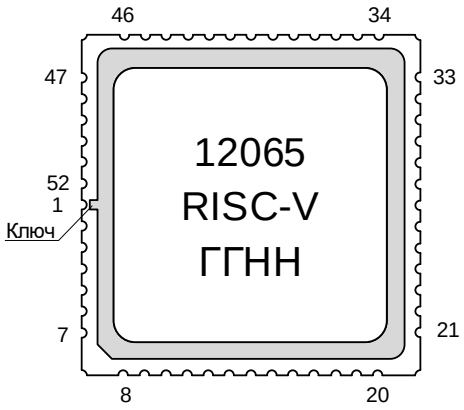
Микросхема многофункционального микроконтроллера MDR12065



ГГ – год выпуска
НН – неделя выпуска

Основные характеристики микросхемы:

- Напряжение источника питания:
  - от 3,0 до 3,6 В (с AFE);
  - от 1,8 до 3,6 В (без AFE);
- 32-разрядная RISC-V архитектура BM-310S0 с системой команд RV32IMCNZX;
- Встроенная Flash-память 512 Кбайт;
- Встроенная память RAM 112 Кбайт;
- Масса микросхем не более 1,0 г;
- Рабочий диапазон температур от минус 60 до плюс 125 °С.

Тип корпуса:

- 52-выводной металлокерамический корпус 5152.52-3 К.

Общее описание и область применения микросхемы

Микросхемы интегральные MDR12065 (далее – микросхемы) представляют собой многофункциональные микроконтроллеры со встроенной Flash-памятью программ и построены на базе процессорного RISC-V ядра BM-310S0. Предназначены для использования в различной радиоэлектронной аппаратуре.

Важно: Спецификация действительна совместно с документом MDR12065 Errata Notice.

Рекомендуется **использовать актуальную версию загрузочной программы**: https://ic.milandr.ru/soft/ "Загрузочная программа для микроконтроллеров серии MDR1206".

Основные характеристики:

Ядро:

– 32-битное RISC-V ядро ВМ-310S0 с системой команд RV32IMCNZX;
– тактовая частота до 50 МГц;
– производительность до 3,70 CoreMark/МГц из интегрированной памяти TCM;
– производительность до 2,15 CoreMark/МГц из Flash-памяти;
– умножение за два цикла.
– команды для ускорения вычислений криптоалгоритмов «Кузнечик», «Стрибог», «Магма», SHA2, AES и т.д.;
– блок физической защиты памяти.

Память:

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

Питание и тактовая частота:

– внешнее питание:
  – от 3,0 до 3,6 В (с AFE);
  – от 1,8 до 3,6 В (без AFE);
– встроенный регулятор напряжения для питания ядра;
– встроенные схемы контроля питания;
– встроенный домен с батарейным питанием;
– встроенный подстраиваемый RC-генератор 8 МГц;
– встроенный подстраиваемый RC-генератор 32 кГц;
– внешний осциллятор от 8 до 16 МГц;
– встроенный умножитель тактовой частоты PLL для ядра.

Режим пониженного энергопотребления:

– батарейный домен с ОЗУ 512 байт.

Аналоговые модули:

– 24-разрядный ΔΣ АЦП (четыре независимых каналов с ПКУ);
– 10-разрядный АЦП (один внешний канал и канал внутреннего термодатчика).

Периферия:

– контроллер прямого доступа в память с функциями передачи Периферия-Память, Память-Память;
– контроллеры интерфейсов 4 – UART, 2 – SSP, 1 – I2C, 1 – ISO7816;
– генератор случайных чисел;
– защитная сетка, датчики напряжения питания;
– до 33-ти пользовательских линий ввода/вывода;
– четыре блока 32-разрядных таймеров с четырьмя каналами захвата событий и ШИМ;
– два сторожевых таймера;
– блок подсчета CRC с изменяемым полиномом.

Режим отладки:

– последовательный отладочный интерфейс JTAG.

Содержание

1 Введение 13
2 Структурная блок-схема 14
3 Условное графическое изображение....15
4 Диаграмма расположения выводов в корпусе 16
5 Описание выводов 17
6 Указания по применению и эксплуатации 20
7 Система питания 21
7.1 Структурные схемы подачи питания 22
7.2 Схема сброса при включении и выключении основного питания....23
8 Организация памяти 25
8.1 Базовые адреса 26
9 Режим запуска 27
9.1 Режим UART+J 28
9.1.1 Параметры связи по UART 28
9.1.2 Протокол обмена по UART....29
9.1.3 Синхронизация с внешним устройством....29
9.2 Команды UART-загрузчика 29
9.2.1 Команда CMD_SYNC 29
9.2.2 Команда CMD_CR 30
9.2.3 Команда CMD_BAUD 30
9.2.4 Команда CMD_LOAD....31
9.2.5 Команда CMD_VFY....31
9.2.6 Команда CMD_RUN 32
9.2.7 Прием параметров команды....32
9.2.8 Сообщение об ошибке 32
10 Контроллер Flash-памяти программ 33
10.1 Работа Flash-памяти программ в обычном режиме....33
10.2 Работа Flash-памяти программ в режиме программирования....34
10.2.1 Стирание всей памяти или 2×256 Кбайт основной памяти 34
10.2.2 Стирание страницы памяти размером 4 Кбайт....35
10.2.3 Запись 32-битного слова в память....36
10.2.4 Чтение 32-битного слова из памяти 37
10.3 Производственная информация, калибровочные значения и защита Flash-памяти 38
10.4 Использование калибровочных значений 41
10.5 Регистры управления контроллера Flash-памяти программ....41
10.5.1 FLASH_CMD....42
10.5.2 FLASH_ADR....43
10.5.3 FLASH_DI....44
10.5.4 FLASH_DO 44
10.5.5 FLASH_KEY....44
10.5.6 CHIP_ID_CTRL 45
10.5.7 OTP_KEY....45
11 Система команд....46
11.1 Набор команд расширений для ускорения криптографии....49
11.2 Поддержка Wait For Interrupt....51

12 Процессорное ядро BM-310S0 52
12.1 Структурная схема процессора 53
12.1.1 Интерфейс TCM 54
12.1.2 Арбитр доступа к блокам памяти TCM 54
12.1.3 Интерфейс AHB I/O 54
12.1.4 Запросы прерываний 54
12.1.5 Интерфейс JTAG 54
12.1.6 LIO crossbar 55
12.1.7 Защита физической памяти (PMP) 55
13 Блок ΔΣ АЦП 60
13.1 Описание регистров управления блока четырех каналов АЦП 62
13.1.1 ADCUI_CTRL1 63
13.1.2 ADCUI_CTRL2 65
13.1.3 ADCUI_F0CTR 65
13.1.4 ADCUI_F0AC 67
13.1.5 ADCUI_F0I0DAT 67
13.1.6 ADCUI_F0I3DAT 67
13.1.7 ADCUI_F0STAT 68
13.1.8 ADCUI_F0MASK 68
13.1.9 ADCUI_F1CTR 69
13.1.10 ADCUI_F1IDAT 70
13.1.11 ADCUI_F1STAT 70
13.1.12 ADCUI_F1MASK 71
13.1.13 ADCUI_F2CTR 71
13.1.14 ADCUI_F2IDAT 72
13.1.15 ADCUI_F2STAT 72
13.1.16 ADCUI_F2MASK 73
13.1.17 ADCUI_CCAL1 73
13.1.18 ADCUI_CCAL2 74
13.1.19 ADCUI_CCAL3 74
13.1.20 ADCUI_CCAL4 74
13.1.21 ADCUI_MSC 75
13.1.22 ADCUI_FILTERCFG 75
13.1.23 ADCUI_CLKPHASE1 76
13.1.24 ADCUI_CLKPHASE2 77
14 Соответствие отчетов АЦП внешним сигналам 78
15 Аппаратный блок вычисления CRC 81
15.1 Описание регистров управления блока CRC 81
15.1.1 CRC_CTRL 81
15.1.2 CRC_STAT 82
15.1.3 CRC_DATAI 83
15.1.4 CRC_VAL 83
15.1.5 CRC_POL 83
16 Сигналы тактовой частоты 84
16.1 Встроенный RC-Генератор HSI 85
16.2 Встроенный RC-генератор LSI 85
16.3 Внешний осциллятор HSE 85
16.4 Встроенный блок умножения системной тактовой частоты 87

16.5 Описание регистров блока контроллера тактовой частоты 88
16.5.1 CLOCK_STATUS 88
16.5.2 PLL_CONTROL 89
16.5.3 HS_CONTROL 89
16.5.4 CPU_CLOCK 90
16.5.5 PER1_CLOCK 90
16.5.6 ADC_CLOCK 91
16.5.7 RTC_CLOCK 93
16.5.8 PER2_CLOCK 94
16.5.9 DMA_DONE_STICK 95
16.5.10 TIM_CLOCK 95
16.5.11 UART_CLOCK 96
16.5.12 SSP_CLOCK 97
16.5.13 DIV_SYS_TIM 98
17 Батарейный домен и часы реального времени 99
17.1 Часы реального времени 99
17.2 Память регистров и криптографический ключей 101
17.3 Описание регистров блока батарейного домена 101
17.3.1 BKP_MEM (128x32) 102
17.3.2 BKP_WPR 102
17.3.3 BKP_LDO 103
17.3.4 BKP_CLK 104
17.3.5 BKP_RTC_CR 105
17.3.6 BKP_RTC_WUTR 107
17.3.7 BKP_RTC_PREDIV_S 107
17.3.8 BKP_RTC_PRL 107
17.3.9 BKP_RTC_ALRM 108
17.3.10 BKP_RTC_CS 108
17.3.11 BKP_RTC_TR 110
17.3.12 BKP_RTC_DR 110
17.3.13 BKP_RTC_ALRMAR, BKP_RTC_ALRMBR 111
18 Порты ввода-вывода 113
18.1 Описание регистров портов ввода-вывода 114
18.1.1 PORT_RXTX 115
18.1.2 PORT_OE 115
18.1.3 PORT_FUNC 116
18.1.4 PORT_ANALOG 116
18.1.5 PORT_PULL 117
18.1.6 PORT_PWR 117
18.1.7 PORT_SETTX 118
18.1.8 PORT_CLRTX 118
18.1.9 PORT_RDTX 119
19 Детектор напряжения питания 120
19.1 Описание регистров блока PVD 121
19.1.1 PVDCS 121
20 Таймеры общего назначения 123
20.1 Основные характеристики 123
20.1.1 Структурная схема 124

20.2 Базовый блок таймера 125
20.2.1 Инициализация тактирования таймера 125
20.2.2 Инициализация основного счетчика таймера 125
20.2.3 Режимы счета 126
20.2.4 Тактовая частота F_{DTS} 128
20.3 Источники событий для счета 129
20.3.1 Внутренний тактовый сигнал (TIM_CLKd) 130
20.3.2 Событие в другом таймере (CNT==ARR) 131
20.3.3 Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CHyi 132
20.3.4 Внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR 134
20.4 Режим захвата 135
20.5 Режим ШИМ 137
20.5.1 Генератор опорного сигнала REF 137
20.5.2 Генератор «мертвой зоны» 139
20.5.3 Выходные блоки 140
20.6 Блок цифрового фильтра 141
20.7 Флаги состояний, прерывания и запросы DMA 143
20.7.1 Флаги состояний 143
20.7.2 Прерывания 143
20.7.3 Запросы DMA 143
20.8 Примеры 144
20.8.1 Обычный счетчик 144
20.8.2 Режим захвата 144
20.8.3 Режим ШИМ 145
20.9 Описание регистров блока таймера 147
20.9.1 CNT 148
20.9.2 PSG 148
20.9.3 ARR 148
20.9.4 CNTRL 149
20.9.5 CHy_CCR 150
20.9.6 CHy_CCR1 150
20.9.7 CHy_CNTRL0 151
20.9.8 CHy_CNTRL1 153
20.9.9 CHy_CNTRL2 154
20.9.10 CHy_DTG 154
20.9.11 BRKETR_CNTRL 155
20.9.12 STATUS 156
20.9.13 IE 158
20.9.14 DMA_RE 159
21 Контроллер SAR АЦП 161
21.1 Преобразование внешнего канала 162
21.2 Последовательное преобразование нескольких каналов 163
21.3 Преобразование с контролем границ 163
21.4 Датчик температуры 164
21.4.1 Формула расчета температуры 164
21.5 Время заряда внутренней емкости 164

21.6 Время заряда внутренней емкости АЦП и время преобразования 165
21.7 Описание регистров блока контроллера АЦП 166
21.7.1 ADC1_CFG 167
21.7.2 ADC1_H_LEVEL 169
21.7.3 ADC1_L_LEVEL 169
21.7.4 ADC1_RESULT 169
21.7.5 ADC1_STATUS 170
21.7.6 ADC1_CHSEL 171
21.7.7 ADC_TRIM 171
21.8 Выводы SAR_AINx 172
22 Контроллер интерфейса I2C 173
22.1 Конфигурация контроллера I2C 173
22.2 Конфигурация системы 174
22.3 Протокол I2C 174
22.3.1 Сигнал START 175
22.3.2 Передача адреса 175
22.3.3 Передача данных 176
22.3.4 Сигнал STOP 176
22.4 Описание регистров контроллера I2C 176
22.4.1 PRL 177
22.4.2 PRH 177
22.4.3 CTR 177
22.4.4 RXD 178
22.4.5 STA 178
22.4.6 TXD 179
22.4.7 CMD 180
23 Контроллер SSP 181
23.1 Основные характеристики модуля SSP 181
23.1.1 Программируемые параметры 182
23.1.2 Характеристики интерфейса SPI 182
23.1.3 Характеристики интерфейса Microwire 183
23.1.4 Характеристики интерфейса SSI 183
23.2 Общий обзор модуля SSP 183
23.2.1 Блок формирования тактового сигнала 184
23.2.2 Буфер FIFO передатчика 184
23.2.3 Буфер FIFO приемника 184
23.2.4 Блок приема и передачи данных 185
23.2.5 Блок формирования прерываний 185
23.3 Интерфейс прямого доступа к памяти 185
23.4 Конфигурирование приемопередатчика 186
23.5 Разрешение работы приемопередатчика 186
23.6 Соотношения между тактовыми сигналами 186
23.7 Программирование регистра управления CR0 187
23.8 Программирование регистра управления CR1 188
23.9 Формирование тактового сигнала обмена данными 188
23.10 Формат информационного кадра 188
23.10.1 Формат синхронного обмена SSI фирмы Texas Instruments 189
23.10.2 Формат синхронного обмена SPI фирмы Motorola 190

23.10.3 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0 .191
23.10.4 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1 .192
23.10.5 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0 .193
23.10.6 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1 .194
23.10.7 Формат синхронного обмена Microwire фирмы National Semiconductor 195
23.11 Примеры конфигурации модуля в ведущем и ведомом режимах 198
23.12 Интерфейс прямого доступа к памяти 200
23.13 Прерывания.....201
23.13.1 SSPRXINTR 202
23.13.2 SSPTXINTR 202
23.13.3 SSPRORINTR 202
23.13.4 SSPRTINTR.....203
23.13.5 SSPRNEINTR.....203
23.13.6 SSPTFEINTR.....203
23.13.7 SSPTNBSYINTR.....203
23.13.8 SSPINTR.....203
23.14 Программное управление модулем.....203
23.14.1 Общая информация.....203
23.14.2 Описание регистров контроллера SSP.....204
24 Контроллер UART 212
24.1 Основные характеристики модуля UART 212
24.2 Программируемые параметры.....213
24.3 Отличия от контроллера UART 16C650 213
24.4 Функциональные возможности 214
24.5 Описание функционирования блока UART 215
24.5.1 Генератор тактового сигнала приемопередатчика 216
24.5.2 Буфер FIFO передатчика 216
24.5.3 Буфер FIFO приемника.....216
24.5.4 Блок передатчика 216
24.5.5 Блок приемника.....216
24.5.6 Блок формирования прерываний.....217
24.5.7 Интерфейс прямого доступа к памяти 217
24.5.8 Блок и регистры синхронизации 217
24.6 Описание функционирования ИК кодека IrDA SIR 217
24.6.1 Кодер ИК передатчика.....218
24.6.2 Декодер ИК приемника 218
24.7 Описание работы UART.....219
24.7.1 Сброс модуля.....219
24.7.2 Тактовые сигналы 219
24.7.3 Работа универсального асинхронного приемопередатчика 219
24.7.4 Коэффициент деления частоты.....220
24.7.5 Передача и прием данных 221
24.7.6 Биты ошибки 222
24.7.7 Бит переполнения буфера 222
24.7.8 Запрет буфера FIFO 222
24.7.9 Работа кодека ИК обмена данными IrDA SIR.....223
24.8 Линии управления модемом 225

