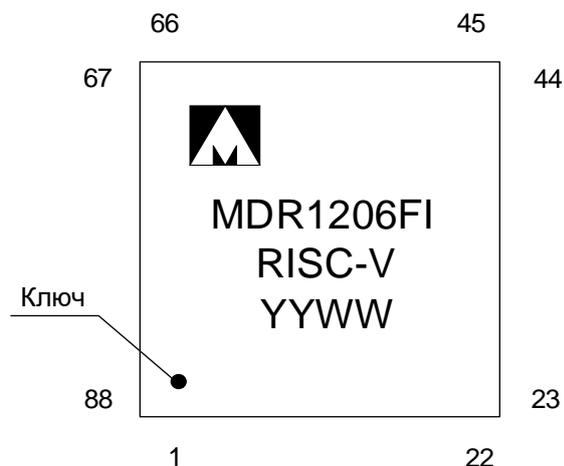




**Микросхема микроконтроллера  
для трехфазного и однофазного электросчетчиков  
MDR1206FI**

**Основные характеристики микросхемы:**

- Напряжение источника питания:
  - от 3,0 до 3,6 В (с AFE);
  - от 1,8 до 3,6 В (без AFE);
- 32-разрядная RISC-V архитектура BM-310S с системой команд RV32IMC;
- Встроенная Flash-память 2x256 Кбайт;
- Встроенная память RAM 112 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.

**Общее описание и область применения микросхемы**

Микросхемы интегральные MDR1206FI (далее – микросхемы) представляют собой микроконтроллеры со встроенной Flash-памятью программ и построены на базе процессорного RISC-V ядра BM-310S. Предназначены для использования в приборах учета потребляемой электрической энергии в однофазных и трехфазных бытовых сетях энергоснабжения 230 В/50 Гц и 110 В/60 Гц.

## Основные характеристики:

### Ядро:

- 32-битное RISC-V ядро BM-310S с системой команд RV32IMC и тактовой частотой до 48 МГц;
- умножение за два цикла.

### Память:

- встроенная энергонезависимая Flash-память программ типа размером 2×256 Кбайт (основная область) + 2×8 Кбайт (информационная область);
- встроенное ОЗУ размером 112 Кбайт.

### Питание и тактовая частота:

- внешнее питание:
  - от 3,0 до 3,6 В (с AFE);
  - от 1,8 до 3,6 В (без AFE);
- 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 от 8 до 16 МГц;
- внешний осциллятор 32 кГц;
- встроенный умножитель тактовой частоты PLL для ядра.

### Режим пониженного энергопотребления:

- батарейный домен с часами реального времени, календарём, 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;
- генератор случайных чисел;
- защитная сетка, датчики напряжения питания;
- до 55-ти 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.....	14
2	Структурная блок-схема.....	15
3	Условное графическое изображение.....	16
4	Диаграмма расположения выводов в корпусе.....	17
5	Описание выводов.....	18
6	Указания по применению и эксплуатации .....	26
7	Система питания.....	27
7.1	Структурная схема подачи питания .....	28
7.2	Схема сброса при включении и выключении основного питания .....	29
8	Организация памяти.....	30
8.1	Базовые адреса процессора .....	32
9	Загрузочное ПО и режимы работы микроконтроллера.....	33
9.1	Пример одного из возможных вариантов загрузочной программы.....	34
9.1.1	UART-загрузчик .....	35
9.1.2	Команды UART-загрузчика.....	36
10	Контроллер Flash-памяти программ.....	40
10.1	Работа Flash-памяти программ в обычном режиме .....	40
10.2	Работа Flash-памяти программ в режиме программирования.....	40
10.2.1	Стирание всей памяти или 2×256 Кбайт основной памяти.....	41
10.2.2	Стирание страницы памяти размером 4 Кбайт.....	42
10.2.3	Запись 32-битного слова в память .....	43
10.2.4	Чтение 32-битного слова из памяти.....	44
10.3	Регистры управления контроллера Flash-памяти программ.....	46
10.3.1	FLASH_CMD.....	46
10.3.2	FLASH_ADR .....	47
10.3.3	FLASH_DI .....	48
10.3.4	FLASH_DO.....	48
10.3.5	FLASH_KEY .....	48
10.3.6	CHIP_ID_CTRL.....	49
11	Система команд .....	50
12	Процессорное ядро BM-310S.....	52
12.1	Структурная схема процессора.....	53
12.1.1	Интерфейс TCM.....	53
12.1.2	Интерфейс АНВ I/O.....	53
12.1.3	Запросы прерываний .....	54
12.1.4	Интерфейс JTAG.....	54
12.1.5	LIO crossbar .....	54
13	Блок АЦП для измерения напряжений и токов в электрической сети .....	55
13.1	Описание регистров управления блока семи каналов АЦП .....	60
13.1.1	ADCUI_CTRL1 .....	63
13.1.2	ADCUI_CTRL2 .....	67
13.1.3	ADCUI_CTRL3 .....	67
13.1.4	ADCUI_F0CTR.....	68
13.1.5	ADCUI_F0WC .....	69
13.1.6	ADCUI_F0WATTP.....	70
13.1.7	ADCUI_F0WATTN.....	70

13.1.8	ADCUI_F0VC	70
13.1.9	ADCUI_F0VARP	71
13.1.10	ADCUI_F0VARN	71
13.1.11	ADCUI_F0AC	71
13.1.12	ADCUI_F0VR	72
13.1.13	ADCUI_F0MD0	72
13.1.14	ADCUI_F0MD1	73
13.1.15	ADCUI_F0VPEAK	74
13.1.16	ADCUI_F0IPEAK	74
13.1.17	ADCUI_F0VDAT	75
13.1.18	ADCUI_F0I0DAT	75
13.1.19	ADCUI_F0I3DAT	75
13.1.20	ADCUI_F0VRMS	76
13.1.21	ADCUI_F0VRMS2	76
13.1.22	ADCUI_F0IRMS	76
13.1.23	ADCUI_F0IRMS2	77
13.1.24	ADCUI_F0STAT	77
13.1.25	ADCUI_F0MASK	79
13.1.26	ADCUI_F1CTR	80
13.1.27	ADCUI_F1WC	81
13.1.28	ADCUI_F1WATTP	82
13.1.29	ADCUI_F1WATTN	82
13.1.30	ADCUI_F1VC	82
13.1.31	ADCUI_F1VARP	83
13.1.32	ADCUI_F1VARN	83
13.1.33	ADCUI_F1AC	83
13.1.34	ADCUI_F1VR	84
13.1.35	ADCUI_F1MD0	84
13.1.36	ADCUI_F1MD1	85
13.1.37	ADCUI_F1MD2	86
13.1.38	ADCUI_F1VPEAK	86
13.1.39	ADCUI_F1IPEAK	86
13.1.40	ADCUI_F1VDAT	87
13.1.41	ADCUI_F1IDAT	87
13.1.42	ADCUI_F1VRMS	88
13.1.43	ADCUI_F1VRMS2	88
13.1.44	ADCUI_F1IRMS	88
13.1.45	ADCUI_F1IRMS2	89
13.1.46	ADCUI_F1STAT	89
13.1.47	ADCUI_F1MASK	91
13.1.48	ADCUI_F2CTR	92
13.1.49	ADCUI_F2WC	93
13.1.50	ADCUI_F2WATTP	93
13.1.51	ADCUI_F2WATTN	94
13.1.52	ADCUI_F2VC	94
13.1.53	ADCUI_F2VARP	94
13.1.54	ADCUI_F2VARN	95
13.1.55	ADCUI_F2AC	95

13.1.56	ADCUI_F2VR	96
13.1.57	ADCUI_F2MD0	96
13.1.58	ADCUI_F2MD1	97
13.1.59	ADCUI_F2MD2	97
13.1.60	ADCUI_F2VPEAK	98
13.1.61	ADCUI_F2IPEAK	98
13.1.62	ADCUI_F2VDAT	98
13.1.63	ADCUI_F2IDAT	99
13.1.64	ADCUI_F2VRMS	99
13.1.65	ADCUI_F2VRMS2	99
13.1.66	ADCUI_F2IRMS	100
13.1.67	ADCUI_F2IRMS2	100
13.1.68	ADCUI_F2STAT	100
13.1.69	ADCUI_F2MASK	102
13.1.70	ADCUI_CCAL1	103
13.1.71	ADCUI_CCAL2	103
13.1.72	ADCUI_CCAL3	104
13.1.73	ADCUI_CCAL4	104
13.1.74	ADCUI_F0IRMS_INACTIVE	104
13.1.75	ADCUI_F0WATTP_L	105
13.1.76	ADCUI_F0WATTN_L	105
13.1.77	ADCUI_F0VARP_L	106
13.1.78	ADCUI_F0VARN_L	106
13.1.79	ADCUI_F0VR_L	106
13.1.80	ADCUI_F1WATTP_L	107
13.1.81	ADCUI_F1WATTN_L	107
13.1.82	ADCUI_F1VARP_L	107
13.1.83	ADCUI_F1VARN_L	108
13.1.84	ADCUI_F1VR_L	108
13.1.85	ADCUI_F2WATTP_L	109
13.1.86	ADCUI_F2WATTN_L	109
13.1.87	ADCUI_F2VARP_L	109
13.1.88	ADCUI_F2VARN_L	110
13.1.89	ADCUI_F2VR_L	110
13.1.90	ADCUI_LOAD	110
13.1.91	ADCUI_F0VRMS_TRUE	111
13.1.92	ADCUI_F1VRMS_TRUE	112
13.1.93	ADCUI_F2VRMS_TRUE	112
13.1.94	ADCUI_MSC	112
13.1.95	ADCUI_FILTERCFG	113
13.1.96	ADCUI_CLKPHASE1	115
13.1.97	ADCUI_CLKPHASE2	115
14	Алгоритмы вычисления окончечных результатов и их соответствия внешним сигналам	117
14.1	Типовая схема включения для учета электроэнергии по трем фазам	123
14.2	Типовая схема включения для учета электроэнергии по одной фазе	125
14.3	Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта	126

15	Аппаратный блок вычисления CRC .....	127
15.1	Описание регистров управления блока CRC.....	127
15.1.1	CRC_CTRL .....	127
15.1.2	CRC_STAT .....	128
15.1.3	CRC_DATA1.....	129
15.1.4	CRC_VAL .....	129
15.1.5	CRC_POL.....	129
16	Сигналы тактовой частоты .....	130
16.1	Встроенный RC-Генератор HSI .....	130
16.2	Встроенный RC-генератор LSI .....	130
16.3	Внешний осциллятор HSE.....	131
16.4	Внешний осциллятор LSE .....	131
16.5	Встроенный блок умножения системной тактовой частоты .....	131
16.6	Описание регистров блока контроллера тактовой частоты .....	132
16.6.1	CLOCK_STATUS.....	132
16.6.2	PLL_CONTROL .....	133
16.6.3	HS_CONTROL .....	134
16.6.4	CPU_CLOCK .....	134
16.6.5	PER1_CLOCK .....	135
16.6.6	ADC_CLOCK .....	136
16.6.7	RTC_CLOCK.....	137
16.6.8	PER2_CLOCK .....	138
16.6.9	DMA_DONE_STICK .....	139
16.6.10	TIM_CLOCK .....	139
16.6.11	UART_CLOCK.....	141
16.6.12	SSP_CLOCK.....	142
16.6.13	DIV_SYS_TIM.....	143
17	Батарейный домен и часы реального времени .....	144
17.1	Часы реального времени.....	144
17.2	Память регистров и криптографический ключей .....	146
17.3	Описание регистров блока батарейного домена .....	146
17.3.1	BKP_MEM (128x32) .....	147
17.3.2	BKP_WPR.....	147
17.3.3	BKP_LDO .....	148
17.3.4	BKP_CLK.....	149
17.3.5	BKP_RTC.....	150
17.3.6	RTC_WUTR.....	151
17.3.7	RTC_PREDIV_S.....	152
17.3.8	RTC_PRL .....	152
17.3.9	RTC_ALARM.....	153
17.3.10	RTC_CS.....	153
17.3.11	RTC_TR .....	155
17.3.12	RTC_DR.....	155
17.3.13	RTC_ALRMAR, RTC_ALRMBR.....	156
17.3.14	RTC_TSTR1-RTC_TSTR3, RTC_TSDR1-RTC_TSDR3 .....	157
17.3.15	RTC_TAFCR.....	157
17.3.16	RTC_TMPCAL1 .....	159
17.3.17	RTC_TMPCAL2 .....	159

17.3.18	RTC_TMPCAL3 .....	160
18	Порты ввода-вывода .....	161
18.1	Описание регистров портов ввода-вывода .....	163
18.1.1	PORTx_RXTX .....	164
18.1.2	PORTx_OE.....	164
18.1.3	PORTx_FUNC .....	164
18.1.4	PORTx_ANALOG .....	165
18.1.5	PORTx_PULL .....	165
18.1.6	PORTx_PWR .....	166
18.1.7	PORTx_SETTX.....	166
18.1.8	PORTx_CLRTX.....	167
19	Детектор напряжения питания.....	168
19.1	Описание регистров блока PVD .....	169
19.1.1	PVDCS .....	169
20	Таймеры общего назначения.....	172
20.1	Основные характеристики.....	172
20.1.1	Структурная схема.....	173
20.2	Базовый блок таймера.....	174
20.2.1	Инициализация тактирования таймера .....	174
20.2.2	Инициализация основного счетчика таймера.....	174
20.2.3	Режимы счета .....	175
20.2.4	Тактовая частота $F_{DTS}$ .....	177
20.3	Источники событий для счета.....	178
20.3.1	Внутренний тактовый сигнал (TIM_CLKd).....	179
20.3.2	Событие в другом таймере (CNT==ARR).....	180
20.3.3	Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CHy1 .....	181
20.3.4	Внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR .....	183
20.4	Режим захвата .....	184
20.5	Режим ШИМ.....	185
20.5.1	Генератор опорного сигнала REF .....	186
20.5.2	Генератор «мертвой зоны» .....	188
20.5.3	Выходные блоки .....	189
20.6	Блок цифрового фильтра .....	190
20.7	Флаги состояний, прерывания и запросы DMA.....	192
20.7.1	Флаги состояний .....	192
20.7.2	Прерывания .....	192
20.7.3	Запросы DMA.....	192
20.8	Примеры .....	193
20.8.1	Обычный счетчик .....	193
20.8.2	Режим захвата .....	193
20.8.3	Режим ШИМ .....	194
20.9	Описание регистров блока таймера.....	196
20.9.1	CNT .....	197
20.9.2	PSG .....	197
20.9.3	ARR .....	197
20.9.4	CNTRL .....	198

20.9.5	CHy_CCR.....	199
20.9.6	CHy_CCR1.....	199
20.9.7	CHy_CNTRL0.....	200
20.9.8	CHy_CNTRL1.....	202
20.9.9	CHy_CNTRL2.....	203
20.9.10	CHy_DTG.....	204
20.9.11	BRKETR_CNTRL .....	204
20.9.12	STATUS .....	205
20.9.13	IE.....	207
20.9.14	DMA_RE .....	209
21	Контроллер АЦП.....	211
21.1	Преобразование внешнего канала .....	212
21.2	Последовательное преобразование нескольких каналов .....	213
21.3	Преобразование с контролем границ .....	213
21.4	Датчик температуры .....	214
21.4.1	Формула расчета температуры.....	214
21.5	Время заряда внутренней емкости .....	214
21.6	Время заряда внутренней емкости АЦП и время преобразования .....	215
21.7	Описание регистров блока контроллера АЦП .....	217
21.7.1	ADC1_CFG.....	217
21.7.2	ADC1_H_LEVEL .....	219
21.7.3	ADC1_L_LEVEL.....	219
21.7.4	ADC1_RESULT.....	219
21.7.5	ADC1_STATUS.....	220
21.7.6	ADC1_CHSEL .....	221
21.7.7	ADC_TRIM.....	221
22	Контроллер интерфейса I2C.....	222
22.1	Конфигурация системы .....	222
22.2	Протокол I2C .....	222
22.2.1	Сигнал START .....	223
22.2.2	Передача адреса .....	223
22.2.3	Передача данных .....	223
22.2.4	Сигнал STOP .....	224
22.3	Описание регистров контроллера I2C.....	224
22.3.1	PRL .....	224
22.3.2	PRH.....	225
22.3.3	CTR.....	225
22.3.4	RXD .....	226
22.3.5	STA.....	226
22.3.6	TXD .....	227
22.3.7	CMD .....	227
23	Контроллер SSP .....	229
23.1	Основные характеристики модуля SSP.....	229
23.1.1	Программируемые параметры.....	230
23.1.2	Характеристики интерфейса SPI.....	231
23.1.3	Характеристики интерфейса Microwire.....	231
23.1.4	Характеристики интерфейса SSI.....	231
23.2	Общий обзор модуля SSP .....	231

23.2.1	Блок формирования тактового сигнала.....	232
23.2.2	Буфер FIFO передатчика.....	232
23.2.3	Буфер FIFO приемника .....	233
23.2.4	Блок приема и передачи данных .....	233
23.2.5	Блок формирования прерываний .....	233
23.3	Интерфейс прямого доступа к памяти .....	234
23.4	Конфигурирование приемопередатчика .....	234
23.5	Разрешение работы приемопередатчика.....	234
23.6	Соотношения между тактовыми сигналами.....	234
23.7	Программирование регистра управления CR0.....	235
23.8	Программирование регистра управления CR1 .....	236
23.9	Формирование тактового сигнала обмена данными.....	236
23.10	Формат информационного кадра .....	236
23.10.1	Формат синхронного обмена SSI фирмы Texas Instruments .....	237
23.10.2	Формат синхронного обмена SPI фирмы Motorola .....	238
23.10.3	Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0...239	
23.10.4	Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1...240	
23.10.5	Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0...241	
23.10.6	Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1...242	
23.10.7	Формат синхронного обмена Microwire фирмы National Semiconductor .....	243
23.11	Примеры конфигурации модуля в ведущем и ведомом режимах .....	246
23.12	Интерфейс прямого доступа к памяти.....	249
23.13	Программное управление модулем.....	251
23.13.1	Общая информация .....	251
23.13.2	Описание регистров контроллера SSP .....	251
23.14	Прерывания.....	258
23.14.1	SSPRXINTR.....	259
23.14.2	SSPTXINTR.....	259
23.14.3	SSPRORINTR .....	259
23.14.4	SSPRTINTR .....	259
23.14.5	SSPRNEINTR .....	260
23.14.6	SSPTFEINTR .....	260
23.14.7	SSPTNBSYINTR .....	260
23.14.8	SSPINTR .....	260
24	Контроллер UART.....	261
24.1	Основные характеристики модуля UART .....	261
24.2	Программируемые параметры .....	262
24.3	Отличия от контроллера UART 16C650 .....	262
24.4	Функциональные возможности .....	263
24.5	Описание функционирования блока UART .....	265
24.5.1	Генератор тактового сигнала приемопередатчика.....	265
24.5.2	Буфер FIFO передатчика.....	265
24.5.3	Буфер FIFO приемника .....	266
24.5.4	Блок передатчика .....	266
24.5.5	Блок приемника .....	266
24.5.6	Блок формирования прерываний .....	266
24.5.7	Интерфейс прямого доступа к памяти.....	267

24.5.8	Блок и регистры синхронизации .....	267
24.6	Описание функционирования ИК кодека IrDA SIR .....	267
24.6.1	Кодер ИК передатчика .....	267
24.6.2	Декодер ИК приемника .....	268
24.7	Описание работы UART .....	268
24.7.1	Сброс модуля .....	268
24.7.2	Тактовые сигналы .....	269
24.7.3	Работа универсального асинхронного приемопередатчика .....	269
24.7.4	Коэффициент деления частоты .....	270
24.7.5	Передача и прием данных .....	271
24.7.6	Биты ошибки .....	272
24.7.7	Бит переполнения буфера .....	272
24.7.8	Запрет буфера FIFO .....	272
24.7.9	Работа кодека ИК обмена данными IrDA SIR .....	273
24.8	Линии управления модемом .....	274
24.8.1	Аппаратное управление потоком данных .....	275
24.8.2	Управление потоком данных по линии RTS .....	276
24.8.3	Управление потоком данных по линии CTS .....	276
24.9	Интерфейс прямого доступа к памяти .....	276
24.10	Прерывания .....	279
24.10.1	UARTMSINTR .....	280
24.10.2	UARTRXINTR .....	280
24.10.3	UARTTXINTR .....	281
24.10.4	UARTRTINTR .....	281
24.10.5	UARTEINTR .....	281
24.10.6	UARTINTR .....	282
24.11	Программное управление модулем .....	282
24.12	Описание регистров контроллера UART .....	282
24.12.1	Регистр данных DR .....	283
24.12.2	Регистр состояния приемника / сброса ошибки RSR_ECR .....	284
24.12.3	Регистр флагов FR .....	285
24.12.4	Регистр управления ИК обменом в режиме пониженного энергопотребления ILPR .....	286
24.12.5	Регистр целой части делителя скорости передачи данных IBRD .....	287
24.12.6	Регистр дробной части делителя скорости передачи данных FBRD .....	287
24.12.7	Регистр управления линией LCR_N .....	289
24.12.8	Регистр управления CR .....	291
24.12.9	Регистр порога прерывания по заполнению буфера FIFO IFLS .....	293
24.12.10	Регистр установки сброса маски прерывания IMSC .....	294
24.12.11	Регистр состояния прерываний RIS .....	295
24.12.12	Регистр маскированного состояния прерываний MIS .....	296
24.12.13	Регистр сброса прерываний ICR .....	298
24.12.14	Регистр управления прямым доступом к памяти DMACR .....	299
24.12.15	Регистр управления тестированием TCR .....	299
25	Контроллер прямого доступа в память DMA .....	300
25.1	Основные свойства контроллера DMA .....	300
25.2	Термины и определения .....	300
25.3	Функциональное описание .....	302

25.3.1	Распределение каналов DMA .....	302
25.3.2	Блок, подключенный к шине APB .....	303
25.3.3	Блок, подключенный к шине AHB .....	303
25.3.4	Управляющий блок DMA .....	303
25.3.5	Типы передач .....	304
25.3.6	Разрядность передаваемых данных .....	304
25.3.7	Управление защитой данных.....	304
25.3.8	Инкремент адреса .....	305
25.4	Управление DMA .....	306
25.4.1	Правила обмена данными .....	306
25.4.2	Диаграммы работы контроллера DMA .....	308
25.4.3	Правила арбитража DMA .....	313
25.4.4	Приоритет .....	314
25.4.5	Типы циклов DMA .....	316
25.5	Структура управляющих данных канала .....	328
25.5.1	Указатель конца данных источника.....	332
25.5.2	Указатель конца данных приемника.....	333
25.5.3	Разряды управления .....	333
25.6	Описание регистров контроллера DMA .....	339
25.6.1	Статусный регистр DMA STATUS .....	341
25.6.2	Регистр конфигурации DMA CFG .....	342
25.6.3	Регистр базового адреса управляющих данных каналов CTRL_BASE_PTR.....	342
25.6.4	Регистр базового адреса альтернативных управляющих данных каналов ALT_CTRL_BASE_PTR.....	343
25.6.5	Регистр статуса ожидания запроса на обработку каналов WAITONREQ_STATUS .....	343
25.6.6	Регистр программного запроса на обработку каналов CHNL_SW_REQUEST.....	344
25.6.7	Регистр установки пакетного обмена каналов CHNL_USEBURST_SET .....	345
25.6.8	Регистр сброса пакетного обмена каналов CHNL_USEBURST_CLR.....	346
25.6.9	Регистр маскирования запросов на обслуживание каналов CHNL_REQ_MASK_SET .....	347
25.6.10	Регистр очистки маскирования запросов на обслуживание каналов CHNL_REQ_MASK_CLR .....	348
25.6.11	Регистр установки разрешения каналов CHNL_ENABLE_SET .....	348
25.6.12	Регистр сброса разрешения каналов CHNL_ENABLE_CLR .....	349
25.6.13	Регистр установки первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_SET .....	350
25.6.14	Регистр сброса первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_CLR.....	351
25.6.15	Регистр установки приоритета каналов CHNL_PRIORITY_SET .....	352
25.6.16	Регистр сброса приоритета каналов CHNL_PRIORITY_CLR .....	352
25.6.17	Регистр сброса флага ошибки ERR_CLR.....	353
26	Прерывания.....	354
27	Контроллер обработки локальных прерываний CLIC.....	355

27.1	Приоритет прерывания .....	355
27.2	Взаимодействие контроллера CLIC с другими локальными прерываниями ..	355
27.3	Описание регистров контроллера CLIC.....	356
27.3.1	Конфигурационный регистр cliccfg.....	356
27.3.2	Информационный регистр clicinfo.....	360
27.3.3	Регистр ожидания прерывания clicintip.....	361
27.3.4	Регистр разрешения прерываний clicintie .....	361
27.3.5	Регистр атрибутов прерывания clicintattr .....	362
27.3.6	Регистр управления входом прерывания clicintctl .....	363
27.3.7	Регистр триггера прерывания clicinttrig .....	363
27.4	Контрольно-статусные регистры CSR .....	364
27.4.1	Регистры mtvt, utvt.....	365
27.4.2	Регистры mnxti, unxti.....	365
27.4.3	Регистры mintstatus, uintstatus .....	366
27.4.4	Регистры mintthresh, uintthresh .....	367
27.4.5	Регистр mclicbase .....	367
27.4.6	Регистры mscratchswl, uscratchswl.....	368
27.5	Карта локальных прерываний.....	368
28	Подсистема отладки .....	370
28.1	Блок DTM (Debug Transport Module) .....	370
28.1.1	Регистры DTM .....	371
28.2	Блок DM (Debug Module) .....	372
28.2.1	Регистры DM.....	373
29	Сторожевые таймеры .....	376
29.1	Регистры блока сторожевых таймеров.....	376
29.1.1	Регистры сторожевого таймера IWDG.....	376
29.1.2	Регистры сторожевого таймера WWDG.....	379
30	Генератор случайных чисел RANDOM .....	384
30.1	Назначение и свойства.....	384
30.2	Регистры модуля.....	385
30.2.1	Регистр статуса и управления STAT_CTRL_REG .....	385
30.2.2	Регистр управления прерыванием INT_CTRL_REG .....	386
30.2.3	Регистр делителя клокка генератора CLK_DIV_REG .....	386
30.2.4	Регистр паузы включения PAUSE_REG .....	386
30.2.5	Регистр случайного значения OUTPUT_REG .....	387
30.2.6	Регистр счетчика паузы PAUSE_CNT_REG.....	387
30.2.7	Регистр сбора случайного числа TEMP_REG .....	387
31	Контроллер UART (ISO7816).....	388
31.1	Введение.....	388
31.2	Особенности модуля USART .....	388
31.3	Функциональное описание USART.....	389
31.3.1	Описание символов USART .....	391
31.3.2	Передачик .....	392
31.3.3	Приемник.....	394
31.3.4	Дробный генератор скорости .....	400
31.3.5	Допуск ухода тактовой частоты для приемника USART .....	403
31.3.6	Многопроцессорный обмен.....	403
31.3.7	Контроль четности.....	405

31.3.8	Синхронный режим USART .....	406
31.3.9	Однопроводной полудуплексный обмен данными .....	409
31.3.10	Режим Smartcard .....	410
31.3.11	Аппаратное управление потоком .....	412
31.4	Прерывания USART .....	414
31.5	Регистры USART .....	415
31.5.1	USART_SR .....	416
31.5.2	USART_DR .....	419
31.5.3	USART_BRR .....	420
31.5.4	USART_CR1 .....	421
31.5.5	USART_CR2 .....	424
31.5.6	USART_CR3 .....	426
31.5.7	USART_GTPR .....	428
32	Электрические параметры микросхем .....	429
33	Предельно-допустимые и предельные параметры .....	430
34	Справочные параметры .....	432
35	Габаритный чертеж микросхемы .....	434
36	Информация для заказа .....	435

## 1 Введение

Микросхема относится к серии микроконтроллеров со встроенной Flash-памятью программ и построена на базе процессорного RISC-V ядра BM-310S. Микроконтроллер работает на тактовой частоте до 48 МГц и содержит 512+16 Кбайт Flash-памяти и 112 Кбайт ОЗУ. Микроконтроллер включает в себя развитую периферию для построения счетчиков электроэнергии 1- и 3-фазных сетей. Периферия включает в себя семь каналов для 3-фазной сети (или 3 канала для 1-фазной 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 микроконтроллера входит пять интерфейсов UART, три SSP и один интерфейс I2C. Криптографическая часть микросхемы включает системы команд *xgost*, генератор случайных чисел и защитную сетку. Микроконтроллер содержит четыре 32-разрядных таймера с четырьмя каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Также микроконтроллер 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 микроконтроллеров.

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 микроконтроллера достаточно одного внешнего напряжения питания в диапазоне от 3,0 до 3,6 В. Также в микроконтроллере 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

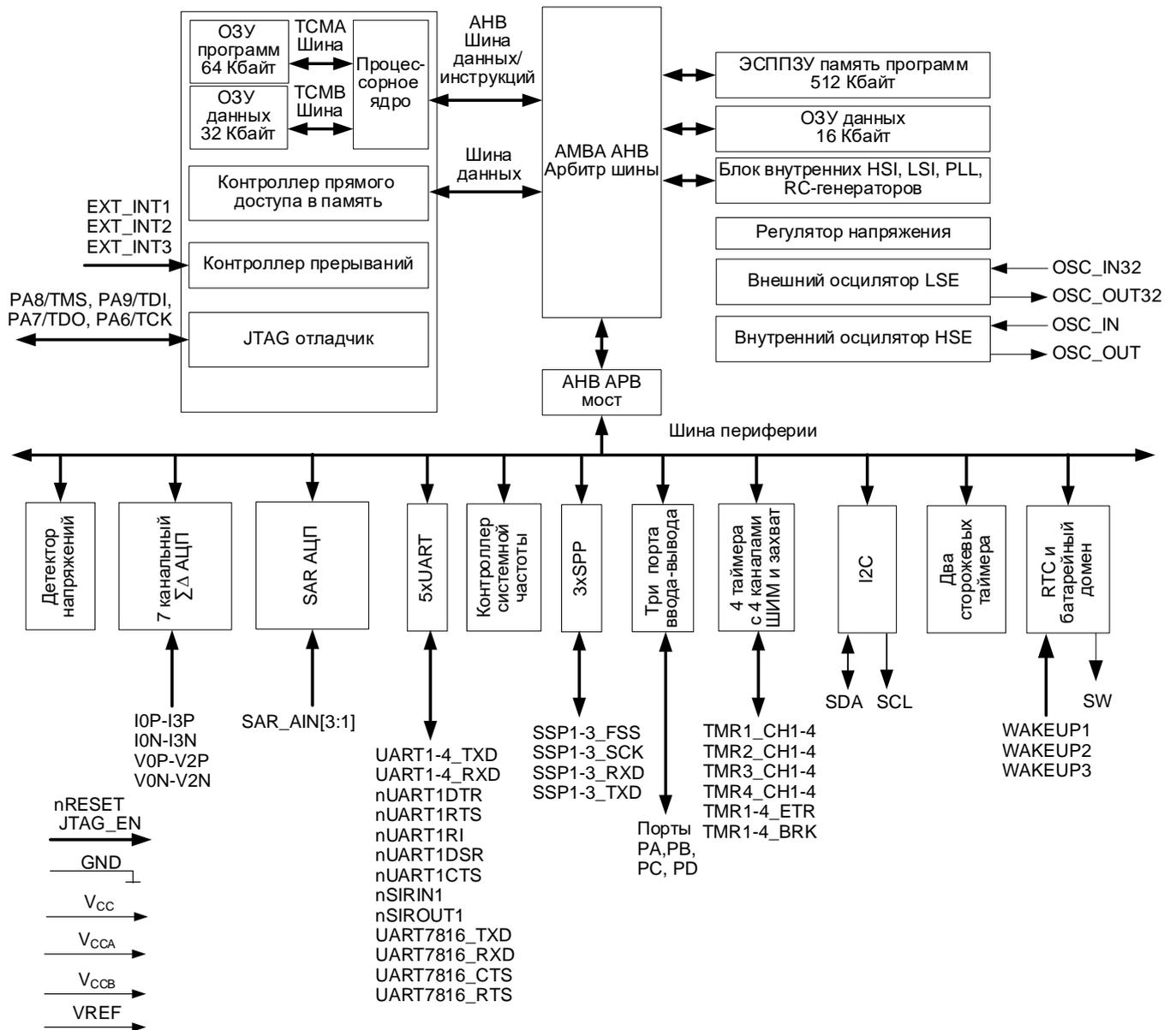


Рисунок 1 – Структурная блок-схема микросхем

### 3 Условное графическое изображение

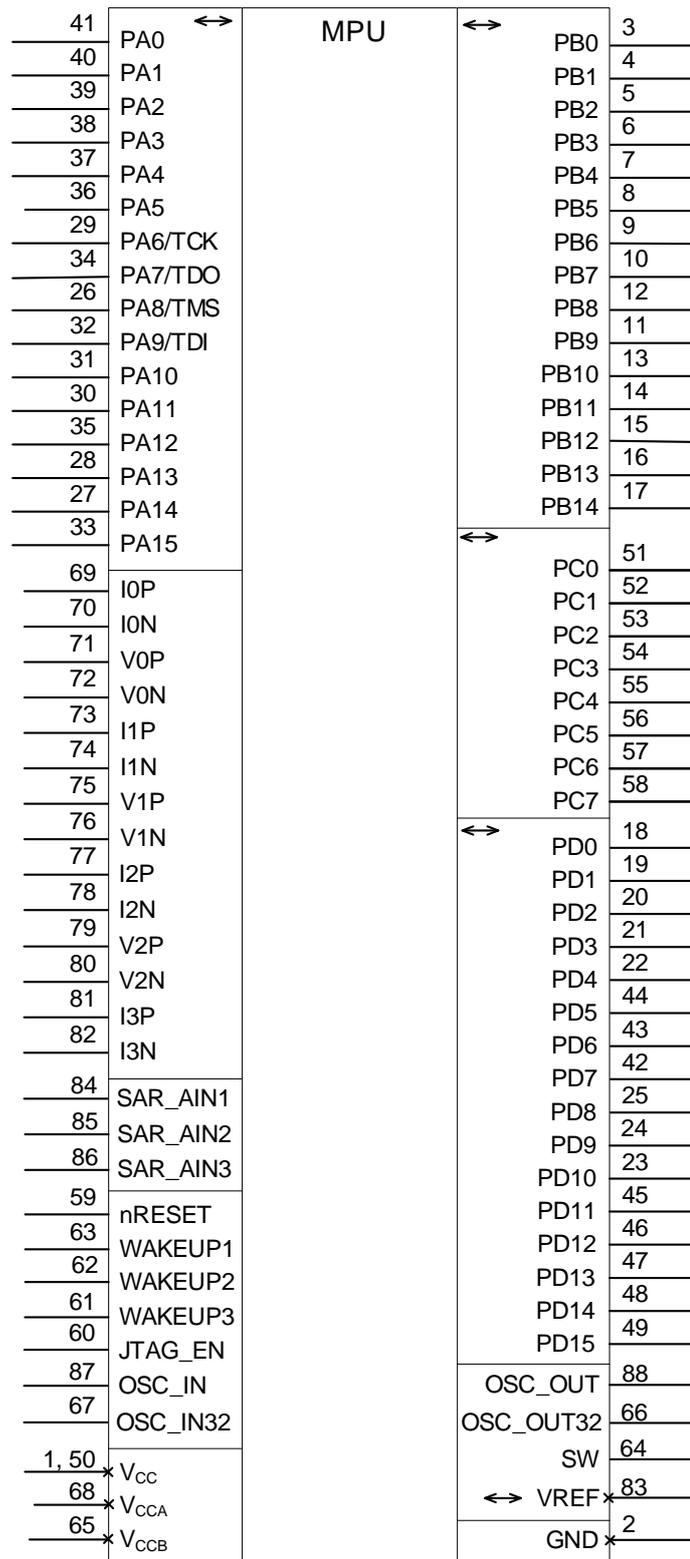


Рисунок 2 – Условное графическое изображение микросхемы

## 4 Диаграмма расположения выводов в корпусе

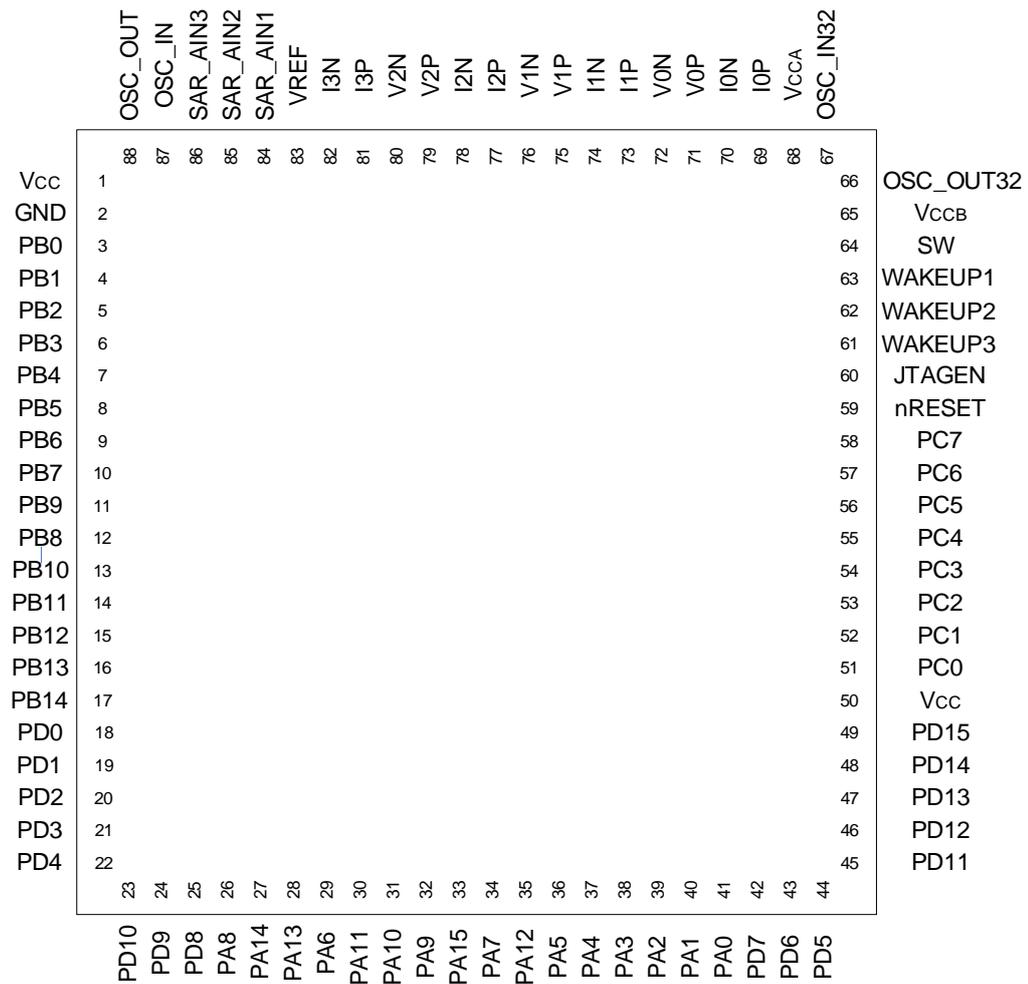


Рисунок 3 – Диаграмма расположения выводов

## 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
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

Номер вывода	Обозначение вывода	Функциональное назначение вывода
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	Порт ввода-вывода C разряд 0
52	PC1	Порт ввода-вывода C разряд 1
53	PC2	Порт ввода-вывода C разряд 2
54	PC3	Порт ввода-вывода C разряд 3
55	PC4	Порт ввода-вывода C разряд 4
56	PC5	Порт ввода-вывода C разряд 5
57	PC6	Порт ввода-вывода C разряд 6
58	PC7	Порт ввода-вывода C разряд 7
59	nRESET	Сигнал внешнего сброса
60	JTAGEN	Вход разрешения отладочного TAP интерфейса на выводах PB[9:6]. Подключить к шине «Общий» в функциональном режиме
61	WAKEUP3	Вход сигнала вскрытия 3
62	WAKEUP2	Вход сигнала вскрытия 2
63	WAKEUP1	Вход сигнала вскрытия 1
64	SW	Выход для управления внешним переключением питания U <sub>CCSV</sub> на U <sub>CC</sub>
65	V <sub>CCB</sub>	Батарейное питание
66	OSC_OUT32	Выход генератора LSE
67	OSC_IN32	Вход генератора LSE
68	V <sub>CCA</sub>	Аналоговое питание ΔΣАЦП и SAR АЦП
69	I0P	Вход канала тока 0 прямой
70	I0N	Вход канала тока 0 инверсный
71	V0P	Вход канала напряжения 0 прямой
72	V0N	Вход канала напряжения 0 инверсный
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

Номер вывода	Обозначение вывода	Функциональное назначение вывода
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	РА0	Порт ввода-вывода А разряд 0		
		Основная	TMR1_CH1	Вход-выход канала 1 таймера 1
		Альтернативная	SSP2_FSS	Вход-выход разрешения SPI 2
		Переопределённая	–	–
40	РА1	Порт ввода-вывода А разряд 1		
		Основная	TMR1_CH1N	Инверсный выход канала 1 таймера 1
		Альтернативная	SSP2_CLK	Вход-выход синхросигнала SPI 2
		Переопределённая	–	–
39	РА2	Порт ввода-вывода А разряд 2		
		Основная	TMR1_CH2	Вход-выход канала 2 таймера 1
		Альтернативная	SSP2_RXD	Входные данные SPI 2
		Переопределённая	–	–
38	РА3	Порт ввода-вывода А разряд 3		
		Основная	TMR1_CH2N	Инверсный выход канала 2 таймера 1
		Альтернативная	SSP2_TXD	Выходные данные SPI 2
		Переопределённая	–	–
37	РА4	Порт ввода-вывода А разряд 4		
		Основная	TMR1_CH3	Вход-выход канала 3 таймера 1
		Альтернативная	–	–
		Переопределённая	–	–
36	РА5	Порт ввода-вывода А разряд 5		
		Основная	TMR1_CH3N	Инверсный выход канала 3 таймера 1
		Альтернативная	–	–
		Переопределённая	–	–
29	РА6/ТСК	Порт ввода-вывода А разряд 6 / Вход синхросигнала JTAG		
		Основная	TMR1_CH4	Вход-выход канала 4 таймера 1
		Альтернативная	–	–
		Переопределённая	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
34	PA7/TDO	Порт ввода-вывода А разряд 7 / Выход данных JTAG		
		Основная	TMR1_CH4N	Инверсный выход канала 4 таймера 1
		Альтернативная	–	–
		Переопределённая	–	–
26	PA8/TMS	Порт ввода-вывода А разряд 8 / Вход управления JTAG		
		Основная	TMR1_ETR	Вход внешнего события таймера 1
		Альтернативная	–	–
		Переопределённая	–	–
32	PA9/TDI	Порт ввода-вывода А разряд 9 / Вход данных JTAG		
		Основная	TMR1_BRK	Вход внешнего события таймера 1
		Альтернативная	–	–
		Переопределённая	–	–
31	PA10	Порт ввода-вывода А разряд 10		
		Основная	EXT_INT1	Вход прерывания 1
		Альтернативная	UART3_RXD	Входные данные UART3
		Переопределённая	TMR4_CH1	Вход-выход канала 1 таймера 4
30	PA11	Порт ввода-вывода А разряд 11		
		Основная	TMR2_CH4N	Инверсный выход канала 4 таймера 2
		Альтернативная	UART3_TXD	Выходные данные UART3
		Переопределённая	TMR4_CH1N	Инверсный выход канала 1 таймера 4
35	PA12	Порт ввода-вывода А разряд 12		
		Основная	SSP1_FSS	Вход-выход разрешения SPI 1
		Альтернативная	–	–
		Переопределённая	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	–	–
		Переопределённая	TMR4_CH2N	Инверсный выход канала 2 таймера 4
27	PA14	Порт ввода-вывода А разряд 14		
		Основная	SSP1_RXD	Входные данные SPI 1
		Альтернативная	–	–
		Переопределённая	TMR4_CH3	Вход-выход канала 3 таймера 4
33	PA15	Порт ввода-вывода А разряд 15		
		Основная	SSP1_TXD	Выходные данные SPI 1
		Альтернативная	–	–
		Переопределённая	TMR4_CH3N	Инверсный выход канала 3 таймера 4

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
Порт В				
3	PB0	Порт ввода-вывода В разряд 0		
		Основная	UART1_TXD	Выходные данные UART1
		Альтернативная	UART7816_TXD	Выходные данные UART7816
		Переопределённая	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
		Переопределённая	TMR3_CH1N	Инверсный выход канала 1 таймера 3
5	PB2	Порт ввода-вывода В разряд 2		
		Основная	UART1_RXD	Входные данные UART1
		Альтернативная	UART7816_RXD	Входные данные UART7816
		Переопределённая	TMR3_CH1N	Инверсный выход канала 1 таймера 3
6	PB3	Порт ввода-вывода В разряд 3		
		Основная	nSIRIN1	Входные данные IRDA UART1
		Альтернативная	UART7816_CTS	Вход управления UART7816
		Переопределённая	TMR3_CH2N	Инверсный выход канала 2 таймера 3
7	PB4	Порт ввода-вывода В разряд 4		
		Основная	nUART1DTR	Выход управления UART1
		Альтернативная	–	–
		Переопределённая	TMR3_CH3	Вход-выход канала 3 таймера 3
8	PB5	Порт ввода-вывода В разряд 5		
		Основная	nUART1RTS	Выход управления UART1
		Альтернативная	UART7816_RTS	Выход управления UART7816
		Переопределённая	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
		Переопределённая	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
		Переопределённая	TMR3_CH4N	Инверсный выход канала 4 таймера 3
12	PB8	Порт ввода-вывода В разряд 8		
		Основная	nUART1CTS	Вход управления UART1
		Альтернативная	TMR2_BRK	Вход внешнего события таймера 2
		Переопределённая	TMR3_BRK	Вход внешнего события таймера 3

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
11	PB9	Порт ввода-вывода В разряд 9		
		Основная	nUART1DSR	Вход управления UART1
		Альтернативная	TMR2_ETR	Вход внешнего события таймера 2
		Переопределённая	TMR3_ETR	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
		Переопределённая	–	–
14	PB11	Порт ввода-вывода В разряд 11		
		Основная	TMR2_CH2N	Инверсный выход канала 2 таймера 2
		Альтернативная	UART4_TXD	Выходные данные UART4
		Переопределённая	–	–
15	PB12	Порт ввода-вывода В разряд 12		
		Основная	TMR2_CH3	Вход-выход канала 3 таймера 2
		Альтернативная	–	–
		Переопределённая	–	–
16	PB13	Порт ввода-вывода В разряд 13		
		Основная	TMR2_CH3N	Инверсный выход канала 3 таймера 2
		Альтернативная	UART2_TXD	Выходные данные UART2
		Переопределённая	–	–
17	PB14	Порт ввода-вывода В разряд 14		
		Основная	TMR2_CH4	Вход-выход канала 4 таймера 2
		Альтернативная	UART2_RXD	Входные данные UART2
		Переопределённая	–	–
Порт С				
51	PC0	Порт ввода-вывода С разряд 0		
		Основная	TMR4_ETR	Вход внешнего события таймера 4
		Альтернативная	–	–
		Переопределённая	TMR4_CH4	Вход-выход канала 4 таймера 4
52	PC1	Порт ввода-вывода С разряд 1		
		Основная	TMR4_BRK	Вход внешнего события таймера 4
		Альтернативная	–	–
		Переопределённая	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
		Переопределённая	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
		Переопределённая	SDA	Вход-выход данных I2C

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
55	PC4	Порт ввода-вывода C разряд 4		
		Основная	EXT_INT2	Вход прерывания 2
		Альтернативная	SSP3_RXD	Входные данные SPI 3
		Переопределённая	–	–
56	PC5	Порт ввода-вывода C разряд 5		
		Основная	EXT_INT3	Вход прерывания 3
		Альтернативная	SSP3_TXD	Выходные данные SPI 3
		Переопределённая	–	–
57	PC6	Порт ввода-вывода C разряд 6		
		Основная	TMR2_ETR	Вход внешнего события таймера 2
		Альтернативная	SCL	Выход синхросигнала I2C
		Переопределённая	–	–
58	PC7	Порт ввода-вывода C разряд 7		
		Основная	TMR2_BRK	Вход внешнего события таймера 2
		Альтернативная	SDA	Вход-выход данных I2C
		Переопределённая	–	–
Порт D				
18	PD0	Порт ввода-вывода D разряд 0		
		Основная	SSP2_FSS	Вход-выход разрешения SPI 2
		Альтернативная	SD_V0_EXT	Данные в SINC3 канала V0
		Переопределённая	–	–
19	PD1	Порт ввода-вывода D разряд 1		
		Основная	SSP2_CLK	Вход-выход синхросигнала SPI 2
		Альтернативная	SD_V1_EXT	Данные в SINC3 канала V1
		Переопределённая	–	–
20	PD2	Порт ввода-вывода D разряд 2		
		Основная	SSP2_RXD	Входные данные SPI 2
		Альтернативная	SD_V2_EXT	Данные в SINC3 канала V2
		Переопределённая	–	–
21	PD3	Порт ввода-вывода D разряд 3		
		Основная	SSP2_TXD	Выходные данные SPI 2
		Альтернативная	–	–
		Переопределённая	–	–
22	PD4	Порт ввода-вывода D разряд 4		
		Основная	UART2_TXD	Выходные данные UART2
		Альтернативная	–	–
		Переопределённая	–	–
44	PD5	Порт ввода-вывода D разряд 5		
		Основная	UART2_RXD	Входные данные UART2
		Альтернативная	–	–
		Переопределённая	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
43	PD6	Порт ввода-вывода D разряд 6		
		Основная	UART3_RXD	Входные данные UART3
		Альтернативная	—	—
		Переопределённая	—	—
42	PD7	Порт ввода-вывода D разряд 7		
		Основная	UART3_TXD	Выходные данные UART3
		Альтернативная	—	—
		Переопределённая	—	—
25	PD8	Порт ввода-вывода D разряд 8		
		Основная	UART4_RXD	Входные данные UART4
		Альтернативная	—	—
		Переопределённая	—	—
24	PD9	Порт ввода-вывода D разряд 9		
		Основная	UART4_TXD	Выходные данные UART4
		Альтернативная	—	—
		Переопределённая	—	—
23	PD10	Порт ввода-вывода D разряд 10		
		Основная	UART7816_TXD	Выходные данные UART7816
		Альтернативная	SD_I3_EXT	Данные в SINC3 канала I3
		Переопределённая	—	—
45	PD11	Порт ввода-вывода D разряд 11		
		Основная	UART7816_RXD	Входные данные UART7816
		Альтернативная	SD_I2_EXT	Данные в SINC3 канала I2
		Переопределённая	—	—
46	PD12	Порт ввода-вывода D разряд 12		
		Основная	UART7816_CLK	Выход синхросигнала UART7816
		Альтернативная	SD_I1_EXT	Данные в SINC3 канала I1
		Переопределённая	—	—
47	PD13	Порт ввода-вывода D разряд 13		
		Основная	UART7816_CTS	Вход управления UART7816
		Альтернативная	SD_I0_EXT	Данные в SINC3 канала I0
		Переопределённая	—	—
48	PD14	Порт ввода-вывода D разряд 14		
		Основная	UART7816_RTS	Выход управления UART7816
		Альтернативная	CLK_SD_EXT	Выход синхросигнала для внешних сигма-дельта модуляторов
		Переопределённая	—	—
49	PD15	Порт ввода-вывода D разряд 15		
		Основная	EXT_INT1	Вход прерывания 1
		Альтернативная	LSE_OUT	Выход частоты генератора LSE после деления на RTC_PRL
		Переопределённая	—	—

## 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 общих) к выходам микросхем, неиспользуемым согласно схеме электрической.

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

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<sub>CC</sub>** – основное питание микросхемы, включает питание пользовательских выводов, встроенного регулятора напряжения, PLL, генераторов;

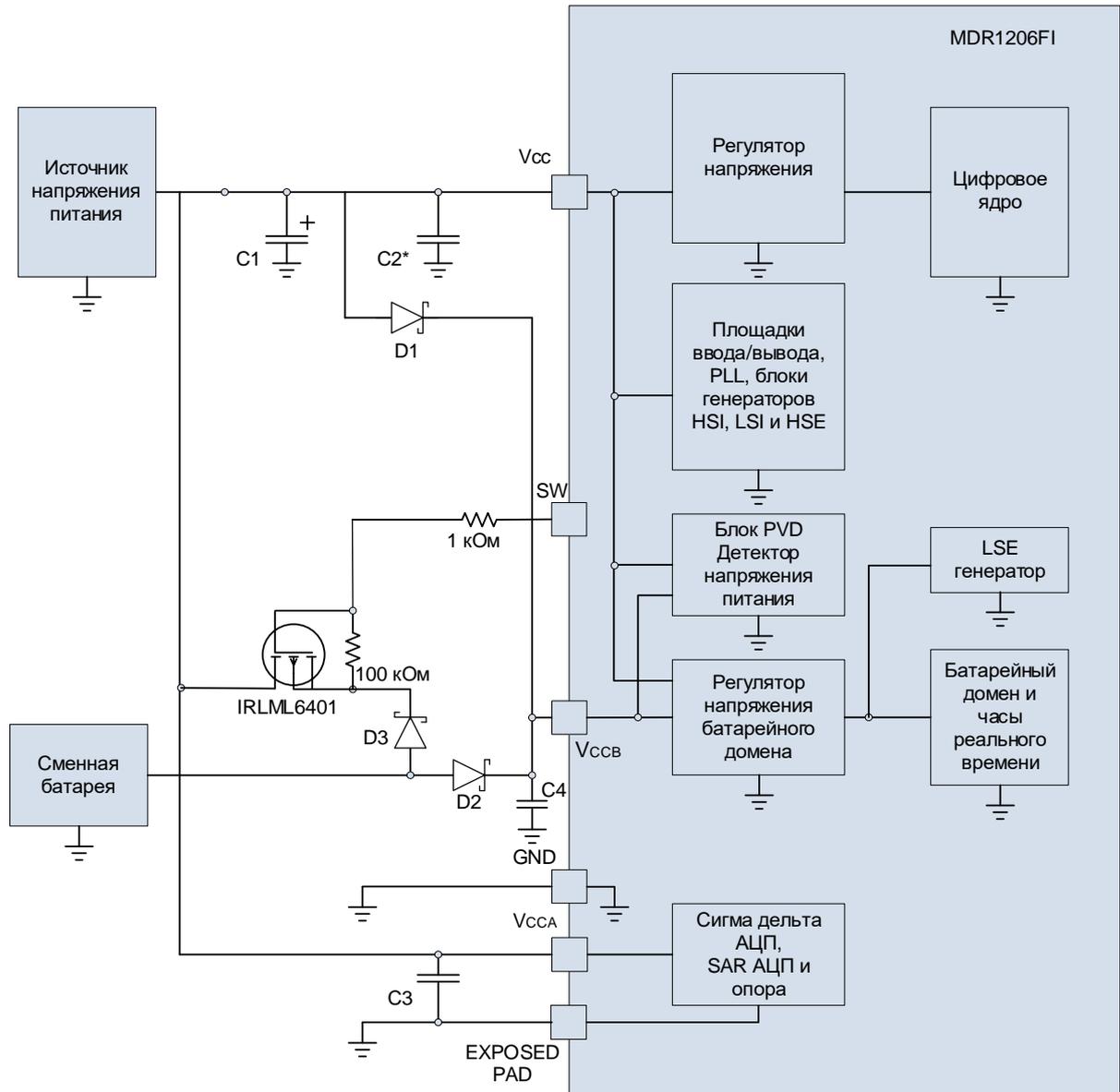
**V<sub>CCB</sub>** – питание батарейного домена используется при отсутствии основного питания U<sub>CC</sub> для питания батарейного домена и LSE генератора. Переключение с основного питания на батарейное происходит автоматически при снижении уровня U<sub>CC</sub> ниже ~1,5 В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 2 мс после превышения уровнем U<sub>CC</sub> значения 1,6 В. Если в системе не требуется батарейное питание, вывод V<sub>CCB</sub> должен быть объединен с V<sub>CC</sub>;

**V<sub>CCA</sub>** – 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<sub>CC</sub>, но при этом на печатной плате должны быть применены меры по снижению наводки помех;

**GND** – основная «земля» питания;

**EXPOSED PAD** – «земля» аналогового питания U<sub>CCA</sub> и цифрового питания U<sub>CC</sub>. Данные выводы должны соединяться с GND.

## 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

### Примечания

1 При отсутствии батарейного питания вывод VCCSV должен быть объединен с VCC. Диоды Шоттки D1, D2 в этом случае могут не устанавливаться;

2 Если используется  $\Sigma\Delta$  АЦП или АЦП последовательного приближения, то напряжение питания UCC (UCCA) должно быть в пределах от 3,0 до 3,6 В.

Микроконтроллер имеет встроенный детектор напряжения питания, подробнее см. раздел «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 1,6 В, сигнал сброса POR удерживается; после превышения данного уровня сигнал POR выдается еще на протяжении  $\sim 2$  мс для того, чтобы гарантированно установилось напряжение питания, после чего сигнал POR снимается, и схема может начать работать.

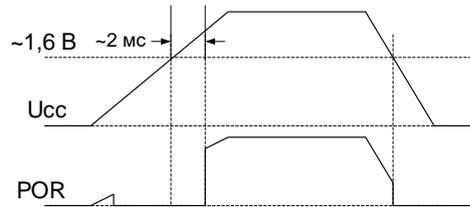


Рисунок 5 – Диаграмма формирования сигнала сброса микроконтроллера

При снижении напряжения питания  $U_{CC}$  ниже уровня 1,5 В сигнал POR вырабатывается без задержки.

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

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

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

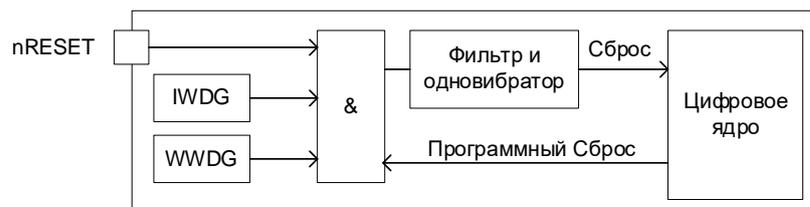


Рисунок 6 – Блок-схема формирования сброса

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.

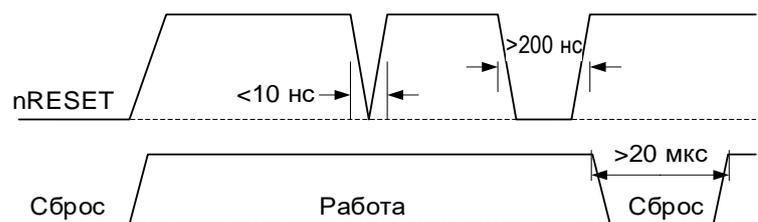


Рисунок 7 – Диаграмма фильтрации помех при формировании сигнала сброса

## 8 Организация памяти

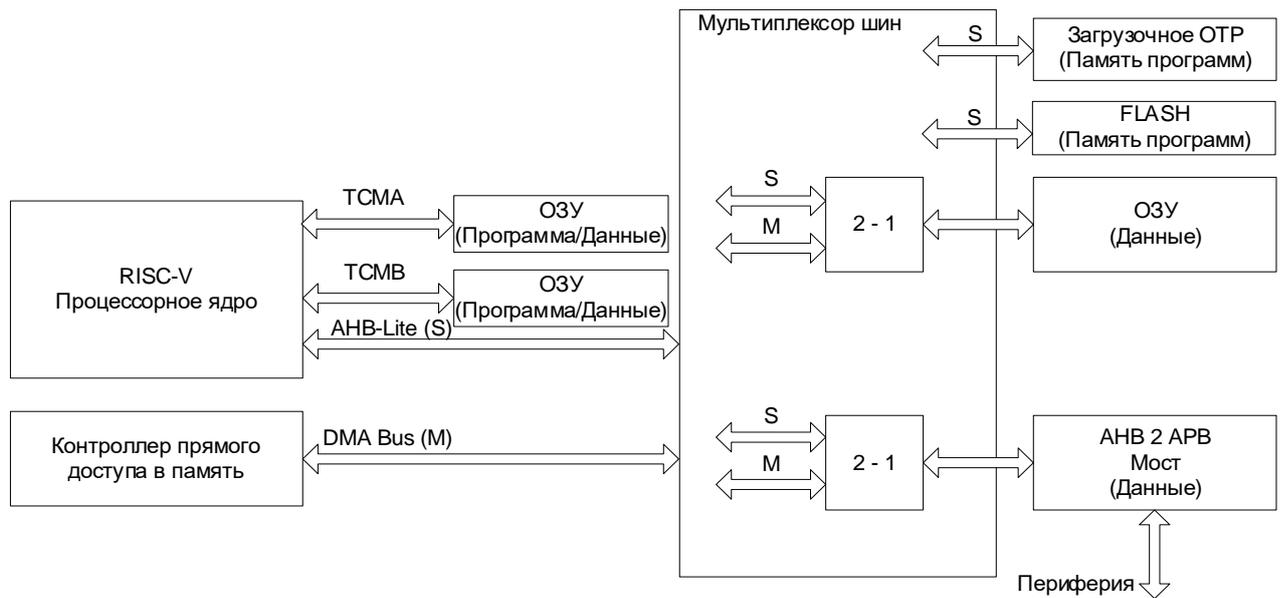


Рисунок 8 – Структурная схема организации памяти

Процессорное ядро имеет две системные шины:

- АHB-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 микроконтроллера разделено на пять основных секций: Debug, CLINT, CLIC, I/O и TCM. В данное адресное пространство отображаются различные модули памяти и периферии.

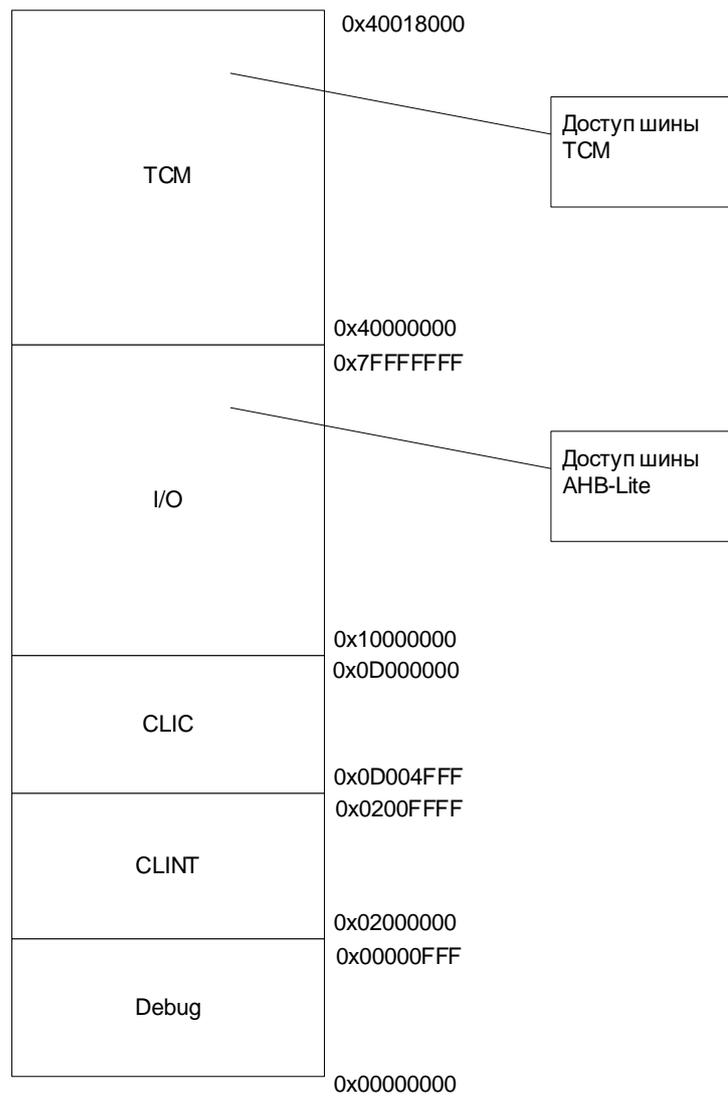


Рисунок 9 – Структура адресного пространства микроконтроллера

## 8.1 Базовые адреса процессора

Таблица 3 – Базовые адреса процессора

Адрес	Размер	Блок	Примечание
Память программ			
0x0002_0000		BOOT	Загрузочная программа
0x1000_0000		FLASH	Область Flash-памяти программ с пользовательской программой
Память данных			
0x2000_0000		SYSTEM RAM	Область внутреннего ОЗУ шины АНВ
Периферия			
0x5000_0000		SSP1	Регистры контроллера интерфейса SSP1
0x5000_8000		UART1	Регистры контроллера интерфейса UART1
0x5001_0000		UART2	Регистры контроллера интерфейса UART2
0x5001_8000		FLASH_CNTRL	Регистры контроллера Flash памяти программ
0x5002_0000		RST_CLK	Регистры контроллера сигналов тактовой частоты
0x5002_8000		DMA	Регистры контроллера прямого доступа в память
0x5003_0000		I2C	Регистры контроллера интерфейса I2C
0x5003_8000		UART3	Регистры контроллера интерфейса UART3
0x5004_0000		ADC	Регистры управления SAR АЦП
0x5004_8000		WWDG	Регистры контроллера сторожевого таймера WWDG
0x5005_0000		IWDG	Регистры контроллера сторожевого таймера IWDG
0x5005_8000		POWER	Регистры детектора напряжения питания
0x5006_0000		BKP	Регистры доступа и управления батарейным доменом
0x5006_8000		ADCUI	Регистры управления $\Delta\Sigma$ АЦП
0x5007_0000		TIMER1	Регистры управления Таймер 1
0x5007_8000		TIMER2	Регистры управления Таймер 2
0x5008_0000		PORTA	Регистры управления порта А
0x5008_8000		PORTB	Регистры управления порта В
0x5009_0000		PORTC	Регистры управления порта С
0x5009_8000		CRC	Регистры управления аппаратного блока вычисления CRC
0x500A_0000		-	-
0x500A_8000		SENSORS	Регистры блока обработки датчиков безопасности
0x500B_0000		CLK_MEASURE	Регистры блока датчика частоты
0x500B_8000		RANDOM	Регистры блока генератора случайных чисел
0x500C_0000		ISO7816	Регистры контроллера интерфейса UART (ISO7816)
0x500C_8000		SSP2	Регистры контроллера интерфейса SSP2
0x500D_0000		SSP3	Регистры контроллера интерфейса SSP3
0x500D_8000		TIMER3	Регистры управления Таймер 3
0x500E_0000		TIMER4	Регистры управления Таймер 4
0x500E_8000		UART4	Регистры контроллера интерфейса UART4
0x500F_0000		PORTD	Регистры управления порта D
SYSTEM REGION			
0x4000_0000		SYSTEM RAM	Область внутреннего ОЗУ шины TCMA
0x4001_0000		SYSTEM RAM	Область внутреннего ОЗУ шины TCMB

## 9 Загрузочное ПО и режимы работы микроконтроллера

После включения питания и снятия внутренних (POR) и внешних (nRESET) сигналов сброса микроконтроллер начинает выполнять программу из загрузочной области BOOT. Загрузочная область – это адресное пространство Flash-памяти.

Ячейки Flash-памяти с адресом 0x0002\_0000 и смещениями с 0x0000\_3FE0 по 0x0000\_3FFF зарезервированы и не могут использоваться для хранения пользовательских данных. Эти регистры содержат идентификационную информацию, калибровочные значения и слово для защиты адресного пространства Flash-памяти.

Таблица 4 – Описание специальных полей в Flash-памяти

Базовый адрес	Смещение	Название	Описание
0x0002_0000	0x0000_3FE0- 0x0000_3FF3	UNIQUE_ID	
	0x0000_3FF4	TRIM_HSI	8 бит – подстройка частоты HSI
	0x0000_3FF5	TRIM_BG	8 бит – подстройка ИОН
	0x0000_3FF6	TRIM_LDO	8 бит – подстройка напряжения LDO
	0x0000_3FF7	TRIM_TS	8 бит – смещение напряжения TS
	0x0000_3FF8	TRIM_CS	Контрольная сумма – биты [7:0] арифметической суммы всех предыдущих 24 байт
	0x0000_3FFC- 0x0000_3FFF	-	Слово для защиты адресного пространства Flash-памяти

Биты защиты считаются активными если в защитное слово записано значение 32'h1234BCxx, где xx – байт который определяют какая область Flash-памяти будет закрыта. Всего 8 областей можно закрыть. Байт FLASH по адресу 0x0002\_3FFC информационной области позволяет установить защищённые адресные пространства Flash-памяти. Установка любого из этих бит запрещает доступ к защищённым адресным пространствам FLASH и блокирует отладочный режим JTAG при выполнении программы из незащищённого адресного пространства FLASH и любого ОЗУ. При выполнении программы из защищённого адресного пространства FLASH, выбранного одним или несколькими нижеописанными битами, доступ к памяти не блокируется. (JTAG остается заблокированным).

Бит 0 адреса 0x0002\_3FFC отвечает за выбор защищённого адресного пространства Flash-памяти с диапазоном адресов 0x10000000-0x1000FFFF.

Бит 1 адреса 0x0002\_3FFC отвечает за выбор защищённого адресного пространства Flash-памяти с диапазоном адресов 0x10010000-0x1001FFFF.

Бит 2 адреса 0x0002\_3FFC отвечает за выбор защищённого адресного пространства Flash-памяти с диапазоном адресов 0x10020000-0x1002FFFF.

Бит 3 адреса 0x0002\_3FFC отвечает за выбор защищённого адресного пространства Flash-памяти с диапазоном адресов 0x10030000-0x1003FFFF.

Бит 4 адреса 0x0002\_3FFC отвечает за выбор защищённого адресного пространства Flash-памяти с диапазоном адресов 0x10040000-0x1004FFFF.

Бит 5 адреса 0x0002\_3FFC отвечает за выбор защищённого адресного пространства Flash-памяти с диапазоном адресов 0x10050000-0x1005FFFF.

Бит 6 адреса 0x0002\_3FFC отвечает за выбор защищённого адресного пространства Flash-памяти с диапазоном адресов 0x10060000-0x1006FFFF.

Бит 7 адреса 0x0002\_3FFC отвечает за выбор защищённого адресного пространства Flash-памяти с диапазоном адресов 0x10070000-0x1007FFFF.

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

В загрузочной программе микроконтроллер определяет, в каком из режимов он будет функционировать, и переходит в этот режим. Режим функционирования определяется внешними выводами MODE[1:0] (PC[0], PC[1]). Также устанавливается бит frog в регистре ВКР\_LDO, который может быть сброшен только при отключении основного питания U<sub>CC</sub>. После перезапуска микроконтроллера уровни на выводах MODE[1:0] не влияют на режим функционирования микроконтроллера, если установлен бит frog, так как в этом случае режим работы будет прочитан из бит mode[1:0] регистра ВКР\_LDO.

