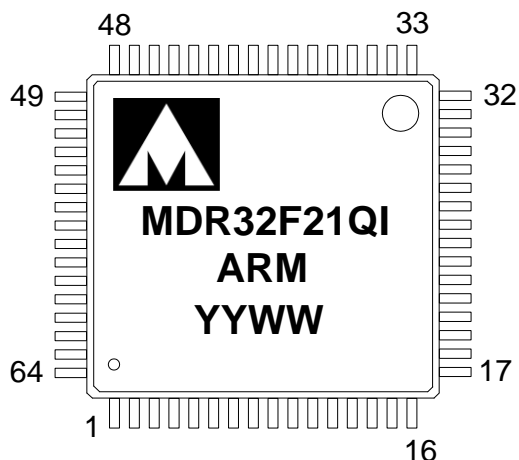




Микросхема контроллера для трехфазного электросчетчика К1986ВК234



Основные параметры микросхемы:

- Напряжение источника питания от 3,0 до 3,6 В;
- 32-разрядная RISC-архитектура ARM Cortex-M0;
- Встроенная память программ 128 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:

- ARM 32-битное RISC-ядро Cortex™-M0, тактовая частота до 36 МГц;
- умножение за один цикл.

Память:

- встроенная энергонезависимая память программ типа FLASH размером 128 Кбайт;
- встроенное ОЗУ размером 16 Кбайт.

Питание и тактовая частота:

- внешнее питания 3,0...3,6 В;
- встроенный регулятор напряжения на 1,8 В 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 семи каналов);
- 12-разрядный АЦП (до восьми каналов);
- измеряемый диапазон от 0 до 3,6 В;
- температурный сенсор.

Периферия:

- контроллер прямого доступа в память с функциями передачи Периферия-Память, Память-Память;
- контроллеры интерфейсов UART, SPI;
- до 39 пользовательских линий ввода/вывода;
- два блока 16-разрядных таймеров с четырьмя каналами захвата событий и ШИМ;
- сторожевой таймер;
- блок подсчета CRC с изменяемым полиномом.

Режим отладки:

- последовательный отладочный интерфейс SWD.

Содержание

1	Введение	15
2	Структурная схема.....	16
3	Условное графическое изображение	17
4	Описание выводов	18
5	Расположение выводов в корпусе.....	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	Базовые адреса процессора.....	33
9	Загрузочное ПЗУ и режимы работы микросхемы.....	34
9.1	UART-загрузчик.....	35
9.1.1	Параметры связи по UART.....	35
9.1.2	Протокол обмена по UART	35
9.1.3	Синхронизация с внешним устройством	36
9.1.4	Команда CMD_SYNC	36
9.1.5	Команда CMD_CR.....	36
9.1.6	Команда CMD_BAUD.....	37
9.1.7	Команда CMD_LOAD.....	37
9.1.8	Команда CMD_VFY	38
9.1.9	Команда CMD_RUN.....	38
9.1.10	Прием параметров команды	39
9.1.11	Сообщения об ошибках	39
9.1.12	Ошибка ERR_CHN	39
9.1.13	Ошибка ERR_CMD	39
9.1.14	Ошибка ERR_BAUD	39
10	Контроллер FLASH-памяти программ	40
10.1	Работа Flash-памяти программ в обычном режиме.....	40
10.2	Работа Flash-памяти программ в режиме программирования	41
10.2.1	Стирание блока памяти размером 2 Кбайт или 32 Кбайт.	42
10.2.2	Стирание страницы памяти размером 512 байт.	42
10.2.3	Запись 32-битного слова в память.	43
10.2.4	Чтение 32-битного слова из памяти.	44
10.3	Описание регистров управления контроллера Flash-памяти программ.....	45
10.3.1	EEPROM_CMD.....	45
10.3.2	EEPROM_ADR	46
10.3.3	EEPROM_DI.....	47
10.3.4	EEPROM_DO	47
10.3.5	EEPROM_KEY.....	47
11	Процессорное ядро	48
11.1	Структурная схема процессора	49

11.2	Программная модель	50
11.3	Стек	50
11.4	Регистры ядра	51
11.4.1	Регистры общего назначения R0-R12	51
11.4.2	Указать стека SP R13	51
11.4.3	Регистр связи LR R14	51
11.4.4	Счетчик команд PC R15	52
11.4.5	Программный регистр состояния PSR	52
11.4.6	APSR	52
11.4.7	IPSR	53
11.4.8	EPSR	53
11.4.9	PRIMASK	54
11.4.10	CONTROL	54
11.5	Типы данных	55
12	Система команд	56
12.1	Встроенные функции	58
12.2	Описание инструкций	59
12.2.1	Операнды	59
12.2.2	Ограничения на использование PC и SP	59
12.2.3	Операции сдвига	59
12.2.4	Выравнивание адресов	61
12.2.5	Адресация относительно счетчика команд PC	62
12.2.6	Условное исполнение	62
12.3	Команды доступа к памяти	64
12.3.1	ADR	64
12.3.2	LDR и STR, непосредственно заданное смещение	65
12.3.3	LDR и STR, смещение задано в регистре	66
12.3.4	LDR, адресация относительно счетчика команд PC	67
12.3.5	LDM и STM	67
12.3.6	PUSH и POP	68
12.4	Инструкции обработки данных	70
12.4.1	ADD, ADC, SUB, SBC и RSB	70
12.4.2	AND, ORR, EOR, BIC	72
12.4.3	ASR, LSL, LSR, ROR	73
12.4.4	CMP и CMN	74
12.4.5	MOV и MVN	75
12.4.6	MULS	76
12.4.7	REV, REV16, REVSH	77
12.4.8	SXT и UXT	78
12.4.9	TST	78
12.5	Инструкции передачи управления	79
12.5.1	B, BL, BX и BLX	79
12.6	Прочие инструкции	81
12.6.1	BKPT	81

12.6.2	CPS	82
12.6.3	DMB	82
12.6.4	DSB	83
12.6.5	ISB	83
12.6.6	MRS	83
12.6.7	MSR	84
12.6.8	NOP	85
12.6.9	SEV	85
12.6.10	SVC	85
12.6.11	WFE	86
12.6.12	WFI	87
13	Блок АЦП для измерения напряжений и токов в электрической сети.....	88
13.1	Описание регистров управления блока семиканального $\Delta\Sigma$ АЦП	90
13.1.1	ADCUI_CTRL1	93
13.1.2	ADCUI_CTRL2	95
13.1.3	ADCUI_CTRL3	96
13.1.4	ADCUI_F0CTR	96
13.1.5	ADCUI_F0WC	97
13.1.6	ADCUI_F0WATTP	98
13.1.7	ADCUI_F0WATTN	98
13.1.8	ADCUI_F0VC	98
13.1.9	ADCUI_F0VARP	99
13.1.10	ADCUI_F0VARN	99
13.1.11	ADCUI_F0AC	99
13.1.12	ADCUI_F0VR	100
13.1.13	ADCUI_F0MD0	100
13.1.14	ADCUI_F0MD1	101
13.1.15	ADCUI_F0VPEAK	101
13.1.16	ADCUI_F0IPEAK	102
13.1.17	ADCUI_F0VDAT	102
13.1.18	ADCUI_F0I0DAT	102
13.1.19	ADCUI_F0I3DAT	103
13.1.20	ADCUI_F0VRMS	103
13.1.21	ADCUI_F0VRMS2	103
13.1.22	ADCUI_F0IRMS	104
13.1.23	ADCUI_F0IRMS2	104
13.1.24	ADCUI_F0STAT	104
13.1.25	ADCUI_F0MASK	106
13.1.26	ADCUI_F1CTR	107
13.1.27	ADCUI_F1WC	108
13.1.28	ADCUI_F1WATTP	109
13.1.29	ADCUI_F1WATTN	109
13.1.30	ADCUI_F1VC	109
13.1.31	ADCUI_F1VARP	110

13.1.32	ADCUI_F1VARN.....	110
13.1.33	ADCUI_F1AC.....	110
13.1.34	ADCUI_F1VR.....	111
13.1.35	ADCUI_F1MD0.....	111
13.1.36	ADCUI_F1MD1.....	112
13.1.37	ADCUI_F1MD2.....	112
13.1.38	ADCUI_F1VPEAK.....	113
13.1.39	ADCUI_F1IPEAK.....	113
13.1.40	ADCUI_F1VDAT.....	113
13.1.41	ADCUI_F1IDAT.....	114
13.1.42	ADCUI_F1VRMS.....	114
13.1.43	ADCUI_F1VRMS2.....	114
13.1.44	ADCUI_F1IRMS.....	114
13.1.45	ADCUI_F1IRMS2.....	115
13.1.46	ADCUI_F1STAT.....	115
13.1.47	ADCUI_F1MASK.....	117
13.1.48	ADCUI_F2CTR.....	118
13.1.49	ADCUI_F2WC.....	119
13.1.50	ADCUI_F2WATTP.....	119
13.1.51	ADCUI_F2WATTN.....	120
13.1.52	ADCUI_F2VC.....	120
13.1.53	ADCUI_F2VARP.....	120
13.1.54	ADCUI_F2VARN.....	121
13.1.55	ADCUI_F2AC.....	121
13.1.56	ADCUI_F2VR.....	121
13.1.57	ADCUI_F2MD0.....	122
13.1.58	ADCUI_F2MD1.....	123
13.1.59	ADCUI_F2MD2.....	123
13.1.60	ADCUI_F2VPEAK.....	123
13.1.61	ADCUI_F2IPEAK.....	124
13.1.62	ADCUI_F2VDAT.....	124
13.1.63	ADCUI_F2IDAT.....	124
13.1.64	ADCUI_F2VRMS.....	125
13.1.65	ADCUI_F2VRMS2.....	125
13.1.66	ADCUI_F2IRMS.....	125
13.1.67	ADCUI_F2IRMS2.....	125
13.1.68	ADCUI_F2STAT.....	126
13.1.69	ADCUI_F2MASK.....	127
13.1.70	ADCUI_CCAL1.....	128
13.1.71	ADCUI_CCAL2.....	129
13.1.72	ADCUI_CCAL3.....	129
13.1.73	ADCUI_CCAL4.....	129
13.2	Алгоритмы вычисления окончательных результатов и их соответствия внешним сигналам.....	130

13.2.1	Типовая схема включения для учета электроэнергии по трем фазам	136
13.2.2	Типовая схема включения для учета электроэнергии по одной фазе	137
13.2.3	Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта	138
14	Аппаратный блок вычисления CRC	139
14.1	Описание регистров управления блока CRC	139
14.1.1	CRC_CTRL	139
14.1.2	CRC_STAT	140
14.1.3	CRC_DATA1	141
14.1.4	CRC_VAL	141
14.1.5	CRC_POL	141
15	Сигналы тактовой частоты	142
15.1	Встроенный RC-Генератор HSI	143
15.2	Встроенный RC-генератор LSI	143
15.3	Внешний осциллятор HSE	144
15.4	Внешний осциллятор LSE	144
15.5	Встроенный блок умножения системной тактовой частоты	144
15.6	Описание регистров блока контроллера тактовой частоты	144
15.6.1	CLOCK_STATUS	145
15.6.2	PLL_CONTROL	145
15.6.3	HS_CONTROL	146
15.6.4	CPU_CLOCK	146
15.6.5	PER1_CLOCK	147
15.6.6	ADC_CLOCK	148
15.6.7	RTC_CLOCK	149
15.6.8	PER2_CLOCK	149
15.6.9	TIM_CLOCK	150
15.6.10	UART_CLOCK	151
15.6.11	SSP_CLOCK	152
16	Батарейный домен и часы реального времени	153
16.1	Часы реального времени	153
16.2	Регистры аварийного сохранения	154
16.3	Описание регистров блока батарейного домена	155
16.3.1	BKP_REG_00	155
16.3.2	BKP_REG_01	155
16.3.3	BKP_REG_02	155
16.3.4	BKP_REG_03	155
16.3.5	BKP_REG_04	155
16.3.6	BKP_REG_05	155
16.3.7	BKP_REG_06	155
16.3.8	BKP_REG_07	155
16.3.9	BKP_REG_08	155
16.3.10	BKP_REG_09	155
16.3.11	BKP_REG_0A	155

16.3.12	BKP_REG_0B.....	155
16.3.13	BKP_REG_0C.....	155
16.3.14	BKP_REG_0D.....	155
16.3.15	BKP_REG_0E.....	156
16.3.16	BKP_REG_0F.....	157
16.3.17	RTC_CNT.....	160
16.3.18	RTC_DIV.....	160
16.3.19	RTC_PRL.....	161
16.3.20	RTC_ALRM.....	161
16.3.21	RTC_CS.....	161
17	Порты ввода-вывода.....	163
17.1	Описание регистров портов ввода-вывода.....	165
17.1.1	PORTx_RXTX.....	166
17.1.2	PORTx_OE.....	166
17.1.3	PORTx_FUNC.....	167
17.1.4	PORTx_ANALOG.....	167
17.1.5	PORTx_PULL.....	167
17.1.6	PORTx_PD.....	168
17.1.7	PORTx_PWR.....	168
17.1.8	PORTx_GFEN.....	169
17.1.9	PORTx_SETTX.....	169
17.1.10	PORTx_CLRTX.....	170
18	Детектор напряжения питания.....	171
18.1	Описание регистров блока PVD.....	171
18.1.1	PVDCS.....	171
19	Таймеры общего назначения.....	174
19.1	Основные характеристики.....	174
19.2	Структурная схема.....	175
19.3	Базовый блок таймера.....	176
19.3.1	Инициализация тактирования таймера.....	176
19.3.2	Инициализация основного счетчика таймера.....	176
19.3.3	Режимы счета.....	177
19.3.4	Тактовая частота F_{DTS}	179
19.4	Источники событий для счета.....	180
19.4.1	Внутренний тактовый сигнал (TIM_CLKd).....	181
19.4.2	Событие в другом таймере (CNT==ARR).....	182
19.4.3	Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CHu1.....	183
19.4.4	Внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR.....	184
19.5	Режим захвата.....	185
19.6	Режим ШИМ.....	186
19.6.1	Генератор опорного сигнала REF.....	187
19.6.2	Генератор «мертвой зоны».....	188

19.6.3	Выходные блоки	189
19.7	Блок цифрового фильтра.....	190
19.8	Флаги состояний, прерываний и запросов DMA.....	191
19.8.1	Флаги состояний.....	192
19.8.2	Прерывания.....	192
19.8.3	Запросы DMA	192
19.9	Примеры	192
19.9.1	Обычный счетчик	192
19.9.2	Режим захвата	193
19.9.3	Режим ШИМ	194
19.10	Описание регистров блока таймера	195
19.10.1	CNT	196
19.10.2	PSG.....	196
19.10.3	ARR.....	197
19.10.4	CNTRL	197
19.10.5	CCRy.....	198
19.10.6	CCRy1	199
19.10.7	CHy_CNTRL	199
19.10.8	CHy_CNTRL1	201
19.10.9	CHy_CNTRL2	202
19.10.10	CHy_DTG	203
19.10.11	BRKETR_CNTRL	203
19.10.12	STATUS	204
19.10.13	IE	206
19.10.14	DMA_RE.....	207
20	Контроллер АЦП	209
20.1	Преобразование внешнего канала.....	210
20.2	Последовательное преобразование нескольких каналов.....	210
20.3	Преобразование с контролем границ.....	211
20.4	Датчик опорного напряжения.....	211
20.5	Датчик температуры.....	212
20.6	Время заряда внутренней емкости.....	213
20.7	Описание регистров блока контроллера АЦП.....	214
20.7.1	ADCx_CFG.....	214
20.7.2	ADC1_H_LEVEL	217
20.7.3	ADC1_L_LEVEL.....	217
20.7.4	ADC1_RESULT.....	218
20.7.5	ADC1_STATUS.....	218
20.7.6	ADC1_CHSEL	219
20.7.7	ADC1_TRIM.....	219
21	Контроллер SSP	220
21.1	Основные характеристики модуля SSP	220
21.2	Программируемые параметры.....	221
21.3	Характеристики интерфейса SPI.....	221

21.4	Характеристики интерфейса Microwire	221
21.5	Характеристики интерфейса SSI	222
21.6	Описание функционирования.....	222
21.6.1	Общий обзор модуля SSP	222
21.6.2	Блок формирования тактового сигнала.....	223
21.6.3	Буфер FIFO передатчика.....	223
21.6.4	Буфер FIFO приемника	223
21.6.5	Блок приема и передачи данных	224
21.6.6	Блок формирования прерываний	224
21.6.7	Конфигурирование приемопередатчика	225
21.6.8	Разрешение работы приемопередатчика.....	225
21.6.9	Соотношения между тактовыми сигналами	225
21.6.10	Программирование регистра управления SSPCR0	226
21.6.11	Программирование регистра управления SSPCR1	227
21.6.12	Формирование тактового сигнала обмена данными.....	227
21.6.13	Формат информационного кадра.....	227
21.6.14	Формат синхронного обмена SSI фирмы Texas Instruments	228
21.6.15	Формат синхронного обмена SPI фирмы Motorola.....	229
21.6.16	Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0.	229
21.6.17	Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1.	231
21.6.18	Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0.	232
21.6.19	Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1.	233
21.6.20	Формат синхронного обмена Microwire фирмы National Semiconductor	234
21.6.21	Примеры конфигурации модуля в ведущем и ведомом режимах....	237
21.6.22	Интерфейс прямого доступа к памяти	239
21.7	Прерывания	241
21.7.1	SSPRXINTR.....	242
21.7.2	SSPTXINTR.....	242
21.7.3	SSPRORINTR.....	242
21.7.4	SSPRTINTR.....	242
21.7.5	SSPINTR	242
21.8	Программное управление модулем.....	242
21.8.1	Общая информация	242
21.8.2	Описание регистров контроллера SSP	243
21.8.2.8	SSP_MIS	248
22	Контроллер UART	250
22.1	Основные характеристики модуля UART	250
22.2	Программное управление скоростью обмена	250
22.3	Программируемые параметры.....	251
22.4	Отличия от контроллера UART 16C650	251
22.5	Функциональные возможности	252
22.6	Модуль приемопередатчика:	252
22.7	Описание функционирования блока UART	253

22.7.1	Генератор тактового сигнала приемопередатчика.....	254
22.7.2	Буфер FIFO передатчика.....	254
22.7.3	Буфер FIFO приемника	254
22.7.4	Блок передатчика.....	254
22.7.5	Блок приемника	254
22.7.6	Блок формирования прерываний	255
22.7.7	Блок и регистры синхронизации.....	255
22.8	Описание функционирования ИК кодека IrDA SIR.....	255
22.8.1	Кодер ИК-передатчика.....	256
22.8.2	Декодер ИК-приемника	256
22.9	Описание работы	257
22.9.1	Тактовые сигналы.....	257
22.9.2	Работа универсального асинхронного приемопередатчика.....	257
22.9.3	Дробный коэффициент деления.....	258
22.9.4	Передача и прием данных.....	258
22.9.5	Биты ошибки	259
22.9.6	Бит переполнения буфера.....	260
22.9.7	Запрет буфера FIFO.....	260
22.9.8	Проверка по шлейфу	260
22.9.9	Работа кодека ИК-обмена данными IrDA SIR.....	260
22.9.10	Проверка по шлейфу	261
22.9.11	Линии управления модемом.....	262
22.9.12	Аппаратное управление потоком данных	262
22.9.13	Управление потоком данных по линии RTS	263
22.9.14	Управление потоком данных по линии CTS	264
22.10	Интерфейс прямого доступа к памяти.....	264
22.11	Прерывания	266
22.11.1	UARTMSINTR	267
22.11.2	UARTRXINTR	267
22.11.3	UARTTXINTR.....	268
22.11.4	UARTRTINTR.....	268
22.11.5	UARTEINTR	268
22.11.6	UARTINTR.....	269
22.12	Программное управление модулем.....	269
22.12.1	Общая информация	269
22.12.2	Обобщенные данные о регистрах контроллеров UART.....	270
23	Контроллер прямого доступа в память DMA	285
23.1	Основные свойства контроллера DMA	285
23.2	Термины и определения.....	285
23.3	Функциональное описание	287
23.3.1	Распределение каналов DMA	287
23.3.2	Блок, подключенный к шине APB.....	288
23.3.3	Блок, подключенный к шине AHB	288
23.3.4	Управляющий блок DMA	288

23.3.5	Пример использования блока DMA	289
23.3.6	Типы передач	289
23.3.7	Разрядность передач данных	289
23.3.8	Управление защитой данных	290
23.3.9	Инкремент адреса	290
23.3.10	Управление DMA	291
23.3.11	Правила обмена данными	291
23.3.12	Диаграммы работы контроллера DMA	293
23.3.13	Импульсный запрос на обработку	294
23.3.14	Запрос на обработку по уровню	294
23.3.15	Флаги завершения	295
23.3.16	Флаги ожидания запроса на обработку	296
23.3.17	Правила арбитража DMA	299
23.3.18	Структура управляющих данных канала	311
23.4	Описание регистров контроллера DMA	322
23.4.1	DMA_STATUS	323
23.4.2	DMA_CFG	324
23.4.3	CTRL_BASE_PTR	325
23.4.4	ALT_CTRL_BASE_PTR	326
23.4.5	DMA_WAITONREQ_STATUS	327
23.4.6	CHNL_SW_REQUEST	328
23.4.7	CHNL_USEBURST_SET	328
23.4.8	CHNL_USEBURST_CLR	330
23.4.9	CHNL_REQ_MASK_SET	331
23.4.10	CHNL_REQ_MASK_CLR	332
23.4.11	CHNL_ENABLE_SET	333
23.4.12	CHNL_ENABLE_CLR	334
23.4.13	CHNL_PRI_ALT_SET	335
23.4.14	CHNL_PRI_ALT_CLR	336
23.4.15	CHNL_PRIORITY_SET	337
23.4.16	CHNL_PRIORITY_CLR	338
23.4.17	ERR_CLR	339
24	Прерывания и исключения	340
24.1	Типы исключений	340
24.1.1	RESET	340
24.1.2	NON MASKABLE INTERRUPT (NMI)	340
24.1.3	HARD FAULT	340
24.1.4	SVCALL	340
24.1.5	PendSV	341
24.1.6	SysTick	341
24.2	Прерывания (IRQ)	341
24.3	Обработчики исключений	342
24.4	Приоритеты исключений	343
24.5	Вход в обработчик и выход из обработчика	344

24.5.1	Приоритетное прерывание	344
24.5.2	Возврат.....	344
24.5.3	Передача управления без восстановления контекста (tail-chaining)	344
24.5.4	Запоздавшее исключение (late-arriving exception)	344
24.5.5	Вход в процедуру обработки исключения	345
24.5.6	Возврат из обработчика исключения	346
25	Управление электропитанием	347
25.1	Переход в режим пониженного энергопотребления	347
25.2	Ожидание прерывания	347
25.3	Ожидание события.....	347
25.4	Выход из ожидания по команде WFE.....	347
25.5	Переход в режим ожидания по выходу из обработчика исключения (режим sleep-on-exit)	348
25.6	Выход из состояния ожидания	348
25.7	Выход из ожидания по команде WFI и в режиме sleep-on-exit.....	348
25.8	Рекомендации по программированию режима энергопотребления.....	348
26	Контроллер прерываний NVIC	349
26.1	Логика работы прерываний контроллера NVIC	349
26.2	Регистр разрешения прерываний	352
26.3	Регистр запрета прерываний.....	353
26.4	Регистр установки состояния ожидания для прерывания	353
26.5	Регистр сброса состояния ожидания для прерывания	354
26.6	Регистры приоритета прерываний	354
26.7	Прерывания, срабатывающие по уровню сигнала	355
26.8	Аппаратное и программное управление прерываниями.....	355
26.9	Рекомендации по работе с контроллером прерываний.....	356
27	Блок управления системой ядра.....	357
27.1	Упрощенный доступ к регистрам блока управления системой из среды разработки программного обеспечения.....	357
27.1.1	Регистр идентификации процессора	357
27.1.2	Регистр управления прерываниями	358
27.1.3	Регистр управления прерываниями и программного сброса	359
27.1.4	Регистр управления системой	360
27.1.5	Регистр конфигурации и управления	360
27.1.6	Регистры приоритета системных обработчиков	361
27.1.7	Регистр №2 приоритета системных обработчиков	361
27.1.8	Регистр №3 приоритета системных обработчиков	362
27.1.9	Рекомендации по программированию блока управления системой	362
28	Сторожевые таймеры	363
28.1	Блок сторожевого таймера IWDG.....	363
28.1.1	Описание регистров блока сторожевого таймера IWDG	364
28.2	Блок сторожевого таймера WWDG	368
28.2.1	Описание регистров блока сторожевого таймера WWDG.....	368
29	Типовая схема подключения питания	370

30	Предельно-допустимые характеристики микросхемы	371
31	Электрические параметры микросхемы.....	373
32	Справочные данные.....	375
33	Габаритный чертеж микросхемы	377
34	Информация для заказа	378

1 Введение

Микросхемы этой серии являются микроконтроллерами со встроенной Flash-памятью программ и построены на базе низкопотребляющего процессорного RISC-ядра ARM Cortex-M0. Микросхема работает на тактовой частоте до 36 МГц и содержит 128 Кбайт Flash-памяти и 16 Кбайт ОЗУ. Микросхема включает в себя развитую периферию для построения счетчиков электроэнергии одно- и трехфазных сетей. Периферия включает в себя семь каналов для трехфазной сети (или три канала для однофазной сети) 24-битных независимых $\Delta\Sigma$ АЦП. Каждый канал АЦП имеет предусилитель, фазовую подстройку (для коррекции фазы не хуже 0,1), а также аппаратный блок для вычисления среднеквадратического значения сигнала. Каждый канал АЦП может быть включен или отключен независимо от других каналов и имеет отдельный канал прямого доступа в память. Еще один дополнительный 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 микросхемы входит два UART-интерфейса и один SPI-интерфейс. Микросхема содержит два 16-разрядных таймера с четырьмя каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Также микросхема содержит системный 24-разрядный таймер и два сторожевых таймера.

Встроенные RC-генераторы HSI (8 МГц) и LSI (40 кГц), 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. Контроллер DMA позволяет ускорить обмен информацией между ОЗУ и периферией без участия процессорного ядра.

Встроенный регулятор для формирования питания внутренней цифровой части формирует напряжения 1,8 В и не требует дополнительных внешних элементов. Таким образом, для работы микросхемы достаточно одного внешнего напряжения питания в диапазоне от 3,0 до 3,6 В. Также в микросхеме реализован батарейный домен, работающий от внешней батареи при отсутствии основного питания. В батарейном домене могут быть сохранены специальные флаги, а также работают часы реального времени. Встроенные детекторы напряжения питания могут отслеживать уровень внешнего основного питания, уровень напряжения питания на батарее. Аппаратные схемы сброса по просадке питания позволяют исключить сбойную работу микросхемы при выходе уровня напряжения питания за допустимые пределы.

2 Структурная схема

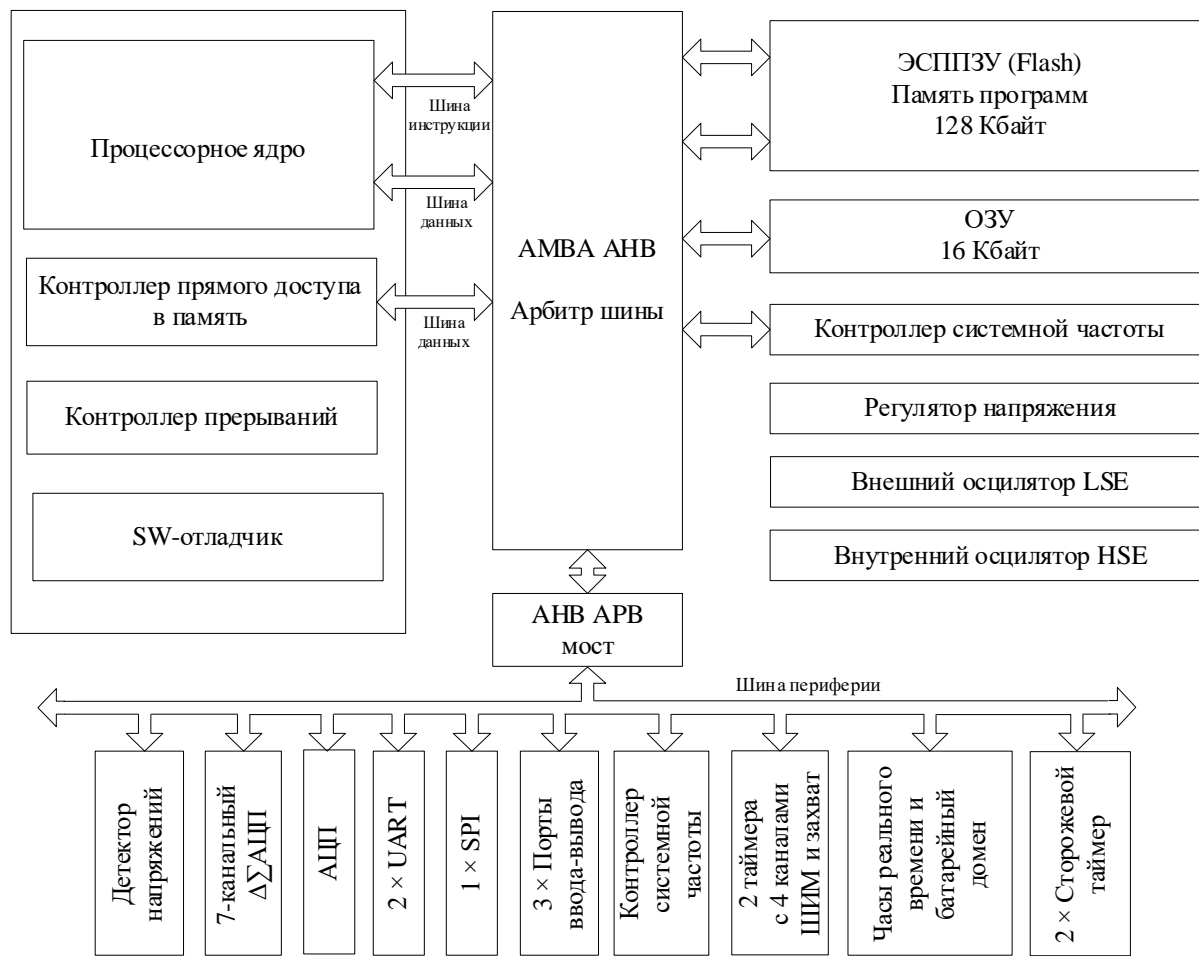


Рисунок 1 – Структурная схема микросхемы

3 Условное графическое изображение

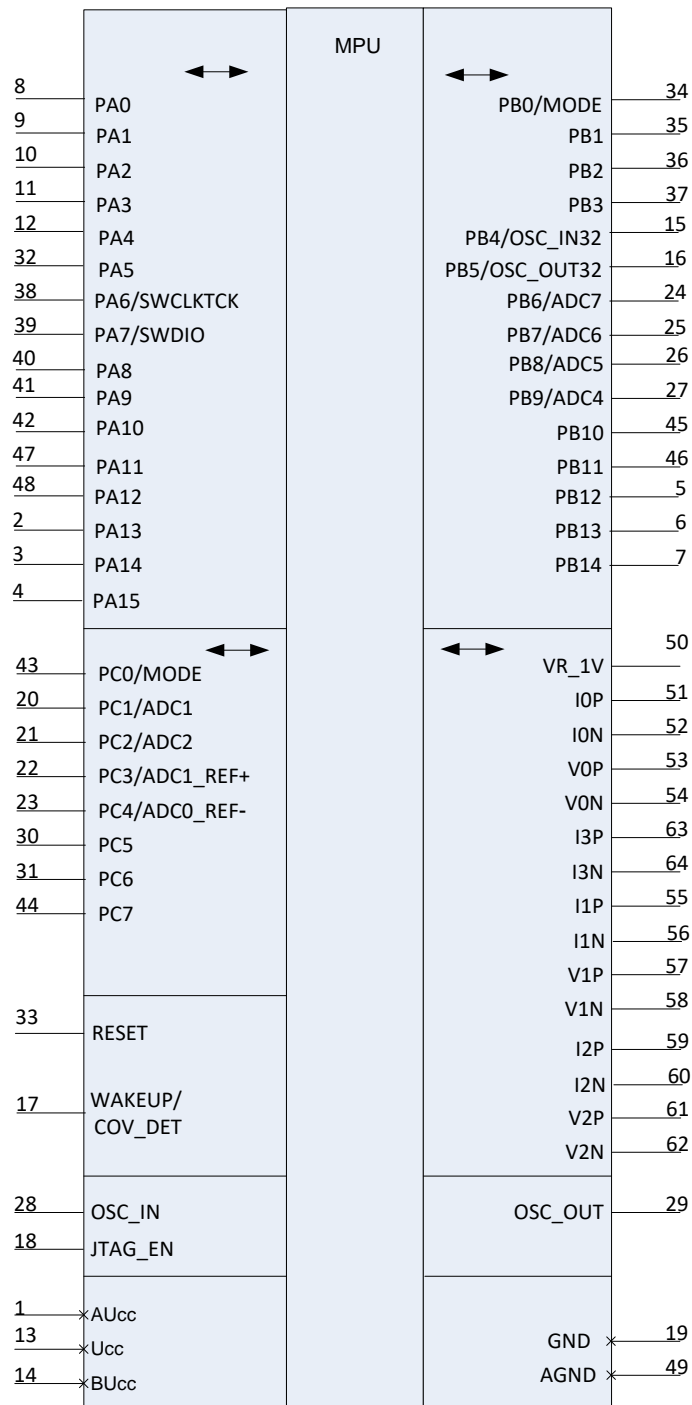


Рисунок 2 – Условное графическое изображение

4 Описание выводов

Таблица 1 – Описание выводов

Номер вывода	Обозначение вывода	Функциональное назначение вывода
1	AU _{CC}	Аналоговое питание $\Delta\Sigma$ АЦП (2,4...3,6) В
2	PA13	Вход/выход 13 порта А
3	PA14	Вход/выход 14 порта А
4	PA15	Вход/выход 15 порта А
5	PB12	Вход/выход 12 порта В
6	PB13	Вход/выход 13 порта В
7	PB14	Вход/выход 14 порта В
8	PA0	Вход/выход 0 порта А
9	PA1	Вход/выход 1 порта А
10	PA2	Вход/выход 2 порта А
11	PA3	Вход/выход 3 порта А
12	PA4	Вход/выход 4 порта А
13	U _{CC}	Питание (3,0...3,6) В
14	BU _{CC}	Батарейное питание (1,8...3,6) В
15	PB4	Вход/выход 4 порта В
16	PB5	Вход/выход 5 порта В
17	WAKEUP/ COV_DET	Сигнал внешнего выхода из режима Standby
18	JTAG_EN	Тестовый вывод. Рекомендуется не подсоединять или подключать к шине общий
19	GND	Общий
20	PC1	Вход/выход 1 порта С
21	PC2	Вход/выход 2 порта С
22	PC3	Вход/выход 3 порта С
23	PC4	Вход/выход 4 порта С
24	PB6	Вход/выход 6 порта В
25	PB7	Вход/выход 7 порта В
26	PB8	Вход/выход 8 порта В
27	PB9	Вход/выход 9 порта В
28	OSC_IN	Вход генератора HSE
29	OSC_OUT	Выход генератора HSE
30	PC5	Вход/выход 5 порта С
31	PC6	Вход/выход 6 порта С
32	PA5	Вход/выход 5 порта А
33	RESET	Сигнал внешнего сброса
34	PB0/MODE	Вход/выход 0 порта В
35	PB1	Вход/выход 1 порта В
36	PB2	Вход/выход 2 порта В
37	PB3	Вход/выход 3 порта В
38	PA6/SWCLKTCK	Вход/выход 6 порта А

Номер вывода	Обозначение вывода	Функциональное назначение вывода
39	PA7/SWDIO	Вход/выход 7 порта А
40	PA8	Вход/выход 8 порта А
41	PA9	Вход/выход 9 порта А
42	PA10	Вход/выход 10 порта А
43	PC0/MODE1	Вход/выход 0 порта С
44	PC7	Вход/выход 7 порта С
45	PB10	Вход/выход 10 порта В
46	PB11	Вход/выход 11 порта В
47	PA11	Вход/выход 11 порта А
48	PA12	Вход/выход 12 порта А
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	I1P	Положительный вход канала 1 тока
56	I1N	Отрицательный вход канала 1 тока
57	V1P	Положительный вход канала 1 напряжения
58	V1N	Отрицательный вход канала 1 напряжения
59	I2P	Положительный вход канала 2 тока
60	I2N	Отрицательный вход канала 2 тока
61	V2P	Положительный вход канала 2 напряжения
62	V2N	Отрицательный вход канала 2 напряжения
63	I3P	Положительный вход канала 3 тока
64	I3N	Отрицательный вход канала 3 тока

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				
8	РА0	Вход/выход 0 порта А		
		Аналоговая	–	–
		Основная	TMR0_CH1	Вход/выход канала 1 таймера 0
		Альтернативная	–	–
9	РА1	Вход/выход 1 порта А		
		Аналоговая	–	–
		Основная	TMR0_CH1N	Инверсный выход канала 1 таймера 0
		Альтернативная	–	–
10	РА2	Вход/выход 2 порта А		
		Аналоговая	–	–
		Основная	TMR0_CH2	Вход/выход канала 2 таймера 0
		Альтернативная	–	–
11	РА3	Вход/выход 3 порта А		
		Аналоговая	–	–
		Основная	TMR0_CH2N	Инверсный выход канала 2 таймера 0
		Альтернативная	–	–
12	РА4	Вход/выход 4 порта А		
		Аналоговая	–	–
		Основная	TMR0_CH3	Вход/выход канала 3 таймера 0
		Альтернативная	–	–
32	РА5	Вход/выход 5 порта А		
		Аналоговая	–	–
		Основная	TMR0_CH3N	Инверсный выход канала 3 таймера 0
		Альтернативная	–	–
38	РА6/ SWCLKTCK	Вход/выход 6 порта А		
		Аналоговая	–	–
		Основная	TMR0_CH4	Вход/выход канала 4 таймера 0
		Альтернативная	–	–
39	РА7/SWDIO	Вход/выход 7 порта А		
		Аналоговая	–	–
		Основная	TMR0_CH4N	Инверсный выход канала 4 таймера 0
		Альтернативная	–	–
40	РА8	Вход/выход 8 порта А		
		Аналоговая	–	–
		Основная	TMR0_ETR	Вход внешнего события таймера 0
		Альтернативная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
41	PA9	Вход/выход 9 порта А		
		Аналоговая	–	–
		Основная	TMR0_BRK	Вход внешнего события таймера 0
		Альтернативная	–	–
42	PA10	Вход/выход 10 порта А		
		Аналоговая	–	–
		Основная	EXT_INT0	Вход внешнего прерывания
		Альтернативная	–	–
47	PA11	Вход/выход 11 порта А		
		Аналоговая	–	–
		Основная	–	–
		Альтернативная	–	–
48	PA12	Вход/выход 12 порта А		
		Аналоговая	–	–
		Основная	SSP_FSS	Вход/выход разрешения SPI
		Альтернативная	–	–
2	PA13	Вход/выход 13 порта А		
		Аналоговая	–	–
		Основная	SSP_CLK	Вход/выход синхросигнала SPI
		Альтернативная	–	–
3	PA14	Вход/выход 14 порта А		
		Аналоговая	–	–
		Основная	SSP_RXD	Входные данные SPI
		Альтернативная	–	–
4	PA15	Вход/выход 15 порта А		
		Аналоговая	–	–
		Основная	SSP_TXD	Выходные данные SPI
		Альтернативная	–	–
Порт В				
34	PB0/ MODE0	Вход/выход 0 порта В		
		Аналоговая	–	–
		Основная	UART0_TXD	Выходные данные UART0
		Альтернативная	–	–
35	PB1	Вход/выход 1 порта В		
		Аналоговая	–	–
		Основная	UART0_RXD	Входные данные UART0
		Альтернативная		
36	PB2	Вход/выход 2 порта В		
		Аналоговая	–	–
		Основная	nSIROUT0	Выходные данные IRDA SIR UART0
		Альтернативная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
37	PB3	Вход/выход 3 порта B		
		Аналоговая	–	–
		Основная	nSIRIN0	Входные данные IRDA SIR UART0
		Альтернативная	–	–
15	PB4	Вход/выход 4 порта B		
		Аналоговая	OSC_IN32	Аналоговый вход тактовой частоты 32 кГц осциллятора LSE
		Основная	nUART0DTR	Выход готовности для передачи данных UART0
		Альтернативная	–	–
16	PB5	Вход/выход 5 порта B		
		Аналоговая	OSC_OUT32	Аналоговый выход осциллятора LSE
		Основная	nUART0RTS	Выход запроса на передачу данных UART0
		Альтернативная	–	–
24	PB6	Вход/выход 6 порта B		
		Аналоговая	ADC7	Аналоговый вход канала 7 АЦП последовательных приближений
		Основная	nUART0RI	Вход сигнала вызова от источника
		Альтернативная	EXT_INT1	Вход внешнего прерывания
25	PB7	Вход/выход 7 порта B		
		Аналоговая	ADC6	Аналоговый вход канала 6 АЦП последовательных приближений
		Основная	nUART0DCD	Вход информационного сигнала от источника
		Альтернативная	EXT_INT2	Вход внешнего прерывания
26	PB8	Вход/выход 8 порта B		
		Аналоговая	ADC5	Аналоговый вход канала 5 АЦП последовательных приближений
		Основная	nUART0DSR	Вход готовности для передачи данных источником
		Альтернативная	TMR1_ETR	Вход внешнего события таймера 1
27	PB9	Вход/выход 9 порта B		
		Аналоговая	ADC4	Аналоговый вход канала 4 АЦП последовательных приближений
		Основная	nUART0CTS	Вход готовности для приема данных источником
		Альтернативная	TMR1_BRK	Вход внешнего события таймера 1
45	PB10	Вход/выход 10 порта B		
		Аналоговая	–	–
		Основная	TMR1_CH2	Вход/выход канала 2 таймера 1
		Альтернативная	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
46	PB11	Вход/выход 11 порта В		
		Аналоговая	–	–
		Основная	TMR1_CH2N	Инверсный выход канала 2 таймера 1
		Альтернативная	–	–
5	PB12	Вход/выход 12 порта В		
		Аналоговая	–	–
		Основная	TMR1_CH3	Вход/выход канала 3 таймера 1
		Альтернативная	–	–
6	PB13	Вход/выход 13 порта В		
		Аналоговая	–	–
		Основная	TMR1_CH3N	Инверсный выход канала 3 таймера 1
		Альтернативная	–	–
7	PB14	Вход/выход 14 порта В		
		Аналоговая	–	–
		Основная	TMR1_CH4	Вход/выход канала 4 таймера 1
		Альтернативная	–	–
Порт С				
43	PC0/ MODE1	Вход/выход 0 порта С		
		Аналоговая	–	–
		Основная	UART1_TXD	Выходные данные UART1
		Альтернативная	–	–
20	PC1	Вход/выход 1 порта С		
		Аналоговая	ADC3	Аналоговый вход канала 3 АЦП последовательных приближений
		Основная	UART1_RXD	Входные данные UART1
		Альтернативная	–	–
21	PC2	Вход/выход 2 порта С		
		Аналоговая	ADC2	Аналоговый вход канала 2 АЦП последовательных приближений
		Основная	TMR1_CH1	Вход/выход канала 1 таймера 1
		Альтернативная	–	–
22	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	–	–

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
23	PC4	Вход/выход 4 порта C		
		Аналоговая	ADC0_REF-	Аналоговый вход канала 0/ Напряжение нижней границы внешней опоры АЦП последовательных приближений
		Основная	EXT_INT1	Вход внешнего прерывания
		Альтернативная	–	–
30	PC5	Вход/выход 5 порта C		
		Аналоговая	–	–
		Основная	EXT_INT2	Вход внешнего прерывания
		Альтернативная	–	–
31	PC6	Вход/выход 6 порта C		
		Аналоговая	–	–
		Основная	TMR1_ETR	Вход внешнего события таймера 1
		Альтернативная	–	–
44	PC7	Вход/выход 7 порта C		
		Аналоговая	–	–
		Основная	TMR1_BRK	Вход внешнего события таймера 1
		Альтернативная	–	–
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	I1P	Положительный вход канала 1 тока		
56	I1N	Отрицательный вход канала 1 тока		
57	V1P	Положительный вход канала 1 напряжения		
58	V1N	Отрицательный вход канала 1 напряжения		
59	I2P	Положительный вход канала 2 тока		
60	I2N	Отрицательный вход канала 2 тока		
61	V2P	Положительный вход канала 2 напряжения		
62	V2N	Отрицательный вход канала 2 напряжения		
63	I3P	Положительный вход канала 3 тока		
64	I3N	Отрицательный вход канала 3 тока		
Системное управление				
33	RESET	Сигнал внешнего сброса		
17	WAKEUP/ COV_DET	Сигнал внешнего выхода из режима Standby		
28	OSC_IN	Вход генератора HSE		
29	OSC_OUT	Выход генератора HSE		

Номер вывода	Обозначение вывода	Назначение и функции вывода		
		Тип функции вывода	Обозначение функции вывода	Функциональное назначение вывода
Питание				
13	U _{CC}	Питание (3,0...3,6) В		
1	AU _{CC}	Аналоговое питание $\Delta\Sigma$ АЦП (2,4...3,6) В		
14	BU _{CC}	Батарейное питание (1,8...3,6) В		
49	AGND	Общий АЦП, ЦАП		
19	GND	Общий		
Выводы для тестирования и исследования				
18	JTAG_EN	Тестовый вывод. Рекомендуется не подсоединять или подключать к шине общий		

5 Расположение выводов в корпусе

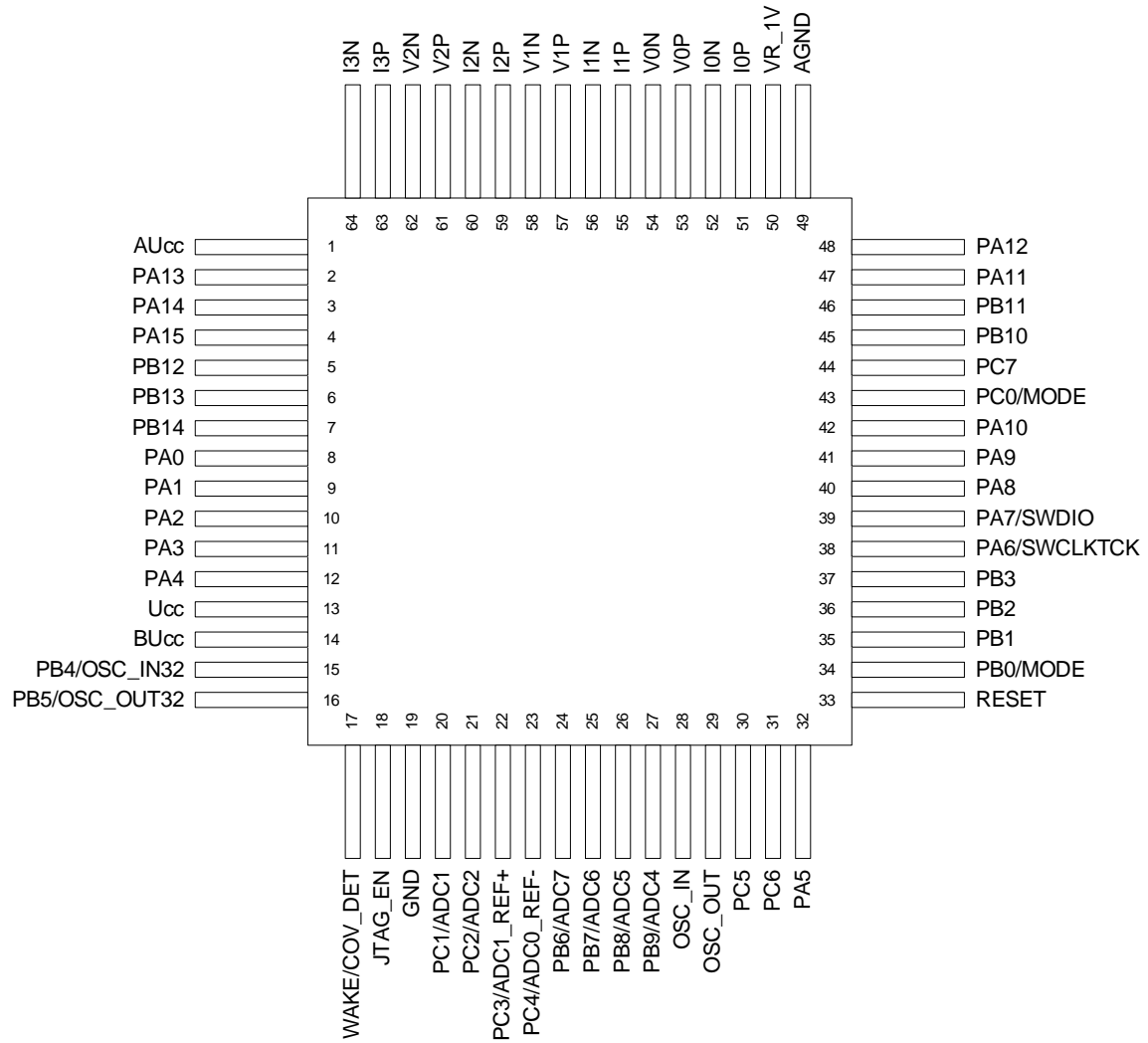


Рисунок 3 – Диаграмма расположения выводов

6 Указания по применению и эксплуатации

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.

В зависимости от рабочей частоты микроконтроллера для правильной работы регулятора LDO должны быть настроены поля SelectRI[2:0] и LOW[2:0] регистра REG_0E блока батарейного домена (при переходе на более высокую рабочую частоту настройка полей выполняется до перехода, при переходе с высокой частоты на низкую настройка полей выполняется после смены частоты).

Неиспользуемые входы микросхем следует подключать к шинам питания и общий.

Типовая схема подключения питания к микросхеме приведена на рисунке 142.

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 имеют несколько типов выводов питания:

U_{CC} – Основное питание микросхемы, включает питание пользовательских выводов, встроенного регулятора напряжения, РНУ, PLL, генераторов, компаратора и АЦП последовательного приближения.

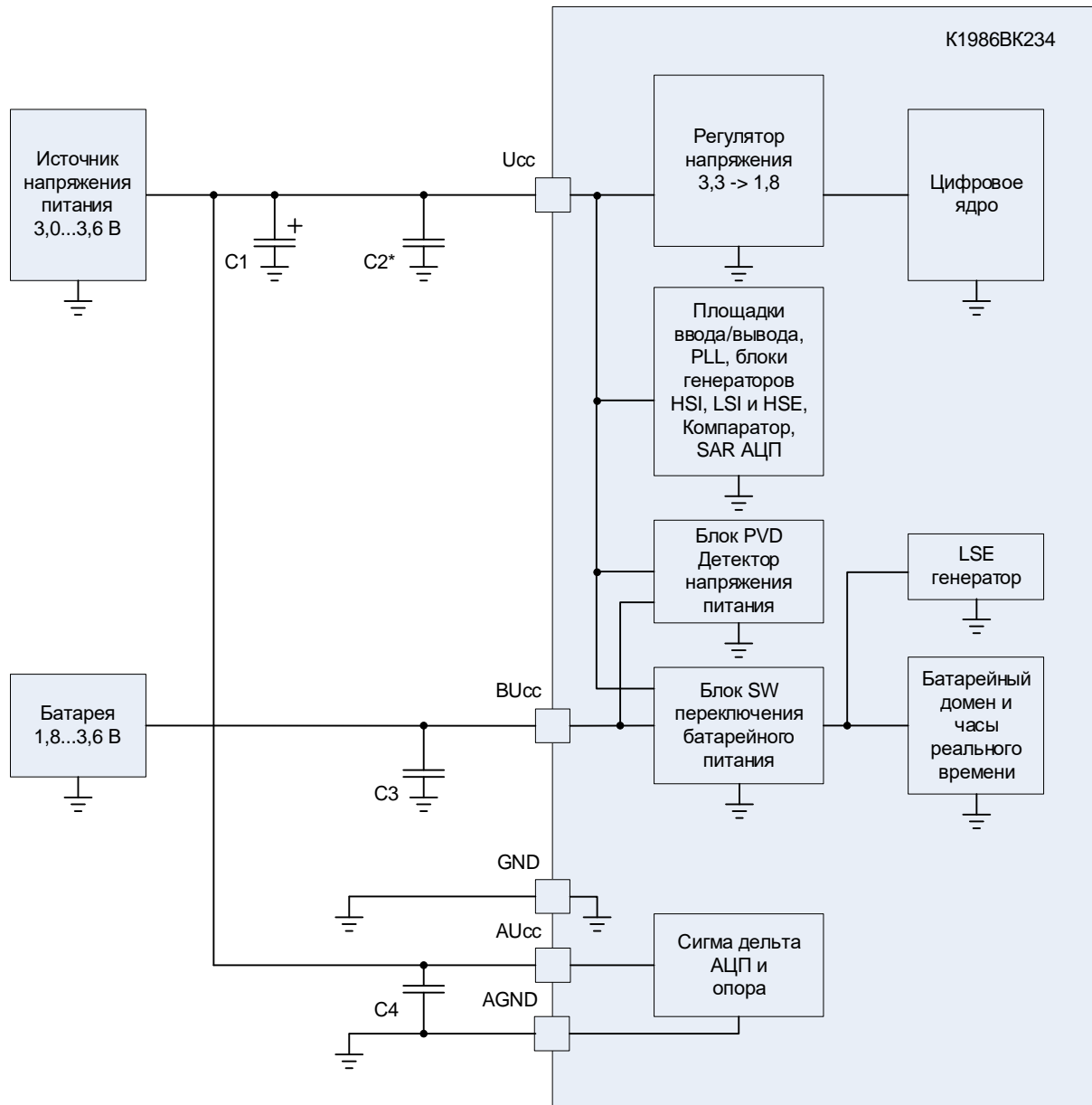
V_{UCC} – Питание батарейного домена используется при отсутствии основного питания U_{CC} для питания батарейного домена и LSE генератора. Переключение с основного питания на батарейное происходит автоматически при снижении уровня U_{CC} ниже 2,0 В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 4 мс после превышения уровнем U_{CC} значения 2,0 В. Если в системе не требуется батарейного питания, вывод V_{UCC} должен быть объединен с U_{CC}.

AU_{CC} – 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_{CC}, но при этом на печатной плате должны быть применены меры по снижению наводки помех.

GND – Основная земля питания.

AGND – Земля аналогового питания AU_{CC}. 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 $C1 = 22 \text{ мкФ}$, $C2 = C3 = C4 = 0,1 \text{ мкФ}$;
- 2 Если не используется батарейное питание, то вывод BU_{CC} должен быть объединен с U_{CC} .

Микросхема имеет встроенный детектор напряжения питания, подробнее см. раздел 18 «Детектор напряжения питания».

Рисунок 4 – Структурная схема подачи питания

7.2 Схема сброса при включении и выключении основного питания

При включении питания, пока питание U_{CC} не превысило уровень U_{POR} (2,0 В) вырабатывается внутренний сигнал сброса POR для цифровой части. После превышения уровня U_{POR} , сигнал POR выдается еще на протяжении t_{POR} (~4 мс), для того чтобы гарантировано установилось напряжение питания, после чего сигнал POR снимается, и схема может начать работать.

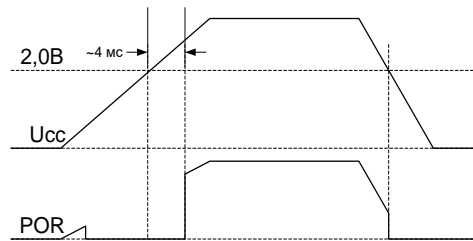


Рисунок 5 – Диаграмма формирования сброса микросхемы

При снижении напряжения питания U_{CC} ниже уровня U_{Por} сигнал POR вырабатывается без задержки.

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

При включении основного напряжения питания U_{CC} автоматически включается встроенный регулятор напряжения для формирования напряжения питания цифрового ядра. В ходе работы микроконтроллера встроенный регулятор может быть отключен.

Микросхема также может быть сброшена внешним сигналом сброса RESET или внутренними сигналами сброса сторожевых таймеров или программным сбросом. При этом сигнал сброса формируется специальной схемой сброса, содержащий фильтр «иглолок» по сигналу сброса и одновибратор для увеличения длительности сигнала сброса.

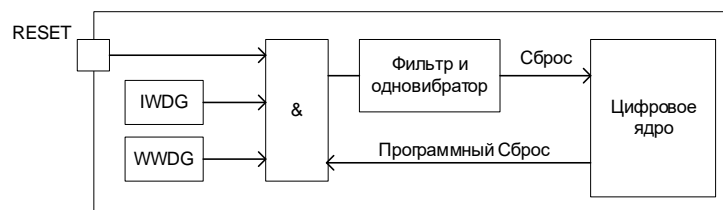


Рисунок 6 – Блок-схема формирования сброса микросхемы

При приходе импульсов сброса длительностью менее $t_{minreset}$, эти импульсы отфильтровываются и не приводят к сбросу процессора. Если длительность импульса больше $t_{maxreset}$ вырабатывается сигнал сброса. При этом длительность сформированного сигнала сброса будет не менее t_{reset} .

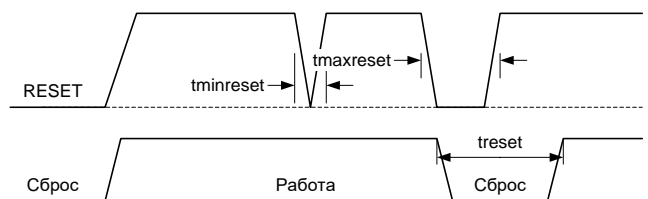


Рисунок 7 – Диаграмма фильтрации помех при формировании сброса

8 Организация памяти

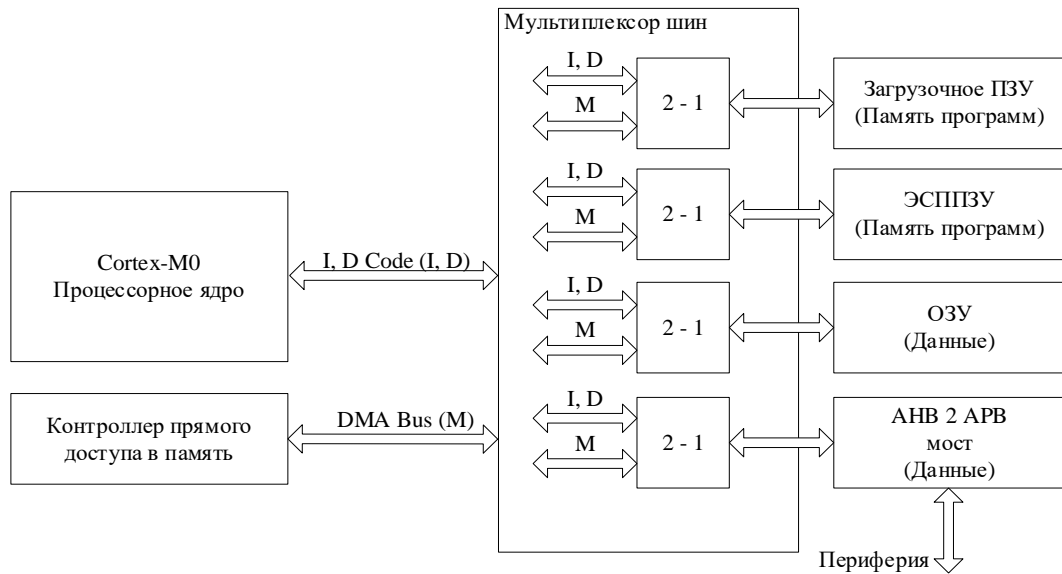


Рисунок 8 – Структурная схема организации памяти

Процессорное ядро имеет три системные шины:

I, D Code – шина выборки инструкций и данных.

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

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

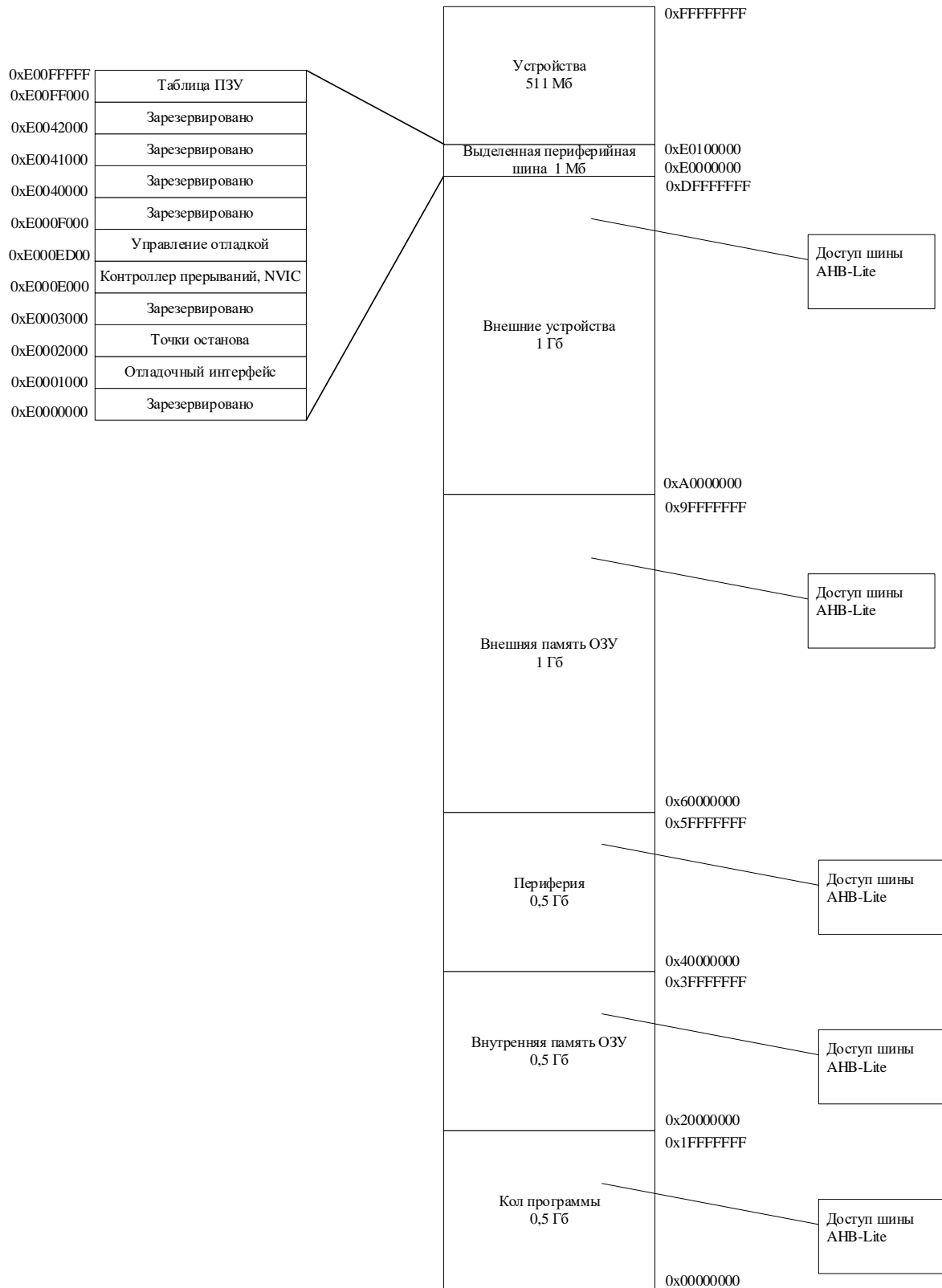


Рисунок 9 – Структура адресного пространства микросхемы

8.1 Базовые адреса процессора

Таблица 3 – Базовые адреса 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		SSP	Регистры контроллера интерфейса SSP
0x4000_8000		UART0	Регистры контроллера интерфейса UART0
0x4001_0000		UART1	Регистры контроллера интерфейса UART1
0x4001_8000		EEPROM_CNTRL	Регистры контроллера Flash-памяти программ
0x4002_0000		RST_CLK	Регистры контроллера сигналов тактовой частоты
0x4002_8000		DMA	Регистры контроллера прямого доступа в память
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 Таймер 0
0x4007_8000		TIMER1	Регистры управления Таймер 1
0x4008_0000		PORTA	Регистры управления порта А
0x4008_8000		PORTB	Регистры управления порта В
0x4009_0000		PORTC	Регистры управления порта С
0x4009_8000		CRC	Регистры управления аппаратного блока вычисления CRC
SYSTEM REGION			
0xE000_0000			Системные регистры процессор ARM Cortex-M0

9 Загрузочное ПЗУ и режимы работы микросхемы

После включения питания и снятия внутренних (POR) и внешних (RESET) сигналов сброса, микросхема начинает выполнять программу из информационной загрузочной области FLASH BOOT ROM. В загрузочной программе микросхема определяет, в каком из режимов он 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, который может быть сброшен только при отключении основного питания U_{CC}. После перезапуска микросхемы уровни на выводах MODE[1:0] не влияют на режим функционирования микросхемы, если установлен бит FPOR.

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

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 UART0 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 UART0 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 Wire адаптер с помощью которого программные средства разработки позволяют работать с микросхемой в отладочном режиме.

В отладочном режиме можно:

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

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

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 (EEPROM, ROM, 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 данных 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 Stop 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 Rx используется как вход. Master постоянно посылает в канал синхросимвол – 0. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ошибки обмена. Как только Загрузчик настроил скорость, он переходит в диспетчер команд и выдает 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
Формат команды:	
Master: Выдает код команды CMD_SYNC	Slave: Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды

9.1.5 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
Формат команды:	
Master: Выдает код команды CMD_CR	Slave: Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. 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]
Формат команды:	
Master: Выдает код команды CMD_BAUD	Slave: Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр	Slave: Если параметр принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. Иначе: – выдает код команды CMD_BAUD; – устанавливает новое значение скорости обмена во время окончания 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
Формат команды:	
Master: Выдает код команды CMD_LOAD	Slave: Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр 1	Slave: Ожидает получения всех параметров.
Master: Выдает параметр 2	Slave: Если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды. 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 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
Формат команды:	
Master: Выдает код команды CMD_VFY	Slave: Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр 1	Slave: Ожидает получения всех параметров
Master: Выдает параметр 2	Slave: Если хотя бы один из параметров принят с ошибками, то выдает сообщение об ошибке 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)
Формат команды:	
Master: Выдает код команды CMD_RUN	Slave: Если команда принята с ошибками, то выдает сообщение об ошибке 0x45 ('E') и завершает обработку текущей команды
Master: Выдает параметр	Slave: Если параметр принят с ошибками, то выдает сообщение об ошибке 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 должен прекратить передачу байт, относящихся к текущей команде.

После принятия сообщения об ошибке Master должен подавать команду CMD_CR до тех пор, пока не получит корректный ответ, соответствующий этой команде.

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 128 Кбайт основной памяти программ и 8 Кбайт информационной памяти. В обычном режиме (бит CON = 0, регистр EEPROM_CMD) 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. Для того, чтобы процессорное ядро могло получать новые инструкции на больших частотах в микросхеме реализуется Flash-память с физической организацией 128К 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, так 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 для работы Flash-программ

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), так и с информационной (бит IFREN = 1) памятью:

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

Структура памяти представлена на рисунке 10.

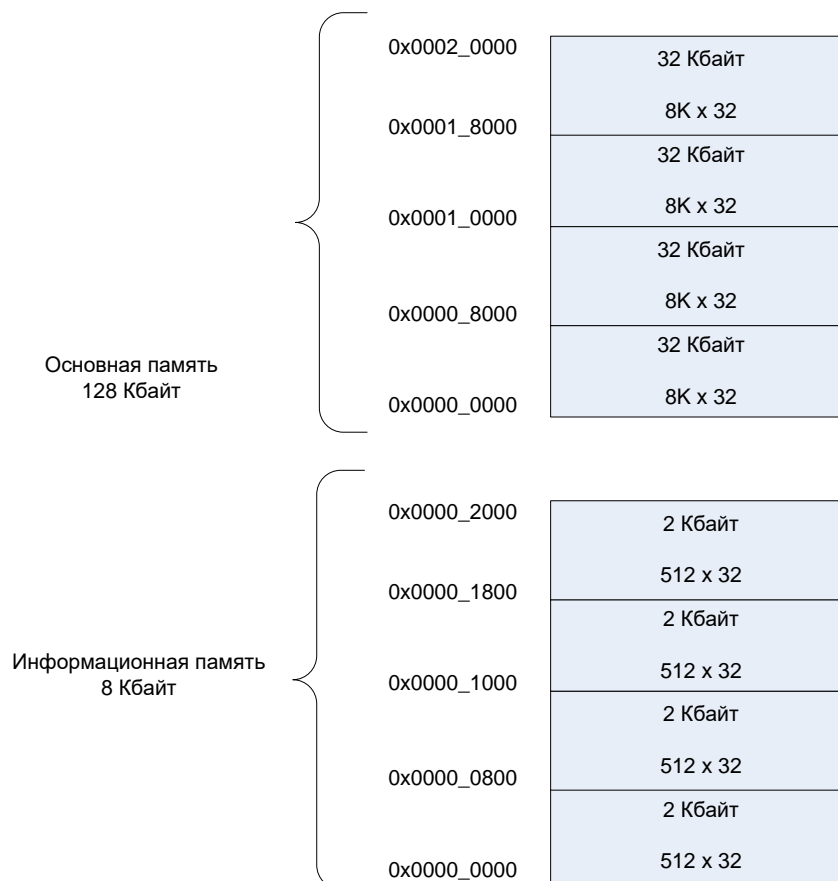


Рисунок 10 – Структура памяти

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 (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 битах ADR[16:15] регистра EEPROM_ADR, затем установить биты XE, MAS1 и ERASE в «1», и спустя время t_{nvs} (не менее 10 мкс) установить бит NVSTR в «1». Полное стирание блока памяти длится время t_{me} (от 20 мс до 40 мс). Спустя это время необходимо очистить бит ERASE, и спустя время t_{nvhl} (не менее 100 мкс) очистить биты XE, MAS1 и NVSTR. Последующие операции с памятью можно выполнять спустя время t_{rcv} (не менее 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 11.