24.8.1 Аппаратное управление потоком данных 225
24.8.2 Управление потоком данных по линии RTS 226
24.8.3 Управление потоком данных по линии CTS 226
24.9 Интерфейс прямого доступа к памяти 227
24.10 Прерывания 229
24.10.1 UARTMSINTR 230
24.10.2 UARTRXINTR 230
24.10.3 UARTTXINTR 231
24.10.4 UARTRTINTR 231
24.10.5 UARTEINTR 231
24.10.6 UARTINTR 232
24.11 Программное управление модулем 232
24.12 Описание регистров контроллера UART 233
24.12.1 Регистр данных DR 233
24.12.2 Регистр состояния приемника / сброса ошибки RSR_ECR 235
24.12.3 Регистр флагов FR 236
24.12.4 Регистр управления ИК обменом в режиме пониженного энергопотребления ILPR 236
24.12.5 Регистр целой части делителя скорости передачи данных IBRD 237
24.12.6 Регистр дробной части делителя скорости передачи данных FBRD 237
24.12.7 Регистр управления линией LCR_H 239
24.12.8 Регистр управления CR 241
24.12.9 Регистр порога прерывания по заполнению буфера FIFO IFLS 243
24.12.10 Регистр установки сброса маски прерывания IMSC 244
24.12.11 Регистр состояния прерываний RIS 245
24.12.12 Регистр маскированного состояния прерываний MIS 247
24.12.13 Регистр сброса прерываний ICR 248
24.12.14 Регистр управления прямым доступом к памяти DMACR 249
24.12.15 Регистр управления тестированием TCR 250
25 Контроллер прямого доступа в память DMA 251
25.1 Основные свойства контроллера DMA 251
25.2 Термины и определения 251
25.3 Функциональное описание 253
25.3.1 Распределение каналов DMA 253
25.3.2 Блок, подключенный к шине APB 254
25.3.3 Блок, подключенный к шине AHB 254
25.3.4 Управляющий блок DMA 254
25.3.5 Типы передач 255
25.3.6 Разрядность передаваемых данных 255
25.3.7 Управление защитой данных 255
25.3.8 Инкремент адреса 256
25.4 Управление DMA 256
25.4.1 Правила обмена данными 256
25.4.2 Диаграммы работы контроллера DMA 259
25.4.3 Правила арбитража DMA 265
25.4.4 Приоритет 265
25.4.5 Типы циклов DMA 267
25.5 Структура управляющих данных канала 278

25.5.1 Указатель конца данных источника 282
25.5.2 Указатель конца данных приемника 283
25.5.3 Разряды управления 283
25.6 Описание регистров контроллера DMA 289
25.6.1 Статусный регистр DMA STATUS 290
25.6.2 Регистр конфигурации DMA CFG 291
25.6.3 Регистр базового адреса управляющих данных каналов CTRL_BASE_PTR 292
25.6.4 Регистр базового адреса альтернативных управляющих данных каналов ALT_CTRL_BASE_PTR 293
25.6.5 Регистр статуса ожидания запроса на обработку каналов WAITONREQ_STATUS 293
25.6.6 Регистр программного запроса на обработку каналов CHNL_SW_REQUEST 294
25.6.7 Регистр установки пакетного обмена каналов CHNL_USEBURST_SET 294
25.6.8 Регистр сброса пакетного обмена каналов CHNL_USEBURST_CLR 296
25.6.9 Регистр маскирования запросов на обслуживание каналов CHNL_REQ_MASK_SET 296
25.6.10 Регистр очистки маскирования запросов на обслуживание каналов CHNL_REQ_MASK_CLR 297
25.6.11 Регистр установки разрешения каналов CHNL_ENABLE_SET 298
25.6.12 Регистр сброса разрешения каналов CHNL_ENABLE_CLR 299
25.6.13 Регистр установки первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_SET 299
25.6.14 Регистр сброса первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_CLR 301
25.6.15 Регистр установки приоритета каналов CHNL_PRIORITY_SET 302
25.6.16 Регистр сброса приоритета каналов CHNL_PRIORITY_CLR 302
25.6.17 Регистр сброса флага ошибки ERR_CLR 303
26 Прерывания 304
27 Контроллер обработки локальных прерываний CLIC 305
27.1 Вытеснение прерываний 305
27.2 Взаимодействие контроллера CLIC с другими локальными прерываниями 305
27.3 Описание регистров контроллера CLIC 306
27.3.1 Конфигурационный регистр cliccfg 306
27.3.2 Информационный регистр clicinfo 309
27.3.3 Регистр ожидания прерывания clicintip 310
27.3.4 Регистр разрешения прерываний clicintie 311
27.3.5 Регистр атрибутов прерывания clicintattr 312
27.3.6 Регистр управления входом прерывания clicintctl 313
27.4 Контрольно-статусные регистры CSR 314
27.4.1 Регистры mtvec, utvec 315
27.4.2 Регистры mtvt, utvt 316
27.4.3 Регистры mnxti, unxti 316
27.4.4 Регистры mintstatus, uintstatus 317

27.4.5 Регистры mintthresh, uintthresh....318
27.4.6 Регистр mclicbase....319
27.4.7 Регистры mscratchcswl, uscratchcswl....320
27.5 Карта локальных прерываний....321
28 Подсистема отладки....323
28.1 Блок DTM (Debug Transport Module)....323
28.1.1 Регистры DTM....324
28.2 Блок DM (Debug Module)....326
28.2.1 Регистры DM....326
28.3 Подключение адаптера JTAG к микросхеме....328
29 Сторожевые таймеры....330
29.1 Регистры блока сторожевых таймеров....330
29.1.1 Блок сторожевого таймера IWDG....330
29.1.2 Регистры сторожевого таймера WWDG....335
30 Генератор случайных чисел RANDOM....339
30.1 Назначение и свойства....339
30.2 Регистры модуля....339
30.2.1 Регистр статуса и управления STAT_CTRL....340
30.2.2 Регистр управления прерыванием INT_CTRL....340
30.2.3 Регистр делителя частоты генератора CLK_DIV....341
30.2.4 Регистр паузы включения PAUSE....341
30.2.5 Регистр случайного значения OUTPUT....341
30.2.6 Регистр счетчика паузы PAUSE_CNT....342
30.2.7 Регистр сбора случайного числа TEMP....342
31 Контроллер USART (ISO7816)....343
31.1 Введение....343
31.2 Особенности модуля USART....343
31.3 Функциональное описание USART....344
31.3.1 Описание символов USART....346
31.3.2 Передатчик....347
31.3.3 Приемник....350
31.3.4 Дробный генератор скорости....354
31.3.5 Допуск ухода тактовой частоты для приемника USART....357
31.3.6 Многопроцессорный обмен....358
31.3.7 Контроль четности....360
31.3.8 Синхронный режим USART....361
31.3.9 Однопроводной полудуплексный обмен данными....363
31.3.10 Режим Smartcard....364
31.3.11 Аппаратное управление потоком....366
31.4 Прерывания USART....368
31.5 Регистры USART....369
31.5.1 USART_SR....369
31.5.2 USART_DR....372
31.5.3 USART_BRR....373
31.5.4 USART_CR1....374
31.5.5 USART_CR2....377
31.5.6 USART_CR3....378
31.5.7 USART_GTPR....379

32 Электрические параметры микросхем 381
33 Предельно-допустимые и предельные параметры 382
34 Справочные параметры 384
34.1 Динамические характеристики ΔΣ АЦП 386
34.2 Производительность в тестах 386
34.3 Характеристики выводов микросхемы 387
35 Типовые зависимости 388
36 Габаритный чертеж микросхемы 389
37 Информация для заказа 390

1 Введение

Микросхема относится к серии микроконтроллеров со встроенной Flash-памятью программ и построена на базе процессорного RISC-V ядра BM-310S0. Микросхема работает на тактовой частоте до 50 МГц и содержит 512+16 Кбайт Flash-памяти и 112 Кбайт ОЗУ. Периферия микросхемы включает в себя четыре канала 24-битных независимых ΔΣ АЦП. Каждый канал ΔΣ АЦП имеет предусилитель и фазовую подстройку Каждый канал ΔΣ АЦП может быть включен или отключен независимо от других каналов и имеет отдельный канал прямого доступа в память. Еще один дополнительный 10-битный АЦП последовательного приближения может быть использован для мониторинга напряжения питания основного или батарейного доменов, а также для измерения температуры или захвата внешнего сигнала. В состав микросхемы входит четыре интерфейса UART, два SSP и один интерфейс I2C. Криптографическая часть микросхемы включает системы команд xgost, генератор случайных чисел и защитную сетку. Микросхема содержит четыре 32-разрядных таймера с четырьмя каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Также микросхема содержит системный 64-разрядный таймер и два сторожевых таймера. Один из сторожевых таймеров работает от напряжения ядра, второй – от основного питания.

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

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 LDO для формирования питания цифровой части не требует дополнительных внешних элементов. Для работы микросхемы достаточно одного внешнего напряжения питания в диапазоне от 1,8 до 3,6 В.Также в микросхеме реализован батарейный домен, работающий от внешней батареи при отсутствии основного питания. В батарейном домене могут быть сохранены криптографические ключи, а также работают часы реального времени. Встроенные детекторы напряжения питания могут отслеживать уровень внешнего основного питания и уровень напряжения питания на батарее. Аппаратные схемы сброса по просадке питания позволяют минимизировать вероятность сбойной работы микросхемы при выходе уровня напряжения питания за допустимые пределы. В случае возникновения сбоя встроенные сторожевые таймеры позволят восстановить работу микросхемы.

