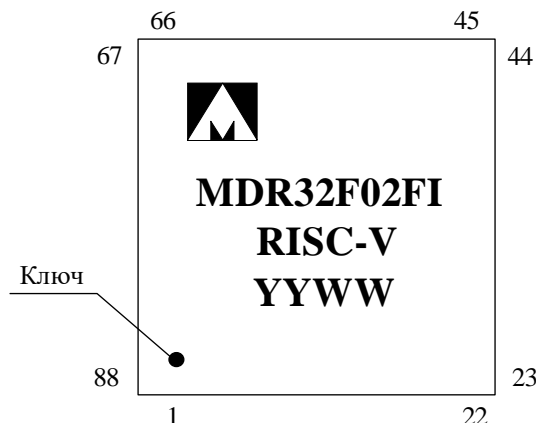




**Микросхема микроконтроллера  
для трехфазного и однофазного электросчетчиков  
K1986BK025**

**Основные характеристики микросхемы:**

- Напряжение источника питания от 2,2 до 3,6 В;
- 32-разрядная RISC-V архитектура BM-310S с системой команд RV32IMC;
- Встроенная память FLASH 256 + 8 Кбайт;
- Встроенная память RAM 112 Кбайт;
- Встроенная память OTP 16 Кбайт;
- Рабочий диапазон температур от минус 50 °С до плюс 85 °С.

YY – год выпуска

WW – неделя выпуска

**Тип корпуса:**

- 88-выводной пластмассовый корпус QFN88 (TICP).

Масса микросхем не более 0,35 г.

**Общее описание и область применения микросхемы**

Микросхемы интегральные K1986BK025 (далее – микросхемы) представляют собой микроконтроллеры со встроенной FLASH-памятью программ и построены на базе процессорного RISC-V ядра BM-310S.

**Важно:** спецификация действительна совместно с документом K1986BK025 Errata Notice.

## Основные характеристики:

### Ядро:

- 32-битное RISC-V ядро BM-310S с системой команд RV32IMC и тактовой частотой до 60 МГц;
- умножение за два цикла.

### Память:

- загрузочное OTP размером 16 Кбайт;
- встроенная энергонезависимая память программ FLASH типа размером 256 Кбайт (основная область) + 8 Кбайт (информационная область);
- встроенное ОЗУ размером 112 Кбайт.

### Питание и тактовая частота:

- внешнее питания 3,0...3,6 В;
- встроенный регулятор напряжения для питания ядра;
- встроенные схемы контроля питания;
- встроенный домен с батарейным питанием;
- встроенный подстраиваемый RC-генератор 8 МГц;
- встроенный подстраиваемый RC-генератор 32 кГц;
- внешний осциллятор 8...16 МГц;
- внешний осциллятор 32 кГц;
- встроенный умножитель тактовой частоты PLL для ядра.

### Режим пониженного энергопотребления:

- батарейный домен с часами реального времени, календарем, тремя детекторами фиксации проникновения и ОЗУ 512 байт.

### Аналоговые модули:

- 24-разрядный  $\Delta\Sigma$ АЦП (семь независимых каналов с ПКУ)\*;
- 10-разрядный АЦП (три внешних мультиплексируемых канала и канал внутреннего термодатчика)\*.

### Периферия:

- контроллер прямого доступа в память с функциями передачи Периферия-Память, Память-Память;
- контроллеры интерфейсов 4 – UART, 3 – SSP, 1 – I2C, 1 – ISO7816;
- сопроцессоры блочных шифров «Кузнечик», «Магма» и AES;

---

\* Работают только при  $U_{CC}$  3,0-3,6 В.

- генератор случайных чисел и сопроцессоры, для обеспечения защищенного обмена данными по протоколу СПОДЭС;
- сетка, датчик частоты и напряжения питания;
- до 55 пользовательских линий ввода/вывода;
- четыре блока 32-разрядных таймеров с четырьмя каналами захвата событий и ШИМ;
- два сторожевых таймера;
- блок подсчета CRC с изменяемым полиномом.

**Режим отладки:**

- последовательный отладочный интерфейс JTAG.

## Содержание

1	Введение .....	17
2	Структурная блок-схема .....	18
3	Условное графическое изображение .....	19
4	Диаграмма расположения выводов в корпусе .....	20
5	Описание выводов .....	21
6	Указания по применению и эксплуатации .....	30
7	Система питания .....	31
	7.1 Структурная схема подачи питания .....	32
	7.2 Схема сброса при включении и выключении основного питания .....	34
8	Организация памяти .....	36
	8.1 Базовые адреса процессора .....	37
9	Загрузочное OTP и режимы работы микросхемы .....	39
	9.1 Разработка кода загрузочной программы и его загрузка в OTP-память .....	40
	9.2 Пример одного из возможных вариантов загрузочной программы .....	43
	9.2.1 UART-загрузчик .....	44
	9.2.2 Команды UART загрузчика .....	45
10	Контроллер FLASH-памяти программ .....	50
	10.1 Работа FLASH-памяти программ в обычном режиме .....	50
	10.2 Работа FLASH-памяти программ в режиме программирования .....	50
	10.2.1 Стирание всей памяти или 256 Кбайт основной памяти. ....	51
	10.2.2 Стирание страницы памяти размером 4 Кбайт. ....	52
	10.2.3 Запись 32-битного слова в память .....	53
	10.2.4 Чтение 32-битного слова из памяти .....	55
	10.3 Регистры управления контроллера FLASH-памяти программ .....	56
	10.3.1 FLASH_CMD .....	56
	10.3.2 FLASH_ADR .....	57
	10.3.3 FLASH_DI .....	58
	10.3.4 FLASH_DO .....	58
	10.3.5 FLASH_KEY .....	59
11	Система команд .....	60
12	Процессорное ядро BM-310S .....	63
	12.1 Структурная схема процессора .....	64
	12.1.1 Интерфейс TCM .....	65
	12.1.2 Интерфейс АНВ I/O .....	65
	12.1.3 Запросы прерываний .....	65
	12.1.4 Интерфейс JTAG .....	65
	12.1.5 LIO crossbar .....	65
13	Блок АЦП для измерения напряжений и токов в электрической сети .....	66
	13.1 Описание регистров управления блока семи каналов АЦП .....	69
	13.1.1 ADCUI_CTRL1 .....	72
	13.1.2 ADCUI_CTRL2 .....	75
	13.1.3 ADCUI_CTRL3 .....	76



13.1.4	ADCUI_F0CTR.....	76
13.1.5	ADCUI_F0WC .....	77
13.1.6	ADCUI_F0WATTP.....	78
13.1.7	ADCUI_F0WATTN.....	78
13.1.8	ADCUI_F0VC.....	78
13.1.9	ADCUI_F0VARP.....	79
13.1.10	ADCUI_F0VARN.....	79
13.1.11	ADCUI_F0AC.....	79
13.1.12	ADCUI_F0VR.....	80
13.1.13	ADCUI_F0MD0.....	80
13.1.14	ADCUI_F0MD1.....	81
13.1.15	ADCUI_F0VPEAK.....	82
13.1.16	ADCUI_F0IPEAK.....	82
13.1.17	ADCUI_F0VDAT.....	82
13.1.18	ADCUI_F0I0DAT.....	83
13.1.19	ADCUI_F0I3DAT.....	83
13.1.20	ADCUI_F0VRMS.....	83
13.1.21	ADCUI_F0VRMS2.....	84
13.1.22	ADCUI_F0IRMS.....	84
13.1.23	ADCUI_F0IRMS2.....	84
13.1.24	ADCUI_F0STAT.....	85
13.1.25	ADCUI_F0MASK.....	87
13.1.26	ADCUI_F1CTR.....	88
13.1.27	ADCUI_F1WC.....	89
13.1.28	ADCUI_F1WATTP.....	89
13.1.29	ADCUI_F1WATTN.....	90
13.1.30	ADCUI_F1VC.....	90
13.1.31	ADCUI_F1VARP.....	90
13.1.32	ADCUI_F1VARN.....	91
13.1.33	ADCUI_F1AC.....	91
13.1.34	ADCUI_F1VR.....	92
13.1.35	ADCUI_F1MD0.....	92
13.1.36	ADCUI_F1MD1.....	93
13.1.37	ADCUI_F1MD2.....	93
13.1.38	ADCUI_F1VPEAK.....	93
13.1.39	ADCUI_F1IPEAK.....	94
13.1.40	ADCUI_F1VDAT.....	94
13.1.41	ADCUI_F1IDAT.....	94
13.1.42	ADCUI_F1VRMS.....	95
13.1.43	ADCUI_F1VRMS2.....	95
13.1.44	ADCUI_F1IRMS.....	95
13.1.45	ADCUI_F1IRMS2.....	96
13.1.46	ADCUI_F1STAT.....	96

13.1.47	ADCUI_F1MASK.....	98
13.1.48	ADCUI_F2CTR.....	99
13.1.49	ADCUI_F2WC.....	100
13.1.50	ADCUI_F2WATTP.....	100
13.1.51	ADCUI_F2WATTN.....	101
13.1.52	ADCUI_F2VC.....	101
13.1.53	ADCUI_F2VARP.....	101
13.1.54	ADCUI_F2VARN.....	102
13.1.55	ADCUI_F2AC.....	102
13.1.56	ADCUI_F2VR.....	103
13.1.57	ADCUI_F2MD0.....	103
13.1.58	ADCUI_F2MD1.....	104
13.1.59	ADCUI_F2MD2.....	104
13.1.60	ADCUI_F2VPEAK.....	105
13.1.61	ADCUI_F2IPEAK.....	105
13.1.62	ADCUI_F2VDAT.....	105
13.1.63	ADCUI_F2IDAT.....	106
13.1.64	ADCUI_F2VRMS.....	106
13.1.65	ADCUI_F2VRMS2.....	106
13.1.66	ADCUI_F2IRMS.....	107
13.1.67	ADCUI_F2IRMS2.....	107
13.1.68	ADCUI_F2STAT.....	107
13.1.69	ADCUI_F2MASK.....	109
13.1.70	ADCUI_CCAL1 (в ревизии 1).....	110
13.1.71	ADCUI_CCAL1 (доступно с ревизии 2).....	110
13.1.72	ADCUI_CCAL2 (в ревизии 1).....	111
13.1.73	ADCUI_CCAL2 (доступно с ревизии 2).....	111
13.1.74	ADCUI_CCAL3 (в ревизии 1).....	111
13.1.75	ADCUI_CCAL3 (доступно с ревизии 2).....	112
13.1.76	ADCUI_CCAL4 (в ревизии 1).....	112
13.1.77	ADCUI_CCAL4 (доступно с ревизии 2).....	112
13.1.78	ADCUI_F0IRMS_INACTIVE.....	113
13.1.79	ADCUI_F0WATTP_L.....	113
13.1.80	ADCUI_F0WATTN_L.....	113
13.1.81	ADCUI_F0VARP_L.....	114
13.1.82	ADCUI_F0VARN_L.....	114
13.1.83	ADCUI_F0VR_L.....	114
13.1.84	ADCUI_F1WATTP_L.....	115
13.1.85	ADCUI_F1WATTN_L.....	115
13.1.86	ADCUI_F1VARP_L.....	115
13.1.87	ADCUI_F1VARN_L.....	116
13.1.88	ADCUI_F1VR_L.....	116
13.1.89	ADCUI_F2WATTP_L.....	116

13.1.90	ADCUI_F2WATTN_L.....	117
13.1.91	ADCUI_F2VARP_L .....	117
13.1.92	ADCUI_F2VARN_L.....	117
13.1.93	ADCUI_F2VR_L .....	118
13.1.94	ADCUI_LOAD (доступно с ревизии 2) .....	118
13.1.95	ADCUI_F0VRMS_TRUE (доступно с ревизии 2).....	119
13.1.96	ADCUI_F1VRMS_TRUE (доступно с ревизии 2).....	119
13.1.97	ADCUI_F2VRMS_TRUE (доступно с ревизии 2).....	120
14	Алгоритмы вычисления окончечных результатов и их соответствия внешним сигналам.....	121
14.1	Типовая схема включения для учета электроэнергии по трем фазам .....	128
14.2	Типовая схема включения для учета электроэнергии по одной фазе .....	130
14.3	Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта .....	131
15	Аппаратный блок вычисления CRC .....	132
15.1	Описание регистров управления блока CRC .....	132
15.1.1	CRC_CTRL .....	132
15.1.2	CRC_STAT .....	133
15.1.3	CRC_DATAI.....	134
15.1.4	CRC_VAL .....	134
15.1.5	CRC_POL.....	134
16	Сигналы тактовой частоты .....	135
16.1	Встроенный RC-Генератор HSI.....	136
16.2	Встроенный RC-генератор LSI.....	136
16.3	Внешний осциллятор HSE .....	136
16.4	Внешний осциллятор LSE.....	136
16.5	Встроенный блок умножения системной тактовой частоты .....	137
16.6	Описание регистров блока контроллера тактовой частоты.....	137
16.6.1	CLOCK_STATUS.....	138
16.6.2	PLL_CONTROL .....	138
16.6.3	HS_CONTROL .....	139
16.6.4	CPU_CLOCK.....	139
16.6.5	PER1_CLOCK .....	140
16.6.6	ADC_CLOCK .....	141
16.6.7	RTC_CLOCK.....	143
16.6.8	PER2_CLOCK .....	144
16.6.9	DMA_DONE_STICK (доступно с ревизии 2) .....	145
16.6.10	TIM_CLOCK .....	145
16.6.11	UART_CLOCK.....	147
16.6.12	SSP_CLOCK .....	148
16.6.13	DIV_SYS_TIM (доступно с ревизии 2) .....	149
17	Батарейный домен и часы реального времени .....	150
17.1	Часы реального времени .....	150

17.2	Память регистров и криптографический ключей.....	153
17.3	Описание регистров блока батарейного домена.....	153
17.3.1	BKP_MEM (128x32) .....	154
17.3.2	BKP_WPR.....	154
17.3.3	BKP_LDO .....	154
17.3.4	BKP_CLK.....	156
17.3.5	BKP_RTC_CR .....	157
17.3.6	BKP_RTC_WUTR.....	159
17.3.7	BKP_RTC_PREDIV_S .....	159
17.3.8	BKP_RTC_PRL .....	159
17.3.9	BKP_RTC_ALRM .....	160
17.3.10	BKP_RTC_CS.....	160
17.3.11	BKP_RTC_TR.....	162
17.3.12	BKP_RTC_DR .....	163
17.3.13	BKP_RTC_ALRMAR, BKP_RTC_ALRMBR .....	163
17.3.14	BKP_RTC_TSTR1-BKP_RTC_TSTR3, BKP_RTC_TSDR1-BKP_RTC_TSDR3 .....	164
17.3.15	BKP_RTC_TAFCR.....	165
18	Порты ввода-вывода.....	168
18.1	Описание регистров портов ввода-вывода.....	170
18.1.1	PORTx_RXTX .....	171
18.1.2	PORTx_OE.....	171
18.1.3	PORTx_FUNC .....	172
18.1.4	PORTx_ANALOG .....	172
18.1.5	PORTx_PULL .....	173
18.1.6	PORTx_PWR .....	173
18.1.7	PORTx_SETTX.....	173
18.1.8	PORTx_CLRTX.....	174
19	Детектор напряжения питания .....	175
19.1	Описание регистров блока PVD .....	175
19.1.1	PVDCS .....	176
20	Таймеры общего назначения .....	178
20.1	Основные характеристики .....	178
20.1.1	Структурная схема.....	179
20.2	Базовый блок таймера.....	179
20.2.1	Инициализация тактирования таймера .....	180
20.2.2	Инициализация основного счетчика таймера.....	180
20.2.3	Режимы счета .....	181
20.2.4	Тактовая частота F <sub>DTS</sub> .....	183
20.3	Источник событий для счета .....	183
20.3.1	Внутренний тактовый сигнал (TIM_CLKd).....	184
20.3.2	Событие в другом таймере (CNT==ARR).....	185

20.3.3	Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала СНу1.....	186
20.3.4	Внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR.....	187
20.4	Режим захвата.....	188
20.5	Режим ШИМ.....	190
20.5.1	Генератор опорного сигнала REF .....	191
20.5.2	Генератор «мертвой зоны» .....	192
20.5.3	Выходные блоки .....	193
20.6	Блок цифрового фильтра.....	194
20.7	Флаги состояний, прерывания и запросы DMA .....	195
20.7.1	Флаги состояний .....	196
20.7.2	Прерывания .....	196
20.7.3	Запросы DMA.....	196
20.8	Примеры.....	196
20.8.1	Обычный счетчик .....	196
20.8.2	Режим захвата .....	197
20.8.3	Режим ШИМ .....	198
20.9	Описание регистров блока таймера .....	199
20.9.1	CNT .....	200
20.9.2	PSG .....	200
20.9.3	ARR .....	201
20.9.4	CNTRL .....	201
20.9.5	CHy_CCR.....	202
20.9.6	CHy_CCR1.....	203
20.9.7	CHy_CNTRL0.....	203
20.9.8	CHy_CNTRL1.....	205
20.9.9	CHy_CNTRL2.....	206
20.9.10	CHy_DTG.....	207
20.9.11	BRKETR_CNTRL .....	208
20.9.12	STATUS .....	209
20.9.13	IE.....	211
20.9.14	DMA_RE.....	212
21	Контроллер АЦП .....	214
21.1	Преобразование внешнего канала .....	215
21.2	Последовательное преобразование нескольких каналов .....	215
21.3	Преобразование с контролем границ .....	216
21.4	Датчик температуры .....	216
21.4.1	Формула расчета температуры.....	217
21.5	Время заряда внутренней емкости .....	217
21.6	Время заряда внутренней емкости АЦП и время преобразования .....	218
21.7	Описание регистров блока контроллера АЦП.....	219
21.7.1	ADC1_CFG .....	219

21.7.2	ADC1_H_LEVEL .....	221
21.7.3	ADC1_L_LEVEL .....	222
21.7.4	ADC1_RESULT .....	222
21.7.5	ADC1_STATUS .....	223
21.7.6	ADC1_CHSEL .....	223
21.7.7	ADC_TRIM .....	224
22	Контроллер интерфейса I2C .....	225
22.1	Конфигурация системы .....	225
22.2	Протокол I2C .....	225
22.2.1	Сигнал START .....	226
22.2.2	Передача адреса .....	226
22.2.3	Передача данных .....	226
22.2.4	Сигнал STOP .....	227
22.3	Описание регистров контроллера I2C .....	227
22.3.1	PRL .....	227
22.3.2	PRH .....	228
22.3.3	CTR .....	228
22.3.4	RXD .....	228
22.3.5	STA .....	229
22.3.6	TXD .....	229
22.3.7	CMD .....	230
23	Контроллер SSP .....	231
23.1	Основные характеристики модуля SSP .....	231
23.1.1	Программируемые параметры .....	232
23.1.2	Характеристики интерфейса SPI .....	233
23.1.3	Характеристики интерфейса Microwire .....	233
23.1.4	Характеристики интерфейса SSI .....	233
23.2	Общий обзор модуля SSP .....	233
23.2.1	Блок формирования тактового сигнала .....	234
23.2.2	Буфер FIFO передатчика .....	235
23.2.3	Буфер FIFO приемника .....	235
23.2.4	Блок приема и передачи данных .....	235
23.2.5	Блок формирования прерываний .....	236
23.3	Интерфейс прямого доступа к памяти .....	236
23.4	Конфигурирование приемопередатчика .....	236
23.5	Разрешение работы приемопередатчика .....	236
23.6	Соотношения между тактовыми сигналами .....	236
23.7	Программирование регистра управления CR0 .....	238
23.8	Программирование регистра управления CR1 .....	238
23.9	Формирование тактового сигнала обмена данными .....	239
23.10	Формат информационного кадра .....	239
23.10.1	Формат синхронного обмена SSI фирмы Texas Instruments .....	240
23.10.2	Формат синхронного обмена SPI фирмы Motorola .....	241

23.10.3	Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0 ...	241
23.10.4	Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1 ...	242
23.10.5	Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0 ...	243
23.10.6	Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1 ...	245
23.10.7	Формат синхронного обмена Microwire фирмы National Semiconductor	246
23.11	Примеры конфигурации модуля в ведущем и ведомом режимах .....	248
23.12	Интерфейс прямого доступа к памяти.....	250
23.13	Прерывания .....	252
23.13.1	SSPRXINTR.....	253
23.13.2	SSPTXINTR.....	253
23.13.3	SSPRORINTR .....	253
23.13.4	SSPRTINTR .....	253
23.13.5	SSPRNEINTR (доступно с ревизии 2) .....	253
23.13.6	SSPTFEINTR (доступно с ревизии 2) .....	253
23.13.7	SSPTNBSYINTR (доступно с ревизии 2) .....	253
23.13.8	SSPINTR .....	254
23.14	Программное управление модулем .....	254
23.14.1	Общая информация .....	254
23.14.2	Описание регистров контроллера SSP .....	254
24	Контроллер UART .....	263
24.1	Основные характеристики модуля UART .....	263
24.2	Программируемые параметры .....	264
24.3	Отличия от контроллера UART 16C650 .....	264
24.4	Функциональные возможности .....	265
24.5	Описание функционирования блока UART .....	266
24.5.1	Генератор тактового сигнала приемопередатчика .....	267
24.5.2	Буфер FIFO передатчика.....	267
24.5.3	Буфер FIFO приемника .....	267
24.5.4	Блок передатчика .....	267
24.5.5	Блок приемника .....	267
24.5.6	Блок формирования прерываний .....	268
24.5.7	Интерфейс прямого доступа к памяти.....	268
24.5.8	Блок и регистры синхронизации .....	268
24.6	Описание функционирования ИК-кодека IrDA SIR.....	268
24.6.1	Кодер ИК-передатчика.....	269
24.6.2	Декодер ИК-приемника.....	269
24.7	Описание работы UART .....	270
24.7.1	Сброс модуля .....	270
24.7.2	Тактовые сигналы .....	270
24.7.3	Работа универсального асинхронного приемопередатчика .....	270
24.7.4	Коэффициент деления частоты .....	271
24.7.5	Передача и прием данных.....	272

24.7.6	Биты ошибки .....	273
24.7.7	Бит переполнения буфера .....	273
24.7.8	Запрет буфера FIFO .....	273
24.7.9	Работа кодека ИК обмена данными IrDA SIR .....	273
24.8	Линии управления модемом .....	275
24.8.1	Аппаратное управление потоком данных .....	276
24.8.2	Управление потоком данных по линии RTS.....	276
24.8.3	Управление потоком данных по линии CTS.....	277
24.9	Интерфейс прямого доступа к памяти .....	277
24.10	Прерывания .....	279
24.10.1	UARTMSINTR .....	280
24.10.2	UARTRXINTR.....	281
24.10.3	UARTTXINTR.....	281
24.10.4	UARTRTINTR.....	281
24.10.5	UARTEINTR.....	282
24.10.6	UARTINTR .....	282
24.11	Программное управление модулем .....	282
24.12	Обобщенные данные о регистрах устройства .....	283
24.12.1	Регистр данных DR.....	283
24.12.2	Регистр состояния приемника / сброса ошибки RSR_ECR.....	285
24.12.3	Регистр флагов FR .....	286
24.12.4	Регистр управления ИК-обменом в режиме пониженного энергопотребления ILPR.....	287
24.12.5	Регистр целой части делителя скорости передачи данных IBRD.....	288
24.12.6	Регистр дробной части делителя скорости передачи данных FBRD .	288
24.12.7	Регистр управления линией LCR_H .....	290
24.12.8	Регистр управления CR.....	291
24.12.9	Регистр порога прерывания по заполнению буфера FIFO IFLS.....	294
24.12.10	Регистр установки сброса маски прерывания IMSC.....	295
24.12.11	Регистр состояния прерываний RIS.....	296
24.12.12	Регистр маскированного состояния прерываний MIS .....	298
24.12.13	Регистр сброса прерываний ICR .....	299
24.12.14	Регистр управления прямым доступом к памяти DMACR .....	300
24.12.15	Регистр управления тестированием TCR.....	301
25	Контроллер прямого доступа в память DMA .....	302
25.1	Основные свойства контроллера DMA.....	302
25.2	Термины и определения .....	302
25.3	Функциональное описание.....	304
25.3.1	Распределение каналов DMA .....	304
25.3.2	Блок, подключенный к шине APB .....	305
25.3.3	Блок, подключенный к шине AHB .....	305
25.3.4	Управляющий блок DMA .....	305
25.3.5	Типы передач .....	306



25.3.6	Разрядность передаваемых данных .....	306
25.3.7	Управление защитой данных.....	306
25.3.8	Инкремент адреса .....	307
25.4	Управление DMA .....	308
25.4.1	Правила обмена данными .....	308
25.4.2	Диаграммы работы контроллера DMA .....	310
25.4.3	Правила арбитража DMA .....	316
25.4.4	Приоритет .....	317
25.4.5	Типы циклов DMA .....	318
25.5	Структура управляющих данных канала.....	330
25.5.1	Указатель конца данных источника.....	334
25.5.2	Указатель конца данных приемника.....	334
25.5.3	Разряды управления .....	335
25.5.4	Вычисление адреса .....	339
25.6	Описание регистров контроллера DMA .....	341
25.6.1	Статусный регистр DMA STATUS .....	343
25.6.2	Регистр конфигурации DMA CFG .....	344
25.6.3	Регистр базового адреса управляющих данных каналов CTRL_BASE_PTR.....	344
25.6.4	Регистр базового адреса альтернативных управляющих данных каналов ALT_CTRL_BASE_PTR .....	345
25.6.5	Регистр статуса ожидания запроса на обработку каналов WAITONREQ_STATUS .....	346
25.6.6	Регистр программного запроса на обработку каналов CHNL_SW_REQUEST .....	346
25.6.7	Регистр установки пакетного обмена каналов CHNL_USEBURST_SET 347	
25.6.8	Регистр сброса пакетного обмена каналов CHNL_USEBURST_CLR	348
25.6.9	Регистр маскирования запросов на обслуживание каналов CHNL_REQ_MASK_SET.....	348
25.6.10	Регистр очистки маскирования запросов на обслуживание каналов CHNL_REQ_MASK_CLR .....	349
25.6.11	Регистр установки разрешения каналов CHNL_ENABLE_SET .....	350
25.6.12	Регистр сброса разрешения каналов CHNL_ENABLE_CLR .....	351
25.6.13	Регистр установки первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_SET .....	351
25.6.14	Регистр сброса первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_CLR .....	352
25.6.15	Регистр установки приоритета каналов CHNL_PRIORITY_SET .....	353
25.6.16	Регистр сброса приоритета каналов CHNL_PRIORITY_CLR .....	354
25.6.17	Регистр сброса флага ошибки ERR_CLR.....	355
26	Прерывания .....	356
27	Контроллер обработки внешних прерываний PLIC.....	359

27.1	Обработка внешних прерываний контроллером PLIC.....	359
27.2	Программная модель обработки внешних прерываний.....	361
27.3	Регистры контроллера PLIC.....	362
27.3.1	Регистры PRI1-PRI31.....	363
27.3.2	Регистр IPM.....	364
27.3.3	Регистры IEM.....	364
27.3.4	Регистр THR.....	365
27.3.5	Регистр ICC.....	365
28	Подсистема отладки.....	367
28.1	Блок DTM (Debug Transport Module).....	367
28.1.1	Регистры DTM.....	368
28.2	Блок DM (Debug Module).....	369
28.2.1	Регистры DM.....	370
28.3	Подключение JTAG-адаптера к микросхеме.....	372
29	Сторожевые таймеры.....	373
29.1	Блок сторожевого таймера IWDG.....	373
29.1.1	Регистры блока сторожевых таймеров.....	374
29.2	Блок сторожевого таймера WWDG.....	378
29.2.1	Описание регистров блока сторожевого таймера WWDG.....	380
30	Блок прямого и обратного L-преобразования.....	383
30.1	Особенности.....	383
30.2	Регистры блока.....	384
30.2.1	Регистр преобразования n-ого слова данных TRANSFORM_n.....	384
30.2.2	Регистр изменения таблицы TABLE_CHANGE_n.....	385
30.2.3	Регистр настройки преобразования SETUP.....	386
30.3	Инициализация.....	386
30.4	Выполнение преобразования.....	387
31	Блок замены S-box.....	388
31.1	Регистры блока.....	388
31.1.1	Регистр преобразования n-ого слова данных TRANSFORM_n.....	389
31.1.2	Регистр изменения таблицы TABLE_CHANGE.....	389
31.2	Инициализация восьмибитного S-BOX.....	390
31.3	Инициализация четырехбитного S-BOX.....	390
31.3.1	Выполнение преобразования.....	391
32	Блок байтовой замены p-byte.....	392
32.1	Регистры модуля.....	392
32.1.1	Регистр задания n-ого слова данных для перестановки TRANSFORM_n 393	
32.1.2	Регистр задания перестановки MUX_SET.....	393
32.1.3	Регистр чтения перестановки MUX_VAL.....	394
32.2	Инициализация.....	394
32.3	Выполнение преобразования.....	394
33	Блок битовой замены p-bit.....	395

33.1	Регистры блока .....	395
33.1.1	Регистр задания n-ого слова данных для перестановки TRANSFORM_n 396	
33.1.2	Регистр задания перестановки MUX_SET .....	396
33.1.3	Регистр чтения перестановки MUX_VAL.....	396
33.2	Инициализация.....	397
33.3	Выполнение преобразования.....	397
34	Модуль обработки датчиков безопасности (SENSORS) .....	398
34.1	Общее описание .....	398
34.2	Описание функционирования блока и режимов работы .....	399
34.3	Карта регистров.....	399
34.3.1	Регистр state_reg.....	400
34.3.2	Регистр real_time.....	400
34.3.3	Регистр int_mask .....	401
34.3.4	Регистр k_res_mask.....	401
34.3.5	Регистр enable.....	401
34.3.6	Регистр meshcntr .....	402
35	Датчик частоты (CLK_MEASURE) .....	403
35.1	Карта регистров.....	405
35.1.1	CLK_CNTR_STAT.....	406
35.1.2	ALARM_SHIFT_RST .....	406
35.1.3	ALARM_SHIFT_INT .....	406
35.1.4	ALARM_PREG_0.....	406
35.1.5	ALARM_PREG_1.....	407
35.1.6	CLK_STAT .....	407
36	Генератор случайных чисел (RANDOM) .....	408
36.1	Назначение и свойства .....	408
36.2	Регистры модуля .....	408
36.2.1	Регистр статуса и управления STAT_CTRL_REG .....	409
36.2.2	Регистр управления прерыванием INT_CTRL_REG .....	410
36.2.3	Регистр делителя клокa генератора CLK_DIV_REG .....	410
36.2.4	Регистр паузы включения PAUSE_REG .....	410
36.2.5	Регистр случайного значения OUTPUT_REG .....	411
36.2.6	Регистр счетчика паузы PAUSE_CNT_REG.....	411
36.2.7	Регистр сбора случайного числа TEMP_REG .....	411
37	Контроллер UART (ISO7816) .....	412
37.1	Введение .....	412
37.2	Особенности модуля USART.....	412
37.3	Функциональное описание USART .....	413
37.3.1	Описание символов USART .....	415
37.3.2	Передатчик .....	415
37.3.3	Приемник.....	418
37.3.4	Дробный генератор скорости .....	423

37.3.5	Допуск ухода тактовой частоты для приемника USART .....	426
37.3.6	Многопроцессорный обмен .....	427
37.3.7	Контроль четности.....	429
37.3.8	Синхронный режим USART .....	430
37.3.9	Однопроводной полудуплексный обмен данными .....	432
37.3.10	Режим Smartcard .....	433
37.3.11	Аппаратное управление потоком.....	435
37.4	Прерывания USART .....	436
37.5	Регистры USART.....	437
37.5.1	USART_SR .....	437
37.5.2	USART_DR.....	440
37.5.3	USART_BRR .....	441
37.5.4	USART_CR1 .....	442
37.5.5	USART_CR2.....	445
37.5.6	USART_CR3.....	446
37.5.7	USART_GTPR.....	448
38	Модуль однократно программируемой памяти (OTP) .....	449
38.1	Назначение и свойства .....	449
38.1.1	Регистры модуля .....	450
39	Электрические параметры микросхем .....	457
40	Предельно-допустимые и предельные параметры .....	458
41	Справочные параметры.....	460
42	Габаритный чертеж микросхемы .....	462
43	Информация для заказа .....	463

## 1 Введение

Микросхема относится к серии микроконтроллеров со встроенной FLASH-памятью программ и построена на базе процессорного RISC-V ядра BM-310S. Микросхема работает на тактовой частоте до 60 МГц и содержит 256+8 Кбайт FLASH-памяти и 112 Кбайт ОЗУ. Микросхема включает в себя развитую периферию для построения счетчиков электроэнергии одно- и трехфазных сетей. Периферия включает в себя семь каналов для трехфазной сети (или три канала для однофазной сети) 24-битных независимых  $\Delta\Sigma$ АЦП. Каждый канал АЦП имеет предусилитель, фазовую подстройку (для коррекции фазы не хуже 0,1), а также аппаратный блок для вычисления среднеквадратического значения сигнала. Каждый канал АЦП может быть включен или отключен независимо от других каналов и имеет отдельный канал прямого доступа в память. Еще один дополнительный 10-битный АЦП последовательного приближения может быть использован для мониторинга напряжения питания основного или батарейного доменов, а также для измерения температуры или захвата внешнего сигнала. В состав микросхемы входит пять интерфейсов UART, три SSP и один интерфейс I2C. Криптографическая часть микросхемы включает сопроцессоры блочных шифров «Кузнечик» (ГОСТ Р34.13-2015), «Магма» и AES, генератор случайных чисел и сопроцессоры для обеспечения защищенного обмена данными по протоколу СПОДЭС, а также инженерные методы защиты криптографической части: сетка, датчик частоты и напряжения питания. Микросхема содержит четыре 32-разрядных таймера с четырьмя каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Также микросхема содержит системный 64-разрядный таймер и два сторожевых таймера.

Встроенные RC-генераторы HSI (8 МГц) и LSI (32 кГц), внешние генераторы HSE (8...16 МГц) и LSE (32 кГц) и схема умножения тактовой частоты PLL для ядра позволяют гибко настраивать скорость работы микросхем.

Архитектура системой шины за счет регулировки частоты периферийных блоков позволяет уменьшить потребление всей системы. Контроллер DMA позволяет ускорить обмен информацией между ОЗУ и периферией без участия процессорного ядра.

Встроенный регулятор для формирования питания внутренней цифровой части формирует напряжения 1,2 В и не требует дополнительных внешних элементов. Таким образом, для работы микросхемы достаточно одного внешнего напряжения питания в диапазоне от 3,0 до 3,6 В. Также в микросхеме реализован батарейный домен, работающий от внешней батареи при отсутствии основного питания. В батарейном домене могут быть сохранены криптографические ключи, а также работают часы реального времени. Встроенные детекторы напряжения питания могут отслеживать уровень внешнего основного питания, уровень напряжения питания на батарее. Аппаратные схемы сброса по просадке питания позволяют исключить сбойную работу микросхемы при выходе уровня напряжения питания за допустимые пределы.

## 2 Структурная блок-схема

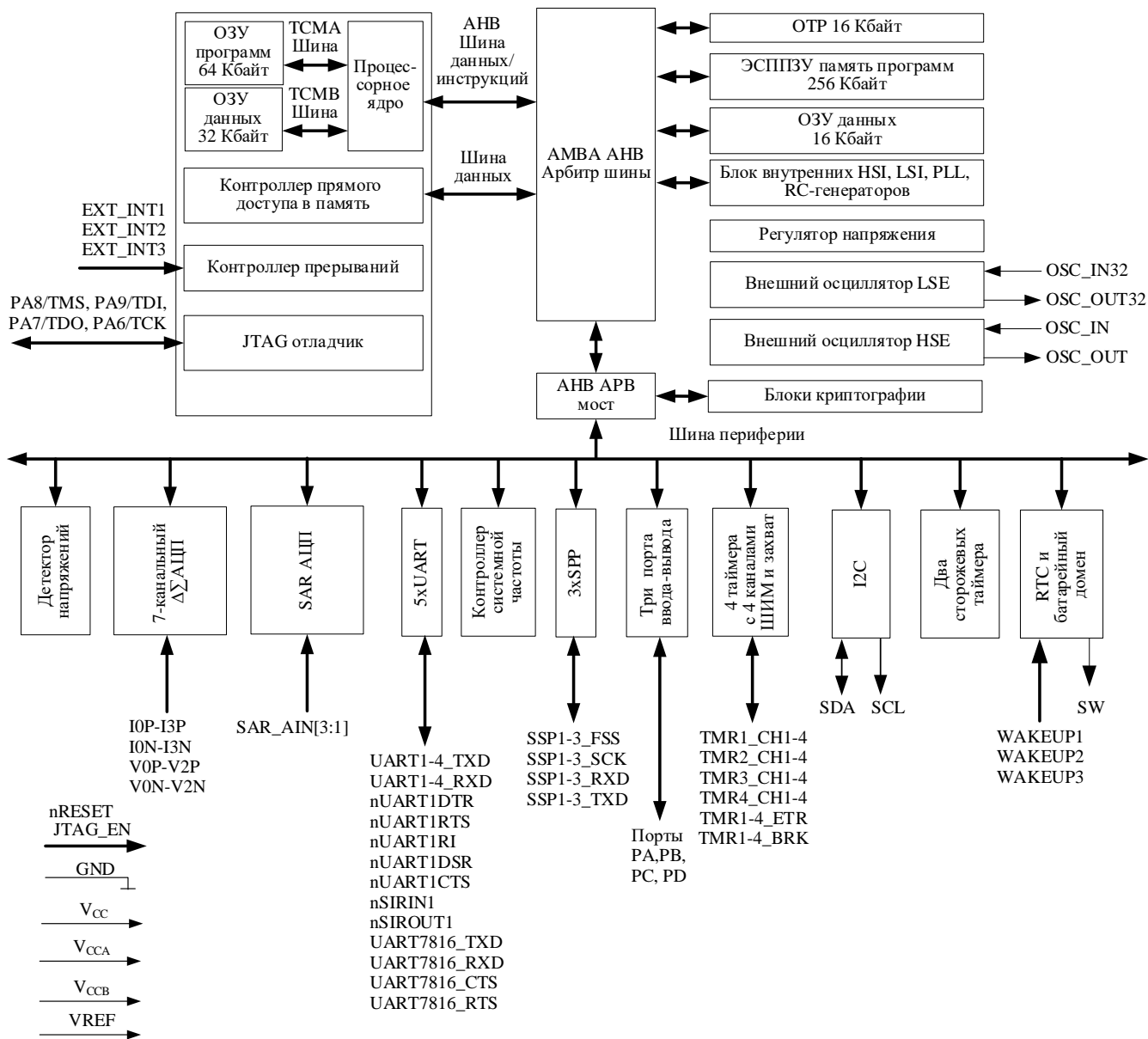
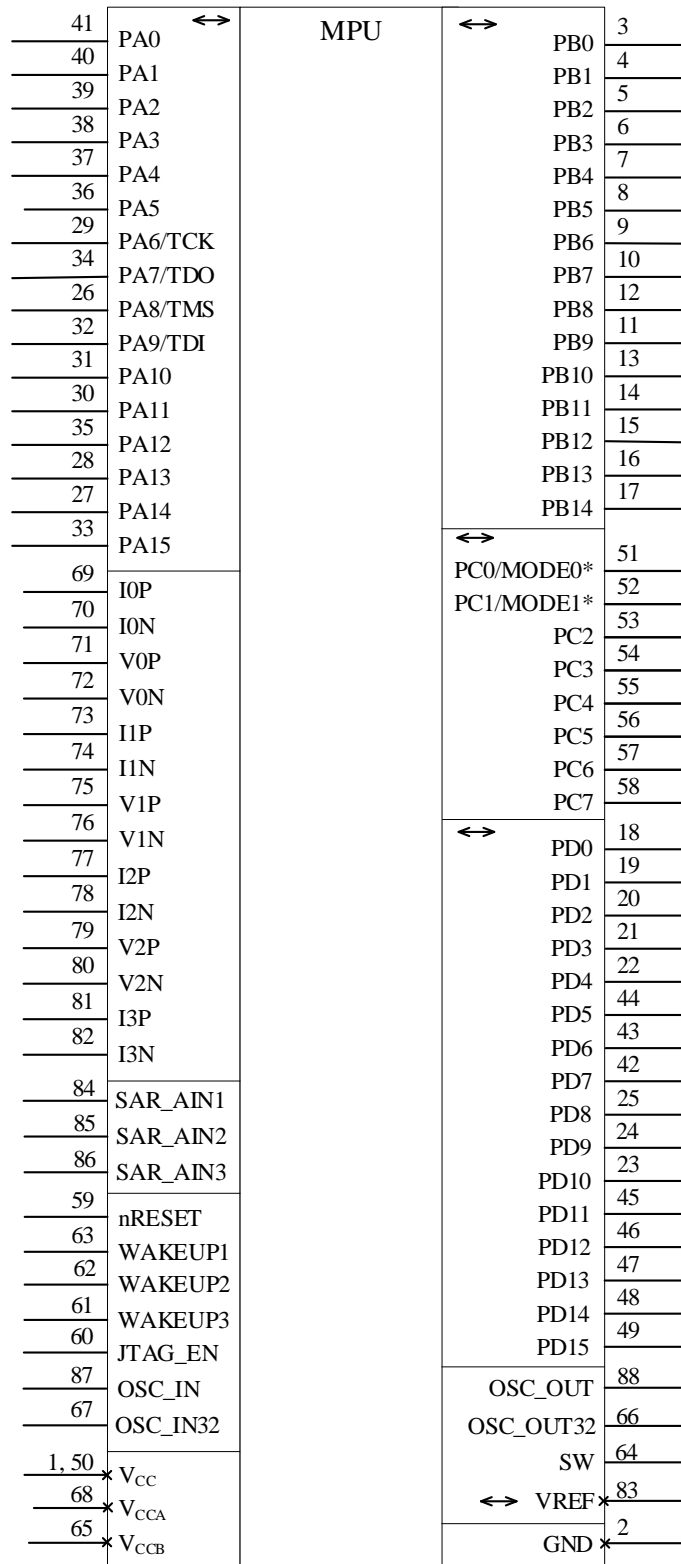


Рисунок 1 – Структурная блок-схема микросхем

### 3 Условное графическое изображение



\* См. описание выводов PC0 и PC1 в таблице 1.

Рисунок 2 – Условное графическое изображение микросхемы

## 4 Диаграмма расположения выводов в корпусе

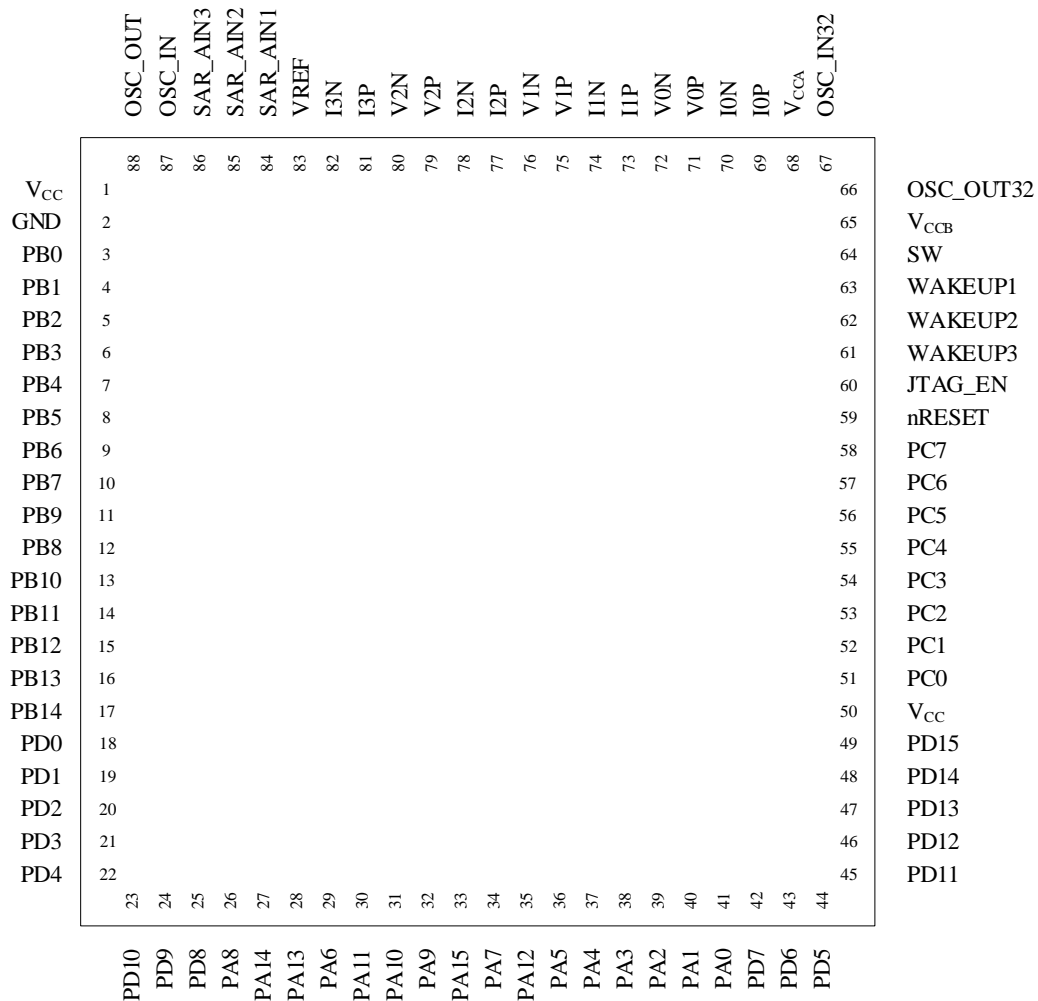


Рисунок 3 – Диаграмма раположения выводов



## 5 Описание выводов

Таблица 1 – Описание выводов

Номер вывода	Обозначение вывода	Функциональное назначение вывода
1	V <sub>CC</sub>	Питание
2	GND	Общий
3	PB0	Порт ввода-вывода В разряд 0
4	PB1	Порт ввода-вывода В разряд 1
5	PB2	Порт ввода-вывода В разряд 2
6	PB3	Порт ввода-вывода В разряд 3
7	PB4	Порт ввода-вывода В разряд 4
8	PB5	Порт ввода-вывода В разряд 5
9	PB6	Порт ввода-вывода В разряд 6
10	PB7	Порт ввода-вывода В разряд 7
11	PB9	Порт ввода-вывода В разряд 9
12	PB8	Порт ввода-вывода В разряд 8
13	PB10	Порт ввода-вывода В разряд 10
14	PB11	Порт ввода-вывода В разряд 11
15	PB12	Порт ввода-вывода В разряд 12
16	PB13	Порт ввода-вывода В разряд 13
17	PB14	Порт ввода-вывода В разряд 14
18	PD0	Порт ввода-вывода D разряд 0
19	PD1	Порт ввода-вывода D разряд 1
20	PD2	Порт ввода-вывода D разряд 2
21	PD3	Порт ввода-вывода D разряд 3
22	PD4	Порт ввода-вывода D разряд 4
23	PD10	Порт ввода-вывода D разряд 10
24	PD9	Порт ввода-вывода D разряд 9
25	PD8	Порт ввода-вывода D разряд 8
26	PA8/ TMS	Порт ввода-вывода А разряд 8 / Вход управления JTAG
27	PA14	Порт ввода-вывода А разряд 14
28	PA13	Порт ввода-вывода А разряд 13
29	PA6/ TCK	Порт ввода-вывода А разряд 6 / Вход синхросигнала JTAG
30	PA11	Порт ввода-вывода А разряд 11
31	PA10	Порт ввода-вывода А разряд 10
32	PA9/ TDI	Порт ввода-вывода А разряд 9 / Вход данных JTAG
33	PA15	Порт ввода-вывода А разряд 15
34	PA7/ TDO	Порт ввода-вывода А разряд 7 / Выход данных JTAG
35	PA12	Порт ввода-вывода А разряд 12
36	PA5	Порт ввода-вывода А разряд 5

Номер вывода	Обозначение вывода	Функциональное назначение вывода
37	PA4	Порт ввода-вывода А разряд 4
38	PA3	Порт ввода-вывода А разряд 3
39	PA2	Порт ввода-вывода А разряд 2
40	PA1	Порт ввода-вывода А разряд 1
41	PA0	Порт ввода-вывода А разряд 0
42	PD7	Порт ввода-вывода D разряд 7
43	PD6	Порт ввода-вывода D разряд 6
44	PD5	Порт ввода-вывода D разряд 5
45	PD11	Порт ввода-вывода D разряд 11
46	PD12	Порт ввода-вывода D разряд 12
47	PD13	Порт ввода-вывода D разряд 13
48	PD14	Порт ввода-вывода D разряд 14
49	PD15	Порт ввода-вывода D разряд 15
50	V <sub>CC</sub>	Питание
51	PC0/ MODE0	Порт ввода-вывода С разряд 0 / Вход выбора режима работы микросхемы разряд 0 (при использовании загрузочной программы, описанной в подразделе 9.2 «Пример одного из возможных вариантов загрузочной программы»)
52	PC1/ MODE1	Порт ввода-вывода С разряд 1 / Вход выбора режима работы микросхемы разряд 1 (при использовании загрузочной программы, описанной в подразделе 9.2 «Пример одного из возможных вариантов загрузочной программы»)
53	PC2	Порт ввода-вывода С разряд 2
54	PC3	Порт ввода-вывода С разряд 3
55	PC4	Порт ввода-вывода С разряд 4
56	PC5	Порт ввода-вывода С разряд 5
57	PC6	Порт ввода-вывода С разряд 6
58	PC7	Порт ввода-вывода С разряд 7
59	nRESET	Сигнал внешнего сброса
60	JTAG_EN	Вход разрешения отладочного TAP интерфейса на выводах JTAG: JTAG_EN = 1 – boundary scan (выводы PB[9:6]); JTAG_EN = 0 – JTAG-интерфейс RISC-V (выводы PA[9:6])
61	WAKEUP3	Вход сигнала вскрытия 3
62	WAKEUP2	Вход сигнала вскрытия 2
63	WAKEUP1	Вход сигнала вскрытия 1
64	SW	Выход управления переключением питания U <sub>CCV</sub> на U <sub>CC</sub>
65	V <sub>CCV</sub>	Батарейное питание
66	OSC_OUT32	Выход генератора LSE
67	OSC_IN32	Вход генератора LSE
68	V <sub>CCA</sub>	Аналоговое питание $\Delta\Sigma$ АЦП и SAR АЦП
69	IOP	Вход канала тока 0 прямой
70	ION	Вход канала тока 0 инверсный
71	VOP	Вход канала напряжения 0 прямой
72	VON	Вход канала напряжения 0 инверсный

Номер вывода	Обозначение вывода	Функциональное назначение вывода
73	I1P	Вход канала тока 1 прямой
74	I1N	Вход канала тока 1 инверсный
75	V1P	Вход канала напряжения 1 прямой
76	V1N	Вход канала напряжения 1 инверсный
77	I2P	Вход канала тока 2 прямой
78	I2N	Вход канала тока 2 инверсный
79	V2P	Вход канала напряжения 2 прямой
80	V2N	Вход канала напряжения 2 инверсный
81	I3P	Вход канала тока 3 прямой
82	I3N	Вход канала тока 3 инверсный
83	VREF	Вход/выход опорного напряжения 2,4 В
84	SAR_AIN1	Вход 1 SAR АЦП
85	SAR_AIN2	Вход 2 SAR АЦП
86	SAR_AIN3	Вход 3 SAR АЦП
87	OSC_IN	Вход генератора HSE
88	OSC_OUT	Выход генератора HSE
0 (металлизация обратной стороны корпуса)	GND	Общий

Примечание – Функции выводов портов А, В, С, D приведены в таблице 2

Таблица 2 – Функции выводов портов А, В, С, D

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
Порт А				
41	PA0	Порт ввода-вывода А разряд 0		
		Основная	TMR1_CH1	Вход-выход канала 1 таймера 1
		Альтернативная	SSP2_FSS	Вход-выход разрешения SPI 2
		Переопределенная	–	–
40	PA1	Порт ввода-вывода А разряд 1		
		Основная	TMR1_CH1N	Инверсный выход канала 1 таймера 1
		Альтернативная	SSP2_CLK	Вход-выход синхросигнала SPI 2
		Переопределенная	–	–
39	PA2	Порт ввода-вывода А разряд 2		
		Основная	TMR1_CH2	Вход-выход канала 2 таймера 1
		Альтернативная	SSP2_RXD	Входные данные SPI 2
		Переопределенная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
38	PA3	Порт ввода-вывода A разряд 3		
		Основная	TMR1_CH2N	Инверсный выход канала 2 таймера 1
		Альтернативная	SSP2_TXD	Выходные данные SPI 2
		Переопределенная	—	—
37	PA4	Порт ввода-вывода A разряд 4		
		Основная	TMR1_CH3	Вход-выход канала 3 таймера 1
		Альтернативная	—	—
		Переопределенная	—	—
36	PA5	Порт ввода-вывода A разряд 5		
		Основная	TMR1_CH3N	Инверсный выход канала 3 таймера 1
		Альтернативная	—	—
		Переопределенная	—	—
29	PA6/ TCK	Порт ввода-вывода A разряд 6 / Вход синхросигнала JTAG		
		Основная	TMR1_CH4	Вход-выход канала 4 таймера 1
		Альтернативная	—	—
		Переопределенная	—	—
34	PA7/ TDO	Порт ввода-вывода A разряд 7 / Выход данных JTAG		
		Основная	TMR1_CH4N	Инверсный выход канала 4 таймера 1
		Альтернативная	—	—
		Переопределенная	—	—
26	PA8/ TMS	Порт ввода-вывода A разряд 8 / Вход управления JTAG		
		Основная	TMR1_ETR	Вход внешнего события таймера 1
		Альтернативная	—	—
		Переопределенная	—	—
32	PA9/ TDI	Порт ввода-вывода A разряд 9 / Вход данных JTAG		
		Основная	TMR1_BRK	Вход внешнего события таймера 1
		Альтернативная	—	—
		Переопределенная	—	—
31	PA10	Порт ввода-вывода A разряд 10		
		Основная	EXT_INT1	Вход прерывания 1
		Альтернативная	UART3_RXD	Входные данные UART3
		Переопределенная	TMR4_CH1	Вход-выход канала 1 таймера 4
30	PA11	Порт ввода-вывода A разряд 11		
		Основная	TMR2_CH4N	Инверсный выход канала 4 таймера 2
		Альтернативная	UART3_TXD	Выходные данные UART3
		Переопределенная	TMR4_CH1N	Инверсный выход канала 1 таймера 4

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
35	PA12	Порт ввода-вывода А разряд 12		
		Основная	SSP1_FSS	Вход-выход разрешения SPI 1
		Альтернативная	–	–
		Переопределенная	TMR4_CH2	Вход-выход канала 2 таймера 4
28	PA13	Порт ввода-вывода А разряд 13		
		Основная	SSP1_CLK	Вход-выход синхросигнала SPI 1
		Альтернативная	–	–
		Переопределенная	TMR4_CH2N	Инверсный выход канала 2 таймера 4
27	PA14	Порт ввода-вывода А разряд 14		
		Основная	SSP1_RXD	Входные данные SPI 1
		Альтернативная	–	–
		Переопределенная	TMR4_CH3	Вход-выход канала 3 таймера 4
33	PA15	Порт ввода-вывода А разряд 15		
		Основная	SSP1_TXD	Выходные данные SPI 1
		Альтернативная	–	–
		Переопределенная	TMR4_CH3N	Инверсный выход канала 3 таймера 4
Порт В				
3	PB0	Порт ввода-вывода В разряд 0		
		Основная	UART1_TXD	Выходные данные UART1
		Альтернативная	UART7816_TXD	Выходные данные UART7816
		Переопределенная	TMR3_CH1	Вход-выход канала 1 таймера 3
4	PB1	Порт ввода-вывода В разряд 1		
		Основная	UART1_RXD	Входные данные UART1
		Альтернативная	UART7816_RXD	Входные данные UART7816
		Переопределенная	TMR3_CH1N	Инверсный выход канала 1 таймера 3
5	PB2	Порт ввода-вывода В разряд 2		
		Основная	nSIROUT1	Выходные данные IRDA UART1
		Альтернативная	UART7816_CLK	Выход синхросигнала UART7816
		Переопределенная	TMR3_CH2	Вход-выход канала 2 таймера 3
6	PB3	Порт ввода-вывода В разряд 3		
		Основная	nSIRIN1	Входные данные IRDA UART1
		Альтернативная	UART7816_CTS	Вход управления UART7816
		Переопределенная	TMR3_CH2N	Инверсный выход канала 2 таймера 3
7	PB4	Порт ввода-вывода В разряд 4		
		Основная	nUART1DTR	Выход управления UART1
		Альтернативная	–	–
		Переопределенная	TMR3_CH3	Вход-выход канала 3 таймера 3

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
8	PB5	Порт ввода-вывода В разряд 5		
		Основная	nUART1RTS	Выход управления UART1
		Альтернативная	UART7816_RTS	Выход управления UART7816
		Переопределенная	TMR3_CH3N	Инверсный выход канала 3 таймера 3
9	PB6	Порт ввода-вывода В разряд 6		
		Основная	nUART1RI	Вход управления UART1
		Альтернативная	EXT_INT2	Вход прерывания 2
		Переопределенная	TMR3_CH4	Вход-выход канала 4 таймера 3
10	PB7	Порт ввода-вывода В разряд 7		
		Основная	nUART1DCD	Вход управления UART1
		Альтернативная	EXT_INT3	Вход прерывания 3
		Переопределенная	TMR3_CH4N	Инверсный выход канала 4 таймера 3
12	PB8	Порт ввода-вывода В разряд 8		
		Основная	nUART1DSR	Вход управления UART1
		Альтернативная	TMR2_ETR	Вход внешнего события таймера 2
		Переопределенная	TMR3_ETR	Вход внешнего события таймера 3
11	PB9	Порт ввода-вывода В разряд 9		
		Основная	nUART1CTS	Вход управления UART1
		Альтернативная	TMR2_BRK	Вход внешнего события таймера 2
		Переопределенная	TMR3_BRK	Вход внешнего события таймера 3
13	PB10	Порт ввода-вывода В разряд 10		
		Основная	TMR2_CH2	Вход-выход канала 2 таймера 2
		Альтернативная	UART4_RXD	Входные данные UART4
		Переопределенная	—	—
14	PB11	Порт ввода-вывода В разряд 11		
		Основная	TMR2_CH2N	Инверсный выход канала 2 таймера 2
		Альтернативная	UART4_TXD	Выходные данные UART4
		Переопределенная	—	—
15	PB12	Порт ввода-вывода В разряд 12		
		Основная	TMR2_CH3	Вход-выход канала 3 таймера 2
		Альтернативная	—	—
		Переопределенная	—	—

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
16	PB13	Порт ввода-вывода В разряд 13		
		Основная	TMR2_CH3N	Инверсный выход канала 3 таймера 2
		Альтернативная	UART2_TXD	Выходные данные UART2
		Переопределенная	–	–
17	PB14	Порт ввода-вывода В разряд 14		
		Основная	TMR2_CH4	Вход-выход канала 4 таймера 2
		Альтернативная	UART2_RXD	Входные данные UART2
		Переопределенная	–	–
Порт С				
51	PC0/ MODE0	Порт ввода-вывода С разряд 0. Выбор режима работы микросхемы (при использовании загрузочной программы, описанной в подразделе 9.2 «Пример одного из возможных вариантов загрузочной программы»)		
		Основная	TMR4_ETR	Вход внешнего события таймера 4
		Альтернативная	–	–
		Переопределенная	TMR4_CH4	Вход-выход канала 4 таймера 4
52	PC1/ MODE1	Порт ввода-вывода С разряд 1. Выбор режима работы микросхемы (при использовании загрузочной программы, описанной в подразделе 9.2 «Пример одного из возможных вариантов загрузочной программы»)		
		Основная	TMR4_BRK	Вход внешнего события таймера 4
		Альтернативная	–	–
		Переопределенная	TMR4_CH4N	Инверсный выход канала 4 таймера 4
53	PC2	Порт ввода-вывода С разряд 2		
		Основная	TMR2_CH1	Вход-выход канала 1 таймера 2
		Альтернативная	SSP3_FSS	Вход-выход разрешения SPI 3
		Переопределенная	SCL	Выход синхросигнала I2C
54	PC3	Порт ввода-вывода С разряд 3		
		Основная	TMR2_CH1N	Инверсный выход канала 1 таймера 2
		Альтернативная	SSP3_CLK	Вход-выход синхросигнала SPI 3
		Переопределенная	SDA	Вход-выход данных I2C
55	PC4	Порт ввода-вывода С разряд 4		
		Основная	EXT_INT2	Вход прерывания 2
		Альтернативная	SSP3_RXD	Входные данные SPI 3
		Переопределенная	–	–
56	PC5	Порт ввода-вывода С разряд 5		
		Основная	EXT_INT3	Вход прерывания 3
		Альтернативная	SSP3_TXD	Выходные данные SPI 3
		Переопределенная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
57	PC6	Порт ввода-вывода C разряд 6		
		Основная	TMR2_ETR	Вход внешнего события таймера 2
		Альтернативная	SCL	Выход синхросигнала I2C
		Переопределенная	—	—
58	PC7	Порт ввода-вывода C разряд 7		
		Основная	TMR2_BRK	Вход внешнего события таймера 2
		Альтернативная	SDA	Вход-выход данных I2C
		Переопределенная	—	—
Порт D				
18	PD0	Порт ввода-вывода D разряд 0		
		Основная	SSP2_FSS	Вход-выход разрешения SPI 2
		Альтернативная	—	—
		Переопределенная	—	—
19	PD1	Порт ввода-вывода D разряд 1		
		Основная	SSP2_CLK	Вход-выход синхросигнала SPI 2
		Альтернативная	—	—
		Переопределенная	—	—
20	PD2	Порт ввода-вывода D разряд 2		
		Основная	SSP2_RXD	Входные данные SPI 2
		Альтернативная	—	—
		Переопределенная	—	—
21	PD3	Порт ввода-вывода D разряд 3		
		Основная	SSP2_TXD	Выходные данные SPI 2
		Альтернативная	—	—
		Переопределенная	—	—
22	PD4	Порт ввода-вывода D разряд 4		
		Основная	UART2_TXD	Выходные данные UART2
		Альтернативная	—	—
		Переопределенная	—	—
44	PD5	Порт ввода-вывода D разряд 5		
		Основная	UART2_RXD	Входные данные UART2
		Альтернативная	—	—
		Переопределенная	—	—
43	PD6	Порт ввода-вывода D разряд 6		
		Основная	UART3_RXD	Входные данные UART3
		Альтернативная	—	—
		Переопределенная	—	—
42	PD7	Порт ввода-вывода D разряд 7		
		Основная	UART3_TXD	Выходные данные UART3
		Альтернативная	—	—
		Переопределенная	—	—



Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
25	PD8	Порт ввода-вывода D разряд 8		
		Основная	UART4_RXD	Входные данные UART4
		Альтернативная	—	—
		Переопределенная	—	—
24	PD9	Порт ввода-вывода D разряд 9		
		Основная	UART4_TXD	Выходные данные UART4
		Альтернативная	—	—
		Переопределенная	—	—
23	PD10	Порт ввода-вывода D разряд 10		
		Основная	UART7816_TXD	Выходные данные UART7816
		Альтернативная	—	—
		Переопределенная	—	—
45	PD11	Порт ввода-вывода D разряд 11		
		Основная	UART7816_RXD	Входные данные UART7816
		Альтернативная	—	—
		Переопределенная	—	—
46	PD12	Порт ввода-вывода D разряд 12		
		Основная	UART7816_CLK	Выход синхросигнала UART7816
		Альтернативная	—	—
		Переопределенная	—	—
47	PD13	Порт ввода-вывода D разряд 13		
		Основная	UART7816_CTS	Вход управления UART7816
		Альтернативная	—	—
		Переопределенная	—	—
48	PD14	Порт ввода-вывода D разряд 14		
		Основная	UART7816_RTS	Выход управления UART7816
		Альтернативная	—	—
		Переопределенная	—	—
49	PD15	Порт ввода-вывода D разряд 15		
		Основная	EXT_INT1	Вход прерывания 1
		Альтернативная	LSE_OUT	Выход частоты генератора LSE
		Переопределенная	—	—

## **6 Указания по применению и эксплуатации**

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

Запрещается подведение каких-либо электрических сигналов (в том числе шин питания и общий) к выходам микросхем, неиспользуемым согласно схеме электрической.

Неиспользуемые входы микросхем должны быть подключены к шине питания или общий.

Порядок подачи и снятия напряжения питания и входных сигналов на микросхемы следующий:

- подача (включение микросхем) – общий», питание, входные сигналы или одновременно;
- снятие (выключение микросхем) – в обратном порядке или одновременно.

## 7 Система питания

Микросхема имеет несколько типов выводов питания:

$V_{CC}$  – основное питание микросхемы, включает питание пользовательских выводов, встроенного регулятора напряжения, PLL, генераторов;

$V_{CCB}$  – питание батарейного домена используется при отсутствии основного питания  $U_{CC}$  для питания батарейного домена и генератора LSE. Переключение с основного питания на батарейное происходит автоматически при снижении уровня  $U_{CC}$  ниже 2,0 В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 12 мс после превышения уровнем  $U_{CC}$  значения 2,0 В. Если в системе не требуется батарейное питание, вывод  $V_{CCB}$  должен быть объединен с  $V_{CC}$ ;

$V_{CCA}$  – питание аналоговых блоков сигма-дельта АЦП, АЦП последовательного приближения и формирователя опоры выведено на отдельные выводы для уменьшения помех создаваемых работой других блоков. На данные выводы должно подаваться напряжение с того же источника что и на  $V_{CC}$ , но при этом на печатной плате должны быть применены меры по снижению наводки помех. Для корректной работы АЦП и  $\Delta\Sigma$ АЦП входное напряжение должно быть в пределах от 3,0 до 3,6 В. Если значение входного напряжения находится в пределах от 2,2 до 3,0 В, то корректная работа АЦП и  $\Delta\Sigma$ АЦП не гарантируется;

**GND** – основная земля питания;

**EXPOSED PAD** – земля аналогового питания  $U_{CCA}$  и цифрового питания  $U_{CC}$ . Данные выводы должны соединяться с GND.

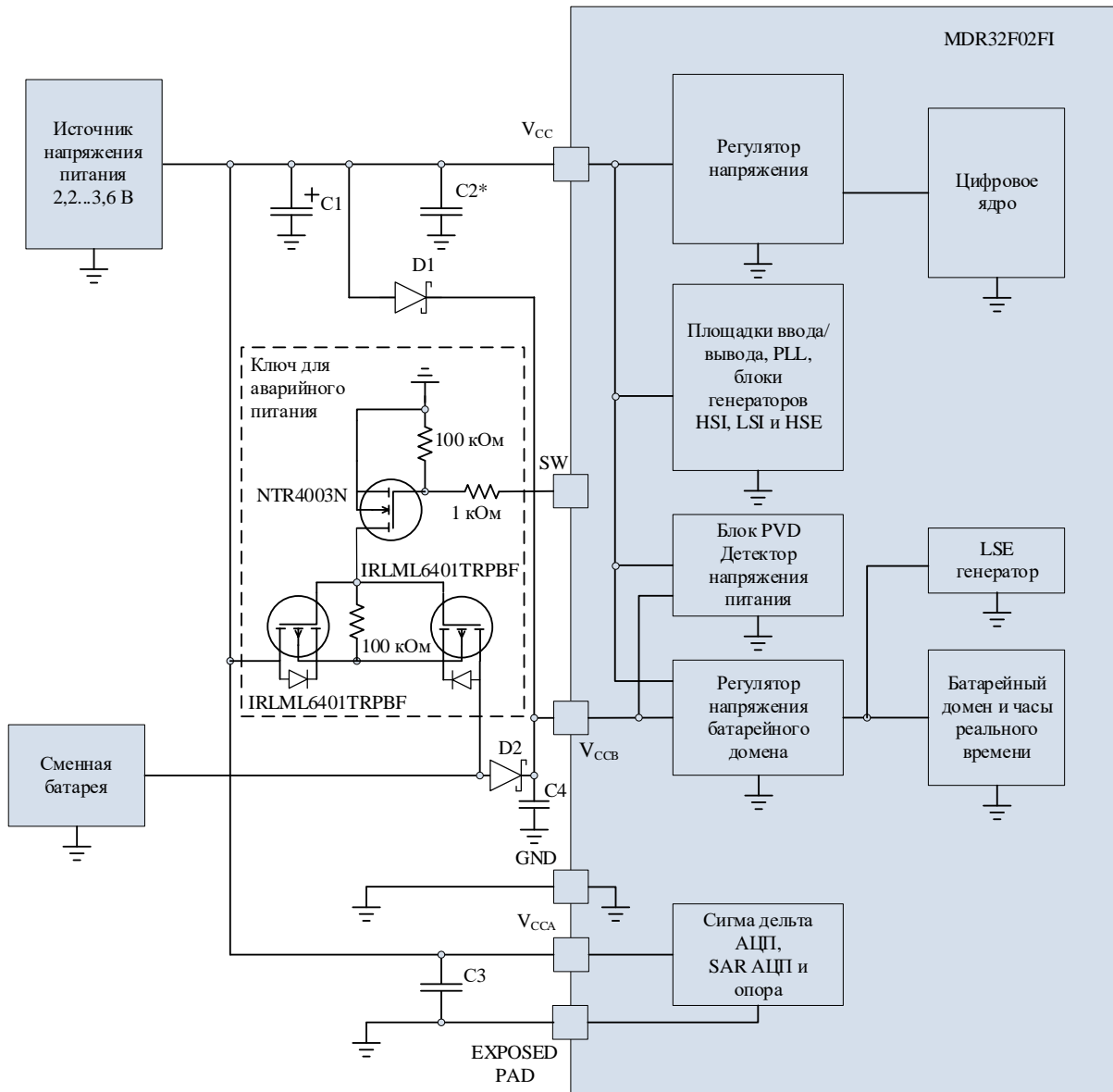
Если используется отдельное батарейное питание  $U_{CCB}$ , то после подачи напряжения питания на вывод  $V_{CCB}$  в пользовательской программе необходимо однократно выполнить следующие действия:

- установить в регистре ВКР\_LDO бит `bldo_boost` в «1»;
- через время не менее 50 мкс сбросить в регистре ВКР\_LDO бит `bldo_boost` в «0».

Во время, когда бит `bldo_boost` установлен в «1», необходимо контролировать напряжение питания  $U_{CC}$ , например, с помощью блока PVD, чтобы при экстренном отключении питания  $U_{CC}$  сбросить бит `bldo_boost` в «0», так как установка бита `bldo_boost` в «1» на длительное время приведет к повышенному потреблению батарейного домена.

Если отдельное батарейное питание  $U_{CCB}$  не используется, и вывод  $V_{CCB}$  объединен с  $V_{CC}$ , то выполнять переключение бита `bldo_boost` в регистре ВКР\_LDO не требуется.

## 7.1 Структурная схема подачи питания



- C1 – конденсатор емкостью 22 мкФ;
- C2 – конденсаторы емкостью 1 мкФ;
- C3 – конденсаторы емкостью 0,1 мкФ;
- C4 – конденсатор емкостью 10 мкФ.

\* Конденсаторы должны быть установлены у каждого вывода питания.

D1, D2 – диоды Шоттки. Рекомендуется выбирать диоды Шоттки с минимально возможным падением напряжения.

Рисунок 4 – Структурная схема подачи питания без защиты батареи

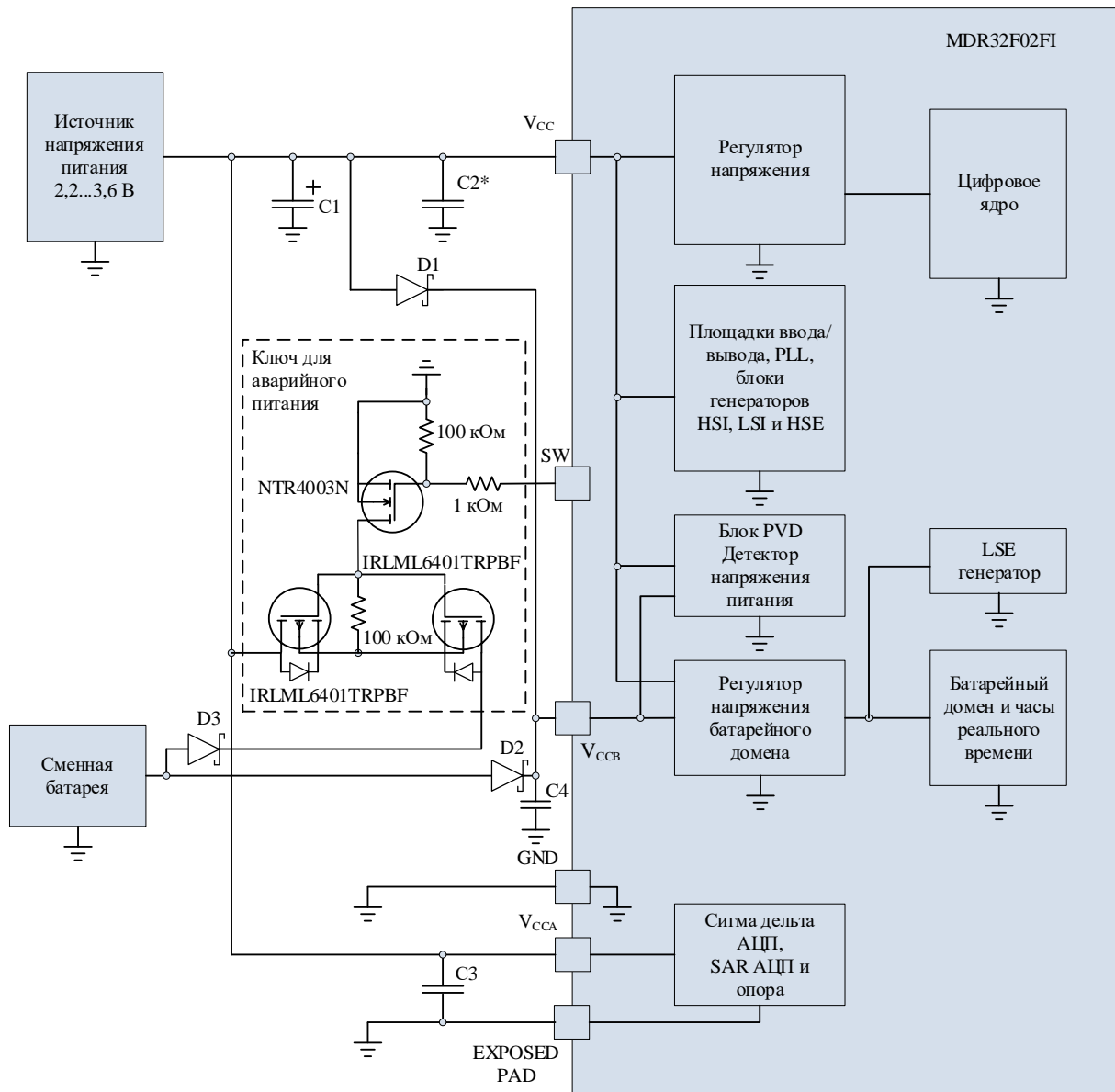
### Примечания

1 Если используется  $\Delta\Sigma$ АЦП или АЦП последовательного приближения, то напряжение питания  $U_{CC}$  ( $U_{CCA}$ ) должно быть в пределах от 3,0 до 3,6 В.

2 При отсутствии батареи вывод  $V_{CCSV}$  должен быть объединен с  $V_{CC}$ . Диоды Шоттки D1, D2, а также ключ для аварийного питания в этом случае могут не устанавливаться;

3 Если в системе не требуется аварийное питание, то ключ для аварийного питания может не устанавливаться, выход SW при этом допускается оставлять неподключенным.

4 Данная схема подачи питания реализует минимальное падение напряжения на внешнем ключе при аварийном питании от батареи. Однако этот вариант схемы не предусматривает защиту батареи в случае, если во время работы на аварийном питании от батареи (ключ замкнут) будет выполнено включение основного источника напряжения питания.



- C1 – конденсатор емкостью 22 мкФ;
- C2 – конденсаторы емкостью 1 мкФ;
- C3 – конденсаторы емкостью 0,1 мкФ;
- C4 – конденсатор емкостью 10 мкФ.

\* Конденсаторы должны быть установлены у каждого вывода питания.

D1, D2, D3 – диоды Шоттки. Рекомендуется выбирать диоды Шоттки с минимально возможным падением напряжения.

Рисунок 5 – Структурная схема подачи питания с защитой батареи

Примечания

1 Если используется  $\Delta\Sigma$ АЦП или АЦП последовательного приближения, то напряжение питания  $U_{CC}$  ( $U_{CCA}$ ) должно быть в пределах от 3,0 до 3,6 В.

2 При отсутствии батареи вывод  $V_{CCSV}$  должен быть объединен с  $V_{CC}$ . Диоды Шоттки D1, D2, D3, а также ключ для аварийного питания в этом случае могут не устанавливаться.

3 Если в системе не требуется аварийное питание, то ключ для аварийного питания и диод D3 могут не устанавливаться, выход SW при этом допускается оставлять неподключенным.

4 Данная схема подачи питания реализует защиту батареи с помощью диода D3 в случае, если во время работы на аварийном питании от батареи (ключ замкнут) будет выполнено включение основного источника напряжения питания. Однако при работе на аварийном питании из-за падения напряжения на диоде D3 напряжение батареи, поступающее на выводы  $V_{CC}$  и  $V_{CCA}$ , уменьшается, таким образом минимальное рабочее напряжение батареи должно составлять  $U_{CC\_MIN} + U_{D3}$ , где  $U_{CC\_MIN}$  – это нижний предел напряжения питания  $U_{CC}$  (см. таблицу 590),  $U_{D3}$  – падение напряжения на диоде D3.

Микросхема имеет встроенный детектор напряжения питания, подробнее см. раздел 19 «Детектор напряжения питания».

## 7.2 Схема сброса при включении и выключении основного питания

При включении питания вырабатывается внутренний сигнал сброса POR для цифровой части, питание  $U_{CC}$  нарастает и, пока оно не превысило уровень 2,0 В, сигнал сброса POR удерживается; после превышения данного уровня сигнал POR выдается еще на протяжении  $\sim 12$  мс для того, чтобы гарантированно установилось напряжение питания, после чего сигнал POR снимается, и схема может начать работать.

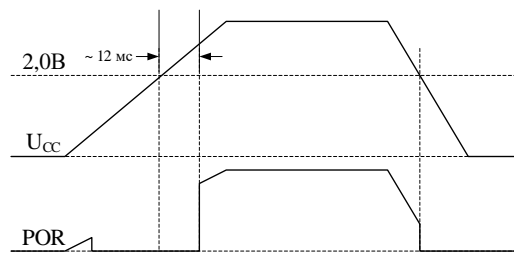


Рисунок 6 – Диаграмма формирования сброса микросхемы

При снижении напряжения питания  $U_{CC}$  ниже уровня 2,0 В сигнал POR вырабатывается без задержки.

Сигнал POR также служит для переключения питания батарейного домена между  $U_{CCB}$  и  $U_{CC}$ .

При включении основного напряжения питания  $U_{CC}$  автоматически включается встроенный регулятор напряжения для формирования напряжения питания цифрового ядра.

Начальная установка микросхемы может быть произведена внешним сигналом сброса nRESET, внутренними сигналами сброса сторожевых таймеров или внутренним сигналом сброса  $n\text{dmreset}^*$ , сформированным отладчиком. При этом сигнал nRESET формируется специальной схемой сброса, содержащей фильтр «иглолок» и одновибратор для увеличения длительности этого сигнала.

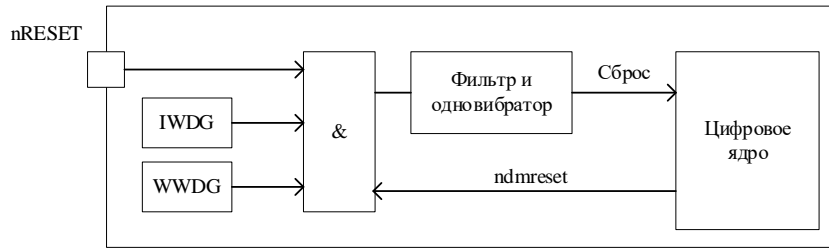


Рисунок 7 – Блок-схема формирования сброса микросхемы

При подаче на вход nRESET импульсов сброса длительностью менее 10 нс эти импульсы 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.

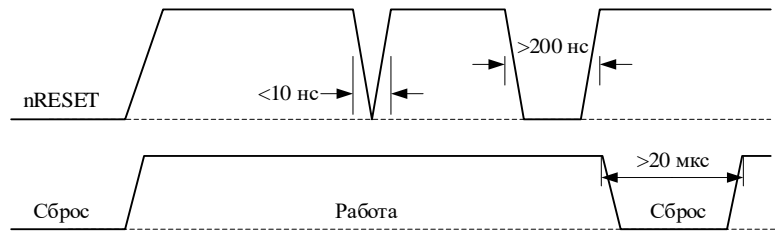


Рисунок 8 – Диаграмма фильтрации помех при формировании сброса

\* ndmreset (non-debug module reset) - сигнал сброса от подсистемы отладки, см. спецификацию RISC-V External Debug Support Version 0.13.

## 8 Организация памяти

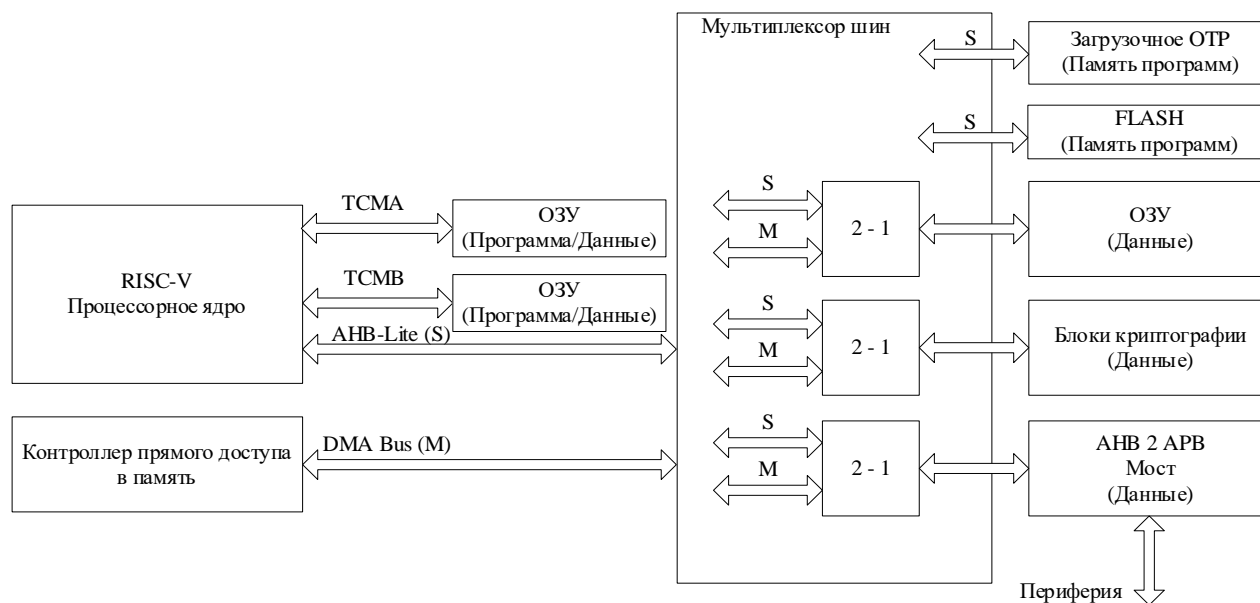


Рисунок 9 – Структурная схема организации памяти

Процессорное ядро имеет две системные шины:

- ANV-Lite – шина выборки инструкций и данных.
- TCMA, TCMB – шина выборки инструкций/данных.

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

Все адресное пространство микросхемы разделено на пять основных секций: Debug, CLINT, PLIC, I/O и TCM. В данное адресное пространство отображаются различные модули памяти и периферии.



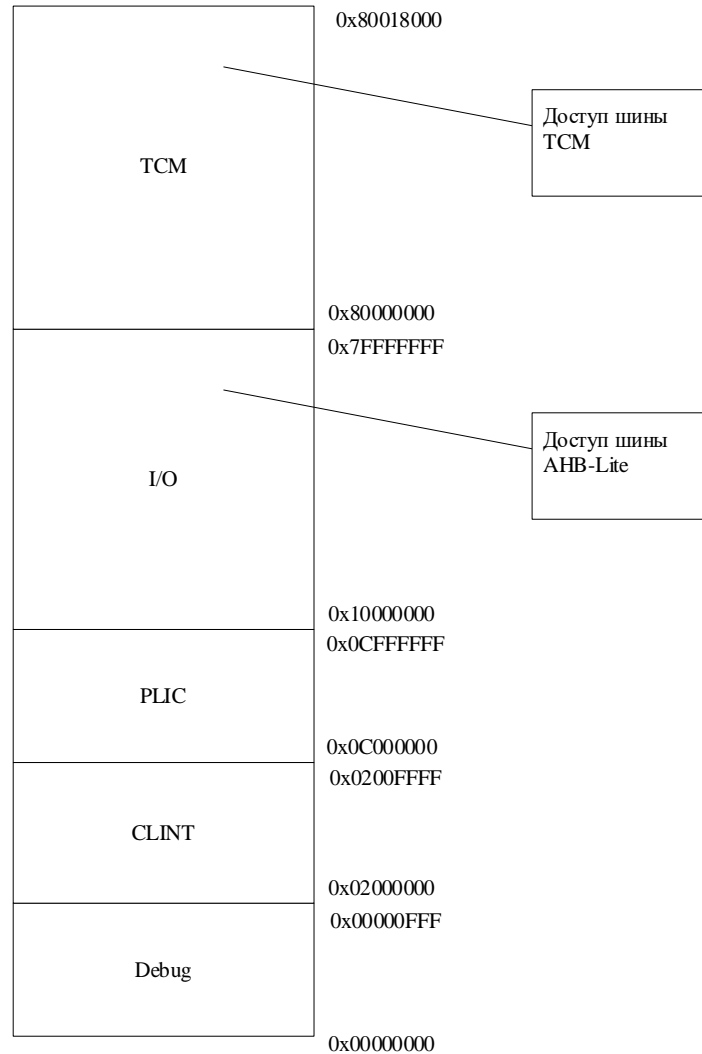


Рисунок 10 – Структура адресного пространства микросхемы

### 8.1 Базовые адреса процессора

Таблица 3 – Базовые адреса процессора

Адрес	Размер	Блок	Примечание
Память программ			
0x0002_0000		BOOT OTP	Загрузочная программа
0x1000_0000		FLASH	Область FLASH-памяти программ с пользовательской программой
Память данных			
0x2000_0000		SYSTEM RAM	Область внутреннего ОЗУ шины АНВ
Периферия			
0x4000_0000		SSP1	Регистры контроллера интерфейса SSP1
0x4000_8000		UART1	Регистры контроллера интерфейса UART1
0x4001_0000		UART2	Регистры контроллера интерфейса UART2
0x4001_8000		FLASH_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		I2C	Регистры контроллера интерфейса I2C
0x4003_8000		UART3	Регистры контроллера интерфейса UART3
0x4004_0000		ADC	Регистры управления АЦП

Адрес	Размер	Блок	Примечание
0x4004_8000		WWDG	Регистры контроллера сторожевого таймера WWDG
0x4005_0000		IWDG	Регистры контроллера сторожевого таймера IWDG
0x4005_8000		POWER	Регистры детектора напряжения питания
0x4006_0000		BKP	Регистры доступа и управления батарейным доменом
0x4006_8000		ADCU	Регистры управления $\Delta\Sigma$ АЦП
0x4007_0000		TIMER1	Регистры управления Таймер 1
0x4007_8000		TIMER2	Регистры управления Таймер 2
0x4008_0000		PORTA	Регистры управления порта А
0x4008_8000		PORTB	Регистры управления порта В
0x4009_0000		PORTC	Регистры управления порта С
0x4009_8000		CRC	Регистры управления аппаратного блока вычисления CRC
0x400A_0000		-	-
0x400A_8000		SENSORS	Регистры блока обработки датчиков безопасности
0x400B_0000		CLK_MEASURE	Регистры блока датчика частоты
0x400B_8000		RANDOM	Регистры блока генератора случайных чисел
0x400C_0000		ISO7816	Регистры контроллера интерфейса UART (ISO7816)
0x400C_8000		SSP2	Регистры контроллера интерфейса SSP2
0x400D_0000		SSP3	Регистры контроллера интерфейса SSP3
0x400D_8000		TIMER3	Регистры управления Таймер 3
0x400E_0000		TIMER4	Регистры управления Таймер 4
0x400E_8000		UART4	Регистры контроллера интерфейса UART4
0x400F_0000		PORTD	Регистры управления порта D
CRYPTO REGION			
0x6000_0000		-	-
0x6000_4000		L_BLOCK	Регистры блока прямого и обратного L преобразования
0x6000_8000		S_BLOCK0	Регистры блока замены S-box0
0x6000_C000		S_BLOCK1	Регистры блока замены S-box1
0x6001_0000		S_BLOCK2	Регистры блока замены S-box2
0x6001_4000		S_BLOCK3	Регистры блока замены S-box3
0x6001_8000		S_BLOCK4	Регистры блока замены S-box4
0x6001_C000		S_BLOCK5	Регистры блока замены S-box5
0x6002_0000		S_BLOCK6	Регистры блока замены S-box6
0x6002_4000		S_BLOCK7	Регистры блока замены S-box7
0x6002_8000		P_BYTE	Регистры блока байтовой замены p-byte
0x6002_C000		P_BIT0	Регистры блока битовой замены p-bit0
0x6003_0000		P_BIT1	Регистры блока битовой замены p-bit1
0x6003_4000		P_BIT2	Регистры блока битовой замены p-bit2
0x6003_8000		P_BIT3	Регистры блока битовой замены p-bit3
0x7000_0000		OTP	Регистры управления блоком OTP и адресное пространство однократно программируемой памяти
SYSTEM REGION			
0x8000_0000		SYSTEM RAM	Область внутреннего ОЗУ шины TCMA
0x8001_0000		SYSTEM RAM	Область внутреннего ОЗУ шины TCMB

## 9 Загрузочное OTP и режимы работы микросхемы

После включения питания и снятия внутренних (POR) и внешних (nRESET) сигналов сброса микросхема начинает выполнять программу из загрузочной области BOOT OTP. Загрузочная область – это адресное пространство OTP-памяти. При поставке микросхемы имеют незапрограммированную OTP-память. Необходимо самостоятельно разработать или опционально запросить вариант кода загрузочной программы и загрузить его в OTP. В разделе 9.1 «Разработка кода загрузочной программы и его загрузка в OTP-память» приводится описание процесса разработки кода загрузочной программы с последующим программированием памяти OTP. Описание одного из возможных вариантов загрузочной программы (исходный код которого может опционально предоставляться по запросу) приведено в разделе 9.2 «Пример одного из возможных вариантов загрузочной программы».

Ячейки памяти OTP с адресом 0x0002\_0000/0x7000\_0000 и смещениями с 0x0000\_3FE0 по 0x0000\_3FFF зарезервированы и не могут использоваться для хранения пользовательских данных. Эти регистры содержат идентификационную информацию.

Таблица 4 – Описание специальных полей в памяти OTP

Базовый адрес	Смещение	Название	Описание
0x0002_0000 / 0x7000_0000	0x0000_3FE0 0x0000_3FE4 0x0000_3FE8 0x0000_3FEC 0x0000_3FF0	UNIQ_ID	Уникальный ID микросхемы
	0x0000_3FF4	BG_TRIM	Подстройка опорного напряжения bandgap.
	0x0000_3FF8	HSI_TRIM	Коэффициент подстройки частоты генератора HSI
	0x0000_3FFC	-	Биты защиты адресного пространства FLASH-памяти

Биты OTP 131064 – 131067 (последний байт) позволяют установить защищенные адресные пространства FLASH-памяти. Активное состояние бит защиты – логическая «1». Каждый бит защиты отвечает за выбор соответствующего защищенного адресного пространства FLASH-памяти:

- бит 131064 – диапазон адресов 0x1000\_0000-0x1000\_FFFF;
- бит 131065 – диапазон адресов 0x1001\_0000-0x1001\_FFFF;
- бит 131066 – диапазон адресов 0x1002\_0000-0x1002\_FFFF;
- бит 131067 – диапазон адресов 0x1003\_0000-0x1003\_FFFF.

При выборе хотя бы одного защищенного адресного пространства FLASH-памяти включается:

- защита выбранных адресных пространств FLASH-памяти;
- защита области BOOT OTP (0x0002\_0000-0x0003\_FFFF) и OTP (0x7000\_0000-0x7FFF\_FFFF);

- защита области периферийных блоков криптографии (0x6000\_0000-0x6FFF\_FFFF);
- блокировка интерфейса JTAG;
- блокировка выполнения операций стирания страницы, записи слова и чтения слова для выбранных защищенных адресных пространств FLASH-памяти в режиме программирования FLASH-памяти;
- блокировка выполнения операции стирания всей FLASH-памяти или 256 Кбайт основной области FLASH-памяти в режиме программирования FLASH-памяти.

Защита адресных пространств FLASH-памяти, BOOT OTP, OTP и периферийных блоков криптографии выполняется следующим образом:

- если программа выполняется из защищенного адресного пространства FLASH-памяти или области BOOT OTP, то разрешается доступ по чтению ко всем областям FLASH-памяти, областям BOOT OTP и OTP, а также доступ по чтению и записи к области периферийных блоков криптографии;
- если программа выполняется из незащищенного адресного пространства FLASH-памяти или любой области ОЗУ, то блокируется доступ по чтению к защищенным адресным пространствам FLASH-памяти и областям BOOT OTP и OTP (при чтении возвращаются нули), также блокируется доступ по чтению и записи к области периферийных блоков криптографии (при чтении возвращаемое значение неопределено, запись не выполняется).

По любому сбросу, кроме как от сигнала ndmreset, для контроллера OTP устанавливается 1 такт паузы для доступа к памяти OTP по шине АНВ за данными и инструкциями, что обеспечивает необходимые задержки доступа для выполнения программы из загрузочной области BOOT OTP. Перед сбросом от сигнала ndmreset для используемой тактовой частоты должно быть вычислено и записано необходимое число тактов паузы для доступа к памяти OTP по шине АНВ, так как регистры контроллера RST\_CLK и модуля OTP не сбрасываются.

## **9.1 Разработка кода загрузочной программы и его загрузка в OTP-память**

Процесс разработки загрузочной программы и ее загрузка в память OTP может состоять из следующих этапов:

- 1 Разработка и отладка кода загрузочной программы из адресов FLASH-памяти или ОЗУ.
- 2 Компиляция загрузочной программы для адресного пространства BOOT\_OTP, начинающегося с адреса 0x0002\_0000, и получение выходного файла в формате Intel HEX.
- 3 Получение массива 32-разрядных данных из выходного файла в формате Intel HEX.
- 4 Разработка программы, осуществляющей программирование OTP-памяти массивом данных.
- 5 Программирование памяти OTP массивом данных, полученном на этапе 3, с последующей верификацией.

Выбор адресного пространства, в котором располагается компилируемая загрузочная программа, задается в специальном linker-файле («Linker configuration file») в настройках среды разработки. В среде разработки IAR linker-файл имеет расширение .icf, в среде Eclipse – расширение .ld. Для компиляции программы в адресном пространстве памяти OTP можно, например, изменить исходный linker-файл, указав адресное пространство памяти OTP вместо адресного пространства FLASH.

Пример для среды IAR:

```
define region FLASH_region32 = mem:[from 0x00020000 to 0x00023FFF];
```

Пример для среды Eclipse:

```
FLASH (rx) : ORIGIN = 0x00020000, LENGTH = 16K
```

Опция, позволяющая сформировать выходной файл в формате Intel HEX из скомпилированного образа, выбирается в настройках проекта в используемой среде программирования. Из полученного выходного файла в формате Intel HEX необходимо получить массив 32-разрядных данных следующего вида:

```
const int mas[]=
{
    0xf1402d73,
    0x000d0363,
    0x4081a001,
    0x41814101,
    .....
    0xfca42c23,
    0xa001bbcd
};
```

Выполнение программы, которая реализует программирование памяти OTP массивом данных, необходимо осуществлять из памяти ОЗУ или FLASH.

Пример программы, осуществляющей программирование памяти OTP массивом данных, приведен ниже:

```
#define OTP_BASE ((uint32_t)0x70000000U)
#define OTP_MEM_BASE (OTP_BASE)
#define OTP_REGS_BASE (OTP_BASE + 0x00010000)
#define OTP_MEM ((uint32_t*)OTP_MEM_BASE)
#define OTP_REG ((MDR_OTP_TypeDef*)OTP_REGS_BASE)

/* Массив образа загрузочной программы */
const int mas[]=
{
    0xf1402d73,
    0x000d0363,
    0x4081a001,
    .....
    0xa001bbcd
};
```

```

void error()
{
    /*Код в случае несовпадения данных */
}

int main (void)
{
    int a, i, j, temp, busy;
    MDR_RST_CLK->PER2_CLOCK=0xFFFFFFFF;
    MDR_RST_CLK->HS_CONTROL=1;
    while((MDR_RST_CLK->CLOCK_STATUS & 4)!=4);
    MDR_RST_CLK->CPU_CLOCK=(1<<8) | 2;
    OTP_REG->STAT_CTRL = 0;
    OTP_REG->READ_PROTECT = 0;
    OTP_REG->WRITE_PROTECT = 0;
    OTP_REG->DELAY0=0 | (0<<8) | (7<<16) | 0<<4;
    OTP_REG->DELAY1=127;
    /* Запись данных в OTP */
    for (i=0;i<(sizeof(mas)/sizeof(mas[0]));i++)
    {
        temp = mas[i];
        for (j=0;j<32;j++)
        {
            OTP_REG->RW_CMD = (((temp>>j)&1)<<24) | (32*i+j);
            OTP_REG->RW_CMD |= 1<<29;
            do
            {
                busy = (OTP_REG->STAT & 0x80);
            } while(busy != 0);
        }
    }
    /* Чтение данных из OTP – верификация запрограммированных данных */
    for (i=0;i<(sizeof(mas)/sizeof(mas[0]));i++)
    {
        a=OTP_MEM[i];
        if(a!=mas[i]) error();
    }
    while(1);
}

```

Описанный выше код осуществляет настройку модуля OTP на работу при частоте HSE = 8 МГц (при использовании другой тактовой частоты необходимо установить соответствующие задержки для модуля OTP в регистрах DELAY\_0 и DELAY\_1 – см. раздел 38 «Модуль однократно программируемой памяти (OTP)»), программирование памяти OTP массивом данных mas[] и верификацию записанных данных с помощью сравнения изначального массива со считываемыми данными. При несовпадении данных программа переходит в функцию error(). Если программа завершила свое выполнение без перехода в функцию error(), то память OTP запрограммирована массивом mas[] успешно.

## 9.2 Пример одного из возможных вариантов загрузочной программы\*

В загрузочной программе микросхема определяет, в каком из режимов она будет функционировать, и переходит в этот режим. Перед определением режима функционирования загрузочная программа анализирует бит FPOR в регистре ВКР\_LDO:

– FPOR = 0: идет выполнение программы после сброса по питанию, подано основное ( $U_{CC}$ ,  $U_{CCA}$ ) и батарейное питание ( $U_{CCB}$ ). Режим функционирования определяется внешними выводами MODE[1:0] (PC[0], PC[1]), при этом перед опросом состояния этих выводов для них включается внутренняя подтяжка к шине общий (встроенные резисторы подтяжки к шине общий имеют сопротивление ~85 кОм). После определения режима функционирования состояние выводов MODE[1:0] записывается в биты MODE[1:0] регистра ВКР\_LDO, а бит FPOR устанавливается в «1». Данное состояние бита FPOR сохраняется до отключения основного и батарейного питания, при условии, что бит FPOR не будет изменен в пользовательской программе.

– FPOR = 1: идет выполнение программы после системного сброса, батарейное питание ( $U_{CCB}$ ) не отключалось. После перезапуска микросхемы уровни на выводах MODE[1:0] не влияют на режим функционирования микросхемы, так как в этом случае режим работы будет прочитан из битов MODE[1:0] регистра ВКР\_LDO.

В пользовательской программе выводам PC[0], PC[1] пользователем могут присваиваться функции самостоятельно.

Таблица 5 – Режимы работы микросхемы

MODE[1:0]	Режим	Стартовый адрес/ таблица векторов прерываний	Описание
00	Режим отладки	0x10000000	Процессор начинает выполнять программу из внутренней OTP, а затем передает управление во FLASH-память программ. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в OTP-памяти
01 или 10	UART-загрузчик	Определяется пользователем	Микросхема через интерфейс UART1 на выводах PB[1], PB[0] получает код программы в ОЗУ для исполнения. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в OTP-памяти
11	Режим отладки	0x80000000	Процессор начинает выполнять программу из внутренней OTP, а затем передает управление в ОЗУ-память программ. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в OTP-памяти

\* Исходный код данного варианта загрузочной программы может предоставляться по запросу, отправленному на support@milandr.ru.

При работе в режиме отладки разрешается работа отладочного интерфейса JTAG. При этом к микросхеме может быть подключен JTAG-адаптер, с помощью которого программные средства разработки позволяют работать с микросхемой 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.

Ячейка памяти OTP размером 1 байт по адресу 0x0002\_3FDF/0x7000\_3FDF используется для хранения номера версии загрузочной программы.

### 9.2.1 UART-загрузчик

Режим UART-загрузчика предоставляет 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 (FLASH, OTP, 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.

При загрузке программы в ОЗУ необходимо учитывать, что часть ОЗУ используется загрузочной программой для хранения следующих объектов:

- 0x8000\_0000-0x8000\_00FF – таблица векторов прерываний;
- 0x8000\_FE00-0x8000\_FFFF – стек;
- 0x8001\_0000-0x8001\_016D – код обработчиков прерываний.

#### 9.2.1.1 Параметры связи по UART

Для связи по UART выбраны следующие параметры канала связи:

- начальная скорость – 9600 бод;
- количество 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.



### 9.2.1.2 *Протокол обмена по UART*

После синхронизации с Master загрузчик переходит в диспетчер команд. Таким образом, Master-у доступны команды, приведенные в таблице 6.

### 9.2.1.3 *Синхронизация с внешним устройством*

Начальные условия.

На этапе синхронизации с внешним устройством (Master) вывод Rx используется как вход. Master постоянно посылает в канал синхросимвол – 0. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ошибки обмена. Как только Загрузчик настроил скорость, он переходит в диспетчер команд и выдает Master-у приглашение (три байта – 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>')).

Master завершает выдачу синхросимволов и теперь может подавать команды, согласно протоколу обмена.

## 9.2.2 **Команды UART загрузчика**

Таблица 6 – 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

### 9.2.2.1 *Команда CMD\_SYNC*

Пустая команда.

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

Таблица 7 – Команда CMD\_SYNC

Код команды	CMD_SYNC = 0x00
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
<b>Формат команды:</b>	
Master: Выдает код команды CMD_SYNC	Slave: если команда принята с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды

**9.2.2.2 Команда CMD\_CR**

Выдача приглашения Master-у.

Таблица 8 – Команда CMD\_CR

Код команды	CMD_CR = 0x0D
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
<b>Формат команды:</b>	
Master: Выдает код команды CMD_CR	Slave: если команда принята с ошибками, то выдает сообщение об об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды. Иначе выдает три байта: – код команды CMD_CR; – код 0x0A; – код 0x3E (ASCII символ '>')

**9.2.2.3 Команда CMD\_BAUD**

Установка скорости обмена.

Таблица 9 – Команда CMD\_BAUD

Код команды	CMD_BAUD = 0x42
ASCII символ, соответствующий коду команды	'B'
Количество параметров команды	1
Параметр	Новое значение скорости обмена [бод]
<b>Формат команды:</b>	
Master: Выдает код команды CMD_BAUD	Slave: если команда принята с ошибками, то выдает сообщение об ошибке об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр	Slave: Если параметр принят с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_BAUD и завершает обработку текущей команды. Иначе: – выдает код команды CMD_BAUD; – после передачи кода команды CMD_BAUD устанавливает новое значение скорости обмена

### 9.2.2.4 Команда CMD\_LOAD

Загрузка массива байт в память микросхемы.

Таблица 10 – Команда CMD\_LOAD

<b>Код команды</b>	<b>CMD_LOAD = 0x4C</b>
ASCII символ, соответствующий коду команды	'L'
Количество параметров команды	2
Параметр 1	Адрес памяти приемника данных
Параметр 2	Размер массива в байтах
<b>Формат команды:</b>	
Master: Выдает код команды CMD_LOAD	Slave: если команда принята с ошибками, то выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр 1	Slave: ожидает получения всех параметров
Master: Выдает параметр 2	Slave: если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке с типом ERR_CHN и завершает обработку текущей команды. Иначе 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 сообщение об ошибке с типом ERR_CHN и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончании принятия массива выдает код ответа REPLY_OK = 0x4B ('K')

### 9.2.2.5 Команда CMD\_VFY

Выдача массива байт из памяти микросхемы.

Таблица 11 – Команда CMD\_VFY

<b>Код команды</b>	<b>CMD_VFY = 0x59</b>
ASCII символ, соответствующий коду команды	'Y'
Количество параметров команды	2
Параметр 1	Адрес памяти источника данных
Параметр 2	Размер массива в байтах
<b>Формат команды:</b>	
Master: Выдает код команды CMD_VFY	Slave: если команда принята с ошибками, выдает сообщение об ошибке с типом ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр 1	Slave: ожидает получения всех параметров
Master: Выдает параметр 2	Slave: если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке с типом ERR_CHN и завершает обработку текущей команды. Иначе: – выдает код команды CMD_VFY; – выдает массив байт младшим байтом вперед; – по окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K')

### 9.2.2.6 Команда *CMD\_RUN*

Запуск программы на выполнение.

Таблица 12 – Команда *CMD\_RUN*

Код команды	<i>CMD_RUN</i> = 0x52
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр	Адрес первой команды загруженной программы
<b>Формат команды:</b>	
Master: Выдает код команды <i>CMD_RUN</i>	Slave: если команда принята с ошибками, то выдает сообщение об ошибке с типом <i>ERR_CHN</i> или <i>ERR_CMD</i> и завершает обработку текущей команды
Master: Выдает параметр	Slave: если параметр принят с ошибками, то выдает сообщение об ошибке с типом <i>ERR_CHN</i> и завершает обработку текущей команды. Иначе: – выдает код команды <i>CMD_RUN</i> ; – устанавливает значение PC согласно принятому адресу (SP и PLIC не перепрограммируются) и, таким образом, Slave завершает свое выполнение. Передача управления загруженной программе происходит после окончания отправки кода команды <i>CMD_RUN</i>

### 9.2.2.7 Прием параметров команды

Параметры команд – это четырехбайтные 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.

### 9.2.2.8 Сообщения об ошибках

Сообщения об ошибках – это двухбайтные 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, если UART установил в 'I' 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.

## 10 Контроллер FLASH-памяти программ

Микросхема содержит встроенную FLASH-память программ с объемом 256 Кбайт основной памяти программ и 8 Кбайт информационной памяти. В обычном режиме (бит CON = 0, регистр FLASH\_CMD) доступна основная память программ через системную шину для выборки инструкций и данных кода программы. В режиме программирования (бит CON = 1, регистр FLASH\_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 системной шины или ОЗУ. Выполнение программного кода из FLASH-памяти программ в режиме программирования невозможно.

### 10.1 Работа FLASH-памяти программ в обычном режиме

Скорость доступа во FLASH-память ограничена и составляет порядка 30 нс, в результате выдача новых значений из FLASH-памяти может происходить с частотой не более 30 МГц. Для того, чтобы процессорное ядро могло получать новые инструкции на больших частотах на каждом такте, из FLASH-памяти извлекаются избыточные 4 байта. И пока ядро выполняет инструкцию, из памяти извлекается следующая порция данных. Таким образом, тактовая частота может в два раза превышать частоты извлечения данных из памяти при линейном выполнении программы. При возникновении переходов в выполнении программы, 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 30 МГц пауза не требуется, так как FLASH-память успевает выдать новые данные за один такт, а при частоте от 30 до 60 МГц требуется один такт паузы. Число тактов паузы задается в регистре FLASH\_CMD битами Delay[2:0]. В таблице 13 приведены характеристики необходимой паузы для работы FLASH-памяти программ.

Таблица 13 – Характеристики паузы для работы FLASH-памяти программ

Delay[2:0]	Тактов паузы	Тактовая частота	Примечание
0b000	0	До 30 МГц	
0b001	1	До 60 МГц	

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

### 10.2 Работа FLASH-памяти программ в режиме программирования

Перед переводом памяти в режим программирования необходимо в регистр FLASH\_KEY записать комбинацию 0x8AAA5551. 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 ОЗУ.

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

- стирание всей памяти или 256 Кбайт;
- стирание страницы памяти размером 4 Кбайт;
- запись 32-битного слова в память;
- чтение 32-битного слова из памяти.

Структура памяти представлена на рисунке 11.

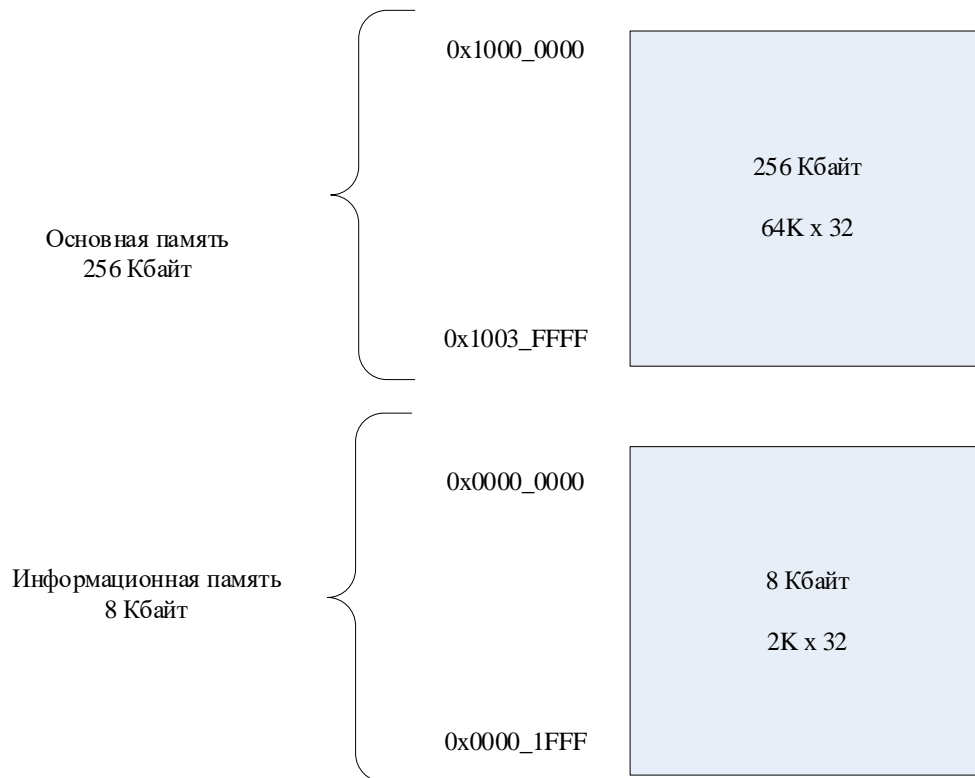


Рисунок 11 – Структура FLASH-памяти

### 10.2.1 Стирание всей памяти или 256 Кбайт основной памяти.

Стирание памяти возможно только в режиме программирования.

Примечание – При стирании информационной области автоматически стирается и основная.

Для стирания всей памяти необходимо:

- установить необходимое значение в бит IFREN (1 – для всей памяти или 0 – для основной памяти);
- затем установить биты XE, MAS1 и ERASE в «1»;
- через время  $T_{nvs}$  (не менее 5 мкс) установить бит NVSTR в «1»;
- полное стирание памяти длится время  $T_{me}$  (от 20 мс до 40 мс). Спустя это время необходимо очистить бит ERASE, и спустя время  $T_{nvhl}$  (не менее 100 мкс) очистить биты XE, MAS1 и NVSTR.

Последующие операции с памятью можно выполнять спустя время  $T_{rcv}$  (не менее 10 мкс).

Временная диаграмма стирания памяти представлена на рисунке 12.

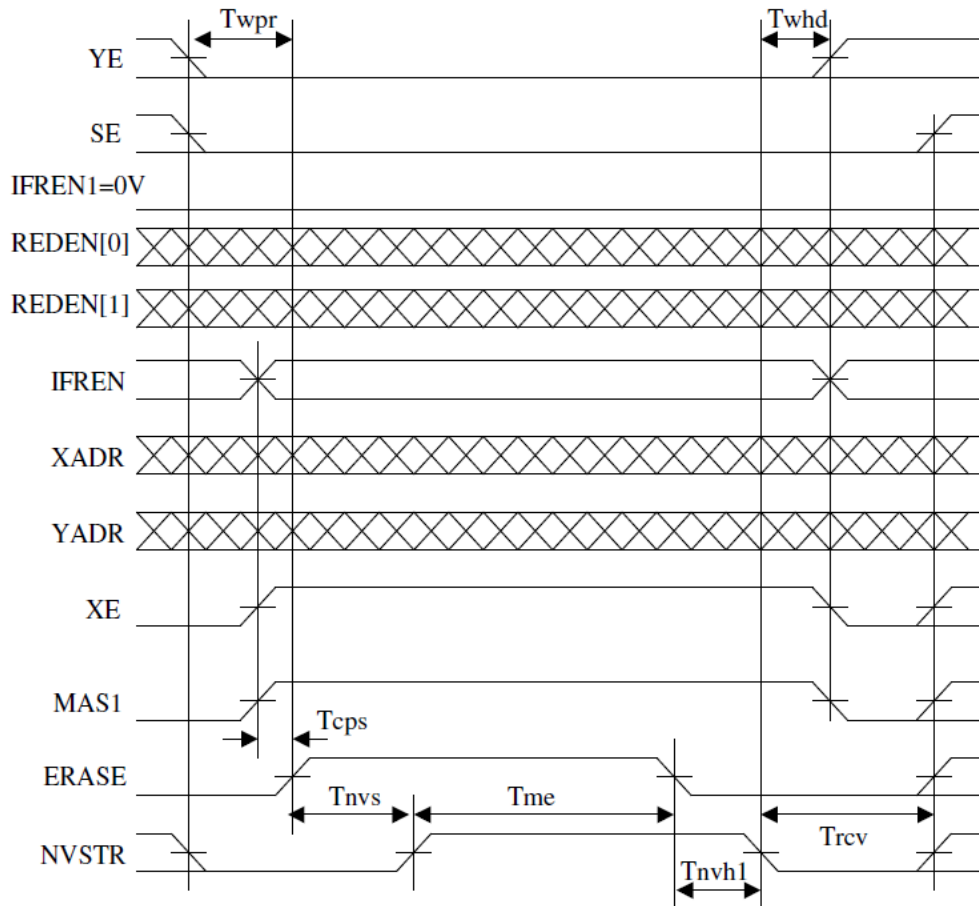


Рисунок 12 – Временная диаграмма стирания памяти

### 10.2.2 Стирание страницы памяти размером 4 Кбайт.

Стирание страницы памяти возможно только в режиме программирования. Для стирания страницы памяти необходимо:

- установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти);
- затем установить адрес стираемой страницы в регистре FLASH\_ADR и установить биты XE и ERASE в «1»;
- спустя время  $T_{nvs}$  (не менее 5 мкс) установить бит NVSTR в «1»;
- стирание страницы памяти длится время  $T_{erase}$  (от 20 мс до 40 мс). Спустя это время необходимо очистить бит ERASE, и спустя время  $T_{nvhl}$  (не менее 5 мкс) очистить биты XE и NVSTR.

