхняя граница зоны допуска.

ADC1_L_LEVEL

Номер	31	12	11…0
Доступ	U	U	R/W
Сброс	0	0	0
	-	-	REG L LEVEL [11:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11…0	REG L LEVEL [11:0]	Нижняя граница зоны допуска.

ADC1_RESULT

Номер	31	21	20…16	15…12	11…0
Доступ	U	U	RO	U	RO
Сброс	0	0	0	0	0
	-	-	CHANNEL [11:0]	-	RESULT [11:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
20..16	CHANNEL [11:0]	Канал результата преобразования
15..12	-	Зарезервировано
11…0	RESULT [11:0]	Значение результата преобразования

ADC1_STATUS

Номер	3		5	4	3	2	1	0
Доступ	U		U	R/W	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0	0
	-		-	ECOIF IE	AWOIF IE	Flg REG EOCIF	Flg REG AWOIF EN	Flg REG OVER WRITE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…5	-	Зарезервировано
4	ECOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_ECOIF 0 – прерывания не генерируется 1 – прерывание генерируется
3	AWOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_AWOIFEN 0 – прерывания не генерируется 1 – прерывание генерируется
2	Flg REG EOCIF	Флаг выставляется, когда закончено преобразования и данные еще не считаны. Очищается считыванием результата из регистра ADCx_RESULT. 1 – есть готовый результат преобразования 0 – нет результата
1	Flg REG AWOIFEN	Флаг выставляется, когда результат преобразования выше верхней или ниже нижней границы автоматического контролирования уровней. Очищается считыванием результата из регистра ADCx_RESULT 0 – результат в допустимой зоне 1 – вне допустимой зоны
0	Flg REG OVERWRITE	Данные в регистре результата были перезаписаны, данный флаг сбрасывается только при записи в регистр флагов. 0 – не было события перезаписи не считанного результата 1 – был результат преобразования, который не был считан

ADC1_CHSEL

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
	Sl_Ch_Ch_REF[31:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	Sl_Ch_Ch_REF [31:0]	Выбор каналов автоматического перебора 0 – канал не участвует в переборе 1 – канал участвует в переборе

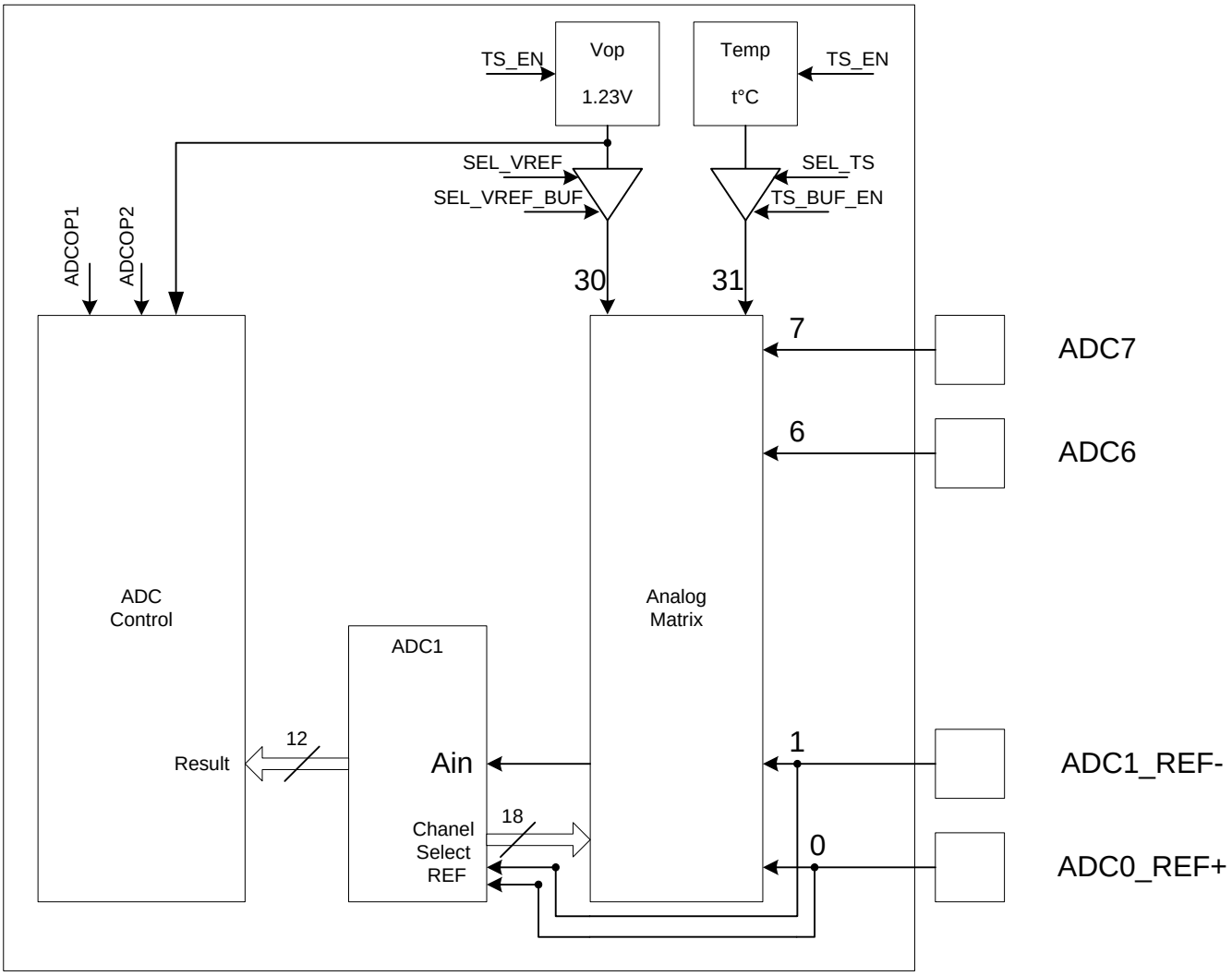
Контроллер АЦП (ревизия 2).

В микроконтроллере реализован 12-ти разрадный АЦП. С помощью АЦП можно оцифровать сигнал с 8 внешних аналоговых выводов порта D и двух внутренних каналов, на которые выводится датчик температуры и источник опорного напряжения. Скорость выборки сотстовляет до 500 тысяч преобразований в секунду.

Контроллер АЦП повзоляет:

- оцифровать один из 8 внешних каналов
- оцифровать значение встроенного датчика температуры
- оцифровать значение встроенного источника опорного напряжения
- осуществить автоматический опрос заданных каналов
- выработать прерывание при выходе оцифрованного значения за заданные пределы

Для осуществления преобразования требуется 28 такта синхронизации CLK. В качестве синхросигнала может выступать частота процессора CPU_CLK либо частота ADC_CLK формируемая в блоке «Сигналы тактовой частоты». Выбор частоты осуществляется с помощью бита Cfg_REG_CLKS. В контроллере АЦП частота может быть поделена с помощью битов Cfg_REG_DIVCLK[3:0]. Максимальная частота CLK не может превышать 14 МГЦ.



Для включения АЦП необходимо установить бит Cfg_REG_ADON. Для снижения тока потребления вместо собственного источника опорного напряжения в АЦП может использоваться источник датчика температуры. Для этого необходимо включить блок датчика тепературы и источник опорного напряжения установив бит TS_EN в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного. Для этого необходимо установить биты ADC1_OP в единицу. Для преобразования необходимо, чтобы выводы, используемые АЦП в порте D, были сконфигурированы как аналоговые и были отключены какие либо внутренние подтяжки.

Преобразование внешнего канала

В регистре ADC1_CFG в битах Cfg_REG_CHS[4:0] необходимо задать соответствующий выводу номер канала. Преобразование может осуществляться при внутренней опоре бит Cfg_M_REF = 0 и внешней Cfg_M_REF = 1, в этом случая опора берется с выводов ADC0_REF+ и ADC1_REF-. Биты Cfg_REG_CHCH, Cfg_REG_RNGC, Cfg_REG_SAMPLE, TS_BUF_EN, SEL_VREF, SEL_TS и Cfg_Sync_Conver должны быть сброшены.

Для начала перобразования необходимо записать 1 в бит Cfg_REG_GO.
После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования.
После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Фдаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.

Последовательное преобразование нескольких каналов

Для автоматического последовательного преобразования нескольких каналов или одного канала в регистре ADC1_CHSEL необходимо установить единицы в битах соответствующих необходимым для преобразования каналам. Преобразование может осуществляться при внутренней опоре бит Cfg_M_REF = 0 и внешней Cfg_M_REF = 1, в этом случае опора берется с выводов ADC0_REF+ и ADC1_REF-. Биты, Cfg_REG_RNGC, TS_BUF_EN, SEL_VREF, SEL_TS и Cfg_Sync_Conver должены быть сброшены, а Cfg_REG_SAMPLE и Cfg_REG_CHCH должны быть установлены. С помощью битов Delay_GO можно задать паузу между преобразованиями при переборе каналов. Для начала перобразования необходимо записать 1 в бит Cfg_REG_GO.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования.
После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Фдаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.

Для последовательного преобразования одного и того же канала можно в регистре ADC1_CHSEL выбрать только один канал и установить бит Cfg_REG_CHCH в 1, либо установить номер канала в битах Cfg_REG_CHS[4:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. Последовательное преобразование значения датчика температуры и

источника опорного напряжения могут выполняться только в режиме последовательного преобразования одного канала.

Преобразование с контролем границ

При необходимости отслеживать нахождение оцифрованных значений в допустимых пределах можно задать нижнюю и верхнюю допустимые границы в регистрах ADC1_L_LEVEL и ADC1_H_LEVEL. При этом если установлен бит Cfg_REG_RNGC, то в случае если результат преобразования выходит за границы выставляется флаг Flg_REG_ AWOIFEN. А в регистре результата будет полученное значение.

Датчик опорного напряжения

С помощью АЦП можно осуществить преобразования источника опорного напряжения. Для этого необходимо включить блок датчика тепературы и источник опорного напряжения установив бит TS_EN в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1_OP в единицу. Для выбора источника опорного напряжения в качестве источника для преобразования необходимо в битах Cfg_REG_CHS установить значение 30 канала. Установить биты SEL_VREF_BUF и SEL_VREF. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать 1 в бит Cfg_REG_GO.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования.
После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Флаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.
Для последовательного преобразования только источника опорного напряжения можно в регистре ADC1_CHSEL выбрать только 30 канал и установить бит Cfg_REG_CHCH в 1, либо установить номер 30-го канала в битах Cfg_REG_CHS[4:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должны быть так же установлены биты SEL_VREF_BUF и SEL_VREF.

Датчик температуры

С помощью первого АЦП можно осуществить преобразования датчика опорного напряжения. Для этого необходимо включить блок датчика тепературы и источник опорного напряжения установив бит TS_EN в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1_OP в единицу. Для выбора датчика тепературы в качестве источника для преобразования необходимо в битах Cfg_REG_CHS установить значение 31 канала. Установить биты TS_BUF_EN и SEL_TS. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать 1 в бит Cfg_REG_GO.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования.
После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от

последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Флаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.
Для последовательного преобразования только датчика тепературы можно в регистре ADC1_CHSEL выбрать только 31 канал и установить бит Cfg_REG_CHCH в 1, либо установить номер 31-го канала в битах Cfg_REG_CHS[4:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должны быть так же установлены биты TS_BUF_EN и SEL_TS.

Время заряда внутренней емкости

Процесс преобразования состоит из двух этапов: сначала происходит заряд внутренней емкости до уровня внешнего сигнала, и затем происходит преобразование уровня заряда внутренней емкости в цифровой вид. Таким образом, для точного преобразования внешнего сигнала в цифровой вид, за время первого этапа внутренняя емкость должна зарядится до уровня внешнего сигнала. Это время определяется соотнорщением номинальной внутренней емкости, входным сопротивлением тракта АЦП и выходным сопротивлением источника сигнала. Приведенная ниже формула позволяет определить максимальное выходное сопротивление источника R_{AIN} для обеспечения качественного преобразования:

$$R_{AIN} < (T_S/(f_{C\ ADC}*C_{ADC}*\ln(2^N)))-R_{ADC}$$

где: T_S - время заряда внутренней емкости в тактах
$f_{C\ ADC}$ - рабочая частота АЦП
C_{ADC} - внутренняя емкость АЦП (~15-20пФ)
N - требуемая точность в разрядах
R_{ADC} - входное сопротивление тракта АЦП (~500 Ом)

Если необходимо обеспечить преобразование с точностью 12 разрядов ± 1/4 LSB, то N = 14. Если необходимо обеспечить преобразование с точностью 10 разрядов ± 1 LSB, то N=10. Время заряда T_S = определяется битами DelayGo[2:0] и схемой самого АЦП и представлена в таблице. Время зарядки внутренней емкости задается битами DelayGo[2:0] определяется в тактах CPU_CLK, независимо от того на какой частоте ADC_CLK или CPU_CLK идет само преобразование.

Время заряда внутренней емкости АЦП и время преобразования

DelayGo[2:0]	Дополнительная задержка перед началом преобразования	Общее время T_S заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	1 x CPU_CLK	4 x CLK + 1 x CPU_CLK	28 x CLK + 1 x CPU_CLK
001	2 x CPU_CLK	4 x CLK + 2 x CPU_CLK	28 x CLK + 2 x CPU_CLK
010	3 x CPU_CLK	4 x CLK + 3 x CPU_CLK	28 x CLK + 3 x CPU_CLK
011	4 x CPU_CLK	4 x CLK + 4 x CPU_CLK	28 x CLK + 4 x CPU_CLK
100	5 x CPU_CLK	4 x CLK + 5 x CPU_CLK	28 x CLK + 5 x CPU_CLK
101	6 x CPU_CLK	4 x CLK + 6 x CPU_CLK	28 x CLK + 6 x CPU_CLK
110	7 x CPU_CLK	4 x CLK + 7 x CPU_CLK	28 x CLK + 7 x CPU_CLK
111	8 x CPU_CLK	4 x CLK + 8 x CPU_CLK	28 x CLK + 8 x CPU_CLK

Помимо точности определяемой временем зарядки внутренней емкости АЦП точность преобразования имеет ошибки связанные с технологическими разбросами схемы и шумами и определяемые параметрами E_{DLADC}, E_{ILADC} и E_{OFFADC}.

Для корректного задание режимов работы АЦП в регистре ADC1_CFG необходимо сделать до задания бита Go, иначе новая конфигурация будет действовать со следующего преобразования.

Описание регистров блока контроллера АЦП

Базовый Адрес	Название	Описание
0x4008_8000	ADC	Контроллер ADC
Смещение		
0x00	ADC1_CFG	Регистр управления ADC
0x04	ADC2_CFG	Регистр управления ADC
0x08	ADC1_H_LEVEL	Регистр верхней границы ADC
0x10	ADC1_L_LEVEL	Регистр нижней границы ADC
0x18	ADC1_RESULT	Регистр результата ADC
0x20	ADC1_STATUS	Регистр статуса ADC
0x28	ADC1_CHSEL	Регистр выбора каналов перебора ADC
0x30	ADC1_TRIM	Регистр настроки термодатчика

ADCx_CFG

Номер	11	10	9	8…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	0
	Cfg REG M_REF	Cfg REG RNGC	Cfg REG CHCH	Cfg REG CHS [4:0]	Cfg REG SAMPLE	Cfg REG CLKS	Cfg REG GO	Cfg REG ADON

Номер	27…25	24…21	20	19	18	17	16	15…12
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	Delay Go [2:0]	TR[3:0]	SEL VREF	SEL TS	TS_BUF EN	TS_EN / ADC1 OP	Cfg Sync Conver	Cfg REG DIVCLK [3:0]

Номер	31…28
Доступ	R/W
Сброс	0
	Delay ADC [3:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28	Delay ADC [3:0]	Задержка между началом преобразования ADC1 при последовательном переборе, либо работе на один канал. 0000 - 0 тактов CLK 0001 - 1 такт CLK … 1111 - 15 тактов CLK
27..25	Delay Go [2:0]	Задержка перед началом следующего преобразования после завершения предыдущего при последовательном переборе каналов 000 - 0 тактов CLK 001 - 1 такт CLK … 111 - 7 тактов CLK
24..21	-	Зарезервировано
20	SEL VREF	Выбор для оцифровки источника опорного напряжения на 1.23В 0 – не выбран 1 – выбран Должен использоваться совместно с выбором канала Cfg_REG_CHS = 30.
19	SEL TS	Выбор для оцифровки датчика температуры 0 – не выбран 1 – выбран Должен использоваться совместно с выбором канала Cfg_REG_CHS = 31.

18	TS BUF EN	**В регистре ADC1_CFG.** Включение выходного усилителя для датчика температуры 0 – выключен 1 – включен Используется при TS_EN = 1. Для уменьшения тока потребления.
17	TS EN	**В регистре ADC1_CFG.** Включение датчика температуры и источника опорного напряжения 0 – выключен 1 – включен При включении датчика температуры и источника опорного напряжения выходной сигнал стабилизируется в течении времени Tstb.
17	ADC1 OP	**В регистре ADC2_CFG.** Выбор источника опорного напряжения 1.23В 0 – внутренний (не точный) 1 – от датчика температуры (точный)
16	Cfg Sync Conver	Записывать всегда ноль
15..12	Cfg REG DIVCLK [3:0]	Выбор коэффициента деления входной частоты 0000 – CLK 0001 – CLK/2 0010 – CLK/4 0011 – CLK/8 … 1111 – CLK/32768
11	Cfg M_REF	Выбор источника опорных напряжений 0 – внутренне опорное напряжение (от AUdd и AUss) 1 – внешнее опорное напряжение (от Uref+ и Uref-)
10	Cfg REG RNGC	Разрешение автоматического контролирования уровней 1 – Разрешено, выработка прерывания при выходе за диапазон в регистрах границы обработки 0 – не разрешено
9	Cfg REG CHCH	Выбор переключения каналов 1 – переключение включено (перебираются каналы выбранные в регистре выбора канала) 0 – используется только выбранный канал
8…4	Cfg REG CHS [4:0]	Выбор аналогового канала, по которому поступает сигнал для преобразования. 00000 – 0 канал 00001 – 1 канал … 11111 – 31 канал
3	Cfg REG SAMPLE	Выбор способа запуска АЦП. 1 – последовательное, автоматический запуск после завершения предыдущего преобразования 0 – одиночное.
2	Cfg REG CLKS	Выбор источника синхросигнала CLK работы ADC 1 – ACLK 0 – PCLK

1	Cfg REG GO	Начало преобразования Запись “1” начинает процесс преобразования, сбрасывается автоматически
0	Cfg REG ADON	Включение АЦП 1 – включено 0 – выключено

ADC1_H_LEVEL

Номер	31	12	11…0
Доступ	U	U	R/W
Сброс	0	0	0
	-		REG H LEVEL [11:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11…0	REG H LEVEL [11:0]	Верхняя граница зоны допуска.

ADC1_L_LEVEL

Номер	31	12	11…0
Доступ	U	U	R/W
Сброс	0	0	0
	-		REG L LEVEL [11:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
11…0	REG L LEVEL [11:0]	Нижняя граница зоны допуска.

ADC1_RESULT

Номер	31		21	20…16	15…12	11…0
Доступ	U		U	RO	U	RO
Сброс	0		0	0	0	0
	-	-	-	CHANNEL [11:0]	-	RESULT [11:0]

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12	-	Зарезервировано
20..16	CHANNEL [11:0]	Канал результата преобразования

15..12	-	Зарезервировано
11…0	RESULT [11:0]	Значение результата преобразования

ADC1_STATUS

Номер	3		5	4	3	2	1	0
Доступ	U		U	R/W	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0	0
	-		-	ECOIFIE	AWOIFIE	Flg REG EOCIF	Flg REG AWOIFEN	Flg REG OVERWRITE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…5	-	Зарезервировано
4	ECOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_ECOIF 0 – прерывания не генерируется 1 – прерывание генерируется
3	AWOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_AWOIFEN 0 – прерывания не генерируется 1 – прерывание генерируется
2	Flg REG EOCIF	Флаг выставляется, когда закончено преобразования и данные еще не считаны. Очищается считыванием результата из регистра ADCx_RESULT. 1 – есть готовый результат преобразования 0 – нет результата
1	Flg REG AWOIFEN	Флаг выставляется, когда результат преобразования выше верней или ниже нижней границы автоматического контролирования уровней. Очищается считывание результата из регистра ADCx_RESULT. 0 – результат в допустимой зоне 1 – вне допустимой зоны
0	Flg REG OVERWRITE	Данные в регистре результата были перезаписаны, данный флаг сбрасывается только при записи в регистр флагов. 0 – не было события перезаписи не считанного результата 1 – был результат преобразования, который не был считан

ADC1_CHSEL

Номер	31	0
Доступ	R/W	R/W
Сброс	0	0
	Sl_Ch_Ch_REF[31:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	Sl_Ch_Ch_REF [31:0]	Выбор каналов автоматического перебора 0 в соответствующем бите канал не участвует в переборе 1 –канал участвует в переборе

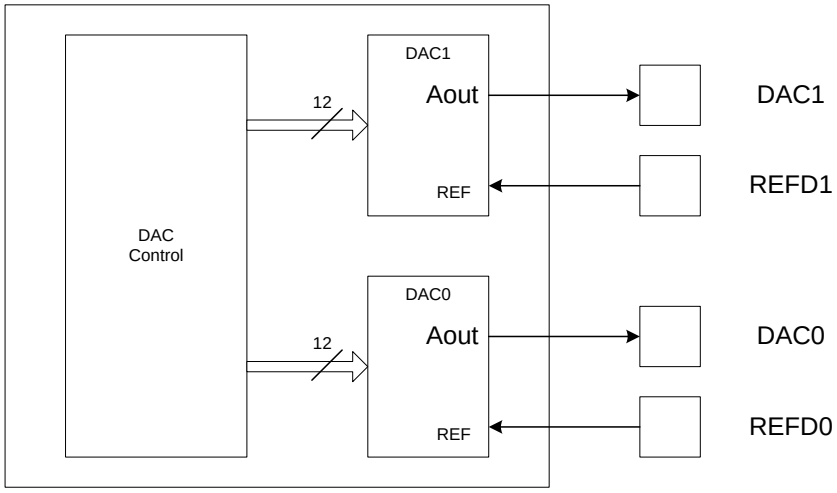
ADC1_TRIM

Номер	31..7	6	5	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	1	0	0	0	0	0
	-	SEL_VREF_BUF	TS_TRIM[4:0]					0

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..7	-	Зарезервировано
6	SEL VREF BUF	Включение выходного усилителя для источника опорного напряжения 0 – выключен 1 – включен Используется при TS_EN = 1. Для уменьшения тока потребления.
5..1	TS_TRIM[4:0]	Подстройка опорного напряжения
0	-	Зарезервировано

Контроллер ЦАП.

В микроконтроллере реализован ЦАП. Для включения ЦАП необходимо установить бит Cfg_ON_DACx в 1, используемые выводы ЦАП в портах D и E сконфигурировать как аналоговые, и отключить какие-либо внутренние подтяжки этих выводов. При работе ЦАП после записи данных в регистр данных DACx_DATA на выходе DACx_OUT формируется уровень напряжения соответсвующий записанному значению. ЦАП может работать от внутреннего (Cfg_M_REFx=0) или внешнего (Cfg_M_REFx=1) опорного напряжения. Если используется внутреннее опорное напряжение, то ЦАП формирует выходной сигнал в диапазоне от 0 до напряжения питания AUcc. В режиме работы с внешним опорным напряжением ЦАП формирует выходное напряжение в диапазоне от 0 до значения DACx_REF.



Описание регистров блока контроллера ЦАП

Базовый Адрес	Название	Описание
0x4009_0000	DAC	Контроллер DAC
Смещение		
0x00	DAC_CFG	Регистр управления DAC
0x04	DAC0_DATA	Регистр данных DAC0
0x08	DAC1_DATA	Регистр данных DAC1

DAC_CFG

Номер	31 … 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
		Cfg_SYNC_A	Cfg_ON_DAC1	Cfg_ON_DAC0	Cfg_M_REF1	Cfg_M_REF0

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…5	-	Зарезервировано
4	Cfg_SYNC_A	Синхронизация DAC0 и DAC1 0 – асинхронные 1 – синхронные
3	Cfg_ON_DAC1	Включение DAC1 1 – включен 0 – выключен
2	Cfg_ON_DAC0	Включение DAC0 1 – включен 0 – выключен
1	Cfg_M_REF1	Выбор источника опорного напряжения DAC1 0 – в качестве опорного напряжения используется напряжение питания с вывода AUcc. 1 – в качестве опорного напряжения используется напряжение на входе DACx_REF.
0	Cfg_M_REF0	Выбор источника опорного напряжения DAC0

DAC0_DATA

Номер	31…28	27	16	15…12	11	0
Доступ	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	0	0
		DAC1_DATA[11:0]			DAC0_DATA[11:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..18	-	Зарезервировано
27..16	DAC1 DATA[11:0]	Данные DAC1 при Cfg_SYNC_A=1. При чтении всегда равны нулю. Читать из DAC1_DATA
15..12	-	Зарезервировано
11…0	DAC0 DATA[11:0]	Данные DAC0

DAC1_DATA

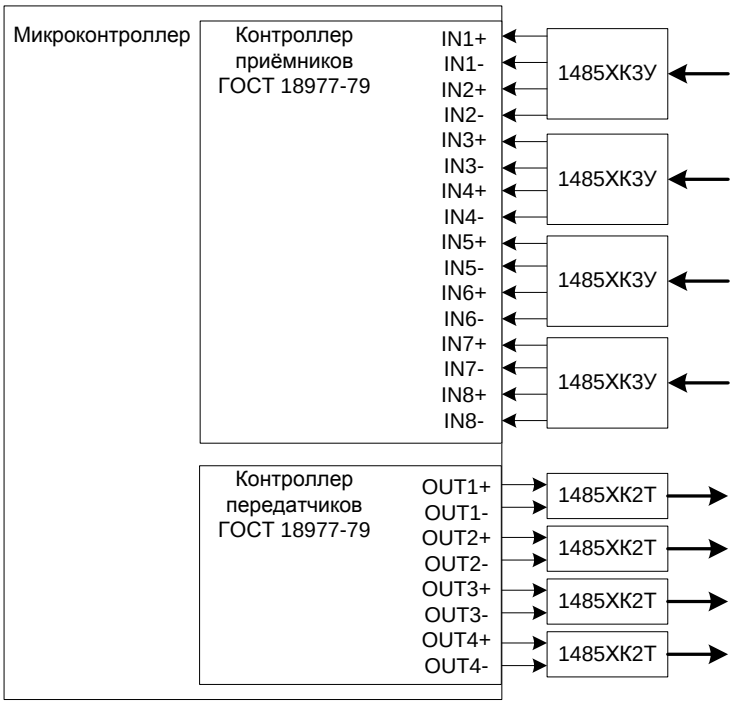
Номер	31…28	27	16	15…12	11	0
Доступ	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	0	0
		DAC0_DATA[11:0]			DAC1_DATA[11:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..18	-	Зарезервировано
27..16	DAC0 DATA[11:0]	Данные DAC0 при Cfg_SYNC_A=1. При чтении всегда равны нулю. Читать из DAC0_DATA
15..12	-	Зарезервировано
11…0	DAC1 DATA[11:0]	Данные DAC1

Примечание:
Если бит конфигурации Cfg_SYNC_A выставлен, то данные для DAC0 и DAC1 задаются записью в один из регистров DACx_DATA.

Контроллер интерфейса по ГОСТ 18977-79

Контроллер содержит в своём составе 8 приёмников и 4 передатчика по ГОСТ 18977-79 (далее ARINC). Каждый приёмник поддерживает функцию распознавания меток (или адресов). Для каждого приёмника может быть запрограммировано до 16 (32 с ревизии 3) 8-разрядных меток. Помимо этого фильтрация входных данных может осуществляться не только на базе меток, но и на базе двух бит Источник/Приёмник. Каждый передатчик поддерживает однонаправленную передачу 32-х разрядных слов по двухпроводной витой паре, используя формат кодирования RZ. Доступна возможность запрограммировать 32-й бит либо как данные, либо как бит паритета. В случае формирования бита паритета, программируется его чётность или нечётность. Каждый приёмник 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 приёма и передачи данных. Работа контроллера осуществляется на базовой частоте 1МГц, что позволяет обнаруживать ошибки в скорости приёма/передачи данных, а также в паузах между сообщениями.



Особенности:

- симплексный режим приёма/передачи со скоростями 12,5 кГц или 100 кГц;
- фильтрация входных данных на базе меток 16x8 (32x8 с ревизии 3) и двух бит Источник/Приёмник для каждого приёмника;
- возможность передачи 32 бита, как данных, так и паритета;
- выбор чётности/нечётности бита паритета;
- размеры буферов FIFO передатчиков: одно 256x32, три 64x32;
- размеры буферов FIFO приёмников: 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

Формат слова

Слова в интерфейсе ARINC всегда 32 разрядные, и включают в себя 5 полей: паритет, SSM, данные, источник/приёмник, метка. Биты передаются младшими разрядами вперёд, за исключением метки, которая передаётся старшими разрядами вперёд. 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 нечётным. Например, если биты 1-31 содержат чётное количество единиц, то бит паритета должен быть установлен в единицу, с другой стороны, если биты 1-31 содержат нечётное 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 в таблице.

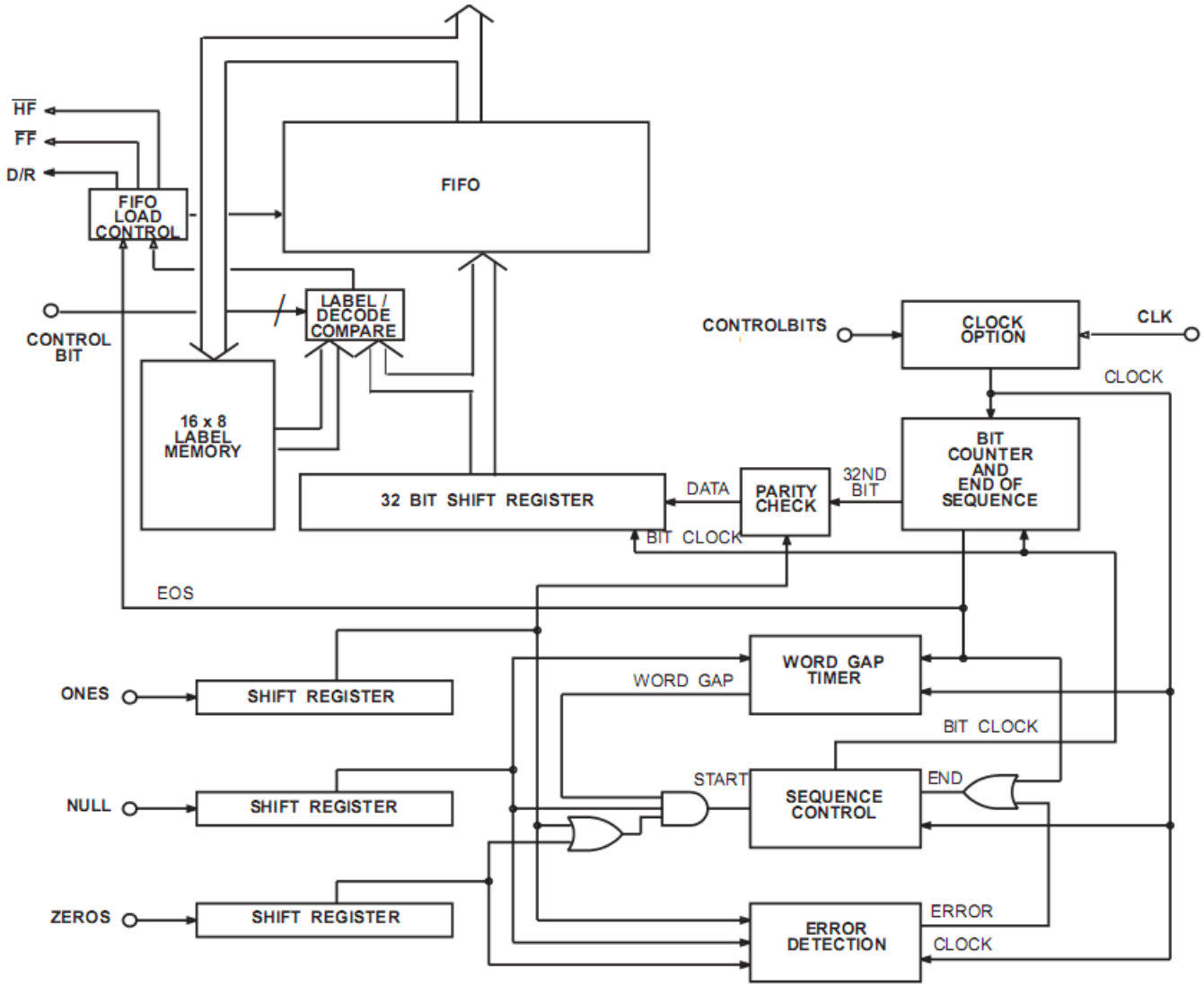
Бит		Значение
31	30	
0	0	плюс, север, восток, справа, к, выше
0	1	не вычислительные данные
1	0	функциональный тест
1	1	минус, юг, запад, слева, от, ниже

Биты 10 и 9 позволяют распознать Источник/Приёмник данных. Это применяется при нескольких приёмниках 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 приёмников. Если включена функция проверки этих бит, то 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 определённой в памяти меток приемного канала, то данные не помещаются в FIFO. Это может служить аналогом того, что приёмник не может интерпретировать метод обработки этих данных, следовательно, эти данные предназначены для другого приёмника.

В случае если приёмник принимает данные с неправильным битом паритета, они также не будут помещены в FIFO.

Структурная схема канала приёма



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 трёх 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, то таймер инкрементируется. Значение таймера равное трём разрешает следующий приём.

Схема паритета считает количество принятых единиц, включая бит паритета. Если результат нечётный, то на выходе схемы формируется сигнал равный нулю.

После того как приняты все 32 бита логика приёмника 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, то слово не загружается в FIFO. Ниже приведена таблица, показывающая, в каком случае происходит 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

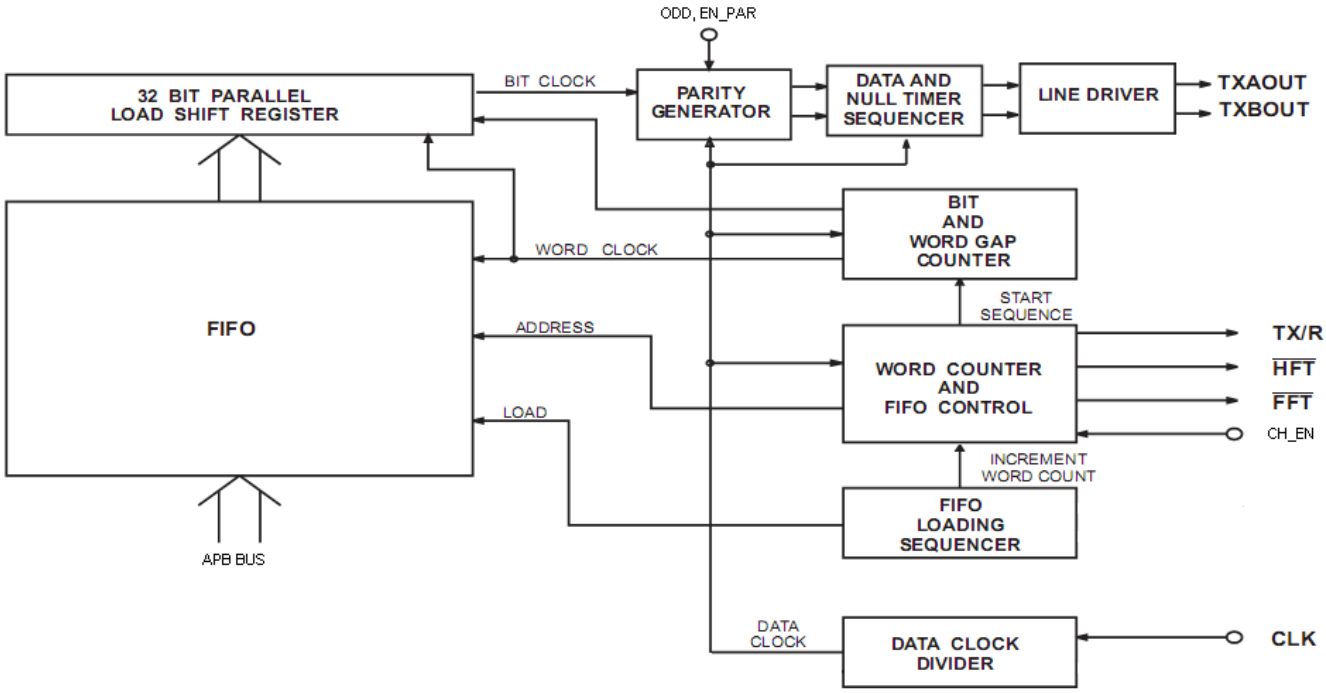
Если хотя бы одно слово загружено в FIFO, то устанавливается в единицу сигнал DR, что отражается в регистре статуса контроллера. Флаг остаётся в неизменном состоянии, пока последнее слово не будет прочитано из FIFO и оно не будет пустым. Помимо этого применяются ещё два сигнала характеризующие состояние 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.

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 полно, 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. Чётность или нечётность выбирается битом ODD регистра управления. Если бит разрешения паритета (EN_PAR) сброшен в ноль, то тогда 32 бит передается, как бит данных из FIFO.

Если бит CH_EN установлен в единицу и FIFO передачи не пусто, то начинается передача слов данных из FIFO до тех пор, пока FIFO не будет пусто или не будет сброшен бит CH_EN.



Описание регистров контроллера ГОСТ 18977-79

Базовый Адрес	Название	Описание
0x400D_0000	ARINC429R	Контроллер интерфейса приёмников ARINC429
Смещение		
0x0000	CONTROL1	Регистр управления 1 приёмников
0x0004	CONTROL2	Регистр управления 2 приёмников
0x0008	CONTROL3	Регистр управления 3 приёмников
0x000C	STATUS1	Регистр состояния 1 приёмников
0x0010	STATUS2	Регистр состояния 2 приёмников
0x0014	CONTROL4	Регистр настройки индивидуального делителя частоты каналов 1-4 с ревизии 2
0x0018	CONTROL5	Регистр настройки индивидуального делителя частоты каналов 5-8 с ревизии 2
0x001C	CHANNEL	Регистр номера канала приёмников
0x0020	LABEL	FIFO меток
0x0024	DATA_R	FIFO принимаемых данных
0x0030	DATA_R1	FIFO принимаемых данных канала 1 при CHANNEL=14 (с ревизии 3)
0x0034	DATA_R2	FIFO принимаемых данных канала 2 при CHANNEL=14 (с ревизии 3)
0x0038	DATA_R3	FIFO принимаемых данных канала 3 при CHANNEL=14 (с ревизии 3)
0x003C	DATA_R4	FIFO принимаемых данных канала 4 при CHANNEL=14 (с ревизии 3)

0x0040	DATA_R5	FIFO принимаемых данных канала 5 при CHANNEL=14 (с ревизии 3)
0x0044	DATA_R6	FIFO принимаемых данных канала 6 при CHANNEL=14 (с ревизии 3)
0x0048	DATA_R7	FIFO принимаемых данных канала 7 при CHANNEL=14 (с ревизии 3)
0x004C	DATA_R8	FIFO принимаемых данных канала 8 при CHANNEL=14 (с ревизии 3)
0x0068	INTMASK	Регистр индивидуальной настройки разрешения прерывания по заполненности FIFO каждого канала

Базовый Адрес	Название	Описание
0x400E_0000	ARINC429T	Контроллер интерфейса передатчиков ARINC429
Смещение		
0x0000	CONTROL1	Регистр управления 1 передатчиков
0x0004	CONTROL2	Регистр управления 2 передатчиков
0x0008	STATUS	Регистр состояния передатчиков
0x000C	DATA1_T	Регистр передаваемых данных канала 1
0x0010	DATA2_T	Регистр передаваемых данных канала 2
0x0014	DATA3_T	Регистр передаваемых данных канала 3
0x0018	DATA4_T	Регистр передаваемых данных канала 4
0x001C	CONTROL3	Регистр настройки индивидуального делителя частоты каналов с ревизии 2

CONTROL1

Регистр управления 1 приёмников

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
Значение после сброса	0	0	0	0	0	0	0	0
	CH_EN8	CH_EN7	CH_EN6	CH_EN5	CH_EN4	CH_EN3	CH_EN2	CH_EN1

Номер	15	14	13	12	11	10	9	8
Доступ*	R/W	R/W	U	U	U	U	U	U
Значение после сброса	0	0						
	CLK2	CLK1	-	-	-	-	-	-

Номер	23	22	21	20	19	18	17	16
Доступ*	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса			0	0	0	0	0	0
	-	-	CLK8	CLK7	CLK6	CLK5	CLK4	CLK3

Номер	31	30	29	28	27	26	25	24
Доступ*	R/W	R/W	R/W	R/W	U	U	U	U
Значение после сброса	0	0	0	0				
	DIV3	DIV2	DIV1	DIV0	-	-	-	-

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28	DIV[3:0]	**Делитель частоты ядра до 1 МГц** Содержит младшие 4 разряда значения, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц. Значение частоты не может быть более 125 МГц.
27..22		Зарезервировано
21..14	CLK8-CLK1	**Скорость приёма данных** 1- частота приёма данных= опорная частота/80 (12,5 кГц если DIV не равен 0) 0- частота приёма данных= опорная частота/10 (100 кГц если DIV не равен 0) Опорная частота для каждого канала задаётся делителем в регистрах CONTROL4 и CONTROL5, если DIV=0 с ревизии 2
13..8		Зарезервировано
7..0	CH_EN8-CH_EN1	**Разрешение работы канала** 1 – приём по каналу разрешён

		0 – канал приёма находится в состоянии сброса

CONTROL2

Регистр управления 2 приёмников

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
Значение после сброса	0	0	0	0	0	0	0	0
	LB_EN5	LB_EN4	LB_EN3	LB_EN2	LB_EN1	DIV6	DIV5	DIV4

Номер	15	14	13	12	11	10	9	8
Доступ*	U	U	U	U	U	R/W	R/W	R/W
Значение после сброса						0	0	0
	-	-	-	-	-	LB_EN8	LB_EN7	LB_EN6

Номер	23	22	21	20	19	18	17	16
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	U
Значение после сброса	0	0	0	0	0	0	0	
	SD_EN7	SD_EN6	SD_EN5	SD_EN4	SD_EN3	SD_EN2	SD_EN1	-

Номер	31	30	29	28	27	26	25	24
Доступ*	U	U	U	U	U	U	U	R/W
Значение после сброса								0
	DA	-	-	-	-	-	-	SD_EN8

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31	DA	**Бит прямого доступа в FIFO1 и FIFO2 (с ревизии 3)** 1 – память приёма каналов 1 и 2 работает не в режиме FIFO, доступ к ней осуществлется в диапазоне адресов 0x400D1000 - 0x400D13FC для FIFO1 0x400D1400 - 0x400D14FC для FIFO2 0 – обычный режим работы FIFO Адрес должен быть кратен четырём, обращения только по 32 разрядным словам. CHANEL= 0 или 1 в зависимости от канала При приёме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения.
30..25		Зарезервировано
24..17	SD_EN8 –	**Разрешение декодирования бит данных 9 и 10**

	SD_EN1	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
16..11		Зарезервировано
10..3	LB_EN8 – LB_EN1	**Разрешение обнаружения меток** 1 – разрешено обнаружение меток в первых 8 принятых битах 0 – обнаружение отключено, все принятые данные помещаются в FIFO
2..0	DIV[6:4]	**Делитель частоты ядра до 1 МГц** Содержит старшие 3 разряда значения, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц

CONTROL3

Регистр управления 3 приёмников

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
Значение после сброса	0	0	0	0	0	0	0	0
	SDI1_8	SDI1_7	SDI1_6	SDI1_5	SDI1_4	SDI1_3	SDI1_2	SDI1_1

Номер	15	14	13	12	11	10	9	8
Доступ*	R/W	R/W	U	U	U	U	U	U
Значение после сброса	0	0						
	SDI2_2	SDI2_1	-	-	-	-	-	-

Номер	23	22	21	20	19	18	17	16
Доступ*	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса			0	0	0	0	0	0
	-	-	SDI2_8	SDI2_7	SDI2_6	SDI2_5	SDI2_4	SDI2_3

Номер	31	30	29	28	27	26	25	24
Доступ*	R/W	R/W	R/W	R/W	U	U	U	U
Значение после сброса		0	0	0				
	INTEHF	INTEFF	INTEER	INTEDR	-	-	-	-

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31	INTEHF	**Разрешение прерывания FIFO наполовину полно** 1 – разрешено прерывание, если FIFO наполовину полно

		0 – прерывание запрещено
30	INTEFF	**Разрешение прерывания FIFO полно** 1 – разрешено прерывание при переполнении FIFO данных 0 – прерывание запрещено
29	INTEER	**Разрешение прерывания ошибка приёма** 1 – разрешено прерывания при возникновении ошибки в скорости приёма или во времени паузы 4T между сообщениями (для сброса ошибки необходимо сбросить канал битом CH_EN) 0 – прерывание запрещено
28	INTEDR	**Разрешение прерывания наличие данных в FIFO** 1 – разрешено прерывание, если FIFO приёма данных не пусто 0 – прерывание запрещено
27..22		Зарезервировано
21..14	SDI2_1– SDI2_8	**Бит сравнения SDI2** Значение бита сравнивается с битом 10 принимаемых данных, если установлен бит SD_EN соответствующего канала
13..8		Зарезервировано
7..0	SDI1_1 – SDI1_8	**Бит сравнения SDI1** Значение бита сравнивается с битом 9 принимаемых данных, если установлен бит SD_EN соответствующего канала

CONTROL4 (с ревизии 2)

Регистр управления 4 приёмников

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
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH1							

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
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH2							

Номер	23	22	21	20	19	18	17	16
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH3							

Номер	31	30	29	28	27	26	25	24
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH4							

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..24	DIV_CH4	Делитель частоты ядра для получения опорной частоты канала 4
23..16	DIV_CH3	Делитель частоты ядра для получения опорной частоты канала 3
15..8	DIV_CH2	Делитель частоты ядра для получения опорной частоты канала 2
7..0	DIV_CH1	Делитель частоты ядра для получения опорной частоты канала 1

CONTROL5 (с ревизии 2)

Регистр управления 5 приёмников

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
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH5							

	15	14	13	12	11	10	9	8
Номер								
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH6							

	23	22	21	20	19	18	17	16
Номер								
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH7							

	31	30	29	28	27	26	25	24
Номер								
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH8							

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..24	DIV_CH8	Делитель частоты ядра для получения опорной частоты канала 8
23..16	DIV_CH7	Делитель частоты ядра для получения опорной частоты канала 7
15..8	DIV_CH6	Делитель частоты ядра для получения опорной частоты канала 6
7..0	DIV_CH5	Делитель частоты ядра для получения опорной частоты канала 5

INTMASK (с ревизии 3)

Регистр индивидульного разрешения прерываний каналов

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
Значение после сброса	0	0	0	0	0	0	0	0
	IEHF2	IEFF2	IEER2	IEDR2	IEHF1	IEFF1	IEER1	IEDR1

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
Значение после сброса	0	0	0	0	0	0	0	0
	IEHF4	IEFF4	IEER4	IEDR4	IEHF3	IEFF3	IEER3	IEDR3

Номер	23	22	21	20	19	18	17	16
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	IEHF6	IEFF6	IEER6	IEDR6	IEHF5	IEFF5	IEER5	IEDR5

Номер	31	30	29	28	27	26	25	24
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	IEHF8	IEFF8	IEER8	IEDR8	IEHF7	IEFF7	IEER7	IEDR7

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..28		**Для канала 8**
27..24		**Для канала 7**
23..20		**Для канала 6**
19..16		**Для канала 5**
15..12		**Для канала 4**
11..8		**Для канала 3**
7..5		**Для канала 2**
3..0		**Для канала 1** **IEDR1** 1 – разрешено прерывание, если FIFO приёма данных не пусто 0 – прерывание запрещено **IEER1** 1 – разрешено прерывания при возникновении ошибки в скорости приёма или во времени паузы 4T между сообщениями (для сброса ошибки необходимо сбросить канал битом CH_EN) 0 – прерывание запрещено **IEFF1**

		1 – разрешено прерывание при переполнении FIFO данных 0 – прерывание запрещено **IEHF1** 1 – разрешено прерывание, если FIFO наполовину полно 0 – прерывание запрещено

STATUS1

Регистр состояния 1 приёмников

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
Значение после сброса	0	0	0	0	0	0	0	0
	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1

Номер	15	14	13	12	11	10	9	8
Доступ*	R/W	R/W	U	U	U	U	U	U
Значение после сброса	0	0						
	ERR2	ERR1	-	-	-	-	-	-

Номер	23	22	21	20	19	18	17	16
Доступ*	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса							0	0
	-	-	ERR8	ERR7	ERR6	ERR5	ERR4	ERR3

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..22		Зарезервировано
21..14	ERR1 – ERR8	**Бит ошибки.** 0 – нет ошибок 1 – возникла ошибка приёма
13..8		Зарезервировано
7..0	DR1 – DR8	**Бит наличия данных в FIFO.** 0 – FIFO пусто 1 – FIFO содержит данные

STATUS2

Регистр состояния 2 приёмников

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
Значение после сброса	0	0	0	0	0	0	0	0
	FF8	FF7	FF6	FF5	FF4	FF3	FF2	FF1

Номер	15	14	13	12	11	10	9	8
Доступ*	R/W	R/W	U	U	U	U	U	U
Значение после сброса	0	0						
	HF2	HF1	-	-	-	-	-	-

Номер	23	22	21	20	19	18	17	16
Доступ*	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса							0	0
	-	-	HF8	HF7	HF6	HF5	HF4	HF3

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..22		Зарезервировано
21..14	HF1 – HF8	**Бит наполненности FIFO .** 0 – FIFO не наполнено до половины 1 – FIFO наполнено до половины
13..8		Зарезервировано
7..0	FF1 – FF8	**Бит полноты FIFO.** 0 – FIFO не полно 1 – FIFO полно

CHANNEL

Регистр номера канала приёмников

Номер	7	6	5	4	3	2	1	0
Доступ*	U	U	U	U	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	-	-	-	-	CHAN3	CHAN2	CHAN1	CHAN0

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..4		Зарезервировано
3..0	CHAN[3:0]	**Биты выбора канала.** Значение этих бит определяет к данным и меткам, какого канала будет осуществляться доступ. 0000 – канал 1 0001 – канал 2 0010 – канал 3 0011 – канал 4 0100 – канал 5 0101 – канал 6 0110 – канал 7 0111 – канал 8 1110 – прямой доступ в память каналов 1 и 2 при DR=1, либо доступ по индивидуальным адресам для каждого FIFO (с ревизии 3)

LABEL

FIFO меток

FIFO меток, с которыми сравниваются первые 8 принимаемых бит, если установлен LB_EN бит соответствующего канала. Размер FIFO для каждого канала 16x8, либо 32x8 начиная с ревизии 3. Выбор необходимого FIFO осуществляется переключением канала в регистре CHANNEL. При записи или чтении FIFO указатель FIFO инкрементируется. Для возврата в начало FIFO необходимо осуществить запись в регистр CHANNEL.

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.

Выбор необходимого FIFO осуществляется переключением канала в регистре CHANNEL. Наличие или отсутствие данных в FIFO контролируется битами статуса DR, HF, FF соответствующего канала.

DATA_R1 - DATA_R8 (с ревизии 3)

FIFO принимаемых данных в случае записи в регистр CHANNEL значения 14.