2 Структурная блок-схема

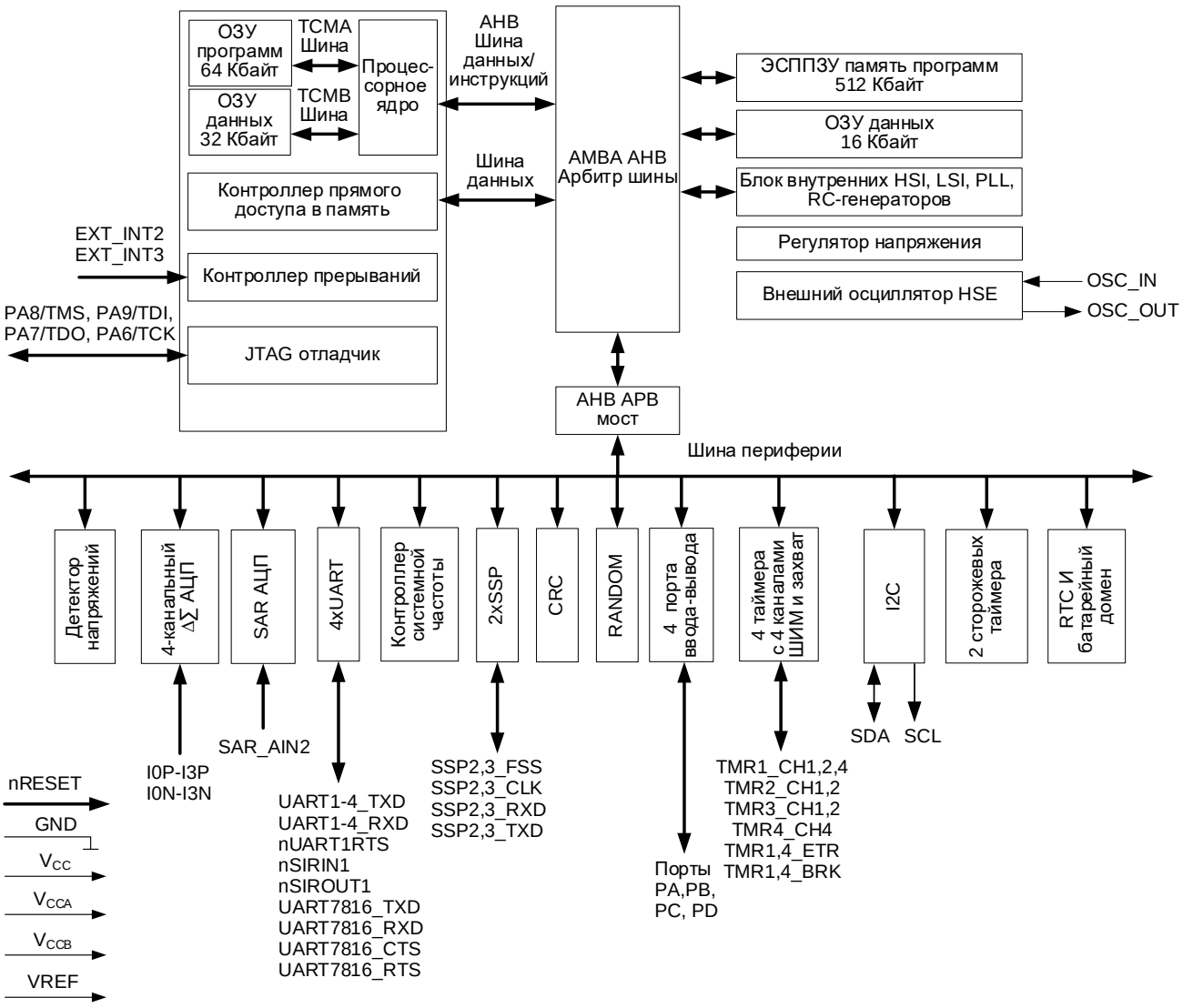


Рисунок 1 – Структурная блок-схема микросхем

3 Условное графическое изображение

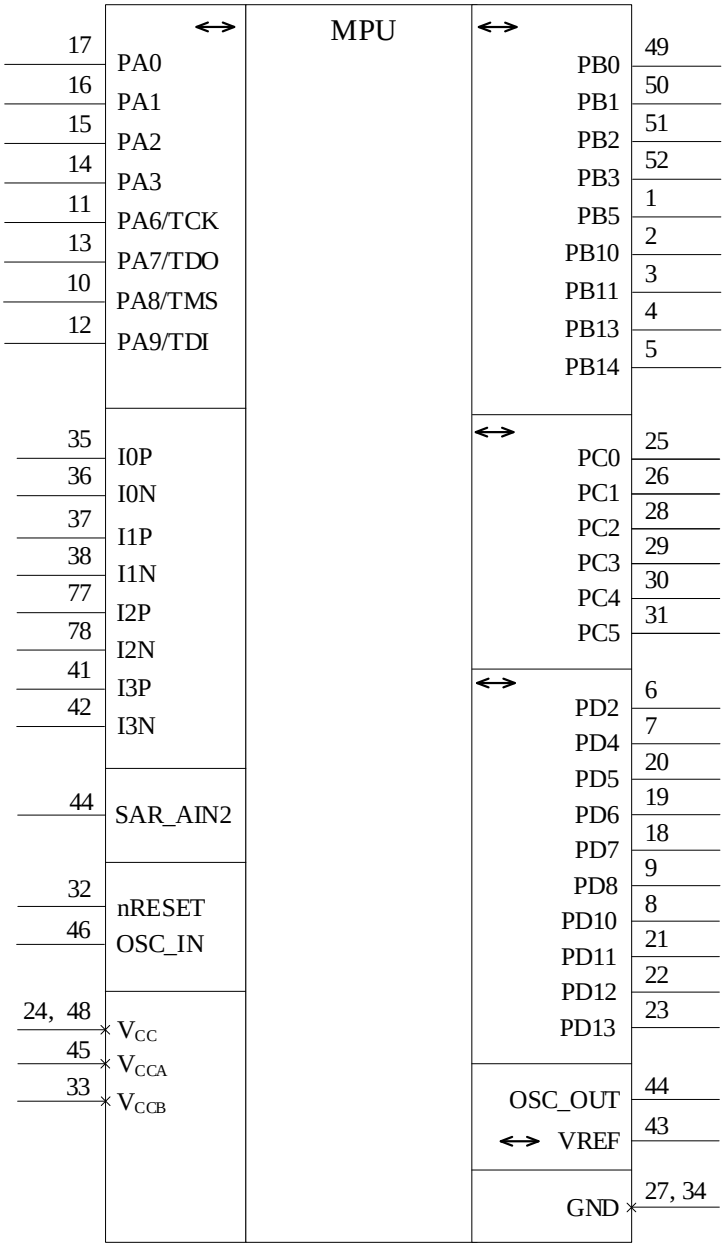


Рисунок 2 – Условное графическое изображение микросхем

4 Диаграмма расположения выводов в корпусе

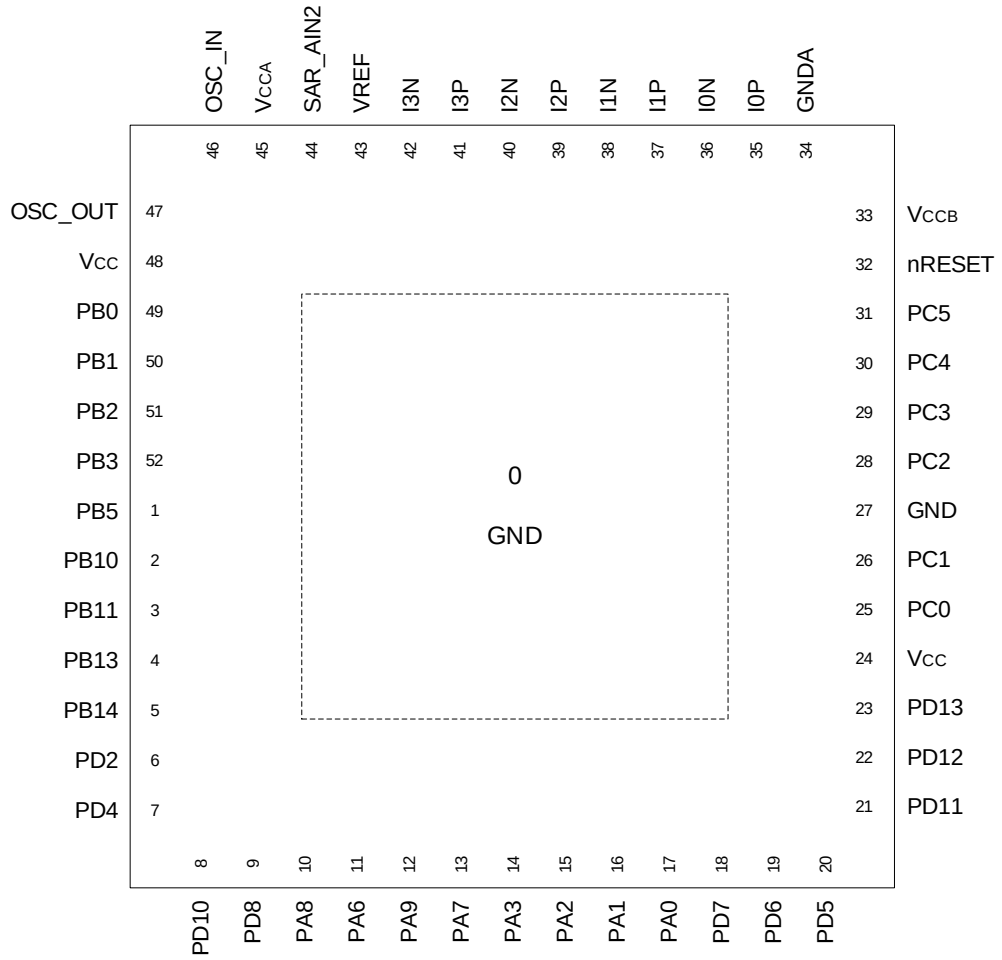


Рисунок 3 – Диаграмма расположения выводов микросхем

5 Описание выводов

Таблица 1 – Описание выводов

Номер вывода	Обозначение вывода	Функциональное назначение вывода
1	PB5	Порт ввода-вывода B разряд 5
2	PB10	Порт ввода-вывода B разряд 10
3	PB11	Порт ввода-вывода B разряд 11
4	PB13	Порт ввода-вывода B разряд 13
5	PB14	Порт ввода-вывода B разряд 14
6	PD2	Порт ввода-вывода D разряд 2
7	PD4	Порт ввода-вывода D разряд 4
8	PD10	Порт ввода-вывода D разряд 10
9	PD8	Порт ввода-вывода D разряд 8
10	PA8/TMS	Порт ввода-вывода A разряд 8 / Вход управления JTAG
11	PA6/TCK	Порт ввода-вывода A разряд 6 / Вход синхросигнала JTAG
12	PA9/TDI	Порт ввода-вывода A разряд 9 / Вход данных JTAG
13	PA7/TDO	Порт ввода-вывода A разряд 7 / Выход данных JTAG
14	PA3	Порт ввода-вывода A разряд 3
15	PA2	Порт ввода-вывода A разряд 2
16	PA1	Порт ввода-вывода A разряд 1
17	PA0	Порт ввода-вывода A разряд 0
18	PD7	Порт ввода-вывода D разряд 7
19	PD6	Порт ввода-вывода D разряд 6
20	PD5	Порт ввода-вывода D разряд 5
21	PD11	Порт ввода-вывода D разряд 11
22	PD12	Порт ввода-вывода D разряд 12
23	PD13	Порт ввода-вывода D разряд 13
24	V_{CC}	Питание
25	PC0	Порт ввода-вывода C разряд 0
26	PC1	Порт ввода-вывода C разряд 1
27	GND	Общий
28	PC2	Порт ввода-вывода C разряд 2
29	PC3	Порт ввода-вывода C разряд 3
30	PC4	Порт ввода-вывода C разряд 4
31	PC5	Порт ввода-вывода C разряд 5
32	nRESET	Сигнал внешнего сброса
33	V_{CCB}	Питание батарейного домена
34	GND	Общий
35	I0P	Вход канала $\Delta\sum$ АЦП I0 прямой
36	I0N	Вход канала $\Delta\sum$ АЦП I0 инверсный

Номер вывода	Обозначение вывода	Функциональное назначение вывода
37	I1P	Вход канала ΔΣ АЦП I1 прямой
38	I1N	Вход канала ΔΣ АЦП I1 инверсный
39	I2P	Вход канала ΔΣ АЦП I2 прямой
40	I2N	Вход канала ΔΣ АЦП I2 инверсный
41	I3P	Вход канала ΔΣ АЦП I3 прямой
42	I3N	Вход канала ΔΣ АЦП I3 инверсный
43	VREF	Вход опорного напряжения 2,4 В / Выход внутреннего источника опорного напряжения
44	SAR_AIN2	Вход 2 SAR АЦП
45	V_{CCA}	Аналоговое питание ΔΣ АЦП и SAR АЦП
46	OSC_IN	Вход генератора HSE
47	OSC_OUT	Выход генератора HSE
48	Vcc	Питание
49	PB0	Порт ввода-вывода B разряд 0
50	PB1	Порт ввода-вывода B разряд 1
51	PB2	Порт ввода-вывода B разряд 2
52	PB3	Порт ввода-вывода B разряд 3
0 (металлизация обратной стороны корпуса)	GND	Общий
Примечание – Функции выводов портов A, B, C, D приведены в таблице 2		

Таблица 2 – Функции выводов портов A, B, C, D

Вывод	Аналоговая функция MODE=00 ANALOG_EN=0 OE=0		Цифровая функция						
			Порт IO MODE=00 ANALOG_EN=0/1 OE=0/1	Основная MODE=01 ANALOG_EN=0/1 OE=X		Альтернативная MODE=10 ANALOG_EN=0/1 OE=X		Переопределённая MODE=11 ANALOG_EN=0/1 OE=X	
Порт A									
PA0	-		PA0	TMR1_CH1	1	SSP2_FSS	9	-	
PA1	-		PA1	TMR1_CH1N		SSP2_CLK		-	
PA2	-		PA2	TMR1_CH2		SSP2_RXD		-	
PA3	-		PA3	TMR1_CH2N		SSP2_TXD		-	
PA6*	-		PA6/TCK	TMR1_CH4		-		-	
PA7*	-		PA7/TDO	TMR1_CH4N		-		-	
PA8*	-		PA8/TMS	TMR1_ETR		-		-	
PA9*	-		PA9/TDI	TMR1_BRK		-		-	
Порт B									
PB0	-		PB0	UART1_TXD	4	UART7816_TXD	11	TMR3_CH1	17
PB1	-		PB1	UART1_RXD		UART7816_RXD		TMR3_CH1N	
PB2	-		PB2	nSIROUT1		UART7816_CLK		TMR3_CH2	
PB3	-		PB3	nSIRIN1		UART7816_CTS		TMR3_CH2N	
PB5	-		PB5	nUART1RTS		UART7816_RTS	12	TMR3_CH3N	
PB10	-		PB10	TMR2_CH2	5	UART4_RXD	20	-	
PB11	-		PB11	TMR2_CH2N		UART4_TXD		-	
PB13	-		PB13	TMR2_CH3N		UART2_TXD	6	-	
PB14	-		PB14	TMR2_CH4		UART2_RXD		-	
Порт C									
PC0	-		PC0/MODE0	TMR4_ETR	19	-		TMR4_CH4	18
PC1	-	3	PC1/MODE1	TMR4_BRK		-		TMR4_CH4N	
PC2	-		PC2	TMR2_CH1	7	SSP3_FSS	14	I2C1_SCL	15
PC3	-		PC3	TMR2_CH1N		SSP3_CLK		I2C1_SDA	
PC4	-		PC4	EXT_INT2		SSP3_RXD		-	
PC5	-		PC5	EXT_INT3		SSP3_TXD		-	
Порт D									
PD2	-		PD2	SSP2_RXD		SD_V2_EXT		-	
PD4	-		PD4	UART2_TXD	6	-		-	
PD5	-		PD5	UART2_RXD		-		-	
PD6	-		PD6	UART3_RXD	10	-		-	
PD7	-		PD7	UART3_TXD		-		-	
PD8	-		PD8	UART4_RXD	20	-		-	
PD10	-		PD10	UART7816_TXD	11	SD_I0_EXT		-	
PD11	-		PD11	UART7816_RXD		SD_I1_EXT		-	
PD12	-		PD12	UART7816_CLK		SD_I2_EXT		-	
PD13	-		PD13	UART7816_CTS		SD_I3_EXT		-	

6 Указания по применению и эксплуатации

Запрещается подведение каких-либо электрических сигналов (в том числе шин питания и общий) к выходам микросхем, неиспользуемым согласно таблице 1.

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

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

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

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

7 Система питания

Микросхема имеет несколько типов выводов питания:

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

V_{CCB} – питание батарейного домена используется при отсутствии основного питания U_{CC} для питания батарейного домена. Переключение с основного питания на батарейное происходит автоматически при снижении уровня U_{CC} ниже ~1,5 В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 2 мс после превышения уровнем U_{CC} значения 1,6 В. Если в системе не требуется батарейное питание, вывод V_{CCB} должен быть объединен с V_{CC};

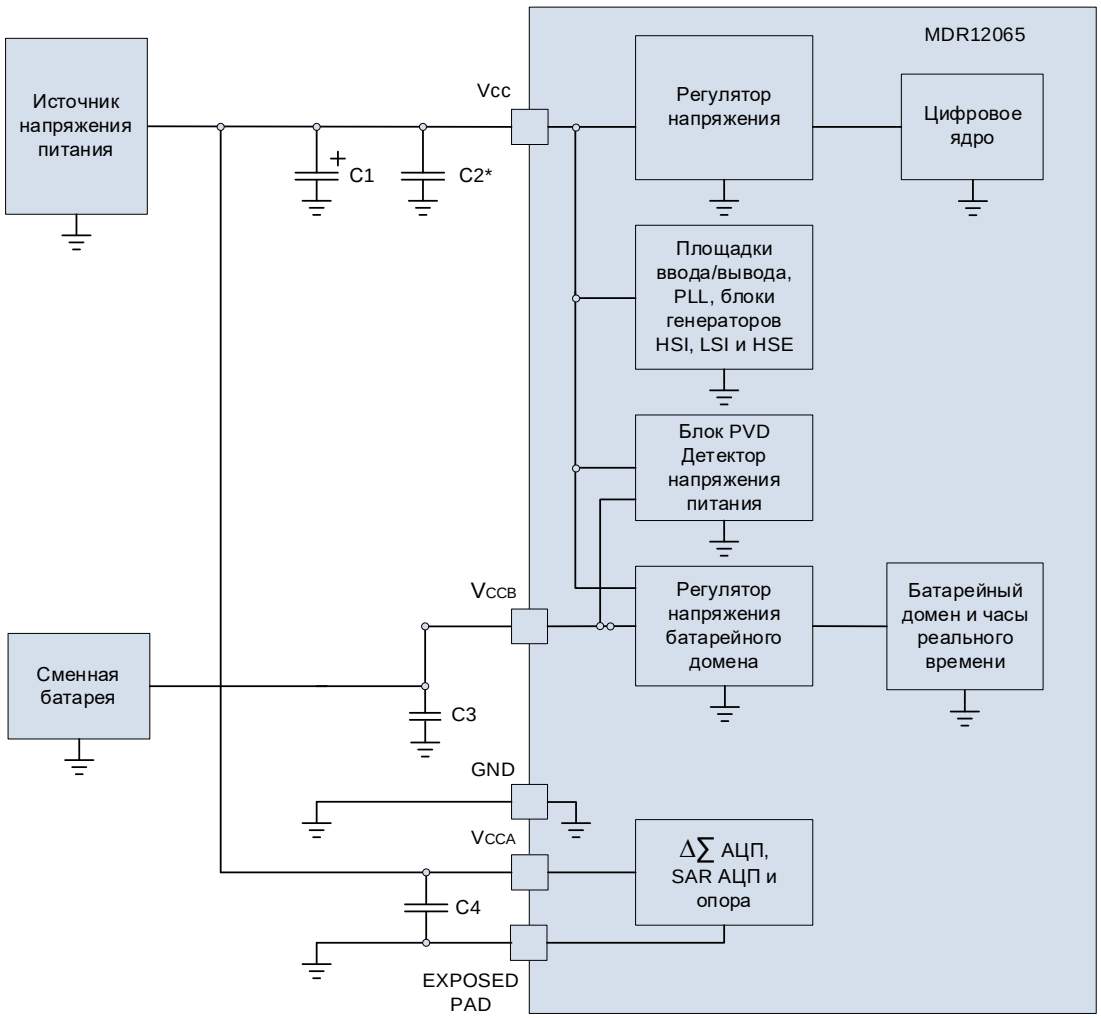
V_{CCA} – питание аналоговых блоков сигма-дельта АЦП, АЦП последовательного приближения и формирователя опоры выведено на отдельные выводы для уменьшения помех, создаваемых работой других блоков. На данные выводы должно подаваться напряжение с того же источника, что и на V_{CC}, но при этом на печатной плате должны быть применены меры по снижению наводки помех. Для корректной работы АЦП и $\Delta\sum$ АЦП напряжение питания U_{CCA} должно быть в пределах от 3,0 до 3,6 В. Если значение напряжения питания U_{CCA} находится в пределах от 1,8 до 3,0 В, то корректная работа АЦП и $\Delta\sum$ АЦП не гарантируется.

GND – основная «земля» питания;

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

Микросхема имеет встроенный детектор напряжений питания PVD и PVDB, подробнее см. раздел 19 «Детектор напряжения питания».

7.1 Структурные схемы подачи питания



C1 – конденсатор емкостью 22 мкФ;

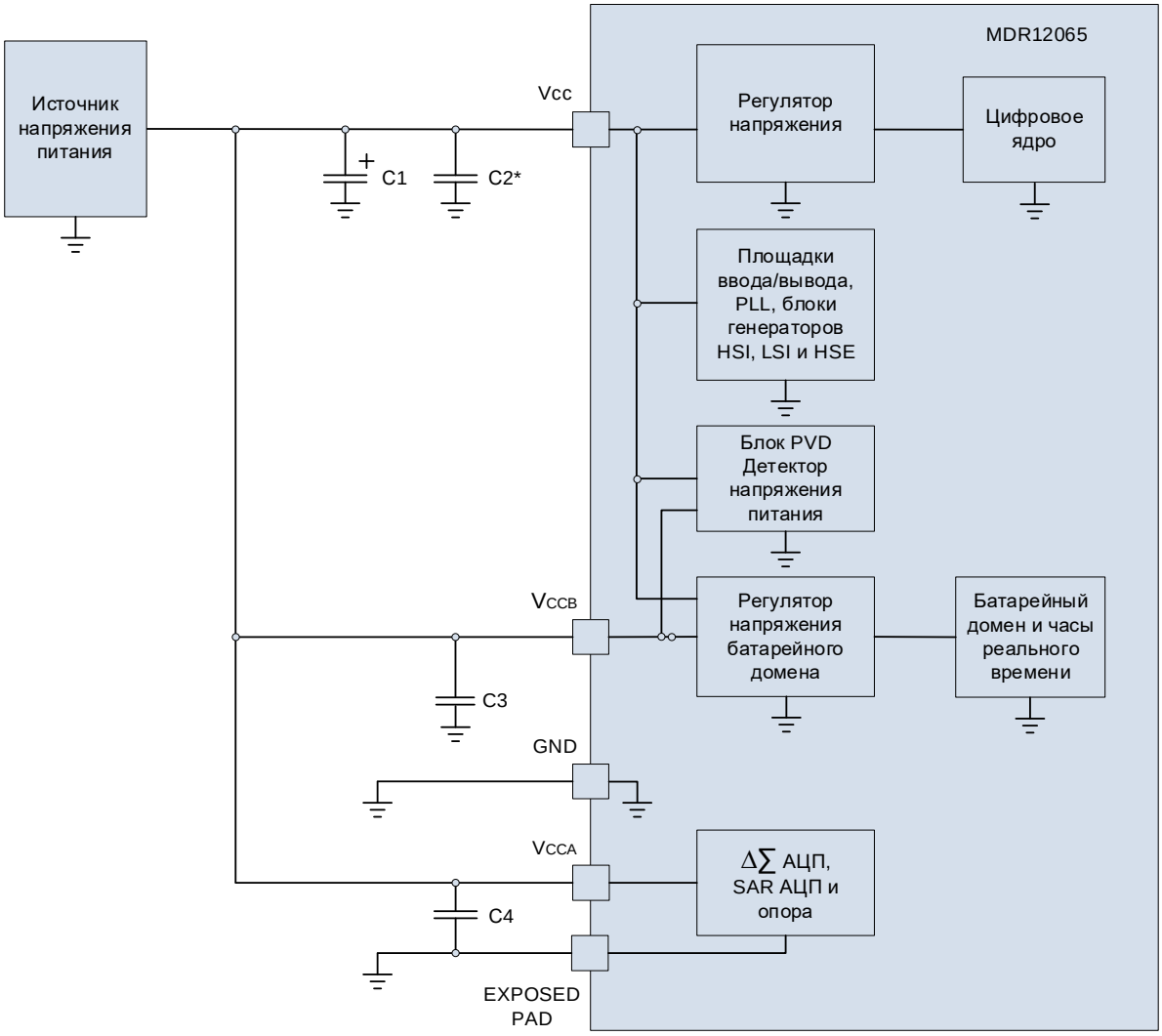
C2 – конденсаторы емкостью 1 мкФ;

C3 – конденсатор емкостью 10 мкФ;

C4 – конденсаторы емкостью 0,1 мкФ

* Конденсаторы должны быть установлены у каждого вывода питания.

Рисунок 4 – Структурная схема подачи питания микросхем с использованием батареи



C1 – конденсатор емкостью 22 мкФ;

C2 – конденсаторы емкостью 1 мкФ;

C3 – конденсатор емкостью 0,1 мкФ;

C4 – конденсатор емкостью 0,1 мкФ;

* Конденсаторы должны быть установлены у каждого вывода питания.

Рисунок 5 – Структурная схема подачи питания микросхем без использования батареи

Примечание – При отсутствии батарейного питания вывод U_{CCB} должен быть объединен с U_{CC}.

7.2 Схема сброса при включении и выключении основного питания

При включении питания вырабатывается внутренний сигнал сброса POR для цифровой части, питание U_{CC} нарастает и, пока оно не превысило уровень 1,6 В, сигнал сброса POR удерживается; после превышения данного уровня сигнал POR выдается еще на протяжении ~ 2 мс для того, чтобы гарантированно установилось напряжение питания, после чего сигнал POR снимается, и схема может начать работать.