Последующие операции с памятью можно выполнять спустя время  $T_{rcv}$  (не менее 10 мкс).

Временная диаграмма стирания страницы памяти представлена на рисунке 13.



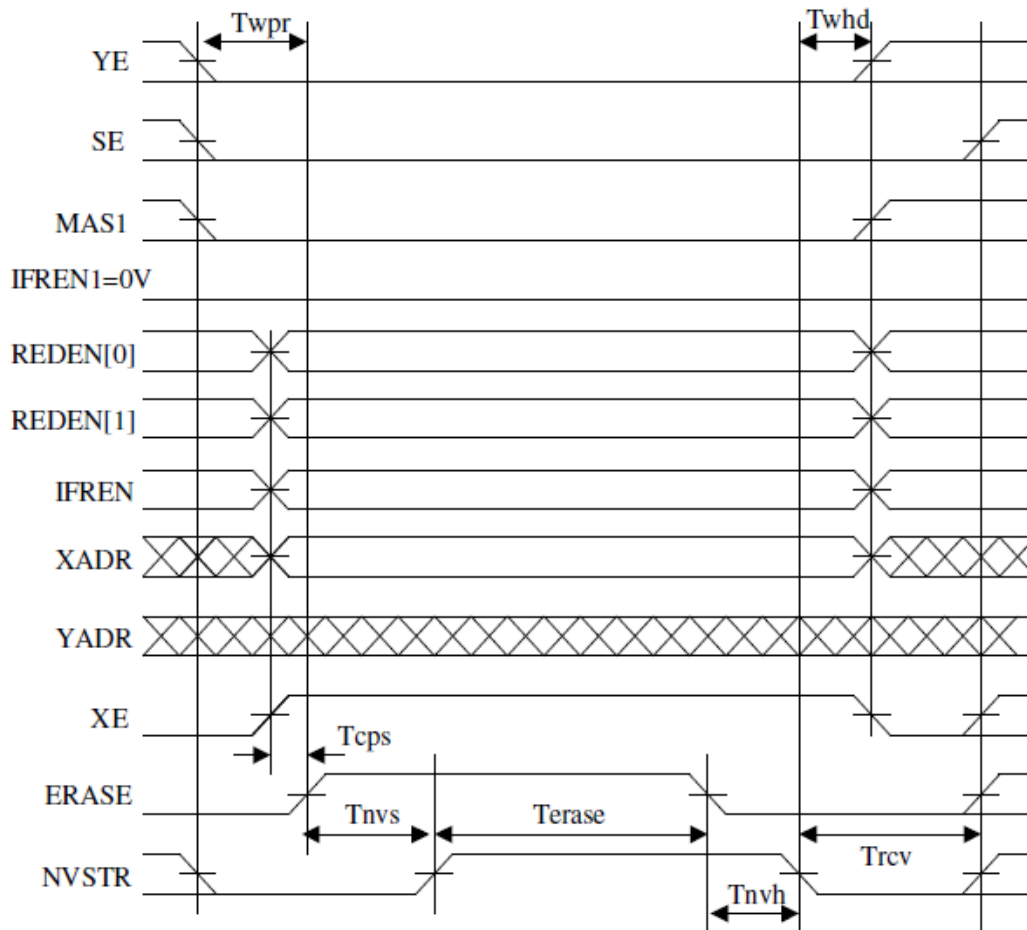


Рисунок 13 – Временная диаграмма стирания страницы памяти

### 10.2.3 Запись 32-битного слова в память

Запись в память возможна только в режиме программирования. Временная диаграмма записи памяти представлена на рисунке 14.

Для записи в память необходимо:

- установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти);
- затем установить адрес, по которому производится запись, в регистре FLASH\_ADR;
- в регистр FLASH\_DI записать записываемое в память слово и установить биты XE и PROG в «1»;
- через время  $T_{nvs}$  (не менее 5 мкс) установить бит NVSTR в «1»;
- через время  $T_{pgs}$  (не менее 10 мкс) установить бит YE в «1»;
- запись в память длится время  $T_{prog}$  (от 20 мкс до 40 мкс). Спустя это время необходимо очистить бит YE.

Если необходимо записать следующее слово в память, то:

- через время  $T_{adh}$  (не менее 20 нс) после очистки бита YE установить новый адрес YADR (ADR[8:2]) и значение для записи в другую ячейку памяти;
- через время  $T_{ads}$  (не менее 20 нс) установить YE в «1»;
- запись в память длится время  $T_{prog}$  (от 20 до 40 мкс). Спустя данное время необходимо очистить бит YE.

Если запись больше не требуется:

- через время  $T_{pgh}$  (не менее 20 нс) после очистки бита YE необходимо очистить бит PROG;
- через время  $T_{nvh}$  (не менее 5 мкс) очистить биты XE и NVSTR.
- последующие операции с памятью можно выполнять спустя время  $T_{rcv}$  (не менее 10 мкс).

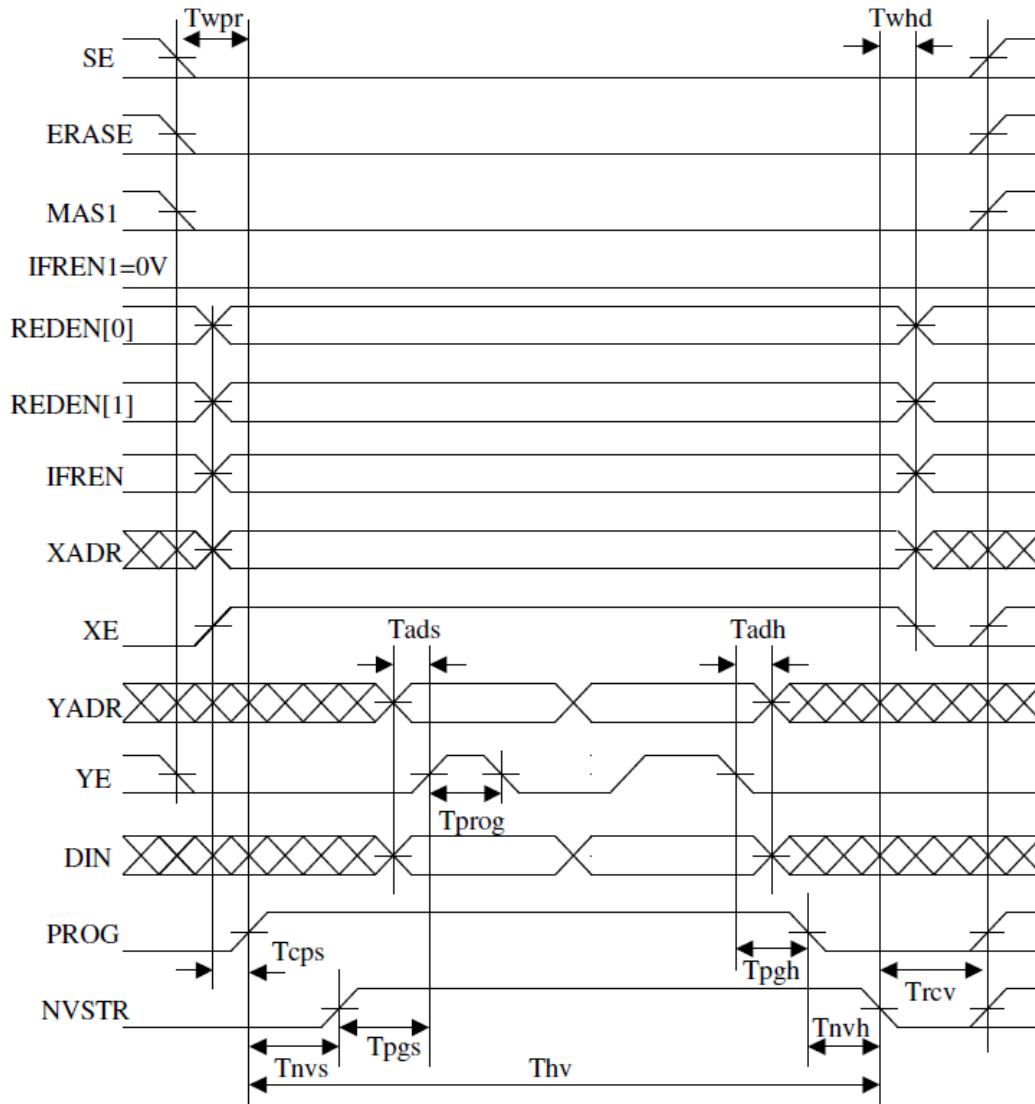


Рисунок 14 – Временная диаграмма записи памяти

Примечания

- 1 FLASH-память поддерживает не менее 10000 циклов перезаписи.
- 2 Время  $T_{hv}$  – суммарное время подачи высокого напряжения программирования на одну и ту же строку (один и тот же XADR) до следующего выполнения операции стирания этой же строки. Время  $T_{hv}$  должно составлять не более 16 мс.
- 3 Данные могут быть запрограммированы по одному и тому же адресу более двух раз, только если после второй операции программирования была выполнена операция стирания.
- 4 В одну и ту же битовую ячейку нельзя записать «0» более двух раз, если после второй записи не была выполнена операция стирания.

### 10.2.4 Чтение 32-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 необходимо:

- 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 FLASH\_ADR, и установить биты XE, YE и SE в «1»;
- через время  $T_{acc}$  (не менее 30 нс) из регистра FLASH\_DO можно считать данные.

Если необходимо считать следующее слово, то:

- в регистр FLASH\_ADR необходимо записать новый адрес;
- сбросить бит SE на время  $T_{nws}$  (не менее 5 нс), после этого установить SE;
- спустя время  $T_{acc}$  (не менее 30 нс) из регистра FLASH\_DO можно считать следующие данные.

Если чтение больше не требуется, то можно очистить все биты управления.

Временная диаграмма чтения памяти представлена на рисунке 15.

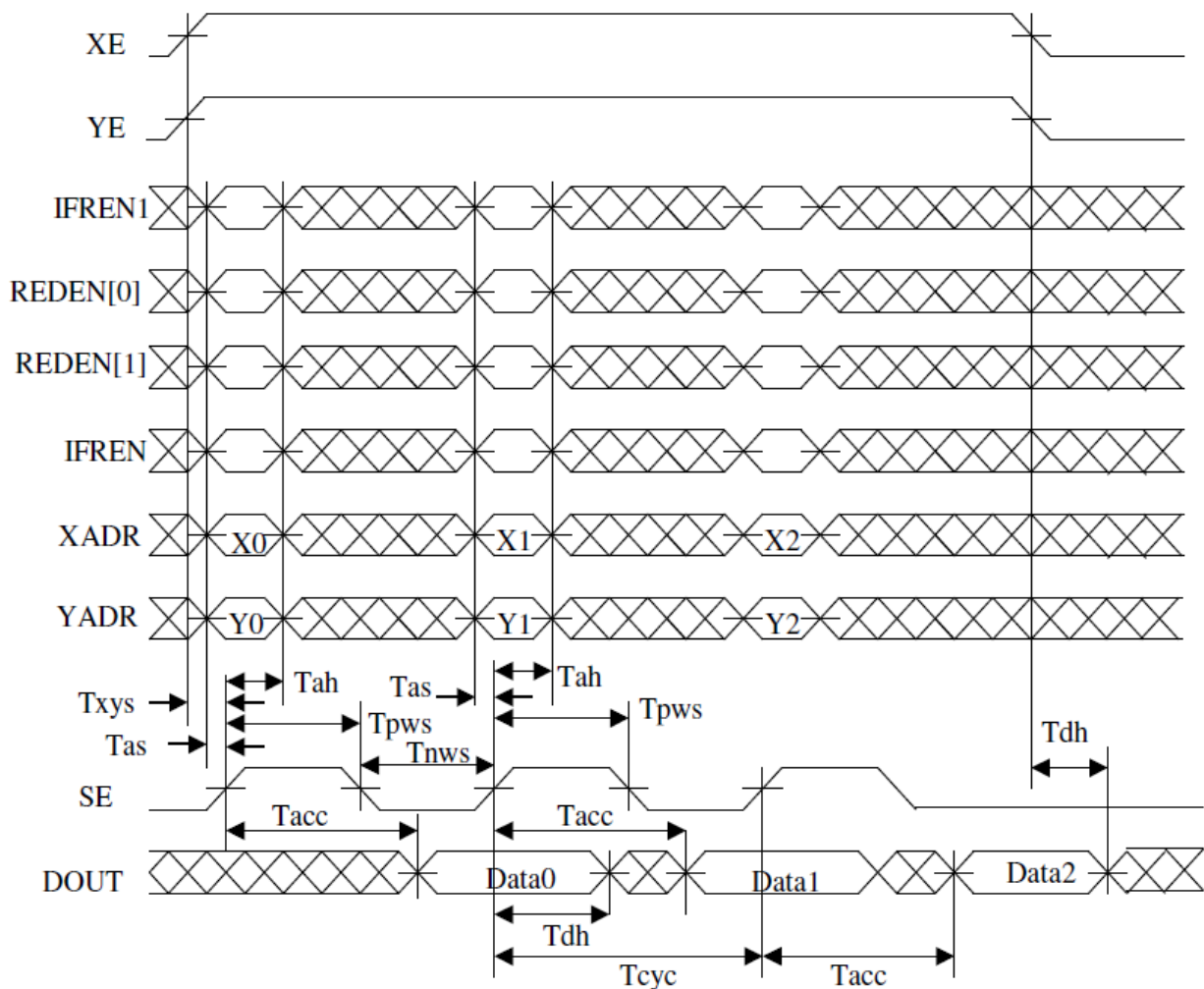


Рисунок 15 – Временная диаграмма чтения памяти

### 10.3 Регистры управления контроллера FLASH-памяти программ

Таблица 14 – Регистры управления контроллера FLASH-памяти программ

Базовый Адрес	Название	Описание
0x4001_8000	FLASH_CNTRL	Регистры контроллера FLASH-памяти программ
Смещение		
0x00	FLASH_CMD	Регистр управления FLASH-памятью
0x04	FLASH_ADR	Регистр адреса
0x08	FLASH_DI	Регистр данных на запись
0x0C	FLASH_DO	Регистр данных считанных
0x10	FLASH_KEY	Регистр ключа

#### 10.3.1 FLASH\_CMD

Таблица 15 – Регистр FLASH\_CMD

Номер	31...15	14	13	12	11	10	9	8
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	1	0	0	0	0	0	0
		TMR	NVSTR	PROG	MAS1	ERASE	IFREN	SE

Номер	7	6	5...3	2, 1	0
Доступ	R/W	R/W	R/W	U	R/W
Сброс	0	0	001	0	0
	YE	XE	Delay[2:0]		CON

Обозначения доступа:

R/W – бит доступен на чтение и запись;

RO – бит доступен только на чтение;

U – бит физически не реализован или зарезервирован.

Таблица 16 – Описание бит регистра FLASH\_CMD

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...15	-	Зарезервировано
14	TMR	Сброс тестового режима FLASH: 0 – разрешение теста 1 – сброс Рекомендовано всегда записывать единицу
13	NVSTR	Операции записи или стирания: 0 – при чтении; 1 – при записи или стирании
12	PROG	Записать данные из регистра FLASH_DI: 0 – нет записи; 1 – есть запись

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
11	MAS1	Стереть весь блок, при ERASE =1: 0 – нет стирания; 1 – стирание
10	ERASE	Стереть страницу памяти размером 4 Кбайт: 0 – нет стирания; 1 – стирание
9	IFREN	Работа с блоком информации: 0 – основная память; 1 – информационная память
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[17:9]: 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
2, 1	-	Зарезервировано
0	CON	Переключение контроллера FLASH-памяти на регистровое управление, не может производиться при исполнении программы из области FLASH: 0 – управление FLASH от ядра, рабочий режим; 1 – управление от регистров, режим программирования

### 10.3.2 FLASH\_ADR

Таблица 17 – Регистр FLASH\_ADR

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

Таблица 18 – Описание бит регистра FLASH\_ADR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	ADR[31:0]	<p>Адрес обращения в память.</p> <p>При IFREN = 0 (основная память):</p> <ul style="list-style-type: none"> <li>– ADR[17:12] – адрес страницы (XADR[8:3]);</li> <li>– ADR[11:9] – адрес строки (XADR[2:0]);</li> <li>– ADR[8:2] – адрес слова (YADR[6:0]);</li> <li>– ADR[1:0] – не имеет значения. Минимально адресуемая ячейка 32 бита.</li> </ul> <p>При IFREN = 1 (информационная память):</p> <ul style="list-style-type: none"> <li>– ADR[12] – адрес страницы (XADR[3]);</li> <li>– ADR[11:9] – адрес строки (XADR[2:0]);</li> <li>– ADR[8:2] – адрес слова (YADR[6:0]);</li> <li>– ADR[1:0] – не имеет значения. Минимально адресуемая ячейка 32 бита</li> </ul>

### 10.3.3 FLASH\_DI

Таблица 19 – Регистр FLASH\_DI

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

Таблица 20 – Описание бит регистра FLASH\_DI

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	DATA[31:0]	Данные для записи в FLASH

### 10.3.4 FLASH\_DO

Таблица 21 – Регистр FLASH\_DO

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

Таблица 22 – Описание бит регистра FLASH\_DO

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	DATA[31:0]	Данные, считанные из FLASH

### 10.3.5 FLASH\_KEY

Таблица 23 – Регистр FLASH\_KEY

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

Таблица 24 – Описание бит регистра FLASH\_KEY

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	KEY[31:0]	<p>Ключ для разрешения доступа к FLASH-памяти через регистровый доступ.</p> <p>Перед переводом памяти в режим программирования необходимо в регистр FLASH_KEY записать комбинацию 0x8AAA5551.</p> <p>После окончания операций записи или стирания FLASH-памяти необходимо сбросить ключ для запрещения регистрового доступа</p>

## 11 Система команд

В процессоре реализована RISC-V система команд RV32IMC.

Поддерживаемые команды представлены в таблице 25.

В таблице используются следующие обозначения:

- rs – регистр источник;
- rd – регистр приемник;
- imm или shamt – непосредственное значение;

Таблица 25 – Система команд процессора BM-310S

Мнемокод команды	Формат	Операнды	Краткое описание
LB	I	rd, rs1, imm	Загрузка байта
LH	I	rd, rs1, imm	Загрузка полуслова
LW	I, Cx	rd, rs1, imm	Загрузка слова
LBU	I	rd, rs1, imm	Загрузка байта без знака
LHU	I	rd, rs1, imm	Загрузка слова без знака
SB	S	rs1, rs2, imm	Сохранение байта
SH	S	rs1, rs2, imm	Сохранение полуслова
SW	S, Cx	rs1, rs2, imm	Сохранение слова
ADD	R, Cx	rd, rs1, rs2	Арифметическое сложение
ADDI	I, Cx	rd, rs1, imm	Арифметическое сложение с непосредственным значением
SUB	R, Cx	rd, rs1, rs2	Арифметическое вычитание
LUI	U	rd, imm	Загрузка верхней части непосредственным значением
AUIPC	U	rd, imm	Арифметическое сложение верхней части PC с непосредственным значением
XOR	R	rd, rs1, rs2	Логическое исключающее «ИЛИ»
XORI	I	rd, rs1, imm	Логическое исключающее «ИЛИ» с непосредственным значением
OR	R, Cx	rd, rs1, rs2	Логическое «ИЛИ»
ORI	I	rd, rs1, imm	Логическое «ИЛИ» с непосредственным значением
AND	R, Cx	rd, rs1, rs2	Логическое «И»
ANDI	I	rd, rs1, imm	Логическое «И» с непосредственным значением
SLL	R	rd, rs1, rs2	Логический сдвиг влево
SLLI	I, Cx	rd, rs1, shamt	Логический сдвиг влево на непосредственное значение
SRL	R	rd, rs1, rs2	Логический сдвиг вправо
SRLI	I	rd, rs1, shamt	Логический сдвиг вправо на непосредственное значение
SRA	R	rd, rs1, rs2	Арифметический сдвиг вправо
SRAI	I	rd, rs1, shamt	Арифметический сдвиг вправо на непосредственное значение



Мнемокод команды	Формат	Операнды	Краткое описание
SLT	R	rd, rs1, rs2	Сравнение «<<»
SLTI	I	rd, rs1, imm	Сравнение «<<» с непосредственным значением
SLTU	R	rd, rs1, rs2	Сравнение «<<» с беззнаковым
SLTIU	I	rd, rs1, imm	Сравнение «<<» с беззнаковым с непосредственным значением
BEQ	SB, Cx	rs1, rs2, imm	Переход в случае равенства
BNE	SB, Cx	rs1, rs2, imm	Переход в случае неравенства
BLT	SB	rs1, rs2, imm	Переход в случае меньше
BGE	SB	rs1, rs2, imm	Переход в случае больше или равно
BLTU	SB	rs1, rs2, imm	Переход в случае меньше беззнакового
BGEU	SB	rs1, rs2, imm	Переход в случае больше или равно беззнакового
JAL	UJ, Cx	rd, imm	Переход & link
JALR	UJ, Cx	rd, rs1, imm	Переход & link register
FENCE.I	I	-	Синхронизация потока подкачки команд и потока чтения/записи данных
RDINSTRET	I	rd	Псевдоинструкция (alias) на команду чтения счетчика числа исполненных инструкций
ECALL	I	-	Запрос на выполнение переменного окружения операционной системы
EBREAK	I	-	Передача управления в переменное окружение отладки
RDCYCLE	I	rd	Псевдоинструкция (alias) на команду чтения счетчика количества выполненных циклов процессора
RDCYCLEN	I	rd	Псевдоинструкция (alias) на команду чтения старших бит счетчика количества выполненных циклов процессора
RDTIME	I	rd	Псевдоинструкция (alias) на команду чтения значения низкочастотного системного таймера ядра
RDTIMEN	I	rd	Псевдоинструкция (alias) на команду чтения значения старших бит низкочастотного системного таймера ядра
RDINSTRET	I	rd	Псевдоинструкция (alias) на команду чтения счетчика числа исполненных инструкций
RDINSTRETH	I	rd	Псевдоинструкция (alias) на команду чтения старших бит счетчика числа исполненных инструкций
MUL	R	rd, rs1, rs2	Умножение
MULH	R	rd, rs1, rs2	Умножение с возвратом старших бит результата

Мнемокод команды	Формат	Операнды	Краткое описание
MULHSU	R	rd, rs1, rs2	Умножение знакового на беззнаковое с возвратом старших бит результата
MULHU	R	rd, rs1, rs2	Умножение беззнакового на беззнаковое с возвратом старших бит результата
DIV	R	rd, rs1, rs2	Деление
DIVU	R	rd, rs1, rs2	Беззнаковое деление
REM	R	rd, rs1, rs2	Остаток от деления
REMU	R	rd, rs1, rs2	Беззнаковый остаток от деления

Подробное описание каждой команды и форматы команд приведены в документе «The RISC-V Instruction Set Manual Volume I: User-Level ISA Document Version 2.2».

## 12 Процессорное ядро BM-310S

BM-310S поддерживает два режима привилегированности: machine и user. Режим user предоставляет механизм изоляции процессов друг от друга и от доверенного кода, исполняемого в режиме machine. Более подробно с режимами привилегированности можно ознакомиться в документе «The RISC-V Instruction Set Manual Volume II: Privileged Architecture».

Конвейер BM-310S состоит из трех стадий, на которых выполняются следующие операции:

- 1 генерация запроса в подсистему памяти программ (PMS);
- 2 чтение фрагмента кода из PMS и декодирование команды;
- 3 исполнение команды.

Подсистема предварительной обработки команд (FE) организывает выполнение операций первых двух стадий конвейера.

Блок FE состоит из следующих блоков:

– IFU – Instruction Fetch Unit – содержит логику формирования адреса следующего фрагмента кода.

– IDU – Instruction Decode Unit – блок декодирования инструкций.

– RAS – Return Address Stack – стек адресов возврата вызовов функций.

*На первой стадии* блок IFU формирует запросы для считывания фрагментов кода в подсистему памяти программ (PMS), используя информацию о перенаправлении подкачки от IDU, RAS, а также подсистемы исполнения команд (BE).

*На второй стадии* происходит чтение фрагмента кода из памяти программ. Фрагмент кода может содержать одну или две команды. После этого осуществляется декодирование одной команды и результат помещается в очередь декодированных команд (состоит из двух элементов) или в ее байпас к подсистеме исполнения команд (BE).

*На третьей стадии* подсистема исполнения команд (BE) получает декодированную команду и ее операнды из регистрового файла, осуществляет проверку возможности ее выполнения и выполняет на соответствующем исполнительном устройстве.

Выполнение всех команд RV32IMC занимает 1 такт, кроме команд умножения/деления (см. таблицу 26).

Таблица 26 – Длительность выполнения команд

Команда	Длительность, такты
MUL	2
MULH	2
MULHS	2
MULHSU	2
DIV	от 2 до 16
DIVU	от 2 до 16
REM	от 2 до 16
REMU	от 2 до 16

Исполнительное устройство умножения является конвейеризованным и отвечает за исполнение команд MUL/MULH[[S]U]. Исполнительное устройство деления является итеративным и отвечает за исполнение команд DIV[U] и REM[U]. Команды доступа к CSR регистрам исполняются в пустом конвейере.

Подсистемы памяти программ (PMS) и данных (DMS) обеспечивают доступ к интегрированной памяти (TCM) и I/O диапазону адресов, часть из которых зарезервирована для внутренних блоков процессорного комплекса, а часть может быть использована для обращения к устройствам на АНВ шине. Доступ к TCM осуществляется с помощью выделенных интерфейсов. Ядро использует последовательную модель доступа (strong ordering memory access) по АНВ шине, т.е. следующий запрос на АНВ шине не будет выставлен, до окончания исполнения текущего. Если по окончании обработки запроса на АНВ шине возникла ошибка, то в ядре будет сгенерировано исключение (access fault).

### 12.1 Структурная схема процессора

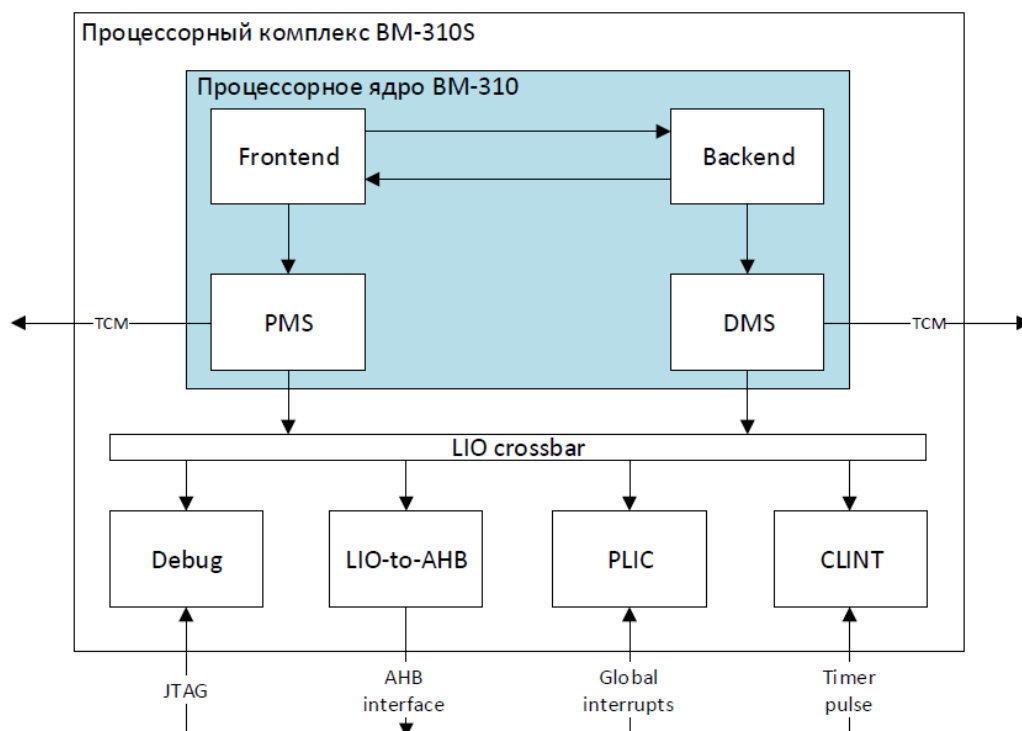


Рисунок 16 – Структурная схема процессора

### 12.1.1 Интерфейс TCM

Интерфейс TCM используется для доступа к интегрированной памяти программ/данных. Интерфейс TCM разделен на две части: интерфейс PMS и интерфейс DMS. Ширина шин адреса интерфейса TCM 14 бит, адресация пословная, поддерживается память размером до 128КБ. Ширина шин данных равна 32 бита.

### 12.1.2 Интерфейс АНВ I/O

Процессорный комплекс использует интерфейс АНВ I/O для доступа к регистрам периферийных устройств или неэкспонируемой памяти. Интерфейс АНВ I/O использует 32-битный физический адрес. Ширина шин данных интерфейса АНВ I/O равна 32 бита. Существует возможность исполнения кода из диапазона I/O адресов.

### 12.1.3 Запросы прерываний

Запросы внешних прерываний Global interrupts подключены к контролеру внешних прерываний PLIC. Сигналы, подключаемые к Global interrupts, должны быть синхронны с тактовым сигналом ядра. Источники внешних прерываний могут использовать как запросы прерываний, формируемые по уровню (level-sensitive), так и импульсные запросы прерываний (edge-triggered).

### 12.1.4 Интерфейс JTAG

Имплементация должна устанавливать в корректное значение сигнал manufid – идентификатор производителя микросхемы (JEDEC ID). Интерфейс JTAG подключается непосредственно к подсистеме отладки (Debug) процессорного комплекса. Блок Debug не сбрасывается от сигнала ndmreset. Сброс блока Debug осуществляется сигналом сброса POR, внешним сигналом сброса nRESET, внутренними сигналами сброса от сторожевых таймеров. Для корректной работы блока Debug он должен быть сброшен при начале работы системы.

### 12.1.5 LIO crossbar

Локальная шина ядра.

### 13 Блок АЦП для измерения напряжений и токов в электрической сети

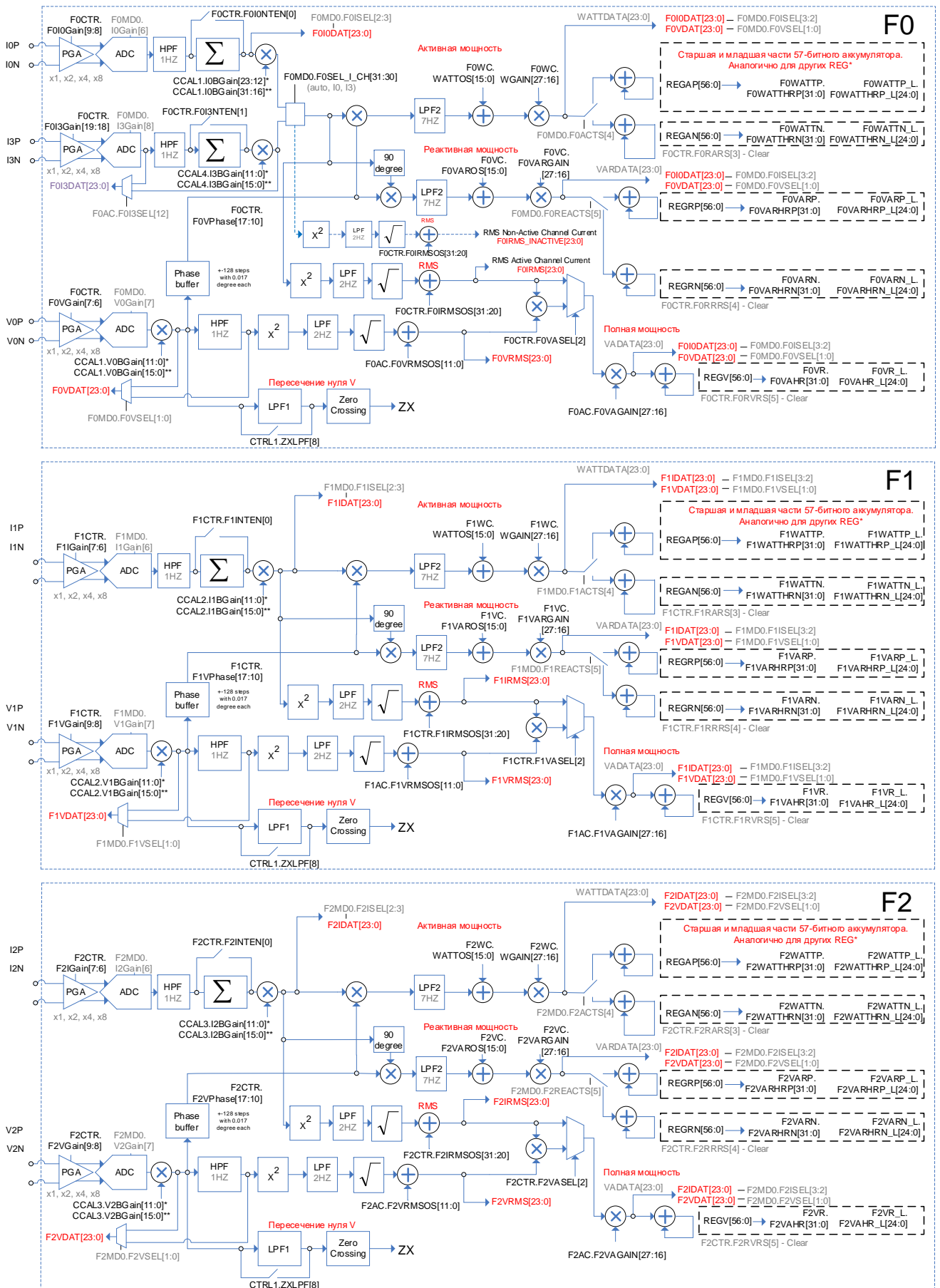
Микросхема имеет в своем составе блок из семи каналов 24-битных  $\Delta\Sigma$  АЦП. Все каналы разбиты на три пары F0-F2 (канал напряжения и канал тока) для трехфазной сети и еще одного независимого канала тока. Каждый из семи каналов оцифровывает входной сигнал с выходной частотой отсчетов до 16 кГц. Кроме этого, в каждой паре каналов F0-F2 реализована возможность рассчитывать среднеквадратические значения тока/напряжения, вычислять активную и реактивную мощности, вычислять потребленную активную и реактивную энергию, частоту сигнала в каналах напряжения, превышение пикового значения, падение сигнала ниже установленного уровня. Эти дополнительные блоки позволяют снизить нагрузку на процессор, что в свою очередь снижает потребляемую мощность всего кристалла. Также каждый АЦП имеет независимый канал DMA, обеспечивая возможность сохранения данных в ОЗУ без участия процессора. Необходимо учитывать, что запросы DMA от АЦП не маскируются, поэтому перед включением каналов АЦП требуется настройка соответствующих каналов DMA.

Структурная схема семи каналов АЦП приведена на рисунке 17.

Список параметров, вычисляемых блоком АЦП:

- Семь независимых АЦП с выходной частотой отсчетов 4/8/16 кГц (четыре канала тока и три канала напряжения). Эти каналы образуют три блока для измерения параметров каждой фазы F0-F2. Размер FIFO данных по каждому каналу АЦП составляет восемь отсчетов;
- В блоке каналов F0 реализуем автоматический выбор канала тока (который имеет максимальное значение) для последующих расчетов мощностных характеристик. Если разница токов превышает 6 %, то формируется прерывание. Кроме этой функции в остальном блоки F0-F2 идентичны;
- Все каналы АЦП имеют независимые калибровочные коэффициенты наклона характеристики;
- Каждый канал тока имеет независимый интегратор. Интегратор необходимо применять только при отставании напряжения от тока в блоке Fx по фазе на 90 градусов (в силу примененных схемотехнических решений): при включении выполняется фазовая коррекция. Если ток и напряжение в блоке Fx синфазны, то применять интегратор не требуется;
- В каждом блоке АЦП (F0-F2) независимо рассчитывается период сигнала по каналу напряжения. Количество периодов, в течение которого рассчитывается эта величина, можно задавать равным 1/2/4/8/16/32/64/128 периодам;
- В каждом блоке есть проверка на пропажу периодического сигнала в канале напряжения;
- В каждом блоке проверяется просадка напряжения ниже заданного уровня, а также превышения сигнала в каналах тока и напряжения установленного лимита;

- Есть возможность скорректировать фазы сигналов в каналах напряжения с точностью до 0,02 %;
- Вычисляются среднеквадратические, квадрат среднеквадратических значений токов и напряжений, а также их независимая калибровка;
- При вычислении активной и реактивной энергиях значение накопленной энергии в течение периода накапливаются в отдельных регистрах (для положительной и отрицательной энергии);
- Вычисляются полная мощность и полная энергия;
- Вычисляется сдвиг фаз по отношению к фазе 0;
- Для измерения постоянной составляющей сигнала (или сигнала до фильтрации), доступны четыре канала из семи: все три канала напряжений, а также канал I3. При этом для каналов напряжений необходимо настроить регистр ADCUI\_FxMD0, поле FxVSEL, в значение «00» для отсчетов напряжений (установлено по умолчанию), когда как для канала I3 необходимо настроить регистр ADCUI\_F0AC, поле F0I3SEL, в значение «1».



- \* Для микросхем ревизии 1;
- \*\* Для микросхем с ревизии 2.

Рисунок 17 – Структурная схема семи каналов АЦП



Для предотвращения влияния высокочастотных помех на результаты вычисления необходимо поставить внешний anti-aliasing фильтр. Можно использовать простейший RC-фильтр первого порядка с частотой среза 100 кГц.

Все цифровые фильтры настроены на указанные частоты среза при тактировании блока ADCUI от HSE = 8,192 МГц, при этом входная частота АЦП равна 1,024 МГц (выходная частота отсчетов 4/8/16 кГц).

### 13.1 Описание регистров управления блока семи каналов АЦП

Таблица 27 – Регистры управления блока семи каналов АЦП

Базовый Адрес	Название	Описание
0x4006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x000	ADCUI_CTRL1	Общее управление для контроллера АЦП
0x004	ADCUI_CTRL2	
0x008	ADCUI_CTRL3	
0x00C	ADCUI_F0CTR	Управление блоком каналов F0
0x010	ADCUI_F0WC	Управление расчетом активной мощности в блоке F0
0x014	ADCUI_F0WATTP	Старшая часть значения положительной активной мощности в блоке F0
0x018	ADCUI_F0WATTN	Старшая часть значения отрицательной активной мощности в блоке F0
0x01C	ADCUI_F0VC	Управление расчетом реактивной мощности в блоке F0
0x020	ADCUI_F0VARP	Старшая часть значения положительной реактивной мощности в блоке F0
0x024	ADCUI_F0VARN	Старшая часть значения отрицательной реактивной мощности в блоке F0
0x028	ADCUI_F0AC	Управление расчетом полной мощности в блоке F0
0x02C	ADCUI_F0VR	Старшая часть значения полной мощности в блоке F0
0x030	ADCUI_F0MD0	Группа параметров 0 блока F0
0x034	ADCUI_F0MD1	Группа параметров 1 блока F0
0x038	ADCUI_F0VPEAK	Пиковое значение в канале напряжения блока F0
0x03C	ADCUI_F0IPEAK	Пиковое значение в канале тока блока F0
0x040	ADCUI_F0VDAT	Буфер FIFO отсчетов в канале напряжения блока F0
0x044	ADCUI_F0I0DAT	Буфер FIFO отсчетов в канале тока (I0) блока F0
0x048	ADCUI_F0I3DAT	Буфер FIFO отсчетов в канале тока (I3) блока F0
0x04C	ADCUI_F0VRMS	Среднеквадратическое значение канала напряжения в блоке F0
0x050	ADCUI_F0VRMS2	Квадрат RMS в канале напряжения блока F0
0x054	ADCUI_F0IRMS	Среднеквадратическое значение канала тока в блоке F0
0x058	ADCUI_F0IRMS2	Квадрат RMS в канале тока блока F0
0x05C	ADCUI_F0STAT	Статус каналов блока F0
0x060	ADCUI_F0MASK	Маска прерываний каналов в блоке F0
0x064	ADCUI_F1CTR	Управление блоком каналов F1
0x068	ADCUI_F1WC	Управление расчетом активной мощности в блоке F1

Базовый Адрес	Название	Описание
0x4006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x06C	ADCUI_F1WATTP	Старшая часть значения положительной активной мощности в блоке F1
0x070	ADCUI_F1WATTN	Старшая часть значения отрицательной активной мощности в блоке F1
0x074	ADCUI_F1VC	Управление расчетом реактивной мощности в блоке F1
0x078	ADCUI_F1VARP	Старшая часть значения положительной реактивной мощности в блоке F1
0x07C	ADCUI_F1VARN	Старшая часть значения отрицательной реактивной мощности в блоке F1
0x080	ADCUI_F1AC	Управление расчетом полной мощности в блоке F1
0x084	ADCUI_F1VR	Старшая часть значения полной мощности в блоке F1
0x088	ADCUI_F1MD0	Группа параметров 0 блока F1
0x08C	ADCUI_F1MD1	Группа параметров 1 блока F1
0x090	ADCUI_F1MD2	Группа параметров 2 блока F1
0x094	ADCUI_F1VPEAK	Пиковое значение в канале напряжения блока F1
0x098	ADCUI_F1IPEAK	Пиковое значение в канале тока блока F1
0x09C	ADCUI_F1VDAT	Буфер FIFO отсчетов в канале напряжения блока F1
0x0A0	ADCUI_F1IDAT	Буфер FIFO отсчетов в канале тока (I) блока F1
0x0A4	ADCUI_F1VRMS	Среднеквадратическое значение канала напряжения в блоке F1
0x0A8	ADCUI_F1VRMS2	Квадрат RMS в канале напряжения блока F1
0x0AC	ADCUI_F1IRMS	Среднеквадратическое значение тока канала в блоке F1
0x0B0	ADCUI_F1IRMS2	Квадрат RMS в канале тока блока F1
0x0B4	ADCUI_F1STAT	Статус каналов блока F1
0x0B8	ADCUI_F1MASK	Маска прерываний каналов в блоке F1
0x0BC	ADCUI_F2CTR	Управление блоком каналов F2
0x0C0	ADCUI_F2WC	Управление расчетом активной мощности в блоке F2
0x0C4	ADCUI_F2WATTP	Старшая часть значения положительной активной мощности в блоке F2
0x0C8	ADCUI_F2WATTN	Старшая часть значения отрицательной активной мощности в блоке F2
0x0CC	ADCUI_F2VC	Управление расчетом реактивной мощности в блоке F2
0x0D0	ADCUI_F2VARP	Старшая часть значения положительной реактивной мощности в блоке F2
0x0D4	ADCUI_F2VARN	Старшая часть значения отрицательной реактивной мощности в блоке F2
0x0D8	ADCUI_F2AC	Управление расчетом полной мощности в блоке F2
0x0DC	ADCUI_F2VR	Старшая часть значения полной мощности в блоке F2
0x0E0	ADCUI_F2MD0	Группа параметров 0 блока F2
0x0E4	ADCUI_F2MD1	Группа параметров 1 блока F2
0x0E8	ADCUI_F2MD2	Группа параметров 2 блока F2
0x0EC	ADCUI_F2VPEAK	Пиковое значение в канале напряжения блока F2

Базовый Адрес	Название	Описание
0x4006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x0F0	ADCUI_F2IPEAK	Пиковое значение в канале тока блока F2
0x0F4	ADCUI_F2VDAT	Буфер FIFO отсчетов в канале напряжения блока F2
0x0F8	ADCUI_F2IDAT	Буфер FIFO отчетов в канале тока (I) блока F2
0x0FC	ADCUI_F2VRMS	Среднеквадратическое значение канала напряжения в блоке F2
0x100	ADCUI_F2VRMS2	Квадрат RMS в канале напряжения блока F2
0x104	ADCUI_F2IRMS	Среднеквадратическое значение канала тока в блоке F2
0x108	ADCUI_F2IRMS2	Квадрат RMS в канале тока блока F2
0x10C	ADCUI_F2STAT	Статус каналов блока F2
0x110	ADCUI_F2MASK	Маска прерываний каналов в блоке F2
0x114	ADCUI_CCAL1	Регистр калибровки каналов тока (I0) и напряжения (V0) в блоке F0
0x118	ADCUI_CCAL2	Регистр калибровки каналов тока и напряжения в блоке F1
0x11C	ADCUI_CCAL3	Регистр калибровки каналов тока и напряжения в блоке F2
0x120	ADCUI_CCAL4	Регистр калибровки канала тока (I3) в блоке F0
0x124	ADCUI_F0IRMS_INACTIVE	Среднеквадратическое значение тока неактивного канала в блоке F0
0x128	ADCUI_F0WATTP_L	Младшая часть значения положительной активной мощности в блоке F0
0x12C	ADCUI_F0WATTN_L	Младшая часть значения отрицательной активной мощности в блоке F0
0x130	ADCUI_F0VARP_L	Младшая часть значения положительной реактивной мощности в блоке F0
0x134	ADCUI_F0VARN_L	Младшая часть значения отрицательной реактивной мощности в блоке F0
0x138	ADCUI_F0VR_L	Младшая часть значения полной мощности в блоке F0
0x13C	ADCUI_F1WATTP_L	Младшая часть значения положительной активной мощности в блоке F1
0x140	ADCUI_F1WATTN_L	Младшая часть значения отрицательной активной мощности в блоке F1
0x144	ADCUI_F1VARP_L	Младшая часть значения положительной реактивной мощности в блоке F1
0x148	ADCUI_F1VARN_L	Младшая часть значения отрицательной реактивной мощности в блоке F1
0x14C	ADCUI_F1VR_L	Младшая часть значения полной мощности в блоке F1
0x150	ADCUI_F2WATTP_L	Младшая часть значения положительной активной мощности в блоке F2
0x154	ADCUI_F2WATTN_L	Младшая часть значения отрицательной активной мощности в блоке F2
0x158	ADCUI_F2VARP_L	Младшая часть значения положительной реактивной мощности в блоке F2

Базовый Адрес	Название	Описание
0x4006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x15C	ADCUI_F2VARN_L	Младшая часть значения отрицательной реактивной мощности в блоке F2
0x160	ADCUI_F2VR_L	Младшая часть значения полной мощности в блоке F2
0x164	ADCUI_LOAD	Общее управление режимом «без нагрузки» (доступно с ревизии 2)
0x168	ADCUI_F0VRMS_TRUE	Действующее значение канала напряжения на периоде в блоке F0 (доступно с ревизии 2)
0x16C	ADCUI_F1VRMS_TRUE	Действующее значение канала напряжения на периоде в блоке F1 (доступно с ревизии 2)
0x170	ADCUI_F2VRMS_TRUE	Действующее значение канала напряжения на периоде в блоке F2 (доступно с ревизии 2)

### 13.1.1 ADCUI\_CTRL1

Таблица 28 – Регистр ADCUI\_CTRL1

Номер	31, 30	29	28	27	26, 25	24, 23
Доступ	R/W	R/W	R/W	R/W	-	R/W
Сброс	00	0	0	0	-	00
	OSR_CONF	tst_in[4]	RESET_DIG	ZXRMS	-	tst_in[3:2]

Номер	22	21	20	19	18, 17	16, 15
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	tst_in[1]	tst_in[0]	VREF_SEL	FREQSEL	VANOLOAD	VARNLOAD

Номер	14	13, 12	11...9	8	7	6
Доступ	-	R/W	R/W	R/W	R/W	R/W
Сброс	-	0	000	0	0	0
	-	APNOLOAD	PER_LENGTH	ZXLPF	RESOL	I3EN

Номер	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	V2EN	I2EN	V1EN	I1EN	V0EN	I0EN

Таблица 29 – Описание бит регистра ADCUI\_CTRL1

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	OSR_CONF*	Выбор коэффициента передискретизации: 00 – 256 (4 кГц); 01 – 128 (8 кГц); 10 – 64 (16 кГц); 11 – Зарезервировано
29	tst_in[4]	Тестовый выход (всегда записывать ноль)
28	RESET_DIG	Сброс цифровой части блоков АЦП: 0 – нет сброса; 1 – цифровая часть под общим сбросом
27	ZXRMS**	Управления обновления регистров со среднеквадратическими значениями: 0 – непрерывное обновление; 1 – обновление при пересечении напряжением «0»
26, 25	-	Зарезервировано
24, 23	tst_in[3:2]	Тестовые выходы (всегда записывать ноль)
22	tst_in[1]	Тестовый выход (всегда записывать ноль)
21	tst_in[0]	Тестовый выход (всегда записывать ноль)
20	VREF_SEL	Выбор опорного напряжения для АЦП: 0 – внешнее опорное напряжение с вывода VREF; 1 – внутреннее опорное напряжение VREF_INT 2,4V (рисунок 60)
19	FREQSEL	Разрешение вычисления длительности периода в каналах напряжения: 1 – разрешено; 0 – хранится последнее вычисленное значение
18, 17	VANOLOAD	Режим «без нагрузки» при вычислении полной энергии: 00 – вся вычисленная энергия накапливается; 01 – не учитывается энергия ниже 0,012 % от полной шкалы; 10 – не учитывается энергия ниже 0,0061 % от полной шкалы; 11 – не учитывается энергия ниже 0,00305 % от полной шкалы
16, 15	VARNLOAD	Режим «без нагрузки» при вычислении реактивной энергии: 00 – вся вычисленная энергия накапливается; 01 – не учитывается энергия ниже 0,012 % от полной шкалы; 10 – не учитывается энергия ниже 0,0061 % от полной шкалы; 11 – не учитывается энергия ниже 0,00305 % от полной шкалы
14	-	Зарезервировано
13, 12	APNOLOAD	Режим «без нагрузки» при вычислении активной энергии: 00 – вся вычисленная энергия накапливается; 01 – не учитывается энергия ниже 0,012 % от полной шкалы; 10 – не учитывается энергия ниже 0,0061 % от полной шкалы; 11 – не учитывается энергия ниже 0,00305 % от полной шкалы
11...9	PER_LENGTH	Диапазон вычисления периода и фазового сдвига: 000 – в течение 1 периода; 001 – в течение 2 периодов; ... 111 – в течение 128 периодов

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8	ZXLPF	Отключение низкочастотного фильтра перед детектором пересечения «0» в каналах напряжения: 0 – фильтр включен; 1 – фильтр отключен
7	RESOL	Разрешение выходных данных: 0 – 16 бит; 1 – 24 бита. Старший бит в регистре данных 15 или 23. Сдвиг регистра данных выполнять не требуется
6	I3EN	Разрешение работы канала I3: 0 – канал отключен; 1 – канал включен***
5	V2EN	Разрешение работы канала V2: 0 – канал отключен; 1 – канал включен***
4	I2EN	Разрешение работы канала I2: 0 – канал отключен; 1 – канал включен***
3	V1EN	Разрешение работы канала V1: 0 – канал отключен; 1 – канал включен***
2	I1EN	Разрешение работы канала I1: 0 – канал отключен; 1 – канал включен***
1	V0EN	Разрешение работы канала V0: 0 – канал отключен; 1 – канал включен***
0	I0EN	Разрешение работы канала I0: 0 – канал отключен; 1 – канал включен***

\* При увеличении частоты дискретизации все внутренние цифровые фильтры соответствующим образом корректируются, что сохраняет их частоты среза постоянными. Также необходимо учитывать, что увеличение частоты дискретизации в два раза ведет к уменьшению SNR как минимум на 3 дБ в полосе от 0 Гц до половины частоты дискретизации (это следует из того, что шум интегрируется в частоте два раза большей).

\*\* Так как происходит одновременное обновление среднеквадратических значений и тока и напряжения, то значение тока будет зависеть от угла между напряжением и током. На графике видна эта зависимость. Исходя из этих данных, можно скорректировать действительное значение тока.

\*\*\* При включении любого канала в блоке Fx, включаются FIFO всех остальных каналов (FIFO FxVDAT FxIDAT) в блоке Fx

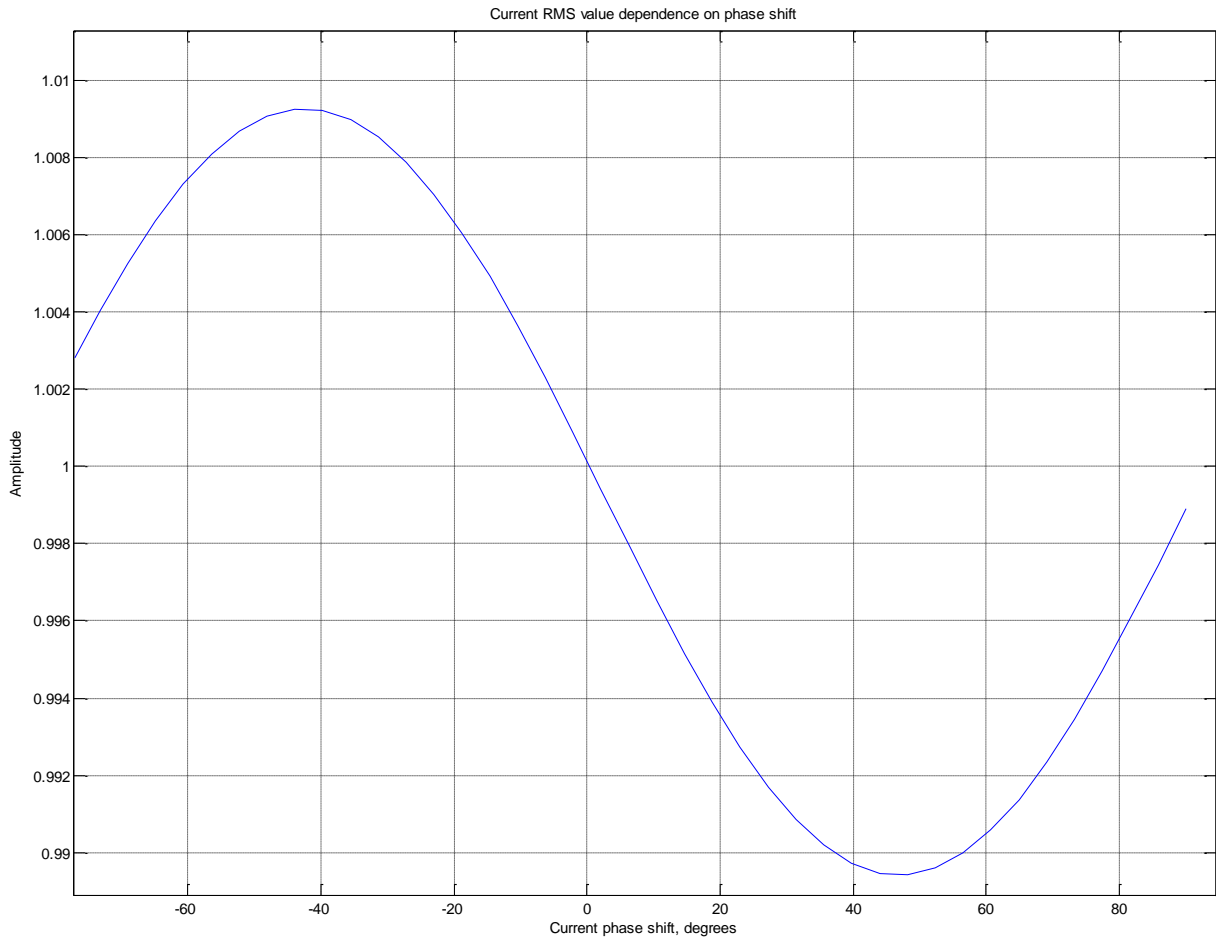


Рисунок 18 – Зависимость RMS от сдвига фазы входного сигнала.

### 13.1.2 ADCUI\_CTRL2

Таблица 30 – Регистр ADCUI\_CTRL2

Номер	31...24	23...16	15...0
Доступ	-	R/W	R/W
Сброс	-	FFh	FFFFh
	-	SAGCYC	SAGLVL

Таблица 31 – Описание бит регистра ADCUI\_CTRL2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...16	SAGCYC	Количество полутактов напряжения для вычисления просадки уровня напряжения
15...0	SAGLVL	Уровень разрешенной просадки напряжения

### 13.1.3 ADCUI\_CTRL3

Таблица 32 – Регистр ADCUI\_CTRL3

Номер	31...12	11...0
Доступ	-	R/W
Сброс	-	0FFh
	-	ZXTOUT

Таблица 33 – Описание бит регистра ADCUI\_CTRL3

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..12	-	Зарезервировано
11..0	ZXTOUT	Значение time-out регистра, который устанавливает это значение при пересечении сигнала напряжения «0»

### 13.1.4 ADCUI\_F0CTR

Таблица 34 – Регистр ADCUI\_F0CTR

Номер	31...20	19, 18	17...10	9, 8	7, 6
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	00	00	0	0
	F0IRMSOS	F0I3GAIN	F0VPHASE	F0I0GAIN	F0VGIN

Номер	5	4	3	2	1	0
Доступ	WO	WO	WO	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	F0RVRS	F0RRRS	F0RARS	F0VASEL	F0I3NTEN	F0I0NTEN

Таблица 35 – Описание бит регистра ADCUI\_F0CTR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F0IRMSOS	Калибровка вычислителя среднеквадратического значения тока. Представлено в виде знаковых величин в двоичном коде с дополнением до 2, при применении в вычислителе сдвигается на 4 бита влево. Пример: 007: IRMSOS = $7 \cdot 16 = 112$ ; FF9: IRMSOS = $-7 \cdot 16 = -112$
19, 18	F0I3GAIN	Предусилитель в канале тока 3: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
17...10	F0VPHASE	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -126d (-123 мкс) до +127d (+124 мкс). «0» соответствует синфазному сигналу с током



Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
9, 8	F0I0GAIN	Предусилитель в канале тока 0: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
7, 6	F0VGAIN	Предусилитель в канале напряжения: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
5	F0RVRS	Запись единицы в этот бит сбрасывает счетчик переданной полной энергии
4	F0RRRS	Запись в этот бит сбрасывает счетчик переданной реактивной энергии
3	F0RARS	Запись в этот бит сбрасывает счетчик переданной активной энергии
2	F0VASEL	Выбор источника сигнала для сохранения в регистре полной энергии: 0 – полная энергия; 1 – среднеквадратическое значение тока
1	F0I3NTEN	Отключение интегратора в канале тока 3: 0 – интегратор включен; 1 – интегратор отключен
0	F0I0NTEN	Отключение интегратора в канале тока 0: 0 – интегратор включен; 1 – интегратор отключен

### 13.1.5 ADCUI\_F0WC

Таблица 36 – Регистр ADCUI\_F0WC

Номер	31...28	27...16	15...0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	F0WGAIN	F0WATTOS

Таблица 37 – Описание бит регистра ADCUI\_F0WC

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0WGAIN	Калибровка усиления сигнала в ветви активной мощности для блока F0
15...0	F0WATTOS	Калибровка смещения сигнала в ветви активной мощности для блока F0

### 13.1.6 ADCUI\_F0WATTP

Таблица 38 – Регистр ADCUI\_F0WATTP

Номер	31...0
Доступ	RO
Сброс	
	F0WATTHRP

Таблица 39 – Описание бит регистра ADCUI\_F0WATTP

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0WATTHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной активной энергии

### 13.1.7 ADCUI\_F0WATTN

Таблица 40 – Регистр ADCUI\_F0WATTN

Номер	31...0
Доступ	RO
Сброс	
	F0WATTHRN

Таблица 41 – Описание бит регистра ADCUI\_F0WATTN

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0WATTHRN	Старшие 32 бита внутреннего 57 битного аккумулятора отрицательной активной энергии

### 13.1.8 ADCUI\_F0VC

Таблица 42 – Регистр ADCUI\_F0VC

Номер	31...28	27...16	15...0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	F0VARGAIN	F0VAROS

Таблица 43 – Описание бит регистра ADCUI\_F0VC

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0VARGAIN	Калибровка усиления сигнала в ветви реактивной мощности для блока F0
15...0	F0VAROS	Калибровка смещения сигнала в ветви реактивной мощности для блока F0

### 13.1.9 ADCUI\_F0VARP

Таблица 44 – Регистр ADCUI\_F0VARP

Номер	31..0
Доступ	RO
Сброс	
	F0VARHRP

Таблица 45 – Описание бит регистра ADCUI\_F0VARP

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0VARHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной реактивной энергии

### 13.1.10 ADCUI\_F0VARN

Таблица 46 – Регистр ADCUI\_F0VARN

Номер	31..0
Доступ	RO
Сброс	
	F0VARHRN

Таблица 47 – Описание бит регистра ADCUI\_F0VARN

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0VARHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной реактивной энергии

### 13.1.11 ADCUI\_F0AC

Таблица 48 – Регистр ADCUI\_F0AC

Номер	31...28	27...16	15...13	12	11...0
Доступ	-	R/W	-	R/W	R/W
Сброс	-	000h	-	0	000h
	-	F0VAGAIN	-	F0I3SEL	F0VRMSOS

Таблица 49 – Описание бит регистра ADCUI\_F0AC

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0VAGAIN	Калибровка усиления сигнала в ветви полной мощности для блока F0
15...13	-	Зарезервировано
12	F0I3SEL	Выбор источника сигнала для FIFO канала тока I3 (регистр ADCUI_F0I3DAT): 0 – отсчеты тока после фильтра высоких частот; 1 – отсчеты тока до фильтра высоких частот

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
11...0	F0VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения. Представлено в виде знаковых величин в двоичном коде с дополнением до 2, при применении в вычислителе сдвигается на 4 бита влево. Пример: 007: VRMSOS = $7 \cdot 16 = 112$ ; FF9: VRMSOS = $-7 \cdot 16 = -112$

### 13.1.12 ADCUI\_F0VR

Таблица 50 – Регистр ADCUI\_F0VR

Номер	31...0
Доступ	RO
Сброс	
	F0VAHR

Таблица 51 – Описание бит регистра ADCUI\_F0VR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0VAHR	Старшие 32 бита внутреннего 57-битного аккумулятора полной энергии

### 13.1.13 ADCUI\_F0MD0

Таблица 52 – Регистр ADCUI\_F0MD0

Номер	31, 30	29...9	8	7
Доступ	R/W	RO	R/W	R/W
Сброс	00		0	0
	F0SEL_I_CH	F0PER_FREQ	I3GAIN	V0GAIN

Номер	6	5	4	3, 2	1, 0
Доступ	R/W	R	R	R/W	R/W
Сброс	0	0	0	00	00
	I0GAIN	F0REACTS	F0ACTS	F0ISEL	F0VSEL

Таблица 53 – Описание бит регистра ADCUI\_F0MD0

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	F0SEL_I_CH	Выбор активного канала тока для вычисления мощностных характеристик: 00, 11 – автоматический выбор канала; 01 – активный канал I0; 10 – активный канал I3
29...9	F0PER_FREQ	Длительность такта в канале напряжения

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8	I3GAIN	Усиление в канале I3: 0 – нет усиления; 1 – +6 дБ усиление
7	V0GAIN	Усиление в канале V0: 0 – нет усиления; 1 – +6 дБ усиление
6	I0GAIN	Усиление в канале I0: 0 – нет усиления; 1 – +6 дБ усиление
5	F0REACTS	Знак реактивной энергии в последний период: 0 – минус; 1 – плюс
4	F0ACTS	Знак активной энергии в последний период: 0 – минус; 1 – плюс
3, 2	F0ISEL	Выбор источника сигнала для FIFO канала тока I0 (регистр ADCUI_F0IDAT): 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1, 0	F0VSEL	Выбор источника сигнала для FIFO канала напряжения V0 (регистр ADCUI_F0VDAT): 00 – отсчеты напряжения до фильтра высоких частот; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты напряжения после фильтра высоких частот

### 13.1.14 ADCUI\_F0MD1

Таблица 54 – Регистр ADCUI\_F0MD1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F0VPKLVL	F0IPKLVL

Таблица 55 – Описание бит регистра ADCUI\_F0MD1

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	F0VPKLVL	Предельный разрешенный уровень напряжения
15...0	F0IPKLVL	Предельный разрешенный уровень тока

### 13.1.15 ADCUI\_F0VPEAK

Таблица 56 – Регистр ADCUI\_F0VPEAK

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F0VPEAK

Таблица 57 – Описание бит регистра ADCUI\_F0VPEAK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0VPEAK	Пиковое значение напряжения. Обнуляется чтением из регистра Запись в этот регистр, сбрасывает пиковое значение

### 13.1.16 ADCUI\_F0IPEAK

Таблица 58 – Регистр ADCUI\_F0IPEAK

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F0IPEAK

Таблица 59 – Описание бит регистра ADCUI\_F0IPEAK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0IPEAK	Пиковое значение тока. Обнуляется чтением из регистра Запись в этот регистр, сбрасывает пиковое значение

### 13.1.17 ADCUI\_F0VDAT

Таблица 60 – Регистр ADCUI\_F0VDAT

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	00h	F0VDAT

Таблица 61 – Описание бит регистра ADCUI\_F0VDAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0VDAT	FIFO отсчетов напряжения (или одной из мощностей) FIFO включается при установке любого из бит IZEN, VOEN, IOEN

### 13.1.18 ADCUI\_F0I0DAT

Таблица 62 – Регистр ADCUI\_F0I0DAT

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	00h	F0I0DAT

Таблица 63 – Описание бит регистра ADCUI\_F0I0DAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0I0DAT	FIFO отсчетов тока 0 (или одной из мощностей) FIFO включается при установке любого из бит I3EN, V0EN, I0EN

### 13.1.19 ADCUI\_F0I3DAT

Таблица 64 – Регистр ADCUI\_F0I3DAT

Номер	31:24	23:0
Доступ		RO
Сброс		000000h
	00h	F0I3DAT

Таблица 65 – Описание бит регистра ADCUI\_F0I3DAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0I3DAT	FIFO отсчетов тока 3. FIFO включается при установке любого из бит I3EN, V0EN, I0EN

### 13.1.20 ADCUI\_F0VRMS

Таблица 66 – Регистр ADCUI\_F0VRMS

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	-	F0VRMS

Таблица 67 – Описание бит регистра ADCUI\_F0VRMS

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0VRMS	Среднеквадратическое значение напряжения

### 13.1.21 ADCUI\_F0VRMS2

Таблица 68 – Регистр ADCUI\_F0VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F0VRMS2

Таблица 69 – Описание бит регистра ADCUI\_F0VRMS2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0VRMS2	Квадрат среднеквадратического значения напряжения

### 13.1.22 ADCUI\_F0IRMS

Таблица 70 – Регистр ADCUI\_F0IRMS

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	-	F0IRMS

Таблица 71 – Описание бит регистра ADCUI\_F0IRMS

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0IRMS	Среднеквадратическое значение тока

### 13.1.23 ADCUI\_F0IRMS2

Таблица 72 – Регистр ADCUI\_F0IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F0IRMS2

Таблица 73 – Описание бит регистра ADCUI\_F0IRMS2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F0IRMS2	Квадрат среднеквадратического значения тока



13.1.24 ADCUI\_F0STAT

Таблица 74 – Регистр ADCUI\_F0STAT

Номер	31...27	26	25	24	23	22
Доступ	-	R/W	R/W	R/W	RO	RO
Сброс	-		0	0	0	0
	-	F0VA ROVN	F0WA TTOVN	C3IF_OVR	C3IF_FLL	C3IF_EMP

Номер	21	20	19	18	17	16
Доступ	R/W	RO	-	RO	R/W	RO
Сброс	0	0	-	0	0	0
	F0ZER OCRS	F0VAN LDFL	-	F0VAR NLDFL	F0VAR SIGN	F0APN LDFL

Номер	15	14	13	12	11	10	9	8	7
Доступ	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0	0
	F0AP SIGN	F0FAUL TCON	F0ICH ANNEL	F0ZX TOF	F0VA OV	F0VA ROVP	F0WAT TOVP	F0PE AKIF	F0PE AKVF

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	RO	RO	R/W	RO	RO
Сброс	0	0	0	0	0	0	0
	F0SAGF	F0IF_OVR	F0IF_FLL	F0IF_EMP	F0VF_OVR	F0VF_FLL	F0VF_EMP

Таблица 75 – Описание бит регистра ADCUI\_F0STAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F0VAROVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной реактивной энергией. Запись «1» сбрасывает этот флаг
25	F0WATTOVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной активной энергией. Запись «1» сбрасывает этот флаг
24	C3IF_OVR	Флаг, что произошло переполнения FIFO F0I3DAT. Запись «1» сбрасывает этот флаг
23	C3IF_FLL	Флаг, что FIFO F0I3DAT заполнено
22	C3IF_EMP	Флаг, что FIFO F0I3DAT пусто
21	F0ZEROCRS	Флаг, что произошло пересечение «0» в канале напряжения. Запись «1» сбрасывает этот флаг
20	F0VANLDFL	Полная мощность ниже уровня сравнения
19	-	Зарезервировано
18	F0VARNLDFL	Реактивная мощность ниже уровня сравнения

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
17	F0VARSIGN	Смена знака реактивной мощности. Запись «1» сбрасывает этот флаг
16	F0APNLDFL	Активная мощность ниже уровня сравнения
15	F0APSIGN	Смена знака активной мощности. Запись «1» сбрасывает этот флаг
14	F0FAULTCON	Произошло автоматическое переключение активного канала тока. Запись «1» сбрасывает этот флаг
13	F0ICHANNEL	Активный канал тока: 0 – активный канал I0; 1 – активный канал I3
12	F0ZXTOF	Флаг, что в течении TimeOut не было пересечение напряжением значения «0». Запись «1» сбрасывает этот флаг
11	F0VAOV	Флаг, что произошло переполнение регистра аккумулятора с полной энергией. Запись «1» сбрасывает этот флаг
10	F0VAROVP	Флаг, что произошло переполнение регистра аккумулятора с положительной реактивной энергией. Запись «1» сбрасывает этот флаг
9	F0WATTOVP	Флаг, что произошло переполнение регистра аккумулятора с положительной активной энергией. Запись «1» сбрасывает этот флаг
8	F0PEAKIF	Флаг, что произошло превышение порогового значения тока. Запись «1» сбрасывает этот флаг
7	F0PEAKVF	Флаг, что произошло превышение порогового значения напряжения. Запись «1» сбрасывает этот флаг
6	F0SAGF	Флаг, что произошла просадка напряжения. Запись «1» сбрасывает этот флаг
5	F0IF_OVR	Флаг, что произошло переполнение FIFO F0IDAT. Запись «1» сбрасывает этот флаг
4	F0IF_FLL	Флаг, что FIFO F0IDAT заполнено
3	F0IF_EMP	Флаг, что FIFO F0IDAT пусто
2	F0VF_OVR	Флаг, что произошло переполнение FIFO F0VDAT. Запись «1» сбрасывает этот флаг
1	F0VF_FLL	Флаг, что FIFO F0VDAT заполнено
0	F0VF_EMP	Флаг, что FIFO F0VDAT пусто

13.1.25 ADCUI\_F0MASK

Таблица 76 – Регистр ADCUI\_F0MASK

Номер	31...27	26	25	24	23	22
Доступ	-	R/W	R/W	R/W	R/W	R/W
Сброс	-		0	0	0	0
	-	F0VAROVNM	F0WATTOVNM	C3IF_OVRM	C3IF_FLLM	C3IF_EMPM

Номер	21	20	19	18	17	16
Доступ	R/W	R/W	-	R/W	R/W	R/W
Сброс	0	0	-	0	0	0
	F0ZEROCRSM	F0VANLDFLM	-	F0VARNLDFLM	F0VARSIGNM	F0APNLDFLM

Номер	15	14	13	12	11	10	9	8	7
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
	F0AP SIGNM	F0FAUL TCONM	-	F0ZX TOFM	F0VA OVM	F0VA ROVPM	F0WAT TOVPM	F0PE AKIFM	F0PEA KVFM

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	RO	RO	R/W	RO	RO
Сброс	0	0	0	0	0	0	0
	F0SA GFM	F0IF_ OVRM	F0IF_ FLLM	F0IF_ EMPM	F0VF_ OVRM	F0VF_ FLLM	F0VF_ EMPM

Таблица 77 – Описание бит регистра ADCUI\_F0MASK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F0VAROVNM	Маска бита F0VAROVN
25	F0WATTOVNM	Маска бита F0WATTOVN
24	C3IF_OVRM	Маска бита C3IF_OVR
23	C3IF_FLLM	Маска бита C3IF_FLL
22	C3IF_EMPM	Маска бита C3IF_EMP
21	F0ZEROCRSM	Маска бита F0ZEROCRS
20	F0VANLDFLM	Маска бита F0VANLDFL
19	-	Зарезервировано
18	F0VARNLDFLM	Маска бита F0VARNLDFL
17	F0VARSIGNM	Маска бита F0VARSIGN
16	F0APNLDFLM	Маска бита F0APNLDFL
15	F0APSIGNM	Маска бита F0APSIGNM
14	F0FAULTCONM	Маска бита F0FAULTCON
13	-	Зарезервировано
12	F0ZXTOFM	Маска бита F0ZXTOF
11	F0VAOVM	Маска бита F0VAOV

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
10	F0VAROVPM	Маска бита F0VAROVP
9	F0WATTOVPM	Маска бита F0WATTOVP
8	F0PEAKIFM	Маска бита F0PEAKIF
7	F0PEAKVFM	Маска бита F0PEAKVF
6	F0SAGFM	Маска бита F0SAGF
5	F0IF_OVRM	Маска бита F0IF_OVR
4	F0IF_FLLM	Маска бита F0IF_FLL
3	F0IF_EMPM	Маска бита F0IF_EMP
2	F0VF_OVRM	Маска бита F0VF_OVR
1	F0VF_FLLM	Маска бита F0VF_FLL
0	F0VF_EMPM	Маска бита F0VF_EMP

### 13.1.26 ADCUI\_F1CTR

Таблица 78 – Регистр ADCUI\_F1CTR

Номер	31:20	19, 18	17:10	9, 8	7, 6
Доступ	R/W	-	R/W	R/W	R/W
Сброс	0	-	00	0	0
	F1IRMSOS	-	F1VPHASE	F1VGAIN	F1IGAIN

Номер	5	4	3	2	1	0
Доступ	WO	WO	WO	R/W	-	R/W
Сброс	0	0	0	0	-	0
	F1RVRS	F1RRRS	F1RARS	F1VASEL	-	F1INTEN

Таблица 79 – Описание бит регистра ADCUI\_F1CTR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F1IRMSOS	Калибровка вычислителя среднеквадратического значения тока. Представлено в виде знаковых величин в двоичном коде с дополнением до 2, при применении в вычислителе сдвигается на 4 бита влево. Пример: 007: IRMSOS = $7 \cdot 16 = 112$ ; FF9: IRMSOS = $-7 \cdot 16 = -112$
19, 18	-	Зарезервировано
17...10	F1VPHASE	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -126d (-123 мкс) до +127d (+124 мкс). «0» соответствует синфазному сигналу с током
9, 8	F1VGAIN	Предусилитель в канале напряжения: 00 – 0 dB; 01 – 6 dB; 10 – 12 dB; 11 – 18 dB

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7, 6	F1IGAIN	Предусилитель в канале тока: 00 – 0 dB; 01 – 6 dB; 10 – 12 dB; 11 – 18 dB
5	F1RVRS	Запись единицы в этот бит сбрасывает счетчик переданной полной энергии
4	F1RRRS	Запись в этот бит сбрасывает счетчик переданной реактивной энергии
3	F1RARS	Запись в этот бит сбрасывает счетчик переданной активной энергии
2	F1VASEL	Выбор источника сигнала для сохранения в регистре полной энергии: 0 – полная энергия; 1 – среднеквадратическое значение тока
1	-	Зарезервировано
0	F1INTEN	Отключение интегратора в канале тока: 0 – интегратор включен; 1 – интегратор отключен

### 13.1.27 ADCUI\_F1WC

Таблица 80 – Регистр ADCUI\_F1WC

Номер	31...28	27...16	15...0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	F1WGAIN	F1WATTOS

Таблица 81 – Описание бит регистра ADCUI\_F1WC

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F1WGAIN	Калибровка усиления сигнала в ветви активной мощности для блока F1
15...0	F1WATTOS	Калибровка смещения сигнала в ветви активной мощности для блока F1

### 13.1.28 ADCUI\_F1WATTP

Таблица 82 – Регистр ADCUI\_F1WATTP

Номер	31...0
Доступ	RO
Сброс	
	F1WATTHRP

Таблица 83 – Описание бит регистра ADCUI\_F1WATTP

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1WATTHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной активной энергии

### 13.1.29 ADCUI\_F1WATTN

Таблица 84 – Регистр ADCUI\_F1WATTN

Номер	31...0
Доступ	RO
Сброс	
	F1WATTHRN

Таблица 85 – Описание бит регистра ADCUI\_F1WATTN

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1WATTHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной активной энергии

### 13.1.30 ADCUI\_F1VC

Таблица 86 – Регистр ADCUI\_F1VC

Номер	31...28	27...16	15...0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	F1VARGAIN	F1VAROS

Таблица 87 – Описание бит регистра ADCUI\_F1VC

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F1VARGAIN	Калибровка усиления сигнала в ветви реактивной мощности для блока F1
15...0	F1VAROS	Калибровка смещения сигнала в ветви реактивной мощности для блока F1

### 13.1.31 ADCUI\_F1VARP

Таблица 88 – Регистр ADCUI\_F1VARP

Номер	31...0
Доступ	RO
Сброс	
	F1VARHRP

Таблица 89 – Описание бит регистра ADCUI\_F1VARP

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1VARHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной реактивной энергии

### 13.1.32 ADCUI\_F1VARN

Таблица 90 – Регистр ADCUI\_F1VARN

Номер	31...0
Доступ	RO
Сброс	
	F1VARHRN

Таблица 91 – Описание бит регистра ADCUI\_F1VARN

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1VARHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной реактивной энергии

### 13.1.33 ADCUI\_F1AC

Таблица 92 – Регистр ADCUI\_F1AC

Номер	31...28	27...16	15...12	11...0
Доступ	-	R/W	-	R/W
Сброс	-	000h	-	000h
	-	F1VAGAIN	-	F1VRMSOS

Таблица 93 – Описание бит регистра ADCUI\_F1AC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F1VAGAIN	Калибровка усиления сигнала в ветви полной мощности для блока F1
15...12	-	Зарезервировано
11...0	F1VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения. Представлено в виде знаковых величин в двоичном коде с дополнением до 2, при применении в вычислителе сдвигается на 4 бита влево. Пример: 007: VRMSOS = 7 · 16 = 112; FF9: VRMSOS = -7 · 16 = -112

### 13.1.34 ADCUI\_F1VR

Таблица 94 – Регистр ADCUI\_F1VR

Номер	31...0
Доступ	RO
Сброс	
	F1VAHR

Таблица 95 – Описание бит регистра ADCUI\_F1VR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1VAHR	Старшие 32 бита внутреннего 57-битного аккумулятора полной энергии

### 13.1.35 ADCUI\_F1MD0

Таблица 96 – Регистр ADCUI\_F1MD0

Номер	31..29	28..8	7	6	5	4	3, 2	1, 0
Доступ	-	RO	R/W	R/W	R	R	R/W	R/W
Сброс	-		0	0	0	0	00	00
	-	F1PER_FREQ	V1GAIN	I1GAIN	F1REACTS	F1ACTS	F1ISEL	F1VSEL

Таблица 97 – Описание бит регистра ADCUI\_F1MD0

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...8	F1PER_FREQ	Длительность такта в канале напряжения
7	V1GAIN	Усиление в канале V1: 0 – нет усиления; 1 – +6 дБ усиление
6	I1GAIN	Усиление в канале I1: 0 – нет усиления; 1 – +6 дБ усиление
5	F1REACTS	Знак реактивной энергии в последний период: 0 – минус; 1 – плюс
4	F1ACTS	Знак активной энергии в последний период: 0 – минус; 1 – плюс
3, 2	F1ISEL	Выбор источника сигнала для FIFO канала тока I1 (регистр ADCUI_F1IDAT): 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности



Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1, 0	F1VSEL	Выбор источника сигнала для FIFO канала напряжения V1 (регистр ADCUI_F1VDAT): 00 – отсчеты напряжения до фильтра высоких частот; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты напряжения после фильтра высоких частот

### 13.1.36 ADCUI\_F1MD1

Таблица 98 – Регистр ADCUI\_F1MD1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F1VPKLVL	F1IPKLVL

Таблица 99 – Описание бит регистра ADCUI\_F1MD1

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	F1VPKLVL	Предельный разрешенный уровень напряжения
15...0	F1IPKLVL	Предельный разрешенный уровень тока

### 13.1.37 ADCUI\_F1MD2

Таблица 100 – Регистр ADCUI\_F1MD2

Номер	31...17	16...0
Доступ	-	RO
Сброс	-	00000h
	-	F1PHASE

Таблица 101 – Описание бит регистра ADCUI\_F1MD2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...0	F1PHASE	Фазовый сдвиг канала напряжения V1 по отношению к V0

### 13.1.38 ADCUI\_F1VPEAK

Таблица 102 – Регистр ADCUI\_F1VPEAK

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F1VPEAK

Таблица 103 – Описание бит регистра ADCUI\_F1VPEAK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1VPEAK	Пиковое значение напряжения. Обнуляется чтением из регистра Запись в этот регистр, сбрасывает пиковое значение

### 13.1.39 ADCUI\_F1IPEAK

Таблица 104 – Регистр ADCUI\_F1IPEAK

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F1IPEAK

Таблица 105 – Описание бит регистра ADCUI\_F1IPEAK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1IPEAK	Пиковое значение тока. Обнуляется чтением из регистра Запись в этот регистр, сбрасывает пиковое значение

### 13.1.40 ADCUI\_F1VDAT

Таблица 106 – Регистр ADCUI\_F1VDAT

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	00h	F1VDAT

Таблица 107 – Описание бит регистра ADCUI\_F1VDAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1VDAT	FIFO отсчетов напряжения (или одной из мощностей). FIFO включается при установке любого из бит V1EN, I1EN

### 13.1.41 ADCUI\_F1IDAT

Таблица 108 – Регистр ADCUI\_F1IDAT

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	00h	F1IDAT

Таблица 109 – Описание бит регистра ADCUI\_F1IDAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1IDAT	FIFO отсчетов тока (или одной из мощностей). FIFO включается при установке любого из бит V1EN, I1EN

### 13.1.42 ADCUI\_F1VRMS

Таблица 110 – Регистр ADCUI\_F1VRMS

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	-	F1VRMS

Таблица 111 – Описание бит регистра ADCUI\_F1VRMS

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23..0	F1VRMS	Среднеквадратическое значение напряжения

### 13.1.43 ADCUI\_F1VRMS2

Таблица 112 – Регистр ADCUI\_F1VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F1VRMS2

Таблица 113 – Описание бит регистра ADCUI\_F1VRMS2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1VRMS2	Квадрат среднеквадратического значения напряжения

### 13.1.44 ADCUI\_F1IRMS

Таблица 114 – Регистр ADCUI\_F1IRMS

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F1IRMS

Таблица 115 – Описание бит регистра ADCUI\_F1IRMS

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1IRMS	Среднеквадратическое значение тока

### 13.1.45 ADCUI\_F1IRMS2

Таблица 116 – Регистр ADCUI\_F1IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F1IRMS2

Таблица 117 – Описание бит регистра ADCUI\_F1IRMS2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F1IRMS2	Квадрат среднеквадратического значения тока

### 13.1.46 ADCUI\_F1STAT

Таблица 118 – Регистр ADCUI\_F1STAT

Номер	31...27	26	25	24...22	21
Доступ	-	R/W	R/W	-	R/W
Сброс	-		0	-	0
	-	F1VAROVN	F1WATTOVN	-	F1ZEROCRS

Номер	20	19	18	17	16	15
Доступ	RO	-	RO	R/W	RO	R/W
Сброс	0	-	0	0	0	0
	F1VANLDFL	-	F1VARNLDFL	F1VARSIGN	F1APNLDFL	F1APSIGN

Номер	14, 13	12	11	10	9	8	7
Доступ	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	-	0	0	0	0	0	0
	-	F1ZXTOF	F1VAOV	F1VAROVP	F1WATTOVP	F1PEAKIF	F1PEAKVF

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	RO	RO	R/W	RO	RO
Сброс	0	0	0	0	0	0	0
	F1SAGF	F1IF_OVR	F1IF_FLL	F1IF_EMP	F1VF_OVR	F1VF_FLL	F1VF_EMP

Таблица 119 – Описание бит регистра ADCUI\_F1STAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F1VAROVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной реактивной энергией. Запись «1» сбрасывает этот флаг
25	F1WATTOVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной активной энергией. Запись «1» сбрасывает этот флаг

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...22	-	Зарезервировано
21	F1ZEROCRS	Флаг, что произошло пересечение «0» в канале напряжения. Запись «1» сбрасывает этот флаг
20	F1VANLDFL	Полная мощность ниже уровня сравнения
19	-	Зарезервировано
18	F1VARNLDFL	Реактивная мощность ниже уровня сравнения
17	F1VARSIGN	Смена знака реактивной мощности. Запись «1» сбрасывает этот флаг
16	F1APNLDFL	Активная мощность ниже уровня сравнения.
15	F1APSIGN	Смена знака активной мощности. Запись «1» сбрасывает этот флаг
14, 13	-	Зарезервировано
12	F1ZXTOF	Флаг, что в течении TimeOut не было пересечение напряжением значения «0». Запись «1» сбрасывает этот флаг
11	F1VAOV	Флаг, что произошло переполнения регистра аккумулятора с полной энергией. Запись «1» сбрасывает этот флаг
10	F1VAROVP	Флаг, что произошло переполнения регистра аккумулятора с положительной реактивной энергией. Запись «1» сбрасывает этот флаг
9	F1WATTOVP	Флаг, что произошло переполнения регистра аккумулятора с положительной активной энергией. Запись «1» сбрасывает этот флаг
8	F1PEAKIF	Флаг, что произошло превышение порогового значения тока. Запись «1» сбрасывает этот флаг
7	F1PEAKVF	Флаг, что произошло превышение порогового значения напряжения. Запись «1» сбрасывает этот флаг
6	F1SAGF	Флаг, что произошла просадка напряжения. Запись «1» сбрасывает этот флаг
5	F1IF_OVR	Флаг, что произошло переполнения FIFO F1IDAT. Запись «1» сбрасывает этот флаг
4	F1IF_FLL	Флаг, что FIFO F1IDAT заполнено
3	F1IF_EMP	Флаг, что FIFO F1IDAT пусто
2	F1VF_OVR	Флаг, что произошло переполнения FIFO F1VDAT. Запись «1» сбрасывает этот флаг
1	F1VF_FLL	Флаг, что FIFO F1VDAT заполнено
0	F1VF_EMP	Флаг, что FIFO F1VDAT пусто

13.1.47 ADCUI\_F1MASK

Таблица 120 – Регистр ADCUI\_F1MASK

Номер	31...27	26	25	24...22	21
Доступ	-	R/W	R/W	-	R/W
Сброс	-		0	-	0
	-	F1VAROVNM	F1WATTOVNM	-	F1ZEROCRSM

Номер	20	19	18	17	16
Доступ	R/W	-	R/W	R/W	R/W
Сброс	0	-	0	0	0
	F1VANLDFLM	-	F1VARNLDFLM	F1VARSIGNM	F1APNLDFLM

Номер	15	14, 13	12	11	10	9	8	7
Доступ	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	-	0	0	0	0	0	0
	F1AP SIGNM	-	F1ZX TOFM	F1VA OVM	F1VA ROVPM	F1WAT TOVPM	F1PEA KIFM	F1PEA KVFM

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
	F1SA GFM	F1IF_ OVRM	F1IF_ FLLM	F1IF_ EMPM	F1VF_ OVRM	F1VF_ FLLM	F1VF_ EMPM

Таблица 121 – Описание бит регистра ADCUI\_F1MASK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F1VAROVNM	Маска бита F1VAROVN
25	F1WATTOVNM	Маска бита F1WATTOVN
24...22	-	Зарезервировано
21	F1ZEROCRSM	Маска бита F1ZEROCRS
20	F1VANLDFLM	Маска бита F1VANLDFL
19	-	Зарезервировано
18	F1VARNLDFLM	Маска бита F1VARNLDFL
17	F1VARSIGNM	Маска бита F1VARSIGN
16	F1APNLDFLM	Маска бита F1APNLDFL
15	F1APSIGNM	Маска бита F1APSIGNM
14, 13	-	Зарезервировано
12	F1ZXTOFM	Маска бита F1ZXTOF
11	F1VAOVM	Маска бита F1VAOV
10	F1VAROVPM	Маска бита F1VAROVP
9	F1WATTOVPM	Маска бита F1WATTOVP
8	F1PEAKIFM	Маска бита F1PEAKIF
7	F1PEAKVFM	Маска бита F1PEAKVF

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
6	F1SAGFM	Маска бита F1SAGF
5	F1IF_OVRM	Маска бита F1IF_OVR
4	F1IF_FLLM	Маска бита F1IF_FLL
3	F1IF_EMPM	Маска бита F1IF_EMP
2	F1VF_OVRM	Маска бита F1VF_OVR
1	F1VF_FLLM	Маска бита F1VF_FLL
0	F1VF_EMPM	Маска бита F1VF_EMP

### 13.1.48 ADCUI\_F2CTR

Таблица 122 – Регистр ADCUI\_F2CTR

Номер	31...20	19, 18	17...10	9, 8	7, 6
Доступ	R/W	-	R/W	R/W	R/W
Сброс	0	-	00	0	0
	F2IRMSOS	-	F2VPHASE	F2VGAIN	F2IGAIN

Номер	5	4	3	2	1	0
Доступ	WO	WO	WO	R/W	-	R/W
Сброс	0	0	0	0	-	0
	F2RVRS	F2RRRS	F2RARS	F2VASEL	-	F2INTEN

Таблица 123 – Описание бит регистра ADCUI\_F2CTR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F2IRMSOS	Калибровка вычислителя среднеквадратического значения тока. Представлено в виде знаковых величин в двоичном коде с дополнением до 2, при применении в вычислителе сдвигается на 4 бита влево. Пример: 007: IRMSOS = $7 \cdot 16 = 112$ ; FF9: IRMSOS = $-7 \cdot 16 = -112$
19, 18	-	Зарезервировано
17..10	F2VPHASE	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -126d (-123мкс) до +127d(+124мкс). «0» соответствует синфазному сигналу с током.
9, 8	F2VGAIN	Предусилитель в канале напряжения: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
7, 6	F2IGAIN	Предусилитель в канале тока: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5	F2RVRS	Запись единицы в этот бит сбрасывает счетчик переданной полной энергии
4	F2RRRS	Запись в этот бит сбрасывает счетчик переданной реактивной энергии
3	F2RARS	Запись в этот бит сбрасывает счетчик переданной активной энергии
2	F2VASEL	Выбор источника сигнала для сохранения в регистре полной энергии: 0 – полная энергия; 1 – среднеквадратическое значение тока
1	-	Зарезервировано
0	F2INTEN	Отключение интегратора в канале тока: 0 – интегратор включен; 1 – интегратор отключен

### 13.1.49 ADCUI\_F2WC

Таблица 124 – Регистр ADCUI\_F2WC

Номер	31...28	27...16	15...0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	F2WGAIN	F2WATTOS

Таблица 125 – Описание бит регистра ADCUI\_F2WC

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F2WGAIN	Калибровка усиления сигнала в ветви активной мощности для блока F2
15...0	F2WATTOS	Калибровка смещения сигнала в ветви активной мощности для блока F2

### 13.1.50 ADCUI\_F2WATTP

Таблица 126 – Регистр ADCUI\_F2WATTP

Номер	31...0
Доступ	RO
Сброс	
	F2WATTHRP



Таблица 127 – Описание бит регистра ADCUI\_F2WATTP

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F2WATTHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной активной энергии

### 13.1.51 ADCUI\_F2WATTN

Таблица 128 – Регистр ADCUI\_F2WATTN

Номер	31...0
Доступ	RO
Сброс	
	F2WATTHRN

Таблица 129 – Описание бит регистра ADCUI\_F2WATTN

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F2WATTHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной активной энергии

### 13.1.52 ADCUI\_F2VC

Таблица 130 – Регистр ADCUI\_F2VC

Номер	31...28	27...16	15...0
Доступ	-	R/W	R/W
Сброс	-	000h	0000h
	-	F2VARGAIN	F2VAROS

Таблица 131 – Описание бит регистра ADCUI\_F2VC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F2VARGAIN	Калибровка усиления сигнала в ветви реактивной мощности для блока F2
15...0	F2VAROS	Калибровка смещения сигнала в ветви реактивной мощности для блока F2

### 13.1.53 ADCUI\_F2VARP

Таблица 132 – Регистр ADCUI\_F2VARP

Номер	31...0
Доступ	RO
Сброс	
	F2VARHRP

Таблица 133 – Описание бит регистра ADCUI\_F2VARP

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F2VARHRP	Старшие 32 бита внутреннего 57-битного аккумулятора положительной реактивной энергии

### 13.1.54 ADCUI\_F2VARN

Таблица 134 – Регистр ADCUI\_F2VARN

Номер	31...0
Доступ	RO
Сброс	
	F2VARHRN

Таблица 135 – Описание бит регистра ADCUI\_F2VARN

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F2VARHRN	Старшие 32 бита внутреннего 57-битного аккумулятора отрицательной реактивной энергии

### 13.1.55 ADCUI\_F2AC

Таблица 136 – Регистр ADCUI\_F2AC

Номер	31...28	27...16	15...12	11...0
Доступ	-	R/W	-	R/W
Сброс	-	000h	-	000h
	-	F2VAGAIN	-	F2VRMSOS

Таблица 137 – Описание бит регистра ADCUI\_F2AC

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F2VAGAIN	Калибровка усиления сигнала в ветви полной мощности для блока F2
15...12	-	Зарезервировано
11...0	F2VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения. Представлено в виде знаковых величин в двоичном коде с дополнением до 2, при применении в вычислителе сдвигается на 4 бита влево. Пример: 007: VRMSOS = $7 \cdot 16 = 112$ ; FF9: VRMSOS = $-7 \cdot 16 = -112$

### 13.1.56 ADCUI\_F2VR

Таблица 138 – Регистр ADCUI\_F2VR

Номер	31...0
Доступ	R
Сброс	
	F2VAHR

Таблица 139 – Описание бит регистра ADCUI\_F2VR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F2VAHR	Старшие 32 бита внутреннего 57-битного аккумулятора полной энергии

### 13.1.57 ADCUI\_F2MD0

Таблица 140 – Регистр ADCUI\_F2MD0

Номер	31..29	28..8	7	6	5	4	3, 2	1, 0
Доступ	-	RO	R/W	R/W	R	R	R/W	R/W
Сброс	-	-	0	0	0	0	00	00
	-	F2PER_FREQ	V2GAIN	I2GAIN	F2REACTS	F2ACTS	F2ISEL	F2VSEL

Таблица 141 – Описание бит регистра ADCUI\_F2MD0

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...8	F2PER_FREQ	Длительность такта в канале напряжения
7	V2GAIN	Усиление в канале V2: 0 – нет усиления; 1 – +6 дБ усиление
6	I2GAIN	Усиление в канале I2: 0 – нет усиления; 1 – +6 дБ усиление
5	F2REACTS	Знак реактивной энергии в последний период: 0 – минус; 1 – плюс
4	F2ACTS	Знак активной энергии в последний период: 0 – минус; 1 – плюс
3, 2	F2ISEL	Выбор источника сигнала для FIFO канала тока I2 (регистр ADCUI_F2IDAT): 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1, 0	F2VSEL	Выбор источника сигнала для FIFO канала напряжения V2 (регистр ADCUI_F2VDAT): 00 – отсчеты напряжения до фильтра высоких частот; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты напряжения после фильтра высоких частот

### 13.1.58 ADCUI\_F2MD1

Таблица 142 – Регистр ADCUI\_F2MD1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F2VPKLV L	F2IPKLV L

Таблица 143 – Описание бит регистра ADCUI\_F2MD1

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	F2VPKLV L	Предельный разрешенный уровень напряжения
15...0	F2IPKLV L	Предельный разрешенный уровень тока

### 13.1.59 ADCUI\_F2MD2

Таблица 144 – Регистр ADCUI\_F2MD2

Номер	31...17	16...0
Доступ	-	RO
Сброс	-	00000h
	-	F2PHASE

Таблица 145 – Описание бит регистра ADCUI\_F2MD2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...0	F2PHASE	Фазовый сдвиг канала напряжения V2 по отношению к V0

### 13.1.60 ADCUI\_F2VPEAK

Таблица 146 – Регистр ADCUI\_F2VPEAK

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F2VPEAK

Таблица 147 – Описание бит регистра ADCUI\_F2VPEAK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2VPEAK	Пиковое значение напряжения. Обнуляется чтением из регистра.

### 13.1.61 ADCUI\_F2IPEAK

Таблица 148 – Регистр ADCUI\_F2IPEAK

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F2IPEAK

Таблица 149 – Описание бит регистра ADCUI\_F2IPEAK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2IPEAK	Пиковое значение тока. Обнуляется чтением из регистра

### 13.1.62 ADCUI\_F2VDAT

Таблица 150 – Регистр ADCUI\_F2VDAT

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	00h	F2VDAT

Таблица 151 – Описание бит регистра ADCUI\_F2VDAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2VDAT	FIFO отсчетов напряжения (или одной из мощностей). FIFO включается при установке любого из бит V2EN, I2EN

### 13.1.63 ADCUI\_F2IDAT

Таблица 152 – Регистр ADCUI\_F2IDAT

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	00h	F2IDAT

Таблица 153 – Описание бит регистра ADCUI\_F2IDAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2IDAT	FIFO отсчетов тока (или одной из мощностей).FIFO включается при установке любого из бит V2EN, I2EN

### 13.1.64 ADCUI\_F2VRMS

Таблица 154 – Регистр ADCUI\_F2VRMS

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F2VRMS

Таблица 155 – Описание бит регистра ADCUI\_F2VRMS

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2VRMS	Среднеквадратическое значение напряжения

### 13.1.65 ADCUI\_F2VRMS2

Таблица 156 – Регистр ADCUI\_F2VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F2VRMS2

Таблица 157 – Описание бит регистра ADCUI\_F2VRMS2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F2VRMS2	Квадрат среднеквадратического значения напряжения

### 13.1.66 ADCUI\_F2IRMS

Таблица 158 – Регистр ADCUI\_F2IRMS

Номер	31...24	23...0
Доступ		R/W
Сброс		000000h
	-	F2IRMS

Таблица 159 – Описание бит регистра ADCUI\_F2IRMS

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2IRMS	Среднеквадратическое значение тока

### 13.1.67 ADCUI\_F2IRMS2

Таблица 160 – Регистр ADCUI\_F2IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F2IRMS2

Таблица 161 – Описание бит регистра ADCUI\_F2IRMS2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	F2IRMS2	Квадрат среднеквадратического значения тока

### 13.1.68 ADCUI\_F2STAT

Таблица 162 – Регистр ADCUI\_F2STAT

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	RO	RO	R/W	RO	RO
Сброс	0	0	0	0	0	0	0
	F2SAGF	F2IF_OVR	F2IF_FLL	F2IF_EMP	F2VF_OVR	F2VF_FLL	F2VF_EMP

Номер	15	14, 13	12	11	10	9	8	7
Доступ	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	-	0	0	0	0	0	0
	F2AP SIGN	-	F2ZX TOF	F2VA OV	F2VA ROVP	F2WAT TOVP	F2PEA KIF	F2PEA KVF

Номер	31..27	26	25	24...22	21	20	19	18	17	16
Доступ	-	R/W	R/W	-	R/W	RO	-	RO	R/W	RO
Сброс	-		0	-	0	0	-	0	0	0
	-	F2VA ROVN	F2WAT TOVN	-	F2ZER OCRS	F2VAN LDLFL	-	F2VAR NLDFL	F2VAR SIGN	F2APN LDLFL

Таблица 163 – Описание бит регистра ADCUI\_F2STAT

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F2VAROVN	Флаг, что произошло переполнение регистра аккумулятора с отрицательной реактивной энергией. Запись «1» сбрасывает этот флаг
25	F2WATTOVN	Флаг, что произошло переполнение регистра аккумулятора с отрицательной активной энергией. Запись «1» сбрасывает этот флаг
24...22	-	Зарезервировано
21	F2ZEROCRS	Флаг, что произошло пересечение «0» в канале напряжения. Запись «1» сбрасывает этот флаг
20	F2VANLDFL	Полная мощность ниже уровня сравнения
19	-	Зарезервировано
18	F2VARNLDFL	Реактивная мощность ниже уровня сравнения.
17	F2VARSIGN	Смена знака реактивной мощности. Запись «1» сбрасывает этот флаг
16	F2APNLDFL	Активная мощность ниже уровня сравнения.
15	F2APSIGN	Смена знака активной мощности. Запись «1» сбрасывает этот флаг
14, 13	-	Зарезервировано.
12	F2ZXTOF	Флаг, что в течении TimeOut не было пересечения напряжением значения «0». Запись «1» сбрасывает этот флаг
11	F2VAOV	Флаг, что произошло переполнение регистра аккумулятора с полной энергией. Запись «1» сбрасывает этот флаг
10	F2VAROVP	Флаг, что произошло переполнение регистра аккумулятора с положительной реактивной энергией. Запись «1» сбрасывает этот флаг
9	F2WATTOVP	Флаг, что произошло переполнение регистра аккумулятора с положительной активной энергией. Запись «1» сбрасывает этот флаг
8	F2PEAKIF	Флаг, что произошло превышение порогового значения тока. Запись «1» сбрасывает этот флаг
7	F2PEAKVF	Флаг, что произошло превышение порогового значения напряжения. Запись «1» сбрасывает этот флаг
6	F2SAGF	Флаг, что произошла просадка напряжения. Запись «1» сбрасывает этот флаг
5	F2IF_OVR	Флаг, что произошло переполнение FIFO F2IDAT. Запись «1» сбрасывает этот флаг
4	F2IF_FLL	Флаг, что FIFO F2IDAT заполнено
3	F2IF_EMP	Флаг, что FIFO F2IDAT пусто



Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
2	F2VF_OVR	Флаг, что произошло переполнение FIFO F2VDAT. Запись «1» сбрасывает этот флаг
1	F2VF_FLL	Флаг, что FIFO F2VDAT заполнено
0	F2VF_EMP	Флаг, что FIFO F2VDAT пусто

### 13.1.69 ADCUI\_F2MASK

Таблица 164 – Регистр ADCUI\_F2MASK

Номер	31..27	26	25	24...22	21	20	19	18	17	16
Доступ	-	R/W	R/W	-	R/W	R/W	-	R/W	R/W	R/W
Сброс	-		0	-	0	0	-	0	0	0
	-	F2VAR OVNM	F2WATT OVNM	-	F2ZERO CRSM	F2VANL DFLM	-	F2VAR NLDFLM	F2VAR SIGNM	F2APNL DFLM

Номер	15	14, 13	12	11	10	9	8	7
Доступ	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	-	0	0	0	0	0	0
	F2APS IGNM	-	F2ZX TOFM	F2VA OVM	F2VAR OVPM	F2WAT TOVPM	F2PEA KIFM	F2PEA KVFM

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
	F2SA GFM	F2IF_ OVRM	F2IF_ FLLM	F2IF_ EMPM	F2VF_ OVRM	F2VF_ FLLM	F2VF_ EMPM

Таблица 165 – Описание бит регистра ADCUI\_F2MASK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F2VAROVNM	Маска бита F2VAROVN
25	F2WATTOVNM	Маска бита F2WATTOVN
24...22	-	Зарезервировано
21	F2ZEROCRSM	Маска бита F2ZEROCRS
20	F2VANLDFLM	Маска бита F2VANLDFL
19	-	Зарезервировано
18	F2VARNLDFLM	Маска бита F2VARNLDFL
17	F2VARSIGNM	Маска бита F2VARSIGN
16	F2APNLDFLM	Маска бита F2APNLDFL
15	F2APSIGNM	Маска бита F2APSIGNM
14, 13	-	Зарезервировано
12	F2ZXTOFM	Маска бита F2ZXTOF

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
11	F2VAOVM	Маска бита F2VAOV
10	F2VAROVPM	Маска бита F2VAROVP
9	F2WATTOVPM	Маска бита F2WATTOVP
8	F2PEAKIFM	Маска бита F2PEAKIF
7	F2PEAKVFM	Маска бита F2PEAKVF
6	F2SAGFM	Маска бита F2SAGF
5	F2IF_OVRM	Маска бита F2IF_OVR
4	F2IF_FLLM	Маска бита F2IF_FLL
3	F2IF_EMPM	Маска бита F2IF_EMP
2	F2VF_OVRM	Маска бита F2VF_OVR
1	F2VF_FLLM	Маска бита F2VF_FLL
0	F2VF_EMPM	Маска бита F2VF_EMP

### 13.1.70 ADCUI\_CCAL1 (в ревизии 1)

Таблица 166 – Регистр ADCUI\_CCAL1

Номер	31...24	23...12	11...0
Доступ			
Сброс			
	-	I0BGAIN	V0BGAIN

Таблица 167 – Описание бит регистра ADCUI\_CCAL1

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...12	I0BGAIN	Калибровочный коэффициент канала I0
11...0	V0BGAIN	Калибровочный коэффициент канала V0

### 13.1.71 ADCUI\_CCAL1 (доступно с ревизии 2)

Таблица 168 – Регистр ADCUI\_CCAL1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0	0
	I0BGAIN	V0BGAIN

Таблица 169 – Описание бит регистра ADCUI\_CCAL1

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	I0BGAIN	Калибровочный коэффициент канала I0
15...0	V0BGAIN	Калибровочный коэффициент канала V0

**13.1.72 ADCUI\_CCAL2 (в ревизии 1)**

Таблица 170 – Регистр ADCUI\_CCAL2

Номер	31...24	23...12	11...0
Доступ			
Сброс			
	-	I1BGAIN	V1BGAIN

Таблица 171 – Описание бит регистра ADCUI\_CCAL2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...12	I1BGAIN	Калибровочный коэффициент канала I1
11...0	V1BGAIN	Калибровочный коэффициент канала V1

**13.1.73 ADCUI\_CCAL2 (доступно с ревизии 2)**

Таблица 172 – Регистр ADCUI\_CCAL2

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0	0
	I1BGAIN	V1BGAIN

Таблица 173 – Описание бит регистра ADCUI\_CCAL2

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	I1BGAIN	Калибровочный коэффициент канала I1
15...0	V1BGAIN	Калибровочный коэффициент канала V1

**13.1.74 ADCUI\_CCAL3 (в ревизии 1)**

Таблица 174 – Регистр ADCUI\_CCAL3

Номер	31...24	23...12	11...0
Доступ			
Сброс			
	-	I2BGAIN	V2BGAIN

Таблица 175 – Описание бит регистра ADCUI\_CCAL3

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...12	I2BGAIN	Калибровочный коэффициент канала I2
11...0	V2BGAIN	Калибровочный коэффициент канала V2

**13.1.75 ADCUI\_CCAL3 (доступно с ревизии 2)**

Таблица 176 – Регистр ADCUI\_CCAL3

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0	0
	I2BGAIN	V2BGAIN

Таблица 177 – Описание бит регистра ADCUI\_CCAL3

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	I2BGAIN	Калибровочный коэффициент канала I2
15...0	V2BGAIN	Калибровочный коэффициент канала V2

**13.1.76 ADCUI\_CCAL4 (в ревизии 1)**

Таблица 178 – Регистр ADCUI\_CCAL4

Номер	31...12	11...0
Доступ		
Сброс		
	-	I3BGAIN

Таблица 179 – Описание бит регистра ADCUI\_CCAL4

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...12	-	Зарезервировано
11...0	I3BGAIN	Калибровочный коэффициент канала I3

**13.1.77 ADCUI\_CCAL4 (доступно с ревизии 2)**

Таблица 180 – Регистр ADCUI\_CCAL4

Номер	31...16	15...0
Доступ	U	R/W
Сброс	-	0
	-	I3BGAIN

Таблица 181 – Описание бит регистра ADCUI\_CCAL4

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...0	I3BGAIN	Калибровочный коэффициент канала I3

### 13.1.78 ADCUI\_F0IRMS\_INACTIVE

Таблица 182 – Регистр ADCUI\_F0IRMS\_INACTIVE

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	-	F0IRMS_INACTIVE

Таблица 183 – Описание бит регистра ADCUI\_F0IRMS\_INACTIVE

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0IRMS_INACTIVE	Среднеквадратическое значение тока неактивного канала

### 13.1.79 ADCUI\_F0WATTP\_L

Таблица 184 – Регистр ADCUI\_F0WATTP\_L

Номер	24...0
Доступ	RO
Сброс	
	F0WATTHRP_L

Таблица 185 – Описание бит регистра ADCUI\_F0WATTP\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F0WATTHRP_L	Младшие 25 бит внутреннего 57 битного аккумулятора положительной активной энергии. Необходимо производить чтение только после старшей части.

### 13.1.80 ADCUI\_F0WATTN\_L

Таблица 186 – Регистр ADCUI\_F0WATTN\_L

Номер	24...0
Доступ	RO
Сброс	
	F0WATTHRN_L

Таблица 187 – Описание бит регистра ADCUI\_F0WATTN\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F0WATTHRN_L	Младшие 25 бит внутреннего 57 битного аккумулятора отрицательной активной энергии. Необходимо производить чтение только после старшей части

### 13.1.81 ADCUI\_F0VARP\_L

Таблица 188 – Регистр ADCUI\_F0VARP\_L

Номер	24..0
Доступ	RO
Сброс	
	F0VARHRP_L

Таблица 189 – Описание бит регистра ADCUI\_F0VARP\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F0VARHRP_L	Младшие 25 бит внутреннего 57-битного аккумулятора положительной реактивной энергии. Необходимо производить чтение только после старшей части

### 13.1.82 ADCUI\_F0VARN\_L

Таблица 190 – Регистр ADCUI\_F0VARN\_L

Номер	24..0
Доступ	RO
Сброс	
	F0VARHRN_L

Таблица 191 – Описание бит регистра ADCUI\_F0VARN\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F0VARHRN_L	Младшие 25 бит внутреннего 57-битного аккумулятора отрицательной реактивной энергии. Необходимо производить чтение только после старшей части.

### 13.1.83 ADCUI\_F0VR\_L

Таблица 192 – Регистр ADCUI\_F0VR\_L

Номер	24...0
Доступ	RO
Сброс	
	F0VAHR_L

Таблица 193 – Описание бит регистра ADCUI\_F0VR\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F0VAHR_L	Младшие 25 бит внутреннего 57-битного аккумулятора полной энергии. Необходимо производить чтение только после старшей части

### 13.1.84 ADCUI\_F1WATTP\_L

Таблица 194 – Регистр ADCUI\_F1WATTP\_L

Номер	24...0
Доступ	RO
Сброс	
	F1WATTHRP_L

Таблица 195 – Описание бит регистра ADCUI\_F1WATTP\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F1WATTHRP_L	Младшие 25 бит внутреннего 57 битного аккумулятора положительной активной энергии. Необходимо производить чтение только после старшей части

### 13.1.85 ADCUI\_F1WATTN\_L

Таблица 196 – Регистр ADCUI\_F1WATTN\_L

Номер	24...0
Доступ	RO
Сброс	
	F1WATTHRN_L

Таблица 197 – Описание бит регистра ADCUI\_F1WATTN\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F1WATTHRN_L	Младшие 25 бит внутреннего 57 битного аккумулятора отрицательной активной энергии. Необходимо производить чтение только после старшей части

### 13.1.86 ADCUI\_F1VARP\_L

Таблица 198 – Регистр ADCUI\_F1VARP\_L

Номер	24..0
Доступ	RO
Сброс	
	F1VARHRP_L

Таблица 199 – Описание бит регистра ADCUI\_F1VARP\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F1VARHRP_L	Младшие 25 бит внутреннего 57-битного аккумулятора положительной реактивной энергии. Необходимо производить чтение только после старшей части

### 13.1.87 ADCUI\_F1VARN\_L

Таблица 200 – Регистр ADCUI\_F1VARN\_L

Номер	24..0
Доступ	RO
Сброс	
	F1VARHRN_L

Таблица 201 – Описание бит регистра ADCUI\_F1VARN\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F1VARHRN_L	Младшие 25 бит внутреннего 57-битного аккумулятора отрицательной реактивной энергии. Необходимо производить чтение только после старшей части

### 13.1.88 ADCUI\_F1VR\_L

Таблица 202 – Регистр ADCUI\_F1VR\_L

Номер	24...0
Доступ	RO
Сброс	
	F1VAHR_L

Таблица 203 – Описание бит регистра ADCUI\_F1VR\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F1VAHR_L	Младшие 25 бит внутреннего 57-битного аккумулятора полной энергии. Необходимо производить чтение только после старшей части

### 13.1.89 ADCUI\_F2WATTP\_L

Таблица 204 – Регистр ADCUI\_F2WATTP\_L

Номер	24...0
Доступ	RO
Сброс	
	F2WATTHRP_L

Таблица 205 – Описание бит регистра ADCUI\_F2WATTP\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F2WATTHRP_L	Младшие 25 бит внутреннего 57 битного аккумулятора положительной активной энергии. Необходимо производить чтение только после старшей части



**13.1.90 ADCUI\_F2WATTN\_L**

Таблица 206 – Регистр ADCUI\_F2WATTN\_L

Номер	24...0
Доступ	RO
Сброс	
	F2WATTHR_N_L

Таблица 207 – Описание бит регистра ADCUI\_F2WATTN\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F2WATTHR_N_L	Младшие 25 бит внутреннего 57 битного аккумулятора отрицательной активной энергии. Необходимо производить чтение только после старшей части

**13.1.91 ADCUI\_F2VARP\_L**

Таблица 208 – Регистр ADCUI\_F2VARP\_L

Номер	24..0
Доступ	RO
Сброс	
	F2VARHRP_L

Таблица 209 – Описание бит регистра ADCUI\_F2VARP\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F2VARHRP_L	Младшие 25 бит внутреннего 57-битного аккумулятора положительной реактивной энергии. Необходимо производить чтение только после старшей части

**13.1.92 ADCUI\_F2VARN\_L**

Таблица 210 – Регистр ADCUI\_F2VARN\_L

Номер	24..0
Доступ	RO
Сброс	
	F2VARHRN_L

Таблица 211 – Описание бит регистра ADCUI\_F2VARN\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F2VARHRN_L	Младшие 25 бит внутреннего 57-битного аккумулятора отрицательной реактивной энергии. Необходимо производить чтение только после старшей части

**13.1.93 ADCUI\_F2VR\_L**

Таблица 212 – Регистр ADCUI\_F2VR\_L

Номер	24...0
Доступ	RO
Сброс	
	F2VAHR_L

Таблица 213 – Описание бит регистра ADCUI\_F2VR\_L

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F2VAHR_L	Младшие 25 бит внутреннего 57-битного аккумулятора полной энергии. Необходимо производить чтение только после старшей части

**13.1.94 ADCUI\_LOAD (доступно с ревизии 2)**

Таблица 214 – Регистр ADCUI\_LOAD

Номер	31...18	17...12	11...6	5...0
Доступ	U	R/W	R/W	R/W
Сброс	-	00	0	0
		VANOLOAD	VARNLOAD	APNOLOAD

Таблица 215 – Описание бит регистра ADCUI\_LOAD

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...18	-	Зарезервировано
17...12	VANOLOAD	Режим «без нагрузки» при вычислении полной энергии: 000000 – при вычислении используются биты 18, 17 регистра ADCUI_CTRL1; 000001 – не учитывается энергия ниже 0,00156 % от полной шкалы; 000010 – не учитывается энергия ниже 0,00312 % от полной шкалы; ... 111111 – не учитывается энергия ниже 0,1 % от полной шкалы
11...6	VARNLOAD	Режим «без нагрузки» при вычислении реактивной энергии: 000000 – при вычислении используются биты 16, 15 регистра ADCUI_CTRL1; 000001 – не учитывается энергия ниже 0,00078 % от полной шкалы; 000010 – не учитывается энергия ниже 0,00156 % от полной шкалы; ... 111111 – не учитывается энергия ниже 0,05 % от полной шкалы

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5...0	APNOLOAD	Режим «без нагрузки» при вычислении активной энергии: 000000 – при вычислении используются биты 13...12 регистра ADCUI_CTRL1; 000001 – не учитывается энергия ниже 0,00078 % от полной шкалы; 000010 – не учитывается энергия ниже 0,00156 % от полной шкалы; ... 111111 – не учитывается энергия ниже 0,05 % от полной шкалы

### 13.1.95 ADCUI\_F0VRMS\_TRUE (доступно с ревизии 2)

Таблица 216 – Регистр ADCUI\_F0VRMS\_TRUE

Номер	31...29	28...0
Доступ	U	R/W
Сброс	-	0
	-	ADCUI_F0VRMS_TRUE

Таблица 217 – Описание бит регистра ADCUI\_F0VRMS\_TRUE

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...0	ADCUI_F0VRMS_TRUE	Действующее значение напряжения на периоде канала F0. Обновляется при пересечении сигнала напряжения «0».