Mass Erase Cycle

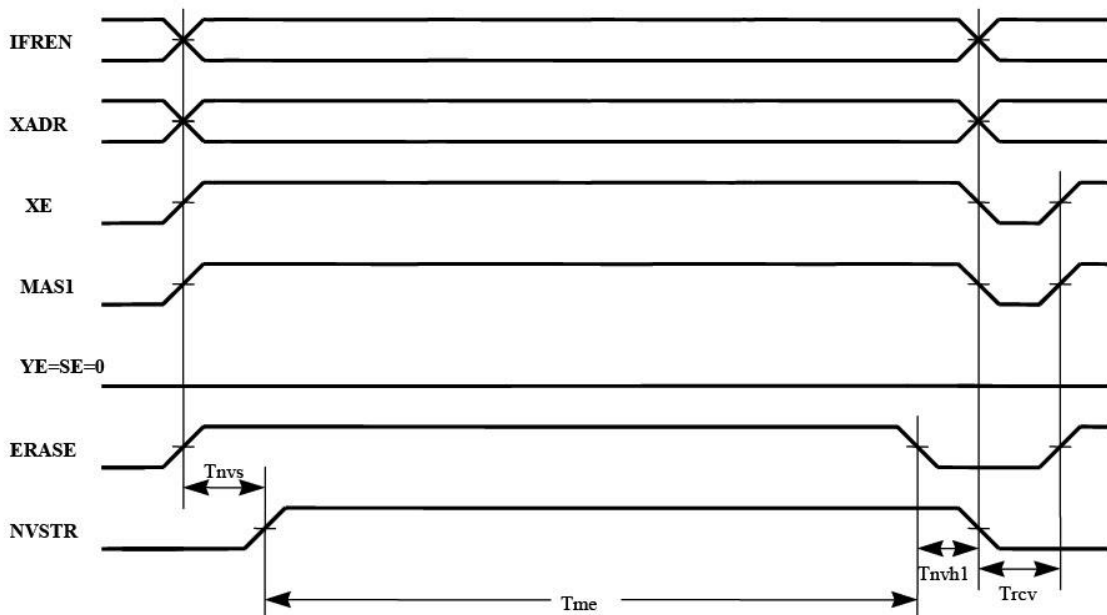


Рисунок 11 – Временная диаграмма стирания памяти

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_{nvhl} (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 12.

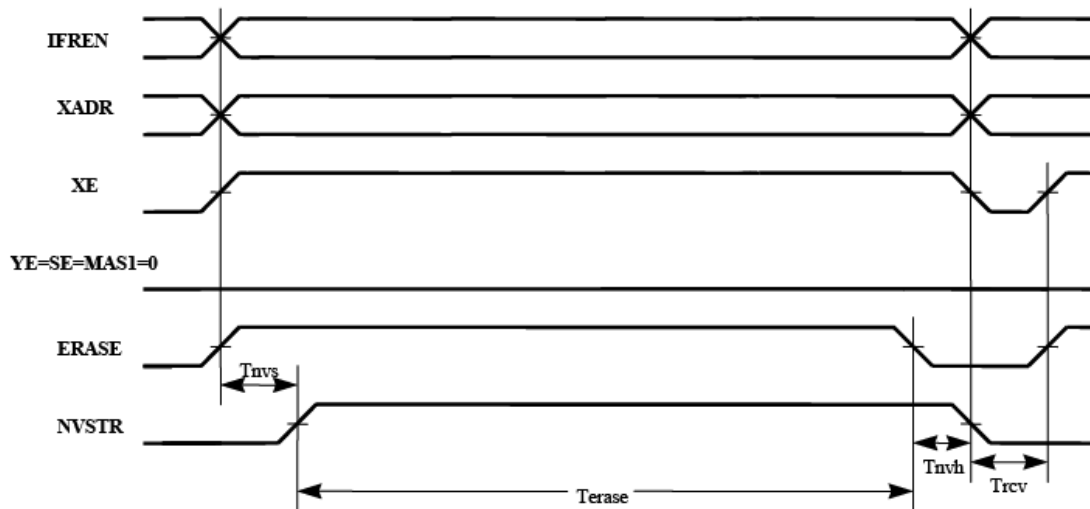


Рисунок 12 – Временная диаграмма стирания страницы памяти

10.2.3 Запись 32-битного слова в память.

Запись в память возможна только в режиме программирования.

Примечание – Перед программированием ячейки необходимо выполнить ее стирание.

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 следующую 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_{rev} (не менее 1 мкс). Временная диаграмма записи памяти представлена на рисунке 13.

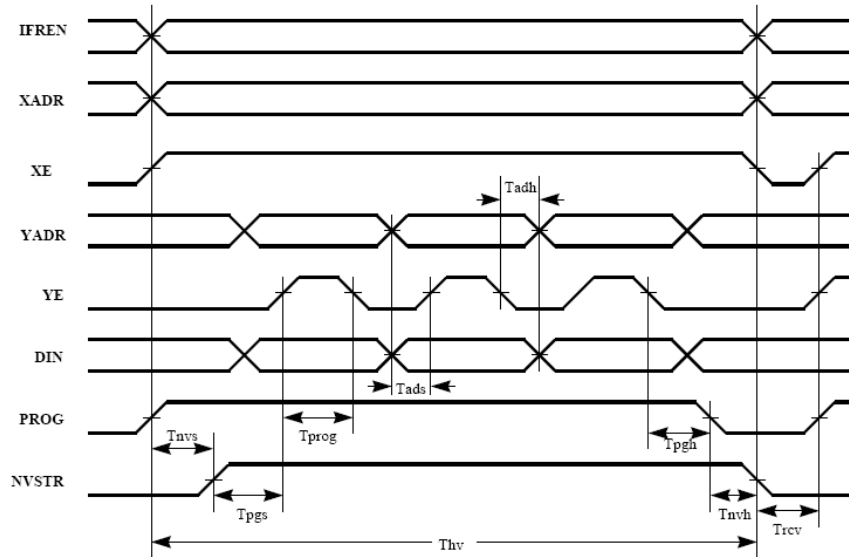


Рисунок 13 – Временная диаграмма записи памяти

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_{xa} (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, затем очистить бит SE и далее повторно установить бит SE в "1". Спустя время t_{xa} (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 14.

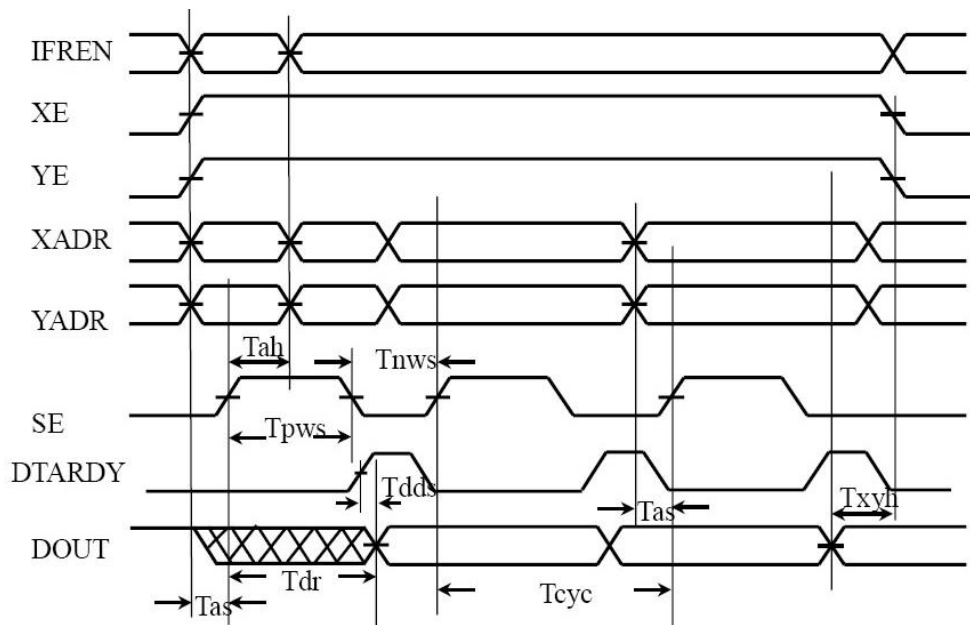


Рисунок 14 – Временная диаграмма чтения памяти

Flash-память программ поддерживает не менее 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 считанных данных
0x10	EEPROM_KEY	Регистр ключа

10.3.1 EEPROM_CMD

Таблица 14 – Регистр EEPROM_CMD

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

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

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

Таблица 15 – Описание бит регистра EEPROM_CMD

Разряды	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;

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

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

10.3.3 EEPROM_DI

Таблица 18 – Регистр EEPROM_DI

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

Таблица 19 – Описание бит регистра EEPROM_DI

Разряды	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 20 – Регистр EEPROM_DO

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

Таблица 21 – Описание бит регистра EEPROM_DO

Разряды	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 22 – Регистр EEPROM_KEY

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

Таблица 23 – Описание бит регистра EEPROM_KEY

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	KEY[31:0]	Ключ для разрешения доступа к Flash-памяти через регистровый доступ. Перед переводом памяти в режим программирования необходимо в регистр 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 ANB-Lite.

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

Полный набор отладочных модулей:

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

11.1 Структурная схема процессора

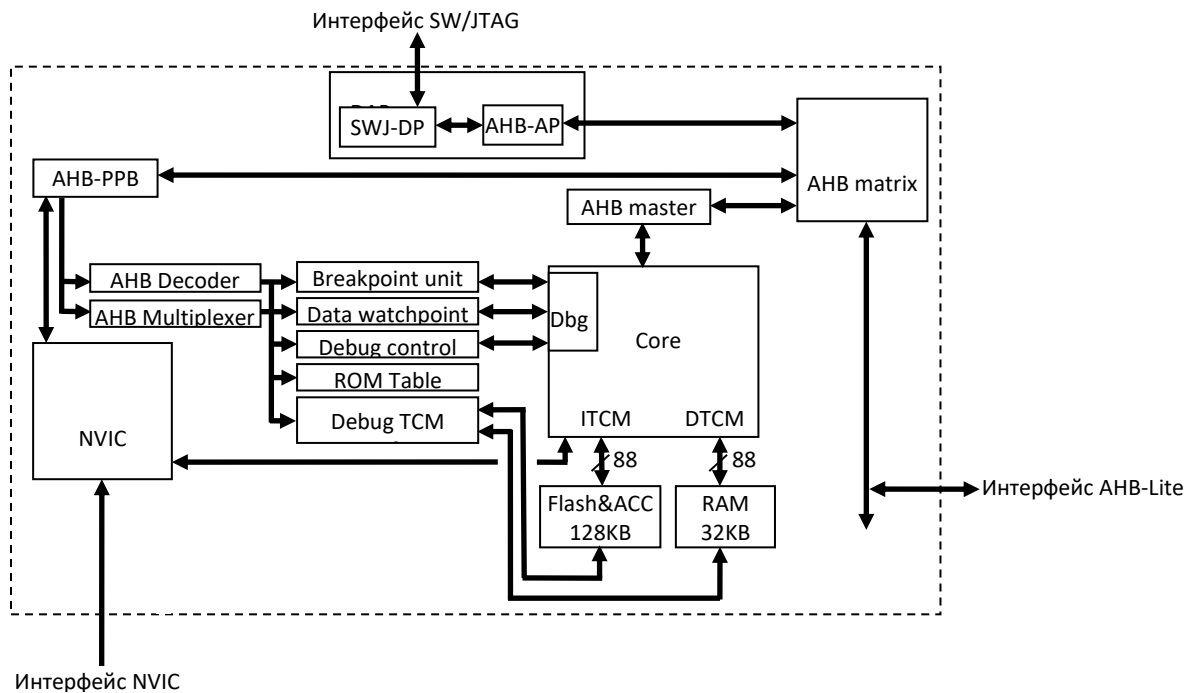


Рисунок 15 – Структурная схема процессора

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 окончании 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.

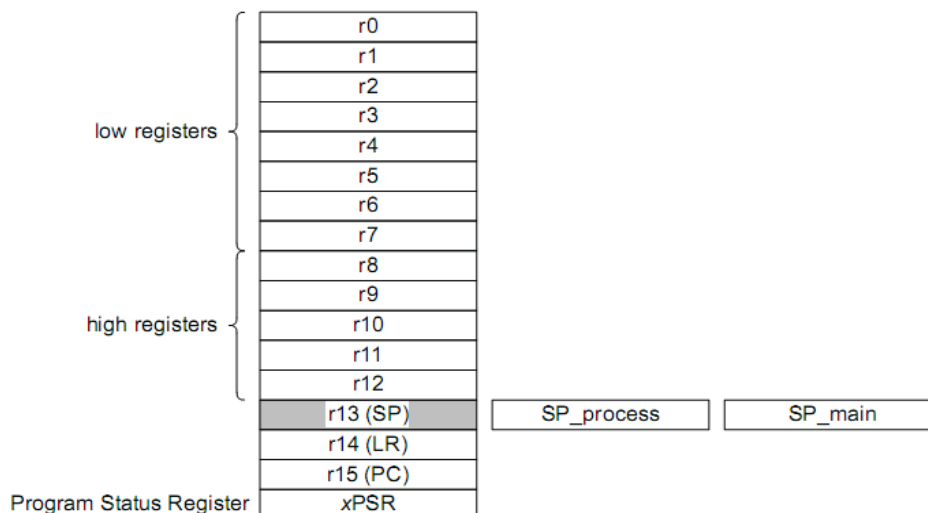


Рисунок 16 – Регистры ядра

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 (четырёх байт). Биты SP[1:0] могут быть очищены инструкцией SBZP. В режиме Handler всегда используется SP_main, а в режиме Thread может быть использован либо SP_main, либо SP_process.

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 24. 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 24 – Комбинация PSR и их атрибуты

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

1 Игнорируется запись в IPSR биты.
 2 При чтении EPSR битов читаются нули, и запись в них игнорируется.
 Подробнее в описании инструкции «MRS» и «MSR»

11.4.6 APSR

Регистр APSR содержит текущие флаги состояния выполнения предыдущей инструкции.

Таблица 25 – Регистр APSR

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

Таблица 26 – Описание бит регистра APSR

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

11.4.7 IPSR

Регистр IPSR содержит номер типа исключения для текущего обработчика прерывания.

Таблица 27 – Регистр IPSR

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

Таблица 28 – 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 29 – Регистр EPSR

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

Таблица 30 – 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 31 – Регистр PRIMASK

Номер	31...1	0
Доступ		
Сброс		
	-	PRIMASK

Таблица 32 – 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 33 – Регистр CONTROL

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

Таблица 34 – 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 35.

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 35 – 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	-

Мнемокод команды	Операнды	Краткое описание	Флаги
LDRB	Rt, [Rn, <Rm #imm>]	Загрузка байта в регистр	-
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	-
LSLS	{Rd,} Rn, <Rs #imm>	Логический сдвиг влево	N, Z, C
LSRS	{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
MULS	Rd, Rn, Rm	Умножение, 32-разрядный результат	N, Z
MVNS	Rd, Rm	Загрузка инверсного значения	N, Z
NOP	-	Нет операции	-
ORRS	{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	-
RORS	{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 36 – 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 37 – 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.

12.2.3 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).

12.2.3.1 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 (рисунок 17).

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.

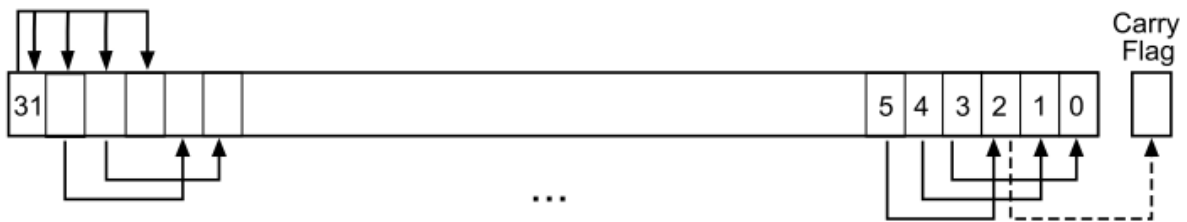


Рисунок 17 – Инструкция ASR # 3

12.2.3.2 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 18).

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.

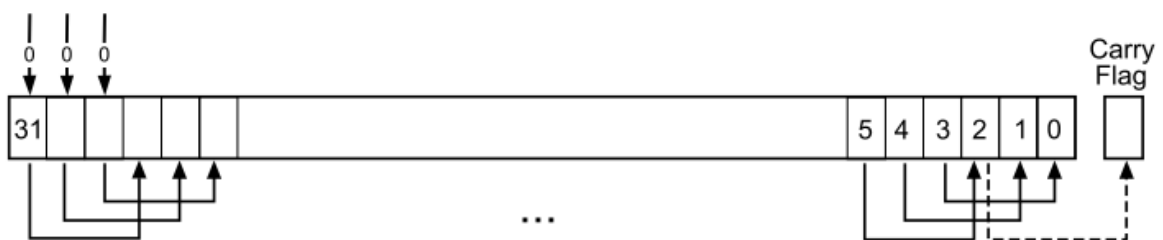


Рисунок 18 – Инструкция LSR # 3

12.2.3.3 LSL

Логический сдвиг влево на n бит переносит крайние справа 32-n бит регистра Rm влево на n позиций, то есть на место крайних слева 32-n. При этом в n крайних слева бит результата записывается 0 (см. рисунок 19).

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

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

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

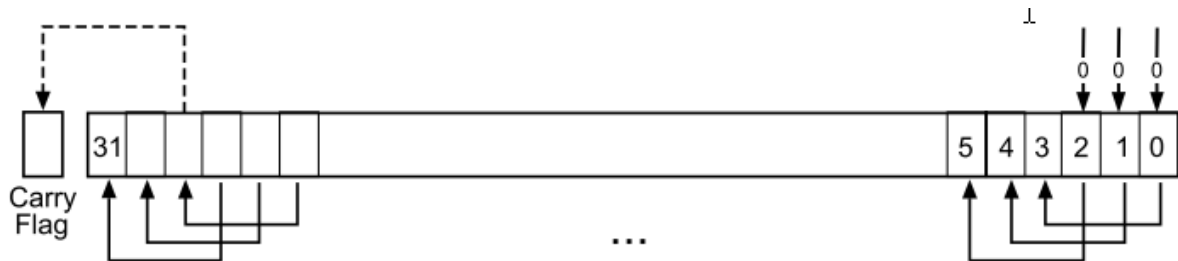


Рисунок 19 – Инструкция LSL # 3

12.2.3.4 ROR

Циклический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. При этом n крайних справа разрядов регистра переносятся в крайние n слева разрядов результата (см. рисунок 20).

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

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

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

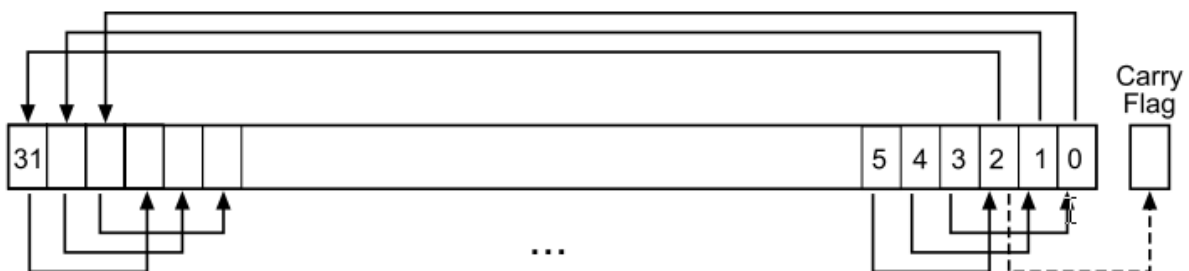


Рисунок 20 – Инструкция ROR # 3

12.2.4 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 HardFault, что указано выставленным битом UNALIGN_TRP регистра конфигурации и управления CCR (см. пункт 27.1.5 «Регистр конфигурации и управления»).

12.2.5 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 4 байта.

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

12.2.6 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. 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.

12.2.6.1 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.

12.2.6.2 Суффиксы условного исполнения

Условный переход в описании синтаксиса это обозначается как $V\{cond\}$. 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 38 приведены доступные коды условий и соответствующие им флаги условий N, Z, C, V.

Таблица 38 – 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	$N \neq V$	Меньше, знаковое сравнение
GT	$Z = 0$ and $N = V$	Больше, знаковое сравнение
LE	$Z = 1$ and $N \neq V$	Меньше или равно, знаковое сравнение
AL	1	Безусловное исполнение.

12.3 Команды доступа к памяти

Обобщенные данные о командах доступа к памяти приведены в таблице 39.

Таблица 39 – 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

12.3.1 ADR

Загрузка адреса, заданного относительно счетчика команд.

Синтаксис

ADR Rd, label

где Rd – регистр-получатель;

label – относительный адрес, см. пункт 12.2.5 «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.

12.3.2 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 12.2.4 «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.

12.3.3 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 12.2.4 «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.

12.3.4 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 12.2.5 «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.

12.3.5 LDM и STM

Загрузка или сохранение множества регистров.

Синтаксис

LDM Rn{!}, reglist

STM Rn!, reglist

где Rn – 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 Rn;

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

Мнемокоды LDMIA и LDMFD – это псевдокоманды LDM. Командой LDMIA обозначают загрузку множества регистров с увеличением значения адреса в Rn 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 Rn 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 Rn.

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

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

Ограничения

В описываемых в разделе командах:

- в списке reglist и в качестве Rn можно использовать только регистры R0-R7;
- суффикс обратной записи должен использоваться всегда. Исключение составляет команда LDM, в которой в списке reglist содержится регистр Rn. В этом случае суффикс обратной записи использовать нельзя;
- значение адреса в регистре Rn должно быть выровнено на границу слова, см. пункт 12.2.4 «Выравнивание адресов»;
- если в команде STM регистр Rn указан в списке reglist, то в таком случае Rn 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.

12.3.6 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.

12.4 Инструкции обработки данных

Таблица 40 показывает инструкции обработки данных.

Таблица 40 – 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
CMR	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
SXTH	Заполнение знаком полуслова
UXTB	Заполнение нулем байта
UXTH	Заполнение нулем полуслова
TST	Проверка значения битов по маске

12.4.1 ADD, ADC, SUB, SBC и RSBS

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

Примечание – Процессорное ядро поддерживает инструкции ADC, SBC и RSBS 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 12.2.6 «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 SBСS 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 41 представлены допустимые комбинации регистров и значений констант imm, которые можно использовать применительно к указанным инструкциям.

Таблица 41 – 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

Инструкция	Rd	Rn	Rm	imm	Ограничения
ADDS	R0-R7	R0-R7	-	0-7	-
	R0-R7	R0-R7	-	0-255	<i>Rd</i> и <i>Rn</i> 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	-	<i>Rd</i> и <i>Rn</i> должны задавать один и тот же регистр
SUB	SP	SP	-	0-508	Значение <i>imm</i> должно быть целым числом кратным 4
SUBS	R0-R7	R0-R7	-	0-7	-
	R0-R7	R0-R7	-	0-255	<i>Rd</i> и <i>Rn</i> 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.

12.4.2 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

12.4.3 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 12.2.3 «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 12.2.3 «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.

12.4.4 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

12.4.5 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 12.2.6 «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. T-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.

12.4.6 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 12.2.6 «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, $R0 = R0 \times R2$.

12.4.7 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.

12.4.8 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 данной инструкции 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.

12.4.9 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.

12.5 Инструкции передачи управления

Таблица 42 показывает список инструкций передачи управления.

Таблица 42– Инструкции передачи управления

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

12.5.1 B, BL, BX и BLX

Команды ветвления.

Синтаксис

B {cond} label

BL label

BX Rm

BLX Rm

где cond – необязательный код условия, см. пункт 12.2.6 «Условное исполнение»;
label – относительный адрес, см. пункт 12.2.5 «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 BL и BLX записывают адрес следующей инструкции в регистр связи LR (R14);
- команды BX и BLX формируют отказ (Hard fault) в случае, если бит [0] регистра Rm равен «0».

Инструкции BL и BLX также устанавливают бит [0] регистра LR в «1». Это гарантирует, что при использовании данного значения инструкциями POP {PC} или BX будет выполнен успешный переход.

Таблица 43 показывает диапазон адресуемых переходов для различных команд ветвления.

Таблица 43 – Диапазон адресуемых переходов для команд ветвления

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

Ограничения

- В командах BX и BLX не допускается использование регистра PC и SP;
- В командах BX 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»;
- V {cond} – единственная условно исполняемая команда.

Флаги

Данные инструкции не влияют на состояние флагов.

Примеры

- V loopA – передача управления по адресу, обозначенному меткой loopA;
- BL funC – переход со связью (вызов функции) в funC, адрес возврата будет записан в регистр LR;
- BX 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.

12.6 Прочие инструкции

Таблица 44 показывает список не рассмотренных в предыдущих разделах инструкций процессора:

Таблица 44 – 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

12.6.1 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.

12.6.2 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).

12.6.3 DMB

Барьер доступа к памяти данных.

Синтаксис

DMB

Описание

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

Команда DMB 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

DMB – Барьер доступа к памяти данных.

12.6.4 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.

12.6.5 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

12.6.6 MRS

Считать содержимое специального регистра в регистр общего назначения.

Синтаксис

MRS Rd, спец_reg

где Rd – регистр-получатель результата;

спец_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-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.

12.6.7 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.

12.6.8 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.

12.6.9 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.

12.6.10 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).

12.6.11 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.

12.6.12 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 семи каналов 24-битных $\Delta\Sigma$ АЦП. Все каналы разбиты на три пары F0-F2 (канал напряжения и канал тока) для трехфазной сети и еще одного независимого канала тока. Каждый из семи каналов оцифровывает входной сигнал с выходной частотой отсчетов до 4 кГц. Кроме этого в каждой паре каналов F0-F2 реализована возможность рассчитывать среднеквадратические значения тока/напряжения, вычислять активную и реактивную мощности, вычислять потребленную активную и реактивную энергию, частоту сигнала в каналах напряжения, превышение пикового значения, падение сигнала ниже установленного уровня. Эти дополнительные блоки позволяют снизить нагрузку на процессор, что в свою очередь снижает потребляемую мощность всего кристалла. Также каждый АЦП имеет независимый канал DMA, обеспечивая возможность сохранения данных в ОЗУ без участия процессора.

Структурная схема семи каналов АЦП приведена на рисунке 21.

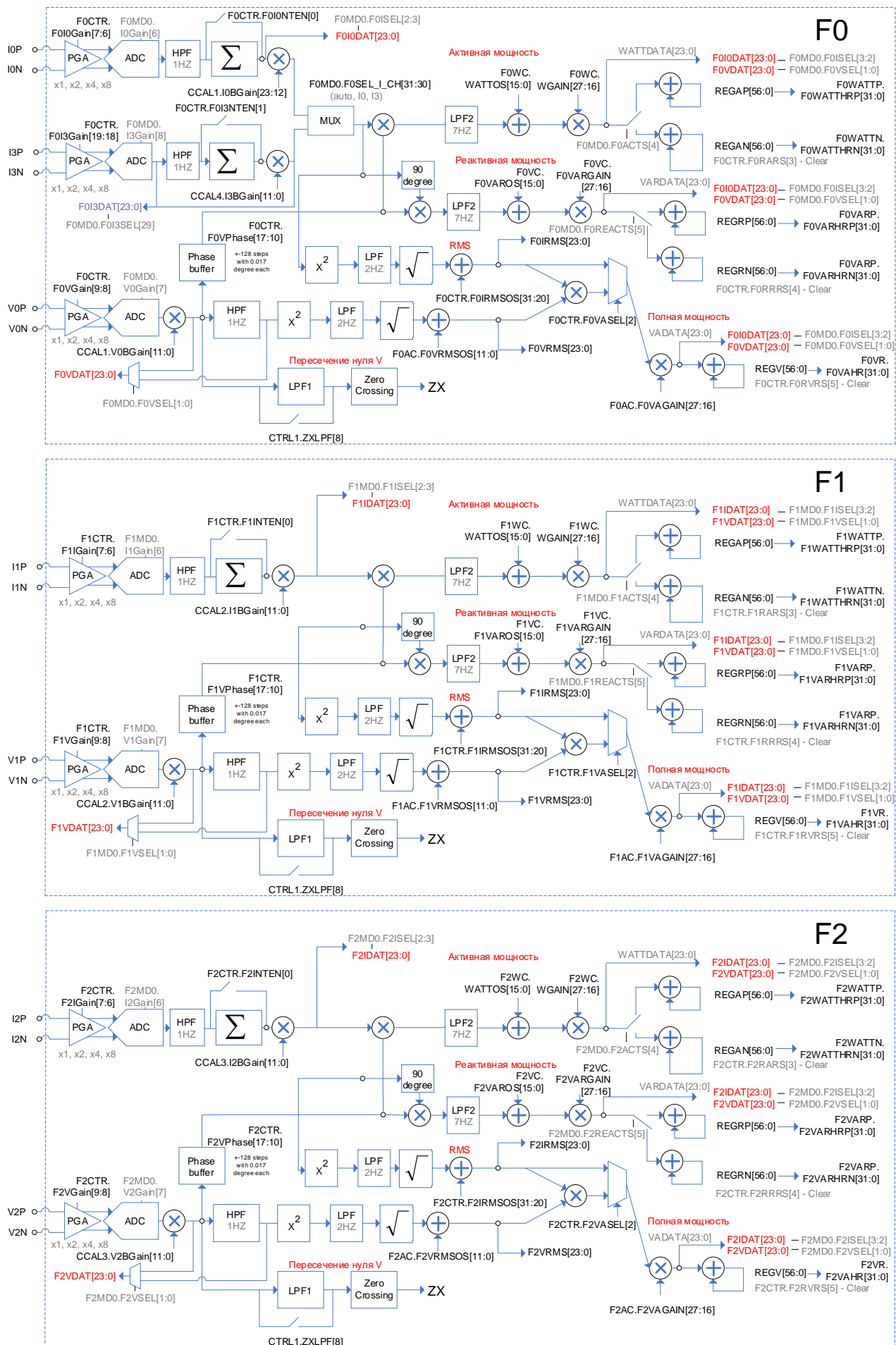


Рисунок 21 – Структурная схема семи каналов АЦП

Список вычисляемых параметров блоком АЦП:

- Семь независимых АЦП с выходной частотой отсчетов 4/8/16 кГц (четыре канала тока и три канала напряжения). Эти каналы образуют три блока для измерения параметров каждой фазы F0-F2. Размер FIFO данных по каждому каналу АЦП составляет восемь отсчетов;

- В блоке каналов F0 реализуем автоматический выбор канала тока (который имеет максимальное значение) для последующих расчетов мощностных характеристик. Если разница токов превышает 6 %, то формируется прерывание. Кроме этой функции в остальном блоке F0-F2 идентичны;

- Все каналы АЦП имеют независимые калибровочные коэффициенты наклона характеристики;

- Каждый канал тока имеет независимый интегратор;

- В каждом блоке АЦП (F0-F2) независимо рассчитывается период сигнала по каналу напряжения. Количество периодов, в течение которого рассчитывается эта величина, можно задавать равным 1/2/4/8/16/32/64/128 периодам;

- В каждом блоке есть проверка на пропажу периодического сигнала в канале напряжения;

- В каждом блоке проверяется просадка напряжения ниже заданного уровня, а также превышения сигнала в каналах тока и напряжения установленного лимита;

- Есть возможность скорректировать фазы сигналов в каналах напряжения с точностью до 0,02 %;

- Вычисляются среднеквадратические, квадрат среднеквадратических значений токов и напряжений, а также их независимая калибровка;

- При вычислении активной и реактивной энергиях значение накопленной энергии в течение периода накапливаются в отдельных регистрах (для положительной и отрицательной энергии);

- Вычисляется полная мощность и полная энергия;

- Вычисляется сдвиг фаз по отношению к фазе 0.

Для предотвращения влияния высокочастотных помех на результаты вычисления необходимо поставить внешний anti-aliasing 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 семиканального ΔΣАЦП

Таблица 45 – Описание регистров управления блока семиканального ΔΣАЦП

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	

Базовый адрес	Название	Описание
0x4006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x00C	ADCUI_F0CTR	Управление блоком каналов F0
0x010	ADCUI_F0WC	Управление расчетом активной мощности в блоке F0
0x014	ADCUI_F0WATTP	Старшая часть значения положительной активной мощности в блоке F0
0x018	ADCUI_F0WATTN	Старшая часть значения отрицательной активной мощности в блоке F0
0x01C	ADCUI_F0VC	Управление расчетом реактивной мощности в блоке F0
0x020	ADCUI_F0VARP	Старшая часть значения положительной реактивной мощности в блоке F0
0x024	ADCUI_F0VARN	Старшая часть значения отрицательной реактивной мощности в блоке F0
0x028	ADCUI_F0AC	Управление расчетом полной мощности в блоке F0
0x02C	ADCUI_F0VR	Старшая часть значения полной мощности в блоке F0
0x030	ADCUI_F0MD0	Группа параметров 0 блока F0
0x034	ADCUI_F0MD1	Группа параметров 1 блока F0
0x038	ADCUI_F0VPEAK	Пиковое значение в канале напряжения блока F0
0x03C	ADCUI_F0IPEAK	Пиковое значение в канале тока блока F0
0x040	ADCUI_F0VDAT	Буфер FIFO отсчетов в канале напряжения блока F0
0x044	ADCUI_F0I0DAT	Буфер FIFO отчетов в канале тока (I0) блока F0
0x048	ADCUI_F0I3DAT	Буфер FIFO отчетов в канале тока (I3) блока F0
0x04C	ADCUI_F0VRMS	Среднеквадратическое значение канала напряжения в блоке F0
0x050	ADCUI_F0VRMS2	Квадрат RMS в канале напряжения блока F0
0x054	ADCUI_F0IRMS	Среднеквадратическое значение канала тока в блоке F0
0x058	ADCUI_F0IRMS2	Квадрат RMS в канале тока блока F0
0x05C	ADCUI_F0STAT	Статус каналов блока F0
0x060	ADCUI_F0MASK	Маска прерываний каналов в блоке F0
0x064	ADCUI_F1CTR	Управление блоком каналов F1
0x068	ADCUI_F1WC	Управление расчетом активной мощности в блоке F1
0x06C	ADCUI_F1WATTP	Старшая часть значения положительной активной мощности в блоке F1
0x070	ADCUI_F1WATTN	Старшая часть значения отрицательной активной мощности в блоке F1
0x074	ADCUI_F1VC	Управление расчетом реактивной мощности в блоке F1
0x078	ADCUI_F1VARP	Старшая часть значения положительной реактивной мощности в блоке F1
0x07C	ADCUI_F1VARN	Старшая часть значения отрицательной реактивной мощности в блоке F1
0x080	ADCUI_F1AC	Управление расчетом полной мощности в блоке F1
0x084	ADCUI_F1VR	Старшая часть значения полной мощности в блоке F1
0x088	ADCUI_F1MD0	Группа параметров 0 блока F1
0x08C	ADCUI_F1MD1	Группа параметров 1 блока F1
0x090	ADCUI_F1MD2	Группа параметров 2 блока F1
0x094	ADCUI_F1VPEAK	Пиковое значение в канале напряжения блока F1
0x098	ADCUI_F1IPEAK	Пиковое значение в канале тока блока F1

Базовый адрес	Название	Описание
0x4006_8000	ADCUI	Контроллер АЦП напряжения/тока
Смещение		
0x09C	ADCUI_F1VDAT	Буфер FIFO отсчетов в канале напряжения блока F1
0x0A0	ADCUI_F1IDAT	Буфер FIFO отчетов в канале тока блока F1
0x0A4	ADCUI_F1VRMS	Среднеквадратическое значение канала напряжения в блоке F1
0x0A8	ADCUI_F1VRMS2	Квадрат RMS в канале напряжения блока F1
0x0AC	ADCUI_F1IRMS	Среднеквадратическое значение канала тока в блоке F1
0x0B0	ADCUI_F1IRMS2	Квадрат RMS в канале тока блока F1
0x0B4	ADCUI_F1STAT	Статус каналов блока F1
0x0B8	ADCUI_F1MASK	Маска прерываний каналов в блоке F1
0x0BC	ADCUI_F2CTR	Управление блоком каналов F2
0x0C0	ADCUI_F2WC	Управление расчетом активной мощности в блоке F2
0x0C4	ADCUI_F2WATTP	Старшая часть значения положительной активной мощности в блоке F2
0x0C8	ADCUI_F2WATTN	Старшая часть значения отрицательной активной мощности в блоке F2
0x0CC	ADCUI_F2VC	Управление расчетом реактивной мощности в блоке F2
0x0D0	ADCUI_F2VARP	Старшая часть значения положительной реактивной мощности в блоке F2
0x0D4	ADCUI_F2VARN	Старшая часть значения отрицательной реактивной мощности в блоке F2
0x0D8	ADCUI_F2AC	Управление расчетом полной мощности в блоке F2
0x0DC	ADCUI_F2VR	Старшая часть значения полной мощности в блоке F2
0x0E0	ADCUI_F2MD0	Группа параметров 0 блока F2
0x0E4	ADCUI_F2MD1	Группа параметров 1 блока F2
0x0E8	ADCUI_F2MD2	Группа параметров 2 блока F2
0x0EC	ADCUI_F2VPEAK	Пиковое значение в канале напряжения блока F2
0x0F0	ADCUI_F2IPEAK	Пиковое значение в канале тока блока F2
0x0F4	ADCUI_F2VDAT	Буфер FIFO отсчетов в канале напряжения блока F2
0x0F8	ADCUI_F2IDAT	Буфер FIFO отчетов в канале тока блока F2
0x0FC	ADCUI_F2VRMS	Среднеквадратическое значение канала напряжения в блоке F2
0x100	ADCUI_F2VRMS2	Квадрат RMS в канале напряжения блока F2
0x104	ADCUI_F2IRMS	Среднеквадратическое значение канала тока в блоке F2
0x108	ADCUI_F2IRMS2	Квадрат RMS в канале тока блока F2
0x10C	ADCUI_F2STAT	Статус каналов блока F2
0x110	ADCUI_F2MASK	Маска прерываний каналов в блоке F2
0x114	ADCUI_CCAL1	Регистр калибровки каналов тока (I0) и напряжения (V0) в блоке F0
0x118	ADCUI_CCAL2	Регистр калибровки каналов тока и напряжения в блоке F1
0x11C	ADCUI_CCAL3	Регистр калибровки каналов тока и напряжения в блоке F2
0x120	ADCUI_CCAL4	Регистра калибровки канала тока (I3) в блоке F0

13.1.1 ADCUI_CTRL1

Таблица 46 – Регистр ADCUI_CTRL1

Номер	31, 30	29	28	27	26...22
Доступ	R/W	-	R/W	R/W	-
Сброс	00	-	0	0	-
	OSR_CONF	-	RESET_DIG	ZXRMS	-

Номер	21	20	19	18, 17	16, 15
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	BUF_BYF	VREF_SEL	FREQSEL	VANOLOAD	VARNLOAD

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

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

Таблица 47 – Описание бит регистра ADCUI_CTRL1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	OSR_CONF*	Выбор коэффициента передискретизации: 00 – 256 (4 кГц); 01 – 128 (8 кГц); 10 – 64 (16 кГц); 11 – Зарезервировано
29	-	Зарезервировано
28	RESET_DIG	Сброс цифровой части блоков АЦП: 0 – нет сброса; 1 – цифровая часть под общим сбросом
27	ZXRMS**	Управления обновления регистров со среднеквадратическими значениями: 0 – непрерывное обновление; 1 – обновление при пересечении напряжением «0»
26...22	-	Зарезервировано
21	BUF_BYF	Буферизация опорного напряжения: 0 – опорное напряжение буферизировано; 1 – опорное напряжение небуферизировано
20	VREF_SEL	Выбор опорного напряжения для АЦП: 0 – внутреннее опорное напряжение (U _{ОБИАС}); 1 – внешнее опорное напряжение с вывода VR_1V

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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	ZXLPF	Отключение низкочастотного фильтра перед детектором пересечения «0» в каналах напряжения: 0 – фильтр включен; 1 – фильтр отключен
7	RESOL	Разрешение выходных данных: 0 – 16 бит; 1 – 24 бита
6	I3EN	Разрешение работы канала I3: 0 – канал отключен; 1 – канал включен
5	V2EN	Разрешение работы канала V2: 0 – канал отключен; 1 – канал включен
4	I2EN	Разрешение работы канала I2: 0 – канал отключен; 1 – канал включен
3	V1EN	Разрешение работы канала V1: 0 – канал отключен; 1 – канал включен

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

* Также необходимо учитывать, что увеличение частоты дискретизации в два раза ведет к уменьшению SNR как минимум на 3 дБ в полосе от 0 Гц до половины частоты дискретизации (это следует из того, что шум интегрируется в частоте два раза большей);

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

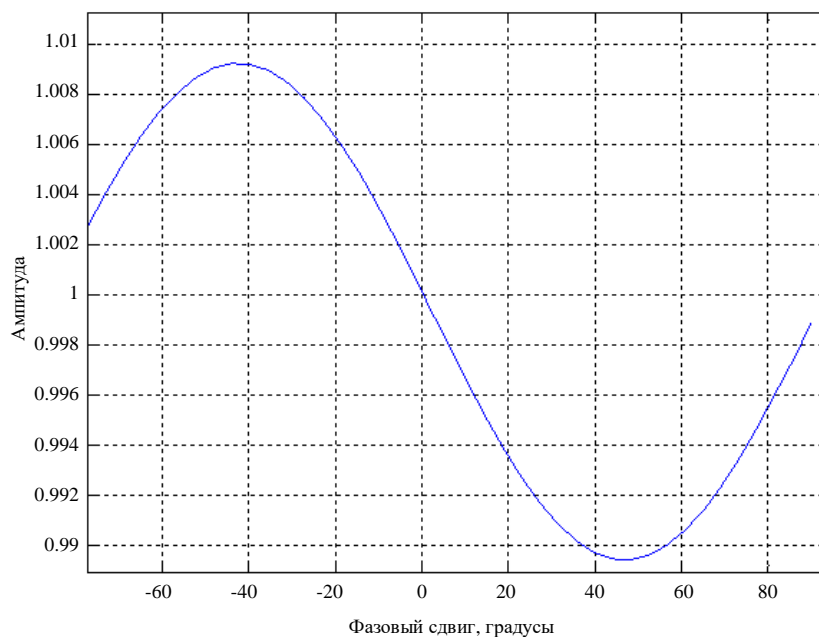


Рисунок 22 – Зависимость действующего значения тока от фазового сдвига

13.1.2 ADCUI_CTRL2

Таблица 48 – Регистр ADCUI_CTRL2

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

Таблица 49 – Описание бит регистра ADCUI_CTRL2

Разряды	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 50 – Регистр ADCUI_CTRL3

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

Таблица 51 – Описание бит регистра ADCUI_CTRL3

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

13.1.4 ADCUI_F0CTR

Таблица 52 – Регистр ADCUI_F0CTR

Номер	31...20	19, 18	17...10	9, 8	7, 6
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	00	00	0	0
	F0IRMSOS	F0I3GAIN	F0VPHASE	F0VGAIN	F0IGAIN

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

Таблица 53 – Описание бит регистра ADCUI_F0CTR

Разряды	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 54 – Регистр ADCUI_F0WC

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

Таблица 55 – Описание бит регистра ADCUI_F0WC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0WGAIN	Калибровка усиления 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 56 – Регистр ADCUI_F0WATTP

Номер	31...0
Доступ	RO
Сброс	
	F0WATTHRP

Таблица 57 – Описание бит регистра ADCUI_F0WATTP

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

13.1.7 ADCUI_F0WATTN

Таблица 58 – Регистр ADCUI_F0WATTN

Номер	31...0
Доступ	RO
Сброс	
	F0WATTHRN

Таблица 59 – Описание бит регистра ADCUI_F0WATTN

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

13.1.8 ADCUI_F0VC

Таблица 60 – Регистр ADCUI_F0VC

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

Таблица 61 – Описание бит регистра ADCUI_F0VC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0VARGAIN	Калибровка усиления 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 62 – Регистр ADCUI_F0VARP

Номер	31..0
Доступ	RO
Сброс	
	F0VARHRP

Таблица 63 – Описание бит регистра ADCUI_F0VARP

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

13.1.10 ADCUI_F0VARN

Таблица 64 – Регистр ADCUI_F0VARN

Номер	31..0
Доступ	RO
Сброс	
	F0VARHRN

Таблица 65 – Описание бит регистра ADCUI_F0VARN

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

13.1.11 ADCUI_F0AC

Таблица 66 – Регистр ADCUI_F0AC

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

Таблица 67 – Описание бит регистра ADCUI_F0AC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F0VAGAIN	Калибровка усиления 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 68 – Регистр ADCUI_F0VR

Номер	31...0
Доступ	RO
Сброс	
	F0VAHR

Таблица 69 – Описание бит регистра ADCUI_F0VR

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

13.1.13 ADCUI_F0MD0

Таблица 70 – Регистр ADCUI_F0MD0

Номер	31...20	29	28...12	11...9	8	7
Доступ	R/W	R/W	RO	-	R/W	R/W
Сброс	00	0		-	0	0
	F0SEL_I_CH	F0I3SEL	F0PER_FREQ	-	I3GAIN	V0GAIN

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

Таблица 71 – Описание бит регистра ADCUI_F0MD0

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...30	F0SEL_I_CH	Выбор активного канала тока для вычисления мощностных характеристик: 00, 11 – 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
28...12	F0PER_FREQ	Длительность такта в канале напряжения
11...9	-	Зарезервировано
8	I3GAIN	Усиление в канале I3: 0 – нет усиления; 1 – +6 дБ усиление
7	V0GAIN	Усиление в канале V0: 0 – нет усиления; 1 – +6 дБ усиление

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
6	IUGAIN	Усиление в канале I0: 0 – нет усиления; 1 – +6 дБ 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_F0IDAT): 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1...0	F0VSEL	Выбор источника сигнала для FIFO канала напряжения V0 (регистр ADCUI_F0VDAT): 00 – отсчеты напряжения до фильтра высоких частот; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты напряжения после фильтра высоких частот

13.1.14 ADCUI_F0MD1

Таблица 72 – Регистр ADCUI_F0MD1

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

Таблица 73 – Описание бит регистра ADCUI_F0MD1

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

13.1.15 ADCUI_F0VPEAK

Таблица 74 – Регистр ADCUI_F0VPEAK

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

Таблица 75 – Описание бит регистра ADCUI_F0VPEAK

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

13.1.16 ADCUI_F0IPEAK

Таблица 76 – Регистр ADCUI_F0IPEAK

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

Таблица 77 – Описание бит регистра ADCUI_F0IPEAK

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

13.1.17 ADCUI_F0VDAT

Таблица 78 – Регистр ADCUI_F0VDAT

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

Таблица 79 – Описание бит регистра ADCUI_F0VDAT

Разряды	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 80 – Регистр ADCUI_F0I0DAT

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

Таблица 81 – Описание бит регистра ADCUI_F0I0DAT

Разряды	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 82 – Регистр ADCUI_F0I3DAT

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

Таблица 83 – Описание бит регистра ADCUI_F0I3DAT

Разряды	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 84 – Регистр ADCUI_F0VRMS

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

Таблица 85 – Описание бит регистра ADCUI_F0VRMS

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

13.1.21 ADCUI_F0VRMS2

Таблица 86 – Регистр ADCUI_F0VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F0VRMS2

Таблица 87 – Описание бит регистра ADCUI_F0VRMS2

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

13.1.22 ADCUI_F0IRMS

Таблица 88 – Регистр ADCUI_F0IRMS

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

Таблица 89 – Описание бит регистра ADCUI_F0IRMS

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

13.1.23 ADCUI_F0IRMS2

Таблица 90 – Регистр ADCUI_F0IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F0IRMS2

Таблица 91 – Описание бит регистра ADCUI_F0IRMS2

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

13.1.24 ADCUI_F0STAT

Таблица 92 – Регистр ADCUI_F0STAT

Номер	31...27	26	25	24	23	22
Доступ	-	R/W	R/W	R/W	RO	RO
Сброс	-		0	0	0	0
	-	F0VAROVN	F0WATTOVN	C3IF_OVR	C3IF_FLL	C3IF_EMP

Номер	21	20	19	18	17	16
Доступ	R/W	RO	-	RO	R/W	RO
Сброс	0	0	-	0	0	0
	F0ZER OCSR	F0VAN LDFL	-	F0VAR NLDFL	F0VAR SIGN	F0AP NLDFL

Номер	15	14	13	12	11	10	9	8	7
Доступ	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0	0
	F0AP SIGN	F0FA ULTCON	F0IC HANNEL	F0Z XTOF	F0V AOV	F0VA ROVP	F0WA TTOVP	F0PE AKIF	F0PE AKVF

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	RO	RO	R/W	RO	RO
Сброс	0	0	0	0	0	0	0
	F0SAGF	F0IF_ OVR	F0IF_ FLL	F0IF_ EMP	F0VF_ OVR	F0VF_ FLL	F0VF_ EMP

Таблица 93 – Описание бит регистра ADCUI_F0STAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F0VAROVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной реактивной энергией. Запись «1» сбрасывает этот флаг
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
13	F0ICHANNEL	Активный канал тока: 0 – активный канал I0; 1 – активный канал I3
12	F0ZXTOF	Флаг, что в течении TimeOut не было пересечение напряжением значения «0». Запись «1» сбрасывает этот флаг
11	F0VAOV	Флаг, что произошло переполнение регистра аккумулятора с полной энергией. Запись «1» сбрасывает этот флаг
10	F0VAROVP	Флаг, что произошло переполнение регистра аккумулятора с положительной реактивной энергией. Запись «1» сбрасывает этот флаг

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
9	F0WATTOVP	Флаг, что произошло переполнение регистра аккумулятора с положительной активной энергией. Запись «1» сбрасывает этот флаг
8	F0PEAKIF	Флаг, что произошло превышение порогового значения тока. Запись «1» сбрасывает этот флаг
7	F0PEAKVF	Флаг, что произошло превышение порогового значения напряжения. Запись «1» сбрасывает этот флаг
6	F0SAGF	Флаг, что произошла просадка напряжения. Запись «1» сбрасывает этот флаг
5	F0IF_OVR	Флаг, что произошло переполнение FIFO 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 переполнение 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 94 – Регистр ADCUI_F0MASK

Номер	31...27	26	25	24	23	22
Доступ	-	R/W	R/W	R/W	RO	RO
Сброс	-		0	0	0	0
	-	F0VAR OVNM	F0WAT TOVNM	C3IF_ OVRM	C3IF_ FLLM	C3IF_ EMPM

Номер	21	20	19	18	17	16
Доступ	R/W	RO	-	RO	R/W	RO
Сброс	0	0	-	0	0	0
	F0ZEROCSRSM	F0VAN LDFLM	-	F0VAR NLDFLM	F0VAR SIGNM	F0AP NLDFLM

Номер	15	14	13	12	11	10	9	8	7
Доступ	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	-	0	0	0	0	0	0
	F0APS IGNM	F0FAUL TCONM	-	F0ZX TOFM	F0VA OVM	F0VAR OVPM	F0WAT TOVPM	F0PE AKIFM	F0PE AKVFM

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 95 – Описание бит регистра ADCUI_F0MASK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F0VAROVNM	Маска бита F0VAROVN
25	F0WATTOVNM	Маска бита F0WATTOVN
24	C3IF_OVRM	Маска бита C3IF_OVR
23	C3IF_FLLM	Маска бита C3IF_FLL
22	C3IF_EMPCM	Маска бита C3IF_EMP
21	F0ZEROCRSM	Маска бита F0ZEROCRS
20	F0VANLDFLM	Маска бита F0VANLDFL
19	-	Зарезервировано
18	F0VARNLDFLM	Маска бита F0VARNLDFL
17	F0VARSIGNM	Маска бита F0VARSIGN
16	F0APNLDFLM	Маска бита F0APNLDFL
15	F0APSIGNM	Маска бита F0APSIGNM
14	F0FAULTCONM	Маска бита F0FAULTCON
13	-	Зарезервировано
12	F0ZXTOFM	Маска бита F0ZXTOF
11	F0VAOVM	Маска бита F0VAOV
10	F0VAROVPM	Маска бита F0VAROVP
9	F0WATTOVPM	Маска бита F0WATTOVP
8	F0PEAKIFM	Маска бита F0PEAKIF
7	F0PEAKVFM	Маска бита F0PEAKVF
6	F0SAGFM	Маска бита F0SAGF
5	F0IF_OVRM	Маска бита F0IF_OVR
4	F0IF_FLLM	Маска бита F0IF_FLL
3	F0IF_EMPCM	Маска бита F0IF_EMP
2	F0VF_OVRM	Маска бита F0VF_OVR
1	F0VF_FLLM	Маска бита F0VF_FLL
0	F0VF_EMPCM	Маска бита F0VF_EMP

13.1.26 ADCUI_F1CTR

Таблица 96 – Регистр ADCUI_F1CTR

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

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

Таблица 97 – Описание бит регистра ADCUI_F1CTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F1IRMSOS	Калибровка вычислителя среднеквадратического значения тока
19...18	-	Зарезервировано
17...10	F1VPHASE	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	F1VGAIN	Предусилитель в канале напряжения: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
7, 6	F1IGAIN	Предусилитель в канале тока: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
5	F1RVRS	Запись «1» в этот бит сбрасывает счетчик переданной полной энергии
4	F1RRRS	Запись «1» в этот бит сбрасывает счетчик переданной реактивной энергии
3	F1RARS	Запись «1» в этот бит сбрасывает счетчик переданной активной энергии
2	F1VASEL	Выбор источника сигнала для сохранения в регистре полной энергии: 0 – полная энергия; 1 – среднеквадратическое значение тока
1	-	Зарезервировано
0	F1INTEN	Отключение интегратора в канале тока: 0 – интегратор включен; 1 – интегратор отключен

13.1.27 ADCUI_F1WC

Таблица 98 – Регистр ADCUI_F1WC

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

Таблица 99 – Описание бит регистра ADCUI_F1WC

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

13.1.28 ADCUI_F1WATTP

Таблица 100 – Регистр ADCUI_F1WATTP

Номер	31...0
Доступ	RO
Сброс	
	F1WATTHRP

Таблица 101 – Описание бит регистра ADCUI_F1WATTP

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

13.1.29 ADCUI_F1WATTN

Таблица 102 – Регистр ADCUI_F1WATTN

Номер	31...0
Доступ	RO
Сброс	
	F1WATTHRN

Таблица 103 – Описание бит регистра ADCUI_F1WATTN

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

13.1.30 ADCUI_F1VC

Таблица 104 – Регистр ADCUI_F1VC

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

Таблица 105 – Описание бит регистра ADCUI_F1VC

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

13.1.31 ADCUI_F1VARP

Таблица 106 – Регистр ADCUI_F1VARP

Номер	31...0
Доступ	RO
Сброс	
	F1VARHRP

Таблица 107 – Описание бит регистра ADCUI_F1VARP

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

13.1.32 ADCUI_F1VARN

Таблица 108 – Регистр ADCUI_F1VARN

Номер	31...0
Доступ	RO
Сброс	
	F1VARHRN

Таблица 109 – Описание бит регистра ADCUI_F1VARN

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

13.1.33 ADCUI_F1AC

Таблица 110 – Регистр ADCUI_F1AC

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

Таблица 111 – Описание бит регистра ADCUI_F1AC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F1VAGAIN	Калибровка усиления сигнала в ветви полной мощности для блока F1
15...12	-	Зарезервировано
11...0	F1VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения

13.1.34 ADCUI_F1VR

Таблица 112 – Регистр ADCUI_F1VR

Номер	31...0
Доступ	RO
Сброс	
	F1VAHR

Таблица 113 – Описание бит регистра ADCUI_F1VR

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

13.1.35 ADCUI_F1MD0

Таблица 114 – Регистр ADCUI_F1MD0

Номер	31..29	28..12	11..8	7	6	5	4	3, 2	1, 0
Доступ	-	RO	-	R/W	R/W	R	R	R/W	R/W
Сброс	-		-	0	0	0	0	00	00
	-	F1PER_FREQ	-	V1 GAIN	I1 GAIN	F1 REACTS	F1 ACTS	F1 ISEL	F1 VSEL

Таблица 115 – Описание бит регистра ADCUI_F1MD0

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...12	F1PER_FREQ	Длительность такта в канале напряжения
11...8	-	Зарезервировано
7	V1GAIN	Усиление в канале V1: 0 – нет усиления; 1 – +6 дБ усиление
6	I1GAIN	Усиление в канале I1: 0 – нет усиления; 1 – +6 дБ усиление

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5	FIREACTS	Знак реактивной энергии в последний период: 0 – минус; 1 – плюс
4	FFACTS	Знак активной энергии в последний период: 0 – минус; 1 – плюс
3...2	FISEL	Выбор источника сигнала для FIFO канала тока I1 (регистр ADCUI_F1IDAT): 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1...0	FVSEL	Выбор источника сигнала для FIFO канала напряжения V1 (регистр ADCUI_F1VDAT): 00 – отсчеты напряжения до фильтра высоких частот; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты напряжения после фильтра высоких частот

13.1.36 ADCUI_F1MD1

Таблица 116 – Регистр ADCUI_F1MD1

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

Таблица 117 – Описание бит регистра ADCUI_F1MD1

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

13.1.37 ADCUI_F1MD2

Таблица 118 – Регистр ADCUI_F1MD2

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

Таблица 119 – Описание бит регистра ADCUI_F1MD2

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

13.1.38 ADCUI_F1VPEAK

Таблица 120 – Регистр ADCUI_F1VPEAK

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

Таблица 121 – Описание бит регистра ADCUI_F1VPEAK

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

13.1.39 ADCUI_F1IPEAK

Таблица 122 – Регистр ADCUI_F1IPEAK

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

Таблица 123 – Описание бит регистра ADCUI_F1IPEAK

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

13.1.40 ADCUI_F1VDAT

Таблица 124 – Регистр ADCUI_F1VDAT

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

Таблица 125 – Описание бит регистра ADCUI_F1VDAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1VDAT	FIFO отсчетов напряжения (или одной из мощностей)

13.1.41 ADCUI_F1IDAT

Таблица 126 – Регистр ADCUI_F1IDAT

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

Таблица 127 – Описание бит регистра ADCUI_F1IDAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F1IDAT	FIFO отсчетов тока (или одной из мощностей)

13.1.42 ADCUI_F1VRMS

Таблица 128 –Регистр ADCUI_F1VRMS

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

Таблица 129 – Описание бит регистра ADCUI_F1VRMS

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

13.1.43 ADCUI_F1VRMS2

Таблица 130 – Регистр ADCUI_F1VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F1VRMS2

Таблица 131 – Описание бит регистра ADCUI_F1VRMS2

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

13.1.44 ADCUI_F1IRMS

Таблица 132 – Регистр ADCUI_F1IRMS

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

Таблица 133 – Описание бит регистра ADCUI_F1IRMS

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

13.1.45 ADCUI_F1IRMS2

Таблица 134 – Регистр ADCUI_F1IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F1IRMS2

Таблица 135 – Описание бит регистра ADCUI_F1IRMS2

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

13.1.46 ADCUI_F1STAT

Таблица 136 – Регистр ADCUI_F1STAT

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

Номер	20	19	18	17	16	15
Доступ	RO	-	RO	R/W	RO	R/W
Сброс	0	-	0	0	0	0
	F1VAN LDFL	-	F1VAR NLDFL	F1VAR SIGN	F1AP NLDFL	F1AP SIGN

Номер	14, 13	12	11	10	9	8	7
Доступ	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	-	0	0	0	0	0	0
	-	F1ZX TOF	F1VA OV	F1VA ROVP	F1WAT TOVP	F1PE AKIF	F1PE AKVF

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	RO	RO	R/W	RO	RO
Сброс	0	0	0	0	0	0	0
	F1SAGF	F1IF_ OVR	F1IF_ FLL	F1IF_ EMP	F1VF_ OVR	F1VF_ FLL	F1VF_ EMP

Таблица 137 – Описание бит регистра ADCUI_F1STAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано.
26	F1VAROVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной реактивной энергией. Запись «1» сбрасывает этот флаг
25	F1WATTOVN	Флаг, что произошло переполнения регистра аккумулятора с отрицательной активной энергией. Запись «1» сбрасывает этот флаг
24...22	-	Зарезервировано
21	F1ZERO CRS	Флаг, что произошло пересечение «0» в канале напряжения. Запись «1» сбрасывает этот флаг
20	F1VANLDFL	Полная мощность ниже уровня сравнения
19	-	Зарезервировано
18	F1VARNLDFL	Реактивная мощность ниже уровня сравнения
17	F1VARSIGN	Смена знака реактивной мощности. Запись «1» сбрасывает этот флаг
16	F1APNLDFL	Активная мощность ниже уровня сравнения.
15	F1APSIGN	Смена знака активной мощности. Запись «1» сбрасывает этот флаг
14, 13	-	Зарезервировано
12	F1ZXTOF	Флаг, что в течении TimeOut не было пересечение напряжением значения «0». Запись «1» сбрасывает этот флаг
11	F1VAOV	Флаг, что произошло переполнения регистра аккумулятора с полной энергией. Запись «1» сбрасывает этот флаг
10	F1VAROVP	Флаг, что произошло переполнения регистра аккумулятора с положительной реактивной энергией. Запись «1» сбрасывает этот флаг
9	F1WATTOVP	Флаг, что произошло переполнения регистра аккумулятора с положительной активной энергией. Запись «1» сбрасывает этот флаг.
8	F1PEAKIF	Флаг, что произошло превышение порогового значения тока. Запись «1» сбрасывает этот флаг
7	F1PEAKVF	Флаг, что произошло превышение порогового значения напряжения. Запись «1» сбрасывает этот флаг
6	F1SAGF	Флаг, что произошла просадка напряжения. Запись «1» сбрасывает этот флаг
5	F1IF_OVR	Флаг, что произошло переполнения FIFO F1IDAT. Запись «1» сбрасывает этот флаг
4	F1IF_FLL	Флаг, что FIFO F1IDAT заполнено
3	F1IF_EMP	Флаг, что FIFO F1IDAT пусто

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
2	F1VF_OVR	Флаг, что произошло переполнения FIFO F1VDAT. Запись «1» сбрасывает этот флаг
1	F1VF_FLL	Флаг, что FIFO F1VDAT заполнено
0	F1VF_EMP	Флаг, что FIFO F1VDAT пусто

13.1.47 ADCUI_F1MASK

Таблица 138 – Регистр ADCUI_F1MASK

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

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

Номер	15	14, 13	12	11	10	9	8	7
Доступ	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0		0	0	0	0	0	0
	F1APSIGNM	-	F1ZXTOFM	F1VAOVM	F1VAROVPM	F1WATTOVPM	F1PEAKIFM	F1PEAKVFM

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0
	F1SAGFM	F1IF_OVRM	F1IF_FLLM	F1IF_EMPM	F1VF_OVRM	F1VF_FLLM	F1VF_EMPM

Таблица 139 – Описание бит регистра ADCUI_F1MASK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F1VAROVNM	Маска бита F1VAROVN
25	F1WATTOVNM	Маска бита F1WATTOVN
24...22	-	Зарезервировано
21	F1ZEROCRSM	Маска бита F1ZEROCRS
20	F1VANLDFLM	Маска бита F1VANLDFL
19	-	Зарезервировано
18	F1VARNLDFLM	Маска бита F1VARNLDFL
17	F1VARSIGNM	Маска бита F1VARSIGN

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
16	F1APNLDFLM	Маска бита F1APNLDFL
15	F1APSIGNM	Маска бита F1APSIGNM
14, 13	-	Зарезервировано
12	F1ZXTOFM	Маска бита F1ZXTOF
11	F1VAOVM	Маска бита F1VAOV
10	F1VAROVPM	Маска бита F1VAROVP
9	F1WATTOVPM	Маска бита F1WATTOVP
8	F1PEAKIFM	Маска бита F1PEAKIF
7	F1PEAKVFM	Маска бита F1PEAKVF
6	F1SAGFM	Маска бита F1SAGF
5	F1IF_OVRM	Маска бита F1IF_OVR
4	F1IF_FLLM	Маска бита F1IF_FLL
3	F1IF_EMPM	Маска бита F1IF_EMP
2	F1VF_OVRM	Маска бита F1VF_OVR
1	F1VF_FLLM	Маска бита F1VF_FLL
0	F1VF_EMPM	Маска бита F1VF_EMP

13.1.48 ADCUI_F2CTR

Таблица 140 – Регистр ADCUI_F2CTR

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

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

Таблица 141 – Описание бит регистра ADCUI_F2CTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	F2IRMSOS	Калибровка вычислителя среднеквадратического значения тока
19, 18	-	Зарезервировано
17..10	F2VPHASE	Фазовый сдвиг канала напряжения относительно канала тока, записанный в дополнительном коде. От -126d (-123мкс) до +127d(+124мкс). «0» соответствует синфазному сигналу с током.
9, 8	F2VGAIN	Предусилитель в канале напряжения: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7, 6	F2IGAIN	Предусилитель в канале тока: 00 – 0dB; 01 – 6dB; 10 – 12dB; 11 – 18dB
5	F2RVRS	Запись «1» в этот бит сбрасывает счетчик переданной полной энергии
4	F2RRRS	Запись «1» в этот бит сбрасывает счетчик переданной реактивной энергии
3	F2RARS	Запись «1» в этот бит сбрасывает счетчик переданной активной энергии
2	F2VASEL	Выбор источника сигнала для сохранения в регистре полной энергии: 0 – полная энергия; 1 – среднееквадратическое значение тока
1	-	Зарезервировано
0	F2INTEN	Отключение интегратора в канале тока: 0 – интегратор включен; 1 – интегратор отключен

13.1.49 ADCUI_F2WC

Таблица 142 – Регистр ADCUI_F2WC

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

Таблица 143 – Описание бит регистра ADCUI_F2WC

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

13.1.50 ADCUI_F2WATTP

Таблица 144 – Регистр ADCUI_F2WATTP

Номер	31...0
Доступ	RO
Сброс	
	F2WATTHRP

Таблица 145 – Описание бит регистра ADCUI_F2WATTP

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

13.1.51 ADCUI_F2WATTN

Таблица 146 – Регистр ADCUI_F2WATTN

Номер	31...0
Доступ	RO
Сброс	
	F2WATTHRN

Таблица 147 – Описание бит регистра ADCUI_F2WATTN

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

13.1.52 ADCUI_F2VC

Таблица 148 – Регистр ADCUI_F2VC

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

Таблица 149 – Описание бит регистра ADCUI_F2VC

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

13.1.53 ADCUI_F2VARP

Таблица 150 – Регистр ADCUI_F2VARP

Номер	31...0
Доступ	RO
Сброс	
	F2VARHRP

Таблица 151 – Описание бит регистра ADCUI_F2VARP

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

13.1.54 ADCUI_F2VARN

Таблица 152 – Регистр ADCUI_F2VARN

Номер	31...0
Доступ	RO
Сброс	
	F2VARHRN

Таблица 153 – Описание бит регистра ADCUI_F2VARN

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

13.1.55 ADCUI_F2AC

Таблица 154 – Регистр ADCUI_F2AC

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

Таблица 155 – Описание бит регистра ADCUI_F2AC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27...16	F2VAGAIN	Калибровка усиления сигнала в ветви полной мощности для блока F2
15...12	-	Зарезервировано
11...0	F2VRMSOS	Калибровка вычислителя среднеквадратического значения напряжения

13.1.56 ADCUI_F2VR

Таблица 156 – Регистр ADCUI_F2VR

Номер	31...0
Доступ	R
Сброс	
	F2VAHR

Таблица 157 – Описание бит регистра ADCUI_F2VR

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

13.1.57 ADCUI_F2MD0

Таблица 158 – Регистр ADCUI_F2MD0

Номер	31..29	28..12	11..8	7	6	5	4	3, 2	1, 0
Доступ	-	RO	-	R/W	R/W	R	R	R/W	R/W
Сброс	-		-	0	0	0	0	00	00
	-	F2PER_FREQ	-	V2 GAIN	I2 GAIN	F2 REACTS	F2 ACTS	F2 ISEL	F2 VSEL

Таблица 159 – Описание бит регистра ADCUI_F2MD0

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	-	Зарезервировано
28...12	F2PER_FREQ	Длительность такта в канале напряжения
11...8	-	Зарезервировано
7	V2GAIN	Усиление в канале V2: 0 – нет усиления; 1 – +6 дБ усиление
6	I2GAIN	Усиление в канале I2: 0 – нет усиления; 1 – +6 дБ усиление
5	F2REACTS	Знак реактивной энергии в последний период: 0 – минус; 1 – плюс
4	F2ACTS	Знак активной энергии в последний период: 0 – минус; 1 – плюс
3, 2	F2ISEL	Выбор источника сигнала для FIFO канала тока I2 (регистр ADCUI_F2IDAT): 00 – отсчеты тока; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты полной мощности
1, 0	F2VSEL	Выбор источника сигнала для FIFO канала напряжения V2 (регистр ADCUI_F2VDAT): 00 – отсчеты напряжения до фильтра высоких частот; 01 – отсчеты активной мощности; 10 – отсчеты реактивной мощности; 11 – отсчеты напряжения после фильтра высоких частот

13.1.58 ADCUI_F2MD1

Таблица 160 – Регистр ADCUI_F2MD1

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

Таблица 161 – Описание бит регистра ADCUI_F2MD1

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

13.1.59 ADCUI_F2MD2

Таблица 162 – Регистр ADCUI_F2MD2

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

Таблица 163 – Описание бит регистра ADCUI_F2MD2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...0	F2PHASE	Фазовый сдвиг канала напряжения V1 по отношению к V0

13.1.60 ADCUI_F2VPEAK

Таблица 164 – Регистр ADCUI_F2VPEAK

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

Таблица 165 – Описание бит регистра ADCUI_F2VPEAK

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

13.1.61 ADCUI_F2IPEAK

Таблица 166 – Регистр ADCUI_F2IPEAK

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

Таблица 167 – Описание бит регистра ADCUI_F2IPEAK

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

13.1.62 ADCUI_F2VDAT

Таблица 168 – Регистр ADCUI_F2VDAT

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

Таблица 169 – Описание бит регистра ADCUI_F2VDAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2VDAT	FIFO отсчетов напряжения (или одной из мощностей)

13.1.63 ADCUI_F2IDAT

Таблица 170 – Регистр ADCUI_F2IDAT

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

Таблица 171 – Описание бит регистра ADCUI_F2IDAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	-	Зарезервировано
23...0	F2IDAT	FIFO отсчетов тока (или одной из мощностей)

13.1.64 ADCUI_F2VRMS

Таблица 172 – Регистр ADCUI_F2VRMS

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

Таблица 173 – Описание бит регистра ADCUI_F2VRMS

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

13.1.65 ADCUI_F2VRMS2

Таблица 174 – Регистр ADCUI_F2VRMS2

Номер	31...0
Доступ	RO
Сброс	
	F2VRMS2

Таблица 175 – Описание бит регистра ADCUI_F2VRMS2

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

13.1.66 ADCUI_F2IRMS

Таблица 176 – Регистр ADCUI_F2IRMS

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

Таблица 177 – Описание бит регистра ADCUI_F2IRMS

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

13.1.67 ADCUI_F2IRMS2

Таблица 178 – Регистр ADCUI_F2IRMS2

Номер	31...0
Доступ	RO
Сброс	
	F2IRMS2

Таблица 179 – Описание бит регистра ADCUI_F2IRMS2

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

13.1.68 ADCUI_F2STAT

Таблица 180 – Регистр ADCUI_F2STAT

Номер	6	5	4	3	2	1	0
Доступ	R/W	R/W	RO	RO	R/W	RO	RO
Сброс	0	0	0	0	0	0	0
	F2SA GF	F2IF_ OVR	F2IF_ FLL	F2IF_ EMP	F2VF_ OVR	F2VF_ FLL	F2VF_ EMP

Номер	15	14...13	12	11	10	9	8	7
Доступ	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	-	0	0	0	0	0	0
	F2AP SIGN	-	F2ZX TOF	F2VA OV	F2VAROVP	F2WATTOVP	F2PEAKIF	F2PEAKVF

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

Таблица 181 – Описание бит регистра ADCUI_F2STAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F2VAROVN	Флаг, что произошло переполнение регистра аккумулятора с отрицательной реактивной энергией. Запись «1» сбрасывает этот флаг
25	F2WATTOVN	Флаг, что произошло переполнение регистра аккумулятора с отрицательной активной энергией. Запись «1» сбрасывает этот флаг
24...22	-	Зарезервировано
21	F2ZEROCRS	Флаг, что произошло пересечение «0» в канале напряжения. Запись «1» сбрасывает этот флаг
20	F2VANLDFL	Полная мощность ниже уровня сравнения
19	-	Зарезервировано
18	F2VARNLDFL	Реактивная мощность ниже уровня сравнения.
17	F2VARSIGN	Смена знака реактивной мощности. Запись «1» сбрасывает этот флаг
16	F2APNLDFL	Активная мощность ниже уровня сравнения.
15	F2APSIGN	Смена знака активной мощности.