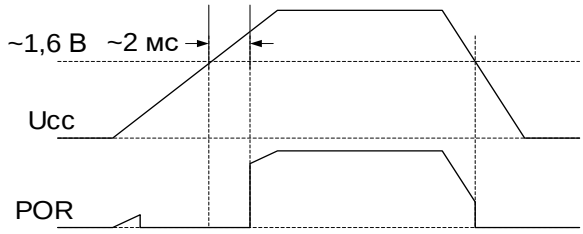


Рисунок 6 – Диаграмма формирования сигнала сброса микросхемы

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

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

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

Начальная установка микросхемы может быть произведена внешним сигналом сброса nRESET, внутренними сигналами сброса сторожевых таймеров или внутренним сигналом сброса ndmreset*, сформированным отладчиком. При этом сигнал nRESET формируется специальной схемой сброса, содержащей фильтр «иголок» и одновибратор для увеличения длительности этого сигнала.

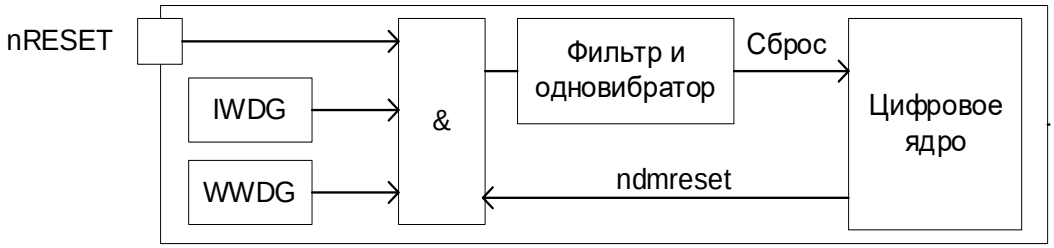


Рисунок 7 – Блок-схема формирования сброса

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

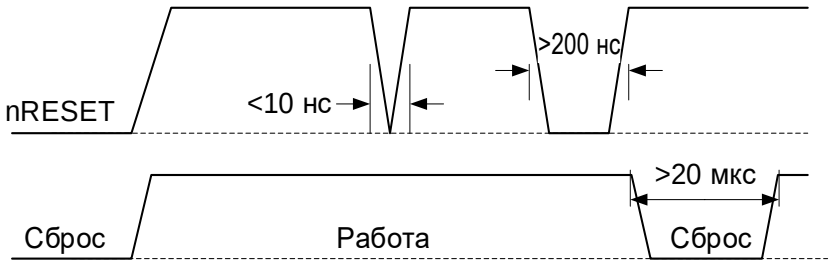


Рисунок 8 – Диаграмма фильтрации помех при формировании сигнала сброса

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

8 Организация памяти

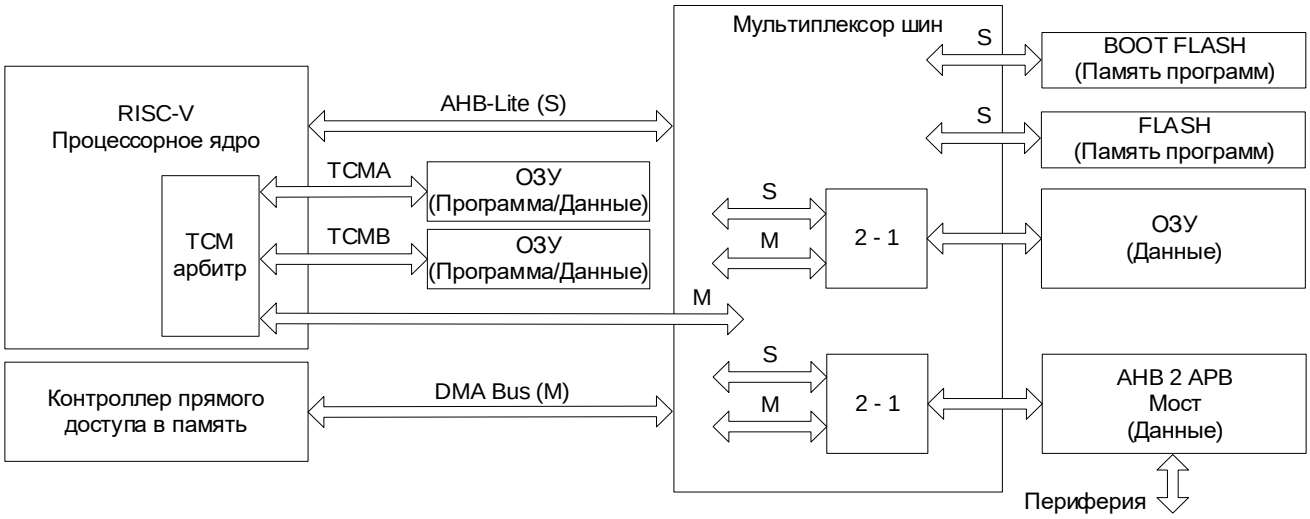


Рисунок 9 – Структурная схема организации памяти

Процессорное ядро имеет две системные шины:

– AHB-Lite – шина выборки инструкций и данных.

– TCMA, TCMB – шина выборки инструкций/данных.

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

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

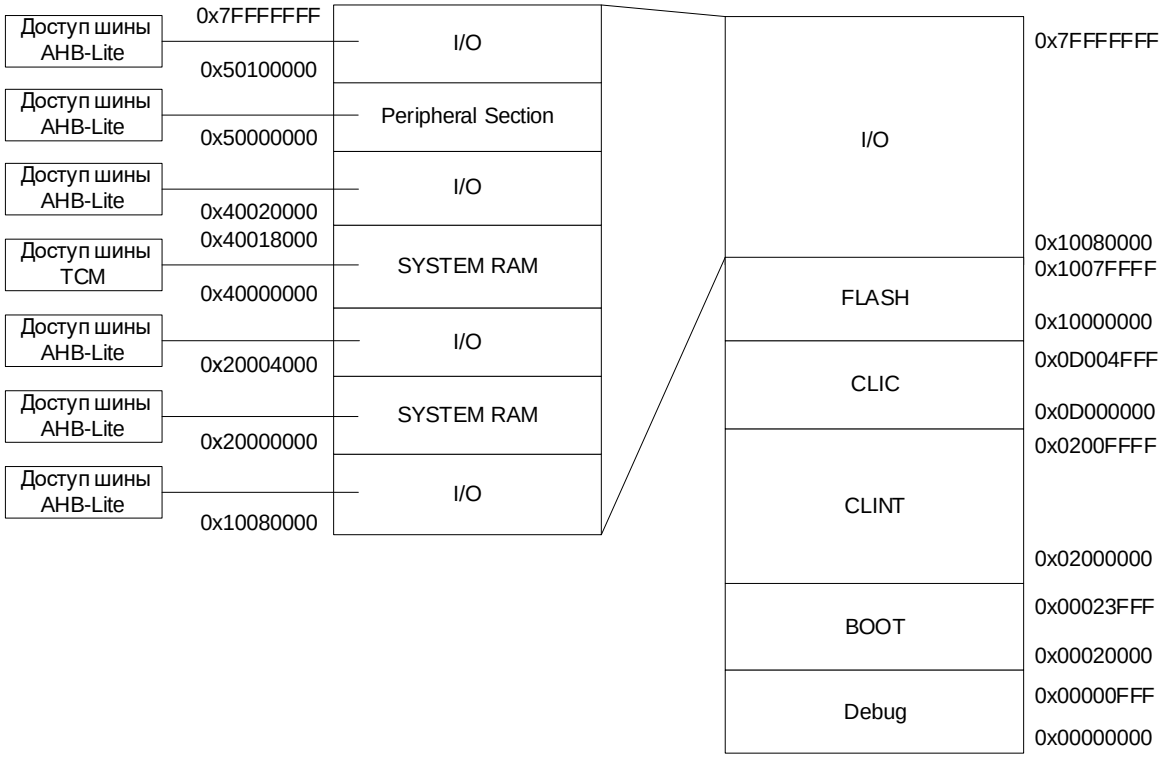


Рисунок 10 – Структура адресного пространства микросхемы

8.1 **Базовые адреса**

Таблица 3 – Базовые адреса микросхемы

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

9 Режим запуска

Режим запуска микросхемы определяется при включении микросхемы или после сброса любого типа.

При первом включении микросхемы режим запуска определяется уровнями на внешних выводах MODE[1:0] (PC[0], PC[1]), которые скофигурированы в цифровой режим на вход с включенной внутренней подтяжкой pull-down. После успешной работы загрузочной программы* устанавливается бит FPOR в регистре BKP_LDO.

При перезапусках микросхемы уровни на выводах MODE[1:0] не влияют на режим запуска микросхемы, если установлен бит FPOR, так как в этом случае режим запуска будет прочитан из бит MODE[1:0] регистра BKP_LDO.

Также загрузочная программа устанавливает калибровочные значения согласно алгоритму, описанному в подразделе 10.4 «Использование калибровочных значений».

Если бит FPOR не установлен, то загрузочная программа дополнительно выполняет алгоритим рекомендаций по пути обхода ошибки №0002, описанный в документе MDR12065 Errata Notice, если основное питание U_{CC} находится не ниже минимального уровня для его сравнения блоком PVD (PLS [2:0] = 0).

Если в процессе работы загрузочной программы было вызвано исключение, то микросхема зависает в бесконечном цикле в обработчике исключений «Trap_IRQHandler».

Загрузочная программа записана по адресам с 0x0002_0000 по 0x0002_07E3 включительно.

Таблица 4 – Режимы запуска микросхемы

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

* Исходный код данного варианта загрузочной программы доступен по ссылке https://ic.milandr.ru/soft/ или может предоставляться по запросу, отправленному на support@milandr.ru. Рекомендуется использовать актуальную версию загрузочной программы.

MODE[1:0]	Режим	Стартовый адрес/ таблица векторов прерываний	Описание
11	RAM+J	0x40000000	Микросхема начинает выполнять программу из информационной Flash-памяти, а затем передаёт управление в ОЗУ память программ. При этом разрешается работа отладочного интерфейса JTAG в случае отсутствия блокировки в Flash-памяти или его отключения в ВКР

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

В режиме отладки 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 Режим UART+J

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

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

9.1.1 Параметры связи по UART

Для связи по UART выбраны следующие параметры канала связи:

- начальная скорость – 9600 бод;
- количество бит данных – 8;
- четность – нет;
- количество Stop бит – 1;
- загрузчик не использует FIFO UART1;
- загрузчик всегда выступает в качестве Slave, а внешнее устройство, подающее команды – в качестве Master;
- данные передаются младшим битом вперед.

9.1.2 Протокол обмена по UART

После синхронизации с Master загрузчик переходит в диспетчер команд. Таким образом, Master-у доступны команды, приведенные в таблице 5.

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 Master-у приглашение (3 байта – 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>')).

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

9.2 Команды UART-загрузчика

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

9.2.1 Команда CMD_SYNC

Пустая команда.

Загрузчик (Slave) ее принимает, но ничего 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

9.2.2 **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 3 байта: – код команды CMD_CR; – код 0x0A; – код 0x3E (ASCII символ '>')

9.2.3 **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 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; – после передачи кода команды CMD_BAUD устанавливает новое значение скорости обмена

9.2.4 **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 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')

9.2.5 **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, 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')

9.2.6 **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
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 флаг FPOR; – устанавливает значение PC согласно принятому адресу (SP и CLIC не перепрограммируются) и, таким образом, Slave завершает свое выполнение. Если было передано управление в запрещенное адресное пространство для исполнения кода (например, адресное просранство прериферийных контроллеров), то выдается символ ошибки 0x66 (‘f’) с последующим зависанием микроконтроллера в бесконечном цикле в обработчике исключений

9.2.7 **Прием параметров команды**

Параметры команд – это 4-байтные числа.

Параметры передаются младшим байтом вперед.

В качестве значения параметра запрещено использовать число 0xFFFFFFFF.

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

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

9.2.8 **Сообщение об ошибке**

Сообщение об ошибке – 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.

10 Контроллер Flash-памяти программ

Микросхема содержит встроенную Flash-память программ объемом 2×256 Кбайт основной памяти программ и 2×8 Кбайт информационной памяти.

Структура памяти представлена на рисунке 11.

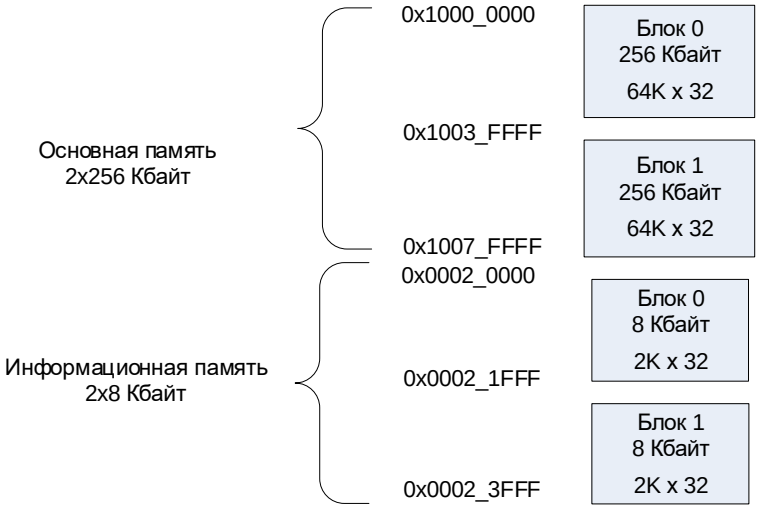


Рисунок 11 – Структура памяти

В обычном режиме (бит CON = 0, регистр FLASH_CMD) доступна основная память программ через системную шину для выборки инструкций и данных кода программы.

В режиме программирования (бит CON = 1, регистр FLASH_CMD) основная и информационная память доступны как периферийное устройство и могут быть использованы для нужд разработчика приложения. Ячейки Flash-памяти с адресом 0x0002_0000 и смещениями с 0x0000_3FE0 по 0x0000_3FFF содержат идентификационную информацию, калибровочные значения и слова для защиты адресного пространства Flash-памяти. Поэтому последние 4 Кбайта информационной памяти не рекомендуется использовать для хранения пользовательских данных или в качестве загрузочной области. При перезаписи загрузочной области необходимо обеспечить защиту от стирания и записи ячеек с адресами 0x0002_3FE0 – 0x0002_3FFF или предварительно сохранить их содержимое для последующего восстановления по адресам, указанным в таблице 13.

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 40 нс, в результате выдача новых значений из Flash-памяти может происходить с частотой не более 25 МГц. Для того, чтобы процессорное ядро могло получать новые инструкции на больших частотах добавляется пауза в один такт процессора для того, чтобы данные успели считаться из Flash-памяти. При работе с частотой ниже 25 МГц пауза

не требуется, так как Flash-память успевает выдать новые данные за один такт, а при частоте от 25 до 50 МГц требуется один такт паузы. Число тактов паузы задается в регистре FLASH_CMD битами Delay[2:0]. В таблице 12 приведены характеристики необходимой паузы для работы Flash-памяти программ.

Таблица 12 – Характеристики паузы для работы Flash-памяти программ

Delay [2:0]	Тактов паузы	Тактовая частота
0b000	0	до 25 МГц
0b001	1	до 50 МГц

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

10.2 Работа Flash-памяти программ в режиме программирования

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

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

- стирание всей памяти или 2×256 Кбайт основной памяти;
- стирание страницы памяти размером 4 Кбайт;
- запись 32-битного слова в память;
- чтение 32-битного слова из памяти.

10.2.1 Стирание всей памяти или 2×256 Кбайт основной памяти

Стирание памяти возможно только в режиме программирования.

Примечание – При стирании информационной области автоматически стирается и основная.

Для стирания всей памяти необходимо:

- установить бит IFREN в необходимое значение:
  - 0 – для основной памяти;
  - 1 – для основной и информационной памяти;
- выбрать блок FLASH 0 или 1, выставив ADR[18] и ADR[13];
- затем установить биты XE, MAS1 и ERASE в «1»;
- через время Tnvs (не менее 5 мкс) установить бит NVSTR в «1»;
- полное стирание памяти длится время Tme (от 20 до 40 мс). Спустя данное время необходимо очистить бит ERASE, и спустя время Tnvh1 (не менее 100 мкс) очистить биты XE, MAS1 и NVSTR.

– повторить для второго банка FLASH, установив инверсные ADR[18] и ADR[13].

Последующие операции с памятью можно выполнять спустя время Trcv (не менее 10 мкс).

Временная диаграмма стирания памяти представлена на рисунке 12.

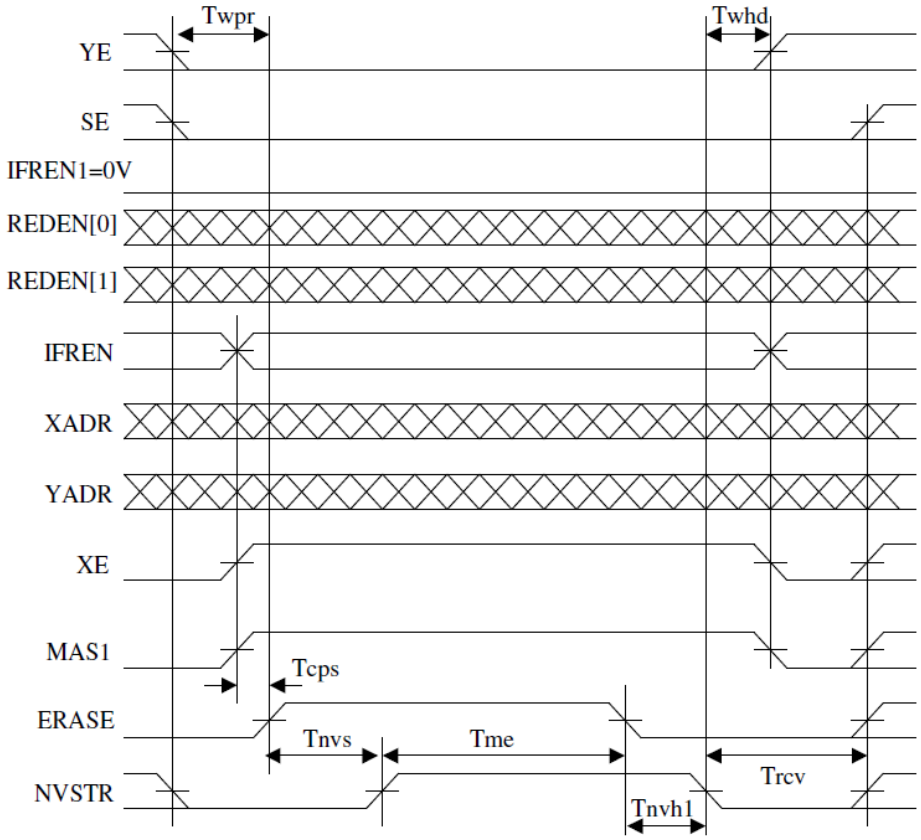


Рисунок 12 – Временная диаграмма стирания памяти

10.2.2 **Стирание страницы памяти размером 4 Кбайт**

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

– установить бит IFREN в необходимое значение:

  – 0 – для основной памяти;

  – 1 – для информационной памяти;

– затем установить адрес стираемой страницы в регистре FLASH_ADR и установить биты XE и ERASE в «1»;

– спустя время Tnvs (не менее 5 мкс) установить бит NVSTR в «1»;

– стирание страницы памяти длится время Terase (от 20 до 40 мс). Спустя данное время необходимо очистить бит ERASE, и спустя время Tnvh (не менее 5 мкс) очистить биты XE и NVSTR.