### 13.1.96 ADCUI\_F1VRMS\_TRUE (доступно с ревизии 2)

Таблица 218 – Регистр ADCUI\_F0VRMS\_TRUE

Номер	31...29	28...0
Доступ	U	R/W
Сброс	-	0
	-	ADCUI_F1VRMS_TRUE

Таблица 219 – Описание бит регистра ADCUI\_F1VRMS\_TRUE

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...0	ADCUI_F1VRMS_TRUE	Действующее значение напряжения на периоде канала F1. Обновляется при пересечении сигнала напряжения «0».

**13.1.97 ADCUI\_F2VRMS\_TRUE (доступно с ревизии 2)**

Таблица 220 – Регистр ADCUI\_F0VRMS\_TRUE

Номер	31...29	28...0
Доступ	U	R/W
Сброс	-	0
	-	ADCUI_F2VRMS_TRUE

Таблица 221 – Описание бит регистра ADCUI\_F2VRMS\_TRUE

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...0	ADCUI_F2VRMS_TRUE	Действующее значение напряжения на периоде канала F2. Обновляется при пересечении сигнала напряжения «0».

## 14 Алгоритмы вычисления окончательных результатов и их соответствия внешним сигналам

Все параметры вычисленных значений зависят от схемы включения микросхемы, а также от формата выходных данных. На рисунках 19, 20 приведены два вида включения АЦП: полностью дифференциальное и недифференциальное включение.

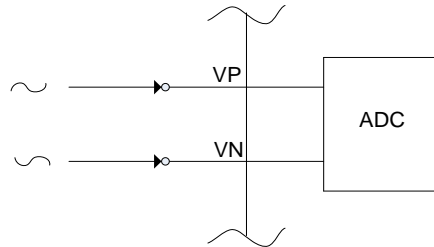


Рисунок 19 – Дифференциальное включение

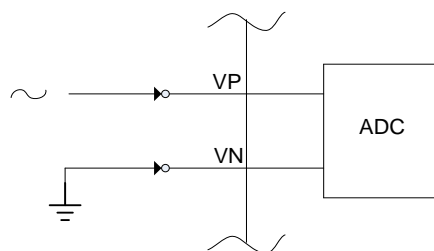


Рисунок 20 – Недифференциальное включение

Необходимо иметь в виду, что значения напряжения и токов после АЦП в случае недифференциального включения в два раза меньше дифференциального, а мощностные характеристики в четыре раза меньше.

Для коррекции фазового сдвига в канале тока относительно канала напряжения в системе присутствует конфигурируемая линия задержки как показано на рисунке 21.

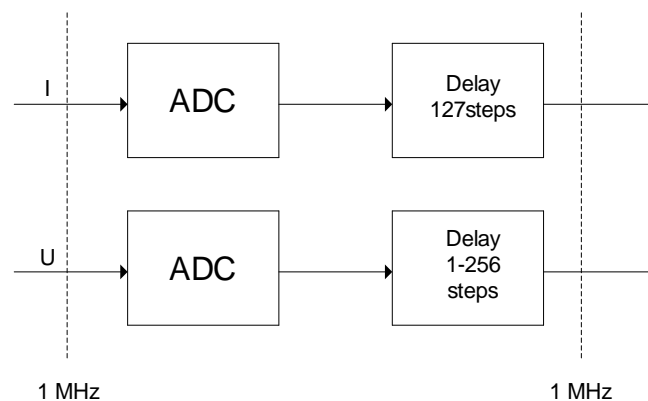


Рисунок 21 – Контролируемый фазовый сдвиг в канале напряжения

Изменяя линию задержки в канале напряжения можно регулировать временной сдвиг одного канала относительно другого. Так как частота отсчетов после АЦП равна 1,024 МГц, то один шаг равен 1/20480 периода сигнала с частотой 50 Гц или 0,018°. Необходимо иметь в виду, что в этой системе сдвиг осуществляется во временной области, поэтому фазовый сдвиг в градусах зависит от частоты.

В качестве децимирующего фильтра используется фильтр со структурой  $\sin^3 c^3$ , его характеристики приведены на рисунках 22, 23.

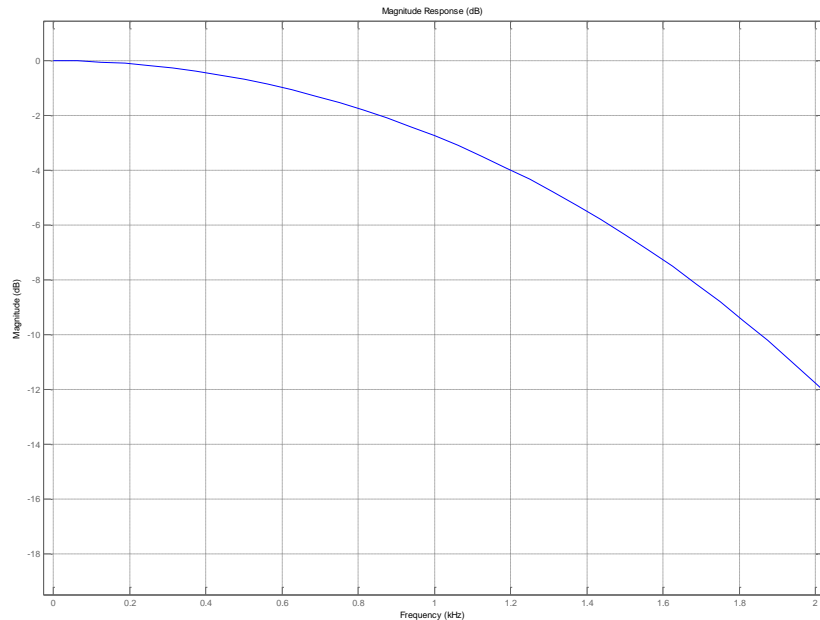


Рисунок 22 – Характеристика децимирующего фильтра в полосе 2 кГц

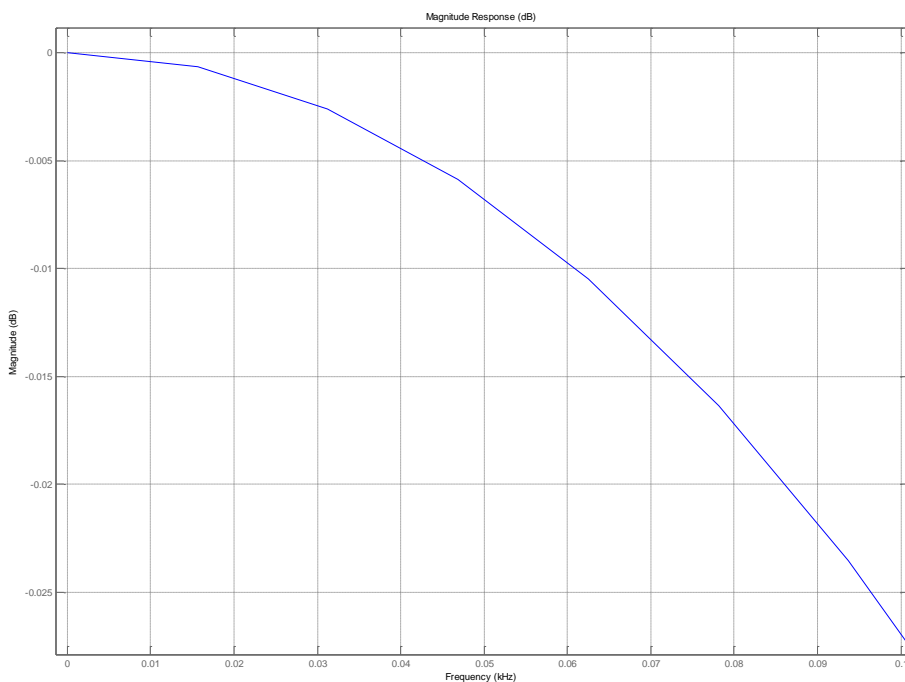


Рисунок 23 – Характеристика децимирующего фильтра в полосе 100 Гц

Как видно из вышеприведенных характеристик фильтр подавляет частоты близкие к 2 кГц до величин 12 дБ, что следует учитывать при измерении гармоник основного тона. В полосе до 100 Гц подавление незначительно (на частоте 50 Гц подавление 0,008 дБ).

Для коррекции влияния смещения в каналах тока и напряжения на вычисленную мощность после децимирующего фильтра в канале тока стоит высокочастотный фильтр. Он убирает постоянную составляющую сигнала. Частота среза фильтра равна 1 Гц.

Каждый канал АЦП имеет буфер FIFO размером восемь отсчетов, предназначенный для хранения отсчетов тока или напряжения, а также отсчетов активной мощности, реактивной мощности и полной мощности (только для каналов I0-I3). Выбор

источника отсчетов для буфера FIFO осуществляется в поле F0I3SEL регистра ADCUI\_F0AC для канала тока I3, и в полях FxISEL, FxVSEL регистра ADCUI\_FxMD0 для остальных каналов. Запись выбранных отсчетов в буфер FIFO выполняется с частотой передискретизации, заданной в поле OSR\_CONF регистра ADCUI\_CTRL1.

Если FIFO каналов сконфигурированы на прием отсчетов тока и напряжения (FxISEL = 00, FxVSEL = 00), то значения отсчетов FIFO можно перевести в напряжения на входе по формулам, приведенным в таблице 222. Значения приведены для усиления PGA равного 0 дБ. Отсчеты, записанные в FIFO, представлены в двоичном формате с дополнением до 2.

Таблица 222 – Формулы расчета входного напряжения

Режим	Входное напряжение АЦП, В	
Дифференциальное включение, 24 бит. режим	$\frac{F_xV_{DAT} \cdot U_{REF0}}{2^{23}}$ ,	$\frac{F_xI_{DAT} \cdot U_{REF0}}{2^{23}}$
Дифференциальное включение, 16 бит. режим	$\frac{F_xV_{DAT} \cdot U_{REF0}}{2^{15}}$ ,	$\frac{F_xI_{DAT} \cdot U_{REF0}}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{F_xV_{DAT} \cdot U_{REF0}}{2^{23}}$ ,	$\frac{F_xI_{DAT} \cdot U_{REF0}}{2^{23}}$
Недифференциальное включение, 16 бит. режим	$\frac{F_xV_{DAT} \cdot U_{REF0}}{2^{15}}$ ,	$\frac{F_xI_{DAT} \cdot U_{REF0}}{2^{15}}$

Каждый из каналов тока может быть скорректирован с помощью коэффициентов FxIyGAIN, IxGAIN и IxBGAIN в соответствии с формулой (1). Значение IxBGAIN записывается в двоичном формате с дополнением до 2.

$$I_{cor} = I_{ADC} \cdot 2^{F_xI_yGAIN} \cdot 2^{I_xGAIN} \cdot \left(1 + \frac{I_xBGAIN}{2^{11}}\right) \text{ (для микросхем до ревизии 2);} \quad (1)$$

$$I_{cor} = I_{ADC} \cdot 2^{F_xI_yGAIN} \cdot 2^{I_xGAIN} \cdot \left(1 + \frac{I_xBGAIN}{2^{15}}\right) \text{ (для микросхем с ревизии 2),}$$

Если для третьего канала тока источник сигнала выбран до фильтра высоких частот (F0I3SEL = 1), то в значении, хранящемся в регистре ADCUI\_F0I3DAT, не учитывается коэффициент IxBGAIN.

Каждый из каналов напряжений может быть скорректирован с помощью коэффициентов FxVGAIN, VxGAIN и VxBGAIN в соответствии с формулой (2). Значение VxBGAIN записывается в двоичном формате с дополнением до 2.

$$V_{cor} = V_{ADC} \cdot 2^{F_xV_GAIN} \cdot 2^{V_xGAIN} \cdot \left(1 + \frac{V_xBGAIN}{2^{11}}\right) \text{ (для микросхем до ревизии 2);} \quad (2)$$

$$V_{cor} = V_{ADC} \cdot 2^{F_xV_GAIN} \cdot 2^{V_xGAIN} \cdot \left(1 + \frac{V_xBGAIN}{2^{15}}\right) \text{ (для микросхем с ревизии 2).}$$

В регистрах FxVRMS и FxIRMS хранится вычисленная величина среднеквадратического значения тока и напряжения в соответствующей фазе. В таблице приведены значения среднеквадратических величин.

Таблица 223 – Значения среднеквадратических величин напряжения

Режим	Напряжение, В	
Дифференциальное включение	$\frac{FxVRMS \cdot UREFO}{2^{23}}$	$\frac{FxIRMS \cdot UREFO}{2^{23}}$
Недифференциальное включение	$\frac{FxVRMS \cdot UREFO}{2^{22}}$	$\frac{FxIRMS \cdot UREFO}{2^{22}}$

Для вычисления среднеквадратического значения используется алгоритм, приведенный ниже (для примера выбран канал напряжения, для канала тока алгоритм идентичный).

Входной сигнал представлен в виде

$$V(t) = \sqrt{2} \cdot V_{rms} \cdot \sin(\omega t) . \quad (3)$$

Отсчеты напряжения поступают с частотой 4 кГц. Далее каждый отсчет возводится в квадрат, что дает следующий результат

$$V^2(t) = 2 \cdot V_{rms} \cdot \sin^2(\omega t) = V_{rms}^2 - V_{rms}^2 \cdot \cos(2\omega t) . \quad (4)$$

Таким образом, мы имеем сигнал с постоянной составляющей равной среднеквадратическому значению напряжения и пульсацией с удвоенной частотой по сравнению с входным сигналом. Для фильтрации пульсации полученный сигнал пропускается через фильтр с частотой среза 2 Гц. Этот фильтр подавляет пульсации на частоте 100 Гц (50 Гц • 2) с коэффициентом 35 дБ. Отфильтрованный сигнал поступает на блок извлечения квадратного корня. Результирующий сигнал имеет также пульсации, но ослабленные фильтром. Поэтому рекомендуется использовать режим синхронизации записи среднеквадратического значения с моментом перехода напряжения через 0 (ZXRMS=1).

После извлечения квадратного корня величину смещения среднеквадратического значения можно скорректировать с помощью 12-битных значений FxVRMSOS и FxIRMSOS по формуле (5). Перед корректировкой значение сдвигается на 8 бит вправо, что дает шаг корректировки в 256 меньше. Эта корректировка нужна для того, чтобы избавиться от ошибки, вызванной шумами на входе АЦП, которые после возведения в квадрат и накопления будут давать отклонения среднего уровня величины  $V^2(t)$ .

$$FxVRMS = FxVRMS + FxVRMSOS \cdot 16 . \quad (5)$$

Значения FxVRMSOS и FxIRMSOS представлены в виде знаковых величин в двоичном коде с дополнением до 2. Значение RMS, полученное в ходе калибровки при помощи FxVRMSOS и FxIRMSOS со знаком «-», должно контролироваться и не опускаться за пределы нуля.



В регистрах FxVRMS2 и FxIRMS2 хранятся значения среднеквадратического значения напряжения и тока до извлечения квадратного корня. В таблице 224 приведены значения квадратов среднеквадратических величин.

Таблица 224 – Значения квадратов среднеквадратических величин

Режим	Напряжение, В <sup>2</sup>	
Дифференциальное включение	$\frac{FxVRMS2 \cdot UREFO}{2^{30}}$	$\frac{FxIRMS2 \cdot UREFO}{2^{30}}$
Недифференциальное включение	$\frac{FxVRMS2 \cdot UREFO}{2^{28}}$	$\frac{FxIRMS2 \cdot UREFO}{2^{28}}$

Для вычисления реактивной мощности необходимо сдвинуть сигнал в канале тока на 90°. Это осуществляется с помощью фильтров, которые в достаточно широком диапазоне сохраняют сдвиг равный 90° для обоих каналов. На рисунке 24 приведена его фазовая характеристика.

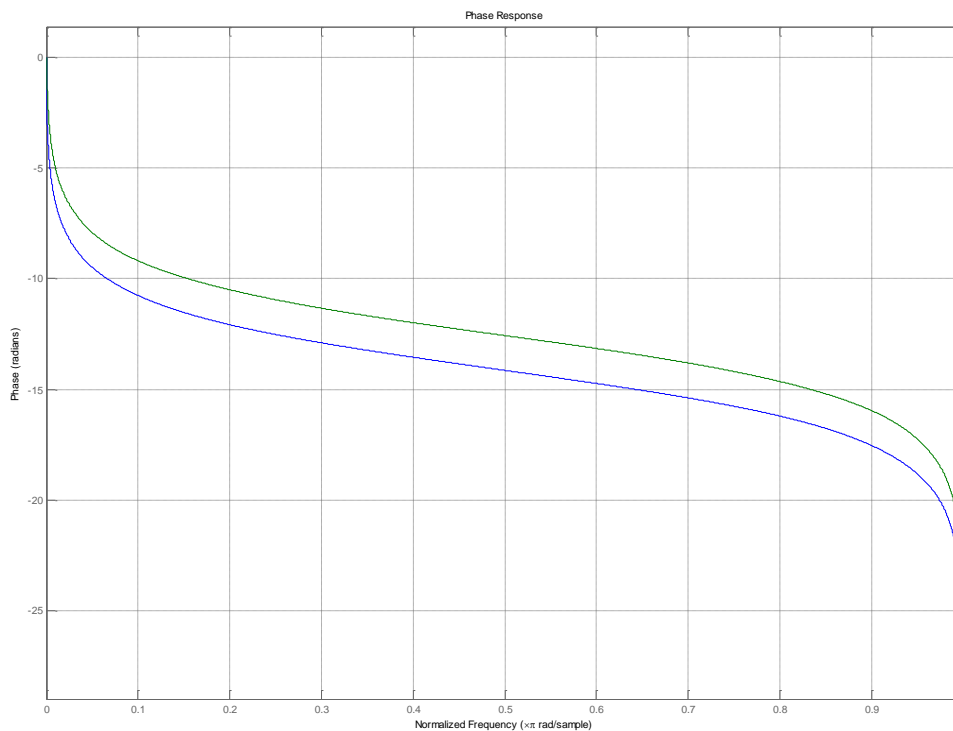


Рисунок 24 – Фильтр для сдвига сигнала на 90°

Для вычисления активной и реактивной энергии используется подход, похожий на вычисление среднеквадратичного значения, только без извлечения квадратного корня

$$V(t) = \sqrt{2} \cdot V_{rms} \cdot \sin(\omega t), \tag{6}$$

$$I(t) = \sqrt{2} \cdot I_{rms} \cdot \sin(\omega t). \tag{7}$$

Тогда мгновенное значение мощности равно произведению тока на напряжение

$$P(t) = V(t) \cdot I(t) = V_{rms} \cdot I_{rms} - V_{rms} \cdot I_{rms} \cdot \cos(2\omega t). \tag{8}$$

Среднее значение мощности за целое количество тактов равно

$$P = \frac{1}{nT} \int_0^{nT} P(t)dt = V_{rms} \cdot I_{rms} . \quad (9)$$

Таким образом, мгновенное значение мощности равно постоянно составляющей произведения тока на напряжение. Для выделения постоянной составляющей используется низкочастотный фильтр с частотой среза 7 Гц. Значение мгновенной мощности можно получить из FIFO  $FxV DAT$  и  $FxIDAT$ .

Если FIFO каналов сконфигурированы на прием отсчетов мощностей ( $FxISEL=01/10/11$ ,  $FxVSEL=01/10$ ), то значения отсчетов FIFO можно перевести в значения мощностей по следующим простейшим формулам. Значение приведены для усиления PGA, равного 0 дБ. Отсчеты, записанные в FIFO, представлены в двоичном формате с дополнением до 2.

Таблица 225 – Перевод значений отсчетов FIFO в значения мощностей

Режим	Мощность, В·А	
Дифференциальное включение, 24 бит. режим	$\frac{FxV DAT \cdot UREFO}{2^{23}}$ ,	$\frac{FxIDAT \cdot UREFO}{2^{23}}$
Дифференциальное включение, 16 бит. режим	$\frac{FxV DAT \cdot UREFO}{2^{15}}$ ,	$\frac{FxIDAT \cdot UREFO}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{FxV DAT \cdot UREFO}{2^{21}}$ ,	$\frac{FxIDAT \cdot UREFO}{2^{21}}$
Недифференциальное включение, 16 бит. режим	$\frac{FxV DAT \cdot UREFO}{2^{13}}$ ,	$\frac{FxIDAT \cdot UREFO}{2^{13}}$

Каждый из каналов мощности имеет независимую калибровку смещения (16 бит), а также усиления (12 бит). Перед корректировкой смещение сдвигается на 8 бит вправо, что уменьшает шаг корректировки в 256 раз. Калибровка осуществляется в соответствии с формулой

$$P_{cor} = \left( P + \frac{P_{os}}{2^{23}} \right) \cdot \left( 1 + \frac{P_{gain}}{2^{11}} \right) \text{ (для микросхем до ревизии 2);} \quad (10)$$

$$P_{cor} = \left( P + \frac{P_{os}}{2^{23}} \right) \cdot \left( 1 + \frac{P_{gain}}{2^{15}} \right) \text{ (для микросхем с ревизии 2).}$$

Вычисленная мощность после калибровки накапливается в регистре аккумулятора. Для каждой из трех мощностей есть свой аккумулятор. Значение в них определяет потребленную энергию. В таблице 226 приведена формула перевода значения в Вт·с.

Таблица 226 – Формула перевода значения в Вт·с

Режим	Энергия, Вт·с	
Дифференциальное включение	$\frac{FxWATTHRP}{512 \cdot 4000}$	$\frac{FxWATTHRN}{512 \cdot 4000}$
Недифференциальное включение	$\frac{FxWATTHRP}{512 \cdot 1000}$	$\frac{FxWATTHRN}{512 \cdot 1000}$
<p>Примечание – Значение 4000 в формуле для дифференциального включения соответствует частоте дискретизации блока <math>\Delta\Sigma</math>АЦП, которая задается полем OSR_CONF регистра ADCUI_CTRL1. В недифференциальном включении в знаменателе приводится значение 1000, то есть мощностные характеристики отличаются в четыре раза</p>		

### 14.1 Типовая схема включения для учета электроэнергии по трем фазам

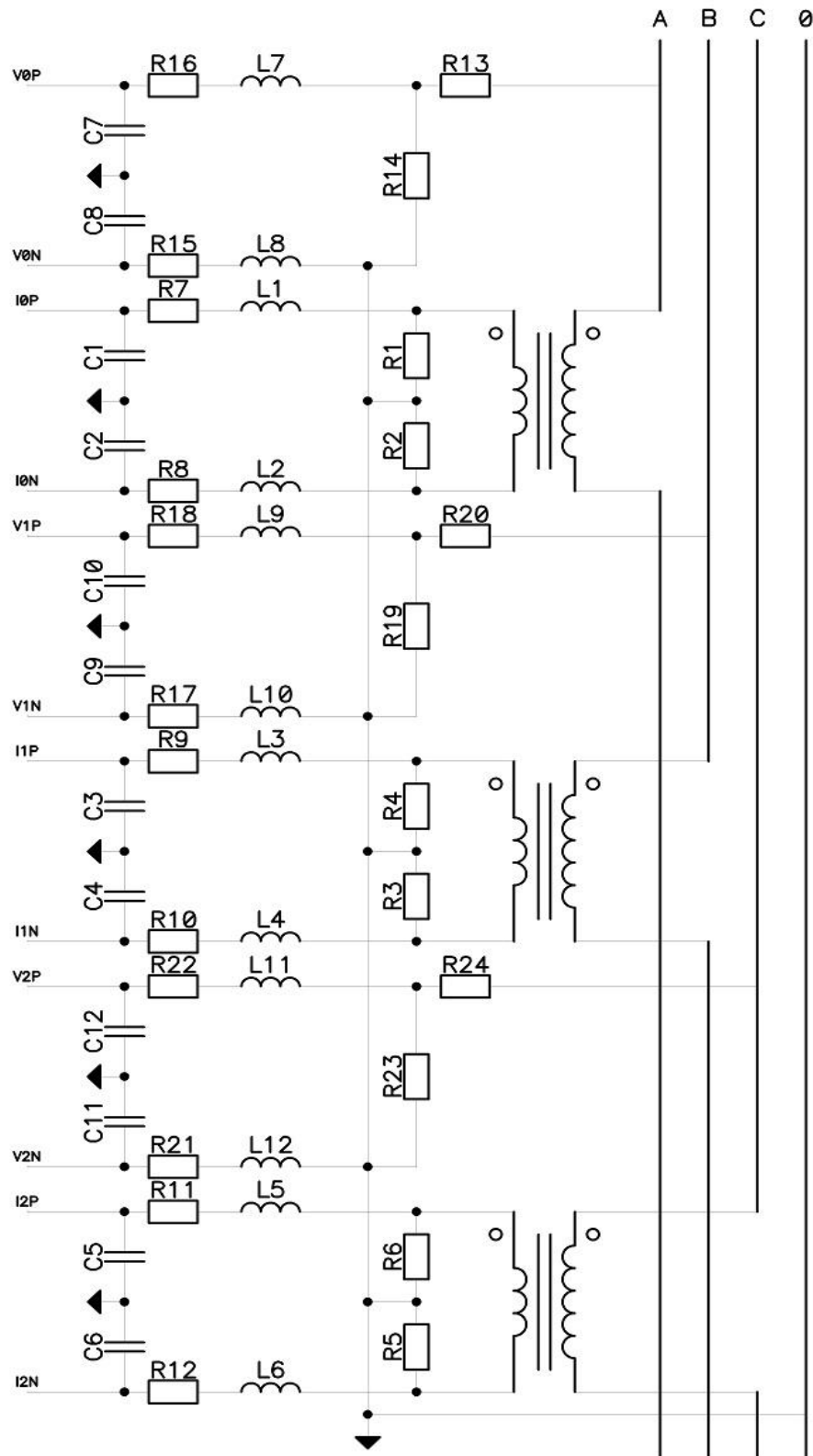


Рисунок 25 – Типовая схема включения для учета электроэнергии по трем фазам

Данная схема включения может быть использована как пример для конструирования трехфазных счетчиков электроэнергии. По каждой из трех фаз установлен трансформатор тока. Выход трансформаторов нагружен на резисторный

делитель. Сумма сопротивлений этих резисторов, например,  $R1+R2$ , должна соответствовать требуемой нагрузке выхода трансформатора. Оба резистора имеют одинаковое сопротивление и создают среднюю точку, относительно которой измеряется ток. В данной схеме включения средней точкой является аналоговая земля. С этой средней точкой соединена нейтраль трехфазной сети, если она используется. Для подачи сигнала на канал напряжения используется резисторный делитель. При выборе трансформатора и расчете резисторного делителя, стоит помнить, что амплитуда сигнала на входе АЦП не должна выходить за рамки  $\pm 500$  мВ относительно аналоговой земли микросхемы. Непосредственно перед входом каналов АЦП должен быть установлен фильтр для устранения эффектов наложения спектров, рассчитанный на частоту срезу примерно в полтора раза большую, чем частота дискретизации АЦП. В данном примере это простой RC-фильтр низких частот первого порядка. Перед RC-фильтром необходимо также установить индуктивности, фильтрующие радиочастотные помехи. Эти индуктивности не относятся функционально к фильтру для устранения эффектов наложения спектров и выбираются для наиболее широкополосного подавления радиочастот. Вы можете использовать свои варианты фильтров в зависимости от требований к конечному изделию.

### 14.2 Типовая схема включения для учета электроэнергии по одной фазе

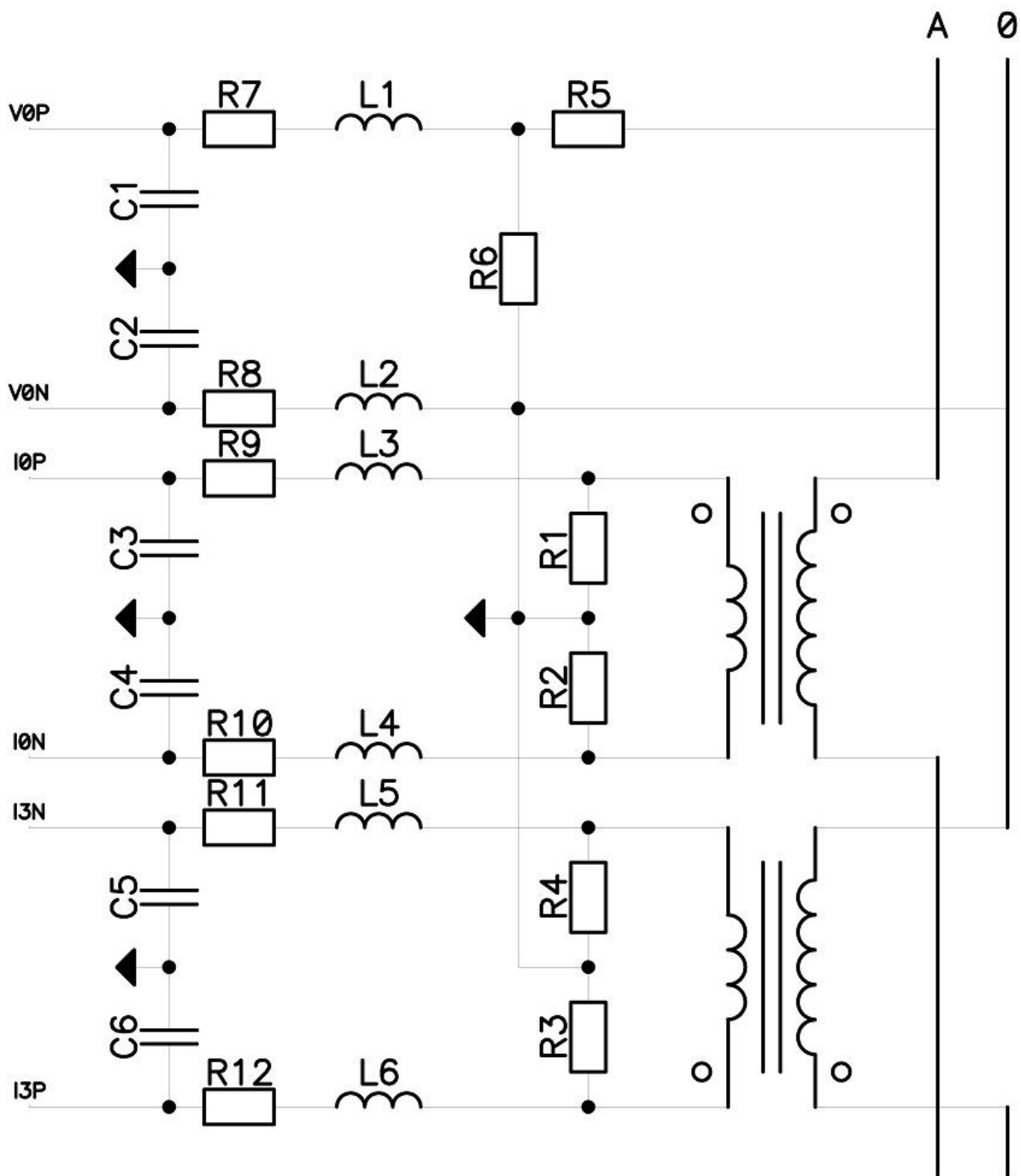


Рисунок 26 – Типовая схема включения для учета электроэнергии по одной фазе

В случае, когда необходимо построить однофазный счетчик электроэнергии на базе данной микросхемы, можно взять за основу схему, приведенную выше. В данной схеме предусмотрено измерение тока в обоих проводах однофазной сети. Микросхема позволяет в автоматическом режиме учитывать то значение тока из каналов I0 и I3, которое будет больше. Если учета тока по «нулю» не требуется, то часть схемы, относящуюся к каналу I3 можно убрать. В остальном, назначение элементов данной схемы аналогично схеме для учета электроэнергии по трем фазам.

14.3 Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта

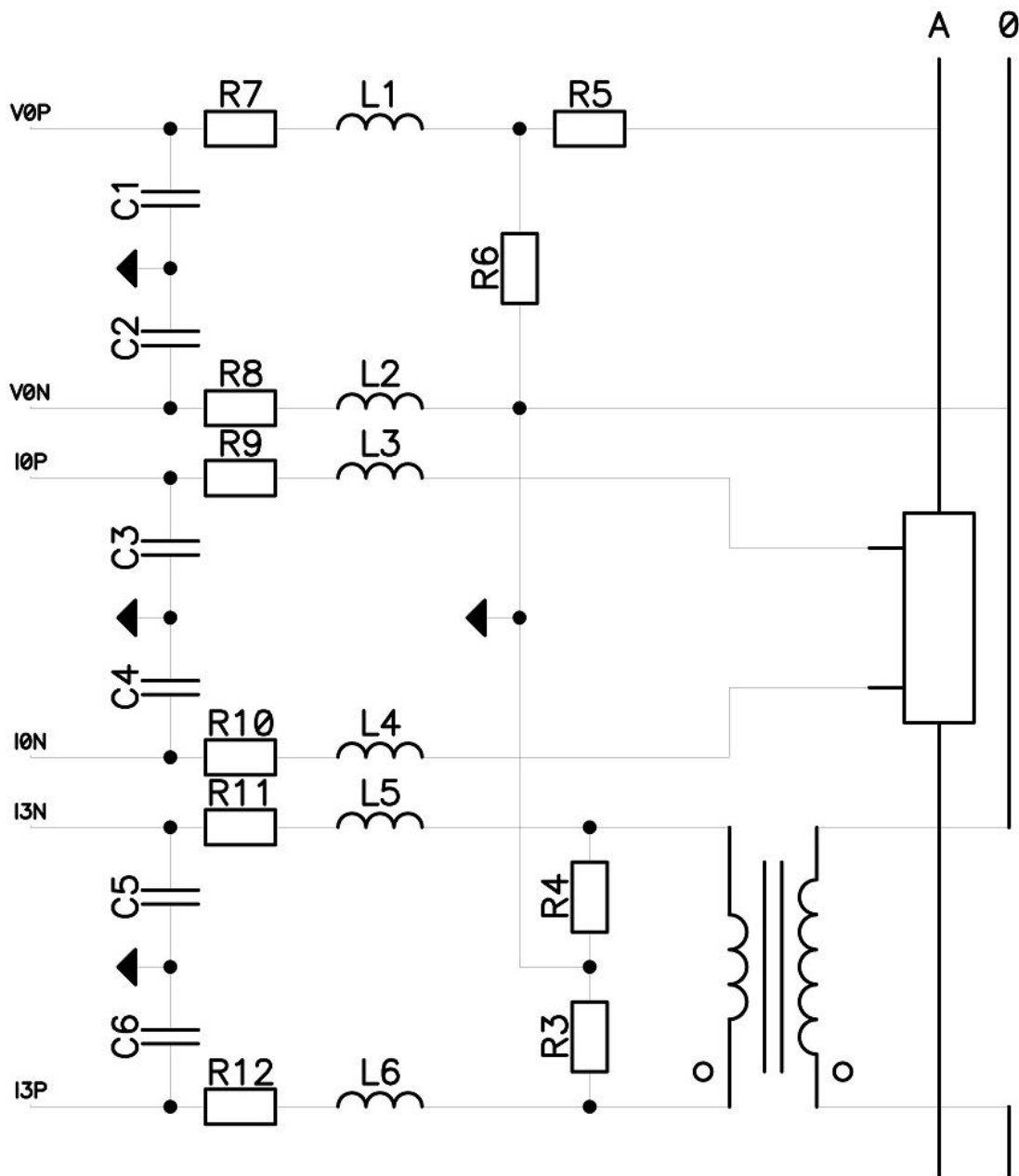


Рисунок 27 – Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта

Также для измерения тока может быть использован шунт. В этом случае следует использовать схему на рисунке выше. Аналогично предыдущей схеме на трансформаторах, здесь можно исключить часть схемы, относящуюся к каналу тока I3, если контроля тока в нулевом проводе не требуется.

## 15 Аппаратный блок вычисления CRC

Микросхема имеет в своем составе блок для вычисления 16-битного CRC с произвольным полиномом. Контроллер принимает 32-битные слова и может их обрабатывать как в прямом порядке (начиная с младшего бита), так и в обратном (начиная со старшего бита). Скорость подсчета составляет два бита / PCLK (частота APB шины). Контроллер имеет FIFO на четыре отчета, а также DMA канал для загрузки новых слов. Запрос для DMA формируется, если в FIFO пусто. Контроллер начинает обрабатывать новые слова, как только они появляются в FIFO и обрабатывает до последнего слова. После обработки каждого слова выставляется флаг. Регистр CRC имеет доступ как на чтение (считать рассчитанное значение), так и на запись (установить начальное значение).

### 15.1 Описание регистров управления блока CRC

Таблица 227 – Описание регистров управления блока CRC

Базовый Адрес	Название	Описание
0x4009_8000	CRC	Контроллер CRC
Смещение		
0x00	CRC_CTRL	Общее управление для контроллера CRC
0x04	CRC_STAT	Статус CRC блока
0x08	CRC_DATAI	Регистр FIFO входных данных
0x0C	CRC_VAL	Регистр подсчитанного CRC
0x10	CRC_POL	Полином для расчета CRC

#### 15.1.1 CRC\_CTRL

Таблица 228 – Регистр CRC\_CTRL

Номер	31...7	6, 5	4, 3	2	1	0
Доступ		R/W	R/W	R/W	R/W	R/W
Сброс		00	00	0	0	0
	-	DCSize	DLSize	DMAEN	DATAINV	CRCEN

Таблица 229 – Описание бит регистра CRC\_CTRL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...7	-	Зарезервировано
6, 5	DCSize	Размер данных при расчете CRC: 00 – вычисление для байта (8 бит), при этом DLSize может быть 00, 01, 10; 01 – вычисление для полуслов (16 бит), при этом DLSize может быть 01, 10; 10 – вычисление для слов (32 бит), при этом DLSize может быть только 10



Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
4, 3	DLSize	Размер загружаемых данных: 00 – байт (8 бит), при этом загружаемый байт записывается в CRC_DATAI[7:0]; 01 – полуслово (16 бит), при этом загружаемое полуслово записывается в CRC_DATAI[15:0]; 10 – слово (32 бита), при этом загружаемое слово записывается в CRC_DATAI[31:0]
2	DMAEN	Разрешение формирования запроса для DMA: 0 – запрос не формируется; 1 – запрос формируется
1	DATAINV	Порядок вычисления CRC: 0 – начиная с младшего разряда; 1 – начиная со старшего разряда
0	CRCEN	Разрешение работы блока: 0 – блок выключен; 1 – блок включен

### 15.1.2 CRC\_STAT

Таблица 230 – Регистр CRC\_STAT

Номер	31:4	3	2	1	0
Доступ		R/W	R	R	R
Сброс		0	0	0	0
	-	FIFOOVER	FIFOEMPTY	FIFOFULL	CONVCOMP

Таблица 231 – Описание бит регистра CRC\_STAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...4	-	Зарезервировано
3	FIFOOVER*	Переполнение FIFO: 0 – корректная работа; 1 – была запись в полное FIFO, что привело к потере данных
2	FIFOEMPTY	FIFO пусто: 0 – FIFO имеет по крайней мере одну заполненную ячейку; 1 – FIFO пусто
1	FIFOFULL	FIFO заполнено: 0 – FIFO имеет по крайней мере одну свободную ячейку; 1 – FIFO не имеет свободных ячеек
0	CONVCOMP	Завершение расчета CRC: 0 – расчет идет; 1 – расчет слова данных завершен или блок отключен
* Сброс бита происходит записью «1» в разряд [3]		

### 15.1.3 CRC\_DATAI

Таблица 232 – Регистр CRC\_DATAI

Номер	31...0
Доступ	W
Сброс	
	DATA_IN

Таблица 233 – Описание бит регистра CRC\_DATAI

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	DATA_IN	Регистр для записи нового отчета в FIFO

### 15.1.4 CRC\_VAL

Таблица 234 – Регистр CRC\_VAL

Номер	31...16	15...0
Доступ		R/W
Сброс		0000000000000000
	-	CRCOUT

Таблица 235 – Описание бит регистра CRC\_VAL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...0	CRCOUT	Рассчитанное значение / начальное значение. Начальное значение нужно записывать, когда блок отключен или, когда закончено преобразование

### 15.1.5 CRC\_POL

Таблица 236 – Регистр CRC\_POL

Номер	31...17	16...0
Доступ		R/W
Сброс		1000000000000001
	-	CRCPOL

Таблица 237 – Описание бит регистра CRC\_POL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...0	CRC_POL	Полином для расчета CRC. Так как это 16-битное CRC, то младший и старший биты всегда «1» и их нельзя изменить

Ниже приведен результирующий полином:

$$f(x) = x^{16} + x^{CRC\_POL[15]} + x^{CRC\_POL[14]} + \dots + x^{CRC\_POL[2]} + x^{CRC\_POL[1]} + 1.$$

## 16 Сигналы тактовой частоты

Микросхема имеет два встроенных генератора, два внешних осциллятора, а также специализированный блок формирования тактовой синхронизации микросхемы.

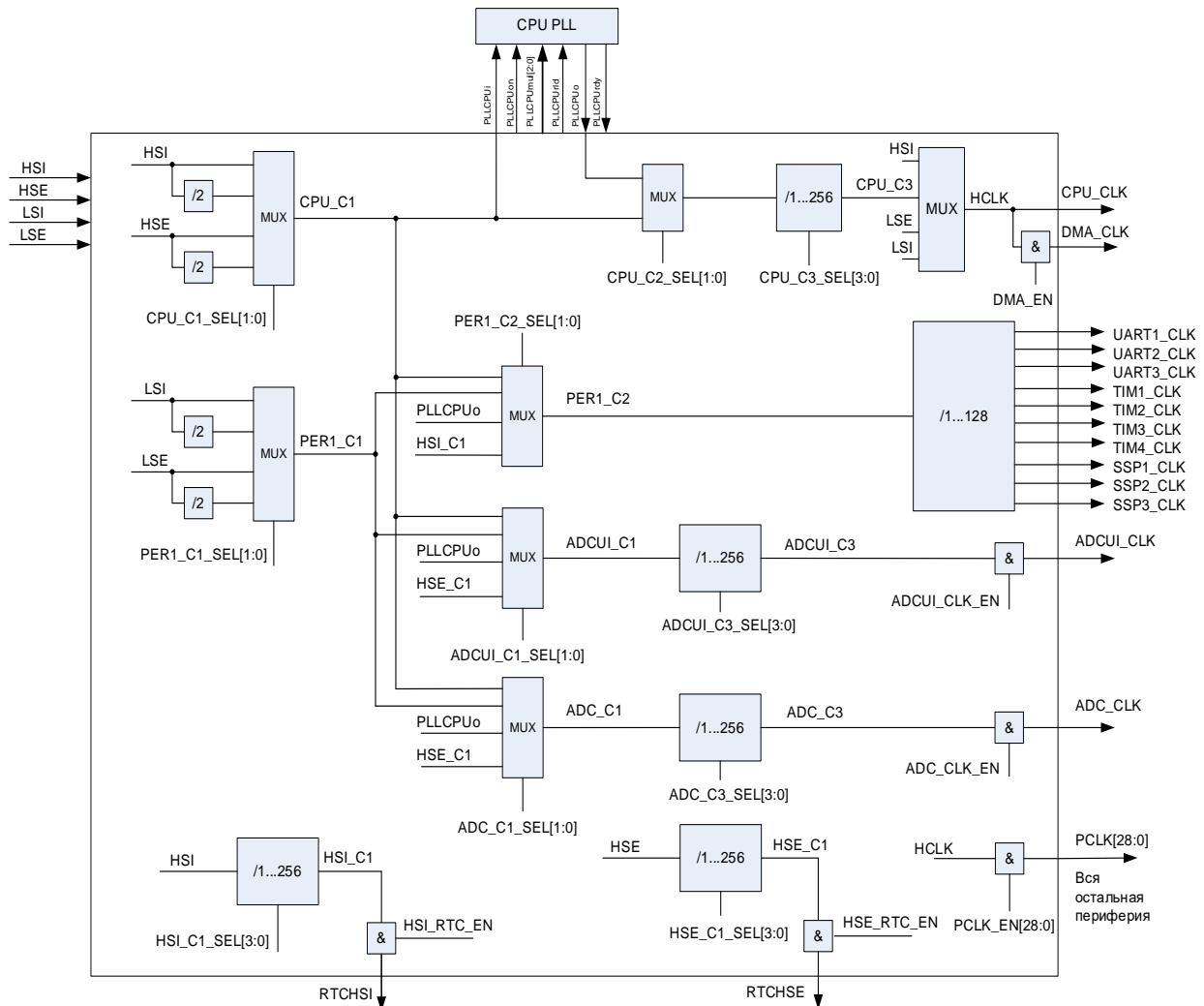


Рисунок 28 – Структурная блок-схема формирования тактовой частоты

Управление тактовыми частотами ведется через периферийный блок RST\_CLK. При включении питания микросхема запускается на частоте генератора HSI. 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 PER2\_CLOCK. Некоторые контроллеры интерфейсов (UART, SSP, Таймеры) могут работать на частотах отличных от частоты процессорного ядра, поэтому в соответствующих регистрах (PER1\_CLOCK, UART\_CLOCK, SSP\_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 мультиплексах управляемом регистре CPU\_CLOCK. Регистры блока контроллера тактовой частоты внутренним сигналом сброса ndmreset, сформированным отладчиком, не сбрасываются.

### 16.1 Встроенный RC-Генератор HSI

Генератор HSI вырабатывает тактовую частоту 8 МГц. Генератор автоматически запускается при появлении питания  $U_{CC}$  и при выходе в нормальный режим работы вырабатывает сигнал HSI\_RDY в регистре батарейного домена BKP\_CLK. Если напряжение питания  $U_{CC}$  находится в диапазоне 2,2 В-2,7 В и HSI\_LSI\_TRIM[5:0]=0, то частота генератора HSI может составлять несколько кГц, что может привести к долгому старту микросхемы, так как первоначально процессорное ядро запускается на тактовой частоте HSI. При дальнейшей работе генератор HSI может быть отключен при помощи сигнала HSI\_ON в регистре BKP\_CLK. Также генератор может быть подстроен при помощи сигнала HSI\_LSI\_TRIM в регистре BKP\_CLK.\*

### 16.2 Встроенный RC-генератор LSI

Генератор LSI вырабатывает тактовую частоту 32 кГц. Генератор включается сигналом LSI\_ON в регистре BKP\_CLK и при выходе в нормальный режим работы вырабатывает сигнал LSI\_RDY в регистре BKP\_CLK. Также генератор может быть подстроен при помощи сигнала HSI\_LSI\_TRIM в регистре BKP\_CLK. Частота генератора LSI равна HSI/256.

### 16.3 Внешний осциллятор HSE

Осциллятор HSE предназначен для выработки тактовой частоты  $f_{C\_HSE}$  в диапазоне 8...16 МГц с помощью внешнего резонатора. Рекомендуемая суммарная нагрузочная емкость на выводе микросхемы (паразитная емкость платы + номинал припаиваемой емкости) – 24 и 16 пФ для резонаторов на 8 и 16 МГц соответственно. Осциллятор запускается при появлении питания  $U_{CC}$  и сигнала разрешения HSEON в регистре HS\_CONTROL. При выходе в нормальный режим работы вырабатывается 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. Допустимый диапазон частот  $f_{C\_HSE}$  в режиме HSEBYP – от 1 до 60 МГц.

### 16.4 Внешний осциллятор LSE

Осциллятор LSE предназначен для выработки тактовой частоты 32 кГц с помощью внешнего резонатора. Рекомендуемая суммарная нагрузочная емкость на выводе микросхемы (паразитная емкость платы + номинал припаиваемой емкости) - от 8 до 12 пФ. Осциллятор запускается при появлении питания  $U_{CCV}$  и сигнала разрешения LSE\_ON в регистре BKP\_CLK. При выходе в нормальный режим работы вырабатывает сигнал LSE\_RDY в регистре BKP\_CLK.

Также осциллятор может работать в режиме LSE\_BYP, когда входная тактовая частота с входа OSC\_IN32 проходит напрямую на выход LSE. Допустимый диапазон

\* См. K1986BK025 Errata Notice – ошибка 0008.

частот в режиме LSE\_BYP – от 20 до 40 кГц. Выход OSC\_OUT32 находится в этом режиме в третьем состоянии.

Так как генератор LSE питается от напряжения питания U<sub>ССВ</sub> и его регистр управления BKP\_CLK расположен в батарейном домене, то генератор может продолжать работать при пропадании основного питания U<sub>СС</sub>. Генератор LSE используется для работы часов реального времени и календаря.

### 16.5 Встроенный блок умножения системной тактовой частоты

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 8, задаваемых на входе PLLCPUMUL[2:0] в регистре PLL\_CONTROL. Входная частота блока умножителя должна быть в диапазоне 8...16 МГц выходная до 60 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 может быть использована как основная частота процессора и периферии.

### 16.6 Описание регистров блока контроллера тактовой частоты

Таблица 238 – 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	PER1_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x14	ADC_CLOCK	Регистр управления тактовой частотой АЦП и ΔΣАЦП
0x18	RTC_CLOCK	Регистр управления формированием высокочастотных тактовых сигналов блока RTC
0x1C	PER2_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x20	DMA_DONE_STICK	Регистр фиксации сигналов окончания работы DMA (доступно с ревизии 2).
0x24	TIM_CLOCK	Регистр управления тактовой частотой TIMER
0x28	UART_CLOCK	Регистр управления тактовой частотой UART
0x2C	SSP_CLOCK	Регистр управления тактовой частотой SSP
0x30	DIV_SYS_TIM	Регистр выбора делителя частоты для системного таймера ядра (доступно с ревизии 2).

### 16.6.1 CLOCK\_STATUS

Таблица 239 – Регистр CLOCK\_STATUS

Номер	31...3	2	1	0
Доступ	U	RO	RO	U
Сброс	0	0	0	0
	-	HSE RDY	PLL CPU RDY	

Таблица 240 – Описание бит регистра CLOCK\_STATUS

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...3	-	Зарезервировано
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	-	Зарезервировано

### 16.6.2 PLL\_CONTROL

Таблица 241 – Регистр PLL\_CONTROL

Номер	31...11	10...8	7...4	3	2	1, 0
Доступ	U	R/W	U	R/W	R/W	U
Сброс	0	000	0000	0	0	0
	-	PLL CPU MUL[2:0]	-	PLL CPU SEL	PLL CPU ON	-

Таблица 242 – Описание бит регистра PLL\_CONTROL

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...11	-	Зарезервировано
10...8	PLL CPU MUL[2:0]	Коэффициент умножения для CPU PLL: $PLL_{CPU0} = PLL_{CPUi} \cdot (PLL_{CPUMUL} + 1)$
7...4	-	Зарезервировано
3	PLL CPU SEL	Бит перезапуска выбора входной частоты PLL. 0 – частота с генератора HSE 1 – частота задается с помощью бит CPU_C1_SEL[1:0] регистра CPU_CLOCK.

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
2	PLL CPU ON	Бит включения PLL: 0 – PLL выключена; 1 – PLL включена
1, 0	-	Зарезервировано

### 16.6.3 HS\_CONTROL

Таблица 243 – Регистр HS\_CONTROL

Номер	31...2	1	0
Доступ	U	R/W	R/W
Сброс		0	0
	-	HSEBYP	HSEON

Таблица 244 – Описание битов регистра HS\_CONTROL

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	-	Зарезервировано
1	HSEBYP	Бит управления HSE осциллятором: 0 – режим осциллятора; 1 – режим внешнего генератора
0	HSEON	Бит управления HSE осциллятором: 0 – выключен; 1 – включен

### 16.6.4 CPU\_CLOCK

Таблица 245 – Регистр CPU\_CLOCK

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

Таблица 246 – Описание бит регистра CPU\_CLOCK

Биты	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

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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. Увеличение коэффициента деления можно осуществлять с любым шагом. Уменьшение коэффициента деления возможно только с шагом 1 и с временным интервалом не менее T (где T – длительность двух периодов выходной частоты делителя до изменения коэффициента деления). Иначе возможно прекращение формирования тактового сигнала
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

### 16.6.5 PER1\_CLOCK

Таблица 247 – Регистр PER1\_CLOCK

Номер	31..14	13	12...10	9, 8
Доступ	U	R/W	R/W	R/W
Сброс	0	0	000	00
	-	WDG CLK EN	WDG BRG [2:0]	WDG C2 SEL[1:0]

Номер	7, 6	5	4	3, 2	1, 0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	00	1	0	00	00
	WDG C1 SEL[1:0]	DMA_EN	KeyResetProg	PER1 C2 SEL	PER1 C1 SEL[1:0]



Таблица 248 – Описание бит регистра PER1\_CLOCK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14	-	Зарезервировано
13	WDG CLK EN	Разрешение тактовой частоты на WDG: 0 – нет частоты; 1 – есть частота
12...10	WDG BRG [2:0]	Делитель тактовой частоты WDG: 000 – WDG_CLK == WDG_C2; 001 – WDG_CLK == WDG_C2/2; 010 – WDG_CLK == WDG_C2/4; ... 111 – WDG_CLK == WDG_C2/128
9, 8	WDG C2 SEL	Биты выбора источника для WDG_C2: 00 – CPU_C1; 01 – WDG_C1; 10 – PLLCPUo; 11 – HSI_CLK
7, 6	WDG C1 SEL[1:0]	Биты выбора источника для WDG_C1: 00 – LSI; 01 – LSI/2; 10 – LSE; 11 – LSE/2
5	DMA_EN	Бит разрешения тактирования DMA контроллера
4	KeyResetProg	Бит сброса памяти криптографических ключей
3, 2	PER1 C2 SEL	Биты выбора источника для PER1_C2: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_CLK
1, 0	PER1 C1 SEL[1:0]	Биты выбора источника для PER1_C1: 00 – LSI; 01 – LSI/2; 10 – LSE; 11 – LSE/2

### 16.6.6 ADC\_CLOCK

Таблица 249 – Регистр ADC\_CLOCK

Номер	31...14	13	12	11...8	7...4	3, 2	1, 0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0000	0000	00	00
	-	ADC CLK EN	ADCUICLK EN	ADC C3 SEL[3:0]	ADCUI C3 SEL[3:0]	ADCUI C1 SEL[1:0]	ADC C1 SEL[1:0]

Таблица 250 – Описание бит регистра ADC\_CLOCK

Биты	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	ADCUI CLK EN	Бит разрешения выдачи тактовой частоты ADCUI CLK: 0 – запрещен; 1 – разрешен
11...8	ADC C3 SEL[3:0]	Биты выбора делителя для ADC_C3: 0xxx – ADC_C3 = ADC_C1; 1000 – ADC_C3 = ADC_C1 / 2; 1001 – ADC_C3 = ADC_C1 / 4; 1010 – ADC_C3 = ADC_C1 / 8; ... 1111 – ADC_C3 = ADC_C1 / 256 Увеличение коэффициента деления можно осуществлять с любым шагом. Уменьшение коэффициента деления возможно только с шагом 1 и с временным интервалом не менее T (где T – длительность двух периодов выходной частоты делителя до изменения коэффициента деления). Иначе возможно прекращение формирования тактового сигнала
7...4	ADCUI C3 SEL[3:0]	Биты выбора делителя для ADCUI_C3: 0xxx – ADCUI_C3 = ADCUI_C1; 1000 – ADCUI_C3 = ADCUI_C1 / 2; 1001 – ADCUI_C3 = ADCUI_C1 / 4; 1010 – ADCUI_C3 = ADCUI_C1 / 8; ... 1111 – ADCUI_C3 = ADCUI_C1 / 256 Увеличение коэффициента деления можно осуществлять с любым шагом. Уменьшение коэффициента деления возможно только с шагом 1 и с временным интервалом не менее T (где T – длительность двух периодов выходной частоты делителя до изменения коэффициента деления). Иначе возможно прекращение формирования тактового сигнала
3, 2	ADCUI C1 SEL[1:0]	Биты выбора источника для ADCUI_C1: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSE_CLK В качестве частоты синхронизации блока ADCUI необходимо всегда выбирать источник HSE = 8,192 МГц без дополнительных делителей

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1, 0	ADC C1 SEL[1:0]	Биты выбора источника для ADC_C1: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSE_CLK

### 16.6.7 RTC\_CLOCK

Таблица 251 – Регистр RTC\_CLOCK

Номер	31...10	9	8	7...4	3...0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0000	0000
	-	HSI RTC EN	HSE RTC EN	HSI_C1 SEL[3:0]	HSE_C1 SEL[3:0]

Таблица 252 – Описание бит регистра RTC\_CLOCK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	-	Зарезервировано
9	HSI RTC EN	Бит разрешения выдачи тактовой частоты HSI_RTC: 0 – запрещен; 1 – разрешен
8	HSE RTC EN	Бит разрешения выдачи тактовой частоты HSE_RTC: 0 – запрещен; 1 – разрешен
7...4	HSI_C1 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 Увеличение коэффициента деления можно осуществлять с любым шагом. Уменьшение коэффициента деления возможно только с шагом 1 и с временным интервалом не менее T (где T – длительность двух периодов выходной частоты делителя до изменения коэффициента деления). Иначе возможно прекращение формирования тактового сигнала

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3...0	HSE_C1 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 Увеличение коэффициента деления можно осуществлять с любым шагом. Уменьшение коэффициента деления возможно только с шагом 1 и с временным интервалом не менее T (где T – длительность двух периодов выходной частоты делителя до изменения коэффициента деления). Иначе возможно прекращение формирования тактового сигнала

### 16.6.8 PER2\_CLOCK

Таблица 253 – Регистр PER2\_CLOCK

Номер	30...0
Доступ	R/W
Сброс	0x00000010
	PCLK_EN[30:0]

Таблица 254 – Описание бит регистра PER2\_CLOCK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
30...0	PCLK EN[29:0]	Биты разрешения тактирования периферийных блоков: 0 – запрещено; 1 – разрешено. PCLK[0] – SSP1; PCLK[1] – UART1; PCLK[2] – UART2; PCLK[3] – FLASH; PCLK[4] – RST_CLK; PCLK[5] – DMA; PCLK[6] – I2C; PCLK[7] – UART3; PCLK[8] – ADC; PCLK[9] – WWDG; PCLK[10] – IWDG; PCLK[11] – POWER; PCLK[12] – BKP; PCLK[13] – ADCUI;

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		PCLK[14] – TIMER1; PCLK[15] – TIMER2; PCLK[16] – PORTA; PCLK[17] – PORTB; PCLK[18] – PORTC; PCLK[19] – CRC; PCLK[20] – зарезервировано; PCLK[21] – SENSORS; PCLK[22] – CLK_MEASURE; PCLK[23] – RANDOM; PCLK[24] – ISO7816; PCLK[25] – SSP2; PCLK[26] – SSP3; PCLK[27] – TIMER3; PCLK[28] – TIMER4; PCLK[29] – UART4; PCLK[30] – PORTD

### 16.6.9 DMA\_DONE\_STICK (доступно с ревизии 2)

Таблица 255 – Регистр DMA\_DONE\_STICK

Номер	31..0
Доступ	R/C
Сброс	0
	DMA_DONE_STICK

Таблица 256 – Описание бит регистра DMA\_DONE\_STICK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	DMA_DONE_STICK	Биты регистра устанавливаются в единицу при окончании работы соответствующего номера 0 – 31 канала DMA (dma_done[C]). Биты регистра сбрасываются только записью нуля

### 16.6.10 TIM\_CLOCK

Таблица 257 – Регистр TIM\_CLOCK

Номер	27	26	25	24	23...19	18...16	15...8	7...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	00000	000	00000000	00000000
	TIM4 CLK EN	TIM3 CLK EN	TIM2 CLK EN	TIM1 CLK EN	TIM4 BRG [4:0]	TIM3 BRG [2:0]	TIM2 BRG [7:0]	TIM1 BRG [7:0]

Таблица 258 – Описание бит регистра TIM\_CLOCK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27	TIM4 CLK EN	Разрешение тактовой частоты на TIM4: 0 – нет частоты; 1 – есть частота
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...19	TIM4 BRG [4:0]	Делитель тактовой частоты TIM4: xx000 – TIM4_CLK == PER1_C2; xx001 – TIM4_CLK == PER1_C2/2; xx010 – TIM4_CLK == PER1_C2/4; ... xx111 – TIM4_CLK == PER1_C2/128
18...16	TIM3 BRG [2:0]	Делитель тактовой частоты TIM3: 000 – TIM3_CLK == PER1_C2; 001 – TIM3_CLK == PER1_C2/2; 010 – TIM3_CLK == PER1_C2/4; ... 111 – TIM3_CLK == PER1_C2/128
15...8	TIM2 BRG [7:0]	Делитель тактовой частоты TIM2: xxxxx000 – TIM2_CLK == PER1_C2; xxxxx001 – TIM2_CLK == PER1_C2/2; xxxxx010 – TIM2_CLK == PER1_C2/4; ... xxxxx111 – TIM2_CLK == PER1_C2/128
7...0	TIM1 BRG [7:0]	Делитель тактовой частоты TIM1: xxxxx000 – TIM1_CLK == PER1_C2; xxxxx001 – TIM1_CLK == PER1_C2/2; xxxxx010 – TIM1_CLK == PER1_C2/4; ... xxxxx111 – TIM1_CLK == PER1_C2/128
Примечание – Частота тактирования таймера TIMx_CLK должна не больше частоты тактирования ядра CPU_CLK		

16.6.11 UART\_CLOCK

Таблица 259 – Регистр UART\_CLOCK

Номер	30	29..27	26	25	24	23...16	15...8	7...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	000	0	0	0	00000000	00000000	00000000
	UART4 CLK EN	UART4BRG [2:0]	UART3 CLK EN	UART2 CLK EN	UART1 CLK EN	UART3 BRG [7:0]	UART2 BRG [7:0]	UART1 BRG [7:0]

Таблица 260 – Описание бит регистра UART\_CLOCK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	-	Зарезервировано
30	UART4 CLK EN	Разрешение тактовой частоты на UART4: 0 – нет частоты; 1 – есть частота
29...27	UART4 BRG [2:0]	Делитель тактовой частоты UART 3: 000 – UART 4_CLK == PER1_C2; 001 – UART 4_CLK == PER1_C2/2; 010 – UART 4_CLK == PER1_C2/4; ... 111 – UART 4_CLK == PER1_C2/128
26	UART3 CLK EN	Разрешение тактовой частоты на UART3: 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	UART3 BRG [7:0]	Делитель тактовой частоты UART 3: xxxxx000 – UART 3_CLK == PER1_C2; xxxxx001 – UART 3_CLK == PER1_C2/2; xxxxx010 – UART 3_CLK == PER1_C2/4; ... xxxxx111 – UART 3_CLK == PER1_C2/128
15...8	UART2 BRG [7:0]	Делитель тактовой частоты UART 2: xxxxx000 – UART 2_CLK == PER1_C2; xxxxx001 – UART 2_CLK == PER1_C2/2; xxxxx010 – UART 2_CLK == PER1_C2/4; ... xxxxx111 – UART 2_CLK == PER1_C2/128

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7...0	UART1 BRG [7:0]	Делитель тактовой частоты UART: xxxxx000 – UART 1_CLK == PER1_C2; xxxxx001 – UART 1_CLK == PER1_C2/2; xxxxx010 – UART 1_CLK == PER1_C2/4; ... xxxxx111 – UART 1_CLK == PER1_C2/128

### 16.6.12 SSP\_CLOCK

Таблица 261 – Регистр SSP\_CLOCK

Номер	26	25	24	23...16	15...8	7...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	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]

Таблица 262 – Описание бит регистра SSP\_CLOCK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	SSP3 CLK EN	Разрешение тактовой частоты на SSP3: 0 – нет частоты; 1 – есть частота
25	SSP2 CLK EN	Разрешение тактовой частоты на SSP2: 0 – нет частоты; 1 – есть частота
24	SSP1 CLK EN	Разрешение тактовой частоты на SSP1: 0 – нет частоты; 1 – есть частота
23...16	SSP3 BRG [7:0]	Делитель тактовой частоты SSP3: xxxxx000 – SSP 3_CLK == PER1_C2; xxxxx001 – SSP 3_CLK == PER1_C2/2; xxxxx010 – SSP 3_CLK == PER1_C2/4; ... xxxxx111 – SSP 3_CLK == PER1_C2/128
15...8	SSP2 BRG [7:0]	Делитель тактовой частоты SSP2: xxxxx000 – SSP 2_CLK == PER1_C2; xxxxx001 – SSP 2_CLK == PER1_C2/2; xxxxx010 – SSP 2_CLK == PER1_C2/4; ... xxxxx111 – SSP 2_CLK == PER1_C2/128



Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7...0	SSP1 BRG [7:0]	Делитель тактовой частоты SSP1: xxxxx000 – SSP 1_CLK == PER1_C2; xxxxx001 – SSP 1_CLK == PER1_C2/2; xxxxx010 – SSP 1_CLK == PER1_C2/4; ... xxxxx111 – SSP 1_CLK == PER1_C2/128

### 16.6.13 DIV\_SYS\_TIM (доступно с ревизии 2)

Таблица 263 – Регистр DIV\_SYS\_TIM

Номер	7..0
Доступ	R/W
Сброс	0
DIV_SYS_TIM	

Таблица 264 – Описание бит регистра DIV\_SYS\_TIM

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..8	-	Зарезервировано
7...0	DIV_SYS_TIM	Делитель тактовой частоты для системного таймера ядра: $CLK\_TIMER\_PULSE = HCLK / (DIV\_SYS\_TIM + 2)$

## 17 Батарейный домен и часы реального времени

Блок батарейного домена предназначен для обеспечения функций календаря и часов реального времени, сохранения некоторого набора пользовательских данных при отключении основного источника питания. Также в батарейном домене реализована функция контроля входов WAKEUP. Это позволяет, даже в отсутствии основного питания определять его состояние. При снижении питания  $U_{CC}$  регулятор напряжения батарейного домена выполняет автоматическое переключение питания батарейного домена с  $U_{CC}$  на  $U_{CCV}$ . Если на  $U_{CCV}$  имеется отдельный источник питания (батарейка), то батарейный домен остается включенным и может выполнять свои функции.

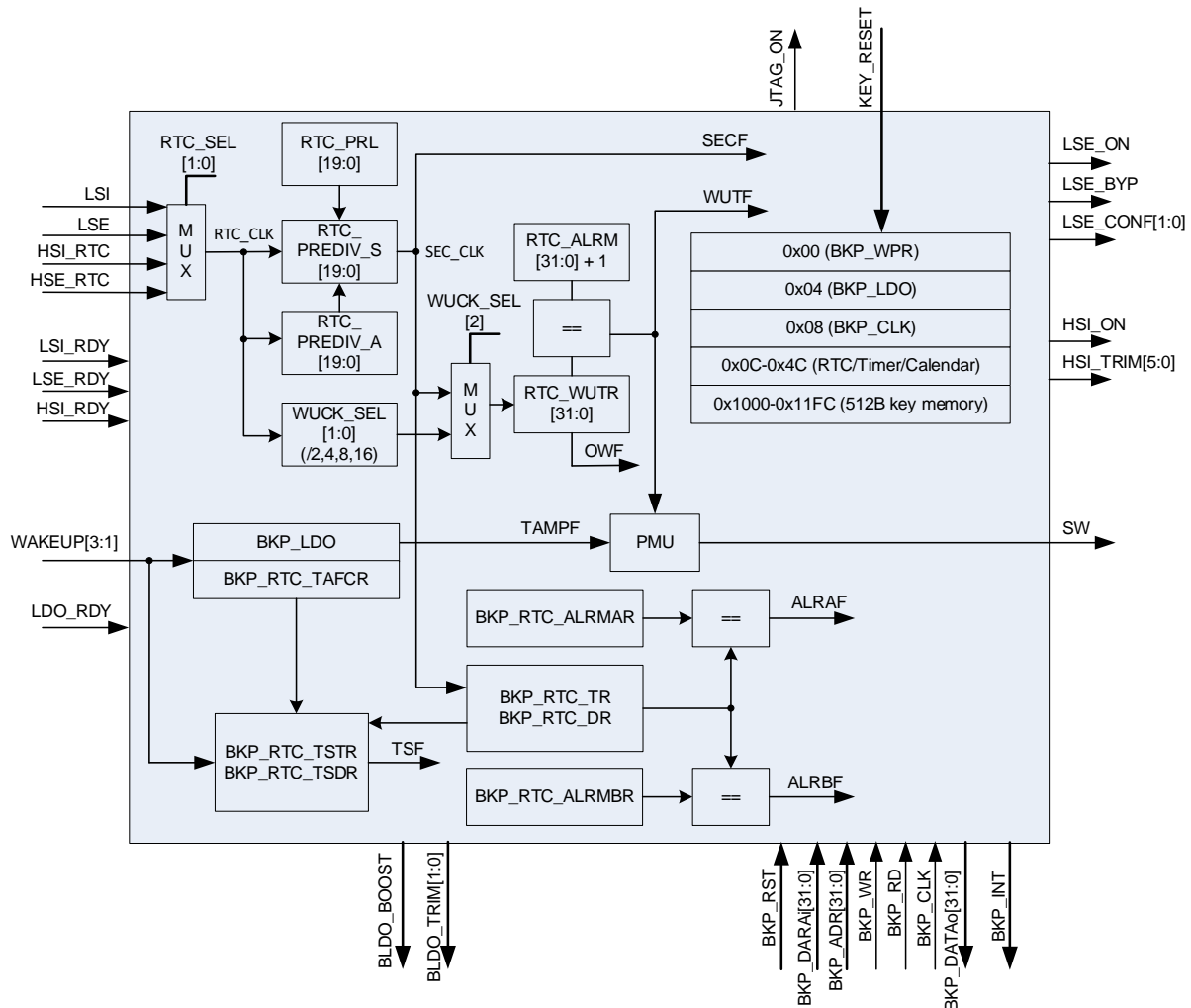


Рисунок 29 – Структурная блок-схема батарейного домена и часов реального времени

### 17.1 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  $RTC\_EN$ . В качестве источника тактовой частоты часов реального времени  $RTC\_CLK$  может выступать генератор LSI или LSE, а также генератор HSI (частота  $HSI\_RTC$ ) или HSE (частота  $HSE\_RTC$ ) с дополнительным делителем до 256. Частоты  $HSI\_RTC$  и  $HSE\_RTC$  формируются в блоке управления тактовыми частотами  $RST\_CLK$  и могут быть выбраны только при наличии питания

цифрового ядра, формируемого встроенным регулятором напряжения. Частота LSI может быть выбрана при наличии питания  $U_{CC}$ , LSE – при наличии  $U_{CCV}$ . Выбор между источниками осуществляется битами RTCSEL. При возможном отключении основного источника питания  $U_{CC}$  в качестве источника тактовой частоты RTC\_CLK должен использоваться осциллятор LSE, так как он также имеет питание  $U_{CCV}$ . 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 также не определены и могут принять любое значение.

Для отсчета секунд в часах реального времени применяется 20-битный предварительный делитель входной тактовой частоты RTC\_CLK, на выходе которого формируется тактовый сигнал SEC\_CLK. Регистр ВКР\_RTC\_PREDIV\_S выступает в качестве счетчика предварительного делителя, который тактируется на частоте RTC\_CLK. Регистр ВКР\_RTC\_PRL задает коэффициент деления предварительного делителя, при этом счетчик ВКР\_RTC\_PREDIV\_S инкрементируется в интервале от 0 до ВКР\_RTC\_PRL. Коэффициент деления в регистре ВКР\_RTC\_PRL должен быть задан таким образом, чтобы частота SEC\_CLK составляла 1 Гц.

Для калибровки тактовой частоты RTC\_CLK используются биты RTC\_CAL[7:0]. Значение RTC\_CAL[7:0] определяет, какое число тактов RTC\_CLK из  $2^{20}$  будет замаскировано с помощью дополнительного счетчика RTC\_PREDIV\_A. Таким образом, с помощью бит RTC\_CAL[7:0] производится замедление хода часов. Для убыстрения хода часов необходимо задать меньшее, чем требуется, значение регистра ВКР\_RTC\_PRL, а затем произвести замедление с помощью бит RTC\_CAL[7:0]. Изменение значения бит RTC\_CAL[7:0] может быть осуществлено в ходе работы часов реального времени.

Основу модуля часов реального времени составляет двоично-десятичный таймер/счетчик. Результат счета отображается в двух 32-разрядных регистрах. Первый из них ВКР\_RTC\_TR содержит информацию о времени, второй – ВКР\_RTC\_DR представляет собой календарь, включающий год, месяц, день недели и дату. Все данные представлены в формате BCD, что позволяет их сразу же использовать для отображения на различных индикаторах. Модуль календаря поддерживает автоматическое определение високосных лет, а также количества дней в текущем месяце. Запись в регистры ВКР\_RTC\_TR и ВКР\_RTC\_DR выполняется, только когда часы реального времени выключены ( $RTC\_EN = 0$ ).

Среди функций модуля RTC следует отметить два отдельных регистра событий (ВКР\_RTC\_ALRMAR и ВКР\_RTC\_ALRMBR), с помощью которых можно реализовать будильники. Формат регистров событий аналогичен счетным регистрам, что несколько упрощает их программирование.

Сторожевой таймер ВКР\_RTC\_WUT может работать либо на частоте SEC\_CLK, которая используется для работы счетчика/календаря, либо на независимой частоте, задаваемой с помощью бит WUCK\_SEL[2:0]. Регистр ВКР\_RTC\_ALRM предназначен для задания времени сторожевого таймера, при превышении которого устанавливается флаг прерывания WUTF.

Еще одной полезной функцией может оказаться функция TIMESTAMP, предназначенная для определения точного времени наступления внешнего события, детектированного на входах WAKEUP1-WAKEUP3. Активный фронт сигнала на входах WAKEUP1-WAKEUP3, по которому фиксируется внешнее событие, устанавливается в регистре ВКР\_RTC\_TAFCR. При детектировании заданного фронта на входах WAKEUP1-WAKEUP3 значения счетных регистров сохраняются в соответствующих регистрах ВКР\_RTC\_TSTR и ВКР\_RTC\_TSDR.

В батарейном домене реализована возможность мониторинга входных сигналов WAKEUP1-WAKEUP3 с помощью функции TAMPER (регистр ВКР\_RTC\_TAFCR). В регистре ВКР\_RTC\_TAFCR записывается контролируемый активный уровень события (передний фронт и высокий уровень или задний фронт и низкий уровень). С ревизии 2 добавлена возможность устанавливать фиксацию события только по уровню или только по фронту. Если сигнал на входе станет идентичным записанному, то это событие регистрируется в управляющих битах TAMP\_IN\_SEL[2:0] и TAMPF.

Для организации аварийного питания микросхемы от батареи в батарейном домене реализован вывод SW, позволяющий управлять внешним ключом, который осуществляет замыкание питания батареи на основное питание микросхемы. Вывод SW всегда работает на выход. Бит SW\_CNTRL регистра ВКР\_LDO разрешает управление выводом SW с помощью внутреннего сигнала SW\_PWR. Если SW\_CNTRL = 0, то на вывод SW выводится состояние низкого логического уровня (внешний ключ разомкнут). Если SW\_CNTRL = 1, то выходное состояние вывода SW определяется сигналом SW\_PWR:

- SW\_PWR = 0 (внешний ключ разомкнут): на вывод SW выводится низкий логический уровень;
- SW\_PWR = 1 (внешний ключ замкнут): на вывод SW выводится высокий логический уровень.

Сигнал SW\_PWR управляется функцией TAMPER, осуществляющей мониторинг входов WAKEUP1-WAKEUP3, и сторожевым таймером ВКР\_RTC\_WUT, выполняющим сброс сигнала SW\_PWR. Начиная с ревизии 2 сторожевой таймер ВКР\_RTC\_WUT также может устанавливать сигнал SW\_PWR. Установка сигнала SW\_PWR в «1» (замыкание внешнего ключа) осуществляется по следующим событиям:

- обнаружение события TAMPER на любом из входов WAKEUP1-WAKEUP3 при отсутствии основного питания микросхемы;
- обнаружение низкого логического уровня на любом из входов WAKEUP1-WAKEUP3 при выключенной функции TAMPER и отсутствии основного питания микросхемы (только для ревизии 1);
- срабатывание сторожевого таймера ВКР\_RTC\_WUT (WUTF = 1) при WAKE\_SLEEP = 1 в регистре ВКР\_RTC\_TAFCR и отсутствии основного питания микросхемы (начиная с ревизии 2).

Сброс сигнала SW\_PWR в «0» (размыкание внешнего ключа) осуществляется по следующим событиям:

- запись «1» в бит SW\_OFF регистра ВКР\_LDO;
- срабатывание сторожевого таймера ВКР\_RTC\_WUT (WUTF = 1) при WAKE\_SLEEP = 0 в регистре ВКР\_RTC\_TAFCR.

## 17.2 Память регистров и криптографический ключей

Батарейный домен имеет 16 встроенных 32-разрядных регистров для хранения бит управления батарейным доменом и RTC. Для разработчика программ предусмотрено ОЗУ криптографических ключей емкостью 512 байт.

## 17.3 Описание регистров блока батарейного домена

Таблица 265 – Описание регистров блока батарейного домена

Базовый Адрес	Название	Описание
0x4006_0000	ВКР	Контроллер батарейного домена и часов реального времени
Смещение		
0x00	ВКР_WPR	Регистр ключа для разрешения работы с регистрами блока.
0x04	ВКР_LDO	Регистр управления блоком LDO
0x08	ВКР_CLK	Регистр управления блоками HSI, LSI и LSE
0x0C	ВКР_RTC_CR	Регистр управления блоком RTC
0x10	ВКР_RTC_WUTR	Регистр счетчика сторожевого таймера
0x14	ВКР_RTC_PREDIV_S	Регистр счетчика предварительного делителя счетчика/календаря
0x18	ВКР_RTC_PRL	Регистр коэффициента деления предварительного делителя счетчика/календаря
0x1C	ВКР_RTC_ALRM	Регистр значения для сравнения со значением счетчика сторожевого таймера ВКР_RTC_WUTR и выработки сигнала WUTF
0x20	ВКР_RTC_CS	Регистр управления и состояния флагов часов реального времени
0x24	ВКР_RTC_TR	Регистр времени счетчика/календаря в BCD формате
0x28	ВКР_RTC_DR	Регистр даты счетчика/календаря в BCD формате
0x2C	ВКР_RTC_ALRMAR	Регистр задания события А – будильника с календарным временем
0x30	ВКР_RTC_ALRMBR	Регистр задания события В – будильника с календарным временем
0x34	ВКР_RTC_TSTR1	Регистр времени внешнего события входа 1
0x38	ВКР_RTC_TSTR2	Регистр времени внешнего события входа 2
0x3C	ВКР_RTC_TSTR3	Регистр времени внешнего события входа 3
0x40	ВКР_RTC_TSDR1	Регистр даты внешнего события входа 1
0x44	ВКР_RTC_TSDR2	Регистр даты внешнего события входа 2
0x48	ВКР_RTC_TSDR3	Регистр даты внешнего события входа 3
0x4C	ВКР_RTC_TAFCR	Регистр управления обнаружением событий
0x1000-0x11FC	ВКР_MEM	Память ОЗУ 512 байт

### 17.3.1 ВКР\_МЕМ (128x32)

Таблица 266 – Память ВКР\_МЕМ

Номер	31...0
Доступ	R/W
Сброс	X
	ВКР_МЕМ[31:0]

Таблица 267 – Описание бит регистра ВКР\_МЕМ

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	ВКР_МЕМ[31:0]	Ячейка памяти ОЗУ

### 17.3.2 ВКР\_WPR

Таблица 268 – Память ВКР\_WPR

Номер	31...0
Доступ	R/W
Сброс	0
	ВКР_WPR[31:0]

Таблица 269 – Описание бит регистра ВКР\_WPR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	ВКР_WPR[31:0]	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока батарейного домена

### 17.3.3 ВКР\_LDO

Таблица 270 – Регистр ВКР\_LDO

Номер	31, 30	29	28, 27	26	25	15...10	9...7
Доступ	U	R/C1	R/W	R/C1	R/W	U	R
Сброс	0	0	00	0	0	0	000
	-	SW_OFF	MODE[1:0]	TAMPF	FPOR	-	WAKEUP3-1

Номер	6	5	4, 3	2	1, 0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	1	0	00	0	00
	JTAG_ON	SW_CNTRL	LDO_BOOST[1:0]	BLDO_BOOST	BLDO_TRIM[1:0]

Таблица 271 – Описание бит регистра ВКР\_LDO

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	-	Зарезервировано
29	SW_OFF	Выключение аварийного питания микросхемы от батареи. Для выключения записывать 1. По чтению содержит состояние сигнала SW_PWR (при SW_CNTRL = 1 это состояние вывода SW)
28, 27	MODE[1:0]	Режим запуска микросхемы, полученный с входов MODE, при использовании загрузочной программы, описанной в подразделе 9.2 «Пример одного из возможных вариантов загрузочной программы». При использовании загрузочной программы, отличной от описанной в подразделе 9.2, данное поле может применяться по усмотрению пользователя
26	TAMPF	Признак детектирования схемой TAMPER несанкционированного вскрытия устройства по входам WAKEUP3-WAKEUP1 (лог. ИЛИ бит TAMP_IN_SEL[2:0]): 0 – вскрытия не было; 1 – осуществлялось вскрытие. В ревизии 1: Для сброса факта вскрытия необходимо выключить TAMPER с помощью TAMP_EN = 0. Выдержать паузу 100 мкс. Осуществить запись TAMPF = 1. Выдержать паузу 100 мкс. При сбросе флага TAMPF также сбрасываются флаги TAMP_IN_SEL[2:0]. Проверить сброс факта вскрытия чтением бита TAMPF = 0 или чтением бит TAMP_IN_SEL[2:0] = 0. С ревизии 2: Осуществить запись TAMPF = 1. При сбросе флага TAMPF также сбрасываются флаги TAMP_IN_SEL[2:0]. Проверить сброс факта вскрытия чтением бита TAMPF = 0 или чтением бит TAMP_IN_SEL[2:0] = 0
25	FPOR	При использовании загрузочной программы, описанной в подразделе 9.2 «Пример одного из возможных вариантов загрузочной программы»: флаг срабатывания POR. При сбросе по питанию устанавливается в 0. Может использоваться расположенной в BOOT OTP загрузочной программой, описанной в подразделе 9.2, как программный флаг для определения типа сброса. Пример алгоритма анализа бита FPOR: – FPOR = 0: идет выполнение программы после сброса по питанию, установить FPOR в 1; – FPOR = 1: идет выполнение программы после системного сброса, не изменять состояние FPOR. При использовании загрузочной программы, отличной от описанной в подразделе 9.2, данное поле может применяться по усмотрению пользователя

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...10	-	Зарезервировано
9...7	WAKEUP3-1	Состояние входов WAKEUP3-WAKEUP1
6	JTAG_ON	Разрешение работы порта JTAG: 0 – запрещен; 1 – разрешен
5	SW_CNTRL	Бит разрешения управления выводом SW для обеспечения аварийного питания микросхемы от батареи: 1 – разрешено; 0 – запрещено
4, 3	LDO_BOOST[1:0]	Вход активации boost режима работы основного регулятора: 00 – типовое значение выхода регулятора; 01...11 – увеличение выходного значения регулятора напряжения
2	BLDO_BOOST	Вход активации boost режима работы регулятора батарейного домена: 0 – типовое значение выхода регулятора; 1 – увеличение выходного значения регулятора напряжения с помощью BLDO_TRIM[1:0]
1, 0	BLDO_TRIM[1:0]	Подстройка регулятора питания батарейного домена: 00 – типовое значение; 01...11 – увеличение опорного напряжения. Шаг подстройки 2 %

### 17.3.4 ВКР\_CLK

Таблица 272 – Регистр ВКР\_CLK

Номер	31..21	20..15	14	13	12
Доступ	U	R/W	RO	R/W	RO
Сброс	0	000000	1	1	0
	-	HSI_LSI_TRIM[5:0]	HSI_RDY	HSI_ON	LSI_RDY

Номер	11...6	5	4, 3	2	1	0
Доступ	U	R/W	R/W	RO	R/W	R/W
Сброс	0	0	00	0	0	0
	-	LSI_ON	LSE_CONF[1:0]	LSE_RDY	LSE_BYN	LSE_ON

Таблица 273 – Описание бит регистра ВКР\_CLK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..21	-	Зарезервировано
20...15	HSI_LSI_TRIM[5:0]	Коэффициент подстройки частот генераторов HSI и LSI. Уточняется по результатам исследования



Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
14	HSI_RDY	Флаг выхода генератора HSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
13	HSI_ON	Бит управления генератором HSI: 1 – генератор включен; 0 – генератор выключен
12	LSI_RDY	Флаг выхода генератора LSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
11...6	-	Зарезервировано
5	LSI_ON	Бит управления генератором LSI: 0 – генератор выключен; 1 – генератор включен
4, 3	LSE_CONF[1:0]	Биты конфигурации LSE генератора. В случае возникновения проблем с запуском при дефолтном значении конфигурации LSE_CONF[1:0] = 00, следует запускать генератор при других трех конфигурациях. Остальные конфигурации менее чувствительны к помехам, но имеют больший ток потребления. Уточняется по результатам исследования
2	LSE_RDY	Флаг выхода генератора LSE в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
1	LSE_BYR	Бит управления генератором LSE: 0 – режим осциллятора; 1 – режим работы на проход (внешний генератор)
0	LSE_ON	Бит управления генератором LSE: 0 – генератор выключен; 1 – генератор включен

### 17.3.5 ВКР\_RTC\_CR

Таблица 274 – Регистр ВКР\_RTC\_CR

Номер	31...26	25	24...15	14...12	11
Доступ	U	R/W	U	R/W	R/W
Сброс	0	0	0	000	0
	-	FMT	-	WUCK_SEL[2:0]	RTC_RESET

Номер	10...3	2	1, 0
Доступ	R/W	R/W	R/W
Сброс	0000000	0	00
	RTC_CAL[7:0]	RTC_EN	RTC_SEL[1:0]

Таблица 275 – Описание бит регистра ВКР\_RTC\_CR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...26	-	Зарезервировано
25	FMT	Формат времени: 0 – 24 часа в сутках; 1 – формат времени AM/PM
24...15	-	Зарезервировано
14	WUCK_SEL[2]	Биты выбора независимой синхронизации для сторожевого таймера: WUCK_SEL[2] = 1 – общее тактирование с счетчиком/календарем WUCK_SEL[2] = 0 – независимое тактирование с выбранного делителя
13, 12	WUCK_SEL[1:0]	Биты выбора делителя частоты для сторожевого таймера: WUCK_SEL[1:0] = 3 – делитель на 2; WUCK_SEL[1:0] = 2 – делитель на 4; WUCK_SEL[1:0] = 1 – делитель на 8; WUCK_SEL[1:0] = 0 – делитель на 16
11	RTC_RESET	Сброс часов реального времени: 0 – часы не сбрасываются; 1 – часы сбрасываются. Выполняет сброс RTC, WUT, TAMPER и TIMESTAMP, а также сброс регистров ВКР_RTC_WUTR, ВКР_RTC_PREDIV_S, ВКР_RTC_DR, ВКР_RTC_TR
10...3	RTC_CAL[7:0]	Коэффициент подстройки тактовой частоты RTC_CLK. Из каждых $2^{20}$ тактов RTC_CLK будет замаскировано RTC_CAL тактов: 00000000 – 0 тактов; 00000001 – 1 такт; .... 11111111 – 255 тактов. Таким образом, если исходная частота RTC_CLK равна 32768,00000 Гц, то после подстройки средняя частота тактирования составляет: RTC_CLK_C = 32768,00000 Гц при RTC_CAL = 0; RTC_CLK_C = 32767,96875 Гц при RTC_CAL = 1; ... RTC_CLK_C = 32760,03125 Гц при RTC_CAL = 255. В общем случае средняя частота тактирования RTC_CLK после подстройки составляет: $RTC\_CLK\_C = RTC\_CLK \cdot (1 - RTC\_CAL[7:0] / 2^{20})$ Подстроенная частота RTC_CLK_C используется только для формирования частоты SEC_CLK

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
2	RTC_EN	Бит разрешения работы часов реального времени: 0 – работа запрещена; 1 – работа разрешена. Разрешает работу RTC, WUT и TIMESTAMP
1, 0	RTC_SEL[1:0]	Биты выбора источника тактовой синхронизации часов реального времени: 00 – LSI; 01 – LSE; 10 – HSI_RTC; 11 – HSE_RTC

### 17.3.6 ВКР\_RTC\_WUTR

Таблица 276 – Регистр ВКР\_RTC\_WUTR

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

Таблица 277 – Описание бит регистра ВКР\_RTC\_WUTR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	RTC_WUTR[31:0]	Значение счетчика сторожевого таймера

### 17.3.7 ВКР\_RTC\_PREDIV\_S

Таблица 278 – Регистр ВКР\_RTC\_PREDIV\_S

Номер	31...20	19...0
Доступ	U	R/W
Сброс	0	0
	-	RTC_DIV[19:0]

Таблица 279 – Описание бит регистра ВКР\_RTC\_PREDIV\_S

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	-	Зарезервировано
19...0	RTC_DIV[19:0]	Значение счетчика предварительного делителя счетчика/календаря

### 17.3.8 ВКР\_RTC\_PRL

Таблица 280 – Регистр ВКР\_RTC\_PRL

Номер	31...20	19...0
Доступ	U	R/W
Сброс	0	0x7FFB
	-	RTC_PRL[19:0]

Таблица 281 – Описание бит регистра ВКР\_RTC\_PRL

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	-	Зарезервировано
19...0	RTC_PRL[19:0]	Коэффициент деления для формирования частоты SEC_CLK: $SEC\_CLK = RTC\_CLK / (RTC\_PRL[19:0] + 1)$ , где RTC_CLK – частота тактирования с учетом подстройки RTC_CAL[7:0]

### 17.3.9 ВКР\_RTC\_ALARM

Таблица 282 – Регистр ВКР\_RTC\_ALARM

Номер	31...0
Доступ	R/W
Сброс	0
	RTC_ALARM[31:0]

Таблица 283 – Описание бит регистра ВКР\_RTC\_ALARM

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	RTC_ALARM[31:0]	Значения для сравнения со счетчиком сторожевого таймера ВКР_RTC_WUTR и выработки сигнала WUTF. Сигнал WUTF вырабатывается в момент превышения счетчиком ВКР_RTC_WUTR значения в регистре ВКР_RTC_ALARM: $RTC\_WUTR[31:0] == RTC\_ALRM[31:0] + 1$

### 17.3.10 ВКР\_RTC\_CS

Таблица 284 – Регистр ВКР\_RTC\_CS

Номер	31...14	13	12	11	10	9	8
Доступ	U	R/C1	R/W	R/W	R/C1	R/C1	R/W
Сброс	0	0	0	0	0	0	0
	-	TSF	ALRBF_IE	ALRAF_IE	ALRBF	ALRAF	ALRB_EN

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	RO	R/W	R/W	R/W	R/C1	R/C1	R/C1
Сброс	0	0	0	0	0	0	0	0
	ALRA_EN	WEC	WUTF_IE	SECF_IE	OWF_IE	WUTF	SECF	OWF

Таблица 285 – Описание бит регистра ВКР\_RTC\_CS

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14	-	Зарезервировано
13	TSF	Флаг внешнего события, детектированного схемой TIMESTAMP на входах WAKEUP1-WAKEUP3 (лог. ИЛИ бит TS_IN_SEL[2:0]). Сброс флага осуществляется записью единицы. При сбросе флага TSF также сбрасываются флаги TS_IN_SEL[2:0]
12	ALRBF_IE	Разрешение прерывания по событию ALRBF: 0 – запрещено; 1 – разрешено
11	ALRAF_IE	Разрешение прерывания по событию ALRAF: 0 – запрещено; 1 – разрешено
10	ALRBF	Флаг установки времени события В. Сброс флага осуществляется записью единицы
9	ALRAF	Флаг установки времени события А. Сброс флага осуществляется записью единицы
8	ALRB_EN	Разрешение события В. Совпадение ВКР_RTC_ALRMBR и текущего календарного времени: 0 – запрещено; 1 – разрешено
7	ALRA_EN	Разрешение события А. Совпадение ВКР_RTC_ALRMAR и текущего календарного времени: 0 – запрещено; 1 – разрешено
6	WEC	Флаг выполнения записи в регистры RTC: 0 – можно записывать в один из регистров ВКР_RTC_WUTR, ВКР_RTC_PREDIV_S, ВКР_RTC_DR, ВКР_RTC_TR; 1 – идет запись в один из регистров ВКР_RTC_WUTR, ВКР_RTC_PREDIV_S, ВКР_RTC_DR, ВКР_RTC_TR, запись в регистры запрещена. Бит записи влияет не на блок регистров, а только на один. После записи каждого из регистров ВКР_RTC_WUTR, ВКР_RTC_PREDIV_S, ВКР_RTC_DR, ВКР_RTC_TR необходимо дождаться окончания записи
5	WUTF_IE	Разрешение прерывания по событию WUTF: 0 – запрещено; 1 – разрешено
4	SECF_IE	Разрешение прерывания по событию SECF: 0 – запрещено; 1 – разрешено
3	OWF_IE	Разрешение прерывания по событию OWF: 0 – запрет прерывания по переполнению; 1 – разрешение прерывания по переполнению

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
2	WUTF	Флаг превышения счетчиком сторожевого таймера ВКР_RTC_WUTR значения ВКР_RTC_ALARM: 0 – нет события превышения; 1 – было событие превышения. Флаг устанавливается при RTC_WUTR[31:0] == RTC_ALARM[31:0] + 1 Сброс флага осуществляется записью единицы
1	SECF	Флаг совпадения значений регистров ВКР_RTC_PREDIV_S и ВКР_RTC_PRL: 0 – не было совпадения; 1 – было совпадение По событиям совпадения ВКР_RTC_PREDIV_S и ВКР_RTC_PRL формируется частота SEC_CLK. Таким образом флаг SECF устанавливается на каждом такте частоты SEC_CLK. Сброс флага осуществляется записью единицы
0	OWF	Флаг переполнения сторожевого таймера ВКР_RTC_WUT: 0 – нет переполнения; 1 – было переполнение Сброс флага осуществляется записью единицы

### 17.3.11 ВКР\_RTC\_TR

Таблица 286 – Регистр ВКР\_RTC\_TR

Номер	31...23	22	21, 20	19..16	15
Доступ	U	R/W	R/W	R/W	U
Сброс	0	0	00	0000	0
	-	PM	HT[1:0]	HU[3:0]	-

Номер	14...12	11...8	7	6..4	3..0
Доступ	R/W	R/W	U	R/W	R/W
Сброс	000	0000	0	000	0000
	MNT[2:0]	MNU[3:0]	-	ST[2:0]	SU[3:0]

Таблица 287 – Описание бит регистра ВКР\_RTC\_TR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...23	-	Зарезервировано
22	PM	АМ/PM значение: 0 – АМ или 24-часовой формат; 1 – PM
21, 20	HT[1:0]	Десятки часов в формате BCD
19..16	HU[3:0]	Единицы часов в формате BCD

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15	-	Зарезервировано
14..12	MNT[2:0]	Десятки минут в формате BCD
11..8	MNU[3:0]	Единицы минут в формате BCD
7	-	Зарезервировано
6..4	ST[2:0]	Десятки секунд в формате BCD
3..0	SU[3:0]	Единицы секунд в формате BCD

### 17.3.12 ВКР\_RTC\_DR

Таблица 288 – Регистр ВКР\_RTC\_DR

Номер	31...24	23...20	19...16	15...13
Доступ	U	R/W	R/W	R/W
Сброс	0	0001	1001	010
	-	YT[3:0]	YU[3:0]	WDU[2:0]

Номер	12	11..8	7, 6	5, 4	3...0
Доступ	R/W	R/W	U	R/W	R/W
Сброс	0	0001	0	00	0001
	MT	MU[3:0]	-	DT[1:0]	DU[3:0]

Таблица 289 – Описание бит регистра ВКР\_RTC\_DR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...20	YT[3:0]	Десятки лет в формате BCD
19...16	YU[3:0]	Единицы лет в формате BCD
15...13	WDU[2:0]	День недели: 000 – не используется; 001 – понедельник; ... 111 – воскресенье
12	MT	Десятки месяца в формате BCD
11...8	MU[3:0]	Единицы месяца в формате BCD
7, 6	-	Зарезервировано
5, 4	DT[1:0]	Десятки даты в формате BCD
3...0	DU[3:0]	Единицы даты в формате BCD

### 17.3.13 ВКР\_RTC\_ALRMAR, ВКР\_RTC\_ALRMBR

Таблица 290 – Регистры ВКР\_RTC\_ALRMAR, ВКР\_RTC\_ALRMBR

Номер	31	30	29, 28	27...24	23	22	21, 20
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	00	0000	0	0	00
	MSK4	WDSEL	DT[1:0]	DU[3:0]	MSK3	PM	HT[1:0]

Номер	19...16	15	14...12	11...8	7	6...4	3...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0000	0	000	0000	0	000	0000
	HU[3:0]	MSK2	MNT[2:0]	MNU[3:0]	MSK1	ST[2:0]	SU[3:0]

Таблица 291 – Описание бит регистров ВКР\_RTC\_ALRMAR, ВКР\_RTC\_ALRMBR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	MSK4	Маска даты: 0 – дата учитывается; 1 – дата не учитывается
30	WDSEL	Выбор дня недели: 0 – используется дата DU[3:0], DT[1:0]; 1 – DU[3:0] определяет день недели, DT[1:0] не используется
29, 28	DT[1:0]	Десятки даты в BCD формате
27...24	DU[3:0]	Единицы даты в BCD формате
23	MSK3	Маска часов: 0 – часы учитываются; 1 – часы не учитываются
22	PM	АМ/PM значение: 0 – АМ или 24-часовой формат; 1 – PM
21, 20	HT[1:0]	Десятки часов в BCD формате
19...16	HU[3:0]	Единицы часов в BCD формате
15	MSK2	Маска минут: 0 – минуты учитываются; 1 – минуты не учитываются
14...12	MNT[2:0]	Десятки минут в BCD формате
11...8	MNU[3:0]	Единицы минут в BCD формате
7	MSK1	Маска секунд: 0 – секунды учитываются; 1 – секунды не учитываются
6...4	ST[2:0]	Десятки секунд в BCD формате
3...0	SU[3:0]	Единицы секунд в BCD формате

### 17.3.14 ВКР\_RTC\_TSTR1-ВКР\_RTC\_TSTR3, ВКР\_RTC\_TSDR1-ВКР\_RTC\_TSDR3

Пара регистров, доступных только для чтения, в которых сохраняется дата наступления внешних событий. Формат полностью аналогичен соответствующим счетным регистрам ВКР\_RTC\_TR, ВКР\_RTC\_DR.



17.3.15 ВКР\_RTC\_TAFCR

Таблица 292 – Регистр ВКР\_RTC\_TAFCR

Номер	31..29	28..26
Доступ	R/W	R/W
Сброс	000	000
	TAMP_LEVEL_W[2:0]	TAMP_EDGE_W[2:0]

Номер	25...23	22	21...19	18...16
Доступ	R/W	R/W	R/C1	R/C1
Сброс	000	0	000	000
	TAMP_EN_W[2:0]	WAKE_SLEEP	TS_IN_SEL[2:0]	TAMP_IN_SEL[2:0]

Номер	15...10	9	8...6	5	4	3...1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс		0	0	0	0	0	0
	-	TS_IE	TS_EDGE[2:0]	TS_EN	TAMP_IE	TAMP_TRG[2:0]	TAMP_EN

Таблица 293 – Описание бит регистра ВКР\_RTC\_TAFCR

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	TAMP_LEVEL_W[2:0]	Отключение фиксации уровня на входах WAKEUP1-WAKEUP3 (доступно с ревизии 2): 0 – фиксация уровня длительностью более двух тактов частоты синхронизации часов реального времени включена; 1 – фиксация уровня отключена. Фронт импульса фиксируется в зависимости от бит TAMP_EDGE_W[2:0]. Каждый разряд TAMP_LEVEL_W[2:0] отвечает за конфигурацию фиксации сигнала на соответствующем входе WAKEUP3– WAKEUP1
28...26	TAMP_EDGE_W[2:0]	Отключение фиксации фронта на входах WAKEUP1-WAKEUP3 (доступно с ревизии 2): 0 – фиксация фронта включена; 1 – фиксация фронта отключена, импульс длительностью менее двух тактов частоты синхронизации часов реального времени зафиксирован не будет. Импульс длительностью более двух тактов фиксируется в зависимости от бит TAMP_LEVEL_W. Каждый разряд TAMP_EDGE_W[2:0] отвечает за конфигурацию фиксации сигнала на соответствующем входе WAKEUP3 – WAKEUP1

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
25...23	TAMP_EN_W[2:0]	Индивидуальное разрешение обнаружения события TAMPER на входах WAKEUP1-WAKEUP3 при TAMP_EN=0 (доступно с ревизии 2): 0 – запрещено; 1 – разрешено. Каждый разряд TAMP_EN_W[2:0] отвечает за разрешение обнаружения события TAMPER на соответствующем входе WAKEUP3 – WAKEUP1
22	WAKE_SLEEP	Бит выбора включения/выключения ключа SW по событию сторожевого таймера BKP_RTC_WUT (WUTF=1) (доступно с ревизии 2): 0 – осуществляется размыкание ключа SW (SW_PWR = 0) при срабатывании сторожевого таймера BKP_RTC_WUT (WUTF=1); 1 – осуществляется замыкание ключа SW (SW_PWR = 1) при срабатывании сторожевого таймера BKP_RTC_WUT (WUTF=1)
21...19	TS_IN_SEL[2:0]	Входы, детектированные схемой TIMESTAMP: 0 – вход не сработал; 1 – вход сработал. Каждый разряд TS_IN_SEL[2:0] отвечает за срабатывание соответствующего входа WAKEUP3 - WAKEUP1. Возможна установка нескольких разрядов одновременно при срабатывании нескольких входов. Сбрасывается записью единицы в соответствующий бит (доступно с ревизии 2)
18...16	TAMP_IN_SEL[2:0]	Входы, детектированные схемой TAMPER: 0 – вход не сработал; 1 – вход сработал. Каждый разряд TAMP_IN_SEL[2:0] отвечает за срабатывание соответствующего входа WAKEUP3 – WAKEUP1. Возможна установка нескольких разрядов одновременно при срабатывании нескольких входов. Сбрасывается записью единицы в соответствующий бит (доступно с ревизии 2)
15...10	-	Зарезервировано
9	TS_IE	Разрешение прерывания по внешнему событию TIMESTAMP (флаг TSF): 0 – запрещено; 1 – разрешено

Биты	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8...6	TS_EDGE[2:0]	Установка фронта срабатывания для внешнего события TIMESTAMP: 0 – передний фронт; 1 – задний фронт Каждый разряд TS_EDGE[2:0] отвечает за конфигурацию фронта срабатывания сигнала на соответствующем входе WAKEUP3 - WAKEUP1
5	TS_EN	Разрешение сохранения времени по внешнему событию TIMESTAMP, зафиксированному на входах WAKEUP1 – WAKEUP3: 0 – запрещено; 1 – разрешено После изменения состояния бита, задержка включения/выключения TIMESTAMP составляет два такта частоты синхронизации часов реального времени
4	TAMP_IE	Разрешение прерывания по событию TAMPER (флаг TAMPF): 0 – запрещено; 1 – разрешено.
3...1	TAMP_TRG[2:0]	Активный уровень события TAMPER: 0 – передний фронт или высокий уровень; 1 – задний фронт или низкий уровень Каждый разряд TAMP_TRG[2:0] отвечает за конфигурацию фиксации сигнала на соответствующем входе WAKEUP3 – WAKEUP1
0	TAMP_EN	Разрешение обнаружения события TAMPER на входах WAKEUP1 – WAKEUP3: 0 – запрещено; 1 – разрешено После изменения состояния бита, задержка включения/выключения TAMPER составляет два такта частоты синхронизации часов реального времени

## 18 Порты ввода-вывода

Микросхема имеет четыре порта ввода-вывода. Порт А – 16-разрядный, порт В – 15-разрядный, порт С – 8-разрядный, порт D – 16-разрядный и их выходы мультиплексируются между различными функциональными блоками, управление для каждого вывода порта отдельное. Для того, чтобы выходы порта перешли под управление того или иного периферийного блока, необходимо задать для нужных выводов выполняемую функцию и настройки.

Таблица 294 – Порты ввода-вывода

Вывод	Аналоговая функция ANALOG_EN=0		Цифровая функция						
			Порт IO		Основная	Альтернативная	Переопределенная		
			MODE=00 ANALOG_EN=1	MODE=01 ANALOG_EN=1	MODE=10 ANALOG_EN=1	MODE=11 ANALOG_EN=1			
Порт А									
PA0	-		PA0	TMR1_CH1	1	SSP2_FSS	9	-	
PA1	-		PA1	TMR1_CH1N		SSP2_CLK		-	
PA2	-		PA2	TMR1_CH2		SSP2_RXD		-	
PA3	-		PA3	TMR1_CH2N		SSP2_TXD		-	
PA4	-		PA4	TMR1_CH3		-		-	
PA5	-		PA5	TMR1_CH3N		-		-	
PA6	-		PA6/TCK	TMR1_CH4		-		-	
PA7	-		PA7/TDO	TMR1_CH4N		-		-	
PA8	-		PA8/TMS	TMR1_ETR		-		-	
PA9	-		PA9/TDI	TMR1_BRK		-		-	
PA10	-		PA10	EXT_INT1		UART3_RXD	10	TMR4_CH1	16
PA11	-		PA11	TMR2_CH4N	2	UART3_TXD		TMR4_CH1N	
PA12	-		PA12	SSP1_FSS	3	-		TMR4_CH2	
PA13	-		PA13	SSP1_CLK		-		TMR4_CH2N	
PA14	-		PA14	SSP1_RXD		-		TMR4_CH3	
PA15	-		PA15	SSP1_TXD		-		TMR4_CH3N	
Порт В									
PB0	-		PB0	UART1_TXD	4	UART7816_TXD	11	TMR3_CH1	17
PB1	-		PB1	UART1_RXD		UART7816_RXD		TMR3_CH1N	
PB2	-		PB2	nSIROUT1		UART7816_CLK		TMR3_CH2	
PB3	-		PB3	nSIRIN1		UART7816_CTS		TMR3_CH2N	
PB4	-	1	PB4	nUART1DTR	-	-	TMR3_CH3		
PB5	-		PB5	nUART1RTS	UART7816_RTS	12	TMR3_CH3N		
PB6	-	2	PB6	nUART1RI	EXT_INT2	-	TMR3_CH4		
PB7	-		PB7	nUART1DCD	EXT_INT3	-	TMR3_CH4N		
PB8	-		PB8	nUART1DSR	TMR2_ETR	13	TMR3_ETR		
PB9	-		PB9	nUART1CTS	TMR2_BRK		TMR3_BRK		

Вывод	Аналоговая функция ANALOG_EN=0		Цифровая функция						
			Порт IO	Основная	Альтернативная	Переопределенная			
			MODE=00 ANALOG_EN=1	MODE=01 ANALOG_EN=1	MODE=10 ANALOG_EN=1	MODE=11 ANALOG_EN=1			
PB10	-		PB10	TMR2_CH2	5	UART4_RXD	20	-	
PB11	-		PB11	TMR2_CH2N		UART4_TXD		-	
PB12	-		PB12	TMR2_CH3		-		-	
PB13	-		PB13	TMR2_CH3N		UART2_TXD	6	-	
PB14	-		PB14	TMR2_CH4		UART2_RXD		-	
Порт C									
PC0	-		PC0/MODE0*	TMR4_ETR	19	-		TMR4_CH4	18
PC1	-	3	PC1/MODE1*	TMR4_BRK			-		
PC2	-			PC2	TMR2_CH1	7	SSP3_FSS	14	SCL
PC3	-		PC3	TMR2_CH1N			SSP3_CLK		SDA
PC4	-		PC4	EXT_INT2		SSP3_RXD		-	
PC5	-		PC5	EXT_INT3		SSP3_TXD		-	
PC6	-		PC6	TMR2_ETR	8	SCL	15	-	
PC7	-		PC7	TMR2_BRK				SDA	-
Порт D									
PD0	-		PD0	SSP2_FSS	9				
PD1	-		PD1	SSP2_CLK					
PD2	-		PD2	SSP2_RXD			-		-
PD3	-		PD3	SSP2_TXD			-		-
PD4	-		PD4	UART2_TXD	6	-		-	
PD5	-		PD5	UART2_RXD			-		-
PD6	-		PD6	UART3_RXD	10	-		-	
PD7	-		PD7	UART3_TXD			-		-
PD8	-		PD8	UART4_RXD	20	-		-	
PD9	-		PD9	UART4_TXD			-		-
PD10	-		PD10	UART7816_TXD	11	-		-	
PD11	-		PD11	UART7816_RXD			-		-
PD12	-		PD12	UART7816_CLK			-		-
PD13	-		PD13	UART7816_CTS		-		-	
PD14	-		PD14	UART7816_RTS	12	-		-	
PD15	-		PD15	EXT_INT1			LSE_OUT		-
<p>* См. описание выводов PC0 и PC1 в таблице 1.</p> <p>Примечания</p> <p>1 Обозначения:</p> <p>1 – Таймер 1;</p> <p>2, 5, 7, 8, 13 – Таймер 2;</p> <p>3 – Последовательный интерфейс SSP1;</p> <p>4 – Последовательный интерфейс UART1;</p>									

Вывод	Аналоговая функция ANALOG_EN=0	Цифровая функция			
		Порт IO	Основная	Альтернативная	Переопределенная
		MODE=00 ANALOG_EN=1	MODE=01 ANALOG_EN=1	MODE=10 ANALOG_EN=1	MODE=11 ANALOG_EN=1
6		– Последовательный интерфейс UART2;			
9		– Последовательный интерфейс SSP2;			
10		– Последовательный интерфейс UART3;			
11, 12		– Последовательный интерфейс UART ISO7816;			
14		– Последовательный интерфейс SSP3;			
15		– Последовательный интерфейс I2C;			
16, 18, 19		– Таймер 4;			
17		– Таймер 3;			
20		– Последовательный интерфейс UART4;			

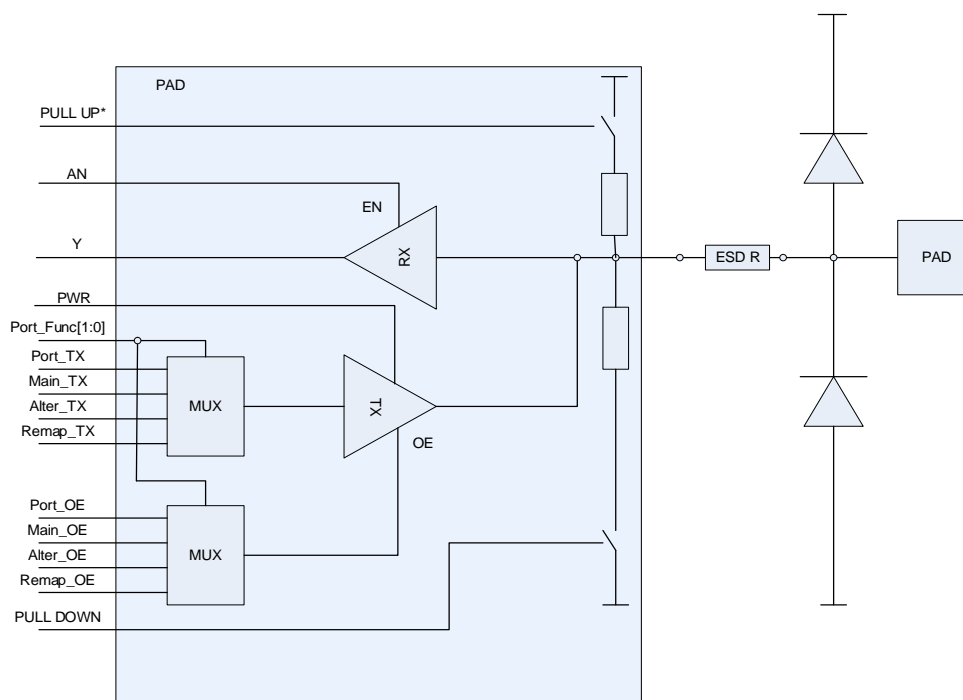
2 Для микросхем до ревизии 2.

**Внимание!** Применяя JTAG на выводах PA6 – PA9 при работе с портом A, рекомендуется использовать только прямую запись в регистр PORTx\_RXTX и не допускать записи логической «1» в бит 7. Для битовых манипуляций необходимо использовать регистры PORTx\_SETTX для установки и регистры PORTx\_CLRTX для сброса, при этом также не допускать установки в логическую «1» бита 7 этих регистров.

Порты PA6, PA8, PA9 должны быть настроены на вход и не должны настраиваться на выход регистром PORTx\_OE.

Если вышеописанные меры не помогают, то не использовать операции с портом A в своей программе при применении на нем функции JTAG

### 18.1 Описание регистров портов ввода-вывода



\* PULL UP доступен с ревизии 2.