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

Таблица 5 – Режимы работы микроконтроллера

MODE[1:0]	Режим	Стартовый адрес/ таблица векторов прерываний	Описание
00	Режим отладки	0x10000000	Процессор начинает выполнять программу из информационной Flash-памяти, а затем передаёт управление во основную Flash-память программ. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в Flash-памяти
01 или 10	UART загрузчик	Определяется пользователем	Микроконтроллер через интерфейс UART1 на выводах PB[1], PB[0] получает код программы в ОЗУ для исполнения
11	Режим отладки	0x40000000	Процессор начинает выполнять программу из информационной Flash-памяти, а затем передаёт управление в ОЗУ память программ. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в Flash-памяти

При работе в режиме отладки разрешается работа отладочного интерфейса JTAG. При этом к микросхеме может быть подключен JTAG-адаптер, с помощью которого

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

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.

### 9.1.1 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, 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.

#### 9.1.1.1 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.

#### 9.1.1.2 Протокол обмена по UART

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

#### 9.1.1.3 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 (3 байта 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>'),).

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

### 9.1.2 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

#### 9.1.2.1 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
Формат команды:	
Master: Выдает код команды CMD_SYNC	Slave: если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды

#### 9.1.2.2 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
Формат команды:	
Master: Выдает код команды CMD_CR	Slave: если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. Выдает код команды CMD_CR. Выдает код 0x0A. Выдает код 0x3E (ASCII символ '>')

### 9.1.2.3 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]
Формат команды:	
Master: Выдает код команды CMD_BAUD	Slave: если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр	Slave: Если параметр принят с ошибками, то выдает код ошибки ERR_CHN или ERR_BAUD и завершает обработку текущей команды. Выдает код команды CMD_BAUD. Устанавливает новое значение скорости обмена

### 9.1.2.4 Команда CMD\_LOAD

Загрузка массива байт в память микроконтроллера.

Таблица 10 – Команда CMD\_LOAD

Код команды	CMD_LOAD = 0x4C
ASCII символ, соответствующий коду команды	'L'
Количество параметров команды	2
Параметр 1	Адрес памяти приемника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master: Выдает код команды CMD_LOAD	Slave: если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр 1	Slave: если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр 2	Slave: если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. Выдает код команды CMD_LOAD
Master: Выдает массив байт младшим байтом вперед	Slave: принимает массив байт. Если хотя бы один байт принят с ошибками, то выдает сообщение об ошибке и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончании принятия массива выдает код ответа REPLY_OK = 0x4B ('K')

### 9.1.2.5 Команда CMD\_VFY

Выдача массива байт из памяти микроконтроллера.

Таблица 11 – Команда CMD\_VFY

Код команды	CMD_VFY = 0x59
ASCII символ, соответствующий коду команды	'Y'
Количество параметров команды	2
Параметр 1	Адрес памяти источника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master: Выдает код команды CMD_VFY	Slave: если команда принята с ошибками, выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр 1	Slave: если хотя бы один из параметров принят с ошибками, выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр 2	Slave: если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. Выдает код команды CMD_VFY. Выдает массив байт младшим байтом вперед. По окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K')

### 9.1.2.6 Команда CMD\_RUN

Запуск программы на выполнение.

Таблица 12 – Команда CMD\_RUN

Код команды	CMD_RUN = 0x52
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр	Адрес таблицы векторов загруженной программы
Формат команды:	
Master: Выдает код команды CMD_RUN	Slave: если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр	Slave: если параметр принят с ошибками, то выдает сообщение об ошибке и завершает обработку текущей команды. Выдает код команды CMD_RUN. Устанавливает значение MSP и PC согласно таблице векторов и, таким образом, Slave завершает свое выполнение

### 9.1.2.7 Прием параметров команды

Параметры команд – это 4-байтные числа.

Параметры передаются младшим байтом вперед.

В качестве значения параметра запрещено использовать число 0xFFFFFFFF.

Если при приеме параметра обнаружена аппаратная ошибка (UART установил в «1» какой-либо из флагов ошибки), то прием параметров не прекращается.

Анализ всех видов ошибок, связанных с передачей параметров, загрузчик производит только после принятия всех параметров команды.

### 9.1.2.8 Сообщения об ошибках

Сообщения об ошибках – это символ 0x45 ('E').

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.

#### **Ошибка ERR\_CHN**

Аппаратная ошибка UART.

Код ошибки 0x69 ('i').

Выдается, если UART установил в '1' один из аппаратных флагов ошибки при приеме очередного байта.

#### **Ошибка ERR\_CMD**

Принята неизвестная команда.

Код ошибки 0x63 ('c').

Выдается диспетчером команд, если принят неизвестный код команды.

#### **Ошибка ERR\_BAUD**

Принята неизвестная команда.

Код ошибки 0x62 ('b').

Выдается диспетчером команд, если по принятому от Master-а значению скорости обмена невозможно вычислить корректное значение делителя частоты UART.

## 10 Контроллер Flash-памяти программ

Возможны два варианта исполнения контроллера Flash-памяти. Далее в разделе приведено описание для варианта CHIP\_ID = 217.

Микросхема содержит встроенную Flash-память программ с объемом 2×256 Кбайт основной памяти программ и 2х8 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 добавляется пауза в один такт процессора для того, чтобы данные успели считаться из Flash-памяти. 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 48 МГц требуется один такт паузы. Число тактов паузы задается в регистре EEPROM\_CMD битами Delay[1:0]. В таблице 13 приведены характеристики необходимой паузы для работы Flash-памяти программ.

Таблица 13 – Характеристики паузы для работы Flash-памяти программ

Delay [1:0]	Тактов паузы	Тактовая частота	Примечание
0x00	0	До 30 МГц	
0x01	1	До 48 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 блока памяти или 2×256 Кбайт;
- стирание страницы памяти размером 4 Кбайт;
- запись 32-битного слова в память;
- чтение 32-битного слова из памяти.

Структура памяти представлена на рисунке 10.

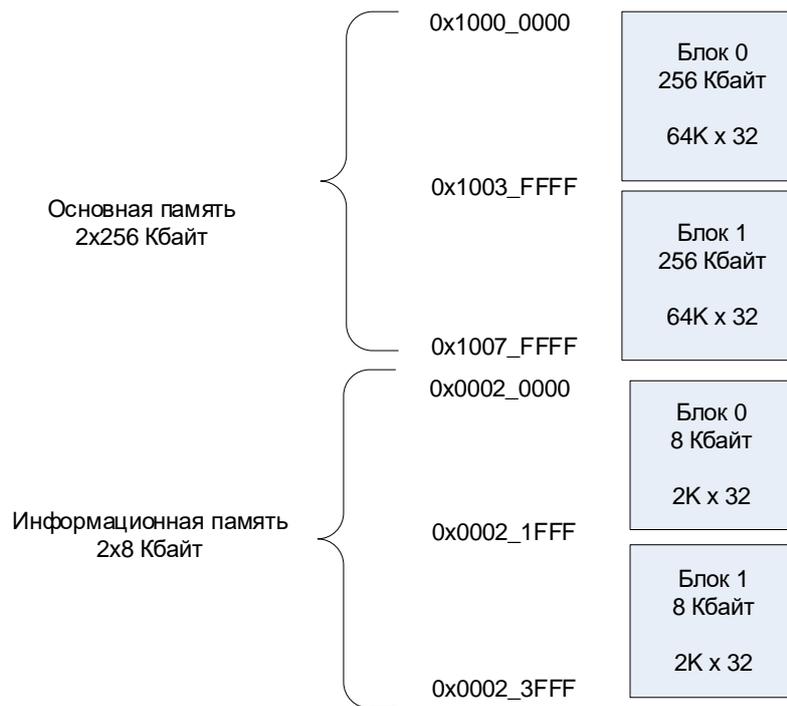


Рисунок 10 – Структура памяти

### 10.2.1 Стирание всей памяти или 2×256 Кбайт основной памяти

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

- установить бит IFREN в необходимое значение:
  - 1 – для всей памяти;
  - 0 – для основной памяти;
- выбрать блок FLASH 0 и 1, выставив ADR[18] и ADR[13].
- затем установить биты XE, MAS1 и ERASE в «1»;
- через время  $T_{nvs} = 5$  мкс установить бит NVSTR в «1»;
- полное стирание памяти длится время  $T_{me} = 40$  мс. Спустя это время необходимо очистить бит ERASE, и спустя время  $T_{nvh1} = 100$  мкс очистить биты XE, MAS1 и NVSTR.
- повторить для второго бака FLASH установив инверсные ADR[18] и ADR[13].

Последующие операции с памятью можно выполнять спустя время  $T_{rcv} = 10$  мкс.

Временная диаграмма стирания памяти представлена на рисунке 11.

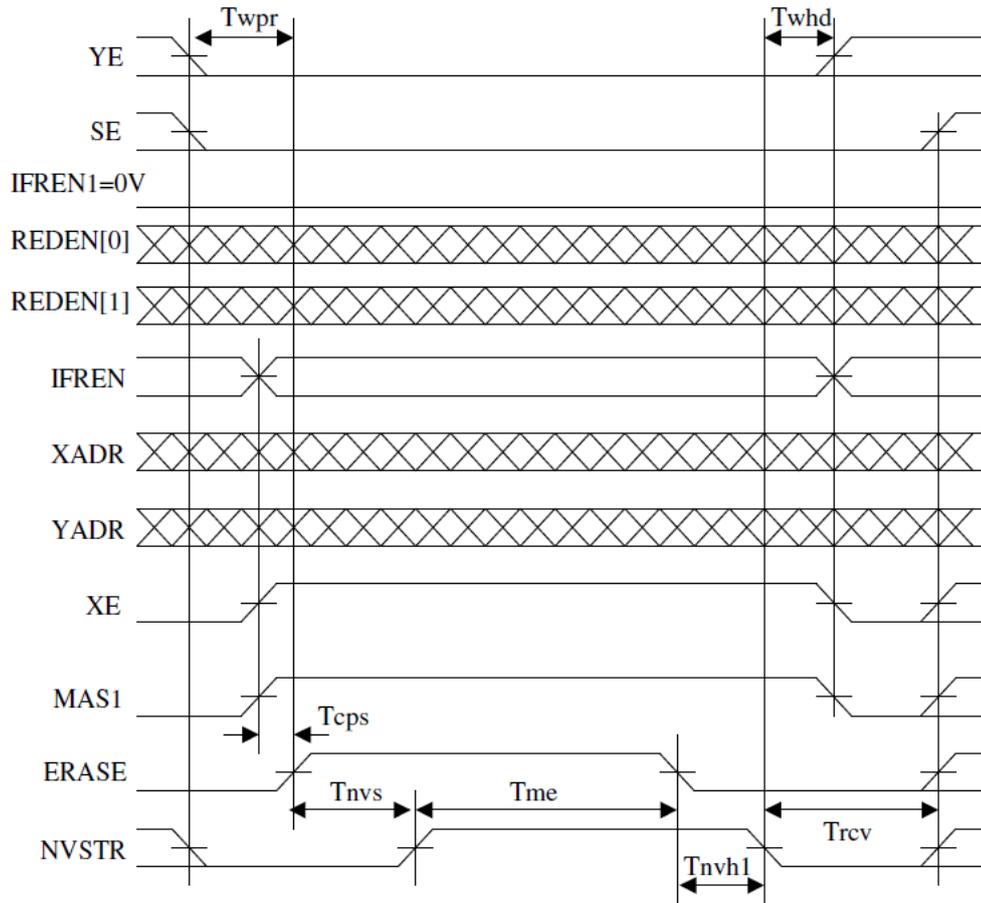


Рисунок 11 – Временная диаграмма стирания памяти

### 10.2.2 Стирание страницы памяти размером 4 Кбайт

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

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

Последующие операции с памятью можно выполнять спустя время  $T_{rcv} = 10$  мкс. Временная диаграмма стирания страницы памяти представлена на рисунке 12.

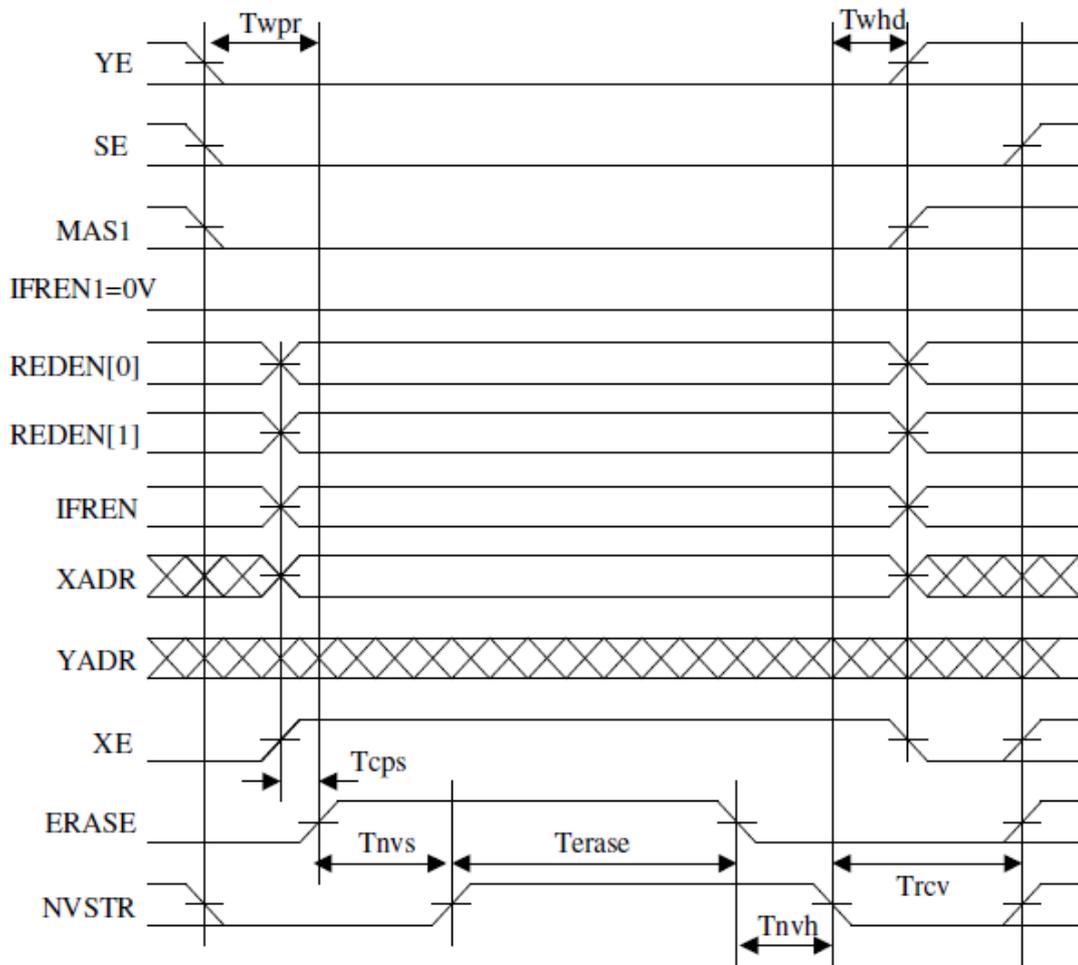


Рисунок 12 – Временная диаграмма стирания страницы памяти

### 10.2.3 Запись 32-битного слова в память

Запись в память возможна только в режиме программирования. Временная диаграмма записи памяти представлена на рисунке 13.

Для записи в память необходимо:

- установить бит IFREN в необходимое значение:
  - 1 – для информационной памяти;
  - 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  $T_{nvs} = 5$  мкс установить бит NVSTR в «1»;
- через время  $T_{cps} = 10$  мкс установить бит YE в «1»;
- запись в память длится время  $T_{prog} = 40$  мкс. Спустя это время необходимо очистить бит YE;
- через время  $T_{adh} = 20$  нс установить новый адрес и значение для записи в другую ячейку памяти;
- через время  $T_{ads} = 20$  нс установить YE в «1» и записать следующее слово;

- если запись больше не требуется:
  - через время  $T_{prgh} = 20$  нс после очистки бита YE необходимо очистить бит PROG;
  - через время  $T_{nvh} = 5$  мкс очистить биты XE и NVSTR.
- последующие операции с памятью можно выполнять спустя время  $T_{rcv} = 10$  мкс.

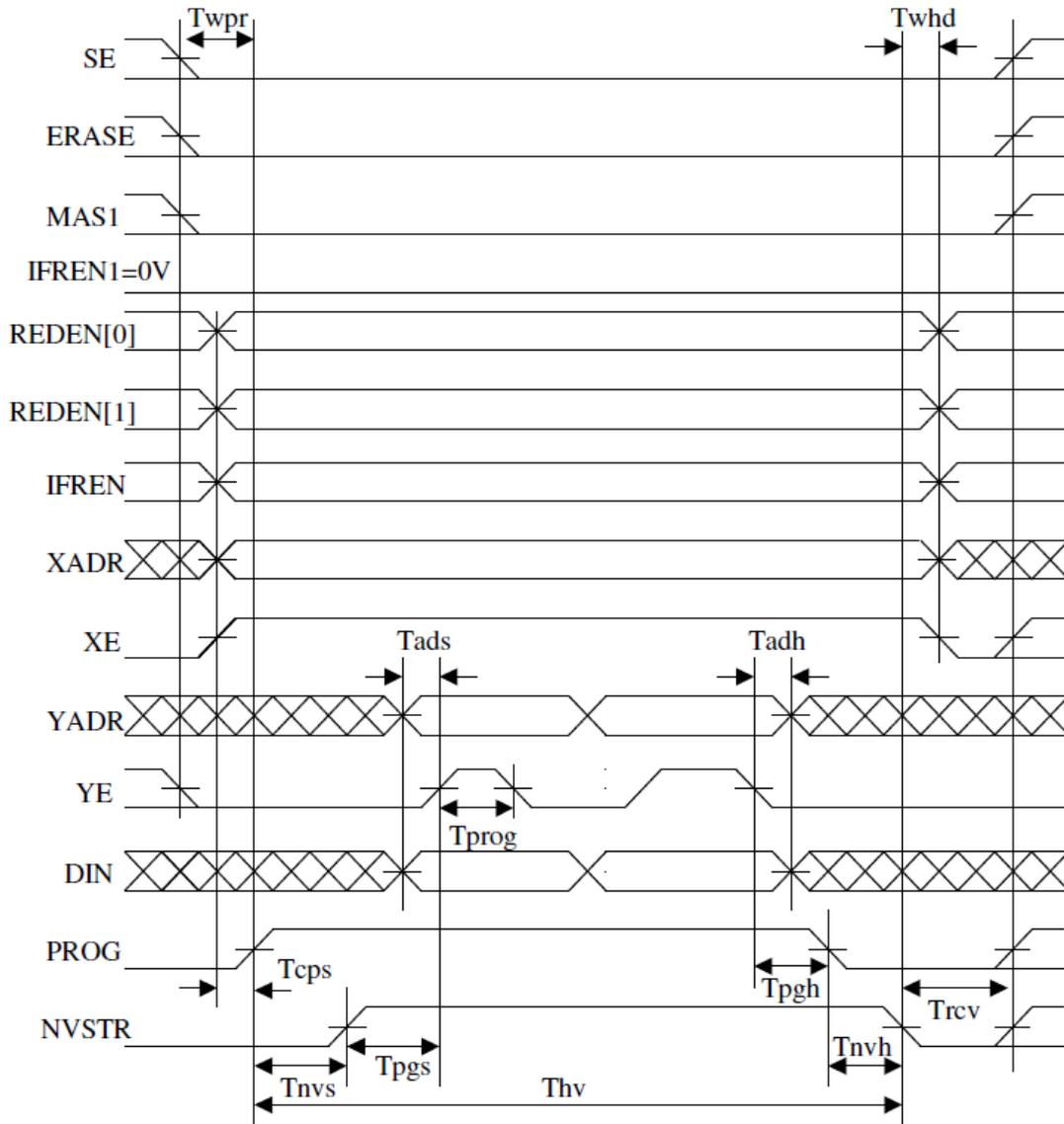


Рисунок 13 – Временная диаграмма записи памяти

#### 10.2.4 Чтение 32-битного слова из памяти

В обычном режиме работы для чтения доступна основная и информационная 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  $T_{acc} = 30$  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  $T_{nws} = 5$  нс, после этого установить SE;
- спустя время  $T_{acc} = 30$  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 14.

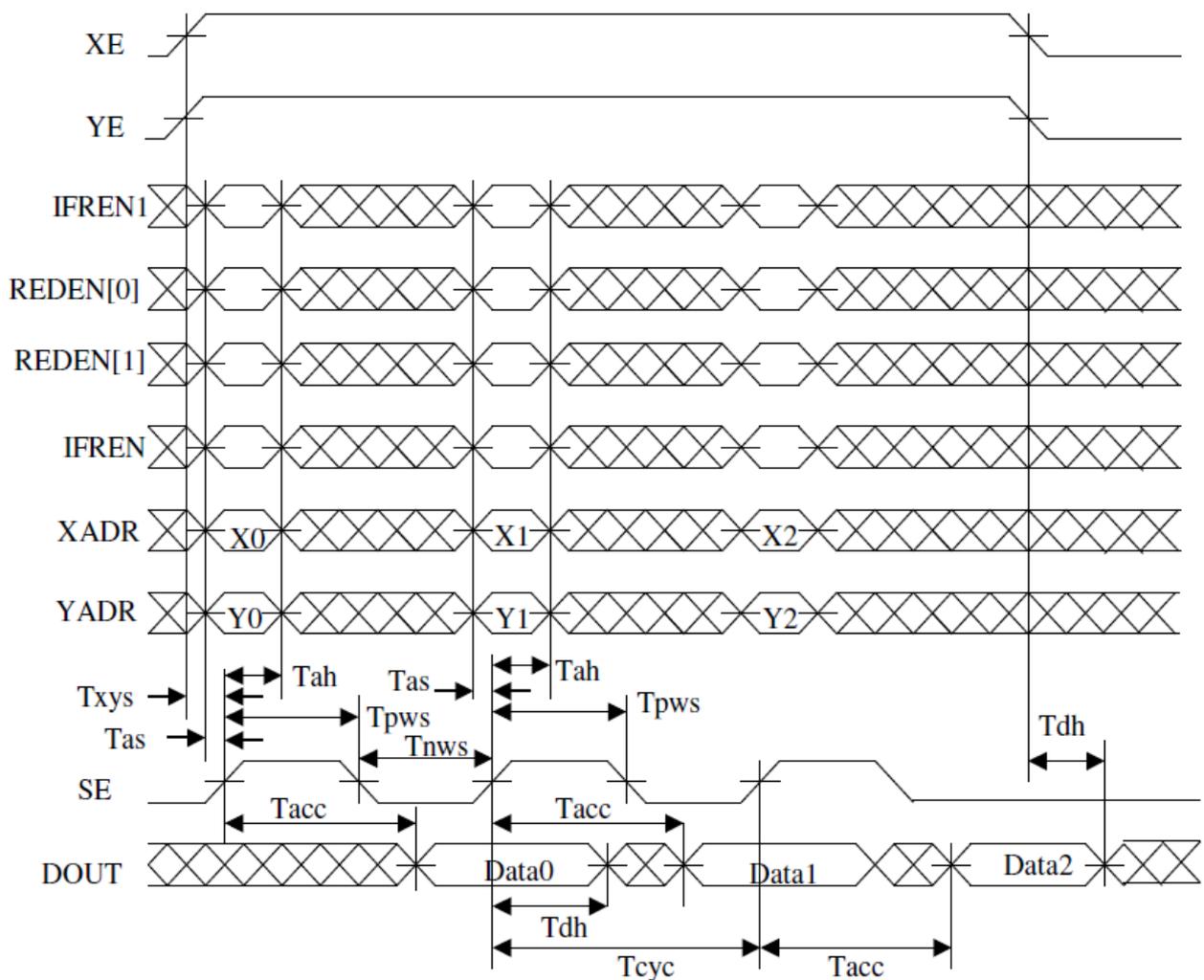


Рисунок 14 – Временная диаграмма чтения памяти

Flash-память программ поддерживает не менее 10000 циклов перезаписи. Нельзя повторять циклы стирания – записи, записи – записи и стирания – стирания одной ячейки памяти с периодом менее 16 мс.

### 10.3 Регистры управления контроллера Flash-памяти программ

Таблица 14 – Регистры управления контроллера Flash-памяти программ

Базовый Адрес	Название	Описание
0x5001_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
0x18	CHIP_ID_CTRL	Регистр CHIP_ID

#### 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-памяти: 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 по ADR[17:2] 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 строку с адресом ADR[17:9], ADR[8:0] значения не имеет: 0 – нет стирания; 1 – стирание
9	IFREN	Работа с блоком информации: 0 – основная память; 1 – информационный блок
8	SE	Усилитель считывания: 0 – не включен; 1 – включен
7	YE	Выдача адреса ADR[8:2]: 0 – не разрешено; 1 – разрешено
6	XE	Выдача адреса ADR[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]	Адрес обращения в память: ADR [18] – выбор блока FLASH; ADR [17:9] – XADR; ADR [8:2] – YADR; ADR [1:0] – не имеет значения. Минимально адресуемая ячейка 32 бита

### 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]	Ключ для разрешения доступа к Flash-памяти через регистровый доступ. Перед переводом памяти в режим программирования необходимо в регистр FLASH_KEY записать комбинацию 0x8AAA5551. После окончания операций записи или стирания Flash-памяти необходимо сбросить ключ для запрещения регистрового доступа

### 10.3.6 CHIP\_ID\_CTRL

Таблица 25 – Регистр CHIP\_ID\_CTRL

Номер	31...10	9...2	1	0
Доступ	U	R	R	R
Сброс	0	CHIP_ID	0/1	0
		CHIP_ID	OTPTST2	OTPTST1

Таблица 26 – Описание бит регистра CHIP\_ID\_CTRL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	-	Зарезервировано
9...2	CHIP_ID	11011001 – ID микросхем версии 217; 11010111 – ID микросхем версии 215
1	OTPTST2	Тестовый бит, может быть в любом состоянии
0	OTPTST1	Тестовый бит

## 11 Система команд

В процессоре реализована система RISC-V команд RV32IMC.

Поддерживаемые команды представлены в таблице 27.

Таблица 27 – 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
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

Мнемокод команды	Формат	Операнды	Краткое описание
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
RDCYCLEH	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
RDTIMEH	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
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
<p>Примечание – В графе Операнды используются следующие обозначения:</p> <ul style="list-style-type: none"> <li>– rs – регистр-источник;</li> <li>– rd – регистр-приёмник;</li> <li>– imm или shamt – непосредственное значение</li> </ul>			

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 подробное описание режимов привилегированности см. 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 стадия генерация запроса в подсистему памяти программ (PMS);
- 2 стадия чтение фрагмента кода из PMS и декодирование команды;
- 3 стадия исполнение команды.

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

Блок FE состоит из следующих подблоков:

- 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 стадии 1 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 стадии 2 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 стадии 3 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, приведенных в таблице 28.

Таблица 28 – 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 регистрам CSR 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 шине АНВ. Доступ к TCM осуществляется с помощью выделенных интерфейсов. Ядро использует последовательную модель доступа (strong ordering memory access) по шине АНВ, т.е. следующий запрос на шине АНВ 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

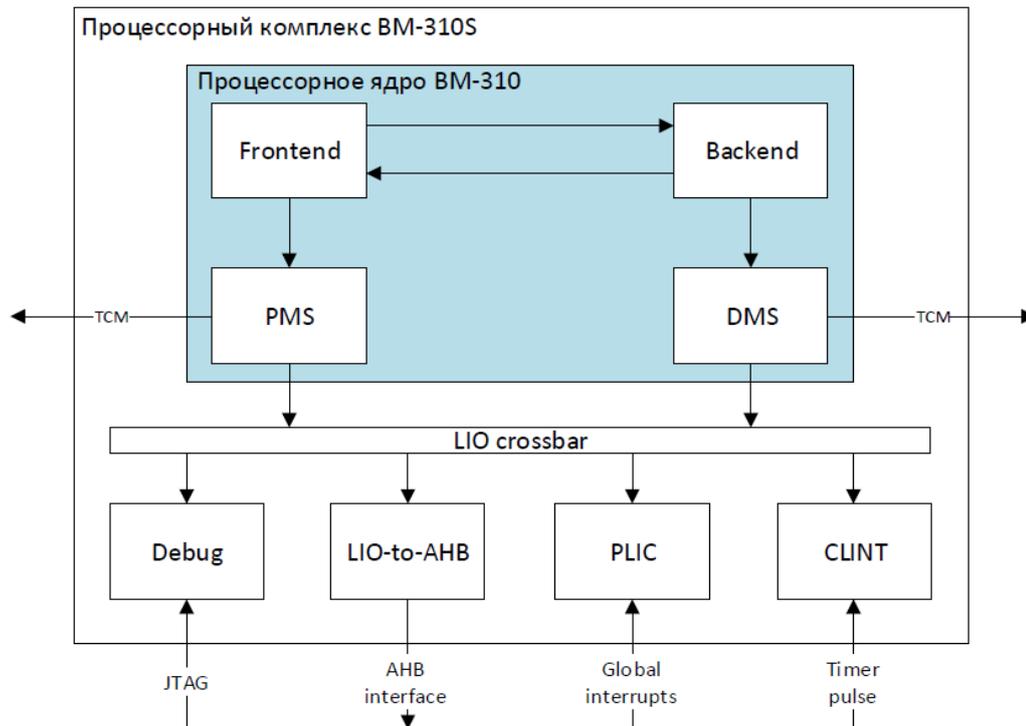


Рисунок 15 – Структурная схема процессора

### 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 128 кБ. 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 неэкшируемой 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 прерываний Local interrupts подключены к контролеру внешних прерываний CLIC. Сигналы, подключаемые к Local interrupts, должны быть синхронны с тактовым сигналом ядра. Запросы прерываний являются edge-sensitive сигналами.

### **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. Подсистема отладки не сбрасывается от сигнала nRESET. Сброс блока Debug осуществляется сигналом POR. 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. Так же каждый АЦП имеет независимый канал DMA, обеспечивая возможность сохранения данных в ОЗУ без участия процессора.

Структурная схема семи каналов АЦП приведена на рисунке 16.

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 ADCIU от HSE = 8,192 МГц, при этом тактовая частота АЦП может быть равна либо 1,024 МГц (выходные частоты отсчетов 1/2/4 кГц в зависимости от настройки OSR\_CONF), либо 2,048 МГц (выходные частоты отсчетов 2/4/8 кГц в зависимости от настройки OSR\_CONF), либо 4,096 МГц (выходные частоты отсчетов 4/8/16 кГц в зависимости от настройки OSR\_CONF) в зависимости от настройки поля CLC\_ADC\_CFG в регистре ADCUI\_CLKPHASE1.

Список параметров, вычисляемых блоком АЦП:

- семь независимых АЦП (четыре канала тока и три канала напряжения) с выходной частотой отсчетов, зависящей от тактовой частоты АЦП, которая настраивается при помощи поля CLC\_ADC\_CFG регистра ADCUI\_CLKPHASE1: выходная частота отсчетов настраивается при помощи поля OSR\_CONF регистра ADCUI\_CTRL1. Эти каналы образуют три блока для измерения параметров каждой фазы F0-F2;
- в блоке каналов F0 реализована опция выбора канала тока I0 или I3 для последующих расчетов мощностных характеристик. Кроме этой функции в остальном блоки F0-F2 идентичны;
- все каналы АЦП имеют независимые калибровочные коэффициенты наклона характеристики;
- каждый канал тока имеет независимый интегратор;

- 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 0,005 %;
- 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 доступны все семь каналов. При этом для каналов напряжений необходимо настроить регистр ADCUI\_FxMD0, поле FxVSEL, в значение «00» для отсчётов напряжений (установлено по умолчанию), когда как для каналов тока I0-3 необходимо настроить регистр ADCUI\_MSC, поле FxISEL\_HPF, в значение «1».

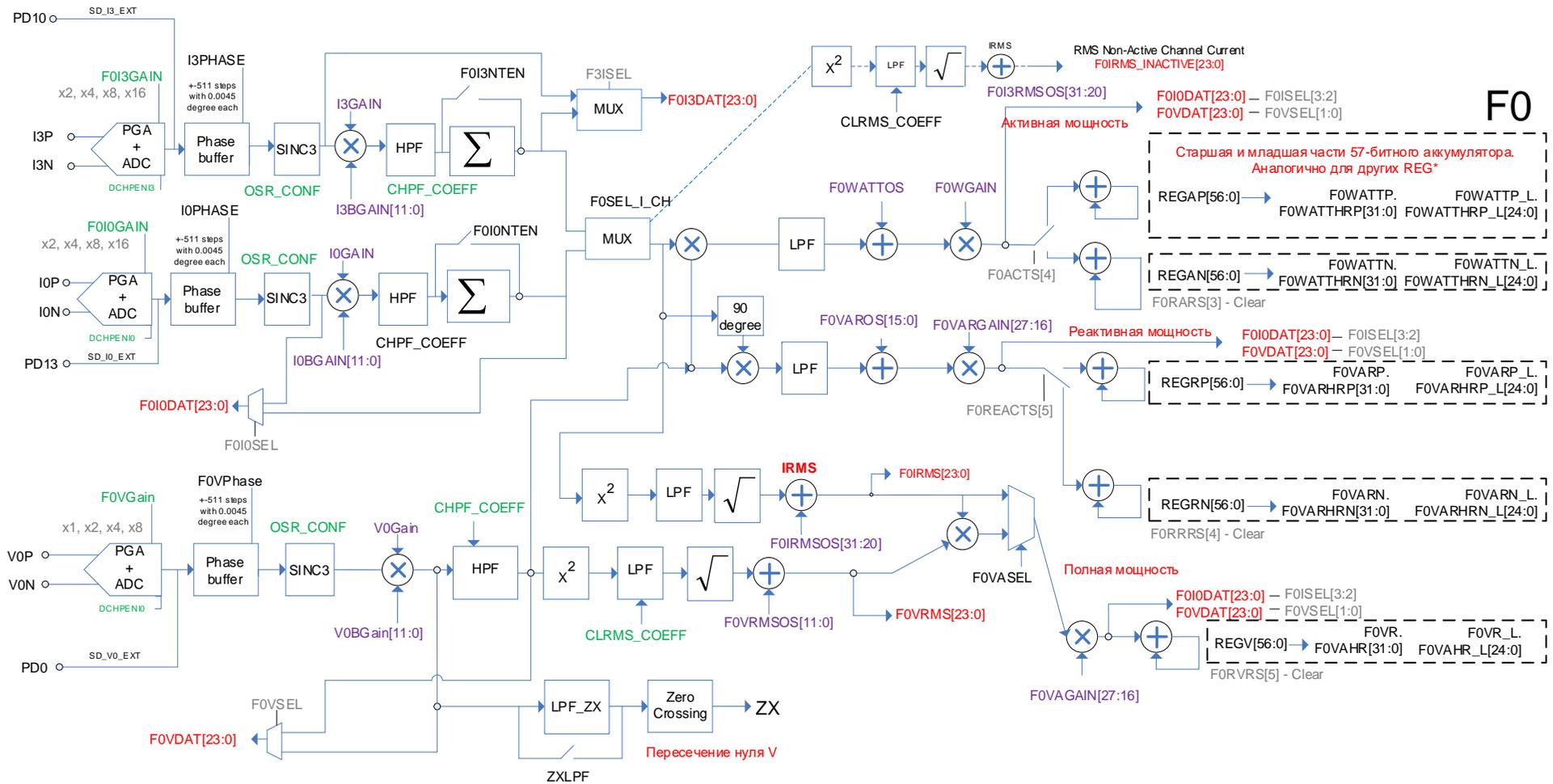


Рисунок 16 – Структурная схема F0

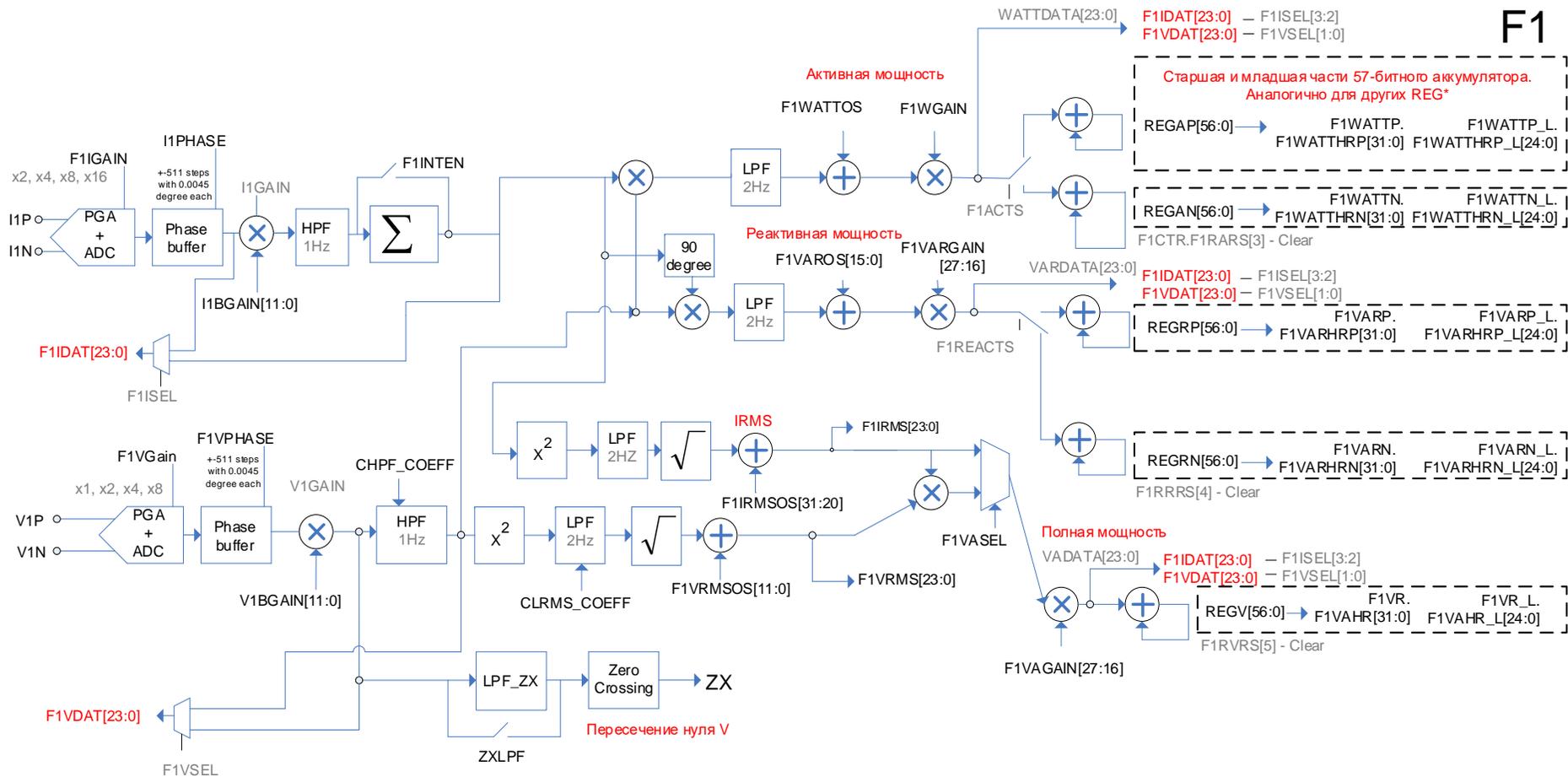


Рисунок 17 – Структурная схема F1

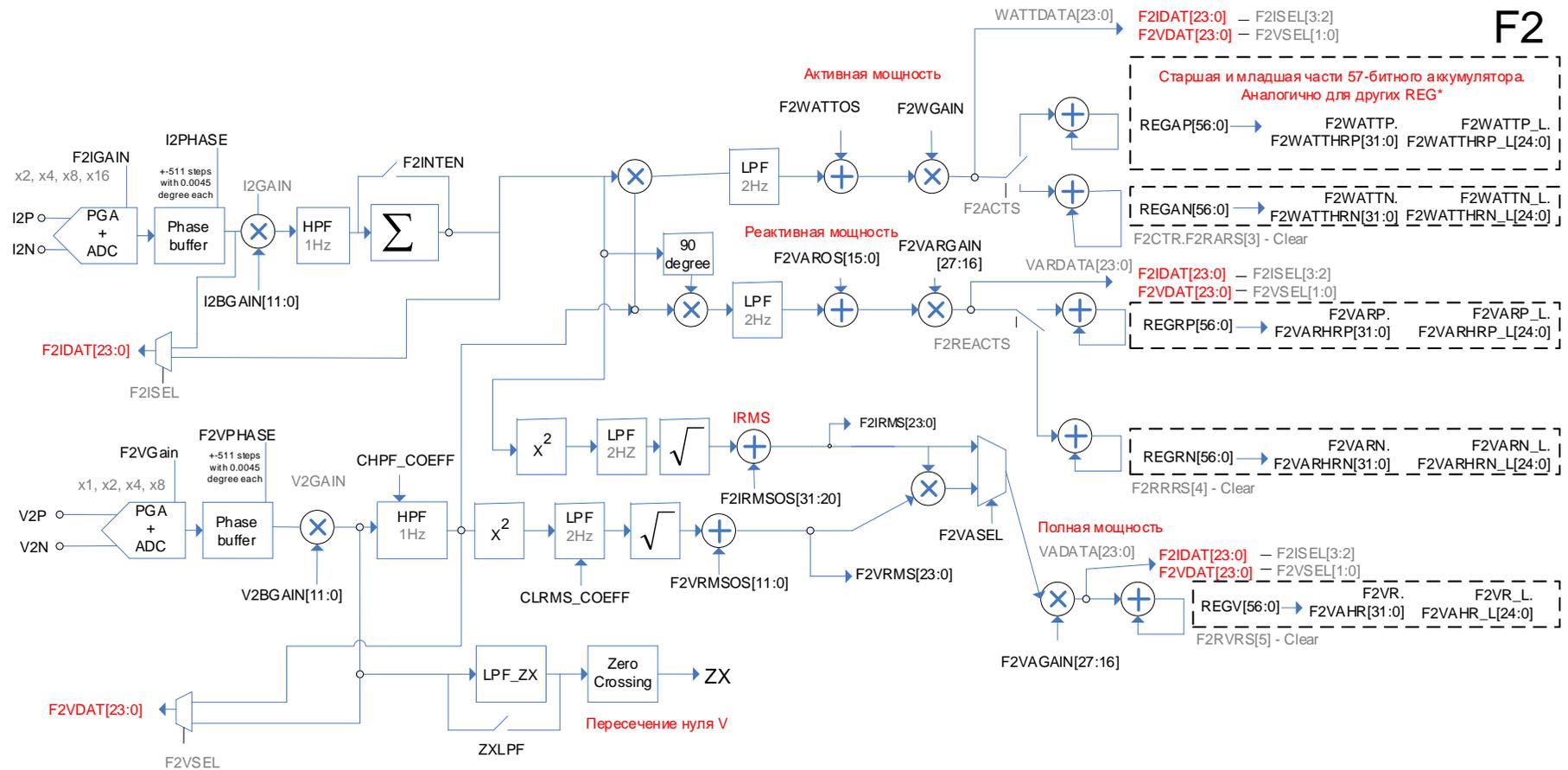


Рисунок 18 – Структурная схема F2

### 13.1 Описание регистров управления блока семи каналов АЦП

Таблица 29 – Регистры управления блока семи каналов АЦП

Базовый Адрес	Название	Описание
0x5006_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 в канале F0
0x010	ADCUI_F0WC	Управление расчета активной мощности в канале F0
0x014	ADCUI_F0WATTP	Старшая часть значения положительной активной мощности в канале F0
0x018	ADCUI_F0WATTN	Старшая часть значения отрицательной активной мощности в канале F0
0x01C	ADCUI_F0VC	Управление расчета реактивной мощности в канале F0
0x020	ADCUI_F0VARP	Старшая часть значения положительной реактивной мощности в канале F0
0x024	ADCUI_F0VARN	Старшая часть значения отрицательной реактивной мощности в канале F0
0x028	ADCUI_F0AC	Управление расчета полной мощности в канале F0
0x02C	ADCUI_F0VR	Старшая часть значения полной мощности в канале F0
0x030	ADCUI_F0MD0	Параметры 0 канала F0
0x034	ADCUI_F0MD1	Параметры 1 канала F0
0x038	ADCUI_F0VPEAK	Пиковое значение в канале напряжения в канале F0
0x03C	ADCUI_F0IPEAK	Пиковое значение в канале тока в канале F0
0x040	ADCUI_F0VDAT	Отсчеты напряжения в канале F0
0x044	ADCUI_F0I0DAT	Отсчеты тока I0 в канале F0
0x048	ADCUI_F0I3DAT	Отсчеты тока I3 в канале F0
0x04C	ADCUI_F0VRMS	Среднеквадратическое значение напряжение канала F0
0x050	ADCUI_F0VRMS2	Квадрат RMS в канале напряжения F0
0x054	ADCUI_F0IRMS	Среднеквадратическое значение тока канала F0
0x058	ADCUI_F0IRMS2	Квадрат RMS в канале тока F0
0x05C	ADCUI_F0STAT	Статус канала F0
0x060	ADCUI_F0MASK	Маска прерываний канала F0
0x064	ADCUI_F1CTR	Управление в канале F1
0x068	ADCUI_F1WC	Управление расчета активной мощности в канале F1
0x06C	ADCUI_F1WATTP	Старшая часть значения положительной активной мощности в канале F1
0x070	ADCUI_F1WATTN	Старшая часть значения отрицательной активной мощности в канале F1
0x074	ADCUI_F1VC	Управление расчета реактивной мощности в канале F1
0x078	ADCUI_F1VARP	Старшая часть значения положительной реактивной мощности в канале F1

Базовый Адрес	Название	Описание
0x5006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x07C	ADCUI_F1VARN	Старшая часть значения отрицательной реактивной мощности в канале F1
0x080	ADCUI_F1AC	Управление расчета полной мощности в канале F1
0x084	ADCUI_F1VR	Старшая часть значения полной мощности в канале F1
0x088	ADCUI_F1MD0	Параметры 0 канала F1
0x08C	ADCUI_F1MD1	Параметры 1 канала F1
0x090	ADCUI_F1MD2	Параметры 2 канала F1
0x094	ADCUI_F1VPEAK	Пиковое значение в канале напряжения в канале F1
0x098	ADCUI_F1IPEAK	Пиковое значение в канале тока в канале F1
0x09C	ADCUI_F1VDAT	Отсчеты напряжения в канале F1
0x0A0	ADCUI_F1IDAT	Отсчеты тока в канале F1
0x0A4	ADCUI_F1VRMS	Среднеквадратическое значение напряжение канала F1
0x0A8	ADCUI_F1VRMS2	Квадрат RMS в канале напряжения F1
0x0AC	ADCUI_F1IRMS	Среднеквадратическое значение тока канала F1
0x0B0	ADCUI_F1IRMS2	Квадрат RMS в канале тока F1
0x0B4	ADCUI_F1STAT	Статус канала F1
0x0B8	ADCUI_F1MASK	Маска прерываний канала F1
0x0BC	ADCUI_F2CTR	Управление в канале F2
0x0C0	ADCUI_F2WC	Управление расчета активной мощности в канале F2
0x0C4	ADCUI_F2WATTP	Старшая часть значения положительной активной мощности в канале F2
0x0C8	ADCUI_F2WATTN	Старшая часть значения отрицательной активной мощности в канале F2
0x0CC	ADCUI_F2VC	Управление расчета реактивной мощности в канале F2
0x0D0	ADCUI_F2VARP	Старшая часть значения положительной реактивной мощности в канале F2
0x0D4	ADCUI_F2VARN	Старшая часть значения отрицательной реактивной мощности в канале F2
0x0D8	ADCUI_F2AC	Управление расчета полной мощности в канале F2
0x0DC	ADCUI_F2VR	Старшая часть значения полной мощности в канале F2
0x0E0	ADCUI_F2MD0	Параметры 0 канала F2
0x0E4	ADCUI_F2MD1	Параметры 1 канала F2
0x0E8	ADCUI_F2MD2	Параметры 2 канала F2
0x0EC	ADCUI_F2VPEAK	Пиковое значение в канале напряжения в канале F2
0x0F0	ADCUI_F2IPEAK	Пиковое значение в канале тока в канале F2
0x0F4	ADCUI_F2VDAT	Отсчеты напряжения в канале F2
0x0F8	ADCUI_F2IDAT	Отсчеты тока в канале F2
0x0FC	ADCUI_F2VRMS	Среднеквадратическое значение напряжение канала F2
0x100	ADCUI_F2VRMS2	Квадрат RMS в канале напряжения F2
0x104	ADCUI_F2IRMS	Среднеквадратическое значение тока канала F2
0x108	ADCUI_F2IRMS2	Квадрат RMS в канале тока F2
0x10C	ADCUI_F2STAT	Статус канала F2

Базовый Адрес	Название	Описание
0x5006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x110	ADCUI_F2MASK	Маска прерываний канала F2
0x114	ADCUI_CCAL1	Регистр 1 калибровки канала тока
0x118	ADCUI_CCAL2	Регистр 2 калибровки канала тока
0x11C	ADCUI_CCAL3	Регистр 3 калибровки канала тока
0x120	ADCUI_CCAL4	Регистр 4 калибровки канала тока
0x124	ADCUI_F0IRMS_INACTIVE	Среднеквадратическое значение тока неактивного канала F0
0x128	ADCUI_F0WATTP_L	Младшая часть значения положительной активной мощности в канале F0
0x12C	ADCUI_F0WATTN_L	Младшая часть значения отрицательной активной мощности в канале F0
0x130	ADCUI_F0VARP_L	Младшая часть значения положительной реактивной мощности в канале F0
0x134	ADCUI_F0VARN_L	Младшая часть значения отрицательной реактивной мощности в канале F0
0x138	ADCUI_F0VR_L	Младшая часть значения полной мощности в канале F0
0x13C	ADCUI_F1WATTP_L	Младшая часть значения положительной активной мощности в канале F1
0x140	ADCUI_F1WATTN_L	Младшая часть значения отрицательной активной мощности в канале F1
0x144	ADCUI_F1VARP_L	Младшая часть значения положительной реактивной мощности в канале F1
0x148	ADCUI_F1VARN_L	Младшая часть значения отрицательной реактивной мощности в канале F1
0x14C	ADCUI_F1VR_L	Младшая часть значения полной мощности в канале F1
0x150	ADCUI_F2WATTP_L	Младшая часть значения положительной активной мощности в канале F2
0x154	ADCUI_F2WATTN_L	Младшая часть значения отрицательной активной мощности в канале F2
0x158	ADCUI_F2VARP_L	Младшая часть значения положительной реактивной мощности в канале F2
0x15C	ADCUI_F2VARN_L	Младшая часть значения отрицательной реактивной мощности в канале F2
0x160	ADCUI_F2VR_L	Младшая часть значения полной мощности в канале F2
0x164	ADCUI_LOAD	Общее управление режимом «без нагрузки»
0x168	ADCUI_F0VRMS_TRUE	Действующее значение напряжения на периоде канала F0
0x16C	ADCUI_F1VRMS_TRUE	Действующее значение напряжения на периоде канала F1
0x170	ADCUI_F2VRMS_TRUE	Действующее значение напряжения на периоде канала F2
0x174	ADCUI_MSC	Разрешение формирования запросов к DMA и др.
0x178	ADCUI_FILTERCFG	Управление конфигурацией фильтров

Базовый Адрес	Название	Описание
0x5006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x17C	ADCUI_CLKPHASE1	Управление сдвигом фазы токовых каналов
0x180	ADCUI_CLKPHASE2	Управление сдвигом фазы токовых каналов

### 13.1.1 ADCUI\_CTRL1

Таблица 30 – 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	VOEN	IOEN

Таблица 31 – Описание бит регистра ADCUI\_CTRL1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	OSR_CONF*	Выбор коэффициента передискретизации: 00 – OSR = 1024; 01 – OSR = 512; 10 – OSR = 256; 11 – зарезервировано.  Примечание – коэффициенты применимы к тактовой частоте $f_{C\_ADC}$ блока АЦП, которая может быть равна либо 1,024 МГц, либо 2,048 МГц, либо 4,096 МГц, в зависимости от настройки поля CLC_ADC_CFG в регистре ADCUI_CLKPHASE1
29	tst_in[4]	Тестовый выход (всегда записывать ноль)

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 61)
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 (любая положительная и отрицательная энергия); 01 – не учитывается энергия ниже 0,0305 % от полной шкалы энергии кванта; 10 – не учитывается энергия ниже 0,0152 % от полной шкалы энергии кванта; 11 – не учитывается энергия ниже 0,0076 % от полной шкалы энергии кванта.  Энергия кванта – энергия за период входного сигнала (20 мс), определяемый событиями детектором пересечения «0»: – при частоте дискретизации 4 кГц – это сумма 80 отчетов; – при частоте дискретизации 8 кГц – это сумма 160 отчетов; – при частоте дискретизации 16 кГц – это сумма 320 отчетов
16, 15	VARNLOAD	Режим «без нагрузки» при вычислении реактивной энергии: 00 – вся вычисленная энергия накапливается (любая положительная и отрицательная энергия); 01 – не учитывается энергия ниже 0,0305 % от полной шкалы энергии кванта; 10 – не учитывается энергия ниже 0,0152 % от полной шкалы энергии кванта; 11 – не учитывается энергия ниже 0,0076 % от полной шкалы энергии кванта.  Энергия кванта – энергия за период входного сигнала (20 мс), определяемый событиями детектором пересечения «0»: – при частоте дискретизации 4 кГц – это сумма 80 отчетов; – при частоте дискретизации 8 кГц – это сумма 160 отчетов; – при частоте дискретизации 16 кГц – это сумма 320 отчетов

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
14	-	Зарезервировано
13, 12	APNOLOAD	Режим «без нагрузки» при вычислении активной энергии: 00 – вся вычисленная энергия накапливается (любая положительная и отрицательная энергия); 01 – не учитывается энергия ниже 0,0305 % от полной шкалы энергии кванта; 10 – не учитывается энергия ниже 0,0152 % от полной шкалы энергии кванта; 11 – не учитывается энергия ниже 0,0076 % от полной шкалы энергии кванта.  Энергия кванта – энергия за период входного сигнала (20 мс), определяемый событиями детектором пересечения «0»: – при частоте дискретизации 4 кГц – это сумма 80 отчетов; – при частоте дискретизации 8 кГц – это сумма 160 отчетов; – при частоте дискретизации 16 кГц – это сумма 320 отчетов
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. Влияет на расчет F1PER_FREQ, F2PER_FREQ, FOPER_FREQ, F1PHASE, F2PHASE
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***

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1	VOEN	Разрешение работы канала V0: 0 – канал отключен; 1 – канал включен***
0	IOEN	Разрешение работы канала IO: 0 – канал отключен; 1 – канал включен***

\* При изменении частоты дискретизации необходимо соответствующим образом корректировать настройки цифровых фильтров HPF и LPF, для сохранения их частоты среза. Так же 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.

\*\* Так как происходит одновременное обновление среднеквадратических значений и тока и напряжение, то значение тока будет зависеть от угла между напряжением и током. На рисунке 17 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

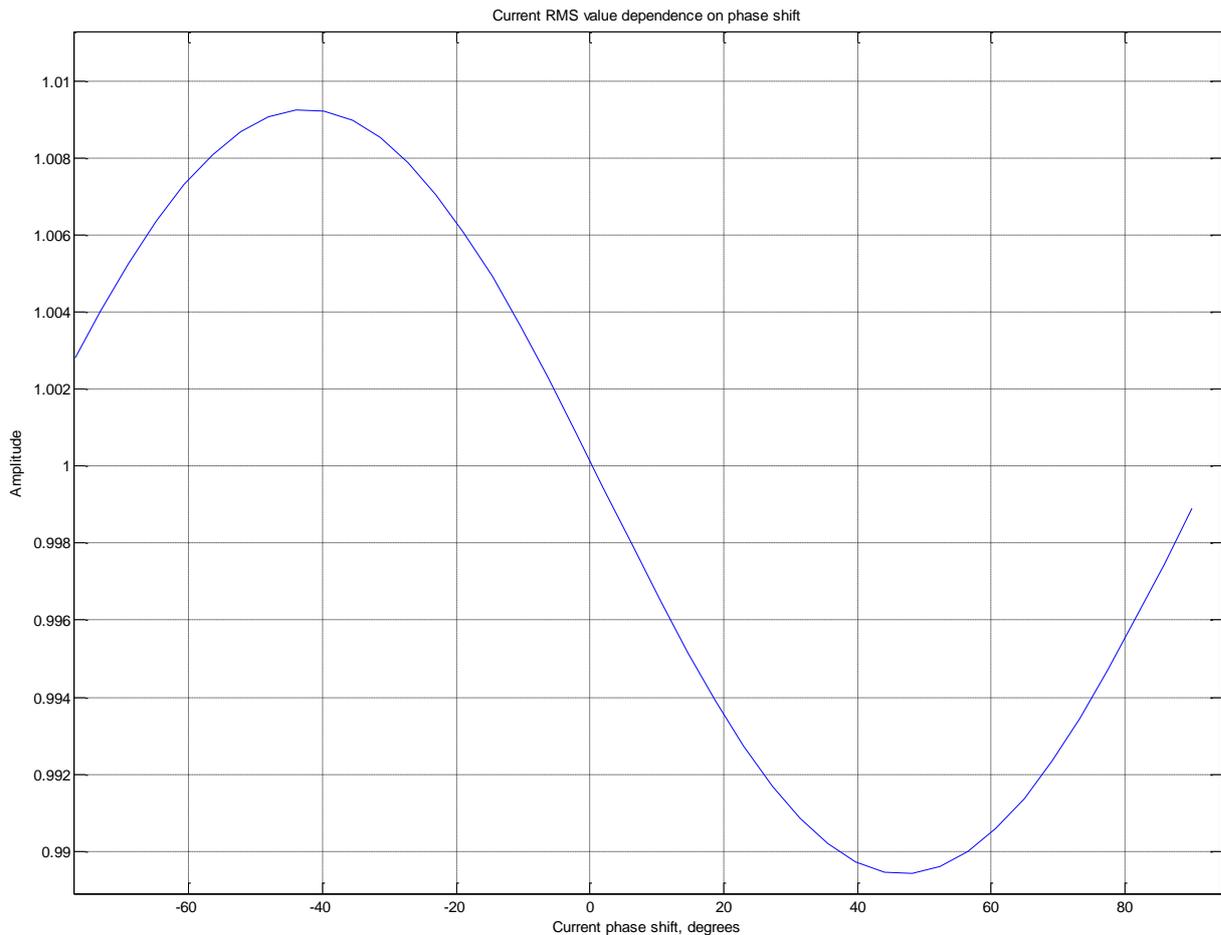


Рисунок 19 – Зависимость RMS от сдвига фазы входного сигнала

### 13.1.2 ADCUI\_CTRL2

Таблица 32 – Регистр ADCUI\_CTRL2

Номер	31	30	29	28	27	26
Доступ	–	R/W	R/W	R/W	R/W	R/W
Сброс	–	0	0	0	0	0
	–	DCHPENI3	DCHPENI2	DCHPENI1	DCHPENI0	DCHPENV 2

Номер	25	24	23...16	15...0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	FFh	FFFFh
	DCHPENV1	DCHPENV0	SAGCYC	SAGLVL

Таблица 33 – Описание бит регистра ADCUI\_CTRL2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	–	Зарезервировано
30	DCHPENI3	0 – чоппер канала I3 выключен; 1 – чоппер канала I3 включен*
29	DCHPENI2	0 – чоппер канала I2 выключен; 1 – чоппер канала I2 включен*
28	DCHPENI1	0 – чоппер канала I1 выключен; 1 – чоппер канала I1 включен*
27	DCHPENI0	0 – чоппер канала I0 выключен; 1 – чоппер канала I0 включен*
26	DCHPENV2	0 – чоппер канала V2 выключен; 1 – чоппер канала V2 включен*
25	DCHPENV1	0 – чоппер канала V1 выключен; 1 – чоппер канала V1 включен*
24	DCHPENV0	0 – чоппер канала V0 выключен; 1 – чоппер канала V0 включен*
23...16	SAGCYC	Количество полутактов напряжения для вычисления просадки уровня напряжения
15...0	SAGLVL	Уровень разрешенной просадки напряжения

\* Рекомендуется использовать при измерении постоянных сигналов. При использовании внешнего дельта-сигма модулятора должен быть выключен

### 13.1.3 ADCUI\_CTRL3

Таблица 34 – Регистр ADCUI\_CTRL3

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

Таблица 35 – Описание бит регистра ADCUI\_CTRL3

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..12	-	Зарезервировано
11..0	ZXTOUT	Значение time-out счетчиков отсутствия пересечения ноля входным сигналом в каналах V0, V1, V2, которые устанавливают флаги F0ZXTOF, F1ZXTOF, F2ZXTOF.  Значение выражено в периодах частоты передискретизации АЦП 4/8/16 кГц (зависит от OSR_CONF и CLC_ADC_CFG)  По сбросу равно ~64мс.

### 13.1.4 ADCUI\_F0CTR

Таблица 36 – 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	F0VGAIN	F0I0GAIN

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 37 – Описание бит регистра ADCUI\_F0CTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F0IRMSOS	Калибровка смещения для вычислителя среднеквадратического значения тока канала I0/ 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 · 16 = 112 FF9: IRMSOS = -7 · 16 = -112
19, 18	F0I3GAIN	Предусилитель в канале тока 3: 00 – 6dB; 01 – 12dB; 10 – 18dB; 11 – 24dB

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
17...10	F0VPHASE[9:2]	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -511*(такт АЦП) до +511*(такт АЦП). “000000000” соответствует синфазному сигналу с током. При использовании DCHPENV0 = 1 сдвигать фазу рекомендуется с шагом 32. Младшие биты F0VPHASE[1:0] находятся в ADCUI_MSC
9, 8	F0V0GAIN	Предусилитель в канале напряжения: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
7, 6	F0I0GAIN	Предусилитель в канале тока 0: 00 – 6dB; 01 – 12dB; 10 – 18dB; 11 – 24dB
5	F0RVRS	Запись в этот регистр сбрасывает счетчик переданной полной энергии
4	F0RRRS	Запись в этот регистр сбрасывает счетчик переданной реактивной энергии
3	F0RARS	Запись в этот регистр 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 38 – Регистр ADCUI\_F0WC

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

Таблица 39 – Описание бит регистра ADCUI\_F0WC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано

27...16	F0WGAIN	Калибровка усиления канала
15...0	F0WATTOS	Калибровка смещения канала

### 13.1.6 ADCUI\_F0WATTP

Таблица 40 – Регистр ADCUI\_F0WATTP

Номер	31...0
Доступ	RO
Сброс	
	F0WATTHRP

Таблица 41 – Описание бит регистра ADCUI\_F0WATTP

Разряды	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 42 – Регистр ADCUI\_F0WATTN

Номер	31...0
Доступ	RO
Сброс	
	F0WATTHRN

Таблица 43 – Описание бит регистра ADCUI\_F0WATTN

Разряды	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 44 – Регистр ADCUI\_F0VC

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

Таблица 45 – Описание бит регистра ADCUI\_F0VC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0VARGAIN	Калибровка усиления канала
15...0	F0VAROS	Калибровка смещения канала

### 13.1.9 ADCUI\_F0VARP

Таблица 46 – Регистр ADCUI\_F0VARP

Номер	31..0
Доступ	RO
Сброс	
	F0VARHRP

Таблица 47 – Описание бит регистра ADCUI\_F0VARP

Разряды	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 48 – Регистр ADCUI\_F0VARN

Номер	31..0
Доступ	RO
Сброс	
	F0VARHRN

Таблица 49 – Описание бит регистра ADCUI\_F0VARN

Разряды	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 50 – 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 51 – Описание бит регистра ADCUI\_F0AC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0VAGAIN	Калибровка усиления канала
15...13	-	Зарезервировано
12	F0I3SEL	Выбор источника сигнала для регистра ADCUI_F0I3DAT: 0 – после фильтра высоких частот; 1 – до фильтра высоких частот
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: IRMSOS = $7 \cdot 16 = 112$ FF9: IRMSOS = $-7 \cdot 16 = -112$

### 13.1.12 ADCUI\_F0VR

Таблица 52 – Регистр ADCUI\_F0VR

Номер	31...0
Доступ	RO
Сброс	
	F0VAHR

Таблица 53 – Описание бит регистра ADCUI\_F0VR

Разряды	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 54 – Регистр ADCUI\_F0MD0

Номер	31, 30	29...9	8	7
Доступ	R/W	RO	R/W	R/W
Сброс	01		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 55 – Описание бит регистра ADCUI\_F0MD0

Разряды	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 – недопустимое значение; 01 – активный канал I0; 10 – активный канал I3
29...9	F0PER_FREQ	Длительность периода входного сигнала в канале напряжения V0. Значение выражено в периодах частоты 1/2/4 МГц (в зависимости от OSR_CONF 0/1/2), вычисленное и усредненное за число периодов входной частоты, определяемых PER_LENGTH. Например, значение для 50 Гц входного сигнала: $20 \text{ мс} \cdot 4 \text{ MHz} = 80000$
8	I3GAIN	Цифровое усиление в канале I3: 0 – нет усиления; 1 – усиление +6 дБ (сдвиг результата влево на 1)
7	V0GAIN	Цифровое усиление в канале V0: 0 – нет усиления; 1 – усиление +6 дБ (сдвиг результата влево на 1)
6	I0GAIN	Цифровое усиление в канале I0: 0 – нет усиления; 1 – усиление +6 дБ (сдвиг результата влево на 1)
5	F0REACTS	Знак реактивной энергии в последний период
4	F0ACTS	Знак активной энергии в последний период
3, 2	F0ISEL	Выбор источника сигнала для регистра ADCUI_F0I0DAT: 00 – отсчеты тока (после или до HPF, см. F0ISEL_HPFF); 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1, 0	F0VSEL	Выбор источника сигнала для регистра ADCUI_F0V0DAT: 00 – отсчеты напряжения (до HPF); 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты напряжения (после HPF)

### 13.1.14 ADCUI\_F0MD1

Таблица 56 – Регистр ADCUI\_F0MD1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F0V0PKLVL	F0I0PKLVL

Таблица 57 – Описание бит регистра ADCUI\_F0MD1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	F0VPKLVL	Предельный разрешенный уровень напряжения. Пороговое значение заданное в этом регистре сравнивается с 16 старшими битами на выходе HPF фильтра канала V0. При превышении выставляется флаг FOPEAKVF
15...0	F0IPKLVL	Предельный разрешенный уровень тока. Пороговое значение заданное в этом регистре сравнивается с 16 старшими битами на выходе HPF фильтра канала I0. При превышении выставляется флаг FOPEAKIF.