CONTROL1

Регистр управления передатчиками 1

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
Значение после сброса	0	0	0	0	0	0	0	0
	ODD2	EN_PAR2	CLK2	CH_EN2	ODD1	EN_PAR1	CLK1	CH_EN1

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
Значение после сброса	0	0	0	0	0	0	0	0
	INTE_FFT1	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0

Номер	20	19	18	17	16
Доступ*	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0
	INTE_HFT2	INTE_TXR2	INTE_FFT2	INTE_HFT1	INTE_TXR1

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..21		Зарезервировано
20	INTE_HFT2	**Разрешение прерывания при заполнении наполовину буфера FIFO канала 2** 1 – разрешено прерывание, FIFO наполовину полно 0 – прерывание запрещено
19	INTE_TXR2	**Разрешение прерывания при опустошении буфера FIFO канала 2** 1 – разрешено прерывание, буфер FIFO пуст 0 – прерывание запрещено
18	INTE_FFT2	**Разрешение прерывания при полном заполнении буфера FIFO канала 2** 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
17	INTE_HFT1	**Разрешение прерывания при заполнении наполовину буфера FIFO канала 1** 1 – разрешено прерывание FIFO наполовину полно 0 –прерывание запрещено
16	INTE_TXR1	**Разрешение прерывания при опустошении буфера FIFO канала 1** 1 – разрешено прерывание FIFO передачи данных пусто 0 – прерывание запрещено
15	INTE_FFT1	**Разрешение прерывания при полном заполнении буфера FIFO канала 1**

		1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
14..8	DIV[6:0]	**Делитель частоты ядра до 1 МГц** Содержит значение, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц
7	ODD2	**Выбор чётности или нечётности бита паритета для канала 2** 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0)
6	EN_PAR2	**Разрешение 32 бита паритета для канала 2** 1 – разрешёна передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных
5	CLK2	**Скорость передачи данных по 2 каналу с ревизии 2** 1- частота передаваемых данных= опорная чатота/80 (12,5 кГц если DIV не равен нулю) 0- частота передаваемых данных= опорная частота/10 (100 кГц если DIV не равен нулю)
4	CH_EN2	**Разрешение работы канала 2** 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса
3	ODD1	**Выбор чётности или нечётности бита паритета для канала 1** 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0)
2	EN_PAR1	**Разрешение 32 бита паритета для канала 1** 1 – разрешёна передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных
1	CLK1	**Скорость передачи данных по 1 каналу с ревизии 2** 1- частота передаваемых данных= опорная чатота/80 (12,5 кГц если DIV не равен нулю) 0- частота передаваемых данных =опорная частота/10 (100 кГц если DIV не равен нулю)
0	CH_EN1	**Разрешение работы канала 1** 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса

CONTROL2

Регистр управления передатчиков 2

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
Значение после сброса	0	0	0	0	0	0	0	0
	ODD4	EN_PAR4	CLK4	CH_EN4	ODD3	EN_PAR3	CLK3	CH_EN3

Номер	15	14	13	12	11	10	9	8
Доступ*	R/W	U	U	U	U	U	U	U
Значение после сброса	0							
	INTE_FFT3	-	-	-	-	-	-	-

Номер	20	19	18	17	16
Доступ*	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0
	INTE_HFT4	INTE_TXR4	INTE_FFT4	INTE_HFT3	INTE_TXR3

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..21		Зарезервировано
20	INTE_HFT4	**Разрешение прерывания при заполнении наполовину буфера FIFO канала 4** 1 – разрешено прерывание, буфер FIFO наполовину полон 0 – прерывание запрещено
19	INTE_TXR4	**Разрешение прерывания при опустошении буфера FIFO канала 4** 1 – разрешено прерывание, буфер FIFO пуст 0 – прерывание запрещено
18	INTE_FFT4	**Разрешение прерывания при полном заполнении буфера FIFO канала 4** 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
17	INTE_HFT3	**Разрешение прерывания при заполнении наполовину буфера FIFO канала 3** 1 – разрешено прерывание, буфер FIFO наполовину полон 0 –прерывание запрещено
16	INTE_TXR3	**Разрешение прерывания при опустошении буфера FIFO канала 3** 1 – разрешено прерывание, буфер FIFO пуст 0 – прерывание запрещено
15	INTE_FFT3	**Разрешение прерывания при полном заполнении**

		буфера FIFO канала 3 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
14..8		Зарезервировано
7	ODD4	**Выбор чётности или нечётности бита паритета для канала 4** 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0)
6	EN_PAR4	**Разрешение 32 бита паритета для канала 4** 1 – разрешёна передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных
5	CLK4	**Скорость передачи данных по 4 каналу с ревизии 2** 1- частота передаваемых данных= опорная чатота/80 (12,5 кГц если DIV не равен нулю) 0- частота передаваемых данных= опорная частота/10 (100 кГц если DIV не равен нулю)
4	CH_EN4	**Разрешение работы канала 4** 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса
3	ODD3	**Выбор чётности или нечётности бита паритета для канала 3** 1 – бит паритета формируется как дополнение до нечётности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до чётности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0).
2	EN_PAR3	**Разрешение 32 бита паритета для канала 3** 1 – разрешёна передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных
1	CLK3	**Скорость передачи данных по 3 каналу с ревизии 2** 1- частота передаваемых данных= опорная чатота/80 (12,5 кГц если DIV не равен нулю) 0- частота передаваемых данных =опорная частота/10 (100 кГц если DIV не равен нулю)
0	CH_EN3	**Разрешение работы канала 3** 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса

CONTROL3

Регистр управления 3 передатчиков

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
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH1							
	15	14	13	12	11	10	9	8
Номер								
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH2							
	23	22	21	20	19	18	17	16
Номер								
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH3							
	31	30	29	28	27	26	25	24
Номер								
Доступ*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	0	0	0	0	0	0	0
	DIV_CH4							

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..24	DIV_CH4	Делитель частоты ядра для получения опорной частоты канала 4
23..16	DIV_CH3	Делитель частоты ядра для получения опорной частоты канала 3
15..8	DIV_CH2	Делитель частоты ядра для получения опорной частоты канала 2
7..0	DIV_CH1	Делитель частоты ядра для получения опорной частоты канала 1

STATUS

Регистр состояния передатчиков

Номер	15	14	13	12	11	10	9	8
Доступ*	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса			0	0	1	0	0	1
	-	-	HFT4	FFT4	TX_R4	HFT3	FFT3	TX_R3

Номер	7	6	5	4	3	2	1	0
Доступ*	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса			0	0	1	0	0	1
	-	-	HFT2	FFT2	TX_R2	HFT1	FFT1	TX_R1

R/W - бит доступен на чтение и запись
RO - бит доступен только на чтение
U - бит физически не реализован или зарезервирован.

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..14		Зарезервировано
13	HFT4	**Флаг наполненности FIFO канала 4** 1 – FIFO наполнено до половины 0 – FIFO не наполнено до половины
12	FFT4	**Флаг полноты FIFO канала 4** 1 – FIFO полно 0 – FIFO не полно
11	TX_R4	**Флаг наличия данных в FIFO канала 4** 1 – FIFO пусто 0 – FIFO содержит данные
10	HFT3	**Флаг наполненности FIFO канала 3** 1 – FIFO наполнено до половины 0 – FIFO не наполнено до половины
9	FFT3	**Флаг полноты FIFO канала 3** 1 – FIFO полно 0 – FIFO не полно
8	TX_R3	**Флаг наличия данных в FIFO канала 3** 1 – FIFO пусто 0 – FIFO содержит данные
7..6		Зарезервировано
5	HFT2	**Флаг наполненности FIFO канала 2** 1 – FIFO наполнено до половины 0 – FIFO не наполнено до половины
4	FFT2	**Флаг полноты FIFO канала 2** 1 – FIFO полно 0 – FIFO не полно
3	TX_R2	**Флаг наличия данных в FIFO канала 2** 1 – FIFO пусто

		0 – FIFO содержит данные
2	HFT1	**Флаг наполненности FIFO канала 1** 1 – FIFO наполнено до половины 0 – FIFO не наполнено до половины
1	FFT1	**Флаг полноты FIFO канала 1** 1 – FIFO полно 0 – FIFO не полно
0	TX_R1	**Флаг наличия данных в FIFO канала 1** 1 – FIFO пусто 0 – FIFO содержит данные

DATA1_T

FIFO передаваемых данных канала 1

FIFO может содержать данные объёмом 256x32 для передачи по каналу 1. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R1, HFT1, FFT1.

DATA2_T

FIFO передаваемых данных канала 2

FIFO может содержать данные объёмом 64x32 для передачи по каналу 2. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R2, HFT2, FFT2.

DATA3_T

FIFO передаваемых данных канала 3

FIFO может содержать данные объёмом 64x32 для передачи по каналу 3. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R3, HFT3, FFT3.

DATA4_T

FIFO передаваемых данных канала 4

FIFO может содержать данные объёмом 64x32 для передачи по каналу 4. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R4, HFT4, FFT4.

Контроллер SSP

Модуль порта синхронной последовательной связи (SSP – Synchronous Serial Port) выполняет функции интерфейса последовательной синхронной связи в режиме ведущего и ведомого устройства и обеспечивает обмен данными с подключенным ведомым или ведущим периферийным устройством в соответствии с одним из протоколов:

- интерфейс SPI фирмы Motorola;
- интерфейс SSI фирмы Texas Instruments;
- интерфейс Microwire фирмы National Semiconductor.

Как в ведущем, так и в ведомом режиме работы модуль SSP обеспечивает:

- преобразование данных, размещенных во внутреннем буфере FIFO передатчика (восемь 16-разрядных ячеек данных), из параллельного в последовательный формат;
- преобразование данных из последовательного в параллельный формат и их запись в аналогичный буфер FIFO приемника (восемь 16-разрядный ячеек данных).

Модуль формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов FIFO приемника и передатчика;
- переполнение буфера FIFO приемника;
- наличие данных в буфере FIFO приемника по истечении времени таймаута.

Основные сведения о модуле представлены в следующих разделах:

- характеристики интерфейса SPI;
- характеристики интерфейса Microwire;
- характеристики интерфейса SSI.

Основные характеристики модуля SSP

- может функционировать как в ведущем, так и в ведомом режиме;
- программное управление скоростью обмена;
- состоит из независимых буферов приема и передачи (8 ячеек по 16 бит) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);
- программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;
- программируемая длительность информационного кадра от 4 до 16 бит;
- независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по переполнению буфера приемника;
- доступна возможность тестирования по шлейфу, соединяющему вход с выходом;
- поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена далее – см. Рис. 1-1.

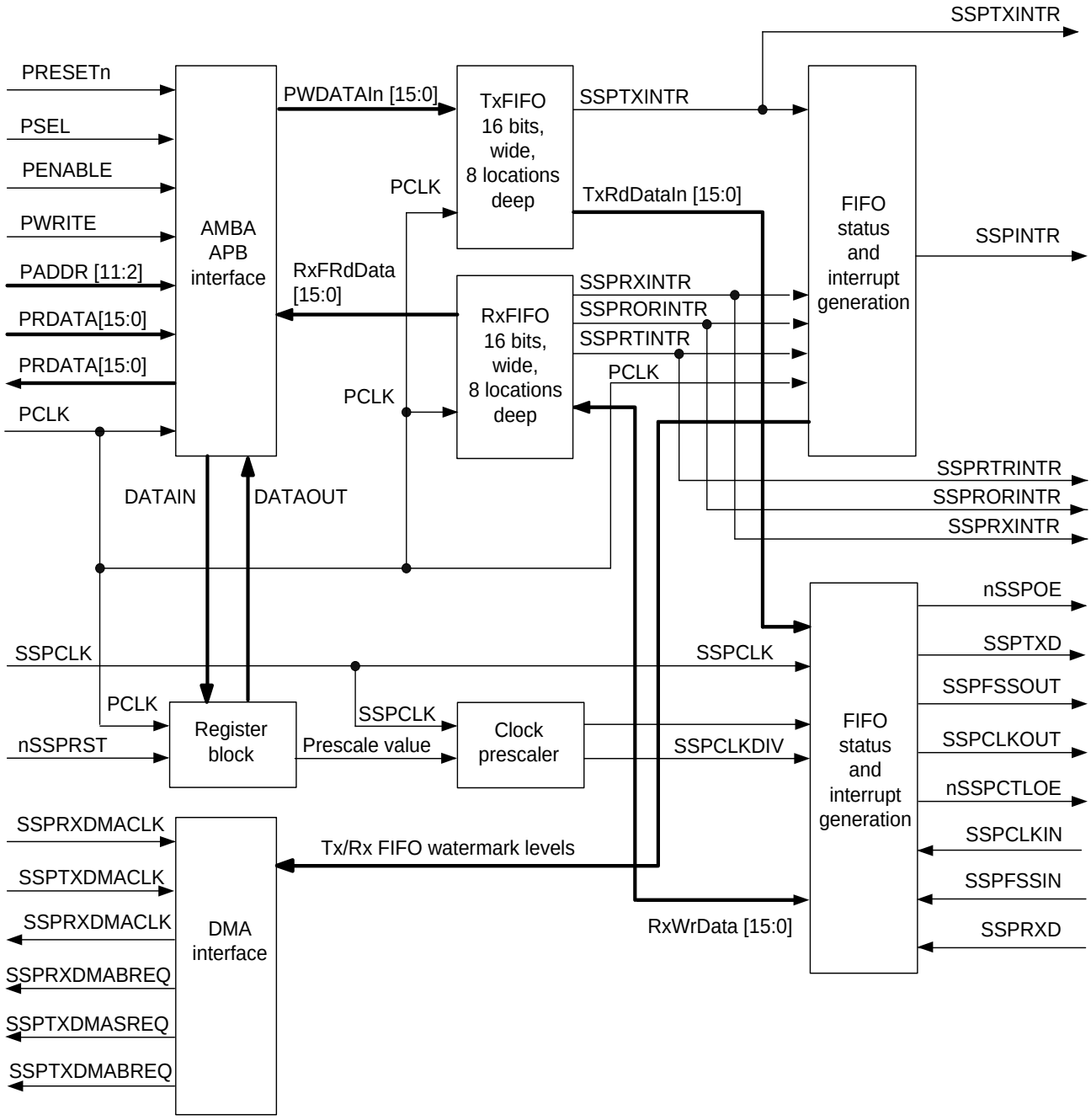


Рис. 1-1. Структурная схема модуля SSP

Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:
- режим функционирования периферийного устройства – ведущее или ведомое;
- разрешение или запрещение функционирования;
- формат информационного кадра;
- скорость передачи данных;
- фаза и полярность тактового сигнала;
- размер блока данных – от 4 до 16 бит;
- маскирование прерываний.

Характеристики интерфейса SPI

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:
- полнодуплексный обмен данными по четырехпроводной линии;
- программное задание фазы и полярности тактового сигнала.

Характеристики интерфейса Microwire

Интерфейс Microwire фирмы National Semiconductor обеспечивает:
- полудуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

Характеристики интерфейса SSI

Интерфейс SSI фирмы Texas Instruments обеспечивает:
- полнодуплексный обмен данными по четырехпроводной линии;
- возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

Общий обзор модуля SSP

Модуль SSP представляет собой интерфейс синхронного последовательного обмена данными способный функционировать в качестве ведущего или ведомого устройства, поддерживающий протоколы передачи данных SPI фирмы Motorola, Microwire фирмы National Semiconductor, а также SSI фирмы Texas Instruments.

Модуль выполняет следующие функции:

- преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;
- преобразование данных, передаваемых на периферийное устройство, из параллельной и последовательную форму;
- центральный процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии;
- прием и передача данных буферизуются с помощью буферов FIFO и обеспечивают хранение до восьми слов данных шириной до 16 бит независимо для режимов приема и передачи.

Последовательные данные передаются по линии SSP_TXD и принимаются с линии SSP_RXD.

Модуль SSP содержит программируемые делители частоты, формирующие тактовый сигнал обмена данными SSP_CLK из сигнала, поступающего на линию SSPCLK. Скорость передачи данных может быть более 2 МГц, в зависимости от частоты SSPCLK и характеристик подключенного периферийного устройства.

Режим обмена данными, формат информационного кадра и количество бит данных задаются программно с помощью регистров управления CR0 и CR1.

Модуль формирует четыре независимо маскируемых прерывания:

SSPTXINTR – запрос на обслуживание буфера передатчика;

SSPRXINTR – запрос на обслуживание буфера приемника;

SSPRORINTR – переполнение приемного буфера FIFO;

SSPRTINTR – таймаут ожидания чтения данных из приемного FIFO.

Кроме того, формируется общий сигнал прерывания SSPINTR, возникающий в случае активности одного из вышеуказанных независимых немаскированных прерываний, который идет на контроллер NVIC.

Модуль также формирует сигналы запроса на прямой доступ к памяти (DMA) для совместной работы с контроллером DMA.

В зависимости от режима работы модуля сигнал SSPFSSOUT используется либо для кадровой синхронизации (интерфейс SSI, активное состояние – высокий уровень), либо для выбора ведомого режима (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

Блок формирования тактового сигнала

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSP_CLK с помощью внутреннего делителя частоты, состоящего из двух последовательно соединенных счетчиков без цепи сброса.

Коэффициент предварительного деления частоты в диапазоне от 2 до 254 с шагом 2 можно задать путем записи значения в регистр SSPCPSR. Так как младший значащий разряд коэффициента деления не используется, то исключается возможность деления частоты на нечетный коэффициент деления. Это, в свою очередь, гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровней).

Сформированный, описанным образом, сигнал далее поступает на второй делитель частоты, с выхода которого и снимается тактовый сигнал обмена данными SSP_CLK.

Коэффициент деления второго делителя задается программно в диапазоне от 1 до 256 путем записи соответствующего значения в регистр управления SSPCR0.

Буфер FIFO передатчика

Буфер передатчика имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Данные от центрального процессора сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных.

Буфер FIFO приемника

Буфер приемника имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Принятые от периферийного устройства данные сохраняются в этом буфере блоком приема данных до тех пор, пока не будут считаны центральным процессором.

Блок приема и передачи данных

Режим ведущего устройства.

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSP_CLK для подключенных ведомых устройств. Как было описано ранее, данный сигнал формируется путем деления частоты сигнала SSPCLK.

Блок передатчика последовательно считывает данные из буфера FIFO передатчика и производит их преобразование из параллельной формы в последовательную. Далее поток последовательных данных и элементов кадровой синхронизации, тактированный сигналом SSP_CLK, передаётся по линии SSP_TXD к подключенным ведомым устройствам.

Блок приемника выполняет преобразование данных, поступающих синхронно с линии SSP_RXD, из последовательной в параллельную форму. После этого загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

Режим ведомого устройства.

В режиме ведомого устройства тактовый сигнал обмена данными формируется одним из подключенных к модулю периферийных устройств и поступает по линии SSP_CLK.

При этом блок передатчика, тактируемый этим внешним сигналом, считывает данные из буфера FIFO, преобразует их из параллельной формы в последовательную. После этого выдает поток последовательных данных и элементов кадровой синхронизации в линию SSP_TXD.

Аналогично, блок приемника выполняет преобразование данных, поступающих с линии SSP_RXD синхронно с сигналом SSP_CLK, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

Блок формирования прерываний

Модуль SSP генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения указанных независимых прерываний по схеме ИЛИ.

Комбинированный сигнал прерывания подаётся на контроллер прерываний NVIC, при этом появляется дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

Конфигурирование приемопередатчика

После сброса работа блоков приемопередатчика запрещается до выполнения процедуры задания конфигурации.

Для этого необходимо выбрать ведущий или ведомый режим работы устройства, а также используемый протокол передачи данных (SPI фирмы Motorola, SSI фирмы Texas Instruments, либо Microwave фирмы National Semiconductor), после чего записать необходимую информацию в регистры управления CR0 и CR1.

Кроме того, для установки требуемой скорости передачи данных необходимо выбрать параметры блока формирования тактового сигнала с учетом значения частоты сигнала SSPCLK и записать соответствующую информацию в регистр PSR.

Разрешение работы приемопередатчика

Разрешение осуществляется путем установки бита SSE регистра управления CR1. Буфер FIFO передатчика может быть либо проинициализирован путем записи в него до восьми 16-разрядных слов заблаговременно перед установкой этого бита, либо может заполняться передаваемыми данными в процедуре обслуживания прерывания.

После разрешения работы модуля приемопередатчик начинает обмен данными по линиям SSP_TXD и SSP_RXD.

Соотношения между тактовыми сигналами

В модуле имеется ограничение на соотношение между частотами тактовых сигналов CPU_CLK и SSPCLK. Частота SSPCLK должна быть меньше или равна частоте CPU_CLK. Выполнение этого требования гарантирует синхронизацию сигналов управления, передаваемых из зоны действия тактового сигнала SSPCLK в зону действия сигнала CPU_CLK в течение времени, меньшего продолжительности передачи одного информационного кадра:

```
FSSPCLK <= FPCLK.
```

В режиме ведомого устройства сигнал SSP_CLK от ведущего внешнего устройства поступает на схемы синхронизации, задержки и обнаружения фронта. Для того чтобы обнаружить фронт сигнала SSP_CLK, необходимо три такта сигнала SSP_CLK. Сигнал SSP_TXD имеет меньшее время установки по отношению к заднему фронту SSP_CLK, по которому и происходит считывание данных из линии. Время установки и удержания сигнала SSP_RXD по отношению к сигналу SSP_CLK должно выбираться с запасом, гарантирующим правильное считывание данных. Для обеспечения корректной работы устройства необходимо, чтобы частота SSPCLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSP_CLK.

Выбор частоты тактового сигнала SSPCLK должен обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала SSPCLK к максимальной частоте сигнала SSP_CLK в режиме ведомого устройства равно 12, в режиме ведущего – двум.

Так в режиме ведущего устройства для обеспечения максимальной скорости обмена 1.8432 Мбит/с частота сигнала SSPCLK должна составлять не менее 3.6864 МГц. В этом случае в регистр CPSR должно быть записано значение 2, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

В режиме ведомого устройства для обеспечения той же информационной скорости необходимо использовать тактовый сигнал SSPCLK с частотой не менее 22.12 МГц. При этом в регистр CPSR должно быть записано значение 12, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

Соотношение между максимальной частотой сигнала SSPCLK и минимальной частотой SSPCLKOUT составляет 254 * 256.

Минимальная допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:

```
FSSPCLK(min) => 2 x FSSPCLKOUT(max) [for master mode]
FSSPCLK(min) => 12 x FSSPCLKIN(max) [for slave mode].
```

Аналогично, максимальная допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:

```
FSSPCLK(max) <= 254 x 256 x FSSPCLKOUT(min) [for master mode]
FSSPCLK(max) <= 254 x 256 x FSSPCLKIN(min) [for slave mode].
```

Программирование регистра управления SSPCR0

Регистр CR0 предназначен для:

- установки скорости информационного обмена;
- выбора одного из трех протоколов обмена данными;
- выбора размера слова данных.

Скорость информационного обмена зависит от частоты внешнего тактового сигнала SSPCLK и коэффициента деления блока формирования тактового сигнала. Последний задается совместно значением поля SCR (Serial Clock Rate – скорость информационного обмена) регистра SSPCR0 и значением поля CPSDVSR (clock prescale divisor value – коэффициент деления тактового сигнала) регистра SSPCPSR.

Формат информационного кадра задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS регистра SSPCR0.

Для протокола SPI фирмы Motorola также задаются полярность и фаза сигнала (биты SPH и SPO).

Программирование регистра управления SSPCR1

Регистр SSPCR1 предназначен для:

- выбора ведущего или ведомого режима функционирования приемопередатчика;
- включения режима проверки канала по шлейфу;
- разрешения или запрещения работы модуля.

Выбор ведущего режима осуществляется путем записи 0 в поле MS регистра SSPCR1 (это значение устанавливается после сброса автоматически).

Запись 1 в поле MS переводит приемопередатчик в режим ведомого устройства. В этом режиме разрешение или запрещение формирования сигнала передатчика SSP_TXD осуществляется путем установки бита SOD (slave mode SSP_TXD output disable – запрет линии SSP_TXD для ведомого режима) регистра CR1. Указанная функция полезна при подключении к одной линии нескольких подчиненных устройств.

Для того чтобы разрешить функционирование приемопередатчика, необходимо установить в 1 бит SSE (Synchronous Serial Port Enable – разрешение последовательного синхронного порта).

Формирование тактового сигнала обмена данными

Тактовый сигнал обмена данными формируется путем деления частоты тактового сигнала SSPCLK. На первом этапе формирования частота этого сигнал делится на четный коэффициент CPSDVSR, лежащий в диапазоне от 2 до 254, доступный для программирования через регистр CPSR. Сформированный сигнал далее поступает на делитель частоты с коэффициентом (1 + SCR) от 1 до 256, где значение SCR доступно для программирование через CR0.

Частота выходного тактового сигнала обмена данными SSP_CLK определяется следующим соотношением:

```
FSSPCLKOUT = FSSPCLK / (CPSDVR * (1+SCR)).
```

Например, в случае, если частота сигнала SSPCLK составляет 3.6864 МГц, а значение CPSDVSR = 2, частота сигнала SSP_CLK лежит в интервале от 7.2 кГц до 1.8432 МГц.

Формат информационного кадра

Каждый информационный кадр содержит в зависимости от запрограммированного значения от 4 до 16 бит данных. Передача данных начинается со старшего значащего разряда. Есть возможность выбрать три базовых структуры построения кадра:

- SSI фирмы Texas Instruments;
- SPI фирмы Motorola;
- Microwire фирмы National Semiconductor.

Во всех трех режимах построения кадра тактовый сигнал SSP_CLK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SSP_CLK в неактивное состояние используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

В режимах SPI и Microwire выходной сигнал кадровой синхронизации передатчика SSP_FSS имеет активный низкий уровень и поддерживается в низком уровне в течение всего периода передачи информационного кадра.

В режиме построения кадра SSI фирмы Texas Instruments перед началом каждого информационного кадра на выходе SSP_FSS формируется импульс с длительностью, равной одному тактовому интервалу обмена данными. В этом режиме приемопередатчик SSP, равно как и ведомые периферийные устройства, передаёт данные в линию по переднему фронту сигнала SSP_CLK, а считывает данные из линии по заднему фронту этого сигнала.

В отличие от полнодуплексных режимов передачи данных SSI и SPI, режим Microwire фирмы National Semiconductor использует специальный способ обмена данными между ведущим и ведомым устройством, функционирующий в режиме полудуплекса. В указанном режиме на внешнее ведомое устройство перед началом передачи информационного кадра посылается специальная восьмибитная управляющая последовательность. В течение всего времени передачи этой последовательности приемник не обрабатывает каких-либо входных данных. После того как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал

после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом общая длительность информационного кадра составляет от 13 до 25 бит.

Формат синхронного обмена SSI фирмы Texas Instruments

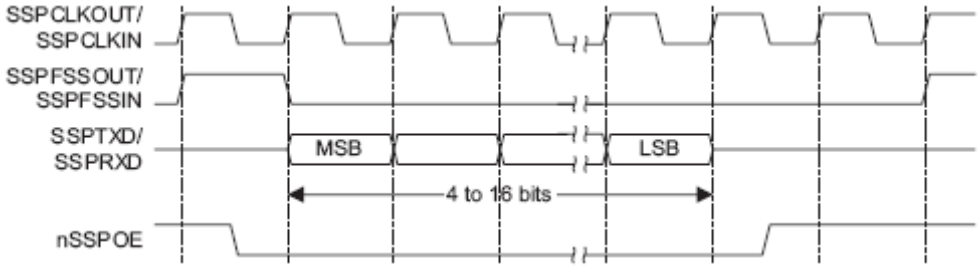


Рис. 2-2. Формат синхронного обмена протокола SSI фирмы Texas Instruments (единичный обмен)

В данном режиме при неактивном приемопередатчике SSP сигналы SSP_CLK и SSP_FSS переводятся в низкий логический уровень, а линия передачи данных SSP_TXD поддерживается в третьем состоянии.
После появления хотя бы одного элемента в буфере FIFO передатчика сигнал SSPFSSOUT переводится в высокий логический уровень на время, соответствующее одному периоду сигнала SSP_CLK. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSP_CLK старший значащий разряд информационного кадра (4 – 16 бит данных) выдается на выход линии SSP_TXD и т.д.
В режиме приема данных как модуль SSP, так и ведомое внешнее устройство последовательно загружают биты данных в сдвиговый регистр по заднему фронту сигнала SSP_CLK. Принятые данные переносятся из сдвигового регистра в буфер FIFO после загрузки в него младшего значащего бита данных по очередному переднему фронту сигнала SSP_CLK.
Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments представлены на рис. 2-2 (передача единичного информационно кадра) и рис. 2-3 (передача последовательности кадров).

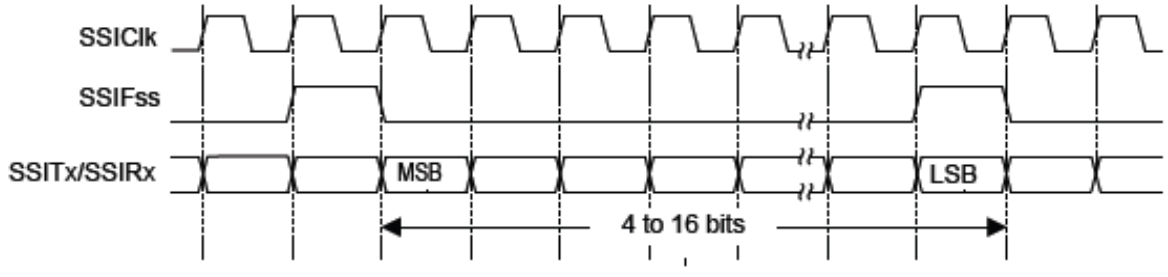


Рис. 2-3. Формат синхронного обмена протокола SSI фирмы Texas Instruments (непрерывный обмен)

Формат синхронного обмена SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSP_FSS выполняет функцию выбора ведомого устройства. Главной

особенностью протокола SPI является возможность выбора состояния и фазы сигнала SSP_CLK в режиме ожидания (неактивном приемопередатчике) путем задания значений бит SPO и SPH регистра управления SSPSCR0.

Выбор полярности тактового сигнала – бит SPO

Если бит SPO равен 0, то в режиме ожидания линия SSP_CLK переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SSP_CLK переводится в высокий логический уровень.

Выбор фазы тактового сигнала – бит SPH

Значение бита SPH определяет фронт тактового сигнала, по которому осуществляется выборка данных и изменение состояния на выходе линии.

В случае если бит SPH установлен в 0, регистрация данных приемником осуществляется после первого обнаружения фронта тактового сигнала, в противном случае – после второго.

Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=0 показаны на рис. 2-4 (одиночный обмен) и рис. 2-5 (непрерывный обмен).

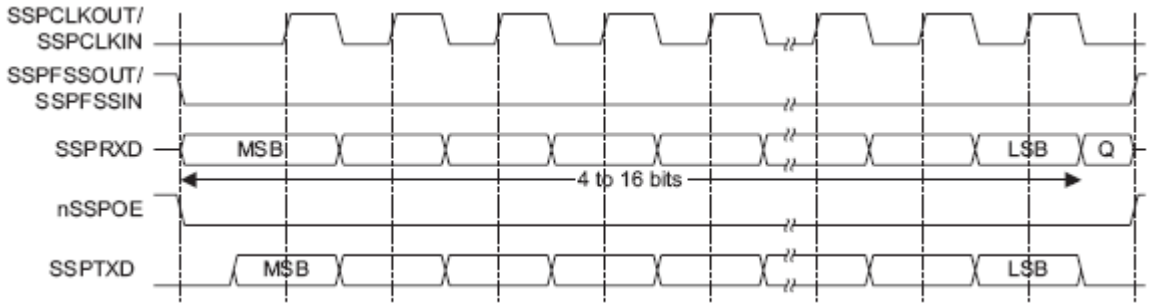


Рис. 2-4. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=0 (одиночный обмен)

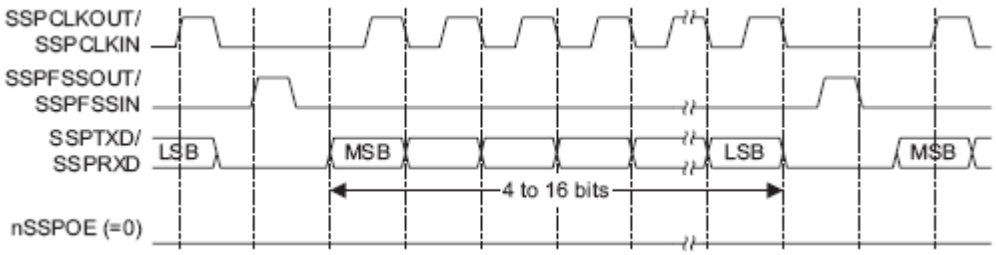


Рис. 2-5. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0, SPH=0 (непрерывный обмен)
Примечание. На рис. 2-4 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP_CLK имеет низкий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;
- сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Контакт передатчика SSP_TXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP_CLK на линии SSP_TXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSP_CLK переводится в высокий логический уровень.

Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP_CLK.

В режиме непрерывной передачи данных на линии SSP_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP_FSS в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSP_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP_CLK.

Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=1 показаны на рис. 2-6 (одиночный и непрерывный обмен).

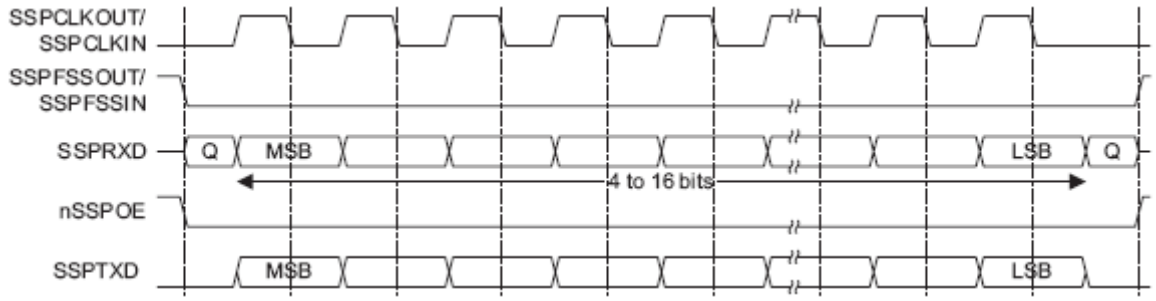


Рис. 2-6. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=1

Примечание. На рис. 2-6 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP_CLK имеет низкий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;
- сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на

начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Выходной контакт передатчика SSP_TXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP_CLK на линиях обмена как ведущего, так и ведомого устройств будут сформированы значения первых бит передаваемых данных. В это же время включается линия SSP_CLK и на ней формируется передний фронт сигнала.

Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSP_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP_CLK.

В режиме непрерывной передачи данных линия SSP_FSS постоянно находится в низком логическом уровне, и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=0 показаны на рис. 2-7 (одиночный обмен) и рис. 2-8 (непрерывный обмен).

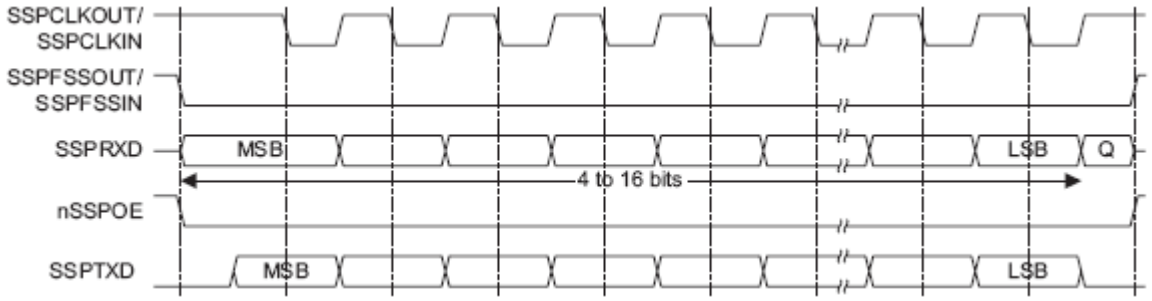


Рис. 2-7. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)

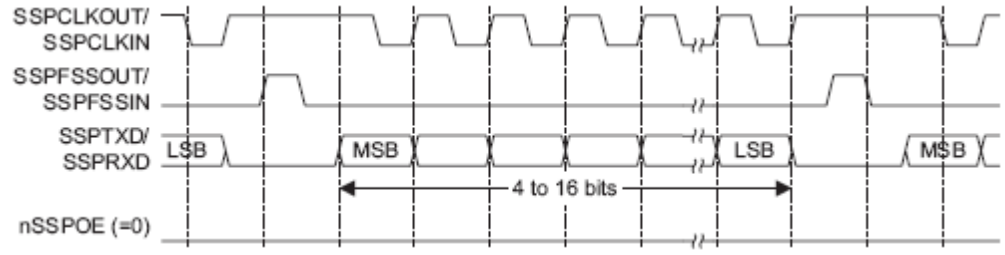


Рис. 2-8. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (непрерывный обмен)

Примечание. На рис. 2-7 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP_CLK имеет высокий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;

– сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Выходной контакт передатчика SSP_TXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP_CLK, на линии SSP_TXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSP_CLK переводится в низкий логический уровень.

Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSP_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP_CLK.

В режиме непрерывной передачи данных на линии SSP_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP_FSS в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSP_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP_CLK.

Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=1 показаны на рис. 2-9 (одиночный и непрерывный обмен).

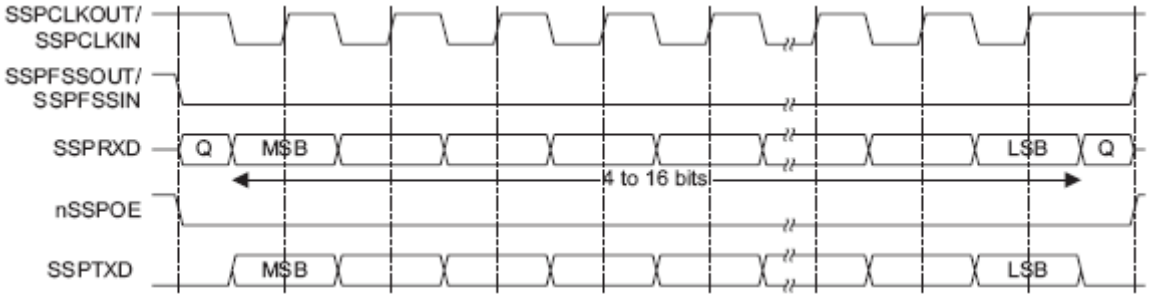


Рис. 2-9. Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=1

Примечание. На рис. 2-9 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

– сигнал SSP_CLK имеет высокий логический уровень;
– сигнал SSP_FSS имеет высокий логический уровень;
– сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную

линию SSP_RXD ведущего. Выходной контакт передатчика SSP_TXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP_CLK на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых бит передаваемых данных. В это же время включается линия SSP_CLK и на ней формируется передний фронт сигнала.

Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP_CLK.

В режиме непрерывной передачи данных линия SSP_FSS постоянно находится в низком логическом уровне и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

Формат синхронного обмена Microwire фирмы National Semiconductor

Временные диаграммы последовательного синхронного обмена в режиме Microwire показаны на рис. 2-10 (одиночный обмен) и рис. 2-11 (непрерывный обмен).

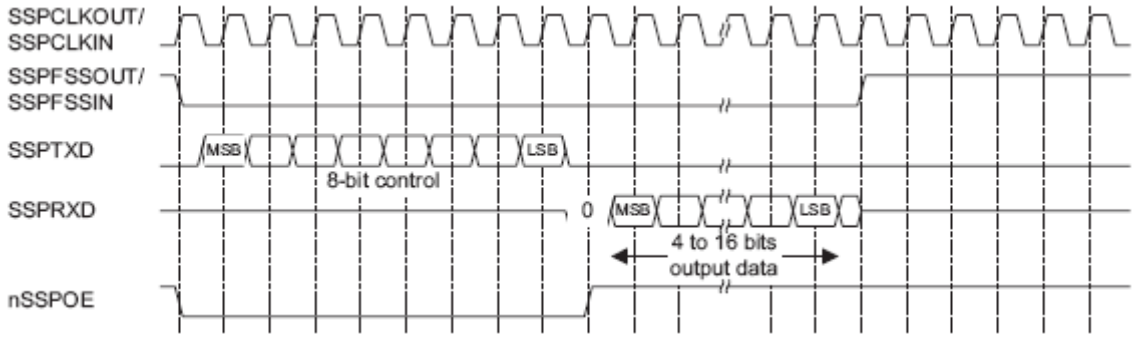


Рис. 2-10. Формат синхронного обмена протокола Microwire фирмы National Semiconductor (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен в нем осуществляется в полудуплексном режиме, с использованием служебных последовательностей. Каждая информационный обмен начинается с передачи ведущим устройством специальной восьмибитной управляющей последовательности. В течение всего времени ее передачи приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

В данном режиме во время ожидания приемопередатчика:
- сигнал SSP_CLK имеет низкий логический уровень;
- сигнал SSP_FSS имеет высокий логический уровень;
- сигнал SSP_TXD переводится в высокоимпедансное состояние.

Переход в режим информационного обмена происходит после записи управляющего байта в буфер FIFO передатчика. По заднему фронту сигнала SSP_FSS данные из буфера переносятся в регистр сдвига блока передатчика, откуда, начиная со старшего значащего разряда, последовательно выдаются в линию SSP_TXD. Линия SSP_FSS остается в низком логическом уровне в течение всей передачи кадра. Линия SSP_RXD при этом находится в высокоимпедансном состоянии.

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSP_CLK. По окончании приема последнего бита управляющей последовательности она декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля SSP. Биты данных выдаются в линию SSP_RXD по заднему фронту сигнала SSP_CLK. Ведущее устройство в свою очередь регистрирует их по переднему фронту этого тактового сигнала. В случае одиночного информационного обмена по окончании приема последнего бита слова данных сигнал SSP_FSS переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных из регистра сдвига в буфер FIFO приемника.

Примечание. Внешнее устройство может перевести линию приемника в третье состояние по заднему фронту сигнала SSP_CLK после приема последнего бита слова данных, либо после перевода линии SSP_FSS в высокий логический уровень.

Непрерывный обмен данными начинается и заканчивается так же, как и одиночный обмен. Однако линия SSP_FSS удерживается в низком логическом уровне в течение всего сеанса передачи данных. Управляющий байт следующего информационного кадра передается сразу же после приема младшего значащего разряда текущего кадра. Данные из сдвигового регистра передаются в буфер приемника после регистрации младшего разряда очередного слова по заднему фронту сигнала SSP_CLK.

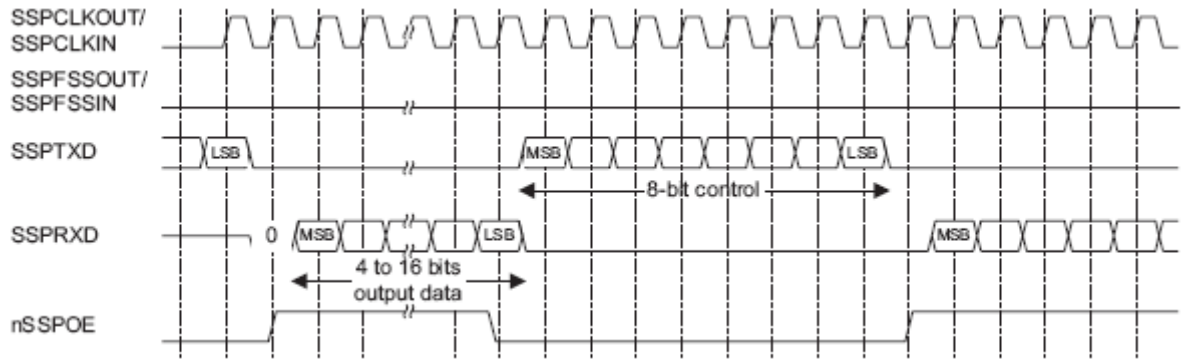


Рис. 2-11. Формат синхронного обмена протокола Microwire фирмы National Semiconductor (непрерывный обмен)

Требования к временным параметрам сигнала SSP_FSS относительно тактового сигнала SSP_CLK в режиме Microwire

Модуль SSP, работающий в режиме Microwire как ведомое устройство, регистрирует данные по переднему фронту сигнала SSP_CLK после установки сигнала SSP_FSS в низкий логический уровень. Ведущие устройства, формирующие сигнал SSP_CKL, должны гарантировать достаточное время установки и удержания сигнала SSP_FSS по отношению к переднему фронту сигнала SSP_CLK.

Данные требования иллюстрирует Рисунок 2-12. По отношению к переднему фронту сигнала SSP_CLK, по которому осуществляется регистрация данных в приемнике ведомого модуля SSP, время установки сигнала SSP_FSS должно быть как минимум в два раза больше периода SSP_CLK, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSP_CLK должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.

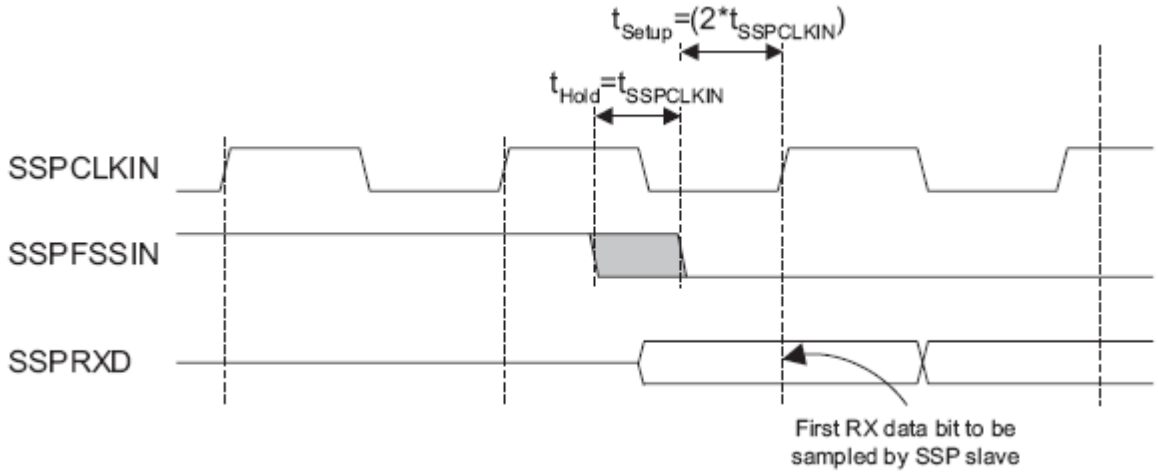


Рис. 2-12. Формат кадра Microwire, требования к времени установки и удержания сигнала SSPFSSIN

Примеры конфигурации модуля в ведущем и ведомом режимах

На рис. 2-13, 2-14 и 2-15 показаны варианты подключения модуля PrimeCell SSP (PL022) к периферийным устройствам, работающим в ведущем или ведомом режиме.

Примечание. Модуль SSP не поддерживает динамическое изменение режима ведущий/ведомый. Каждый приемопередатчик должен быть изначально сконфигурирован в одном из этих режимов.

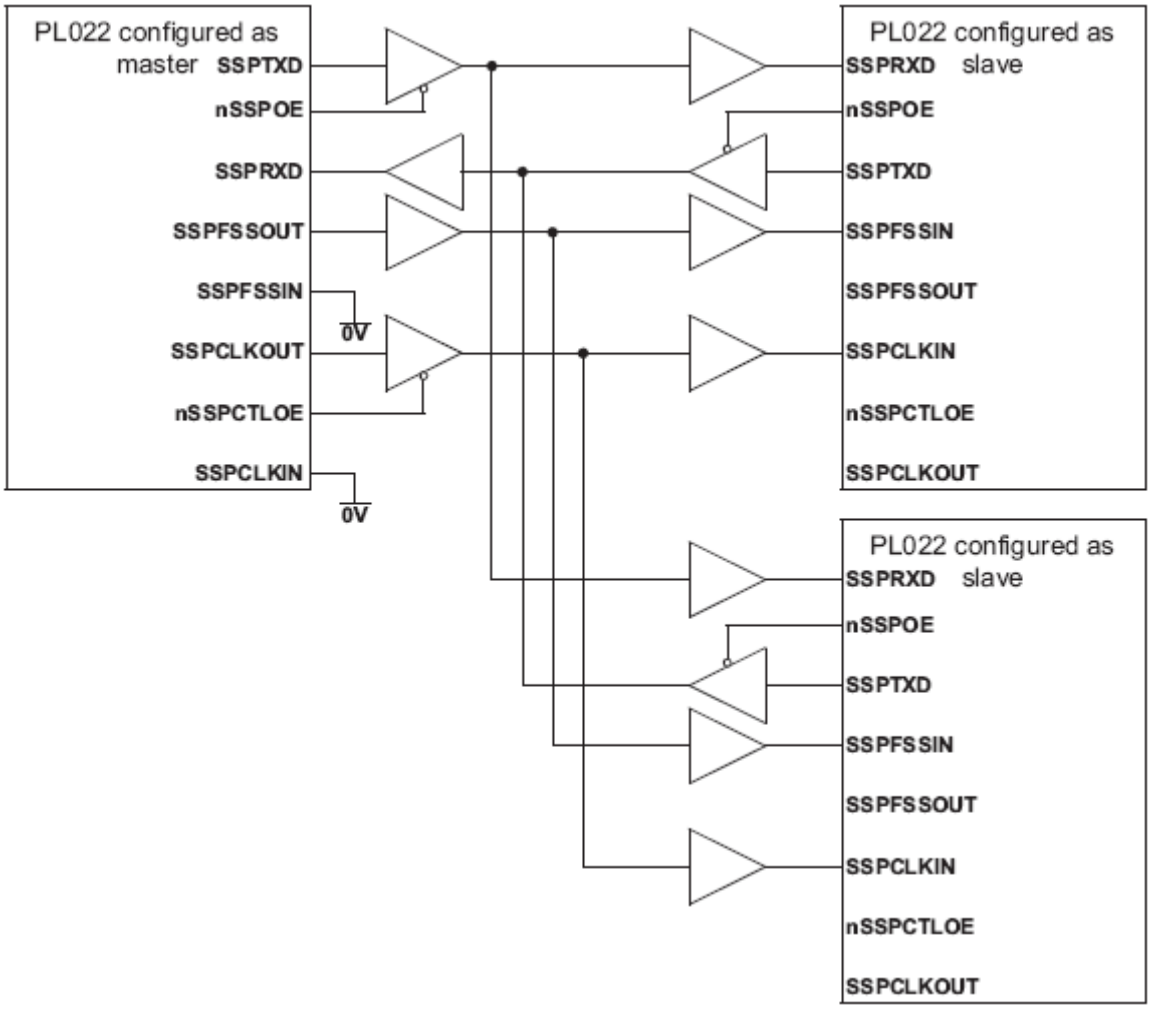


Рис. 2-13. Ведущее устройство SSP подключено к двум ведомым.

Рис. 2-13 показывает совместную работу трех модулей SSP, один из которых сконфигурирован в качестве ведущего, а два – в качестве ведомых устройств. Ведущее устройство способно передавать данные циркулярно в адрес двух ведомых по линии SSP_TXD.

Для ответной передачи данных один из ведомых модулей разрешает прохождение сигнала от своей линии SSP_TXD на вход SSP_RXD ведущего.

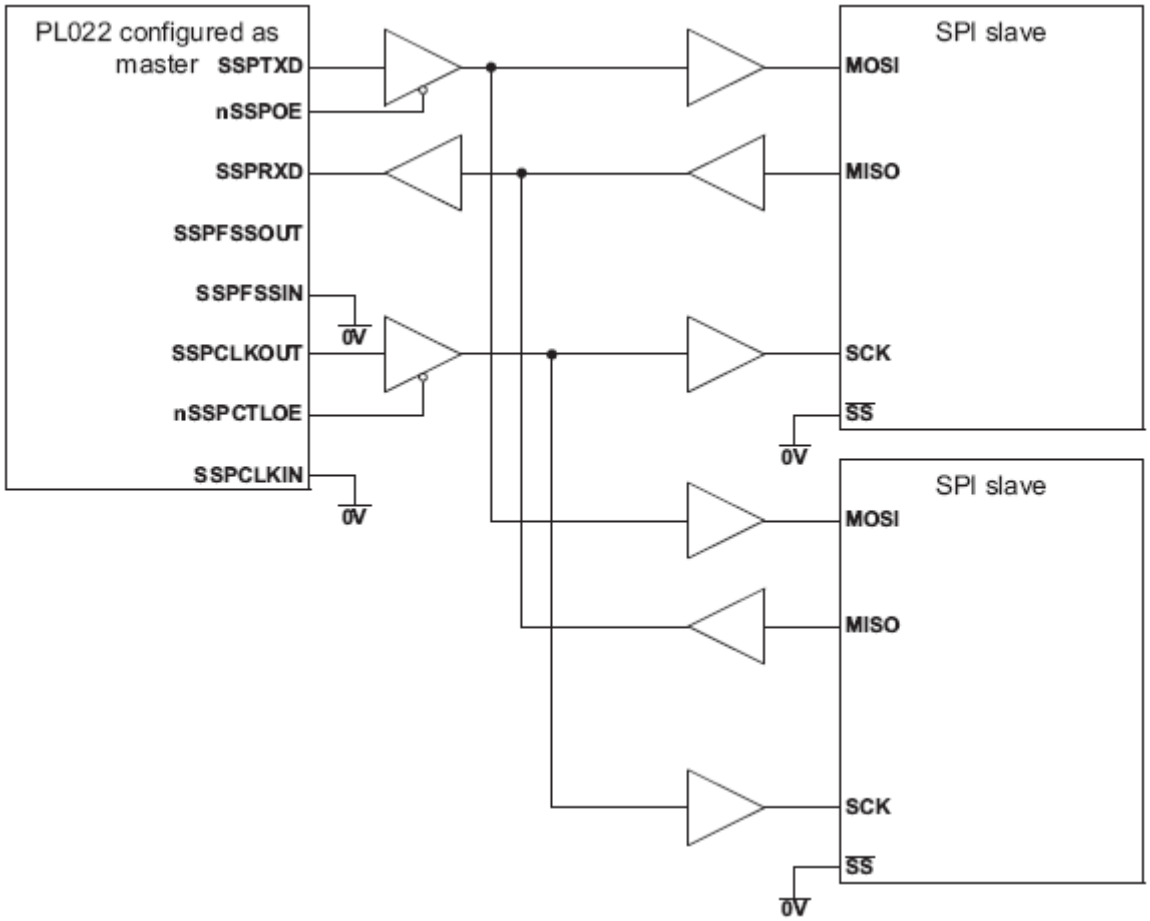


Рис. 2-14. Ведущее устройство SSP подключено к двум ведомым, поддерживающим протокол SPI.