Рисунок 30 – Блок схема разряда порта ввода-вывода

Таблица 295 – Описание регистров портов ввода-вывода

Базовый Адрес	Название	Описание
0x4008_0000	GPIO1	Порт А
0x4008_8000	GPIO2	Порт В
0x4009_0000	GPIO3	Порт С
0x400F_0000	GPIO4	Порт D
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
0x18	PORT_PWR[31:0]	Режим мощности передатчика
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

### 18.1.1 PORTx\_RXTX

Таблица 296 – Регистр PORTx\_RXTX

Номер	31..16	15...0
Доступ	U	R/W
Сброс	0	0
	-	PORT RXTX[15:0]

Таблица 297 – Описание бит регистра PORTx\_RXTX

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...0	PORT RXTX[15:0]	Режим работы вывода порта Данные для выдачи на выводы порта и для чтения

### 18.1.2 PORTx\_OE

Таблица 298 – Регистр PORTx\_OE

Номер	31...16	15...0
Доступ	U	R/W
Сброс	0	0
	-	PORT OE[15:0]

Таблица 299 – Описание бит регистра PORTx\_OE

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...0	PORT OE[15:0]	Режим работы вывода порта Направление передачи данных на выводах порта: 1 – выход; 0 - вход

### 18.1.3 PORTx\_FUNC

Таблица 300 – Регистр PORTx\_FUNC

Номер	31, 30	...	3, 2	1, 0
Доступ	R/W	...	R/W	R/W
Сброс	0	...	0	0
	MODE15[1:0]	...	MODE1[1:0]	MODE0[1:0]

Таблица 301 – Описание бит регистра PORTx\_FUNC

Разряды	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

### 18.1.4 PORTx\_ANALOG

Таблица 302 – Регистр PORTx\_ANALOG

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

Таблица 303 – Описание бит регистра PORTx\_ANALOG

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16		
15...0	ANALOG EN[15:0]	Режим работы вывода порта: 0 – аналоговый; 1 – цифровой



### 18.1.5 PORTx\_PULL

Таблица 304 – Регистр PORTx\_PULL

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

Таблица 305 – Описание бит регистра PORTx\_PULL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	PULL UP[15:0]	Режим работы вывода порта (доступно с ревизии 2) Разрешение подтяжки вверх: 1 – подтяжка в единицу включена (есть подтяжка); 0 – подтяжка в единицу выключена
15...0	PULL DOWN[15:0]	Режим работы вывода порта Разрешение подтяжки вниз: 1 – подтяжка в ноль включена (есть подтяжка); 0 – подтяжка в ноль выключена

### 18.1.6 PORTx\_PWR

Таблица 306 – Регистр PORTx\_PWR

Номер	31	30	...	3	2	1	0
Доступ	U	R/W	...	U	R/W	U	R/W
Сброс	-	0	...	-	0	-	0
		PWR15			PWR1		PWR0

Таблица 307 – Описание бит регистра PORTx\_PWR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	PWRx	Аналогично PWR0 для остальных бит порта
1	-	Зарезервировано
0	PWR0	0 – номинальная мощность (нагрузка до 2 мА); 1 – увеличенная мощность (нагрузка до 4 мА)

### 18.1.7 PORTx\_SETTX

Таблица 308 – Регистр PORTx\_SETTX

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

Таблица 309 – Описание бит регистра PORTx\_SETTX

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15...0	SETTX[15:0]	Регистр индивидуальной установки выхода порта Запись единицы в соответствующий разряд регистра устанавливает в единицу соответствующий разряд выхода порта PORTx_RXTX. Запись нуля не влияет на состояние соответствующего выхода порта PORTx_RXTX. Читается ранее записанным в регистр значениями, а не состояние соответствующих входов порта PORT_RXTX

### 18.1.8 PORTx\_CLRTX

Таблица 310 – Регистр PORTx\_CLRTX

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

Таблица 311 – Описание бит регистра PORTx\_CLRTX

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15...0	CLRTX[15:0]	Регистр индивидуального сброса выхода порта Запись единицы в соответствующий разряд регистра сбрасывает в ноль соответствующий разряд выхода порта PORTx_RXTX. Запись нуля не влияет на состояние соответствующего выхода порта PORTx_RXTX. Читается ранее записанным в регистр значениями, а не состояние соответствующих входов порта PORTx_RXTX

## 19 Детектор напряжения питания

Блок детектора напряжения питания (далее – блок PVD) предназначен для контроля двух напряжений питания при работе микросхемы: основного питания  $U_{CC}$  и батарейного питания  $U_{CCB}$ . Блок PVD позволяет сравнивать внешние уровни напряжения с внутренними опорными уровнями и в случае превышения или снижения ниже опорного уровня выработать сигнал или прерывание для последующей программной обработки.

Уровень опорного напряжения для сравнения с напряжением основного питания задается битами PLS[2:0] в регистре PVDCS, для сравнения с батарейным питанием задается битами PLSB[2:0] в регистре PVDCS. В соответствии с уровнями напряжения формируются флаги PVD и PVDB. Данные флаги выставляются при возникновении события и сбрасываются программно.

Таблица 312 – Параметры формирования флагов PVD и PVDB

Параметр	Не менее	Типовое	Не более
Входное основное напряжение, В	2,3	-	3,6
Входное напряжение батареи, В	1,8	-	3,6
Уровень срабатывания PVD, при PLS = «000», В		2,3	
Уровень срабатывания PVDB, при PLSB = «000», В		1,8	
Уровень срабатывания PVD, при PLS = «001», В		2,4	
Уровень срабатывания PVDB, при PLSB = «001», В		2,0	
Уровень срабатывания PVD, при PLS = «010», В		2,5	
Уровень срабатывания PVDB, при PLSB = «010», В		2,2	
Уровень срабатывания PVD, при PLS = «011», В		2,6	
Уровень срабатывания PVDB, при PLSB = «011», В		2,4	
Уровень срабатывания PVD, при PLS = «100», В		2,7	
Уровень срабатывания PVDB, при PLSB = «100», В		2,6	
Уровень срабатывания PVD, при PLS = «101», В		2,8	
Уровень срабатывания PVDB, при PLSB = «101», В		2,8	
Уровень срабатывания PVD, при PLS = «110», В		2,9	
Уровень срабатывания PVDB, при PLSB = «110», В		3,0	
Уровень срабатывания PVD, при PLS = «111», В		3,0	
Уровень срабатывания PVDB, при PLSB = «111», В		3,2	

### 19.1 Описание регистров блока PVD

Таблица 313 – Описание регистров блока PVD

Базовый Адрес	Название	Описание
0x4005_8000	POWER	Датчик подсистемы питания
Смещение		
0x00	PVDCS [26:0]	Регистр управления и состояния датчика питания

19.1.1 PVDCS

Таблица 314 – Регистр PVDCS

Номер	26	25	24	23	22	21	20	19
Доступ	R/W	U	R/W	U	R/W	R/W	R/W	U
Сброс	0	0	0	0	0	0	1	0
	INVB	-	INV	-	IEPVD	IEPVDB	PVD	-

Номер	18	15...13	12...6	5...3	2	1	0
Доступ	R/W	R/W	U	R/W	R/W	U	R/W
Сброс	1	000	00000	000	0	0	0
	PVDB	PLS	-	PLSB	PVDEN	-	PVDBEN

Таблица 315 – Описание бит регистра PVDCS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	INVB	Флаг инверсии выхода от датчика PVDB: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то он выставляется при превышении заданного уровня. Если инвертируется - то при снижении ниже заданного уровня
25	-	Зарезервировано
24	INV	Флаг инверсии выхода от датчика PVD: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то он выставляется при превышении заданного уровня. Если инвертируется - то при снижении ниже заданного уровня
23	-	Всегда записывать ноль
22	IEPVD	Флаг разрешения прерывания от датчика PVD: 0 – прерывание запрещено; 1 – прерывание разрешено
21	IEPVDB	Флаг разрешения прерывания от датчика PVDB: 0 – прерывание запрещено; 1 – прерывание разрешено
20	PVD	Результат сравнения напряжения основного питания: 0 – напряжение питания меньше, чем уровень, задаваемый PLS; 1 – напряжение питания больше, чем уровень, задаваемый PLS. Очищается двойной записью 0. Если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен
19	-	Всегда записывать ноль

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
18	PVDB	Результат сравнения напряжения батарейного питания: 0 – напряжение питания меньше, чем уровень, задаваемый PLSB; 1 – напряжение питания больше, чем уровень, задаваемый PLSB Очищается двойной записью 0. Если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен
17, 16	-	Зарезервировано
15...13	PLS[2:0]	Уровень напряжения для сравнения с напряжением основного питания: 000 – более 2,3 В 001 – более 2,4 В ... 110 – более 2,9 В 111 – более 3,0 В Шаг уровня напряжения 100 мВ
12...6	-	Зарезервировано
5...3	PLSB[2:0]	Уровень напряжения для сравнения с напряжением батарейного питания: 000 – более 1,8 В; 001 – более 2,0 В; ... 110 – более 3,0 В; 111 – более 3,2 В. Шаг уровня напряжения 200 мВ
2	PVDEN	Бит разрешения работы блока датчика напряжения основного питания и блока OverVoltage детектора: 0 – датчик отключен; 1 – датчик включен. Сбрасывается только схемой POR. Записывается только в единицу
1	-	Зарезервировано
0	PVDBEN	Бит разрешения работы блока датчика напряжения батарейного питания: 0 – датчик отключен; 1 – датчик включен

## 20 Таймеры общего назначения

В микросхеме реализовано четыре блока таймеров общего назначения, каждый из которых может быть использован для широкого спектра применений, включая:

- подсчет циклов частоты TIM\_CLK или каких-либо внешних событий;
- формирование прерываний и запросов DMA по заданным событиям;
- захват входных сигналов, в том числе измерение длительности импульсов входных сигналов;
- генерацию различных форм выходных сигналов.

Основу таймеров составляет 32-битный перезагружаемый счетчик. Счет может быть прямой, обратный или двунаправленный. В качестве источника синхросигнала может выступать внутренняя тактовая частота TIM\_CLK, внешние сигналы или другие таймеры.

В каждый блок таймера входит четыре канала, которые имеют в своем составе схему захвата и блок ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки.

Каждый из таймеров позволяет генерировать прерывания и запросы DMA.

### 20.1 Основные характеристики

Основные характеристики блока таймера:

- 32-битный счетчик: счет прямой, обратный или двунаправленный;
- 32-разрядный предварительный делитель частоты TIM\_CLK;
- схема выбора источника тактирования основного счетчика от внешних сигналов или от других таймеров;
- четыре независимых канала, каждый канал может работать в одном из режимов:
  - режим захвата: позволяет захватить (сохранить) текущее значение счетчика при изменении некоторого входного сигнала;
  - режим ШИМ: позволяет осуществлять непрерывное сравнение заданных значений со значением счетчика для формирования выходных сигналов;
- формирование выходных сигналов в режиме ШИМ:
  - сброс в НИЗКИЙ уровень при совпадении;
  - установка в ВЫСОКИЙ уровень при совпадении;
  - переключение (инвертирование) при совпадении;
  - переключение при некотором условии;
- формирование прерываний и запросов DMA по событиям:
  - обновление счетчика;
  - захват;
  - сравнение;
  - внешние события по входам ETR и BRK.

### 20.1.1 Структурная схема

Структурная схема блока «Таймер» представлена на рисунке 31.

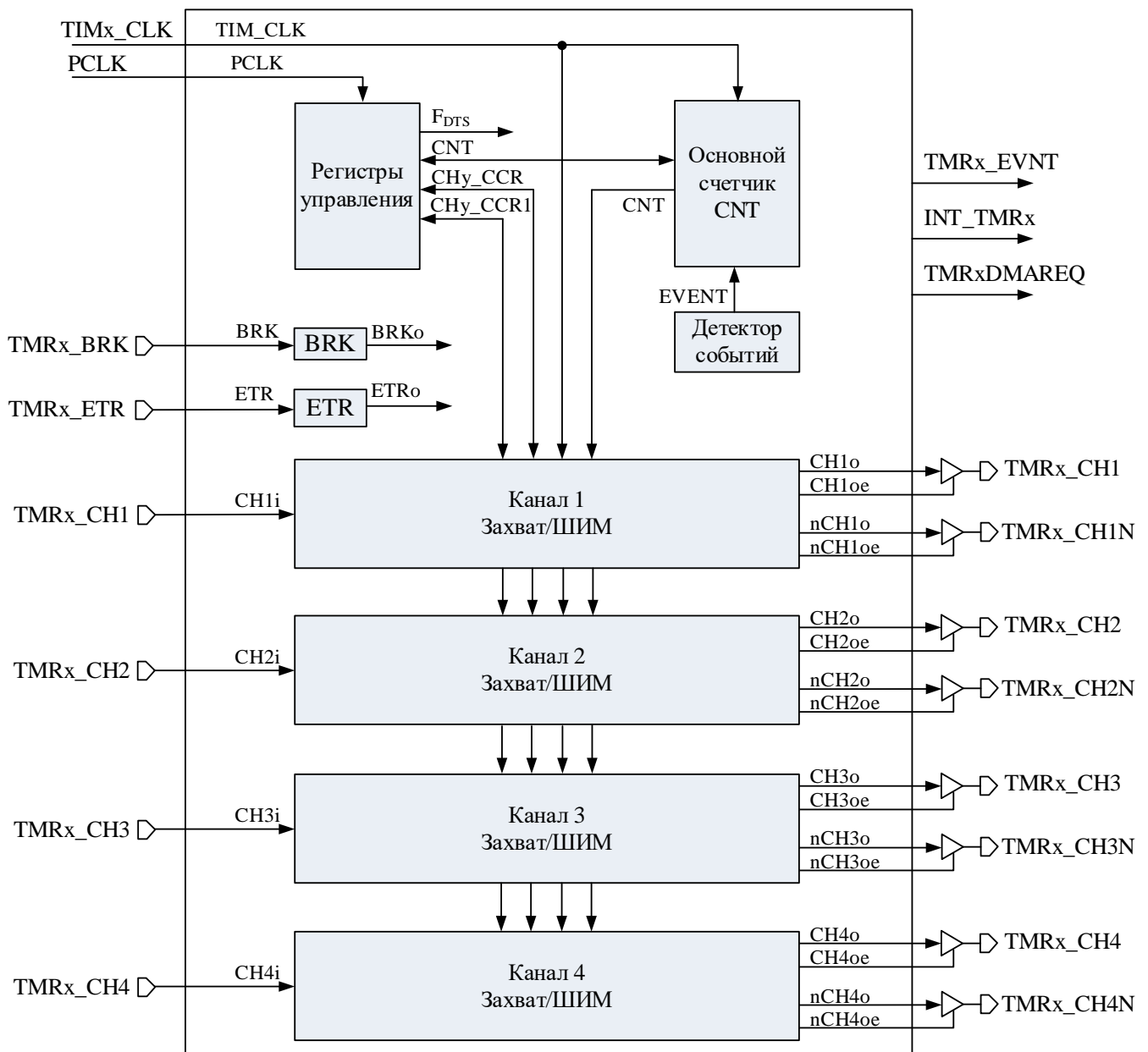


Рисунок 31 – Структурная схема блока «Таймер»

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

Таймер позволяет работать в режимах:

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

### 20.2 Базовый блок таймера

Таймер построен на базе 32-битного счетчика. Базовый блок таймера включает в себя:

- основной счетчик таймера (CNT);
- основание счета (максимальное значение) основного счетчика (ARR);
- делитель частоты TIM\_CLK (PSG), используемый для тактирования основного счетчика;
- регистр управления основным счетчиком (CNTRL).

Сигналом для изменения основного счетчика CNT может служить как внутренняя частота TIM\_CLK, так и события в других счетчиках, либо внешние входные сигналы (см. подраздел 20.3 «Источник событий для счета»).

### 20.2.1 Инициализация тактирования таймера

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

Таймер общего назначения тактируется частотами PCLK и TIM\_CLK. Частота PCLK используется для записи/чтения регистров блока по шине APB. Работа блоков таймера осуществляется на частоте TIM\_CLK.

Для разрешения тактовой частоты PCLK необходимо установить бит тактирования блока в регистре PER2\_CLOCK: бит 14 для таймера 1, бит 15 для таймера 2, бит 27 для таймера 3, бит 28 для таймера 4. Задание тактовой частоты TIM\_CLK осуществляется в регистре TIM\_CLOCK: в поле TIMxBRG устанавливается коэффициент деления тактовой частоты PER1\_C2 для формирования частоты TIM\_CLK, разрешение подачи частоты TIM\_CLK на блок таймера управляется битом TIMxCLKEN. Тактовая частота TIM\_CLK должна быть меньше или равна тактовой частоте PCLK (CPU\_CLK).

После подачи тактовых сигналов можно приступить к работе с таймером.

### 20.2.2 Инициализация основного счетчика таймера

Чтобы запустить работу основного счетчика необходимо задать:

- начальное значение основного счетчика таймера в регистре CNT;
- значение основания счета для основного счетчика в регистре ARR;
- режим работы счетчика в регистре CNTRL:
  - выбрать источник события переключения счетчика EVNT\_SEL[3:0];
  - режим счета основного счетчика CNT\_MODE[1:0]:
    - значения 00 и 01 при тактировании внутренней частотой;
    - значения 10 при тактировании внешними сигналами;
  - направление счета основного счетчика DIR;
- при тактировании внутренней частотой установить значение предварительного делителя в регистре PSG, основной счетчик при этом будет считать на частоте  $TIM\_CLKd = TIM\_CLK / (PSG + 1)$ ;
- разрешить работу счетчика CNT\_EN.

Значения регистров CNT, PSG и ARR можно изменять даже во время работы счетчика. Значения регистров CNT и PSG вступят в силу мгновенно после их записи. Значение регистра основания счета (ARR) может вступить в силу сразу после записи, если в регистре CNTRL бит ARRB\_EN = 0.



При установленном бите  $ARRB\_EN = 1$  записанное значение  $ARR$  применяется при  $CNT == ARR$ . Необходимо учитывать, что если установлен прямой счет таймера, то новое значение  $ARR$  будет использоваться в следующем периоде счета. Если установлен обратный счет таймера, то новое значение  $ARR$  будет использовано через один период счета.

Поле  $CNT\_MODE[1:0]$  в регистре  $CNTRL$  определяет режим работы основного счетчика:

- $CNT\_MODE[1:0] = 00$  или  $10$  – направление счета определяется битом  $DIR$ :
  - $DIR = 0$  – счет прямой;
  - $DIR = 1$  – счет обратный;
- $CNT\_MODE[1:0] = 01$  – счет двунаправленный с автоматическим изменением  $DIR$ .

### 20.2.3 Режимы счета

#### 20.2.3.1 Счет прямой: $CNT\_MODE[1:0] = 00, DIR = 0$

```
TIMERx->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMERx->CNT = 0x00000004; //Начальное значение счетчика
TIMERx->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMERx->ARR = 0x00000013; //Основание счета
//Разрешение работы таймера
TIMERx->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
```

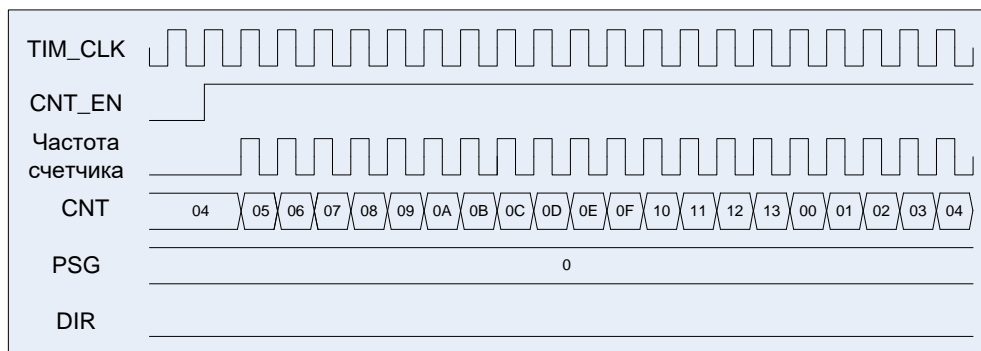


Рисунок 32 – Диаграммы работы таймера, счет прямой от 0 до 0x13, стартовое значение 0x04

#### 20.2.3.2 Счет обратный: $CNT\_MODE[1:0] = 00, DIR = 1$

```
TIMERx->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMERx->CNT = 0x00000004; //Начальное значение счетчика
TIMERx->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMERx->ARR = 0x00000013; //Основание счета
//Разрешение работы таймера.
TIMERx->CNTRL = 0x00000009; //Счет обратный по TIM_CLK
```

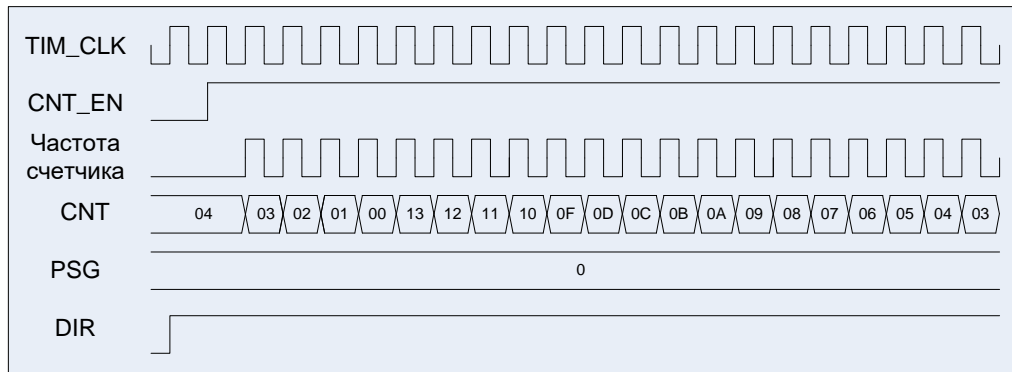


Рисунок 33 – Диаграммы работы таймера, счет обратный от 0x13 до 0, стартовое значение 0x04

### 20.2.3.3 Счет двунаправленный: $CNT\_MODE = 01, DIR = 0$

*TIMERx->CNTRL = 0x00000000; //Режим инициализации таймера*

*//Настраиваем работу основного счетчика*

*TIMERx->CNT = 0x00000004; //Начальное значение счетчика*

*TIMERx->PSG = 0x00000000; //Предделитель частоты TIM\_CLK*

*TIMERx->ARR = 0x00000013; //Основание счета*

*//Разрешение работы таймера.*

*TIMERx->CNTRL = 0x00000041; //Счет двунаправленный по TIM\_CLK*

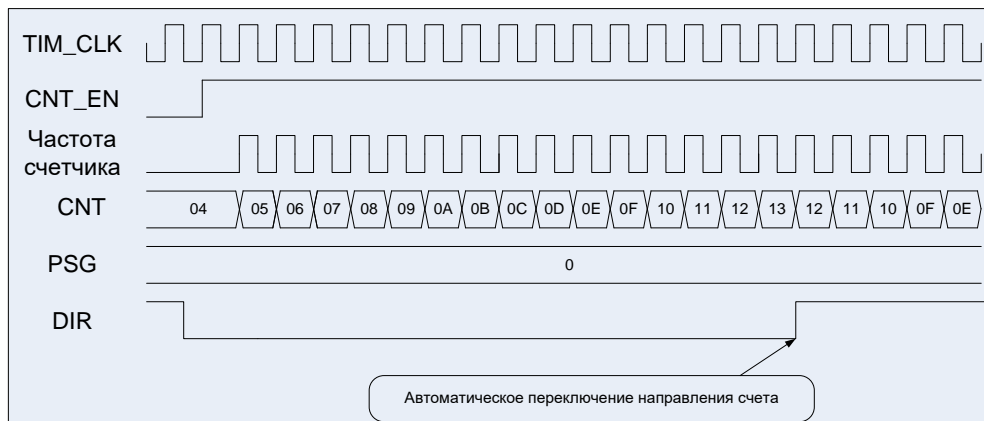


Рисунок 34 – Диаграммы работы таймера, счет двунаправленный, сначала прямой

### 20.2.3.4 Счет двунаправленный: $CNT\_MODE = 01, DIR = 1$

*TIMERx->CNTRL = 0x00000000; //Режим инициализации таймера*

*//Настраиваем работу основного счетчика*

*TIMERx->CNT = 0x00000004; //Начальное значение счетчика*

*TIMERx->PSG = 0x00000000; //Предделитель частоты TIM\_CLK*

*TIMERx->ARR = 0x00000013; //Основание счета*

*//Разрешение работы таймера.*

*TIMERx->CNTRL = 0x00000049; //Счет двунаправленный по TIM\_CLK*

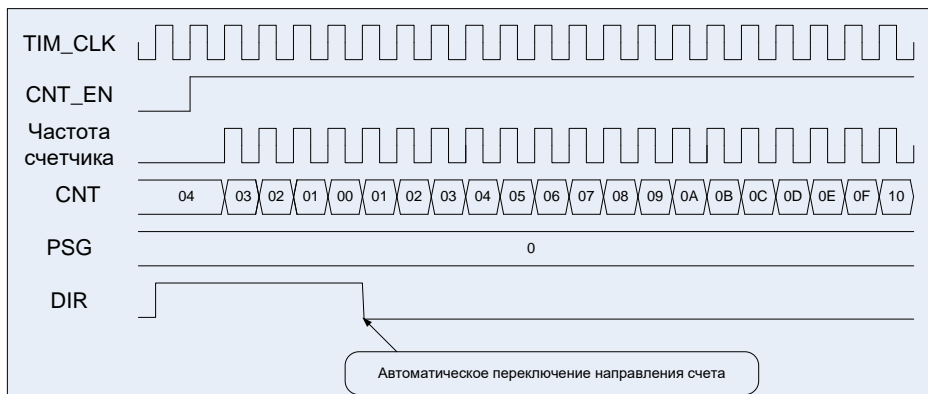


Рисунок 35 – Диаграммы работы таймера, счет двунаправленный, сначала обратный

### 20.2.4 Тактовая частота $F_{DTS}$

В блоке таймера предусмотрено формирование дополнительной тактовой частоты  $F_{DTS}$ , которая может использоваться для работы генератора «мертвой зоны» и цифровых фильтров на входах ETR и CHu<sub>i</sub>.

Тактовая частота  $F_{DTS}$  формируется из частоты TIM\_CLK путем прореживания на заданный коэффициент (1, 2, 3 или 4). Настройка частоты  $F_{DTS}$  осуществляется в поле FDTS[1:0] регистра CNTRL.

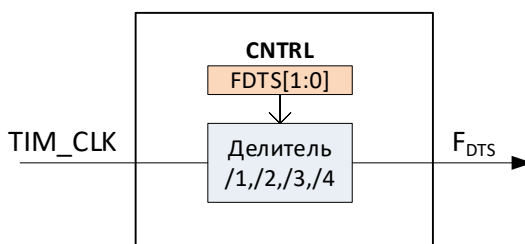


Рисунок 36 – Схема формирования тактовой частоты  $F_{DTS}$

Диаграмма возможных частот  $F_{DTS}$  приведена на рисунке 37.

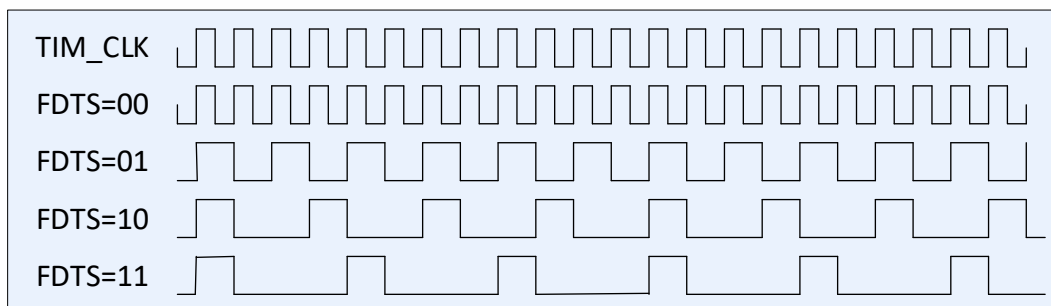


Рисунок 37 – Диаграмма тактовой частоты  $F_{DTS}$  в зависимости от значения FDTS[1:0] в регистре CNTRL

### 20.3 Источник событий для счета

Тактирование основного счетчика таймера может осуществляться от следующих источников:

- внутренний тактовый сигнал (TIM\_CLKd);
- событие в другом таймере (CNT==ARR);
- внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CHu<sub>i</sub>;

– внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR.

Выбор источника тактирования основного счетчика осуществляется в регистре CNTRL, поле EVNT\_SEL[3:0]. При выборе любого источника, кроме внутреннего тактового сигнала (EVNT\_SEL[3:0] = 0000), необходимо также установить CNT\_MODE[1:0] = 10 в регистре CNTRL.

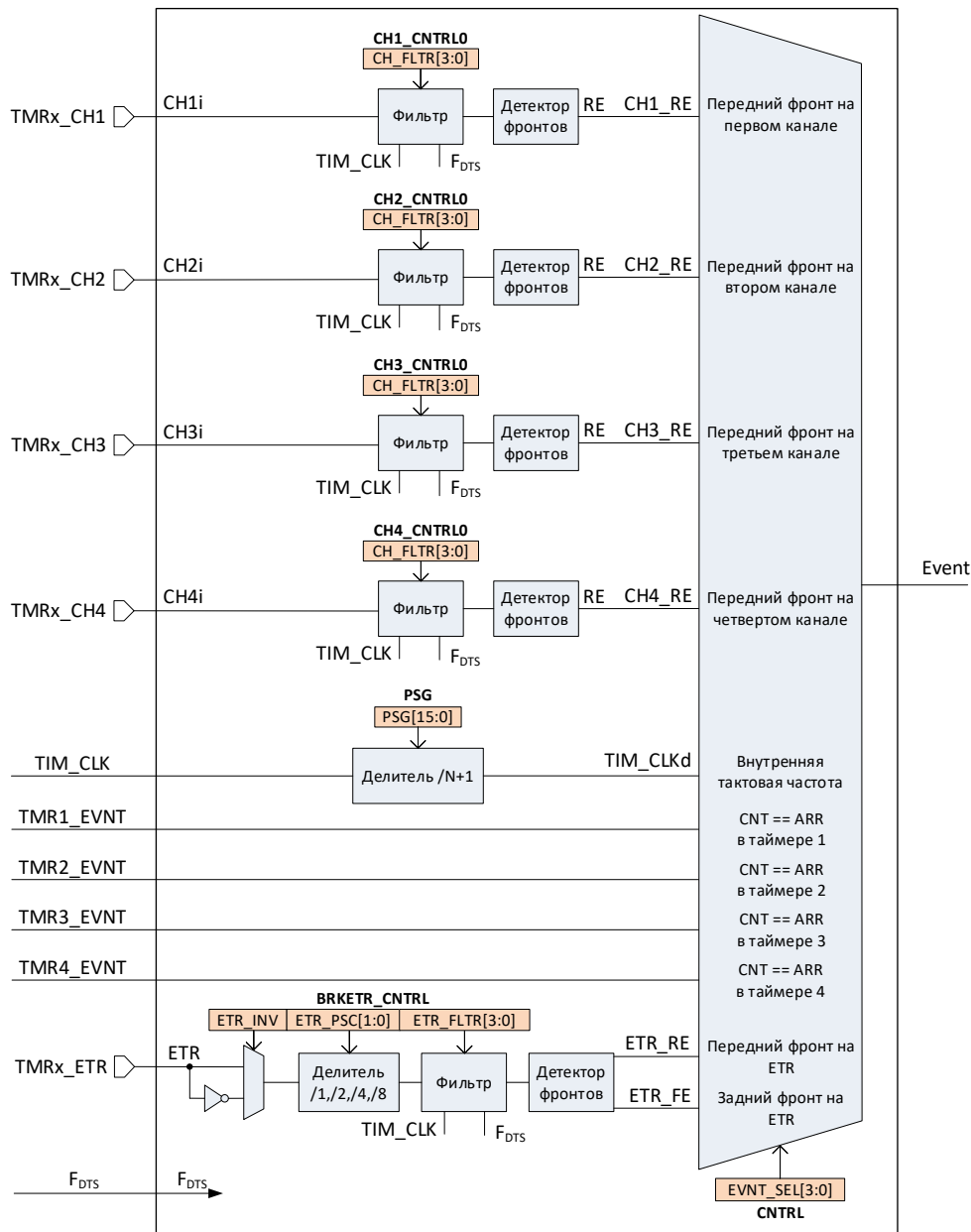


Рисунок 38 – Структурная схема формирования события для счета

### 20.3.1 Внутренний тактовый сигнал (TIM\_CLKd)

Данный режим выбирается, когда EVNT\_SEL[3:0] = 0000 и CNT\_MODE[1:0] = 0x в регистре CNTRL. Основной счетчик таймера тактируется от внутренней частоты TIM\_CLKd, которая формируется путем деления частоты TIM\_CLK в соответствии с коэффициентом деления, записанным в регистре PSG.

Если значение предварительного делителя основного счетчика (PSG) 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 изменяется на 1 и счет начинается заново. Таким образом выходная частота предварительного делителя составляет:

$$TIM\_CLKd = \frac{TIM\_CLK}{PSG + 1} \quad (11)$$

Значение регистра PSG можно изменять даже во время работы счетчика, новое значение предделителя вступит в силу сразу после записи. На рисунках 39 и 40 приведены диаграммы работы счетчика при обновлении значения PSG.

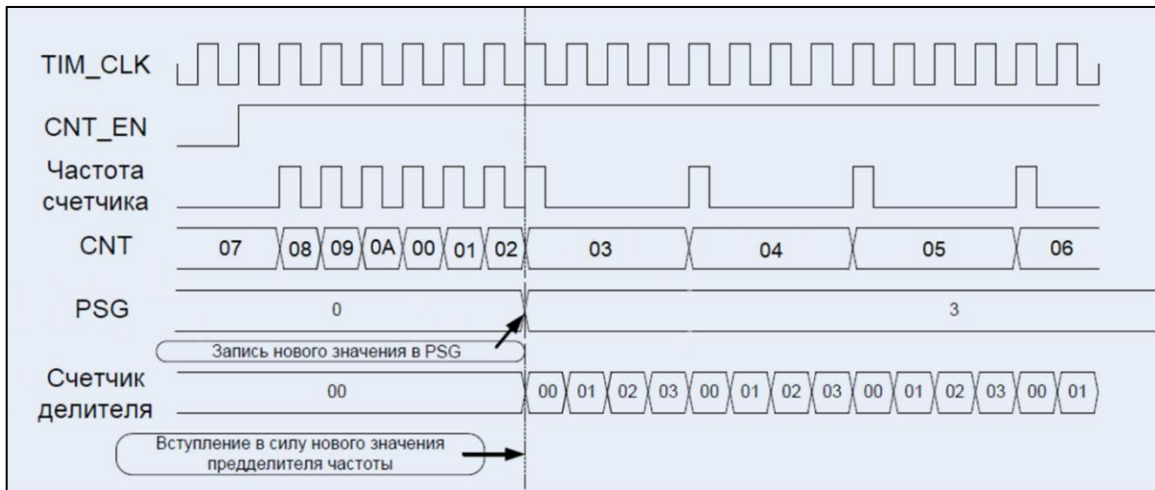


Рисунок 39 – Диаграмма работы счетчика: счет прямой (CNT\_MODE[1:0] = 00, EVNT\_SEL[3:0] = 0000, DIR = 0)

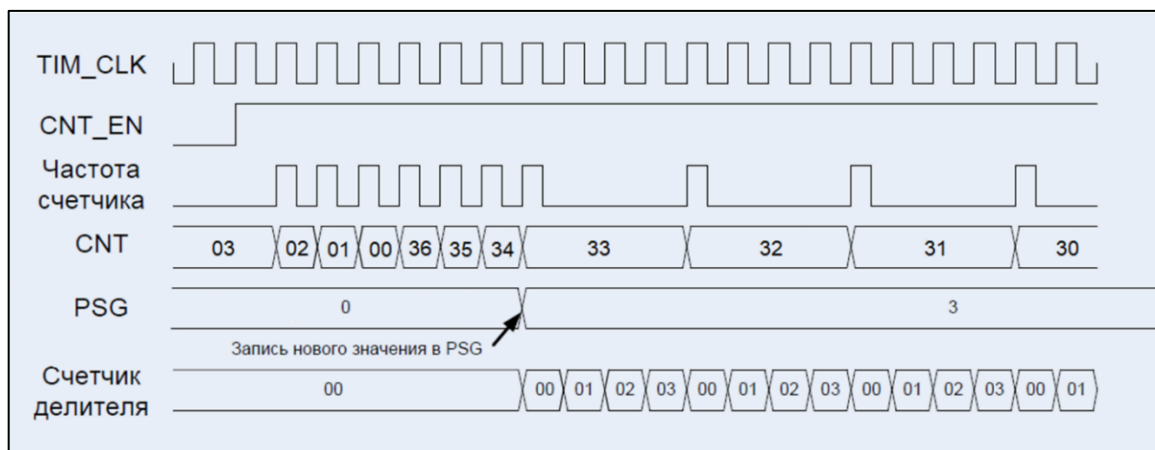


Рисунок 40 – Диаграмма работы счетчика: счет обратный (CNT\_MODE[1:0] = 00, EVNT\_SEL[3:0] = 0000, DIR = 1)

### 20.3.2 Событие в другом таймере (CNT==ARR)

Все таймеры полностью независимы друг от друга, но при этом у них предусмотрена возможность синхронизированной 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 имеется выход запуска TMRx\_EVNT, который соединен с входами других таймеров. Тактирование от другого таймера выбирается, когда EVNT\_SEL[3:0] = 0001 – 0011 или 1010, а также CNT\_MODE[1:0] = 10 в регистре CNTRL. Основной счетчик таймера тактируется от другого таймера по сигналу TMRx\_EVNT, который устанавливается при CNT == ARR. Пересинхронизация сигнала TMRx\_EVNT (CNT == ARR) с одного таймера на другой происходит с задержкой один такт частоты TIM\_CLK.

Синхронизация таймеров возможна в различных режимах. На рисунке 41 показан пример каскадного соединения таймеров, диаграммы работы данных таймеров приведены на рисунке 42.

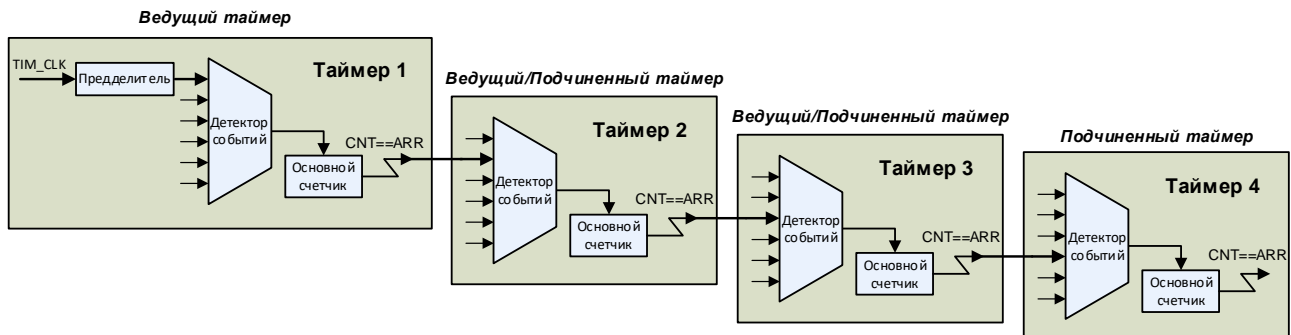


Рисунок 41 – Пример каскадного соединения таймеров

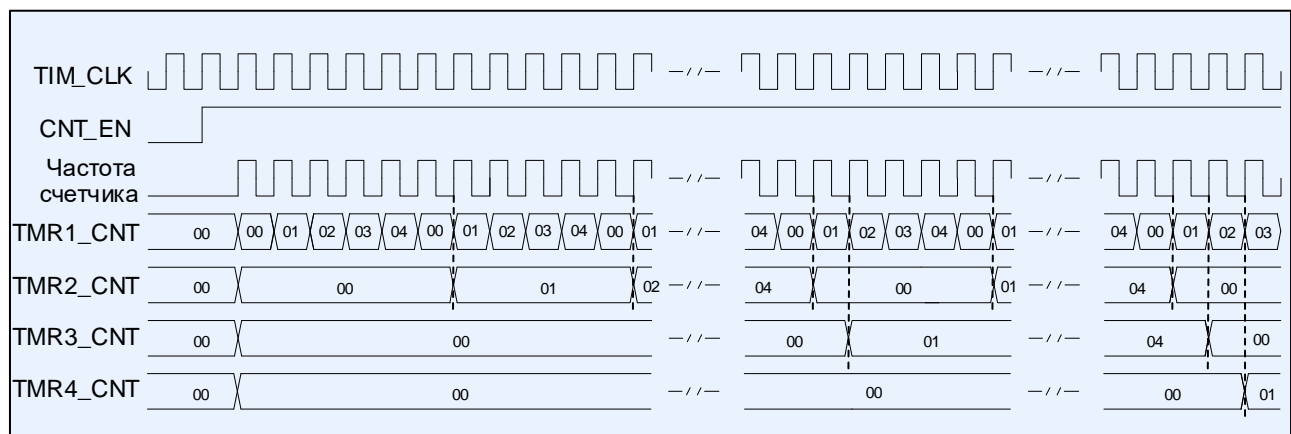


Рисунок 42 – Диаграммы работы четырех таймеров в каскаде  
 TIMER1: DIR = 0, EVNT\_SEL[3:0] = 0000, CNT\_MODE[1:0] = 00;  
 TIMER2: DIR = 0, EVNT\_SEL[3:0] = 0001, CNT\_MODE[1:0] = 10;  
 TIMER3: DIR = 0, EVNT\_SEL[3:0] = 0010, CNT\_MODE[1:0] = 10;  
 TIMER4: DIR = 0, EVNT\_SEL[3:0] = 0011, CNT\_MODE[1:0] = 10

### 20.3.3 Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CHyі

Данный режим выбирается, когда EVNT\_SEL[3:0] = 01xx и CNT\_MODE[1:0] = 10. Основной счетчик таймера считает по переднему фронту внешнего сигнала, поступающего на вход канала CHyі. Биты CH\_SEL[1:0] регистра CHy\_CNTRL0 не оказывают влияния, так как они применяются для работы канала таймера только в режиме захвата.

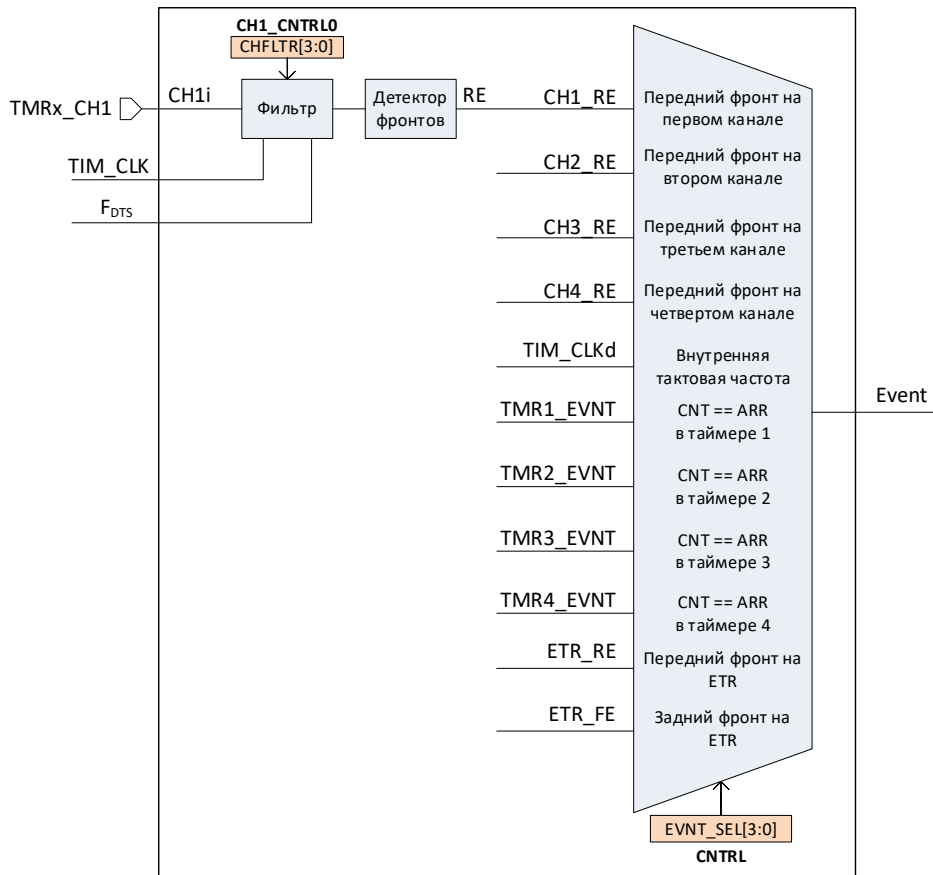


Рисунок 43 – Схема тактирования сигналом со входа первого канала

Со входа  $CH1i$  внешний тактовый сигнал поступает в блок цифрового фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. подраздел 20.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле  $CH\_FLTR[3:0]$  регистра  $CH1\_CNTRL0$ .

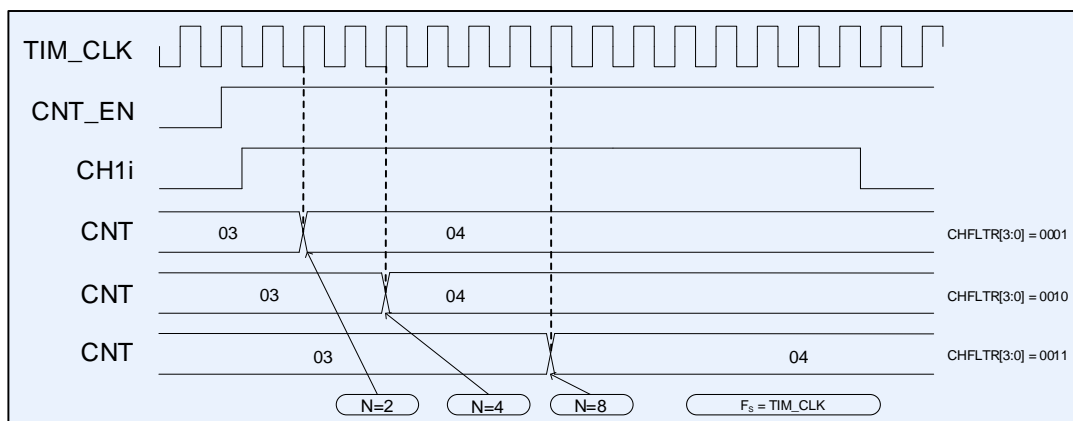


Рисунок 44 – Диаграмма внешнего тактирования с разными вариантами фильтра

### 20.3.4 Внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR

Данный режим выбирается, когда  $EVNT\_SEL[3:0] = 100x$  и  $CNT\_MODE[1:0] = 10$  в регистре  $CNTRL$ . Основной счетчик таймера может тактироваться по переднему или по заднему фронту внешнего сигнала, поступающего на вход  $ETR$ , в зависимости от значения в поле  $EVNT\_SEL[3:0]$ .

Конфигурация тактового сигнала со входа ETR задается в регистре BRKETR\_CNTRL. Бит ETR\_INV позволяет установить инверсию входного сигнала. Поле ETR\_PSC[1:0] задает коэффициент деления асинхронного делителя внешней частоты (1, 2, 4 или 8). После делителя тактовый сигнал поступает в блок цифрового фильтра, где он может быть дополнительно отфильтрован с целью устранения импульсов, длительность которых меньше заданного порога (см. подраздел 20.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле ETR\_FLTR[3:0].

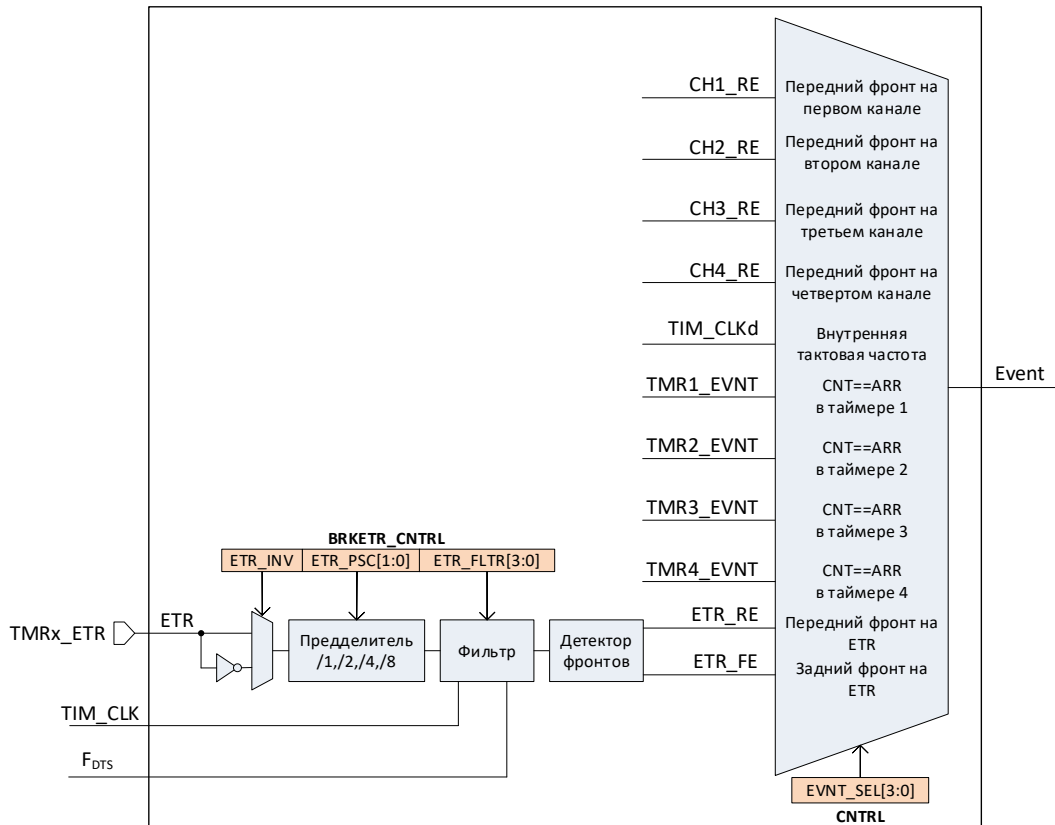


Рисунок 45 – Схема тактирования сигналом со входа ETR

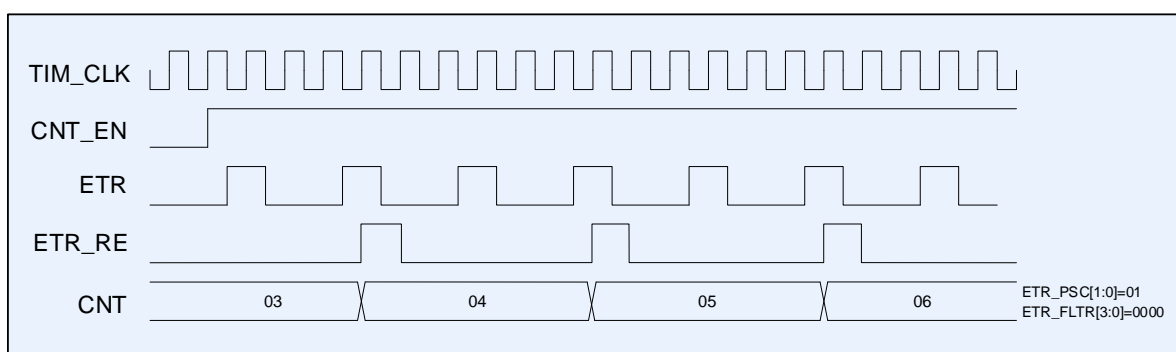


Рисунок 46 – Диаграмма тактирования со входа ETR, EVNT\_SEL[3:0] = 1000

## 20.4 Режим захвата

Каждый канал таймера может быть независимо переведен в режим захвата. В режиме захвата по событию от внешнего входного сигнала происходит фиксация значения основного счетчика CNT в регистры CHy\_CCR и CHy\_CCR1. Регистрация событий осуществляется только по входам CHy<sub>i</sub> (выводы, настроенные в функцию TMRx\_CHy).



Структурная схема блока захвата представлена на рисунке 47.

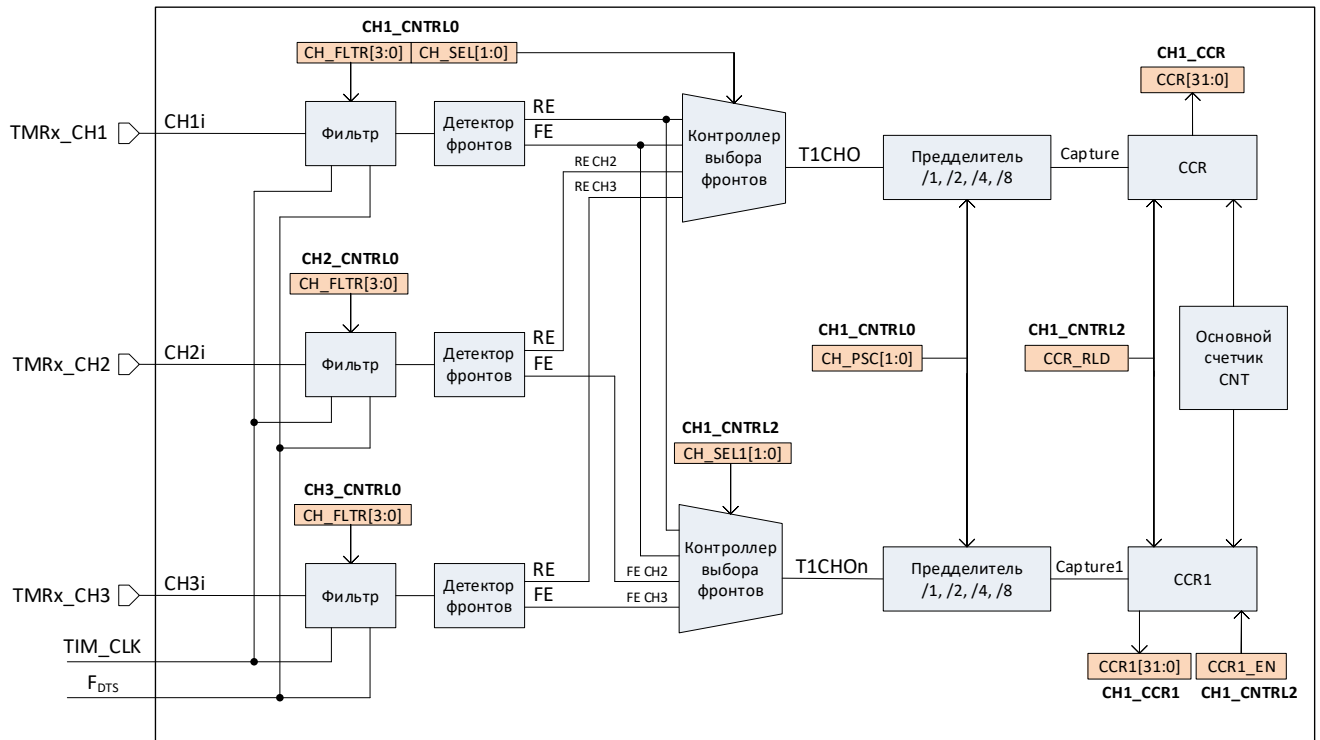


Рисунок 47 – Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо записать «1» в бит CAP\_NPWM регистра управления каналом CHy\_CNTRL0. Для использования регистра CHy\_CCR1 необходимо записать «1» в бит CCR1\_EN регистра CHy\_CNTRL2.

Внешний сигнал со входа CHyi сначала поступает в блок фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. подраздел 20.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле CH\_FLTR[3:0] регистра CHy\_CNTRL0.

Сигнал с блока фильтра поступает в блок «Детектор фронтов». При обнаружении положительного фронта входного сигнала данный блок вырабатывает сигнал RE, а при обнаружении отрицательного фронта входного сигнала – сигнал FE.

В блоке «Контроллер выбора фронтов» производится выбор используемого для захвата сигнала между положительным фронтом канала, отрицательным фронтом канала и положительными и отрицательными фронтами сигналов от других каналов. Настройка блока «Контроллер выбора фронтов» для регистра CHy\_CCR осуществляется в поле CH\_SEL[1:0] регистра CHy\_CNTRL0, а для регистра CHy\_CCR1 – в поле CH\_SEL[1:0] регистра CHy\_CNTRL2. Выбранный для захвата сигнал поступает в предварительный делитель, который в зависимости от значения в поле CH\_PSC[1:0] регистра CHy\_CNTRL0 позволяет фиксировать все события, либо каждое второе, каждое четвертое или каждое восьмое событие.

Предварительный делитель для регистра CHy\_CCR формирует сигнал Capture, а предварительный делитель для регистра CHy\_CCR1 формирует сигнал Capture1. По сигналам Capture и Capture1 выполняется запись текущего значения основного счетчика CNT в регистры CHy\_CCR и CHy\_CCR1.

На рисунке 48 показан пример захвата значения основного счетчика CNT в регистр CHy\_CCR по положительному фронту на входе канала, а в регистр CHy\_CCR1 – по отрицательному фронту на входе канала.

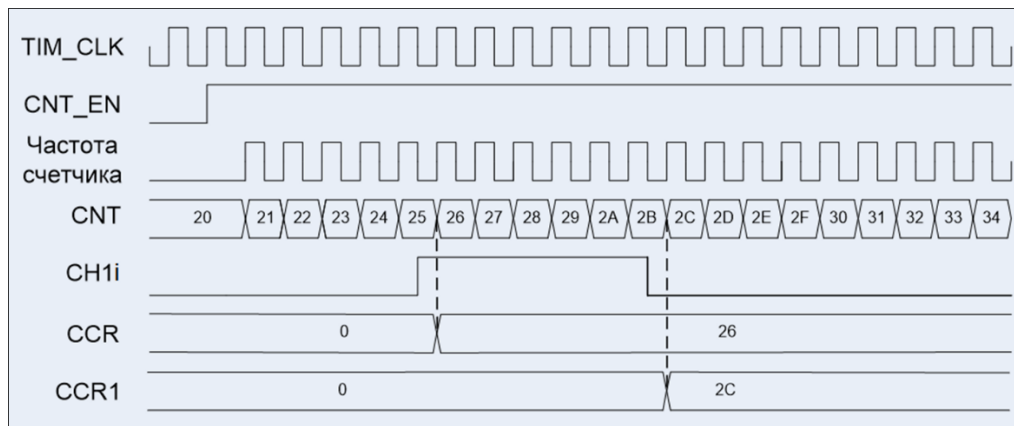


Рисунок 48 – Диаграмма захвата события со входа канала 1

По событию захвата на определенном канале в регистре IE можно разрешить выработку прерываний, а в регистре DMA\_RE можно разрешить формирование запросов DMA.

Между формированием события захвата и записью текущего значения основного счетчика CNT в регистры CHy\_CCR и CHy\_CCR1 может быть установлена задержка с помощью бита EVNT\_DLY в регистре CHy\_CNTRL2. Если бит EVNT\_DLY равен «0», то сначала формируется событие захвата, а затем через один такт сигнала синхронизации TIM\_CLK выполняется запись CNT в регистры CHy\_CCR и CHy\_CCR1. Если бит EVNT\_DLY равен «1», то обновление информации в регистрах CHy\_CCR и CHy\_CCR1 происходит синхронно с событием захвата. При реализации чтения регистров CHy\_CCR и CHy\_CCR1 по событию захвата рекомендуется устанавливать бит EVNT\_DLY в «1».

## 20.5 Режим ШИМ

Каждый канал таймера может быть независимо переведен в режим ШИМ для формирования выходных сигналов с возможностью задания «мертвой зоны». Структурная схема блока формирования ШИМ представлена на рисунке 49.

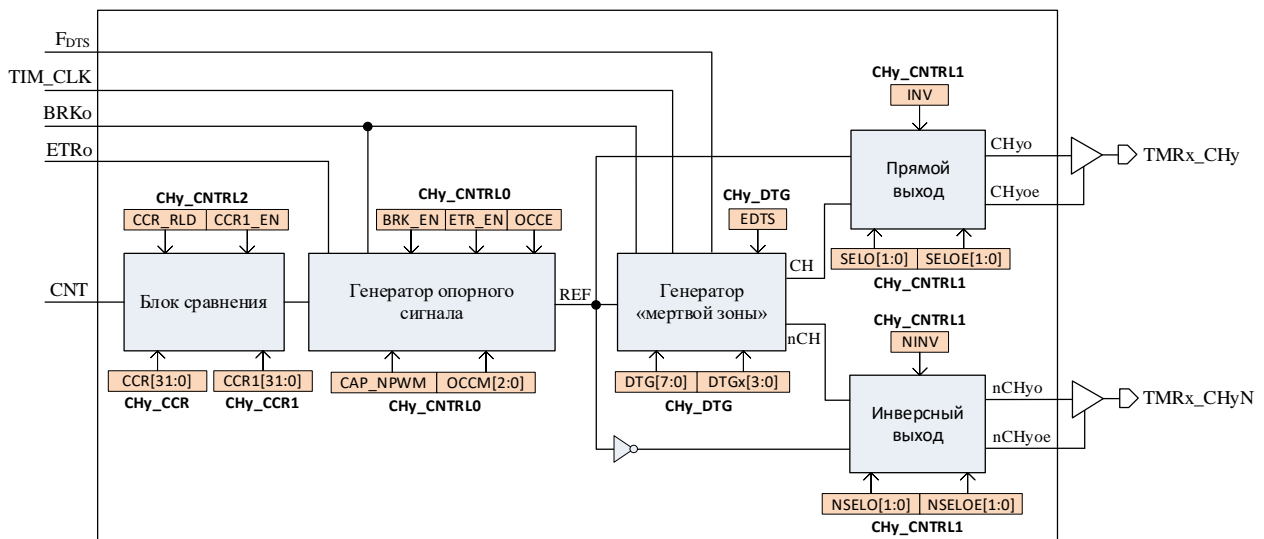


Рисунок 49 – Структурная схема блока формирования ШИМ

Для включения режима ШИМ для определенного канала необходимо в регистре управления каналом CHy\_CNTRL0 записать «0» в бит CAP\_NPWM.

### 20.5.1 Генератор опорного сигнала REF

При работе в режиме ШИМ блок генератора опорного сигнала формирует сигнал REF. Данный сигнал формируется на основании сравнения значения в регистрах CHy\_CCR, CHy\_CCR1 и основного счетчика CNT. Формат выработки сигнала REF устанавливается в поле OCCM[2:0] регистра управления каналом таймера CHy\_CNTRL0.

Если в регистре CHy\_CNTRL2 бит CCR1\_EN = 0, то для формирования сигнала REF используется только результат сравнения значения в регистре CHy\_CCR и основного счетчика CNT.

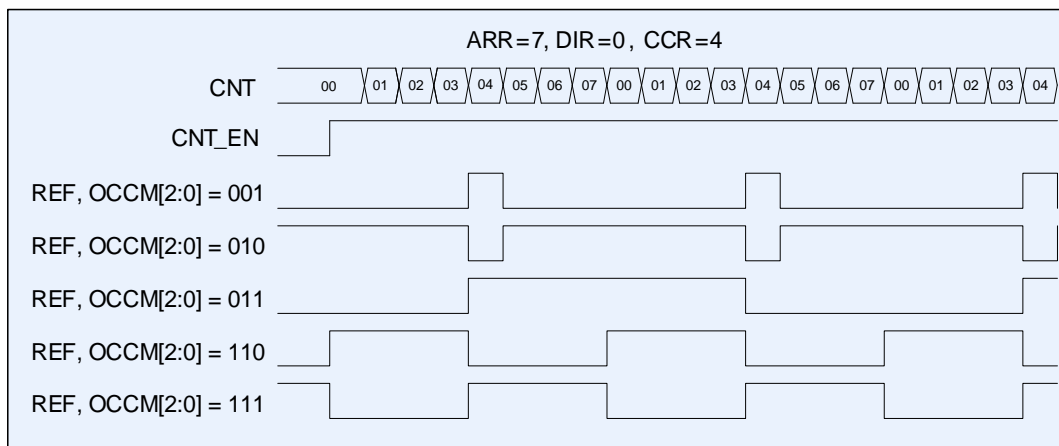


Рисунок 50 – Диаграмма работы в режиме ШИМ, CCR1\_EN=0

Если в регистре CHy\_CNTRL2 бит CCR1\_EN = 1, то для формирования сигнала REF задействуются оба результата сравнения значения в регистрах CHy\_CCR, CHy\_CCR1 и основного счетчика CNT.

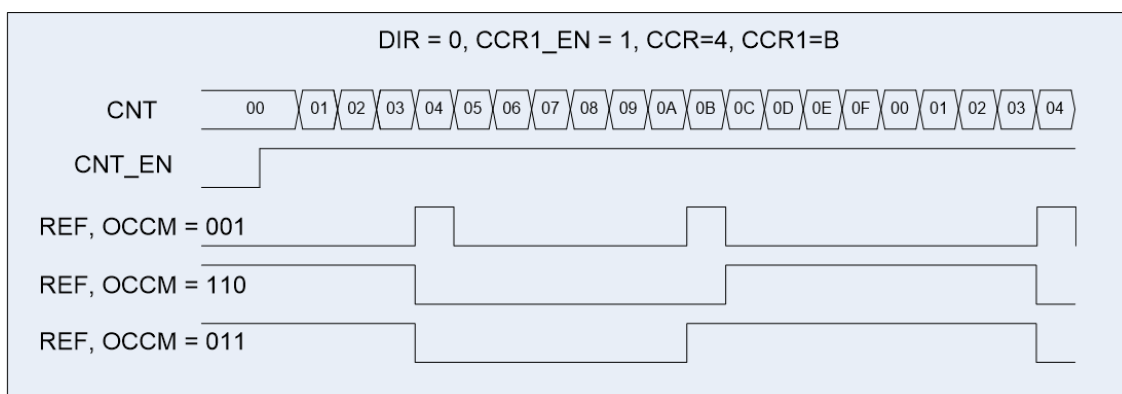


Рисунок 51 – Диаграмма работы в режиме ШИМ, CCR1\_EN = 1

Запись новых значений в регистры CHy\_CCR и CHy\_CCR1 осуществляется немедленно, если в регистре CHy\_CNTRL2 бит CCR\_RLD установлен в «0», иначе регистры CHy\_CCR и CHy\_CCR1 получают новые значения только при CNT == 0. Процесс обновления значений в регистрах CHy\_CCR и CHy\_CCR1 обозначается в регистре CHy\_CNTRL0 с помощью флагов WR\_CMPL и WR\_CMPL1, соответственно. На время

выполнения записи флаг WR\_CMPL/WR\_CMPL1 устанавливается в «1», по окончании записи флаг WR\_CMPL/WR\_CMPL1 сбрасывается в «0».

Сигнал REF может быть принудительно установлен в «0» с использованием внешнего сигнала сброса, поступающего со входа ETR (высокий активный уровень) или со входа BRK (низкий активный уровень). Активный уровень на входах ETR и BRK может быть изменен с помощью инверсии входного сигнала, регистр BRKETR\_CNTRL, биты ETR\_INV и BRK\_INV, соответственно.

Для разрешения сброса сигнала REF по входу ETR необходимо установить бит ETR\_EN и OCCE в регистре CHy\_CNTRL0. Активный уровень на входе ETR сбрасывает сигнал REF в «0». После снятия активного уровня на входе ETR сигнал REF остается в «0» до следующего события установки REF в «1», рисунок 52.

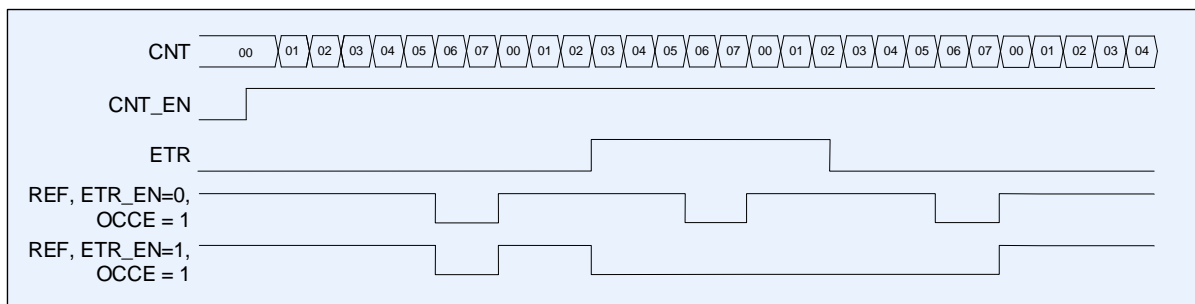


Рисунок 52 – Диаграмма сброса сигнала REF по выводу ETR

Для разрешения сброса сигнала REF по входу BRK необходимо установить бит BRK\_EN в регистре CHy\_CNTRL0. Активный уровень на входе BRK сбрасывает сигнал REF в «0» путем маскирования. После снятия активного уровня на входе BRK генерация сигнала REF сразу же восстанавливается, рисунок 53.

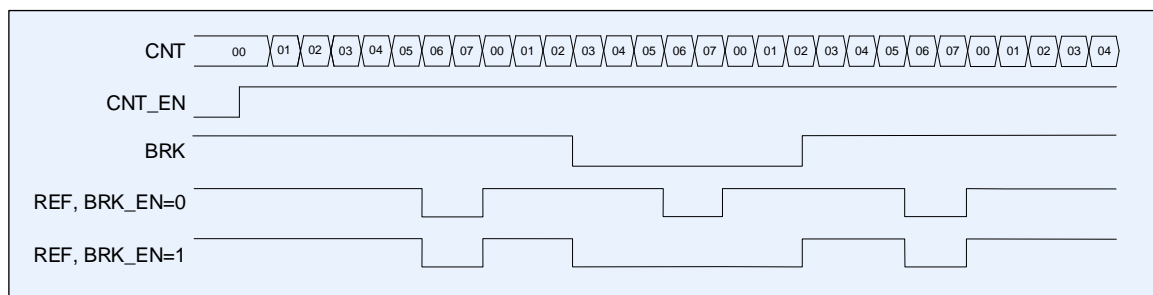


Рисунок 53 – Диаграмма сброса сигнала REF по выводу BRK

### 20.5.2 Генератор «мертвой зоны»

Блок генератора «мертвой зоны» (dead-time generator, DTG) позволяет на основе сигнала REF формировать комплементарную пару сигналов с «мертвой зоной». Выходные сигналы с блока DTG передаются на выходные блоки следующим образом:

- сигнал на прямом выходе (CHy0, CHy0e) представляет собой инвертированный сигнал REF, в котором передний фронт задержан на величину DTGdel относительно заднего фронта опорного сигнала REF;
- сигнал на инверсном выходе (nCHy0, nCHy0e) представляет собой сигнал REF, в котором передний фронт задержан на величину DTGdel относительно переднего фронта опорного сигнала REF.

Значение «мертвой зоны» между сигналами на прямом и инверсном выходах рассчитывается в тактах частоты TIM\_CLK или F<sub>DTS</sub> по формуле

$$DTGdel = DTG \cdot (DTGx + 1), \quad (12)$$

где DTGx – предварительный делитель частоты;

DTG – основной делитель частоты.

Управление блоком DTG осуществляется через регистр CH<sub>y</sub>\_DTG. Выбор источника тактирования для задания «мертвой зоны» задается битом EDTS. Значения делителей DTGx и DTG задаются в полях DTGx[3:0] и DTG[7:0], соответственно. Если задержка DTGdel больше ширины импульса высокого уровня, то соответствующий импульс не генерируется.

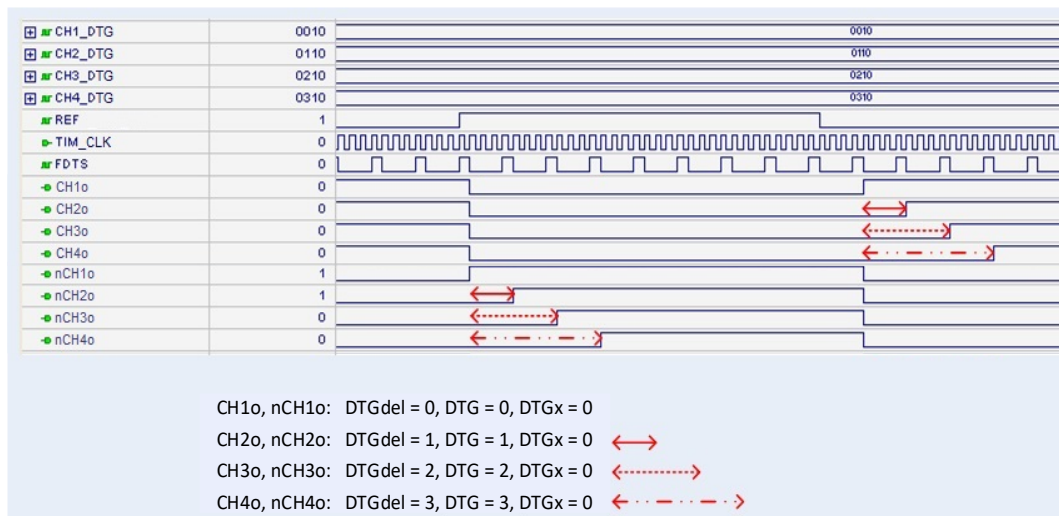


Рисунок 54 – Диаграмма работы блока DTG

Выходные сигналы блока DTG могут быть принудительно установлены в «0» с использованием внешнего сигнала сброса, поступающего со входа BRK (низкий активный уровень). Активный уровень на входе BRK может быть изменен с помощью инверсии входного сигнала, бит BRK\_INV в регистре BRKETR\_CNTRL. Для разрешения сброса выходных сигналов блока DTG по входу BRK необходимо установить бит BRK\_EN в регистре CH<sub>y</sub>\_CNTRL0. Активный уровень на входе BRK сбрасывает выходные сигналы блока DTG в «0» путем их маскирования.

### 20.5.3 Выходные блоки

Каждый канал таймера имеет два выходных блока – прямой и инверсный. Каждый выходной блок формирует как сигнал выдачи (CH<sub>yo</sub>, nCH<sub>yo</sub>), так и сигнал разрешения выдачи (CH<sub>yo</sub>e, nCH<sub>yo</sub>e). В качестве сигналов для прямого (CH<sub>yo</sub>, CH<sub>yo</sub>e) и инверсного (nCH<sub>yo</sub>, nCH<sub>yo</sub>e) выходов в регистре CH<sub>y</sub>\_CNTRL1 могут быть заданы либо постоянные уровни (0 или 1), либо сигналы, формируемые на основе сигнала REF. К таким сигналам относится сам сигнал REF, а также сигналы, формируемые блоком DTG.

Выбор источника сигнала выдачи для прямого (CH<sub>yo</sub>) и инверсного (nCH<sub>yo</sub>) выходов задается в полях SELO[1:0] и NSELO[1:0] регистра CH<sub>y</sub>\_CNTRL1. Дополнительно каждый сигнал выдачи для прямого (CH<sub>yo</sub>) и инверсного (nCH<sub>yo</sub>) выходов может быть инвертирован путем установки битов INV и NINV в

регистре CHy\_CNTRL1. Выбор источника сигнала разрешения выдачи для прямого (CHyое) и инверсного (nCHyое) выходов задается в полях SELOE[1:0] и NSELOE[1:0] регистра CHy\_CNTRL1. При этом, если сигнал разрешения выдачи равен «0», то соответствующий вывод работает в режиме входа, если сигнал разрешения выдачи равен «1» – то в режиме выхода.

## 20.6 Блок цифрового фильтра

В тракте входа ETR и входов каналов таймера CHy<sub>i</sub> предусмотрен блок цифрового фильтра, который позволяет исключить из входного сигнала импульсы высокого и низкого уровня, длительность которых меньше заданного порога.

Конфигурация фильтра для входа ETR выполняется в поле ETR\_FLTR[3:0] регистра BRKETR\_CNTRL, для входов каналов CHy<sub>i</sub> – в поле CH\_FLTR[3:0] регистра CHy\_CNTRL0. Значение в данных полях позволяет настроить два параметра фильтра:

- частота выборки  $F_S$ , на которой входной сигнал захватывается в сдвиговый регистр для накопления. В качестве частоты  $F_S$  может использоваться частота TIM\_CLK или  $F_{DTS}$ ;

- количество выборок (длина фильтра)  $N$ , на протяжении которых входной сигнал должен оставаться стабильным, чтобы не подвергнуться фильтрации.

Если в течение заданного количества выборок  $N$  на частоте  $F_S$  входной сигнал не изменяется, то значение входного сигнала передается на выход фильтра. Иначе внутренний счетчик накопления сбрасывается и захват сигнала начинается заново.

Таким образом, настраивая частоту  $F_S$  и количество выборок  $N$ , задается минимальная длительность импульсов входного сигнала, которые не будут отфильтрованы. Диаграмма работы фильтра при использовании частоты TIM\_CLK приведена на рисунке 55, частота  $F_S = TIM\_CLK$ , количество выборок  $N = 4$ .

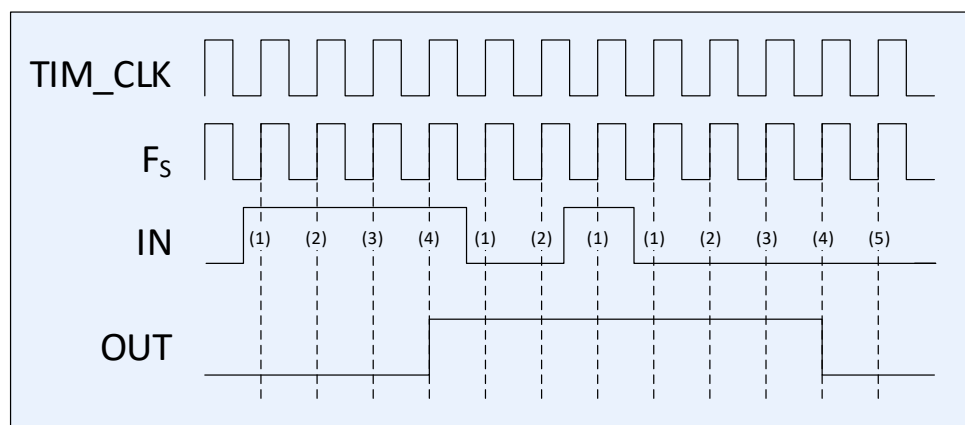


Рисунок 55 – Диаграмма работы фильтра,  $F_S = TIM\_CLK$ ,  $N = 4$

Для задания длительных интервалов накопления входного сигнала имеется возможность использовать частоту  $F_{DTS}$ , которая формируется из частоты TIM\_CLK путем прореживания на заданный коэффициент (см. пункт 20.2.4 «Тактовая частота  $F_{DTS}$ »).

Диаграмма работы фильтра при использовании частоты  $F_{DTS}$  приведена на рисунке 56, частота выборки  $F_S = F_{DTS}/2$ , количество выборок  $N = 6$ .

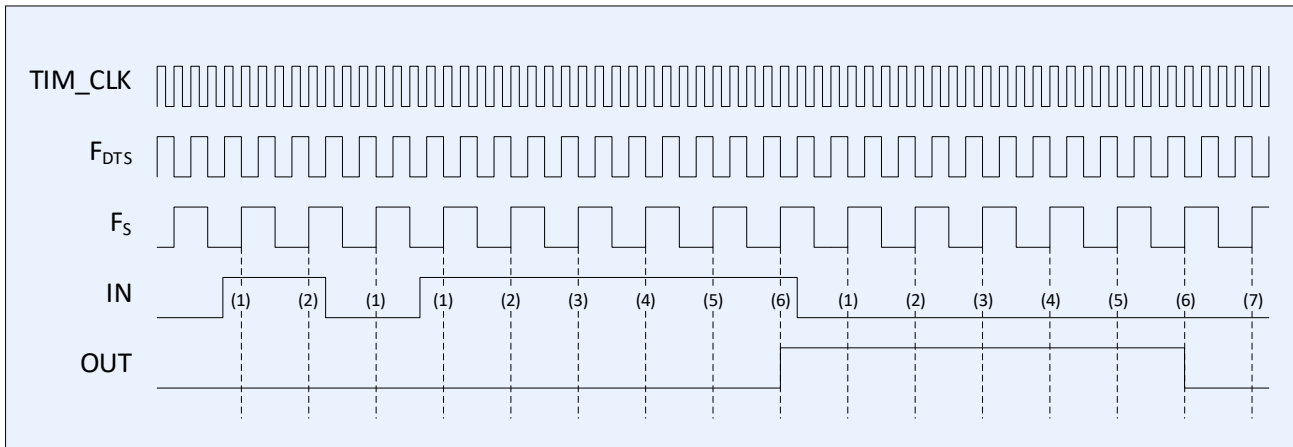


Рисунок 56 – Диаграмма работы фильтра,  $F_S = F_{DTS}/2$ ,  $N = 6$

Возможные варианты настройки блока фильтра приведены в таблице 316.

Таблица 316 – Возможные конфигурации фильтра

CH_FLTR[3:0], ETR_FLTR[3:0]	Частота выборки $F_S$	Количество выборок $N$	Минимальная длительность импульсов, которые не будут отфильтрованы
0000	$F_{DTS}$	1	-
0001	$TIM\_CLK$	2	$2 \times T_{TIM\_CLK}$
0010	$TIM\_CLK$	4	$4 \times T_{TIM\_CLK}$
0011	$TIM\_CLK$	8	$8 \times T_{TIM\_CLK}$
0100	$F_{DTS}/2$	6	$12 \times T_{F_{DTS}}$
0101	$F_{DTS}/2$	8	$16 \times T_{F_{DTS}}$
0110	$F_{DTS}/4$	6	$24 \times T_{F_{DTS}}$
0111	$F_{DTS}/4$	8	$32 \times T_{F_{DTS}}$
1000	$F_{DTS}/8$	6	$48 \times T_{F_{DTS}}$
1001	$F_{DTS}/8$	8	$64 \times T_{F_{DTS}}$
1010	$F_{DTS}/16$	5	$80 \times T_{F_{DTS}}$
1011	$F_{DTS}/16$	6	$96 \times T_{F_{DTS}}$
1100	$F_{DTS}/16$	8	$128 \times T_{F_{DTS}}$
1101	$F_{DTS}/32$	5	$160 \times T_{F_{DTS}}$
1110	$F_{DTS}/32$	6	$192 \times T_{F_{DTS}}$
1111	$F_{DTS}/32$	8	$256 \times T_{F_{DTS}}$

## 20.7 Флаги состояний, прерывания и запросы DMA

В процессе работы блок таймера отслеживает состояние внутренних блоков и формирует 17 событий:

- CNT\_ZERO\_EVNT – совпадение значения счетчика CNT с нулем;
- CNT\_ARR\_EVNT – совпадение значения счетчика CNT со значением в регистре ARR;
- ETR\_RE\_EVNT – фиксация переднего фронта на входе ETR;
- ETR\_FE\_EVNT – фиксация заднего фронта на входе ETR;
- BRK\_EVNT – фиксация высокого уровня на входе BRK;



- CCR\_CAP\_EVNT[3:0] – запись значения счетчика CNT в регистр CHy\_CCR по захвату настроенного фронта на входе канала CHy<sub>i</sub>, события формируются индивидуально для каждого канала;
- CCR\_REF\_EVNT[3:0] – фиксация переднего фронта на выходе генератора опорного сигнала REF, события формируются индивидуально для каждого канала;
- CCR\_CAP1\_EVNT[3:0] – запись значения счетчика CNT в регистр CHy\_CCR1 по захвату настроенного фронта на входе канала CHy<sub>i</sub>, события формируются индивидуально для каждого канала.

### 20.7.1 Флаги состояний

При возникновении события устанавливается соответствующий флаг в регистре STATUS. Сброс флагов в регистре STATUS осуществляется записью «0», запись «1» не оказывает влияния. Если запись «0» выполняется одновременно с новым событием, то приоритет у нового события.

### 20.7.2 Прерывания

Блок таймера на основе флагов в регистре STATUS формирует один общий сигнал запроса прерывания INT\_TMRx. Выбор флагов, формирующих запрос прерывания, осуществляется через регистр разрешения прерываний IE. При формировании запроса прерывания маскированные состояния флагов из регистра STATUS объединяются по схеме ИЛИ.

### 20.7.3 Запросы DMA

На основе отслеживаемых событий блок таймера формирует сигнал запроса DMA TMRxDMAREQ. Выбор событий, формирующих запрос DMA, осуществляется через регистр DMA\_RE.

## 20.8 Примеры

В данном разделе приведены примеры инициализации таймера 1 в различных режимах работы. Для других таймеров инициализация выполняется аналогично.

### 20.8.1 Обычный счетчик

```
RST_CLK->PER2_CLOCK |= 0x00004000; //Разрешение частоты PCLK для таймера 1
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
//для таймера 1 (TIM_CLK = PER1_C2)
TIMER1->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER1->CNT = 0x00000000; //Начальное значение счетчика
TIMER1->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMER1->ARR = 0x0000000F; //Основание счета
TIMER1->IE = 0x00000002; //Разрешение генерировать прерывание при CNT = ARR
//Разрешение работы таймера
TIMER1->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
```



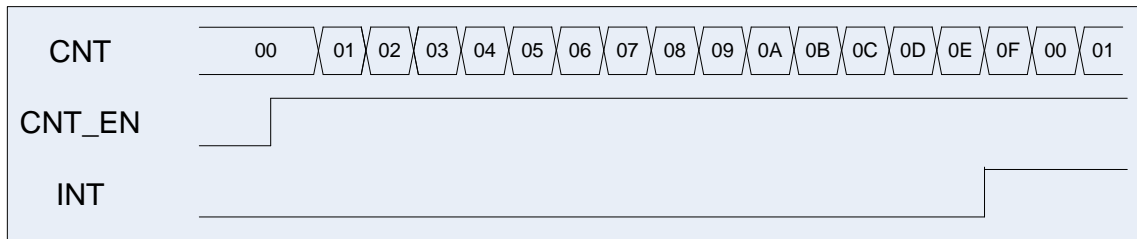


Рисунок 57 – Режим обычного счетчика

### 20.8.2 Режим захвата

```

RST_CLK->PER2_CLOCK /= 0x00004000; //Разрешение частоты PCLK для таймера 1
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
//для таймера 1 (TIM_CLK = PER1_C2)

TIMER1->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER1->CNT = 0x00000000; //Начальное значение счетчика
TIMER1->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMER1->ARR = 0x000000FF; //Основание счета
TIMER1->IE = 0x000001E0; //Разрешение генерировать прерывание по событию
//настроенного фронта на входах CH1i-CH4i

TIMER1->CH1_CNTRL0 = 0x00008000; //Захват по положительному фронту сигнала
//на входе CH1i, фильтрация отключена
TIMER1->CH2_CNTRL0 = 0x00008010; //Захват по отрицательному фронту сигнала
//на входе CH2i, фильтрация отключена
TIMER1->CH3_CNTRL0 = 0x00008001; //Захват по положительному фронту сигнала
//на входе CH3i, фильтрация выполняется
//по двум выборкам на частоте TIM_CLK
TIMER1->CH4_CNTRL0 = 0x00008011; //Захват по отрицательному фронту сигнала
//на входе CH4i, фильтрация выполняется
//по двум выборкам на частоте TIM_CLK

//Режим работы выхода канала – канал работает на вход
TIMER1->CH1_CNTRL1 = 0x00000000;
TIMER1->CH2_CNTRL1 = 0x00000000;
TIMER1->CH3_CNTRL1 = 0x00000000;
TIMER1->CH4_CNTRL1 = 0x00000000;
//Разрешение работы таймера
TIMER1->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
    
```

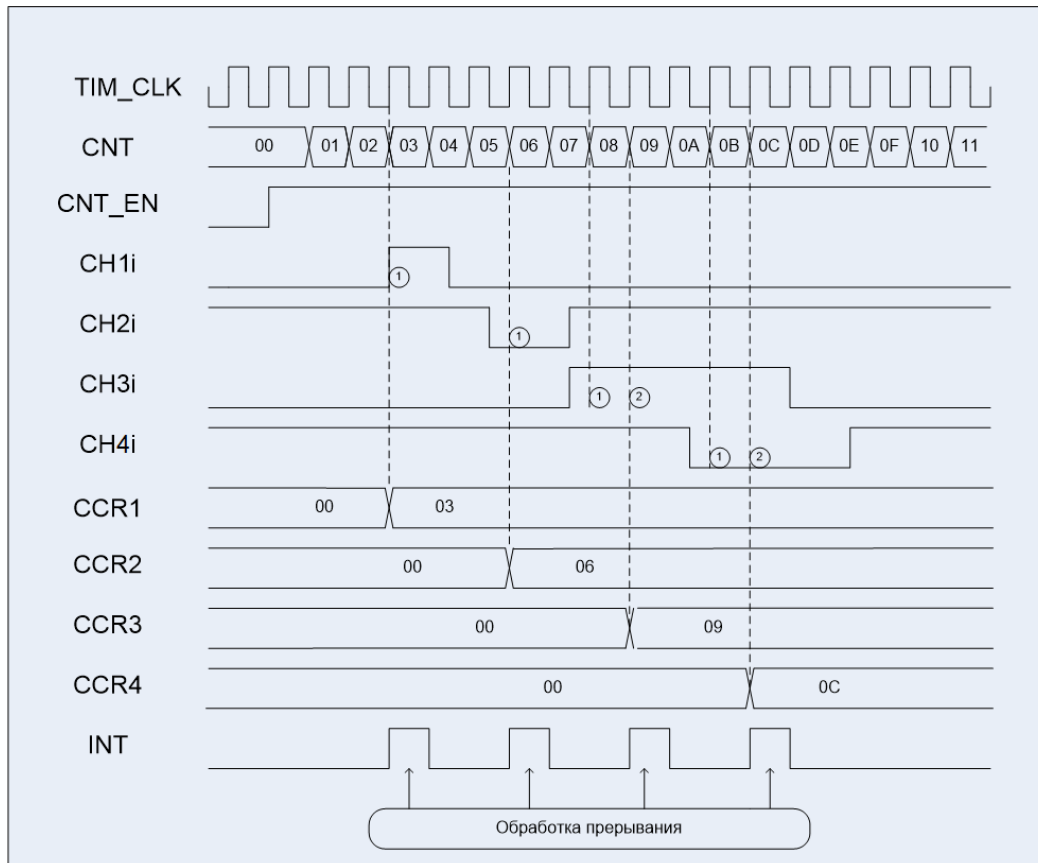


Рисунок 58 – Диаграммы примера работы в режиме захвата

### 20.8.3 Режим ШИМ

```

RST_CLK->PER2_CLOCK |= 0x00004000; //Разрешение частоты PCLK для таймера 1
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
//для таймера 1 (TIM_CLK = PER1_C2)

TIMER1->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER1->CNT = 0x00000000; //Начальное значение счетчика
TIMER1->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMER1->ARR = 0x00000010; //Основание счета
TIMER1->IE = 0x00001E00; //Разрешение генерировать прерывание по событию
//переднего фронта на выходе REF для всех каналов

//Режим работы каналов – ШИМ
TIMER1->CH1_CNTRL0 = 0x00000200; //REF = 1, если CNT == CCR
TIMER1->CH2_CNTRL0 = 0x00000200; //REF = 1, если CNT == CCR
TIMER1->CH3_CNTRL0 = 0x00000400; //REF = 0, если CNT == CCR
TIMER1->CH4_CNTRL0 = 0x00000600; //Переключение REF, если CNT == CCR
//Режим работы выхода канала – канал работает на выход,
//на выходы канала выдается сигнал REF
TIMER1->CH1_CNTRL1 = 0x00000909;
TIMER1->CH2_CNTRL1 = 0x00000909;
TIMER1->CH3_CNTRL1 = 0x00000909;
TIMER1->CH4_CNTRL1 = 0x00000909;
//Установка значений CCR, с которыми сравнивается CNT при работе в режиме ШИМ
TIMER1->CH1_CCR = 0x00000003;
    
```

```
TIMER1->CH2_CCR = 0x00000006;
TIMER1->CH3_CCR = 0x00000009;
TIMER1->CH4_CCR = 0x0000000F;
//Разрешение работы таймера
TIMER1->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
```

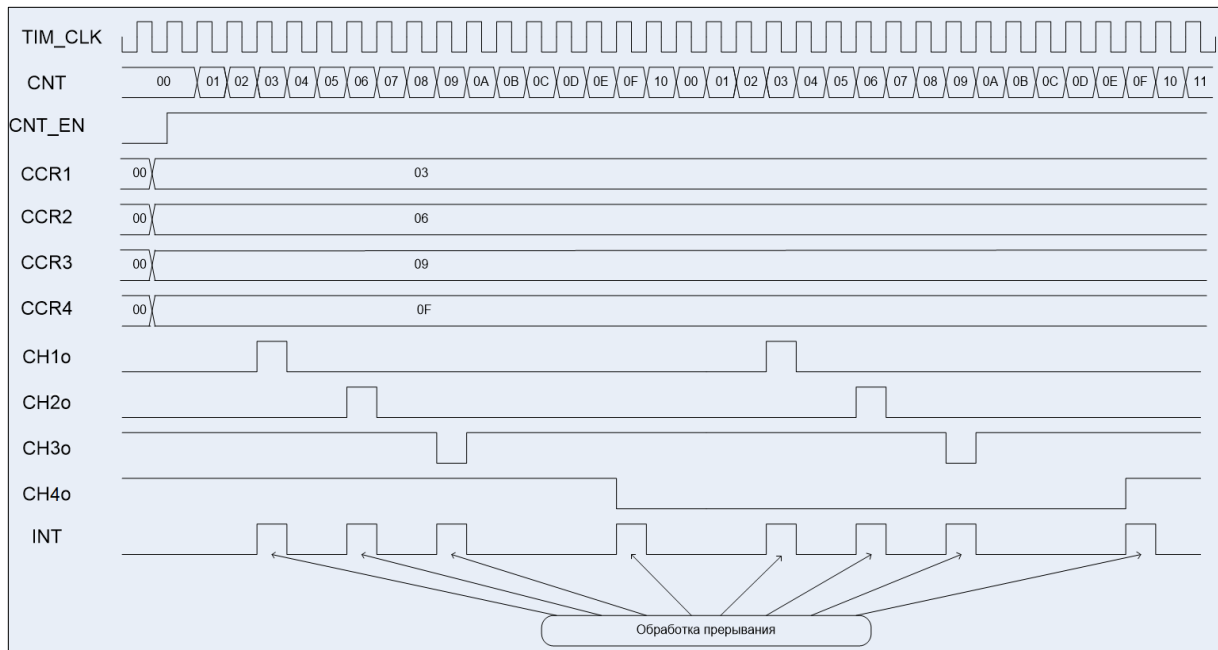


Рисунок 59 – Диаграммы примера работы в режиме ШИМ

## 20.9 Описание регистров блока таймера

Таблица 317 – Базовые адреса и смещения регистров управления таймера

Адрес	Название	Описание
0x4007_0000	TIMER1	Контроллер TIMER1
0x4007_8000	TIMER2	Контроллер TIMER2
0x400D_8000	TIMER3	Контроллер TIMER3
0x400E_0000	TIMER4	Контроллер TIMER4
Смещение		
0x00	CNT[31:0]	Основной счетчик таймера
0x04	PSG[31:0]	Делитель частоты TIM_CLK для тактирования основного счетчика
0x08	ARR[31:0]	Основание счета основного счетчика
0x0C	CNTRL[31:0]	Регистр управления основным счетчиком
0x50	BRKETR_CNTRL[31:0]	Регистр управления входом BRK и ETR
0x54	STATUS[31:0]	Регистр статуса таймера
0x58	IE[31:0]	Регистр разрешения прерываний
0x5C	DMA_RE[31:0]	Регистр разрешения формирования запроса DMA
Канал 1		
0x10	CH1_CCR[31:0]	Регистр сравнения/захвата для 1 канала таймера
0x20	CH1_CNTRL0[31:0]	Регистр управления 0 для 1 канала таймера
0x30	CH1_CNTRL1[31:0]	Регистр управления 1 для 1 канала таймера
0x40	CH1_DTG[31:0]	Регистр управления DTG для 1 канала таймера

Адрес	Название	Описание
0x60	CH1_CNTRL2[31:0]	Регистр управления 2 для 1 канала таймера
0x70	CH1_CCR1[31:0]	Регистр сравнения/захвата 1 для 1 канала таймера
Канал 2		
0x14	CH2_CCR[31:0]	Регистр сравнения/захвата для 2 канала таймера
0x24	CH2_CNTRL0[31:0]	Регистр управления 0 для 2 канала таймера
0x34	CH2_CNTRL1[31:0]	Регистр управления 1 для 2 канала таймера
0x44	CH2_DTG[31:0]	Регистр управления DTG для 2 канала таймера
0x64	CH2_CNTRL2[31:0]	Регистр управления 2 для 2 канала таймера
0x74	CH2_CCR1[31:0]	Регистр сравнения/захвата 1 для 2 канала таймера
Канал 3		
0x18	CH3_CCR[31:0]	Регистр сравнения/захвата для 3 канала таймера
0x28	CH3_CNTRL[31:0]	Регистр управления 0 для 3 канала таймера
0x38	CH3_CNTRL1[31:0]	Регистр управления 1 для 3 канала таймера
0x48	CH3_DTG[31:0]	Регистр управления DTG для 3 канала таймера
0x68	CH3_CNTRL2[31:0]	Регистр управления 2 для 3 канала таймера
0x78	CH3_CCR1[31:0]	Регистр сравнения/захвата 1 для 3 канала таймера
Канал 4		
0x1C	CH4_CCR[31:0]	Регистр сравнения/захвата для 4 канала таймера
0x2C	CH4_CNTRL[31:0]	Регистр управления 0 для 4 канала таймера
0x3C	CH4_CNTRL1[31:0]	Регистр управления 1 для 4 канала таймера
0x4C	CH4_DTG[31:0]	Регистр управления DTG для 4 канала таймера
0x6C	CH4_CNTRL2[31:0]	Регистр управления 2 для 4 канала таймера
0x7C	CH4_CCR1[31:0]	Регистр сравнения/захвата 1 для 4 канала таймера

### 20.9.1 CNT

Таблица 318 – Основной счетчик таймера CNT

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

Таблица 319 – Описание бит регистра CNT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	CNT[31:0]	Значение основного счетчика таймера

### 20.9.2 PSG

Таблица 320 – Делитель частоты TIM\_CLK для счета основного счетчика PSG

Номер	31..0
Доступ	R/W
Сброс	0
	PSG[31:0]

Таблица 321 – Описание бит регистра PSG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	PSG[31:0]	Значение предварительного делителя счетчика. Основной счетчик считает на частоте: $TIM\_CLKd = TIM\_CLK/(PSG+1)$

### 20.9.3 ARR

Таблица 322 – Основание счета основного счетчика ARR

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

Таблица 323 – Описание бит регистра ARR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	ARR[31:0]	Основание счета для основного счетчика. $CNT = [0...ARR]$

### 20.9.4 CNTRL

Таблица 324 – Регистр управления основным счетчиком CNTRL

Номер	31...12	11...8
Доступ	U	R/W
Сброс	0	0
	-	EVNT_SEL[3:0]

Номер	7, 6	5, 4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	00	00	0	0	0	0
	CNT_MODE[1:0]	FDTS[1:0]	DIR	WR_CMPL	ARRB_EN	CNT_EN

Таблица 325 – Описание бит регистра CNTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...12	-	Зарезервировано
11...8	EVNT_SEL[3:0]	Биты выбора источника событий: 0000 – внутренняя тактовая частота TIM_CLKd (формируется путем деления частоты TIM_CLK); 0001 – CNT == ARR в таймере 1; 0010 – CNT == ARR в таймере 2; 0011 – CNT == ARR в таймере 3;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
		0100 – событие переднего фронта на канале 1, «Режим 1»; 0101 – событие переднего фронта на канале 2, «Режим 1»; 0110 – событие переднего фронта на канале 3, «Режим 1»; 0111 – событие переднего фронта на канале 4, «Режим 1»; 1000 – событие переднего фронта на ETR, «Режим 2»; 1001 – событие заднего фронта на ETR, «Режим 2»; 1010 – CNT == ARR в таймере 4; 1011-1111 – зарезервировано
7, 6	CNT_MODE[1:0]	Режим счета основного счетчика: 00 – счетчик прямой при DIR = 0; счетчик обратный при DIR = 1; 01 – счетчик двунаправленный с автоматическим изменением DIR при CNT == 0 или CNT == ARR; 10 – счетчик прямой при DIR = 0; счетчик обратный при DIR = 1; 11 – зарезервировано. Режим счета CNT_MODE[1:0] необходимо устанавливать в соответствии со значением в поле EVNT_SEL[3:0]: – EVNT_SEL[3:0] = 0000: CNT_MODE[1:0] = 00 или 01; – EVNT_SEL[3:0] != 0000: CNT_MODE[1:0] = 10
5, 4	FDTS[1:0]	Делитель тактовой частоты F <sub>DTS</sub> : 00 – F <sub>DTS</sub> = TIM_CLK; 01 – F <sub>DTS</sub> = TIM_CLK/2; 10 – F <sub>DTS</sub> = TIM_CLK/3; 11 – F <sub>DTS</sub> = TIM_CLK/4
3	DIR	Направление счета основного счетчика: 0 – прямой, от 0 до ARR; 1 – обратный, от ARR до 0
2	WR_CMPL	Флаг выполнения записи нового значения в регистры CNT, PSG и ARR: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
1	ARRB_EN	Режим обновления регистра ARR: 0 – ARR будет перезаписан в момент записи в ARR; 1 – ARR будет перезаписан при CNT == ARR
0	CNT_EN	Разрешение работы таймера: 0 – таймер отключен; 1 – таймер включен

### 20.9.5 CHy\_CCR

Таблица 326 – Регистр сравнения/захвата для 'у' канала таймера CHy\_CCR

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

Таблица 327 – Описание бит регистра CHy\_CCR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	CCR[31:0]	Режим захвата: значение CNT, при котором произошел факт захвата события. Режим ШИМ: значение CCR, с которым сравнивается CNT

### 20.9.6 CHy\_CCR1

Таблица 328 – Регистр сравнения/захвата для ‘у’ канала таймера CHy\_CCR1

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

Таблица 329 – Описание бит регистра CHy\_CCR1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	CCR1[31:0]	Режим захвата: значение CNT, при котором произошел факт захвата события Режим ШИМ: значение CCR1, с которым сравнивается CNT

### 20.9.7 CHy\_CNTRL0

Таблица 330 – Регистр управления для ‘у’ канала таймера CHy\_CNTRL0

Номер	31...17	16	15	14	13
Доступ	U	RO	R/W	RO	R/W
Сброс	0	0	0	0	0
	-	WR_CMPL1	CAP_NPWM	WR_CMPL	ETR_EN

Номер	12	11...9	8	7, 6	5, 4	3...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	000	0	00	00	0000
	BRK_EN	OCCM[2:0]	OCCE	CH_PSC[1:0]	CH_SEL[1:0]	CH_FLTR[3:0]

Таблица 331 – Описание бит регистра CHy\_CNTRL0

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16	WR_CMPL1	Флаг выполнения записи нового значения в регистр CHy_CCR1: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
15	CAP_NPWM	Режим работы канала: 0 – канал работает в режиме ШИМ; 1 – канал работает в режиме захвата

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
14	WR_CMPL	Флаг выполнения записи нового значения в регистр CHy_CCR: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
13	ETR_EN	Разрешение сброса сигнала REF в «0» при высоком уровне на входе ETR: 0 – запрещен; 1 – разрешен
12	BRK_EN	Разрешение сброса сигналов REF и DTG в «0» при низком уровне на входе BRK: 0 – запрещен; 1 – разрешен
11...9	OSCM[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 прямой), CNT<CCR, иначе 0; 0, если DIR=1 (счет обратный), CNT>CCR, иначе 1; 111 – 0, если DIR=0 (счет прямой), CNT<CCR, иначе 1; 1, если DIR=1 (счет обратный), 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 – 0, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 1; 0, если DIR=1 (счет обратный), CCR<CNT<CCR1, иначе 1; 111 – 1, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 0; 1, если DIR=1 (счет обратный), CCR<CNT<CCR1, иначе 0; Необходимо соблюдать условие CCR<CCR1
8	OSCE	Разрешение работы ETR: 0 – запрещен; 1 – разрешен
7, 6	CH_PSC[1:0]	Предварительный делитель входного канала: 00 – нет деления; 01 – /2; 10 – /4; 11 – /8
5, 4	CH_SEL[1:0]	Выбор события по входному каналу CHy <sub>i</sub> для фиксации значения основного счетчика (регистр CNT) в регистр CHy_CCR:



Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		00 – положительный фронт на входном канале CHy <sub>i</sub> ; 01 – отрицательный фронт на входном канале CHy <sub>i</sub> ; 10 – положительный фронт от других каналов: для первого канала от второго канала; для второго канала от третьего канала; для третьего канала от четвертого канала; для четвертого канала от первого канала; 11 – положительный фронт от других каналов: для первого канала от третьего канала; для второго канала от четвертого канала; для третьего канала от первого канала; для четвертого канала от второго канала
3...0	CH_FLTR[3:0]	Конфигурация фильтра на входе канала 'у'. Выбор частоты выборки F <sub>S</sub> и количества выборок N: 0000 – нет фильтрации, F <sub>S</sub> = F <sub>DTS</sub> ; 0001 – F <sub>S</sub> = TIM_CLK, N = 2; 0010 – F <sub>S</sub> = TIM_CLK, N = 4; 0011 – F <sub>S</sub> = TIM_CLK, N = 8; 0100 – F <sub>S</sub> = F <sub>DTS</sub> /2, N = 6; 0101 – F <sub>S</sub> = F <sub>DTS</sub> /2, N = 8; 0110 – F <sub>S</sub> = F <sub>DTS</sub> /4, N = 6; 0111 – F <sub>S</sub> = F <sub>DTS</sub> /4, N = 8; 1000 – F <sub>S</sub> = F <sub>DTS</sub> /8, N = 6; 1001 – F <sub>S</sub> = F <sub>DTS</sub> /8, N = 8; 1010 – F <sub>S</sub> = F <sub>DTS</sub> /16, N = 5; 1011 – F <sub>S</sub> = F <sub>DTS</sub> /16, N = 6; 1100 – F <sub>S</sub> = F <sub>DTS</sub> /16, N = 8; 1101 – F <sub>S</sub> = F <sub>DTS</sub> /32, N = 5; 1110 – F <sub>S</sub> = F <sub>DTS</sub> /32, N = 6; 1111 – F <sub>S</sub> = F <sub>DTS</sub> /32, N = 8

### 20.9.8 CHy\_CNTRL1

Таблица 332 – Регистр управления 1 для 'у' канала таймера CHy\_CNTRL1

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

Таблица 333 – Описание бит регистра CHy\_CNTRL1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...13	-	Зарезервировано
12	NINV	Инверсия инверсного выхода nCHy: 0 – выход не инвертируется; 1 – выход инвертируется
11, 10	NSELO[1:0]	Выбор источника сигнала для инверсного выхода nCHy: 00 – на nCHyо выдается 0; 01 – на nCHyо выдается 1; 10 – на nCHyо выдается сигнал nREF; 11 – на nCHyо выдается сигнал с DTG
9, 8	NSELOE[1:0]	Режим работы инверсного выхода nCHy: 00 – на nCHyое выдается 0; 01 – на nCHyое выдается 1; 10 – на nCHyое выдается сигнал nREF; 11 – на nCHyое выдается сигнал с DTG. При nCHyое = 0 вывод канала в третьем состоянии, при nCHyое = 1 вывод канала работает в режиме выхода
7...5	-	Зарезервировано
4	INV	Инверсия прямого выхода CHy: 0 – выход не инвертируется; 1 – выход инвертируется
3, 2	SELO[1:0]	Выбор источника сигнала для прямого выхода CHy: 00 – на CHyо выдается 0; 01 – на CHyо выдается 1; 10 – на CHyо выдается сигнал REF; 11 – на CHyо выдается сигнал с DTG
1, 0	SELOE[1:0]	Режим работы прямого выхода CHy: 00 – на CHyое выдается 0; 01 – на CHyое выдается 1; 10 – на CHyое выдается сигнал REF; 11 – на CHyое выдается сигнал с DTG. При CHyое = 0 вывод канала работает в режиме входа, при CHyое = 1 вывод канала работает в режиме выхода

### 20.9.9 CHy\_CNTRL2

Таблица 334 – Регистр управления 2 для ‘y’ канала таймера CHy\_CNTRL2

Номер	31...5	4	3	2	1, 0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	00
	-	EVNT_DLY	CCR_RLD	CCR1_EN	CH_SEL1[1:0]

Таблица 335 – Описание бит регистра CHy\_CNTRL2

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...5	-	Зарезервировано
4	EVNT_DLY	Задержка события захвата до обновления регистров CHy_CCR и CHy_CCR1: 0 – сигнал события захвата устанавливается в момент обнаружения события, при этом обновление регистров CHy_CCR и CHy_CCR1 выполняется через один такт TIM_CLK; 1 – сигнал события захвата устанавливается синхронно с обновлением информации в регистрах CHy_CCR и CHy_CCR1
3	CCR_RLD	Режим обновления регистров CHy_CCR и CHy_CCR1: 0 – обновление возможно в любой момент времени; 1 – обновление будет осуществлено только при CNT == 0
2	CCR1_EN	Разрешение работы регистра CHy_CCR1: 0 – CHy_CCR1 не используется; 1 – CHy_CCR1 используется
1, 0	CH_SEL1[1:0]	Выбор события по входному каналу CHy <sub>i</sub> для фиксации значения основного счетчика (регистр CNT) в регистр CHy_CCR1: 00 – положительный фронт на входном канале CHy <sub>i</sub> ; 01 – отрицательный фронт на входном канале CHy <sub>i</sub> ; 10 – отрицательный фронт от других каналов: для первого канала от второго канала; для второго канала от третьего канала; для третьего канала от четвертого канала; для четвертого канала от первого канала; 11 – отрицательный фронт от других каналов: для первого канала от третьего канала; для второго канала от четвертого канала; для третьего канала от первого канала; для четвертого канала от второго канала

### 20.9.10 CHy\_DTG

Таблица 336 – Регистр управления генератором «мертвой зоны» CHy\_DTG

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

Таблица 337 – Описание бит регистра СНу\_DTG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...8	DTG[7:0]	Основной делитель частоты DTG. Задержка DTGdel = DTG • (DTGx + 1)
7...5	-	Зарезервировано.
4	EDTS	Частота работы DTG: 0 – TIM_CLK; 1 – F <sub>DTS</sub>
3...0	DTGx[3:0]	Предварительный делитель частоты DTG

### 20.9.11 BRKETR\_CNTRL

Таблица 338 – Регистр BRKETR\_CNTRL управления входом BRK и ETR

Номер	31...8	7...4	3, 2	1	0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0000	00	0	0
	-	ETR_FLTR[3:0]	ETR_PSC[1:0]	ETR_INV	BRK_INV

Таблица 339 – Описание бит регистра BRKETR\_CNTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7...4	ETR_FLTR[3:0]	Конфигурация фильтра на входе ETR. Выбор частоты выборки F <sub>s</sub> и количества выборок N: 0000 – нет фильтрации, F <sub>s</sub> = F <sub>DTS</sub> ; 0001 – F <sub>s</sub> = TIM_CLK, N = 2; 0010 – F <sub>s</sub> = TIM_CLK, N = 4; 0011 – F <sub>s</sub> = TIM_CLK, N = 8; 0100 – F <sub>s</sub> = F <sub>DTS</sub> /2, N = 6; 0101 – F <sub>s</sub> = F <sub>DTS</sub> /2, N = 8; 0110 – F <sub>s</sub> = F <sub>DTS</sub> /4, N = 6; 0111 – F <sub>s</sub> = F <sub>DTS</sub> /4, N = 8; 1000 – F <sub>s</sub> = F <sub>DTS</sub> /8, N = 6; 1001 – F <sub>s</sub> = F <sub>DTS</sub> /8, N = 8; 1010 – F <sub>s</sub> = F <sub>DTS</sub> /16, N = 5; 1011 – F <sub>s</sub> = F <sub>DTS</sub> /16, N = 6; 1100 – F <sub>s</sub> = F <sub>DTS</sub> /16, N = 8; 1101 – F <sub>s</sub> = F <sub>DTS</sub> /32, N = 5; 1110 – F <sub>s</sub> = F <sub>DTS</sub> /32, N = 6; 1111 – F <sub>s</sub> = F <sub>DTS</sub> /32, N = 8
3, 2	ETR_PSC[1:0]	Асинхронный предделитель частоты со входа ETR: 00 – без деления; 01 – /2; 10 – /4; 11 – /8

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1	ETR_INV	Инверсия входа ETR: 0 – без инверсии; 1 – инверсия
0	BRK_INV	Инверсия входа BRK: 0 – без инверсии; 1 – инверсия

### 20.9.12 STATUS

Таблица 340 – Регистр статуса таймера STATUS

Номер	31...17	16...13	12...9
Доступ	U	R/W	R/W
Сброс	0	0	0
	-	CCR_CAP1_EVNT[3:0]	CCR_REF_EVNT[3:0]

Номер	8...5	4	3
Доступ	R/W	R/W	R/W
Сброс	0	0	0
	CCR_CAP_EVNT[3:0]	BRK_EVNT	ETR_FE_EVNT

Номер	2	1	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
	ETR_RE_EVNT	CNT_ARR_EVNT	CNT_ZERO_EVNT

Таблица 341 – Описание бит регистра STATUS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...13	CCR_CAP1_EVNT[3:0]	Событие записи значения счетчика CNT в регистр СНу_CCR1 по захвату настроенного фронта на входе канала СНуi: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал; Бит 3 – четвертый канал
12...9	CCR_REF_EVNT[3:0]	Событие переднего фронта на выходе генератора опорного сигнала REF: 0 – нет события; 1 – есть событие.

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал; Бит 3 – четвертый канал
8...5	CCR_CAP_EVNT[3:0]	Событие записи значения счетчика CNT в регистр СНу_CCR по захвату настроенного фронта на входе канала СНуi: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал; Бит 3 – четвертый канал
4	BRK_EVNT	Событие высокого уровня на входе BRK: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», при условии наличия низкого уровня на входе BRK
3	ETR_FE_EVNT	Событие заднего фронта на входе ETR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
2	ETR_RE_EVNT	Событие переднего фронта на входе ETR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
1	CNT_ARR_EVNT	Событие совпадения CNT с ARR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса флага регистры CNT и ARR не изменили состояния, то флаг повторно не взводится
0	CNT_ZERO_EVNT	Событие совпадения CNT с нулем: 0 – нет события; 1 – есть событие.

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		Сбрасывается записью «0», если запись происходит одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса флага регистр CNT не изменил состояния, то флаг повторно не взводится

### 20.9.13 IE

Таблица 342 – Регистр разрешения прерываний таймера IE

Номер	31...17	16...13	12...9
Доступ	U	R/W	R/W
Сброс	0	0	0
	-	CCR_CAP1_EVNT_IE[3:0]	CCR_REF_EVNT_IE[3:0]

Номер	8...5	4	3
Доступ	R/W	R/W	R/W
Сброс	0	0	0
	CCR_CAP_EVNT_IE[3:0]	BRK_EVNT_IE	ETR_FE_EVNT_IE

Номер	2	1	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
	ETR_RE_EVNT_IE	CNT_ARR_EVNT_IE	CNT_ZERO_EVNT_IE

Таблица 343 – Описание бит регистра IE

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...13	CCR_CAP1_EVNT_IE[3:0]	Флаг разрешения прерывания по событию CCR_CAP1_EVNT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
12...9	CCR_REF_EVNT_IE[3:0]	Флаг разрешения прерывания по событию CCR_REF_EVNT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8...5	CCR_CAP_EVNT_IE[3:0]	Флаг разрешения прерывания по событию CCR_CAP_EVNT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
4	BRK_EVNT_IE	Флаг разрешения прерывания по событию BRK_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
3	ETR_FE_EVNT_IE	Флаг разрешения прерывания по событию ETR_FE_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
2	ETR_RE_EVNT_IE	Флаг разрешения прерывания по событию ETR_RE_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
1	CNT_ARR_EVNT_IE	Флаг разрешения прерывания по событию CNT_ARR_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
0	CNT_ZERO_EVNT_IE	Флаг разрешения прерывания по событию CNT_ZERO_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено

### 20.9.14 DMA\_RE

Таблица 344 – Регистр DMA\_RE разрешения запроса DMA

Номер	31...17	16...13	12...9
Доступ	U	R/W	R/W
Сброс	0	0	0
	-	CCR_CAP1_EVNT_RE[3:0]	CCR_REF_EVNT_RE[3:0]

Номер	8...5	4	3
Доступ	R/W	R/W	R/W
Сброс	0	0	0
	CCR_CAP_EVNT_RE[3:0]	BRK_EVNT_RE	ETR_FE_EVNT_RE

Номер	2	1	0
Доступ	R/W	R/W	R/W
Сброс	0	0	0
	ETR_RE_EVNT_RE	CNT_ARR_EVNT_RE	CNT_ZERO_EVNT_RE



Таблица 345 – Описание бит регистра DMA\_RE

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано.
16...13	CCR_CAP1_EVNT_RE[3:0]	Флаг разрешения запроса DMA по событию CCR_CAP1_EVNT[3:0]: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – первый канал; Бит 3 – четвертый канал
12...9	CCR_REF_EVNT_RE[3:0]	Флаг разрешения запроса DMA по событию CCR_REF_EVNT[3:0]: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – первый канал; Бит 3 – четвертый канал
8...5	CCR_CAP_EVNT_RE[3:0]	Флаг разрешения запроса DMA по событию CCR_CAP_EVNT[3:0]: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – первый канал; Бит 3 – четвертый канал
4	BRK_EVNT_RE	Флаг разрешения запроса DMA по событию BRK_EVNT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
3	ETR_FE_EVNT_RE	Флаг разрешения запроса DMA по событию ETR_FE_EVNT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
2	ETR_RE_EVNT_RE	Флаг разрешения запроса DMA по событию ETR_RE_EVNT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
1	CNT_ARR_EVNT_RE	Флаг разрешения запроса DMA по событию CNT_ARR_EVNT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
0	CNT_ZERO_EVNT_RE	Флаг разрешения запроса DMA по событию CNT_ZERO_EVNT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен

## 21 Контроллер АЦП

В микросхеме реализован 10-разрядный АЦП. С помощью АЦП можно оцифровать сигнал с трех внешних аналоговых выводов и двух внутренних каналов, на которые выводится датчик температуры и источник опорного напряжения. Скорость выборки составляет до 500 тысяч преобразований в секунду.