### 13.1.15 ADCUI\_F0VPEAK

Таблица 58 – Регистр ADCUI\_F0VPEAK

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

Таблица 59 – Описание бит регистра ADCUI\_F0VPEAK

Разряды	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 60 – Регистр ADCUI\_F0IPEAK

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

Таблица 61 – Описание бит регистра ADCUI\_F0IPEAK

Разряды	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\_F0V DAT

Таблица 62 – Регистр ADCUI\_F0V DAT

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	-	F0V DAT

Таблица 63 – Описание бит регистра ADCUI\_F0V DAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0V DAT	FIFO отсчетов напряжения (или одной из мощностей) FIFO включается при установке любого из битов I3EN, V0EN, I0EN

### 13.1.18 ADCUI\_F0I0 DAT

Таблица 64 – Регистр ADCUI\_F0I0 DAT

Номер	31...24	23...0
Доступ		RO
Сброс		000000h
	-	F0I0 DAT

Таблица 65 – Описание бит регистра ADCUI\_F0I0 DAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0I0 DAT	FIFO отсчетов тока 0 (или одной из мощностей) FIFO включается при установке любого из битов I3EN, V0EN, I0EN

### 13.1.19 ADCUI\_F0I3 DAT

Таблица 66 – Регистр ADCUI\_F0I3 DAT

Номер	31:24	23:0
Доступ		RO
Сброс		000000h
	-	F0I3 DAT

Таблица 67 – Описание бит регистра ADCUI\_F0I3 DAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F0I3 DAT	FIFO отсчетов тока 3. FIFO включается при установке любого из битов I3EN, V0EN, I0EN

### 13.1.20 ADCUI\_F0VRMS

Таблица 68 – Регистр ADCUI\_F0VRMS

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

Таблица 69 – Описание бит регистра ADCUI\_F0VRMS

Разряды	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 70 – Регистр ADCUI\_F0VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F0VRMS2

Таблица 71 – Описание бит регистра ADCUI\_F0VRMS2

Разряды	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 72 – Регистр ADCUI\_F0IRMS

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

Таблица 73 – Описание бит регистра ADCUI\_F0IRMS

Разряды	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 74 – Регистр ADCUI\_F0IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F0IRMS2

Таблица 75 – Описание бит регистра ADCUI\_F0IRMS2

Разряды	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 76 – 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
	-	F0VAROVN	F0WATTOVN	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
	F0ZEROCRS	F0VANLDFL	-	F0VARNLDFL	F0VARSIGN	F0APNLDFL

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
	F0APSIGN	F0FAULTCON	F0ICHANNEL	F0ZXTOF	F0VAOV	F0VAROVVP	F0WATTOVP	F0PEAKIF	F0PEAKVF

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 77 – Описание бит регистра ADCUI\_F0STAT

Разряды	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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
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	Запись «1» сбрасывает этот флаг
13	F0ICHANNEL	Активный канал тока: 0 – активный канал I0; 1 – активный канал I3
12	F0ZXTOF	Флаг, что в течении ZXTOUT не было пересечения напряжением значения «0» в канале V0. 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 78 – 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
	F0APSI GNM	F0FAUL TCONM	-	F0ZXTO FM	F0VAOV M	F0VARO VPM	F0WATT OVPM	F0PEAKI FM	F0PEAK VFM

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
	F0SAGFM	F0IF_ OVRM	F0IF_ FLLM	F0IF_ EMPM	F0VF_ OVRM	F0VF_ FLLM	F0VF_ EMPM

Таблица 79 – Описание бит регистра ADCUI\_F0MASK

Разряды	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
11	F0VAOVM	Маска бита F0VAOV
10	F0VAROVP	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 80 – Регистр ADCUI\_F1CTR

Номер	31:20	19, 18	17:10	9, 8	7, 6
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	00	0	0
	F1IRMSOS	F1VPHASE	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 81 – Описание бит регистра ADCUI\_F1CTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F1IRMSOS	Калибровка смещения для вычислителя среднеквадратического значения тока канала I1/ 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 · 16 = 112 FF9: IRMSOS = -7 · 16 = -112
19, 18	F1VPHASE[1:0]	Младшие биты F1VPHASE

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
17...10	F1VPHASE[9:2]	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -511·(такт АЦП) до +511·(такт АЦП). «0000000000» соответствует синфазному сигналу с током. При использовании DCHPENV1 = 1 сдвигать фазу рекомендуется с шагом 32
9, 8	F1VGAIN	Предусилитель в канале напряжения: 00 – 0 дБ; 01 – 6 дБ; 10 – 12 дБ; 11 – 18 дБ
7, 6	F1IGAIN	Предусилитель в канале тока: 00 – 6 дБ; 01 – 12 дБ; 10 – 18 дБ; 11 – 24 дБ
5	F1RVRS	Запись в этот регистр сбрасывает счетчик переданной полной энергии
4	F1RRRS	Запись в этот регистр сбрасывает счетчик переданной реактивной энергии
3	F1RARS	Запись в этот регистр 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 82 – Регистр ADCUI\_F1WC

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

Таблица 83 – Описание бит регистра ADCUI\_F1WC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F1WGAIN	Калибровка усиления канала
15...0	F1WATTOS	Калибровка смещения канала

### 13.1.28 ADCUI\_F1WATTP

Таблица 84 – Регистр ADCUI\_F1WATTP

Номер	31...0
Доступ	RO
Сброс	
	F1WATTHRP

Таблица 85 – Описание бит регистра ADCUI\_F1WATTP

Разряды	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 86 – Регистр ADCUI\_F1WATTN

Номер	31...0
Доступ	RO
Сброс	
	F1WATTHRN

Таблица 87 – Описание бит регистра ADCUI\_F1WATTN

Разряды	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 88 – Регистр ADCUI\_F1VC

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

Таблица 89 – Описание бит регистра ADCUI\_F1VC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F1VARGAIN	Калибровка усиления канала
15...0	F1VAROS	Калибровка смещения канала

### 13.1.31 ADCUI\_F1VARP

Таблица 90 – Регистр ADCUI\_F1VARP

Номер	31...0
Доступ	RO
Сброс	
	F1VARHRP

Таблица 91 – Описание бит регистра ADCUI\_F1VARP

Разряды	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 92 – Регистр ADCUI\_F1VARN

Номер	31...0
Доступ	RO
Сброс	
	F1VARHRN

Таблица 93 – Описание бит регистра ADCUI\_F1VARN

Разряды	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 94 – Регистр ADCUI\_F1AC

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

Таблица 95 – 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 канала
15...12	-	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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: IRMSOS = $7 \cdot 16 = 112$ FF9: IRMSOS = $-7 \cdot 16 = -112$

### 13.1.34 ADCUI\_F1VR

Таблица 96 – Регистр ADCUI\_F1VR

Номер	31...0
Доступ	RO
Сброс	
	F1VAHR

Таблица 97 – Описание бит регистра ADCUI\_F1VR

Разряды	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 98 – 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 99 – Описание бит регистра ADCUI\_F1MD0

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...8	F1PER_FREQ	Длительность периода входного сигнала в канале напряжения V1. Значение выражено в периодах частоты 1/2/4 МГц (в зависимости от OSR_CONF 0/1/2), вычисленное и усредненное за число периодов входной частоты, определяемых PER_LENGTH. Например, значение для 50 Гц входного сигнала: $20 \text{ мс} \cdot 4 \text{ МГц} = 80000$

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	V1GAIN	Цифровое усиление в канале V1: 0 – нет усиления; 1 – усиление +6 дБ (сдвиг результата влево на 1)
6	I1GAIN	Цифровое усиление в канале I1: 0 – нет усиления; 1 – усиление +6 дБ (сдвиг результата влево на 1)
5	F1REACTS	Знак реактивной энергии в последний период
4	F1ACTS	Знак активной энергии в последний период
3, 2	F1ISEL	Выбор источника сигнала для регистра ADCUI_F1IDAT: 00 – отсчеты тока (до или после HPF, см. F1ISEL_HPFF); 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1, 0	F1VSEL	Выбор источника сигнала для регистра ADCUI_F1VDAT: 00 – отсчеты напряжения; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности

### 13.1.36 ADCUI\_F1MD1

Таблица 100 – Регистр ADCUI\_F1MD1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F1VPKLV L	F1IPKLV L

Таблица 101 – Описание бит регистра ADCUI\_F1MD1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	F1VPKLV L	Предельный разрешенный уровень напряжения. Пороговое значение заданное в этом регистре сравнивается с 16 старшими битами на выходе HPF фильтра канала V1. При превышении выставляется флаг F1PEAKVF
15...0	F1IPKLV L	Предельный разрешенный уровень тока. Пороговое значение заданное в этом регистре сравнивается с 16 старшими битами на выходе HPF фильтра канала I1. При превышении выставляется флаг F0PEAKIF

### 13.1.37 ADCUI\_F1MD2

Таблица 102 – Регистр ADCUI\_F1MD2

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

Таблица 103 – Описание бит регистра ADCUI\_F1MD2

Разряды	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. Значение выражено в периодах частоты 1/2/4 МГц (в зависимости от OSR_CONF 0/1/2), вычисленное и усредненное за число периодов входной частоты, определяемых PER_LENGTH

### 13.1.38 ADCUI\_F1VPEAK

Таблица 104 – Регистр ADCUI\_F1VPEAK

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

Таблица 105 – Описание бит регистра ADCUI\_F1VPEAK

Разряды	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 106 – Регистр ADCUI\_F1IPEAK

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

Таблица 107 – Описание бит регистра ADCUI\_F1PEAK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1PEAK	Пиковое значение тока. Обнуляется чтением из регистра. Запись в этот регистр, сбрасывает пиковое значение

### 13.1.40 ADCUI\_F1VDAT

Таблица 108 – Регистр ADCUI\_F1VDAT

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

Таблица 109 – Описание бит регистра ADCUI\_F1VDAT

Разряды	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 в зависимости от F1VSEL). FIFO включается при установке любого из бит V1EN, I1EN

### 13.1.41 ADCUI\_F1IDAT

Таблица 110 – Регистр ADCUI\_F1IDAT

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

Таблица 111 – Описание бит регистра ADCUI\_F1IDAT

Разряды	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 в зависимости от F1ISEL). FIFO включается при установке любого из бит V1EN, I1EN

### 13.1.42 ADCUI\_F1VRMS

Таблица 112 – Регистр ADCUI\_F1VRMS

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

Таблица 113 – Описание бит регистра ADCUI\_F1VRMS

Разряды	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 114 – Регистр ADCUI\_F1VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F1VRMS2

Таблица 115 – Описание бит регистра ADCUI\_F1VRMS2

Разряды	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 116 – Регистр ADCUI\_F1IRMS

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

Таблица 117 – Описание бит регистра ADCUI\_F1IRMS

Разряды	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 118 – Регистр ADCUI\_F1IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F1IRMS2

Таблица 119 – Описание бит регистра ADCUI\_F1IRMS2

Разряды	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 120 – 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 121 – Описание бит регистра ADCUI\_F1STAT

Разряды	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
25	F1WATTOVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной активной энергией. Запись «1» сбрасывает этот флаг
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 ZXTOUT не было пересечения напряжением значения «0» в канале V1. 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 122 – 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
	F1APSIGNM	-	F1ZXTOFM	F1VAOVM	F1VAROVPM	F1WATTOVPM	F1PEAKIFM	F1PEAKVFM

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
	F1SAGFM	F1IF_OVRM	F1IF_FLLM	F1IF_EMPM	F1VF_OVRM	F1VF_FLLM	F1VF_EMPM

Таблица 123 – Описание бит регистра ADCUI\_F1MASK

Разряды	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

Разряды	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 124 – Регистр ADCUI\_F2CTR

Номер	31...20	19, 18	17...10	9, 8	7, 6
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	00	0	0
	F2IRMSOS	F2VPHASE	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 125 – Описание бит регистра ADCUI\_F2CTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F2IRMSOS	Калибровка смещения для вычислителя среднеквадратического значения тока канала I2/ 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 · 16 = 112 FF9: IRMSOS = -7 · 16 = -112
19, 18	F2VPHASE[1:0]	Младшие биты F2VPHASE
17..10	F2VPHASE[9:2]	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -511· (такт АЦП) до +511· (такт АЦП). “0000000000” соответствует синфазному сигналу с током. При использовании DCHPENV2 = 1 сдвигать фазу рекомендуется с шагом 32
9, 8	F2VGAIN	Предусилитель в канале напряжения: 00 – 0 дБ; 01 – 6 дБ; 10 – 12 дБ; 11 – 18 дБ

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7, 6	F2IGAIN	Предусилитель в канале тока: 00 – 6 дБ; 01 – 12 дБ; 10 – 18 дБ; 11 – 24 дБ
5	F2RVRS	Запись в этот регистр сбрасывает счетчик переданной полной энергии
4	F2RRRS	Запись в этот регистр сбрасывает счетчик переданной реактивной энергии
3	F2RARS	Запись в этот регистр 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 126 – Регистр ADCUI\_F2WC

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

Таблица 127 – Описание бит регистра ADCUI\_F2WC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F2WGAIN	Калибровка усиления канала
15...0	F2WATTOS	Калибровка смещения канала

### 13.1.50 ADCUI\_F2WATTP

Таблица 128 – Регистр ADCUI\_F2WATTP

Номер	31...0
Доступ	RO
Сброс	
	F2WATTHRP

Таблица 129 – Описание бит регистра ADCUI\_F2WATTP

Разряды	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 130 – Регистр ADCUI\_F2WATTN

Номер	31...0
Доступ	RO
Сброс	
	F2WATTHRN

Таблица 131 – Описание бит регистра ADCUI\_F2WATTN

Разряды	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 132 – Регистр ADCUI\_F2VC

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

Таблица 133 – 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 канала
15...0	F2VAROS	Калибровка смещения канала

### 13.1.53 ADCUI\_F2VARP

Таблица 134 – Регистр ADCUI\_F2VARP

Номер	31...0
Доступ	RO
Сброс	
	F2VARHRP

Таблица 135 – Описание бит регистра ADCUI\_F2VARP

Разряды	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 136 – Регистр ADCUI\_F2VARN

Номер	31...0
Доступ	RO
Сброс	
	F2VARHRN

Таблица 137 – Описание бит регистра ADCUI\_F2VARN

Разряды	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 138 – Регистр ADCUI\_F2AC

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

Таблица 139 – Описание бит регистра ADCUI\_F2AC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F2VAGAIN	Калибровка усиления канала
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: IRMSOS = 7 · 16 = 112 FF9: IRMSOS = -7 · 16 = -112

### 13.1.56 ADCUI\_F2VR

Таблица 140 – Регистр ADCUI\_F2VR

Номер	31...0
Доступ	R
Сброс	
	F2VAHR

Таблица 141 – Описание бит регистра ADCUI\_F2VR

Разряды	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 142 – 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 143 – Описание бит регистра ADCUI\_F2MD0

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...8	F2PER_FREQ	Длительность периода входного сигнала в канале напряжения V2. Значение выражено в периодах частоты 1/2/4 МГц (в зависимости от OSR_CONF 0/1/2), вычисленное и усредненное за число периодов входной частоты, определяемых PER_LENGTH. Например, значение для 50 Гц входного сигнала: $20 \text{ мс} \cdot 4 \text{ МГц} = 80000$
7	V2GAIN	Цифровое усиление в канале V2: 0 – нет усиления; 1 – усиление +6 дБ (сдвиг результата влево на 1)
6	I2GAIN	Цифровое усиление в канале I2: 0 – нет усиления; 1 – усиление +6 дБ (сдвиг результата влево на 1)
5	F2REACTS	Знак реактивной энергии в последний период
4	F2ACTS	Знак активной энергии в последний период
3, 2	F2ISEL	Выбор источника сигнала для регистра ADCUI_F2IDAT: 00 – отсчеты тока (до или после HPF, см. F2ISEL_HPF); 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности

1, 0	F2VSEL	Выбор источника сигнала для регистра ADCUI_F2VDAT: 00 – отсчеты напряжения; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
------	--------	--

### 13.1.58 ADCUI\_F2MD1

Таблица 144 – Регистр ADCUI\_F2MD1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F2VPKLVL	F2IPKLVL

Таблица 145 – Описание бит регистра ADCUI\_F2MD1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	F2VPKLVL	Предельный разрешенный уровень напряжения. Пороговое значение заданное в этом регистре сравнивается с 16 старшими битами на выходе HPF фильтра канала V2. При превышении выставляется флаг F2PEAKVF.
15...0	F2IPKLVL	Предельный разрешенный уровень тока. Пороговое значение заданное в этом регистре сравнивается с 16 старшими битами на выходе HPF фильтра канала I2. При превышении выставляется флаг F2PEAKIF

### 13.1.59 ADCUI\_F2MD2

Таблица 146 – Регистр ADCUI\_F2MD2

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

Таблица 147 – Описание бит регистра ADCUI\_F2MD2

Разряды	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. Значение выражено в периодах частоты 1/2/4 МГц (в зависимости от OSR_CONF 0/1/2), вычисленное и усредненное за число периодов входной частоты определяемых PER_LENGTH

### 13.1.60 ADCUI\_F2VPEAK

Таблица 148 – Регистр ADCUI\_F2VPEAK

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

Таблица 149 – Описание бит регистра ADCUI\_F2VPEAK

Разряды	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 150 – Регистр ADCUI\_F2IPEAK

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

Таблица 151 – Описание бит регистра ADCUI\_F2IPEAK

Разряды	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 152 – Регистр ADCUI\_F2VDAT

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

Таблица 153 – Описание бит регистра ADCUI\_F2VDAT

Разряды	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 154 – Регистр ADCUI\_F2IDAT

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

Таблица 155 – Описание бит регистра ADCUI\_F2IDAT

Разряды	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 156 – Регистр ADCUI\_F2VRMS

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

Таблица 157 – Описание бит регистра ADCUI\_F2VRMS

Разряды	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 158 – Регистр ADCUI\_F2VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F2VRMS2

Таблица 159 – Описание бит регистра ADCUI\_F2VRMS2

Разряды	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 160 – Регистр ADCUI\_F2IRMS

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

Таблица 161 – Описание бит регистра ADCUI\_F2IRMS

Разряды	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 162 – Регистр ADCUI\_F2IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F2IRMS2

Таблица 163 – Описание бит регистра ADCUI\_F2IRMS2

Разряды	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 164 – 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
	F2APSIGN	-	F2ZXTOF	F2VAOV	F2VAROV	F2WATT	F2PEAKIF	F2PEAKVF

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
	-	F2VAROVN	F2WATTOVN	-	F2ZEROCRS	F2VANLDFL	-	F2VARNLDFL	F2VARSIGN	F2APNLDFL

Таблица 165 – Описание бит регистра ADCUI\_F2STAT

Разряды	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 ZXTOUT не было пересечения напряжением значения “0” в канале V1. 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.

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		Запись «1» сбрасывает этот флаг
4	F2IF_FLL	Флаг, что FIFO F2IDAT заполнено
3	F2IF_EMP	Флаг, что FIFO F2IDAT пусто
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 166 – 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
	-	F2VAROVNM	F2WATTOVNM	-	F2ZEROCRSM	F2VANLDFLM	-	F2VARNLDFLM	F2VARSIGNM	F2APNLDFLM

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
	F2APSIGNM	-	F2ZXTOFM	F2VAOVVM	F2VAROVPM	F2WATTOVPM	F2PEAKIFM	F2PEAKVFM

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
	F2SAGFM	F2IF_OVRM	F2IF_FLLM	F2IF_EMPM	F2VF_OVRM	F2VF_FLLM	F2VF_EMPM

Таблица 167 – Описание бит регистра ADCUI\_F2MASK

Разряды	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15	F2APSIGNM	Маска бита F2APSIGNM
14, 13	-	Зарезервировано
12	F2ZXTOFM	Маска бита F2ZXTOF
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

Таблица 168 – Регистр ADCUI\_CCAL1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000	0000
	I0BGAIN	V0BGAIN

Таблица 169 – Описание бит регистра ADCUI\_CCAL1

Разряды	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

### 13.1.71 ADCUI\_CCAL2

Таблица 170 – Регистр ADCUI\_CCAL2

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000	0000
	I1BGAIN	V1BGAIN

Таблица 171 – Описание бит регистра ADCUI\_CCAL2

Разряды	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

### 13.1.72 ADCUI\_CCAL3

Таблица 172 – Регистр ADCUI\_CCAL3

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000	0000
	I2BGAIN	V2BGAIN

Таблица 173 – Описание бит регистра ADCUI\_CCAL3

Разряды	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

### 13.1.73 ADCUI\_CCAL4

Таблица 174 – Регистр ADCUI\_CCAL4

Номер	31...16	15...0
Доступ	U	R/W
Сброс	-	0000
	-	I3BGAIN

Таблица 175 – Описание бит регистра ADCUI\_CCAL4

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

### 13.1.74 ADCUI\_F0IRMS\_INACTIVE

Таблица 176 – Регистр ADCUI\_F0IRMS\_INACTIVE

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

Таблица 177 – Описание бит регистра ADCUI\_F0IRMS\_INACTIVE

Разряды	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

### 13.1.75 ADCUI\_F0WATTP\_L

Таблица 178 – Регистр ADCUI\_F0WATTP\_L

Номер	24...0
Доступ	RO
Сброс	
	F0WATTHRP_L

Таблица 179 – Описание бит регистра ADCUI\_F0WATTP\_L

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

### 13.1.76 ADCUI\_F0WATTN\_L

Таблица 180 – Регистр ADCUI\_F0WATTN\_L

Номер	24...0
Доступ	RO
Сброс	
	F0WATTHRN_L

Таблица 181 – Описание бит регистра ADCUI\_F0WATTN\_L

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

### 13.1.77 ADCUI\_F0VARP\_L

Таблица 182 – Регистр ADCUI\_F0VARP\_L

Номер	24..0
Доступ	RO
Сброс	
	F0VARHRP_L

Таблица 183 – Описание бит регистра ADCUI\_F0VARP\_L

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

### 13.1.78 ADCUI\_F0VARN\_L

Таблица 184 – Регистр ADCUI\_F0VARN\_L

Номер	24..0
Доступ	RO
Сброс	
	F0VARHRN_L

Таблица 185 – Описание бит регистра ADCUI\_F0VARN\_L

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

### 13.1.79 ADCUI\_F0VR\_L

Таблица 186 – Регистр ADCUI\_F0VR\_L

Номер	24...0
Доступ	RO
Сброс	
	F0VAHR_L

Таблица 187 – Описание бит регистра ADCUI\_F0VR\_L

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

### 13.1.80 ADCUI\_F1WATTP\_L

Таблица 188 – Регистр ADCUI\_F1WATTP\_L

Номер	24...0
Доступ	RO
Сброс	
	F1WATTHRP_L

Таблица 189 – Описание бит регистра ADCUI\_F1WATTP\_L

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

### 13.1.81 ADCUI\_F1WATTN\_L

Таблица 190 – Регистр ADCUI\_F1WATTN\_L

Номер	24...0
Доступ	RO
Сброс	
	F1WATTHRN_L

Таблица 191 – Описание бит регистра ADCUI\_F1WATTN\_L

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

### 13.1.82 ADCUI\_F1VARP\_L

Таблица 192 – Регистр ADCUI\_F1VARP\_L

Номер	24..0
Доступ	RO
Сброс	
	F1VARHRP_L

Таблица 193 – Описание бит регистра ADCUI\_F1VARP\_L

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

### 13.1.83 ADCUI\_F1VARN\_L

Таблица 194 – Регистр ADCUI\_F1VARN\_L

Номер	24..0
Доступ	RO
Сброс	
	F1VARHRN_L

Таблица 195 – Описание бит регистра ADCUI\_F1VARN\_L

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

### 13.1.84 ADCUI\_F1VR\_L

Таблица 196 – Регистр ADCUI\_F1VR\_L

Номер	24...0
Доступ	RO
Сброс	
	F1VAHR_L

Таблица 197 – Описание бит регистра ADCUI\_F1VR\_L

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

### 13.1.85 ADCUI\_F2WATTP\_L

Таблица 198 – Регистр ADCUI\_F2WATTP\_L

Номер	24...0
Доступ	RO
Сброс	
	F2WATTHRP_L

Таблица 199 – Описание бит регистра ADCUI\_F2WATTP\_L

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

### 13.1.86 ADCUI\_F2WATTN\_L

Таблица 200 – Регистр ADCUI\_F2WATTN\_L

Номер	24...0
Доступ	RO
Сброс	
	F2WATTHRN_L

Таблица 201 – Описание бит регистра ADCUI\_F2WATTN\_L

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24...0	F2WATTHRN_L	Младшие 25 битов внутреннего 57-битного аккумулятора отрицательной активной энергии. Необходимо производить чтение только после старшей части

### 13.1.87 ADCUI\_F2VARP\_L

Таблица 202 – Регистр ADCUI\_F2VARP\_L

Номер	24..0
Доступ	RO
Сброс	
	F2VARHRP_L

Таблица 203 – Описание бит регистра ADCUI\_F2VARP\_L

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

### 13.1.88 ADCUI\_F2VARN\_L

Таблица 204 – Регистр ADCUI\_F2VARN\_L

Номер	24..0
Доступ	RO
Сброс	
	F2VARHRN_L

Таблица 205 – Описание бит регистра ADCUI\_F2VARN\_L

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

### 13.1.89 ADCUI\_F2VR\_L

Таблица 206 – Регистр ADCUI\_F2VR\_L

Номер	24...0
Доступ	RO
Сброс	
	F2VAHR_L

Таблица 207 – Описание бит регистра ADCUI\_F2VR\_L

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

### 13.1.90 ADCUI\_LOAD

Таблица 208 – Регистр ADCUI\_LOAD

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

Таблица 209 – Описание бит регистра ADCUI\_LOAD

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...18	-	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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
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

### 13.1.91 ADCUI\_F0VRMS\_TRUE

Таблица 210 – Регистр ADCUI\_F0VRMS\_TRUE

Номер	31...29	28...0
Доступ	U	R/W
Сброс	-	0000_0000
	-	ADCUI_F0VRMS_TRUE

Таблица 211 – Описание бит регистра ADCUI\_F0VRMS\_TRUE

Разряды	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»

### 13.1.92 ADCUI\_F1VRMS\_TRUE

Таблица 212 – Регистр ADCUI\_F0VRMS\_TRUE

Номер	31...29	28...0
Доступ	U	R/W
Сброс	-	0000_0000
	-	ADCUI_F1VRMS_TRUE

Таблица 213 – Описание бит регистра ADCUI\_F1VRMS\_TRUE

Разряды	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»

### 13.1.93 ADCUI\_F2VRMS\_TRUE

Таблица 214 – Регистр ADCUI\_F0VRMS\_TRUE

Номер	31...29	28...0
Доступ	U	R/W
Сброс	-	0000_0000
	-	ADCUI_F2VRMS_TRUE

Таблица 215 – Описание бит регистра ADCUI\_F2VRMS\_TRUE

Разряды	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»

### 13.1.94 ADCUI\_MSC

Таблица 216 – Регистр ADCUI\_MSC

Номер	31	30..29	28	27	26	25...14	13...7	6...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	000	00	00
	F90_CO NF	F0VPHASE	F2ISEL_ HPF	F1ISEL_ HPF	F0ISEL_ HPF	F3IRMSOS	DMA_EN _RQ_S	DMA_EN _RQ_B

Таблица 217 – Описание бит регистра ADCUI\_MSC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	F90_CONF	Выбор напряжения для расчета активной и реактивной мощности: 0 – используется значение напряжения до фильтра HPF (режим legacy); 1 – используется значение напряжения после фильтра HPF (рекомендуемое значение)
30...29	F0VPHASE[1:0]	Младшие биты F0VPHASE См. ADCUI_F0CTRL
28	F2ISEL_HP	В FIFO F2IDAT записываются данные: 0 – взятые после HPF фильтра; 1 – взятые до HPF фильтра
27	F1ISEL_HP	В FIFO F1IDAT записываются данные: 0 – взятые после HPF фильтра 1 – взятые до HPF фильтра
26	F0ISEL_HP	В FIFO F0IDAT записываются данные: 0 – взятые после HPF фильтра; 1 – взятые до HPF фильтра
25...14	F3IRMSOS	Калибровка смещения для вычислителя среднеквадратического значения тока канала I3: 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 · 16 = 112 FF9: IRMSOS = -7 · 16 = -112
13...7	DMA_EN_RQ_S	Разрешение формирования запросов к DMA (одиночные транзакции) от каналов 7..1 $\Delta\Sigma$ АЦП
6...0	DMA_EN_RQ_B	Разрешение формирования запросов к DMA (блочные транзакции) от каналов 7..1 $\Delta\Sigma$ АЦП

### 13.1.95 ADCUI\_FILTERCFG

Таблица 218 – Регистр ADCUI\_FILTERCFG

Номер	31..26	25...24	23...12	11...0
Доступ	-	R/W	R/W	R/W
Сброс	-	0	803h	802h
	-	CHP_ADC_CFG	CLRMS_COEFF	CHPF_COEFF

Таблица 219 – Описание бит регистра ADCUI\_FILTERCFG

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...26	-	Зарезервировано
25...24	CHP_ADC_CFG	Настройка периода чоппера АЦП всех каналов: 0 – 256 периодов синхросигнала АЦП (рекомендуется при OSR = 256); 1 – 512 периодов синхросигнала АЦП (рекомендуется при OSR = 512); 2 – 1024 периода синхросигнала АЦП (рекомендуется при OSR = 1024)
23...12	CLRMS_COEFF	Коэффициент для фильтров LPF в блоках расчёта RMS, определяющий частоту среза. Значение по сбросу «-2046», что соответствует частоте среза 0,5 Гц при частоте дискретизации 4 кГц. Формула расчета $CLRMS\_COEFF = \frac{2^{11} \times \left( \tan \left( \pi \times \left( \frac{freq\_lpf}{samplerate\_adc} \right) \right) - 1 \right)}{\tan \left( \pi \times \left( \frac{freq\_lpf}{samplerate\_adc} \right) \right) + 1}$ где freq_lpf – необходимая частота среза фильтра; samplerate_adc – частота дискретизации АЦП. Результат необходимо округлить до целого числа со знаком. Результат не может быть меньше значения -2047.  Код для вычисления в САПР: CLRMS_COEFF=max(-2047, round(2^11*(tan(pi*(freq_lpf/samplerate_adc))-1)/(tan(pi*(freq_lpf/samplerate_adc))+1)));
11...0	CHPF_COEFF	Коэффициент для фильтров HPF во всех каналах АЦП, определяющий частоту среза. Значение по сбросу «-2045», что соответствует частоте среза 1 Гц при частоте дискретизации 4 кГц. Рекомендованное значение «-2016». Формула расчета $CHPF\_COEFF = \frac{2^{11} \times \left( \tan \left( \pi \times \left( \frac{freq\_hpf}{samplerate\_adc} \right) \right) - 1 \right)}{\tan \left( \pi \times \left( \frac{freq\_hpf}{samplerate\_adc} \right) \right) + 1}$ где freq_hpf – необходимая частота среза фильтра, а samplerate_adc – частота дискретизации АЦП. Результат необходимо округлить до целого числа со знаком.  Код для вычисления в САПР: CHPF_COEFF=round(2^11*(tan(pi*(freq_hpf/samplerate_adc))-1)/(tan(pi*(freq_hpf/samplerate_adc))+1));

### 13.1.96 ADCUI\_CLKPHASE1

Таблица 220 – Регистр ADCUI\_CLKPHASE1

Номер	31...30	29...20	19...10	9..0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	200h	200h	200h
	CLC_ADC_CFG	I2PHASE	I1PHASE	I0PHASE

Таблица 221 – Описание бит регистра ADCUI\_CLKPHASE1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...30	CLC_ADC_CFG	Настройка частоты тактирования АЦП для всех каналов: 00 – 4 МГц; 01 – 2 МГц; 10 – 1 МГц; 11 – 1 МГц. Эта частота также выводится на PD14 в режиме использования внешних дельта-сигма модуляторов
29...20	I2PHASE[9:0]	Фазовый сдвиг канала тока I2. От 0 · (такт АЦП) до +1023 · (такт АЦП). «1000000000» соответствует синфазному сигналу с напряжением V2. При использовании DCHPENI2 = 1 сдвигать фазу рекомендуется с шагом 32
19...10	I1PHASE[9:0]	Фазовый сдвиг канала тока I1. От 0 · (такт АЦП) до +1023 · (такт АЦП). «1000000000» соответствует синфазному сигналу с напряжением V1. При использовании DCHPENI1 = 1 сдвигать фазу рекомендуется с шагом 32
9...0	I0PHASE[9:0]	Фазовый сдвиг канала тока I0. От 0 · (такт АЦП) до +1023 · (такт АЦП). «1000000000» соответствует синфазному сигналу с напряжением V0. При использовании DCHPENI0 = 1 сдвигать фазу рекомендуется с шагом 32

### 13.1.97 ADCUI\_CLKPHASE2

Таблица 222 – Регистр ADCUI\_CLKPHASE2

Номер	31...26	9..0
Доступ	-	R/W
Сброс	-	200h
	-	I3PHASE

Таблица 223 – Описание бит регистра ADCUI\_CLKPHASE2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...26	-	Зарезервировано
9...0	I3PHASE[9:0]	Фазовый сдвиг канала тока I3. От 0 · (такт АЦП) до +1023 · (такт АЦП). «1000000000» соответствует синфазному сигналу с напряжением V0. При использовании DCHPENI3 = 1 сдвигать фазу рекомендуется с шагом 32

## 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 20, 21 приведены два вида включения АЦП: полностью дифференциальное и недифференциальное включение.

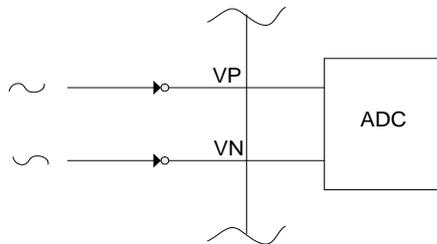


Рисунок 20 – Дифференциальное включение

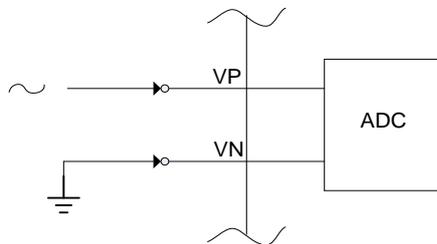


Рисунок 21 – Недифференциальное включение

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 22.

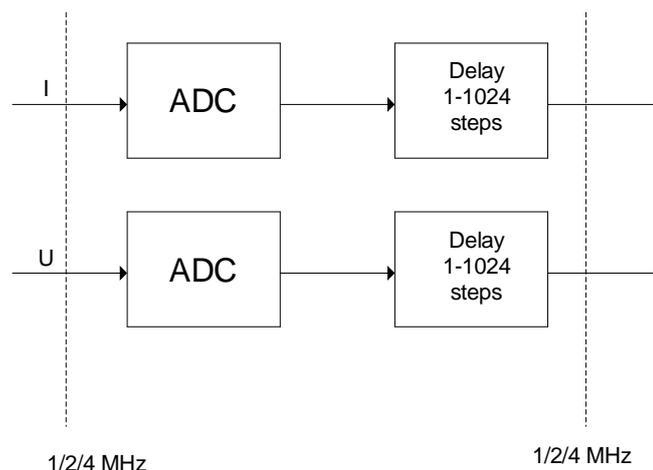


Рисунок 22 – Контролируемый фазовый сдвиг в каналах тока и напряжения

Изменяя линию задержки в канале напряжения можно регулировать временной сдвиг одного канала относительно другого. Так как частота отсчетов после АЦП равна 1/24 МГц, то один шаг при частоте входного сигнала 50 Гц равен 0,018 °/0,009 °/0,0045 °.

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  $\sin c^3$ , его характеристики приведены на рисунках 23, 24.

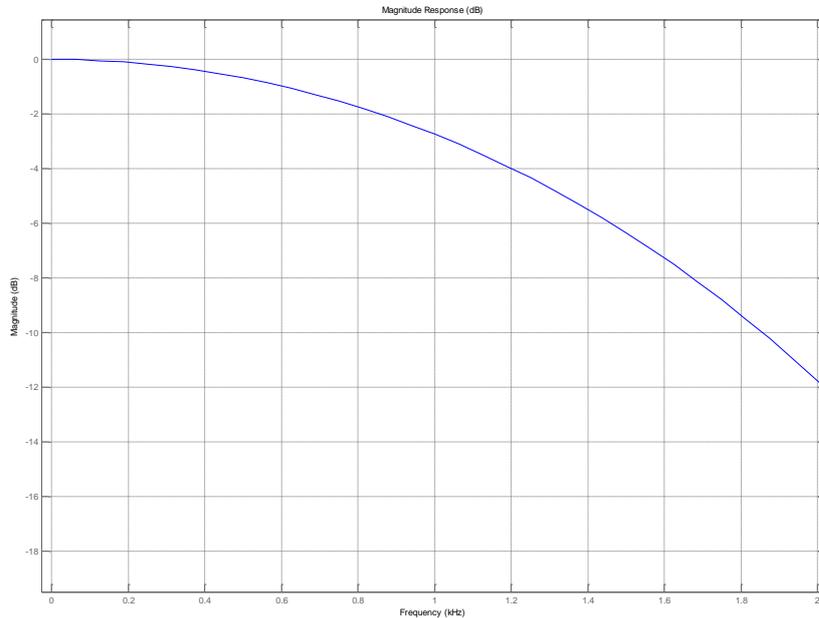


Рисунок 23 – Характеристика децимирующего фильтра в полосе 2 кГц

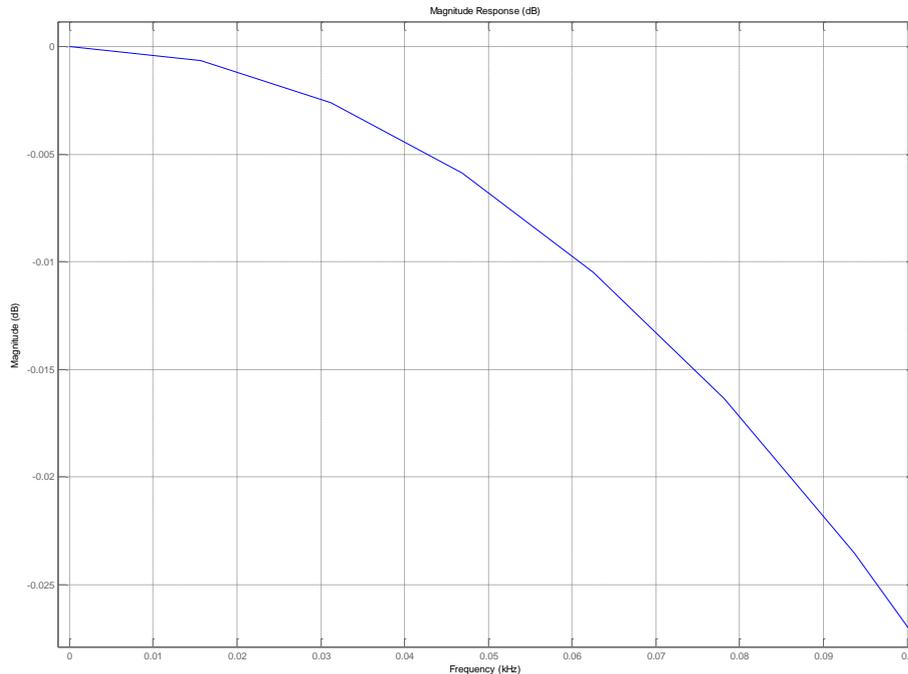


Рисунок 24 – Характеристика децимирующего фильтра в полосе 100 Гц

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.

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 224. Значения приведены для усиления PGA равного 0 дБ. Отсчеты, записанные в FIFO, представлены в двоичном формате с дополнением до 2.

Таблица 224 – Формулы расчета входного напряжения

Режим	Входное напряжение АЦП, В	
Дифференциальное включение, 24 бит. режим	$\frac{FxV\text{DAT} \cdot U\text{REFO}}{2^{23}}$	$\frac{FxI\text{DAT} \cdot U\text{REFO}}{2^{23}}$
Дифференциальное включение, 16 бит. режим	$\frac{FxV\text{DAT} \cdot U\text{REFO}}{2^{15}}$	$\frac{FxI\text{DAT} \cdot U\text{REFO}}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{FxV\text{DAT} \cdot U\text{REFO}}{2^{23}}$	$\frac{FxI\text{DAT} \cdot U\text{REFO}}{2^{23}}$
Недифференциальное включение, 16 бит. режим	$\frac{FxV\text{DAT} \cdot U\text{REFO}}{2^{15}}$	$\frac{FxI\text{DAT} \cdot U\text{REFO}}{2^{15}}$

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.

$$I_{cor} = I_{ADC} \cdot 2^{FxIyGAIN} \cdot 2^{IxGAIN} \cdot \left(1 + \frac{IxBGAIN}{2^{11}}\right). \quad (1)$$

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.

$$V_{cor} = V_{ADC} \cdot 2^{FxVGAIN} \cdot 2^{VxGAIN} \cdot \left(1 + \frac{VxBGAIN}{2^{11}}\right). \quad (2)$$

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

Таблица 225 – Значения среднеквадратических величин напряжения

Режим	Напряжение, В	
Дифференциальное включение	$\frac{F_xVRMS \cdot U_{REFO}}{2^{23}}$	$\frac{F_xIRMS \cdot U_{REFO}}{2^{23}}$
Недифференциальное включение	$\frac{F_xVRMS \cdot U_{REFO}}{2^{22}}$	$\frac{F_xIRMS \cdot U_{REFO}}{2^{22}}$

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 так же 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 4 бита влево. 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  $V^2(t)$ .

$$F_xVRMS = F_xVRMS + F_xVRMSOS * 16 . \quad (5)$$

Значения FxVRMSOS и FxIRMSOS представлены в виде знаковых величин в двоичном коде с дополнением до 2. В рамках калибровки канала рекомендуется в качестве FxVRMSOS устанавливать среднее значение регистра FxVRMS при закороченных входах АЦП соответствующего канала, за интервал не менее 4096 отчетов после установления фильтров HPF и LPF. Определять его для всех 4 режимов PGA. Подставлять значения в регистр в зависимости от режима PGA и с учетом цифрового усиления IxGAIN, VxGAIN, IxBGAIN, VxBGAIN

В регистрах FxVRMS2 и FxIRMS2 хранятся значения среднеквадратического значения напряжения и тока до извлечения квадратного корня. В таблице 226 приведены значения квадратов среднеквадратических величин.

Таблица 226 – Значения квадратов среднеквадратических величин

Режим	Напряжение, В <sup>2</sup>	
Дифференциальное включение	$\frac{F_x VRMS2 \cdot U_{REFO}}{2^{30}}$	$\frac{F_x IRMS2 \cdot U_{REFO}}{2^{30}}$
Недифференциальное включение	$\frac{F_x VRMS2 \cdot U_{REFO}}{2^{28}}$	$\frac{F_x IRMS2 \cdot U_{REFO}}{2^{28}}$

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 25 приведена его фазовая характеристика.

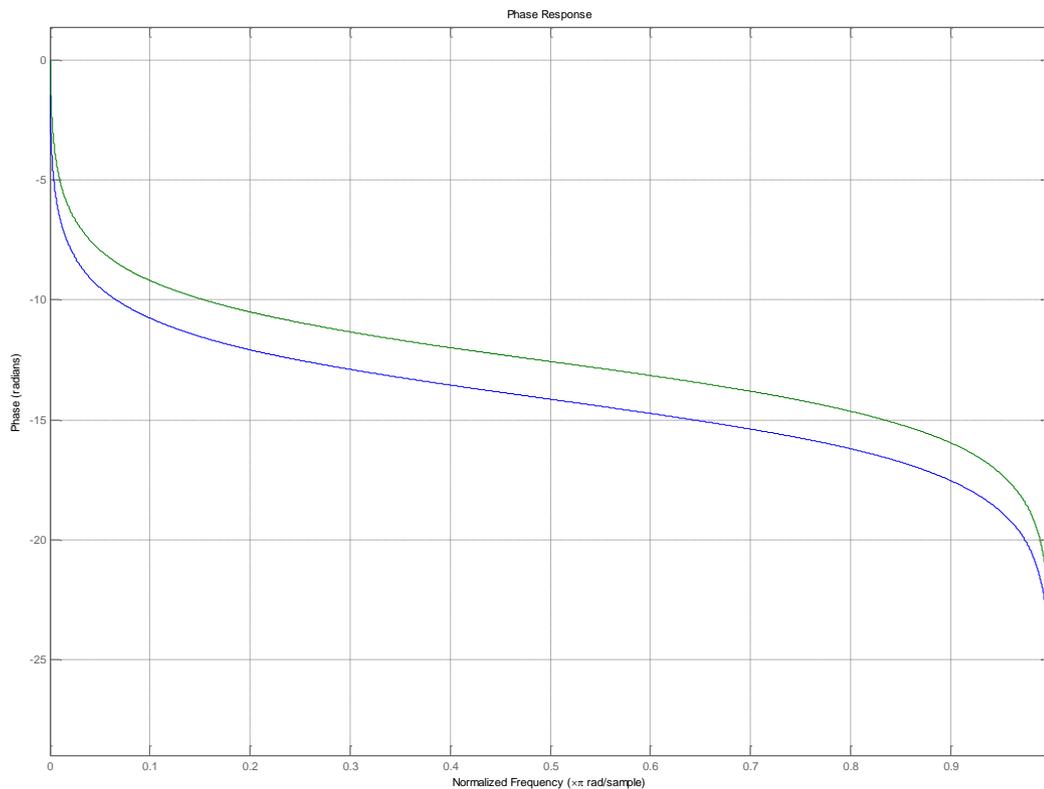


Рисунок 25 – Фильтр для сдвига сигнала на 90 °

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

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

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

Тогда мгновенное значение мощности равно произведению тока на напряжение

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

Среднее значение мощности за целое количество тактов равно

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

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 ( $FxISEL=01/10/11$ ,  $FxVSEL=01/10/11$ ), 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 227 – 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

$$P_{cor} = \left( P + \frac{P_{os}}{2^{23}} \right) \cdot \left( 1 + \frac{P_{gain}}{2^{11}} \right). \quad (10)$$

Вычисленная мощность после калибровки накапливается в регистре аккумулятора. Для каждой из трех мощностей есть свой аккумулятор. Значение в них определяет потребленную энергию. В таблице 228 приведена формула перевода значения в Вт·с.

Таблица 228 – Формула перевода значения в Вт·с

Режим	Энергия, Вт·с	
Дифференциальное включение	$\frac{FxWATTHR P}{512 \cdot 4000}$ ,	$\frac{FxWATTHR N}{512 \cdot 4000}$
Недифференциальное включение	$\frac{FxWATTHR P}{512 \cdot 1000}$ ,	$\frac{FxWATTHR N}{512 \cdot 1000}$

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

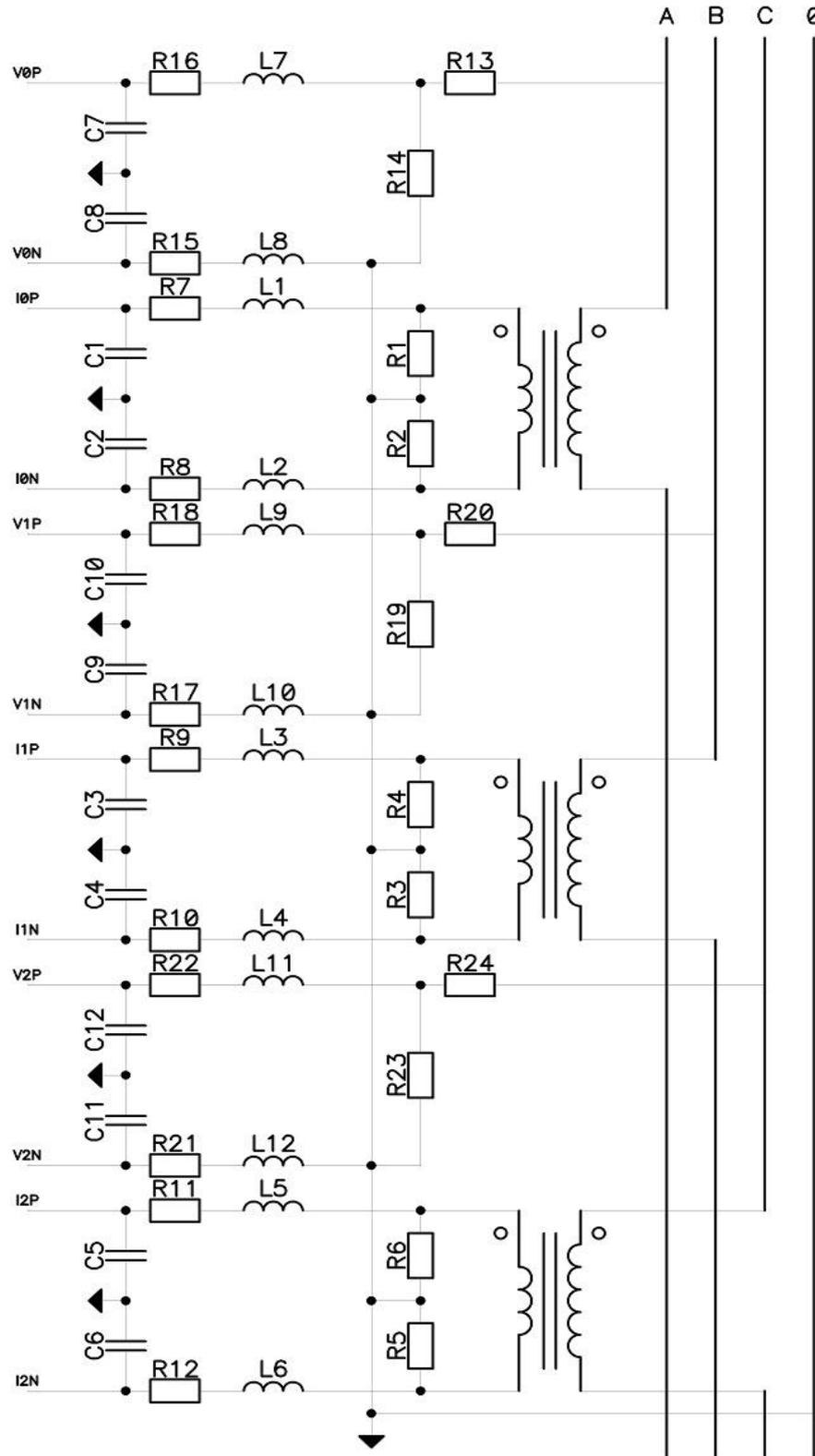


Рисунок 26 – Типовая схема включения для учета электроэнергии по трем фазам

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,  $R_1+R_2$ , 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

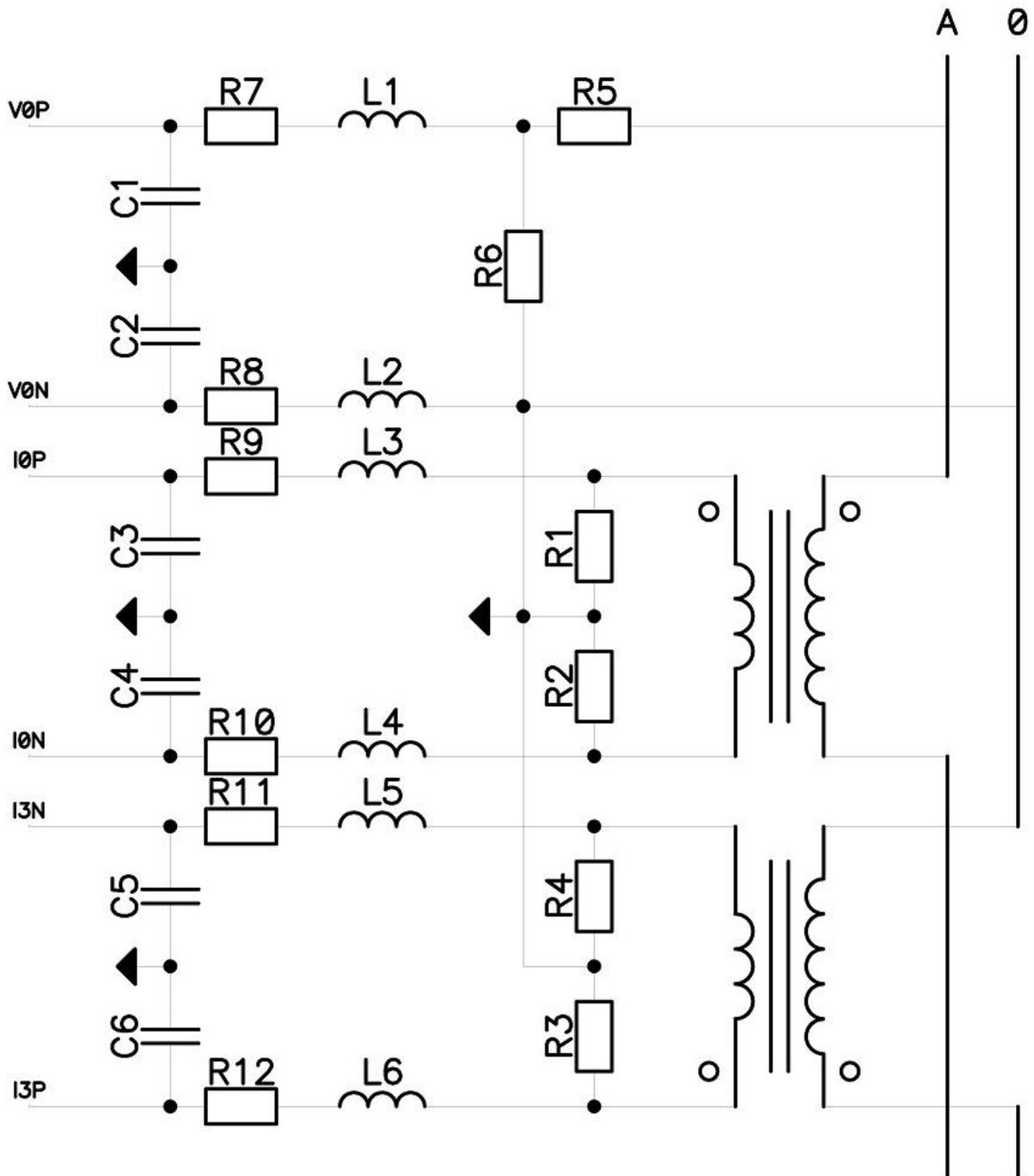


Рисунок 27 – Типовая схема включения для учета электроэнергии по одной фазе

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

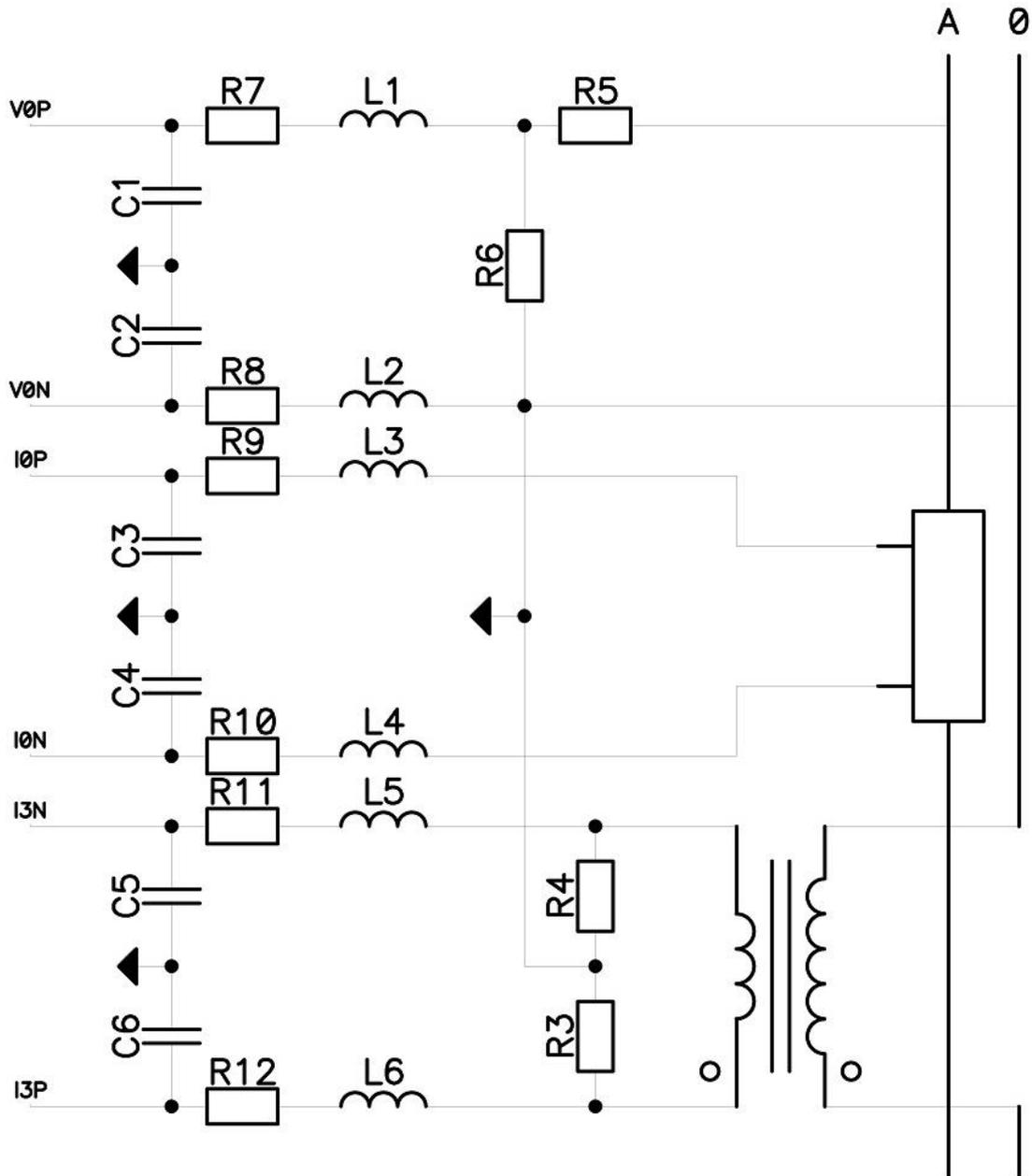


Рисунок 28 – Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта

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

Микроконтроллер 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 2 бита / PCLK (частота APB шины). Контроллер имеет FIFO на 4 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 229 – Описание регистров управления блока CRC

Базовый Адрес	Название	Описание
0x5009_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 230 – 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 231 – 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 232 – 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 233 – 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\_DATA1

Таблица 234 – Регистр CRC\_DATA1

Номер	31...0
Доступ	W
Сброс	
	DATA_IN

Таблица 235 – Описание бит регистра CRC\_DATA1

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 236 – Регистр CRC\_VAL

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

Таблица 237 – 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	Расчитанное значение /начальное значение. Начальное значение нужно записывать, когда блок отключен или, когда закончено преобразование

### 15.1.5 CRC\_POL

Таблица 238 – Регистр CRC\_POL

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

Таблица 239 – 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

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

Микроконтроллер имеет два встроенных генератора, два внешних осциллятора, а также специализированный блок формирования тактовой синхронизации микроконтроллера.

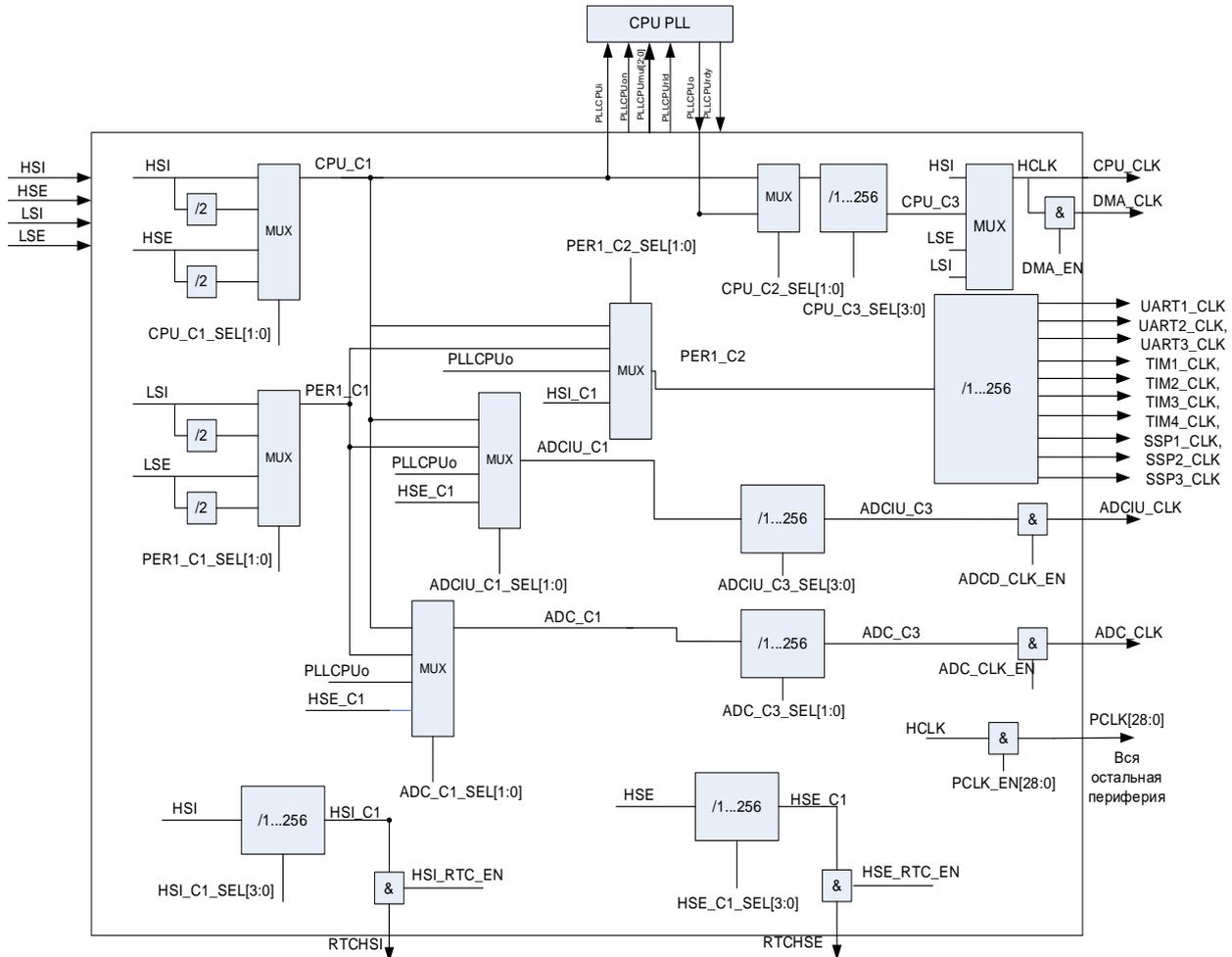


Рисунок 29 – Структурная схема

### 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 `hsirdy` в регистре батарейного домена `BKP_CLK`. Первоначально процессорное ядро запускается на тактовой частоте `HSI/16`. При дальнейшей работе генератор HSI может быть отключен при помощи сигнала `hsion` в регистре `BKP_CLK`. Также генератор может быть подстроен при помощи сигнала `hstrim` в регистре `BKP_CLK`.

### 16.2 Встроенный RC-генератор LSI

Генератор LSI вырабатывает тактовую частоту 32 кГц. Генератор включается сигналом `lsion` в регистре `BKP_CLK` и при выходе в нормальный режим работы вырабатывает сигнал `lsirdy` в регистре `BKP_CLK`.

### 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 48 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 lseon в регистре BKP\_CLK. При выходе в нормальный режим работы вырабатывает сигнал lserdy в регистре BKP\_CLK.

Также осциллятор может работать в режиме lsebyp, когда входная тактовая частота с входа OSC\_IN32 проходит напрямую на выход LSE. Допустимый диапазон частот в режиме lsebyp - от 20 до 40 кГц. Выход OSC\_OUT32 находится в этом режиме в третьем состоянии.

Так как генератор LSE питается от напряжения питания  $U_{CCV}$  и его регистр управления BKP\_CLK расположен в батарейном домене, то генератор может продолжать работать при пропадании основного питания  $U_{CC}$ . 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 48 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.

Управление тактовыми частотами ведется через периферийный блок RST\_CLK. При включении питания микроконтроллер запускается на частоте HSI/16 генератора. Выдача тактовых сигналов синхронизации для всех периферийных блоков кроме RST\_CLK отключена. Для начала работы с нужным периферийным блоком необходимо включить его тактовую частоту в регистре 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 мультиплексорах управляемом регистре CPU\_CLOCK.

## 16.6 Описание регистров блока контроллера тактовой частоты

Таблица 240 – Описание регистров блока контроллера тактовой частоты

Базовый Адрес	Название	Описание
0x5002_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 $\Delta\Sigma$ АЦП
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 .
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 .

### 16.6.1 CLOCK\_STATUS

Таблица 241 – 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 242 – Описание бит регистра CLOCK\_STATUS

Разряды	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 243 – Регистр PLL\_CONTROL

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

Таблица 244 – Описание бит регистра PLL\_CONTROL

Разряды	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: $PLL_{CPU0} = PLL_{CPUi} \times (PLL_{CPUMUL} + 1)$
7...4	-	Зарезервировано
3	PLL CPU SEL	Бит перезапуска выбора входной частоты PLL: 0 – частота с генератора HSE; 1 – частота задаётся с помощью бит CPU_C1_SEL[1:0] регистра CPU_CLOCK
2	PLL CPU ON	Бит включения PLL: 0 – PLL выключена; 1 – PLL включена
1	-	Зарезервировано
0	-	Зарезервировано

### 16.6.3 HS\_CONTROL

Таблица 245 – Регистр HS\_CONTROL

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

Таблица 246 – Описание битов регистра HS\_CONTROL

Разряды	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. Для работы в режиме внешнего генератора (bypass) бит hseon должен быть равен 1
0	HSEON	Бит управления HSE осциллятором: 0 – выключен; 1 – включен

### 16.6.4 CPU\_CLOCK

Таблица 247 – 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	1011	0	0	00
	-	HCLK SEL[1:0]	CPU C3 SEL[3:0]	-	CPU C2 SEL	CPU C1 SEL[1:0]

Таблица 248 – Описание бит регистра CPU\_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	-	Зарезервировано
9, 8	HCLK SEL[1:0]	Биты выбора источника для HCLK: 00 – HSI; 01 – CPU_C3; 10 – LSE; 11 – LSI
7...4	CPU C3 SEL[3:0]	Биты выбора делителя для CPU_C3: 0xxx – CPU_C3 = CPU_C2; 1000 - CPU_C3 = CPU_C2 / 2; 1001 - CPU_C3 = CPU_C2 / 4; 1010 - CPU_C3 = CPU_C2 / 8; ... 1111 - CPU_C3 = CPU_C2 / 256

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	-	Зарезервировано
2	CPU C2 SEL	Биты выбора источника для CPU_C2: 0 – CPU_C1; 1 – PLLCPU <sub>0</sub>
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 249 – 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 250 – Описание бит регистра PER1\_CLOCK

Разряды	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 – PLLCPU <sub>0</sub> ; 11 – HSI_CLK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 – PLLCPU0; 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 251 – 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	ADCUI LK EN	ADC C3 SEL[3:0]	ADCUI C3 SEL[3:0]	ADCUI C1 SEL[1:0]	ADC C1 SEL[1:0]

Таблица 252 – Описание бит регистра ADC\_CLOCK

Разряды	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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
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 рекомендуется выбирать источник HSE = 8,192 МГц без дополнительных делителей
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 253 – 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 SEL[1:0]	HSE SEL[1:0]

Таблица 254 – Описание бит регистра RTC\_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	-	Зарезервировано
9	HSI RTC EN	Бит разрешения HSI RTC: 0 – запрещен; 1 - разрешен
8	HSE RTC EN	Бит разрешения HSE RTC: 0 – запрещен; 1 - разрешен

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7...4	HSI SEL[3:0]	Биты выбора делителя для HSI_C1: 0xxx – HSI_C1 = HSI; 1000 - HSI_C1 = HSI / 2; 1001 - HSI_C1 = HSI / 4; 1010 - HSI_C1 = HSI / 8; ... 1111 - HSI_C1 = HSI / 256
3...0	HSE SEL[3:0]	Биты выбора делителя для HSE_C1: 0xxx – HSE_C1 = HSE; 1000 - HSE_C1 = HSE / 2; 1001 - HSE_C1 = HSE / 4; 1010 - HSE_C1 = HSE / 8; ... 1111 - HSE_C1 = HSE / 256

### 16.6.8 PER2\_CLOCK

Таблица 255 – Регистр PER2\_CLOCK

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

Таблица 256 – Описание бит регистра PER2\_CLOCK

Разряды	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 PCLK[14] – TIMER1 PCLK[15] – TIMER2 PCLK[16] – PORTA PCLK[17] – PORTB

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		PCLK[18] – PORTC PCLK[19] – CRC PCLK[20] – зарезервировано PCLK[21] – зарезервировано PCLK[22] – зарезервировано 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

Таблица 257 – Регистр DMA\_DONE\_STICK

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

Таблица 258 – Описание бит регистра DMA\_DONE\_STICK

Разряды	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. Сбрасываются только записью нуля

### 16.6.10 TIM\_CLOCK

Таблица 259 – 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 260 – Описание бит регистра TIM\_CLOCK

Разряды	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
* Частота тактирования таймера TIMx_CLK должна быть менее или равна частоте тактирования ядра CPU_CLK		

16.6.11 UART\_CLOCK

Таблица 261 – 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	UART4 BRG [2:0]	UART3 CLK EN	UART2 CLK EN	UART 1 CLK EN	UART 3 BRG [7:0]	UART 2 BRG [7:0]	UART 1 BRG [7:0]

Таблица 262 – Описание бит регистра UART\_CLOCK

Разряды	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
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 263 – 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 264 – Описание бит регистра SSP\_CLOCK

Разряды	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
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

Таблица 265 – Регистр DIV\_SYS\_TIM

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

Таблица 266 – 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  $U_{CC}$  в блоке SW происходит автоматическое переключение питания  $U_{CCVB}$  с  $U_{CC}$  на  $U_{CCVB}$ . Если на  $U_{CCVB}$  имеется отдельный источник питания (батарея), то батарейный домен остается включенным и может выполнять свои функции.

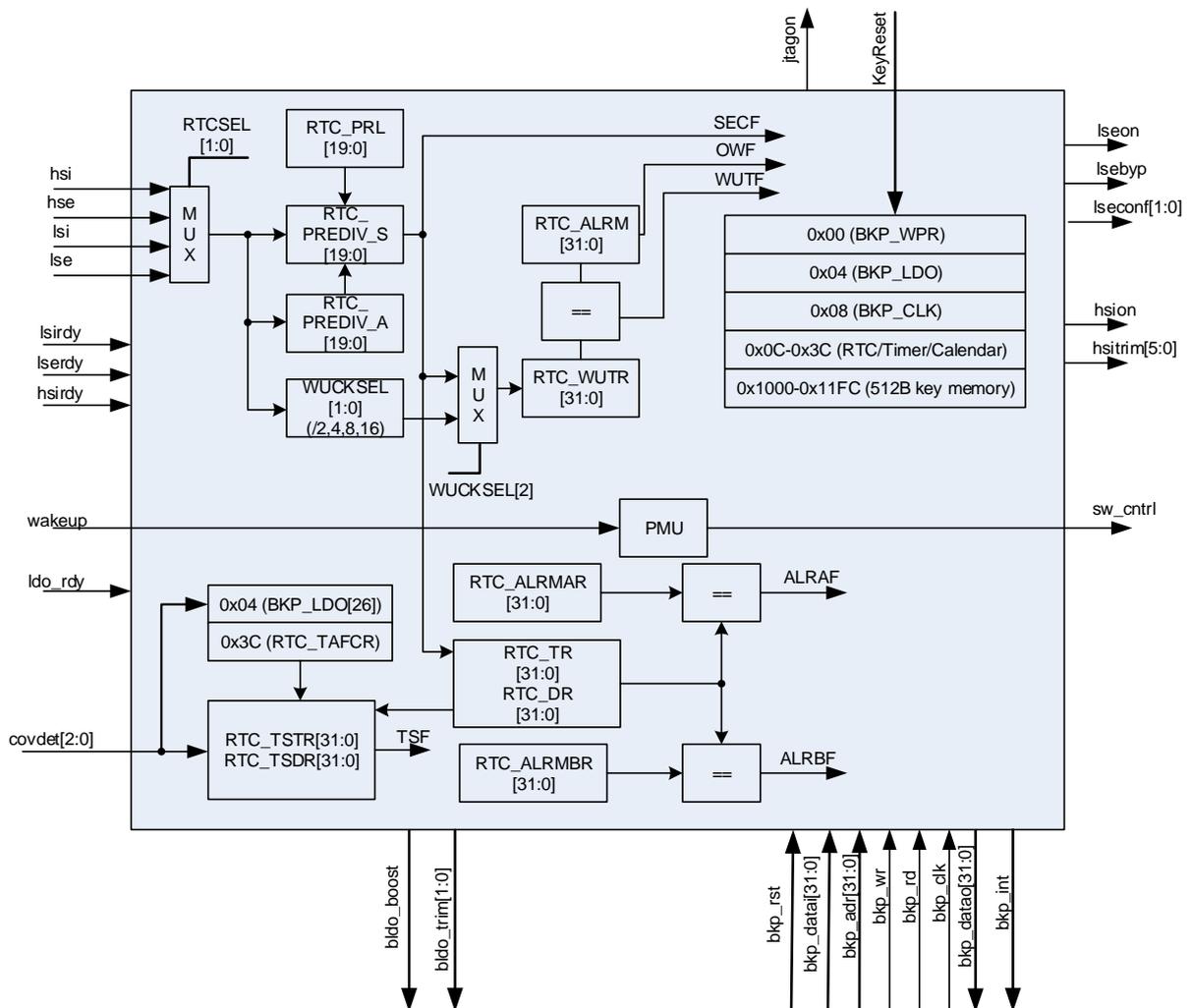


Рисунок 30 – Структурная схема

### 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 RTCEN. В качестве источника тактовой частоты часов реального времени может выступать генератор LSI, осциллятор LSE, входы hse или hsi с дополнительным делителем до 256 (hse и hsi формируются в блоке

управления тактовыми частотами и могут быть выбраны только при наличии питания цифрового ядра, формируемого встроенным регулятором напряжения, LSI может быть выбран при наличии питания  $U_{CC}$ , LSE может быть выбран при наличии  $U_{CCV}$ ). Выбор между источниками осуществляется битами RTCSEL. При возможном отключении основного источника питания  $U_{CC}$  в качестве источника тактовой частоты 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 так же не определены и могут принять любое значение.

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

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

Основу модуля часов реального времени составляет двоично-десятичный таймер/счетчик. Результат счета отображается в двух 32-разрядных регистрах. Первый из них RTC\_TR содержит информацию о времени, второй – RTC\_DR представляет собой календарь, включающий год, месяц, день недели и дату. Все данные представлены в BCD формате, 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.

Среди функций модуля RTC следует отметить два отдельных регистра событий (RTC\_ALRMAR и RTC\_ALRMBR), с помощью которых можно реализовать будильники. Формат регистров событий аналогичен счетным регистрам, что несколько упрощает их программирование. Сторожевой таймер RTC\_WUTR работает независимо от часов/календаря на выходной частоте делителя RTC\_PREDIV\_S, либо независимой частоте, задаваемой с помощью бит WUCKSEL. Регистр RTC\_ALRM предназначен для задания времени, при совпадении с которым вырабатывается флаг прерывания.