Последующие операции с памятью можно выполнять спустя время Trcv (не менее 10 мкс).

Временная диаграмма стирания страницы памяти представлена на рисунке 13.

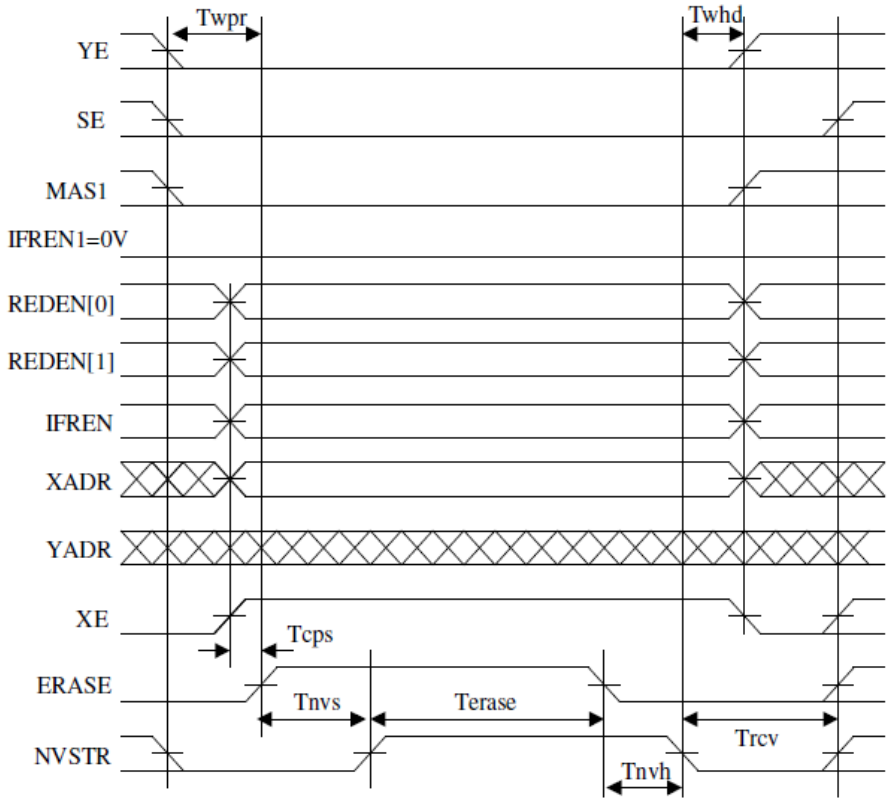


Рисунок 13 – Временная диаграмма стирания страницы памяти

10.2.3 **Запись 32-битного слова в память**

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

Для записи в память необходимо:

– установить бит IFREN в необходимое значение:

  – 0 – для основной памяти;

  – 1 – для информационной памяти;

– затем установить адрес, по которому производится запись, в регистре FLASH_ADR;

– в регистр FLASH_DI записать записываемое в память слово и установить биты XE и PROG в «1»;

– через время Tnvs (не менее 5 мкс) установить бит NVSTR в «1»;

– через время Tpgs (не менее 10 мкс) установить бит YE в «1»;

– запись в память длится время Tprog (от 20 до 40 мкс). Спустя данное время необходимо очистить бит YE.

Если необходимо записать следующее слово в память, то:

– через время Tadh (не менее 20 нс) после очистки бита YE установить новый адрес YADR (ADR[8:2]) и значение для записи в другую ячейку памяти;

– через время Tads (не менее 20 нс) установить YE в «1»;

– запись в память длится время Tprog (от 20 до 40 мкс). Спустя данное время необходимо очистить бит YE.

Если запись больше не требуется:

– через время Tpgh (не менее 20 нс) после очистки бита YE необходимо очистить бит PROG;

– через время Tnvh (не менее 5 мкс) очистить биты XE и NVSTR.

– последующие операции с памятью можно выполнять спустя время Trcv (не менее 10 мкс).

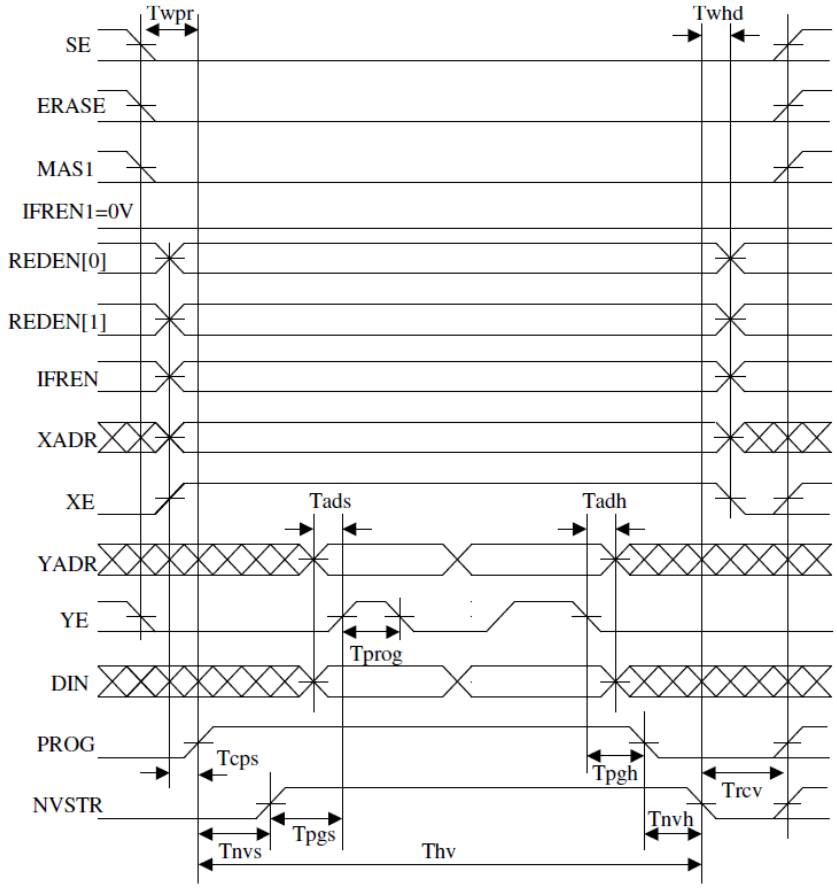


Рисунок 14 – Временная диаграмма записи памяти

Примечания

1 Flash-память поддерживает не менее 10000 циклов перезаписи.

2 Время Thv – суммарное время подачи высокого напряжения программирования на одну и ту же строку (один и тот же XADR) до следующего выполнения операции стирания этой же строки. Время Thv должно составлять не более 16 мс.

3 Данные могут быть запрограммированы по одному и тому же адресу более двух раз, только если после второй операции программирования была выполнена операция стирания.

4 В одну и ту же битовую ячейку нельзя записать «0» более двух раз, если после второй записи не была выполнена операция стирания.

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

В обычном режиме работы для чтения доступна основная и информационная память. Для этого необходимо просто считать требуемый адрес памяти. В режиме программирования для чтения доступна и основная и информационная память.

Для чтения из памяти необходимо:

– установить необходимое значение в бит IFREN:

  – 0 – для основной памяти;

  – 1 – для информационной памяти;

– в регистр FLASH_ADR записать адрес, из которого необходимо считать данные;

– установить биты XE и YE в «1», а затем через время Txys/Tas (не менее 1 нс) установить бит SE в «1»;

– через время Tacc (не менее 40 нс) из регистра FLASH_DO можно считать данные.

Если необходимо считать следующее слово, то:

– сбросить бит SE в «0» на время Tnws (не менее 5 нс);

– в регистр FLASH_ADR записать новый адрес;

– установить бит SE в «1»;

– спустя время Tacc (не менее 40 нс) из регистра FLASH_DO можно считать следующие данные.

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

Временная диаграмма чтения памяти представлена на рисунке 15.

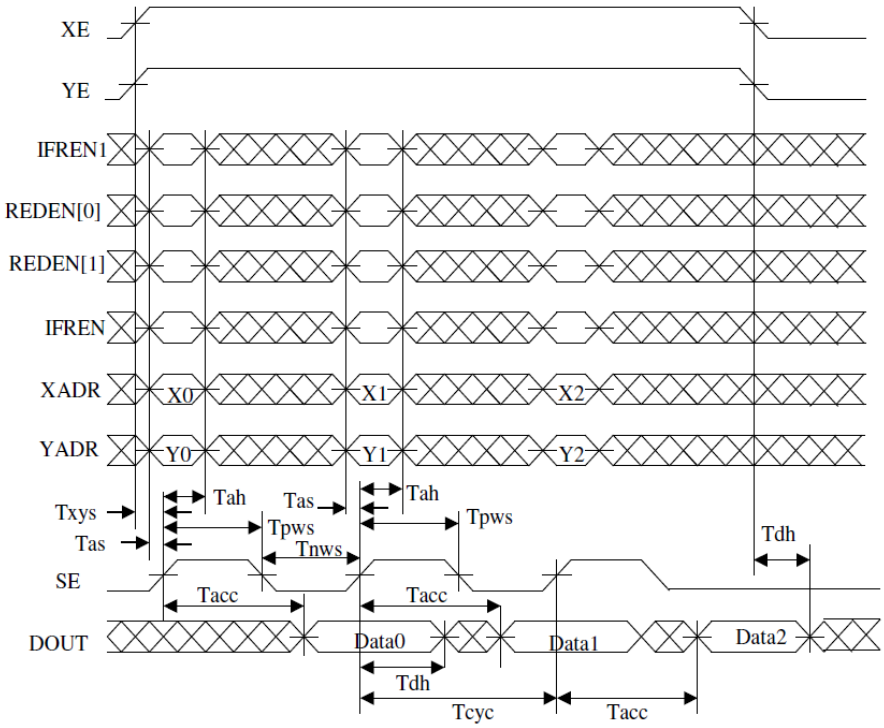


Рисунок 15 – Временная диаграмма чтения памяти

10.3 Производственная информация, калибровочные значения и защита Flash-памяти

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

Ячейки Flash-памяти с адресом 0x0002_0000 и смещениями с 0x0000_3FE0 по 0x0000_3FFF содержат идентификационную информацию, калибровочные значения и слово для защиты адресного пространства Flash-памяти. Поэтому данные области информационной памяти не рекомендуется использовать для хранения пользовательских данных или в качестве загрузочной области. При перезаписи загрузочной области необходимо обеспечить защиту от стирания и записи ячеек с адресами 0x0002_3FE0 – 0x0002_3FFF, либо предварительно сохранить их содержимое для последующего восстановления по адресам, указанным в таблице 13.

Таблица 13 – Описание специальных полей в Flash-памяти

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

Слово для защиты адресного пространства Flash-памяти состоит из двух полей:

– биты 7:0 – биты защиты, которые определяют, для каких областей Flash-памяти будет включена защита;

– биты 31:8 – ключ активации защиты, при значении 0x1234BC разрешается включение защиты областей Flash-памяти и блокировка интерфейса JTAG.

Чтение защитного слова выполняется при включении основного питания. Если значение ключа совпадает со значением 0x1234BC, то выполняется включение защиты адресного пространства Flash-памяти в соответствии с установленными битами защиты, а также блокировка интерфейса JTAG.

Биты защиты, расположенные по адресу 0x0002_3FFC, позволяют установить защиту для 8 областей Flash-памяти. Активное состояние бит защиты – логический «0». Каждый бит защиты отвечает за выбор соответствующего защищённого адресного пространства Flash-памяти:

- бит 0 – диапазон адресов 0x10000000-0x1000FFFF;
- бит 1 – диапазон адресов 0x10010000-0x1001FFFF;
- бит 2 – диапазон адресов 0x10020000-0x1002FFFF;
- бит 3 – диапазон адресов 0x10030000-0x1003FFFF;
- бит 4 – диапазон адресов 0x10040000-0x1004FFFF;
- бит 5 – диапазон адресов 0x10050000-0x1005FFFF;
- бит 6 – диапазон адресов 0x10060000-0x1006FFFF;

– бит 7 – диапазон адресов 0x10070000-0x1007FFFF.

При выборе хотя бы одного защищённого адресного пространства Flash-памяти также включается защита загрузочной области BOOT (информационная область Flash-памяти). Защита адресного пространства Flash-памяти выполняется следующим образом:

– в режиме программирования для выбранных защищённых адресных пространств Flash-памяти и загрузочной области BOOT блокируется выполнение операций стирания страницы, записи слова, а также чтения слова;

– в режиме программирования блокируется выполнение операции стирания всей памяти;

– в рабочем режиме выполняется защита чтения данных через системную шину из защищённых адресных пространств Flash-памяти и загрузочной области BOOT:

  – если программа выполняется из защищённого адресного пространства Flash-памяти или загрузочной области BOOT, то разрешается доступ по чтению ко всем областям Flash-памяти;

  – если программа выполняется из незащищённого адресного пространства Flash-памяти или любой области ОЗУ, то доступ по чтению к защищённым адресным пространствам Flash-памяти и загрузочной области BOOT блокируется (при чтении возвращаются нули).

Необходимо учитывать, что после включения защиты хотя бы для одного адресного пространства Flash-памяти перезапись защитного слова будет невозможна.

Если в защитное слово будет записано значение, в котором указано верное значение ключа, но не выбрано ни одно защищённое адресное пространство Flash-памяти (0x1234BCFF), то интерфейс JTAG будет заблокирован, при этом доступ ко всем областям Flash-памяти будет открыт как в рабочем режиме, так и в режиме программирования.

Защита Flash-памяти распространяется только на 8 областей в диапазоне адресов 0x10000000-0x1007FFFF, при этом доступ по чтению к Flash-памяти также дублируется в диапазоне адресов 0x10080000-0x1FFFFFFF, в котором защита Flash-памяти не осуществляется. Для ограничения доступа по чтению к Flash-памяти через адресное пространство 0x10080000-0x1FFFFFFF необходимо использовать блок защиты физической памяти (PMP).

10.4 Использование калибровочных значений

Значения битов UNIQUE_ID, TRIM_HSI, TRIM_BG, TRIM_LDO, TRIM_TS, TRIM_CS не обрабатываются микросхемой, их можно использовать только программно.

Если контрольная сумма TRIM_CS совпадает с рассчитанной по TRIM_HSI, TRIM_BG, TRIM_LDO, TRIM_TS, то в начале работы загрузочной программы выполняется:

- запись значения TRIM_HSI в биты HSI_LSI_TRIM регистра BKP_CLK;
- запись значения TRIM_BG в биты BG_TRIM регистра ADC_TRIM;
- запись значения TRIM_LDO в биты LDO_BOOST регистра BKP_LDO.

Если контрольная сумма TRIM_CS не совпадает с рассчитанной, то:

- в бит HSI_LSI_TRIM регистра BKP_CLK записывается значение «01 1110»;
- в бит BG_TRIM регистра ADC_TRIM записывается значение «1 0100».

Запись «медианных» значений в случае отсутствия/стирания калибровочных значений, записанных при производстве, не обеспечивает лучшие характеристики микросхемы и не гарантирует её стабильную работу во всех режимах. Поэтому в таких случаях рекомендуется производить калибровку самостоятельно.

Калибровка TRIM_TS не реализована.

Для микросхем без загрузочной программы алгоритм записи триммингов, приведенный выше, должен быть реализован самостоятельно.