Контроллер АЦП позволяет:

- оцифровать один из трех внешних каналов;
- оцифровать значение встроенного датчика температуры;
- оцифровать значение встроенного источника опорного напряжения;
- оцифровать значение напряжения батарейного домена;
- осуществить автоматический опрос заданных каналов;
- выработать прерывание при выходе оцифрованного значения за заданные пределы.

Для осуществления преобразования требуется не менее 25 тактов синхронизации  $C\_ADCS$ . В качестве синхросигнала может выступать поделенная частота периферийного блока PCLKd или 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 бит  $Cfg\_REG\_DIVCLK[3:0]$ . Максимальная частота CLK не может превышать 28 МГц.

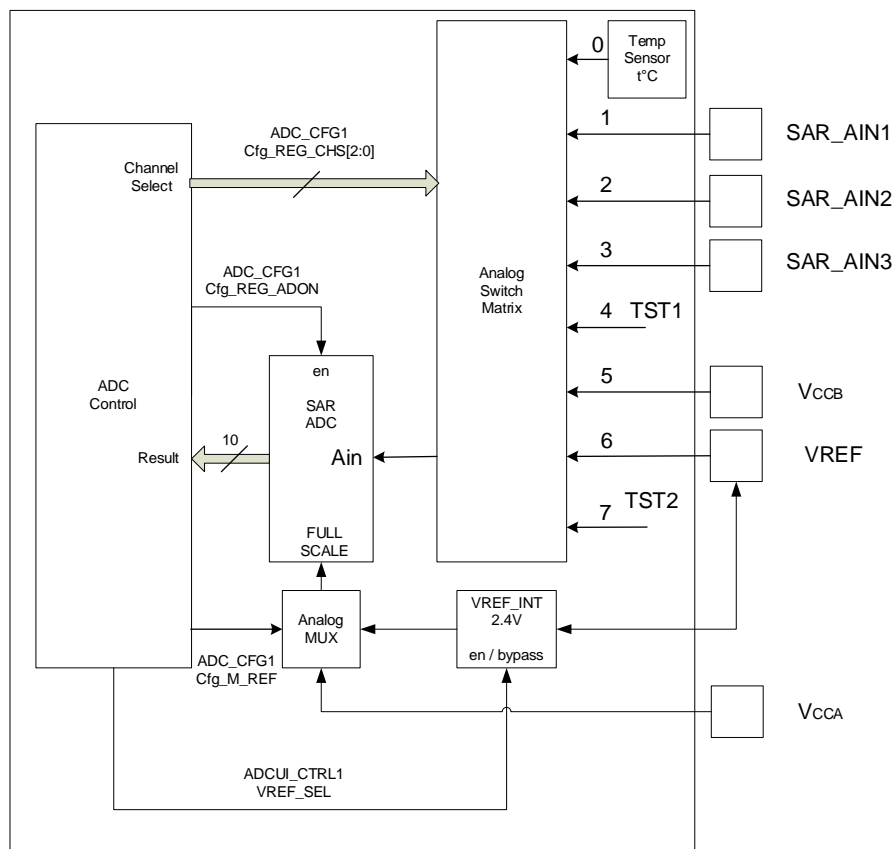


Рисунок 60 – Структурная схема АЦП

Для включения АЦП необходимо установить бит  $Cfg\_REG\_ADON$ .

## 21.1 Преобразование внешнего канала

В регистре ADC1\_CFG в битах Cfg\_REG\_CHS[2:0] необходимо задать соответствующий выводу номер канала. Преобразование может осуществляться при опоре равной  $U_{CCA}$  (бит Cfg\_M\_REF = 0) и внешней (Cfg\_M\_REF = 1), в этом случае опора берется с вывода VREF. Биты Cfg\_REG\_CHCH, Cfg\_REG\_RNGC, Cfg\_REG\_SAMPLE должны быть сброшены.

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 ADCx\_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.

## 21.2 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. Выставление данных бит необходимо обеспечить до установки конфигурации АЦП, то есть до записи в регистр ADCx\_CFG. Преобразование может осуществляться при внутренней опоре (бит Cfg\_M\_REF = 0) и внешней (Cfg\_M\_REF = 1), в этом случае опора берется с выводов VREF. Бит Cfg\_REG\_RNGC должен быть сброшен, а Cfg\_REG\_SAMPLE и Cfg\_REG\_CHCH должны быть установлены. С помощью бит Delay\_GO можно задать паузу между преобразованиями при переборе каналов. При включенном АЦП (Cfg\_REG\_ADON=1) и установленном бите Cfg\_REG\_SAMPLE=1 процесс преобразования запускается автоматически и значение бита Cfg\_REG\_GO контролируется аппаратно.

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 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[2:0] 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.

### 21.3 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, когда результат преобразования выходит за границы выставляется флаг Flg\_REG\_AWOIFEN. А в регистре результата будет полученное значение.

Результат преобразования вычисляется по формуле

$$\text{Напряжение на входе АЦП} = \frac{\text{напряжение полной шкалы АЦП} \cdot \text{значение регистра } ADC1\_RESULT}{1024} \quad (13)$$

### 21.4 Датчик температуры

С помощью АЦП можно осуществить преобразования датчика температуры. Для выбора датчика температуры в качестве источника для преобразования необходимо в битах Cfg\_REG\_CHS установить значение 0 канала. 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 температуры можно в регистре ADC1\_CHSEL выбрать только 0 канал и установить бит Cfg\_REG\_CHCH в «1», либо установить номер 0-го канала в битах Cfg\_REG\_CHS[2:0] и сбросить бит Cfg\_REG\_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала.

Параметры температурного датчика не регламентируются. В зависимости от необходимой точности может быть достаточно провести градуировку в двух–трех точках. При необходимости более точных измерений необходимо построить градуировочную таблицу.

Градуировка производится индивидуально для каждой микросхемы.

### 21.4.1 Формула расчета температуры

Для определения значения температуры кристалла в градусах Цельсия следует использовать формулу:

$$TEMP = \frac{\text{напряжение полной шкалы АЦП} \cdot \text{значение регистра } ADC1\_RESULT - 1,403}{1024} - 25 - \frac{\quad}{0,0037} \quad (14)$$

Например, при опросе 0 канала АЦП выдает код = 625, в качестве полной шкалы при этом выбран внешний источник 2,5 В, подключенный ко входу микросхемы VREF. Тогда температура кристалла равна

$$TEMP = 25 - \frac{2,5 \cdot 625}{1024} - 1,403}{0,0037} = -8,2 \text{ } ^\circ\text{C} . \quad (15)$$

### 21.5 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 соотношением 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.

$$R_{AIN} < \frac{T_{track}}{C_{ADC} \cdot \ln(2^N)} - R_{ADC} , \quad (16)$$

где  $C_{ADC}$  – внутренняя емкость АЦП (~ 2 – 3 пФ);

$N$  – требуемая точность в разрядах;

$R_{ADC}$  – входное сопротивление тракта АЦП (~1500 Ом);

$T_{track}$  – время заряда внутренней емкости в тактах, определяется по формуле

$$T_{track} = 4 \cdot T_{C\_ADCS} + N_{PCLKd} \cdot T_{PCLKd} = \frac{4}{f_{C\_ADCS}} + \frac{(ConvDelay)}{f_{PCLKd}} , \quad (17)$$

где  $ConvDelay$  – дополнительная задержка перед началом преобразования, определяется битами DelayGo[2:0] исходя из таблиц 346 – 348;

$f_{C\_ADCS}$  – частота следования импульсов тактовых сигналов SAR АЦП;

$f_{PCLKd}$  – определяется формулой

$$f_{PCLKd} = \frac{f_{PCLK}}{2^{Cfg\_REG\_DIVCLK}} . \quad (18)$$

Если необходимо обеспечить преобразование с точностью 10 разрядов  $\pm 1$  LSB, то  $N = 10$ . Время заряда  $T_{\text{track}}$  определяется битами DelayGo[2:0] и схемой самого АЦП и представлено в таблицах 346 – 348. Время заряда  $T_{\text{track}}$  должно составлять не менее 1 мкс.

### 21.6 Время заряда внутренней емкости АЦП и время преобразования

Таблица 346 – Время заряда внутренней емкости АЦП и время преобразования. CfgREGCLKS = C\_ADCS/PCLKd, C\_ADCS=PCLKd=PCLK

DelayGo[2:0]	Дополнительная задержка перед началом преобразования	Общее время $T_{\text{track}}$ заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	1×PCLKd	4×C_ADCS+1×PCLKd	25×C_ADCS+1×PCLKd
001	3×PCLKd	4×C_ADCS+3×PCLKd	25×C_ADCS+3×PCLKd
010	3×PCLKd	4×C_ADCS+3×PCLKd	25×C_ADCS+3×PCLKd
011	5×PCLKd	4×C_ADCS+5×PCLKd	25×C_ADCS+5×PCLKd
100	5×PCLKd	4×C_ADCS+5×PCLKd	25×C_ADCS+5×PCLKd
101	7×PCLKd	4×C_ADCS+7×PCLKd	25×C_ADCS+7×PCLKd
110	7×PCLKd	4×C_ADCS+7×PCLKd	25×C_ADCS+7×PCLKd
111	9×PCLKd	4×C_ADCS+9×PCLKd	25×C_ADCS+9×PCLKd

Таблица 347 – Время заряда внутренней емкости АЦП и время преобразования. CfgREGCLKS = PCLKd

DelayGo[2:0]	Дополнительная задержка перед началом преобразования	Общее время $T_{\text{track}}$ заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	0×PCLKd	4×C_ADCS+0×PCLKd	25,5×C_ADCS+2×PCLK+0×PCLKd
001	0×PCLKd	4×C_ADCS+0×PCLKd	25,5×C_ADCS+2×PCLK+0×PCLKd
010	2×PCLKd	4×C_ADCS+2×PCLKd	25,5×C_ADCS+2×PCLK+2×PCLKd
011	2×PCLKd	4×C_ADCS+2×PCLKd	25,5×C_ADCS+2×PCLK+2×PCLKd
100	4×PCLKd	4×C_ADCS+4×PCLKd	25,5×C_ADCS+2×PCLK+4×PCLKd
101	4×PCLKd	4×C_ADCS+4×PCLKd	25,5×C_ADCS+2×PCLK+4×PCLKd
110	6×PCLKd	4×C_ADCS+6×PCLKd	25,5×C_ADCS+2×PCLK+6×PCLKd
111	6×PCLKd	4×C_ADCS+6×PCLKd	25,5×C_ADCS+2×PCLK+6×PCLKd

Таблица 348 – Время заряда внутренней емкости АЦП и время преобразования. CfgREGCLKS = C\_ADCS

DelayGo [2:0]	Дополнительная задержка перед началом преобразования	Общее время Ttrack заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	1×PCLKd	4×C_ADCS+1×PCLKd	25×C_ADCS+1×PCLKd +1,5×PCLK
001	2×PCLKd	4×C_ADCS+2×PCLKd	25×C_ADCS+2×PCLKd+1,5×PCLK
010	3×PCLKd	4×C_ADCS+3×PCLKd	25×C_ADCS+3×PCLKd+1,5×PCLK
011	4×PCLKd	4×C_ADCS+4×PCLKd	25×C_ADCS+4×PCLKd+1,5×PCLK
100	5×PCLKd	4×C_ADCS+5×PCLKd	25×C_ADCS+5×PCLKd+1,5×PCLK
101	6×PCLKd	4×C_ADCS+6×PCLKd	25×C_ADCS+6×PCLKd+1,5×PCLK
110	7×PCLKd	4×C_ADCS+7×PCLKd	25×C_ADCS+7×PCLKd+1,5×PCLK
111	8×PCLKd	4×C_ADCS+8×PCLKd	25×C_ADCS+8×PCLKd+1,5×PCLK

Помимо точности определяемой временем зарядки внутренней емкости АЦП точность преобразования имеет ошибки, связанные с технологическими разбросами схемы и шумами и определяемые параметрами E<sub>DLADC</sub>, E<sub>LADC</sub> и E<sub>OFFADC</sub>.

Для корректного задания режимов работы АЦП в регистре ADCx\_CFG необходимо сделать до задания бита Cfg REG GO, иначе новая конфигурация будет действовать со следующего преобразования.

## 21.7 Описание регистров блока контроллера АЦП

Таблица 349 – Описание регистров блока контроллера АЦП

Базовый Адрес	Название	Описание
0x4004_0000	ADC	Контроллер ADC
Смещение		
0x00	ADC1_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	ADC_TRIM	Регистр настройки термодатчика

### 21.7.1 ADC1\_CFG

Таблица 350 – Регистр ADC1\_CFG

Номер	31...28	27...25	24...20	19
Доступ	U	R/W	U	R/W
Сброс	0	0	0	0
	-	Delay Go[2:0]	-	en_noise_rng

Номер	18..16	15...12	11	10	9
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	-	Cfg REG DIVCLK[3:0]	Cfg M_REF	Cfg REG RNGC	Cfg REG CHCH

Номер	8, 7	6...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 REG CHS[2:0]	Cfg REG SAMPLE	Cfg REG CLKS	Cfg REG GO	Cfg REG ADON

Таблица 351 – Описание бит регистра ADC1\_CFG

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...25	Delay Go[2:0]	Задержка перед началом следующего преобразования после завершения предыдущего при последовательном переборе каналов: см. таблицы 346 – 348
24...20	-	Должны быть в нуле для правильного функционирования
19	en_noise_rng	Бит включения rng генераторов для генерации шума по основному питанию микросхемы 1 – включено; 0 – выключено
18..16	-	Должно быть 010 для правильного функционирования
15...12	Cfg REG DIVCLK[3:0]	Выбор коэффициента деления частоты периферийного блока: 0000 – PCLKd = PCLK; 0001 – PCLKd = PCLK/2; 0010 – PCLKd = PCLK/4; 0011 – PCLKd = PCLK/8; ... 1011 – PCLKd = PCLK/2048 Остальные PCLKd = PCLK
11	Cfg M_REF	Выбор шкалы АЦП: 0 – шкала АЦП от 0 до напряжения U <sub>ССА</sub> ; 1 – шкала АЦП от 0 до напряжения VREF
10	Cfg REG RNGC	Разрешение автоматического контролирования уровней: 1 – разрешено: выработка прерывания при выходе за диапазон в регистрах границы обработки; 0 – не разрешено



Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
9	Cfg REG CHCH	Выбор переключения каналов: 1 – переключение включено (перебираются каналы, выбранные в регистре выбора канала Sl_Ch_Ch_REF); 0 – используется только выбранный канал
8, 7	-	Должны быть в нуле для правильного функционирования
6...4	Cfg REG CHS[2:0]	Выбор аналогового канала, по которому поступает сигнал для преобразования: 111 – 0 канал (термодатчик); 110 – 1 канал (подключение к SAR_AIN1); 101 – 2 канал (подключение к SAR_AIN2); 100 – 3 канал (подключение к SAR_AIN3); 011 – 4 канал (тестовый режим); 010 – 5 канал (подключение к VDD_BD); 001 – 6 канал (подключение к VREF); 000 – 7 канал (тестовый режим)
3	Cfg REG SAMPLE	Выбор способа запуска АЦП: 1 – последовательный: автоматический запуск после завершения предыдущего преобразования; 0 – одиночный
2	Cfg REG CLKS	Выбор источника синхросигнала C_ADCS работы ADC: 1 – ADC_CLK (формируется от выбранной частоты, подробнее см. раздел 16 «Сигналы тактовой частоты»); 0 – PCLKd (формируется от частоты ядра, подробнее см. формулу (18))
1	Cfg REG GO	Начало преобразования Запись «1» начинает процесс преобразования, сбрасывается автоматически. Запись при Cfg_REG_SAMPLE = 1 не влияет на работу АЦП, бит устанавливается и сбрасывается автоматически
0	Cfg REG ADON	Включение АЦП: 1 – включено; 0 – выключено

### 21.7.2 ADC1\_H\_LEVEL

Таблица 352 – Регистр ADC1\_H\_LEVEL

Номер	31...10	9...0
Доступ	U	R/W
Сброс	0	0
	-	REG H LEVEL [9:0]

Таблица 353 – Описание бит регистра ADC1\_H\_LEVEL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	-	Зарезервировано
9...0	REG H LEVEL [9:0]	Верхняя граница зоны допуска

### 21.7.3 ADC1\_L\_LEVEL

Таблица 354 – Регистр ADC1\_L\_LEVEL

Номер	31...10	9...0
Доступ	U	R/W
Сброс	0	0
		REG L LEVEL [9:0]

Таблица 355 – Описание бит регистра ADC1\_L\_LEVEL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..10	-	Зарезервировано
9...0	REG L LEVEL [9:0]	Нижняя граница зоны допуска

### 21.7.4 ADC1\_RESULT

Таблица 356 – Регистр ADC1\_RESULT

Номер	31...19	18...16	15...10	9...0
Доступ	U	RO	U	RO
Сброс	0	0	0	0
	-	CHANNEL [2:0]	-	RESULT[9:0]

Таблица 357 – Описание бит регистра ADC1\_RESULT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...19	-	Зарезервировано
18...16	CHANNEL [2:0]	Канал результата преобразования: 111 – 0 канал (термодатчик); 110 – 1 канал (подключение к SAR_AIN1); 101 – 2 канал (подключение к SAR_AIN2); 100 – 3 канал (подключение к SAR_AIN3); 011 – 4 канал (тестовый режим); 010 – 5 канал (подключение к VDD_BD); 001 – 6 канал (подключение к VREF); 000 – 7 канал (тестовый режим)
15...10	-	Зарезервировано
9...0	RESULT[9:0]	Значение результата преобразования

### 21.7.5 ADC1\_STATUS

Таблица 358 – Регистр ADC1\_STATUS

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
	-	ECOIF_IE	AWOIF_IE	Flg REG EOCIF	Flg REG AWOIFEN	Flg REG OVERWRITE

Таблица 359 – Описание бит регистра ADC1\_STATUS

Разряды	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 – прерывание не генерируется; 1 – прерывание генерируется
3	AWOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_AWOIFEN: 0 – прерывание не генерируется; 1 – прерывание генерируется
2	Flg REG EOCIF	Флаг выставляется, когда закончено преобразование 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 контроля уровней. Сбрасывается только при записи нуля в данный бит. 0 – результат в допустимой зоне; 1 – вне допустимой зоны
0	Flg REG OVERWRITE	Данные в регистре результата были перезаписаны. Сбрасывается только при записи нуля в данный бит. 0 – не было события перезаписи не считанного результата; 1 – был результат преобразования, который не был считан

### 21.7.6 ADC1\_CHSEL

Таблица 360 – Регистр ADC1\_CHSEL

Номер	31...8	7...0
Доступ	U	R/W
Сброс	0	0
	-	SI_Ch_Ch_REF[7:0]

Таблица 361 – Описание бит регистра ADC1\_CHSEL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Должны быть в нуле для правильного функционирования
7...0	SI_Ch_Ch_REF[7:0]	Выбор каналов автоматического перебора: 0 – в соответствующем бите канал не участвует в переборе; 1 – в соответствующем бите канал участвует в переборе. Номера каналов, участвующих в переборе, соответствует номерам установленных бит в регистре

### 21.7.7 ADC\_TRIM

Таблица 362 – Регистр ADC\_TRIM

Номер	31..6	5...1	0
Доступ	U	R/W	U
Сброс	0	10000	0
	-	BG_TRIM[4:0]	-

Таблица 363 – Описание бит регистра ADC\_TRIM

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...6	-	Зарезервировано
5...1	BG_TRIM[4:0]	Подстройка опорного напряжения bandgap. Рекомендуется не изменять значение бит установленных по умолчанию при сбросе
0	-	Зарезервировано

## 22 Контроллер интерфейса I2C

I2C является двухпроводным, двунаправленным последовательным каналом связи с простым и эффективным методом обмена данными между устройствами. Интерфейс применяется, когда надо организовать обмен на коротком расстоянии между несколькими устройствами. Стандарт интерфейса I2C является многомастерным с обнаружением коллизий и арбитражем, исключающим потерю данных при обмене, когда два или более мастера пытаются осуществить передачу одновременно. Контроллер интерфейса I2C в микросхеме работает только в режиме Master.

Интерфейс работает на трех скоростях:

- нормальная 100 Kbps;
- быстрая 400 Kbps;
- очень быстрая 1 Mbps.

Приблизительная скорость обмена данными блоком I2C рассчитывается по формуле

$$F_{scl} = \frac{HCLK}{5 \cdot (DIV + 1)} \quad (19)$$

Более точное значение скорости обмена можно установить опытным путем, значение делителя DIV настраивается в регистрах PRL (младшая часть) и PRH (старшая часть) – если примерно рассчитанное значение делителя DIV не задействует регистр PRH, то он должен быть равен нулю.

### 22.1 Конфигурация системы

I2C системы используют последовательную линию данных SDA и линию тактового сигнала SCL. Все устройства, подсоединенные к этим двум линиям, должны работать в режиме открытого стока, обеспечивая тем самым создание на линии «проводного И» за счет внешних резисторов подтяжки обеих линий к питанию.

Передача данных между мастером и ведомым осуществляется по линии SDA и синхронизируется по линии SCL. После завершения передачи информации осуществляется передача в обратную сторону одного бита подтверждения. Каждый принимаемый бит фиксируется принимающей стороной при высоком уровне SCL и может изменяться передатчиком при низком уровне. Изменение линии SDA при высоком уровне SCL является командным состоянием (см. «Сигнал START» и «Сигнал STOP»).

### 22.2 Протокол I2C

Нормальная передача по интерфейсу I2C содержит четыре фазы:

- сигнал START;
- передача адреса;
- передача данных;
- сигнал STOP.

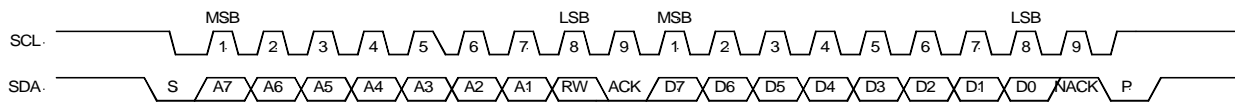


Рисунок 61 – Передача по I2C

### 22.2.1 Сигнал START

Когда шина находится в свободном состоянии, т.е. не одно из устройств не осуществляет передачи (на линиях SCL и SDA высокий уровень), мастер может инициализировать процесс передачи через создание сигнала START на линии. Сигнал START или S бит задается, когда уровень на линии SDA переходит из высокого в низкий при высоком уровне на линии SCL. Появление сигнала START не означает начала передачи данных.

Повторный сигнал START является обычным сигналом START, но без предварительно сгенерированного до этого сигнала STOP. Мастер может использовать метод для начала соединения с другим ведомым или с тем же ведомым, но с изменением режима работы (например, чтение после записи, или, наоборот) без перевода шины в свободное состояние.

Контроллер интерфейса генерирует сигнал START при записи единицы в бит START регистра I2C\_CMD при установленных битах RD или WR. В зависимости от состояния линии SCL генерируется либо сигнал START, либо повторный сигнал START.

### 22.2.2 Передача адреса

Первым байтом данных, передаваемым мастером сразу после сигнала START, является адрес ведомого. Это семибитный адрес и следующий за ним бит RW. Бит RW определяет дальнейшее направление передачи данных. В системе на одной шине не может быть несколько ведомых устройств с одним адресом. Ведомое устройство, у которого совпадает адрес с адресом в сообщении, подтверждает прием, выставляя ACK и опуская линию SDA в низкий уровень на девятый SCL тактовый импульс. Контроллер также поддерживает 10-битный адрес путем генерации двух циклов передачи адреса.

Процесс выдачи адреса выполняется как цикл записи. Необходимо записать адрес ведомого в регистр I2C\_TXD и установить бит WR в регистре I2C\_CMD. Контроллер осуществит передачу адреса в линию.

### 22.2.3 Передача данных

После успешного подтверждения приема адреса одним ведомым устройством может быть начата передача данных в направлении, задаваемым битом RW в посылке мастера. Каждый передаваемый бит подтверждается ACK на 9-й SCL тактовый импульс. Если ведомое устройство выдало NACK (нет подтверждения), то мастер может сгенерировать либо сигнал STOP для прекращения передачи, либо повторный сигнал START для начала нового цикла передачи.

Если мастер является принимающим устройством и выдает NACK, то ведомое устройство отпускает линию SDA и мастер может сгенерировать сигнал STOP или повторный сигнал START.

Для записи данных в ведомое устройство запишите данные в регистр I2C\_TXD и установите бит WR. Для чтения данных из устройства установите бит RD. На время выполнения передачи контроллер интерфейса выставляет флаг TR\_PROG в регистре I2C\_STA. Когда передача завершена, этот флаг снимается и устанавливается флаг INT. Если при этом установлен бит разрешения INT\_EN, то генерируется прерывание контроллеру прерываний. Регистр I2C\_RXD содержит корректные принятые данные после установки флага INT. Пользователь может начать новый цикл чтения или записи только тогда, когда флаг TR\_PROG сброшен.

### 22.2.4 Сигнал STOP

Мастер может завершить соединение путем создания сигнала STOP. Сигнал STOP или P бит определяется переходом линии SDA из низкого состояния в высокое, когда SCL находится в высоком состоянии.

## 22.3 Описание регистров контроллера I2C

Таблица 364 – Описание регистров контроллера I2C

Базовый Адрес	Название	Описание
0x4003_0000	I2C	Контроллер I2C
Смещение		
0x00	PRL	Младшая часть предделителя частоты
0x04	PRH	Старшая часть предделителя частоты
0x08	CTR	Управление контроллером I2C
0x0C	RXD	Принятые данные по I2C
0x10	STA	Статус I2C
0x14	TXD	Передаваемые данные по I2C
0x18	CMD	Управление I2C

### 22.3.1 PRL

Таблица 365 – Регистр PRL

Номер	31...8	7... 0
Доступ	U	R/W
Сброс	0	0xFF
	-	PR[7:0]

Таблица 366 – Описание бит регистра PRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7...0	PR[7:0]	Младшая часть предделителя

### 22.3.2 PRH

Таблица 367 – Регистр PRH

Номер	31...8	7... 0
Доступ	U	R/W
Сброс	0	0xFF
	-	PR[15:8]

Таблица 368 – Описание бит регистра PRH

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7...0	PR[15:8]	Старшая часть предделителя

### 22.3.3 CTR

Таблица 369 – Регистр CTR

Номер	31...8	7	6	5	4...0
Доступ	U	R/W	R/W	R/W	U
Сброс	0	0	0	0	0
	-	EN_I2C	EN_INT	S_I2C	-

Таблица 370 – Описание бит регистра CTR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7	EN_I2C	Разрешение работы контроллера I2C: 0 – выключен; 1 – включен
6	EN_INT	Разрешение прерывания от I2C: 0 – запрещено; 1 – разрешено
5	S_I2C	Скорость работы I2C: 0 – до 400 кГц; 1 – до 1 МГц
4...0	-	Зарезервировано

### 22.3.4 RXD

Таблица 371 – Регистр RXD

Номер	31...8	7... 0
Доступ	U	R/O
Сброс	0	0
	-	RXD[7:0]

Таблица 372 – Описание бит регистра RXD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7...0	RXD[7:0]	Последний полученный по I2C байт



### 22.3.5 STA

Таблица 373 – Регистр STA

Номер	31...8	7	6	5	4...2	1	0
Доступ	U	R/O	R/O	R/O	U	R/O	R/O
Сброс	0	0	0	0	0	0	0
	-	Rx ACK	BUSY	LOST ARB	-	TR PROG	INT

Таблица 374 – Описание бит регистра STA

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7	Rx ACK	Полученный от ведомого ACK: 0 – ACK получен; 1 – получен NACK
6	BUSY	Состояние шины I2C: 0 – после получения Stop bit; 1 – после получения состояния Start bit
5	LOST ARB	Потеря арбитража: 0 – нет потери арбитража; 1 – потерян арбитраж. Этот бит выставляется если: – получен Stop bit, но он не был инициализирован этим контроллером; – контроллер пытается выставить SDA в высокий уровень, но SDA остается в низком
4...2	-	Зарезервировано
1	TR PROG	Процесс передачи: 0 – передача завершена; 1 – передаются данные
0	INT	Флаг прерывания, выставляется всегда. Прерывание для процессора выдается, если есть флаг EN_INT: 0 – нет прерывания; 1 – есть прерывание. Флаг выставляется если: – передача байта завершена; – был потерян арбитраж

### 22.3.6 TXD

Таблица 375 – Регистр TXD

Номер	31...8	7... 0
Доступ	U	R/W
Сброс	0	0
	-	TXD[7:0]

Таблица 376 – Описание бит регистра TXD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7...0	TXD[7:0]	Байт для отправки по I2C. При передаче адреса нулевой бит определяет режим передачи: 0 – запись в ведомое устройство; 1 – чтение из ведомого устройства

### 22.3.7 CMD

Таблица 377 – Регистр CMD

Номер	31...8	7	6	5	4	3	2, 1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W	U	R/W
Сброс	0	0	0	0	0	0	0	0
	-	START	STOP	RD	WR	ACK	-	CLR INT

Таблица 378 – Описание бит регистра CMD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7	START	Отправить START bit. Инициализируется записью 1. После завершения отправки автоматически не сбрасывается, очищается записью нуля
6	STOP	Отправить STOP bit. Инициализируется записью 1. После завершения отправки автоматически не сбрасывается, а очищается записью нуля
5	RD	Чтение из ведомого: 0 – нет действия; 1 – начать чтение
4	WR	Запись в ведомого; 0 – нет действия; 1 – начать запись
3	ACK	Отправить ACK: 0 – отправить ACK; 1 – отправить NACK
2, 1	-	Зарезервировано
0	CLR INT	Очистить прерывание INT. Запись 1 очищает прерывание

## 23 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 (с ревизии 2 – восемь 32-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-разрядных ячеек данных (с ревизии 2 – восемь 32-разрядных ячеек данных)).

Модуль формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов FIFO приемника или передатчика;
- переполнение буфера FIFO приемника;
- наличие данных в буфере FIFO приемника по истечении времени таймаута;
- наличие данных в FIFO приемника (доступно с ревизии 2);
- отсутствие данных в FIFO передатчика (доступно с ревизии 2);
- отсутствие данных в сдвиговом регистре передатчика (доступно с ревизии 2).

Основные сведения о модуле представлены в следующих подразделах:

- Характеристики интерфейса SPI;
- Характеристики интерфейса Microwire;
- Характеристики интерфейса SSI.

### 23.1 Основные характеристики модуля SSP

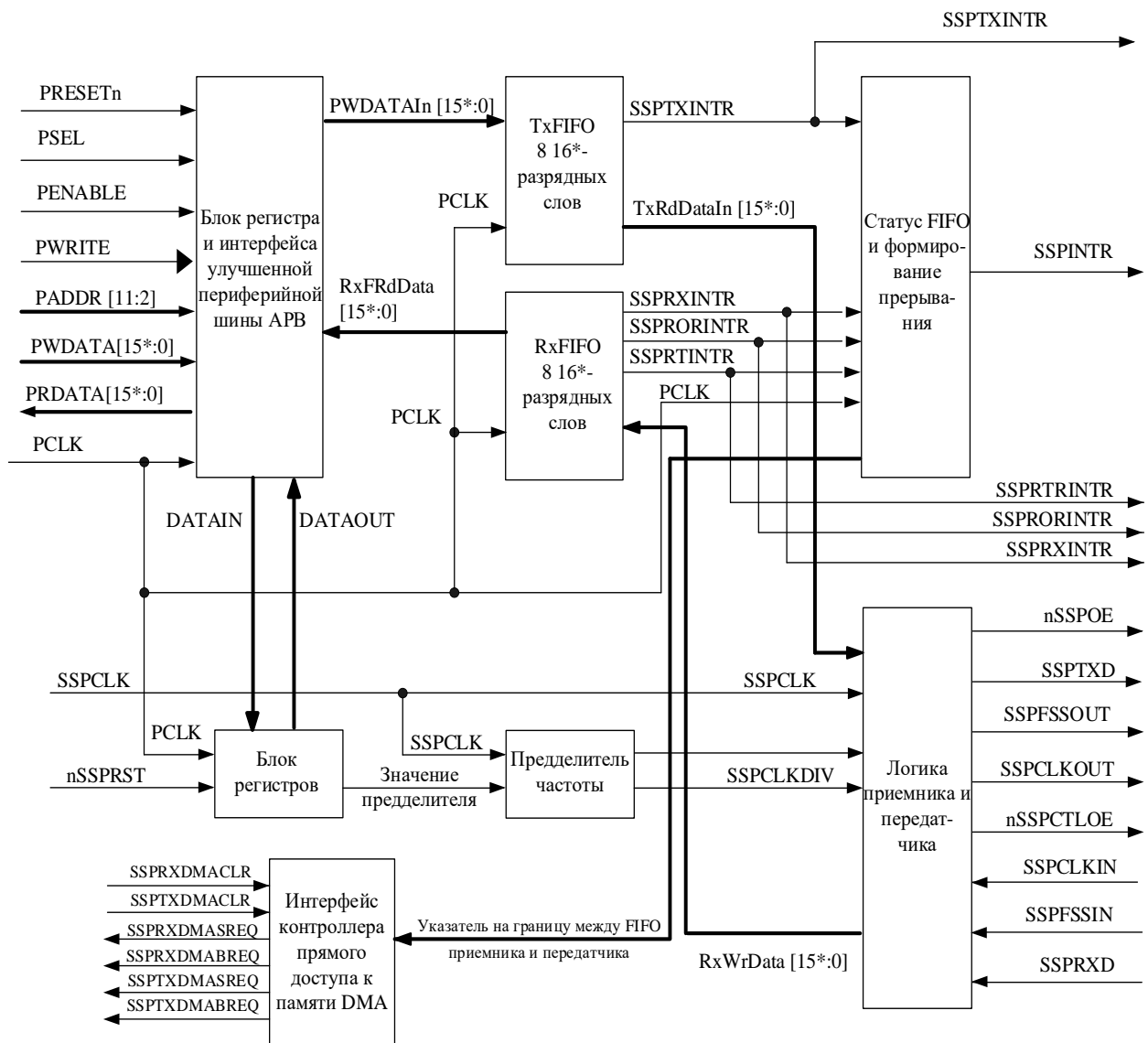
- Функционирует как в ведущем, так и в ведомом режиме;
- Программное управление скоростью обмена;
- Состоит из независимых буферов приема и передачи (восемь ячеек по 16 бит (с ревизии 2 – по 32 бита)) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);
- Программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;
- Программируемая длительность информационного кадра от 4 до 16 бит (с ревизии 2 – по 32 бита);
- Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по переполнению буфера приемника (в ревизии 1);

– Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по переполнению буфера приемника, по наличию данных в FIFO приемника, по отсутствию данных в FIFO передатчика, а также по отсутствию данных в сдвиговом регистре передатчика (в ревизии 2);

– Доступна возможность тестирования по шлейфу, соединяющему вход с выходом;

– Поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена на рисунке 62.



\* Разрядность повышена с 16 бит до 32 бит с ревизии 2.

Рисунок 62 – Структурная схема модуля SSP

### 23.1.1 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 (с ревизии 2 – до 32 бит);
- сброс FIFO передатчика (доступно с ревизии 2);
- маскирование прерываний.

### 23.1.2 Характеристики интерфейса SPI

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии;
- программное задание фазы и полярности тактового сигнала.

### 23.1.3 Характеристики интерфейса Microwire

Интерфейс Microwire фирмы National Semiconductor обеспечивает:

- полудуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

### 23.1.4 Характеристики интерфейса SSI

Интерфейс SSI фирмы Texas Instruments обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии;
- возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

## 23.2 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 в 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 8-ми слов данных шириной до 16 бит (с ревизии 2 – до 32 бит) независимо для режимов приема и передачи.

Последовательные данные передаются по линии SSPTXD и принимаются с линии SSPRXD.

Модуль SSP содержит программируемые делители частоты, формирующие тактовый сигнал обмена данными SSPCLKOUT (сигнал, поступающий на линию SSPCKOUT и предназначенный для синхронизации приема и передачи данных) из сигнала, поступающего на линию SSPCLK (сигнал, который подается на модуль SSP с блока формирования тактовых частот). 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 следующие независимо маскируемые 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;
- SSPRNEINTR – наличие данных в буфере FIFO приемника (доступно с ревизии 2);
- SSPTFEINTR – отсутствие данных в буфере FIFO передатчика (доступно с ревизии 2)
- SSPTNBSYINTR – отсутствие данных в сдвиговом регистре передатчика (доступно с ревизии 2).

Кроме того, формируется общий сигнал прерывания SSPINTR, возникающий в случае активности одного из вышеуказанных независимых немаскированных прерываний, который идет на контроллер PLIC.

Модуль также формирует сигналы запроса на прямой доступ к памяти (DMA) для совместной работы с контроллером DMA.

**Важно!** Примечание – по сбросу каждый блок SSP микросхемы направляет запросы к контроллеру DMA, что в том числе приводит к срабатыванию ложного прерывания в блоке PLIC. При работе с DMA для снятия запросов со стороны блоков SSP рекомендуется включить их тактирование в блоке тактовых частот, регистре PER2\_CLOCK.

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).

### 23.2.1 Блок формирования тактового сигнала

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSPCLK 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 CPSR. 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 SSPCLK.

Коэффициент деления второго делителя задается программно в диапазоне от 1 до 256 путем записи соответствующего значения в регистр управления CR0.

### **23.2.2 Буфер FIFO передатчика**

Буфер передатчика имеет ширину 16 бит (с ревизии 2 – 32 бита), глубину – восемь 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.

В случае работы контроллера SPI в режиме SLAVE, чтение пустого FIFO приводит к выдаче некорректных данных.

### **23.2.3 Буфер FIFO приемника**

Буфер приемника имеет ширину 16 бит (с ревизии 2 – 32 бита), глубину – восемь 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.

### **23.2.4 Блок приема и передачи данных**

#### **23.2.4.1 Режим ведущего устройства**

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSPCLK 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 SSPCLKOUT, передается по линии SSPTXD к подключенным ведомым устройствам.

Блок приемника выполняет преобразование данных, поступающих синхронно с линии SSPRXD, из последовательной в параллельную форму. После этого загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

#### **23.2.4.2 Режим ведомого устройства**

В режиме ведомого устройства тактовый сигнал обмена данными формируется одним из подключенных к модулю периферийных устройств и поступает по линии SSPCLKIN.

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 SSPTXD.

Аналогично, блок приемника выполняет преобразование данных, поступающих с линии SSPRXD синхронно с сигналом SSPCLKIN, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

### 23.2.5 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 подается на контроллер прерываний PLIC, при этом появляется дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

### 23.3 Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

### 23.4 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.

### 23.5 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 (с ревизии 2 – 32-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 SSPTXD и SSPRXD.

### 23.6 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.

$$F_{SSPCLK} \leq F_{PCLK} \cdot$$



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.

В режиме ведомого устройства сигнал SSPCLKIN от ведущего внешнего устройства поступает на схемы синхронизации, задержки и обнаружения фронта. Для того чтобы обнаружить фронт сигнала SSPCLKIN, необходимо три такта сигнала SSPCLK. Сигнал SSPTXD имеет меньшее время установки по отношению к заднему фронту SSPCLKIN, по которому и происходит считывание данных из линии. Время установки и удержания сигнала SSPRXD по отношению к сигналу SSPCLKIN 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 SSPCLKIN.

Выбор частоты тактового сигнала SSPCLK должен обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала SSPCLK к максимальной частоте сигнала SSPCLKOUT 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  $254 \times 256$ .

Минимальная допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:

$$\begin{cases} F_{SSPCLK} (min) \geq 2 \cdot F_{SSPCLKOUT} (max) \text{ [режим ведущего]}, \\ F_{SSPCLK} (min) \geq 12 \cdot F_{SSPCLKIN} (max) \text{ [режим ведомого]}. \end{cases}$$

При активации сигнала SSPFRX (доступно в ревизии 2) в регистре управления CR0 в режиме SPI это соотношение приобретает вид:

$$\begin{cases} F_{SSPCLK} (min) \geq 2 \cdot F_{SSPCLKOUT} (max) \text{ [режим ведущего]}, \\ F_{SSPCLK} (min) \geq 4 \cdot F_{SSPCLKIN} (max) \text{ [режим ведомого]}. \end{cases}$$

Аналогично, максимальная допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:

$$\begin{cases} F_{SSPCLK} (max) \leq 254 \cdot 256 \cdot F_{SSPCLKOUT} (min) \text{ [режим ведущего]}, \\ F_{SSPCLK} (max) \leq 254 \cdot 256 \cdot F_{SSPCLKIN} (min) \text{ [режим ведомого]}. \end{cases}$$

### 23.7 Программирование регистра управления CR0

Регистр CR0 предназначен для:

- установки скорости информационного обмена;
- выбора одного из трех протоколов обмена данными;
- выбора размера слова данных;
- активации быстрого режима работы ведомого устройства SPI (доступно с ревизии 2).

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 CR0 и значением поля CPSDVSR (clock prescale divisor value – коэффициент деления тактового сигнала) регистра CPSR.

Формат информационного кадра задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS регистра CR0.

Сигнал SSPFRX (доступно с ревизии 2) используется для активации у ведомого модуля SPI быстрого режима работы и включения синхронизации сигнала SSPRXD.

Для протокола SPI фирмы Motorola также задаются полярность и фаза сигнала (биты SPN и SPO).

### 23.8 Программирование регистра управления CR1

Регистр CR1 предназначен для:

- выбора ведущего или ведомого режима функционирования приемопередатчика;
- включения режима проверки канала по шлейфу;
- разрешения или запрещения работы модуля;
- программного сброса FIFO передатчика (доступно с ревизии 2).

Выбор ведущего режима осуществляется путем записи «0» в поле MS регистра CR1 (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 SSPTXD осуществляется путем установки бита SOD (slave mode SSPTXD output disable – запрет линии SSPTXD 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).

Для программного сброса FIFO передатчика необходимо установить в «1» бит RESTxFIFO (доступно с ревизии 2). После сброса сигнал RESTxFIFO автоматически сбросится в «0».

### 23.9 Формирование тактового сигнала обмена данными

Тактовый сигнал обмена данными формируется путем деления частоты тактового сигнала SSPCLK. На первом этапе формирования частота этого сигнала делится на четный коэффициент CPSDVR, 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 программирования через CR0.

Частота выходного тактового сигнала обмена данными SSPCLK определяется следующим соотношением

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \cdot (1 + SCR)} \quad (20)$$

Например, в случае, если частота сигнала SSPCLK составляет 3,6864 МГц, а значение CPSDVR = 2, частота сигнала SSPCLK лежит в интервале от 7,2 кГц до 1,8432 МГц.

### 23.10 Формат информационного кадра

Каждый информационный кадр содержит в зависимости от запрограммированного значения от 4 до 16 бит (с ревизии 2 – до 32 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 SSPCLK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SSPCLK в неактивное состояние используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

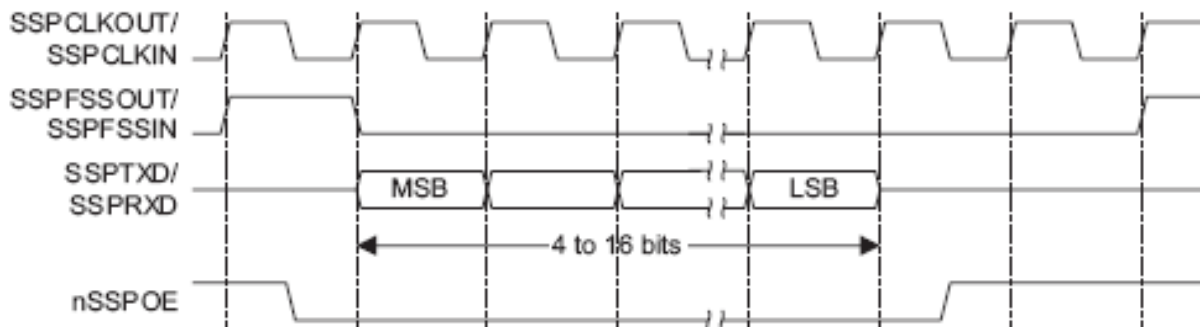
В режимах SPI и Microwire выходной сигнал кадровой синхронизации передатчика SSPFSS имеет активный низкий уровень и поддерживается в низком уровне в течение всего периода передачи информационного кадра.

В режиме построения кадра SSI фирмы Texas Instruments перед началом каждого информационного кадра на выходе SSPFSS формируется импульс с длительностью, равной одному тактовому интервалу обмена данными. В этом режиме приемопередатчик SSP, равно как и ведомые периферийные устройства, передает данные в линию по переднему фронту сигнала SSPCLK, 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 (с ревизии 2 – до 32 бит), таким образом общая длительность информационного кадра составляет от 13 до 25 бит (с ревизии 2 – до 41 бита).

### 23.10.1 Формат синхронного обмена SSI фирмы Texas Instruments



\* До 32 бит с ревизии 2.

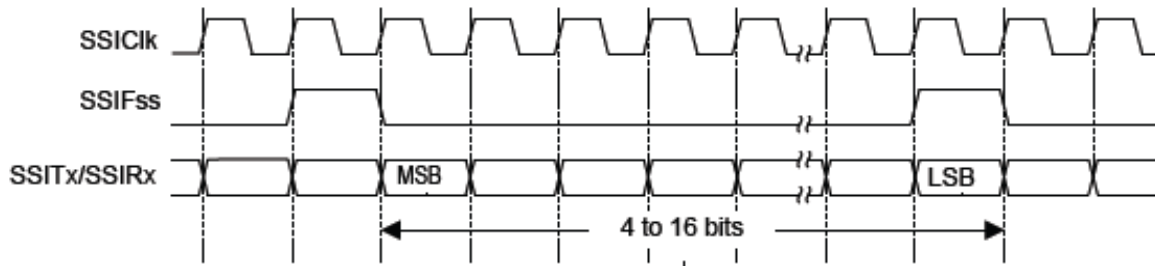
Рисунок 63 – Формат синхронного обмена протокола SSI (единичный обмен)

В данном режиме при неактивном приемопередатчике SSP сигналы SSPCLK и SSPFSS переводятся в низкий логический уровень, а линия передачи данных SSPTXD 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 SSPCLK. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSPCLK старший значащий разряд информационного кадра (4 – 16 бит (с ревизии 2 – 32 бит) данных) выдается на выход линии SSPTXD 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 SSPCLK. Принятые данные переносятся из сдвигового регистра в буфер FIFO после загрузки в него младшего значащего бита данных по очередному переднему фронту сигнала SSPCLK.

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments показывает рисунок 63 (передача единичного информационно кадра) и рисунок 64 (передача последовательности кадров).



\* До 32 бит с ревизии 2.

Рисунок 64 – Формат синхронного обмена протокола SSI (непрерывный обмен)

### 23.10.2 Формат синхронного обмена SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSPFSS выполняет функцию выбора ведомого устройства. Главной особенностью протокола SPI является возможность выбора состояния и фазы сигнала SSPCLK в режиме ожидания (неактивном приемопередатчике) путем задания значений бит SPO и SPH регистра управления CR0.

#### 23.10.2.1 Выбор полярности тактового сигнала – бит SPO

Если бит SPO равен 0, то в режиме ожидания линия SSPCLK переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SSPCLK переводится в высокий логический уровень.

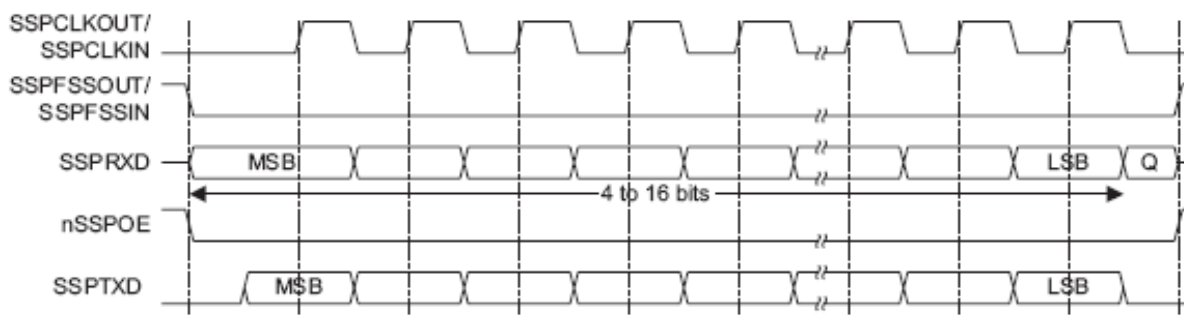
#### 23.10.2.2 Выбор фазы тактового сигнала – бит SPH

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 по нарастающему фронту тактового сигнала, в противном случае по спадающему фронту.

### 23.10.3 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0

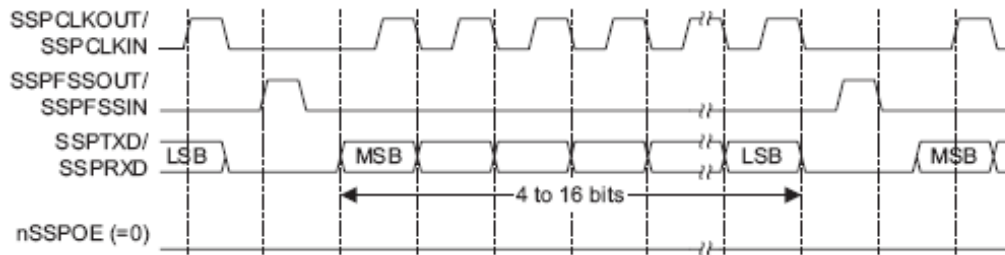
Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 0, SPH = 0 показывает рисунок 65 (одиночный обмен) и рисунок 66 (непрерывный обмен).



\* До 32 бит с ревизии 2.

Рисунок 65 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (одиночный обмен)

Примечание – На рисунке 65 буквой Q обозначен сигнал с неопределенным уровнем.



\* До 32 бит с ревизии 2.

Рисунок 66 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLK имеет низкий логический уровень;
- сигнал SSPFSS имеет высокий логический уровень;
- сигнал SSPTXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. При этом линия передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLK на линии SSPTXD 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 SSPCLK переводится в высокий логический уровень.

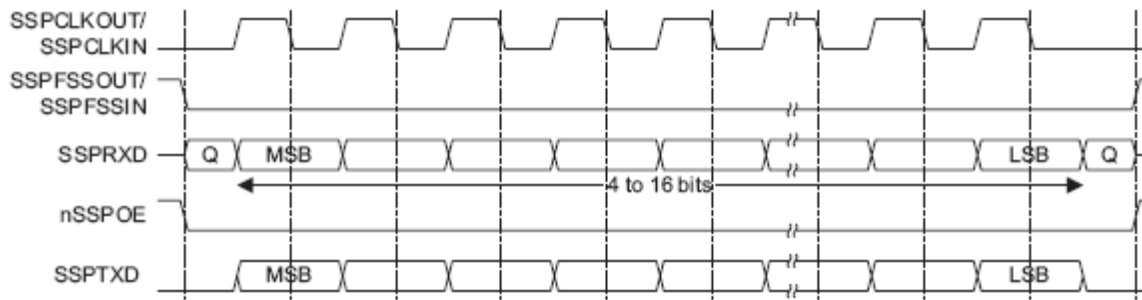
Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSPCLK.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLK.

В режиме непрерывной передачи данных на линии SSPFSS 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 SSPFSS в высокий уровень по окончании передачи каждого кадра, разрешая запись новых данных. По окончании приема последнего бита блока данных линия SSPFSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLK.

#### 23.10.4 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=1 показывает рисунок 67 (одиночный и непрерывный обмен).



\* До 32 бит с ревизии 2.

Рисунок 67 – Формат синхронного обмена протокола SPI, SPO=0, SPH=1

Примечание – На рисунке 67 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLK имеет низкий логический уровень;
- сигнал SSPFSS имеет высокий логический уровень;
- сигнал SSPTXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLK на линиях обмена, как ведущего, так и ведомого устройств будут сформированы значения первых бит передаваемых данных. В это же время включается линия SSPCLK и на ней формируется передний фронт сигнала.

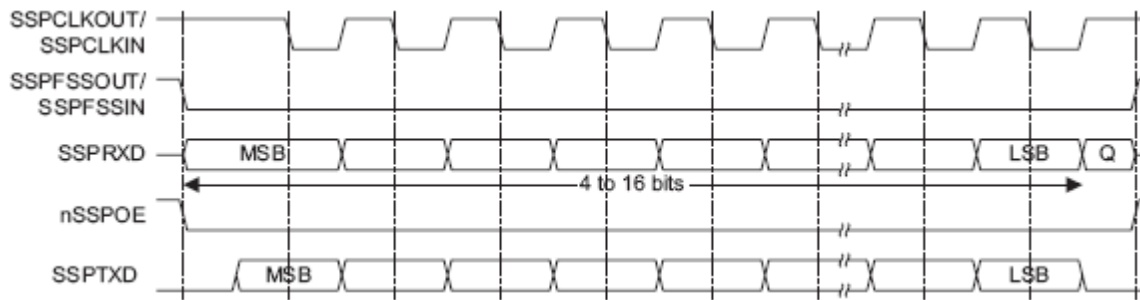
Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSPCLK.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLK.

В режиме непрерывной передачи данных линия SSPFSS 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.

### 23.10.5 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0

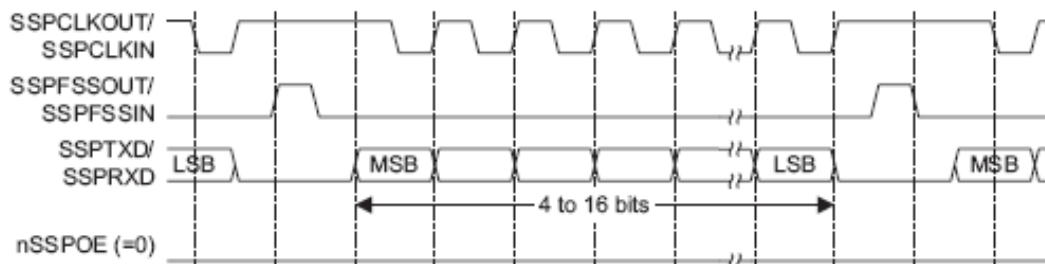
Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 1, SPH = 0 показывает рисунок 68 (одиночный обмен) и рисунок 69 (непрерывный обмен).



\* До 32 бит с ревизии 2.

Рисунок 68 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (одиночный обмен)

Примечание – На рисунке 68 буквой Q обозначен сигнал с неопределенным уровнем.



\* До 32 бит с ревизии 2.

Рисунок 69 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLK имеет высокий логический уровень;
- сигнал SSPFSS имеет высокий логический уровень;
- сигнал SSPTXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLK, на линии SSPTXD 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 SSPCLK переводится в низкий логический уровень.

Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSPCLK.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLK.

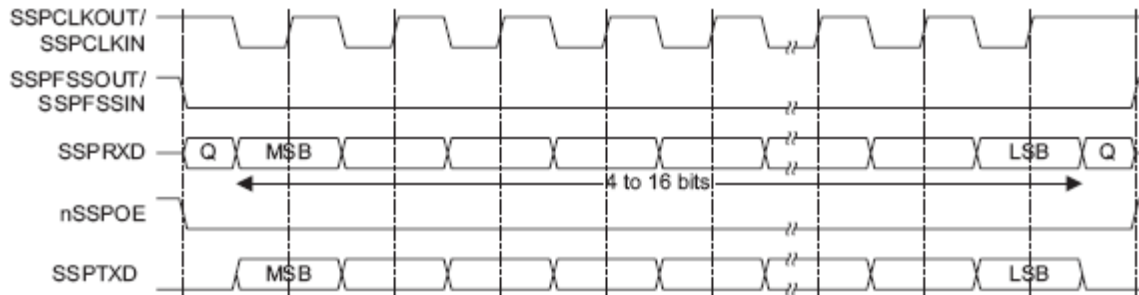
В режиме непрерывной передачи данных на линии SSPFSS должны формироваться импульсы высокого логического уровня между передачами каждого из



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 SSPFSS в высокий уровень по окончании передачи каждого кадра, разрешая запись новых данных. По окончании приема последнего бита блока данных линия SSPFSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLK.

### 23.10.6 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 1, SPH = 1 показывает рисунок 70 (одиночный и непрерывный обмен).



\* До 32 бит с ревизии 2.

Рисунок 70 – Формат синхронного обмена протокола SPI, SPO=1, SPH=1

Примечание – На рисунке 70 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLK имеет высокий логический уровень;
- сигнал SSPFSS имеет высокий логический уровень;
- сигнал SSPTXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутака сигнала SSPCLK на линиях обмена, как ведущего, так и ведомого устройств сформированы значения первых бит передаваемых данных. В это же время включается линия SSPCLK и на ней формируется передний фронт сигнала.

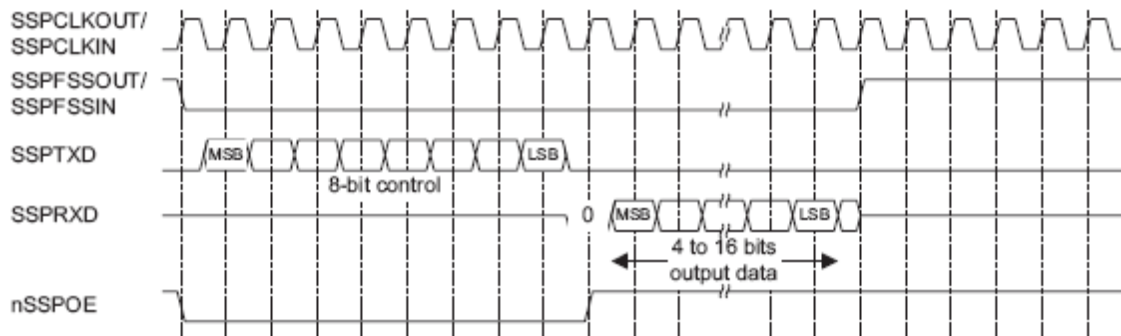
Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSPCLK.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLK.

В режиме непрерывной передачи данных линия SSPFSS 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.

### 23.10.7 Формат синхронного обмена Microwire фирмы National Semiconductor

Временные диаграммы последовательного синхронного обмена в режиме Microwire показывает рисунок 71 (одиночный обмен) и рисунок 72 (непрерывный обмен).



\* До 32 бит с ревизии 2.

Рисунок 71 – Формат синхронного обмена протокола Microwire (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 (с ревизии 2 – до 32 бит), таким образом, общая длительность информационного кадра составляет от 13 до 25 бит (с ревизии 2 – до 41 бита).

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLK имеет низкий логический уровень;
- сигнал SSPFSS имеет высокий логический уровень;
- сигнал SSPTXD переводится в высокоимпедансное состояние.

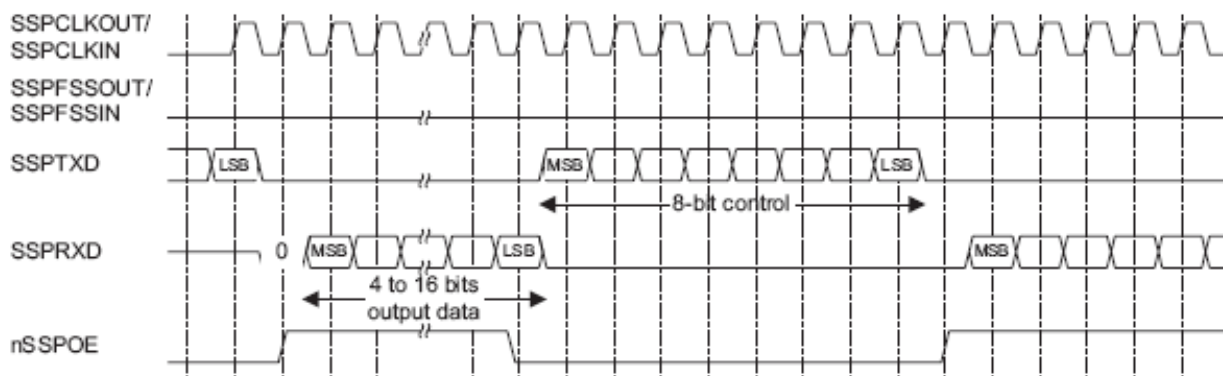
Переход в режим информационного обмена происходит после записи управляющего байта в буфер FIFO передатчика. По заднему фронту сигнала SSPFSS данные из буфера переносятся в регистр сдвига блока передатчика, откуда, начиная со старшего значащего разряда, последовательно выдаются в линию SSPTXD. Линия SSPFSS остается в низком логическом уровне в течение всей передачи кадра. Линия SSPRXD при этом находится в высокоимпедансном состоянии.

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSPCLK. 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 SSPRXD по заднему фронту сигнала SSPCLK. Ведущее устройство в свою очередь регистрирует их по переднему фронту этого тактового сигнала. В случае

одионого информационного обмена по окончании приема последнего бита слова данных сигнал SSPFSS 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 SSPCLK после приема последнего бита слова данных, либо после перевода линии SSPFSS в высокий логический уровень.

Непрерывный обмен данными начинается и заканчивается также, как и одионый обмен. Однако линия SSPFSS 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 SSPCLK.



\* До 32 бит с ревизии 2.

Рисунок 72 – Формат синхронного обмена протокола Microwire (непрерывный обмен)

### 23.10.7.1 Требования к временным параметрам сигнала SSPFSS относительно тактового сигнала SSPCLK в режиме Microwire

Модуль SSP, работающий в режиме Microwire как ведомое устройство, регистрирует данные по переднему фронту сигнала SSPCLK после установки сигнала SSPFSS в низкий логический уровень. Ведущие устройства, формирующие сигнал SSPCLK, должны гарантировать достаточное время установки и удержания сигнала SSPFSS по отношению к переднему фронту сигнала SSPCLK.

Данные требования иллюстрирует рисунок 73. По отношению к переднему фронту сигнала SSPCLK, по которому осуществляется регистрация данных в приемнике ведомого модуля SSP, время установки сигнала SSPFSS должно быть, как минимум в два раза больше периода SSPCLK, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSPCLK должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.

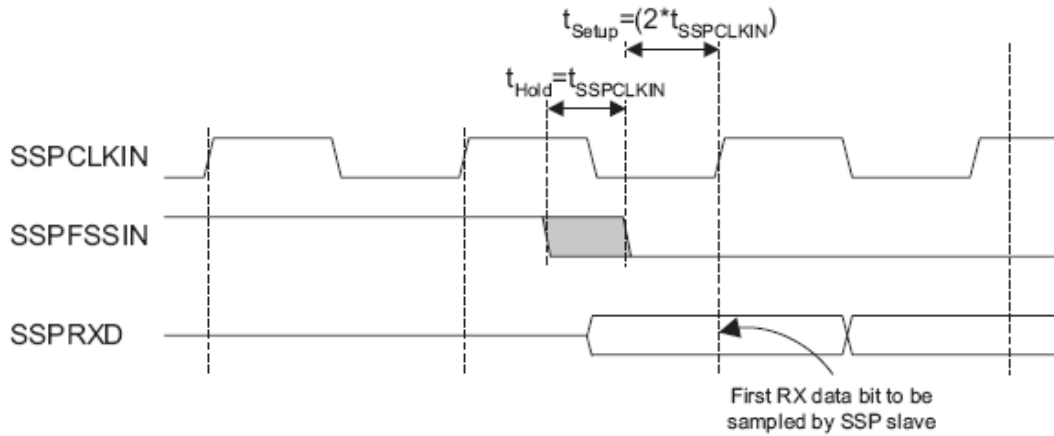


Рисунок 73 – Формат Microwire, требования к времени установки и удержания сигнала SSPFSSIN

### 23.11 Примеры конфигурации модуля в ведущем и ведомом режимах

Рисунки 74 – 76 показывают варианты подключения модуля SSP 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.

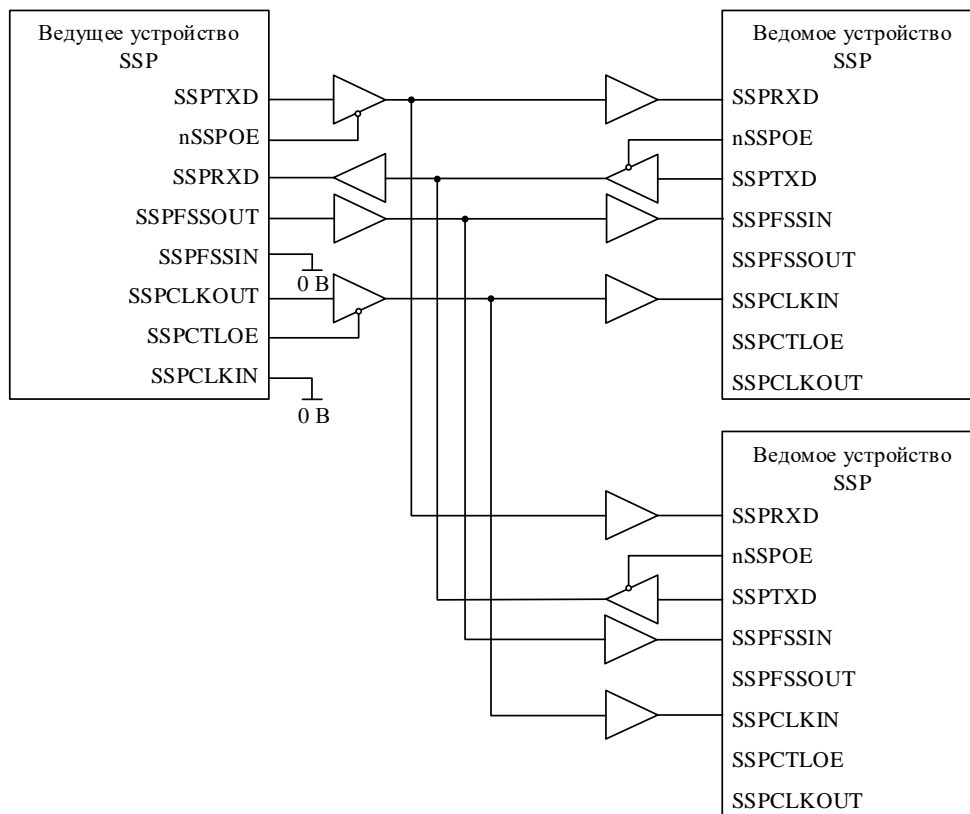


Рисунок 74 – Ведущее устройство SSP подключено к двум ведомым

Рисунок 74 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 SSPTXD.

Для ответной передачи данных один из ведомых модулей разрешает прохождение сигнала от своей линии SSPTXD на вход SSPRXD ведущего.

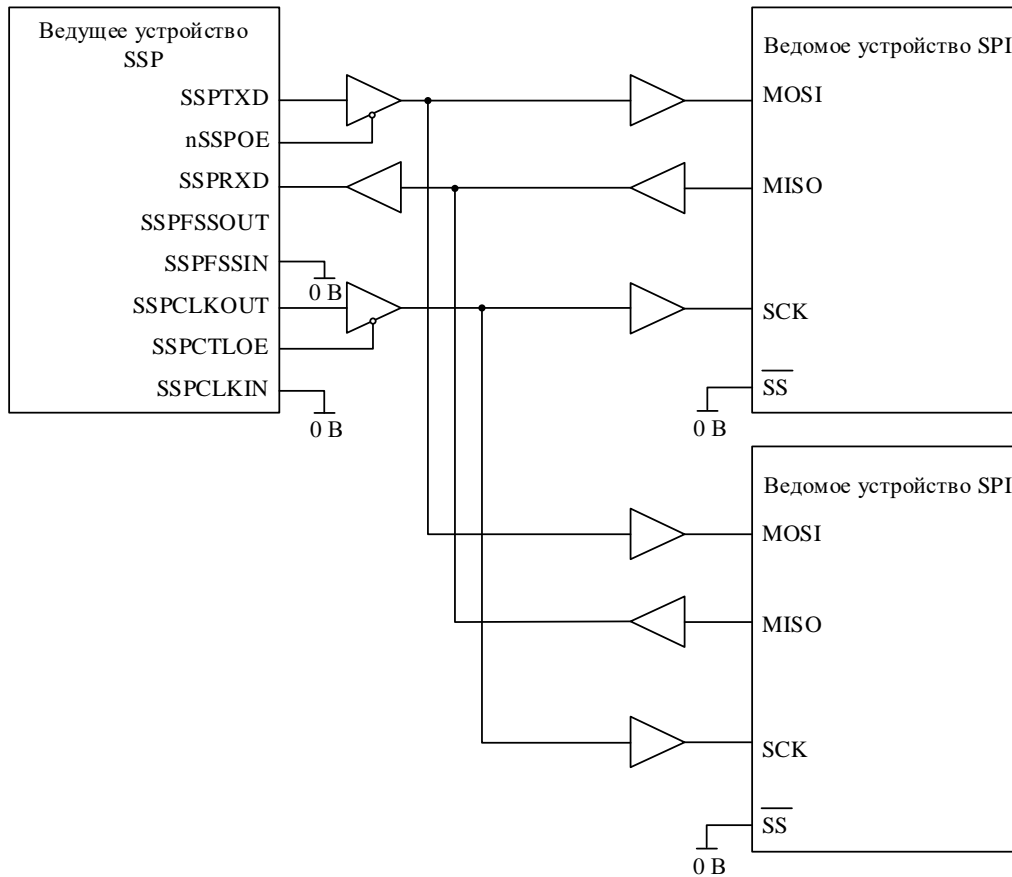


Рисунок 75 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим SPI

Рисунок 75 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 SSPTXD. Ответная передача данных на входную линию SSPRXD ведущего устройства одновременно осуществляется только одним из ведомых по соответствующей линии MISO.

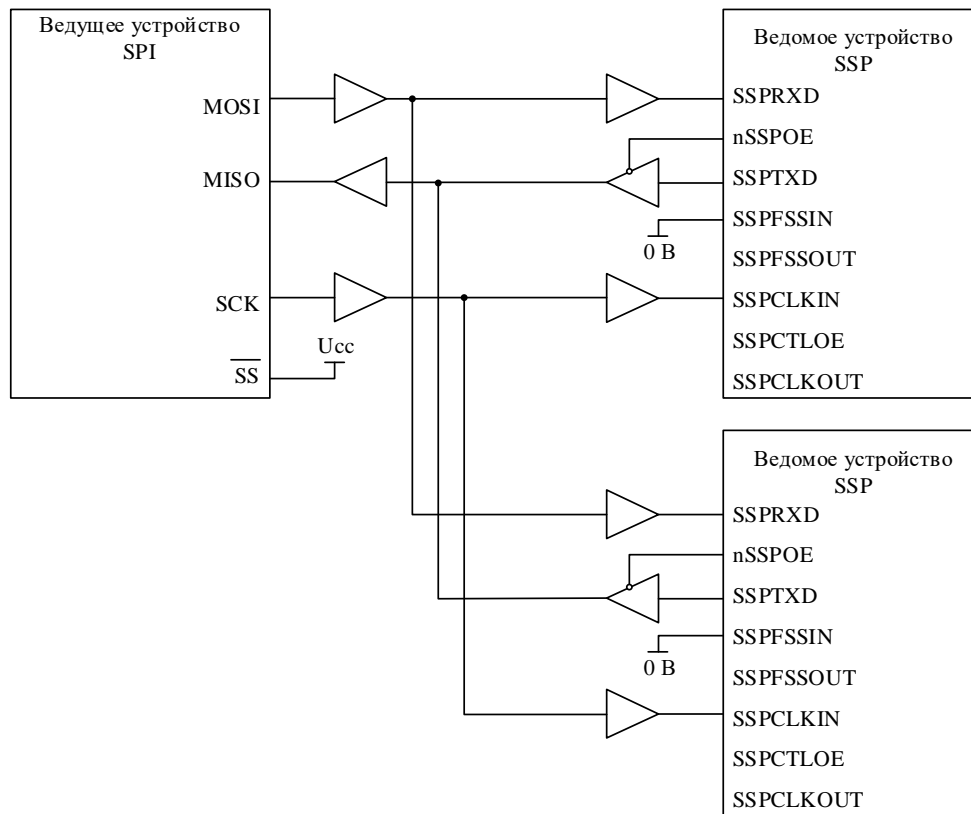


Рисунок 76 – Ведущее устройство, протокол SPI, подключено к двум ведомым модулям SSP

Рисунок 76 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 SSPTXD в активное состояние, разрешая прохождение сигнала от своей линии SSPTXD на вход SSPRXD ведущего.

### 23.12 Интерфейс прямого доступа к памяти

Модуль SSP предоставляет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA DMACR.

Интерфейс DMA включает в себя следующие сигналы:

– для приема:

– SSPRXDMASREQ – запрос передачи отдельного слова, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит, по меньшей мере, одно слово;

– SSPRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если буфер FIFO приемника содержит четыре или более слов;

– 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 слова данных в блоке.

– для передачи:

– SSPTXDMASREQ – запрос передачи отдельного слова, 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 слов;

– 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 слова 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.

Таблица 379 показывает значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 379 – 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

Рисунок 77 показывает временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

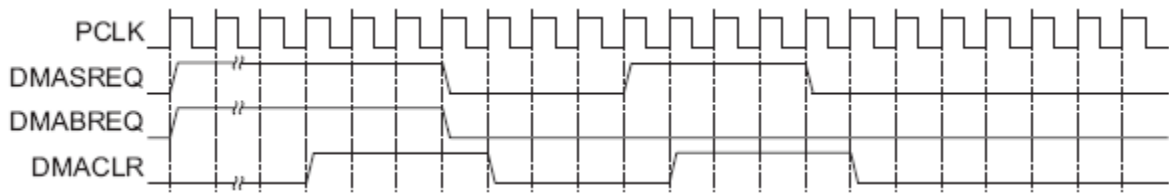


Рисунок 77 – Временные диаграммы обмена в режиме DMA

### 23.13 Прерывания

В модуле предусмотрено семь маскируемых линий запроса на прерывание с выводом на 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;
- SSPRNEINTR – наличие данных в FIFO приемника (доступно с ревизии 2);
- SSPTFEINTR – отсутствие данных в FIFO передатчика (доступно с ревизии 2);
- SSPTNBSYINTR – отсутствие данных в сдвиговом регистре передатчика (доступно с ревизии 2);
- SSPINTR – логическое ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR, SSPRORINTR, SSPRNEINTR, SSPTFEINTR и SSPTNBSYINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSPIMSC. Установка бита в «1» разрешает соответствующее прерывание, а в «0» – запрещает.

Доступность индивидуальных линий, 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 согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний SSPRIS, либо из маскированного регистра прерываний SSPMIS.



### 23.13.1 SSPRXINTR

Прерывание по заполнению буфера FIFO приемника. Формируется в случае, если буфер приемника содержит четыре или более несчитанных слов данных.

### 23.13.2 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.

### 23.13.3 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.

### 23.13.4 SSPRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение времени таймаута, равного 32 тактам частоты SSPCLKOUT (для ведущего и ведомого режимов работы). 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 SSP\_RXD. Кроме того, оно может быть снято путем записи 1 в бит RTIC регистра сброса прерывания SSPTICR.

### 23.13.5 SSPRNEINTR (доступно с ревизии 2)

Прерывание по наличию данных в FIFO приемника. Сигнал формируется при записи хотя бы одного слова в FIFO приемника и сохраняется до тех пор, пока все данные не будут считаны.

### 23.13.6 SSPTFEINTR (доступно с ревизии 2)

Прерывание по отсутствию данных в FIFO передатчика. Находится в активном уровне до тех пор, пока в FIFO не будет записано хотя бы одно слово. После передачи всех данных возвращается в активный уровень.

### 23.13.7 SSPTNBSYINTR (доступно с ревизии 2)

Прерывание по отсутствию данных в сдвиговом регистре передатчика.

### 23.13.8 SSPINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRNEINTR, SSPTNBSYINTR, SSPTBSYINTR, SSPRTINTR и SSPRORINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерываний, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

## 23.14 Программное управление модулем

### 23.14.1 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.

### 23.14.2 Описание регистров контроллера SSP

Данные о регистрах модуля SSP приведены в таблице 380.

Таблица 380 – Обобщенные данные о регистрах модуля SSP

Базовый Адрес	Наименование				Описание
0x4000_0000	SSP1				Регистры контроллера интерфейса SSP1
0x400C_8000	SSP2				Регистры контроллера интерфейса SSP2
0x400D_0000	SSP3				Регистры контроллера интерфейса SSP3
Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x000	CR0	RW	0x0000	16 (18*)	Регистр управления 0
0x004	CR1	RW	0x0	4 (5*)	Регистр управления 1
0x008	DR	RW	0x----	16 (32*)	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись)
0x00C	SR	RO	0x03	5	Регистр состояния
0x010	CPSR	RW	0x00	8	Регистр делителя тактовой частоты
0x014	IMSC	RW	0x0	4	Регистр маски прерывания
0x018	RIS	RO	0x8	4 (7*)	Регистр состояния прерываний без учета маскирования

Базовый Адрес	Наименование				Описание
0x01C	MIS	RO	0x0	4 (7*)	Регистр состояния прерываний с учетом маскирования
0x020	ICR	WO	0x0	2 (7*)	Регистр сброса прерывания
0x024	DMACR	RW	0x0	2	Регистр управления прямым доступом к памяти
<p>* С ревизии 2.                      Примечание – В поле «тип» указан вид доступа к регистру:                      RW – чтение и запись;                      RO – только чтение;                      WO – только запись</p>					

### 23.14.2.1 Регистр CR0

Регистр управления 0.

Регистр CR0 содержит пять (семь с ревизии 2) битовых полей, предназначенных для управления блоками модуля SSP.

Назначение бит регистра CR0 представлено в таблице 381.

Таблица 381 – Формат регистра CR0

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...18	-	Зарезервировано
17	SSPFRX	Для ведомого модуля включает быстрый режим протокола SPI и активирует синхронизацию сигнала RXD (доступно с ревизии 2). 1 – активация быстрого режима работы ведомого 0 – нормальный режим работы
16	DSS[4]	Старший разряд размера слова данных (доступно с ревизии 2): 0 – DSS[3:0]; 1 – DSS[3:0] + 16 бит
15...8	SCR	Скорость последовательного обмена. Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению $\frac{F_{SSPCLK}}{(CPSDVR \cdot (1 + SCR))}$ где CPSDVR – четное число в диапазоне от 2 до 254 (см. подпункт 23.14.2.5 «Регистр CPSR»), SCR – число от 0 до 255
7	SPH	Фаза сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. пункт 23.10.2 «Формат синхронного обмена SPI фирмы Motorola»
6	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. пункт 23.10.2 «Формат синхронного обмена SPI фирмы Motorola»

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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[3:0]	Младшие четыре разряда размера слова данных. При DSS[4] = 0: 0000 – резерв 0001 – резерв 0010 – резерв 0011 – 4 бита 0100 – 5 бит 0101 – 6 бит ... 1110 – 15 бит 1111 – 16 бит  При DSS[4] = 1: 0000 – 17 бит 0001 – 18 бит 0010 – 19 бит 0011 – 20 бит 0100 – 21 бит 0101 – 22 бита ... 1110 – 31 бит 1111 – 32 бита

### 23.14.2.2 Регистр CR1

#### Регистр управления 1.

Регистр CR1 содержит четыре (пять с ревизии 2) битовых поля, предназначенных для управления блоками модуля SSP.

Назначение бит регистра CR1 представлено в таблице 382.

Таблица 382 – Регистр CR1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...5		Резерв, при чтении результат не определен. При записи следует устанавливать в 0
4	RESTxFIFO	Программный сброс FIFO передатчика (доступно с ревизии 2): 0 – нормальный режим работы FIFO передатчика; 1 – активация сброса FIFO передатчика. Сбрасывается в ноль автоматически по завершении сброса

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 данными ведомых устройств можно соединить параллельно. 0 – управление линией SSP_TXD в ведомом режиме разрешено; 1 – управление линией SSP_TXD в ведомом режиме запрещено
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 со входом регистра сдвига приемника

### 23.14.2.3 Регистр DR

*Регистр данных.*

Регистр SSPDR имеет разрядность 16 бит (32 бита с ревизии 2) 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 (32 бит с ревизии 2), перед записью в регистр SSPDR 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 бит регистра DR представлено в таблице 386.

Таблица 383 – Формат регистра DR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	DATA	Принимаемые данные 16 бит (в ревизии 1) или 32 бита (доступно с ревизии 2) (чтение). Передаваемые данные 16 бит (в ревизии 1) или 32 бита (доступно с ревизии 2) (запись). В случае, если выбран размер информационного слова менее 16 бит (32 бит с ревизии 2), перед записью в регистр SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника

#### 23.14.2.4 Регистр SR

*Регистр состояния.*

Регистр SR доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика, и занятости модуля SSP.

Назначение бит регистра SR представлено в таблице 384.

Таблица 384 – Регистр SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...5	-	Зарезервировано. При чтении результат не определен. При записи следует заполнить нулями
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

### 23.14.2.5 Регистр CPSR

*Регистр делителя таковой частоты.*

Регистр 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 CPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Назначение бит регистра CPSR представлено в таблице 385.

Таблица 385 – Регистр CPSR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...8	-	Зарезервировано. 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

### 23.14.2.6 Регистр 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 бит регистра IMSC представлено в таблице 386.

Таблица 386 – Регистр IMSC

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...7		Резерв. Не модифицируйте. При чтении выдаются нули
6	TNBSYIM	Маска прерывания по пустоте сдвигового регистра передатчика (доступно с ревизии 2). 1 – не маскирована; 0 – маскирована
5	TFEIM	Маска прерывания по пустоте FIFO передатчика (доступно с ревизии 2). 1 – не маскирована; 0 – маскирована
4	RNEIM	Маска прерывания по наличию данных в FIFO приемника (доступно с ревизии 2). 1 – не маскирована; 0 – маскирована

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	TXIM	Маска прерывания по заполнению на 50% и менее буфера FIFO передатчика. 1 – не маскирована; 0 – маскирована
2	RXIM	Маска прерывания по заполнению на 50% 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 попуток его чтения в течение времени таймаута). 1 – не маскирована; 0 – маскирована
0	RORIM	Маска прерывания по переполнению буфера приемника. 1 – не маскирован; 0 – маскирована

### 23.14.2.7 Регистр 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 регистра RIS представлено в таблице 387.

Таблица 387 – Регистр RIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...7		Резерв. Не модифицируйте. При чтении выдаются нули.
6	TNBSYRIS	Состояние до маскирования прерывания SSPTNBSYINTR (доступно с ревизии 2)
5	TFERIS	Состояние до маскирования прерывания SSPTFEINTR (доступно с ревизии 2)
4	RNERIS	Состояние до маскирования прерывания SSPRNEINTR (доступно с ревизии 2)
3	TXRIS	Состояние до маскирования прерывания SSPTXINTR
2	RXRIS	Состояние до маскирования прерывания SSPRXINTR
1	RTRIS	Состояние до маскирования прерывания SSPRTINTR
0	RORRIS	Состояние до маскирования прерывания SSPRORINTR

### 23.14.2.8 Регистр MIS

*Регистр маскированного состояния прерываний.*

Регистр MIS доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит регистра MIS представлено в таблице 388.



Таблица 388 – Регистр MIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...7		Резерв. Не модифицируйте. При чтении выдаются нули
6	TNBSYMIS	Состояние маскированного прерывания SSPTNBSYINTR (доступно с ревизии 2)
5	TFEMIS	Состояние маскированного прерывания SSPTFEINTR (доступно с ревизии 2)
4	RNEMIS	Состояние маскированного прерывания SSPRNEINTR (доступно с ревизии 2)
3	TXMIS	Состояние маскированного прерывания SSPTXINTR
2	RXMIS	Состояние маскированного прерывания SSPRXINTR
1	RTMIS	Состояние маскированного прерывания SSPRTINTR
0	RORMIS	Состояние маскированного прерывания SSPRORINTR

### 23.14.2.9 Регистр ICR

*Регистр сброса прерываний.*

Регистр ICR 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 регистра ICR представлено в таблице 389.

Таблица 389 – Регистр ICR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...2	-	Зарезервировано. При записи следует заполнить нулями
1	RTIC	Сброс прерывания SSPRTINTR
0	RORIC	Сброс прерывания SSPRORINTR

### 23.14.2.10 Регистр DMACR

*Регистр управления прямым доступом.*

Регистр DMACR доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра DMACR представлено в таблице 390.

Таблица 390 – Регистр DMACR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...2	-	Зарезервировано. При чтении выдаются нули. При записи следует заполнить нулями

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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

## 24 Контроллер UART

Модуль универсального асинхронного приемопередатчика (UART – Universal Asynchronous Receiver-Transmitter) представляет собой периферийное устройство микросхемы.

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).

### 24.1 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 (16 слов по 13 бит) и передачи (16 слов по 9 бит) 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 программно настраивать скорость обмена данными, путем деления тактовой частоты опорного генератора в диапазоне ( $1 \times 16 - 65535 \times 16$ ). 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 микросхемы.

Также модуль обеспечивает обнаружение ложных стартовых бит.

Формирование и обнаружения сигнала разрыва линии.

Возможность организации аппаратного управления потоком данных.

Полностью программируемый UART имеет следующие характеристики:

- данные длиной 5, 6, 7, 8 или 9 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 – до  $UART\_CLK/16$  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 UART\_CLK для получения заданной длительности бит в режиме пониженного энергопотребления.

## 24.2 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 от 1,42 до 2,12 МГц для обеспечения возможности формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления (для ИК-обмена);
- режим аппаратного управления потоком данных.

## 24.3 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.

## 24.4 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 слов по 13 бит для режима приема и до 16 слов по 9 бит для режима передачи.

Модуль приемопередатчика:

- содержит программируемый генератор, формирующий тактовый сигнал одновременно для передачи и для приема данных на основе внутреннего тактового сигнала UART\_CLK;
- обеспечивает возможности, сходные с возможностями промышленного стандарта – контроллера UART 16C650;
- позволяет осуществлять обмен информацией с максимальной скоростью:
  - 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 LCR\_H регистрами делителя скорости передачи данных – целой части (IBRD) и дробной части (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 UARTTXD и UARTRXD, а посредством сигналов nSIROUT и SIRIN.

В этом случае устройство переводит линию UARTTXD в пассивное состояние (высокий уровень), и перестает реагировать на изменение состояния модема, а также сигнала на линии UARTRXD. 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.

### 24.5 Описание функционирования блока UART

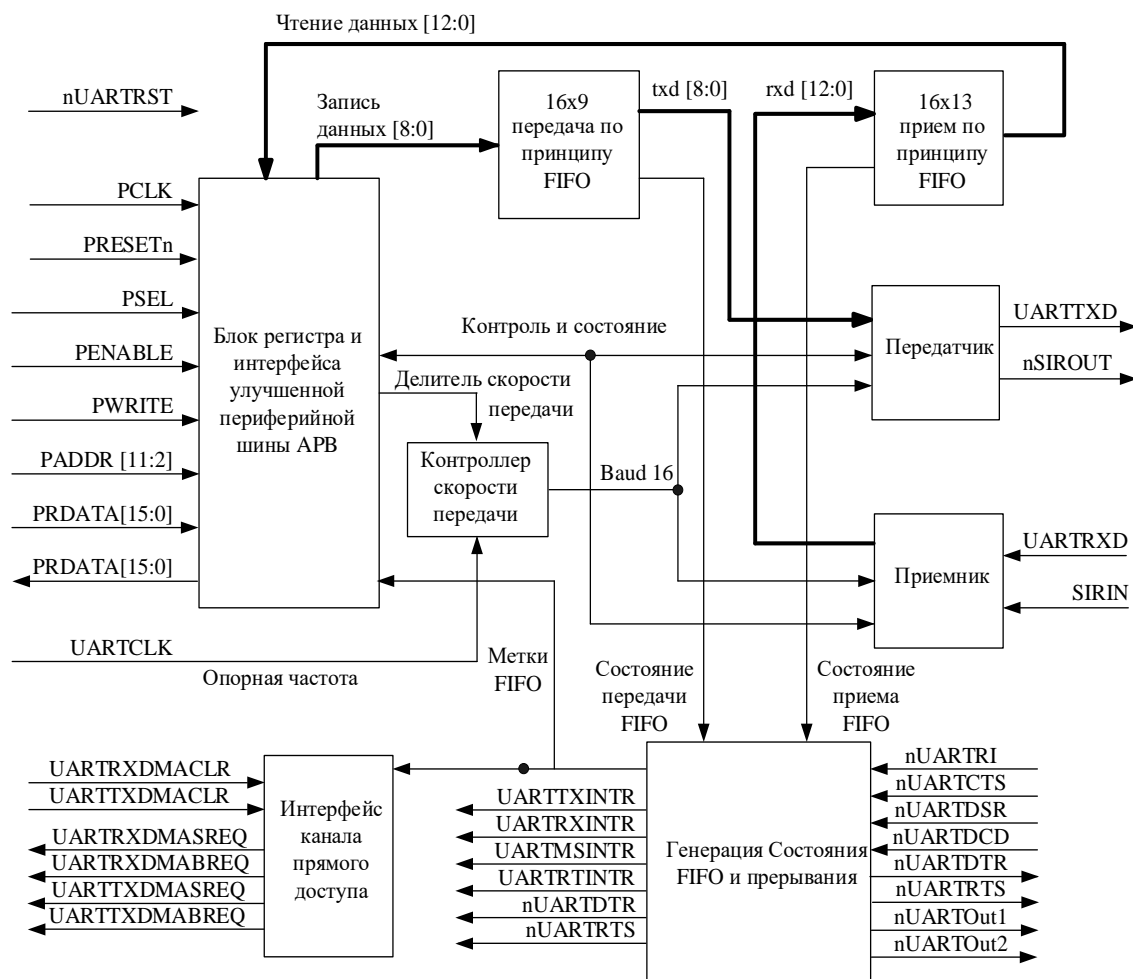


Рисунок 78 – Блок-схема универсального асинхронного приемопередатчика (UART)

### 24.5.1 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\_CLK 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.

### 24.5.2 Буфер FIFO передатчика

Буфер передатчика имеет ширину 9 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.

### 24.5.3 Буфер FIFO приемника

Буфер приемника имеет ширину 13 бит, глубину 16 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Принятые от периферийного устройства данные и соответствующие коды ошибок 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.

### 24.5.4 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.

### 24.5.5 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 на: переполнение буфера, ошибки контроля четности, ошибки в структуре сигнала и на разрыв линии. Признаки обнаружения этих ошибок также сохраняются в выходном буфере.

### 24.5.6 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 сигналов запроса на прерывание от приемопередатчика. В этом случае процедура обработки сможет одновременно считать информацию обо всех источниках прерываний. 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. подраздел 24.10 «Прерывания».

### 24.5.7 Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

### 24.5.8 Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов CPU\_CLK и UART\_CLK. 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\_CLK, так и наоборот.

## 24.6 Описание функционирования ИК-кодека IrDA SIR

Структурная схема кодека представлена на рисунке 79.

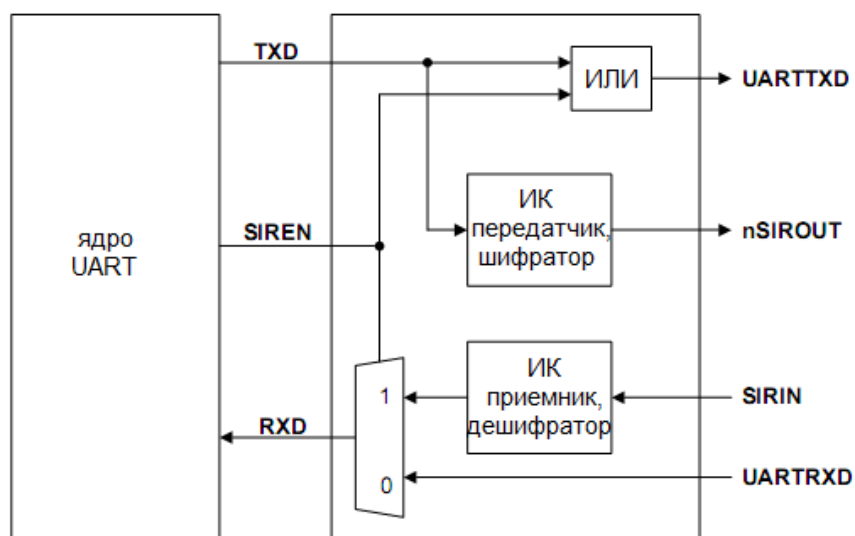


Рисунок 79 – Структурная схема кодека IrDA



### 24.6.1 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\_CLK. Значение частоты IrLPBaud16 задается путем записи соответствующего коэффициента деления частоты в регистр 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\_CLK. Можно показать, что в наихудшем случае величина джиттера в потоке ИК-импульсов может достигать трех периодов UART\_CLK. В соответствии со спецификацией стандарта IrDA SIR, джиттер не должен превышать величины 13 %. В случае, если частота сигнала UART\_CLK 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.

### 24.6.2 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 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 SIRIN игнорируются импульсы с длительностью менее, чем:

- 3/16 длительности Baud16 в режиме IrDA;
- 3/16 длительности IrLPBaud16 в режиме IrDA с пониженным энергопотреблением.

## 24.7 Описание работы UART

### 24.7.1 Сброс модуля

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса процессора. Значения регистров после сброса описаны в подразделе 24.11 «Программное управление модулем».

### 24.7.2 Тактовые сигналы

Частота тактового сигнала  $F_{UART\_CLK}$  должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

$$F_{UART\_CLK}(\min) \geq 16 \cdot \text{baud\_rate\_max};$$

$$F_{UART\_CLK}(\max) \leq 16 \cdot 65535 \cdot \text{baud\_rate\_min}.$$

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота  $F_{UART\_CLK}$  должна находиться в интервале от 7,3728 до 115,34 МГц.

Частота  $F_{UART\_CLK}$ , кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами  $F_{CPU\_CLK}$  и  $F_{UART\_CLK}$ . Частота  $F_{UART\_CLK}$  должна быть не более чем в 5/3 раз выше частоты  $F_{CPU\_CLK}$ .

$$F_{UART\_CLK} \leq \frac{5}{3} \cdot F_{CPU\_CLK}.$$

Например, при работе в режиме UART с максимальной скоростью передачи данных 921600 бод, при частоте  $F_{UART\_CLK} = 14,7456$  МГц, частота  $F_{CPU\_CLK}$  должна быть не менее 8,85276 МГц. Это гарантирует, что контроллер UART будет иметь достаточно времени для записи принятых данных в буфер FIFO.

### 24.7.3 Работа универсального асинхронного приемопередатчика

Управляющая информация хранится в регистре управления линией LCR. Этот регистр имеет внутреннюю ширину 30 бит, однако внешний доступ по шине APB к нему осуществляется через следующие регистры:

- LCR\_H – определяет:
  - параметры передачи данных;
  - длину слова;
  - режим буферизации;
  - количество передаваемых стоповых бит;
  - режим контроля четности;
  - формирование сигнала разрыва линии;

- IBRD – определяет целую часть коэффициента деления для скорости передачи данных;
- FBRD – определяет дробную часть коэффициента деления для скорости передачи данных.

Примечание – Изменение значений трех регистров можно осуществить корректно двумя способами:

- запись IBRD, запись FBRD, запись LCR\_H;
- запись FBRD, запись IBRD, запись LCR\_H.

Чтобы изменить значение лишь одного из регистров (IBRD или FBRD), необходимо выполнить следующие шаги:

- запись IBRD (или FBRD), запись LCR\_H.

#### 24.7.4 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, не превышающими 230400, при этом используя в качестве UART\_CLK тактовый сигнал с произвольной частотой более 3,6864 МГц.

Целая часть коэффициента деления записывается в 16-битный регистр IBRD. Шестиразрядная дробная часть записывается в регистр FBRD. Значение коэффициента деления связано с содержимым указанных регистров следующим образом

$$\text{Коэффициент деления} = \frac{F_{UART\_CLK}}{16 \cdot \text{скорость передачи данных}} = IBRD + FBRD, \quad (21)$$

где *IBRD* – целая часть коэффициента деления;  
*FBRD* – дробная часть коэффициента деления.

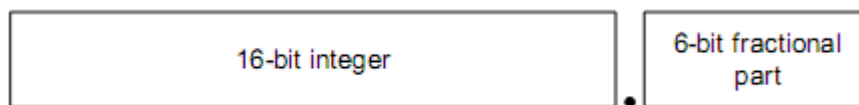


Рисунок 80 – Коэффициент деления

Шестибитное значение, записываемое в регистр FBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на  $2^n$ , где  $n$  – ширина регистра FBRD) и округления до ближайшего целого числа

$$m = \text{integer}(FBRD \cdot 2^n + 0,5), \quad (22)$$

где *integer* – операция отсечения дробной части числа,  $n = 6$ .

В модуле формируется внутренний сигнал Vaud16, представляющий собой последовательность импульсов с длительностью, равной периоду сигнала UART\_CLK и средней частотой, в 16 раз большей требуемой скорости обмена данными.

### 24.7.5 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 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 UARTRXD постоянно присутствует 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 UARTRXD 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 бита 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 UARTRXD). 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.

### 24.7.6 Биты ошибки

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся на позиции [11:9] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO, расположенный на позиции 12-го слова данных.

Таблица 391 показывает назначение всех битов слова данных в FIFO буфере приемника.

Таблица 391 – Назначение бит слова данных в FIFO-буфере приемника

Бит буфера FIFO	Назначение
12	Признак переполнения буфера
11	Ошибка – разрыв линии
10	Ошибка проверки на четность
09	Ошибка формирования кадра
08...00	Принятые данные

### 24.7.7 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.

### 24.7.8 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.

#### 24.7.8.1 Проверка по шлейфу

Проверка по шлейфу (замыкание выхода передатчика на вход приемника) выполняется путем установки в 1 бита LBE в регистре управления контроллером CR.

### 24.7.9 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:

– Режим IrDA – уровень логического нуля передается на линию nSIROUT 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 SIRIN.

– Режим IrDA с пониженным энергопотреблением – длительность передаваемых импульсов ИК излучения устанавливается в три раза больше 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 CR.

Как в режиме IrDA, так и в режиме IrDA с пониженным энергопотреблением:

– 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\_CLK в соответствии с коэффициентом деления, записанным в регистре ILPR.

Коэффициент деления вычисляется по формуле

$$\text{Коэффициент деления} = \frac{F_{UART\_CLK}}{IrLPBaud16}, \quad (23)$$

где номинальное значение *IrLPBaud16* составляет 1,8432 МГц. Коэффициент деления должен быть выбран так, чтобы выполнялось соотношение

$$1,42 \text{ МГц} < IrLPBaud16 < 2,12 \text{ МГц}. \quad (24)$$

### 24.7.9.1 Проверка по шлейфу

Проверка по шлейфу выполняется после установки в 1 бита LBE регистра управления контроллером CR с одновременной установкой в 1 бита SIRTEST регистра управления тестированием TCR.

В этом режиме данные, передаваемые на выход nSIROUT, должны подаваться на вход SIRIN.

Примечание – Это единственный случай использования тестового регистра в нормальном режиме функционирования модуля.

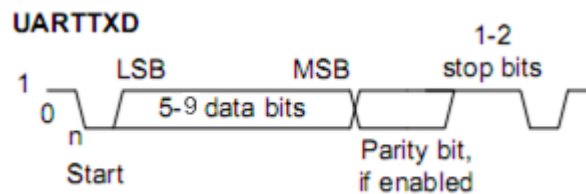


Рисунок 81 – Кадр передачи данных

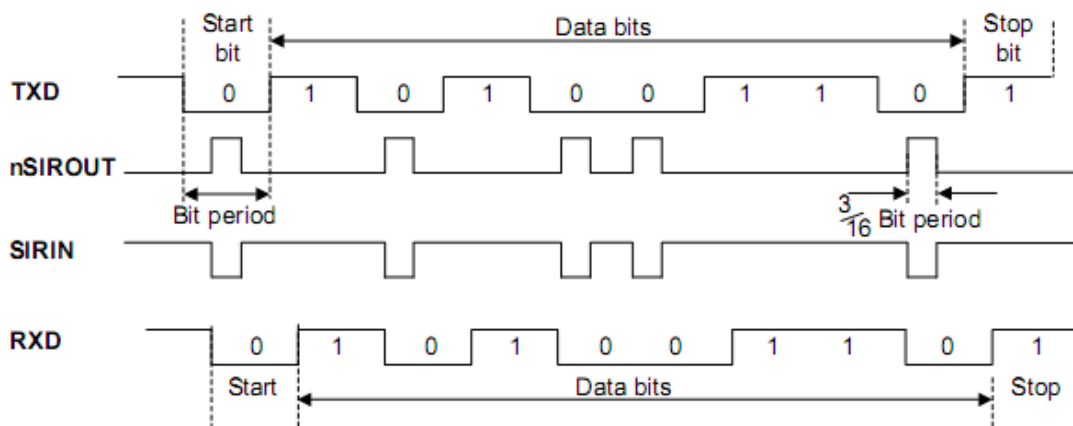


Рисунок 82 – Модуляция данных IrDA

## 24.8 Линии управления модемом

Модуль универсального асинхронного приемопередатчика может использоваться как в режиме оконечного оборудования (DTE), так и в режиме оборудования передачи данных (DCE).

Таблица 392 – Назначение управления модемом в режимах DTE и DCE

Сигнал	Назначение	
	Режим оконечного оборудования	Режим оборудования передачи данных
nUARTCTS	Готов к передаче данных	Запрос передачи данных
nUARTDSR	Источник данных готов	Приемник данных готов
nUARTDCD	Обнаружен информационный сигнал	-
nUARTRI	Индикатор вызова	-
nUARTRTS	Запрос передачи данных	Готов к передаче данных
nUARTDTR	Приемник данных готов	Источник данных готов
nUARTOut1	-	Обнаружен информационный сигнал
nUARTOut2	-	Индикатор вызова

### 24.8.1 Аппаратное управление потоком данных

Программно активизируемый режим аппаратного управления потоком данных позволяет контролировать (приостанавливать и возобновлять) информационный обмен с помощью сигналов nUARTRTS и nUARTCTS. Иллюстрация взаимодействия двух устройств последовательной связи с аппаратным управлением потоком данных представлена на рисунке 83.

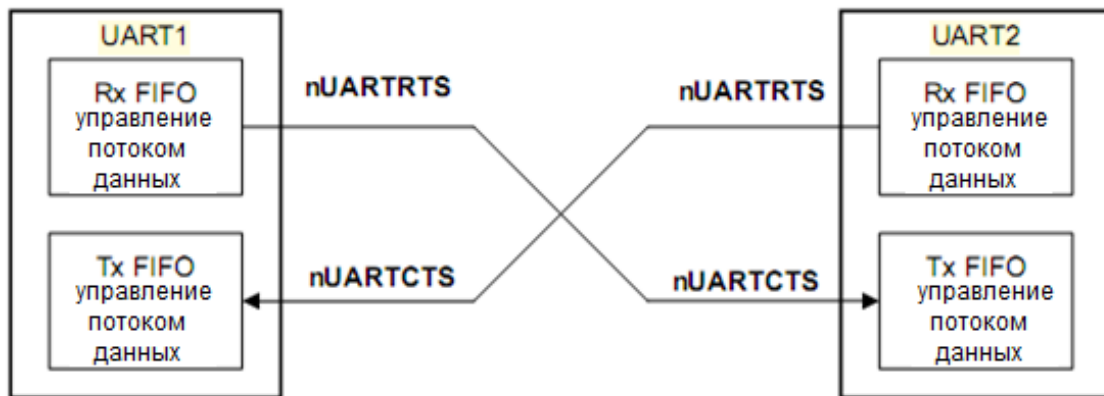


Рисунок 83 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

Если разрешено управление потоком данных по сигналу RTS, линия nUARTRTS переводится в активное состояние 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 nUARTCTS в активное состояние.

Режим аппаратного управления потоком данных задается путем установки значений бит RTSEn и CTSEn в регистре управления CR. В таблице 393 показаны необходимые установки для различных режимов управления потоком данных.

Таблица 393 – 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 RTS регистра CR для проверки состояния линии RTS

### 24.8.2 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 nUARTRTS 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



nUARTRTS 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 nUARTRTS 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.

### 24.8.3 Управление потоком данных по линии CTS

В случае выбора одного из режимов с управлением потоком данных по CTS передатчик осуществляет проверку состояния линии nUARTCTS 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 nUARTCTS 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.

## 24.9 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 DMACR.

Интерфейс DMA включает в себя следующие сигналы:

#### Для приема:

– UARTRXDMASREQ – запрос передачи отдельного символа, инициируется контроллером UART. Размер символа в режиме приема данных – до 13 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ.

– UARTRXDMABREQ – 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 IFLS.

– UARTRXDMACLR – 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:

– UARTTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Размер символа в режиме передачи данных – до 9 бит.

Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку.

– UARTTXDMABREQ – 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 IFLS.

– UARTTXDMACLR – 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 исключаящими, 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 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 установки в ноль бита управления DMA TXDMAE или RXDMAE в регистре управления DMA DMACR.

В случае запрета буферов FIFO устройство способно передавать и принимать только одиночные символы, как следствие, контроллер может инициировать DMA только в одноэлементном режиме. При этом модуль в состоянии формировать только сигналы управления DMA UARTRXDMASREQ и UARTTXDMASREQ. Для информации о запрете буферов FIFO см. описание регистра управления линией 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 394 показывает значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

Таблица 394 – 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 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: UARTRXDMASREQ и UARTRXDMABREQ переводятся в неактивное состояние (маскируются) до сброса UARTEINTR. На линии запроса DMA, обслуживающие передатчик, состояние UARTEINTR не влияет.

Рисунок 84 показывает временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с CPU\_CLK. В интересах ясности изложения предполагается, что синхронизация сигналов запроса DMA в контроллере DMA не производится.

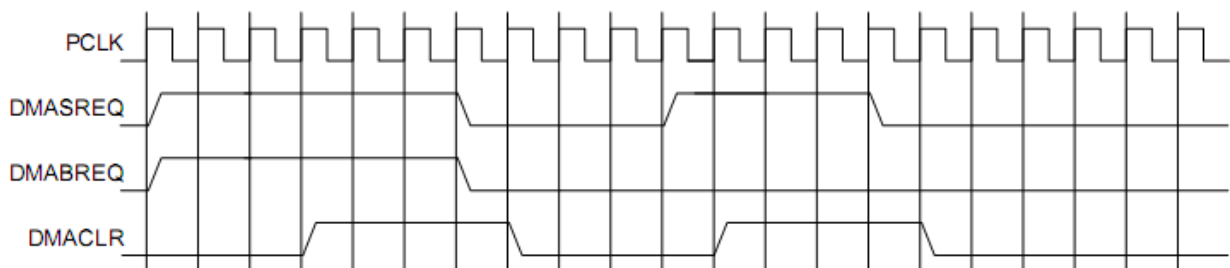


Рисунок 84 – Временные диаграммы одноэлементного и блочного запросов DMA

### 24.10 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;
- UARTRXINTR – прерывание от передатчика;
- UARTRNEINTR – FIFO приемника не пусто;
- UARTRTFEINTR – FIFO передатчика пусто;
- UARTRTNBSYINTR – сдвиговый регистр передатчика пуст;
- UARTRRTINTR – прерывание по таймауту приемника;
- UARTRMSINTR – прерывание по состоянию модема:
  - UARTRRIINTR, изменение состояния линии nUARTRI;
  - UARTRCTSINTR, 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, UARTEINTR, UARTRNEINTR, UARTTFEINTR и UARTTNBSYINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски UARTIMSC. 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 UARTRXINTR и UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Прерывание по обнаружению ошибке UARTEINTR 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.

Прерывание UARTRNEINTR возникает в случае, если FIFO приемника получает хотя бы одно слово данных, то есть становится не пусто.

Прерывание UARTTFEINTR выставляется, если FIFO передатчика не имеет никаких данных, то есть пусто в данный момент.

UARTTNBSYINTR выставляется в том случае, если сдвиговый регистр передатчика пуст.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний UARTRIS, либо из маскированного регистра прерываний UARTRMIS.

#### 24.10.1 UARTMSINTR

Прерывание по состоянию модема возникает в случае изменения любой из линий состояний модема (nUARTCTS, nUARTDCD, nUARTDSR, nUARTRI). Сброс прерывания осуществляется путем записи 1 в соответствующий (в зависимости от линии состояния модема, вызвавшей прерывание) разряд регистра сброса прерывания ICR.

### 24.10.2 UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен 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.

### 24.10.3 UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен 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 основано на переходе через пороговое значение, а не на состоянии заполненности буфера FIFO передатчика относительно порогового значения. 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.

### 24.10.4 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 ICR.

### 24.10.5 UARTEINTR

Прерывание по обнаружению ошибки происходит в случае возникновения 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 RIS, либо содержимое маскированного регистра прерываний MIS.

Сброс прерывания осуществляется путем записи соответствующих бит в регистр сброса прерывания ICR. За прерываниями по обнаружению ошибки закреплены биты с 7 по 10.

### 24.10.6 UARTINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR, UARTEINTR, UARTRNEINTR, UARTTFEINTR и UARTTNBSYINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

## 24.11 Программное управление модулем

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» таблицы 395 определяет режим доступа к регистру в соответствии с обозначениями:
  - RW – чтение и запись;
  - RO – только чтение;
  - WO – только запись.

## 24.12 Обобщенные данные о регистрах устройства

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в таблице 395.

Таблица 395 – Обобщенные данные о регистрах устройства

Базовый Адрес	Название				Описание
0x40008000	UART1				Регистры контроллера интерфейса UART1
0x40010000	UART2				Регистры контроллера интерфейса UART2
0x40038000	UART3				Регистры контроллера интерфейса UART3
0x400E8000	UART4				Регистры контроллера интерфейса UART4
Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x000	DR	RW	0x---	13/9	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	9	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	14	Регистр маски прерывания
0x03C	RIS	RO	0x300-	14	Регистр состояния прерываний
0x040	MIS	RO	0x00-	14	Регистр состояния прерываний с маскированием
0x044	ICR	WO	-	11	Регистр сброса прерывания
0x048	DMACR	RW	0x00	3	Регистр управления DMA
0x080	TCR	RW	0x00	3	Регистр управления тестированием

### 24.12.1 Регистр данных DR

#### 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 396).

Таблица 396 – Формат регистра UARTDR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...13	-	Зарезервировано
12	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
11	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
10	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 LCR_H. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер
9	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
8...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.

### 24.12.2 Регистр состояния приемника / сброса ошибки RSR\_ECR

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.

Таблица 397 показывает назначение бит регистра RSR\_ECR.

Таблица 397 – Регистр RSR\_ECR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7...4	-	Зарезервировано. 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 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 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 LCR_H. Бит сбрасывается в 0 после записи в регистр ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера
<p>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</p>		

### 24.12.3 Регистр флагов FR

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. Таблица 398 содержит информацию о назначении бит регистра.

Таблица 398 – Регистр FR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 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 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 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 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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

#### 24.12.4 Регистр управления ИК-обменом в режиме пониженного энергопотребления ILPR

Этот восьмиразрядный регистр, доступный для чтения и записи, содержит значение коэффициента деления частоты UART\_CLK, для формирования тактового сигнала IrLPBaud16. Назначение разрядов регистра показано в таблице 399.

Требуемое значение коэффициента деления для формирования сигнала IrLPBaud16 вычисляется по формуле

$$ILPDVSR = \frac{F_{UART\_CLK}}{F_{IrLPBaud16}}, \quad (25)$$

где номинальное значение частоты  $F_{IrLPBaud16}$  составляет 1,8432 МГц.

Коэффициент деления должен быть установлен таким образом, чтобы выполнялось соотношение

$$1,42 \text{ МГц} < F_{IrLPBaud16} < 2,12 \text{ МГц}, \quad (26)$$

что, в свою очередь, гарантирует формирование кодеком импульсов данных с длительностью (1,41 – 2,11) мкс (в три раза длиннее периода сигнала IrLPBaud16).

Таблица 399 – Регистр ILPR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7...0	ILPDVSR	Коэффициент деления частоты UART_CLK, 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 SIRIN импульсы с длительностью, меньшей трех периодов сигнала IrLPBaud16.

### 24.12.5 Регистр целой части делителя скорости передачи данных IBRD

Назначение бит регистра представлено в таблице 400.

Таблица 400 – Регистр IBRD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в «0»

### 24.12.6 Регистр дробной части делителя скорости передачи данных FBRD

Назначение бит регистра представлено в таблице ниже.

Таблица 401 – Регистр BFRD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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

$$BAUDDIV = \frac{F_{UART\_CLK}}{16 \cdot Baud\_rate}, \quad (27)$$

где  $F_{UART\_CLK}$  – тактовая частота контроллера UART;

$Baud\_rate$  – требуемая скорость передачи данных (в бит/с).

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUDDIV\_INT и BAUDDIV\_FRAC, соответственно.

Примечание – Изменение содержимого регистров IBRD и 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 IBRD, равное 0 является недопустимым, при этом значение регистра FBRD игнорируется.

Аналогично, при IBRD равном 65535 (0xFFFF) значение FBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

*Пример. Вычисление коэффициента деления*

Пусть требуемая скорость передачи данных составляет 230400 бит/с, частота тактового сигнала  $F_{UART\_CLK} = 4$  МГц. Тогда

$$\text{Коэффициент деления} = \frac{4 \cdot 10^6}{16 \cdot 230400} = 1,085.$$

Таким образом, BRDI = 1, BRDF = 0,085.

Следовательно, значение, записываемое в регистр BFRD, равно

$$m = \text{integer}((0,085 \cdot 64) + 0,5) = 5 .$$

Реальное значение коэффициента деления =  $1 + 5/64 = 1,078$ .

Реальная скорость передачи данных =  $(4 \cdot 10^6)/(16 \cdot 1,078) = 231911$  бит/с.

Ошибка установки скорости =  $(231911 - 230400)/230400 \cdot 100\% = 0,656\%$ .

Максимальная ошибка установки скорости передачи данных с использованием шестизрядного регистра BFRD =  $1/64 \cdot 100\% = 1,56\%$ . Такая ошибка возникает в случае  $m = 1$ , при этом разница накапливается в течение 64 тактовых интервалов.

Таблица 402 содержит значения коэффициента деления для типичных скоростей передачи данных при частоте  $F_{\text{UART\_CLK}} = 7,3728$  МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр FBRD должен быть записан ноль.

Таблица 402 – Коэффициенты деления при частоте  $F_{\text{UART\_CLK}} = 7,3728$  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

Таблица 403 содержит значения коэффициента деления для типичных скоростей передачи данных при частоте  $F_{\text{UART\_CLK}} = 4$  МГц.

Таблица 403 – Коэффициенты деления при частоте  $F_{\text{UART\_CLK}} = 4$  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
0x01A	0x03	9600	9598	0.021
0x068	0x0B	2400	2400	~0

### 24.12.7 Регистр управления линией LCR\_H

Данный регистр обеспечивает доступ к разрядам с 29 по 22 регистра LCR. При сбросе все биты регистра LCR\_H обнуляются.

Назначение разрядов регистра описано в таблице 404.

