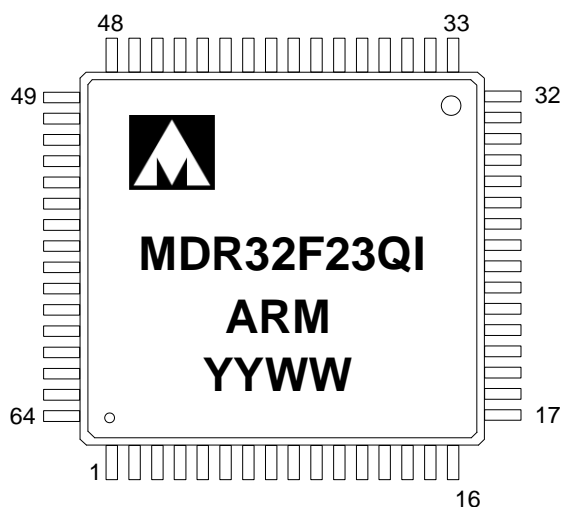




Микросхема контроллера для однофазного электросчетчика  
К1986ВК214



**Основные параметры микросхемы:**

- Напряжение источника питания от 2,2 до 3,6 В;
- 32-разрядная RISC-архитектура;
- Встроенная память программ 64 Кбайт;
- Встроенная память данных 16 Кбайт;
- Температурный диапазон от минус 40 °С до плюс 85 °С

YY – год выпуска

WW – неделя выпуска

**Тип корпуса:**

- 64-выводной пластмассовый корпус LQFP64.

Масса микросхем не более 3 г.

**Важно:** спецификация действительна совместно с документом К1986ВК214, К1986ВК234 Errata Notice.

## Основные характеристики

### Ядро:

- 32-битное RISC-ядро, тактовая частота до 36 МГц;
- Умножение за один цикл.

### Память:

- Встроенная энергонезависимая память программ FLASH типа размером 64 Кбайт;
- Встроенное ОЗУ размером 16 Кбайт.

### Питание и тактовая частота:

- Внешнее питание 2,2...3,6В;
- Встроенный регулятор напряжения на 1,8В для питания ядра;
- Встроенные схемы контроля питания;
- Встроенный домен с батарейным питанием;
- Встроенный подстраиваемый RC-генератор HSI 8 МГц;
- Встроенный подстраиваемый RC-генератор LSI 40 кГц;
- Внешний осциллятор HSE 2...16 МГц;
- Внешний осциллятор LSE 32 кГц;
- Встроенный умножитель тактовой частоты PLL для ядра.

### Режим пониженного энергопотребления:

- Режим SLEEP, DEEPSLEEP и Standby;
- Батарейный домен с часами реального времени и регистрами аварийного сохранения.

### Аналоговые модули:

- 24-разрядный  $\Delta\Sigma$ АЦП (до трех каналов);
- 12-разрядный АЦП (до восьми каналов) измеряемый диапазон от 0 до 3,6 В;
- Температурный датчик.

### Периферия:

- Контроллеры интерфейсов UART, SPI, I2C;
- Контроллер LCD-дисплея;
- До 47 пользовательских линий ввода-вывода;
- Два блока 16-разрядных таймеров;
- Два сторожевых таймера.

### Режим отладки:

- Последовательный отладочный интерфейс SWD.

## Содержание

1	Введение .....	14
2	Структурная схема.....	15
3	Условное графическое изображение .....	16
4	Описание выводов .....	17
5	Диаграмма расположения выводов в корпусах .....	26
6	Указания по применению и эксплуатации .....	27
7	Система питания .....	28
7.1	Структурная схема подачи питания.....	29
7.2	Схема сброса при включении и выключении основного питания .....	29
8	Организация памяти .....	31
8.1	Структурная схема.....	31
8.2	Базовые адреса процессора.....	32
9	Загрузочное ПЗУ и режимы работы микросхемы .....	33
9.1	UART-загрузчик.....	34
9.1.1	Параметры связи по UART.....	34
9.1.2	Протокол обмена по UART .....	34
9.1.3	Синхронизация с внешним устройством .....	35
9.1.4	Команда CMD_SYNC .....	35
9.1.5	Команда CMD_CR.....	35
9.1.6	Команда CMD_BAUD.....	36
9.1.7	Команда CMD_LOAD .....	36
9.1.8	Команда CMD_VFY .....	37
9.1.9	Команда CMD_RUN.....	37
9.1.10	Прием параметров команды .....	38
9.1.11	Сообщения об ошибках .....	38
9.1.12	Ошибка ERR_CHN .....	38
9.1.13	Ошибка ERR_CMD .....	38
9.1.14	Ошибка ERR_BAUD .....	38
10	Контроллер FLASH-памяти программ .....	39
10.1	Работа Flash-памяти программ в обычном режиме.....	39
10.2	Работа Flash-памяти программ в режиме программирования .....	40
10.2.1	Стирание блока памяти размером 2 Кбайт или 32 Кбайт .....	40
10.2.2	Стирание страницы памяти размером 512 байт .....	41
10.2.3	Запись 32-битного слова в память .....	42
10.2.4	Чтение 32-битного слова из памяти. ....	42
10.3	Описание регистров управления контроллера Flash-памяти программ.....	43
10.3.1	EEPROM_CMD.....	43
10.3.2	EEPROM_ADR .....	45
10.3.3	EEPROM_DI.....	45
10.3.4	EEPROM_DO .....	45
10.3.5	EEPROM_KEY .....	46
11	Процессорное ядро .....	47

11.1	Структурная схема процессора .....	48
11.2	Программная модель .....	49
11.3	Стек .....	49
11.4	Регистры ядра.....	50
11.4.1	Регистры общего назначения R0-R12 .....	50
11.4.2	Указать стека SP R13 .....	50
11.4.3	Регистр связи LR R14.....	50
11.4.4	Счетчик команд PC R15.....	51
11.4.5	Программный регистр состояния PSR .....	51
11.4.6	APSR .....	51
11.4.7	IPSR.....	52
11.4.8	EPSR.....	52
11.4.9	PRIMASK .....	53
11.4.10	CONTROL .....	53
11.5	Типы данных .....	54
12	Система команд.....	55
12.1	Встроенные функции.....	57
12.2	Описание инструкций .....	58
12.2.1	Операнды.....	58
12.2.2	Ограничения на использование PC и SP .....	58
12.3	Операции сдвига .....	58
12.3.1	ASR .....	59
12.3.2	LSR.....	59
12.3.3	LSL .....	60
12.3.4	ROR.....	60
12.4	Выравнивание адресов .....	61
12.5	Адресация относительно счетчика команд PC .....	61
12.6	Условное исполнение .....	61
12.6.1	Флаги условий .....	62
12.6.2	Суффиксы условного исполнения .....	62
12.7	Команды доступа к памяти.....	63
12.7.1	ADR.....	63
12.8	LDR и STR, непосредственно заданное смещение .....	64
12.8.1	LDR и STR, смещение задано в регистре .....	65
12.8.2	LDR, адресация относительно счетчика команд PC .....	66
12.8.3	LDM и STM.....	66
12.8.4	PUSH и POP .....	68
12.9	Инструкции обработки данных .....	69
12.9.1	ADD, ADC, SUB, SBC и RSB.....	70
12.9.2	AND, ORR, EOR, BIC.....	72
12.9.3	ASR, LSL, LSR, ROR.....	73
12.9.4	CMR и CMN .....	74
12.9.5	MOV и MVN .....	75



12.9.6	MULS .....	76
12.9.7	REV, REV16, REVSH .....	77
12.9.8	SXT и UXT .....	77
12.9.9	TST .....	78
12.10	Инструкции передачи управления .....	79
12.10.1	B, BL, BX и BLX .....	79
12.11	Прочие инструкции .....	81
12.11.1	BKPT .....	81
12.11.2	CPS .....	82
12.11.3	DMB .....	82
12.11.4	DSB .....	83
12.11.5	ISB .....	83
12.11.6	MRS .....	84
12.11.7	MSR .....	84
12.11.8	NOP .....	85
12.11.9	SEV .....	85
12.11.10	SVC .....	86
12.11.11	WFE .....	86
12.11.12	WFI .....	87
13	Блок АЦП для измерения напряжений и токов в электрической сети .....	88
13.1	Описание регистров управления блока трехканального $\Delta\Sigma$ АЦП .....	89
13.1.1	ADCUI_CTRL1 .....	90
13.1.2	ADCUI_CTRL2 .....	93
13.1.3	ADCUI_CTRL3 .....	93
13.1.4	ADCUI_F0CTR .....	93
13.1.5	ADCUI_F0WC .....	94
13.1.6	ADCUI_F0WATTP .....	95
13.1.7	ADCUI_F0WATTN .....	95
13.1.8	ADCUI_F0VC .....	95
13.1.9	ADCUI_F0VARP .....	96
13.1.10	ADCUI_F0VARN .....	96
13.1.11	ADCUI_F0AC .....	96
13.1.12	ADCUI_F0VR .....	97
13.1.13	ADCUI_F0MD0 .....	97
13.1.14	ADCUI_F0MD1 .....	98
13.1.15	ADCUI_F0VPEAK .....	99
13.1.16	ADCUI_F0IPEAK .....	99
13.1.17	ADCUI_F0VDAT .....	99
13.1.18	ADCUI_F0I0DAT .....	100
13.1.19	ADCUI_F0I3DAT .....	100
13.1.20	ADCUI_F0VRMS .....	100
13.1.21	ADCUI_F0VRMS2 .....	101
13.1.22	ADCUI_F0IRMS .....	101

13.1.23	ADCUI_F0IRMS2 .....	101
13.1.24	ADCUI_F0STAT .....	102
13.1.25	ADCUI_F0MASK.....	103
13.1.26	ADCUI_CCAL1 .....	105
13.1.27	ADCUI_CCAL4.....	105
13.2	Алгоритмы вычисления окончательных результатов и их соответствия внешним сигналам.....	105
13.2.1	Типовая схема включения для учета электроэнергии по одной фазе .....	111
13.2.2	Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта.....	113
14	Сигналы тактовой частоты .....	114
14.1	Структурная схема.....	114
14.2	Встроенный RC-Генератор HSI.....	114
14.3	Встроенный RC-генератор LSI.....	115
14.4	Внешний осциллятор HSE .....	115
14.5	Внешний осциллятор LSE.....	115
14.6	Встроенный блок умножения системной тактовой частоты.....	115
14.7	Описание регистров блока контроллера тактовой частоты .....	116
14.7.1	CLOCK_STATUS .....	116
14.7.2	PLL_CONTROL .....	117
14.7.3	HS_CONTROL .....	117
14.7.4	CPU_CLOCK.....	118
14.7.5	PER1_CLOCK .....	119
14.7.6	ADC_CLOCK .....	120
14.7.7	RTC_CLOCK.....	121
14.7.8	PER2_CLOCK .....	121
14.7.9	TIM_CLOCK .....	122
14.7.10	UART_CLOCK.....	123
14.7.11	SSP_CLOCK.....	124
15	Батарейный домен и часы реального времени.....	125
15.1	Часы реального времени .....	125
15.2	Регистры аварийного сохранения .....	126
15.3	Описание регистров блока батарейного домена.....	126
15.3.1	BKP_REG_00 .....	127
15.3.2	BKP_REG_01 .....	127
15.3.3	BKP_REG_02 .....	127
15.3.4	BKP_REG_03 .....	127
15.3.5	BKP_REG_04 .....	127
15.3.6	BKP_REG_05 .....	127
15.3.7	BKP_REG_06 .....	127
15.3.8	BKP_REG_07 .....	127
15.3.9	BKP_REG_08 .....	127

15.3.10	BKP_REG_09 .....	127
15.3.11	BKP_REG_0A .....	127
15.3.12	BKP_REG_0B.....	127
15.3.13	BKP_REG_0C.....	127
15.3.14	BKP_REG_0D .....	127
15.3.15	BKP_REG_0E.....	128
15.3.16	BKP_REG_0F .....	129
15.3.17	RTC_CNT .....	131
15.3.18	RTC_DIV .....	132
15.3.19	RTC_PRL .....	132
15.3.20	RTC_ALRM.....	132
15.3.21	RTC_CS.....	133
16	Порты ввода-вывода .....	134
16.1	Описание регистров портов ввода-вывода.....	136
16.1.1	PORTx_RXTX.....	137
16.1.2	PORTx_OE.....	137
16.1.3	PORTx_FUNC .....	137
16.1.4	PORTx_ANALOG.....	138
16.1.5	PORTx_PULL .....	138
16.1.6	PORTx_PD.....	138
16.1.7	PORTx_PWR .....	139
16.1.8	PORTx_GFEN .....	139
16.1.9	PORTx_SETTX .....	140
16.1.10	PORTx_CLRTX.....	140
17	Детектор напряжения питания .....	141
17.1	Описание регистров блока PVD.....	142
17.1.1	PVDCS .....	142
18	Таймеры общего назначения .....	145
18.1	Основные характеристики .....	145
18.1.1	Структурная схема .....	146
18.2	Базовый блок таймера .....	146
18.2.1	Инициализация тактирования таймера .....	147
18.2.2	Инициализация основного счетчика таймера.....	147
18.2.3	Режимы счета .....	148
18.2.4	Тактовая частота $F_{DTS}$ .....	150
18.3	Источники событий для счета .....	151
18.3.1	Внутренний тактовый сигнал (TIM_CLKd).....	152
18.3.2	Событие в другом таймере (CNT==ARR).....	153
18.3.3	Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CH1i .....	154
18.3.4	Внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR.....	155
18.4	Режим захвата .....	156

18.5	Режим ШИМ.....	157
18.5.1	Генератор опорного сигнала REF .....	158
18.5.2	Генератор «мертвой зоны» .....	159
18.5.3	Выходные блоки .....	160
18.6	Блок цифрового фильтра.....	160
18.7	Флаги состояний и прерывания.....	162
18.7.1	Флаги состояний.....	162
18.7.2	Прерывания .....	163
18.8	Примеры .....	163
18.8.1	Обычный счетчик .....	163
18.8.2	Режим захвата .....	163
18.8.3	Режим ШИМ .....	165
18.9	Описание регистров блока таймера .....	166
18.9.1	CNT .....	167
18.9.2	PSG.....	167
18.9.3	ARR.....	167
18.9.4	CNTRL .....	168
18.9.5	CCR1 .....	169
18.9.6	CCR11 .....	169
18.9.7	CH1_CNTRL .....	170
18.9.8	CH1_CNTRL1 .....	172
18.9.9	CH1_CNTRL2 .....	173
18.9.10	CH1_DTG .....	174
18.9.11	BRKETR_CNTRL .....	174
18.9.12	STATUS .....	175
18.9.13	IE .....	177
19	Контроллер АЦП.....	179
19.1	Преобразование внешнего канала.....	180
19.2	Последовательное преобразование нескольких каналов .....	180
19.3	Преобразование с контролем границ .....	181
19.4	Датчик опорного напряжения.....	181
19.5	Датчик температуры.....	182
19.6	Время заряда внутренней емкости.....	183
19.7	Описание регистров блока контроллера АЦП.....	184
19.7.1	ADCx_CFG .....	184
19.7.2	ADC1_H_LEVEL .....	187
19.7.3	ADC1_L_LEVEL.....	187
19.7.4	ADC1_RESULT.....	187
19.7.5	ADC1_STATUS.....	188
19.7.6	ADC1_CHSEL .....	189
19.7.7	ADC1_TRIM.....	189
20	Контроллер интерфейса I2C .....	190
20.1	Конфигурация системы.....	190

20.2	Протокол I2C.....	190
20.3	Сигнал START .....	191
20.4	Передача адреса .....	191
20.5	Передача данных.....	191
20.6	Сигнал STOP .....	192
20.7	Описание регистров контроллера I2C .....	192
20.7.1	I2C->PRL .....	192
20.7.2	I2C->PRH.....	193
20.7.3	I2C->CTR.....	193
20.7.4	I2C->RXD .....	193
20.7.5	I2C->STA .....	194
20.7.6	I2C->TXD .....	194
20.7.7	I2C->CMD .....	195
21	Контроллер SSP.....	196
21.1	Основные характеристики модуля SSP .....	196
21.2	Программируемые параметры.....	197
21.3	Характеристики интерфейса SPI .....	197
21.4	Характеристики интерфейса Microwire .....	198
21.5	Характеристики интерфейса SSI .....	198
21.6	Общий обзор модуля SSP .....	198
21.6.1	Блок формирования тактового сигнала.....	199
21.6.2	Буфер FIFO передатчика.....	199
21.6.3	Буфер FIFO приемника .....	199
21.6.4	Блок приема и передачи данных.....	199
21.6.5	Блок формирования прерываний .....	200
21.6.6	Конфигурирование приемопередатчика .....	200
21.6.7	Разрешение работы приемопередатчика.....	200
21.6.8	Соотношения между тактовыми сигналами .....	201
21.6.9	Программирование регистра управления SSPCR0 .....	202
21.6.10	Программирование регистра управления SSPCR1 .....	202
21.6.11	Формирование тактового сигнала обмена данными.....	202
21.6.12	Формат информационного кадра.....	203
21.6.13	Формат синхронного обмена SSI фирмы Texas Instruments .....	204
21.6.14	Формат синхронного обмена SPI фирмы Motorola.....	204
21.6.15	Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0. 205	
21.6.16	Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1. 206	
21.6.17	Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0. 207	
21.6.18	Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1. 208	
21.6.19	Формат синхронного обмена Microwire фирмы National Semiconductor .....	209
21.6.20	Примеры конфигурации модуля в ведущем и ведомом режимах....	211
21.7	Прерывания .....	213
21.7.1	Сигналы запроса на прерывание:.....	213

21.7.2	SSPRXINTR.....	214
21.7.3	SSPTXINTR.....	214
21.7.4	SSPRORINTR.....	214
21.7.5	SSPRTINTR.....	214
21.7.6	SSPINTR.....	215
21.8	Программное управление модулем. Общая информация.....	215
21.8.1	Описание регистров контроллера SSP.....	215
22	Контроллер UART.....	221
22.1	Основные характеристики модуля UART.....	221
22.2	Программное управление скоростью обмена.....	221
22.3	Программируемые параметры.....	222
22.4	Отличительные особенности контроллера UART.....	222
22.5	Функциональные возможности.....	223
22.6	Модуль приемопередатчика:.....	223
22.7	Описание функционирования блока UART.....	224
22.7.1	Генератор тактового сигнала приемопередатчика.....	225
22.7.2	Буфер FIFO передатчика.....	225
22.7.3	Буфер FIFO приемника.....	225
22.7.4	Блок передатчика.....	225
22.7.5	Блок приемника.....	225
22.7.6	Блок формирования прерываний.....	226
22.7.7	Блок и регистры синхронизации.....	226
22.8	Описание функционирования ИК-кодека IrDA SIR.....	226
22.8.1	Кодер ИК-передатчика.....	227
22.8.2	Декодер ИК-приемника.....	227
22.9	Описание работы UART.....	228
22.9.1	Сброс модуля.....	228
22.9.2	Тактовые сигналы.....	228
22.9.3	Работа универсального асинхронного приемопередатчика.....	228
22.9.4	Коэффициент деления частоты.....	229
22.9.5	Передача и прием данных.....	229
22.9.6	Биты ошибки.....	230
22.9.7	Бит переполнения буфера.....	230
22.9.8	Запрет буфера FIFO.....	231
22.9.9	Проверка по шлейфу.....	231
22.9.10	Работа кодека ИК-обмена данными IrDA SIR.....	231
22.9.11	Проверка по шлейфу.....	232
22.10	Линии управления модемом.....	233
22.10.1	Аппаратное управление потоком данных.....	233
22.10.2	Управление потоком данных по линии RTS.....	234
22.10.3	Управление потоком данных по линии CTS.....	234
22.11	Прерывания.....	235
22.11.1	UARTMSINTR.....	236

22.11.2	UARTRXINTR .....	236
22.11.3	UARTTXINTR.....	236
22.11.4	UARTRTINTR.....	237
22.11.5	UARTEINTR .....	237
22.11.6	UARTINTR.....	237
22.12	Программное управление модулем.....	237
22.12.1	Общая информация .....	237
22.13	Обобщенные данные о регистрах контроллеров UART.....	238
22.13.1	DR.....	239
22.13.2	RSR_ECR.....	240
22.13.3	FR .....	241
22.13.4	ILPR.....	242
22.13.5	IBRD.....	243
22.13.6	FBRD.....	243
22.13.7	LCR_H.....	245
22.13.8	CR.....	246
22.13.9	IFLS.....	248
22.13.10	IMSC .....	249
22.13.11	RIS .....	250
22.13.12	MIS .....	251
22.13.13	ICR.....	252
22.13.14	UARTTCR .....	252
23	Контроллер LCD .....	253
23.1	Организация памяти данных LCD .....	254
23.2	Описание регистров контроллера LCD .....	257
23.2.1	CR. Регистр управления.....	257
23.2.2	ROW4-ROW1 .....	258
23.2.3	AROW4-AROW1 .....	258
24	Прерывания и исключения .....	259
24.1	Типы исключений .....	259
24.1.1	RESET .....	259
24.1.2	NON MASKABLE INTERRUPT (NMI).....	259
24.1.3	HARD FAULT .....	259
24.1.4	SVCALL.....	259
24.1.5	PendSV .....	260
24.1.6	SysTick .....	260
24.2	Прерывания (IRQ).....	260
24.3	Обработчики исключений.....	261
24.4	Приоритеты исключений .....	262
24.5	Вход в обработчик и выход из обработчика .....	263
24.5.1	Приоритетное прерывание .....	263
24.5.2	Возврат.....	263
24.5.3	Передача управления без восстановления контекста (tail-chaining) .....	263

24.5.4	Запоздавшее исключение (late-arriving exception) .....	263
24.5.5	Вход в процедуру обработки исключения .....	264
24.5.6	Возврат из обработчика исключения .....	265
25	Управление электропитанием .....	266
25.1	Переход в режим пониженного энергопотребления .....	266
25.2	Ожидание прерывания .....	266
25.3	Переход в режим ожидания по выходу из обработчика исключения (режим sleep-on-exit) .....	266
25.4	Выход из состояния ожидания .....	266
25.5	Выход из ожидания по команде WFE.....	267
25.6	Рекомендации по программированию режима энергопотребления .....	267
26	Контроллер прерываний NVIC.....	268
26.1	Логика работы прерываний контроллера NVIC.....	268
26.2	Регистр разрешения прерываний .....	271
26.3	Регистр запрета прерываний.....	272
26.4	Регистр установки состояния ожидания для прерывания .....	272
26.5	Регистр сброса состояния ожидания для прерывания .....	273
26.6	Регистры приоритета прерываний .....	273
26.7	Прерывания, срабатывающие по уровню сигнала .....	274
26.8	Аппаратное и программное управление прерываниями.....	274
26.9	Рекомендации по работе с контроллером прерываний.....	275
27	Блок управления системой ядра .....	276
27.1	Упрощенный доступ к регистрам блока управления системой из среды разработки программного обеспечения.....	276
27.1.1	Регистр идентификации процессора .....	276
27.1.2	Регистр управления прерываниями .....	277
27.1.3	Регистр управления прерываниями и программного сброса .....	278
27.1.4	Регистр управления системой .....	279
27.1.5	Регистр конфигурации и управления .....	280
27.1.6	Регистры приоритета системных обработчиков .....	280
27.1.7	Регистр №2 приоритета системных обработчиков .....	281
27.1.8	Регистр №3 приоритета системных обработчиков .....	281
27.1.9	Рекомендации по программированию блока управления системой .....	281
28	Сторожевые таймеры.....	282
28.1	Блок сторожевого таймера IWDG .....	282
28.1.1	Описание регистров блока сторожевого таймера IWDG .....	283
28.2	Блок сторожевого таймера WWDG .....	287
28.2.1	Описание регистров блока сторожевого таймера WWDG.....	287
29	Типовая схема подключения питания .....	289
30	Предельно-допустимые характеристики микросхемы .....	290
31	Электрические параметры микросхемы .....	292
32	Справочные данные .....	294
33	Габаритный чертеж микросхемы .....	296



34 Информация для заказа ..... 297

## 1 Введение

Микросхемы интегральные K1986BK214 (далее – микросхемы) предназначены для использования в приборах учета электроэнергии в однофазных (~220 В/50 Гц) сетях.

Микросхемы K1986BK214 являются микроконтроллерами со встроенной FLASH-памятью программ и построены на базе низкопотребляемого процессорного RISC-ядра. Микросхемы работают на тактовой частоте до 36 МГц и содержит 64 Кбайт FLASH-памяти и 16 Кбайт ОЗУ. Микросхемы включают в себя развитую периферию для построения счетчиков электроэнергии однофазных сетей. Периферия включает в себя три канала для однофазной сети 24-битных независимых  $\Delta\Sigma$ АЦП. Каждый канал  $\Delta\Sigma$ АЦП имеет предусилитель, фазовую подстройку (для коррекции фазы не хуже 0,1), а также аппаратный блок для вычисления среднеквадратического значения сигнала. Каждый канал  $\Delta\Sigma$ АЦП может быть включен или отключен независимо от других каналов. Еще один дополнительный 12-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 микросхем входят два интерфейса UART, интерфейс I2C и интерфейс SPI. Микросхемы содержат два 16-разрядных таймера, а также системный 24-разрядный таймер и два сторожевых таймера.

Встроенные RC-генераторы HSI (8 МГц) и LSI (40 кГц) и внешние генераторы HSE (2...16 МГц) и LSE (32 кГц) и схема умножения тактовой частоты PLL для ядра позволяют гибко настраивать скорость работы микросхем.

Архитектура системой шины за счет регулировки частоты периферийных блоков позволяет уменьшить потребление всей системы.

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

## 2 Структурная схема

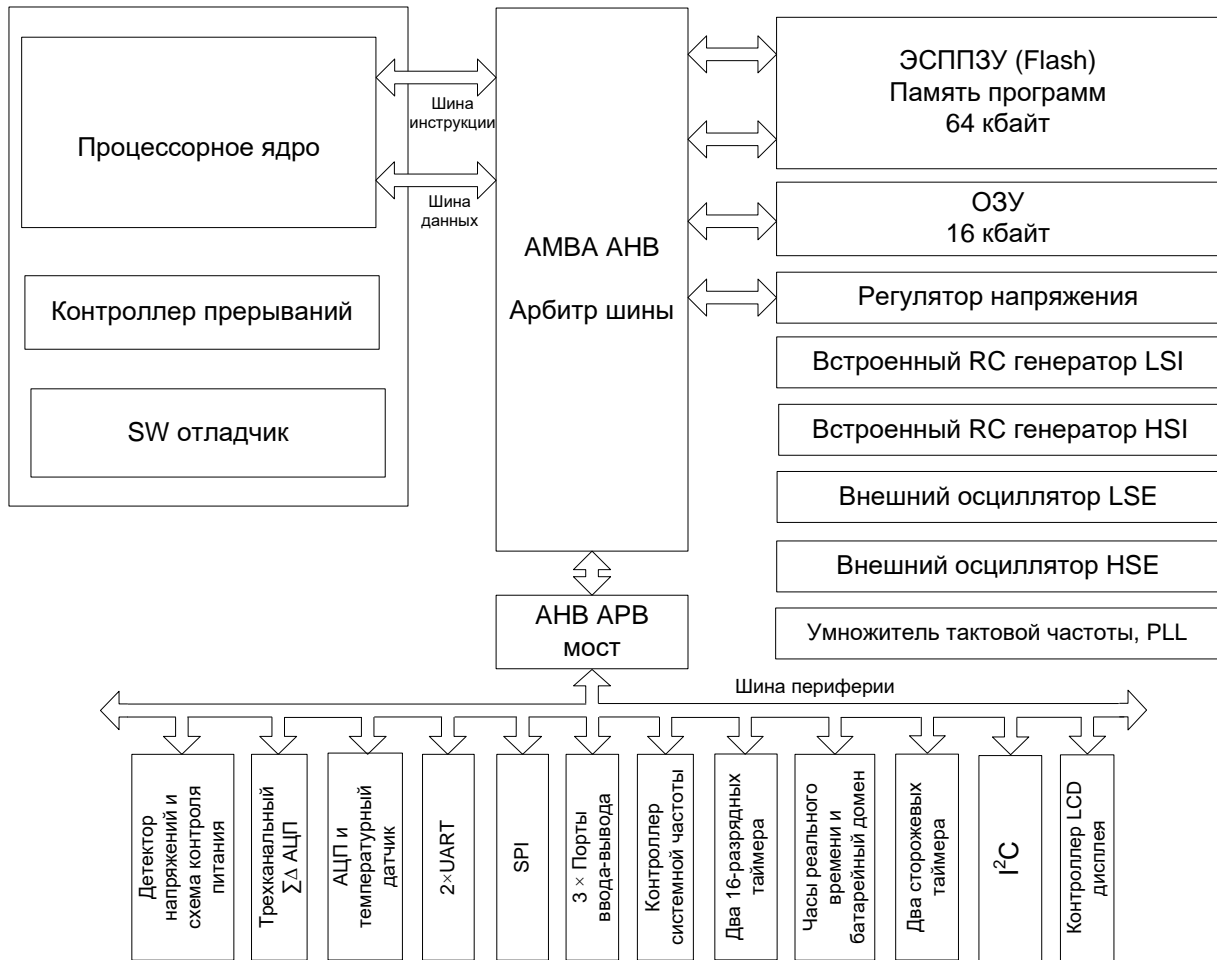


Рисунок 1 – Структурная схема микросхемы

### 3 Условное графическое изображение

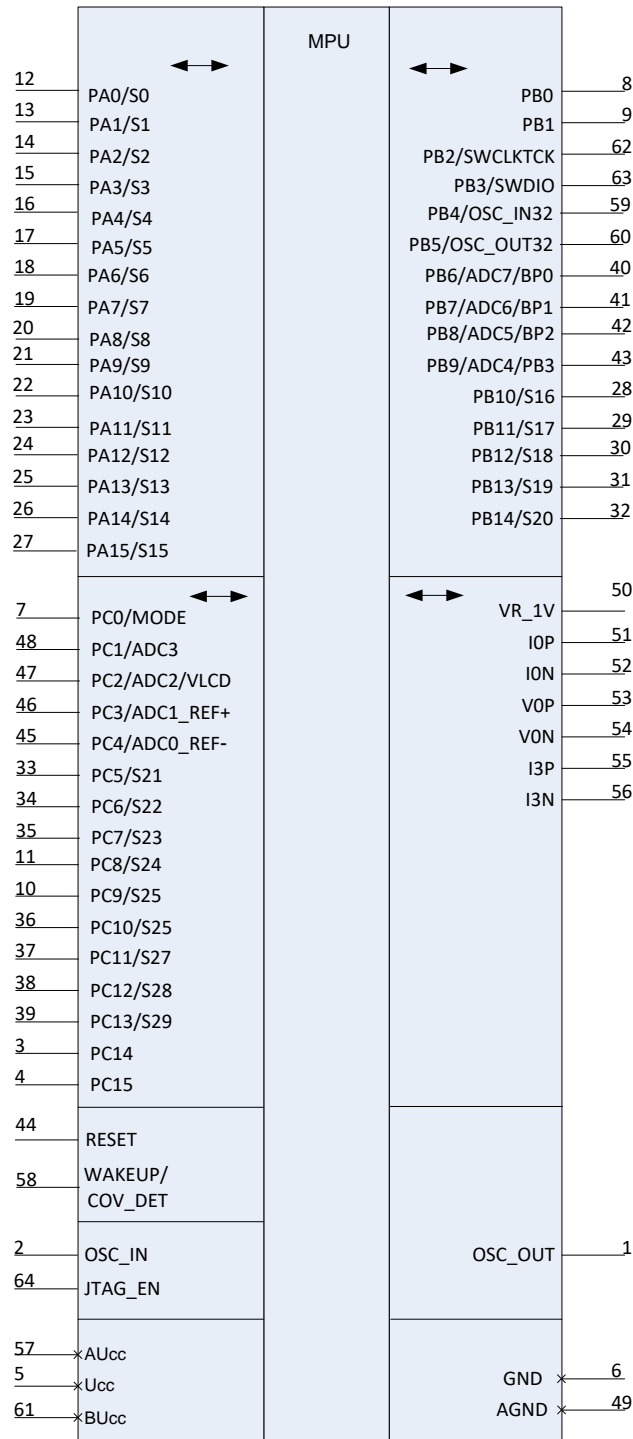


Рисунок 2 – Условное графическое изображение

## 4 Описание выводов

Таблица 1 – Описание выводов

Номер вывода	Обозначение вывода	Функциональное назначение вывода
1	OSC_OUT	Выход генератора HSE
2	OSC_IN	Вход генератора HSE
3	PC14	Вход/выход 14 порта C
4	PC15	Вход/выход 15 порта C
5	U <sub>cc</sub>	Питание (2,2...3,6) В
6	GND	Общий
7	PC0/MODE	Вход/выход 0 порта C
8	PB0/MODE0	Вход/выход 0 порта B
9	PB1	Вход/выход 1 порта B
10	PC9	Вход/выход 9 порта C
11	PC8	Вход/выход 8 порта C
12	PA0	Вход/выход 0 порта A
13	PA1	Вход/выход 1 порта A
14	PA2	Вход/выход 2 порта A
15	PA3	Вход/выход 3 порта A
16	PA4	Вход/выход 4 порта A
17	PA5	Вход/выход 5 порта A
18	PA6	Вход/выход 6 порта A
19	PA7	Вход/выход 7 порта A
20	PA8	Вход/выход 8 порта A
21	PA9	Вход/выход 9 порта A
22	PA10	Вход/выход 10 порта A
23	PA11	Вход/выход 11 порта A
24	PA12	Вход/выход 12 порта A
25	PA13	Вход/выход 13 порта A
26	PA14	Вход/выход 14 порта A
27	PA15	Вход/выход 15 порта A
28	PB10	Вход/выход 10 порта B
29	PB11	Вход/выход 11 порта B
30	PB12	Вход/выход 12 порта B
31	PB13	Вход/выход 13 порта B
32	PB14	Вход/выход 14 порта B
33	PC5	Вход/выход 4 порта C
34	PC6	Вход/выход 5 порта C
35	PC7	Вход/выход 6 порта C
36	PC10	Вход/выход 10 порта C
37	PC11	Вход/выход 11 порта C
38	PC12	Вход/выход 12 порта C
39	PC13	Вход/выход 13 порта C
40	PB6	Вход/выход 6 порта B

Номер вывода	Обозначение вывода	Функциональное назначение вывода
41	PB7	Вход/выход 7 порта В
42	PB8	Вход/выход 8 порта В
43	PB9	Вход/выход 9 порта В
44	RESET	Сигнал внешнего сброса
45	PC4	Вход/выход 4 порта С
46	PC3	Вход/выход 3 порта С
47	PC2	Вход/выход 2 порта С
48	PC1	Вход/выход 1 порта С
49	AGND	Общий АЦП, ЦАП
50	VR_1V	Вход внешнего опорного напряжения 1 В
51	I0P	Положительный вход канала 0 тока
52	I0N	Отрицательный вход канала 0 тока
53	V0P	Положительный вход канала 0 напряжения
54	V0N	Отрицательный вход канала 0 напряжения
55	I3P	Положительный вход канала 3 тока
56	I3N	Отрицательный вход канала 3 тока
57	AU <sub>CC</sub>	Аналоговое питание $\Delta\Sigma$ АЦП (2,4...3,6) В
58	WAKEUP/COV_DET	Сигнал внешнего выхода из режима Standby
59	PB4	Вход/выход 4 порта В
60	PB5	Вход/выход 5 порта В
61	BU <sub>CC</sub>	Батарейное питание (1,8...3,6) В
62	PB2/SWCLKTCK	Вход/выход 2 порта В
63	PB3/SWDIO	Вход/выход 3 порта В
64	JTAG_EN	Тестовый вывод. Рекомендуется не подсоединять или подключать к шине общий

Таблица 2 – Назначение выводов по блокам

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
Порт А				
12	РА0	Вход/выход 0 порта А		
		Аналоговая	S0	Выход 0 управления сегментами LCD-дисплея
		Основная	TMR0_CH1	Вход/выход канала 1 таймера 0
		Альтернативная	–	–
13	РА1	Вход/выход 1 порта А		
		Аналоговая	S1	Выход 1 управления сегментами LCD-дисплея
		Основная	TMR0_CH1N	Инверсный выход канала 1 таймера 0
		Альтернативная	–	–
14	РА2	Вход/выход 2 порта А		
		Аналоговая	S2	Выход 2 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
15	РА3	Вход/выход 3 порта А		
		Аналоговая	S3	Выход 3 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
16	РА4	Вход/выход 4 порта А		
		Аналоговая	S4	Выход 4 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
17	РА5	Вход/выход 5 порта А		
		Аналоговая	S5	Выход 5 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
18	РА6	Вход/выход 6 порта А		
		Аналоговая	S6	Выход 6 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
19	РА7	Вход/выход 7 порта А		
		Аналоговая	S7	Выход 7 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
20	PA8	Вход/выход 8 порта A		
		Аналоговая	S8	Выход 8 управления сегментами LCD-дисплея
		Основная	TMR0_ETR	Вход внешнего события таймера 0
		Альтернативная	–	–
21	PA9	Вход/выход 9 порта A		
		Аналоговая	S9	Выход 9 управления сегментами LCD-дисплея
		Основная	TMR0_BRK	Вход внешнего события таймера 0
		Альтернативная	–	–
22	PA10	Вход/выход 10 порта A		
		Аналоговая	S10	Выход 10 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
23	PA11	Вход/выход 11 порта A		
		Аналоговая	S11	Выход 11 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
24	PA12	Вход/выход 12 порта A		
		Аналоговая	S12	Выход 12 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
25	PA13	Вход/выход 13 порта A		
		Аналоговая	S13	Выход 13 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
26	PA14	Вход/выход 14 порта A		
		Аналоговая	S14	Выход 14 управления сегментами LCD дисплея
		Основная	–	–
		Альтернативная	–	–
27	PA15	Вход/выход 15 порта A		
		Аналоговая	S15	Выход 15 управления сегментами LCD дисплея
		Основная	–	–
		Альтернативная	–	–



Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
Порт В				
8	PB0/ MODE0	Вход/выход 0 порта В		
		Аналоговая	–	–
		Основная	UART0_TXD	Выходные данные UART0
		Альтернативная	–	–
9	PB1	Вход/выход 1 порта В		
		Аналоговая	–	–
		Основная	UART0_RXD	Входные данные UART0
		Альтернативная	–	–
62	PB2/ SWCLKTCK	Вход/выход 2 порта В		
		Аналоговая	–	–
		Основная	nSIROUT0	Выходные данные IRDA SIR UART0
		Альтернативная	–	–
63	PB3/ SWDIO	Вход/выход 3 порта В		
		Аналоговая	–	–
		Основная	nSIRIN0	Входные данные IRDA SIR UART0
		Альтернативная	–	–
59	PB4	Вход/выход 4 порта В		
		Аналоговая	OSC_IN32	Аналоговый вход тактовой частоты 32 кГц осциллятора LSE
		Основная	nUART0DTR	Выход готовности для передачи данных UART0
		Альтернативная	–	–
60	PB5	Вход/выход 5 порта В		
		Аналоговая	OSC_OUT32	Аналоговый выход осциллятора LSE
		Основная	nUART0RTS	Выход запроса на передачу данных UART0
		Альтернативная	–	–
40	PB6	Вход/выход 6 порта В		
		Аналоговая	ADC7/BP0	Аналоговый вход канала 7 АЦП последовательных приближений/общий выход управления LCD-дисплеем
		Основная	nUART0RI	Вход сигнала вызова от источника
		Альтернативная	–	–
41	PB7	Вход/выход 7 порта В		
		Аналоговая	ADC6/BP1	Аналоговый вход канала 6 АЦП последовательных приближений/общий выход управления LCD-дисплеем
		Основная	nUART0DCD	Вход информационного сигнала от источника
		Альтернативная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
42	PB8	Вход/выход 8 порта В		
		Аналоговая	ADC5/BP2	Аналоговый вход канала 5 АЦП последовательных приближений/ общий выход управления LCD-дисплеем
		Основная	nUART0DSR	Вход готовности для передачи данных источником
		Альтернативная	–	–
43	PB9	Вход/выход 9 порта В		
		Аналоговая	ADC4/BP3	Аналоговый вход канала 4 АЦП последовательных приближений/ общий выход управления LCD-дисплеем
		Основная	nUART0CTS	Вход готовности для приема данных источником
		Альтернативная	–	–
28	PB10	Вход/выход 10 порта В		
		Аналоговая	S16	Выход 16 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
29	PB11	Вход/выход 11 порта В		
		Аналоговая	S17	Выход 17 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
30	PB12	Вход/выход 12 порта В		
		Аналоговая	S18	Выход 18 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
31	PB13	Вход/выход 13 порта В		
		Аналоговая	S19	Выход 19 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–
32	PB14	Вход/выход 14 порта В		
		Аналоговая	S20	Выход 20 управления сегментами LCD-дисплея
		Основная	–	–
		Альтернативная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
Порт С				
7	PC0/ MODE	Вход/выход 0 порта С		
		Аналоговая	–	–
		Основная	UART1_TXD	Выходные данные UART1
		Альтернативная	–	–
48	PC1	Вход/выход 1 порта С		
		Аналоговая	ADC3	Аналоговый вход канала 3 АЦП последовательных приближений
		Основная	UART1_RXD	Входные данные UART1
		Альтернативная	–	–
47	PC2	Вход/выход 2 порта С		
		Аналоговая	ADC2, VLCD	Аналоговый вход канала 2 АЦП последовательных приближений
		Основная	TMR1_CH1	Вход/выход канала 1 таймера 1
		Альтернативная	–	–
46	PC3	Вход/выход 3 порта С		
		Аналоговая	ADC1_REF+	Аналоговый вход канала 1/Напряжение верхней границы внешней опоры АЦП последовательных приближений
		Основная	TMR1_CH1N	Инверсный выход канала 1 таймера 1
		Альтернативная	–	–
45	PC4	Вход/выход 4 порта С		
		Аналоговая	ADC0_REF-	Аналоговый вход канала 0/ Напряжение нижней границы внешней опоры АЦП последовательных приближений
		Основная	–	–
		Альтернативная	–	–
33	PC5	Вход/выход 5 порта С		
		Аналоговая	S21	Выход 21 управления сегментами LCD-дисплея
		Основная	TMR1_ETR	Вход внешнего события таймера 1
		Альтернативная	–	–
34	PC6	Вход/выход 6 порта С		
		Аналоговая	S22	Выход 22 управления сегментами LCD-дисплея
		Основная	TMR1_BRK	Вход внешнего события таймера 1
		Альтернативная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
35	PC7	Вход/выход 7 порта C		
		Аналоговая	S23	Выход 23 управления сегментами LCD-дисплея
		Основная	EXT_INT0	Вход внешнего прерывания
		Альтернативная	–	–
11	PC8	Вход/выход 8 порта C		
		Аналоговая	S24	Выход 24 управления сегментами LCD-дисплея
		Основная	EXT_INT1	Вход внешнего прерывания
		Альтернативная	–	–
10	PC9	Вход/выход 9 порта C		
		Аналоговая	S25	Выход 25 управления сегментами LCD-дисплея
		Основная	EXT_INT2	Вход внешнего прерывания
		Альтернативная	–	–
36	PC10	Вход/выход 10 порта C		
		Аналоговая	S26	Выход 26 управления сегментами LCD-дисплея
		Основная	SSP_FSS	Вход/выход разрешения SPI
		Альтернативная	–	–
37	PC11	Вход/выход 11 порта C		
		Аналоговая	S27	Выход 27 управления сегментами LCD-дисплея
		Основная	SSP_CLK	Вход/выход синхросигнала SPI
		Альтернативная	–	–
38	PC12	Вход/выход 12 порта C		
		Аналоговая	S28	Выход 28 управления сегментами LCD-дисплея
		Основная	SSP_RXD	Входные данные SPI
		Альтернативная	–	–
39	PC13	Вход/выход 13 порта C		
		Аналоговая	S29	Выход 29 управления сегментами LCD-дисплея
		Основная	SSP_TXD	Выходные данные SPI
		Альтернативная	–	–
3	PC14	Вход/выход 14 порта C		
		Аналоговая	–	–
		Основная	SCL	Выход синхросигнала I2C
		Альтернативная	–	–
4	PC15	Вход/выход 14 порта C		
		Аналоговая	–	–
		Основная	SDA	Вход/выход данных I2C
		Альтернативная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
Порт АЦП				
50	VR_1V	Вход внешнего опорного напряжения 1 В		
51	I0P	Положительный вход канала 0 тока		
52	I0N	Отрицательный вход канала 0 тока		
53	V0P	Положительный вход канала 0 напряжения		
54	V0N	Отрицательный вход канала 0 напряжения		
55	I3P	Положительный вход канала 3 тока		
56	I3N	Отрицательный вход канала 3 тока		
Системное управление				
44	RESET	Сигнал внешнего сброса		
58	WAKEUP/ COV_DET	Сигнал внешнего выхода из режима Standby		
2	OSC_IN	Вход генератора HSE		
1	OSC_OUT	Выход генератора HSE		
Питание				
5	U <sub>CC</sub>	Питание (2,2...3,6) В		
57	AU <sub>CC</sub>	Аналоговое питание ΔΣАЦП (2,4...3,6) В		
61	BU <sub>CC</sub>	Батарейное питание (1,8...3,6) В		
49	AGND	Общий АЦП, ЦАП		
6	GND	Общий		
Выводы для тестирования и исследования				
64	JTAG_EN	Тестовый вывод. Рекомендуется не подсоединять или подключать к шине общий		

## 5 Диаграмма расположения выводов в корпусах

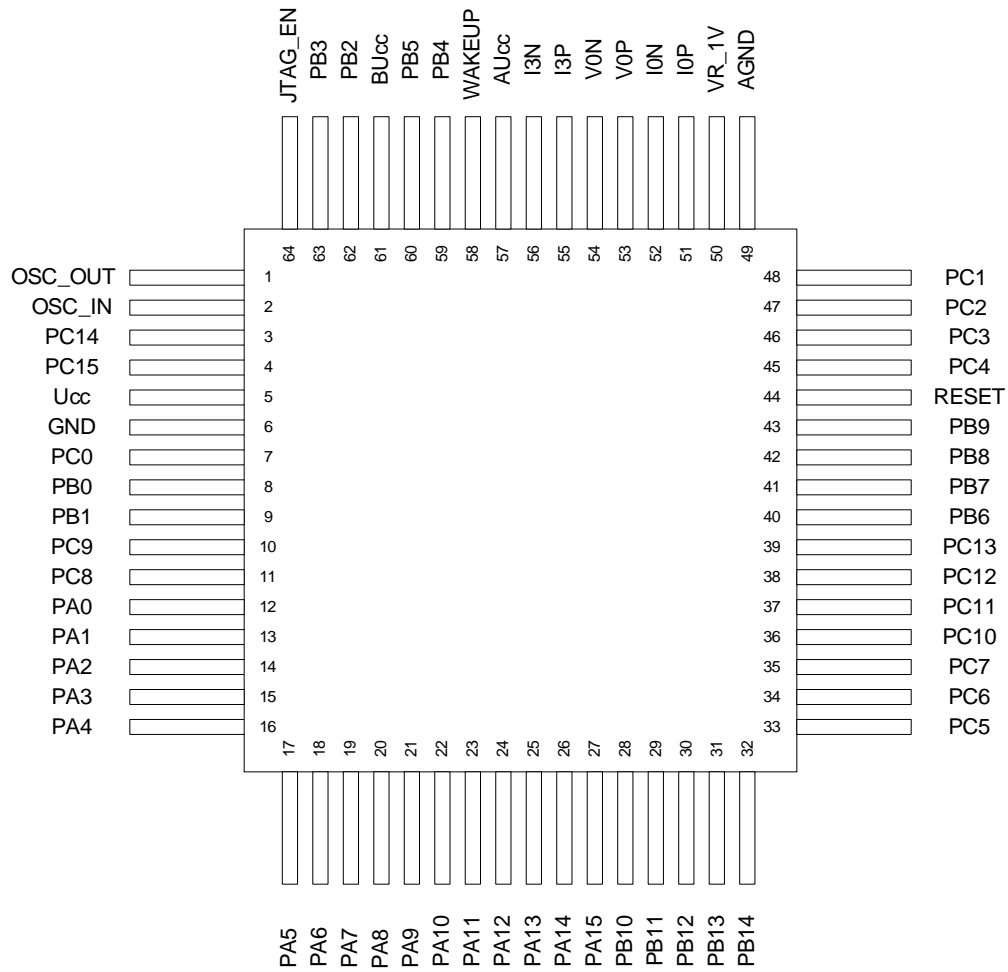


Рисунок 3 – Диаграмма расположения выводов

## 6 Указания по применению и эксплуатации

Длительность фронта/спада входного сигнала ( $T_r$ ,  $T_f$ ) составляет не более 5 нс.

Режимы и условия монтажа микросхем в аппаратуре по ОСТ 11 073.063.

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

Запрещается подведение каких-либо электрических сигналов (в том числе шин питания и общий) к выходу неиспользуемому в электрической схеме.

Неиспользуемые входы микросхем следует подключать к шинам питания и общий.

Порядок подачи и снятия напряжения питания и входных сигналов на микросхему:

– подача (включение микросхемы) – общий, питание, входные сигналы или одновременно;

– снятие (выключение микросхемы) – одновременно или в обратном порядке.

## 7 Система питания

Микросхемы этой серии имеет несколько типов выводов питания:

**U<sub>CC</sub>** – Основное питание микросхемы, включает питание пользовательских выводов, встроенного регулятора напряжения, умножителя тактовой частоты PLL, генераторов и АЦП последовательного приближения.

**V<sub>U<sub>CC</sub></sub>** – Питание батарейного домена используется при отсутствии основного питания U<sub>CC</sub> для питания батарейного домена и генератора LSE. Переключение с основного питания на батарейное происходит автоматически при снижении уровня U<sub>CC</sub> ниже 2,0 В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 4 мс после превышения уровнем U<sub>CC</sub> 2,0 В. Если в системе не требуется батарейного питания вывод V<sub>U<sub>CC</sub></sub> должен быть объединен с U<sub>CC</sub>.

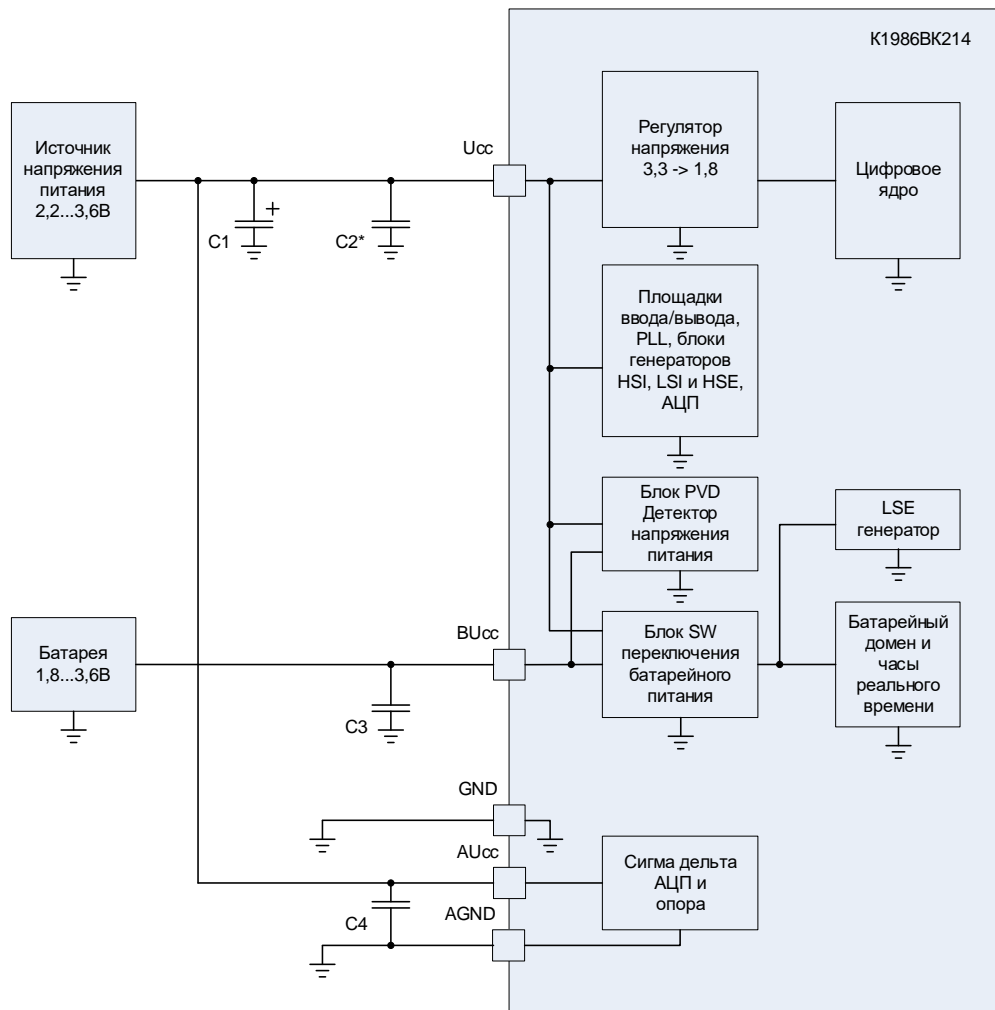
**AU<sub>CC</sub>** – Питание аналоговых блоков сигма дельта АЦП и формирователя опоры выведено на отдельные выводы для уменьшения помех, создаваемых работой других блоков. На данные выводы должно подаваться напряжения с того же источника что и U<sub>CC</sub>, но при этом на печатной плате должны быть применены меры по снижению наводки помех. Допускается использование отдельного источника для питания аналоговых блоков, при этом его выходное напряжение не должно отличаться от U<sub>CC</sub> более чем на ± 0,2 В. Для корректной работы АЦП и ΔΣАЦП входное напряжение должно быть в пределах от 3,0 до 3,6 В. Если входное напряжение будет в пределах от 2,2 до 3,0 В, то корректная работа АЦП не гарантируется.

**GND** – Основная земля питания.

**AGND** – Земля аналогового питания AU<sub>CC</sub>. Данные выводы должны соединяться с GND, но при этом на печатной плате должны быть применены меры по снижению наводки помех.



## 7.1 Структурная схема подачи питания



### Примечания

\* Конденсаторы должны быть установлены у каждого вывода питания;

1 Конденсатор C1 = 22 мкФ, C2 = C3 = C4 = 0,1 мкФ;

2 Если не используется батарейное питание, то вывод BU<sub>CC</sub> должен быть объединен с U<sub>CC</sub>;

3 Если используется АЦП последовательного приближения, то напряжение питания U<sub>CC</sub> должно быть в пределах от 2,4 до 3,6 В;

4 Если используется ΔΣАЦП, то напряжение питания U<sub>CC</sub> (AU<sub>CC</sub>) должно быть в пределах от 3,0 до 3,6 В.

Рисунок 4 – Схема подачи питания микросхемы

Микросхема имеет несколько режимов энергопотребления, подробнее смотри подраздел 25 «Управление электропитанием».

Микросхема имеет встроенный детектор напряжения питания, подробнее смотри раздел 17 «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<sub>CC</sub> нарастает и, пока оно не превысило уровень 2,0 В, сигнал сброса POR удерживается; после превышения данного уровня сигнал POR выдается еще

на протяжении ~ 4 мс для того, чтобы гарантировано установилось напряжение питания, после чего сигнал POR снимается, и схема может начать работать.

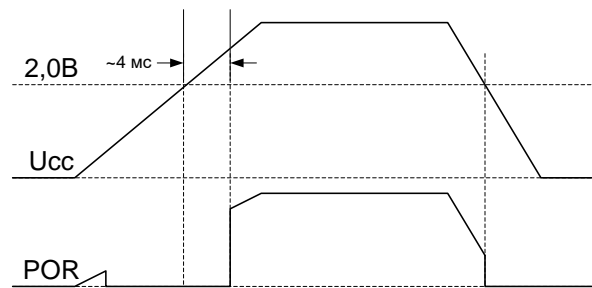


Рисунок 5 – Схема включения основного питания U<sub>CC</sub>

При снижении напряжения питания U<sub>CC</sub> ниже уровня 2,0В сигнал POR вырабатывается без задержки.

Сигнал POR также служит для переключения питания батарейного домена между BU<sub>CC</sub> и U<sub>CC</sub>.

При включении основного напряжения питания U<sub>CC</sub> автоматически включается встроенный регулятор напряжения для формирования напряжения DU<sub>CC</sub> (см рисунок 1) питания цифрового ядра. В ходе работы микросхемы встроенный регулятор может быть отключен.

Установка микросхемы в исходное состояние может быть произведена внешним сигналом сброса nRESET, или внутренними сигналами сброса сторожевых таймеров (IWDG, WWDG), или программным сбросом. При этом сигнал nRESET формируется специальной схемой сброса, содержащий фильтр импульсных помех и одновибратор для увеличения длительности этого сигнала.

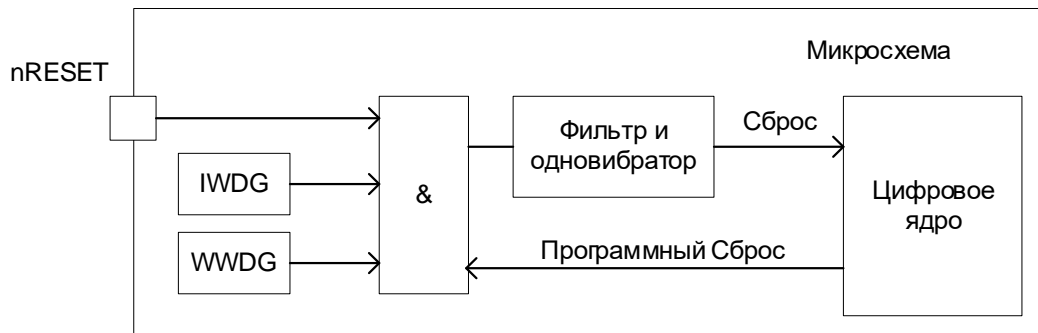


Рисунок 6 – Схема формирования сигнала сброса

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

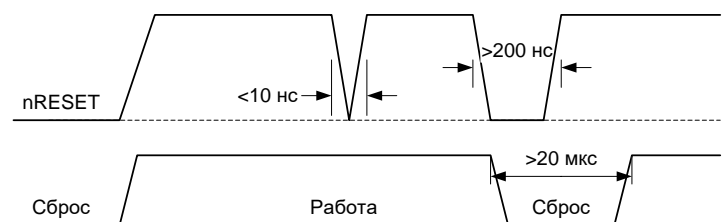


Рисунок 7 – Временная диаграмма формирования сигнала сброса

## 8 Организация памяти

### 8.1 Структурная схема

Процессорное ядро имеет одну системную шину АНВ-Lite (шина выборки инструкций и данных).

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

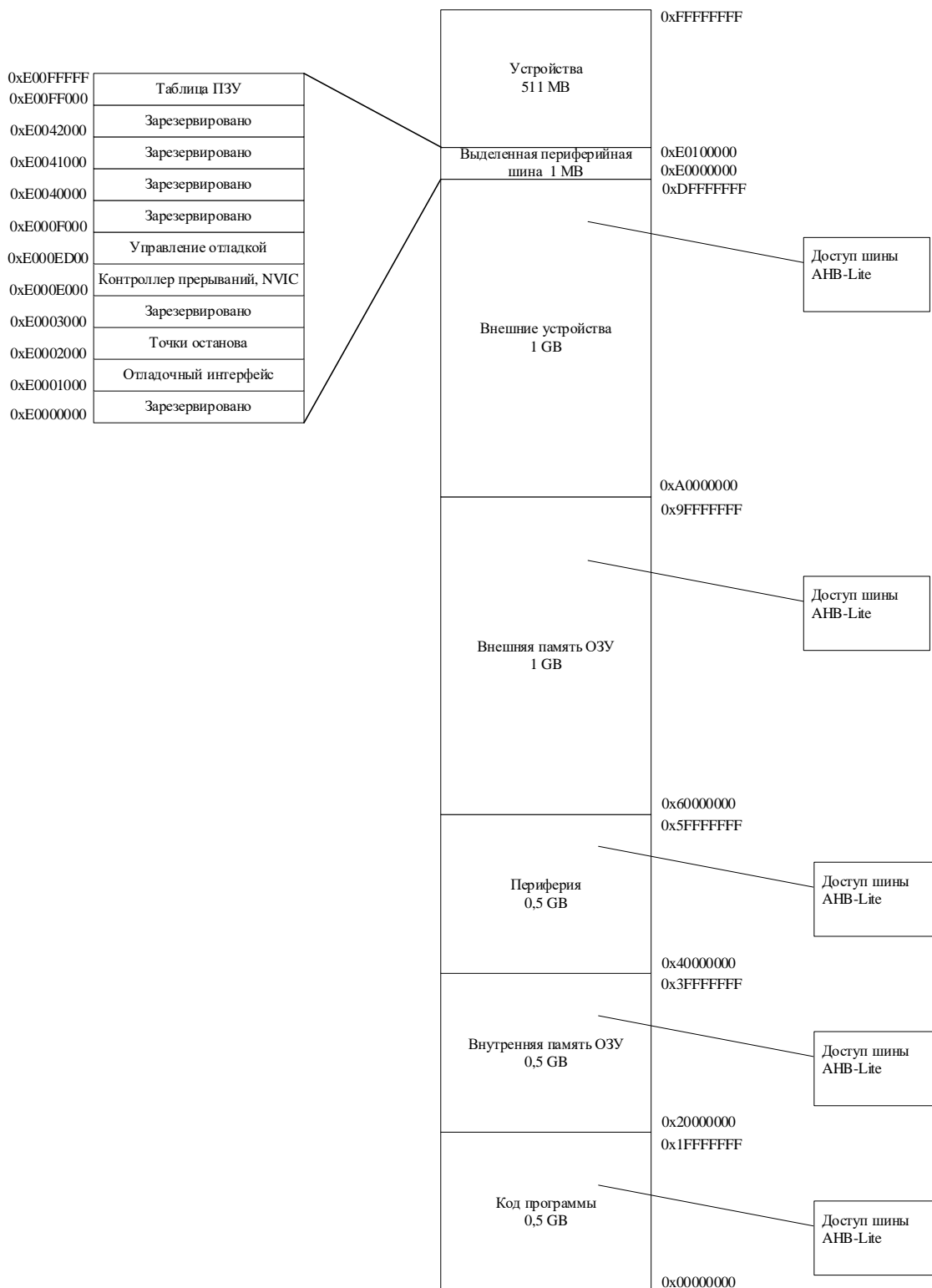


Рисунок 8 – Структура адресного пространства микросхемы

## 8.2 Базовые адреса процессора

Таблица 3 – Перечень базовых адресов процессора

Адрес	Размер	Блок	Примечание
Память программ			
0x0000_0000		BOOT ROM	Загрузочная программа FPOR=0
0x0000_0000		EEPROM	Область Flash-памяти программ с пользовательской программой FPOR=1
Память данных			
0x2000_0000		SYSTEM RAM	Область внутреннего ОЗУ
Периферия			
0x4000_0000		SPI	Регистры контроллера интерфейса SSP (см. подраздел 21 «Контроллер SSP»)
0x4000_8000		UART0	Регистры контроллера интерфейса UART0 (см. пункт 9.1 «UART-загрузчик»)
0x4001_0000		UART1	Регистры контроллера интерфейса UART1 (см. пункт 9.1 «UART-загрузчик»)
0x4001_8000		EEPROM_CNTRL	Регистры контроллера Flash-памяти программ
0x4002_0000		RST_CLK	Регистры контроллера сигналов тактовой частоты
0x4003_0000		I <sup>2</sup> C	Регистры контроллера интерфейса I <sup>2</sup> C
0x4003_8000		LCD	Регистры контроллера LCD-дисплея
0x4004_0000		ADC	Регистры управления АЦП
0x4004_8000		WWDG	Регистры контроллера сторожевого таймера WWDG
0x4005_0000		IWDG	Регистры контроллера сторожевого таймера IWDG
0x4005_8000		POWER	Регистры детектора напряжения питания
0x4006_0000		BKP	Регистры доступа и управления батарейным доменом
0x4006_8000		ADCUI	Регистры управления $\Delta\Sigma$ АЦП
0x4007_0000		TIMER0	Регистры управления 16-разрядного таймера 0
0x4007_8000		TIMER1	Регистры управления 16-разрядного таймера 1
0x4008_0000		PORTA	Регистры управления порта ввода-вывода А
0x4008_8000		PORTB	Регистры управления порта ввода-вывода В
0x4009_0000		PORTC	Регистры управления порта ввода-вывода С
SYSTEM REGION			
0xE000_0000			Системные регистры, процессор RISC

## 9 Загрузочное ПЗУ и режимы работы микросхемы

После включения питания и снятия сигналов сброса: внутренних (POR) и внешних (RESET), – микросхема начинает выполнять программу из информационной загрузочной области FLASH. В загрузочной программе микросхема определяет, в каком из режимов она будет функционировать и переходит в этот режим. Режим функционирования определяется внешними выводами MODE[1:0] (PB[0], PC[0]). Также устанавливается бит FPOR в регистре ВКР\_REG\_0E (см. таблицу 125), который может быть сброшен только при отключении основного питания U<sub>CC</sub>. После перезапуска микросхемы уровни на выводах MODE[1:0] не влияют на режим функционирования микросхемы, если установлен бит FPOR. В пользовательской программе выводы PB[0], PC[0] могут использоваться пользователем.

Таблица 4 – Режимы работы микросхемы

MODE[1:0]	Режим	Стартовый адрес/таблица векторов прерываний	Описание
00	Микроконтроллер в режиме отладки	0x00000000	Процессор начинает выполнять программу из внутренней FLASH-памяти программ. При этом разрешается работа отладочного интерфейса SW (Serial Wire)
01	UART-загрузчик	Определяется пользователем	Микросхема через интерфейс UART1 на выводах PB[1], PB[0] получает код программы в ОЗУ для исполнения. При этом разрешается работа отладочного интерфейса SW (Serial Wire)
10	UART-загрузчик	Определяется пользователем	Микросхема через интерфейс UART1 на выводах PB[1], PB[0] получает код программы в ОЗУ для исполнения. При этом разрешается работа отладочного интерфейса SW (Serial Wire)
11	Запрещенная ситуация	–	Режим для проверки микросхемы после производства. Микросхема перестает работать в функциональном режиме

При работе в режиме отладки разрешается работа отладочного интерфейса Serial Wire. При этом к микросхеме может быть подключен Serial адаптер (не входит в комплект поставки) с помощью которого программные средства разработки позволяют работать с микросхемой в отладочном режиме.

В отладочном режиме можно:

- стирать, записывать, считывать внутреннюю FLASH-память программ;
- считывать и записывать содержимое ОЗУ, периферии;
- выполнять программу в пошаговом режиме;

- запускать программу в нормальном режиме;
- останавливать программу по точкам остановки;
- просматривать переменные выполняемой программы;
- проводить трассировку хода выполнения программного обеспечения.

## 9.1 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 (ЭСППЗУ, ПЗУ, Периферия).

В качестве источника тактовой частоты UART0 используется внутренний RC-генератор HSI с частотой 8 МГц. Так как имеется разброс значений частоты HSI, то требуется этап подбора значения делителя частоты UART0 для синхронизации с внешним устройством.

При загрузке программы в ОЗУ необходимо учитывать, что диапазон адресов 0x2000\_0000-0x2000\_04FF используется загрузочной программой для работы со стеком и глобальными переменными.

### 9.1.1 Параметры связи по UART

Для связи по UART выбраны следующие параметры канала связи:

- Начальная скорость – 9600 бод;
- Количество бит данных – 8;
- Четность – нет;
- Количество стоп бит – 1;
- Загрузчик не использует FIFO UART0;
- Загрузчик всегда выступает в качестве Slave, а внешнее устройство, подающее команды – в качестве Master;
- Данные передаются младшим битом вперед.

### 9.1.2 Протокол обмена по UART

После синхронизации с Master загрузчик переходит в диспетчер команд.

Таким образом, Master-у доступны следующие команды:

Таблица 5 – 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

### 9.1.3 Синхронизация с внешним устройством

#### Начальные условия

На этапе синхронизации с внешним устройством (Master) вывод PB1 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 приглашение (три байта 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>'),) Master-у.

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

### 9.1.4 Команда CMD\_SYNC

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

Таблица 6 – Команда CMD\_SYNC

Код команды	CMD_SYNC = 0x00
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
<b>Формат команды:</b>	
Master: выдает код команды CMD_SYNC	Slave: принимает команду. Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') (см. подпункт 9.1.11 «Сообщения об ошибках») и завершает обработку текущей команды

### 9.1.5 Команда CMD\_CR

Выдача приглашения Master-у.

Таблица 7 – Команда CMD\_CR

Код команды	CMD_CR = 0x0D
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
<b>Формат команды:</b>	
Master: выдает код команды CMD_CR	Slave: принимает команду. Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. Иначе выдает три байта: – код команды CMD_CR; – код 0x0A; – код 0x3E (ASCII символ '>')

### 9.1.6 Команда CMD\_BAUD

Установка скорости обмена.

Таблица 8 – Команда CMD\_BAUD

Код команды	CMD_BAUD = 0x42
ASCII символ, соответствующий коду команды	'B'
Количество параметров команды	1
Параметр	Новое значение скорости обмена [бод]
<b>Формат команды:</b>	
Master: выдает код команды CMD_BAUD	Slave: принимает команду. Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: выдает параметр	Slave: принимает параметр. Если параметр принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') (см. подпункт 9.1.11 «Сообщения об ошибках») и завершает обработку текущей команды. Иначе: – выдает код команды CMD_BAUD; – устанавливает новое значение скорости обмена во время отправки кода команды CMD_BAUD

### 9.1.7 Команда CMD\_LOAD

Загрузка массива байт в память микросхемы.

Таблица 9 – 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
<b>Формат команды:</b>	
Master: выдает код команды CMD_LOAD	Slave: принимает команду. Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: выдает параметр 1	Slave: принимает параметр 1
Master: выдает параметр 2	Slave: принимает параметр 2. Если параметр принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. Иначе выдает код команды CMD_LOAD
Master: выдает массив байт младшим байтом вперед	Slave: принимает массив байт. Если хотя бы один байт принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончании принятия массива выдает код ответа REPLY_OK = 0x4B ('K')



### 9.1.8 Команда CMD\_VFY

Выдача массива байт из памяти микросхемы.

Таблица 10 – 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
<b>Формат команды:</b>	
Master: выдает код команды CMD_VFY	Slave: принимает команду. Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: выдает параметр 1	Slave: принимает параметр 1
Master: выдает параметр 2	Slave: принимает параметр 2. Если параметр принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. Иначе: – выдает код команды CMD_VFY; – выдает массив байт младшим байтом вперед; – по окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K')

### 9.1.9 Команда CMD\_RUN

Запуск программы на выполнение.

Таблица 11 – Команда CMD\_RUN

Код команды	CMD_RUN = 0x52
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр	Адрес первой команды загруженной программы (младший бит адреса должен быть равен 0)
<b>Формат команды:</b>	
Master: выдает код команды CMD_RUN	Slave: принимает команду. Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: выдает параметр	Slave: принимает параметр. Если параметр принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. Иначе: – выдает код команды CMD_RUN; – устанавливает значение PC согласно принятому адресу (MSP не перепрограммируется) и, таким образом, Slave завершает свое выполнение. Передача управления загруженной программе происходит, не дожидаясь окончания отправки кода команды CMD_RUN

### 9.1.10 Прием параметров команды

Параметры команд – это четырехбайтные числа.

Параметры передаются младшим байтом вперед.

В качестве значения параметра запрещено использовать число 0xFFFFFFFF.

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

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

### 9.1.11 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 прекращает 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.

### 9.1.12 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.

Примечание – не реализовано в загрузчике.

### 9.1.13 Ошибка ERR\_CMD

Принята неизвестная команда.

Код ошибки 0x63 ('c').

Выдается диспетчером команд, если принят неизвестный код команды.

Примечание – не реализовано в загрузчике.

### 9.1.14 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.

Примечание – не реализовано в загрузчике.

## 10 Контроллер FLASH-памяти программ

Микросхема содержит встроенную Flash-память программ с объемом 64 Кбайт основной памяти программ и 4 Кбайта информационной памяти. В микросхеме реализованы два режима работы:

- обычный режим;
- режим программирования.

В обычном режиме (бит CON = 0, регистр EEPROM\_CMD (см. таблицу 13)) доступна основная память программ через системную шину для выборки инструкций и данных кода программы.

В режиме программирования (бит CON=1, регистр EEPROM\_CMD) основная и информационная память доступны как периферийное устройство и могут быть использованы для нужд разработчика приложения. В режиме программирования программный код должен выполняться из области 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 55 нс, в результате выдача новых значений из Flash-памяти может происходить с частотой не более 18 МГц. Чтобы процессорное ядро могло получать новые инструкции на больших частотах, в микросхеме реализуется Flash-память с физической организацией 64К на 32 разряда. Таким образом, за 55 нс из Flash-памяти извлекается восемь байт, в которых может быть закодировано две инструкции процессора. И пока ядро выполняет эти инструкции, из памяти извлекается следующая порция данных. Таким образом, тактовая частота может превышать частоты извлечения данных из памяти в несколько раз при линейном выполнении программы. При возникновении переходов в выполнении программы, когда из памяти программ не выбраны нужные инструкции, возникает пауза в несколько тактов процессора для того, чтобы данные успели считаться из Flash. Число тактов паузы зависит от тактовой частоты процессора. При работе с частотой ниже 18 МГц пауза не требуется, так как Flash-память успевает выдать новые данные за один такт, при частоте от 18 до 36 МГц требуется один такт паузы, и так далее. Число тактов паузы задается в регистре EEPROM\_CMD битами Delay[1:0]. В таблице 12 приведены характеристики необходимой паузы для работы Flash-памяти программ.