Еще одной полезной функцией может оказаться функция TimeStamp, предназначенная для определения точного времени наступления внешнего события. Для ее активации также требуется наличие внешнего сигнала, по фронту которого, значения счетных регистров сохраняются в регистрах RTC\_TSTR и RTC\_TSDR.

В батарейном домене реализована возможность мониторинга входных сигналов WAKEUP с помощью функции Tamper (регистр RTC\_TAFCR). Во внутреннем регистре записывается контролируемый активный уровень события (передний фронт и высокий уровень или задний фронт и низкий уровень), и если сигнал на входе станет идентичным записанному, то это событие регистрируется в управляющем бите. С

помощью бита `sw_cntrl` регистра `BKP_LDO` присутствует возможность разрешить выдачу сигнала управления внешним ключом на вывод `SW`. Внешний ключ замыкает питание батареи на основное питание микросхемы в случае его отсутствия и фиксации на любом из входов `WAKEUP` активного высокого или низкого уровня (в случае отключения функции `Tamper`, только низкого уровня на любом из входов `WAKEUP`). Размыкание ключа осуществляется либо при срабатывании сторожевого таймера `RTC_WUTR` (`WUTF=1`), либо при записи единицы в бит `sw_off` регистра `BKP_LDO`.

## 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 ёмкостью 512 байт.

## 17.3 Описание регистров блока батарейного домена

Таблица 267 – Описание регистров блока батарейного домена

Базовый Адрес	Название	Описание
0x5006_0000	BKP	Контроллер батарейного домена и часов реального времени
Смещение		
0x00	BKP_WPR	Регистр ключа для разрешения работы с регистрами блока.
0x04	BKP_LDO	Регистр управления блоком LDO
0x08	BKP_CLK	Регистр управления блоками LSE, HSE, LSI и HSI
0x0C	BKP_RTC	Регистр управления блоком RTC
0x10	RTC_WUTR	Регистр счетчика выхода из спящего режима
0x14	RTC_PREDIV_S	Регистр предварительного делителя основного счетчика/календаря
0x18	RTC_PRL	Регистр основания счета предварительного делителя основного счетчика/календаря
0x1C	RTC_ALRM	Регистр значения для сравнения счетчика выхода из спящего режима и выработки сигнала ALRF
0x20	RTC_CS	Регистр управления и состояния флагов часов реального времени
0x24	RTC_TR	Регистр времени счетчика/календаря в BCD формате
0x28	RTC_DR	Регистр даты счетчика/календаря в BCD формате
0x2C	RTC_ALRMAR	Регистр задания события А – будильника с календарным временем
0x30	RTC_ALRMBR	Регистр задания события В – будильника с календарным временем
0x34	RTC_TSTR1	Регистр времени внешнего события входа 1
0x38	RTC_TSTR2	Регистр времени внешнего события входа 2
0x3C	RTC_TSTR3	Регистр времени внешнего события входа 3
0x40	RTC_TSDR1	Регистр даты внешнего события входа 1

0x44	RTC_TSDR2	Регистр даты внешнего события входа 2
0x48	RTC_TSDR3	Регистр даты внешнего события входа 3
0x4C	RTC_TAFCR	Регистр управления обнаружением событий
0x50	RTC_TMPICAL1	Регистр калибровки и управления блоком термокомпенсации частоты LSE
0x54	RTC_TMPICAL2	Регистр калибровки и управления блоком термокомпенсации частоты LSE
0x58	RTC_TMPICAL3	Регистр калибровки и управления блоком термокомпенсации частоты LSE
0x1000-0x11FC	ВКР_MEM	Память ОЗУ 512 байт

### 17.3.1 ВКР\_MEM (128x32)

Таблица 268 – Память ВКР\_MEM

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

Таблица 269 – Описание бит регистра ВКР\_MEM

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

### 17.3.2 ВКР\_WPR

Таблица 270 – Память ВКР\_WPR

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

Таблица 271 – Описание бит регистра ВКР\_WPR

Разряды	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 272 – Регистр ВКР\_LDO

Номер	31..30	29	28, 27	26	25
Доступ	U	R/C1	R/W	R/C1	R/W
Сброс	0	0	00	0	0
	-	sw_off	mode	covdet	fpor

Номер	15..10	9..7	6	5	4, 3	2	1, 0
Доступ	U	R	R/W	R/W	R/W	R/W	R/W
Сброс	0	000	1	0	00	000	000
	-	WAKEUP 3-1	jtagon	sw_cntrl	ldo_boost	bldo_boost	bldo_trim

Таблица 273 – Описание бит регистра ВКР\_LDO

Разряды	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 единицу. По чтению содержит состояние вывода SW при sw_cntrl = 1
28, 27	mode[1:0]	При использовании загрузочной программы, описанной в подразделе 9.1: режим запуска микросхемы, полученный с входов MODE[1:0] (PC[0], PC[1]).  При использовании загрузочной программы, отличной от описанной в подразделе 9.1, данное поле может применяться по усмотрению пользователя
26	covdet	Признак несанкционированного вскрытия устройства по входам WAKEUP3-WAKEUP1: 0 – вскрытия не было; 1 – осуществлялось вскрытие. Для сброса факта вскрытия осуществить запись covdet = 1. Проверить сброс факта вскрытия чтением бита covdet = 0 или чтением бит TAMPINSEL[2:0] = 0
25	fpor	При использовании загрузочной программы, описанной в подразделе 9.1: флаг срабатывания POR. При сбросе по питанию устанавливается в 0. Может использоваться расположенной в области памяти BOOT загрузочной программой, описанной в подразделе 9.1, 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 9.1, данное поле может применяться по усмотрению пользователя

Разряды	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	jtagon	Разрешение работы порта JTAG: 0 – запрещен; 1 – разрешен
5	sw_cntrl	Бит разрешения аварийного питания микросхемы от батареи при наличии активного постоянного высокого или низкого уровня на любом из входов WAKEUP1- WAKEUP3 1 – разрешено 0 – запрещено
4, 3	ldo_boost	Вход активации boost режима работы основного регулятора: 00 – типовое значение выхода регулятор; 01...11 – увеличение выходного значения регулятора напряжения
2	bldo_boost	Вход активации boost режима работы регулятора батарейного домена: 0 – типовое значение выхода регулятор; 1 – увеличение выходного значения регулятора напряжения с помощью bldo_trim
1, 0	bldo_trim[1:0]	Подстройка регулятора питания батарейного домена: 00 – типовое значение; 01...11 – увеличение опорного напряжения

### 17.3.4 ВКР\_CLK

Таблица 274 – Регистр ВКР\_CLK

Номер	31..21	20..15	14
Доступ	U	R/W	RO
Сброс	0	000000	1
	-	hsitrim	hsirdy

Номер	13	12	11...6	5	4, 3	2	1	0
Доступ	R/W	RO	U	R/W	R/W	RO	R/W	R/W
Сброс	1	0	0	0	00	0	0	0
	hsion	lsirdy	-	lsion	lsecnf	lserdy	lsebyp	lseon

Таблица 275 – Описание бит регистра ВКР\_CLK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..21	-	Зарезервировано
20...15	hsitrim[5:0]	Коэффициент подстройки частоты генератора HSI. Уточняется по результатам исследования
14	hsirdy	Флаг выхода генератора HSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
13	hsion	Бит управления генератором HSI: 1 – генератор включен; 0 – генератор выключен
12	lsirdy	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	lsion	Бит управления генератором LSI: 0 – генератор выключен; 1 – генератор включен
4, 3	lseconf	Биты конфигурации LSE генератора. В случае возникновения проблем с запуском при дефолтном значении конфигурации lseconf[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	lserdy	Флаг выхода генератора LSE в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
1	lsebyp	Бит управления генератором LSE: 0 – режим осциллятора; 1 – режим работы на проход (внешний генератор)
0	lseon	Бит управления генератором LSE: 0 – генератор выключен; 1 – генератор включен

### 17.3.5 ВКР\_RTC

Таблица 276 – Регистр ВКР\_RTC

Номер	25	24...15	14...12	11	10...3	2	1, 0
Доступ	R/W	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	000	0	0000000	0	00
	FMT	-	WUCKSEL	RTCRESET	RTCCAL	RTCEN	RTCSEL

Таблица 277 – Описание бит регистра ВКР\_RTC

Разряды	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
14	WUCKSEL[2]	Биты выбора независимой синхронизации для счетчика выхода из спящего режима: WUCKSEL[2] = 1 – общее тактирование с счётчиком/календарём; WUCKSEL[2] = 0 – независимое тактирование с выбранного делителя
13...12	WUCKSEL[1:0]	Биты выбора делителя частоты для счетчика выхода из спящего режима: WUCKSEL[1:0] = 3 – делитель на 2; WUCKSEL[1:0] = 2 – делитель на 4; WUCKSEL[1:0] = 1 – делитель на 8; WUCKSEL[1:0] = 0 – делитель на 16
11	RTCRESET	Сброс часов реального времени: 0 – часы не сбрасываются; 1 – часы сбрасываются
10...3	RTCCAL	Коэффициент подстройки тактовой частоты часов реального времени, из каждых $2^{20}$ тактов будет замаскировано CAL тактов: 00000000 – 0 тактов; 00000001 – 1 такт; .... 11111111 – 256 тактов. Таким образом, при частоте 32768,00000 Гц: при CAL = 0 тактов, частота = 32768,00000 Гц; при CAL = 1 такт, частота = 32767,96875 Гц; ... при CAL = 256 тактов, частота = 32760,00000 Гц
2	RTCEN	Бит разрешения работы часов реального времени: 0 – работа запрещена; 1 – работа разрешена
1, 0	RTCSEL[1:0]	Биты выбора источника тактовой синхронизации часов реального времени: 00 – lsi; 01 – lse; 10 – hsi; 11 – hse

### 17.3.6 RTC\_WUTR

Таблица 278 – Регистр RTC\_WUTR

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

Таблица 279 – Описание бит регистра RTC\_WUTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	RTC WUTR[31:0]	Значение основного счетчика выхода из спящего режима

### 17.3.7 RTC\_PREDIV\_S

Таблица 280 – Регистр RTC\_PREDIV\_S

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

Таблица 281 – Описание бит регистра RTC\_PREDIV\_S

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

### 17.3.8 RTC\_PRL

Таблица 282 – Регистр RTC\_PRL

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

Таблица 283 – Описание бит регистра RTC\_PRL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	-	-
19...0	RTC PRL[19:0]	Значение основания для счета счетчика предварительного делителя счетчика/календаря

### 17.3.9 RTC\_ALARM

Таблица 284 – Регистр RTC\_ALARM

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

Таблица 285 – Описание бит регистра RTC\_ALARM

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	RTC ALRM[31:0]	Значения для сравнения счетчика выхода из спящего режима и выработки сигнала ALRF

### 17.3.10 RTC\_CS

Таблица 286 – Регистр RTC\_CS

Номер	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
	TSF	ALRB_IE	ALRA_IE	ALRBF	ALRAF	ALRBE	ALRAE

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

Таблица 287 – Описание бит регистра RTC\_CS

Разряды	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
12	ALRB_IE	Разрешение прерывания по событию В: 0 – не разрешено; 1 – разрешено
11	ALRA_IE	Разрешение прерывания по событию А: 0 – не разрешено; 1 – разрешено
10	ALRBF	Флаг установки времени события В
9	ALRAF	Флаг установки времени события А:

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8	ALRBE	Разрешение события В. Совпадения RTC_ALRMBR и текущего календарного времени: 0 – не разрешено; 1 – разрешено
7	ALRAE	Разрешение события А. Совпадения RTC_ALRMAR и текущего календарного времени: 0 – не разрешено; 1 – разрешено
6	WEC	Запись завершена: 0 – можно записывать в один из регистров RTC_WUTR, RTC_PREDIV_S, RTC_DR, RTC_TR; 1 – идет запись в один из регистров RTC_WUTR, RTC_PREDIV_S, RTC_DR, RTC_TR, запись в регистры запрещена. Бит записи влияет не на блок регистров, а только на один. После записи каждого из регистров RTC_WUTR, RTC_PREDIV_S, RTC_DR, RTC_TR необходимо дождаться окончания записи
5	WUTF_IE	Флаг разрешения прерывания по совпадению счетчика выхода из спящего режима и регистра RTC_ALARM: 0 – нет совпадения; 1 – есть совпадение
4	SECF_IE	Флаг разрешения прерывания по разрешению счета основного секундного счетчика/календаря от счетчика предварительного деления: 0 – запрет прерывания один раз в секунду; 1 – разрешение прерывания один раз в секунду
3	OWF_IE	Флаг разрешения прерывания по переполнению счетчика выхода из спящего режима RTC_WUTR: 0 – запрет прерывания по переполнению; 1 – разрешение прерывания по переполнению
2	WUTF	Флаг совпадения счетчика выхода из спящего режима и регистра RTC_ALARM: 0 – нет совпадения; 1 – есть совпадение
1	SECF	Флаг разрешения счета основного счетчика/календаря от счетчика предварительного деления: 0 – нет разрешения счета; 1 – разрешение счета
0	OWF	Флаг переполнения счетчика выхода из спящего режима RTC_WUTR: 0 – нет переполнения; 1 – было переполнение

### 17.3.11 RTC\_TR

Таблица 288 – Регистр RTC\_TR

Номер	22	21, 20	19..16	14...12	11...8	6..4	3...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	00	0000	000	0000	000	0000
	PM	HT	HU	MNT	MNU	ST	SU

Таблица 289 – Описание бит регистра RTC\_TR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...23	-	Зарезервировано
22	PM	AM/PM значение: 0 – AM 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	-	Зарезервировано
14..12	MNT[2:0]	Десятки минут в BCD формате
11..8	MNU[3:0]	Единицы минут в BCD формате
7	-	Зарезервировано
6..4	ST[2:0]	Десятки секунд в BCD формате
3..0	SU[3:0]	Единицы секунд в BCD формате

### 17.3.12 RTC\_DR

Таблица 290 – Регистр RTC\_DR

Номер	23...20	19...16	15...13	12	11..8	5, 4	3...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0001	1001	010	0	0001	00	0001
	YT	YU	WDU	MT	MU	DT	DU

Таблица 291 – Описание бит регистра RTC\_DR

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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5, 4	DT[1:0]	Десятки даты в BCD формате
3...0	DU[3:0]	Единицы даты в BCD формате

### 17.3.13 RTC\_ALRMAR, RTC\_ALRMBR

Таблица 292 – Регистры RTC\_ALRMAR, 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	DU	MSK3	PM	HT

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	00	000	0000	0	000	0000
	HU	MSK2	MNT	MNU	MSK1	ST	SU

Таблица 293 – Описание бит регистров RTC\_ALRMAR, RTC\_ALRMBR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	MSK4	Маска даты: 0 – даты учитываются; 1 – даты не учитываются
30	WDSEL	Выбор дня недели: 0 – используется дата; 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 RTC\_TSTR1-RTC\_TSTR3, RTC\_TSDR1-RTC\_TSDR3

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

### 17.3.15 RTC\_TAFCR

Таблица 294 – Регистр RTC\_TAFCR

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

Номер	25	24	23	22	21..19	18..16
Доступ	R/W	R/W	R/W	R/W	R/C	R/C
Сброс	0	0	0	0	000	000
	TAMPE_W3	TAMPE_W2	TAMPE_W1	WAKE_SLEEP	TSINSEL[2:0]	TAMPINSEL[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
	-	TSIE	TSEEDGE[2:0]	TSE	TAMPIE	TAMPTRG[2:0]	TAMPE

Таблица 295 – Описание бит регистра RTC\_TAFCR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..29	EVNT_LEVEL_W	Отключение фиксации уровня на входах WAKEUP1-WAKEUP3: 0 – фиксация уровня длительностью более двух тактов частоты синхронизации часов реального времени включена; 1 – фиксация уровня отключена. Фронт импульса фиксируется в зависимости от бит EVNT_FRONT_W
28..26	EVNT_FRONT_W	Отключение фиксации фронта на входах WAKEUP1-WAKEUP3: 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 EVNT_LEVEL_W
25	TAMPE_W3	Разрешение обнаружения события TAMPER на выводе WAKEUP3 при TAMPE = 0: 0 – запрещено; 1 – разрешено
24	TAMPE_W2	Разрешение обнаружения события TAMPER на выводе WAKEUP2 при TAMPE = 0: 0 – запрещено;

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		1 – разрешено
23	TAMPE_W1	Разрешение обнаружения события TAMPER на выводе WAKEUP1 при TAMPE = 0: 0 – запрещено; 1 – разрешено
22	WAKE_SLEEP	Бит выбора включения/выключения ключа SW по событию RTC_WUTR (WUTF = 1): 0 – осуществляется размыкание ключа SW при срабатывании сторожевого таймера RTC_WUTR (WUTF = 1); 1 – осуществляется замыкание ключа SW при срабатывании сторожевого таймера RTC_WUTR (WUTF = 1)
21..19	TSINSEL[2:0]	Входы, детектированные схемой TIMESTAMP. Каждый разряд отвечает за срабатывания соответствующего входа WAKEUP1 – WAKEUP3. Если разряд установлен – вход сработал, если сброшен – вход не срабатывал. Возможна установка нескольких разрядов одновременно при срабатывании нескольких входов. Сбрасывается записью единицы в соответствующий бит
18..16	TAMPINSEL[2:0]	Входы, детектированные схемой TAMPER. Каждый разряд отвечает за срабатывания соответствующего входа WAKEUP1 – WAKEUP3. Если разряд установлен – вход сработал, если сброшен – вход не срабатывал. Возможна установка нескольких разрядов одновременно при срабатывании нескольких входов. Сбрасывается записью единицы в соответствующий бит
15..10	-	Зарезервировано
9	TSIE	Разрешение прерывания по внешнему событию TIMESTAMP: 0 – запрещено; 1 – разрешено
8..6	TSEEDGE[2:0]	Установка фронта срабатывания для внешнего события TIMESTAMP: 0 – передний фронт; 1 – задний фронт
5	TSE	Разрешение сохранения времени по внешнему событию TIMESTAMP: 0 – запрещено; 1 – разрешено После изменения состояния бита, задержка включения/выключения TIMESTAMP составляет два такта частоты синхронизации часов реального времени
4	TAMPIE	Разрешение прерывания по событию TAMPER: 0 – запрещено; 1 – разрешено После изменения состояния бита, задержка включения/выключения TAMPER составляет два такта частоты синхронизации часов реального времени
3..1	TAMPTRG[2:0]	Активный уровень события TAMPER: 0 - передний фронт или высокий уровень; 1 - задний фронт или низкий уровень
0	TAMPE	Разрешение обнаружения события TAMPER: 0 – запрещено;

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		1 – разрешено

### 17.3.16 RTC\_TMPICAL1

Таблица 296 – Регистр RTC\_TMPICAL1

Номер	31...30	29...25	24...20	19...15	14...10	9...5	4...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	00	01000	01100	10000	10100	10110	11010
	const6	const5	const4	const3	const2	const1	const0

Таблица 297 – Описание бит регистра RTC\_TMPICAL1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...30	const6	Младшие два разряда калибровочной константы для кода 6 с термодатчика
29...25	const5	Калибровочная константа для кода 5 с термодатчика
24...20	const4	Калибровочная константа для кода 4 с термодатчика
19...15	const3	Калибровочная константа для кода 3 с термодатчика
14...10	const2	Калибровочная константа для кода 2 с термодатчика
9...5	const1	Калибровочная константа для кода 1 с термодатчика
4..0	const0	Калибровочная константа для кода 0 с термодатчика

### 17.3.17 RTC\_TMPICAL2

Таблица 298 – Регистр RTC\_TMPICAL2

Номер	31...28	27...23	22...18	17...13	12...8	7...3	2...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	1110	01100	01000	00100	00000	00000	001
	const12	const11	const10	const9	const8	const7	const6

Таблица 299 – Описание бит регистра RTC\_TMPICAL2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	const12	Младшие четыре разряда калибровочной константы для кода 12 с термодатчика
27...23	const11	Калибровочная константа для кода 11 с термодатчика
22...18	const10	Калибровочная константа для кода 10 с термодатчика
17...13	const9	Калибровочная константа для кода 9 с термодатчика
12...8	const8	Калибровочная константа для кода 8 с термодатчика
7...3	const7	Калибровочная константа для кода 7 с термодатчика
2..0	const6	Старшие три разряда калибровочной константы для кода 6 с термодатчика

### 17.3.18 RTC\_TMPICAL3

Таблица 300 – Регистр RTC\_TMPICAL3

Номер	31...29	28	27...24	23...19
Доступ	U	R	R	R/W
Сброс	-	0	0000	10110
	-	result_ready	result_temp	const15

Номер	18...14	13...9	8	7...5	4	3	7...5
Доступ	R/W	R/W	R/W	U	R/W	R/W	U
Сброс	10100	10000	0	-	0	0	-
	const14	const13	const12		en_temp	sw_cntrl_inv	

Таблица 301 – Описание бит регистра RTC\_TMPICAL3

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28	result_ready	Флаг валидности данных в битах result_temp: 1 – данные валидны; 0 – данные не валидны
27...24	result_temp	Значение кода с термодатчика
23...19	const15	Калибровочная константа для кода 15 с термодатчика
18...14	const14	Калибровочная константа для кода 14 с термодатчика
13...9	const13	Калибровочная константа для кода 13 с термодатчика
8	const12	Старший разряд калибровочной константы для кода 12 с термодатчика
7...5	-	Зарезервировано
4	en_temp	Разрешение работы схемы термодатчика: 1 – разрешено; 0 – запрещено
3	lseconf_b3	lseconf[3]. Всегда записывать ноль
2	bldo_trim_b2	bldo_trim[2]. Всегда записывать ноль
1	-	Зарезервировано
0	swctr_inv	Инверсия логики управления выходом SW (open drain): 0 – при срабатывании переходит из HZ в 0; 1 – при срабатывании переходит из 0 в HZ

## 18 Порты ввода-вывода

Микроконтроллер 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 302 – Порты ввода-вывода

Вывод	Аналоговая функция ANALOG_EN=0	Цифровая функция						
		Порт IO MODE=00 ANALOG_EN=1	Основная MODE=01 ANALOG_EN=1	Альтернативная MODE=10 ANALOG_EN=1	Переопределённая MODE=11 ANALOG_EN=1			
<b>Порт А</b>								
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	
<b>Порт В</b>								
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	-	PB4	nUART1DTR		-		TMR3_CH3	
PB5	-	PB5	nUART1RTS		UART7816_RTS	12	TMR3_CH3N	
PB6	-	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	
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		-	

Вывод	Аналоговая функция ANALOG_EN=0	Цифровая функция							
		Порт IO MODE=00 ANALOG_EN=1	Основная MODE=01 ANALOG_EN=1	Альтернативная MODE=10 ANALOG_EN=1	Переопределённая MODE=11 ANALOG_EN=1				
Порт C									
PC0	-		PC0/MODE0	TMR4_ETR	19	-		TMR4_CH4	18
PC1	-	3	PC1/MODE1	TMR4_BRK		-		TMR4_CH4N	
PC2	-		PC2	TMR2_CH1	7	SSP3_FSS	14	I2C1_SCL	15
PC3	-		PC3	TMR2_CH1N		SSP3_CLK		I2C1_SDA	
PC4	-	PC4	EXT_INT2		SSP3_RXD	-			
PC5	-		PC5	EXT_INT3		SSP3_TXD		-	
PC6	-		PC6	TMR2_ETR	8	I2C1_SCL	15	-	
PC7	-		PC7	TMR2_BRK		I2C1_SDA		-	
Порт D									
PD0	-		PD0	SSP2_FSS	9	SD_V0_EXT			
PD1	-		PD1	SSP2_CLK		SD_V1_EXT			
PD2	-		PD2	SSP2_RXD		SD_V2_EXT		-	
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		SD_I3_EXT		-
PD11	-		PD11	UART7816_RXD		SD_I2_EXT		-	
PD12	-		PD12	UART7816_CLK		SD_I1_EXT		-	
PD13	-		PD13	UART7816_CTS		SD_I0_EXT		-	
PD14	-		PD14	UART7816_RTS	12	CLK_SD_EXT		-	
PD15	-		PD15	EXT_INT1		RTCCLK_1Hz		-	
<p><b>Примечания</b></p> <p>1 – Таймер 1;                  2, 5, 7, 8, 13 – Таймер 2;                  3 – Последовательный интерфейс SSP1;                  4 – Последовательный интерфейс UART1;                  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</p>									

**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 А в своей программе при применении на нём функции JTAG.

### 18.1 Описание регистров портов ввода-вывода

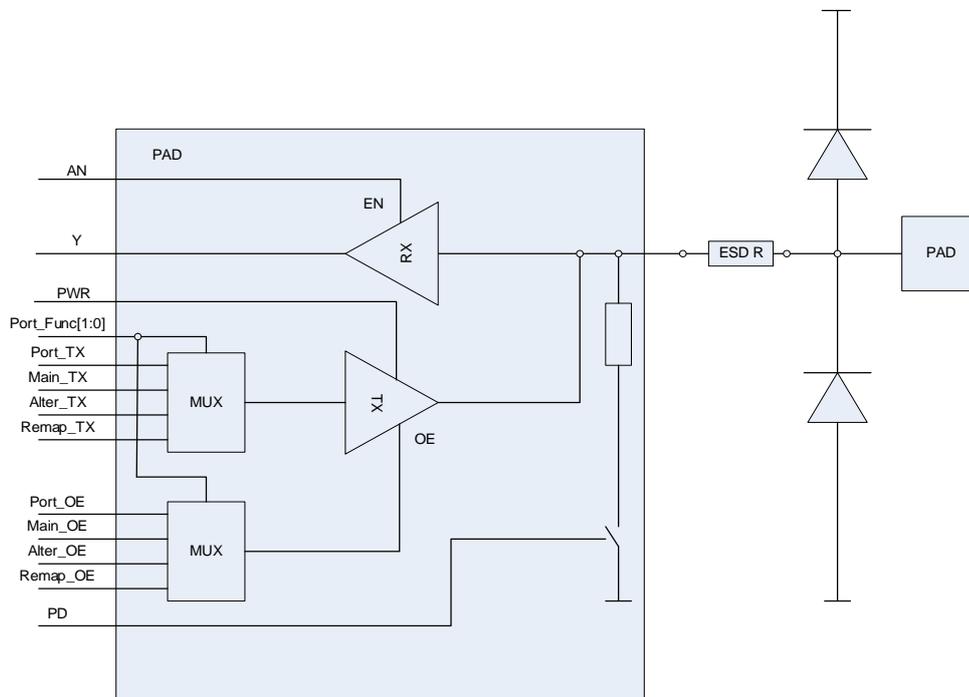


Рисунок 31 – Блок схема разряда порта ввода-вывода

Таблица 303 – Описание регистров портов ввода-вывода

Базовый Адрес	Название	Описание
0x5008_0000	GPIO1	Порт А
0x5008_8000	GPIO2	Порт В
0x5009_0000	GPIO3	Порт С
0x500F_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]	Настройка мощности передатчика
0x20	PORT_SETTX[15:0]	Регистр SET_TX записью 1 устанавливает 1 в регистре RXTX
0x24	PORT_CLR_TX[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 304 – Регистр PORTx\_RXTX

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

Таблица 305 – 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 306 – Регистр PORTx\_OE

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

Таблица 307 – 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 308 – 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 309 – 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 310 – Регистр PORTx\_ANALOG

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

Таблица 311 – 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 312 – 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 313 – 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 Разрешение подтяжки вверх: 1 – подтяжка до «1» включена (есть подтяжка); 0 – подтяжка до «1» выключена
15...0	PULL DOWN[15:0]	Режим работы вывода порта Разрешение подтяжки вниз: 1 – подтяжка до «0» включена (есть подтяжка); 0 – подтяжка до «0» выключена

### 18.1.6 PORT<sub>x</sub>\_PWR

Таблица 314 – Регистр PORT<sub>x</sub>\_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 315 – Описание бит регистра PORT<sub>x</sub>\_PWR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	PWR <sub>x</sub>	Аналогично PWR <sub>0</sub> для остальных бит порта
1	-	Зарезервировано
0	PWR <sub>0</sub>	1 – номинальная мощность площадка (нагрузка до 4 мА) 0 – уменьшенная мощность площадка (нагрузка до 2 мА)

### 18.1.7 PORT<sub>x</sub>\_SETTX

Таблица 316 – Регистр PORT<sub>x</sub>\_SETTX

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

Таблица 317 – Описание бит регистра PORT<sub>x</sub>\_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 PORT <sub>x</sub> _RXTX. Запись нуля не влияет на состояние соответствующего выхода порта PORT <sub>x</sub> _RXTX. Читается ранее записанным в регистр значениями, а не состояние соответствующих входов порта PORT <sub>x</sub> _RXTX

### 18.1.8 PORTx\_CLRTX

Таблица 318 – Регистр PORTx\_CLRTX

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

Таблица 319 – 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 «1» в соответствующий разряд регистра сбрасывает в «0» 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 питания трёх напряжений питания при работе микроконтроллера. 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 320 – Параметры формирования флагов PVD и PVDB

Наименование параметра, единица измерения	Норма параметра		
	не менее	типовое	не более
Напряжение питания, В	2,3	-	3,6
Напряжение питания батарейного домена, В	1,8	-	3,6
Уровень срабатывания PVD, В при PLS = “000”		2,3	
Уровень срабатывания PVDB, В при PLSB = “000”		1,8	
Уровень срабатывания PVD, В при PLS = “001”		2,4	
Уровень срабатывания PVDB, В при PLSB = “001”		2,0	
Уровень срабатывания PVD, В при PLS = “010”		2,5	
Уровень срабатывания PVDB, В при PLSB = “010”		2,2	
Уровень срабатывания PVD, В при PLS = “011”		2,6	
Уровень срабатывания PVDB, В при PLSB = “011”		2,4	
Уровень срабатывания PVD, В при PLS = “100”		2,7	
Уровень срабатывания PVDB, В при PLSB = “100”		2,6	
Уровень срабатывания PVD, В при PLS = “101”		2,8	
Уровень срабатывания PVDB, В при PLSB = “101”		2,8	
Уровень срабатывания PVD, В при PLS = “110”		2,9	
Уровень срабатывания PVDB, В при PLSB = “110”		3,0	

Наименование параметра, единица измерения	Норма параметра		
	не менее	типовое	не более
Уровень срабатывания PVD, В при PLS = "111"		3,0	
Уровень срабатывания PVDB, В при PLSB = "111"		3,2	

## 19.1 Описание регистров блока PVD

Таблица 321 – Описание регистров блока PVD

Базовый Адрес	Название	Описание
0x5005_8000	POWER	Датчик подсистемы питания
Смещение		
0x00	PVDCS [26:0]	Регистр управления и состояния датчика питания

### 19.1.1 PVDCS

Таблица 322 – Регистр PVDCS

Номер	26	25	24	23	22	21	20	19
Доступ	R/W	U	R/W	R/W	U	R/W	R/W	U
Сброс	0	0	0	0	0	0	1	0
	INV	-	INVB	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 323 – Описание бит регистра PVDCS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	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
25	-	Зарезервировано
24	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
23	-	Всегда записывать «0»
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 «0»
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. <i>Рекомендуется включать блок при PLS = 000, затем выставлять нужный уровень, затем инверсию, затем разрешать прерывания</i>
1	-	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
0	PVDBEN	<p>Бит разрешения работы блока датчика напряжения батарейного питания:</p> <p>0 – датчик отключен;</p> <p>1 – датчик включен</p> <p><i>Рекомендуется включать блок при PLSB = 000, затем выставлять нужный уровень, затем инверсию, затем разрешать прерывания</i></p>

## 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 32.

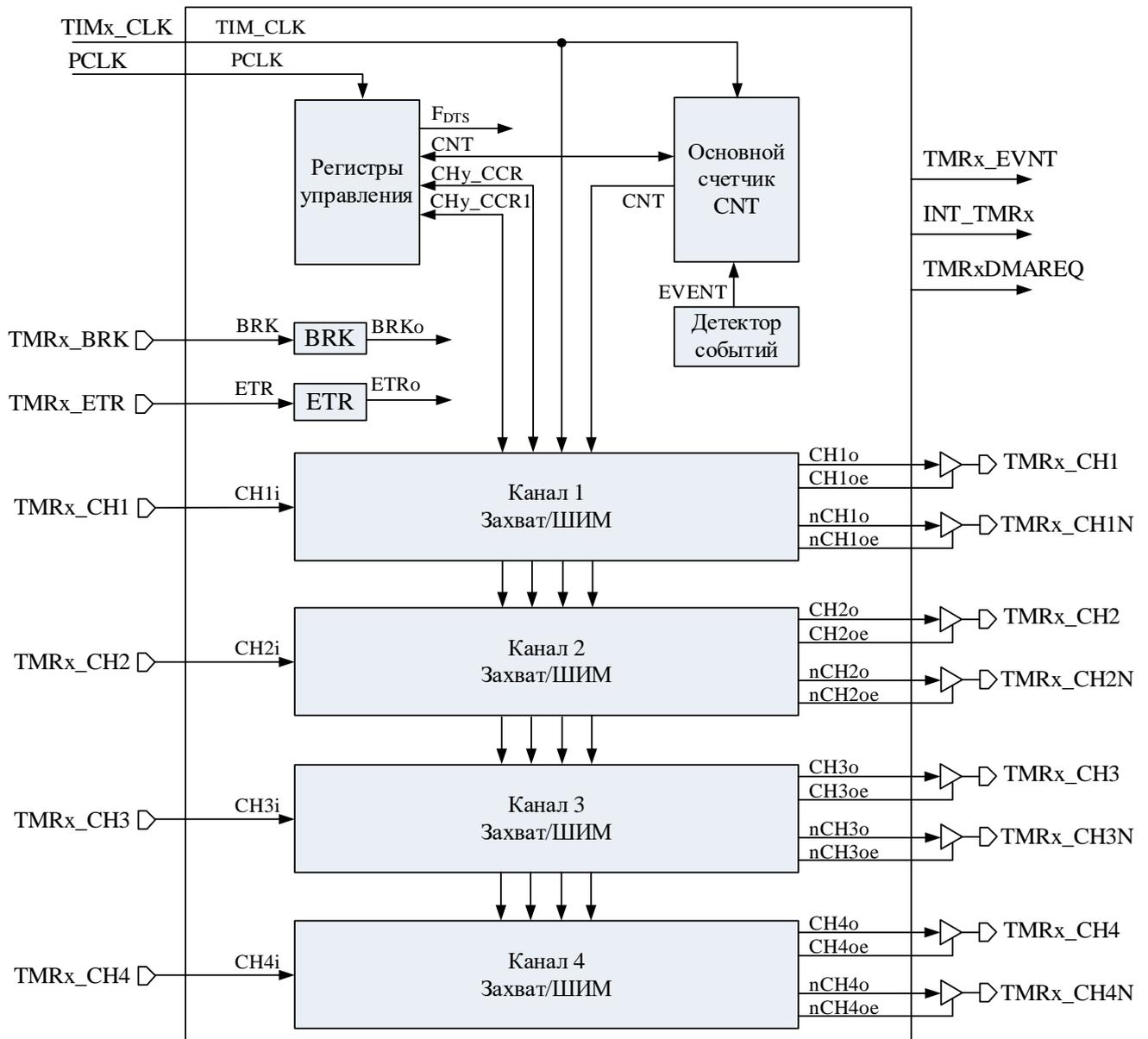


Рисунок 32 – Структурная схема блока «Таймер»

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 «Источники 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.

Частота тактирования таймера TIMx\_CLK должна быть менее или равна частоте тактирования ядра 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 вступают 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
```

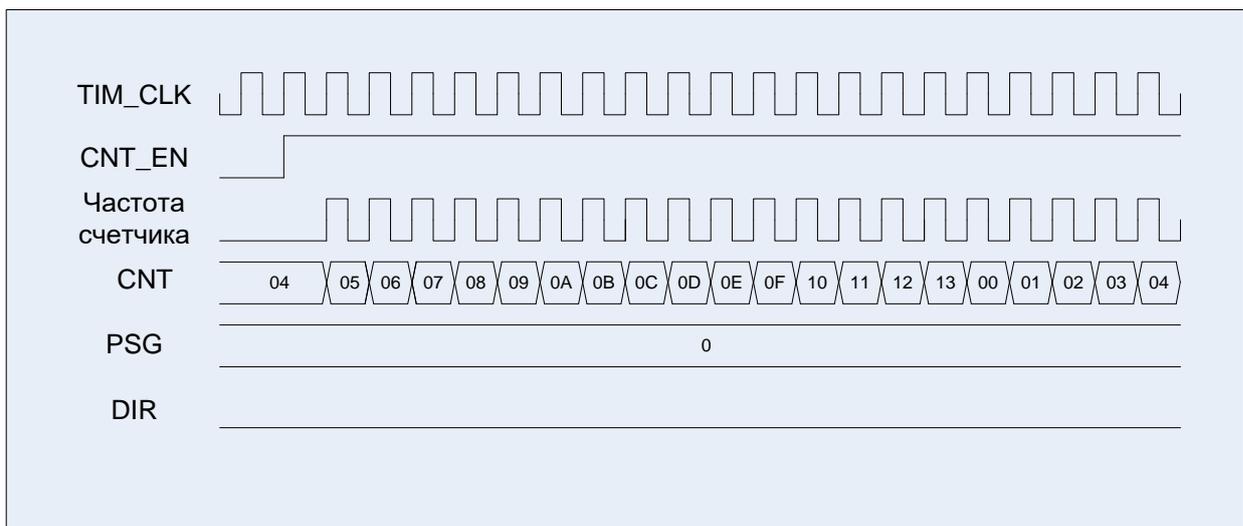


Рисунок 33 – Диаграммы работы таймера, счет прямой от 0 до 0x13, стартовое значение 0x04

**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

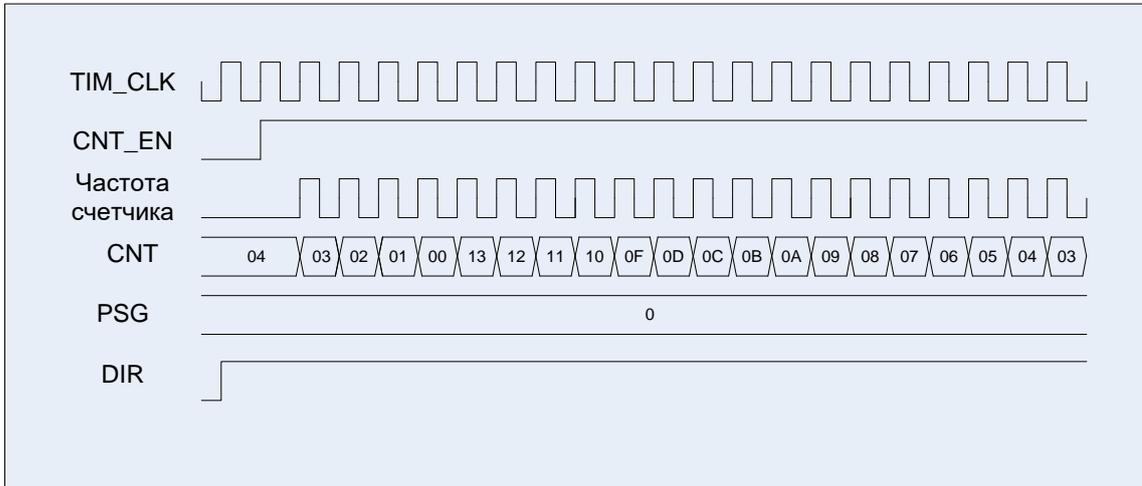


Рисунок 34 – Диаграммы работы таймера, счет обратный от 0x13 до 0, стартовое значение 0x04

**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 35 – 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

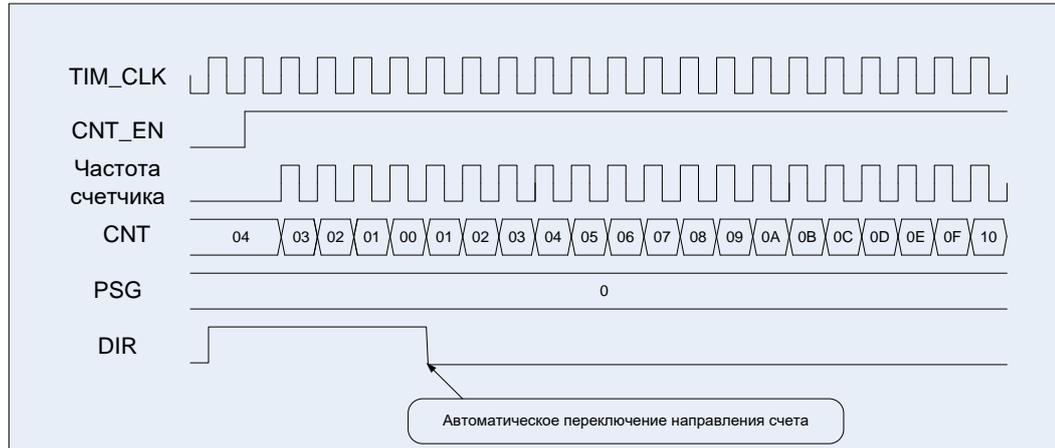


Рисунок 36 – Диаграммы работы таймера, счет двунаправленный, сначала обратный

**20.2.4 Тактовая частота F<sub>DTS</sub>**

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

Тактовая частота F<sub>DTS</sub> формируется из частоты TIM\_CLK путём прореживания на заданный коэффициент (1, 2, 3 или 4). Настройка частоты F<sub>DTS</sub> осуществляется в регистре CNTRL, поле FDTS[1:0].

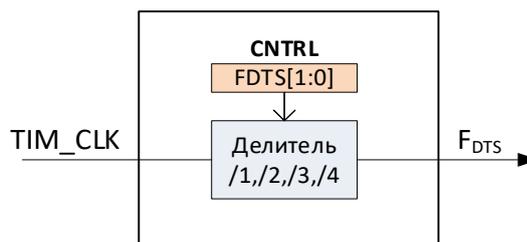


Рисунок 37 – Схема формирования тактовой частоты F<sub>DTS</sub>

Диаграмма возможных частот F<sub>DTS</sub> приведена на рисунке 38.

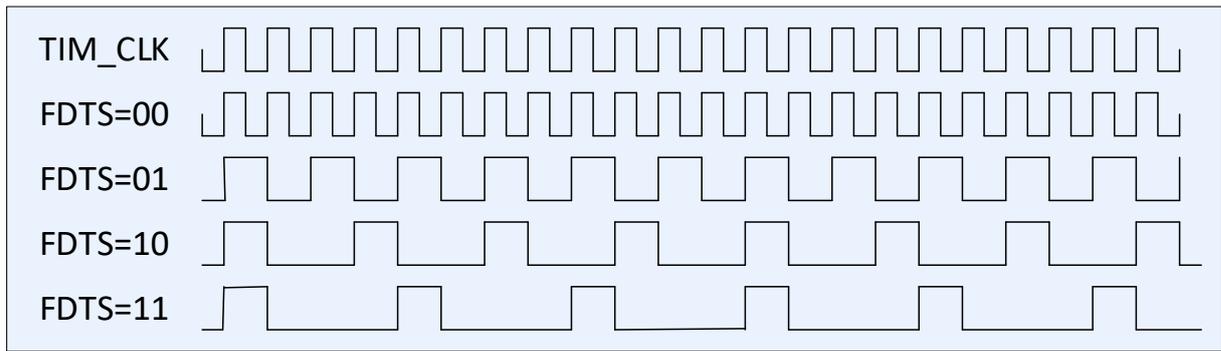


Рисунок 38 – Диаграмма тактовой частоты  $F_{DTS}$  в зависимости от значения  $F_{DTS}[1:0]$  в регистре CNTRL

### 20.3 Источники 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 CH<sub>U</sub>i;
- 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.

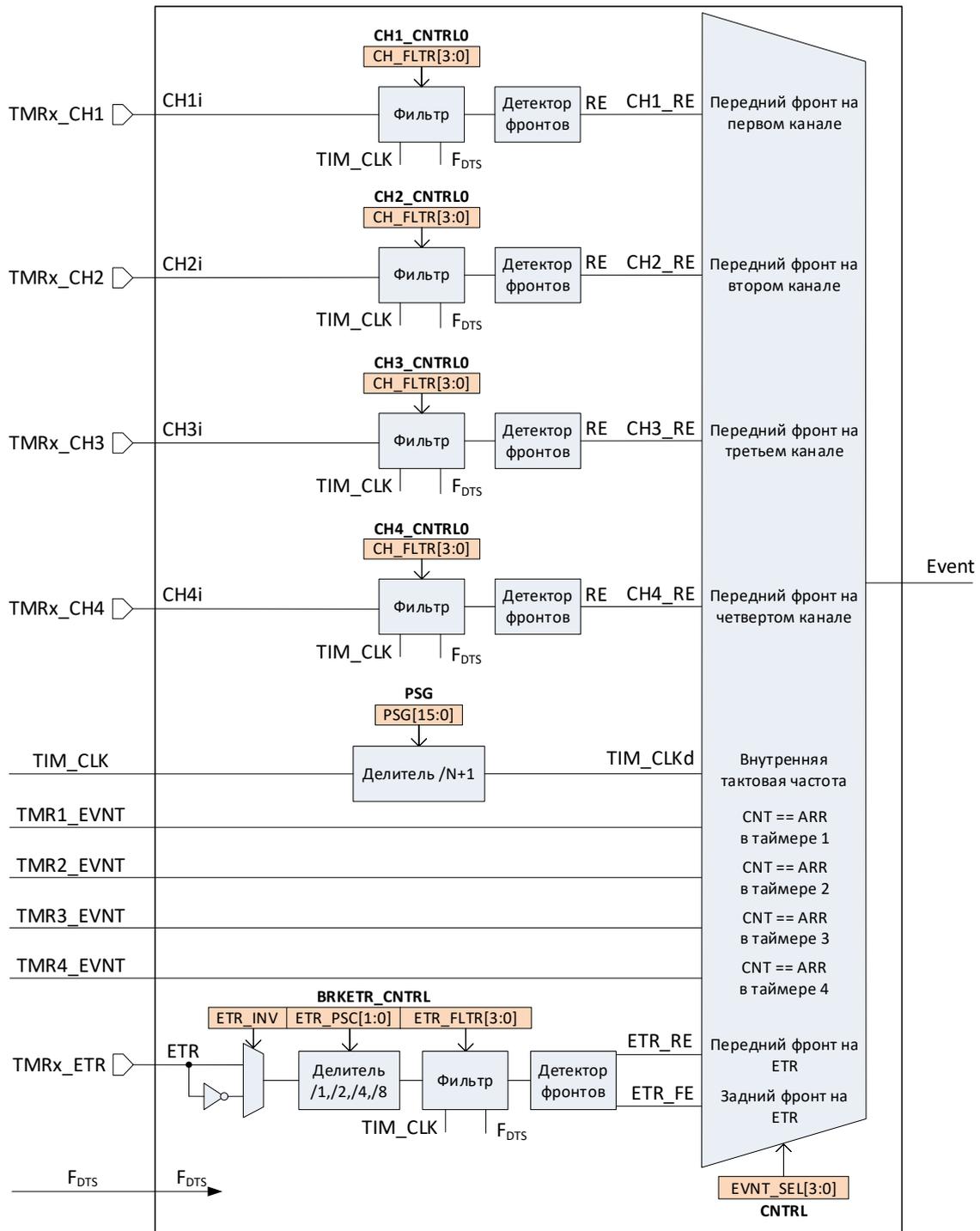


Рисунок 39 – Структурная схема формирования события для счета

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

Данный режим выбирается, когда  $EVENT\_SEL[3:0] = 0000$  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) не равно нулю, то счетный регистр делителя будет инкрементироваться по каждому импульсу сигнала TIM\_CLK до тех пор, пока не достигнет значения, находящегося в регистре

делителя. Далее счетный регистр делителя сбрасывается в ноль, содержимое основного счетчика таймера 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 40, 41 приведены диаграммы работы счетчика при обновлении значения PSG.

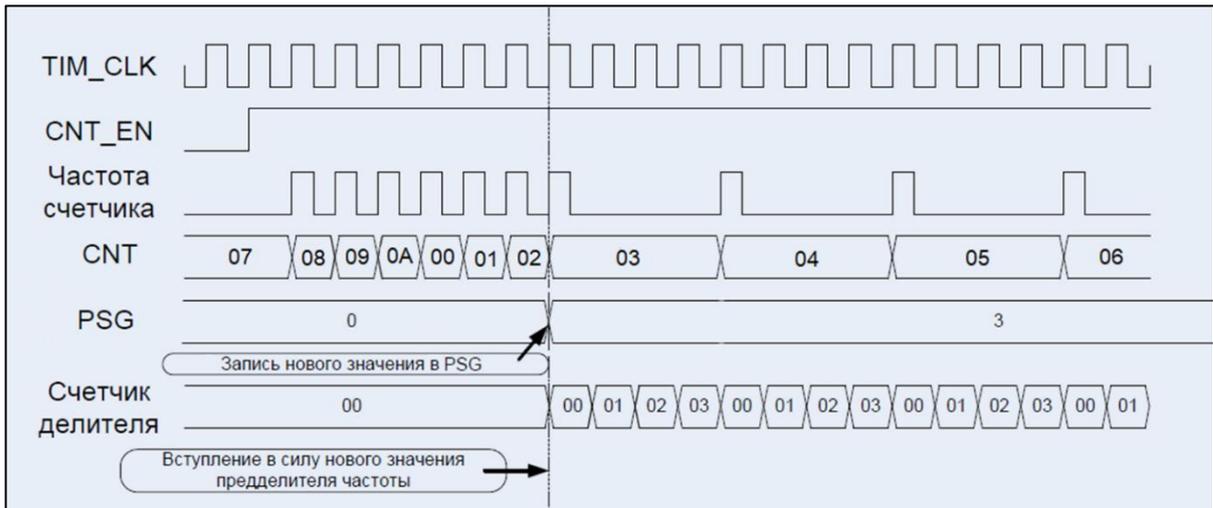


Рисунок 40 – Диаграмма работы счетчика: счет прямой (CNT\_MODE[1:0] = 00, EVENT\_SEL[3:0] = 0000, DIR = 0)

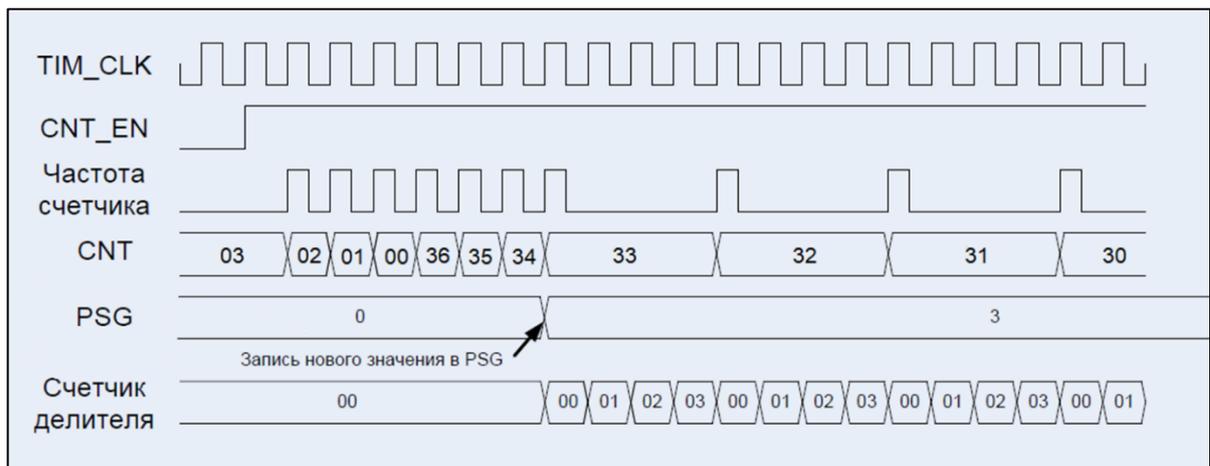


Рисунок 41 – Диаграмма работы счетчика: счет обратный (CNT\_MODE[1:0] = 00, EVENT\_SEL[3:0] = 0000, DIR = 1)

### 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 42 показан пример каскадного соединения таймеров, диаграммы работы данных таймеров приведены на рисунке 43.

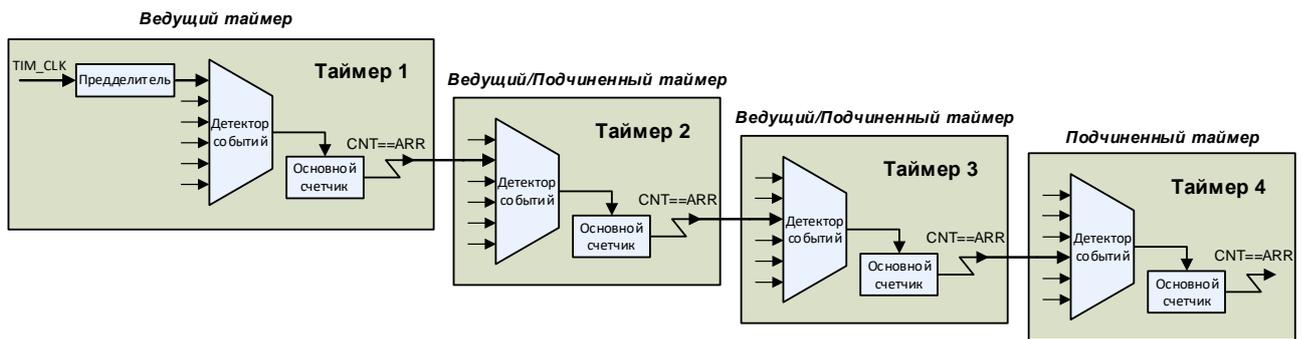


Рисунок 42 – Пример каскадного соединения таймеров

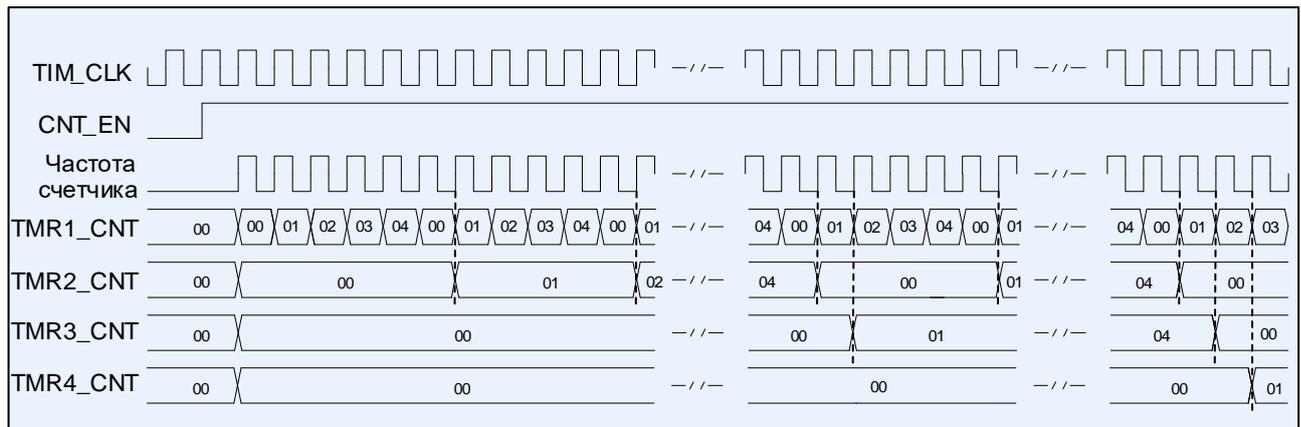


Рисунок 43 – Диаграммы работы четырех таймеров в каскаде  
 TIMER1: DIR = 0, EVNT\_SEL[3:0] = 0000, CNT\_MODE[1:0] = 00;  
 TIMER2: DIR = 0, EVNT\_SEL[3:0] = 0001, CNT\_MODE[1:0] = 10;  
 TIMER3: DIR = 0, EVNT\_SEL[3:0] = 0010, CNT\_MODE[1:0] = 10;  
 TIMER4: DIR = 0, EVNT\_SEL[3:0] = 0011, CNT\_MODE[1:0] = 10.

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

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 CHy*i*. Биты CH\_SEL[1:0] регистра CHy\_CNTRL0 не оказывают влияния, так как они применяются для работы канала таймера только в режиме захвата.

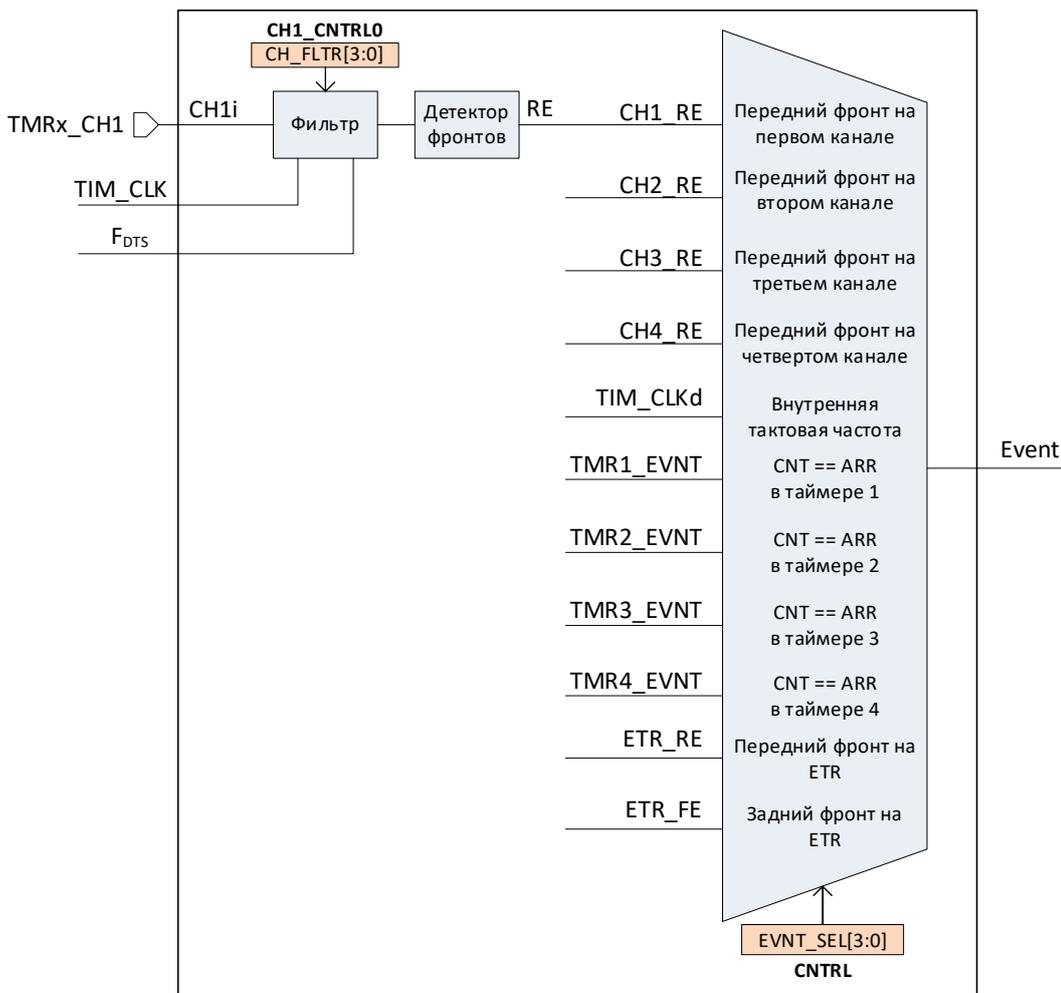


Рисунок 44 – Схема тактирования сигналом со входа первого канала

Со входа CH<sub>у</sub>1 внешний тактовый сигнал поступает в блок цифрового фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. «Блок цифрового фильтра»). Настройки фильтра задаются в поле CH\_FLTR[3:0] регистра CH<sub>у</sub>\_CNTRL0.

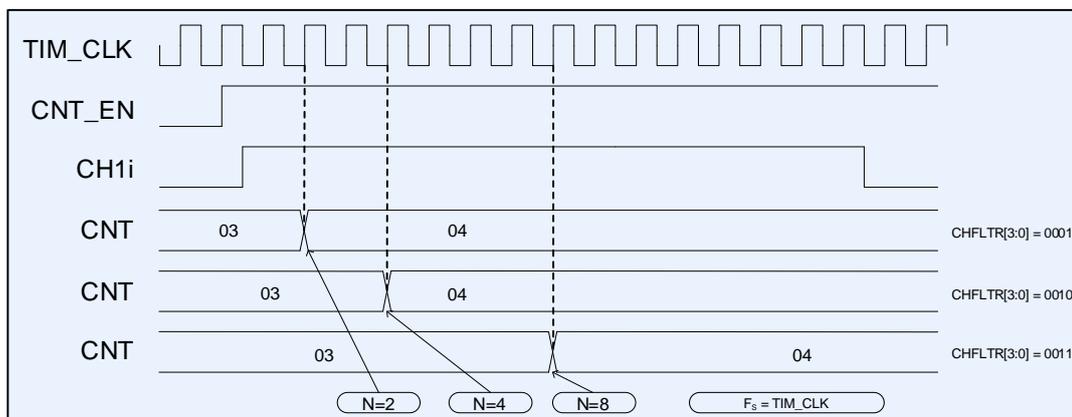


Рисунок 45 – Диаграмма внешнего тактирования с разными вариантами фильтра

### 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 предделителя внешней частоты (1, 2, 4 или 8). После предделителя тактовый сигнал поступает в блок цифрового фильтра, где он может быть дополнительно отфильтрован с целью устранения импульсов, длительность которых меньше заданного порога (см. «Блок цифрового фильтра»). Настройки фильтра задаются в поле  $ETR\_FLTR[3:0]$ .

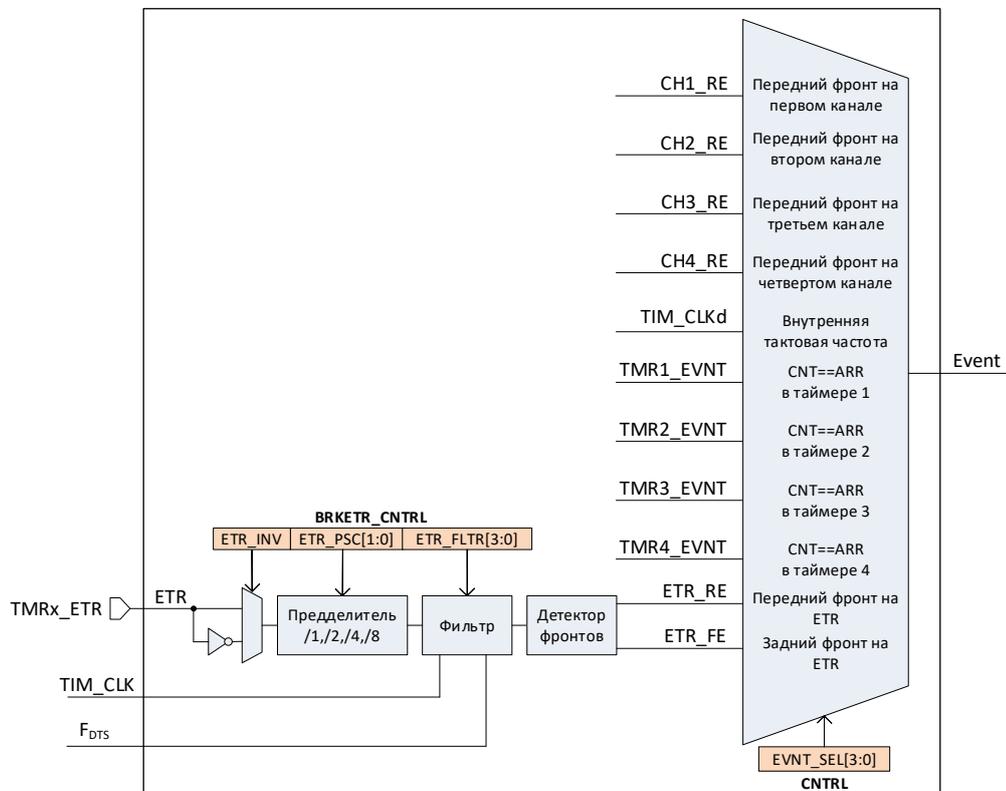


Рисунок 46 – Схема тактирования сигналом со входа ETR

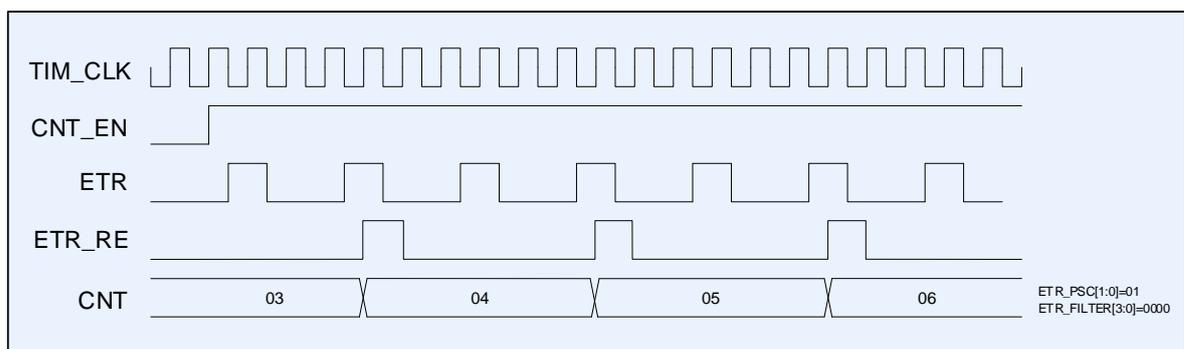


Рисунок 47 – Диаграмма тактирования со входа ETR,  $EVNT\_SEL[3:0] = 1000$

## 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 CH<sub>y</sub>\_CCR и CH<sub>y</sub>\_CCR1. Регистрация событий осуществляется только по входам CH<sub>y</sub><sub>i</sub> (выводы, настроенные в функцию TMR<sub>x</sub>\_CH<sub>y</sub>).

Структурная схема блока захвата представлена на рисунке 48.

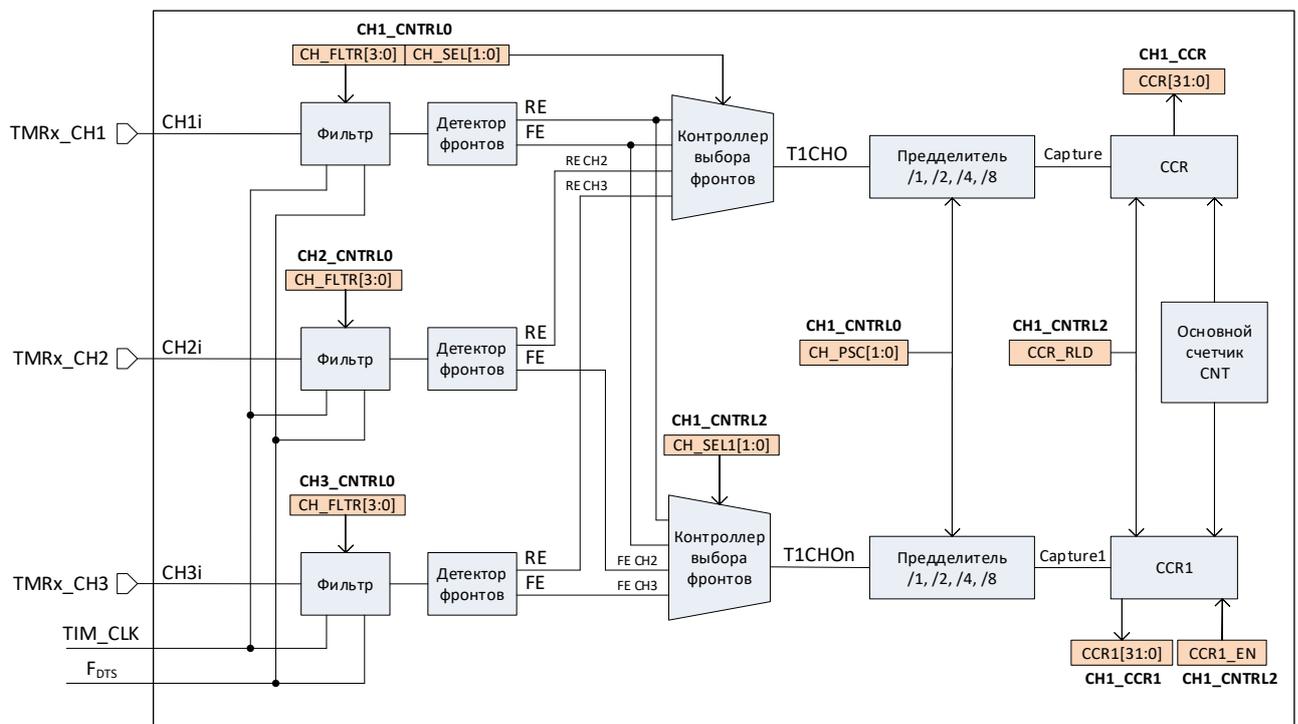


Рисунок 48 – Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо записать «1» в бит CAP\_NPWM регистра управления каналом CH<sub>y</sub>\_CNTRL0. Для использования регистра CH<sub>y</sub>\_CCR1 необходимо записать «1» в бит CCR1\_EN регистра CH<sub>y</sub>\_CNTRL2.

Внешний сигнал со входа CH<sub>y</sub><sub>i</sub> сначала поступает в блок фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. «Блок цифрового фильтра»). Настройки фильтра задаются в поле CH\_FLTR[3:0] регистра CH<sub>y</sub>\_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 CH<sub>y</sub>\_CCR осуществляется в поле CH\_SEL[1:0] регистра CH<sub>y</sub>\_CNTRL0, а для регистра CH<sub>y</sub>\_CCR1 – в поле CH\_SEL1[1:0] регистра CH<sub>y</sub>\_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 X18 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.

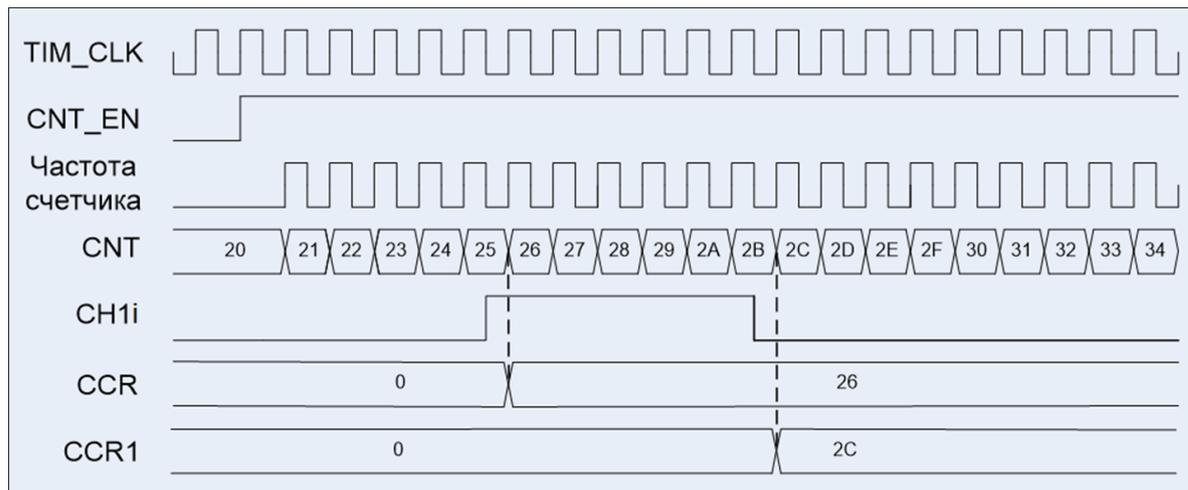


Рисунок 49 – Диаграмма захвата события со входа канала 1

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 50.

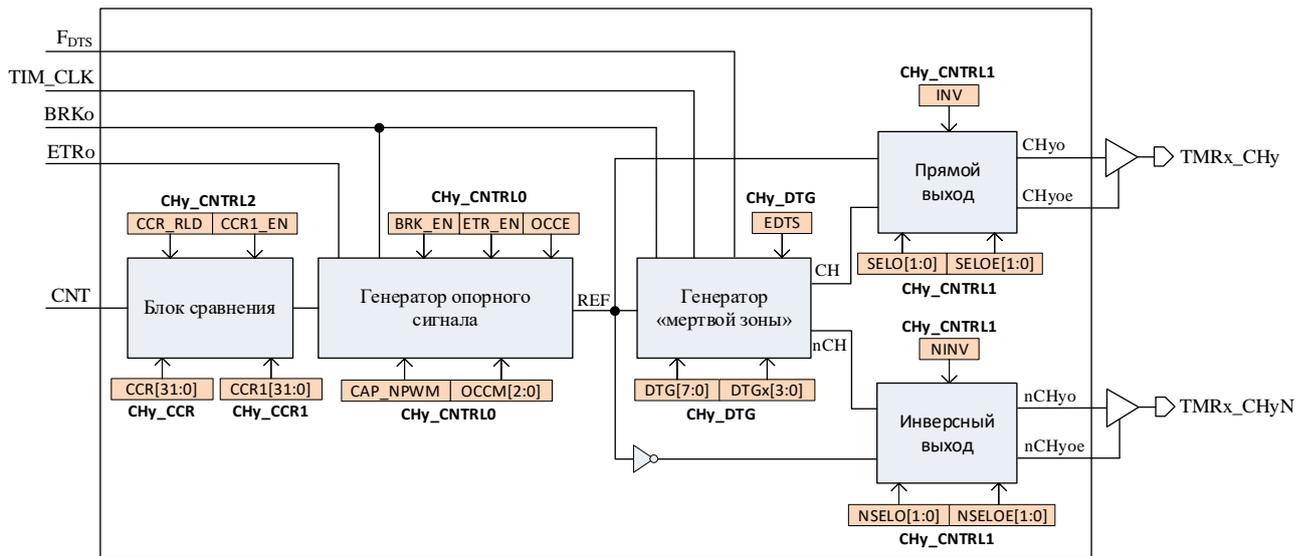


Рисунок 50 – Структурная схема блока формирования ШИМ

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 регистре управления каналом таймера CHy\_CNTRL0, поле OCCM[2:0].

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.

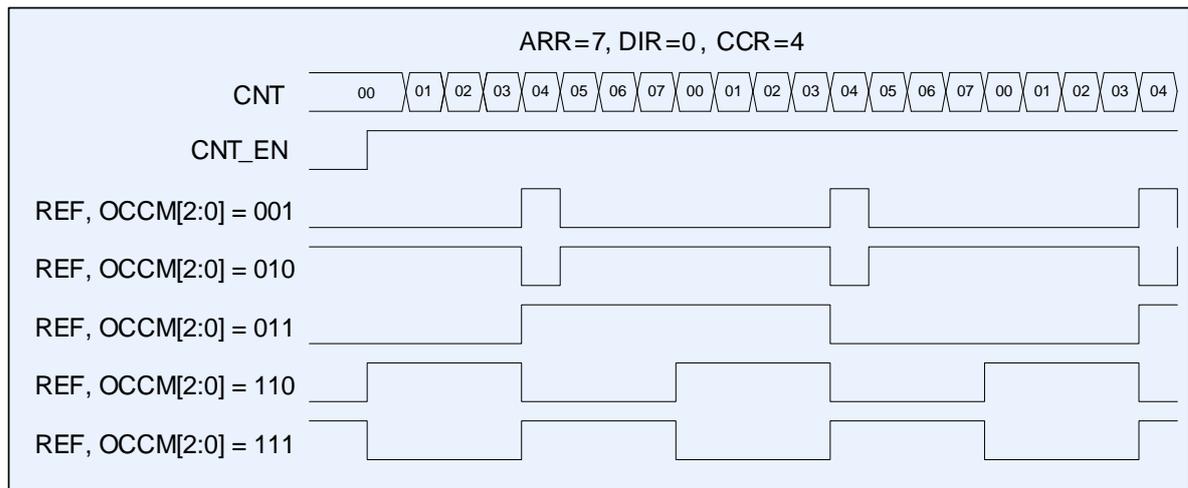


Рисунок 51 – Диаграмма работы в режиме ШИМ, CCR1\_EN=0

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.

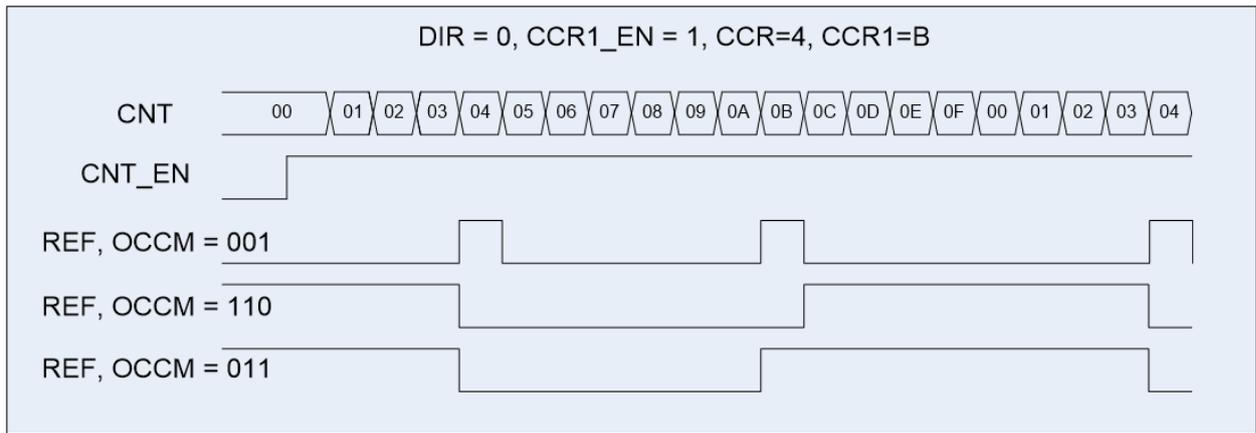


Рисунок 52 – Диаграмма работы в режиме ШИМ, CCR1\_EN = 1

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 изменён 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» (см. рисунок 53).

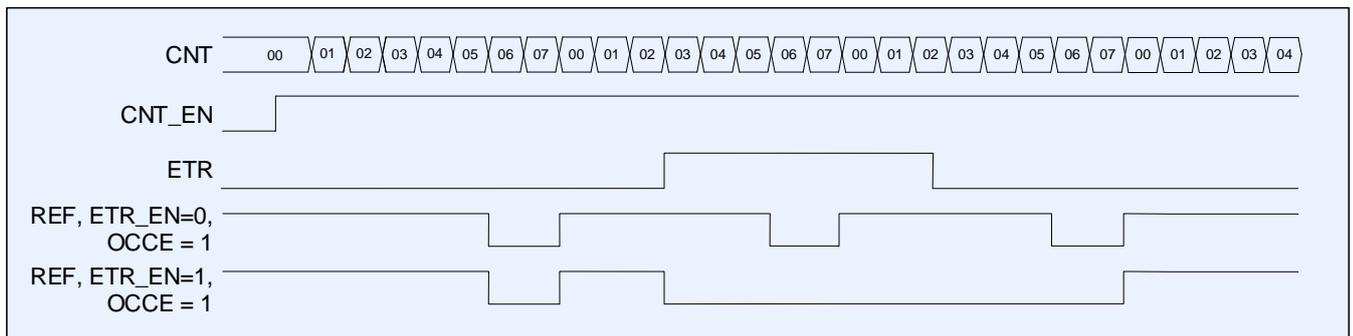


Рисунок 53 – Диаграмма сброса сигнала REF по выводу ETR

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 (см. рисунок 54).

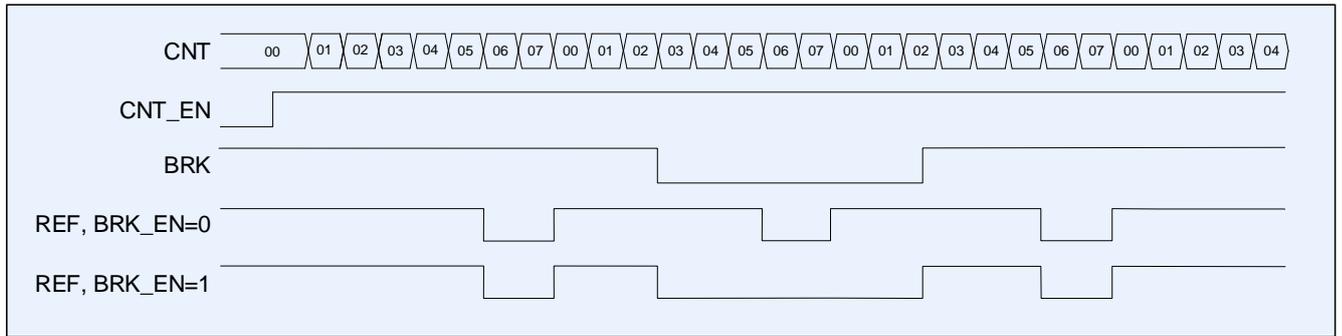


Рисунок 54 – Диаграмма сброса сигнала REF по выводу BRK

### 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 (CHyo, CHyoe) представляет собой инвертированный сигнал REF, в котором передний фронт задержан на величину DTGdel относительно заднего фронта опорного сигнала REF;

- сигнал на инверсном выходе (nCHyo, nCHyoe) 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  $F_{DTS}$  по формуле

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

где DTGx – предварительный делитель частоты;  
DTG – основной делитель частоты.

Управление блоком DTG осуществляется через регистр CHy\_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.

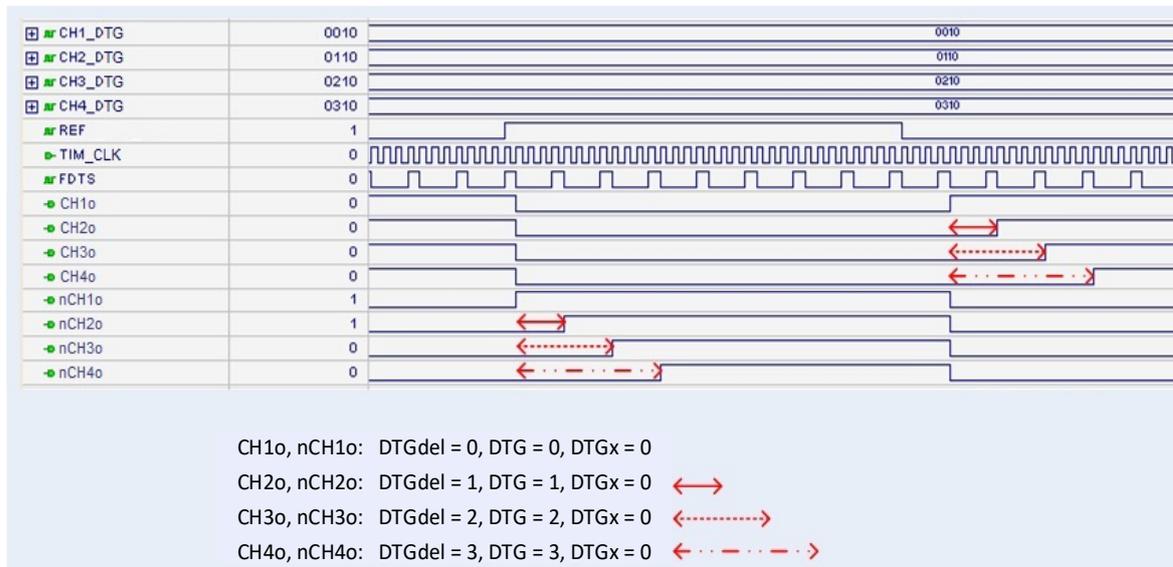


Рисунок 55 – Диаграмма работы блока DTG

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 изменён 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 CHy\_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 (CHyo, nCHyo), так и сигнал разрешения выдачи (CHyoe, nCHyoe). В качестве сигналов для прямого (CHyo, CHyoe) и инверсного (nCHyo, nCHyoe) выходов в регистре CHy\_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 (CHyo) и инверсного (nCHyo) выходов задается в полях SELO[1:0] и NSELO[1:0] регистра CHy\_CNTRL1. Дополнительно каждый сигнал выдачи для прямого (CHyo) и инверсного (nCHyo) выходов может быть инвертирован путем установки битов INV и NINV в регистре CHy\_CNTRL1. Выбор источника сигнала разрешения выдачи для прямого (CHyoe) и инверсного (nCHyoe) 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 CHy1 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 CHy1 – 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 передаётся 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 56, частота  $F_S = \text{TIM\_CLK}$ , количество выборок  $N = 4$ .

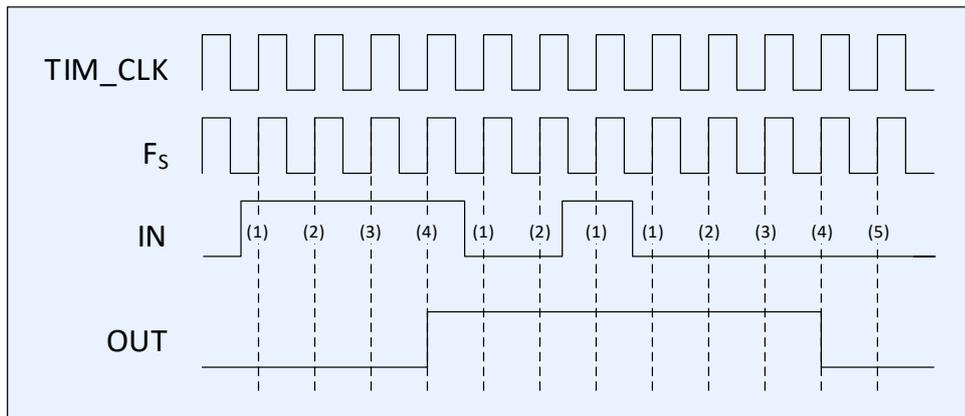


Рисунок 56 – Диаграмма работы фильтра,  $F_S = \text{TIM\_CLK}$ ,  $N = 4$

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

Диаграмма работы фильтра при использовании частоты  $F_{DTS}$  приведена на рисунке 57, частота выборки  $F_S = F_{DTS}/2$ , количество выборок  $N = 6$ .

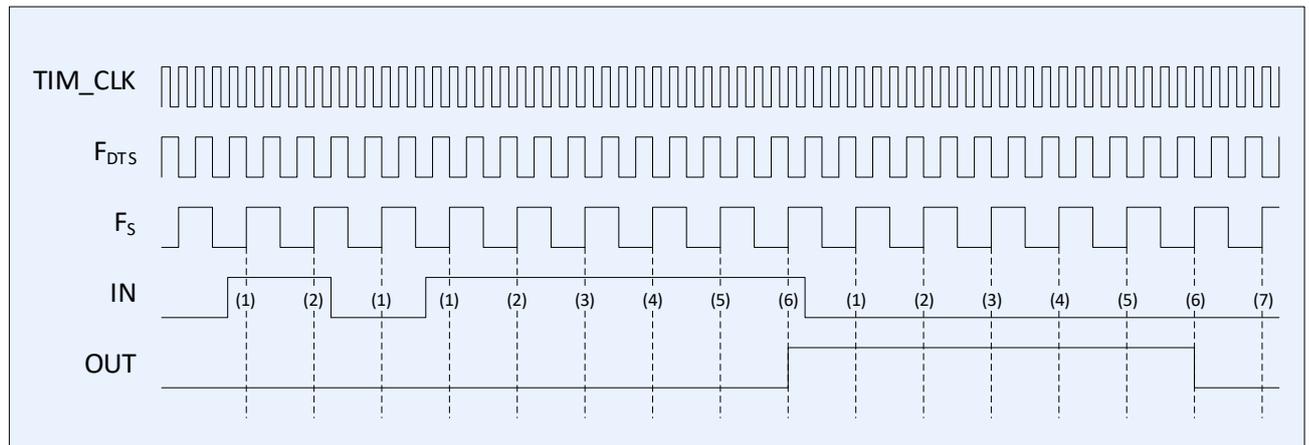


Рисунок 57 – Диаграмма работы фильтра,  $F_S = F_{DTS}/2$ ,  $N = 6$

Возможные варианты настройки блока фильтра приведены в таблице 324.

Таблица 324 – 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\_TMR<sub>x</sub>. 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 TMR<sub>x</sub>DMAREQ. 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
```