Рис. 2-14 показывает подключение модуля SSP, сконфигурированного как ведущее устройство, к двум ведомым устройствам, поддерживающим протокол SPI фирмы Motorola. Внешние устройства сконфигурированы как ведомые путем установки в низкий логический уровень сигнала выбора ведомого устройства Slave Select (SS). Как и в предыдущем примере, ведущее устройство способно передавать данные в адрес ведомых циркулярно по линии SSP_TXD. Ответная передача данных на входную линю SSP_RXD ведущего устройства одновременно осуществляется только одним из ведомых по соответствующей линии MISO.

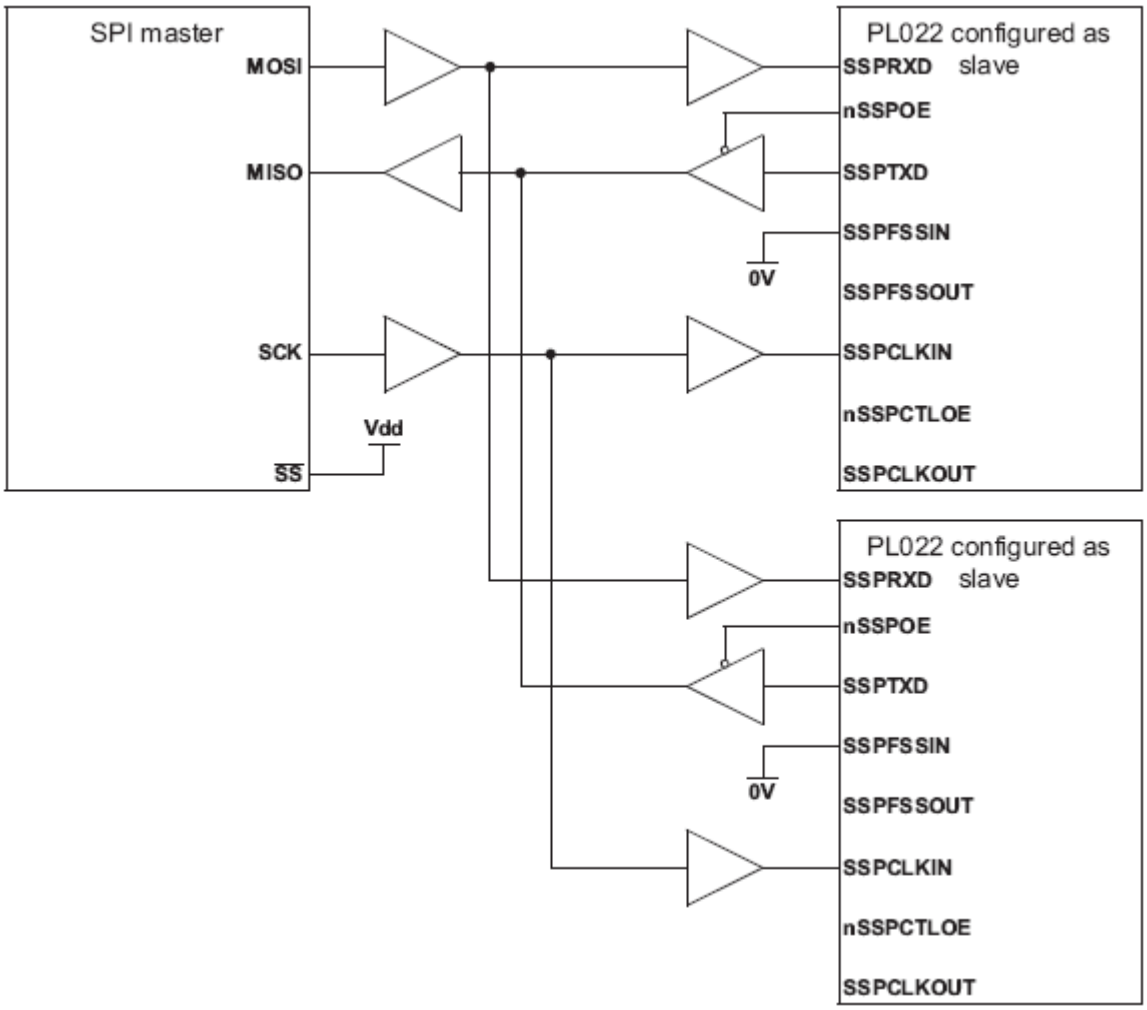


Рис. 2-15. Ведущее устройство, поддерживающее протокол SPI подключено к двум ведомым модулям SSP

Рис. 2-15 показывает ведущее устройство, поддерживающее протокол SPI фирмы Motorola, соединенное с двумя модулями SSP, сконфигурированными для работы в ведомом режиме. Линия Slave Select (SS) ведущего устройства в этом случае установлена в высокий логический уровень. Ведущее устройство осуществляет передачу данных по линии MOSI циркулярно в адрес двух ведомых модулей.

Для ответной передачи данных один из ведомых модулей переводит линию SSP_TXD в активное состояние, разрешая таким образом прохождение сигнала от своей линии SSP_TXD на вход SSP_RXD ведущего.

Интерфейс прямого доступа к памяти

Модуль SSP предоставляет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA SSPDMACR.

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

- SSPRXDMASREQ – запрос передачи отдельного символа, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ;
- SSPRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если буфер FIFO приемника содержит четыре или более символов;
- SSPRXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Для передачи:

- SSPTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит по меньшей мере одну свободную ячейку;
- SSPTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит четыре или менее символов;
- SSPTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимоисключающими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение четыре, формируются как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае если количество данных в буфере приема меньше порогового значения, формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

Примечание. Для оставшихся трех символов контроллер SSP не инициирует процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае снятия сигнала разрешения DMA.

В таблице 2-1 приведены значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 2-1. Параметры срабатывания запросов блочного обмена данными в режиме DMA.

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
½	4	4

На рис. 2-16 показаны временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

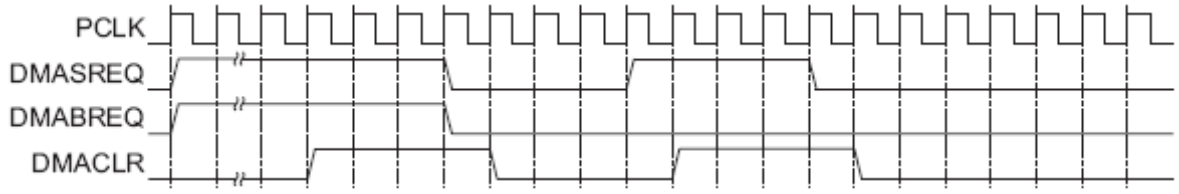


Рис. 2-16. Временные диаграммы обмена в режиме ПДП

Программное управление модулем

Общая информация

Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:
адреса со смещениями в диапазоне +0x028 ... +0x07C и +0xFD0 ... +0xFDC зарезервированы для перспективных расширений возможностей модуля;
адреса со смещениями в диапазоне +0x080 ... +0x088 зарезервированы для тестирования.

Описание регистров контроллера SSP

Данные о регистрах модуля SSP приведены в таблице 3-1.

Таблица 3-1. Обобщенные данные о регистрах модуля SSP.

Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x000	SSPx_CR0	RW	0x0000	16	Регистр управления 0 (табл. 3-2)
0x004	SSPx_CR1	RW	0x0	4	Регистр управления 1 (табл. 3-3)
0x008	SSPx_DR	RW	0x----	16	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись) (табл. 3-4)
0x00C	SSx_PSR	RO	0x03	3	Регистр состояния (табл. 3-5)
0x010	SSPx_CPSR	RW	0x00	8	Регистр делителя тактовой частоты (табл. 3-6)
0x014	SSPx_IMSC	RW	0x0	4	Регистр маски прерывания (табл. 3-7)
0x018	SSPx_RIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования (табл. 3-8)
0x01C	SSPx_MIS	RO	0x0	4	Регистр состояния прерываний с учетом маскирования (табл. 3-9)
0x020	SSPx_ICR	WO	0x0	4	Регистр сброса прерывания (табл. 3-10)
0x024	SSPx_DMACR	RW	0x0	2	Регистр управления прямым доступом к памяти (табл. 3-11)

Примечание. В поле «тип» указан вид доступа к регистру: RW – чтение и запись, RO – только чтение, WO – только запись.

SSPx_CR0

Регистр управления 0

Регистр SSPx_CR0 содержит пять битовых полей, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 3-2.

Таблица 3-2. Формат регистра SSPx_CR0.

Бит	Наименование	Назначение
15:8	SCR	Скорость последовательного обмена. Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению: F_SSPCLK / (CPSDVR * (1 + SCR)), где CPSDVR – четное число в диапазоне от 2 до 254 (см. регистр SSPCPSR), а SCR – число от 0 до 255.
7	SPH	Фаза сигнала SSP_CLK (используется только в режиме обмена SPI фирмы Motorola). См. раздел «Формат SPI фирмы Motorola» .
6	SPO	Полярность сигнала SSP_CLK (используется только в режиме обмена SPI фирмы Motorola). См. раздел «Формат SPI фирмы Motorola» .
5:4	FRF	Формат информационного кадра. 00 – протокол SPI фирмы Motorola; 01 – протокол SSI фирмы Texas Instruments; 10 – протокол Microwire фирмы National Semiconductor; 11 – резерв.
3:0	DSS	Размер слова данных. 0000 – резерв. 0001 – резерв. 0010 – резерв. 0011 – 4 бита. 0100 – 5 бит. 0101 – 6 бит. 0110 – 7 бит. 0111 – 8 бит. 1000 – 9 бит. 1001 – 10 бит. 1010 – 11 бит. 1011 – 12 бит. 1100 – 13 бит. 1110 – 14 бит. 1111 – 15 бит.

SSPx_CR1

Регистр управления 1

Регистр SSPx_CR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 3-3.

Таблица 3-3. Регистр SSPx_CR1.

Биты	Наименование	Назначение
15:4		Резерв, при чтении результат не определен. При записи следует устанавливать в 0.
3	SOD	Запрет выходных линий в режиме ведомого устройства. Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. Бит SOD следует установить в случае, если данный ведомый модуль SSP не должен в настоящее время осуществлять передачу данных в линию SSP_TXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSP_TXD в ведомом режиме разрешена. 1 – управление линией SSP_TXD в ведомом режиме запрещена.
2	MS	Выбор ведущего или ведомого режима работы: 0 – ведущий модуль (устанавливается по умолчанию); 1 – ведомый модуль.
1	SSE	Разрешение работы приемопередатчика: 0 – работа запрещена; 1 – работа разрешена.
0	LBM	Тестирование по шлейфу: 0 – нормальный режим работы приемопередатчика; 1 – выход регистра сдвига передатчика соединен с входом регистра сдвига приемника.

SSPx_DR

Регистр данных

Регистр SSPx_DR имеет разрядность 16 бит и предназначен для чтения принятых и записи передаваемых данных.

Операция чтения обеспечивает доступ к последней несчитанной ячейке буфера FIFO приемника. Запись данных в этот буфер FIFO осуществляет блок приемника.

Операция записи позволяет занести очередное слово в буфер FIFO передатчика. Извлечение данных из этого буфера осуществляет блок передатчика. При этом извлеченные данные помещаются в регистр сдвига передатчика, откуда последовательно выдаются на линию SSP_TXD с заданной скоростью информационного обмена.

В случае если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPx_DR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника.

В режиме обмена данными Microwire фирмы National Semiconductor модуль SSP по умолчанию работает с восьмиразрядными информационными словами (старший значащий байт игнорируется). Размер принимаемых данных задается программно. Буфера FIFO приемника и передатчика автоматически не очищаются даже в случае, если бит SSE установлен в 0. Это позволяет заполнить буфер передатчика необходимой информацией заблаговременно, перед разрешением работы модуля.

Назначение разрядов регистра SSPx_DR описано в таблице 3-4.

Таблица 3-4. Формат регистра SSPx_DR.

Бит	Наименование	Назначение
15:0	DATA	Принимаемые данные (чтение) Передаваемые данные (запись) В случае если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPx_DR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника.

SSPx_SR

Регистр состояния

Регистр состояния доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика и занятости модуля SSP.
В таблице 3-5 представлено назначение бит регистра SSPx_SR.

Таблица 3-5. Регистр SSPx_SR.

Биты	Наименование	Назначение
15:5		Резерв, при чтении результат не определен.
4	BSY	Флаг занятости модуля: 0 – модуль SSP неактивен; 1 – модуль SSP в настоящее время передает и/или принимает данные, либо буфер FIFO передатчика не пуст.
3	RFF	Буфер FIFO приемника заполнен: 0 – не заполнен; 1 – заполнен.
2	RNE	Буфер FIFO приемника не пуст: 0 – пуст; 1 – не пуст.
1	TNF	Буфер FIFO передатчика не заполнен: 0 – заполнен; 1 – не заполнен.
0	TFE	Буфер FIFO передатчика пуст: 0 – не пуст; 1 – пуст.

SSPx_CPSR

Регистр делителя тактовой частоты

Регистр SSPx_CPSR используется для установки параметров делителя тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль. Если записать в регистр

SSPx_CPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.
Назначение бит регистра SSPx_CPSR представлено в таблице 3-6.

Таблица 3-6. Регистр SSPx_CPSR.

Биты	Наименование	Назначение
15:8		Резерв, при чтении результат не определен. При записи следует заполнить нулями.
7:0	CPSDVSR	Коэффициент деления тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль.

SSPx_IMSC

Регистр установки и сброса маски прерывания

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание. При этом запись 1 в разряд разрешает соответствующее прерывание, запись 0 – запрещает.
После сброса все биты регистра маски устанавливаются в нулевое состояние.
Назначение битов регистра SSPx_IMSC показано в таблице 3-7.

Таблица 3-7. Регистр SSPx_IMSC.

Биты	Наименование	Назначение
15:4		Резерв. Не модифицируйте. При чтении выдаются нули.
3	TXIM	Маска прерывания по заполнению наполовину и менее буфера FIFO передатчика. 1 – не маскирована; 0 – маскирована.
2	RXIM	Маска прерывания по заполнению нанаполовину и менее буфера FIFO приемника. 1 – не маскирована; 0 – маскирована.
1	RTIM	Маска прерывания по таймауту приемника (буфер FIFO приемника не пуст и не было попыток его чтения в течение времени таймаута). 1 – не маскирована, 0 – маскирована.
0	RORIM	Маска прерывания по переполнению буфера приемника. 1 – не маскирована, 0 – маскирована.

SSPx_RIS

Регистр состояния прерываний,

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре SSPx_RIS представлено в таблице 3-8.

Таблица 3-8. Регистр SSPx_RIS.

Биты	Наименование	Назначение
15:4		Резерв. При чтении выдаются нули.
3	TXRIS	Состояние до маскирования прерывания SSPTXINTR. 1 – буфер FIFO передатчика заполнен наполовину или менее; 0 – буфер FIFO передатчика заполнен более чем наполовину.
2	RXRIS	Состояние до маскирования прерывания SSPRXINTR. 1 – буфер FIFO приемника заполнен нанаполовину или менее; 0 – буфер FIFO приемника заполнен более чем нанаполовину.
1	RTRIS	Состояние до маскирования прерывания SSPRTINTR. 1 – истекло время таймаута приемника; 0 – время таймаута приемника не истекло.
0	RORRIS	Состояние до маскирования прерывания SSPRORINTR. 1 – возникло событие переполнения буфера приемника; 0 – событие переполнения буфера приемника не возникало.

SSPx_MIS
Регистр маскированного состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.
Назначение бит в регистре SSPx_MIS представлено в таблице 3-9.

Таблица 3-9. Регистр SSPx_MIS.

Биты	Наименование	Назначение
15:4		Резерв. При чтении выдаются нули.
3	TXMIS	Состояние маскированного прерывания SSPTXINTR. 1 – буфер FIFO передатчика заполнен наполовину или менее; 0 – буфер FIFO передатчика заполнен более чем наполовину.
2	RXMIS	Состояние маскированного прерывания SSPRXINTR. 1 – буфер FIFO приемника заполнен нанаполовину или менее; 0 – буфер FIFO приемника заполнен более чем нанаполовину.
1	RTMIS	Состояние маскированного прерывания SSPRTINTR. 1 – истекло время таймаута приемника; 0 – время таймаута приемника не истекло.
0	RORMIS	Состояние маскированного прерывания SSPRORINTR. 1 – возникло событие переполнения буфера приемника; 0 – событие переполнения буфера приемника не возникало.

SSPx_ICR

Регистр сброса прерываний

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись в любой из разрядов регистра 0 игнорируется.
Назначение бит в регистре SSPx_ICR представлено в таблице 3-10.

Таблица 3-10. Регистр SSPx_ICR.

Биты	Наименование	Назначение
15:2		Резерв. Не модифицируйте. При чтении выдаются нули.
1	RTIC	Сброс прерывания SSPRTINTR.
0	RORIC	Сброс прерывания SSPRORINTR.

SSPx_DMACR

Регистр управления прямым доступом к памяти,

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.
Назначение бит регистра SSPx_DMACR представлено в таблице 3-11.

Таблица 3-11. Регистр SSPx_DMACR.

Биты	Наименование	Назначение
15:2		Резерв. Не модифицируйте. При чтении выдаются нули.
1	TXDMAE	Использование DMA при передаче. 1 – разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика; 0 – запрещено формирование запросов DMA для обслуживания буфера FIFO передатчика.
0	RXDMAE	Использование DMA при приеме. 1 – разрешено формирование запросов DMA для обслуживания буфера FIFO приемника; 0 – запрещено формирование запросов DMA для обслуживания буфера FIFO приемника.

Прерывания

В модуле предусмотрено пять маскируемых линий запроса на прерывание, в том числе, четыре независимые линии запроса с активным высоким логическим уровнем, а также один общий сигнал, представляющий собой комбинацию независимых по схеме ИЛИ.
Сигналы запроса на прерывание:
SSPRXINTR – запрос на обслуживание буфера FIFO приемника.
SSPTXINTR – запрос на обслуживание буфера FIFO передатчика.
SSPRORINTR – переполнение буфера FIFO приемника.
SSPRTINTR – таймаут приемника.
SSPINTR – логическое ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSPx_IMSC. Установка бита в 1 разрешает соответствующее прерывание, в 0 – запрещает.
Доступность как индивидуальных, так и общей линии запроса позволяет организовать обслуживание прерываний в системе как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.
Прерывания от приемника и передатчика SSPRXINTR и SSPTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать данные сигналы запроса для обеспечения чтения и записи данных, согласованной с достижением заданного порога заполнения буферов FIFO приемника и передатчика.
Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний SSPx_RIS, либо из маскированного регистра прерываний SSPx_MIS.

SSPRXINTR

Прерывание по заполнению буфера FIFO приемника формируется в случае, если буфер приемника содержит четыре или более несчитанных слов данных.

SSPTXINTR

Прерывание по заполнению буфера FIFO передатчика формируется в случае, если буфер передатчика содержит четыре или менее корректных слов данных.
Состояние прерывания не зависит от значения сигнала разрешения работы модуля SSP. Это позволяет организовать взаимодействие программного обеспечения с передатчиком одним из двух способов. Во-первых, можно записать данные в буфер заблаговременно, перед активизацией передатчика и разрешения прерываний. Во-вторых, можно предварительно разрешить работу модуля и формирование прерываний и заполнять буфер передатчика в ходе работы процедуры обслуживания прерываний.

SSPRORINTR

Прерывание по переполнению буфера FIFO приемника формируется в случае, если буфер уже заполнен и блоком приемника осуществлена попытка записать в него еще одно слово. При этом принятое слово данных регистрируется в регистре сдвига приемника, но в буфер приемника не заносится.

SSPRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Данный механизм гарантирует, что пользователь будет знать о наличии в буфере приемника необработанных данных.
Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения, либо после приема новых слов данных по входной линии SSPRXD. Кроме того оно может быть снято путем записи 1 в бит RTIC регистра сброса прерывания SSPTICR.

SSPINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

Контроллер UART

Модуль универсального асинхронного приемопередатчика (UART – Universal Synchronous Asynchronous Receiver Transmitter) представляет собой периферийное устройство микроконтроллера.

В состав контроллера включен кодек (ENDEC – ENcoder/DEcoder) последовательного интерфейса инфракрасной (ИК) передачи данных в соответствии с протоколом SIR (SIR – Serial Infra Red) ассоциации Infrared Data Association (IrDA).

Основные характеристики модуля UART

Модуль UART может быть запрограммирован для использования как в качестве универсального асинхронного приемопередатчика, так и для инфракрасного обмена данными (SIR).

Модуль содержит независимые буферы приема (16x12) и передачи (16x8) типа FIFO (First In First Out – первый вошел, первый вышел), что позволяет снизить интенсивность прерываний центрального процессора.

Программное отключение FIFO позволяет ограничить размер буфера одним словом.

Есть возможность программн настраивать скоростью обмена данными, путем деления тактовой частоты опорного генератора в диапазоне (1x16 – 65535x16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3.6864 МГц.

Модулем поддерживаются стандартные элементы асинхронного протокола связи – стартового и стопового бит, а также бита контроля четности, которые добавляются перед передачей и удаляются после приема.

Независимо могут быть маскированы прерывания от буфера FIFO передатчика, буфера FIFO приемника, по таймауту приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки.

Модуль UART позволяет использовать DMA-контроллер для организации обмена данными и снижения нагрузки на ядро микроконтроллера.

Также модуль обеспечивает обнаружение ложных стартовых бит.

Формирование и обнаружения сигнала разрыва линии.

Функция управления модемом (линии CTS, DCD, DSR, RTS, DTR и RI) будет поддерживаться с ревизии 3.

Возможность организации аппаратного управления потоком данных.

Полностью программируемый асинхронный последовательный интерфейс имеет следующие характеристики:

- данные длиной 5, 6, 7 или 8 бит;
- формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение, либо не передается);
- формирование 1 или 2 стоповых бит;
- скорость передачи данных – от 0 до UART_CLOCK/16 Бод.

Кодек ИК-обмена данными IrDA SIR обеспечивает:

- программный выбор обмена данными по линиям асинхронного приемопередатчика либо кодека ИК связи IrDA SIR;

- поддержку функционирования с информационной скоростью до 115200 бит/с в режиме полудуплекса;
- поддержку длительности бит для нормального режима (3/16) и для режима пониженного энергопотребления (1.41 – 2.23 мкс);
- программируемое деление опорной частоты UARTCLK для получения заданной длительности бит в режиме пониженного энергопотребления.

Наличие идентификационного регистра, однозначно идентифицирующего модуль, что позволяет операционной системе выполнять автоматическую конфигурацию.

Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

- скорость передачи данных – целая и дробная часть числа;
- количество бит данных;
- количество стоповых бит;
- режим контроля четности;
- разрешение или запрет использования буферов FIFO (глубина очереди данных – 16 элементов или один элемент, соответственно);
- порог срабатывания прерывания по заполнению буферов FIFO (1/8, 1/4, 1/2, 3/4 и 7/8);
- частота внутреннего тактового генератора (номинальное значение – 1.8432 МГц) может быть задана в диапазоне 1.42 – 2.12 МГц для обеспечения возможности формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления;
- режим аппаратного управления потоком данных.

Отличия от контроллера UART 16C650

Контроллер отличается от промышленного стандарта асинхронного приемопередатчика 16C650 следующими характеристиками:

- пороги срабатывания прерывания по заполнению буфера FIFO приемника – 1/8, 1/4, 1/2, 3/4 и 7/8;
- пороги срабатывания прерывания по заполнению буфера FIFO передатчика – 1/8, 1/4, 1/2, 3/4 и 7/8;
- отличается распределение адресов внутренних регистров и назначение бит в регистрах;
- недоступны изменения сигналов состояния модема.

Следующие возможности контроллера 16C650 не поддерживаются:

- полуторная длительность стопового бита (поддерживается только 1 или 2 стоповых бита);
- независимое задание тактовой частоты приемника и передатчика.

Функциональные возможности

Устройство выполняет следующие функции:

- преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;
- преобразование данных, передаваемых на периферийное устройство, из параллельной в последовательную форму.

Процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии модуля. Прием и передача данных буферизуются с помощью внутренней памяти FIFO, позволяющей сохранить до 16 байтов независимо для режимов приема и передачи.

Модуль приемопередатчика:

- содержит программируемый генератор, формирующий тактовый сигнал одновременно для передачи и для приема данных на основе внутреннего тактового сигнала UART_CLOCK;
- обеспечивает возможности, сходные с возможностями индустриального стандарта - контроллера UART 16C650;
- позволяет осуществлять обмен информацией с максимальной скоростью:
  - в режиме UART – до 921600 бит/с;
  - в режиме IrDA – до 460800 бит/с;
  - в режиме IrDA с пониженным энергопотреблением – до 115200 бит/с.

Режим работы приемопередатчика и скорость обмена данными контролируются регистром управления линией UART_LCR_H и регистрами делителя скорости передачи данных – целой части (UART_IBRD) и дробной части (UART_FBRD).

Устройство может формировать следующие сигналы:

- независимые маскируемые прерывания от приемника (в том числе по таймауту), передатчика, а также по изменению состояния модема и в случае обнаружения ошибки;
- общее прерывание, возникающее в случае, если возникло одно из независимых немаскированных прерываний;
- сигналы запроса на прямой доступ к памяти (DMA) для совместной работы с контроллером DMA.

В случае возникновения ошибки в структуре сигнала, четности данных, а также разрыва линии соответствующий бит ошибки устанавливается и сохраняется в буфере FIFO. В случае переполнения буфера немедленно устанавливается соответствующий бит в регистре переполнения, а доступ к записи в буфер FIFO блокируется.

Существует возможность программно ограничить размер буфера FIFO одним байтом, что позволяет реализовать общепринятый интерфейс асинхронной последовательной связи с двойной буферизацией.

Поддерживаются входные линии состояния модема: «готовность к приему» (Clear To Send, CTS), «обнаружен информационный сигнал» (Data Carrier Detected, DCD), «источник данных готов» (Data Set Ready, DSR) и «индикатор вызова» (Ring Indicator, RI), а также выходные линии: «запрос на передачу» (Request to Send, RTS) и «приемник данных готов» (Data Terminal Ready, DTR).

Доступна возможность аппаратного управления потоком данных по линиям nUARTCTS и nUARTRTS.

Блок последовательного интерфейса инфракрасной передачи данных в соответствии с протоколом IrDA SIR реализует протокол обмена данными ENDEC. В случае его активизации обмен информацией осуществляется не с помощью сигналов UART_TXDx и UART_RXDx, а посредством сигналов SIR_OUTx и SIR_INx.

В этом случае устройство переводит линию UART_TXDx в пассивное состояние (высокий уровень), и перестает реагировать на изменение состояния модема, а также сигнала на линии UART_RXDx. Протокол SIR ENDEC обеспечивает возможность обмена данными исключительно в режиме полудуплекса, то есть он не может передавать во время приема данных и принимать во время передачи данных.

В соответствии со спецификацией физического уровня протокола IrDA SIR, задержка между передачей и приемом должна составлять не менее 10 мс.

Описание функционирования блока UART

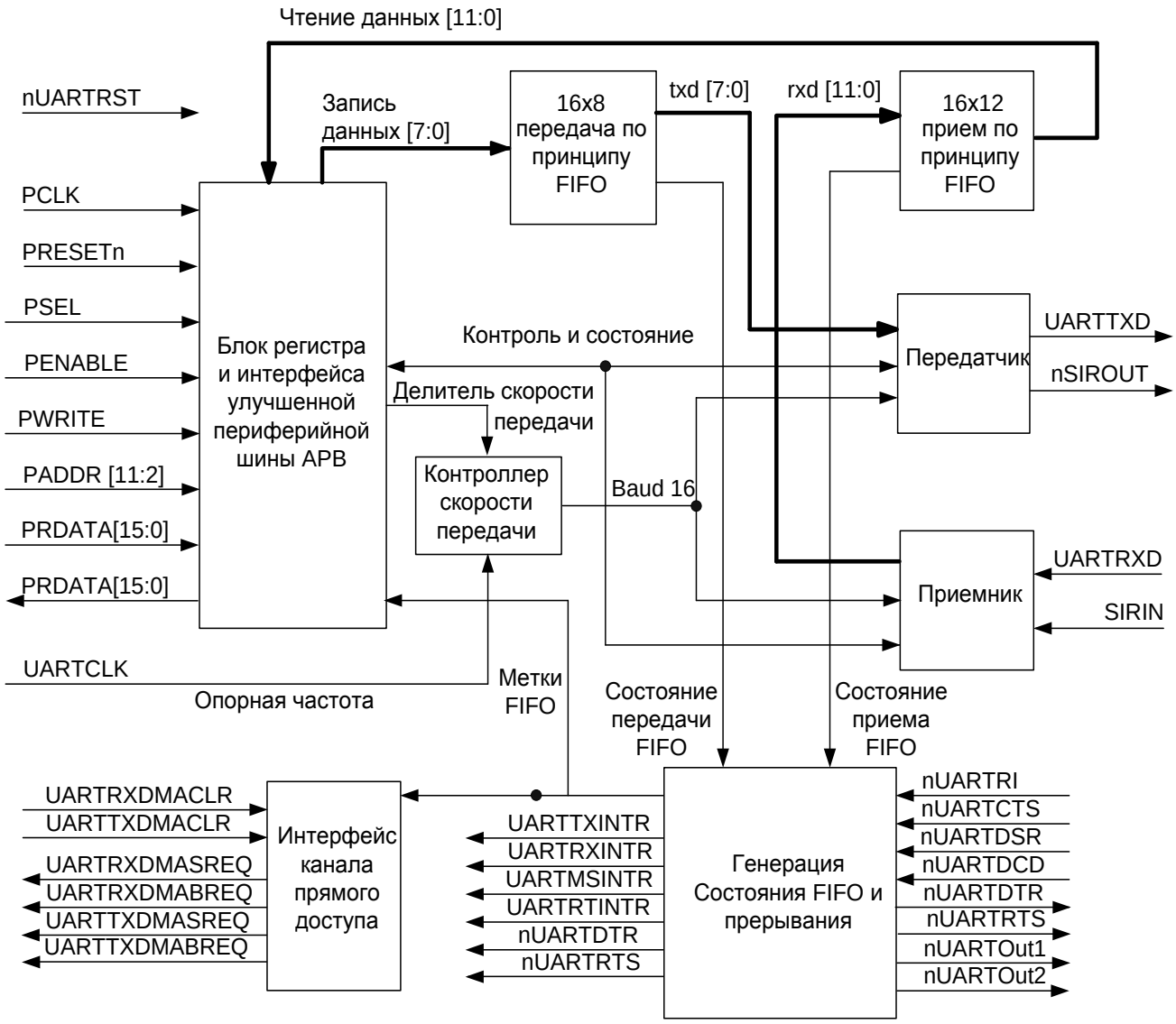


Рисунок 2-1. Блок-схема универсального асинхронного приёмопередатчика (UART)

Генератор тактового сигнала приемопередатчика

Генератор содержит счетчики без цепи сброса, формирующие внутренние тактовые сигналы Baud16 и IrLPBaud16.

Сигнал Baud16 используется для синхронизации схем управления приемником и передатчиком последовательного обмена данными. Он представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UART_CLOCK и частотой, в 16 раз выше скорости передачи данных.

Сигнал IrLPBaud16 предназначен для синхронизации схемы формирования импульсов с длительностью, требуемой для ИК обмена данными в режиме с пониженным энергопотреблением.

Буфер FIFO передатчика

Буфер передатчика имеет ширину 8 бит, глубину 16 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Данные от центрального процессора, записанные через шину APB, сохраняются в буфере до тех пор, пока не будут считаны логической схемой передачи данных. Существует возможность запретить буфер FIFO передатчика, в этом случае он будет функционировать как однобайтовый буферный регистр.

Буфер FIFO приемника

Буфер приемника имеет ширину 12 бит, глубину 16 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Принятые от периферийного устройства данные и соответствующие кодов ошибки сохраняются логикой приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину APB. Буфер FIFO приемника может быть запрещен, в этом случае он будет действовать как однобайтовый буферный регистр.

Блок передатчика

Логические схемы передатчика осуществляют преобразование данных, считанных из буфера передатчика, из параллельной в последовательную форму. Управляющая логика выдает последовательный поток бит в порядке: стартовый бит, биты данных, начиная с младшего значащего разряда, бит проверки на четность, и, наконец, стоповые биты, в соответствии с конфигурацией, записанной в регистре управления.

Блок приемника

Логические схемы приемника преобразуют данные, полученные от периферийного устройства, из последовательной в параллельную форму после обнаружения корректного стартового импульса. Кроме того производятся проверки переполнения буфера, проверки на ошибки контроля четности, на ошибки в структуре сигнала, а также на разрыв линии. Признаки обнаружения этих ошибок также сохраняются в выходном буфере.

Блок формирования прерываний

Контроллер генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения указанных независимых прерываний по схеме ИЛИ.

Комбинированный сигнал прерывания может быть подан на внешний контроллер прерываний системы, при этом появится дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

Другой подход состоит в подаче на системный контроллер прерываний независимых линий запроса на прерывание от приемопередатчика. В этом случае процедура обработки сможет одновременно считать информацию обо всех источниках прерывания. Данный подход привлекателен в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

Для более подробной информации см. раздел Прерывания.

Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов CPU_CLK и UART_CLOCK. Регистры синхронизации и логика квитирования постоянно находятся в активном состоянии. Это практически не отражается на характеристиках устройства и занимаемой площади. Синхронизация сигналов управления осуществляется в обоих направлениях потока данных, то есть как из области действия CPU_CLK в область действия UART_CLOCK, так и наоборот.

Описание функционирования ИК кодека IrDA SIR

Структурная схема кодека представлена на рис. 2-2.

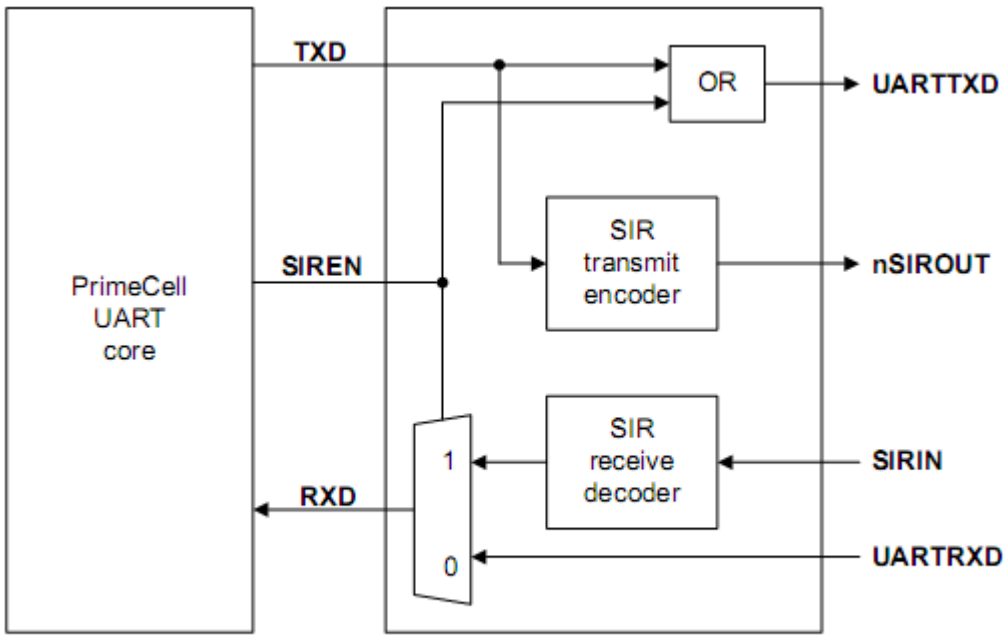


Figure 2-2 IrDA SIR ENDEC block diagram

Кодер ИК передатчика

Кодер преобразует поток данных с выхода асинхронного передатчика, сформированный по закону модуляции без возврата к нулю (NRZ). Спецификация физического уровня протокола IrDA SIR подразумевает использование модуляции с возвратом к нулю и инверсией (RZI), в соответствии с которой передача логического нуля соответствует излучению одного светового ИК импульса. Сформированный выходной поток импульсов подается на усилитель и, далее, на ИК светодиод.

Длительность импульса в режиме IrDA составляет, согласно спецификации, 3 периода внутреннего тактового генератора с частотой Baud16, то есть 3/16 периода времени, выделенного на передачу одного бита.

В режиме IrDA с пониженным энергопотреблением ширина импульса задана как 3/16 периода, выделенного на передачу бита, при скорости передачи данных 115200 бит/с. Данное требование реализуется за счет формирования трех периодов тактового сигнала IrLPBaud16 с номинальной частотой 1.8432 МГц, в свою очередь, формируемого путем деления частоты UART_CLOCK. Значение частоты IrLPBaud16 задается путем записи соответствующего коэффициента деления частоты в регистр UART_ILPR.

Выход кодера имеет активное низкое состояние. При передаче логической единицы выход кодера остается в низком состоянии, при передаче логического нуля – формируется импульс, при этом выход кратковременно переводится в высокое состояние.

Как в нормальном режиме, так и в режиме пониженного энергопотребления использование нецелых значений коэффициента деления скорости передачи данных увеличивает джиттер («дребезжание») фронтов импульсов данных. Наличие джиттера в случае использования дробных коэффициентов деления связано с тем, что интервалы между тактовыми импульсами Baud16 будут нерегулярными – период сигнала Baud16 в разное время будет содержать различное количество периодов сигнала UART_CLOCK. Можно показать, что в наихудшем случае величина джиттера в потоке ИК импульсов может достигать трех периодов UART_CLOCK. В соответствии со спецификацией стандарта IrDA SIR, джиттер не должен превышать величины 13%. В случае, если частота сигнала UART_CLOCK составляет более 3.6834 МГц, а скорость передачи данных меньше или равна 115200 бит/с, величина джиттера не превышает 9%. Таким образом, требования стандарта выполняются.

Декодер ИК приемника

Декодер преобразует поток данных, сформированных по закону возврата к нулю, полученного от приемника ИК сигнала, и выдает поток данных без возврата к нулю на вход приемника UART. В неактивном состоянии вход декодера находится нормально в высоком состоянии. Выходной сигнал кодера имеет полярность, противоположную полярности входа декодера.

Обнаружение стартового бита осуществляется при низком уровне сигнала на входе декодера.

Примечание. Для того чтобы исключить ложные срабатывания UART от импульсных помех, на входе SIR_INx игнорируются импульсы с длительностью менее, чем:

- 3/16 длительности Baud16 в режиме IrDA;
- 3/16 длительности IrLPBaud16 в режиме IrDA с пониженным энергопотреблением.

Описание работы UART

Сброс модуля

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса процессора. Значения регистров после сброса описаны в разделе «Программное управление модулем».

Тактовые сигналы

Частота тактового сигнала UART_CLOCK должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

```
F_UART_CLOCK(min) >= 16 * baud_rate_max;
F_UART_CLOCK(max) <= 16 * 65535 * baud_rate_min.
```

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота UART_CLOCK должна находиться в интервале от 7.3728 МГц до 115.34 МГц.

Частота UART_CLOCK, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами CPU_CLK и UART_CLOCK. Частота UART_CLOCK должна быть не более чем в 5/3 раз выше частоты CPU_CLOCK.

```
F_UART_CLOCK <= 5/3 * F_ CPU_CLK.
```

Например, при работе в режиме UART с максимальной скоростью передачи данных 921600 бод, при частоте UART_CLOCK 14.7456 МГц, частота CPU_CLOCK должна быть не менее 8.85276 МГц. Это гарантирует, что контроллер UART будет иметь достаточно времени для записи принятых данных в буфер FIFO.

Работа универсального асинхронного приемопередатчика

Управляющая информация хранится в регистре управления линией UART_LCR. Этот регистр имеет внутреннюю ширину 30 бит, однако внешний доступ по шине APB к нему осуществляется через следующие регистры:

– UART_LCR_H – определяет:
  - параметры передачи данных;
  - длину слова;
  - режим буферизации;
  - количество передаваемых стоповых бит;
  - режим контроля четности;
  - формирование сигнала разрыва линии;
– UART_IBRD – определяет целую часть коэффициента деления для скорости передачи данных;
– UART_FBRD – определяет дробную часть коэффициента деления для скорости передачи данных.

Коэффициент деления частоты

Коэффициент деления для формирования скорости передачи данных состоит из 22 бит, при этом 16 бит выделено для представления его целой части, а 6 бит – дробной части. Возможность задания нецелых коэффициентов деления позволяет осуществлять обмен данными со стандартными информационными скоростями, при этом используя в качестве UART_CLOCK тактовый сигнал с произвольной частотой более 3.6864 МГц.

Целая часть коэффициента деления записывается в 16-битный регистр UART_IBRD. Шестиразрядная дробная часть записывается в регистр UART_FBRD. Значение коэффициента деления связано с содержимым указанных регистров следующим образом:

```
Коэффициент деления = UART_CLOCK / (16 * скорость передачи
данных) = UART_IBRD + UART_FBRD,
```

где

UART_IBRD – целая часть, а UART_FBRD – дробная часть коэффициента деления.

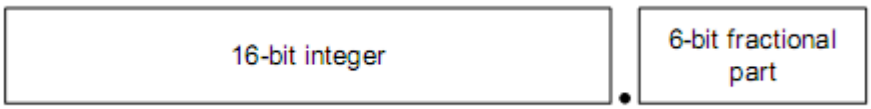


Рисунок 1. Коэффициент деления

Шестибитное значение, записываемое в регистр UART_FBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на 2^n, где n – ширина регистра UART_FBRD) и округления до ближайшего целого числа:

$$M = integer(UART\_FBRD * 2^n + 0.5),$$

где

integer – операция отсечения дробной части числа, n = 6.

В модуле формируется внутренний сигнал Baud16, представляющий собой последовательность импульсов с длительностью, равной периоду сигнала UART_CLOCK и средней частотой, в 16 раз большей требуемой скорости обмена данными.

Передача и прием данных

Принятые или передаваемые данные заносятся в 16-элементные буферы FIFO, при этом каждый элемент приемного буфера FIFO кроме байта данных хранит также четыре бита информации о состоянии модема.

Данные для передачи заносятся в буфер FIFO передатчика. Если работа приемопередатчика разрешена, начинается передача информационного кадра с параметрами, указанными в регистре управления линией UART_LCR_H. Передача данных продолжается до опустошения буфера FIFO передатчика. После записи элемента в буфер FIFO передатчика сигнал BUSY переходит в высокое состояние. Это состояние сохраняется в течение всего времени передачи данных. В низкое состояние сигнал BUSY переходит только после того, как буфер FIFO передатчика станет пуст, а последний бит данных (включая стоповые биты) будет передан. Сигнал BUSY может находиться в высоком состоянии даже в случае, если приемопередатчик будет переведен из разрешенного состояния в запрещенное.

Для каждого бита данных (в приемной линии) производится три измерения уровня, решение принимается по мажоритарному принципу.

В случае если приемник находился в неактивном состоянии (на линии входного сигнала UART_RXDx постоянно присутствовала единица) и произошел переход входного сигнала из высокого в низкий логический уровень (обнаружен стартовый бит), включается счетчик, тактируемый сигналом Baud16, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов (в режиме асинхронного приемопередатчика) или каждые четыре такта (в режиме ИК обмена данными) сигнала Baud16. Более частая выборка данных в режиме ИК обмена связана с необходимостью корректной обработки импульсов данных согласно протоколу SIR IrDA.

Стартовый бит считается достоверным в случае, если сигнал на линии UART_RXDx сохраняет низкий логический уровень в течение восьми отсчетов сигнала Baud16 с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

В случае если обнаружен достоверный стартовый бит, производится регистрация последовательности данных на входе приемника. Очередной бит данных фиксируются каждые 16 отсчетов тактового сигнала Baud16 (что соответствует длительности одного символа). Производится регистрация всех бит данных (согласно запрограммированным параметрам) и бита четности (если включен режим контроля четности).
Наконец, производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UART_RXDx). В случае если последнее условие не выполняется, устанавливается признак ошибки формирования кадра. После того, как слово данных принято полностью, оно заносится в буфер FIFO приемника, наряду с четырьмя битами признаков ошибки, связанных с принятым словом (см. таблицу 2-1).

Биты ошибки

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся на позиции [10:8] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO, расположенный на позиции 11 слова данных.
В таблице 2-1 описано назначение всех битов слова данных в FIFO буфере приемника.

Бит переполнения буфера

Бит переполнения непосредственно не связан с конкретным символом в буфере приемника. Признак переполнения фиксируется в случае, если буфер FIFO заполнен к моменту, когда очередной символ данных полностью принят (находится в регистре сдвига). При этом данные из регистра сдвига не попадают в буфер приемника и теряются с началом приема очередного символа. Как только в буфере приемника появляется свободное место, очередной принятый символ данных заносится в буфер FIFO вместе с текущим значением признака переполнения. После успешной записи данных в буфер признак переполнения сбрасывается.

Таблица 2-1.

Бит буфера FIFO	Назначение
11	Признак переполнения буфера
10	Ошибка – разрыв линии

09	Ошибка проверки на четность
08	Ошибка формирования кадра
07:00	Принятые данные

Запрет буфера FIFO

Предусмотрена возможность отключения FIFO буферов приемника и передатчика. В этом случае приемная и передающая сторона контроллера UART располагают лишь однобайтными буферными регистрами. Бит переполнения буфера устанавливается при этом тогда, когда очередной символ данных уже принят, однако предыдущий еще не был считан.

В настоящей реализации модуля буферы FIFO физически не отключаются, необходимая функциональность достигается за счет логических манипуляций с флагами. При этом в случае, если буфер FIFO отключен, а сдвиговый регистр передатчика пуст (не используется), запись байта данных происходит непосредственно в регистр сдвига, минуя буферный регистр.

Проверка по шлейфу

Проверка по шлейфу (замыкание выхода передатчика на вход приемника) выполняется путем установки в 1 бита LBE в регистре управления контроллером UART_CR.

Работа кодека ИК обмена данными IrDA SIR

Кодек обеспечивает сопряжение асинхронного потока данных, сформированного приемопередатчиком, с полудуплексным последовательным интерфейсом IrDA SIR. Какая-либо аналоговая обработка сигнала при этом не выполняется. Назначение кодека – сформировать цифровой поток данных на вход приемника асинхронного сигнала и обработать цифровой поток данных с выхода передатчика.

Предусмотрено два режима работы:

В режиме IrDA уровень логического нуля передается на линию SIR_OUT в виде импульса с высоким логическим уровнем и длительностью 3/16 от выбранного периода следования бит данных. Логическая единица при этом передается в виде постоянного низкого уровня сигнала. Сформированный выходной сигнал далее подается на передатчик ИК сигнала, обеспечивая излучение светового импульса всякий раз при передаче нулевого бита. На приемной стороне световые импульсы воздействуют на базу фототранзистора ИК приемника, который в результате формирует низкий логический уровень. Это, в свою очередь, обуславливает низкий уровень на входе SIR_IN.

В режиме IrDA с пониженным энергопотреблением длительность передаваемых импульсов ИК излучения устанавливается в три раза выше длительности импульсов внутреннего опорного сигнала IrLPBaud16 (равной 1.63 мкс при номинальной частоте 1.8432 МГц). Данный режим активизируется путем установки бита SIR_LP в регистре управления UART_CR.

Как в нормальном режиме, так и в режиме пониженного энергопотребления:

- кодирование осуществляется на основе бит данных, сформированных асинхронным передатчиком модуля;

– в ходе приема данных декодированные биты далее обрабатываются блоком асинхронного приема.

В соответствии со спецификацией физического уровня протокола IrDA SIR, обмен данными должен осуществляться в режиме полудуплекса, при этом задержка между передачей и приемом данных должна составлять не менее 10 мс. Эта задержка должна формироваться программно. Необходимость ее введения обусловлена тем, что воздействие передающего ИК светодиода на находящийся рядом ИК приемник может привести к искажению принимаемого сигнала или даже ввести приемный тракт в состояние насыщения. Задержка между окончанием передачи и началом приема данных именуется латентность, или время установки (готовности) приемника.

Сигнал IrLPBaud16 формируется путем деления частоты сигнала UART_CLOCK в соответствии с коэффициентом деления, записанным в регистре UART_ILPR.

Коэффициент деления вычисляется по формуле:

```
F_UART_CLOCK / F_IrLPBaud16,
```

где номинальное значение `IrLPBaud16` составляет 1.8432 МГц. Коэффициент деления должен быть выбран так, чтобы выполнялось соотношение:

```
1.42 МГц < F_IrLPBaud16 < 2.12 МГц.
```

Проверка по шлейфу

Проверка по шлейфу выполняется после установки в 1 бита LBE регистра управления контроллером UART_CR с одновременной установкой в 1 бита SIRTEST регистра управления тестированием UARTTCR.

В этом режиме данные, передаваемые на выход nSIR_OUT, должны подаваться на вход SIR_IN.

Примечание. Это единственный случай использования тестового регистра в нормальном режиме функционирования модуля.

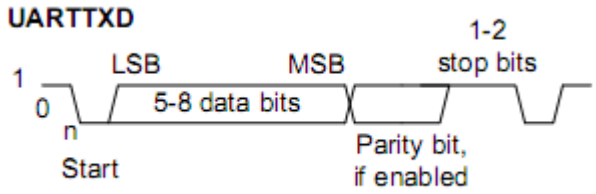


Рис. 2-4 Кадр передачи данных

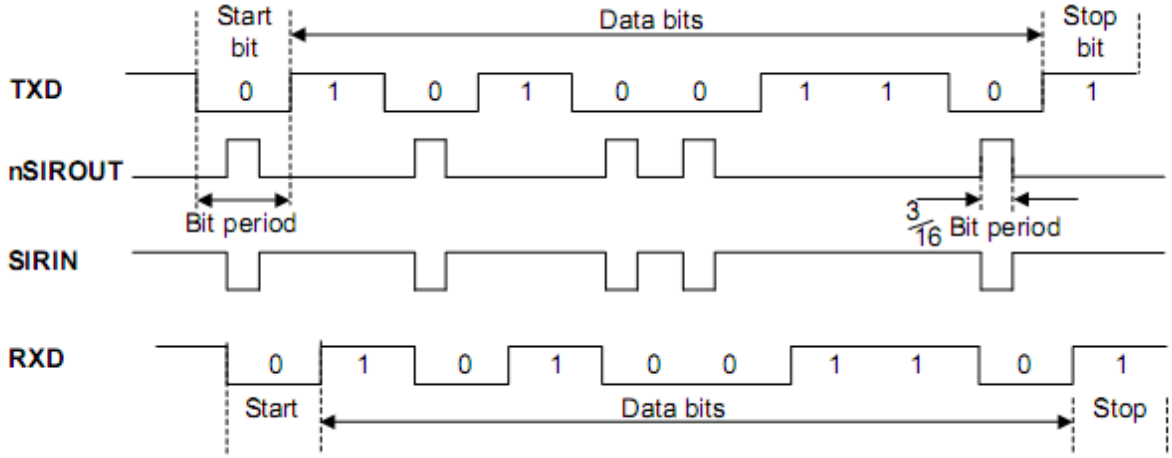


Рис. 2-5 Модуляция данных IrDA

Линии управления модемом

Модуль универсального асинхронного приемопередатчика может использоваться как в режиме оконечного оборудования (DTE), так и в режиме оборудования передачи данных (DCE). На рис. 2-1 стр. 2-4 показаны сигналы модема в режиме DTE. Назначение сигналов в режиме DCE представлены в таблице 2-2.

Таблица 2-2. Назначение управления модемом в режимах DTE и DCE

Сигнал	Назначение	
	Режим оконечного оборудования	Режим оборудования передачи данных
nUART_CTS	Готов к передаче данных	Запрос передачи данных
nUART_DSR	Источник данных готов	Приемник данных готов
nUART_DCD	Обнаружен информационный сигнал	-
nUART_RI	Индикатор вызова	-
nUART_CTS	Запрос передачи данных	Готов к передаче данных
nUART_DTR	Приемник данных готов	Источник данных готов
nUART_OUT1	-	Обнаружен информационный сигнал
nUART_OUT2	-	Индикатор вызова

Аппаратное управление потоком данных

Программно активизируемый режим аппаратного управления потоком данных позволяет контролировать (приостанавливать и возобновлять) информационный обмен с помощью сигналов nUART_RTS и nUART_CTS. Иллюстрация взаимодействия двух устройств последовательной связи с аппаратным управлением потоком данных представлена на рис. 2-6.