Таблица 12 – Характеристики паузы

Delay[1:0]	Тактов паузы	Тактовая частота	Примечание
0x00	0	До 18 МГц	
0x01	1	До 36 МГц	

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

## 10.2 Работа Flash-памяти программ в режиме программирования

В режиме программирования Flash-память программ не может выдавать инструкции и данные процессору, поэтому перевод памяти в режим программирования (установка бита CON = 1) возможен только программой, исполняемой из ОЗУ.

Информационная память может быть назначена на адрес 0x0000\_0000 (вместо загрузочной программы ROM) с помощью управляющего бита FPOR (расположен в батарейном домене).

В режиме программирования возможны следующие операции как с основной (бит IFREN = 0, регистр EEPROM\_CON (см. таблицу 13)), так и с информационной (бит IFREN = 1) памятью:

- стирание блока памяти размером 2 Кбайт или 32 Кбайт;
- стирание страницы памяти размером 512 байт;
- запись 32-битного слова в память;
- чтение 32-битного слова из памяти.

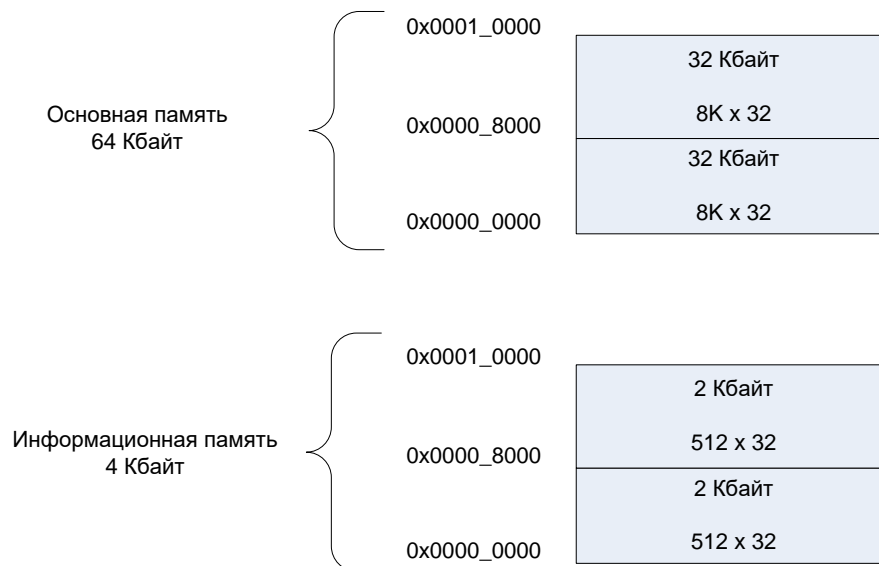


Рисунок 9 – Структура памяти

За выбор блоков памяти (как основной, так и информационной) отвечают биты ADR[16:15].

### 10.2.1 Стирание блока памяти размером 2 Кбайт или 32 Кбайт

Стирание памяти возможно только в режиме программирования. Для стирания блока памяти надо установить необходимое значение в бит IFREN в регистре EEPROM\_CMD (1 – для основной и информационной памяти (стирание блоков памяти размером 32 Кбайт и 2 Кбайт); 0 – для основной памяти (стирание блока памяти размером 32 Кбайт)). Установить номер стираемого блока в регистре EEPROM\_ADR, биты ADR[16:15], затем установить биты XE, MAS1 и ERASE (см. таблицу 15) в «1», и спустя время T<sub>nvS</sub> (не менее 10 мкс) установить бит NVSTR в «1». Полное стирание блока памяти длится время T<sub>me</sub> (от 20 мс до 40 мс). Спустя это время необходимо очистить бит ERASE, и спустя время T<sub>nvh1</sub> (не менее 100 мкс) очистить

биты XE, MAS1 и NVSTR. Последующие операции с памятью можно выполнять спустя время  $T_{rcv}$  (не менее 1 мкс). При стирании выбранного битами ADR[16:15] блока информационной памяти стирается и соответствующий блок основной памяти.

Временная диаграмма стирания памяти представлена на рисунке 10.

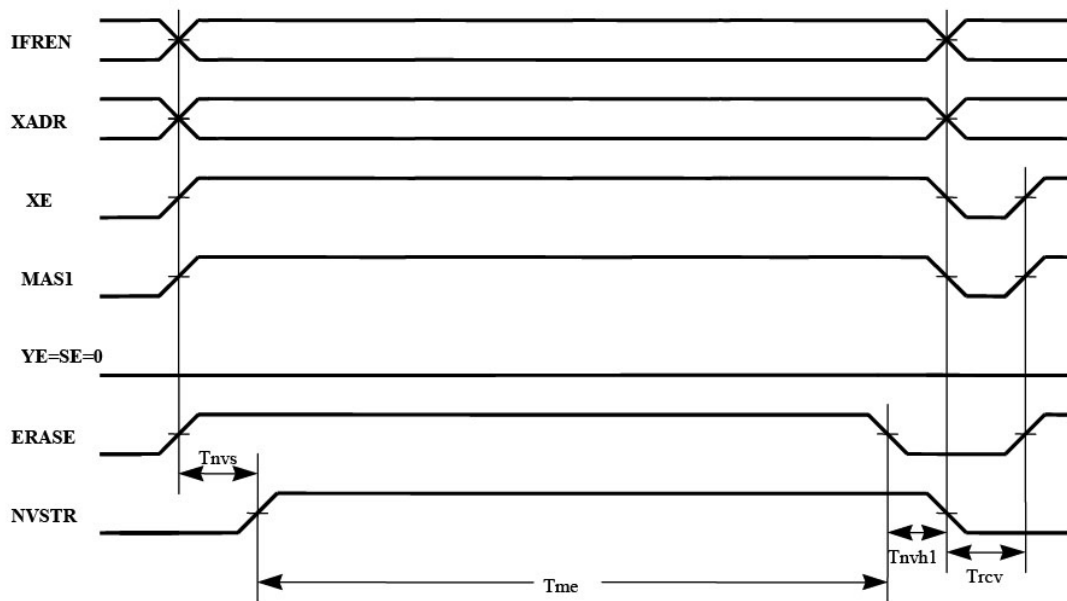


Рисунок 10 – Временная диаграмма стирания памяти

### 10.2.2 Стирание страницы памяти размером 512 байт

Стирание страницы памяти возможно только в режиме программирования. Для стирания страницы памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес стираемой страницы в регистре EEPROM\_ADR и установить биты XE и ERASE в «1», и спустя время  $T_{nvs}$  (не менее 10 мкс) установить бит NVSTR в «1». Стирание страницы памяти длится время  $T_{erase}$  (от 20 мс до 40 мс). Спустя это время необходимо очистить бит ERASE, и спустя время  $T_{nvh}$  (не менее 5 мкс) очистить биты XE и NVSTR. Последующие операции с памятью можно выполнять спустя время  $T_{rcv}$  (не менее 1 мкс). Временная диаграмма стирания страницы памяти представлена на рисунке 11.

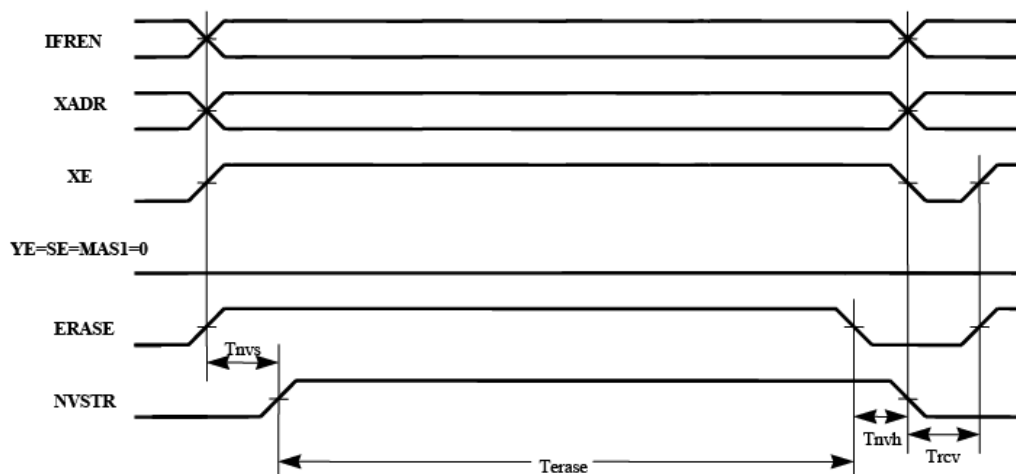


Рисунок 11 – Временная диаграмма стирания страницы

### 10.2.3 Запись 32-битного слова в память

Запись в память возможно только в режиме программирования. Для записи в память надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес, по которому производится запись в регистре EEPROM\_ADR, в регистр EEPROM\_DI записать записываемое в память слово и установить биты XE и PROG в «1», и спустя время  $T_{nvs}$  (не менее 5 мкс) установить бит NVSTR в «1». Спустя время  $T_{pgs}$  (не менее 10 мкс) установить бит YE в «1». Запись в память длится время  $T_{prog}$  (от 20 мкс до 40 мкс). Спустя это время необходимо очистить бит YE, и спустя время  $T_{adh}$  (не менее 20 нс) установить новый адрес и значение для записи в другую ячейку памяти. И спустя  $T_{ads}$  (не менее 20 нс) установить бит YE в «1» и записать следующее слово. Если запись больше не требуется, то спустя время  $T_{pgh}$  (не менее 20 нс) после очистки бита YE необходимо очистить бит PROG и спустя время  $T_{nvh}$  (не менее 5 мкс) очистить биты XE и NVSTR. Последующие операции с памятью можно выполнять спустя время  $T_{rcv}$  (не менее 1 мкс). Временная диаграмма записи памяти представлена на рисунке 12.

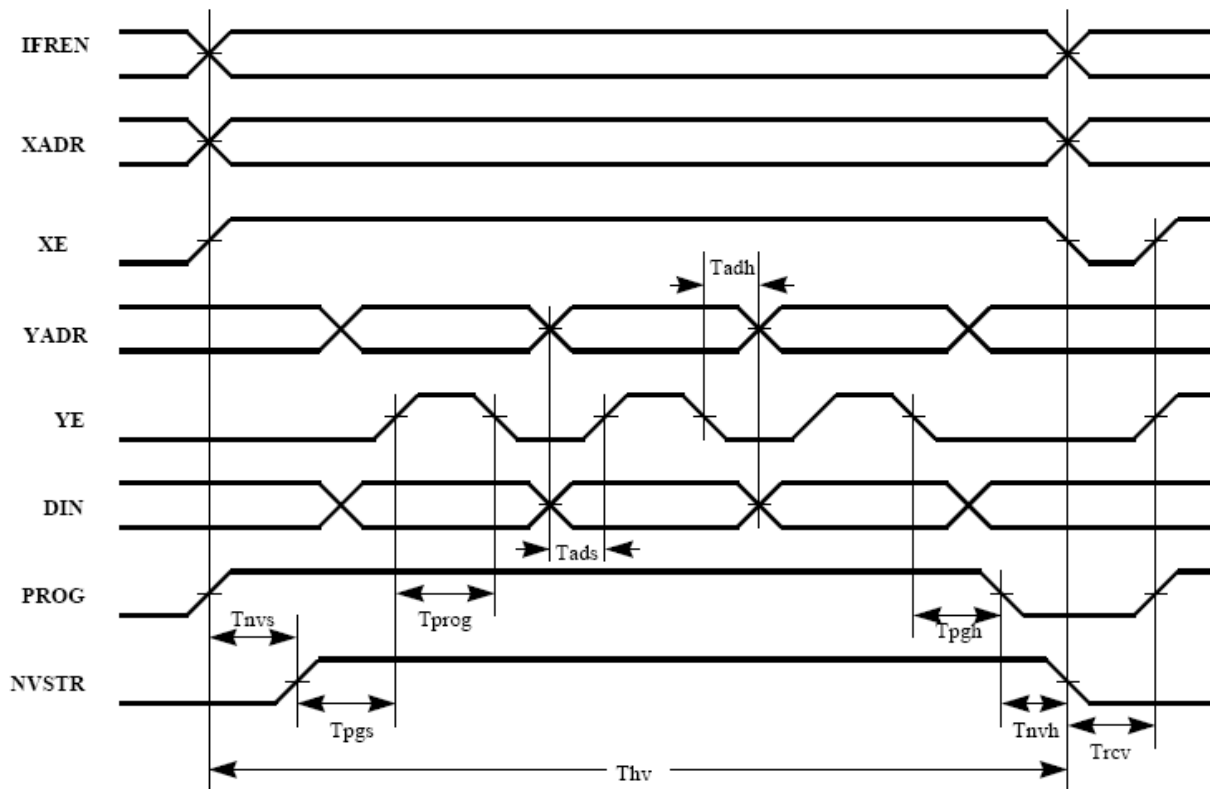


Рисунок 12 – Временная диаграмма записи памяти

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

В обычном режиме работы для чтения доступна только основная память. Для этого необходимо просто считать требуемый адрес памяти. В режиме программирования для чтения доступна и основная и информационная память. Для чтения из памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес, из которого необходимо считать данные в регистре EEPROM\_ADR и установить биты XE, YE и SE в «1», и спустя

время  $T_{ха}$  (не менее 55 нс) из регистра EEPROM\_DO можно считать данные. Если необходимо считать следующее слово, то в регистр EEPROM\_ADR необходимо записать новый адрес и спустя время  $T_{ха}$  (не менее 55 нс) из регистра EEPROM\_DO можно считать следующие данные. Если чтение больше не требуется, то можно очистить все биты управления. Временная диаграмма чтения памяти представлена на рисунке 13.

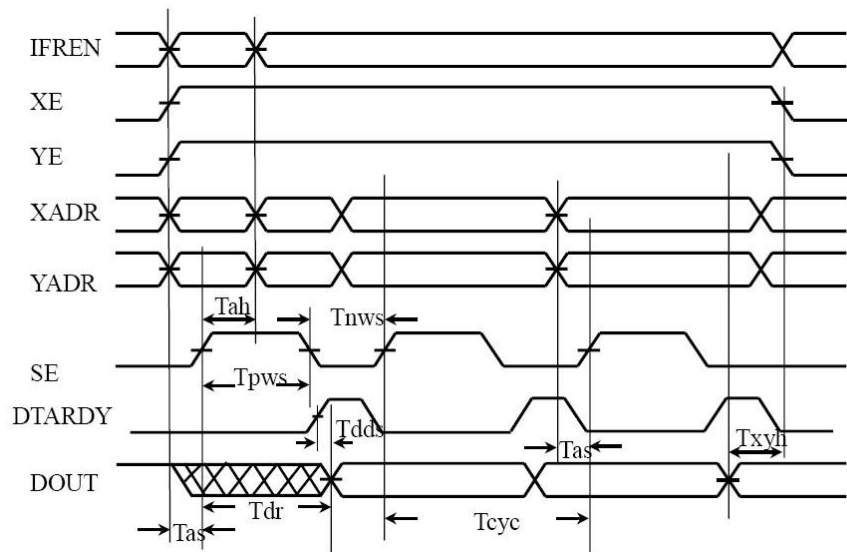


Рисунок 13 – Временная диаграмма чтения памяти

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

### 10.3 Описание регистров управления контроллера Flash-памяти программ

Таблица 13 – описание регистров управления контроллера Flash-памяти программ

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

#### 10.3.1 EEPROM\_CMD

Таблица 14 – Регистр EEPROM\_CMD

Номер	9	8	7	6	5...3	2, 1	0
Доступ	R/W	R/W	R/W	R/W	R/W		R/W
Сброс	0	0	0	0	100		0
	IFREN	SE	YE	XE	Delay[2:0]		CON

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

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

Таблица 15 – Описание битов регистра EEPROM\_CMD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..14	-	Зарезервировано
13	NVSTR	Операции записи или стирания: 0 – при чтении; 1- при записи или стирании
12	PROG	Записать данные по ADR[16:2] из регистра EEPROM_DI: 0 – нет записи; 1 – есть запись
11	MAS1	Стереть весь блок, при ERASE =1: 0 – нет стирания; 1 – стирание
10	ERASE	Стереть строку с адресом ADR[16:9], ADR[8:0] значения не имеет: 0 – нет стирания; 1 – стирание
9	IFREN	Работа с блоком информации: 0 – основная память; 1 – информационный блок
8	SE	Усилитель считывания: 0 – не включен; 1 – включен
7	YE	Выдача адреса ADR[8:2]: 0 – не разрешено; 1 – разрешено
6	XE	Выдача адреса ADR[16:9]: 0 – не разрешено 1 – разрешено
5...3	Delay[2:0]	Задержка памяти программ при чтении в циклах (в рабочем режиме): 000 – 0 цикл; 001 – 1 цикл
2, 1	-	Зарезервировано
0	CON	Переключение контроллера памяти EEPROM на регистровое управление, не может производиться при исполнении программы из области EEPROM: 0 – управление EEPROM от ядра, рабочий режим; 1 – управление от регистров, режим программирования



### 10.3.2 EEPROM\_ADR

Таблица 16 – Регистр EEPROM\_ADR

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

Таблица 17 – Описание битов регистра EEPROM\_ADR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31..0	ADR[31:0]	Адрес обращения в память ADR[1:0] – не имеет значения, минимально адресуемая ячейка 32 бита

### 10.3.3 EEPROM\_DI

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

Таблица 18 – Описание битов регистра EEPROM\_DI

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

### 10.3.4 EEPROM\_DO

Таблица 19 – Регистр EEPROM\_DO

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

Таблица 20 – Описание битов регистра EEPROM\_DO

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

### 10.3.5 EEPROM\_KEY

Таблица 21 – Регистр EEPROM\_KEY

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

Таблица 22 – Описание битов регистра EEPROM\_KEY

Номер бита	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 EEPROM_KEY записать комбинацию 0x8AAA5551

## 11 Процессорное ядро

Описание процессора и периферии ядра.

Процессорное ядро с минимизированным количеством вентиляей, обладающее следующими характеристиками:

- процессорное ядро содержит в своем составе трехуровневый конвейер;
- набор инструкций архитектуры ARM v6-M, включающий 32-битные инструкции Thumb-2, такие как BL, MRS, MSR, ISB, DSB и DMB;
- возможность запуска операционной системы и доступные для этого режима работы инструкции SVC, групповой регистр указателя стека и интегрированный системный таймер;
- системная модель исключительных ситуаций;
- режимы Handler и Thread;
- два указателя стека;
- возможность работы только в режиме Thumb;
- отсутствие аппаратной поддержки невыровненного доступа;
- содержит 13 32-разрядных регистров общего назначения, link-регистр (LR), счетчик команд (PC), программный регистр статуса xPSR, и два групповых регистра указателя стека (SP).

Контроллер прерываний NVIC. Контроллер интегрирован в процессор для уменьшения задержек в процессе прерываний. Обладает следующими характеристиками:

- поддержка до 32 внешних прерываний;
- два бита приоритета, обеспечивающие четырехуровневый приоритет прерываний;
- состояние процессора автоматически сохраняется при входе в прерывание и восстанавливается при выходе, что не вызывает потерь на выполнение инструкций.

Интерфейс памяти ITCM, DTCM, а также внешний интерфейс АНВ-Lite.

TСM интерфейс не поддерживает тактов ожидания, поэтому при тактовой частоте ядра выше 25 МГц, акселератор Flash-памяти выключает тактовую частоту ядра на необходимое количество тактов.

Полный набор отладочных модулей:

- полный доступ в режиме останова ко всей памяти и регистрам;
- отладочный порт DAP;
- модуль точек останова BPU;
- модуль наблюдения данных DW;
- 32-разрядный аппаратный умножитель.

## 11.1 Структурная схема процессора

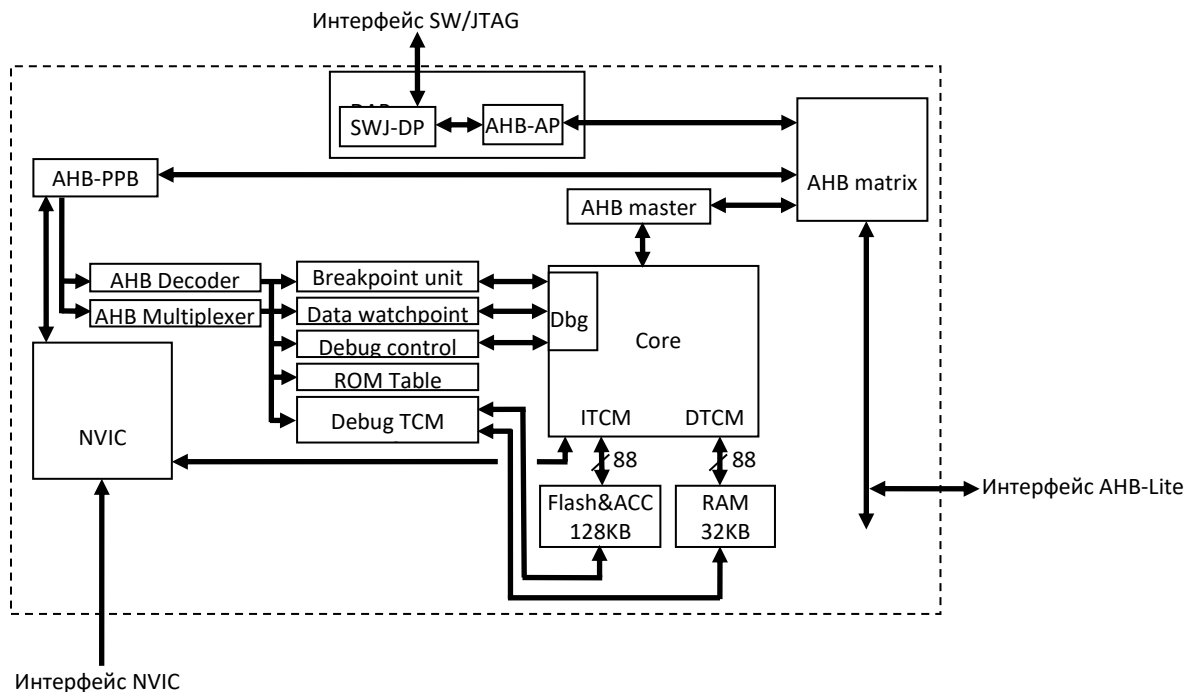


Рисунок 14 – Структурная схема процессора

Периферийными блоками ядра являются:

Контроллер прерываний NVIC – реализует высокоскоростную обработку прерываний.

Bus master – обеспечивает два интерфейса. Один связывает внутренние Private Peripheral Bus (PPB) сигналы с шиной АHB PPB. Второй интерфейс связывает сигналы внешней шины с АHB портом.

АHB Private Peripheral Bus (АHB-PPB) – обеспечивает доступ к контроллеру NVIC и компонентам модулей отладки.

АHB decoder – дешифрирует адреса АHB шины для выработки сигналов выбора для периферии системы отладки.

АHB multiplexer – объединяет все ответы ведомых для отладочных блоков.

АHB matrix – выполняет функцию арбитража между процессором и отладочной системой при доступе к внутренней PPB и внешнему интерфейсу АHB-Lite.

DAP – процессор содержит АHB-Access Port (АHB-AP). АHB-AP преобразует выходы от внешних DP компонентов в интерфейс АHB-Lite. АHB-AP master имеет наивысший приоритет в АHB matrix. Serial-Wire JTAG Debug Port (SWJ-DP) это комбинация JTAG порта и Serial Wire порта, а также механизма, позволяющего переключаться между Serial Wire и JTAG.

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

Breakpoint Unit – содержит в своем составе компаратор 4-х адресов инструкций. Можно сконфигурировать каждый компаратор адреса инструкции для выполнения останова программы с использованием аппаратной точки останова. Каждый компаратор может сравнивать адрес выбираемой инструкции с установленным адресом. Если адрес

совпал, то VPU обеспечивает останов процессора в момент выполнения инструкции, вызвавшей совпадение. Точки останова поддерживаются только в области кода карты памяти.

Data Watchpoint unit – содержит в своем составе два компаратора адреса. Можно сконфигурировать компараторы для сравнения адреса инструкции или адреса данных. Поддерживается также маскирование компараторов. Watchpoint частично. Это означает, что останов ядра происходит после выполнения следующей инструкции, после той, адрес которой вызвал совпадение компаратора.

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

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

## 11.2 Программная модель

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

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

Процессор может функционировать в режимах:

- Режим Thread– используется для исполнения приложений, процессор находится в этом режиме сразу после сброса

- Режим Handler– используется для обработки исключений. После обработки процессор переходит в режим Thread.

Процессор может функционировать в одном из состояний:

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

- Debug state – это состояние, при котором ядро остановлено.

## 11.3 Стек

По окончанию сброса весь код использует main стек. Обработчик прерываний, такой как SVCcall, может переключить стек, который отображался в режиме Thread, из main в process, модификацией значения EXC\_RETURN при выходе. Все прерывания продолжают использовать main стек. Указатель стека, R13, совмещенный регистр переключается между main и process стеком. Только один стек, process или main, виден посредством регистра R13 в данный момент времени.

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

## 11.4 Регистры ядра

Процессор содержит следующие 32-разрядные регистры:

- 13 регистров общего назначения, R0-R12;
- указатель стека (SP, R13) и объединенные регистры, SP\_process и SP\_main;
- Link-регистр (LR, R14);
- счетчик команд (PC, R15);
- программный регистр состояния, xPSR.

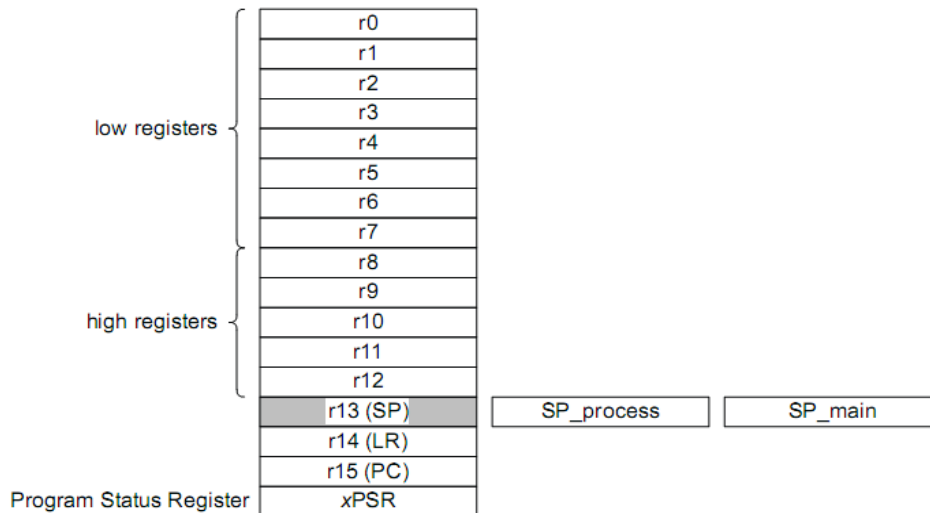


Рисунок 15 – Регистры ядра

### 11.4.1 Регистры общего назначения R0-R12

Low registers или R0-R7 доступны для все инструкций, которые определены для работы с регистрами общего назначения.

High registers или R8-R12 не доступны для 16-разрядных инструкций.

### 11.4.2 Указать стека SP R13

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

### 11.4.3 Регистр связи LR R14

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

#### 11.4.4 Счетчик команд PC R15

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

#### 11.4.5 Программный регистр состояния PSR

Регистр Program Status Register (PSR) объединяет:

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

Эти регистры разделяют различные битовые поля в 32-разрядном PSR. Описание регистров приведено в таблице 23. Доступ к этим регистрам может быть, как индивидуальный, так и комбинированный к двум или всем трем разом, используя имена регистров как аргументы инструкций MSR или MRS, например:

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

Таблица 23 – Комбинация PSR и их атрибуты

Регистр	Тип	Комбинация
XPSR	RW <sup>(1),(2)</sup>	APSR, EPSR и IPSR
IEPSR	RO	EPSR и IPSR
IAPSR	RW <sup>(1)</sup>	APSR и IPSR
EAPSR	RW <sup>(2)</sup>	APSR и EPSR
1 Игнорируется запись в IPSR биты. 2 При чтении EPSR битов читаются нули, и запись в них игнорируется. Подробнее в описании инструкции «MRS» и «MSR»		

#### 11.4.6 APSR

Регистр APSR содержит текущие флаги состояния выполнения предыдущей инструкции.

Таблица 24 – Регистр APSR

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

Таблица 25 – Описание бит регистра APSR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	N	<b>Negative</b> 0 – результат операции положительный, нулевой, больше или равен 1 – результат операции отрицательный или меньше

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
30	Z	<b>Zero</b> 0 – результат операции не нулевой 1 – результат операции нулевой
29	C	<b>Carry</b> 0 – при суммировании не было переноса, при вычитании не было заема 1 – при суммировании был перенос, при вычитании был заем
28	V	<b>Overflow</b> 0 – в результате операции не было переполнения 1 – в результате операции было переполнение
27...0	-	Зарезервировано

#### 11.4.7 IPSR

Регистр IPSR содержит номер типа исключения для текущего обработчика прерывания.

Таблица 26 – Регистр IPSR

Номер	31...6	5...0
Доступ		
Сброс		
	-	ISR_NUMBER

Таблица 27 – Описание бит регистра IPSR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...6	-	Зарезервировано
5...0	ISR_NUMBER	Номер текущего исключения: 0 – режим Thread; 2 – NMI; 3 – Hard Fault; 11 – SVCcall; 14 – PendSV; 15 – SysTick; 16 – IRQ0; ... 47 – IRQ31

#### 11.4.8 EPSR

Регистр EPSR содержит бит состояния инструкции Thumb.

Таблица 28 – Регистр EPSR

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



Таблица 29 – Описание бит регистра EPSR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..25	-	Зарезервировано
24	T	Этот бит устанавливается в соответствии с вектором сброса, когда процессор выходит из состояния reset. Выполнение инструкции очистки T-бита регистра EPSR приводит к возникновению аппаратной ошибки Hard Fault. Это позволяет быть уверенным, что переключение в состояние ARM, не приведет к непредсказуемым последствиям
23..0	-	Зарезервировано

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

#### 11.4.9 PRIMASK

Регистр PRIMASK используется для повышения приоритета.

Таблица 30 – Регистр PRIMASK

Номер	31...1	0
Доступ		
Сброс		
	-	PRIMASK

Таблица 31 – Описание бит регистра PRIMASK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...1	-	Зарезервировано
0	PRIMASK	0 – не влияет; 1 – увеличивает приоритет исполнения до 0

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

#### 11.4.10 CONTROL

Контрольный регистр специального назначения. Регистр определяет текущий указатель стека.

Таблица 32 – Регистр CONTROL

Номер	31...2	1	0
Доступ			
Сброс			
	-	Active Stack Pointer	-

Таблица 33 – Описание бит регистра CONTROL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	-	Зарезервировано
1	Active Stack Pointer	0 – SP_main используется, как текущий указатель стека 1 – для режима Thread, SP_process используется, как текущий указатель стека *
0	-	Зарезервировано
* Попытка установить этот бит в режиме Handler игнорируется		

## 11.5 Типы данных

Процессор поддерживает следующие типы данных:

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

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

## 12 Система команд

В процессоре реализована версия системы команд Thumb. Поддерживаемые команды представлены в таблице 34.

В таблице используются следующие обозначения:

- в угловых скобках  $\langle \rangle$  записываются альтернативные формы представления операндов;
- в фигурных скобках  $\{ \}$  указываются необязательные операнды;
- информация в столбце «операнды» может быть неполной;

Более подробная информация представлена в детальном описании команд.

Таблица 34 – Система команд процессора

Мнемокод команды	Операнды	Краткое описание	Флаги
ADCS	{Rd,} Rn, Rm	Сложение с переносом	N, Z, C, V
ADD{S}	{Rd,} Rn, <Rm   #imm>	Сложение	N, Z, C, V
ADR	Rd, label	Загрузка адреса, заданного относительно счетчика команд	-
ANDS	{Rd,} Rn, Rm	Логическое И	N, Z
ASRS	{Rd,} Rm, <Rs   #imm>	Арифметический сдвиг вправо	N, Z, C
B{c}	label	Переход {суффикс условного исполнения}	-
BICS	{Rd,} Rn, Rm	Сброс битов по маске	N, Z
BKPT	#imm	Точка останова	-
BL	label	Переход со связью	-
BLX	Rm	Косвенный переход со связью	-
BX	Rm	Косвенный переход	-
CMN	Rn, Rm	Сравнить с противоположным знаком	N, Z, C, V
CMP	Rn, <Rm   #imm>	Сравнить	N, Z, C, V
CPSID	iflags	Изменить состояние процессора, запретить прерывания	-
CPSIE	iflags	Изменить состояние процессора, разрешить прерывания	-
CPY	Rd, Rm	Загрузка (аналогична MOV)	N, Z
DMB	-	Барьер синхронизации доступа к памяти данных	-
DSB	-	Барьер синхронизации доступа к памяти данных	-
EORS	{Rd,} Rn, Rm	Исключающее ИЛИ	N, Z
ISB	-	Барьер синхронизации доступа к инструкциям	-
LDM	Rn{!}, reglist	Загрузка множества регистров, инкремент после доступа	-
LDR	Rt, label	Загрузка слова в регистр, адрес задан относительно счетчика команд	-
LDR	Rt, [Rn, <Rm   #imm>]	Загрузка слова в регистр	-
LDRB	Rt, [Rn, <Rm   #imm>]	Загрузка байта в регистр	-

Мнемокод команды	Операнды	Краткое описание	Флаги
LDRH	Rt, [Rn, <Rm   #imm>]	Загрузка полуслова в регистр	-
LDRSB	Rt, [Rn, <Rm   #imm>]	Загрузка в регистр байта со знаком	-
LDRSH	Rt, [Rn, <Rm   #imm>]	Загрузка в регистр полуслова со знаком	-
LSL	{Rd,} Rn, <Rs   #imm>	Логический сдвиг влево	N, Z, C
LSR	{Rd,} Rn, <Rs   #imm>	Логический сдвиг вправо	N, Z, C
MOV{S}	Rd, Rm	Загрузка	N, Z
MRS	Rd, spec_reg	Считать специальный регистр в регистр общего назначения	-
MSR	spec_reg, Rm	Записать регистр общего назначения в специальный регистр	N, Z, C, V
MUL	Rd, Rn, Rm	Умножение, 32-разрядный результат	N, Z
MVNS	Rd, Rm	Загрузка инверсного значения	N, Z
NOP	-	Нет операции	-
ORR	{Rd,} Rn, Rm	Логическое ИЛИ	N, Z
POP	reglist	Извлечь регистры из стека	-
PUSH	reglist	Занести регистры в стек	-
REV	Rd, Rm	Изменить на обратный порядок байтов в слове	-
REV16	Rd, Rm	Изменить на обратный порядок байтов в полусловах	-
REVSH	Rd, Rm	Изменить на обратный порядок байт в младшем полуслове, произвести распространение знакового бита в старшее полуслово	-
ROR	{Rd,} Rn, Rs	Циклический сдвиг вправо	N, Z, C
RSBS	{Rd,} Rn, #0	Вычитание с противоположным порядком аргументов	N, Z, C, V
SBCS	{Rd,} Rn, Rm	Вычитание с учетом переноса	N, Z, C, V
SEV	-	Установить признак события	-
STM	Rn!, reglist	Сохранение множества регистров, инкремент после доступа	-
STR	Rt, [Rn, <Rm   #imm>]	Сохранение регистра, слово	-
STRB	Rt, [Rn, <Rm   #imm>]	Сохранение регистра, байт	-
STRH	Rt, [Rn, <Rm   #imm>]	Сохранение регистра, полуслово	-
SUB{S}	{Rd,} Rn, <Rm   #imm>	Вычитание	N, Z, C, V
SVC	#imm	Вызов супервизора	-
SXTB	Rd, Rm	Преобразовать байт со знаком в слово	-
SXTH	Rd, Rm	Преобразовать полуслово со знаком в слово	-
TST	Rn, Rm	Проверка значения битов по маске	N, Z
UXTB	Rd, Rm	Преобразовать байт без знака в слово	-
UXTH	Rd, Rm	Преобразовать полуслово без знака в слово	-
WFE	-	Ожидание события	-
WFI	-	Ожидание прерывания	-

## 12.1 Встроенные функции

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

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

Таблица 35 – Встроенные функции CMSIS, позволяющие генерировать некоторые инструкции процессора

Мнемокод команды процессора	Описание встроенной функции
CPSIE I	void__enable_irq(void)
CPSID I	void__disable_irq(void)
ISB	void__ISB(void)
DSB	void__DSB(void)
DMB	void__DMB(void)
NOP	void__NOP(void)
REV	uint32_t__REV(uint32_t int value)
REV16	uint32_t__REV16(uint32_t int value)
REVSH	uint32_t__REVSH(uint32_t int value)
SEV	void__SEV(void)
WFE	void__WFE(void)
WFI	void__WFI(void)

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

Таблица 36 – Встроенные функции CMSIS для доступа к специальным регистрам процессора

Наименование специального регистра	Режим доступа	Описание встроенной функции
PRIMASK	Чтение	uint32_t__get_PRIMASK (void)
	Запись	void__set_PRIMASK (uint32_t value)
CONTROL	Чтение	uint32_t__get_CONTROL (void)
	Запись	void__set_CONTROL (uint32_t value)
MSP	Чтение	uint32_t__get_MSP (void)
	Запись	void__set_MSP (uint32_t TopOfMainStack)
PSP	Чтение	uint32_t__get_PSP (void)
	Запись	void__set_PSP (uint32_t TopOfProcStack)

## 12.2 Описание инструкций

В разделе представлена подробная информация об инструкциях процессора:

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

### 12.2.1 Операнды

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

### 12.2.2 Ограничения на использование PC и SP

Многие инструкции не позволяют использовать или имеют ограничение на использование регистров счетчика команд (PC) и указателя стека (SP) в качестве операнда или регистра-получателя. Подробная информация содержится в описании конкретных инструкций.

Бит [0] адреса, загружаемого в PC с помощью одной из команд BX, BLX или POP, должен быть равен 1, так как этот бит указывает на требуемый набор команд, а процессор поддерживает только инструкции из набора Thumb. Когда команда BL или BLX записывает адрес в регистр LR, то биту [0] записываемого адреса автоматически присваивается значение 1.

## 12.3 Операции сдвига

Операции сдвига переносят значение битов содержимого регистра влево или вправо на заданное количество позиций – длина сдвига. Сдвиг может выполняться непосредственно с помощью инструкций ASR, LSR, LSL и ROR, при этом результат сдвига заносится в регистр-получатель.

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

### 12.3.1 ASR

Арифметический сдвиг вправо на  $n$  бит переносит крайние слева  $32-n$  бит регистра  $Rm$  вправо на  $n$  позиций, то есть на место крайних справа  $32-n$ . Бит [31] исходного значения регистра записывается в  $n$  крайних слева бит результата (см. рисунок 16).

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

При использовании инструкции  $ASRS$  флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита  $[n-1]$  регистра  $Rm$ .

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

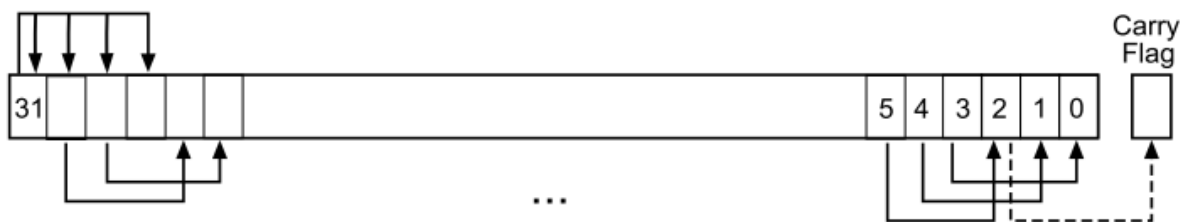


Рисунок 16 – Инструкция  $ASR \# 3$

### 12.3.2 LSR

Логический сдвиг вправо на  $n$  бит переносит крайние слева  $32-n$  бит регистра  $Rm$  вправо на  $n$  позиций, то есть на место крайних справа  $32-n$ . При этом в  $n$  крайних слева бит результата записывается 0 (см. рисунок 17).

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

При использовании инструкции  $LSRS$  флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита  $[n-1]$  регистра  $Rm$ .

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

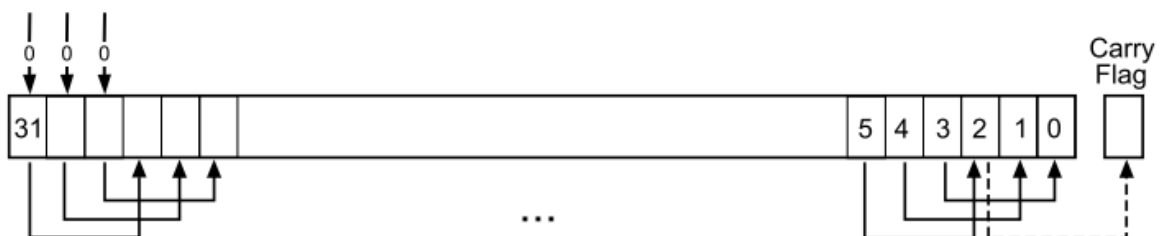


Рисунок 17 – Инструкция  $LSR \# 3$

### 12.3.3 LSL

Логический сдвиг влево на  $n$  бит переносит крайние справа 32- $n$  бит регистра  $R_m$  влево на  $n$  позиций, то есть на место крайних слева 32- $n$ . При этом в  $n$  крайних слева бит результата записывается 0 (см. рисунок 18).

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

При использовании инструкции  $LSLS$  флаг переноса принимает значение последнего бита, вытесненного в результате операции сдвига, то есть бита  $[32-n]$  регистра  $R_m$ . Инструкция  $LSL \# 0$  не влияет на значение флага переноса.

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

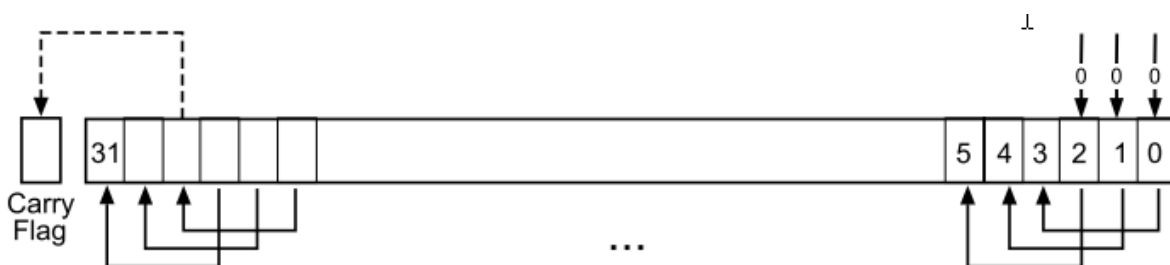


Рисунок 18 – Инструкция  $LSL \# 3$

### 12.3.4 ROR

Циклический сдвиг вправо на  $n$  бит переносит крайние слева 32- $n$  бит регистра  $R_m$  вправо на  $n$  позиций, то есть на место крайних справа 32- $n$ . При этом  $n$  крайних справа разрядов регистра переносятся в крайние  $n$  слева разрядов результата (см. рисунок 19).

При использовании инструкции  $RORS$  флаг переноса принимает значение последнего сдвинутого бита, то есть бита  $[n-1]$  регистра  $R_m$ .

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

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

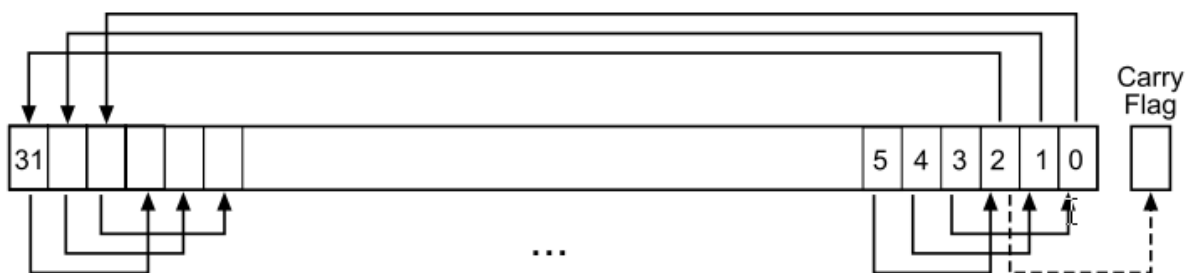


Рисунок 19 – Инструкция  $ROR \# 3$



## 12.4 Выравнивание адресов

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

Процессор не поддерживает доступ по невыровненным адресам. В связи с этим рекомендуется программно обеспечивать необходимое выравнивание данных.

При попытке доступа по невыровненному адресу процессором формируется исключение Hard Fault, что указано выставленным битом UNALIGN\_TRP регистра конфигурации и управления CCR (см. подпункт 27.1.5 «Регистр конфигурации и управления»). Более подробно вопрос формирования исключений рассмотрен в пункте 24.3 «Обработчики исключений».

## 12.5 Адресация относительно счетчика команд PC

В системе команд предусмотрена адресация команды или области данных в виде суммы значения счетчика команд PC плюс/минус численное смещение. Смещение вычисляется ассемблером автоматически, исходя из адреса метки и текущего адреса. В случае если смещение слишком велико, диагностируется ошибка.

Для большинства инструкций значение счетчика команд PC определяется как адрес текущей инструкции плюс четыре байта.

Ассемблер может поддерживать расширенные варианты синтаксиса для адресации относительно PC, например, «метка плюс/минус число» или выражения типа [PC, #imm].

## 12.6 Условное исполнение

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

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

Возможность исполнения или неисполнения команды, в зависимости от значения флагов условий, сформированных ранее, может быть достигнута за счет использования условных переходов. В таблице 68 представлен список суффиксов, которые можно добавить к инструкции для того, чтобы сделать ее условной. Условный переход может быть выполнен:

- сразу после команды, которая обновляет флаги условий;
- после любого количества промежуточных команд, которые не обновляют флаги условий.

Процессорное ядро поддерживает только одну инструкцию условного перехода: В<с> (Branch), где <с> - один из суффиксов условного исполнения.

Ниже в разделе рассматриваются:

- флаги условий;
- суффиксы условного исполнения.

### 12.6.1 Флаги условий

Регистр состояния прикладной программы APSR содержит следующие флаги:

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

Перенос возникает в следующих случаях:

- результат сложения оказался больше или равен  $2^{32}$ ;
- результат вычитания больше или равен нулю;
- в результате выполнения арифметического, логического или циклического сдвига.

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

Операция сравнения CMP аналогична операции вычитания, а операция сравнения CMN аналогична операции сложения, за исключением того, что результат отбрасывается.

### 12.6.2 Суффиксы условного исполнения

Условный переход в описании синтаксиса это обозначается как B{cond}. Команда перехода с кодом условия выполняется только в том случае, если флаги регистра APSR соответствуют указанному условию, в противном случае команда перехода игнорируется. В таблице 68 приведены доступные коды условий и соответствующие им флаги условий N, Z, C, V.

Таблица 37 – Суффиксы условного исполнения

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

## 12.7 Команды доступа к памяти

Обобщенные данные о командах доступа к памяти приведены в таблице 38.

Таблица 38 – Команды доступа к памяти

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

### 12.7.1 ADR

Загрузка адреса, заданного относительно счетчика команд.

#### Синтаксис

ADR Rd, label

где Rd – регистр-получатель;

label – относительный адрес, см. пункт 12.5 «Адресация относительно счетчика команд PC».

#### Описание

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

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

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

#### Ограничения

В качестве регистра Rd должен быть указан один из регистров R0-R7. Значение адреса должно быть выровнено на границу слова и задано в пределах от 0 до 1020 относительно текущего значения PC.

#### Флаги

Данная инструкция не влияет на состояние флагов.

### Примеры

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

ADR R3, [PC,#996] – Загрузить в регистр R3 результат сложения PC + 996.

## **12.8 LDR и STR, непосредственно заданное смещение**

Загрузка или сохранение регистра в режиме адресации со смещением.

### Синтаксис

LDR Rt, [<Rn | SP> {, #imm}]

LDR<B|H> Rt, [Rn {, #imm}]

STR Rt, [<Rn | SP>, {#imm}]

STR<B|H> Rt, [Rn {,#imm}]

где Rt – регистр, в который должна производиться загрузка, или регистр, значение которого должно быть сохранено;

Rn – регистр, содержащий базовый адрес памяти;

imm – смещение относительно базового адреса Rn. В случае, если смещение не указано, оно подразумевается равным нулю.

### Описание

Инструкции LDR, LDRB и LDRH загружают из памяти в регистр Rt слово, байт и полуслово соответственно. При загрузке байта и полуслова значение, записываемое в регистр Rt, расширяется нулями до слова.

Инструкции STR, STRB и STRH сохраняют в память из регистра Rt слово, младший байт и младшее полуслово соответственно.

Адрес памяти для инструкций загрузки и сохранения рассчитывается как сумма значения в регистре Rn или SP и непосредственно заданного смещения imm.

### Ограничения

Для данных команд:

- в качестве Rt и Rn можно использовать только регистры R0-R7;
- смещение imm должно быть задано:
  - целым числом от 0 до 1020, кратным 4, для инструкций LDR и STR с использованием SP в качестве регистра, содержащий базовый адрес памяти;
  - целым числом от 0 до 124, кратным 4, для инструкций LDR и STR с использованием R0-R7 в качестве регистра, содержащий базовый адрес памяти;
  - целым числом от 0 до 62, кратным 2, для инструкций LDRH и STRH;
  - целым числом от 0 до 31 для инструкций LDRB и STRB;
- вычисленный адрес должен делиться без остатка на количество байт в транзакции, см. пункт 12.4 «Выравнивание адресов».

### Флаги

Данные команды не влияют на состояние флагов.

### Примеры

LDR R4, [R7] – Загрузка регистра R4 из ячейки по адресу, содержащемуся в R7;

STR R2, [R0,#const-struc] – Сохранение значения из регистра R2 по адресу, вычисленному как сумма значения в R0 и const-struc. Const-struc – выражение с постоянным значением, лежащим в диапазоне 0-124.

### **12.8.1 LDR и STR, смещение задано в регистре**

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

### Синтаксис

LDR Rt, [Rn, Rm]

LDR<B|H> Rt, [Rn, Rm]

LDR<SB|SH> Rt, [Rn, Rm]

STR Rt, [Rn, Rm]

STR<B|H> Rt, [Rn, Rm]

где Rt – регистр, в который должна производиться загрузка, или регистр, значение которого должно быть сохранено;

Rn – регистр, содержащий базовый адрес памяти;

Rm – регистр, содержащий смещение относительно базового адреса.

### Описание

Инструкции LDR, LDRB, LDRH, LDRSB и LDRSH загружают из памяти в регистр Rt следующее значение:

- LDR – загрузка слова;
- LDRB – загрузка байта, расширенного нулями до слова;
- LDRH – загрузка полуслова, расширенного нулями до слова;
- LDRSB – загрузка байта, расширенного знаковым битом до слова;
- LDRSH – загрузка полуслова, расширенного знаковым битом до слова.

Инструкции STR, STRB и STRH сохраняют в память из регистра Rt слово, младший байт и младшее полуслово соответственно.

Адрес памяти для инструкций загрузки и сохранения рассчитывается как сумма значений в регистре Rn и Rm.

### Ограничения

Для данных команд:

- в качестве операндов Rt, Rn и Rm можно использовать только регистры R0-R7;
- вычисленный адрес должен делиться без остатка на количество байт в транзакции, см. пункт 12.4 «Выравнивание адресов».

Флаги

Данные команды не влияют на состояние флагов.

Примеры

STR R0, [R5, R1] – Записать значение в регистре R0 по адресу, равному сумме значений в R5 и R1;

LDRSH R1, [R2, R3] – Считать полуслово по адресу, равному сумме значений в R2 и R3, распространить значение знакового бита на старшие значащие байты слова, загрузить результат в регистр R1.

**12.8.2 LDR, адресация относительно счетчика команд PC**

Загрузка регистра из памяти.

Синтаксис

LDR Rt, label

где Rt – регистр, в который должна производиться загрузка;

Label – относительный адрес, см. пункт 12.5 «Адресация относительно счетчика команд PC».

Описание

Загружает в регистр Rt слово из памяти по адресу, заданному в виде метки, относительно счетчика команд PC.

Ограничения

Значение смещения, указанное в метке, должно быть задано целым числом от 0 до 1020 и быть кратным 4.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

LDR R0, LookUpTable – Загрузить регистр R0 словом данных по адресу с меткой LookUpTable;

LDR R3, [PC, #100] – Загрузить регистр R3 словом данных по адресу PC + 100.

**12.8.3 LDM и STM**

Загрузка или сохранение множества регистров.

Синтаксис

LDM Rn{!}, reglist

STM Rn!, reglist

где  $R_n$  – регистр, содержащий базовый адрес памяти;

! – суффикс обратной записи значения базового регистра. В случае если он присутствует в команде, последний адрес, по которому осуществлялся доступ, будет записан обратно в регистр  $R_n$ ;

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

Мнемокоды LDMIA и LDMFD – это псевдокоманды LDM. Командой LDMIA обозначают загрузку множества регистров с увеличением значения адреса в  $R_n$  после каждого доступа (Increment After). Командой LDMFD обозначают извлечение данных из полного нисходящего стека с указателем на последний загруженный элемент (Full Descending stack);

Мнемокоды STMIA и STMEA – это псевдокоманды STM. Командой STMIA обозначают сохранение множества регистров с увеличением значения адреса в  $R_n$  после каждого доступа (Increment After). Командой STMEA обозначают сохранение данных в пустой восходящий стек с указателем на последнюю свободную ячейку (Empty Ascending stack).

### Описание

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

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

Команды LDM, LDMIA, LDMFD, STM, STMIA и STMEA для доступа используют адреса памяти в интервале от  $R_n$  до  $R_n + 4 \cdot (n - 1)$ , где  $n$  – количество регистров в списке reglist. Доступ осуществляется в порядке увеличения номера регистра, при этом регистр с наименьшим номером соответствует наименьшему адресу памяти, а регистр с наибольшим номером – наибольшему адресу. Если указан суффикс обратной записи, то значение  $R_n + 4 \cdot n$  записывается обратно в регистр  $R_n$ .

### Ограничения

В описываемых в разделе командах:

- в списке reglist и в качестве  $R_n$  можно использовать только регистры R0-R7;
- суффикс обратной записи должен использоваться всегда. Исключение составляет команда LDM, в которой в списке reglist содержится регистр  $R_n$ . В этом случае суффикс обратной записи использовать нельзя;
- значение адреса в регистре  $R_n$  должно быть выровнено на границу слова, см. пункт 12.4 «Выравнивание адресов»;
- если в команде STM регистр  $R_n$  указан в списке reglist, то в таком случае  $R_n$  должен быть первым регистром в списке, т.е. иметь наименьший номер.

### Флаги

Данные команды не влияют на состояние флагов.

### Примеры

LDM R0,{R0,R3,R4} – LDMIA – синоним LDM;  
STMIA R1!,{R2-R4,R6}.

### Примеры неправильного использования

STM R5!,{R4,R5,R6} – Сохраненное значение R5 является;  
LDM R2!, {} – Список должен содержать хотя бы один регистр.

## 12.8.4 PUSH и POP

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

### Синтаксис

PUSH reglist

POP reglist

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

### Описание

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

Команда POP восстанавливает значения регистров из стека в порядке увеличения номеров регистров, при этом регистр с наименьшим номером считывается из памяти с наименьшим значением адреса.

Команда PUSH использует значение в регистре SP минус четыре в качестве наибольшего адреса памяти для сохранения регистров в стек. Команда POP использует значение в регистре SP в качестве наименьшего адреса памяти для загрузки регистров из стека. Таким образом реализуется полный нисходящий стек (Full Descending stack).

По завершении команды PUSH регистр SP обновляется таким образом, чтобы он указывал на расположение последнего сохраненного значения, имеющего наименьший адрес в памяти. По завершении команды POP регистр SP обновляется таким образом, чтобы он указывал на расположение выше последнего загруженного значения, имеющего наибольший адрес в памяти.

В случае, если команда POP содержит в списке reglist регистр счетчика команд PC, то переход будет выполнен после завершения POP. Бит [0] загружаемого значения в регистр PC должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу.



Ограничения

В данных командах:

- в списке регистров reglist можно использовать только регистры R0-R7;
- исключением является регистр LR для команды PUSH, а также регистр PC для команды POP.

Флаги

Данные команды не влияют на состояние флагов.

Примеры

PUSH {R0,R4-R7} – Сохранение значений регистров R0, R4, R5, R6 и R7 в стек;

PUSH {R2,LR} – Сохранение значений регистров R0 и LR в стек;

POP {R0,R6,PC} – Загрузка регистров R0, R6 и PC значениями из стека, после выполнения команды POP будет выполнен переход по новому адресу, загруженному в регистр PC.

**12.9 Инструкции обработки данных**

Таблица 39 показывает инструкции обработки данных.

Таблица 39 – Команды обработки данных

Мнемокод	Краткое описание
ADCS	Сложение с учетом переноса
ADD{S}	Сложение
ANDS	Логическое И
ASRS	Арифметический сдвиг вправо
BICS	Сброс битов по маске
CMN	Сравнить с противоположным знаком
CMP	Сравнить
EORS	Исключающее ИЛИ
LSLS	Логический сдвиг влево
LSRS	Логический сдвиг вправо
MOV{S}	Загрузка
MULS	Умножение
MVNS	Загрузка инверсного значения
ORRS	Логическое ИЛИ
REV	Изменить на обратный порядок байтов в слове
REV16	Изменить на обратный порядок байтов в полусловах
REVSH	Изменить на обратный порядок байт в младшем полуслове, произвести распространение знакового бита в старшее полуслово
RORS	Циклический сдвиг вправо
RSBS	Вычитание с противоположным порядком аргументов
SBCS	Вычитание с учетом переноса
SUBS	Вычитание
SXTB	Заполнение знаком байта

Мнемокод	Краткое описание
SXTH	Заполнение знаком полуслова
UXTB	Заполнение нулем байта
UXTH	Заполнение нулем полуслова
TST	Проверка значения битов по маске

### 12.9.1 ADD, ADC, SUB, SBC и RSB

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

Примечание – Процессорное ядро поддерживает инструкции ADC, SBC и RSB только как инструкции, которые обновляют флаги, то есть инструкции – ADCS, SBCS и RSBS.

#### Синтаксис

ADD{S} {Rd,} Rn, <Rm|#imm>

ADCS {Rd,} Rn, Rm

SUB{S} {Rd,} Rn, <Rm|#imm>

SBCS {Rd,} Rn, Rm

RSBS {Rd,} Rn, #0

где S – необязательный суффикс для инструкций ADD и SUB. Если он указан, результат выполнения операции приводит к обновлению флагов, см. пункт 12.6 «Условное исполнение»;

Rd – регистр-получатель результата;

Rn – регистр, содержащий значение первого операнда;

Rm – регистр, содержащий значение второго операнда;

imm – определяет непосредственное значение константы.

В случае если регистр Rd не указан, то результат записывается в Rn. Например, запись ADDS R1, R2 равносильна ADDS R1, R1, R2.

#### Описание

Команда ADD складывает значение Rn со значением регистра Rm или значением imm, результат записывается в Rd.

Команда ADDS выполняет то же самое, что и ADD, а также обновляет флаги N, Z, C, V.

Команда ADCS складывает значение Rn со значением регистра Rm. Если флаг переноса установлен, то к результату также добавляется единица. Результат записывается в регистр Rd, при этом обновляются флаги N, Z, C, V.

Команда SUB вычитает значение Rm или imm из значения регистра Rn. Результат помещается в регистр Rd.

Команда SUBS выполняет то же самое, что и SUB, а также обновляет флаги N, Z, C, V.

Команда SBCS вычитает значение Rm из значения регистра Rn. Если флаг переноса установлен, то из результата также вычитается единица. Результат записывается в регистр Rd, при этом обновляются флаги N, Z, C, V.

Команда RSBS вычитает значение Rn из нуля, результат записывает в регистр Rd, при этом также обновляет флаги N, Z, C, V.

Инструкции ADC и SBC полезны при реализации вычислений с повышенной разрядностью, см. «Примеры».

См. также описание команды «ADR».

### Ограничения

В таблице 40 представлены допустимые комбинации регистров и значений констант imm, которые можно использовать применительно к указанным инструкциям.

Таблица 40 – Ограничения команд ADC, ADD, RSB, SBC и SUB

Инструкция	Rd	Rn	Rm	imm	Ограничения
ADCS	R0-R7	R0-R7	R0-R7	-	Rd и Rn должны задавать один и тот же регистр
ADD	R0-R15	R0-R15	R0-R15	-	Rd и Rn должны задавать один и тот же регистр Rn и Rm не должны оба задавать PC (R15)
	R0-R7	SP или PC	-	0-1020	Значение imm должно быть целым числом кратным 4
	SP	SP	-	0-508	Значение imm должно быть целым числом кратным 4
ADDS	R0-R7	R0-R7	-	0-7	-
	R0-R7	R0-R7	-	0-255	Rd и Rn должны задавать один и тот же регистр
	R0-R7	R0-R7	R0-R7	-	-
RSBS	R0-R7	R0-R7	-	-	-
SBCS	R0-R7	R0-R7	R0-R7	-	Rd и Rn должны задавать один и тот же регистр
SUB	SP	SP	-	0-508	Значение imm должно быть целым числом кратным 4
SUBS	R0-R7	R0-R7	-	0-7	-
	R0-R7	R0-R7	-	0-255	Rd и Rn должны задавать один и тот же регистр
	R0-R7	R0-R7	R0-R7	-	-

### Примеры

#### 64-разрядное сложение

Следующий пример показывает, как осуществить сложение 64-разрядного целого числа, записанного в паре регистров R0 и R1, с другим 64-разрядным числом, записанным в паре регистров R2 и R3. Результат записывается в пару регистров R0 и R1.

ADDS R0, R0, R2 – сложить младшие значащие слова;

ADCS R1, R1, R3 – сложить старшие значащие слова с учетом флага переноса.

### 96-разрядное вычитание

Данные с повышенной разрядностью не обязательно содержать в смежных регистрах. В примере, приведенном ниже, показан фрагмент кода, осуществляющий вычитание 96-разрядного целого числа, записанного в регистрах R1, R2 и R3, из другого числа, содержащегося в R4, R5 и R6. Результат записывается в регистрах R4, R5 и R6.

SUBS R4, R4, R1 – вычитание младших значащих слов;

SBCS R5, R5, R2 – вычитание средних значащих слов с учетом флага переноса;

SBCS R6, R6, R3 – вычитание старших значащих слов с учетом флага переноса.

Во фрагменте кода ниже приведен пример использования команды RSBS.

RSBS R7, R7, #0 ; вычитание R7 из нуля.

## 12.9.2 AND, ORR, EOR, BIC

Логические операции И, ИЛИ, исключающее ИЛИ и сброс битов по маске.

Примечание – Процессорное ядро поддерживает инструкции AND, ORR, EOR, BIC только как инструкции, которые обновляют флаги, то есть инструкции – ANDS, ORRS, EORS, BICS.

### Синтаксис

ANDS {Rd,} Rn, Rm

ORRS {Rd,} Rn, Rm

EORS {Rd,} Rn, Rm

BICS {Rd,} Rn, Rm

где Rd – регистр назначения;

Rn – регистр, который содержит первый операнд, при этом он совпадает с регистром назначения;

Rm – второй регистр.

### Описание

Инструкции AND, ORR и EOR осуществляют, соответственно, операции побитового И, ИЛИ и исключающего ИЛИ между аргументами, содержащимися в регистрах Rn и Rm.

Инструкция BIC выполняет операцию побитового И между аргументом, содержащимся в регистре Rn, и инверсным значением второго операнда Rm.

### Ограничения

В качестве операндов Rd, Rn, Rm можно использовать только R0 – R7.

### Флаги

Данные инструкции:

- обновляют флаги N и Z в соответствии с результатом выполнения операции;
- не изменяют значения флагов C и V.

Примеры

ANDS R2, R2, R1  
 ORRS R2, R2, R5  
 ANDS R5, R5, R8  
 EORS R7, R7, R6  
 BICS R0, R0, R1

**12.9.3 ASR, LSL, LSR, ROR**

Арифметический сдвиг вправо, логический сдвиг влево, логический сдвиг вправо, циклический сдвиг вправо.

Синтаксис

ASRS {Rd,} Rm, Rs  
 ASRS {Rd,} Rm, #imm  
 LSLS {Rd,} Rm, Rs  
 LSLS {Rd,} Rm, #imm  
 LSRS {Rd,} Rm, Rs  
 LSRS {Rd,} Rm, #imm  
 RORS {Rd,} Rm, Rs

где Rd – регистр-получатель результата. Если Rd не указан, то результат записывается в Rm;

Rm – регистр, значение которого должно быть подвергнуто сдвигу;

Rs – регистр, содержащий величину сдвига значения регистра Rm;

imm – длина сдвига. Диапазон допустимых значений параметра зависит от инструкции:

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

Примечание – Инструкция MOVS Rd, Rm является псевдонимом инструкции LSLS Rd, Rm, #0.

Описание

Инструкции ASR, LSL, LSR и ROR выполняют арифметический сдвиг влево, логический сдвиг влево, логический сдвиг вправо и циклический сдвиг вправо на заданное количество позиций, определяемое константой imm или значением наименее значимого байта регистра Rs.

Детальное описание операций сдвига представлено в пункте 12.3 «Операции сдвига».

### Ограничения

В данных командах в качестве операндов Rd, Rm и Rs можно использовать только регистры R0-R7.

В командах, в которых величина сдвига задается значением регистра Rs, в качестве Rd и Rm должен использоваться один и тот же регистр.

### Флаги

Данные инструкции обновляют флаги N и Z в соответствии с результатом.

Флаг C обновляется до значения последнего сдвинутого бита, кроме случая, когда длина сдвига равна нулю, см. пункт 12.3 «Операции сдвига». Флаг V не изменяется.

### Примеры

ASRS R7, R5, #9 – Арифметический сдвиг вправо на девять бит;

LSLS R1, R2, #3 – Логический сдвиг влево на три бита с установкой флагов;

LSRS R4, R5, #6 – Логический сдвиг вправо на шесть бит;

RORS R4, R4, R6 – Циклический сдвиг вправо на значение, указанное в младшем байте регистра R6.

## 12.9.4 CMP и CMN

Сравнение и сравнение с противоположным знаком.

### Синтаксис

CMN Rn, Rm

CMP Rn, #imm

CMP Rn, Rm

где Rn – регистр, хранящий значение первого операнда;

Rm – регистр, с которым нужно проводить сравнение;

imm – значение, с которым нужно проводить сравнение.

### Описание

Данные инструкции осуществляют сравнение значений регистра Rn со значением, хранящимся в регистре Rm, или со значением imm. По результатам сравнения устанавливаются соответствующие флаги, однако сам результат в регистр не записывается.

Команда CMP вычитает из регистра Rn значение регистра Rm или значение константы imm и обновляет флаги. Она аналогична инструкции SUBS, за исключением того, что результат вычитания не сохраняется.

Команда CMN складывает значение регистра Rm и значение регистра Rn и обновляет флаги. Она аналогична инструкции ADDS, за исключением того, что результат сложения не сохраняется.

### Ограничения

Для инструкции CMN в качестве Rn и Rm можно использовать только R0 – R7.

Для инструкции CMP:

- в качестве Rn и Rm могут быть использованы только R0 – R14;
- значение константы imm должно лежать в диапазоне 0-255.

### Флаги

Данные инструкции устанавливают флаги N, Z, C и V в соответствии с результатом сравнения.

### Примеры

CMP R2, R9

CMN R0, R2

## 12.9.5 MOV и MVN

Загрузка в регистр прямого или инверсного значения

### Синтаксис

MOV{S} Rd, Rm

MOVS Rd, #imm

MVNS Rd, Rm

где S – необязательный суффикс. Если он указан, результат выполнения операции приводит к обновлению соответствующих флагов, см. пункт 12.6 «Условное исполнение»;

Rd – регистр-получатель результата;

Rm – регистр-источник данных;

imm – любое значение в диапазоне 0-255.

### Описание

Инструкция MOV копирует значение, записанное в регистре Rm, в регистр Rd.

Инструкция MOVS выполняет ту же операцию, что и инструкция MOV, но при этом обновляет значения флагов N и Z.

Инструкция MVNS считывает значение операнда Rm, производит его побитную инверсию, после чего помещает результат в регистр Rd.

### Ограничения

В данных инструкциях в качестве Rd и Rm можно использовать только R0 – R7.

Если при вызове инструкции MOV регистром Rd является счетчик команд PC, то:

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

Примечание – Несмотря на то, что инструкцию MOV можно использовать в качестве инструкции перехода, рекомендуется использовать инструкции BX или BLX для реализации перехода, чтобы обеспечить переносимость программного обеспечения.

#### Флаги

В случае, если указан суффикс S, то инструкция:

- обновляет флаги N и Z в соответствии с результатом выполнения операции;
- не изменяет значения флагов C и V.

#### Примеры

MOVS R0, #0x000B – Записать значение 0x000B в R0, флаги обновляются;

MOVS R1, #0x0 – Записать значение нуля в регистр R1, флаги обновляются;

MOV R10, R12 – Записать значение регистра R12 в R10, флаги не обновляются;

MOVS R3, #23 – Записать значение 23 в R3;

MOV R8, SP – Записать значение указателя стека в регистр R8;

MVNS R2, R0 – Записать инверсное значение R0 в R2 и обновить флаги.

### 12.9.6 MULS

Умножение с использованием 32-битных операндов. Результат имеет 32-битную разрядность.

#### Синтаксис

MULS Rd, Rn, Rm

где Rd – регистр-получатель результата;

Rn, Rm – регистры, содержащие значения, которые нужно перемножить.

#### Описание

Инструкция MULS производит умножение значений регистров Rn и Rm и помещает младшие 32 бита результата в регистр Rd. Флаги обновляются в соответствии с результатом выполнения операции, см. пункт 12.6 «Условное исполнение».

Результат выполнения операции не зависит от того, используются ли знаковые или беззнаковые операнды.

#### Ограничения

В качестве операндов Rd, Rn, Rm можно использовать только регистры R0 – R7.

Rd должен быть тем же регистром, что и Rm.

#### Флаги

Данная инструкция:

- обновляет флаги N и Z в соответствии с результатом;
- не изменяет значения флагов C и V.

#### Примеры

MULS R0, R2, R0 – умножение с обновлением флагов,  $R0 = R0 \times R2$ .



### 12.9.7 REV, REV16, REVSH

Изменение порядка байтов в слове.

#### Синтаксис

REV Rd, Rn

REV16 Rd, Rn

REVSH Rd, Rn

где Rd – регистр-получатель результата;

Rn – регистр-источник данных.

#### Описание

Инструкции предназначены для изменения формата представления (endianness) данных:

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

#### Ограничения

В данных инструкциях в качестве операндов Rd и Rn можно использовать только регистры R0 – R7.

#### Флаги

Данные инструкции не влияют на состояние флагов.

#### Примеры

REV R3, R7 – Изменить на обратный порядок байтов в R7, результат записать в R3;

REV16 R0, R0 – Изменить на обратный порядок байтов в каждом 16-разрядном полуслове R0;

REVSH R0, R5 – Изменить на обратный порядок байтов в полуслове R5 со знаком, 32-разрядный результат записать в R0.

### 12.9.8 SXT и UXT

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

Синтаксис

SXTB Rd, Rm

SXTH Rd, Rm

UXTB Rd, Rm

UXTH Rd, Rm

где Rd – регистр-получатель результата;

Rm – регистр, содержащий значение, которое нужно дополнить.

Описание

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

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

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

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

Ограничения

В данных инструкция в качестве операндов Rd и Rm можно использовать только регистры R0 – R7.

Флаги

Данные инструкции не влияют на состояние флагов.

Примеры

SXTH R4, R6 – Извлечь младшие 16 бит значения, записанного в R6, распространить знак в полученном полуслове до 32 бит, записать результат в R4;

UXTB R3, R1 – Извлечь младший байт значения, записанного в R1, дополнить полученный байт нулями до 32 бит, результат записать в R3.

**12.9.9 TST**

Проверить значение битов по маске.

Синтаксис

TST Rn, Rm

где Rn – регистр, содержащий первый операнд.

Rm – регистр, использующийся для проверки соответствия.

Описание

Данная инструкция позволяет проверить значение регистра с учетом значения, которое содержится в другом регистре. По результату проверки обновляются флаги, сам результат не сохраняется.

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

Для того, чтобы проверить, имеет ли бит регистра Rn значение «0» или «1», используйте инструкцию TST совместно с регистром Rm, в котором этот бит установлен в «1», а все остальные биты имеют значение «0».

Ограничения

В качестве операндов Rd и Rm можно использовать только регистры R0 – R7.

Флаги

Данная инструкция:

- обновляет флаги N и Z в соответствии с результатом;
- не изменяет значения флагов C и V.

Примеры

TST R0, R1 – Выполняет побитовое И между R0 и R1, обновляет флаги, результат не сохраняется.

**12.10 Инструкции передачи управления**

Таблица 41 показывает список инструкций передачи управления.

Таблица 41 – Инструкции передачи управления

Мнемокод команды	Краткое описание
B{с}	Переход {суффикс условного исполнения}
BL	Переход со связью
BLX	Косвенный переход со связью
BX	Косвенный переход

**12.10.1 B, BL, BX и BLX**

Команды ветвления.