Таблица 404 – Регистр LCR\_H

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...9		Зарезервировано. Не модифицируйте. При чтении выдаются нули
8	WLEN[2]	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0 – длина слова определяется WLEN[1:0]; 1 – 9 бит
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 (таблица 405)
6...5	WLEN[1:0]	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 (таблица 405)
1	PEN	Разрешение проверки четности. 0 – кадр не содержит бита четности; 1 – бит четности передается в кадре и проверяется при приеме данных (таблица 405)

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
0	BRK	Разрыв линии. Если этот бит установлен в 1, то по завершении передачи текущего символа на выходе линии UART_TXD 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 LCR\_H, IBRD и FBRD совместно образует общий 31-разрядный регистр LCR, который обновляется по стробу, формируемому при записи в LCR\_H. Таким образом, для того, чтобы изменение параметров коэффициента деления частоты обмена данными вступило в силу, после их изменения значения регистров IBRD и/или FBRD необходимо осуществить запись данных в регистр LCR\_H.

Таблица 405 содержит данные об истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией LCR\_H.

Таблица 405 – 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**

1 Регистры LCR\_H, IBRD и FBRD не должны изменяться:

- при разрешенной работе приемопередатчика;
- во время завершения приема или передачи данных в процессе остановки (перевода в запрещенное состояние) приемопередатчика.

2 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.

**24.12.8 Регистр управления CR**

После сброса все биты регистра управления, за исключением битов 9 и 8 устанавливаются в состояние логического «0». Биты 9 и 8 устанавливаются в состояние логической «1».

Назначение разрядов регистра управления показано в таблице 406.

Таблица 406 – Регистр управления CR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15	CTSEn	Разрешение управления потоком данных по CTS. 1 – разрешено, данные передаются в линию только при активном значении сигнала nUARTCTS; 0 – запрещено
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; 0 – запрещено
13	Out2	Инверсия сигнала на линии состояния модема nUARTOut2. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «сигнал вызова» (RI). 1 – сигнал разрешен; 0 – сигнал запрещен
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). 1 – сигнал разрешен; 0 – сигнал запрещен
11	RTS	Инверсия сигнала на линии состояния модема nUARTRTS. 1 – сигнал разрешен; 0 – сигнал запрещен
10	DTR	Инверсия сигнала на линии состояния модема nUARTDTR. 1 – сигнал разрешен; 0 – сигнал запрещен
9	RXE	Разрешение приема. 1 – работа приемника разрешена. 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. 0 – работа приемника запрещена



Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8	TXE	<p>Разрешение передачи.</p> <p>1 – работа передатчика разрешена. Передача осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК-обмена SIR, в зависимости от значения бита SIREN.</p> <p>В случае перевода приемопередатчик в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой.</p> <p>0 – работа передатчика запрещена</p>
7	LBE	<p>Режим тестирования по шлейфу</p> <p>1 – шлейф разрешен;</p> <p>0 – запрещен.</p> <p>В режиме разрешенного шлейфа:</p> <p>Если установлены бит SIREN=1 и бит регистра управления тестированием TCR SIRTEST=1, то сигнал с выхода кодека nSIROUT инвертируется и подается на вход кодека SIRIN. 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.</p> <p>Если бит SIRTEST=0, то выходная линия передатчика UART_TXDx коммутируется на вход приемника UART_RXDx.</p> <p>Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии.</p> <p>После сброса бит устанавливается в 0</p>
6...3		Резерв. Не модифицируйте. При чтении выдаются нули.
2	SIRLP	<p>Выбор режима ИК-обмена с пониженным энергопотреблением.</p> <p>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.</p> <p>0 – длительность импульсов данных равна 3/16 длительности передачи бита</p>
1	SIREN	<p>Разрешение работы кодека ИК-передачи данных IrDA SIR:</p> <p>1 – разрешена работа кодека ИК. Данные передаются на выход nSIROUT и принимаются с входа SIRIN. Линия UART_TXDx находится в высоком состоянии. Данные на входе UART_RXDx и линиях состояния модема не обрабатываются.</p> <p>В случае если UARTEN=0 значение бита не играет роли.</p> <p>0 – запрещен. Сигнал nSIROUT находится в низком состоянии, данные на входе SIRIN не обрабатываются</p>

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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		

Для программирования регистров управления рекомендуется следующая последовательность действий:

- 1 Остановите работу приемопередатчика;
- 2 Дождаться окончания приема и/или передачи текущего символа данных;
- 3 Сбросьте буфер передатчика путем установки бита FEN регистра LCR\_N в 0;
- 4 Изменить настройки регистра CR;
- 5 Возобновите работу приемопередатчика.

#### 24.12.9 Регистр порога прерывания по заполнению буфера FIFO IFLS

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 407.

Таблица 407 – Регистр IFLS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		b100 = Буфер заполнен на 7/8 b101-b111 = резерв. Также стоит помнить, что в случае, когда сдвиговый регистр передатчика пуст, то слово, записанное в FIFO, будет сразу же переписано в сдвиговый регистр. Следовательно, для генерирования события прерывания от передатчика блока UART необходимо произвести запись в FIFO такого количества слов, которое превысит установленный порог хотя бы на одно слово с учетом описанного случая

#### 24.12.10 Регистр установки сброса маски прерывания IMSC

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 IMSC показано в таблице 408.

Таблица 408 – Регистр IMSC

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14		Резерв. Не модифицируйте. При чтении выдаются нули
13	TNBSYIM	Маска прерывания при отсутствии данных в сдвиговом регистре передатчика UARTTNBSYINTR: 1 – установлена; 0 – сброшена
12	TFEIM	Маска прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR: 1 – установлена; 0 – сброшена
11	RNEIM	Маска прерывания по наличию данных в FIFO приемника UARTRNEINTR: 1 – установлена; 0 – сброшена
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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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

### 24.12.11 Регистр состояния прерываний RIS

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 RIS представлено в таблице 409.

Таблица 409 – Регистр RIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14		Резерв. Не модифицируйте. При чтении выдаются нули
13	TNBSYRIS	Состояние прерывания при отсутствии данных в сдвиговом регистре передатчика UARTTNBSYINTR
12	TFERIS	Состояние прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
11	RNERIS	Состояние прерывания по наличию данных в FIFO приемника UARTRNEINTR
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 UARTRTINTR*: 1 – вышло 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
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: 1 – возникло прерывание; 0 – прерывание не возникало
2	DCDRMIS	Состояние прерывания UARTDCDINTR по изменению линии nUARTDCD: 1 – возникло прерывание; 0 – прерывание не возникало
1	CTSRMIS	Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS: 1 – возникло прерывание; 0 – прерывание не возникало
0	RIRMIS	Состояние прерывания UARTRIINTR по изменению линии nUARTRI: 1 – возникло прерывание; 0 – прерывание не возникало

\* Сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Поэтому чтение состояния прерывания по таймауту из регистров MIS и RIS даст одинаковый результат

### 24.12.12 Регистр маскированного состояния прерываний MIS

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 MIS представлено в таблице 410.

Таблица 410 – Регистр MIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14		Резерв. Не модифицируйте. При чтении выдаются нули
13	TNBSYMIS	Маскированное состояние прерывания при отсутствии данных в сдвиговом регистре передатчика UARTTNBSYINTR
12	TFEMIS	Маскированное состояние прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR
11	RNEMIS	Маскированное состояние прерывания по наличию данных в FIFO приемника UARTRNEINTR
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 – вышло 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	DSRMMIS	Маскированное состояние прерывания UARTDSRINTR по изменению линии nUARTDSR: 1 – возникло прерывание; 0 – прерывание не возникало
2	DCDMMIS	Маскированное состояние прерывания UARTDCDINTR по изменению линии nUARTDCD: 1 – возникло прерывание; 0 – прерывание не возникало
1	CTSMMIS	Маскированное состояние прерывания UARTCTSINTR по изменению линии nUARTCTS: 1 – возникло прерывание; 0 – прерывание не возникало
0	RIMMIS	Маскированное состояние прерывания UARTRIINTR по изменению линии nUARTRI: 1 – возникло прерывание; 0 – прерывание не возникало

### 24.12.13 Регистр сброса прерываний ICR

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 ICR представлено в таблице 411.

Таблица 411 – Регистр ICR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...11		Резерв. Не модифицируйте. При чтении выдаются нули.
10	OEIC	Сброс прерывания по переполнению буфера UARTOEINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
9	BEIC	Сброс прерывания по разрыву линии UARTBEINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
8	PEIC	Сброс прерывания по ошибке контроля четности UARTPEINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
7	FEIC	Сброс прерывания по ошибке в структуре кадра UARTFEINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
6	RTIC	Сброс прерывания по таймауту приема данных UARTRTINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
5	TXIC	Сброс прерывания от передатчика UARTTXINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
4	RXIC	Сброс прерывания от приемника UARTRXINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
3	DSRMIC	Сброс прерывания UARTDSRINTR по изменению линии nUARTDSR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
2	DCDMIC	Сброс прерывания UARTDCDINTR по изменению линии nUARTDCD: 1 – сброс прерывания; 0 – не влияет на состояние регистра
1	CTSMIC	Сброс прерывания UARTCTSINTR по изменению линии nUARTCTS: 1 – сброс прерывания; 0 – не влияет на состояние регистра.
0	RIMIC	Сброс прерывания UARTRIINTR по изменению линии nUARTRI: 1 – сброс прерывания; 0 – не влияет на состояние регистра

#### 24.12.14 Регистр управления прямым доступом к памяти DMACR

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра DMACR представлено в таблице 412.

Таблица 412 – Регистр DMACR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...3		Резерв. Не модифицируйте. При чтении выдаются нули
2	DMAONERR	Блокирование DMA запросов при приеме при возникновении ошибок на линии. 1 – в случае возникновения прерывания по обнаружению ошибки блокируются запросы DMA от приемника UARTRXDMASREQ и UARTRXDMAABREQ. 0 – DMA запросы не блокируются
1	TXDMAE	Использование DMA при передаче. 1 – разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика. 0 – запрещено формирование DMA запросов
0	RXDMAE	Использование DMA при приеме. 1 – разрешено формирование запросов DMA для обслуживания буфера FIFO приемника; 0 – запрещено формирование DMA запросов



### 24.12.15 Регистр управления тестированием TCR

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра TCR представлено в таблице 413.

Таблица 413 – Регистр TCR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...13		Резерв. При чтении значение непредсказуемо.
12...3	-	Зарезервировано
2	SIRTEST	Разрешение приема данных в кольцевом режиме с выхода IrDA передатчика: 1 – разрешено; 0 – запрещено. Используется совместно с установкой бита LBE в регистре CR
1	TESTFIFO	Разрешение чтения данных из FIFO передатчика и запись в FIFO приемника: 1 – разрешено; 0 – запрещено
0	ITEN	Перевод контроллера UART в тестовый режим: 1 – тестовый режим разрешен; 0 – тестовый режим запрещен

## 25 Контроллер прямого доступа в память DMA

### 25.1 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.

### 25.2 Термины и определения

Таблица 414 – Термины и определения

Альтернативная	Альтернативная структура управляющих данных канала. Вы можете установить соответствующий регистр для изменения типа структуры данных (см. подраздел 25.5 «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
N	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 <code>chnl_pri_alt_set</code> 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 10
Исполнение 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 <code>dma_done</code> , если окончание подобного режима работы происходит после выполнения обычного цикла

### 25.3 Функциональное описание

На рисунке 85 показана упрощенная структурная схема контроллера.

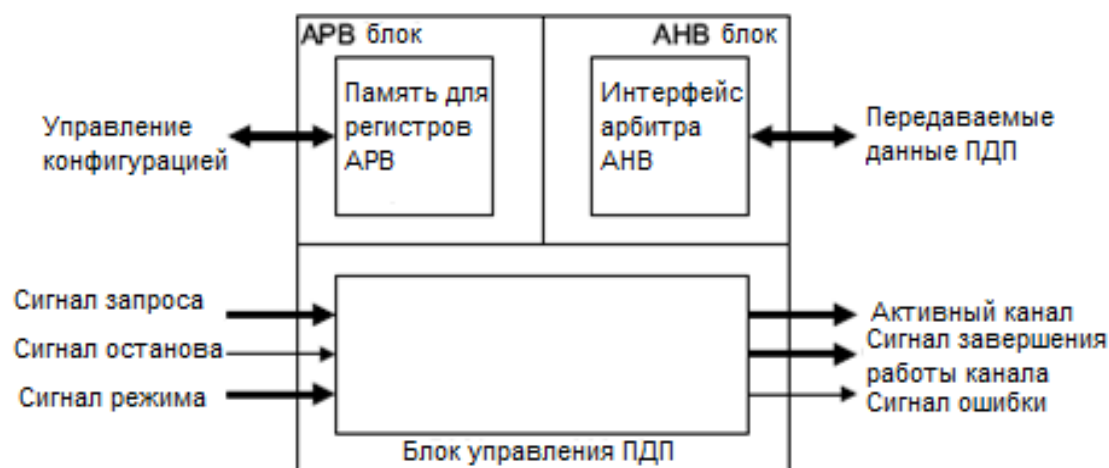


Рисунок 85 – Структурная схема контроллера

Контроллер состоит из следующих основных функциональных блоков:

- блок, подключенный к шине АРВ;
- блок, подключенный к шине АНВ;
- управляющий блок DMA.

#### 25.3.1 Распределение каналов DMA

Таблица 415 – Распределение каналов DMA

Номер запроса	dma_req	dma_sreq	Описание
0	UART1TXDMABREQ	UART1TXDMASREQ	Запрос от передатчика UART1
1	UART1RXDMABREQ	UART1RXDMASREQ	Запрос от приемника UART1
2	UART2TXDMABREQ	UART2TXDMASREQ	Запрос от передатчика UART2
3	UART2RXDMABREQ	UART2RXDMASREQ	Запрос от приемника UART2
4	SSP1TXDMABREQ	SSP1TXDMASREQ	Запрос от передатчика SSP1
5	SSP1RXDMABREQ	SSP1RXDMASREQ	Запрос от приемника SSP1
6	CRCDMABREQ	CRCDMASREQ	Запрос от аппаратного блока вычисления CRC
7	UART3TXDMABREQ	UART3TXDMASREQ	Запрос от передатчика UART3
8	UART3RXDMABREQ	UART3RXDMASREQ	Запрос от приемника UART3
9	TMR3DMAREQ	TMR3DMAREQ	Запрос от таймера общего назначения TIMER3
10	TMR1DMAREQ	TMR1DMAREQ	Запрос от таймера общего назначения TIMER1
11	TMR2DMAREQ	TMR2DMAREQ	Запрос от таймера общего назначения TIMER2
12	ADCUIDMABREQ1 (I0)	ADCUIDMASREQ1 (I0)	Запросы от блока АЦП для измерения напряжений и токов в электрической сети
13	ADCUIDMABREQ2 (V0)	ADCUIDMASREQ2 (V0)	

Номер запроса	dma_req	dma_sreq	Описание
14	ADCUIDMABREQ3 (I1)	ADCUIDMASREQ3 (I1)	
15	ADCUIDMABREQ4 (V1)	ADCUIDMASREQ4 (V1)	
16	ADCUIDMABREQ5 (I2)	ADCUIDMASREQ5 (I2)	
17	ADCUIDMABREQ6 (V2)	ADCUIDMASREQ6 (V2)	
18	ADCUIDMABREQ7 (I3)	ADCUIDMASREQ7 (I3)	
19	SSP2TXDMABREQ	SSP2TXDMASREQ	Запрос от передатчика SSP2
20	SSP2RXDMABREQ	SSP2RXDMASREQ	Запрос от приемника SSP2
21	SSP3TXDMABREQ	SSP3TXDMASREQ	Запрос от передатчика SSP3
22	SSP3RXDMABREQ	SSP3RXDMASREQ	Запрос от приемника SSP3
23	TMR4DMAREQ	TMR4DMAREQ	Запрос от таймера общего назначения TIMER4
24	UART4TXDMABREQ	UART4TXDMASREQ	Запрос от передатчика UART4
25	UART4RXDMABREQ	UART4RXDMASREQ	Запрос от приемника UART4
26	-	-	-
27	-	-	-
28	-	-	-
29	-	-	-
30	-	ADCDMASREQ	Запрос от АЦП последовательных приближений
31	-	-	-

### 25.3.2 Блок, подключенный к шине APB

Блок содержит набор регистров, позволяющих настраивать контроллер, используя ведомый APB интерфейс. Регистры занимают адресное пространство емкостью 4 Кбайт.

### 25.3.3 Блок, подключенный к шине AHB

Контроллер содержит один блок типа «ведущий» шины DMA Bus, который позволяет, используя 32-разрядную шину, передавать данные от источника к приемнику. Источник и приемник являются ведомыми шины AHB.

### 25.3.4 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.

### 25.3.5 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 АНВ, где блоки должны «захватывать» шину или обращаться к внешней памяти. В тоже время контроллер DMA предназначен для использования в многоуровневых системах с шиной АНВ, включающих встроенную память.

### 25.3.6 Разрядность передаваемых данных

Контроллер интерфейса предоставляет возможность осуществлять передачу 8, 16 и 32 разрядных данных. Таблица 416 отображает значения комбинаций шины HSIZE.

Таблица 416 – Комбинации шины HSIZE

HSIZE[2]*	HSIZE[1]	HSIZE[0]	Разрядность данных (бит)
0	0	0	8
0	0	1	16
0	1	0	32
0	1	1	**
* Сигнал постоянно удерживается в состоянии логический ноль; ** Запрещенная комбинация			

Контроллер всегда использует передачи 32-разрядными данными при обращении к управляющим данным канала. Необходимо устанавливать разрядность данных источника соответствующую разрядности данных приемника.

### 25.3.7 Управление защитой данных

Контроллер позволяет устанавливать режимы защиты данных протокола АНВ-Lite, определяемые шиной HPROT[3:1]. Возможен выбор следующих режимов защиты:

- кэширование;
- буферизация;
- привилегированный.

Таблица 417 перечисляет значения комбинаций шины HPROT.

Таблица 417 – Режимы защиты данных

HPROT[3] кэширование	HPROT[2] буферизация	HPROT[1] привилегированный	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 подразделе 25.4 «Управление DMA»

### 25.3.8 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 таблице 418 перечислены возможные комбинации.

Таблица 418 – 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 подразделе 25.5 «Структура управляющих данных канала». 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. подраздел 25.5 «Структура управляющих данных канала»).

## 25.4 Управление DMA

### 25.4.1 Правила обмена данными

Контроллер использует правила обмена данными (см. таблицу 419), 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]`;

Таблица 419 – Правила, при которых передача данных по каналам разрешена, и запросы не маскируются

Правило	Описание
1	Если <code>dma_active[C]</code> установлен в 0, то установка в 1 <code>dma_req[C]</code> или <code>dma_sreq[C]</code> 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 <code>dma_active[C]</code>
3	Контроллер устанавливает в 1 <code>dma_active[C]</code> в момент начала передачи по каналу C
4	Для типов циклов DMA, отличных от периферийного «Исполнение с изменением конфигурации», <code>dma_active[C]</code> 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 <code>n_minus_1</code> . В периферийном режиме «Исполнение с изменением конфигурации», <code>dma_active[C]</code> 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 <code>n_minus_1</code> ), используя альтернативную структуру управляющих данных. По окончании последней передачи <code>dma_active[C]</code> сбрасывается в 0
5	Контроллер устанавливает <code>dma_active[C]</code> в 0 как минимум на один такт сигнала HCLK перед тем, как снова установит <code>dma_active[C]</code> или <code>dma_active[]</code> в 1
6	Для каналов, по которым разрешена передача, контроллер осуществляет установку в 1 только одного <code>dma_done[]</code>
7	Если <code>dma_req[C]</code> устанавливается в состояние 1 в момент, когда <code>dma_active[C]</code> или <code>dma_stall</code> также в состоянии 1, то это означает, что контроллер обнаружил запрос
8	Если разряды <code>cycle_ctrl</code> для канала установлены в состояние <code>3'b100</code> , <code>3'b101</code> , <code>3'b110</code> , <code>3'b111</code> , то <code>dma_done[C]</code> никогда не будет установлен в 1
9	Если все передачи по каналу завершены, и разряды <code>cycle_ctrl</code> позволяют удержание <code>dma_done[C]</code> , то по срезу сигнала <code>dma_active[]</code> произойдут события: – если <code>dma_stall</code> в состоянии 0, контроллер устанавливает <code>dma_done[]</code> в состояние 1 продолжительностью один такт HCLK



Правило	Описание
	– 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

Правило	Описание
18	Когда <code>chnl_req_mask_set[C]</code> установлен в 1, контроллер игнорирует запросы по <code>dma_sreq[C]</code> и <code>dma_req[C]</code>
<p>* Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре <code>n_minus_1</code> меньше, чем значение <math>2^R</math>, то контроллер не очистит разряды <code>chnl_useburst_set</code> и поэтому запросы по <code>dma_sreq[C]</code> будут маскированы. Если периферия не устанавливает <code>dma_req[C]</code> в состояние 1, то контроллер никогда не выполнит необходимых передач</p>	

При отключении канала контроллер осуществляет DMA передачи согласно правилам, представленным в таблице 420.

Таблица 420 – Правила осуществления DMA передач при «запрещенных» каналах

Правило	Описание
19	Если <code>dma_req[C]</code> установлен в 1, то контроллер устанавливает <code>dma_done[C]</code> в 1. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен)
20	Если <code>dma_sreq[C]</code> установлен в 1, то контроллер устанавливает <code>dma_done[C]</code> в 1 при условии <code>dma_waitonreq[C]</code> в 1 и <code>chnl_useburst_set[C]</code> в состоянии 0. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен)
21	<code>dma_active[C]</code> всегда удерживается в состоянии 0

## 25.4.2 Диаграммы работы контроллера DMA

Данный раздел описывает следующие примеры функционирования контроллера с использованием правил обмена данными (см. таблицу 419):

- импульсный запрос на обработку;
- запрос по уровню на обработку;
- флаги завершения;
- флаги ожидания запроса на обработку.

Примечание – Все диаграммы, показанные на рисунках 86 – 89, подразумевают следующее:

- `hready` находится в состоянии 1;
- АНВ «ведомый» всегда дает ответ «ОКAY».

### 25.4.2.1 Импульсный запрос на обработку

Рисунок 86 показывает временную диаграмму работы контроллера DMA при получении импульсного запроса от периферии.

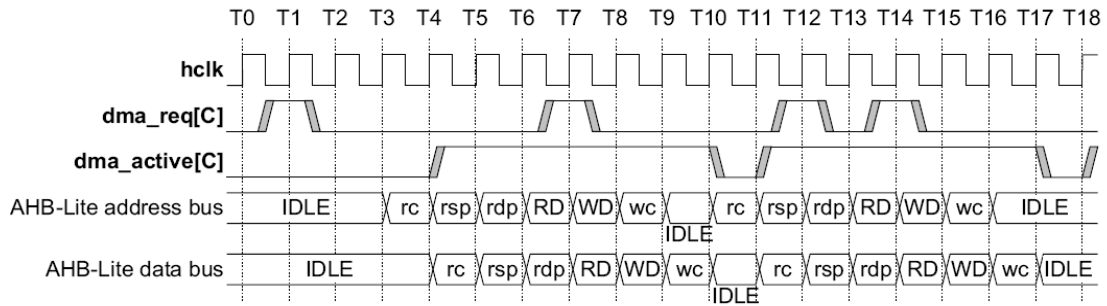


Рисунок 86 – Диаграмма работы при получении импульсного запроса от периферийного блока

Таблица 421 – Пояснения к диаграмме работы при получении импульсного запроса

T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что <code>chnl_req_mask_set[C]</code> находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает <code>dma_active[C]</code> (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, <code>channel_cfg</code> ; rsp – чтение указателя адреса окончания данных источника, <code>src_data_end_ptr</code> ; rdp – чтение указателя адреса окончания данных приемника, <code>dst_data_end_ptr</code>
T7	При установленном <code>dma_active[C]</code> в 1 и при условии, что <code>chnl_req_mask_set[C]</code> 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, <code>channel_cfg</code> , где wc – запись настроек канала, <code>channel_cfg</code>
T10	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4)
T10-T11	Контроллер удерживает <code>dma_active[C]</code> в 0 как минимум на один такт HCLK (см. правило 5)
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает <code>dma_active[C]</code> , так как ранее на такте T7 был получен запрос на обработку (см. правила 2 и 3)
T12	При установленном <code>dma_active[C]</code> в 1 и при условии, что <code>chnl_req_mask_set[C]</code> 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 <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4)
T17-T18	Контроллер удерживает <code>dma_active[C]</code> как минимум на один такт HCLK (см. правило 5)
T18	Если канал C имеет более высокий приоритет, то контроллер устанавливает <code>dma_active[C]</code> , так как ранее на такте T12 был получен запрос на обработку (см. правила 2 и 3)

25.4.2.2 Запрос на обработку по уровню.

Рисунок 87 показывает временную диаграмму работы контроллера DMA при получении от периферии запроса на обработку по уровню.

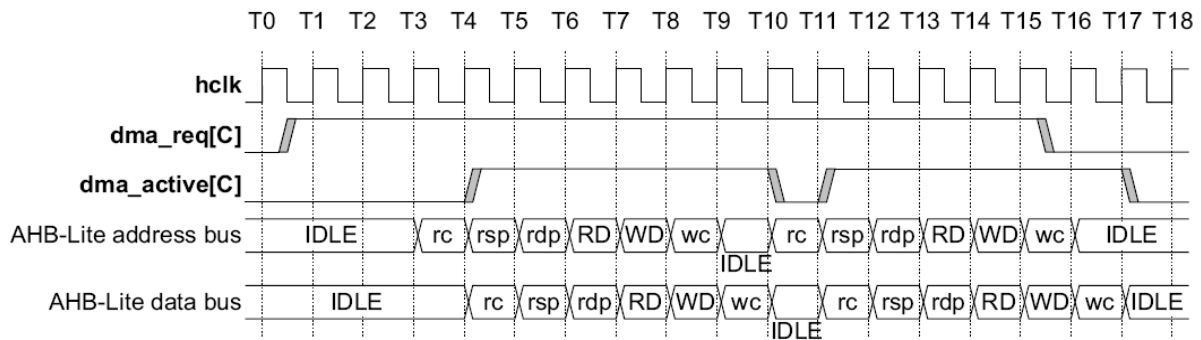


Рисунок 87 – Диаграмма работы при получении запроса на обработку по уровню

Таблица 422 – Пояснения к диаграмме работы при получении запроса на обработку по уровню

T1	Контроллер обнаружил запрос на обработку по каналу C (Таблица 419, правило 1) при условии, что <code>chnl_req_mask_set[C]</code> находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает <code>dma_active[C]</code> (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, <code>channel_cfg</code> ; rsp – чтение указателя адреса окончания данных источника, <code>src_data_end_ptr</code> ; rdp – чтение указателя адреса окончания данных приемника, <code>dst_data_end_ptr</code> .
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных.
T9-T10	Контроллер осуществляет запись настроек канала, <code>channel_cfg</code> , где wc – запись настроек канала, <code>channel_cfg</code>
T10	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4). Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что <code>chnl_req_mask_set[C]</code> находится в состоянии 0 (см. правило 18).
T10-T11	Контроллер удерживает <code>dma_active[C]</code> в 0 как минимум на один такт HCLK (см. правило 5)
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает <code>dma_active[C]</code> 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 <code>dma_req[C]</code>
T16-T17	Контроллер осуществляет запись настроек канала <code>channel_cfg</code>
T17	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4)

При использовании запроса на обработку по уровню, периферийный блок может не обладать достаточным быстродействием, чтобы вовремя снять сигнал запроса, в этом случае он должен установить сигнал `dma_stall`. Установка сигнала `dma_stall` предотвращает повторение выполненной передачи.

### 25.4.2.3 Флаги завершения

Рисунок 88 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.

Выставление сигнала `dma_done` транслируется в регистр `DMA_DONE_STICK` блока контроллера тактовой частоты (см. пункт 16.6.9 «`DMA_DONE_STICK`»).

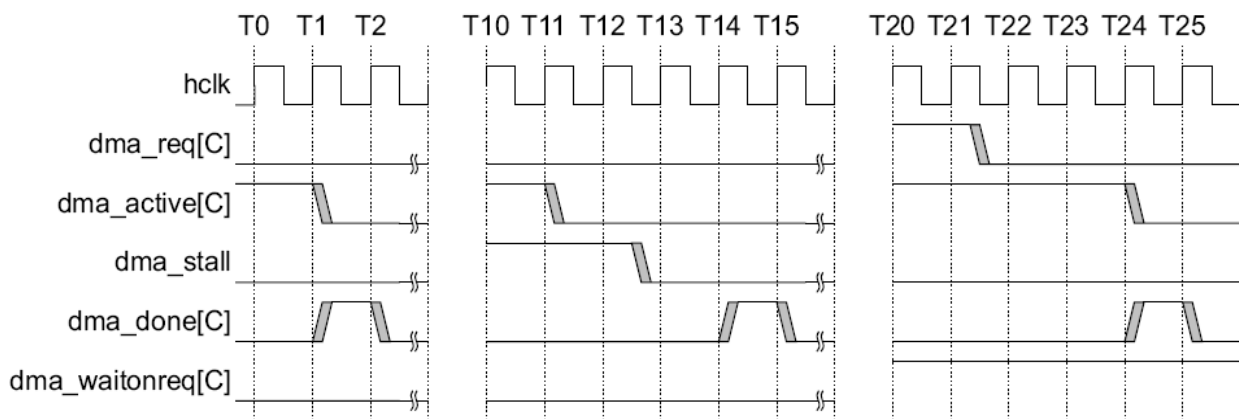


Рисунок 88 – Диаграммы функционирования `dma_done`

Таблица 423 – Пояснения функционирования `dma_done`, такты от T0 до T2

T1	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4)
T1-T2	Контроллер завершает цикл DMA, и если <code>cycle_ctrl[2]</code> установлен в 0, то устанавливает в 1 <code>dma_done[C]</code> на один такт HCLK (см. правила 8 и 9). Для других разрешенных каналов сигнал <code>dma_done[C]</code> останется в состоянии 0 (см. правило 6)

Таблица 424 – Пояснения функционирования `dma_done`, такты от T10 до T15

T11	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4)
T12-T13	Периферийный блок сбрасывает сигнал <code>dma_stall</code>
T14-T15	Контроллер завершает цикл DMA, и если <code>cycle_ctrl[2]</code> установлен в 0, то устанавливает в 1 <code>dma_done[C]</code> на один такт HCLK (см. правила 8 и 9). Для других разрешенных каналов сигнал <code>dma_done[C]</code> 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).

Таблица 425 – Пояснения функционирования dma\_done, такты 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)

#### 25.4.2.4 Флаги ожидания запроса на обработку

Рисунки 89, 90 демонстрируют примеры использования флагов ожидания запроса на обработку при выполнении 2<sup>R</sup> 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.

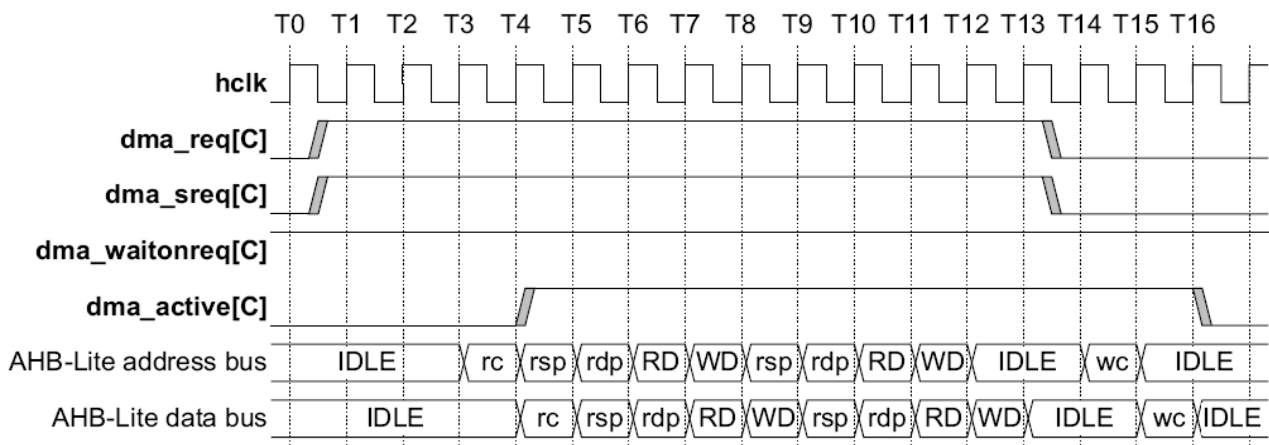


Рисунок 89 – Диаграмма работы контроллера DMA при использовании dma\_waitonreq

Таблица 426 – Пояснения работы контроллера DMA при использовании dma\_waitonreq

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 <sup>R</sup> (см. правило 15)

Рисунок 90 показывает работу контроллера DMA при установке dma\_waitonreq в 1 и выполнении одиночной передачи DMA.

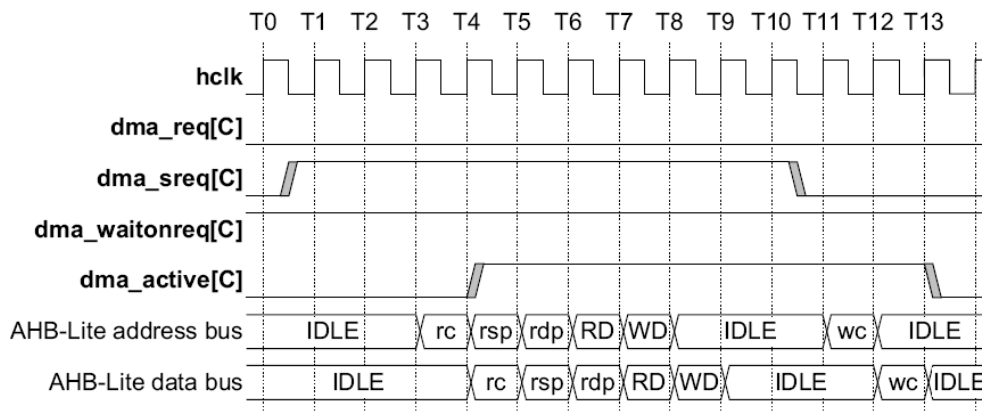


Рисунок 90 – Работа DMA при использовании dma\_waitonreq совместно с dma\_sreq

Таблица 427 – Пояснения работы DMA при использовании dma\_waitonreq совместно с dma\_sreq

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)

### 25.4.3 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 четыре разряда, которые определяют количество транзакций по шине АНВ 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 428 показывает возможную периодичность арбитража.

Таблица 428 – Периодичность арбитража в единицах передач по шине АНВ

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 подразделе 25.5 «Структура управляющих данных канала».



#### 25.4.4 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 429 показывает уровень приоритета каналов DMA в порядке его уменьшения.

Таблица 429 – 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 91 показывает процесс выбора следующего канала для обслуживания.

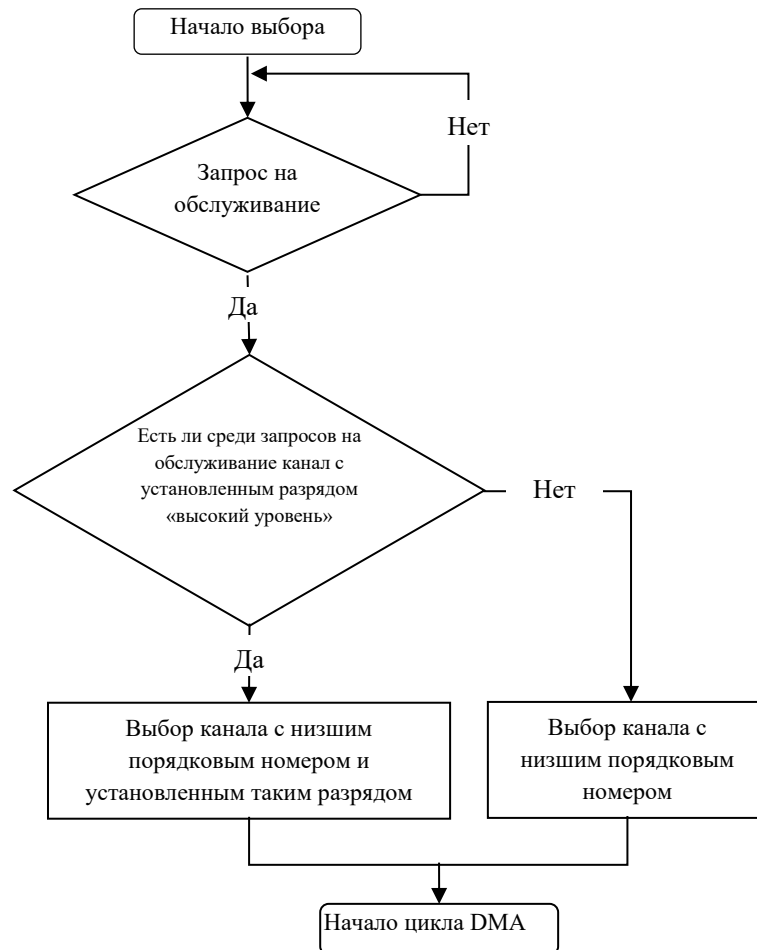


Рисунок 91 – Алгоритм выбора следующего канала для обслуживания

### 25.4.5 Типы циклов DMA

Разряды `cycle_ctrl` определяют, как контроллер будет выполнять циклы DMA. Описание значений этих разрядов приведено ниже.

Таблица 430 – 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 <code>cycle_ctrl</code> находятся в области памяти, отведенной под <code>channel_cfg</code> – 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».

#### **25.4.5.1 Недействительный**

После окончания цикла DMA контроллер устанавливает тип цикла в значение «недействительный» для предотвращения повтора выполненного цикла DMA.

#### **25.4.5.2 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.

#### **25.4.5.3 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 Контроллер осуществляет арбитраж. Когда канал C становится каналом с самым высоким приоритетом, 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.

#### 25.4.5.4 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 «недействительную» структуру данных или пока процессор не запретит работу канала.

Рисунок 92 демонстрирует пример функционирования контроллера в режиме «ПИНГ-ПОНГ».

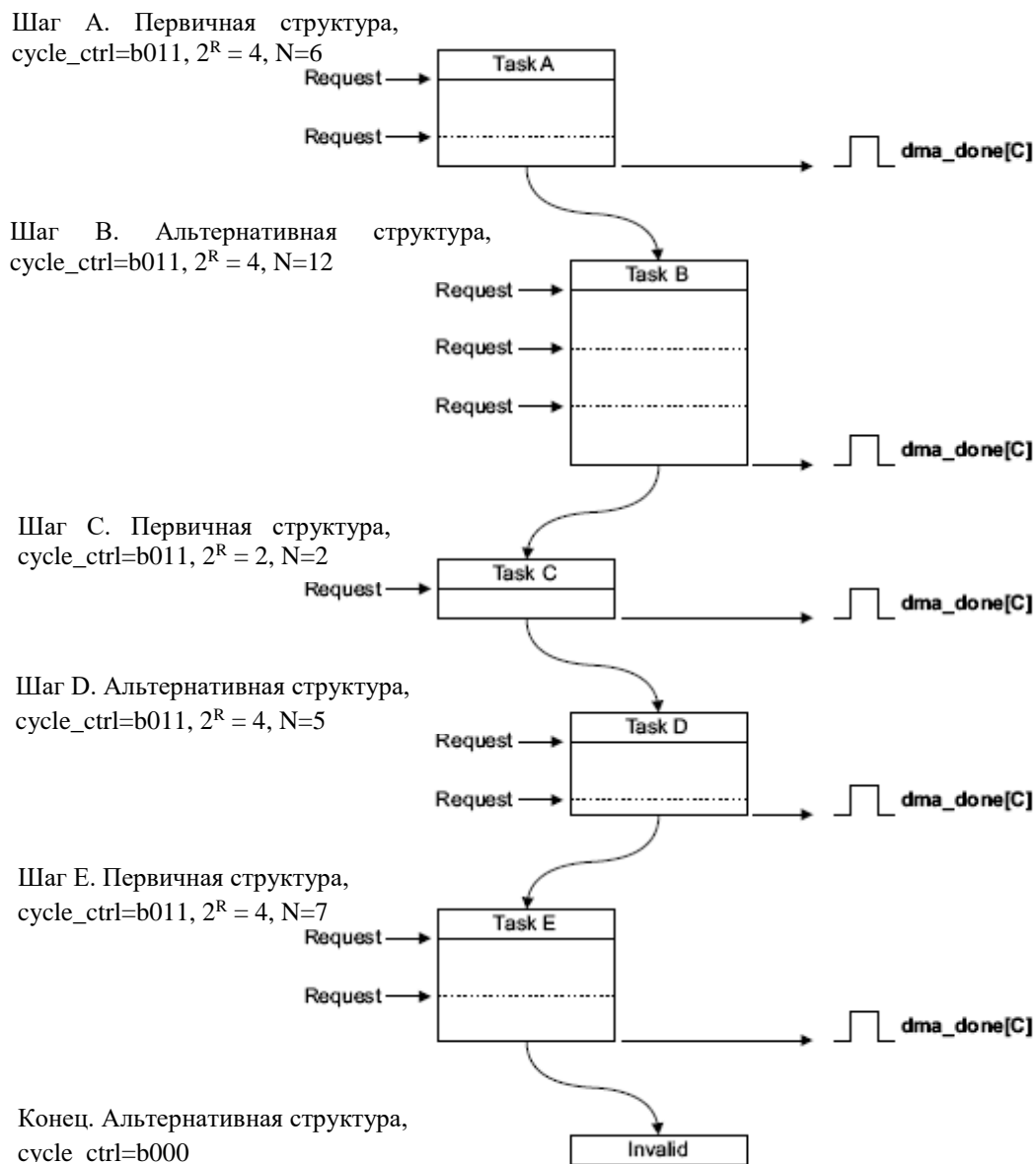


Рисунок 92 – Пример функционирования контроллера в режиме «пинг-понг»

Пояснения к рисунку 92:

Шаг А Процессор устанавливает первичную структуру управляющих данных для шага А. Процессор устанавливает альтернативную структуру управляющих данных для шага В. Это позволит контроллеру переключиться к шагу В незамедлительно после выполнения шага А, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.  
Контроллер получает запрос и выполняет четыре 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 две передачи DMA.  
Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража

После выполнения шага А процессор может установить первичные управляющие данные канала для шага С. Это позволит контроллеру переключиться к шагу С незамедлительно после выполнения шага В, 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 В:

Шаг В Контроллер выполняет четыре передачи DMA.  
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.  
Контроллер выполняет четыре 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 четыре передачи DMA.  
Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража.

После выполнения шага В 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 С:

Шаг С Контроллер выполняет две передачи DMA.  
Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража.

После выполнения шага С процессор может установить первичные управляющие данные канала для шага Е.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг D:

- Шаг D    Контроллер выполняет четыре 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 четыре 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 три 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 окончании 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.

#### **25.4.5.5    *Режим работы с памятью «исполнение с изменением конфигурации»***

В данном режиме контроллер, получая начальный запрос на обработку, выполняет четыре передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные. Затем контроллер выполняет еще четыре 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 431 перечисляет области памяти channel\_cfg, те которые должны быть определены константами, и те, значения которых определяются пользователем.

Таблица 431 – 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 четыре передачи DMA
3	next_useburst	1'b0	Для данного режима этот разряд должен быть равен 0
2...0	cycle_ctrl	3'b100	Контроллер 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 четырем. Число равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные			

Рисунок 93 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<sup>R</sup>=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 <sup>R</sup> = 4, N = 3	0xFFFFFFFF
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b101, 2 <sup>R</sup> = 2, N = 8	0xFFFFFFFF
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b101, 2 <sup>R</sup> = 8, N = 5	0xFFFFFFFF
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2 <sup>R</sup> = 4, N = 4	0xFFFFFFFF

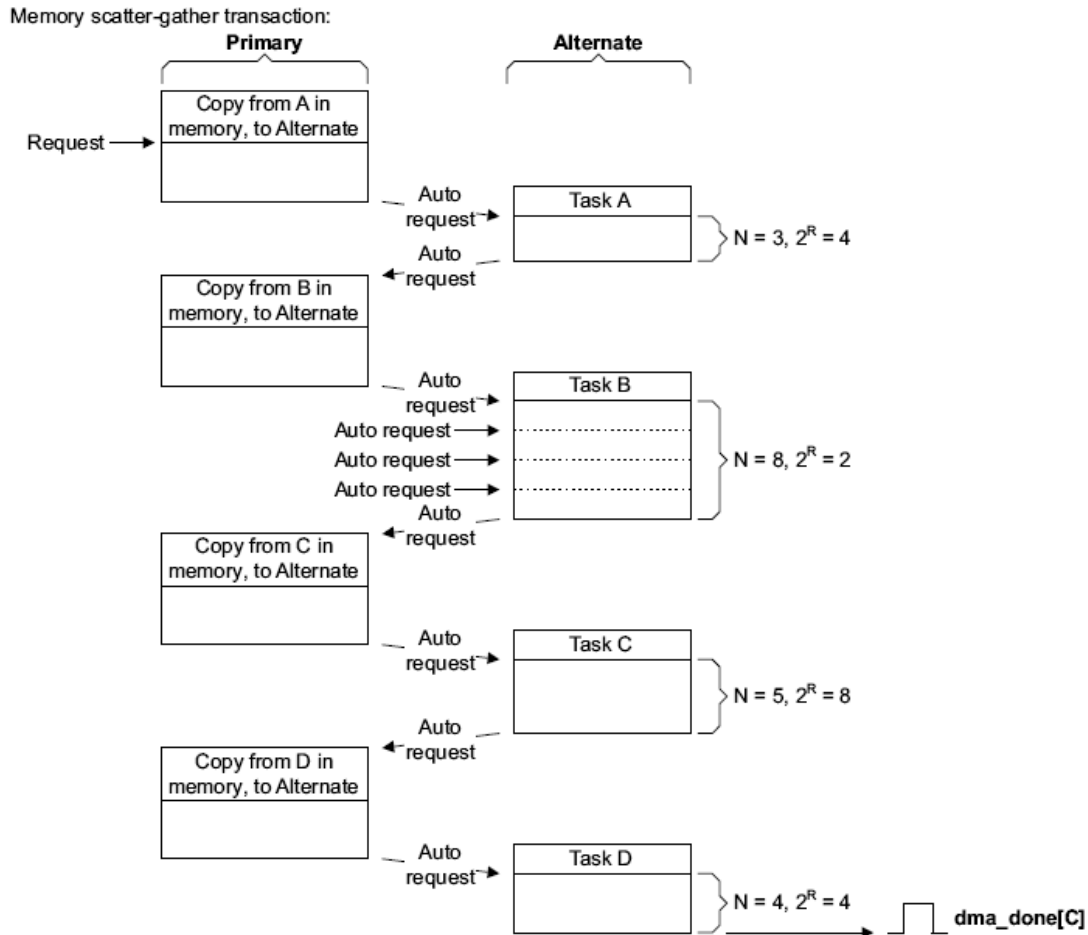


Рисунок 93 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

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 четырех слов, необходимо установить 2<sup>R</sup> в 4. В этом примере количество передач равно четырем 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 А**

По получению запроса на обслуживание контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага А.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

**Шаг А**

Контроллер выполняет шаг А. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

**Первичная, копирование В**

Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага В.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

**Шаг В**

Контроллер выполняет шаг В. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

**Первичная, копирование С**

Контроллер выполняет 4 четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага С.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

**Шаг С**

Контроллер выполняет шаг С. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

**Первичная, копирование D**

Контроллер выполняет четыре 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.

**25.4.5.6 Режим работы с периферией «исполнение с изменением конфигурации»**

В данном режиме контроллер, получая начальный запрос на обработку, выполняет четыре 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 четыре 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 432 перечисляет области памяти channel\_cfg, которые должны быть определены константами, и те области, значения которых определяются пользователем.

Таблица 432 – 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 четыре передачи DMA
2...0	cycle_ctrl	3'b110	Контролер работает в режиме работы с периферией «исполнение с изменением конфигурации»

Разряды	Обозначение	Значение	Описание
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
<p>* Так как разряды R_power установлены в состояние 2, необходимо задавать значение N кратное четырем. Число равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные</p>			

Рисунок 94 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<sup>R</sup>=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 <sup>R</sup> = 4, N = 3	0xFFFFFFFF
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b111, 2 <sup>R</sup> = 2, N = 8	0xFFFFFFFF
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b111, 2 <sup>R</sup> = 8, N = 5	0xFFFFFFFF
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2 <sup>R</sup> = 4, N = 4	0xFFFFFFFF

Peripheral scatter-gather transaction:

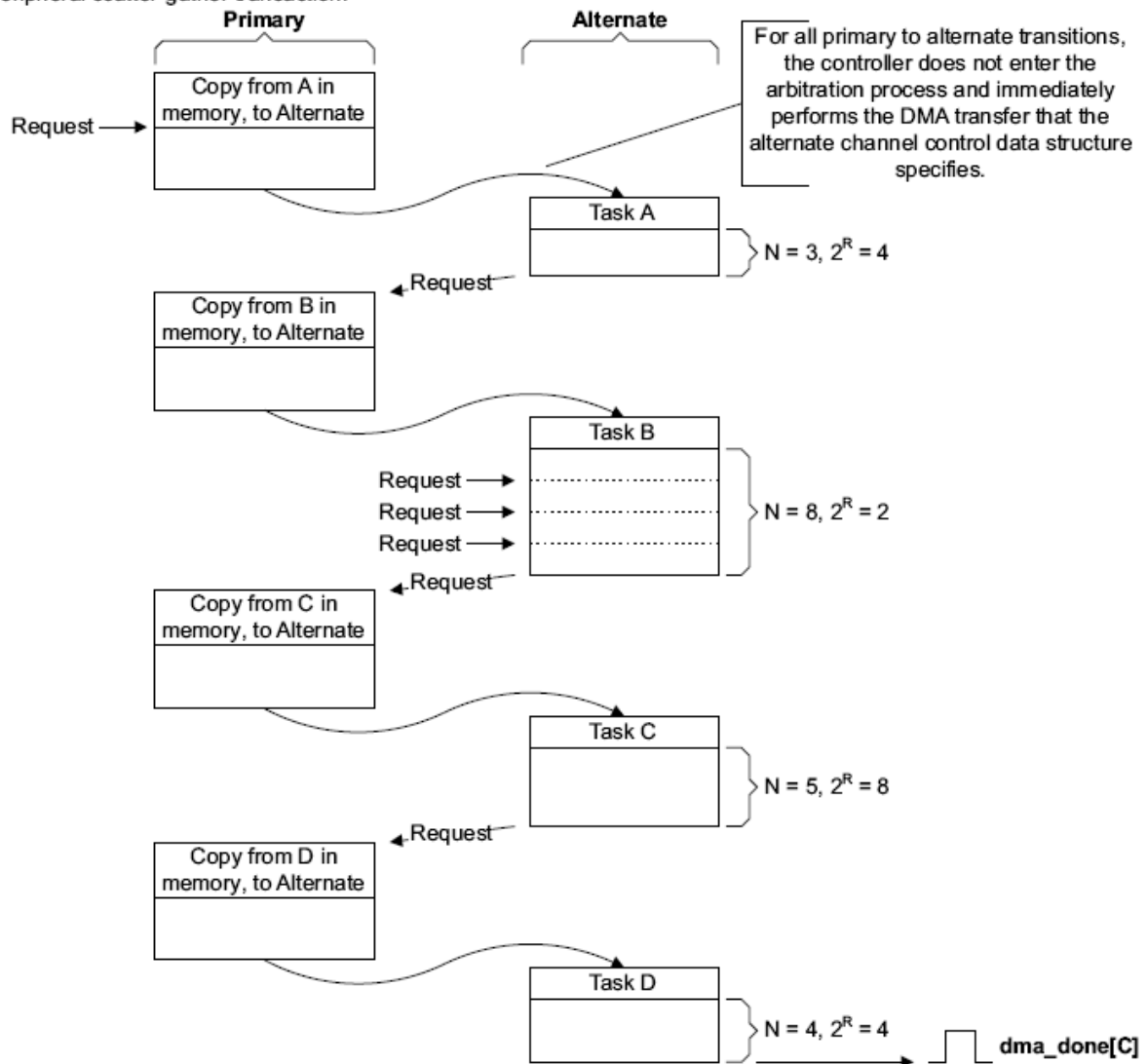


Рисунок 94 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

Пояснения к рисунку:

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 четырех слов, необходимо установить 2<sup>R</sup> в 4. В этом примере количество задач равно четырем и поэтому N установлено в 16.

2 Процессор записывает управляющие данные для шагов А, В, С, 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 А памяти**

По получению запроса на обслуживание, контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага А.

**Шаг А**

Контроллер выполняет шаг А.

По окончании контроллер проводит процедуру арбитража.

Первичная, копирование из области В памяти.

Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага В.

**Шаг В**

Контроллер выполняет шаг В. Для завершения задачи периферия должна установить последовательно три запроса.

По окончании контроллер проводит процедуру арбитража.

Первичная, копирование из области С памяти.

Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага С.

**Шаг С**

Контроллер выполняет шаг С.

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 четыре 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.

**25.4.5.7 Индикация ошибок**

При получении контроллером по шине АНВ 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:

1 Чтение регистра `chnl_enable_set` с целью создания списка отключенных каналов;

2 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[]`. Выставление сигнала `dma_done` транслируется в регистр `DMA_DONE_STICK` блока контроллера тактовой частоты (см. пункт 16.6.9 «`DMA_DONE_STICK`»);

3 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.

## 25.5 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 95 показывает область памяти необходимую контроллеру для структур управляющих данных канала, при использовании всех 32 каналов и опциональной альтернативной структуры управляющих данных.

Alternate data structure		Primary data structure	
Alternate_Ch_31	0x3F0	Primary_Ch_31	0x1F0
Alternate_Ch_30	0x3E0	Primary_Ch_30	0x1E0
Alternate_Ch_29	0x3D0	Primary_Ch_29	0x1D0
Alternate_Ch_28	0x3C0	Primary_Ch_28	0x1C0
Alternate_Ch_27	0x3B0	Primary_Ch_27	0x1B0
Alternate_Ch_26	0x3A0	Primary_Ch_26	0x1A0
Alternate_Ch_25	0x390	Primary_Ch_25	0x190
Alternate_Ch_24	0x380	Primary_Ch_24	0x180
Alternate_Ch_23	0x370	Primary_Ch_23	0x170
Alternate_Ch_22	0x360	Primary_Ch_22	0x160
Alternate_Ch_21	0x350	Primary_Ch_21	0x150
Alternate_Ch_20	0x340	Primary_Ch_20	0x140
Alternate_Ch_19	0x330	Primary_Ch_19	0x130
Alternate_Ch_18	0x320	Primary_Ch_18	0x120
Alternate_Ch_17	0x310	Primary_Ch_17	0x110
Alternate_Ch_16	0x300	Primary_Ch_16	0x100
Alternate_Ch_15	0x2F0	Primary_Ch_15	0x0F0
Alternate_Ch_14	0x2E0	Primary_Ch_14	0x0E0
Alternate_Ch_13	0x2D0	Primary_Ch_13	0x0D0
Alternate_Ch_12	0x2C0	Primary_Ch_12	0x0C0
Alternate_Ch_11	0x2B0	Primary_Ch_11	0x0B0
Alternate_Ch_10	0x2A0	Primary_Ch_10	0x0A0
Alternate_Ch_9	0x290	Primary_Ch_9	0x090
Alternate_Ch_8	0x280	Primary_Ch_8	0x080
Alternate_Ch_7	0x270	Primary_Ch_7	0x070
Alternate_Ch_6	0x260	Primary_Ch_6	0x060
Alternate_Ch_5	0x250	Primary_Ch_5	0x050
Alternate_Ch_4	0x240	Primary_Ch_4	0x040
Alternate_Ch_3	0x230	Primary_Ch_3	0x030
Alternate_Ch_2	0x220	Primary_Ch_2	0x020
Alternate_Ch_1	0x210	Primary_Ch_1	0x010
Alternate_Ch_0	0x200	Primary_Ch_0	0x000

Unused	
Control	0x00C
Destination End Pointer	0x008
Source End Pointer	0x004
	0x000

Рисунок 95 – Карта памяти для 32-х каналов, включая альтернативную структуру управляющих данных

Пример 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 433 перечисляет разряды адреса, которые используются контроллером при доступе к различным элементам структуры управляющих данных, в зависимости от количества каналов, используемых в контроллере.

Таблица 433 – 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 96 демонстрирует пример реализации контроллера с использованием трех каналов DMA и альтернативной структурой управляющих данных.



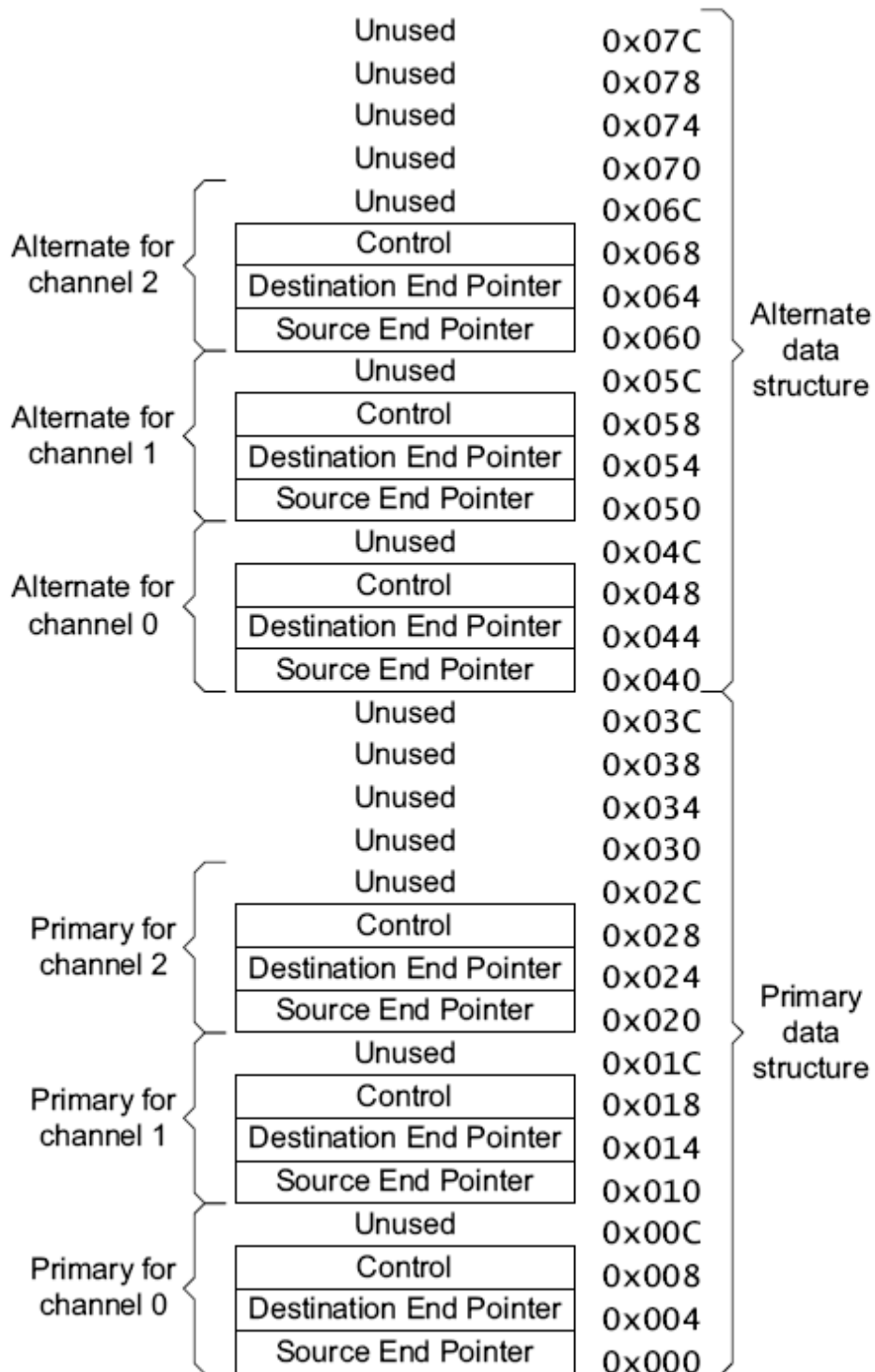


Рисунок 96 – Карта памяти для трех каналов DMA, включая альтернативную структуру управляющих данных (где Destination end pointer – указатель конца данных приемника; Source end pointer – указатель конца данных источника; Control – управление)

Пример структуры управляющих данных 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 434 перечисляет все разрешенные значения базового адреса для первичной структуры управляющих данных, в зависимости от количества каналов DMA, использованных в контроллере.

Таблица 434 – Разрешенные базовые адреса

Кол-во каналов DMA	Разрешенные значения базового адреса для первичной структуры управляющих данных
1	0хXXXXXXXX00, 0хXXXXXXXX20, 0хXXXXXXXX40, 0хXXXXXXXX60, 0хXXXXXXXX80, 0хXXXXXXXXA0, 0хXXXXXXXXC0, 0хXXXXXXXXE0
2	0хXXXXXXXX00, 0хXXXXXXXX40, 0хXXXXXXXX80, 0хXXXXXXXXC0
3-4	0хXXXXXXXX00, 0хXXXXXXXX80
5-8	0хXXXXXXXX000, 0хXXXXXXXX100, 0хXXXXXXXX200, 0хXXXXXXXX300, 0хXXXXXXXX400, 0хXXXXXXXX500, 0хXXXXXXXX600, 0хXXXXXXXX700, 0хXXXXXXXX800, 0хXXXXXXXX900, 0хXXXXXXXXA00, 0хXXXXXXXXB00, 0хXXXXXXXXC00, 0хXXXXXXXXD00, 0хXXXXXXXXE00, 0хXXXXXXXXF00
9-16	0хXXXXXXXX000, 0хXXXXXXXX200, 0хXXXXXXXX400, 0хXXXXXXXX600, 0хXXXXXXXX800, 0хXXXXXXXXA00, 0хXXXXXXXXC00, 0хXXXXXXXXE00
17-32	0хXXXXXXXX000, 0хXXXXXXXX400, 0хXXXXXXXX800, 0хXXXXXXXXC00

Контроллер использует системную память для доступа к двум указателям адреса конца данных и разрядам управления каждого канала. Следующие подразделы описывают эти 32-разрядные области памяти и процедуру вычисления контроллером адреса передачи DMA:

- указатель конца данных источника;
- указатель конца данных приемника;
- разряды управления;
- вычисление адреса.

### 25.5.1 Указатель конца данных источника

Область памяти под названием `src_data_end_ptr` содержит указатель на последний адрес месторасположения данных источника. Таблица 435 перечисляет значения разрядов этой области.

Таблица 435 – Значения разрядов `src_data_end_ptr`

Разряд	Имя	Описание
31...0	<code>src_data_end_ptr</code>	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.

### 25.5.2 Указатель конца данных приемника

Область памяти под названием `dst_data_end_ptr` содержит указатель на последний адрес месторасположения данных приемника. Таблица 436 перечисляет значения разрядов этой области.

Таблица 436 – 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.

### 25.5.3 Разряды управления

Область памяти под названием channel\_cfg обеспечивает управление каждой передачей DMA. Таблица 437 показывает название разрядов этой области.

Таблица 437 – Название разрядов области памяти channel\_cfg

Номер	31...30	29...28	27...26	25...24	23...21
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	10	10	10	10	-
	dst_inc	dst_size	src_inc	src_size	dst_prot_ctrl

Номер	20...18	17...14	13...4	3	2...0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	-	0010	-	0	100
	Src_prot_ctrl	R_power	n_minus_1	next_useburst	cycle_ctrl

Таблица 438 – Назначение разрядов channel\_cfg

Разряд	Имя	Описание
31, 30	dst_inc	Шаг инкремента адреса приемника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: 2'b00 = байт; 2'b01 = полуслово (16-разрядное слово); 2'b10 = слово (32-разрядное 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

Разряд	Имя	Описание
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 (16-разрядное слово); 2'b10 = слово (32-разрядное 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	Задаёт размерность данных источника: 2'b00 = байт; 2'b01 = полуслово (16-разрядное слово); 2'b10 = слово (32-разрядное слово); 2'b11 = зарезервировано
23...21	dst_prot_ctrl	Задаёт 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	Задаёт 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;

Разряд	Имя	Описание
		<p>1 = HPROT[3] в состоянии 1 и доступ кэшируется.</p> <p>Разряд [19] управляет разрядом HPROT[2]:</p> <p>0 = HPROT[2] в состоянии 0 и доступ не буферизуется;</p> <p>1 = HPROT[2] в состоянии 1 и доступ буферизуется.</p> <p>Разряд [18] управляет разрядом HPROT[1]:</p> <p>0 = HPROT[1] в состоянии 0 и доступ не привилегированный;</p> <p>1 = HPROT[1] в состоянии 1 и доступ привилегированный</p>
17...14	R_power	<p>Задаёт количество передач DMA до выполнения контроллером процедуры арбитража.</p> <p>Возможные значения:</p> <p>4'b0000 – арбитраж производится после каждой передачи DMA;</p> <p>4'b0001 – арбитраж производится после 2 передач DMA;</p> <p>4'b0010 – арбитраж производится после 4 передач DMA;</p> <p>4'b0011 – арбитраж производится после 8 передач DMA;</p> <p>4'b0100 – арбитраж производится после 16 передач DMA;</p> <p>4'b0101 – арбитраж производится после 32 передач DMA;</p> <p>4'b0110 – арбитраж производится после 64 передач DMA;</p> <p>4'b0111 – арбитраж производится после 128 передач DMA;</p> <p>4'b1000 – арбитраж производится после 256 передач DMA;</p> <p>4'b1001 – арбитраж производится после 512 передач DMA;</p> <p>4'b1010 – 4'b1111 – арбитраж производится после 1024 передач DMA. Это означает, что арбитраж не производится, так как максимальное количество передач DMA равно 1024</p>
13...4	n_minus_1	<p>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.</p> <p>10-разрядное число плюс 1 задаёт количество передач DMA. Возможные значения:</p> <p>10'b0000000000 = 1 передача DMA;</p> <p>10'b0000000001 = 2 передачи DMA;</p> <p>10'b0000000010 = 3 передачи DMA;</p> <p>10'b0000000011 = 4 передачи DMA;</p> <p>10'b0000000100 = 5 передач DMA;</p> <p>10'b0000000101 = 6 передач DMA;</p> <p>....</p> <p>10'b1111111111 = 1024 передачи DMA.</p> <p>Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач DMA до завершения цикла DMA</p>
3	next_useburst	<p>Контролирует, не установлен ли chnl_useburst_set[C] в состояние 1, если контроллер работает в режиме работы с</p>

Разряд	Имя	Описание
		<p>периферией «Исполнение с изменением конфигурации», и, если контроллер завершает цикл DMA, используя альтернативные управляющие данные.</p> <p>Примечание – Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер устанавливает <code>chnl_useburst_set[C]</code> в значение 0, если количество оставшихся передач DMA меньше, чем <math>2^R</math>. Установка <code>next_useburst</code> разряда определяет, будет ли контроллер дополнительно переопределять разряд <code>chnl_useburst_set[C]</code>.</p> <p>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 <code>next_useburst</code>:</p> <p>0 – контроллер не изменяет значение <code>chnl_useburst_set[C]</code>. Если <code>chnl_useburst_set[C]</code> установлен в 0, то для всех оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер отвечает на запросы по <code>dma_req[]</code> и <code>dma_sreq[]</code>, при выполнении циклов DMA он использует альтернативные управляющие данные.</p> <p>1 – контроллер изменяет значение <code>chnl_useburst_set[C]</code> в состояние 1. Поэтому для оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер реагирует только на запросы по <code>dma_req[]</code>, при выполнении циклов DMA он использует альтернативные управляющие данные</p>
2...0	cycle_ctrl	<p>Режим работы при выполнении цикла DMA:</p> <p>3'b000 Стоп. Означает, что структура управляющих данных является «неправильной»;</p> <p>3'b001 Основной. Контроллер должен получить новый запрос для окончания цикла DMA, перед этим он должен выполнить процедуру арбитража;</p> <p>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;</p> <p>3'b011 Пинг-понг. Контроллер выполняет цикл DMA используя одну из структур управляющих данных.</p> <p>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.</p>

Разряд	Имя	Описание
		<p>Контроллер продолжает выполнять циклы DMA, до тех пор, пока он не прочитает «неправильную» структуру данных или пока процессор не изменит <code>cycle_ctrl</code> поле в состоянии 3'b001 или 3'b010;</p> <p>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;</p> <p>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;</p> <p>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;</p> <p>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</p>

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.

#### 25.5.4 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* =  $src\_data\_end\_ptr - (n\_minus\_1 \ll 1)$
- *адрес приемника* =  $dst\_data\_end\_ptr - (n\_minus\_1 \ll 1)$ .

`src_inc=b01` and `dst_inc=b10`

- *адрес источника* =  $src\_data\_end\_ptr - (n\_minus\_1 \ll 2)$
- *адрес приемника* =  $dst\_data\_end\_ptr - (n\_minus\_1 \ll 2)$ .

`src_inc=b11` and `dst_inc=b11`

- *адрес источника* =  $src\_data\_end\_ptr$
- *адрес приемника* =  $dst\_data\_end\_ptr$ .

Таблица 439 перечисляет адреса приемника цикла DMA для шести слов.

Таблица 439 – Цикла DMA для шести слов с пословным инкрементом

Начальные значения <code>channel_cfg</code> перед циклом DMA				
<code>src_size=2'b10, dst_inc=2'b10, n_minus_1=3'b101, cycle_ctrl=1</code>				
	Указатель конца данных	Счетчик	Отличие*	Адрес
Передачи DMA	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 <code>channel_cfg</code> после цикла DMA				
<code>src_size=2'b10, dst_inc=2'b10, n_minus_1=0, cycle_ctrl=0</code>				

\* Это значение, полученное после сдвига влево значения счетчика на количество разрядов соответствующее `dst_inc`.

Таблица 440 перечисляет адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.



Таблица 440 – 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				
Передачи DMA	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 <sup>R</sup> передач DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=3'b011, cycle_ctrl=1, R_power=2'b11				
Передачи DMA	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 контроллер DMA делает channel\_cfg «неправильным», сбрасывая в 0 поле cycle\_ctrl.

## 25.6 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:

- запрещается обращаться к зарезервированным или неиспользуемым адресам, поскольку это может привести к непредсказуемому поведению контроллера;
- 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 441 – 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
<p>* Значение по сбросу зависит от количества каналов DMA, использованных в контроллере, а также от наличия интегрированной схемы тестирования.</p> <p>** Значение по сбросу зависит от количества каналов DMA, использованных в контроллере</p>				

### 25.6.1 Статусный регистр DMA STATUS

Данный регистр имеет доступ только на чтение. При чтении регистр возвращает состояние контроллера. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

Таблица 442 – Статусный регистр DMA

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

Таблица 443 – Назначение разрядов регистра dma\_status

Номер 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
0	master_enable	Состояние контроллера: 0 = работа контроллера запрещена; 1 = работа контроллера разрешена

### 25.6.2 Регистр конфигурации DMA CFG

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера.

Таблица 444 – Регистр конфигурации DMA

Номер	31...8	7...5	4...1	0
Доступ	U	WO	U	WO
Сброс	0	0	0	0
	-	chnl_prot_ctrl	-	master_enable

Таблица 445 – Назначение разрядов регистра dma\_cfg

Номер 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 АНВ-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

### 25.6.3 Регистр базового адреса управляющих данных каналов CTRL\_BASE\_PTR

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 446 – Регистр базового адреса управляющих данных каналов

Номер	31...10	9...0
Доступ	R/W	U
Сброс	0	0
	ctrl_base_ptr	-

Таблица 447 – Назначение разрядов регистра ctrl\_base\_ptr

Номер 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

#### 25.6.4 Регистр базового адреса альтернативных управляющих данных каналов ALT\_CTRL\_BASE\_PTR

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 448 – Регистр базового адреса альтернативных управляющих данных каналов

Номер	31... 0
Доступ	RO
Сброс	0
	alt_ctrl_base_ptr

Таблица 449 – Назначение разрядов регистра alt\_ctrl\_base\_ptr

Номер 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

### 25.6.5 Регистр статуса ожидания запроса на обработку каналов WAITONREQ\_STATUS

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 450 – Регистр статуса ожидания запроса на обработку каналов

Номер	31	.....	2	1	0
Доступ	RO	.....	RO	RO	RO
Сброс	0	.....	0	0	0
	dma_waitonreg_status for dma_waitnreg [31]	.....	dma_waitonreg_status for dma_waitnreg [2]	dma_waitonreg_status for dma_waitnreg [1]	dma_waitonreg_status for dma_waitnreg [0]

Таблица 451 – Назначение разрядов регистра dma\_waitonreq\_status

Номер 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] = 1 означает, что dma_waitonreq[C] в состоянии 1; Разряд [C] = 0 означает, что dma_waitonreq[C] в состоянии 0

### 25.6.6 Регистр программного запроса на обработку каналов CHNL\_SW\_REQUEST

Данный регистр имеет доступ только на запись. Регистр позволяет устанавливать программно запрос на выполнение цикла DMA.

Таблица 452 – Регистр программного запроса на обработку каналов

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]

Таблица 453 – Назначение разрядов регистра chnl\_sw\_request

Номер 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;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		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

### 25.6.7 Регистр установки пакетного обмена каналов CHNL\_USEBURST\_SET

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.

Таблица 454 – Регистр установки пакетного обмена каналов

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]

Таблица 455 – Назначение разрядов регистра chnl\_useburst\_set

Номер 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 <sup>R</sup> передач. Разряд [C] = 1 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные только от dma_req[]. Контроллер выполняет 2 <sup>R</sup> 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 <sup>R</sup> 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.

### 25.6.8 Регистр сброса пакетного обмена каналов CHNL\_USEBURST\_CLR

Данный регистр имеет доступ только на запись. Регистр разрешает выполнение одиночных запросов по установке `dma_sreq[]`.

Таблица 456 – Регистр сброса пакетного обмена каналов

Номер	31	...	2	1	0
Доступ	WO	...	WO	WO	WO
Сброс	0	...	0	0	0
	<code>chnl_useburst_clr for channel [31]</code>	...	<code>chnl_useburst_clr for channel [2]</code>	<code>chnl_useburst_clr for channel [1]</code>	<code>chnl_useburst_clr for channel [0]</code>

Таблица 457 – Назначение разрядов регистра `chnl_useburst_clr`

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	<code>chnl_useburst_clr</code>	Установка соответствующего разряда разрешает обработку запросов на выполнение циклов DMA от <code>dma_sreq[]</code> . При записи: Разряд <code>[C] = 0</code> не дает эффекта. Необходимо использовать <code>chnl_useburst_set</code> регистр для отключения обработки запросов от <code>dma_sreq[]</code> ; Разряд <code>[C] = 1</code> разрешает обрабатывать запросы на выполнение циклов DMA, полученные от <code>dma_sreq[]</code> . Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

### 25.6.9 Регистр маскирования запросов на обслуживание каналов CHNL\_REQ\_MASK\_SET

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.



Таблица 458 – Регистр маскирования запросов на обслуживание каналов

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_req [31] and dma_sreq [31]	.....	chnl_reg_mask_set for dma_req [2] and dma_sreq [2]	chnl_reg_mask_set for dma_req [1] and dma_sreq [1]	chnl_reg_mask_set for dma_req [0] and dma_sreq [0]

Таблица 459 – Назначение разрядов регистра chnl\_req\_mask\_set

Номер 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

### 25.6.10 Регистр очистки маскирования запросов на обслуживание каналов CHNL\_REQ\_MASK\_CLR

Данный регистр имеет доступ только на запись. Регистр разрешает установку запросов на выполнение циклов DMA на dma\_sreq[] и dma\_req[].

Таблица 460 – Регистр очистки маскирования запросов на обслуживание каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_reg_mask_clr for dma_req [31] and dma_sreq [31]	.....	chnl_reg_mask_clr for dma_req [2] and dma_sreq [2]	chnl_reg_mask_clr for dma_req [1] and dma_sreq [1]	chnl_reg_mask_clr for dma_req [0] and dma_sreq [0]

Таблица 461 – Назначение разрядов регистра chnl\_req\_mask\_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_req_mask_clr	<p>Установка соответствующего разряда разрешает установку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов.</p> <p>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[].</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта</p>

### 25.6.11 Регистр установки разрешения каналов CHNL\_ENABLE\_SET

Данный регистр имеет доступ на чтение и запись. Регистр разрешает работу каналов DMA. Регистр возвращает при чтении состояние разрешений работы каналов DMA.

Таблица 462 – Регистр установки разрешения каналов

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]

Таблица 463 – Назначение разрядов регистра chnl\_enable\_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_enable_set	<p>Разрешает работу каналов DMA и возвращает при чтении состоянии этих настроек.</p> <p>При чтении:                      Разряд [C] = 0 означает, что канал DMA C отключен;                      Разряд [C] = 1 означает, что работа канала DMA C разрешена.</p> <p>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.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта</p>

### 25.6.12 Регистр сброса разрешения каналов CHNL\_ENABLE\_CLR

Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов DMA.

Таблица 464 – Регистр сброса разрешения каналов

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

Таблица 465 – Назначение разрядов регистра chnl\_enable\_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_enable_clr	<p>Установка соответствующего разряда запрещает работу соответствующего канала DMA.</p> <p>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 С.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p> <p>Примечание – Контроллер может отключить канал DMA, установив соответствующий разряд в следующих случаях:</p> <ul style="list-style-type: none"> <li>– при завершении цикла DMA;</li> <li>– при чтении из channel_cfg с полем cycle_ctrl установленным в 3'b000;</li> <li>– при появлении ошибки на шине AHB-Lite</li> </ul>

### 25.6.13 Регистр установки первичной/альтернативной структуры управляющих данных каналов CHNL\_PRI\_ALT\_SET

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).

Таблица 466 – Регистр установки первичной/альтернативной структуры управляющих данных каналов

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]

Таблица 467 – Назначение разрядов регистра chnl\_pri\_alt\_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_pri_alt_set	<p>Установка соответствующего разряда подключает использование альтернативных управляющих данных для соответствующего канала DMA, чтение возвращает состояние этих настроек.</p> <p>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.</p> <p>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.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p> <p>Примечание – Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях:                      – при завершении четырех 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»</p>

**25.6.14 Регистр сброса первичной/альтернативной структуры управляющих данных каналов CHNL\_PRI\_ALT\_CLR**

Данный регистр имеет доступ только на запись. Регистр разрешает работу канала DMA с использованием первичной структуры управляющих данных.

Таблица 468 – Регистр сброса первичной/альтернативной структуры управляющих данных каналов

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]

Таблица 469 – Назначение разрядов регистра chnl\_pri\_alt\_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_pri_alt_clr	<p>Установка соответствующего разряда подключает использование первичных управляющих данных для соответствующего канала DMA.</p> <p>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.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p> <p>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»</p>

### 25.6.15 Регистр установки приоритета каналов CHNL\_PRIORITY\_SET

Данный регистр имеет доступ на запись и чтение. Регистр позволяет присвоить высокий приоритет каналу DMA. Чтение регистра возвращает состояние приоритета каналов DMA.

Таблица 470 – Регистр установки приоритета каналов

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]

Таблица 471 – Назначение разрядов регистра chnl\_priority\_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_priority_set	<p>Установка высокого приоритета каналу DMA, чтение возвращает состояние приоритета каналов DMA.</p> <p>При чтении:</p> <p>Разряд [C] = 0 означает, что каналу DMA C присвоен уровень приоритета по умолчанию;</p> <p>Разряд [C] = 1 означает, что каналу DMA C присвоен высокий уровень приоритета.</p> <p>При записи:</p> <p>Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_clr регистр для установки каналу C уровня приоритета по умолчанию;</p> <p>Разряд [C] = 1 устанавливает каналу DMA C высокий уровень приоритета.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта</p>

### 25.6.16 Регистр сброса приоритета каналов CHNL\_PRIORITY\_CLR

Данный регистр имеет доступ только на запись. Регистр позволяет присвоить каналу DMA уровень приоритета по умолчанию.

Таблица 472 – Регистр сброса приоритета каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_prioirit_clr for channel [31]	.....	chnl_priority_clr for channel [2]	chnl_priority_clr for channel [1]	chnl_priority_clr for channel [0]

Таблица 473 – Назначение разрядов регистра chnl\_priority\_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_priority_clr	<p>Установка разряда присваивает соответствующему каналу DMA уровень приоритета по умолчанию.</p> <p>При записи:</p> <p>Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_set регистр для установки каналу C высокого уровня приоритета.</p> <p>Разряд [C] = 1 устанавливает каналу DMA C уровень приоритета по умолчанию.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта</p>

### 25.6.17 Регистр сброса флага ошибки ERR\_CLR

Данный регистр имеет доступ на запись и чтение. Регистр позволяет сбрасывать сигнал dma\_err в 0. Чтение регистра возвращает состояние сигнала dma\_err.

Таблица 474 – Регистр сброса флага ошибки

Номер	31...1	0
Доступ	U	R/W
Сброс	0	0
	-	err_clr

Таблица 475 – Назначение разрядов регистра err\_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...1	-	Не определено. Следует записывать 0
0	err_clr	<p>Установка сигнала в состояние 0, чтение возвращает состояние сигнала (флага) dma_err.</p> <p>При чтении:</p> <p>Разряд [C] = 0 означает, что dma_err находится в состоянии 0;</p> <p>Разряд [C] = 1 означает, что dma_err находится в состоянии 1.</p> <p>При записи:</p> <p>Разряд [C] =0 не дает эффекта. Состояние dma_err останется неизменным;</p> <p>Разряд [C] =1 сбрасывает сигнал (флаг) dma_err в состояние 0.</p> <p>Примечание – При сбросе сигнала dma_err одновременно с появлением ошибки на шине АНВ-Lite, приоритет отдается ошибке, и, следовательно, значение регистра (и dma_err) останется неизменным (несброшенным)</p>

## 26 Прерывания

Согласно спецификации привилегированной системы команд, RISC-V ядро должно обеспечивать поддержку обработки программных прерываний (machine software interrupt) и прерываний таймера (machine timer interrupt). За генерацию запросов (сигналов) этих прерываний отвечает локальный контроллер прерываний (Core Local Interrupt Controller, CLINT). CLINT содержит регистры для управления программными прерываниями и прерываниями таймера. Карта регистров CLINT приведена в таблице 476.

Таблица 476 – Регистры CLINT

Адрес	Название	Доступ
0x0200_0000	Регистр запроса/снятия программного прерывания	R/W
0x0200_4000	Регистр сравнения таймера	R/W
0x0200_BFF8	Регистр значения таймера	R/W

Таблица 477 – Регистр запроса/снятия прерывания

Биты	Название	Доступ	Описание	Значение
0	msip	R/W	Ожидание программного прерывания	0

Таблица 478 – Регистр базового адреса вектора прерывания mtvec

Биты	Название	Доступ	Описание	Значение
31...2	BASE	R/W	Базовый адрес таблицы векторов прерываний. Должен быть предварительно инициализирован и выровнен по границе 4 байта	0
1, 0	MODE	R/W	0 – все прерывания устанавливают PC в значение BASE; 1, 2, 3 – зарезервированные комбинации бит MODE	0

Таблица 479 – Регистр номера события mcause

Биты	Название	Доступ	Описание	Значение
31	Interrupt	R/W	1 – в поле Exception Code номер события прерывания; 0 – в поле Exception Code номер исключительной ситуации	0
30...0	Exception Code	R/W	Номер события прерывания или исключительной ситуации, где исключительная ситуация: 0 – инструкция по невыровненному адресу; 1 – ошибка доступа инструкции; 2 – нелегальная инструкция; 3 – точка останова; 4 – загрузка по невыровненному адресу; 5 – ошибка доступа при загрузке; 6 – сохранение по невыровненному адресу; 7 – ошибка доступа при сохранении;	0



Биты	Название	Доступ	Описание	Значение
			8 – environment вызывается из U-режима; 9 – environment вызывается из S-режима; 10 – зарезервировано; 11 – environment вызывается из M-режима; 12 – ошибка страницы инструкции; 13 – ошибка загрузки страницы; 14 – зарезервировано; 15 – ошибка сохранения страницы	

Таблица 480 – Регистр значения таймера

Биты	Название	Доступ	Описание	Значение
31...0	mtime_lo	R/W	Младшая часть значения таймера	0
64...32	mtime_hi	R/W	Старшая часть значения таймера	0

Регистр содержит 64-битное значение таймера. Значение увеличивается по сигналу timer\_pulse (в ревизии 1: передний и задний фронт частоты LSE генератора 32 кГц; с ревизии 2: передний фронт частоты HCLK, деленной на значение, установленное в регистре DIV\_SYS\_TIM) внешнего интерфейса процессорного комплекса.

Таблица 481 – Регистр сравнения таймера

Биты	Название	Доступ	Описание	Значение
31...0	mtimesmp_lo	R/W	Младшая часть значения регистра сравнения	0
64...32	mtimesmp_hi	R/W	Старшая часть значения регистра сравнения	0

Регистр содержит 64-битное значение сравнения с таймером. Запрос на прерывание таймера установлен, если значение таймера строго больше значения сравнения.

Периферийные блоки формируют прерывания с int\_global0 по int\_global29.

Таблица 482 – Прерывания, формируемые периферийными блоками

Прерывания	Блок	Принцип формирования
int_global0	DMA	Прерывания от DMA DMA_ERR или DMA_DONE. Обработка прерываний от DMA в соответствии с подпунктом 25.4.5.7 «Индикация ошибок»
int_global1	UART1	Сигнал UARTINTR
int_global2	UART2	Сигнал UARTINTR
int_global3	SSP1	Сигнал SSPINTR
int_global4	POWER	Сигнал прерывания от POWER Detector
int_global5	WWDG	Сигнал прерывания от WWDG
int_global6	TIMER1	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
int_global7	TIMER2	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE

Прерывания	Блок	Принцип формирования
int_global8	ADC	Сигналы прерываний от АЦП EOCIF_1 или AWOIF_1 или EOCIF_2 или AWOIF_2
int_global9	I2C	Сигнал I2C_INT
int_global10	BACKUP	Прерывание от ВКР и часов реального времени
int_global11	Внешнее прерывание 1	Сигнал EXT_INT1 Выходы PA[10] и PD[15] в основном режиме
int_global12	Внешнее прерывание 2	Сигнал EXT_INT2 Выходы PB[6] в альтернативном режиме и PC[4] в основном режиме
int_global13	Внешнее прерывание 3	Сигнал EXT_INT3 Выходы PB[7] в альтернативном режиме и PC[5] в основном режиме
int_global14	ADCUI	Прерывание от АЦП для измерения напряжений и токов
int_global15	ADCUI	Прерывание от АЦП для измерения напряжений и токов
int_global16	ADCUI	Прерывание от АЦП для измерения напряжений и токов
int_global17	L_BLOCK	Прерывание блока L-преобразования
int_global18	-	-
int_global19	SENSORS	Прерывание блока обработки датчиков безопасности
int_global20	CLK_MEASURE	Прерывание блока датчика частоты
int_global21	RANDOM	Прерывание блока генератора случайных чисел
int_global22	ISO7816	Прерывание блока UART ISO7816
int_global23	UART3	Сигнал UARTINTR
int_global24	SSP2	Сигнал SSPINTR
int_global25	SSP3	Сигнал SSPINTR
int_global26	TIMER3	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
int_global27	TIMER4	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
int_global28	UART4	Сигнал UARTINTR
int_global29	otp_corrupt	Сигнал прерывания от блока проверки контрольной суммы (CRC) OTP
<p>Примечание – Выводы в функции EXT_INT1-EXT_INT3 являются входами прерываний по уровню. То есть пока на вывод EXT_INTx подан сигнал высокого уровня, генерируется прерывание. Если на входе сигнал низкого уровня, то прерывание не генерируется. Подробнее см. подраздел 27.1 «Обработка внешних прерываний контроллером PLIC»</p>		

## 27 Контроллер обработки внешних прерываний PLIC

В данном разделе приведено описание контроллера обработки внешних прерываний PLIC (Platform Level Interrupt Controller).

Структурная схема контроллера PLIC представлена на рисунке 97.

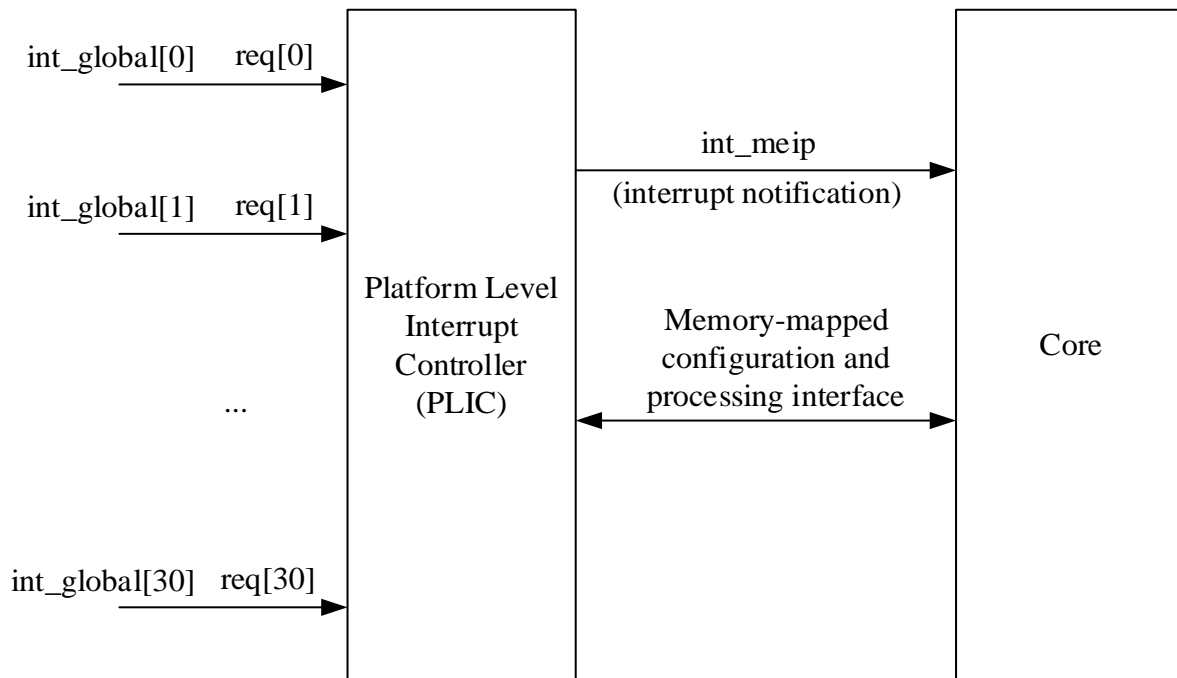


Рисунок 97 – Структурная схема контроллера обработки внешних прерываний

Контроллер PLIC поддерживает 31 внешнее прерывание. Сигналы внешних источников прерываний `int_global[i]` подключаются ко входам `req[i]` контроллера PLIC. Внешние источники прерываний могут использовать как запросы прерываний, формируемые по уровню (level-sensitive), так и импульсные запросы прерываний (edge-triggered).

Прерывания в контроллере PLIC нумеруются с 1-ого по 31-ое. Сигнал `int_global[0]` соответствует 1-му прерыванию, `int_global[30]` соответствует 31-му прерыванию. Нулевой номер прерывания зарезервирован и имеет семантику «нет прерывания».

### 27.1 Обработка внешних прерываний контроллером PLIC

Контроллер PLIC получает входные запросы прерываний и формирует запрос к ядру, сигнализирующий о наличии хотя бы одного прерывания, ожидающего обработки в ядре (сигнал `int_meip`).

Сигнал `int_meip` подключается в регистр CSR `mip`, бит 11 MEIP (machine external interrupt pending). Данный сигнал используются для передачи сообщения в ядро о наличии хотя бы одного прерывания в контроллере PLIC. Подтверждение факта «взятия» прерывания на исполнение (interrupt claiming) и оповещение о завершении выполнения прерывания (interrupt completion), а также включение/выключение посылки прерываний от отдельных источников выполняется через обращение к регистрам, отображенным в память (memory-mapped registers, MMR), командами чтения/записи памяти.

Детальная структурная схема PLIC представлена на рисунке 98. Внешние запросы прерываний передаются в контроллер PLIC по входам req[i]. Каждый запрос попадает в блок проверки допустимости отправки прерывания, где проверяется, является ли данное прерывание ожидающим обработки (бит IP (interrupt pending) в регистре IPM), разрешено ли данное прерывание для обработки (бит IE (interrupt enable) регистра IEM) и больше ли нуля приоритет прерывания. Бит наличия прерывания (IP) устанавливается в «1» в регистре IPM, если прерывание в данный момент не обрабатывается, и на соответствующем входе req[i] обнаружен высокий уровень. Данный бит удерживается в «1» до получения оповещения о «взятии» прерывания на обработку. Бит разрешения прерывания (IE) и значение приоритета прерывания устанавливаются программно. Если прерывание ожидает обработки (IP=1), разрешено (IE=1) и его приоритет больше нуля, то оно становится кандидатом на отправку для обработки. При наличии хотя бы одного кандидата, приоритет которого больше порогового значения в регистре THR, контроллер PLIC устанавливает в «1» сигнал int\_meip, сообщая тем самым ядру о наличии прерывания, ожидающего обработки. Среди всех кандидатов в каждый момент времени выбирается одно прерывание с наибольшим приоритетом, номер данного прерывания возвращается ядру в момент чтения регистра ICC (interrupt claiming/completion). После «взятия» прерывания на обработку бит наличия прерывания (IP) в регистре IPM сбрасывается, обнаружение высокого уровня на соответствующем входе req[i] во время обработки прерывания не выполняется. Повторная установка бита наличия прерывания при обнаружении высокого уровня на входе req[i] возможна только после получения оповещения о завершении обработки прерывания.

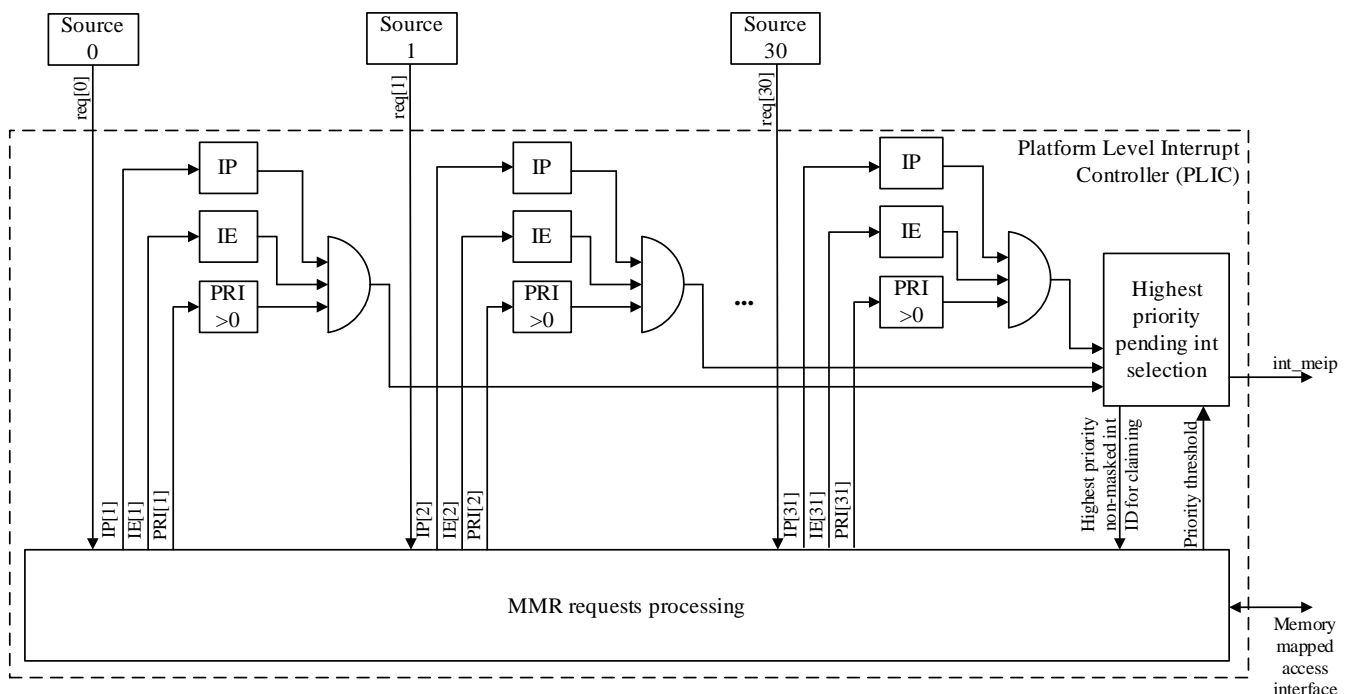


Рисунок 98 – Структурная схема PLIC

## 27.2 Программная модель обработки внешних прерываний

Временная диаграмма обработки внешнего прерывания представлена на рисунке 99. Обработка внешнего прерывания состоит из следующих шагов:

- внешний источник прерывания (interrupt source) выполняет запрос прерывания к контроллеру PLIC (устанавливает высокий уровень на соответствующем входе req[i]);
- контроллер PLIC фиксирует наличие ожидающего прерывания в регистре IPM, после выполнения необходимых аппаратных проверок о допустимости выполнения прерывания и выбора наиболее приоритетного прерывания (при готовности нескольких прерываний) контроллер PLIC формирует сигнал int\_meir к ядру, который указывает на наличие прерывания, готового к обработке;
- если прерывания не запрещены (в регистре CSR mie установлен бит MEIE и в регистре CSR mstatus установлен бит MIE), управление передается в обработчик прерывания (выполняется переход на адрес общего обработчика прерываний), а в регистр CSR mcause записывается «причина», то есть тип источника, вызвавшего прерывание. Для внешних прерываний тип устанавливается в значение «Machine external interrupt»;
- обработчик прерывания осуществляет «взятие» внешнего прерывания на исполнение (claiming), выполняя чтение регистра ICC, который возвращает номер наиболее приоритетного прерывания, ожидающего обработки. Эта транзакция сигнализирует контроллеру PLIC о том, что аппаратный поток принял прерывание к исполнению. Контроллер PLIC сбрасывает соответствующий бит наличия ожидающего прерывания в регистре IPM и отключает обнаружение запросов (высокий уровень) на соответствующем входе req[i];
- если внешний источник использует импульсный запрос прерывания (например, контроллер DMA), то обработчик прерывания осуществляет «завершение» обработки прерывания путем записи в регистр ICC номера прерывания, ранее взятого на обработку. Эта транзакция сигнализирует контроллеру PLIC о том, что обработка прерывания завершена. Контроллер PLIC разрешает обнаружение запросов (высокий уровень) на соответствующем входе req[i];
- обработчик прерывания выполняет обслуживание прерывания;
- если внешний источник использует запрос прерывания, формируемый по уровню, то обработчик прерывания осуществляет «завершение» обработки прерывания путем записи в регистр ICC номера прерывания, ранее взятого на обработку. Эта транзакция сигнализирует контроллеру PLIC о том, что обработка прерывания завершена. Контроллер PLIC разрешает обнаружение запросов (высокий уровень) на соответствующем входе req[i].

Если внешний источник использует импульсный запрос прерывания, то «завершение» обработки прерывания (запись в регистр ICC) перед вызовом непосредственного обработчика, осуществляющего обслуживание прерывания, выполняется во избежание потери новых запросов прерывания от данного источника, которые могут быть получены после «взятия» и до «завершения» обработки прерывания.

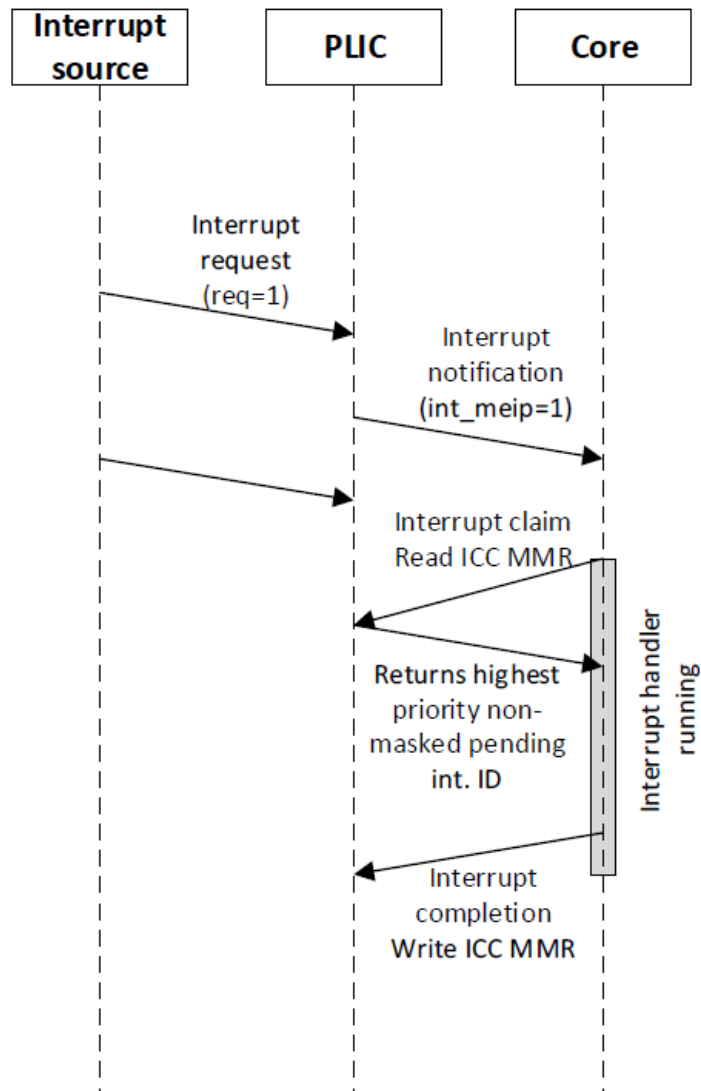


Рисунок 99 – Временная диаграмма обработки прерывания

Помимо использования модели обработки прерываний, описанной выше, для отслеживания наличия прерываний может быть использована модель опроса (poll model), в которой вместо «ожидания» появления прерывания и «заброса» исполнения программы в обработчик прерывания, ПО может самостоятельно проверять наличие запросов на прерывания путем чтения регистра IPM. ПО также может выполнять «взятие» прерывания на обработку путем чтения регистра ICC.

При использовании данной модели программирования надо учитывать, что чтение IPM не гарантирует, что последующее чтение ICC вернет прерывание с тем же номером. Последующее чтение ICC может вернуть прерывание с тем же номером или с другим номером, а также вернуть номер 0, указывающий на отсутствие прерывания.

### 27.3 Регистры контроллера PLIC

Основной интерфейс для взаимодействия с PLIC – это регистры, отображенные в память. Карта регистров PLIC представлена в таблице 483.

Таблица 483 – Регистры PLIC

Адрес	Название	Доступ	Описание
0x0C00_0000	PLIC		Контроллер PLIC
Смещение			
0x0000_0000 + 4 · i	PRIi	R/W	Регистр приоритета i-го прерывания
0x0000_0080 – 0x0000_0FFC	–	–	Зарезервировано
0x0000_1000	IPM	R	Регистр-маска наличия прерываний
0x0000_1004 – 0x0000_1FFC	–	–	Зарезервировано
0x0000_2000	IEM	R/W	Регистр-маска разрешения прерываний
0x0000_2004 – 0x001F_FFFC	–	–	Зарезервировано
0x0020_0000	THR	R/W	Регистр порога приоритета прерываний
0x0020_0004	ICC	R/W	Регистр взятия/завершения обработки прерывания
Примечание – i – номер прерывания от 0 до 31 (регистр приоритета 0-го прерывания (PRI0) зарезервирован)			

### 27.3.1 Регистры PRI1-PRI31

Для каждого (i–1) источника прерываний существует отдельный регистр PRIi, позволяющий задать его приоритет. Прерывания нумеруются с 1-го по 31-ое, что соответствует регистрам PRI1-PRI31. Регистр PRI0 зарезервирован.

Контроллер PLIC поддерживает 7 уровней приоритета прерываний. Приоритеты прерываний нумеруются с 1 (низкий) по 7 (высокий). Значение приоритета 0 означает, что прерывание запрещено. После сброса значения приоритетов не определены, поэтому перед разрешением прерывания, необходимо обязательно сконфигурировать его приоритет.

Если два или более внешних источников прерываний имеют одинаковый приоритет, то для определения очередности обработки используются номера прерываний. Меньшее значение номера прерывания имеет приоритет над большим значением номера прерывания.

Таблица 484 – Регистр приоритета i-ого прерывания (PRIi)

Номер	31...3	2...0
Доступ	U	R/W
Сброс	0	X
		PRI[2:0]

Таблица 485 – Описание бит регистра PRIi

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...3	-	Зарезервировано
2...0	PRI[2:0]	Приоритет прерывания: 0 – прерывание запрещено; 1 – низкий приоритет; ... 7 – высокий приоритет

### 27.3.2 Регистр IPM

Регистр представляет собой битовую маску. Единица в  $i$ -ом бите регистра указывает на наличие прерывания от  $(i-1)$ -го источника; ноль – на отсутствие прерывания. Если прерывание уже обрабатывается, бит соответствующего прерывания в маске сброшен. Запись в регистр запрещена. Попытка записи может вызывать исключение в ядре.

Таблица 486 – Регистр-маска наличия прерываний (IPM)

Номер	31...1	0
Доступ	R	R
Сброс	X	0
	IP[31:1]	IP[0]

Таблица 487 – Описание бит регистра IPM

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	IP[31:0]	Наличие прерывания, ожидающего обработки: 1 – есть прерывание; 0 – нет прерывания. Бит $i$ соответствует $(i-1)$ -му источнику прерывания. 0-ой бит регистра всегда равен нулю. Если прерывание уже обрабатывается, бит соответствующего прерывания в маске равен нулю

### 27.3.3 Регистры IEM

Регистр представляет собой битовую маску. Единица в  $i$ -ом бите указывает на разрешение отправки прерывания от  $(i-1)$ -го источника в аппаратный поток исполнения; ноль – на запрет. После сброса PLIC значения всех битов в регистре IEM сброшены в «0». По готовности принимать прерывания, ПО должно разрешить обработку желаемых прерываний путем записи «1» в соответствующие биты регистра IEM. Рекомендуется устанавливать в «1» только те биты IE, которые соответствуют действительно существующим источникам, подключенным к ядру, чтобы избежать появления прерываний от несуществующих источников.

Таблица 488 – Регистр маски разрешения прерываний (IEM)

Номер	31...1	0
Доступ	R/W	R
Сброс	0	0
	IE[31:1]	IE[0]



Таблица 489 – Описание бит регистра IEM

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	IE[31:0]	Разрешение прерывания: 1 – прерывание разрешено; 0 – прерывание запрещено. Бит <i>i</i> соответствует ( <i>i</i> -1)-му источнику прерывания. 0-ой бит регистра всегда равен нулю

### 27.3.4 Регистр THR

Контроллер прерываний маскирует (запрещает) все прерывания, значения приоритета которых меньше либо равны порогу (значению регистра THR). Например, значение порога 0 разрешает все прерывания с ненулевым приоритетом, а значение порога 7 запрещает все прерывания.

Таблица 490 – Регистр порога приоритета прерываний (THR)

Номер	31...3	2..0
Доступ	U	R/W
Сброс	0	X
		THR[2:0]

Таблица 491 – Описание бит регистра THR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...3	-	Зарезервировано
2...0	THR[2:0]	Порог приоритета прерывания: 0 – разрешены прерывания с приоритетом 1 или более; 1 – разрешены прерывания с приоритетом 2 или более; ... 6 – разрешены прерывания с приоритетом 7; 7 – все прерывания запрещены

### 27.3.5 Регистр ICC

Регистр ICC (interrupt claiming/completion) отвечает за «взятие» прерывания на исполнение (claiming) и сообщение о завершении обработки прерывания (completion). Чтение регистра возвращает номер наиболее приоритетного прерывания, ожидающего обработки в аппаратном потоке исполнения, и переводит прерывание в статус обрабатываемого. При отсутствии прерываний, ожидающих обработки, чтение возвращает 0. Запись сигнализирует о завершении обработки прерывания, взятого ранее в обработку. Записываемое значение должно совпадать со считанным значением при взятии на обработку. Попытка записи другого значения оставит прерывание в статусе обрабатываемого.

Таблица 492 – Регистр взятия/завершения обработки прерывания (ICSS)

Номер	31..0
Доступ	R/W
Сброс	X
	ICSS[31:0]

Таблица 493 – Описание бит регистра ICSS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	ICSS[31:0]	Чтение (claim) возвращает номер наиболее приоритетного прерывания, ожидающего обработки, и переводит данное прерывание в состояние обрабатываемого. При отсутствии прерываний, ожидающих обработки, возвращает 0. Запись (complete) номера прерывания, ранее взятого на обработку, сигнализирует о завершении обработки данного прерывания

## 28 Подсистема отладки

Подсистема отладки ядра соответствует спецификации RISC-V External Debug Support Version 0.13. Поддерживаются следующие опции:

- чтение/запись регистров общего назначения с помощью абстрактных команд;
- Read/Write программный буфер 64 байта;
- четыре регистра абстрактных данных;
- память, доступная только по чтению (Debug ROM).

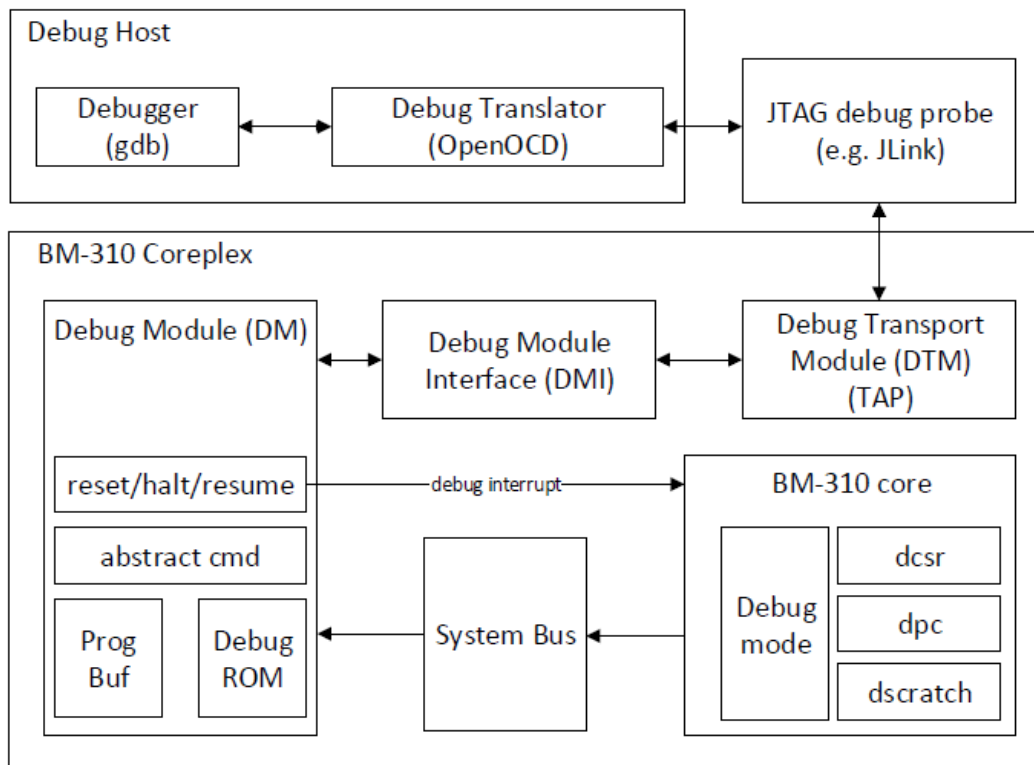


Рисунок 100 – Архитектура системы отладки

Подсистема отладки состоит из блока DTM (Debug Transport Module) и блока DM (Debug Module). DTM связан с DM через блок DMI (Debug Module Interface).

### 28.1 Блок DTM (Debug Transport Module)

DTM содержит TAP (Test Access Port), соответствующий спецификации интерфейса JTAG. Размер IR – 5 бит. Доступ к DM через JTAG осуществляется через регистр DMI. Список доступных по JTAG регистров приведен в таблице ниже. Общая схема цепочки отладки процессорного ядра представлена на рисунке 101. Переход из области синхросигнала JTAG в область синхросигнала ядра происходит в блоке DMI.

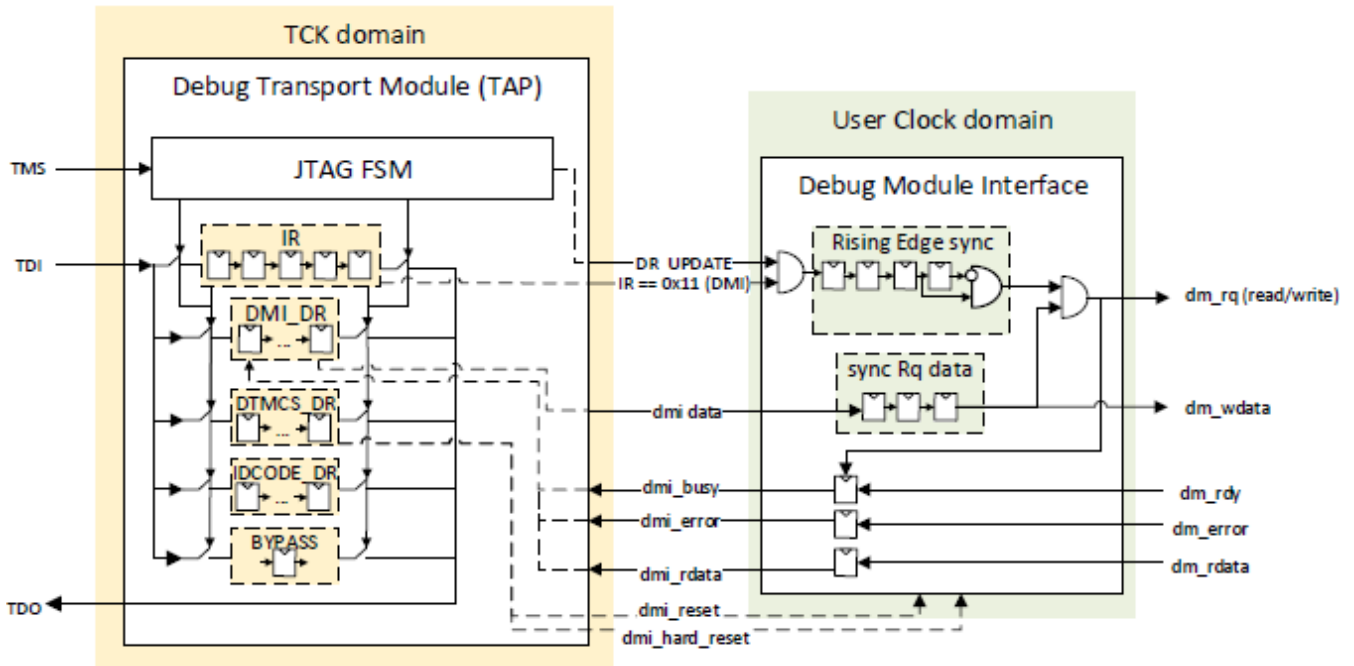


Рисунок 101 – Интерфейс DM

### 28.1.1 Регистры DTM

Таблица 494 – Регистры DTM

Адрес	Название	Доступ
0x00	BYPASS	R
0x01	Регистр идентификатора производителя микросхемы (IDCODE)	R
0x10	Регистр статуса и управления DTM (DTMCS)	R/W
0x11	Регистр доступа к DMI	R/W
0x12-0x1A	BYPASS	R/W

Таблица 495 – Регистр статуса и управления DTM (DTMCS)

Биты	Название	Доступ	Описание	Значение
17	dmihardreset	W1	Запись 1 сбрасывает DMI	
16	dmireset	W1	Запись 1 сбрасывает ошибку в DMI	
14...12	idle	R	Количество циклов, которое отладчик должен провести в состоянии IDLE, между запросом и чтением результата из DMI, для того, чтобы избежать состояния busy	5
11, 10	dmistat	R	Состояние DMI: 0 – ошибок нет; 1 – см.2; 2 – ошибка выполнения операции; 3 – операция еще не выполнена (состояние busy)	
9...4	abits	R	Размер поля адреса регистра DMI	7
3...0	version	R	Версия реализованной спецификации 0 – 0,11; 1 – 0,13	1

Таблица 496 – Регистр доступа к DMI

Биты	Название	Доступ	Описание	Значение
abits+33...34	address	R/W	Адрес запроса на DMI	
33...2	data	R/W	Данные для записи, отправляемые в DM во время состояния Update-DR, либо данные, считанные из DM	
1...0	op	R/W	Код операции. При записи: 0 – Пустой запрос. В DM ничего не посылается; 1 – Чтение по адресу; 2 – Запись по адресу; При чтении: 0 – Предыдущий запрос завершен успешно; 2 – Предыдущий запрос вернул ошибку; 3 – Предыдущий запрос еще не выполнен. Чтобы сбросить это состояние необходим сброс DMI	

Взаимодействие отладчика с DM происходит через DMI путем чтения и записи регистров DM. Для этого используется регистр DMI в DTM. Значение поля abits равно 7. Список доступных по DMI регистров приведен в таблице 497.