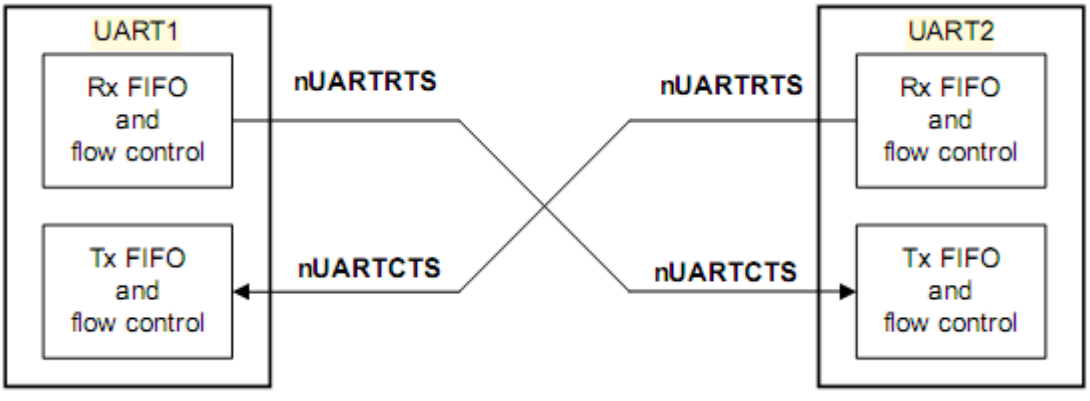


Рис. 2-6. Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

Если разрешено управление потоком данных по сигналу RTS, линия nUART_RTS переводится в активное состоянии только после того, как в FIFO буфере приема появляется заданное количество свободных элементов.

Если разрешено управление потоком данных по сигналу CTS, передача данных осуществляется только после перевода линии nUART_CTS в активное состояние.
Режим аппаратного управления потоком данных задается путем установки значений бит RTSEn и CTSEn в регистре управления UART_CR (стр. 3-15). В таблице 2-3 показаны необходимые установки для различных режимов управления потоком данных.

Таблица 2-3. Режимы управления потоком данных

CTSEn	RTSEn	Описание
1	1	Разрешено управление потоком данных по CTS и RTS
1	0	Управления потоком данных осуществляется по линии CTS
0	1	Управления потоком данных осуществляется по линии RTS
0	0	Управления потоком данных запрещено

Примечание. В случае если выбран режим управления потоком данных по RTS, программное обеспечение не может использовать бит RTSEn регистра UART_CR (стр. 3-15) для проверки состоянии линии RTS.

Управление потоком данных по линии RTS

Логика управления потоком данных по RTS использует данные о превышении пороговых уровней заполнения буфера FIFO приемника. В случае выбора режимов с управлением по RTS, сигнал на линии nUART_RTS переводится в активное состояние только после того, как в FIFO буфере приема появляется заданное количество свободных элементов. После достижения порогового уровня заполнения буфера приемника сигнал nUART_RTS снимается (переводится в пассивное состояние), указывая таким образом на отсутствие свободного места для сохранения принятых данных. При этом дальнейшая передача данных должна быть прекращена по завершении передачи текущего символа.

Обратно в активное состояние сигнал nUART_RTS переводится после считывания данных из приемного буфера FIFO в количестве, достаточном для того, чтобы заполнение буфера оказалось ниже порогового уровня.

В случае если управление потоком данных по RTS запрещено, однако работа приемопередатчика UART разрешена, прием будет осуществляться до полного заполнения буфера FIFO, либо до завершения передачи данных.

Управление потоком данных по линии CTS

В случае выбора одного из режимов с управлением потоком данных по CTS передатчик осуществляет проверку состояния линии nUART_CTS перед началом передачи очередного байта данных. Передача осуществляется только в случае, если данная линия активна, и продолжается до тех пор, пока активное состояние линии сохраняется и буфер передатчика не пуст.

При переходе линии nUART_CTS в неактивное состояние модуль завершает выдачу текущего передаваемого символа, после чего передача данных прекращается.

Если управление потоком данных по CTS запрещено, и при этом работа приемопередатчика UART разрешена – данные будут выдаваться до опустошения буфера FIFO передатчика.

Интерфейс прямого доступа к памяти

Модуль универсального асинхронного приемопередатчика оснащен интерфейсом подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA UART_DMACR.

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

UART_RX_DMA_SREQ – запрос передачи отдельного символа, инициируется контроллером UART. Размер символа в режиме приема данных – до 12 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ.

UART_RX_DMA_BREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если заполнение буфера FIFO приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр UART_IFLS.

UART_RX_DMA_CLR – сброс запроса на DMA, инициируется модулем приемопередатчика с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Для передачи:

UART_TX_DMA_SREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Размер символа в режиме передачи данных – до восьми бит. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку.

UART_TX_DMA_BREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если заполнение буфера FIFO передатчика ниже заданного порога. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр UART_IFLS.

UART_TX_DMA_CLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимно исключающими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение, формируется как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае если количество данных в буфере приема меньше порогового значения формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов, а порог заполнения буфера FIFO установлен равным четырем. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

Примечание. Для оставшихся трех символов контроллер UART не может инициировать процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMA_CLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае установки в ноль бита управления DMA TXDMAE или RXDMAE в регистре управления DMA UART_DMACR.

В случае запрета буферов FIFO устройство способно передавать и принимать только одиночные символы, как следствие, контроллер может инициировать DMA только в одно-элементном режиме. При этом модуль в состоянии формировать только сигналы управления DMA UART_RX_DMA_SREQ и UART_TX_DMA_SREQ. Для информации о запрете буферов FIFO см. описание регистра управления линией UART_LCR_H.

Когда буферы FIFO включены, обмен данными может производиться в ходе как одноэлементных, так и блочных передач данных, в зависимости от установленной величины порога заполнения буферов и их фактического заполнения. Таблица 2-4 показывает значения параметров срабатывания запросов блочного обмена UART_RX_DMA_BREQ и UART_TX_DMA_BREQ в зависимости от порога заполнения буфера.

Таблица 2-4. Параметры срабатывания запросов блочного обмена данными в режиме DMA.

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/8	14	2
¼	12	4
½	8	8
¾	4	12
7/8	2	14

В регистре управления DMA UART_DMACR предусмотрен бит DMAONERR, который позволяет запретить DMA от приемника в случае активного состояния линии прерывания по обнаружению ошибки UARTEINTR. При этом соответствующие линии запроса DMA: UART_RX_DMA_SREQ и UART_RX_DMA_BREQ переводятся в неактивное состояние (маскируются) до сброса UARTEINTR. На линии запроса DMA, обслуживающие передатчик, состояние UARTEINTR не влияет.
На рис. 2-7 показаны временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMA_CLR. Все сигналы должны быть синхронизированы с CPU_CLOCK. В интересах ясности изложения предполагается, что синхронизация сигналов запроса DMA в контроллере DMA не производится.

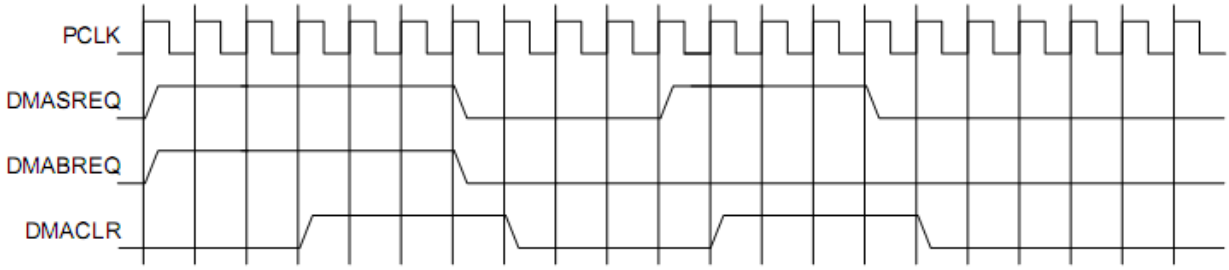


Рис. 2-7. Временные диаграммы одноэлементного и блочного запросов DMA

Прерывания

В модуле предусмотрено 11 маскируемых источников прерывания. В результате формируется один общий сигнал, представляющий собой комбинацию независимых сигналов, объединенных по схеме ИЛИ.

Сигналы запроса на прерывание:

UARTRXINTR – прерывание от приемника.
UARTTXINTR – прерывание от передатчика.
UARTRTINTR – прерывание по таймауту приемника.
UARTMSINTR – прерывание по состоянию модема:
UARTRIINTR, изменение состояния линии nUARTRI;
UARTCTSINTR, изменение состояния линии nUARTCTS;
UARTDCDINTR, изменение состояния линии nUARTDCD;
UARTDSRINTR, изменение состояния линии nUARTDSR.
UARTEINTR – ошибка:
UARTOEINTR, переполнение буфера;
UARTBEINTR, прерывание приема – разрыв линии;
UARTPEINTR, ошибка контроля четности;
UARTFEINTR, ошибка в структуре кадра.
UARTINTR – логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски UART_IMSC. Установка бита в 1 разрешает соответствующее прерывание, в 0 – запрещает.

Доступность, как индивидуальных линий, так и общей линии запроса позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика UART_RXINTR и UART_TXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UART_RXINTR и UART_TXINTR для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Прерывание по обнаружению ошибки UART_EINTR формируется в случае возникновения той или иной ошибки приема данных. Предусмотрен ряд условий формирования признака ошибки.

Прерывание по состоянию модема представляет собой комбинацию признаков изменения отдельных линий состояния модема.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний UART_RIS, либо из маскированного регистра прерываний UART_MIS.

UARTMSINTR

Прерывание по состоянию модема возникает в случае изменения любой из линий состояний модема (nUART_CTS, nUART_DCD, nUART_DSR, nUART_RI). Сброс прерывания осуществляется путем записи 1 в соответствующий (в зависимости от линии

состояния модема, вызвавшей прерывание) разряд регистра сброса прерывания UART_ICR.

UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешон и его заполнение достигло заданного порогового значения. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения данных из буфера приемника до тех пор, пока его заполнение не станет меньше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), принят один символ данных. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения одного байта данных, либо после сброса прерывания.

UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешон и его заполнение меньше или равно заданному пороговому значению. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи данных в буфера передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), данные в буферном регистре передатчика отсутствуют. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи одного байта данных, либо после сброса прерывания.

Для занесения данных в буфер FIFO передатчика необходимо записать данные в буфер либо перед разрешением работы приемопередатчика и прерываний, либо после разрешения работы приемопередатчика и прерываний.

Примечание. Прерывание передатчика работает по фронту, а не по уровню сигнала. В случае если модуль и прерывания от него разрешены до осуществления записи данных в буфер FIFO передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера FIFO.

UARTRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения (или считывания одного байта в случае, если буфер FIFO запрещен), либо путем записи 1 в соответствующий бит регистра сброса прерывания UARTICR.

UARTEINTR

Прерывание по обнаружению ошибки возникает в случае ошибки при приеме данных. Оно может быть вызвано рядом факторов:

- ошибка в структуре кадра;

- ошибка контроля четности;
- разрыв линии;
- переполнение буфера.

Причину возникновения прерывания можно определить, прочитав содержимое регистра прерываний UART_RIS, либо содержимое маскированного регистра прерываний UART_MIS.

Сброс прерывания осуществляется путем записи соответствующих бит в регистр сброса прерывания UART_ICR. За прерываниями по обнаружению ошибки закреплены биты с 7 по 10.

UARTINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

Программное управление модулем

Общая информация

Следующая информация применима ко всем регистрам контроллера:

- Базовый адрес контроллера фиксирован. Смещение каждого регистра относительно базового адреса постоянно.
- Не следует пытаться получить доступ к зарезервированным или неиспользуемым адресам. Это может привести к непредсказуемому поведению модуля.
- За исключением специально оговоренных в настоящей спецификации случаев:
  - не следует изменять значения не определенных в спецификации разрядов регистров;
  - не следует использовать значения не определенных в спецификации разрядов регистров;
  - все биты регистров (за исключением специально оговоренных случаев) устанавливаются в значение 0 после сброса по включению питания или системного сброса.
- Столбец «Тип» Таблице 3-1 определяет режим доступа к регистру в соответствии с обозначениями:
  - RW – чтение и запись;
  - RO – только чтение;
  - WO – только запись.

Обобщенные данные о регистрах устройства

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в таблице 3-1.

Таблица 3-1. Обобщенные данные о регистрах устройства.

Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x000	DR	RW	0x---	12/8	Регистр данных
0x004	RSR_ECR	RW	0x0	4/0	Регистра состояния приемника / Сброс ошибки приемника
0x008 - 0x014					Резерв
0x018	FR	RO	0b-10010---	9	Регистр флагов
0x01C					Резерв
0x020	ILPR	RW	0x00	8	Регистр управления ИК обменом в режиме пониженного энергопотребления
0x024	IBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	FBRD	RW	0x00	6	Дробная часть делителя скорости обмена данными
0x02C	LCR_H	RW	0x00	8	Регистр управления линией
0x030	CR	RW	0x0300	16	Регистр управления
0x034	IFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	IMSC	RW	0x000	11	Регистр маски прерывания
0x03C	RIS	RO	0x00-	11	Регистр состояния прерываний
0x040	MIS	RO	0x00-	11	Регистр состояния прерываний с маскированием
0x044	ICR	WO	-	11	Регистр сброса прерывания
0x048	DMACR	RW	0x00	3	Регистр управления ПДП
0x080	UARTTCR	RW	0x00	3	Регистр управления тестированием

DR

Регистр данных

В ходе передачи данных:
Если буфер FIFO передатчика разрешен, то слово данных, записанное в рассматриваемый регистр, направляется в буфер FIFO передатчика.
В противном случае, записанное слово фиксируется в буферный регистр передатчика (последний элемент буфера FIFO).
Операция записи в регистр инициирует передачу данных. Слово данных предваряется стартовым битом, дополняется битом контроля четности (если режим контроля четности включен) и стоповым битом. Сформированное слово отправляется в линию передачи данных.
В ходе приема данных:
Если буфер FIFO приемника разрешен, байт данных и четыре бита состояния (разрыв, ошибка формирования кадра, четность, переполнение) сохраняются в 12-битном буфере.
В противном случае байт данных и биты состояния записываются в буферный регистр (последний элемент буфера FIFO).

Полученные из линии связи байты данных считывается путем чтения из регистра UART_DR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра RSR_ECR (см. таблицу 3-3).

Таблица 3-2. Формат регистра DR

Бит	Наименование	Назначение
15:12		Резерв
11	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время как буфер заполнен. Сбрасывается в 0 после того, как в буфере появится свободное место.
10	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита.
9	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией UART_LCR_H (см. стр. 3-12). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
8	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит

		(корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
7:0	DATA	Принимаемые данные (чтение) Передаваемые данные (запись)

Примечание. Необходимо запрещать работу приемопередатчика перед любым перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние во время передачи или приема символа, то перед остановкой он завершает выполняемую операцию.

RSR_ECR

Регистр состояния приемника / сброса ошибки

Состояние приемника также может быть считано из регистра RSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра относится к последнему символу, считанному из регистра данных DR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним, считанным из регистра DR, байтом данных).
Запись в регистр ECR приводит к сбросу признаков ошибок переполнения, четности, структуры кадра, разрыва линии. Кроме того, все эти признаки устанавливаются в 0 после сброса устройства.
В таблице 3-3 представлено назначение бит регистра RSR_ECR.

Таблица 3-3. Регистр RSR_ECR

Биты	Наименование	Назначение
7:0		Запись в регистр сбрасывает признаки ошибок формирования кадра, проверки на четность, разрыва линии и переполнения буфера.
7:4		Резерв, при чтении результат не определен
3	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время как буфер заполнен. Сбрасывается в 0 после записи в регистр UART_ECR. Содержимое буфера остается верным, так как перезаписан был только регистр сдвига. Центральный процессор должен считать данные для того, чтобы освободить буфер FIFO.
2	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). Бит сбрасывается в 0 после записи в регистр UART_ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящемся на вершине буфера. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется

		только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита.
1	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией UART_LCR_H. Бит сбрасывается в 0 после записи в регистр UART_ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера.
0	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). Бит сбрасывается в 0 после записи в регистр UART_ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера.

Примечание. Перед чтением регистра состояния RSR необходимо считать данные, принятые из линии, путем обращения к регистру данных DR. Противоположная последовательность действий не допускается, так как регистр RSR обновляет свое состояние только после чтения регистра DR. Вместе с тем, информация о состоянии приемника может быть получена непосредственно из регистра данных DR.

FR

Регистр флагов

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. В таблице 3-4 представлена информация о назначении битов регистра.

Таблица 3-4. Регистр FR

Биты	Наименование	Назначение
15:9		Резерв. Не модифицируйте. При чтении заполняются нулями.
8	RI	Инверсия линии nUARTRI
7	TXFE	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией UART_LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр передатчика пуст. В противном случае он равен 1, если пуст буфер FIFO передатчика. Данный бит не дает никакой информации о наличии данных в регистре сдвига передатчика.
6	RXFF	Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UART_LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника занят. В противном случае он равен 1, если заполнен буфер FIFO приемника.
5	TXFF	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией

		UART_LCR_H. Если буфер FIFO запрещен, бит равен 1, когда буферный регистр передатчика занят. В противном случае он равен 1, если заполнен буфер FIFO передатчика.
4	RXFE	Буфер FIFO приемника пуст. Значение бита зависит от состояния бита FEN регистра управления линией UAR_LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника пуст. В противном случае он равен 1, если пуст буфер FIFO приемника.
3	BUSY	UART занят. Бит устанавливается в 1 в случае, если контроллер передает в линию данные. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Кроме того, бит занятости устанавливается в 1 при наличии данных в буфере FIFO передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен).
2	DCD	Инверсия линии nUARTDCD.
1	DSR	Инверсия линии nUARTDSR.
0	CTS	Инверсия линии nUARTCTS.

ILPR

Регистр управления ИК обменом в режиме пониженного энергопотребления

Этот восьмиразрядный регистр, доступный для чтения и записи, содержит значение коэффициента деления частоты UART_CLOCK, для формирования тактового сигнала IrLPBaud16. Назначение разрядов регистра показано в таблице 3-5.
Требуемое значение коэффициента деления для формирования сигнала IrLPBaud16 вычисляется по формуле: ILPDVSR = F_UART_CLOCK / F_IrLPBaud16, где номинальное значение частоты F_IrLPBaud16 составляет 1.8432 МГц.
Коэффициент деления должен быть установлен таким образом, чтобы выполнялось соотношение: 1.42 МГц < F_IrLPBaud16 < 2.12 МГц, что, в свою очередь, гарантирует формирование кодеком импульсов данных с длительностью 1.41-2.11мкс (в три раза длиннее периода сигнала IrLPBaud16).

Таблица 3-5. Регистр ILPR

Биты	Наименование	Назначение
7:0	ILPDVSR	Коэффициент деления частоты UART_CLOCK, для формирования тактового сигнала IrLPBaud16. После сброса устанавливается в 0. *Примечание.* Коэффициент 0 – запрещенное значение. В случае его установки импульсы IrLPBaud16 формироваться не будут.

Примечание. В интересах подавления помех, при работе в режиме IrDA с пониженным энергопотреблением кодек игнорирует поступающие на вход SIR_IN импульсы с длительностью, меньшей трех периодов сигнала IrLPBaud16.

IBRD

Регистр целой части делителя скорости передачи данных

Назначение бит регистра представлено в таблице 3-6.

Таблица 3-6. Регистр IBRD.

Биты	Наименование	Назначение
15:0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0.

FBRD

Регистр дробной части делителя скорости передачи данных,

Назначение бит регистра представлено в таблице 3-7.

Таблица 3-7. Регистр BFRD.

Биты	Наименование	Назначение
5:0	BAUDDIV_FRAC	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0.

Коэффициент деления вычисляется по формуле:
BAUDDIV = F_UART_CLOCK / (16 * Baud_rate),
где F_UART_CLOCK – тактовая частота контроллера UART, Baud_rate – требуемая скорость передачи данных.

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUDDIV_INT и BAUDDIV_FRAC, соответственно.

Примечания. Изменение содержимого регистров UART_IBRD и UART_FBRD вступают в силу только после завершения передачи и приема текущего символа данных.
Минимальный допустимый коэффициент деления – 1, максимальный 65535 (2^16 - 1). Таким образом, значение UART_IBRD, равное 0 является недопустимым, при этом значение регистра UART_FBRD игнорируется.
Аналогично, при UART_IBRD равном 65535 (0xFFFF), значение UART_FBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

Далее приведен пример вычисления коэффициента деления.

Пример 3-1. Вычисление коэффициента деления.

Пусть требуемая скорость передачи данных составляет 230400 бит/с, частота тактового сигнала UART_CLOCK равна 4 МГц. Тогда:

Коэффициент деления = (4*10^6)/(16*230400) = 1.085.
Таким образом, UART_BRDI = 1, UART_BRDF = 0.085.

Следовательно, значение, записываемое в регистр BFRD, равно
m = integer((0.085*64)+0.5) = 5.

Реальное значение коэффициента деления = 1+5/64 = 1.078.
Реальная скорость передачи данных = (4*10^6)/(16*1.078) = 231911 бит/с.
Ошибка установки скорости = (231911-230400)/230400 * 100% = 0.656%.

Максимальная ошибка установки скорости передачи данных с использованием шестиразрядного регистра UART_BFRD = 1/64*100% = 1.56%. Такая ошибка возникает в случае m = 1, при этом разница накапливается в течение 64 тактовых интервалов.
В таблице 3-8 представлены значения коэффициента деления для типичных скоростей передачи данных при частоте UART_CLOCK = 7.3728 МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр UART_FBRD должен быть записан ноль.

Таблица 3-8. Коэффициенты деления для типичных скоростей передачи данных при частоте UART_CLOCK = 7.3728 МГц.

Коэффициент деления	Скорость передачи данных
0x0001	460800
0x0002	230400
0x0004	115200
0x0006	76800
0x0008	57600
0x000C	38400
0x0018	19200
0x0020	14400
0x0030	9600
0x00C0	2400
0x0180	1200
0x105D	110

В таблице 3-9 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UART_CLOCK = 4 МГц.

Таблица 3-9. Коэффициенты деления для типичных скоростей передачи данных при частоте UART_CLOCK = 4 МГц.

Целая часть	Дробная часть	Требуемая скорость	Реальная скорость	Ошибка, %
0x001	0x05	230400	231911	0.656
0x002	0x0B	115200	115101	0.086
0x003	0x10	76800	76923	0.160
0x006	0x21	38400	38369	0.081
0x011	0x17	14400	14401	0.007
0x068	0x0B	2400	2400	~0
0x8E0	0x2F	110	110	~0

LCR_H

Регистр управления линией

Данный регистр обеспечивает доступ к разрядам с 29 по 22 регистра UART_LCR. При сбросе все биты регистра UART_LCR_H обнуляются.
Назначение разрядов регистра описано в таблице 3-10.

Таблица 3-10. Регистр LCR_H

Биты	Наименование	Назначение
15:8		Резерв. Не модифицируйте. При чтении выдаются нули.
7	SPS	Передача бита четности с фиксированным значением. 0 – запрещена; 1 – на месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. (При EPS=0 на месте бита четности передается 1, при EPS=1 – передается 0). Значение бита SPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено (см. таблицу 3-11).
6:5	WLEN	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0b11 – 8 бит, 0b10 – 7 бит, 0b01 – 6 бит, 0b00 – 5 бит.
4	FEN	Разрешение работы буфера FIFO приемника и передатчика. 0 – запрещено, 1 – разрешено.
3	STP2	Режим передачи двух стоповых бит. 0 – один стоповый бит, 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре.
2	EPS	Четность/нечетность. 0 – бит четности дополняет количество единиц в информационной части кадра до нечетного, 1 – до четного числа. Значение бита EPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено (см. таблицу 3-11).
1	PEN	Разрешение проверки четности. 0 – кадр не содержит бита четности, 1 – бит четности передается в кадре и проверяется при приеме данных (см. таблицу 3-11).
0	BRK	Разрыв линии. Если этот бит установлен в 1, то по завершении передачи текущего символа на выходе UART_TXDx устанавливается низкий уровень сигнала. Для правильного выполнения этой операции программное обеспечение должно обеспечить передачу сигнала разрыва в течение, как минимум, времени передачи двух информационных кадров. В нормальном режиме функционирования бит должен быть установлен в 0.

Содержимое регистров UART_LCR_H, UART_IBRD и UART_FBRD совместно образует общий 30-разрядный регистр LCR, который обновляется по стробу, формируемому при записи в UART_LCR_H. Таким образом, для того, чтобы изменение параметров

коэффициента деления частоты обмена данными вступило в силу, после их изменения значения регистров UART_IBRD и/или UART_FBRD необходимо осуществить запись данных в регистр UART_LCR_H.

Примечания.
Изменение значений трех регистров можно осуществить корректно двумя способами:
Запись UART_IBRD, запись UART_FBRD, запись UART_LCR_H;
Запись UART_FBRD, запись UART_IBRD, запись UART_LCR_H;
Для того чтобы изменить значение лишь одного из регистров (UART_IBRD или UART_FBRD) необходимо выполнить следующие шаги:
Запись UART_IBRD (или UART_FBRD), запись UART_LCR_H.

В таблице 3-11 приведена таблица истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией LCR_H.

Таблица 3-11. Управление режимом контроля четности

PEN	EPS	SPS	Бит контроля четности
0	X	X	Не передается, не проверяется
1	1	0	Проверка четности слова данных
1	0	0	Проверка нечетности слова данных
1	0	1	Бит четности постоянно равен 1
1	1	1	Бит четности постоянно равен 0

Примечания:
Регистры UART_LCR_H, UART_IBRD, and UART_FBRD не должны изменяться:
При разрешенной работе приемопередатчика;
Во время завершения приема или передачи данных в процессе остановки (перевода в запрещенное состояние) приемопередатчика.
Целостность данных в буферах FIFO не гарантируется в следующих случаях:
После установки бита разрыва линии BRK;
Если программное обеспечение произвело остановку приемопередатчика при наличии данных в буферах FIFO, после его повторного перевода в разрешенное состояние.

CR

Регистр управления
После сброса все биты регистра управления, за исключением битов 9 и 8 устанавливаются в нулевое состояние. Биты 9 и 8 устанавливаются в единичное состояние.
Назначение разрядов регистра управления показано в таблице 3-12.

Таблица 3-12. Регистр управления UART_CR

Биты	Наименование	Назначение
15	CTSEn	Разрешение управления потоком данных по CTS. 1 – разрешено, данные передаются в линию только при активном значении сигнала nUARTCTS.
14	RTSEn	Разрешение управления потоком данных по RTS. 1 – разрешено, запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере FIFO приемника.

13	Out2	Инверсия сигнала на линии состояния модема nUARTOut2. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «сигнал вызова» (RI).
12	Out1	Инверсия сигнала на линии состояния модема nUARTOut1. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «обнаружен информационный сигнал» (DCD).
11	RTS	Инверсия сигнала на линии состояния модема nUARTRTS.
10	DTR	Инверсия сигнала на линии состояния модема nUARTDTR.
9	RXE	Прием разрешен. Установка бита в 1 разрешает работу приемника. Прием данных осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчика в запрещенное состояние в ходе приема данных, он завершает прием текущего символа перед остановкой.
8	TXE	Передача разрешена. Установка бита в 1 разрешает работу передатчика. Передача осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчик в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой.
7	LBE	1 – шлейф разрешен, 0 – запрещен. В режиме разрешенного шлейфа: Если установлены бит SIREN=1 и бит регистра управления тестированием TCR SIRTEST=1, то сигнал с выхода кодека nSIR_OUT инвертируется и подается на вход кодека SIR_IN. Бит SIRTEST устанавливается в 1 для того, чтобы вывести устройство из полудуплексного режима, характерного для интерфейса SIR. После окончания тестирования по шлейфу бит SIRTEST должен быть установлен в 0. Если бит SIRTEST=0, то выходная линия передатчика UART_TXDx коммутируется на вход приемника UART_RXDx. Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии. После сброса бит устанавливается в 0.
6:3		Резерв. Не модифицируйте. При чтении выдаются нули.
2	SIRLP	Выбор режима ИК обмена с пониженным энергопотреблением: 0 – длительность импульсов данных равна 3/16 длительности передачи бита. 1 – длительность импульсов данных равна трем тактам сигнала IrLPBaud16 вне зависимости от выбранной скорости передачи данных. Выбор этого режима снижает

		энергопотребление, однако может привести к уменьшению дальности связи.
1	SIREN	Разрешение работы кодека ИК передачи данных IrDA SIR: 0 – запрещен. Сигнал nSIR_OUT находится в низком состоянии, данные на входе SIR_IN не обрабатываются. 1 – разрешен. Данные передаются на выход nSIR_OUT и принимаются с входа SIR_IN. Линия UART_TXDx находится в высоком состоянии. Данные на входе UART_RXDx и линиях состояния модема не обрабатываются. В случае если UARTEN=0 значение бита не играет роли.
0	UARTEN	Разрешение работы приемопередатчика: 0 – работа запрещена. Перед остановкой завершается прием и/или передача обрабатываемого в текущий момент символа. 1 – работа разрешена. Производится обмен данными либо по линиям асинхронного обмена, либо по линиям ИК обмена SIR, в зависимости от состояния бита SIREN.

Примечание. Для того чтобы разрешить передачу данных, необходимо установить в 1 биты TXE и UARTEN. Аналогично, для разрешения приема данных необходимо установить в 1 биты RXE и UARTEN.

Примечание. Рекомендуется следующая последовательность действий для программирования регистров управления:
Остановите работу приемопередатчика.
Дождаться окончания приема и/или передачи текущего символа данных.
Сбросьте буфер передатчика путем установки бита FEN регистра UART_LCR_H в 0.
Изменить настройки регистра UART_CR.
Возобновите работу приемопередатчика.

IFLS

Регистр порога прерывания по заполнению буфера FIFO

Данный регистр используется для установки порогового значения заполнения буферов передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода величины заполнения буфера через заданное значение.
После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра и значения его битов представлены в таблице 3-13.

Таблица 3-13. Регистр UART_IFLS.

Биты	Наименование	Назначение
15:6		Резерв. Не модифицируйте. При чтении выдаются нули.
5:3	RXIFLSEL	Порог прерывания по заполнению буфера приемника: b000 = Буфер заполнен на 1/8 b001 = Буфер заполнен на 1/4 b010 = Буфер заполнен на 1/2 b011 = Буфер заполнен на 3/4

		b100 = Буфер заполнен на 7/8 b101-b111 = резерв.
2:0	TXIFLSEL	Порог прерывания по заполнению буфера передатчика: b000 = Буфер заполнен на 1/8 b001 = Буфер заполнен на 1/4 b010 = Буфер заполнен на 1/2 b011 = Буфер заполнен на 3/4 b100 = Буфер заполнен на 7/8 b101-b111 = резерв.

IMSC

Регистр установки сброса маски прерывания

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание.
После сброса все биты регистра маски устанавливаются в нулевое состояние.
Назначение битов регистра UART_IMSC показано в таблице 3-14.

Таблица 3-14. Регистр UART_IMSC

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OEIM	Маска прерывания по переполнению буфера UARTOEINTR. 1 – установлена; 0 – сброшена.
9	BEIM	Маска прерывания по разрыву линии UARTBEINTR. 1 – установлена; 0 – сброшена.
8	PEIM	Маска прерывания по ошибке контроля четности UARTPEINTR. 1 – установлена; 0 – сброшена.
7	FEIM	Маска прерывания по ошибке в структуре кадра UARTFEINTR. 1 – установлена; 0 – сброшена.
6	RTIM	Маска прерывания по таймауту приема данных UARTRTINTR. 1 – установлена; 0 – сброшена.
5	TXIM	Маска прерывания от передатчика UARTTXINTR. 1 – установлена; 0 – сброшена.
4	RXIM	Маска прерывания от приемника UARTRXINTR. 1 – установлена; 0 – сброшена.
3	DSRMIM	Маска прерывания UARTDSRINTR по изменению состояния линии nUARTDSR. 1 – установлена, 0 – сброшена.

2	DCDMIM	Маска прерывания UARTDCDINTR по изменению состояния линии nUARTDCD. 1 – установлена, 0 – сброшена.
1	CTSMIM	Маска прерывания UARTCTSINTR по изменению состояния линии nUARTCTS. 1 – установлена, 0 – сброшена.
0	RIMIM	Маска прерывания UARTRIINTR по изменению состояния линии nUARTRI. 1 – установлена, 0 – сброшена.

RIS

Регистр состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.
Предупреждение. После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено.
Назначение бит в регистре UART_RIS представлено в таблице 3-15.

Таблица 3-15. Регистр UART_RIS.

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OERIS	Состояние прерывания по переполнению буфера UARTOEINTR. 1 – буфер приемника переполнен; 0 – буфер приемника не переполнен.
9	BERIS	Состояние прерывания по разрыву линии UARTBEINTR. 1 – произошел разрыв линии приема; 0 – разрыва линии приема не происходило.
8	PERIS	Состояние прерывания по ошибке контроля четности UARTPEINTR. 1 – возникла ошибка контроля четности; 0 – ошибки контроля четности не возникало.
7	FERIS	Состояние прерывания по ошибке в структуре кадра UARTFEINTR. 1 – возникла ошибка в структуре кадра; 0 – ошибки в структуре кадра не возникало.
6	RTRIS	Состояние прерывания по таймауту приема данных UARTRTINTR1. 1 – вылшло время таймаута приема данных; 0 – время таймаута приема данных не вышло.
5	TXRIS	Состояние прерывания от передатчика UARTTXINTR. 1 – возникло прерывание от передатчика; 0 – прерывания от передатчика нет.

[1] Сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Поэтому чтение состояния прерывания по таймауту из регистров MIS and RIS даст одинаковый результат.

4	RXRIS	Состояние прерывания от приемника UARTRXINTR. 1 – возникло прерывание от приемника; 0 – прерывание от приемника не возникало.
3	DSRRMIS	Состояние прерывания UARTDSRINTR по изменению линии nUARTDSR.
2	DCDRMIS	Состояние прерывания UARTDCDINTR по изменению линии nUARTDCD.
1	CTSRMIS	Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS.
0	RIRMIS	Состояние прерывания UARTRIINTR по изменению линии nUARTRI.

MIS

Регистр маскированного состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.
После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено.
Назначение бит в регистре UART_MIS представлено в таблице 3-16.

Таблица 3-16. Регистр UART_MIS.

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OEMIS	Маскированное состояние прерывания по переполнению буфера UARTOEINTR. 1 – буфер приемника переполнен; 0 – буфер приемника не переполнен.
9	BEMIS	Маскированное состояние прерывания по разрыву линии UARTBEINTR. 1 – произошел разрыв линии приема; 0 – разрыва линии приема не происходило.
8	PEMIS	Маскированное состояние прерывания по ошибке контроля четности UARTPEINTR. 1 – возникла ошибка контроля четности; 0 – ошибки контроля четности не возникало.
7	FEMIS	Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR. 1 – возникла ошибка в структуре кадра; 0 – ошибки в структуре кадра не возникало.
6	RTMIS	Маскированное состояние прерывания по таймауту приема данных UARTRTINTR. 1 – вылшло время таймаута приема данных; 0 – время таймаута приема данных не вышло.
5	TXMIS	Маскированное состояние прерывания от передатчика UARTTXINTR.

		1 – возникло прерывание от передатчика; 0 – прерывания от передатчика нет.
4	RXMIS	Маскированное состояние прерывания от приемника UARTRXINTR. 1 – возникло прерывание от приемника; 0 – прерывание от приемника не возникало.
3	DSRMMIS	Маскированное состояние прерывания UARTDSRINTR по изменению линии nUARTDSR.
2	DCDMMIS	Маскированное состояние прерывания UARTDCDINTR по изменению линии nUARTDCD.
1	CTSMMIS	Маскированное состояние прерывания UARTCTSINTR по изменению линии nUARTCTS.
0	RIMMIS	Маскированное состояние прерывания UARTRIINTR по изменению линии nUARTRI.

ICR

Регистр сброса прерываний

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись 0 в любой из разрядов регистра игнорируется.
Назначение бит в регистре UART_ICR представлено в таблице 3-17.

Таблица 3-17. Регистр UART_ICR.

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OEIC	Сброс прерывания по переполнению буфера UARTOEINTR.
9	BEIC	Сброс прерывания по разрыву линии UARTBEINTR.
8	PEIC	Сброс прерывания по ошибке контроля четности UARTPEINTR.
7	FEIC	Сброс прерывания по ошибке в структуре кадра UARTFEINTR.
6	RTIC	Сброс прерывания по таймауту приема данных UARTRTINTR.
5	TXIC	Сброс прерывания от передатчика UARTTXINTR.
4	RXIC	Сброс прерывания от приемника UARTRXINTR.
3	DSRMIC	Сброс прерывания UARTDSRINTR по изменению линии nUARTDSR.
2	DCDMIC	Сброс прерывания UARTDCDINTR по изменению линии nUARTDCD.
1	CTSMIC	Сброс прерывания UARTCTSINTR по изменению линии nUARTCTS.
0	RIMIC	Сброс прерывания UARTRIINTR по изменению линии nUARTRI.

DMACR

Регистр управления прямым доступом к памяти

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются. Назначение бит регистра UART_DMACR представлено в таблице 3-18.

Таблица 3-18. Регистр UART_DMACR.

Биты	Наименование	Назначение
15:13		Резерв. Не модифицируйте. При чтении выдаются нули.
2	DMAONERR	Если бит установлен в 1, в случае возникновения прерывания по обнаружению ошибки блокируются запросы DMA от приемника UART_RX_DMA_SREQ и UART_RX_DMA_BREQ.
1	TXDMAE	Использование DMA при передаче. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика.
0	RXDMAE	Использование DMA при приеме. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO приемника.

UARTTCR

Регистр управления тестированием

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются. Назначение бит регистра UARTTCR представлено в таблице 3-19.

Таблица 3-19. Регистр UARTTCR.

Биты	Наименование	Назначение
15:13		Резерв. При чтении значение непредсказуемо.
2	SIRTEST	Разрешение приёма данных в кольцевом режиме с выхода IrDA передатчика. 1- разрешено 0 – запрещено Используется совместно с установкой бита LBE в регистре UART_CR
1	TESTFIFO	Разрешение чтения данных из FIFO передатчика и запись в FIFO приёмника. 1- разрешено 0 - запрещено
0	ITEN	Перевод контроллера UART в тестовый режим 1- тестовый режим разрешён 0- тестовый режим запрещён

Контроллер прямого доступа в память DMA

Основные свойства контроллера DMA

Основные свойства и отличительные особенности:

- 32 канала DMA;
- каждый канал DMA имеет свои сигналы управления передачей данных;
- каждый канал DMA имеет программируемый уровень приоритета;
- каждый уровень приоритета обрабатывается, исходя из уровня приоритета, определяемого номером канала DMA;
- поддержка различного типа передачи данных:
  - память – память;
  - память – периферия;
  - периферия – память;
- поддержка различных типов DMA циклов;
- поддержка передачи данных различной разрядности;
- каждому каналу DMA доступна первичная и альтернативная структура управляющих данных канала;
- все управляющие данные канала хранятся в системной памяти;
- разрядность данных приемника равна разрядности данных передатчика;
- количество передач в одном цикле DMA может программироваться от 1 до 1024;
- инкремент адреса передачи может быть больше чем разрядность данных.

Термины и определения

При описании блока используются следующие термины:

Альтернативная	Альтернативная структура управляющих данных канала. Вы можете установить соответствующий регистр для изменения типа структуры данных (см. раздел «Структура управляющих данных канала»)
C	Идентификатор номера канала прямого доступа. Например: C=1 – канал DMA 1 C=23 – канал DMA 23
Канал	Возможны конфигурации контроллера с числом каналов до 32. Каждый канал содержит независимые сигналы управления передачей данных, которые могут инициировать передачу данных по каналу DMA
Управляющие данные канала	Структура данных находится в системной памяти. Вы можете программировать эту структуру данных так, что контроллер может выполнять передачу данных по каналу DMA в желаемом режиме. Контроллер должен иметь доступ к области системной памяти, где находится эта информация. *Примечание.* Любое упоминание в спецификации структуры данных означает управляющие данные канала
Цикл DMA	Все передачи DMA, которые контроллер должен выполнить для передачи N пакетов данных

Передача DMA	Акция пересылки одного байта, полуслова или слова. Общее количество передач DMA, которые контроллер выполняет для канала
Пинг – понг	Режим работы для выбранного канала, при котором контроллер получает начальный запрос и затем выполняет цикл DMA, используя первичную или альтернативную структуру данных. После завершения этого цикла DMA контроллер начинает выполнять новый цикл DMA, используя другую (первичную или альтернативную) структуру данных. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя главному процессору перенастраивать неактивную структуру данных. Контроллер продолжает переключаться от первичной к альтернативной структуре данных и обратно до тех пор, пока он не прочитает «неправильную» структуру данных или пока он не завершит цикл без переключения к другой структуре
Первичная	Первичная структура управляющих данных канала. Контроллер использует эту структуру данных, если соответствующий разряд в регистре chnl_pri_alt_set установлен в 0.
R	Степень числа 2, устанавливающее число передач DMA, которые могут произойти перед сменой арбитража. Количество передач DMA программируется в диапазоне от 1 до 1024 двоичными шагами от 2 в степени 0 до 2 в степени 100
Исполнение с изменением конфигурации	Режим работы для выбранного канала, при котором контроллер получает запрос от периферии и выполняет 4 DMA передачи, используя первичную структуру управляющих данных, которые настраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. После того, как цикл закончится и если периферия устанавливает новый запрос на обслуживание, контроллер выполняет снова 4 DMA передачи, используя первичную структуру управляющих данных, которые опять перенастраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. Контроллер будет продолжать работать вышеописанным способом до тех пор, пока не прочитает неправильную структуру данных или процессор не установит альтернативную структуру данных для обычного цикла. Контроллер устанавливает флаг dma_done, если окончание подобного режима работы происходит после выполнения обычного цикла

Функциональное описание.

На Рисунке 2-1 показана упрощенная структурная схема контроллера.

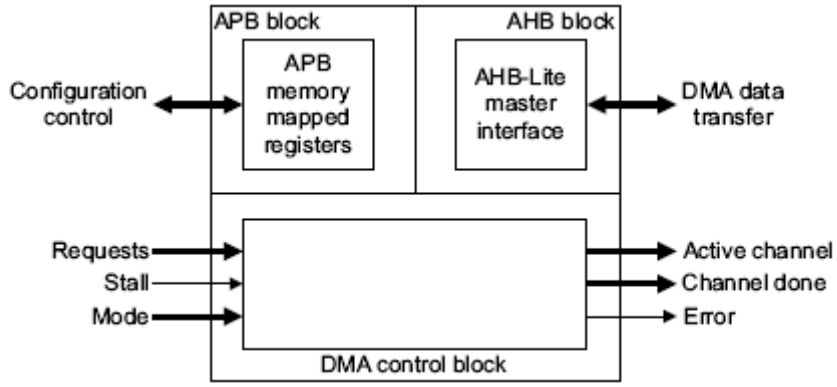


Рисунок 2-1. Структурная схема контроллера

Контроллер состоит из следующих основных функциональных блоков
- блок, подключенный к шине APB
- блок, подключенный к шине AHB
- управляющий блок DMA

Распределение каналов DMA

Номер канала	Источник req	Источник sreq	Описание
0	UART1_TX_DMA_BREQ	UART1_TX_DMA_SREQ	Запрос от передатчика UART1
1	UART1_RX_DMA_BREQ	UART1_RX_DMA_SREQ	Запрос от приёмника UART1
2	UART2_TX_DMA_BREQ	UART2_TX_DMA_SREQ	Запрос от передатчика UART2
3	UART2_RX_DMA_BREQ	UART2_RX_DMA_SREQ	Запрос от приёмника UART2
4	SSP1_TX_DMA_BREQ	SSP1_TX_DMA_SREQ	Запрос от передатчика SPI1
5	SSP1_RX_DMA_BREQ	SSP1_RX_DMA_SREQ	Запрос от приёмника SPI1
6	SSP2_TX_DMA_BREQ	SSP2_TX_DMA_SREQ	Запрос от передатчика SPI2
7	SSP2_RX_DMA_BREQ	SSP2_RX_DMA_SREQ	Запрос от приёмника SPI2
8	SSP3_TX_DMA_BREQ	SSP3_TX_DMA_SREQ	Запрос от передатчика SPI3
9	SSP3_RX_DMA_BREQ	SSP3_RX_DMA_SREQ	Запрос от приёмника SPI3
10	TIM1_DMA_REQ	TIM1_DMA_REQ	Запрос от таймера общего назначения TIMER1
11	TIM2_DMA_REQ	TIM2_DMA_REQ	Запрос от таймера

			общего назначения TIMER2
12	TIM3_DMA_REQ	TIM3_DMA_REQ	Запрос от таймера общего назначения TIMER3
13	TIM4_DMA_REQ	TIM4_DMA_REQ	Запрос от таймера общего назначения TIMER4
14	-	TIM1_DMA_REQ1	Запрос от канала 1 таймера общего назначения TIMER1
15	-	TIM1_DMA_REQ2	Запрос от канала 2 таймера общего назначения TIMER1
16	-	TIM1_DMA_REQ3	Запрос от канала 3 таймера общего назначения TIMER1
17	-	TIM1_DMA_REQ4	Запрос от канала 4 таймера общего назначения TIMER1
18	-	TIM2_DMA_REQ1	Запрос от канала 1 таймера общего назначения TIMER2
19	-	TIM2_DMA_REQ2	Запрос от канала 2 таймера общего назначения TIMER2
20	-	TIM2_DMA_REQ3	Запрос от канала 3 таймера общего назначения TIMER2
21	-	TIM2_DMA_REQ4	Запрос от канала 4 таймера общего назначения TIMER2
22	-	TIM3_DMA_REQ1	Запрос от канала 1 таймера общего назначения TIMER3
23	-	TIM3_DMA_REQ2	Запрос от канала 2 таймера общего назначения TIMER3
24	-	TIM3_DMA_REQ3	Запрос от канала 3 таймера общего назначения TIMER3
25	-	TIM3_DMA_REQ4	Запрос от канала 4 таймера общего назначения TIMER3
26	-	TIM4_DMA_REQ1	Запрос от канала 1 таймера общего назначения TIMER4
27	-	TIM4_DMA_REQ2	Запрос от канала 2 таймера общего назначения TIMER4

28	-	TIM4_DMA_REQ3	Запрос от канала 3 таймера общего назначения TIMER4
29	-	TIM4_DMA_REQ4	Запрос от канала 4 таймера общего назначения TIMER4
30	-	ADC_DMA_SREQ	Запрос от АЦП последовательных приближений
31	-	-	Программный

Блок, подключенный к шине APB

Блок содержит набор регистров, позволяющих настраивать контроллер, используя ведомый APB интерфейс. Регистры занимают адресное пространство емкостью 4 кбайт.

Блок, подключенный к шине AHB

Контроллер содержит один блок типа «ведущий» шины DMA Bus, который позволяет, используя 32-х разрядную шину, передавать данные от источника к приемнику. Источник и приемник являются ведомыми шины AHB.

Управляющий блок DMA

Этот блок содержит схему управления, позволяющую реализовать следующие функции:

- осуществление арбитража поступающих запросов;
- индикацию активного канала;
- индикацию завершения обмена по каналу;
- индикацию состояния ошибки обмена по шине DMA Bus;
- разрешение медленным устройствам приостанавливать исполнение цикла DMA;
- ожидание запроса на очистку до завершения цикла DMA;
- осуществление одиночных или множественных передач DMA для каждого запроса;
- осуществление следующих типов DMA передач:
  - память – память;
  - память – периферия;
  - периферия – память.

Типы передач

Контроллер интерфейса не поддерживает пакетные передачи. Контроллер выполняет одиночные передачи. Отсутствие возможности осуществлять пакетные передачи оказывает минимальное влияние на производительность системы, так как пакетные передачи более эффективны в одноуровневых системах с шиной AHB, где блоки должны «захватывать» шину или обращаться к внешней памяти. В тоже время контроллер DMA предназначен для использования в многоуровневых системах с шиной AHB, включающих встроенную память.

Разрядность передач данных

Контроллер интерфейса предоставляет возможность осуществлять передачу 8, 16 и 32 разрядных данных. Таблица перечисляет значения комбинаций шины HSIZE.

Таблица 2.1 Комбинации шины HSIZE

HSIZE[2]*	HSIZE[1]	HSIZE[0]	Разрядность данных (бит)
0	0	0	8
0	0	1	16
	1	0	32
	1	1	**

* - сигнал постоянно удерживается в состоянии логический ноль.
**- запрещенная комбинация

Контроллер всегда использует передачи 32-х разрядными данными при обращении к управляющим данным канала. Необходимо устанавливать разрядность данных источника соответствующую разрядности данных приемника.

Управление защитой данных

Контроллер позволяет устанавливать режимы защиты данных протокола AHB-Lite, определяемые шиной HPROT[3:1]. Возможен выбор следующих режимов защиты:

- кэширование;
- буферизация;
- привилегированный.

Таблица 2-3 перечисляет значения комбинаций шины HPROT.

Таблица 2-3 Режимы защиты данных.

HPROT[3] Кэширование	HPROT[2] буферизация	HPROT[1] Привелигиро-ванный	HPROT[0] Данные/команда	Описание
-	-	-	1*	Доступ к данным
-	-	0	-	Пользовательский доступ
-	-	1	-	Привилегированный доступ
-	0	-	-	Без буферизации
-	1	-	-	Буферизованный
0	-	-	-	Без кэширования
1	-	-	-	Кэшированный

Контроллер удерживает HPROT[0] в состоянии логической единицы, чтобы обозначить доступ к данным.

Для каждого цикла DMA возможен выбор режимов защиты данных передач источника и приемника. Более подробно это описано в разделе «Настройка управляющих данных».

Для каждого канала DMA также возможен выбор режима защиты данных. Более подробно это описано в разделе Управление DMA.

Инкремент адреса

Контроллер позволяет управлять инкрементом адреса при чтении данных из источника и при записи данных в приемник. Инкремент адреса зависит от разрядности передаваемых данных. В следующей таблице перечислены возможные комбинации.

Таблица 2-4 Инкремент адреса.

Разрядность данных	Величина инкремента
8	Байт, полуслово, слово
16	Полуслово, слово
32	слово

Минимальная величина инкремента адреса всегда соответствует разрядности передаваемых данных. Максимальная величина инкремента адреса, осуществляемая контроллером, одно слово. Более подробно о настройке инкремента адреса написано в разделе Настройка управляющих данных. Этот раздел описывает разряды управления величиной инкремента адреса в управляющих данных канала.

Примечание. Если необходимо оставлять адрес неизменным при чтении или записи данных, для примера, при работе с FIFO, можно соответствующим образом настроить контроллер на работу с фиксированным адресом (см. раздел «Структура управляющих данных канала»).

Управление DMA

Правила обмена данными

Контроллер использует правила обмена данными, перечисленные далее в Таблице 2-5, при соблюдении следующих условий:

- канал DMA включен, что выполняется установкой в состояние логической единицы разрядов управления chnl_enable_set[C] и master_enable;
- флаги запроса dma_req[C] и dma_sreq[C] не замаскированы, что выполняется установкой в состояние логического нуля разряда управления chnl_req_mask_set [C];
- контроллер находится не в тестовом режиме, что выполняется установкой в состояние логического нуля разряда управления int_test_en_bit[C].

Таблица 2-5. Правила, при которых передача данных по каналам разрешена, и запросы не маскируются.

Правило	Описание
1	Если dma_active[C] установлен в 0, то установка в 1 dma_req[C] или dma_sreq[C] на один или более тактов сигнала hclk, следующих или не следующих друг за другом, инициирует передачу по каналу номер C
2	Контроллер осуществляет установку в 1 только одного разряда dma_active[C]
3	Контроллер устанавливает в 1 dma_active[C] в момент начала передачи по каналу C
4	Для типов циклов DMA, отличных от периферийного «Исполнение с

	изменением конфигурации», dma_active[C] остается в состоянии 1 до тех пор, пока контроллер не окончит передачи с номерами меньше, чем значение 2^R или чем число передач, указанное в регистре n_minus_1. В периферийном режиме «Исполнение с изменением конфигурации», dma_active[C] остается в состоянии 1 в течение каждой пары DMA передач, с использованием первичной и альтернативной структур управляющих данных. Таким образом, контроллер выполняет 2^R передач, используя первичную структуру управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных. По окончании последней передачи dma_active[C] сбрасывается в 0
5	Контроллер устанавливает dma_active[C] в 0 как минимум на один такт сигнала hclk перед тем, как снова установит dma_active[C] или dma_active[] в 1
6	Для каналов, по которым разрешена передача, контроллер осуществляет установку в 1 только одного dma_done[]
7	Если dma_req[C] устанавливается в состояние 1 в момент, когда dma_active[C] или dma_stall также в состоянии 1, то это означает, что контроллер обнаружил запрос
8	Если разряды cycle_ctrl для канала установлены в состояние 3'b100, 3'b101, 3'b110, 3'b111, то dma_done[C] никогда не будет установлен в 1
9	Если все передачи по каналу завершены, и разряды cycle_ctrl позволяют удержание dma_done[C], то по срезу сигнала dma_active[] произойдут события: - если dma_stall в состоянии 0, контроллер устанавливает dma_done[] в состояние 1 продолжительностью один такт hclk - если dma_stall в состоянии 1, работа контроллера приостановлена. После того, как dma_stall будет установлен в 0, контроллер устанавливает dma_done[] в состояние 1 продолжительностью один такт hclk
10	Состояние dma_waitonreq[C] можно изменять только при выключенном канале.
11	Если dma_waitonreq[C] в состоянии 1, то сигнал dma_active[C] не перейдет в состояние 0 до тех пор, пока: – контроллер завершит 2^R передач (или число передач, указанное в регистре n_minus_1); – dma_req[C] будет установлен в 0; – dma_sreq[C] будет установлен в 0
12	Если за один такт сигнала hclk перед установкой dma_active[C] в 0 dma_stall устанавливается в 1, то – контроллер установит dma_active[C] в 0 на следующем такте сигнала hclk; – передача по каналу C не завершится, пока не будет сброшен в 0 dma_stall
13	Контроллер игнорирует dma_sreq[C], если dma_waitonreq[C] в состоянии 0

14	Контроллер игнорирует dma_sreq[C], если chnl_useburst_set[C] в состоянии 1[*)]
15	Для циклов DMA, отличных по типу от периферийного режима «Исполнение с изменением конфигурации», по окончании 2^R передач контроллер устанавливает значение chnl_useburst_set[C] в состояние 0, если количество оставшихся передач меньше, чем 2^R. В периферийном режиме «Исполнение с изменением конфигурации» контроллер устанавливает значение chnl_useburst_set[C] в состояние 0 только, если количество оставшихся передач с использованием альтернативной структуры управляющих данных меньше, чем 2^R.
16	Для типов циклов DMA, отличных от периферийного режима «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, то контроллер выполняет одну DMA передачу. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных. Затем без осуществления арбитража выполняет одну передачу, используя альтернативную структуру управляющих данных
17	Для типов циклов DMA, отличных от периферийного режима «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1, а dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c], и контроллер выполняет 2^R (или число передач, указанное в регистре n_minus_1) DMA передач. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c], и контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных
18	Когда chnl_req_mask_set[C] установлен в 1, контроллер игнорирует запросы по dma_sreq[C] и dma_req[C]

[*)] Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре n_minus_1 меньше, чем значение 2^R, то контроллер не очистит разряды chnl_useburst_set и поэтому запросы по dma_sreq[C] будут маскированы. Если периферия не устанавливает dma_req[C] в состояние 1, то контроллер никогда не выполнит необходимых передач.