Синтаксис

B {cond} label

BL label

BX Rm

BLX Rm

где cond – необязательный код условия, см. пункт 12.6 «Условное исполнение»;

label – относительный адрес, см. пункт 12.5 «Адресация относительно счетчика команд PC»;

Rm – регистр, содержащий адрес, по которому необходимо передать управление. Бит [0] этого регистра должен быть установлен в «1», однако передача управления будет выполнена по адресу, соответствующему значению бита [0], равному «0».

Описание

Все рассматриваемые в данном разделе инструкции осуществляют передачу управления по адресу, заданному меткой, либо содержащемуся в регистре Rm. Кроме того:

- команды VL и BLX записывают адрес следующей инструкции в регистр связи LR (R14);
- команды VX и BLX формируют отказ (Hard fault) в случае, если бит [0] регистра Rm равен «0».
- Инструкции VL и BLX также устанавливают бит [0] регистра LR в «1». Это гарантирует, что при использовании данного значения инструкциями POP {PC} или VX будет выполнен успешный переход.

Таблица 42 показывает диапазон адресуемых переходов для различных команд ветвления.

Таблица 42 – Диапазон адресуемых переходов для команд ветвления

Инструкция	Диапазон адресации
B label	от -2 Кбайт до +2 Кбайт относительно текущей позиции
B {cond} label	от -256 байт до +254 байт относительно текущей позиции
VL label	от -16 Мбайт до +16 Мбайт относительно текущей позиции
VX Rm	любое значение, записанное в регистре Rm
BLX Rm	любое значение, записанное в регистре Rm

Ограничения

- В командах VX и BLX не допускается использование регистра PC и SP;
- В командах VX и BLX бит [0] регистра Rm должен быть установлен в «1», при этом передача управления будет, выполнена по адресу, соответствующему значению бита [0], равному «0»;
- B {cond} – единственная условно исполняемая команда.

Флаги

Данные инструкции не влияют на состояние флагов.

Примеры

- B loorA – передача управления по адресу, обозначенному меткой loorA;
- VL funC – переход со связью (вызов функции) в funC, адрес возврата будет записан в регистр LR;
- VX LR – возврат из функции;
- BLX R0 – переход со связью (вызов функции) по адресу, записанному в R0;
- BEQ labelD – условный переход на метку labelD, если последняя инструкция, изменяющая флаги в регистре APSR, установила флаг Z, иначе переход выполнен не будет.

## 12.11 Прочие инструкции

Таблица 43 показывает список не рассмотренных в предыдущих разделах инструкций процессора:

Таблица 43 – Прочие инструкции

Мнемокод команды	Краткое описание
BKPT	Точка останова
CPSID	Изменить состояние процессора, запретить прерывания
CPSIE	Изменить состояние процессора, разрешить прерывания
CPY	Аналогична MOV
DMB	Барьер синхронизации доступа к памяти данных
DSB	Барьер синхронизации доступа к памяти данных
ISB	Барьер синхронизации доступа к инструкциям
MRS	Загрузка из специального регистра в регистр общего назначения
MSR	Загрузка из регистра общего назначения в специальный регистр
NOP	Нет операции
SEV	Установить признак события
SVC	Вызов супервизора
WFE	Ожидать событие
WFI	Ожидать прерывание

### 12.11.1 BKPT

Точка останова.

#### Синтаксис

BKPT #imm

где imm – целое число в диапазоне от 0 до 255.

#### Описание

Команда BKPT переводит процессор в состояние отладки. Инструменты отладки могут использовать эту возможность для исследования состояния системы в определенных местах программы, вставляя команду BKPT по требуемому адресу. Значение imm игнорируется процессором. При необходимости отладчик может использовать значение imm для хранения дополнительной информации о точке останова.

#### Ограничения

Данная инструкция не имеет ограничений.

#### Флаги

Данная инструкция не влияет на состояние флагов.

#### Примеры

BKPT #0 – Точка останова со значением 0x00.

### 12.11.2 CPS

Изменить состояние процессора.

#### Синтаксис

CPSID i

CPSIE i

#### Описание

Команда CPS позволяет изменить значение специального регистра PRIMASK. Команда CPSID устанавливает специальный регистр PRIMASK в 1, отключая прерывания. Команда CPSIE сбрасывает специальный регистр PRIMASK в 0, включая прерывания.

#### Ограничения

Данная инструкция не имеет ограничений.

#### Флаги

Данная инструкция не влияет на состояние флагов.

#### Примеры

CPSID i – Запретить все прерывания, кроме NMI (установить PRIMASK);

CPSIE i – Разрешить прерывания (сбросить PRIMASK).

### 12.11.3 DMV

Барьер доступа к памяти данных.

#### Синтаксис

DMV

#### Описание

Команда DMV выполняет функцию барьера доступа к памяти для синхронизации данных. Она гарантирует, что все явные операции доступа к памяти, которые были инициированы перед выполнением инструкции DMV, будут завершены до того, как начнется выполнение любой явной операции доступа к памяти после этой инструкции.

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

#### Ограничения

Данная инструкция не имеет ограничений.

#### Флаги

Данная инструкция не влияет на состояние флагов.

#### Примеры

DMV – Барьер доступа к памяти данных.

#### 12.11.4 DSB

Барьер синхронизации доступа к памяти данных.

##### Синтаксис

DSB

##### Описание

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

##### Ограничения

Данная инструкция не имеет ограничений.

##### Флаги

Данная инструкция не влияет на состояние флагов.

##### Примеры

DSB – Барьер синхронизации доступа к памяти данных.

#### 12.11.5 ISB

Барьер синхронизации доступа к инструкциям.

##### Синтаксис

ISB

##### Описание

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

##### Ограничения

Данная инструкция не имеет ограничений.

##### Флаги

Данная инструкция не влияет на состояние флагов.

##### Примеры

ISB – Барьер синхронизации доступа к инструкциям.

### 12.11.6 MRS

Считать содержимое специального регистра в регистр общего назначения.

#### Синтаксис

MRS Rd, spec\_reg

где Rd – регистр-получатель результата.

spec\_reg – один из специальных регистров: APSR, IPSR, EPSR, IEPSR, IAPSR, EAPSR, PSR, MSP, PSP, PRIMASK или CONTROL.

#### Описание

Команда MRS сохраняет содержимое специального регистра в регистр общего назначения Rd. Команда MRS может быть объединена с командой MSR для выполнения последовательности чтения-модификации-записи, например, для изменения требуемого флага в PSR.

См. также описание инструкции «MSR».

#### Ограничения

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

#### Флаги

Данная инструкция не влияет на состояние флагов.

#### Примеры

MRS R0, PRIMASK – Считать значение PRIMASK и записать это в R0.

### 12.11.7 MSR

Записать регистр общего назначения в специальный регистр.

#### Синтаксис

MSR spec\_reg, Rn

где Rn – регистр-источник данных.

spec\_reg – один из специальных регистров: APSR, IPSR, EPSR, IEPSR, IAPSR, EAPSR, PSR, MSP, PSP, PRIMASK или CONTROL.

#### Описание

Команда MSR загружает содержимое регистра общего назначения Rn в специальный регистр.

См. также описание инструкции «MRS».

#### Ограничения

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



Флаги

Данная инструкция обновляет флаги на основе значения в регистре Rn.

Примеры

MSR CONTROL, R1 – Считать значение из регистра R1 и записать это значение в регистр CONTROL.

**12.11.8 NOP**

Нет операции.

Синтаксис

NOP

Описание

Команда NOP не выполняет никаких операций. Процессор может автоматически исключить NOP из конвейера команд до того, как команда достигнет стадии выполнения.

Команду NOP рекомендуется использовать для заполнения, например, с целью разместить очередную инструкцию по адресу, выровненному по 64-битной границе.

Ограничения

Данная инструкция не имеет ограничений.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

NOP – Нет операции.

**12.11.9 SEV**

Установить признак события.

Синтаксис

SEV

Описание

Команда SEV сигнализирует о событии всем процессорам в составе многопроцессорной системы. Кроме того, она устанавливает собственный регистр события в 1.

См. также описание инструкции WFE.

Ограничения

Данная инструкция не имеет ограничений.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

SEV – Послать признак события.

**12.11.10 SVC**

Вызов супервизора.

Синтаксис

SVC #imm

где imm – целое число в диапазоне от 0 до 255.

Описание

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

Ограничения

Данная инструкция не имеет ограничений.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

SVC 0x32 – Вызов супервизора (функция обработчика исключения SVC может извлечь параметр imm, прочитав по сохраненному в стеке адресу PC значение команды SVC).

**12.11.11 WFE**

Ожидание события.

Описание

Если в регистре события записан «0», то WFE временно прекращает исполнение команд до тех пор, пока не возникнет одно из следующих событий:

- исключение, кроме тех случаев, когда оно замаскировано регистрами, маскирующими прерывания, или текущим уровнем приоритета;
- исключение переходит в состояние ожидания обработки запроса, при этом бит SEVONPEND установлен в регистре System Control;
- запрос на вход в режим отладки, если режим отладки разрешен;
- событие, о котором сигнализирует периферия или другой процессор в мультипроцессорной системе при помощи инструкции SEV.

Если в регистре события записана «1», то инструкция WFE очищает его (устанавливает «0») и немедленно завершается.

Примечание – Инструкция WFE предназначена только для режима энергосбережения. При написании программного обеспечения необходимо принимать во внимание, что эта инструкция может вести себя как NOP.

#### Ограничения

Данная инструкция не имеет ограничений.

#### Флаги

Данная инструкция не влияет на состояние флагов.

#### Примеры

WFE – Ожидание события.

### **12.11.12 WFI**

Ожидание прерывания.

#### Описание

Инструкция WFI приостанавливает исполнение команд, пока не возникнет одно из следующих событий:

- исключение;
- появление отложенного прерывания, которое будет отменено, если регистр PRIMASK был очищен;
- запрос на вход в режим отладки Debug.

Примечание – Инструкция WFI предназначена только для режима энергосбережения. При написании программного обеспечения необходимо принимать во внимание, что эта инструкция может вести себя как NOP.

#### Ограничения

Данная инструкция не имеет ограничений.

#### Флаги

Данная инструкция не влияет на состояние флагов.

#### Примеры

WFI – Ожидание прерывания.

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

Микросхема имеет в своем составе блок из трех каналов 24-битных  $\Delta\Sigma$  АЦП. Каждый из трех каналов оцифровывает входной сигнал с выходной частотой отсчетов 4/8/16кГц. Кроме этого по вычисленным значениям тока и напряжения 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.

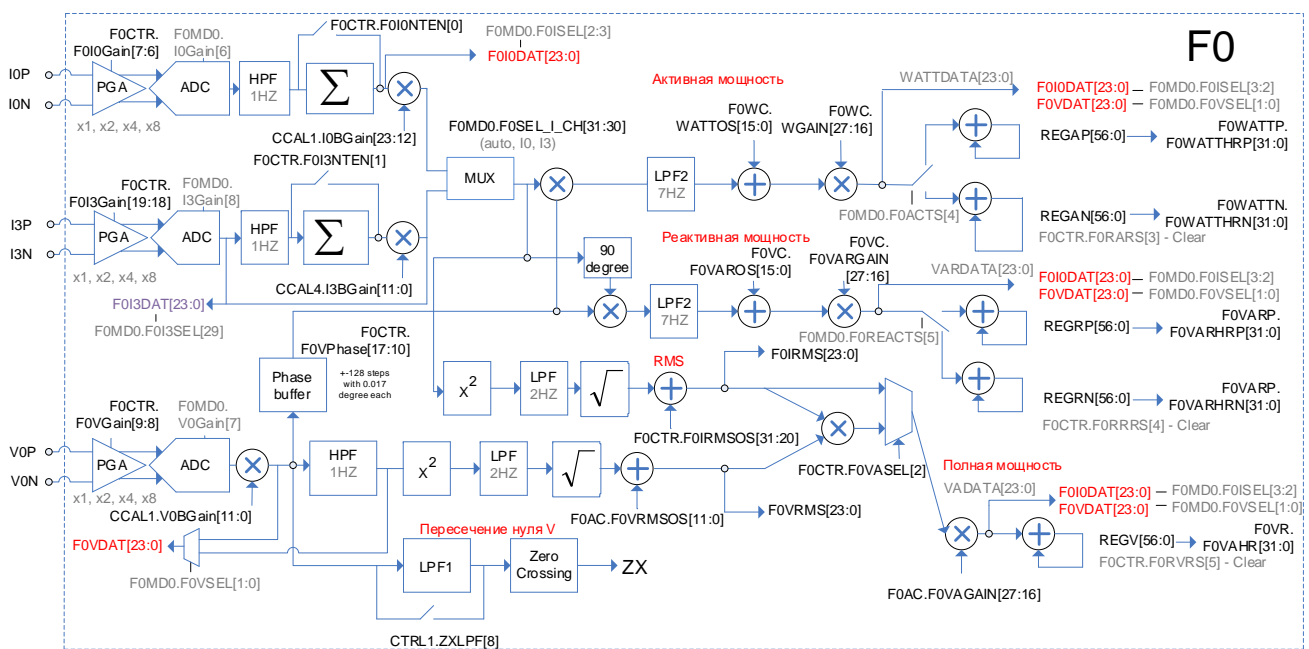


Рисунок 20 – Структурная схема трех каналов АЦП

Список вычисляемых параметров блоком АЦП:

- Три независимых АЦП с выходной частотой отсчетов 4/8/16кГц (два канала тока и один канал напряжения). Размер FIFO данных по каждому каналу АЦП составляет восемь отсчетов;
- В блоке каналов реализован автоматический выбор канала тока (который имеет максимальное значение) для последующих расчетов мощностных характеристик. Если разница токов превышает 6%, то формируется прерывание;
- Все каналы АЦП имеют независимые калибровочные коэффициенты наклона характеристики;
- Каждый канал тока имеет независимый интегратор;
- Рассчитывается период сигнала по каналу напряжения. Количество периодов, в течении которого рассчитывается эта величина, можно задавать 1/2/4/8/16/32/64/128 периодам;
- Есть проверка на пропажу периодического сигнала в канале напряжения;

- Проверяется просадка напряжения ниже заданного уровня, а также превышения сигнала в каналах тока и напряжения установленного лимита;
- Есть возможность скорректировать фазы сигналов в канале напряжения с точностью до 0,02 %;
- Вычисляются среднеквадратические, квадрат среднеквадратических значений токов и напряжений, а также их независимая калибровка;
- При вычислении активной и реактивной энергий значение накопленной энергии в течении периода накапливаются в отдельных регистрах (для положительной и отрицательной энергии);
- Вычисляется полная мощность и полная энергия.

Для предотвращения влияния высокочастотных помех на результаты вычисления необходимо поставить внешний антиэлайзинговый фильтр. Можно использовать простейший RC-фильтр первого порядка с частотой среза 10 кГц.

Все цифровые фильтры настроены на указанные частоты среза при тактировании блока ADCUI от HSE=8,192 МГц, при этом входная частота АЦП равна 2,048 МГц (выходная частота отсчетов 4/8/16 кГц).

### 13.1 Описание регистров управления блока трехканального $\Delta\Sigma$ АЦП

Таблица 44 – Перечень регистров управления блока трехканального  $\Delta\Sigma$ АЦП

Базовый адрес	Название	Описание
0x4006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x000	ADCUI_CTRL1	Общее управление для контроллера АЦП
0x004	ADCUI_CTRL2	
0x008	ADCUI_CTRL3	
0x00C	ADCUI_F0CTR	Управление блоком каналов F0
0x010	ADCUI_F0WC	Управление расчетом активной мощности в блоке F0
0x014	ADCUI_F0WATTP	Старшая часть значения положительной активной мощности в блоке F0
0x018	ADCUI_F0WATTN	Старшая часть значения отрицательной активной мощности в блоке F0
0x01C	ADCUI_F0VC	Управление расчетом реактивной мощности в блоке F0
0x020	ADCUI_F0VARP	Старшая часть значения положительной реактивной мощности в блоке F0
0x024	ADCUI_F0VARN	Старшая часть значения отрицательной реактивной мощности в блоке F0
0x028	ADCUI_F0AC	Управление расчетом полной мощности в блоке F0
0x02C	ADCUI_F0VR	Старшая часть значения полной мощности в блоке F0
0x030	ADCUI_F0MD0	Группа параметров 0 блока F0
0x034	ADCUI_F0MD1	Группа параметров 1 блока F0
0x038	ADCUI_F0VPEAK	Пиковое значение в канале напряжения блока F0
0x03C	ADCUI_F0IPEAK	Пиковое значение в канале тока блока F0

Базовый адрес	Название	Описание
0x040	ADCUI_F0VDAT	Буфер FIFO отсчетов в канале напряжения блока F0
0x044	ADCUI_F0IODAT	Буфер FIFO отчетов в канале тока (I0) блока F0
0x048	ADCUI_F0I3DAT	Буфер FIFO отсчетов в канале тока (I3) блока F0
0x04C	ADCUI_F0VRMS	Среднеквадратическое значение канала напряжения в блоке F0
0x050	ADCUI_F0VRMS2	Квадрат RMS в канале напряжения блока F0
0x054	ADCUI_F0IRMS	Среднеквадратическое значение канала тока в блоке F0
0x058	ADCUI_F0IRMS2	Квадрат RMS в канале тока блока F0
0x05C	ADCUI_F0STAT	Статус каналов блока F0
0x060	ADCUI_F0MASK	Маска прерываний каналов в блоке F0
0x064-0x110		Зарезервировано
0x114	ADCUI_CCAL1	Регистр калибровки каналов тока (I0) и напряжения (V0) в блоке F0
0x118-0x11C		Зарезервировано
0x120	ADCUI_CCAL4	Регистра калибровки канала тока (I3) в блоке F0

### 13.1.1 ADCUI\_CTRL1

Таблица 45 – Регистр ADCUI\_CTRL

Номер	31, 30	29	28	27	26...22	21	20	19	18, 17	16, 15
Доступ	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Сброс	00	0	0	0	-	0	0	0	0	0
	OSR_CONF	IBOOST	RESET_DIG	ZXRMS	-	BUF_BY	VREF_SEL	FREQ_SEL	VANO_LOAD	VARNO_LOAD

Номер	14	13, 12	11...9	8	7	6	5...2	1	0
Доступ	-	R/W	R/W	R/W	R/W	R/W	-	-	R/W
Сброс	-	0	000	0	0	0	-	0	0
	-	APNOLOAD	PER_LENGTH	ZXLPF	RESOL	I3EN	-	V0EN	I0EN

Таблица 46 – Описание битов регистра ADCUI\_CTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	OSR_CONF*	Выбор коэффициента передискретизации: 00 – 256 (4 кГц); 01 – 128 (8 кГц); 10 – 64 (16 кГц); 11 – Зарезервировано
29	IBOOST	Увеличение тока АЦП: 0 – Нормальный режим; 1 – Увеличение тока

Номер бита	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 с напряжением «0»
26...22	-	-
21	BUF_BYR	Буферизация опорного напряжения: 0 – опорное напряжение буферизировано; 1 – опорное напряжение не буферизировано
20	VREF_SEL	Выбор опорного напряжения для АЦП 0 – внутреннее опорное напряжение ( $U_{OBIAS}$ ) 1 – внешнее опорное напряжение с вывода VR_1V
19	FREQSEL	Разрешение вычисления длительности периода в каналах напряжения: 1 – разрешено; 0 – хранится последнее вычисленное значение
18, 17	VANOLOAD	Режим «без нагрузки» при вычислении полной энергии: 00 – вся вычисленная энергия накапливается; 01 – не учитывается энергия ниже 0,012 % от полной шкалы; 10 – не учитывается энергия ниже 0,0061 % от полной шкалы; 11 – не учитывается энергия ниже 0,00305 % от полной шкалы
16, 15	VARNLOAD	Режим «без нагрузки» при вычислении реактивной энергии: 00 – вся вычисленная энергия накапливается; 01 – не учитывается энергия ниже 0,012% от полной шкалы; 10 – не учитывается энергия ниже 0,0061% от полной шкалы; 11 – не учитывается энергия ниже 0,00305% от полной шкалы
14	-	Зарезервировано
13, 12	APNOLOAD	Режим «без нагрузки» при вычислении активной энергии: 00 – вся вычисленная энергия накапливается; 01 – не учитывается энергия ниже 0,012 % от полной шкалы; 10 – не учитывается энергия ниже 0,0061 % от полной шкалы; 11 – не учитывается энергия ниже 0,00305 % от полной шкалы
11...9	PER_LENGTH	Диапазон вычисления периода и фазового сдвига: 000 – в течение 1 периода; 001 – в течение 2 периодов; ... 111 – в течение 128 периодов
8	ZXLRF	Отключение низкочастотного фильтра перед детектором пересечения «0» в каналах напряжения: 0 – фильтр включен; 1 – фильтр отключен

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	RESOL	Разрешение выходных данных: 0 – 16 бит; 1 – 24 бита
6	I3EN	Разрешение работы канала I3: 0 – канал отключен; 1 – канал включен
5...2	-	-
1	V0EN	Разрешение работы канала V0: 0 – канал отключен; 1 – канал включен
0	I0EN	Разрешение работы канала I0: 0 – канал отключен; 1 – канал включен

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

\*\* Так как происходит одновременное обновление среднеквадратических значений, тока и напряжения, то значение тока будет зависеть от угла между напряжением и током. Эта зависимость представлена на рисунке 21. Исходя из этих данных, можно скорректировать действительное значение тока

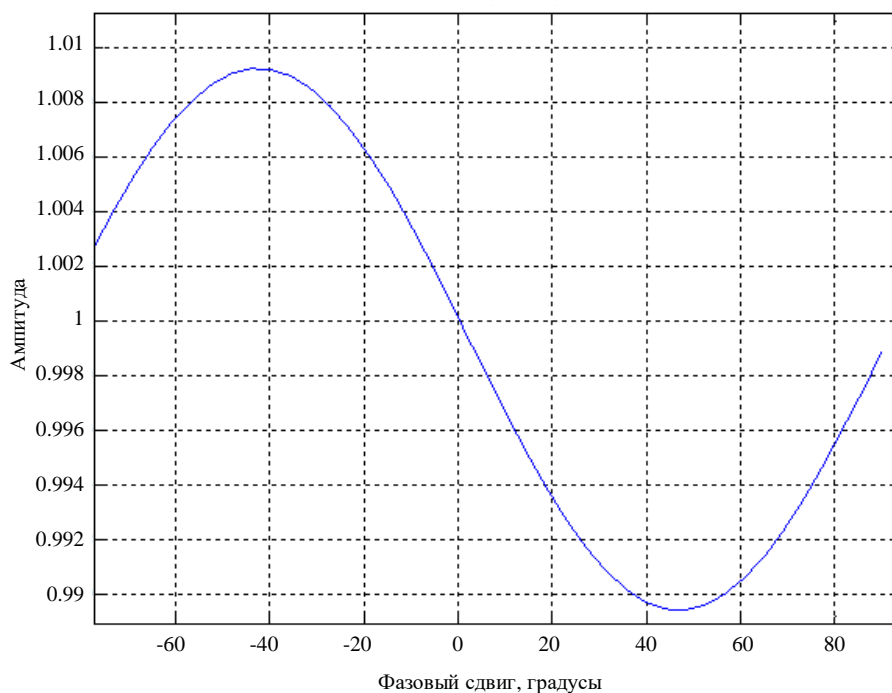


Рисунок 21 – Зависимость действующего значения тока от фазового сдвига



### 13.1.2 ADCUI\_CTRL2

Таблица 47 – Регистр ADCUI\_CTRL2

Номер	31...24	23...16	15...0
Доступ	-	R/W	R/W
Сброс	-	00h	0000h
	-	SAGCYC	SAGLVL

Таблица 48 – Описание битов регистра ADCUI\_CTRL2

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

### 13.1.3 ADCUI\_CTRL3

Таблица 49 – Регистр ADCUI\_CTRL3

Номер	31...12	11...0
Доступ	-	R/W
Сброс	-	000h
	-	ZXTOUT

Таблица 50 – Описание битов регистра ADCUI\_CTRL3

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

### 13.1.4 ADCUI\_F0CTR

Таблица 51 – 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
	FOIRMSOS	FOI3GAIN	F0VPHASE	F0VGAIN	FOIGAIN

Номер	5	4	3	2	1	0
Доступ	WO	WO	WO	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	FORVRS	FORRRS	FORARS	F0VASEL	FOI3NTEN	FOI0NTEN

Таблица 52 – Описание битов регистра ADCUI\_F0CTR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F0IRMSOS	Калибровка вычислителя среднеквадратического значения тока
19, 18	F0I3GAIN	Предусилитель в канале тока 3: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
17...10	F0VPHASE	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -126d (-123мкс) до +127d(+124мкс). “0” соответствует синфазному сигналу с током
9,8	F0VGAIN	Предусилитель в канале напряжения: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
7,6	F0I0GAIN	Предусилитель в канале тока 0: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
5	F0RVRS	Запись 1 в этот бит сбрасывает счетчик переданной полной энергии
4	F0RRRS	Запись 1 в этот бит сбрасывает счетчик переданной реактивной энергии
3	F0RARS	Запись 1 в этот бит сбрасывает счетчик переданной активной энергии
2	F0VASEL	Выбор источника сигнала для сохранения в регистре полной энергии: 0 – полная энергия; 1 – среднеквадратическое значение тока
1	F0I3NTEN	Отключение интегратора в канале тока 3: 0 – интегратор включен; 1 – интегратор отключен
0	F0I0NTEN	Отключение интегратора в канале тока 0: 0 – интегратор включен; 1 – интегратор отключен

### 13.1.5 ADCUI\_F0WC

Таблица 53 – Регистр ADCUI\_F0WC

Номер	31...28	27...16	15...0
Доступ	-	R/W	R/W
Сброс	-	7FFh	0000h
	-	F0WGAIN	F0WATTOS

Таблица 54 – Описание битов регистра ADCUI\_F0WC

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

### 13.1.6 ADCUI\_F0WATTP

Таблица 55 – Регистр ADCUI\_F0WATTP

Номер	31...0
Доступ	RO
Сброс	
	F0WATTHRP

Таблица 56 – Описание битов регистра ADCUI\_F0WATTP

Номер бита	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 57 – Регистр ADCUI\_F0WATTN

Номер	31...0
Доступ	RO
Сброс	
	F0WATTHRN

Таблица 58 – Описание битов регистра ADCUI\_F0WATTN

Номер бита	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 59 – Регистр ADCUI\_F0VC

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

Таблица 60 – Описание битов регистра ADCUI\_F0VC

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

### 13.1.9 ADCUI\_F0VARP

Таблица 61 – Регистр ADCUI\_F0VARP

Номер	31...0
Доступ	RO
Сброс	
	F0VARHRP

Таблица 62 – Описание битов регистра ADCUI\_F0VARP

Номер бита	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 63 – Регистр ADCUI\_F0VARN

Номер	31...0
Доступ	RO
Сброс	
	F0VARHRN

Таблица 64 – Описание битов регистра ADCUI\_F0VARN

Номер бита	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 65 – Регистр ADCUI\_F0AC

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

Таблица 66 – Описание битов регистра ADCUI\_F0AC

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0VAGAIN	Калибровка усиления сигнала в ветви полной мощности для блока F0
15...12	-	Зарезервировано
11...0	F0VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения

### 13.1.12 ADCUI\_F0VR

Таблица 67 – Регистр ADCUI\_F0VR

Номер	31...0
Доступ	RO
Сброс	
	F0VAHR

Таблица 68 – Описание битов регистра ADCUI\_F0VR

Номер бита	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 69 – Регистр ADCUI\_F0MD0

Номер	31...20	29	28...12	11...9	8	7	6	5	4	3, 2	1, 0
Доступ	R/W	R/W	RO	-	R/W	R/W	R/W	R	R	R/W	R/W
Сброс	00	0		-	0	0	0	0	0	00	00
	F0SEL_I_CH	F0I3SEL	F0PER_FREQ	-	I3GAIN	V0GAIN	I0GAIN	F0REACTS	F0ACTS	F0ISEL	F0VSEL

Таблица 70 – Описание битов регистра ADCUI\_F0MD0

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...30	F0SEL_I_CH	Выбор активного канала тока для вычисления мощностных характеристик: 00,11 – автоматический выбор канала; 01 – активный канал I0; 10 – активный канал I3
29	F0I3SEL	Выбор источника сигнала для FIFO канала тока I3 (регистр ADCUI_F0I3DAT): 0 – отсчеты тока после фильтра высоких частот; 1 – отсчеты тока до фильтра высоких частот

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
28...12	FOPER_FREQ	Длительность такта в канале напряжения
11...9	-	Зарезервировано
8	I3GAIN	Усиление в канале I3: 0 – нет усиления; 1 – +6дБ усиление
7	V0GAIN	Усиление в канале V0: 0 – нет усиления; 1 – +6дБ усиление
6	I0GAIN	Усиление в канале I0: 0 – нет усиления; 1 – +6дБ усиление
5	FOREACTS	Знак реактивной энергии в последний период: 0 – минус; 1 – плюс
4	F0ACTS	Знак активной энергии в последний период: 0 – минус; 1 – плюс
3, 2	FOISEL	Выбор источника сигнала для FIFO канала тока I0 (регистр ADCUI_F0I0DAT): 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1, 0	F0VSEL	Выбор источника сигнала для FIFO канала напряжения V0 (регистр ADCUI_F0VDAT): 00 – отсчеты напряжения до фильтра высоких частот; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты напряжения после фильтра высоких частот

### 13.1.14 ADCUI\_F0MD1

Таблица 71 – Регистр ADCUI\_F0MD1

Номер	31...16	15...0
Доступ	R/W	R/W
Сброс	0000h	0000h
	F0VPKLV L	F0IPKLV L

Таблица 72 – Описание битов регистра ADCUI\_F0MD1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...16	F0VPKLV L	Предельный разрешенный уровень напряжения
15...0	F0IPKLV L	Предельный разрешенный уровень тока

### 13.1.15 ADCUI\_F0VPEAK

Таблица 73 – Регистр ADCUI\_F0VPEAK

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

Таблица 74 – Описание битов в регистре ADCUI\_F0VPEAK

Номер бита	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 пиковой значение

### 13.1.16 ADCUI\_F0IPEAK

Таблица 75 – Регистр ADCUI\_F0IPEAK

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

Таблица 76 – Описание битов в регистре ADCUI\_F0IPEAK

Номер бита	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 пиковой значение

### 13.1.17 ADCUI\_F0VDAT

Таблица 77 – Регистр ADCUI\_F0VDAT

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

Таблица 78 – Описание битов в регистре ADCUI\_F0VDAT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...24	-	Зарезервировано
23...0	F0VDAT	FIFO отсчетов напряжения (или одной из мощностей)

### 13.1.18 ADCUI\_F0I0DAT

Таблица 79 – Регистр ADCUI\_F0I0DAT

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

Таблица 80 – Описание битов в регистре ADCUI\_F0I0DAT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...24	-	Зарезервировано
23...0	F0I0DAT	FIFO отсчетов тока 0 (или одной из мощностей)

### 13.1.19 ADCUI\_F0I3DAT

Таблица 81 – Регистр ADCUI\_F0I3DAT

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

Таблица 82 – Описание битов в регистре ADCUI\_F0I3DAT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...24	-	Зарезервировано
23...0	F0I3DAT	FIFO отсчетов тока 3

### 13.1.20 ADCUI\_F0VRMS

Таблица 83 – Регистр ADCUI\_F0VRMS

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

Таблица 84 – Описание битов в регистре ADCUI\_F0VRMS

Номер бита	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 85 – Регистр ADCUI\_F0VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F0VRMS2

Таблица 86 – Описание битов в регистре ADCUI\_F0VRMS2

Номер бита	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 87 – Регистр ADCUI\_F0IRMS

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

Таблица 88 – Описание битов в регистре ADCUI\_F0IRMS

Номер бита	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 89 – Регистр ADCUI\_F0IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F0IRMS2

Таблица 90 – Описание битов в регистре ADCUI\_F0IRMS2

Номер бита	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 91 – Регистр ADCUI\_F0STAT

Номер	31...27	26	25	24	23	22	21	20	19	18	17	16
Доступ	-	R/W	R/W	R/W	RO	RO	R/W	RO	-	RO	R/W	RO
Сброс	-		0	0	0	0	0	0	-	0	0	0
	-	F0VAROVN	F0WATTOVN	C3IF_OVR	C3IF_FLL	C3IF_EMP	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	F0ICCHANNEL	F0ZXTOF	F0VAOV	F0VAROVP	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	F0VFOVR	F0VF_FLL	F0VF_EMP

Таблица 92 – Описание битов в регистре ADCUI\_F0STAT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..27	-	Зарезервировано
26	F0VAROVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной реактивной энергией. Запись «1» сбрасывает этот флаг
25	F0WATTOVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной активной энергией. Запись «1» сбрасывает этот флаг
24	C3IF_OVR	Флаг, что произошло переполнения FIFO C3IDAT. Запись «1» сбрасывает этот флаг
23	C3IF_FLL	Флаг, что FIFO C3IDAT заполнено
22	C3IF_EMP	Флаг, что FIFO C3IDAT 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	Произошло автоматическое переключение активного канала тока. Запись «1» сбрасывает этот флаг

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
13	F0CHANNEL	Активный канал тока: 0 – активный канал I0; 1 – активный канал I3
12	F0ZXTOF	Флаг, что в течении TimeOut не было пересечение напряжением значения «0». Запись «1» сбрасывает этот флаг
11	F0VAOV	Флаг, что произошло переполнения регистра аккумулятора с полной энергией. Запись «1» сбрасывает этот флаг
10	F0VAROVP	Флаг, что произошло переполнения регистра аккумулятора с положительной реактивной энергией. Запись «1» сбрасывает этот флаг
9	F0WATTOVP	Флаг, что произошло переполнения 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 переполнения FIFO F0I0DAT. Запись «1» сбрасывает этот флаг
4	F0IF_FLL	Флаг, что FIFO F0I0DAT заполнено
3	F0IF_EMP	Флаг, что FIFO F0I0DAT пусто
2	F0VF_OVR	Флаг, что произошло переполнения FIFO F0V0DAT. Запись «1» сбрасывает этот флаг
1	F0VF_FLL	Флаг, что FIFO F0V0DAT заполнено
0	F0VF_EMP	Флаг, что FIFO F0V0DAT пусто

### 13.1.25 ADCUI\_F0MASK

Таблица 93 – Регистр ADCUI\_F0MASK

Номер	31...27	26	25	24	23	22	21
Доступ	-	R/W	R/W	R/W	RO	RO	R/W
Сброс	-		0	0	0	0	0
	-	F0VA ROVNM	F0WA TTOVNM	C3IF_ OVRM	C3IF_ FLLM	C3IF_ EMPM	F0ZERO CRSM

Номер	20	19	18	17	16	15	14
Доступ	RO	-	RO	R/W	RO	R/W	R/W
Сброс	0	-	0	0	0	0	0
	F0VANL DFLM	-	F0VARNL DFLM	F0VARSI GNM	F0APNL DFLM	F0APSI GNM	F0FAUL TCONM

Номер	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
	-	F0ZX TOFM	F0VA OVM	F0VA ROVPM	F0WA TTOVPM	FOPEAK IFM	FOPEAK 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
	F0SA GFM	F0IF_ OVRM	F0IF_ FLLM	F0IF_ EMPM	F0VF_ OVRM	F0VF_ FLLM	F0VF_ EMPM

Таблица 94 – Описание битов в регистре ADCUI\_F0MASK

Номер бита	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 F0VANLDLFL
19	-	Зарезервировано
18	F0VARNLDFLM	Маска бита F0VARNLDFL
17	F0VARSIGNM	Маска бита F0VARSIGN
16	F0APNLDFLM	Маска бита F0APNLDFL
15	F0APSIGNM	Маска бита F0APSIGNM
14	F0FAULTCONM	Маска бита F0FAULTCON
13	-	Зарезервировано
12	F0ZXTOFM	Маска бита F0ZXTOF
11	F0VAOVM	Маска бита F0VAOV
10	F0VAROVPM	Маска бита F0VAROVP
9	F0WATTOVPM	Маска бита F0WATTOVP
8	FOPEAKIFM	Маска бита FOPEAKIF
7	FOPEAKVFM	Маска бита FOPEAKVF
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\_CCAL1

Таблица 95 – Описание битов регистра ADCUI\_CCAL1

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

### 13.1.27 ADCUI\_CCAL4

Таблица 96 – Описание битов регистра ADCUI\_CCAL4

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

## 13.2 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 22, 23 приведены два вида включения АЦП: полностью дифференциальное и недифференциальное включение.

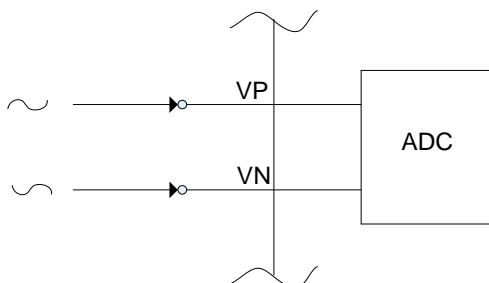


Рисунок 22 – Дифференциальное включение

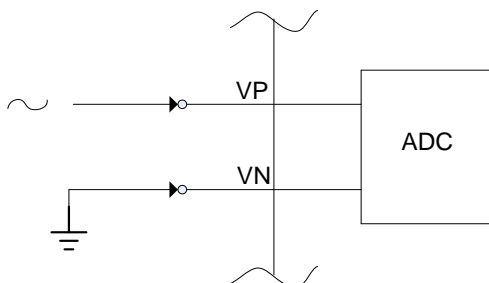


Рисунок 23 – Недифференциальное включение

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 24.

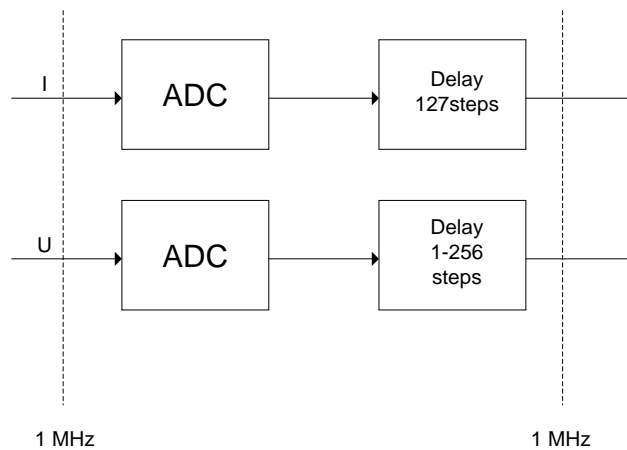


Рисунок 24 – Контролируемый фазовый сдвиг в канале напряжения

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

В качестве децимирующего фильтра используется фильтр со структурой  $\sin^3$ , его характеристика приведена на рисунках 25, 26:

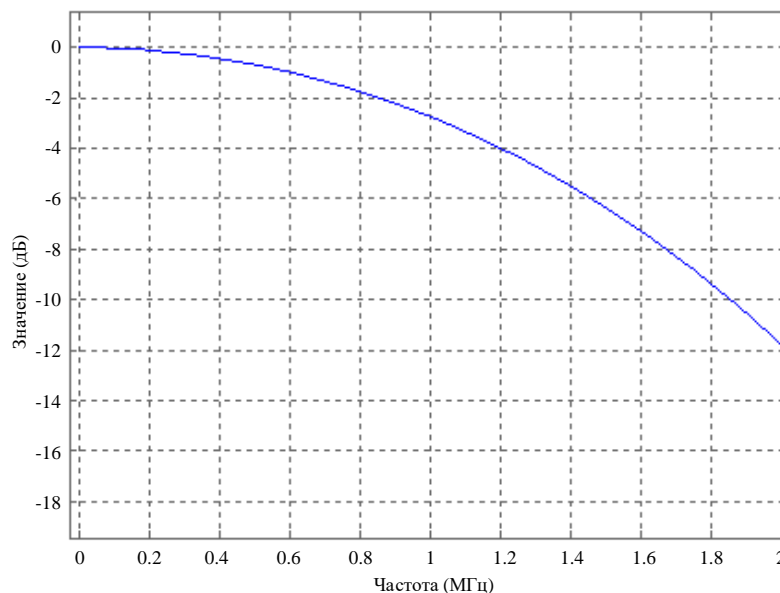


Рисунок 25 – Характеристика децимирующего фильтра в полосе 2 кГц

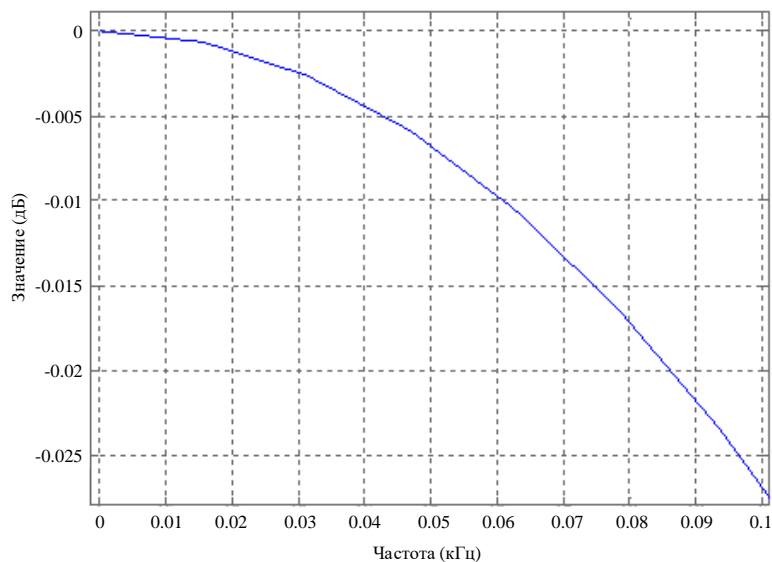


Рисунок 26 – Характеристика децимирующего фильтра в полосе 100 Гц

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

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

Каждый канал АЦП имеет буфер FIFO размером восемь отсчетов, предназначенный для хранения отсчетов тока или напряжения, а также отсчетов активной мощности, реактивной мощности и полной мощности (только для каналов I0, I3). Выбор источника отсчетов для буфера FIFO осуществляется в регистре ADCUI\_F0MD0: поле F0VSEL для канала напряжения V0, поле F0ISEL для канала тока I0, поле F0I3SEL для канала тока I3. Запись выбранных отсчетов в буфер FIFO выполняется с частотой передискретизации, заданной в поле OSR\_CONF регистра ADCUI\_CTRL1.

Если FIFO каналов сконфигурированы на прием отсчетов тока и напряжения (F0VSEL=00, F0ISEL=00, F0I3SEL=x), то значения отсчетов FIFO можно перевести в напряжения на входе по следующим формулам (таблица 97). Значение приведены для усиления PGA, равному 0 дБ. Отсчеты, записанные в FIFO представлены в двоичном формате с дополнением до 2.

Таблица 97 – Перевод значений отсчетов FIFO в напряжения на входе

Режим	Входное напряжение АЦП, Вольт	
Дифференциальное включение, 24 бит. режим	$\frac{F0V\text{DAT}}{2^{23}}$ ,	$\frac{F0I\text{x}DAT}{2^{23}}$
Дифференциальное включение, 16 бит. режим	$\frac{F0V\text{DAT}}{2^{15}}$ ,	$\frac{F0I\text{x}DAT}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{F0V\text{DAT}}{2^{23}}$ ,	$\frac{F0I\text{x}DAT}{2^{23}}$
Недифференциальное включение, 16 бит. режим	$\frac{F0V\text{DAT}}{2^{15}}$ ,	$\frac{F0I\text{x}DAT}{2^{15}}$

Каждый из каналов тока может быть скорректирован с помощью коэффициентов F0IxGAIN, IxGAIN и IxBGAIN в соответствии с формулой (1). Значение IxBGAIN записывается в двоичном формате с дополнением до 2.

$$I_{cor} = I_{ADC} \cdot 2^{F0IxGAIN} \cdot 2^{IxGAIN} \cdot \left(1 + \frac{IxBGAIN}{2^{11}}\right). \quad (1)$$

Если для 3 канала тока источник сигнала выбран до фильтра высоких частот (F0I3SEL = 1), то в значении, хранящемся в регистре ADCUI\_F0I3DAT, не учитывается коэффициент IxBGAIN.

Канал напряжения может быть скорректирован с помощью коэффициентов F0VGAIN, V0GAIN и V0BGAIN в соответствии с формулой (2). Значение V0BGAIN записывается в двоичном формате с дополнением до 2.

$$V_{cor} = V_{ADC} \cdot 2^{F0VGAIN} \cdot 2^{V0GAIN} \cdot \left(1 + \frac{V0BGAIN}{2^{11}}\right) \quad (2)$$

В регистрах F0VRMS и F0IRMS (см. таблицу 44) хранится вычисленная величина среднеквадратического значения тока и напряжения в соответствующей фазе. Значения среднеквадратических величин приведены в таблице 98.

Таблица 98 – Расчет среднеквадратичных значений напряжения

Режим	Напряжение, Вольт	
Дифференциальное включение	$\frac{F0VRMS}{2^{23}}$ ,	$\frac{F0IRMS}{2^{23}}$
Недифференциальное включение	$\frac{F0VRMS}{2^{22}}$ ,	$\frac{F0IRMS}{2^{22}}$

Для вычисления среднеквадратического значения используется следующий алгоритм (для примера выбран канал напряжения, но 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 4кГц. 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 также имеет 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 F0VRMSOS и F0IRMSOS. Перед корректировкой значение сдвигается на 8 бит вправо, что дает шаг корректировки в 256 меньше. Эта корректировка нужна для того, чтобы избавиться от ошибки, вызванной шумами на входе АЦП, которые после возведения в квадрат и накопления будут давать отклонения среднего уровня величины  $V^2(t)$ .

Формула коррекции приведена ниже:

$$V_{cor}(t) = V_{rms}(t) + \frac{F0VRMSOS}{2^{20}}. \tag{5}$$

Значения F0VRMSOS и F0IRMSOS представлены в виде знаковых величин в двоичном коде с дополнением до 2.

В регистрах F0VRMS2 и F0IRMS2 хранятся значения среднеквадратического значения напряжения и тока до извлечения квадратного корня. В таблице 99 приведены значения квадратов среднеквадратических величин.

Таблица 99 – Значения квадратов среднеквадратических величин

Режим	Напряжение, Вольт <sup>2</sup>	
Дифференциальное включение	$\frac{F0VRMS2}{2^{30}}$ ,	$\frac{F0IRMS2}{2^{30}}$ ,
Недифференциальное включение	$\frac{F0VRMS2}{2^{28}}$ ,	$\frac{F0IRMS2}{2^{28}}$ ,

Для вычисления реактивной мощности необходимо сдвинуть сигнал в канале тока на 90 градусов. Это осуществляется с помощью фильтров, которые в достаточно широком диапазоне сохраняют сдвиг равный 90 градусам для обоих каналов. На рисунке 27 приведена его фазовая характеристика.

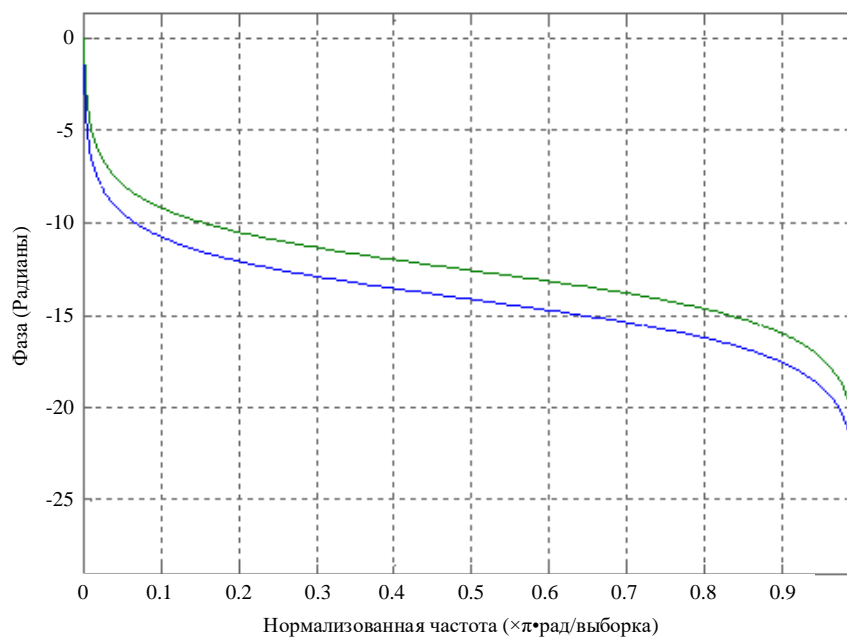


Рисунок 27 – Фильтр для сдвига сигнала на 90 градусов

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

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

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

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:

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

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 FOVDAT и FOIODAT.

Если FIFO каналов сконфигурированы на прием отсчетов мощностей (FOISEL=01/10/11, FOVSEL=01/10/11), то значения отсчетов FIFO можно перевести в значения мощностей по следующим простейшим формулам (см. таблицу 100). Значение приведены для усиления равному 0 дБ. Отсчеты, записанные в FIFO представлены в двоичном формате с дополнением до 2.

Таблица 100 – Перевод значений отсчетов FIFO в значения мощностей

Режим	Мощность, Вольт • Ампер	
Дифференциальное включение, 24 бит. режим	$\frac{FOVDAT}{2^{23}}$ ,	$\frac{FOIODAT}{2^{23}}$
Дифференциальное включение, 16 бит. режим	$\frac{FOVDAT}{2^{15}}$ ,	$\frac{FOIODAT}{2^{15}}$
Недифференциальное включение, 24 бит. режим	$\frac{FOVDAT}{2^{21}}$ ,	$\frac{FOIODAT}{2^{21}}$
Недифференциальное включение, 16 бит. режим	$\frac{FOVDAT}{2^{13}}$ ,	$\frac{FOIODAT}{2^{13}}$

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. Корректировка осуществляется в соответствии со следующей 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 101 приведена формула перевода значения в Ватт•с.

Таблица 101 – Формула перевода значения в Ватт • с

Режим	Энергия, Ватт • с	
Дифференциальное включение	$\frac{FOWATTHRP}{512 \cdot 4000}$	$\frac{FOWATTHRN}{512 \cdot 4000}$
Недифференциальное включение	$\frac{FOWATTHRP}{512 \cdot 1000}$	$\frac{FOWATTHRN}{512 \cdot 1000}$

Примечание – Значение 4000 в формуле для дифференциального включения соответствует частоте дискретизации блока  $\Delta\Sigma$ АЦП, которая задается полем OSR\_CONF регистра ADCUI\_CTRL1. В недифференциальном включении в знаменателе приводится значение 1000, то есть мощностные характеристики отличаются в четыре раза.

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

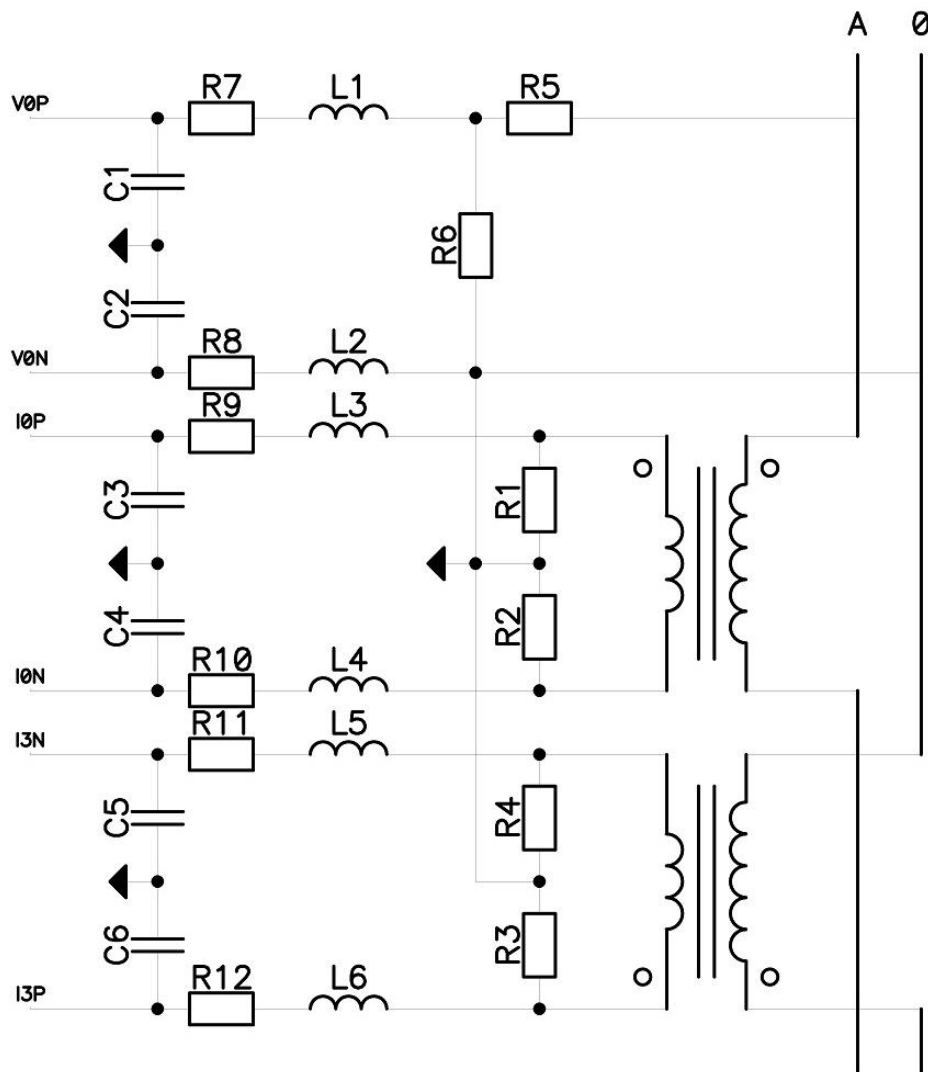


Рисунок 28 – Типовая схема включения для учета электроэнергии по одной фазе

В данной схеме предусмотрено измерение тока в обоих проводах однофазной сети. Микросхема позволяет в автоматическом режиме учитывать тот ток из каналов I0 и I3, который будет больше. Если учета тока по «нулю» не требуется, то часть схемы, относящаяся к каналу I3 можно убрать. Выход трансформаторов нагружен на резисторный делитель. Сумма сопротивлений этих резисторов, например, R1+R2 должна соответствовать требуемой нагрузке выхода трансформатора. Оба резистора имеют одинаковое сопротивление и создают среднюю точку, относительно которой измеряется ток. В данной схеме включения средней точкой является аналоговая земля. С этой средней точкой соединен нулевой провод. 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 антиалиасинговый фильтр, 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 антиалиасинговому фильтру и выбираются для наиболее широкополосного подавления радиочастот. Вы можете использовать свои варианты фильтров в зависимости от требований к конечному изделию.

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

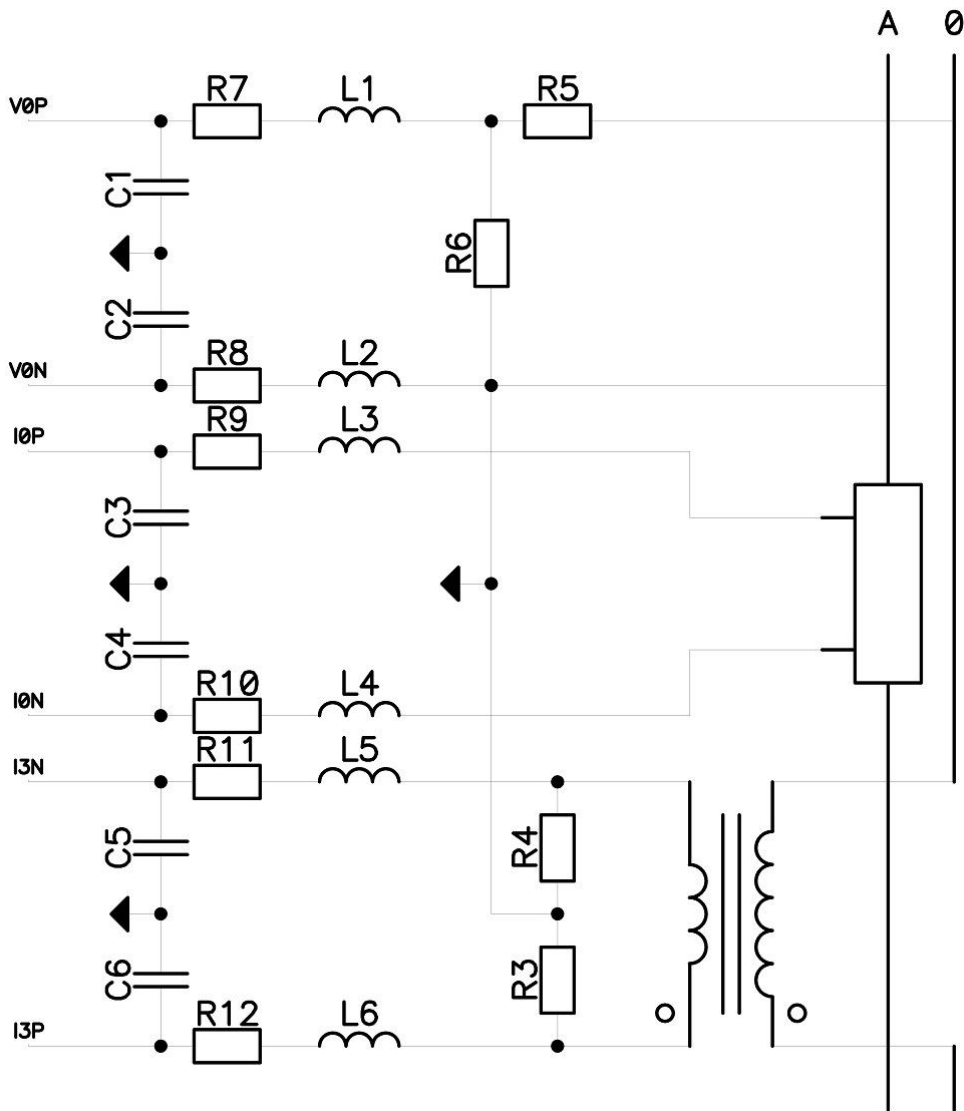


Рисунок 29 – Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта

Также для измерения тока может быть использован шунт. В этом случае следует использовать схему на рисунке 29. Аналогично предыдущей схеме на трансформаторах, здесь можно исключить часть схемы, относящуюся к каналу тока I3, если контроля тока в нулевом проводе не требуется.

## 14 Сигналы тактовой частоты

Микросхема имеет два встроенных генератора и два внешних осциллятора. А также специализированный блок формирования тактовой синхронизации PLL микросхемы.

Для переключения с одного источника частоты на другой, нужно, чтобы оба источника оставались включенными.

### 14.1 Структурная схема

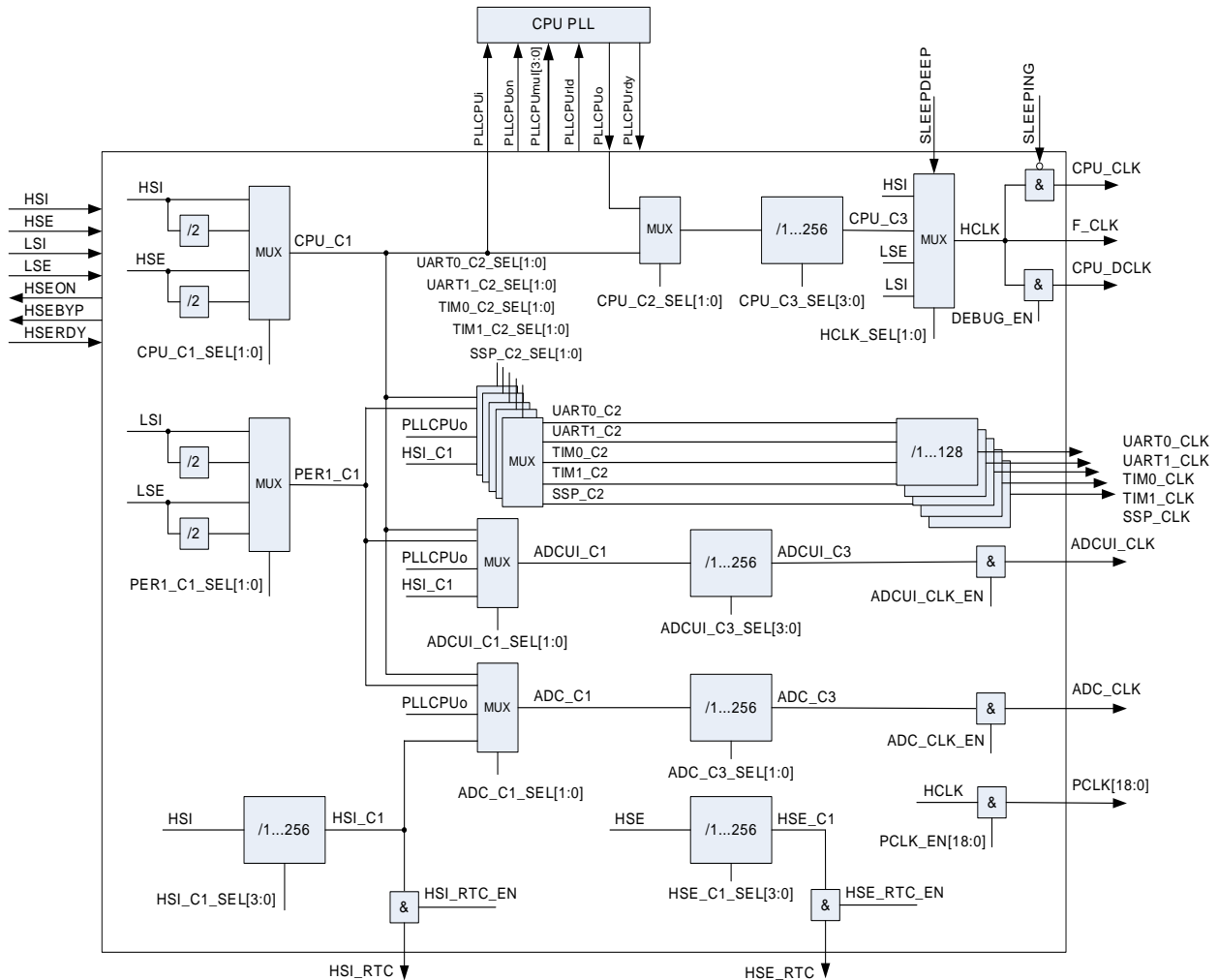


Рисунок 30 – Структурная блок-схема формирования тактовой частоты

### 14.2 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\_REG\_0F (см. таблицу 125). Первоначально процессорное ядро запускается на тактовой частоте HSI. При дальнейшей работе генератор HSI может быть отключен при помощи сигнала HSION в регистре BKP\_REG\_0F. Также генератор может быть подстроен при помощи сигнала HSITRIM в регистре BKP\_REG\_0F.

### 14.3 Встроенный RC-генератор LSI

Генератор LSI вырабатывает тактовую частоту 40 КГц. Генератор автоматически запускается при появлении питания  $U_{cc}$  и при выходе в нормальный режим работы вырабатывает сигнал LSIRDY в регистре BKP\_REG\_0F. Первоначально тактовая частота генератор LSI используется для формирования дополнительной задержки схемы POR. При дальнейшей работе генератор LSI может быть отключен при помощи сигнала LSION в регистре BKP\_REG\_0F.

### 14.4 Внешний осциллятор HSE

Осциллятор HSE предназначен для выработки тактовой частоты 2...16 МГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания  $U_{cc}$  и сигнала разрешения HSEON в регистре HS\_CONTROL (см. таблицу 102). При выходе в нормальный режим работы вырабатывает сигнал HSERDY в регистре CLOCK\_STATUS (см. таблицу 102). 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 третьем состоянии.

### 14.5 Внешний осциллятор LSE

Осциллятор LSE предназначен для выработки тактовой частоты 32 КГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания  $BDU_{cc}$  (рисунок 31) и сигнала разрешения LSEON в регистре BKP\_REG\_0F. При выходе в нормальный режим работы вырабатывает сигнал LSERDY в регистре BKP\_REG\_0F. Также осциллятор может работать в режиме LSEBYP, когда входная тактовая частота с входа OSC\_IN32 проходит напрямую на выход LSE. Выход OSC\_OUT32 находится в этом режиме третьем состоянии. Так как генератор LSE питается от напряжения питания  $BDU_{cc}$  и его регистр управления BKP\_REG\_0F расположен в батарейном домене, то генератор может продолжать работать при пропадании основного питания  $U_{cc}$ . Генератор LSE используется для работы часов реального времени.

### 14.6 Встроенный блок умножения системной тактовой частоты

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 16, задаваемый на входе PLLCPUMUL[3:0] в регистре PLL\_CONTROL (таблица 102). Входная частота блока умножителя должна быть в диапазоне 2...16 МГц, выходная частота в диапазоне 2...36 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 микросхема запускается на частоте HSI-генератора. Выдача тактовых сигналов синхронизации для всех периферийных блоков кроме RST\_CLK отключена. Для начала работы с нужным периферийным блоком необходимо включить его тактовую частоту в регистре PER\_CLOCK. Некоторые контроллеры интерфейсов

(UART, Таймеры) могут работать на частотах отличных от частоты процессорного ядра, поэтому в соответствующих регистрах (UART\_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.

### 14.7 Описание регистров блока контроллера тактовой частоты

Таблица 102 – Перечень регистров блока контроллера тактовой частоты

Базовый адрес	Название	Описание
0x4002_0000	RST_CLK	Контроллер тактовой частоты
Смещение		
0x00	CLOCK_STATUS	Регистр состояния блока управления тактовой частотой
0x04	PLL_CONTROL	Регистр управления блоками умножения частоты
0x08	HS_CONTROL	Регистр управления высокочастотным генератором и осциллятором
0x0C	CPU_CLOCK	Регистр управления тактовой частотой процессорного ядра
0x10	PER1_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x14	ADC_CLOCK	Регистр управления тактовой частотой АЦП и $\Delta\Sigma$ АЦП
0x18	RTC_CLOCK	Регистр управления формированием высокочастотных тактовых сигналов блока RTC
0x1C	PER2_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x24	TIM_CLOCK	Регистр управления тактовой частотой TIMER
0x28	UART_CLOCK	Регистр управления тактовой частотой UART
0x2C	SSP_CLOCK	Регистр управления тактовой частотой SSP

#### 14.7.1 CLOCK\_STATUS

Таблица 103 – Регистр CLOCK\_STATUS

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

Таблица 104 – Описание битов регистра CLOCK\_STATUS

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



Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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	-	Зарезервирован

### 14.7.2 PLL\_CONTROL

Таблица 105 – Регистр PLL\_CONTROL

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

Таблица 106 – Описание битов регистра PLL\_CONTROL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..12	-	Зарезервировано
11...8	PLL CPU MUL[3:0]	Коэффициент умножения для CPU PLL: $PLL_{CPU0} = PLL_{CPUi} \times (PLL_{CPUMUL} + 1)$
7..4	-	Зарезервировано
3	PLL CPU RLD	Бит перезапуска PLL. При смене коэффициента умножения в рабочем режиме необходимо задать равным 1, а после этого сбросить в 0
2	PLL CPU ON	Бит включения PLL: 0 – PLL выключена; 1 – PLL включена
1	-	Зарезервировано
0	-	Зарезервировано

### 14.7.3 HS\_CONTROL

Таблица 107 – Регистр HS\_CONTROL

Номер	31...2	1	0
Доступ	U	R/W	R/W
Сброс	0	0	0
	-	HSE BYP	HSE ON

Таблица 108 – Описание битов регистра HS\_CONTROL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..2	-	Зарезервировано
1	HSE BYP	Бит управления HSE осциллятором: 0 – режим осциллятора; 1 – режим внешнего генератора. Устанавливать совместно с HSE_ON=1
0	HSE ON	Бит управления HSE осциллятором: 0 – выключен; 1 – включен

#### 14.7.4 CPU\_CLOCK

Таблица 109 – Регистр CPU\_CLOCK

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

Таблица 110 – Описание битов регистра CPU\_CLOCK

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	-	Зарезервировано
9, 8	HCLK SEL[1:0]	Биты выбора источника для HCLK: 00 – HSI; 01 – CPU_C3; 10 – LSE; 11 – LSI
7...4	CPU C3 SEL[3:0]	Биты выбора делителя для CPU_C3: 0xxx – CPU_C3 = CPU_C2; 1000 - CPU_C3 = CPU_C2 / 2; 1001 - CPU_C3 = CPU_C2 / 4; 1010 - CPU_C3 = CPU_C2 / 8; ... 1111 - CPU_C3 = CPU_C2 / 256
3	-	Зарезервировано
2	CPU C2 SEL	Биты выбора источника для CPU_C2: 0 – CPU_C1; 1 – PLLCPUo
1, 0	CPU C1 SEL[1:0]	Биты выбора источника для CPU_C1: 00 – HSI; 01 – HSI/2; 10 – HSE; 11 – HSE/2

### 14.7.5 PER1\_CLOCK

Таблица 111 – Регистр PER1\_CLOCK

Номер	31...14	13...12	11, 10	9, 8	7, 6	5	4	3, 2	1, 0
Доступ	U	R/W	R/W	R/W	R/W	U	R/W	R/W	R/W
Сброс	0	00	00	00	00	0	1	00	00
	-	SSP C2 SEL	UART1 C2 SEL	UART0 C2 SEL	TIM1 C2 SEL	-	DEBUG_EN	TIM0 C2 SEL	PER C1 SEL[1:0]

Таблица 112 – Описание битов регистра PER1\_CLOCK

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14	-	Зарезервировано
13, 12	SSP C2 SEL	Биты выбора источника для SSP_C2: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_CLK
11, 10	UART1 C2 SEL	Биты выбора источника для UART1_C2: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_CLK
9, 8	UART0 C2 SEL	Биты выбора источника для UART0_C2: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_CLK
7, 6	TIM1 C2 SEL	Биты выбора источника для TIM1_C2: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_CLK
5	-	Зарезервировано
4	DEBUG_EN	Бит разрешения тактирования блока отладки ядра
3, 2	TIM0 C2 SEL	Биты выбора источника для TIM0_C2: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_CLK
1, 0	PER1 C1 SEL[1:0]	Биты выбора источника для PER1_C1: 00 – LSI; 01 – LSI/2; 10 – LSE; 11 – LSE/2

14.7.6 ADC\_CLOCK

Таблица 113 – 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 CLK EN	ADC C3 SEL[3:0]	ADCUI C3 SEL[3:0]	ADCUI C1 SEL[1:0]	ADC C1 SEL[1:0]

Таблица 114 – Описание битов регистра ADC\_CLOCK

Номер бита	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
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 – PLLCPU <sub>0</sub> ; 11 – HSI_CLK
1, 0	ADC C1 SEL[1:0]	Биты выбора источника для ADC_C1: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPU <sub>0</sub> ; 11 – HSI_CLK

### 14.7.7 RTC\_CLOCK

Таблица 115 – Регистр RTC\_CLOCK

Номер	31...10	9	8	7...4	3...0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0000	0000
	-	HSI_RTC_EN	HSE_RTC_EN	HSI_C1_SEL[3:0]	HSE_C1_SEL[3:0]

Таблица 116 – Описание битов регистра RTC\_CLOCK

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..10	-	Зарезервировано
9	HSI_RTC_EN	Бит разрешения выдачи тактовой частоты HSI_RTC: 0 – запрещен; 1 – разрешен
8	HSE_RTC_EN	Бит разрешения выдачи тактовой частоты HSE_RTC: 0 – запрещен; 1 – разрешен
7...4	HSI_C1_SEL[3:0]	Биты выбора делителя для HSI_C1: 0xxx – HSI_C1 = HSI; 1000 – HSI_C1 = HSI / 2; 1001 – HSI_C1 = HSI / 4; 1010 – HSI_C1 = HSI / 8; ... 1111 – HSI_C1 = HSI / 256
3...0	HSE_C1_SEL[3:0]	Биты выбора делителя для HSE_C1: 0xxx – HSE_C1 = HSE; 1000 – HSE_C1 = HSE / 2; 1001 – HSE_C1 = HSE / 4; 1010 – HSE_C1 = HSE / 8; ... 1111 – HSE_C1 = HSE / 256

### 14.7.8 PER2\_CLOCK

Таблица 117 – Регистр PER2\_CLOCK

Номер	18..0
Доступ	R/W
Сброс	0
	PCLK_EN[18:0]

Таблица 118 – Описание битов регистра PER2\_CLOCK

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
18...0	PCLK EN[18:0]	Биты разрешения тактирования периферийных блоков: 0 – запрещено; 1 – разрешено. PCLK[0] – SPI; PCLK[1] – UART0; PCLK[2] – UART1; PCLK[3] – EEPROM; PCLK[4] – RST_CLK; PCLK[6] – I2C; PCLK[7] – LCD; PCLK[8] – ADC; PCLK[9] – WWDG; PCLK[10] – IWDG; PCLK[11] – POWER; PCLK[12] – BKP; PCLK[13] – ADCUI; PCLK[14] – TIMER0; PCLK[15] – TIMER1; PCLK[16] – PORTA; PCLK[17] – PORTB; PCLK[18] – PORTC

### 14.7.9 TIM\_CLOCK

Таблица 119 – Регистр TIM\_CLOCK

Номер	31...26	25	24	23...16	15...0	7...0
Доступ	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	00000000	00000000
	-	TIM1 CLK EN	TIM0 CLK EN	-	TIM1 BRG [7:0]	TIM0 BRG [7:0]

Таблица 120 – Описание битов регистра TIM\_CLOCK

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...26	-	Зарезервировано
25	TIM1 CLK EN	Разрешение тактовой частоты на TIM1: 0 – нет частоты; 1 – есть частота
24	TIM0 CLK EN	Разрешение тактовой частоты на TIM0: 0 – нет частоты; 1 – есть частота

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
23...16	-	Зарезервировано
15...8	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
7...0	TIM0 BRG [7:0]	Делитель тактовой частоты TIM0: xxxxx000 – TIM0_CLK == PER1_C2; xxxxx001 – TIM0_CLK == PER1_C2/2; xxxxx010 – TIM0_CLK == PER1_C2/4; ... xxxxx111 – TIM0_CLK == PER1_C2/128
Примечание – Частота тактирования таймера TIMx_CLK должна быть меньше или равна частоте тактирования ядра CPU_CLK		

#### 14.7.10 UART\_CLOCK

Таблица 121 – Регистр UART\_CLOCK

Номер	31...26	25	24	23...16	15...0	7...0
Доступ	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	00000000	00000000
	-	UART1 CLK EN	UART0 CLK EN	-	UART1 BRG [7:0]	UART0 BRG [7:0]

Таблица 122 – Описание битов регистра UART\_CLOCK

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	-	Зарезервировано
25	UART1 CLK EN	Разрешение тактовой частоты на UART1: 0 – нет частоты; 1 – есть частота
24	UART0 CLK EN	Разрешение тактовой частоты на UART0: 0 – нет частоты; 1 – есть частота
23...16	-	Зарезервировано

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...8	UART1 BRG [7:0]	Делитель тактовой частоты UART1: xxxxx000 – UART1_CLK == PER1_C2; xxxxx001 – UART1_CLK == PER1_C2/2; xxxxx010 – UART1_CLK == PER1_C2/4; ... xxxxx111 – UART1_CLK == PER1_C2/128
7...0	UART0 BRG [7:0]	Делитель тактовой частоты UART0: xxxxx000 – UART0_CLK == PER1_C2; xxxxx001 – UART0_CLK == PER1_C2/2; xxxxx010 – UART0_CLK == PER1_C2/4; ... xxxxx111 – UART0_CLK == PER1_C2/128

### 14.7.11 SSP\_CLOCK

Таблица 123 – Регистр SSP\_CLOCK

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

Таблица 124 – Описание битов регистра SSP\_CLOCK

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



## 15 Батарейный домен и часы реального времени

Блок батарейного домена предназначен для обеспечения функций часов реального времени и сохранения некоторого набора пользовательских данных при отключении основного источника питания. Также в батарейном домене реализована функция контроля выхода COV\_DET (см. таблицу 1). Это позволяет даже в отсутствии основного питания определять его состояние. При снижении питания  $U_{CC}$  в блоке SW происходит автоматическое переключение питания  $BDU_{CC}$  с  $U_{CC}$  на  $BU_{CC}$ . Если на  $BU_{CC}$  имеется отдельный источник питания (батарейка), то батарейный домен остается включенным и может выполнять свои функции.