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		Запись «1» сбрасывает этот флаг
14...13	-	Зарезервировано.
12	F2ZXTOF	Флаг, что в течении TimeOut не было пересечения напряжением значения «0». Запись «1» сбрасывает этот флаг
11	F2VAOV	Флаг, что произошло переполнение регистра аккумулятора с полной энергией. Запись «1» сбрасывает этот флаг
10	F2VAROVP	Флаг, что произошло переполнение регистра аккумулятора с положительной реактивной энергией. Запись «1» сбрасывает этот флаг
9	F2WATTOVP	Флаг, что произошло переполнение регистра аккумулятора с положительной активной энергией. Запись «1» сбрасывает этот флаг
8	F2PEAKIF	Флаг, что произошло превышение порогового значения тока. Запись «1» сбрасывает этот флаг.
7	F2PEAKVF	Флаг, что произошло превышение порогового значения напряжения. Запись «1» сбрасывает этот флаг
6	F2SAGF	Флаг, что произошла просадка напряжения. Запись «1» сбрасывает этот флаг
5	F2IF_OVR	Флаг, что произошло переполнение FIFO F2IDAT. Запись «1» сбрасывает этот флаг
4	F2IF_FLL	Флаг, что FIFO F2IDAT заполнено
3	F2IF_EMP	Флаг, что FIFO F2IDAT пусто
2	F2VF_OVR	Флаг, что произошло переполнение FIFO F2VDAT. Запись «1» сбрасывает этот флаг
1	F2VF_FLL	Флаг, что FIFO F2VDAT заполнено
0	F2VF_EMP	Флаг, что FIFO F2VDAT пусто

13.1.69 ADCUI_F2MASK

Таблица 182 – Регистр ADCUI_F2MASK

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
	F2SA GFM	F2IF_ OVRM	F2IF_ FLLM	F2IF_ EMPM	F2VF_ OVRM	F2VF_ FLLM	F2VF_ EMPM

Номер	15	14, 13	12	11	10	9	8	7
Доступ	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	-	0	0	0	0	0	0
	F2AP SIGNM	-	F2ZX TOFM	F2VA OVM	F2VA ROVPM	F2WAT TOVPM	F2PEAK IFM	F2PEAK VFM

Номер	31..27	26	25	24..22	21	20	19	18	17	16
Доступ	-	R/W	R/W	-	R/W	RO	-	RO	R/W	RO
Сброс	-		0	-	0	0	-	0	0	0
	-	F2V ARO VNM	F2W ATTO VNM	-	F2Z ERO RSM	F2V ANLD FLM	-	F2V ARNL DFLM	F2V ARSI GNM	F2A PNLD FLM

Таблица 183 – Описание бит регистра ADCUI_F2MASK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	F2VAROVNM	Маска бита F2VAROVN
25	F2WATTOVNM	Маска бита F2WATTOVN
24...22	-	Зарезервировано
21	F2ZEROCRSM	Маска бита F2ZEROCRS
20	F2VANLDFLM	Маска бита F2VANLDFL
19	-	Зарезервировано
18	F2VARNLDFLM	Маска бита F2VARNLDFL
17	F2VARSIGNM	Маска бита F2VARSIGN
16	F2APNLDFLM	Маска бита F2APNLDFL
15	F2APSIGNM	Маска бита F2APSIGNM
14, 13	-	Зарезервировано
12	F2ZXTOFM	Маска бита F2ZXTOF
11	F2VAOVM	Маска бита F2VAOV
10	F2VAROVPM	Маска бита F2VAROVPM
9	F2WATTOVPM	Маска бита F2WATTOVPM
8	F2PEAKIFM	Маска бита F2PEAKIF
7	F2PEAKVFM	Маска бита F2PEAKVF
6	F2SAGFM	Маска бита F2SAGF
5	F2IF_OVRM	Маска бита F2IF_OVR
4	F2IF_FLLM	Маска бита F2IF_FLL
3	F2IF_EMPM	Маска бита F2IF_EMP
2	F2VF_OVRM	Маска бита F2VF_OVR
1	F2VF_FLLM	Маска бита F2VF_FLL
0	F2VF_EMPM	Маска бита F2VF_EMP

13.1.70 ADCUI_CCAL1

Таблица 184 – Регистр ADCUI_CCAL1

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

Таблица 185 – Описание бит регистра ADCUI_CCAL1

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

13.1.71 ADCUI_CCAL2

Таблица 186 – Регистр ADCUI_CCAL2

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

Таблица 187 – Описание бит регистра ADCUI_CCAL2

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

13.1.72 ADCUI_CCAL3

Таблица 188 – Регистр ADCUI_CCAL3

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

Таблица 189 – Описание бит регистра ADCUI_CCAL3

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

13.1.73 ADCUI_CCAL4

Таблица 190 – Регистр ADCUI_CCAL4

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

Таблица 191 – Описание бит регистра ADCUI_CCAL4

Разряды	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 23, 24 приведены два вида включения АЦП: полностью дифференциальное и недифференциальное включение.

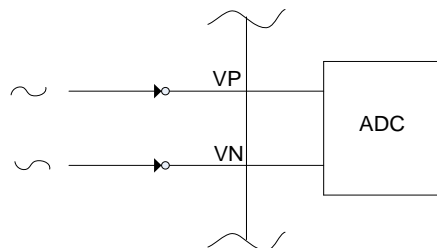


Рисунок 23 – Дифференциальное включение

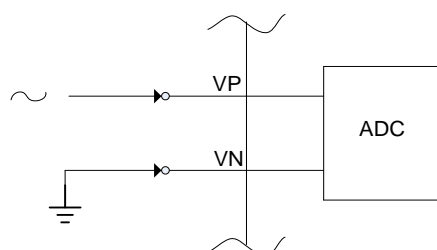


Рисунок 24 – Недифференциальное включение

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 25.

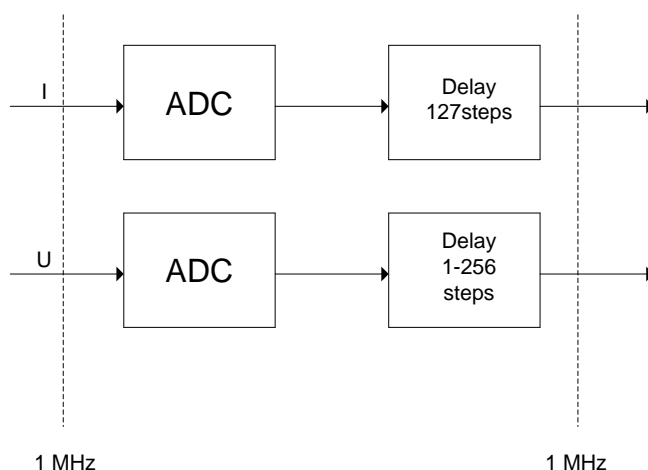


Рисунок 25 – Контролируемый фазовый сдвиг в канале напряжения

Изменяя линию задержки в канале напряжения можно регулировать временной сдвиг одного канала относительно другого. Так как частота отсчетов после АЦП равна 1,024 МГц, то один шаг равен 1/20480 периода сигнала с частотой 50 Гц или 0,018 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 26, 27:

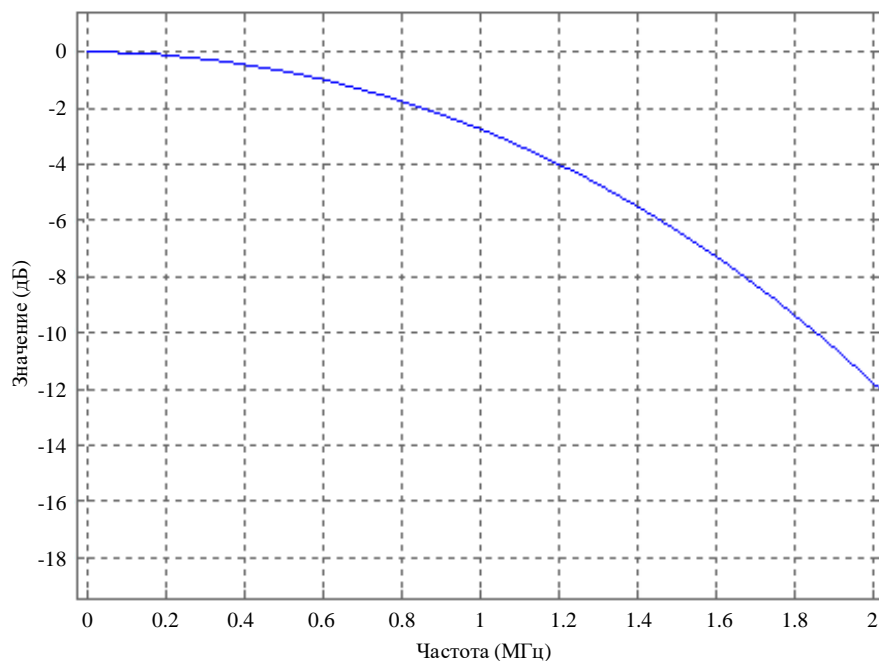


Рисунок 26 – Характеристика децимирующего фильтра в полосе 2 кГц

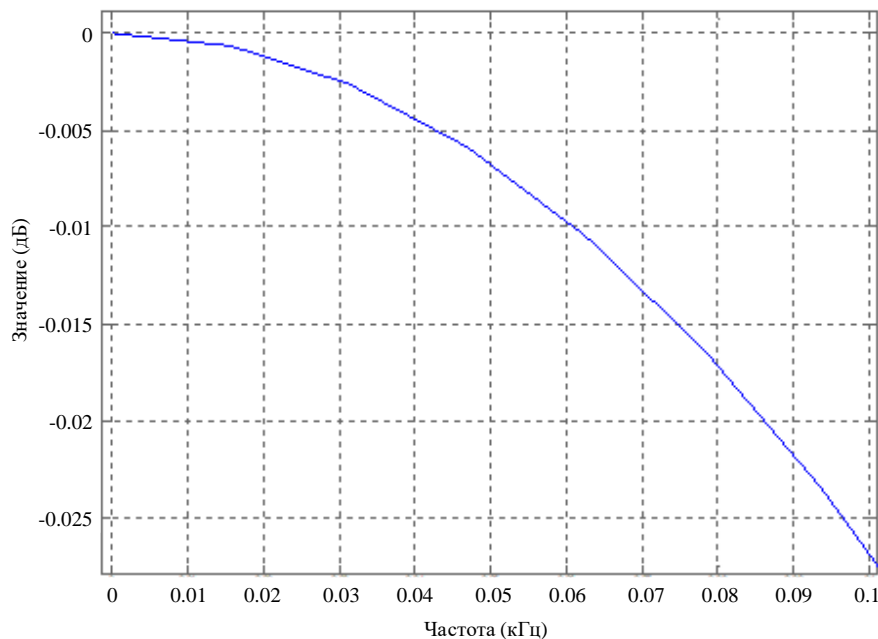


Рисунок 27 – Характеристика децимирующего фильтра в полосе 100 Гц

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 8 отсчетов, предназначенный для хранения отсчетов тока или напряжения, а также отсчетов активной мощности, реактивной мощности и полной мощности (только для каналов I0-I3). Выбор источника отсчетов для буфера FIFO осуществляется в поле F0I3SEL регистра ADCUI_F0MD0 для канала тока I3, и в полях FxISEL, FxVSEL регистра ADCUI_FxMD0 для остальных каналов. Запись выбранных отсчетов в буфер FIFO выполняется с частотой передискретизации, заданной в поле OSR_CONF регистра ADCUI_CTRL1.

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

Таблица 192 – Формулы перевода значения отсчетов FIFO в напряжение на входе

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

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

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

Если для 3 канала тока источник сигнала выбран до фильтра высоких частот (F0I3SEL = 1), то в значении, хранящемся в регистре ADCUI_F0I3DAT, не учитывается коэффициент IxBGAIN.

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

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

В регистрах FxVRMS и FxIRMS хранится вычисленная величина среднеквадратического значения тока и напряжения в соответствующей фазе. В таблице 193 приведены значения среднеквадратических величин.

Таблица 193 – Значения вадратов среднеквадратичных величин напряжения

Режим	Напряжение, Вольт	
Дифференциальное включение	$\frac{F_{xVRMS}}{2^{23}}$	$\frac{F_{xIRMS}}{2^{23}}$
Недифференциальное включение	$\frac{F_{xVRMS}}{2^{22}}$	$\frac{F_{xIRMS}}{2^{22}}$

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 4 кГц. Далее каждый отсчет возводится в квадрат, что дает следующий результат:

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

Таким образом, мы имеем сигнал с постоянной составляющей равной среднеквадратическому значению напряжения и пульсацией с удвоенной частотой по сравнению с входным сигналом. Для фильтрации пульсации полученный сигнал пропускается через фильтр с частотой среза 2 Гц. Этот фильтр подавляет пульсации на частоте 100 Гц (50 Гц · 2) с коэффициентом 35 дБ. Отфильтрованный сигнал поступает на блок извлечения квадратного корня. Результирующий сигнал 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 FxVRMSOS и FxIRMSOS. Перед корректировкой значение сдвигается на восемь 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:

$$V_{cor}(t) = V_{rms}(t) + \frac{F_{xVRMSOS}}{2^{20}}. \quad (5)$$

Значения FxVRMSOS и FxIRMSOS представлены в виде знаковых величин в двоичном коде с дополнением до 2.

В регистрах FxVRMS2 и FxIRMS2 хранятся значения среднеквадратического значения напряжения и тока до извлечения квадратного корня. В таблице 194 приведены значения квадратов среднеквадратических величин.

Таблица 194 – Значения вадратов среднеквадратичных величин напряжения

Режим	Напряжение, Вольт ²	
Дифференциальное включение	$\frac{FxVRMS\ 2}{2^{30}}$,	$\frac{FxIRMS\ 2}{2^{30}}$,
Недифференциальное включение	$\frac{FxVRMS\ 2}{2^{28}}$,	$\frac{FxIRMS\ 2}{2^{28}}$,

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 28 приведена его фазовая характеристика.

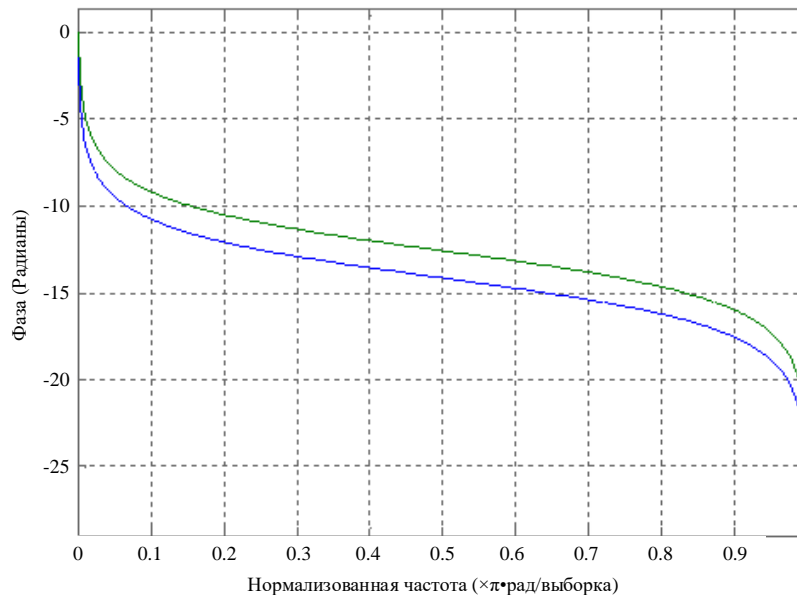


Рисунок 28 – Фильтр для сдвига сигнала на 90 градусов

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

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

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

Тогда мгновенное значение мощности равно произведению тока на напряжение:

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

Среднее значение мощности за целое количество тактов равно:

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

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

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

Таблица 195 – Формулы перевода значения отсчетов FIFO в мощность

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

Каждый из каналов мощности имеет независимую калибровку смещения (16 бит), а также усиления (12 бит). Перед корректировкой смещение сдвигается на 8 бит вправо, что уменьшает шаг корректировки в 256 раз. Калибровка осуществляется в соответствии 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 3 мощностей есть свой аккумулятор. Значение в них определяет потребленную энергию. В таблице 195 приведена формула перевода значения в Ватт • с.

Таблица 196 – Формула перевода значения в Ватт • с

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

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 трем фазам

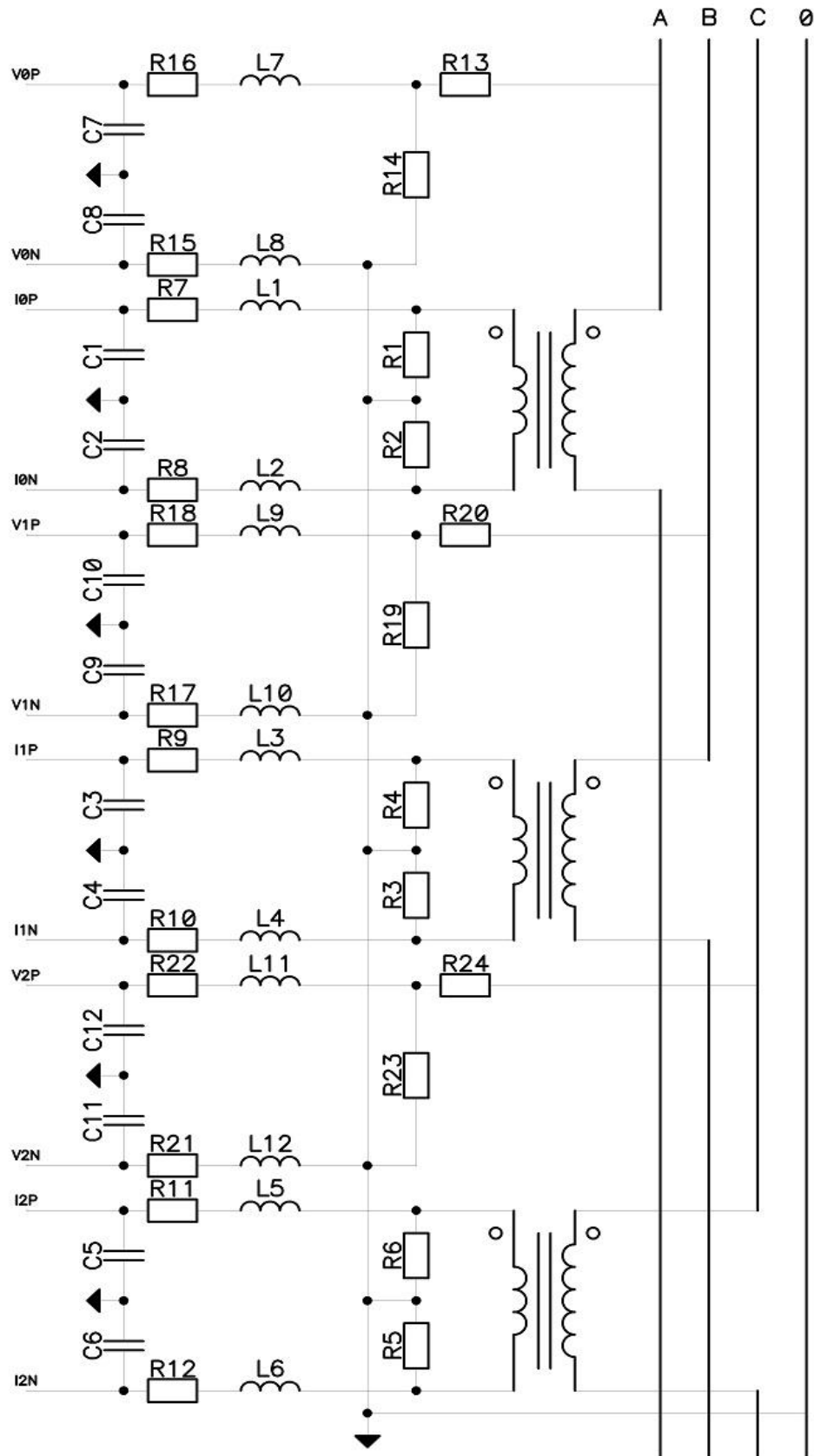


Рисунок 29 – Типовая схема включения для учета электроэнергии по трем фазам

Данная схема включения может быть использована как пример для конструирования трехфазных счетчиков электроэнергии. По каждой из трех фаз

установлен трансформатор тока. Выход трансформаторов нагружен на резисторный делитель. Сумма сопротивлений этих резисторов, например, $R1+R2$, должна соответствовать требуемой нагрузке выхода трансформатора. Оба резистора имеют одинаковое сопротивление и создают среднюю точку, относительно которой измеряется ток. В данной схеме включения средней точкой является аналоговая земля. С этой средней точкой соединена нейтраль трехфазной сети, если она используется. Для подачи сигнала на канал напряжения используется резисторный делитель. При выборе трансформатора и расчете резисторного делителя, стоит помнить, что амплитуда сигнала на входе АЦП не должна выходить за рамки $\pm 500\text{мВ}$ 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

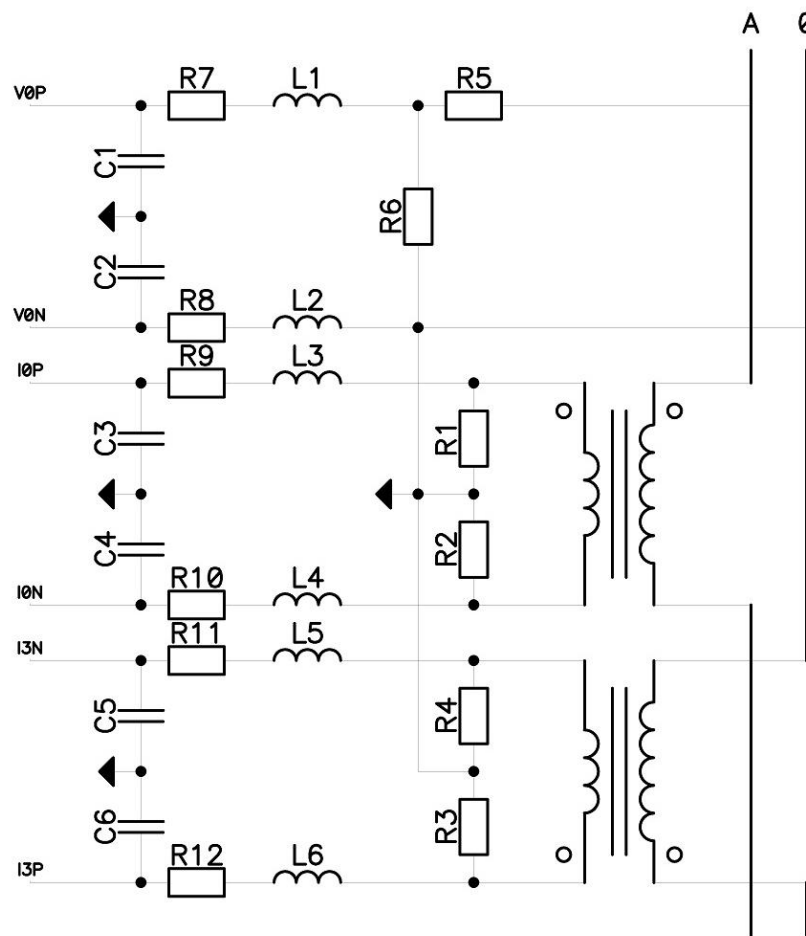


Рисунок 30 – Типовая схема включения для учета электроэнергии по одной фазе

В случае, когда необходимо построить однофазный счетчик электроэнергии на базе данной микросхемы, можно взять за основу схему, приведенную на рисунке 30. В данной схеме предусмотрено измерение тока в обоих проводах однофазной сети. Микросхема позволяет в автоматическом режиме учитывать то значение тока из каналов I0 и I3, которое будет больше. Если учета тока по «нулю» не требуется, то часть схемы, относящуюся к каналу I3 можно убрать. В остальном, назначение элементов данной схемы аналогично схеме для учета электроэнергии по трем фазам.

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

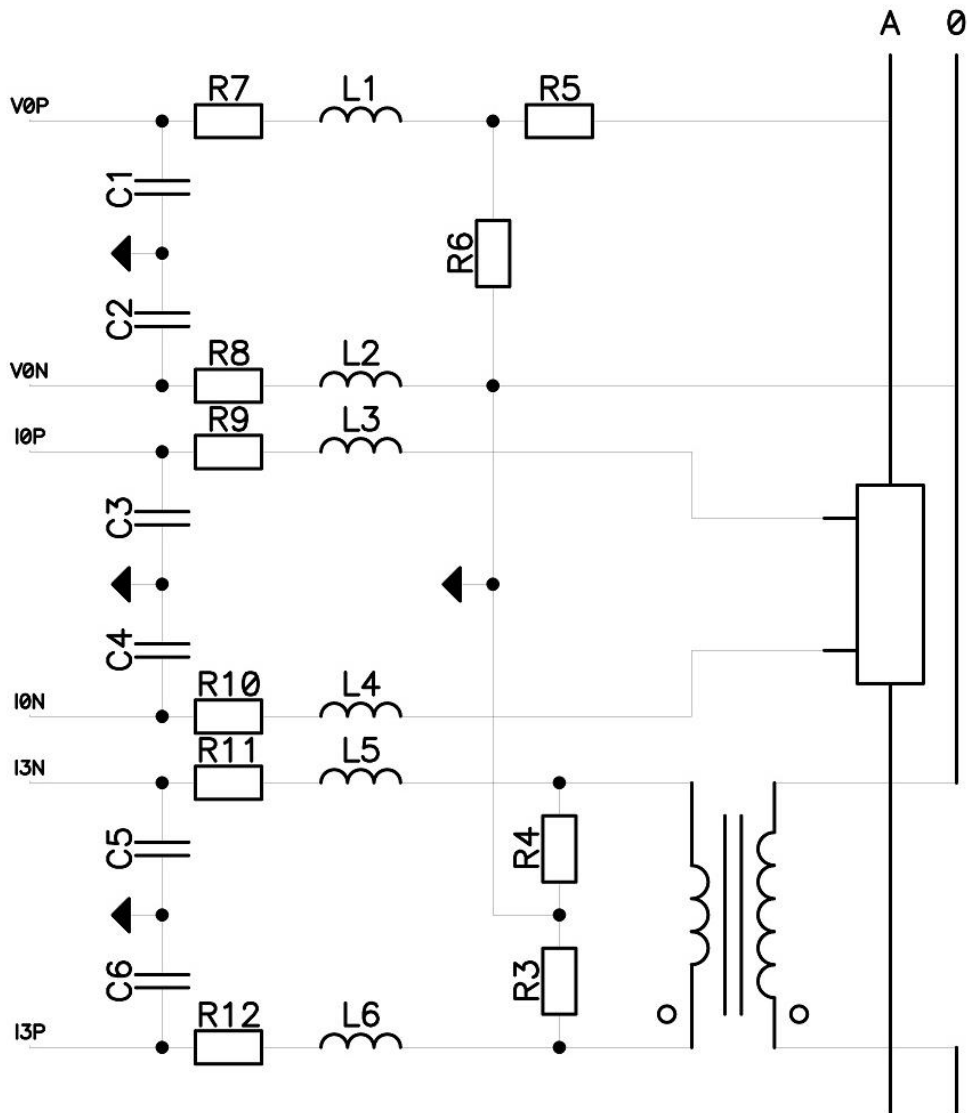


Рисунок 31 – Типовая схема включения для учета электроэнергии по одной фазе с использованием шунта

Также для измерения тока может быть использован шунт. В этом случае следует использовать схему на рисунке 31. Аналогично предыдущей схеме на трансформаторах, здесь можно исключить часть схемы, относящуюся к каналу тока I3, если контроля тока в нулевом проводе не требуется.

14 Аппаратный блок вычисления CRC

Микросхема имеет в своем составе блок для вычисления 16-битного CRC с произвольным полиномом. Контроллер принимает 32-битные слова и может их обрабатывать как в прямом порядке (начиная с младшего бита), так и в обратном (начиная со старшего бита). Скорость подсчета составляет 2 бита / PCLK (частота APB шины). Контроллер имеет FIFO на 4 отчета, а также канал DMA для загрузки новых слов. Запрос для DMA формируется, если в FIFO пусто. Контроллер начинает обрабатывать новые слова, как только они появляются в FIFO и обрабатывает до последнего слова. После обработки последнего слова выставляется флаг. Регистр CRC имеет доступ как на чтение (считать рассчитанное значение), так и на запись (установить начальное значение).

14.1 Описание регистров управления блока CRC

Таблица 197 – Описание регистров управления блока CRC

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

14.1.1 CRC_CTRL

Таблица 198 – Регистр CRC_CTRL

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

Таблица 199 – Описание бит регистра CRC_CTRL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...7	-	Зарезервировано
6, 5	DCSize	Размер данных при расчете CRC: 00 – вычисление для байт (8 бит), при этом DLSize может быть 00, 01, 10; 01 – вычисление для полуслов (16 бит), при этом DLSize может быть 01, 10; 10 – вычисление для слов (32 бит), при этом DLSize может быть только 10
4, 3	DLSize	Размер загружаемых данных: 00 – байт (8 бит), при этом загружаемый байт записывается в CRC_DATAI[7:0]

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

14.1.2 CRC_STAT

Таблица 200 – Регистр CRC_STAT

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

Таблица 201 – Описание бит регистра CRC_STAT

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

14.1.3 CRC_DATA1

Таблица 202 – Регистр CRC_DATA1

Номер	31...0
Доступ	W
Сброс	
	DATA_IN

Таблица 203 – Описание бит регистра CRC_DATA1

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

14.1.4 CRC_VAL

Таблица 204 – Регистр CRC_VAL

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

Таблица 205 – Описание бит регистра CRC_VAL

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

14.1.5 CRC_POL

Таблица 206 – Регистр CRC_POL

Номер	31...17	16...0
Доступ		R/W
Сброс		10000000000000001
	-	CRC_POL

Таблица 207 – Описание бит регистра CRC_POL

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

Ниже приведен результирующий полином

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

15 Сигналы тактовой частоты

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

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 мультиплексах управляемом регистре CPU_CLOCK.

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

При смене тактовой частоты процессорного ядра следует:

1. Настроить число тактов паузы для доступа к Flash-памяти, задаваемое полем Delay в регистре EEPROM_CMD контроллера Flash-памяти EEPROM_CNTRL.
2. Настроить режим работы встроенного регулятора напряжения DU_{CC}, задаваемый полем LOW регистра ВКР_REG_0E батарейного домена ВКР, совместно с дополнительной стабилизирующей нагрузкой для встроенного регулятора напряжения DU_{CC}, задаваемой полем SelectRI регистра ВКР_REG_0E батарейного домена ВКР.

Если настраиваемая частота процессорного ядра выше текущей, то данные настройки нужно проводить до смены частоты, в противном случае - после её смены.

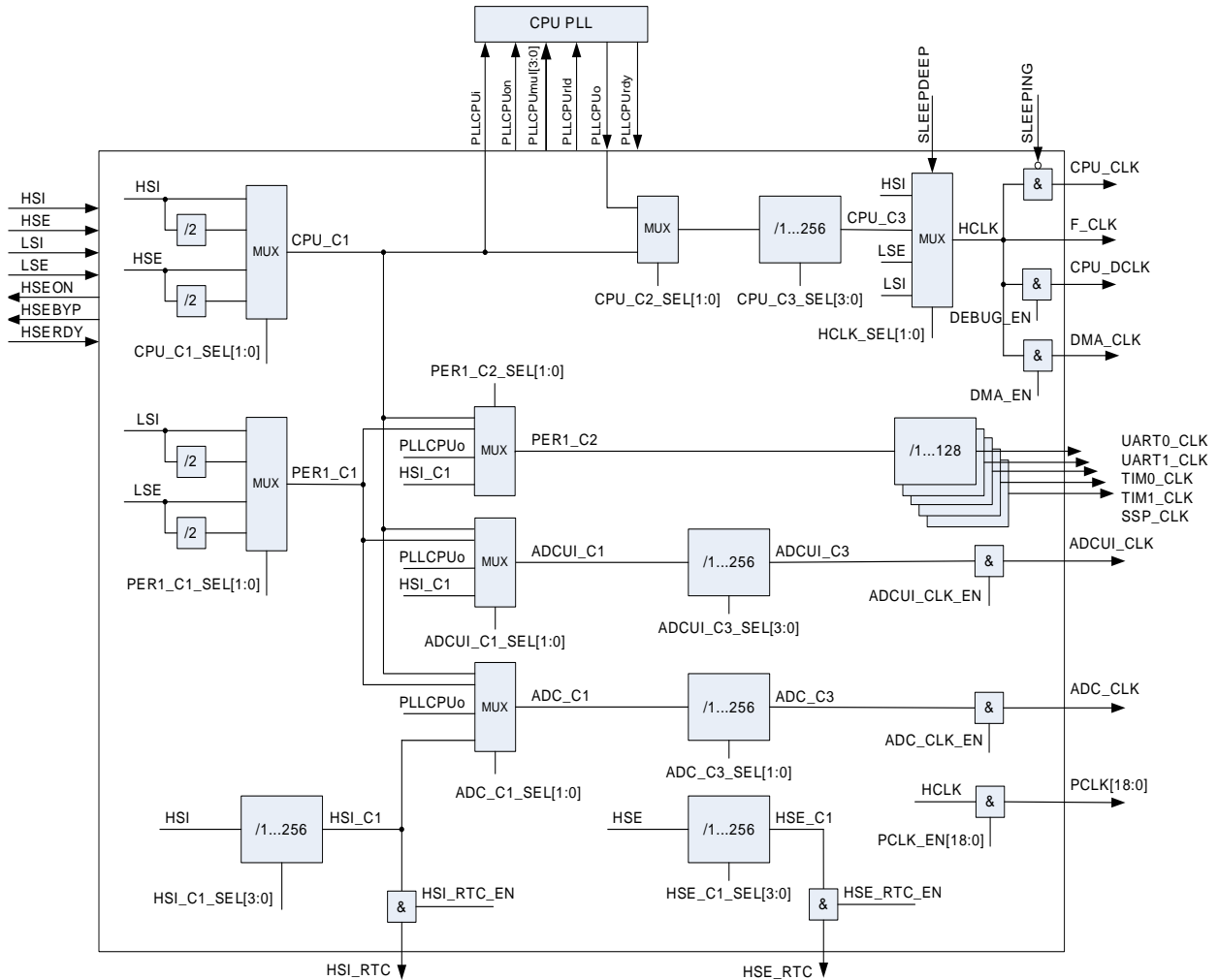


Рисунок 32 – Структурная блок-схема формирования тактовой частоты

15.1 Встроенный RC-Генератор HSI

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

15.2 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 t_{por} . При дальнейшей работе генератор LSI может быть отключен при помощи сигнала LSION в регистре BKP_REG_0F.

15.3 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. При выходе в нормальный режим работы вырабатывает сигнал HSERDY в регистре CLOCK_STATUS. Также осциллятор может работать в режиме HSEBYP, когда входная тактовая частота с входа OSC_IN проходит напрямую на выход HSE. Выход OSC_OUT находится в этом режиме третьем состоянии.

15.4 Внешний осциллятор LSE

Осциллятор LSE предназначен для выработки тактовой частоты 32 кГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания BDU_{CC} 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.

15.5 Встроенный блок умножения системной тактовой частоты

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 16, задаваемых на входе PLLCPUMUL[3:0] в регистре PLL_CONTROL. 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.

15.6 Описание регистров блока контроллера тактовой частоты

Таблица 208 – Описание регистров блока контроллера тактовой частоты

Базовый Адрес	Название	Описание
0x4002_0000	RST_CLK	Контроллер тактовой частоты
Смещение		
0x00	CLOCK_STATUS	Регистр состояния блока управления тактовой частотой
0x04	PLL_CONTROL	Регистр управления блоками умножения частоты
0x08	HS_CONTROL	Регистр управления высокочастотным генератором и осциллятором
0x0C	CPU_CLOCK	Регистр управления тактовой частотой процессорного ядра
0x10	PER1_CLOCK	Регистр управления тактовой частотой периферийных

Базовый Адрес	Название	Описание
		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

15.6.1 CLOCK_STATUS

Таблица 209 – Регистр CLOCK_STATUS

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

Таблица 210 – Описание бит регистра CLOCK_STATUS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...3	-	Зарезервировано
2	HSE RDY	Флаг выхода в рабочий режим осциллятора HSE: 0 – осциллятор не запущен или не стабилен; 1 – осциллятор запущен и стабилен
1	PLL CPU RDY	Флаг выхода в рабочий режим CPU PLL: 0 – PLL не запущена или не стабильна; 1 – PLL запущена и стабильна
0	-	Зарезервировано

15.6.2 PLL_CONTROL

Таблица 211 – Регистр PLL_CONTROL

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

Таблица 212 – Описание бит регистра PLL_CONTROL

Разряды	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 ноль
2	PLL CPU ON	Бит включения PLL: 0 – PLL выключена; 1 – PLL включена
1, 0	-	Зарезервировано

15.6.3 HS_CONTROL

Таблица 213 – Регистр HS_CONTROL

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

Таблица 214 – Описание бит регистра HS_CONTROL

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

15.6.4 CPU_CLOCK

Таблица 215 – 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 216 – Описание бит регистра CPU_CLOCK

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

15.6.5 PER1_CLOCK

Таблица 217 – Регистр PER1_CLOCK

Номер	31...6	5	4	3, 2	1, 0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	1	1	00	00
	-	DMA_EN	DEBUG_EN	PER1_C2 SEL[1:0]	PER1_C1 SEL[1:0]

Таблица 218 – Описание бит регистра PER1_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...6	-	Зарезервировано
5	DMA_EN	Бит разрешения тактирования DMA контроллера
4	DEBUG_EN	Бит разрешения тактирования блока отладки ядра
3, 2	PER1 C2 SEL[1:0]	Биты выбора источника для PER1_C2: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_CLK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1, 0	PER1 C1 SEL[1:0]	Биты выбора источника для PER1_C1: 00 – LSI; 01 – LSI/2; 10 – LSE; 11 – LSE/2

15.6.6 ADC_CLOCK

Таблица 219 – 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 220 – Описание бит регистра ADC_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14	-	Зарезервировано
13	ADC CLK EN	Бит разрешения выдачи тактовой частоты ADC CLK: 0 – запрещен; 1 – разрешен
12	ADCUI CLK EN	Бит разрешения выдачи тактовой частоты ADCUI CLK: 0 – запрещен; 1 – разрешен
11...8	ADC C3 SEL[3:0]	Биты выбора делителя для ADC_C3: 0xxx – ADC_C3 = ADC_C1; 1000 – ADC_C3 = ADC_C1 / 2; 1001 – ADC_C3 = ADC_C1 / 4; 1010 – ADC_C3 = ADC_C1 / 8; ... 1111 – ADC_C3 = ADC_C1 / 256
7...4	ADCUI C3 SEL[3:0]	Биты выбора делителя для ADCUI_C3: 0xxx – ADCUI_C3 = ADCUI_C1; 1000 – ADCUI_C3 = ADCUI_C1 / 2; 1001 – ADCUI_C3 = ADCUI_C1 / 4; 1010 – ADCUI_C3 = ADCUI_C1 / 8; ... 1111 – ADCUI_C3 = ADCUI_C1 / 256
3, 2	ADCUI C1 SEL[1:0]	Биты выбора источника для ADCUI_C1: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_CLK

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

15.6.7 RTC_CLOCK

Таблица 221 – 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 222 – Описание бит регистра RTC_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	-	Зарезервировано
9	HSI_RTC_EN	Бит разрешения 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

15.6.8 PER2_CLOCK

Таблица 223 – Регистр PER2_CLOCK

Номер	19...0
Доступ	R/W
Сброс	0
	PCLK_EN[19:0]

Таблица 224 – Описание бит регистра PER2_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
19...0	PCLK EN[19:0]	Биты разрешения тактирования периферийных блоков: 0 – запрещено; 1 – разрешено. PCLK[0] – SSP; PCLK[1] – UART0; PCLK[2] – UART1; PCLK[3] – EEPROM; PCLK[4] – RST_CLK; PCLK[5] – DMA; 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; PCLK[19] – CRC

15.6.9 TIM_CLOCK

Таблица 225 – 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 226 – Описание бит регистра TIM_CLOCK

Разряды	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
23...16	-	Зарезервировано

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

15.6.10 UART_CLOCK

Таблица 227 – 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 228 – Описание бит регистра UART_CLOCK

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

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

15.6.11 SSP_CLOCK

Таблица 229 – Регистр SSP_CLOCK

Номер	31...25	24	23...8	7...0
Доступ	U	R/W	U	R/W
Сброс	0	0	0	00000000
	-	SSP CLK EN	-	SSP BRG [7:0]

Таблица 230 – Описание бит регистра SSP_CLOCK

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

16 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. 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 (батарея), то батарейный домен остается включенным и может выполнять свои функции.

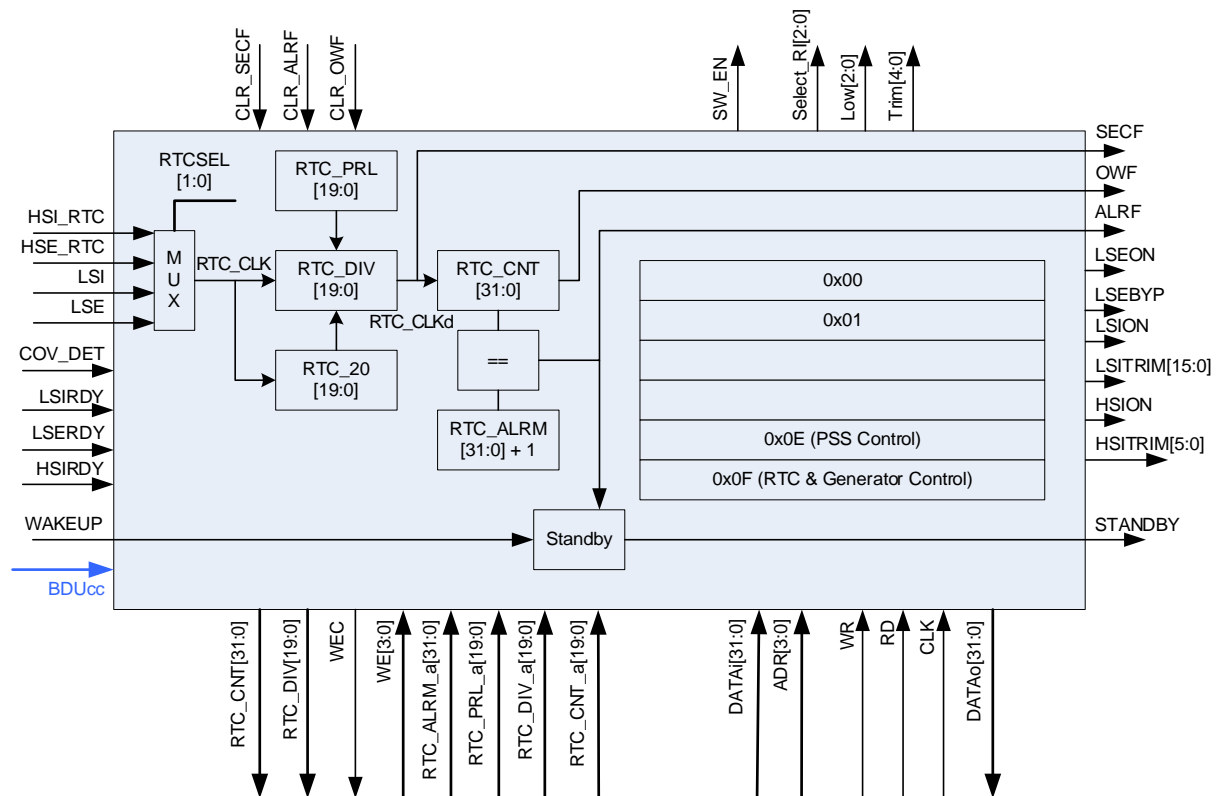


Рисунок 33 – Структурная блок-схема батарейного домена и часов реального времени

16.1 Часы реального времени

Часы реального времени позволяют организовать механизм отсчета времени в кристалле, в том числе при отключении основного источника питания. Включение часов реального времени осуществляется битом RTC_EN . В качестве источника тактовой частоты часов реального времени RTC_CLK может выступать генератор LSI или LSE , а также генератор HSI (частота HSI_RTC) или HSE (частота HSE_RTC) с дополнительным делителем до 256. Частоты HSI_RTC и HSE_RTC формируются в блоке управления тактовыми частотами RST_CLK и могут быть выбраны только при наличии питания 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 STANDBY, отключающий внутренний регулятор напряжения автоматически сбрасывается при превышении RTC_CNT значения RTC_ALRM.

Бит STANDBY 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.

16.2 Регистры аварийного сохранения

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

16.3 Описание регистров блока батарейного домена

Таблица 231 – 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
...		
0x38	ВКР_REG_0E	Регистр аварийного сохранения 14
0x3C	ВКР_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

16.3.1 ВКР_REG_00

16.3.2 ВКР_REG_01

16.3.3 ВКР_REG_02

16.3.4 ВКР_REG_03

16.3.5 ВКР_REG_04

16.3.6 ВКР_REG_05

16.3.7 ВКР_REG_06

16.3.8 ВКР_REG_07

16.3.9 ВКР_REG_08

16.3.10 ВКР_REG_09

16.3.11 ВКР_REG_0A

16.3.12 ВКР_REG_0B

16.3.13 ВКР_REG_0C

16.3.14 ВКР_REG_0D

Таблица 232 – Регистры ВКР_REG_[0D...00]

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

Таблица 233 – Описание бит регистров ВКР_REG_[0D...00]

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

16.3.15 ВКР_REG_0E

Таблица 234 – Регистр ВКР_REG_0E

Номер	31..16	15	14	13, 12	11
Доступ		R/W	R/W	R/W	R/W
Сброс		0	0	00	0
	-	ilimen	COVDET	Trim[4:3]	FPOR

Номер	10...8	7, 6	5...3	2...0
Доступ	R/W	R/W	R/W	R/W
Сброс	000	10	000	000
	Trim[2:0]	SW_EN[1:0]	SelectRI[2:0]	LOW[2:0]

Таблица 235 – Описание бит регистра ВКР_REG_0E

Разряды	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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
10...8	Trim[2:0]	Коэффициент настройки опорного напряжения встроенного регулятора напряжения DU _{CC} . С помощью Trim осуществляется подстройка напряжения DU _{CC} : 000 – DU _{CC} + 0,10 В – значение по умолчанию; 001 – DU _{CC} + 0,06 В; 010 – DU _{CC} + 0,04 В; 011 – DU _{CC} + 0,01 В; 100 – DU _{CC} – 0,01 В; 101 – DU _{CC} – 0,04 В; 110 – DU _{CC} – 0,06 В; 111 – DU _{CC} – 0,10 В
7, 6	SW_EN[1:0]	Разрешение работы порта Serial Wire: 00 – запрещен; 01-11 – разрешен
5...3	SelectRI[2:0]	Выбор дополнительной стабилизирующей нагрузки для встроенного регулятора напряжения DU _{CC} : 000 – ~6 кОм (дополнительный ток потребления 300 мкА); 001 – ~270 кОм (дополнительный ток потребления 6,6 мкА); 010 – ~90 кОм (дополнительный ток потребления 20 мкА); 011 – ~24 кОм (дополнительный ток потребления 80 мкА); 100 – ~900 кОм (собственное потребление 2 мкА); 101 – ~2 кОм (дополнительный ток потребления 900 мкА); 110 – ~400 Ом (дополнительный ток потребления 4,4 мА); 111 – ~100 Ом (дополнительный ток потребления 19 мА)
2...0	LOW[2:0]	Выбор режима работы встроенного регулятора напряжения DU _{CC} . Значение LOW должно совпадать со 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

16.3.16 ВКР_REG_0F

Таблица 236 – Регистр ВКР_REG_0F

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]

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

Таблица 237 – Описание бит регистра ВКР_REG_0F

Разряды	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 _{CC} на 1,8 В: 0 – регулятор включен и выдает напряжение; 1 – выключение регулятора. Триггер сбрасывается по событию ALRF или по низкому уровню на выводе WAKEUP
29...24	HSI TRIM[5:0]	Коэффициент подстройки частоты генератора HSI. Смотри диаграмму зависимости
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
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:

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		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: 00 – LSI; 01 – LSE; 10 – HSI_RTC (формируется в блоке RST_CLK); 11 – HSE_RTC (формируется в блоке RST_CLK)
1	LSE BYP	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

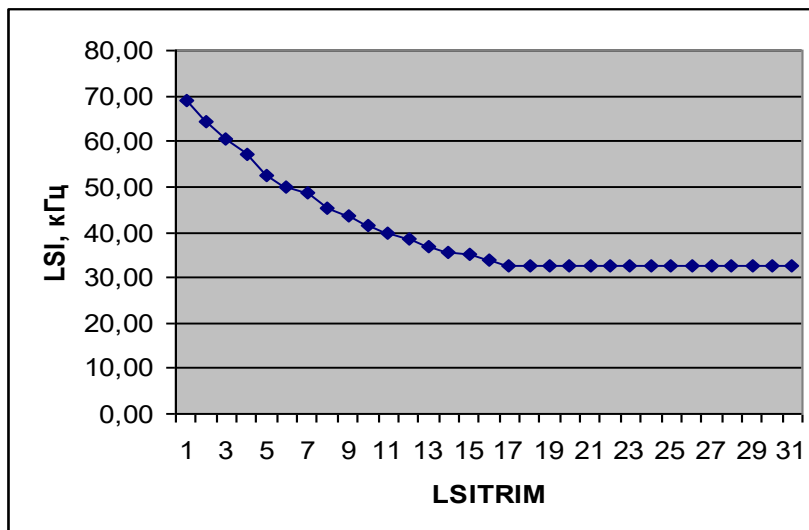


Рисунок 34 – Зависимость частоты LSI от значения LSITRIM

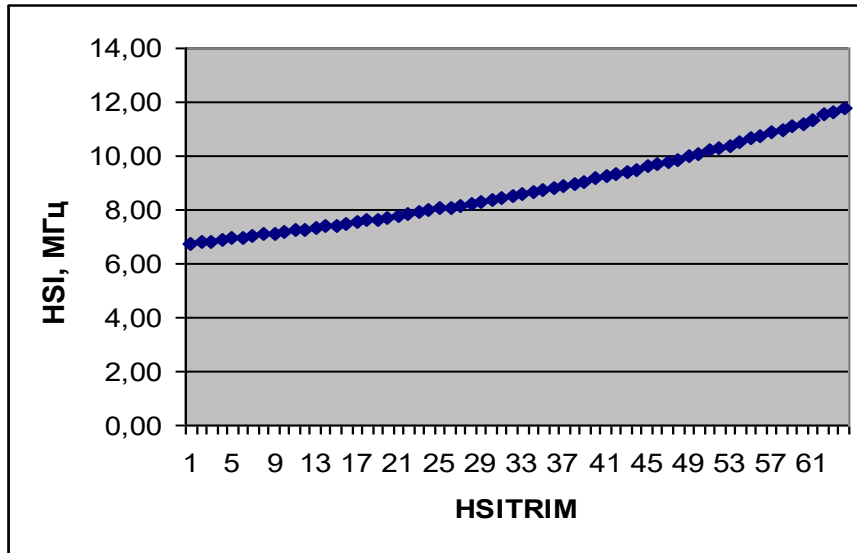


Рисунок 35 – Зависимость частоты HSI от значения HSITRIM

16.3.17 RTC_CNT

Таблица 238 – Регистр RTC_CNT

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

Таблица 239 – Описание бит регистра RTC_CNT

Разряды	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

16.3.18 RTC_DIV

Таблица 240 – Регистр RTC_DIV

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

Таблица 241 – Описание бит регистра RTC_DIV

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

16.3.19 RTC_PRL

Таблица 242 – Регистр RTC_PRL

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

Таблица 243 – Описание бит регистра RTC_PRL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	-	-
19...0	RTC_PRL[19:0]	Коэффициент деления тактовой частоты RTC_CLK: $RTC_CLKd = RTC_CLK / (RTC_PRL + 1)$

16.3.20 RTC_ALRM

Таблица 244 – Регистр RTC_ALRM

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

Таблица 245 – Описание бит регистра RTC_ALRM

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

16.3.21 RTC_CS

Таблица 246 – 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 247 – Описание бит регистра RTC_CS

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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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: 0 – нет события превышения; 1 – было событие превышения. Флаг устанавливается при $RTC_CNT == (RTC_ALRM+1)$, 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

17 Порты ввода-вывода

Микросхема имеет три порта ввода-вывода. Порты 16-разрядные и их выходы мультиплексируются между различными функциональными блоками, управление для каждого вывода порта отдельное. Для того, чтобы выходы порта перешли под управление того или иного периферийного блока, необходимо задать для нужных выводов выполняемую функцию и настройки.

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 248 – Порты ввода-вывода

Вывод	Аналоговая функция ANALOG_EN=0		Цифровая функция			
			Порт IO MODE[1:0]=00 ANALOG_EN=1	Основная MODE[1:0]=01 ANALOG_EN=1	Альтернативная MODE[1:0]=10 ANALOG_EN=1	
Порт А						
PA0	-		PA0	TMR0_CH1	5	-
PA1	-		PA1	TMR0_CH1N		-
PA2	-		PA2	TMR0_CH2		-
PA3	-		PA3	TMR0_CH2N		-
PA4	-		PA4	TMR0_CH3		-
PA5	-		PA5	TMR0_CH3N		-
PA6	-		PA6 SWCLKTCK	TMR0_CH4		-
PA7	-		PA7 SWDIO	TMR0_CH4N		-
PA8	-		PA8	TMR0_ETR		-
PA9	-		PA9	TMR0_BRK		-
PA10	-		PA10	EXT_INT0		-
PA11	-		PA11	-		-
PA12	-		PA12	SSP_FSS	6	-
PA13	-		PA13	SSP_CLK		-
PA14	-		PA14	SSP_RXD		-
PA15	-		PA15	SSP_TXD		-
Порт В						
PB0	-		PB0 MODE0	UART0_TXD	7	-
PB1	-		PB1	UART0_RXD		-
PB2	-		PB2	nSIROUT0		-
PB3	-		PB3	nSIRIN0		-
PB4	OSC_IN32	1	PB4	nUART0DTR		-
PB5	OSC_OUT32		PB5	nUART0RTS		-
PB6	ADC7S	2	PB6	nUART0RI		EXT_INT1
PB7	ADC6S		PB7	nUART0DCD		EXT_INT2
PB8	ADC5S		PB8	nUART0DSR		TMR1_ETR
PB9	ADC4S		PB9	nUART0CTS		TMR1_BRK

Вывод	Аналоговая функция ANALOG_EN=0		Цифровая функция			
			Порт IO MODE[1:0]=00 ANALOG_EN=1	Основная MODE[1:0]=01 ANALOG_EN=1		Альтернативная MODE[1:0]=10 ANALOG_EN=1
PB10	-		PB10	TMR1_CH2	8	-
PB11	-		PB11	TMR1_CH2N		-
PB12	-		PB12	TMR1_CH3		-
PB13	-		PB13	TMR1_CH3N		-
PB14	-		PB14	TMR1_CH4		-
Порт C						
PC0	-		PC0 MODE1	UART1_TXD	9	-
PC1	ADC3S	3	PC1	UART1_RXD		-
PC2	ADC2S/	4	PC2	TMR1_CH1	10	-
PC3	ADC1S/ ADCS_REF+		PC3	TMR1_CH1N		-
PC4	ADC0S/ ADCS_REF-		PC4	EXT_INT1	-	
PC5	-		PC5	EXT_INT2	-	
PC6	-	PC6	TMR1_ETR	11	-	
PC7	-	PC7	TMR1_BRK		-	
<p>Обозначения</p> <p>1 – Генератор LSE;</p> <p>2,3 – АЦП последовательного приближения;</p> <p>4 – АЦП последовательного приближения;</p> <p>5 – Таймер 0;</p> <p>6 – Последовательный интерфейс SSP;</p> <p>7 – UART0;</p> <p>8, 10, 12 – Таймер 1;</p> <p>9 – UART1</p>						

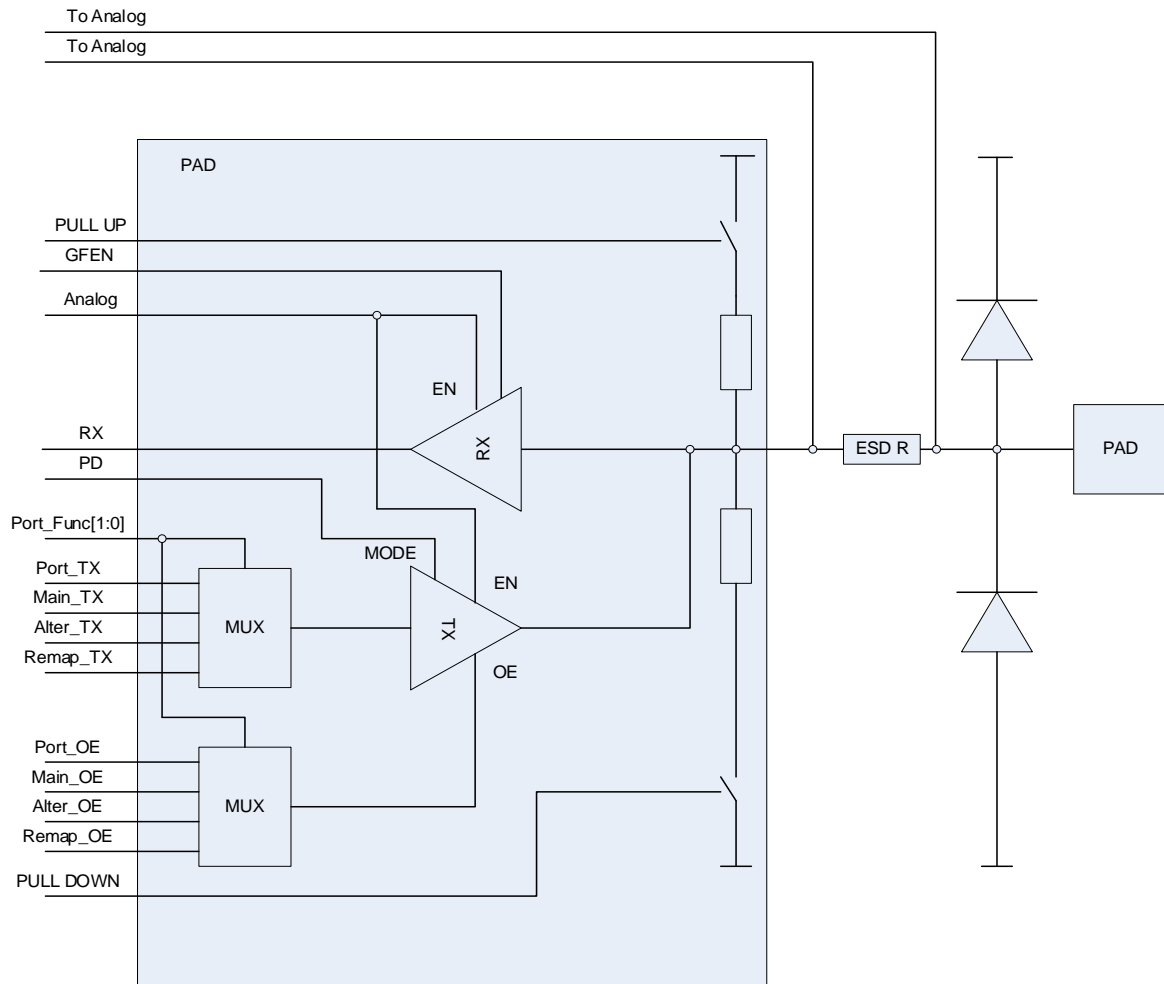


Рисунок 36 – Блок схема разряда порта ввода-вывода

17.1 Описание регистров портов ввода-вывода

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 249 – 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

Базовый Адрес	Название	Описание
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

17.1.1 PORTx_RXTX

Таблица 250 – Регистр PORTx_RXTX

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

Таблица 251 – Описание бит регистра PORTx_RXTX

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

17.1.2 PORTx_OE

Таблица 252 – Регистр PORTx_OE

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

Таблица 253 – Описание бит регистра PORTx_OE

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

17.1.3 PORTx_FUNC

Таблица 254 – 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 255 – Описание бит регистра PORTx_FUNC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	MODEx	Аналогично MODE0 для остальных битов порта
1...0	MODE0[1:0]	Режим работы вывода порта: 00 – порт; 01 – основная функция; 10 – альтернативная функция; 11 – переопределенная функция

17.1.4 PORTx_ANALOG

Таблица 256 – Регистр PORTx_ANALOG

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

Таблица 257 – Описание бит регистра PORTx_ANALOG

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

17.1.5 PORTx_PULL

Таблица 258 – Регистр PORTx_PULL

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

Таблица 259 – Описание бит регистра PORTx_PULL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	PULL UP[15:0]	Режим работы 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

17.1.6 PORTx_PD

Таблица 260 – Регистр PORTx_PD

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

Таблица 261 – Описание бит регистра PORTx_PD

Разряды	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

17.1.7 PORTx_PWR

Таблица 262 – Регистр PORTx_PWR

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

Таблица 263 – Описание бит регистра PORTx_PWR

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

17.1.8 PORTx_GFEN

Таблица 264 – Регистр PORTx_GFEN

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

Таблица 265 – Описание бит регистра PORTx_GFEN

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

17.1.9 PORTx_SETTX

Таблица 266 – Регистр PORTx_SETTX

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

Таблица 267 – Описание бит регистра PORTx_SETTX

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

17.1.10 PORTx_CLRTX

Таблица 268 – Регистр PORTx_CLRTX

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

Таблица 269 – Описание бит регистра PORTx_CLRTX

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15...0	CLRTX[15:0]	Регистр индивидуального сброса выхода порта Запись «1» в соответствующий разряд регистра сбрасывает в «0» соответствующий разряд выхода порта PORTx_RXTX. Запись нуля не влияет на состояние соответствующего выхода порта PORTx_RXTX. Читается ранее записанным в регистр значениями, а не состояние соответствующих входов порта PORTx_RXTX

18 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 последующей программной обработки.

Уровень опорного напряжения для сравнения с U_{CC} задается битами PLS[2:0] в регистре PVDCS, для сравнения с BU_{CC} задается битами PLBS[1:0] в регистре PVDCS. В соответствии с уровнями напряжения формируются флаги PVD и PBVD. Данные флаги выставляются при возникновении события и сбрасываются программно.

Таблица 270 – Значение параметров детектора напряжения питания

Параметр	Не менее	Типовое	Не более
Входное напряжение U_{CC} , В	3,0	-	3,6
Входное напряжение BU_{CC} , В	1,8	-	3,6
Уровень срабатывания PVD от U_{CC} , при PLS = «000», В		2,0	
Уровень срабатывания PVD от U_{CC} , при PLS = «001», В		2,2	
Уровень срабатывания PVD от U_{CC} , при PLS = «010», В		2,4	
Уровень срабатывания PVD от U_{CC} , при PLS = «011», В		2,6	
Уровень срабатывания PVD от U_{CC} , при PLS = «100», В		2,8	
Уровень срабатывания PVD от U_{CC} , при PLS = «101», В		3,0	
Уровень срабатывания PVD от U_{CC} , при PLS = «110», В		3,2	
Уровень срабатывания PVD от U_{CC} , при PLS = «111», В		3,4	
Уровень срабатывания PBVD от BU_{CC} , при PBLBS = «00», В		1,8	
Уровень срабатывания PBVD от BU_{CC} , при PBLBS = «01», В		2,2	
Уровень срабатывания PBVD от BU_{CC} , при PBLBS = «10», В		2,6	
Уровень срабатывания PBVD от BU_{CC} , при PBLBS = «11», В		3,0	

18.1 Описание регистров блока PVD

Таблица 271 – Описание регистров блока PVD

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

18.1.1 PVDCS

Таблица 272 – Регистр PVDCS

Номер	31...13	12	11	10
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-	PVDBEN	INV	INVB

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

Таблица 273 – Описание бит регистра PVDCS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...13	-	Зарезервировано
12	PVDBEN	Бит разрешения работы блока датчика напряжения питания ВU _{CC} : 0 – датчик отключен; 1 – датчик включен
11	INV	Флаг инверсии выхода от датчика PVD: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то он выставляется при превышении заданного уровня. Если инвертируется - то при снижении ниже заданного уровня
10	INVB	Флаг инверсии выхода от датчика PVBD: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то он выставляется при превышении заданного уровня. Если инвертируется - то при снижении ниже заданного уровня
9	IEPVD	Флаг разрешения прерывания от датчика PVD: 0 – прерывание запрещено; 1 – прерывание разрешено. Очищается записью 0. Если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен
8	IEPVBD	Флаг разрешения прерывания от датчика PVBD: 0 – прерывание запрещено; 1 – прерывание разрешено. Очищается записью 0. Если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен
7	PVD	Результат сравнения напряжения основного питания: 0 – напряжение питания меньше, чем уровень, задаваемый PLS; 1 – напряжение питания больше, чем уровень, задаваемый PLS. Примечание – Сброс флага необходимо проводить с подтверждением – сбрасывать дважды
6	PVBD	Результат сравнения напряжения батарейного питания: 0 – напряжение питания меньше, чем уровень, задаваемый PBLS; 1 – напряжение питания больше, чем уровень, задаваемый PBLS. Примечание – Сброс флага необходимо проводить с подтверждением – сбрасывать дважды

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5...3	PLS[2:0]	Уровень напряжения для сравнения с напряжением основного питания: 000 – 2,0 В; 001 – 2,2 В; 010 – 2,4 В; 011 – 2,6 В; 100 – 2,8 В; 101 – 3,0 В; 110 – 3,2 В; 111 – 3,4 В
2...1	PBLS[1:0]	Уровень напряжения для сравнения с напряжением батарейного питания: 00 – 1,8 В; 01 – 2,2 В; 10 – 2,6 В; 11 – 3,0 В
0	PVDEN	Бит разрешения работы блока датчика напряжения питания U _{CC} : 0 – датчик отключен; 1 – датчик включен

19 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 и запросы DMA по заданным событиям;
- захват входных сигналов, в том числе измерение длительности импульсов входных сигналов;
- 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 другие таймеры.

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

Каждый из таймеров позволяет генерировать прерывания и запросы DMA.

19.1 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 других таймеров;
- до четырех независимых каналов, каждый канал может работать в одном из режимов:
 - режим захвата: позволяет захватить (сохранить) текущее значение счетчика при изменении некоторого входного сигнала;
 - режим ШИМ: позволяет осуществлять непрерывное сравнение заданных значений со значением счетчика для формирования выходных сигналов;
- формирование выходных сигналов в режиме ШИМ:
 - сброс в НИЗКИЙ уровень при совпадении;
 - установка в ВЫСОКИЙ уровень при совпадении;
 - переключение (инвертирование) при совпадении;
 - переключение при некотором условии;
- формирование прерываний и запросов DMA по событиям:
 - обновление счетчика;
 - захват;
 - сравнение;
 - внешние события по входам ETR и BRK.

19.2 Структурная схема

Структурная схема блока «Таймер» представлена на рисунке 37.