Таблица 497 – Регистры отладчика DM

Адрес	Название	Доступ
0x04-0x07	Abstract Data 0-3	R/W
0x10	DM control	R/W
0x11	DM status	R/W
0x12	Hart info	R/W
0x16	Abstract Control and Status	R/W
0x17	Abstract Command	R/W
0x18	Abstract Command Autoexec	R/W
0x20-0x2F	Program Buffer 0-15	R/W

## 28.2 Блок DM (Debug Module)

Взаимодействие процессорного ядра с DM происходит с помощью чтения/записи отображенных в памяти регистров (DM MMRs (Memory Mapped Registers)). Карта памяти DM представлена в таблице 498.

Таблица 498 – Карта памяти DM

Адрес	Название	Доступ
0x0000_0000	Safe zero	R/W
0x0000_0100	Halted ack	W
0x0000_0104	Going ack	W
0x0000_0108	Resume ack	W
0x0000_010C	Exception	W

Адрес	Название	Доступ
0x0000_0300	Where to	R
0x0000_0338	Abstract command 0	R
0x0000_033C	Abstract command 1	R
0x0000_0340-0x37C	Program buf	R/W
0x0000_0380-0x38C	Abstract data	R/W
0x0000_0400	CPU flags	R
0x0000_0800-0xFFF	Debug ROM	R

Таблица 499 – CPU flags

Биты	Название	Доступ	Описание	Значение
0	Going	R	Запрос на выполнение абстрактной команды. Находясь в режиме отладки, ядро считывает данное поле и делает переход по адресу первой абстрактной команды (0x338)	
1	Resumereq	R	Запрос на выход из режима отладки. Находясь в режиме отладки, ядро считывает данное поле и выходит из режима отладки, если его значение равно 1. Исполнение программы продолжается со значения PC, сохраненного в регистре dpc	

### 28.2.1 Регистры DM

Для взаимодействия с внешним отладчиком внутри процессорного ядра реализован специальный режим исполнения: режим отладки. В таблице 500 представлен список контрольных регистров режима отладки. Процессор имеет право читать и писать контрольные регистры отладки только в режиме отладки. В любом другом режиме попытка доступа к данным регистрам вызовет исключение. Исполнение в режиме отладки имеет следующие свойства:

- операции обращения к контрольным регистрам и к памяти происходят также как в режиме привилегированности M;
  - любые прерывания не обрабатываются;
  - исключения не обновляют значения контрольных регистров;
  - все инструкции смены уровня привилегированности не обрабатываются.
- Исключение – инструкция *ebreak*. При исполнении инструкции *ebreak* в режиме отладки процессор выполняет переход на начало Debug ROM, без обновления dpc и dcsr.

Таблица 500 – Контрольные регистры режима отладки

Адрес	Название	Доступ
0x7B0	Регистр управления режимом отладки dcsr	R/W
0x7B1	Регистр счетчика команд dpc	R/W
0x7B2	Дополнительный регистр режима отладки dscratch	R/W

Таблица 501 – Регистр управления режимом отладки dcsr

Биты	Название	Доступ	Описание	Значение
31...28	version	R/W	Версия: 0 – Режим отладки не поддерживается; 4 – Режим отладки поддерживается и соответствует спецификации версии 0.13	4
15	ebreakm	R/W	Значение 1 – при исполнении инструкции ebreak в режиме M процессор переходит в режим отладки	0
12	ebreaku	R/W	Значение 1 – при исполнении инструкции ebreak в режиме U процессор переходит в режим отладки	0
8...6	cause	R	Причина перехода в режим отладки: 1 – Выполнена инструкция ebreak; 3 – Прерывание от DM; 4 – Исполнена очередная инструкция в режиме пошагового исполнения	0
2	step	R/W	Значение 1 – режим пошагового исполнения включен. В данном режиме после выхода из режима отладки процессор исполняет очередную инструкцию пользовательского приложения и снова переключается в режим отладки. При пошаговом исполнении прерывания выключены	0
1, 0	prv	R/W	Режим привилегированности, в котором находилось ядро перед переключением в режим отладки. Отладчик может изменить значение этого поля, чтобы изменить режим процессора после выхода из режима отладки	0

Таблица 502 – Регистр счетчика команд dpc

Биты	Название	Доступ	Описание	Значение
31...0	dpc	R/W	Сохраненное значение счетчика команд. При входе в режим отладки в данный регистр сохраняется текущее значение счетчика команд. Запись в данный регистр возможна только в режиме отладки. При выходе из режима отладки исполнение продолжается с адреса, сохраненного в данном регистре	

Таблица 503 – Дополнительный регистр режима отладки dscratch

Биты	Название	Доступ	Описание	Значение
31...0	dscratch	R/W	Дополнительный регистр, используемый DM для сохранения промежуточной информации	

После входа в режим отладки процессорное ядро выполняет переход по адресу начала Debug ROM (адрес 0x800). Программный код, сохраненный в Debug ROM,

осуществляет опрос флагов DM. При определении запроса на исполнение абстрактной команды (флаг «Going» в регистре 0x400) происходит переход исполнения на адрес первой абстрактной команды (регистр по адресу 0x338). В случае возникновения исключения во время исполнения абстрактной команды или кода из программного буфера ядро переходит по адресу внутри Debug ROM, где сохранен код обработчика исключений. В данном обработчике находится инструкция записи по адресу 0x10c. После чего отладчик может считать значение регистра Abstract Control and Status и определить факт возникновения исключения в процессе выполнения абстрактной команды. Если при опросе считывается запрос на продолжение исполнения пользовательского приложения (флаг «Resumereq»), то исполняется код, осуществляющий переход по адресу, сохраненному в регистре dpc.

Любые операции отладчика: считывание/запись регистров, считывание/запись памяти, установка точек останова в коде, последовательное исполнение – осуществляются с помощью модификации кода в программном буфере и исполнения абстрактных команд.

### 28.3 Подключение JTAG-адаптера к микросхеме

Если отладочный интерфейс не заблокирован, к микросхеме может быть подключен JTAG адаптер, с помощью которого программные средства разработки позволяют работать с микросхемой в отладочном режиме. Линии JTAG должны быть подтянуты к питанию резисторами сопротивлением 4,7 кОм – 10 кОм с учетом, чтобы эти подтяжки не влияли на работу системы (см рисунок 102).

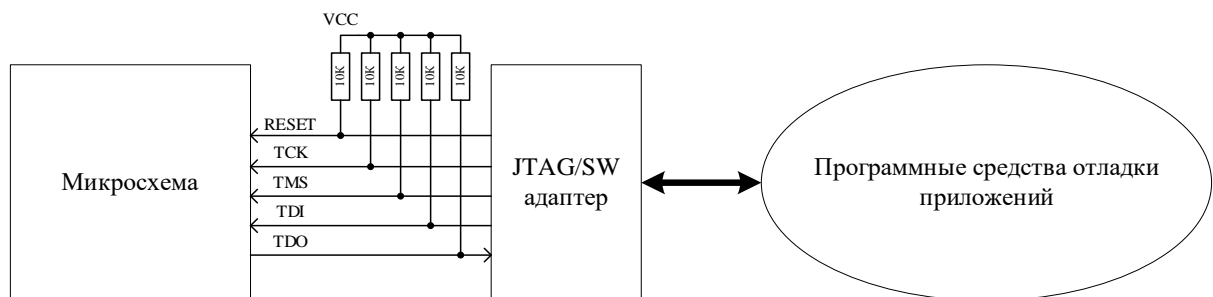


Рисунок 102 – Схема работы в режиме отладки

Отладочный интерфейс JTAG расположен на выводах порта А (см. таблицу 504). В отладочном режиме следует соблюдать рекомендации, приведенные в разделе 18 «Порты ввода-вывода».

Таблица 504 – Переопределение выводов интерфейса JTAG

Вывод JTAG	Вывод микросхемы	Описание
TCK	PA6/TMR1_CH4/TCK	В качестве выводов интерфейса JTAG используются выводы порта А, совмещенные с выводами Таймера 1. Использование совмещенных с JTAG выводов Таймера 1 при разрешенной работе порта JTAG запрещено
TDO	PA7/TMR1_CH4N/TDO	
TMS	PA8/TMR1_ETR/TMS	
TDI	PA9/TMR1_BRK/TDI	
JTAG_EN	JTAG_EN	Необходимо доопределять до логического «0» для работы с отладочным JTAG-интерфейсом



## 29 Сторожевые таймеры

### 29.1 Блок сторожевого таймера IWDG

IWDG – независимый 12-разрядный сторожевой таймер, который считает вниз от основания счета (значения перезагрузки, настраиваемого начального значения) до нуля. При достижении счетчиком нуля сторожевой таймер формирует сигнал сброса, который перезагружает систему. Для предотвращения формирования сброса от IWDG необходимо заблаговременно перезагрузить таймер записью ключевого значения 0хАААА в регистр IWDG\_KR – после чего таймер снова начнет обратный отсчет от установленного значения. Тем самым и обеспечивается контроль над зависанием программы в определенных точках, где такое возможно (не перезагрузив IWDG вовремя, система сбросится). Сброс системы от независимого сторожевого таймера сбрасывает IWDG. Любой другой системный сброс (например, внешний сброс) не влияет на настройки и работу таймера IWDG.

Блок-схема независимого сторожевого таймера приведена на рисунке 103.

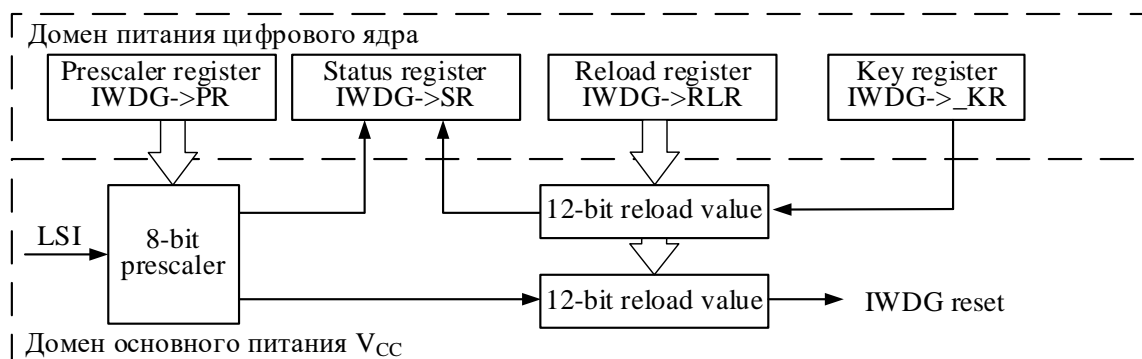


Рисунок 103 – Блок-схема независимого сторожевого таймера

Структурно IWDG состоит из двух частей. Часть с регистрами управления расположена в домене питания цифрового ядра, доступна по APB, тактируется частотой PCLK и сбрасывается по всем типам сброса. Часть непосредственно с независимым сторожевым таймером расположена в домене основного питания микросхемы V<sub>CC</sub> и тактируется от генератора LSI – поэтому для фактической записи в регистры IWDG в домене основного питания V<sub>CC</sub> генератор LSI должен находиться в рабочем режиме. Часть сторожевого таймера в домене основного питания V<sub>CC</sub> сбрасывается только от сброса, формируемого IWDG – после сброса от IWDG сторожевой таймер останавливает свою работу.

Период счета сторожевого таймера от разрешения работы до формирования сброса является функцией от основания счета (значения перезагрузки) и поделенной на значение предделителя частоты LSI.

Обновление значений предделителя IWDG в домене основного питания осуществляется записью регистра IWDG\_PR соответственно. Обновление основания счета (значение, от которого таймер при старте начинает обратный отсчет) IWDG в домене основного питания осуществляется значением регистра IWDG\_RLR записью ключевого значения 0хАААА в регистр ключа. Для возможности записи в IWDG\_PR и IWDG\_RLR предварительно в регистр ключа должно быть записано ключевое значение 0х5555 для разрешения доступа по записи в них. При записи в регистр IWDG\_PR

значения предделителя и ключа 0xAAAA в регистр IWDG\_KR формируются запросы на обновление соответствующих параметров (предделителя и основания счета), которые передаются IWDG в домене основного питания. Запросы на обновления не захватываются (не сохраняются) частью в домене основного питания, то есть при любом типе сброса запросы на обновление от IWDG в домене питания ядра сбрасываются. Наличие запроса обновления от IWDG в домене питания ядра и/или информация о процессе фактического обновления на стороне IWDG в домене основного питания транслируются флагами PVU (для предделителя) и RVU (для основания счета). Фактическое обновление значений (со сбросом соответствующих флагов после обновления) осуществляется только при разрешенной работе IWDG и наличии частоты LSI. При фактическом обновлении предделителя и основания счета недоступна запись в регистры IWDG\_PR и IWDG\_RLR соответственно. После фактического обновления снимаются запросы обновления со стороны IWDG в домене питания ядра V<sub>CC</sub>.

Пример работы с блоком IWDG:

- 1 Включение генератора LSI и ожидание его выхода в рабочий режим.
- 2 Разрешение доступа к регистрам IWDG\_PR и IWDG\_RLR записью значения 0x5555 в регистр IWDG\_KR.
- 3 Запись значения предделителя в регистр IWDG\_PR.
- 4 Разрешение работы IWDG записью значения 0xCCCC в регистр IWDG\_KR.
- 5 Ожидание обновления значения предделителя (сброса флага PVU в 0).
- 6 Разрешение доступа к регистрам IWDG\_PR и IWDG\_RLR записью значения 0x5555 в регистр IWDG\_KR.
- 7 Проверка флага RVU = 0 перед записью в IWDG\_RLR.
- 8 Запись значения перезагрузки в регистр IWDG\_RLR.
- 9 Периодическая перезагрузка сторожевого таймера в процессе выполнения программы значением регистра IWDG\_RLR с помощью записи ключевого значения 0xAAAA в регистр IWDG\_KR.

### 29.1.1 Регистры блока сторожевых таймеров

Таблица 505 – Описание регистров блока сторожевого таймера IWDG

Базовый Адрес	Название	Описание
0x4005_0000	IWDG	Сторожевой таймер IWDG
Смещение		
0x00	IWDG_KR[15:0]	Регистр ключа
0x04	IWDG_PR[2:0]	Делитель частоты сторожевого таймера
0x08	IWDG_RLR[11:0]	Регистр основания счета сторожевого таймера
0x0C	IWDG_SR[1:0]	Регистр статуса сторожевого таймера

**29.1.1.1 IWDG\_KR**

Таблица 506 – Регистр IWDG\_KR

Номер	31...16	15...0
Доступ	U	W
Сброс	0	0
	-	KEY[15:0]

Таблица 507 – Описание бит регистра IWDG\_KR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...0	KEY[15:0]	<p>Значение ключа (только запись, читается 0x0000).</p> <p>При записи:</p> <ul style="list-style-type: none"> <li>– 0xAAAA – перезагрузка значения таймера значением регистра IWDG_RLR. Значение должно периодически программно записываться при разрешенной работе IWDG, в противном случае сторожевой таймер генерирует сброс, если таймер достиг значения нуля;</li> <li>– 0x5555 – разрешение доступа по записи к регистрам IWDG_PR и IWDG_RLR. Если после разрешения доступа в регистр ключа записывается другое ключевое значение, доступ к регистрам IWDG_PR и IWDG_RLR запрещается;</li> <li>– 0xCCCC – разрешение работы сторожевого таймера (если работа сторожевого таймера не разрешена; работа таймера останавливается аппаратно сбросом от IWDG)</li> </ul>

**29.1.1.2 IWDG\_PR**

Таблица 508 – Регистр IWDG\_PR

Номер	31...3	2...0
Доступ	U	R/W
Сброс		0
	-	PR[2:0]

Таблица 509 – Описание бит регистра IWDG\_PR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...3	-	Зарезервировано
2...0	PR[2:0]	<p>Делитель частоты сторожевого таймера:</p> <ul style="list-style-type: none"> <li>000 – делитель на 4: LSI/4;</li> <li>001 – делитель на 8: LSI/8;</li> <li>010 – делитель на 16: LSI/16;</li> <li>011 – делитель на 32: LSI/32;</li> <li>100 – делитель на 64: LSI/64;</li> </ul>

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		<p>101 – делитель на 128: LSI/128;                      110 – делитель на 256: LSI/256;                      111 – делитель на 256: LSI/256.</p> <p>Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR.</p> <p>При записи в регистр формируется запрос на обновление значения делителя частоты в часть IWDG в домене основного питания (см. рисунок 103).</p> <p>Запись в регистр IWDG_PR недоступна при фактической записи предделителя в часть IWDG в домене основного питания V<sub>CC</sub> (см. рисунок 103).</p> <p>При чтении возвращает записанное в регистр значение, а не значение предделителя IWDG в домене основного питания</p>

### 29.1.1.3 IWDG\_RLR

Таблица 510 – Регистр IWDG\_RLR

Номер	31...12	11...0
Доступ	U	R/W
Сброс	0	0xFFF
	-	RLR[11:0]

Таблица 511 – Описание бит регистра IWDG\_RLR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...12	-	Зарезервировано
11...0	RLR[11:0]	<p>Значение перезагрузки (основания счета) сторожевого таймера. Сторожевой таймер декрементируется, начиная с этого значения.</p> <p>Определяет значение, загружаемое в сторожевой таймер при записи значения 0xAAAA в регистр IWDG_KR.</p> <p>Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR.</p> <p>Запись в регистр IWDG_RLR недоступна при фактической записи основания счета в часть IWDG в домене основного питания V<sub>CC</sub> (см. рисунок 103).</p> <p>При чтении возвращает записанное в регистр значение, а не значение перезагрузки IWDG в домене основного питания</p>

29.1.1.4 IWDG\_SR

Таблица 512 – Регистр IWDG\_SR

Номер	31...2	1	0
Доступ	U	R	R
Сброс		0	0
	-	RVU	PVU

Таблица 513 – Описание бит регистра IWDG\_SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	-	Зарезервировано
1	RVU	<p>Устанавливается аппаратно и служит признаком того, что обновляется значение сторожевого таймера IWDG в домене основного питания V<sub>CC</sub> из регистра перезагрузки IWDG_RLR (см. рисунок 103).</p> <p>Работа IWDG разрешена и LSI работает: Отображает запрос от регистров управления на обновление значения сторожевого таймера значением регистра IWDG_RLR и/или фактическое обновление значения сторожевого таймера в домене основного питания V<sub>CC</sub>. Сбрасывается при завершении фактического обновления значения перезагрузки. Значение регистра перезагрузки IWDG_RLR может быть обновлено только, если этот бит равен нулю.</p> <p>Работа IWDG не разрешена или LSI не работает: Отображает запрос от регистров управления на обновление значения сторожевого таймера в домене основного питания V<sub>CC</sub> значением регистра IWDG_RLR. Сбрасывается любым типом сброса. Значение регистра перезагрузки IWDG_RLR может быть обновлено независимо от состояния этого бита – после запуска IWDG будет обновлен последним записанным в IWDG_RLR значением</p>

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
0	PVU	<p>Флаг обновления делителя частоты сторожевого таймера. Устанавливается аппаратно и служит признаком того, что обновляется значение делителя частоты IWDG в домене основного питания V<sub>CC</sub> из регистра предделителя IWDG_PR (см. рисунок 103).</p> <p>Работа IWDG разрешена и LSI работает: Отображает запрос от регистров управления на обновление делителя частоты сторожевого таймера значением регистра IWDG_PR и/или фактическое обновление значения делителя в домене основного питания V<sub>CC</sub>. Сбрасывается при завершении фактического обновления делителя. Значение регистра предделителя IWDG_PR может быть обновлено только, если этот бит равен нулю.</p> <p>Работа IWDG не разрешена или LSI не работает: Отображает запрос от регистров управления на обновление значения сторожевого таймера в домене основного питания V<sub>CC</sub> значением регистра IWDG_PR. Сбрасывается любым типом сброса. Значение регистра перезагрузки IWDG_PR может быть обновлено независимо от состояния этого бита – после запуска IWDG будет обновлен последним записанным в IWDG_PR значением</p>

## 29.2 Блок сторожевого таймера WWDG

WWDG в данном контроллере – это оконный сторожевой таймер, счетчик которого считает сверху вниз, начиная со значения, которое мы вносим в регистр WWDG\_CR (биты T[6:0]), но не до нуля, а до 63 (0x3F). Только кроме значения, с которого считает WWDG, и значения 0x63, до которого считает данный сторожевой таймер, существует еще одно промежуточное значение, находящееся между данными величинами. Оно также является задаваемой величиной и, таким образом, образуется окно между данной величиной и значением 0x3F, поэтому данный сторожевой таймер и является оконным. Подобно IWDG, сторожевой таймер WWDG также, досчитав до 0x3F, даст команду на перезагрузку системы, если же мы, конечно, не перезагрузим данный оконный сторожевой таймер своевременно. Только перезагрузить мы должны его не только не позже, чем его счетчик досчитает вниз до 0x3F, но еще и не раньше, чем он досчитает до промежуточной величины, которая также заносится в регистр WWDG\_CFR (биты W[6:0]). Данная величина поэтому должна находиться в интервале от числа 0x3F до заданного значения счетчика, с которого он начинает обратный отсчет. Таким образом, мы получаем возможность проверки определенных временных интервалов хода нашей программы.

Блок-схема оконного сторожевого таймера приведена на рисунке 104.

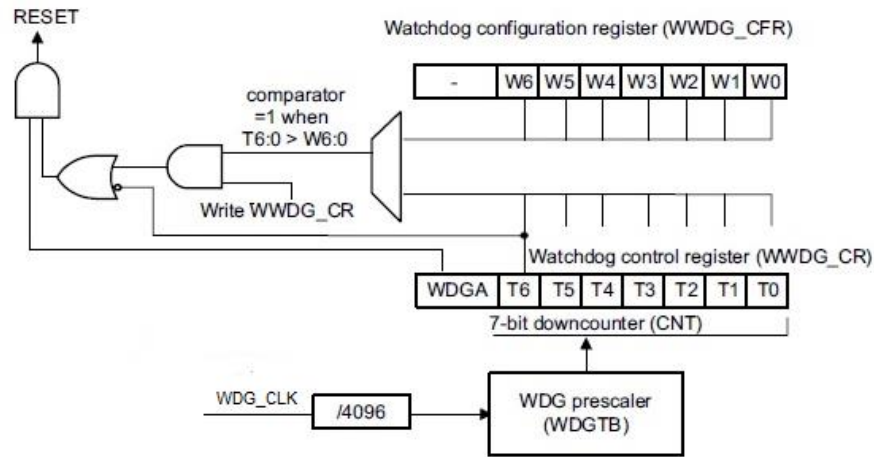


Рисунок 104 – Блок-схема оконного сторожевого таймера

Тактовые сигналы счетчика идут через первый предделитель с постоянным коэффициентом деления 4096 с шины WDG\_CLK на второй, регулируемый предделитель, значение коэффициента деления которого мы можем устанавливать с помощью битов регистра WGTB. После второго предделителя тактовые сигналы попадают на счетчик обратного отсчета.

Число, которое находится в битах T6:T0 регистра WWDG\_CR после старта оконного таймера начинает инкрементироваться, как только оно достигнет значения 0x3F, то есть как только бит T6 очистится, на один из мультиплексов пойдет сигнал о перезагрузке системы. Но на то он и мультиплексор, что сигнал о перезагрузке он может получить не только от очистки бита T6. Аналогичный сигнал он может получить от компаратора, в котором сравниваются значения битов T6:T0 регистра WWDG\_CR со значениями битов W6:W0 регистра WWDG\_CFR. И, если первое число больше второго, то есть если счетчик обратного отсчета еще не досчитал до значения первой границы, а в это время мы пытались перезагрузить оконный сторожевой таймер, то мультиплексор и получит данный сигнал, который также заставит систему перезагрузиться.

Также процесс работы оконного сторожевого таймера показан на графике на рисунке 105.

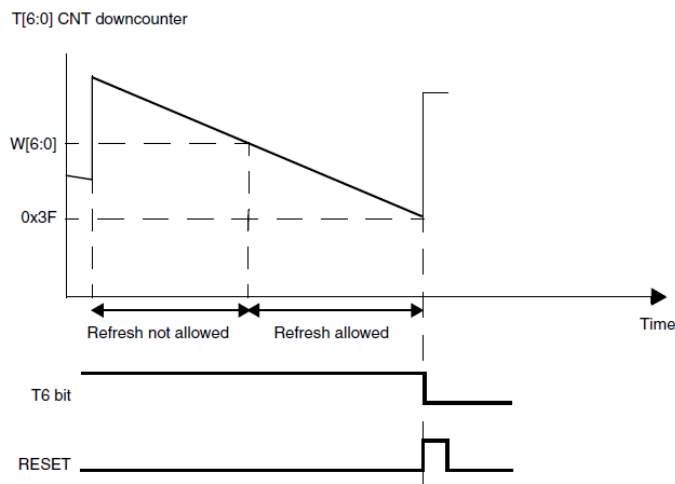


Рисунок 105 – Временная диаграмма оконного сторожевого таймера

Можно рассчитать время работы WWDG от старта до окончания счета и перезагрузки по следующей формуле

$$t_{WWDG} = t_{WDG\_CLK} \cdot 4096 \cdot 2^{WG_{TB}} \cdot (T[5:0] + 1), \text{ мс} \quad (28)$$

где  $t_{WWDG}$  – таймаут;

$t_{WDG\_CLK}$  – период синхросигнала APB шины в мс.

### 29.2.1 Описание регистров блока сторожевого таймера WWDG

Таблица 514 – Описание регистров блока сторожевого таймера WWDG

Базовый Адрес	Название	Описание
0x4004_8000	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

#### 29.2.1.1 WWDG\_CR

Таблица 515 – Регистр WWDG\_CR

Номер	31...8	7	6...0
Доступ	U	R/S	R/W
Сброс		0	1
	-	WDGA	T[6:0]

Таблица 516 – Описание бит регистра WWDG\_CR

Номер бита	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 счетчика (от старших разрядов к младшим). Эти биты содержат значение сторожевого таймера, который декрементируется каждые $4096 \times 2^{WG_{TB}}$ циклов частоты WDG_CLK периферийной шины APB



**29.2.1.2 WWDG\_CFR**

Таблица 517 – Регистр WWDG\_CFR

Номер	31...10	9	8, 7	6...0
Доступ	U	R/S	R/W	R/W
Сброс		0	0	1
	-	EWI	WDGTB[1:0]	W[6:0]

Таблица 518 – Описание бит регистра WWDG\_CFR

Номер бита	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 значения 40h. Прерывание запрещается только аппаратным сбросом
8, 7	WGTB[1:0]	Делитель частоты сторожевого таймера: 00 – частота таймера (WDG_CLK / 4096) /1; 01 – частота таймера (WDG_CLK / 4096) /2; 10 – частота таймера (WDG_CLK / 4096) /4; 11 – частота таймера (WDG_CLK / 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

**29.2.1.3 WWDG\_SR**

Таблица 519 – Регистр WWDG\_SR

Номер	31...2	1	0
Доступ	U	R	R/C
Сброс		1	0
	-	WDG_WEC	EWIF

Таблица 520 – Описание бит регистра WWDG\_SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	-	Зарезервировано
1	WDG_WEC	Флаг окончания записи в регистры WWDG_CR и WWDG_CFR: 1-запись завершена; 0-запись не завершена
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 $EWI = 0$
--	--	--

### 30 Блок прямого и обратного L-преобразования

Блок преобразования  $L / L^{-1}$  соответствует ГОСТ Р34.12-2015.

#### 30.1 Особенности

- Возможность задания направления преобразования  $L / L^{-1}$ ;
- Возможность выбора числа  $R$  преобразований ( $0 - 31$ );
- Возможность задания коэффициентов  $1$  преобразования через  $16$  таблиц соответствия  $a \rightarrow K \cdot a$ ;
- Время выполнения преобразований равно числу преобразований  $R$  плюс  $1$  такт;
- Подключение АНВ;
- Сигнал занятости модуля текущим преобразованием.

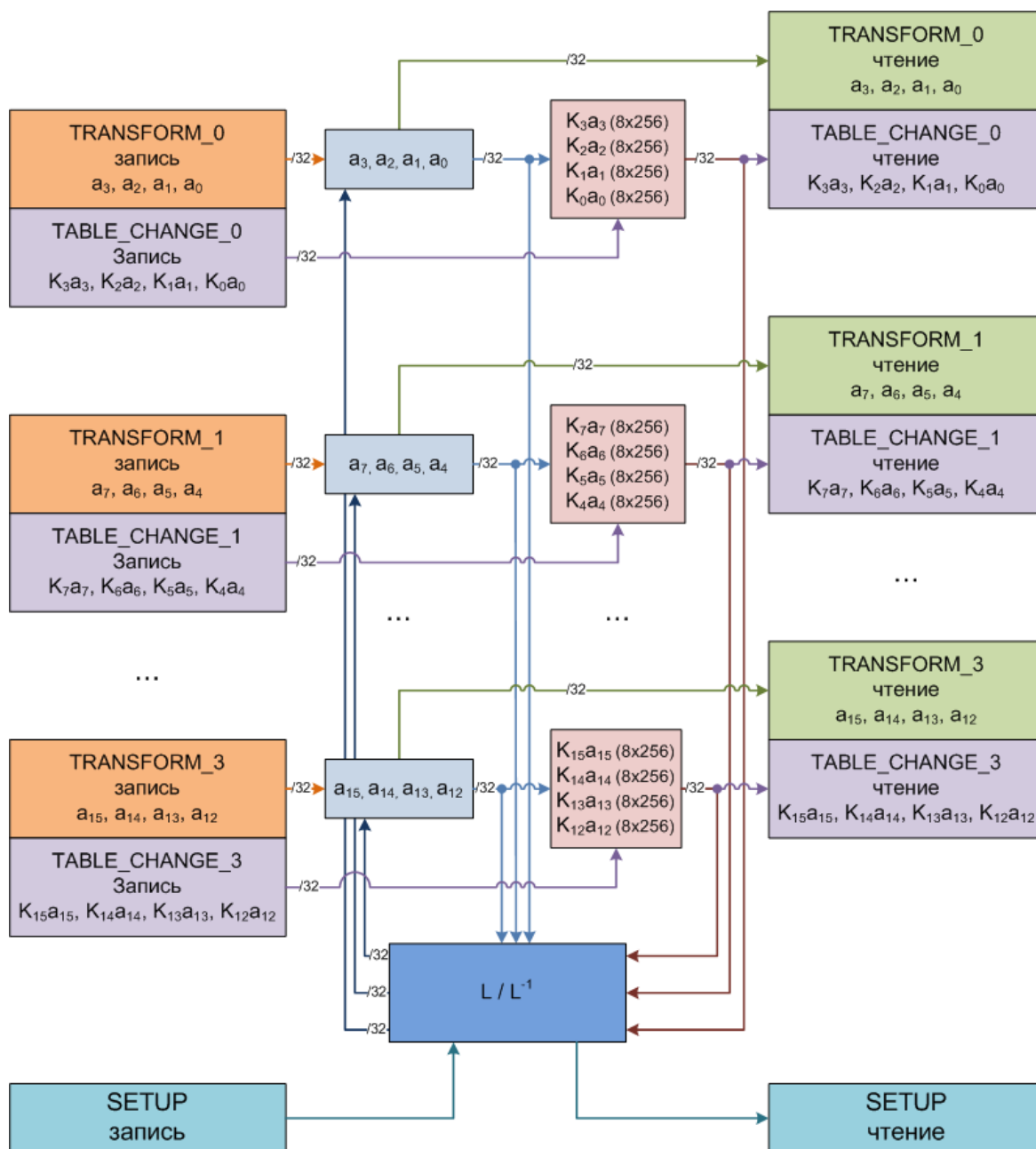


Рисунок 106 – Структурная схема L-блока

### 30.2 Регистры блока

Таблица 521 – Регистры блока прямого и обратного L-преобразования

Базовый Адрес	Название	Описание
0x6000_4000	L_BLOCK	Блок прямого и обратного L-преобразования
Смещение		
0x00	TRANSFORM_0	Регистр преобразования 1
0x04	TRANSFORM_1	Регистр преобразования 2
0x08	TRANSFORM_2	Регистр преобразования 3
0x0C	TRANSFORM_3	Регистр преобразования 4
0x10	TABLE_CHANGE_0	Регистр изменения таблицы 1
0x14	TABLE_CHANGE_1	Регистр изменения таблицы 2
0x18	TABLE_CHANGE_2	Регистр изменения таблицы 3
0x1C	TABLE_CHANGE_3	Регистр изменения таблицы 4
0x20	SETUP	Регистр настройки преобразования

#### 30.2.1 Регистр преобразования n-ого слова данных TRANSFORM\_n

Относительный адрес:  $0x00 + 4 \cdot n \dots 0x0c$ .

Таблица 522 – Регистры TRANSFORM\_n

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31...0	DATA	<p>Запись в данный регистр передает n-ое слово для L / L-1 преобразования (запуск преобразования выполняется при записи в старший байт последнего регистра, результат преобразования доступен через (SET_R_COUNT + 1) тактов (см. пункт 30.2.3 «Регистр настройки преобразования SETUP»)).</p> <p>Запись в данный регистр выбирает ячейки для изменения в n-ой группы таблиц коэффициентов 1 преобразования (см. пункт 30.2.2 «Регистр изменения таблицы TABLE_CHANGE_n»).</p> <p>Чтение из данного регистра возвращает n-ое слово результата L / L-1 преобразования</p>	RW

**30.2.2 Регистр изменения таблицы TABLE\_CHANGE\_n**

Относительный адрес:  $0x10 + 4 \cdot n \dots 0x1C$ .

Таблица 523 – Регистры TABLE\_CHANGE\_n

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
7...0	TABLE_VAL_n0	Запись в данное поле изменяет значение перевода $a \rightarrow K \cdot a$ , 0-го (младшего) байта n-го входного слова (значение a выбирается записью значение в TRANSFORM_n). Изменения необходимо производить в режиме прямого преобразования с нулевым количеством преобразований R (см. пункт 30.2.3 «Регистр настройки преобразования SETUP»). Чтение данного поля возвращает текущее значение перевода $a \rightarrow K \cdot a$ , 0-го (младшего) байта n-го входного слова (значение a выбирается записью значение в TRANSFORM_n)	RW
15...8	TABLE_VAL_n1	Запись в данное поле изменяет значение перевода $a \rightarrow K \cdot a$ , 1-го байта n-го входного слова (значение a выбирается записью значение в TRANSFORM_n). Изменения необходимо производить в режиме прямого преобразования с нулевым количеством преобразований R (см. пункт 30.2.3 «Регистр настройки преобразования SETUP»). Чтение данного поля возвращает текущее значение перевода $a \rightarrow K \cdot a$ , 1-го байта n-го входного слова (значение a выбирается записью значение в TRANSFORM_n)	RW
23...16	TABLE_VAL_n2	Запись в данное поле изменяет значение перевода $a \rightarrow K \cdot a$ , 2-го байта n-го входного слова (значение a выбирается записью значение в TRANSFORM_n). Изменения необходимо производить в режиме прямого преобразования с нулевым количеством преобразований R (см. пункт 30.2.3 «Регистр настройки преобразования SETUP»). Чтение данного поля возвращает текущее значение перевода $a \rightarrow K \cdot a$ , 2-го байта n-го входного слова (значение a выбирается записью значение в TRANSFORM_n)	RW
31...24	TABLE_VAL_n3	Запись в данное поле изменяет значение перевода $a \rightarrow K \cdot a$ , 3-го (старшего) байта n-го входного слова (значение a выбирается записью значение в TRANSFORM_n). Изменения необходимо производить в режиме прямого преобразования с нулевым количеством преобразований R (см. пункт 30.2.3 «Регистр настройки преобразования SETUP»). Чтение данного поля возвращает текущее значение перевода $a \rightarrow K \cdot a$ , 3-го (старшего) байта n-го входного слова (значение a выбирается записью значение в TRANSFORM_n)	RW

### 30.2.3 Регистр настройки преобразования SETUP

Относительный адрес: 0x20.

Таблица 524 – Регистр SETUP

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
4...0	SET_R_COUNT	Настройка количества $R / R^{-1}$ преобразований при реализации $L / L^{-1}$ преобразования. Нулевое значение – выключение преобразования. Число $R / R^{-1}$ преобразований выбирается на все $L / L^{-1}$ преобразование при старте (при записи в старший байт регистра TRANSFORM_3). Изменение данного поля не влияет на результат выполняемого преобразования	RW
7...5	-	Зарезервировано	RO
12...8	CURR_R_COUNT	Число оставшихся $R / R^{-1}$ в текущем, выполняем $L / L^{-1}$ преобразовании. Нулевое значение означает что преобразование не начато или уже выполнено. Запись в данное поле игнорируется	RO
15...13	-	Зарезервировано	RO
16	SET_L_DIR	Направление преобразования. 0 – прямое, L-преобразование; 1 – обратное, $L^{-1}$ -преобразование. Направление выбирается на все преобразование при старте (при записи в старший байт регистра TRANSFORM_3). Изменение данного поля не влияет на результат выполняемого преобразования	RW
17	CURR_L_DIR	Направление текущего, выполняемого преобразования. Запись в данное поле игнорируется	RO
31...18	-	Зарезервировано	RO

### 30.3 Инициализация

- в регистр SETUP задать значение 0x00000000;
- в регистры TRANSFORM\_0 ... TRANSFORM\_3 задать значение 0x00\_00\_00\_00;
- в регистры TABLE\_CHANGE\_0 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>3a</sub>, PP = K<sub>2a</sub>, RR = K<sub>1a</sub>, SS = K<sub>0a</sub>, для a = 0x00;
- в регистры TABLE\_CHANGE\_1 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>7a</sub>, PP = K<sub>6a</sub>, RR = K<sub>5a</sub>, SS = K<sub>4a</sub>, для a = 0x00;
- в регистры TABLE\_CHANGE\_2 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>11a</sub>, PP = K<sub>10a</sub>, RR = K<sub>9a</sub>, SS = K<sub>8a</sub>, для a = 0x00;
- в регистры TABLE\_CHANGE\_3 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>15a</sub>, PP = K<sub>14a</sub>, RR = K<sub>13a</sub>, SS = K<sub>12a</sub>, для a = 0x00;
- в регистры TRANSFORM\_0 ... TRANSFORM\_3 задать значение 0x01\_01\_01\_01;

- в регистры TABLE\_CHANGE\_0 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>3a</sub>, PP = K<sub>2a</sub>, RR = K<sub>1a</sub>, SS = K<sub>0a</sub>, для a = 0x01;
- в регистры TABLE\_CHANGE\_1 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>7a</sub>, PP = K<sub>6a</sub>, RR = K<sub>5a</sub>, SS = K<sub>4a</sub>, для a = 0x01;
- в регистры TABLE\_CHANGE\_2 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>11a</sub>, PP = K<sub>10a</sub>, RR = K<sub>9a</sub>, SS = K<sub>8a</sub>, для a = 0x01;
- в регистры TABLE\_CHANGE\_3 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>15a</sub>, PP = K<sub>14a</sub>, RR = K<sub>13a</sub>, SS = K<sub>12a</sub>, для a = 0x01;
- ...
- в регистры TRANSFORM\_0 ... TRANSFORM\_3 задать значение 0xFF\_FF\_FF\_FF;
- в регистры TABLE\_CHANGE\_0 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>3a</sub>, PP = K<sub>2a</sub>, RR = K<sub>1a</sub>, SS = K<sub>0a</sub>, для a = 0xFF;
- в регистры TABLE\_CHANGE\_1 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>7a</sub>, PP = K<sub>6a</sub>, RR = K<sub>5a</sub>, SS = K<sub>4a</sub>, для a = 0xFF;
- в регистры TABLE\_CHANGE\_2 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>11a</sub>, PP = K<sub>10a</sub>, RR = K<sub>9a</sub>, SS = K<sub>8a</sub>, для a = 0xFF;
- в регистры TABLE\_CHANGE\_3 задать значение 0xTT\_PP\_RR\_SS, где TT = K<sub>15a</sub>, PP = K<sub>14a</sub>, RR = K<sub>13a</sub>, SS = K<sub>12a</sub>, для a = 0xFF;
- в регистр SETUP задать значение необходимого преобразования, направление и число повторений преобразования R / R<sup>-1</sup>.

### 30.4 Выполнение преобразования

В регистры TRANSFORM\_0 ... TRANSFORM\_3 необходимо записать данные для преобразования. При записи в старший байт регистра TRANSFORM\_3 автоматически запускается преобразование. Результат преобразования доступен через количество тактов равное SET\_R\_COUNT + 1, где SET\_R\_COUNT – значение поля регистра SETUP. Также ход преобразование можно отслеживать по счетчику CURR\_R\_COUNT регистра SETUP. При достижении счетчиком значения 0 преобразование закончено. Результат преобразования считывается из регистров TRANSFORM\_0 ... TRANSFORM\_3. Для смены направления преобразования перед записью регистров TRANSFORM\_0...TRANSFORM\_3 необходимо изменить значение регистра SETUP. Изменять значение регистра SETUP можно сразу после старта очередного преобразования, новое значение будет применено к следующему преобразованию.

### 31 Блок замены S-box

Блок замены S-box выполняет ускорение типовой задачи блочной замены. При работе данного блока производится преобразование 8bit → 8bit по таблице из 256-ти элементов. Каждое входное число заменяется на число из ячейки таблицы с индексом соответствующим входному числу.

Блок замены S-box имеет следующие особенности:

- возможность задания таблицы замены 8bit → 8bit;
- обработка 32-битного слова за один такт;
- прием до восьми 32-битных слов с конвейерной обработкой и сохранением результата преобразования;
- АНВ подключение.

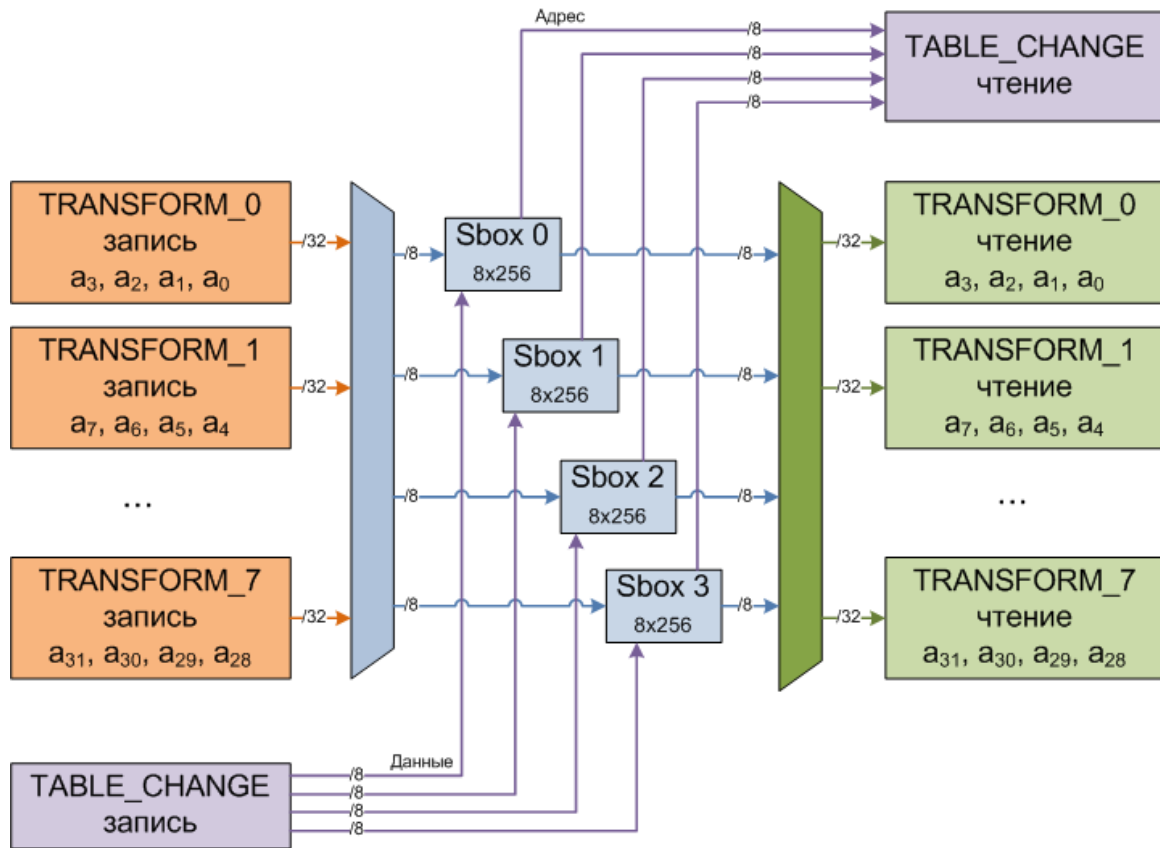


Рисунок 107 – Структурная схема S-блока

#### 31.1 Регистры блока

Таблица 525 – Регистры блока замены S-box

Базовый Адрес	Название	Описание
0x6000_8000	S_BLOCK0	Блок замены S-box0
0x6000_C000	S_BLOCK1	Блок замены S-box1
0x6001_0000	S_BLOCK2	Блок замены S-box2
0x6001_4000	S_BLOCK3	Блок замены S-box3
0x6001_8000	S_BLOCK4	Блок замены S-box4
0x6001_C000	S_BLOCK5	Блок замены S-box5



Базовый Адрес	Название	Описание
0x6002_0000	S_BLOCK6	Блок замены S-box6
0x6002_4000	S_BLOCK7	Блок замены S-box7
Смещение		
0x00	TRANSFORM_0	Регистр преобразования 1
0x04	TRANSFORM_1	Регистр преобразования 2
0x08	TRANSFORM_2	Регистр преобразования 3
0x0C	TRANSFORM_3	Регистр преобразования 4
0x10	TRANSFORM_4	Регистр преобразования 5
0x14	TRANSFORM_5	Регистр преобразования 6
0x18	TRANSFORM_6	Регистр преобразования 7
0x1C	TRANSFORM_7	Регистр преобразования 8
0x20	TABLE_CHANGE	Регистр изменения таблицы

### 31.1.1 Регистр преобразования n-ого слова данных TRANSFORM\_n

Относительный адрес:  $0x00 + 4 \cdot n \dots 0x1C$ .

Таблица 526 – Регистры TRANSFORM\_n

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31:0	DATA	Запись в данный регистр передает n-ое слово для S-преобразования (результат преобразования доступен через один такт). Запись в любой из этих регистров выбирает ячейки таблицы для изменения (см. регистр изменения таблицы). Чтение из данного регистра возвращает результат S-преобразования n-го слова	RW

### 31.1.2 Регистр изменения таблицы TABLE\_CHANGE

Относительный адрес: 0x20.

Таблица 527 – Регистр TABLE\_CHANGE

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
7...0	TABLE_VAL_0	Запись в данное поле изменяет значение преобразования 0 - го (младшего) байта входного слова (адрес записи выбирается записью значение в TRANSFORM_n). Чтение данного поля возвращает адрес изменяемой ячейки таблицы	RW
15...8	TABLE_VAL_1	Запись в данное поле изменяет значение преобразования 1-го байта входного слова (адрес записи выбирается записью значение в TRANSFORM_n). Чтение данного поля возвращает адрес изменяемой ячейки таблицы	RW

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
23...16	TABLE_VAL_2	Запись в данное поле изменяет значение преобразования 2-го байта входного слова (адрес записи выбирается записью значение в TRANSFORM_n). Чтение данного поля возвращает адрес изменяемой ячейки таблицы	RW
31...24	TABLE_VAL_3	Запись в данное поле изменяет значение преобразования 3-го (старшего) байта входного слова (адрес записи выбирается записью значение в TRANSFORM_n). Чтение данного поля возвращает адрес изменяемой ячейки таблицы	RW

### 31.2 Инициализация восьмибитного S-BOX

- В регистр TRANSFORM\_0 задать значение 0x00\_00\_00\_00;
- В регистр TABLE\_CHANGE задать значение 0xТТ\_ТТ\_ТТ\_ТТ, где ТТ значение преобразования, 0x00 → ТТ;
- В регистр TRANSFORM\_0 задать значение 0x01\_01\_01\_01;
- В регистр TABLE\_CHANGE задать значение 0xТТ\_ТТ\_ТТ\_ТТ, где ТТ значение преобразования, 0x01 → ТТ;
- ...
- В регистр TRANSFORM\_0 задать значение 0xFF\_FF\_FF\_FF;
- В регистр TABLE\_CHANGE задать значение 0xТТ\_ТТ\_ТТ\_ТТ, где ТТ значение преобразования, 0xFF → ТТ;

### 31.3 Инициализация четырехбитного S-BOX

- В регистр TRANSFORM\_0 задать значение 0x00\_00\_00\_00;
- В регистр TABLE\_CHANGE задать значение 0xТР\_ТР\_ТР\_ТР, где Т, Р значение преобразования, 0x0 → Т, 0x0 → Р;
- В регистр TRANSFORM\_0 задать значение 0x01\_01\_01\_01;
- В регистр TABLE\_CHANGE задать значение 0xТР\_ТР\_ТР\_ТР, где Т, Р значение преобразования, 0x0 → Т, 0x1 → Р;
- ...
- В регистр TRANSFORM\_0 задать значение 0x10\_10\_10\_10;
- В регистр TABLE\_CHANGE задать значение 0xТР\_ТР\_ТР\_ТР, где Т, Р значение преобразования, 0x1 → Т, 0x0 → Р;
- ...
- В регистр TRANSFORM\_0 задать значение 0xFF\_FF\_FF\_FF;
- В регистр TABLE\_CHANGE задать значение 0xТТ\_ТТ\_ТТ\_ТТ, где Т, Р значение преобразования, 0xF → Т, 0xF → Р.

### **31.3.1 Выполнение преобразования**

В регистры TRANSFORM\_0 ... TRANSFORM\_7 надо записать необходимое число слов для преобразования, затем считать преобразованное значение из регистров TRANSFORM\_0 ... TRANSFORM\_7. В случае преобразования одного слова (преобразование 32 бит) между записью в регистр TRANSFORM\_0 и чтением результата из этого регистра необходима пауза в один такт. В случае преобразования более 32 бит чтение можно выполнять сразу же после записи всех регистров. Для преобразования можно использовать любое подмножество регистров, результаты преобразования будут в регистрах с соответствующими номерами.

## 32 Блок байтовой замены p-byte

Блок байтовой замены p-byte осуществляет перестановку байт входного 64-байтового массива.

Блок байтовой замены p-byte имеет следующие особенности:

- настраиваемая таблица перестановки (каждому выходному байту задается номер используемого входного байта);
- обновление выходного массива данных за такт после изменения входного массива;
- конвейерный прием и обработка до 16 слов размером 32 бита;
- подключение АНВ.

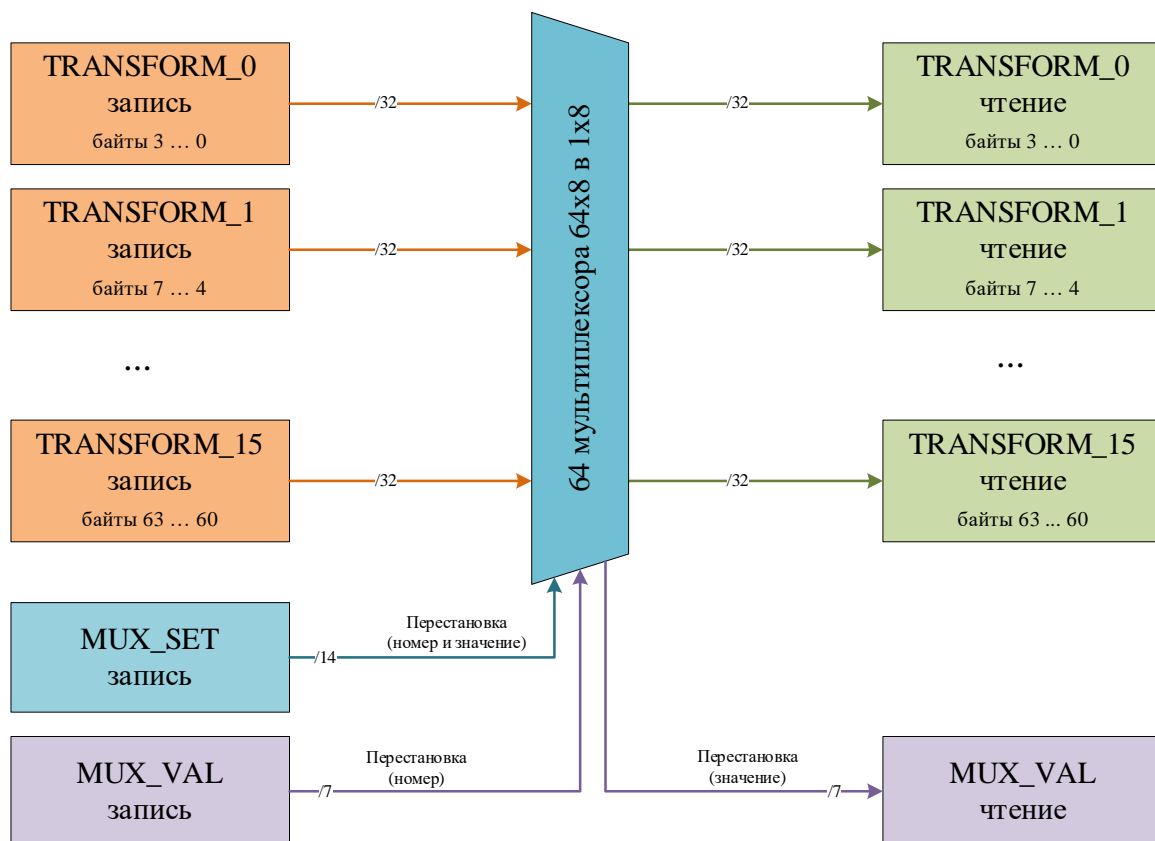


Рисунок 108 – Структурная схема блока p-byte

### 32.1 Регистры модуля

Таблица 528 – Регистры модуля байтовой замены p-byte

Базовый Адрес	Название	Описание
0x6002_8000	P_BYTE	Блок байтовой замены p-byte
Смещение		
0x00	TRANSFORM_0	Регистр задания слова данных для перестановки 1
0x04	TRANSFORM_1	Регистр задания слова данных для перестановки 2
0x08	TRANSFORM_2	Регистр задания слова данных для перестановки 3
0x0C	TRANSFORM_3	Регистр задания слова данных для перестановки 4
0x10	TRANSFORM_4	Регистр задания слова данных для перестановки 5
0x14	TRANSFORM_5	Регистр задания слова данных для перестановки 6

Базовый Адрес	Название	Описание
0x18	TRANSFORM_6	Регистр задания слова данных для перестановки 7
0x1C	TRANSFORM_7	Регистр задания слова данных для перестановки 8
0x20	TRANSFORM_8	Регистр задания слова данных для перестановки 9
0x24	TRANSFORM_9	Регистр задания слова данных для перестановки 10
0x28	TRANSFORM_10	Регистр задания слова данных для перестановки 11
0x2C	TRANSFORM_11	Регистр задания слова данных для перестановки 12
0x30	TRANSFORM_12	Регистр задания слова данных для перестановки 13
0x34	TRANSFORM_13	Регистр задания слова данных для перестановки 14
0x38	TRANSFORM_14	Регистр задания слова данных для перестановки 15
0x3C	TRANSFORM_15	Регистр задания слова данных для перестановки 16
0x40	MUX_SET	Регистр задания перестановки
0x44	MUX_VAL	Регистр чтения перестановки

### 32.1.1 Регистр задания n-ого слова данных для перестановки TRANSFORM\_n

Относительный адрес:  $0x00 + 4 \cdot n \dots 0x3C$ .

Таблица 529 – Регистры TRANSFORM\_n

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31...0	DATA	Запись в данный регистр передает n-ое слово для перестановки. Чтение из данного регистра возвращает n-ое слово результата перестановки	RW

### 32.1.2 Регистр задания перестановки MUX\_SET

Относительный адрес: 0x40.

При записи в данный регистр (младшие 16 бит требуется записывать одновременно) задается перестановка. Выходному байту с номером OUT\_SEL ставится в соответствие входной байт с номером IN\_SEL.

Таблица 530 – Регистр MUX\_SET

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
5...0	IN_SEL	Запись задает перестановку IN_SEL → OUT_SEL Данное поле только для записи, при чтении возвращается 0	WO
7, 6	-	Зарезервировано	RO
13...8	OUT_SEL	Запись задает перестановку IN_SEL → OUT_SEL Данное поле только для записи, при чтении возвращается 0	WO
31...14	-	Зарезервировано	RO

### 32.1.3 Регистр чтения перестановки MUX\_VAL

Относительный адрес: 0x44.

Таблица 531 – Регистр MUX\_VAL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
5...0	VAL	Данное поле только для чтения, запись игнорируется. При чтении из данного поля возвращается значение перестановки для выбранного через SEL выходного байта	RO
7, 6	-	Зарезервировано	RO
13...8	SEL	Запись в данное поле задает выходной байт, для которого буде возвращено значение перестановки при чтении регистра. Данное поле только для записи, при чтении возвращается 0	WO
3...14	-	Зарезервировано	RO

### 32.2 Инициализация

- В регистр MUX\_SET задать значение 0x0000\_00\_ТТ, где ТТ номер входного байта, который необходимо выдавать на 0 байт;
- В регистр MUX\_SET задать значение 0x0000\_01\_ТТ, где ТТ номер входного бита, который необходимо выдавать на 1 байт;
- ...
- В регистр MUX\_SET задать значение 0x0000\_3F\_ТТ, где ТТ номер входного бита, который необходимо выдавать на 63 байт.

### 32.3 Выполнение преобразования

В регистры TRANSFORM\_0 ... TRANSFORM\_15 записать данные для преобразования. Сразу после этого из регистров TRANSFORM\_0 ... TRANSFORM\_15 можно считать результат перестановки. Обновление выходных данных происходит одновременно с записью любой части входных данных.

### 33 Блок битовой замены p-bit

Блок битовой замены p-bit выполняет перестановку по таблице бит входного 128 битного слова.

Блок битовой замены p-bit имеет следующие особенности:

- возможность задания таблицы перестановки бит (каждому выходному биту задается номер используемого входного бита);
- обновление выходного слова в том же такте после изменения входного;
- конвейерный прием и обработка до четырех 32 битных слов;
- АНВ подключение.

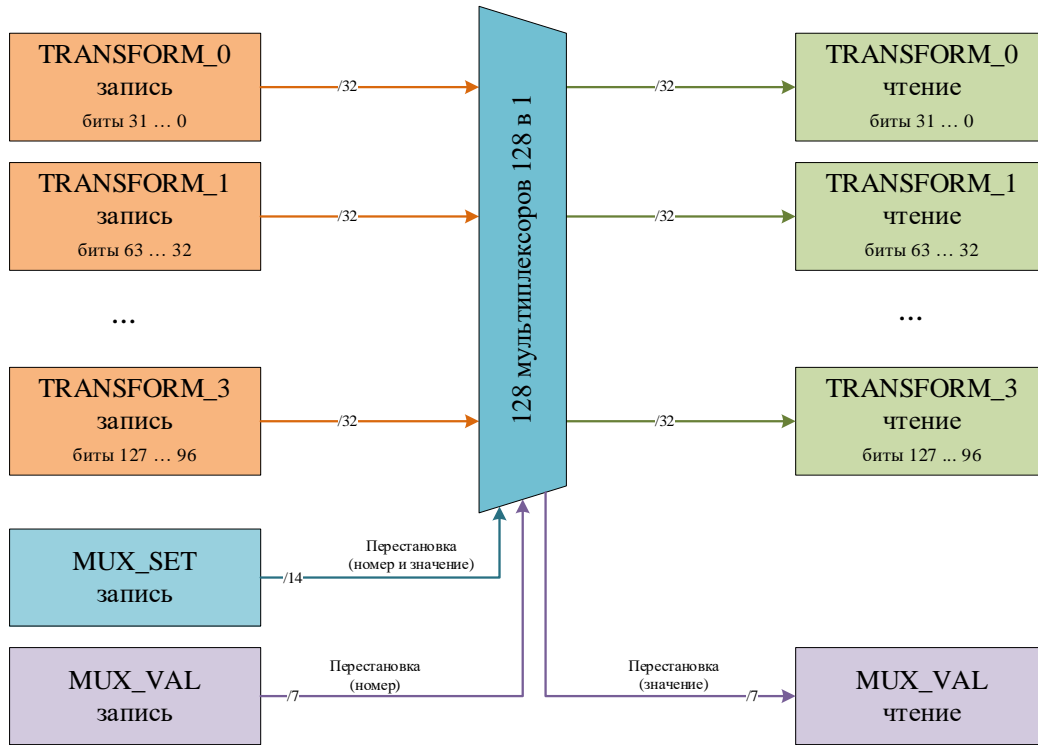


Рисунок 109 – Структурная схема блока p-bit

#### 33.1 Регистры блока

Таблица 532 – Регистры блока

Базовый Адрес	Название	Описание
0x6002_C000	P_BIT0	Блок битовой замены p-bit
0x6003_0000	P_BIT1	Блок битовой замены p-bit
0x6003_4000	P_BIT2	Блок битовой замены p-bit
0x6003_8000	P_BIT3	Блок битовой замены p-bit
Смещение		
0x00	TRANSFORM_0	Регистр задания слова данных для перестановки 1
0x04	TRANSFORM_1	Регистр задания слова данных для перестановки 2
0x08	TRANSFORM_2	Регистр задания слова данных для перестановки 3

Базовый Адрес	Название	Описание
0x0C	TRANSFORM_3	Регистр задания слова данных для перестановки 4
0x10	MUX_SET	Регистр задания перестановки
0x14	MUX_VAL	Регистр чтения перестановки

### 33.1.1 Регистр задания n-ого слова данных для перестановки TRANSFORM\_n

Относительный адрес:  $0x00 + 4 \cdot n \dots 0x0C$ .

Таблица 533 – Регистр TRANSFORM\_n

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31...0	DATA	Запись в данный регистр передает n-ое слово для перестановки. Чтение из данного регистра возвращает n-ое слово результата перестановки	RW

### 33.1.2 Регистр задания перестановки MUX\_SET

Относительный адрес: 0x10.

При записи в данный регистр (младшие 16 бит требуется записывать одновременно) задается перестановка. Выходному биту с номером OUT\_SEL ставится в соответствие входной бит с номером IN\_SEL.

Таблица 534 – Регистр MUX\_SET

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
6...0	IN_SEL	Запись задает перестановку IN_SEL → OUT_SEL. Данное поле только для записи, при чтении возвращается 0	WO
7	-	Зарезервировано	RO
14...8	OUT_SEL	Запись задает перестановку IN_SEL → OUT_SEL. Данное поле только для записи, при чтении возвращается 0	WO
31...15	-	Зарезервировано	RO

### 33.1.3 Регистр чтения перестановки MUX\_VAL

Относительный адрес: 0x14.

Таблица 535 – Регистр MUX\_VAL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
6...0	VAL	Данное поле только для чтения, запись игнорируется. При чтении из данного поля возвращается значение перестановки для выбранного через SEL выходного бита	RO



Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
7	-	Зарезервировано	RO
14...8	SEL	Запись в данное поле задает выходной бит, для которого будет возвращено значение перестановки при чтении регистра. Данное поле только для записи, при чтении возвращается 0	WO
31...15	-	Зарезервировано	RO

### 33.2 Инициализация

- В регистр MUX\_SET задать значение 0x0000\_00\_ТТ, где ТТ номер входного бита, который необходимо выдавать на 0 бит;
- В регистр MUX\_SET задать значение 0x0000\_01\_ТТ, где ТТ номер входного бита, который необходимо выдавать на 1 бит;
- ...
- В регистр MUX\_SET задать значение 0x0000\_7F\_ТТ, где ТТ номер входного бита, который необходимо выдавать на 127 бит.

### 33.3 Выполнение преобразования

В регистры TRANSFORM\_0 ... TRANSFORM\_3 необходимо записать данные для преобразования. Сразу после этого из регистров TRANSFORM\_0 ... TRANSFORM\_3 можно считать результат перестановки. Обновление выходных данных происходит одновременно с записью любой части входных данных.

## 34 Модуль обработки датчиков безопасности (SENSORS)

### 34.1 Общее описание

Блок используется для обработки датчиков обнаружения атак на микросхему. Модуль обрабатывает активную защитную сетку, датчики ЭМИ и датчик света.

Защитная сетка включает в себя блок формирования последовательности данных по ГОСТ 28147-89, работающий в режиме простой замены. В качестве входных данных задается 256-битный ключ (регистры **mesh\_key1-mesh\_key7**) и 64-битные данные инициализации (**mesh\_int0-mesh\_int1**). Эти данные передаются на плавающую сетку с последующим контролем соответствия переданных и полученных по сетке уровней сигналов (нарушение сетки или замыкание ее слоев приводит к расхождению заданного и принятого уровней, что вызывает формирование сигнала тревоги). Защитная сетка находится в верхнем слое микросхемы и покрывает всю микросхему, кроме блока ADCUI.

Защитная сетка после включения основного питания микросхемы и перевода сигнала сброса микросхемы в неактивное состояние начинает работать с нулевым ключом и данными инициализации на частоте генератора HSI. Запись ключа и данных инициализации в регистры **mesh\_key1-mesh\_key7** и **mesh\_init0-mesh\_init1** можно осуществить только один раз после сброса. После каждых 32 раундов ГОСТ 28147-89 блок формирования данных осуществляет повтор рандомизации. Если данные инициализации блока формирования данных по ГОСТ 28147-89 после сброса микросхемы были нулевыми, то после каждых 32 раундов они меняют свое значение и каждые следующие 32 раунда начинаются с новыми данными. Значения регистров **mesh\_key1-mesh\_key7** и **mesh\_init0-mesh\_init1** при этом не изменяются. Предусмотрена возможность настройки интервала переключения защитной сетки в диапазоне от 3 до 256 тактов рабочей частоты HSI. Настройка интервала переключения после сброса осуществляется однократно в регистре **meshcntr**.

Датчики ЭМИ обрабатывают периодически переключаемые к нулю и питанию линии (в противофазе, то есть в каждый момент времени сигналы на линиях должны быть разных уровней) с последующим контролем, что сигналы на линиях отличаются. При наличии на линиях одинаковых уровней формируется сигнал тревоги. Переключение линий осуществляется на частоте HSI.

Датчик света контролирует схему распределенных по микросхеме элементов с фиксированным по сбросу состоянием – при несоответствии состояния этих элементов фиксированному значению формируется сигнал тревоги.

Условная схема блока SENSORS представлена на рисунке 110.

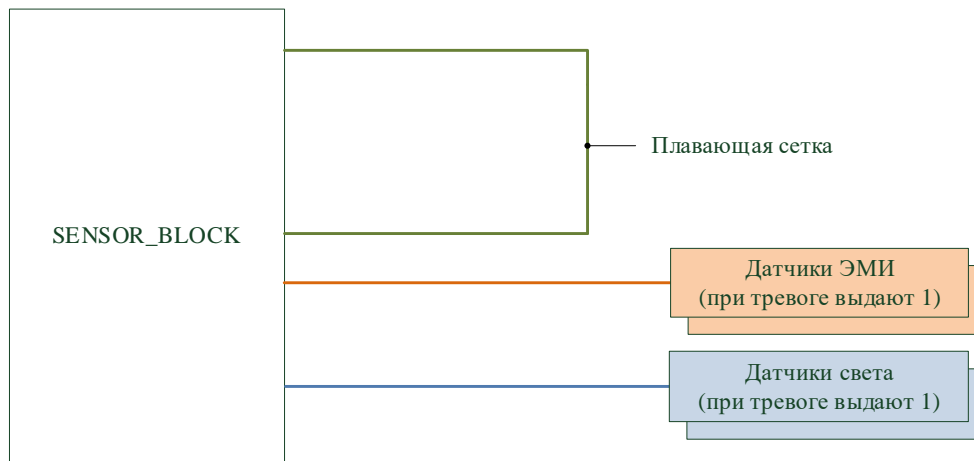


Рисунок 110 – Условная схема блока SENSORS

По сигналу тревоги от датчиков возможен сброс памяти регистров и криптографических ключей и формирование сигнала прерывания.

### 34.2 Описание функционирования блока и режимов работы

Блок непрерывно отслеживает состояние входов датчиков и защитной сетки. Работа датчиков разрешается записью в регистр **enable**, возможна только однократная запись. Состояние датчиков отражается в регистры **state\_reg** (фиксируется факт срабатывания датчиков за все время после сброса микросхемы) и **real\_time** (текущее состояние сигналов тревоги от датчиков). Сброс флагов срабатывания датчиков света и ЭМИ производится записью 0 в биты 0-3 **state\_reg**, только при условии отсутствия ненулевого входа. Сброс флага срабатывания защитной сетки производится записью 0 в бит 4 регистра **state\_reg**, при записи в момент проверки (конец временного интервала, заданного в регистре **meshcntr**) сброс произойдет только при условии совпадения уровней входа и выхода. Запись 0 в середине интервала переключения сетки, сбросит флаг срабатывания до ближайшей проверки. Запись 1 в биты регистра **state\_reg** не меняет его состояния.

Для плавающей защитной сетки могут быть однократно настроены ключ и данные инициализации по ГОСТ 28147-89 через регистры **mesh\_key1-mesh\_key7** и **mesh\_init0-mesh\_init1**, а также время ее переключения в тактах частоты HSI записью в регистр **meshcntr**.

Модуль SENSORS имеет специальные регистры настройки с выбором датчиков, вызывающих формирование сигнала запроса прерывания и сигнала запроса сброса памяти ключей. Сигнал запроса прерывания и сброса формируется при наличии в **status\_reg** единичных битов, помеченных маской в регистрах **int\_mask** и **k\_res\_mask** соответственно.

### 34.3 Карта регистров

Таблица 536 содержит названия и адреса всех программно-доступных регистров блока SENSORS. Поле Таблицы «Адрес» указывает относительный адрес регистра в шестнадцатеричной форме. Поле Таблицы «Доступ» указывает на политику доступа к заданному регистру: r/w – чтение и запись, r/o – только чтение (запись не имеет эффекта).

Таблица 536 – Карта регистров SENSORS

Название регистра	Адрес	Доступ	Общее описание регистра
state_reg	0x00	r/w	Регистр статуса → сигналы с датчиков
real_time	0x04	r/o	Регистр мониторинга состояния входов сенсоров
int_mask	0x0C	r/w	Регистр маски запроса прерывания
k_res_mask	0x10	r/w	Регистр маски запроса сброса
enable	0x18	r/w	Регистр включения датчиков
mesh_key7	0x1C	r/w	Регистр ключа защитной сетки 0 (биты 255:224)
mesh_key6	0x20	r/w	Регистр ключа защитной сетки 1 (биты 223:192)
mesh_key5	0x24	r/w	Регистр ключа защитной сетки 2 (биты 191:160)
mesh_key4	0x28	r/w	Регистр ключа защитной сетки 3 (биты 159:128)
mesh_key3	0x2C	r/w	Регистр ключа защитной сетки 4 (биты 127:96)
mesh_key2	0x30	r/w	Регистр ключа защитной сетки 5 (биты 95:64)
mesh_key1	0x34	r/w	Регистр ключа защитной сетки 6 (биты 63:32)
mesh_key0	0x38	r/w	Регистр ключа защитной сетки 7 (биты 31:0)
mesh_init1	0x3C	r/w	Регистр инициализации данных защитной сетки 0 (биты 63:32)
mesh_init0	0x40	r/w	Регистр инициализации данных защитной сетки 1 (биты 31:0)
meshcntr	0x44	r/w	Регистр управления защитной сетки

### 34.3.1 Регистр state\_reg

Относительный адрес: 0x00.

Доступ: r/w

Значение после сброса: 0x0.

Регистр статуса **state\_reg** предназначен для отображения срабатывания датчиков.

Таблица 537 – Регистр state\_reg

Разряды	Назначения разрядов
31...5	Зарезервировано
4	Сигнал атаки на защитную сетку – mesh active alarm
3	Сигнал датчика ЭМИ 3 – mesh emfi alarm3
2	Сигнал датчика ЭМИ 2 – mesh emfi alarm2
1	Сигнал датчика ЭМИ 1 – mesh emfi alarm1
0	Сигнал датчика света

### 34.3.2 Регистр real\_time

Относительный адрес: 0x04.

Доступ: r/o.

Значение после сброса: 0x0.

Регистр **real\_time** предназначен для непрерывного мониторинга состояния входов сенсоров.

Таблица 538 – Регистр real\_time

Разряды	Назначения разрядов
31...5	Зарезервировано
4	Сигнал атаки на защитную сетку - mesh active alarm
3	Сигнал датчика ЭМИ 3 – mesh emfi alarm3
2	Сигнал датчика ЭМИ 2 – mesh emfi alarm2
1	Сигнал датчика ЭМИ 1 – mesh emfi alarm1
0	Сигнал датчика света

### 34.3.3 Регистр int\_mask

Относительный адрес: 0x0C.

Доступ: r/w.

Значение после сброса: 0x0.

Регистр **int\_mask** предназначен для выбора датчиков, вызывающих формирование сигнала запроса прерывания. Разряды распределены так же, как в регистре **state\_reg**. Сигнал прерывания формируется при наличии единичных значений в регистре статуса (**state\_reg**) в битах, помеченных единицами в регистре маски (**int\_mask**). Возможна только однократная запись в регистр.

### 34.3.4 Регистр k\_res\_mask

Относительный адрес: 0x10.

Доступ: r/w.

Значение после сброса: 0x0.

Регистр **k\_res\_mask** предназначен для выбора датчиков, вызывающих формирование сигнала запроса сброса памяти ключей. Разряды распределены так же, как в регистре **state\_reg** (таблица 537). Сигнал прерывания формируется при наличии единичных значений в регистре статуса (**state\_reg**) в битах, помеченных единицами в регистре маски (**key\_res\_mask**). Возможна только однократная запись в регистр.

### 34.3.5 Регистр enable

Относительный адрес: 0x18.

Доступ: r/w.

Значение после сброса: *все датчики выключены*

Регистр **enable** предназначен для включения-выключения датчиков безопасности, соответствие бит датчикам такое же, как в регистре **state\_reg**. Единица в бите соответствующего датчика – датчик включен, ноль – датчик выключен. Возможна только однократная запись в регистр.

Таблица 539 – Регистр enable

Разряды	Назначения разрядов
31...5	Зарезервировано
4	Разрешение mesh active alarm
3	Разрешение mesh emfi alarm3
2	Разрешение mesh emfi alarm2
1	Разрешение mesh emfi alarm1
0	Разрешение датчика света

### 34.3.6 Регистр meshcntr

Относительный адрес: 0x44.

Доступ: r/w.

Значение после сброса: 0x0.

Возможна только однократная запись в регистр.

Таблица 540 – Регистр meshcntr

Разряды	Назначения разрядов
31...3	Зарезервировано
2...0	Регистр делителя опорной частоты для работы сетки 000 – переключение каждые 3 такта 001 – переключение каждые 4 такта 010 – переключение каждые 8 тактов 011 – переключение каждые 16 такта 100 – переключение каждые 32 такта 101 – переключение каждые 64 такта 110 – переключение каждые 128 такта 111 – переключение каждые 256 тактов

### 35 Датчик частоты (CLK\_MEASURE)

Блок датчика частоты является важным элементом защиты защищенной части микросхемы. Чтобы избежать возможных атак через изменение тактовой частоты, реализован датчик частоты, который сравнивает подаваемый синхросигнал с опорным, и генерирует один из двух типов событий: сброс ключей батарейного домена, прерывание, в случае существенного отличия синхросигналов. Необходимо обеспечить невозможность отключения синхросигнала HSI.

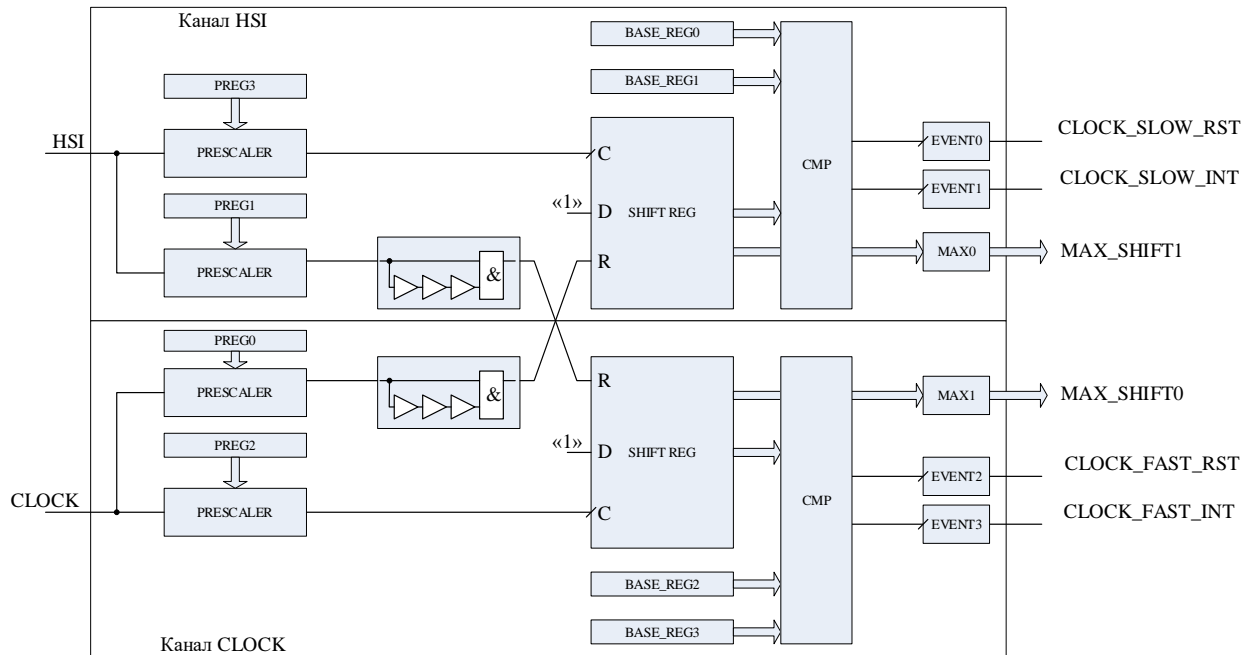


Рисунок 111 – Структурная схема блока датчика частоты

При настройке блока, исходя из ожидаемых значений частот сигналов HSI и CLOCK, необходимо задать значения предварительных делителей в регистрах **PREGx** так, чтобы

$$\left\{ \begin{array}{l} \frac{f_{HSI}}{PREG3 + 1} = K_0, \text{ где } K_0 \in [1..15] \\ \frac{f_{CLOCK}}{PREG0 *} \\ \frac{f_{CLOCK}}{PREG2 + 1} = K_1, \text{ где } K_1 \in [1..15] \\ \frac{f_{HSI}}{PREG1 *} \end{array} \right.$$

- \*  $PREG0$  – при  $f_{HSI} > f_{CLOCK}$ ,  
 $PREG0 + 1$  – при  $f_{HSI} < f_{CLOCK}$ ,
- \*  $PREG1$  – при  $f_{HSI} < f_{CLOCK}$ ,  
 $PREG1 + 1$  – при  $f_{HSI} > f_{CLOCK}$ .

Тогда в регистре **SHIFT\_REG** канала HSI будут возникать значения от 0 до  $2^{K_0}-1$ , а в канале CLOCK – от 0 до  $2^{K_1}-1$ .

При снижении частоты CLOCK:

– в регистре канала HSI будут возникать значения от 0 до  $2^{K_2}-1$ , причем  $K_2 > K_0$ ;

– в регистре канала CLOCK будут возникать значения от 0 до  $2^{K_3}-1$ , причем  $K_3 < K_1$ .

Если значение  $2^{K_2}-1$  превысит или равно значению  $\text{BASE\_REG0}=2^{K_0}-1$ , то возникнет событие EVENT0, а если  $2^{K_3}-1$  превысит или равно значению  $\text{BASE\_REG1}=2^{K_0}-1$ , то возникнет событие EVENT1.

$$\begin{cases} f_{\text{CLOCK}} \leq \frac{f_{\text{HSI}} \times \text{PREG0} *}{(\text{PREG3} + 1) \times (K_0 + 1)} \Rightarrow \text{EVENT0} \\ f_{\text{CLOCK}} \leq \frac{f_{\text{HSI}} \times \text{PREG0} *}{(\text{PREG3} + 1) \times (K_0 + 1)} \Rightarrow \text{EVENT1} \end{cases}$$

При увеличении частоты CLOCK:

– в регистре канала HSI будет возникать значения от 0 до  $2^{K_2}-1$ , причем  $K_2 < K_0$ ;

– в регистре канала CLOCK будет возникать значения от 0 до  $2^{K_3}-1$ , причем  $K_3 > K_1$ .

Если значение  $2^{K_3}-1$  превысит или равно значению  $\text{BASE\_REG2}=2^{K_1}-1$ , то возникнет событие EVENT2, если  $2^{K_3}-1$  превысит или равно значению  $\text{BASE\_REG3}=2^{K_1}-1$ , то возникнет событие EVENT3.

$$\begin{cases} f_{\text{CLOCK}} \geq \frac{f_{\text{HSI}} \times (\text{PREG2} + 1) \times (K_1 + 1)}{\text{PREG1} *} \Rightarrow \text{EVENT2} \\ f_{\text{CLOCK}} \geq \frac{f_{\text{HSI}} \times (\text{PREG2} + 1) \times (K_1 + 1)}{\text{PREG1} *} \Rightarrow \text{EVENT3} \end{cases}$$

Для иллюстрации принципов работы блока возможные комбинации настроек приведены в таблице 541.

Таблица 541 – Возможные комбинации настроек

Частота HSI	Ожидаемая частота CLOCK	Реальная частота CLOCK	PREG3	PREG1	PREG0	PREG2	BASE REG0	BASE REG1	BASE REG2	BASE REG3	Примечание
8 МГц	60 МГц	60 МГц	1	4	29	9	4	4	8	8	K0 = 2 K1 = 3 Нет событий



Частота HSI	Ожидаемая частота CLOCK	Реальная частота CLOCK	PREG3	PREG1	PREG0	PREG2	BASE REG0	BASE REG1	BASE REG2	BASE REG3	Примечание
8 МГц	60 МГц	80 МГц	1	4	29	9	4	4	8	8	K0 = 2 K1 = 3 K2 = 1,5 K3 = 4 EVENT2 EVENT3 Высокая частота
8 МГц	60 МГц	30 МГц	1	4	29	9	4	4	8	8	K0 = 2 K1 = 3 K2 = 4 K3 = 1,5 EVENT0 EVENT1 Низкая частота

Также можно программно считать максимальное значение регистра **SHIFT\_REG** каждого из каналов, которое он достигал. При чтении регистра **CLK\_STAT** считываются теньевые регистры, в которых отражаются максимальные достигнутые регистрами **SHIFT\_REG1** и **SHIFT\_REG0** значения с последнего возникновения прерывания или события сброса за период с последнего сброса теневого регистра. Сброс теневого регистра осуществляется программно, записью управляющего бита **EN=0**.

### 35.1 Карта регистров

Таблица 542 – Карта регистров

Базовый Адрес		Название	Описание
0x400B_0000		CLK_MEASURE	Блок датчика частоты
Смещение			
0x0000	0	CLK_CNTR_STAT	Регистр настроек и состояния
0x0004	1	ALARM_SHIFT_RST	Регистр задания пороговых значений для сигнала сброса
0x0008	2	ALARM_SHIFT_INT	Регистр задания пороговых значений для сигнала прерывания
0x000c	3	ALARM_PREG_0	Регистр настройки параметров сброса счетчиков
0x0010	4	ALARM_PREG_1	Регистр настройки параметров сброса счетчиков
0x0014	5	CLK_STAT	Регистр максимального значения <b>SHIFT_REG</b> каждого из каналов

### 35.1.1 CLK\_CNTR\_STAT

Таблица 543 – Регистр CLK\_CNTR\_STAT

Бит	Имя	Значение	Описание
15...6	Зарезервировано		Зарезервировано, читается как 0
5	KEY_RESET		Флаг возникновения события сброса ключей
4	INTERRUPT		Флаг возникновения прерывания
3...1	Зарезервировано		Зарезервировано, читается как 0
0	EN		Бит разрешения работы блока контроля частоты: 0 – блок выключен; 1 – блок включен

### 35.1.2 ALARM\_SHIFT\_RST

Таблица 544 – Регистр ALARM\_SHIFT\_RST

Бит	Имя	Значение	Описание
31...16	BASE_REG2		Значение MAX_SHIFT0, при котором возникает событие сброса ключей
15...0	BASE_REG0		Значение MAX_SHIFT1, при котором возникает событие сброса ключей

### 35.1.3 ALARM\_SHIFT\_INT

Таблица 545 – Регистр ALARM\_SHIFT\_INT

Бит	Имя	Значение	Описание
31...16	BASE_REG3		Значение MAX_SHIFT0, при котором возникает прерывание
15...0	BASE_REG1		Значение MAX_SHIFT1, при котором возникает прерывание

### 35.1.4 ALARM\_PREG\_0

Таблица 546 – Регистр ALARM\_PREG\_0

Бит	Имя	Значение	Описание
31...16	PREG0		Поле основания счета для счетчика 0. Всегда задавать больше нуля
15...0	PREG1		Поле основания счета для счетчика 1. Всегда задавать больше нуля

### 35.1.5 ALARM\_PREG\_1

Таблица 547 – Регистр ALARM\_PREG\_1

Бит	Имя	Значение	Описание
31...16	PREG2		Поле основания счета для счетчика 2. Всегда задавать больше нуля
15...0	PREG3		Поле основания счета для счетчика 3. Всегда задавать больше нуля

### 35.1.6 CLK\_STAT

Таблица 548 – Регистр CLK\_STAT

Бит	Имя	Значение	Описание
31...16	MAX_SHIFT0	0x0000	Максимальное значение регистра MAX_SHIFT0 с последнего возникновения прерывания или события сброса
15...0	MAX_SHIFT1	0x0000	Максимальное значение регистра MAX_SHIFT1 с последнего возникновения прерывания или события сброса

## 36 Генератор случайных чисел (RANDOM)

### 36.1 Назначение и свойства

Модуль служит для управления двумя кольцевыми генераторами, их включения, тестирования, а также формирования случайного 32-битного числа.

Модуль имеет следующие особенности:

- содержит два генератора;
- позволяет включать и выключать генераторы группами по одному;
- позволяет задать паузу после включения генераторов, перед началом сбора случайного числа;
- позволяет включать генераторы по биту регистра управления;
- имеет несколько режимов запуска сбора случайного числа:
  - одиночный;
  - постоянный.
- формирует сигнал прерывания по окончании сбора случайного числа.

Датчик случайных чисел представляет собой механизм получения непредсказуемой последовательности чисел, основанный на физическом недетерминированном процессе.

Для увеличения количества случайных чисел и улучшения их статистических характеристик, пользователь может воспользоваться аппаратно-программным механизмом генерации псевдослучайных чисел. В этом случае выход генератора случайных чисел является рандомизирующим фактором для выполнения операции шифрования с использованием стойкого шифра.



Рисунок 112 – Структурная схема блока генератора случайных чисел

### 36.2 Регистры модуля

Таблица 549 – Регистры модуля

Базовый Адрес	Название	Описание
0x400B_8000	RANDOM	Блок генератора случайных чисел
Смещение		
0x00	STAT_CTRL_REG	Регистр статуса и управления
0x04	INT_CTRL_REG	Регистр управления прерыванием
0x08	CLK_DIV_REG	Регистр делителя синхросигнала генератора
0x0C	PAUSE_REG	Регистр паузы включения
0x10	OUTPUT_REG	Регистр случайного значения
0x14	PAUSE_CNT_REG	Регистр счетчика паузы
0x18	TEMP_REG	Регистр сбора случайного числа

**36.2.1 Регистр статуса и управления STAT\_CTRL\_REG**

Относительный адрес: 0x00, начальное значение 0x000000F0.

Таблица 550 – Регистр STAT\_CTRL\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
0	–	Зарезервировано. Записывать всегда 0	RW
1	EN_STATE	Генераторы включены, данный бит устанавливается в 1, если есть хоть одна группа включенных генераторов (значение MASK не нулевое), и закончилась пауза после включения (см. PAUSE_REG)	RO
2	BUSY	Генератор занят, идет сбор случайного числа	RO
3	UNREAD_DATA	Флаг наличия несчитанных данных. После сбора очередного случайного числа в данном бите появляется 1, бит сбрасывается при чтении случайного числа (см. OUTPUT_REG)	RO
5, 4	MASK	Маска выбора активного генератора: 4 бит – генератор 0; 5 бит – генератор 1	RW
7, 6	–	Зарезервировано	RO
9, 8	WORK_MODE	Режим сбора случайного числа: 00 – сбор случайных чисел остановлен; 01 – однократный запуск сбора случайного числа (после начала сбора поле сбрасывается в 00 автоматически); 10 – сбор нового случайного числа начинается если нет непрочитанного случайного числа (UNREAD_DATA равен 0); 11 – постоянный сбор случайных чисел, сбор нового случайного числа начинается сразу по окончании сбора очередного числа	RW
11, 10	–	Зарезервировано	RO
17...12	BIT_CNT	Счетчик оставшихся бит до сбора нового случайного числа	RO
31...18	–	Зарезервировано	RO

### 36.2.2 Регистр управления прерыванием INT\_CTRL\_REG

Относительный адрес: 0x04.

Таблица 551 – Регистр INT\_CTRL\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
0	INT_EN	Разрешение прерывания. При наличии 1 в этом бите и 1 в поле UNREAD_DATA, на выходе interrupt появляется 1	RW
1	UNREAD_DATA	Флаг наличия несчитанных данных. После сбора очередного случайного числа в данном бите появляется 1, бит сбрасывается при чтении случайного числа (см. OUTPUT_REG)	RO
31...2	-	Зарезервировано	RO

### 36.2.3 Регистр делителя клона генератора CLK\_DIV\_REG

Относительный адрес: 0x08, начальное значение 7.

Таблица 552 – Регистр CLK\_DIV\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
15...0	DIV	Для формирования тактового сигнала защелки данных генератора используется входной тактовый сигнал модуля, деленный на $2(DIV + 1)$	RW
31...16	-	Зарезервировано	RO

### 36.2.4 Регистр паузы включения PAUSE\_REG

Относительный адрес: 0x0C.

Таблица 553 – Регистр PAUSE\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31...0	PAUSE	После включения генераторов через внешний сигнал или бит управляющего регистра, при включении любого блока через маску, при добавлении новых включенных блоков к работающим, сбор случайного числа может быть начат не ранее, чем закончится данная пауза. Если включение происходит во время сбора случайного числа, сбор приостанавливается до тех пор, пока не закончится данная пауза	RW

### 36.2.5 Регистр случайного значения OUTPUT\_REG

Относительный адрес: 0x10.

Таблица 554 – Регистр OUTPUT\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31...0	DATA	Последнее собранное случайное число. Во время сбора очередного случайного числа, значение данного регистра неизменно. Обновление данных происходит по окончании сбора случайного числа	RO

### 36.2.6 Регистр счетчика паузы PAUSE\_CNT\_REG

Относительный адрес: 0x14.

Таблица 555 – Регистр PAUSE\_CNT\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31...0	CUR_PAUSE	Текущее значение счетчика паузы включения генераторов (значение для справки). 0 значение означает, что пауза выдержана	RO

### 36.2.7 Регистр сбора случайного числа TEMP\_REG

Относительный адрес: 0x18.

Таблица 556 – Регистр TEMP\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31...0	TEMP_DATA	Текущее значение сдвигового регистра сбора случайного числа (значение для справок)	RO

## 37 Контроллер UART (ISO7816)

### 37.1 Введение

Универсальный синхронно-асинхронный приемопередатчик (USART) предлагает гибкие средства полнодуплексного обмена с внешним оборудованием, требующим стандартного формата кодирования данных NRZ. Модуль USART поддерживает широкий диапазон скорости передачи данных с использованием генератора дробной скорости передачи данных.

Модуль поддерживает синхронную одностороннюю передачу данных и полудуплексную однопроводную передачу данных. Также поддерживается протокол Smartcard, и операции модема (CTS/RTS).

### 37.2 Особенности модуля USART

- Полный дуплекс, асинхронный обмен данными;
- стандартный формат данных NRZ (Mark/Space);
- конфигурируемый метод передискретизации на 16 или 8 для обеспечения гибкости между скоростью и допуском по тактовой частоте;
- системы генератора дробной скорости передачи данных:
  - программируемая скорость передачи и приема данных (значение скорости передачи данных при максимальной частоте шины APB см. в спецификациях);
  - программируемая длина слова данных (восемь или девять бит);
  - поддержка 2 стоп-бит;
  - пактовый выход передатчика для синхронной передачи данных;
  - возможность эмуляции смарт-карты (режим Smartcard);
    - интерфейс Smartcard поддерживает асинхронный протокол для смарт-карт в соответствии со стандартом ISO 7816-3;
    - стоп-биты 0,5, 1,5 для интерфейса Smartcard без возможности применения back2back-кадров в режиме 0,5-бит. Подробнее см. в пункте 37.3.1 «Описание символов USART»;
- однопроводная полудуплексная передача данных;
- отдельные биты разрешения (enable) для передатчика и приемника;
- флаги обнаружения передачи:
  - буфер приемника полон;
  - буфер передатчика пуст;
  - флаг завершения передачи;
- проверка четности:
  - передача бита четности;
  - проверка четности принятого байта данных;
- флаги обнаружения ошибки:
  - переполнение;
  - обнаружение шума;
  - ошибка фрейма;



- ошибка четности;
- прерывания с флагами:
  - изменение состояния CTS;
  - регистр данных передатчика пуст;
  - передача завершена;
  - регистр данных приемника заполнен;
  - ошибка переполнения;
  - ошибка фрейма;
  - обнаружение шума;
  - ошибка четности.

### 37.3 Функциональное описание USART

Интерфейс подключается к внешним приборам при помощи трех выводов (см. рисунок 113). Любая двунаправленная передача данных USART требует минимум два вывода: вход для данных (RX) и выход для передачи данных (TX):

- **RX**: вход для последовательных принимаемых данных. Для восстановления данных используется техника передискретизации, чтобы отделить нужные входящие данные от шума;
- **TX**: выход для передачи данных. Когда передатчик запрещен, вывод возвращается в состояние, заданное конфигурацией порта ввода-вывода (I/O).

Когда передатчик разрешен, но никакие данные не передаются, на выводе TX устанавливается высокий уровень (логическая 1). В однопроводном режиме и режиме смарт-карты, данный порт ввода-вывода используется для передачи и приема данных (SW\_RX).

При помощи данных выводов происходит передача и прием последовательных данных в нормальном режиме работы модуля USART, в виде фреймов, содержащих:

- сигнал ожидания линии (Idle) до передачи или приема;
- стартовый бит;
- слово данных (8 или 9 бит), LSB передается первым;
- 0,5, 1,5, 2 стоп-бита, показывающие, что фрейм завершен;
- данный интерфейс использует дробный генератор скорости передачи данных - с 12-битной мантиссой и четырехбитной дробной частью;
- регистр статуса (USART\_SR);
- регистр данных (USART\_DR);
- регистр скорости передачи (USART\_BRR) – 12-битная мантисса и четырехбитная дробная часть;
- регистр защитного интервала (Guardtime register) (USART\_GTPR) в случае использования режима Smartcard.

Описание регистров и их бит приведено в подразделе 37.5 «Регистры USART». Для работы в синхронном режиме требуется дополнительный вывод:

– **СК:** Выход тактов передатчика. На этот вывод выдаются такты данных для синхронной передачи, соответствующей режиму ведущий SPI (нет тактовых импульсов в стартовых и стоп- битах, и программная опция отправки тактового импульса на последнем бите данных). Параллельно данные могут синхронно приниматься по выводу RX. Это можно использовать для управления внешними периферийными устройствами, у которых есть регистры сдвига (например, LCD драйверы). Фаза и полярность тактов выбирается программно. В режиме Smartcard, вывод СК обеспечивает такты для смарт-карты.

В режиме аппаратного управления потоком данных требуются дополнительные выводы:

- **CTS:** сигнал Clear To Send блокирует передачу данных по окончании текущей передачи при высоком уровне сигнала;
- **RTS:** сигнал Request to send показывает готовность USART к приему данных (при низком уровне сигнала).

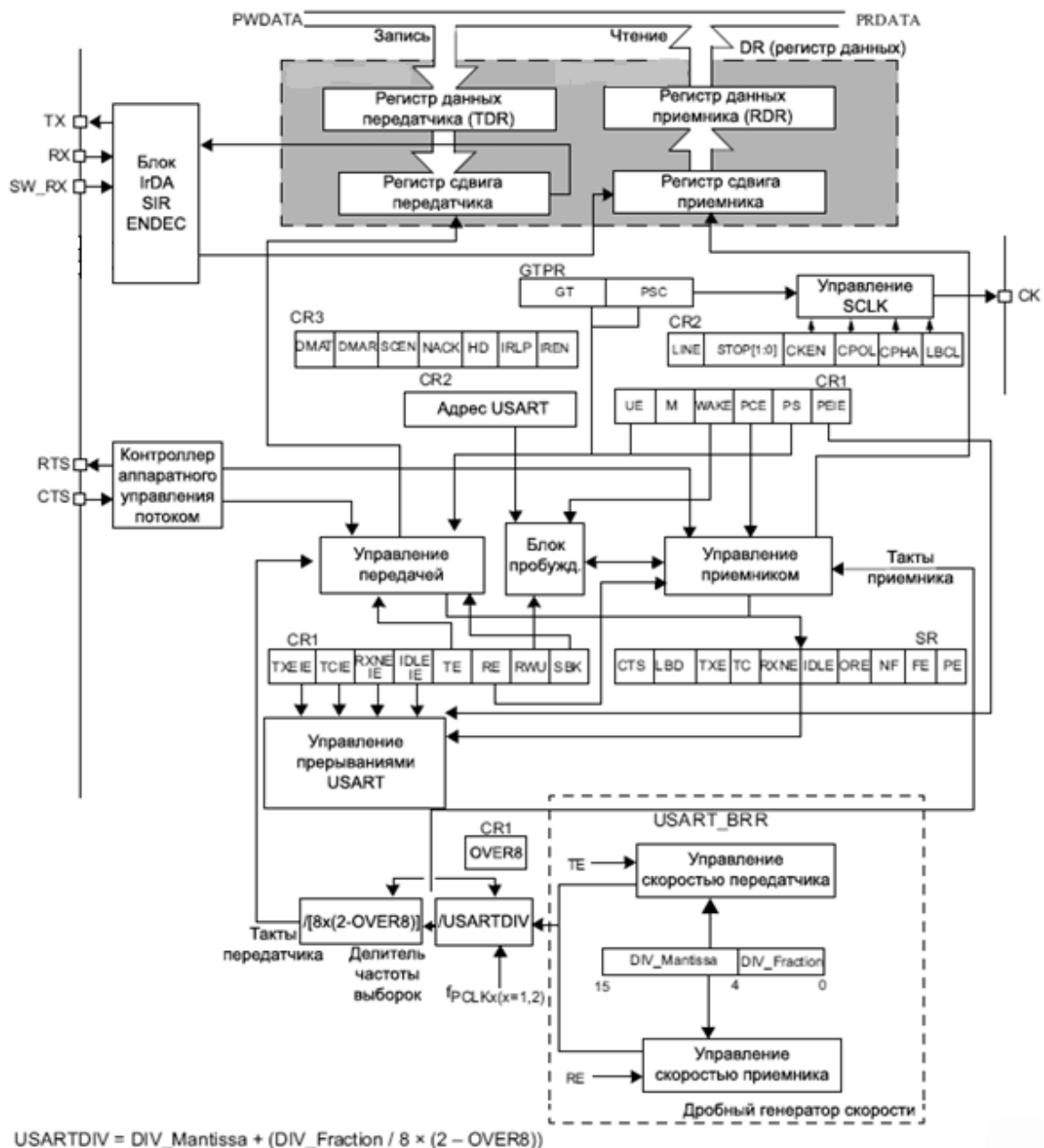


Рисунок 113 – Блок схема USART

### 37.3.1 Описание символов USART

Для передачи можно выбрать длину слова восемь или девять бит путем программирования бита M в регистре USART\_CR1 (см. рисунок 114).

Вывод TX находится в состоянии логического 0 во время передачи стартового бита, и в состоянии логической 1 во время передачи стоп-бита.

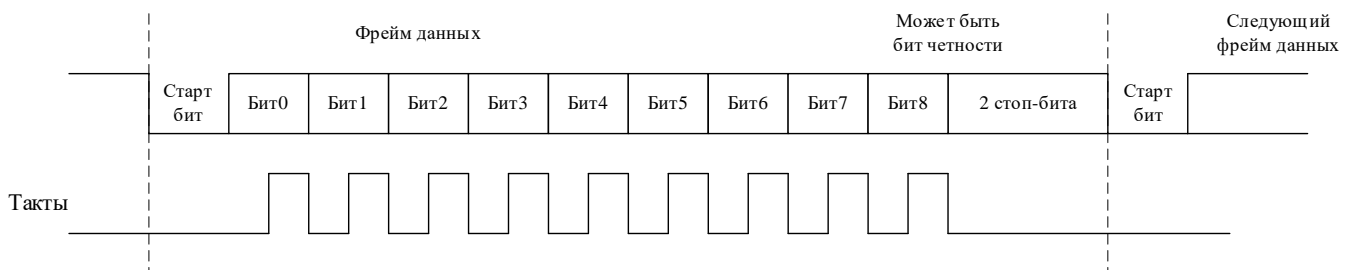
Передача и прием управляются генератором скорости передачи. Такты для передатчика и приемника генерируются, когда установлен соответствующий бит разрешения.

Ниже подробно описан каждый блок.

Примечание – При выборе режима работы 0,5 стоп-бит не применять back2back-кадры\*, так как модуль USART не успевает принять кадр из-за ожидания в один такт между кадрами (режим T=1, ISO7816-3).

\* back2back-кадры – это кадры, время паузы между которыми меньше одного такта.

Длина слова 9 бит (бит M установлен), 2 стоп-бита



Длина слова 8 бит (бит M сброшен), 2 стоп-бита

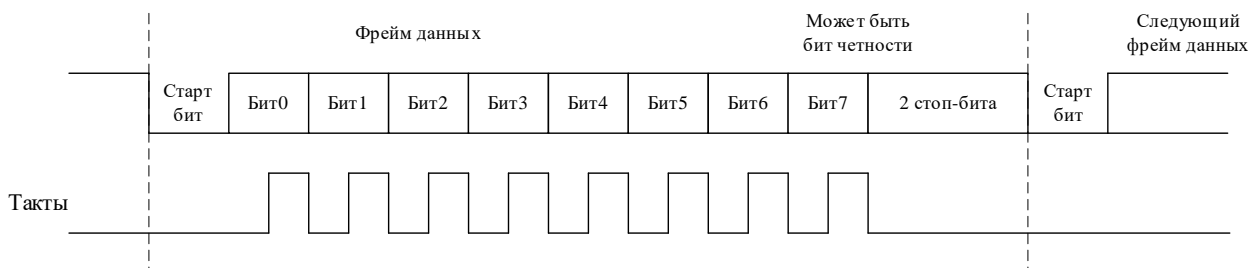


Рисунок 114 – Программирование длины слова

### 37.3.2 Передатчик

Передатчик может отправлять слова данных по 8 или 9 бит, в зависимости от статуса бита M.

Когда установлен бит разрешения передачи (TE), данные из регистра сдвига выводятся на вывод TX, и соответствующие тактовые импульсы выводятся на вывод СК.

#### 37.3.2.1 Передача символа

Во время передачи USART данные сдвигаются на вывод TX, младший значащий бит (LSB) идет первым. В данном режиме, регистр USART\_DR состоит из буфера (TDR) между внутренней шиной и регистром сдвига передачи (см. рисунок 113).

Каждому символу предшествует стартовый бит низкого уровня (логический 0) в течение длительности одного бита. Символ завершается конфигурируемым количеством стоп-битов.

USART модуль поддерживает следующие стоп-биты: 0,5, 1,5 и 2 стоп-бита.

Примечание – бит TE не должен сбрасываться по время передачи данных. Сброс бита TE во время передачи приведет к повреждению данных на выводе TX, так как счетчики генератора скорости остановятся. Текущие передаваемые данные будут потеряны.

После разрешения бита TE будет отправлен фрейм idle.

### 37.3.2.2 Конфигурируемые стоп-биты

Количество передаваемых стоп-бит с каждым символом может быть запрограммировано в Регистре Управления 2 битами 13, 12.

- 2 стоп-бита: Это значение поддерживается в обычном режиме USART, однопроводном режиме и режиме модема;
- 0,5 стоп-бит: для использования при приеме данных в режиме Smartcard;
- 1,5 стоп-бит: для использования при передаче и приеме данных в режиме Smartcard.

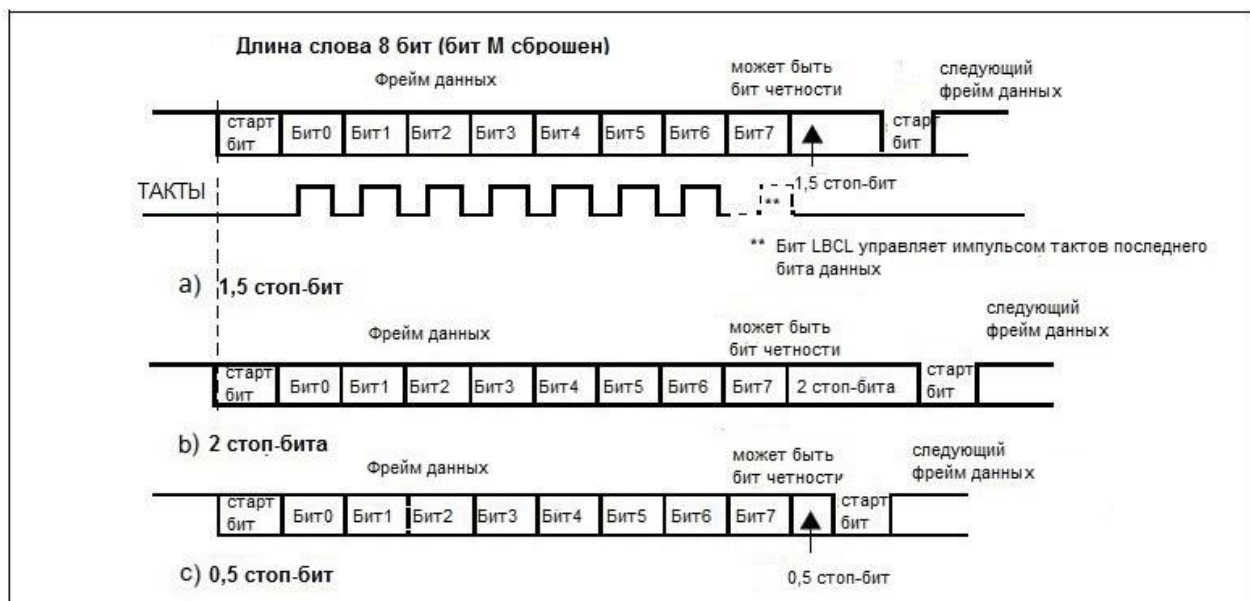


Рисунок 115 – Конфигурируемые стоп-биты

Процедура программирования:

- 1 Разрешить USART запись в 1 бита UE в регистре USART\_CR1;
- 2 Запрограммировать бит M в регистре USART\_CR1, чтобы определить длину слова данных;
- 3 Запрограммировать количество стоп-бит в регистре USART\_CR2;
- 4 Выбрать необходимую скорость передачи при помощи регистра USART\_BRR;
- 5 Установить бит TE в регистре USART\_CR1;
- 6 Записать данные для передачи в регистр USART\_DR (это сбросит бит TXE). Повторить данную операцию для каждого передаваемого символа в случае использования одиночного буфера;

7 После записи последних данных в регистр USART\_DR, подождать, пока ТС станет равным 1. Это покажет, что передача последнего фрейма завершена. Например, это требуется, когда модуль USART отключен или переходит в режиме Halt, чтобы избежать повреждения последней передачи.

### 37.3.2.3 *Однobaйтный обмен*

Бит TXE всегда сбрасывается при записи в регистр данных.

Бит TXE устанавливается аппаратно, и указывает, что:

- данные были перемещены из регистра TDR в регистр сдвига, и началась передача;
- регистр TDR пуст;
- следующие данные могут быть записаны в регистр USART\_DR без риска перезаписи предыдущих данных.

Данный флаг генерирует прерывание, если установлен бит TXE.

Во время передачи данных команда записи в регистр USART\_DR помещает данные в регистр TDR, откуда затем данные копируются в регистр сдвига при завершении текущей передачи.

Если передачи данных не происходит, команда записи в регистр USART\_DR помещает данные напрямую в регистр сдвига, начинается передача данных, и бит TXE незамедлительно устанавливается.

Если передается фрейм (после стоп-бита) и бит TXE установлен, бит ТС устанавливается в высокий уровень (логическая 1). Генерируется прерывание, если бит TCIE установлен в регистре USART\_CR1.

После записи последних данных в регистр USART\_DR важно дождаться, когда ТС установится в 1, перед запретом USART или переходом микросхемы в режим пониженного потребления (см. рисунок 116).

Бит ТС сбрасывается следующей программной последовательностью:

- 1 Чтение из регистра USART\_SR;
- 2 Запись в регистр USART\_DR.

Примечание – Бит ТС может быть также сброшен при помощи записи «0». Рекомендуется использовать данную последовательность для сброса бита только в случае мультибуферного обмена данными.

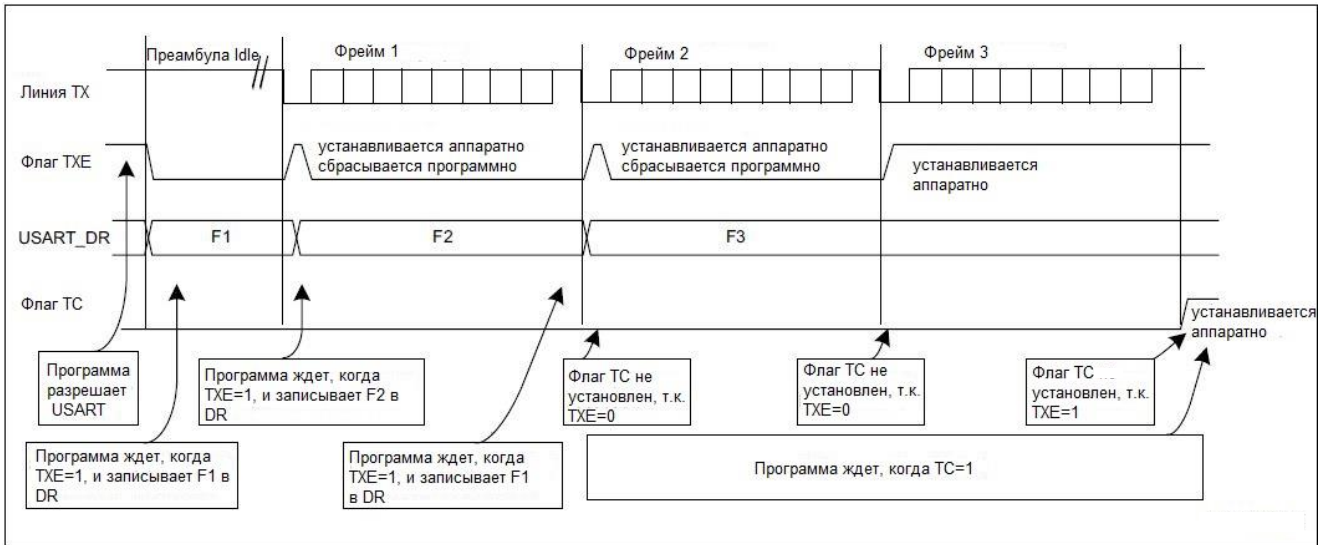


Рисунок 116 – Состояние TC/TXE во время передачи

### 37.3.3 Приемник

USART может принимать слово данных разрядностью 8 или 9 бит, в зависимости от значения бита M в регистре USART\_CR1.

#### 37.3.3.1 Обнаружение стартового бита

Последовательность обнаружения стартового бита одинаковая при передискретизации сигнала с кратностью 16 или 8.

В приемопередатчике USART стартовый бит детектируется, когда распознана определенная последовательность выборки сигнала: 1110X0X0000.

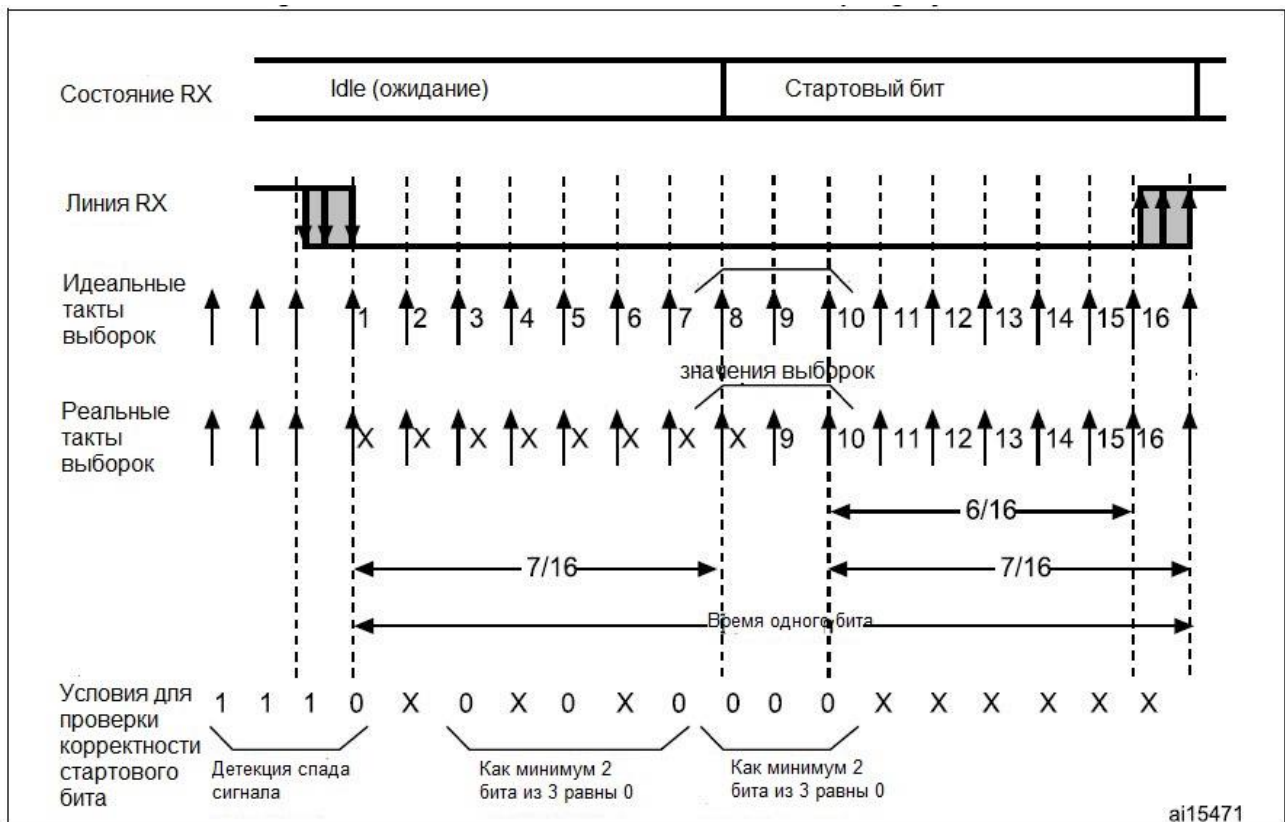


Рисунок 117 – Обнаружение стартового бита при передискретизации на 16 и 8

Примечание – Если последовательность не завершена, обнаружение стартового бита прекращается, и приемник переходит в состояние ожидания (idle) (никакие флаги не установлены) и ожидает заднего фронта.