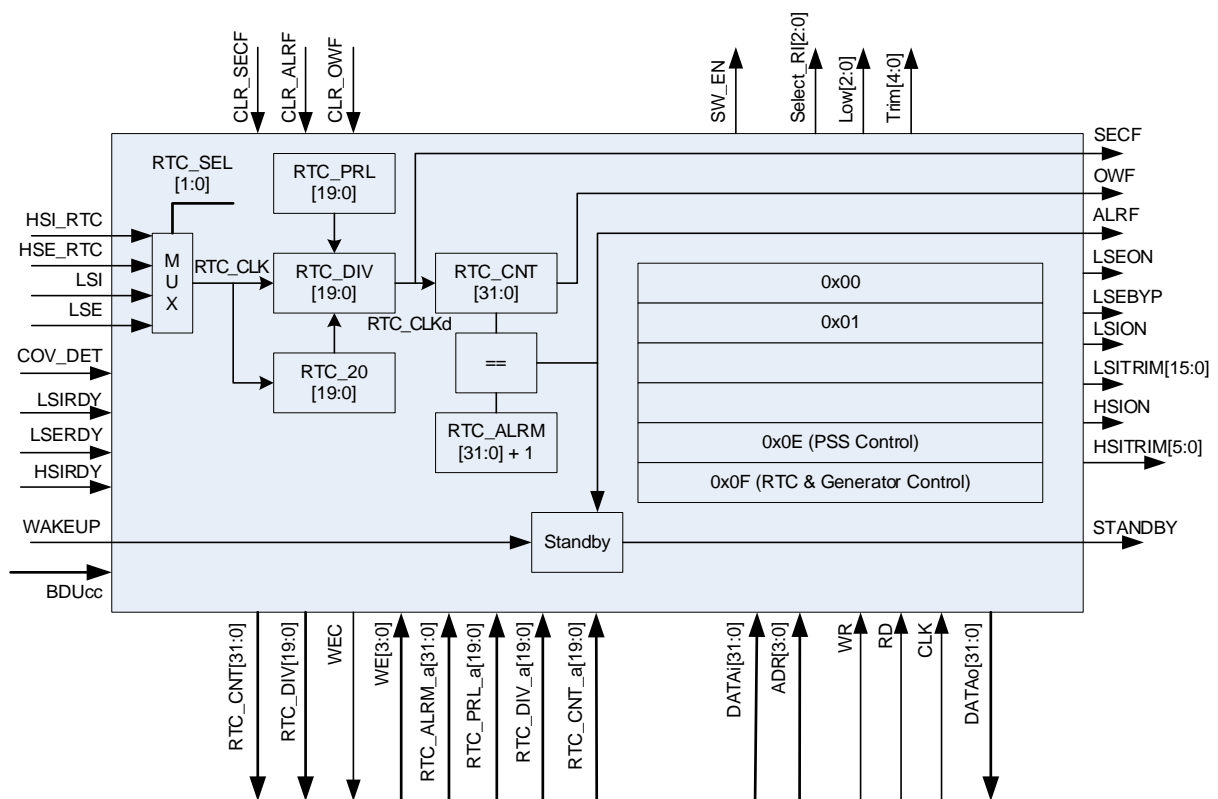


Рисунок 31 – Структурная блок-схема батарейного домена и часов реального времени

### 15.1 Часы реального времени

Часы реального времени позволяют организовать механизм отсчета времени в кристалле, в том числе при отключении основного источника питания. Включение часов реального времени осуществляется битом  $RTC\_EN$  (таблица 131). В качестве источника тактовой частоты часов реального времени  $RTC\_CLK$  может выступать генератор LSI или LSE, а также генератор HSI (частота  $HSI\_RTC$ ) или HSE (частота  $HSE\_RTC$ ) с дополнительным делителем до 256. Частоты  $HSI\_RTC$  и  $HSE\_RTC$  формируются в блоке управления тактовыми частотами  $RST\_CLK$  и могут быть выбраны только при наличии питания  $DU_{CC}$ . Частота LSI может быть выбрана при наличии питания  $U_{CC}$ , LSE – при наличии  $U_{CC}$  или  $BU_{CC}$ . Выбор между источниками осуществляется битами  $RTC\_SEL$ . При возможном отключении основного источника питания  $U_{CC}$  в качестве источника тактовой частоты  $RTC\_CLK$  должен использоваться осциллятор LSE, так как он также имеет питание  $BDU_{CC}$ . Биты управления осциллятором LSE расположены в батарейном

домене и таким образом при отключении основного питания они не сбрасываются. При этом при первоначальном включении эти биты также не определены и могут принять любое значение.

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

Для калибровки тактовой частоты RTC\_CLK используются биты RTC\_CAL[7:0]. Значение RTC\_CAL[7:0] определяет, какое число тактов RTC\_CLK из  $2^{20}$  будет замаскировано с помощью дополнительного счетчика RTC\_20. Таким образом, с помощью бит RTC\_CAL[7:0] производится замедление хода часов. Для убыстрения хода часов необходимо задать меньшее, чем требуется, значение регистра RTC\_PRL, а затем произвести замедление с помощью бит RTC\_CAL[7:0]. Изменение значения бит RTC\_CAL[7:0] может быть осуществлено в ходе работы часов реального времени.

Регистр RTC\_CNT предназначен для отсчета времени в секундах. И работает на выходной частоте предварительного делителя RTC\_CLKd. Регистр RTC\_ALRM предназначен для задания времени, при превышении которого вырабатывается флаг прерывания и пробуждения процессора ALRF. Таким образом, бит STANBY, отключающий внутренний регулятор напряжения, автоматически сбрасывается при превышении RTC\_CNT значения RTC\_ALRM.

Бит STANDBY (таблица 131) также может быть сброшен с помощью вывода WAKEUP.

В батарейном домене реализована возможность мониторинга входного сигнала на COV\_DET. Во внутреннем регистре записывается контролируемый уровень («0» или «1»), и если сигнал на входе станет отличным от записанного, то это событие регистрируется в управляющем бите.

## 15.2 Регистры аварийного сохранения

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

## 15.3 Описание регистров блока батарейного домена

Таблица 125 – Описание регистров блока батарейного домена

Базовый адрес	Название	Описание
0x4006_0000	ВКР	Контроллер батарейного домена и часов реального времени
Смещение		
0x00	ВКР_REG_00	Регистр аварийного сохранения 0
...		

Базовый адрес	Название	Описание
0x38	BKP_REG_0E	Регистр аварийного сохранения 14
0x3C	BKP_REG_0F	Регистр аварийного сохранения 15 и управления блоками RTC, LSE, LSI и HSI
0x40	RTC_CNT	Регистр основного счетчика часов реального времени
0x44	RTC_DIV	Регистр счетчика предварительного делителя
0x48	RTC_PRL	Регистр коэффициента деления предварительного делителя
0x4C	RTC_ALRM	Регистр значения для сравнения с основным счетчиком RTC_CNT и выработки сигнала ALRF
0x50	RTC_CS	Регистр управления и состояния флагов часов реального времени

- 15.3.1 BKP\_REG\_00
- 15.3.2 BKP\_REG\_01
- 15.3.3 BKP\_REG\_02
- 15.3.4 BKP\_REG\_03
- 15.3.5 BKP\_REG\_04
- 15.3.6 BKP\_REG\_05
- 15.3.7 BKP\_REG\_06
- 15.3.8 BKP\_REG\_07
- 15.3.9 BKP\_REG\_08
- 15.3.10 BKP\_REG\_09
- 15.3.11 BKP\_REG\_0A
- 15.3.12 BKP\_REG\_0B
- 15.3.13 BKP\_REG\_0C
- 15.3.14 BKP\_REG\_0D

Таблица 126 – Регистры BKP\_REG\_[0D...00]

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

Таблица 127 – Описание битов регистров Регистры BKP\_REG\_[0D...00]

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	BKP_REG[31:0]	Регистр аварийного сохранения

15.3.15 ВКР\_REG\_0E

Таблица 128 – Регистр ВКР\_REG\_0E

Номер	15	14	13, 12	11	10...8	7,6	5...3	2...0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	00	0	000	10	000	000
	ilimen	COVDET	Trim [4:3]	FPOR	Trim[2:0]	SW_EN[1:0]	SelectRI [2:0]	LOW [2:0]

Таблица 129 – Описание битов регистра ВКР\_REG\_0E

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15	ilimen	Бит разрешения защиты ограничения регулятора по току 150 мА
14	COVDET	Признак несанкционированного вскрытия устройства: 1 – вскрытия не было; 0 – осуществлялось вскрытие
13, 12	Trim[4:3]	Коэффициент настройки опорного напряжения регулятора: 00 – 1,8 В; 01 – 1,6 В; 10 – 1,4 В; 11 – 1,2 В
11	FPOR	Флаг срабатывания POR. Устанавливается в 1 загрузочным ПЗУ после сброса по питанию, при сбросе по питанию устанавливается в 0. Служит для анализа загрузочным ПЗУ, что сейчас идет выполнение программы после системного или программного сброса, либо после сброса по питанию
10...8	Trim[2:0]	Коэффициент настройки опорного напряжения встроенного регулятора напряжения DU <sub>CC</sub> . С помощью Trim осуществляется подстройка напряжения DU <sub>CC</sub> : 000 – DU <sub>CC</sub> + 0,10 В – значение по умолчанию; 001 – DU <sub>CC</sub> + 0,06 В; 010 – DU <sub>CC</sub> + 0,04 В; 011 – DU <sub>CC</sub> + 0,01 В; 100 – DU <sub>CC</sub> – 0,01 В; 101 – DU <sub>CC</sub> – 0,04 В; 110 – DU <sub>CC</sub> – 0,06 В; 111 – DU <sub>CC</sub> – 0,10 В
7, 6	SW_EN[1:0]	Разрешение работы порта Serial Wire: 00 – запрещен; 01,10 – разрешен; 11 – зарезервировано
5...3	SelectRI[2:0]	Выбор дополнительной стабилизирующей нагрузки для встроенного регулятора напряжения DU <sub>CC</sub> : 000 – ~6 КОм (дополнительный ток потребления 300 мкА); 001 – ~270 КОм (дополнительный ток потребления 6,6 мкА);

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		010 – ~90 КОм (дополнительный ток потребления 20 мкА); 011 – ~24 КОм (дополнительный ток потребления 80 мкА); 100 – ~900 КОм (собственное потребление 2 мкА); 101 – ~2 КОм (дополнительный ток потребления 900 мкА); 110 – ~400 Ом (дополнительный ток потребления 4,4 мА); 111 – ~100 Ом (дополнительный ток потребления 19 мА)
2...0	LOW[2:0]	Выбор режима работы встроенного регулятора напряжения DU <sub>CC</sub> . Значение LOW должно совпадать с значением SelectRI и выставляться в зависимости от тактовой частоты микросхемы: 000 – Частота до 10 МГц; 001 – Частота до 200 КГц; 010 – Частота до 500 КГц; 011 – Частота до 1 МГц; 100 – При выключении всех генераторов; 101 – Частота до 40 МГц; 110 – Частота до 80 МГц; 111 – Частота более 80 МГц

### 15.3.16 BKP\_REG\_0F

Таблица 130 – Регистр BKP\_REG\_0F

Номер	15	14	13	12...5	4	3, 2	1	0
Доступ	R/W	U	RO	R/W	R/W	R/W	R/W	R/W
Сброс	1	0	0	0000000	0	00	0	0
	LSI_ON	-	LSE_RDY	RTC_CAL[7:0]	RTC_EN	RTC_SEL[1:0]	LSE_BYP	LSE_ON

Номер	31	30	29...24	23	22	21	20...16
Доступ	R/W	R/W	R/W	R/W	R/W	RO	R/W
Сброс	0	0	100000	1	1	1	10000
	RTC_RESET	STANDBY	HSI_TRIM[5:0]	HSI_RDY	HSI_ON	LSI_RDY	LSI_TRIM[4:0]

Таблица 131 – Описание битов регистра BKP\_REG\_0F

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	RTC_RESET	Сброс часов реального времени: 0 – часы не сбрасываются; 1 – часы сбрасываются
30	STANDBY	Режим отключения регулятора DU <sub>CC</sub> на 1,8В: 0 – регулятор включен и выдает напряжение; 1 – выключение регулятора. Триггер сбрасывается по событию ALRF или по низкому уровню на выводе WAKEUP

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
29..24	HSI_TRIM[5:0]	Коэффициент подстройки частоты генератора HSI. Смотри диаграмму зависимости (рисунок 33)
23	HSI_RDY	Флаг выхода генератора HSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
22	HSI_ON	Бит управления генератором HSI: 0 – генератор выключен; 1 – генератор включен
21	LSI_RDY	Флаг выхода генератора LSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
20..16	LSI_TRIM[4:0]	Коэффициент подстройки частоты генератора LSI. Смотри диаграмму зависимости (рисунок 32)
15	LSI_ON	Бит управления генератором LSI: 0 – генератор выключен; 1 – генератор включен
14	-	Зарезервировано
13	LSE_RDY	Флаг выхода генератора LSE в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
12..5	RTC_CAL[7:0]	Коэффициент подстройки тактовой частоты часов реального времени RTC_CLK. Из каждых $2^{20}$ тактов RTC_CLK будет замаскировано RTC_CAL тактов: 00000000 – 0 тактов; 00000001 – 1 такт; .... 11111111 – 255 тактов. Таким образом, если исходная частота равна 32768,00000 Гц, то: при RTC_CAL = 0 частота RTC_CLK = 32768,00000 Гц; при RTC_CAL = 1 частота RTC_CLK = 32767,96875 Гц; ... при RTC_CAL = 255 частота RTC_CLK = 32760,03125 Гц. В общем случае тактовая частота RTC_CLK после подстройки равна: $RTC\_CLK = CLK - RTC\_CAL \cdot CLK / 2^{20}$ , где CLK – исходная тактовая частота часов реального времени в Гц
4	RTC_EN	Бит разрешения работы часов реального времени: 0 – работа запрещена; 1 – работа разрешена
3, 2	RTC_SEL[1:0]	Биты выбора источника тактовой синхронизации часов реального времени RTC_CLK: 00 – LSI; 01 – LSE; 10 – HSI_RTC (формируется в блоке RST_CLK); 11 – HSE_RTC (формируется в блоке RST_CLK)

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1	LSE_BYR	Бит управления генератором LSE: 0 – режим осциллятора; 1 – режим работы на проход (внешний генератор). Устанавливать совместно с LSE_ON=1
0	LSE_ON	Бит управления генератором LSE: 0 – генератор выключен; 1 – генератор включен

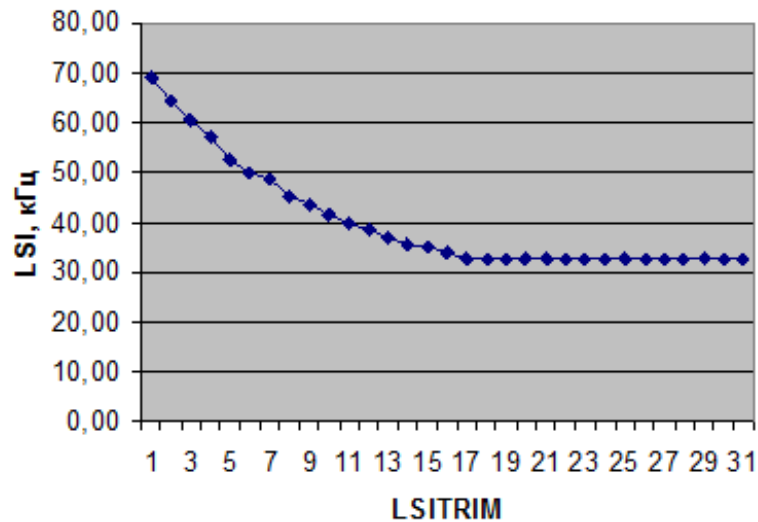


Рисунок 32 – Зависимость частоты LSI от значения LSITRIM

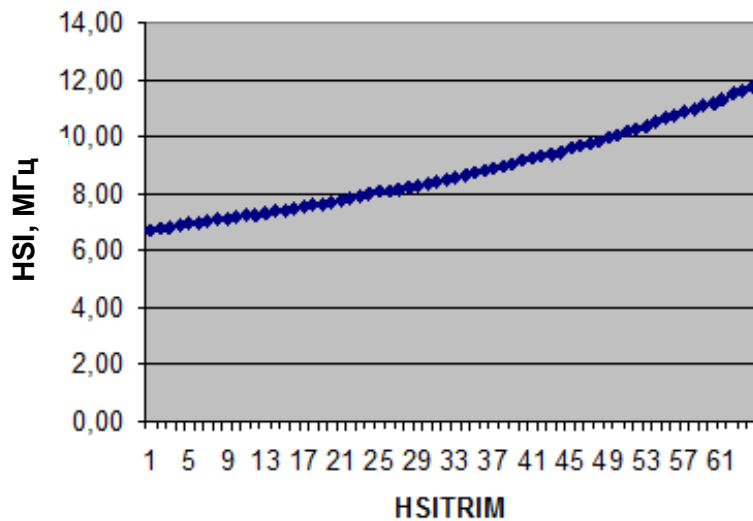


Рисунок 33 – Зависимость частоты HSI от значения HSITRIM

### 15.3.17 RTC\_CNT

Таблица 132 – Регистр RTC\_CNT

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

Таблица 133 – Описание битов регистра RTC\_CNT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...0	RTC_CNT[31:0]	Значение основного счетчика часов реального времени

### 15.3.18 RTC\_DIV

Таблица 134 – Регистр RTC\_DIV

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

Таблица 135 – Описание битов регистра RTC\_DIV

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

### 15.3.19 RTC\_PRL

Таблица 136 – Регистр RTC\_PRL

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

Таблица 137 – Описание битов регистра RTC\_PRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	-	Зарезервировано
31...0	RTC_PRL[19:0]	Коэффициент деления тактовой частоты RTC_CLK: $RTC\_CLKd = RTC\_CLK / (RTC\_PRL + 1)$

### 15.3.20 RTC\_ALRM

Таблица 138 – Регистр RTC\_ALRM

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



Таблица 139 – Описание битов регистра RTC\_ALARM

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..0	RTC_ALARM[31:0]	Значение для сравнения с основным счетчиком RTC_CNT. Сигнал ALRF вырабатывается в момент превышения основным счетчиком RTC_CNT значения RTC_ALARM (RTC_CNT == (RTC_ALARM+1))

### 15.3.21 RTC\_CS

Таблица 140 – Регистр RTC\_CS

Номер	31...7	6	5	4	3	2	1	0
Доступ	U	RO	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	-	WEC	ALRF_IE	SECF_IE	OWF_IE	ALRF	SECF	OWF

Таблица 141 – Описание битов регистра RTC\_CS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
30...7	-	Зарезервировано
6	WEC	Флаг выполнения записи в регистры RTC: 0 – можно записывать в регистры RTC; 1 – идет запись в регистры RTC, запись в регистры запрещена
5	ALRF_IE	Флаг разрешения прерывания по событию ALRF: 0 – прерывание запрещено; 1 – прерывание разрешено
4	SECF_IE	Флаг разрешения прерывания по событию SECF: 0 – прерывание запрещено; 1 – прерывание разрешено
3	OWF_IE	Флаг разрешения прерывания по событию OWF: 0 – прерывание запрещено; 1 – прерывание разрешено
2	ALRF	Флаг превышения основным счетчиком RTC_CNT значения RTC_ALARM: – нет события превышения; – было событие превышения. Флаг устанавливается при RTC_CNT == (RTC_ALARM+1). Сброс флага осуществляется записью 1
1	SECF	Флаг изменения значения основного счетчика RTC_CNT (инкремент счетчика RTC_CNT выполняется один раз в секунду): 0 – нет изменения RTC_CNT; 1 – есть изменение RTC_CNT. Сброс флага осуществляется записью 1
0	OWF	Флаг переполнения основного счетчика RTC_CNT: 0 – нет переполнения; 1 – было переполнение. Сброс флага осуществляется записью 1

## 16 Порты ввода-вывода

Микросхема имеет три порта ввода/вывода – А, В, С. Порты 16-разрядные (кроме порта В) и их выходы мультиплексируются между различными функциональными блоками, управление для каждого вывода отдельное. Чтобы выходы порта перешли под управление того или иного периферийного блока необходимо задать для нужных выводов выполняемую функцию и настройки.

При работе в режиме отладки не допускается изменение функций выводов, совмещенных с выводами SWD, путем записи 1 в соответствующие биты регистров RXTX, SETTX и OE. Это может привести к блокировке интерфейса отладки.

Таблица 142 – Описание выводов по портам

Вывод	Аналоговая функция ANALOG_EN=0		Цифровая функция		
			Порт IO MODE=0 ANALOG_EN=1	Основная MODE=1 ANALOG_EN=1	
Порт А					
PA0	S0	10	PA0	TMR0_CH1	3
PA1	S1		PA1	TMR0_CH1N	
PA2	S2		PA2	-	
PA3	S3		PA3	-	
PA4	S4		PA4	-	
PA5	S5		PA5	-	
PA6	S6		PA6	-	
PA7	S7		PA7	-	
PA8	S8		PA8	TMR0_ETR	
PA9	S9		PA9	TMR0_BRK	
PA10	S10		PA10	-	
PA11	S11		PA11	-	
PA12	S12		PA12	-	
PA13	S13		PA13	-	
PA14	S14	10	PA14	-	
PA15	S15		PA15	-	
Порт В					
PB0	–		PB0 MODE0	UART0_TXD	4
PB1	–		PB1	UART0_RXD	
PB2	–		PB2 SWCLKTCK	nSIROUT0	
PB3	–		PB3 SWDIO	nSIRIN0	
PB4	OSC_IN32	1	PB4	nUART0DTR	
PB5	OSC_OUT32		PB5	nUART0RTS	
PB6	ADC7S/BP0	2,10	PB6	nUART0RI	
PB7	ADC6S/BP1		PB7	nUART0DCD	
PB8	ADC5S/BP2		PB8	nUART0DSR	
PB9	ADC4S/BP3		PB9	nUART0CTS	

Вывод	Аналоговая функция ANALOG_EN=0		Цифровая функция		
			Порт IO MODE=0 ANALOG_EN=1	Основная MODE=1 ANALOG_EN=1	
PB10	S16	10	PB10	-	5
PB11	S17		PB11	-	
PB12	S18		PB12	-	
PB13	S19		PB13	-	
PB14	S20		PB14	-	
Порт C					
PC0	-	2	PC0 MODE1	UART1_TXD	6
PC1	ADC3S		PC1	UART1_RXD	
PC2	ADC2S/VLCD		PC2	TMR1_CH1	5
PC3	ADC1S/ADCS_REF+		PC3	TMR1_CH1N	
PC4	ADC0S/ADCS_REF-		PC4	-	
PC5	S21	10	PC5	TMR1_ETR	5
PC6	S22		PC6	TMR1_BRK	
PC7	S23		PC7	EXT_INT0	9
PC8	S24		PC8	EXT_INT1	
PC9	S25		PC9	EXT_INT2	
PC10	S26		PC10	SSP_FSS	7
PC11	S27		PC11	SSP_CLK	
PC12	S28		PC12	SSP_RXD	
PC13	S29		PC13	SSP_TXD	
PC14	-			PC14	SCL
PC15	-	PC15		SDA	
<p>Обозначения</p> <p>1 – Генератор LSE;</p> <p>2 – АЦП последовательного приближения;</p> <p>3 – Таймер 0;</p> <p>4 – UART0;</p> <p>5 – Таймер 1;</p> <p>6 – UART1;</p> <p>7 – Последовательный интерфейс SSP;</p> <p>8 – Последовательный интерфейс I2C;</p> <p>9 – Внешние прерывания;</p> <p>10 – Контроллер LCD</p>					

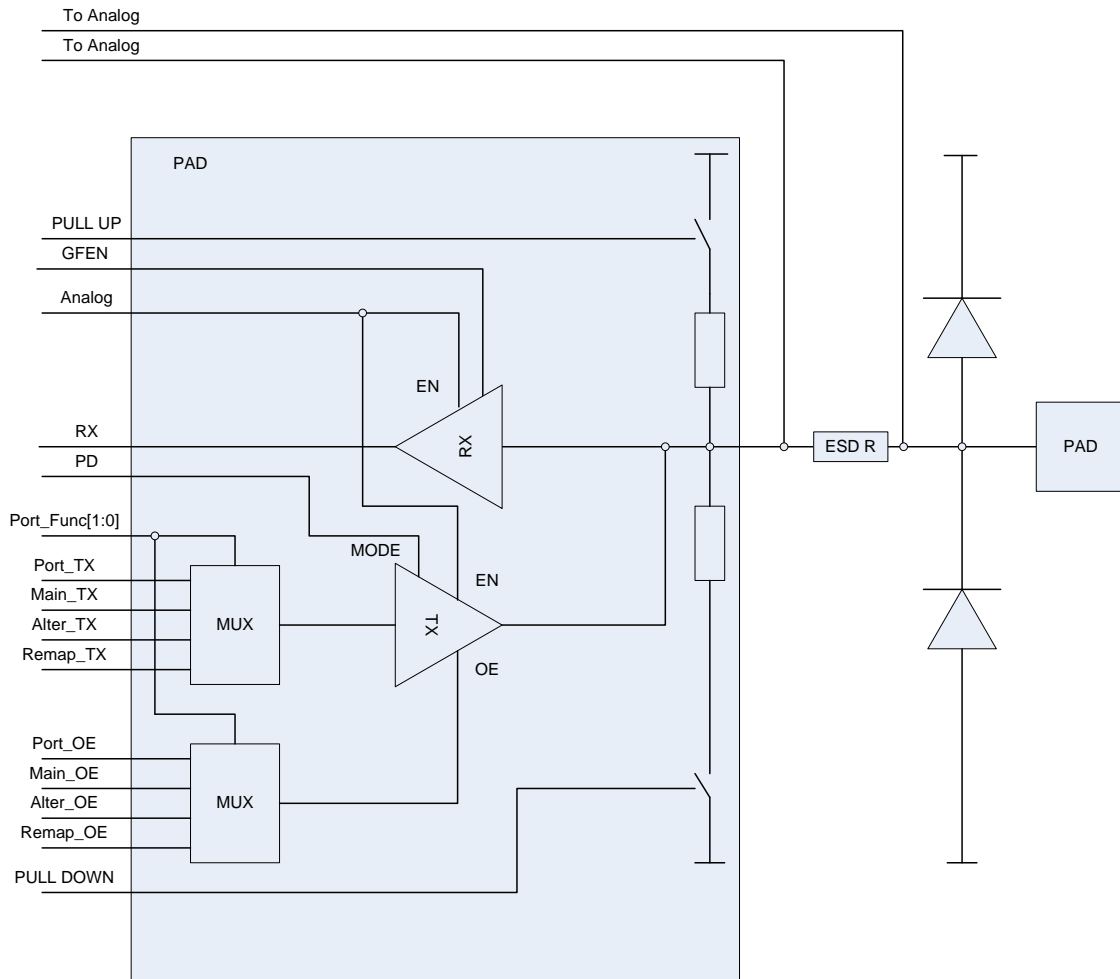


Рисунок 34 – Структурная схема

### 16.1 Описание регистров портов ввода-вывода

Таблица 143 – Описание регистров портов ввода-вывода

Базовый адрес	Название	Описание
0x4008_0000	GPIO1	Порт А
0x4008_8000	GPIO2	Порт В
0x4009_0000	GPIO3	Порт С
Смещение		
0x00	PORT_RXTX[15:0]	Данные порта
0x04	PORT_OE[15:0]	Направление порта
0x08	PORT_FUNC[31:0]	Режим работы порта
0x0C	PORT_ANALOG[15:0]	Аналоговый режим работы порта
0x10	PORT_PULL[31:0]	Подтяжка порта
0x14	PORT_PD[31:0]	Режим работы выходного драйвера
0x18	PORT_PWR[31:0]	Режим мощности передатчика
0x1C	PORT_GFEN[15:0]	Режим работы входного фильтра
0x20	PORT_SETTX[15:0]	Регистр SET_TX записью 1 устанавливает 1 в регистре PORT_RXTX
0x24	PORT_CLRTX[15:0]	Регистр CLR_TX записью 1 устанавливает 0 в регистре RXTX
0x28	PORT_RDTX	Регистр позволяет читать то, что записано в выходной регистр порта

### 16.1.1 PORTx\_RXTX

Таблица 144 – Регистр PORTx\_RXTX

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

Таблица 145 – Описание битов регистра PORTx\_RXTX

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...0	PORT_RXTX[15:0]	Режим работы контроллера. Данные для выдачи на выходы порта и для чтения

### 16.1.2 PORTx\_OE

Таблица 146 – Регистр PORTx\_OE

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

Таблица 147 – Описание битов регистра PORTx\_OE

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

### 16.1.3 PORTx\_FUNC

Таблица 148 – Регистр PORTx\_FUNC

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

Таблица 149 – Описание битов регистра PORTx\_FUNC

Номер бита	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

### 16.1.4 PORTx\_ANALOG

Таблица 150 – Регистр PORTx\_ANALOG

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

Таблица 151 – Описание битов регистра PORTx\_ANALOG

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

### 16.1.5 PORTx\_PULL

Таблица 152 – Регистр PORTx\_PULL

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

Таблица 153 – Описание битов регистра PORTx\_PULL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	PULL UP15:0]	Режим работы контроллера: Разрешение подтяжки вверх; 0 – подтяжка в питание выключена; 1 – подтяжка в питание включена (есть подтяжка)
15...0	PULL DOWN[15:0]	Режим работы контроллера. Разрешение подтяжки вниз: 1 – подтяжка в ноль включена (есть подтяжка); 0 – подтяжка в ноль выключена

### 16.1.6 PORTx\_PD

Таблица 154 – Регистр PORTx\_PD

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

Таблица 155 – Описание битов регистра PORTx\_PD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	PORT SHM[15:0]	Режим работы контроллера. Режим работы входа: 0 – триггер Шмитта выключен, гистерезис 200 мВ; 1 – триггер Шмитта включен, гистерезис 400 мВ
15...0	PORT PD[15:0]	Режим работы контроллера. Режим работы выхода: 0 – управляемый драйвер; 1 – открытый сток

### 16.1.7 PORTx\_PWR

Таблица 156 – Регистр PORTx\_RXTX

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

Таблица 157 – Описание битов регистра PORTx\_PWR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..2	PWRx	Аналогично PWR0 для остальных битов порта
1...0	PWR0[1:0]	Режим работы вывода порта: 00 – зарезервировано; 01 – медленный фронт; 10 – быстрый фронт; 11 – максимально быстрый фронт

### 16.1.8 PORTx\_GFEN

Таблица 158 – Регистр PORTx\_GFEN

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

Таблица 159 – Описание битов регистра PORTx\_GFEN

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16		
15...0	GFEN[15:0]	Режим работы входного фильтра: 0 – фильтр выключен; 1 – фильтр включен

### 16.1.9 PORTx\_SETTX

Таблица 160 – Регистр PORTx\_SETTX

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

Таблица 161 – Описание битов регистра PORTx\_SETTX

Номер бита	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 1 в соответствующий разряд регистра устанавливает в 1 соответствующий разряд выхода порта PORTx_RXTX. Запись 0 не влияет на состояние соответствующего выхода порта PORTx_RXTX. Читается ранее записанным в регистр значениями, а не состояние соответствующих входов порта PORT_RXTX

### 16.1.10 PORTx\_CLRTX

Таблица 162 – Регистр PORTx\_CLRTX

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

Таблица 163 – Описание битов регистра PORTx\_CLRTX

Номер бита	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 ноль соответствующий разряд выхода порта PORTx_RXTX Запись 0 не влияет на состояние соответствующего выхода порта PORTx_RXTX. Читается ранее записанным в регистр значениями, а не состояние соответствующих входов порта PORTx_RXTX



## 17 Детектор напряжения питания

Блок детектора напряжения питания PVD предназначен для контроля питания  $U_{CC}$  и  $BU_{CC}$  при работе микросхемы. 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 программной обработки.

Уровень опорного напряжения для сравнения с  $U_{CC}$  задается битами PLS[3:0] в регистре PVDCS, для сравнения с  $BU_{CC}$  задается битами PLBS[3:0] в регистре PVDCS. В соответствии с уровнями напряжения формируются флаги PVD и PBVD. Данные флаги выставляются при возникновении события и сбрасываются программно.

Таблица 164 – Параметры детектора напряжения питания

Параметр	Не менее	Типовое	Не более
Входное напряжение, $U_{CC}$ , В	2,2	-	3,6
Входное напряжение, $BU_{CC}$ , В	1,8	-	3,6
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «0000», В		1,0	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «0001», В		1,2	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «0010», В		1,4	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «0011», В		1,6	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «0100», В		1,8	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «0101», В		2,0	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «0110», В		2,2	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «0111», В		2,4	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «1000», В		2,6	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «1001», В		2,8	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «1010», В		3,0	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «1011», В		3,2	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «1100», В		3,4	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «1101», В		3,6	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «1110», В		3,8	
Уровень срабатывания PVD от $U_{CC}$ , при PLS = «1111», В		4,0	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «0000», В		1,0	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «0001», В		1,2	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «0010», В		1,4	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «0011», В		1,6	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «0100», В		1,8	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «0101», В		2,0	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «0110», В		2,2	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «0111», В		2,4	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «1000», В		2,6	
Уровень срабатывания PBVD от $BU_{CC}$ , при PBLBS = «1001», В		2,8	

Параметр	Не менее	Типовое	Не более
Уровень срабатывания PBVD от BU <sub>CC</sub> , при PBLIS = «1010», В		3,0	
Уровень срабатывания PBVD от BU <sub>CC</sub> , при PBLIS = «1011», В		3,2	
Уровень срабатывания PBVD от BU <sub>CC</sub> , при PBLIS = «1100», В		3,4	
Уровень срабатывания PBVD от BU <sub>CC</sub> , при PBLIS = «1101», В		3,6	
Уровень срабатывания PBVD от BU <sub>CC</sub> , при PBLIS = «1110», В		3,8	
Уровень срабатывания PBVD от BU <sub>CC</sub> , при PBLIS = «1111», В		4,0	

## 17.1 Описание регистров блока PVD

Таблица 165 – Перечень регистров блока PVD

Базовый адрес	Название	Описание
0x4005_8000	POWER	Датчик подсистемы питания
Смещение		
0x00	PVDCS [15:0]	Регистр управления и состояния датчика питания

### 17.1.1 PVDCS

Номер	31...16	15	14	13
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-	PVDBEN	INV	INVB

Номер	12	11	10	9	8...5	4...1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	000	00	0
	IEPVD	IEPVBD	PVD	PVBD	PLS[3:0]	PBLIS[3:0]	PVDEN

Таблица 166 – Описание битов регистра PVDCS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15	PVDBEN	Бит разрешения работы блока датчика напряжения питания BU <sub>CC</sub> : 0 – датчик отключен; 1 – датчик включен
14	INV	Флаг инверсии выхода от датчика PVD: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то флаг выставляется при превышении заданного уровня, если инвертируется, то при снижении ниже заданного уровня

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
13	INVB	Флаг инверсии выхода от датчика PVBD: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то флаг выставляется при превышении заданного уровня, если инвертируется, то при снижении ниже заданного уровня
12	IEPVD	Флаг разрешения прерывания от датчика PVD: 0 – прерывание запрещено; 1 – прерывание разрешено. Очищается записью 0, если при очистке, датчик продолжает выдавать сигнал, то флаг не будет очищен
11	IEPVBD	Флаг разрешения прерывания от датчика PVBD: 0 – прерывание запрещено; 1 – прерывание разрешено. Очищается записью 0, если при очистке, датчик продолжает выдавать сигнал, то флаг не будет очищен
10	PVD	Результат сравнения напряжения основного питания: 0 – напряжение питания меньше чем уровень, задаваемый PLS; 1 – напряжение питания больше чем уровень, задаваемый PLS. Примечание – сброс флага необходимо проводить с подтверждением - сбрасывать дважды
9	PVBD	Результат сравнения напряжения батарейного питания: 0 – напряжение питания меньше чем уровень, задаваемый PBLIS; 1 – напряжение питания больше чем уровень, задаваемый PBLIS. Примечание – сброс флага необходимо проводить с подтверждением - сбрасывать дважды
8...5	PLS[3:0]	Уровень напряжения для сравнения с напряжением основного питания; 0000 – 1,0 В; 0001 – 1,2 В; 0010 – 1,4 В; 0011 – 1,6 В; 0100 – 1,8 В; 0101 – 2,0 В; 0110 – 2,2 В; 0111 – 2,4 В; 1000 – 2,6 В; 1001 – 2,8 В; 1010 – 3,0 В; 1011 – 3,2 В; 1100 – 3,4 В; 1101 – 3,6 В; 1110 – 3,8 В; 1111 – 4,0 В

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
4...1	PBLS[3:0]	Уровень напряжения для сравнения с напряжением батарейного питания: 0000 – 1,0 В; 0001 – 1,2 В; 0010 – 1,4 В; 0011 – 1,6 В; 0100 – 1,8 В; 0101 – 2,0 В; 0110 – 2,2 В; 0111 – 2,4 В; 1000 – 2,6 В; 1001 – 2,8 В; 1010 – 3,0 В; 1011 – 3,2 В; 1100 – 3,4 В; 1101 – 3,6 В; 1110 – 3,8 В; 1111 – 4,0 В
0	PVDEN	Бит разрешения работы блока датчика напряжения питания $U_{CC}$ : 0 – датчик отключен; 1 – датчик включен

## 18 Таймеры общего назначения

В микросхеме реализовано два 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 по заданным событиям;
- захват входного сигнала, в том числе измерение длительности импульсов входного сигнала;
- генерацию различных форм выходных сигналов.

Основу таймеров составляет 16-битный перезагружаемый счетчик. Счет может быть прямой, обратный или двунаправленный. В качестве источника синхросигнала может выступать внутренняя тактовая частота TIM\_CLK, внешние сигналы или другой таймер.

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

Каждый из таймеров позволяет генерировать прерывания.

### 18.1 Основные характеристики

Основные характеристики блока таймера:

- 16-битный счетчик: счет прямой, обратный или двунаправленный;
- 16-разрядный предварительный делитель частоты TIM\_CLK;
- схема выбора источника тактирования основного счетчика от внешних сигналов или от другого таймера;
- один канал, который 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 по событиям:
  - обновление счетчика;
  - захват;
  - сравнение;
  - внешние события по входам ETR и BRK.

### 18.1.1 Структурная схема

Структурная схема блока «Таймер» представлена на рисунке 35.

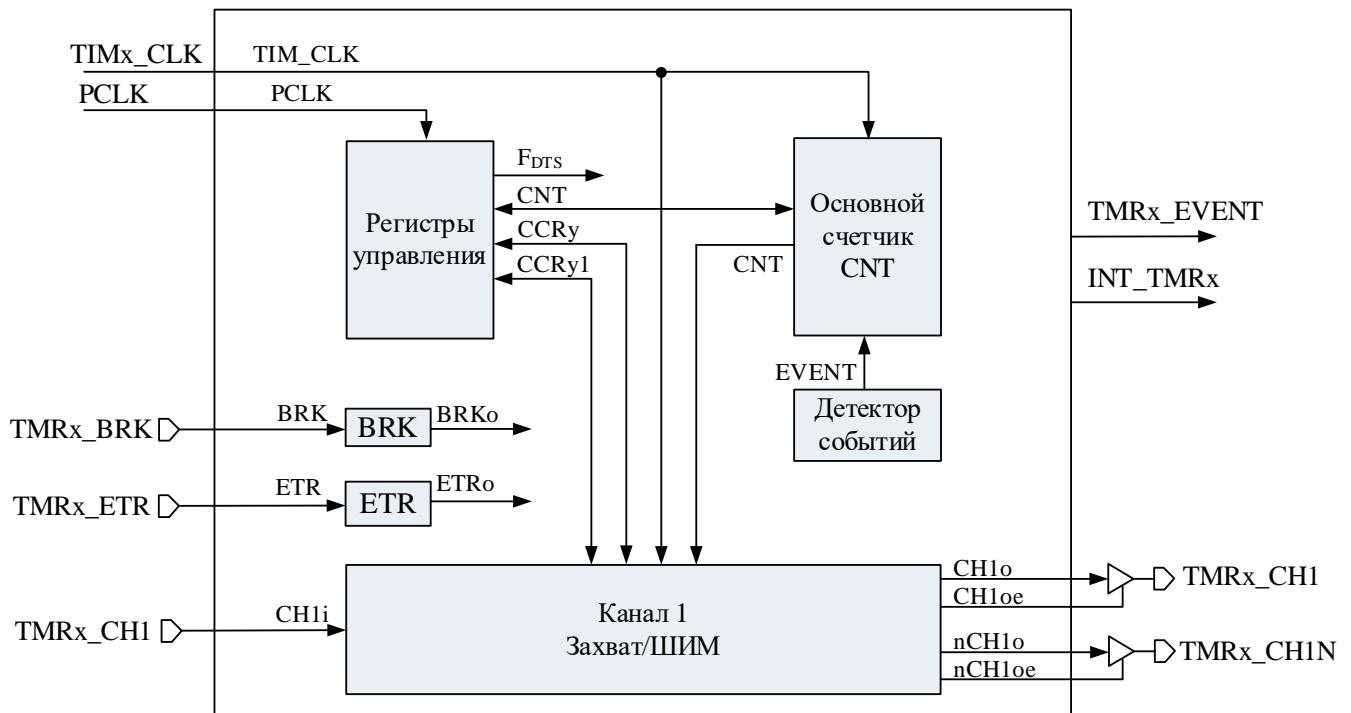


Рисунок 35 – Структурная схема блока «Таймер»

Таймер содержит основной 16-битный счетчик CNT, блок регистров управления и один канал со схемой захвата/ШИМ.

Таймер позволяет работать в режимах:

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

### 18.2 Базовый блок таймера

Таймер построен на базе 16-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 18.3 «Источники событий для счета»).

### 18.2.1 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 0, бит 15 для таймера 1. Задание тактовой частоты TIM\_CLK осуществляется в регистре TIM\_CLOCK: в поле TIMxBRG[7:0] устанавливается коэффициент деления тактовой частоты TIMx\_C2 для формирования частоты TIMx\_CLK, разрешение подачи частоты TIMx\_CLK на блок таймера управляется битом TIMxCLKEN. Тактовая частота TIM\_CLK должна быть меньше или равна тактовой частоте PCLK (CPU\_CLK).

После подачи тактовых сигналов можно приступить к работе с таймером.

### 18.2.2 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 EVENT\_SEL[3:0];
  - режим счета основного счетчика CNT\_MODE[1:0]:
    - значения 00 и 01 при тактировании внутренней частотой;
    - значения 10 при тактировании внешними сигналами;
  - направление счета основного счетчика DIR;
- при тактировании внутренней частотой установить значение предварительного делителя в регистре PSG, основной счетчик при этом будет считать на частоте  $TIM\_CLKd = TIM\_CLK / (PSG + 1)$ ;
- разрешить работу счетчика CNT\_EN.

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

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

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

- CNT\_MODE[1:0] = 00 или 10 – направление счета определяется битом DIR:
  - DIR = 0 – счет прямой;
  - DIR = 1 – счет обратный;

– CNT\_MODE[1:0] = 01 – счет двунаправленный с автоматическим изменением DIR.

### 18.2.3 Режимы счета

#### 18.2.3.1 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
```

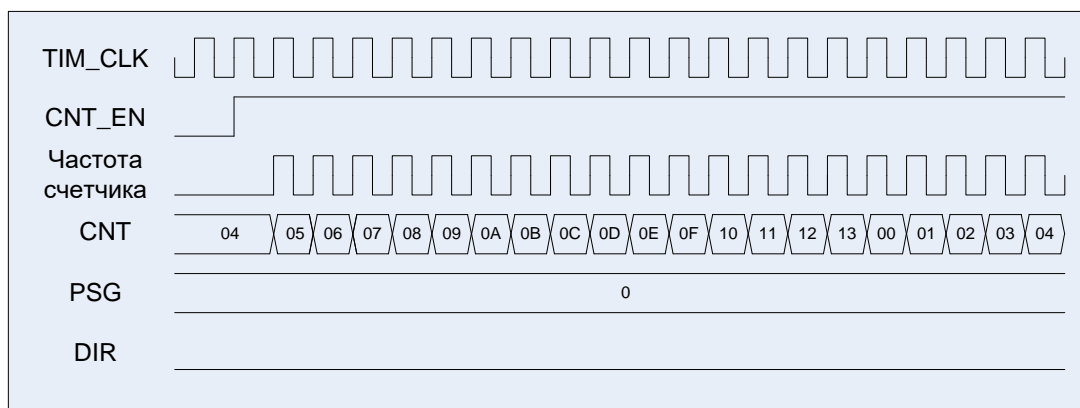


Рисунок 36 – Диаграммы работы таймера, счет прямой от 0 до 0x13, стартовое значение 0x04

#### 18.2.3.2 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
```

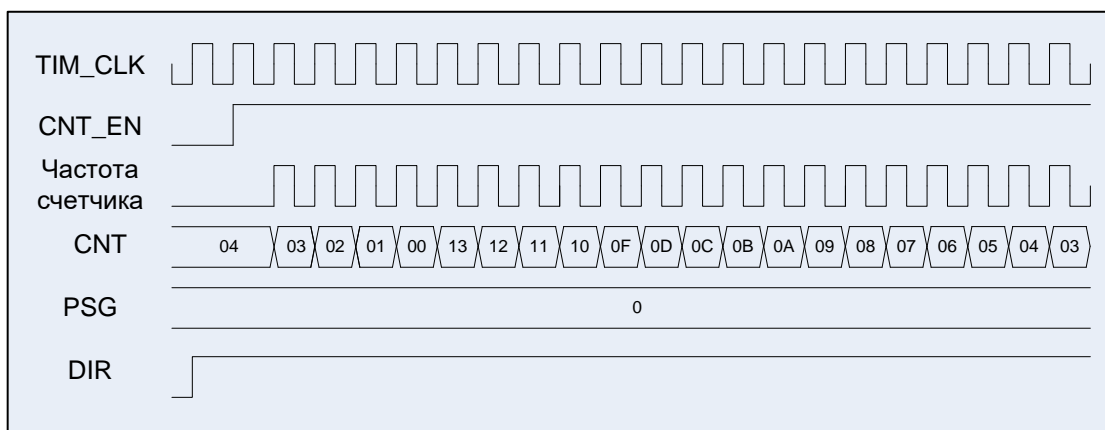


Рисунок 37 – Диаграммы работы таймера, счет обратный от 0x13 до 0, стартовое значение 0x04



**18.2.3.3**      *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
```

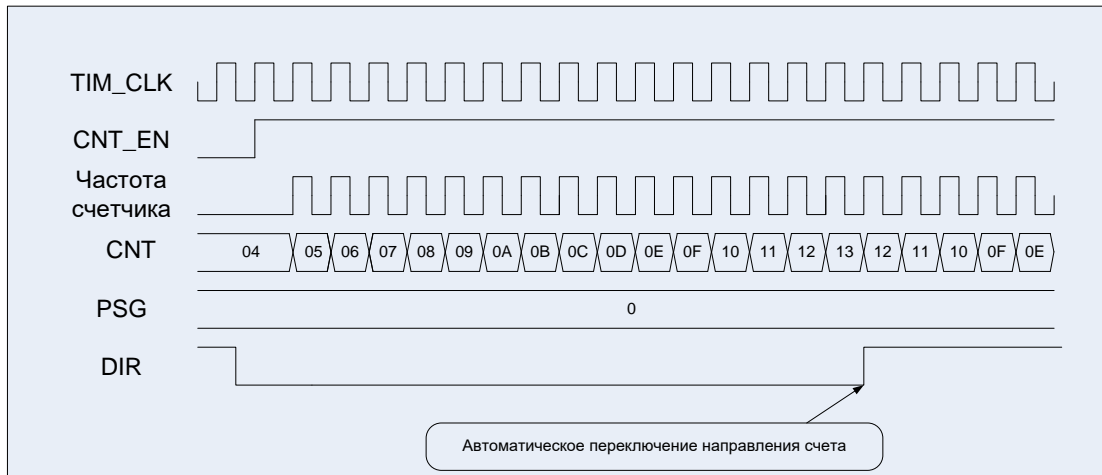


Рисунок 38 – Диаграммы работы таймера, счет двунаправленный, сначала прямой

**18.2.3.4**      *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
```

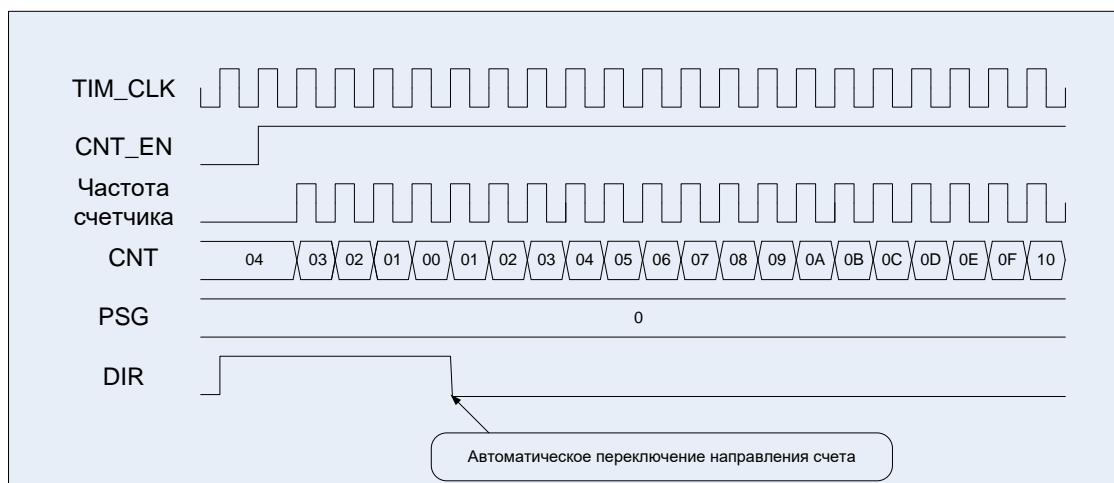


Рисунок 39 – Диаграммы работы таймера, счет двунаправленный, сначала обратный

### 18.2.4 Тактовая частота $F_{DTS}$

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

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

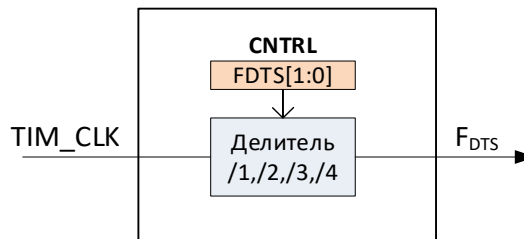


Рисунок 40 – Схема формирования тактовой частоты  $F_{DTS}$

Диаграмма возможных частот  $F_{DTS}$  приведена на рисунке 41.

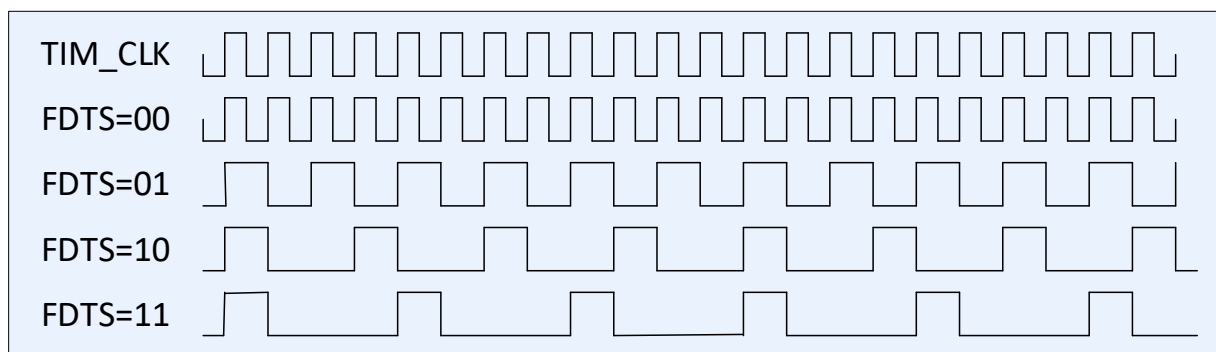


Рисунок 41 – Диаграмма тактовой частоты  $F_{DTS}$  в зависимости от значения FDTs[1:0] в регистре CNTRL

### 18.3 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 CH1i;
- внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR.

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

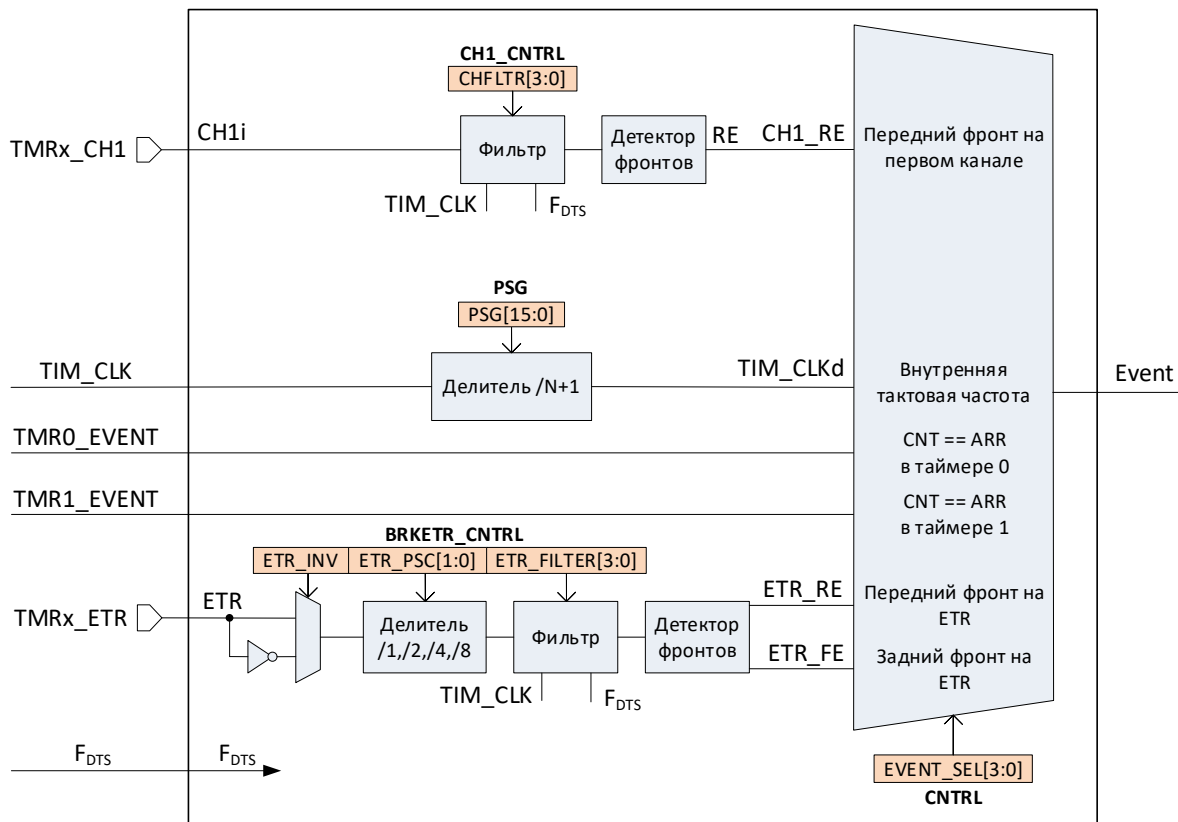


Рисунок 42 – Структурная схема формирования события для счета

### 18.3.1 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 43 и 44 приведены диаграммы работы счетчика при обновлении значения PSG.

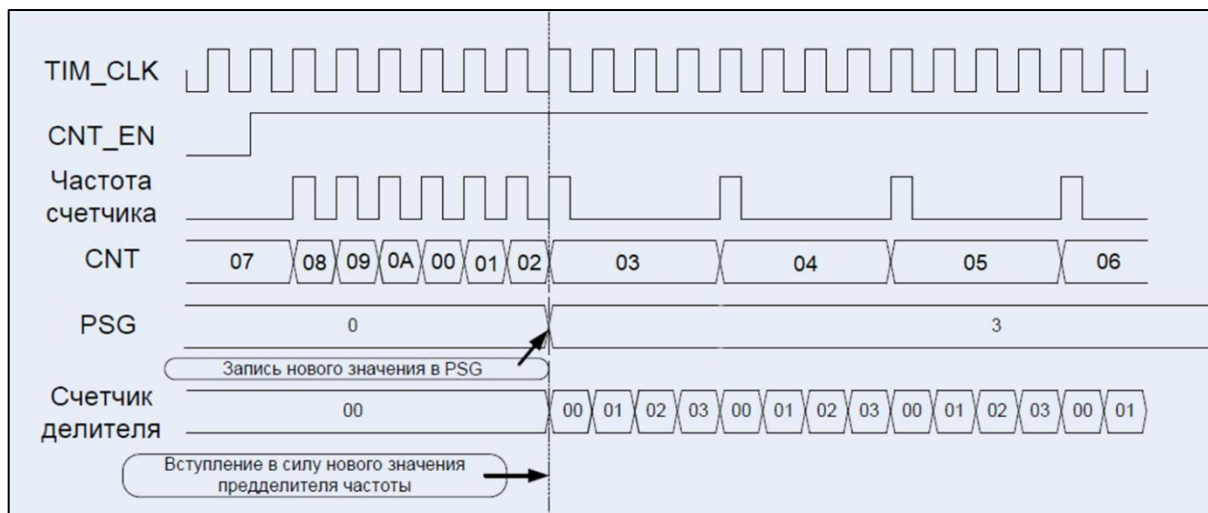


Рисунок 43 – Диаграмма работы счетчика: счет прямой (CNT\_MODE[1:0] = 00, EVENT\_SEL[3:0] = 0000, DIR = 0)

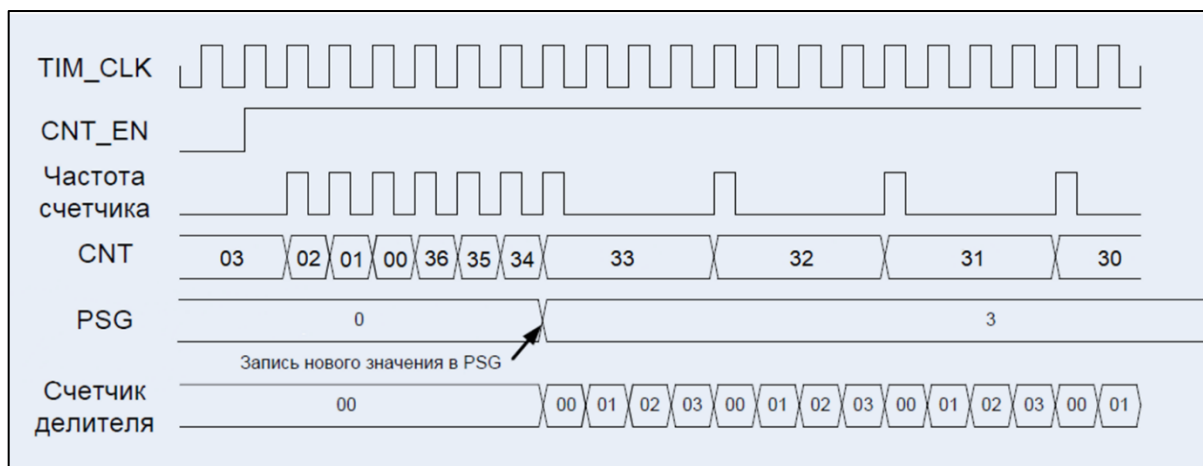


Рисунок 44 – Диаграмма работы счетчика: счет обратный (CNT\_MODE[1:0] = 00, EVENT\_SEL[3:0] = 0000, DIR = 1)

### 18.3.2 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\_EVENT, который соединен с входом другого таймера. Тактирование от другого таймера выбирается, когда EVENT\_SEL[3:0] = 0001 или 0010, а также CNT\_MODE[1:0] = 10 в регистре CNTRL. Основной счетчик таймера тактируется от другого таймера по сигналу TMRx\_EVENT, который устанавливается при CNT == ARR. Пересинхронизация сигнала TMRx\_EVENT (CNT == ARR) с одного таймера на другой происходит с задержкой один такт частоты TIM\_CLK.

Синхронизация таймеров возможна в различных режимах. На рисунке 45 показан пример каскадного соединения таймеров, диаграммы работы данных таймеров приведены на рисунке 46.

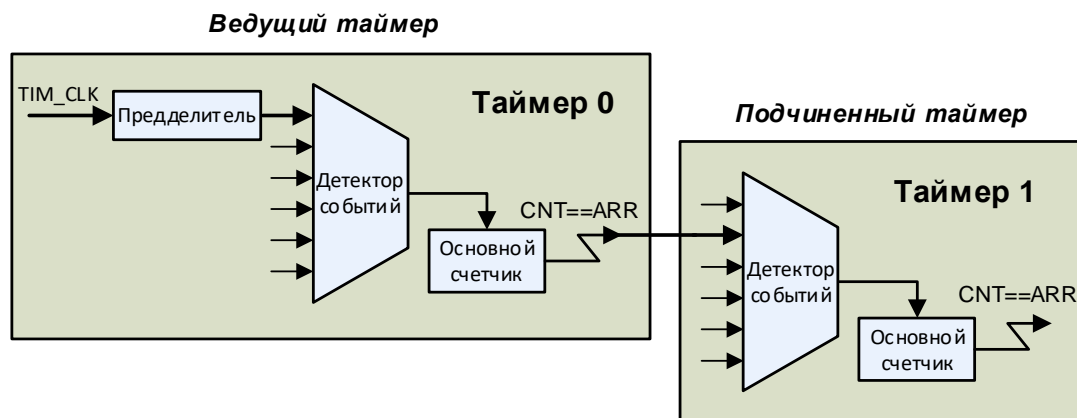


Рисунок 45 – Пример каскадного соединения таймеров

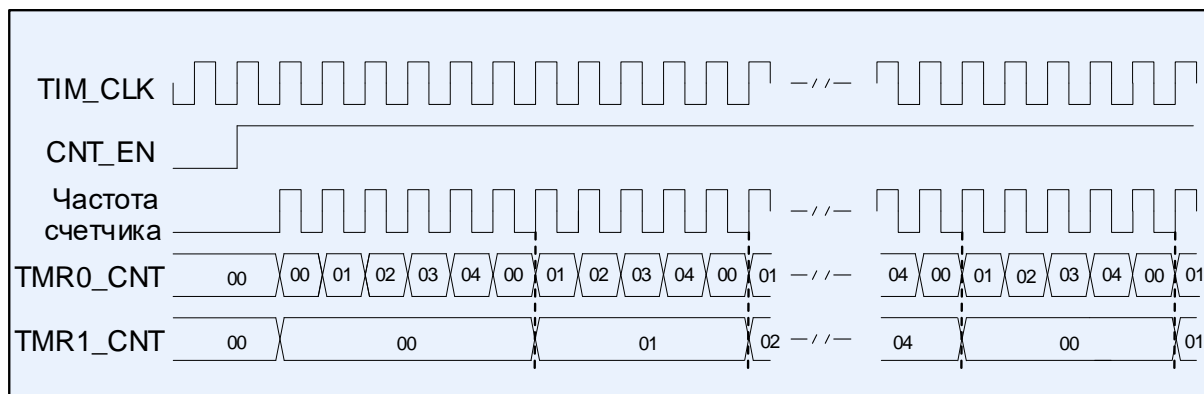


Рисунок 46 – Диаграммы работы двух таймеров в каскаде  
 TIMER0: DIR = 0, EVENT\_SEL[3:0] = 0000, CNT\_MODE[1:0] = 00;  
 TIMER1: DIR = 0, EVENT\_SEL[3:0] = 0001, CNT\_MODE[1:0] = 10

### 18.3.3 Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CH1i

Данный режим выбирается, когда  $EVENT\_SEL[3:0] = 0100$  и  $CNT\_MODE[1:0] = 10$ . Основной счетчик таймера считает по переднему фронту внешнего сигнала, поступающего на вход канала CH1i. Биты CHSEL[1:0] регистра CH1\_CNTRL не оказывают влияния, так как они применяются для работы канала таймера только в режиме захвата.

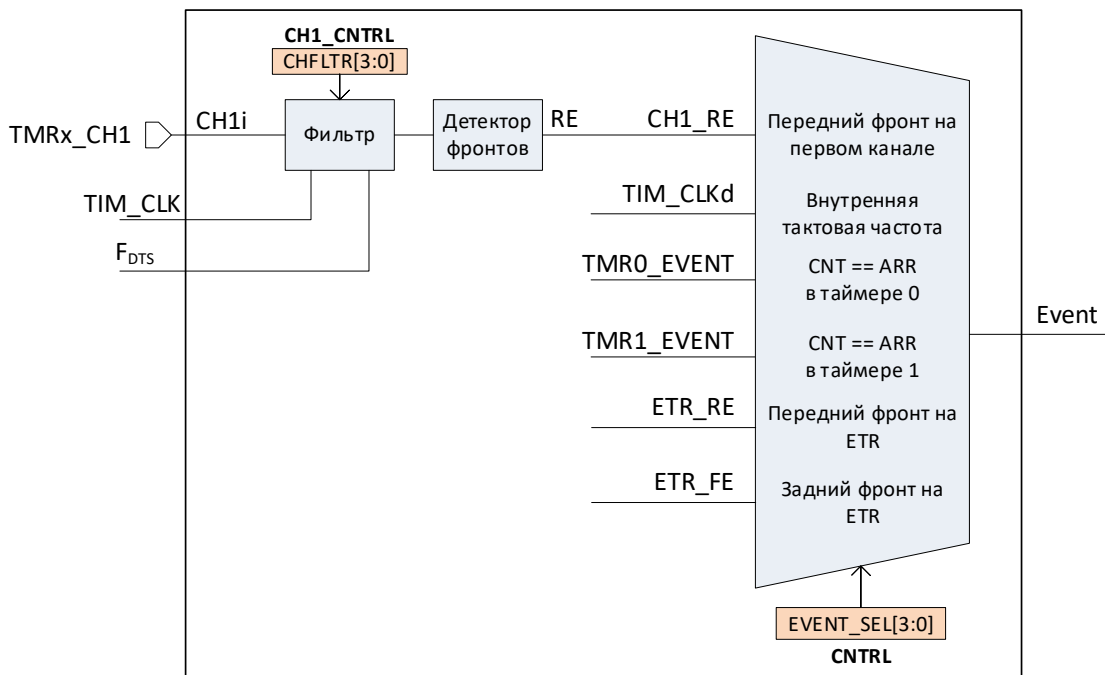


Рисунок 47 – Схема тактирования сигналом со входа первого канала

Со входа CH1i внешний тактовый сигнал поступает в блок цифрового фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. подраздел 18.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле CHFLTR[3:0] регистра CH1\_CNTRL.

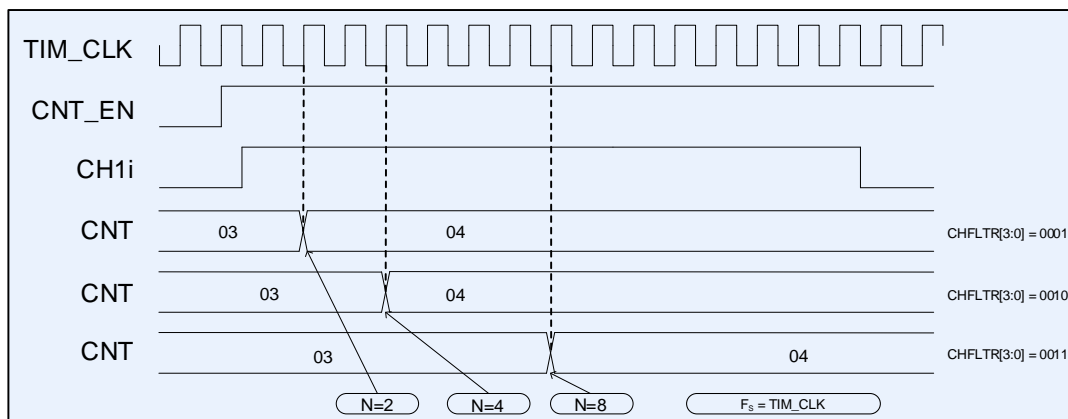


Рисунок 48 – Диаграмма внешнего тактирования с разными вариантами фильтра

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

Данный режим выбирается, когда  $EVENT\_SEL[3:0] = 100x$  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  $EVENT\_SEL[3:0]$ .

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. подраздел 18.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле ETR\_FILTER[3:0].