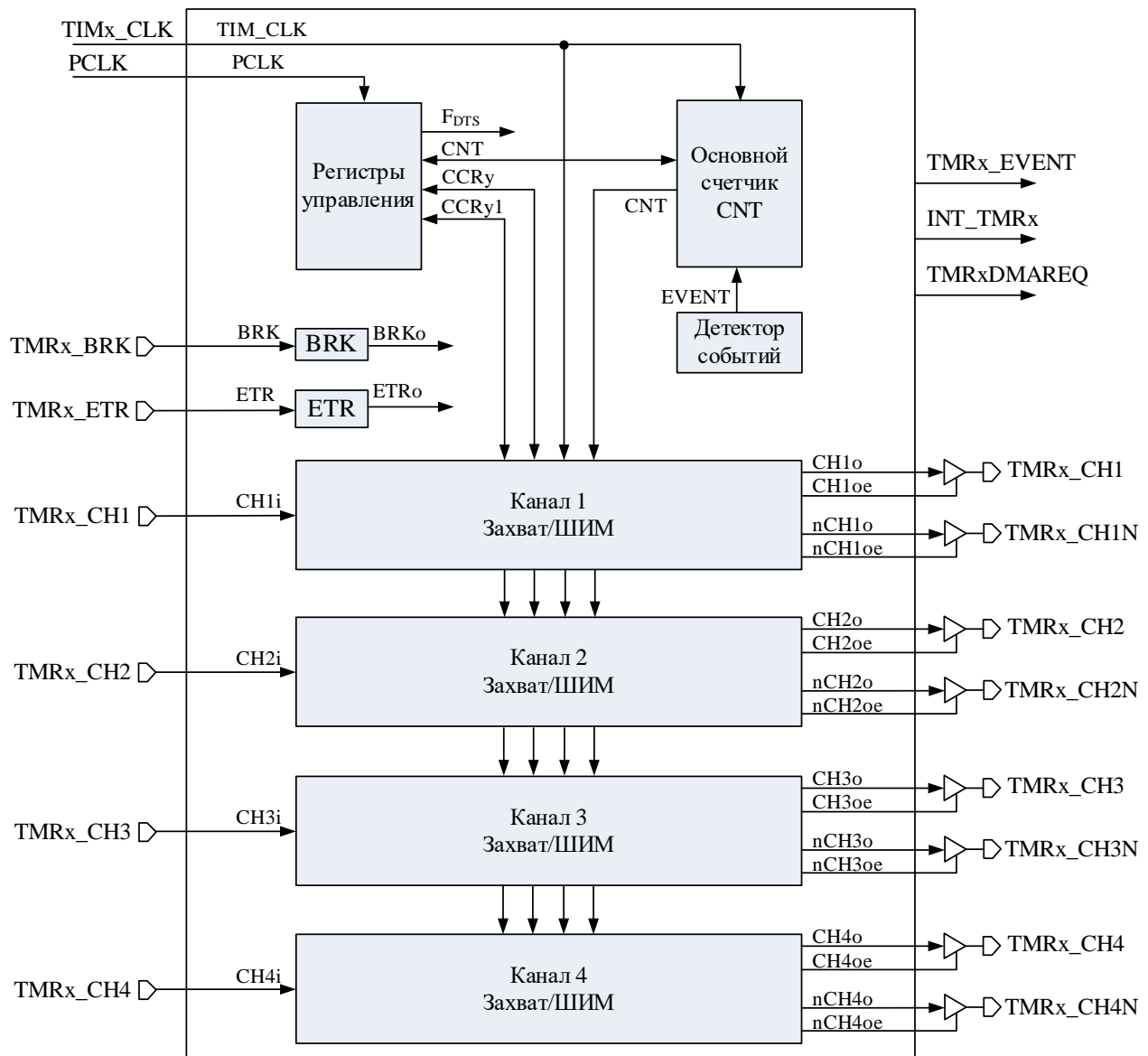


Рисунок 37 – Структурная схема блока «Таймер»

Таймер содержит основной 16-битный счетчик CNT, блок регистров управления и четыре канала схем захвата/ШИМ.

Таймер позволяет работать в режимах:

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

19.3 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 19.4 «Источники событий для счета»).

19.3.1 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 PER1_C2 для формирования частоты TIM_CLK, разрешение подачи частоты TIM_CLK на блок таймера управляется битом TIMxCLKEN. Тактовая частота TIM_CLK должна быть меньше или равна тактовой частоте PCLK (CPU_CLK).

После подачи тактовых сигналов можно приступить к работе с таймером.

19.3.2 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 $TIM_CLK_d = TIM_CLK / (PSG + 1)$;
- разрешить работу счетчика CNT_EN.

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

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

Поле CNT_MODE[1:0] в регистре CNTRL определяет режим работы основного счетчика:

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

19.3.3 Режимы счета

19.3.3.1 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
```

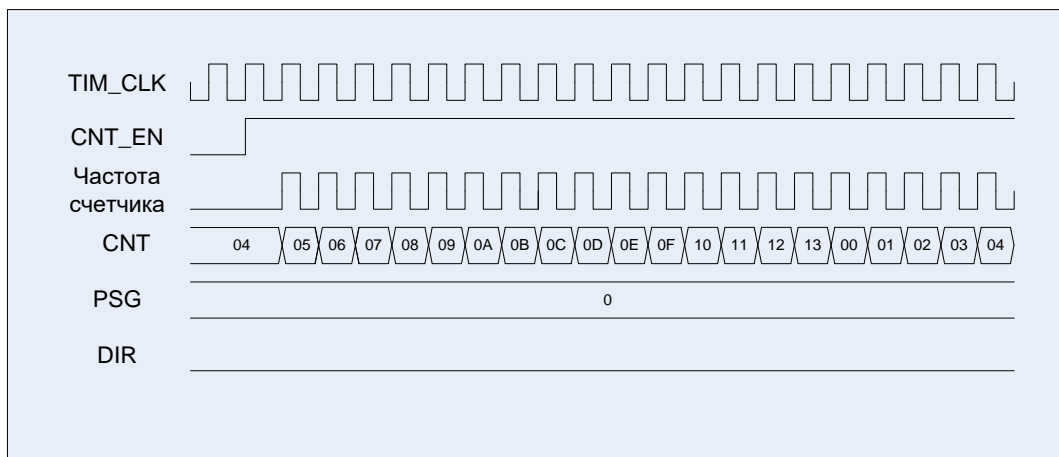


Рисунок 38 – Диаграммы работы таймера, счет прямой от 0 до 0x13, стартовое значение 0x04

19.3.3.2 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

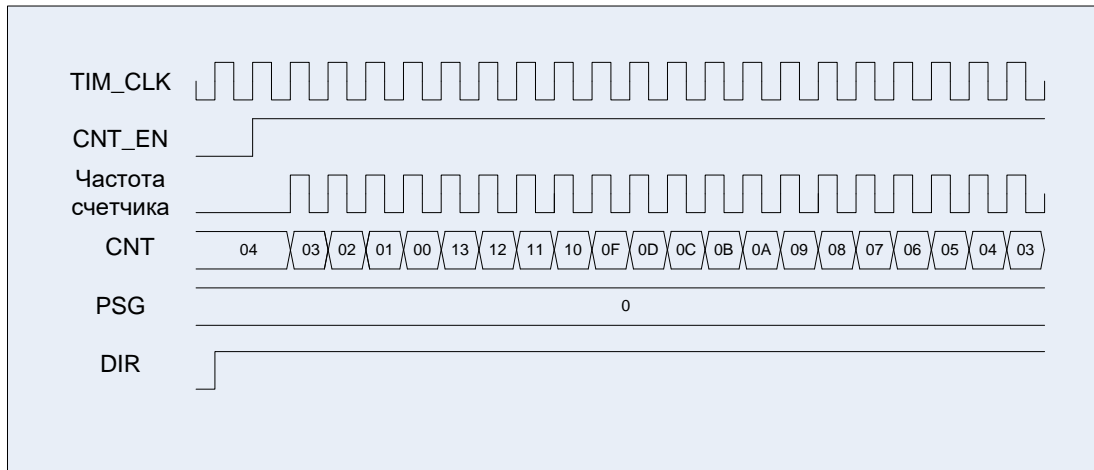


Рисунок 39 – Диаграммы работы таймера, счет обратный от 0x13 до 0, стартовое значение 0x04

19.3.3.3 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

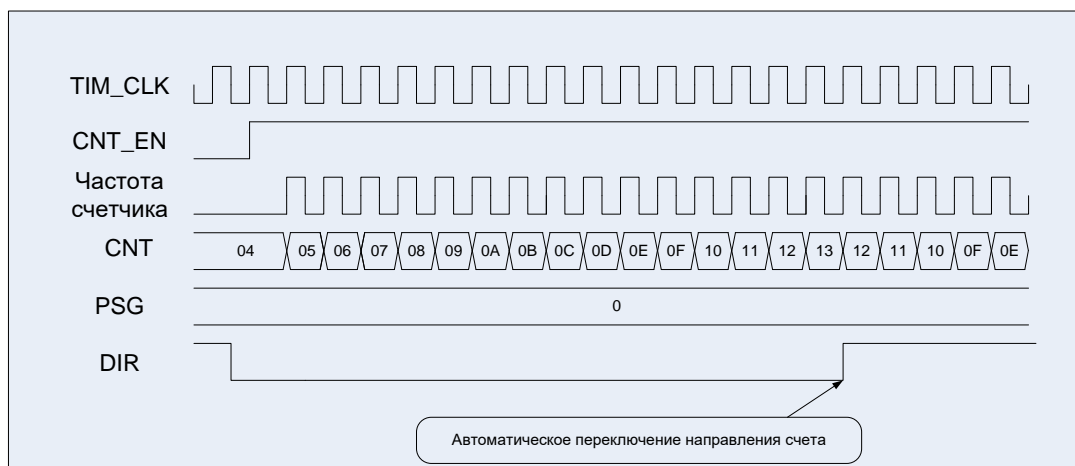


Рисунок 40 – Диаграммы работы таймера, счет двунаправленный, сначала прямой

19.3.3.4 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

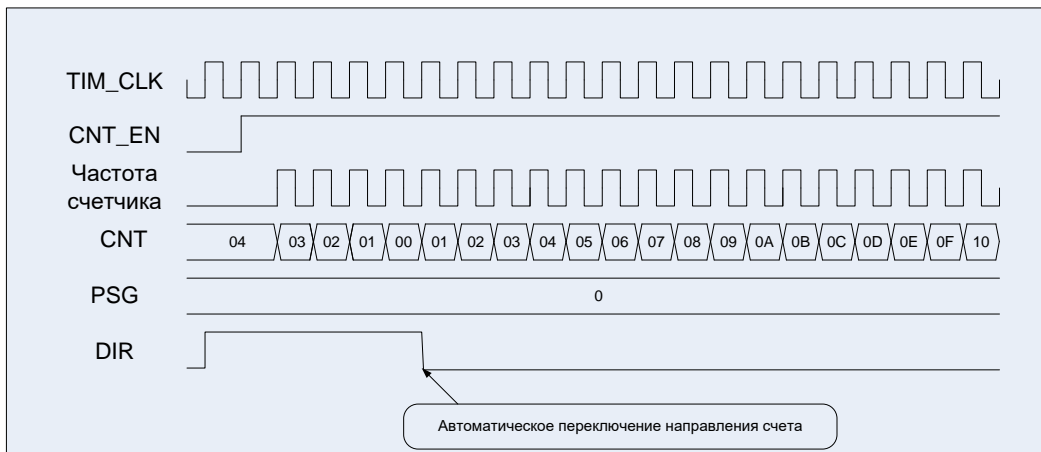


Рисунок 41 – Диаграммы работы таймера, счет двунаправленный, сначала обратный

19.3.4 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_i.

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 F_{DTS}[1:0].

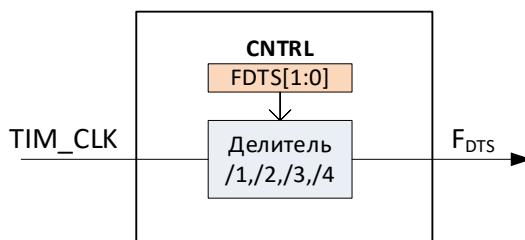


Рисунок 42 – Схема формирования тактовой частоты F_{DTS}

Диаграмма возможных частот F_{DTS} приведена на рисунке 43.

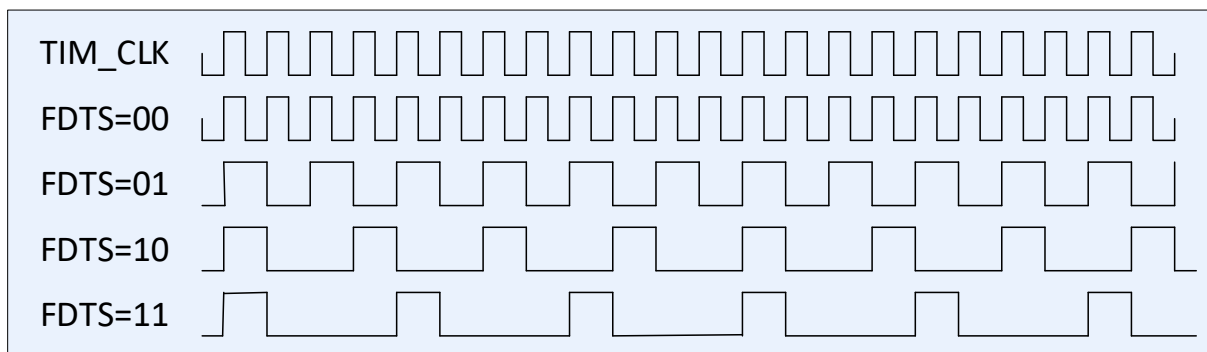


Рисунок 43 – Диаграмма тактовой частоты F_{DTS} в зависимости от значения F_{DTS}[1:0] в регистре CNTRL

19.4 Источники событий для счета

Тактирование основного счетчика таймера может осуществляться от следующих источников:

- внутренний тактовый сигнал (TIM_CLKd);
- событие в другом таймере (CNT==ARR);
- внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CH*u*;
- 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.

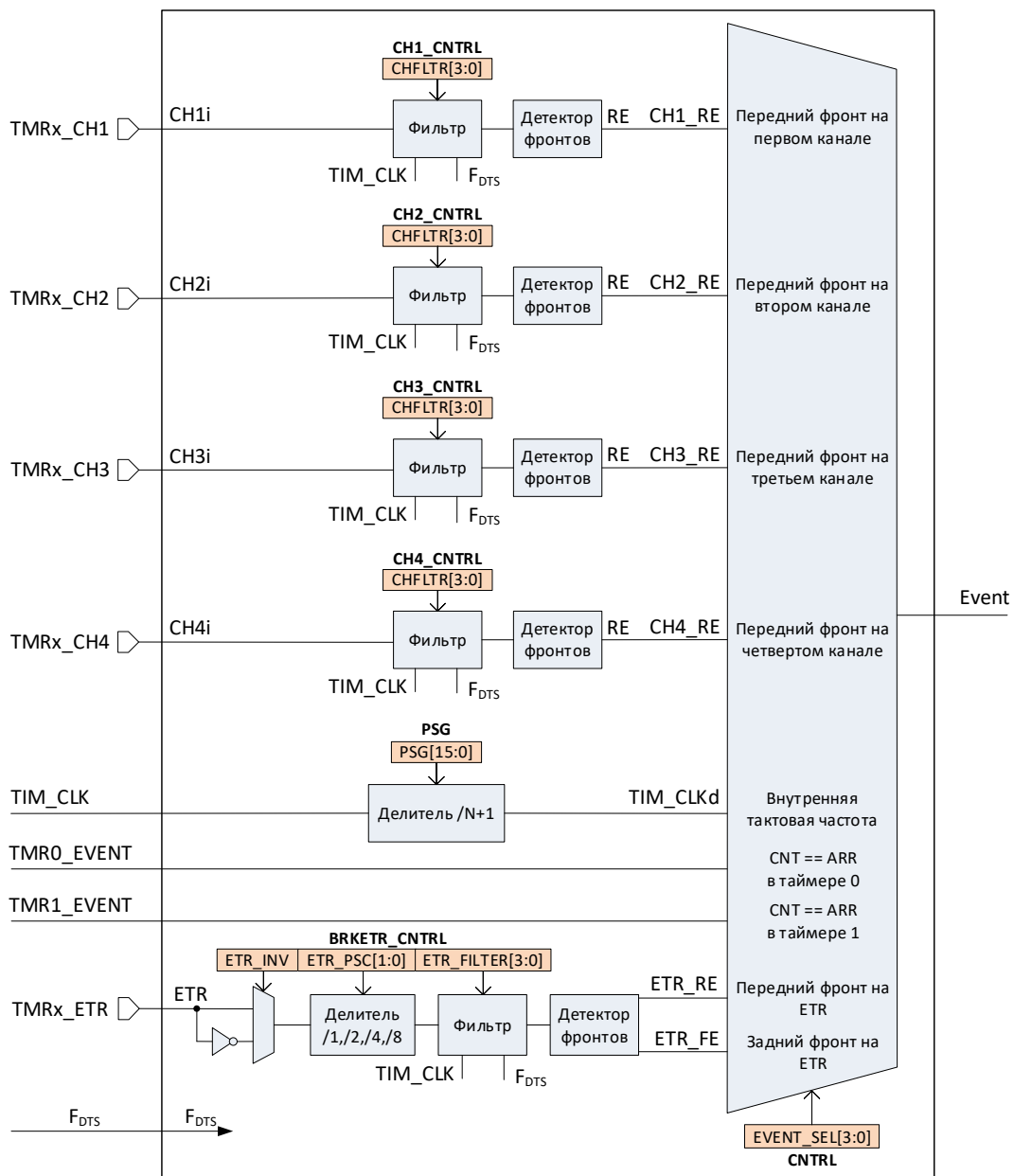


Рисунок 44 – Структурная схема формирования события для счета

19.4.1 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) неравно нулю, то счетный регистр делителя будет инкрементироваться по каждому импульсу сигнала TIM_CLK до тех пор, пока не достигнет значения, находящегося в регистре делителя. Далее счетный регистр делителя сбрасывается в ноль, содержимое основного счетчика таймера изменяется на 1 и счет начинается заново. Таким образом выходная частота предварительного делителя составляет:

$$TIM_CLKd = \frac{TIM_CLK}{PSG + 1} \quad (11)$$

Значение регистра PSG можно изменять даже во время работы счетчика, новое значение предделителя вступит в силу сразу после записи. На рисунках 45 и 46 приведены диаграммы работы счетчика при обновлении значения PSG.

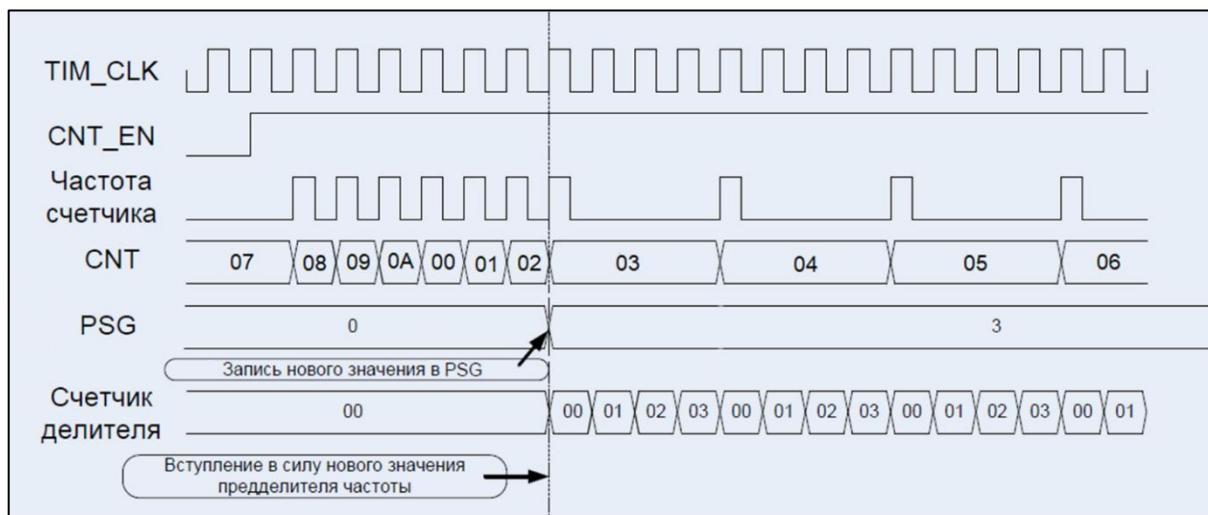


Рисунок 45 – Диаграмма работы счетчика: счет прямой (CNT_MODE[1:0] = 00, EVENT_SEL[3:0] = 0000, DIR = 0)

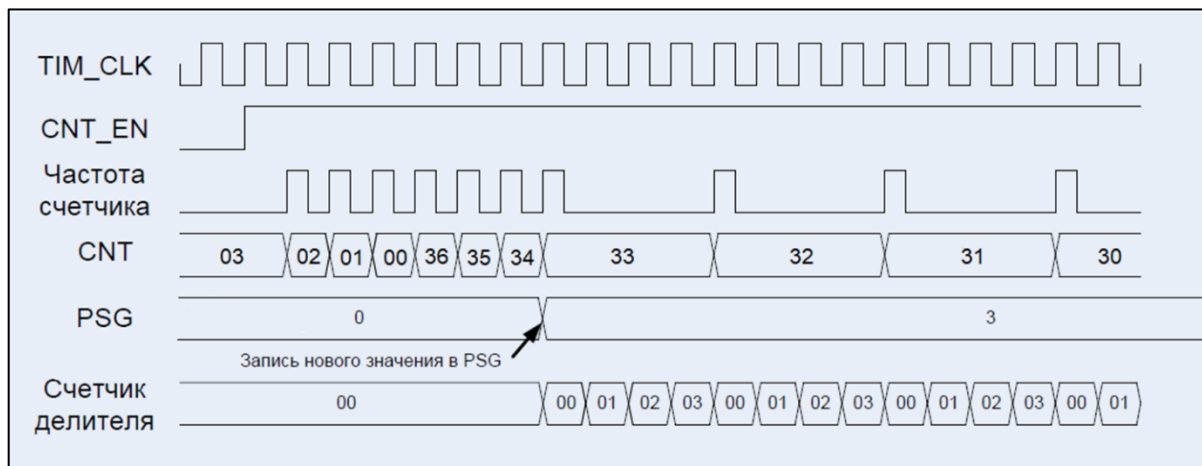


Рисунок 46 – Диаграмма работы счетчика: счет обратный (CNT_MODE[1:0] = 00, EVENT_SEL[3:0] = 0000, DIR = 1)

19.4.2 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 47 показан пример каскадного соединения таймеров, диаграммы работы данных таймеров приведены на рисунке 48.

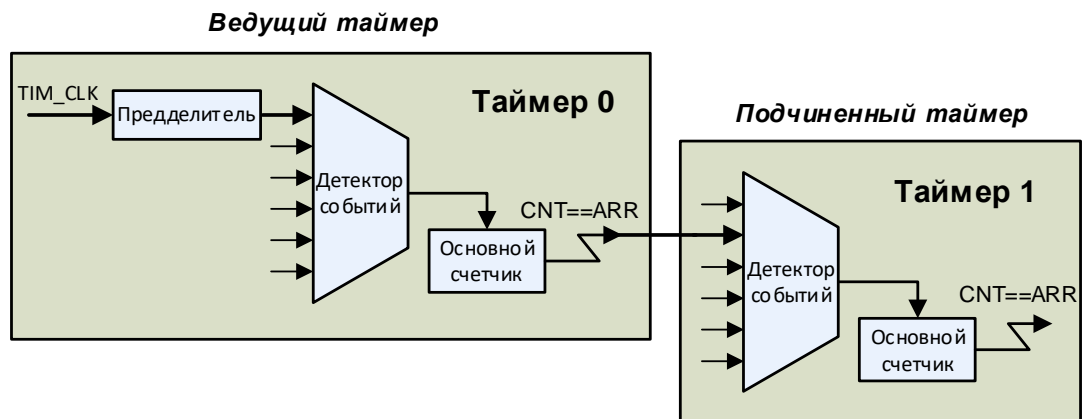


Рисунок 47 – Пример каскадного соединения таймеров

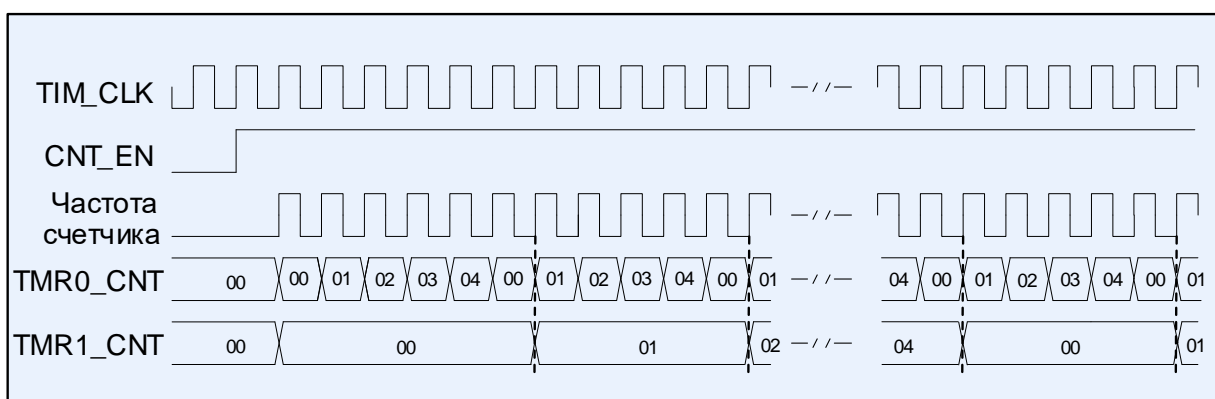


Рисунок 48 – Диаграммы работы двух таймеров в каскаде
 TIMER0: DIR = 0, EVENT_SEL[3:0] = 0000, CNT_MODE[1:0] = 00;
 TIMER1: DIR = 0, EVENT_SEL[3:0] = 0001, CNT_MODE[1:0] = 10

19.4.3 Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CH_уi

Данный режим выбирается, когда EVENT_SEL[3:0] = 01xx и CNT_MODE[1:0] = 10. Основной счетчик таймера считает по переднему фронту внешнего сигнала, поступающего на вход канала CH_уi. Биты CHSEL[1:0] регистра CH_у_CNTRL не оказывают влияния, так как они применяются для работы канала таймера только в режиме захвата.

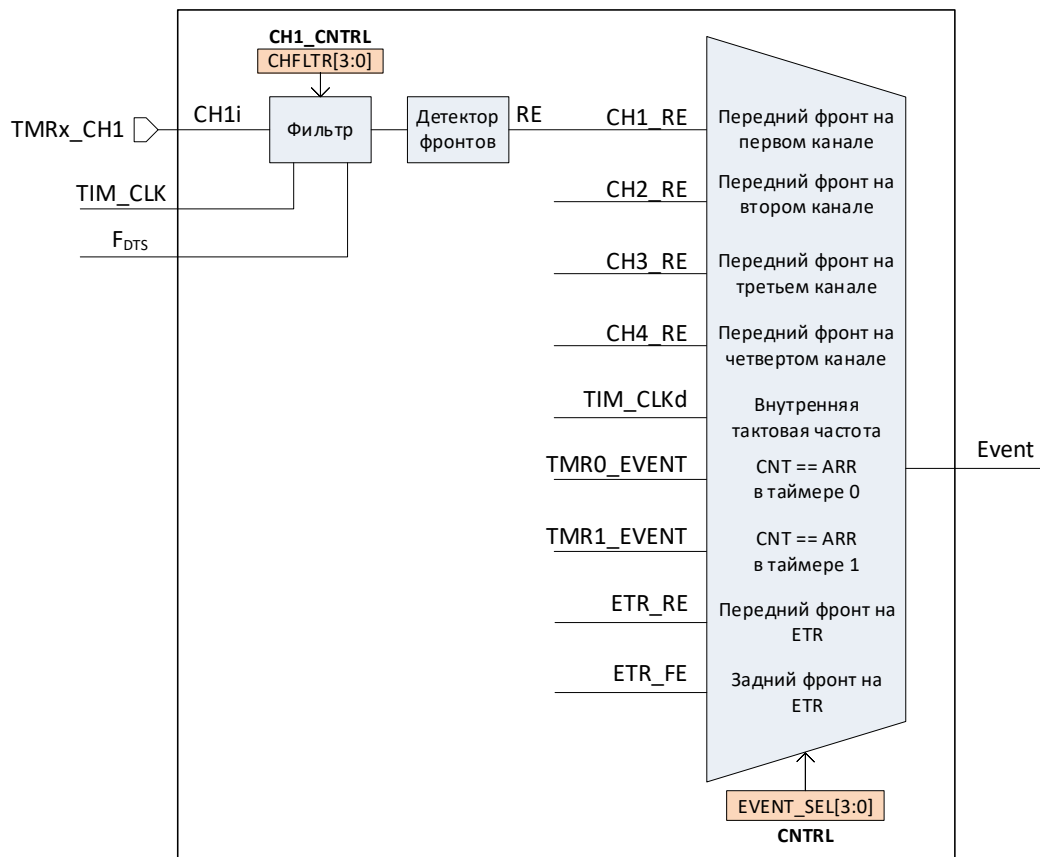


Рисунок 49 – Схема тактирования сигналом со входа первого канала

Со входа CH_уi 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 19.7 «Блок цифрового фильтра»). Настройки фильтра задаются в поле CHFLTR[3:0] регистра CH_у_CNTRL.

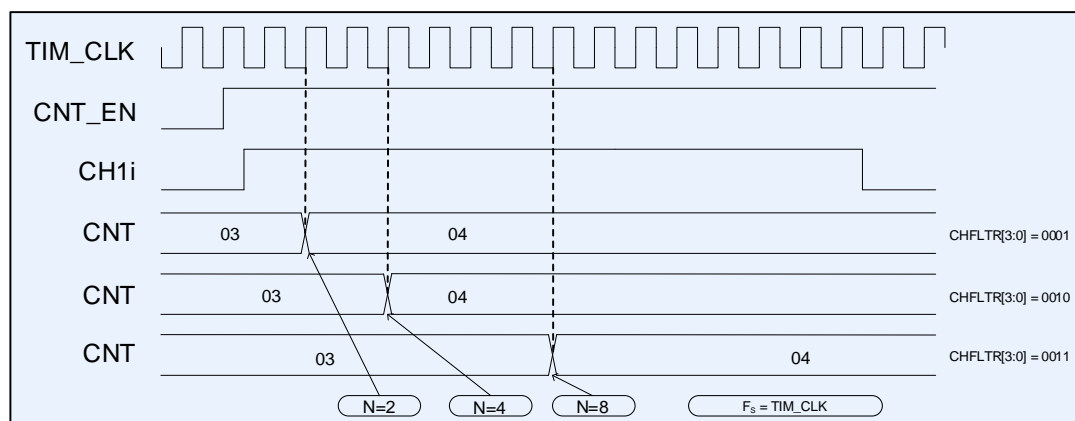


Рисунок 50 – Диаграмма внешнего тактирования с разными вариантами фильтра

19.4.4 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 19.7 «Блок цифрового фильтра»). Настройки фильтра задаются в поле ETR_FILTER[3:0].

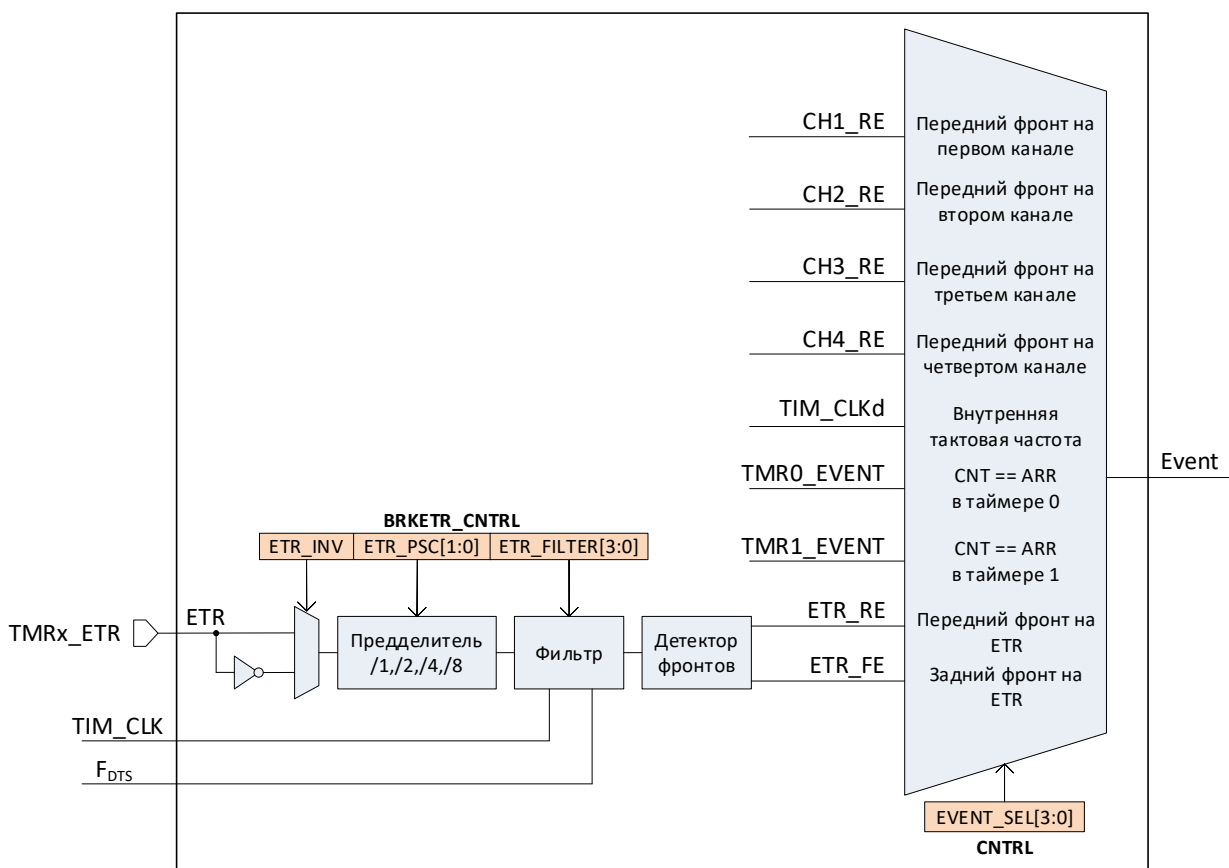


Рисунок 51 – Схема тактирования сигналом со входа ETR

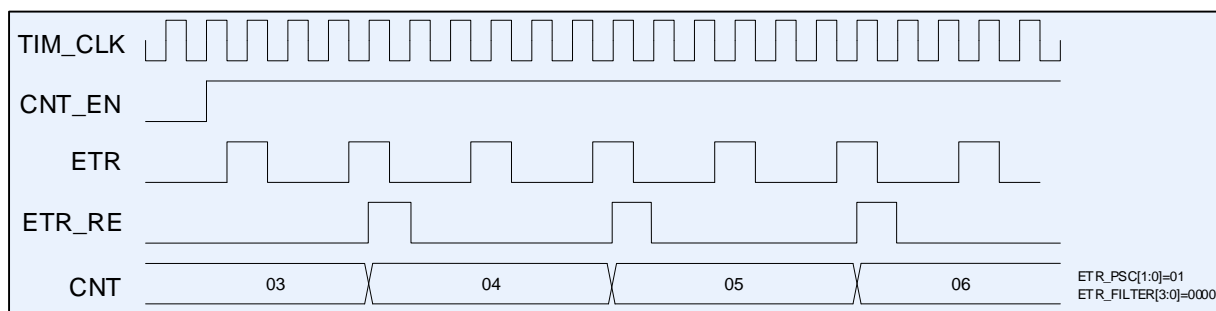


Рисунок 52 – Диаграмма тактирования со входа ETR, $EVENT_SEL[3:0] = 1000$

19.5 Режим захвата

Каждый канал таймера может быть независимо переведен в режим захвата. В режиме захвата по событию от внешнего входного сигнала происходит фиксация значения основного счетчика CNT в регистры CCRy (CCR) и CCRy1 (CCR1). Регистрация событий осуществляется только по входам CHy_i (выводы, настроенные в функцию TMRx_CHy).

Структурная схема блока захвата представлена на рисунке 53.

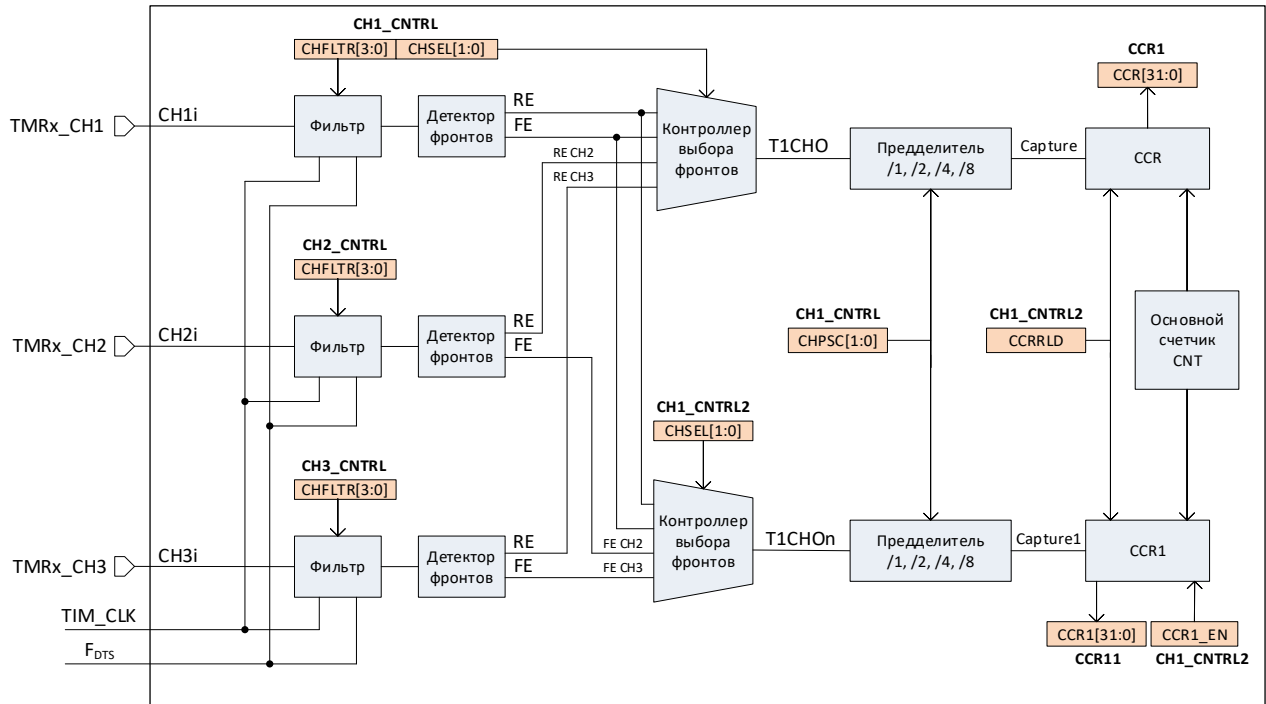


Рисунок 53 – Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо записать «1» в бит CAP_NPWM регистра управления каналом CHy_CNTRL. Для использования регистра CCRy1(CCR1) необходимо записать «1» в бит CCR1_EN регистра CHy_CNTRL2.

Внешний сигнал со входа CHy_i сначала поступает в блок фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. подраздел 19.7 «Блок цифрового фильтра»). Настройки фильтра задаются в поле CHFLTR[3:0] регистра CHy_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 фронтом канала, отрицательным фронтом канала и положительными и отрицательными фронтами сигналов от других каналов. Настройка блока «Контроллер выбора фронтов» для регистра CCRy осуществляется в поле CHSEL[1:0] регистра CHy_CNTRL, а для регистра CCRy1 – в поле CHSEL[1:0] регистра CHy_CNTRL2. Выбранный для захвата сигнал поступает в предварительный делитель, который в зависимости от значения в поле CHPSC[1:0] регистра CHy_CNTRL

позволяет фиксировать все события, либо каждое второе, каждое четвертое или каждое восьмое событие.

Предварительный делитель для регистра CCRy формирует сигнал Capture, а предварительный делитель для регистра CCRy1 формирует сигнал Capture1. По сигналам Capture и Capture1 выполняется запись текущего значения основного счетчика CNT в регистры CCRy и CCRy1.

На рисунке 54 показан пример захвата значения основного счетчика CNT в регистр CCRy по положительному фронту на входе канала, а в регистр CCRy1 – по отрицательному фронту на входе канала.

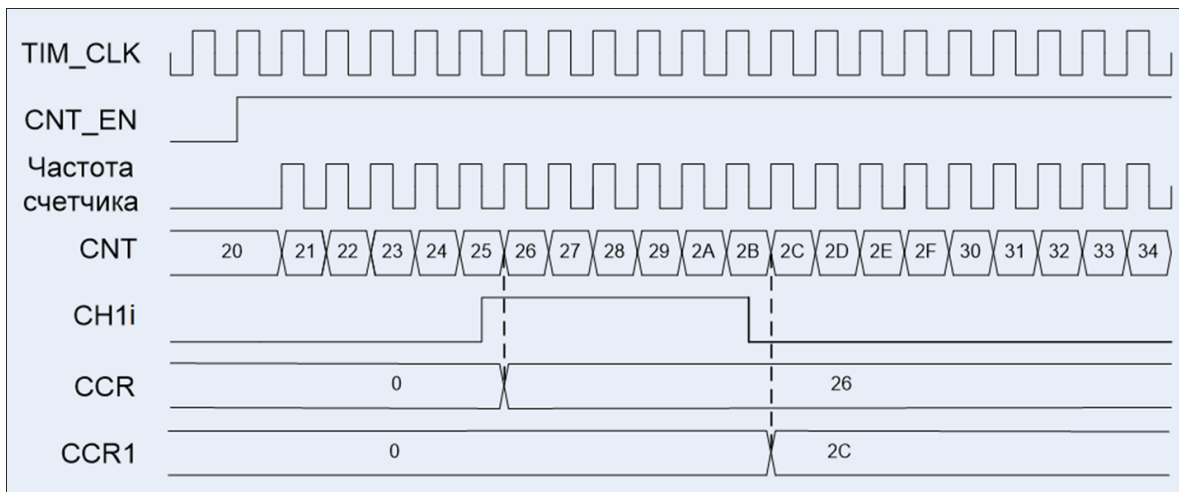


Рисунок 54 – Диаграмма захвата события со входа канала 1

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

19.6 Режим ШИМ

Каждый канал таймера может быть независимо переведен в режим ШИМ для формирования выходных сигналов с возможностью задания «мертвой зоны». Структурная схема блока формирования ШИМ представлена на рисунке 55.

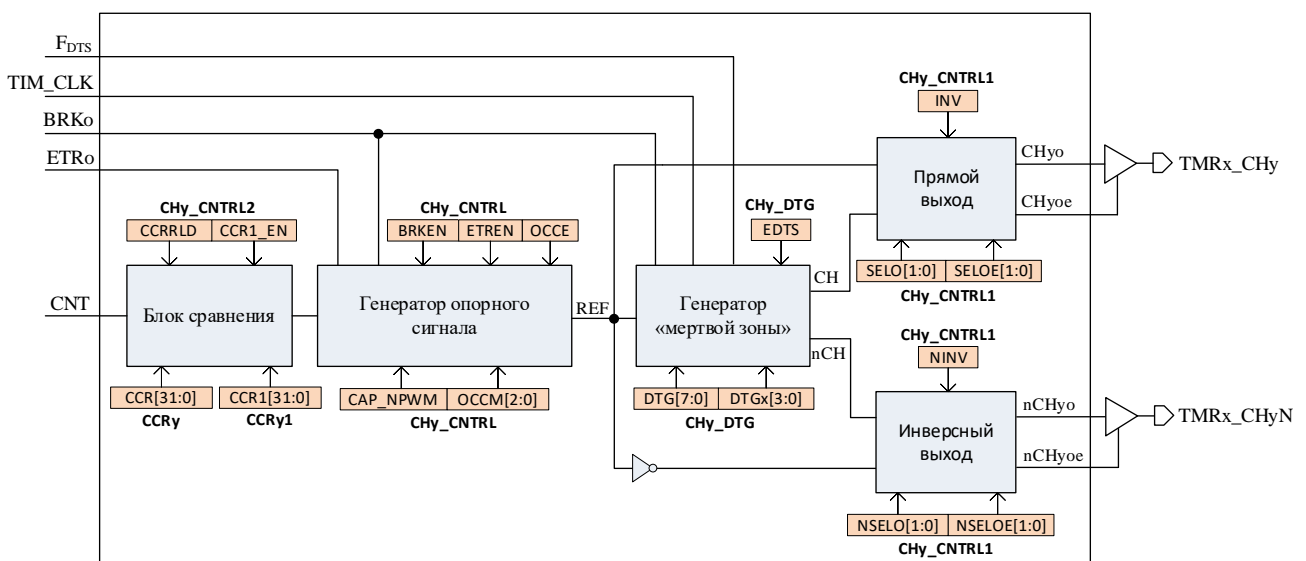


Рисунок 55 – Структурная схема блока формирования ШИМ

Для включения режима ШИМ для определенного канала необходимо в регистре управления каналом `CHy_CNTRL` записать «0» в бит `CAP_NPWM`.

19.6.1 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 `CCRu` (`CCR`), `CCRu1` (`CCR1`) и основного счетчика `CNT`. Формат выработки сигнала REF устанавливается в поле `OCCM[2:0]` регистра управления каналом таймера `CHy_CNTRL`.

Если в регистре `CHy_CNTRL2` бит `CCR1_EN` = 0, то для формирования сигнала REF используется только результат сравнения значения в регистре `CCRu` и основного счетчика `CNT`.

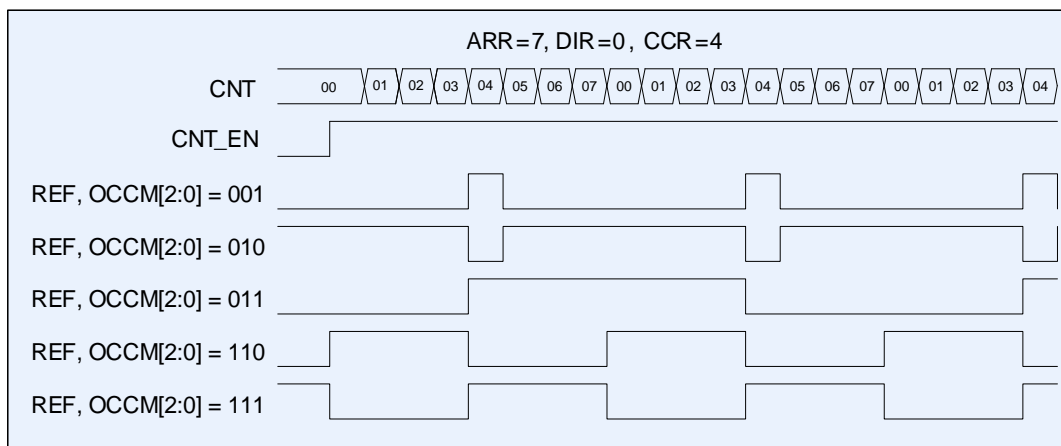


Рисунок 56 – Диаграмма работы в режиме ШИМ, `CCR1_EN`=0

Если в регистре `CHy_CNTRL2` бит `CCR1_EN` = 1, то для формирования сигнала REF задействуются оба результата сравнения значения в регистрах `CCRu`, `CCRu1` и основного счетчика `CNT`.

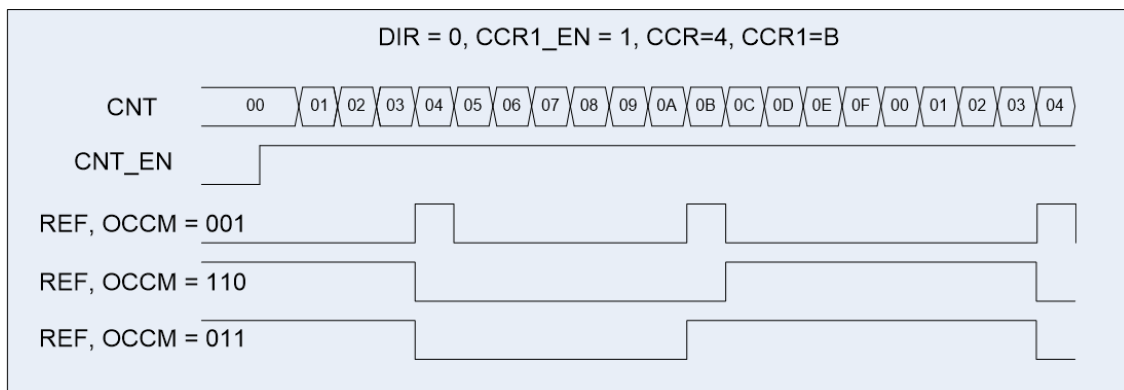


Рисунок 57 – Диаграмма работы в режиме ШИМ, `CCR1_EN` = 1

Запись новых значений в регистры `CCRu` и `CCRu1` осуществляется немедленно, если в регистре `CHy_CNTRL2` бит `CCRRLD` установлен в «0», иначе регистры `CCRu` и `CCRu1` получают новые значения только при `CNT` == 0. Процесс обновления значений в регистрах `CCRu` и `CCRu1` обозначается в регистре `CHy_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 СНу_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 58.

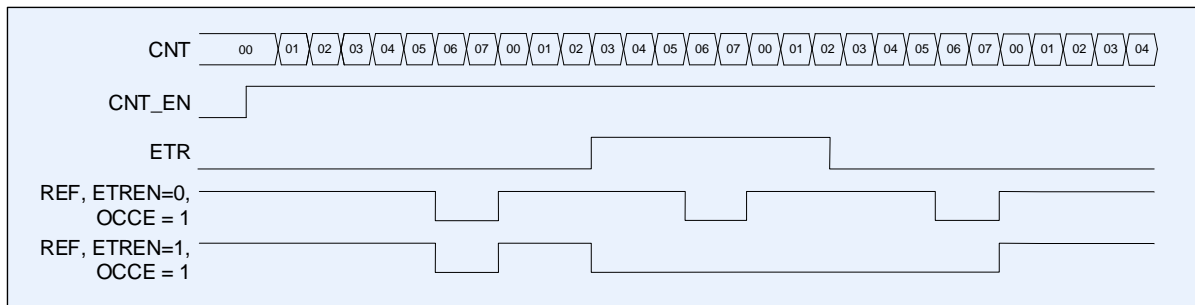


Рисунок 58 – Диаграмма сброса сигнала REF по выводу ETR

Для разрешения сброса сигнала REF по входу BRK необходимо установить бит BRKEN в регистре СНу_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 59.

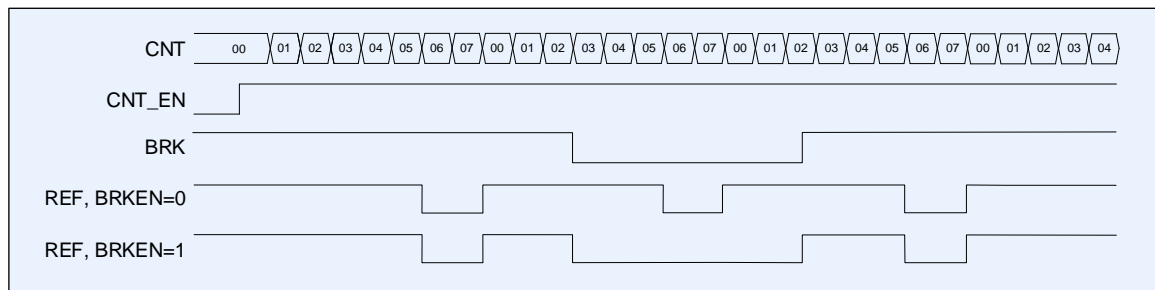


Рисунок 59 – Диаграмма сброса сигнала REF по выводу BRK

19.6.2 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 (СН_{у0}, СН_{у0e}) представляет собой инвертированный сигнал REF, в котором передний фронт задержан на величину DTGdel относительно заднего фронта опорного сигнала REF;

- сигнал на инверсном выходе (nСН_{у0}, nСН_{у0e}) представляет собой сигнал REF, в котором передний фронт задержан на величину DTGdel относительно переднего фронта опорного сигнала REF.

Значение «мертвой зоны» между сигналами на прямом и инверсном выходах рассчитывается в тактах частоты TIM_CLK или F_{DTs} по формуле

$$DTG_{del} = DTG \cdot (DTG_x + 1), \tag{12}$$

где DTG_x – предварительный делитель частоты;

DTG – основной делитель частоты.

Управление блоком DTG осуществляется через регистр CH_u_DTG . Выбор источника тактирования для задания «мертвой зоны» задается битом $EDTS$. Значения делителей DTG_x и DTG задаются в полях $DTG_x[3:0]$ и $DTG[7:0]$, соответственно. Если задержка DTG_{del} больше ширины импульса высокого уровня, то соответствующий импульс не генерируется.

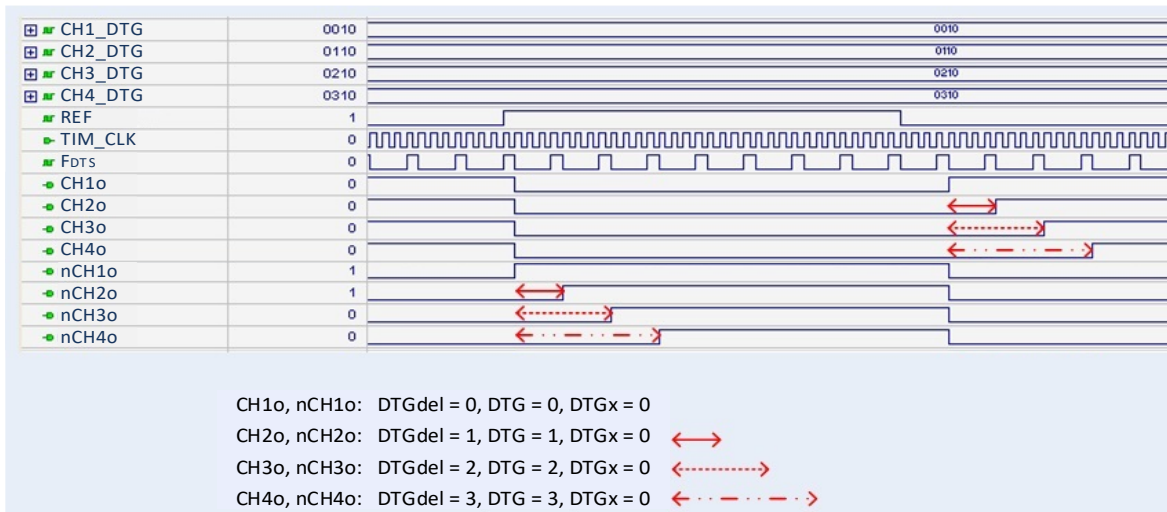


Рисунок 60 – Диаграмма работы блока DTG

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 CH_u_CNTRL . Активный уровень на входе BRK сбрасывает выходные сигналы блока DTG в «0» путем их маскирования.

19.6.3 Выходные блоки

Каждый канал таймера имеет два выходных блока – прямой и инверсный. Каждый выходной блок формирует как сигнал выдачи (CH_{uo} , nCH_{uo}), так и сигнал разрешения выдачи (CH_{uoe} , nCH_{uoe}). В качестве сигналов для прямого (CH_{uo} , CH_{uoe}) и инверсного (nCH_{uo} , nCH_{uoe}) выходов в регистре CH_u_CNTRL1 могут быть заданы либо постоянные уровни (0 или 1), либо сигналы, формируемые на основе сигнала REF . К таким сигналам относится сам сигнал REF , а также сигналы, формируемые блоком DTG .

Выбор источника сигнала выдачи для прямого (CH_{uo}) и инверсного (nCH_{uo}) выходов задается в полях $SELO[1:0]$ и $NSELO[1:0]$ регистра CH_u_CNTRL1 . Дополнительно каждый сигнал выдачи для прямого (CH_{uo}) и инверсного (nCH_{uo}) выходов может быть инвертирован путем установки битов INV и $NINV$ в регистре CH_u_CNTRL1 . Выбор источника сигнала разрешения выдачи для прямого (CH_{uoe}) и инверсного (nCH_{uoe}) выходов задается в полях $SELOE[1:0]$ и

NSELOE[1:0] регистра CHy_CNTRL1. При этом, если сигнал разрешения выдачи равен «0», то соответствующий вывод работает в режиме входа, если сигнал разрешения выдачи равен «1» – то в режиме выхода.

19.7 Блок цифрового фильтра

В тракте входа ETR и входов каналов таймера CHy_i 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 входов каналов CHy_i – в поле CHFLTR[3:0] регистра CHy_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 61, частота $F_S = TIM_CLK$, количество выборок $N = 4$.

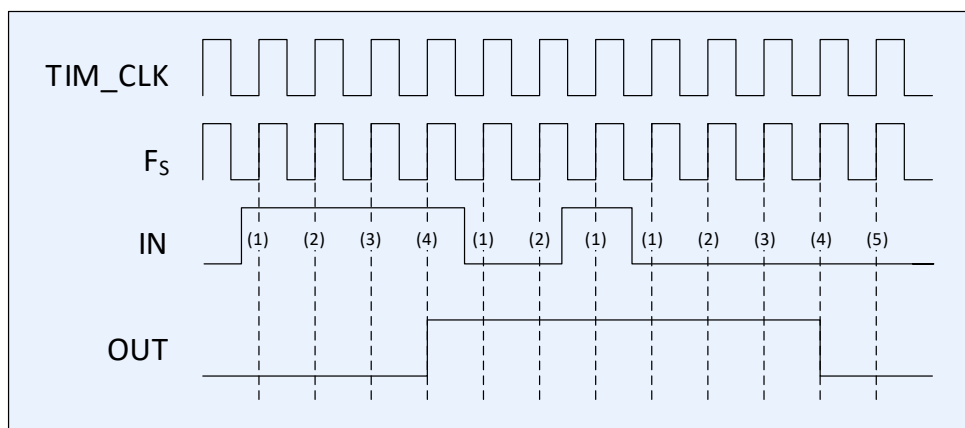


Рисунок 61 – Диаграмма работы фильтра, $F_S = TIM_CLK$, $N = 4$

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 19.3.4 «Тактовая частота F_{DTS} »).

Диаграмма работы фильтра при использовании частоты F_{DTS} приведена на рисунке 62, частота выборки $F_S = F_{DTS}/2$, количество выборок $N = 6$.

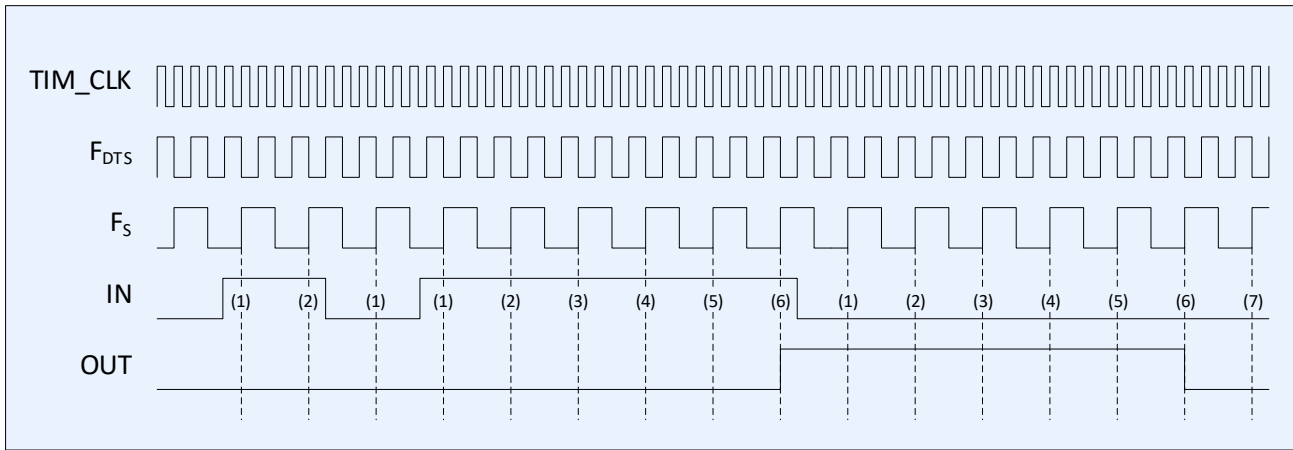


Рисунок 62 – Диаграмма работы фильтра, $F_S = F_{DTS}/2$, $N = 6$

Возможные варианты настройки блока фильтра приведены в таблице 274.

Таблица 274 – Возможные конфигурации фильтра

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

19.8 Флаги состояний, прерываний и запросов DMA

В процессе работы блок таймера отслеживает состояние внутренних блоков и формирует 17 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[3:0] – запись значения счетчика CNT в регистр CCRy по захвату настроенного фронта на входе канала CH_{yi}, события формируются индивидуально для каждого канала;
- CCR REF EVENT[3:0] – фиксация переднего фронта на выходе генератора опорного сигнала REF, события формируются индивидуально для каждого канала;
- CCR CAP1 EVENT[3:0] – запись значения счетчика CNT в регистр CCRy1 по захвату настроенного фронта на входе канала CH_{yi}, события формируются индивидуально для каждого канала.

19.8.1 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.

19.8.2 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.

19.8.3 Запросы DMA

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

19.9 Примеры

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

19.9.1 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 = PER1_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
```

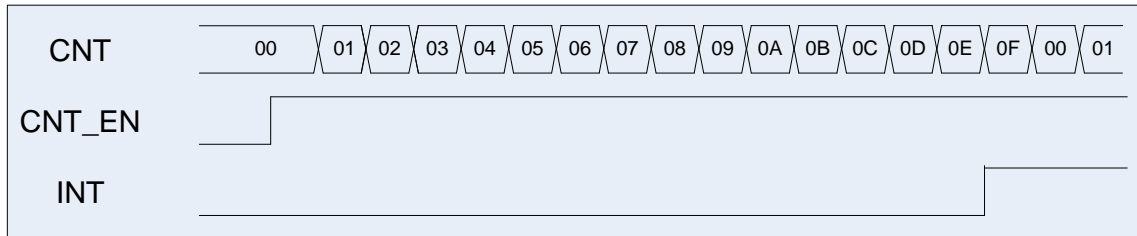


Рисунок 63 – Режим обычного счетчика

19.9.2 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 = PER1_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 = 0x000001E0; //Разрешение генерировать прерывание по событию
//настроенного фронта на входах CH1i-CH4i

TIMER0->CH1_CNTRL = 0x00008000; //Захват по положительному фронту сигнала
//на входе CH1i, фильтрация отключена
TIMER0->CH2_CNTRL = 0x00008010; //Захват по отрицательному фронту сигнала
//на входе CH2i, фильтрация отключена
TIMER0->CH3_CNTRL = 0x00008001; //Захват по положительному фронту сигнала
//на входе CH3i, фильтрация выполняется
//по двум выборкам на частоте TIM_CLK
TIMER0->CH4_CNTRL = 0x00008011; //Захват по отрицательному фронту сигнала
//на входе CH4i, фильтрация выполняется
//по двум выборкам на частоте TIM_CLK

//Режим работы выхода канала – канал работает на вход
TIMER0->CH1_CNTRL1 = 0x00000000;
TIMER0->CH2_CNTRL1 = 0x00000000;
TIMER0->CH3_CNTRL1 = 0x00000000;
TIMER0->CH4_CNTRL1 = 0x00000000;
//Разрешение работы таймера
TIMER0->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
    
```

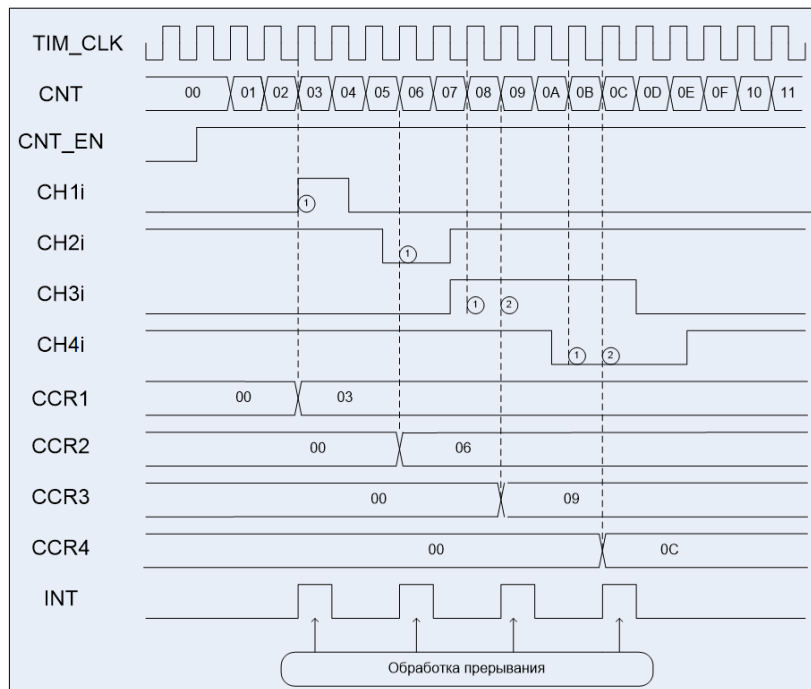


Рисунок 64 – Диаграммы примера работы в режиме захвата

19.9.3 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 = PER1_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 = 0x00001E00; //Разрешение генерировать прерывание по событию
//переднего фронта на выходе REF для всех каналов

//Режим работы каналов – ШИМ
TIMER0->CH1_CNTRL = 0x00000200; //REF = 1, если CNT == CCR
TIMER0->CH2_CNTRL = 0x00000200; //REF = 1, если CNT == CCR
TIMER0->CH3_CNTRL = 0x00000400; //REF = 0, если CNT == CCR
TIMER0->CH4_CNTRL = 0x00000600; //Переключение REF, если CNT == CCR
//Режим работы выхода канала – канал работает на выход,
//на выходы канала выдается сигнал REF
TIMER0->CH1_CNTRL1 = 0x00000909;
TIMER0->CH2_CNTRL1 = 0x00000909;
TIMER0->CH3_CNTRL1 = 0x00000909;
TIMER0->CH4_CNTRL1 = 0x00000909;
//Установка значений CCR, с которыми сравнивается CNT при работе в режиме ШИМ
TIMER0->CCR1 = 0x00000003;
TIMER0->CCR2 = 0x00000006;
TIMER0->CCR3 = 0x00000009;
TIMER0->CCR4 = 0x0000000F;
//Разрешение работы таймера
TIMER0->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
    
```

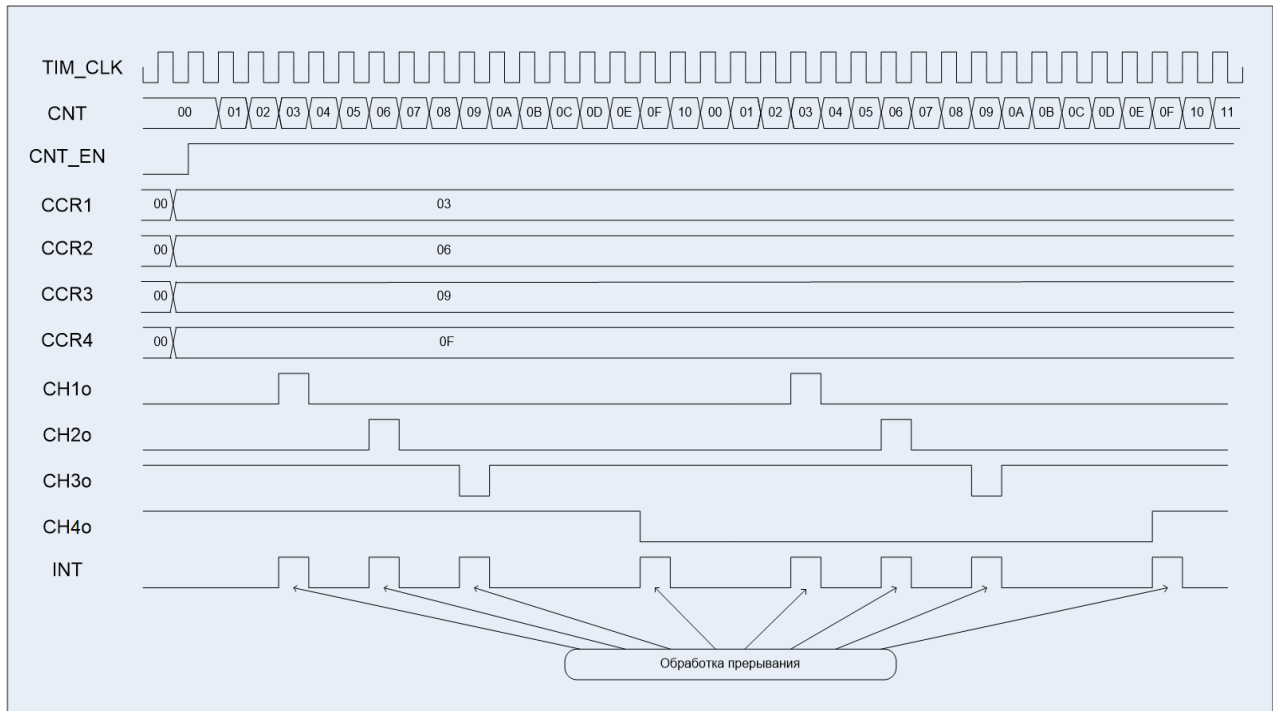


Рисунок 65 – Диаграммы примера работы в режиме ШИМ

19.10 Описание регистров блока таймера

Таблица 275 – Описания регистров блока 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 прерывания
0x5C	DMA_RE[31:0]	Регистр разрешения формирования запроса DMA
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
Канал 2		
0x14	CCR2[15:0]	Регистр сравнения/захвата для 2 канала таймера
0x24	CH2_CNTRL[31:0]	Регистр управления для 2 канала таймера

Адрес	Название	Описание
0x34	CH2_CNTRL1[31:0]	Регистр управления 1 для 2 канала таймера
0x44	CH2_DTG[31:0]	Регистр управления DTG для 2 канала таймера
0x64	CH2_CNTRL2[31:0]	Регистр управления 2 для 2 канала таймера
0x74	CCR21[15:0]	Регистр сравнения/захвата 1 для 2 канала таймера
Канал 3		
0x18	CCR3[15:0]	Регистр сравнения/захвата для 3 канала таймера
0x28	CH3_CNTRL[31:0]	Регистр управления для 3 канала таймера
0x38	CH3_CNTRL1[31:0]	Регистр управления 1 для 3 канала таймера
0x48	CH3_DTG[31:0]	Регистр управления DTG для 3 канала таймера
0x68	CH3_CNTRL2[31:0]	Регистр управления 2 для 3 канала таймера
0x78	CCR31[15:0]	Регистр сравнения/захвата 1 для 3 канала таймера
Канал 4		
0x1C	CCR4[15:0]	Регистр сравнения/захвата для 4 канала таймера
0x2C	CH4_CNTRL[31:0]	Регистр управления для 4 канала таймера
0x3C	CH4_CNTRL1[31:0]	Регистр управления 1 для 4 канала таймера
0x4C	CH4_DTG[31:0]	Регистр управления DTG для 4 канала таймера
0x6C	CH4_CNTRL2[31:0]	Регистр управления 2 для 4 канала таймера
0x7C	CCR41[15:0]	Регистр сравнения/захвата 1 для 4 канала таймера

19.10.1 CNT

Таблица 276 – Основной счетчик таймера CNT

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

Таблица 277 – 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

19.10.2 PSG

Таблица 278 – Делитель частоты TIM_CLK для счета основного счетчика PSG

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

Таблица 279 – 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)

19.10.3 ARR

Таблица 280 – Основание счета основного счетчика ARR

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

Таблица 281 – 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]

19.10.4 CNTRL

Таблица 282 – 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...4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	00	00	0	0	0	0
	CNT_MODE[1:0]	FDTS[1:0]	DIR	WR_CMPL	ARRB_EN	CNT_EN

Таблица 283 – 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 – событие переднего фронта на канале 2, «Режим 1»; 0110 – событие переднего фронта на канале 3, «Режим 1»; 0111 – событие переднего фронта на канале 4, «Режим 1»; 1000 – событие переднего фронта на ETR, «Режим 2»; 1001 – событие заднего фронта на ETR, «Режим 2»; 1010-1111 – зарезервировано

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
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
5..4	F_DTS[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

19.10.5 CCRy

Таблица 284 – Регистр сравнения/захвата для ‘у’ канала таймера CCRy

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

Таблица 285 – Описание бит регистра CCRy

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

19.10.6 CCRy1

Таблица 286 – Регистр сравнения/захвата для ‘у’ канала таймера CCRy1

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

Таблица 287 – Описание бит регистра CCRy1

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.