Стартовый бит подтвержден (флаг RXNE установлен, генерируется прерывание при RXNEIE=1), если 3 бита выборки равны 0 (первая выборка на 3, 5 и 7 бите находит три бита в логическом 0, а вторая выборка на 8, 9 и 10 битах также находит три бита в 0).

Стартовый бит подтвержден (флаг RXNE установлен, генерируется прерывание при RXNEIE=1), но устанавливается флаг NF, если при обеих выборках как минимум 2 из 3 бит установлены в 0 (выборка на 3, 5 и 7 битах и выборка на 8, 9 и 10 битах). Если данное условие не выполняется, не происходит обнаружение стартового бита, и приемник возвращается в состояние ожидания (idle) (нет установленных флагов).

Если при одной из выборок (выборка на 3, 5 и 7 битах или выборка на 8, 9 и 10 битах) 2 из 3 бит находятся в логическом 0, стартовый бит подтверждается, но устанавливается флаг NE.

### 37.3.3.2 Прием символа

При приеме данных данные перемещаются на вывод RX, начиная с LSB. В данном режиме регистр USART\_DR состоит из буфера (RDR) между внутренней шиной и регистром сдвига приемника.

Процедура программирования:

- 1 Разрешить USART путем записи в 1 бита UE в регистре USART\_CR1.
- 2 Запрограммировать бит M в регистре USART\_CR1 для определения длины слова данных.
- 3 Запрограммировать количество стоп-бит в регистре USART\_CR2.
- 4 Выбрать необходимую скорость передачи данных при помощи регистра USART\_BRR
- 5 Установить бит RE в регистре USART\_CR1. Это активирует приемник, и он начнет обнаружение стартового бита.

Когда принят символ:

- устанавливается бит RXNE. Он показывает, что содержимое сдвигового регистра передано в RDR. Другими словами, данные были получены и могут быть прочитаны (а также связанные с ними флаги ошибок);
- генерируется прерывание, если установлен бит RXNEIE;
- флаг ошибки устанавливается, если во время приема была обнаружена ошибка фрейма, шума или переполнения;
- в мультибуферном режиме бит RXNE устанавливается после каждого принятого байта и сбрасывается чтением в регистре данных;
- в режиме одиночного буфера бит RXNE сбрасывается программно при чтении регистра;
- USART\_DR. Флаг RXNE сбрасывается записью 0. Бит RXNE должен быть сброшен перед окончанием приема следующего символа, чтобы избежать ошибки переполнения.

Примечание – бит RE не должен сбрасываться при приеме данных. Если бит RE запрещен во время приема, прием текущего байта будет прерван.

### 37.3.3.3 Ошибка переполнения (*overflow*)

Ошибка переполнения возникает, когда получен символ, но бит RXNE не сброшен. Данные не могут быть переданы из регистра сдвига в регистр RDR, пока бит RXNE не будет сброшен.

Флаг RXNE устанавливается после каждого полученного байта. Ошибка переполнения возникает, если флаг RXNE установлен, когда приняты следующие данные. При появлении ошибки переполнения:

- бит ORE будет установлен;
- содержимое регистра RDR не будет потеряно. Предыдущие данные будут доступны при чтении регистра USART\_DR;
- регистр сдвига будет перезаписан. В этот момент любые данные полученные во время ошибки переполнения будут потеряны;
- генерируется прерывание, если либо установлены биты RXNEIE или EIE;
- бит ORE сбрасывается чтением регистра USART\_SR, за которым следует операция чтения регистра USART\_DR.

Примечание – установленный бит ORE показывает, что как минимум один элемент данных потерян. Существует два варианта:

- если  $RXNE=1$ , то последние корректные данные сохраняются в регистр RDR и могут быть прочитаны,
- если  $RXNE=0$ , это означает, что последние корректные данные уже были прочитаны, поэтому нечего считывать из регистра RDR. Это может произойти, когда последние корректные данные были прочитаны из регистра RDR одновременно с получением новых данных. Также это может произойти, когда новые данные приняты во время последовательности чтения (между доступом на чтение регистра USART\_SR и доступом на чтение регистра USART\_DR).

### 37.3.3.4 Выбор правильного метода передискретизации

Приемник использует разные техники передискретизации, конфигурируемые пользователем (кроме синхронного режима), для восстановления данных путем отделения друг от друга входящих данных и шума.

Метод передискретизации можно выбрать путем программирования бита OVER8 в регистре USART\_CR1, передискретизация может быть или 16- или 8-кратная тактам скорости (Рисунки 118 и 119).

В зависимости от применения:

- Выберите передискретизацию на 8 ( $OVER8=1$ ) для достижения увеличенной скорости (до  $fPCLK/8$ ). В данном случае максимальный допуск на отклонение тактов для приемника снижается (см. Допуск отклонения тактовой частоты USART приемника);
- Выберите передискретизацию на 16 ( $OVER8=0$ ) для увеличения допуска приемника к отклонению тактов. В данном случае максимальная скорость ограничена  $fPCLK/16$ .



Программирование бита ONEBIT в регистре USART\_CR3 выбирает метод, используемый для оценки логического уровня. Есть две опции:

- мажоритарная выборка из 3 выборок по центру принятого бита. В данном случае, если эти 3 выборки не одинаковые, установится бит NF;
- одна выборка в центре принятого бита.

В зависимости от применения:

- выберите метод мажоритарности из 3-х выборок (ONEBIT=0) при работе в условиях шума и отклоните данные, если обнаружен шум, потому что это показывает, что во время выборки произошел сбой.
- выберите метод одиночной выборки (ONEBIT=1), если нет шумов на линии, чтобы увеличить допуск на отклонение тактов приемника (см. Допуск на отклонения тактовой частоты USART приемника). В данном случае бит NF не будет установлен.

Когда во фрейме обнаружен шум:

- бит NF будет установлен по переднему фронту бита RXNE;
- некорректные данные будут перемещены из регистра сдвига в регистр USART\_DR;
- не будет генерироваться прерывание в случае однобайтной передачи данных. Однако этот бит будет установлен одновременно с битом RXNE, который сам генерирует прерывание. В случае мультибуферного обмена данными прерывание возникнет, если установлен бит EIE в регистре USART\_CR3.

Бит NF сбрасывается чтением регистра USART\_SR, после чего следует чтение регистра USART\_DR.

Примечание – передискретизация на 8 недоступна в режимах Smartcard. В данном режиме бит OVER8 аппаратно сбрасывается в 0.

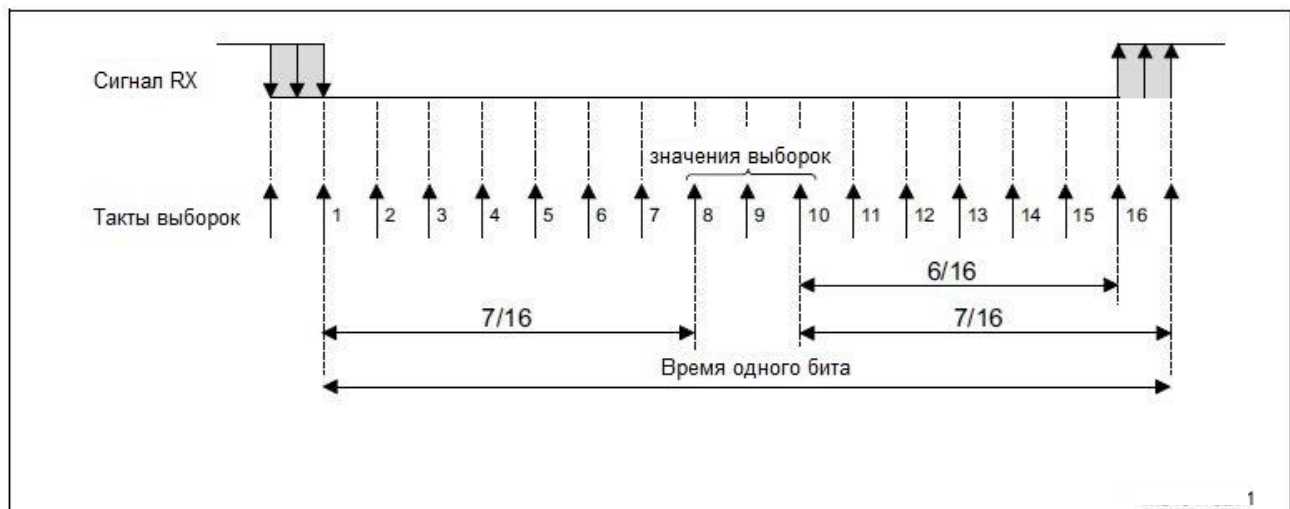


Рисунок 118 – Выборка данных при передискретизации на 16

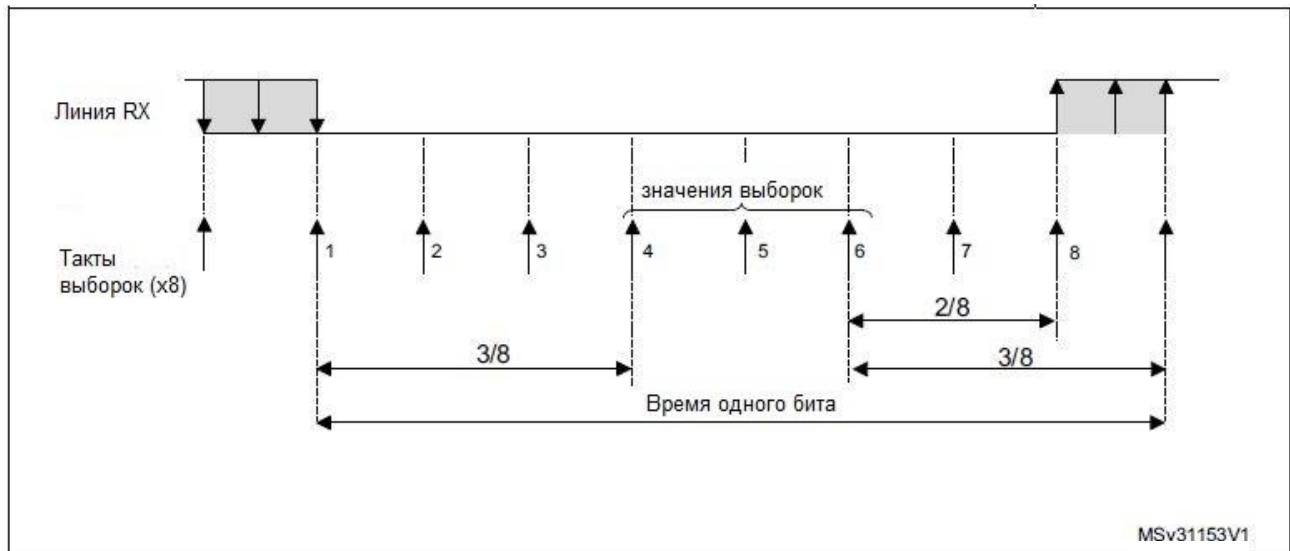


Рисунок 119 – Выборка данных при передискретизации на 8

Таблица 557 – Обнаружение шумов для считанных данных

Значение	Статус NF	Значение принятого бита
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

### 37.3.3.5 Ошибка фрейма (framing error)

Ошибка фрейма обнаруживается, когда стоп-бит не распознан во время приема данных в ожидаемый промежуток времени из-за рассинхронизации или из-за чрезмерного шума;

Когда обнаружена ошибка фрейма:

- бит FE устанавливается аппаратно;
- некорректные данные передаются из регистра сдвига в регистр USART\_DR;
- в случае однобайтного обмена никакое прерывание не генерируется. Однако данный бит устанавливается одновременно с битом RXNE, который сам генерирует прерывание. В случае мультибуферного обмена прерывание генерируется, если установлен бит EIE в регистре USART\_CR3.

Бит FE сбрасывается чтением регистра USART\_SR, после чего следует чтение регистра USART\_DR.

### 37.3.3.6 Конфигурируемые стоп-биты во время приема

Количество принимаемых стоп-бит можно конфигурировать в Регистре Управления 2. Это может быть 2 бита в нормальном режиме и 0,5 или 1,5 в режиме Smartcard.

1 **0,5 стоп-бит (прием в режиме Smartcard)**: для данного режиме выборки не делаются. Как следствие, при выборе 0,5 стоп-бита ошибка фрейма или сигнал break не могут быть обнаружены.

3 **1,5 стоп-бита (режим Smartcard)**: при передаче данных в режиме smartcard, прибор должен проверить корректность отправляемых данных. Таким образом, блок приемника должен быть разрешен (RE=1 в регистре USART\_CR1), а стоп-бит проверяется на наличие ошибки четности. В случае наличия ошибки четности, смарт-карта подтягивает сигнал данных в логический 0 во время выборки (сигнал NACK), который опознается как ошибка фрейма. После чего флаг FE устанавливается с RXNE по окончании 1,5 стоп-бита. Анализ уровня для 1.5 стоп-бита осуществляется на 16, 17 и 18 выборках (1 период скорости после начала стоп-бита). 1,5 стоп-бит может быть разложен на 2 части: одна часть из 0,5 периода скорости, во время которой ничего не происходит, и вторая часть – 1 период нормального стоп-бита, во время которого происходит анализ сигнала. См. Режим Smartcard.

4 **2 стоп-бита**: Анализ для 2 стоп-бит осуществляется на 8, 9 и 10 выборках первого стоп-бита. Если была обнаружена ошибка фрейма во время первого стоп-бита, устанавливается флаг ошибки. Второй стоп-бит не проверяется на наличие ошибки фрейма. Флаг RXNE устанавливается по окончании первого стоп-бита.

### 37.3.4 Дробный генератор скорости

Скорость обмена для приемника и передатчика (Rx и Tx) устанавливается в одинаковое значение, программируемое коэффициентами Mantissa (целая часть) и Fraction (дробная часть) делителя USARTDIV.

Скорость для стандартного USART (включая режим SPI) вычисляется по формуле

$$Tx/Rx \text{ baud} = \frac{f_{СК}}{8 \cdot (2 - OVER8) \cdot USARTDIV} \quad (29)$$

Скорость для режимов Smartcard вычисляется по формуле

$$Tx/Rx \text{ baud} = \frac{f_{СК}}{16 \times USARTDIV} \quad (30)$$

где USARTDIV – это число с фиксированной запятой без знака, запрограммированное в регистре USART\_BRR.

– Если OVER8=0, дробная часть кодируется 4 битами и программируется битами DIV\_fraction [3:0] в регистре USART\_BRR

– Если OVER8=1, дробная часть кодируется 3 битами и программируется битами DIV\_fraction [2:0] в регистре USART\_BRR, бит DIV\_fraction [3] должен быть сброшен.

Примечание – Счетчики скорости обновляются в регистрах скорости после операции записи в регистр USART\_BRR. Следовательно, значение регистра скорости не должно изменяться во время активного обмена данными.

#### 37.3.4.1 *Примеры расчета USARTDIV для значений регистра USART\_BRR при OVER8=0*

*Пример 1:*

Если DIV\_Mantissa = 0d27 и DIV\_Fraction = 0d12 (USART\_BRR = 0x1BC), тогда Mantissa (USARTDIV) = 0d27

Fraction (USARTDIV) = 12/16 = 0d0.75

Следовательно, USARTDIV = 0d27.75

*Пример 2:*

Для программирования USARTDIV = 0d25.62

получается:

DIV\_Fraction = 16 • 0d0.62 = 0d9.92

Ближайшее действительное число 0d10 = 0xA

DIV\_Mantissa = mantissa (0d25.620) = 0d25 = 0x19

Тогда, USART\_BRR = 0x19A, следовательно, USARTDIV = 0d25.625

*Пример 3:*

Для программирования USARTDIV = 0d50.99

получается:

DIV\_Fraction = 16 • 0d0.99 = 0d15.84

Ближайшее действительное число 0d16 = 0x10 => переполнение DIV\_frac[3:0] => должен быть добавлен перенос (carry) к мантиссе

DIV\_Mantissa = mantissa (0d50.990 + carry) = 0d51 = 0x33

Тогда, USART\_BRR = 0x330, следовательно, USARTDIV = 0d51.000

#### 37.3.4.2 *Примеры расчета USARTDIV для значений регистра USART\_BRR при OVER8=1*

*Пример 1:*

Если DIV\_Mantissa = 0x27 и DIV\_Fraction[2:0]= 0d6 (USART\_BRR = 0x1B6), тогда Mantissa (USARTDIV) = 0d27

Fraction (USARTDIV) = 6/8 = 0d0.75

Therefore USARTDIV = 0d27.75

*Пример 2:*

Для программирования USARTDIV = 0d25.62

получается:

DIV\_Fraction = 8 • 0d0.62 = 0d4.96

Ближайшее действительное число 0d5 = 0x5

DIV\_Mantissa = mantissa (0d25.620) = 0d25 = 0x19

Тогда, USART\_BRR = 0x195 => USARTDIV = 0d25.625

Пример 3:

Для программирования USARTDIV = 0d50.99

Получается:

$$DIV\_Fraction = 8 \cdot 0d0.99 = 0d7.92$$

Ближайшее действительное число 0d8 = 0x8 => переполнение DIV\_frac[2:0] => должен быть добавлен перенос (carry) к мантиссе

$$DIV\_Mantissa = mantissa (0d50.990 + carry) = 0d51 = 0x33$$

$$\text{Тогда, USART\_BRR} = 0x0330 \Rightarrow \text{USARTDIV} = 0d51.000$$

Таблица 558 – Определение погрешности для запрограммированных скоростей передачи при  $f_{CLK} = 60 \text{ МГц}^*$

№	Требуемая скорость	Передискретизация на 16 (OVER8=0)			Передискретизация на 8 (OVER8=1)		
		Реальная скорость	Значение в регистре скорости передачи USART_BRR	Ошибка, %	Реальная скорость	Значение в регистре скорости передачи USART_BRR	Ошибка, %
1	2,4 Кбит/с	2,4 Кбит/с	1562,5	0	2,4 Кбит/с	3125	0
2	9,6 Кбит/с	9,6 Кбит/с	390,625	0	9,6 Кбит/с	781,25	0
3	19,2 Кбит/с	19,2 Кбит/с	195,3125	0	19,2 Кбит/с	390,625	0
4	57,6 Кбит/с	57,582 Кбит/с	65,125	0,03	57,582 Кбит/с	130,25	0,03
5	115,2 Кбит/с	115,163 Кбит/с	32,5625	0,03	115,163 Кбит/с	65,125	0,03
6	230,4 Кбит/с	230,769 Кбит/с	16,25	0,16	230,769 Кбит/с	32,5	0,16
7	460,8 Кбит/с	461,538 Кбит/с	8,125	0,16	461,538 Кбит/с	16,25	0,16
8	896 Кбит/с	895,522 Кбит/с	4,1875	0,05	895,522 Кбит/с	8,375	0,05
9	921,6 Кбит/с	923,077 Кбит/с	4,0625	0,16	923,077 Кбит/с	8,125	0,16
10	2 Мбит/с	2 Мбит/с	1,875	0	2 Мбит/с	3,75	0
11	3 Мбит/с	3 Мбит/с	1,25	0	3 Мбит/с	2,5	0
12	4 Мбит/с	NA	NA	NA	4 Мбит/с	1,875	0

№	Требуемая скорость	Передискретизация на 16 (OVER8=0)			Передискретизация на 8 (OVER8=1)		
		Реальная скорость	Значение в регистре скорости передачи USART_BRR	Ошибка, %	Реальная скорость	Значение в регистре скорости передачи USART_BRR	Ошибка, %
13	5 Мбит/с	NA	NA	NA	5 Мбит/с	1,5	0
14	6 Мбит/с	NA	NA	NA	6 Мбит/с	1,25	0

\* Чем ниже тактовая частота центрального процессора, тем меньше точность для конкретного значения скорости передачи. Верхний предел достижимой скорости передачи может быть зафиксирован с этими данными

### 37.3.5 Допуск ухода тактовой частоты для приемника USART

Асинхронный приемник USART корректно работает, только если общее отклонение тактовой частоты меньше, чем допуск приемника USART. Причины, которые способствуют общему отклонению:

- DTRA: отклонение из-за ошибки передатчика (что также включает отклонение от локального тактового генератора передатчика);
- DQUANT: ошибка квантования скорости приемника;
- DREC: отклонение локального генератора тактов приемника;
- DTCL: отклонение из-за линии передачи (обычно из-за того, что приемопередатчики могут давать асимметричные перепады от 0 к 1 по сравнению с перепадами от 1 к 0).

$$DTRA + DQUANT + DREC + DTCL < \text{допуск приемника USART}$$

Допуск приемника USART для правильного приема данных равен максимально допустимому отклонению и зависит от следующих параметров:

- длина символа 10 или 11 бит, что определяется битом M в регистре USART\_CR1
- передискретизация на 8 или 16, что определяется битом OVER8 в регистре USART\_CR1
- используется или нет дробная установка скорости
- используется 1 или 3 бита для оцифровки данных, в зависимости от бита ONEBIT в регистре USART\_CR3.

Таблица 559 – Допуск приемника USART при DIV\_fraction = 0

Бит M	OVER8 = 0		OVER8 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3,75%	4,375%	2,50%	3,75%
1	3,41%	3,97%	2,27%	3,41%

Таблица 560 – Допуск приемника USART при DIV\_Fraction, не равным 0

Бит М	OVER8 = 0		OVER8 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3,33%	3,88%	2%	3%
1	3,03%	3,53%	1,82%	2,73%

### 37.3.6 Многопроцессорный обмен

Существует возможность многопроцессорного обмена данными через USART (несколько приемопередатчиков USART, объединенных в одну сеть). Например, один USART может быть ведущим устройством, его выход TX подключается ко входу RX другого приемопередатчика USART. Другие приемопередатчики являются ведомыми, их соответствующие выходы TX логически объединены операцией логическое «И» и подсоединены ко входу RX ведущего устройства.

В многопроцессорных конфигурациях предпочтительно, чтобы только один получатель сообщения активно принимал все сообщение целиком, чтобы уменьшить обработку избыточных данных для приемников, которым эти данные не адресованы.

Неадресованные устройства могут быть переведены в режим «молчания» (mute mode) при помощи функции приостановки. В данном режиме:

- никакой статус бит приема не может быть установлен;
- все прерывания приема запрещены;

– бит RWU в регистре USART\_CR1 установлен в 1. Бит RWU может управляться автоматически аппаратно или может быть записан программой при определенных условиях.

USART может войти в режим молчания или выйти из него при помощи одного из двух методов, в зависимости от значения бита WAKE в регистре USART\_CR1:

- обнаружение линии Idle, если бит WAKE сброшен;
- обнаружение сигнала метки адреса (Address Mark), если бит WAKE установлен.

#### 37.3.6.1 Обнаружение линии Idle (WAKE=0)

USART переключится в режим молчания, если бит RWU установлен в «1».

Приемопередатчик выйдет из режима молчания, когда будет обнаружен фрейм Idle. После чего бит RWU сбрасывается аппаратной частью, но бит IDLE в регистре USART\_SR не устанавливается. Бит RWU может быть записан в 0 программно.

На рисунке 120 показан пример поведения устройства в режиме молчания с использованием обнаружения линии Idle.

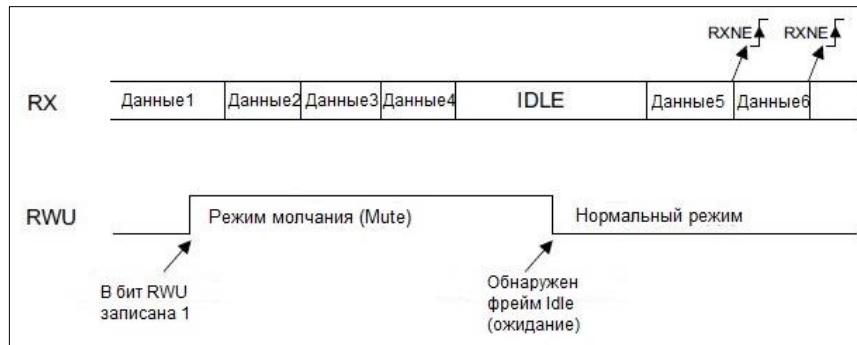


Рисунок 120 – Режим молчания (Mute) с обнаружением линии Idle

### 37.3.6.2 Обнаружение метки адреса (Address mark) (WAKE=1)

В данном режиме, байты распознаются как адреса, если MSB равен 1, в другом случае они распознаются как данные. В байте адреса адрес приемника помещается в 4 LSB. Приемник сравнивает это четырехбитное слово с собственным адресом, запрограммированным в битах ADD в регистре USART\_CR2.

USART переключится в режим молчания, когда будет получен символ адреса, который не соответствует запрограммированному режиму. В данном случае, бит RWU устанавливается аппаратно. Флаг RXNE не установится для данного байта адреса, и не будет сгенерировано прерывание, так как приемопередатчик перейдет в режим молчания.

Приемопередатчик выйдет из режима молчания, когда будет принят символ адреса, совпадающий с запрограммированным адресом. После чего бит RWU будет сброшен и последующие биты будут приняты. Символ адреса будет установлен в бите RXNE, так как бит RWU был сброшен.

Бит RWU может быть записан в 0 или 1, когда буфер приемника не содержит данных (RXNE=0 в регистре USART\_SR). Иначе попытка записи будет проигнорирована.

Пример поведения приемопередатчика в режиме молчания с использованием режима обнаружения метки адреса приведен на рисунке 121.

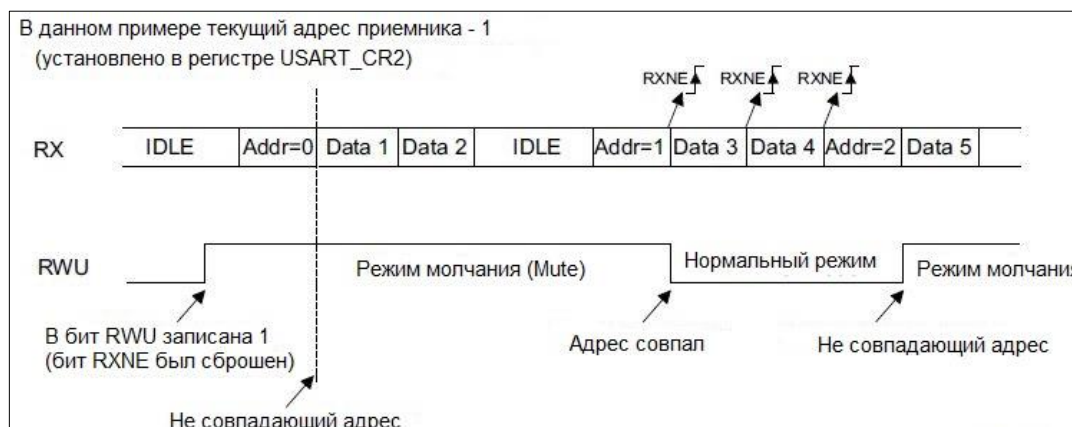


Рисунок 121 – Режим молчания (Mute) с обнаружением метки адреса



### 37.3.7 Контроль четности

Контроль четности бит (генерация бита четности при передаче и контроль бита четности при приеме) разрешается установкой бита PCE в регистре USART\_CR1. В зависимости от длины фрейма, определяемой битом M, возможные форматы фрейма приемопередатчика USART приведены в таблице 561.

Таблица 561 – Форматы фрейма

Бит M	Бит PCE	Фрейм USART*
0	0	SB   8 бит данных   STB
0	1	SB   7 бит данных   PB   STB
1	0	SB   9 бит данных   STB
1	1	SB   8 бит данных   PB   STB
* SB: стартовый бит, STB: стоп-бит, PB: бит четности		

#### 37.3.7.1 Проверка на четность

Бит четности вычисляется так, чтобы получить четную сумму всех «1» во фрейме из 7 или 8 бит (в зависимости от значения бита M) и бита четности.

Например, data=00110101; установлено 4 бита => бит четности = 0, если выбрана проверка на четность (бит PS в USART\_CR1 = 0).

#### 37.3.7.2 Проверка на нечетность

Бит четности вычисляется так, чтобы получить нечетную сумму всех «1» во фрейме из 7 или 8 бит (в зависимости от значения бита M) и бита четности.

Например, data=00110101; установлено 4 бита => бит четности = 1, если выбрана проверка на нечетность (бит PS в USART\_CR1 = 1).

#### 37.3.7.3 Проверка четности при приеме

Если проверка четности показала ошибку данных, устанавливается флаг PE в регистре USART\_SR, и генерируется прерывание, если в регистре USART\_CR1 установлен бит PEIE. Флаг PE сбрасывается программной последовательностью (чтение регистра статуса, за которым следует чтение или запись регистра данных USART\_DR).

Примечание – в случае пробуждения по метке адреса: MSB используется для идентификации адреса, а не бит четности. Приемник не проверяет бит четности данных адреса (бит PE не устанавливается в случае ошибки четности).

#### 37.3.7.4 Генерация бита четности при передаче

Если в регистре USART\_CR1 установлен бит PCE, то MSB данных, записанных в регистре данных, будет передан, но он будет изменен битом четности (четное количество «1», если выбрана проверка на четность (PS=0) или нечетное количество «1», если выбрана проверка на нечетность (PS=1)).

Примечание – Программная часть, которая контролирует передачу, может активировать программную последовательность для сброса флага PE (чтение регистра статуса, за которым следует доступ на чтение или запись регистра данных). При работе в полудуплексном режиме, в зависимости от программной части, это может привести к неожиданному сбросу флага PE.

### 37.3.8 Синхронный режим USART

Синхронный режим выбирается записью бита CLKEN в «1» в регистре USART\_CR2.

В синхронном режиме следующие биты должны быть сброшены:

- SCEN и HDSEL в регистре USART\_CR3.

USART дает возможность пользователю управлять двунаправленным синхронным последовательным обменом данных в режиме ведущего приемопередатчика. Вывод СК является выходом тактов передатчика USART. Во время стартового и стоп-битов никакие тактовые импульсы не посылаются на вывод СК. В зависимости от состояния бита LBCL в регистре USART\_CR2 тактовые импульсы будут или не будут генерироваться во время последнего достоверного бита данных (маркер адреса). Бит CPOL в регистре USART\_CR2 позволяет пользователю выбрать полярность тактов, а бит CPHA в регистре USART\_CR2 позволяет выбрать фазу внешних тактов (см. рисунки 122, 123 и 124)

Во время состояния ожидания (Idle), преамбулы и отправки символа break, внешний тактовый сигнал СК не активируется.

В синхронном режиме, передатчик работает точно также, как и в асинхронном режиме. Но так как сигнал СК синхронизируется с сигналом TX (в зависимости от CPOL и CPHA), данные на TX синхронные.

Приемник в данном режиме работает не так, как в асинхронном режиме. Если RE=1, данные тактируются синхронно с сигналом СК (передний и задний фронт, в зависимости от значений CPOL и CPHA), без какой-либо передискретизации. Время установки и удержания сигнала должно соблюдаться (зависит от скорости обмена: 1/16 от времени бита).

Примечание – Вывод СК работает совместно с выводом TX. Таким образом, тактирование обеспечивается только, если разрешен передатчик (TE=1) и передаются данные (регистр данных USART\_DR записан). Это означает, что невозможно принимать синхронные данные без передачи данных.

Должны быть выбраны биты LBCL, CPOL и CPHA, когда передатчик и приемник отключены (TE=RE=0) для обеспечения корректной работы тактовых импульсов. Значения данных бит не должны изменяться во время включения передатчика или приемника.

Рекомендуется устанавливать биты TE и RE одной командой с целью минимизации времени установки и удержания приемника.

Приемопередатчик USART поддерживает только режим ведущего устройства: он не может принимать или передавать данные, связанные со входом тактирования (СК всегда работает как выход).

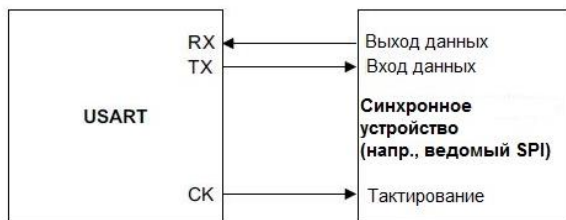


Рисунок 122 – Пример синхронной передачи USART

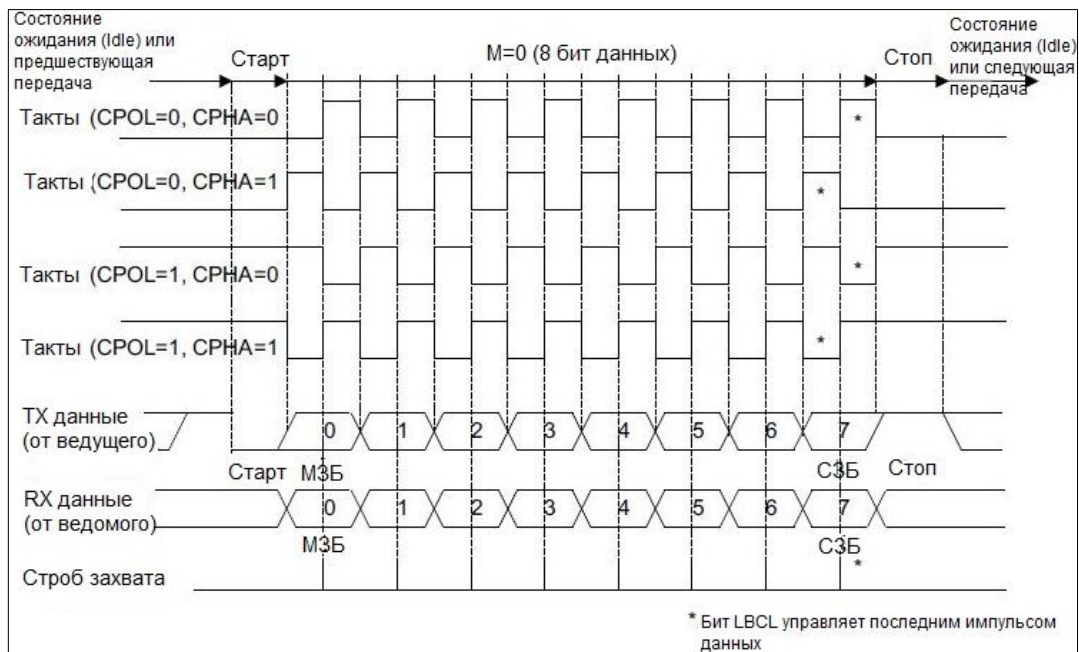


Рисунок 123 – Диаграмма тактирования данных USART (M=0)

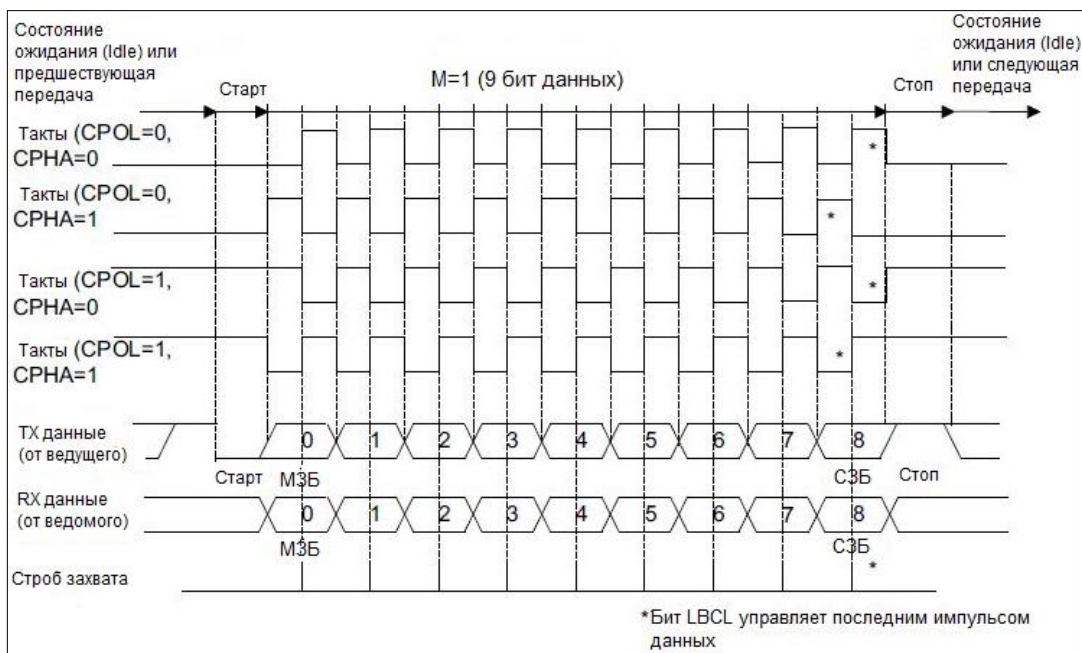


Рисунок 124 – Диаграмма тактирования данных USART (M=1)

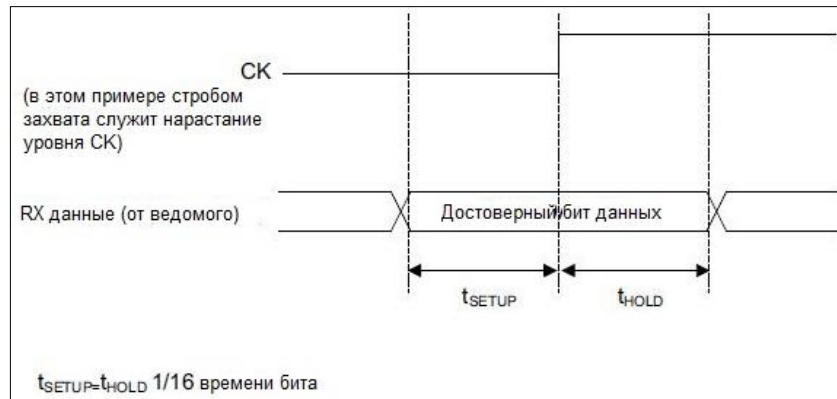


Рисунок 125 – Время установки/удержания RX

Примечание – Функция тактирования СК отличается в режиме Smartcard (См. далее описание режима Smartcard).

### 37.3.9 Однопроводной полудуплексный обмен данными

Однопроводной полудуплексный режим обмена данными (single-wire half-duplex mode) выбирается путем установки бита HDSEL в регистре USART\_CR3. В данном режиме должны быть сброшены следующие биты:

- SCEN в регистре USART\_CR3.

USART может быть сконфигурирован для соответствия однопроводному полудуплексному протоколу, где линии сигналов TX и RX имеют внутреннее соединение. Выбор между полудуплексным или полнодуплексным режимом передачи данных осуществляется управляющим битом 'HALF DUPLEX SEL' (HDSEL в регистре USART\_CR3).

Как только бит HDSEL устанавливается в «1»:

- линии сигналов TX и RX соединяются внутри;
- вывод RX больше не используется;
- вывод TX всегда свободен, если данные не передаются. Таким образом, он используется как стандартный ввод/вывод в режиме ожидания или при приеме данных. Это означает, что данный ввод/вывод должен быть сконфигурирован так, чтобы он был высокоомным входом (или выходом с открытым стоком), когда он не управляется USART.

Кроме этого, обмен данными происходит также, как и обычном режиме передачи данных USART.

Конфликты на линии связи должны контролироваться программно (например, с использованием централизованного арбитра шины). В частности, передача данных никогда не блокируется аппаратной частью и будет продолжаться, как только данные будут записаны в регистр данных, пока установлен бит TE.

### 37.3.10 Режим Smartcard

Режим смарт-карты (Smartcard mode) выбирается путем установки бита SCEN в регистре USART\_CR3. В режиме Smartcard следующие биты должны быть сброшены:

- бит HDSEL в регистре USART\_CR3.

Более того, бит CLKEN может быть установлен с целью тактирования смарт-карты.

Интерфейс режима Smartcard разработан для поддержки асинхронного протокола в соответствии со стандартом ISO 7816-3. USART должен быть сконфигурирован следующим образом:

- 8 бит + бит четности: где M=1 и PCE=1 в регистре USART\_CR1
- 1,5 стоп-бита при передаче и приеме: где STOP=11 в регистре USART\_CR2.

Примечание – Также можно выбрать 0,5 стоп-бит для приема, но рекомендуется использовать 1,5 стоп-бита для передачи и приема данных, чтобы избежать переключения между двумя конфигурациями.

Рисунок 126 показывает пример, как выглядит сигнал данных с ошибкой четности и без ошибки четности.

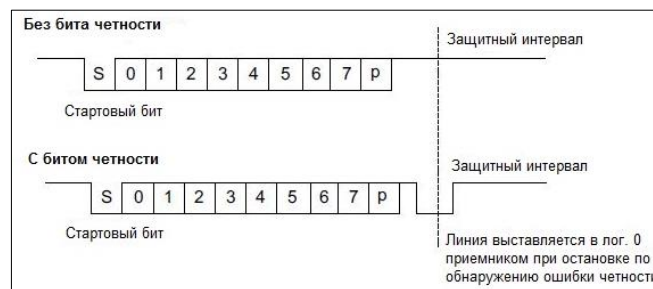


Рисунок 126 – Асинхронный протокол ISO 7816-3

При подключении к смарт-карте, выход TX приемопередатчика USART управляет двунаправленной линией, которая также управляется и смарт-картой. Вывод TX должен быть сконфигурирован как открытый сток.

Smartcard – это однопроводный полудуплексный протокол передачи данных.

– Передача данных из регистра сдвига передатчика осуществляется с гарантированной задержкой минимум  $\frac{1}{2}$  такта скорости. При нормальном режиме работы полная передача из регистра сдвига начнется сдвигом на следующем перепаде тактов скорости. В режиме Smartcard эта передача дополнительно задерживается на  $\frac{1}{2}$  такта скорости.

– Если обнаружена ошибка четности при приеме фрейма, запрограммированного на 0,5 или 1,5 стоп-бита, линия передачи подтягивается в 0 на период такта скорости после завершения приема фрейма. Это показывает смарт-карте, что данные, переданные на приемопередатчик USART, не были корректно приняты. Данный сигнал NACK (подтягивание линии передачи в 0 на период 1 такта скорости) сгенерирует ошибку фрейма на стороне передатчика (конфигурация 1,5 стоп-бита). Приложение должно обработать эту ситуацию повторной отправкой данных в соответствии с протоколом.

Ошибка четности не подтверждается приемником (сигнал NACK), если установлен управляющий бит NACK, иначе сигнал NACK не передается.

– Установка флага ТС может быть задержана путем программирования регистра защитного интервала времени (Guard Time). При нормальной работе флаг ТС устанавливается, когда регистр сдвига передатчика пуст и нет никаких запросов на передачу. В режиме смарт-карты пустой регистр сдвига передатчика запускает счетчик защитного интервала для счета запрограммированного значения в регистре защитного интервала времени (Guard Time register). Флаг ТС подтягивается в 0. Когда счетчик достигает запрограммированного значения, устанавливается флаг ТС.

– На снятие флага ТС режим смарт-карты не влияет.

– Если обнаружена ошибка фрейма на стороне передатчика (ответ NACK от приемника), сигнал NACK не будет обнаружен как стартовый бит блоком приема передатчика. В соответствии со стандартом ISO, длительность принятого сигнала NACK может составлять 1 или 2 периода тактов скорости.

– На стороне приемника, если была обнаружена ошибка четности, и был передан сигнал NACK, приемник не обрабатывает сигнал NACK как стартовый бит.

Примечание – Символ break не имеет значения в режиме Smartcard. Данные 0x00 с ошибкой фрейма будут обработаны как данные, а не как символ break.

Фрейм Idle не передается при переключении бита TE. Фрейм Idle (как определено для других конфигураций) не определен в протоколе ISO.

Рисунок 127 показывает, как обрабатывается сигнал NACK. В данном примере USART передает данные и сконфигурирован на 1,5 стоп-бита. Блок приемника USART разрешен для проверки целостности данных и сигнала NACK.

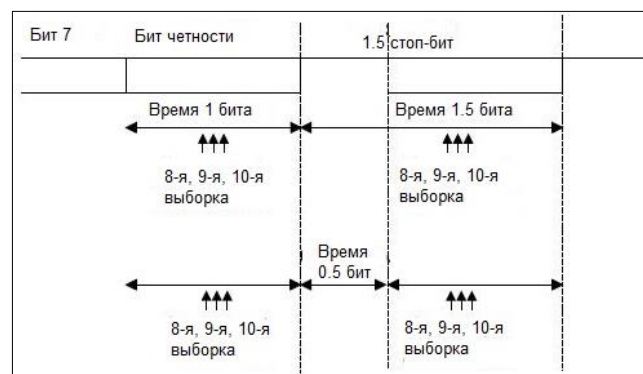


Рисунок 127 – Обнаружение ошибки четности с использованием 1,5 стоп-бит

USART может тактировать смарт-карту при помощи выхода СК. В режиме смарт-карты, выход СК не связан с обменом данными, он просто генерирует тактовую частоту, полученную из внутреннего входа тактирования периферии при помощи 5-битного делителя. Коэффициент деления конфигурируется в регистре делителя USART\_GTPR. Частота СК программируется от  $f_{СК}/2$  до  $f_{СК}/62$ , где  $f_{СК}$  – это частота тактов входа периферии.



### 37.3.11 Аппаратное управление потоком

Можно управлять потоком последовательных данных между двумя устройствами, используя вход CTS и выход RTS. Рисунок 128 показывает схему соединения двух устройств для данного режима:

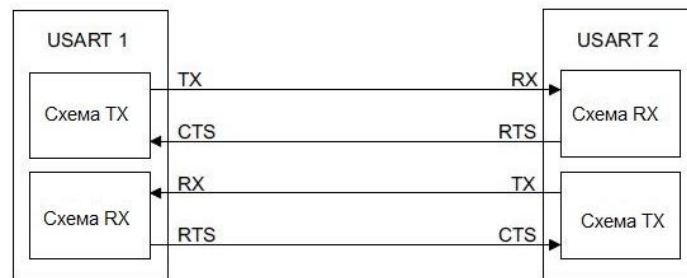


Рисунок 128 – Аппаратное управление потоком данных между двумя устройствами USART

Управление потоком RTS и CTS можно разрешить независимо записью битов RTSE и CTSE соответственно в «1» (в регистре USART\_CR3).

#### 37.3.11.1 Управление потоком RTS

Если разрешено управление потоком RTS (RTSE=1), выставляется сигнал RTS (лог. 0), когда приемник USART готов принимать новые данные. Если регистр приема заполнен, то сигнал RTS снимается, тем самым показывая, что ожидается остановка передачи в конце текущего фрейма.

Рисунок 129 показывает пример обмена с разрешенным управлением потоком RTS.

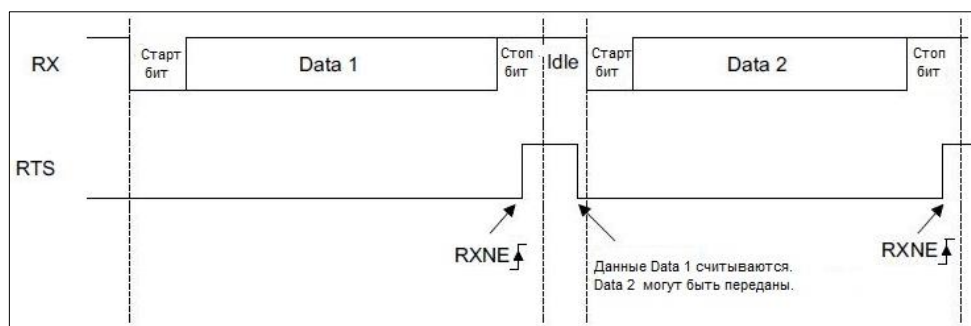


Рисунок 129 – Управление потоком RTS

#### 37.3.11.2 Управление потоком CTS

Если управление потоком CTS разрешено (CTSE=1), передатчик проверяет вход CTS перед передачей следующего фрейма. Если устанавливается сигнал CTS (лог. 0), тогда начинается передача следующих данных (подразумевается, что есть данные для передачи, т.е. другими словами TXE=0), в ином случае передача не произойдет. Когда снимается сигнал CTS во время передачи, текущая передача завершится перед остановкой передатчика.

Если CTSE=1, статус CTSIF будет установлен автоматически аппаратурой, как только вход CTS переключится. Это показывает, когда приемник будет готов к обмену

данными. Прерывание генерируется, если установлен бит CTSIE в регистре USART\_CR3. Рисунок 130 показывает пример обмена с разрешенным управлением потоком CTS.

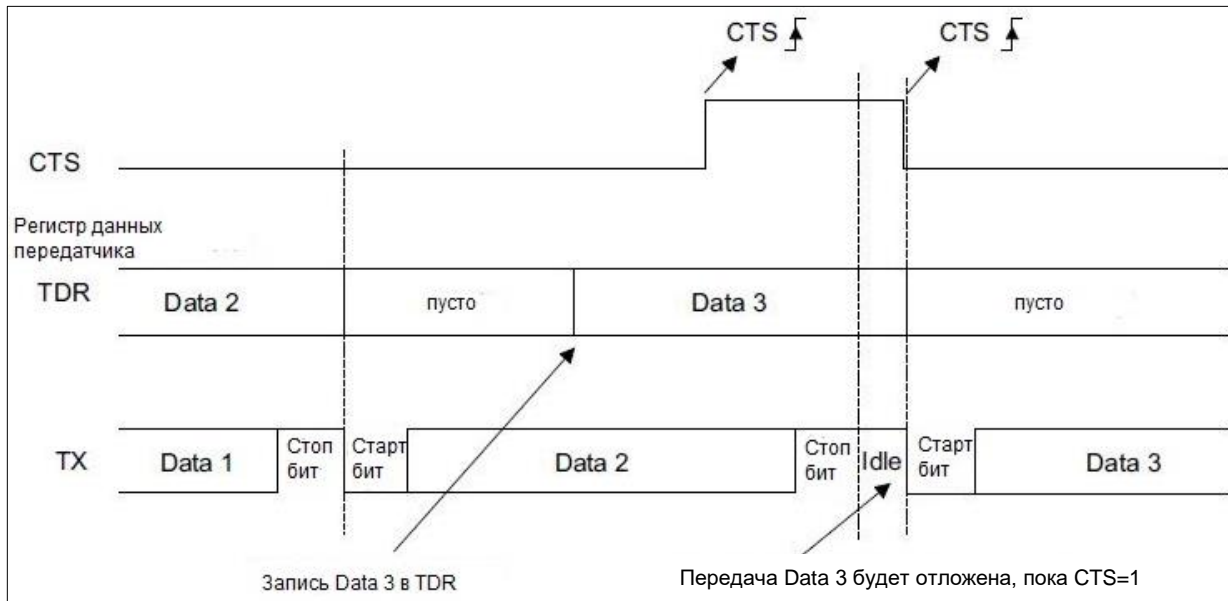


Рисунок 130 – Управление потоком CTS

Примечание – Особое поведение фреймов break: при разрешенном управлении потоком CTS, передатчик не проверяет статус входа CTS для отправки сигнала break

### 37.4 Прерывания USART

Таблица 562 – Запрос на прерывание USART

Событие	Флаг события	Бит разрешения
Регистр данных передачи пуст	TXE	TXEIE
Флаг CTS	CTS	CTSIE
Передача завершена	TC	TCIE
Принятые данные готовы для чтения	RXNE	RXNEIE
Обнаружена ошибка переполнения	ORE	
Обнаружена линия Idle	IDLE	IDLEIE
Ошибка четности	PE	PEIE
Флаг шума, ошибки переполнения или ошибки фрейма в случае мультибуферного обмена	NF или ORE или FE	EIE

События прерываний подсоединены к одному и тому же вектору прерывания (см. Рисунок 131).

– Во время передачи: Передача завершена (Transmission Complete), Линия пуста для начала передачи (Clear to Send) или Регистр данных передачи пуст (Transmit Data Register Empty).

– Во время приема: Обнаружение линии Idle, ошибка переполнения, Регистр данных приема не пуст (Receive Data register not empty), Ошибка четности, флаг шума (только при мультибуферном обмене) и ошибка фрейма (только при мультибуферном обмене).



Все эти события генерируют прерывания, если соответствующий бит разрешения прерывания установлен.

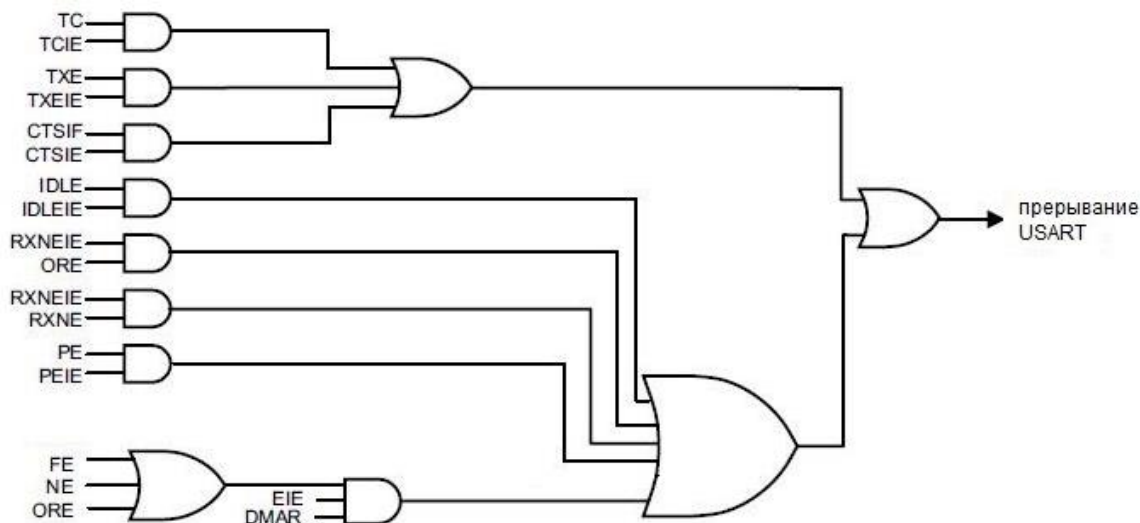


Рисунок 131 – Диаграмма отображений событий USART на прерывание

### 37.5 Регистры USART

Доступ к регистрам периферии осуществляется по 16-битным полусловам или 32-битными словами. Таблицы 564 – 576 показывают карту памяти регистров USART и их значения после сброса.

Таблица 563 – Регистры блока USART\_CNTR

Базовый Адрес		Название	Описание
0x400C_0000		USART_CNTR	Блок контроллера USART (ISO7816)
Смещение			
0x0000	0	USART_SR	Регистр статуса
0x0004	1	USART_DR	Регистр данных
0x0008	2	USART_BRR	Регистр настройки скорости передачи данных
0x000c	3	USART_CR1	Регистр управления 1
0x0010	4	USART_CR2	Регистр управления 2
0x0014	5	USART_CR3	Регистр управления 3
0x0018	6	USART_GTPR	Регистр защищенного интервала предделителя

#### 37.5.1 USART\_SR

Таблица 564 – Регистр USART\_SR

Base ADDR=	0x400C_0000	Offset=	0x0000_0000	Reset=	0x0000_00C0										
REG Name:	USART_SR														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-						CTS	-	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						RW		R	RW	RW	R	R	R	R	R

Таблица 565 – Описание бит регистра USART\_SR

Бит	Имя	Значение	Описание
31...10	-	0	Зарезервировано
9	CTS	0	Флаг CTS. Данный бит устанавливается аппаратно, когда на входе CTS меняется уровень, если установлен бит CTSE. Бит сбрасывается программно записью 1. Прерывание генерируется, если CTSIE=1 в регистре USART_CR3. 0 – не было изменения статуса линии CTS; 1 – на линии CTS обнаружено изменение уровня
8	-	0	Зарезервировано
7	TXE	1	Регистр данных передатчика пуст. Данный бит устанавливается аппаратно, когда содержимое регистра TDR было перемещено в регистр смещения. Прерывание генерируется, если TXEIE=1 в регистре USART_CR1. Бит сбрасывается записью регистра USART_DR. 0 – данные не были переданы в регистр сдвига; 1 – данные переданы в регистр сдвига. Примечание – Данный бит используется во время передачи с использованием одного буфера
6	TC	1	Передача завершена. Данный бит устанавливается аппаратно, когда передача фрейма, содержащего данные, завершена, и бит TXE установлен. Прерывание генерируется, если TCIE=1 в регистре USART_CR1. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем запись регистра USART_DR). Бит TC может быть также сброшен записью 1. Данный сброс рекомендуется выполнять только при мультибуферном обмене. 0 – передача не завершена; 1 – передача завершена
5	RXNE	0	Регистр данных приема не пуст. Данный бит устанавливается аппаратно, когда содержимое регистра сдвига RDR было передано в регистр USART_DR. Прерывание генерируется, если RXNEIE=1 в регистре USART_CR1. Бит сбрасывается чтением регистра USART_DR. Флаг RXNE может быть также сброшен записью 1. Данная последовательность сброса рекомендована для мультибуферного обмена. 0 – данные не получены; 1 – полученные данные готовы для чтения

Бит	Имя	Значение	Описание
4	IDLE	0	<p>Обнаружение линии IDLE (ожидание).</p> <p>Данный бит устанавливается аппаратно, когда обнаружена линия Idle. Прерывание генерируется, если IDLEIE=1 в регистре USART_CR1. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем чтения регистра USART_DR).</p> <p>0 – не обнаружена линия Idle; 1 – линия Idle обнаружена.</p> <p>Примечание – Бит IDLE не будет снова установлен, пока не установится бит RXNE (т.е. пока не будет обнаружено состояние ожидания на линии)</p>
3	ORE	0	<p>Ошибка переполнения</p> <p>Данный бит устанавливается аппаратно, когда принятое в настоящий момент слово в регистре сдвига готово к передаче в регистр RDR, если RXNE=1. Прерывание генерируется, если RXNEIE=1 в регистре USART_CR1. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем чтение регистра USART_DR).</p> <p>0 – нет ошибки переполнения; 1 – обнаружена ошибка переполнения.</p> <p>Примечание – Когда установлен этот бит, содержимое регистра RDR не будет потеряно, но регистр сдвига будет перезаписан. Прерывание генерируется по флагу ORE в случае мультибуферного обмена, если установлен бит EIE</p>
2	NF	0	<p>Флаг обнаружения шума.</p> <p>Данный бит устанавливается аппаратно, когда в принятом фрейме обнаружен шум. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем чтение регистра USART_DR).</p> <p>0 – шум не обнаружен; 1 – шум обнаружен.</p> <p>Примечания</p> <p>1 Данный бит не генерирует прерывание, так как он появляется одновременно с битом RXNE, который сам по себе генерирует прерывание по флагу NF в случае мультибуферного обмена, если установлен бит EIE.</p> <p>2 Если шумов на линии нет, флаг NF может быть отключен установкой бита ONEBIT в 1 для увеличения допуска на отклонения приемопередатчика (см. Допуск ухода тактовой частоты приемника USART)</p>

Бит	Имя	Значение	Описание
1	FE	0	<p>Ошибка фрейма.</p> <p>Данный бит устанавливается аппаратно, когда произошла рассинхронизация, чрезмерный шум или был получен символ break. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем чтение регистра USART_DR).</p> <p>0 – ошибки фрейма не обнаружено 1 – обнаружена ошибка фрейма или символ break</p> <p>Примечания</p> <p>1 Данный бит не генерирует прерывание, так как он появляется одновременно с битом RXNE, который сам по себе генерирует прерывание по флагу NF в случае мультибуферного обмена, если установлен бит EIE.</p> <p>2 Если шумов на линии нет, флаг NF может быть отключен установкой бита ONEBIT в 1 для увеличения допуска на отклонения приемопередатчика (см. пункт 37.3.5 «Допуск ухода тактовой частоты для приемника USART»)</p>
0	PE	0	<p>Ошибка четности.</p> <p>Данный бит устанавливается аппаратно, когда в режиме приема обнаружена ошибка четности. Бит сбрасывается программной последовательностью (чтение регистра статуса, затем чтение или запись регистра данных USART_DR). Программа должна дождаться, пока установится флаг RXNE, перед тем как сбрасывать бит PE.</p> <p>Прерывание генерируется, если PEIE = 1 в регистре USART_CR1. Бит PE устанавливается одновременно со всеми остальными флагами RX (RXNE, ORE, NF, FE), поэтому данный бит может сгенерировать запрос прерывания только по окончании приема.</p> <p>0 – нет ошибки четности; 1 – обнаружена ошибка четности</p>

### 37.5.2 USART\_DR

Таблица 566 – Регистр USART\_DR

Base ADDR=			0x400C_0000			Offset=		0x0000_0004			Reset=		0xFFFF_FFFF		
REG Name:			USART_DR												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9									
-							8...0								
-							DR								
-							RW								

Таблица 567 – Описание бит регистра USART\_DR

Бит	Имя	Значение	Описание
31...9	-	0	Зарезервировано
8...0	DR [8:0]	X	<p>Значение данных.</p> <p>Содержит принятый или передаваемый символ данных в зависимости от операции с регистром – чтение или запись. Регистр данных выполняет двойную функцию (чтение и запись), так как состоит из двух регистров, один для передачи данных (TDR), другой для приема данных (RDR). Регистр TDR обеспечивает параллельный интерфейс между внутренней шиной и выходным регистром сдвига. Регистр RDR обеспечивает параллельный интерфейс между входным регистром сдвига и внутренней шиной.</p> <p>Когда разрешена передача с генерацией бита четности (бит PCE установлен в 1 в регистре USART_CR1), значение, записываемое в MSB (бит 7 или бит 8 в зависимости от длины данных) не играет значения, поскольку он заменяется битов четности. Когда на приеме разрешен контроль четности, значение, считываемое из MSB, является принятым битом четности</p>

### 37.5.3 USART\_BRR

Примечание – Счетчики скорости останавливают счет, если биты TE или RE запрещены, соответственно.

Таблица 568 – Регистр USART\_BRR

Base ADDR=	0x400C_0000					Offset=	0x0000_0008					Reset=	0x0000_0000				
REG Name:	USART_BRR																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
-																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Mantissa[11:0]											DIV_Fraction[3:0]				
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Таблица 569 – Описание бит регистра USART\_BRR

Бит	Имя	Значение	Описание
31...16	-	0	Зарезервировано
15...4	DIV_Mantissa[11:0]	0	<p>Мантисса USARTDIV.</p> <p>Эти 12 бит определяют мантиссу делителя USART (USARTDIV).</p>

Бит	Имя	Значение	Описание
3...0	DIV_Fraction[3:0]	0	дробная часть USARTDIV. Эти 4 бита определяют дробную часть делителя USART (USARTDIV). Если OVER8=1, бит DIV_Fraction3 не учитывается и должен быть сброшен

### 37.5.4 USART\_CR1

Таблица 570 – Регистр USART\_CR1

Base ADDR=		0x400C_0000				Offset=		0x0000_000C				Reset=		0x0000_0000			
REG Name:		USART_CR1															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
-																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	-	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Таблица 571 – Описание бит регистра USART\_CR1

Бит	Имя	Значение	Описание
31...16	-	0	Зарезервировано
15	OVER8	0	Режим передискретизации: 0 – передискретизация на 16; 1 – передискретизация на 8. Примечание – Передискретизация на 8 недоступна в режиме Smartcard: если SCEN=1, тогда OVER8 принудительно сбрасывается аппаратурой в 0
14	-	0	Зарезервировано
13	UE	0	Разрешение USART. Сброс данного бита в 0 моментально отключает TX, RX и генератор СК. При этом текущие передачи по TX и RX обрываются (если они идут). Данный бит устанавливается и сбрасывается программно: 0 – предделитель и выходы USART запрещены; 1 – USART разрешен
12	M	0	Длина слова. Данный бит определяет длину слова. Данный бит устанавливается и сбрасывается программно: 0 – 1 стартовый бит, 8 бит данных, n стоп-бит; 1 – 1 стартовый бит, 9 бит данных, n стоп-бит. Примечание – Бит M не должен быть модифицирован во время передачи данных (и приема и передачи)

Бит	Имя	Значение	Описание
11	WAKE	0	Метод пробуждения. Данный бит определяет метод пробуждения USART. Данный бит устанавливается и сбрасывается программно: 0 – линия Idle (состояние ожидания на линии); 1 – метка адреса
10	PCE	0	Разрешение контроля четности. Данный бит выбирает аппаратную генерацию и обнаружение бита четности. Если разрешен контроль бита четности, вычисленное значение четности вставляется на место MSB (9 бит, если M=1; 8 бит, если M=0) и на приеме этот бит проверяется на четность. Данный бит устанавливается и сбрасывается программно. Как только бит установлен, PCE активируется после текущего байта (при приеме и при передаче): 0 – контроль четности запрещен; 1 – контроль четности разрешен
9	PS	0	Выбор четности/нечетности. Данный бит выбирает, как вычисляется и проверяется бит четности – либо на четность, либо на нечетность (бит PCE установлен). Данный бит устанавливается и сбрасывается программно. Вариант контроля четности/нечетности выбирается после текущего байта. 0 – контроль на четность; 1 – контроль на нечетность
8	PEIE	0	Разрешение прерывание по событию PE. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено 1 – прерывание генерируется при PE=1 в регистре USART_SR
7	TXEIE	0	Разрешение прерывание по событию TXE. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при TXE=1 в регистре USART_SR
6	TCIE	0	Разрешение прерывания по завершению передачи Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при TC=1 в регистре USART_SR
5	RXNEIE	0	Разрешение прерывания по событию RXNE. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при ORE=1 или RXNE=1 в регистре USART_SR
4	IDLEIE	0	Разрешение прерывания по событию IDLE. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при IDLE=1 в регистре USART_SR

Бит	Имя	Значение	Описание
3	TE	0	<p>Разрешение передатчика.</p> <p>Данный бит разрешает передатчик. Данный бит устанавливается и сбрасывается программно:</p> <p>0 – передатчик запрещен;</p> <p>1 – передатчик разрешен.</p> <p>Примечание – Во время передачи, нулевой импульс в бите TE («0», за которым следует «1») передает преамбулу (сигнала ожидания на линии) после текущего слова, кроме режима smartcard.</p> <p>Когда установлен бит TE, существует 1-битная задержка перед началом передачи</p>
2	RE	0	<p>Разрешение приемника.</p> <p>Данный бит разрешает приемник. Данный бит устанавливается и сбрасывается программно:</p> <p>0 – приемник запрещен;</p> <p>1 – приемник разрешен, и он начинает искать стартовый бит</p>
1	RWU	0	<p>Пробуждение приемника.</p> <p>Данный бит определяет, находится ли приемопередатчик в режиме молчания или нет. Данный бит устанавливается и сбрасывается программно, а также может быть сброшен аппаратно при распознавании последовательности пробуждения:</p> <p>0 – приемник в активном режиме;</p> <p>1 – приемник в режиме молчания.</p> <p>Примечание – Перед выбором режима молчания (Mute) (установка бита RWU) приемопередатчик USART должен сначала получить байт данных, иначе он не сможет работать в режиме молчания с пробуждением по обнаружению сигнала ожидания на линии.</p> <p>При конфигурации пробуждения по метке адреса (WAKE=1) бит RWU не может быть модифицирован программно, пока установлен бит RXNE</p>
0	SBK	0	<p>Отправка сигнала break.</p> <p>Данный бит используется для отправки символов break. Данный бит устанавливается и сбрасывается программно. Он должен быть установлен программой и будет сброшен аппаратно во время стопового бита или символа break:</p> <p>0 – символ break не передается;</p> <p>1 – символ Break передается</p>



### 37.5.5 USART\_CR2

Таблица 572 – Регистр USART\_CR2

Base ADDR=		0x400C_0000				Offset=		0x0000_0010				Reset=		0x0000_0000		
REG Name:		USART_CR2														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-		STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	-				ADD[3:0]				
		RW	RW	RW	RW	RW	RW					RW	RW	RW	RW	

Таблица 573 – Описание бит регистра USART\_CR2

Бит	Имя	Значение	Описание
31...14	-	0	Зарезервировано
13..12	STOP[1:0]	0	Стоповые биты. Данные биты используются для программирования стоп-бит: 00 – зарезервировано; 01 – 0,5 стоп-бит; 10 – 2 стоп-бита; 11 – 1,5 стоп-бита
11	CLKEN	0	разрешение тактирования Данный бит позволяет разрешить работу вывода СК: 0 – вывод СК запрещен; 1 – вывод СК разрешен
10	CPOL	0	Полярность тактов. Данный бит позволяет выбрать полярность выхода тактов на выводе СК в синхронном режиме. Совместно с битом CPHA данный бит определяет взаимосвязь тактов и данных: 0 – постоянное значение 0 на выводе СК вне окна передачи; 1 – постоянное значение 1 на выводе СК вне окна передачи
9	CPHA	0	Фаза тактов. Данный бит позволяет выбрать фазу выхода тактов на выводе СК в синхронном режиме. Совместно с битом CPOL данный бит определяет взаимосвязь тактов и данных: 0 – первое изменение сигнала синхронизации является первым перепадом захвата данных; 1 – второе изменение сигнала синхронизации является первым перепадом захвата данных

Бит	Имя	Значение	Описание
8	LBCL	0	Тактовый импульс последнего бита. Бит позволяет выбрать, будет ли тактовый импульс последнего переданного бита (MSB) выдаваться на вывод СК в синхронном режиме: 0 – тактовый импульс последнего бита данных не выводится на вывод СК; 1 – тактовый импульс последнего бита данных выводится на вывод СК. Примечание – 1: последний бит – это 8 или 9 бит данных, в зависимости от выбранного формата, задаваемого битом M в регистре USART_CR1
7...4	-	0	Резерв
3...0	ADD[3:0]	0	Адреса узла USART. Данное битовое поле указывает адрес режима USART. Данные биты используются в многопроцессорном обмене во время режима молчания, для пробуждения приемопередатчика при обнаружении метки адреса. Примечание – Эти три бита (CPOL, CPHA, LBCL) не должны быть записаны, пока разрешен передатчик

### 37.5.6 USART\_CR3

Таблица 574 – Регистр USART\_CR3

Base ADDR=		0x400C_0000				Offset=		0x0000_0014				Reset=		0x0000_0000			
REG Name:		USART_CR3															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
-																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
				ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL			EIE		
				RW	RW	RW	RW	RW	RW	RW	RW	RW			RW		

Таблица 575 – Описание бит регистра USART\_CR3

Бит	Имя	Значение	Описание
31...12	-	0	Зарезервировано
11	ONEBIT	0	Разрешение режима одной выборки на бит. Бит позволяет выбрать метод оцифровки сигнала. Когда выбран метод одной выборки, флаг обнаружения шума (NF) запрещен: 0 – метод трех выборок; 1 – метод одной выборки. Примечание – функция ONEBIT применима только к битам данных. Это не относится к стартовому биту

Бит	Имя	Значение	Описание
10	CTSIE	0	Разрешение прерывания CTS: 0 – прерывание запрещено; 1 – прерывание генерируется каждый раз при CTS=1 в регистре USART_SR
9	CTSE	0	Разрешение CTS: 0 – запрещено аппаратное управление потоком CTS; 1 – режим CTS разрешен, данные передаются только, когда вход CTS имеет логический 0. Если вход CTS установлен на лог. 1 во время передачи данных, тогда передача завершается до остановки. Если данные записаны в регистр данных, когда CTS в логическую «1», то передача будет отложена до тех пор, пока на входе CTS не появится логический «0»
8	RTSE	0	Разрешение RTS: 0 – запрещено аппаратное управление потоком RTS; 1 – прерывание RTS разрешено, данные запрашиваются, когда есть место в буфере приемника. Ожидается, что передача данных прекратится после того, как будет передан последний текущий символ данных. Выход RTS имеет логический 0, когда приемопередатчик может принимать данные
7	DMAT	0	Разрешение передатчика DMA. Данный бит устанавливается и сбрасывается программно: 1 – режим DMA разрешен для передачи; 0 – режим DMA запрещен для передачи
6	DMAR	0	Разрешение приемника DMA. Данный бит устанавливается и сбрасывается программно: 1 – режим DMA разрешен для приема; 0 – режим DMA запрещен для приема
5	SCEN	0	Разрешение режима Smartcard. Данный бит используется для разрешения режима Smartcard: 0 – режим Smartcard запрещен; 1 – режим Smartcard разрешен
4	NACK	0	Разрешение NACK в режиме Smartcard. 0 – ответ NACK в случае ошибки четности запрещен; 1 – ответ NACK в случае ошибки четности разрешен
3	HDSEL	0	Выбор полудуплексного режима. Бит позволяет выбрать однопроводной режим или полудуплексный режим: 0 – полудуплексный режим не выбран; 1 – полудуплексный режим выбран
2, 1	-	0	Зарезервировано

Бит	Имя	Значение	Описание
0	EIE	0	Разрешение прерывания по ошибке. Бит используется для разрешения генерации прерывания по ошибке фрейма, ошибке переполнения или при флаге шума (FE=1 или ORE=1 или NF=1 в регистре USART_SR) в случае мультибуферного обмена (DMAR=1 в регистре USART_CR3): 0 – прерывание запрещено; 1 – прерывание генерируется каждый раз, когда DMAR=1 в регистре USART_CR3 и FE=1 или ORE=1 или NF=1 в регистре USART_SR

### 37.5.7 USART\_GTPR

Таблица 576 – Регистр USART\_GTPR

Base ADDR=		0x400C_0000				Offset=		0x0000_0018				Reset=		0x0000_0000		
REG Name:		USART_GTPR														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
-																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT[7:0]								-			PSC[4:0]				
RW	RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW

Таблица 577 – Описание бит регистра USART\_GTPR

Бит	Имя	Значение	Описание
31...16	-	0	Зарезервировано
15...8	GT[7:0]	0	Значение защитного интервала. Это битовое поле показывает значение защитного интервала времени в единицах тактов скорости. Биты используются в режиме Smartcard. Флаг завершения передачи устанавливается после этого значения защитного интервала
4...0	PSC[4:0]	0	Значение предделителя в режиме smartcard. Используется для программирования предделителя для деления системной тактовой частоты, чтобы обеспечить тактирование смарт-карты. Значение, заданное в регистре (5 значащих бит), умножается на 2, чтобы получить коэффициент деления исходной тактовой частоты: 00000 – зарезервировано – не программировать это значение; 00001 – делит источник тактов на 2; 00010 – делит источник тактов на 4; 00011 – делит источник тактов на 6

## 38 Модуль однократно программируемой памяти (ОТР)

### 38.1 Назначение и свойства

Модуль предназначен для чтения и однократного программирования блока постоянной памяти:

- модуль позволяет читать данные из адресного пространства памяти по АНВ, формируя необходимую задержку сигналом готовности;
- модуль позволяет записывать (программировать) данные через регистровый интерфейс, доступный по АНВ;
- объем ОТР составляет 128 Кбит (16 Кбайт);
- модуль осуществляет автономную блокировку диагностических функций по данным из специального адреса ОТР.

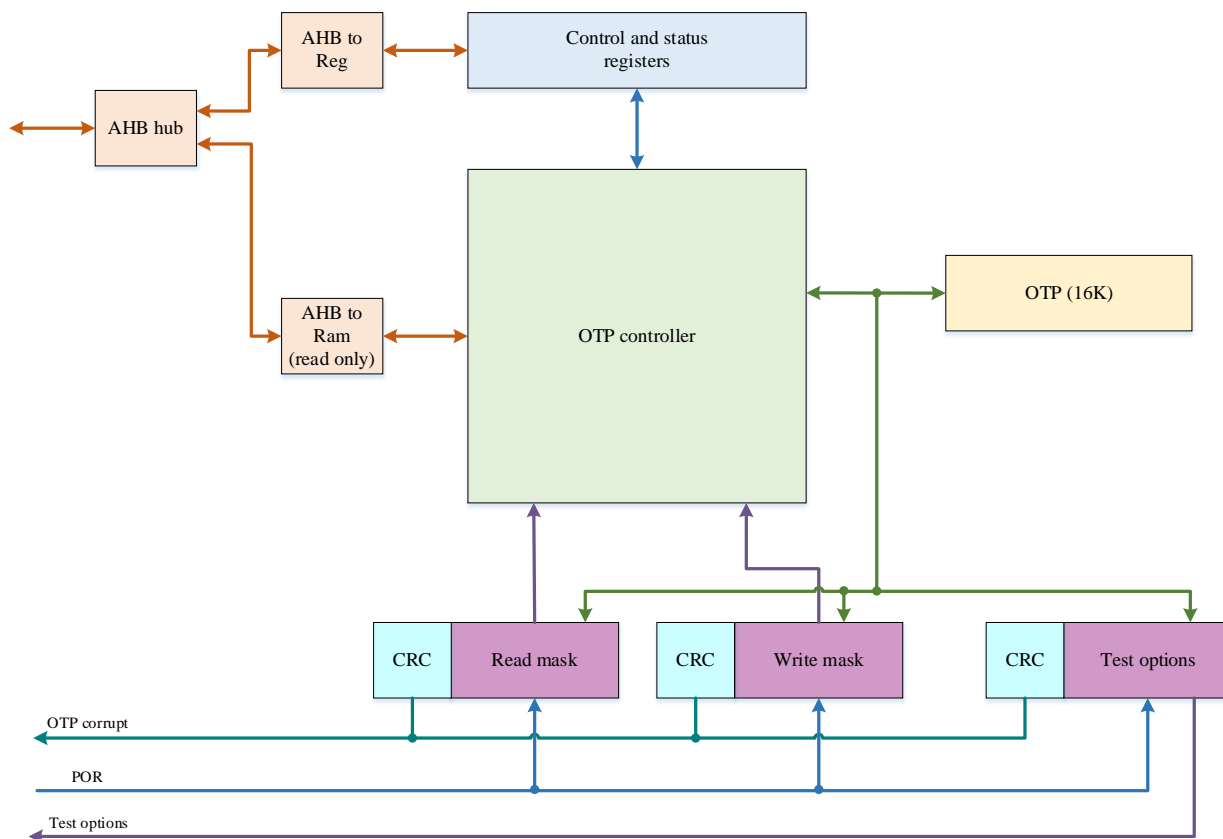


Рисунок 132 – Структурная схема модуля ОТР

Для правильной работы операций записи/чтения через регистровый интерфейс необходимо рассчитать и записать значения задержек DELAY\_20NS, DELAY\_50NS, DELAY\_01US, DELAY\_16US в регистрах DELAY\_0\_REG и DELAY\_1\_REG. Проверку готовности принять новый запрос на чтение или запись можно осуществлять через проверку флага BUSY регистра STAT\_CTRL или через флаг OTP\_BUSY регистра OTP\_STAT\_REG.

Для правильного чтения данных из памяти ОТР по шине АНВ (в том числе для исполнения из ОТР) необходимо рассчитать и записать значение задержки DELAY\_70NS в регистр DELAY\_0\_REG. При тактировании от HSI значение поля DELAY\_70NS по сбросу обеспечивает формирование требуемой задержки (в том числе без применения тримминга HSI). Число тактов паузы следует устанавливать до момента повышения

тактовой частоты или после снижения тактовой частоты. В таблице 578 приведены характеристики необходимой задержки для чтения памяти OTP по шине АНВ.

Таблица 578 – Характеристики паузы DELAY\_70NS для чтения памяти OTP по АНВ

DELAY_70NS [3:0]	Тактов паузы	Тактовая частота	Примечание
0x0	1	До 14 МГц	Установлено по умолчанию после сброса
0x1	2	До 28 МГц	
0x2	3	До 42 МГц	
0x3	4	До 57 МГц	
0x4	5	До 71 МГц	Работа микросхемы с частотой более 60 МГц не гарантируется

Регистры модуля OTP внутренним сигналом сброса ndmreset, сформированным отладчиком, не сбрасываются.

Модулем предусмотрена защита от записи/чтения в регистрах WRITE\_PROTECT\_REG и READ\_PROTECT\_REG для отдельных областей памяти в 2048 байт – защиту можно только установить, сброс осуществляется только общим сбросом по питанию. При попытке писать в защищенную от записи область или читать из защищенной от чтения области, а также при попытке чтения или записи по время установленного сигнала BUSY, выставляется флаг ошибки RW\_ERROR в регистрах STAT\_CTRL и RW\_CMD\_REG. Для защиты от модификации регистров WRITE\_PROTECT\_REG, READ\_PROTECT\_REG и TEST\_OPT сторонними методами после сброса и при штатной записи данных в регистры (для TEST\_OPT – после чтения битов защиты адресного пространства FLASH-памяти при включении основного питания микросхемы) для нового значения рассчитывается контрольная сумма, по которой контролируется целостность данных при чтении. Состояние сигнала целостности данных регистров заносится в биты READ\_PCRC, WRITE\_PCRC, TEST\_CRC регистра STAT\_CTRL.

### 38.1.1 Регистры модуля

Таблица 579 – Регистры модуля

Базовый Адрес		Название	Описание
0x7000_0000		OTP_MEM	Адресное пространство однократно программируемой памяти
0x7001_0000		OTP_REG	Блок регистров OTP
Смещение			
0x0000	0	STAT_CTRL	Регистр статуса и управления
0x0004	1	OTP_STAT	Регистр статуса модуля OTP
0x0008	2	DELAY_0	Регистр задержки 0
0x000c	3	DELAY_1	Регистр задержки 1
0x0010	4	RW_CMD	Регистр команды чтения-записи
0x0014	5	READ_DATA	Регистр считанных данных

Базовый Адрес		Название	Описание
0x0018	6	WRITE_PROTECT	Регистр защиты от записи
0x001C	7	READ_PROTECT	Регистр защиты от чтения
0x0020	8	-	Зарезервировано
0x0024	9	-	Зарезервировано
0x0028	10	TEST_OPT	Регистр включения тестовых опций

### 38.1.1.1 Регистр статуса и управления STAT\_CTRL

Относительный адрес: 0x00.

Таблица 580 – Регистр STAT\_CTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
2...0	FSM_STATE	Состояние конечного автомата общего контроллера модуля: 0 – бездействие; 1 – инициализация; 2 – готовность к работе; 3 – ожидание окончания чтения через регистры; 4 – ожидание окончания чтения через шину; 5 – ожидание окончания записи; 6 – тестовый режим прямого управления OTP	RO
3	BUSY	Флаг занятости OTP. OTP выполняет какие-либо действия и не готово принять очередной запрос на чтение или запись	RO
4	OTP_LOCK	Флаг блокировки OTP от записи. Единица в данном бите означает, что память OTP недоступна для записи	RO
5	OTP_PLOCK	Флаг частичной блокировки OTP от записи. Равен биту OTP_LOCK. Единица в данном бите означает, что память OTP недоступна для записи	RO
6	RW_ERROR	Флаг ошибки чтения или записи. Ошибка возникает: – при попытке чтения или записи во время установленного сигнала BUSY; – при попытке писать в защищенную от записи область или читать из защищенной от чтения области	RO
7	TEST_CRC	Флаг состояния целостности данных специальных тестовых опций (TEST_OPT). Единица в данном бите означает, что контрольная сумма данных корректна	RO
8	WRITE_PCRC	Флаг состояния целостности данных регистра защиты от записи (WRITE_PROTECT). Единица в данном бите означает, что контрольная сумма данных корректна	RO

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
9	READ_PCRC	Флаг состояния целостности данных регистра защиты от чтения (READ_PROTECT). Единица в данном бите означает, что контрольная сумма данных корректна	RO
10	DIR_EN	Запрос режима прямого управления. Не рекомендуется использовать. Всегда записывать 0	RW
31...11	-	Зарезервировано	RO

### 38.1.1.2 Регистр статуса модуля OTP, *OTP\_STAT\_REG*

Относительный адрес: 0x04.

Таблица 581 – Регистр статуса

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
4...0	OTP_FSM_STATE	Состояние конечного автомата модуля OTP: 0 – бездействие; 1 – переход в режим энергосбережения; 2 – подготовка сброса; 3 – сброс; 4 – окончание сброса; 5 – активный режим, ожидание чтения-записи; 6 – пауза режима чтения после записи; 7 – подготовка чтения; 8 – чтение; 9 – окончание чтения; 10 – подготовка записи; 11 – сохранение данных для записи; 12 – окончание сохранения данных; 13 – подготовка программирования; 14 – подготовка схемы повышения напряжения; 15 – программирование; 16 – выключение схемы повышения напряжения; 17 – ожидание окончания программирования; 18 – ожидание окончания записи; 19 – подготовка режима прямого управления; 20 – режим прямого управления	RO
5	-	Зарезервировано	RO
6	OTP_LOCK	Флаг аппаратной блокировки от записи модуля OTP	RO
7	OTP_BUSY	Флаг занятости: Модуль OTP выполняет какие-либо действия и не готов принять очередной запрос на чтение или запись	RO
31...8	-	Зарезервировано	RO



**38.1.1.3 Регистр задержки 0, DELAY\_0\_REG**

Относительный адрес: 0x08. Значение по сбросу: 0x000B\_0000

Таблица 582 – Регистр DELAY\_0

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
3...0	DELAY_20NS	Параметр задержки для формирования паузы 20 нс. Величина задержки задается в тактах основной частоты работы модуля, пауза равна N+1 такту, где N – заданное в поле число. И должна быть не меньше 20 нс	RW
7...4	DELAY_70NS	Параметр задержки для формирования паузы 70 нс при чтении данных из OTP по шине АНВ. Величина задержки задается в тактах основной частоты работы модуля, пауза равна N+1 такту, где N – заданное в поле число. И должна быть не меньше 70 нс	RW
12...8	DELAY_50NS	Параметр задержки для формирования паузы 50 нс. Величина задержки задается в тактах основной частоты работы модуля, пауза равна N+1 такту, где N – заданное в поле число. И должна быть не меньше 50 нс	RW
15...13	-	Зарезервировано	RO
24...16	DELAY_01US	Параметр задержки для формирования паузы 1 мкс. Величина задержки задается в тактах основной частоты работы модуля, пауза равна N+1 такту, где N – заданное в поле число. И должна быть не меньше 1 мкс и не более 2 мкс	RW
31...25	-	Зарезервировано	RO

**38.1.1.4 Регистр задержки 1, DELAY\_1\_REG**

Относительный адрес: 0x0C. Значение по сбросу: 0x0000\_007F

Таблица 583 – Регистр DELAY\_1\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
16...0	DELAY_16US	Параметр задержки для формирования паузы 16 мкс. Величина задержки задается в тактах основной частоты работы модуля, пауза равна N+1 такту, где N – заданное в поле число. И должна быть не меньше 16 мкс и не больше 17 мкс	RW
31...17	-	Зарезервировано	RO

**38.1.1.5 Регистр команды чтения-записи RW\_CMD\_REG**

Относительный адрес: 0x10.

Таблица 584 – Регистр RW\_CMD\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
16...0	ADDR	Битовый адрес записи и чтения Запись производится битами, чтение производится байтами, при чтении младшие три бита игнорируются и считаются равными 0	RW
23...17	-	Зарезервировано	RO
24	DATA	Данные для записи в OTP, 0 – не производить запись. Начальное состояние OTP – 0, записать можно только 1, стереть 1 уже нельзя	RW
27...25	-	Зарезервировано	RO
28	READ	Команда на чтение из указанного адреса При наличии 1 в данном поле и 0 в поле команды записи производится процедура чтения. Если выбранная область не защищена от чтения и OTP не занято, произойдет чтение данных, считанные данные будут возвращены в регистр считанных данных. Окончание контролируется по флагу BUSY в регистре статуса	RW
29	WRITE	Команда на запись в указанный адрес При наличии 1 в данном поле и 0 в поле команды чтения производится процедура записи. Если выбранная область не защищена от записи и OTP не занято, произойдет запись данных. Окончание контролируется по флагу BUSY в регистре статуса	RW
30	-	Зарезервировано	RO
31	RW_ERROR	Ошибка выполнения последней операции чтения или записи. Ошибка возникает при попытке чтения или записи во время установленного флага BUSY При попытке писать в защищенную от записи область или читать из защищенной от чтения области	RO

**38.1.1.6 Регистр считанных данных READ\_DATA\_REG**

Относительный адрес: 0x14.

Таблица 585 – Регистр READ\_DATA\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
7...0	DATA	Данные считанные из памяти OTP в случае успешного завершения процедуры чтения, при ошибках в поле заноситься константа 0xEF	RO
31...8	-	Зарезервировано	RO

**38.1.1.7 Регистр защиты от записи WRITE\_PROTECT\_REG**

Относительный адрес: 0x18.

Таблица 586 – Регистр WRITE\_PROTECT\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
7...0	WP_OTP	Каждый бит регистра отвечает за область памяти в 2048 байт. 0 – за адреса 0 – 2047; 1 – за адреса 2048 – 4095 и так далее. 1 в бите запрещает запись в данный регион памяти. Биты защиты можно только установить, сбросить биты защиты можно только общим сбросом по питанию. Сброс процессора через регистры управления не снимает установленные биты. При штатной записи данных в регистр, автоматически рассчитывается контрольная сумма для нового значения. По этой сумме контролируется целостность данных, на случай модификаций бит защиты сторонними методами	RW
31...8	-	Зарезервировано	RO

**38.1.1.8 Регистр защиты от чтения READ\_PROTECT\_REG**

Относительный адрес: 0x1C.

Таблица 587 – Регистр READ\_PROTECT\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
7...0	RP_OTP	Каждый бит регистра отвечает за область памяти в 2048 байт. 0 – за адреса 0 – 2047; 1 – за адреса 2048 – 4095 и так далее. 1 в бите запрещает чтение из данного региона памяти. При попытке чтения из защищенного региона возвращается значение 0xEF. Биты защиты можно только установить, сбросить биты защиты можно только сбросом процессора. При штатной записи данных в регистр, автоматически рассчитывается контрольная сумма для нового значения. По этой сумме контролируется целостность данных, на случай модификаций бит защиты сторонними методами	RW
31...8	-	Зарезервировано	RO

**38.1.1.9 Регистр включения тестовых опций TEST\_OPT\_REG**

Относительный адрес: 0x28.

Таблица 588 – Регистр TEST\_OPT\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
7...0	DEBUG	<p>Регистр содержит биты разрешения включения отладчика. Если не все биты данного поля будут единицами или будет нарушена целостность данных регистра, отладчик будет заблокирован.</p> <p>Регистр обновляется при любом чтении из последнего адреса ОТР, в регистр заносится инверсия значения, считанного из ОТР.</p> <p>При штатном обновлении данных в регистре, автоматически рассчитывается контрольная сумма для нового значения. По этой сумме контролируется целостность данных, на случай модификаций сторонними методами.</p> <p>Сброс регистра происходит общим сбросом питания, сброс процессора через регистры управления не изменяет регистр.</p> <p>В сброшенном состоянии все опции отключены</p>	RO
31...8	-	Зарезервировано. Всегда читается единицами	RO

### 39 Электрические параметры микросхем

Таблица 589 – Электрические параметры микросхем при приемке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра*		Температура среды, °С
		не менее	не более	
Выходное напряжение высокого уровня, В, на выводах PA, PB, PC, PD	U <sub>OH</sub>	U <sub>CC</sub> – 0,6	-	25, 85, – 50
Выходное напряжение низкого уровня, В, на выводах PA, PB, PC, PD	U <sub>OL</sub>	-	0,4	
Ток утечки низкого уровня цифровых входов, мкА, – на выводах: PA, PB, PC, PD, nRESET, WAKEUP, JTAG_EN, OSC_IN32, SAR_AIN; – на выводе OSC_IN	I <sub>ILL</sub>	- 1,0	1,0	
		- 20	- 1	
Входной ток утечки высокого уровня, мкА, – на выводах: PA, PB, PC, PD, nRESET, WAKEUP, OSC_IN32, SAR_AIN; – на выводе JTAG_EN; – на выводе OSC_IN	I <sub>ILH</sub>	- 1,0	1,0	
		10	80	
		1	20	
Динамический ток потребления, мА, при f <sub>C</sub> ≤ 60 МГц	I <sub>CCO</sub>	-	70	
Выходная частота HSI RC-генератора, МГц, после выполнения тримминга	f <sub>O_HSI</sub>	7,9	8,1	
Выходная частота LSI RC-генератора, кГц, после выполнения тримминга	f <sub>O_LSI</sub>	30	32	
Выходная частота PLL, МГц: – максимальная; – минимальная	f <sub>O_PLL</sub>	60	-	
		-	8	
Параметры ΔΣАЦП				
Соотношение сигнал/шум, дБ, усиление 0 дБ, U <sub>ID_ADCD</sub> = 1 В, f <sub>NADCO</sub> ** = 4 кГц	SNR <sub>D0</sub>	78	-	25, 85, – 50
<p>* Значение уточняется в ходе ОКР.  ** f<sub>NADCO</sub> – частота передискретизации на выходе микросхем.  Примечание – Режимы измерения параметров приведены в разделе 3</p>				

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В\*.

## 40 Предельно-допустимые и предельные параметры

Таблица 590 – Предельно-допустимые электрические режимы эксплуатации и предельные электрические режимы микросхем

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра*			
		Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение питания, В, на выводах V <sub>CC</sub>	U <sub>CC</sub>	2,2	3,6	-	3,9
Напряжение питания аналоговое, В, на выводах V <sub>CCA</sub>	U <sub>CCA</sub>	3,0	3,6	-	3,9
Напряжение питания батарейного домена, В, на выводе V <sub>CCB</sub>	U <sub>CCB</sub>	1,8	3,6	-	3,9
Входное напряжение высокого уровня, В, на входах портов PA, PB, PC, PD и на входах JTAG_EN, nRESET при: – U <sub>CC</sub> ≥ 3,0 В; – U <sub>CC</sub> < 3,0 В	U <sub>IH</sub>	2,0			
		1,5	U <sub>CC</sub>	-	U <sub>CC</sub> + 0,3
Входное напряжение низкого уровня, В, на входах портов PA, PB, PC, PD и на входах JTAG_EN, nRESET	U <sub>IL</sub>	0	0,8	-0,3	-
Выходной ток высокого уровня, мА, на выходах портов PA, PB, PC, PD	I <sub>OH</sub>	-4,0	-	-6,0	-
Выходной ток низкого уровня, мА, на выходах портов PA, PB, PC, PD	I <sub>OL</sub>	-	4,0	-	6,0
Частота следования импульсов тактовых сигналов PLL, МГц	f <sub>C_PLL</sub>	8	16	-	-
Частота следования импульсов тактовых сигналов HSE, МГц, на выводе OSC_IN, при: – BYPASS = «0»; – BYPASS = «1»	f <sub>C_HSE</sub>	8	16	-	-
		-	60	-	-
Частота следования импульсов тактовых сигналов LSE, кГц, на выводе OSC_IN32, при: – BYPASS = «0»; – BYPASS = «1»	f <sub>C_LSE</sub>	32	33	-	-
		-	1000	-	-
Емкость нагрузки, пФ, на выводах: PA, PB, PC, PD	C <sub>L</sub>	-	30	-	-

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра*			
		Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Параметры $\Delta\Sigma$ АЦП					
Опорное напряжение, В, на выводе VREF, при выключенном внутреннем источнике опорного напряжения	U <sub>REF_ADC</sub>	2,39	2,51	–	–
Входное напряжение дифференциальное, В, на выводах I0P – I3P, I0N – I3N, V0P – V2P, V0N – V2N	U <sub>ID_ADCD</sub>	– 1,0	1,0	–	–
Напряжение, В, на выводах I0P – I3P, I0N – I3N, V0P – V2P, V0N – V2N	U <sub>I_ADCD</sub>	– 0,5	0,5	– 0,8	U <sub>CC</sub> + 0,3
Частота следования импульсов тактовых сигналов $\Delta\Sigma$ АЦП, МГц	f <sub>C_ADCD</sub>	0,9	1,1	–	–
Параметры SAR АЦП					
Входное напряжение SAR АЦП, В, на входах SAR_AIN1, SAR_AIN2, SAR_AIN3 при: – использовании внутреннего источника опорного напряжения; – подаче опорного напряжения на вывод VREF;	U <sub>AIN</sub>	0	U <sub>REF_ADC</sub>	– 0,3	3,9
			U <sub>CCA</sub>		
– без использования опорного напряжения					
Частота следования импульсов тактовых сигналов SAR АЦП, МГц	f <sub>C_ADCS</sub>	–	28	–	–
<p>Примечание – Не допускается одновременное задание двух предельных режимов. * Значение уточняется в ходе ОКР</p>					

Коэффициент программируемости N<sub>PR</sub> должен быть не менее 0,8.

## 41 Справочные параметры

Тепловое сопротивление кристалл-корпус не более 4,6 °C/Вт.

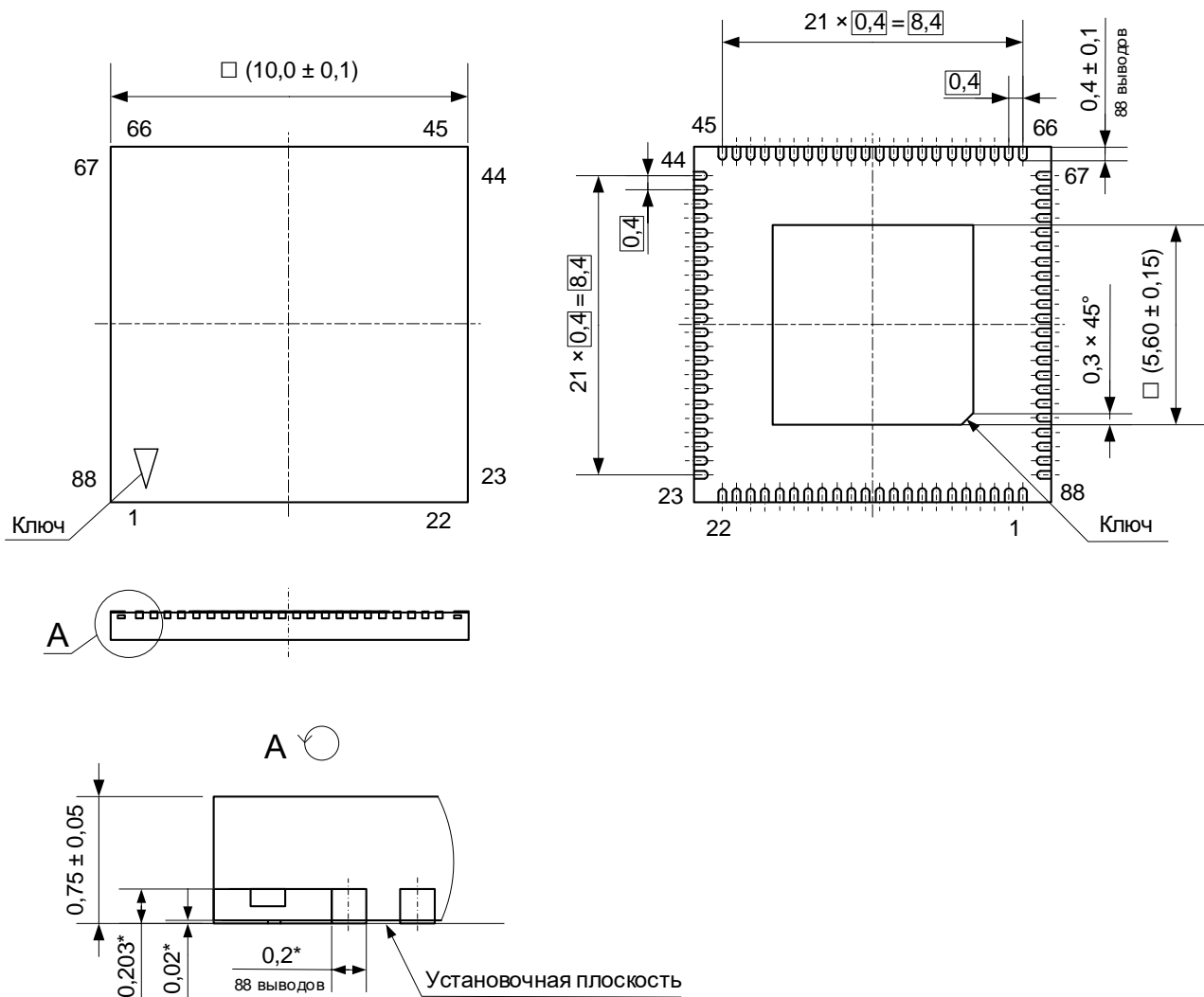
Таблица 591 – Справочные параметры микросхем

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра*			Температура среды, °C
		не менее	типовое	не более	
Статический ток потребления, мА	I <sub>CC</sub>	-	1,86	3,50	25, 85, – 50
Статический ток потребления батареино домена, мкА	I <sub>CCB</sub>	-	-	1	
Время хранения информации во FLASH, лет, при: – T = 25 °C – T = 85 °C	t <sub>SG</sub>	25	-	-	
		10	-	-	
Тактовая частота ядра микроконтроллера, МГц	f <sub>C</sub>	-	-	60	
Сопротивление внутренних программируемых резисторов доопределения к шине обций, кОм	R <sub>PD</sub>	55	85	144	
Сопротивление внутренних программируемых резисторов доопределения к шине питание, кОм	R <sub>PU</sub> **	56	78	116	
Параметры SPI					
Частота синхросигнала SSP_CLK, МГц, – в режиме master; – в режиме slave	f <sub>OP</sub>	-	-	$\frac{f_c}{2}$	25, 85, – 50
		-	-	$\frac{f_c}{4}$	
Длительность сигнала высокого/низкого уровня синхросигнала SSP_CLK, нс	t <sub>WSSP_CLK</sub>	$\frac{1}{2 \cdot f_{OP}}$	-	-	
Время задержки распространения данных при переходе выхода SSP_TXD из состояния «Выключено» в состояние высокого/низкого уровня по сигналу SSP_FSS, нс	t <sub>PZH</sub> (SSP_CLK- SSP_TXD)	$\frac{1}{2 \cdot f_{OP}}$	-	-	
	t <sub>PZL</sub> (SSP_CLK- SSP_TXD)				
Время задержки распространения данных при переходе выхода SSP_TXD из состояния высокого /низкого уровня в состояние «Выключено» по сигналу SSP_FSS, нс	t <sub>PHZ</sub> (SSP_CLK- SSP_TXD) t <sub>PLZ</sub> (SSP_CLK- SSP_TXD)	$\frac{1}{2 \cdot f_{OP}}$	-	-	



Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра*			Температура среды, °С
		не менее	типовое	не более	
Время установления входных данных SSP_RXD относительно SSP_CLK, нс	$t_{SU(SSP\_RXD-SSP\_CLK)}$	2,5	-	-	25, 85, - 50
Время удержания сигнала входных данных SSP_RXD относительно SSP_CLK, нс	$t_{H(SSP\_CLK-SSP\_RXD)}$	3,5	-	-	
Время сохранения выходных данных SSP_TXD относительно сигнала SSP_CLK, нс	$t_{V(SSP\_CLK-SSP\_TXD)}$	0	-	-	
Параметры SAR АЦП					
Разрядность АЦП, бит	$E_{NADC}$	10	-	-	25, 85, - 50
Дифференциальная нелинейность, ЕМР	$E_{DLADC}$	- 1,00	от - 0,60 до 0,26 включ.	1,00	
Интегральная нелинейность, ЕМР	$E_{ILADC}$	- 1,0	от - 0,4 до 0,4 включ.	1,0	
Ошибка смещения, ЕМР	$E_{OFFADC}$	- 1,00	- 0,25	1,00	
Ошибка усиления, %	$E_{GAINADC}$	- 0,10	- 0,05	0,10	
Параметры $\Delta\Sigma$ АЦП					
Напряжение внутреннего источника опорного напряжения, В, на выводе VREF	$U_{REFO}$	-	2,45	-	25, 85, - 50
Соотношение сигнал / нелинейные искажения, дБ, усиление + 0 дБ, $U_{ID\_ADCD} = 1$ В, $f_{NADCO} = 4$ кГц	$THD_{D0}$	90	96	-	
Соотношение сигнал / шум, дБ, усиление + 0 дБ, $U_{ID\_ADCD} = 1$ В, $f_{NADCO} = 4$ кГц	$SNR_{D0}$	-	86	-	
Соотношение сигнал / шум, дБ, усиление + 6 дБ, $U_{ID\_ADCD} = 0,5$ В, $f_{NADCO} = 4$ кГц	$SNR_{D6}$	-	84	-	
Соотношение сигнал / шум, дБ усиление + 12 дБ, $U_{ID\_ADCD} = 0,25$ В, $f_{NADCO} = 4$ кГц	$SNR_{D12}$	-	80	-	
Соотношение сигнал / шум, дБ усиление + 18 дБ, $U_{ID\_ADCD} = 0,125$ В, $f_{NADCO} = 4$ кГц	$SNR_{D18}$	-	75	-	
Ошибка усиления предусилителя, дБ	$GAIN_{ERR}$	- 2,0	- 0,5	2,0	
* Значение уточняется в ходе ОКР; ** Доступно с ревизии 2					

## 42 Габаритный чертеж микросхемы



1 \* Размеры для справок.

2 Нумерация выводов показана условно.

Рисунок 133 – Габаритный чертеж микросхемы в корпусе QFN88 (TICP)

### 43 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон, °С
K1986BK025	MDR32F02FI RISC-V	QFN88 (TICP)	от – 50 до 85

## Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	24.11.2020	0.1.0	Введена впервые	
2	13.01.2021	0.2.0	Внесение дополнений ГК, исправление опечаток	По тексту
3	11.03.2021	0.3.0	Замена корпуса на QFN88 (TICP). Уточнения и дополнения в разделах 11, 18.4, 19.2, 20. В таблице 33 исправлены номера битов для FOVGAIN, FOIOGAIN Исправлен доступ битов в таблицах 346, 348. Исправлено описание битов 0, 1 в таблице 519	1, 390, 391 46, 147, 166, 174 53  177 333
4	01.04.2021	0.4.0	Исправлено описание битов 27...25 в таблице 326	170
5	07.05.2021	0.4.1	Добавлено международное обозначение микросхемы	391
6	06.09.2021	0.5.0	Исправление замечаний и опечаток Обновлена структурная блок-схема Добавлено УГО Таблицы 589, 590 приведены в соответствие с ТУ Добавлен раздел «Справочные параметры» Добавлены регистры и их описания в раздел 13.1 Скорректированы пункты 7.1, 9, 13, 14, 16.3, 16.4, 20.3, 20.3.1, 21, 0, 0 – 23.14.2.4, 23.14.2.6 – 23.14.2.8 Скорректированы рисунки 4, 17, 39, 40 Скорректированы таблицы 29, 222 – 226, 238, 250, 254 – 256, 263, 264, 295, 292, 293, 304, 305, 346 – 348, 482	По тексту 18 19 457,458  460 69, 110 – 112, 118, 120 32,39 – 48 66, 136, 121, 183, 184, 214, 217, 255 – 257, 259 32, 68, 185, 185 73, 123 – 126, 137, 142 – 145, 149, 154, 165, 173, 218, 219, 357
7	08.11.2021	0.6.0	Исправление опечаток Раздел 22 – добавлена информация о настройке значения предделителя DIV Подраздел 25.4.1 – исключено условие о нахождение контроллера не в тестовом режиме Подраздел 36.1 – удалена информация о прохождении статистических тестов Раздел 37 – исключена информация о возможности использования 1 стоп-бита Рисунок 116 – изменено состояние флага ТС после 3 фрейма	По тексту 225  308  408  412  418

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			Подраздел 37.3.4.2 – удалены таблицы определения погрешностей. Добавлена таблица определения погрешностей (при fPCLK = 60 МГц) Рисунок 130 – изменено условие передачи Data 3 Подраздел 37.5 обновлен Таблица 570 – изменено примечание после бита 15	424 436 437 442
8	23.12.2021	0.7.0	Исправление опечаток Таблица 2 – функции выводов PB8, PB9 исправлены Таблица 294 – исправлены функции портов PC2, PC3, PC6, PC7 Рисунок 30 – добавлен резистор подтяжки Подраздел 0 – формула исправлена Подраздел 25.3.1 – изменены параметры «ADCUIDMABREQx» и «ADCUIDMASREQx» Таблица 566 исправлена Добавлено значение коэффициента программируемости	По тексту 23 168 170 217 304 440 458
9	19.10.2022	0.8.0	Исправление опечаток Основные характеристики: аналоговые модули – добавлено примечание Раздел 7 дополнен, описание вывода V <sub>ССА</sub> дополнено Подраздел 7.2 скорректирован список сигналов для начальной установки микросхемы Подраздел 9.2 скорректирован Раздел 13 – добавлена информация о применении интегратора Рисунок 17 обновлен Таблица 27 – изменено описание регистров ADCUI_FxV DAT, *_F0Ix DAT, *_FxIDAT Таблицы 53, 97, 141 – описание битов FxVSEL скорректировано Раздел 14 добавлена информация о буфере FIFO; о значении RMS Подраздел 16.1 дополнен Подраздел 17.1 дополнен Раздел 20 скорректирован и дополнен Подраздел 21.4 дополнен Раздел 26 – скорректировано описание сигнала timer_pulse после таблицы 480 Подраздел 28.3 добавлен Раздел 41 – добавлен параметр THD <sub>D0</sub>	По тексту 2 31 34 43-48 66 68 69 80, 92, 103 121 136 150 178-213 216 357 372 460

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
10	25.07.2023	0.9.0	<p>Исправление опечаток</p> <p>Подраздел 7.1 дополнен</p> <p>Раздел 9 скорректирован</p> <p>Подраздел 9.2 скорректирован</p> <p>Таблица 27 скорректирована</p> <p>Таблицы 37, 43, 48, 52; 81, 87, 93, 97; 125, 131 136, 141 – описание битов скорректировано</p> <p>Таблица 226 – добавлено примечание</p> <p>Рисунок 28 скорректирован</p> <p>Таблицы 246, 250, 252 – описание битов CPU C3 SEL, ADC C3 SEL и ADCUI C3 SEL, HSI SEL и HSE SEL скорректировано</p> <p>Таблица 295 – описание бита FPOR скорректировано</p> <p>Таблица 307 – описание бита PWR0 скорректировано</p> <p>Таблицы 314, 315 – имена битов 26, 24 исправлены</p> <p>Раздел 20 обновлен</p> <p>Раздел 21 – условия преобразования скорректированы</p> <p>Подразделы 21.5, 21.6 – «ACLK» → «C_ADCS»</p> <p>Пункт 21.7.1 – описание битов Cfg REG CLKS и Cfg REG DIVCLK скорректировано</p> <p>Таблица 482 – Принципы формирования прерываний int_global(11-13) скорректированы</p> <p>Подразделы 34.1, 34.2 скорректированы</p> <p>Пункт 34.3.5 – информация об OTP исключена</p> <p>Пункт 34.3.6 дополнен</p> <p>Пункт 35.1.6 добавлен</p> <p>Таблица 590 – параметры <math>U_{IH}</math>, <math>U_{IL}</math> скорректированы</p> <p>Раздел 41 добавлен параметры: тепловое сопротивление кристалл-корпус, <math>R_{PD}</math>, <math>R_{PU}</math></p>	<p>По тексту</p> <p>32</p> <p>39</p> <p>43</p> <p>69</p> <p>77-103</p> <p>127</p> <p>135</p> <p>139, 142, 143</p> <p>154</p> <p>173</p> <p>176</p> <p>178-212</p> <p>214</p> <p>217, 218</p> <p>219</p> <p>357</p> <p>398, 399</p> <p>401</p> <p>402</p> <p>407</p> <p>458</p> <p>460</p>
11	04.06.2024	1.0.0	<p>Рисунок 2 – добавлено примечание</p> <p>Таблицы 1, 2 – назначение выводов PC0, PC1 скорректировано</p> <p>Раздел 7 – время удержания сброса схемой POR: ~ 4 мс → ~ 12 мс</p> <p>Рисунки 4, 5 – добавлены паразитные диоды транзисторов</p> <p>Рисунок 5 – примечание 3 дополнено</p> <p>Раздел 9 скорректирован</p> <p>Пункты 10.2.1-10.2.4 скорректированы</p>	<p>19</p> <p>21, 23</p> <p>31</p> <p>32, 33</p> <p>33</p> <p>39</p> <p>51-55</p>

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			Таблица 18 – описание битов скорректировано	58
			Рисунок 17 обновлен	68
			Таблицы 35, 49, 79, 93, 123, 137 – описание битов *RMSOS дополнено	76, 79, 88, 91, 99, 102
			Формула (4) исправлена	124
			Подраздел 17.1 скорректирован и дополнен	150
			Таблица 265 скорректирована	153
			Таблица 271, 281, 283, 285, 293 – описание битов скорректировано	155-165
			Таблица 275 – описание битов 11-0 скорректировано	158
			Раздел 20 – названия регистров скорректированы: *EVENT* → *EVNT*, CCRy → CHy_CCR, CHy_CTRL → CHy_CTRL0	178-212
			Пункты 20.2.1, 20.3.1 дополнены	180, 184
			Пункт «Тактовая частота F <sub>DTS</sub> » добавлен	183
			Таблица 325 – описание бит EVNT_SEL[3:0]=0000 скорректировано	201
			Подраздел 21.2, таблица 351 – дополнена информация о значении бита Cfg_REG_GO	215, 220
			Раздел 23 – добавлена информация о ревизии 2	231-257
			Подраздел 23.2 – добавлено примечание	233
			Пункт 23.13.4 дополнен	253
			Подраздел 25.6 – скорректировано положение об обращении к зарезервированным адресам	341
			Раздел 27 скорректирован и дополнен	359-365
			Таблицы 476, 480, 481 – доступ скорректирован	356-357
			Подраздел 29.1 скорректирован	373
			Таблица 504 – описание дополнено	372
			Таблица 552 – описание битов скорректировано	410
			Таблица 562 – событие «Флаг Break» исключено	436
			Рисунок 131 исключены линии LBD и LBDIE	437
			Подраздел 38.1 дополнен	449
			Таблица 580 – описание битов скорректировано	451
			Таблица 581 – описание битов 6, 7 скорректировано	452
			Таблица 588 – биты 31:8 зарезервированы	456
			Рисунок 133 – ГЧ обновлен	462
12	30.07.2024	1.1.0	Подраздел 7.2 исправлено примечание под рисунком 7	35
			Подраздел 10.1 «Delay [1:0]» исправлено на «Delay[2:0]»	50
			Таблица 13 исправлены наименования строк в первом столбце	50

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			Пункт 12.1.4 скорректирован	65
			Таблица 241 исправлено значение по сбросу для «PLL_CPU_MUL[2:0]»	138
			Подпункт 25.4.5.3 заменен пункт 2 в нумерованном списке	319
			Раздел 25 – «2R» исправлено на «2 <sup>R</sup> » под таблицей 432	326
			Подраздел 29.1 – «Пример работы с блоком IWDG» - добавлен пункт 6, исправлена опечатка в пункте 5	373