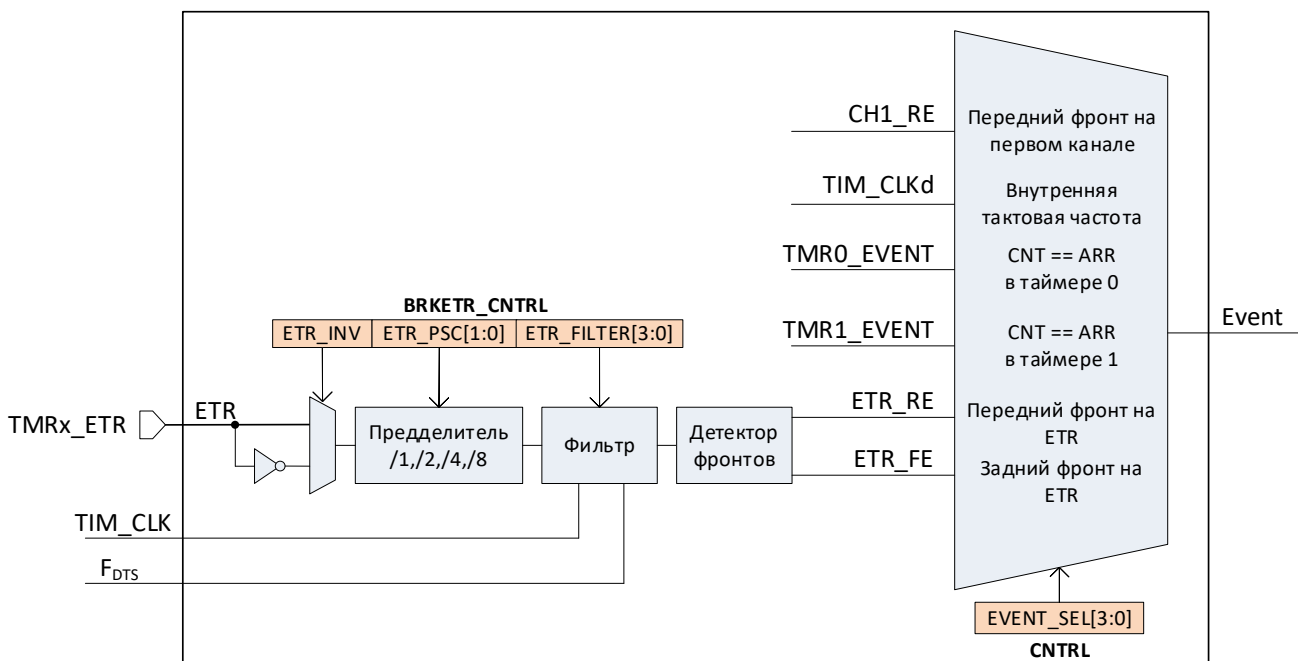


Рисунок 49 – Схема тактирования сигналом со входа ETR

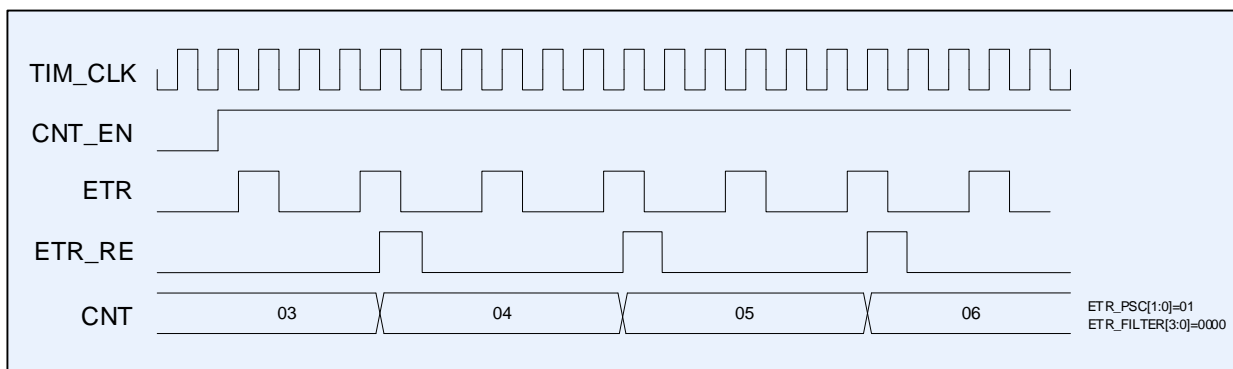


Рисунок 50 – Диаграмма тактирования со входа ETR,  $EVENT\_SEL[3:0] = 1000$

## 18.4 Режим захвата

В режиме захвата по событию от внешнего входного сигнала происходит фиксация значения основного счетчика CNT в регистры CCR1 (CCR) и CCR11 (CCR1). Регистрация событий осуществляется только по входу CH1i (вывод, настроенный в функцию TMRx\_CH1).

Структурная схема блока захвата представлена на рисунке 51.

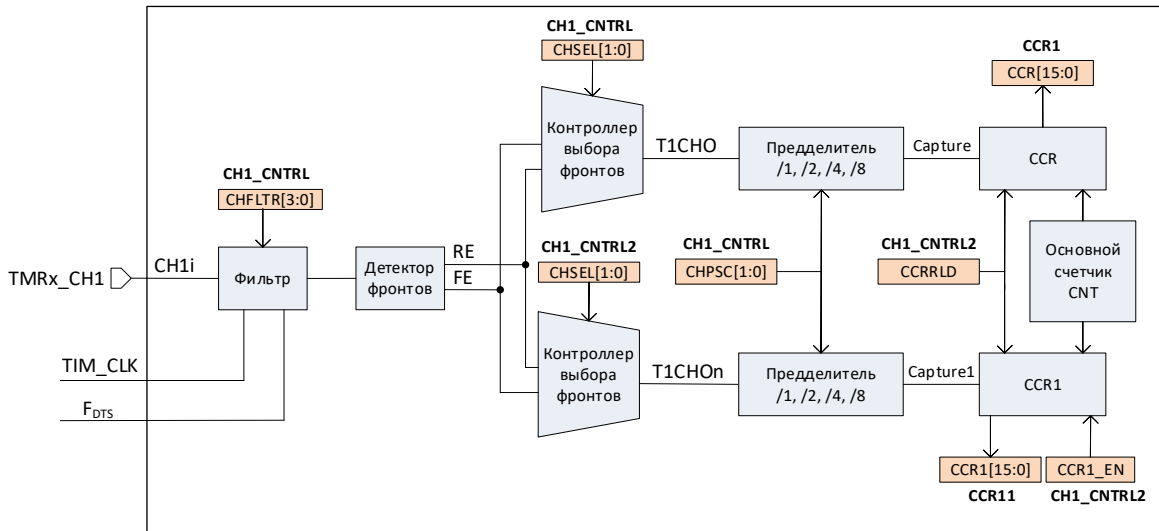


Рисунок 51 – Структурная схема блока захвата на примере канала 1

Для включения режима захвата необходимо записать «1» в бит CAP\_NPWM регистра управления каналом CH1\_CNTRL. Для использования регистра CCR11(CCR1) необходимо записать «1» в бит CCR1\_EN регистра CH1\_CNTRL2.

Внешний сигнал со входа CH1i сначала поступает в блок фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. подраздел 18.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле CHFLTR[3:0] регистра CH1\_CNTRL.

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 и отрицательным фронтом канала. Настройка блока «Контроллер выбора фронтов» для регистра CCR1 осуществляется в поле CHSEL[1:0] регистра CH1\_CNTRL, а для регистра CCR11 – в поле CHSEL[1:0] регистра CH1\_CNTRL2. Выбранный для захвата сигнал поступает в предварительный делитель, который в зависимости от значения в поле CHPSC[1:0] регистра CH1\_CNTRL позволяет фиксировать все события, либо каждое второе, каждое четвертое или каждое восьмое событие.

Предварительный делитель для регистра CCR1 формирует сигнал Capture, а предварительный делитель для регистра CCR11 формирует сигнал Capture1. По сигналам Capture и Capture1 выполняется запись текущего значения основного счетчика CNT в регистры CCR1 и CCR11.



На рисунке 52 показан пример захвата значения основного счетчика CNT в регистр CCR1 по положительному фронту на входе канала, а в регистр CCR11 – по отрицательному фронту на входе канала.

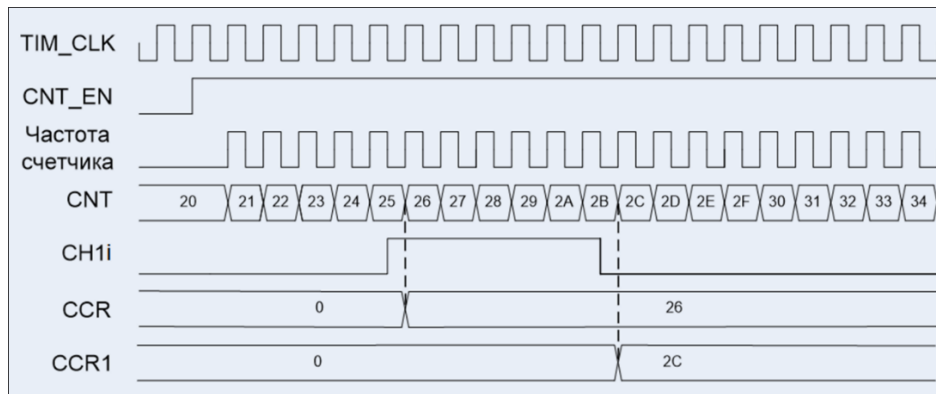


Рисунок 52 – Диаграмма захвата события со входа канала 1

По событию захвата на определенном канале в регистре IE можно разрешить выработку прерываний.

Между формированием события захвата и записью текущего значения основного счетчика CNT в регистры CCR1 и CCR11 может быть установлена задержка с помощью бита EV\_DELAY в регистре CH1\_CNTRL2. Если бит EV\_DELAY равен «0», то сначала формируется событие захвата, а затем через один такт сигнала синхронизации TIM\_CLK выполняется запись CNT в регистры CCR1 и CCR11. Если бит EV\_DELAY равен «1», то обновление информации в регистрах CCR1 и CCR11 происходит синхронно с событием захвата. При реализации чтения регистров CCR1 и CCR11 по событию захвата рекомендуется устанавливать бит EV\_DELAY в «1».

### 18.5 Режим ШИМ

В режиме ШИМ канал таймера позволяет формировать выходные сигналы с возможностью задания «мертвой зоны». Структурная схема блока формирования ШИМ представлена на рисунке 53.

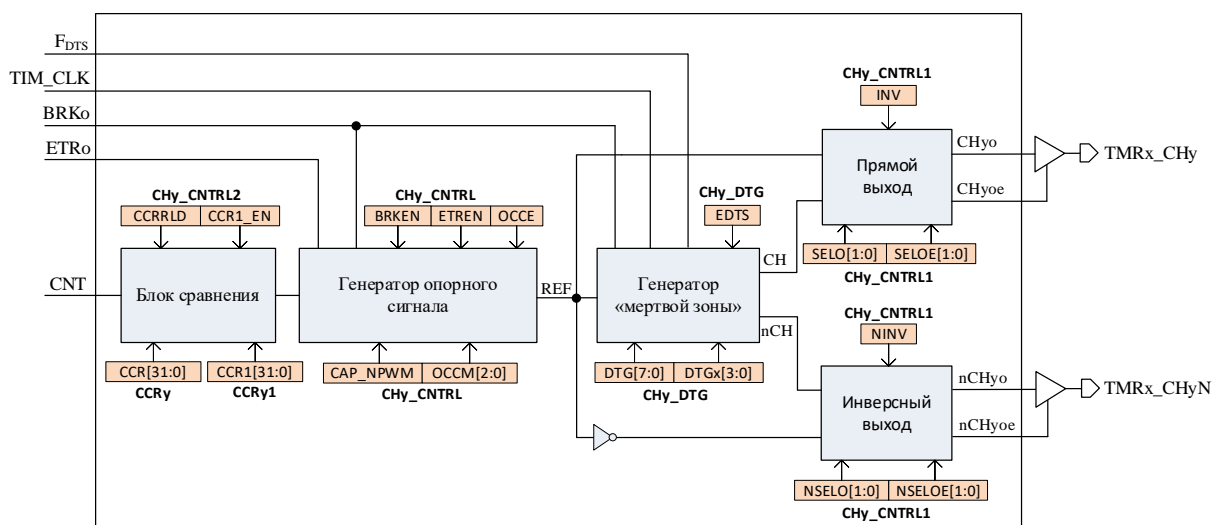


Рисунок 53 – Структурная схема блока формирования ШИМ

Для включения режима ШИМ необходимо в регистре управления каналом CH1\_CNTRL записать «0» в бит CAP\_NPWM.

### 18.5.1 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 CCR1 (CCR), CCR11 (CCR1) и основного счетчика CNT. Формат выработки сигнала REF устанавливается в регистре управления каналом таймера CH1\_CNTRL, поле OCCM[2:0].

Если в регистре CH1\_CNTRL2 бит CCR1\_EN = 0, то для формирования сигнала REF используется только результат сравнения значения в регистре CCR1 и основного счетчика CNT.

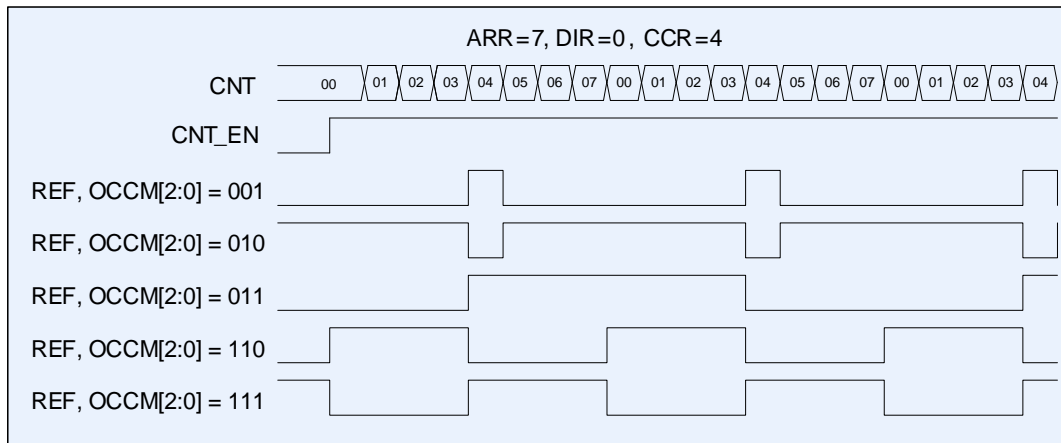


Рисунок 54 – Диаграмма работы в режиме ШИМ, CCR1\_EN=0

Если в регистре CH1\_CNTRL2 бит CCR1\_EN = 1, то для формирования сигнала REF задействуются оба результата сравнения значения в регистрах CCR1, CCR11 и основного счетчика CNT.

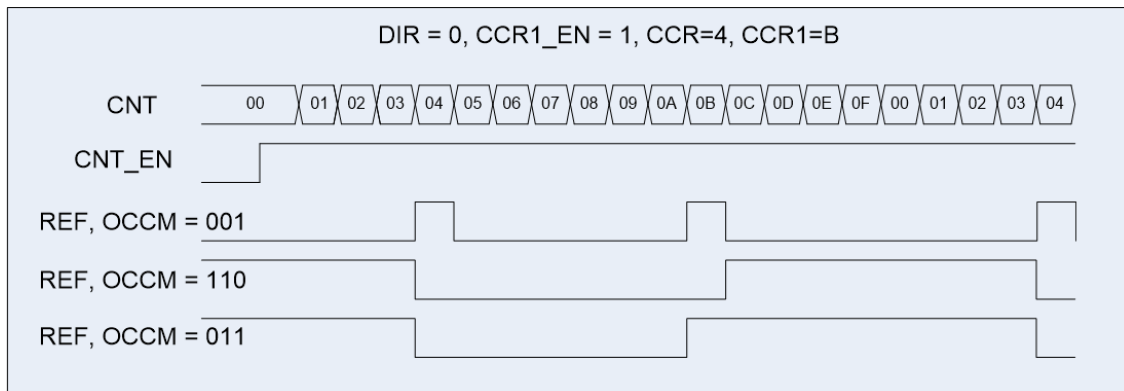


Рисунок 55 – Диаграмма работы в режиме ШИМ, CCR1\_EN = 1

Запись новых значений в регистры CCR1 и CCR11 осуществляется немедленно, если в регистре CH1\_CNTRL2 бит CCRRLD установлен в «0», иначе регистры CCR1 и CCR11 получают новые значения только при CNT == 0. Процесс обновления значений в регистрах CCR1 и CCR11 обозначается в регистре CH1\_CNTRL с помощью флагов WR\_CMPL и WR\_CMPL1, соответственно. На время выполнения записи флаг WR\_CMPL/WR\_CMPL1 устанавливается в «1», по окончании записи флаг WR\_CMPL/WR\_CMPL1 сбрасывается в «0».

Сигнал REF может быть принудительно установлен в «0» с использованием внешнего сигнала сброса, поступающего со входа ETR (высокий активный уровень) или

со входа BRK (низкий активный уровень). Активный уровень на входах ETR и BRK может быть изменен с помощью инверсии входного сигнала, регистр BRKETR\_CNTRL, биты ETR\_INV и BRK\_INV, соответственно.

Для разрешения сброса сигнала REF по входу ETR необходимо установить бит ETREN и OCCE в регистре CH1\_CNTRL. Активный уровень на входе ETR сбрасывает сигнал REF в «0». После снятия активного уровня на входе ETR сигнал REF остается в «0» до следующего события установки REF в «1», рисунок 56.

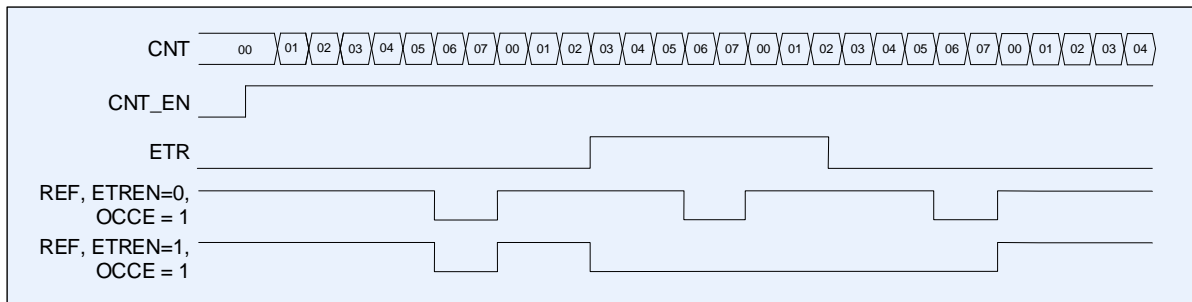


Рисунок 56 – Диаграмма сброса сигнала REF по выводу ETR

Для разрешения сброса сигнала REF по входу BRK необходимо установить бит BRKEN в регистре CH1\_CNTRL. Активный уровень на входе BRK сбрасывает сигнал REF в «0» путем маскирования. После снятия активного уровня на входе BRK генерация сигнала REF сразу же восстанавливается, рисунок 57.

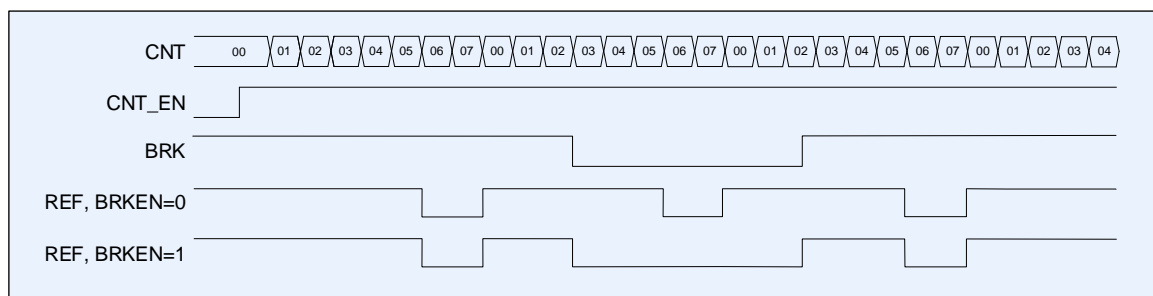


Рисунок 57 – Диаграмма сброса сигнала REF по выводу BRK

### 18.5.2 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 (CH1o, CH1oe) представляет собой инвертированный сигнал REF, в котором передний фронт задержан на величину DTGdel относительно заднего фронта опорного сигнала REF;
- сигнал на инверсном выходе (nCH1o, nCH1oe) представляет собой сигнал REF, в котором передний фронт задержан на величину DTGdel относительно переднего фронта опорного сигнала REF.

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

$$DTGdel = DTG \cdot (DTGx + 1), \tag{12}$$

где DTGx – предварительный делитель частоты;

DTG – основной делитель частоты.

Управление блоком DTG осуществляется через регистр CH1\_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.

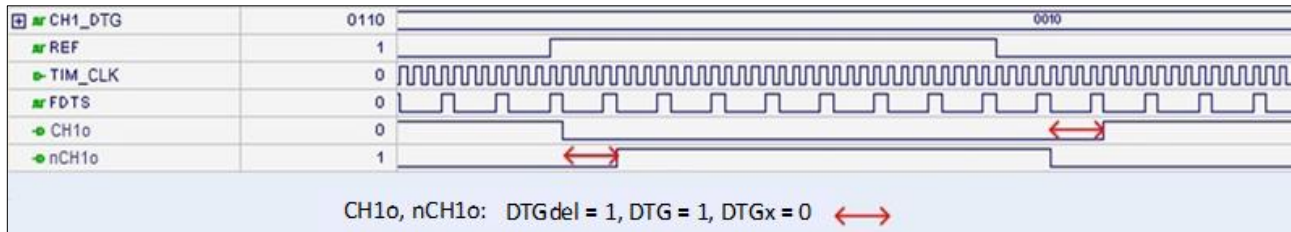


Рисунок 58 – Диаграмма работы блока DTG

Выходные сигналы блока DTG могут быть принудительно установлены в «0» с использованием внешнего сигнала сброса, поступающего со входа BRK (низкий активный уровень). Активный уровень на входе BRK может быть изменен с помощью инверсии входного сигнала, бит BRK\_INV в регистре BRKETR\_CNTRL. Для разрешения сброса выходных сигналов блока DTG по входу BRK необходимо установить бит BRKEN в регистре CH1\_CNTRL. Активный уровень на входе BRK сбрасывает выходные сигналы блока DTG в «0» путем их маскирования.

### 18.5.3 Выходные блоки

Каждый канал таймера имеет два выходных блока – прямой и инверсный. Каждый выходной блок формирует как сигнал выдачи (CH1o, nCH1o), так и сигнал разрешения выдачи (CH1oe, nCH1oe). В качестве сигналов для прямого (CH1o, CH1oe) и инверсного (nCH1o, nCH1oe) выходов в регистре CH1\_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 (CH1o) и инверсного (nCH1o) выходов задается в полях SELO[1:0] и NSELO[1:0] регистра CH1\_CNTRL1. Дополнительно каждый сигнал выдачи для прямого (CH1o) и инверсного (nCH1o) выходов может быть инвертирован путем установки битов INV и NINV в регистре CH1\_CNTRL1. Выбор источника сигнала разрешения выдачи для прямого (CH1oe) и инверсного (nCH1oe) выходов задается в полях SELOE[1:0] и NSELOE[1:0] регистра CH1\_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.

## 18.6 Блок цифрового фильтра

В тракте входа ETR и входа канала таймера CH1i 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\_FILTER[3:0] регистра BRKETR\_CNTRL, для входа канала CH*i* – в поле CHFLTR[3:0] регистра CH1\_CNTRL. Значение в данных полях позволяет настроить два параметра фильтра:

- частота выборки  $F_S$ , на которой входной сигнал захватывается в сдвиговый регистр для накопления. В качестве частоты  $F_S$  может использоваться частота TIM\_CLK или  $F_{DTS}$ ;
- количество выборок (длина фильтра)  $N$ , на протяжении которых входной сигнал должен оставаться стабильным, чтобы не подвергнуться фильтрации.

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

Таким образом, настраивая частоту  $F_S$  и количество выборок  $N$ , задается минимальная длительность импульсов входного сигнала, которые не будут отфильтрованы. Диаграмма работы фильтра при использовании частоты TIM\_CLK приведена на рисунке 59, частота  $F_S = TIM\_CLK$ , количество выборок  $N = 4$ .

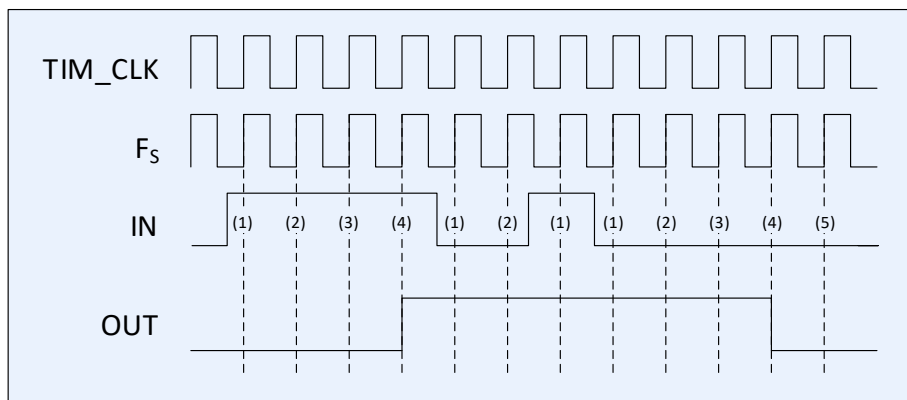


Рисунок 59 – Диаграмма работы фильтра,  $F_S = TIM\_CLK$ ,  $N = 4$

Для задания длительных интервалов накопления входного сигнала имеется возможность использовать частоту  $F_{DTS}$ , которая формируется из частоты TIM\_CLK путем прореживания на заданный коэффициент (см. пункт 18.2.4 «Тактовая частота  $F_{DTS}$ »).

Диаграмма работы фильтра при использовании частоты  $F_{DTS}$  приведена на рисунке 60, частота выборки  $F_S = F_{DTS}/2$ , количество выборок  $N = 6$ .

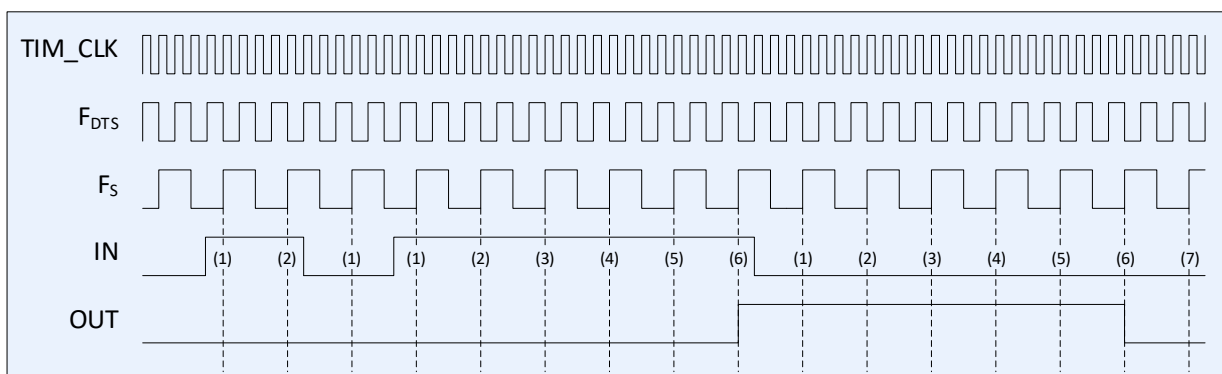


Рисунок 60 – Диаграмма работы фильтра,  $F_S = F_{DTS}/2$ ,  $N = 6$

Возможные варианты настройки блока фильтра приведены в таблице 167.

Таблица 167 – Возможные конфигурации фильтра

CHFLTR[3:0], ETR_FILTER[3:0]	Частота выборки Fs	Количество выборок N	Минимальная длительность импульсов, которые не будут отфильтрованы
0000	F <sub>DTS</sub>	1	-
0001	TIM_CLK	2	2 × T <sub>TIM_CLK</sub>
0010	TIM_CLK	4	4 × T <sub>TIM_CLK</sub>
0011	TIM_CLK	8	8 × T <sub>TIM_CLK</sub>
0100	F <sub>DTS</sub> /2	6	12 × T <sub>F<sub>DTS</sub></sub>
0101	F <sub>DTS</sub> /2	8	16 × T <sub>F<sub>DTS</sub></sub>
0110	F <sub>DTS</sub> /4	6	24 × T <sub>F<sub>DTS</sub></sub>
0111	F <sub>DTS</sub> /4	8	32 × T <sub>F<sub>DTS</sub></sub>
1000	F <sub>DTS</sub> /8	6	48 × T <sub>F<sub>DTS</sub></sub>
1001	F <sub>DTS</sub> /8	8	64 × T <sub>F<sub>DTS</sub></sub>
1010	F <sub>DTS</sub> /16	5	80 × T <sub>F<sub>DTS</sub></sub>
1011	F <sub>DTS</sub> /16	6	96 × T <sub>F<sub>DTS</sub></sub>
1100	F <sub>DTS</sub> /16	8	128 × T <sub>F<sub>DTS</sub></sub>
1101	F <sub>DTS</sub> /32	5	160 × T <sub>F<sub>DTS</sub></sub>
1110	F <sub>DTS</sub> /32	6	192 × T <sub>F<sub>DTS</sub></sub>
1111	F <sub>DTS</sub> /32	8	256 × T <sub>F<sub>DTS</sub></sub>

## 18.7 Флаги состояний и прерывания

В процессе работы блок таймера отслеживает состояние внутренних блоков и формирует 8 событий:

- CNT ZERO EVENT – совпадение значения счетчика CNT с нулем;
- CNT ARR EVENT – совпадение значения счетчика CNT со значением в регистре ARR;
- ETR RE EVENT – фиксация переднего фронта на входе ETR;
- ETR FE EVENT – фиксация заднего фронта на входе ETR;
- BRK EVENT – фиксация высокого уровня на входе BRK;
- CCR CAP EVENT – запись значения счетчика CNT в регистр CCR1 по захвату настроенного фронта на входе канала CH<sub>i</sub>;
- CCR REF EVENT – фиксация переднего фронта на выходе генератора опорного сигнала REF;
- CCR CAP1 EVENT – запись значения счетчика CNT в регистр CCR11 по захвату настроенного фронта на входе канала CH<sub>i</sub>.

### 18.7.1 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.

### 18.7.2 Прерывания

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

### 18.8 Примеры

В данном разделе приведены примеры инициализации таймера 0 в различных режимах работы. Для других таймеров инициализация выполняется аналогично.

#### 18.8.1 Обычный счетчик

```
RST_CLK->PER2_CLOCK |= 0x4000; //Разрешение частоты PCLK для таймера 0
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
//для таймера 0 (TIM_CLK = TIM0_C2)

TIMER0->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER0->CNT = 0x00000000; //Начальное значение счетчика
TIMER0->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMER0->ARR = 0x0000000F; //Основание счета
TIMER0->IE = 0x00000002; //Разрешение генерировать прерывание при CNT = ARR
//Разрешение работы таймера
TIMER0->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
```

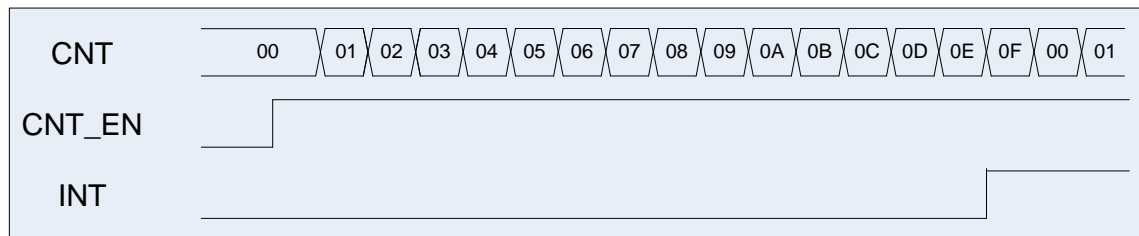


Рисунок 61 – Режим обычного счетчика

#### 18.8.2 Режим захвата

```
RST_CLK->PER2_CLOCK |= 0x4000; //Разрешение частоты PCLK для таймера 0
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
//для таймера 0 (TIM_CLK = TIM0_C2)

TIMER0->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER0->CNT = 0x00000000; //Начальное значение счетчика
TIMER0->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMER0->ARR = 0x000000FF; //Основание счета
TIMER0->IE = 0x00000020; //Разрешение генерировать прерывание по событию
//настроенного фронта на входе CH1i
TIMER0->CH1_CNTRL = 0x00008000; //Захват по положительному фронту сигнала
//на входе CH1i, фильтрация отключена
```



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

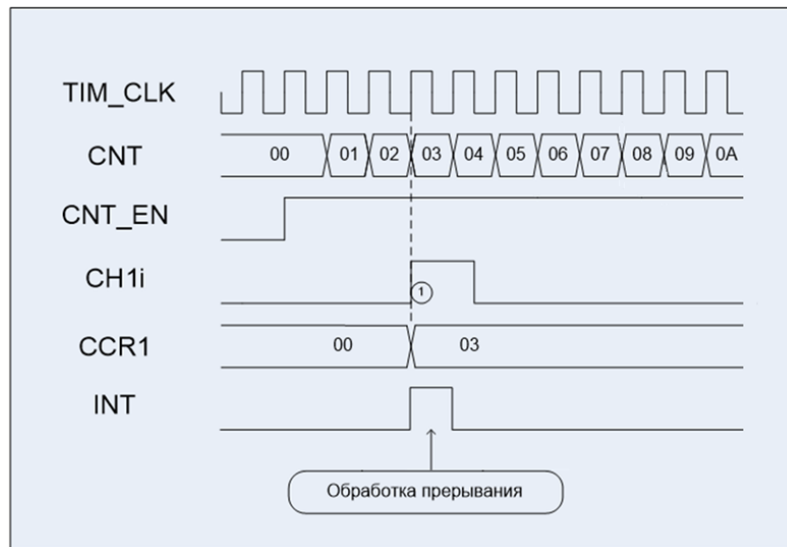


Рисунок 62 – Диаграммы примера работы в режиме захвата (захват по положительному фронту)

```
RST_CLK->PER2_CLOCK |= 0x4000; //Разрешение частоты PCLK для таймера 0
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
//для таймера 0 (TIM_CLK = TIM0_C2)

TIMERO->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMERO->CNT = 0x00000000; //Начальное значение счетчика
TIMERO->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMERO->ARR = 0x000000FF; //Основание счета
TIMERO->IE = 0x00000020; //Разрешение генерировать прерывание по событию
//настроенного фронта на входе CH1i

TIMERO->CH1_CNTRL = 0x00008011; //Захват по отрицательному фронту сигнала
//на входе CH1i, фильтрация выполняется
//по 2 выборкам на частоте TIM_CLK

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



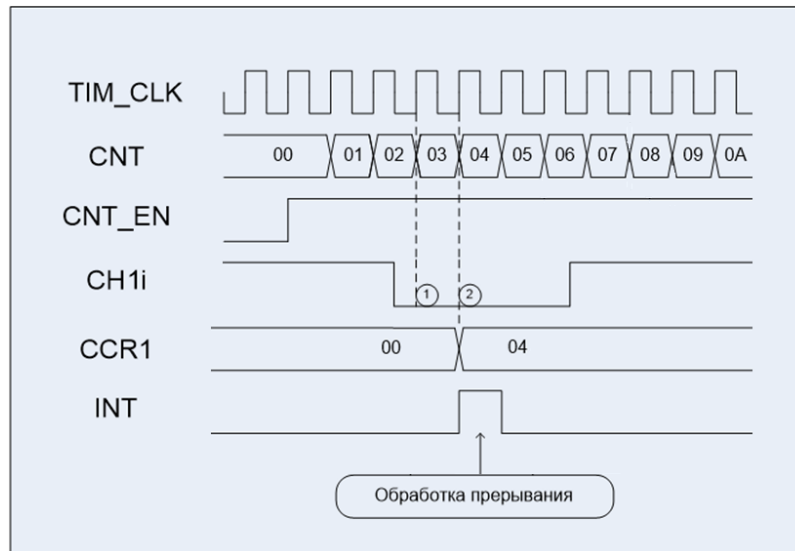


Рисунок 63 – Диаграммы примера работы в режиме захвата (захват по отрицательному фронту)

### 18.8.3 Режим ШИМ

```

RST_CLK->PER2_CLOCK /= 0x4000; //Разрешение частоты PCLK для таймера 0
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
//для таймера 0 (TIM_CLK = TIM0_C2)

TIMER0->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER0->CNT = 0x00000000; //Начальное значение счетчика
TIMER0->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMER0->ARR = 0x00000010; //Основание счета
TIMER0->IE = 0x00000200; //Разрешение генерировать прерывание по событию
//переднего фронта на выходе REF для канала 1

//Режим работы каналов – ШИМ
TIMER0->CH1_CNTRL = 0x00000200; //REF = 1, если CNT == CCR
//Режим работы выхода канала – канал работает на выход, на выходы канала выдается
//сигнал REF
TIMER0->CH1_CNTRL1 = 0x00000909;
//Установка значений CCR, с которыми сравнивается CNT при работе в режиме ШИМ
TIMER0->CCR1 = 0x00000003;
//Разрешение работы таймера
TIMER0->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
    
```

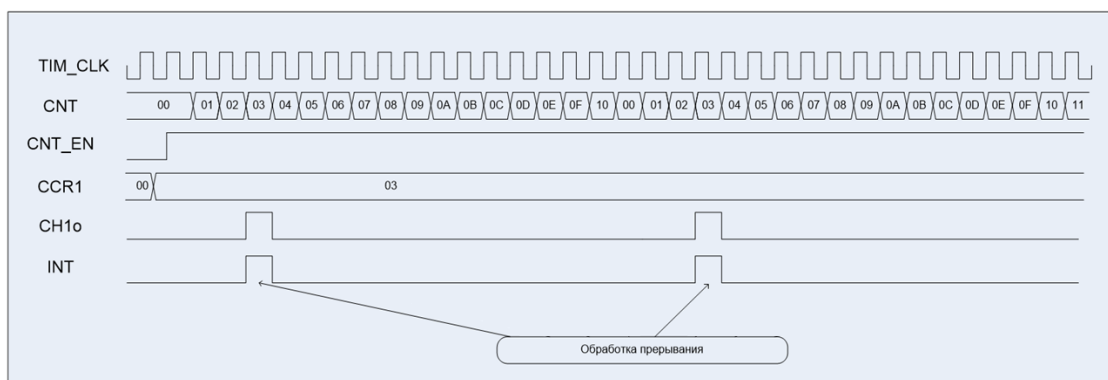


Рисунок 64 – Диаграммы примера работы в режиме ШИМ (OCM[2:0] = 001)

```

RST_CLK->PER2_CLOCK |= 0x4000; //Разрешение частоты PCLK для таймера 0
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
//для таймера 0 (TIM_CLK = TIM0_C2)

TIMER0->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER0->CNT = 0x00000000; //Начальное значение счетчика
TIMER0->PSG = 0x00000000; //Предделитель частоты TIM_CLK
TIMER0->ARR = 0x00000010; //Основание счета
TIMER0->IE = 0x00000200; //Разрешение генерировать прерывание по событию
//переднего фронта на выходе REF для канала 1

//Режим работы каналов – ШИМ
TIMER0->CH1_CNTRL = 0x00000600; //Переключение, если CNT == CCR
//Режим работы выхода канала – канал работает на выход, на выходы канала выдается
сигнал REF
TIMER0->CH1_CNTRL1 = 0x00000909;
//Установка значений CCR, с которыми сравнивается CNT при работе в режиме ШИМ
TIMER0->CCR1 = 0x0000000F;
//Разрешение работы таймера
TIMER0->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
    