19.10.7 CHy_CNTRL

Таблица 288 – Регистр управления для ‘у’ канала таймера CHy_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 289 – Описание бит регистра CHy_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 CCRy1: 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 CCRy: 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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; 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 CHu _i для фиксации значения основного счетчика (регистр CNT) в регистр CCRy: 00 – положительный фронт на входном канале CHu _i ; 01 – отрицательный фронт на входном канале CHu _i ; 10 – положительный фронт от других каналов: для первого канала от второго канала; для второго канала от третьего канала; для третьего канала от четвертого канала; для четвертого канала от первого канала;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		11 – положительный фронт от других каналов: для первого канала от третьего канала; для второго канала от четвертого канала; для третьего канала от первого канала; для четвертого канала от второго канала
3...0	CHFLTR[3:0]	Конфигурация фильтра на входе канала 'у'. Выбор частоты выборки F_s и количества выборок N : 0000 – нет фильтрации, $F_s = F_{DTS}$; 0001 – $F_s = TIM_CLK$, $N = 2$; 0010 – $F_s = TIM_CLK$, $N = 4$; 0011 – $F_s = TIM_CLK$, $N = 8$; 0100 – $F_s = F_{DTS}/2$, $N = 6$; 0101 – $F_s = F_{DTS}/2$, $N = 8$; 0110 – $F_s = F_{DTS}/4$, $N = 6$; 0111 – $F_s = F_{DTS}/4$, $N = 8$; 1000 – $F_s = F_{DTS}/8$, $N = 6$; 1001 – $F_s = F_{DTS}/8$, $N = 8$; 1010 – $F_s = F_{DTS}/16$, $N = 5$; 1011 – $F_s = F_{DTS}/16$, $N = 6$; 1100 – $F_s = F_{DTS}/16$, $N = 8$; 1101 – $F_s = F_{DTS}/32$, $N = 5$; 1110 – $F_s = F_{DTS}/32$, $N = 6$; 1111 – $F_s = F_{DTS}/32$, $N = 8$

19.10.8 CHy_CNTRL1

Таблица 290 – Регистр управления 1 для 'у' канала таймера CHy_CNTRL1

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

Таблица 291 – Описание бит регистра CHy_CNTRL1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31...13	-	Зарезервировано.
12	NINV	Инверсия инверсного выхода nCHy: 0 – выход не инвертируется; 1 – выход инвертируется
11...10	NSELO[1:0]	Выбор источника сигнала для инверсного выхода nCHy: 00 – на nCHyо выдается 0; 01 – на nCHyо выдается 1; 10 – на nCHyо выдается сигнал nREF; 11 – на nCHyо выдается сигнал с DTG

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

19.10.9 CHy_CNTRL2

Таблица 292 – Регистр управления 2 для ‘у’ канала таймера CHy_CNTRL2

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

Таблица 293 – Описание бит регистра CHy_CNTRL2

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...4	-	Зарезервировано
3	CCRRLD	Режим обновления регистров CCRy и CCRy1: 0 – обновление возможно в любой момент времени; 1 – обновление будет осуществлено только при CNT == 0
2	CCR1_EN	Разрешение работы регистра CCRy1: 0 – CCRy1 не используется; 1 – CCRy1 используется

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1...0	CHSEL[1:0]	Выбор события по входному каналу CHy _i для фиксации значения основного счетчика (регистр CNT) в регистр CCRy1: 00 – положительный фронт на входном канале CHy _i ; 01 – отрицательный фронт на входном канале CHy _i ; 10 – отрицательный фронт от других каналов: для первого канала от второго канала; для второго канала от третьего канала; для третьего канала от четвертого канала; для четвертого канала от первого канала; 11 – отрицательный фронт от других каналов: для первого канала от третьего канала; для второго канала от четвертого канала; для третьего канала от первого канала; для четвертого канала от второго канала

19.10.10 CHy_DTG

Таблица 294 – Регистр управления генератором «мертвой зоны» CHy_DTG

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

Таблица 295 – Описание бит регистра CHy_DTG

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...8	DTG[7:0]	Основной делитель частоты DTG. Задержка $DTG_{del} = DTG \cdot (DTG_x + 1)$
7...5	-	Зарезервировано
4	EDTS	Частота работы DTG: 0 – TIM_CLK; 1 – F _{DTS}
3...0	DTGx[3:0]	Предварительный делитель частоты DTG

19.10.11 BRKETR_CNTRL

Таблица 296 – 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 297 – 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_S и количества выборок N : 0000 – нет фильтрации, $F_S = F_{DTS}$; 0001 – $F_S = TIM_CLK$, $N = 2$; 0010 – $F_S = TIM_CLK$, $N = 4$; 0011 – $F_S = TIM_CLK$, $N = 8$; 0100 – $F_S = F_{DTS}/2$, $N = 6$; 0101 – $F_S = F_{DTS}/2$, $N = 8$; 0110 – $F_S = F_{DTS}/4$, $N = 6$; 0111 – $F_S = F_{DTS}/4$, $N = 8$; 1000 – $F_S = F_{DTS}/8$, $N = 6$; 1001 – $F_S = F_{DTS}/8$, $N = 8$; 1010 – $F_S = F_{DTS}/16$, $N = 5$; 1011 – $F_S = F_{DTS}/16$, $N = 6$; 1100 – $F_S = F_{DTS}/16$, $N = 8$; 1101 – $F_S = F_{DTS}/32$, $N = 5$; 1110 – $F_S = F_{DTS}/32$, $N = 6$; 1111 – $F_S = F_{DTS}/32$, $N = 8$
3...2	ETR_PSC[1:0]	Асинхронный предделитель частоты со входа ETR: 00 – без деления; 01 – /2; 10 – /4; 11 – /8
1	ETR_INV	Инверсия входа ETR: 0 – без инверсии; 1 – инверсия
0	BRK_INV	Инверсия входа BRK: 0 – без инверсии; 1 – инверсия

19.10.12 STATUS

Таблица 298 – Регистр статуса таймера STATUS

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

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

Таблица 299 – Описание бит регистра STATUS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...13	CCR CAP1 EVENT[3:0]	Событие записи значения счетчика CNT в регистр CCRy1 по захвату настроенного фронта на входе канала CHy1: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал; Бит 3 – четвертый канал
12...9	CCR REF EVENT[3:0]	Событие переднего фронта на выходе генератора опорного сигнала REF: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал; Бит 3 – четвертый канал
8...5	CCR CAP EVENT[3:0]	Событие записи значения счетчика CNT в регистр CCRy по захвату настроенного фронта на входе канала CHy1: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал; Бит 3 – четвертый канал
4	BRK 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

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

19.10.13 IE

Таблица 300 – Регистр разрешения прерываний таймера IE

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

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

Таблица 301 – Описание бит регистра IE

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...13	CCR CAP1 EVENT IE[3:0]	Флаг разрешения прерывания по событию CCR CAP1 EVENT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
12...9	CCR REF EVENT IE[3:0]	Флаг разрешения прерывания по событию CCR REF EVENT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
8...5	CCR CAP EVENT IE[3:0]	Флаг разрешения прерывания по событию CCR CAP EVENT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
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
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

19.10.14 DMA_RE

Таблица 302 – Регистр DMA_RE разрешения запросов DMA

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

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

Таблица 303 – Описание бит регистра DMA_RE

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16...13	CCR CAP1 EVENT RE [3:0]	Флаг разрешения запроса DMA по событию CCR CAP1 EVENT[3:0]: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – первый канал; Бит 3 – четвертый канал
12...9	CCR REF EVENT RE[3:0]	Флаг разрешения запроса DMA по событию CCR REF EVENT[3:0]: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – первый канал; Бит 3 – четвертый канал
8...5	CCR CAP EVENT RE [3:0]	Флаг разрешения запроса DMA по событию CCR CAP EVENT[3:0]: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – первый канал; Бит 3 – четвертый канал
4	BRK EVENT RE	Флаг разрешения запроса DMA по событию BRK EVENT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
3	ETR FE EVENT RE	Флаг разрешения запроса DMA по событию ETR FE EVENT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
2	ETR RE EVENT RE	Флаг разрешения запроса DMA по событию ETR RE EVENT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
1	CNT ARR EVENT RE	Флаг разрешения запроса DMA по событию CNT ARR EVENT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
0	CNT ZERO EVENT RE	Флаг разрешения запроса DMA по событию CNT ZERO EVENT: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен

20 Контроллер АЦП

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

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 бит Cfg_REG_DIVCLK[3:0]. Максимальная частота CLK не может превышать 14 МГц.

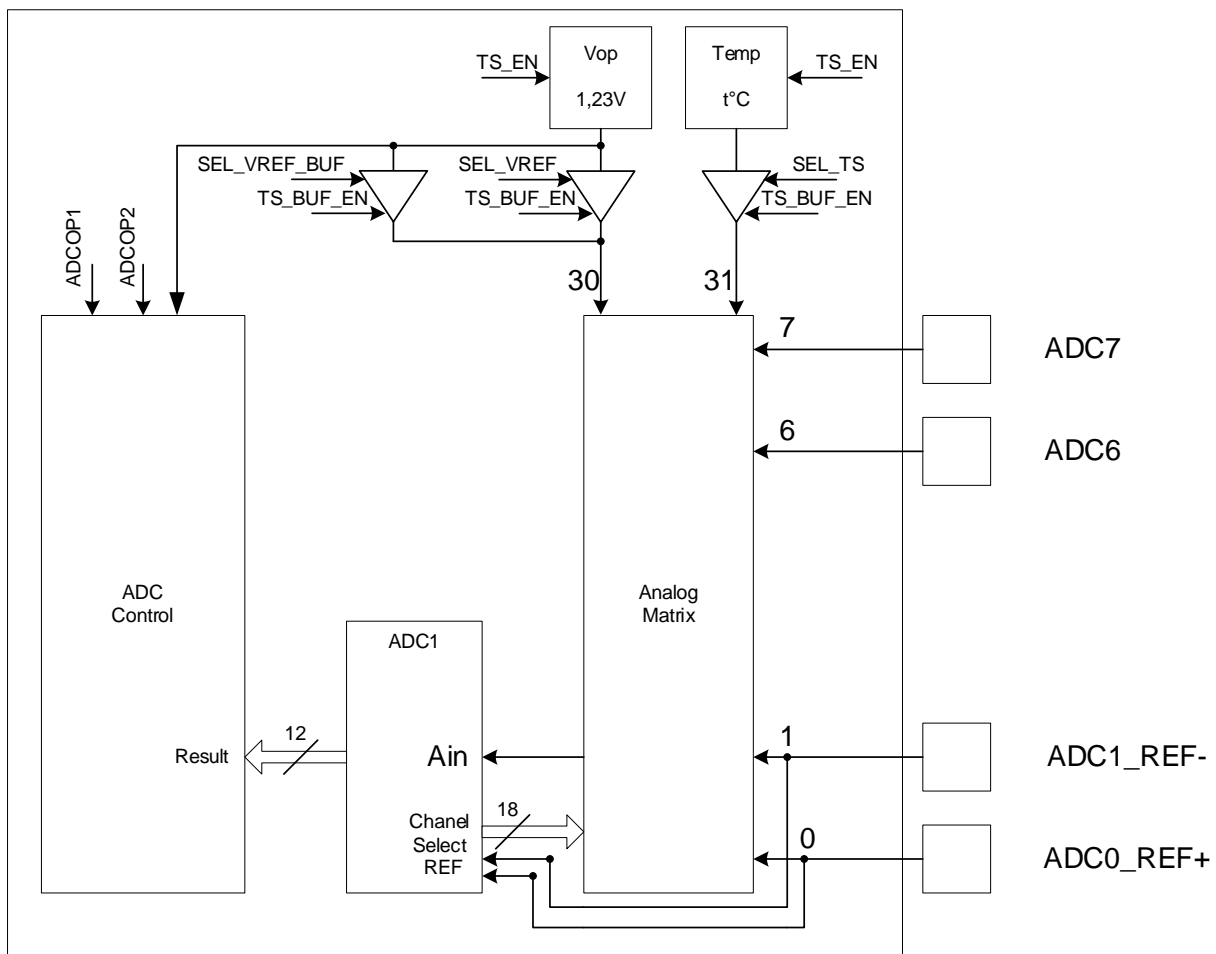


Рисунок 66 – Структурная схема контроллера АЦП

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 в «1». После включения можно использовать источник опорного напряжения для АЦП вместо его собственного. Для этого необходимо установить биты ADCx_OP в единицу. Для преобразования необходимо, чтобы выводы, используемые АЦП в порте C, были сконфигурированы как аналоговые и были отключены какие-либо внутренние подтяжки.

20.1 Преобразование внешнего канала

В регистре ADCx_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 ADCx_STATUS. А в регистре ADCx_RESULT будет результат преобразования. После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADCx_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Флаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADCx_STATUS.

20.2 Последовательное преобразование нескольких каналов

Для автоматического последовательного преобразования нескольких каналов или одного канала в регистре ADCx_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 бит Delay_GO можно задать паузу между преобразованиями при переборе каналов. Для начала преобразования необходимо записать «1» в бит Cfg_REG_GO.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADCx_STATUS. А в регистре ADCx_RESULT будет результат преобразования. После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADCx_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Флаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADCx_STATUS.

Для последовательного преобразования одного и того же канала можно в регистре ADCx_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.

20.3 Преобразование с контролем границ

При необходимости отслеживать нахождение оцифрованного значения в допустимых пределах можно задать нижнюю и верхнюю допустимые границы в регистрах ADCx_L_LEVEL и ADCx_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.

20.4 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 ADCx_OP в «1». 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.

20.5 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 ADCx_OP в «1». 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.

20.6 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:

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

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

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

N – требуемая точность в разрядах;

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

T_{track} – время заряда внутренней емкости (определяется формулой (14)), [с];

f_{C_ADC} – рабочая частота АЦП (определяется Cfg REG CLKS в регистре ADC1_CFG), [с⁻¹];

f_{PCLKd} – определяется формулой:

$$f_{PCLKd} = \frac{f_{PCLK}}{2^{Cfg_REG_DIVCLK}}. \quad (15)$$

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

Если необходимо обеспечить преобразование с точностью 12 разрядов $\pm 1/4$ 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 $\pm 1/4$ 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 304.

Таблица 304 – 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
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 EDLADC, EILADC и EOFFADC.

Для корректного задания режимов работы АЦП в регистре ADCx_CFG необходимо сделать до задания бита Cfg REG GO, иначе новая конфигурация будет действовать со следующего преобразования.

20.7 Описание регистров блока контроллера АЦП

Таблица 305 – Описание регистров блока контроллера АЦП

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

20.7.1 ADCx_CFG

Таблица 306 – Регистр ADCx_CFG

Номер	31...28	27...25	24...21	20	19
Доступ	R/W	U	U	R/W	R/W
Сброс	0	0	0	0	0
	DelayADC[3:0]	DelayGo[2:0]	-	SELVREF	SELTS

Номер	18	17	16	15...12	11	10
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	TS_BUF EN/ADC2 OP	TS_EN /ADC1 OP	Cfg Sync Conver	Cfg REG DIVCLK[3:0]	Cfg M_REF	Cfg REG RNGC

Номер	9	8...4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	Cfg REG CHCH	Cfg REG CHS[4:0]	Cfg REG SAMPLE	Cfg REG CLKS	Cfg REG GO	Cfg REG ADON

Таблица 307 – Описание бит регистра ADCx_CFG

Разряды	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 1,23 В: 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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
18	TS BUF EN	В регистре ADC1_CFG Включения выходного усилителя для датчика температуры и источника опорного напряжения: 0 – выключен; 1 – включен. Используется при TS_EN = 1. Для уменьшения тока потребления
17	TS EN	В регистре ADC1_CFG Включения датчика температуры и источника опорного напряжения: 0 – выключен; 1 – включен. При включении датчика температуры и источника опорного напряжения выходной сигнал стабилизируется в течение времени Tstb
17	ADC1 OP	В регистре ADC2_CFG Выбор источника опорного напряжения 1,23 В: 0 – внутренний (не точный); 1 – от датчика температуры (точный)
16	CfgSyncConver	Всегда записывать ноль
15...12	Cfg REG DIVCLK [3:0]	Выбор коэффициента деления частоты PCLK 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

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	Cfg REG SAMPLE	Выбор способа запуска АЦП: 1 – последовательный: автоматический запуск после завершения предыдущего преобразования; 0 – одиночный
2	Cfg REG CLKS	Выбор источника синхросигнала работы C_ADC работы ADC: 1 – PCLKd (определяется по формуле(15)); 0 – ADC_CLK
1	Cfg REG GO	Начало преобразования Запись «1» начинает процесс преобразования, сбрасывается автоматически
0	Cfg REG ADON	Включение АЦП: 1 – включено; 0 – выключено

20.7.2 ADC1_H_LEVEL

Таблица 308 – Регистр ADC1_H_LEVEL

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

Таблица 309 – Описание бит регистра ADC1_H_LEVEL

Разряды	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.

20.7.3 ADC1_L_LEVEL

Таблица 310 – Регистр ADC1_L_LEVEL

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

Таблица 311 – Описание бит регистра ADC1_L_LEVEL

Разряды	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.

20.7.4 ADC1_RESULT

Таблица 312 – 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 313 – Описание бит регистра ADC1_RESULT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...21	-	Зарезервировано
20...16	CHANNEL[11:0]	Канал результата преобразования
15...12	-	Зарезервировано
11...0	RESULT[11:0]	Значение результата преобразования

20.7.5 ADC1_STATUS

Таблица 314 – Регистр ADC1_STATUS

Номер	31...5	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	-	ECOIF_IE	AWOIF_IE	FlgREG EOCIF	FlgREG AWOIFEN	FlgREG OVERWRITE

Таблица 315 – Описание бит регистра ADC1_STATUS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...5	-	Зарезервировано
4	ECOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_ECOIF: 0 – прерывание не генерируется; 1 – прерывание генерируется
3	AWOIF_IE	Флаг разрешения генерирования прерывания по событию Flg_REG_AWOIFEN: 0 – прерывание не генерируется; 1 – прерывание генерируется
2	Flg REG EOCIF	Флаг выставляется, когда закончено преобразование и данные еще не считаны. Очищается считыванием результата из регистра ADCx_RESULT: 1 – есть готовый результат преобразования; 0 – нет результата
1	Flg REG AWOIFEN	Флаг выставляется, когда результат преобразования выше верхней или ниже нижней границы автоматического контроля уровней. Сбрасывается только при записи нуля в данный бит: 0 – результат в допустимой зоне; 1 – вне допустимой зоны

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

20.7.6 ADC1_CHSEL

Таблица 316 – Регистр ADC1_CHSEL

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

Таблица 317 – Описание бит регистра ADC1_CHSEL

Разряды	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

20.7.7 ADC1_TRIM

Таблица 318 – 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 319 – Описание бит регистра ADC1_TRIM

Разряды	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

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.
- Содержит независимые буферы приема и передачи (8 ячеек, 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.
- Поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена на рисунке 93.

21.5 Характеристики интерфейса SSI

Интерфейс SSI фирмы Texas Instruments обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии;
- возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

21.6 Описание функционирования

Глава содержит описание основных функциональных блоков синхронного последовательного интерфейса SSP и содержит следующие разделы:

- Общий обзор модуля.
- Функциональное описание модуля.
- Описание работы модуля.

21.6.1 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 и принимаются с линии SSP_RXD.

Модуль SSP содержит программируемые делители частоты, формирующие тактовый сигнал обмена данными SSPCLKOUT из сигнала, поступающего на линию SSP_CLK. Скорость передачи данных может достигать более 2 МГц, в зависимости от частоты SSP_CLK и характеристик подключенного периферийного устройства.

Режим обмена данными, формат информационного кадра и количество бит данных задаются программно с помощью регистров управления SSPCR0 и SSPCR1.

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.

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

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

21.6.2 Блок формирования тактового сигнала

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSPCLKOUT 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, исключается возможность деления частоты на нечетный коэффициент, что, в свою очередь гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровня).

Сформированный описанным образом сигнал далее поступает на второй делитель частоты, с выход которого и снимается тактовый сигнал обмена данными SSPCLKOUT.

Коэффициент деления второго делителя задается программно в диапазоне от 1 до 256, путем записи соответствующего значения в регистр управления SSPCR0.

21.6.3 Буфер FIFO передатчика

Буфер передатчика имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа «первый вошел, первый вышел». Данные от центрального процессора, записанные через шину AMBA APB, сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных.

21.6.4 Буфер FIFO приемника

Буфер приемника имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа «первый вошел, первый вышел». Принятые от периферийного устройства данные сохраняются блоком приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину AMBA APB.

21.6.5 Блок приема и передачи данных

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSPCLKOUT для подключенных ведомых устройств. Как было описано ранее, данный сигнал формируется путем деления частоты сигнала SSPCLK.

Блок передатчика последовательно считывает значения из буфера FIFO передатчика и производит их преобразование из параллельной в последовательную форму. Далее поток последовательных данных и элементов кадровой синхронизации, тактированных сигналом SSPCLKOUT, 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 через интерфейс шины APB.

В режиме ведомого устройства тактовый сигнал обмена данными формируется одним из подключенных к модулю периферийных устройств и поступает по линии SSPCLKIN. При этом блок передатчика, тактируемый этим внешним сигналом, считывает данные из буфера FIFO, преобразует их из параллельной формы в последовательную, после 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 SSPCLKIN, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны через интерфейс шины APB.

Примечание – В режиме работы ведомого устройства, запросы ведущим устройством, на выдачу информации от ведомого, необходимо осуществлять при наличии данных в FIFO передатчика ведомого.

21.6.6 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 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.

Модуль SSP позволяет использовать оба описанных выше подхода.

Предусмотрены независимые линии запроса прерывания по готовности приемника и передатчика SSPTXINTR и SSPRXINTR, что позволяет обслуживание устройства (чтение или запись данных) по достижению заданного уровня заполнения буферов FIFO приемника или передатчика.

21.6.7 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 SSPCR0 и SSPCR1.

Кроме того, для установки требуемой скорости передачи данных необходимо выбрать параметры блока формирования тактового сигнала с учетом значения частоты внешнего сигнала SSPCLK и записать соответствующую информацию в регистр SSPCPSR.

21.6.8 Разрешение работы приемопередатчика

Разрешение осуществляется путем установки бита SSE регистра управления SSPCR1. Буфер FIFO передатчика может быть либо проинициализирован путем записи в него до восьми 16-разрядных слов заблаговременно перед установкой этого бита, либо заполняться передаваемыми данными в процедуре обслуживания прерывания.

После разрешения работы модуля приемопередатчик начинает обмен данными по линиям SSP_TXD и SSP_RXD.

21.6.9 Соотношения между тактовыми сигналами

В модуле имеется ограничение на соотношение между частотами тактовых сигналов PCLK и SSP_CLK. Частота SSP_CLK должна меньше или равна частоте PCLK. Выполнение этого требования гарантирует синхронизацию сигналов управления, передаваемых из зоны действия тактового сигнала SSP_CLK в зону действия сигнала PCLK в течение времени, меньшего продолжительности передачи одного информационного кадра:

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

В режиме ведомого устройства сигнал SSPCLKIN от ведущего внешнего устройства поступает на схемы синхронизации, задержки и обнаружения фронта. Для того чтобы обнаружить фронт сигнала SSPCLKIN необходимо три такта сигнала SSPCLK. Сигнал SSP_TXD имеет меньшее время установки по отношению к заднему фронту SSPCLKIN, по которому и происходит считывание данных из линии. Время установки и удержания сигнала SSP_RXD по отношению к сигналу SSPCLKIN должно выбираться с запасом, гарантирующим правильное считывание данных. Для

обеспечения корректной работы устройства необходимо, чтобы частота SSP_CLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSPCLKIN.

Выбор частоты тактового сигнала SSP_CLK должен обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала SSP_CLK к максимальной частоте сигнала SSPCLKOUT в режиме ведомого устройства равно 12, в режиме ведущего – двум.

Так, в режиме ведущего устройства для обеспечения максимальной скорости обмена 1,8432 Мбит/с частота сигнала SSP_CLK должна составлять не менее 3,6864 МГц. В этом случае в регистр SSPCPSR должно быть записано значение 2, а поле SCR[7:0] регистра SSPCR0 должно быть установлено в 0.

В режиме ведомого устройства для обеспечения той же информационной скорости необходимо использовать тактовый сигнал SSP_CLK с частотой не менее 22,12 МГц. При этом в регистр SSPCPSR должно быть записано значение 12, а поле SCR[7:0] регистра SSPCR0 должно быть установлено в 0.

Соотношение между максимальной частотой сигнала SSP_CLK и минимальной частотой SSPCLKOUT составляет $254 \cdot 256$.

Минимальная допустимая частота сигнала SSP_CLK определяется следующей системой соотношений, которые должны выполняться одновременно:

$$F_{SSPCLK}(\min) \Rightarrow 2 \times F_{SSPCLKOUT}(\max) \text{ [for master mode]}$$

$$F_{SSPCLK}(\min) \Rightarrow 12 \times F_{SSPCLKIN}(\max) \text{ [for slave mode].}$$

Аналогично, максимальная допустимая частота сигнала SSP_CLK определяется следующей системой соотношений, которые должны выполняться одновременно:

$$F_{SSPCLK}(\max) \leq 254 \times 256 \times F_{SSPCLKOUT}(\min) \text{ [for master mode]}$$

$$F_{SSPCLK}(\max) \leq 254 \times 256 \times F_{SSPCLKIN}(\min) \text{ [for slave mode].}$$

21.6.10 Программирование регистра управления SSPCR0

Регистр SSPCR0 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 SSP_CLK 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, кроме того, задается полярность и фаза сигнала (биты SPH и SPO).

21.6.11 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 SSPTXD output disable – запрет линии SSP_TXD для ведомого режима) регистра SSPCR1. 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).

21.6.12 Формирование тактового сигнала обмена данными

Тактовый сигнал обмена данными формируется путем деления частоты тактового сигнала SSP_CLK. На первом этапе формирования частота этого сигнала делится на четный коэффициент CPSDVSR, лежащий в диапазоне от 2 до 254, доступный для программирования через регистр SSPCPSR. Сформированный сигнал далее поступает на делитель частоты с коэффициентом (1 + SCR) от 1 до 256, где значение SCR доступно для программирования через SSPCR0.

Частота выходного тактового сигнала обмена данными SSPCLKOUT определяется следующим соотношением:

$$F_{SSPCLKOUT} = F_{SSPCLK} / (CPSDVR \times (1+SCR))$$

Например, в случае если частота сигнала SSP_CLK составляет 3,6864 МГц, а значение CPSDVSR = 2, частота сигнала SSPCLKOUT лежит в интервале от 7,2 кГц до 1,8432 МГц.

21.6.13 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 SSPCLKOUT формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SSPCLKOUT 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 SSPFSSOUT 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 SSPFSSOUT формируется импульс с длительностью, равной одному тактовому интервалу обмена данными. В этом режиме приемопередатчик SSP, равно как ведомые периферийные устройства, передает данные в линию по переднему фронту сигнала SSPCLKOUT, 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.

21.6.14 Формат синхронного обмена SSI фирмы Texas Instruments

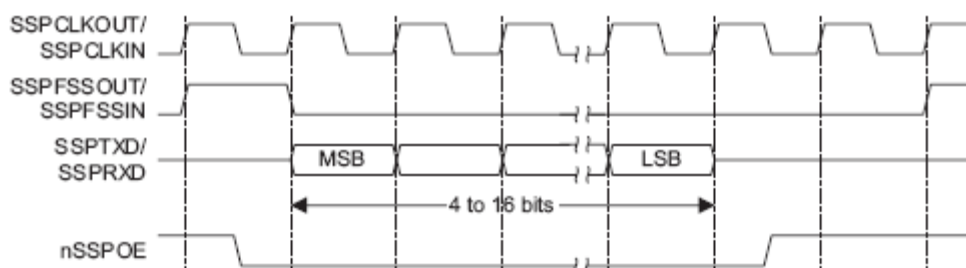


Рисунок 68 – Формат синхронного обмена протокола SSI фирмы Texas Instruments (единичный обмен)

В данном режиме при неактивном приемопередатчике SSP сигналы SSPCLKOUT и SSPFSSOUT 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 SSPCLKOUT. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему

фронту сигнала SSPCLKOUT старший значащий разряд информационного кадра (4-16 бит данных) выдается на выход линии SSPTXD и т.д.

В режиме приема данных как модуль SSP, так и ведомое внешнее устройство последовательно загружают биты данных в сдвиговый регистр по заднему фронту сигнала SSPCLKOUT. Принятые данные переносятся из сдвигового регистра в буфер FIFO после загрузки в него младшего значащего бита данных по очередному переднему фронту сигнала SSPCLKOUT.

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments представлены на рисунке 69.

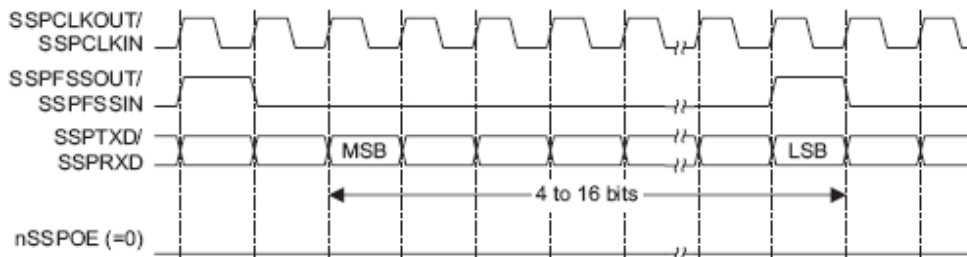


Рисунок 69 – Формат синхронного обмена протокола SSI фирмы Texas Instruments (непрерывный обмен)

21.6.15 Формат синхронного обмена SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSPFSSOUT выполняет функцию выбора ведомого устройства. Главной особенностью протокола SPI является возможность выбора состояния и фазы сигнала SSPCLKOUT 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 SSPCLKOUT переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SSPCLKOUT 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.

21.6.16 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 рисунках 70, 71.

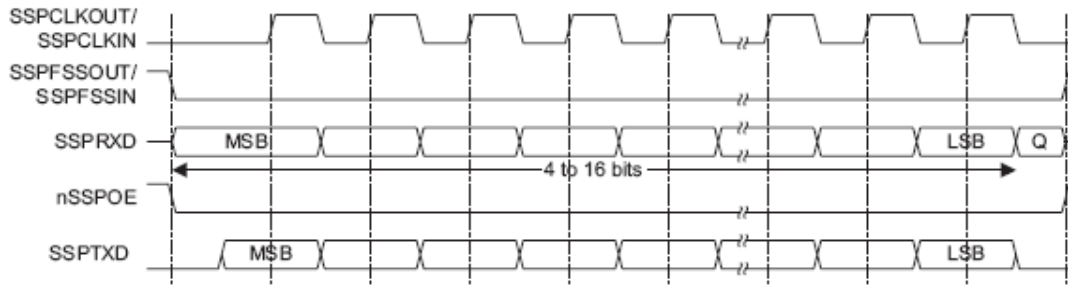


Рисунок 70 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=0(одиночный обмен)

Примечание – на рисунках 70 буквой Q обозначен сигнал с неопределенным уровнем.

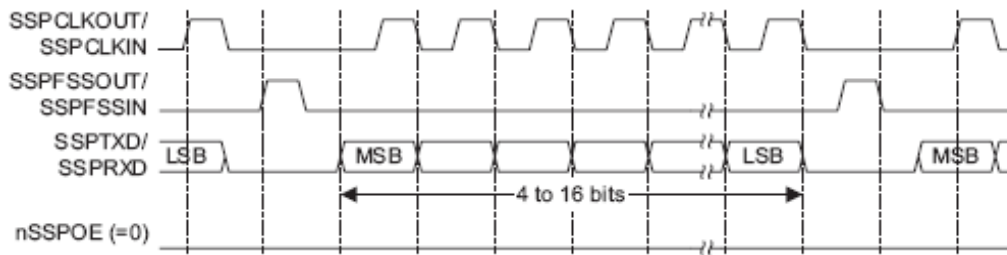


Рисунок 71 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет низкий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя таким образом выходной контакт SSPTXD передатчика в высокоимпедансное состояние;
- если модуль сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится в низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT;
- если модуль сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится в высокий уровень, отключая выходной контакт SSPCLKOUT.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится в низкий уровень, переводя выходной контакт передатчика SSPTXD из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLKOUT, на линии SSPTXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSPCLKOUT переводится в высокий логический уровень.

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

В случае передачи одного слова данных, после приема его последнего бита линия SSPFSSOUT переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной передачи данных, на линии SSPFSSOUT 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 SSPFSSOUT в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSPFSSOUT переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLKOUT.

21.6.17 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).

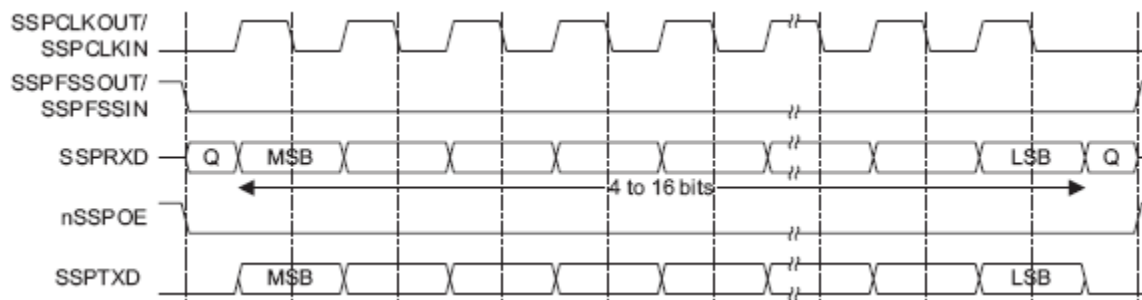


Рисунок 72 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=0,SPH=1

Примечание – на рисунке 72 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет низкий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя таким образом выходной контакт SSPTXD передатчика в высокоимпедансное состояние;
- если модуль сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится в низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT;
- если модуль сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится в высокий уровень, отключая выходной контакт SSPCLKOUT.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится в

низкий уровень, переводя выходной контакт передатчика SSPTXD из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLKOUT на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых битов передаваемых данных. В это же время включается линия SSPCLKOUT и на ней формируется передний фронт сигнала.

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

В случае передачи одного слова данных, после приема его последнего бита линия SSPFSSOUT переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной передачи данных, линия SSPFSSOUT 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.

21.6.18 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 рисунках 73, 74.

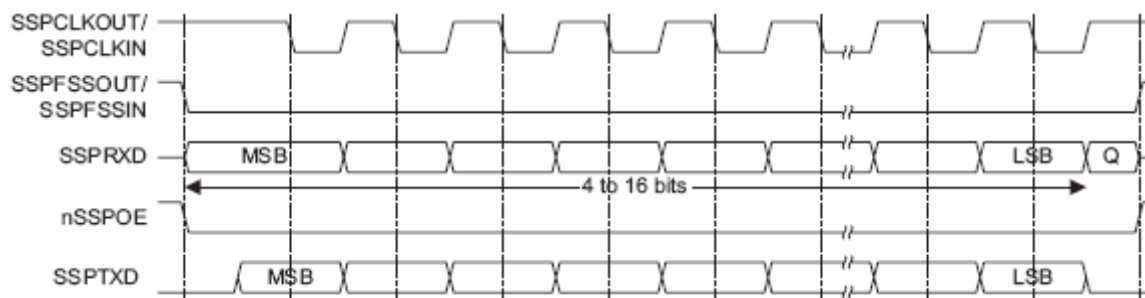


Рисунок 73 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)

Примечание – на рисунке 73 буквой Q обозначен сигнал с неопределенным уровнем.

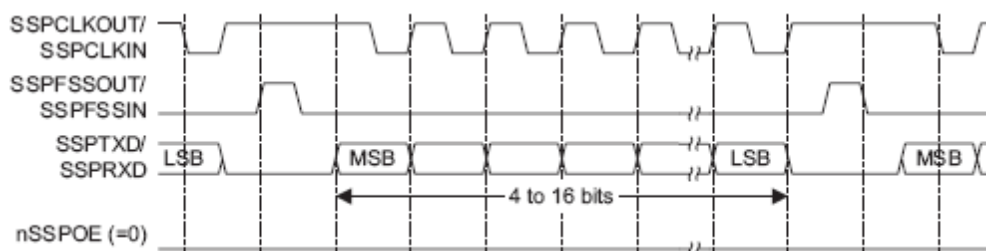


Рисунок 74 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет высокий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;

– сигнал nSSPOE переводится в высокий уровень, переводя таким образом выходной контакт SSPTXD передатчика в высокоимпедансное состояние;

- если модуль сконфигурирован как ведущее устройство, линия nSSPCTL0E переводится в низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT;
- если модуль сконфигурирован как ведомое устройство, линия nSSPCTL0E переводится в высокий уровень, отключая выходной контакт SSPCLKOUT.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится в низкий уровень, переводя выходной контакт передатчика SSPTXD из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLKOUT, на линии SSPTXD формируется значение первого бита передаваемых данных. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSPCLKOUT переводится в низкий логический уровень.

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

В случае передачи одного слова данных, после приема его последнего бита линия SSPFSSOUT переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной передачи данных, на линии SSPFSSOUT 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 SSPFSSOUT в высокий уровень по окончании передачи каждого кадра, разрешая таким образом запись новых данных. По окончании приема последнего бита блока данных линия SSPFSSOUT переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLKOUT.

21.6.19 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).

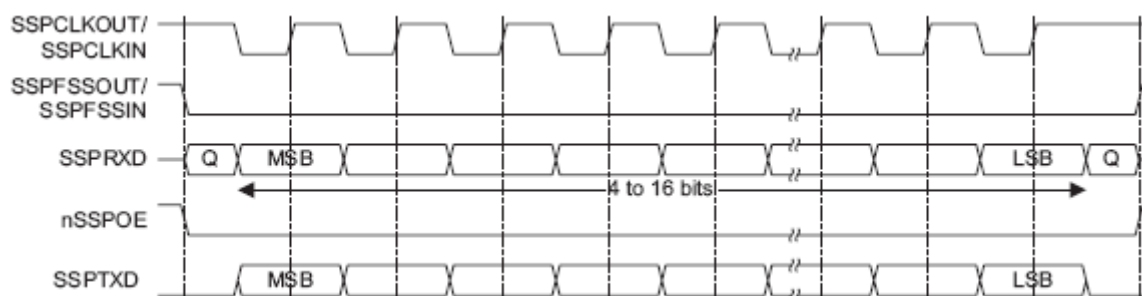


Рисунок 75 – Формат синхронного обмена протокола SPI фирмы Motorola, SPO=1, SPH=1

Примечание – на рисунке 75 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет высокий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя таким образом выходной контакт SSPTXD передатчика в высокоимпедансное состояние;
- если модуль сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится в низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT;
- если модуль сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится в высокий уровень, отключая выходной контакт SSPCLKOUT.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится в низкий уровень, переводя выходной контакт передатчика SSPTXD из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSPCLKOUT на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых битов передаваемых данных. В это же время включается линия SSPCLKOUT и на ней формируется передний фронт сигнала.

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

В случае передачи одного слова данных, после приема его последнего бита линия SSPFSSOUT переводится в высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной передачи данных, линия SSPFSSOUT 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.

21.6.20 Формат синхронного обмена Microwire фирмы National Semiconductor

Временные диаграммы последовательного синхронного обмена в режиме Microwire показаны на рисунках 76, 77.

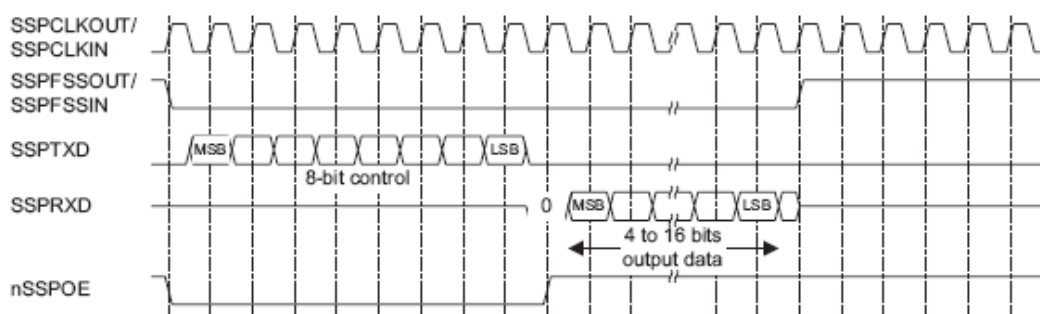


Рисунок 76 – Формат синхронного обмена протокола Microwire фирмы National Semiconductor (одиночный обмен)

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

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT имеет низкий логический уровень;
- сигнал SSPFSSOUT имеет высокий логический уровень;
- сигнал SSPTXD переводится в низкий логический уровень;
- сигнал nSSPOE переводится в высокий уровень, переводя, таким образом, выходной контакт SSPTXD передатчика в высокоимпедансное состояние.

Переход в режим информационного обмена происходит после записи управляющего байта в буфер FIFO передатчика. По заднему фронту сигнала SSPFSSOUT данные из буфера переносятся в регистр сдвига блока передатчика, откуда, начиная со старшего значащего разряда, последовательно выдаются в линию SSPTXD. Линия SSPFSSOUT остается в низком логическом уровне в течение всей передачи кадра. Линия SSPRXD при этом находится в высокоимпедансном состоянии.

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSPCLKOUT. По окончании приема последнего бита управляющей последовательности она декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля SSP. Биты данных выдаются в линию SSPRXD по заднему фронту сигнала SSPCLKOUT. 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 SSPFSSOUT переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных их регистра сдвига в буфер FIFO приемника.

Примечание – Внешнее устройство может перевести линию приемника в третье состояние по заднему фронту сигнала SSPCLKOUT после приема последнего бита слова данных, либо после перевода линии SSPFSSOUT в высокий логический уровень.

Непрерывный обмен данными начинается и заканчивается так же, как и в одиночный обмен. Однако линия SSPFSSOUT 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 SSPCLKOUT.

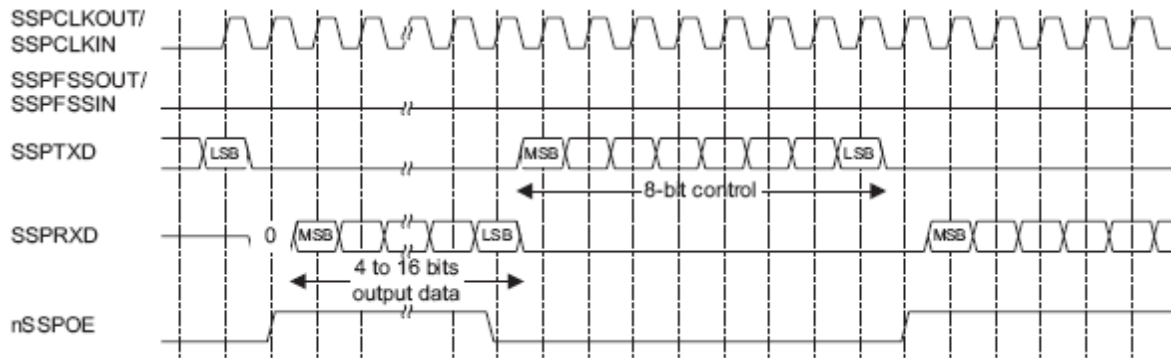


Рисунок 77 – Формат синхронного обмена протокола Microwire фирмы National Semiconductor (непрерывный обмен)

Требования к временным параметрам сигнала SSPFSSIN относительно тактового сигнала SSPCLKIN в режиме Microwire

Модуль SSP, работающий в режиме Microwire как ведомое устройство, регистрирует данные по переднему фронту сигнала SSPCLKIN после установки сигнала SSPFSSIN в низкий логический уровень. Ведущие устройства, формирующие сигнал SSPCLKIN должны гарантировать достаточное время установки и удержания сигнала SSPFSSIN по отношению к переднему фронту сигнала SSPCLKIN.

Иллюстрация данных требований представлена на рисунке 78. По отношению к переднему фронту сигнала SSPCLKIN, по которому осуществляется регистрация данных в приемнике ведомого модуля SSP, время установки сигнала SSPFSSIN должно быть, как минимум, в два раза больше периода SSPCLKIN, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSPCLKIN должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.

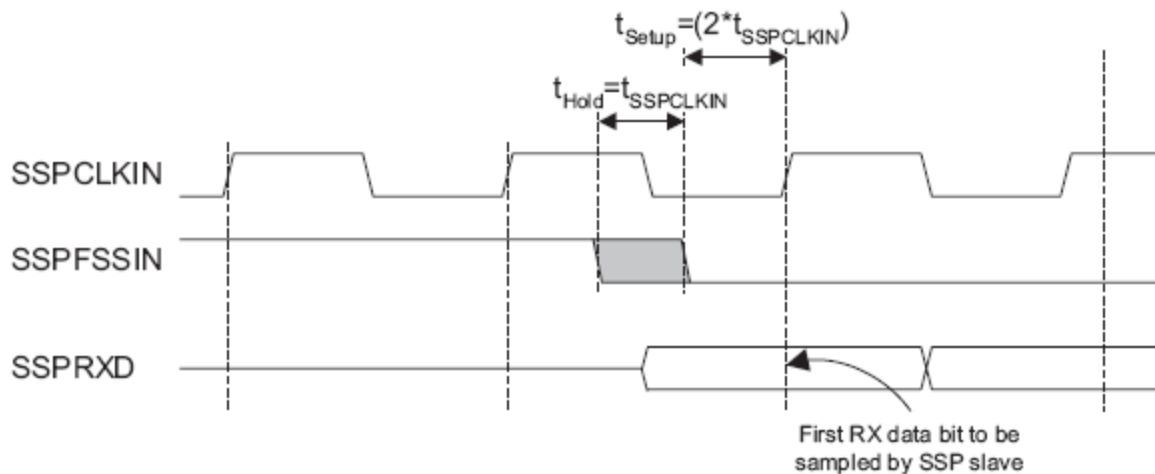


Рисунок 78 – Формат кадра Microwire, требования к времени установки и удержания сигнала SSPFSSIN

21.6.21 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.

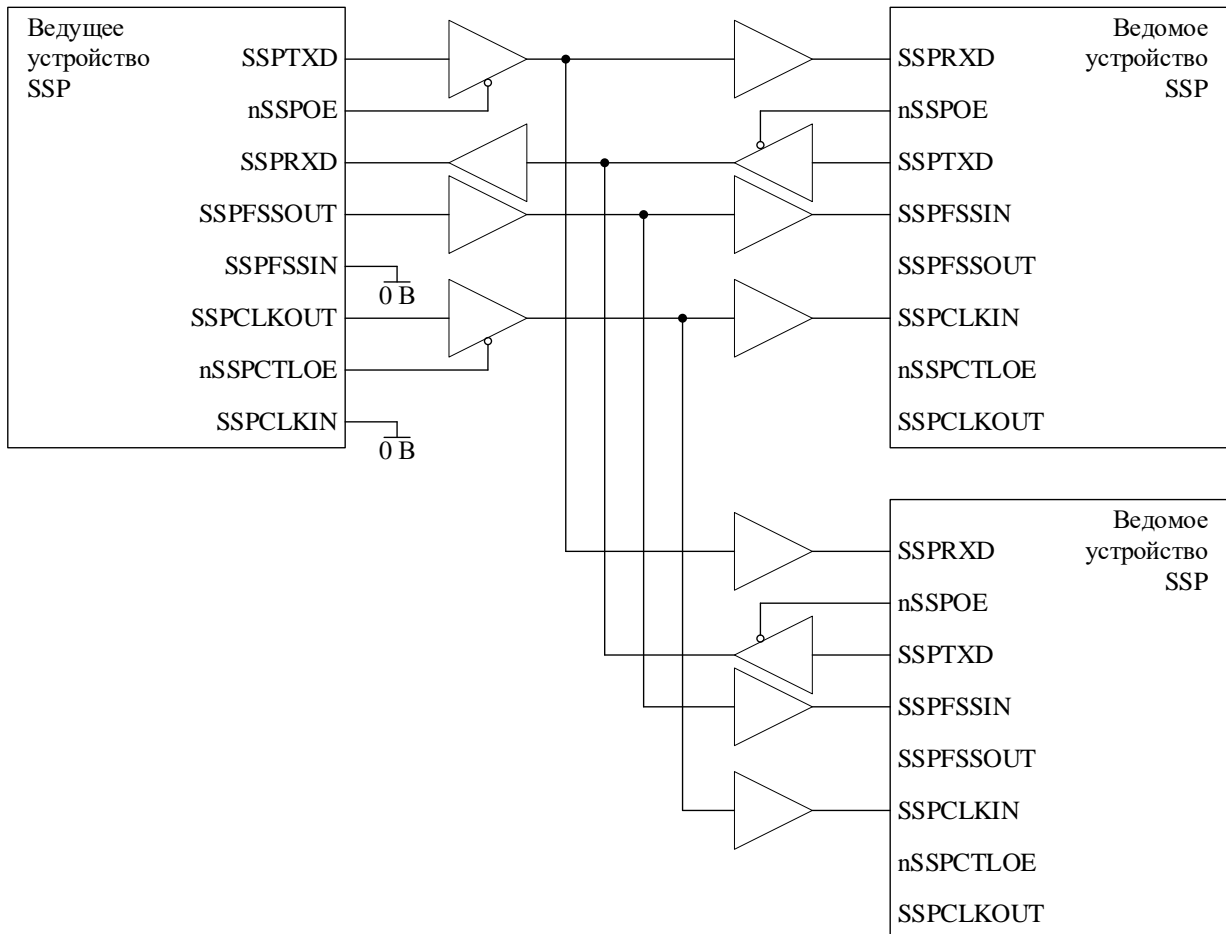


Рисунок 79 – Ведущее устройство SSP подключено к двум ведомым

На рисунке 79 показана совместная работа трех модулей SSP, один из которых сконфигурирован в качестве ведущего, а два – в качестве ведомых устройств. Ведущее устройство способно передавать данные циркулярно в адрес двух ведомых по линии SSPTXD.

Для ответной передачи данных один из ведомых модулей переводит линию nSSPOE в активное состояние, разрешая таким образом прохождение сигнала от своей линии SSPTXD на вход SSPRXD ведущего.

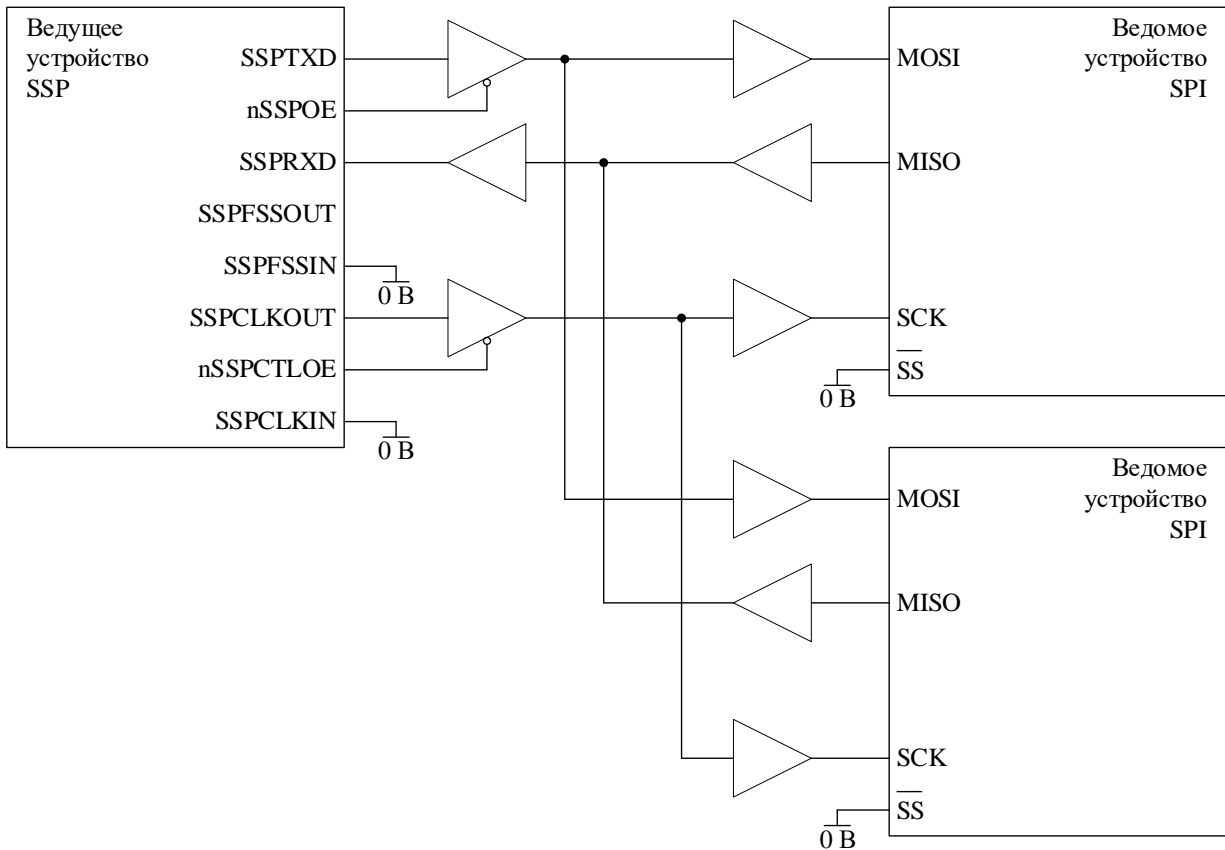


Рисунок 80 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим протокол SPI

На рисунке 80 показано подключение модуля SSP, сконфигурированного как ведущее устройство, к двум ведомым устройствам, поддерживающим протокол SPI фирмы Motorola. Внешние устройства сконфигурированы как ведомые путем установки в низкий логический уровень сигнала выбора ведомого устройства Slave Select (SS). Как и в предыдущем примере, ведущее устройство способно передавать данные в адрес ведомых циркулярно по линии SSPTXD. Ответная передача данных на входную линию SSPRXD ведущего устройства одновременно осуществляется только одним из ведомых по соответствующей линии MISO.

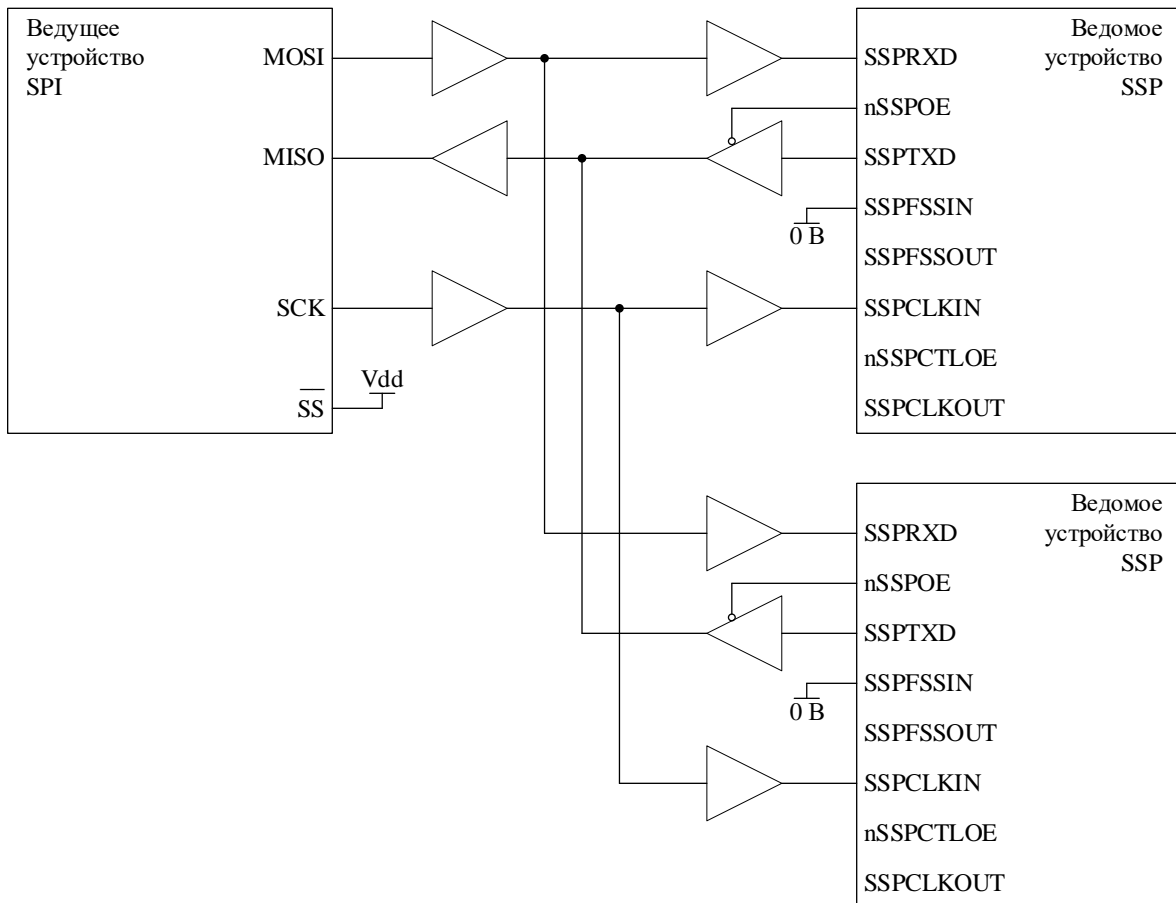


Рисунок 81 – Ведущее устройство, поддерживающее протокол SPI подключено к двум ведомым модулям SSP

На рисунке 81 показано 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 nSSPOE в активное состояние, разрешая, таким образом, прохождение сигнала от своей линии SSPTXD на вход SSPRXD ведущего.

21.6.22 Интерфейс прямого доступа к памяти

Модуль SSP предоставляет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA SSPDMACR.

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

- SSPRXDMASREQ – запрос передачи отдельного символа, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит, по меньшей мере, один символ;

- SSPRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если буфер FIFO приемника содержит четыре или более символов;

– SSPRXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Для передачи:

– SSPTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку;

– SSPTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит четыре или менее символов;

– SSPTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимно исключаящими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение четыре, формируется как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае если количество данных в буфере приема меньше порогового значения, формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

Примечание – Для оставшихся трех символов контроллер SSP не инициирует процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае снятия сигнала разрешения DMA.

В таблице 320 приведены значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 320 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/8	14	2
1/4	12	4

1/2	8	8
3/4	4	12
7/8	2	14

На рисунке 82 показаны временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

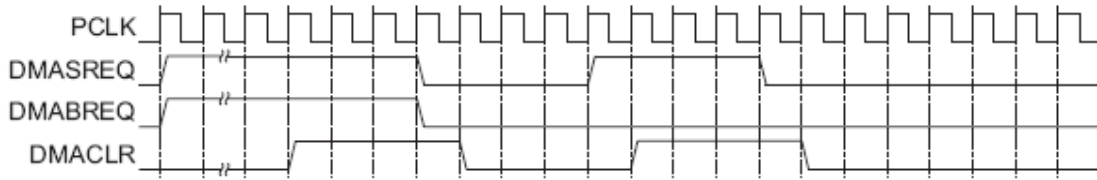


Рисунок 82 – Временные диаграммы обмена в режиме DMA

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.

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 SSPIMSC. 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 SSPRIS, либо из маскированного регистра прерываний SSPMIS.

21.7.1 SSPRXINTR

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

21.7.2 SSPTXINTR

Прерывание по заполнению буфера FIFO передатчика формируется в случае, если буфер передатчика содержит четыре или менее корректных слов данных.

Состояние прерывания не зависит от значения сигнала разрешения работы модуля SSP. Это позволяет организовать взаимодействие программного обеспечения с передатчиком одним из двух способов. Во-первых, можно записать данные в буфер заблаговременно, перед активизацией передатчика и разрешения прерываний. Во-вторых, можно предварительно разрешить работу модуля и формирование прерываний и заполнять буфер передатчика в ходе работы процедуры обслуживания прерываний.

21.7.3 SSPRORINTR

Прерывание по переполнению буфера FIFO приемника формируется в случае, если буфер уже заполнен и блоком приемника осуществлена попытка записать в него еще одно слово. При этом принятое слово данных регистрируется в регистре сдвига приемника, но в буфер приемника не заносится.

21.7.4 SSPRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение времени таймаута, равного 32 тактам частоты SSPCLKOUT (для ведущего и ведомого режимов работы). Данный механизм гарантирует, что пользователь будет знать о наличии в буфере приемника необработанных данных.

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

21.7.5 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

21.8.1 Общая информация

Базовый адрес модуля не фиксирован и может быть различным в разных системах. Смещение каждого регистра относительно базового адреса постоянно.

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.

21.8.2 Описание регистров контроллера SSP

Данные о регистрах модуля SSP приведены в таблице 321.

Таблица 321 – Обобщенные данные о регистрах модуля SSP

Базовый адрес	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x4000_0000	SSP				Контроллер SSP
Смещение					
0x000	SSPCR0	RW	0x0000	16	Регистр управления 0
0x004	SSPCR1	RW	0x0	4	Регистр управления 1
0x008	SSPDR	RW	0x----	16	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись)
0x00C	SSPSR	RO	0x03	3	Регистр состояния
0x010	SSPCPSR	RW	0x00	8	Регистр делителя тактовой частоты
0x014	SSPIMSC	RW	0x0	4	Регистр маски прерывания
0x018	SSPRIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования
0x01C	SSPMIS	RO	0x0	4	Регистр состояния прерываний с учетом маскирования
0x020	SSPICR	WO	0x0	4	Регистр сброса прерывания
0x024	SSPDMACR	RW	0x0	2	Регистр управления прямым доступом к памяти
Примечание – В поле «тип» указан вид доступа к регистру: RW – чтение и запись, RO – только чтение, WO – только запись					

21.8.2.1 SSP_CR0

Регистр управления 0.

Регистр SSPCR0 содержит пять битовых полей, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 322.

Таблица 322 –Регистр SSPCR0

Разряды	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

Разряды	Наименование	Назначение
		SSPCPSR), а SCR – число от 0 до 255
7	SPH	Фаза сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. пункт 21.6.15 «Формат синхронного обмена SPI фирмы Motorola»
6	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. пункт 21.6.15 «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

21.8.2.2 SSP_CRI

Регистр управления 1.

Регистр SSPCR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 323.

Таблица 323 – Регистр SSPCR1

Разряды	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

Разряды	Наименование	Назначение
		данных в линию SSPTXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSPTXD в ведомом режиме разрешена; 1 – управление линией SSPTXD 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

21.8.2.3 SSP_DR

Регистр данных.

Регистр SSPDR 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 SSPTXD с заданной скоростью информационного обмена.

В случае, если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника.

В режиме обмена данными Microwire фирмы National Semiconductor модуль SSP по умолчанию работает с восьмиразрядными информационными словами (старший значащий байт игнорируется). Размер принимаемых данных задается программно. Буфера FIFO приемника и передатчика автоматически не очищаются даже в случае, если бит SSE установлен в 0. Это позволяет заполнить буфер передатчика необходимой информацией заблаговременно, перед разрешением работы модуля.

Назначение разрядов регистра SSPDR описано в таблице 324.

Таблица 324 – Формат регистра SSPDR

Разряды	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 SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника
--	--	---

21.8.2.4 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 325 представлено назначение бит регистра SSPSR.

Таблица 325 – Регистр SSPSR

Разряды	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

21.8.2.5 SSP_CPSR

Регистр делителя тактовой частоты.

Регистр SSPCPSR 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 SSPCPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Назначение бит регистра SSPCPSR представлено в таблице 326.

Таблица 326 – Регистр SSPCPSR

Разряды	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
-------	---------	---

21.8.2.6 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 SSPIMSC показано в таблице 327.

Таблица 327 – Регистр SSPIMSC

Разряды	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

21.8.2.7 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 SSPRIS представлено в таблице 328.

Таблица 328 – Регистр SSPRIS

Разряды	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

21.8.2.8 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 SSPMIS представлено в таблице 329.

Таблица 329 – Регистр SSPMIS

Разряды	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

21.8.2.9 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 SSPICR представлено в таблице 330.

Таблица 330 – Регистр SSPICR

Разряды	Наименование	Назначение
15...2		Резерв. Не модифицируйте. При чтении выдаются нули
1	RTIC	Сброс прерывания SSPRTINTR
0	RORIC	Сброс прерывания SSPRORINTR

21.8.2.10 SSP_DMACR

Регистр управления прямым доступом к памяти.

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра SSPDMACR представлено в таблице 331.

Таблица 331 – Регистр SSPDMACR

Разряды	Наименование	Назначение
15...2		Резерв. Не модифицируйте. При чтении выдаются нули
1	TXDMAE	Использование DMA при передаче. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания

		буфера FIFO передатчика
0	RXDMAE	Использование DMA при приеме. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO приемника

22 Контроллер UART

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

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

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 ($1 \times 16 - 65535 \times 16$). 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, стопового битов и 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 битов.

Формирование и обнаружения сигнала разрыва линии.

Поддержка функция управления модемом (линии CTS, DCD, DSR, RTS, DTR и RI).

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.

Для проверки функционирования и соединений модуля предусмотрены дополнительные регистры тестирования.

22.4 Отличия от контроллера UART 16C650

Контроллер отличается от промышленного стандарта асинхронного приемопередатчика 16C650 следующими характеристиками:

- Пороги срабатывания прерывания по заполнению буфера FIFO приемника – 1/8, 1/4, 1/2, 3/4, и 7/8.
- Пороги срабатывания прерывания по заполнению буфера FIFO передатчика – 1/8, 1/4, 1/2, 3/4, и 7/8.
- Отличается распределение адресов внутренних регистров и назначение бит в регистрах.
- Недоступны изменения сигналов в состоянии модема.

Следующие возможности контроллера 16C650 не поддерживаются:

- Полуторная длительность стопового бита (поддерживается только 1 или 2 стоповых бита).
- Независимое задание тактовой частоты приемника и передатчика.

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.

Центральный процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии через интерфейс шины AMBA APB. Прием и передача данных буферизуются с помощью внутренней памяти FIFO, позволяющей сохранить до 16 байт 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 индустриального стандарта - контроллера UART 16C650.

Позволяет осуществлять обмен информацией с максимальной скоростью:

- в режиме UART – до 921600 бит/с;
- в режиме IrDA – до 460800 бит/с;
- в режиме IrDA с пониженным энергопотреблением – до 115200 бит/с.

Режим работы приемопередатчика и скорость обмена данными контролируются регистром управления линией UARTLCR_N 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.
- Сигналы запроса на прямой доступ к памяти (DMA) для совместной работы с контроллером DMA.

В случае возникновения ошибки в структуре сигнала, четности данных, а также разрыва линии соответствующий бит ошибки устанавливается и сохраняется в буфере FIFO. В случае переполнения буфера немедленно устанавливается соответствующий бит в регистре переполнения, а доступ к записи в буфер FIFO блокируется.

Существует возможность программно ограничить размер буфера FIFO одним байтом, что позволяет реализовать общепринятый интерфейс асинхронной последовательной связи с двойной буферизацией.

Поддерживаются входные линии состояния модема: «готовность к приему» (Clear To Send, CTS), «обнаружен информационный сигнал» (Data Carrier Detected,

DCD), «источник данных готов» (Data Set Ready, DSR) и «индикатор вызова» (Ring Indicator, RI); а также выходные линии: «запрос на передачу» (Request to Send, RTS) и «приемник данных готов» (Data Terminal Ready, DTR).

Доступна возможность аппаратного управления потоком данных по линиям nUARTCTS и nUARTRTS.

Блок последовательного интерфейса инфракрасной (ИК) 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

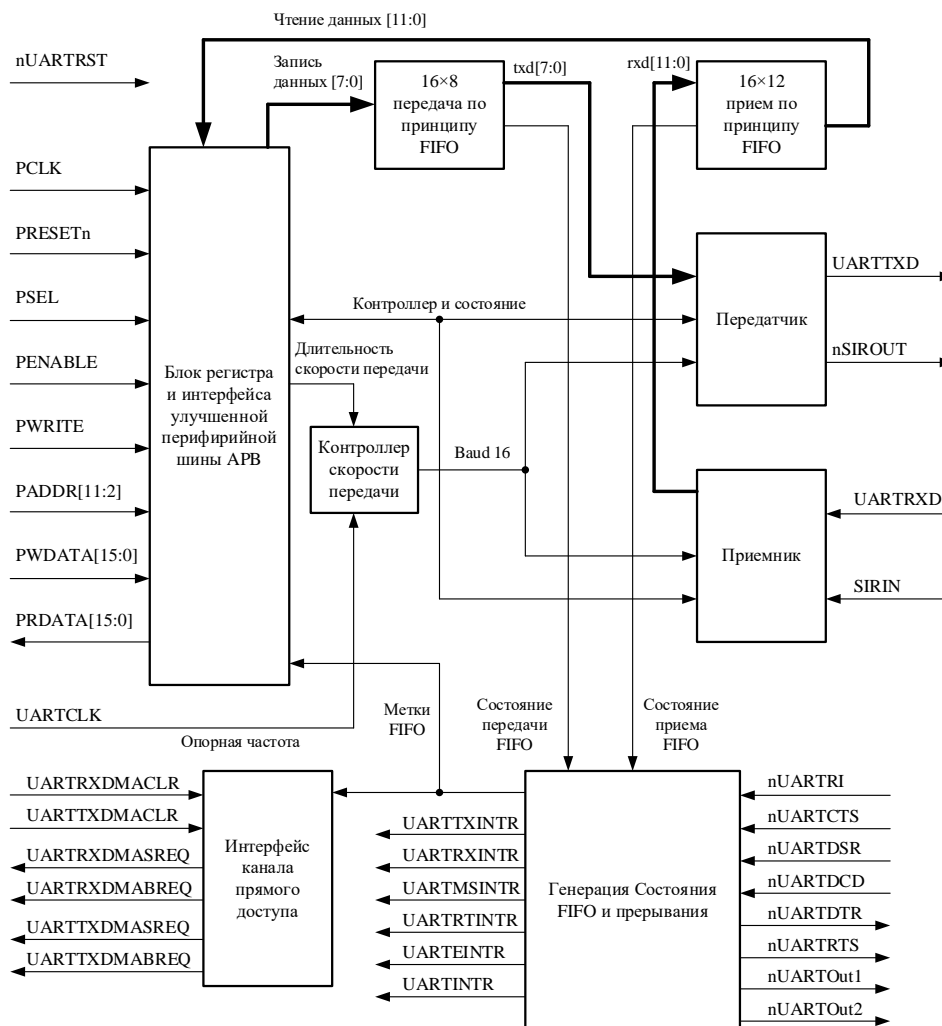


Рисунок 83 – Блок-схема универсального асинхронного приемопередатчика (УАПП)

Примечание – С целью обеспечения ясности на схеме не показаны схемы тестирования.