При отключении канала контролер осуществляет DMA передачи согласно правилам представленным в таблице 2-6.

Таблица 2-6. Правила осуществления DMA передач при «запрещенных» каналах.

Правило	**Описание**

19	Если dma_req[C] установлен в 1, то контроллер устанавливает dma_done[C] в 1. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен)
20	Если dma_sreq[C] установлен в 1, то контроллер устанавливает dma_done[C] в 1 при условии dma_waitonreq[C] в 1 и chnl_useburst_set[C] в состоянии 0. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен)
21	dma_active[C] всегда удерживается в состоянии 0

Диаграммы работы контроллера DMA.

Данный раздел описывает следующие примеры функционирования контроллера с использованием правил обмена данными представленными в таблице 2-5:
- импульсный запрос на обработку
- запрос по уровню на обработку
- флаги завершения
- флаги ожидания запроса на обработку

Примечание:
Все диаграммы показанные на рисунках 2-6 – 2-10 подразумевают следующее:
- hready находится в состоянии 1
- AHB «ведомый» всегда дает ответ «OKAY»

Импульсный запрос на обработку.

Рисунок 2-6 показывает временную диаграмму работы контроллера DMA при получении импульсного запроса от периферии.

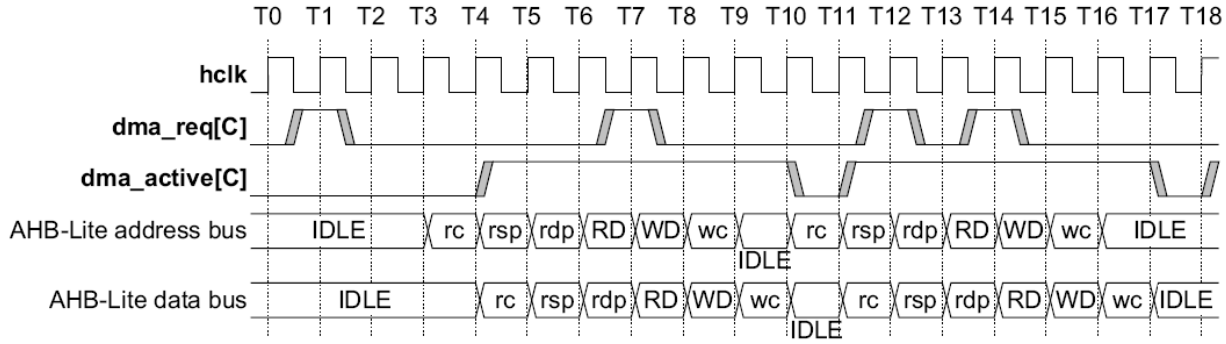


Рисунок 2-6. Диаграмма работы при получении импульсного запроса от периферийного блока.

Пояснения к рис. 2-6.

T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18) .
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C.
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr.
T7	При установленном dma_active[C] в 1 и при условии, что chnl_req_mask_set[C] находится в состоянии 0, контроллер обнаруживает импульс запроса на обработки по каналу C (см. правило 7). Контроллер обработает этот запрос в течение следующего арбитража
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg.
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4).
T10-T11	Контроллер удерживает dma_active[C] как минимум на один такт hclk (см. правило 5).
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C], так как ранее на такте T7 был получен запрос на обработку (см. правила 2 и 3).
T12	При установленном dma_active[C] в 1 и при условии, что chnl_req_mask_set[C] находится в состоянии 0, контроллер обнаруживает импульс запроса на обработку по каналу C (см. правило 7). Контроллер обработает этот запрос в течение следующего арбитража.
T14	Контроллер игнорирует запрос по каналу C из-за отложенного запроса полученного на такте T12.

T17	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4).
T17-T18	Контроллер удерживает dma_active[C] как минимум на один такт hclk (см. правило 5).
T18	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C], так как ранее на такте T12 был получен запрос на обработку (см. правила 2 и 3).

Запрос на обработку по уровню.

Рисунок 2-7 показывает временную диаграмму работы контроллера DMA при получении от периферии запроса на обработку по уровню.

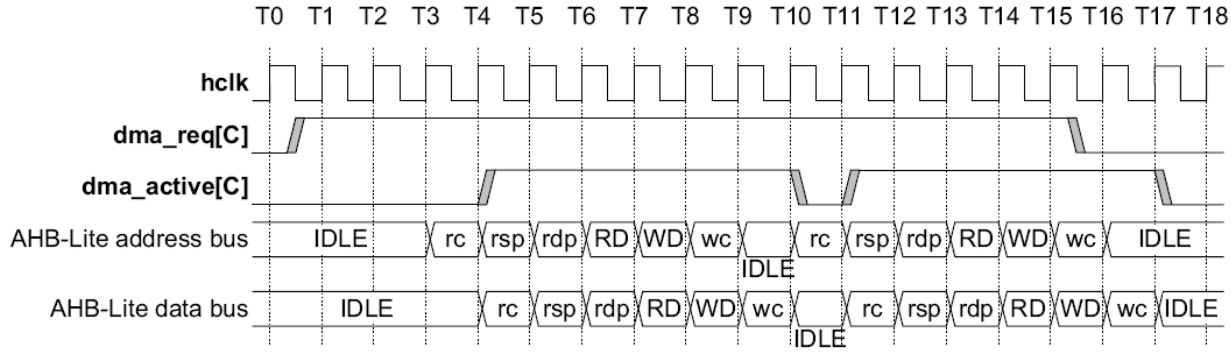


Рисунок 2-7. Диаграмма работы при получении от периферийного блока запроса на обработку по уровню.

Пояснения к рис. 2-7.

T1	Контроллер обнаружил запрос на обработку по каналу C (Таблица 2-5), правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C.
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr.
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных.
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg.
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4). Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
T10-T11	Контроллер удерживает dma_active[C] как минимум на один такт hclk (см. правило 5).
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C] и начинает вторую DMA передачу по каналу C.

T11-T14	Контроллер считывает управляющие данные канала.
T14-T16	Контроллер выполняет передачу DMA по каналу C.
T15-T16	Периферийный блок обнаруживает, что передача DMA началась и сбрасывает dma_req[C].
T16-T17	Контроллер осуществляет запись настроек канала channel_cfg.
T17	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4).

При использовании запроса на обработку по уровню, периферийный блок может не обладать достаточным быстродействием, чтобы во время снять сигнал запроса, в этом случае он должен установить сигнал dma_stall. Установка сигнала dma_stall предотвращает повторение выполненной передачи.

Флаги завершения.

Рисунок 2-8 демонстрирует функционирование сигнала (флага) dma_done[] при следующих условиях:

- dma_stall и dma_waitonreq[] находятся в состоянии 0;
- dma_stall установлен в 1;
- dma_waitonreq[]установлен в 1.

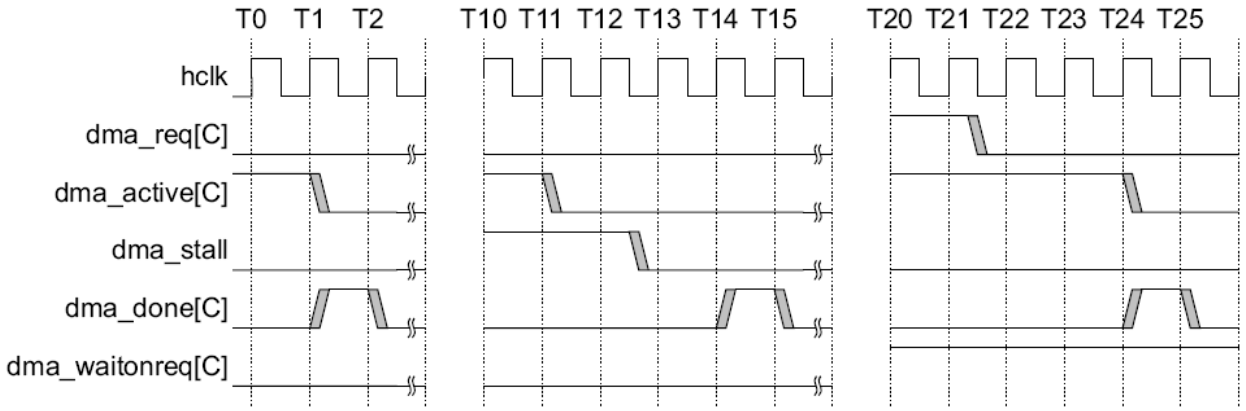


Рисунок 2-8. Диаграммы функционирования dma_done.

Пояснения к рис. 2-8. от T0 до T2

T1	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см.Таблица 2-5, правило 4).
T1-T2	Контроллер завершает цикл DMA, и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6).

Пояснения к рис. 2-8. от T10 до T15

T11	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4).
T12-T13	Периферийный блок сбрасывает сигнал dma_stall.

T14-T15	Контроллер завершает цикл DMA, и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6).

Примечание к T11. Контроллер не устанавливает сигнал dma_done[C], так как сигнал dma_stall установлен в 1 в предшествующем такте hclk (см. правила 9 и 12).

Пояснения к рис. 2-8. от T20 до T25

T20	Контроллер выполнил передачу DMA, но из-за установленного в 1 dma_waitonreq[C] он должен ожидать сброса в 0 сигнала dma_req[C], перед тем как сбросить dma_active[C] (см. правило 11) и установить dma_done[C] (см. правило 9).
T21-T25	Периферийный блок сбрасывает dma_req[C].
T24	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4).
T24-T25	Контроллер завершает цикл DMA и, если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт hclk (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6)

Флаги ожидания запроса на обработку.

Ниже приведены рисунки, которые демонстрируют примеры использования флагов ожидания запроса на обработку при выполнении 2^R передач и одиночных передач:

- диаграмма работы контроллера DMA при использовании периферией dma_waitonreq;

диаграмма работы контроллера DMA при использовании периферией dma_waitonreq совместно с dma_sreq.

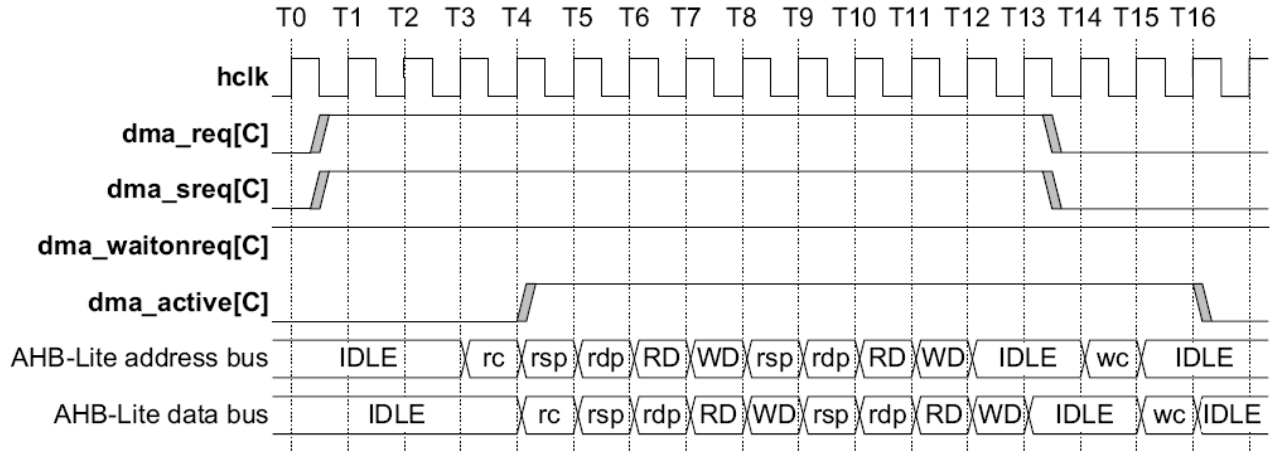


Рисунок 2-9. Диаграмма работы контроллера DMA при использовании периферией dma_waitonreq.

Пояснения к рис. 2-9.

T0-T16	Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10).
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1)

	при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
T3-T4	Периферийный блок удерживает dma_req[C] и dma_sreq[C] в 1. Контроллер игнорирует dma_sreq[C] запрос и отвечает на dma_req[C] запрос (см. правила 16 и 17).
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr.
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных.
T9-T11	Контроллер считывает 2 указателя адреса окончания данных rsp и rdp.
T11-T13	Периферийный блок сбрасывает сигналы dma_req[C] и dma_sreq[C].
T15-T16	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg.
T16	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11). Контроллер устанавливает значение по чтению регистра chnl_useburst_set[C] в 0, если количество оставшихся передач менее 2^R (см. правило 15).

Рисунок 2-10 показывает работу контроллера DMA при установке dma_waitonreq в 1 и выполнении одиночной DMA передачи.

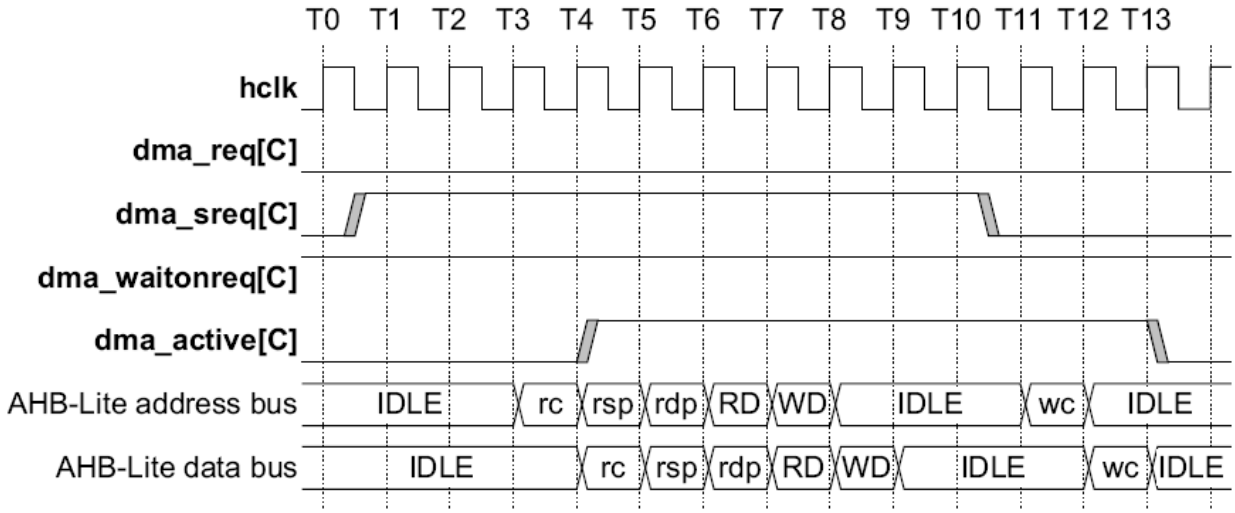


Рисунок 2-10. Диаграмма работы контроллера DMA при использовании периферией dma_waitonreq совместно с dma_sreq.

Пояснения к рис. 2-10.

T0-T13	Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10).
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_useburst_set[C] находится в состоянии 0 (см. правила

	13 и 14).
T3-T4	Контроллер отвечает на dma_sreq[C] запрос (см. правила 16).
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C.
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr.
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных. Это запрос в ответ на dma_sreq[], таким образом, R=0 и, следовательно, контроллер исполнит 1 DMA передачу.
T10-T11	Периферийный блок сбрасывает сигнал dma_sreq[C].
T12_T13	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg.
T13	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11).

Правила арбитража DMA

Контроллер имеет возможность настройки момента арбитража при передачах DMA. Эта возможность позволяет уменьшить время отклика при обслуживании каналов с высоким приоритетом.

Контроллер имеет 4 разряда, которые определяют количество транзакций по шине AHB до повторения арбитража. Эти разряды задают степень R числа 2; изменение R напрямую устанавливает периодичность арбитража как 2 в степени R. Для примера, если R равно 4, то арбитраж будет проводиться через каждые 16 передач DMA.
Таблица 2-7 показывает возможную периодичность арбитража.

Таблица 2-7. Периодичность арбитража в единицах передач по шине AHB.

Значение R	Периодичность арбитража каждые x передач DMA
4’b0000	1
4’b0001	2
4’b0010	4
4’b0011	8
4’b0100	16
4’b0101	32
4’b0110	64
4’b0111	128
4’b1000	256
4’b1001	512
4’b1010 – 4’b1111	1024

Примечание. Необходимо с осторожностью устанавливать большие значения R для низкоприоритетных каналов, так как это может привести к невозможности обслуживать запросы по высокоприоритетным каналам.

При $N > 2^R$ (N – номер передачи) и если результат деления 2^R на N не целое число, контроллер всегда выполняет последовательность из 2^R передач до тех пор, пока не станет верным $N<2^R$. Контроллер выполняет оставшихся N передач в конце цикла DMA.

Разряды степени R числа 2 находятся в структуре управляющих данных канала. Местонахождение этих разрядов описано в разделе «Управляющие данные канала».

Приоритет

При проведении арбитража определяется канал для обслуживания в следующем цикле DMA. На выбор следующего канала влияют:

- номер канала;
- уровень приоритета, присвоенного каналу.

Каждому каналу может быть присвоен уровень приоритета по умолчанию (низкий) или высокий уровень приоритета. Присвоение уровня приоритета осуществляется установкой или сбросом разряда chnl_priority_set.
Канал номер 0 имеет высший уровень приоритета, уровень приоритета снижается с увеличением номера канала. Таблица 2-8 показывает уровень приоритета каналов DMA в порядке его уменьшения.

Таблица 2-8. Уровень приоритета каналов DMA.

Номер канала	Установка уровня приоритета	Уровень приоритета в порядке его уменьшения
0	Высокий	Наивысший уровень приоритета
1	Высокий	-
2	Высокий	-
-	Высокий	-
-	Высокий	-
-	Высокий	-
30	Высокий	-
31	Высокий	-
0	По умолчанию (низкий)	-
1	По умолчанию (низкий)	-
2	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
30	По умолчанию (низкий)	-
31	По умолчанию (низкий)	Низший уровень приоритета

После окончания цикла DMA контроллер выбирает следующий для обслуживания канал из всех включенных каналов DMA. Рисунок 2-11 показывает процесс выбора следующего канала для обслуживания.

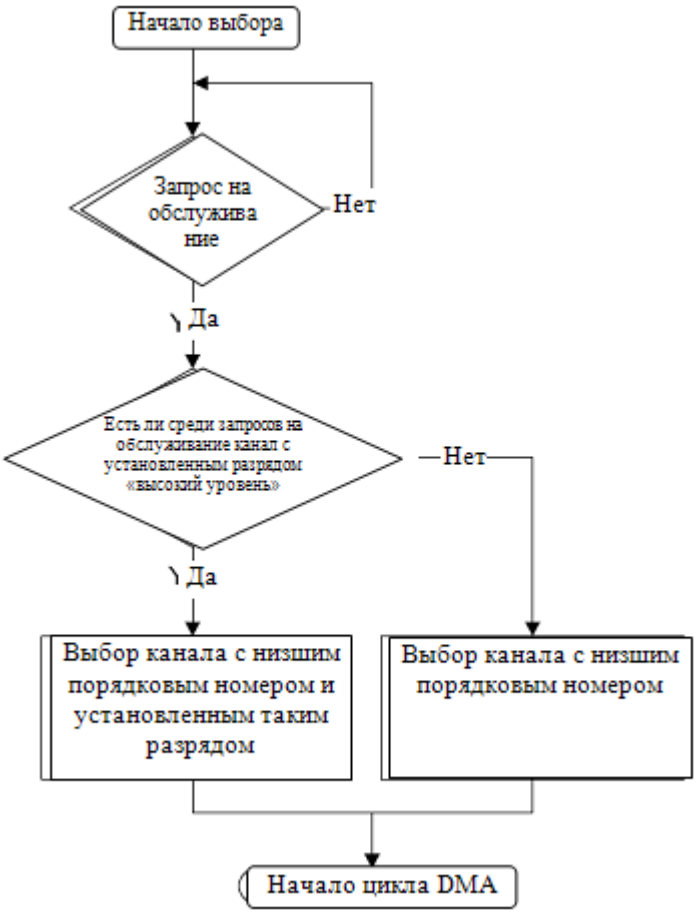


Рисунок 2-11. Алгоритм выбора следующего канала для обслуживания:
Начало выбора.
Есть ли запрос на обслуживание.
Есть ли среди запросов на обслуживание канал с установленным разрядом «высокий уровень».
Выбор канала с низшим порядковым номером и установленным таким разрядом.
Выбор канала с низшим порядковым номером.
Начало цикла DMA.

Типы циклов DMA

Разряды cycle_ctrl определяют, как контроллер будет выполнять циклы DMA. Описание значений этих разрядов приведено ниже.

Таблица 2-9. Типы циклов DMA.

cycle_ctrl	Описание
3'b000	Структура управляющих данных канала в запрещенном состоянии
3'b001	Обычный цикл DMA
3'b010	Авто-запрос
3'b011	Режим пинг-понг
3'b100	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
3'b101	Работа с памятью в режиме «Исполнение с изменением

	конфигурации» с использованием альтернативных управляющих данных канала
3’b110	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
3’b111	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала

Примечание. Разряды cycle_ctrl находятся в области памяти, отведенной под channel_cfg – см. раздел «Настройка управляющих данных канала».

Для всех типов циклов DMA повторный арбитраж происходит после 2^R передач DMA. Если установить длинный период арбитража на низкоприоритетном канале, то это заблокирует все запросы на обработку от других каналов до тех пор, пока не будут выполнены 2^R передач DMA по данному каналу. Поэтому, устанавливая значение R, необходимо учитывать, что это может привести к повышенному времени отклика на запрос на обработку от высокоприоритетных каналов.

Данный раздел описывает следующие типы циклов DMA:

- недействительный;
- основной;
- авто-запрос;
- «пинг-понг»;
- работа с памятью в режиме «исполнение с изменением конфигурации»;
- работа с периферией в режиме «исполнение с изменением конфигурации».

Недействительный

После окончания цикла DMA контроллер устанавливает тип цикла в значение «недействительный» для предотвращения повтора выполненного цикла DMA.

Основной

В этом режиме контроллер работает только с основными или альтернативными управляющими данными канала. После того, как разрешена работа канала и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

1. Контроллер выполняет 2^R передач. Если число оставшихся передач 0, контроллер переходит к шагу 3.
2. Осуществление арбитража:
   - если высокоприоритетный канал выдает запрос на обработку, то контроллер начинает обслуживание этого канала;
   - если периферийный блок или программное обеспечение выдает запрос на обработку (повторный запрос на обработку по каналу), то контроллер переходит к шагу 1.
3. Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала hclk. Это указывает центральному процессору на завершение цикла DMA.

Авто-запрос

Функционируя в данном режиме, контроллер ожидает получения одиночного запроса на обработку для разрешения работы и выполнения цикла DMA. Такая работа позволяет выполнять передачу больших пакетов данных без существенного увеличения времени отклика на обслуживание высокоприоритетных запросов и не требует множественных запросов на обработку от процессора или периферийных блоков.

Контроллер позволяет выбрать для использования первичную или альтернативную структуру управляющих данных канала. После того как разрешена работа канала и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

1. Контроллер выполняет 2^R передач для канала C. Если число оставшихся передач 0, контроллер переходит к шагу 3.
2. Осуществление арбитража:
   - если высокоприоритетный канал выдает запрос на обработку, то контроллер начинает обслуживание этого канала;
   - если периферийный блок или программное обеспечение выдает запрос на обработку (повторный запрос на обработку по каналу), то контроллер переходит к шагу 1.
3. Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала hclk. Это указывает центральному процессору на завершение цикла DMA.

Пинг-понг

В данном режиме контроллер выполняет цикл DMA, используя одну из структур управляющих данных, а затем выполняет еще один цикл DMA, используя другую структуру управляющих данных. Контроллер выполняет циклы DMA с переключением структур до тех пор, пока не считает «неправильную» структуру данных или пока процессор не запретит работу канала.
Рисунок 2-12 демонстрирует пример функционирования контроллера в режиме Пинг-понг.

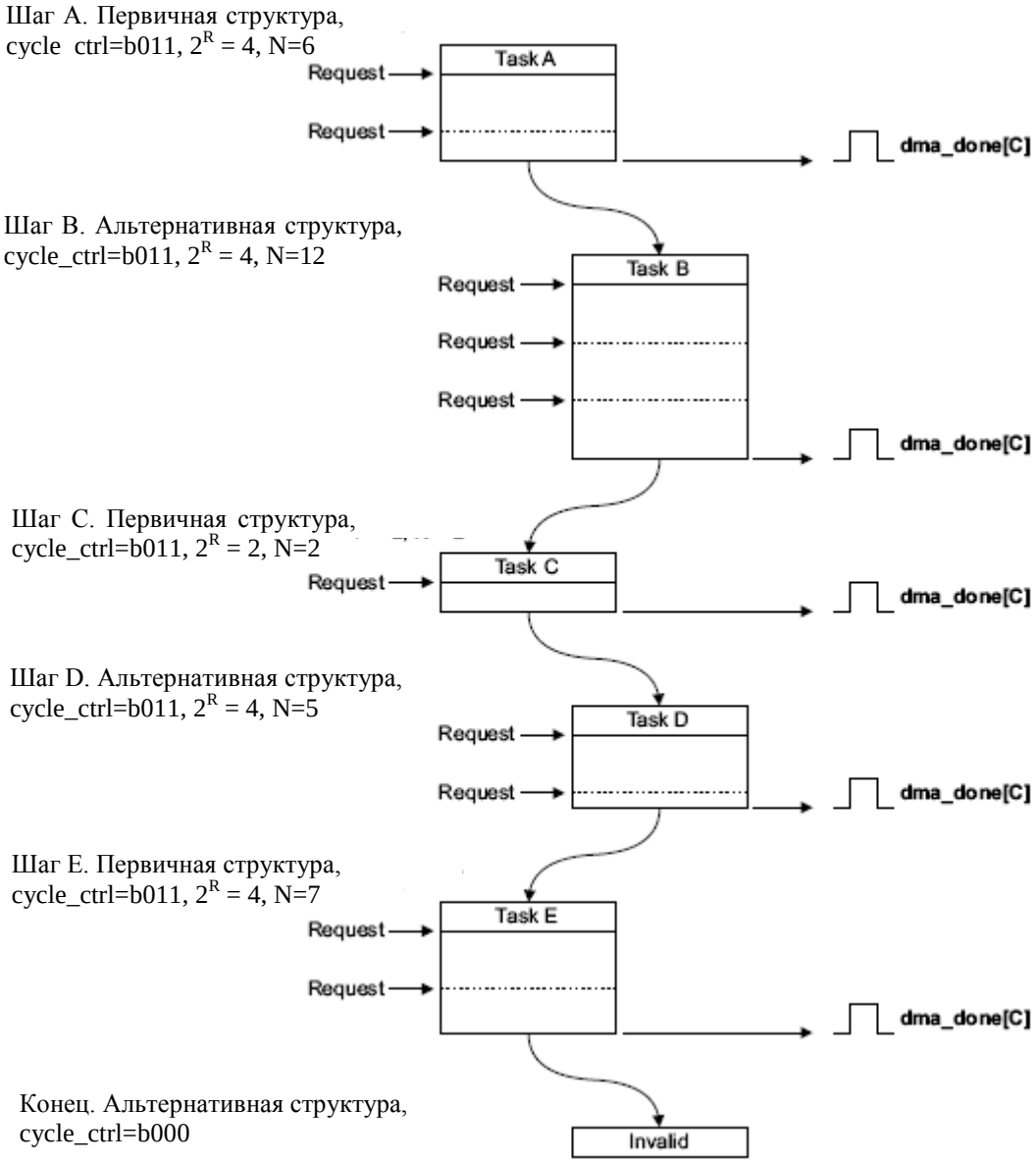


Рисунок 2-12. Пример работы в режиме Пинг-понг

Пояснения к рисунку 2-12:

Шаг A

Процессор устанавливает первичную структуру управляющих данных для шага A.

Процессор устанавливает альтернативную структуру управляющих данных для шага B. Это позволит контроллеру переключиться к шагу B незамедлительно после выполнения шага A, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.

Контроллер получает запрос и выполняет 4 передачи DMA.

Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.

Контроллер выполняет оставшиеся 2 передачи DMA. Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После выполнения шага A процессор может установить первичные управляющие данные канала для шага C. Это позволит контроллеру переключиться к шагу C незамедлительно после выполнения шага B, при условии что контроллер не получит запрос на обработку от высокоприоритетного канала.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг B:

Шаг B
Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 4 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

После выполнения шага B процессор может установить альтернативные управляющие данные канала для шага D.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг C:

Шаг C
Контроллер выполняет 2 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

После выполнения шага C процессор может установить первичные управляющие данные канала для шага E.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг D:

Шаг D
Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшуюся передачу DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг E:

Шаг E
Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 3 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража.

Если контроллер получит новый запрос на обработку от данного канала и этот запрос будет самым приоритетным, контроллер предпримет попытку выполнения следующего шага. Однако из-за того, что процессор не установил альтернативные управляющие данные, и по окончанию шага D контроллер установил cycle_ctrl в состояние b000, передачи DMA прекращаются.

Примечание. Для прерывания цикла DMA, исполняемого в режиме «пинг-понг», также возможен перевод режима работы контроллера на шаге E в режим «Основной цикл DMA» путем установки cycle_ctrl в 3’b001.

Режим работы с памятью «исполнение с изменением конфигурации»

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные. Затем контроллер выполняет еще 4 передачи DMA, используя первичные управляющие данные. Контроллер продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;
- контроллер считает «неправильную» структуру управляющих данных.

Примечание. После исполнения контроллером N передач с использованием первичных управляющих данных он делает эти управляющие данные «неправильными» путем установки cycle_ctrl в 3’b000.

Контроллер устанавливает флаг dma_done[C] в этом режиме работы только тогда, когда передача DMA заканчивается с использованием основного цикла.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 2-10 перечисляет области памяти channel_cfg, те которые должны быть определены константами, и те, значения которых определяются пользователем.

Таблица 2-10. Channel_cfg для первичной структуры управляющих данных в режиме работы с памятью «исполнение с изменением конфигурации».

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
[31:30]	dst_inc	2’b10	Контроллер производит инкремент адреса пословно
[29:28]	dst_size	2’b10	Контроллер осуществляет передачу пословно
[27:26]	src_inc	2’b10	Контроллер производит инкремент адреса пословно
[25:24]	src_size	2’b10	Контроллер осуществляет передачу пословно
[17:14]	R_power	4’b0010	Контроллер выполняет 4 передачи DMA
[3]	next_useburst	1’b0	Для данного режима этот разряд должен быть равен 0
[2:0]	cycle_ctrl	3’b100	Контролер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями определяемыми пользователем			
[23:21]	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
[20:18]	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
[13:4]	n_minus_1	N*	Настраивает контроллер на выполнение N передач DMA, где N кратно 4.

* - Так как разряды R_power установлены в состояние 2, необходимо задавать значение N кратное 4. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные.

Рисунок 2-13 демонстрирует пример функционирования в режиме работы с памятью «исполнение с изменением конфигурации».

Инициализация:

1. Настройка первичных управляющих данных для разрешения копирования A, B, C и D: cycle_ctrl=3'b100, 2^R=4, N=16.

2. Запись первичных данных в память с использованием структуры, показанной в таблице ниже.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	0x0A000000	0x0AE00000	cycle_ctrl = b101, 2^R = 4, N = 3	0xXXXXXXXX
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b101, 2^R = 2, N = 8	0xXXXXXXXX
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b101, 2^R = 8, N = 5	0xXXXXXXXX
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2^R = 4, N = 4	0xXXXXXXXX

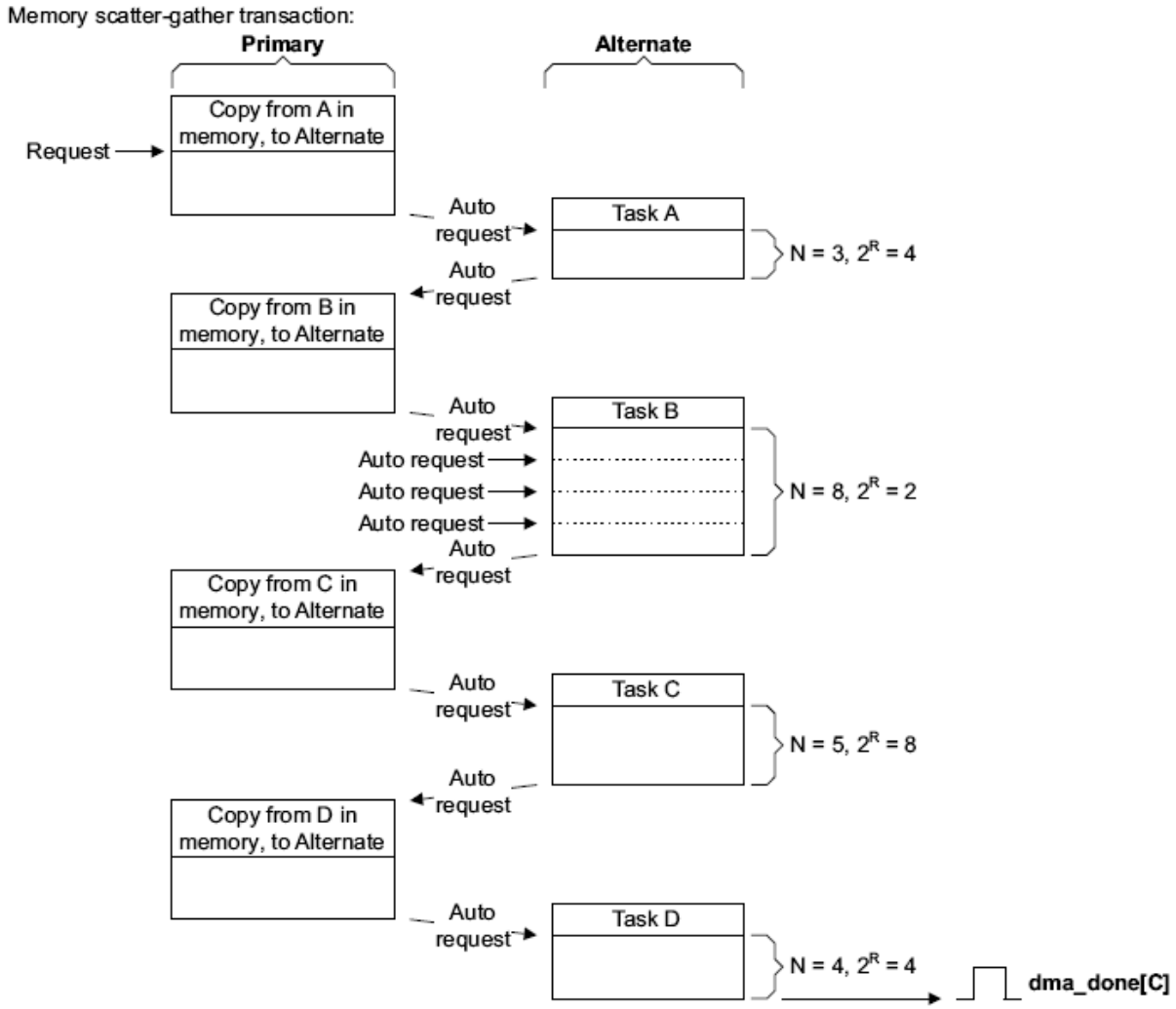


Рисунок 2-13. Пример функционирования контроллера в режиме работы с памятью «исполнение с изменением конфигурации».

Инициализация:

1. Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с памятью «исполнение с изменением конфигурации» путем установки cycle_ctrl в 3b'100. Так как управляющие данные канала состоят из 4 слов, необходимо установить 2^R в 4. В этом примере количество задач равно 4 и поэтому N установлен в 16.

2. Процессор записывает управляющие данные для шагов A, B, C, D в область памяти с адресом, указанным в src_data_end_ptr.

3. Процессор разрешает работу канала DMA.

Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по dma_req[] или запроса от процессора. Порядок выполнения следующий:

Первичная, копирование A

По получению запроса на обслуживание контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага A.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг A.

Контроллер выполняет шаг A. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование B

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага B.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг B.

Контроллер выполняет шаг B. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование C

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага C.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг C.

Контроллер выполняет шаг C. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование D

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D.

Контроллер устанавливает cycle_ctrl первичных управляющих данных в 3'b000 для индикации о том, что эта структура управляющих данных является «неправильной».

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг D.

Контроллер выполняет шаг D, используя основной цикл DMA.

Контроллер устанавливает флаг dma_done[C] в состояние 1 на один такт сигнала hclk и входит в процедуру арбитража.

Режим работы с периферией «исполнение с изменением конфигурации».

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал dma_active[C] в 0.

Примечание. Это единственный случай, при котором контроллер не осуществляет процедуру арбитража после выполнения передачи DMA, используя первичные управляющие данные.

После того, как этот цикл завершился, контроллер выполняет арбитраж и по получении запроса на обслуживание от периферии, имеющего наивысший приоритет, он выполняет еще 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал dma_active[C] в 0.

Контроллер продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;
- контроллер считает «неправильную» структуру управляющих данных.

Примечание. После исполнения контроллером N передач с использованием первичных управляющих данных, он делает эти управляющие данные «неправильными» путем установки cycle_ctrl в 3’b000.

Контроллер устанавливает флаг dma_done[C] в этом режиме работы только тогда, когда передача DMA заканчивается с использованием основного цикла.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 2-11 перечисляет области памяти channel_cfg, те которые должны быть определены константами, и те, значения которых определяются пользователем.

Таблица 2-11. Channel_cfg для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации».

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
[31:30]	dst_inc	2’b10	Контроллер производит инкремент адреса пословно
[29:28]	dst_size	2’b10	Контроллер осуществляет передачу пословно
[27:26]	src_inc	2’b10	Контроллер производит инкремент адреса пословно
[25:24]	src_size	2’b10	Контроллер осуществляет передачу пословно
[17:14]	R_power	4’b0010	Контроллер выполняет 4 передачи DMA
[2:0]	cycle_ctrl	3’b110	Контролер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями определяемыми пользователем			

[23:21]	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
[20:18]	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
[13:4]	n_minus_1	N*	Настраивает контроллер на выполнение N передач DMA, где N кратно 4.
[3]	next_useburst	-	При установке в 1, контроллер установит chnl_useburst_set[C] в 1 после выполнения передачи с альтернативной структурой.

* Так как разряды R_power установлены в состояние 2, необходимо задавать значение N кратное 4. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные.

Рисунок 2-14 демонстрирует пример функционирования в режиме работы с периферией «исполнение с изменением конфигурации».

Инициализация:

1. Настройка первичных управляющих данных для разрешения копирования A, B, C и D: cycle_ctrl=3'b110, 2^R=4, N=16.

2. Запись первичных данных в память с использованием структуры, показанной в таблице ниже.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	0x0A000000	0x0AE00000	cycle_ctrl = b111, 2^R = 4, N = 3	0xXXXXXXXX
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b111, 2^R = 2, N = 8	0xXXXXXXXX
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b111, 2^R = 8, N = 5	0xXXXXXXXX
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2^R = 4, N = 4	0xXXXXXXXX

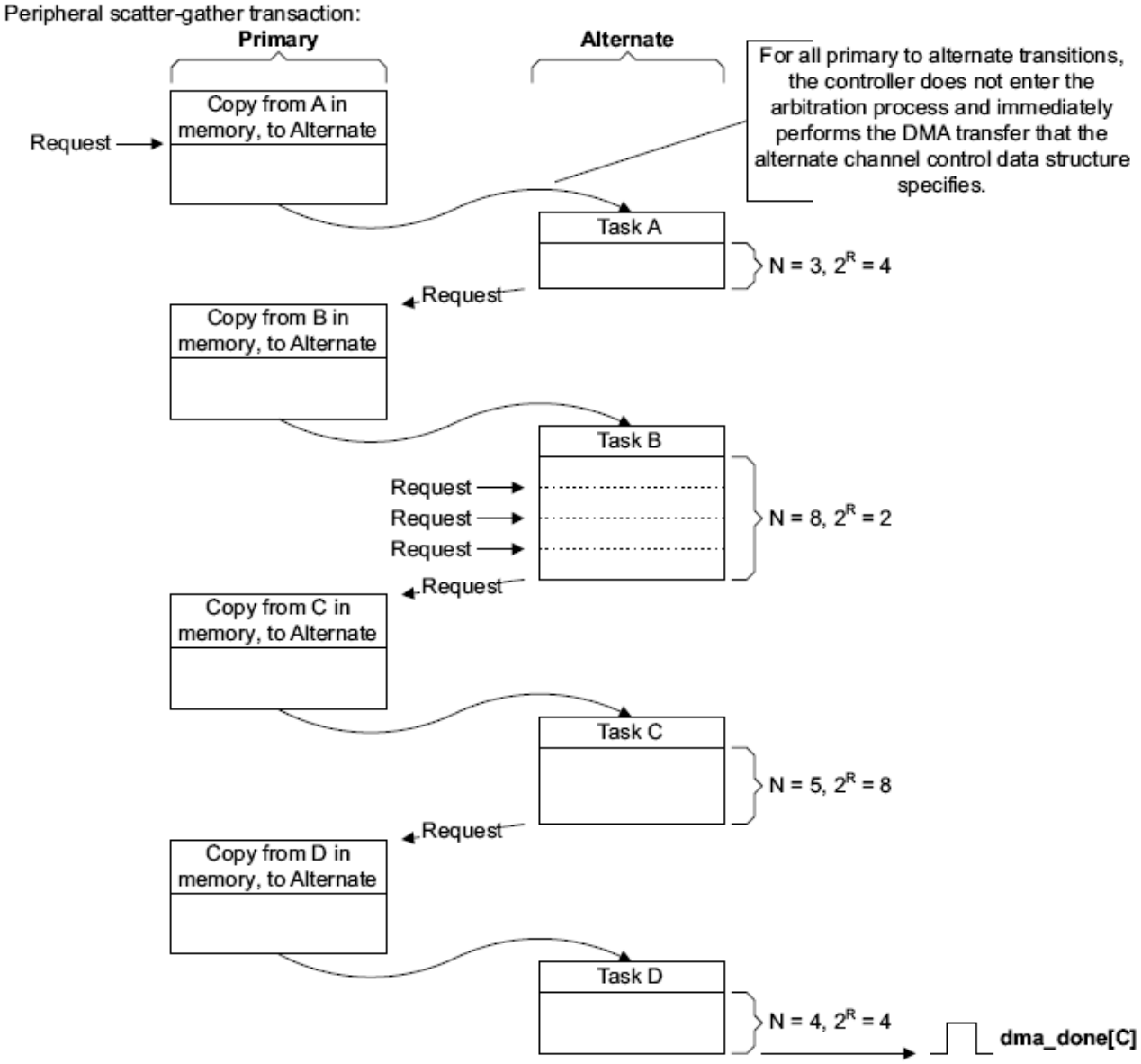


Рисунок 2-14. Пример функционирования контроллера в режиме работы с периферией «исполнение с изменением конфигурации».

Пояснения

Инициализация:

1. Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации» путем установки cycle_ctrl в 3'b110. Так как управляющие данные канала состоят из 4 слов, необходимо установить 2^R в 4. В этом примере количество задач равно 4 и поэтому N установлено в 16.

2. Процессор записывает управляющие данные для шагов A, B, C, D в область памяти с адресом, указанным в src_data_end_ptr.
3. Процессор разрешает работу канала DMA.

Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по dma_req[]. Передачи выполняются следующим образом:

Первичная, копирование из области A памяти

По получению запроса на обслуживание, контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага A.

Шаг A.

Контроллер выполняет шаг A.

По окончании контроллер проводит процедуру арбитража.

Первичная, копирование из области B памяти.

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага B.

Шаг B.

Контроллер выполняет шаг B. Для завершения задачи периферия должна установить последовательно 3 запроса.

По окончании контроллер проводит процедуру арбитража.

Первичная, копирование из области C памяти.

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага C.

Шаг C.

Контроллер выполняет шаг C.

По окончании контроллер проводит процедуру арбитража.

После выставления периферией нового запроса на обслуживание, при условии что этот запрос является наиболее приоритетным, процесс продолжается следующим образом:

Первичная, копирование из области D памяти.

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D.

Контроллер устанавливает cycle_ctrl первичных управляющих данных в 3'b000 для индикации о том, что эта структура управляющих данных является «неправильной».

Шаг D.

Контроллер выполняет шаг D, используя основной цикл DMA.

Контроллер устанавливает флаг dma_done[C] в состояние 1 на один такт сигнала hclk и входит в процедуру арбитража.

Индикация ошибок

При получении контроллером по шине AHB ответа об ошибке, он выполняет следующие действия:

- отключает канал, связанный с ошибкой;
- устанавливает флаг dma_err в состояние 1.

После обнаружения процессором флага dma_err процессор определяет номер канала, который был активен в момент появления ошибки. Для этого он осуществляет следующее:

- чтение регистра chnl_enable_set с целью создания списка отключенных каналов;
- если канал установил флаг dma_done[], то контроллер отключает канал. Программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно установили флаги dma_done[];
- процессор должен сравнить список выключенных каналов, полученный в шаге 1, с данными о каналах, которые недавно устанавливали флаги dma_done[]. Канал, по которому отсутствуют данные об установке флага dma_done[], это и есть канал, с которым связана ошибка.

Структура управляющих данных канала

В системной памяти должна быть отведена область для хранения управляющих данных каналов. Системная память должна:
- предоставлять смежную область системной памяти, к которой контроллер и процессор имеют доступ;
- иметь базовый адрес, который целочисленно кратен общему размеру структуры управляющих данных канала.

Рисунок 2-15 показывает область памяти необходимую контроллеру для структур управляющих данных канала, при использовании всех 32 каналов и опциональной альтернативной структуры управляющих данных.

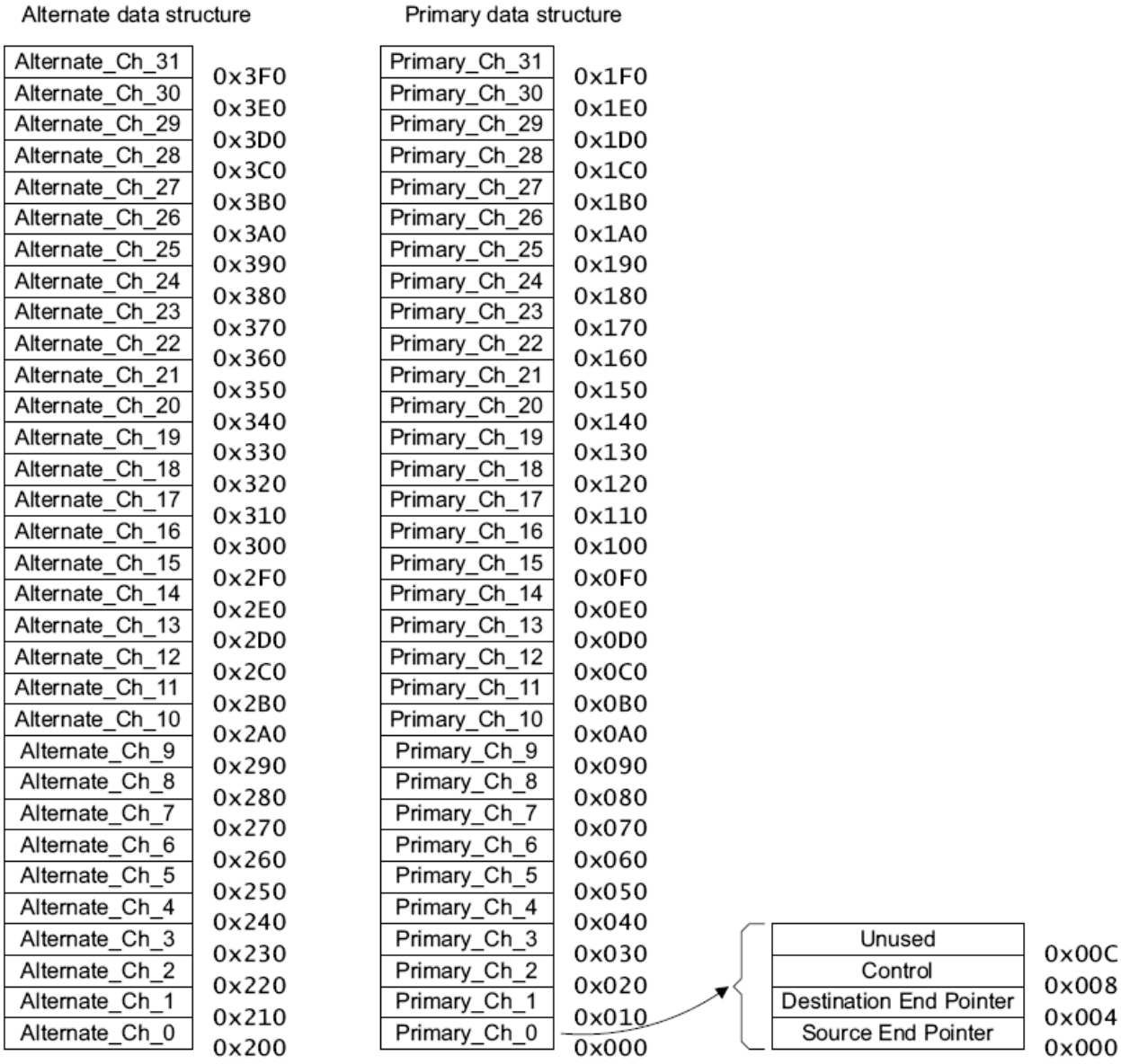


Рисунок 2-15. Карта памяти для 32-х каналов, включая альтернативную структуру управляющих данных.

Пример показанный на рисунке 2-15 использует 1 Кбайт системной памяти. В этом примере контроллер использует младшие 10 разрядов адреса для доступа ко всем

элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXX000, далее 0xXXXXX400, далее 0xXXXXX800, далее 0xXXXXXC00.
Базовый адрес для первичной структуры управляющих данных устанавливается путем записи соответствующего значения в регистр ctrl_base_ptr.
Необходимый размер области системной памяти зависит от:
- количества каналов используемых в контроллере;
- от того, используется или нет альтернативная структура управляющих данных.
Таблица 2-12 перечисляет разряды адреса, которые используются контроллером при доступе к различным элементам структуры управляющих данных, в зависимости от количества каналов, используемых в контроллере.

Таблица 2-12. Разряды адреса соответствующие элементам структуры управляющих данных.

Количество каналов, используемых в контроллере	[9]	[8]	[7]	[6]	[5]	[4]	[3:0]
1						A	0x0 0x4 0x8
2					A	C[0]	
3-4				A	C[1]	C[0]	
5-8			A	C[2]	C[1]	C[0]	
9-16		A	C[3]	C[2]	C[1]	C[0]	
17-32	A	C[4]	C[3]	C[2]	C[1]	C[0]	

, где

A — Выбирает одну из структур управляющих данных канала
A = 0 выбирает первичную структуру управляющих данных
A = 1 выбирает альтернативную структуру управляющих данных

C[x:0] — Выбирает канал DMA.

Address[3:0] — Выбирает один из управляющих элементов:
0x0 выбирает указатель конца данных источника;
0x4 выбирает указатель конца данных приемника;
0x8 выбирает конфигурацию управляющих данных;
0xC контроллер не имеет доступа к этому адресу.

Если это необходимо, то возможно разрешить процессору использовать эти адреса в качестве системной памяти.