10.5 Регистры управления контроллера Flash-памяти программ

Таблица 14 – Регистры управления контроллера Flash-памяти программ

Базовый Адрес	Название	Описание
0x5001_8000	FLASH_CNTRL	Регистры контроллера Flash-памяти программ
Смещение		
0x00	FLASH_CMD	Регистр управления Flash-памятью
0x04	FLASH_ADR	Регистр адреса
0x08	FLASH_DI	Регистр данных на запись
0x0C	FLASH_DO	Регистр данных считанных
0x10	FLASH_KEY	Регистр ключа FLASH
0x18	CHIP_ID_CTRL	Регистр CHIP_ID и управления OTP
0x1C	OTP_KEY	Регистр ключа OTP

10.5.1 **FLASH_CMD**

Таблица 15 – Регистр FLASH_CMD

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

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

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

Таблица 16 – Описание бит регистра FLASH_CMD

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...15	-	Зарезервировано
14	TMR	Сброс тестового режима Flash-памяти: 0 – разрешение теста; 1 – сброс. Рекомендовано всегда записывать «1»
13	NVSTR	Операции записи или стирания: 0 – при чтении; 1 – при записи или стирании
12	PROG	Записать данные из регистра FLASH_DI: 0 – нет записи; 1 – есть запись
11	MAS1	Стереть весь блок при ERASE = 1: 0 – нет стирания; 1 – стирание. При IFREN = 0 стирается основная память. При IFREN = 1 стирается основная и информационная память
10	ERASE	Стереть страницу памяти размером 4 Кбайт: 0 – нет стирания; 1 – стирание
9	IFREN	Работа с блоком информации: 0 – основная память; 1 – информационная память
8	SE	Усилитель считывания: 0 – не включен; 1 – включен

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	YE	Выдача адреса ADR[8:2]: 0 – не разрешено; 1 – разрешено
6	XE	Выдача адреса ADR[17:9]: 0 – не разрешено; 1 – разрешено
5…3	Delay[2:0]	Задержка памяти программ при чтении в циклах (в рабочем режиме): 000 – 0 цикл; 001 – 1 цикл
2, 1	-	Зарезервировано
0	CON	Переключение контроллера Flash-памяти на регистровое управление, не может производиться при исполнении программы из области FLASH: 0 – управление FLASH от ядра, рабочий режим; 1 – управление от регистров, режим программирования

10.5.2 **FLASH_ADR**

Таблица 17 – Регистр FLASH_ADR

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

Таблица 18 – Описание бит регистра FLASH_ADR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	ADR[31:0]	Адрес обращения в память. При IFREN = 0 (основная память): – ADR[18] – номер блока; – ADR[17:12] – адрес страницы (XADR[8:3]); – ADR[11:9] – адрес строки (XADR[2:0]); – ADR[8:2] – адрес слова (YADR[6:0]); – ADR[1:0] – не имеет значения. Минимально адресуемая ячейка 32 бита. При IFREN = 1 (информационная память): – ADR[13] – номер блока; – ADR[12] – адрес страницы (XADR[3]); – ADR[11:9] – адрес строки (XADR[2:0]); – ADR[8:2] – адрес слова (YADR[6:0]); – ADR[1:0] – не имеет значения. Минимально адресуемая ячейка 32 бита

10.5.3 **FLASH_DI**

Таблица 19 – Регистр FLASH_DI

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

Таблица 20 – Описание бит регистра FLASH_DI

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

10.5.4 **FLASH_DO**

Таблица 21 – Регистр FLASH_DO

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

Таблица 22 – Описание бит регистра FLASH_DO

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

10.5.5 **FLASH_KEY**

Таблица 23 – Регистр FLASH_KEY

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

Таблица 24 – Описание бит регистра FLASH_KEY

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

10.5.6 **CHIP_ID_CTRL**

Таблица 25 – Регистр CHIP_ID_CTRL

Номер	31…10	9…2	1	0
Доступ	U	R	R	R/W
Сброс	0	CHIP_ID	0/1	0
		CHIP_ID	OTPTST2	OTP_PROG

Таблица 26 – Описание бит регистра CHIP_ID_CTRL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10	-	Зарезервировано
9...2	CHIP_ID	11011001
1	OTPTST2	Тестовый бит, может быть в любом состоянии
0	OTP_PROG	При записи последовательности 1 – delay(1ms) – 0 – delay(1ms) – 1 – delay(1ms) – 0 – delay(1ms) – 1 – delay(1ms) – 0 происходит программирование ячеек памяти OTP. После этого при первой подаче питания на микросхему будет заблокирован JTAG (реализовано в микросхемах с ревизии 2) без возможности восстановления. Возможность записи в этот регистр по умолчанию заблокирована, разблокируется записью ключа в OTP_KEY

10.5.7 **OTP_KEY**

Таблица 27 – Регистр OTP_KEY

Номер	31…0
Доступ	R/W
Сброс	0
	OTP_KEY [31:0]

Таблица 28 – Описание бит регистра OTP_KEY

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	OTP_KEY[31:0]	Ключ для разрешения программирования ячеек памяти OTP. Для разблокирования записи в регистр CHIP_ID_CTRL нужно записать ключ 0x1555AAA8

11 Система команд

В микросхеме реализована система RISC-V команд RV32IMCNZX, включая:

– стандартное расширение RV32I;

– стандартное расширение M (целочисленное умножение и деление);

– стандартное расширение C (16-битные команды для увеличения плотности кода);

– стандартное расширение N (поддержка прерываний в пользовательском режиме привилегированности);

– стандартные расширения ZBA, ZBB, ZBC, ZBS (команды для битовых манипуляций);

– стандартные расширения ZBKB, ZBKC, ZBKX, ZKND, ZKNE, ZKNH, ZKSED, ZKSH для скалярной криптографии;

– специализированное XGOST расширение.

Поддерживаемые команды представлены в таблице 29.

Таблица 29 – Система команд процессора ВМ-310S0

Мнемокод команды	Формат	Операнды	Краткое описание
lb	I	rd, rs1, imm	Загрузка байта
lh	I	rd, rs1, imm	Загрузка полуслова
lw	I, Cx	rd, rs1, imm	Загрузка слова
lbu	I	rd, rs1, imm	Загрузка байта без знака
lhu	I	rd, rs1, imm	Загрузка слова без знака
sb	S	rs1, rs2, imm	Сохранение байта
sh	S	rs1, rs2, imm	Сохранение полуслова
sw	S, Cx	rs1, rs2, imm	Сохранение слова
add	R, Cx	rd, rs1, rs2	Арифметическое сложение
addi	I, Cx	rd, rs1, imm	Арифметическое сложение с непосредственным значением
sub	R, Cx	rd, rs1, rs2	Арифметическое вычитание
lui	U	rd, imm	Загрузка верхней части непосредственным значением
auipc	U	rd, imm	Арифметическое сложение верхней части PC с непосредственным значением
xor	R	rd, rs1, rs2	Логическое исключающее “ИЛИ”
xori	I	rd, rs1, imm	Логическое исключающее “ИЛИ” с непосредственным значением
or	R, Cx	rd, rs1, rs2	Логическое “ИЛИ”
ori	I	rd, rs1, imm	Логическое “ИЛИ” с непосредственным значением
and	R, Cx	rd, rs1, rs2	Логическое “И”
andi	I	rd, rs1, imm	Логическое “И” с непосредственным значением
sll	R	rd, rs1, rs2	Логический сдвиг влево
slli	I, Cx	rd, rs1, shamt	Логический сдвиг влево на непосредственное значение
srl	R	rd, rs1, rs2	Логический сдвиг вправо
srli	I	rd, rs1, shamt	Логический сдвиг вправо на непосредственное значение
sra	R	rd, rs1, rs2	Арифметический сдвиг вправо
srai	I	rd, rs1, shamt	Арифметический сдвиг вправо на непосредственное значение

Мнемокод команды	Формат	Операнды	Краткое описание
slt	R	rd, rs1, rs2	Сравнение “<”
slti	I	rd, rs1, imm	Сравнение “<” с непосредственным значением
sltu	R	rd, rs1, rs2	Сравнение “<” с беззнаковым
sltiu	I	rd, rs1, imm	Сравнение “<” с беззнаковым с непосредственным значением
beq	SB, Cx	rs1, rs2, imm	Переход в случае равенства
bne	SB, Cx	rs1, rs2, imm	Переход в случае неравенства
blt	SB	rs1, rs2, imm	Переход в случае меньше
bge	SB	rs1, rs2, imm	Переход в случае больше или равно
bltu	SB	rs1, rs2, imm	Переход в случае меньше беззнакового
bgeu	SB	rs1, rs2, imm	Переход в случае больше или равно беззнакового
jal	UJ, Cx	rd, imm	Переход & link
jalr	UJ, Cx	rd, rs1, imm	Переход & link register
fence.i	I	-	Синхронизация потока подкачки команд и потока чтения/записи данных
rdinstret	I	rd	Псевдоинструкция (alias) на команду чтения счетчика числа исполненных инструкций
ecall	I	-	Запрос на выполнение переменного окружения операционной системы
ebreak	I	-	Передача управления в переменное окружение отладки
rdcycle	I	rd	Псевдоинструкция (alias) на команду чтения счетчика количества выполненных циклов процессора
rdcycleh	I	rd	Псевдоинструкция (alias) на команду чтения старших бит счетчика количества выполненных циклов процессора
rdtime	I	rd	Псевдоинструкция (alias) на команду чтения значения низкочастотного системного таймера ядра
rdtimeh	I	rd	Псевдоинструкция (alias) на команду чтения значения старших бит низкочастотного системного таймера ядра
rdinstret	I	rd	Псевдоинструкция (alias) на команду чтения счетчика числа исполненных инструкций
rdinstreth	I	rd	Псевдоинструкция (alias) на команду чтения старших бит счетчика числа исполненных инструкций
mul	R	rd, rs1, rs2	Умножение
mulh	R	rd, rs1, rs2	Умножение с возвратом старших бит результата
mulhsu	R	rd, rs1, rs2	Умножение знакового на беззнаковое с возвратом старших бит результата
mulhu	R	rd, rs1, rs2	Умножение беззнакового на беззнаковое с возвратом старших бит результата
div	R	rd, rs1, rs2	Деление
divu	R	rd, rs1, rs2	Беззнаковое деление
rem	R	rd, rs1, rs2	Остаток от деления
remu	R	rd, rs1, rs2	Беззнаковый остаток от деления
andn	R	rd, rs1, rs2	Побитовое «И» rs1 и инверсии rs2
clmul	R	rd, rs1, rs2	Умножение без переноса (младшая часть)
clmulh	R	rd, rs1, rs2	Умножение без переноса (старшая часть)
clmulr	R	rd, rs1, rs2	Умножение без переноса реверсированных операндов с последующим реверсированием результата
clz	R	rd, rs	Количество ведущих нулей в rs1
cpop	R	rd, rs	Количество установленных бит в rs1

Мнемокод команды	Формат	Операнды	Краткое описание
ctz	R	rd, rs	Количество завершающих нулей в rs1
max	R	rd, rs1, rs2	Большее значение из двух знаковых целых
maxu	R	rd, rs1, rs2	Большее значение из двух беззнаковых целых
min	R	rd, rs1, rs2	Меньшее значение из двух знаковых целых
minu	R	rd, rs1, rs2	Меньшее значение из двух беззнаковых целых
orc.b	R	rd, rs1, rs2	Побитовое «ИЛИ» для всех бит каждого байта слова.
orn	R	rd, rs1, rs2	Побитовое «ИЛИ» для rs1 и побитно инвертированного rs2
rev8	R	rd, rs	Перестановка байт в обратном порядке
rol	R	rd, rs1, rs2	Циклический сдвиг влево rs1
ror	R	rd, rs1, rs2	Циклический сдвиг вправо rs1
rori	I	rd, rs1, shamt	Циклический сдвиг вправо на непосредственное значение
bclr	R	rd, rs1, rs2	Сброс бита с номером, заданным в rs2
bclri	I	rd, rs1, imm	Сброс бита с номером, заданным непосредственным значением
bext	R	rd, rs1, rs2	Извлечение бита с номером, заданным в rs2
bexti	I	rd, rs1, imm	Извлечение бита с номером, заданным непосредственным значением
binv	R	rd, rs1, rs2	Инвертирование бита с номером, заданным в rs2
binvi	I	rd, rs1, imm	Инвертирование бита с номером, заданным непосредственным значением
bset	R	rd, rs1, rs2	Установка бита с номером, заданным в rs2
bseti	I	rd, rs1, imm	Установка бита с номером, заданным непосредственным значением
sext.b	R	rd, rs	Расширение знака младшего байта rs
sext.h	R	rd, rs	Расширение знака полуслова rs
sh1add	R	rd, rs1, rs2	Сдвиг rs1 влево на 1 и сложение с rs2
sh2add	R	rd, rs1, rs2	Сдвиг rs1 влево на 2 и сложение с rs2
sh3add	R	rd, rs1, rs2	Сдвиг rs1 влево на 3 и сложение с rs2
xnor	R	rd, rs1, rs2	Инверсия исключающего «ИЛИ»
zext.h	R	rd, rs	Расширение нулями полуслова до размера слова

Примечание – В графе Операнды используются следующие обозначения:

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

Подробное описание и форматы команд приведены в следующих документах:

– «RISC-V Instruction Set Manual Volume I: User-Level ISA Document Version 2.2»;

– «RISC-V Bit-Manipulation ISA-extensions. Version 1.0.0»;

– «RISC-V Cryptography Extensions Volume I. Scalar and Entropy Source Instructions. Version 1.0.0».

11.1 Набор команд расширений для ускорения криптографии

Процессорное ядро реализует дополнительное расширение (xgost) системы команд для ускорения вычисления криптоалгоритмов, определяемых стандартами ГОСТ Р 34.12-2015 (блочные шифры), ГОСТ Р 34.11-2012 (функция хэширования) и ГОСТ Р 34.10-2012 (процессы формирования и проверки электронной цифровой подписи). Кодировка команд представлена в таблице 30. Команды xgost расширения исполняются один такт.

Таблица 30 – Криптографические инструкции GOST

funct7	rs2	rs1	funct3	rd	opcode	R-type
1000010	rs2	rs1	100	rd	0101011	gost32sb
1000011	rs2	rs1	100	rd	0101011	gost32sbi
1100101	rs2	rs1	100	rd	0101011	gost32kuleh
1100100	rs2	rs1	100	rd	0101011	gost32kule
1100110	rs2	rs1	100	rd	0101011	gost32kuldh
1100111	rs2	rs1	100	rd	0101011	gost32kuld
1101000	rs2	rs1	100	rd	0101011	gost32mgma
1010000	rs2	rs1	100	rd	0101011	gost32tau1
1010001	rs2	rs1	100	rd	0101011	gost32tau2
1010010	rs2	rs1	100	rd	0101011	gost32lin1
1010011	rs2	rs1	100	rd	0101011	gost32lin2

Команда gost32sb выполняет S-преобразование для каждого байта результата, исключающего ИЛИ регистров rs1 и rs2, и записывает результаты в соответствующие байты регистра rd. S-преобразование с исключающим ИЛИ используется в алгоритмах «Кузнечик» (блочный шифр) и «Стрибог» (функция хэширования). S-преобразование определено в разделе 4.1.1 стандарта ГОСТ Р 34.12-2015 и в разделе 5.2 стандарта ГОСТ Р 34.11-2012. Оба стандарта используют одинаковое S-преобразование.

Команда gost32sbi выполняет обратное S-преобразование для каждого байта регистра rs1 и выполняет с результатом исключающее ИЛИ с регистром rs2, после чего записывает результат регистр rd.

L-преобразование, используемое в алгоритме «Кузнечик» и определенное в разделах 4.1.2 и 4.2 ГОСТ Р 34.12-2015, выполняется при помощи команд gost32kuleh и gost32kule. L-преобразование выполняется над блоком данных размером 128 бит. L-преобразование является линейный преобразованием, которое эквивалетно умножению входного вектора на константную матрицу в поле Галуа GF(8). Для вычисления L-преобразования исходный блок данных размещается в четырех регистрах. Команды gost32kuleh и gost32kule используются для частичного вычисления результата умножения исходного вектора на матрицу преобразования. Для завершения вычисления каждой 32-битной части результата необходимо выполнить сложение результатов в поле GF(8) частичного умножения матрицы на часть исходного вектора, полученных командами gost32kuleh и gost32kule. Это сложение выполняется командой xor.

Пример кода, реализующего L-преобразования для блока, хранящегося в регистрах x1 (младшие 32 бита), x2, x3, x4 (старшие 32 бита):

```
gost32kule x5, x3, x4
gost32kuleh x4, x1, x2
xor x4, x4, x5
gost32kule x5, x2, x3
gost32kuleh x3, x4, x1
xor x3, x3, x5

gost32kule x5, x1, x2
gost32kuleh x2, x3, x4
xor x2, x2, x5
gost32kule x5, x4, x1
gost32kuleh x1, x2, x3
xor x1, x1, x5
```

Обратное L-преобразование выполняется аналогичным образом, при помощи команд gost32kuldh и gost32kuld:

```
gost32kuld x5, x3, x4
gost32kuldh x1, x1, x2
xor x1, x3, x5
gost32kuld x5, x4, x1
gost32kuldh x2, x2, x3
xor x2, x3, x5
gost32kuld x5, x1, x2
gost32kuldh x3, x3, x4
xor x3, x3, x5
gost32kuld x5, x2, x3
gost32kuldh x4, x4, x1
xor x4, x4, x5
```

G[k]-преобразование, используемое в алгоритме «Магма» (блочный шифр) и определенное в разделе 5.2 стандарта ГОСТ Р 34.12-2015, выполняется над блоком данных размером 64 бита, а его результат зависит от 32-битного раундового ключа k.

Для вычисления G[k]-преобразования используется команда gost32mgma. При этом старшие 32 бита исходного блока данных помещаются в регистр rs1, а раундовый ключ k – в младшие биты регистра rs2, младшие 32 бита результата G[k]-преобразования записывается в регистр rd. По свойству G[k]-преобразования старшие 32 бита результата преобразования равны младшим 32 битам исходного блока данных.

P-преобразование и l-преобразование, используемые в алгоритме «Стрибог» (функция хэширования) и определенные в разделах 5.3 и 5.4 стандарта ГОСТ Р 34.11-2012 соответственно, выполняются совместно при помощи команд gost32tau1, gost32tau2, gost32lin1, gost32lin2.