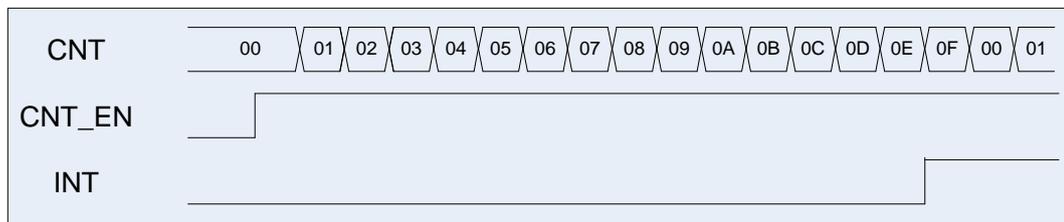


Рисунок 58 – Режим обычного счетчика

### 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 2 выборкам на частоте TIM_CLK
TIMER1->CH4_CNTRL0 = 0x00008011; //Захват по отрицательному фронту сигнала
//на входе CH4i, фильтрация выполняется
//по 2 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
```

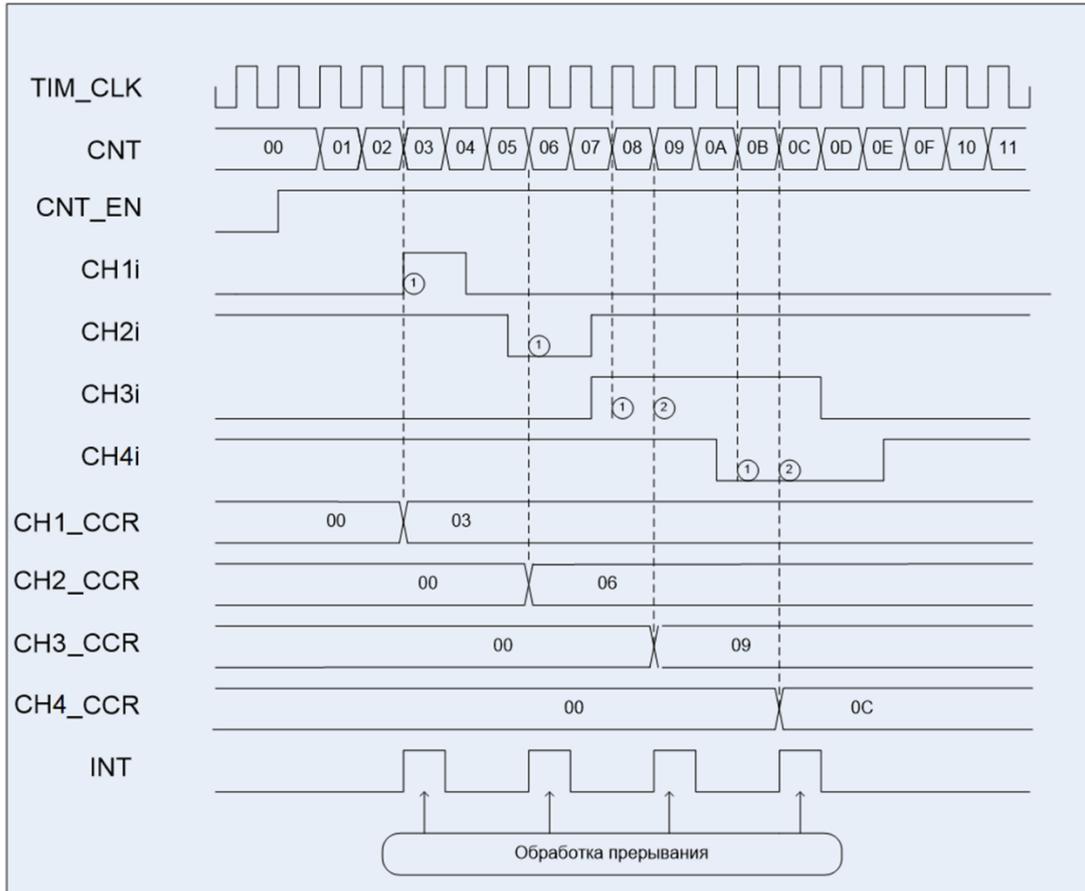


Рисунок 59 – Диаграммы примера работы в режиме захвата

### 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
    
```

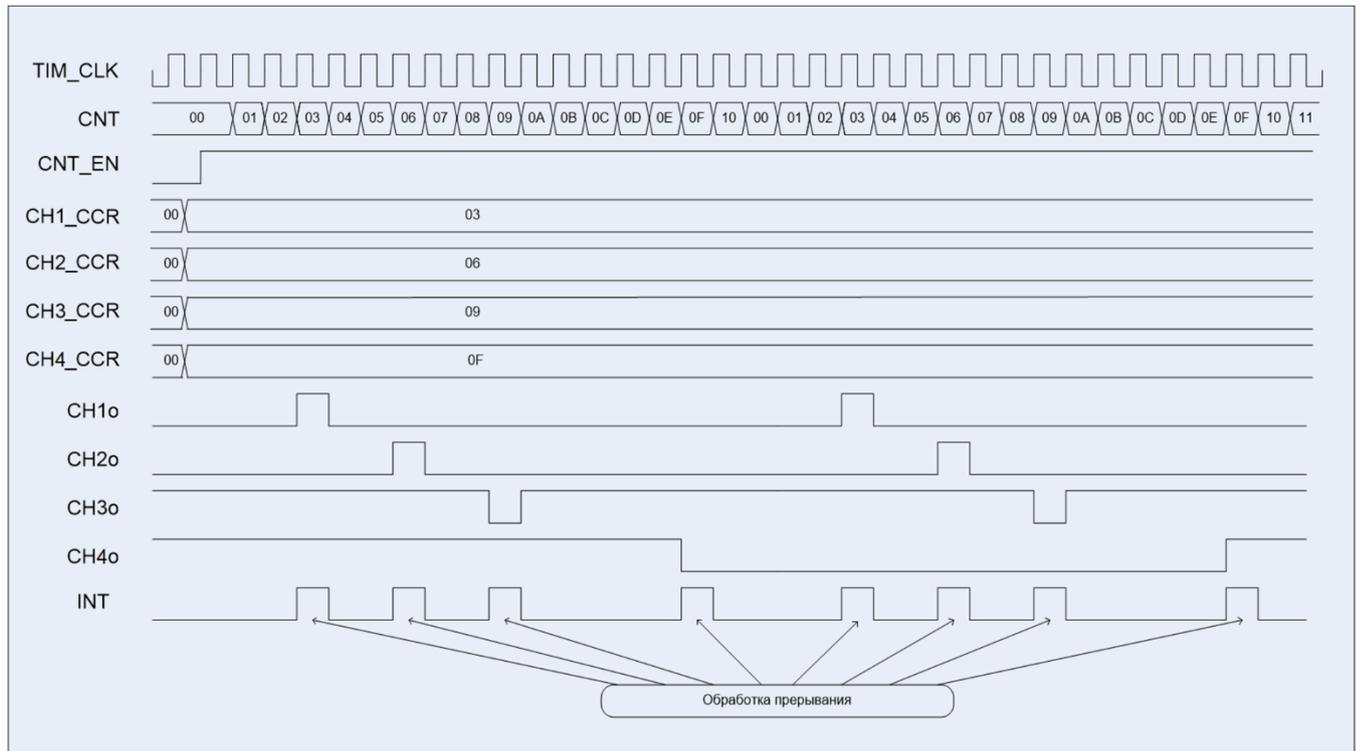


Рисунок 60 – Диаграммы примера работы в режиме ШИМ

## 20.9 Описание регистров блока таймера

Таблица 325 – Базовые адреса и смещения регистров управления таймера

Адрес	Название	Описание
0x5007_0000	TIMER1	Контроллер TIMER1
0x5007_8000	TIMER2	Контроллер TIMER2
0x500D_8000	TIMER3	Контроллер TIMER3
0x500E_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
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_CNTRL0[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_CNTRL0[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 326 – Основной счетчик таймера CNT

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

Таблица 327 – 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 328 – Делитель частоты TIM\_CLK для счета основного счетчика PSG

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

Таблица 329 – 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 330 – Основание счета основного счетчика ARR

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

Таблица 331 – 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 332 – 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...9	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 333 – 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; 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 СНу\_CCR

Таблица 334 – Регистр сравнения/захвата для 'у' канала таймера СНу\_CCR

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

Таблица 335 – Описание бит регистра СНу\_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 СНу\_CCR1

Таблица 336 – Регистр сравнения/захвата для 'у' канала таймера СНу\_CCR1

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

Таблица 337 – Описание бит регистра СНу\_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 338 – 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 339 – 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
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	OCCM[2:0]	Формат выработки сигнала REF в режиме ШИМ:  Если CCR1_EN=0: 000 – всегда 0; 001 – 1, если CNT==CCR; 010 – 0, если CNT==CCR; 011 – переключение REF, если CNT==CCR; 100 – всегда 0; 101 – всегда 1; 110 – 1, если DIR=0 (счет 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.

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		<p>Если CCR1_EN=1:</p> <p>000 – всегда 0;</p> <p>001 – 1, если CNT==CCR или CNT==CCR1;</p> <p>010 – 0, если CNT==CCR или CNT==CCR1;</p> <p>011 – переключение REF, если CNT==CCR или CNT==CCR1;</p> <p>100 – всегда 0;</p> <p>101 – всегда 1;</p> <p>110 – 0, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 1; 0, если DIR=1 (счет обратный), CCR&lt;CNT&lt;CCR1, иначе 1;</p> <p>111 – 1, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 0; 1, если DIR=1 (счет обратный), CCR&lt;CNT&lt;CCR1, иначе 0;</p> <p>Необходимо соблюдать условие CCR&lt;CCR1</p>
8	OCCE	<p>Разрешение работы ETR:</p> <p>0 – запрещен;</p> <p>1 – разрешен</p>
7...6	CH_PSC[1:0]	<p>Предварительный делитель входного канала:</p> <p>00 – нет деления;</p> <p>01 – /2;</p> <p>10 – /4;</p> <p>11 – /8</p>
5...4	CH_SEL[1:0]	<p>Выбор события по входному каналу CH<sub>у</sub> для фиксации значения основного счетчика (регистр CNT) в регистр CH<sub>у</sub>_CCR:</p> <p>00 – положительный фронт на входном канале CH<sub>у</sub>;</p> <p>01 – отрицательный фронт на входном канале CH<sub>у</sub>;</p> <p>10 – положительный фронт от других каналов:</p> <ul style="list-style-type: none"> <li>– для первого канала от второго канала;</li> <li>– для второго канала от третьего канала;</li> <li>– для третьего канала от четвертого канала;</li> <li>– для четвертого канала от первого канала;</li> </ul> <p>11 – положительный фронт от других каналов:</p> <ul style="list-style-type: none"> <li>– для первого канала от третьего канала;</li> <li>– для второго канала от четвертого канала;</li> <li>– для третьего канала от первого канала;</li> <li>– для четвертого канала от второго канала</li> </ul>
3...0	CH_FLTR[3:0]	<p>Конфигурация фильтра на входе канала 'у'. Выбор частоты выборки F<sub>s</sub> и количества выборок N:</p> <p>0000 – нет фильтрации, F<sub>s</sub> = F<sub>DTS</sub>;</p> <p>0001 – F<sub>s</sub> = TIM_CLK, N = 2;</p> <p>0010 – F<sub>s</sub> = TIM_CLK, N = 4;</p> <p>0011 – F<sub>s</sub> = TIM_CLK, N = 8;</p> <p>0100 – F<sub>s</sub> = F<sub>DTS</sub>/2, N = 6;</p> <p>0101 – F<sub>s</sub> = F<sub>DTS</sub>/2, N = 8;</p> <p>0110 – F<sub>s</sub> = F<sub>DTS</sub>/4, N = 6;</p> <p>0111 – F<sub>s</sub> = F<sub>DTS</sub>/4, N = 8;</p> <p>1000 – F<sub>s</sub> = F<sub>DTS</sub>/8, N = 6;</p> <p>1001 – F<sub>s</sub> = F<sub>DTS</sub>/8, N = 8;</p> <p>1010 – F<sub>s</sub> = F<sub>DTS</sub>/16, N = 5;</p> <p>1011 – F<sub>s</sub> = F<sub>DTS</sub>/16, N = 6;</p> <p>1100 – F<sub>s</sub> = F<sub>DTS</sub>/16, N = 8;</p>

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		1101 – $F_S = F_{DTS}/32$ , $N = 5$ ; 1110 – $F_S = F_{DTS}/32$ , $N = 6$ ; 1111 – $F_S = F_{DTS}/32$ , $N = 8$

### 20.9.8 CHy\_CNTRL1

Таблица 340 – 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 341 – 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 nCHy0 выдается 0; 01 – на nCHy0 выдается 1; 10 – на nCHy0 выдается сигнал nREF; 11 – на nCHy0 выдается сигнал с DTG
9...8	NSELOE[1:0]	Режим работы инверсного выхода nCHy: 00 – на nCHy0е выдается 0; 01 – на nCHy0е выдается 1; 10 – на nCHy0е выдается сигнал nREF; 11 – на nCHy0е выдается сигнал с DTG. При nCHy0е = 0 вывод канала в третьем состоянии, при nCHy0е = 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 CHy0 выдается 0; 01 – на CHy0 выдается 1; 10 – на CHy0 выдается сигнал REF; 11 – на CHy0 выдается сигнал с DTG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1...0	SELOE[1:0]	Режим работы прямого выхода СНу: 00 – на СНуое выдается 0; 01 – на СНуое выдается 1; 10 – на СНуое выдается сигнал REF; 11 – на СНуое выдается сигнал с DTG. При СНуое = 0 вывод канала работает в режиме входа, при СНуое = 1 вывод канала работает в режиме выхода

### 20.9.9 СНу\_CNTRL2

Таблица 342 – Регистр управления 2 для ‘у’ канала таймера СНу\_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 343 – Описание бит регистра СНу\_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 СНу_CCR и СНу_CCR1: 0 – сигнал события захвата устанавливается в момент обнаружения события, при этом обновление регистров СНу_CCR и СНу_CCR1 выполняется через один такт TIM_CLK; 1 – сигнал события захвата устанавливается синхронно с обновлением информации в регистрах СНу_CCR и СНу_CCR1
3	CCR_RLD	Режим обновления регистров СНу_CCR и СНу_CCR1: 0 – обновление возможно в любой момент времени; 1 – обновление будет осуществлено только при CNT == 0
2	CCR1_EN	Разрешение работы регистра СНу_CCR1: 0 – СНу_CCR1 не используется; 1 – СНу_CCR1 используется
1...0	CH_SEL1[1:0]	Выбор события по входному каналу СНу <sub>и</sub> для фиксации значения основного счетчика (регистр CNT) в регистр СНу_CCR1: 00 – положительный фронт на входном канале СНу <sub>и</sub> ; 01 – отрицательный фронт на входном канале СНу <sub>и</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 344 – 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 345 – Описание бит регистра CHy\_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 $DTG_{del} = DTG \cdot (DTGx + 1)$
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 346 – 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 347 – 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;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		1000 – $F_S = F_{DTS}/8, N = 6$ ; 1001 – $F_S = F_{DTS}/8, N = 8$ ; 1010 – $F_S = F_{DTS}/16, N = 5$ ; 1011 – $F_S = F_{DTS}/16, N = 6$ ; 1100 – $F_S = F_{DTS}/16, N = 8$ ; 1101 – $F_S = F_{DTS}/32, N = 5$ ; 1110 – $F_S = F_{DTS}/32, N = 6$ ; 1111 – $F_S = F_{DTS}/32, N = 8$
3...2	ETR_PSC[1:0]	Асинхронный предделитель частоты со входа ETR: 00 – без деления; 01 – /2; 10 – /4; 11 – /8
1	ETR_INV	Инверсия входа ETR: 0 – без инверсии; 1 – инверсия
0	BRK_INV	Инверсия входа BRK: 0 – без инверсии; 1 – инверсия

### 20.9.12 STATUS

Таблица 348 – 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 349 – 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. 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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. 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 350 – 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 351 – 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
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 352 – 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 353 – 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 микроконтроллере 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 28 тактов синхронизации CLK. В качестве синхросигнала может выступать частота процессора CPU\_CLK 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.

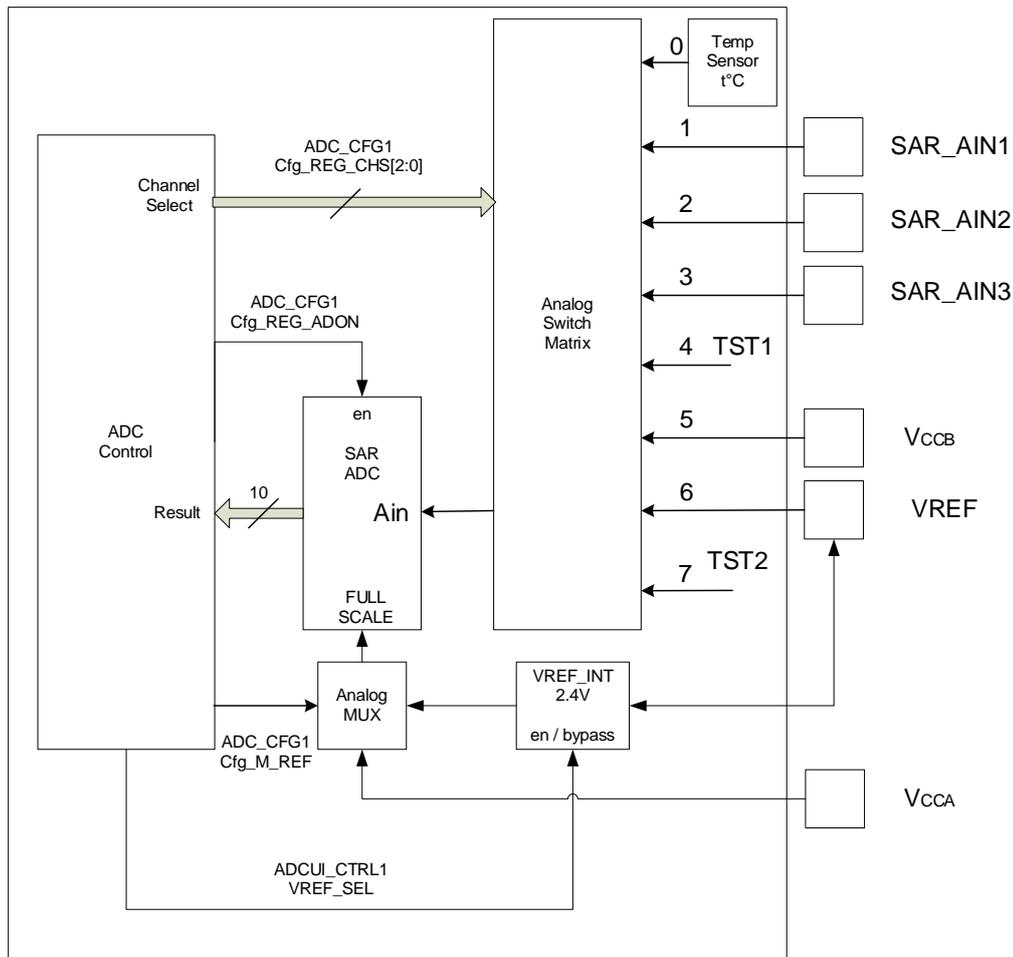


Рисунок 61 – Структурная схема АЦП

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. 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 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.

### 21.4.1 Формула расчета температуры

Для определения значения температуры кристалла в градусах Цельсия следует использовать формулу:

$$TEMP == \frac{25 - \frac{\text{напряжение полной шкалы АЦП} \cdot \text{значение регистра } ADC1\_RESULT}{1024} - 1,403}{0,0037} \quad (14)$$

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

$$TEMP = \frac{25 - \frac{2,5 \cdot 625}{1024} - 1,403}{0,0037} = -8,2 \text{ } ^\circ\text{C} \quad (15)$$

## 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

$$T_{track} = 4 \cdot T_{ACLK} + N_{PCLKd} \cdot T_{PCLKd} = \frac{4}{f_{ACLK}} + \frac{(ConvDelay)}{f_{PCLKd}}, \quad (17)$$

где  $ConvDelay$  – дополнительная задержка перед началом преобразования, определяется битами  $DelayGo[2:0]$  исходя из таблиц 354 – 356;  
 $f_{ACLK}$  – рабочая частота АЦП;  
 $f_{PCLKd}$  – определяется формулой

$$f_{PCLKd} = \frac{f_{PCLK}}{2^{CfgREGDIVCLK}}. \quad (18)$$

Если необходимо обеспечить преобразование с точностью 10 разрядов  $\pm 1$  LSB, то  $N = 10$ . Время заряда  $T_{track}$  определяется битами  $DelayGo[2:0]$  и схемой самого АЦП и представлено в таблицах 354 – 356. Время заряда  $T_{track}$  должно составлять не менее 1 мкс.

### 21.6 Время заряда внутренней емкости АЦП и время преобразования

Таблица 354 – Время заряда внутренней емкости АЦП и время преобразования.  
 $CfgREGCLKS = ACLK/PCLKd, ACLK=PCLKd=PCLK$

DelayGo[2:0]	Дополнительная задержка перед началом преобразования	Общее время $T_{track}$ заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	1xPCLKd	4xACLK+1xPCLKd	25xACLK+1xPCLKd
001	3xPCLKd	4xACLK+3xPCLKd	25xACLK+3xPCLKd
010	3xPCLKd	4xACLK+3xPCLKd	25xACLK+3xPCLKd
011	5xPCLKd	4xACLK+5xPCLKd	25xACLK+5xPCLKd
100	5xPCLKd	4xACLK+5xPCLKd	25xACLK+5xPCLKd
101	7xPCLKd	4xACLK+7xPCLKd	25xACLK+7xPCLKd
110	7xPCLKd	4xACLK+7xPCLKd	25xACLK+7xPCLKd
111	9xPCLKd	4xACLK+9xPCLKd	25xACLK+9xPCLKd

Таблица 355 – Время заряда внутренней емкости АЦП и время преобразования.  
CfgREGCLKS = PCLKd

DelayGo[2:0]	Дополнительная задержка перед началом преобразования	Общее время Ttrack заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	0xPCLKd	4xACLK+0xPCLKd	25.5xACLK+2xPCLK+0xPCLKd
001	0xPCLKd	4xACLK+0xPCLKd	25.5xACLK+2xPCLK+0xPCLKd
010	2xPCLKd	4xACLK+2xPCLKd	25.5xACLK+2xPCLK+2xPCLKd
011	2xPCLKd	4xACLK+2xPCLKd	25.5xACLK+2xPCLK+2xPCLKd
100	4xPCLKd	4xACLK+4xPCLKd	25.5xACLK+2xPCLK+4xPCLKd
101	4xPCLKd	4xACLK+4xPCLKd	25.5xACLK+2xPCLK+4xPCLKd
110	6xPCLKd	4xACLK+6xPCLKd	25.5xACLK+2xPCLK+6xPCLKd
111	6xPCLKd	4xACLK+6xPCLKd	25.5xACLK+2xPCLK+6xPCLKd

Таблица 356 – Время заряда внутренней емкости АЦП и время преобразования.  
CfgREGCLKS = ACLK

DelayGo[2:0]	Дополнительная задержка перед началом преобразования	Общее время Ttrack заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	1xPCLKd	4xACLK+1xPCLKd	25xACLK+1xPCLKd +1.5xPCLK
001	2xPCLKd	4xACLK+2xPCLKd	25xACLK+2xPCLKd+1.5xPCLK
010	3xPCLKd	4xACLK+3xPCLKd	25xACLK+3xPCLKd+1.5xPCLK
011	4xPCLKd	4xACLK+4xPCLKd	25xACLK+4xPCLKd+1.5xPCLK
100	5xPCLKd	4xACLK+5xPCLKd	25xACLK+5xPCLKd+1.5xPCLK
101	6xPCLKd	4xACLK+6xPCLKd	25xACLK+6xPCLKd+1.5xPCLK
110	7xPCLKd	4xACLK+7xPCLKd	25xACLK+7xPCLKd+1.5xPCLK
111	8xPCLKd	4xACLK+8xPCLKd	25xACLK+8xPCLKd+1.5xPCLK

Помимо точности определяемой временем зарядки внутренней емкости АЦП точность преобразования имеет ошибки, связанные с технологическими разбросами схемы и шумами и определяемые параметрами  $E_{DLADC}$ ,  $E_{ILADC}$  и  $E_{OFFADC}$ .

Для корректного 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 357 – Описание регистров блока контроллера АЦП

Базовый Адрес	Название	Описание
0x5004_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 358 – 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]	-	-

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 359 – 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 354 – 356
24...19	-	Должны быть в нуле для правильного функционирования

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
18..16	-	Должно быть 010 для правильного функционирования
15...12	Cfg REG DIVCLK[3:0]	Выбор коэффициента деления частоты процессора: 0000 – PCLKd = HCLK; 0001 – PCLKd = HCLK/2; 0010 – PCLKd = HCLK/4; 0011 – PCLKd = HCLK/8; ... 1011 – PCLKd = HCLK/2048. Остальные PCLKd = HCLK
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
9	Cfg REG CHCH	Выбор переключения каналов: 1 – переключение включено (перебираются каналы, выбранные в регистре выбора канала SI_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 CLK работы ADC: 1 – ACLK (формируется от выбранной частоты, подробнее см. раздел «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
0	Cfg REG ADON	Включение АЦП: 1 – включено; 0 – выключено

### 21.7.2 ADC1\_H\_LEVEL

Таблица 360 – Регистр ADC1\_H\_LEVEL

Номер	31...10	9...0
Доступ	U	R/W
Сброс	0	0
	-	REG H LEVEL [9:0]

Таблица 361 – 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 362 – Регистр ADC1\_L\_LEVEL

Номер	31...10	9...0
Доступ	U	R/W
Сброс	0	0
	-	REG L LEVEL [9:0]

Таблица 363 – 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 364 – 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 365 – 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 366 – 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 367 – 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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 368 – Регистр ADC1\_CHSEL

Номер	31...8	7...0
Доступ	U	R/W
Сброс	0	0
	-	SI_Ch_Ch_REF[7:0]

Таблица 369 – 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 370 – Регистр ADC\_TRIM

Номер	31..6	5...1	0
Доступ	U	R/W	U
Сброс	0	10000	0
	-	BG_TRIM[4:0]	-

Таблица 371 – 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 использовать значение бит TRIM_BG записанных во Flash-памяти при производстве
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 микроконтроллере 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

$$F_{scl} = \frac{HCLK}{5 \cdot (DIV + 1)} \cdot \quad (19)$$

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.

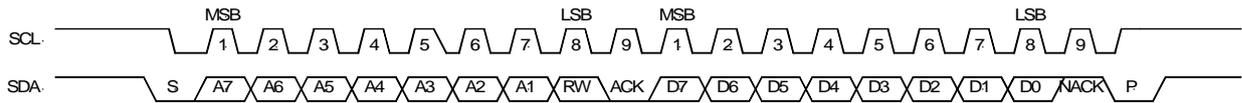


Рисунок 62 – Передача по I2C

### 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 7-битный 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, выставя ACK и опуская линию SDA в низкий уровень на 9-ый 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 372 – Описание регистров контроллера I2C

Базовый Адрес	Название	Описание
0x5003_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 373 – Регистр PRL

Номер	31...8	7... 0
Доступ	U	R/W
Сброс	0	0xFF
	-	PR[7:0]

Таблица 374 – 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 375 – Регистр PRH

Номер	31...8	7... 0
Доступ	U	R/W
Сброс	0	0xFF
	-	PR[15:8]

Таблица 376 – 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 377 – 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 378 – 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 379 – Регистр RXD

Номер	31...8	7... 0
Доступ	U	R/O
Сброс	0	0
	-	RXD[7:0]

Таблица 380 – 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 381 – 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 382 – 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 383 – Регистр TXD

Номер	31...8	7...0
Доступ	U	R/W
Сброс	0	0
	-	TXD[7:0]

Таблица 384 – 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 385 – 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 386 – 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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), 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).

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.

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 (8 ячеек по 16 бит) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);
- Программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;
- Программируемая длительность информационного кадра от 4 до 16 бит;
- Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по переполнению буфера приемника;
- Доступна возможность тестирования по шлейфу, соединяющему вход с выходом;
- Поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена на рисунке 63.

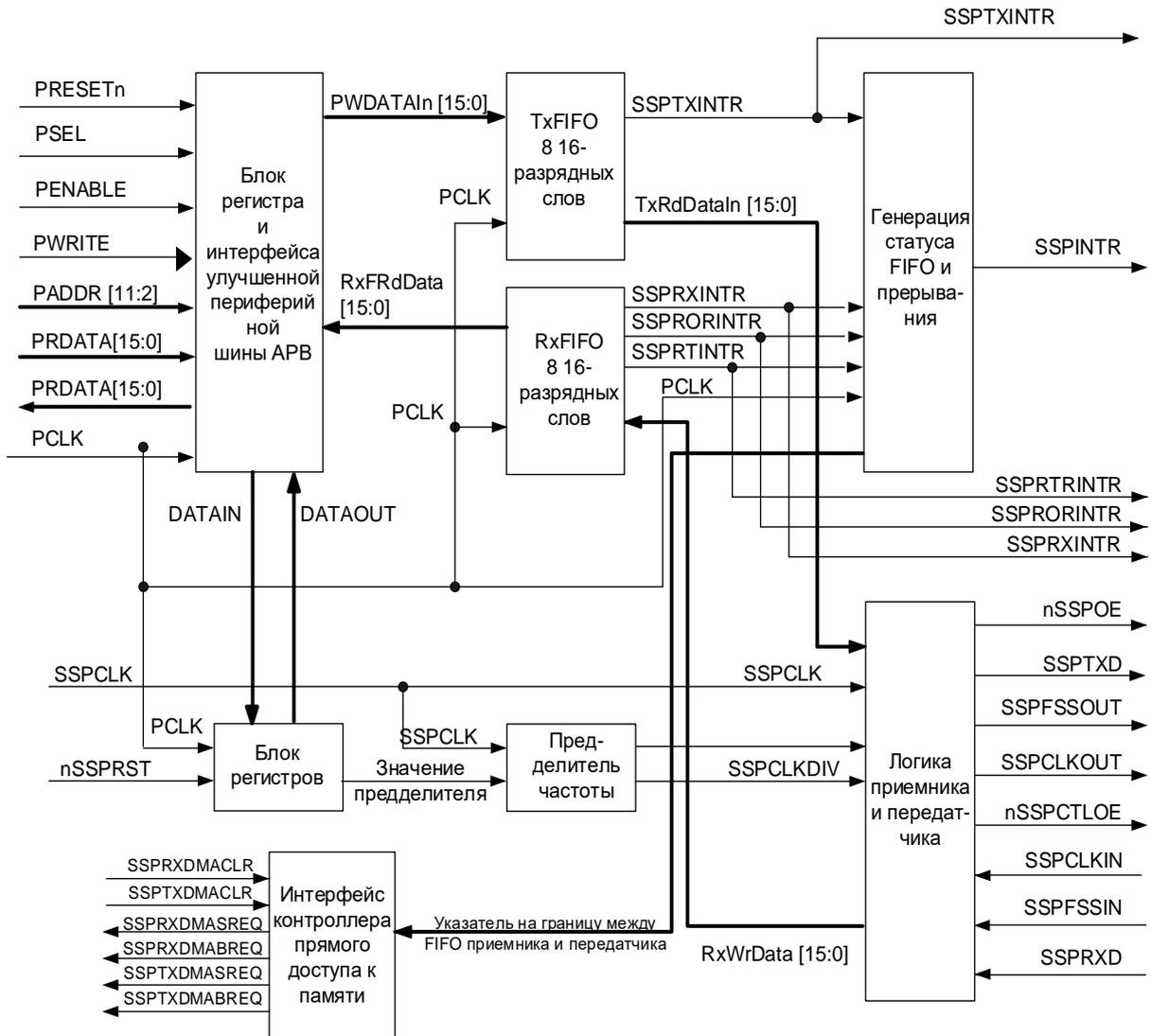


Рисунок 63 – Структурная схема модуля SSP

### 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;
- 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 8-битных 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 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 SSPSCKOUT 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 четыре независимо маскируемых прерывания:

- SSPTXINTR – запрос на обслуживание буфера передатчика;
- SSPRXINTR – запрос на обслуживание буфера приемника;
- SSPRORINTR – переполнение приемного буфера FIFO;
- SSPRTINTR – таймаут ожидания чтения данных из приемного FIFO.

Кроме того, формируется общий сигнал прерывания SSPINTR, возникающий в случае активности одного из вышеуказанных независимых немаскированных прерываний, который идет на контроллер CLIC.

Модуль также формирует сигналы запроса на прямой доступ к памяти (DMA) для совместной работы с контроллером DMA.

В зависимости от режима работы модуля сигнал SSPFSSOUT используется либо для кадровой синхронизации (интерфейс SSI, активное состояние – высокий уровень), либо для выбора ведомого режима (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

### 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, глубину 8 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Данные от центрального процессора сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных.

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, глубину 8 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Принятые от периферийного устройства данные сохраняются в этом буфере блоком приема данных до тех пор, пока не будут считаны центральным процессором.

### **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 CLIC, 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 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.

$$F_{SSPCLK} \leq F_{PCLK}.$$

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:

$$\begin{cases} F_{SSPCLK} (min) \geq 2 \times F_{SSPCLKOUT} (max) [for\ master\ mode], \\ F_{SSPCLK} (min) \geq 12x \times F_{SSPCLKIN} (max) [for\ slave\ mode]. \end{cases}$$

Аналогично, максимально допустимая частота сигнала SSPCLK определяется следующей системой соотношений, которые должны выполняться одновременно:

$$\begin{cases} F_{SSPCLK} (max) \leq 254 \times 256 \times F_{SSPCLKOUT} (min) [for\ master\ mode], \\ F_{SSPCLK} (max) \leq 254 \times 256 \times F_{SSPCLKIN} (min) [for\ slave\ mode]. \end{cases}$$

### 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.

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.

Для протокола SPI фирмы Motorola также задаются полярность и фаза сигнала (биты SPH 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.

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).

## 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

$$F_{SSPCLK} = \frac{F_{SSPCLK}}{CPSDVR \cdot (1 + SCR)} \quad (20)$$

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 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 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 8-битная управляющая последовательность. В течение всего времени передачи этой последовательности приемник не обрабатывает каких-либо входных данных. После того как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом общая длительность информационного кадра составляет от 13 до 25 бит.

### 23.10.1 Формат синхронного обмена SSI фирмы Texas Instruments

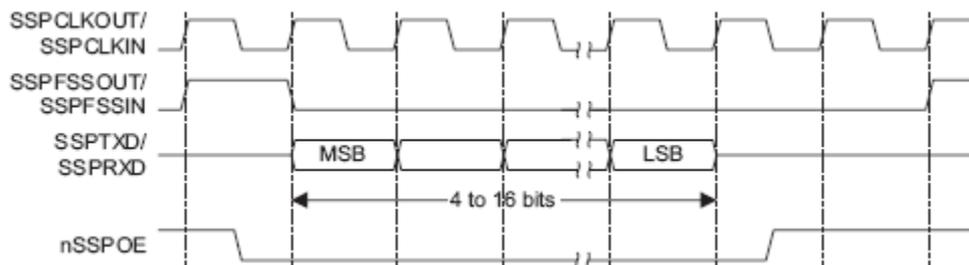


Рисунок 64 – Формат синхронного обмена протокола SSI (единичный обмен)

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 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 64 (передача единичного информационно кадра) и рисунок 65 (передача последовательности кадров).

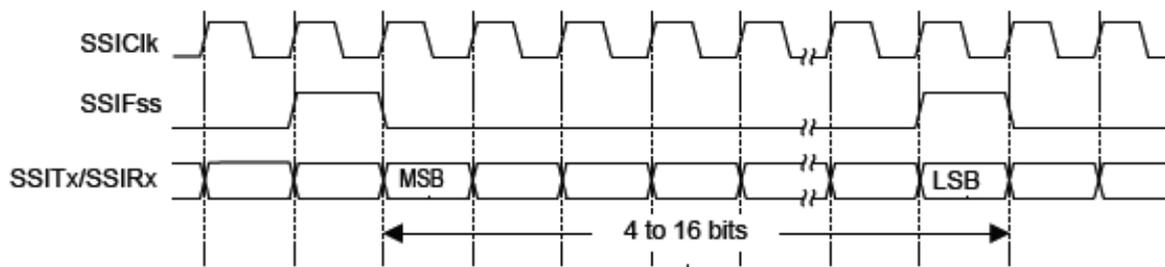


Рисунок 65 – Формат синхронного обмена протокола SSI (непрерывный обмен)

### 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 66 (одиночный обмен) и рисунок 67 (непрерывный обмен).

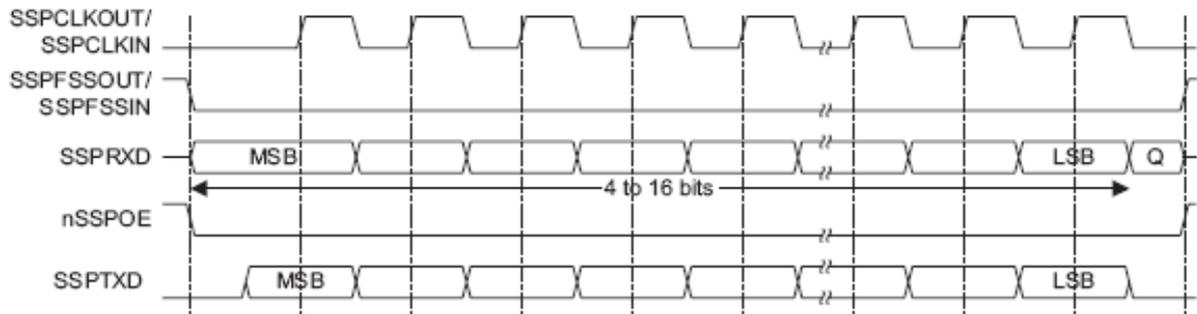


Рисунок 66 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (одиночный обмен)

Примечание – На рисунке 66 буквой Q обозначен сигнал с неопределенным уровнем.

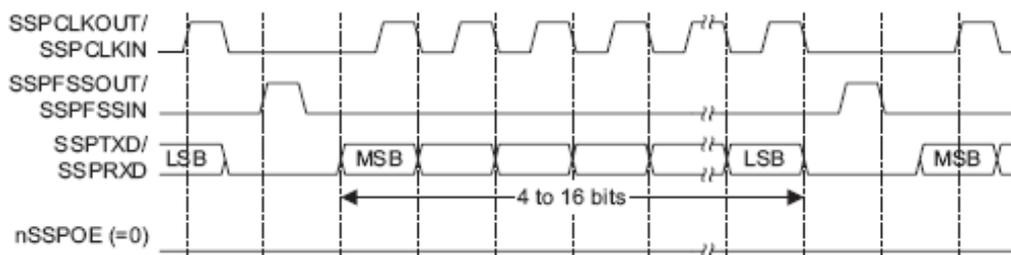


Рисунок 67 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (непрерывный обмен)

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 SSPTXD 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.

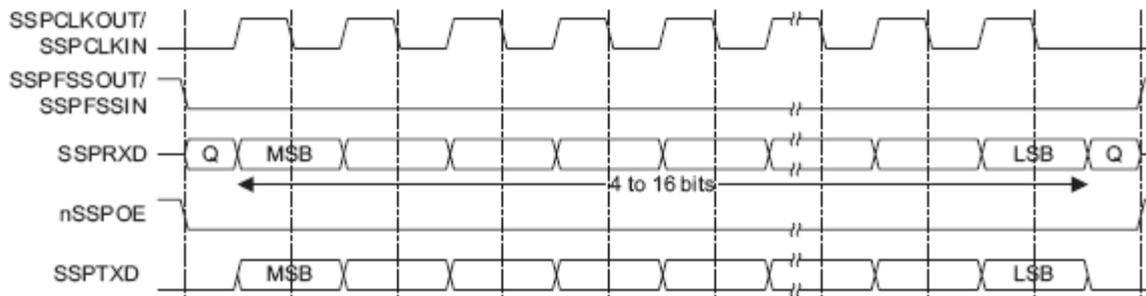
Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSPCLK.

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 68 (одиночный и непрерывный обмен).



Q – сигнал с неопределенным уровнем.

Рисунок 68 – Формат синхронного обмена протокола SPI, SPO=0, SPH=1

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 SSPTXD 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 будут сформированы значения первых бит передаваемых данных. В это же время включается линия SSPCLK и на ней формируется передний фронт сигнала.

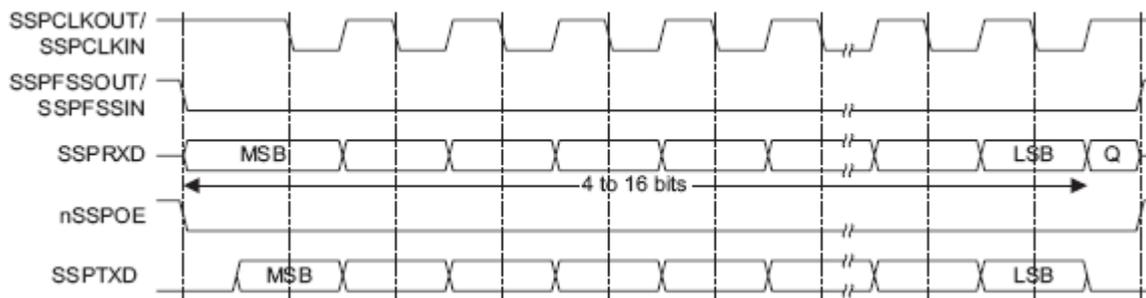
Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSPCLK.

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 69 (одиначный обмен) и рисунок 70 (непрерывный обмен).



Q – сигнал с неопределенным уровнем.

Рисунок 69 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (одиначный обмен)

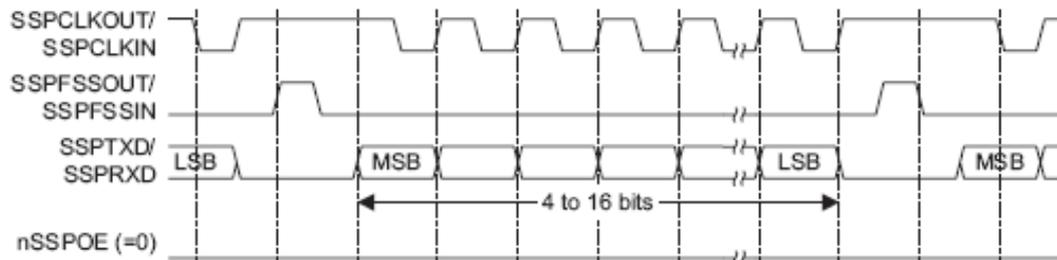


Рисунок 70 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (непрерывный обмен)

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 SSPTXD 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.

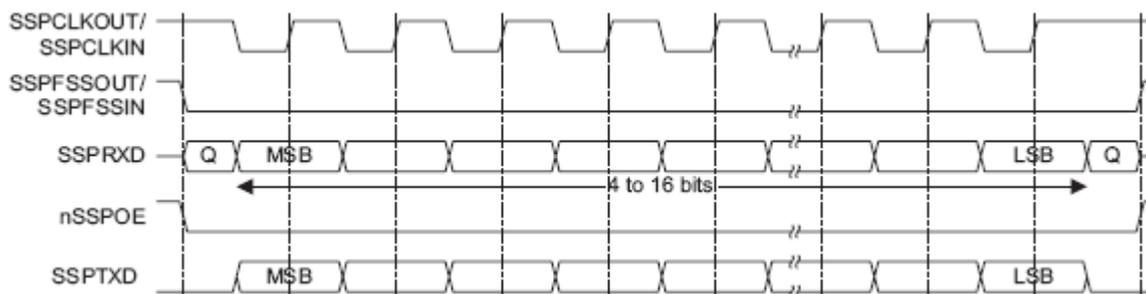
Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSPCLK.

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 71 (одиночный и непрерывный обмен).



Q – сигнал с неопределенным уровнем.

Рисунок 71 – Формат синхронного обмена протокола SPI, SPO=1, SPH=1

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 SSPTXD 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 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 72 (одиночный обмен) и рисунок 73 (непрерывный обмен).

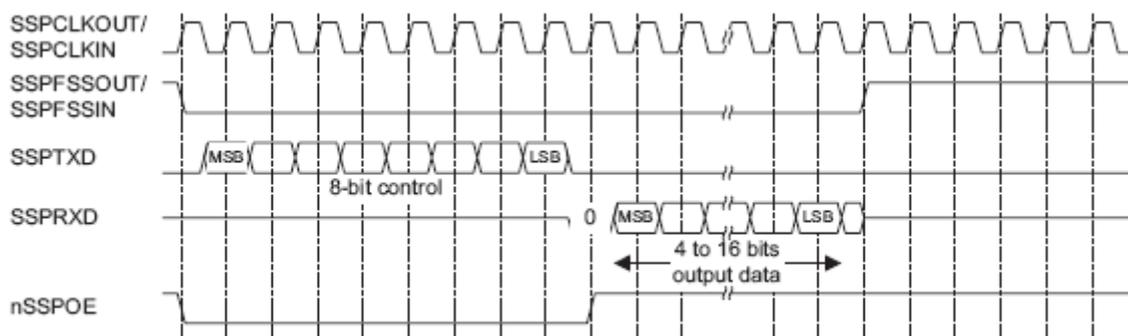


Рисунок 72 – Формат синхронного обмена протокола Microwire (одиночный обмен)

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 8-битной управляющей последовательности. В течение всего времени ее передачи приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

В данном режиме во время ожидания приемопередатчика:

- сигнал 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 одиночного 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 так же, как и одиночный 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.

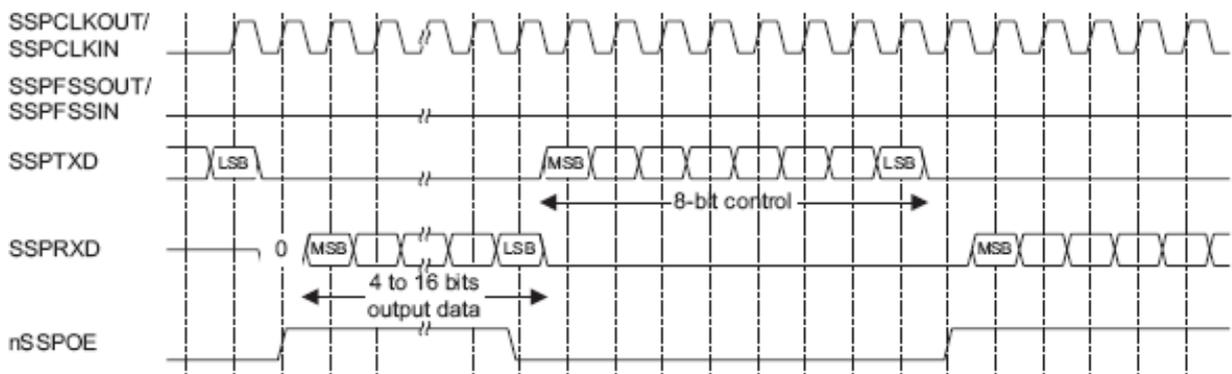


Рисунок 73 – Формат синхронного обмена протокола Microwire (непрерывный обмен)

### 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 74. 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.

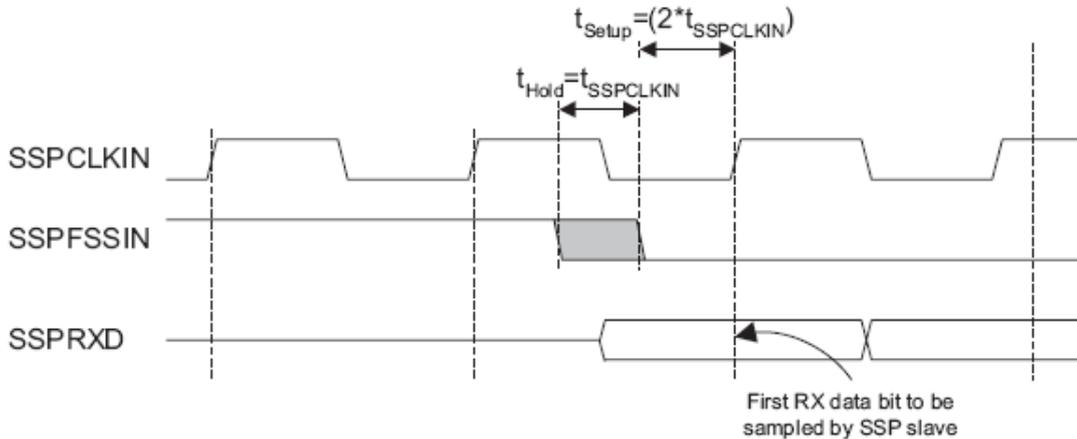


Рисунок 74 – Формат Microwire, требования к времени установки и удержания сигнала SSPFSSIN

### 23.11 Примеры конфигурации модуля в ведущем и ведомом режимах

Рисунки 75 – 77 показывают варианты подключения модуля PrimeCell SSP (PL022) к периферийным устройствам, работающим в ведущем или ведомом режиме.

Примечание – Модуль SSP не поддерживает динамическое изменение режима ведущий/ведомый. Каждый приемопередатчик должен быть изначально сконфигурирован в одном из этих режимов.

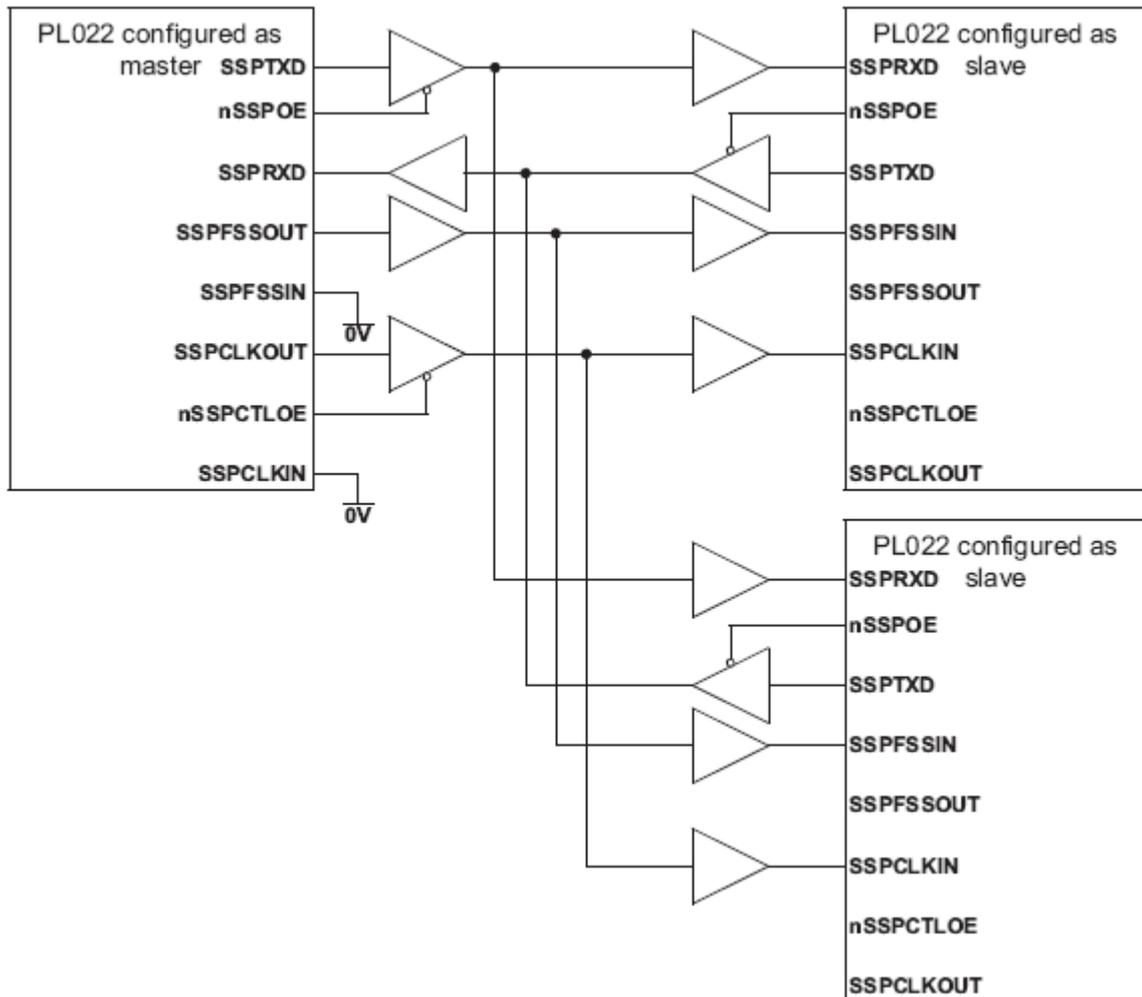


Рисунок 75 – Ведущее устройство SSP подключено к двум ведомым

Рисунок 75 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.