Примечание. Совсем не обязательно вычислять базовый адрес альтернативной структуры управляющих данных, так как регистр alt_ctrl_base_ptr содержит эту информацию.

Рисунок 2-16 демонстрирует пример реализации контроллера с использованием 3 канлов DMA и альтернативной структурой управляющих данных.

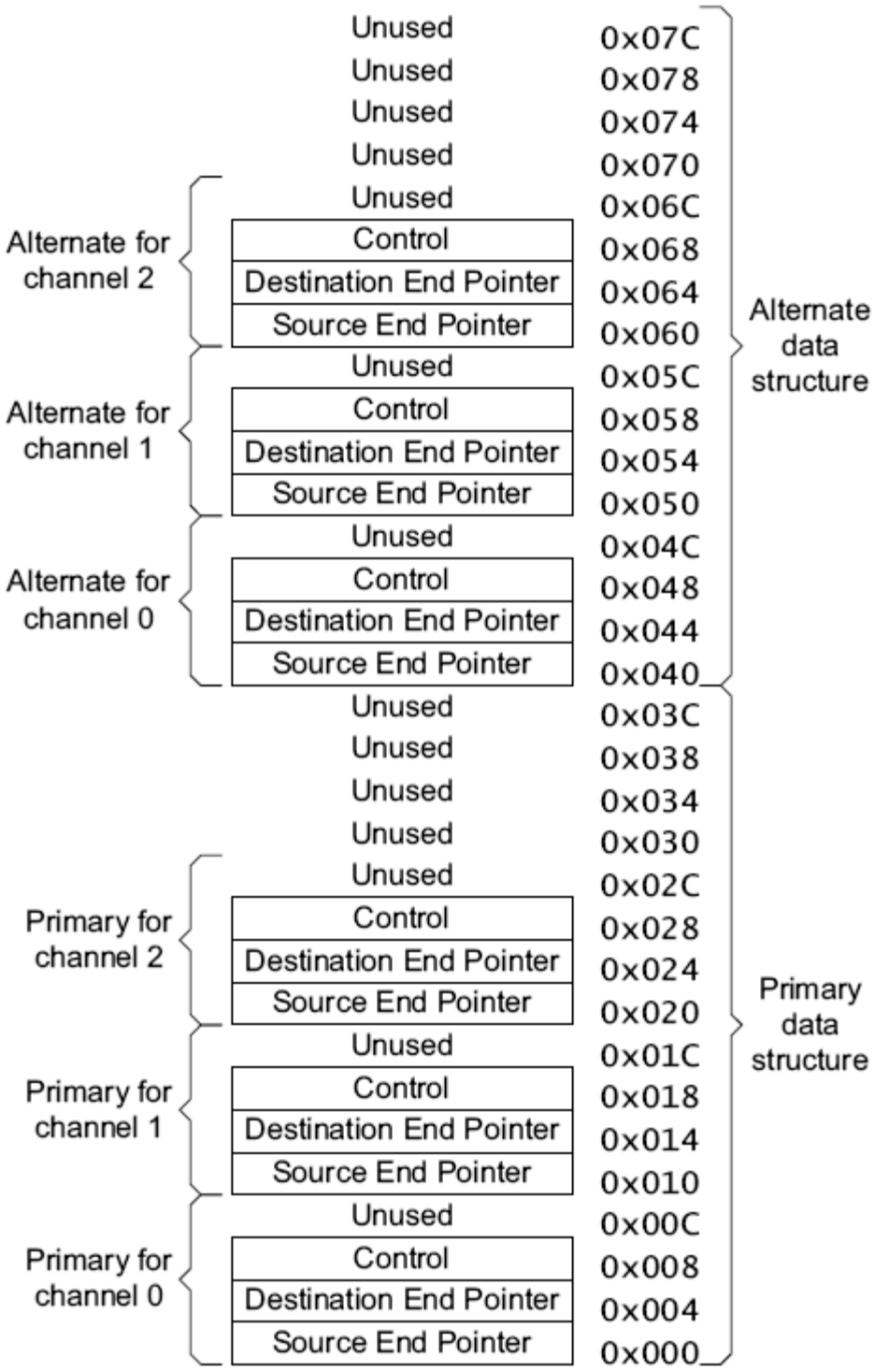


Рисунок 2-16. Карта памяти для трех каналов DMA, включая альтернативную структуру управляющих данных (где Destination end pointer - указатель конца данных приемника; Source end pointer - указатель конца данных источника; Control – управление).

Пример структуры управляющих данных на рисунке 2-16 использует 128 байт системной памяти. В этом примере контроллер использует младшие 6 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXXX00, далее 0xXXXXXX80.

Таблица 2-13 перечисляет все разрешенные значения базового адреса для первичной структуры управляющих данных, в зависимости от количества каналов DMA, использованных в контроллере.

Таблица 2-13. Разрешенные базовые адреса.

Кол-во каналов DMA	Разрешенные значения базового адреса для первичной структуры управляющих данных
1	0xXXXXXX00, 0xXXXXXX20, 0xXXXXXX40, 0xXXXXXX60, 0xXXXXXX80, 0xXXXXXXA0, 0xXXXXXXC0, 0xXXXXXXE0
2	0xXXXXXX00, 0xXXXXXX40, 0xXXXXXX80, 0xXXXXXXC0
3-4	0xXXXXXX00, 0xXXXXXX80
5-8	0xXXXXX000, 0xXXXXX100, 0xXXXXX200, 0xXXXXX300, 0xXXXXX400, 0xXXXXX500, 0xXXXXX600, 0xXXXXX700, 0xXXXXX800, 0xXXXXX900, 0xXXXXXA00, 0xXXXXXB00, 0xXXXXXC00, 0xXXXXXD00, 0xXXXXXE00, 0xXXXXXF00,
9-16	0xXXXXX000, 0xXXXXX200, 0xXXXXX400, 0xXXXXX600, 0xXXXXX800, 0xXXXXXA00, 0xXXXXXC00, 0xXXXXXE00
17-32	0xXXXXX000, 0xXXXXX400, 0xXXXXX800, 0xXXXXXC00

Контроллер использует системную память для доступа к двум указателям адреса конца данных и разрядам управления каждого канала. Следующие подразделы описывают эти 32-х разрядные области памяти и процедуру вычисления контроллером адреса передачи DMA:
- указатель конца данных источника;
- указатель конца данных приемника;
- разряды управления;
- вычисление адреса.

Указатель конца данных источника.

Область памяти под названием src_data_end_ptr содержит указатель на последний адрес месторасположения данных источника. Таблица 2-14 перечисляет значения разрядов этой области.

Таблица 2-14. Значения разрядов src_data_end_ptr.

Разряд	Имя	Описание
[31:0]	src_data_end_ptr	указатель на последний адрес данных источника

Перед тем как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.
Примечание. Контроллер не имеет доступа по записи в эту область памяти.

Указатель конца данных приемника.

Область памяти под названием dst_data_end_ptr содержит указатель на последний адрес месторасположения данных приемника. Таблица 2-15 перечисляет значения разрядов этой области.

Таблица 2-15. Значения разрядов dst_data_end_ptr.

Разряд	Имя	Описание
[31:0]	dst_data_end_ptr	указатель на последний адрес данных приемника

Перед тем как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.
Примечание. Контроллер не имеет доступа по записи в эту область памяти.

Разряды управления.

Область памяти под названием channel_cfg обеспечивает управление каждой передачей DMA. Рисунок 2-17 показывает название разрядов этой области.

Номер	31	30	29	28	27	26	25	24	23...21	20...18	17...14	13...4	3	2...0
Доступ														
Сброс														
	dst_inc		**dst_size**		**src_inc**		**src__size**		**dst_prot_crtl**	**Src_prot_ctrl**	**R_power**	**n_minus_1**	**next_useburst**	**cycle_ctrl**

Рисунок 2-17. Название разрядов channel_cfg.

Таблица 2-16 перечисляет назначение разрядов этой области памяти.

Таблица 2-16. Назначение разрядов channel_cfg.

Разряд	Имя	Описание
[31:30]	dst_src	Шаг инкремента адреса приемника Шаг инкремента адреса зависит от разрядности данных источника Разрядность данных источника = байт 2'b00 = байт; 2'b01 = полуслово (в русском обычно слово); 2'b10 = слово (в русском обычно двойное слово) 2'b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr. Разрядность данных источника = полуслово 2'b00 = зарезервировано; 2'b01 = полуслово; 2'b10 = слово; 2'b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr. Разрядность данных источника = слово 2'b00 = зарезервировано; 2'b01 = зарезервировано; 2'b10 = слово; 2'b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr.
[29:28]	dst_size	Размерность данных приемника

		Примечание. Значение этого поля должно быть равно значению поля src_size.
[27:26]	src_inc	Шаг инкремента адреса источника Шаг инкремента адреса зависит от разрядности данных источника Разрядность данных источника = байт 2'b00 = байт; 2'b01 = полуслово (в русском обычно слово); 2'b10 = слово (в русском обычно двойное слово); 2'b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr. Разрядность данных источника = полуслово 2'b00 = зарезервировано; 2'b01 = полуслово; 2'b10 = слово; 2'b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr. Разрядность данных источника = слово 2'b00 = зарезервировано; 2'b01 = зарезервировано; 2'b10 = слово; 2'b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr.
[25:24]	src_size	Задает размерность данных истояника 2'b00 = байт; 2'b01 = полуслово (в русском обычно слово); 2'b10 = слово (в русском обычно двойное слово); 2'b11 = зарезервировано.
[23:21]	dst_prot_ctrl	Задает состояние HPROT[3:1], когда контроллер записывает данные в приемник. Разряд [23] управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд [22] управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд [21] управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ не привилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный.
[20:18]	src_prot_ctrl	Задает состояние HPROT[3:1], когда контроллер считывает данные из источника. Разряд [20] управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд [19] управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд [18] управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ не

		привилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный.
[17:14]	R_power	Задает количество передач DMA до выполнения контроллером процедуры арбитража. Возможные значения: 4'b0000 - арбитраж производится после каждой передачи DMA; 4'b0001 – арбитраж производится после 2 передач DMA; 4'b0010 – арбитраж производится после 4 передач DMA; 4'b0011 – арбитраж производится после 8 передач DMA; 4'b0100 – арбитраж производится после 16 передач DMA; 4'b0101 – арбитраж производится после 32 передач DMA; 4'b0110 – арбитраж производится после 64 передач DMA; 4'b0111 – арбитраж производится после 128 передач DMA; 4'b1000 – арбитраж производится после 256 передач DMA; 4'b1001 – арбитраж производится после 512 передач DMA; 4'b1010 – 4'b1111 – арбитраж производится после 1024 передач DMA. Это означает, что арбитраж не производится, так как максимальное количество передач DMA равно 1024
[13:4]	n_minus_1	Перед выполнением цикла DMA эти разряды указывают общее количество передач DMA, из которых состоит цикл DMA. Необходимо установить эти разряды в значение, соответствующее размеру желаемого цикла DMA. 10-разрядное число плюс 1 задает количество передач DMA. Возможные значения: 10'b0000000000 = 1 передача DMA; 10'b0000000001 = 2 передачи DMA; 10'b0000000010 = 3 передачи DMA; 10'b0000000011 = 4 передачи DMA; 10'b0000000100 = 5 передач DMA; 10'b0000000101 = 6 передач DMA; …. 10'b1111111111 = 1024 передачи DMA. Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач DMA до завершения цикла DMA
[3]	next_useburst	Контролирует, не установлен ли chnl_useburst_set[C] в состояние 1, если контроллер работает в режиме работы с периферией «Исполнение с изменением конфигурации», и если контроллер завершает цикл DMA, используя альтернативные управляющие данные. *Примечание.* Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер устанавливает chnl_useburst_set[C] в значение 0, если количество оставшихся передач DMA меньше, чем 2^R. Установка next_useburst разряда определяет, будет ли контроллер дополнительно переопределять разряд chnl_useburst_set[C]. Если контроллер выполняет цикл DMA в режиме работы с

		периферией «Исполнение с изменением конфигурации», то после окончания цикла, использующего альтернативные управляющие данные, происходит следующее в зависимости от состояния next_useburst: 0 – контроллер не изменяет значение chnl_useburst_set[C]. Если chnl_useburst_set[C] установлен в 0, то для всех оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер отвечает на запросы по dma_req[] и dma_sreq[], при выполнении циклов DMA он использует альтернативные управляющие данные. 1 – контроллер изменяет значение chnl_useburst_set[C] в состояние 1. Поэтому для оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер реагирует только на запросы по dma_req[], при выполнении циклов DMA он использует альтернативные управляющие данные.
[2: 0]	cycle_ctrl	Режим работы при выполнении цикла DMA: 3'b000 **Стоп**. Означает, что структура управляющих данных является «неправильной»; 3'b001 **Основной**. Контроллер должен получить новый запрос для окончания цикла DMA, перед этим он должен выполнить процедуру арбитража; 3'b010 **Авто-запрос**. Контроллер автоматически осуществляет запрос на обработку по соответствующему каналу в течение процедуры арбитража. Это означает, что начального запроса на обработку достаточно для выполнения цикла DMA; 3'b011 **Пинг-понг**. Контроллер выполняет цикл DMA используя одну из структур управляющих данных. По окончании выполнения цикла DMA, контроллер; выполняет следующий цикл DMA, используя другую структуру. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя процессору перенастраивать неактивную структуру данных. Контроллер продолжает выполнять циклы DMA, до тех пор, пока он не прочитает «неправильную» структуру данных или пока процессор не изменит cycle_ctrl поле в состояние 3'b001 или 3'b010; 3'b100 Режим работы с памятью «Исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть 3'b100; 3'b101 Режим работы с памятью «Исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть 3'b101; 3'b110 Режим работы с периферией «исполнение с

		изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть 3'b110; 3'b111 Режим работы с периферией «исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть 3'b111.

В начале цикла DMA или 2^R передачи DMA контроллер считывает значение channel_cfg из системной памяти. После выполнения 2^R или N передач он сохраняет обновленное значение channel_cfg в системную память.

Контроллер не поддерживает значений dst_size, отличных от значений src_size. Если контроллер обнаруживает неравные значения этих полей, он использует значение src_size в качестве размера данных и приемника, и источника и при ближайшем обновлении поля n_minus_1, он также устанавливает значение поля dst_size, равное src_size.

После выполнения контроллером N передач, контроллер устанавливает значение поля cycle_ctrl в 3'b000, делая тем самым channel_cfg данные «неправильными». Это позволяет избежать повторения выполненной передачи DMA.

Вычисление адреса

Для вычисления адреса источника передачи DMA, контроллер выполняет сдвиг влево значения n_minus_1 на количество разрядов, соответствующее полю src_inc, и затем вычитает получившееся значение от значения указателя адреса конца данных источника. Подобным образом вычисляется адрес передатчика передачи DMA, контроллер выполняет сдвиг влево значения n_minus_1 на количество разрядов, соответствующее полю dst_inc, и затем вычитает получившееся значение от значения указателя адреса конца данных приемника.

В зависимости от значения полей src_inc и dst_inc вычисления адресов приемника и источника выполняются по следующим уравнениям:

```
src_inc=b00 and dst_inc=b00
 -  адрес источника = src_data_end_ptr - n_minus_1
 -  адрес приемника = dst_data_end_ptr - n_minus_1.

src_inc=b01 and dst_inc=b01
 -  адрес источника = src_data_end_ptr – (n_minus_1<<1)
 -  адрес приемника = dst_data_end_ptr – (n_minus_1<<1).

src_inc=b01 and dst_inc=b10
 -  адрес источника = src_data_end_ptr – (n_minus_1<<2)
 -  адрес приемника = dst_data_end_ptr – (n_minus_1<<2).

src_inc=b11 and dst_inc=b11
 -  адрес источника = src_data_end_ptr
 -  адрес приемника = dst_data_end_ptr.
```

Таблица 2-17 перечисляет адреса приемника цикла DMA для 6 слов.

Таблица 2-17. Цикла DMA для 6 слов с пословным инкрементом.

Начальные значения channel_cfg перед циклом DMA				
src_size=2'b10, dst_inc=2'b10, n_minus_1=3'b101, cycle_ctrl=1				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения channel_cfg после цикла DMA				
src_size=2'b10, dst_inc=2'b10, n_minus_1=0, cycle_ctrl=0				

* это значение, полученное после сдвига влево значения счетчика на количество разрядов соответствующее dst_inc.

Таблица 2-18 перечисляет адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.

Таблица 2-18. Цикла DMA для 12 байт с «полусловным» инкрементом.

Начальные значения channel_cfg перед циклом DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=4'b1011, cycle_ctrl=1, R_power=2'b11				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x5E7	11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3
	0x5E7	9	0x12	0x5D5
	0x 5E7	8	0x10	0x5D7
	0x 5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
	0x5E7	4	0x8	0x5DF
Значения channel_cfg после 2^R передач DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=3'b011, cycle_ctrl=1, R_power=2'b11				
DMA передачи	0x 5E7	3	0x6	0x5E1
	0x 5E7	2	0x4	0x5E3
	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7
Конечные значения channel_cfg после цикла DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=0, cycle_ctrl=0**, R_power=2'b11				

* это значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc.

** после окончания цикла DMA контроллер делает channel_cfg «неправильным», сбрасывая в 0 поле cycle_ctrl.

Описание регистров контроллера DMA

Данный раздел описывает регистры контроллера и управление контроллером через них.

Раздел содержит следующие сведения:

- о регистровой модели контроллера;
- описание регистров.

Основные положения регистровой модели контроллера:

- нужно избегать адресации при доступе к зарезервированным или неиспользованным адресам, так как это может привести к непредсказуемым результатам;
- необходимо заполнять неиспользуемые или зарезервированные разряды регистров нулями при записи и игнорировать значения таких разрядов при считывании, кроме случаев, специально описанных в разделе;
- системный сброс или сброс по установке питания сбрасывает все регистры в состояние 0, кроме случаев, специально описанных в разделе;
- все регистры поддерживают доступ по чтению и записи, кроме случаев, специально описанных в разделе. Доступ по записи обновляет содержание регистра, а доступ по чтению возвращает содержимое регистра.

Таблица 3-1. Перечень регистров контроллера.

Наименование	Смещение относительно базового адреса	Тип	Значение по сбросу	Описание
status	0x000	RO	0x-nn0000*	Статусный регистр DMA
cfg	0x004	WO	-	Регистр конфигурации DMA
ctrl_base_ptr	0x008	R/W	0x00000000	Регистр базового адреса управляющих данных каналов
alt_ctrl_base_ptr	0x00C	RO	0x000000nn**	Регистр базового адреса альтернативных управляющих данных каналов
waitonreq_status	0x010	RO	0x00000000	Регистр статуса ожидания запроса на обработку каналов
chnl_sw_request	0x014	WO	-	Регистр программного запроса на обработку каналов
chnl_useburst_set	0x018	R/W	0x00000000	Регистр установки пакетного обмена каналов
chnl_useburst_clr	0x01C	WO	-	Регистр сброса пакетного обмена каналов
chnl_req_mask_set	0x020	R/W	0x00000000	Регистр маскирования запросов на обслуживание каналов
chnl_req_mask_clr	0x024	WO	-	Регистр очистки маскирования запросов на обслуживание каналов
chnl_enable_set	0x028	R/W	0x00000000	Регистр установки разрешения каналов
chnl_enable_clr	0x02C	WO	-	Регистр сброса разрешения каналов

chnl_pri_alt_set	0x030	R/W	0x00000000	Регистр установки первичной/альтернативной структуры управляющих данных каналов
chnl_pri_alt_clr	0x034	WO	-	Регистр сброса первичной/альтернативной структуры управляющих данных каналов
chnl_priority_set	0x038	R/W	0x00000000	Регистр установки приоритета каналов
chnl_priority_clr	0x03C	WO	-	Регистр сброса приоритета каналов
-	0x040-0x048		-	зарезервировано
err_clr	0x04C	R/W	0x00000000	Регистр сброса флага ошибки
-	0x050-0xDFC	-		зарезервировано

* - значение по сбросу зависит от количества каналов DMA, использованных в контроллере, а также от наличия интегрированной схемы тестирования.
** - значение по сбросу зависит от количества каналов DMA, использованных в контроллере.
*** - значение зависит от номера версии контроллера.

STATUS

Статусный регистр DMA

Данный регистр имеет доступ только на чтение. При чтении регистр возвращает состояние контроллера. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 3-1 показывает наименование разрядов этого регистра. Таблица 3-2 перечисляет назначение разрядов регистра.

Номер	31...28	27...21	20...16	15...8	7...4	3...1	0
Доступ	RO	U	RO	U	RO	U	RO
Сброс	0	0	0	0	0	0	0
	test_status	**-**	**chnls_minus1**	**-**	**state**	**-**	**master_enable**

Рисунок 3-1. Наименование разрядов регистра status.

Таблица 3-2. Назначение разрядов регистра status.

№ бита	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...28	test_status	Значение при чтении: 4'b0000 = контроллер не имеет интегрированной схемы тестирования; 4'b0001 = контроллер имеет интегрированную схему тестирования; 4'b0010 – 4'b1111 = не определено
27...21	-	Не определено

20...16	chnls_minus1	Количество доступных каналов DMA минус 1. Например: 5'b00000 = контроллер имеет 1 канал DMA; 5'b00001 = контроллер имеет 2 канала DMA; 5'b00010 = контроллер имеет 3 канала DMA; … 5'b11111 = контроллер имеет 32 канала DMA
15...8	-	Не определено
7...4	state	Текущее состояние автомата управления контроллера. Состояние может быть одним из следующих: 4'b0000 = в покое; 4'b0001 = чтение управляющих данных канала; 4'b0010 = чтение указателя конца данных источника; 4'b0011 = чтение указателя конца данных приемника; 4'b0100 = чтение данных источника; 4'b0101 = запись данных в приемник; 4'b0110 = ожидание запроса на выполнение DMA; 4'b0111 = запись управляющих данных канала; 4'b1000 = приостановлен; 4'b1001 = выполнен; 4'b1010 = режим работы с периферией «Исполнение с изменением конфигурации»; 4'b1011 – 4'b1111 = не определено
3...1	-	Не определено
0	master_enable	Состояние контроллера: 0 = работа контроллера запрещена; 1 = работа контроллера разрешена

CFG

Регистр конфигурации DMA

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера. Рисунок 3-2 показывает наименование разрядов этого регистра. Таблица 3-3 перечисляет назначение разрядов регистра.

Номер	31...8	7...5	4...1	0
Доступ	U	WO	U	WO
Сброс	0	0	0	0
	-	**chnl_prot_ctrl**	**-**	**master_enable**

Рисунок 3-2. Наименование разрядов регистра cfg.

Таблица 3-3. Назначение разрядов регистра cfg.

№ бита	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...8	-	Не определено, следует записывать 0.
7...5	chnl_prot_ctrl	Определяет уровни индикации сигналов HPROT[3:1] защиты шины AHB-Lite:

		Разряд 7 управляет сигналом HPROT[3], с целью индикации о появлении доступа с кэшированием; Разряд 6 управляет сигналом HPROT[2], с целью индикации о появлении доступа с буферизацией; Разряд 5 управляет сигналом HPROT[1], с целью индикации о появлении привилегированного доступа. *Примечание.* Если разряд[n] = 1, то соответствующий сигнал HPROT в состоянии 1. Если разряд[n] = 0, то соответствующий сигнал HPROT в состоянии 0.
4...1	-	Не определено. Следует записывать 0.
0	master_enable	Определяет состояние контроллера: 0 – запрещает работу контроллера; 1 – разрешает работу контроллера.

CTRL_BASE_PTR

Регистр базового адреса управляющих данных каналов

Данный регистр имеет доступ на запись и чтение. Регистр определяет базовый адрес системной памяти размещения управляющих данных каналов.

Примечание. Контроллер не содержит внутреннюю память для хранения управляющих данных каналов.

Размер системной памяти, предназначенной контроллеру, зависит от количества каналов DMA, использующихся контроллером, а также от возможности использования альтернативных управляющих данных каналов. Поэтому количество разрядов регистра, необходимых для задания базового адреса, варьируется и зависит от варианта построения системы.

Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Таблица 3-4 перечисляет назначение разрядов регистра.

Номер	31...10	9...0
Доступ	R/W	U
Сброс	0	0
	ctrl_base_ptr	-

Рисунок 3-3. Наименование разрядов регистра ctrl_base_ptr.

Таблица 3-4. Назначение разрядов регистра ctrl_base_ptr.

№ бита	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…10	ctrl_base_ptr	Указатель на базовый адрес первичной структуры управляющих данных. См. соответствующий раздел
9…0	-	Не определено. Следует записывать 0

ALT_CTRL_BASE_PTR

Регистр базового адреса альтернативных управляющих данных каналов

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении указатель базового адреса альтернативных управляющих данных каналов. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Этот регистр позволяет не производить вычисления базового адреса альтернативных управляющих данных каналов.
Таблица 3-5 перечисляет назначение разрядов регистра.

Номер	31... 0
Доступ	RO
Сброс	0
	Alt_ctrl_base_ptr

Рисунок 3-4. Наименование разрядов регистра alt_ctrl_base_ptr.

Таблица 3-5. Назначение разрядов регистра alt_ctrl_base_ptr.

№ бита	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…0	alt_ctrl_base_ptr	Указатель базового адреса альтернативной структуры управляющих данных

WAITONREQ_STATUS

Регистр статуса ожидания запроса на обработку каналов

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении состояние сигналов dma_waitonreq[]. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.
Таблица 3-6 перечисляет назначение разрядов регистра.

Номер	31	**.....**	2	1	0
Номер	RO	**.....**	RO	RO	RO
Доступ	0	**.....**	0	0	0
	dma_waitonreq_status for dma_waitnreg [31]	**.....**	**dma_waitonreq_status for dma_waitnreg [2]**	**dma_waitonreq_status for dma_waitnreg [1]**	**dma_waitonreq_status for dma_waitnreg [0]**

Рисунок 3-5. Наименование разрядов регистра dma_waitonreq_status.

Таблица 3-6. Назначение разрядов регистра dma_waitonreq_status.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	dma_waitonreq_status	Состояние сигналов ожидания запроса на обработку каналов DMA. **При чтении:** Разряд [C] = 0 означает, что dma_waitonreq[C] в состоянии 0; Разряд [C] = 1 означает, что dma_waitonreq[C] в состоянии 1.

CHNL_SW_REQUEST

Регистр программного запроса на обработку каналов

Данный регистр имеет доступ только на запись. Регистр позволяет устанавливать программно запрос на выполнение цикла DMA.
Рисунок 3-6 показывает наименование разрядов этого регистра. Таблица 3-7 перечисляет назначение разрядов регистра.

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_sw_request for channel [31]	.....	**chnl_sw_request for channel [2]**	**chnl_sw_request for channel [1]**	**chnl_sw_request for channel [0]**

Рисунок 3-6. Наименование разрядов регистра chnl_sw_request.

Таблица 3-7. Назначение разрядов регистра chnl_sw_request.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_sw_request	Устанавливает соответствующий разряд для генерации программного запроса на выполнение цикла DMA по соответствующему каналу DMA. **При записи:** Разряд [C] = 0 означает, что запрос на выполнение цикла DMA по каналу C не будет установлен; Разряд [C] =1 означает, что запрос на выполнение цикла DMA по каналу C будет установлен. Запись разряда соответствующего нереализованному каналу, означает, что запрос на выполнение цикла DMA не будет установлен.

CHNL_USEBURST_SET

Регистр установки пакетного обмена каналов

Данный регистр имеет доступ на чтение и запись. Регистр отключает выполнение одиночных запросов по установке dma_sreq[] и поэтому будут обрабатываться и исполняться только запросы по dma_req[]. Регистр возвращает при чтении состояние установок пакетного обмена

Рисунок 3-7 показывает наименование разрядов этого регистра. Таблица 3-8 перечисляет назначение разрядов регистра.

Номер	31	.....	2	1	0
Доступ	R/W	.....	R/W	R/W	R/W
Сброс	0	.....	0	0	0
	chnl_useburst_set for channel [31]	.....	**chnl_useburst_set for channel [2]**	**chnl_useburst_set for channel [1]**	**chnl_useburst_set for channel [0]**

Рисунок 3-7. Наименование разрядов регистра chnl_useburst_set.

Таблица 3-8. Назначение разрядов регистра chnl_useburst_set.

№ бита	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…0	chnl_useburst_set	Отключает обработку запросов на выполнение циклов DMA от dma_sreq[] и возвращает при чтении состоянии этих настроек. **При чтении:** Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные от dma_sreq[] и dma_req[]. Контроллер выполняет одиночные передачи или 2^R передач. Разряд [C] = 1 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные только от dma_req[]. Контроллер выполняет 2^R передач. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_clr регистр и установить соответствующий разряд C в 0; Разряд [C] = 1 отключает возможность обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Контроллер выполняет 2^R передач. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

После выполнения предпоследней передачи из 2^R передач, в том случае, если число оставшихся передач (N) меньше чем 2^R, контроллер сбрасывает разряд chnl_useburst_set в 0. Это позволяет выполнять оставшиеся передачи, используя dma_sreq[] и dma_req[].

Примечание. При программировании channel_cfg значением N меньшим, чем 2^R, запрещена установка соответствующего разряда chnl_useburst_set в случае, если периферийный блок не поддерживает сигнал dma_req[].

В режиме работы с периферией «исполнение с изменением конфигурации», если разряд next_useburst установлен в channel_cfg, то контроллер устанавливает chnl_useburst_set [C] в 1 после окончания цикла DMA, использующего альтернативные управляющие данные.

CHNL_USEBURST_CLR

Регистр сброса пакетного обмена каналов

Данный регистр имеет доступ только на запись. Регистр разрешает выполнение одиночных запросов по установке dma_sreq[]. Рисунок 3-8 показывает наименование разрядов этого регистра. Таблица 3-9 перечисляет назначение разрядов регистра.

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_useburst_clr for channel [31]	.....	**chnl_useburst_clr for channel [2]**	**chnl_useburst_clr for channel [1]**	**chnl_useburst_clr for channel [0]**

Рисунок 3-8. Наименование разрядов регистра chnl_useburst_clr.

Таблица 3-9. Назначение разрядов регистра chnl_useburst_clr.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_useburst_clr	Установка соответствующего разряда разрешает обработку запросов на выполнение циклов DMA от dma_sreq[]. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_set регистр для отключения обработки запросов от dma_sreq[]; Разряд [C] = 1 разрешает обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_REQ_MASK_SET

Регистр маскирования запросов на обслуживание каналов

Данный регистр имеет доступ на чтение и запись. Регистр отключает установку запросов на выполнение циклов DMA на dma_sreq[] и dma_req[]. Регистр возвращает при чтении

состояние установок маскирования запросов от dma_sreq[] и dma_req[] на обслуживание каналов. Рисунок 3-9 показывает наименование разрядов этого регистра. Таблица 3-10 перечисляет назначение разрядов регистра.

Номер	31	**.....**	2	1	0
Доступ	R/W	**.....**	R/W	R/W	R/W
Сброс	0	**.....**	0	0	0
	chnl_reg_mask_set for dma_reg [31] and dma_sreg [31]	**.....**	**chnl_reg_mask_set for dma_reg [2] and dma_sreg [2]**	**chnl_reg_mask_set for dma_reg [1] and dma_sreg [1]**	**chnl_reg_mask_set for dma_reg [0] and dma_sreg [0]**

Рисунок 3-9. Наименование разрядов регистра chnl_req_mask_set.

Таблица 3-10. Назначение разрядов регистра chnl_req_mask_set.

№ бита	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…0	chnl_req_mask_set	Отключает обработку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов и возвращает при чтении состоянии этих настроек. **При чтении:** Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на поступающие запросы; Разряд [C] = 1 означает, что канал DMA C не выполняет циклы DMA в ответ на поступающие запросы. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_req_mask_clr регистр для разрешения установки запросов; Разряд [C] = 1 отключает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_REQ_MASK_CLR

Регистр очистки маскирования запросов на обслуживание каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает установку запросов на выполнение циклов DMA на dma_sreq[] и dma_req[].
Рисунок 3-10 показывает наименование разрядов этого регистра. Таблица 3-11 перечисляет назначение разрядов регистра.

Номер	31	**.....**	2	1	0
Доступ	WO	**.....**	WO	WO	WO
Сброс	0	**.....**	0	0	0
	chnl_reg_mask_clr for dma_reg [31] and dma_sreg [31]	**.....**	**chnl_reg_mask_clr for dma_reg [2] and dma_sreg [2]**	**chnl_reg_mask_clr for dma_reg [1] and dma_sreg [1]**	**chnl_reg_mask_clr for dma_reg [0] and dma_sreg [0]**

Рисунок 3-10. Наименование разрядов регистра chnl_req_mask_clr.

Таблица 3-11. Назначение разрядов регистра chnl_req_mask_clr.

№ бита	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…0	chnl_req_mask_clr	Установка соответствующего разряда разрешает установку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов. **При записи**: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_req_mask_set регистр для отключения установки запросов; Разряд [C] = 1 разрешает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_ENABLE_SET

Регистр установки разрешения каналов

Данный регистр имеет доступ на чтение и запись. Регистр разрешает работу каналов DMA. Регистр возвращает при чтении состояние разрешений работы каналов DMA.
Рисунок 3-11 показывает наименование разрядов этого регистра. Таблица 3-12 перечисляет назначение разрядов регистра.

Номер	31	**.....**	2	1	0
Доступ	R/W	**.....**	R/W	R/W	R/W
Сброс	0	**.....**	0	0	0

	chnl_enable_set for channel [31]	.....	chnl_enable_set for channel [2]	chnl_enable_set for channel [1]	chnl_enable_set for channel [0]

Рисунок 3-11. Наименование разрядов регистра chnl_enable_set.

Таблица 3-12. Назначение разрядов регистра chnl_enable_set.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_enable_set	Разрешает работу каналов DMA и возвращает при чтении состоянии этих настроек. **При чтении:** Разряд [C] = 0 означает, что канал DMA C отключен; Разряд [C] = 1 означает, что работа канала DMA C разрешена. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_enable_clr регистр для отключения канала; Разряд [C] = 1 разрешает работу канала DMA C. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

CHNL_ENABLE_CLR

Регистр сброса разрешения каналов

Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов DMA. Рисунок 3-12 показывает наименование разрядов этого регистра. Таблица 3-13 перечисляет назначение разрядов регистра.

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_enable_clr for channel 31	.....	chnl_enable_clr for channel 2	chnl_enable_clr for channel 1	chnl_enable_clr for channel 0

Рисунок 3-12. Наименование разрядов регистра chnl_enable_clr.

Таблица 3-13. Назначение разрядов регистра chnl_enable_clr.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_enable_clr	Установка соответствующего разряда запрещает работу соответствующего канала DMA. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_enable_set регистр для разрешения работы канала; Разряд [C] = 1 запрещает работу канала DMA C. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта. *Примечание.* Контроллер может отключить канал DMA, установив соответствующий разряд в следующих случаях: - при завершении цикла DMA; - при чтении из channel_cfg с полем cycle_ctrl установленным в 3'b000; - при появлении ошибки на шине AHB-Lite.

CHNL_PRI_ALT_SET

Регистр установки первичной/альтернативной структуры управляющих данных каналов

Данный регистр имеет доступ на запись и чтение. Регистр разрешает работу канала DMA с использованием альтернативной структуры управляющих данных. Чтение регистра возвращает состояние каналов DMA (какую структуру управляющих данных использует каждый канал DMA).
Рисунок 3-13 показывает наименование разрядов этого регистра. Таблица 3-14 перечисляет назначение разрядов регистра.

Номер	31	.....	2	1	0
Доступ	R/W	.....	R/W	R/W	R/W
Сброс	0	.....	0	0	0
	chnl_pri_alt_set for channel [31]	.....	**chnl_pri_alt_set for channel [2]**	**chnl_pri_alt_set for channel [1]**	**chnl_pri_alt_set for channel [0]**

Рисунок 3-13. Наименование разрядов регистра chnl_pri_alt_set.

Таблица 3-14. Назначение разрядов регистра chnl_pri_alt_set.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_pri_alt_set	Установка соответствующего разряда подключает использование альтернативных управляющих данных для соответствующего канала DMA, чтение возвращает состояние этих настроек. **При чтении:** Разряд [C] = 0 означает, что канал DMA C использует

		первичную структуру управляющих данных; Разряд [C] = 1 означает, что канал DMA C использует альтернативную структуру управляющих данных. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_clr регистр для сброса разряда [C] в 0; Разряд [C] = 1 подключает использование альтернативной структуры управляющих данных каналом DMA C. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта. *Примечание.* Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях: - при завершении 4-х передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; - при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «Пинг-понг»; - при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: - «пинг-понг»; - работа с памятью «Исполнение с изменением конфигурации»; - работа с периферией «Исполнение с изменением конфигурации;

CHNL_PRI_ALT_CLR

Регистр сброса первичной/альтернативной структуры управляющих данных каналов

Данный регистр имеет доступ только на запись. Регистр разрешает работу канала DMA с использованием первичной структуры управляющих данных.
Рисунок 3-14 показывает наименование разрядов этого регистра. Таблица 3-15 перечисляет назначение разрядов регистра.

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_pri_alt_clr for channel [31]	.....	**chnl_pri_alt_clr for channel [2]**	**chnl_pri_alt_clr for channel [1]**	**chnl_pri_alt_clr for channel [0]**

Рисунок 3-14. Наименование разрядов регистра chnl_pri_alt_clr.

Таблица 3-15. Назначение разрядов регистра chnl_pri_alt_clr.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_pri_alt_clr	Установка соответствующего разряда подключает использование первичных управляющих данных для соответствующего канала DMA. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_set регистр для выбора альтернативных управляющих данных; Разряд [C] = 1 подключает использование первичной структуры управляющих данных каналом DMA C. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта. *Примечание:* Контроллер может переключить значение разряда chnl_pri_alt_clr[C] в следующих случаях: - при завершении 4-х передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; - при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «пинг-понг»; - при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: - «пинг-понг» - работа с памятью «Исполнение с изменением конфигурации» - работа с периферией «Исполнение с изменением конфигурации»

CHNL_PRIORITY_SET

Регистр установки приоритета каналов

Данный регистр имеет доступ на запись и чтение. Регистр позволяет присвоить высокий приоритет каналу DMA. Чтение регистра возвращает состояние приоритета каналов DMA. Рисунок 3-15 показывает наименование разрядов этого регистра. Таблица 3-16 перечисляет назначение разрядов регистра.

Номер	31	.....	2	1	0
Доступ	R/W	.....	R/W	R/W	R/W
Сброс	0	.....	0	0	0

	chnl_priorit_set for channel [31]	.....	chnl_priority_set for channel [2]	chnl_priority_set for channel [1]	chnl_priority_set for channel [0]

Рисунок 3-15. Наименование разрядов регистра chnl_priority_set.

Таблица 3-16. Назначение разрядов регистра chnl_priority_set.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_priority_set	Установка высокого приоритета каналу DMA, чтение возвращает состояние приоритета каналов DMA. **При чтении:** Разряд [C] = 0 означает, что каналу DMA C присвоен уровень приоритета по умолчанию; Разряд [C] = 1 означает, что каналу DMA C присвоен высокий уровень приоритета. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_clr регистр для установки каналу C уровня приоритета по умолчанию; Разряд [C] = 1 устанавливает каналу DMA C высокий уровень приоритета. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта;

CHNL_PRIORITY_CLR

Регистр сброса приоритета каналов

Данный регистр имеет доступ только на запись. Регистр позволяет присвоить каналу DMA уровень приоритета по умолчанию.
Рисунок 3-16 показывает наименование разрядов этого регистра. Таблица 3-17 перечисляет назначение разрядов регистра.

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_priorit_clr for channel [31]	.....	chnl_priority_clr for channel [2]	chnl_priority_clr for channel [1]	chnl_priority_clr for channel [0]

Рисунок 3-16. Наименование разрядов регистра chnl_priority_clr.

Таблица 3-17. Назначение разрядов регистра chnl_priority_clr.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
[31:0]	chnl_priority_clr	Установка разряда присваивает соответствующему каналу DMA уровень приоритета по умолчанию. **При записи:** Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_set регистр для установки каналу C высокого уровня приоритета. Разряд [C] = 1 устанавливает каналу DMA C уровень приоритета по умолчанию. Запись разряда соответствующего нереализованному каналу не дает никакого эффекта.

ERR_CLR

Регистр сброса флага ошибки

Данный регистр имеет доступ на запись и чтение. Регистр позволяет сбрасывать сигнал dma_err в 0. Чтение регистра возвращает состояние сигнала dma_err.
Рисунок 3-17 показывает наименование разрядов этого регистра. Таблица 3-18 перечисляет назначение разрядов регистра.

Номер	31…1	0
Доступ	U	R/W
Сброс	0	0
	-	**err_clr**

Рисунок 3-17. Наименование разрядов регистра err_clr.

Таблица 3-18. Назначение разрядов регистра err_clr.

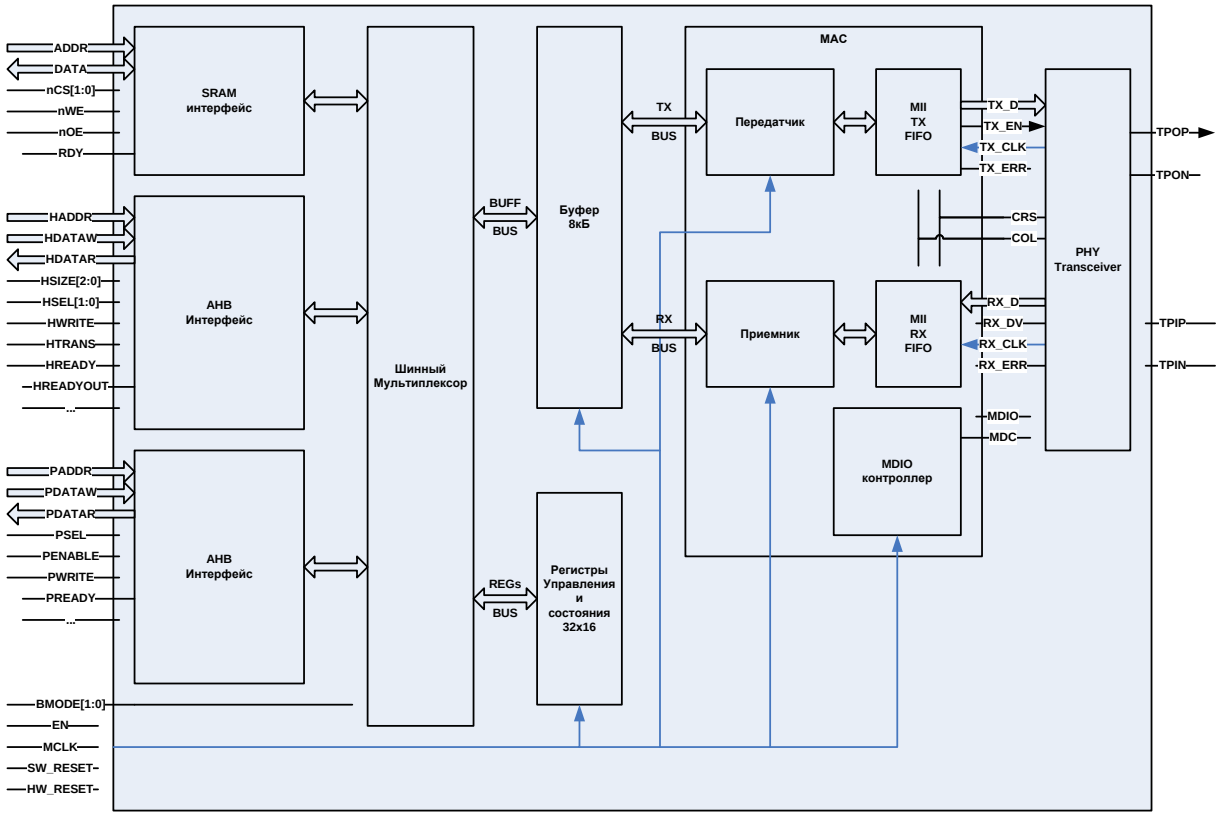
№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…1	-	Не определено. Следует записывать 0
0	chnl_priority_set	Установка сигнала в состояние 0, чтение возвращает состояние сигнала (флага) dma_err. **При чтении**: Разряд [C] = 0 означает, что dma_err находится в состоянии 0; Разряд [C] = 1 означает, что dma_err находится в состоянии 1. **При записи:** Разряд [C] =0 не дает эффекта. Состояние dma_err останется неизменным; Разряд [C] =1 сбрасывает сигнал (флаг) dma_err в состояние 0.

		Для целей тестирования возможно использовать регистр err_set, чтобы установить сигнал dma_err в состояние 1. *Примечание.* При сбросе сигнала dma_err одновременно с появлением ошибки на шине AHB-Lite, то приоритет отдается ошибке и следовательно, значение регистра (и dma_err) останется неизменным (несброшенным).

Контроллер интерфейса Ethernet.

При работе блок MAC выполняет 2 основные функции

- формирование пакета уровня звена данных протоколов Ethernet/IEEE802.3 и передача его на физический уровень;
- прием с физического уровня и разбор пакета уровня звена данных протоколов Ethernet/IEEE802.3.



Работа блока возможна в полно- и полудуплексном режимах. Переключение режима осуществляется битом G_CFG.HD_EN («1» - полудуплексный режим работы).

Также в общее управление блоком входит разрешение обработки пакета PAUSE (G_CFG.PAUSE_EN), управление размером окна распознаваний коллизий (G_CFG.ColWnd). И режимом работы буферов приемника и передатчика (G_CFG.BUFF_MODE). Режимы работы буферов линейный (BUFF_MODE=00b), автоматический (BUFF_MODE=01b) и режим FIFO (BUFF_MODE=10b). Режимы различаются способом обработки указателей границ пустых и полных областей буферов приемника и передатчика. В линейном режиме границы начала пустой области буфера приемника (head_R) и передатчика (tail_X) определяются записью в соответствующие регистры управления. В автоматическом режиме эти границы определяются автоматически по адресам последних чтения или записи в соответствующие области. В режиме FIFO границы определяются автоматически, при этом обращение к приемнику производится через адрес 0x0000, а передатчику через адрес 0x0004.

Передача пакета

Для передачи пакета(ов) необходима предварительная настройка блока передатчика MAC. Настройка параметров работы передатчика осуществляется посредством регистра управления передатчика X_CFG:

- разрешение работы передатчика (бит EN);
- порядок следования байт в буфере (бит BE);
- порядок следования бит в байте (бит MSB1st);
- выбор события при передаче, выводимого на вывод EVNT[1] (поле EVNT_MODE);
- управление дополнением пакета до минимальной длины PAD-ами (бит PAD_EN);
- управление дополнением пакета преамбулой (бит PRE_EN), SFD добавляется в любом случае;
- управление дополнением пакета, автоматически подсчитываемым полем CRC (CRC_EN);
- управление интервалом между отправлением пакетов (бит IPG_EN);
- управление максимальным числом повторений (поле RtryCnt).

Далее необходимо записать пакет для передачи в буфер передатчика.

Пакет для передачи содержит 3 поля (все поля должны быть выровнены по границе слова буферного ОЗУ):

- поле управления передачей пакета
- собственно данные пакета уровня звена данных;
- поле состояния передачи пакета.

Поле управления содержит кол-во байт пакета в буфере, которые должны быть переданы.

Поле состояния заполняется по завершении процедуры отправки пакета (успешной или нет) и содержит статусную информацию по отправке пакета: о наличии ошибок при его передаче, о кол-ве попыток передачи пакета и пр.

Если выбран линейный режим работы буферов, то после помещения пакета для передачи необходимо записать в регистр управления tail_X границу пустой области (адрес следующий за последним словом пакета).

По завершении передачи пакета блок MAC выставит один из флагов прерываний передатчика.

Во время отправки пакета или по его завершении выставляется событие, запрограммированное в поле X_CFG. EVNT_MODE.

Принцип работы передатчика

Передатчик начинает работать, прочитав ненулевое поле длины из буфера передатчика. Для этого необходимо чтобы буфер передатчика был не пуст (ненулевая разница между значениями head_X и tail_X) и передатчику было разрешено работать (X_CFG.EN=1). Прочитав слово управления, передатчик перемещает указатель head_X на первое слово пакета данных. При получении управляющего слова в передатчике так же фиксируется вся управляющая информация для работы передатчика, препятствуя срыву передачи текущего пакета и позволяя сменить настройки для отправки следующего пакета во время передачи текущего.

По завершении передачи в слово, следующее за последним словом данных, записывается статусная информация отправки пакета.

Прием пакета

Для приема пакета (ов) необходима предварительная настройка блока приемника MAC. Настройка параметров работы приемника осуществляется посредством регистра управления приемника R_CFG:

- разрешение работы приемника (бит EN);
- порядок следования байт в буфере (бит BE);
- порядок следования бит в байте (бит MSB1st);
- выбор события при передаче, выводимого на вывод EVNT[0] (поле EVNT_MODE);
- управление разрешением приема пакетов:
  - длины меньше минимально разрешенной (SF_EN);
  - длины больше максимально разрешенной (LF_EN);
  - пакетов управления (CF_EN);
  - пакетов содержащих ошибки (EF_EN);
- управление фильтрацией по MAC-адресу:
  - разрешение приема пакетов с заданным MAC-адресом (UCA_EN);
  - разрешение приема пакетов с широковещательным MAC-адресом (BCA_EN);
  - разрешение приема пакетов с групповым MAC-адресом (MCA_EN);
  - разрешение приема пакетов с любым MAC-адресом (AC_EN).

Для приема пакета необходимо чтобы в буфере приемника было достаточно пустого места для того чтобы вместить принимаемый пакет.

Принятый пакет содержит 2 поля (все поля выровнены по границе слова буферного ОЗУ):

- поле состояния приема пакета;
- собственно данные пакета уровня звена данных.

Поле состояния заполняется по успешном завершении процедуры приема пакета и содержит кол-во байт в пакете (включая заголовок пакет уровня звена дынных) а так же статусную информацию по приему пакета, о наличии ошибок при приеме.

Если выбран линейный режим работы буферов, то указание начала свободной для приема данных области указывается в регистре head_R.

По завершении приема пакета блок MAC выставит один из флагов прерываний приемника.

Во время приема пакета или по его завершении выставляется событие, запрограммированное в поле R_CFG.EVNT_MODE.

Принцип работы приемника

Приемник начинает работать сразу же после разрешения работы приемника в регистре R_CFG (R_CFG.EN=1), после обнаружения свободного места в буфере приемника. Обнаружив наличие свободного места, приемник фиксирует всю управляющую информацию для работы приемника, препятствуя срыву приема

изменениями настроек и позволяя сменить настройки для приема следующего пакета, и переходит в режим ожидания данных на входе, после поступления данных – в режим приема. По завершении приема в слово, следующее за последним словом данных, записывается статусная информация по приему пакета. Пакеты, отброшенные по причине ошибок в них или не прошедшие фильтрацию по MAC-адресу, переводят приемник в режим ожидания нового пакета, т.о. не изменяя общего состояния приемника, лишь изменяя состояние регистра флагов прерываний.

Линейный режим работы буферов

Данный режим включается сбросом поля BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE = 2'b00). В данном режиме все управление границами свободных областей в буферах осуществляется вручную.

Автоматический режим работы буферов

Для включения данного режима необходимо установить значение 1 в поле BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE=2'b01). В данном режиме в буфере автоматически отслеживаются указатели границ достоверных данных для передачи и приема по адресу записи в буфер передатчика и адресу чтения из буфера приемника. В данном режиме нет необходимости ручного управления границей свободного места в приемнике и передатчике через запись в соответствующие регистры. Это позволяет упростить алгоритм запуска передачи и приема и передавать данные одновременно с их помещением в буфер передатчика. Граница достоверных данных в буфере приемника перемещается по завершении приема пакета и т.о. данный режим не допускает одновременного приема пакета и его чтения из буфера

Режим FIFO работы буферов

Режим FIFO отличается от предыдущих полностью автоматическим отслеживанием данных в буфере. В данный режим модель переводится установкой значения 2 в поле BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE = 2'b10). В данном режиме чтение/запись в буферы статусной и управляющей информации, а также данных осуществляются через один адрес – 0x0000 для приемника и 0x0004 для передатчика. Для обеспечения корректной работы в режиме отладки необходимо использовать биты G_CFG.DBG_XF_EN и G_CFG.DBG_RF_EN для корректной работы со средствами отладки (для исключения некорректной отработки указателей буферов при обновлении карты памяти средствами отладки). Данный режим позволяет работать на максимальной скорости, если инструментальные средства управляющего контроллера не обеспечивают режимы адресации с автоинкрементом и циклической буферизацией.

События приемника и передатчика

В блоке MAC присутствуют 2 вывода индикации событий: события передатчика (EVNT[1]) и приемника (EVNT[0]). Основным назначением этих выводов является информирование управляющего процессора или DMA-контроллера о наличии данных для перемещения. Выводы EVNT программируемые. В качестве источника события могут быть выбраны:

- состояние буфера;
- начало приема/передачи пакета;
- завершение приема/передачи пакета;

– перемещение слова из/в буфера.