```

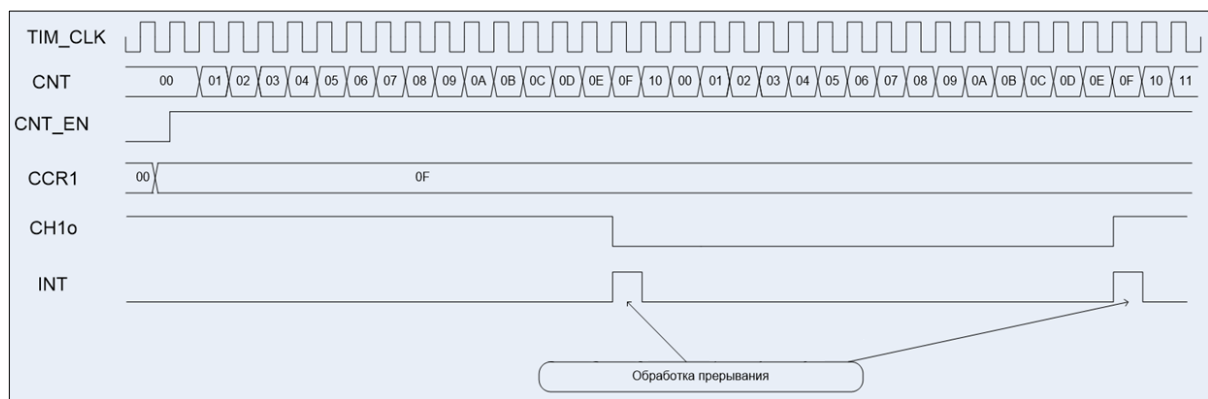


Рисунок 65 – Диаграммы примера работы в режиме ШИМ (ОССМ[2:0] = 011)

## 18.9 Описание регистров блока таймера

Таблица 168 – Базовые адреса и смещения регистров управления таймера

Адрес	Название	Описание
0x4007_0000	TIMER0	Контроллер TIMER0
0x4007_8000	TIMER1	Контроллер TIMER1
Смещение		
0x00	CNT[15:0]	Основной счетчик таймера
0x04	PSG[15:0]	Делитель частоты TIM_CLK для тактирования основного счетчика
0x08	ARR[15:0]	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

Адрес	Название	Описание
Канал 1		
0x10	CCR1[15:0]	Регистр сравнения/захвата для 1 канала таймера
0x20	CH1_CNTRL[31:0]	Регистр управления 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	CCR11[15:0]	Регистр сравнения/захвата 1 для 1 канала таймера

### 18.9.1 CNT

Таблица 169 – Основной счетчик таймера CNT

Номер	15...0
Доступ	R/W
Сброс	0
	CNT[15:0]

Таблица 170 – Описание бит регистра CNT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...0	CNT[15:0]	Значение основного счетчика таймера

### 18.9.2 PSG

Таблица 171 – Делитель частоты TIM\_CLK для счета основного счетчика PSG

Номер	15..0
Доступ	R/W
Сброс	0
	PSG[15:0]

Таблица 172 – Описание бит регистра PSG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...0	PSG[15:0]	Значение предварительного делителя счетчика. Основной счетчик считает на частоте: $TIM\_CLKd = TIM\_CLK/(PSG+1)$

### 18.9.3 ARR

Таблица 173 – Основание счета основного счетчика ARR

Номер	15...0
Доступ	R/W
Сброс	0
	ARR[15:0]

Таблица 174 – Описание бит регистра ARR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...0	ARR[15:0]	Основание счета для основного счетчика. CNT = [0...ARR]

### 18.9.4 CNTRL

Таблица 175 – Регистр управления основным счетчиком CNTRL

Номер	31...12	11...8
Доступ	U	R/W
Сброс	0	0
	-	EVENT_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 176 – Описание бит регистра CNTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...12	-	Зарезервировано
11...8	EVENT_SEL[3:0]	Биты выбора источника событий: 0000 – внутренняя тактовая частота TIM_CLKd (формируется путем деления частоты TIM_CLK); 0001 – CNT == ARR в таймере 0; 0010 – CNT == ARR в таймере 1; 0011 – зарезервировано; 0100 – событие переднего фронта на канале 1, «Режим 1»; 0101-0111 – зарезервировано; 1000 – событие переднего фронта на ETR, «Режим 2»; 1001 – событие заднего фронта на ETR, «Режим 2»; 1010-1111 – 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 EVENT_SEL[3:0]: – EVENT_SEL[3:0] = 0000: CNT_MODE[1:0] = 00 или 01; – EVENT_SEL[3:0] != 0000: CNT_MODE[1:0] = 10

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
5..4	FDTS[1:0]	Делитель тактовой частоты $F_{DTS}$ : 00 – $F_{DTS} = TIM\_CLK$ ; 01 – $F_{DTS} = TIM\_CLK/2$ ; 10 – $F_{DTS} = TIM\_CLK/3$ ; 11 – $F_{DTS} = TIM\_CLK/4$
3	DIR	Направление счета основного счетчика: 0 – прямой, от 0 до ARR; 1 – обратный, от ARR до 0
2	WR_CMPL	Флаг выполнения записи нового значения в регистры CNT, PSG и ARR: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
1	ARRB_EN	Режим обновления регистра ARR: 0 – ARR будет перезаписан в момент записи в ARR; 1 – ARR будет перезаписан при $CNT == ARR$
0	CNT_EN	Разрешение работы таймера: 0 – таймер отключен; 1 – таймер включен

### 18.9.5 CCR1

Таблица 177 – Регистр сравнения/захвата для 1 канала таймера CCR1

Номер	15...0
Доступ	R/W
Сброс	0
	CCR[15:0]

Таблица 178 – Описание бит регистра CCR1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...0	CCR[15:0]	Режим захвата: значение CNT, при котором произошел факт захвата события. Режим ШИМ: значение CCR, с которым сравнивается CNT

### 18.9.6 CCR11

Таблица 179 – Регистр сравнения/захвата для 1 канала таймера CCR11

Номер	15...0
Доступ	R/W
Сброс	0
	CCR1[15:0]

Таблица 180 – Описание бит регистра CCR11

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...0	CCR1[15:0]	Режим захвата: значение CNT, при котором произошел факт захвата события. Режим ШИМ: значение CCR1, с которым сравнивается CNT

### 18.9.7 CH1\_CNTRL

Таблица 181 – Регистр управления для 1 канала таймера CH1\_CNTRL

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	ETREN

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
	BRKEN	OCCM[2:0]	OCCE	CHPSC[1:0]	CHSEL[1:0]	CHFLTR[3:0]

Таблица 182 – Описание бит регистра CH1\_CNTRL

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 CCR11: 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 CCR1: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
13	ETREN	Разрешение сброса сигнала REF в «0» при высоком уровне на входе ETR: 0 – запрещен; 1 – разрешен
12	BRKEN	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;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		010 – 0, если CNT==CCR; 011 – переключение REF, если CNT==CCR; 100 – всегда 0; 101 – всегда 1; 110 – 1, если DIR=0 (счет прямой), CNT<CCR, иначе 0; 0, если DIR=1 (счет обратный), CNT>CCR, иначе 1; 111 – 0, если DIR=0 (счет прямой), CNT<CCR, иначе 1; 1, если DIR=1 (счет обратный), CNT>CCR, иначе 0. Если CCR1_EN=1: 000 – всегда 0; 001 – 1, если CNT==CCR или CNT==CCR1; 010 – 0, если CNT==CCR или CNT==CCR1; 011 – переключение REF, если CNT==CCR или CNT==CCR1; 100 – всегда 0; 101 – всегда 1; 110 – 0, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 1; 0, если DIR=1 (счет обратный), CCR<CNT<CCR1, иначе 1; 111 – 1, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 0; 1, если DIR=1 (счет обратный), CCR<CNT <CCR1, иначе 0; Необходимо соблюдать условие CCR<CCR1
8	OCCE	Разрешение работы ETR: 0 – запрещен; 1 – разрешен
7...6	CHPSC[1:0]	Предварительный делитель входного канала: 00 – нет деления; 01 – /2; 10 – /4; 11 – /8
5...4	CHSEL[1:0]	Выбор события по входному каналу CHi для фиксации значения основного счетчика (регистр CNT) в регистр CCR1: 00 – положительный фронт на входном канале CHi; 01 – отрицательный фронт на входном канале CHi; 10-11 – зарезервировано
3...0	CHFLTR[3:0]	Конфигурация фильтра на входе канала 1. Выбор частоты выборки Fs и количества выборок N: 0000 – нет фильтрации, Fs = F <sub>DTS</sub> ; 0001 – Fs = TIM_CLK, N = 2; 0010 – Fs = TIM_CLK, N = 4; 0011 – Fs = TIM_CLK, N = 8; 0100 – Fs = F <sub>DTS</sub> /2, N = 6; 0101 – Fs = F <sub>DTS</sub> /2, N = 8; 0110 – Fs = F <sub>DTS</sub> /4, N = 6; 0111 – Fs = F <sub>DTS</sub> /4, N = 8; 1000 – Fs = F <sub>DTS</sub> /8, N = 6; 1001 – Fs = F <sub>DTS</sub> /8, N = 8;

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

### 18.9.8 CH1\_CNTRL1

Таблица 183 – Регистр управления 1 для 1 канала таймера CH1\_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 184 – Описание бит регистра CH1\_CNTRL1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...13	-	Зарезервировано
12	NINV	Инверсия инверсного выхода nCH1: 0 – выход не инвертируется; 1 – выход инвертируется
11...10	NSELO[1:0]	Выбор источника сигнала для инверсного выхода nCH1: 00 – на nCH1o выдается 0; 01 – на nCH1o выдается 1; 10 – на nCH1o выдается сигнал nREF; 11 – на nCH1o выдается сигнал с DTG
9...8	NSELOE[1:0]	Режим работы инверсного выхода nCH1: 00 – на nCH1oe выдается 0; 01 – на nCH1oe выдается 1; 10 – на nCH1oe выдается сигнал nREF; 11 – на nCH1oe выдается сигнал с DTG. При nCH1oe = 0 вывод канала в третьем состоянии, при nCH1oe = 1 вывод канала работает в режиме выхода
7...5	-	Зарезервировано
4	INV	Инверсия прямого выхода CH1: 0 – выход не инвертируется; 1 – выход инвертируется
3...2	SELO[1:0]	Выбор источника сигнала для прямого выхода CH1: 00 – на CH1o выдается 0; 01 – на CH1o выдается 1; 10 – на CH1o выдается сигнал REF; 11 – на CH1o 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 CH1: 00 – на CH1oe выдается 0; 01 – на CH1oe выдается 1; 10 – на CH1oe выдается сигнал REF; 11 – на CH1oe выдается сигнал с DTG. При CH1oe = 0 вывод канала работает в режиме входа, при CH1oe = 1 вывод канала работает в режиме выхода

### 18.9.9 CH1\_CNTRL2

Таблица 185 – Регистр управления 2 для 1 канала таймера CH1\_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
	-	EV_DELAY	CCRRLD	CCR1_EN	CHSEL[1:0]

Таблица 186 – Описание бит регистра CH1\_CNTRL2

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...5	-	Зарезервировано
4	EV_DELAY	Задержка события захвата до обновления регистров CCR1 и CCR11: 0 – сигнал события захвата устанавливается в момент обнаружения события, при этом обновление регистров CCR1 и CCR11 выполняется через один такт TIM_CLK; 1 – сигнал события захвата устанавливается синхронно с обновлением информации в регистрах CCR1 и CCR11
3	CCRRLD	Режим обновления регистров CCR1 и CCR11: 0 – обновление возможно в любой момент времени; 1 – обновление будет осуществлено только при CNT == 0
2	CCR1_EN	Разрешение работы регистра CCR11: 0 – CCR11 не используется; 1 – CCR11 используется
1...0	CHSEL[1:0]	Выбор события по входному каналу CH1i для фиксации значения основного счетчика (регистр CNT) в регистр CCR11: 00 – положительный фронт на входном канале CH1i; 01 – отрицательный фронт на входном канале CH1i; 10-11 – зарезервировано

### 18.9.10 CH1\_DTG

Таблица 187 – Регистр управления генератором «мертвой зоны» CH1\_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 188 – Описание бит регистра CH1\_DTG

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

### 18.9.11 BRKETR\_CNTRL

Таблица 189 – 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_FILTER[3:0]	ETR_PSC[1:0]	ETR_INV	BRK_INV

Таблица 190 – 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_FILTER[3:0]	Конфигурация фильтра на входе ETR. Выбор частоты выборки F <sub>s</sub> и количества выборок N: 0000 – нет фильтрации, F <sub>s</sub> = F <sub>DTS</sub> ; 0001 – F <sub>s</sub> = TIM_CLK, N = 2; 0010 – F <sub>s</sub> = TIM_CLK, N = 4; 0011 – F <sub>s</sub> = TIM_CLK, N = 8; 0100 – F <sub>s</sub> = F <sub>DTS</sub> /2, N = 6; 0101 – F <sub>s</sub> = F <sub>DTS</sub> /2, N = 8; 0110 – F <sub>s</sub> = F <sub>DTS</sub> /4, N = 6; 0111 – F <sub>s</sub> = F <sub>DTS</sub> /4, N = 8; 1000 – F <sub>s</sub> = F <sub>DTS</sub> /8, N = 6; 1001 – F <sub>s</sub> = F <sub>DTS</sub> /8, N = 8;

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

### 18.9.12 STATUS

Таблица 191 – Регистр статуса таймера STATUS

Номер	31...14	13	12..10	9	8..6	5
Доступ	U	R/W	U	R/W	U	R/W
Сброс	0	0	0	0	0	0
	-	CCR CAP1 EVENT	-	CCR REF EVENT	-	CCR CAP EVENT

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

Таблица 192 – Описание бит регистра STATUS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14	-	Зарезервировано.
13	CCR CAP1 EVENT	Событие записи значения счетчика CNT в регистр CCR11 по захвату настроенного фронта на входе канала CH1i: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
12...10	-	Зарезервировано

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
9	CCR REF EVENT	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
8...6	-	Зарезервировано
5	CCR CAP EVENT	Событие записи значения счетчика CNT в регистр CCR1 по захвату настроенного фронта на входе канала CH1i: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
4	BRK EVENT	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 EVENT	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 EVENT	Событие переднего фронта на входе ETR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
1	CNT ARR EVENT	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 регистра CNT и ARR не изменили состояния, то флаг повторно не взводится
0	CNT ZERO EVENT	Событие совпадения CNT с нулем: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса флага регистра CNT не изменил состояния, то флаг повторно не взводится

18.9.13 IE

Таблица 193 – Регистр разрешения прерываний таймера IE

Номер	31...14	13	12..10	9	8..6	5
Доступ	U	R/W	U	R/W	U	R/W
Сброс	0	0	0	0	0	0
	-	CCR CAP1 EVENT IE	-	CCR REF EVENT IE	-	CCR CAP EVENT IE

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

Таблица 194 – Описание бит регистра IE

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14	-	Зарезервировано
13	CCR CAP1 EVENT IE	Флаг разрешения прерывания по событию CCR CAP1 EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
12...10	-	Зарезервировано
9	CCR REF EVENT IE	Флаг разрешения прерывания по событию CCR REF EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
8...6	-	Зарезервировано
5	CCR CAP EVENT IE	Флаг разрешения прерывания по событию CCR CAP EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
4	BRK EVENT IE	Флаг разрешения прерывания по событию BRK EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
3	ETR FE EVENT IE	Флаг разрешения прерывания по событию ETR FE EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
2	ETR RE EVENT IE	Флаг разрешения прерывания по событию ETR RE EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1	CNT ARR EVENT IE	Флаг разрешения прерывания по событию CNT ARR EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
0	CNT ZERO EVENT IE	Флаг разрешения прерывания по событию CNT ZERO EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено

## 19 Контроллер АЦП

В микросхеме реализован 12-разрядный АЦП. С помощью данного АЦП можно оцифровать сигнал с восьми внешних аналоговых выводов портов В и С и двух внутренних каналов, на которые выводится датчик температуры и источник опорного напряжения. Скорость выборки составляет до 500 000 преобразований в секунду.

Контроллер АЦП позволяет:

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

Для осуществления преобразования требуется 28 тактов синхронизации C\_ADC. В качестве синхросигнала может выступать поделенная частота периферийного блока PCLKd или частота ADC\_CLK, формируемая в блоке «Сигналы тактовой частоты». Выбор частоты осуществляется с помощью бита Cfg\_REG\_CLKS. В контроллере АЦП частота может быть поделена с помощью битов Cfg\_REG\_DIVCLK[3:0]. Максимальная частота CLK не может превышать 14 МГц.

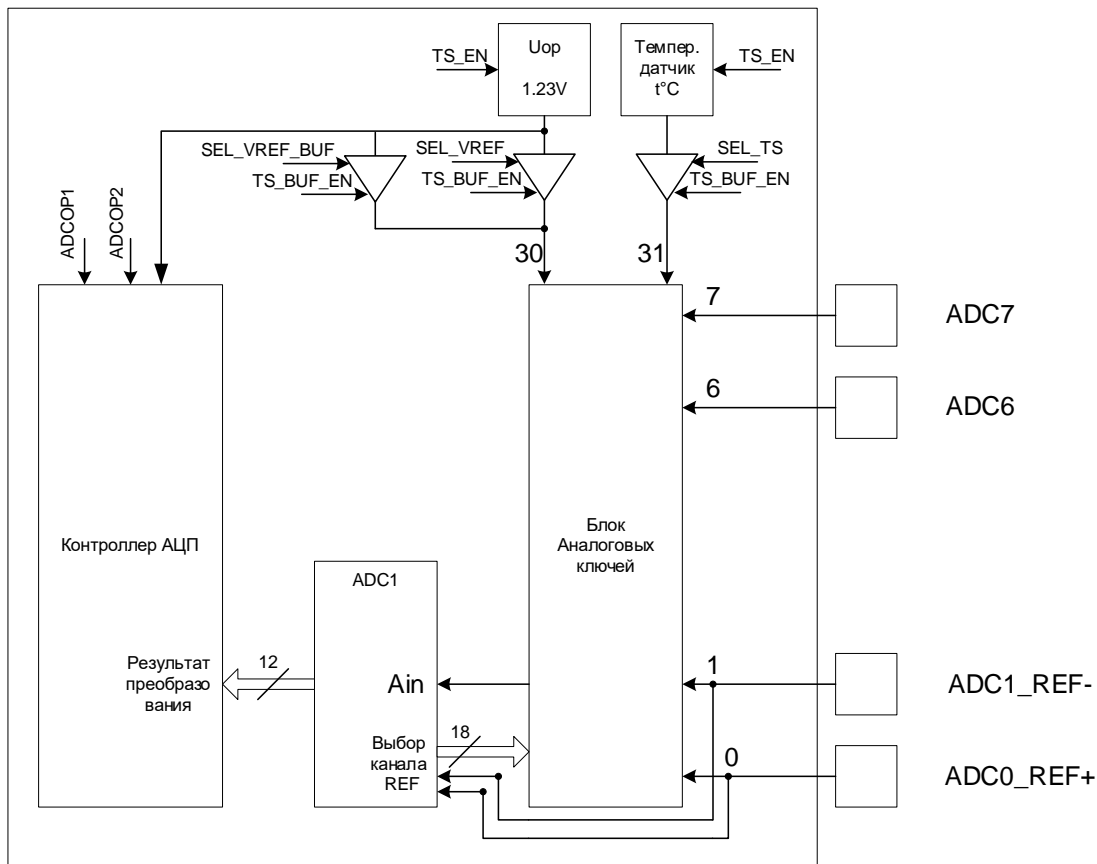


Рисунок бб – Структурная схема контроллера АЦП

Для включения АЦП необходимо установить бит Cfg\_REG\_ADON. Для снижения тока потребления вместо собственного источника опорного напряжения в АЦП может использоваться источник датчика температуры. Для этого необходимо включить блок

датчика температуры и источник опорного напряжения, установив бит TS\_EN (таблица 198) в «1». После включения можно использовать источник опорного напряжения для АЦП вместо его собственного. Для этого необходимо установить биты ADC1\_OP в «1». Для преобразования необходимо, чтобы выводы, используемые АЦП в порте C, были сконфигурированы как аналоговые и были отключены какие-либо внутренние подтяжки.

### 19.1 Преобразование внешнего канала

В регистре ADC1\_CFG в битах Cfg\_REG\_CHS[4:0] необходимо задать соответствующий выводу номер канала. Преобразование может осуществляться при внутренней опоре бит Cfg\_M\_REF = 0 и внешней Cfg\_M\_REF = 1, в этом случае опора берется с выводов ADC0\_REF+ и ADC1\_REF-. Биты Cfg\_REG\_CHCH, Cfg\_REG\_RNGC, Cfg\_REG\_SAMPLE, TS\_BUF\_EN, SEL\_VREF, SEL\_VREF\_BUF, SEL\_TS и Cfg\_Sync\_Conver 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 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.

### 19.2 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. Преобразование может осуществляться при внутренней опоре бит Cfg\_M\_REF = 0 и внешней Cfg\_M\_REF = 1, в этом случае опора берется с выводов ADC0\_REF+ и ADC1\_REF-. Биты Cfg\_REG\_RNGC, TS\_BUF\_EN, SEL\_VREF, SEL\_VREF\_BUF, SEL\_TS и Cfg\_Sync\_Conver должны быть сброшены, а Cfg\_REG\_SAMPLE и Cfg\_REG\_CHCH должны быть установлены. С помощью битов Delay\_GO можно задать паузу между преобразованиями при переборе каналов. Для начала 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[4:0] и сбросить бит Cfg\_REG\_CHCH в «0». В этом случае процесс последовательного преобразования будет выполняться только для данного канала. Последовательное преобразование значения датчика температуры и источника опорного напряжения могут выполняться только в режиме последовательного преобразования одного канала.

### 19.3 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 если результат преобразования выходит за границы выставляется флаг Flg\_REG\_AWOIFEN. А в регистре результата будет полученное значение.

### 19.4 Датчик опорного напряжения

С помощью АЦП можно осуществить преобразования источника опорного напряжения. Для этого необходимо включить блок датчика температуры и источник опорного напряжения установив бит TS\_EN в 1. После включения можно использовать источник опорного напряжения для АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1\_OP в единицу. Для выбора источника опорного напряжения в качестве источника для преобразования необходимо в битах Cfg\_REG\_CHS установить значение 30 канала. Установить бит TS\_BUF\_EN, а также бит SEL\_VREF или SEL\_VREF\_BUF. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать «1» в бит Cfg\_REG\_GO.

После завершения преобразования будет взведен бит Flg\_REG\_EOCIF в регистре ADC1\_STATUS. А в регистре ADC1\_RESULT будет результат преобразования.

После считывания результата бит Flg\_REG\_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1\_RESULT будет значение от последнего преобразования, а помимо бита Flg\_REG\_EOCIF будет взведен бит Flg\_REG\_OVERWRITE. Флаг Flg\_REG\_OVERWRITE может быть сброшен только записью в регистр ADC1\_STATUS.

Для последовательного преобразования только источника опорного напряжения можно в регистре ADC1\_CHSEL выбрать только 30 канал и установить бит Cfg\_REG\_CHCH в «1», либо установить номер 30-го канала в битах Cfg\_REG\_CHS[4:0] и сбросить бит Cfg\_REG\_CHCH в «0». В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должен быть установлен бит SEL\_VREF\_BUF или SEL\_VREF, а также бит TS\_BUF\_EN.

Примечание – При настройке преобразования сигналов с датчика температуры или источника опорного напряжения допустимо устанавливать в логическую «1» только один из битов: SEL\_VREF, SEL\_VREF\_BUF или SEL\_TS (запрещено одновременно устанавливать в «1» более одного из этих разрядов). Если необходимо в последствии проводить преобразование сигналов со внешних каналов АЦП, то предварительно необходимо сбросить в «0» биты SEL\_VREF, SEL\_VREF\_BUF и SEL\_TS регистров ADC1\_CFG, ADC1\_TRIM.

Источник опорного напряжения может быть выбран для более точного результата преобразования АЦП и не может быть использован для задания опорного напряжения преобразования.

## 19.5 Датчик температуры

С помощью первого АЦП можно осуществить преобразования датчика опорного напряжения. Для этого необходимо включить блок датчика температуры и источник опорного напряжения установив бит TS\_EN в «1». После включения можно использовать источник опорного напряжения для АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1\_OP в единицу. Для выбора датчика температуры в качестве источника для преобразования необходимо в битах Cfg\_REG\_CHS установить значение 31 канала. Установить биты TS\_BUF\_EN и SEL\_TS. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать «1» в бит Cfg\_REG\_GO.

После завершения преобразования будет взведен бит Flg\_REG\_EOCIF в регистре ADC1\_STATUS. А в регистре ADC1\_RESULT будет результат преобразования.

После считывания результата бит Flg\_REG\_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1\_RESULT будет значение от последнего преобразования, а помимо бита Flg\_REG\_EOCIF будет взведен бит Flg\_REG\_OVERWRITE. Флаг Flg\_REG\_OVERWRITE может быть сброшен только записью в регистр ADC1\_STATUS.

Для последовательного преобразования только датчика температуры можно в регистре ADC1\_CHSEL выбрать только 31 канал и установить бит Cfg\_REG\_CHCH в «1», либо установить номер 31-го канала в битах Cfg\_REG\_CHS[4:0] и сбросить бит Cfg\_REG\_CHCH в «0». В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должны быть также установлены биты TS\_BUF\_EN и SEL\_TS.

Примечание – При настройке преобразования сигналов с датчика температуры или источника опорного напряжения допустимо устанавливать в логическую «1» только один из битов: SEL\_VREF, SEL\_VREF\_BUF или SEL\_TS (запрещено одновременно устанавливать в «1» более одного из этих разрядов). Если необходимо в последствии проводить преобразование сигналов со внешних каналов АЦП, то предварительно необходимо сбросить в «0» биты SEL\_VREF, SEL\_VREF\_BUF и SEL\_TS регистров ADC1\_CFG, ADC1\_TRIM.

Параметры температурного датчика не регламентируются. В зависимости от необходимой точности может быть достаточно провести градуировку в двух – трех точках. При необходимости более точных измерений необходимо построить градуировочную таблицу. Градуировка производится индивидуально для каждой микросхемы.

### 19.6 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:

$$R_{AIN} < \frac{T_{track}}{C_{ADC} \cdot \ln(2^N)} - R_{ADC}, \tag{13}$$

$$T_{track} = 4 \cdot T_{C\_ADC} + N_{PCLKd} \cdot T_{PCLKd} = \frac{4}{f_{C\_ADC}} + \frac{(DelayGo + 1)}{f_{PCLKd}}, \tag{14}$$

где  $C_{ADC}$  – внутренняя емкость АЦП (~15-20пФ);

$N$  – требуемая точность в разрядах;

$R_{ADC}$  – входное сопротивление тракта АЦП (~500Ом);

$T_{track}$  – время заряда внутренней емкости (определяется формулой), [с];

$f_{C\_ADC}$  – рабочая частота АЦП (определяется Cfg REG CLKS в регистре ADC1\_CFG), [с<sup>-1</sup>];

$f_{PCLKd}$  – определяется формулой:

$$f_{PCLKd} = \frac{f_{PCLK}}{2^{Cfg\_REG\_DIVCLK}}. \tag{15}$$

Время зарядки внутренней емкости задается битами DelayGo[2:0].

Если необходимо обеспечить преобразование с точностью 12 разрядов ± 1/4 LSB, то  $N = 14$ . Если необходимо обеспечить преобразование с точностью 10 разрядов ± 1 LSB, то  $N=10$ . Если необходимо обеспечить преобразование с точностью 12 разрядов ± 1/4 LSB, то  $N = 14$ . Если необходимо обеспечить преобразование с точностью 10 разрядов ± 1 LSB, то  $N=10$ . Время заряда  $T_{track}$  определяется битами DelayGo[2:0] и схемой самого АЦП и представлено в таблице 195.

Таблица 195 – Время заряда внутренней емкости АЦП и время преобразования

DelayGo [2:0]	Дополнительная задержка перед началом преобразования	Общее время $T_{track}$ заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
000	1×PCLKd	4×C_ADC+1×PCLKd	28×C_ADC+1×PCLKd
001	2×PCLKd	4×C_ADC+2×PCLKd	28×C_ADC+2×PCLKd
010	3×PCLKd	4×C_ADC+3×PCLKd	28×C_ADC+3×PCLKd
011	4×PCLKd	4×C_ADC+4×PCLKd	28×C_ADC+4×PCLKd

DelayGo [2:0]	Дополнительная задержка перед началом преобразования	Общее время Ttrack заряда емкости АЦП перед началом преобразования	Общее время преобразования АЦП
100	5×PCLKd	4×C_ADC+5×PCLKd	28×C_ADC+5×PCLKd
101	6×PCLKd	4×C_ADC+6×PCLKd	28×C_ADC+6×PCLKd
110	7×PCLKd	4×C_ADC+7×PCLKd	28×C_ADC+7×PCLKd
111	8×PCLKd	4×C_ADC+8×PCLKd	28×C_ADC+8×PCLKd

Помимо точности, определяемой временем зарядки внутренней емкости АЦП, точность преобразования имеет ошибки, связанные с технологическими разбросами схемы и шумами, и определяемые параметрами E<sub>DLADC</sub>, E<sub>ILADC</sub> и E<sub>OFFADC</sub>.

Для корректного задания режимов работы АЦП в регистре ADC1\_CFG необходимо сделать до задания бита Go, иначе новая конфигурация будет действовать со следующего преобразования.

### 19.7 Описание регистров блока контроллера АЦП

Таблица 196 – Перечень регистров блока контроллера АЦП

Базовый адрес	Название	Описание
0x4004_0000	ADC	Контроллер ADC
Смещение		
0x00	ADC1_CFG	Регистр управления ADC
0x04	ADC2_CFG	Регистр управления ADC
0x08	ADC1_H_LEVEL	Регистр верхней границы ADC
0x10	ADC1_L_LEVEL	Регистр нижней границы ADC
0x18	ADC1_RESULT	Регистр результата ADC
0x20	ADC1_STATUS	Регистр статуса ADC
0x28	ADC1_CHSEL	Регистр выбора каналов перебора ADC
0x30	ADC1_TRIM	Регистр настройки термодатчика

#### 19.7.1 ADCx\_CFG

Таблица 197 – Регистр ADCx\_CFG

Номер	31...28	27...25	24...21	20	19	18
Доступ	R/W	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	Delay ADC[3:0]	Delay Go[2:0]	TR[3:0]	SEL VREF	SEL TS	TS_BUF EN

Номер	17	16	15...12	11	10	9
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	TS_EN/ ADC1 OP	Cfg Sync Conver	Cfg REG DIVCLK[3:0]	Cfg M_REF	Cfg REG RNGC	Cfg REG CHCH

Номер	8...4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	Cfg REG CHS[4:0]	Cfg REG SAMPLE	Cfg REG CLKS	Cfg REG GO	Cfg REG ADON

Таблица 198 – Описание битов регистра ADCx\_CFG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	Delay ADC [3:0]	Задержка между началом преобразования ADC1 при последовательном переборе, либо работе на один канал: 0000 – 0 тактов CLK; 0001 – 1 такт CLK; ... 1111 – 15 тактов CLK
27...25	Delay Go [2:0]	Задержка перед началом следующего преобразования после завершения предыдущего при последовательном переборе каналов: 000 – 0 тактов CLK; 001 – 1 такт CLK; ... 111 – 7 тактов CLK
24...21	-	Зарезервировано
20	SEL VREF	Выбор для оцифровки источника опорного напряжения на 1,23В: 0 – не выбран; 1 – выбран. Должен использоваться совместно с выбором канала Cfg_REG_CHS = 30
19	SEL TS	Выбор для оцифровки датчика температуры: 0 – не выбран; 1 – выбран. Должен использоваться совместно с выбором канала Cfg_REG_CHS = 31
18	TS BUF EN	<b>В регистре ADC1_CFG</b> Включение выходного усилителя для датчика температуры: 0 – выключен; 1 – включен. Используется при TS_EN = 1. Для уменьшения тока потребления
17	TS EN	<b>В регистре ADC1_CFG</b> Включение датчика температуры и источника опорного напряжения: 0 – выключен; 1 – включен.

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		При включении датчика температуры и источника опорного напряжения выходной сигнал стабилизируется в течении времени Tstb
17	ADC1 OP	<b>В регистре ADC2_CFG</b> Выбор источника опорного напряжения 1,23 В: 0 – внутренний (не точный); 1 – от датчика температуры (точный)
16	Cfg Sync Conver	Записывать всегда ноль
15..12	Cfg REG DIVCLK [3:0]	Выбор коэффициента деления частоты PCLKd периферийного блока: 0000 – PCLKd = PCLK; 0001 – PCLKd = PCLK/2; 0010 – PCLKd = PCLK/4; 0011 – PCLKd = PCLK/8; ... 1011 – PCLKd = PCLK/2048; Остальные PCLKd = PCLK
11	Cfg M_REF	Выбор источника опорных напряжений: 0 – внутренне опорное напряжение (от AUdd и AUss); 1 – внешнее опорное напряжение (от Uref+ и Uref-)
10	Cfg REG RNGC	Разрешение автоматического контролирования уровней: 1 – Разрешено, выработка прерывания при выходе за диапазон в регистрах границы обработки; 0 – не разрешено
9	Cfg REG CHCH	Выбор переключения каналов: 1 – переключение включено (перебираются каналы выбранные в регистре выбора канала); 0 – используется только выбранный канал
8...4	Cfg REG CHS [4:0]	Выбор аналогового канала, по которому поступает сигнал для преобразования: 00000 – 0 канал; 00001 – 1 канал; ... 11111 – 31 канал
3	Cfg REG SAMPLE	Выбор способа запуска АЦП: 1 – последовательное, автоматический запуск после завершения предыдущего преобразования; 0 – одиночное
2	Cfg REG CLKS	Выбор источника синхросигнала C_ADC работы ADC: 1 – ACD_CLK; 0 – PCLKd (определяется по формуле (15))

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1	Cfg REG GO	Начало преобразования. Запись «1» начинает процесс преобразования, сбрасывается автоматически
0	Cfg REG ADON	Включение АЦП: 1 – включено; 0 – выключено

### 19.7.2 ADC1\_H\_LEVEL

Таблица 199 – Регистр ADC1\_H\_LEVEL

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

Таблица 200 – Описание битов регистра ADC1\_H\_LEVEL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...12	-	Зарезервировано
11...0	REG_H_LEVEL [11:0]	Верхняя граница зоны допуска

### 19.7.3 ADC1\_L\_LEVEL

Таблица 201 – Регистр ADC1\_L\_LEVEL

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

Таблица 202 – Описание битов регистра ADC1\_L\_LEVEL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...12	-	Зарезервировано
11...0	REG_L_LEVEL [11:0]	Нижняя граница зоны допуска

### 19.7.4 ADC1\_RESULT

Таблица 203 – Регистр ADC1\_RESULT

Номер	31...21	20...16	15...12	11...0
Доступ	U	RO	U	RO
Сброс	0	0	0	0
	-	CHANNEL [11:0]	-	RESULT [11:0]

Таблица 204 – Описание битов регистра ADC1\_RESULT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...12	-	Зарезервировано
20...16	CHANNEL [11:0]	Канал результата преобразования
15...12	-	Зарезервировано
11...0	RESULT [11:0]	Значение результата преобразования

### 19.7.5 ADC1\_STATUS

Таблица 205 – 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	AWOIFIE	Flg REG EOCIF	Flg REG AWOIFEN	Flg REG OVERWRITE

Таблица 206 – Описание битов регистра ADC1\_STATUS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...5	-	Зарезервировано
4	ECOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_ECOIF: 0 – прерывания не генерируется; 1 – прерывание генерируется
3	AWOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_AWOIFEN: 0 – прерывания не генерируется; 1 – прерывание генерируется
2	Flg REG EOCIF	Флаг выставляется, когда закончено преобразования и данные еще не считаны. Очищается считыванием результата из регистра ADCx_RESULT: 1 – есть готовый результат преобразования; 0 – нет результата
1	Flg REG AWOIFEN	Флаг выставляется, когда результат преобразования выше верхней или ниже нижней границы автоматического 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, данный флаг сбрасывается только при записи в регистр флагов: 0 – не было события перезаписи не считанного результата; 1 – был результат преобразования, который не был считан



### 19.7.6 ADC1\_CHSEL

Таблица 207 – Регистр ADC1\_CHSEL

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

Таблица 208 – Описание битов регистра ADC1\_CHSEL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..0	SI_Ch_Ch_REF[31:0]	Выбор каналов автоматического перебора: 0 – в соответствующем бите канал не участвует в переборе; 1 – канал участвует в переборе

### 19.7.7 ADC1\_TRIM

Таблица 209 – Регистр ADC1\_TRIM

Номер	31...7	6	5...1	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	10000	0
	-	SEL_VREF_BUF	TS_TRIM[4:0]	0

Таблица 210 – Описание битов регистра ADC1\_TRIM

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...7	-	Зарезервировано
6	SEL_VREF_BUF	Включение выходного усилителя для источника опорного напряжения: 0 – выключен; 1 – включен. Используется при TS_EN = 1. Для уменьшения тока потребления. Должен использоваться совместно с выбором канала Cfg_REG_CHS = 30
5..1	TS_TRIM[4:0]	Подстройка опорного напряжения
0	-	Зарезервировано

## 20 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.

Интерфейс работает на трех скоростях:

- Нормальная: 100 кбит/с;
- Быстрая: 400 кбит/с;
- Очень быстрая: 3,5 Мбит/с.

Контроллер интерфейса I2C в микросхеме K1986BK214 работает только в режиме Master.

Приблизительная скорость обмена данными блоком I2C рассчитывается по формуле:

$$F_{scl} = \frac{HCLK}{(5 \cdot (DIV + 1))}. \quad (16)$$

Более точное значение скорости обмена можно установить опытным путем, значение предделителя DIV настраивается в регистрах PRL (младшая часть) и PRH (старшая часть) – если примерно рассчитанное значение предделителя DIV не задействует регистр PRH, то он должен быть равен нулю.

### 20.1 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. подразделы 20.3 «Сигнал START» и 20.6 «Сигнал STOP»).

### 20.2 Протокол I2C

Нормальная передача по интерфейсу I2C содержит четыре фазы:

- Сигнал START;
- Передача Адреса;
- Передача Данных;
- Сигнал STOP

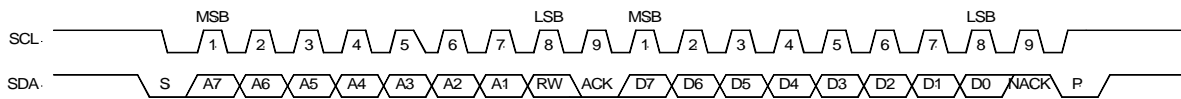


Рисунок 1 – Передача по I2C

### 20.3 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 подраздел 20.7 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 «1» в бит START регистра I2C\_CMD (см. подраздел 20.7 «Описание регистров контроллера I2C») при установленных битах RD или WR. В зависимости от состояния линии SCL генерируется либо сигнал START, либо повторный сигнал START.

### 20.4 Передача адреса

Первым байтом данных, передаваемым мастером сразу после сигнала START, является адрес ведомого. Это семибитный адрес и следующий за ним бит RW. Бит RW определяет дальнейшее направление передачи данных. В системе не может быть несколько ведомых устройств с одним адресом. Ведомое устройство, у которого совпадает адрес с адресом в сообщении, подтверждает прием, выставляя ACK и опуская линию SDA в низкий уровень на 9-й 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 (см. подраздел 20.7 «Описание регистров контроллера I2C») и установить бит WR в регистре I2C\_CMD. Контроллер осуществит передачу адреса в линию.

### 20.5 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 подраздел 20.7 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.

## 20.6 Сигнал STOP

Мастер может завершить соединение путем создания сигнала STOP. Сигнал STOP или Р бит определяется переходом линии SDA из низкого состояния в высокое, когда SCL находится в высоком состоянии.

## 20.7 Описание регистров контроллера I2C

Таблица 211 – Перечень регистров контроллера I2C

Базовый адрес	Название	Описание
0x4003_0000	I2C	Контроллер I2C
Смещение		
0x00	I2C->PRL	Младшая часть предделителя частоты
0x04	I2C->PRH	Старшая часть предделителя частоты
0x08	I2C->CTR	Управление контроллером I2C
0x0C	I2C->RXD	Принятые данные по I2C
0x10	I2C->STA	Статус I2C
0x14	I2C->TXD	Передаваемые данные по I2C
0x18	I2C->CMD	Управление I2C

### 20.7.1 I2C->PRL

Таблица 212 – Регистр PRL

Номер	7...0
Доступ	R/W
Сброс	0
	PR[7:0]

Таблица 213 – Описание битов 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 пред. делителя

### 20.7.2 I2C->PRH

Таблица 214 – Регистр PRH

Номер	7...0
Доступ	R/W
Сброс	0
	PR[15:8]

Таблица 215 – Описание битов 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 пред. делителя

### 20.7.3 I2C->CTR

Таблица 216 – Регистр CTR

Номер	7	6	5	4...0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	EN_I2C	EN_INT	S_I2C	-

Таблица 217 – Описание битов 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

### 20.7.4 I2C->RXD

Таблица 218 – Регистр RXD

Номер	7...0
Доступ	R/W
Сброс	0
	RXD[7:0]

Таблица 219 – Описание битов 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

**20.7.5 I2C->STA**

Таблица 220 – Регистр STA

Номер	7	6	5	4...2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	Rx ACK	BUSY	LOST ARB	-	TR PROG	INT

Таблица 221 – Описание битов 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: 1 – после получения состояния Start Bit; 0 – после получения Stop bit
5	LOST ARB	Потеря арбитража: 1 – потерян арбитраж; 0 – нет потери арбитража. Этот бит выставляется если: – Получен Stop bit, но он не был инициализирован этим контроллером; – Если контроллер пытается выставить SDA в высокий уровень, но SDA остается в низком
4...2	-	Зарезервировано
1	TR PROG	Процесс передачи: 1 – передаются данные; 0 – передача завершена
0	INT	Флаг прерывания, выставляется всегда. Прерывание для процессора выдается если есть флаг EN_INT: 1 – есть прерывание; 0 – нет прерывания. Флаг выставляется если: – передача байта завершена; – был потерян арбитраж

**20.7.6 I2C->TXD**

Таблица 222 – Регистр TXD

Номер	7...0
Доступ	R/W
Сброс	0
	TXD[7:0]

Таблица 223 – Описание битов 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

### 20.7.7 I2C->CMD

Таблица 224 – Регистр CMD

Номер	7	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
	START	STOP	RD	WR	ACK	-	CLR INT

Таблица 225 – Описание битов регистра CMD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7	START	Отправить START bit Инициализируется записью 1 После завершения отправки автоматически не сбрасывается, очищается записью нуля
6	STOP	Отправить STOP bit Инициализируется записью 1 После завершения отправки автоматически не сбрасывается, 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 при чтении: 0 – отправить ACK; 1 – отправить NACK
2, 1	-	Зарезервировано
0	CLR INT	Очистить прерывание INT Запись 1 очищает прерывание

## 21 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 и передатчика;
- переполнение буфера FIFO приемника;
- наличие данных в буфере FIFO приемника по истечении времени таймаута.

Основные сведения о модуле представлены в следующих разделах:

- характеристики интерфейса SPI;
- характеристики интерфейса Microwire;
- характеристики интерфейса SSI.

### 21.1 Основные характеристики модуля SSP

- может функционировать как в ведущем, так и в ведомом режиме;
- программное управление скоростью обмена;
- состоит из независимых буферов приема и передачи (8 ячеек по 16 бит) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);
- программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;
- программируемая длительность информационного кадра от 4 до 16 бит;
- независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по переполнению буфера приемника;
- доступна возможность тестирования по шлейфу, соединяющему вход с выходом;

Структурная схема модуля представлена на рисунке 67.



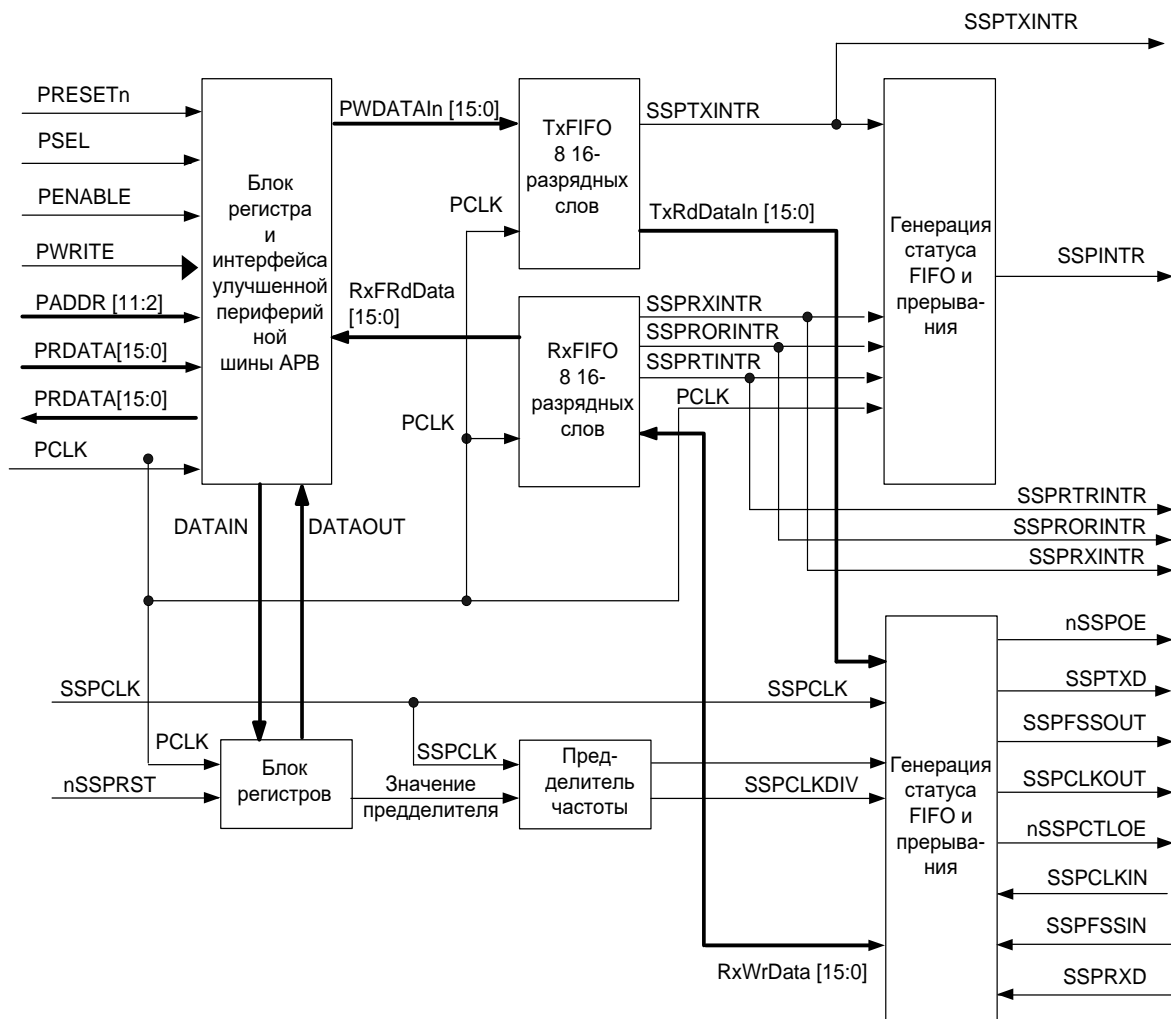


Рисунок 67 – Структурная схема модуля SSP

## 21.2 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.

## 21.3 Характеристики интерфейса SPI

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии;
- программное задание фазы и полярности тактового сигнала.

## 21.4 Характеристики интерфейса Microwire

Интерфейс Microwire фирмы National Semiconductor обеспечивает:

- полдуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

## 21.5 Характеристики интерфейса SSI

Интерфейс SSI фирмы Texas Instruments обеспечивает:

- полдуплексный обмен данными по четырехпроводной линии;
- возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

## 21.6 Общий обзор модуля SSP

Модуль SSP представляет собой интерфейс синхронного последовательного обмена данными, способный функционировать в качестве ведущего или ведомого устройства и 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, обеспечивающих хранение до восьми слов данных шириной 16 бит независимо для режимов приема и передачи.

Последовательные данные передаются по линии SSP\_TXD (см. пункт 21.8.1 «Описание регистров контроллера SSP») и принимаются с линии SSP\_RXD.

Модуль SSP содержит программируемые делители частоты, формирующие тактовый сигнал обмена данными SSP\_CLK из сигнала, поступающего на линию SSPCLK. Скорость передачи данных может достигать более 2 МГц, в зависимости от частоты SSPCLK и характеристик подключенного периферийного устройства.

Режим обмена данными, формат информационного кадра и количество бит данных задаются программно с помощью регистров управления CR0 и CR1.

Модуль формирует четыре независимо маскируемых прерывания:

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

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

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).

### 21.6.1 Блок формирования тактового сигнала

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSP\_CLK с помощью внутреннего делителя частоты, состоящего из двух последовательно соединенных счетчиков без цепи сброса.

Путем записи значения в регистр SSPCPSR можно задать коэффициент предварительного деления частоты в диапазоне от 2 до 254 с шагом 2. Так как младший значащий разряд коэффициента деления не используется, то исключается возможность деления частоты на нечетный коэффициент деления. Это, в свою очередь, гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровней).

Сформированный описанным образом сигнал далее поступает на второй делитель частоты, с выхода которого и снимается тактовый сигнал обмена данными SSP\_CLK.

Коэффициент деления второго делителя задается программно в диапазоне от 1 до 256, путем записи соответствующего значения в регистр управления SSPCR0.

### 21.6.2 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.

### 21.6.3 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.

### 21.6.4 Блок приема и передачи данных

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSP\_CLK для подключенных ведомых устройств. Как было описано ранее, данный сигнал формируется путем деления частоты сигнала SSPCLK.

Блок передатчика последовательно считывает данные из буфера FIFO передатчика и производит их преобразование из параллельной формы в последовательную. Далее поток последовательных данных и элементов кадровой синхронизации, тактированный сигналом SSP\_CLK, передается по линии SSP\_TXD к подключенным ведомым устройствам.

Блок приемника выполняет преобразование данных, поступающих синхронно с линии SSP\_RXD, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

В режиме ведомого устройства тактовый сигнал обмена данными формируется одним из подключенных к модулю периферийных устройств и поступает по линии SSP\_CLK.

При этом блок передатчика, тактируемый этим внешним сигналом, считывает данные из буфера FIFO, преобразует их из параллельной формы в последовательную, после чего выдает поток последовательных данных и элементов кадровой синхронизации в линию SSP\_TXD.

Аналогично, блок приемника выполняет преобразование данных, поступающих с линии SSP\_RXD синхронно с сигналом SSP\_CLK, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

Примечание – в режиме работы ведомого устройства, запросы ведущим устройством, на выдачу информации от ведомого, необходимо осуществлять при наличии данных в FIFO передатчика ведомого.

### **21.6.5 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 NVIC, при этом появляется дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

### **21.6.6 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.

### **21.6.7 Разрешение работы приемопередатчика**

Разрешение осуществляется путем установки бита SSE регистра управления CR1. Буфер FIFO передатчика может быть либо проинициализирован путем записи в него до восьми 16-разрядных слов заблаговременно перед установкой этого бита, либо может заполняться передаваемыми данными в процедуре обслуживания прерывания.

После разрешения работы модуля приемопередатчик начинает обмен данными по линиям SSP\_TXD и SSP\_RXD.

### 21.6.8 Соотношения между тактовыми сигналами

В модуле имеется ограничение на соотношение между частотами тактовых сигналов CPU\_CLK и SSPCLK. Частота SSPCLK должна меньше или равна частоте CPU\_CLK. Выполнение этого требования гарантирует синхронизацию сигналов управления, передаваемых из зоны действия тактового сигнала SSPCLK в зону действия сигнала CPU\_CLK в течение времени, меньшего продолжительности передачи одного информационного кадра:

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

В режиме ведомого устройства сигнал SSP\_CLK от ведущего внешнего устройства поступает на схемы синхронизации, задержки и обнаружения фронта. Для того, чтобы обнаружить фронт сигнала SSP\_CLK, необходимо три такта сигнала SSPCLK. Сигнал SSP\_TXD имеет меньшее время установки по отношению к заднему фронту SSP\_CLK, по которому и происходит считывание данных из линии. Время установки и удержания сигнала SSP\_RXD по отношению к сигналу SSP\_CLK должно выбираться с запасом, гарантирующим правильное считывание данных. Для обеспечения корректной работы устройства необходимо, чтобы частота SSPCLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSP\_CLK.

Выбор частоты тактового сигнала SSPCLK должен обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала SSPCLK к максимальной частоте сигнала SSP\_CLK в режиме ведомого устройства равно 12, в режиме ведущего – двум.

Так, в режиме ведущего устройства для обеспечения максимальной скорости обмена 1,8432 Мбит/с частота сигнала SSPCLK должна составлять не менее 3,6864 МГц. В этом случае в регистр CPSR должно быть записано значение 2, а поле SCR[7:0] регистра CR0 должно быть установлено в «0».

В режиме ведомого устройства для обеспечения той же информационной скорости необходимо использовать тактовый сигнал SSPCLK с частотой не менее 22,12 МГц. При этом в регистр CPSR должно быть записано значение 12, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

Соотношение между максимальной частотой сигнала SSPCLK и минимальной частотой SSPCLKOUT составляет  $254 \times 256$ .

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

$$F_{SSPCLK}(\min) \Rightarrow 2 \times F_{SSPCLKOUT}(\max) \text{ [в режиме Master]}$$

$$F_{SSPCLK}(\min) \Rightarrow 12 \times F_{SSPCLKIN}(\max) \text{ [в режиме Slave].}$$

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

$$F_{SSPCLK}(\max) \leq 254 \times 256 \times F_{SSPCLKOUT}(\min) \text{ [в режиме Master]}$$

$$F_{SSPCLK}(\max) \leq 254 \times 256 \times F_{SSPCLKIN}(\min) \text{ [в режиме Slave].}$$

### 21.6.9 Программирование регистра управления SSPCR0

Регистр CR0 предназначен для:

- установки скорости информационного обмена;
- выбора одного из трех протоколов обмена данными;
- выбора размера слова данных.

Скорость информационного обмена зависит от частоты внешнего тактового сигнала SSPCLK и коэффициента деления блока формирования тактового сигнала. Последний задается совместно значением поля SCR (Serial Clock Rate – скорость информационного обмена) регистра SSPCR0 и значением поля CPSDVSR (clock prescale divisor value – коэффициент деления тактового сигнала) регистра SSPCPSR.

Формат информационного кадра задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS регистра SSPCR0.

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

### 21.6.10 Программирование регистра управления SSPCR1

Регистр SSPCR1 предназначен для:

- выбора ведущего или ведомого режима функционирования приемопередатчика;
- включения режима проверки канала по шлейфу;
- разрешения или запрещения работы модуля.

Выбор ведущего режима осуществляется путем записи «0» в поле MS регистра SSPCR1 (это значение устанавливается после сброса автоматически).

Запись 1 в поле MS переводит приемопередатчик в режим ведомого устройства. В этом режиме разрешение или запрещение формирования сигнала передатчика SSP\_TXD осуществляется путем установки бита SOD (slave mode SSP\_TXD output disable – запрет линии SSP\_TXD для ведомого режима) регистра CR1. Указанная функция полезна при подключении к одной линии нескольких подчиненных устройств.

Для того, чтобы разрешить функционирование приемопередатчика, необходимо установить в «1» бит SSE (Synchronous Serial Port Enable – разрешение последовательного синхронного порта).

### 21.6.11 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 CPSDVSR, лежащий в диапазоне от 2 до 254, доступный для программирования через регистр CPSR. Сформированный сигнал далее поступает на делитель частоты с коэффициентом (1 + SCR) от 1 до 256, где значение SCR доступно для программирования через CR0.

Частота выходного тактового сигнала обмена данными SSP\_CLK определяется следующим соотношением:

$$FSSPCLKOUT = FSSPCLK / (CPSDVR \cdot (1+SCR)).$$

Например, в случае, если частота сигнала SSPCLK составляет 3,6864 МГц, а значение CPDVSР = 2, частота сигнала SSP\_CLK лежит в интервале от 7,2 кГц до 1,8432 МГц.

### 21.6.12 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. Возможно выбрать три базовых структуры построения кадра:

- SSI фирмы Texas Instruments;
- SPI фирмы Motorola;
- Microwire фирмы National Semiconductor.

Во всех трех режимах построения кадра тактовый сигнал SSP\_CLK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SSP\_CLK в неактивное состояние используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

В режимах SPI и Microwire выходной сигнал кадровой синхронизации передатчика SSP\_FSS имеет активный низкий уровень и поддерживается в низком уровне в течение всего периода передачи информационного кадра.

В режиме построения кадра SSI фирмы Texas Instruments перед началом каждого информационного кадра на выходе SSP\_FSS формируется импульс с длительностью, равной одному тактовому интервалу обмена данными. В этом режиме приемопередатчик SSP, равно как и ведомые периферийные устройства, передает данные в линию по переднему фронту сигнала SSP\_CLK, а считывает данные из линии по заднему фронту этого сигнала.

В отличие от полнодуплексных режимов передачи данных SSI и SPI, режим Microwire фирмы National Semiconductor использует специальный способ обмена данными между ведущим и ведомым устройством, функционирующий в режиме полудуплекса. В указанном режиме на внешнее ведомое устройство перед началом передачи информационного кадра посылается специальная восьмибитная управляющая последовательность. В течение всего времени передачи этой последовательности приемник не обрабатывает каких-либо входных данных. После того как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом общая длительность информационного кадра составляет от 13 до 25 бит.

### 21.6.13 Формат синхронного обмена SSI фирмы Texas Instruments

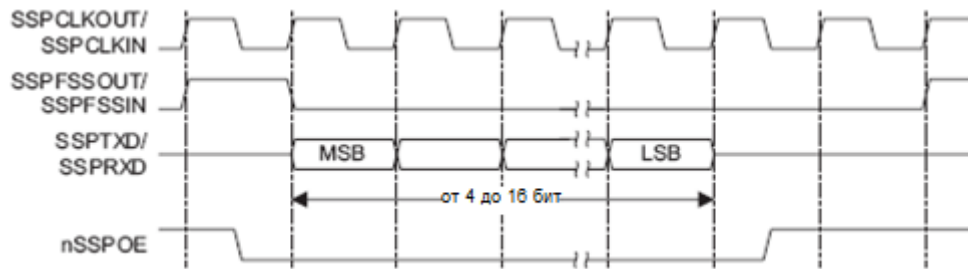


Рисунок 68 – Формат синхронного обмена протокола SSI фирмы Texas Instruments (единичный обмен)

В данном режиме при неактивном приемопередатчике SSP сигналы SSP\_CLK и SSP\_FSS переводятся в низкий логический уровень, а линия передачи данных SSP\_TXD поддерживается в третьем состоянии.

После появления хотя бы одного элемента в буфере FIFO передатчика сигнал SSPFSSOUT переводится в высокий логический уровень на время, соответствующее одному периоду сигнала SSP\_CLK. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSP\_CLK старший значащий разряд информационного кадра (4 – 16 бит данных) выдается на выход линии SSP\_TXD и т.д.

В режиме приема данных как модуль SSP, так и ведомое внешнее устройство последовательно загружают биты данных в сдвиговый регистр по заднему фронту сигнала SSP\_CLK. Принятые данные переносятся из сдвигового регистра в буфер FIFO после загрузки в него младшего значащего бита данных по очередному переднему фронту сигнала SSP\_CLK.

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments представлены на рисунке 68 (передача единичного информационно кадра) и на рисунке 69 (передача последовательности кадров).

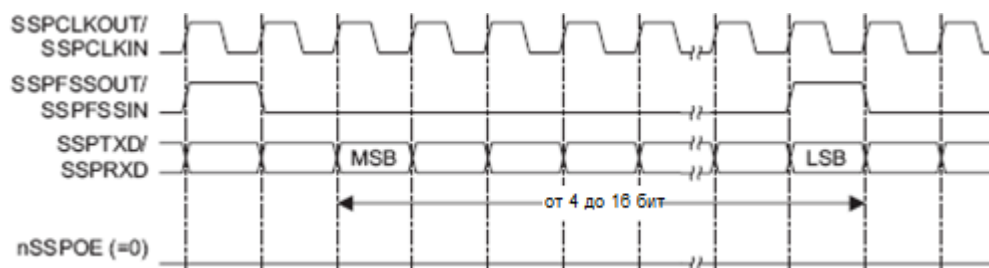


Рисунок 69 – Формат синхронного обмена протокола SSI фирмы Texas Instruments (непрерывный обмен)

### 21.6.14 Формат синхронного обмена SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSP\_FSS выполняет функцию выбора ведомого устройства. Главной особенностью протокола SPI является возможность выбора состояния и фазы сигнала SSP\_CLK в режиме ожидания (неактивном приемопередатчике) путем задания значений бит SPO и SPH регистра управления SSPSCR0.



Выбор полярности тактового сигнала – бит SPO

Если бит SPO равен «0», то в режиме ожидания линия SSP\_CLK переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SSP\_CLK переводится в высокий логический уровень.

Выбор фазы тактового сигнала – бит SPH

Значение бита SPH определяет фронт тактового сигнала, по которому осуществляется выборка данных и изменение состояния на выходе линии.

В случае, если бит SPH установлен в 0, регистрация данных приемником осуществляется после первого обнаружения фронта тактового сигнала, в противном случае – после второго.

### 21.6.15 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=0 показаны на рисунке 70 (одиночный обмен) и на рисунке 71 (непрерывный обмен).

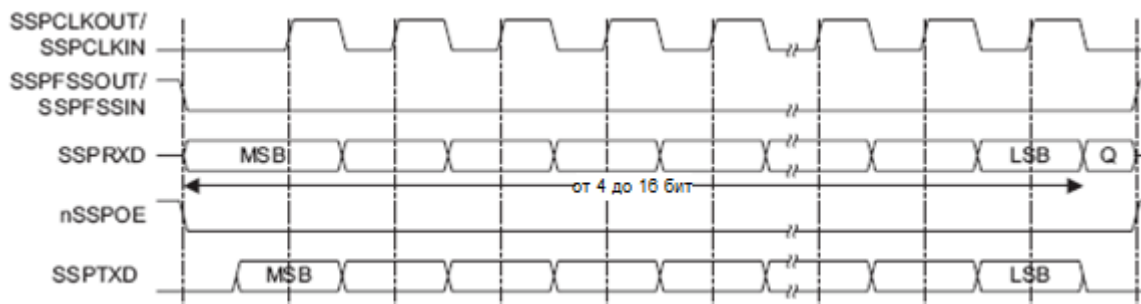


Рисунок 70 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=0 (одиночный обмен)

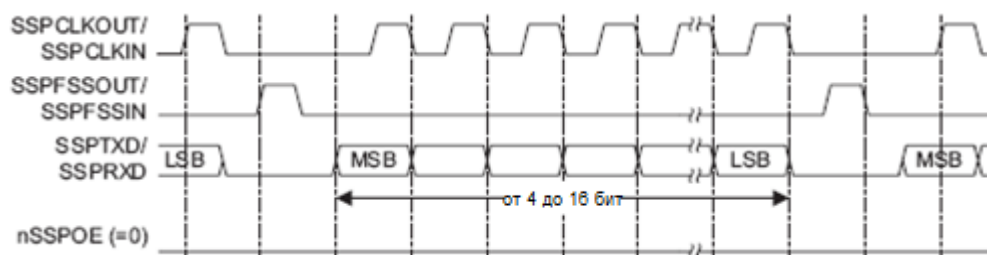


Рисунок 71 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0, SPH=0 (непрерывный обмен)

Примечание – На рисунке 70 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет низкий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP\_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого

устройства на входную линию SSP\_RXD ведущего. Контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP\_CLK на линии SSP\_TXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSP\_CLK переводится в высокий логический уровень.

Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP\_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP\_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP\_CLK.

В режиме непрерывной передачи данных на линии SSP\_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP\_FSS в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSP\_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP\_CLK.

### 21.6.16 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=1 показаны на рисунке 72 (одиночный и непрерывный обмен).

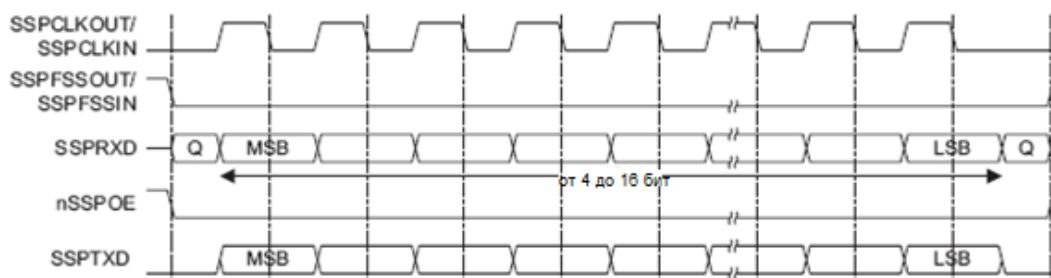


Рисунок 72 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=1

Примечание – На рисунке 72 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет низкий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP\_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого

устройства на входную линию SSP\_RXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP\_CLK на линиях обмена как ведущего, так и ведомого устройств будут сформированы значения первых бит передаваемых данных. В это же время включается линия SSP\_CLK и на ней формируется передний фронт сигнала.

Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSP\_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP\_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP\_CLK.

В режиме непрерывной передачи данных линия SSP\_FSS постоянно находится в низком логическом уровне, и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

### 21.6.17 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=0 показаны на рисунке 73 (одиночный обмен) и на рисунке 74 (непрерывный обмен).

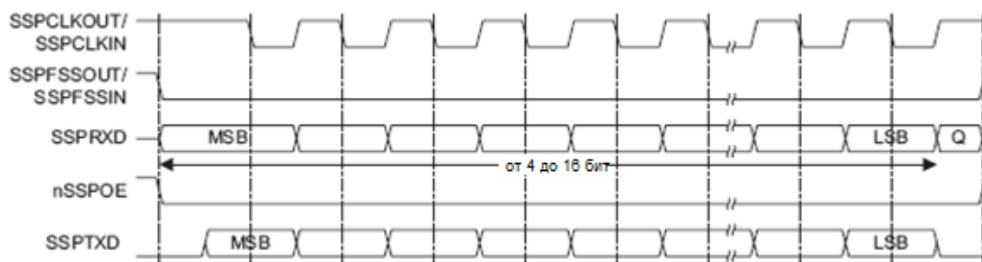


Рисунок 73 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)

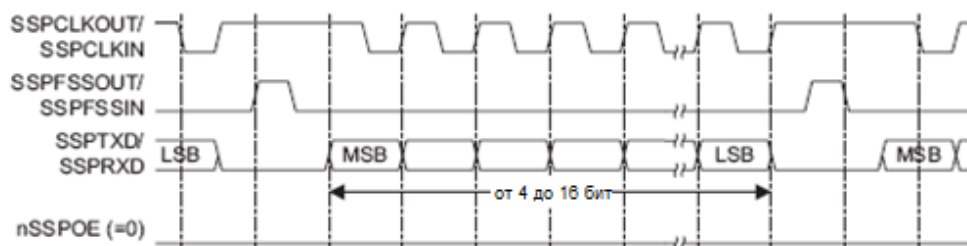


Рисунок 74 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (непрерывный обмен)

Примечание – На рисунке 73 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет высокий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP\_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого

устройства на входную линию SSP\_RXD ведущего. Выходной контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP\_CLK, на линии SSP\_TXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSP\_CLK переводится в низкий логический уровень.

Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSP\_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP\_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP\_CLK.

В режиме непрерывной передачи данных на линии SSP\_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP\_FSS в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSP\_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP\_CLK.

### 21.6.18 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=1 показаны на рисунке 75 (одиночный и непрерывный обмен).

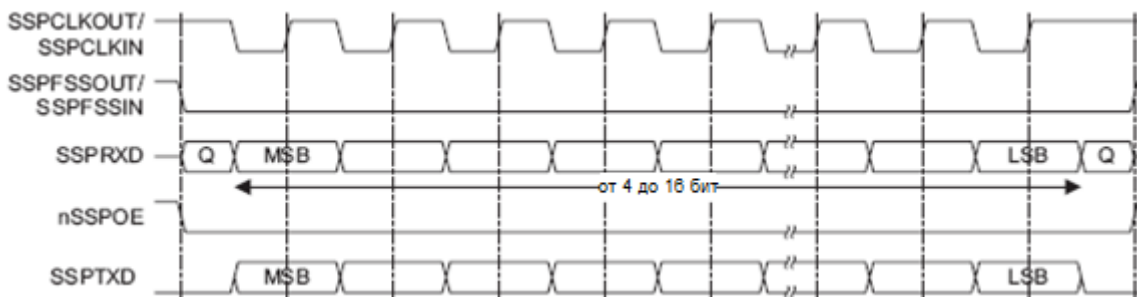


Рисунок 75 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=1

Примечание – На рисунке 75 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет высокий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP\_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP\_RXD ведущего. Выходной контакт передатчика SSP\_TXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP\_CLK на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых бит передаваемых данных. В это же время включается линия SSP\_CLK и на ней формируется передний фронт сигнала.

Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP\_CLK.

В случае передачи одного слова данных после приема его последнего бита линия SSP\_FSS переводится в высокий логический уровень по истечении одного периода тактового сигнала SSP\_CLK.

В режиме непрерывной передачи данных линия SSP\_FSS постоянно находится в низком логическом уровне и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

### 21.6.19 Формат синхронного обмена Microwire фирмы National Semiconductor

Временные диаграммы последовательного синхронного обмена в режиме Microwire показаны на рисунке 76 (одиночный обмен) и на рисунке 77 (непрерывный обмен).

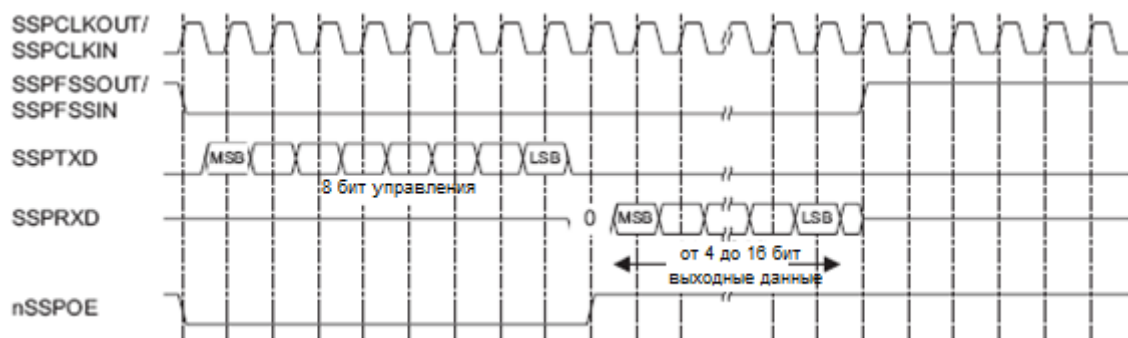


Рисунок 76 – Формат синхронного обмена протокола Microwire фирмы National Semiconductor (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен в нем осуществляется в полудуплексном режиме, с использованием служебных последовательностей. Каждая информационный обмен начинается с передачи ведущим устройством специальной восьмьбитной управляющей последовательности. В течение всего времени ее передачи приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSP\_CLK имеет низкий логический уровень;
- сигнал SSP\_FSS имеет высокий логический уровень;
- сигнал SSP\_TXD переводится в высокоимпедансное состояние.



Переход в режим информационного обмена происходит после записи управляющего байта в буфер FIFO передатчика. По заднему фронту сигнала SSP\_FSS данные из буфера переносятся в регистр сдвига блока передатчика, откуда, начиная со старшего значащего разряда, последовательно выдаются в линию SSP\_TXD. Линия SSP\_FSS остается в низком логическом уровне в течение всей передачи кадра. Линия SSP\_RXD при этом находится в высокоимпедансном состоянии.

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSP\_CLK. По окончании приема последнего бита управляющей последовательности она декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля SSP. Биты данных выдаются в линию SSP\_RXD по заднему фронту сигнала SSP\_CLK. Ведущее устройство, в свою очередь, регистрирует их по переднему фронту этого тактового сигнала. В случае одиночного информационного обмена по окончании приема последнего бита слова данных сигнал SSP\_FSS переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных из регистра сдвига в буфер FIFO приемника.

Примечание – Внешнее устройство может перевести линию приемника в третье состояние по заднему фронту сигнала SSP\_CLK после приема последнего бита слова данных, либо после перевода линии SSP\_FSS в высокий логический уровень.

Непрерывный обмен данными начинается и заканчивается также, как и одиночный обмен. Однако линия SSP\_FSS удерживается в низком логическом уровне в течение всего сеанса передачи данных. Управляющий байт следующего информационного кадра передается сразу же после приема младшего значащего разряда текущего кадра. Данные из сдвигового регистра передаются в буфер приемника после регистрации младшего разряда очередного слова по заднему фронту сигнала SSP\_CLK.

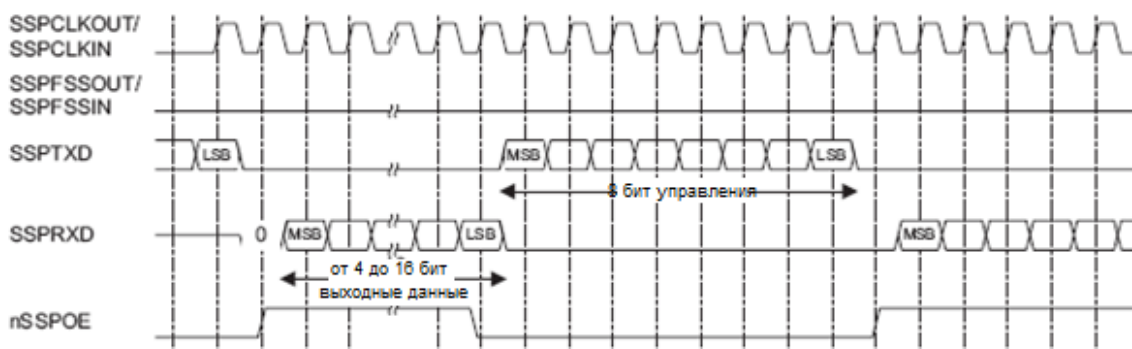


Рисунок 77 – Формат синхронного обмена протокола Microwire фирмы National Semiconductor (непрерывный обмен)

Требования к временным параметрам сигнала SSP\_FSS относительно тактового сигнала SSP\_CLK в режиме Microwire.

Модуль SSP, работающий в режиме Microwire как ведомое устройство, регистрирует данные по переднему фронту сигнала SSP\_CLK после установки сигнала SSP\_FSS в низкий логический уровень. Ведущие устройства, формирующие сигнал SSP\_CLK, должны гарантировать достаточное время установки и удержания сигнала SSP\_FSS по отношению к переднему фронту сигнала SSP\_CLK.

Данные требования иллюстрирует рисунок 78. По отношению к переднему фронту сигнала SSP\_CLK, по которому осуществляется регистрация данных в приемнике ведомого модуля SSP, время установки сигнала SSP\_FSS должно быть, как минимум в два раза больше периода SSP\_CLK, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSP\_CLK должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.

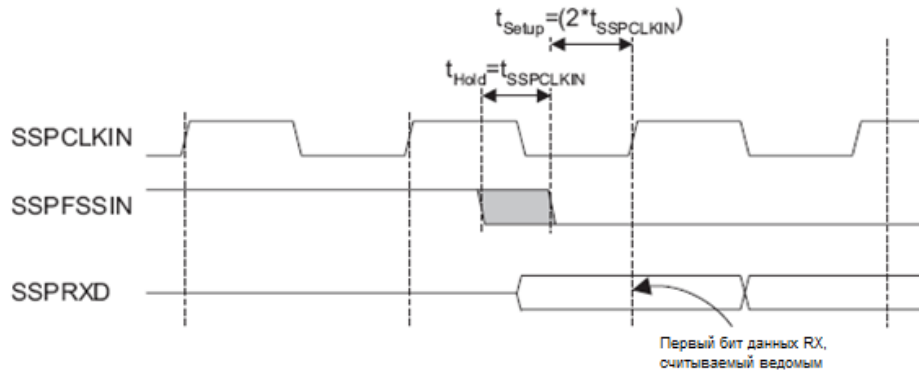


Рисунок 78 – Формат кадра Microwire, требования к времени установки и удержания сигнала SSPFSSIN

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

На рисунках 79 – 81 показаны варианты подключения модуля SSP к периферийным устройствам, работающим в ведущем или ведомом режиме.

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

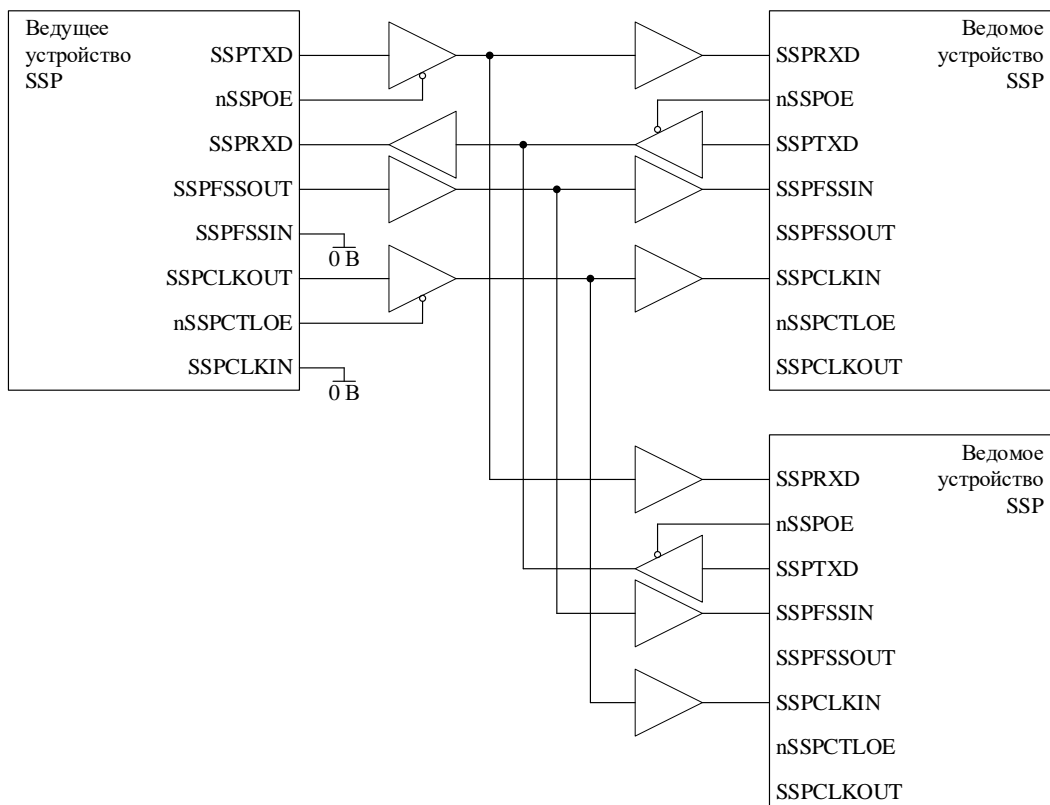


Рисунок 79 – Ведущее устройство SSP подключено к двум ведомым

Рисунок 79 показывает совместную работу трех модулей SSP, один из которых сконфигурирован в качестве ведущего, а два – в качестве ведомых устройств. Ведущее устройство способно передавать данные циркулярно в адрес двух ведомых по линии SSP\_TXD.

Для ответной передачи данных один из ведомых модулей разрешает прохождение сигнала от своей линии SSP\_TXD на вход SSP\_RXD ведущего.

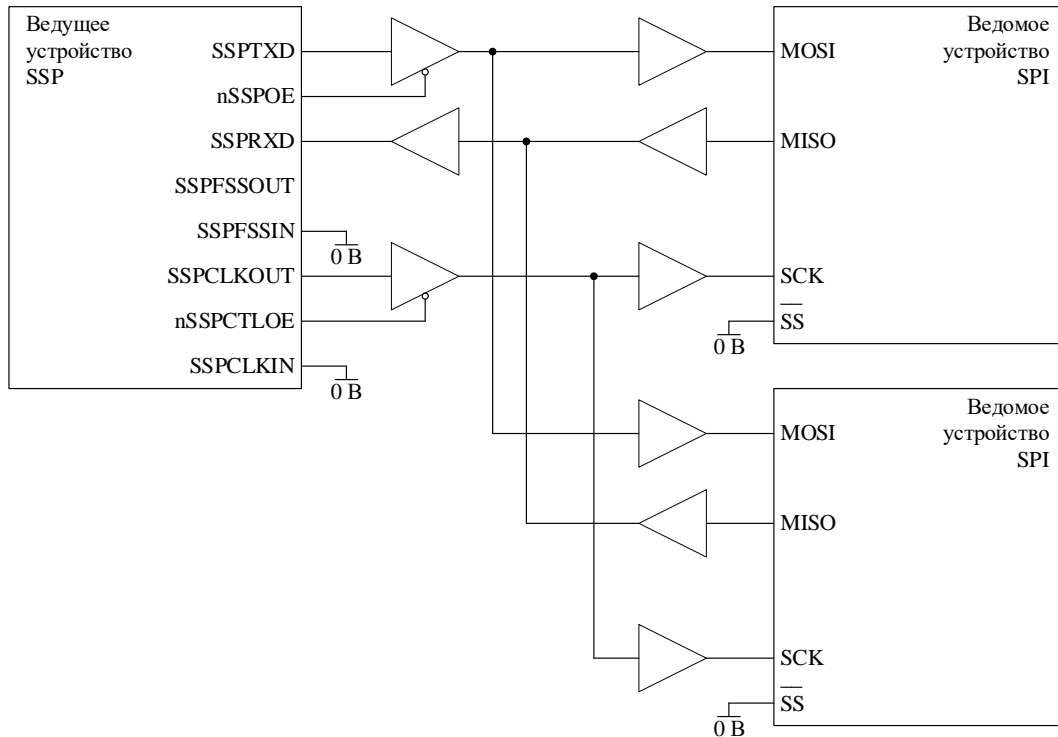


Рисунок 80 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим протокол SPI

Рисунок 80 показывает подключение модуля SSP, сконфигурированного как ведущее устройство, к двум ведомым устройствам, поддерживающим протокол SPI фирмы Motorola. Внешние устройства сконфигурированы как ведомые путем установки в низкий логический уровень сигнала выбора ведомого устройства Slave Select (SS). Как и в предыдущем примере, ведущее устройство способно передавать данные в адрес ведомых циркулярно по линии SSP\_TXD. Ответная передача данных на входную линию SSP\_RXD ведущего устройства одновременно осуществляется только одним из ведомых по соответствующей линии MISO.



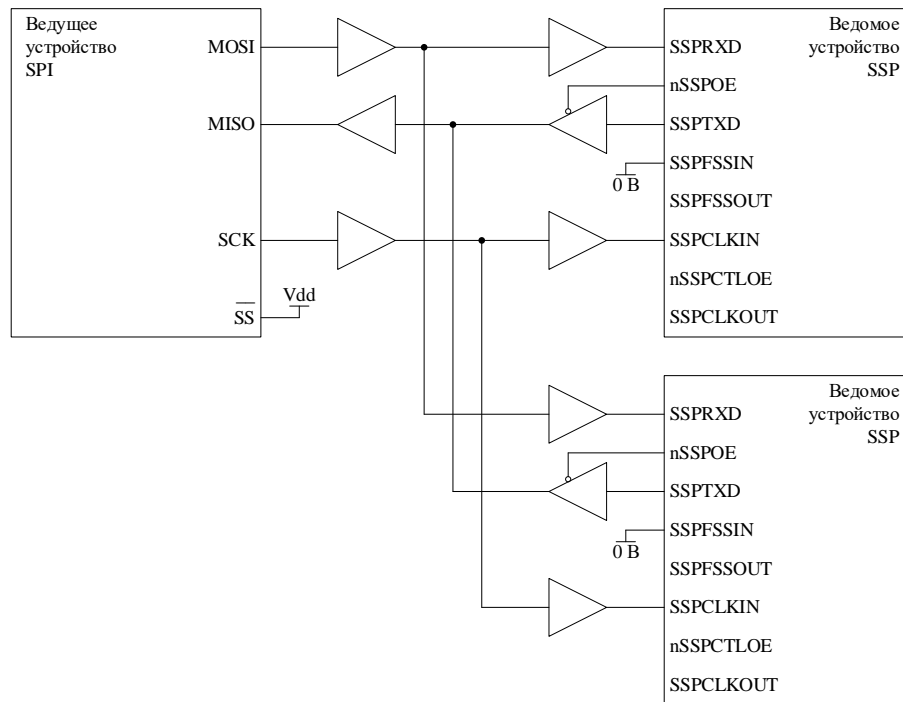


Рисунок 81 – Ведущее устройство, поддерживающее протокол SPI подключено к двум ведомым модулям SSP

Рисунок 81 показывает ведущее устройство, поддерживающее протокол SPI фирмы Motorola, соединенное с двумя модулями SSP, сконфигурированными для работы в ведомом режиме. Линия Slave Select (SS) ведущего устройства в этом случае установлена в высокий логический уровень. Ведущее устройство осуществляет передачу данных по линии MOSI циркулярно в адрес двух ведомых модулей.

Для ответной передачи данных один из ведомых модулей переводит линию SSP\_TXD в активное состояние, разрешая таким образом прохождение сигнала от своей линии SSP\_TXD на вход SSP\_RXD ведущего.

## 21.7 Прерывания

В модуле предусмотрено пять маскируемых линий запроса на прерывание, в том числе, четыре независимые линии запроса с активным высоким логическим уровнем, а также один общий сигнал, представляющий собой комбинацию независимых по схеме ИЛИ.

### 21.7.1 Сигналы запроса на прерывание:

- SSPRXINTR – запрос на обслуживание буфера FIFO приемника.
- SSPTXINTR – запрос на обслуживание буфера FIFO передатчика.
- SSPRORINTR – переполнение буфера FIFO приемника.
- SSPRTINTR – таймаут приемника.
- SSPINTR – логическое ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSP\_IMSC. Установка бита в «1» разрешает соответствующее прерывание, в 0 – запрещает.

Доступность как индивидуальных, так и общей линии запроса позволяет организовать обслуживание прерываний в системе как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика SSPRXINTR и SSPTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать данные сигналы запроса для обеспечения чтения и записи данных, согласованной с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний SSP\_RIS, либо из маскированного регистра прерываний SSP\_MIS.

### **21.7.2 SSPRXINTR**

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

### **21.7.3 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.

### **21.7.4 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.

### **21.7.5 SSPRTINTR**

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение времени таймаута, равного 32 тактам частоты SSPCLKOUT (для ведущего и ведомого режимов работы). Данный механизм гарантирует, что пользователь будет знать о наличии в буфере приемника необработанных данных.

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

### 21.7.6 SSPINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

## 21.8 Программное управление модулем. Общая информация

Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:

- адреса со смещениями в диапазоне +0x028 ... +0x07C и +0xFD0 ... +0xFDC зарезервированы для перспективных расширений возможностей модуля;
- адреса со смещениями в диапазоне +0x080 ... +0x088 зарезервированы для тестирования.

### 21.8.1 Описание регистров контроллера SSP

Данные о регистрах модуля SSP приведены в таблице 226.

Таблица 226 – Обобщенные данные о регистрах модуля SSP

Базовый адрес	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x4000_0000	SSP				Контроллер SSP
Смещение					
0x000	SSP_CR0	RW	0x0000	16	Регистр управления 0 (таблица 227)
0x004	SSP_CR1	RW	0x0	4	Регистр управления 1 (таблица 228)
0x008	SSP_DR	RW	0x----	16	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись) (таблица 229)
0x00C	SSP_SR	RO	0x03	3	Регистр состояния (таблица 230)
0x010	SSP_CPSR	RW	0x00	8	Регистр делителя тактовой частоты (таблица 231)
0x014	SSP_IMSC	RW	0x0	4	Регистр маски прерывания (таблица 232)
0x018	SSP_RIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования (таблица 233)
0x01C	SSP_MIS	RO	0x0	4	Регистр состояния прерываний с учетом маскирования (таблица 234)
0x020	SSP_ICR	WO	0x0	4	Регистр сброса прерывания (таблица 235)
Примечание – В поле «тип» указан вид доступа к регистру: RW – чтение и запись, RO – только чтение, WO – только запись					

21.8.1.1 SSP\_CR0

Регистр управления 0.

Регистр SSP\_CR0 содержит пять битовых полей, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 227.

Таблица 227 – Регистр SSP\_CR0

Бит	Наименование	Назначение
15:8	SCR	Скорость последовательного обмена. Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению: $F_{SSPCLK} / (CPSDVR \cdot (1 + SCR))$ , где CPSDVR – четное число в диапазоне от 2 до 254 (см. регистр SSP_CPSR), а SCR – число от 0 до 255
7	SPH	Фаза сигнала SSP_CLK (используется только в режиме обмена SPI фирмы Motorola). См. пункт 21.6.14 «Формат синхронного обмена SPI фирмы Motorola»
6	SPO	Полярность сигнала SSP_CLK (используется только в режиме обмена SPI фирмы Motorola). См. пункт 21.6.14 «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	Размер слова данных: 0000 – резерв; 0001 – резерв; 0010 – резерв; 0011 – 4 бита; 0100 – 5 бит; 0101 – 6 бит; 0110 – 7 бит; 0111 – 8 бит; 1000 – 9 бит; 1001 – 10 бит; 1010 – 11 бит; 1011 – 12 бит; 1100 – 13 бит; 1101 – 14 бит; 1110 – 15 бит; 1111 – 16 бит

### 21.8.1.2 SSP\_CR1

Регистр управления 1.

Регистр SSP\_CR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 228.

Таблица 228 – Регистр SSP\_CR1

Биты	Наименование	Назначение
15:4		Резерв, при чтении результат не определен. При записи следует устанавливать в 0
3	SOD	Запрет выходных линий в режиме ведомого устройства. Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. Бит SOD следует установить в случае, если данный ведомый модуль SSP не должен в настоящее время осуществлять передачу данных в линию SSP_TXD. При этом линии обмена данных ведомых устройств можно соединить параллельно: 0 – управление линией SSP_TXD в ведомом режиме разрешена; 1 – управление линией SSP_TXD в ведомом режиме запрещена
2	MS	Выбор ведущего или ведомого режима работы: 0 – ведущий модуль (устанавливается по умолчанию); 1 – ведомый модуль
1	SSE	Разрешение работы приемопередатчика: 0 – работа запрещена; 1 – работа разрешена
0	LBM	Тестирование по шлейфу: 0 – нормальный режим работы приемопередатчика; 1 – выход регистра сдвига передатчика соединен со входом регистра сдвига приемника

### 21.8.1.3 SSP\_DR

Регистр данных.

Регистр SSP\_DR имеет разрядность 16 бит и предназначен для чтения принятых и записи передаваемых данных.

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

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

В случае, если выбран размер информационного слова менее 16 бит, перед записью в регистр SSP\_DR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые

информационные слова автоматически выравниваются по правой границе в блоке приемника.

В режиме обмена данными Microwire фирмы National Semiconductor модуль SSP по умолчанию работает с восьмиразрядными информационными словами (старший значащий байт игнорируется). Размер принимаемых данных задается программно. Буфера FIFO приемника и передатчика автоматически не очищаются даже в случае, если бит SSE установлен в 0. Это позволяет заполнить буфер передатчика необходимой информацией заблаговременно, перед разрешением работы модуля.

Назначение разрядов регистра SSP\_DR описано в таблице 229 .

Таблица 229 – Формат регистра SSP\_DR

Бит	Наименование	Назначение
15:0	DATA	Принимаемые данные (чтение) Передаваемые данные (запись) В случае, если выбран размер информационного слова менее 16 бит, перед записью в регистр SSP_DR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника

#### 21.8.1.4 SSP\_SR

Регистр состояния.

Регистр состояния доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика, и занятости модуля SSP.

В таблице 230 представлено назначение бит регистра SSP\_SR.

Таблица 230 – Регистр SSP\_SR

Биты	Наименование	Назначение
15:5		Резерв, при чтении результат не определен
4	BSY	Флаг занятости модуля: 0 – модуль SSP неактивен; 1 – модуль SSP в настоящее время передает и/или принимает данные, либо буфер FIFO передатчика не пуст
3	RFF	Буфер FIFO приемника заполнен: 0 – не заполнен; 1 – заполнен
2	RNE	Буфер FIFO приемника не пуст: 0 – пуст; 1 – не пуст
1	TNF	Буфер FIFO передатчика не заполнен: 0 – заполнен; 1 – не заполнен
0	TFE	Буфер FIFO передатчика пуст: 0 – не пуст; 1 – пуст

### 21.8.1.5 SSP\_CPSR

Регистр делителя тактовой частоты.

Регистр SSP\_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 SSP\_CPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Назначение бит регистра SSP\_CPSR представлено в таблице 231.

Таблица 231 – Регистр SSP\_CPSR

Биты	Наименование	Назначение
15:8		Резерв, при чтении результат не определен. При записи следует заполнить нулями
7:0	CPSDVSR	Коэффициент деления тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль

### 21.8.1.6 SSP\_IMSC

Регистр установки и сброса маски прерывания.

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание. При этом запись 1 в разряд разрешает соответствующее прерывание, запись 0 – запрещает.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение битов регистра SSP\_IMSC показано в таблице 232.

Таблица 232 – Регистр SSP\_IMSC

Биты	Наименование	Назначение
15:4		Резерв. Не модифицируйте. При чтении выдаются нули.
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 более 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

### 21.8.1.7 SSP\_RIS

Регистр состояния прерываний.

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре SSP\_RIS представлено в таблице 233.

Таблица 233 – Регистр SSP\_RIS

Биты	Наименование	Назначение
15:4		Резерв. Не модифицируйте. При чтении выдаются нули
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

### 21.8.1.8 SSP\_MIS.

Регистр маскированного состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре SSP\_MIS представлено в таблице 234.

Таблица 234 – Регистр SSP\_MIS

Биты	Наименование	Назначение
15:4		Резерв. Не модифицируйте. При чтении выдаются нули
3	TXMIS	Состояние маскированного прерывания SSPTXINTR
2	RXMIS	Состояние маскированного прерывания SSPRXINTR
1	RTMIS	Состояние маскированного прерывания SSPRTINTR
0	RORMIS	Состояние маскированного прерывания SSPRORINTR

### 21.8.1.9 SSP\_ICR

Регистр сброса прерываний.

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись в любой из разрядов регистра 0 игнорируется.

Назначение бит в регистре SSP\_ICR представлено в таблице 235.

Таблица 235 – Регистр SSP\_ICR

Биты	Наименование	Назначение
15:2		Резерв. Не модифицируйте. При чтении выдаются нули
1	RTIC	Сброс прерывания SSPRTINTR
0	RORIC	Сброс прерывания SSPRORINTR



## 22 Контроллер UART

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

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

### 22.1 Основные характеристики модуля UART

Может быть запрограммирован для использования как в качестве универсального асинхронного приемопередатчика, так и для инфракрасного обмена данными (SIR).

Содержит независимые буферы приема (16×12) и передачи (16×8) 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 байтом.

### 22.2 Программное управление скоростью обмена

Обеспечивается возможность деления тактовой частоты опорного генератора в диапазоне (1х16 – 65535х16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц.

Поддержка стандартных элементов асинхронного протокола связи – стартового и стопового бит, а та же бита контроля четности, которые добавляются перед передачей и удаляются после приема.

Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по таймауту приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки.

Поддержка прямого доступа к памяти.

Обнаружение ложных стартовых бит.

Формирование и обнаружения сигнала разрыва линии.

Функция управления модемом (линии nUART0CTS, nUART0DCD, nUART0DSR, nUART0RTS, nUART0DTR и nUART0RI (см. таблицу 1)).

Возможность организации аппаратного управления потоком данных.

Полностью программируемый асинхронный последовательный интерфейс с характеристиками:

- данные длиной 5, 6, 7 или 8 бит;
- формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение, либо не передается);
- формирование 1 или 2 стоповых бит;
- скорость передачи данных – от 0 до UARTCLK/16 Бод.

Кодек ИК-обмена данными IrDA SIR обеспечивает:

- программный выбор обмена данными по линиям асинхронного приемопередатчика либо кодека ИК-связи IrDA SIR;
- поддержку функционирования с информационной скоростью до 115200 бит/с в режиме полудуплекса;
- поддержку длительности бит для нормального режима (3/16) и для режима пониженного энергопотребления (1,41 – 2,23 мкс);
- программируемое деление опорной частоты UARTCLK для получения заданной длительности бит в режиме пониженного энергопотребления.

Наличие идентификационного регистра, однозначно идентифицирующего модуль, что позволяет операционной системе выполнять автоматическую конфигурацию.

### 22.3 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

- скорость передачи данных – целая и дробная часть числа;
- количество бит данных;
- количество стоповых бит;
- режим контроля четности;
- разрешение или запрет использования буферов FIFO (глубина очереди данных – 16 элементов или один элемент, соответственно);
- порог срабатывания прерывания по заполнению буферов FIFO (1/8, 1/4, 1/2, 3/4 и 7/8);
- частота внутреннего тактового генератора (номинальное значение -1,8432 МГц) может быть задана в диапазоне 1,42 – 2,12 МГц для обеспечения возможности формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления;
- режим аппаратного управления потоком данных.

### 22.4 Отличительные особенности контроллера UART

Контроллер отличается от промышленного стандарта асинхронного приемопередатчика 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 не поддерживаются:

- полуторная длительность стопового бита (поддерживается только 1 или 2 стоповых бита);
- независимое задание тактовой частоты приемника и передатчика.

## 22.5 Функциональные возможности

Устройство выполняет следующие функции:

- преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;
- преобразование данных, передаваемых на периферийное устройство, из параллельной в последовательную форму.

Процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии модуля. Прием и передача данных буферизуются с помощью внутренней памяти FIFO, позволяющей сохранить до 16 байтов независимо для режимов приема и передачи.

## 22.6 Модуль приемопередатчика:

- содержит программируемый генератор, формирующий тактовый сигнал одновременно для передачи и для приема данных на основе внутреннего тактового сигнала UARTCLK;
- обеспечивает возможности, сходные с возможностями промышленного стандарта - контроллера UART;
- позволяет осуществлять обмен информацией с максимальной скоростью:
  - в режиме UART – до 921600 бит/с;
  - в режиме IrDA – до 460800 бит/с;
  - в режиме IrDA с пониженным энергопотреблением – до 115200 бит/с.

Режим работы приемопередатчика и скорость обмена данными контролируются регистром управления линией UARTLCR\_H и регистрами делителя скорости передачи данных – целой части (UARTIBRD) и дробной части (UARTFBRD).

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.

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.

## 22.7 Описание функционирования блока UART

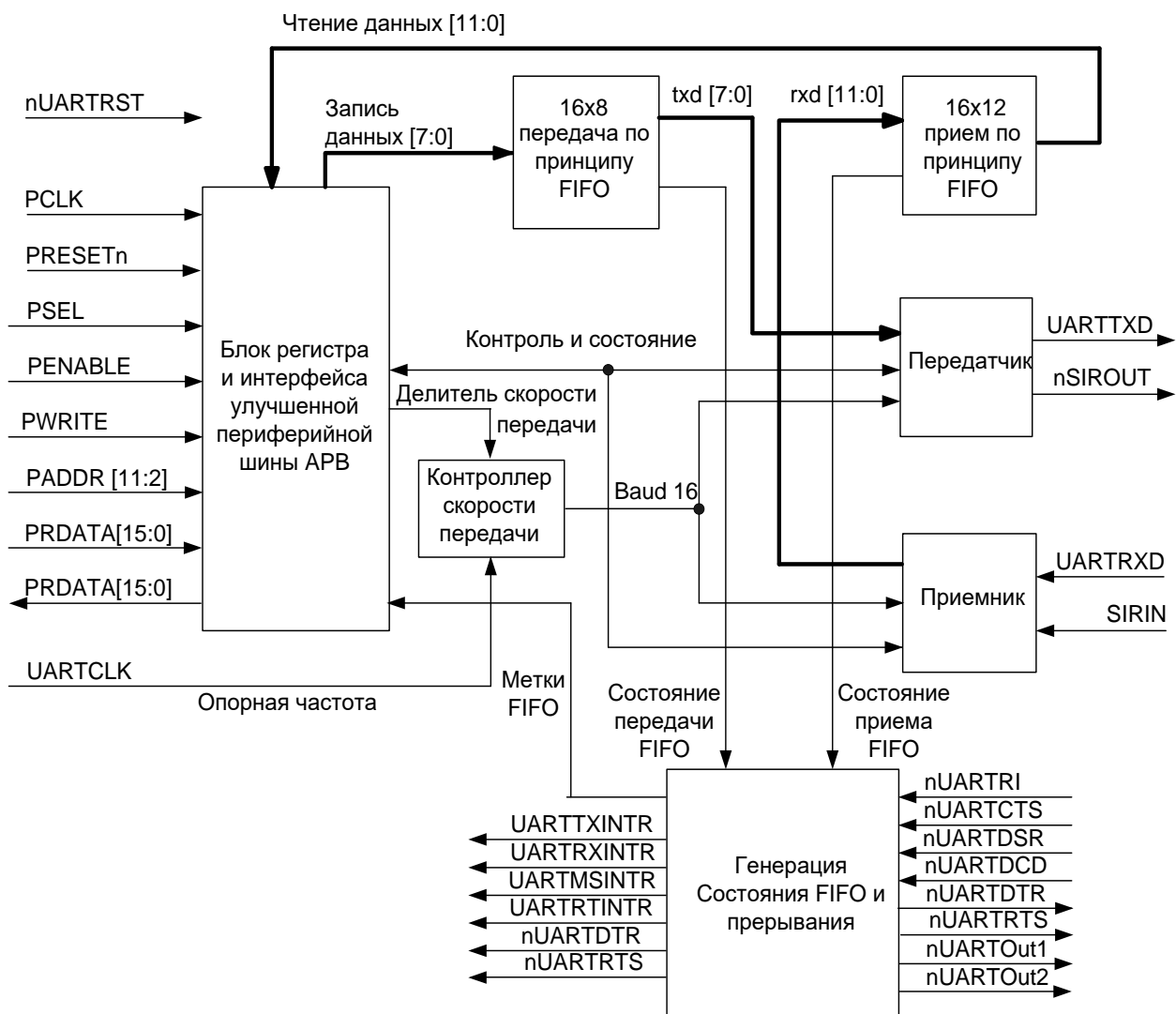


Рисунок 82 – Блок-схема универсального асинхронного приемопередатчика (УАПП)

### 22.7.1 Генератор тактового сигнала приемопередатчика

Генератор содержит счетчики без цепи сброса, формирующие внутренние тактовые сигналы Vaud16 и IrLPVaud16.

Сигнал Vaud16 используется для синхронизации схем управления приемником и передатчиком последовательного обмена данными. Он представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UARTCLK и частотой, в 16 раз выше скорости передачи данных.

Сигнал IrLPVaud16 предназначен для синхронизации схемы формирования импульсов с длительностью, требуемой для ИК-обмена данными в режиме с пониженным энергопотреблением.

### 22.7.2 Буфер FIFO передатчика

Буфер передатчика имеет ширину 8 бит, глубину 16 слов, схему организации доступа типа «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.

### 22.7.3 Буфер FIFO приемника

Буфер приемника имеет ширину 12 бит, глубину 16 слов, схему организации доступа типа «первый вошел, первый вышел». Принятые от периферийного устройства данные и соответствующие коды ошибки сохраняются логикой приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину APB. Буфер FIFO приемника может быть запрещен, в этом случае он будет действовать как однобайтовый буферный регистр.

### 22.7.4 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.

### 22.7.5 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 переполнения буфера, проверки на ошибки контроля четности, на ошибки в структуре сигнала, а также на разрыв линии. Признаки обнаружения этих ошибок также сохраняются в выходном буфере.

### 22.7.6 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 линий запроса на прерывание от приемопередатчика. В этом случае процедура обработки сможет одновременно считать информацию обо всех источниках прерывания. Данный подход привлекателен в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

Для более подробной информации см. подраздел 22.11 «Прерывания».

### 22.7.7 Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов CPU\_CLK и UARTCLK. 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 UARTCLK, так и наоборот.

## 22.8 Описание функционирования ИК-кодека IrDA SIR

Структурная схема кодека представлена на рисунке 83.

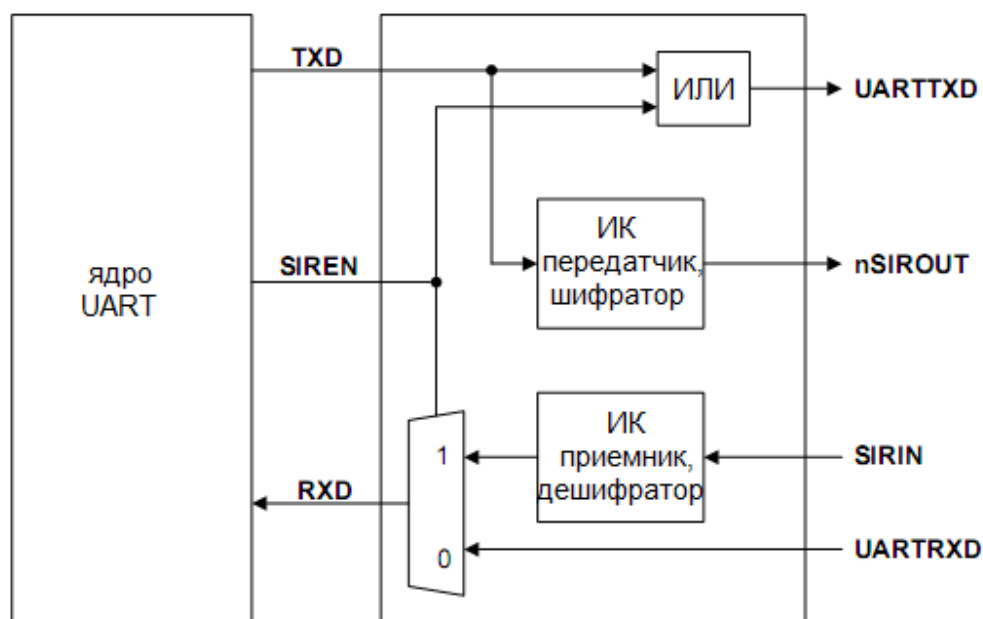


Рисунок 83 – Структурная схема кодека

### 22.8.1 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, три периода внутреннего тактового генератора с частотой Baud16, то есть 3/16 периода времени, выделенного на передачу одного бита.

В режиме IrDA с пониженным энергопотреблением ширина импульса задана как 3/16 периода, выделенного на передачу бита, при скорости передачи данных 115200 бит/с. Данное требование реализуется за счет формирования трех периодов тактового сигнала IrLPBaud16 с номинальной частотой 1.8432 МГц, в свою очередь, формируемого путем деления частоты UARTCLK. Значение частоты IrLPBaud16 задается путем записи соответствующего коэффициента деления частоты в регистр UARTILPR.

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 UARTCLK. Можно показать, что в наихудшем случае величина джиттера в потоке ИК-импульсов может достигать трех периодов UARTCLK. В соответствии со спецификацией стандарта IrDA SIR, джиттер не должен превышать величины 13 %. В случае, если частота сигнала UARTCLK 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.

### 22.8.2 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 нормально в высоком состоянии. Выходной сигнал кодера имеет полярность, противоположную полярности входа декодера.

Обнаружение стартового бита осуществляется при низком уровне сигнала на входе декодера.

Примечание – Для того, чтобы исключить ложные срабатывания UART от импульсных помех, на входе SIRIN игнорируются импульсы с длительностью менее, чем:

- 3/16 длительности Baud16 в режиме IrDA;
- 3/16 длительности IrLPBaud16 в режиме IrDA с пониженным энергопотреблением.

## 22.9 Описание работы UART

### 22.9.1 Сброс модуля

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса процессора. Значения регистров после сброса описаны в подразделе 22.12 «Программное управление модулем».

### 22.9.2 Тактовые сигналы

Частота тактового сигнала UARTCLK должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

$$F\_UARTCLK(\min) \geq 16 \cdot \text{baud\_rate\_max};$$

$$F\_UARTCLK(\max) \leq 16 \cdot 65535 \cdot \text{baud\_rate\_min}.$$

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота UARTCLK должна находиться в интервале от 7,3728 до 115,34 МГц.

Частота UARTCLK, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами CPU\_CLK и UARTCLK. Частота UARTCLK должна быть не более, чем в 5/3 раз выше частоты CPU\_CLK.

$$F\_UARTCLK \leq 5/3 \cdot F\_CPU\_CLK.$$

Например, при работе в режиме UART с максимальной скоростью передачи данных 921600 бод, при частоте UARTCLK 14.7456 МГц, частота 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.

### 22.9.3 Работа универсального асинхронного приемопередатчика

Управляющая информация хранится в регистре управления линией UARTLCR. Этот регистр имеет внутреннюю ширину 30 бит, однако внешний доступ по шине APB к нему осуществляется через следующие регистры:

- UARTLCR\_H – определяет:
  - параметры передачи данных;
  - длину слова;
  - режим буферизации;
  - количество передаваемых стоповых бит;
  - режим контроля четности;
  - формирование сигнала разрыва линии;
- UARTIBRD – определяет целую часть коэффициента деления для скорости передачи данных;
- UARTFBRD – определяет дробную часть коэффициента деления для скорости передачи данных.



#### 22.9.4 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, при этом используя в качестве UARTCLK тактовый сигнал с произвольной частотой более 3,6864 МГц.

Целая часть коэффициента деления записывается в 16-битный регистр UARTIBRD. Шестиразрядная дробная часть записывается в регистр UARTFBRD. Значение коэффициента деления связано с содержимым указанных регистров следующим образом:

$$\text{Коэффициент деления} = \frac{UARTCLK}{(16 \cdot \text{скорость передачи данных})} = BRD\_I + BRD\_F ,$$

где BRD\_I – целая часть, а BRD\_F – дробная часть коэффициента деления.

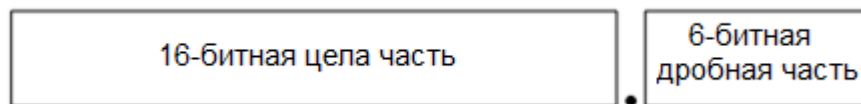


Рисунок 84 – Коэффициент деления

Шестибитное значение, записываемое в регистр UARTFBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на  $2^6$ , где  $n$  – ширина регистра UARTFBRD) и округления до ближайшего целого числа:

$$M = \text{integer}(BRD\_F \cdot 2^n + 0,5) ,$$

где integer – операция отсечения дробной части числа,  $n = 6$ .

В модуле формируется внутренний сигнал Vaud16, представляющий собой последовательность импульсов с длительностью, равной периоду сигнала UARTCLK и средней частотой, в 16 раз большей требуемой скорости обмена данными.

#### 22.9.5 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.

Для передачи данные заносятся в буфер FIFO передатчика. Если работа приемопередатчика разрешена, начинается передача информационного кадра с параметрами, указанными в регистре управления линией UARTLCR\_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 состояний в запрещенное.

Для каждого бита данных (в приемной линии) производится три измерения уровня, решение принимается по мажоритарному принципу.

В случае, если приемник находился в неактивном состоянии (на линии входного сигнала UART\_RXD постоянно присутствовала единица) и произошел переход входного сигнала из высокого в низкий логический уровень (обнаружен стартовый бит), включается счетчик, тактируемый сигналом Baud16, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов (в режиме асинхронного приемопередатчика) или каждые четыре такта (в режиме ИК обмена данными) сигнала Baud16. Более частая выборка данных в режиме ИК обмена связана с необходимостью корректной обработки импульсов данных согласно протоколу SIR IrDA.

Стартовый бит считается достоверным в случае, если сигнал на линии UART\_RXD сохраняет низкий логический уровень в течение восьми отсчетов сигнала Baud16 с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

В случае, если обнаружен достоверный стартовый бит, производится регистрация последовательности данных на входе приемника. Очередной бит данных фиксируются каждые 16 отсчетов тактового сигнала Baud16 (что соответствует длительности одного символа). Производится регистрация всех бит данных (согласно запрограммированным параметрам) и бита четности (если включен режим контроля четности).

Наконец, производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UART\_RXD). В случае, если последнее условие не выполняется, устанавливается признак ошибки формирования кадра. После того, как слово данных принято полностью, оно заносится в буфер FIFO приемника, наряду с четырьмя битами признаков ошибки, связанных с принятым словом (см. таблицу 236).

#### **22.9.6 Биты ошибки**

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся на позиции [10:8] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO, расположенный на позиции 11 слова данных.

В таблице 236 описано назначение всех битов слова данных в FIFO буфере приемника.

#### **22.9.7 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.

Таблица 236 – Биты буфера FIFO

Бит буфера FIFO	Назначение
11	Признак переполнения буфера
10	Ошибка – разрыв линии
09	Ошибка проверки на четность
08	Ошибка формирования кадра
07:00	Принятые данные

### 22.9.8 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.

### 22.9.9 Проверка по шлейфу

Проверка по шлейфу (замыкание выхода передатчика на вход приемника) выполняется путем установки в «1» бита LBE в регистре управления контроллером UARTCR.

### 22.9.10 Работа кодека ИК-обмена данными IrDA SIR

Кодек обеспечивает сопряжение асинхронного потока данных, сформированного приемопередатчиком, с полудуплексным последовательным интерфейсом IrDA SIR. Какая-либо аналоговая обработка сигнала при этом не выполняется. Назначение кодека – сформировать цифровой поток данных на вход приемника асинхронного сигнала и обработать цифровой поток данных с выхода передатчика.

Предусмотрено два режима работы:

– В режиме IrDA уровень логического нуля передается на линию nSIROUT в виде импульса с высоким логическим уровнем и длительностью 3/16 от выбранного периода следования бит данных. Логическая единица при этом передается в виде постоянного низкого уровня сигнала. Сформированный выходной сигнал далее подается на передатчик ИК-сигнала, обеспечивая излучение светового импульса всякий раз при передаче нулевого бита. На приемной стороне световые импульсы воздействуют на базу фототранзистора ИК приемника, который в результате формирует низкий логический уровень. Это, в свою очередь, обуславливает низкий уровень на входе SIRIN.

– В режиме IrDA с пониженным энергопотреблением длительность передаваемых импульсов ИК излучения устанавливается в три раза выше длительности импульсов внутреннего опорного сигнала IrLPBaud16 (равной 1,63 мкс при номинальной

частоте 1,8432 МГц). Данный режим активизируется путем установки бита SIRLP в регистре управления UARTCR.

Как в нормальном режиме, так и в режиме пониженного энергопотребления:

- кодирование осуществляется на основе бит данных, сформированных асинхронным передатчиком модуля;
- в ходе приема данных декодированные биты далее обрабатываются блоком асинхронного приема.

В соответствии со спецификацией физического уровня протокола IrDA SIR, обмен данными должен осуществляться в режиме полудуплекса, при этом задержка между передачей и приемом данных должна составлять не менее 10 мс. Эта задержка должна формироваться программно. Необходимость ее введения обусловлена тем, что воздействие передающего ИК светодиода на находящийся рядом ИК приемник может привести к искажению принимаемого сигнала или даже ввести приемный тракт в состояние насыщения. Задержка между окончанием передачи и началом приема данных именуется латентность, или время установки (готовности) приемника.

Сигнал IrLPBaud16 формируется путем деления частоты сигнала UARTCLK в соответствии с коэффициентом деления, записанным в регистре UARTILPR.

Коэффициент деления вычисляется по формуле

$$F\_UARTCLK / F\_IrLPBaud16 .,$$

где номинальное значение IrLPBaud16 составляет 1,8432 МГц. Коэффициент деления должен быть выбран так, чтобы выполнялось соотношение:

$$1,42 \text{ МГц} < F\_IrLPBaud16 < 2,12 \text{ МГц} .$$

### 22.9.11 Проверка по шлейфу

Проверка по шлейфу выполняется после установки в 1 бита LBE регистра управления контроллером UARTCR с одновременной установкой в 1 бита SIRTEST регистра управления тестированием UARTTCR.

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.

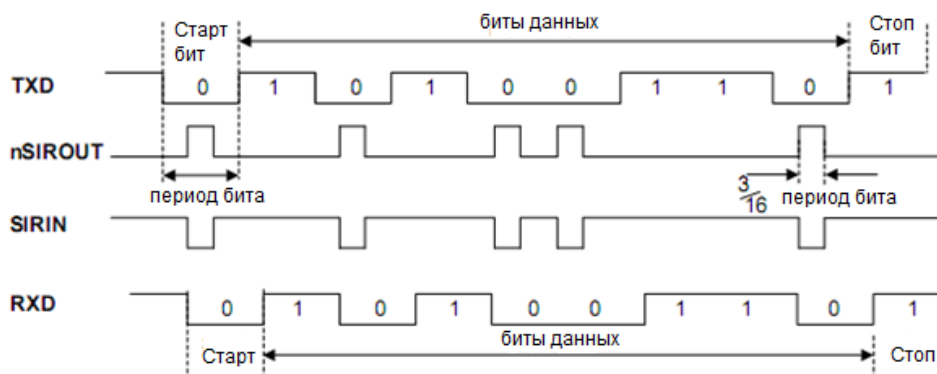


Рисунок 85 – Модуляция данных IrDA

## 22.10 Линии управления модемом

Модуль универсального асинхронного приемопередатчика может использоваться как в режиме оконечного оборудования (DTE), так и в режиме оборудования передачи данных (DCE). На рисунке 86 показаны сигналы модема в режиме DTE. Назначение сигналов в режиме DCE представлены в таблице 237.

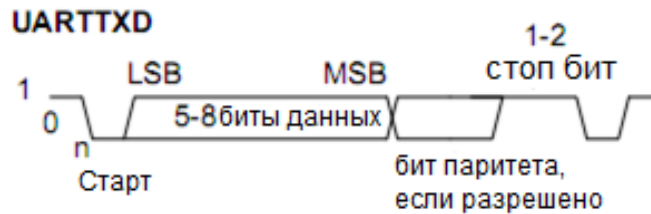


Рисунок 86 – Кадр передачи данных

Таблица 237 – 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	-
nUARTCTS	Запрос передачи данных	Готов к передаче данных
nUARTDTR	Приемник данных готов	Источник данных готов
nUARTOUT1	-	Обнаружен информационный сигнал
nUARTOUT2	-	Индикатор вызова

### 22.10.1 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 87.

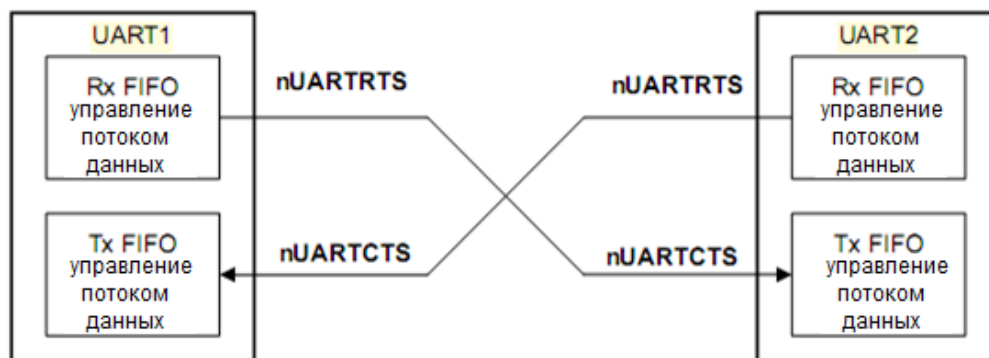


Рисунок 87 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

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 UARTCR. В таблице 238 показаны необходимые установки для различных режимов управления потоком данных.

Таблица 238 – Режимы управления потоком данных

CTSEn	RTSEn	Описание
1	1	Разрешено управление потоком данных по CTS и RTS
1	0	Управления потоком данных осуществляется по линии CTS
0	1	Управления потоком данных осуществляется по линии RTS
0	0	Управления потоком данных запрещено
Примечание – В случае, если выбран режим управления потоком данных по RTS, программное обеспечение не может использовать бит RTSEn регистра UARTCR для проверки состояния линии RTS		

### 22.10.2 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.

### 22.10.3 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.

## 22.11 Прерывания

В модуле предусмотрено 11 маскируемых источников прерывания. В результате формируется один общий сигнал, представляющий собой комбинацию независимых сигналов, объединенных по схеме ИЛИ.

Сигналы запроса на прерывание:

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

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски UARTIMSC. Установка бита в 1 разрешает соответствующее прерывание, в 0 – запрещает.

Доступность, как индивидуальных линий, так и общей линии запроса позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика UARTRXINTR и UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Прерывание по обнаружению ошибке UARTEINTR формируется в случае возникновения той или иной ошибки приема данных. Предусмотрен ряд условий формирования признака ошибки.

Прерывание по состоянию модема представляет собой комбинацию признаков изменения отдельных линий состояния модема.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний UARTRIS, либо из маскированного регистра прерываний UARTMIS.

### 22.11.1 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 UARTICR.

### 22.11.2 UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен и его заполнение достигло заданного порогового значения. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения данных из буфера приемника до тех пор, пока его заполнение не станет меньше порога, либо после сброса прерывания;

- буфер FIFO запрещен (имеет размер один символ), принят один символ данных. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения одного байта данных, либо после сброса прерывания.

### 22.11.3 UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен и его заполнение меньше или равно заданному пороговому значению. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи данных в буфера передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания;

- буфер FIFO запрещен (имеет размер один символ), данные в буферном регистре передатчика отсутствуют. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи одного байта данных, либо после сброса прерывания.

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

Примечание – Прерывание передатчика основано на переходе через пороговое значение, а не на состоянии заполненности буфера FIFO передатчика относительно порогового значения.



#### 22.11.4 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 UARTICR.

#### 22.11.5UARTEINTR

Прерывание по обнаружению ошибки возникает в случае ошибки при приеме данных. Оно может быть вызвано рядом факторов:

- ошибка в структуре кадра;
- ошибка контроля четности;
- разрыв линии;
- переполнение буфера.

Причину возникновения прерывания можно определить, прочитав содержимое регистра прерываний UARTRIS, либо содержимое маскированного регистра прерываний UARTRMIS.

Сброс прерывания осуществляется путем записи соответствующих бит в регистр сброса прерывания UARTICR. За прерываниями по обнаружению ошибки закреплены биты с 7 по 10.

#### 22.11.6 UARTINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов UARTRXINTR, UARTRTXINTR, UARTRTINTR, UARTRMSINTR и UARTEINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

### 22.12 Программное управление модулем

#### 22.12.1 Общая информация

Следующая информация применима ко всем регистрам контроллера.

Базовый адрес контроллера фиксирован. Смещение каждого регистра относительно базового адреса постоянно.

Не следует пытаться получить доступ к зарезервированным или неиспользуемым адресам. Это может привести к непредсказуемому поведению модуля.

За исключением специально оговоренных в настоящем документе случаев:

- не следует изменять значения не определенных в документе разрядов регистров;
- не следует использовать значения не определенных в документе разрядов регистров;

– все биты регистров (за исключением специально оговоренных случаев) устанавливаются в значение «0» после сброса по включению питания или системного сброса.

Столбец «Тип» в таблице 239 определяет режим доступа к регистру в соответствии с обозначениями:

- RW – чтение и запись;
- RO – только чтение;
- WO – только запись.

### 22.13 Обобщенные данные о регистрах контроллеров UART

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в таблице 239.

Таблица 239 – Обобщенные данные о регистрах контроллеров UART

Базовый адрес	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x4000_8000	UART0				Контроллер UART0
0x4001_0000	UART1				Контроллер UART1
Смещение					
0x000	DR	RW	0x---	12/8	Регистр данных
0x004	RSR_ECR	RW	0x0	4/0	Регистра состояния приемника / Сброс ошибки приемника
0x008-0x014					Резерв
0x018	FR	RO	0b-10010---	9	Регистр флагов
0x01C					Резерв
0x020	ILPR	RW	0x00	8	Регистр управления ИК-обменом в режиме пониженного энергопотребления
0x024	IBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	FBRD	RW	0x00	6	Дробная часть делителя скорости обмена данными
0x02C	LCR_H	RW	0x00	8	Регистр управления линией
0x030	CR	RW	0x0300	16	Регистр управления
0x034	IFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	IMSC	RW	0x000	11	Регистр маски прерывания
0x03C	RIS	RO	0x00-	11	Регистр состояния прерываний
0x040	MIS	RO	0x00-	11	Регистр состояния прерываний с маскированием
0x044	ICR	WO	-	11	Регистр сброса прерывания
0x080	UARTTCR	RW	0x00	3	Регистр управления тестированием

### 22.13.1 DR

Регистр данных.

*В ходе передаче данных:*

Если буфер FIFO передатчика разрешен, то слово данных, записанные 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 UARTDR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра RSR\_ECR (таблица 241).

Таблица 240 – Формат регистра DR

Бит	Наименование	Назначение
15:12		Резерв
11	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время, как буфер заполнен. Сбрасывается в 0 после того, как в буфере появится свободное место
10	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита
9	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией UARTLCR_H. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер

Бит	Наименование	Назначение
8	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер
7:0	DATA	Принимаемые данные (чтение). Передаваемые данные (запись)

Примечание – Необходимо запрещать работу приемопередатчика перед любым перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние во время передачи или приема символа, то перед остановкой он завершает выполняемую операцию.

### 22.13.2 RSR\_ECR

Регистр состояния приемника/сброса ошибки.

Состояние приемника также может быть считано из регистра RSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра относится к последнему символу, считанному из регистра данных DR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним считанным из регистра DR байтом данных).

Запись в регистр ECR приводит к сбросу признаков ошибок переполнения, четности, структуры кадра, разрыва линии. Кроме того, все эти признаки устанавливаются в 0 после сброса устройства.

В таблице 241 представлено назначение бит регистра RSR\_ECR.

Таблица 241 – Регистр RSR\_ECR

Биты	Наименование	Назначение
7:0		Запись в регистр сбрасывает признаки ошибок формирования кадра, проверки на четность, разрыва линии и переполнения буфера
7:4		Резерв, при чтении результат не определен
3	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время, как буфер заполнен. Сбрасывается в 0 после записи в регистр UARTECR. 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 UARTECR. При

Биты	Наименование	Назначение
		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.

### 22.13.3 FR

Регистр флагов.

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. В таблице 242 представлена информация о назначении битов регистра.

Таблица 242 – Регистр FR

Биты	Наименование	Назначение
15:9		Резерв. Не модифицируйте. При чтении заполняются нулями
8	RI	Инверсия линии nUARTRI
7	TXFE	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией 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

Биты	Наименование	Назначение
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 равен 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

#### 22.13.4 ILPR

Регистр управления ИК-обменом в режиме пониженного энергопотребления.

Этот восьмиразрядный регистр, доступный для чтения и записи, содержит значение коэффициента деления частоты UARTCLK, для формирования тактового сигнала IrLPBaud16. Назначение разрядов регистра показано в таблице 243.

Требуемое значение коэффициента деления для формирования сигнала IrLPBaud16 вычисляется по формуле:  $ILPDVSR = F\_UARTCLK / F\_IrLPBaud16$ , где номинальное значение частоты  $F\_IrLPBaud16$  составляет 1,8432 МГц.

Коэффициент деления должен быть установлен таким образом, чтобы выполнялось соотношение:  $1,42 \text{ МГц} < F\_IrLPBaud16 < 2,12 \text{ МГц}$ , что, в свою очередь, гарантирует формирование кодеком импульсов данных с длительностью 1,41-2,11 мкс (в три раза длиннее периода сигнала IrLPBaud16).

Таблица 243 – Регистр ILPR

Биты	Наименование	Назначение
7:0	ILPDVSR	Коэффициент деления частоты UARTCLK, для формирования тактового сигнала IrLPBaud16. После сброса устанавливается в 0 Примечание – Коэффициент 0 – запрещенное значение. В случае его установки импульсы IrLPBaud16 формироваться не будут

Примечание – В интересах подавления помех, при работе в режиме IrDA с пониженным энергопотреблением кодек игнорирует поступающие на вход SIRIN импульсы с длительностью, меньшей трех периодов сигнала IrLPBaud16.

### 22.13.5 IBRD

Регистр целой части делителя скорости передачи данных.

Назначение бит регистра представлено в таблице 244.

Таблица 244 – Регистр IBRD

Биты	Наименование	Назначение
15:0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

### 22.13.6 FBRD

Регистр дробной части делителя скорости передачи данных.

Назначение бит регистра представлено в таблице 245.

Таблица 245 – Регистр BFRD

Биты	Наименование	Назначение
5:0	BAUDDIV_FRAC	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

Коэффициент деления вычисляется по формуле:

$$BAUDDIV = FUARTCLK / (16 \cdot \text{Baud\_rate}) ,$$

где FUARTCLK – тактовая частота контроллера UART, Baud\_rate – требуемая скорость передачи данных.

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUDDIV\_INT и BAUDDIV\_FRAC, соответственно.

#### Примечания

1 Изменение содержимого регистров IBRD и FBRD вступают в силу только после завершения передачи и приема текущего символа данных.

2 Минимальный допустимый коэффициент деления – 1, максимальный – 65535 ( $2^{16}-1$ ). Таким образом, значение IBRD, равное 0 является недопустимым, при этом значение регистра FBRD игнорируется.

3 Аналогично, при IBRD равном 65535 (0xFFFF), значение FBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

Далее приведен пример вычисления коэффициента деления.

#### Пример. Вычисление коэффициента деления.

Пусть требуемая скорость передачи данных составляет 230400 бит/с, частота тактового сигнала UARTCLK равна 4 МГц. Тогда:

$$\text{Коэффициент деления} = (4 \cdot 10^6) / (16 \cdot 230400) = 1,085.$$

Таким образом, BRDI = 1, BRDF = 0,085.

Следовательно, значение, записываемое в регистр BFRD, равно

$$m = \text{integer}((0,085 \cdot 64) + 0,5) = 5 .$$

Реальное значение коэффициента деления =  $1 + 5/64 = 1,078$ .

Реальная скорость передачи данных =  $(4 \cdot 10^6) / (16 \cdot 1,078) = 231911$  бит/с.

Ошибка установки скорости =  $(231911 - 230400) / 230400 \cdot 100 \% = 0,656 \%$ .

Максимальная ошибка установки скорости передачи данных с использованием шестизрядного регистра BFRD =  $1/64 \cdot 100 \% = 1,56 \%$ . Такая ошибка возникает в случае  $m = 1$ , при этом разница накапливается в течение 64 тактовых интервалов.

В таблице 246 представлены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 7,3728 МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр FBRD должен быть записан ноль.

Таблица 246 – Коэффициенты деления для типичных скоростей передачи данных при частоте UARTCLK = 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 247 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц.

Таблица 247 – Коэффициенты деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц

Целая часть	Дробная часть	Требуемая скорость	Реальная скорость	Ошибка, %
0x001	0x05	230400	231911	0,656
0x002	0x0B	115200	115101	0,086
0x003	0x10	76800	76923	0,160
0x006	0x21	38400	38369	0,081
0x011	0x17	14400	14401	0,007
0x068	0x0B	2400	2400	~0
0x8E0	0x2F	110	110	~0



### 22.13.7 LCR\_H

Регистр управления линией.

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

Назначение разрядов регистра описано в таблице 248.

Таблица 248 – Регистр LCR\_H

Биты	Наименование	Назначение
15:8		Резерв. Не модифицируйте. При чтении выдаются нули.
7	SPS	Передача бита четности с фиксированным значением. 0 – запрещена; 1 – на месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. (При EPS=0 на месте бита четности передается 1, при EPS=1 – передается 0). Значение бита SPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещены
6:5	WLEN	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0b11 – 8 бит; 0b10 – 7 бит; 0b01 – 6 бит; 0b00 – 5 бит
4	FEN	Разрешение работы буфера FIFO приемника и передатчика 0 – запрещено; 1 – разрешено
3	STP2	Режим передачи двух стоповых бит: 0 – один стоповый бит; 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре
2	EPS	Четность/нечетность: 0 – бит четности дополняет количество единиц в информационной части кадра до нечетного; 1 – до четного числа. Значение бита EPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено
1	PEN	Разрешение проверки четности: 0 – кадр не содержит бита четности; 1 – бит четности передается в кадре и проверяется при приеме данных
0	BRK	Разрыв линии. Если этот бит установлен в 1, то по завершении передачи текущего символа на выходе UARTTXD 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 30-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.

**Примечания**

1 Изменение значений трех регистров можно осуществить корректно двумя способами:

- Запись IBRD, запись FBRD, запись LCR\_H;
- Запись FBRD, запись IBRD, запись LCR\_H.

2 Для того, чтобы изменить значение лишь одного из регистров (IBRD или FBRD) необходимо выполнить следующие шаги:

- Запись IBRD (или FBRD), запись LCR\_H.

В таблице 249 приведена таблица истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией LCR\_H.

Таблица 249 – Управление режимом контроля четности

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

**Примечания**

1 Регистры LCR\_H, IBRD, and FBRD не должны изменяться:

- При разрешенной работе приемопередатчика;
- Во время завершения приема или передачи данных в процессе остановки (перевода в запрещенное состояние) приемопередатчика.

2 Целостность данных в буферах FIFO не гарантируется в следующих случаях:

- После установки бита разрыва линии BRK;
- Если программное обеспечение произвело остановку приемопередатчика при наличии данных в буферах FIFO, после его повторного перевода в разрешенное состояние.

### 22.13.8 CR

Регистр управления.

После сброса все биты регистра управления, за исключением битов 9 и 8 устанавливаются в нулевое состояние. Биты 9 и 8 устанавливаются в единичное состояние.

Назначение разрядов регистра управления показано в таблице 250.

Таблица 250 – Регистр управления CR

Биты	Наименование	Назначение
15	CTSEn	Разрешение управления потоком данных по CTS. 1 – разрешено, данные передаются в линию только при активном значении сигнала nUARTCTS
14	RTSEn	Разрешение управления потоком данных по RTS. 1 – разрешено, запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере FIFO приемника
13	Out2	Инверсия сигнала на линии состояния модема nUARTOut2. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «сигнал вызова» (RI)
12	Out1	Инверсия сигнала на линии состояния модема nUARTOut1. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «обнаружен информационный сигнал» (DCD)
11	RTS	Инверсия сигнала на линии состояния модема nUARTRTS
10	DTR	Инверсия сигнала на линии состояния модема nUARTDTR
9	RXE	Прием разрешен. Установка бита в 1 разрешает работу приемника. Прием данных осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК-обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчика в запрещенное состояние в ходе приема данных, он завершает прием текущего символа перед остановкой
8	TXE	Передача разрешена. Установка бита в 1 разрешает работу передатчика. Передача осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК-обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчик в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой
7	LBE	1 – шлейф разрешен; 0 – запрещен. В режиме разрешенного шлейфа: Если установлены бит SIREN=1 и бит регистра управления тестированием TCR SIRTEST=1, то сигнал с выхода кодека 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 UARTTXD коммутируется на вход приемника UARTRXD. Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии. После сброса бит устанавливается в 0

Биты	Наименование	Назначение
6:3		Резерв. Не модифицируйте. При чтении выдаются нули
2	SIRLP	Выбор режима ИК обмена с пониженным энергопотреблением: 0 – длительность импульсов данных равна 3/16 длительности передачи бита. 1 – длительность импульсов данных равна трем тактам сигнала IrLPBaud16 вне зависимости от выбранной скорости передачи данных. Выбор этого режима снижает энергопотребление, однако может привести к уменьшению дальности связи
1	SIREN	Разрешение работы кодека ИК передачи данных IrDA SIR: 0 – запрещен. Сигнал nSIROUT находится в низком состоянии, данные на входе SIRIN не обрабатываются. 1 – разрешен. Данные передаются на выход nSIROUT и принимаются с входа SIRIN. Линия UARTTXD находится в высоком состоянии. Данные на входе UARTRXD и линиях состояния модема не обрабатываются. В случае, если UARTEN=0 значение бита не играет роли
0	UARTEN	Разрешение работы приемопередатчика: 0 – работа запрещена. Перед остановкой завершается прием и/или передача обрабатываемого в текущий момент символа. 1 – работа разрешена. Производится обмен данными либо по линиям асинхронного обмена, либо по линиям ИК обмена SIR, в зависимости от состояния бита SIREN
<p><b>Примечания</b></p> <p>1 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.</p> <p>2 Рекомендуется следующая последовательность действий для программирования регистров управления:</p> <ul style="list-style-type: none"> <li>– Остановите работу приемопередатчика;</li> <li>– Дождитесь окончания приема и/или передачи текущего символа данных;</li> <li>– Сбросьте буфер передатчика путем установки бита FEN регистра LCR_N в 0;</li> <li>– Изменить настройки регистра CR;</li> <li>– Возобновите работу приемопередатчика</li> </ul>		

### 22.13.9 IFLS

Регистр порога прерывания по заполнению буфера FIFO.

Данный регистр используется для установки порогового значения заполнения буферов передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода величины заполнения буфера через заданное значение.

После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра и значения его битов представлены в таблице 251.

Таблица 251 – Регистр IFLS

Биты	Наименование	Назначение
15:6		Резерв. Не модифицируйте. При чтении выдаются нули
5:3	RXIFLSEL	Порог прерывания по заполнению буфера приемника: b000 = Буфер заполнен на 1/8; b001 = Буфер заполнен на 1/4; b010 = Буфер заполнен на 1/2; b011 = Буфер заполнен на 3/4; b100 = Буфер заполнен на 7/8; b101-b111 = резерв
2:0	TXIFLSEL	Порог прерывания по заполнению буфера передатчика: b000 = Буфер заполнен на 1/8; b001 = Буфер заполнен на 1/4; b010 = Буфер заполнен на 1/2; b011 = Буфер заполнен на 3/4; b100 = Буфер заполнен на 7/8; b101-b111 = резерв

### 22.13.10 IMSC

Регистр установки сброса маски прерывания.

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение битов регистра IMSC показано в таблице 252.

Таблица 252 – Регистр IMSC

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули
10	OEIM	Маска прерывания по переполнению буфера UARTOEINTR: 1 – установлена; 0 – сброшена
9	BEIM	Маска прерывания по разрыву линии UARTBEINTR: 1 – установлена; 0 – сброшена
8	PEIM	Маска прерывания по ошибке контроля четности UARTPEINTR: 1 – установлена; 0 – сброшена.
7	FEIM	Маска прерывания по ошибке в структуре кадра UARTFEINTR: 1 – установлена; 0 – сброшена
6	RTIM	Маска прерывания по таймауту приема данных UARTRTINTR: 1 – установлена; 0 – сброшена
5	TXIM	Маска прерывания от передатчика UARTTXINTR: 1 – установлена; 0 – сброшена

Биты	Наименование	Назначение
4	RXIM	Маска прерывания от приемника UARTRXINTR: 1 – установлена; 0 – сброшена
3	DSRMIM	Маска прерывания UARTDSRINTR по изменению состояния линии nUARTDSR: 1 – установлена; 0 – сброшена
2	DCDMIM	Маска прерывания UARTDCDINTR по изменению состояния линии nUARTDCD: 1 – установлена; 0 – сброшена
1	CTSMIM	Маска прерывания UARTCTSINTR по изменению состояния линии nUARTCTS: 1 – установлена; 0 – сброшена
0	RIMIM	Маска прерывания UARTRIINTR по изменению состояния линии nUARTRI: 1 – установлена; 0 – сброшена

### 22.13.11 RIS

Регистр состояния прерываний.

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Предупреждение. После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре RIS представлено в таблице 253.

Таблица 253 – Регистр RIS

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули
10	OERIS	Состояние прерывания по переполнению буфера UARTOEINTR
9	BERIS	Состояние прерывания по разрыву линии UARTBEINTR
8	PERIS	Состояние прерывания по ошибке контроля четности UARTPEINTR
7	FERIS	Состояние прерывания по ошибке в структуре кадра UARTFEINTR
6	RTRIS	Состояние прерывания по таймауту приема данных UARTRTINTR.  Примечание – Бит RTRIS может быть установлен только при установленной маске прерывания по таймауту приема данных UARTRTINTR в регистре IMSC. Это вызвано тем, что сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Чтение состояния прерывания по таймауту из регистров MIS и RIS приводит к одинаковым результатам

Биты	Наименование	Назначение
5	TXRIS	Состояние прерывания от передатчика UARTTXINTR.
4	RXRIS	Состояние прерывания от приемника UARTRXINTR.
3	DSRRMIS	Состояние прерывания UARTDSRINTR по изменению линии nUARTDSR
2	DCDRMIS	Состояние прерывания UARTDCDINTR по изменению линии nUARTDCD
1	CTSRMIS	Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS
0	RIRMIS	Состояние прерывания UARTRIINTR по изменению линии nUARTRI

### 22.13.12 MIS

Регистр маскированного состояния прерываний.

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре MIS представлено в таблице 254.

Таблица 254 – Регистр MIS

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули
10	OEMIS	Маскированное состояние прерывания по переполнению буфера UARTOEINTR
9	BEMIS	Маскированное состояние прерывания по разрыву линии UARTBEINTR
8	PEMIS	Маскированное состояние прерывания по ошибке контроля четности UARTPEINTR
7	FEMIS	Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR
6	RTMIS	Маскированное состояние прерывания по таймауту приема данных UARTRTINTR
5	TXMIS	Маскированное состояние прерывания от передатчика UARTTXINTR
4	RXMIS	Маскированное состояние прерывания от приемника UARTRXINTR
3	DSRMMIS	Маскированное состояние прерывания UARTDSRINTR по изменению линии nUARTDSR
2	DCDMMIS	Маскированное состояние прерывания UARTDCDINTR по изменению линии nUARTDCD
1	CTSMMIS	Маскированное состояние прерывания UARTCTSINTR по изменению линии nUARTCTS
0	RIMMIS	Маскированное состояние прерывания UARTRIINTR по изменению линии nUARTRI

### 22.13.13 ICR

Регистр сброса прерываний.

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись в любой из разрядов регистра 0 игнорируется.

Назначение бит в регистре ICR представлено в таблице 255.

Таблица 255 – Регистр ICR

Биты	Наименование	Назначение
15:11		Резерв. Не модифицируйте. При чтении выдаются нули
10	OEIC	Сброс прерывания по переполнению буфера UARTOEINTR
9	BEIC	Сброс прерывания по разрыву линии UARTBEINTR
8	PEIC	Сброс прерывания по ошибке контроля четности UARTPEINTR
7	FEIC	Сброс прерывания по ошибке в структуре кадра UARTFEINTR
6	RTIC	Сброс прерывания по таймауту приема данных UARTRTINTR.
5	TXIC	Сброс прерывания от передатчика UARTTXINTR
4	RXIC	Сброс прерывания от приемника UARTRXINTR
3	DSRMIC	Сброс прерывания UARTDSRINTR по изменению линии nUARTDSR
2	DCDMIC	Сброс прерывания UARTDCDINTR по изменению линии nUARTDCD
1	CTSMIC	Сброс прерывания UARTCTSINTR по изменению линии nUARTCTS
0	RIMIC	Сброс прерывания UARTRIINTR по изменению линии nUARTRI

### 22.13.14 UARTTCR

Регистр управления тестированием.

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра UARTTCR представлено в таблице 256.

Таблица 256 – Регистр UARTTCR

Биты	Наименование	Назначение
15:13		Резерв. При чтении значение непредсказуемо
2	SIRTEST	Разрешение приема данных в кольцевом режиме с выхода IrDA передатчика. 1 – разрешено; 0 – запрещено. Используется совместно с установкой бита LBE в регистре UART_CR
1	TESTFIFO	Разрешение чтения данных из FIFO передатчика и запись в FIFO приемника. 1 – разрешено; 0 – запрещено
0	ITEN	Перевод контроллера UART в тестовый режим 1 – тестовый режим разрешен; 0 – тестовый режим запрещен



## 23 Контроллер LCD

Контроллер LCD – это периферийный блок, который сопрягается с любым жидкокристаллическим дисплеем, имеющим низкую величину мультиплекса. Контроллер обеспечивает генерацию сигналов управления как для статических, так и для LCD с мультиплексом содержащих до четырех общих выводов и до тридцати сегментов. Все возможные конфигурации дисплеев приведены в таблице 257.

Таблица 257 – Конфигурации дисплеев

Количество активных общих выводов	Число сегментов	Число 7-сегментных цифр	Число 14-сегментных букв	Число точек
4	120	15 цифр + 15 символьных обозначений	8 букв + 8 символьных обозначений	120 (4x30)
3	90	11 цифр + 11 символьных обозначений	6 букв + 6 символьных обозначений	90 (3x30)
2	60	7 цифр + 7 символьных обозначений	4 буквы + 4 символьных обозначения	60 (2x30)
1	30	3 цифры + 3 символьных обозначения	2 буквы + 2 символьных обозначения	30

Структурная схема контроллера LCD приведена на рисунке 88.

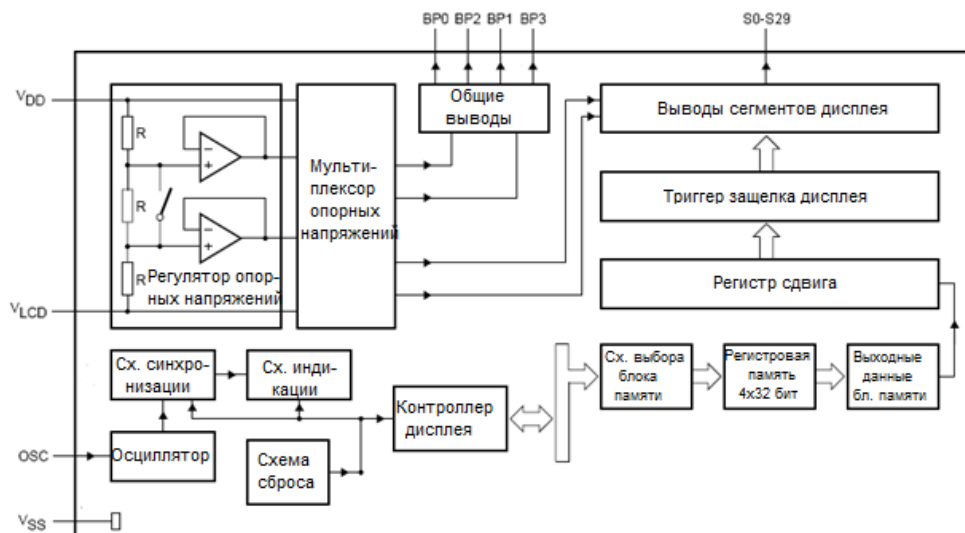


Рисунок 88 – Структурная схема контроллера LCD

Соответствующее напряжение смещения для LCD с мультиплексом генерируется внутри аналоговой площадки. Для подключения соответствующего дисплея необходимы только питающие напряжения площадки ( $V_{DD}$ ,  $V_{SS}$  и  $V_{LCD}$ ).

После сброса или при отключении контроллера все выходы сегментов и общие устанавливаются в состояние  $V_{DD}$ , схема управления выбирается исходя из мультиплекса 1:4 и 1/3 bias. Мерцание дисплея отключено. Полная шкала напряжения LCD ( $V_{op}$ ) равна  $V_{DD} - V_{LCD}$ . Для температурной компенсации напряжение LCD может подаваться через



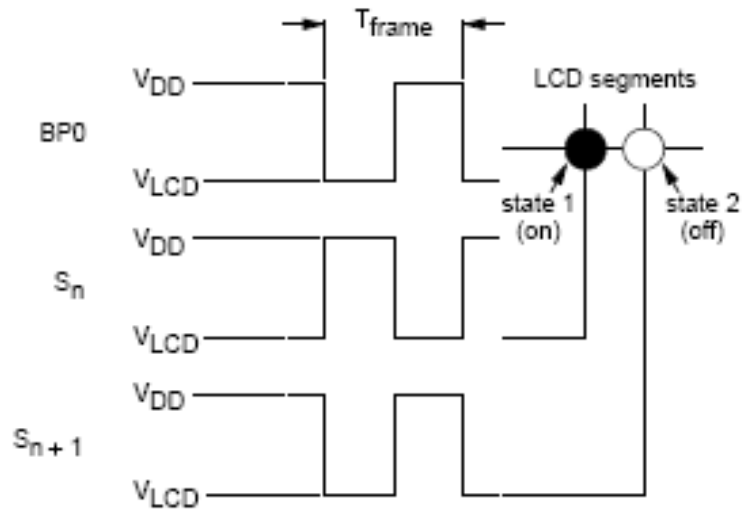


Рисунок 90 – Статический режим управления

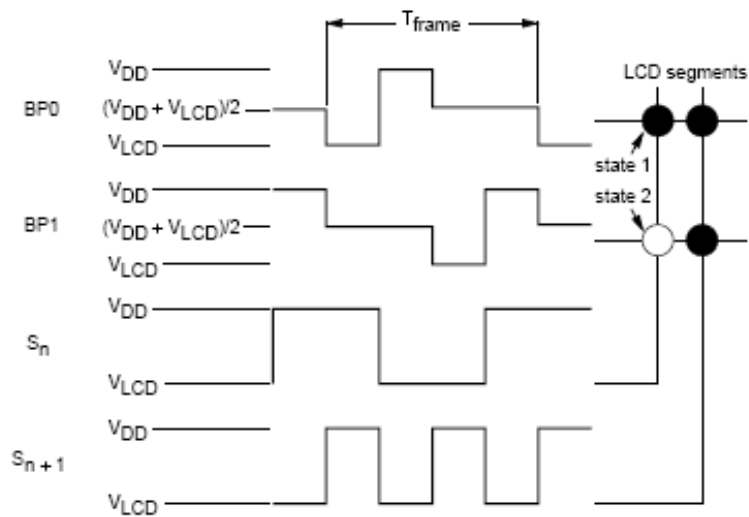


Рисунок 91 – Режим управления с мультиплексом 1:2 и 1/2 bias

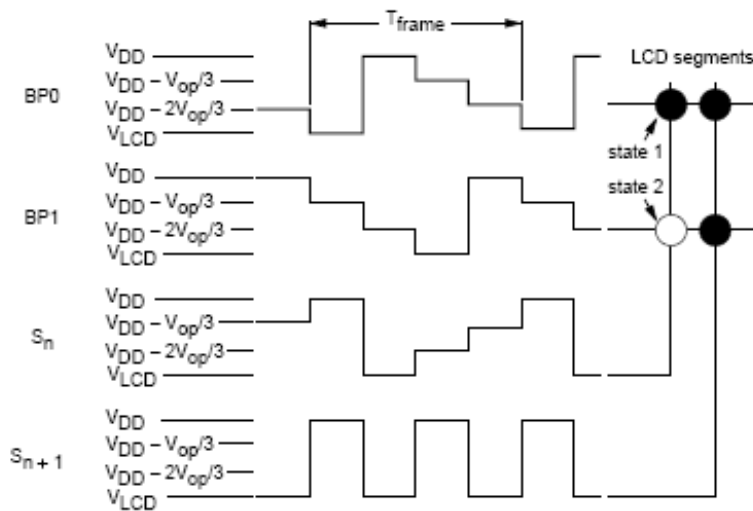


Рисунок 92 – Режим управления с мультиплексом 1:2 и 1/3 bias

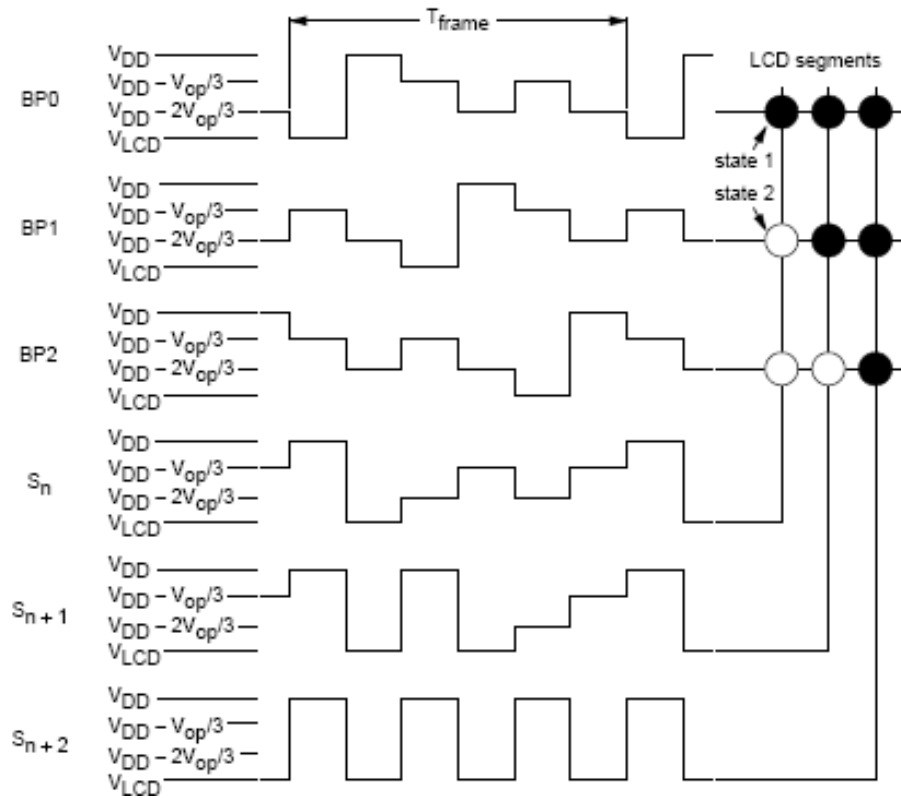


Рисунок 93 – Режим управления с мультиплексом 1:3

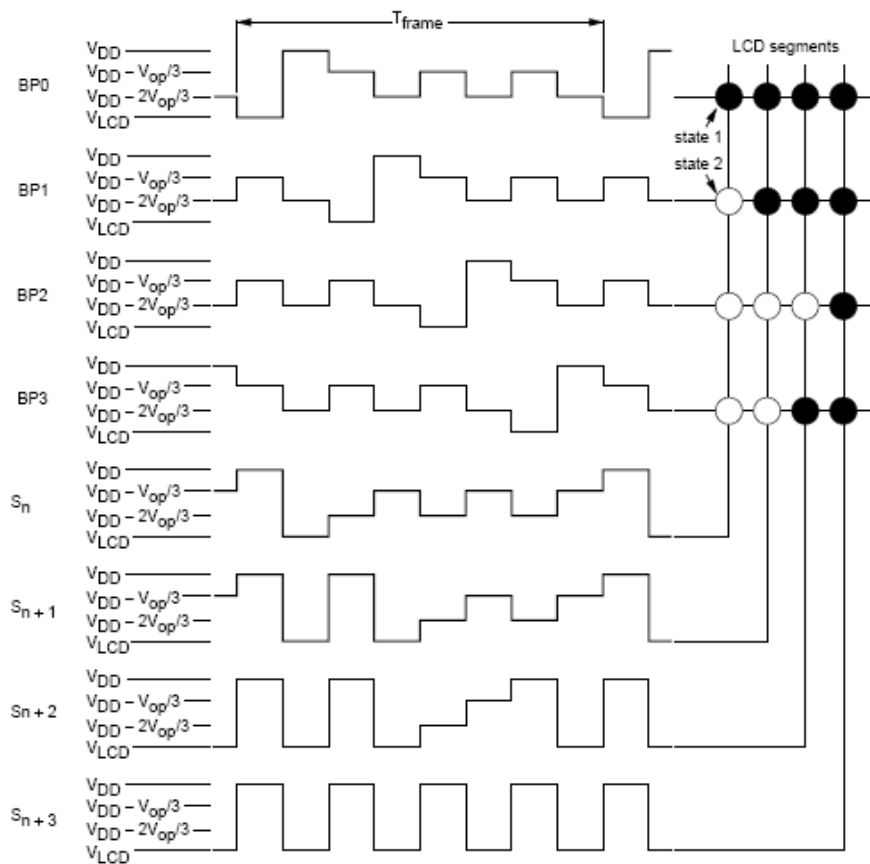


Рисунок 94 – Режим управления с мультиплексом 1:4

### 23.2 Описание регистров контроллера LCD

Таблица 258 – Перечень регистров контроллера LCD

Базовый адрес	Название	Описание
0x4003_8000	LCD	Контроллер LCD дисплея
Смещение		
0x00	CR	Регистр управление контроллером
0x04	ROW1	Строка данных основной памяти для общего выхода BP0
0x08	ROW2	Строка данных основной памяти для общего выхода BP1
0x0C	ROW3	Строка данных основной памяти для общего выхода BP2
0x10	ROW4	Строка данных основной памяти для общего выхода BP3
0x14	AROW1	Строка данных альтернативной памяти для общего выхода BP0
0x18	AROW2	Строка данных альтернативной памяти для общего выхода BP1
0x1C	AROW3	Строка данных альтернативной памяти для общего выхода BP2
0x20	AROW4	Строка данных альтернативной памяти для общего выхода BP3

#### 23.2.1 CR. Регистр управления

Таблица 259 – Регистр CR

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	1	1
	BF1	BF0	E	B	M1	M0

Номер	31...25	24	23...8	7	6
Доступ*	U	R/W	R/W	R/W	R/W
Сброс		0	0	0	0
	-	CLK_SEL	DIV[15:0]	VLCDON	A

\* Обозначения:

R/W – бит доступен на чтение и запись

RO – бит доступен только на чтение

U – бит физически не реализован или зарезервирован

Таблица 260 – Описание битов регистра CR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..25		Зарезервировано
24	CLK_SEL	Выбор частоты синхронизации блока: 1 – частота LSI; 0 – частота LSE
23..8	DIV[15:0]	Делитель опорной частоты осциллятора 32,768 кГц. Значение делителя определяет длительность фрейма для соответствующей схемы управления. Выбирается с учетом следующих соотношений: 1:1 MUX $f_{frame}=32,768 / DIV$ ;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		1:2 MUX $f_{\text{frame}} = 32,768 / 2 \cdot \text{DIV}$ ; 1:3 MUX $f_{\text{frame}} = 32,768 / 3 \cdot \text{DIV}$ ; 1:4 MUX $f_{\text{frame}} = 32,768 / 4 \cdot \text{DIV}$ ;
7	VLCDON	Выбор источника питания $V_{\text{LCD}}$ : 1 – питания $V_{\text{LCD}}$ формируется на внешней площадке; 0 – питание $V_{\text{LCD}}$ формируется внутренним источником опорного напряжения
6	A	Выбор схемы мерцания LCD: 1 – альтернативная схема мерцания (из альтернативной памяти); 0 – нормальная схема мерцания (гашения с частотой, заданной битами BF). Если BF=0, то мерцание отключено
5, 4	BF[1:0]	Разрешение мерцания LCD: 0 – мерцание запрещено; 1 – мерцание с частотой 2 Гц; 2 – мерцание с частотой 1 Гц; 3 – мерцание с частотой 0,5 Гц
3	E	Разрешение работы контроллера: 1 – контроллер включен; 0 – контроллер отключен
2	B	Выбор режима bias: 0 – 1/3 bias; 1 – 1/2 bias
1, 0	M[1:0]	Выбор схемы управления: 0 – статическое управление; 1 – 1:2 MUX; 2 – 1:3 MUX; 3 – 1:4 MUX

### 23.2.2 ROW4-ROW1

Таблица 261 – Регистры данных основной памяти

Номер	31, 30	29...0
Доступ*	U	R/W
Сброс		0
	-	S29-S0

### 23.2.3 AROW4-AROW1

Таблица 262 – Регистры данных альтернативной памяти

Номер	31, 30	29...0
Доступ*	U	R/W
Сброс		0
	-	S29-S0

## 24 Прерывания и исключения

Состояние исключений:

- Inactive – исключение не находится в стадии Active или Pending.
- Pending – исключение находится в состоянии ожидания обработки процессором. Запрос прерывания от периферийных блоков или программы может изменить состояние соответствующего прерывания на состояние pending.
- Active – исключение начало обрабатываться процессором, но еще не закончено. Обработчик исключения может быть прерван другим обработчиком исключения. В этом случае оба исключения находятся в состоянии Active.
- Active и Pending – исключения начало обрабатываться процессором, но появилось новое исключение в состоянии pending от того же источника.

### 24.1 Типы исключений

Исключения бывают следующих типов

#### 24.1.1 RESET

RESET вызывается при включении питания и горячем сбросе. Модель исключений трактует RESET как специальная форма исключения. Когда выставляется RESET, работа процессора останавливается потенциально в любой точке инструкций. Когда RESET убирается, выполнение перезапускается с адреса заданном в таблице векторов для сброса. Выполнение перезапускается в thread режиме.

#### 24.1.2 NON MASKABLE INTERRUPT (NMI)

Не маскируемое прерывание (NMI) может быть вызвано периферией или установлено программой. Это самое высокоприоритетное исключение после сброса. Всегда разрешено и имеет фиксированный приоритет -2.

NMI не может быть:

- замаскировано или предотвращено от активации из другого исключения;
- прерывает любые исключения кроме RESET.

#### 24.1.3 HARD FAULT

Hard Fault исключение происходит при ошибке при обработке исключений или потому что исключение не может быть обработано каким либо другим механизмом. Hard fault имеет фиксированный приоритет -1, означающий, что он имеет больший приоритет чем любое из исключений с конфигурируемым приоритетом.

#### 24.1.4 SVCALL

Исключение Supervisor Call (SVCALL) возникает при выполнении инструкции SVC. В приложениях с использованием Операционных Сред инструкция SVC может использоваться для доступа к функциям ОС и драйверам устройств.

### 24.1.5 PendSV

PendSV является прерыванием запросом сервисов системного уровня. В приложениях с использованием ОС PendSV используется для переключения контекстов, когда нет других активных исключений.

### 24.1.6 SysTick

Исключение SysTick является исключением генерируемым системным таймером, когда он обнуляется. Программное обеспечение также может генерировать исключение SysTick. В приложениях с использованием ОС процессор может использовать это исключение для подсчета системных циклов

## 24.2 Прерывания (IRQ)

Прерывания или IRQ – это исключения, вызываемые периферийными устройствами или программными запросами. Все прерывания асинхронны по отношению к выполняемым инструкциям. В системе прерывания используются для коммуникации периферии и процессора

Таблица 263 – Таблица различных типов исключений

Номер исключения	IRQ номер	Тип	Приоритет	Адрес вектора обработчика (смещение)	Активация
1	-	RESET	-3, наивысший	0x0000_0004	Асинхронный
2	-14	NMI	-2	0x0000_0008	Асинхронный
3	-13	Hard Fault	-1	0x0000_000C	-
4-10	-	Reserved	-	Зарезервировано	-
11	-5	SVCall	Конфигурируемый	0x0000_002C	Синхронный
12-13	-	-	-	Зарезервировано	-
14	-2	PendSV	Конфигурируемый	0x0000_0038	Асинхронный
15	-1	SysTick	Конфигурируемый	0x0000_003C	Асинхронный
16 и выше	0 и выше	IRQ	Конфигурируемый	0x0000_0040 и выше	Асинхронный

Для асинхронных исключений, кроме RESET, процессор может выполнить другие инструкции между возникновением сигнала исключения и входом в обработчик.

Программа в Privileged режиме может запретить прерывания, имеющие конфигурируемый приоритет.

Таблица 264 – Прерывания, формируемые периферийными блоками

Прерывания	Блок	Принцип формирования
IRQ1	UART0	Сигнал UARTINTR
IRQ2	UART1	Сигнал UARTINTR
IRQ3	SSP	Сигнал SSPINTR
IRQ4	POWER	Сигнал прерывания от POWER Detecor



Прерывания	Блок	Принцип формирования
IRQ5	WWDG	Сигнал прерывания от WWDG
IRQ6	TIMER0	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
IRQ7	TIMER1	Аналогично
IRQ8	ADC	Сигналы прерываний от АЦП EOCIF_1 или AWOIF_1 или EOCIF_2 или AWOIF_2
IRQ9	I2C	Сигнал INT при EN_INT
IRQ10	BACKUP	Прерывание от ВКР и часов реального времени
IRQ11	Внешнее прерывание 1	Сигнал EXT_INT0 Вывод PC[7] в основном режиме
IRQ12	Внешнее прерывание 2	Сигнал EXT_INT1 Вывод PC[8] в основном режиме
IRQ13	Внешнее прерывание 3	Сигнал EXT_INT2 Вывод PC[9] в основном режиме
IRQ14	ADCUI	Прерывание от АЦП для измерения напряжений и токов

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

### 24.3 Обработчики исключений

Для обработки исключений используются:

#### Процедуры обработки прерываний (Interrupt Service Routines – ISRs)

Прерывания с IRQ0 по IRQ31 обрабатываются ISRs.

#### Обработчики ошибок (Fault Handlers)

Обрабатываются только исключения Hard Fault.

#### Системные обработчики (System handlers)

Обрабатываются исключения NMI, PendSV, SVCall, SysTick и и Hard Fault.

#### Таблица векторов

Таблица векторов содержит указатель стека, вектор входа по RESET и стартовые адреса обработчиков, также называемых векторами. На рисунке 95 представлена последовательность векторов в таблице. Младший бит всех векторов должен быть равен 1, указывая на то, что обработчик выполняется в Thumb-режиме.

Номер исключения	номер IRQ	Вектор	Смещение
16+n	n	IRQn	0x40+4n
.	.	.	.
.	.	.	.
18	2	IRQ2	0x48
17	1	IRQ1	0x44
16	0	IRQ0	0x40
15	-1	если применяется SysTick	0x3C
14	-2	PendSV	0x38
13		Зарезервировано	
12			
11	-5	SVCcall	0x2C
10			
9			
8			
7			
6			
5			
4			
3	-13	HardFault	0x10
2	-14	NMI	0x0C
		Reset	0x08
1		Начальное значение указателя стека	0x04
			0x00

Рисунок 95 – Таблица векторов

При системном сбросе, таблица векторов располагается по фиксированному адресу 0x00000000.

#### 24.4 Приоритеты исключений

Меньшее значение приоритета означает больший приоритет.

Конфигурируемы все приоритеты, кроме RESET и Hard Fault

Если программное обеспечение не задает приоритетов, то все они имеют приоритет 0.

Конфигурируемый приоритет может быть в диапазоне от 0 до 192 с шагом 64. Это означает, что RESET, Hard Fault и NMI имеющие отрицательное значение приоритета, всегда имеют больший приоритет.

Если имеется несколько исключений с одинаковым приоритетом, то больший приоритет имеет исключение с меньшим порядковым номером.

Если процессор выполняет обработчик исключения и происходит исключение с большим приоритетом, то происходит переход на обработчик исключения с большим приоритетом. Если при выполнении обработчика произошло исключение с таким же приоритетом, то это исключение будет выполнено по завершению текущего обработчика, несмотря на порядковый номер исключения.

## 24.5 Вход в обработчик и выход из обработчика

При описании используются следующие термины:

### 24.5.1 Приоритетное прерывание

Выполнение процессором процедуры обработки исключительной ситуации (далее по тексту – исключения), может быть прервано в случае возникновения исключения с приоритетом выше, чем у обрабатываемого. В случае, если внутри обработчика исключения возникает прерывание более высокого приоритета возникает ситуация, называемая вложенным исключением.

### 24.5.2 Возврат

Возврат из обработчика осуществляется по завершении обработки исключительной ситуации, с одновременным выполнением следующих условий:

- в системе отсутствуют необработанные исключения с достаточным приоритетом;
- завершённый обработчик не обрабатывал запоздавшее исключение (late-arriving exception).

Процессор обращается к стеку и восстанавливает состояние, имевшее место до вызова обработчика.

### 24.5.3 Передача управления без восстановления контекста (tail-chaining)

Данный механизм ускоряет процесс обработки исключений. По завершении выполнения обработчика осуществляется проверка наличия необработанных исключений и в случае, если исключения, требующие вызова обработчика, присутствуют, восстановление состояния процессора из стека не производится, а управление передается непосредственно на новый обработчик.

### 24.5.4 Запоздавшее исключение (late-arriving exception)

В случае, если во время сохранения состояния при входе в обработчик возникла исключительная ситуация с более высоким приоритетом, процессор передает управление непосредственно высокоприоритетному обработчику.

Подобный способ обработки высокоприоритетного исключения возможен до момента начала выполнения первой инструкции процедуры обработки исключительной ситуации. После возврата из обработчика запоздавшего исключения осуществляется передача управления на прерванный низкоприоритетный обработчик без восстановления контекста.

### 24.5.5 Вход в процедуру обработки исключения

Вызов процедуры обработки исключения возникает в случае наличия необработанных исключительных ситуаций с достаточным приоритетом и выполнения одного из следующих условий:

- процессор находится в режиме приложения (thread mode);
- новая исключительная ситуация имеет приоритет выше, чем обрабатываемая в текущий момент времени, что приводит к приоритетному прерыванию выполнения текущего обработчика. В этом случае возникает вложение одного исключения в другое.

Для того, чтобы исключительная ситуация имела достаточный приоритет, необходимо, чтобы уровень ее приоритета был выше значений, заданных в регистрах маскирования. В противном случае исключение находится в состоянии ожидания, процедура его обработки не вызывается.

При необходимости вызова обработчика, за исключением случаев обработки запоздавшего исключения и передачи управления на обработчик без восстановления контекста, процессор заносит в текущий стек восемь слов данных, называемые далее стековым фреймом. Этот фрейм включает в себя следующие значения:

- Регистры R0-R3, R12;
- Адрес возврата;
- Регистр PSR;
- Регистр LR.

Указанная операция далее будет называться сохранением контекста. Непосредственно после ее выполнения указатель стека равен младшему адресу стекового фрейма.

Во время сохранения контекста производится выравнивание адреса стека по границе двойного слова.

Стековый фрейм содержит адрес возврата, указывающий на ближайшую невыполненную инструкцию прерванной программы. По завершении процедуры обработки исключения значений адреса возврата заносится в счетчик команд, после чего выполнение программы возобновляется с прерванной точки.

Одновременно с сохранением контекста процессор осуществляет выборку адреса точки входа в процедуру обработки исключения из таблицы векторов исключений. По завершении операции сохранения контекста процессор передает управление на полученный из таблицы адрес.

Одновременно в регистр LR записывается значение EXC\_RETURN, позволяющее определить, какой из двух указателей стека соответствует данному стековому фрейму и в каком режиме находился процессор перед входом в обработчик.

Если во время передачи управления не возникло исключения с более высоким приоритетом, процессор начинает выполнение вызванной процедуры обработки и автоматически изменяет состояние текущего прерывания с ожидающего обработки на активное.

В противном случае процессор передает управление обработчика высокоприоритетной исключительной ситуации без изменения состояния отложенного прерывания в соответствии с правилами, изложенными в разделе «Запоздавшее исключение (late-arriving exception)».

### 24.5.6 Возврат из обработчика исключения

Возврат из обработчика исключения осуществляется в случае, если процессор находится в режиме обработчика (handler mode) и выполняет одну из следующих инструкций, позволяющих загрузить значение EXC\_RETURN в регистр PC:

- инструкцию POP с аргументом PC;
- инструкцию BX с любым регистром.

Значение EXC\_RETURN загружается в регистр LR по входу в обработчик исключения. Механизм обработки исключений использует это значение для того, чтобы определить, завершил ли процессор выполнение процедуры обработки исключительной ситуации. Младшие четыре бита EXC\_RETURN содержат информацию о состоянии стека и режиме работы процессора. Информация о назначении разрядов EXC\_RETURN[3:0] и особенности процесса возврата из обработчика исключения представлены в таблице 265.

Процессор устанавливает биты EXC\_RETURN [31:4] в 0xFFFFFFFF. Загрузка данного значения в PC указывает на завершение процедуры обработки исключения и заставляет процессор выполнить необходимые действия для возврата из обработчика.

Таблица 265 – Возврат из обработчика исключения

EXC_RETURN[3:0]	Описание
bXXX0	Резерв
b0001	Возврат в режим обработчика. Восстановление контекста осуществляется из стека MSP. Дальнейшая работа осуществляется со стеком MSP
b0011	Резерв
b01X1	Резерв
b1001	Возврат в режим приложения. Восстановление контекста осуществляется из стека MSP. Дальнейшая работа осуществляется со стеком MSP
b1101	Возврат в режим приложения. Восстановление контекста осуществляется из стека PSP. Дальнейшая работа осуществляется со стеком PSP
b1X11	Резерв

## 25 Управление электропитанием

В процессоре RISC предусмотрены следующие режимы ожидания (пониженного энергопотребления):

- Sleep – останов синхросигнала для процессора
- Deep sleep – останов синхросигнала для процессора, PLL и Flash

Выбор процессором конкретного режима ожидания определяется значением бита SLEEPDEEP регистра SCR (см. пункт 27.1.4 «Регистр управления системой»)

Далее в разделе описаны механизмы перехода в режим пониженного энергопотребления и условия выхода из этого режима.

### 25.1 Переход в режим пониженного энергопотребления

Система может формировать ложные сигналы событий, выводящие процессор из ожидания, например, они возникают при работе отладчика. Следовательно, программное обеспечение должно быть способно перевести процессор обратно в указанный ожидания. Для этого можно, например, организовать в программе пустой цикл.

### 25.2 Ожидание прерывания

Инструкция ожидания прерывания WFI (wait for interrupt) после своего выполнения немедленно переводит процессор в режим пониженного энергопотребления.

Ожидание события

Инструкция ожидания сигнала события WFE (wait for event) переводит или не переводит процессор в режим пониженного энергопотребления в зависимости от результата проверки одноразрядного регистра события. При этом процессор проверяет значение регистра события, и в случае, если он равен 0, приостанавливает дальнейшее выполнение команд и переходит в состояние ожидания. В случае, если он равен 1, процессор записывает в регистр события 0 и продолжает нормальную работы без перехода в режим ожидания.

### 25.3 Переход в режим ожидания по выходу из обработчика исключения (режим sleep-on-exit)

В случае, если бит SLEEPONEXIT регистра SCR установлен в 1, по завершении выполнения обработчика исключения процессор возвращается в режим приложения, после чего немедленно переходит в состояние пониженного энергопотребления.

Данный механизм рекомендуется использовать в задачах, в которых процессора используется только для обработки исключений.

### 25.4 Выход из состояния ожидания

Условия выхода процессора из режима ожидания зависят от причины, по которой он был переведен в этот режим.

Выход из ожидания по команде WFI и в режиме sleep-on-exit

Как правило, процессор выходит из режима ожидания только в случае возникновения исключительной ситуации с приоритетом, достаточным для активизации соответствующего обработчика.

В некоторых приложениях может возникнуть необходимость выполнения процедур восстановления системы после выхода процессора из режима пониженного энергопотребления, однако до того, как он начнет выполнять обслуживание прерываний. Для того, чтобы добиться этого, достаточно установить бит PRIMASK в 1. В случае возникновения в системе разрешенного прерывания с приоритетом, выше текущего приоритета, процессор будет выведен из ожидания, однако не сможет передать управление обработчику прерывания до тех пор, пока бит PRIMASK не будет установлен в 0.

Более подробная информация о бите PRIMASK представлена в пункте 11.4.9 «PRIMASK».

## 25.5 Выход из ожидания по команде WFE

Процессор выходит из режима ожидания в случае обнаружения исключительной ситуации с приоритетом, достаточным для активизации обработчика.

Кроме того, в случае установки бита SEVONPEND регистра SCR в 1, любое новое не обслуженное прерывание формирует сигнал события, и выводит процессор из ожидания, даже если оно запрещено или имеет приоритет, недостаточно высокий для запуска обработчика.

Более подробная информация о регистре SCR представлена в пункте 27.1.4 «Регистр управления системой».

## 25.6 Рекомендации по программированию режима энергопотребления

В стандарте ANSI языка C отсутствует возможность непосредственной генерации инструкций WFI и WFE. В CMSIS предусмотрены встроенные функции, предназначенные для включения в код этих инструкций:

```
void __WFE(void) // Wait for Event
void __WFI(void) // Wait for Interrupt
```

Периферийные блоки формируют прерывания с IRQ0 до IRQ31

## 26 Контроллер прерываний NVIC

В разделе описан векторный контроллер прерываний с возможностью вложения (NVIC – Nested Vectored Interrupt Controller) и используемые им регистры.

Контроллер обеспечивает поддержку:

- программное задание уровня приоритета в диапазоне от 0 до 192 с шагом 64 независимо каждому прерыванию. Более высокое значение соответствует меньшему приоритету, таким образом, уровень 0 отвечает наивысшему приоритету прерывания;
- срабатывание сигнала прерывания по импульсу и по уровню;
- передача управления из одного обработчика исключения на другой без восстановления контекста.

Процессор автоматически сохраняет в стеке свое состояние (контекст) по входу в обработчик прерывания и восстанавливает его по завершению обработчика, без необходимости непосредственного программирования этих операций. Это обеспечивает обработку исключительных ситуаций с малой задержкой.

Назначение регистров контроллера прерываний представлено в таблице 266.

Таблица 266 – Обобщенная информация о регистрах контроллера NVIC

Адрес	Имя	Тип	Значение после сброса	Описание
0xE000E100	ISER	RW	0x00000000	Регистр разрешения прерываний
0xE000E180	ICER	RW	0x00000000	Регистр запрета прерывания
0xE000E200	ISPR	RW	0x00000000	Регистр перевода прерывания в состояние ожидания обслуживания
0xE000E280	ICPR	RW	0x00000000	Регистр сброса состояния ожидания обслуживания
0xE000E400 - 0xE000E41C	IPR0-7	RW	0x00000000	Регистр приоритета прерываний

### 26.1 Логика работы прерываний контроллера NVIC.

В данном разделе описывается функционирование контроллера NVIC при поступлении на его вход запросов прерываний IRQ от различных модулей периферии микросхемы.

Первоначальным условием работы прерывания является его разрешение в модуле NVIC. За это отвечают регистры:

- ISER – за разрешение прерываний;
- ICER – за запрет прерываний.

В случае, когда соответствующий запрос разрешен (при данном условии рассмотрены все диаграммы в разделе), и приходит сигнал активации прерывания – запрос IRQ request, то возникает признак отложенного прерывания IRQ pending. Данный признак переводит прерывание в состояние ожидания его обработки ядром.



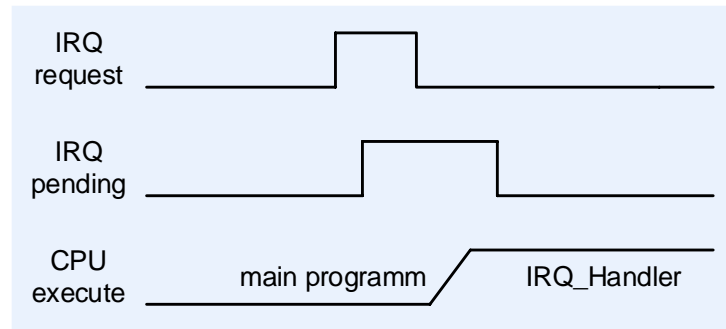


Рисунок 96 – Выставление отложенного запроса на прерывание и последующая его обработка

Pending биты выставляются в регистрах ISPR/ICPR, которые в свою очередь позволяют программно управлять признаком отложенного прерывания. ISPR – для установки pending бит, ICPR – для сброса соответственно. Если после прихода запроса на прерывание IRQ request, сбросить pending бит в регистре ICPR до того, как ядро приступит к его обработке, то прерывание будет проигнорировано - рисунок 97.

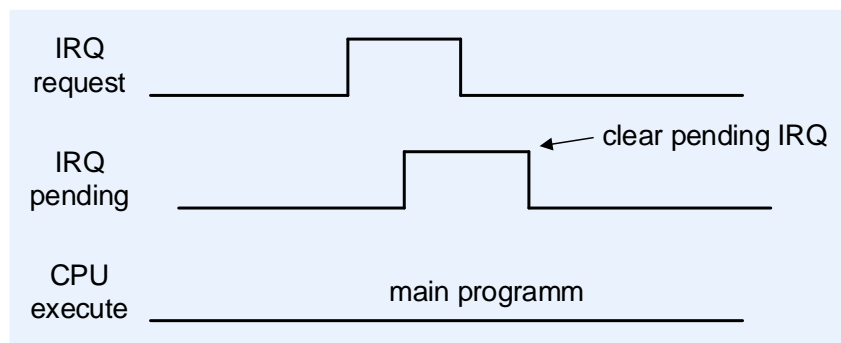


Рисунок 97 – Сброс признака отложенного прерывания, до обработки ядром

Если произойдет снятие запроса IRQ request от источника, «защелкивание» признака отложенного прерывания гарантирует отработку его ядром в соответствии с приоритетом. – рисунок 98. Сам IRQ pending признак снимается автоматически, когда прерывание становится активным, о чем сигнализирует признак IRQ active.

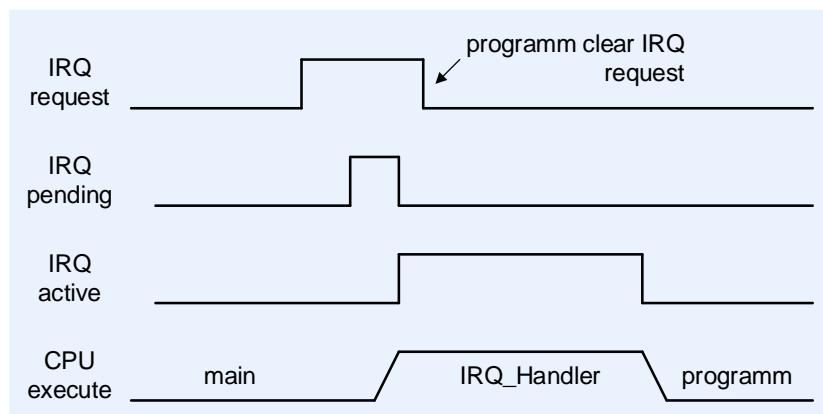


Рисунок 98 – Сброс признака отложенного прерывания, до обработки ядром

После того как прерывание стало активным, повторно запустить обработчик того же прерывания будет невозможно до тех пор, пока не будет завершена процедура обработки прерывания командой выхода из исключения. После выполнения команды выхода происходит сброс признака активности IRQ active.

При удержании источником на входе NVIC запроса на обработку IRQ request, по окончании обработки прерывания и снятия признака активного прерывания IRQ active, происходит повторное выставление признака отложенного прерывания IRQ pending – «защелкивание» pending бита, сброс которого в дальнейшем инициирует повторную активность и обработку того же исключения – рисунок 99.

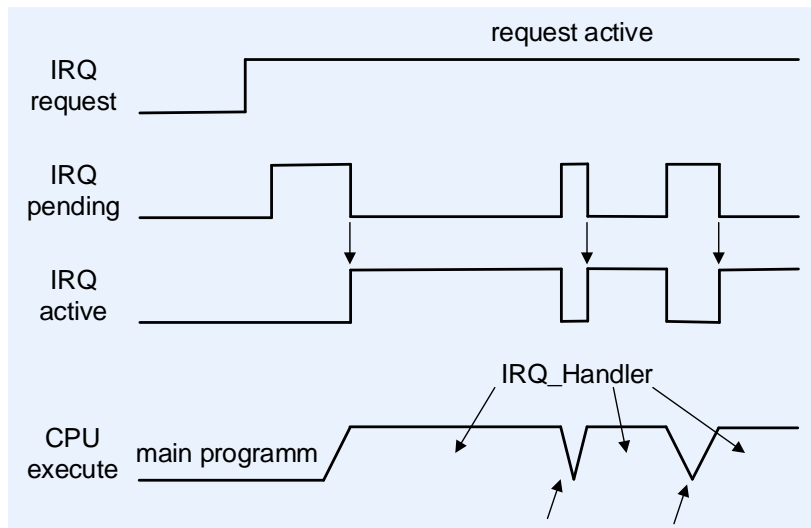


Рисунок 99 – Повторная обработка прерываний при удержании запроса от источника

Необходимо учитывать, что если источник прерываний выдает многократную установку и снятие запроса IRQ request на входе контроллера NVIC, то в таком случае только первый запрос выставляет признак отложенного прерывания IRQ pending, а остальные запросы до начала процедуры обработки прерывания (в момент активного признака отложенного прерывания) будут проигнорированы ядром – рисунок 100.

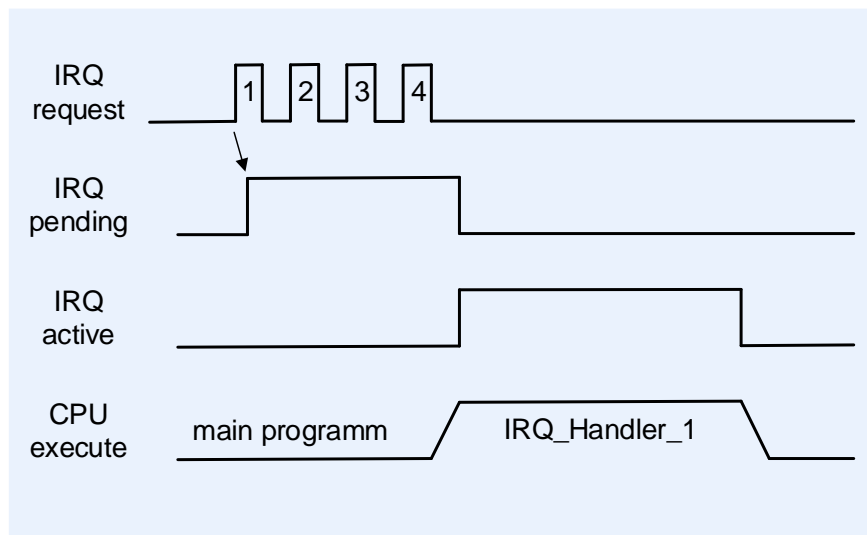


Рисунок 100 – Многократная установка снятие запроса IRQ request

Если запрос на прерывание пришел в момент активного прерывания, то в такой ситуации уже будут отработаны оба запроса на прерывание. В отличие от случая, изображенного на рисунке 100, запрос приходит тогда, когда признак отложенного прерывания IRQ pending уже сброшен, и новый запрос как раз его выставляет, что в дальнейшем позволяет провести повторную обработку прерывания – рисунок 101.

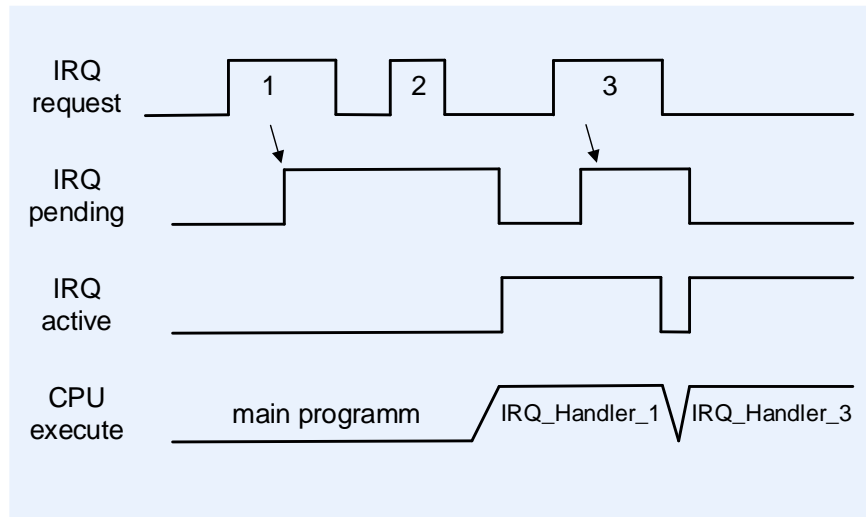


Рисунок 101 – Повторная установка запроса на прерывание в момент выполнения обработчика исключения

Выставление признака отложенного прерывания возможно даже в тех случаях, когда соответствующее прерывание запрещено. Все отложенные прерывания будут отражены в ISPR/ICPR, и в случае разрешения таких прерываний регистром ISPR, ядро тут же приступит к их обработке. Рекомендуется перед разрешением соответствующего прерывания убедиться в отсутствии признака отложенного запроса и, при необходимости, сбросить его.

## 26.2 Регистр разрешения прерываний

Регистр ISER предназначен для разрешения прерываний (запись) и определения, какие из прерываний разрешены (чтение). Более подробная информация представлена в таблице 266.

Распределение битов регистра ISER представлено на рисунке 102:

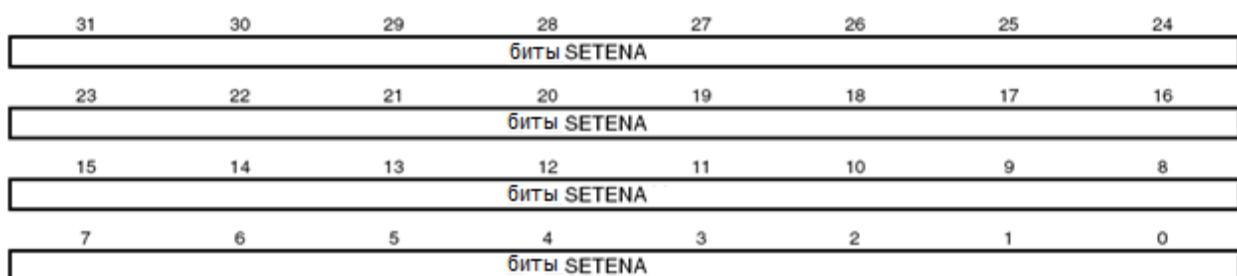


Рисунок 102 – Распределение битов регистра

Назначение бит SETENA:

- запись: 0 – не влияет; 1 – разрешение прерывания;
- чтение: 0 – прерывание запрещено; 1 – прерывание разрешено.

При разрешении прерывания, находящегося в состоянии ожидания обработки, контроллер NVIC активизирует его в зависимости от приоритета. Запрос запрещенного прерывания, переводит его в состояние ожидания обработки, однако контроллер NVIC не активизирует его вне зависимости от приоритета.

### 26.3 Регистр запрета прерываний

Регистр ICER предназначен для запрета прерываний (запись) и определения, какие из прерываний разрешены (чтение). Более подробная информация представлена в таблице 266.

Распределение битов регистра ICER представлено на рисунке 103:

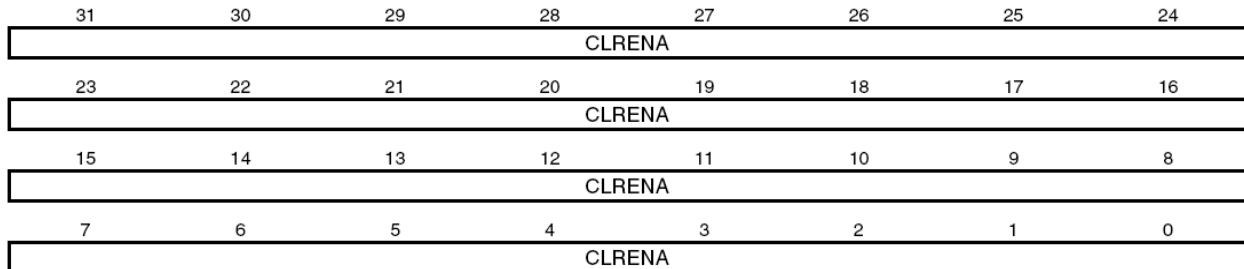


Рисунок 103 – Распределение битов регистра ICER

Назначение бит CLRENA:

- запись: 0 – не влияет; 1 – запрет прерывания.
- чтение: 0 – прерывание запрещено; 1 – прерывание разрешено.

### 26.4 Регистр установки состояния ожидания для прерывания

Регистр ISPR предназначен для принудительного перевода прерываний в состояние ожидания обслуживания (запись) и определения, какие из прерываний находятся в этом состоянии (чтение).

Более подробная информация представлена в таблице 266.

Распределение битов регистра ISPR представлено на рисунке 104:

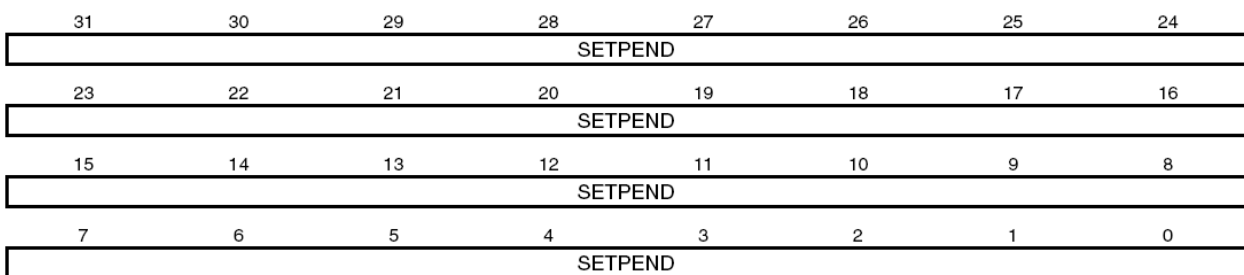


Рисунок 104 – Распределение битов регистра ISPR

Назначение бит SETPEND:

- запись: 0 – не влияет; 1 – перевод прерывания в состояние ожидания.
- чтение: 0 – прерывание не ожидает обслуживания; 1 – прерывание ожидает обслуживания.

Запись 1 в бит регистра ISPR, соответствующий:

- прерыванию, уже ожидающему обслуживания – не влияет на работу системы;
- запрещенному прерыванию – переводит его в состояние ожидания.

## 26.5 Регистр сброса состояния ожидания для прерывания

Регистр ICPR предназначен для принудительного сброса состояния ожидания обслуживания прерывания (запись) и определения, какие из прерываний находятся в состоянии ожидания (чтение).

Более подробная информация представлена в таблице 266.

Распределение битов регистра ICPR представлено на рисунке 105:

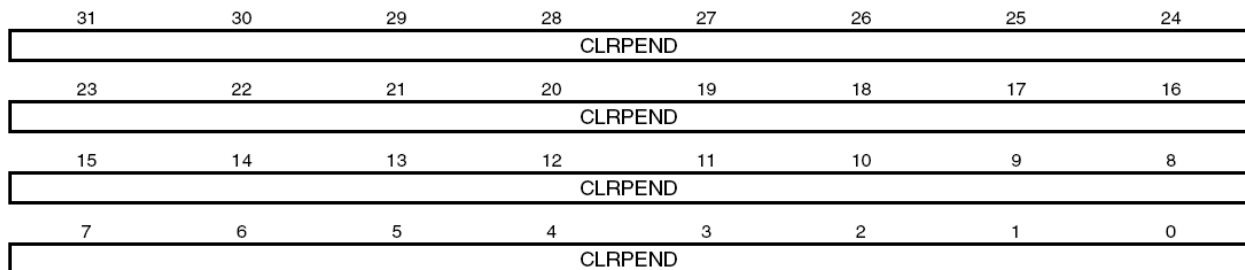


Рисунок 105 – Распределение битов регистра ICPR

Назначение бит CLRPEND:

- запись: 0 – не влияет; 1 – сброс состояния ожидания.
- чтение: 0 – прерывание не ожидает обслуживания; 1 – прерывание ожидает обслуживания.

Запись 1 в разряд регистра ICPR, соответствующего прерыванию в активном состоянии, не влияет на работу системы.

## 26.6 Регистры приоритета прерываний

Регистры IPR0-IPR7 представляют собой набор 8-битовых полей, каждое из которых соответствует одному прерыванию. Регистры доступны пословно. Обобщенная информация об их характеристиках представлена в таблице 266.

Каждый из регистров содержит четыре поля приоритета, которые отображаются на четыре элемента массива PRI[0] ... PRI[31] CMSIS, как показано на рисунке 106.

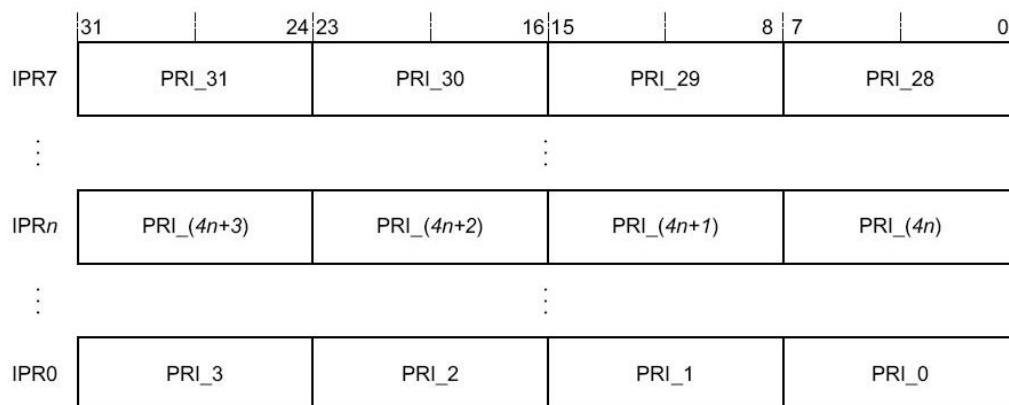


Рисунок 106 – Поля приоритетов регистров IPR0-IPR7

Каждое поле содержит значение приоритета в диапазоне от 0 до 192, причем меньшие значения соответствуют более высокому приоритету соответствующего прерывания. Процессор обеспечивает доступ только к битам [7:6] приоритета, биты [5:0] при чтении всегда равны нулю, а при записи игнорируются. Поэтому, например, запись 255 в регистр запишется как 192.

Для того, чтобы определить номер регистра IPR и смещение данных в регистре необходимо выполнить следующие операции:

- для заданного номера прерывания N номер M соответствующего регистра приоритета равен  $M = N \text{ DIV } 4$ ;
- смещение данных в регистре в зависимости от значения  $N \text{ MOD } 4$  равно:
  - 0 – биты регистра [7:0];
  - 1 – биты регистра [15:8];
  - 2 – биты регистра [23:16];
  - 3 – биты регистра [31:24].

## 26.7 Прерывания, срабатывающие по уровню сигнала

Процессор способен обрабатывать прерывания, сформированные по уровню сигнала. Формирование запроса на прерывание по уровню происходит при условии удержания сигнала не менее двух тактов процессорного ядра.

Прерывание такого типа считается активным до тех пор, пока периферийное устройство не снимет активный уровень сигнала запроса. Как правило, это происходит после соответствующего обращения процедуры обработки прерывания к периферийному устройству.

После того, как процессор передал управление на обработчик, он автоматически снимает признак ожидания обслуживания прерывания (см. подраздел 26.8 «Аппаратное и программное управление прерываниями»). Если прерывание формируется по уровню сигнала, а сигнал запроса не снят до возврата из обработчика, процессор вновь переведет прерывание в состояние ожидания обслуживания, что, в свою очередь, приведет к повторному вызову его обработчика. Таким образом, периферийное устройство может поддерживать сигнал запроса прерывания в активном состоянии до тех пор, пока не перестанет нуждаться в обслуживании.

## 26.8 Аппаратное и программное управление прерываниями

Процессор RISC регистрирует все поступающие прерывания. Перевод прерывания, сформированного периферийным устройством, в состояние ожидания обслуживания осуществляется в одном из следующих случаев:

- контроллер прерываний NVIC обнаруживает, что сигнал запроса имеет высокий логический уровень, а прерывание не активно;
- контроллер прерываний NVIC обнаруживает передний фронт сигнала запроса прерывания;
- программное обеспечение осуществляет запись в соответствующий разряд регистра ISPR0 или соответствующего значения в регистр STIR.

Прерывание находится в состоянии ожидания до тех пор, пока не произойдет одно из следующих событий:

- процессор передаст управление процедуре обработки прерывания. В этом случае прерывание переходит в активное состояние, после чего:
- по завершении обработки прерывания, срабатывающего по уровню, контроллер NVIC проверяет состояние сигнала запроса на прерывание. Если этот сигнал

активен, прерывание вновь переводится в состояние ожидания обслуживания, что приводит к немедленной повторной передаче управления на обработчик. В противном случае прерывание переводится в неактивное состояние.

– если в период выполнения процедуры обработки прерывания, настроенного на срабатывание по фронту, не было зафиксировано импульсов на линии запроса, прерывание переводится в неактивное состояние.

– программное обеспечение осуществляет запись в соответствующий разряд регистра сброса состояния ожидания прерывания.

## 26.9 Рекомендации по работе с контроллером прерываний

Доступ к регистрам контроллера из программного обеспечения должен осуществляться по корректно выровненным адресам. Процессор не поддерживает возможность доступа к контроллеру по невыровненным адресам. Требования по выравниванию приведены в описании регистров.

Прерывание может быть переведено в состояние ожидания обслуживания даже в случае, если оно запрещено.

Программное разрешение или запрещение прерываний может осуществляться с помощью инструкций CPSIE I и CPSID I. В CMSIS предусмотрены следующие встроенные функции, генерирующие эти инструкции:

```
void __disable_irq(void) // Disable Interrupts
void __enable_irq(void) // Enable Interrupts
```

Кроме того, в CMSIS имеется ряд дополнительных функций, обеспечивающих управление контроллером прерываний NVIC:

Таблица 267 – Функции CMSIS для управления контроллером прерываний

Функция	Описание
void NVIC_EnableIRQ(IRQn_t IRQn)	Разрешить IRQn
void NVIC_DisableIRQ(IRQn_t IRQn)	Запретить IRQn
uint32_t NVIC_GetPendingIRQ (IRQn_t IRQn)	Вернуть истину, если прерывание IRQn ожидает обслуживания, ложь – в противном случае
void NVIC_SetPendingIRQ (IRQn_t IRQn)	Перевести IRQn в состояние ожидания обслуживания
void NVIC_ClearPendingIRQ (IRQn_t IRQn)	Сбросить состояние ожидания обслуживания для IRQn
void NVIC_SetPriority (IRQn_t IRQn, uint32_t priority)	Установить приоритет для IRQn
uint32_t NVIC_GetPriority (IRQn_t IRQn)	Считать приоритет IRQn
void NVIC_SystemReset (void)	Сбросить систему

Более подробная информация отражена в документации по CMSIS.

## 27 Блок управления системой ядра

Блок управления системой (SCB – System control block) обеспечивает доступ к информации о конфигурации и управление работой системы. Регистры блока управления системой представлены в таблице 268.

Таблица 268 – Обобщенная информация о регистрах блока управления системой

Адрес	Имя	Тип	Значение после сброса	Описание
0xE000ED00	CPUID	RO	0x410CC200	Регистр идентификации процессора
0xE000ED04	ICSR	RW	0x00000000	Регистр управления прерываниями
0xE000ED0C	AIRCR	RW	0xFA050000	Регистр управления прерываниями и программного сброса
0xE000ED10	SCR	RW	0x00000000	Регистр управления системой
0xE000ED14	CCR	RW	0x00000200	Регистр конфигурации и управления
0xE000ED1C	SHPR2	RW	0x00000000	Регистр №2 приоритета системных обработчиков
0xE000ED20	SHPR3	RW	0x00000000	Регистр №3 приоритета системных обработчиков

### 27.1 Упрощенный доступ к регистрам блока управления системой из среды разработки программного обеспечения

В целях повышения эффективности разработки программного обеспечения в CMSIS предусмотрен упрощенный доступ к регистрам SCB, а именно, регистры SHPR2-SHPR3 в CMSIS отображаются на массив SHP[1].

#### 27.1.1 Регистр идентификации процессора

Регистр CPUID содержит информацию о модели процессора, версии и варианте его реализации. Подробная информация о регистре представлена в таблице 268.

Назначение разрядов регистра CPUID представлено на рисунке 107:

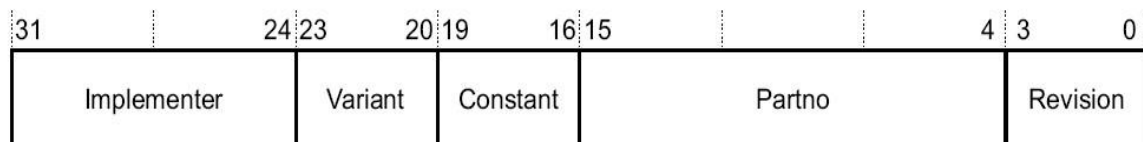


Рисунок 107 – Назначение разрядов регистра CPUID

Implementer – код разработчика 0x41;

Variant – значение r в номере версии rnpn изделия: 0x0 = r0p0;

Constant – постоянное значение 0xC;

PartNo – номер модели процессора: 0xC20 = RISC;

Revision – значение p в номере версии rnpn изделия: 0x0 = r0p0.



### 27.1.2 Регистр управления прерываниями

Регистр ICSR обеспечивает возможность установки и сброса состояния ожидания обслуживания для исключений PendSV и SysTick, а также доступ к следующей информации:

- номер текущего обрабатываемого исключения;
- наличие активных исключений, обработка которых была прервана;
- номер исключения с наивысшим приоритетом, ожидающего обслуживания;
- наличие прерываний, ожидающих обслуживания.

Подробная информация о регистре представлена в таблице 268.

Назначение разрядов регистра ICSR представлено на рисунке 108:

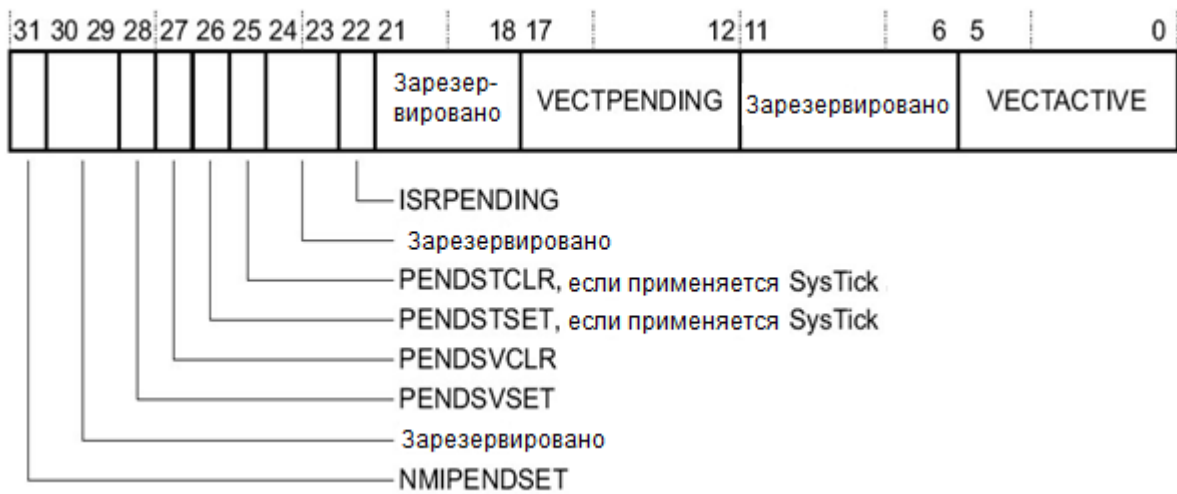


Рисунок 108 – Назначение разрядов регистра ICSR

NMIPENDSET (RW) – так как NMI имеет самый высокий приоритет, процессор переходит на обработчик прерывания, как только произошла запись “1” в этот бит. После перехода на обработчик прерывания, процессор очищает этот бит. Поэтому если обработчик прочитал “1”, это значит, что сигнал NMI опять перешел в активный уровень во время обработки прерывания.

PENDSVSET (RW) – бит установки состояния ожидания обслуживания для исключения PendSV. Запись: 0 – не влияет на работу системы, 1 – переводит исключение PendSV в состояние ожидания обслуживания. Чтение: 0 – исключение PendSV не ожидает обслуживания, 1 – ожидает.

Запись 1 в разряд PENDSVSET это единственно возможный способ перевода исключения PendSV в состояние ожидания обслуживания.

PENDSVCLR (WO) – бит сброса состояния ожидания обслуживания для исключения PendSV. Запись: 0 – не влияет на работу системы, 1 – сбрасывает состояние ожидания обслуживания для исключения PendSV.

PENDSTSET (RW) – бит установки состояния ожидания обслуживания для исключения SysTick. Запись: 0 – не влияет на работу системы, 1 – переводит исключение SysTick в состояние ожидания обслуживания. Чтение: 0 – исключение SysTick не ожидает обслуживания, 1 – ожидает.

PENDSTCLR (WO) – бит сброса состояния ожидания обслуживания для исключения SysTick. Запись: 0 – не влияет на работу системы, 1 – сбрасывает состояние ожидания обслуживания для исключения SysTick.

Данный бит доступен только для записи, при чтении результат не определен.

ISR\_PENDING (RO) – флаг наличия в системе прерываний (за исключением отказов), ожидающих обслуживания. 0 – ожидающие обслуживания прерывания отсутствуют, 1 – присутствуют.

VECT\_PENDING (RO) – содержит номер ожидающего обслуживания исключения с наивысшим приоритетом, обработка которого в системе разрешена. 0 – не обслуженных исключений нет, другое число – номер ожидающего обслуживания исключения.

Значение данного поля формируется с учетом полей BASEPRI и FAULTMASK, однако не учитывает влияние поля PRIMASK.

VECT\_ACTIVE (RO) – содержит номер активного исключения. 0 – режим приложения, другое число – номер текущего обслуживаемого исключения. Для получения номера запроса прерывания (IRQ) из значения VECT\_ACTIVE необходимо вычесть 16.

Запись в регистр ICSR может привести к непредсказуемым результатам в случае:

- одновременной установки в 1 битов PENDSVSET и PENDSVCLR;
- одновременной установки в 1 битов PENDSTSET и PENDSTCLR;

### 27.1.3 Регистр управления прерываниями и программного сброса

Регистр AIRCR позволяет задавать группировку приоритетов исключений, порядок следования байт в слове (endian) при доступе к данным, а также управлять процессом сброса системы.

Подробная информация о регистре представлена в таблице 268.

Для записи данных в регистр необходимо установить его поле VECTKEY в значение 0x05FA, в противном случае попытка записи будет проигнорирована процессором.

Назначение разрядов регистра AIRCR представлено на рисунке 109:



Рисунок 109 – Назначение разрядов регистра AIRCR

VECTKEY – ключ доступа к регистру. При записи должен быть равен 0x05FA, в противном случае попытка записи в регистр будет проигнорирована процессором.

ENDIANESS (RO) – порядок следования значащих разрядов при доступе к данным. 0 – младший байт идет первым (little-endian), 1 – старший байт идет первым (big-endian).

SYSRESETREQ (WO) – запрос сброса системы. 0 – не влияет на работу, 1 – инициирует сигнал сброса процессора. При чтении возвращает 0.

VECTCLRACTIVE (WO) – зарезервировано для целей отладки. При чтении возвращает 0. При записи данных в регистр значение поля должно быть равно 0, в противном случае результат непредсказуем.

### 27.1.4 Регистр управления системой

Регистр SCR позволяет определить требования к переходу в и выходу из режима пониженного энергопотребления.

Подробная информация о регистре представлена в таблице 268. Назначение разрядов регистра SCR представлено на рисунке 110:

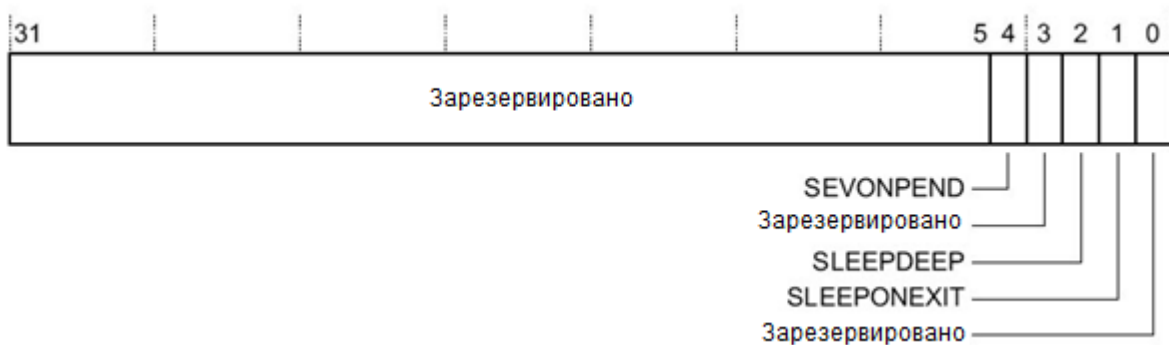


Рисунок 110 – Назначение разрядов регистра SCR

SEVONPEND – разрешает или запрещает формирование сигнала события при переводе исключения в состояние ожидания обработки. 0 – выход из режима пониженного энергопотребления по прерыванию могут инициировать только разрешенные прерывания или события; 1 – выход может инициироваться разрешенными событиями и любыми, в том числе запрещенными, прерываниями.

Перевод прерывания в состояние ожидания обслуживания формирует событие, что в свою очередь приводит к выходу процессора из режима пониженного потребления, инициированного инструкцией WFE, либо к регистрации факта события, если эта инструкция еще не выполнялась.

Кроме того, процессор может быть выведен из режима пониженного энергопотребления при поступлении внешнего события, а также после выполнения инструкции SEV.

SLEEPDEEP – определяет режим пониженного энергопотребления процессора: 0 – спящий режим (sleep), 1 – режим глубокого сна (deep sleep).

SLEEPONEXIT – разрешает или запрещает перевод процессора в режим пониженного энергопотребления при выходе из обработчика события в режим выполнения прикладной программы: 0 – не переводить, 1 – переводить.

### 27.1.5 Регистр конфигурации и управления

Регистр CCR управляет процессом перехода процессора в режим приложения, а также позволяет запретить или разрешить:

- игнорирование отказов доступа к шине в обработчиках тяжелых отказов и при эскалации отказа по FAULTMASK;
- генерацию исключений при делении на ноль и при доступе по невыровненному адресу;
- доступ к регистру STIR из непривилегированного приложения.

Подробная информация о регистре представлена в таблице 268. Назначение разрядов регистра CCR представлено на рисунке 111:



Рисунок 111 – Назначение разрядов регистра CCR

STKALIGN определяет режим выравнивания адреса стека при обработке исключений: 0 – выравнивание по границе 4 байт; 1 – по границе 8 байт. При передаче управления на обработчик исключения процессор анализирует бит [9] сохраненного в стеке слова состояния PSR и определяет по нему режим выравнивания стека. При возврате из обработчика процессор использует сохраненный в стеке бит этого слова для восстановления требуемого режима выравнивания.

UNALIGN\_TRP всегда читается как «1», информируя, что при любом невыровненном доступе к данным, происходит переход на Hard Fault.

### 27.1.6 Регистры приоритета системных обработчиков

Регистры приоритета системных обработчиков SHPR2-SHPR3 позволяют установить уровень приоритета обработки исключений.

Доступ к регистрам осуществляется пословно. Подробная информация о регистре представлена в таблице 268.

Поля PRI\_N регистров имеют ширину 8 бит, однако в процессоре реализована поддержка доступа только к старшему двум битам [7:6], при чтении данных из младшего бит [5:0] процессор возвращает нули.

Таблица 269 – Поля приоритета обработчиков системных отказов

Обработчик отказа	Поле	Описание регистра
Вызов SVCcall	PRI_11	Регистр №2 приоритета системных обработчиков
Вызов PendSV	PRI_14	Регистр №3 приоритета системных обработчиков
Вызов SysTick	PRI_15	

**27.1.7 Регистр №2 приоритета системных обработчиков**

Назначение разрядов регистра SHPR2 представлено на рисунке 112:

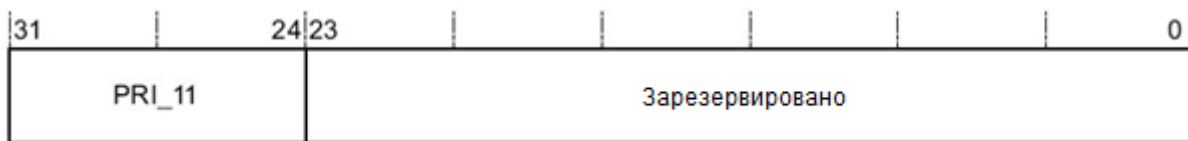


Рисунок 112 – Назначение разрядов регистра SHPR2  
 PRI\_11 Приоритет системного обработчика 11, вызов SVCall

**27.1.8 Регистр №3 приоритета системных обработчиков**

Назначение разрядов регистра SHPR3 представлено на рисунке 113:

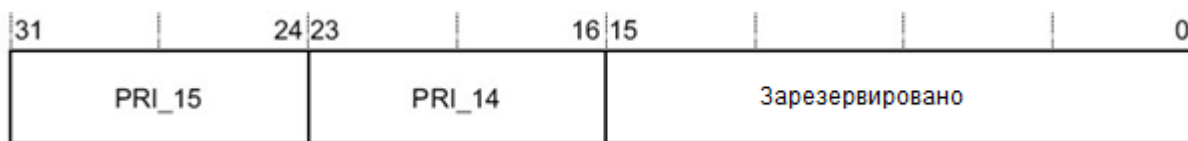


Рисунок 113 – Назначение разрядов регистра SHPR3  
 PRI\_15 Приоритет системного обработчика 15, вызов SysTick  
 PRI\_14 Приоритет системного обработчика 14, вызов PendSV

**27.1.9 Рекомендации по программированию блока управления системой**

Необходимо убедиться, что программа использует для обращения к регистрам блока управления системой доступ по корректно выровненным адресам. Обращение ко всем регистрам должно быть выровнено по границе слова.

## 28 Сторожевые таймеры

### 28.1 Блок сторожевого таймера IWDG

IWDG – независимый 12-разрядный сторожевой таймер, который считает вниз от основания счета (значения перезагрузки, настраиваемого начального значения) до нуля. При достижении счетчиком нуля сторожевой таймер формирует сигнал сброса, который перезагружает систему. Для предотвращения формирования сброса от IWDG необходимо заблаговременно перезагрузить таймер записью ключевого значения 0хАААА в регистр IWDG\_KR – после чего таймер снова начнет обратный отсчет от установленного значения. Тем самым и обеспечивается контроль над зависанием программы в определенных точках, где такое возможно (не перезагрузив IWDG вовремя, система сбросится). Сброс системы от независимого сторожевого таймера сбрасывает IWDG. Любой другой системный сброс (например, внешний сброс) не влияет на настройки и работу таймера IWDG.

Блок-схема независимого сторожевого таймера приведена на рисунке 114.

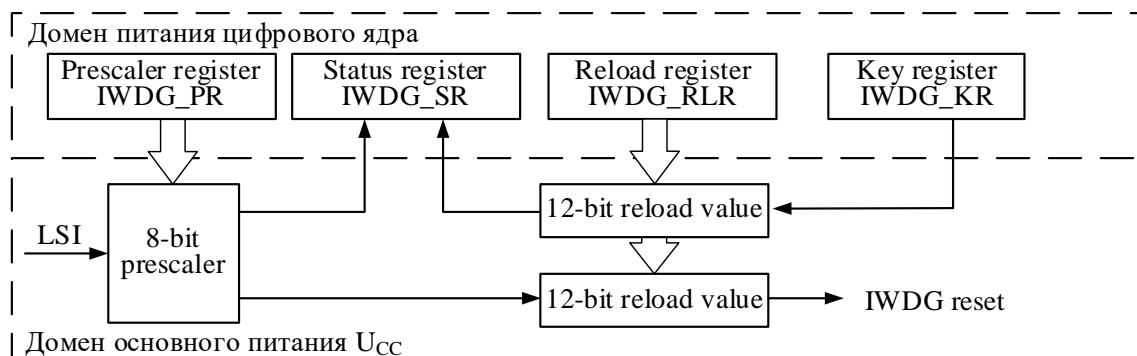


Рисунок 114 – Блок-схема независимого сторожевого таймера

Структурно IWDG состоит из двух частей. Часть с регистрами управления расположена в домене питания цифрового ядра, доступна по APB, тактируется частотой PCLK и сбрасывается по всем типам сброса. Часть непосредственно с независимым сторожевым таймером расположена в домене основного питания микросхемы UCC и тактируется от генератора LSI – поэтому для фактической записи в регистры IWDG в домене основного питания UCC генератор LSI должен находиться в рабочем режиме. Часть сторожевого таймера в домене основного питания UCC сбрасывается только от сброса, формируемого IWDG – после сброса от IWDG сторожевой таймер останавливает свою работу.

Период счета сторожевого таймера от разрешения работы до формирования сброса является функцией от основания счета (значения перезагрузки) и поделенной на значение делителя частоты LSI.

Обновление значений делителя IWDG в домене основного питания осуществляется записью регистра IWDG\_PR соответственно. Обновление основания счета (значение, от которого таймер при старте начинает обратный отсчет) IWDG в домене основного питания осуществляется значением регистра IWDG\_RLR записью ключевого значения 0хАААА в регистр ключа. Для возможности записи в регистры IWDG\_PR и IWDG\_RLR предварительно в регистр ключа должно быть

записано ключевое значение 0x5555 для разрешения доступа по записи в них. При записи в регистр IWDG\_PR значения предделителя и ключа 0xAAAA в регистр IWDG\_KR формируются запросы на обновление соответствующих параметров (предделителя и основания счета), которые передаются IWDG в домене основного питания. Запросы на обновления не захватываются (не сохраняются) частью в домене основного питания, то есть при любом типе сброса запросы на обновление от IWDG в домене питания ядра сбрасываются. Наличие запроса обновления от IWDG в домене питания ядра и/или информация о процессе фактического обновления на стороне IWDG в домене основного питания транслируются флагами PVU (для предделителя) и RVU (для основания счета). Фактическое обновление значений (со сбросом соответствующих флагов после обновления) осуществляется только при разрешенной работе IWDG и наличии частоты LSI. При фактическом обновлении предделителя и основания счета недоступна запись в регистры IWDG\_PR и IWDG\_RLR соответственно. После фактического обновления снимаются запросы обновления со стороны IWDG в домене питания ядра Ucc.

Пример работы с блоком IWDG:

- 1 Включение генератора LSI и ожидание его выхода в рабочий режим;
- 2 Разрешение доступа к регистрам IWDG\_PR и IWDG\_RLR записью значения 0x5555 в регистр IWDG\_KR;
- 3 Запись значения предделителя в регистр IWDG\_PR;
- 4 Разрешение работы IWDG записью значения 0xCCCC в регистр IWDG\_KR;
- 5 Ожидание обновления значения предделителя (сброса флага PVU в 0);
- 6 Разрешение доступа к регистрам IWDG\_PR и IWDG\_RLR записью значения 0x5555 в регистр IWDG\_KR.
- 7 Проверка флага RVU = 0 перед записью в IWDG\_RLR;
- 8 Запись значения перезагрузки в регистр IWDG\_RLR;
- 9 Периодическая перезагрузка сторожевого таймера в процессе выполнения программы значением регистра IWDG\_RLR с помощью записи ключевого значения 0xAAAA в регистр IWDG\_KR.

### 28.1.1 Описание регистров блока сторожевого таймера IWDG

Таблица 270 – Описание регистров блока сторожевого таймера IWDG

Базовый адрес	Название	Описание
0x4005_0000	IWDG	Сторожевой таймер IWDG
Смещение		
0x00	IWDG_KR[15:0]	Регистр ключа
0x04	IWDG_PR[2:0]	Делитель частоты сторожевого таймера
0x08	IWDG_RLR[11:0]	Регистр основания счета сторожевого таймера
0x0C	IWDG_SR[1:0]	Регистр статуса сторожевого таймера

**28.1.1.1 IWDG\_KR**

Таблица 271 – Регистр IWDG\_KR

Номер	31...16	15...0
Доступ	U	W
Сброс	0	0
	-	KEY[15:0]

Таблица 272 – Описание битов регистра IWDG\_KR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...0	KEY[15:0]	<p>Значение ключа (только запись, читается 0x0000).</p> <p>При записи:</p> <ul style="list-style-type: none"> <li>– 0xAAAA – перезагрузка значения таймера значением регистра IWDG_RLR. Значение должно периодически программно записываться при разрешенной работе IWDG, в противном случае сторожевой таймер генерирует сброс, если таймер достиг значения нуля.</li> <li>– 0x5555 – разрешение доступа по записи к регистрам IWDG_PR и IWDG_RLR. Если после разрешения доступа в регистр ключа записывается другое ключевое значение, доступ к регистрам IWDG_PR и IWDG_RLR запрещается.</li> <li>– 0xCCCC – разрешение работы сторожевого таймера (если работа сторожевого таймера не разрешена; работа таймера останавливается аппаратно сбросом от IWDG)</li> </ul>

**28.1.1.2 IWDG\_PR**

Таблица 273 – Регистр IWDG\_PR

Номер	31...3	2...0
Доступ	U	R/W
Сброс	0	0
	-	PR[2:0]

Таблица 274 – Описание битов регистра IWDG\_PR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...3	-	Зарезервировано
2...0	PR[2:0]	<p>Делитель частоты сторожевого таймера:</p> <ul style="list-style-type: none"> <li>000 – делитель на 4: LSI/4;</li> <li>001 – делитель на 8: LSI/8;</li> <li>010 – делитель на 16: LSI/16;</li> <li>011 – делитель на 32: LSI/32;</li> <li>100 – делитель на 64: LSI/64;</li> <li>101 – делитель на 128: LSI/128;</li> <li>110 – делитель на 256: LSI/256;</li> <li>111 – делитель на 256: LSI/256.</li> </ul>



Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		<p>Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR.</p> <p>При записи в регистр формируется запрос на обновление значения делителя частоты в часть IWDG в домене основного питания (см. рисунок 114).</p> <p>Запись в регистр IWDG_PR недоступна при фактической записи предделителя в часть IWDG в домене основного питания U<sub>CC</sub> (см. рисунок 114).</p> <p>При чтении возвращает записанное в регистр значение, а не значение предделителя IWDG в домене основного питания</p>

### 28.1.1.3 IWDG\_RLR

Таблица 275 – Регистр IWDG\_RLR

Номер	31...12	11...0
Доступ	U	R/W
Сброс	0	0xFFF
	-	RLR[11:0]

Таблица 276 – Описание битов регистра IWDG\_RLR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...12	-	Зарезервировано
11...0	RLR[11:0]	<p>Значение перезагрузки (основания счета) сторожевого таймера. Сторожевой таймер декрементируется, начиная с этого значения. Определяет значение, загружаемое в сторожевой таймер при записи значения 0xAAAA в регистр IWDG_KR.</p> <p>Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR.</p> <p>Запись в регистр IWDG_RLR недоступна при фактической записи основания счета в часть IWDG в домене основного питания U<sub>CC</sub> (см. рисунок 114).</p> <p>При чтении возвращает записанное в регистр значение, а не значение перезагрузки IWDG в домене основного питания</p>

### 28.1.1.4 IWDG\_SR

Таблица 277 – Регистр IWDG\_SR

Номер	31...2	1	0
Доступ	U	R	R
Сброс	0	0	0
	-	RVU	PVU

Таблица 278 – Описание битов регистра IWDG\_SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	-	Зарезервировано
1	RVU	<p>Флаг обновления значения сторожевого таймера.</p> <p>Устанавливается аппаратно и служит признаком того, что обновляется значение сторожевого таймера IWDG в домене основного питания U<sub>CC</sub> из регистра перезагрузки IWDG_RLR (см. рисунок 114).</p> <p>Работа IWDG разрешена и LSI работает: Отображает запрос от регистров управления на обновление значения сторожевого таймера значением регистра IWDG_RLR и/или фактическое обновление значения сторожевого таймера в домене основного питания U<sub>CC</sub>. Сбрасывается при завершении фактического обновления значения перезагрузки. Значение регистра перезагрузки IWDG_RLR может быть обновлено только, если этот бит равен нулю.</p> <p>Работа IWDG не разрешена или LSI не работает: Отображает запрос от регистров управления на обновление значения сторожевого таймера в домене основного питания U<sub>CC</sub> значением регистра IWDG_RLR. Сбрасывается любым типом сброса. Значение регистра перезагрузки IWDG_RLR может быть обновлено независимо от состояния этого бита – после запуска IWDG будет обновлен последним записанным в IWDG_RLR значением</p>
0	PVU	<p>Флаг обновления делителя частоты сторожевого таймера.</p> <p>Устанавливается аппаратно и служит признаком того, что обновляется значение делителя частоты IWDG в домене основного питания U<sub>CC</sub> из регистра предделителя IWDG_PR (см. рисунок 114).</p> <p>Работа IWDG разрешена и LSI работает: Отображает запрос от регистров управления на обновление делителя частоты сторожевого таймера значением регистра IWDG_PR и/или фактическое обновление значения делителя в домене основного питания U<sub>CC</sub>. Сбрасывается при завершении фактического обновления делителя. Значение регистра предделителя IWDG_PR может быть обновлено только, если этот бит равен нулю.</p> <p>Работа IWDG не разрешена или LSI не работает: Отображает запрос от регистров управления на обновление значения сторожевого таймера в домене основного питания U<sub>CC</sub> значением регистра IWDG_PR. Сбрасывается любым типом сброса. Значение регистра перезагрузки IWDG_PR может быть обновлено независимо от состояния этого бита – после запуска IWDG будет обновлен последним записанным в IWDG_PR значением</p>

## 28.2 Блок сторожевого таймера WWDG

### 28.2.1 Описание регистров блока сторожевого таймера WWDG

Таблица 279 – Описание регистров блока сторожевого таймера WWDG

Базовый адрес	Название	Описание
0x4004_8000	WWDG	Сторожевой таймер WWDG
Смещение		
0x00	WWDG_CR[7:0]	Регистр управления
0x04	WWDG_CFR[9:0]	Регистр конфигурации
0x08	WWDG_SR[0]	Регистр статуса

#### 28.2.1.1 WWDG\_CR

Таблица 280 – Регистр WWDG\_CR

Номер	31....18	7	6	5	4	3	2	1	0
Доступ		R/S	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс		0	1	1	1	1	1	1	1
	-	WDGA	T6	T5	T4	T3	T2	T1	T0

Таблица 281 – Описание битов регистра WWDG\_CR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..18	-	Зарезервировано
7	WDGA	Бит активации. Этот бит устанавливается программно и очищается только аппаратно при сбросе. Когда WDGA = 1, сторожевой таймер может генерировать сброс. 0 – сторожевой таймер отключен; 1 – сторожевой таймер включен
6..0	T[6:0]	Значение семиразрядного счетчика (от старших разрядов к младшим). Эти биты содержат значение сторожевого таймера, который декрементируется каждые $4096 \cdot 2^{WDGTB}$ циклов частоты PCLK периферийной шины APB

#### 28.2.1.2 WWDG\_CFR

Таблица 282 – Регистр WWDG\_CFR

Номер	31...10	9	8
Доступ	U	R/S	R/W
Сброс		0	0
	-	EWI	WDGTB1

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	1	1	1	1	1	1	1
	WDGTB0	W6	W5	W4	W3	W2	W1	W0

Таблица 283 – Описание битов 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, разрешается генерация прерывания при достижении сторожевым таймером значения 40h. Прерывание запрещается только аппаратным сбросом
8, 7	WGTV[1:0]	Делитель частоты сторожевого таймера: 00 – частота таймера (PCLK / 4096) /1; 01 – частота таймера (PCLK / 4096) /2; 10 – частота таймера (PCLK / 4096) /4; 11 – частота таймера (PCLK / 4096) /8
6...0	W[6:0]	Значение окна. Эти биты содержат значение окна, в пределах которого возможна инициализация битов T[6:0] значением в пределах 40h-7Fh. Если происходит инициализация битов в момент T > W, формируется сброс на выходе RESET. Если таймер достигнет значения T = 3Fh, также формируется сброс

### 28.2.1.3 WWDG\_SR

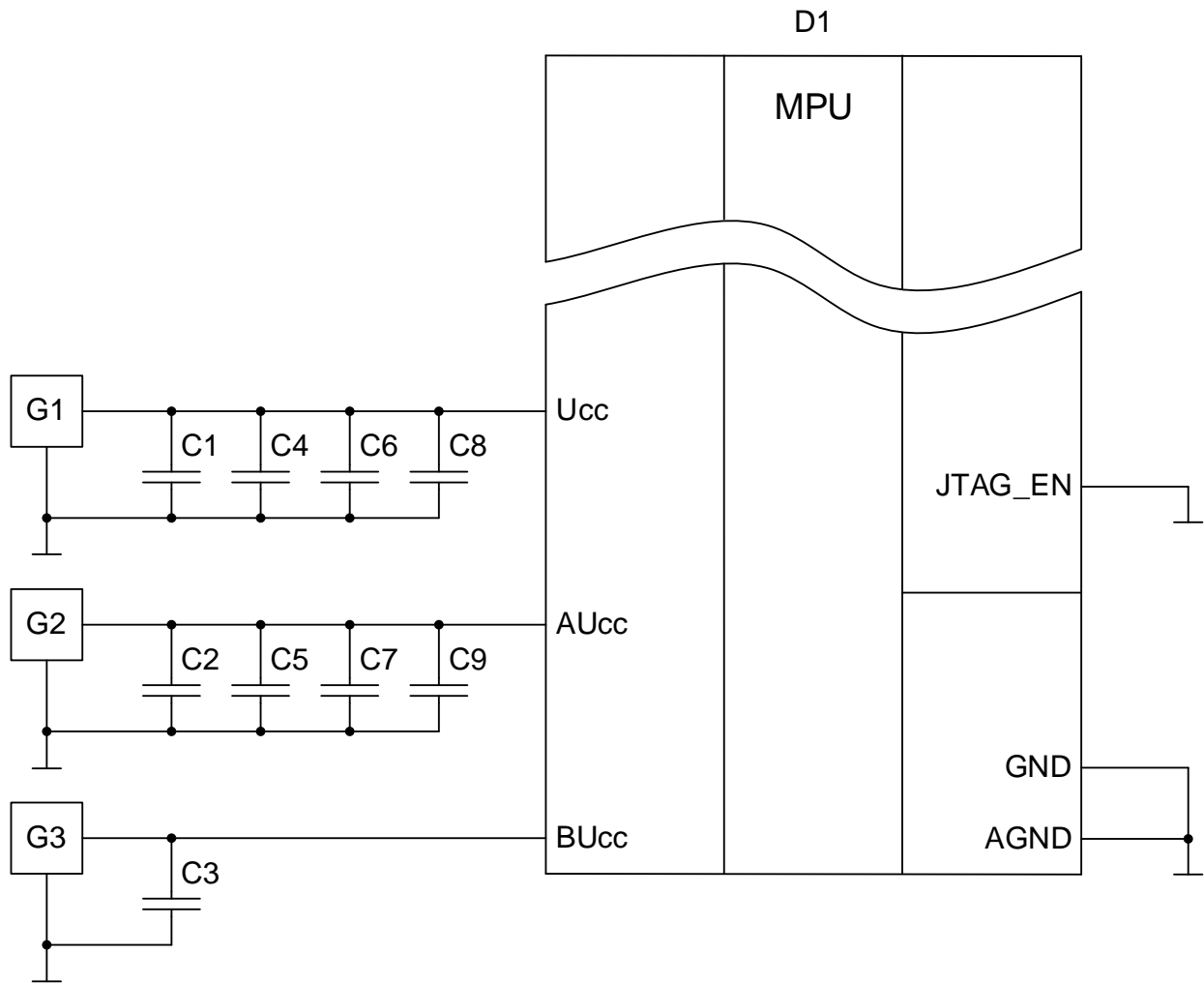
Таблица 284 – Регистр WWDG\_SR

Номер	31...1	0
Доступ	U	R/C
Сброс		0
	-	EWIF

Таблица 285 – Описание битов регистра WWDG\_SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...1	-	Зарезервировано
0	EWIF	Флаг раннего предупреждающего прерывания. Этот бит устанавливается аппаратно, когда сторожевой таймер достигает значения 40h. Бит очищается программно записью нуля. Запись единицы не влияет. Этот бит также устанавливается, если прерывание запрещено EWI = 0

## 29 Типовая схема подключения питания



C1-C9 – конденсаторы емкостью 0,1 мкФ ± 10 %;

D1 – микросхема;

G1 – источник напряжения питания U<sub>CC</sub>;

G2 – источник напряжения питания U<sub>CCA</sub>;

G3 – источник напряжения питания U<sub>CCB</sub>

Рисунок 115 – Типовая схема подключения питания

### 30 Пределно-допустимые характеристики микросхемы

Таблица 286 – Пределно-допустимые режимы эксплуатации и предельные электрические режимы микросхем

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Пределно- допустимый режим		Пределный режим	
		не менее	не более	не менее	не более
Напряжение источника питания, В	U <sub>CC</sub>	2,2	3,6	–	4,0
Напряжение источника питания при использовании АЦП, В	U <sub>CCA</sub>	3,0	3,6		4,0
Напряжение источника питания батареиногo домена, В	U <sub>CCB</sub>	1,8	U <sub>CC</sub> +0,3	–	4,0
Входное напряжение высокого уровня, В, на выводах: PB(4-9), PC (1-6) на выводе OSC_IN при HSE BYPASS=1	U <sub>IH</sub>	2,0	U <sub>CC</sub>	–	U <sub>CC</sub> +0,3
на выводах: PA, PB(0-3,10-14), PC(0, 7-15), RESET, WAKEUP, JTAG_EN			5,25	–	5,3
Входное напряжение низкого уровня, В, при работе в цифровом режиме на выводах: PA, PB, PC, RESET, WAKEUP, JTAG_EN на выводе: OSC_IN при HSE BYPASS=1	U <sub>IL</sub>	0	0,8	–0,3	–
Выходной ток высокого уровня, мА, при работе в цифровом режиме на выводах: PA, PB, PC	I <sub>OH</sub>	–3	–	–10	–
Выходной ток низкого уровня, мА на выводах: PA, PB, PC	I <sub>OL</sub>	–	6	–	10
Частота следования импульсов тактовых сигналов, МГц	f <sub>C</sub>	–	36	–	–
Частота следования импульсов тактовых сигналов HSE, МГц при BYPASS=0 при BYPASS=1	f <sub>C_HSE</sub>	2	16	–	–
		–	36		
Частота следования импульсов тактовых сигналов LSE, кГц при BYPASS=0 при BYPASS=1	f <sub>C_LSE</sub>	32	33	–	–
		–	1 000		
Частота следования импульсов тактовых сигналов PLL, МГц	f <sub>C_PLL</sub>	2	16	–	–
Емкость нагрузки, пФ, на выводах: PA, PB, PC	C <sub>L</sub>	–	30	–	–

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Время хранения информации, лет, при T=25 °C	t <sub>GS</sub>	25	–	–	–
при T=85 °C		10	–	–	–
Параметры АЦП					
Напряжение нижней границы внешней опоры АЦП, В	U <sub>ADC1_REF-</sub>	0	U <sub>CCA-2,4</sub>	–	4,0
Напряжение верхней границы внешней опоры АЦП, В	U <sub>ADC0_REF+</sub>	2,4	U <sub>CCA</sub>	–	4,0
Диапазон напряжения внешнего опорного источника АЦП, В U <sub>REF(ADC)</sub> = U <sub>ADC0_REF+</sub> – U <sub>ADC1_REF-</sub>	U <sub>REF(ADC)</sub>	2,4	U <sub>CCA</sub>	–	–
Диапазон напряжения на входе АЦП, В U <sub>ADC1_REF-</sub> = AGND, U <sub>ADC0_REF+</sub> = AU <sub>CC</sub>	U <sub>AIN</sub>	U <sub>ADC1_REF-</sub>	U <sub>ADC0_REF+</sub>	–0,3	4,0
Частота следования импульсов тактовых сигналов АЦП, МГц	f <sub>C_ADC_S</sub>	–	14	–	–
Параметры ΔΣАЦП					
Частота следования импульсов тактовых сигналов ΔΣАЦП, МГц	f <sub>C_ADC_D</sub>	–	8,196	–	–
Амплитуда входного дифференциального сигнала ΔΣАЦП, В	A <sub>NADC_D</sub>	–	1	–	–
Напряжение на входе ΔΣАЦП, В	U <sub>IADC_D</sub>	–0,5	0,5	–0,8	U <sub>CC</sub> +0,3
Параметры площадок ЖК дисплея					
Напряжение на входе VLCD, В	V <sub>LCD</sub>	0	2,0	–	U <sub>CC</sub> +0,3
Примечание – Не допускается одновременное задание двух предельных режимов					

### 31 Электрические параметры микросхемы

Таблица 287 – 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	U <sub>OH</sub>	1,6	–	25, 85, – 40
Выходное напряжение низкого уровня, В, на выводах PA, PB, PC	U <sub>OL</sub>	–	0,4	
Напряжение срабатывания схемы генерации сброса, В	U <sub>BOR</sub>	1,8	2,1	
Входной ток утечки высокого уровня, мкА, (при работе в цифровом режиме) на выводах: PA, PB, PC, RESET, WAKEUP на выводе: JTAG_EN на выводе OSC_IN	I <sub>ILH</sub>	– 1,0	1,0	
		– 180	180	
		– 40	40	
Ток утечки низкого уровня цифровых входов, мкА, (при работе в цифровом режиме) на выводах: PA, PB, PC, RESET, WAKEUP, JTAG_EN на выводе OSC_IN	I <sub>ILL</sub>	– 1	1	
		– 40	40	
Статический ток потребления, мкА при выключенном стабилизаторе напряжения при включенном стабилизаторе напряжения	I <sub>CCS</sub>	–	10	
		–	100	
Динамический ток потребления, мА, при 33 кГц < f <sub>c</sub> ≤ 36 МГц при 32 ≤ f <sub>c</sub> ≤ 33 кГц	I <sub>OCC</sub>	–	21	
		–	0,4	
Выходная частота HSI RC-генератора, МГц	f <sub>O_HSI</sub>	6	10	
Выходная частота LSI RC-генератора, кГц	f <sub>O_LSI</sub>	10	60	
Выходная частота PLL, МГц максимальная минимальная	f <sub>O_PLL</sub>	36	–	
		–	2	
Параметры АЦП последовательного приближения				
Разрядность АЦП	E <sub>NADC</sub>	12	–	25, 85, – 40
Дифференциальная нелинейность, единица младшего разряда	E <sub>DLADC</sub>	–1	2	
Интегральная нелинейность, единица младшего разряда	E <sub>ILADC</sub>	–3	3	
Ошибка смещения, единица младшего разряда	E <sub>OFFADC</sub>	–6	6	
Ошибка усиления, %	E <sub>GAINADC</sub>	–1	1	



Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Параметры $\Delta\Sigma$ АЦП				
Выходное напряжение VR_1V, В	U <sub>OBIAS</sub>	0,96	1,01	25, 85, – 40
Соотношение сигнал / шум, дБ усиление 0 дБ, A <sub>NADC_D</sub> = 1 В, f <sub>NADCO</sub> = 4 кГц	SNR <sub>D_0</sub>	77	–	
Соотношение сигнал / шум, дБ усиление +6 дБ, A <sub>NADC_D</sub> = 0,5 В, f <sub>NADCO</sub> = 4 кГц	SNR <sub>D_6</sub>	74	–	
Соотношение сигнал / шум, дБ усиление +12 дБ, A <sub>NADC_D</sub> = 0,25 В, f <sub>NADCO</sub> = 4 кГц	SNR <sub>D_12</sub>	71	–	
Соотношение сигнал / шум, дБ усиление +18 дБ, A <sub>NADC_D</sub> = 0,125 В, f <sub>NADCO</sub> = 4 кГц	SNR <sub>D_18</sub>	70	–	
Ошибка усиления предусилителя, дБ	GAIN <sub>ERR</sub>	–	0,25	
Параметры площадок ЖК-дисплея				
Отклонение выходного напряжения при формировании эквивалента U <sub>CC</sub> и U <sub>CC</sub> - U <sub>LCD</sub> , В, на выводах PA, PB(6-14), PC(5-13)	$\Delta U_{OUT1}$	–0,1	0,1	25, 85, – 40
Отклонение выходного напряжения при формировании эквивалента общий и U <sub>LCD</sub> , В, на выводах PA, PB(6-14), PC(5-13)	$\Delta U_{OUT2}$	– 0,1	0,1	
Относительное отклонение выходного напряжения при формировании эквивалента $2/3 \cdot U_{CC}$ и $2/3 \cdot (U_{CC} - U_{LCD})$ , %, на выводах PA, PB(6-14), PC(5-13)	$\Delta U_{отнOUT1}$	–4	0	
Относительное отклонение выходного напряжения при формировании эквивалента $1/3 \cdot U_{CC}$ и $1/3 \cdot (U_{CC} - U_{LCD})$ , %, на выводах PA, PB(6-14), PC(5-13)	$\Delta U_{отнOUT2}$	–3	2	
Относительное отклонение выходного напряжения при формировании эквивалента $1/2 \cdot U_{CC}$ и $1/2 \cdot (U_{CC} - U_{LCD})$ , %, на выводах PA, PB(6-14), PC(5-13)	$\Delta U_{отнOUT3}$	– 3	2	
Частота смены напряжений, Гц на выходах PA, PB(6-14), PC(5-13)	F <sub>VOUT</sub>	60	1000	

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В.

### 32 Справочные данные

Таблица 288 – Справочные параметры микросхем

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Ток потребления батарейного домена, мкА, при: $U_{CC} = 0$ В $V_{UCC} = 3,6$ В	$I_{CCB}$	–	5	25, 85, – 40
Время установления сигналов PBD и PBVD, мкс	$t_{SU(PBD)}$ $t_{SU(PBVO)}$	–	2	
Гистерезис портов ввода/вывода, мВ, на выводах: PA-PC при: ModeRX = 0 ModeRX = 1	$\Delta U_{TH(PA-PF)}$	100 200	400 500	
на выводах: PA – PC при: $U_{CC} = 2,2$ В, PowerTX=00, $C_I = 50$ пФ PowerTX=01, $C_I = 50$ пФ PowerTX=10, $C_I = 50$ пФ PowerTX=11, $C_I = 50$ пФ PowerTX=11, $C_I = 30$ пФ	$t_W(PA-PF)$	– – – – –	10 100 20 10 5	
Тактовые частоты и генераторы				
Время установления сигнала HSIRDY относительно HSION, мкс, при: $U_{CC} = 2,2$ В	$t_{SU(HSI)}$	–	1	25, 85, – 40
Время установления сигнала LSIRDY относительно LSION, мс, при: $U_{CC} = 2,2$ В	$t_{SU(LSI)}$	–	80	
Время установления сигнала HSERDY относительно HSEON, мкс, при: $U_{CC} = 2,2$ В	$t_{SU(HSE)}$	–	$2048/f_{C\_HSE}$	
Время установления сигнала LSERDY относительно LSEON, мкс, при: $U_{CC} = 2,2$ В	$t_{SU(LSE)}$	–	$4096/f_{C\_LSE}$	
Время установления сигнала PLLRDY относительно PLLON, мкс, при: $U_{CC} = 2,2$ В	$t_{SU(PLL)}$	–	100	
Длительность сигнала сброса, мкс, при: $U_{CC} = 2,2$ В	$t_W(RESET)$	20	–	
Время запуска после сброса по POR, мс	$t_{POR}$	–	6	
АЦП				
Время выборки заряда АЦП, нс, при: $U_{CC} = 3,6$ В	$t_{A\_ADC}$	–	$4 \times f_{C\_ADC}$	25, 85, – 40
Время преобразования АЦП, нс при: $U_{CC} = 3,6$ В	$t_{AO\_ADC}$	–	$28 \times f_{C\_ADC}$	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Ток потребления по входу внешней верхней границы опоры АЦП, мкА при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	I <sub>ADC0_VREF+</sub>	–	50	25, 85, – 40
Ток потребления по входу внешней нижней опоры опоры АЦП, мкА при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	I <sub>ADC0_VREF-</sub>	-50	–	
Ток потребления по питанию АЦП, мА при: f <sub>ADC</sub> =14 МГц, U <sub>CCA</sub> =3,6В	I <sub>ОCCADC</sub>	–	3	
Минимальная частота преобразования АЦП, кГц	f <sub>C_ADCMIN</sub>	10	–	
$\Delta\Sigma$ АЦП				
Выходная частота дискретизации, кГц	f <sub>NADCO</sub>	–	4, 8, 16	25, 85, – 40
Входное сопротивление, кОм	R <sub>NADC_D</sub>	30	–	

### 33 Габаритный чертеж микросхемы

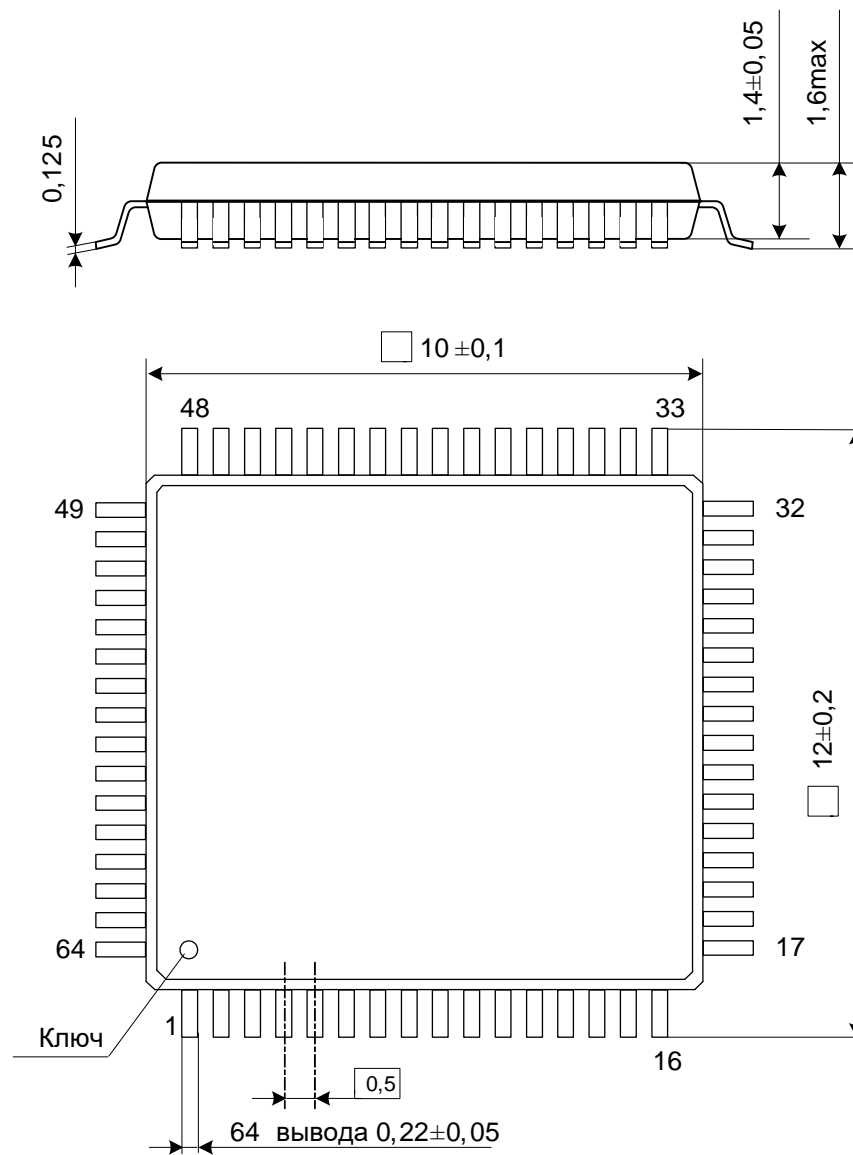


Рисунок 116 – Микросхема в корпусе LQFP64

### 34 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон, °С
K1986BK214	MDR32F23QI ARM	LQFP64	от – 40 до 85

Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	30.07.13	1.0.0	Введена впервые	
2	28.11.2013	1.1.0	Перевод надписей рисунков на русский язык	По тексту
3	15.01.2014	1.2.0	Исправление названий разделов	26, 27
4	21.01.2014	1.3.0	Внесение исправлений в таблицу 3	18
5	31.01.2014	1.4.0	Исправлена разрядность CNT, PSG, ARR	145, 146
6	04.03.2014	1.4.1	В таблице 170 удалены 2 последние строки	255
7	28.05.2014	1.5.1	Добавлено примечание 2 к таблице 184	273
8	25.06.2014	1.6.0	Испралена маркировка. Исправлено обозначение спецификации	1, 280
9	08.07.2014	1.7.0	Добавлено значение статического электричества	277
10	09.07.2014	1.8.0	Исправлен габаритный чертеж	281
11	12.01.2015	1.9.0	Добвлены типовые схемы включения для учета электроэнергии	56, 57
12	26.06.2015	1.10.0	Изменено обозначение микросхем	По тексту
13	01.10.2015	1.11.0	Изменено обозначение микросхем	По тексту
14	14.10.2015	1.12.0	Исправлено функциональное назначение Внесены исправления в подраздел «Система питания» Исправлена таблица электрических параметров	1 12 274, 275
15	02.03.2017	2.0.0	Изменено обозначение микросхем Таблицы параметров приведены в соответствие с ТУ	По тексту 271 – 276
16	24.05.2018	2.1.0	Исправление замечаний	По тексту
17	29.03.2021	2.2.0	Изменение раздела 7.5 Внесения изменений в рисунки: 9, 17, 29, 30, 31; Внесение изменений в таблицы: 1, 28, 64, 75, 76, 81, 96, 129, 135, 144, 158, 171, 182; Внесение изменений в разделы: 7.6.7.1, 7.7.2.4, 7.11, 7.13.6, 7.13.6.1, 7.14.6, 7.15, 7.16.6, 7.16.6.4	19, 26, 62, 124-126 8, 68, 95, 105, 107, 111, 121, 189, 193, 216, 229, 252, 263 30, 38, 113, 127, 128, 155, 162, 170, 171
18	05.05.2021	2.2.1	Добавлено международное обозначение	260
19	15.09.2021	2.3.0	Исправление замечаний Замена термина «пластиковый» на «пластмассовый» Добавлено примечание после таблицы 264 (перенесена из подраздела 25)	По тексту 1, 6 260

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			Таблица 4 – Дополнено описание режимов «UART-загрузчик»	33
			Таблица 106 – Исправлено описание бита PLL_CPU_RLD	117
			Таблица 184 – описание полей NINV и INV исправлено	172
			Исправлено описание раздела 0	152
			Скорректированы рисунки 43, 44	152
			Рисунок 66 – добавлен блок с сигналами SEL_VREF_BUF, TS_BUF_EN	179
			Подразделы 19.1, 19.2, 19.4, 19.5 – описание скорректировано с учетом информации о бите SEL_VREF_BUF	180 – 182
			Таблица 210 – дополнено описание бита SEL_VREF_BUF в регистре ADC1_TRIM	189
			Подраздел 26.7 – дополнена информация о прерываниях по уровню сигнала	274
20	16.02.2022	2.4.0	Исправление опечаток	По тексту
			Таблица 1 – добавлено описание выводов JTAG_EN, GND, AGND	17
			Таблицы 6 – 11 – описание уточнено	35-37
			Подразделы 9.1, 10.2.1, 19.4, 19.5, 20 – описания дополнены	34, 40, 181 182, 190
			Добавлен подраздел «Логика работы прерываний контроллера NVIC»	268
21	11.09.2023	2.5.0	Исправление опечаток	По тексту
			Информация о «PrimeCell» исключена	По тексту
			Информация об интерфейсе DMA исключена	По тексту
			Добавлено УГО	16
			Таблица 1 обновлена	17
			Таблица 2 – Назначение выводов по блокам добавлена	19
			Раздел 7 – информация о выводе AU <sub>CC</sub> дополнена	28
			Пункты 10.2.1-10.2.4 скорректированы	40-42
			Рисунок 20 обновлен	88
			Таблица 44 описания регистров скорректированы	89
			Описания бит регистров ADCUI_F0CTR, ADCUI_F0WC, ADCUI_F0VC, ADCUI_F0AC, ADCUI_F0MD0 скорректированы	93-97
			Подраздел 13.2:	
			– добавлена информация о буфере FIFO;	105,
			– таблица 97, формула (1) скорректированы,	107

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			– формула (2) ( $V_{COR}$ ) добавлена Рисунки 30, 31 скорректированы Таблицы 116, 125, 131, 136, 139, 141 – описания битов скорректированы Подраздел 15.1 скорректирован Таблица 164 обновлена Раздел 18 скорректирован Таблица 227 описание бит DSS исправлено Пункт 22.11.3 – примечание скорректировано	108 114, 125 121, 126-133  125 141 145-177 216 236
22	19.06.2024	2.6.0	Таблица 46 – примечание скорректировано Таблица 120 – примечание добавлено Пункт 18.2.1 – информация о тактовой частоте TIM_CLK дополнена Пункт 22.11.4 скорректирован Раздел 28 обновлен	90 122 147  237 282-288