Команда gost32tau1 выполняет перемежение байт регистров rs1 и rs2, а затем записывает результат в регистр rd согласно следующему алгоритму:

```
uint32_t gost32tau1(uint32_t rs1, uint32_t rs2)
{
    uint32_t rd;
    rd =  (((rs1 >> 0) & 0xFF) << 0)      |
          (((rs1 >> 16) & 0xFF) << 8)     |
          (((rs2 >> 0) & 0xFF) << 16)     |
          (((rs2 >> 16) & 0xFF) << 24);

    return rd;
}
```

Команда gost32tau2 выполняет перемежение байт регистров rs1 и rs2, а затем записывает результат в регистр rd согласно следующему алгоритму:

```
uint32_t gost32tau2(uint32_t rs1, uint32_t rs2)
{
    uint32_t rd;
        rd = (((rs1 >> 8) & 0xFF) << 0) |
             (((rs1 >> 24) & 0xFF) << 8) |
             (((rs2 >> 8) & 0xFF) << 16) |
             (((rs2 >> 24) & 0xFF) << 24);
    return rd;
}
```

Команды gost32lin1 и gost32lin2 вычисляют младшие и старшие 32 бита результата l-преобразования соответственно. Результат записывается в регистр rd, а исходный 64-битный вектор в регистры rs1 и rs2.

11.2 **Поддержка Wait For Interrupt**

Процессорное ядро поддерживает команду WFI (Wait For Interrupt). При исполнении команды WFI процессорное ядро прекращает любую активность на внешних интерфейсах и переходит в режим ожидания прерываний. В режиме ожидания прерываний процессорное ядро для снижения энергопотребления отключает тактовый сигнал до появления прерывания.

Подробное описание и формат команды WFI приведен в документе «The RISC-V Instruction Set Manual, Volume II: Privileged Architecture. Version 1.11».

12 Процессорное ядро ВМ-310S0

ВМ-310S0 поддерживает два режима привилегированности: machine и user. Режим user предоставляет механизм изоляции процессов друг от друга и от доверенного кода, исполняемого в режиме machine. Более подробное описание режимов привилегированности см. в документе «The RISC-V Instruction Set Manual Volume II: Privileged Architecture».

Конвейер ВМ-310S0 состоит из трех стадий, на которых выполняются следующие операции:

1 стадия – генерация запроса в подсистему памяти программ (PMS);

2 стадия – чтение фрагмента кода из PMS и декодирование команды;

3 стадия – исполнение команды.

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

Блок FE состоит из следующих подблоков:

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

– IDU – Instruction Decode Unit – блок декодирования инструкций;

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

На стадии 1 блок IFU формирует запросы для считывания фрагментов кода в подсистему памяти программ (PMS), используя информацию о перенаправлении подкачки от IDU, RAS, а также подсистемы исполнения команд (BE).

На стадии 2 происходит чтение фрагмента кода из памяти программ. Фрагмент кода может содержать одну или две команды. После этого осуществляется декодирование одной команды и результат помещается в очередь декодированных команд (состоит из двух элементов) или в ее байпас к подсистеме исполнения команд (BE).

На стадии 3 подсистема исполнения команд (BE) получает декодированную команду и ее операнды из регистрового файла, осуществляет проверку возможности ее выполнения и выполняет на соответствующем исполнительном устройстве.

Команды, исполняемые процессорным ядром, имеют различную задержку исполнения. Команды I-расширения исполняются один такт, кроме команд чтения, которые исполняются два такта. Команды доступа к CSR-регистрам исполняются в пустом конвейере. Исполнительное устройство деления является итеративным и отвечает за исполнение команд DIV[U] и REM[U]. Длительности команд M-расширения приведены в таблице 31. Команды CLMUL, CLMULH и CLMULR из ZBC расширения команд исполняются восемь тактов. Остальные команды расширений для битовых манипуляций исполняются один такт. Команды из расширения для скалярной криптографии исполняются один такт.

Таблица 31 – Длительность выполнения команд

Команда	Длительность, такты
MUL	1
MULH	1
MULHS	1
MULHSU	1
DIV	от 2 до 16
DIVU	от 2 до 16
REM	от 2 до 16
REMU	от 2 до 16

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

12.1 Структурная схема процессора

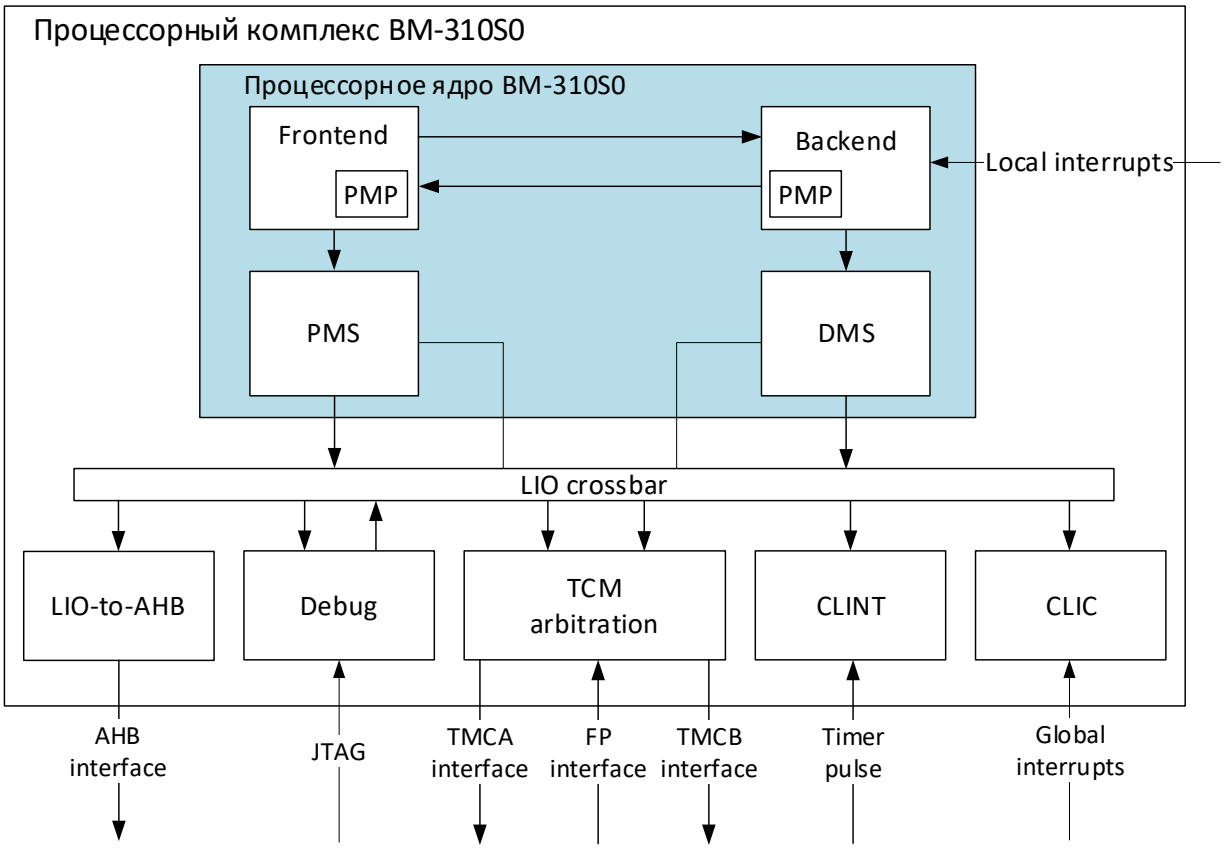


Рисунок 16 – Структурная схема процессора

12.1.1 **Интерфейс TCM**

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

12.1.2 **Арбитр доступа к блокам памяти TCM**

Внешние ведущие блоки на шине AHB (например, блок DMA) могут использовать ведомый AHB-интерфейс процессорного комплекса для доступа к блокам памяти TCM. При этом запросы от внешнего ведущего блока попадают в арбитр доступа к блокам памяти TCM, где между ними и запросами подсистем памяти программ (PMS) и данных (DMS) осуществляется арбитраж. Если запросы от PMS/DMS и от внешнего блока в одном такте попадают в разные блоки TCMA и TCMB, то доступ происходит одновременно (без конфликта). При попадании запросов в один блок памяти TCM запрос выбирается по следующему принципу:

– запрос от DMS имеет больший приоритет, чем запрос от PMS;

– для выбранного между PMS и DMS запроса и запроса внешнего ведущего блока осуществляется арбитраж по принципу round-robin.

Атомарность между доступами процессорного ядра и внешними ведущими блоками не гарантируется.

12.1.3 **Интерфейс AHB I/O**

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

12.1.4 **Запросы прерываний**

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

12.1.5 **Интерфейс JTAG**

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

12.1.6 **LIO crossbar**

Локальная шина ядра.

12.1.7 **Защита физической памяти (PMP)**

Для обеспечения безопасной обработки данных и предотвращения сбоев желательно ограничить физические адреса, доступные программному обеспечению. Дополнительный модуль защиты физической памяти (PMP) предоставляет регистры управления в M-режиме, позволяющие задавать права доступа к физической памяти (чтение, запись, выполнение) для каждой области физической памяти.

Степень детализации настроек контроля доступа PMP зависит от платформы и может варьироваться в зависимости от области физической памяти, но стандартная кодировка PMP поддерживает области размером до четырех байт. Привилегии определенных областей могут быть заданы программно – например, некоторые области могут быть видны только в M-режиме, но не на уровнях с более низкими привилегиями.

PMP-проверки применяются ко всем обращениям, когда ядро работает в U-режиме, а также к операциям загрузки и сохранения, когда бит MPRV установлен в регистре mstatus, а поле MPP в регистре mstatus содержит U. Дополнительно, PMP-проверки могут применяться к доступам в M-режиме, и в этом случае сами регистры PMP блокируются, так что даже программное обеспечение M-режима не может изменить их без сброса ядра. По сути, PMP может предоставлять разрешения для U-режима, которые по умолчанию отсутствуют, и может отозвать разрешения для M-режима, который по умолчанию имеет полные разрешения. Нарушения PMP всегда фиксируются ядром.

12.1.7.1 **CSR-регистры защиты физической памяти**

PMP-записи описываются 8-разрядным регистром конфигурации и одним 32-разрядным регистром адреса. В некоторых настройках PMP дополнительно используется регистр адреса, связанный с предыдущей PMP-записью. Поддерживается до 8 записей PMP. Все поля PMP CSR-регистров являются WARL (запись любого значения, чтение только допустимых, легальных, значений) и могут быть обнулены. PMP CSR-регистры доступны только в M-режиме. Регистры конфигурации PMP плотно упакованы в CSR, чтобы свести к минимуму время переключения контекста. Для представленной архитектуры ядра реализовано два CSR-регистра pmpcfg0–pmpcfg1, которые содержат конфигурации pmp0cfg–pmp7cfg для 8 PMP-записей.

Таблица 32 – CSR регистры PMP (Machine)

Адрес	Название	Доступ	Описание
0x3A0	pmpcfg0	R/W	Регистр 0 конфигурации физической защиты памяти
0x3A1	pmpcfg1	R/W	Регистр 1 конфигурации физической защиты памяти
0x3B0	pmpaddr0	R/W	Регистр 0 адреса физической защиты памяти
0x3B1	pmpaddr1	R/W	Регистр 1 адреса физической защиты памяти
0x3B2	pmpaddr2	R/W	Регистр 2 адреса физической защиты памяти

Адрес	Название	Доступ	Описание
0x3B3	pmpaddr3	R/W	Регистр 3 адреса физической защиты памяти
0x3B4	pmpaddr4	R/W	Регистр 4 адреса физической защиты памяти
0x3B5	pmpaddr5	R/W	Регистр 5 адреса физической защиты памяти
0x3B6	pmpaddr6	R/W	Регистр 6 адреса физической защиты памяти
0x3B7	pmpaddr7	R/W	Регистр 7 адреса физической защиты памяти

Таблица 33 – Компоновка CSR-регистра конфигурации PMP pmpcfg0

Номер	31...24	23...16	15…8	7…0
Доступ	WARL	WARL	WARL	WARL
Сброс	00000XXX	00000XXX	00000XXX	00000XXX
	pmp3cfg	pmp2cfg	pmp1cfg	pmp0cfg

Таблица 34 – Компоновка CSR-регистра конфигурации PMP pmpcfg1

Номер	31...24	23…16	15...8	7...0
Доступ	WARL	WARL	WARL	WARL
Сброс	00000XXX	00000XXX	00000XXX	00000XXX
	pmp7cfg	pmp6cfg	pmp5cfg	pmp4cfg

Регистры PMP-адресов – это CSR с именами pmpaddr0-pmpaddr7. Каждый регистр PMP-адреса кодирует биты 33…2 34-разрядного физического адреса, как показано в таблице 35. Не все биты физического адреса могут быть реализованы, и поэтому регистры pmpaddr являются WARL.

Таблица 35 – Регистры адреса PMP pmpaddr0-pmpaddr7

Номер	31..0
Доступ	WARL
Сброс	0
	address

Таблица 36 – Описание бит регистров адреса PMP pmpaddr0-pmpaddr7

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	address	Биты 33…2 для 34-разрядного физического адреса

В таблице 37 показан формат регистра конфигурации PMP. Биты R, W и X, если они установлены, указывают, что PMP-запись разрешает чтение, запись и выполнение команды соответственно. Если один из этих бит очищен, то соответствующий тип доступа запрещен. Комбинация R = 0 и W = 1 зарезервирована для использования в будущем. Поля A и L описаны в следующих разделах.

Таблица 37 – Регистры конфигурации PMP pmp0cfg-pmp7cfg

Номер	7	5…6	4...3	2	1	0
Доступ	WARL	WARL	WARL	WARL	WARL	WARL
Сброс	0	00	00	X	X	X
	L	-	A	X	W	R

Таблица 38 – Описание бит регистров конфигурации PMP pmp0cfg-pmp7cfg

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	L	Блокировка соответствующей PMP-записи: 1 – PMP-запись заблокирована, разрешения R/W/X применяются для доступа в M-режиме и U-режиме, запись в регистр конфигурации и соответствующие регистры адресов игнорируются; 0 – PMP-запись разблокирована, доступ в M-режиме осуществляется без ограничений, разрешения R/W/X применяются только для доступа в U-режиме
6...5	-	Зарезервировано. Рекомендуется не изменять значение поля
4...3	A	Кодирование режима сопоставления адресов: 00 – нулевая область (запись отключена); 01 – верхняя граница диапазона (TOR); 10 – область с выравниванием по четыре байта (NA4); 11 – области с выравниванием по степени 2 (NAPOT), ≥8 байт
2	X	Разрешение выполнение команды: 1 – разрешено; 0 – запрещено
1	W	Разрешение выполнения записи: 1 – разрешено; 0 – запрещено
0	R	Разрешение выполнения чтения: 1 – разрешено; 0 – запрещено

Попытка получить команду из области PMP, у которой нет разрешений на выполнение, вызывает исключение доступа при выборке. Попытка выполнить команду загрузки, адрес которой находится в области PMP, без разрешения на чтение вызывает исключение доступа при загрузке. Попытка выполнить команду сохранения, адрес которой находится в области PMP, без разрешений на запись вызывает исключение доступа при сохранении.

12.1.7.2 **Сопоставление адресов**

Поле A в регистре конфигурации PMP-записи кодирует режим сопоставления адресов соответствующего регистра адреса PMP. Кодировка этого поля приведена в таблице 39. Когда A = 0, эта PMP-запись отключена и не соответствует адресам. Поддерживаются два других режима сопоставления адресов: области с выравниванием по степени 2 (NAPOT), включая особый случай областей с выравниванием по четыре байта (NA4); и верхняя граница произвольного диапазона (TOR). Эти режимы поддерживают четырехбайтовую детализацию.

Таблица 39 – Декодирование поля A в регистре конфигурации PMP

A	Обозначение режима	Расшифровка
0	OFF	Нулевая область (запись отключена)
1	TOR	Верхняя граница диапазона
2	NA4	Область с выравниванием по четыре байта
3	NAPOT	Области с выравниванием по степени 2, ≥8 байт

Если выбран параметр TOR, соответствующий регистр адреса формирует верхнюю часть диапазона адресов, а предыдущий регистр адреса PMP формирует нижнюю часть диапазона адресов. Если для PMP-записи *i* в поле A задано значение TOR, то запись соответствует любому адресу *y* таким образом, что $pmpaddr_{i-1} \leq y < pmpaddr_i$. Если в PMP записи *0* в поле A задано значение TOR, в качестве нижней границы используется ноль, и поэтому оно соответствует любому адресу $y < pmpaddr_0$.

При выборе области NAPOT соответствующий адресный регистр кодирует размер области (младшие разряды) и базовый адрес области (старшие разряды), как показано в таблице 40. Разделителем между размером и базовым адресом выступает бит равный нулю.

Таблица 40 – Декодирование области NAPOT в PMP-регистрах адреса и конфигурации

pmpaddr	pmpcfg.A	Соответствие типу и размеру
yyyy…yyyy	NA4	4 байта
yyyy…yyy0	NAPOT	8 байт
yyyy…yy01	NAPOT	16 байт
yyyy…y011	NAPOT	32 байта
…	…	
yy01…1111	NAPOT	2^{32} байт
y011…1111	NAPOT	2^{33} байт
0111…1111	NAPOT	2^{34} байт
1111…1111	NAPOT	2^{35} байт

В таблице 41 приведены примеры кодирования адресного регистра при использовании режима NAPOT.

Таблица 41 – Примеры кодирования адресного регистра в режиме NAPOT

Значение pmpaddr	Базовый адрес области	Размер области
0x1000_0000	0x4000_0000	8 байт
0x1000_0003	0x4000_0000	32 байта
0x1000_01FF	0x4000_0000	4 Кбайт

Программное обеспечение может определить степень детализации PMP, записав ноль в pmp0cfg, затем записав все единицы в pmpaddr0, а затем прочитав обратно pmpaddr0. Если G – это индекс набора младших разрядов, установленных в единицу, то степень детализации PMP составляет 2^{G+2} байт.