Первое событие предназначено для непрерывного обмена информацией с контролем состояния приема/передачи пакетов при использовании высокоуровневых протоколов. Следующие два события обеспечивают прием и передачу пактов в интерактивном режиме с непосредственным контролем их приема и передачи. Последнее событие предназначено для непосредственного контроля записи/чтения слов данных в/из буферов, и основное назначение этого события – режим отладки.

Прерывания

Прием и передача пакетов сопровождаются не только формированием событий, но и формированием прерываний, которые служат для непосредственного отражения оперативной статусной информации о состоянии Ethernet-контроллера. Вывод прерывания один, он обеспечивает общую индикацию наличия флагов в регистре флагов прерываний (IFR), разрешенных регистром маски прерываний (IMR). Все прерывания маскируемые. 1 в бите регистра маски прерываний (IMR) разрешает соответствующее прерывание, 0 – запрещает соответствующее прерывание. Прерывания делятся на три группы:

– прерывания MDIO интерфейса;

– прерывания передатчика;

– прерывания приемника.

Прерывания MDIO интерфейса информируют о завершении затребованной операции по MDIO интерфейсу.

Прерывания передатчика показывают состояние отправки пакетов, включая информацию об успешной отправке или наличие ошибок.

Прерывания приемника отражают состояние приема пакета, включая информацию о приеме пакета без ошибок или наличие ошибок при приеме.

Все флаги прерываний кумулятивные. Сброс флагов производится чтением регистра, если бит RCLR_EN регистра G_CFGl установлен (G_CFGl. RCLR_EN = 1), или записью 1 в соответствующий разряд регистра IFR.

Режим детерминированного времени доставки

Данный режим является расширением стандарта IEEE 802.3/Ethernet для обеспечения детерминированного времени доставки. Режим включается установкой бита DTRM_EN регистра G_CFGl (G_CFGl.DTRM_EN = 1). Данный режим может использоваться только в полнодуплексном режиме работы (значение бита G_CFGl.HD_EN = 1 блокирует данный режим).

В данном режиме для начала передачи пакета выделяется интервал, размером задваемым регистром JitterWnd (размер окна = JitterWnd + 1), с периодом задаваемым регистром BAG (период = BAG + 1). Единица измерения периода и размера джиттера задается регистром PSC в тактах основной частоты работы блока (размер едницы = PSC + 1).

Режим КЗ

В блоке MAC для целей тестирования алгоритмов обработки данных предусмотрен режим короткого замыкания (КЗ). В данном режиме выход передатчика переключается на вход приемника. Также в данном режиме блок принудительно переводится в полнодуплексный режим работы.

Режимы отладки

В блоке MAC предусмотрено различное поведение блока в режиме отладки. Это поведение определяется полем G_CFGh.DBG_MODE. Переход из штатного режима работы в одитн из отладочных режимов осуществляется при подаче сигнала DBG_EN = 1 на вход модуля. Доступны 3 режима отладки:

- FreeRun (G_CFGh.DBG_MODE = 2'b0X). В этом случае блок продолжает работать в шатном режиме.
- Halt (G_CFGh.DBG_MODE = 2'b10). Останов приемника и передатчика осуществляется сразу после перехода в данный режим.
- Stop (G_CFGh.DBG_MODE = 2'b11). Останов приемника и передатчика осуществляется по завершении прима/передачи текущего пакета.

Регистры

Базовый адрес	Название		Описание
0x38000000	Ethernet		Буфер данных контроллера интерфейса Ethernet
0x30000000	Ethernet		Регистры контроллера интерфейса Ethernet
Смещение (в байтах)	Название	Доступ и значение по умолчанию	Описание
0x00	Dilimiter	RW, 0x800	Регистр границы буферов приемника и передатчика
0x02	MAC-Address		Регистр индивидуального MAC-адреса
0x02	MAC_T	RW, 0x78AB	Младшая* часть индивидуального MAC-адреса
0x04	MAC_M	RW, 0x3456	Средняя часть индивидуального MAC-адреса
0x06	MAC_H	RW, 0x0012	Старшая часть индивидуального MAC-адреса
0x08	HASH		HASH-таблица групповых адресов
0x08	HASH0	RW, 0x0000	Младшая часть HASH-таблицы
0x0A	HASH1	RW, 0x0000	Средняя часть HASH-таблицы
0x0C	HASH2	RW, 0x0000	Средняя часть HASH-таблицы
0x0E	HASH3	RW, 0x8000	Старшая часть HASH-таблицы
0x10	IPG	RW, 0x0060	Регистр задания межпакетного интервала для полнодуплекного режима

0x12	PSC	RW, 0x0031	Регистр задания предделителя шага изменения значений BAG и JitterWnd (1 мкс при частоте 50 МГц)
0x14	BAG	RW, 0x0064	Регистр задания периода следования пакетов (100 мкс при частоте 50 МГц)
0x16	JitterWnd	RW, 0x0004	Регистр задания джиттера при передаче пакетов (5 мкс при частоте 50 МГц)
0x18	R_CFG	RW, 0x0507	Регистр управления приемника
0x1A	X_CFG	RW, 0x01FA	Регистр управления передатчика
0x1C	G_CFGl	RW, 0x4080	Регистр общего управления блоком, младшее полуслово
0x1E	G_CFGh	RW, 0x3000	Регистр общего управления блоком, старшее полуслово
0x20	IMR	RW, 0x0000	Регистр маски прерываний
0x22	IFR	RW, 0x0000	Регистр флагов прерываний
0x24	MDIO_CTRL	RW, 0x0000	Регистр управления канала MDIO интерфейса MII
0x26	MDIO_DATA	RW, 0x0000	Регистр данных канала MDIO интерфейса MII
0x28	R_Head	RW, 0x0000	Указатель начала области действительных данных приемника (указывает на первое непустое слово)
0x2A	X_Tail	RW, 0x0800	Указатель конца области действительных данных передатчика (указывает на первое пустое слово)
0x2C	R_Tail	R, 0x0000	Указатель конца области действительных данных приемника (указывает на первое пустое слово)
0x2E	X_Head	R, 0x0800	Указатель начала области действительных данных передатчика (указывает на первое непустое слово)
0x30	STAT	R, 0x0303	Регистр статуса

ПРИМЕЧАНИЕ:
* - указано назначение для режима MSB.

Поле управления передачи пакета

15	8
Length[15:8]	
R/W	
7	0
Length[7:0]	
R/W	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:16	-	Зарезервировано
15:0	Length[15:0]	Количество байт в пакете

Поле состояния передачи пакета

23	22	21	20	19	18	17	16
-	UR	LC	RL	RCOUNT			
U	R/W	R/W	R/W	R/W			

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:23	-	Зарезервировано
22	UR	Флаг опустошения буфера передатчика 1 – буфер передатчика пуст; 0 – буфер передатчика не пуст.
21	LC	Флаг индикации Late collision во время передачи пакета 1 – произошла Late collision во время передачи пакета; 0 – Late collision во время передачи пакета не происходила.
20	RL	Флаг исчерпания попыток передачи пакета 1 – превышено разрешенное количество попыток передачи пакета; 0 – количество попыток передачи пакета не превысило разрешенного значения;
19:16	RCOUNT[3:0]	Число попыток передачи пакета
15:0	-	Зарезервировано

Поле состояния приёма пакета

31	28	27	26	25	24
-			UCA	BCA	MCA
R/W					

23	22	21	20	19	18	17	16
SMB_ERR	CRC_ERR	DN_ERR	LEN_ERR	SF_ERR	LF_ERR	CF_ERR	PF_ERR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	8
Length[15:8]	
R/W	

7	0
Length[7:0]	
R/W	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:27	-	Зарезервировано
26	UCA	Признак индивидуального пакета (MAC-адрес соответствует установленному) 1 – MAC-адрес принятого пакета совпадает с MAC-адресом Ethernet-контроллера; 0 – MAC-адрес принятого пакета не совпадает с MAC-адресом Ethernet-контроллера.
25	BCA	Признак широковещательного пакета (MAC = FF_FF_FF) 1 – принят широковещательный пакет; 0 – широковещательный пакет не принят.
24	MCA	Признак группового пакета (MAC соответствует HASH) 1 – принят пакет, удовлетворяющий фильтрации по HASH-таблице; 0 – принятый пакет не удовлетворяет фильтрации по HASH-таблице или фильтрация отключена.
23	SMB_ERR	Признак наличия в пакете ошибочных nibbles 1 – наличие 0 – отсутствие
22	CRC_ERR	Признак несоответствия CRC пакета 1 – произошла ошибка сравнения CRC-пакета с вычисленной CRC; 0 – CRC-пакета и вычисленной CRC совпадают.
21	DN_ERR	Количество бит в пакете не кратно 8 1 – не кратно 8 0 – кратно 8
20	LEN_ERR	Признак несоответствия между реальной длиной и длинной указанной в поле длины – 13,14 октеты 1 – несоответствие 0 – соответствие
19	SF_ERR	Признак недостаточной длины пакета 64 октетов 1 – ошибочная длина 0 – корректная длина
18	LF_ERR	Признак превышение длины пакета 1518 октетов 1 – превышение 0 – норма

17	CF_ERR	Признак пакета управления (фильтрация по специальным MAC и тэгам в поле длины – 13,14 – октеты) 1 – пакет управления 0 – другой пакет
16	PF_ERR	Признак пакета PAUSE 1 – пакет PAUSE 0 – другой пакет
15:0	Length[15:0]	Количество байт в пакете, включая заголовок и CRC

G_CFGh

31	30	29	28	27			24
DEG_mode		DBG_XF_EN	DBG_RF_EN	-			
R/W,+0	R/W,+0	R/W,+1	R/W,+1				
23				19	18	17	16
-					DLB	RRST	XRST
					R/W,+0	R/W,+0	R/W,+0

G_CFGl

15	14	13	12	11	10	9	8
-	RCLR_EN	BUFF_MODE		-	HD_EN	DTRM_EN	PAUSE_EN
	R/W,+0	R/W,+0			R/W,+0	R/W,+0	R/W,+0
7							0
ColWnd							
R/W,+0							

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31:30	DBG_mode	Режим работы в режиме отладки. 2’b00 – FreeRun; 2’b10 – Halt; 2’b11 – Stop.
29	DBG_XF_EN	Разрешение автоматического изменения указателей FIFO передатчика. 0 – запрещено; 1 – разрешено.
28	DBG_RF_EN	Разрешение автоматического изменения указателей FIFO приемника в режиме отладки. 0 – запрещено; 1 – разрешено.
18	DLB	Режим КЗ. 0 – выключен; 1 – включен.
17	RRST	Сброс приемника. 0 – работает;

		1 – сброшен.
16	XRST	Сброс передатчика. 0 – работает; 1 – сброшен.
14	RCLR_EN	Сброс регистров статуса 0 – производиться записью в регистры статуса; 1 – регистры статуса сбрасываются при чтении.
13:12	BUFF_MODE	Режим работы буфера. 2'b00 – линейный режим; 2'b 01 – режим с автоматическим изменением указателей; 2'b 10 – режим FIFO; 2'b 11 – зарезервировано (линейный режим).
11	EXT_EN	Включение режима дополнения коротких пакетов до размера slotTime полем "Extension" (При приеме отбрасывание слова осуществляется по полю length пакета, если оно отражает длину пакета). 0 – выключен; 1 – включен.
10	HD_EN	Полудуплексный режим работы. 0 – выключен; 1 – включен.
9	DTRM_EN	Режим детерминированного времени доставки. 0 – выключен 1 – включен
8	PAUSE_EN	Режим автоматической обработки пакета PAUSE. 0 – выключен; 1 – включен.
7:0	ColWnd[7:0]	Размер «окна коллизий».

X_CFG

15	14	13	12	11	10	8
EN	-	BE	MSB1st	-	EVNT_MODE	
R/W,+1		R/W,+0	R/W,+0		R/W,+5	
7	6	5	4	3		0
PAD_EN	PRE_EN	CRC_EN	IPG_EN		RtryCnt	
R/W,+0	R/W,+0	R/W,+0	R/W,+0		R/W,+0	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
15	EN	Разрешение работы передатчика. 0 – остановлен; 1 – разрешена работа.
13	BE	Порядок седования байт в слове передатчика. 0 – LitteleEndian; 1 – BigEndian.
12	MSB1st	Порядок следования бит при передаче байтов данных.

		0 – первым передается LSB; 1 – первым передается MSB.
10:8	EVNT_MODE[2:0]	Выбор режима работы вывода EVNT[1]. 3'b000 – XFIFO пуст; 3'b001 – XFIFO почти пуст; 3'b010 –XFIFO наполовину полон; 3'b011 – XFIFO почти полон; 3'b100 – XFIFO полон; 3'b101 – отправка пакета завершена; 3'b110 – передатчик считал слово данных из буфера; 3'b111 – передатчик начал очередную попытку передачи пакета.
7	PAD_EN	Дополнение пакета до минимальной длины PAD-ами. 0 – выключено; 1 – включено.
6	PRE_EN	Дополнение пакета преамбулой. 0 – выключено; 1 – включено.
5	CRC_EN	Дополнение пакета автоматически высчитанным CRC. 0 – выключено; 1 – включено.
4	IPG_EN	Режим выдержки паузы между отправкой пакетов. 0 – выключен; 1 – включен.
3:0	RtryCnt[3:0]	Масимальное кол-во попыток отправки пакета

R_CFG

15	14	13	12	11	10		8
EN	-	BE	MSB1st	-		EVNT_MODE	
R/W,+1		R/W,+0	R/W,+0			R/W,+5	
7	6	5	4	3	2	1	0
SF_EN	LF_EN	CF_EN	EF_EN	AC_EN	UCA_EN	BCA_EN	MCA_EN
R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
15	EN	Разрешение работы приемника. 0 – приемник остановлен; 1 – разрешена работа.
13	BE	Порядок седования байт в слове. 0 – LitteleEndian; 1 – BigEndian.

12	MSB1st	Порядок следования бит при приеме байтов данных. 0 – первым принимается LSB; 1 – первым принимается MSB.
10:8	EVNT_MODE[2:0]	Выбор режима работы вывода EVNT[1]. 3'b000 – RFIFO не пуст; 3'b001 – RFIFO почти не пуст; 3'b010 – RFIFO наполовину пуст; 3'b011 – RFIFO почти не полон; 3'b100 – RFIFO не полон; 3'b101 – прием пакета завершен; 3'b110 – приемник положил данных в буфер; 3'b111 – приемник отбросил пакет.
7	SF_EN	Разрешение приема пакетов длиной меньше минимальной. 0 – выключено; 1 – включено.
6	LF_EN	Разрешение приема пакетов длиной больше максимальной. 0 – выключено; 1 – включено.
5	CF_EN	Разрешение приема управляющих пакетов. 0 – выключено; 1 – включено.
4	EF_EN	Разрешение приема пакетов с ошибками. 0 – выключено; 1 – включено.
3	AC_EN	Прием пакетов без фильтрации MAC-адреса. 0 – выключен; 1 – включен.
2	UCA_EN	Прием пакетов с MAC-адресом указанным в регистре MAC_Address. 0 – выключен; 1 – включен.
1	BCA_EN	Прием пакетов с широковезательным MAC-адресом. 0 – выключен; 1 – включен.
0	MCA_EN	Прием пакетов с групповым MAC-адресом с фильтрацией по HAS-таблице. 0 – выключен; 1 – включен.

IMR/IFR

15	14	13	12	11	10	9	8
MII_RDY	MDIO_INT		CRS_LOST	LC	UNDF	XF_ERR	XF_OK
			R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0
7	6	5	4	3	2	1	0
SF	LF	CF	CRC_ERR	SMB_ERR	OVF	MISSED_F	RF_OK

R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0	R/W,+0

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
15	MII_RDY	Индиктор завершения текущей команды обмена по MDIO интерфейсу
14	MDIO_INT	Индиктор наличия прерывания по MDIO интерфейсу
12	CRS_LOST	Индикатор потери несущей во время передачи в полудуплексном режиме работы
11	LC	Индикатор наличия LateCollision в линии
10	UNDF	Индикатор опустошения буфера передатчика
9	XF_ERR	Индикатор наличия ошибок при передаче пакета
8	XF_OK	Индикатор успешной отправки пакета
7	SF	Индикатор приема пакета длинной менее минимальной
6	LF	Индикатор приема пакета длинной более максимальной
5	CF	Индикатор приема управляющих пакетов
4	CRC_ERR	Индикатор наличия несовпадения CRC пакета приятых данных с CRC пакета
3	SMB_ERR	Индикатор наличия ошибок в данных при вриеме пакета
2	OVF	Индикатор переполнения буфера приемника
1	MISSED_F	Индикатор потери пакета из-за отсутствия места в буфере приемника
0	RF_OK	Индикатор успешно принятого пакета

Примечание: Индикатор в состоянии единицы означает наличие события, в нуле отсутствие события

STAT

15	13	12	11	10	9	8
-		X_FULL	X_AFULL	X_HALF	X_AEMPY	X_EMPY
		R,+0	R,+0	R,+0	R,+0	R,+0
7	5	4	3	2	1	0
RCOUNT		R_FULL	R_AFULL	R_HALF	R_AEMPY	R_EMPY
		R,+0	R,+0	R,+0	R,+0	R,+0

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
12	X_FULL	1 — буфер передатчика полон 0 — буфер передатчика не полон
11	X_AFULL	1 — буфер передатчика почти полон 0 — буфер передатчика не в состоянии почти полон

10	X_HALF	1 — буфер передатчика полуполон 0 — буфер передатчика не полуполон
9	X_AEMPTY	1 — буфер передатчика почти пуст 0 — буфер передатчика не в состоянии почти пуст
8	X_EMPTY	1 — буфер передатчика пуст 0 — буфер передатчика не пуст
7:5	R_COUNT	Кол-во принятых но не считанных пакетов 0..6 — кол-во пакетов 7 — кол-во несчитанных пакетов >=7
4	R_FULL	1 — буфер приемника полон 0 — буфер приемника не полон
3	R_AFULL	1 — буфер приемника почти полон 0 — буфер приемника не в состоянии почти полон
2	R_HALF	1 — буфер приемника полуполон 0 — буфер приемника не полуполон
1	R_AEMPTY	1 — буфер приемника почти пуст 0 — буфер приемника не в состоянии почти пуст
0	R_EMPTY	1 — буфер приемника пуст 0 — буфер приемника не пуст

MDIO_CTRL

15	14	13	12				8
RDY	PRE_EN	OP			PHY_A		
	R/W,+0	R/W,+0			R/W,+0		
7		5	4				0
	DIV				RG_A		
	R/W,+0				R/W,+0		

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
15	RDY	Управление/индикатор обмена по MDIO После записи команды необходимо установить в единицу для инициирования исполнения команды в регистре MDIO_CTRL после одного такта сбрасывается в ноль и снова устанавливается в единицу после завершения цикла обмена по интерфейсу MDIO.
14	PRE_EN	Режим передачи. 1 – с передачей преамбулы (32 бита «1»); 0 – без передачи преамбулы.
13	OP	Операция. 1 – чтение; 0 – запись.
12:8	PHY_A[4:0]	Адрес модуля PHY

7:5	DIV	Коэффициент деления основной частоты для работы MDIO интерфейса MDC= ETH_CLK / [(DIV+1)*16]
4:0	RG_A	Номер регистра PHY

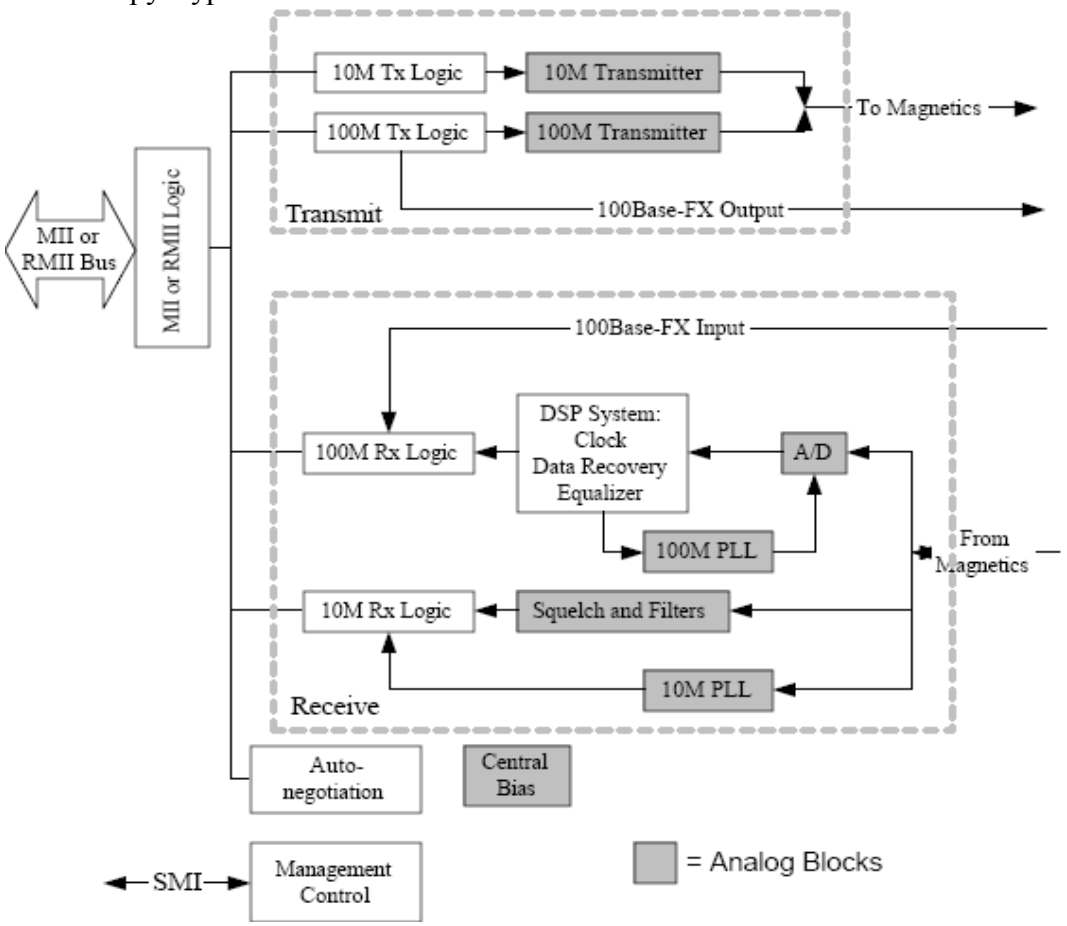
Блок PHY

Блок PHY реализует физический уровень протоколов Ethernet/IEEE 802.3. Он может функционировать в одном из следующих режимов:

- 10Base-T FD (full duplex);
- 10Base-T HD (half duplex);
- 100Base-T FD (full duplex);
- 100Base-T HD (half duplex);
- 100Base-FX.

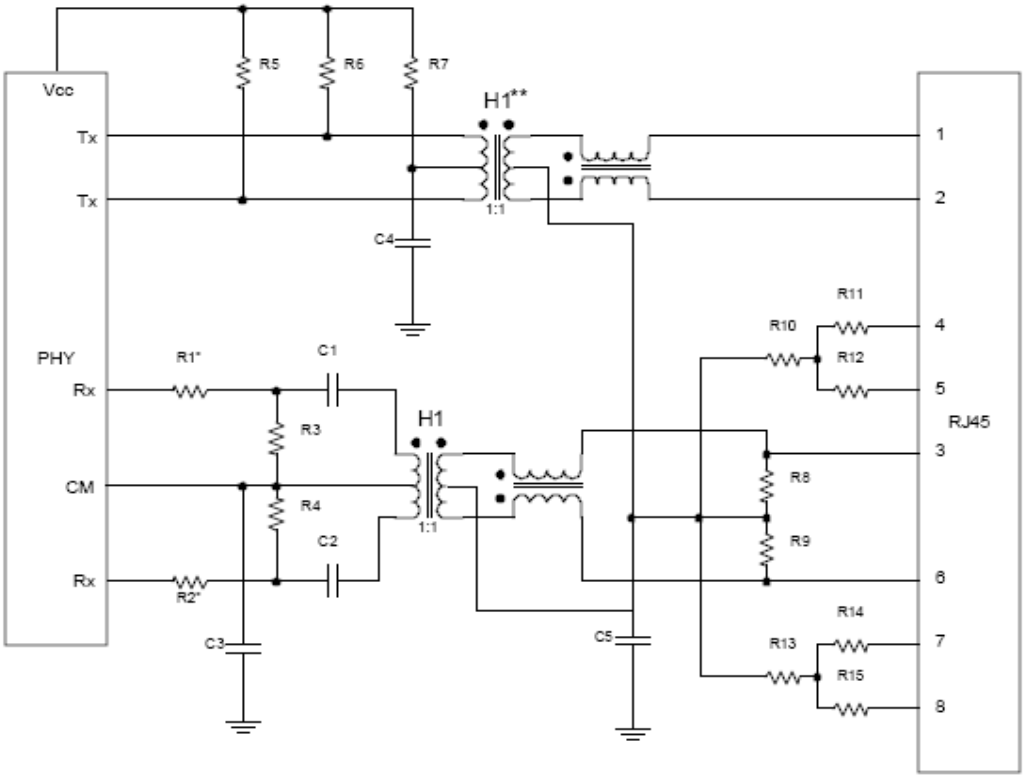
Кроме того данный блок обладает автоподстройкой (AutoNegotioation) параметров линии и обеспечивает автоматическое их определение для выбора режима работы с наибольше пропускной способностью.

Рис.1. Структурная схема блока PHY



Блок PHY обеспечивает простое подключение к линии с использованием небольшого количества внешних элементов. Частота тактирования блока PHY должна быть 25 МГц с джиттером менее 100 пс и длительностью фронтов менее 3 нс.

Рис.2 Схема подключения к линии.



Управление режимами блока осуществляется через регистры PHY_CTRL и PHY_STAT блока MAC.
При этом через регистр PHY_CTRL осуществляется программный сброс блока, а также настрйока режимов его работы после сброса.
В регистре PHY_STAT отражается информация о текущем состоянии блока PHY.

Примечание. После аппаратного сброса необходимо выдержать паузу 16мс для выхода блока PHY в рабочий режим.

Регистры

Базовый адрес	Название		Описание
0x30000000	Ethernet		Контроллер интерфейса Ethernet
Смещение (в байтах)	Название	Доступ и значение по умолчанию	описание
0x34	PHY_Control		Регистр флагов статуса PHY
0x36	PHY_Status		Регистр управления PHY

PHY_Control

15	14	13	12	11	10	9	8
		PHYADD			MDC	MDIO_SEL	MDI
		R/W			R/W	R/W	R/W
7	6	5	4	3	2	1	0
FX_EN				MODE			nRST
					R/W	R/W	R/W

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
15:11	PHYADD[4:0]	Адрес PHY используемый для SMI интерфейса и для инициализации скрамблера
10	MDC	Тактовый сигнал обмена через SMI блока PHY (для ручного управления работой через SMI)
9	MDIO_SEL	Выбор режима управления обмена данными через SMI интерфейс. 1 – ручное управление работой через SMI; 0 – обмен через SMI осуществляется через регистры и автомат, встроенные в блок MAC.
8	MDI	Состояние входа данных SMI блока PHY (для ручного управления работой через SMI)
7	FX_EN	Выбор режима работы блока PHY 100BaseFX. 1 – включен режим 100BaseFX; 0 – режим 100BaseFX выключен.
6:4	-	Зарезервировано
3..1	MODE[2:0]	Режим работы блока PHY. 3’b000 – 10BaseT HD без автоподсткойки; 3’b001 – 10BaseT FD без автоподсткойки; 3’b010 – 100BaseT HD без автоподсткойки; 3’b011 – 100BaseT FD без автоподсткойки; 3’b100 – 100BaseT HD с автоподсткойкой; 3’b101 – режим повторителя; 3’b110 – режим пониженого потребелния; 3’b111 – Полностью автоматический режим.
0	nRST	Разрешение работы былока PHY. 0 – блок PHY сброшен; 1 – блок PHY в штатном режиме.

PHY_Status

15	14	13	12	11	10	9	8
					MDINT	MDO	FX_VALID
		RO			RO	RO	RO
7	6	5	4	3	2	1	0

COL	CRS	READY	LED[3:0]
RO	RO	RO	RO

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
15:11	-	Зарезервировано
10	MDINT	Флаг запроса прерывания от блока PHY. 1 – имеется прерывание от блока PHY; 0 – от блока PHY прерывания отсутствуют (дублируется в регистре прерываний блока MAC).
9	MDO	Состояние выхода данных SMI блока PHY (для ручного управления работой через SMI)
8	FX_VALID	Флаг наличия обмена данными в оптоволоконной линии. 1 – присутствует обмен в линии FX; 0 – линия FX в исходном состоянии.
7..6	COL	Флаг наличия коллизии в линии. 1 – в линии присутствует коллизия; 0 – в линии коллизия отсутствует.
5	CRS	Флаг наличия обмена данными по витой паре. 1 – в линии идет обмен данными; 0 – линия в исходном состоянии.
4	READY	Флаг готовности к работе блока PHY. 1 – блок PHY вышел в рабочий режим после аппаратного сброса/отключения; 0 – блок PHY не в рабочем режиме.
3	LED3	Индикация режима работы блока PHY. 0 – режим работы full-duplex; 1 – режим работы half-duplex.
2	LED2	Индикация наличия Carrier sense. 0 – наличие Carrier sense (CRS); 1 – отсутствие Carrier sense (CRS).
1	LED1	Индикация наличия Link сигнала. 0 – сигнал Link включён; 1 – сигнал Link выключен.
0	LED0	Индикация выбранной скорости обмена данными. 0 – выбрана скорость 100 Мбит; 1 – выбрана скорость 10 Мбит.

Для доступа к внутренним регистрам блока PHY предназначены регистры SMI_CTRL и SMI_DATA блока MAC.

Ниже приведен перечень и описание внутренних регистров блока PHY

Регистр №	**Описание**	**Группа**
0	Основной регистр управления	основные
1	Основной регистр состояния	основные
2	Регистр идентификатора PHY 1	расширенные

3	Регистр идентификатора PHY 2	расширенные
4	Регистр рекомендаций автоподстройки	расширенные
5	Регистр возможностей оппонента по автоподстройке	расширенные
6	Регистр расширеного управления автоподсткройкой	расширенные
18	Регистр расширенного управления режимами	производителя
29	Регистр флагов прерываний	производителя
30	Регистр маски прерываний	производителя
31	Регистр расширенного управления и состояни PHY	производителя

Основной регистр управления (0)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15	Reset	Программный сброс блока PHY. 1 – программный сброс. Самоочищаемый. Рекомендуется не изменять остальные биты данного регистра во время его установки.	RW/SC	0
14	Loopback	Режим КЗ блока PHY. 1 – режим КЗ; 0 – штатный режим.	RW	0
13	Speed Select	Выбор скорости обмена данными. 1 – 100Mbps; 0 – 10Mbps. Игнорируется, если установлен бит AutoNegotiation (0.12 = 1).	RW	PHY_CTRL
12	Auto-Negotiation Enable	Разрешение режима автоподстройки. 1 – режим автоподстройки включен; 0 – режим автоподстройки отключен.	RW	PHY_CTRL
11	Power Down	Разрешение режима пониженного энергопотребления. 1 – режим пониженого потребления; 0 – штатный режим.	RW	0
10	Isolate	Разрешение отключения интерфейса MII от PHY. 1 – PHY отключён от интерфейса MII 0 – нормальное функционирование	RW	PHY_CTRL
9	Restart Auto-Negotiate	Перезапуск автоподстройки. 1 – перезапуск автоподстройки; 0 – штатный режим. Самоочищаемый.	RW/SC	0
8	Duplex Mode	Выбор режима работы блока PHY. 1 – полнодуплексный режим; 0 – полудуплексный режим. Игнорируется, если установлен бит AutoNegotiation (0.12 = 1).	RW	PHY_CTRL
7	Collision Test	Выбор режима тестирования Collision Test.	RW	0

		1 – включен COL test; 0 – COL test отключен.		
6:0	Reserved		RO	0

Основной регистр состояния (1)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15	100Base-T4	1 – доступен режим 100Base-T4, 0 – режим 100Base-T4 не доступен	RO	0
14	100Base-TX Full Duplex	1 – возможен режим полного дуплекса 100Mbps, 0 – режим полного дуплекса 100Mbps не возможен	RO	1
13	100Base-TX Half Duplex	1 – возможен режим полудуплекса 100Mbps, 0 – режим полудуплекса 100Mbps не возможен	RO	1
12	10Base-T Full Duplex	1 – возможен режим полного дуплекса 10Mbps, 0 – режим полного дуплекса 10Mbps не возможен	RO	1
11	10Base-T Half Duplex	1 – возможен режим полудуплекса 10Mbps, 0 – режим полудуплекса 10Mbps не возможен	RO	1
10:6	Reserved		RO	0
5	Auto-Negotiate Complete	1 – автоподстройка завершена 0 – автоподстройка не завершена	RO	0
4	Remote Fault	1 – обнаружено состояние remote fault 0 – remote fault отсутствует	RO/LH	0
3	Auto-Negotiate Ability	1 – возможна автоподстройка 0 – автоподстройка не возможна	RO	1
2	Link Status	1 – линия подключена 0 – линия отключена	RO/LL	0
1	Jabber Detect	1 – обнаружено состояние jabber 0 – состояние jabber отсутствует	RO/LH	0
0	Extended Capabilities	1 – поддерживаются расширенные регистры 0 – расширенные регистры не поддерживаются	RO	1

Регистры идентификатора PHY (2, 3)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15:0	PHY ID	32-битный идентификатор модели PHY	RW	

	Number.			

Регистр рекомендаций автоподстройки (4)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15	Next Page	1 – next page поддерживается, 0 – next page не поддерживается	RO	0
14	Reserved		RO	0
13	Remote Fault	1 – обнаружена remote fault, 0 – remote fault не обнаружена	RW	0
12	Reserved		R/W	0
11:10	Pause Operation	00 – PAUSE не обрабатывается 01 – Асимметричная обработка PAUSE 10 – Симметричная обработка PAUSE 11 – Возмоны и симметричная и асимметричная обраотки PAUSE	R/W	00
9	100Base-T4	1 – доступен режим 100Base-T4, 0 – режим 100Base-T4 не доступен.	RO	0
8	100Base-TX Full Duplex	1 – доступен режим 100Base-T с полным дуплексом, 0 – режим 100Base-T с полным дуплексом не доступен	RW	PHY_CTRL
7	100Base-TX	1 – доступен режим 100Base-T, 0 – режим 100Base-T не доступен	RW	1
6	10Base-T Full Duplex	1 – доступен режим 10Base-T с полным дуплексом, 0 – режим 10Base-T с полным дуплексом не доступен	RW	PHY_CTRL
5	10Base-T	1 – доступен режим 10Base-T, 0 – режим 10Base-T не доступен	RW	PHY_CTRL
4:0	Selector Field	[00001] – IEEE 802.3	RW	00001

Регистр возможностей оппонента по автоподстройке (5)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15	Next Page	1 – next page поддерживается, 0 – next page не поддерживается	RO	0
14	Acknowledge	1 – получено кодовое словое link 0 – кодовое словое link еще не получено	RO	0
13	Remote Fault	1 – обнаружена remote fault, 0 – remote fault не обнаружена	RO	0
12:11	Reserved		RO	0
10	Pause Operation	1 – обработка PAUSE поддерживается 0 – обработка PAUSE не поддерживается	RO	0
9	100Base-T4	1 – доступен режим 100Base-T4, 0 – режим 100Base-T4 не доступен.	RO	0
8	100Base-TX	1 – доступен режим 100Base-T с полным	RO	0

	Full Duplex	дуплексом, 0 – режим 100Base-T с полным дуплексом не доступен		
7	100Base-TX	1 – доступен режим 100Base-T, 0 – режим 100Base-T не доступен	RO	0
6	10Base-T Full Duplex	1 – доступен режим 10Base-T с полным дуплексом, 0 – режим 10Base-T с полным дуплексом не доступен	RO	0
5	10Base-T	1 – доступен режим 10Base-T, 0 – режим 10Base-T не доступен	RO	0
4:0	Selector Field	[00001] – IEEE 802.3	RO	00001

Регистр расширеного управления автоподсткройкой (6)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15:5	Reserved		RO	0
4	Parallel Detection Fault	1 – обнаружена ошибка parallel detection logic 0 – ошибки отсутствуют	RO/LH	0
3	Link Partner Next Page Able	1 – оппонент поддерживает next page 0 – оппонент не поддерживает next page	RO	0
2	Next Page Able	next page не поддерживается	RO	0
1	Page Received	1 – получена новая страница 0 – новая страница еще не получена	RO/LH	0
0	Link Partner Auto-Negotiation Able	1 – link partner has auto-negotiation ability 0 – link partner does not have auto-negotiation ability	RO	0

Регистр расширенного управления режимами (18)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15:14	MIIMODE	Режим работы MII: должен быть установлен в «00» (MII)	RW, NASR	0
13	CLKSELFREQ	Опорная частота должен быть установлен в «0» (25MHz)	RO, NASR	0
12	DSPBP	Режим обзода DSP. Тольбко для лабораторных тестов	RW, NASR	0
11	SQBP	Режим обхода SQUELCH.	RW, NASR	0
10	FXMODE	Разрешение режима 100Base-FX. Если включен то режим (MODE) должен	RW, NASR	PHY_CTRL

		быть выставлен только "011" (100Base-TX FD) илиг "010" (100Base-TX FD).		
9	PLLBP	Режим обхода PLL.	RW, NASR	0
8	ADCBP	Режим обхода АЦП.	RW, NASR	0
7:5	MODE	Текущий режим работы PHY.	RW, NASR	PHY_CTRL
4:0	PHYADD	PHY Address. Используется для достума посредством SMI, а так же в качестве ключа для скремблирования.	RW, NASR	0

Регистр флагов прерываний (29)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15:8	Reserved	Ignore on read.	RO/LH	0
7	INT7	1 – ENERGYON 0 – нет прерывания	RO/LH	0
6	INT6	1 – автоподсктройка завершена 0 – нет прерывания	RO/LH	0
5	INT5	1 – обнаружена Remote Fault Detected 0 – нет прерывания	RO/LH	0
4	INT4	1 – отсутствет подключение к линии 0 – нет прерывания	RO/LH	0
3	INT3	1 – подтвержение автоподстройки от оппонента 0 – нет прерывания	RO/LH	0
2	INT2	1 – Parallel Detection Fault 0 – нет прерывания	RO/LH	0
1	INT1	1 – получена страница автоподстройки 0 – нет прерывания	RO/LH	0
0	Reserved		RO/LH	0

Регистр маски прерываний (30)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15:8	Reserved		RO	0
7:0	Mask Bits	1 – прерывание разрешено 0 – прерывание запрещено	RW	0

Регистр расширенного управления и состояни PHY (31)

Бит	Наим-е	Описание	режим	Знач. По умолч.
15:14	Reserved		RW	0
13			RO	0
12	Autodone	Индикатор завершения автоподстройки: 0 – автоподстройка не завершена или отключена 1 – автоподстройка завершена	RO	0
11-7	Reserved		RW	0
6	enable 4B5B	0 – пропустить кодирование/декодирование. 1 – включить кодирование/декодирование 4B5B.	RW	1
5	Reserved		RW	0
4:2	Speed Indication	Значение HCDSPEED: [001] – 10Mbps HD [101] – 10Mbps FD [010] – 100Base-TX HD [110] – 100Base-TX FD	RO	0
1	Reserved		RW	0
0	Scramble Disable	0 – скремблирование включено 1 – скремблирование отключено	RW	0

Прерывания и исключения

Процессор и вложенный векторный контроллер прерываний (NVIC) назначают приоритет и обрабатывают все исключения. Все исключения обрабатываются в режиме Handler. Состояние процессора автоматически сохраняется в стек при возникновении исключения и автоматически восстанавливается из стека по завершению обработки исключения. Следующие характеристики позволяют увеличить эффективность обработки исключений и уменьшить задержки:

- Автоматическое сохранение состояния. Процессор помещает содержимое регистров в стек при входе в исключение и извлекает их при выходе из исключения, при этом не затрачивается дополнительных инструкций.

- Автоматическое считывание таблицы векторов при входе, которая содержит адрес обработчика. Бит 0 значения вектора загружается в T-бит регистра EPSR при входе в исключение. Создание таблицы входа с очищенным битом 0 генерирует аппаратную ошибку Hard Fault на первой инструкции обработчика соответствующему этому вектору.

- Тесно связанный интерфейс между процессором и NVIC позволяет эффективно обрабатывать прерывания и поздно поступающие (late-arriving) запросы прерывания с высоким приоритетом.

- Два бита конфигурирования приоритета прерываний обеспечивают 4 уровня приоритета.

- Разделённый стек для режимов Handler и Thread.

- Команды передачи управления исключению соответствуют соглашению C/C++ стандарту ARM Architecture Procedure Call Standard (AAPCS).

- Маскирование приоритета для поддержки критических регионов.

Типы исключений

В процессоре существуют различные типы исключений. Ошибка это исключение, возникшее вследствие ошибочных условий. Ошибки могут генерироваться синхронно или асинхронно с соответствующей инструкцией, которая её вызвала. Обычно ошибки генерируются синхронно. Ошибки, вызванные записью на внешней шине AHB асинхронные. Синхронные ошибки всегда генерируются совместно с инструкцией, которая её вызвала. Генерация асинхронных ошибок совместно с вызвавшей её инструкцией не гарантирована.

В таблице представлены типы исключений, их номера и приоритет. Номер показывает словное смещение векторов исключений относительно стартового адреса таблицы векторов, которая всегда располагается с адреса 0x0. Исключения с наименьшими числами, представленные в столбце приоритета таблицы, имеют наивысший приоритет. Как формируются исключения, асинхронно или синхронно, также показано.

Таблица различных типов исключений

Номер	Тип	Приоритет	Описание	Активация
-	-	-	Вершина стека загружается в начало таблицы после сброса	
1	RESET	-3 (наивысший)	Вызывается при включении питания или горячем сбросе. На первой инструкции в режиме Thread падает до низшего приоритета.	Асинхронный
2	NMI	-2	Это исключение не может быть: - маскировано или задержано активизацией любого другого исключения; - заменено любым другим исключением кроме сброса	Асинхронный
3	Hard Fault	-1	Все виды ошибок	Синхронный/ Асинхронный
4-10	-	-	Зарезервировано	-
11	SVCall		Системное обслуживание, вызванное инструкцией SVC	Синхронный
12-13	-	-	Зарезервировано	-
14	PendSV	конфигурируемый	Запрос ожидания обработки для обслуживания системы. Ожидание обработки генерируется программным обеспечением.	Асинхронный
15	SysTick	конфигурируемый	Системный таймер закончил работу	Асинхронный
16-47	IRQ	конфигурируемый	Запрос устанавливается извне процессора или обусловлен программным обеспечением.	Асинхронный

Приоритет исключений

В таблице показано влияние приоритетов, на то когда и как процессор обрабатывает исключения.

Сценарий исключений

Сценарий	Описание
Приоритетное прерывание обслуживания	Ожидающее обработки исключение может прервать текущее выполнение задачи, если приоритет этого исключения выше, чем приоритет текущего прерывания. Когда одно исключение прерывает обслуживание другого, то появляется вложенность исключений. При входе в исключение процессор автоматически сохраняет своё состояние, помещая его в стек. Выбирается вектор соответствующий исключению. Выполнение начинается с адреса указанного в таблице векторов. Выполнение первой инструкции начинается, когда состояние

	процессора сохранено. Сохранение состояния происходит через ITCM, DTCM или AHB-Lite интерфейсы в зависимости от: - значения указателя стека, когда процессор обнаружил исключение; - размера памяти TCM. Выбор вектора происходит по внешнему AHB-Lite интерфейсу или ITCM интерфейсу памяти в зависимости от конфигурации ITCM.
Возврат	Когда выполняется инструкция возврата, процессор выгружает стек и возвращается к помещённому в стек исключению или в Thread режим. По завершению выполнения обработчика исключения процессор автоматически восстанавливает состояние, выгружая стек и переходит в состояние предшествующее исключению.
Запаздывание (Late-arriving)	Это механизм, используемый процессором для ускорения прерывания обслуживания. Если исключение с более высоким приоритетом прибывает во время сохранения состояния предыдущего прерывания обслуживания, то процессор переключается на обработку исключения с более высоким приоритетом вместо выполнения выборки вектора для этого исключения. На сохранение состояния процессора запоздавшее исключение не влияет, потому что сохраняемое состояние аналогично для обоих исключений и поэтому продолжается не прерываясь. Запаздывающие исключения распознаются в момент, когда происходит выборка вектора. Если исключение с более высоким приоритетом распознаётся слишком поздно, чтобы быть обслужено как запаздывающее, то оно ожидает обработки и позднее прерывает обслуживание исходного обработчика исключения.

Уровни приоритета

NVIC поддерживает программное присвоение уровней приоритета. Можно установить требуеый уровень приоритета прерыванию записью значения от 0 до 3 в поле IP_N регистра приоритета прерывания. Приоритет с уровнем 0 считается наивысшим, а с уровнем 3 самым низким. Например, если присвоить уровень приоритета 1 для IRQ[0] и уровень приоритета 0 для IRQ[31], то IRQ[31] имеет приоритет выше IRQ[0]. Программное присвоение приоритетов не влияет на немаскируемое прерывание NMI и исключение Hard Fault. Они всегда имеют приоритет выше, чем внешние прерывания.

Когда несколько исключений имеют одинаковый приоритет, тогда сначала будет обработано исключение с меньшим порядковым номером, а самым последним – с наибольшим номером. Например, если оба IRQ[0] и IRQ[1] имеют приоритет 1, тогда IRQ[0] предшествует IRQ[1].

Новое исключение прерывает обработку текущего исключения, если его приоритет выше. Если новое исключение с таким же приоритетом, как обрабатываемое в текущий момент, то обработка не прерывается независимо от номера прерывания.

Стек.

Процессор поддерживает два индивидуальных стека:

Процессорный стек.

Необходимо установить Thread режим, чтобы использовать SP_process или SP_main.

Основной стек.

Режим Handler использует только основной стек.

Когда происходит прерывание обслуживания, контекст автоматически сохраняется в стек, который был активным в момент обнаружения исключения. Если исключение прерывает обслуживание в Thread режиме, то контекст прерванной задачи может быть помещён в стек с применением SP_process или SP_main в зависимости от значения бита CONTROL[1].

Если исключение прерывает обслуживание другого исключения выполняемого в режиме Handler, то прерванный контекст можем быть помещён только в стек SP_main, так как только этот указатель стека может быть активным в режиме Handler.

При возвращении из прерывания значение EXC_RETURN определяет, какой стек использовать для извлечения контекста. Значение EXC_RETURN помещается в регистр R14 в процессе входа в исключение, и соответствующий стек используется для сохранения контекста. Если ваш код обработчика исключения изменяет стек, то вы должны быть уверены, что значение EXC_RETURN для возвращения из исключения корректно.

Все обработчики исключений должны использовать SP_main для их локальных переменных. Инструкциям MSR и MRS доступны оба указателя стека.

Приоритетное прерывание обслуживания

Когда процессор обрабатывает исключение, то автоматически помещаются в стек следующие 8 регистров:

- xPSR;
- Адрес возврата;
- Регистр связи (LR);
- R12;
- R3;
- R2;
- R1;
- R0.

SP декрементируется на 8 слов по завершению загрузки стека. На рисунке показано содержимое стека после того как исключение прервало обслуживание текущего программного потока.

предыдущее состояние	← Старый
xPSR	
Адрес Возврата	
LR	
R12	
R2	
R3	
R1	
R0	
	← SP

После возвращения из исключения процессор автоматически выгружает 8 регистров из стека. Значение возврата из исключения EXC_RETURN автоматически загружается в LR при входе в исключение, позволяя описывать обработчики прерываний как обычные C/C++ функции.

Таблица описывает шаги процессора, прежде чем происходит вход в исключение.

Действие	Описание

Сохранение 8 регистров	Сохранение xPSR, Адрес возврата, LR, R12, R3, R2, R1 и R0 в стек
Чтение таблицы векторов	Чтение вектора входа из соответствующего адреса таблицы векторов: (0x0)+(номер исключения*4). Чтение таблицы векторов происходит после того как все 8 регистров помещены в стек.
Чтение SP_main из таблицы векторов	SP_main обновляется в таблице векторов только после сброса. Другие исключения не модифицируют SP_main таким способом.
Обновление LR	LR устанавливается в соответствии с EXC_RETURN для корректного возврата из исключения
Обновление PC	Обновление PC прочитанными данными из таблицы векторов. Никакие другие запаздывающие прерывания не могут выполняться, пока первая инструкция исключения не начнёт исполняться.
Загрузка конвейера	Конвейер заполняется последовательностью инструкций из адреса вектора.

Выход из исключений

Инструкция возврата из исключения загружает PC значением EXC_RETURN, которое было загружено в LR при входе в обработчик прерывания. Это сигнализирует процессору о том, то исключение завершено и процессор инициирует последовательность выхода из исключения.

При выходе их исключения процессор либо возвращается в последнее, помещённое в стек исключение, либо переходит в режим Thread.

Таблица описывает шаги процессора, прежде чем происходит вход в исключение.

Действие	Описание
Выбор SP	Установка CONTROL[1] в соответствии с EXC_RETURN.
Выгрузка 8 регистров из стека	Выгрузка R0, R1, R2, R3, R12, LR, PC и xPSR из стека выбранного EXC_RETURN. Значение xPSR[5:0] выгружаемое из стека определяет номер исключения, что в свою очередь определяет приоритет задачи, к которой необходимо вернуться. Значение EXC_RETURN определяет, в какой режим вернуться.

Возврат из исключения происходит при выполнении одной из следующих инструкций выполняемых в режиме Handler и загружающих в PC значение 0xFXXXXXXX:

- POP, которая включает загрузку PC;
- BX с любым регистром.

Если используется такой способ, значение, записываемое в PC, заменяет значение EXC_RETURN.

Таблица описывает поведение при выходе из исключения при различных EXC_RETURN[3:0].

EXC_RETURN[3:0]	Описание
4'bXXX0	Зарезервировано
4'b0001	Возврат в режим Handler. Возврат из исключения с получением состояния из основного стека. Далее после выхода из обработчика исключения при исполнении

	кода использует SP_Main.
4’b0011	Зарезервировано
4’b01X1	Зарезервировано
4’b1001	Возврат в режим Thread. Возврат из исключения с получением состояния из стека Main. Далее после выхода из обработчика исключения при исполнении кода использует SP_Main.
4’b1101	Возврат в режим Thread. Возврат из исключения с получением состояния из стека Process. Далее после выхода из обработчика исключения при исполнении кода использует SP_Process.
4’b1X11	Зарезервировано

Если значение EXC_RETURN загружается в PC в режиме Thread или из таблицы векторов или любой другой инструкцией, значение рассматривается как адрес, а не как специальное значение. Если адрес из диапазона адресов имеющий атрибут XN (выполнение запрещено), то возникает аппаратная ошибка Hard Fault.

Обработчик прерывания должен сохранить значение EXC_RETURN[28:4] или записать их как все единицы.

Запаздывание (late-arriving)

Запаздывающее исключение может иметь преимущество в обслуживании по отношению к предыдущему исключению, если выборка вектора не началась и запаздывающее исключение имеет:

- приоритет выше предыдущего исключения;
- одинаковый приоритет, но меньший порядковый номер, чем предыдущее исключение.

Запаздывающее исключение вызывает изменение выборки вектора адреса и предвыборки исключения. Сохранение состояния не выполняется для запаздывающего исключения, потому что это уже было выполнено для исходного исключения. В этом случае, выполнение начинается с вектора запаздывающего исключения, в то время как предыдущее исключение ожидает обслуживания.

Если исключение с высоким приоритетом распознаётся после выборки вектора исходного исключения, то запаздывающее исключение не может использовать контекст, который помещён в стек для исходного исключения. В этом случае прерывается обслуживание исходного прерывания, и контекст сохраняется в стеке.

Передача управления исключению

Таблица показывает, в соответствии с какими правилами процессор передаёт управление исключению.

Активность процессора и обнаружение исключения	Передача управления исключению
Инструкция	Завершается инструкция и исключение начинается перед выполнением следующей инструкции.
Вход в исключение	Это классифицируется как запаздывающее исключение. Если новое исключение имеет приоритет выше или такой же приоритет и номер

	исключения меньше, чем первое исключение, то ядру необходимо обслужить запаздывающее исключение первым. Если нет, то запаздывающее прерывания ожидает обслуживания и используются обычные правила прерывания обслуживания. Если запаздывающее исключение поступило достаточно рано во время фазы помещения в стек, то оно рассматривается как запаздывающее. В этом случае, ядро выбирает вектор для запаздывающего исключения вместо вектора первого исключения. Если запаздывающее прерывание поступает слишком поздно (позже времени фазы помещения контекста в стек), то оно не может обрабатываться как запаздывающее. Вместо этого выбирается вектор этого первого исключения, начинается выполнение по вектору адреса первого исключения, а запаздывающее прерывание ожидает обслуживания в соответствии с обычными правилами прерывания обслуживания.
Завершение исключение	Завершается последовательность возврата из исключения и восстанавливается выполнение задачи возврата. Обычные правила прерывания обслуживания применяются в этом случае.

Уровни активации

Если нет активных исключений, то процессор находится в режиме Thread. Если исключения или Hard Fault активны, то процессор входит в режим Handler.