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, ошибки контроля четности, ошибки в структуре сигнала, а также разрыва 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 PCLK и UARTCLK. Регистры синхронизации и логика квитирования реализованы и 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 PCLK в область действия UARTCLK, так и наоборот.

22.8 Описание функционирования ИК кодека IrDA SIR

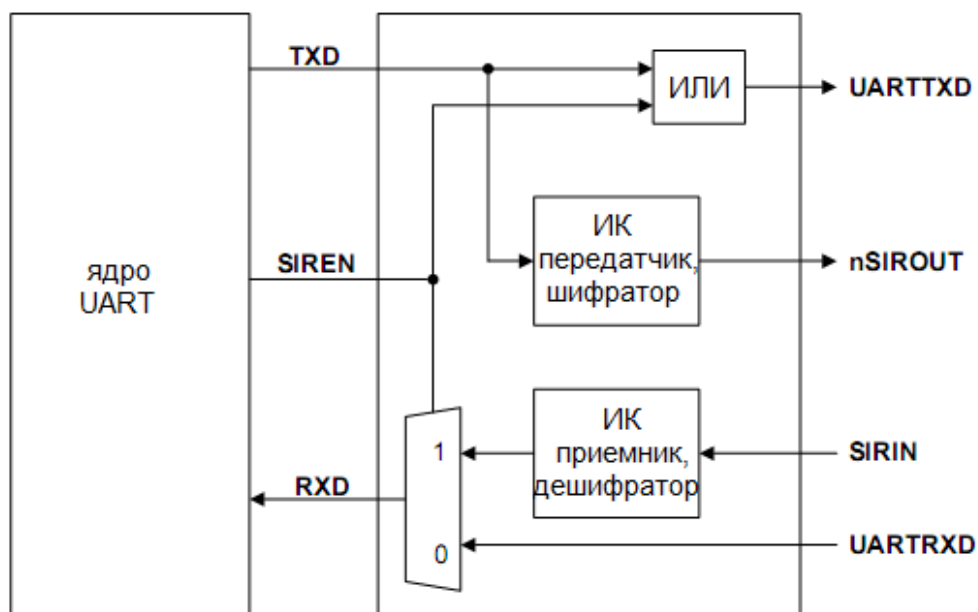


Рисунок 84 – Структурная схема кодека

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 $Baud_{16}$, 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 $IrLPBaud_{16}$ с номинальной частотой 1,8432 МГц, который, в свою очередь, формируется путем деления частоты $UARTCLK$. Значение частоты $IrLPBaud_{16}$ 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 приводит к увеличению джиттера фронтов импульсов данных. Наличие джиттера в случае использования дробных коэффициентов деления связано с тем, что интервалы между тактовыми импульсами $Baud_{16}$ будут нерегулярными – период сигнала $Baud_{16}$ 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

Сброс модуля

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса процессора. Значения регистров после сброса описаны в подразделе 22.12 «Программное управление модулем».

22.9.1 Тактовые сигналы

Частота, тактового сигнала UARTCLK должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

$$F_{UARTCLK}(min) \geq 16 \cdot baud_rate_max ;$$

$$F_{UARTCLK}(max) \leq 16 \cdot 65535 \cdot baud_rate_min .$$

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота UARTCLK должна находиться в интервале от 7,3728 МГц 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 PCLK и UARTCLK. Частота UARTCLK должна быть не более чем в 5/3 раз выше частоты PCLK.

$$F_{UARTCLK} \leq 5/3 \cdot F_{PCLK} .$$

Например, при работе в режиме UART с максимальной скоростью передачи данных 921600 бод, при частоте UARTCLK 14,7456 МГц, частота PCLK должна быть не менее 8,85276 МГц. Это гарантирует, что контроллер UART будет иметь достаточно времени для записи принятых данных в буфер FIFO.

22.9.2 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.

22.9.3 Дробный коэффициент деления

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:

$$\text{Коэффициент деления} = \text{UARTCLK} / (16 \cdot \text{скорость передачи данных}) = \text{BRD}_I + \text{BRD}_F,$$

где BRD_I – целая часть, а BRD_F – дробная часть коэффициента деления.

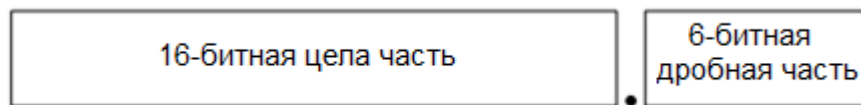


Рисунок 85 – Представление коэффициента деления для формирования скорости передачи данных

– Шестибитное значение, записываемое в регистр UARTFBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на 2^n , где n – ширина регистра UARTFBRD) и округления до ближайшего целого числа:

$$M = \text{integer}(\text{BRD}_F \cdot 2^n + 0,5),$$

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 больше требуемой скорости обмена данными.

22.9.4 Передача и прием данных

Принятые или передаваемые данные заносятся в 16-элементные буфера 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 приемо-передатчика 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 состояния в запрещенное.

Для каждого отсчета данных (в приемной линии) производится три измерения уровня, решение принимается по принципу большинства голосов. В следующих разделах будет определено понятие среднего отсчета, при этом в качестве дополнительных двух отсчетов берутся предыдущий и последующий – во всем остальном документ про это нет ни слова – я бы убрал этот текст.

В случае если приемник находился в неактивном состоянии (на линии входного сигнала UARTRXD 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 UARTRXD сохраняет низкий логический уровень в течение восьми отсчетов сигнала Baud16 с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

В случае если обнаружен достоверный стартовый бит, производится регистрация последовательности данных на входе приемника. Очередной бит данных фиксируются каждые 16 отсчетов тактового сигнала Baud16 (что соответствует длительности одного символа). Производится регистрация всех бит данных (согласно запрограммированным параметрам) и бита четности (если включен режим контроля четности).

Наконец, производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UARTRXD). В случае если последнее условие не выполняется, устанавливается признак ошибки формирования кадра. После того, как слово данных принято полностью, оно заносится в буфер FIFO приемника, наряду с четырьмя битами признаков ошибки, связанных с принятым.

22.9.5 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 332 описано назначение всех битов слова данных в FIFO буфере приемника.

22.9.6 Бит переполнения буфера

Бит переполнения непосредственно не связан с конкретным символом в буфере приемника. Признак переполнения фиксируется в случае, если буфер FIFO заполнен, в то время как 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 332 – Назначение бит 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

22.9.7 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.

22.9.8 Проверка по шлейфу

Проверка по шлейфу (замыкание выхода передатчика на вход приемника) выполняется путем установки в «1» бита LBE в регистре управления контроллером UARTCR.

22.9.9 Работа кодека ИК-обмена данными IrDA SIR

Кодек обеспечивает сопряжение асинхронного потока данных, сформированного приемопередатчиком, с полудуплексным последовательным интерфейсом IrDA SIR. Какая-либо аналоговая обработка сигнала при этом не выполняется. Назначение кодека – сформировать цифровой поток данных на вход приемника асинхронного сигнала и обработать цифровой поток данных с выхода передатчика.

Предусмотрено два режима работы:

1 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.

2 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:

$$1,42 \text{ МГц} < F_IrLPBaud16 < 2,12 \text{ МГц} .$$

22.9.10 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 UARTTTCR.

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.

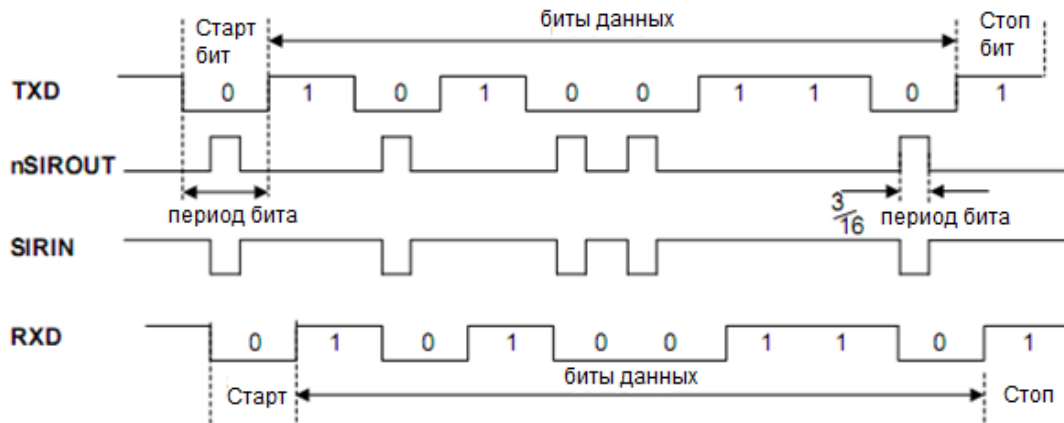


Рисунок 86 – Модуляция данных IrDA

22.9.11 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 87 показаны сигналы модема в режиме DTE. Назначение сигналов в режиме DCE представлено в таблице 333.

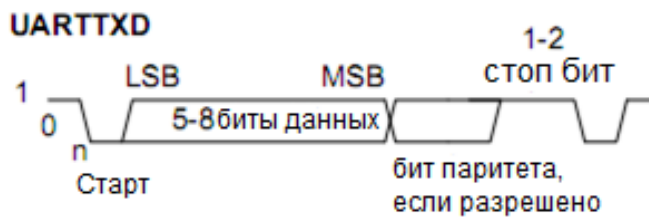


Рисунок 87 – Кадр передачи данных

Таблица 333 – 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

22.9.12 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 88.

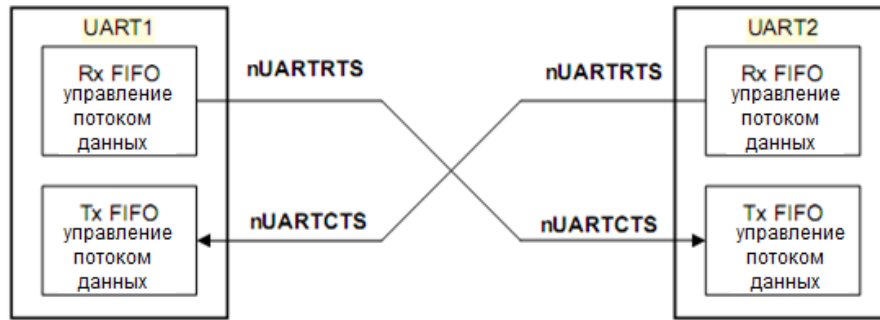


Рисунок 88 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

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 334 показаны необходимые установки для различных режимов управления потоком данных.

Таблица 334 – 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

22.9.13 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 его заполнение 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.

22.9.14 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, однако работа приемопередатчика UART разрешена, данные будут выдаваться до опустошения буфера FIFO передатчика.

22.10 Интерфейс прямого доступа к памяти

Модуль универсального асинхронного приемопередатчика оснащен интерфейсом подключения к контроллеру прямого доступа к памяти (далее – контроллер DMA). Работа в данном режиме контролируется регистром управления DMA UARTDMACR.

Интерфейс DMA включает в себя следующие сигналы.

Для приема:

– UARTRXDMASREQ – запрос передачи отдельного символа, инициируется контроллером UART. Размер символа в режиме приема данных – до 12 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит, по меньшей мере, один символ.

– UARTRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если заполнение буфера FIFO приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр UARTIFLS17).

– UARTRXDMACLR – сброс запроса на DMA, инициируется модулем приемопередатчика с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Для передачи:

– UARTRXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Размер символа в режиме передачи данных – до восьми бит. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку.

– UARTTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если заполнение буфера FIFO передатчика ниже заданного порога. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр UARTIFLS).

– UARTTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимно исключаящими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение, формируется как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае если количество данных в буфере приема меньше порогового значения формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов, а порог заполнения буфера FIFO установлен равным четырем. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

Примечание – Для оставшихся трех символов контроллер UART не может инициировать процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае установки в ноль бита управления DMA TXDMAE или RXDMAE в регистре управления DMA UARTDMACR.

В случае запрета буферов FIFO устройство способно передавать и принимать только одиночные символы, как следствие, контроллер может инициировать DMA только в одноэлементном режиме. При этом модуль в состоянии формировать только сигналы управления DMA UARTRXDMASREQ и UARTTXDMASREQ. Для информации о запрете буферов FIFO см. описание регистра управления линией UARTLCR_H.

Когда буферы FIFO включены, обмен данными может производиться в ходе как одноэлементных, так и блочных передач данных, в зависимости от установленной величины порога заполнения буферов и их фактического заполнения. В таблице 335 приведены значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

Таблица 335 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/8	28	4
1/4	24	8
1/2	16	16
3/4	8	24
7/8	4	28

В регистре управления DMA UARTDMACR предусмотрен бит DMAONERR, который позволяет запретить DMA от приемника в случае активного состояния линии прерывания по обнаружению ошибка UARTEINTR. При этом соответствующие линии запроса DMA – UARTRXDMASREQ и UARTRXDMABREQ переводятся в неактивное состояние (маскируются) до сброса UARTEINTR. На линии запроса DMA, обслуживающие передатчик, состояние UARTEINTR не влияет.

На рисунке 89 показаны временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK. В интересах ясности изложения предполагается, что синхронизация сигналов запроса DMA в контроллере DMA не производится.

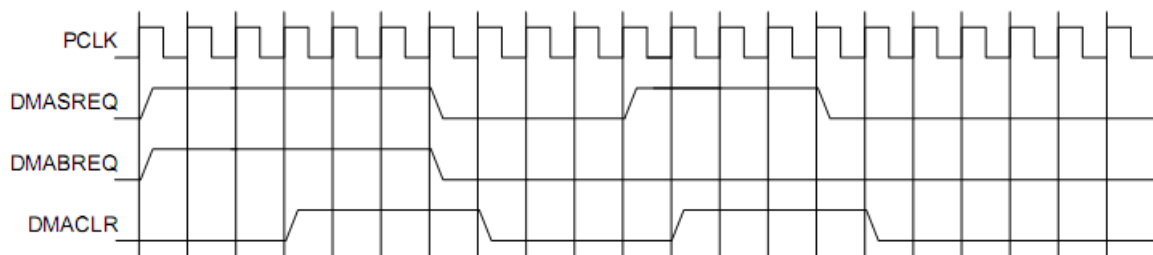


Рисунок 89 – Одноэлементный и блочный запросы DMA

22.11 Прерывания

В модуле предусмотрено 11 маскируемых источников прерывания. Их комбинации формируют пять независимых выходных сигналов запроса на прерывание, а также один общий сигнал, представляющий собой комбинацию независимых по схеме ИЛИ.

Сигналы запроса на прерывание:

- UARTRXINTR – прерывание от приемника;
- UARTRXINTR – прерывание от передатчика;
- UARTRTINTR – прерывание по таймауту приемника;
- UARTRMSINTR – прерывание по состоянию модема:
 - UARTRRIINTR, изменение состояния линии nUARTRRI;
 - UARTRCTSINTR, изменение состояния линии nUARTRCTS;
 - UARTRDCDINTR, изменение состояния линии nUARTRDCD;
 - UARTRDSRINTR, изменение состояния линии nUARTRDSR.

- UARTEINTR – ошибка:
 - UARTOEINTR, переполнение буфера;
 - UARTBEINTR, прерывание приема – разрыв линии;
 - UARTPEINTR, ошибка контроля четности;
 - UARTFEINTR, ошибка в структуре кадра.
- UARTINTR – логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски UARTIMSC. 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 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, согласованной с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Прерывание по обнаружению ошибки UARTEINTR формируется в случае возникновения той или иной ошибки приема данных. Предусмотрен ряд условий формирования признака ошибки.

Прерывание по состоянию модема представляет собой комбинацию признаков изменения отдельных линий состояния модема.

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:

1 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.

2 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:

1 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.

2 Буфер FIFO запрещен (имеет размер один символ), данные в буферном регистре передатчика отсутствуют. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи одного байта данных, либо после сброса прерывания.

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

Примечание – Прерывание передатчика основано на переходе через пороговое значение, а не на состоянии заполненности буфера FIFO передатчика относительно порогового значения. В случае если модуль и прерывания от него разрешены до осуществления записи данных в буфер FIFO передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера FIFO.

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.

22.11.5 UARTEINTR

Прерывание по обнаружению ошибки возникает в случае ошибки при приеме данных. Оно может быть вызвано рядом факторов:

- ошибка в структуре кадра;
- ошибка контроля четности;
- разрыв линии;
- переполнение буфера.

Причину возникновения прерывания можно определить, прочитав содержимое регистра прерываний UARTRIS, либо маскированного регистра прерываний UARTMIS.

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, UARTTXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

22.12 Программное управление модулем

22.12.1 Общая информация

Следующая информация применима ко всем регистрам контроллера.

Базовый адрес контроллера не фиксирован и может быть различным в разных системах. 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 336 определяет режим доступа к регистру в соответствии с обозначениями:

- RW – чтение и запись;
- RO – только чтение;
- WO – только запись.

22.12.2 Обобщенные данные о регистрах контроллеров UART

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в таблице 336.

Таблица 336 – 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	UARTDR	RW	0x---	12/8	Регистр данных
0x004	UARTSR/ UARTECR	RW	0x0	4/0	Регистр состояния приемника / Сброс ошибки приемника
0x008-0x014					Резерв
0x018	UARTFR	RO	0b-10010---	9	Регистр флагов
0x01C					Резерв
0x020	UARTILPR	RW	0x00	8	Регистр управления ИК-обменом в режиме пониженного энергопотребления
0x024	UARTIBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	UARTFBRD	RW	0x00	6	Дробная часть делителя скорости обмена данными
0x02C	UARTLCR_H	RW	0x00	8	Регистр управления линией
0x030	UARTCR	RW	0x0300	16	Регистр управления
0x034	UARTIFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	UARTIMSC	RW	0x000	11	Регистр маски прерывания
0x03C	UARTRIS	RO	0x00-	11	Регистр состояния прерываний
0x040	UARTMIS	RO	0x00-	11	Регистр состояния прерываний с маскированием
0x044	UARTICR	WO	-	11	Регистр сброса прерывания
0x048	UARTDMACR	RW	0x00	3	Регистр управления DMA

22.12.2.1 UARTx_DR

Регистр данных.

В ходе передаче данных:

Если буфер FIFO передатчика разрешен, то слово данных, записанное в рассматриваемый регистр, направляется в буфер FIFO передатчика. В противном случае, записанное слово фиксируется в буферный регистр передатчика (последний элемент буфера FIFO).

Операция записи в регистр инициирует передачу данных. Слово данных предваряется стартовым битом, дополняется битом контроля четности (если режим

контроля четности включен) и стоповым битом. Сформированное слово отправляется в линию передачи данных.

В ходе приема данных:

Если буфер FIFO приемника разрешен, байт данных и четыре бита состояния (разрыв, ошибка формирования кадра, четность, переполнение) сохраняются в 12-битном буфере. В противном случае байт данных и биты состояния записываются в буферный регистр (последний элемент буфера FIFO).

Полученные из линии связи байты данных считываются путем чтения из регистра UARTDR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра UARTRSR/UARTECR (см. таблицу 338).

Таблица 337 – Формат регистра UARTDR

Разряды	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
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.

22.12.2.2 UARTx_RSR_ECR

Регистр состояния приемника / сброса ошибки.

Состояние приемника также может быть считано из регистра UARTRSR. 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 UARTDR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним считанным из регистра UARTDR байтом данных).

Запись в регистр UARTECR приводит к сбросу признаков ошибок переполнения, четности, структуры кадра, разрыва линии. Кроме того, все эти признаки устанавливаются в 0 после сброса устройства.

В таблице 338 представлено назначение бит регистра UARTRSR/UARTECR.

Таблица 338 – Регистр UARTRSR/UARTECR

Разряды	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 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 UARTLCR_H. Бит сбрасывается в 0 после записи в регистр UARTECR. 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

Разряды	Наименование	Назначение
		записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера

Примечание – Перед чтением регистра состояния UARTRSR необходимо считать данные, принятые из линии, путем обращения к регистру данных UARTDR. Противоположная последовательность действий не допускается, так как регистр UARTRSR обновляет свое состояние только после чтения регистра UARTDR. Вместе с тем, информация о состоянии приемника может быть получена непосредственно из регистра данных UARTDR.

22.12.2.3 UARTx_FR

Регистр флагов.

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. В таблице 339 представлена информация о назначении битов регистра.

Таблица 339 – Регистр UARTFR

Разряды	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 UARTLCR_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 UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника занят. В противном случае он равен 1, если заполнен буфер FIFO приемника
5	TXFF	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_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 UARTLCR_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

Разряды	Наименование	Назначение
		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

22.12.2.4 *UARTx_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 340. 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 1,41-2,11мкс (в три раза длиннее периода сигнала IrLPBaud16).

Таблица 340 – Регистр UARTILPR

Разряды	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.

22.12.2.5 *UARTx_IBRD*

Регистр целой части делителя скорости передачи данных.

Назначение бит регистра представлено в таблице 341.

Таблица 341 – Регистр UARTBIRD

Разряды	Наименование	Назначение
15...0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

22.12.2.6 *UARTx_FBRD*

Регистр дробной части делителя скорости передачи данных.

Назначение бит регистра представлено в таблице 342.

Таблица 342 – Регистр UARTBFRD

Разряды	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 UARTIBRD и UARTFBRD 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 UARTIBRD, равное 0 является недопустимым, при этом значение регистра UARTFBRD игнорируется;

3 Аналогично, при UARTIBRD, равном 65535 (0xFFFF), значение UARTFBRD 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 UARTBFRD, равно

$$m = \text{integer}((0,085 \cdot 64) + 0,5) = 5.$$

$$\text{Реальное значение коэффициента деления} = 1 + 5/64 = 1,078.$$

$$\text{Реальная скорость передачи данных} = (4 \cdot 10^6) / (16 \cdot 1,078) = 231911 \text{ бит/с.}$$

$$\text{Ошибка установки скорости} = (231911 - 230400) / 230400 \cdot 100 \% = 0,656 \%.$$

Максимальная ошибка установки скорости передачи данных с использованием шестизрядного регистра UARTBFRD = $1/64 \cdot 100 \% = 1,56 \%$. Такая ошибка возникает в случае $m = 1$, при этом разница накапливается в течение 64 тактовых интервалов.

В таблице 343 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 UARTFBRD должен быть записан ноль.

Таблица 343 – 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 344 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц.

Таблица 344 – 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

22.12.2.7 *UARTx_LCR_H*

Регистр управления линией.

Данный регистр обеспечивает доступ к разрядам с 29 по 22 регистра UARTLCR. При сбросе все биты регистра UARTLCR_H обнуляются.

Назначение разрядов регистра описано таблице 345.

Таблица 345 – Регистр UARTLCR_H

Разряды	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 UARTLCR_H, UARTIBRD и UARTFBRD совместно образует общий 30-разрядный регистр UARTLCR, который обновляется по стробу, формируемому при записи в UARTLCR_H. Таким образом, для того чтобы изменение параметров коэффициента деления частоты обмена данными вступило в силу, после их

изменения значения регистров UARTIBRD и/или UARTFBRD необходимо осуществить запись данных в регистр UARTLCR_H.

Примечания:

1 Изменение значений трех регистров можно осуществить корректно двумя способами:

– Запись UARTIBRD, запись UARTFBRD, запись UARTLCR_H;

– Запись UARTFBRD, запись UARTIBRD, запись UARTLCR_H.

2 Для того чтобы изменить значение лишь одного из регистров (UARTIBRD или UARTFBRD) необходимо выполнить следующие шаги:

– Запись UARTIBRD (или UARTFBRD), запись UARTLCR_H.

В таблице 346 приведена таблица истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией UARTLCR_H.

Таблица 346 – 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 UARTLCR_H, UARTIBRD, and UARTFBRD 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.

22.12.2.8 UARTx_CR

Регистр управления.

После сброса все биты регистра управления, за исключением бит 9 и 8, устанавливаются в нулевое состояние. Биты 9 и 8 устанавливаются в единичное состояние.

Назначение разрядов регистра управления показано в таблице 347.

Таблица 347 – Регистр управления UARTCR

Разряды	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

Разряды	Наименование	Назначение
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 UARTTCR (стр. 4-5) 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
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

Разряды	Наименование	Назначение
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

Примечания

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;

2 Рекомендуется следующая последовательность действий для программирования регистров управления:

- Остановите работу приемопередатчика;
- Дождитесь окончания приема и/или передачи текущего символа данных;
- Сбросьте буфер передатчика путем установки бита FEN регистра UARTLCR_N в 0;
- Измените настройки регистра UARTCR;
- Возобновите работу приемопередатчика

22.12.2.9 UARTx_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 таблицу 348.

Таблица 348 – Регистр UARTIFLS

Разряды	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

Разряды	Наименование	Назначение
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

22.12.2.10 UARTIMSC

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 UARTIMSC показано в таблице 349.

Таблица 349 – Регистр UARTIMSC

Разряды	Наименование	Назначение
15...11		Резерв. Не модифицируйте. При чтении выдаются нули
10	OEIM	Маска прерывания по переполнению буфера UARTOEINTR: 1 – установлена; 0 – сброшена
9	BEIM	Маска прерывания по разрыву линии UARTBEINTR: 1 – установлена; 0 – сброшена
8	PEIM	Маска прерывания по ошибке контроля четности UARTPEINTR: 1 – установлена; 0 – сброшена
7	FEIM	Маска прерывания по ошибке в структуре кадра UARTFEINTR: 1 – установлена; 0 – сброшена
6	RTIM	Маска прерывания по таймауту приема данных UARTRTINTR: 1 – установлена; 0 – сброшена
5	TXIM	Маска прерывания от передатчика UARTTXINTR: 1 – установлена; 0 – сброшена
4	RXIM	Маска прерывания от приемника UARTRXINTR: 1 – установлена; 0 – сброшена
3	DSRMIM	Маска прерывания UARTDSRINTR по изменению состояния линии nUARTDSR: 1 – установлена; 0 – сброшена
2	DCDMIM	Маска прерывания UARTDCDINTR по изменению состояния линии nUARTDCD: 1 – установлена; 0 – сброшена

Разряды	Наименование	Назначение
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

22.12.2.11 UARTx_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 бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре UARTRIS представлено в таблице 350.

Таблица 350 – Регистр UARTRIS

Разряды	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 UARTRIS. 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 UARTRIS и UARTRIS приводит к одинаковым результатам
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

22.12.2.12 UARTx_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 бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре UARTMIS представлено в таблице 351.

Таблица 351 – Регистр UARTMIS

Разряды	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

22.12.2.13 UARTx_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 UARTICR представлено в таблице 352.

Таблица 352 – Регистр UARTICR

Разряды	Наименование	Назначение
15...11		Резерв. Не модифицируйте. При чтении выдаются нули
10	OEIC	Сброс прерывания по переполнению буфера UARTOEINTR

Разряды	Наименование	Назначение
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

22.12.2.14 UARTx_DMAR

Регистр управления прямым доступом к памяти.

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Назначение бит регистра UARTDMAR представлено в таблице 353.

Таблица 353 – Регистр UARTDMAR

Разряды	Наименование	Назначение
15...13		Резерв. Не модифицируйте. При чтении выдаются нули
2	DMAONERR	Если бит установлен в 1, в случае возникновения прерывания по обнаружению ошибки блокируются запросы DMA от приемника UARTRXDMASREQ и UARTRXDMABREQ
1	TXDMAE	Использование DMA при передаче. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика
0	RXDMAE	Использование DMA при приеме. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO приемника

23 Контроллер прямого доступа в память DMA

23.1 Основные свойства контроллера DMA

Основные свойства и отличительные особенности:

- настраиваемое количество каналов DMA;
- каждый канал DMA имеет свои сигналы управления передачи данных;
- каждый канал DMA имеет программируемый уровень приоритета;
- каждый уровень приоритета обрабатывается исходя из уровня приоритета, определяемого номером канала DMA;
- поддержка различного типа передачи данных:
- память – память;
- память – периферия;
- периферия – память;
- поддержка различных типов DMA циклов;
- поддержка передачи данных различной разрядности;
- каждому каналу DMA доступна первичная и альтернативная структура управляющих данных канала;
 - все управляющие данные канала хранятся в системной памяти в формате «первый – младший значащий разряд»;
 - все данные передаются с использованием одиночных АНВ-Lite пакетов;
 - разрядность данных приемника равна разрядности данных передатчика;
 - количество передач в одном цикле DMA может программироваться от 1 до 1024;
 - инкремент адреса передачи может быть больше чем разрядность данных;
 - наличие выходного сигнала состояния ошибки на шине АНВ.

23.2 Термины и определения

При описании блока используются следующие термины:

Альтернативная	Альтернативная структура управляющих данных канала. Вы можете установить соответствующий регистр для изменения типа структуры данных (см. пункт 23.3.18 «Структура управляющих данных канала»)
C	Идентификатор номера канала прямого доступа. Например: C=1 – канал DMA 1 C=23 – канал DMA 23
Канал	Возможны конфигурации контроллера с числом каналов до 32. Каждый канал содержит независимые сигналы управления передачей данных, которые могут инициировать передачу данных по каналу DMA
Управляющие данные канала	Структура данных находится в системной памяти. Вы можете запрограммировать эту структуру данных так, что контроллер может выполнять передачу данных по каналу DMA в желаемом режиме. Контроллер должен иметь доступ к области системной памяти, где находится эта информация.

	Примечание – Любое упоминание в спецификации структуры данных означает управляющие данные канала
Цикл DMA	Все передачи DMA, которые контроллер должен выполнить для передачи N пакетов данных
Передача DMA	Акция пересылки одного байта, полуслова или слова. Общее количество передач DMA, которые контроллер выполняет для канала
Пинг – понг	Режим работы для выбранного канала, при котором контроллер получает начальный запрос и затем выполняет цикл DMA, используя первичную или альтернативную структуру данных. После завершения этого цикла DMA контроллер начинает выполнять новый цикл DMA, используя другую (первичную или альтернативную) структуру данных. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя главному процессору перенастраивать неактивную структуру данных. Контроллер продолжает переключаться от первичной к альтернативной структуре данных и обратно до тех пор, пока он не прочитает «неправильную» структуру данных или пока он не завершит цикл без переключения к другой структуре
Первичная	Первичная структура управляющих данных канала. Контроллер использует эту структуру данных, если соответствующий разряд в регистре <code>chnl_pri_alt_set</code> установлен в 0
R	Степень числа 2, устанавливающая число передач DMA, которые могут произойти перед сменой арбитража. Количество передач DMA программируется в диапазоне от 1 до 1024 двоичными шагами от 2 в степени 0 до 2 в степени 100
Исполнение с изменением конфигурации	Режим работы для выбранного канала, при котором контроллер получает запрос от периферии и выполняет 4 DMA передачи, используя первичную структуру управляющих данных, которые настраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. После того, как цикл закончится и если периферия устанавливает новый запрос на обслуживание, контроллер выполняет снова 4 DMA передачи, используя первичную структуру управляющих данных, которые опять перенастраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. Контроллер будет продолжать работать вышеописанным способом до тех пор, пока не прочитает неправильную структуру данных или процессор не установит альтернативную структуру данных для обычного цикла. Контроллер устанавливает флаг <code>dma_done</code> , если окончание подобного режима работы происходит после выполнения обычного цикла

23.3 Функциональное описание

На рисунке 90 показана упрощенная структурная схема контроллера.

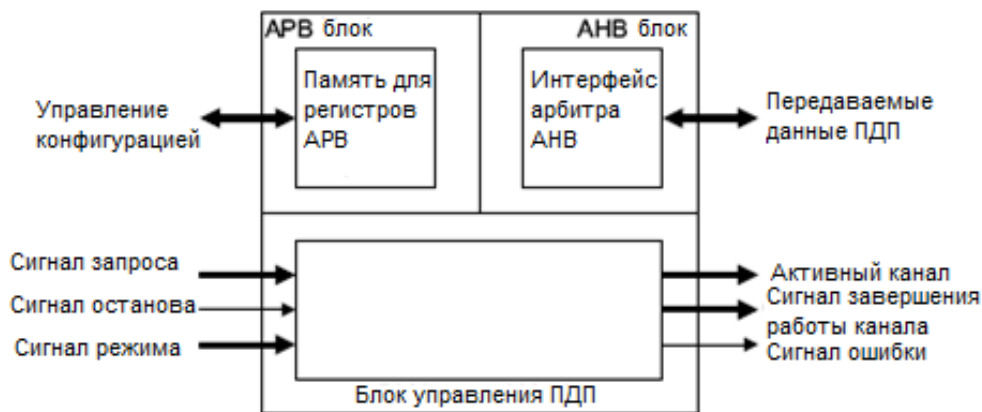


Рисунок 90 – Структурная схема контроллера

Контроллер состоит из следующих основных функциональных блоков:

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

23.3.1 Распределение каналов DMA

Таблица 354 – Таблица запросов dma_req и dma_sreq от периферийных блоков

Номер запроса	dma_req	dma_sreq	Описание
0	UART0TXDMABREQ	UART0TXDMASREQ	Запрос от передатчика UART0
1	UART0RXDMABREQ	UART0RXDMASREQ	Запрос от приёмника UART0
2	UART1TXDMABREQ	UART1TXDMASREQ	Запрос от передатчика UART1
3	UART1RXDMABREQ	UART1RXDMASREQ	Запрос от приёмника UART1
4	SSPTXDMABREQ	SSPTXDMASREQ	Запрос от передатчика SPI
5	SSPRXDMABREQ	SSPRXDMASREQ	Запрос от приёмника SPI
6	CRCDMABREQ	CRCDMASREQ	Запрос от аппаратного блока вычисления CRC
7...9	-	-	-
10	TMR0DMAREQ	TMR0DMAREQ	Запрос от таймера общего назначения TIMER0
11	TMR1DMAREQ	TMR1DMAREQ	Запрос от таймера общего назначения TIMER1
12	ADCIUDMABREQ1 (I0)	ADCIUDMASREQ1 (I0)	Запросы от блока АЦП для измерения напряжений и токов в электрической сети
13	ADCIUDMABREQ2 (V0)	ADCIUDMASREQ2 (V0)	
14	ADCIUDMABREQ3 (I1)	ADCIUDMASREQ3 (I1)	
15	ADCIUDMABREQ4 (V1)	ADCIUDMASREQ4 (V1)	
16	ADCIUDMABREQ5 (I2)	ADCIUDMASREQ5 (I2)	
17	ADCIUDMABREQ6 (V2)	ADCIUDMASREQ6 (V2)	
18	ADCIUDMABREQ7 (I3)	ADCIUDMASREQ7 (I3)	
19...29	-	-	-

Номер запроса	dma_req	dma_sreq	Описание
30	-	ADCDMASREQ	Запрос от АЦП последовательных приближений
31	-	-	-

23.3.2 Блок, подключенный к шине APB

Блок содержит набор регистров, позволяющих настраивать контроллер, используя ведомый APB интерфейс. Регистры занимают адресное пространство емкостью 4 Кбайт.

23.3.3 Блок, подключенный к шине АНВ

Контроллер содержит один блок типа «ведущий» шины АНВ-Lite, который позволяет, используя 32-разрядную шину, передавать данные от источника к приемнику. Источник и приемник являются ведомыми шины АНВ. Контроллер соответствует протоколу АМВА 3 АНВ-Lite. Подробное описание протокола приведено в документе «Описание протокола АМВА 3 АНВ-Lite v1.0».

23.3.4 Управляющий блок DMA

Этот блок содержит схему управления, позволяющую реализовать следующие функции:

- осуществление арбитража поступающих запросов;
- индикацию активного канала;
- индикацию завершения обмена по каналу;
- индикацию состояния ошибки обмена по интерфейсу АНВ-Lite;
- разрешение медленным устройствам приостанавливать исполнение цикла DMA;
- ожидание запроса на очистку до завершения цикла DMA;
- осуществление одиночных или множественных передач DMA для каждого запроса;
- осуществление следующих типов DMA передач:
 - память – память;
 - память – периферия;
 - периферия – память.

Примечание – Передачи типа периферия – периферия не поддерживаются, так как каждый канал имеет один интерфейс запроса DMA.

На рисунке 91 показаны сигналы управления интерфейса DMA.

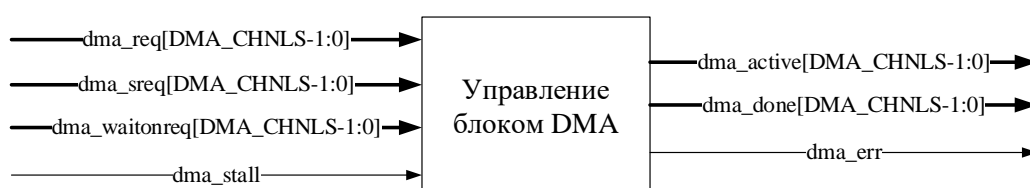


Рисунок 91 – Сигналы интерфейса управления DMA

23.3.5 Пример использования блока DMA

На рисунке 92 показана примерная структурная схема микросхемы с использованием контроллера DMA.

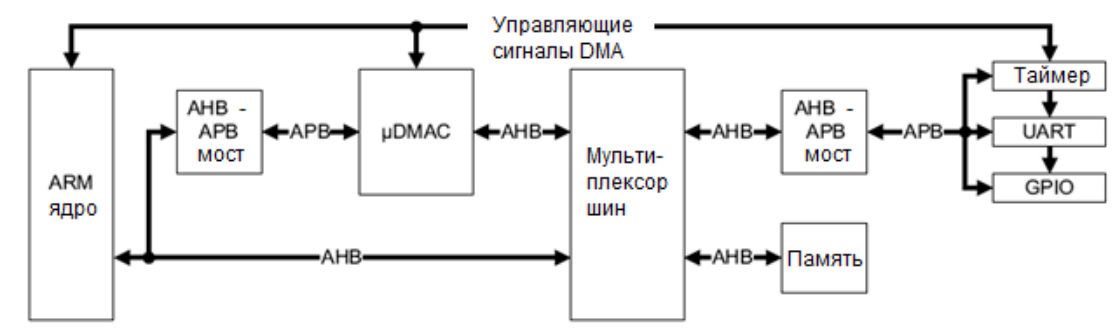


Рисунок 92 – Пример структурной схемы микросхемы с использованием контроллера DMA

23.3.6 Типы передач

Контроллер интерфейса не поддерживает пакетные передачи и поэтому сигнал NBURST удерживается всегда в состоянии логического нуля. Контроллер выполняет одиночные передачи, в соответствии с протоколом AHB-Lite, используя сигналы комбинации шины HTRANS, перечисленные в таблице 355.

Таблица 355 – Комбинации шины HTRANS

HTRANS[1]	HTRANS[0]*	Значение
0	0	IDLE (нет передачи)
1	0	NONSEQ (одиночная передача)

* Сигнал постоянно удерживается в состоянии логического «0»

Отсутствие возможности осуществлять пакетные передачи имеет минимальное влияние на производительность системы, так как пакетные передачи более эффективны в одноуровневых системах с шиной AHB, где блоки должны «захватывать» шину или обращаться к внешней памяти. В тоже время контроллер DMA предназначен для использования в многоуровневых системах с шиной AHB-Lite, включающих встроенную память.

23.3.7 Разрядность передач данных

Контроллер интерфейса предоставляет возможность осуществлять передачу 8, 16 и 32 разрядных данных. Таблица 356 перечисляет значение комбинаций шины HSIZE.

Таблица 356 – Комбинации шины HSIZE

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

* Сигнал постоянно удерживается в состоянии логического «0»;
 ** Запрещенная комбинация

Контроллер всегда использует передачи 32-разрядными данными при обращении к управляющим данным канала. Необходимо устанавливать разрядность данных источника соответствующую разрядности данных приемника.

23.3.8 Управление защитой данных

Контроллер позволяет устанавливать режимы защиты данных протокола АНВ-Lite, определяемые шиной HPROT[3:1]. Возможен выбор следующих режимов защиты:

- кэширование;
- буферизация;
- привилегированный.

Таблица 357 перечисляет значения комбинаций шины HPROT.

Таблица 357 – Режимы защиты данных

HPROT[3] Кэширование	HPROT[2] Буферизация	HPROT[1] Привилегированный	HPROT[0] Данные/ команда	Описание
-	-	-	1*	Доступ к данным
-	-	0	-	Пользовательский доступ
-	-	1	-	Привилегированный доступ
-	0	-	-	Без буферизации
-	1	-	-	Буферизованный
0	-	-	-	Без кэширования
1	-	-	-	Кэшированный

Контроллер удерживает HPROT[0] в состоянии логической «1», чтобы обозначить доступ к данным.

Для каждого цикла DMA возможен выбор режимов защиты данных передач источника и приемника. Более подробно это описано в пункте 23.3.18 «Структура управляющих данных канала».

Для каждого канала DMA также возможен выбор режима защиты данных. Более подробно это описано в разделе Управление DMA.

23.3.9 Инкремент адреса.

Контроллер позволяет управлять инкрементом адреса при чтении данных из источника и при записи данных в приемник. Инкремент адреса производится в зависимости от разрядности передаваемых данных. В таблице 358 перечислены возможные комбинации.

Таблица 358 – Инкремент адреса

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

Минимальная величина инкремента адреса всегда соответствует разрядности передаваемых данных. Максимальная величина инкремента адреса, осуществляемая контроллером - одно слово. Более подробно о настройке инкремента адреса написано в пункте 23.3.18 «Структура управляющих данных канала». Этот раздел описывает разряды управления величиной инкремента адреса в управляющих данных канала.

Примечание – Если необходимо оставлять адрес неизменным при чтении или записи данных, для примера, при работе с FIFO, можно соответствующим образом настроить контроллер на работу с фиксированным адресом (см. пункт 23.3.18 «Структура управляющих данных канала».).

23.3.10 Управление DMA

Раздел описывает:

- правила обмена данными;
- диаграммы работы контроллера DMA;
- правила арбитража DMA;
- приоритет;
- типы циклов DMA;
- индикация ошибок.

23.3.11 Правила обмена данными.

Контроллер использует правила обмена данными перечисленные в таблице 359 при соблюдении следующих условий:

- канал DMA включен, что выполняется установкой в состояние логической единицы разрядов управления `chnl_enable_set[C]` и `master_enable`;
- флаги запроса `dma_req[C]` и `dma_sreq[C]` не замаскированы, что выполняется установкой в состояние логического нуля разряда управления `chnl_req_mask_set [C]`;
- контроллер находится не в тестовом режиме, что выполняется установкой в состояние логического нуля разряда управления `int_test_en bit[C]`.

Таблица 359 – Правила, при которых передача данных по каналам разрешена и запросы не маскируются

Правило	Описание
1	Если <code>dma_active[C]</code> установлен в 0, то установка в 1 <code>dma_req[C]</code> или <code>dma_sreq[C]</code> на один или более тактов сигнала <code>hclk</code> , следующих или не следующих друг за другом, инициирует передачу по каналу номер C
2	Контроллер осуществляет установку в 1 только одного разряда <code>dma_active[C]</code>
3	Контроллер устанавливает в 1 <code>dma_active[C]</code> в момент начала передачи по каналу C
4	Для типов циклов DMA отличных от периферийного «Исполнение с изменением конфигурации», <code>dma_active[C]</code> остается в состоянии 1 до тех пор, пока контроллер не окончит передачи с номерами меньше, чем значение 2^R или число передач, указанное в регистре <code>n_minus_1</code> . В периферийном режиме «Исполнение с изменением конфигурации», <code>dma_active[C]</code> остается в состоянии 1 в течение каждой пары DMA передач, с использованием первичной и альтернативной структур управляющих данных. Так, что контроллер выполняет 2^R передач, используя первичную структуру

Правило	Описание
	управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или число передач, указанное в регистре <code>n_minus_1</code>) передач, используя альтернативную структуру управляющих данных. По окончании последней передачи <code>dma_active[C]</code> сбрасывается в 0
5	Контроллер устанавливает <code>dma_active[C]</code> в 0 на, как минимум, один такт сигнала <code>hclk</code> , перед тем как снова установит <code>dma_active[C]</code> или <code>dma_active[]</code> в 1
6	Для каналов, по которым разрешена передача, контроллер осуществляет установку в 1 только одного <code>dma_done[]</code>
7	Если <code>dma_req[C]</code> устанавливается в состояние 1 в момент, когда <code>dma_active[C]</code> или <code>dma_stall</code> также в состоянии 1, то это означает, что контроллер обнаружил запрос
8	Если разряды <code>cycle_ctrl</code> для канала установлены в состояние <code>3'b100</code> , <code>3'b101</code> , <code>3'b110</code> , <code>3'b111</code> , то <code>dma_done[C]</code> никогда не будет установлен в 1
9	<p>Если все передачи по каналу завершены, и разряды <code>cycle_ctrl</code> позволяют удержание <code>dma_done[C]</code>, то по срезу сигнала <code>dma_active[]</code> произойдут события:</p> <ul style="list-style-type: none"> – если <code>dma_stall</code> в состоянии 0, контроллер устанавливает <code>dma_done[]</code> в состояние 1 продолжительностью один такт <code>hclk</code>; – если <code>dma_stall</code> в состоянии 1 работа контроллера приостановлена. После того, как <code>dma_stall</code> будет установлен в 0, контроллер устанавливает <code>dma_done[]</code> в состояние 1 продолжительностью один такт <code>hclk</code>
10	Состояние <code>dma_waitonreq[C]</code> можно изменять только при выключенном канале
11	<p>Если <code>dma_waitonreq[C]</code> в состоянии 1, то сигнал <code>dma_active[C]</code> не перейдет в состояние 0 до тех пор, пока:</p> <ul style="list-style-type: none"> – контроллер завершит 2^R передач (или число передач, указанное в регистре <code>n_minus_1</code>); – <code>dma_req[C]</code> будет установлен в 0; – <code>dma_sreq[C]</code> будет установлен в 0
12	<p>Если за один такт сигнала <code>hclk</code> перед установкой <code>dma_active[C]</code> в 0, устанавливается в 1 <code>dma_stall</code>, то:</p> <ul style="list-style-type: none"> – контроллер установит <code>dma_active[C]</code> в 0 на следующем такте сигнала <code>hclk</code>; – передача по каналу C не завершится пока не будет сброшен в 0 <code>dma_stall</code>
13	Контроллер игнорирует <code>dma_sreq[C]</code> если <code>dma_waitonreq[C]</code> в состоянии 0
14	Контроллер игнорирует <code>dma_sreq[C]</code> если <code>chnl_useburst_set[C]</code> в состоянии 1*
15	<p>Для типов циклов DMA, отличных от периферийного «Исполнение с изменением конфигурации» по окончании 2^R передач, контроллер устанавливает значение <code>chnl_useburst_set[C]</code> в состояние 0, если количество оставшихся передач меньше, чем 2^R</p> <p>В периферийном режиме «Исполнение с изменением конфигурации», контроллер устанавливает значение <code>chnl_useburst_set[C]</code> в состояние 0 только, если количество оставшихся передач с использованием альтернативной структуры управляющих данных меньше, чем 2^R</p>
16	<p>Для типов циклов DMA, отличных от периферийного «Исполнение с изменением конфигурации», если за один такт <code>hclk</code> до установки <code>dma_active[C]</code> в 1 <code>dma_sreq[C]</code> и <code>dma_waitonreq[C]</code> установлены в 1 и <code>dma_req[C]</code> установлен в 0, то контроллер выполняет одну DMA передачу.</p> <p>В периферийном режиме «Исполнение с изменением конфигурации», если за один такт <code>hclk</code> до установки <code>dma_active[C]</code> в 1 <code>dma_sreq[C]</code> и <code>dma_waitonreq[C]</code> установлены в 1 и <code>dma_req[C]</code> установлен в 0, то контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных, затем без</p>

Правило	Описание
	осуществления арбитража выполняет одну передачу, используя альтернативную структуру управляющих данных
17	Для типов циклов DMA отличных от периферийного «Исполнение с изменением конфигурации», если за один такт <code>hclk</code> до установки <code>dma_active[C]</code> в 1, <code>dma_sreq[C]</code> и <code>dma_req[C]</code> установлены в 1, то приоритет предоставляется <code>dma_req[c]</code> и контроллер выполняет 2^R (или число передач, указанное в регистре <code>n_minus_1</code>) DMA передач. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт <code>hclk</code> до установки <code>dma_active[C]</code> в 1, <code>dma_sreq[C]</code> и <code>dma_req[C]</code> установлены в 1, то приоритет предоставляется <code>dma_req[c]</code> и контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или число передач, указанное в регистре <code>n_minus_1</code>), используя альтернативную структуру управляющих данных
18	Когда <code>chnl_req_mask_set[C]</code> установлен в 1, контроллер игнорирует запросы по <code>dma_sreq[C]</code> и <code>dma_req[C]</code>
* Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре <code>n_minus_1</code> , меньше, чем значение 2^R , то контроллер не очистит разряды <code>chnl_useburst_set</code> и поэтому запросы по <code>dma_sreq[C]</code> будут маскированы. Если периферия не устанавливает <code>dma_req[C]</code> в состояние 1, то контроллер никогда не выполнит необходимых передач	

При отключении канала контроллер осуществляет DMA передачи, согласно правилам, представленным в таблице 360.

Таблица 360 – Правила осуществления DMA передач при «запрещенных» каналах

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

23.3.12 Диаграммы работы контроллера DMA

Данный раздел описывает следующие примеры функционирования контроллера с использованием правил обмена данными представленными в таблице 359:

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

Примечание – Все диаграммы, показанные на рисунках 93 - 97 подразумевают следующее:

- hready находится в состоянии 1;
- АНВ «ведомый» всегда дает ответ «ОКАУ».

23.3.13 Импульсный запрос на обработку

Рисунок 93 показывает временную диаграмму работы контроллера DMA при получении импульсного запроса от периферии.

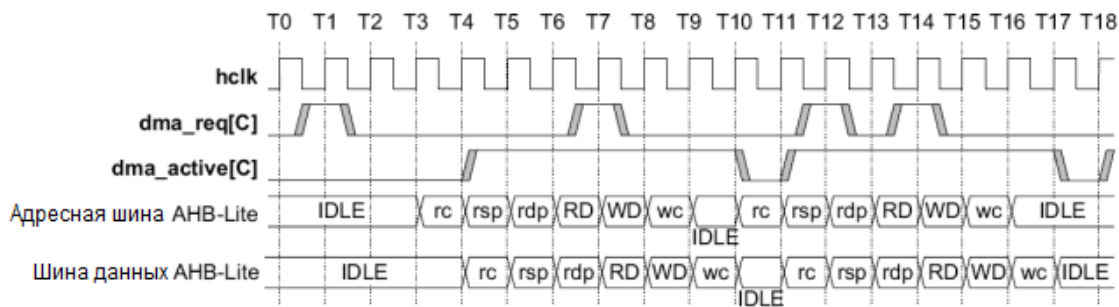


Рисунок 93 – Диаграмма работы при получении импульсного запроса от периферийного блока

23.3.14 Запрос на обработку по уровню

Рисунок 94 показывает временную диаграмму работы контроллера DMA при получении от периферии запроса на обработку по уровню.

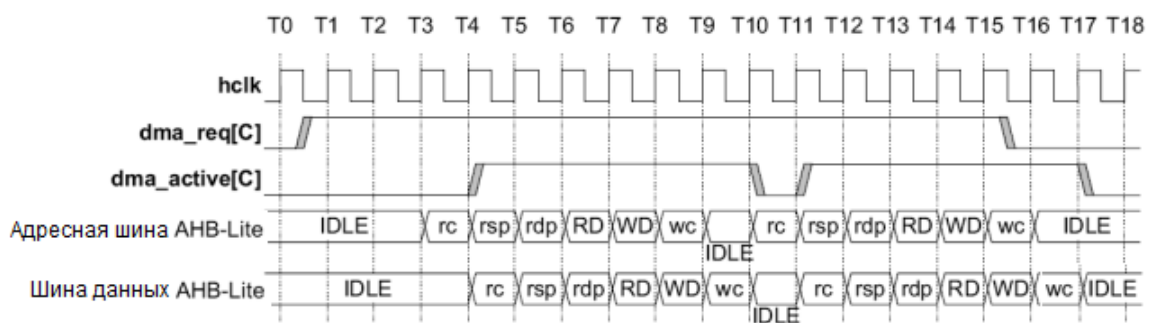


Рисунок 94 – Диаграмма работы при получении от периферийного блока запроса на обработку по уровню.

Пояснения к рисунку 94:

- T1 Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
- T4 Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C.
- T4-T7 Контроллер считывает управляющие данные канала, где:
 - rc – чтение настроек канала, channel_cfg;
 - rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr;
 - rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr.
- T7-T9 Контроллер выполняет передачу DMA по каналу C, где:
 - RD – чтение данных;
 - WD – запись данных.

- T9-T10 Контроллер осуществляет запись настроек канала, `channel_cfg`, где `ws` – запись настроек канала, `channel_cfg`.
- T10 Контроллер сбрасывает сигнал `dma_active[C]`, что указывает на окончание передачи DMA (см. правило 4).
Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что `chnl_req_mask_set[C]` находится в состоянии 0 (см. правило 18).
- T10-T11 Контроллер удерживает `dma_active[C]` на, как минимум, один такт `hclk` (см. правило 5).
- T11 Если канал C имеет более высокий приоритет, то контроллер устанавливает `dma_active[C]` и начинает вторую DMA передачу по каналу C.
- T11-T14 Контроллер считывает управляющие данные канала.
- T14-T16 Контроллер выполняет передачу DMA по каналу C.
- T15-T16 Периферийный блок обнаруживает, что передача DMA началась и сбрасывает `dma_req[C]`.
- T16-T17 Контроллер осуществляет запись настроек канала, `channel_cfg`.
- T17 Контроллер сбрасывает сигнал `dma_active[C]`, что указывает на окончание передачи DMA (см. правило 4).

При использовании запроса на обработку по уровню, периферийный блок может не обладать достаточным быстродействием, чтобы во время снять сигнал запроса, в этом случае он должен установить сигнал `dma_stall`. Установка сигнала `dma_stall` предотвращает повторение выполненной передачи.

23.3.15 Флаги завершения

Рисунок 95 демонстрирует функционирование сигнала (флага) `dma_done[]` при следующих условиях:

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

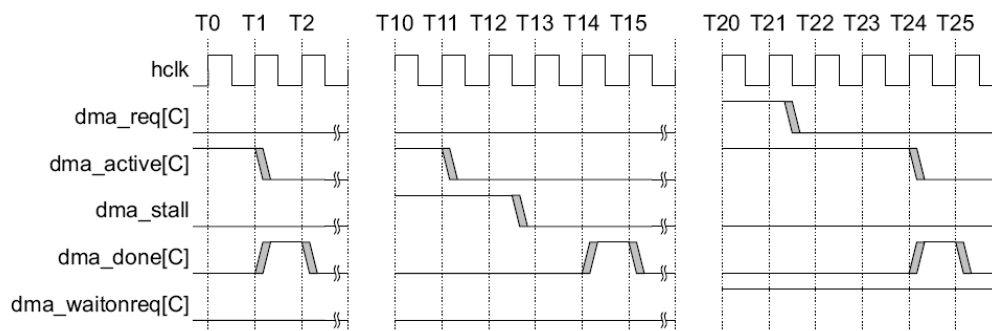


Рисунок 95 – Диаграммы функционирования `dma_done`

Пояснения к рисунку 95 от T0 до T2:

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

Пояснения к рисунку 95 от T10 до T15:

- T11 Контроллер сбрасывает сигнал `dma_active[C]`, что указывает на окончание передачи DMA (см. правило 4).

Примечание – Контроллер не устанавливает сигнал `dma_done[C]`, так как сигнал `dma_stall` установлен в 1 в предшествующем такте `hclk` (см. правила 9 и 12).

- T12-T13 Периферийный блок сбрасывает сигнал `dma_stall`.
- T14-T15 Контроллер завершает цикл DMA и если `cycle_ctrl[2]` установлен в 0, то устанавливает в 1 `dma_done[C]` на один такт `hclk` (см. правила 8 и 9). Для других разрешенных каналов сигнал `dma_done[C]` останется в состоянии 0 (см. правило 6).

Пояснения к рисунку 95 от T20 до T25:

- T20 Контроллер выполнил передачу DMA, но из-за установленного в 1 `dma_waitonreq[C]` он должен ожидать сброса в 0 сигнала `dma_req[C]`, перед тем как сбросить `dma_active[C]` (см. правило 11) и установить `dma_done[C]` (см. правило 9).
- T21-T25 Периферийный блок сбрасывает `dma_req[C]`.
- T24 Контроллер сбрасывает сигнал `dma_active[C]`, что указывает на окончание передачи DMA (см. правило 4).
- T24-T25 Контроллер завершает цикл DMA и если `cycle_ctrl[2]` установлен в 0, то устанавливает в 1 `dma_done[C]` на один такт `hclk` (см. правила 8 и 9). Для других разрешенных каналов сигнал `dma_done[C]` останется в состоянии 0 (см. правило 6).

23.3.16 Флаги ожидания запроса на обработку

Рисунки 96, 97 демонстрируют примеры использования флагов ожидания запроса на обработку при выполнении 2^R передач и одиночных передач:

- диаграмма работы контроллера DMA при использовании периферией `dma_waitonreq`;
- диаграмма работы контроллера DMA при использовании периферией `dma_waitonreq` совместно с `dma_sreq`.

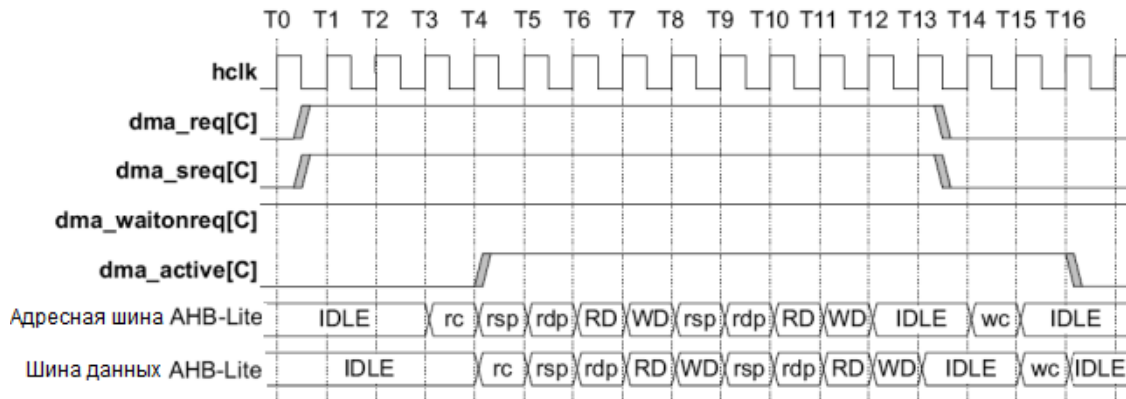


Рисунок 96 – Диаграмма работы контроллера DMA при использовании периферийной dma_waitonreq

Пояснения к рисунку 96:

- T0-T16 Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10).
- T0-T1 Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
- T3-T4 Периферийный блок удерживает dma_req[C] и dma_sreq[C] в 1. Контроллер игнорирует dma_sreq[C] запрос и отвечает на dma_req[C] запрос (см. правила 16 и 17).
- T4 Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C.
- T4-T7 Контроллер считывает управляющие данные канала, где:
rc – чтение настроек канала, channel_cfg;
rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr;
rdp - чтение указателя адреса окончания данных приемника, dst_data_end_ptr.
- T7-T9 Контроллер выполняет передачу DMA по каналу C, где:
RD – чтение данных;
WD – запись данных.
- T9-T11 Контроллер считывает 2 указателя адреса окончания данных rsp и rdp.
- T11-T13 Периферийный блок сбрасывает сигналы dma_req[C] и dma_sreq[C].
- T15-T16 Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg.
- T16 Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11).
Контроллер устанавливает значение по чтению регистра chnl_useburst_set[C] в 0, если количество оставшихся передач менее 2^R (см. правило 15).

Рисунок 97 показывает работу контроллера DMA при установке `dma_waitonreq` в 1 и выполнении одиночной DMA передачи.

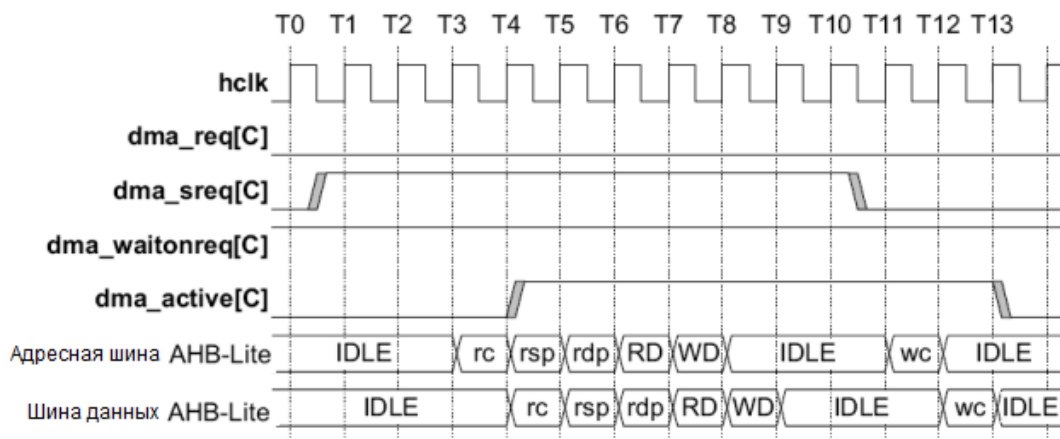


Рисунок 97 – Диаграмма работы контроллера DMA при использовании периферией `dma_waitonreq` совместно с `dma_sreq`

Пояснения к рисунку 97:

- T0-T13 Периферийный блок должен оставлять состояние `dma_waitonreq[C]` постоянно (см. правило 10).
- T0-T1 Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что `chnl_useburst_set[C]` находится в состоянии 0 (см. правила 13 и 14).
- T3-T4 Контроллер отвечает на `dma_sreq[C]` запрос (см. правила 16).
- T4 Контроллер устанавливает `dma_active[C]` (см. правила 2 и 3) и начинает DMA передачи по каналу C.
- T4-T7 Контроллер считывает управляющие данные канала, где:
`rc` – чтение настроек канала, `channel_cfg`;
`rsp` – чтение указателя адреса окончания данных источника, `src_data_end_ptr`;
`rdp` – чтение указателя адреса окончания данных приемника, `dst_data_end_ptr`.
- T7-T9 Контроллер выполняет передачу DMA по каналу C, где:
`RD` – чтение данных;
`WD` – запись данных.
 Это запрос в ответ на `dma_sreq[]`, таким образом `R=0` и следовательно контроллер исполнит 1 DMA передачу.
- T10-T11 Периферийный блок сбрасывает сигнал `dma_sreq[C]`.
- T12-T13 Контроллер осуществляет запись настроек канала, `channel_cfg`, где
`wc` – запись настроек канала, `channel_cfg`.
- T13 Контроллер сбрасывает сигнал `dma_active[C]`, что указывает на окончание передачи DMA (см. правило 11).

23.3.17 Правила арбитража DMA

Контроллер имеет возможность настройки момента арбитража при передачах DMA. Эта возможность позволяет уменьшить время отклика при обслуживании каналов с высоким приоритетом.

Контроллер имеет четыре разряда, которые определяют количество транзакций по шине АНВ до повторения арбитража. Эти разряды, так называемая степень R числа 2, изменение R напрямую устанавливает периодичность арбитража. Для примера, если R равно четырем, то арбитраж будет проводиться через каждые 16 передач DMA.

Таблица 361 показывает возможную периодичность арбитража.

Таблица 361 – Периодичность арбитража в единицах передач по шине АНВ

Значение R	Периодичность арбитража каждые x передач DMA
b0000	1
b0001	2
b0010	4
b0011	8
b0100	16
b0101	32
b0110	64
b0111	128
b1000	256
b1001	512
b1010-b1111	1024

Примечание – Необходимо с осторожностью устанавливать большие значения R для низкоприоритетных каналов, так как это может привести к невозможности обслуживать запросы по высокоприоритетным каналам.

При $N > 2^R$ (N – номер передачи) и если результат деления 2^R на N не целое число, то контроллер всегда выполняет последовательность из 2^R передач до тех пор, пока не станет верным $N < 2^R$. Контроллер выполняет оставшиеся N передач в конце цикла DMA.

Разряды степени R числа 2 находятся в структуре управляющих данных канала. Местонахождение этих разрядов описано в разделе «Управляющие данные канала».

Приоритет

При проведении арбитража, определяется канал для обслуживания в следующем цикле DMA. Определение следующего канала происходит по следующим признакам:

- номер канала;
- уровень приоритета, присвоенного каналу.

Каждому каналу может быть присвоен уровень приоритета по умолчанию (низкий) или высокий уровень приоритета. Присвоение уровня приоритета осуществляется установкой или сбросом разряда `chnl_priority_set`.

Канал номер 0 имеет высший уровень приоритета и уровень приоритета снижается с увеличением номера канала. Таблица 362 показывает уровень приоритета каналов DMA в порядке его уменьшения.

Таблица 362 – Уровень приоритета каналов DMA

Номер канала	Установка уровня приоритета	Уровень приоритета в порядке его уменьшения
0	Высокий	Наивысший уровень приоритета
1	Высокий	-
2	Высокий	-
-	Высокий	-
-	Высокий	-
-	Высокий	-
30	Высокий	-
31	Высокий	-
0	По умолчанию (низкий)	-
1	По умолчанию (низкий)	-
2	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
30	По умолчанию (низкий)	-
31	По умолчанию (низкий)	Низший уровень приоритета

После окончания цикла DMA, контроллер выбирает следующий для обслуживания канал из всех включенных каналов DMA. Рисунок 98 показывает процесс выбора следующего канала для обслуживания.

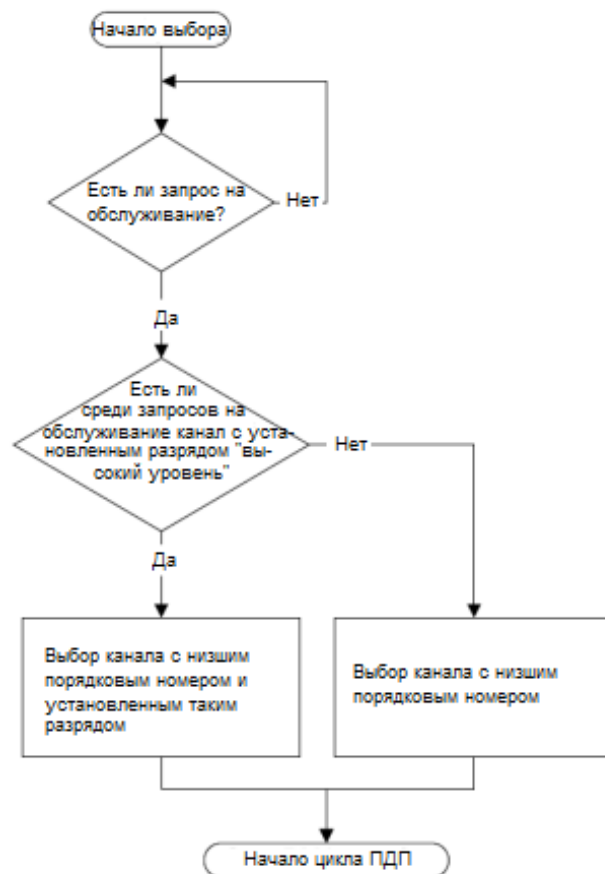


Рисунок 98 – Алгоритм выбора следующего канала для обслуживания.

Начало выбора

Есть ли запрос на обслуживание.

Есть ли среди запросов на обслуживание канал с установленным разрядом «высокий уровень».

Выбор канала с низшим порядковым номером и установленным таким разрядом.

Выбор канала с низшим порядковым номером.

Начало цикла DMA.

Типы циклов DMA.

Разряды `cycle_ctrl` определяют, как контроллер будет выполнять циклы DMA.

Описание значений этих разрядов приведено в таблице 363.

Таблица 363 – Типы циклов DMA

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

Примечание – Разряды `cycle_ctrl` находятся в области памяти отведенной под `channel_cfg`, описанной в пункте 23.3.18 «Структура управляющих данных канала»..

Для всех типов циклов DMA повторный арбитраж происходит после 2^R передач DMA. Если установить длинный период арбитража на низко приоритетном канале, то это заблокирует все запросы на обработку от других каналов до тех пор, пока не будут выполнены 2^R передач DMA по данному каналу. Поэтому устанавливая значение R необходимо учитывать, что это может привести к повышенному времени отклика на запрос на обработку от высокоприоритетных каналов.

Данный раздел описывает следующие типы циклов DMA:

- недействительный;
- основной;
- авто-запрос;
- пинг-понг;
- работа с памятью «исполнение с изменением конфигурации»;
- работа с периферией «исполнение с изменением конфигурации».

Недействительный

После окончания цикла DMA контроллер устанавливает тип цикла в значение «недействительный», для предотвращения повтора выполненного цикла DMA.

Основной

В этом режиме контроллер работает только с основными или альтернативными управляющими данными канала. После того, как разрешена работа канала и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

1 Контроллер выполняет 2^R передач. Если число оставшихся передач 0, контроллер переходит к шагу 3;

2 Осуществление арбитража:

– если высокоприоритетный канал выдает заброс на обработку, то контроллер начинает обслуживание этого канала;

– если периферийный блок или программное обеспечение выдает заброс на обработку (повторный запрос на обработку по каналу), то контроллер переходит к выполнению шагу 1.

3 Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала `hclk`. Это указывает центральному процессору о завершении цикла DMA.

Авто-запрос

Функционируя в данном режиме, контроллер ожидает получения одиночного запроса на обработку для разрешения работы и выполнения цикла DMA. Такая работа позволяет выполнять передачу больших пакетов данных без существенного увеличения времени отклика на обслуживание высокоприоритетных запросов и не требует множественных запросов на обработку от процессора или периферийных блоков.