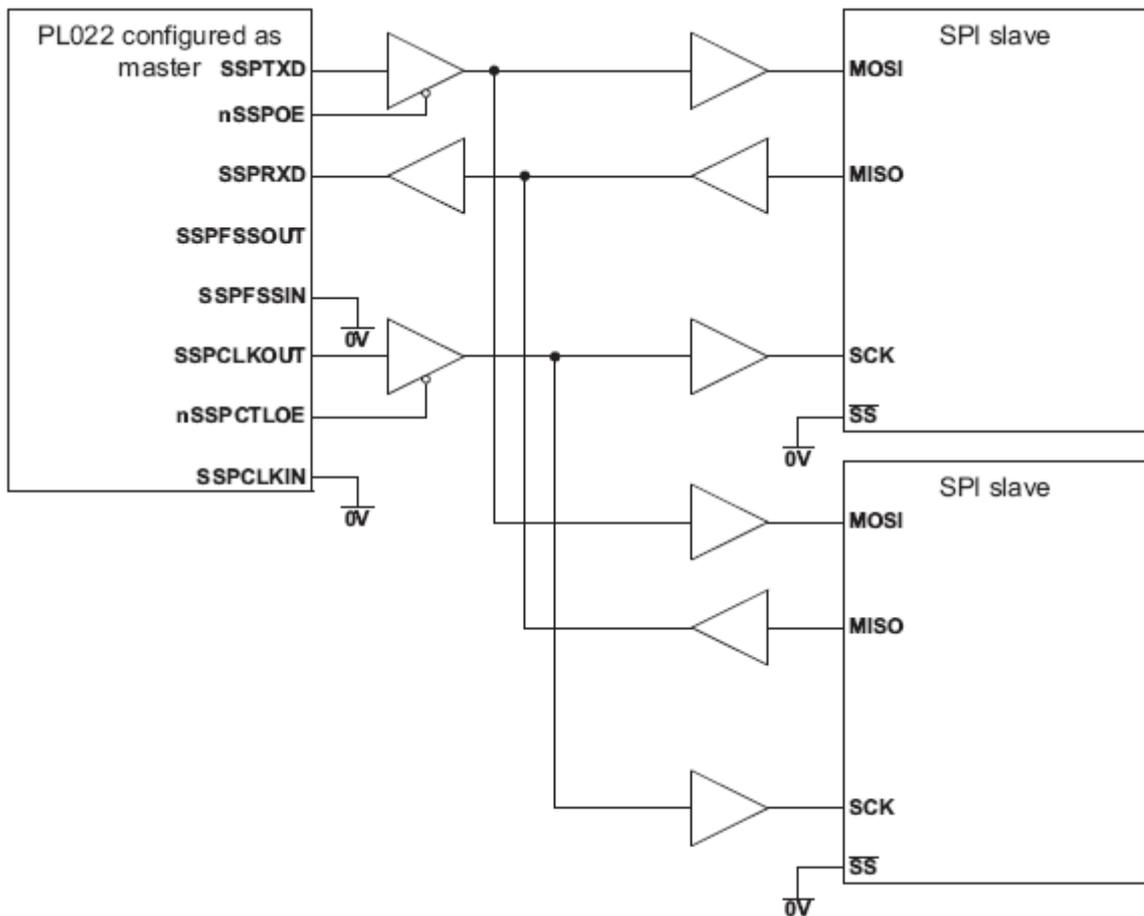


Рисунок 76 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим SPI

Рисунок 76 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.

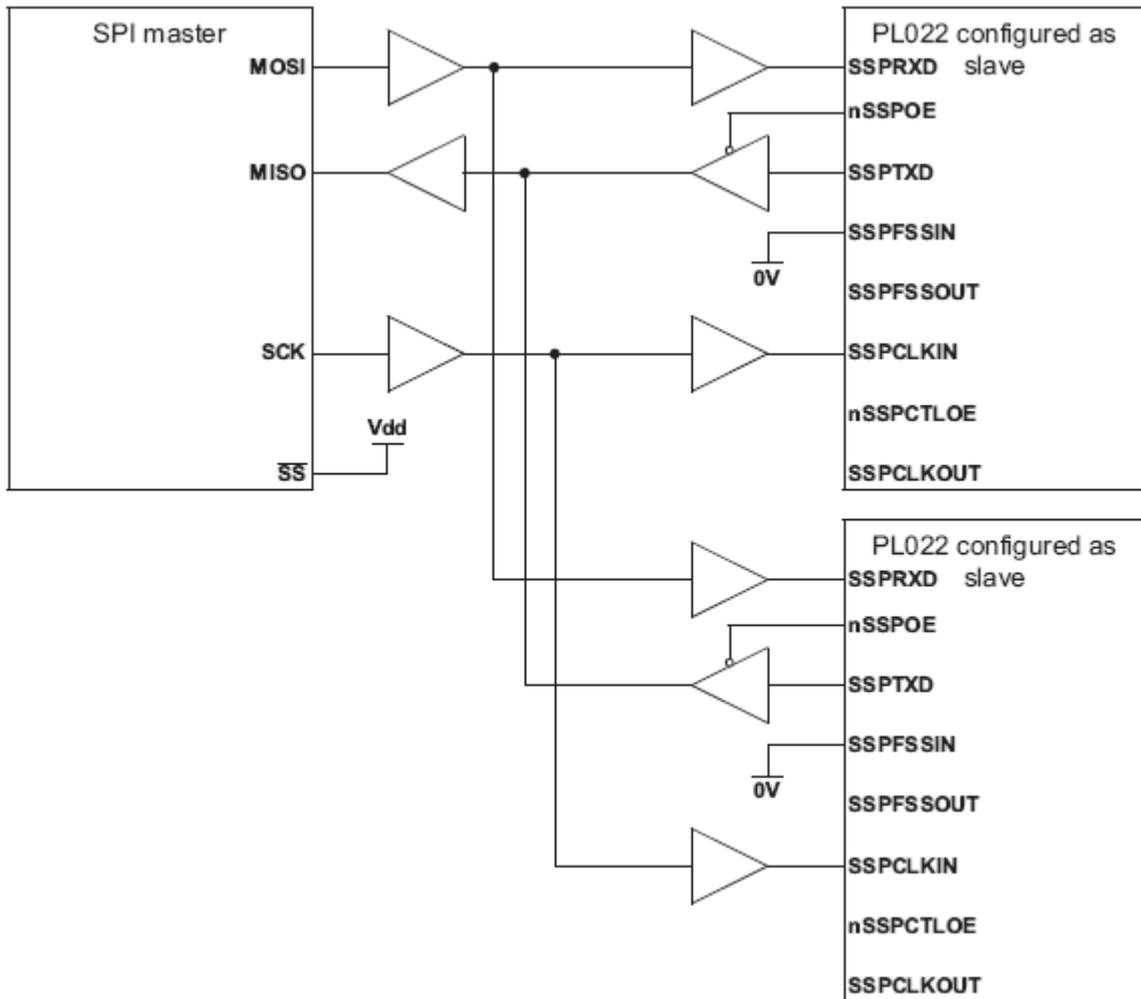


Рисунок 77 – Ведущее устройство, протокол SPI, подключено к двум ведомым модулям SSP

Рисунок 77 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 387 показывает значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 387 – 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)
1/2	4	4

Рисунок 78 показывает временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

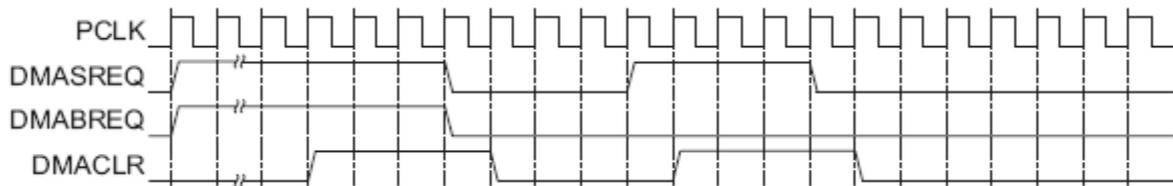


Рисунок 78 – Временные диаграммы обмена в режиме DMA

## 23.13 Программное управление модулем

### 23.13.1 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.

### 23.13.2 Описание регистров контроллера SSP

Данные о регистрах контроллера SSP приведены в таблице 388.

Таблица 388 – Обобщенные данные о регистрах модуля SSP

Базовый адрес	Наименование				Описание
0x5000_0000	SSP1				Регистры контроллера интерфейса SSP1
0x500C_8000	SS2				Регистры контроллера интерфейса SSP2
0x500D_0000	SSP3				Регистры контроллера интерфейса SSP3
Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x000	CR0	RW	0x0000	16	Регистр управления 0
0x004	CR1	RW	0x0	4	Регистр управления 1
0x008	DR	RW	0x----	16	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись)
0x00C	SR	RO	0x03	3	Регистр состояния
0x010	CPSR	RW	0x00	8	Регистр делителя тактовой частоты
0x014	IMSC	RW	0x0	4	Регистр маски прерывания
0x018	RIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования
0x01C	MIS	RO	0x0	4	Регистр состояния прерываний с учетом маскирования
0x020	ICR	WO	0x0	2	Регистр сброса прерывания
0x024	DMACR	RW	0x0	2	Регистр управления прямым доступом к памяти
<p>Примечание – В графе Тип указан вид доступа к регистру:                      RW – чтение и запись;                      RO – только чтение;                      WO – только запись</p>					

### 23.13.2.1 Регистр CR0

Регистр управления 0

Регистр CR0 содержит семь битовых полей, предназначенных для управления блоками модуля SSP.

Назначение бит регистра CR0 представлено в таблице 389.

Таблица 389 – 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: 1 – активация быстрого режима работы ведомого; 0 – нормальный режим работы
16	DSS[4]	Старший разряд размера слова данных: 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 $\frac{F_{SSPCLK}}{(CPSDVR * (1 + SCR))}$ где CPSDVR – четное число в диапазоне от 2 до 254 (см. Регистр CPSR), SCR – число от 0 до 255
7	SPH	Фаза сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. подраздел «Формат синхронного обмена SPI фирмы Motorola»
6	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. подраздел «Формат синхронного обмена SPI фирмы Motorola»
5...4	FRF	Формат информационного кадра: 00 – протокол SPI фирмы Motorola; 01 – протокол SSI фирмы Texas Instruments; 10 – протокол Microwire фирмы National Semiconductor; 11 – резерв
3...0	DSS[3:0]	Младшие 4 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		0001 – 18 бит 0010 – 19 бит 0011 – 20 бит 0100 – 21 бит 0101 – 22 бита ... 1110 – 31 бит 1111 – 32 бита

### 23.13.2.2 Регистр CR1

Регистр управления 1

Регистр CR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP.

Назначение бит регистра CR1 представлено в таблице 390.

Таблица 390 – 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: 0 – нормальный режим работы FIFO передатчика; 1 – активация сброса FIFO передатчика. Сбрасывается в ноль автоматически по завершении сброса
3	SOD	Запрет выходных линий в режиме ведомого устройства. Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. Бит SOD следует установить в случае, если данный ведомый модуль SSP не должен в настоящее время осуществлять передачу данных в линию SSP_TXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSP_TXD в ведомом режиме 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

### 23.13.2.3 Регистр DR

Регистр данных

Регистр SSPDR имеет разрядность 32 бита и предназначен для чтения принятых, и записи передаваемых данных.

Операция чтения обеспечивает доступ к последней несчитанной ячейке буфера FIFO приемника. Запись данных в этот буфер FIFO осуществляет блок приемника.

Операция записи позволяет занести очередное слово в буфер FIFO передатчика. Извлечение данных из этого буфера осуществляет блок передатчика. При этом извлеченные данные помещаются в регистр сдвига передатчика, откуда последовательно выдаются на линию SSP\_TXD с заданной скоростью информационного обмена.

В случае если выбран размер информационного слова менее 16 бит, перед записью в регистр 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 394.

Таблица 391 – Формат регистра DR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	DATA	Принимаемые данные 32 бита (чтение). Передаваемые данные 32 бита (запись). В случае, если выбран размер информационного слова менее 16 бит, 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

### 23.13.2.4 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 392.

Таблица 392 – 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

### 23.13.2.5 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 393.

Таблица 393 – 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

### 23.13.2.6 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 394.

Таблица 394 – 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: 1 – не маскирована; 0 – маскирована
5	TFEIM	Маска прерывания по пустоте FIFO передатчика: 1 – не маскирована; 0 – маскирована
4	RNEIM	Маска прерывания по наличию данных в FIFO приемника: 1 – не маскирована; 0 – маскирована
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 попыток его чтения в течение времени таймаута): 1 – не маскирована; 0 – маскирована
0	RORIM	Маска прерывания по переполнению буфера приемника: 1 – не маскирована; 0 – маскирована

### 23.13.2.7 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 395.

Таблица 395 – 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
5	TFERIS	Состояние до маскирования прерывания SSPTFEINTR
4	RNERIS	Состояние до маскирования прерывания SSPRNEINTR
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

### 23.13.2.8 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 396.

Таблица 396 – 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
5	TFEMIS	Состояние маскированного прерывания SSPTFEINTR
4	RNEMIS	Состояние маскированного прерывания SSPRNEINTR
3	TXMIS	Состояние маскированного прерывания SSPTXINTR
2	RXMIS	Состояние маскированного прерывания SSPRXINTR
1	RTMIS	Состояние маскированного прерывания SSPRTINTR
0	RORMIS	Состояние маскированного прерывания SSPRORINTR

### 23.13.2.9 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 397.

Таблица 397 – 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

### 23.13.2.10 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 398.

Таблица 398 – 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
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

### 23.14 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;
- SSPTFEINTR – отсутствие данных в FIFO передатчика;
- SSPTNBSYINTR – отсутствие данных в сдвиговом регистре передатчика;
- 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.

#### **23.14.1 SSPRXINTR**

Прерывание по заполнению буфера FIFO приемника. Формируется в случае, если буфер приемника содержит четыре или более несчитанных слов данных.

#### **23.14.2 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.

#### **23.14.3 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.

#### **23.14.4 SSPRTINTR**

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Данный механизм гарантирует, что пользователь будет знать о наличии в буфере приемника необработанных данных.

Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения, либо после приема новых слов данных по входной линии SSP\_RXD. Кроме того, оно может быть снято путем записи 1 в бит RTIC регистра сброса прерывания SSPTICR.

#### **23.14.5 SSPRNEINTR**

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.

#### **23.14.6 SSPTFEINTR**

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.

#### **23.14.7 SSPTNBSYINTR**

Прерывание по отсутствию данных в сдвиговом регистре передатчика.

#### **23.14.8 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.

## 24 Контроллер UART

Модуль универсального асинхронного приемопередатчика (UART – Universal Asynchronous Receiver-Transmitter) представляет собой периферийное устройство микроконтроллера.

В состав контроллера включен кодек (ENDEC – Encoder/Decoder) последовательного интерфейса инфракрасной (ИК) передачи данных в соответствии с протоколом SIR (SIR – Serial Infra Red) ассоциации Infrared Data Association (IrDA).

### 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 (1x16 – 65535x16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц.

Модулем поддерживаются стандартные элементы асинхронного протокола связи – стартового и стопового бит, а также бита контроля четности, которые добавляются перед передачей и удаляются после приема.

Независимо могут быть маскированы прерывания от буфера FIFO передатчика, буфера FIFO приемника, по таймауту приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки.

Модуль UART позволяет использовать DMA-контроллер для организации обмена данными и снижения нагрузки на ядро микроконтроллера.

Также модуль обеспечивает обнаружение ложных стартовых бит.

Формирование и обнаружения сигнала разрыва линии.

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\_N 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

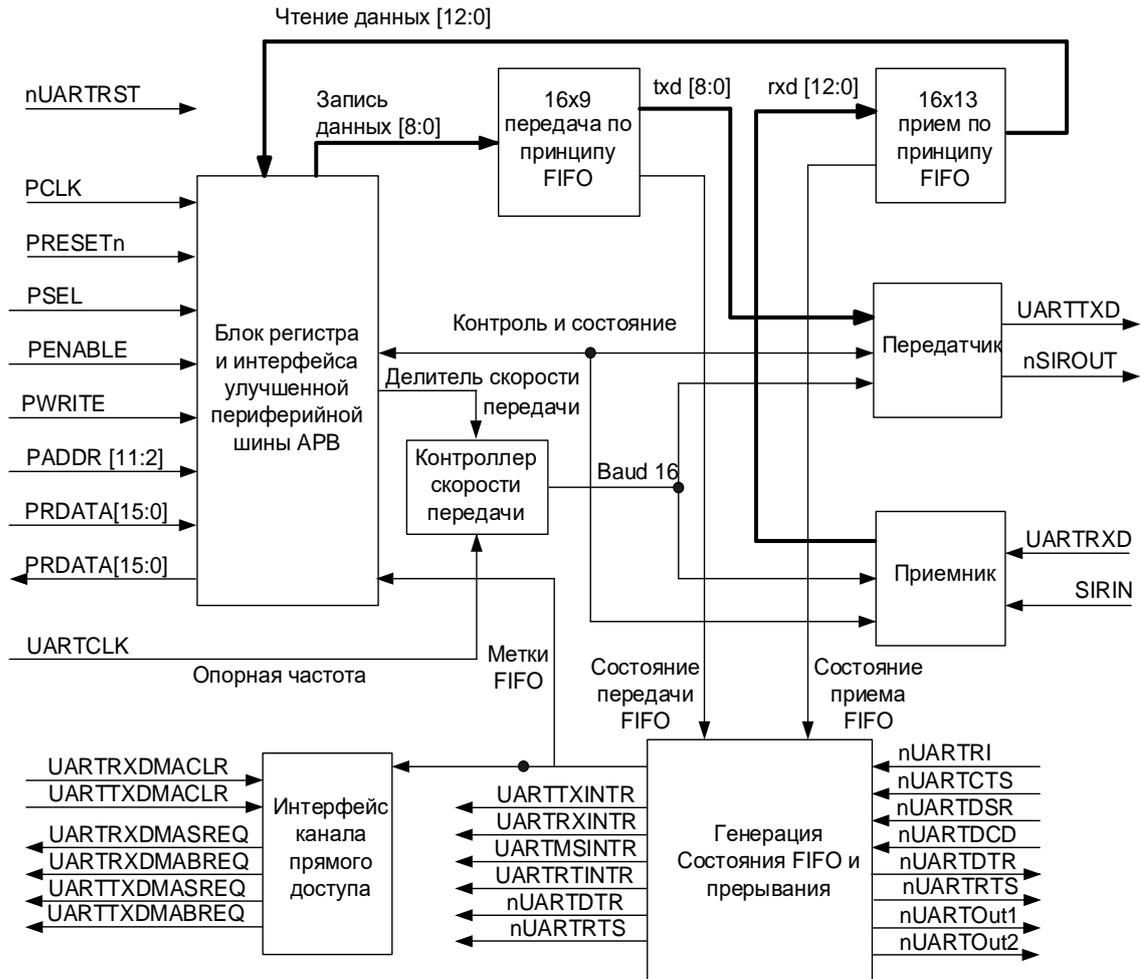


Рисунок 79 – Блок-схема универсального асинхронного приёмопередатчика (UART)

### 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 «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 80.

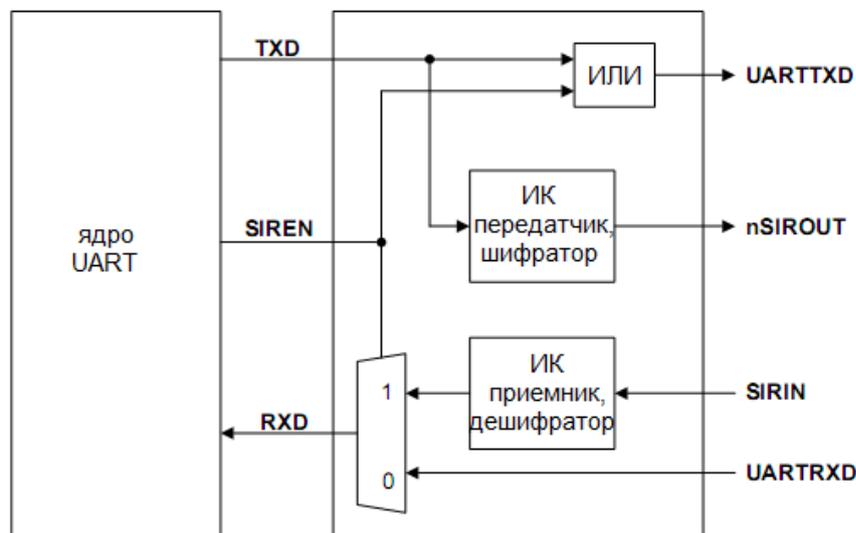


Рисунок 80 – Структурная схема кодека IrDA

### 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  $V_{aud16}$ , 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 «Программное управление модулем».

### 24.7.2 Тактовые сигналы

Частота тактового сигнала  $F_{UART\_CLK}$  должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

$$F_{UART\_CLK}(\min) \geq 16 \cdot \text{baud\_rate\_max};$$

$$F_{UART\_CLK}(\max) \leq 16 \cdot 65535 \cdot \text{baud\_rate\_min}.$$

$$F_{UART\_CLK}(\min) \geq 16 \cdot \text{baud\_rate\_max};$$

$$F_{UART\_CLK}(\max) \leq 16 \cdot 65535 \cdot \text{baud\_rate\_min}.$$

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\_N – 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

$$\begin{aligned} \text{Коэффициент деления} &= \frac{F_{UART\_CLK}}{16 \cdot \text{скорость передачи данных}} \\ &= IBRD + FBRD, \end{aligned} \quad (21)$$

где *IBRD* – целая часть коэффициента деления;  
*FBRD* – дробная часть коэффициента деления.



Рисунок 81 – 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\_N. 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 399 показывает назначение всех битов слова данных в FIFO буфере приемника.

Таблица 399 – 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 nSROUT 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

$$1,48 \text{ МГц} < IrLPBaud16 < 2,12 \text{ МГц} . \quad (24)$$

### 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.

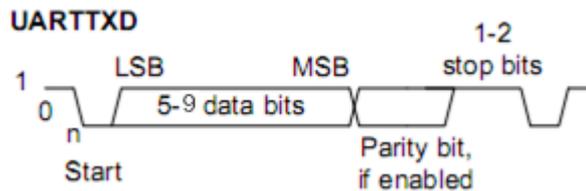


Рисунок 82 – Кадр передачи данных

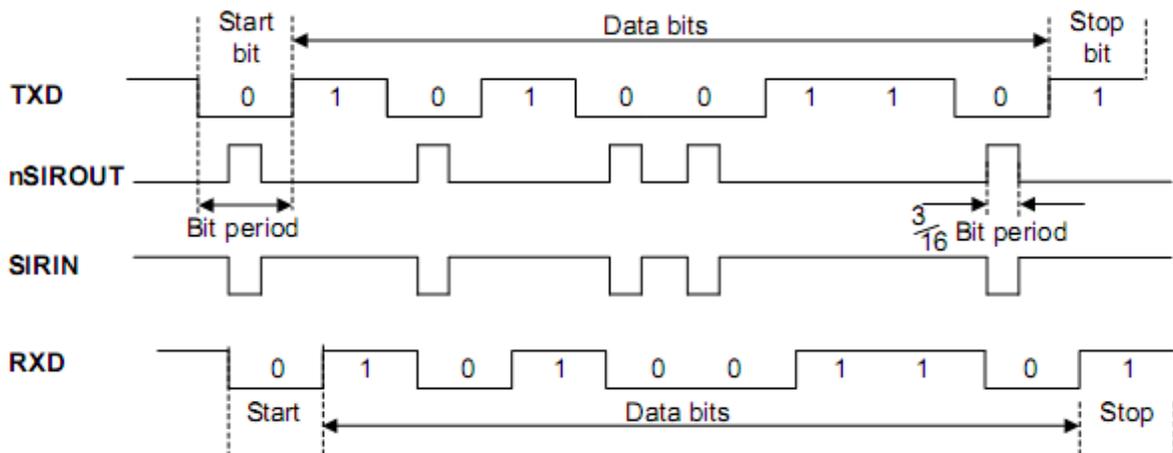


Рисунок 83 – Модуляция данных IrDA

## 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 400 – 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 84.

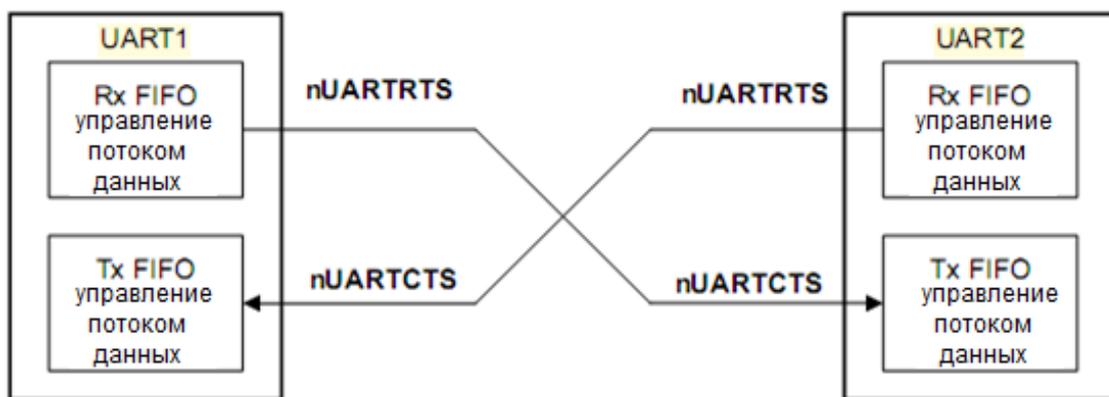


Рисунок 84 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

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 401 показаны необходимые установки для различных режимов управления потоком данных.

Таблица 401 – 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 402 показывает значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

Таблица 402 – 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
1/4	12	4
1/2	8	8
3/4	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 85 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.

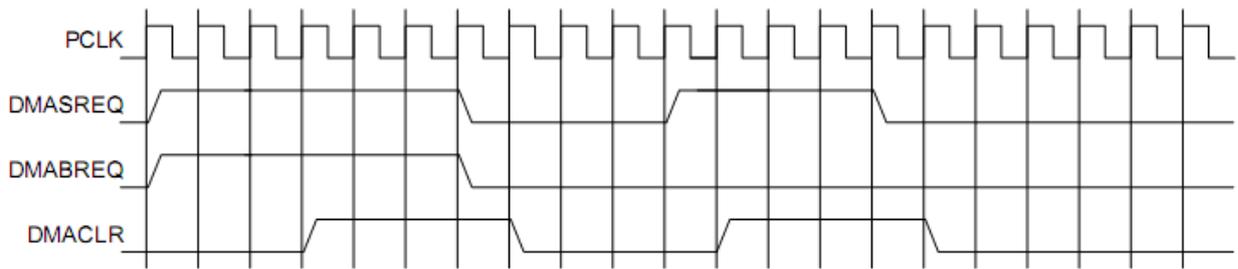


Рисунок 85 – Временные диаграммы одноэлементного и блочного запросов DMA

## 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;
- UARTTXINTR – прерывание от передатчика;
- UARTRNEINTR – FIFO приемника не пусто;
- UARTTFEINTR – FIFO передатчика пусто;
- UARNTNBSYINTR – сдвиговый регистр передатчика пуст;
- UARTRTINTR – прерывание по таймауту приемника;
- UARTMSINTR – прерывание по состоянию модема:
  - UARTRIINTR, изменение состояния линии nUARTRI;
  - UARTCTSINTR, изменение состояния линии nUARTCTS;
  - UARTDCDINTR, изменение состояния линии nUARTDCD;
  - UARTDSRINTR, изменение состояния линии nUARTDSR.
- UARTEINTR – ошибка:
  - UARTOEINTR, переполнение буфера;
  - UARTBEINTR, прерывание приема – разрыв линии;
  - UARTPEINTR, ошибка контроля четности;
  - UARTFEINTR, ошибка в структуре кадра.
- UARTINTR – логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR, UARTEINTR, UARTRNEINTR, UARTTFEINTR и UARNTNBSYINTR.

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 ошибки 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 UARTEMIS.

#### **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 разрешён 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 разрешён и его заполнение меньше или равно заданному пороговому значению. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи данных в буфера передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), данные в буферном регистре передатчика отсутствуют. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи одного байта данных, либо после сброса прерывания.

Для занесения данных в буфер FIFO передатчика необходимо записать данные в буфер либо перед разрешением работы приемопередатчика и прерываний, либо после разрешения работы приемопередатчика и прерываний.

Примечание – Прерывание передатчика работает по фронту, а не по уровню сигнала. В случае если модуль и прерывания от него разрешены до осуществления записи данных в буфер FIFO передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера FIFO.

### 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, UARTTTFEINTR 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 403 определяет режим доступа к регистру в соответствии с обозначениями:
  - RW – чтение и запись;
  - RO – только чтение;
  - WO – только запись.

### 24.12 Описание регистров контроллера UART

Данные о регистрах контроллера UART приведены в таблице 403.

Таблица 403 – Обобщенные данные о регистрах устройства

Базовый Адрес	Название				Описание
0x50008000	UART1				Регистры контроллера интерфейса UART1
0x50010000	UART2				Регистры контроллера интерфейса UART2
0x50038000	UART3				Регистры контроллера интерфейса UART3
0x500E8000	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	IPLS	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 ПДП
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 (таблица 404).

Таблица 404 – 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_N. 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.

Назначения битов регистра RSR\_ECR приведены в таблице 405.

Таблица 405 – 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
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

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.

### 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. Назначения битов регистра приведены в таблице 406.

Таблица 406 – 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
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 407.

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

$$1,48 \text{ МГц} < F_{IrLPBaud16} < 2,12 \text{ МГц}, \quad (26)$$

что, в свою очередь, гарантирует формирование кодеком импульсов данных с длительностью (1,41 – 2,11) мкс (в три раза длиннее периода сигнала IrLPBaud16).

Таблица 407 – 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. <b>Примечание</b> – 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 408.

Таблица 408 – 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 BFRD

Назначение бит регистра представлено в таблице ниже.

Таблица 409 – 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 шестиразрядного 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 410 содержит значения коэффициента деления для типичных скоростей передачи данных при частоте  $F_{UART\_CLK} = 7,3728$  МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр FBRD должен быть записан ноль.

Таблица 410 – Коэффициенты деления при частоте  $F_{UART\_CLK} = 7,3728$  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

В таблице 411 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте  $F_{UART\_CLK} = 4$  МГц.

Таблица 411 – Коэффициенты деления при частоте  $F_{UART\_CLK} = 4$  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\_N

Данный регистр обеспечивает доступ к разрядам с 29 по 22 регистра LCR. При сбросе все биты регистра LCR\_N обнуляются.

Назначение битов регистра описано в таблице 412.

Таблица 412 – Регистр LCR\_N

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 413)
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 битов: 0 – один стоповый бит; 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 413)
1	PEN	Разрешение проверки четности: 0 – кадр не содержит бита четности; 1 – бит четности передается в кадре и проверяется при приеме данных (таблица 413)
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 413 содержит данные об истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией LCR\_H.

Таблица 413 – Управление режимом контроля четности

PEN	EPS	SPS	Бит контроля четности
0	X	X	Не передается, не проверяется
1	1	0	Проверка четности слова данных
1	0	0	Проверка нечетности слова данных
1	0	1	Бит четности постоянно равен 1
1	1	1	Бит четности постоянно равен 0

Примечания

- Регистры 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.
- 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 битов регистра управления показано в таблице 414.

Таблица 414 – 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	Разрешение передачи. 1 – работа передатчика разрешена. Передача осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчик в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой. 0 – работа передатчика запрещена
7	LBE	Режим тестирования по шлейфу: 1 – шлейф разрешен; 0 – запрещен. В режиме разрешенного шлейфа: Если установлены бит SIREN=1 и бит регистра управления тестированием TCR SIRTEST=1, то сигнал с выхода кодека 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. Если бит SIRTEST=0, то выходная линия передатчика UART_TXDx коммутируется на вход приемника UART_RXDx. Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии. После сброса бит устанавливается в 0
6...3		Резерв. Не модифицируйте. При чтении выдаются нули
2	SIRLP	Выбор режима ИК обмена с пониженным энергопотреблением: 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; 0 – длительность импульсов данных равна 3/16 длительности передачи бита
1	SIREN	Разрешение работы кодека ИК передачи данных IrDA SIR: 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. В случае если UARTEN=0 значение бита не играет роли. 0 – запрещен. Сигнал nSIROUT находится в низком состоянии, данные на входе SIRIN не обрабатываются
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. Дождитесь 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 приведены в таблице 415.

Таблица 415 – 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; 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 приведено в таблице 416.

Таблица 416 – 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
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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 битов регистра RIS приведены в таблице 417.

Таблица 417 – 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
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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 битов регистра MIS приведены в таблице 418.

Таблица 418 – 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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
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 битов регистра ICR приведено в таблице 419.

Таблица 419 – 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
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 битов регистра DMACR приведено в таблице 420.

Таблица 420 – 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 UARTRXDMABREQ; 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 битов регистра TCR приведено в таблице 421.

Таблица 421 – 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 приёма 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 приёмника: 1 – разрешено; 0 – запрещено
0	ITEN	Перевод контроллера UART в тестовый режим: 1 – тестовый режим разрешён; 0 – тестовый режим запрещён

## 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 422 – Термины и определения

Альтернативная	Альтернативная структура управляющих данных канала. Вы можете установить соответствующий регистр для изменения типа структуры данных (см. раздел «Структура управляющих данных канала»)
C	Идентификатор номера канала прямого доступа. Например: C=1 – канал DMA 1 C=23 – канал DMA 23
Канал	Возможны конфигурации контроллера с числом каналов до 32. Каждый канал содержит независимые сигналы управления передачей данных, которые могут инициировать передачу данных по каналу DMA
Управляющие данные канала	Структура данных находится в системной памяти. Вы можете запрограммировать 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 86 показана упрощенная структурная схема контроллера.

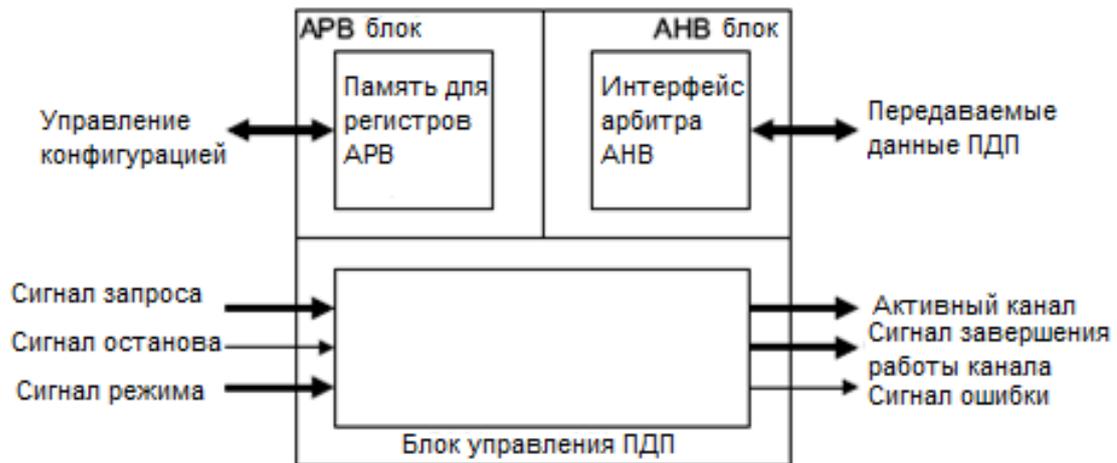


Рисунок 86 – Структурная схема контроллера

Контроллер состоит из следующих основных функциональных блоков:

- блок, подключенный к шине APB;
- блок, подключенный к шине AHB;
- управляющий блок DMA.

#### 25.3.1 Распределение каналов DMA

Таблица 423 – Распределение каналов DMA

Номер запроса	dma_req	dma_sreq	Описание
0	UART1TXDMABREQ	UART1TXDMASREQ	Запрос от передатчика UART1
1	UART1RXDMABREQ	UART1RXDMASREQ	Запрос от приёмника UART1
2	UART2TXDMABREQ	UART2TXDMASREQ	Запрос от передатчика UART2
3	UART2RXDMABREQ	UART2RXDMASREQ	Запрос от приёмника UART2
4	SSP1TXDMABREQ	SSP1TXDMASREQ	Запрос от передатчика SSP1
5	SSP1RXDMABREQ	SSP1RXDMASREQ	Запрос от приёмника SSP1
6	CRCDMABREQ	CRCDMASREQ	Запрос от аппаратного блока вычисления CRC
7	UART3TXDMABREQ	UART3TXDMASREQ	Запрос от передатчика UART3
8	UART3RXDMABREQ	UART3RXDMASREQ	Запрос от приёмника 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	ADCUIDMASREQ1	Запросы от блока АЦП для измерения напряжений и токов в электрической сети
13	ADCUIDMABREQ2	ADCUIDMASREQ2	
14	ADCUIDMABREQ3	ADCUIDMASREQ3	
15	ADCUIDMABREQ4	ADCUIDMASREQ4	
16	ADCUIDMABREQ5	ADCUIDMASREQ5	

Номер запроса	dma_req	dma_sreq	Описание
17	ADCUIDMABREQ6	ADCUIDMASREQ6	
18	ADCUIDMABREQ7	ADCUIDMASREQ7	
19	SSP2TXDMABREQ	SSP2TXDMASREQ	Запрос от передатчика SSP2
20	SSP2RXDMABREQ	SSP2RXDMASREQ	Запрос от приёмника SSP2
21	SSP3TXDMABREQ	SSP3TXDMASREQ	Запрос от передатчика SSP3
22	SSP3RXDMABREQ	SSP3RXDMASREQ	Запрос от приёмника SSP3
23	TMR4DMAREQ	TMR4DMAREQ	Запрос от таймера общего назначения TIMER4
24	UART4TXDMABREQ	UART4TXDMASREQ	Запрос от передатчика UART4
25	UART4RXDMABREQ	UART4RXDMASREQ	Запрос от приёмника 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 АНВ

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 АНВ.

### 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 374 отображает значения комбинаций шины HSIZE.

Таблица 424 – Комбинации шины HSIZE

HSIZE[2]*	HSIZE[1]	HSIZE[0]	Разрядность данных (бит)
0	0	0	8
0	0	1	16
0	1	0	32
	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.

В таблице 425 приведены значения комбинаций шины HPROT.

Таблица 425 – 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 «1», чтобы обозначить доступ к данным.

Для каждого цикла DMA возможен выбор режимов защиты данных передач источника и приемника. Более подробно это описано в подразделе «Структура управляющих данных канала».

Для каждого канала DMA также возможен выбор режима защиты данных. Более подробно это описано в подразделе «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 следующей таблице перечислены возможные комбинации.

Таблица 426 – Инкремент адреса

Разрядность данных	Величина инкремента
8	Байт, полуслово, слово
16	Полуслово, слово
32	Слово

Минимальная величина инкремента адреса всегда соответствует разрядности передаваемых данных. Максимальная величина инкремента адреса, осуществляемая контроллером, одно слово. Более подробно о настройке инкремента адреса написано в разделе Настройка управляющих данных. Этот раздел описывает разряды управления величиной инкремента адреса в управляющих данных канала.

**Примечание** – Если необходимо оставлять адрес неизменным при чтении или записи данных, для примера, при работе с FIFO, можно соответствующим образом настроить контроллер на работу с фиксированным адресом (см. раздел «Структура управляющих данных канала»).

## 25.4 Управление DMA

### 25.4.1 Правила обмена данными

Контроллер использует правила обмена данными (см. таблицу 427), 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 427 – 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 – если <code>dma_stall</code> в состоянии 1, работа контроллера приостановлена. После того, как <code>dma_stall</code> будет установлен в 0, контроллер устанавливает <code>dma_done[]</code> в состояние 1 продолжительностью один такт HCLK
10	Состояние <code>dma_waitonreq[C]</code> можно изменять только при выключенном канале

Правило	Описание
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 <sup>R</sup> 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 <sup>R</sup> передач контроллер устанавливает значение chnl_useburst_set[C] в состояние 0, если количество оставшихся передач меньше, чем 2 <sup>R</sup> . 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 <sup>R</sup>
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 <sup>R</sup> 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 <sup>R</sup> (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 <sup>R</sup> передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2 <sup>R</sup> (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных
18	Когда chnl_req_mask_set[C] установлен в 1, контроллер игнорирует запросы по dma_sreq[C] и dma_req[C]
*) Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре n_minus_1 меньше, чем значение 2 <sup>R</sup> , то контроллер не очистит разряды chnl_useburst_set и поэтому запросы по dma_sreq[C] будут маскированы. Если периферия не устанавливает dma_req[C] в состояние 1, то контроллер никогда не выполнит необходимых передач	

При отключении канала контроллер осуществляет DMA\_передачи согласно правилам, приведенным в таблице 428.

Таблица 428 – Правила осуществления DMA передач при «запрещенных» каналах

Правило	Описание
19	Если dma_req[C] установлен в 1, то контроллер устанавливает dma_done[C] в 1. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен)
20	Если dma_sreq[C] установлен в 1, то контроллер устанавливает dma_done[C] в 1 при условии dma_waitonreq[C] в 1 и chnl_useburst_set[C] в состоянии 0. Это позволяет контроллеру показать центральному процессору запрос готовности, даже если канал выключен (запрещен)
21	dma_active[C] всегда удерживается в состоянии 0

### 25.4.2 Диаграммы работы контроллера DMA

Данный раздел описывает следующие примеры функционирования контроллера с использованием правил обмена данными (Таблица 427):

- импульсный запрос на обработку;
- запрос по уровню на обработку;
- флаги завершения;
- флаги ожидания запроса на обработку.

Примечание – Все диаграммы, показанные на рисунках 87 – 90, подразумевают следующее:

#### 25.4.2.1 Импульсный запрос на обработку

Рисунок 87 показывает временную диаграмму работы контроллера DMA при получении импульсного запроса от периферии.

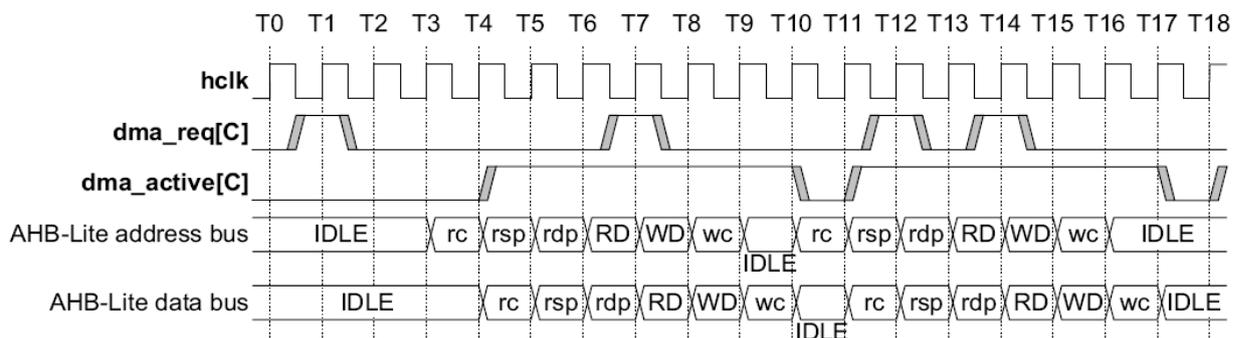


Рисунок 87 – Диаграмма работы при получении импульсного запроса от периферийного блока

Таблица 429 – Пояснения к диаграмме работы при получении импульсного запроса

T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C

T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7	При установленном dma_active[C] в 1 и при условии, что chnl_req_mask_set[C] находится в состоянии 0, контроллер обнаруживает импульс запроса на обработки по каналу C (см. правило 7). Контроллер обработает этот запрос в течение следующего арбитража
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T10-T11	Контроллер удерживает dma_active[C] в 0 как минимум на один такт HCLK (см. правило 5)
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C], так как ранее на такте T7 был получен запрос на обработку (см. правила 2 и 3)
T12	При установленном dma_active[C] в 1 и при условии, что chnl_req_mask_set[C] находится в состоянии 0, контроллер обнаруживает импульс запроса на обработку по каналу C (см. правило 7). Контроллер обработает этот запрос в течение следующего арбитража
T14	Контроллер игнорирует запрос по каналу C из-за отложенного запроса, полученного на такте T12
T17	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T17-T18	Контроллер удерживает dma_active[C] как минимум на один такт HCLK (см. правило 5)
T18	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C], так как ранее на такте T12 был получен запрос на обработку (см. правила 2 и 3)

### 25.4.2.2 Запрос на обработку по уровню.

Рисунок 88 показывает временную диаграмму работы контроллера DMA при получении от периферии запроса на обработку по уровню.

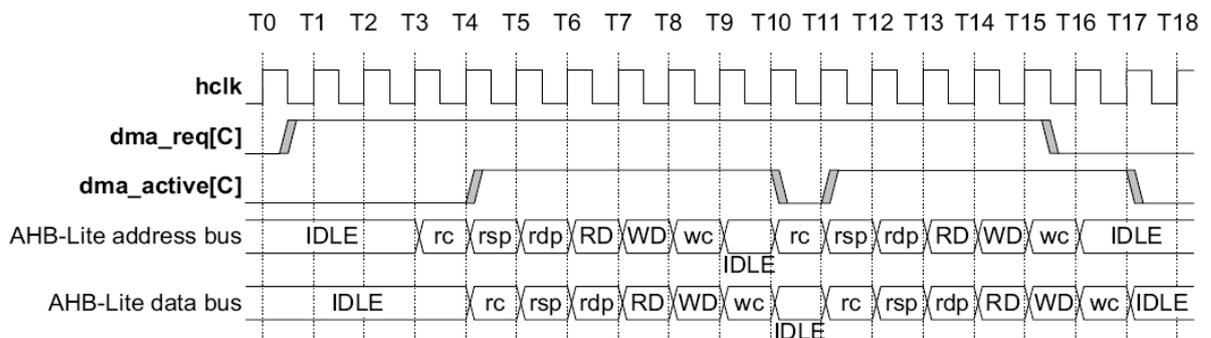


Рисунок 88 – Диаграмма работы при получении запроса на обработку по уровню

Таблица 430 – Пояснения к диаграмме работы при получении запроса на обработку по уровню

T1	Контроллер обнаружил запрос на обработку по каналу C (Таблица 427, 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: <code>rc</code> – чтение настроек канала, <code>channel_cfg</code> ; <code>rsp</code> – чтение указателя адреса окончания данных источника, <code>src_data_end_ptr</code> ; <code>rdp</code> – чтение указателя адреса окончания данных приемника, <code>dst_data_end_ptr</code> .
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: <code>RD</code> – чтение данных; <code>WD</code> – запись данных.
T9-T10	Контроллер осуществляет запись настроек канала, <code>channel_cfg</code> , где <code>wc</code> – 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 89 демонстрирует функционирование сигнала (флага) `dma_done[]` при следующих условиях:

- `dma_stall` и `dma_waitonreq[]` находятся в состоянии 0;
- `dma_stall` установлен в 1;
- `dma_waitonreq[]` установлен в 1.

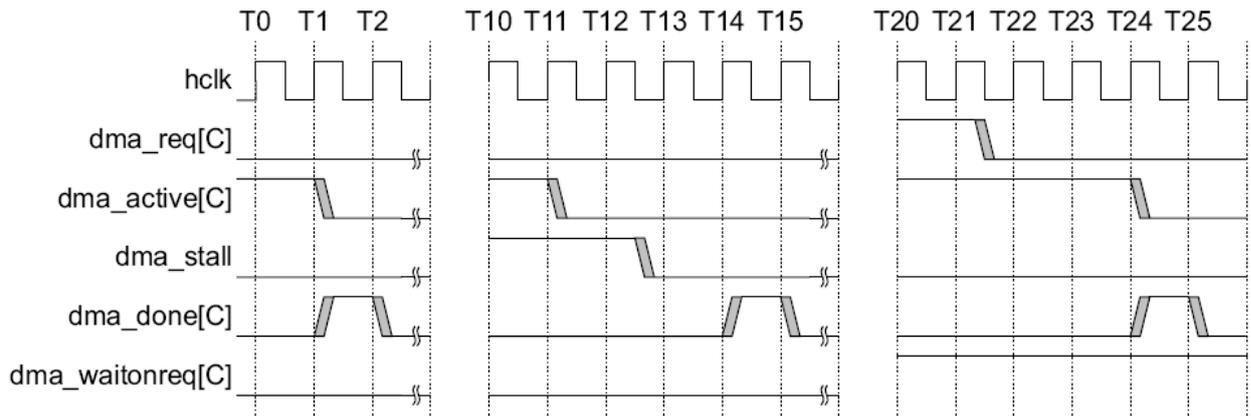


Рисунок 89 – Диаграммы функционирования dma\_done

Таблица 431 – Пояснения функционирования dma\_done, такты от T0 до T2

T1	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T1-T2	Контроллер завершает цикл DMA, и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт HCLK (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6)

Таблица 432 – Пояснения функционирования dma\_done, такты от T10 до T15

T11*	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T12-T13	Периферийный блок сбрасывает сигнал dma_stall
T14-T15	Контроллер завершает цикл DMA, и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт HCLK (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6)
* 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 433 – 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 90, 91 демонстрируют примеры использования флагов ожидания запроса на обработку при выполнении  $2^R$  передач и одиночных передач:

- диаграмма работы контроллера DMA при использовании периферией dma\_waitonreq;
- диаграмма работы контроллера DMA при использовании периферией dma\_waitonreq совместно с dma\_sreq.

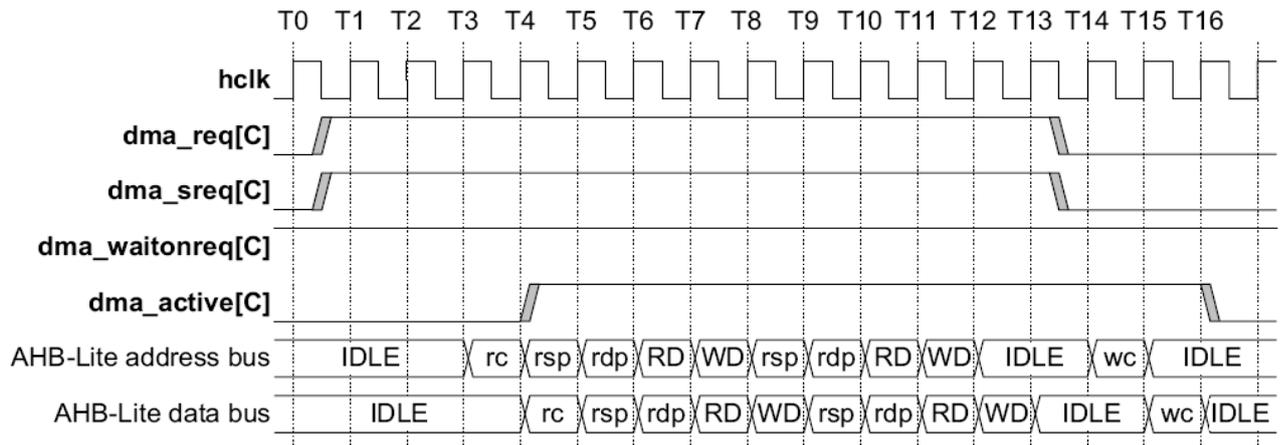


Рисунок 90 – Диаграмма работы контроллера DMA при использовании dma\_waitonreq

Таблица 434 – 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 $2^R$ (см. правило 15)

На рисунке 91 приведена работа контроллера DMA при установке `dma_waitonreq` в 1 и выполнении одиночной DMA передачи.

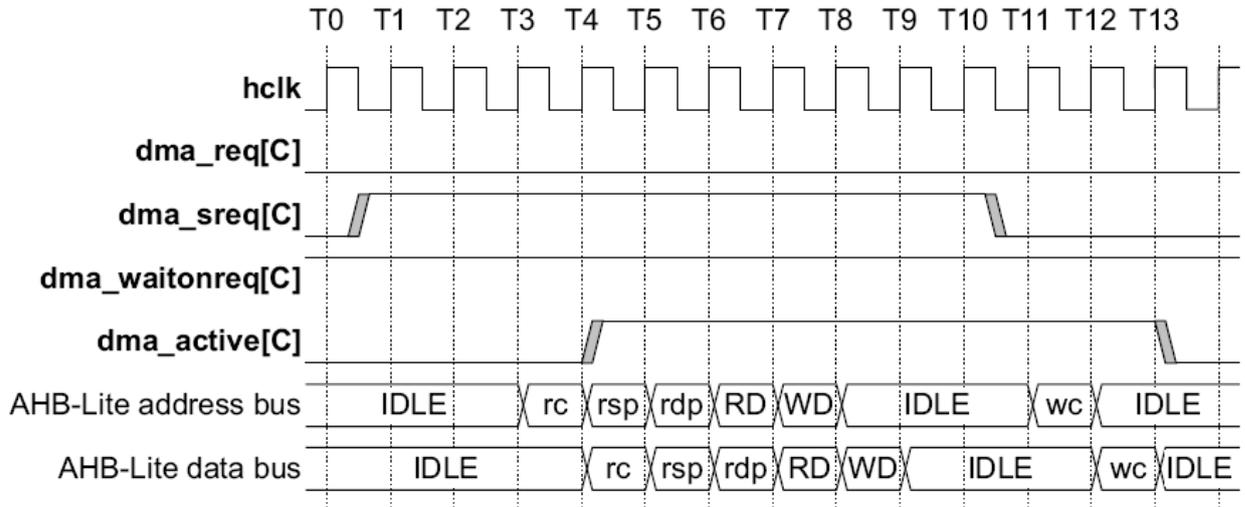


Рисунок 91 – Работа DMA при использовании `dma_waitonreq` совместно с `dma_sreq`

Таблица 435 – Пояснения работы DMA при использовании `dma_waitonreq` совместно с `dma_sreq`

T0-T13	Периферийный блок должен оставлять состояние <code>dma_waitonreq[C]</code> постоянно (см. правило 10)
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что <code>chnl_useburst_set[C]</code> находится в состоянии 0 (см. правила 13 и 14)
T3-T4	Контроллер отвечает на <code>dma_sreq[C]</code> запрос (см. правила 16).
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. Это запрос в ответ на <code>dma_sreq[]</code> , таким образом, R=0 и, следовательно, контроллер исполнит 1 DMA передачу
T10-T11	Периферийный блок сбрасывает сигнал <code>dma_sreq[C]</code>
T12_T13	Контроллер осуществляет запись настроек канала, <code>channel_cfg</code> , где wc – запись настроек канала, <code>channel_cfg</code>
T13	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , 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 436 показывает возможную периодичность арбитража.

Таблица 436 – 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 разделе «Управляющие данные 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. В таблице 437 приведены уровни приоритета каналов DMA в порядке его уменьшения.

Таблица 437 – 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. На рисунке 92 приведен алгоритм выбора следующего канала для обслуживания.



Рисунок 92 – 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 438 – Типы циклов DMA

cycle_ctrl	Описание
3'b000	Структура управляющих данных канала в запрещенном состоянии
3'b001	Обычный цикл DMA
3'b010	Авто-запрос
3'b011	Режим пинг-понг
3'b100	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
3'b101	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала
3'b110	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
3'b111	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала

Примечание – Разряды `cycle_ctrl` находятся в области памяти, отведенной под `channel_cfg` – см. подраздел «Структура 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. 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.

#### 25.4.5.4 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 93 демонстрирует пример функционирования контроллера в режиме «пинг-понг».

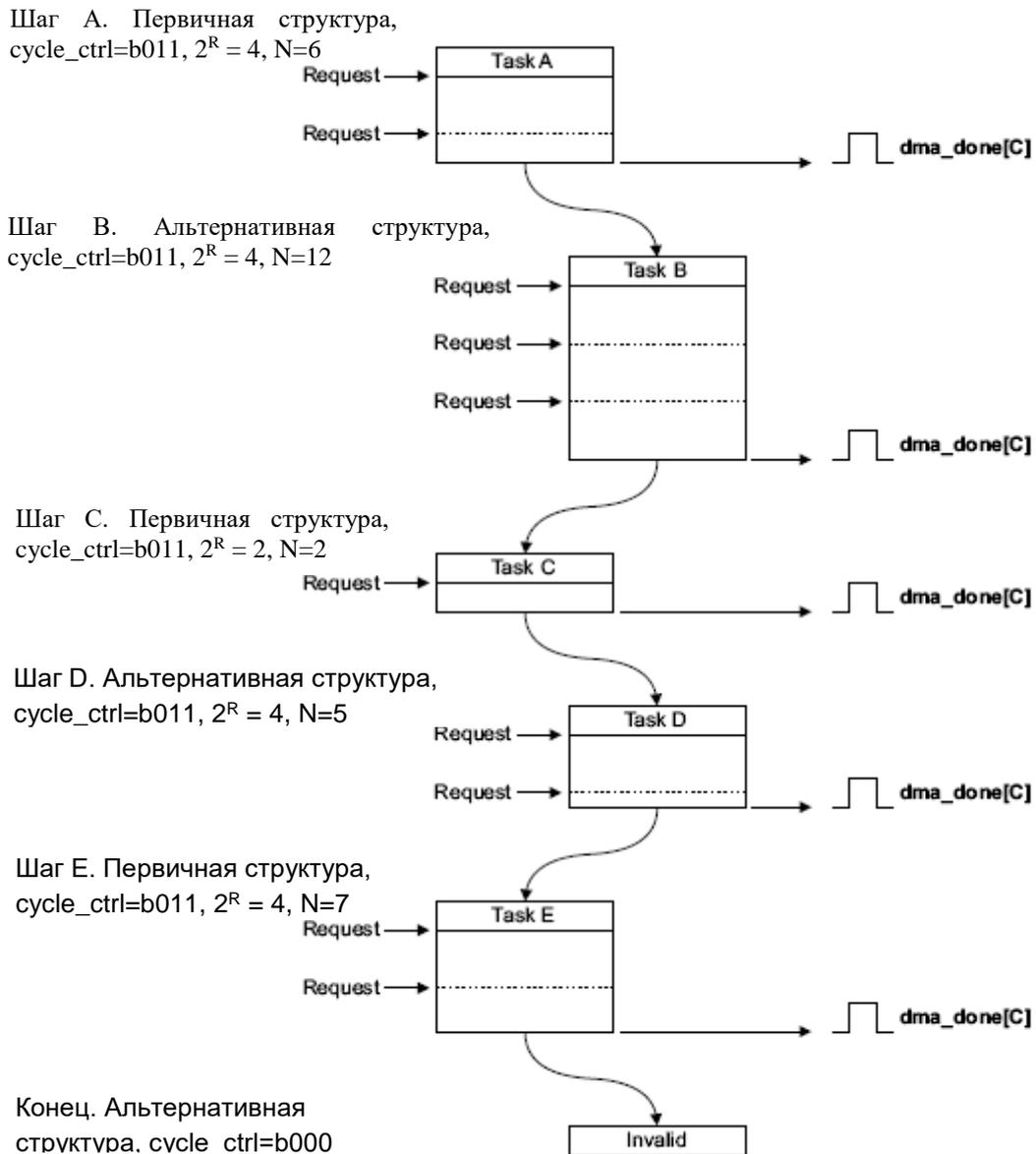


Рисунок 93 – Пример функционирования контроллера в режиме «пинг-понг»

Пояснения к рисунку

- 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 4 передачи DMA.  
 Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.  
 Контроллер выполняет оставшиеся 2 передачи DMA.  
 Контроллер устанавливает dma\_done[C] в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража

После выполнения шага 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 4 передачи DMA.  
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.  
Контроллер выполняет 4 передачи DMA.  
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.  
Контроллер выполняет оставшиеся 4 передачи DMA.  
Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража.

После выполнения шага 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 2 передачи DMA.  
Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража.

После выполнения шага С процессор может установить первичные управляющие данные канала для шага E.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг D:

Шаг D    Контроллер выполняет 4 передачи DMA.  
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.  
Контроллер выполняет оставшуюся передачу DMA.  
Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг E:

Шаг E    Контроллер выполняет 4 передачи DMA.  
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.  
Контроллер выполняет оставшиеся 3 передачи DMA.  
Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража.

Если контроллер получит новый запрос на обработку от данного канала и этот запрос будет самым приоритетным, контроллер предпримет попытку выполнения следующего шага. Однако из-за того, что процессор не установил альтернативные управляющие данные, и по 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 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные. Затем контроллер выполняет еще 4 передачи DMA, используя первичные управляющие данные. Контроллер продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;
- контроллер считает «неправильную» структуру управляющих данных.

Примечание – После исполнения контроллером N передач с использованием первичных управляющих данных он делает эти управляющие данные «неправильными» путем установки `cycle_ctrl` в `3'b000`.

Контроллер устанавливает флаг `dma_done[C]` в этом режиме работы только тогда, когда передача DMA заканчивается с использованием основного цикла.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. В таблице 439 приведены области памяти `channel_cfg`, которые должны быть определены константами, а также, значения которых определяются пользователем.

Таблица 439 – 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	<code>dst_inc</code>	<code>2'b10</code>	Контроллер производит инкремент адреса пословно
29, 28	<code>dst_size</code>	<code>2'b10</code>	Контроллер осуществляет передачу пословно
27, 26	<code>src_inc</code>	<code>2'b10</code>	Контроллер производит инкремент адреса пословно
25, 24	<code>src_size</code>	<code>2'b10</code>	Контроллер осуществляет передачу пословно
17...14	<code>R_power</code>	<code>4'b0010</code>	Контроллер выполняет 4 передачи DMA
3	<code>next_useburst</code>	<code>1'b0</code>	Для данного режима этот разряд должен быть равен 0
2...0	<code>cycle_ctrl</code>	<code>3'b100</code>	Контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями, определяемыми пользователем			
23...21	<code>dst_prot_ctrl</code>	-	Определяет состояние HPROT при записи данных в приемник

Разряды	Обозначение	Значение	Описание
20...18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13...4	n_minus_1	N*	Настраивает контроллер на выполнение N передач DMA, где N кратно 4

\* Так как разряды R\_power установлены в состояние 2, необходимо задавать значение N кратное 4. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные

Рисунок 94 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	0XXXXXXXXXX
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b101, 2 <sup>R</sup> = 2, N = 8	0XXXXXXXXXX
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b101, 2 <sup>R</sup> = 8, N = 5	0XXXXXXXXXX
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2 <sup>R</sup> = 4, N = 4	0XXXXXXXXXX

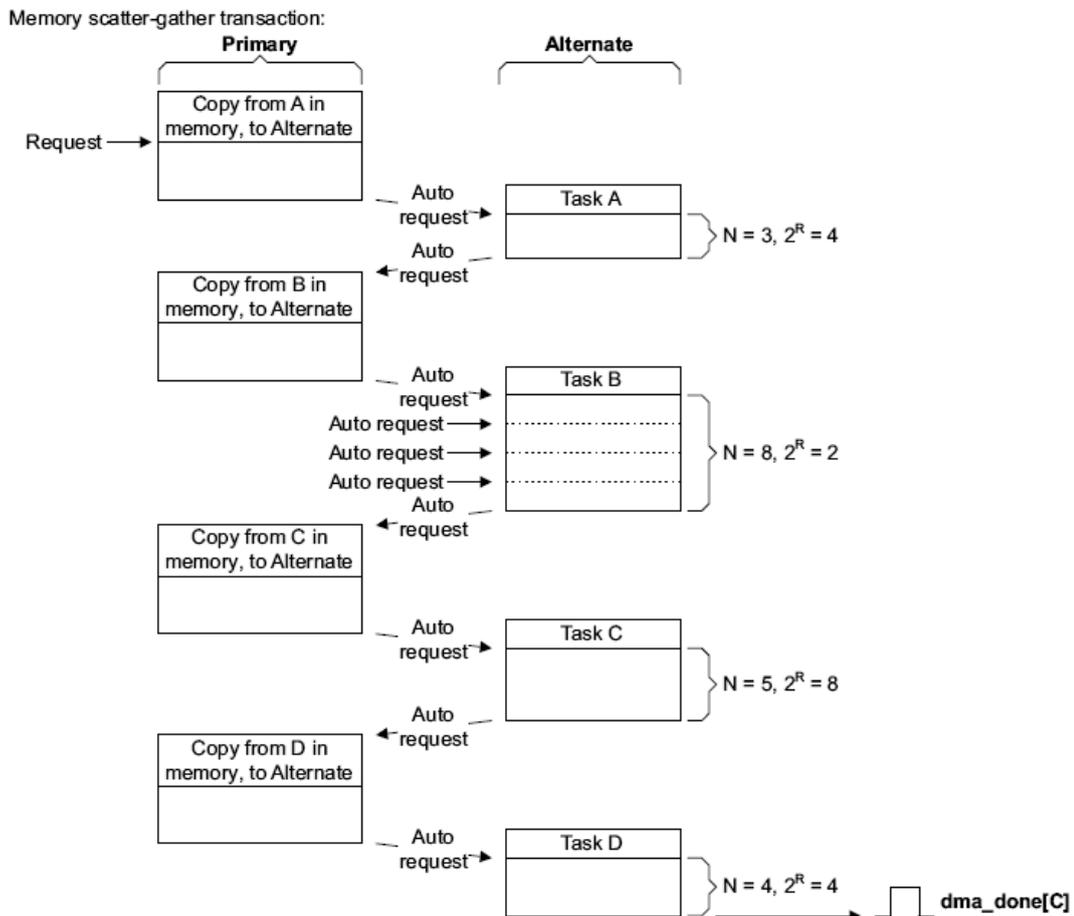


Рисунок 94 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

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 4 слов, необходимо установить  $2^R$  в 4. В этом примере количество передач равно 4 и поэтому  $N$  установлен в 16.

2. Процессор записывает управляющие данные для шагов A, B, C, D в область памяти с адресом, указанным в `src_data_end_ptr`.

3. Процессор разрешает работу канала DMA.

Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по `dma_req[]` или запроса от процессора. Порядок выполнения следующий:

Первичная, копирование A

По получению запроса на обслуживание контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага A.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг A

Контроллер выполняет шаг A. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование B

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага B.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг B

Контроллер выполняет шаг B. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование C

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага C.

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг C

Контроллер выполняет шаг C. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование D

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D.

Контроллер устанавливает `cycle_ctrl` первичных управляющих данных в `3'b000` для индикации о том, что эта структура управляющих данных является «неправильной».

Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг D

Контроллер выполняет шаг D, используя основной цикл DMA.

Контроллер устанавливает флаг `dma_done[C]` в состояние 1 на один такт сигнала HCLK и входит в процедуру арбитража.

#### 25.4.5.6 Режим работы с периферией «исполнение с изменением конфигурации»

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал `dma_active[C]` в 0.

Примечание – Это единственный случай, при котором контроллер не осуществляет процедуру арбитража после выполнения передачи DMA, используя первичные управляющие данные.

После того, как этот цикл завершился, контроллер выполняет арбитраж и по получении запроса на обслуживание от периферии, имеющего наивысший приоритет, он выполняет еще 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал `dma_active[C]` в 0.

Контроллер продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;
- контроллер считает «неправильную» структуру управляющих данных.

Примечание – После исполнения контроллером N передач с использованием первичных управляющих данных, он делает эти управляющие данные «неправильными» путем установки `cycle_ctrl` в `3'b000`.

Контроллер устанавливает флаг `dma_done[C]` в этом режиме работы только тогда, когда передача DMA заканчивается с использованием основного цикла.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 440 перечисляет области памяти `channel_cfg`, которые должны быть определены константами, и те области, значения которых определяются пользователем.

Таблица 440 – Channel\_cfg для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации»

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
31, 30	dst_inc	2'b10	Контроллер производит инкремент адреса пословно
29, 28	dst_size	2'b10	Контроллер осуществляет передачу пословно
27, 26	src_inc	2'b10	Контроллер производит инкремент адреса пословно
25, 24	src_size	2'b10	Контроллер осуществляет передачу пословно
17...14	R_power	4'b0010	Контроллер выполняет 4 передачи DMA
2...0	cycle_ctrl	3'b110	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
3	next_useburst	-	При установке в 1, контроллер установит chnl_useburst_set[C] в 1 после выполнения передачи с альтернативной структурой

\* Так как разряды R\_power установлены в состояние 2, необходимо задавать значение N кратное 4. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные.

Рисунок 95 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, 2R=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:

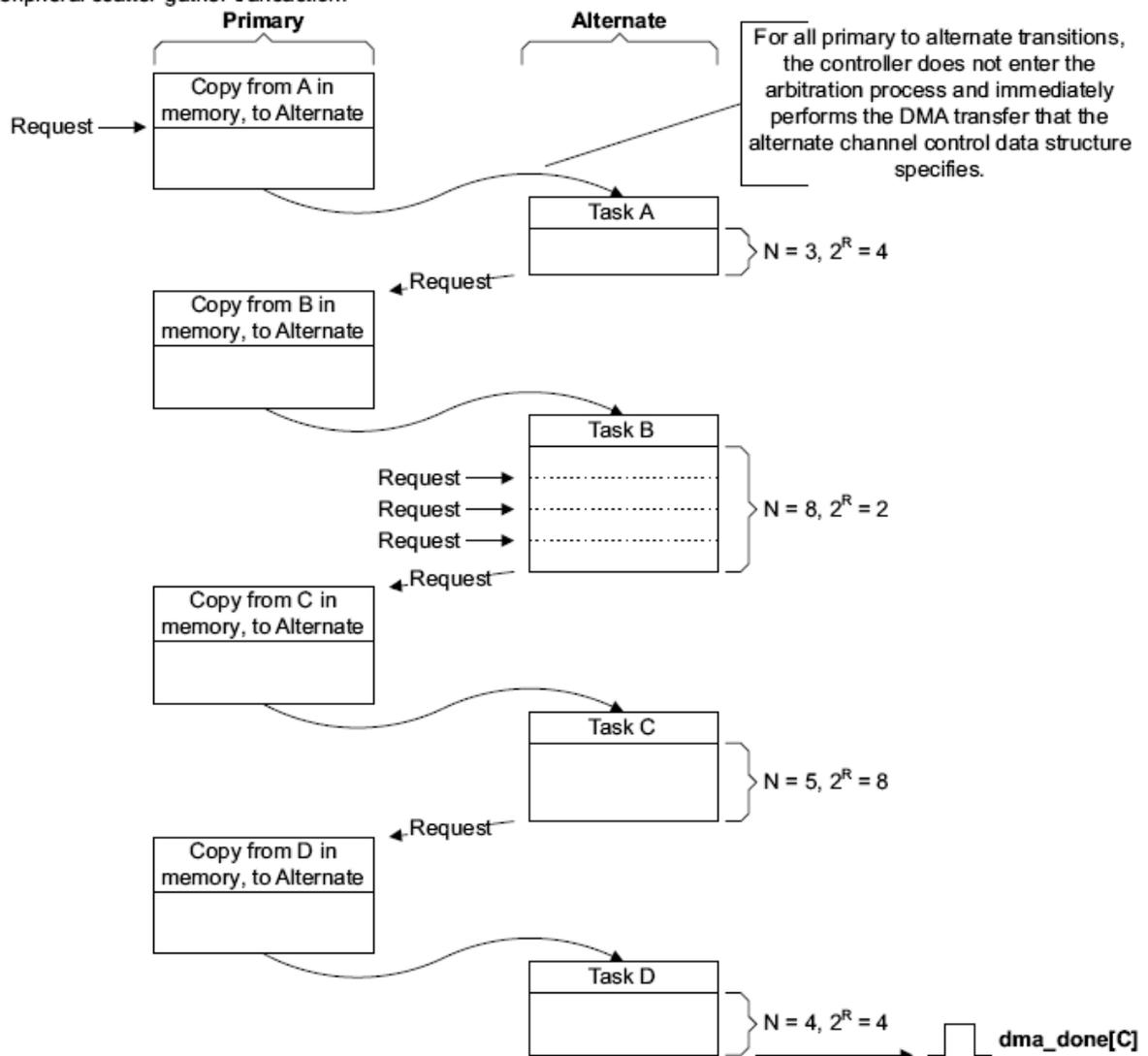


Рисунок 95 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

Инициализация:

1. Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации» путем установки `cycle_ctrl` в `3'b110`. Так как управляющие данные канала состоят из 4 слов, необходимо установить  $2^R$  в 4. В этом примере количество задач равно 4 и поэтому N установлено в 16.
2. Процессор записывает управляющие данные для шагов A, B, C, D в область памяти с адресом, указанным в `src_data_end_ptr`.
3. Процессор разрешает работу канала DMA.

Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по `dma_req[]`. Передачи выполняются следующим образом:

Первичная, копирование из области А памяти

По получению запроса на обслуживание, контроллер выполняет 4 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 4 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 3 запроса.

По окончании контроллер проводит процедуру арбитража.

Первичная, копирование из области С памяти.

Контроллер выполняет 4 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 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D.

Контроллер устанавливает `csycle_ctrl` первичных управляющих данных в `3'b000` для индикации о том, что эта структура управляющих данных является «неправильной».

Шаг D

Контроллер выполняет шаг D, используя основной цикл DMA.

Контроллер устанавливает флаг `dma_done[C]` в состояние 1 на один такт сигнала HCLK и входит в процедуру арбитража.

Индикация ошибок

При получении контроллером по шине 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:

- чтение регистра `chnl_enable_set` с целью создания списка отключенных каналов;
- если канал установил флаг `dma_done[]`, то контроллер отключает канал. Программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно установили флаги `dma_done[]`;
- процессор должен сравнить список выключенных каналов, полученный в шаге 1, с данными о каналах, которые недавно устанавливали флаги `dma_done[]`. Канал, по которому отсутствуют данные об установке флага `dma_done[]`, это и есть канал, с которым связана ошибка.

## 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.

На рисунке 96 приведена область памяти, необходимая контроллеру для структур управляющих данных канала при использовании 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 96 – 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.

В таблице 441 приведены разряды адреса, которые используются контроллером при доступе к различным элементам структуры управляющих данных, в зависимости от количества каналов, используемых в контроллере.

Таблица 441 – 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 97 демонстрирует пример реализации контроллера с использованием трех каналов DMA и альтернативной структурой управляющих данных.

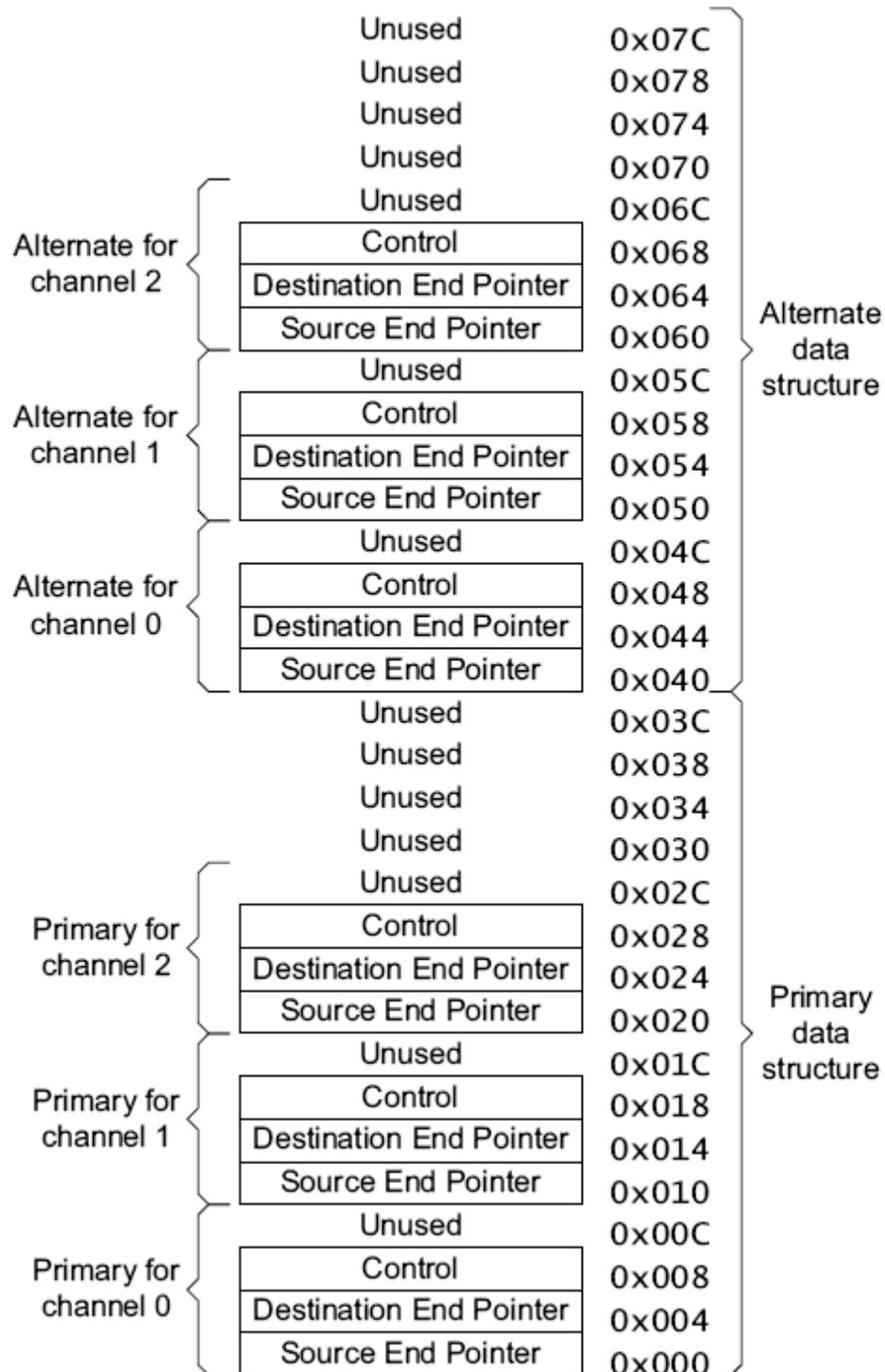


Рисунок 97 – Карта памяти для трех каналов DMA, включая альтернативную структуру управляющих данных (где Destination end pointer – указатель конца данных приемника; Source end pointer – указатель конца данных источника; Control – управление)

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.

В таблице 442 приведены разрешенные значения базового адреса для первичной структуры управляющих данных, в зависимости от количества каналов DMA, использованных в контроллере.

Таблица 442 – Разрешенные базовые адреса

Кол-во каналов DMA	Разрешенные значения базового адреса для первичной структуры управляющих данных
1	0xXXXXXXXX00, 0xXXXXXXXX20, 0xXXXXXXXX40, 0xXXXXXXXX60, 0xXXXXXXXX80, 0xXXXXXXXXA0, 0xXXXXXXXXC0, 0xXXXXXXXXE0
2	0xXXXXXXXX00, 0xXXXXXXXX40, 0xXXXXXXXX80, 0xXXXXXXXXC0
3-4	0xXXXXXXXX00, 0xXXXXXXXX80
5-8	0xXXXXXXXX000, 0xXXXXXXXX100, 0xXXXXXXXX200, 0xXXXXXXXX300, 0xXXXXXXXX400, 0xXXXXXXXX500, 0xXXXXXXXX600, 0xXXXXXXXX700, 0xXXXXXXXX800, 0xXXXXXXXX900, 0xXXXXXXXXA00, 0xXXXXXXXXB00, 0xXXXXXXXXC00, 0xXXXXXXXXD00, 0xXXXXXXXXE00, 0xXXXXXXXXF00
9-16	0xXXXXXXXX000, 0xXXXXXXXX200, 0xXXXXXXXX400, 0xXXXXXXXX600, 0xXXXXXXXX800, 0xXXXXXXXXA00, 0xXXXXXXXXC00, 0xXXXXXXXXE00
17-32	0xXXXXXXXX000, 0xXXXXXXXX400, 0xXXXXXXXX800, 0xXXXXXXXXC00

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. Значения битов этой области приведены в таблице 443.

Таблица 443 – Значения разрядов `src_data_end_ptr`

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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. Значения битов этой области приведены в таблице 444.

Таблица 444 – Значения разрядов `dst_data_end_ptr`

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	<code>dst_data_end_ptr</code>	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. Значение битов этой области приведены в таблице 445.

Таблица 445 – 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	-
	<code>dst_inc</code>	<code>dst_size</code>	<code>src_inc</code>	<code>src_size</code>	<code>dst_prot_ctrl</code>

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
	<code>Src_prot_ctrl</code>	<code>R_power</code>	<code>n_minus_1</code>	<code>next_useburst</code>	<code>cycle_ctrl</code>

Таблица 446 – Назначение разрядов `channel_cfg`

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	<code>dst_inc</code>	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 <code>dst_data_end_ptr</code> . Разрядность данных источника = полуслово: 2'b00 = зарезервировано; 2'b01 = полуслово;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		<p>2'b10 = слово;                      2'b11 = нет инкремента. Адрес остается равным значению области памяти <code>dst_data_end_ptr</code>.                      Разрядность данных источника = слово:                      2'b00 = зарезервировано;                      2'b01 = зарезервировано;                      2'b10 = слово;                      2'b11 = нет инкремента. Адрес остается равным значению области памяти <code>dst_data_end_ptr</code></p>
29, 28	<code>dst_size</code>	<p>Размерность данных приемника.                      Примечание – Значение этого поля должно быть равно значению поля <code>src_size</code></p>
27, 26	<code>src_inc</code>	<p>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 <code>src_data_end_ptr</code>.                      Разрядность данных источника = полуслово:                      2'b00 = зарезервировано;                      2'b01 = полуслово;                      2'b10 = слово;                      2'b11 = нет инкремента. Адрес остается равным значению области памяти <code>src_data_end_ptr</code>.                      Разрядность данных источника = слово:                      2'b00 = зарезервировано;                      2'b01 = зарезервировано;                      2'b10 = слово;                      2'b11 = нет инкремента. Адрес остается равным значению области памяти <code>src_data_end_ptr</code></p>
25, 24	<code>src_size</code>	<p>Задаёт размерность данных источника:                      2'b00 = байт;                      2'b01 = полуслово (16-разрядное слово);                      2'b10 = слово (32-разрядное слово);                      2'b11 = зарезервировано</p>
23...21	<code>dst_prot_ctrl</code>	<p>Задаёт состояние <code>HPROT[3:1]</code>, когда контроллер записывает данные в приемник.                      Разряд [23] управляет разрядом <code>HPROT[3]</code>:                      0 = <code>HPROT[3]</code> в состоянии 0 и доступ не кэшируется;                      1 = <code>HPROT[3]</code> в состоянии 1 и доступ кэшируется.                      Разряд [22] управляет разрядом <code>HPROT[2]</code>:                      0 = <code>HPROT[2]</code> в состоянии 0 и доступ не буферизуется;                      1 = <code>HPROT[2]</code> в состоянии 1 и доступ буферизуется.                      Разряд [21] управляет разрядом <code>HPROT[1]</code>:                      0 = <code>HPROT[1]</code> в состоянии 0 и доступ не привилегированный;</p>

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		1 = HPROT[1] в состоянии 1 и доступ привилегированный
20...18	src_prot_ctrl	<p>Задаёт состояние HPROT[3:1], когда контроллер считывает данные из источника.</p> <p>Разряд [20] управляет разрядом HPROT[3]:                      0 = HPROT[3] в состоянии 0 и доступ не кэшируется;                      1 = HPROT[3] в состоянии 1 и доступ кэшируется.</p> <p>Разряд [19] управляет разрядом HPROT[2]:                      0 = HPROT[2] в состоянии 0 и доступ не буферизуется;                      1 = HPROT[2] в состоянии 1 и доступ буферизуется.</p> <p>Разряд [18] управляет разрядом HPROT[1]:                      0 = HPROT[1] в состоянии 0 и доступ не привилегированный;                      1 = HPROT[1] в состоянии 1 и доступ привилегированный</p>
17...14	R_power	<p>Задаёт количество передач DMA до выполнения контроллером процедуры арбитража.</p> <p>Возможные значения:                      4'b0000 - арбитраж производится после каждой передачи DMA;                      4'b0001 – арбитраж производится после 2 передач DMA;                      4'b0010 – арбитраж производится после 4 передач DMA;                      4'b0011 – арбитраж производится после 8 передач DMA;                      4'b0100 – арбитраж производится после 16 передач DMA;                      4'b0101 – арбитраж производится после 32 передач DMA;                      4'b0110 – арбитраж производится после 64 передач DMA;                      4'b0111 – арбитраж производится после 128 передач DMA;                      4'b1000 – арбитраж производится после 256 передач DMA;                      4'b1001 – арбитраж производится после 512 передач DMA;                      4'b1010 – 4'b1111 – арбитраж производится после 1024 передач DMA. Это означает, что арбитраж не производится, так как максимальное количество передач DMA равно 1024</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:                      10'b0000000000 = 1 передача DMA;                      10'b0000000001 = 2 передачи DMA;                      10'b0000000010 = 3 передачи DMA;                      10'b0000000011 = 4 передачи DMA;                      10'b0000000100 = 5 передач DMA;                      10'b0000000101 = 6 передач DMA;                      ....                      10'b1111111111 = 1024 передачи DMA.</p> <p>Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить</p>

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		количество оставшихся передач DMA до завершения цикла DMA
3	next_useburst	<p>Контролирует, не установлен ли chnl_useburst_set[C] в состояние 1, если контроллер работает в режиме работы с периферией «Исполнение с изменением конфигурации», и, если контроллер завершает цикл DMA, используя альтернативные управляющие данные.</p> <p>Примечание – Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер устанавливает chnl_useburst_set[C] в значение 0, если количество оставшихся передач DMA меньше, чем 2<sup>R</sup>. Установка next_useburst разряда определяет, будет ли контроллер дополнительно переопределять разряд chnl_useburst_set[C].</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 next_useburst:</p> <p>0 – контроллер не изменяет значение chnl_useburst_set[C]. Если chnl_useburst_set[C] установлен в 0, то для всех оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер отвечает на запросы по dma_req[] и dma_sreq[], при выполнении циклов DMA он использует альтернативные управляющие данные.</p> <p>1 – контроллер изменяет значение chnl_useburst_set[C] в состояние 1. Поэтому для оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер реагирует только на запросы по dma_req[], при выполнении циклов DMA он использует альтернативные управляющие данные</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>

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		<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.

### 25.5.3.1 Вычисление адреса

Для вычисления адреса источника передачи DMA, контроллер выполняет сдвиг влево значения `n_minus_1` на количество разрядов, соответствующее полю `src_inc`, и затем вычитает получившееся значение от значения указателя адреса конца данных источника. Подобным образом вычисляется адрес передатчика передачи DMA, контроллер выполняет сдвиг влево значения `n_minus_1` на количество разрядов, соответствующее полю `dst_inc`, и затем вычитает получившееся значение от значения указателя адреса конца данных приемника.

В зависимости от значения полей `src_inc` и `dst_inc` вычисления адресов приемника и источника выполняются по следующим уравнениям:

`src_inc=b00` and `dst_inc=b00`

- адрес источника = `src_data_end_ptr - n_minus_1`
- адрес приемника = `dst_data_end_ptr - n_minus_1`.

`src_inc=b01` and `dst_inc=b01`

- адрес источника = `src_data_end_ptr - (n_minus_1<<1)`
- адрес приемника = `dst_data_end_ptr - (n_minus_1<<1)`.

`src_inc=b01` and `dst_inc=b10`

- адрес источника = `src_data_end_ptr - (n_minus_1<<2)`
- адрес приемника = `dst_data_end_ptr - (n_minus_1<<2)`.

`src_inc=b11` and `dst_inc=b11`

- адрес источника = `src_data_end_ptr`
- адрес приемника = `dst_data_end_ptr`.

В таблице 447 приведены адреса приемника цикла DMA для 6 слов.

Таблица 447 – Цикла DMA для 6 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>				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения <code>channel_cfg</code> после цикла DMA				
<code>src_size=2'b10, dst_inc=2'b10, n_minus_1=0, cycle_ctrl=0</code>				
* Значение, полученное после сдвига влево значения счетчика на количество разрядов соответствующее <code>dst_inc</code>				

В таблице 448 приведены адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.

Таблица 448 – Цикла DMA для 12 байт с «полусловным» инкрементом

Начальные значения channel_cfg перед циклом DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=4'b1011, cycle_ctrl=1, R_power=2'b11				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x5E7	11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3
	0x5E7	9	0x12	0x5D5
	0x 5E7	8	0x10	0x5D7
	0x 5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
	0x5E7	4	0x8	0x5DF
Значения channel_cfg после 2 <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				
DMA передачи	0x 5E7	3	0x6	0x5E1
	0x 5E7	2	0x4	0x5E3
	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7
Конечные значения channel_cfg после цикла DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=0, cycle_ctrl=0**, R_power=2'b11				
* Значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc. ** После окончания цикла DMA контроллер делает channel_cfg «неправильным», сбрасывая в 0 поле cycle_ctrl				

## 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:

- нужно избегать адресации при доступе к зарезервированным или неиспользованным адресам, так как это может привести к непредсказуемым результатам;
- необходимо заполнять неиспользуемые или зарезервированные разряды регистров нулями при записи и игнорировать значения таких разрядов при считывании, кроме случаев, специально описанных в разделе;

- системный сброс или сброс по установке питания сбрасывает все регистры в состояние 0, кроме случаев, специально описанных в разделе;
- все регистры поддерживают доступ по чтению и записи, кроме случаев, специально описанных в разделе. Доступ по записи обновляет содержание регистра, а доступ по чтению возвращает содержимое регистра.

Таблица 449 – 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 450 – 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 451 – 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
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 452 – 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 453 – 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	<p>Определяет уровни индикации сигналов HPROT[3:1] защиты шины АНВ-Lite:</p> <p>Разряд 7 управляет сигналом HPROT[3], с целью индикации о появлении доступа с кэшированием;</p> <p>Разряд 6 управляет сигналом HPROT[2], с целью индикации о появлении доступа с буферизацией;</p> <p>Разряд 5 управляет сигналом HPROT[1], с целью индикации о появлении привилегированного доступа.</p> <p>Примечание – Если разряд[n] = 1, то соответствующий сигнал HPROT в состоянии 1. Если разряд[n] = 0, то соответствующий сигнал HPROT в состоянии 0</p>
4...1	-	Не определено. Следует записывать 0
0	master_enable	<p>Определяет состояние контроллера:</p> <p>0 – запрещает работу контроллера;</p> <p>1 – разрешает работу контроллера</p>

### 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, используемых 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 454 – Регистр базового адреса управляющих данных каналов

Номер	31...10	9...0
Доступ	R/W	U
Сброс	0	0
	ctrl_base_ptr	-

Таблица 455 – 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 456 – Регистр базового адреса альтернативных управляющих данных каналов

Номер	31... 0
Доступ	RO
Сброс	0
	alt_ctrl_base_ptr

Таблица 457 – 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 458 – 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 459 – 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 460 – 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 461 – 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; 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 462 – 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 463 – 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<sup>R</sup> передач, в том случае, если число оставшихся передач (N) меньше чем 2<sup>R</sup>, 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<sup>R</sup>, 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 464 – Регистр сброса пакетного обмена каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	<code>chnl_useburst_clr</code> for channel [31]	.....	<code>chnl_useburst_clr</code> for channel [2]	<code>chnl_useburst_clr</code> for channel [1]	<code>chnl_useburst_clr</code> for channel [0]

Таблица 465 – 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>	<p>Установка соответствующего разряда разрешает обработку запросов на выполнение циклов DMA от <code>dma_sreq[]</code>.</p> <p>При записи:</p> <p>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>;</p> <p>Разряд <code>[C] = 1</code> разрешает обрабатывать запросы на выполнение циклов DMA, полученные от <code>dma_sreq[]</code>.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p>

### 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 466 – 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 467 – 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 468 – 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 469 – 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	Установка соответствующего разряда разрешает установку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_req_mask_set регистр для отключения установки запросов; Разряд [C] = 1 разрешает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

### 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 470 – 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 471 – 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 472 – 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 473 – 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 C.</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:                      - при завершении цикла DMA;                      - при чтении из channel_cfg с полем cycle_ctrl установленным в 3'b000;                      - при появлении ошибки на шине AHB-Lite</p>

**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 474 – 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 475 – 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.                      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:</p> <ul style="list-style-type: none"> <li>- 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»;</li> <li>- при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «Пинг-понг»;</li> <li>- при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах:                             <ul style="list-style-type: none"> <li>- «пинг-понг»;</li> <li>- работа с памятью «Исполнение с изменением конфигурации»;</li> <li>- работа с периферией «Исполнение с изменением конфигурации»;</li> </ul> </li> </ul>

**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 476 – 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 477 – 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.                      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:</p> <ul style="list-style-type: none"> <li>- 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»;</li> <li>- при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «пинг-понг»;</li> <li>- при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах:                             <ul style="list-style-type: none"> <li>- «пинг-понг»</li> <li>- работа с памятью «Исполнение с изменением конфигурации»</li> <li>- работа с периферией «Исполнение с изменением конфигурации»</li> </ul> </li> </ul>

### 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 478 – 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 479 – 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 480 – Регистр сброса приоритета каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_priorit_clr for channel [31]	.....	chnl_priority_clr for channel [2]	chnl_priority_clr for channel [1]	chnl_priority_clr for channel [0]

Таблица 481 – Назначение разрядов регистра `chnl_priority_clr`

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	<code>chnl_priority_clr</code>	Установка разряда присваивает соответствующему каналу DMA уровень приоритета по умолчанию. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать <code>chnl_priority_set</code> регистр для установки каналу C высокого уровня приоритета. Разряд [C] = 1 устанавливает каналу DMA C уровень приоритета по умолчанию. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

### 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 482 – Регистр сброса флага ошибки

Номер	31...1	0
Доступ	U	R/W
Сброс	0	0
	-	<code>err_clr</code>

Таблица 483 – 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	<code>err_clr</code>	Установка сигнала в состояние 0, чтение возвращает состояние сигнала (флага) <code>dma_err</code> . При чтении: Разряд [C] = 0 означает, что <code>dma_err</code> находится в состоянии 0; Разряд [C] = 1 означает, что <code>dma_err</code> находится в состоянии 1. При записи: Разряд [C] = 0 не дает эффекта. Состояние <code>dma_err</code> останется неизменным; Разряд [C] = 1 сбрасывает сигнал (флаг) <code>dma_err</code> в состояние 0.  Примечание – При сбросе сигнала <code>dma_err</code> одновременно с появлением ошибки на шине АНВ-Lite, приоритет отдается ошибке, и, следовательно, значение регистра (и <code>dma_err</code> ) останется неизменным (несброшенным)

## 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 Interruptor, CLINT). CLINT содержит регистры для управления программными прерываниями и прерываниями таймера. Карта регистров CLINT приведена в таблице 484

Таблица 484 – 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

Таблица 485 – Регистр запроса/снятия прерывания

Биты	Название	Доступ	Описание	Значение
0	msip	R/W	Ожидание программного прерывания	0

Таблица 486 – Регистр значения таймера

Биты	Название	Доступ	Описание	Значение
31...0	mtime_lo	R	Младшая часть значения таймера	0
64...32	mtime_hi	R	Старшая часть значения таймера	0

Регистр содержит 64-битное значение таймера. Значение увеличивается по сигналу timer\_pulse (передний фронт частоты HCLK деленной на значение, установленное в регистре DIV\_SYS\_TIM) внешнего интерфейса процессорного комплекса.

Таблица 487 – Регистр сравнения таймера

Биты	Название	Доступ	Описание	Значение
31...0	mtimespr_lo	R	Младшая часть значения регистра сравнения	0
64...32	mtimespr_hi	R	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.

## 27 Контроллер обработки локальных прерываний CLIC

Контроллер обработки локальных прерываний (контроллер CLIC) получает сигналы прерывания и передаёт ядру прерывание, подлежащее обработке. Контроллер CLIC поддерживает до 4096 входов прерывания в ядро. Каждый вход прерывания  $i$  имеет четыре 8-разрядных регистра управления, отображенных в памяти: бит ожидания прерывания  $clicintip[i]$ , бит разрешения прерывания  $clicintie[i]$ , атрибуты прерывания  $clicintattr[i]$  для указания режима привилегированности и типа триггера, и биты управления прерыванием для указания уровня и приоритета  $clicintctl[i]$ . Первые 16 входов прерывания зарезервированы для прерываний базового режима, присутствующих в младших 16 битах регистров  $mp$  ( $uip$ ) и  $mie$  ( $uie$ ), таким образом, может быть добавлено до 4080 локальных внешних прерываний.

### 27.1 Приоритет прерывания

Контроллер CLIC расширяет функцию приоритета прерываний, поддерживая до 256 уровней прерываний для каждого режима привилегированности, где уровни прерываний с более высоким номером могут вытеснять уровни прерываний с более низким номером. Уровень прерывания 0 соответствует обычному выполнению вне обработчика прерываний. Уровни 1 – 255 соответствуют уровням обработчика прерываний. Входящие прерывания с более высоким уровнем прерывания могут вытеснить активный обработчик прерываний, работающий с более низким уровнем в том же режиме привилегированности, при условии, что прерывания глобально разрешены в этом режиме привилегированности. Аналогично поведение прерываний, при котором входящие прерывания для режима с более высокими привилегиями могут вытеснять активный обработчик прерываний, запущенный в режиме с более низкими привилегиями, независимо от разрешения глобального прерывания в режиме с более низкими привилегиями.

### 27.2 Взаимодействие контроллера CLIC с другими локальными прерываниями

Контроллер CLIC включает в себя функциональность базовых локальных прерываний, представленных в 16 битах  $mp/mie$  ( $uip/uie$ ) и выше, которые не отображаются в  $mp/mie$  ( $uip/uie$ ). Прерывания таймера ( $mtip/utip$ ), программные прерывания ( $msip/usip$ ) и входы внешних прерываний ( $meip/ueip$ ) рассматриваются как дополнительные локальные источники прерываний, где режим привилегированности, уровень прерывания и приоритет могут быть изменены с помощью отображаемых в памяти регистров  $clicintattr[i]$  и  $clicintctl[i]$ . В режиме CLIC делегирование прерывания для этих сигналов достигается путем изменения режима привилегированности прерывания в регистре атрибутов прерывания CLIC  $clicintattr$ , как и для любого другого входа прерываний CLIC.

### 27.3 Описание регистров контроллера CLIC

Данные о регистрах контроллера CLIC приведены в таблице 488.

Таблица 488 – Регистры контроллера CLIC

Адрес	Название	Доступ	Описание
0x0D00_0000	CLIC		Контроллер CLIC
Смещение			
0x0000	cliccfg	R/W	Регистр управления
0x0004	clicinfo	R/O	Регистр информации
0x0040	clicintrig[0]	R/O	Регистр управления триггером 0
0x0044	clicintrig[1]	R/O	Регистр управления триггером 1
0x0048	clicintrig[2]	R/O	Регистр управления триггером 2
...	...	...	...
0x00B4	clicintrig[29]	R/O	Регистр управления триггером 29
0x00B8	clicintrig[30]	R/O	Регистр управления триггером 30
0x00BC	clicintrig[31]	R/O	Регистр управления триггером 31
0x1000 + 4 * i	clicintip[i]	R/W	Регистр наличия прерывания
0x1001 + 4 * i	clicintie[i]	R/W	Регистр включения прерывания
0x1002 + 4 * i	clicintattr[i]	R/W	Регистр настройки прерывания
0x1003 + 4 * i	clicintctl[i]	R/W	Регистр управления уровнем прерывания
0x4FFC	clicintip[4095]	R/W	Регистр наличия прерывания
0x4FFD	clicintie[4095]	R/W	Регистр включения прерывания
0x4FFE	clicintattr[4095]	R/W	Регистр настройки прерывания
0x4FFF	clicintctl[4095]	R/W	Регистр управления уровнем прерывания

Доступ к регистрам cliccfg, clicinfo и clicintrig возможен только из режима привилегированности Machine. При доступе к данным регистрам из других режимов привилегированности будет сгенерировано исключение. Доступ к регистрам clicintip, clicintie, clicintattr и clicintctl возможен из режима привилегированности machine. для доступа из других режимов привилегированности необходимо настроить соответствующее прерывание с помощью регистров clicintattr.

Триггеры прерываний не реализованы и соответствующие регистры clicintrig[i] подключены к логическому «0».

#### 27.3.1 Конфигурационный регистр cliccfg

Единый 8-разрядный регистр глобальной конфигурации cliccfg, отображаемый в памяти, который определяет, сколько поддерживается режимов привилегированности, как регистры clicintctl[i] подразделяются на поля уровня и приоритета, а также поддерживается ли выборочная аппаратная векторизация. Регистр cliccfg имеет три поля, 2-битное поле nmbits, 4-битное поле nlbits и 1-битное поле nvbits, плюс зарезервированный бит WPRI, аппаратно подключен к логическому «0».

WPRI означает «Запись сохраняет значения, чтение игнорирует значения», указывая на полное чтение/запись полей, зарезервированных для будущих применений. Программное обеспечение должно игнорировать значения, считываемые из этих полей, и должно сохранять значения, хранящиеся в этих полях, при записи значений в другие поля того же регистра.

Таблица 489 – Регистр cliccfg

Номер	7	6	5	4..1	0
Доступ	U	U	R/W	R/W	RO
Сброс	0	0	0	XXXX	1
		nmbits[1]	nmbits[0]	nlbits[3:0]	nvbits

### 27.3.1.1 Определение режима привилегированности прерывания

Двухразрядное поле cliccfg.nmbits указывает, сколько бит физически реализовано в clicintattr[i].mode для представления режима привилегированности входа i. Хотя поле cliccfg.nmbits всегда имеет ширину в 2 бита, физически реализованных бит в этом поле может быть меньше двух (в зависимости от того, сколько режимов привилегированности прерывания поддерживается).

Например, в системах, поддерживающих только M-режим не нужен дополнительный бит для представления поддерживаемых режимов привилегированности. В этом случае никакие физически реализованные биты в clicintattr.mode не требуются и, таким образом, cliccfg.nmbits = 0, т.е. cliccfg.nmbits может быть аппаратно подключен к логическому «0».

В системах с M/U-режимами и поддержкой пользовательских прерываний с расширением N, cliccfg.nmbits может быть установлено равным 0 или 1. Если cliccfg.nmbits = 0, то все прерывания обрабатываются как прерывания в M-режиме. Если cliccfg.nmbits = 1, то значение 1 в старшем бите MSB регистра clicintattr[i].mode указывает, что ввод прерывания выполняется в M-режиме, в то время как значение 0 указывает, что прерывание выполняется в U-режиме.

Аналогично, в системах, которые поддерживают M/S/U-режимы прерываний, cliccfg.nmbits может быть установлен равным 0, 1 или 2 для представления режимов привилегированности. Значение cliccfg.nmbits = 0 указывает, что все локальные прерывания обрабатываются в M-режиме. Значение cliccfg.nmbits = 1 указывает, что MSB выбирает между M-режимом (1) и S-режимом (0). Значение cliccfg.nmbits = 2 указывает, что два бита MSB каждого регистра clicintattr[i].mode кодируют режим привилегированности прерывания, используя ту же кодировку, что и поле mstatus.mpp.

Неизолированный S-режим (без MMU, satp=0) может использоваться в микроконтроллерах для обеспечения аппаратного делегирования прерываний вне M-режима. Неизолированный S-режим уже одобрен как часть привилегированной архитектуры. Также есть предложения по добавлению S-режима с поддержкой PMP, чтобы разрешить запуск RTOS, работающей в S-режиме, изолируя себя от задач, выполняемых в U-режиме.

Таблица 490 – Расшифровка уровней привилегированности (mstatus.mpp)

Уровень	Кодирование	Название	Аббревиатура
0	00	User/Application	U
1	01	Supervisor	S
2	10	Зарезервировано	–
3	11	Machine	M

Таблица 491 – Режимы привилегированности

Режим привилегированности	nmbits	clicintattr[i].mode	Интерпретация
M	0	xx	M-режим прерывание
M/U	0	xx	M-режим прерывание
M/U	1	0x	U-режим прерывание
M/U	1	1x	M-режим прерывание
M/S/U	0	xx	M-режим прерывание
M/S/U	1	0x	S-режим прерывание
M/S/U	1	1x	M-режим прерывание
M/S/U	2	00	U-режим прерывание
M/S/U	2	01	S-режим прерывание
M/S/U	2	10	Зарезервировано
M/S/U	2	11	M-режим прерывание
M/S/U	3	xx	Зарезервировано

### 27.3.1.2 Определение уровня прерывания

Четырёхразрядное поле cliccfg.nlbits указывает, сколько старших бит в clicintctl[i] выделено для кодирования уровня прерывания. Важно инициализировать это поле из программного обеспечения. В настоящее время для уровня поддерживается только 0 или 8 бит, остальные значения зарезервированы. По факту, управляющие биты переключаются для использования только для уровня или только для приоритета.

Хотя уровень прерывания представляет собой 8-разрядное целое число без знака, количество фактически назначенных или реализованных бит может быть меньше 8. Как описано выше, количество присвоенных бит указано в cliccfg.nlbits. Количество фактически реализованных битов может быть получено из cliccfg.nlbits и фиксированного параметра clicinfo. CLICINTCTLBITS (со значением от 0 до 8) определяет биты, реализованные как для уровня прерывания, так и для приоритета. Количество доступных бит уровня может быть определено путем вычитания количества бит режима из CLICINTCTLBITS.

Если фактически назначенных или реализованных бит меньше 8, то эти биты выравниваются по левому краю и заполняются единицами для младших пропущенных бит. Например, если nlbits > CLICINTCTLBITS, то предполагается, что младшие биты 8-разрядного уровня прерывания равны всем единицам. Аналогично, если nlbits < 8, то предполагается, что младшие биты 8-битного уровня прерывания равны всем единицам. В таблице 492 приведено, как кодируются уровни для этих случаев.

Таблица 492 – Режимы привилегированности

Биты	Кодирование	Уровни прерываний
0	.....	255
1	I.....	127, 255
2	II.....	63, 127, 191, 255
3	III....	31, 63, 95, 127, 159, 191, 223, 255
4	IIII....	15, 31, 47, 63, 79, 95, 111, 127, 143, 159, 175, 191, 207, 223, 239, 255

Примечание – Обозначения в графе Кодирование:  
 «I» – изменяемые биты доступные для задания уровня;  
 «.» – несуществующие биты для кодирования уровня, предполагается установлены в «1»

Если  $nlbits = 0$ , то все прерывания рассматриваются с уровнем 255.

Таблица 493 – Пример настройки cliccfg

CLICINTCTLBITS	nlbits	clicintctl[i]	Уровень прерывания
0	2	.....	255
1	2	I.....	127, 255
2	2	II.....	63,127,191,255
3	3	III....	31,63,95,127,159,191,223,255
4	1	Ippp....	127,255

Примечание – Обозначения в графе clicintctl[i]:  
 «.» несуществующие биты для кодирования уровня, предполагается установлены в «1»;  
 «I» изменяемые биты, доступные для задания уровня;  
 «p» изменяемые биты, доступные для задания приоритета

### 27.3.1.3 Определение приоритета прерывания

Наименее значимые биты в  $clicintctl[i]$ , которые не настроены как часть уровня прерывания, являются приоритетом прерывания, которые используются для определения приоритета среди ожидающих и разрешённых прерываний в том же режиме привилегированности и на том же уровне прерывания. Прерывание с наивысшим приоритетом при заданном режиме привилегированности и уровне прерывания выполняется первым. В случае наличия нескольких ожидающих и разрешенных прерываний с одинаковым наивысшим приоритетом, прерывание с наибольшим номером выполняется первым. Прерывание с наибольшим номером не имеет приоритета, когда режим привилегированности, уровень и приоритет идентичны.

Стоит обратить внимание, что 8-битный уровень прерывания используется для определения вытеснения (для вложенных прерываний). В отличии от случая, когда 8-разрядный приоритет прерывания не влияет на вытеснение, а используется только в качестве тай-брейка при наличии нескольких ожидающих прерываний с одинаковым уровнем прерывания.

Любые реализованные биты приоритета рассматриваются как наиболее значимые биты 8-разрядного целого числа без знака с младшими нереализованными битами, установленными в «1». Например, с одним битом приоритета ( $p111\_1111$ ) прерывания могут быть установлены так, чтобы они имели приоритеты 127 или 255, и с двумя битами приоритета ( $pp11\_1111$ ) прерывания могут быть установлены с приоритетами 63, 127, 191 или 255.

### 27.3.1.4 Определение аппаратной векторизации выборочных прерываний

Поле `nvbits`, доступное только для чтения в `cliccfg`, указывает, реализована ли функция аппаратной векторизации выборочных прерываний или нет. Функция выборочной аппаратной векторизации предоставляет пользователям гибкость в выборе поведения для каждого прерывания: либо аппаратная векторизация, либо без нее. В результате это позволяет пользователям оптимизировать каждое прерывание и пользоваться преимуществами обоих вариантов поведения. Более точно, аппаратная векторизация имеет преимущество в более быстрой реакции на прерывания за счёт небольшого увеличения размера кода (для сохранения/восстановления контекстов). С другой стороны, не векторная имеет преимущество в меньшем размере кода (за счет совместного использования и повторного использования одной копии общего кода для сохранения/ восстановления контекстов) ценой более медленного отклика на прерывание.

Когда `nvbits` равен 0, аппаратная векторизация выборочных прерываний не реализована. В этом случае все прерывания не векторные и направляются в общий код регистра `mtvec` (`utvec`).

Когда `nvbits` равно 1, реализуется аппаратная векторизация выборочных прерываний. Бит `clicintattr[i].shv` управляет поведением векторизации прерывания `i`. Если `clicintattr[i].shv` равен 0, то прерывание не является векторным и всегда переходит к общему коду `mtvec` (`utvec`). Если `clicintattr[i].shv` равен 1, то прерывание аппаратно преобразуется в указатель функции обработчика прерывания, указанный в `mtvt` (`utvt`) CSR. Это позволяет некоторым прерываниям переходить к общему базовому адресу, хранящемуся в `mtvec` (`utvec`), в то время как другие обрабатываются аппаратно через таблицу, на которую указывает дополнительный `mtvt` (`utvt`) CSR.

### 27.3.2 Информационный регистр `clicinfo`

Регистр `clicinfo` доступен только для чтения и отображает отладочную информацию.

Таблица 494 – Регистр `clicinfo`

Номер	31	30..25	24..21	20..13	12..0
Доступ	U	R/O	R/O	R/O	RO
Сброс	0	000000	0100	00000000	00110000
		<code>num_trigger</code>	<code>CLICINTCTLBITS</code>	<code>version</code>	<code>num_interrupt</code>

Поле `num_interrupt` указывает фактическое количество максимальных входных сигналов прерывания, поддерживаемых в данной реализации.

В поле `version` указывается версия реализации CLIC. Старшие четыре разряда указывает версию архитектуры, а младшие четыре разряда указывает версию реализации.

Поле `CLICINTCTLBITS` указывает, сколько аппаратных бит фактически реализовано в регистрах `clicintctl`, при этом  $0 \leq \text{CLICINTCTLBITS} \leq 8$ . Реализованные биты хранятся выровненными по левому краю в старших битах каждого 8-разрядного регистра `clicintctl[i]`, при этом младшие нереализованные биты аппаратно установлены в логическую «1».

Поле `num_trigger` указывает максимальное количество триггеров прерывания, поддерживаемых в данной реализации. Допустимые значения от 0 до 32.

### 27.3.3 Регистр ожидания прерывания `clcintip`

Каждый входной сигнал прерывания имеет выделенный бит ожидания прерывания `clcintip[i]` и для удобства доступа занимает один байт в карте памяти. Ожидающий бит расположен в бите 0 байта. Программное обеспечение должно рассматривать значение `clcintip[i] = 0` как отсутствие ожидания прерывания, а `clcintip[i] != 0` как ожидание прерывания, чтобы учесть возможное расширение поля `clcintip`.

Когда вход сконфигурирован как `level-sensitive`, бит `clcintip[i]` отражает значение входного сигнала в контроллер прерывания после операции условной инверсии, указанной в поле `clcintattr[i]`, а программные записи в бит игнорируются. Программное обеспечение очищает прерывание на устройстве источнике.

Когда вход сконфигурирован как `edge-sensitive`, `clcintip[i]` представляет собой регистр чтения-записи, который может обновляться как с помощью аппаратных прерываний, так и с помощью программного обеспечения. Бит устанавливается аппаратно после того, как на входе прерывания наблюдается фронт соответствующей полярности, как определено полем `clcintattr[i]`. Соответствующий бит ожидания прерывания очищается аппаратно, когда прерывание обслуживается в векторном режиме. Смотрите дополнительные сведения об аппаратной очистке в описании регистра `mtvec` (`utvec`). Программные записи могут устанавливать или очищать `edge-triggered` биты ожидания прерывания непосредственно путем записи в регистр, отображенный в память. `Edge-triggered` биты ожидания прерывания, также могут быть очищены с помощью команды записи в CSR регистр `mnxti` (`unxti`).

Для повышения производительности, когда выбрано векторное прерывание и обрабатывается, аппаратно очищается соответствующий `edge-triggered` бит ожидания, поэтому программному обеспечению не нужно очищать бит ожидания в процедуре обработки.

Напротив, когда выбрано не векторное прерывание (общий код), аппаратно не будет очищаться `edge-triggered` бит ожидания. Ожидается, что в программном обеспечении будет использована инструкция записи CSR в регистр `mnxti` (`unxti`), для очистки `edge-triggered` бита ожидания в не векторном режиме. Дополнительная информация об этом в описании регистра `mnxti` (`unxti`).

Значение в `clcintip[i]` не определено при переключении из режима `level-sensitive` в режим `edge-triggered` в `clcintattr[i]`. Программное обеспечение не может полагаться на сохранённое состояние в битах регистра `clcintip[i]` режима `edge-triggered` при переходе в режим `level-sensitive`.

### 27.3.4 Регистр разрешения прерываний `clcintie`

Каждый вход прерывания имеет выделенный бит разрешения прерывания `clcintie[i]` и для удобства доступа занимает один байт в карте памяти. Этот управляющий

бит доступен по чтению-записи и используется для включения/выключения соответствующего прерывания. Бит включения находится в бите 0 байта. Программное обеспечение должно рассматривать  $clcintie[i]=0$  как прерывание выключено, а  $clcintie[i]!=0$  как прерывание включено для учета возможного будущего расширения поля  $clcintie$ .

Бит  $clcintie[i]$  – это индивидуальный бит включения, в то время как  $mstatus.xie$  ( $ustatus.xie$ ) – глобальный бит включения для текущего режима привилегированности. Следовательно, для включения прерывания  $i$  в текущем режиме привилегированности должны быть установлены как  $clcintie[i]$ , так и  $mstatus.xie$  ( $ustatus.xie$ ).

Поскольку  $mstatus.xie$  ( $ustatus.xie$ ) имеет силу только в текущем режиме привилегированности в соответствии с соглашением RISC-V, прерывание  $i$  из режима с более высокими привилегиями включено до тех пор, пока установлен  $clcintie[i]$  (независимо от настройки  $mstatus.xie$  ( $ustatus.xie$ ) в режимах с более высокими привилегиями).

### 27.3.5 Регистр атрибутов прерывания $clcintattr$

Регистр  $clcintattr$  – 8-разрядный доступный по чтению/записи регистр, определяющий различные атрибуты для каждого прерывания.

Таблица 495 – Регистр  $clcintattr$

Номер	7..6	5..3	2..1	0
Доступ	R/W	U	R/W	R/W
Сброс	00	000	00	1
	mode		trig	shv

Одноразрядное поле  $shv$  используется для выбора векторного режима прерывания. Если  $shv = 0$ , то прерывание не векторное и, таким образом, переходит к общему коду в  $mtvec$  ( $utvec$ ). Если  $shv = 1$ , то выбирается векторный режим для прерывания и, таким образом, автоматически переходит к указателю функции обработчика прерываний, указанному в  $mtvt$  ( $utvt$ ) CSR. Эта опция позволяет некоторым прерываниям переходить к общему базовому адресу, хранящемуся в  $mtvec$  ( $utvec$ ), в то время как другие прерывания обрабатываются аппаратно через таблицу, на которую указывает дополнительный  $mtvt$  ( $utvt$ ) CSR.

Если  $clccfg.nvbits = 0$ , функция векторизации для выборочных прерываний недоступна и поле  $shv$  аппаратно подключено к логическому «0».

Двухразрядное поле  $trig$  определяет тип триггера и полярность для каждого входного сигнала прерывания. Бит 0 –  $trig[0]$ , определяет уровень или фронт (0: level-triggered, 1: edge-triggered); в то время как бит 1 –  $trig[1]$ , определяет полярность фронта (0: положительный фронт, 1: отрицательный фронт). Могут быть четыре возможные комбинации: срабатывание по положительному уровню, срабатывание по отрицательному уровню, срабатывание по положительному фронту и срабатывание по отрицательному фронту.

В некоторых реализациях могут быть доступны только определенные типы поддерживаемых триггеров. В этом случае эти биты аппаратно доопределены к фиксированным значениям.

Двухразрядное поле `mode` указывает в каком режиме привилегированности работает это прерывание. Это поле использует ту же кодировку, что и `mstatus.mpp` (11: M-режим, 01: S-режим, 00: U-режим). Допустимую длину этого поля можно запрограммировать с помощью `cliccfg.nmbits`.

В целях безопасности поле `mode` может быть установлено только в такой уровень привилегированности, который равен или ниже текущего уровня привилегированности.

### 27.3.6 Регистр управления входом прерывания `clicintctl`

Регистр `clicintctl[i]` – это 8-разрядный регистр управления доступный по чтению/записи, отображаемый в память, для указания уровня прерывания и приоритета прерывания. Количество бит, фактически реализованных в этом регистре, задается фиксированным параметром `CLICINTCTLBITS` (в `clinfo`), который имеет значение от 0 до 8. Реализованные биты сохраняются выровненными по левому краю в старших битах каждого восьмиразрядного регистра `clicintctl[i]`, при этом младшие нереализованные биты подключаются к логической единице. Эти управляющие биты интерпретируются как уровень и приоритет в соответствии с настройками в регистре конфигурации `CLIC cliccfg.nlbits`.

Чтобы выбрать прерывание для перенаправления ядру, аппаратный `CLIC` объединяет валидные биты из `clicintattr.mode` и `clicintctl` для формирования целого числа без знака, затем выбирает глобальный максимум для всех ожидающих и разрешенных прерываний на основе этого значения. Далее, настройка `cliccfg` определяет, как разделить значение `clicintctl` на уровень прерывания и приоритет прерывания. Наконец, уровень выбранного прерывания сравнивается с пороговым значением соответствующего режима привилегированности, чтобы определить, является ли оно разрешённым или замаскированным с помощью порога (прерывание не обрабатывается).

Выбор прерывания в режиме с высокими привилегиями маскирует любое прерывание в режиме с более низкими привилегиями, поскольку режим с более высокими привилегиями указывает, что сигнал прерывания более неотложный, чем любое прерывание в режиме с более низкими привилегиями.

#### 27.3.6.1 Идентификационный номер ввода прерывания

4096-и векторам прерываний `CLIC` присваиваются уникальные идентификационные номера со значениями кода исключения `mcause` (`ucause`) (`exccode`). Для поддержания обратной совместимости базовые прерывания сохраняют свои первоначальные значения, в то время как новые прерывания нумеруются, начиная с 16.

### 27.3.7 Регистр триггера прерывания `clicintrig`

Необязательные триггеры прерывания `clicintrig[i]` используются для генерации исключения точки останова в режиме отладки или режиме отслеживания. Фактическое количество поддерживаемых триггеров указано в `clinfo.num_trigger`.

Каждый триггер прерывания представляет собой 32-разрядный регистр доступный по чтению/записи, отображённый в память.

Таблица 496 – Регистр clicintrig

Номер	31	30..13	12..0
Доступ	R/O	U	R/O
Сброс	0	0	0
	enable		interrupt_number

Поле interrupt\_number выбирает, какой номер входного сигнала прерывания используется в качестве источника для триггера прерывания.

Управляющий бит enable используется для включения/отключения триггера прерывания.

Например, триггер срабатывает только в том случае, если прерывание действительно принято, а не когда прерывание замаскировано или не принято. Кроме того, запрошенное действие, например, точка останова или трассировка, выполняется непосредственно перед выполнением первой команды обработчика прерывания.

## 27.4 Контрольно-статусные регистры CSR

В данном подразделе приведено описание контрольно-статусных регистров CSR, участвующих в настройке прерываний.

Список реализованных контрольно-статусных регистров, которые доступны из режима привилегированности Machine, приведен в таблице 513.

Таблица 497 – Контрольно-статусные регистры CSR (Machine)

Адрес	Название	Доступ	Описание
0x307	mtvt	R/W	Адрес таблицы с адресами обработчиков векторных прерываний
0x308	mclicbase	R/O	Базовый адрес контроллера локальных прерываний
0x345	mnxti	R/W	Адрес следующего обработчика прерываний и управление включением прерываний
0x346	mintstatus	R/O	Статус прерывания
0x347	mintthresh	R/W	Управление порогом уровня прерываний
0x348	mscratchsw	R/W	Scratch регистр с проверкой режима привилегированности
0x349	mscratchswl	R/W	Scratch регистр с проверкой уровня прерывания

Список реализованных контрольно-статусных регистров, которые доступны из режима привилегированности User, приведен в таблице 498.

Таблица 498 – Контрольно-статусные регистры CSR (User)

Адрес	Название	Доступ	Описание
0x007	utvt	R/W	Адрес таблицы с адресами обработчиков векторных прерываний
0x045	unxti	R/W	Адрес следующего обработчика прерываний и управление включением прерываний
0x046	uintstatus	R/O	Статус прерывания
0x047	uintthresh	R/W	Управление порогом уровня прерываний
0x048	uscratchcsw	R/W	Scratch регистр с проверкой режима привилегированности
0x049	uscratchcswl	R/W	Scratch регистр с проверкой уровня привилегированности

#### 27.4.1 Регистры mtvt, utvt

32-разрядный регистр доступный по чтению записи содержит базовый адрес таблицы векторов прерываний, выровненный по границе в 64 байта или больше степени двойки. Фактическое выравнивание может быть определено путем записи единиц в младшие биты, а затем считывания их обратно. Значения, отличные от 0 в младших шести битах зарезервированы.

#### 27.4.2 Регистры mnxti, unxti

Регистр может использоваться программным обеспечением для обслуживания следующего горизонтального прерывания для того же режима привилегированности, если оно имеет уровень, превышающий уровень сохраненного контекста прерывания (сохраненный в mcause.pil (ucause.pil)), и уровень порога больший чем порог прерывания соответствующего режима привилегированности, без дополнительных затрат на очистку конвейера прерываний и сохранение/восстановление контекста. Регистр предназначен для доступа с использованием инструкций CSRRSI/CSRRCI, где считываемое значение является указателем на запись в таблице обработчика прерываний, а обратная запись обновляет статус разрешения прерывания. Кроме того, запись в регистр имеет побочные эффекты, которые обновляют состояние контекста прерывания.

В отличии от обычной команды CSR возвращаемое значение отличается от значения, используемого в операции чтения-изменения-записи.

Все инструкции CSR могут быть использованы с этим регистром. Операция будет такой же, как и соответствующая инструкция CSR, используемая с регистром mstatus (ustatus).

Чтение регистра возвращает либо ноль, указывая на отсутствие подходящего прерывания для обработки, либо на то, что прерывание с наивысшим рангом – SHV, или, что система не находится в режиме CLIC, или возвращает ненулевой адрес таблицы обработчиков прерываний для программных векторов прерываний.

Регистры mtvt(utvt) можно установить на адреса памяти таким образом, чтобы запись в таблице находилась по нулевому адресу, и это не отличимо от случая без прерывания.

Если CSR инструкция, которая обращается к `mnxti` (`unxti`) содержит операцию записи, то только для регистра `mstatus` (`ustatus`) используется операция чтение-изменение-запись, в то время как поле `exccode` регистра `mcause` (`ucase`), поле `xl` регистра `mintstatus` (`uintstatus`) также могут быть обновлены новым идентификатором прерывания, уровнем, а ожидающий бит принятого прерывания соответственно очищен.

Если инструкция CSR не содержит операцию записи (например, `csrr t0, mnxti`), то обновление состояния для любого CSR не происходит. Это может быть использовано для определения того, может ли быть принято прерывание без фактического обновления `xl` и `exccode`.

Регистры `mnxti` (`unxti`) предназначены для использования внутри обработчика прерываний после того, как было принято первоначальное прерывание и регистры `mcause` (`ucase`) и `mpers` (`ucers`) обновлены с учетом контекста прерывания и идентификатора прерывания.

Если ожидающее прерывание `edge-triggered`, то аппаратное обеспечение автоматически очистит соответствующий ожидающий бит во время обращения CSR инструкции записи в регистр `mnxti`(`unxti`). Однако, если команда CSR не содержит запись (например, `csrr t0, mnxti`), то обновление состояния ни для одного CSR не происходит, и, таким образом, бит ожидания прерывания не очищается. Такое поведение позволяет программному обеспечению оптимизировать выбор и выполнение прерываний с помощью `mnxti`(`unxti`).

### 27.4.3 Регистры `mintstatus`, `uintstatus`

В регистре `mintstatus` хранится уровень активного прерывания для каждого уровня привилегированности. Поля регистра `mintstatus` доступны только на чтение. Основная причина предоставления доступа к этим полям заключается в поддержке отладки.

Таблица 499 – Регистр `mintstatus`

Номер	31..24	23..16	15..8	7..0
Доступ	R/O	U	R/O	R/O
Сброс	0	-	0	0
	<code>mil</code>		<code>sil</code>	<code>uil</code>

Регистр `uintstatus` предоставляет ограниченный доступ к `mintstatus`.

Таблица 500 – Регистр `uintstatus`

Номер	31..8	7..0
Доступ	U	R/O
Сброс	-	0
		<code>uil</code>

#### 27.4.4 Регистры `mintthresh`, `uintthresh`

Регистр уровня порога прерывания доступен для чтения-записи, содержит 8-разрядное поле (`th`) порогового уровня соответствующего режима привилегированности. Поле `th` хранится в младших 8 битах, в старшие биты должен быть записан ноль.

Обычно уровень порога прерывания используется для реализации критических разделов. Текущий обработчик может временно повысить свой эффективный уровень прерывания, чтобы реализовать критическую секцию среди подмножества уровней, в то же время позволяя более высоким уровням прерывания выполнять вытеснение текущего прерывания.

Эффективный уровень прерывания вычисляется как:

$$\text{effective\_level} = \max(\text{mintstatus.xil}, \text{mintthresh.th}),$$

$$\text{effective\_level} = \max(\text{uintstatus.xil}, \text{uintthresh.th}).$$

Значение `max` используется для предотвращения падения ниже его первоначального уровня, что привело бы к нарушению прав, а также упростило программному обеспечению удаление порога, не зная его собственного уровня, просто записав ноль.

Пороговое значение уровня прерывания допустимо только при запуске в режиме связанной привилегированности, но не в других режимах. Это связано с тем, что прерывания для режимов с более низкими привилегиями всегда отключены, в то время как прерывания для режимов с более высокими привилегиями всегда включены. Например, прерывания в машинном режиме не будут маскироваться установкой порога в пользовательском режиме. Это аналогично тому, как работает `mstatus.mie` не маскирует прерывания машинного режима при запуске в режимах с более низкими привилегиями.

Такое поведение значительно упрощает аппаратную реализацию, поскольку требуется выбрать только одно глобальное максимальное прерывание и сравнить с пороговым значением соответствующего режима привилегированности (игнорируя пороговые значения в других режимах). В противном случае аппаратному обеспечению пришлось бы выбирать несколько максимальных прерываний (по одному на режим привилегированности), сравнить и сопоставить с соответствующими пороговыми значениями, а затем выбрать максимальное прерывание с режимом наивысшей привилегированности.

#### 27.4.5 Регистр `mclicbase`

Регистр `mclicbase` – 32-разрядный регистр доступный только для чтения, предоставляющий базовый адрес регистров, отображенных в память CLIC. Его значение должно быть сконфигурировано или настроено на аппаратном уровне платформы для указания начального адреса регистров, отображенных в память CLIC.

Поскольку карта памяти CLIC должна быть выровнена по границе 4 Кбайт, в регистре `mclicbase` двенадцать младших разрядов доопределены в логический «0».

Регистр используется программным обеспечением для определения местоположения отображаемых в памяти регистров CLIC.

### 27.4.6 Регистры mscratchswl, uscratchswl

Регистр добавлен для быстрого обмена указателем стека между выполняемым кодом прерывания и не прерывания в одинаковом режиме привилегированности.

Пример:

```
csrrw rd, mscratchswl, rs1
// Pseudocode operation.
if ( (mcause.pil==0) != (mintstatus.mil==0) ) then {
t = rs1; rd = mscratch; mscratch = t;
} else {
rd = rs1; // mscratch unchanged.
}
// Usual use: csrrw sp, mscratchswl, sp
```

Регистр mscratchswl (usratchswl) работает аналогично регистру mscratchsw (usratchsw), за исключением того, что условие обмена истинно, когда прерыватель и прерванный не являются одновременно прикладными задачами или оба не являются обработчиками прерываний.

Как и для mscratchsw (usratchsw) для mscratchswl (usratchswl) определена только операция чтения-изменения-записи (swap/CSRRW).

Реализация вариантов без CSRRW (когда rd, либо rs1 – это x0, или rs1 является непосредственным операндом) в mscratchswl (usratchswl) не определена/зарезервирована.

## 27.5 Карта локальных прерываний

Таблица 501 – Прерывания, формируемые периферийными блоками

Номер	Прерывания	Блок	Принцип формирования
0	usip	Ядро	Программное прерывание для User режима привилегированности
1	-	-	Зарезервировано
2	-	-	Зарезервировано
3	msip	Ядро	Программное прерывание для Machine режима привилегированности
4	utip	Системный таймер	Прерывание таймера для User режима привилегированности
5	-	-	Зарезервировано
6	-	-	Зарезервировано
7	mtip	Системный таймер	Прерывание таймера для Machine режима привилегированности
8	-	-	Зарезервировано
9	-	-	Зарезервировано
10	-	-	Зарезервировано

Номер	Прерывания	Блок	Принцип формирования
11	-	-	Зарезервировано
12	csip	Ядро	Программное прерывание CLIC
13	-	-	Зарезервировано
14	-	-	Зарезервировано
15	-	-	Зарезервировано
16	int_local0	DMA	Прерывания от DMA DMA_ERR или DMA_DONE. Обработка прерываний от DMA в соответствии с разделом Error signaling технического описания DMA
17	int_local1	UART1	Сигнал UARTINTR
18	int_local2	UART2	Сигнал UARTINTR
19	int_local3	SSP1	Сигнал SSPINTR
20	int_local4	POWER	Сигнал прерывания от POWER Detector
21	int_local5	WWDG	Сигнал прерывания от WWDG
22	int_local6	Timer1	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
23	int_local7	Timer2	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
24	int_local8	ADC	Сигналы прерываний от АЦП EOCIF_1 или AWOIF_1 или EOCIF_2 или AWOIF_2
25	int_local9	I2C	Сигнал I2C_INT
26	int_local10	BACKUP	Прерывание от ВКР и часов реального времени
27	int_local11	Внешнее прерывание 1	Сигнал EXT_INT0 Вывод PA[10] в основном режиме
28	int_local12	Внешнее прерывание 2	Сигнал EXT_INT1 Вывод PC[4], PB[6] в альтернативном режиме
29	int_local13	Внешнее прерывание 3	Сигнал EXT_INT2 Вывод PC[5], PB[7] в альтернативном режиме
30	int_local14	ADCUI	Прерывание от АЦП для измерения напряжений и токов
31	int_local15	ADCUI	Прерывание от АЦП для измерения напряжений и токов
32	int_local16	ADCUI	Прерывание от АЦП для измерения напряжений и токов
33	int_local17	-	-
34	int_local18	-	-
35	int_local19	-	-
36	int_local20	-	-
37	int_local21	RANDOM	Прерывание блока генератора случайных чисел
38	int_local22	ISO7816	Прерывание блока UART ISO7816
39	int_local23	UART3	Сигнал UARTINTR
40	int_local24	SSP2	Сигнал SSPINTR
41	int_local25	SSP3	Сигнал SSPINTR
42	int_local26	Timer3	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
43	int_local27	Timer4	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
44	int_local28	UART4	Сигнал UARTINTR

Примечание – Выводы в функции EXT\_INT0-EXT\_INT2 являются входами прерываний по уровню. Т.е. пока на вывод EXT\_INTx подан сигнал высокого уровня, генерируется прерывание. Если на входе сигнал низкого уровня, то прерывание не генерируется

## 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).

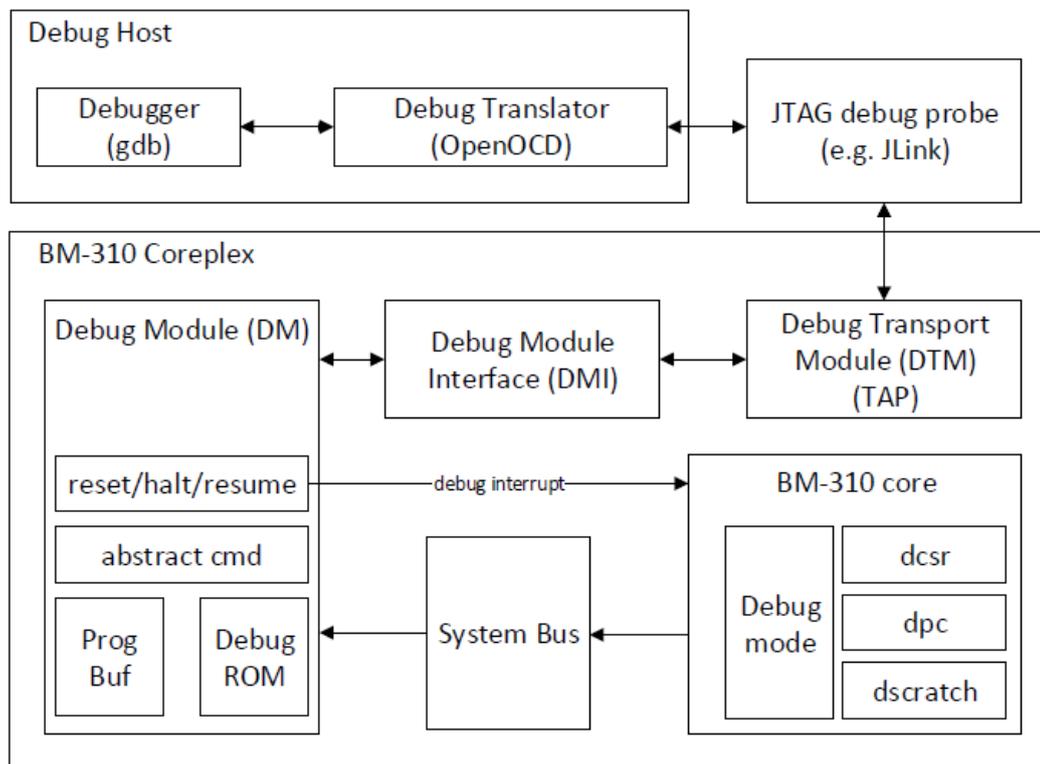


Рисунок 98 – Архитектура системы отладки

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 99. Переход из области синхросигнала JTAG в область синхросигнала ядра происходит в блоке DMI.

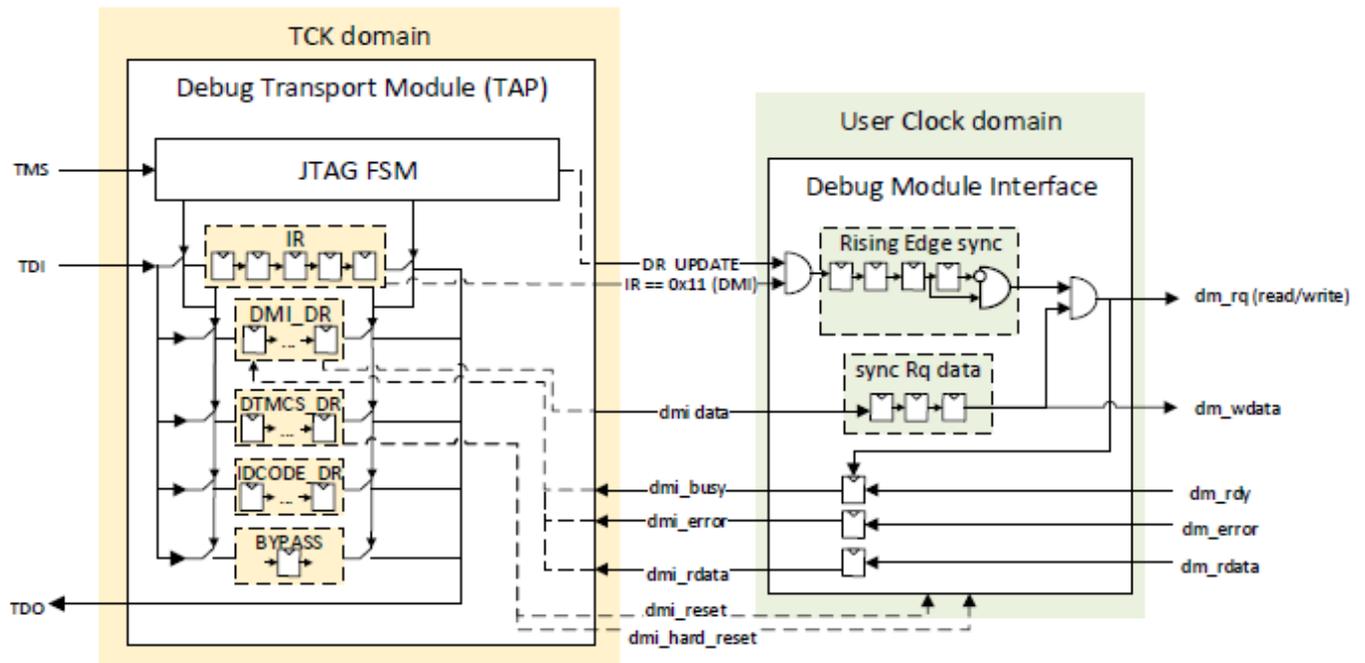


Рисунок 99 – Интерфейс DM

### 28.1.1 Регистры DTM

Таблица 502 – 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 503 – 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 ещё 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 504 – 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 ещё 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 505.

Таблица 505 – 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 506.

Таблица 506 – 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
0x0000_0300	Where to	R
0x0000_0338	Abstract command 0	R

Адрес	Название	Доступ
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_0800-0xFFFF	Debug ROM	R

Таблица 507 – 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 508 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 508 – 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 509 – 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 выполнении инструкции ebreak в режиме M процессор переходит в режим отладки	0
12	ebreaku	R/W	Значение 1 – при выполнении 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 510 – 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 511 – 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 0x500) 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.

## 29 Сторожевые таймеры

### 29.1 Регистры блока сторожевых таймеров

Таблица 512 – Описание регистров блока сторожевых таймеров

Базовый Адрес	Название	Описание
0x5005_0000	IWDG	Сторожевой таймер IWDG
Смещение		
0x00	IWDG_KR[15:0]	Регистр Ключа
0x04	IWDG_PR[2:0]	Делитель частоты сторожевого таймера
0x08	IWDG_PRL[11:0]	Регистр основания счета сторожевого таймера
0x0C	IWDG_SR[1:0]	Регистр статуса сторожевого таймера

Таблица 513 – Описание регистров блока сторожевых таймеров

Базовый Адрес	Название	Описание
0x5004_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

#### 29.1.1 Регистры сторожевого таймера IWDG

IWDG – это независимый сторожевой таймер, который считает сверху вниз до нуля, начиная со значения, которое мы в нём установим. Как только счётчик дойдёт до нуля, то от сторожевого таймера поступит сигнал сброса, который заставит систему перезагрузиться. Но чтобы этого не произошло, мы должны заблаговременно перезагрузить IWDG или восстановить регистр, внося заново в него требуемое число и тогда таймер начнёт снова обратный отсчёт.

Блок-схема независимого сторожевого таймера приведена на рисунке 100

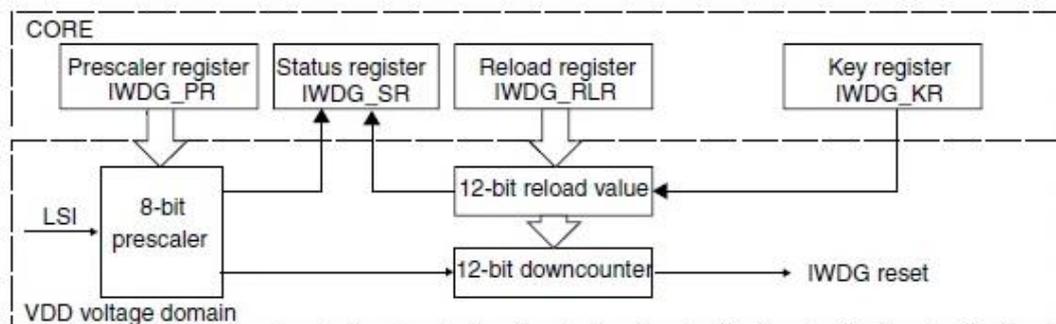


Рисунок 100 – Блок-схема независимого сторожевого таймера

Таймер тактируется от отдельного низкоскоростного источника тактовых сигналов LSI (low-speed clock), от которого тактовые сигналы поступают в 8-битный

пределитель, коэффициент которого устанавливается с помощью регистра IWDG\_PR. В регистр IWDG\_PLR мы заносим значение, от которого таймер при старте начнёт обратный отсчёт. В регистр статуса IWDG\_SR устанавливается соответствующий бит в момент изменения значения в регистре пределителя IWDG\_PR, а также в момент занесения или обновления значения в регистре IWDG\_PLR. Предварительно в ключевой регистр IWDG\_KR для доступа к данным операциям изменений мы заносим определённое ключевое число. После того, как 12-битный обратный счётчик досчитает до 0, IWDG выдаст на своём выходе сигнал на перезагрузку системы.

Чтобы перезагрузка системы не произошла, мы должны своевременно подать команду на перезагрузку счётчика, чтобы он заново начал обратный отсчёт. Тем самым и обеспечивается контроль над зависанием программы в определённых точках, где такое возможно. Если мы не перезагрузим IWDG вовремя, это будет означать, что программа зависла и система перезагрузится.