Таблица показывает уровни активации стека.

Активное исключение	Уровень активации	Стек
Нет	Режим Thread	Основной или стек процессов
Исключение активно	Асинхронное прерывание обслуживания	Основной
Fault handler активен	Асинхронное или синхронное прерывание обслуживания	Основной

Таблица показывает транзакции исключений.

Активное исключение	Старт события	Тип транзакции	Стек
Сброс	Сигнал сброса	Thread	Основной
ISR или NMI[a]	Установка запроса обслуживания программной инструкцией или аппаратным сигналом	Асинхронное прерывание обработки	Основной
Hard Fault	любая ошибка	Синхронное или асинхронное прерывание обработки	Основной
SVC[b]	SVC инструкция	Синхронное прерывание обработки	Основной

a Немаскируемое прерывание
b Вызов супервизора

Таблица показывает подгруппы транзакций исключений.

Группа активации	Старт события	Активация	Приоритет
Thread	Сигнал сброса	Асинхронная	Незамедлительный,

			thread имеет низкий приоритет
Прерывание или NMI	Аппаратный сигнал или установка запроса обслуживания	Асинхронная	Прерывание обслуживания согласно приоритету
SVC	Инструкция SVC	Синхронная	Если приоритет для SVCall исключения запрограммирован выше чем для текущего исключения, то выполняется SVCall. Если нет, то SVC вызывает Hard Fault.
PendSV	Программный запрос обслуживания	Асинхронная	Прерывание обслуживания согласно приоритету
SysTick	Счётчик достиг нуля или установлен запрос обслуживания	Асинхронная	Прерывание обслуживания согласно приоритету
Hard Fault	любая ошибка	Синхронная или асинхронная[a]	Выше чем другие за исключением NMI[b]

a Активация зависит от причины вызвавшей ошибку
b Если Hard Fault происходит когда процессор выполняет обработчик NMI или Hard Fault, процессор входит в состояние lock-up.

Lock-up

Процессор имеет состояние lock-up, в которое входит когда случается неисправимая ситуация. Причины неисправимой ситуации могут быть синхронные или асинхронны ошибки, включая конфликтную SVC инструкцию.

Процессор может войти в состояние lock-up с приоритетом -1 или -2. NMI может быть причиной выхода процессора из состояния lock-up, если это будет приоритет -1. Отладчик также может быть причиной выхода процессора из состояния lock-up.

Периферийные блоки формируют прерывания с IRQ0 до IRQ31

Прерывания	Блок	Принцип формирования
IRQ0	MIL-STD-1553B2	Сигнал прерывания от контроллера интерфейса по ГОСТ Р52070-2003. Канал 2. Сигнал VALMESS, ERR, RFLAGN, IDLE.
IRQ1	MIL-STD-1553B1	Аналогично. Канал 1.
IRQ2	USB	Прерывания от USB Host при наличии соответствующих флагов разрешения HostSOFSent или HostConnEvent или HostResume или HostTransDone. Прерывания от USB Slave при наличии

		соответствующих флагов разрешения SlaveNAKSent или SlaveSOFRXed или SlaveResetEvent или SlaveResume или SlaveTransDone.
IRQ3	CAN1	Сигнал прерывания от блока CAN. Возникает при установленном бите GLB_INT_EN и при сигналах RX_INT_EN[31:0] и RX_INT[31:0] или TX_INT_EN[31:0] и EX_INT[31:0] или ERR_INT_EN и (ACKERR или FRAMEERR или CRCERR или BSERR или BITERR) или ERR_OVER_INT_EN и REC > CAN_ERR_MAX или TEC > CAN_ERR_MAX.
IRQ4	CAN2	Аналогично
IRQ5	DMA	Прерывания от DMA DMA_ERR или DMA_DONE. Обработка прерываний от DMA в соответствии с разделом Error signaling технического описания DMA.
IRQ6	UART1	Сигнал UARTINTR.
IRQ7	UART2	Сигнал UARTINTR.
IRQ8	SSP1	Сигнал SSPINTR.
IRQ9	BUSY	Сигнал знаятости от NAND флеш.
IRQ10	ARINC429R1-ARINC429R8	Сигнал перрывания от одного из приёмников ARINC-429. Сигналы DR, ERROR, FF, HF.
IRQ11	POWER	Сигнал прерывания от POWER Detecor.
IRQ12	WWDG	Сигнал прерывания от WWDG.
IRQ13	Timer4	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE.
IRQ14	Timer1	Аналогично.
IRQ15	Timer2	Аналогично.
IRQ16	Timer3	Аналогично.
IRQ17	ADC	Сигналы прерываний от АЦП EOCIF_1 или AWOIF_1 или EOCIF_2 или AWOIF_2.
IRQ18	Ethernet	Сигнал прерывания от контроллера интерфейса Ehternet.
IRQ19	SSP3	Сигнал SSPINTR.
IRQ20	SSP2	Сигнал SSPINTR.
IRQ21	ARINC429T1	Сигнал прерывания от передатчика по ГОСТ 18977-79. Сигналы FFT, HFT, TX_R.
IRQ22	ARINC429T2	Аналогично.
IRQ23	ARINC429T3	Аналогично.
IRQ24	ARINC429T4	Аналогично.
IRQ25…IRQ26	Зарезервировано	

IRQ27	BACKUP	Прерывание от BKP и часов реального времени.
IRQ28	Внешнее прерывание 1	Сигнал EXT_INT1 Вывод PC[5] в основном режиме.
IRQ29	Внешнее прерывание 2	Сигнал EXT_INT2 Вывод PC[6] в основном режиме.
IRQ30	Внешнее прерывание 3	Сигнал EXT_INT3 Вывод PC[7] в основном режиме.
IRQ31	Внешнее прерывание 4	Сигнал EXT_INT4 Вывод PC[8] в основном режиме.

Контроллер прерываний NVIC

NVIC (Nested Vectored Interrupt Controller) поддерживает прерывания, у которых может быть переопределён приоритет. NVIC и ядро процессора тесно связаны, что позволяет уменьшить задержки обработки прерываний и повысить эффективность обработки запаздывающих прерываний.

Все регистры NVIC доступны только при использовании словных транзакций. Любая попытка записать полуслово или индивидуальный байт вызывает порчу бит регистра. Регистры NVIC используют режим доступа little-endian. Доступ процессора к ним корректно обрабатывается, несмотря на конфигурацию endian процессора. Доступ DAP также должен быть интерпретирован как little-endian.

Программная модель NVIC

В этом разделе описываются регистры NVIC. Описание содержит:

- карту памяти NVIC;
- описание регистров NVIC.

Таблица карты памяти регистров

Имя регистра	Тип операции	Адрес	Значение после сброса	Описание
ISER	R/W	0xE000E100	0x00000000	Регистр разрешения прерываний
ICER	R/W	0xE000E180	0x00000000	Регистр запрета прерывания
ISPR	R/W	0xE000E200	0x00000000	Регистр перевода прерывания в состояние ожидания обслуживания
ICPR	R/W	0xE000E280	0x00000000	Регистр сброса состояния ожидания обслуживания
IPR0	R/W	0xE000E400	0x00000000	Регистр приоритета прерываний 0
IPR1	R/W	0xE000E404	0x00000000	Регистр приоритета прерываний 1
IPR2	R/W	0xE000E408	0x00000000	Регистр приоритета прерываний 2
IPR3	R/W	0xE000E40C	0x00000000	Регистр приоритета прерываний 3
IPR4	R/W	0xE000E410	0x00000000	Регистр приоритета прерываний 4
IPR5	R/W	0xE000E414	0x00000000	Регистр приоритета прерываний 5
IPR6	R/W	0xE000E418	0x00000000	Регистр приоритета прерываний 6
IPR7	R/W	0xE000E41C	0x00000000	Регистр приоритета прерываний 7

Регистр разрешения прерываний

Этот регистр используется для разрешения прерываний и определения, какие прерывания разрешены. Каждый бит этого регистра соответствует одному из 32-х прерываний. Установка бита в этом регистре разрешает соответствующее прерывание.

Когда бит разрешения обслуживания прерывания установлен, процессор активизирует прерывания на основе их приоритета. Когда бит разрешения очищен, установка сигнала запроса обслуживания прерывания не приводит к активации прерывания, несмотря на его приоритет. Следовательно, запрещенное прерывание может служить, как защёлка бита общего назначения. Вы можете прочитать этот бит или сбросить его не вызывая прерывания.

Сброс разрешения производится записью соответствующего бита Регистр запрета прерывания. Это также очищает соответствующий бит в Регистр разрешения прерываний.
Адрес регистра: 0xE000E100
Доступ: Чтение/запись
Значение после сброса: 0x00000000

Таблица назначения бит

Биты	Поле	Функция
31..0	SETENA	Биты разрешения прерывания. При записи: 1 – разрешение прерывания; 0 – не оказывает влияния. При чтении: 1 – прерывание разрешено; 0 – прерывание запрещено. Запись нуля в SETENA не оказывает влияние. Чтение бита возвращает текущее состояние разрешения прерывания. Сброс очищает поле SETENA.

Регистр запрета прерывания

Этот регистр используется для запрета прерываний и определения, какие прерывания разрешены. Каждый бит этого регистра соответствует одному из 32-х прерываний. Установка бита в этом регистре запрещает соответствующее прерывание.
Адрес регистра: 0xE000E180
Доступ: Чтение/запись
Значение после сброса: 0x00000000

Запись единицы в этот регистр не оказывает влияние на текущее активное прерывание, а только предотвращает новую активацию.

Таблица назначения бит

Биты	Поле	Функция
31..0	CLRENA	Биты запрещения прерывания. При записи: 1 – запрещает прерывание; 0 – не оказывает влияния. При чтении:

		1 – прерывание разрешено; 0 – прерывание запрещено. Запись нуля в CLRENA не оказывает влияние. Чтение бита возвращает текущее состояние разрешения. Сброс очищает поле CLRENA.

Регистр перевода прерывания в состояние ожидания обслуживания

Этот регистр используется для принудительно перевода прерываний в состояние ожидания обслуживания, а также определения какие прерывания находятся в этом состоянии.

Каждый бит этого регистра соответствует одному из 32-х прерываний. Установка бит этого регистра переводит в состояние ожидания обслуживания соответствующего прерывания. Запись нуля в этот регистр не оказывает влияние на состояние соответствующего прерывания.

Сброс бита перевода в состояние ожидания обслуживания производиться записью единицы в соответствующий бит Регистр сброса состояния ожидания обслуживания.

Запись в регистр перевода прерывания в состояние ожидания обслуживания не оказывает влияние на прерывание, которое обслуживается.

Адрес регистра: 0xE000E200
Доступ: Чтение/запись
Значение после сброса: 0x00000000

Таблица назначения бит

Биты	Поле	Функция
31..0	SETPEND	При записи: 1 – разрешение ожидания обслуживания; 0 – не оказывает влияния. При чтении: 1 – прерывание в состоянии ожидания обслуживания; 0 – прерывание не в состоянии ожидания обслуживания.

Регистр сброса состояния ожидания обслуживания

Этот регистр используется для сброса состояние ожидания обслуживания прерывания, а также определения какие прерывания находятся в состоянии ожидания обслуживания.

Каждый бит этого регистра соответствует одному из 32-х прерываний. Установка бит этого регистра сбрасывает состояние ожидания обслуживания соответствующего прерывания.

Запись в регистр сброса состояния ожидания обслуживания не оказывает влияние на прерывание, которое обслуживается.

Адрес регистра: 0xE000E280
Доступ: Чтение/запись
Значение после сброса: 0x00000000
Таблица назначения бит

Биты	Поле	Функция
31..0	CLRPEND	При записи:

		1 – сбрасывает состояние ожидания обслуживания; 0 – не оказывает влияния. При чтении: 1 – прерывание в состоянии ожидания обслуживания; 0 – прерывание не в состоянии ожидания обслуживания.

Регистр приоритета прерываний

Этот регистр используется для присвоения приоритета от 0 до 3 каждому из доступных прерываний. Ноль наивысший приоритет, а 3 самый низший. Два бита приоритета хранятся в битах [7:6] каждого байта.

Адрес регистра: 0xE000E400- 0xE000E41C
Доступ: Чтение/запись
Значение после сброса: 0x00000000

Таблица назначения бит

	31 30	29 24	23 22	21 16	15 14	13 8	7 6	5 0
E000E400	IP_3	Зарезерв.	IP_2	Зарезерв.	IP_1	Зарезерв.	IP_0	Зарезер.
E000E404	IP_7		IP_6		IP_5		IP_4	
E000E408	IP_11		IP_10		IP_9		IP_8	
E000E40C	IP_15		IP_14		IP_13		IP_12	
E000E410	IP_19		IP_18		IP_17		IP_16	
E000E414	IP_23		IP_22		IP_21		IP_20	
E000E418	IP_27		IP_26		IP_25		IP_24	
E000E41C	IP_31		IP_30		IP_29		IP_28	

Биты	Поле	Функция
7..6	IP_n	Приоритет прерывания n

Прерывания по уровню и по фронту

Процессор поддерживает два вида прерываний: по уровню и по фронту. Уровень сигнала удерживается установленным до тех пор, пока не будет сброшен программой обработки прерывания (ISR) устройства. Прерывание по фронту, заключается в том, что процессор сэмплирует линию прерывания по переднему фронту синхросигнала. Процессор распознаёт фронт, когда наблюдается низкий уровень сигнала и высокий уровень сигнала в двух его последовательных выборках по переднему фронту процессорного синхросигнала.

Для прерываний по уровню, если сигнал не снимается, прежде чем происходит возврат из обработчика прерывания, то прерывания повторно активизируется и ожидает обслуживания. Это часто применяется для FIFO и в устройствах, основанных на буферах, потому что гарантирует считывание информации либо одним ISR, либо повторным запросом без дополнительной работы. Это означает, что устройство удерживает сигнал прерывания установленным пока устройство пусто.

Прерывания по фронту должны быть установлены, по крайней мере, один процессорный такт процессора, чтобы NVIC отследил их.

Прерывания по фронту могут быть заново установлены во время ISR, поэтому прерывания могут быть активными и ожидать обслуживания в одно и тоже время. Должно гарантироваться условие, что второй фронт не придёт прежде, чем первый фронт,

вызвавший прерывание будет активирован. Если второй фронт придёт прежде, чем прерывание активируется, то второй фронт не окажет никакого воздействия, так как обслуживание запущено. Когда ISR активирован, бит ожидания обслуживания очищается. Если прерывание вызывается вновь, когда ISR активирован, NVIC защёлкивает опять бит ожидания обслуживания.

Прерывания по фронту используются в основном для внешних сигналов и для частотных или повторяющихся сигналов.

Повторная выборка уровня прерываний

ISR может детектировать, происходили или нет прерывания при обработке текущего прерывания, для того чтобы избежать потерь на вход и выход из обработчика прерывания. Эта информация доступна в регистрах ISPR и ICPR.

Для прерываний по фронту бит, который установился в единицу, показывает, что другое прерывание произошло с момента старта ISR.

Если прерывание по уровню было очищено, а затем установлено, то статусный бит, считанный из регистров ISPR и ICPR установленный в единицу, аналогичен ситуации с прерыванием по фронту.

Для прерываний по уровню, у которых линия сигнала может оставаться произвольно долго с момента входа в ISR необходимо записать соответствующий бит в ISPR или ICPR. Регистр ICPR не очиститься, если линия прерывания в состоянии единицы, и может быть считан повторно, чтобы определить статус.

Прерывания как входы общего назначения

Вы можете использовать линии прерываний как линии общего назначения. При таком использовании необходимо гарантировать, что прерывания запрещены в регистре ICER.

Вы можете использовать ICPR, чтобы проверить перешёл ли вход в состоянии единицы с момента последней проверки.

Для проверки текущего состояния, необходимо записать 1 в соответствующий бит ICPR. Значение статусного бита очиститься, если состояние линии ноль и ICPR может быть считан вновь, чтобы определить статус.

Блок управления системой ядра

Блок управления системой (SCB – System control block) обеспечивает доступ к информации о конфигурации и управление работой системы. Регистры блока управления системой представлены в таблице.

Имя регистра	Тип операции	Адрес	Значение после сброса
Вспомогательный управляющий регистр	R/W	0xE000E008	Нули в старших 28 битах, состояние вывода ITCMLAEN в бите [3]. Нули в младших 3 битах.
Регистр управления и статуса SysTick	R/W	0xE000E010	0x00000004
Регистр перегружаемого значения SysTick	R/W	0xE000E014	0x00000000
Регистр текущего значения SysTick	R/W очистка	0xE000E018	0x00000000
Регистр калибровочного значения SysTick	RO	0xE000E01C	0x80000000
Регистр CPUID	RO	0xE000ED00	0x411CC210
Регистр управления состоянием прерываний	a	0xE000ED04	0x00000000
Регистр управления прерываниями и программным сбросом	b	0xE000ED0C	0xFA050000[c] 0xFA058000[d]
Регистр конфигурации и управления	R/W	0xE000ED14	0x00000208
Регистр приоритета системного обработчика 2	R/W	0xE000ED1C	0x00000000
Регистр приоритета системного обработчика 3	R/W	0xE000ED20	0x00000000
Регистр управления и состояния системного обработчика	R/W	0xE000ED24	0x00000000

a Тип доступа зависит от конкретного бита.
b Тип доступа зависит от конкретного бита.
c Значение сброса для little-endian
d Значение сброса для big-endian

Все регистры системного контроля доступны только с использованием словных транзакций. Любая попытка записать полуслово или байт вызывает искажение бит регистра.

Вспомогательный управляющий регистр

Применяется для разрешения дополнительных верхнего и нижнего адресного пространства инструкций ITCM.

Адрес: 0xE000E008
Тип доступа: Чтение/запись
Значение после сброса: Верхние 28 бит нули, состояние вывода ITCMLAEN в бите [3], нули в трёх младших битах.

Назначение разрядов регистра

31 … 5	4	3	2 … 0
	ITCMUAEN	ITCMLAEN	

Информация о битах регистра

Биты	Поле	Функция
31..5	-	Зарезервировано
4	ITCMUAEN	Разрешение верхнего адресного пространства инструкций ITCM
3	ITCMLAEN	Разрешение нижнего адресного пространства инструкций ITCM
2..0	-	

Когда установлен бит ITCMLAEN, все допустимые инструкции и данные считываются с адресного пространства 0x00000000 – 0x0001FFFF (128КБ флеш-памяти на кристалле) через интерфейс ITCM. Когда бит ITCMLAEN очищен, этот доступ выполняется через внешний интерфейс AHB-Lite с адресным пространством 1 МБ.

Когда установлен бит ITCMUAEN, все допустимые инструкции и данные считываются с адресного пространства 0x10000000-0x1000FFFF через интерфейс ITCM. Когда бит ITCMUAEN сброшен, этот доступ выполняется через внешний интерфейс AHB-Lite.

Регистр управления и статуса SysTick

Адрес: 0xE000E010
Тип доступа: Чтение/запись
Значение после сброса: 0x00000004

Назначение разрядов регистра

31 … 17	16	15 … 3	2	1	0
Зарезервировано	COUNTFLAG	Зарезервировано	CLKSOURCE	TICKINT	ENABLE

Информация о битах регистра

Биты	Поле	Функция
31..17	-	Зарезервировано
16	COUNTFLAG	Возвращает 1, если таймер досчитал до нуля с последнего момента чтения. Очищается при чтении приложением или отладчиком.
15..3	-	Зарезервировано
2	CLKSOURCE	Всегда читается как единица: 1 – синхросигнал процессора. Признак того, что SysTick использует процессорный синхросигнал HCLK.
1	TCKINT	Бит разрешения прерывания от системного таймера: 0 – если таймер досчитал до нуля, то прерывание не возникает; 1 – если таймер досчитал до нуля, то возникает запрос на прерывание. Программное обеспечение может использовать бит COUNTFLAG, чтобы определить досчитал таймер до нуля или нет.
0	ENABLE	Разрешение работы таймера: 1 – работа счётчика разрешена. Это означает, что счётчик

		загружает значение Reload и начинает считать вниз. При достижении нуля, устанавливается флаг COUNTFLAG в единицу, и дополнительно, в зависимости от TCKINT, формируется запрос на обслуживание прерывания (SysTick_Handler) от системного таймера. Затем загружается значение Reload и опять начинается счёт; 0 – счётчик отключён.

Регистр перегружаемого значения SysTick

Регистр используется для определения стартового значения, загружаемого в регистр текущего значения SysTick, когда счётчик достигает нуля. Значение Reload может быть любым в диапазоне 0x00000001-0x00FFFFFF. Значение 0 допустимо, но не оказывает эффекта, потому что запрос на прерывание и установка бита COUNTFLAG происходит только при переходе таймера из состояния 1 в 0.

Расчёт значения Reload происходит в соответствии с использованием таймера:

- Для формирования короткого интервала времени с периодом N процессорных тактов, применяется значение RELOAD равное N-1. Например, если требуется прерывание каждые 100 циклов, то устанавливается значение RELOAD равное 99.
- Для формирования одиночного прерывания после задержки в N тактов процессора, используется значение N. Например, если требуется прерывание после 400 тактов процессора, то устанавливается RELOAD равное 400.

Адрес: 0xE000E014
Тип доступа: Чтение/запись
Значение после сброса: 0x00000000

Назначение разрядов регистра

31 24	23 0
Зарезервировано	Reload

Информация о битах регистра

Биты	Поле	Функция
31..24	-	Зарезервировано
23..0	Reload	Значение, загружаемое в регистр текущего значения SysTick, когда счётчик достигает нуля.

Регистр текущего значения SysTick

Используется для определения текущего значения таймера SysTick.

Адрес: 0xE000E018
Тип доступа: Чтение/запись очистка
Значение после сброса: 0x00000000

Назначение разрядов регистра

31 24	23 0
Зарезервировано	Current

Информация о битах регистра

Биты	Поле	Функция
31..24	-	Зарезервировано
23..0	Current	Чтение возвращает текущее значение системного таймера. Запись любого значения очищает регистр в ноль, и также очищает бит COUNTFLAG регистра управления и статуса SysTick.

Регистр калибровочного значения SysTick

Регистр используется программным обеспечением для масштабирования до любой желаемой скорости, используя деление и умножение.

Адрес: 0xE000E01C
Тип доступа: Чтение
Значение после сброса: 0x80000000

Назначение разрядов регистра

31	30	29 24	23 0
NOREF	SKEW	Зарезервировано	TENMS

Информация о битах регистра

Биты	Поле	Функция
31	NOREF	Читается как единица. Показывает, что отдельный синхросигнал таймера не поддерживается.
30	SKEW	Читается как ноль. Калибровочное значение с неточностью синхронизации 10 мс, так как неизвестно TENMS.
23..0	TENMS	Читается как ноль. Показывает, что калибровочное значение неизвестно.

Регистр CPUID

При чтении регистра можно определить:

- Номер ID процессорного ядра;
- Номер версии процессорного ядра;
- Подробности реализации ядра.

Адрес: 0xE000ED00
Тип доступа: Чтение
Значение после сброса: 0x410CC210

Назначение разрядов регистра

31 24	23 20	19 16	15 4	3 0
IMPLEMENTER	VARIANT	Constant	PARTNO	REVISION

Информация о битах регистра

Биты	Поле	Функция
31..24	IMPLEMENTER	Код производителя: 0x41 – ARM.
23..20	VARIANT	Исполнения определяет номер варианта: 0x0 – для r0p0 и r0p1;

		0x1 – для r1p0.
19..6	Constant	Читается как 0xC.
15..4	PARTNO	Номер процессора в пределах семейства: 0xC21.
3..0	REVISION	Исполнение определяет номер ревизии: 0x0 – для r0p0 и r1p0; 0x1 – для r0p1.

Регистр управления состоянием прерываний

Регистр используется для:

- установки состояния ожидания обслуживания NMI;
- установки или сброса состояния ожидания обслуживания для PendSV;
- установки или сброса состояния ожидания обслуживания для SysTick;
- проверки состояния ожидания обслуживания для исключений;
- определения номера вектора исключения наивысшего приоритета, ожидающего обслуживания;
- определения номера вектора активного исключения.

Адрес: 0xE000ED04
Тип доступа: Зависит от индивидуальных бит
Значение после сброса: 0x00000000

Назначение разрядов регистра

31	30	29 28	27	26	25
NMIPENDSET	-	PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR

24	23	22	21 18	17 12	11 6	5 0
-	ISRPREEMPT	ISRPENDING	-	VECTPENDING	-	VECTACTIVE

Информация о битах регистра

Биты	Поле	Тип доступа	Функция
31	NMIPENDSET	R/W	При записи: 1 – устанавливает запрос обслуживания для NMI; 0 – не оказывает влияния. NMIPENDSET ожидает обслуживания и активирует NMI. Так как NMI имеет наивысший приоритет, то его обслуживание начинается сразу, как только обнаружено, за исключением случая, когда процессор имеет приоритет -2. При чтении возвращает информацию о состоянии ожидания обслуживания NMI.
30..29	-		Зарезервировано
28	PENDSVSET	R/W	Бит установки состояния ожидания обслуживания для исключения PendSV. При записи: 0 – не влияет на работу системы;

			1 – переводит исключение PendSV в состояние ожидания обслуживания. При чтении: 0 – исключение PendSV не ожидает обслуживания; 1 – исключение PendSV ожидает обслуживания. Запись 1 в разряд PENDSVSET это единственный возможный способ перевода исключения PendSV в состояние ожидания обслуживания.
27	PENDSVCLR	WO	Бит сброса состояния ожидания обслуживания для исключения PendSV. При записи: 0 – не влияет на работу системы; 1 – сбрасывает состояние ожидания обслуживания для исключения PendSV.
26	PENDSTSET	R/W	Бит установки состояния ожидания обслуживания для исключения SysTick. При записи: 0 – не влияет на работу системы; 1 – переводит исключение SysTick в состояние ожидания обслуживания. При чтении: 0 – исключение SysTick не ожидает обслуживания; 1 – ожидает.
25	PENDSTCLR	WO	Бит сброса состояния ожидания обслуживания для исключения SysTick. При записи: 0 – не влияет на работу системыя; 1 – сбрасывает состояние ожидания обслуживания для исключения SysTick.
24	-		Зарезервировано
23	ISRPREEMPT[a]	RO	Этот бит используется в режиме отладки. Бит сигнализирует, что прерывание, ожидающее обслуживания, станет активным при запуске следующего цикла. Если бит C_MASKINTS очищен в регистре управления и статуса отладочного режима останова, то прерывание обслуживается следующим образом: 1 – ожидающее обслуживание исключение обслуживается при выходе из состояния останова режима отладки; 0 – ожидающее обслуживание исключение не обслуживается.
22	ISRPENDING[a]	RO	Флаг наличия в системе прерываний, ожидающих обслуживания. 0 – ожидающие обслуживания прерывания отсутствуют; 1 – присутствуют.
21..18	-		Зарезервировано
17..12	VECTPENDING	RO	Содержит номер исключения, ожидающего обслуживания, с наивысшим приоритетом,

			обработка которого в системе разрешена. 0 – необслуженных исключений нет; 5'bXXXXXX – номер ожидающего обслуживания исключения. Значение данного поля не учитывает влияние поля PRIMASK.
11..6	-		Зарезервировано
5..0	VECTACTIVE[b]	RO	Содержит номер активного исключения. 0 – Thread режим; 6'bXXXXXX – номер[b] текущего обслуживаемого исключения.

a Только для режима отладки;
b Это значение аналогично битам [5:0] регистра IPSR.

Регистр управления прерываниями и программным сбросом

Этот регистр используется для:

- определения порядка следования байт в слове (endianness) при доступе к данным;
- очистка всей информации об активных состояниях из отладочного режима останова;
- запроса сброса системы.

Адрес: 0xE000ED0C
Тип доступа: Зависит от индивидуальных бит
Значение после сброса: 0xFA050000 в случае режима данных little-endian
0xFA058000 в случае режима данных big-endian

Назначение разрядов регистра

31 16	15	14 3	2	1	0
VECTKEY	ENDIANNESS	-	SYSRESETREQ	VECTCLRACTIVE	-

Информация о битах регистра

Биты	Поле	Тип доступа	Функция
31..16	VECTKEY	WO	Ключ доступа к регистру. При записи должен быть равен 0x05FA, в противном случае попытка записи в регистр будет проигнорирована процессором.
15	ENDIANNESS	RO	Порядок следования значащих разрядов при доступе к данным. 0 – младший байт идет первым (little-endian); 1 – старший байт идет первым (big-endian).
14..3	-		Зарезервировано
2	SYSRESETREQ	WO	Запись единицы в этот бит вызовет установку сигнала SYSRESETREQ на выходе системы для запроса сброса. В результате произойдёт сброс всей системы и основных компонентов за исключением отладочных. Бит C_HALT в регистре DHCSR очиститься как результат запроса на сброс системы. Но отладчик не

			потеряет связь с устройством.
1	VECTCLRACTIVE	WO	Очищает всю информацию об активных состояниях фиксированных и конфигурируемых исключениях. Этот бит: - самоочищающийся; - может быть установлен только DAP в режиме останова процессора. Когда этот бит установлен: - очищается статус всех активных исключений процессора; - принудительный возврат в режим Thread; - принудительная установка IPSR в ноль.
0	-		Зарезервировано

Регистр конфигурации и управления

Это регистр используется для разрешения выравнивания стека и служит причиной Hard Fault в случае невыровненного доступа.

Адрес: 0xE000ED14
Тип доступа: Чтение
Значение после сброса: 0x00000208

Назначение разрядов регистра

31 10	9	8 4	3	2 0
-	STKALIGN	-	UNALIGN_TRP	-

Информация о битах регистра

Биты	Поле	Функция
31..10	-	Зарезервировано
9	STKALIGN	Всегда в единице. Вход в любое исключение происходит с 8 байтовым выравниванием стека и это сохраняется при восстановлении контекста. SP восстанавливается при соответствующем возвращении из исключения.
8..4	-	Зарезервировано
3	UNALIGN_TRP	Показывает, что любой невыровненный доступ приводит к Hard Fault. Ловушка для невыровненного доступа устанавливается в 1.
2..0	-	Зарезервировано

Регистры приоритета системных обработчиков

Системные обработчики это специальный класс обработчиков исключений, которые могут иметь свой приоритет, установленный в любое значение из уровней приоритета.

Существует два регистра приоритета системных обработчиков для задания приоритета следующим системным обработчикам:

- SVCall;

- SysTick;
- PendSV

PendSV и SVCall постоянно разрешены. Вы можете разрешить или запретить SysTick записью в регистр управления и статуса SysTick.

Регистр приоритета системного обработчика 2

Адрес: 0xE000ED1C
Тип доступа: Чтение/Запись
Значение после сброса: 0x00000000

Назначение разрядов регистра

31 30	29 0
PRI_11	-

Информация о битах регистра

Биты	Поле	Функция
31..30	PRI_11	Приоритет системного обработчика 11, SVCall
29..0	-	Зарезервировано

Регистр приоритета системного обработчика 3

Адрес: 0xE000ED20
Тип доступа: Чтение/Запись
Значение после сброса: 0x00000000

Назначение разрядов регистра

31 30	29 24	23 22	21 0
PRI_15	-	PRI_14	-

Информация о битах регистра

Биты	Поле	Функция
31..30	PRI_15	Приоритет системного обработчика 15, SysTick
29..24	-	Зарезервировано
23..22	PRI_14	Приоритет системного обработчика 14, PendSV
21..0	-	Зарезервировано

Регистр управления и состояния системного обработчика

Этот регистр используется для чтения или записи статуса ожидания обслуживания исключения SVCall.

Адрес: 0xE000ED24
Тип доступа: Чтение/Запись
Значение после сброса: 0x00000000

Назначение разрядов регистра

31 16	15	14 0
-	SVCALLPENDED	-

Информация о битах регистра

Биты	Поле	Функция
31..16	-	Зарезервировано
15	SVCALLPENDED	Читается как 1, если SVCall ожидает обслуживания. При записи: 1 – установка состояния ожидания обслуживания SVCall; 0 – сброс состояния ожидания обслуживания для SVCall.
14..0	-	Зарезервировано

Этот регистр доступен только как часть отладки и не доступен через карту памяти.

Сторожевые таймеры

Описание регистров блока сторожевых таймеров

Базовый Адрес	Название	Описание
0x4006_8000	IWDG	Сторожевой таймер IWDG
Смещение		
0x00	IWDG_KR[15:0]	Регистр Ключа
0x04	IWDG_PR[2:0]	Делитель частоты сторожевого таймера
0x08	IWDG_PRL[11:0]	Регистр основания счета сторожевого таймера
0x0C	IWDG_SR[1:0]	Регистр статуса сторожевого таймера

Базовый Адрес	Название	Описание
0x4006_0000	WWDG	Сторожевой таймер WWDG
Смещение		
0x00	WWDG_CR[7:0]	Регистр управления
0x04	WWDG_CFR[9:0]	Регистр конфигурации
0x08	WWDG_SR[0]	Регистр статуса

IWDG_KR

Номер	15	0
Доступ *	W	W
Значение после сброса	0	0
	KEY[15:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..16		Зарезервировано
15..0	KEY[15:0]	**Значение ключа (только запись, читается 0000h).** Эти биты должны перезаписываться программно через определённые интервалы ключевым значением AAAAh, в противном случае сторожевой таймер генерирует сброс, если таймер достиг значения нуля. Запись ключевого значения 5555h разрешает доступ по записи к регистрам IWDG_PR и IWDG_RLR. Запись ключевого значения CCCCh разрешает работу сторожевого таймера (за исключением, если сторожевой таймер уже разрешён аппаратно битами конфигурации).

IWDG_PR

Номер	7	6	5	4	3	2	1	0
Доступ *	U	U	U	U	U	R/W	R/W	R/W
Значение после сброса						0	0	0
	-	-	-	-	-	PR2	PR1	PR0

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..3		Зарезервировано
2..0	PR[2:0]	**Делитель частоты сторожевого таймера.** 3'b000 – делитель на 4; 3'b001 – делитель на 8; 3'b010 – делитель на 16; 3'b011 – делитель на 32; 3'b100 – делитель на 64; 3'b101 – делитель на 128; 3'b110 – делитель на 256; 3'b111 – делитель на 256. Чтение и запись этого регистра правомерна только, если бит PVU=0 в регистре IWDG_SR.

IWDG_RLR

Номер	11	0
Доступ *	R/W	R/W
Значение после сброса	1	1
	RLR[11:0]	

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..12		Зарезервировано
11..0	RLR[11:0]	**Значение перезагрузки сторожевого таймера.** Значение этих битов по доступу защищено с помощью регистра IWDG_KR. Эти биты записываются программно и определяют значение, загружаемое в сторожевой таймер в момент записи значение AAAAh в регистр IWDG_KR. Сторожевой таймер декрементируется, начиная с этого значения. Период таймаута сторожевого таймера – функция от этого значения и делителя частоты. Чтение и запись этого регистра правомерна только, если бит RVU=0 в регистре IWDG_SR.

IWDG_SR

Номер	7	6	5	4	3	2	1	0
Доступ *	U	U	U	U	U	U	R	R
Значение после сброса							0	0
	-	-	-	-	-	-	RVU	PVU

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..2		Зарезервировано
1	RVU	**Флаг обновления значения сторожевого таймера.** Этот бит устанавливается аппаратно и служит признаком того, что обновляется значение сторожевого таймера из регистра перезагрузки. Этот бит сбрасывается, если обновление завершено. Значение регистра перезагрузки может быть обновлено только, если этот бит равен нулю.
0	PVU	**Флаг обновления делителя частоты сторожевого таймера.** Этот бит устанавливается аппаратно и служит признаком того, что обновляется значение делителя частоты сторожевого таймера. Этот бит сбрасывается, если обновление завершено. Значение регистра делителя частоты может быть обновлено только, если этот бит равен нулю.

WWDG_CR

Номер	7	6	5	4	3	2	1	0
Доступ*	R/S	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значение после сброса	0	1	1	1	1	1	1	1
	WDGA	T6	T5	T4	T3	T2	T1	T0

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..18		Зарезервировано
7	WDGA	**Бит активации.** Этот бит устанавливается программно и очищается только аппаратно при сбросе. Когда WDGA=1, сторожевой таймер может генерировать сброс. 0 – сторожевой таймер отключен; 1 – сторожевой таймер включен.
6..0	T[6:0]	**Значение семиразрядного счётчика (от старших разрядов к младшим).** Эти биты содержат значение сторожевого таймера, который декрементируется каждые 4096×2^{WDGTB} циклов частоты PCLK периферийной шины APB.

WWDG_CFR

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
Значение после сброса	0	1	1	1	1	1	1	1
	WDGTB0	W6	W5	W4	W3	W2	W1	W0

Номер	15	14	13	12	11	10	9	8
Доступ*	U	U	U	U	U	U	R/S	R/W
Значение после сброса							0	0
	-	-	-	-	-	-	EWI	WDGTB1

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..10		Зарезервировано
9	EWI	**Раннее предупреждающее прерывание.** Если бит установлен, то разрешается генерация прерывания при достижении сторожевым таймером значении 40h. Прерывание запрещается только аппаратным сбросом.
8..7	WGTB[1:0]	**Делитель частоты сторожевого таймера.** 2'b00 – частота таймера (PCLK / 4096) /1; 2'b01 – частота таймера (PCLK / 4096) /2; 2'b10 – частота таймера (PCLK / 4096) /4; 2'b11 – частота таймера (PCLK / 4096) /8.
6..0	W[6:0]	**Значение окна.** Эти биты содержат значение окна, в пределах которого возможна инициализация битов T[6:0] значением в пределах 40h – 7Fh. Если происходит инициализация битов в момент T>W, то формируется сброс на выходе RESET. Если таймер достигнет значения T=3Fh, то также формируется сброс.

WWDG_SR

Номер	7	6	5	4	3	2	1	0
Доступ *	U	U	U	U	U	U	U	R/C
Значение после сброса								0
	-	-	-	-	-	-	-	EWIF

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..1		Зарезервировано
0	EWIF	**Флаг раннего предупреждающего прерывания.** Этот бит устанавливается аппаратно, когда сторожевой таймер достигает значения 40h. Бит очищается программно записью нуля. Запись единицы не влияет. Этот бит также устанавливается, если прерывание запрещено EWI=0.

Предельно-допустимые характеристики микросхемы

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение источника питания, В	U_{CC}	3,0	3,6	–	4,0
Напряжение источника питания АЦП, и ЦАП, В,	U_{CCA}	3,0	3,6	–	4,0
Напряжение источника питания батарейного домена, В	U_{CCB}	1,8	3,6	–	4,0
Входное напряжение низкого уровня, В, (при работе в цифровом режиме) на выводах: PA, PB, PC, PD, PE, PF, RESET, WAKEUP, DN, DP	U_{IL}	0	0,8	минус 0,3	–
на выводе: OSC_IN BYPASS=1	U_{IL_BHSE}	0	0,8	минус 0,3	–
Входное напряжение высокого уровня, В, на выводах: PD (7-15), PE (0-2,6-7), DN, DP	U_{IH}	2,0	U_{CC}	–	U_{CC} +0.3
на выводах: PA, PB, PC, PD(0-6), PE (3-5,8-15), PF, RESET, WAKEUP		2,0	5,25	–	5,3
на выводе: OSC_IN BYPASS=1	U_{IH_BHSE}	2,0	U_{CC}	–	U_{CC} +0.3
Выходной ток низкого уровня, мА, (при работе в цифровом режиме) на выводах: PA, PB, PC, PD, PE, PF, DN, DP	I_{OL}	минус 6	–	минус 10	–
Выходной ток высокого уровня, мА, на выводах: PA, PB, PC, PD, PE (0-5, 8-15), PF, DN, DP	I_{OH}	–	6	–	10
на выводах: PE 6, 7		–	3	–	10
Частота следования импульсов тактовых сигналов, МГц	f_C	–	140	–	–
Частота следования импульсов тактовых сигналов АЦП, МГц	f_{C_ADC}	–	14	–	–
Частота следования импульсов тактовых сигналов HSE, МГц при: BYPASS=0	f_{C_HSE}	2	16	–	–
при: BYPASS=1		–	140	–	–
Частота следования импульсов тактовых сигналов LSE, кГц при: BYPASS=0	f_{C_LSE}	32	33	–	–
при: BYPASS=1		-	1 000	–	–
Частота следования импульсов тактовых сигналов PLL, МГц	f_{C_PLL}	2	16	–	–

Параметры ЦАП

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение верхней границы опорного напряжения ЦАП, В, на выводе: DACx_REF при: Cfg_M_REFx=1	$U_{REF(DACx)}$	2,4	U_{CCA}	–	–
Резистивная нагрузка ЦАП, кОм	R_{LOAD}	10	–	–	–
Емкостная нагрузка ЦАП, пФ	C_{LOAD}	–	100	–	–
Параметры АЦП					
Напряжение нижней границы внешнего опорного напряжения АЦП, В, при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	U_{ADC1_REF-}	0	U_{CCA}-2,4	–	4,0
Напряжение верхней границы внешнего опорного напряжения АЦП,В при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	U_{ADC0_REF+}	2,4	U_{CCA}	–	4,0
Диапазон напряжения внешнего опорного источника АЦП, В, $U_{REF(ADC)}= U_{ADC0\_REF+} – U_{ADC1\_REF-}$	$U_{REF(ADC)}$	2,4	U_{CCA}	–	–
Диапазон напряжения на входе** АЦП,В	U_{AIN}	U_{ADC1_REF-}	U_{ADC0_REF+}	минус 0,3	4,0
Емкость нагрузки, пФ, на выводах: PA, PB, PC	C_L	–	30	–	–
Время хранения информации, лет, при: T=25 °C	t_{GS}	25	–	–	–
при: T=85 °C		10	–	–	–
при: T=125 °C		1	–	–	–

Допускается использование отдельного источника для питания АЦП и ЦАП, но при этом его выходное напряжение не должно отличаться от Ucc более чем на ±0,2В.

При использовании внутреннего опорного напряжения U_{ADC1_REF-} = AGND и U_{ADC0_REF+} = Ucc_A.

Примечание – Не допускается одновременное воздействие двух и более предельных режимов.

Электрические параметры микросхемы

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Выходное напряжение низкого уровня, В, на выводах: PA, PB, PC, PD, PE, PF, DN, DP	U_{OL}	–	0,4	25, 125, минус 60
Выходное напряжение высокого уровня, В, на выводах: PA, PB, PC, PD, PE (0-5, 8-15), PF, DN, DP при: I_{OH}= минус 6,0 мА	U_{OH}	2,4	–	25, 125, минус 60
на выводах: PE 6, 7 при: I_{OH}= 3,0 мА		2,4	–	
Уровень напряжения срабатывания схемы формирования сигнала общего сброса, В	U_{BOR}	1,8	2,1	25, 125, минус 60
Статический ток потребления в режиме покоя (регулятор напряжения выключен), мА	I_{CCS}	–	15	25, 125 минус 60
Динамический ток потребления, мА	I_{OCC}	–	300	25, 125, минус 60
Входной ток утечки высокого уровня, мкА, на выводах: PA, PB, PC, PD, PE, PF, DN, DP при: U_I= 3,6 В	I_{ILH}	минус 1	1	25, 125, минус 60
на выводах: PA, PB, PC, PD(0-6), PE (3-5,8-15), PF, RESET, WAKEUP при: U_I= 5,25 В		минус 1	1	25, 125, минус 60
на выводе: OSC_IN при: BYPASS=1, U_I= 3,6 В	I_{ILH_BHSE}	минус 40	40	25, 125, минус 60
Входной ток утечки низкого уровня, мкА, на выводах: PA, PB, PC, PD, PE, PF, RESET, WAKEUP, DN, DP при: U_I= 0 В	I_{ILL}	минус 1	1	25, 125, минус 60
на выводе: OSC_IN при: BYPASS=1, при: U_I= 0 В	I_{ILL_BHSE}	минус 1	1	25, 125, минус 60
Выходная частота LSI RC-генератора, кГц	f_{O_LSI}	10	60	25, 125, минус 60
Выходная частота HSI RC-генератора, МГц	f_{O_HSI}	6	10	25, 125, минус 60
Выходная частота PLL, МГц	f_{Omax_PLL}	2	140	25, 125, минус 60
Выходная минимальная частота PLL, МГц	f_{Omin_PLL}	–	2	25, 125, минус 60

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Параметры АЦП				
Разрядность АЦП	E_{NADC_S}	12	–	25, 125, минус 60
Дифференциальная нелинейность, единица младшего разряда	E_{DLADC_S}	минус 1	2	25, 125, минус 60
Интегральная нелинейность, единица младшего разряда	E_{ILADC_S}	минус 3	3	25, 125 минус 60
Ошибка смещения, единица младшего разряда	E_{OFFADC_S}	минус 6	6	25, 125, минус 60
Ошибка усиления, %	$E_{GAINADC}$	минус 1	1	25, 125, минус 60
Параметры ЦАП *(2 преобразователя)*				
Разрядность ЦАП	E_{NDAC}	12	–	25, 125, минус 60
Дифференциальная нелинейность ЦАП, единица младшего разряда	E_{DLDAC}	минус 1	2	25, 125, минус 60
Интегральная нелинейность ЦАП, единица младшего разряда	E_{ILDAC}	минус 6	6	25, 125, минус 60
Ошибка смещения ЦАП, мВ	E_{OFFDAC}	минус 40	40	25, 125, минус 60
Ошибка усиления ЦАП, %	$E_{GAINDAC}$	минус 2	2	25, 125, минус 60
Минимальное выходное напряжение ЦАП, мВ	$U_{O\_DAC\ min}$	–	0,08	25, 125, минус 60
Максимальное выходное напряжение ЦАП, мВ	$U_{O\_DAC\ max}$	$U_{REF(DAC)}$-0,08	–	25, 125, минус 60

Справочные данные

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Выходное напряжение регулятора LDO, В, при: I_{OL}= 80 мА	U_{O_LDO}	1,62	1,98	25, 125, минус 60
Длительность фронта/спада входного сигнала, нс, на выводе: OSC_IN при: HSE BYPASS=1	τ_r τ_f	–	3,5	
Параметры ЦАП				
Время установления сигнала ЦАП, мкс, при: U_{CC} = 3,6 В, C_l = 100 пФ, R_l = 10 кОм	$t_{SU(DAC)}$	–	5,2	
Время включения ЦАП, мкс, при: U_{CC} = 2,4 В	t_{ON_DAC}	–	10	
Ток потребления по входу опоры, мкА, при: Cfg_M_REF0 = 1	I_{DAC1_UREF}	–	500	
Ток потребления по входу опоры, мкА, при: Cfg_M_REF1 = 1	I_{DAC2_UREF}	–	500	
Ток потребления ЦАП, мА, при отсутствии нагрузки	I_{OCCDAC}	–	2	
Параметры АЦП				
Время выборки заряда АЦП, нс	t_{A_ADC}	–	$4 \bullet f_{C\_ADC}$	
Время преобразования АЦП, нс	t_{AO_ADC}	–	$28 \bullet f_{C\_ADC}$	
Ток потребления по входу внешней верхней границы опоры АЦП, мкА при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	I_{ADC0_VREF+}	–	50	
Ток потребления по входу внешней нижней опоры опоры АЦП, мкА при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	I_{ADC0_VREF-}	минус 50	–	
Ток потребления по питанию АЦП, мА при: fadc=14 МГц, U_{CCA}=3,6В	I_{OCCADC}	–	3	
Минимальная частота преобразования АЦП, кГц	f_{C_ADCMIN}	10	–	
Параметры PHY Ethernet				
Отношение сигнал/шум АЦП, дБ	$N_{N\_RF(PHY)}$	28	–	
Дифференциальная нелинейность АЦП, единица младшего разряда	$E_{DLDAC(PHY)}$	минус 1,6	1,6	
Интегральная нелинейность ЦАП, единица младшего разряда	$E_{ILDAC(PHY)}$	минус 1	2	
Выходное дифференциальное напряжение передатчика, В, при: скорость передачи данных 100 Мбит	$U_{OD(PHY)}$	0,8	–	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Выходное дифференциальное напряжение передатчика, В, при: скорость передачи данных 10 Мбит	$U_{OD1(PHY)}$	4,4	5,6	25, 125, минус 60
Время задержки распространения сигнала приемника при включении\ выключении, нс, при: скорость передачи данных 100 Мбит	$t_{PLH\_R(PHY)}$ $t_{PHL\_R(PHY)}$	2	5	
Разность задержек распространения сигнала приемника, нс, $\lvert t_{PLH\ R} - t_{PHL\ R}\rvert$ при: скорость передачи данных 100 Мбит	$t_{SKEW\_R(PHY)}$	–	0,5	
Искажение рабочего цикла передатчика, нс, при: скорость передачи данных 100 Мбит	$\Delta CY_{R(PHY)}$	–	0,5	
Частота следования импульсов тактовых сигналов PLL, МГц при: скорость передачи данных 10 Мбит	$f_{C\_PLL(PHY)}$	124,875	125,125	
Частота следования импульсов тактовых сигналов PLL, МГц при: скорость передачи данных 10 Мбит	$f_{C\_PLL1(PHY)}$	19,99	20,01	
Дрожание фронта тактового сигнала PLL, нс при: скорость передачи данных 100 Мбит	$\Delta\tau_{\_PLL(PHY)}$	–	1,4	
Дрожание фронта тактового сигнала PLL, нс при: скорость передачи данных 10 Мбит	$\Delta\tau_{\_PLL1(PHY)}$	минус 3,5	3,5	
Выходное напряжение на выводе EXRES1, В	$U_{O(PHY)}$	1,14	1,34	

Габаритный чертеж микросхемы

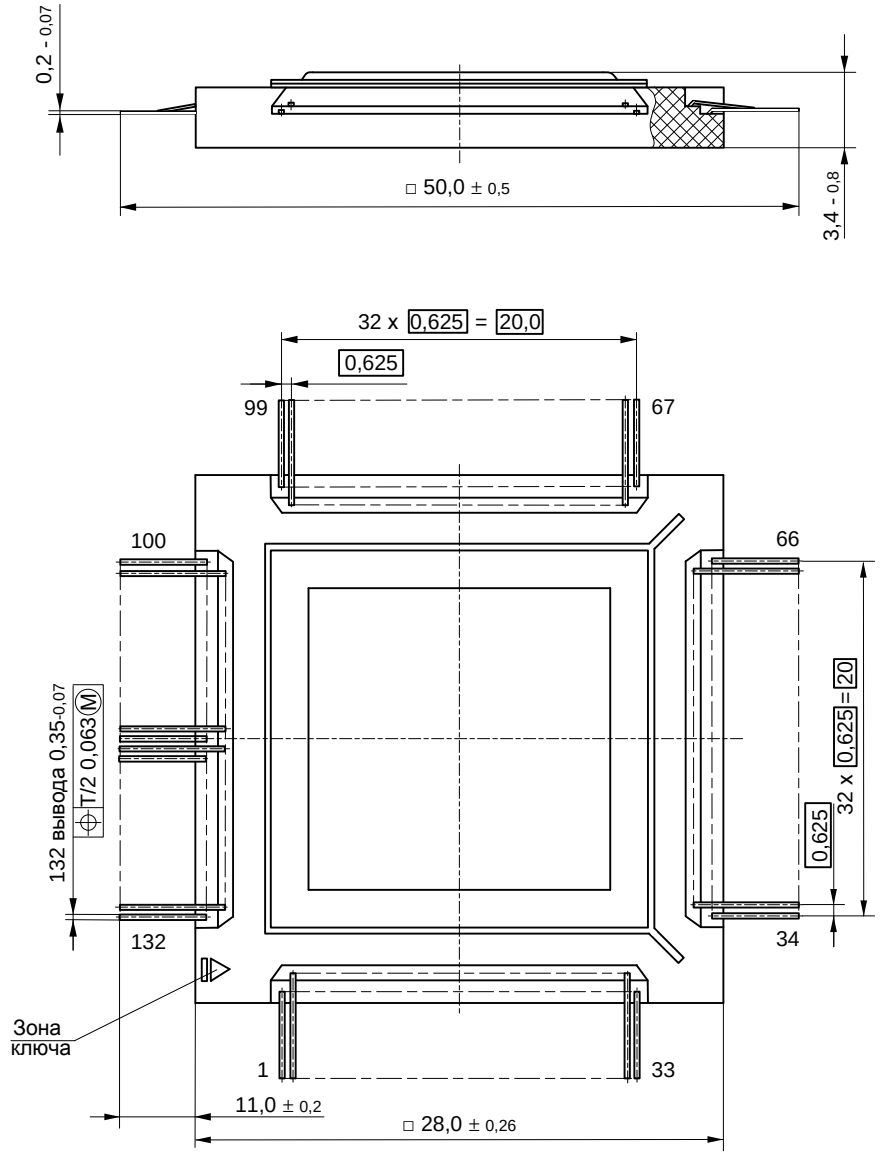


Рисунок – Корпус 4229.132-3

Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон
1986BE1T	1986BE1T	4229.132	минус 60 – 125 °С
K1986BE1T	K1986BE1T	4229.132	минус 60 – 125 °С
K1986BE1TK	K1986BE1T•	4229.132	0 – 70 °С

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	23.12.2011	1.0.0		
2	13.09.2012	1.1.0	Таблицы параметров	
3	12.10.2012	1.1.1	Приведение в соответствие с ТУ	