Контроллер позволяет настроить для использования первичную или альтернативную структуру управляющих данных канала. После того, как разрешена работа канала и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

1 Контроллер выполняет 2^R передач для канала C. Если число оставшихся передач 0, контроллер переходит к шагу 3.

2 Контроллер осуществляет арбитраж. Когда канал C становится каналом с самым высоким приоритетом, контроллер переходит к шагу 1.

3 Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала `hclk`. Это указывает центральному процессору о завершении цикла DMA.

Пинг-понг

В данном режиме работы контроллер выполняет цикл DMA, используя одну из структур управляющих данных, а затем выполняет еще один цикл DMA, используя другую структуру управляющих данных. Контроллер выполняет циклы DMA с переключением структур до тех пор, пока не считает «неправильную» структуру данных и ли пока процессор не запретит работу канала.

Рисунок 99 демонстрирует пример функционирования контроллера в режиме Пинг-понг.

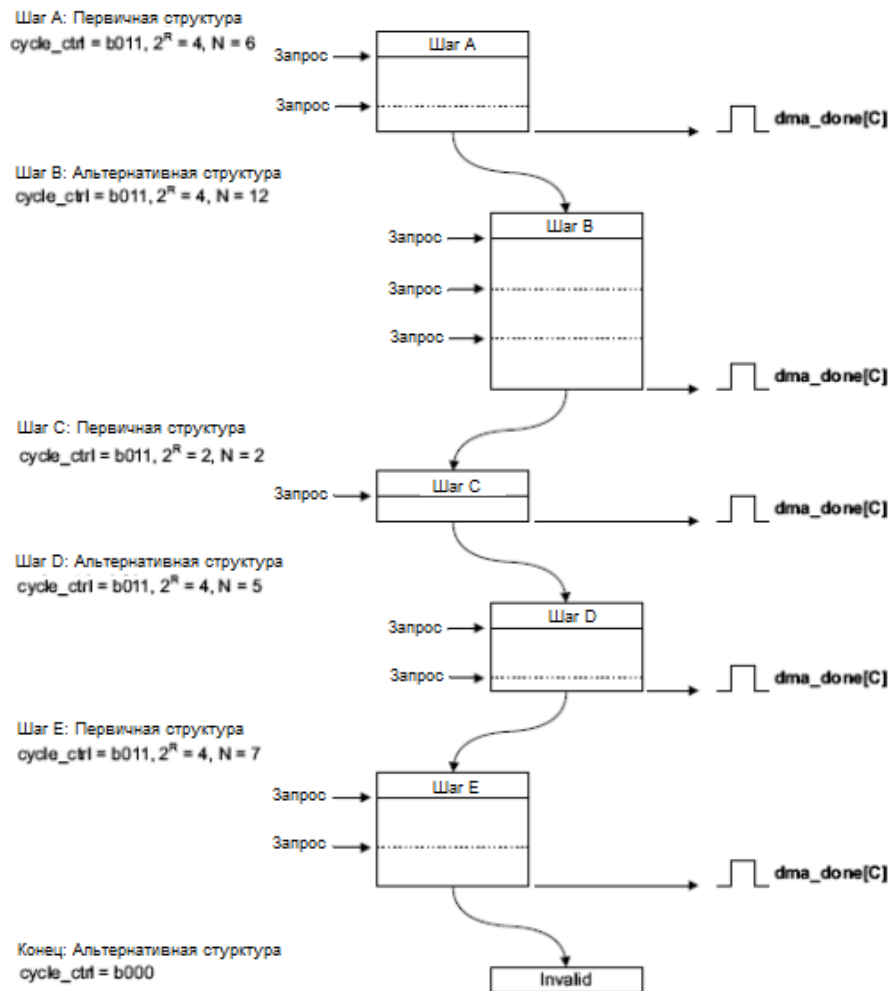


Рисунок 99 – Пример работы в режиме Пинг-понг

Пояснения к рисунку

Шаг А. Первичная структура, $cycle_ctrl=b011$, $2^R = 4$, $N=6$

Шаг В. Альтернативная структура, $cycle_ctrl=b011$, $2^R = 4$, $N=12$

Шаг С. Первичная структура, $cycle_ctrl=b011$, $2^R = 2$, $N=2$

Шаг D. Альтернативная структура, $cycle_ctrl=b011$, $2^R = 4$, $N=5$

Шаг Е. Первичная структура, $cycle_ctrl=b011$, $2^R = 4$, $N=7$

Конец. Альтернативная структура, $cycle_ctrl=b000$

Пояснения к рисунку 99:

- Шаг А Процессор устанавливает первичную структуру управляющих данных для шага А. Процессор устанавливает альтернативную структуру управляющих данных для шага В. Это позволит контроллеру переключиться к шагу В незамедлительно после выполнения шага А, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала. Контроллер получает запрос и выполняет четыре передачи DMA. Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл, в ситуации отсутствия высокоприоритетных запросов. Контроллер выполняет оставшиеся две передачи DMA. Контроллер устанавливает $dma_done[C]$ в состояние 1 на один такт сигнала синхронизации $hclk$ и входит в процедуру арбитража.

После выполнения шага А процессор может установить первичные управляющие данные канала для шага С. Это позволит контроллеру переключиться к шагу С незамедлительно после выполнения шага В, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.

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

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

После выполнения шага В процессор может установить альтернативные управляющие данные канала для шага D.

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

Шаг С Контроллер выполняет две передачи DMA.
 Контроллер устанавливает `dma_done[C]` в состояние 1 на один такт сигнала синхронизации `hclk` и входит в процедуру арбитража.

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

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

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

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

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

Если контроллер получит новый запрос на обработку от данного канала, и он будет самым приоритетным, он предпримет попытку выполнения следующего шага. Однако, из-за того, что процессор не установил альтернативные управляющие данные и по окончании шага D контроллер установил `cycle_ctrl` в состояние `b000`, передачи DMA прекращаются.

Примечание – Для прерывания цикла DMA, исполняемого в режиме Пинг-понг, также возможен перевод режима работы контроллера на шаге E в Основной цикл DMA, путем установки `cycle_ctrl` в `3'b001`.

Режим работы с памятью «исполнение с изменением конфигурации».

В данном режиме контроллер, получая начальный запрос на обработку, выполняет четыре передачи DMA, используя первичные управляющие данные. По окончании этих передач, контроллер начинает цикл DMA, используя альтернативные управляющие данные. После чего контроллер выполняет еще четыре передачи DMA, используя первичные управляющие данные. Контроллер продолжает выполнять циклы ПДА, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

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

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

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

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 364 перечисляет области памяти `channel_cfg`, те которые должны быть определены константами, и те, значения которых определяются пользователем.

Таблица 364 – `Channel_cfg` для первичной структуры управляющих данных в режиме работы с памятью «исполнение с изменением конфигурации»

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
31...30	<code>dst_inc</code>	<code>b'10</code>	Контроллер производит инкремент адреса пословно
29...:28	<code>dst_size</code>	<code>b'10</code>	Контроллер осуществляет передачу пословно
27...26	<code>src_inc</code>	<code>b'10</code>	Контроллер производит инкремент адреса пословно
25...24	<code>src_size</code>	<code>b'10</code>	Контроллер осуществляет передачу пословно
17...14	<code>R_power</code>	<code>b'0010</code>	Контроллер выполняет 4 передачи DMA
3	<code>next_useburst</code>	<code>b'0</code>	Для данного режима этот разряд должен быть равен 0
2...0	<code>cycle_ctrl</code>	<code>b'100</code>	Контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями определяемыми пользователем			
23...21	<code>dst_prot_ctrl</code>	-	Определяет состояние HPROT при записи данных в приемник

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

* Так как R_power установлены в состояние 4, необходимо задавать значение N кратное четырем. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные.

Рисунок 100 демонстрирует пример функционирования в режиме работы с памятью «исполнение с изменением конфигурации».

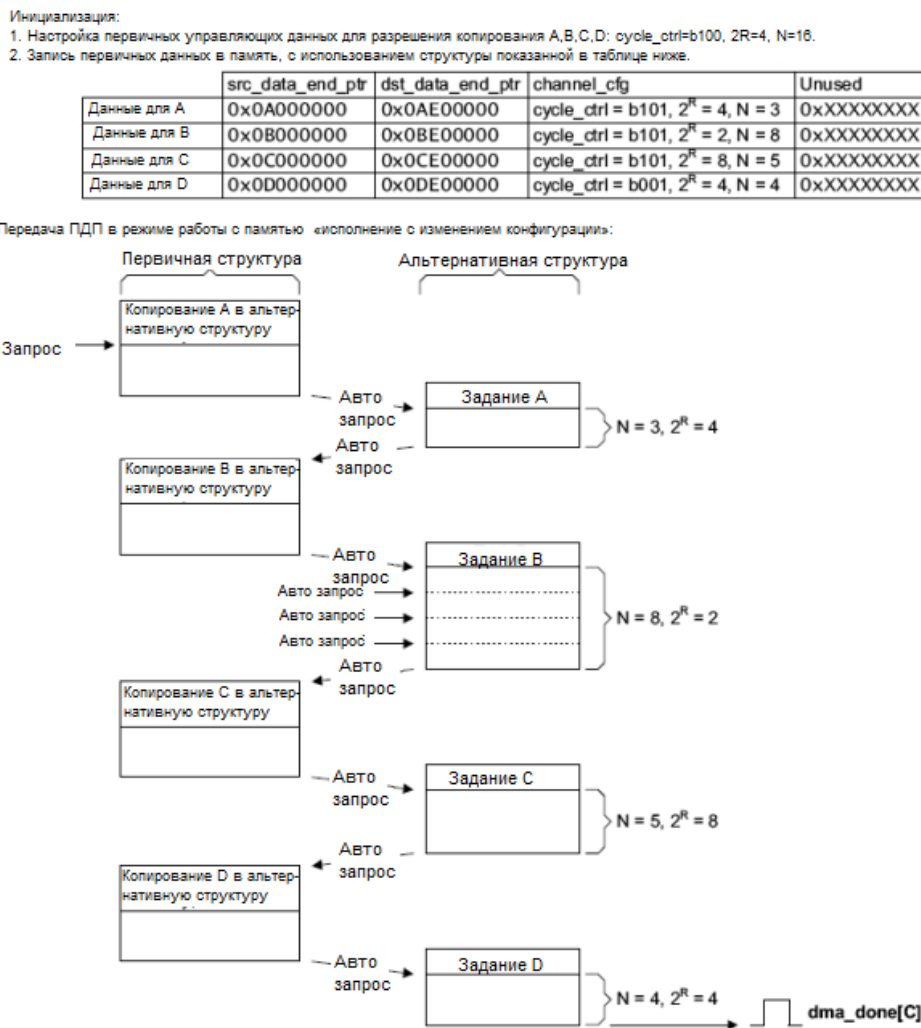


Рисунок 100 – Пример функционирования контроллера в режиме работы с памятью «исполнение с изменением конфигурации»

Пояснения к рисунку:

Инициализация

1 Настройка первичных управляющих данных для разрешения копирования A, B, C, и D: cycle_ctrl=b100, 2^R=4, N=16.

2 Запись первичных данных в память, с использованием структуры показанной в таблице 365.

Передача DMA в режиме работы с памятью «исполнение с изменением конфигурации».

Пояснения к рисунку 100:

Инициализация:

1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с памятью «исполнение с изменением конфигурации», путем установки `cycle_ctrl` в `b100`. Так как управляющие данные канала состоят из четырех слов необходимо установить 2^R в 4. В этом примере количество задач равно четырем и, поэтому `N` установлен в 16.

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

3 Процессор разрешает работу канала DMA.

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

Первичная, копирование A.

По получению запроса на обслуживание, контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи A.

Контроллер генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Задача A.

Контроллер выполняет задачу A. По окончании, контроллер генерирует авто-запрос для канала и проводит процедуру арбитража.

Первичная, копирование B.

Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи B.

Контроллер генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Задача B.

Контроллер выполняет задачу B. По окончании, контроллер генерирует авто-запрос для канала и проводит процедуру арбитража.

Первичная, копирование C.

Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи C.

Контроллер генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Задача C.

Контроллер выполняет задачу C. По окончании, контроллер генерирует авто-запрос для канала и проводит процедуру арбитража.

Первичная, копирование D.

Контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи D.

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

Контроллер генерирует авто-запрос для канала, после чего проводит процедуру арбитража.

Задача D.

Контроллер выполняет задачу D, используя основной цикл DMA.

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

Режим работы с периферией «исполнение с изменением конфигурации».

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

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

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

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

процессор переведет контроллер в режим Основной во время цикла с альтернативной структурой;

контроллер считает «неправильную» структуру управляющих данных.

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

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

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 365 перечисляет области памяти `channel_cfg`, те которые должны быть определены константами, и те, значения которых определяются пользователем.

Таблица 365 – `Channel_cfg` для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации»

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
31...30	<code>dst_inc</code>	b'10	Контроллер производит инкремент адреса пословно
29...28	<code>dst_size</code>	b'10	Контроллер осуществляет передачу пословно

Разряды	Обозначение	Значение	Описание
27...26	src_inc	b'10	Контроллер производит инкремент адреса пословно
25...24	src_size	b'10	Контроллер осуществляет передачу пословно
17...14	R_power	b'0010	Контроллер выполняет 4 передачи DMA
2...0	cycle_ctrl	b'110	Контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями определяемыми пользователем			
23...21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20...18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13...4	n_minus_1	N*	Настраивает контроллер на выполнение N передач DMA, где N кратно 4.
3	next_useburst	-	При установке в 1, контроллер установит chnl_useburst_set[C] в 1 после выполнения передачи с альтернативной структурой.

* Так как R_power установлены в состояние 4, необходимо задавать значение N, кратное четырем. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные.

Рисунок 101 демонстрирует пример функционирования в режиме работы с периферией «исполнение с изменением конфигурации».

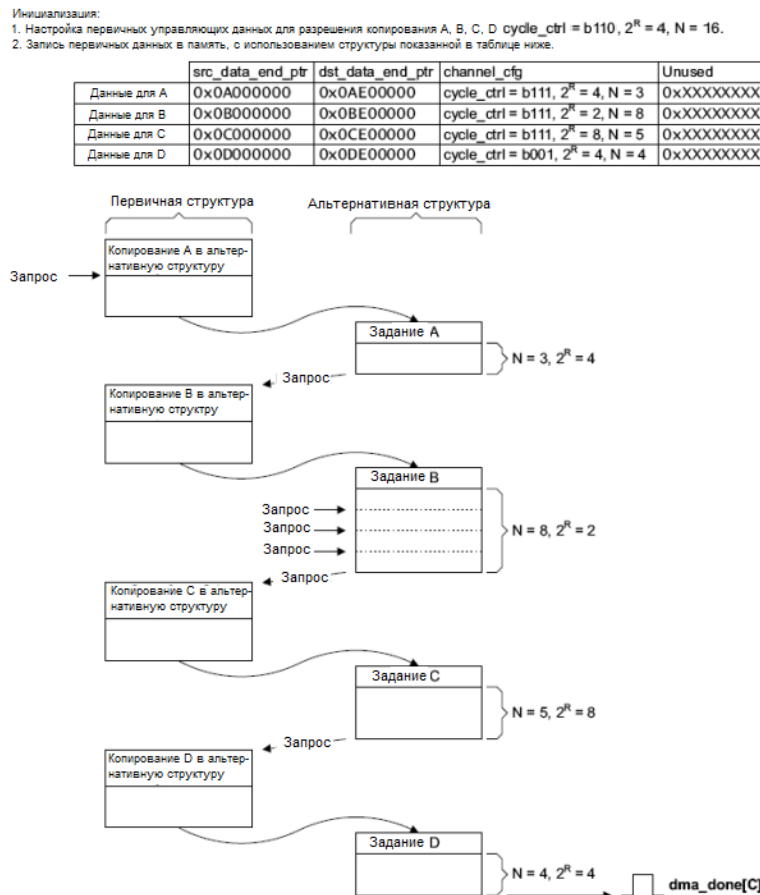


Рисунок 101 – Пример функционирования контроллера в режиме работы с периферией «исполнение с изменением конфигурации»

Пояснения к рисунку:

Инициализация

1 Настройка первичных управляющих данных для разрешения копирования А, В, С, и D: $cycle_ctrl=b110$, $2^R=4$, $N=16$.

2 Запись первичных данных в память, с использованием структуры показанной в таблице 366.

Передача DMA в режиме работы с периферией «исполнение с изменением конфигурации».

Пояснения к рисунку 101.

Инициализация

1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации», путем установки $cycle_ctrl$ в $b110$. Так как управляющие данные канала состоят из 4 слов необходимо установить 2^R в 4. В этом примере количество задач равно 4 и, поэтому N установлен в 16.

2 Процессор записывает управляющие данные для задач А, В, С, D в область памяти с адресом, указанным в $src_data_end_ptr$.

3 Процессор разрешает работу канала DMA.

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

Первичная, копирование А.

По получению запроса на обслуживание, контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи А.

Задача А.

Контроллер выполняет задачу А.

По окончании, контроллер проводит процедуру арбитража.

Первичная, копирование В.

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи В.

Задача В.

Контроллер выполняет задачу В. Для завершения задачи периферия должна установить последовательно 3 запроса.

По окончании, контроллер проводит процедуру арбитража.

Первичная, копирование С.

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи С.

Задача С.

Контроллер выполняет задачу С.

По окончании, контроллер проводит процедуру арбитража.

После выставления периферией нового запроса на обслуживание при условии, что этот запрос является наиболее приоритетным, процесс продолжается следующим образом:

Первичная, копирование D.

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

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

Задача D.

Контроллер выполняет задачу D, используя основной цикл DMA.

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

Индикация ошибок

При получении контроллером по шине АНВ-Lite ответа об ошибке, он выполняет следующие действия:

- отключает канал связанный с ошибкой;
- устанавливает флаг `dma_err` в состояние 1.

После обнаружения процессором флага `dma_err`, процессор определяет номер канала, который был активен в момент появления ошибки. Для этого он осуществляет следующее:

Чтение регистра `chnl_enable_set` с целью создания списка отключенных каналов.

Если канал установил флаг `dma_done[]`, то контроллер отключает канал. Программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно установили флаги `dma_done[]`.

Процессор должен сравнить список выключенных каналов, полученный в шаге 1, с данными о каналах, которые недавно устанавливали флаги `dma_done[]`. Канал, по которому отсутствуют данные об установке флага `dma_done[]`, это и есть канал, с которым связана ошибка.

23.3.18 Структура управляющих данных канала

В системной памяти должна быть отведена область для хранения управляющих данных каналов. Системная память должна:

- предоставлять смежную область системной памяти, к которой контроллер и процессор имеют доступ;
- иметь базовый адрес, который целочисленно кратен общему размеру структуры управляющих данных канала.

Рисунок 102 показывает область памяти необходимую контроллеру для структур управляющих данных канала, при использовании всех 32 каналов и опциональной альтернативной структуры управляющих данных.

Данные альтернативной структуры

Alternate_Ch_31	0x3F0
Alternate_Ch_30	0x3E0
Alternate_Ch_29	0x3D0
Alternate_Ch_28	0x3C0
Alternate_Ch_27	0x3B0
Alternate_Ch_26	0x3A0
Alternate_Ch_25	0x390
Alternate_Ch_24	0x380
Alternate_Ch_23	0x370
Alternate_Ch_22	0x360
Alternate_Ch_21	0x350
Alternate_Ch_20	0x340
Alternate_Ch_19	0x330
Alternate_Ch_18	0x320
Alternate_Ch_17	0x310
Alternate_Ch_16	0x300
Alternate_Ch_15	0x2F0
Alternate_Ch_14	0x2E0
Alternate_Ch_13	0x2D0
Alternate_Ch_12	0x2C0
Alternate_Ch_11	0x2B0
Alternate_Ch_10	0x2A0
Alternate_Ch_9	0x290
Alternate_Ch_8	0x280
Alternate_Ch_7	0x270
Alternate_Ch_6	0x260
Alternate_Ch_5	0x250
Alternate_Ch_4	0x240
Alternate_Ch_3	0x230
Alternate_Ch_2	0x220
Alternate_Ch_1	0x210
Alternate_Ch_0	0x200

Данные первичной структуры

Primary_Ch_31	0x1F0
Primary_Ch_30	0x1E0
Primary_Ch_29	0x1D0
Primary_Ch_28	0x1C0
Primary_Ch_27	0x1B0
Primary_Ch_26	0x1A0
Primary_Ch_25	0x190
Primary_Ch_24	0x180
Primary_Ch_23	0x170
Primary_Ch_22	0x160
Primary_Ch_21	0x150
Primary_Ch_20	0x140
Primary_Ch_19	0x130
Primary_Ch_18	0x120
Primary_Ch_17	0x110
Primary_Ch_16	0x100
Primary_Ch_15	0x0F0
Primary_Ch_14	0x0E0
Primary_Ch_13	0x0D0
Primary_Ch_12	0x0C0
Primary_Ch_11	0x0B0
Primary_Ch_10	0x0A0
Primary_Ch_9	0x090
Primary_Ch_8	0x080
Primary_Ch_7	0x070
Primary_Ch_6	0x060
Primary_Ch_5	0x050
Primary_Ch_4	0x040
Primary_Ch_3	0x030
Primary_Ch_2	0x020
Primary_Ch_1	0x010
Primary_Ch_0	0x000

Не используются	0x00C
Управление	0x008
Указатель конца данных прием.	0x004
Указатель конца данных ист.	0x000

Рисунок 102 – Карта памяти для 32 каналов, включая альтернативную структуру управляющих данных

Пример, показанный на рисунке 102, использует 1 Кбайт системной памяти. В этом примере контроллер использует младшие 10 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXX000, далее 0xXXXXX400, далее 0xXXXXX800, далее 0xXXXXXC00.

Возможно установить базовый адрес для первичной структуры управляющих данных путем записи соответствующего значения в регистр ctrl_base_ptr.

Необходимый размер области системной памяти зависит от:

- количества каналов, используемых в контроллере;
- от того, используется или нет альтернативная структура управляющих данных.

Таблица 366 перечисляет разряды адреса, которые используются контроллером при доступе к различным элементам структуры управляющих данных, в зависимости от количества каналов, используемых в контроллере.

Таблица 366 – Разряды адреса соответствующие элементам структуры управляющих данных

Количество каналов, используемых в контроллере	[9]	[8]	[7]	[6]	[5]	[4]	[3...0]
1						A	0x0 0x4 0x8
2					A	C[0]	
3-4				A	C[1]	C[0]	
5-8			A	C[2]	C[1]	C[0]	
9-16		A	C[3]	C[2]	C[1]	C[0]	
17-32	A	C[4]	C[3]	C[2]	C[1]	C[0]	

где A Выбирает одну из структур управляющих данных канал;
A = 0 Выбирает первичную структуру управляющих данные;
A = 1 Выбирает альтернативную структуру управляющих данные;
C[x:0] Выбирает канал DMA;
Address[3:0] Выбирает один из управляющих элементов:
– 0x0 Выбирает указатель конца данных источник;
– 0x4 Выбирает указатель конца данных приемник;
– 0x8 Выбирает конфигурацию управляющих данные;
– 0xC Контроллер не имеет доступа к этому адресу. Если это необходимо, то возможно разрешить процессору использовать эти адреса в качестве системной памяти.

Примечание – Совсем не обязательно вычислять базовый адрес альтернативной структуры управляющих данных, так как регистр alt_ctrl_base_ptr содержит эту информацию.

Рисунок 103 демонстрирует пример реализации контроллера с использованием 3 каналов DMA и альтернативной структурой управляющих данных.



Рисунок 103 – Карта памяти для трех каналов DMA, включая альтернативную структуру управляющих данных

Пример структуры управляющих данных на рисунке 103 использует 128 байт системной памяти. В этом примере контроллер использует младшие 6 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXXX00, далее 0xXXXXXX80.

Таблица 367 перечисляет все разрешенные значения базового адреса для первичной структуры управляющих данных, в зависимости от количества каналов DMA, использованных в контроллере.

Таблица 367 – Разрешенные базовые адреса

Кол-во каналов DMA	Разрешенные значения базового адреса для первичной структуры управляющих данных
1	0хXXXXXXXX00, 0хXXXXXXXX20, 0хXXXXXXXX40, 0хXXXXXXXX60, 0хXXXXXXXX80, 0хXXXXXXXXA0, 0хXXXXXXXXC0, 0хXXXXXXXXE0
2	0хXXXXXXXX00, 0хXXXXXXXX40, 0хXXXXXXXX80, 0хXXXXXXXXC0
3-4	0хXXXXXXXX00, 0хXXXXXXXX80
5-8	0хXXXXXX000, 0хXXXXXX100, 0хXXXXXX200, 0хXXXXXX300, 0хXXXXXX400, 0хXXXXXX500, 0хXXXXXX600, 0хXXXXXX700, 0хXXXXXX800, 0хXXXXXX900, 0хXXXXXXA00, 0хXXXXXXB00, 0хXXXXXXC00, 0хXXXXXXD00, 0хXXXXXXE00, 0хXXXXXXF00,
9-16	0хXXXXXX000, 0хXXXXXX200, 0хXXXXXX400, 0хXXXXXX600, 0хXXXXXX800, 0хXXXXXXA00, 0хXXXXXXC00, 0хXXXXXXE00
17-32	0хXXXXXX000, 0хXXXXXX400, 0хXXXXXX800, 0хXXXXXXC00

Контроллер использует системную память для доступа к двум указателям адреса конца данных и разрядам управления каждого канала. Следующие подразделы описывают эти 32-разрядные области памяти и процедуру вычисления контроллером адреса передачи DMA:

- указатель конца данных источника;
- указатель конца данных приемника;
- разряды управления;
- вычисление адреса.

Указатель конца данных источника

Область памяти под названием `src_data_end_ptr` содержит указатель на последний адрес месторасположения данных источника. таблица 368 перечисляет значения разрядов этой области.

Таблица 368 – Значения разрядов `src_data_end_ptr`

Разряд	Имя	Описание
31...0	<code>src_data_end_ptr</code>	Указатель на последний адрес данных источника

Перед тем как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.

Примечание – Контроллер не имеет доступа по записи в эту область памяти.

Указатель конца данных приемника

Область памяти под названием `dst_data_end_ptr` содержит указатель на последний адрес месторасположения данных приемника. Таблица 369 перечисляет значения разрядов этой области.

Таблица 369 – Значения разрядов dst_data_end_ptr

Разряд	Имя	Описание
31...0]	dst_data_end_ptr	Указатель на последний адрес данных приемника

Перед тем как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.

Примечание – Контроллер не имеет доступа по записи в эту область памяти.

Разряды управления

Область памяти под названием channel_cfg обеспечивает управление каждой передачей DMA. Рисунок 104 показывает название разрядов этой области.

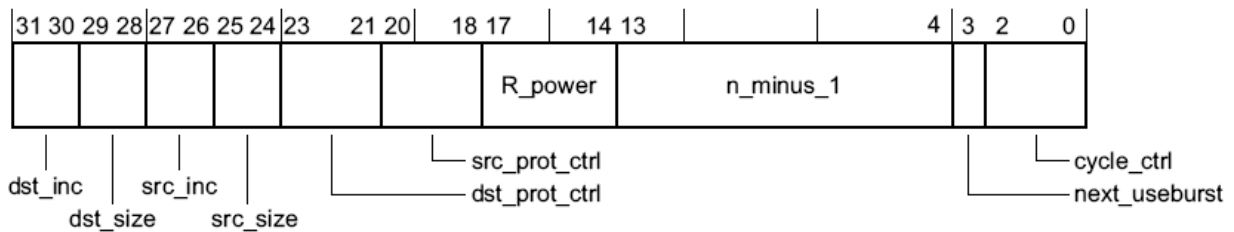


Рисунок 104 – Название разрядов channel_cfg

Таблица 370 перечисляет назначение разрядов этой области памяти.

Таблица 370 – Назначение разрядов channel_cfg

Разряд	Имя	Описание
31, 30	dst_inc	Шаг инкремента адреса приемника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: b00 = байт; b01 = полуслово (в русском обычно слово); b10 = слово (в русском обычно двойное слово); b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr; Разрядность данных источника = полуслово; b00 = зарезервировано; b01 = полуслово; b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr; Разрядность данных источника = слово: b00 = зарезервировано; b01 = зарезервировано; b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr
29, 28	dst_size	Размерность данных приемника. Примечание – Значение этого поля должно быть равно значению поля src_size

Разряд	Имя	Описание
27, 26	src_inc	Шаг инкремента адреса источника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: b00 = байт; b01 = полуслово (в русском обычно слово); b10 = слово (в русском обычно двойное слово); b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr; Разрядность данных источника = полуслово; b00 = зарезервировано; b01 = полуслово; b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr; Разрядность данных источника = слово: b00 = зарезервировано; b01 = зарезервировано; b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr
25, 24	src_size	Задаёт размерность данных источника: b00 = байт; b01 = полуслово (в русском обычно слово); b10 = слово (в русском обычно двойное слово); b11 = зарезервировано
23...21	dst_prot_ctrl	Задаёт состояние HPROT[3:1], когда контроллер записывает данные в приемник. Разряд [23] – управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется; Разряд [22] – управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется; Разряд [21] – управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ не привилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный
20...18	src_prot_ctrl	Задаёт состояние HPROT[3:1], когда контроллер считывает данные из источника. Разряд [20] – управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется; Разряд [19] – управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется; Разряд [18] – управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ не привилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный

Разряд	Имя	Описание
17...14	R_power	<p>Задаёт количество передач DMA до выполнения контроллером процедуры арбитража.</p> <p>Возможные значения:</p> <p>b0000 -- арбитраж производится после каждой передачи DMA; b0001 -- арбитраж производится после 2 передач DMA; b0010 -- арбитраж производится после 4 передач DMA; b0011 -- арбитраж производится после 8 передач DMA; b0100 -- арбитраж производится после 16 передач DMA; b0101 -- арбитраж производится после 32 передач DMA; b0110 -- арбитраж производится после 64 передач DMA; b0111 -- арбитраж производится после 128 передач DMA; b1000 -- арбитраж производится после 256 передач DMA; b1001 -- арбитраж производится после 512 передач DMA; b1010- b1111 -- арбитраж производится после 1024 передач DMA.</p> <p>Это означает, что арбитраж не производится, так как максимальное количество передач DMA равно 1024</p>
13...4	n_minus_1	<p>Перед выполнением цикла DMA, эти разряды указывают общее количество передач DMA, из которых состоит цикл DMA. Необходимо установить эти разряды в значение соответствующие размеру желаемого цикла DMA.</p> <p>10-разрядное число задаёт количество передач DMA минус 1.</p> <p>Возможные значения:</p> <p>b0000000000 = 1 передача DMA; b0000000001 = 1 передача DMA; b0000000010 = 2 передачи DMA; b0000000011 = 3 передачи DMA; b0000000100 = 4 передачи DMA; b0000000101 = 5 передач DMA; ... b1111111111 = 1024 передачи DMA.</p> <p>Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач DMA до завершения цикла DMA</p>
3	next_useburst	<p>Контролирует, не установлен ли chnl_useburst_set[C] в состояние 1, если контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации» и, если контроллер завершает цикл DMA, используя альтернативные управляющие данные.</p> <p>Примечание – Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер устанавливает chnl_useburst_set[C] в состояние 0, если количество оставшихся передач DMA меньше, чем 2^R. Установка next_useburst разряда определяет, будет ли контроллер дополнительно переопределять разряд chnl_useburst_set[C].</p> <p>Если контроллер выполняет цикл DMA в режиме работы с периферией «исполнение с изменением конфигурации», то после окончания цикла, использующего альтернативные управляющие</p>

Разряд	Имя	Описание
		<p>данные, происходит следующее в зависимости от состояния next_useburst:</p> <p>0 – контроллер не изменяет значение chnl_useburst_set[C]. Если chnl_useburst_set[C] установлен в 0, то для всех оставшихся циклов DMA в режиме работы с периферией «исполнение с изменением конфигурации», контроллер отвечает на запросы по dma_req[] и dma_sreq[], при выполнении циклов DMA он использует альтернативные управляющие данные;</p> <p>1 – контроллер изменяет значение chnl_useburst_set[C] в состояние 1. Поэтому для оставшихся циклов DMA в режиме работы с периферией «исполнение с изменением конфигурации», контроллер реагирует только на запросы по dma_req[], при выполнении циклов DMA он использует альтернативные управляющие данные</p>
2...0	cycle_ctrl	<p>Режим работы при выполнении цикла DMA:</p> <p>b000 – Стоп. Означает, что структура управляющих данных является «неправильной»;</p> <p>b001 – Основной. Контроллер должен получить новый запрос для окончания цикла DMA, перед этим он должен выполнить процедуру арбитража;</p> <p>b010 – Авто-запрос. Контроллер автоматически осуществляет запрос на обработку по соответствующему каналу в течение процедуры арбитража. Это означает, что начального запроса на обработку достаточно для выполнения цикла DMA;</p> <p>b011 – Пинг-понг. Контроллер выполняет цикл DMA используя одну из структур управляющих данных. По окончании выполнения цикла DMA, контроллер выполняет следующий цикл DMA, используя другую структуру. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя процессору перенастраивать неактивную структуру данных. Контроллер продолжает выполнять циклы DMA до тех пор, пока он не прочитает «неправильную» структуру данных или пока процессор не изменит cycle_ctrl поле в состояние b001 или b 010;</p> <p>b100 – Режим работы с памятью «исполнение с изменением конфигурации». Смотри соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть b100;</p> <p>b101 – Режим работы с памятью «исполнение с изменением конфигурации». Смотри соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть b101;</p> <p>b110 – Режим работы с периферией «исполнение с изменением конфигурации». Смотри соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть b110;</p> <p>b111 – Режим работы с периферией «исполнение с изменением конфигурации». Смотри соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть b111</p>

В начале цикла DMA или 2^R передачи DMA, контроллер считывает значение `channel_cfg` из системной памяти. После выполнения 2^R или N передач, он сохраняет обновленное значение `channel_cfg` в системную память.

Контроллер не поддерживает значений `dst_size` отличных от значений `src_size`. Если контроллер обнаруживает неравные значения этих полей, он использует значение `src_size` для размерности данных приемника и источника и при ближайшем обновлении поля `n_minus_1`, он также устанавливает значение поля `dst_size` равное `src_size`.

После выполнения контроллером N передач, контроллер устанавливает значение поля `cycle_ctrl` в `b000`, делая тем самым `channel_cfg` данные «неправильными». Это позволяет избежать повторения выполненной передачи DMA.

Вычисление адреса

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

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

`src_inc=b00 and dst_inc=b00`

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

`src_inc=b01 and dst_inc=b01`

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

`src_inc=b01 and dst_inc=b10`

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

`src_inc=b11 and dst_inc=b11`

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

Таблица 371 перечисляет адреса приемника цикла DMA для шести слов.

Таблица 371 – Цикла DMA для шести слов с пословным инкрементом

Начальные значения channel_cfg перед циклом DMA				
src_size=b10, dst_inc=b10, n_minus_1=b101, cycle_ctrl=1				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения channel_cfg после цикла DMA				
src_size=b10, dst_inc=b10, n_minus_1=0, cycle_ctrl=0				

* Это значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc.

Таблица 372 перечисляет адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.

Таблица 372 – Цикла DMA для 12 байт с «полусловным» инкрементом

Начальные значения channel_cfg перед циклом DMA				
src_size=b00, dst_inc=b01, n_minus_1=b1011, cycle_ctrl=1, R_power=b11				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x5E7	11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3
	0x5E7	9	0x12	0x5D5
	0x 5E7	8	0x10	0x5D7
	0x 5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
0x5E7	4	0x8	0x5DF	
Значения channel_cfg после 2 ^R передач DMA				
src_size=b00, dst_inc=b01, n_minus_1=b011, cycle_ctrl=1, R_power=b11				
DMA передачи	0x 5E7	3	0x6	0x5E1
	0x 5E7	2	0x4	0x5E3
	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7
Конечные значения channel_cfg после цикла DMA				
src_size=b00, dst_inc=b01, n_minus_1=0, cycle_ctrl=0**, R_power=b11				

* Это значение, полученное после сдвига влево значения счетчика на количество разрядов соответствующее dst_inc.

**После окончания цикла DMA, контроллер делает channel_cfg «неправильным» сбрасывая в 0 поле cycle_ctrl.

23.4 Описание регистров контроллера DMA

Данная глава описывает регистры контроллера и управление контроллером через них.

Глава содержит следующие разделы:

- о регистровой модели контроллера.
- описание регистров.

Основные положения регистровой модели контроллера:

- нужно избегать адресации при доступе к зарезервированным или неиспользованным адресам, так как это может привести к непредсказуемым результатам;

- необходимо записывать неиспользуемые или зарезервированные разряды регистров нулями и игнорировать значения таких разрядов при считывании, кроме случаев, специально описанных в разделе;

- системный сброс или сброс по установке питания сбрасывает все регистры в состояние 0, кроме случаев, специально описанных в разделе;

- все регистры поддерживают доступ по чтению и записи, кроме случаев, специально описанных в разделе. Доступ по записи обновляет содержание регистра, а доступ по чтению возвращает содержимое регистра.

Таблица 373 – Обобщенные данные о регистрах контроллера DMA

Наименование	Смещение относительно базового адреса	Тип	Значение по сбросу	Описание
dma_status	0x000	RO	0x-0nn0000*	Статусный регистр DMA
dma_cfg	0x004	WO	-	Регистр конфигурации DMA
ctrl_base_ptr	0x008	R/W	0x00000000	Регистр базового адреса управляющих данных каналов
alt_ctrl_base_ptr	0x00C	RO	0x000000nn**	Регистр базового адреса альтернативных управляющих данных каналов
dma_waitonreq_status	0x010	RO	0x00000000	Регистр статуса ожидания запроса на обработку каналов
chnl_sw_request	0x014	WO	-	Регистр программного запроса на обработку каналов
chnl_useburst_set	0x018	R/W	0x00000000	Регистр установки пакетного обмена каналов
chnl_useburst_clr	0x01C	WO	-	Регистр сброса пакетного обмена каналов
chnl_req_mask_set	0x020	R/W	0x00000000	Регистр маскирования запросов на обслуживание каналов
chnl_req_mask_clr	0x024	WO	-	Регистр очистки маскирования запросов на обслуживание каналов
chnl_enable_set	0x028	R/W	0x00000000	Регистр установки разрешения каналов

Наименование	Смещение относительно базового адреса	Тип	Значение по сбросу	Описание
chnl_enable_clr	0x02C	WO	-	Регистр сброса разрешения каналов
chnl_pri_alt_set	0x030	R/W	0x00000000	Регистр установки первичной/альтернативной структуры управляющих данных каналов
chnl_pri_alt_clr	0x034	WO	-	Регистр сброса первичной/альтернативной структуры управляющих данных каналов
chnl_priority_set	0x038	R/W	0x00000000	Регистр установки приоритета каналов
chnl_priority_clr	0x03C	WO	-	Регистр сброса приоритета каналов
-	0x040-0x048		-	зарезервировано
err_clr	0x04C	R/W	0x00000000	Регистр сброса флага ошибки

* Значение по сбросу зависит от количества каналов DMA, использованных в контроллере, а также от того, интегрирована ли схема тестирования;
 ** Значение по сбросу зависит от количества каналов DMA, использованных в контроллере

23.4.1 DMA_STATUS

Статусный регистр DMA

Данный регистр имеет доступ только на чтение. При чтении регистр возвращает состояние контроллера. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 105 показывает наименование разрядов этого регистра.

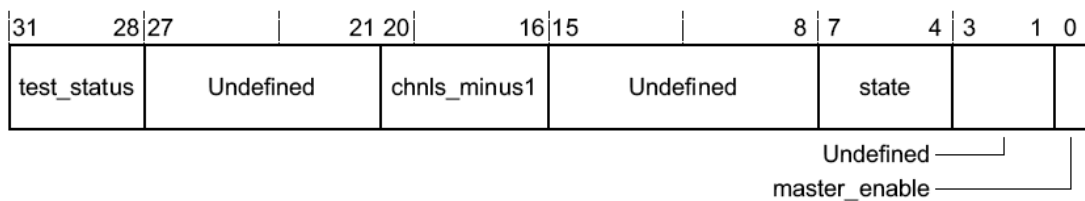


Рисунок 105 – Наименование разрядов регистра dma_status

Таблица 374 перечисляет назначение разрядов регистра.

Таблица 374 – Назначение разрядов регистра dma_status

Разряд	Наименование	Описание
31...28	test_status	Значение при чтении: 0x0 = контроллер не имеет интегрированной схемы тестирования; 0x1 = контроллер имеет интегрированную схему тестирования; 0x2 – 0xF = не определено
27...21	-	Не определено

Разряд	Наименование	Описание
20...16	chnls_minus1	Количество доступных каналов DMA минус 1. Например: b00000 = контроллер имеет 1 канал DMA; b00001 = контроллер имеет 2 канала DMA; b00010 = контроллер имеет 3 канала DMA; ... b11111 = контроллер имеет 32 канала DMA
15...8	-	Не определено
7...4	state	Текущее состояние автомата управления контроллера. Состояние может быть одним из следующих: b0000 = в покое; b0001 = чтение управляющих данных канала; b0010 = чтение указателя конца данных источника; b0011 = чтение указателя конца данных приемника; b0100 = чтение данных источника; b0101 = запись данных в приемник; b0110 = ожидание запроса на выполнение DMA; b0111 = запись управляющих данных канала; b1000 = приостановлен; b1001 = выполнен; b1010 = режим работы с периферией «исполнение с изменением конфигурации»; b1011-b1111 = не определено
3...1	-	Не определено
0	master_enable	Состояние контроллера: 0 = работа контроллера запрещена; 1 = работа контроллера разрешена

23.4.2 DMA_CFG

Регистр конфигурации DMA.

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера. Рисунок 106 показывает наименование разрядов этого регистра.



Рисунок 106 – Наименование разрядов регистра dma_cfg

Таблица 375 перечисляет назначение разрядов регистра.

Таблица 375 – Назначение разрядов регистра dma_cfg

Разряд	Наименование	Описание
31...8	-	Не определено, следует записывать 0
7...5	chnl_prot_ctrl	Определяет уровни индикации сигналов HPROT[3:1] защиты шины АНВ-Lite:

Разряд	Наименование	Описание
		<p>Разряд [7] – управляет сигналом HPROT[3], с целью индикации о появлении доступа с кэшированием;</p> <p>Разряд [6] – управляет сигналом HPROT[2], с целью индикации о появлении доступа с буферизацией;</p> <p>Разряд [5] – управляет сигналом HPROT[1], с целью индикации о появлении привилегированного доступа.</p> <p>Примечание – Если разряд[n] = 1, то соответствующий сигнал HPROT в состоянии 1.</p> <p>Если разряд[n] = 0, то соответствующий сигнал HPROT в состоянии 0</p>
4...1	-	Не определено, следует записывать 0
0	master_enable	<p>Определяет состояние контроллера:</p> <p>0 = запретить работу контроллера;</p> <p>1 = разрешить работу контроллера</p>

23.4.3 CTRL_BASE_PTR

Регистр базового адреса управляющих данных каналов.

Данный регистр имеет доступ на запись и чтение. Регистр определяет базовый адрес системной памяти размещения управляющих данных каналов.

Примечание:

Контроллер не содержит внутреннюю память для хранения управляющих данных каналов.

Размер системной памяти предназначенной контроллеру зависит от количества каналов DMA, использующихся контроллером, а также от возможности использования альтернативных управляющих данных каналов. Поэтому количество разрядов регистра необходимых для задания базового адреса варьируется, и зависит от варианта построения системы.

Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

Рисунок 107 показывает наименование разрядов этого регистра, в зависимости от количества использующихся каналов DMA.

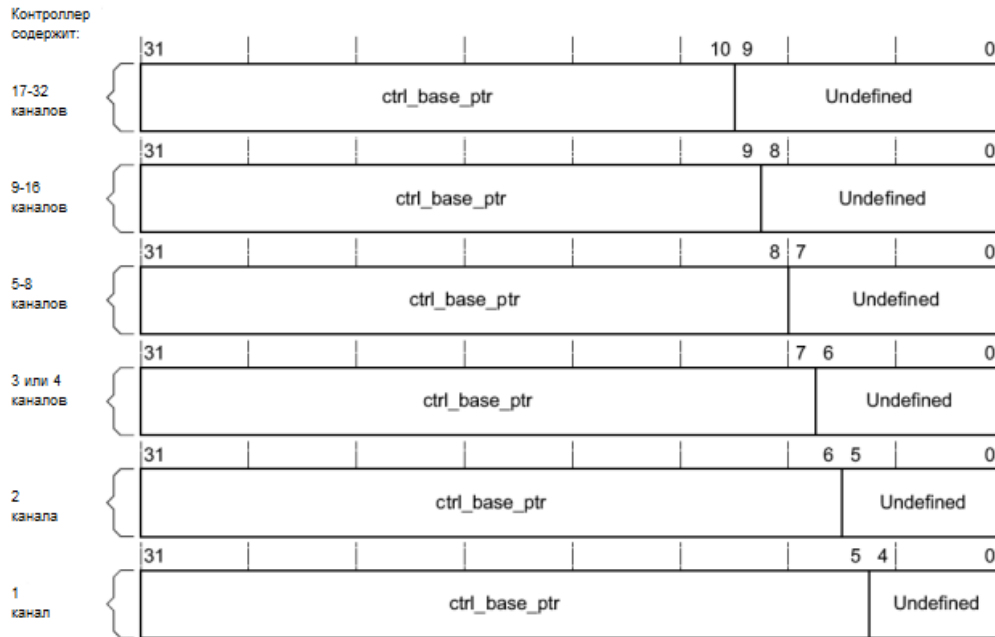


Рисунок 107 – Наименование разрядов регистра ctrl_base_ptr

Таблица 376 перечисляет назначение разрядов регистра.

Таблица 376 – Назначение разрядов регистра ctrl_base_ptr

Разряд	Наименование	Описание
[31:DMA_CHNL_BITS+5]	ctrl_base_ptr	Указатель на базовый адрес первичной структуры управляющих данных. См. соответствующий раздел
[DMA_CHNL_BITS+4:0]	-	Не определено, следует записывать 0

где DMA_CHNL_BITS – минимальное число разрядов необходимых для представления количества использующихся каналов минус 1. Значения, которые могут присваиваться DMA_CHNL_BITS, следующие:

- 0 – контроллер содержит 1 канал DMA;
- 1 – контроллер содержит 2 канала DMA;
- 2 – контроллер содержит 3 или 4 канала DMA;
- 3 – контроллер содержит от 5 до 8 каналов DMA;
- 4 – контроллер содержит от 9 до 16 каналов DMA;
- 5 – контроллер содержит от 17 до 32 каналов DMA.

23.4.4 ALT_CTRL_BASE_PTR

Регистр базового адреса альтернативных управляющих данных каналов.

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении указатель базового адреса альтернативных управляющих данных каналов. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 108 показывает наименование разрядов этого регистра. Этот регистр позволяет не производить вычисления базового адреса альтернативных управляющих данных каналов.

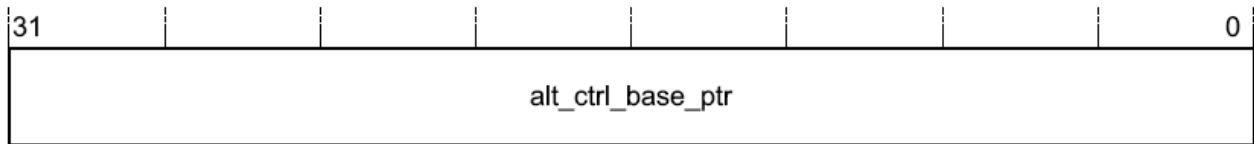


Рисунок 108 – Наименование разрядов регистра alt_ctrl_base_ptr

Таблица 377 перечисляет назначение разрядов регистра.

Таблица 377 – Назначение разрядов регистра alt_ctrl_base_ptr

Разряд	Наименование	Описание
31...0	alt_ctrl_base_ptr	Указатель на базовый адрес альтернативной структуры управляющих данных

23.4.5 DMA_WAITONREQ_STATUS

Регистр статуса ожидания запроса на обработку каналов.

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении состояние сигналов dma_waitonreq[]. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Рисунок 109 показывает наименование разрядов этого регистра.

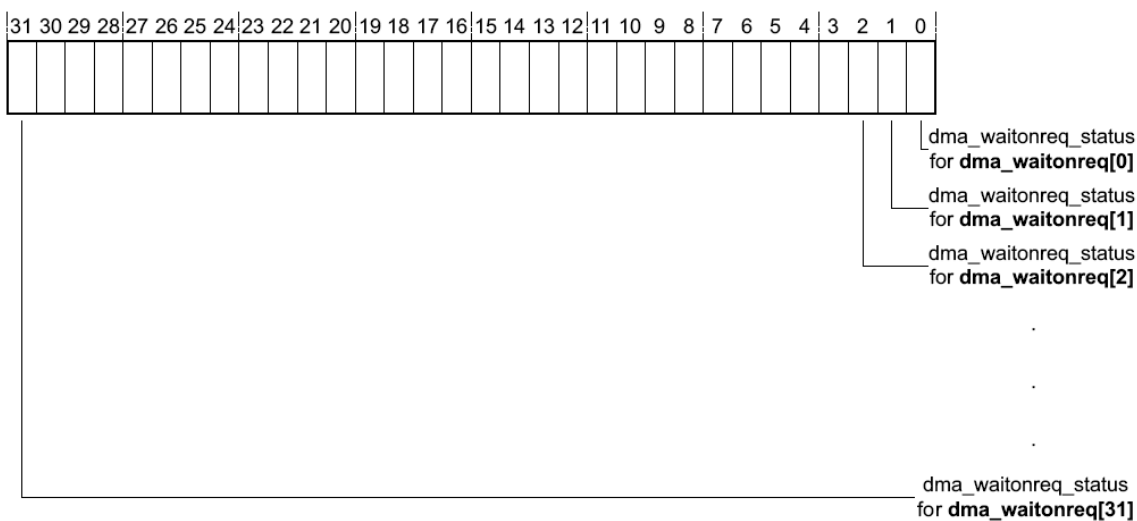


Рисунок 109 – Наименование разрядов регистра dma_waitonreq_status

Таблица 378 перечисляет назначение разрядов регистра.

Таблица 378 – Назначение разрядов регистра dma_waitonreq_status

Разряд	Наименование	Описание
31...0	dma_waitonreq_status	Состояние сигналов ожидания запроса на обработку каналов DMA. Если при чтении: Разряд [C] =0 означает, что dma_waitonreq[C] в состоянии 0; Разряд [C] =1 означает, что dma_waitonreq[C] в состоянии 1

23.4.6 CHNL_SW_REQUEST

Регистр программного запроса на обработку каналов.

Данный регистр имеет доступ только на запись. Регистр позволяет устанавливать программно запрос на выполнение цикла DMA. Рисунок 110 показывает наименование разрядов этого регистра.

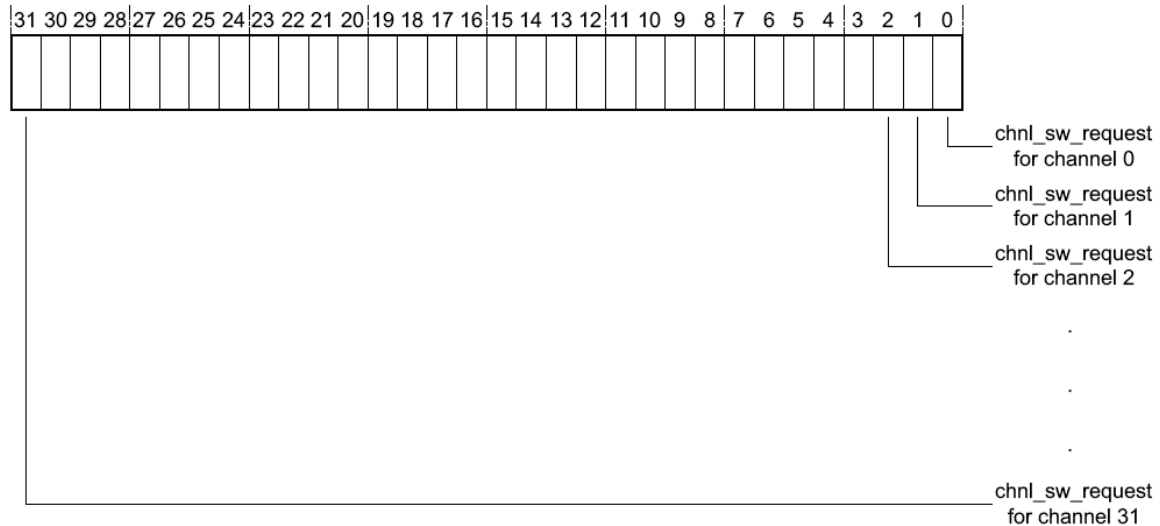


Рисунок 110 – Наименование разрядов регистра chnl_sw_request

Таблица 379 перечисляет назначение разрядов регистра.

Таблица 379 – Назначение разрядов регистра chnl_sw_request

Разряд	Наименование	Описание
31...0	chnl_sw_request	<p>Устанавливает соответствующий разряд для генерации программного запроса на выполнение цикла DMA по соответствующему каналу DMA.</p> <p>Если при записи:</p> <p>Разряд [C] =0 означает, что запрос на выполнение цикла DMA по каналу C не будет установлен;</p> <p>Разряд [C] =1 означает, что запрос на выполнение цикла DMA по каналу C будет установлен.</p> <p>Запись разряда, соответствующего нереализованному каналу, означает, что запрос на выполнение цикла DMA не будет установлен</p>

23.4.7 CHNL_USEBURST_SET

Регистр установки пакетного обмена каналов.

Данный регистр имеет доступ на чтение и запись. Регистр отключает выполнение одиночных запросов по установке dma_sreq[] и поэтому, будут обрабатываться и исполняться только запросы по dma_req[]. Регистр возвращает при чтении состояние установок пакетного обмена. Рисунок 111 показывает наименование разрядов этого регистра.

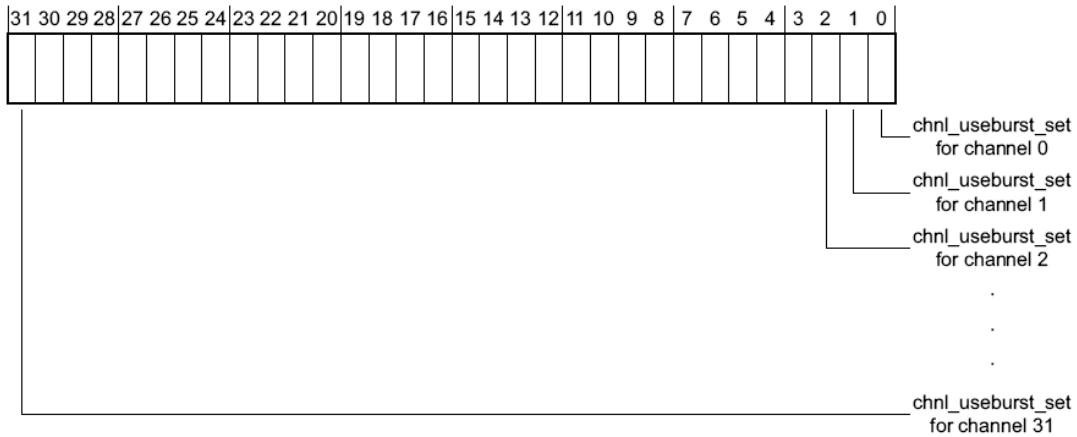


Рисунок 111 – Наименование разрядов регистра chnl_useburst_set

Таблица 380 перечисляет назначение разрядов регистра.

Таблица 380 – Назначение разрядов регистра chnl_useburst_set

Разряд	Наименование	Описание
31...0	chnl_useburst_set	<p>Отключает обработку запросов на выполнение циклов DMA от dma_sreq[] и возвращает при чтении состояние этих настроек.</p> <p>Если при чтении:</p> <p>Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные от dma_sreq[] и dma_req[]. Контроллер выполняет одиночные передачи или 2^R передач;</p> <p>Разряд [C] = 1 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные только от dma_req[]. Контроллер выполняет 2^R передач.</p> <p>При записи:</p> <p>Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_clr регистр и установить соответствующий разряд C в 0;</p> <p>Разряд [C] = 1 отключает возможность обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Контроллер выполняет 2^R передач.</p> <p>Запись разряда, соответствующего нереализованному каналу не дает никакого эффекта</p>

После выполнения предпоследней передачи из 2^R передач, в том случае если число оставшихся передач (N) меньше чем 2^R, контроллер сбрасывает разряд chnl_useburst_set в 0. Это позволяет выполнять оставшиеся передачи, используя dma_sreq[] и dma_req[].

Примечание – При программировании channel_cfg значением N меньшим, чем 2^R, запрещена установка соответствующего разряда chnl_useburst_set, в случае если периферийный блок не поддерживает сигнал dma_req[].

В режиме работы с периферией «исполнение с изменением конфигурации», если разряд next_useburst установлен в channel_cfg, то контроллер устанавливает chnl_useburst_set [C] в 1, после окончания цикла DMA, использующего альтернативные управляющие данные.

23.4.8 CHNL_USEBURST_CLR

Регистр сброса пакетного обмена каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает выполнение одиночных запросов по установке dma_sreq[]. Рисунок 112 показывает наименование разрядов этого регистра.

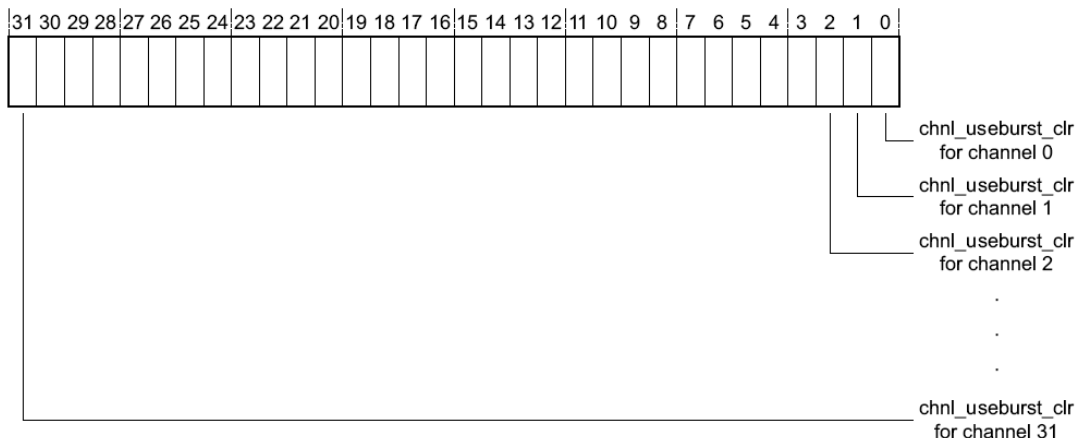


Рисунок 112 – Наименование разрядов регистра chnl_useburst_clr

Таблица 381 перечисляет назначение разрядов регистра.

Таблица 381 – Назначение разрядов регистра chnl_useburst_clr

Разряд	Наименование	Описание
31...0	chnl_useburst_clr	Установка соответствующего разряда разрешает обработку запросов на выполнение циклов DMA от dma_sreq[]. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_useburst_set регистр для отключения обработки запросов от dma_sreq[]; Разряд [C] =1 разрешает обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Запись разряда, соответствующего нереализованному каналу не дает никакого эффекта

23.4.9 CHNL_REQ_MASK_SET

Регистр маскирования запросов на обслуживание каналов.

Данный регистр имеет доступ на чтение и запись. Регистр отключает установку запросов на выполнение циклов DMA на `dma_sreq[]` и `dma_req[]`. Регистр возвращает при чтении состояние установок маскирования запросов от `dma_sreq[]` и `dma_req[]` на обслуживание каналов. Рисунок 113 показывает наименование разрядов этого регистра.

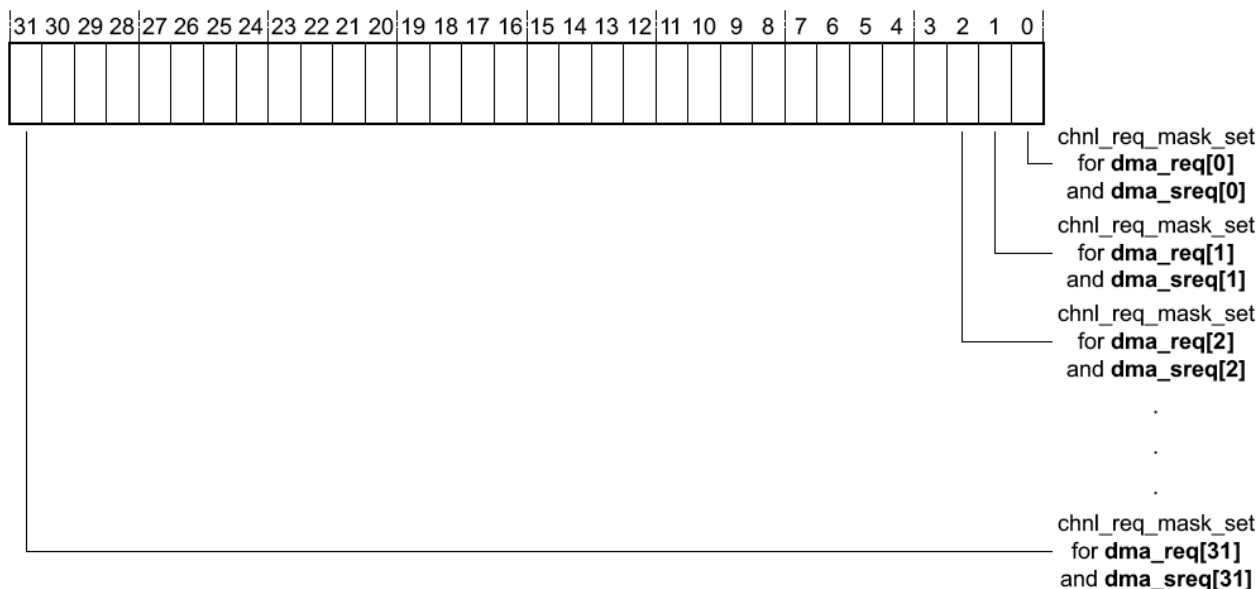


Рисунок 113 – Наименование разрядов регистра `chnl_req_mask_set`

Таблица 382 перечисляет назначение разрядов регистра.

Таблица 382 – Назначение разрядов регистра `chnl_req_mask_set`

Разряд	Наименование	Описание
31...0	<code>chnl_req_mask_set</code>	<p>Отключает обработку запросов по <code>dma_sreq[]</code> и <code>dma_req[]</code> на выполнение циклов DMA от каналов и возвращает при чтении состоянии этих настроек.</p> <p>Если при чтении:</p> <p>Разряд [C] =0 означает, что канал DMA C выполняет циклы DMA в ответ на запросы;</p> <p>Разряд [C] =1 означает, что канал DMA C не выполняет циклы DMA в ответ на запросы.</p> <p>При записи:</p> <p>Разряд [C] =0 не дает эффекта. Необходимо использовать <code>chnl_req_mask_clr</code> регистр для разрешения установки запросов;</p> <p>Разряд [C] =1 отключает установку запросов на выполнение циклов DMA, по <code>dma_sreq[]</code> и <code>dma_req[]</code>.</p> <p>Запись разряда, соответствующего нереализованному каналу не дает никакого эффекта</p>

23.4.10 CHNL_REQ_MASK_CLR

Регистр очистки маскирования запросов на обслуживание каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает установку запросов на выполнение циклов DMA на `dma_sreq[]` и `dma_req[]`. Рисунок 114 показывает наименование разрядов этого регистра.

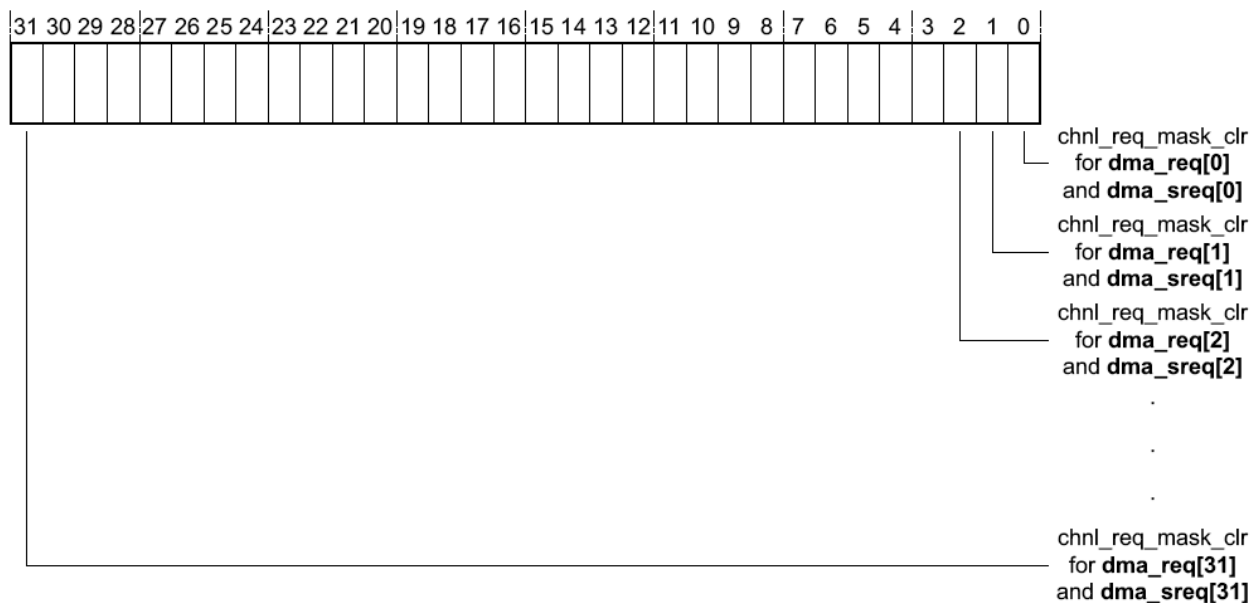


Рисунок 114 – Наименование разрядов регистра `chnl_req_mask_clr`

Таблица 383 перечисляет назначение разрядов регистра.

Таблица 383 – Назначение разрядов регистра `chnl_req_mask_clr`

Разряд	Наименование	Описание
31...0	<code>chnl_req_mask_clr</code>	<p>Установка соответствующего разряда разрешает установку запросов по <code>dma_sreq[]</code> и <code>dma_req[]</code> на выполнение циклов DMA от каналов.</p> <p>При записи:</p> <p>Разряд [C] =0 не дает эффекта. Необходимо использовать <code>chnl_req_mask_set</code> регистр для отключения установки запросов;</p> <p>Разряд [C] =1 разрешает установку запросов на выполнение циклов DMA, по <code>dma_sreq[]</code> и <code>dma_req[]</code>.</p> <p>Запись разряда, соответствующего нереализованному каналу не дает никакого эффекта</p>

23.4.11 CHNL_ENABLE_SET

Регистр установки разрешения каналов.

Данный регистр имеет доступ на чтение и запись. Регистр разрешает работу каналов DMA. Регистр возвращает при чтении состояние разрешений работы каналов DMA. Рисунок 115 показывает наименование разрядов этого регистра.

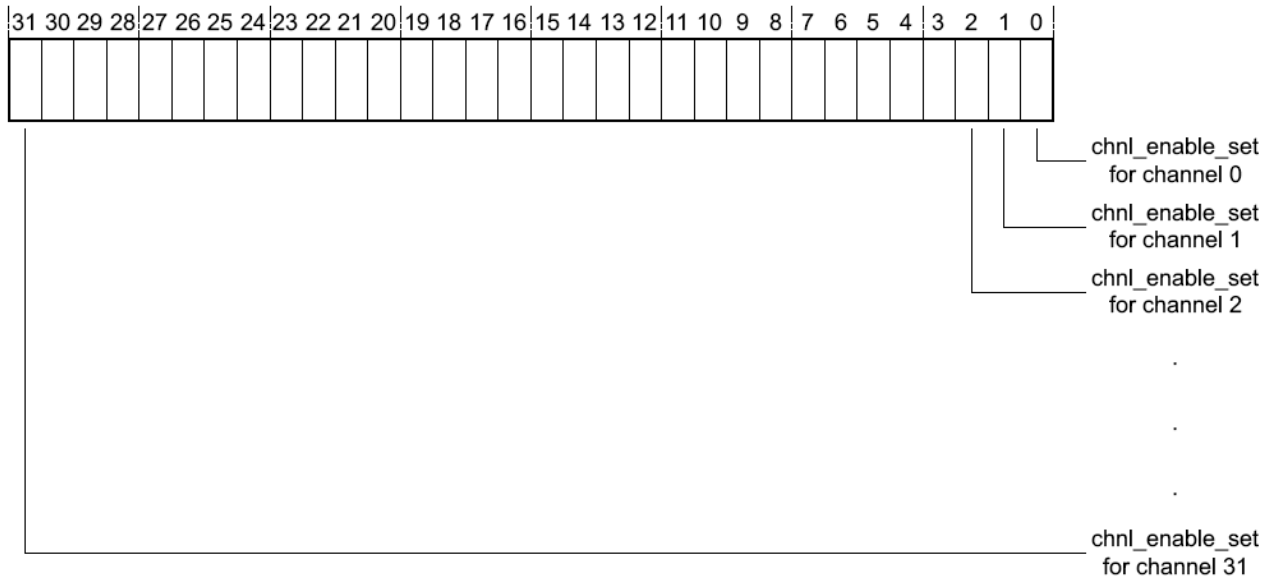


Рисунок 115 – Наименование разрядов регистра chnl_enable_set

Таблица 384 перечисляет назначение разрядов регистра.

Таблица 384 – Назначение разрядов регистра chnl_enable_set

Разряд	Наименование	Описание
31...0	chnl_enable_set	<p>Разрешает работу каналов DMA и возвращает при чтении состоянии этих настроек.</p> <p>Если при чтении:</p> <p>Разряд [C] =0 означает, что канал DMA C отключен;</p> <p>Разряд [C] =1 означает, что работа канала DMA C разрешена.</p> <p>При записи:</p> <p>Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_enable_clr регистр для отключения канала;</p> <p>Разряд [C] =1 разрешает работу канала DMA C.</p> <p>Запись разряда соответствующего нереализованному каналу не дает никакого эффекта</p>

23.4.12 CHNL_ENABLE_CLR

Регистр сброса разрешения каналов.

Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов DMA. Рисунок 116 показывает наименование разрядов этого регистра.