### 29.1.1.1 IWDG\_KR

Таблица 514 – Регистр IWDG\_KR

Номер	31...16	15...0
Доступ	U	W
Сброс		0
	-	KEY[15:0]

Таблица 515 – 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]	Значение ключа (только запись, читается 0000h). Эти биты должны перезаписываться программно через определённые интервалы ключевым значением AAAAh, в противном случае сторожевой таймер генерирует сброс, если таймер достиг значения нуля. Запись ключевого значения 5555h разрешает доступ по записи к регистрам IWDG_PR и IWDG_RLR. Запись ключевого значения CCCCCh разрешает работу сторожевого таймера (за исключением, если сторожевой таймер разрешается аппаратно битами конфигурации)

### 29.1.1.2 IWDG\_PR

Таблица 516 – Регистр IWDG\_PR

Номер	31...3	2	1	0
Доступ	U	R/W	R/W	R/W
Сброс		0	0	0
	-	PR2	PR1	PR0

Таблица 517 – 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> <p>000 – делитель на 4                      001 – делитель на 8                      010 – делитель на 16                      011 – делитель на 32                      100 – делитель на 64                      101 – делитель на 128                      110 – делитель на 256                      111 – делитель на 256</p> <p>Чтение и запись этого регистра правомерна только, если бит PVU=0 в регистре IWDG_SR</p>

### 29.1.1.3 IWDG\_RLR

Таблица 518 – Регистр IWDG\_RLR

Номер	31...12	11...0
Доступ	U	R/W
Сброс		1
	-	RLR[11:0]

Таблица 519 – 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 сторожевого таймера.</p> <p>Значение этих бит по доступу защищено с помощью регистра IWDG_KR. Эти биты записываются программно и определяют значение, загружаемое в сторожевой таймер в момент записи значение AAAAh в регистр IWDG_KR. Сторожевой таймер декрементируется, начиная с этого значения. Период таймаута сторожевого таймера - функция от этого значения и делителя частоты. Чтение и запись этого регистра правомерна только, если бит RVU=0 в регистре IWDG_SR</p>

### 29.1.1.4 IWDG\_SR

Таблица 520 – Регистр IWDG\_SR

Номер	31...2	1	0
Доступ	U	R	R
Сброс		0	0
	-	RVU	PVU

Таблица 521 – 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	Флаг обновления значения сторожевого таймера. Этот бит устанавливается аппаратно и служит признаком того, что обновляется значение сторожевого таймера из регистра перезагрузки. Этот бит сбрасывается, если обновление завершено. Значение регистра перезагрузки может быть обновлено только, если этот бит равен нулю
0	PVU	Флаг обновления делителя частоты сторожевого таймера. Этот бит устанавливается аппаратно и служит признаком того, что обновляется значение делителя частоты сторожевого таймера. Этот бит сбрасывается, если обновление завершено. Значение регистра делителя частоты может быть обновлено только, если этот бит равен нулю

### 29.1.2 Регистры сторожевого таймера WWDG

WWDG в данном контроллере – это оконный сторожевой таймер, счётчик 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 ещё 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 счётчик досчитает вниз до 0x3F, но ещё 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 счётчика, с которого он начинает обратный отсчёт. Таким образом, мы получаем возможность проверки определённых временных интервалов хода нашей программы.

Блок-схема оконного сторожевого таймера приведена на рисунке 101.

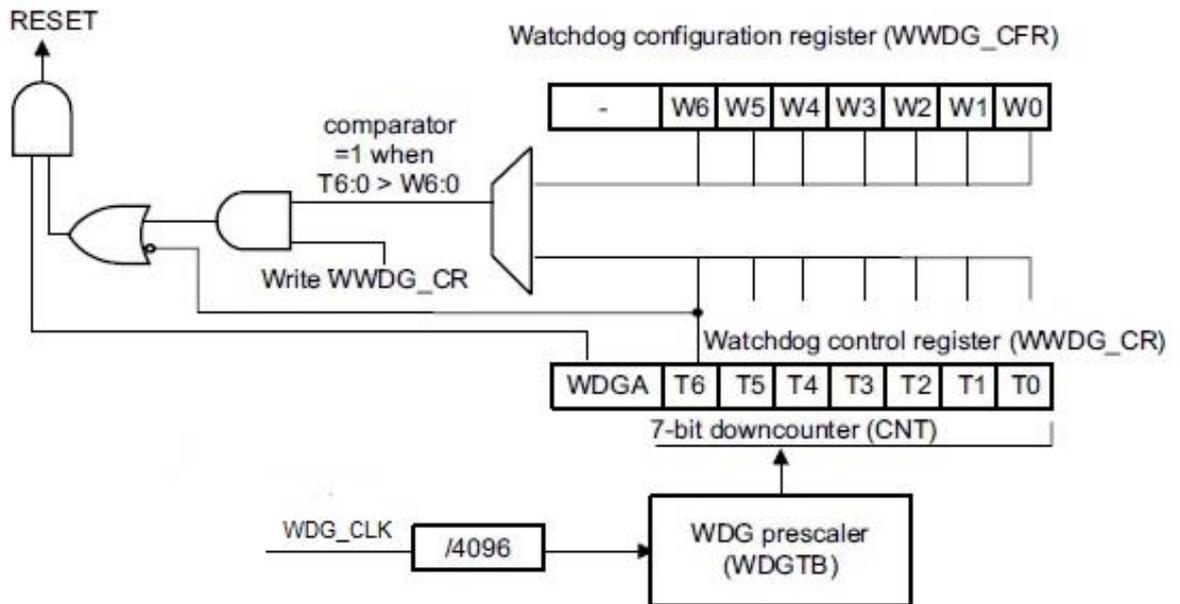


Рисунок 101 – Блок-схема оконного сторожевого таймера

Тактовые сигналы счётчика идут через первый делитель с постоянным коэффициентом деления 4096 с шины WDG\_CLK на второй, регулируемый делитель, значение коэффициента деления которого мы можем устанавливать с помощью битов регистра WGTB. После второго делителя тактовые сигналы попадают на счётчик обратного отсчёта.

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 пойдёт 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 счётчик обратного отсчёта ещё 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 102.

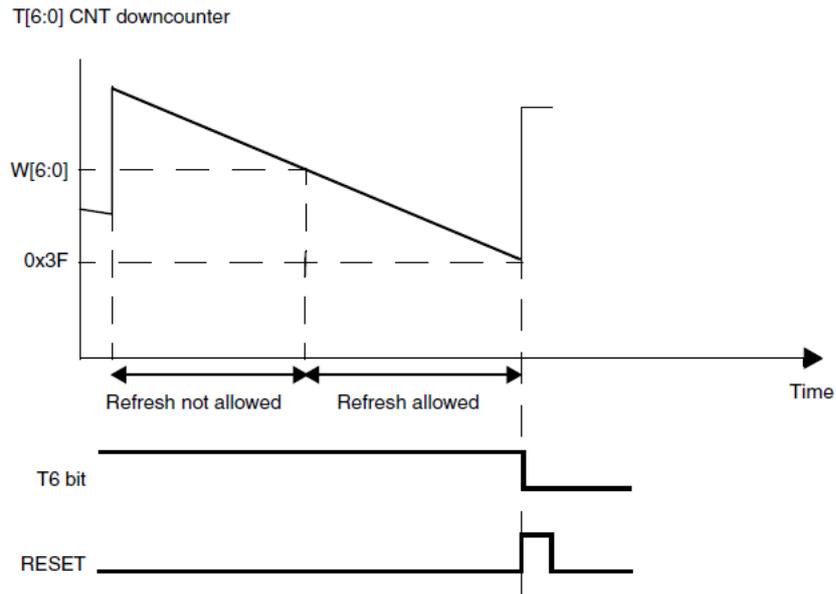


Рисунок 102 – Временная диаграмма оконного сторожевого таймера

Можно рассчитать время работы WWDG от старта до окончания счёта и перезагрузки по следующей формуле

$$t_{WWDG} = t_{WDG\_CLK} \cdot 4096 \cdot 2^{WGTB} \cdot (T[5:0] + 1), \text{ мс} \quad (28)$$

где  $t_{WWDG}$  – таймаут;  
 $t_{WDG\_CLK}$  – период синхросигнала APB шины в мс.

### 29.1.2.1 WWDG\_CR

Таблица 522 – Регистр WWDG\_CR

Номер	31...8	7	6...0
Доступ	U	R/S	R/W
Сброс		0	1
	-	WDGA	T6...T0

Таблица 523 – 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
6...0	T[6:0]	Значение семиразрядного счётчика (от старших разрядов к младшим) Эти биты содержат значение сторожевого таймера, который декрементируется каждые $4096 \times 2^{WDGTB}$ циклов частоты WDG_CLK периферийной шины APB

### 29.1.2.2 WWDG\_CFR

Таблица 524 – Регистр WWDG\_CFR

Номер	31...10	9	8	7	6...0
Доступ	U	R/S	R/W	R/W	R/W
Сброс		0	0	0	1
	-	EWI	WDGTB1	WDGTB0	W6...W0

Таблица 525 – 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

### 29.1.2.3 WWDG\_SR

Таблица 526 – Регистр WWDG\_SR

Номер	31...2	1	0
Доступ	U	R	R/C
Сброс		1	0
	-	WDG_WEC	EWIF

Таблица 527 – 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 Генератор случайных чисел RANDOM

### 30.1 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 1;
- 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 окончанию 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 103 – Структурная схема блока генератора случайных чисел

## 30.2 Регистры модуля

Таблица 528 – Регистры модуля

Базовый Адрес	Название	Описание
0x500B_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 счётчика паузы
0x18	TEMP_REG	Регистр сбора случайного числа

### 30.2.1 Регистр статуса и управления STAT\_CTRL\_REG

Относительный адрес: 0x00, начальное значение 0x000000F0.

Таблица 529 – 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 вышла 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 после окончания сбора очередного числа	RW
11, 10	–	Зарезервировано	RO
17...12	BIT_CN	Счетчик оставшихся бит до сбора нового случайного числа	RO
31...18	–	Зарезервировано	RO

### 30.2.2 Регистр управления прерыванием INT\_CTRL\_REG

Относительный адрес: 0x04.

Таблица 530 – 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

### 30.2.3 Регистр делителя клона генератора CLK\_DIV\_REG

Относительный адрес: 0x08, начальное значение 7.

Таблица 531 – 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 клона зашелки данных генератора используется входной клок модуля, деленный на $2 \cdot (DIV + 1)$	RW
31...16	-	Зарезервировано	RO

### 30.2.4 Регистр паузы включения PAUSE\_REG

Относительный адрес: 0x0C.

Таблица 532 – 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 выйдет данная пауза. Если включение происходит во время сбора случайного числа, сбор приостанавливается до тех пор, пока не выйдет данная пауза	RW

### 30.2.5 Регистр случайного значения OUTPUT\_REG

Относительный адрес: 0x10.

Таблица 533 – Регистр OUTPUT\_REG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31...0	DATA	Последние собранное случайного число. Во время сбора очередного случайного числа, значение данного регистра неизменно. Обновление данных происходит по окончанию сбора случайного числа	RO

### 30.2.6 Регистр счетчика паузы PAUSE\_CNT\_REG

Относительный адрес: 0x14.

Таблица 534 – 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

### 30.2.7 Регистр сбора случайного числа TEMP\_REG

Относительный адрес: 0x18.

Таблица 535 – 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 справки)	RO

## 31 Контроллер UART (ISO7816)

### 31.1 Введение

Универсальный синхронно-асинхронный приемопередатчик (USART) предлагает гибкие средства полдуплексного 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).

### 31.2 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 (8 или 9 бит)
- Поддержка 2 стоп-бит
- Тактовый 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
- 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

### 31.3 Функциональное описание USART

Интерфейс подключается к внешним приборам при помощи трех выводов (см. рисунок 104). 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 4-битной дробной частью
  - Регистр статуса (USART\_SR)
  - Регистр данных (USART\_DR)
  - Регистр скорости передачи (USART\_BRR) - 12-битная мантисса и 4-битная дробная часть
  - Регистр защитного интервала (Guardtime register) (USART\_GTPR) в случае использования режима Smartcard.

Описание регистров и их бит приведено в разделе «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 дополнительно 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).

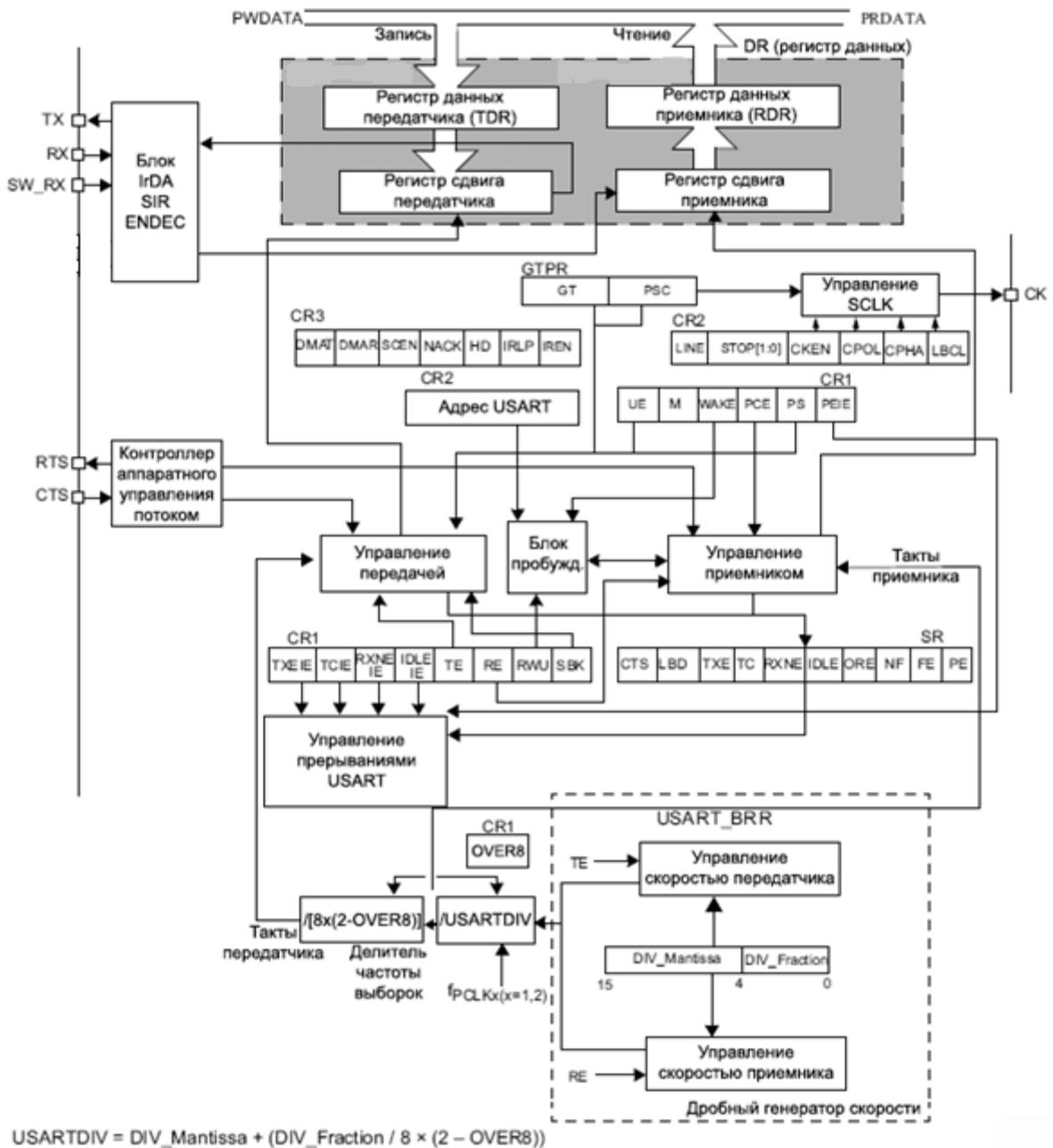


Рисунок 104 – Блок схема USART

### 31.3.1 Описание символов USART

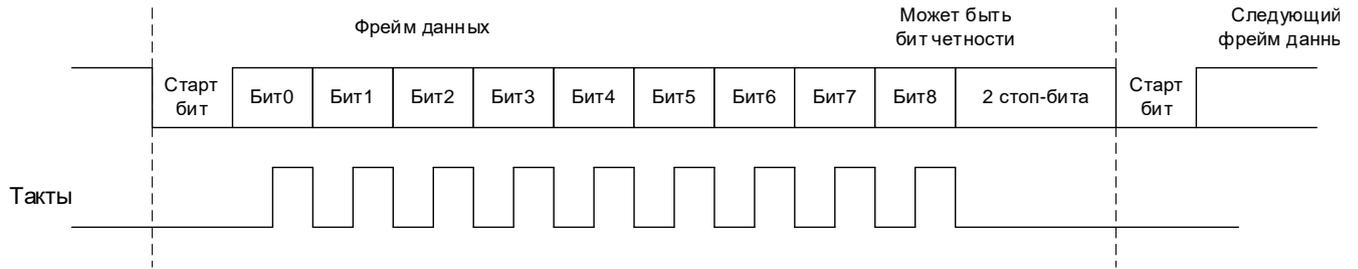
Для передачи можно выбрать длину слова 8 или 9 бит путем программирования бита M в регистре USART\_CR1 (см. рисунок 105).

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.

**Длина слова 9 бит (бит М установлен), 2 стоп-бита**



**Длина слова 8 бит (бит М сброшен), 2 стоп-бита**

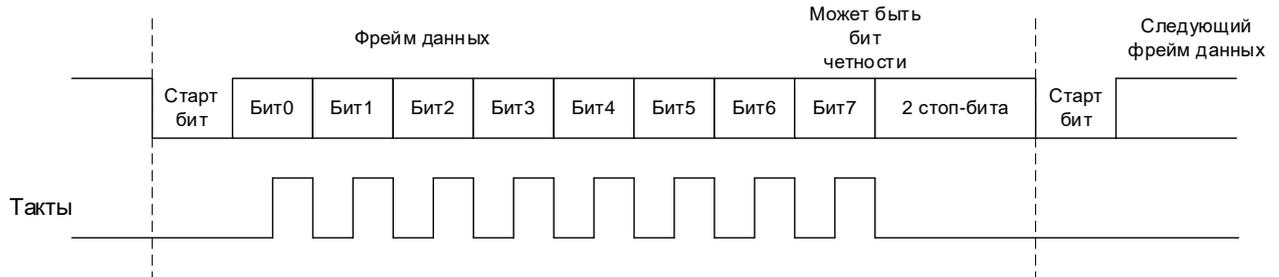


Рисунок 105 – Программирование длины слова

**31.3.2 Передатчик**

Передатчик может отправлять слова данных по 8 или 9 бит, в зависимости от статуса бита М.

Когда установлен бит разрешения передачи (TE), данные из регистра сдвига выводятся на вывод TX, и соответствующие тактовые импульсы выводятся на вывод СК.

**31.3.2.1 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 104).

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 останутся. Текущие передаваемые данные будут потеряны.

После разрешения бита TE будет отправлен фрейм idle.

**31.3.2.2 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.

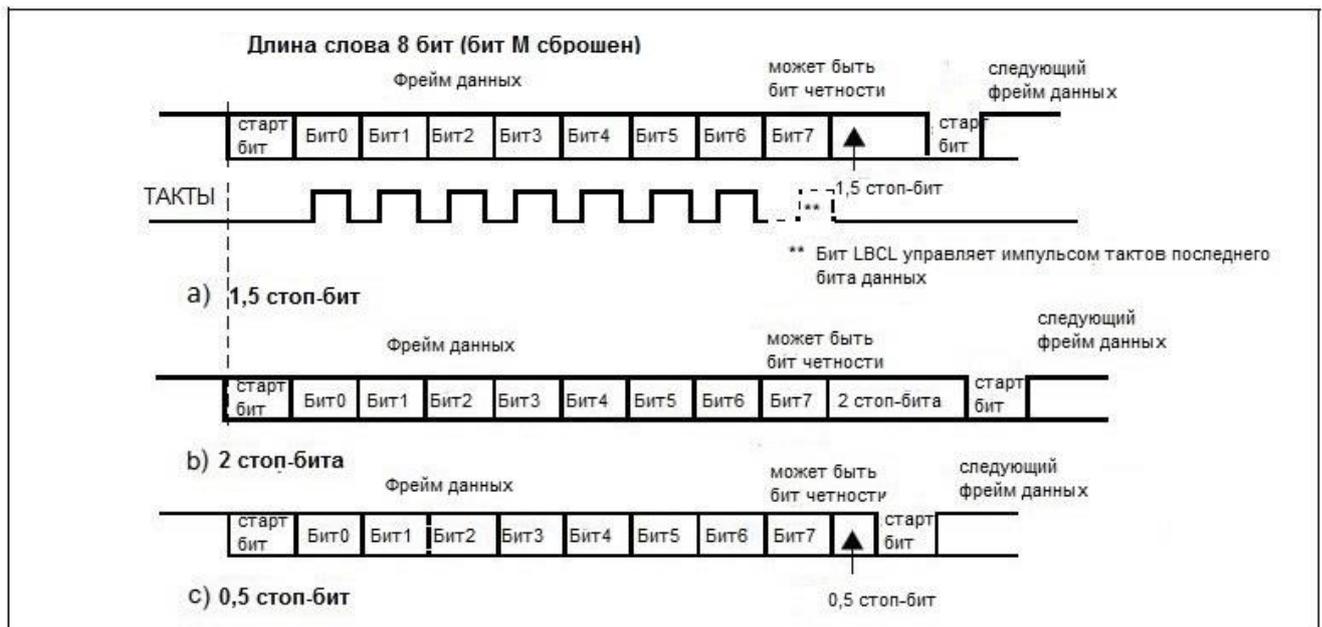


Рисунок 106 – Конфигурируемые стоп-биты

Процедура программирования:

1. Разрешить USART записью в 1 бита UE в регистре USART\_CR1;
  2. Запрограммировать бит М 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 TC 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.

### 31.3.2.3 Однобайтный 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 TC 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 TC установится в 1, перед запретом USART или переходом микроконтроллера в режим пониженного потребления (см. рисунок 107).

Бит TC сбрасывается следующей программной последовательностью:

1. Чтение из регистра USART\_SR;
2. Запись в регистр USART\_DR.

Примечание – Бит TC может быть также сброшен при помощи записи «0». Рекомендуется использовать данную последовательность для сброса бита только в случае мультибуферного обмена данными.

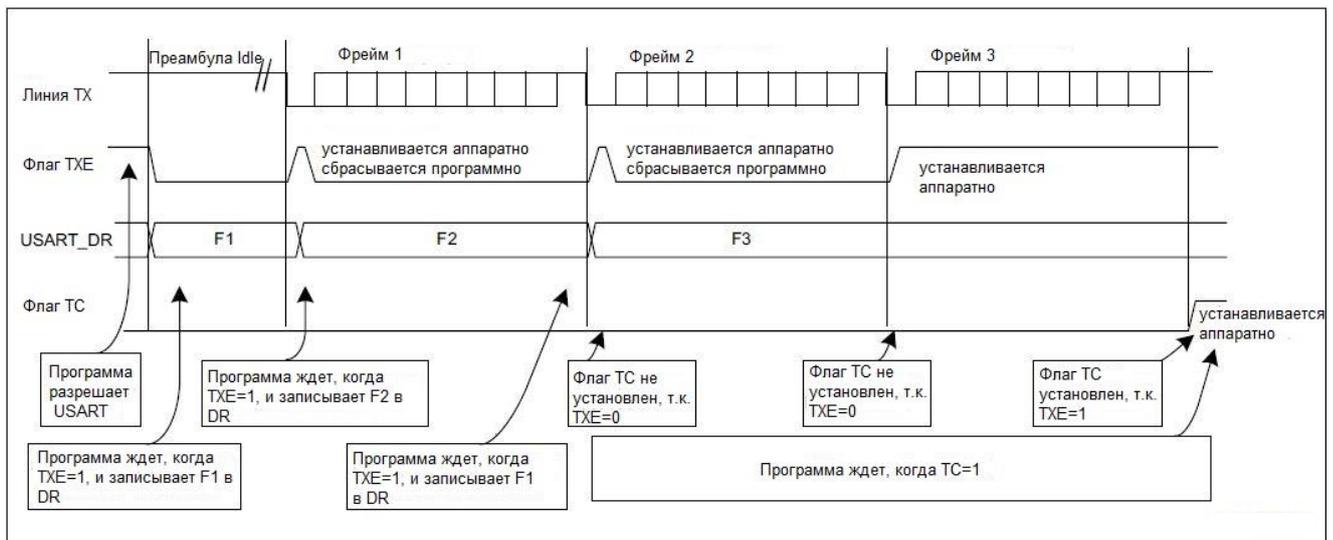


Рисунок 107 – Состояние TC/TXE во время передачи

### 31.3.3 Приемник

USART может принимать слово данных разрядностью 8 или 9 бит, в зависимости от значения бита M в регистре USART\_CR1.

#### 31.3.3.1 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 последовательностью выборки сигнала: 1110X0X0X0000.

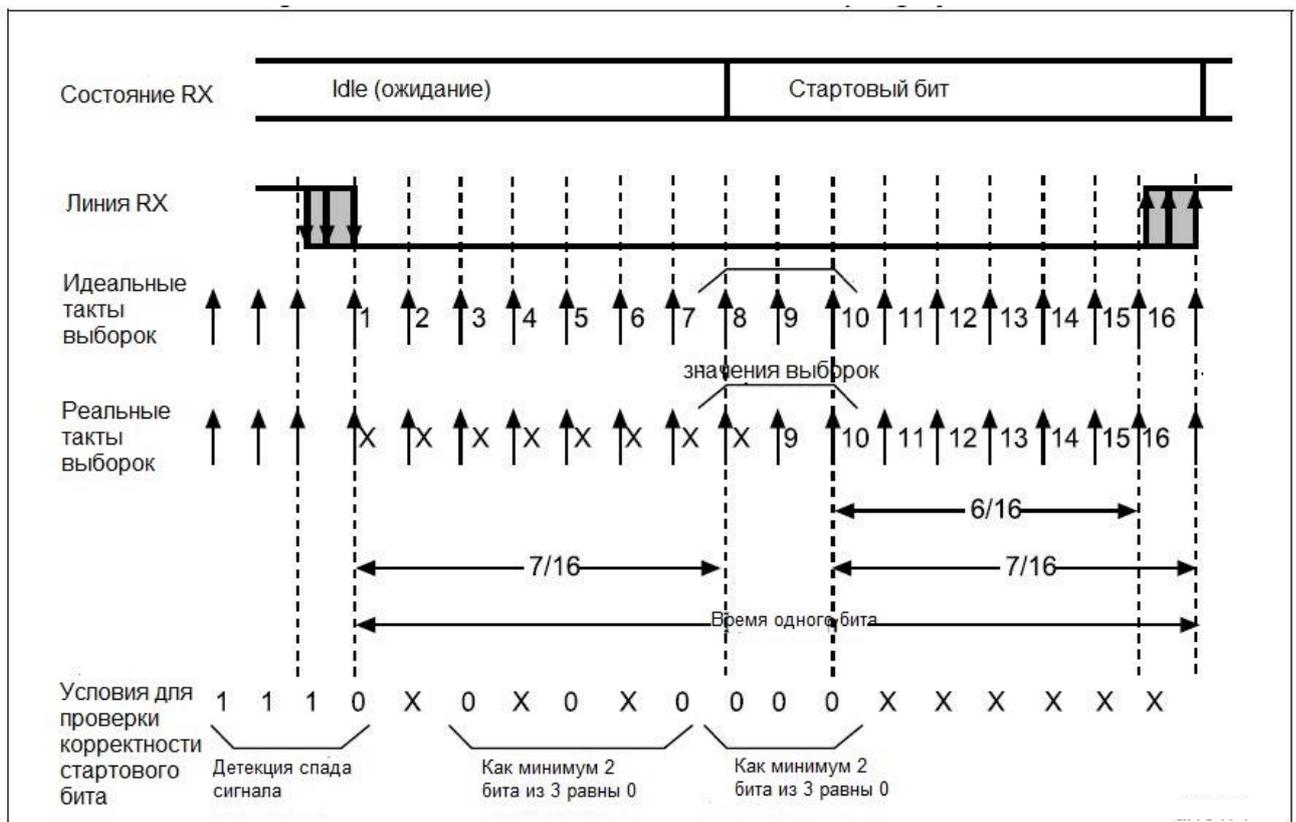


Рисунок 108 – Обнаружение стартового бита при передискретизации на 16 и 8

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.

### 31.3.3.2 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.

### 31.3.3.3 Ошибка переполнения (overrun)

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).

#### 31.3.3.4 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 109 и 110).

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.

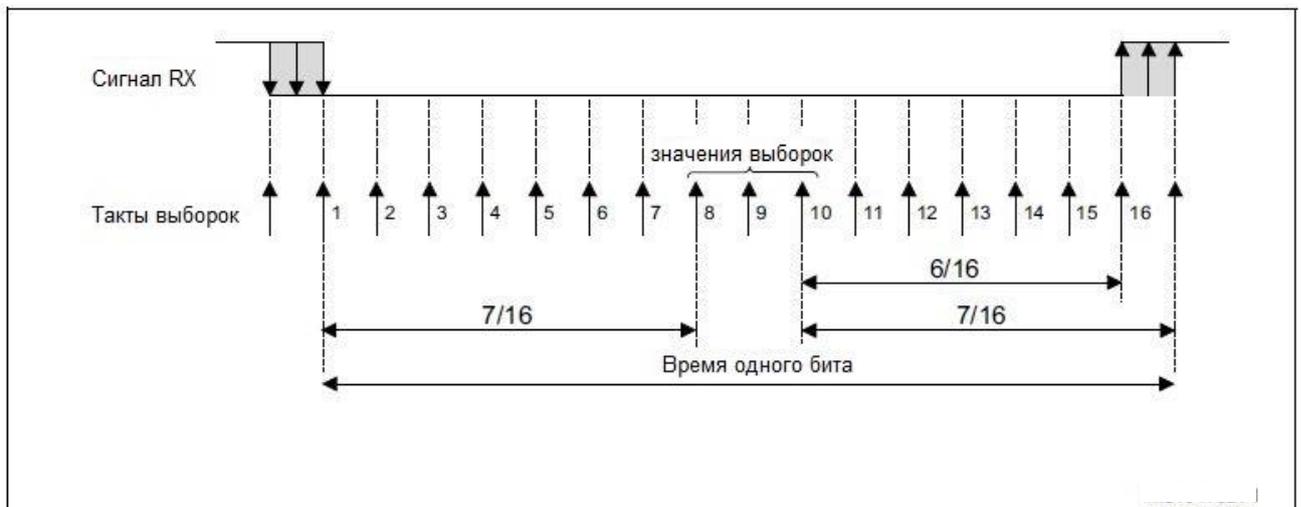


Рисунок 109 – Выборка данных при передискретизации на 16

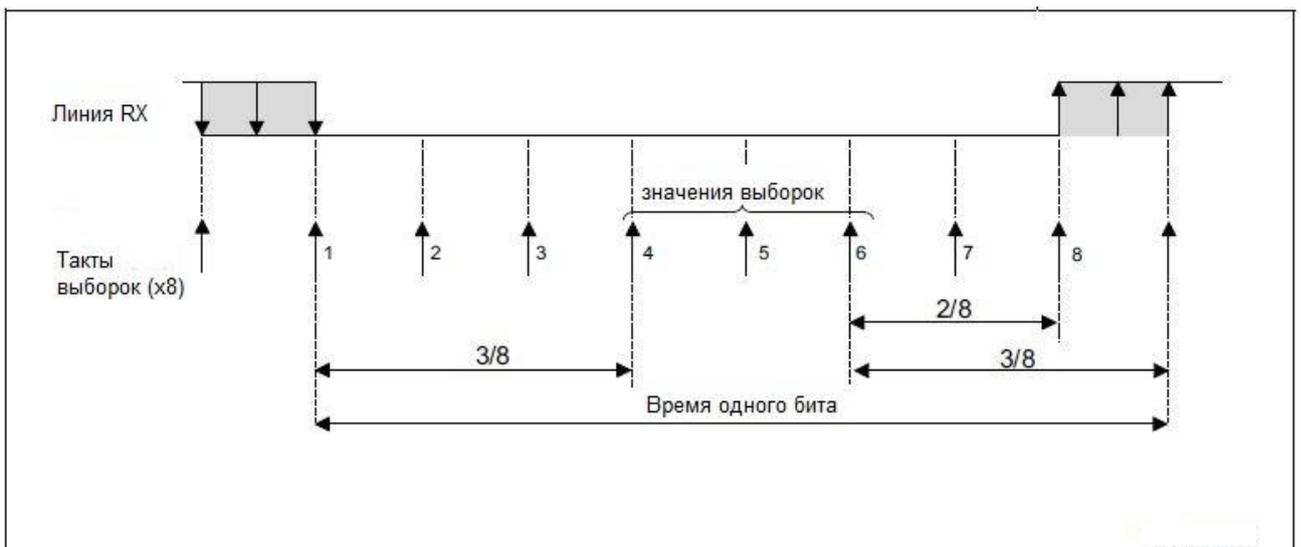


Рисунок 110 – Выборка данных при передискретизации на 8

Таблица 536 – Обнаружение шумов для считанных данных

Значение	NF статус	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

### 31.3.3.5 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.

### 31.3.3.6 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.

### 31.3.4 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

$$\text{Tx/Rx baud} = \frac{f_{\text{СК}}}{8 \cdot (2 - \text{OVER8}) \cdot \text{USARTDIV}} \quad (29)$$

Скорость для режимов Smartcard вычисляется по формуле

$$\text{Tx/Rx baud} = \frac{f_{\text{СК}}}{16 \times \text{USARTDIV}} \quad (30)$$

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.

#### 31.3.4.1 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:

$$DIV\_Fraction = 16 \cdot 0d0.62 = 0d9.92$$

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:

$$DIV\_Fraction = 16 \cdot 0d0.99 = 0d15.84$$

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

### 31.3.4.2 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

$$Mantissa(USARTDIV) = 0d27$$

$$Fraction(USARTDIV) = 6/8 = 0d0.75$$

$$\text{Therefore } USARTDIV = 0d27.75$$

Пример 2:

Для программирования USARTDIV = 0d25.62

получается:

$$DIV\_Fraction = 8 \cdot 0d0.62 = 0d4.96$$

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  $0d8 = 0x8 \Rightarrow$  переполнение  $DIV\_frac[2:0] \Rightarrow$  должен быть добавлен перенос (carry) к мантиссе

$$DIV\_Mantissa = mantissa (0d50.990 + carry) = 0d51 = 0x33$$

$$\text{Тогда, } USART\_BRR = 0x0330 \Rightarrow USARTDIV = 0d51.000$$

Таблица 537 – Определение погрешности для запрограммированных скоростей передачи при  $f_{PCLK} = 60 \text{ МГц}^{(1)}$

№	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
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

<sup>(1)</sup> 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.

### 31.3.5 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 538 – 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 539 – Допуск приемника USART при DIV\_Fraction, не равным 0

Бит M	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%

### 31.3.6 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.

### 31.3.6.1 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 111 показан пример поведения устройства в режиме молчания с использованием обнаружения линии Idle.

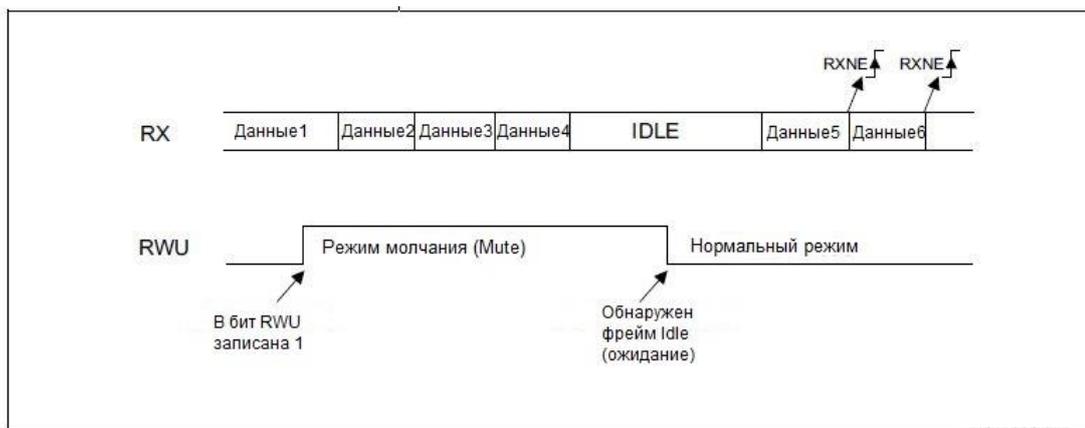


Рисунок 111 – Режим молчания (Mute) с обнаружением линии Idle

### 31.3.6.2 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 4-битное 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 112.

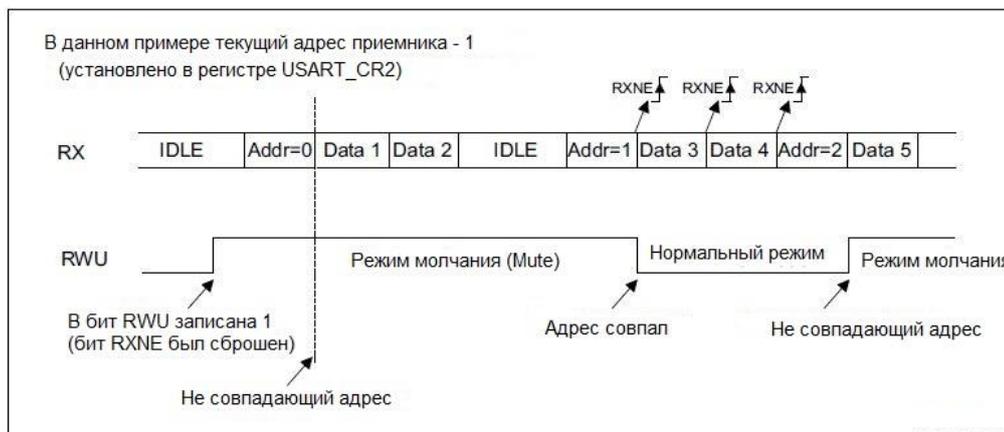


Рисунок 112 – Режим молчания (Mute) с обнаружением метки адреса

### 31.3.7 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 540.

Таблица 540 – Форматы фрейма

Бит M	Бит PCE	Фрейм USART <sup>(1)</sup>
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

<sup>(1)</sup> SB: стартовый бит, STB: стоп-бит, PB: бит четности

### 31.3.7.1 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).

### 31.3.7.2 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).

### 31.3.7.3 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).

### 31.3.7.4 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.

## 31.3.8 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 113, 114 и 115)

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).

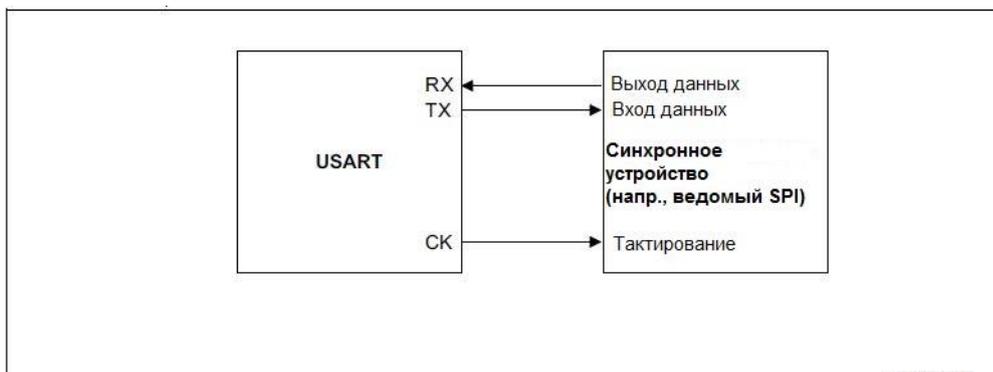


Рисунок 113 – Пример синхронной передачи USART

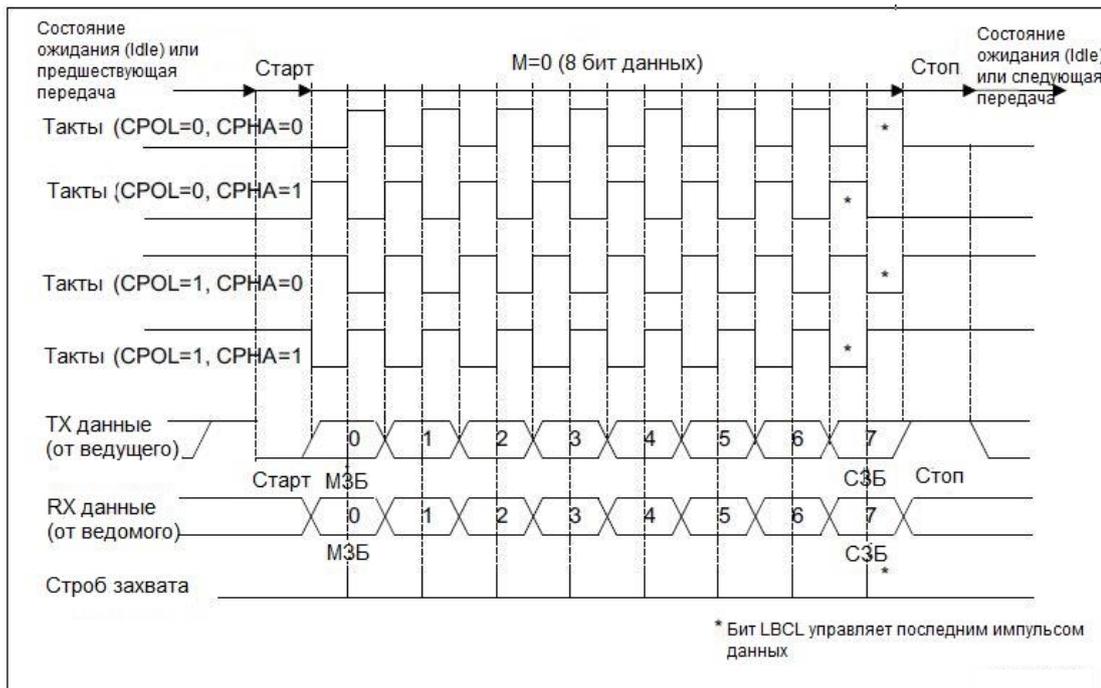


Рисунок 114 – Диаграмма тактирования данных USART (M=0)

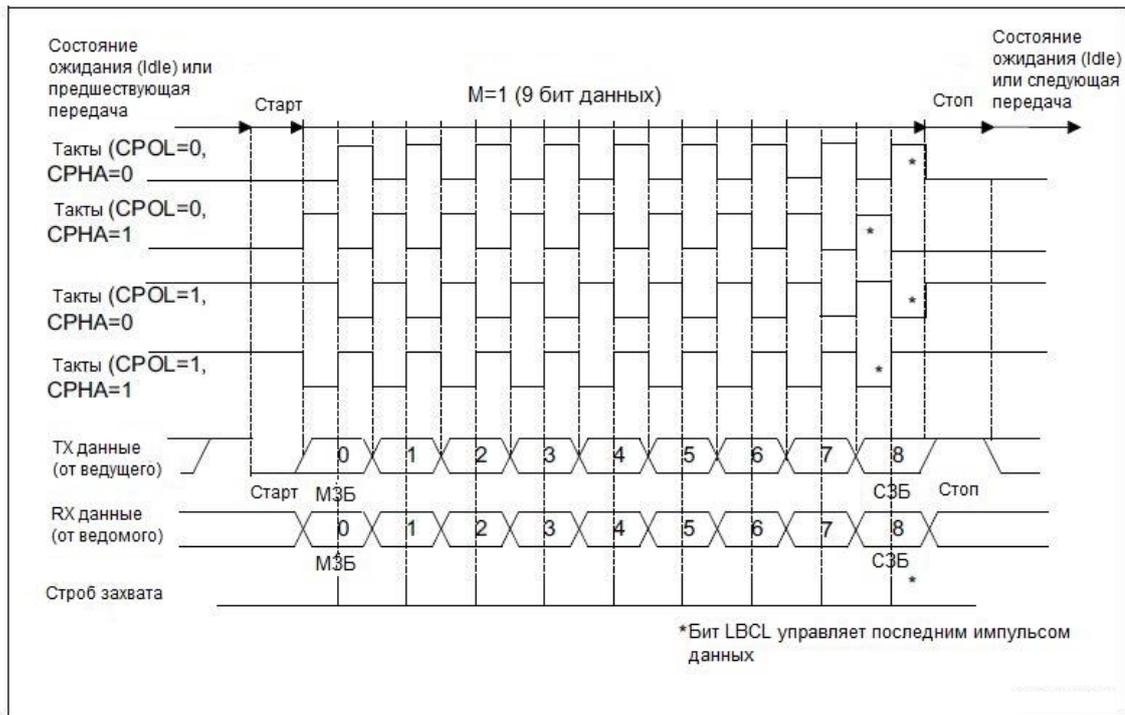


Рисунок 115 – Диаграмма тактирования данных USART (M=1)

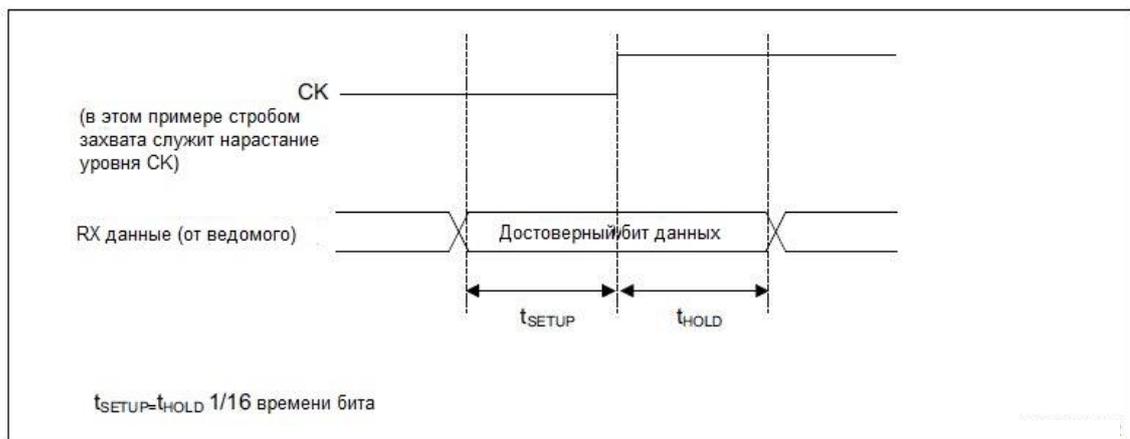


Рисунок 116 – Время установки/удержания RX

Примечание – Функция тактирования СК отличается в режиме Smartcard (См. далее описание режима Smartcard).

### 31.3.9 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 так же, 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.

### 31.3.10 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 117 показывает пример, как выглядит сигнал данных с ошибкой четности и без ошибки четности.

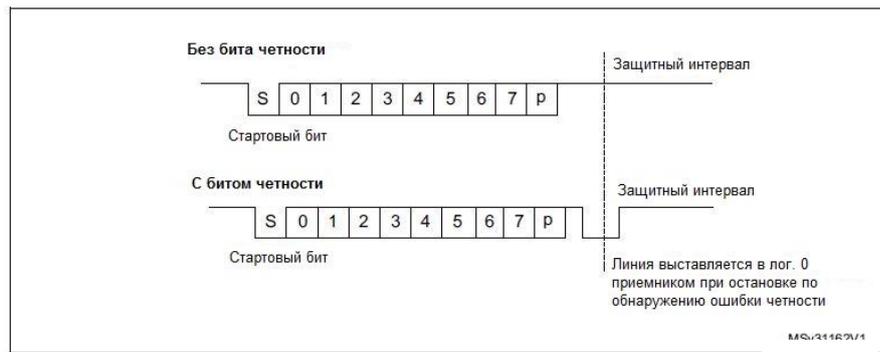


Рисунок 117 – Асинхронный протокол ISO 7816-3

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) генерирует 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 TC может быть задержана путем программирования регистра защитного интервала времени (Guard Time). При нормальной работе флаг TC 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 TC подтягивается в 0. Когда счетчик достигает запрограммированного значения, устанавливается флаг TC.

– На снятие флага TC 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 обработает 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 118 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.

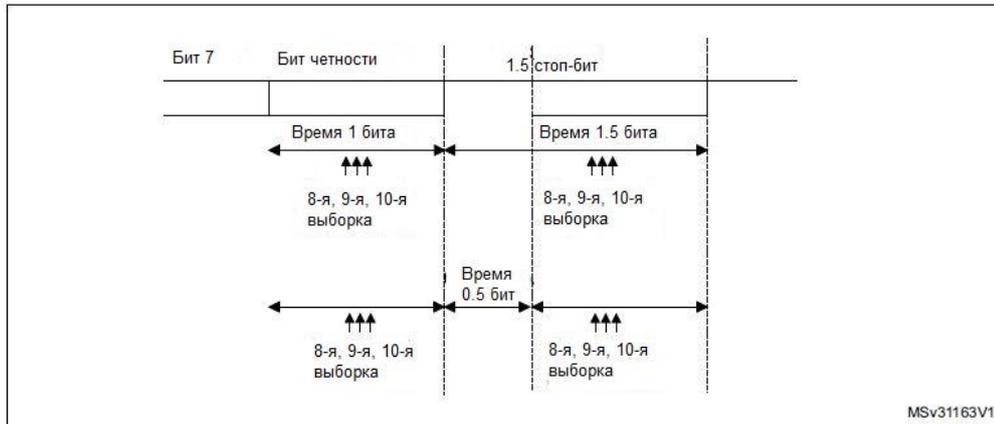


Рисунок 118 – Обнаружение ошибки четности с использованием 1,5 стоп-бит

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  $f_{CK}/2$  до  $f_{CK}/62$ , где  $f_{CK}$  это частота тактов входа периферии.

### 31.3.11 Аппаратное управление потоком

Можно управлять потоком последовательных данных между двумя устройствами, используя вход CTS и выход RTS. Рисунок 119 показывает схему соединения двух устройств для данного режима:

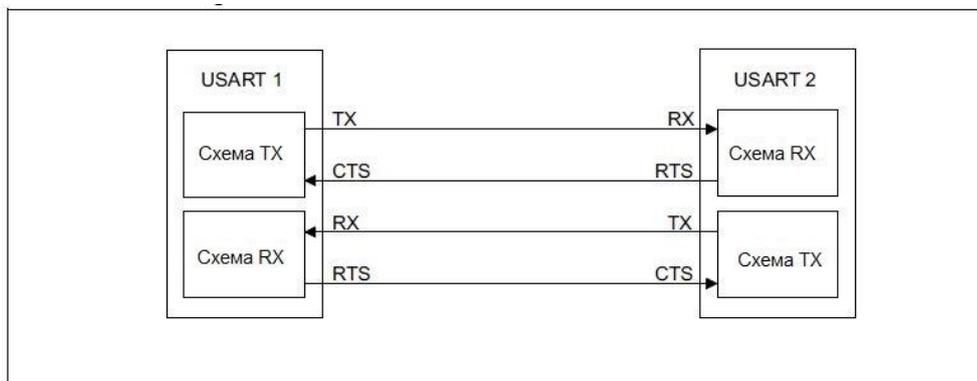


Рисунок 119 – Аппаратное управление потоком данных между двумя устройствами USART

Управление потоком RTS и CTS можно разрешить независимо записью битов RTSE и CTSE соответственно в 1 (в регистре USART\_CR3).

### 31.3.11.1 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 120 показывает пример обмена с разрешенным управлением потоком RTS.

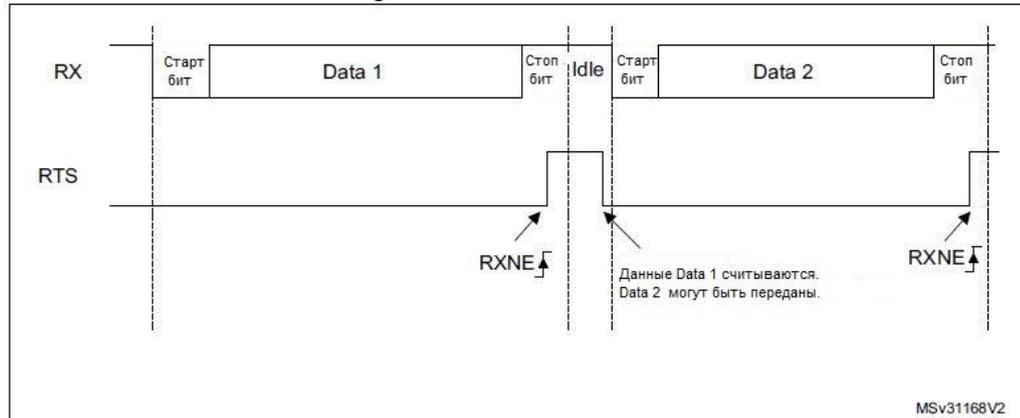


Рисунок 120 – Управление потоком RTS

### 31.3.11.2 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 121 показывает пример обмена с разрешенным управлением потоком CTS.

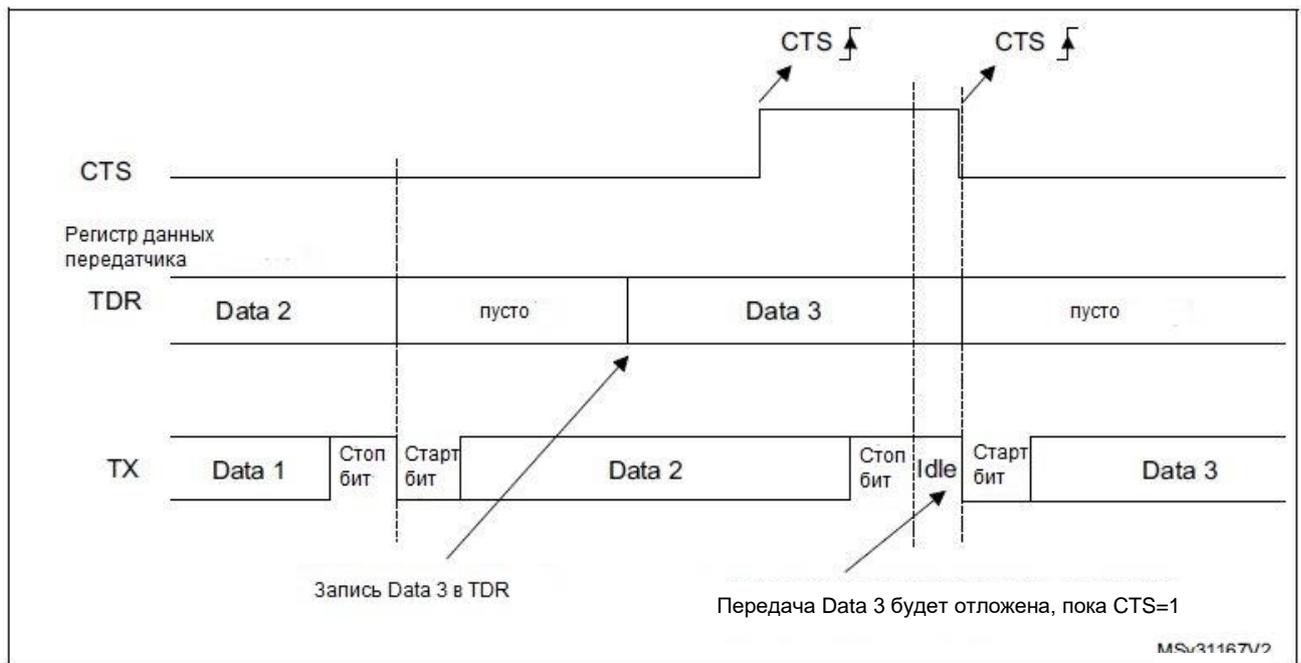


Рисунок 121 – Управление потоком CTS

Примечание – Особое поведение фреймов break: при разрешенном управлении потоком CTS, передатчик не проверяет статус входа CTS для отправки сигнала break

### 31.4 Прерывания USART

Таблица 541 – 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 Break	LBD	LBDIE
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 122).

– 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.

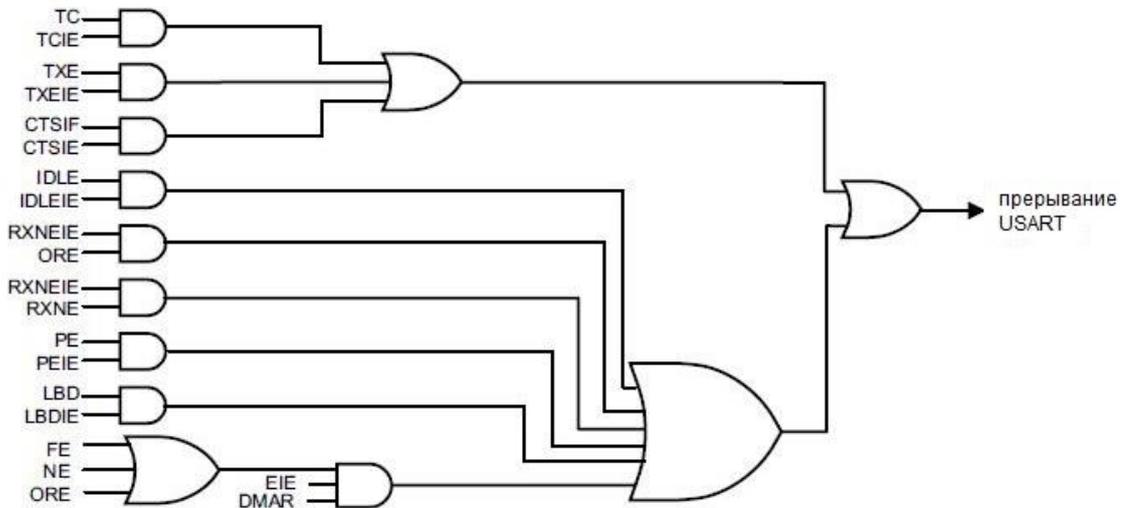


Рисунок 122 – Диаграмма отображений событий USART на прерывание

### 31.5 Регистры USART

Доступ к регистрам периферии осуществляется по 16-битным полусловам или 32-битными словами. Таблицы 543 – 549 показывают карту памяти регистров USART и их значения после сброса.

Таблица 542 – Регистры блока USART\_CNTR

Базовый Адрес		Название	Описание
0x500C_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

### 31.5.1 USART\_SR

Таблица 543 – Описание бит регистра USART\_SR

Base ADDR=		0x500C_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	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CTS	Res.	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						RW		R	RW	RW	R	R	R	R	R

Бит	Имя	Значение	Описание
31...10	Res.	0	Резерв
9	CTS	0	<p>Флаг CTS</p> <p>Данный бит устанавливается аппаратно, когда на входе CTS меняется уровень, если установлен бит CTSE. Бит сбрасывается программно записью 0. Прерывание генерируется, если CTSIE=1 в регистре USART_CR3.</p> <p>0: не было изменения статуса линии CTS 1: на линии CTS обнаружено изменение уровня</p>
8	Res.	0	Резерв
7	TXE	1	<p>Регистр данных передатчика пуст</p> <p>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.</p> <p>0: данные не были переданы в регистр сдвига 1: данные переданы в регистр сдвига</p> <p>Примечание – Данный бит используется во время передачи с использованием одного буфера.</p>
6	TC	1	<p>Передача завершена</p> <p>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 0. Данный сброс рекомендуется выполнять только при мультибуферном обмене.</p> <p>0: Передача не завершена; 1: Передача завершена</p>

Бит	Имя	Значение	Описание
5	RXNE	0	<p>Регистр данных приема не пуст</p> <p>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 0. Данная последовательность сброса рекомендована для мультибуферного обмена.</p> <p>0: данные не получены; 1: полученные данные готовы для чтения</p>
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 чтение 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. Допуск ухода тактовой частоты 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. 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 приёма.</p> <p>0: нет ошибки четности 1: обнаружена ошибка четности</p>

### 31.5.2 USART\_DR

Таблица 544 – Описание бит регистра USART\_DR

Base ADDR=		<b>0x500C_0000</b>				Offset=		<b>0x0000_0004</b>				Reset=		0xXXXX_XXXX		
REG Name:		USART_DR														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CTS	Res.	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						RW		R	RW	RW	R	R	R	R	R

Бит	Имя	Значение	Описание
31...9	Res.	0	Резерв
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>

### 31.5.3 USART\_BRR

Примечание – Счетчики скорости останавливают счет, если биты TE или RE запрещены, соответственно.

Таблица 545 – Описание бит регистра USART\_BRR

Base ADDR=		0x500C_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		
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		

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

Бит	Имя	Значение	Описание
31...16	Res.	0	Резерв
15...4	DIV_Mantissa [11:0]	0	Мантисса USARTDIV. Эти 12 бит определяют мантиссу делителя USART (USARTDIV)
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

### 31.5.4 USART\_CR1

Таблица 546 – Описание бит регистра USART\_CR1

Base ADDR=		0x500C_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	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Res.	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

Бит	Имя	Значение	Описание
31...16	Res.	0	Резерв
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	Res.	0	Резерв
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: пределитель 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)
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

Бит	Имя	Значение	Описание
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 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 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 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 1: Символ Break передается</p>

### 31.5.5 USART\_CR2

Таблица 547 – Описание бит регистра USART\_CR2

Base ADDR=		0x500C_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		
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	Res.	Res.	Res.	ADD [3:0]			
		RW	RW	RW	RW	RW	RW					RW	RW	RW	RW

Бит	Имя	Значение	Описание
31...14	Res.	0	Резерв
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	<p>Тактовый импульс последнего бита</p> <p>Бит позволяет выбрать, будет ли тактовый импульс последнего переданного бита (MSB) выдаваться на вывод СК в синхронном режиме.</p> <p>0: тактовый импульс последнего бита данных не выводится на вывод СК</p> <p>1: тактовый импульс последнего бита данных выводится на вывод СК</p> <p>Примечание – 1: последний бит это 8 или 9 бит данных, в зависимости от выбранного формата, задаваемого битом М в регистре USART_CR1.</p>
7...4	Res.	0	Резерв
3...0	ADD [3:0]	0	<p>адреса узла USART</p> <p>Данное битовое поле указывает адрес режима USART. Данные биты используются в многопроцессорном обмене во время режима молчания, для пробуждения приемопередатчика при обнаружении метки адреса.</p> <p>Примечание – Эти 3 бита (CPOL, CPHA, LBCL) не должны быть записаны, пока разрешен передатчик.</p>

### 31.5.6 USART\_CR3

Таблица 548 – Описание бит регистра USART\_CR3

Base ADDR=		<b>0x500C_0000</b>				Offset=		<b>0x0000_0014</b>				Reset=		0x0000_0000		
REG Name:		USART_CR3														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ONEBIT	CSTIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	Res.	Res.	EIE
				RW	RW	RW	RW	RW	RW	RW	RW	RW			RW

Бит	Имя	Значение	Описание
31...12	Res.	0	Резерв
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.
10	CSTIE	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.

Бит	Имя	Значение	Описание
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	Res.	0	Резерв
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

### 31.5.7 USART\_GTPR

Таблица 549 – Описание бит регистра USART\_GTPR

Base ADDR=		<b>0x500C_0000</b>				Offset=		<b>0x0000_0018</b>				Reset=		0x0000_0000		
REG Name:		USART_GTPR														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT [7:0]								Res.	Res.	Res.	PSC [4:0]				
RW	RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW

Бит	Имя	Значение	Описание
31...16	Res.	0	Резерв
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.

## 32 Электрические параметры микросхем

Таблица 550 – 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	$U_{OH}$	$U_{CC} - 0,6$	-	25, 85, - 50
Выходное напряжение низкого уровня, В, на выводах PA, PB, PC, PD	$U_{OL}$	-	0,4	
Ток утечки низкого уровня цифровых входов, мкА, – на выводах: PA, PB, PC, PD, nRESET, WAKEUP, JTAGEN, OSC_IN32, SAR_AIN; – на выводе OSC_IN	$I_{LL}$	- 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 JTAGEN; – на выводе OSC_IN	$I_{LH}$	- 1,0	1,0	
		10	80	
		1	20	
Динамический ток потребления, мА, при $f_c \leq 48$ МГц	$I_{CCO}$	-	35	
Выходная частота HSI RC-генератора, МГц, после выполнение тримминга	$f_{O\_HSI}$	7,9	8,1	
Выходная частота LSI RC-генератора, кГц, после выполнение тримминга	$f_{O\_LSI}$	30	32	
Выходная частота PLL, МГц: – максимальная; – минимальная	$f_{O\_PLL}$	48	-	
		-	8	
Параметры $\Delta \Sigma$ АЦП				
Соотношение сигнал/шум, дБ, усиление 0 дБ, $U_{ID\_ADCD} = 1$ В, $f_{NADCO}^{**} = 4$ кГц	$SNR_{D0}$	82	-	25, 85, - 50
* Значение уточняется в ходе ОКР. ** $f_{NADCO}$ – частота передискретизации на выходе микросхем				

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В\*.

### 33 Пределно-допустимые и предельные параметры

Таблица 551 – Пределно-допустимые электрические режимы эксплуатации и предельные электрические режимы микросхем

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение пара- метра	Норма параметра*			
		пределно- 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>	1,8	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 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	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	-	-
Частота следования импульсов тактовых сигналов LSE, кГц, на выводе OSC_IN32, при: – BYPASS = «0»; – BYPASS = «1»	f <sub>C_LSE</sub>	32	33	-	-
Емкость нагрузки, пФ, на выводах: PA, PB, PC, PD	C <sub>L</sub>	-	30	-	-
Параметры ΔΣ АЦП					
Опорное напряжение, В, на выводе VREF, при выключенном внутреннем источнике опорного напряжения	U <sub>REF_ADC</sub>	2,33	2,53	-	-
Входное напряжение дифференциальное, В, на выводах V0P – V2P, V0N – V2N	U <sub>ID_ADCDV</sub>	-1,0	1,0	-	-

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение пара- метра	Норма параметра*			
		предельно- допустимый режим		предельный режим	
		не менее	не более	не менее	не более
Входное напряжение дифференциальное, В, на выводах I0P – I3P, I0N – I3N	U <sub>ID_ADCDI</sub>	– 0,5	0,5	–	–
Напряжение, В, на выводах V0P – V2P, V0N – V2N	U <sub>L_ADCDV</sub>	– 0,5	0,5	– 0,8	U <sub>CC</sub> + 0,3
Напряжение, В, на выводах I0P – I3P, I0N – I3N	U <sub>L_ADCDI</sub>	– 0,25	0,25	– 0,8	U <sub>CC</sub> + 0,3
Частота следования импульсов тактовых сигналов $\Delta\Sigma$ АЦП, МГц	f <sub>C_ADCD</sub>	0,9	4,1	–	–
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	U <sub>AIN</sub>	0	U <sub>REF_ADC</sub>	– 0,3	3,9
на вывод VREF;					
- без использования опорного напряжения			U <sub>CCA</sub>		
Частота следования импульсов тактовых сигналов SAR АЦП, МГц	f <sub>C_ADCS</sub>	–	28	–	–
* Значение уточняется в ходе ОКР. Примечание – Не допускается одновременное задание двух предельных режимов					

### 34 Справочные параметры

Таблица 552 – 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>	-	0,6		25, 85, – 50
Статический ток потребления батареиногo домена, мкА	I <sub>CCB</sub>	-	0,5		
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>	-	-	48	
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)	$\frac{1}{2 \cdot f_{OP}}$	-	-	
	t <sub>PLZ</sub> (SSP_CLK- SSP_TXD)				
Время установления входных данных SSP_RXD относительно SSP_CLK, нс	t <sub>SU</sub> (SSP_RXD- SSP_CLK)	2,5	-	-	
Время удержания сигнала входных данных SSP_RXD относительно SSP_CLK, нс	t <sub>H</sub> (SSP_CLK- SSP_RXD)	3,5	-	-	
Время сохранения выходных данных SSP_TXD относительно сигнала SSP_CLK, нс	t <sub>V</sub> (SSP_CLK- SSP_TXD)	0	-	-	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра*			Температура среды, °C
		не менее	типовое	не более	
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	25, 85, - 50
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	$U_{REF0}$	-	2,45	-	25, 85, - 50
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					

### 35 Габаритный чертеж микросхемы

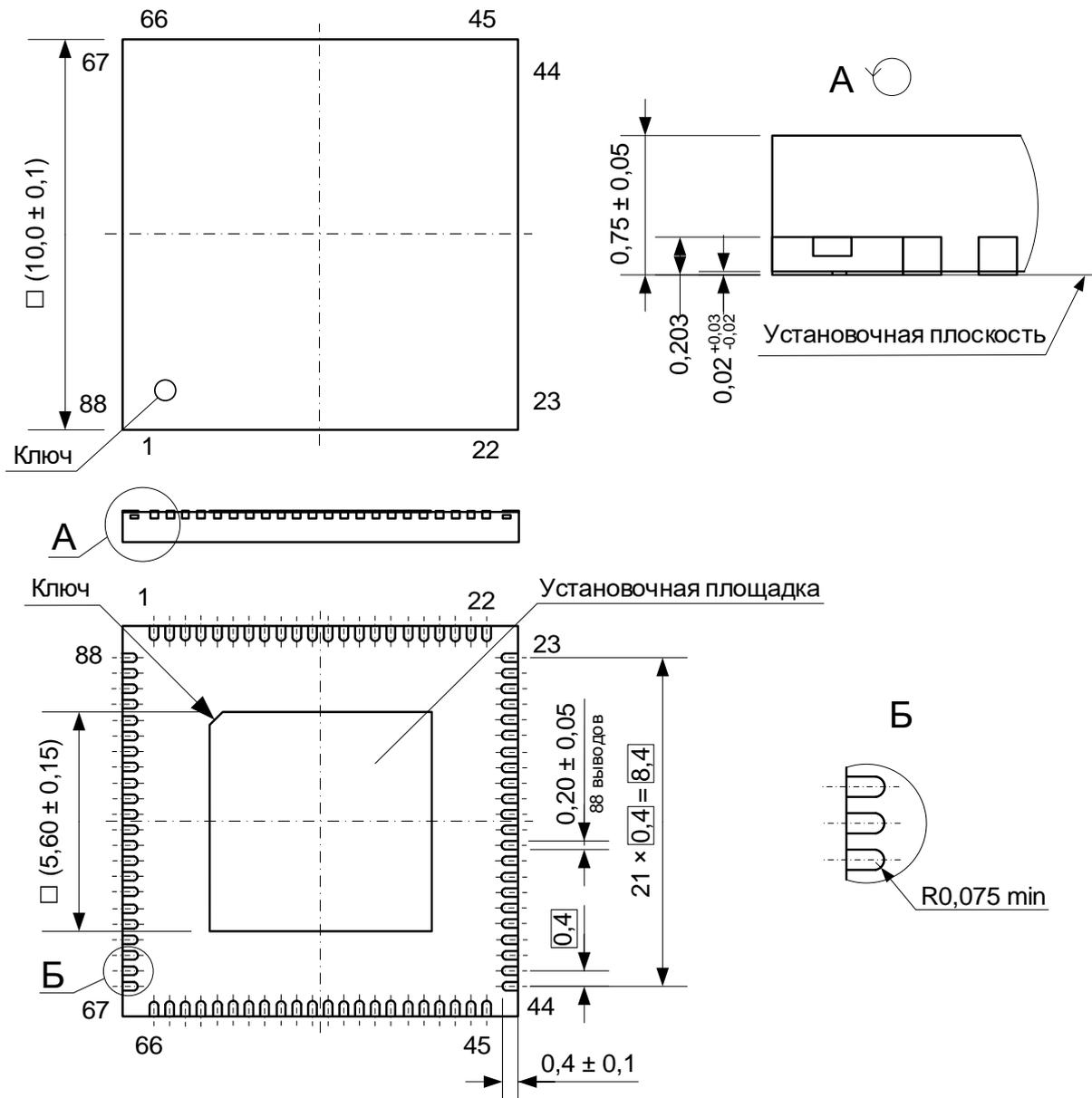


Рисунок 123 – Микросхема в корпусе QFN88

### 36 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон, °С
MDR1206FI	MDR1206FI RISC-V	QFN88 (TICP)	от – 50 до 85