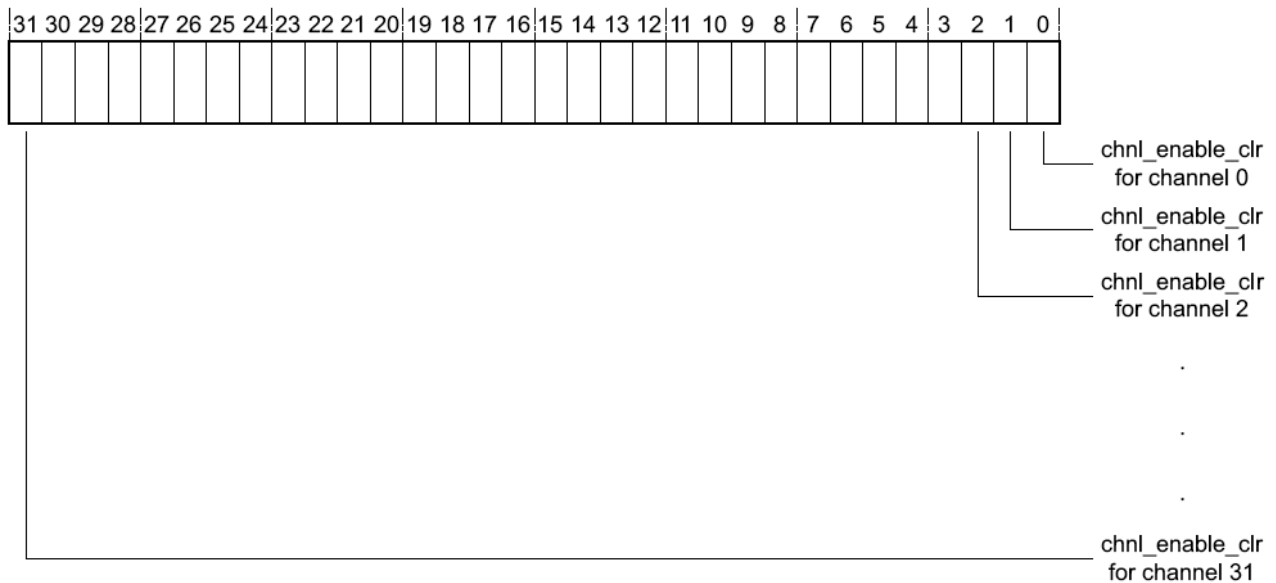


Рисунок 116 – Наименование разрядов регистра chnl_enable_clr

Таблица 385 перечисляет назначение разрядов регистра.

Таблица 385 – Назначение разрядов регистра chnl_enable_clr

Разряд	Наименование	Описание
31...0	chnl_enable_clr	<p>Установка соответствующего разряда запрещает работу соответствующего канала DMA.</p> <p>При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_enable_set регистр для разрешения работы канала; Разряд [C] =1 запрещает работу канала DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p> <p>Примечание – Контроллер может отключить канал DMA, установив соответствующий разряд в следующих случаях: – при завершении цикла DMA; – при чтении из channel_cfg с полем cycle_ctrl установленным в b000; – при появлении ошибки на шине АНВ-Lite</p>

23.4.13 CHNL_PRI_ALT_SET

Регистр установки первичной/альтернативной структуры управляющих данных каналов.

Данный регистр имеет доступ на запись и чтение. Регистр разрешает работу канала DMA с использованием альтернативной структуры управляющих данных. Чтение регистра возвращает состояние каналов DMA (какую структуру управляющих данных использует каждый канал DMA). Рисунок 117 показывает наименование разрядов этого регистра.

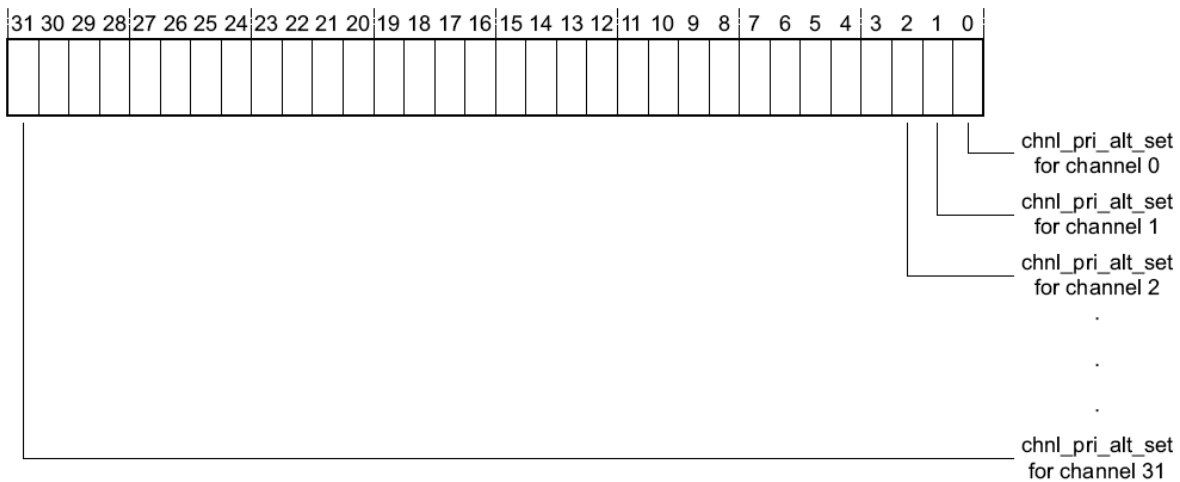


Рисунок 117 – Наименование разрядов регистра chnl_pri_alt_set

Таблица 386 перечисляет назначение разрядов регистра.

Таблица 386 – Назначение разрядов регистра chnl_pri_alt_set

Разряд	Наименование	Описание
31...0	chnl_pri_alt_set	<p>Установка соответствующего разряда подключает использование альтернативных управляющих данных для соответствующего канала DMA, чтение возвращает состояние этих настроек.</p> <p>Если при чтении:</p> <p>Разряд [C] = 0 означает, что канал DMA C использует первичную структуру управляющих данных;</p> <p>Разряд [C] = 1 означает, что канал DMA C использует альтернативную структуру управляющих данных.</p> <p>При записи:</p> <p>Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_clr регистр для сброса разряда [C] в 0;</p> <p>Разряд [C] = 1 подключает использование альтернативной структуры управляющих данных каналом DMA C.</p> <p>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.</p> <p>Примечание – Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях:</p> <ul style="list-style-type: none"> – при завершении четырех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с

Разряд	Наименование	Описание
		изменением конфигурации»; – при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме Пинг-понг; – при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: – пинг-понг; – работа с памятью «исполнение с изменением конфигурации»; – работа с периферией «исполнение с изменением конфигурации»

23.4.14 CHNL_PRI_ALT_CLR

Регистр сброса первичной/альтернативной структуры управляющих данных каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает работу канала DMA с использованием первичной структуры управляющих данных. Рисунок 118 показывает наименование разрядов этого регистра.

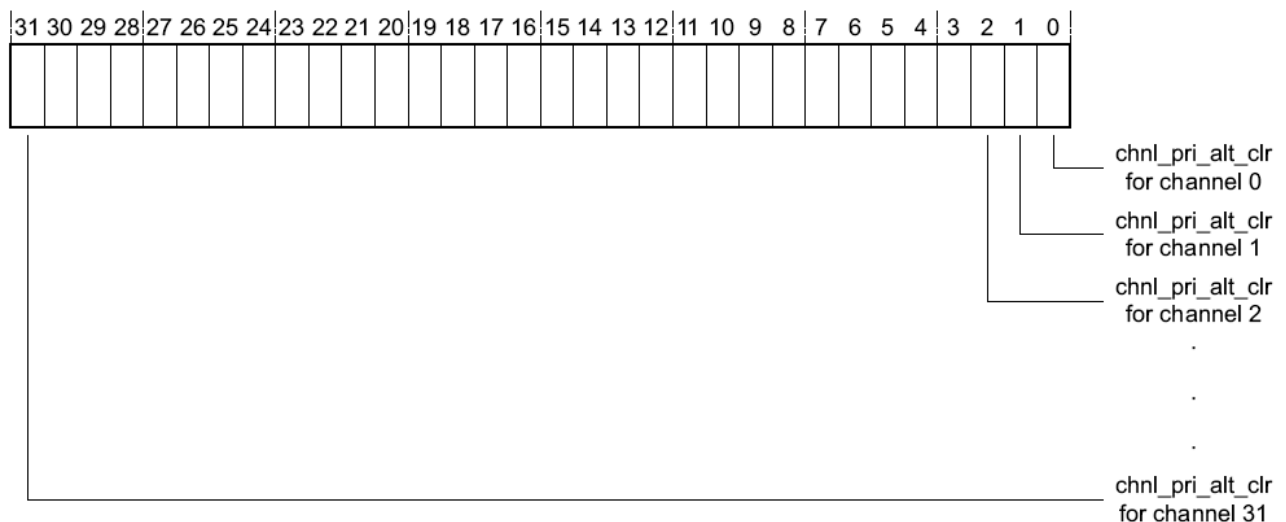


Рисунок 118 – Наименование разрядов регистра chnl_pri_alt_clr

Таблица 387 перечисляет назначение разрядов регистра.

Таблица 387 – Назначение разрядов регистра chnl_pri_alt_clr

Разряд	Наименование	Описание
31...0	chnl_pri_alt_clr	Установка соответствующего разряда подключает использование первичных управляющих данных для соответствующего канала DMA. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_pri_alt_set регистр для выбора альтернативных управляющих данных; Разряд [C] =1 подключает использование первичной структуры управляющих данных каналом DMA C. Запись разряда, соответствующего нереализованному каналу, не

Разряд	Наименование	Описание
		<p>дает никакого эффекта.</p> <p>Примечание – Контроллер может переключить значение разряда <code>chnl_pri_alt_clr[C]</code> в следующих случаях:</p> <ul style="list-style-type: none"> – при завершении четырех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; – при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме Пинг-понг; – при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: <ul style="list-style-type: none"> – пинг-понг; – работа с памятью «исполнение с изменением конфигурации»; – работа с периферией «исполнение с изменением конфигурации»

23.4.15 CHNL_PRIORITY_SET

Регистр установки приоритета каналов.

Данный регистр имеет доступ на запись и чтение. Регистр позволяет присвоить высокий приоритет каналу DMA. Чтение регистра возвращает состояние приоритета каналов DMA. Рисунок 119 показывает наименование разрядов этого регистра.

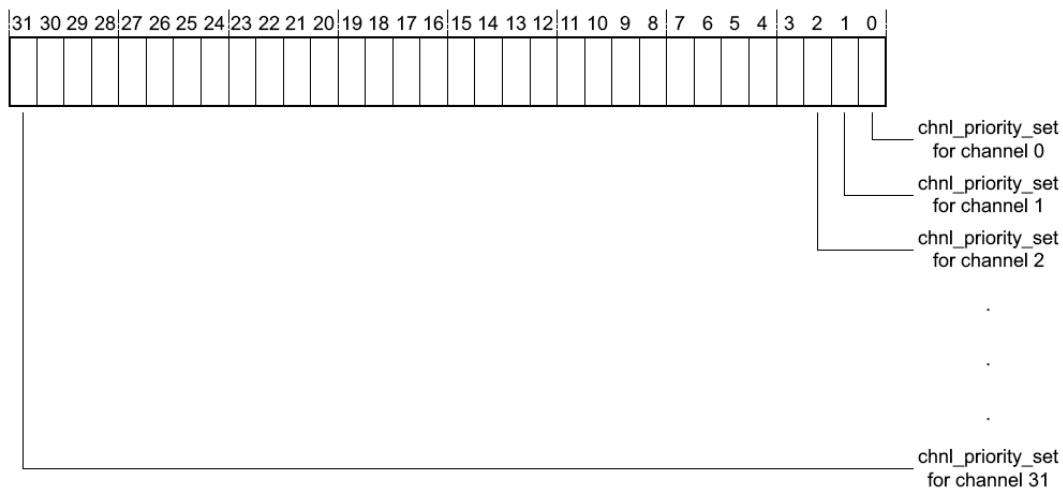


Рисунок 119 – Наименование разрядов регистра `chnl_priority_set`

Таблица 388 перечисляет назначение разрядов регистра.

Таблица 388 – Назначение разрядов регистра `chnl_priority_set`

Разряд	Наименование	Описание
31...0	<code>chnl_priority_set</code>	<p>Установка высокого приоритета каналу DMA, чтение возвращает состояние приоритета каналов DMA.</p> <p>Если при чтении:</p> <p>Разряд <code>[C]</code> =0 означает, что каналу DMA <code>C</code> присвоен уровень приоритета по умолчанию;</p>

		<p>Разряд [C] =1 означает, что каналу DMA C присвоен высокий уровень приоритета.</p> <p>При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_priority_clr регистр для установки каналу C уровня приоритета по умолчанию; Разряд [C] =1 устанавливает каналу DMA C высокий уровень приоритета.</p> <p>Запись разряда соответствующего нереализованному каналу не дает никакого эффекта</p>
--	--	--

23.4.16 CHNL_PRIORITY_CLR

Регистр сброса приоритета каналов.

Данный регистр имеет доступ только на запись. Регистр позволяет присвоить каналу DMA уровень приоритета по умолчанию. Рисунок 120 показывает наименование разрядов этого регистра.

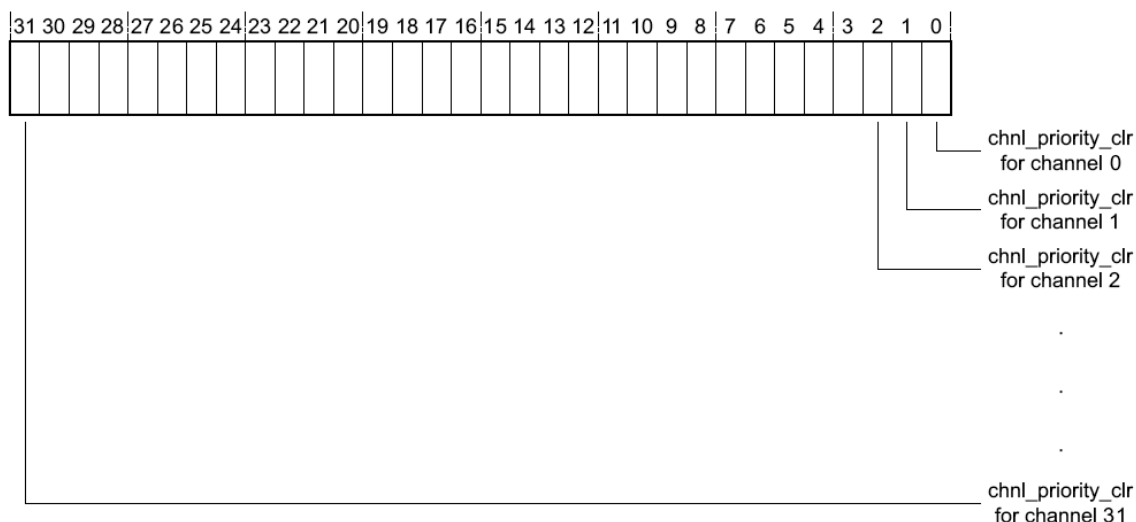


Рисунок 120 – Наименование разрядов регистра chnl_priority_clr

Таблица 389 перечисляет назначение разрядов регистра.

Таблица 389 – Назначение разрядов регистра chnl_priority_clr

Разряд	Наименование	Описание
31...0	chnl_priority_clr	<p>Установка разряда присваивает соответствующему каналу DMA уровень приоритета по умолчанию.</p> <p>При записи: Разряд [C] = не дает эффекта. Необходимо использовать chnl_priority_set регистр для установки каналу C высокого уровня приоритета; Разряд [C] = 1 устанавливает каналу DMA C уровень приоритета по умолчанию.</p> <p>Запись разряда, соответствующего нереализованному каналу не дает никакого эффекта</p>

23.4.17 ERR_CLR

Регистр сброса флага ошибки.

Данный регистр имеет доступ на запись и чтение. Регистр позволяет сбрасывать сигнал dma_err в 0. Чтение регистра возвращает состояние сигнала dma_err. Рисунок 121 показывает наименование разрядов этого регистра.

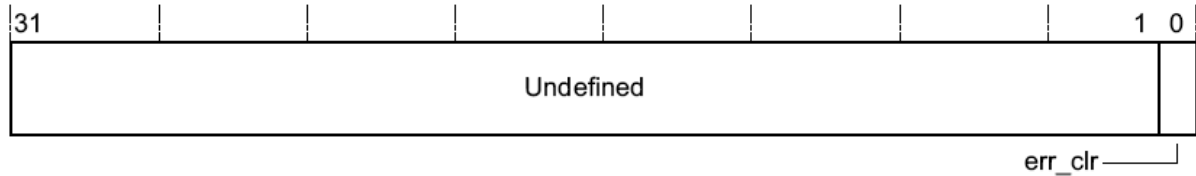


Рисунок 121 – Наименование разрядов регистра err_clr

Таблица 390 перечисляет назначение разрядов регистра.

Таблица 390 – Назначение разрядов регистра err_clr

Разряд	Наименование	Описание
31...1	-	Не определено. Следует записывать 0
0	chnl_priority_set	<p>Установка сигнала в состояние 0, чтение возвращает состояние сигнала (флага) dma_err.</p> <p>Если при чтении:</p> <p>Разряд [C] =0 означает, что dma_err находится в состоянии 0;</p> <p>Разряд [C] =1 означает, что dma_err находится в состоянии 1.</p> <p>При записи:</p> <p>Разряд [C] =0 не дает эффекта. Состояние dma_err останется неизменным;</p> <p>Разряд [C] =1 сбрасывает сигнал (флаг) dma_err в состояние 0.</p> <p>Для целей тестирования возможно использовать регистр err_set, чтобы установить сигнал dma_err в состояние 1.</p> <p>Примечание – При сбросе сигнала dma_err одновременно с появлением ошибки на шине АНВ-Lite, то приоритет отдается ошибке и следовательно, значение регистра (и dma_err) останется неизменным (несброшенным)</p>

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 начали 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 специальную форму исключения. Когда выставляется RESET, работа процессора останавливается потенциально в любой точке инструкций. Когда RESET убирается, выполнение перезапускается с адреса заданного 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 391 – 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 392 – Прерывания, формируемые периферийными блоками

Прерывания	Блок	Принцип формирования
IRQ0	DMA	Прерывания от DMA DMA_ERR или DMA_DONE. Обработка прерываний от DMA в соответствии с разделом Error signaling технического описания DMA

Прерывания	Блок	Принцип формирования
IRQ1	UART0	Сигнал UARTINTR
IRQ2	UART1	Сигнал UARTINTR
IRQ3	SSP	Сигнал SSPINTR
IRQ4	POWER	Сигнал прерывания от POWER Detecor
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	-	-
IRQ10	BACKUP	Прерывание от ВКР и часов реального времени
IRQ11	Внешнее прерывание 1	Сигнал EXT_INT0 Вывод PA[10] в основном режиме
IRQ12	Внешнее прерывание 2	Сигнал EXT_INT1 Вывод PC[4], PB[6] в альтернативном режиме
IRQ13	Внешнее прерывание 3	Сигнал EXT_INT2 Вывод PC[5], PB[7] в альтернативном режиме
IRQ14	ADCUI	Прерывание от АЦП для измерения напряжений и токов
IRQ15	ADCUI	Прерывание от АЦП для измерения напряжений и токов
IRQ16	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, SVCcall, SysTick и Hard Fault.

Таблица векторов

Таблица векторов содержит указатель стека, вектор входа по RESET и стартовые адреса обработчиков, также называемых векторами. На рисунке 122 представлена последовательность векторов в таблице. Младший бит всех векторов должен быть равен 1, указывая на то, что обработчик выполняется в Thumb-режиме.

Exception number	IRQ number	Vector	Offset
16+n	n	IRQn	0x40+4n
.	.	.	.
.	.	.	.
.	.	.	.
18	2	IRQ2	0x48
17	1	IRQ1	0x44
16	0	IRQ0	0x40
15	-1	SysTick, if implemented	0x3C
14	-2	PendSV	0x38
13		Reserved	
12			
11	-5	SVCall	0x2C
10			
9			
8			
7		Reserved	
6			
5			
4			
3	-13	HardFault	0x10
2	-14	NMI	0x0C
		Reset	0x08
1		Initial SP value	0x04
			0x00

Рисунок 122 –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. Подробнее данный вопрос рассмотрен в разделе «Вход в процедуру обработки исключения».

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. Более подробная информация дана в разделе «Возврат из обработчика исключения».

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 (см. пункт 11.4.9 «PRIMASK»). 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 значение 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 пункте 24.5.4 «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 VX 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 393.

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 393 – 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 Cortex-M0 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 режим 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. Более подробная информация представлена на стр. 165.

25.3 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 работу без перехода в режим ожидания.

25.4 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».

25.5 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.

25.6 Выход из состояния ожидания

Условия выхода процессора из режима ожидания зависят от причины, по которой он был переведен в этот режим.

25.7 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».

25.8 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 394.

Таблица 394 – 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.

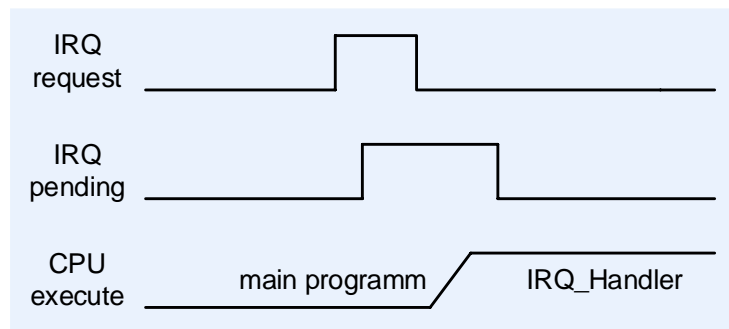


Рисунок 123 – Выставление отложенного запроса на прерывание и последующая его обработка

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 124.

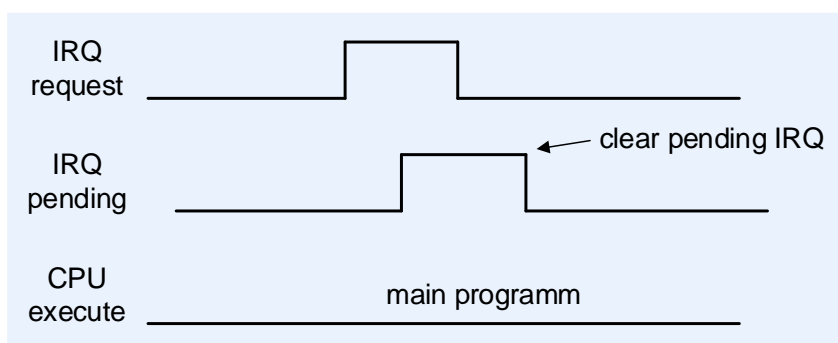


Рисунок 124 – Сброс признака отложенного прерывания, до обработки ядром

Если произойдет снятие запроса IRQ request от источника, «защелкивание» признака отложенного прерывания гарантирует отработку его ядром в соответствии с приоритетом. – рисунок 125. Сам IRQ pending признак снимается автоматически, когда прерывание становится активным, о чем сигнализирует признак IRQ active.

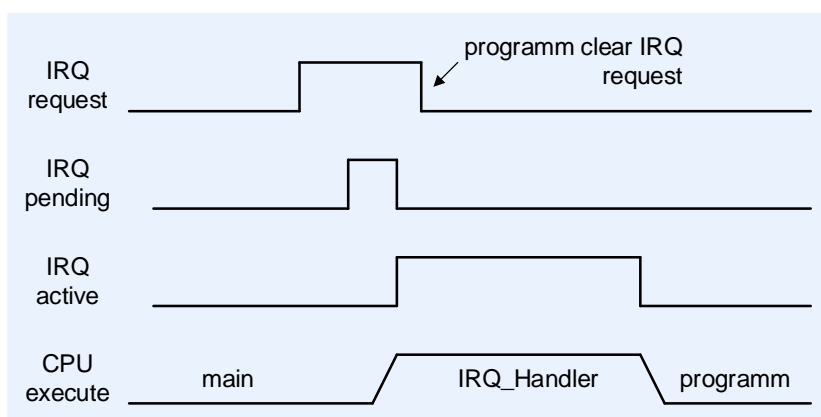


Рисунок 125 – Сброс признака отложенного прерывания, до обработки ядром

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 126.

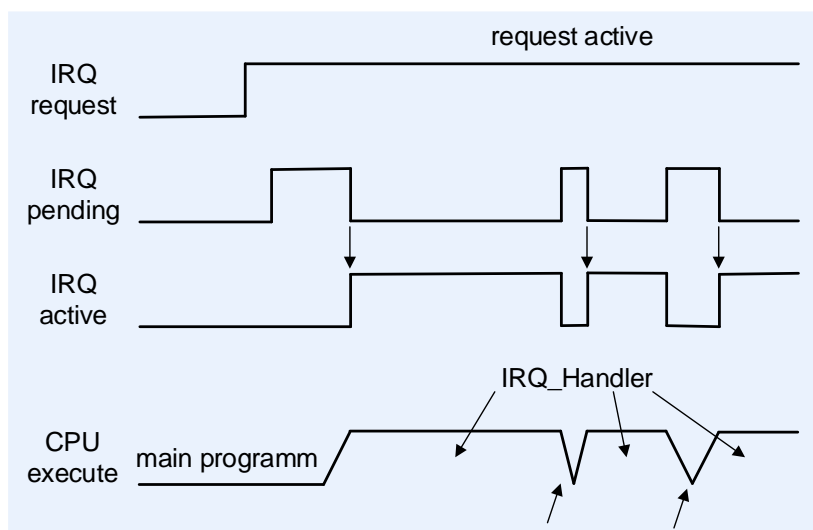


Рисунок 126 – Повторная обработка прерываний при удержании запроса от источника

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 127.

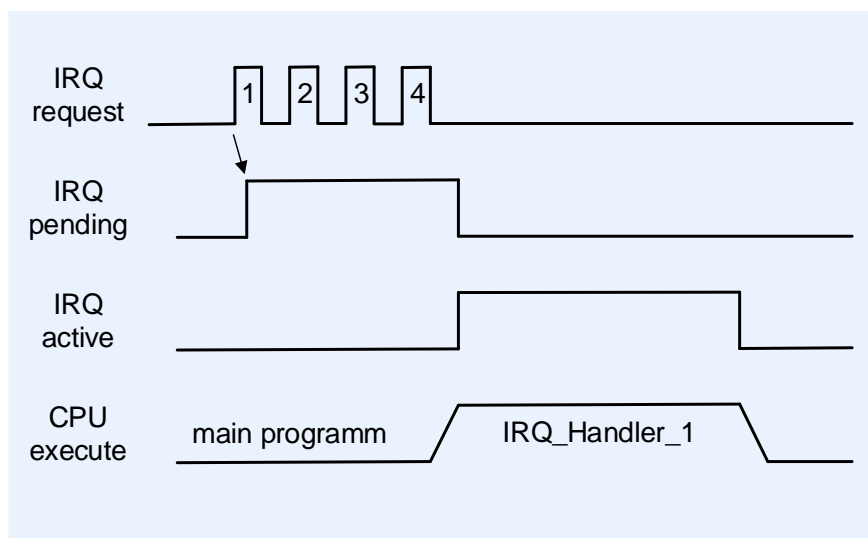


Рисунок 127 – Многократная установка снятие запроса IRQ request

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 127, 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 128.

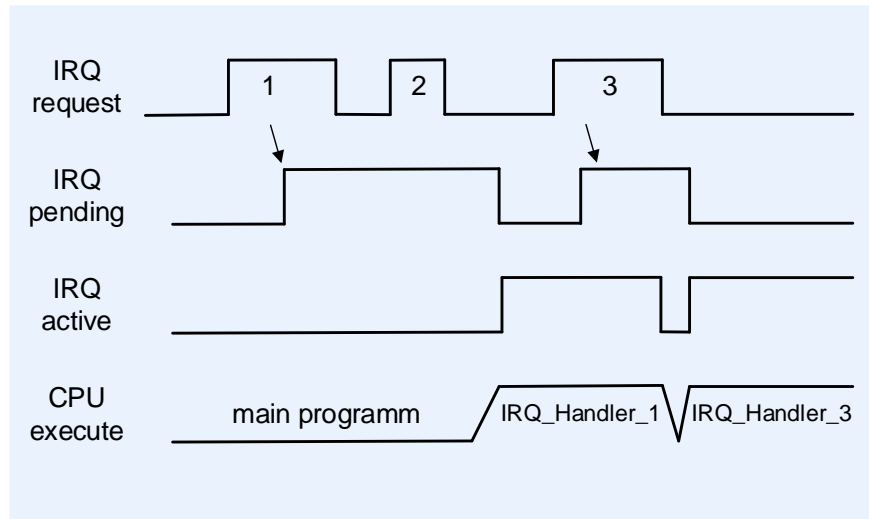


Рисунок 128 – Повторная установка запроса на прерывание в момент выполнения обработчика исключения

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).

Распределение бит регистра ISER представлено на рисунке 129:

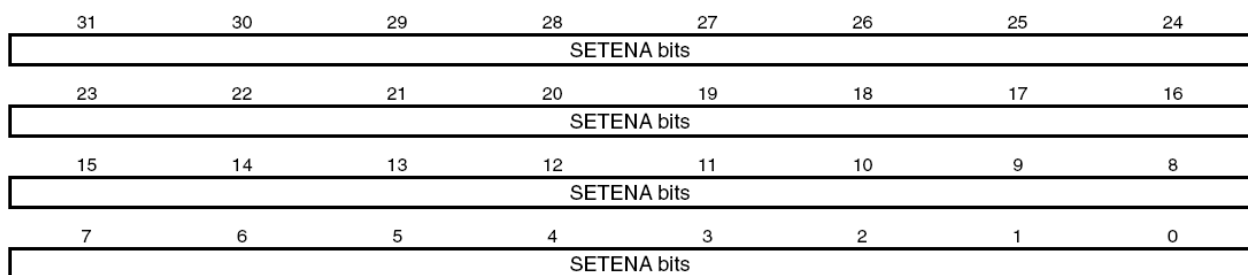


Рисунок 129 – Распределение бит регистра ISER

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).

Распределение бит регистра ICER представлено на рисунке 130:

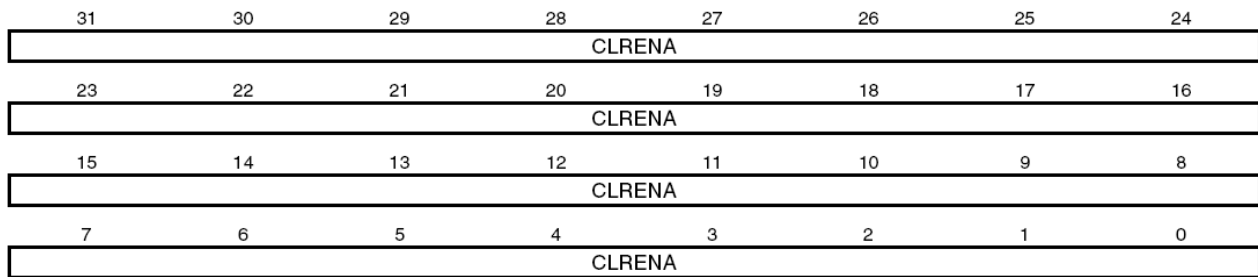


Рисунок 130 – Распределение бит регистра ICER

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).

Распределение бит регистра ISPR представлено на рисунке 131:

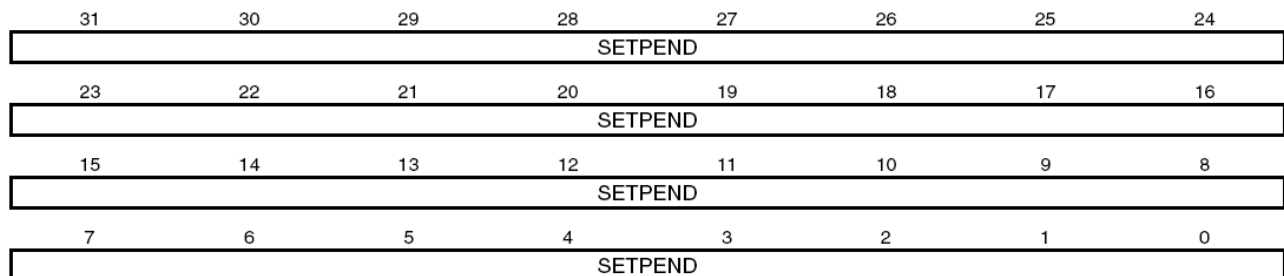


Рисунок 131 – Распределение бит регистра ISPR

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).

Распределение бит регистра ICPR представлено на рисунке 132:

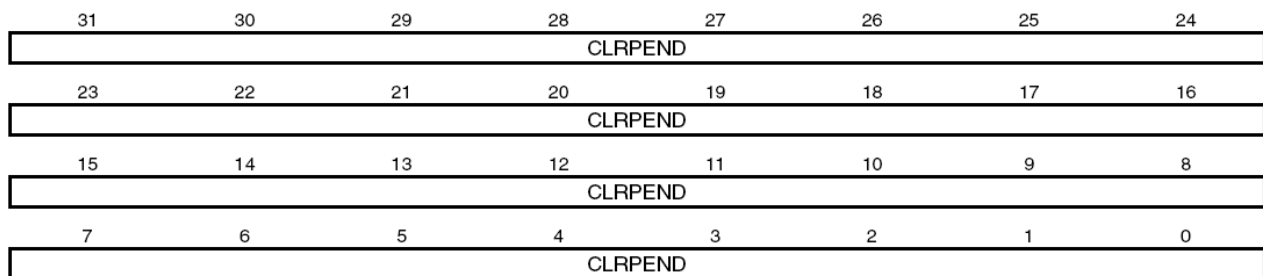


Рисунок 132 – Распределение бит регистра ICPR

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.

Каждый из регистров содержит четыре поля приоритета, которые отображаются на четыре элемента массива PRI[0] ... PRI[31] CMSIS, как показано на рисунке 133.

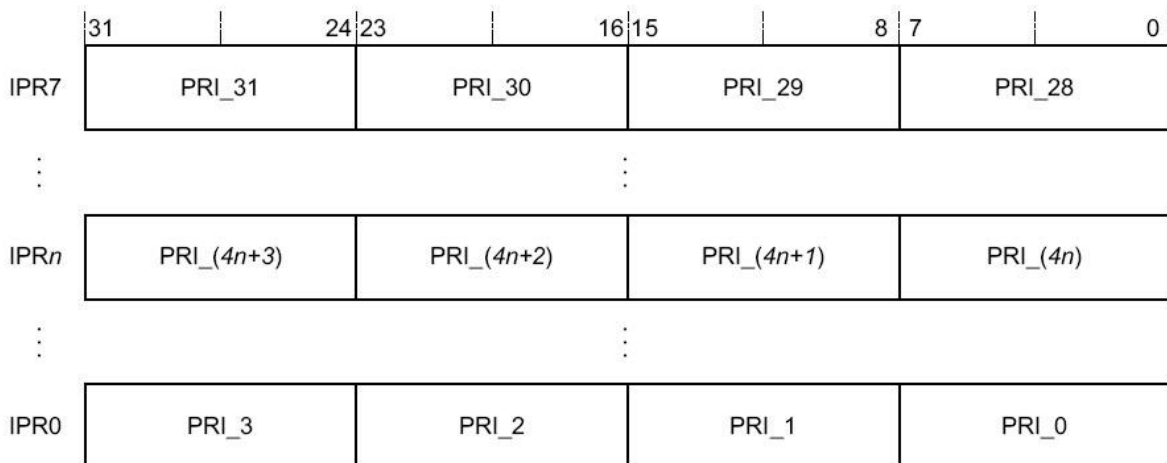


Рисунок 133 – Поля приоритета регистров IPR0-IPR7

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 Cortex-M0 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 (см. подраздел 26.4 «Регистр установки состояния ожидания для прерывания») или соответствующего значения в регистр STIR (см. «Регистр программного формирования прерывания»).

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 395 – 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 396.

Таблица 396 –Обобщенна 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, стр
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 таблицах 13-30.

Назначение разрядов регистра CPUID представлено на рисунке 134:

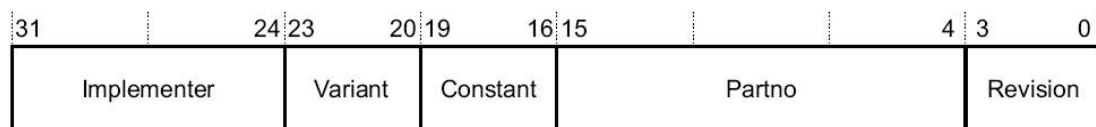


Рисунок 134 – Назначение разрядов регистраCPUID

Implementer – код разработчика 0x41 = ARM.

Variant – значение r в номере версии rnpn изделия: 0x0 = r0p0;

Constant – постоянное значение 0xC;

PartNo – номер модели процессора: 0xC20 = Cortex-M0;

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.

Назначение разрядов регистра ICSR представлено на рисунке 135:

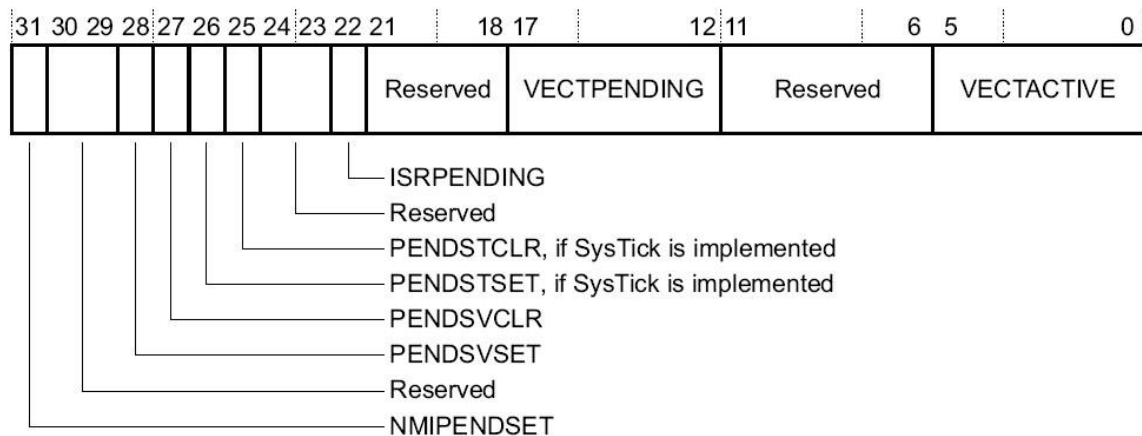


Рисунок 135 – Назначение разрядов регистра ICSR

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.

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 136:

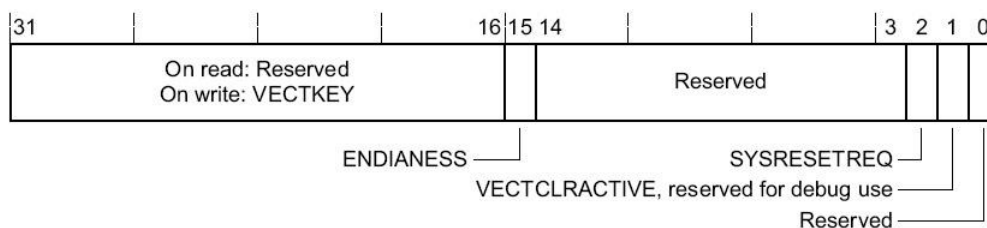


Рисунок 136 – Назначение разрядов регистра AIRCR

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.

Назначение разрядов регистра SCR представлено на рисунке 137:

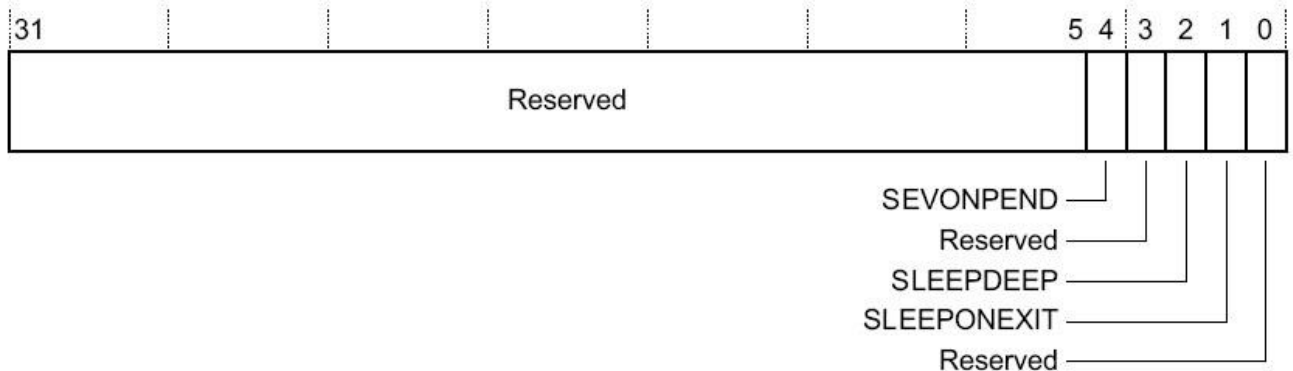


Рисунок 137 – Назначение разрядов регистра SCR

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 (см. «Регистр программного формирования прерывания»).

Назначение разрядов регистра CCR представлено на рисунке 138:

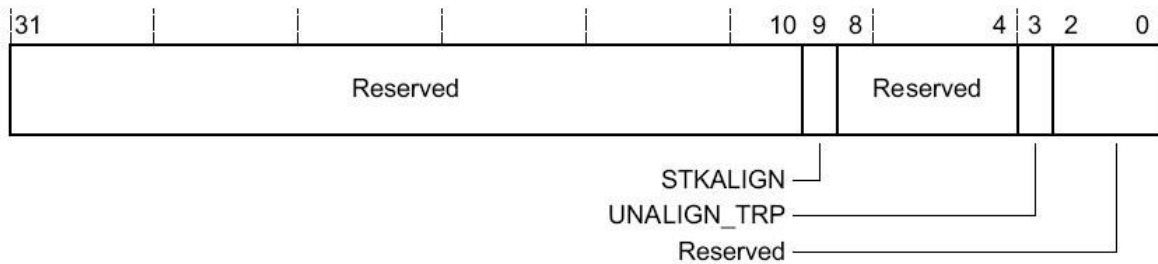


Рисунок 138 – Назначение разрядов регистра CCR

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 HardFault.

27.1.6 Регистры приоритета системных обработчиков

Регистры приоритета системных обработчиков SHPR2-SHPR3 позволяют установить уровень приоритета обработки исключений.

Доступ к регистрам осуществляется пословно.

Поля PRI_N регистров имеют ширину 8 бит, однако в процессоре реализована поддержка доступа только к старшем двум битам [7:6], при чтении данных из младшего бит [5:0] процессор возвращает нули.

Таблица 397 – 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 №2 представлено на рисунке 139:

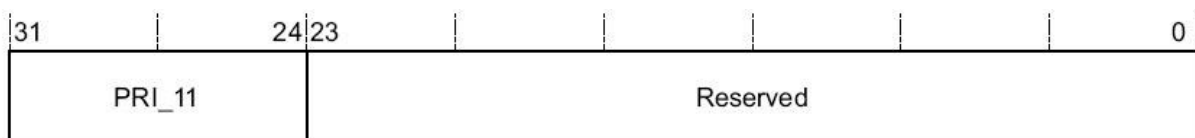


Рисунок 139 – Назначение разрядов регистра №2

PRI_11 Приоритет системного обработчика 11, вызов SVCcall.

27.1.8 Регистр №3 приоритета системных обработчиков

Назначение разрядов регистра №3 представлено на рисунке 140:

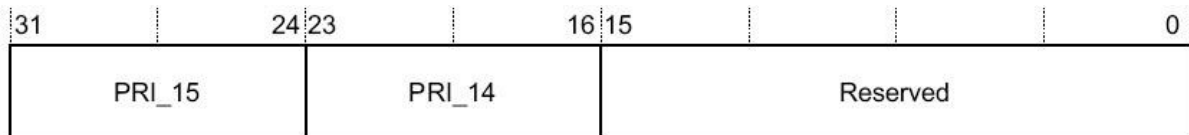


Рисунок 140 – Назначение разрядов регистра №3

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 определённых 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 141.

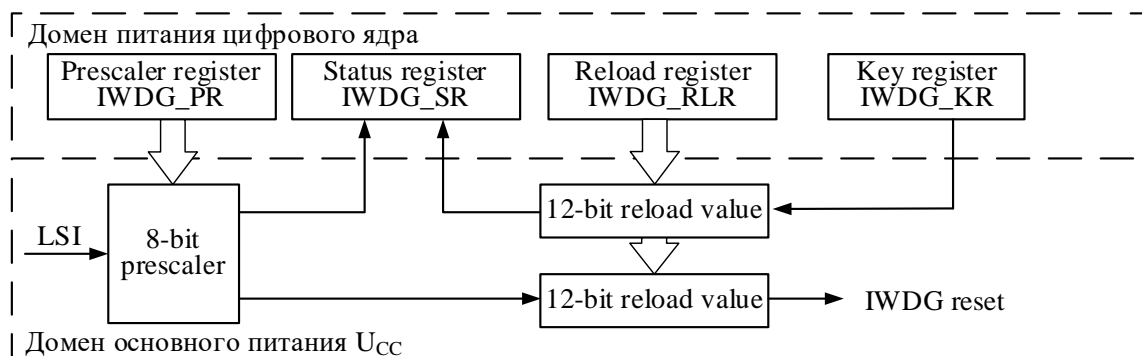


Рисунок 141 – Блок-схема независимого сторожевого таймера

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 предделителя частоты LSI.

Обновление значений предделителя IWDG в домене основного питания осуществляется записью регистра IWDG_PR соответственно. Обновление основания счета (значение, от которого таймер при старте начинает обратный отсчёт) 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 0х5555 для разрешения доступа по записи в них. При

записи в регистр IWDG_PR значения предделителя и ключа 0хАААА 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 U_{CC}.

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 0х5555 в регистр IWDG_KR;
- 3 Запись значения предделителя в регистр IWDG_PR;
- 4 Разрешение работы IWDG записью значения 0хСССС 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 0х5555 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 0хАААА в регистр IWDG_KR.

28.1.1 Описание регистров блока сторожевого таймера IWDG

Таблица 398 – Описание регистров блока сторожевого таймера IWDG

Базовый адрес	Название	Описание
0х4005_0000	IWDG	Сторожевой таймер IWDG
Смещение		
0х00	IWDG_KR[15:0]	Регистр ключа
0х04	IWDG_PR[2:0]	Делитель частоты сторожевого таймера
0х08	IWDG_RLR[11:0]	Регистр основания счета сторожевого таймера
0х0С	IWDG_SR[1:0]	Регистр статуса сторожевого таймера

28.1.1.1 IWDG_KR

Таблица 399 – Регистр IWDG_KR

Номер	31...16	15...0
Доступ	U	W
Сброс	0	0
	-	KEY[15:0]

Таблица 400 – 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"> – 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. – 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. – 0xCCCC – разрешение работы сторожевого таймера (если работа сторожевого таймера не разрешена; работа таймера останавливается аппаратно сбросом от IWDG)

28.1.1.2 IWDG_PR

Таблица 401 – Регистр IWDG_PR

Номер	31...3	2...0
Доступ	U	R/W
Сброс	0	0
	-	PR[2:0]

Таблица 402 – 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"> 000 – делитель на 4: LSI/4; 001 – делитель на 8: LSI/8; 010 – делитель на 16: LSI/16; 011 – делитель на 32: LSI/32; 100 – делитель на 64: LSI/64; 101 – делитель на 128: LSI/128; 110 – делитель на 256: LSI/256; 111 – делитель на 256: LSI/256.

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 141).</p> <p>Запись в регистр IWDG_PR недоступна при фактической записи предделителя в часть IWDG в домене основного питания U_{CC} (см. рисунок 141).</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 403 – Регистр IWDG_RLR

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

Таблица 404 – 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.</p> <p>Определяет значение, загружаемое в сторожевой таймер при записи значения 0xAAAA в регистр IWDG_KR.</p> <p>Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR.</p> <p>Запись в регистр IWDG_RLR недоступна при фактической записи основания счета в часть IWDG в домене основного питания U_{CC} (см. рисунок 141).</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 405 – Регистр IWDG_SR

Номер	31...2	1	0
Доступ	U	R	R
Сброс	0	0	0
	-	RVU	PVU

Таблица 406 – 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. Устанавливается аппаратно и служит признаком того, что обновляется значение сторожевого таймера IWDG в домене основного питания U_{CC} из регистра перезагрузки IWDG_RLR (см. рисунок 141).</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_{CC}. 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_{CC} 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. Устанавливается аппаратно и служит признаком того, что обновляется значение делителя частоты IWDG в домене основного питания U_{CC} из регистра предделителя IWDG_PR (см. рисунок 141).</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_{CC}. 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_{CC} 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 407 – 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 408 – 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 409 – 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 410 – 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 411 – 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	WGTB[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

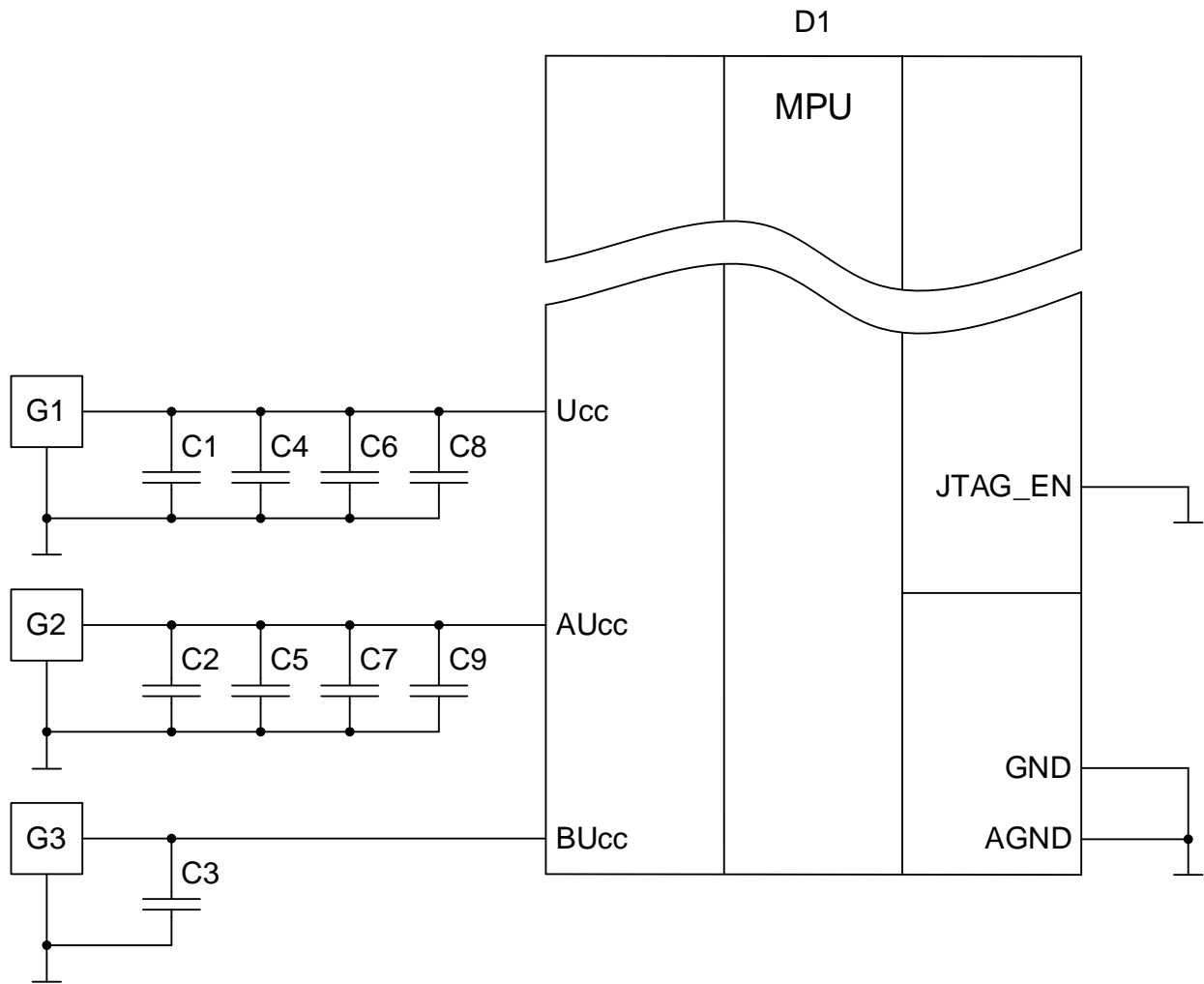
Таблица 412 – Регистр WWDG_SR

Номер	31...1	0
Доступ	U	R/C
Сброс		0
	-	EWIF

Таблица 413 – 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_{cc};

G2 – источник напряжения питания U_{CCA};

G3 – источник напряжения питания U_{CCB}

Рисунок 142 – Типовая схема подключения питания

30 Предельно-допустимые характеристики микросхемы

Таблица 414 – Предельно-допустимые и предельные режимы эксплуатации микросхем

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение источника питания, В	U _{CC}	3,0	3,6	–	4,0
Напряжение источника питания при использовании АЦП, В	U _{CCA}	3,0	3,6		4,0
Напряжение источника питания батареиногo домена, В	U _{CCV}	1,8	3,6	–	4,0
Входное напряжение высокого уровня, В, на выводах: PA(5), PB(4-9), PC (1-6) на выводе OSC_IN при HSE BYPASS=1	U _{IH}	2,0	U _{CC}	–	U _{CC} +0,3
на выводах: PA(0-4, 6-15), PB(0-3,10-14), PC(0, 7), 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 _{IL}	0	0,8	–0,3	–
Выходной ток высокого уровня, мА, при работе в цифровом режиме на выводах: PA, PB, PC	I _{OH}	–6	–	–10	–
Выходной ток низкого уровня, мА на выводах: PA, PB, PC	I _{OL}	–	6	–	10
Частота следования импульсов тактовых сигналов, МГц	f _C	–	36	–	–
Частота следования импульсов тактовых сигналов HSE, МГц при BYPASS=0	f _{C_HSE}	2	16	–	–
при BYPASS=1		–	36		
Частота следования импульсов тактовых сигналов LSE, кГц при BYPASS=0	f _{C_LSE}	32	33	–	–
при BYPASS=1		–	1 000		
Частота следования импульсов тактовых сигналов PLL, МГц	f _{C_PLL}	2	16	–	–
Емкость нагрузки, пФ, на выводах: PA, PB, PC	C _L	–	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 _{GS}	25	–	–	–
при T=85 °C		10	–	–	–
Параметры АЦП					
Напряжение нижней границы внешней опоры АЦП, В	U _{ADC1_REF-}	0	U _{CCA-2,4}	–	4,0
Напряжение верхней границы внешней опоры АЦП, В	U _{ADC0_REF+}	2,4	U _{CCA}	–	4,0
Диапазон напряжения внешнего опорного источника АЦП, В U _{REF(ADC)} = U _{ADC0_REF+} – U _{ADC1_REF-}	U _{REF(ADC)}	2,4	U _{CCA}	–	–
Диапазон напряжения на входе АЦП, В U _{ADC1_REF-} = AGND, U _{ADC0_REF+} = AU _{CC}	U _{AIN}	U _{ADC1_REF-}	U _{ADC0_REF+}	–0,3	4,0
Частота следования импульсов тактовых сигналов АЦП, МГц	f _{C_ADC_S}	–	14	–	–
Параметры ΔΣАЦП					
Частота следования импульсов тактовых сигналов ΔΣАЦП, МГц	f _{C_ADC_D}	–	8,196	–	–
Амплитуда входного дифференциального сигнала ΔΣАЦП, В	A _{NADC_D}	–	1	–	–
Напряжение на входе ΔΣАЦП, В	U _{IADC_D}	–0,5	0,5	–0,8	U _{CC} +0,3
Примечание – Не допускается одновременное задание двух предельных режимов					

31 Электрические параметры микросхемы

Таблица 415 – 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 _{OH}	2,4	–	25, 85, – 40
Выходное напряжение низкого уровня, В, на выводах PA, PB, PC	U _{OL}	–	0,4	
Напряжение срабатывания схемы генерации сброса, В	U _{ВОР}	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 _{ILH}	– 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 _{ILL}	– 1	1	
		– 40	40	
Статический ток потребления, мкА при выключенном стабилизаторе напряжения при включенном стабилизаторе напряжения	I _{CCS}	–	10	
		–	50	
Динамический ток потребления, мА, при 33 кГц $f_c \leq 36 \text{ МГц}$ при 32 $\leq f_c \leq 33 \text{ кГц}$	I _{occ}	–	20	
		–	0,5	
Выходная частота HSI RC-генератора, МГц	f _{O_HSI}	6	10	
Выходная частота LSI RC-генератора, кГц	f _{O_LSI}	10	50	
Выходная частота PLL, МГц максимальная минимальная	f _{O_PLL}	36	–	
		–	2	
Параметры АЦП последовательного приближения				
Разрядность АЦП	E _{NADC}	12	–	25, 85, – 40
Дифференциальная нелинейность, единица младшего разряда	E _{DLADC}	–1	2	
Интегральная нелинейность, единица младшего разряда	E _{ILADC}	–3	3	
Ошибка смещения, единица младшего разряда	E _{OFFADC}	–6	6	
Ошибка усиления, %	E _{GAINADC}	–1	1	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Параметры $\Delta\Sigma\text{АЦП}$				
Выходное напряжение VR_1V, В	U _{OBIAS}	0,96	1,01	25, 85, – 40
Соотношение сигнал / шум, дБ усиление 0дБ, A _{NADC_D} = 1В, f _{NADCO} = 4 кГц	SNR _{D_0}	77	–	
Соотношение сигнал / шум, дБ усиление +6дБ, A _{NADC_D} = 0,5В, f _{NADCO} = 4 кГц	SNR _{D_6}	74	–	
Соотношение сигнал / шум, дБ усиление +12дБ, A _{NADC_D} = 0,25В, f _{NADCO} = 4 кГц	SNR _{D_12}	71	–	
Соотношение сигнал / шум, дБ усиление +18дБ, A _{NADC_D} = 0,125В, f _{NADCO} = 4 кГц	SNR _{D_18}	70	–	
Ошибка усиления предусилителя, дБ	GAIN _{ERR}	–	0,25	

32 Справочные данные

Таблица 416 – 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: $U_{CC} = 0\text{ В}$ $V_{U_{CC}} = 3,6\text{ В}$	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: $U_{CC} = 2,2\text{ В}$, PowerTX=00, $C_1 = 50\text{ пФ}$ PowerTX=01, $C_1 = 50\text{ пФ}$ PowerTX=10, $C_1 = 50\text{ пФ}$ PowerTX=11, $C_1 = 50\text{ пФ}$ PowerTX=11, $C_1 = 30\text{ пФ}$	$t_{W(PA-PF)}$	– – – – –	10 100 20 10 5	
Тактовые частоты и генераторы				
Время установления сигнала HSIRDY относительно HSION, мкс, при: $U_{CC} = 2,2\text{ В}$	$t_{SU(HSI)}$	–	1	25, 85, – 40
Время установления сигнала LSIRDY относительно LSION, мс, при: $U_{CC} = 2,2\text{ В}$	$t_{SU(LSI)}$	–	80	
Время установления сигнала HSERDY относительно HSEON, мкс, при: $U_{CC} = 2,2\text{ В}$	$t_{SU(HSE)}$	–	$2048/f_{C_HSE}$	
Время установления сигнала LSERDY относительно LSEON, мкс, при: $U_{CC} = 2,2\text{ В}$	$t_{SU(LSE)}$	–	$4096/f_{C_LSE}$	
Время установления сигнала PLLRDY относительно PLLON, мкс, при: $U_{CC} = 2,2\text{ В}$	$t_{SU(PLL)}$	–	100	
Длительность сигнала сброса, мкс, при: $U_{CC} = 2,2\text{ В}$	$t_{W(RESET)}$	20	–	
Время запуска после сброса по POR, мс	t_{POR}	–	6	
АЦП				
Время выборки заряда АЦП, нс, при: $U_{CC} = 3,6\text{ В}$	t_{A_ADC}	–	$4 \times f_{C_ADC}$	25, 85, – 40
Время преобразования АЦП, нс при: $U_{CC} = 3,6\text{ В}$	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 _{ADC0_VREF+}	–	50	25, 85, – 40
Ток потребления по входу внешней нижней опоры АЦП, мкА при: ADC1_Cfg_M_REF=1 или ADC2_Cfg_M_REF=1	I _{ADC0_VREF-}	-50	–	
Ток потребления по питанию АЦП, мА при: f _{ADC} =14 МГц, U _{CCA} =3,6В	I _{ОCCADC}	–	3	
Минимальная частота преобразования АЦП, кГц	f _{C_ADCMIN}	10	–	
$\Delta\Sigma$ АЦП				
Выходная частота дискретизации, кГц	f _{NADCO}	–	4, 8, 16	25, 85, – 40
Входное сопротивление, кОм	R _{NADC_D}	30	–	

33 Габаритный чертеж микросхемы

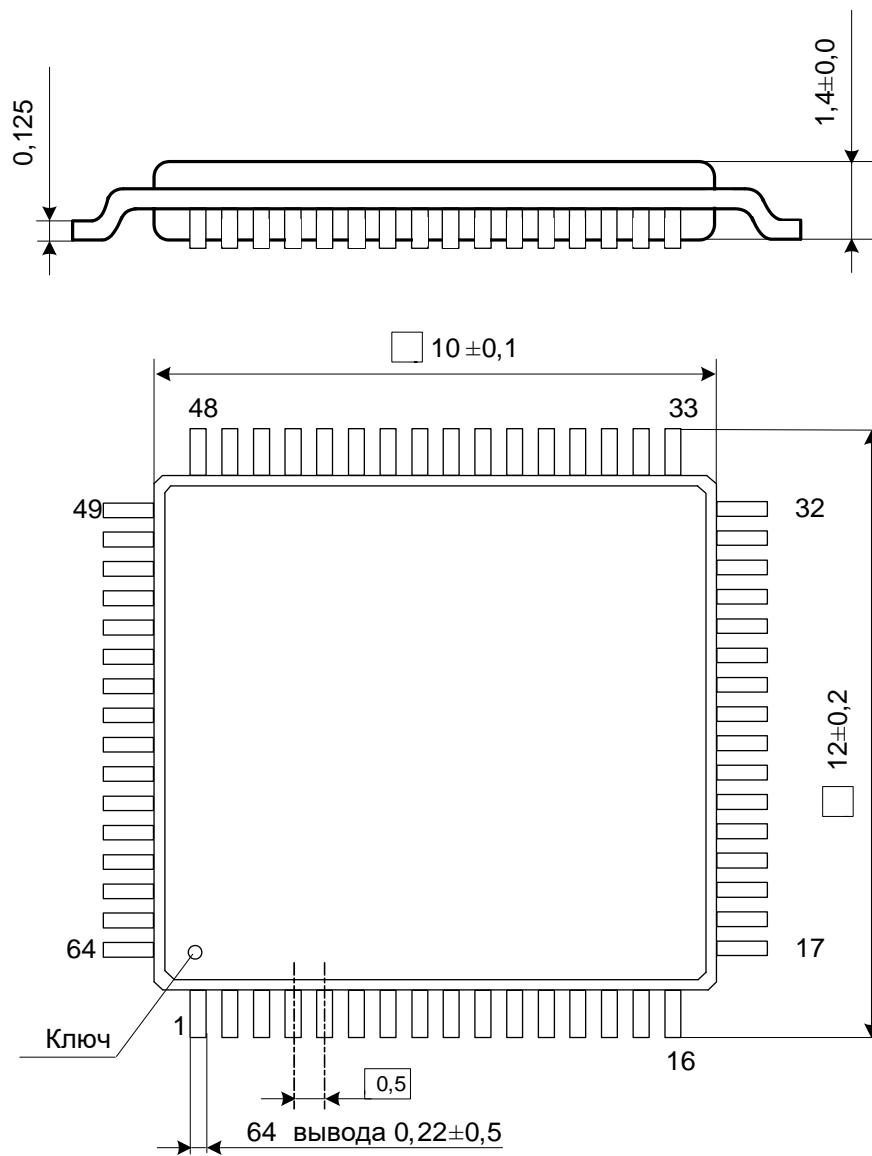


Рисунок 143 – Микросхема в корпусе LQFP64

34 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон, °С
К1986ВК234	MDR32F21QI ARM	LQFP64	от – 40 до 85

Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	24.09.12	1.1.0	Введена впервые	
2	10.06.13	1.2.0	Добавлена структурная схема. Правки по тексту.	5
3	23.07.2013	1.2.1	Правки по тексту	
4	27.12.2013	1.3.1	Исправлена разрядность сигма-дельта АЦП. Перевод надписей на рисунках.	4, По тексту
5	15.01.2014	1.4.1	Исправление заголовков разделов.	24, 25
6	21.01.2014	1.5.1	Внесение исправлений в таблицу 3	16
7	04.03.2014	1.6.1	Исправление в табл.259 (бит 0)	170
8	28.05.2014	1.7.1	Добавлено примечание 2 к таблице 467	380
9	01.07.2014	1.8.0	На рисунке исправлена маркировка микросхемы в корпусе LQFP64, исправлена ориентация корпуса	1
10	12.01.2015	1.9.0	Добавлены типовые схемы включения для учета электроэнергии	152 – 155
11	26.06.2015	1.10.0	Изменено обозначение микросхем. Удален металлокерамический корпус	По тексту
12	01.10.2015	1.11.0	Изменено обозначение микросхем	По тексту
13	13.10.2015	1.12.0	Исправлено функциональное назначение Исправлено значение напряжения источника питания Исправлена таблица электрических параметров	1 По тексту 389, 390
14	02.03.2017	2.0.0	Изменено обозначение микросхем Таблицы параметров приведены в соответствии с ТУ	По тексту 387 – 392
15	24.05.2018	2.1.0	Исправление замечаний	По тексту
16	29.03.2021	2.2.0	Исправление замечаний; Внесены изменения в рисунки: 3, 20, 32, 34; Внесены изменения в таблицы: 1, 3, 13, 14, 137, 213, 233 - 236, 248, 271, 280, 317, 323, 354, 429 Внесение изменений в разделы: 12.2, 14.2.4, 16, 20.1, 22.3, 23.6, 24.6.5,	По тексту 10, 73, 137, 144 6, 15, 27, 101, 130, 140-143, 149, 155, 177, 221, 226, 266, 360 22, 44, 77, 150, 162, 192, 204
17	05.05.2021	2.2.1	Добавлено международное обозначение	375
18	05.10.2021	2.3.0	Исправление опечаток Замена термина «пластиковый» на «пластмассовый» Скорректированы рисунки 45, 46, а также описание режима «Внутренний тактовый сигнал	По тексту 1 181

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			(TIM_CLK)» Рисунок 66 – добавлен блок с сигналами SEL_VREF_BUF, TS_BUF_EN Таблица 4 – дополнено описание режимов UART-загрузчик Таблица 47 – скорректировано описание бита VREF_SEL Таблица 192 – скорректированы формулы для режимов недифференциального включения. Скорректировано описание после таблицы Таблицы 211, 212 – исправлено название и описание бита 3 Таблица 291 – скорректировано описание битов NINV и INV Таблица 319 – дополнено описание бита SEL_VREF_BUF Таблица 392 (перенесена из подраздела 25.8) – добавлено примечание Подразделы 20.1, 20.2, 20.4, 20.5 – описание скорректировано с учетом информации о бите SEL_VREF_BUF Добавлен раздел «Логика работы прерываний контроллера NVIC» Подраздел 26.7 – дополнена информация о прерываниях по уровню сигнала	209 34 93 132 145 201 219 341 210 – 212 349 355
19	01.02.2022	2.4.0	Исправление опечаток Таблица 1 – добавлено описание вывода JTAG_EN Таблицы 6 – 11 – описание уточнено	По тексту 18 36 - 38
20	05.09.2023	2.5.0	Исправление опечаток Исключена информация о «PrimeCell» Информация о тестовых регистрах контроллера DMA исключена Таблица 1 обновлена Таблица 2 – Назначение выводов по блокам добавлена Раздел 4 обновлен Подраздел 10.2 – название управляющего бита исправлено Пункты 10.2.1-10.2.4 скорректированы Таблица 23 исправлена Рисунок 21 обновлен Таблицы 45 – описания регистров скорректированы	По тексту По тексту По тексту 18 20 18 41 42-44 47 89 90

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			Пункт 13.1.1 – биты 29, 26...22 зарезервированы Описания бит регистров ADCUI_FxCTR, ADCUI_FxWC, ADCUI_FxVC, ADCUI_FxAC, ADCUI_FxMD0 скорректированы Раздел 13.2 – добавлена информация о буфере FIFO Таблица 196 – примечание добавлено Пункт 14.1.5 – полином исправлен Рисунки 32, 33 скорректированы Таблицы 209, 210 – разряд 0 зарезервирован Таблицы 222, 231, 237, 243, 245, 247 описания битов скорректированы Таблица 248 дополнена Подраздел 16.1 скорректирован Раздел 19 скорректирован Раздел 20 скорректированы требования для существования преобразования Подразделы 20.4, 20.5 дополнены Подраздел 20.7 «Описание регистров блока контроллера АЦП» добавлен Пункт 20.7.1 – разряды 24...21 зарезервированы, рисунок удален Таблица 322 – описание бит DSS исправлено Раздел 22 скорректирован Пункт 22.11.3 – примечание скорректировано Таблица 370 – название разрядов 31, 30 исправлено	93 96-122 130 135 141 143, 153 145, 145 149, 155-161 163 153 174-207 209 211, 212 214 214 243 250 268 316
21	18.06.2024	2.6.0	Таблица 47 – примечание скорректировано Раздел 14 → раздел 13.2 Таблица 226 – примечание добавлено Пункт 19.3.1 дополнена информация о тактовой частоте TIM_CLK Пункт 21.7.4 скорректирован Раздел 28 обновлен Исправление опечаток	93 130 150 176 242 363-369 По тексту
22	14.02.2024	2.6.1	Заменено "EERPOM" на "EEPROM", «IWDТ» на «IWDG», «WWDТ» на «WWDG» по всему тексту СП; Раздел 6 – дополнена информация о регуляторе LDO; Таблица 3, Таблица 224 – заменено "SPI" на "SSP"; Раздел 10.2.4 – дополнено описание чтения слова;	27 33, 150 44

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			Раздел 15 дополнен;	142