12.1.7.3 **Режим блокировки и привилегий**

Бит L указывает на то, что PMP-запись заблокирована, то есть записи в регистр конфигурации и соответствующие регистры адресов игнорируются. Заблокированные PMP-записи могут быть разблокированы только при сбросе ядра. Если PMP-запись *i* заблокирована, записи в pmp*i*cfg и pmpaddr*i* игнорируются. Кроме того, если в pmp*i*cfg.A установлено значение TOR, записи в $pmpaddr_{i-1}$ игнорируются. В дополнение к блокировке PMP-записи, бит L указывает, применяются ли разрешения R/W/X для доступа в M-режиме. Если установлен бит L, эти разрешения применяются для всех режимов привилегий. Если бит L сброшен, любой доступ в M-режиме, соответствующий PMP-записи, будет успешным, то разрешения R/W/X применяются только к U-режиму.

12.1.7.4 **Логика приоритета и сопоставления**

PMP-записи имеют статический приоритет. PMP-запись с наименьшим номером, соответствующая любому выравниванию байт доступа, определяет, будет ли этот доступ успешным или неудачным. PMP-запись должна соответствовать количеству байт запрашиваемых при доступе, в противном случае доступ завершится неудачей, независимо от бит L, R, W и X.

Если PMP-запись соответствует всем запрашиваемым при доступе байтам, то биты L, R, W и X определяют, будет ли доступ успешным или нет. Если бит L сброшен и режим привилегий доступа равен M, доступ будет успешным. В противном случае, если установлен бит L или режим привилегированного доступа U, доступ будет успешным, только если установлен бит R, W или X, соответствующий типу доступа.

13 Блок ΔΣ АЦП

Микросхема имеет в своем составе блок из четырех каналов 24-битных ΔΣ АЦП. Каждый канал АЦП оцифровывает входной сигнал с выходной частотой отсчетов до 16 кГц. Также каждый АЦП имеет независимый канал DMA, обеспечивая возможность сохранения данных в ОЗУ без участия процессора.

Размер FIFO данных по каждому каналу АЦП составляет восемь отсчётов. Отсчёты в буферах FIFO (регистры F0IxDAT и FxIDAT) каждого из каналов представлены в дополнительном коде, то есть являются знаковыми величинами;

Буфер FIFO каждого канала позволяет формировать запросы к DMA. Одиночный запрос ADCUIDMASREQx переходит в активное состояние, если буфер FIFO содержит 1 или более отсчетов. Блочный запрос ADCUIDMABREQx переходит в активное состояние, если буфер FIFO содержит 4 или более отсчетов;

Все каналы АЦП имеют независимые калибровочные коэффициенты наклона характеристики;

Есть возможность скорректировать фазы сигналов в каналах;

Для измерения постоянной составляющей необходимо настроить регистр ADCUI_MSC, поле FxISEL_HPF, в значение «1».

Для большинства применений нужно отключить интерграторы после фильтров HPF, включенные по умолчанию.

Структурные схемы каналов АЦП приведены на рисунках 17, 18, 19.

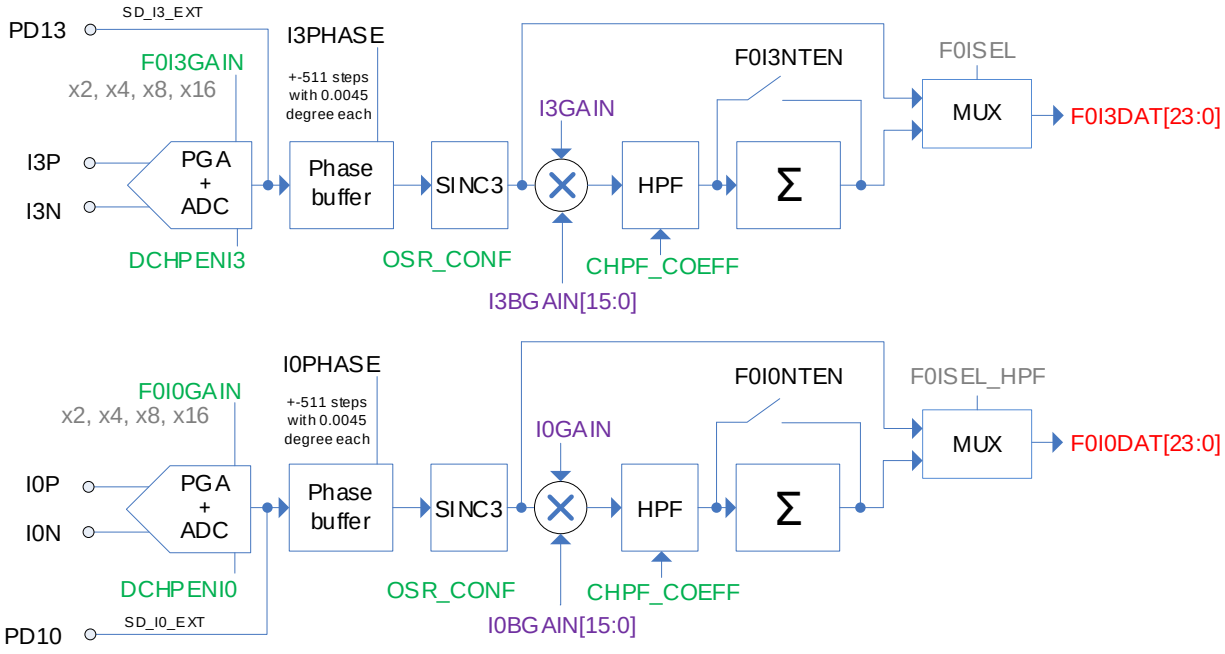


Рисунок 17 – Структурная схема F0

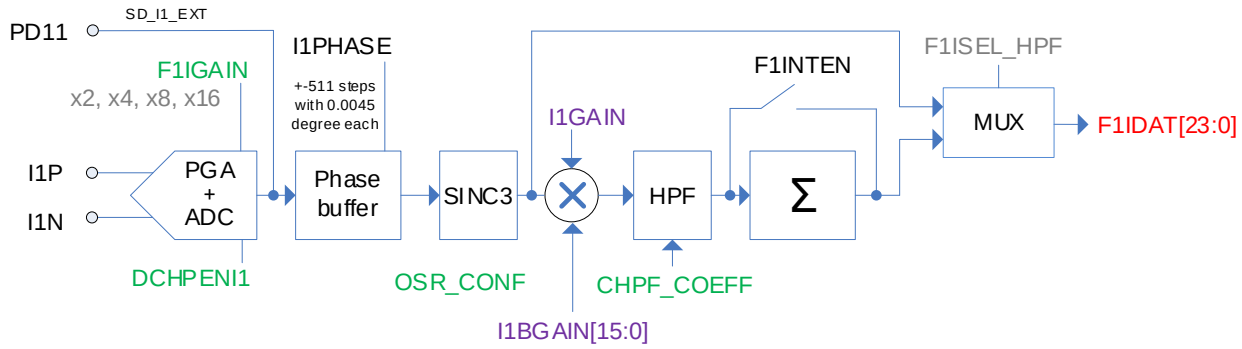


Рисунок 18 – Структурная схема F1

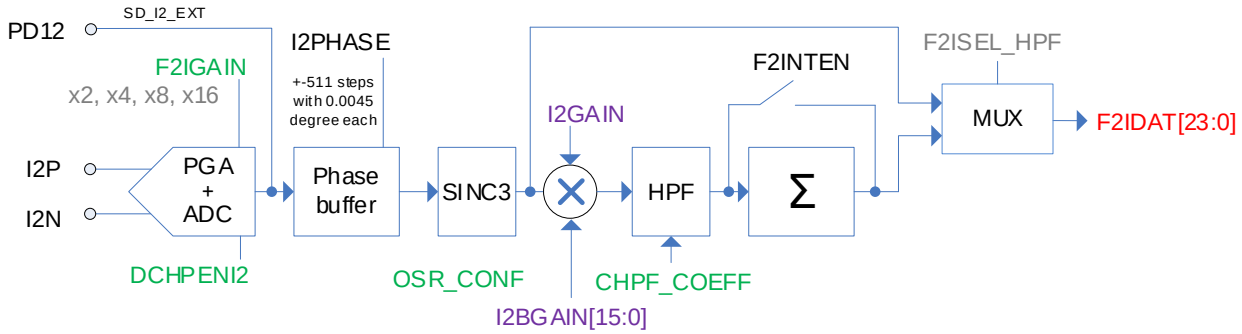


Рисунок 19 – Структурная схема F2

Для предотвращения влияния высокочастотных помех на результаты вычисления необходимо поставить внешний anti-aliasing фильтр. Например, можно использовать простейший RC-фильтр первого порядка с частотой среза порядка 10 – 100 кГц в зависимости от применения микросхемы.

Все цифровые фильтры (HPF, SINC3) настроены на указанные частоты среза при тактировании блока ADCUI от HSE = 8,192 МГц, при этом тактовая частота АЦП может быть равна либо 1,024 МГц (выходные частоты отсчётов 1/2/4 кГц в зависимости от настройки OSR_CONF), либо 2,048 МГц (выходные частоты отсчётов 2/4/8 кГц в зависимости от настройки OSR_CONF), либо 4,096 МГц (выходные частоты отсчётов 4/8/16 кГц в зависимости от настройки OSR_CONF) в зависимости от настройки поля CLC_ADC_CFG в регистре ADCUI_CLKPHASE1.

Передаточные характеристики фильтров HPF описываются формулой

$$H(z) = \frac{1}{2} \cdot \frac{1-c+(c-1)\cdot z^{-1}}{1+c\cdot z^{-1}}, \quad (1)$$

где c – значение в дополнительном коде поля CHPF_COEFF регистра ADCUI_FILTERCFG, делённое на 2^{11}.

Для значения CHPF_COEFF по сбросу c = 802h / 2^{11} = – 0,999023.

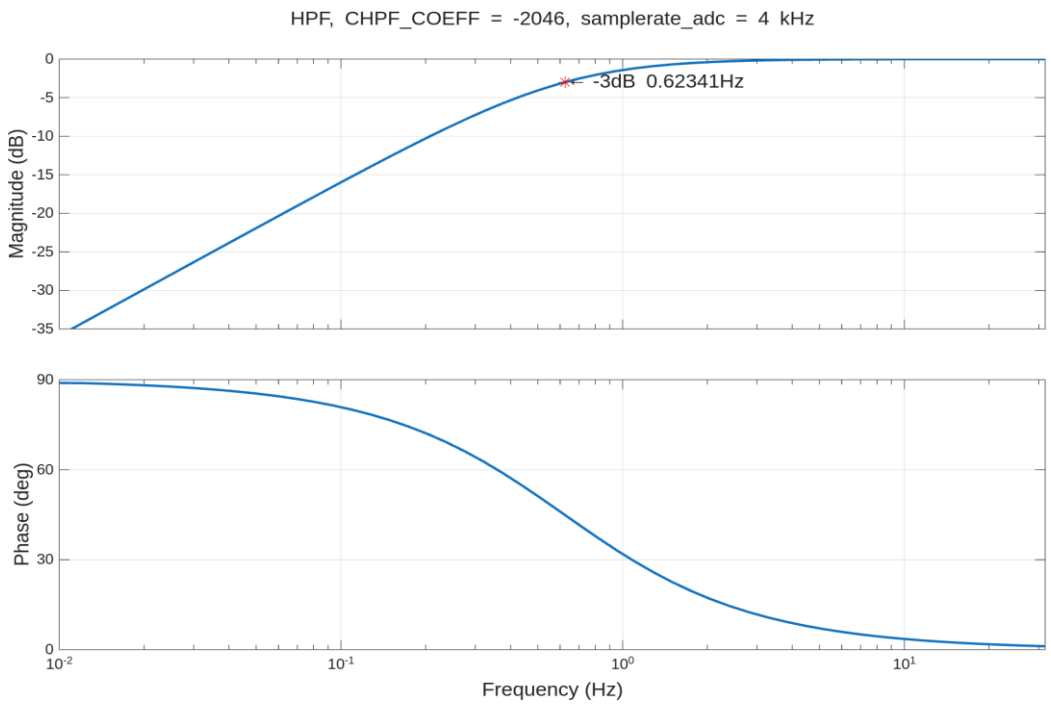


Рисунок 20 – ЛАЧХ и ФЧХ фильтра HPF при настройках по умолчанию на 4 кГц

13.1 Описание регистров управления блока четырех каналов АЦП

Таблица 42 – Регистры управления блока четырех каналов АЦП

Базовый Адрес	Название	Описание
0x5006_8000	ADCUI	Контроллер ΔΣ АЦП
Смещение		
0x000	ADCUI_CTRL1	Общее управление для контроллера АЦП
0x004	ADCUI_CTRL2	
0x008	–	Зарезервировано
0x00C	ADCUI_F0CTR	Управление в канале F0
0x010 – 0x024	–	Зарезервировано
0x028	ADCUI_F0AC	Управление FIFO в канале F0
0x02C – 0x040	–	Зарезервировано
0x044	ADCUI_F0I0DAT	Буфер FIFO отсчетов в канале АЦП I0 блока F0
0x048	ADCUI_F0I3DAT	Буфер FIFO отсчетов в канале АЦП I3 блока F0
0x04C – 0x058	–	Зарезервировано
0x05C	ADCUI_F0STAT	Статус канала F0
0x060	ADCUI_F0MASK	Маска прерываний канала F0
0x064	ADCUI_F1CTR	Управление в канале F1
0x068 – 0x09C	–	Зарезервировано
0x0A0	ADCUI_F1IDAT	Буфер FIFO отсчетов в канале АЦП I1 блока F1
0x0A4 – 0x0B0	–	Зарезервировано
0x0B4	ADCUI_F1STAT	Статус канала F1
0x0B8	ADCUI_F1MASK	Маска прерываний канала F1
0x0BC	ADCUI_F2CTR	Управление в канале F2
0x0C0 – 0x0F4	–	Зарезервировано
0x0F8	ADCUI_F2IDAT	Буфер FIFO отчетов в канале АЦП I2 блока F2

Базовый Адрес	Название	Описание
0x5006_8000	ADCUI	Контроллер ΔΣ АЦП
Смещение		
0x0FC – 0x108	–	Зарезервировано
0x10C	ADCUI_F2STAT	Статус канала F2
0x110	ADCUI_F2MASK	Маска прерываний канала F2
0x114	ADCUI_CCAL1	Регистр калибровки канала АЦП I0 в блоке F0
0x118	ADCUI_CCAL2	Регистр калибровки каналов АЦП I1 в блоке F1
0x11C	ADCUI_CCAL3	Регистр калибровки каналов АЦП I2 в блоке F2
0x120	ADCUI_CCAL4	Регистра калибровки канала АЦП I3 в блоке F0
0x124 – 0x170	–	Зарезервировано
0x174	ADCUI_MSC	Разрешение формирования запросов к DMA и др.
0x178	ADCUI_FILTERCFG	Управление конфигурацией фильтров
0x17C	ADCUI_CLKPHASE1	Управление сдвигом фазы каналов АЦП
0x180	ADCUI_CLKPHASE2	Управление сдвигом фазы каналов АЦП

13.1.1 **ADCUI_CTRL1**

Таблица 43 – Регистр ADCUI_CTRL1

Номер	31, 30	29	28	27…25	24, 23	22
Доступ	R/W	R/W	R/W	–	R/W	R/W
Сброс	00	0	0	–	00	0
	OSR_CONF	tst_in[4]	RESET_DIG	–	tst_in[3:2]	tst_in[1]

Номер	21	20	19…8	7	6	5
Доступ	R/W	R/W	–	R/W	R/W	R/W
Сброс	0	0	–	0	0	0
	tst_in[0]	VREF_SEL	–	RESOL	I3EN	V2EN

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

Таблица 44 – Описание бит регистра ADCUI_CTRL1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	OSR_CONF*	Выбор коэффициента передискретизации: 00 – OSR = 1024; 01 – OSR = 512; 10 – OSR = 256; 11 – зарезервировано. Примечание – Коэффициенты применимы к тактовой частоте f_{C_ADCD} блока АЦП, которая может быть равна 1,024, 2,048 или 4,096 МГц, в зависимости от настройки поля CLC_ADC_CFG в регистре ADCUI_CLKPHASE1
29	tst_in[4]	Тестовый выход (всегда записывать ноль)
28	RESET_DIG	Сброс цифровой части блоков АЦП: 0 – нет сброса; 1 – цифровая часть под общим сбросом
27…25	–	Зарезервировано
24, 23	tst_in[3:2]	Тестовые выходы (всегда записывать ноль)
22	tst_in[1]	Тестовый выход (всегда записывать ноль)
21	tst_in[0]	Тестовый выход (всегда записывать ноль)
20	VREF_SEL	Выбор опорного напряжения для АЦП: 0 – внешнее опорное напряжение с вывода VREF; 1 – внутреннее опорное напряжение VREF_INT 2,4 В (рисунок 58). При выборе VREF_SEL == 1 включается внутренний источник опорного напряжения VREF_INT 2,4 В, выходное напряжение VREF_INT 2,4 В выводится на вывод VREF
19…8	–	Зарезервировано
7	RESOL	Разрешение выходных данных: 0 – 16 бит; 1 – 24 бита Старший бит в регистре данных 15 или 23. Сдвиг регистра данных выполнять не требуется
6	I3EN	Разрешение работы канала I3: 0 – канал отключен; 1 – канал включен**
5	V2EN	Должен быть 0 для правильного функциорирования
4	I2EN	Разрешение работы канала I2: 0 – канал отключен; 1 – канал включен**
3	V1EN	Должен быть 0 для правильного функциорирования
2	I1EN	Разрешение работы канала I1: 0 – канал отключен; 1 – канал включен**
1	V0EN	Должен быть 0 для правильного функциорирования
0	I0EN	Разрешение работы канала I0: 0 – канал отключен; 1 – канал включен**

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
* При изменении частоты дискретизации необходимо соответствующим образом корректировать настройки цифровых фильтров HPF для сохранения их частоты среза. 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. *** При включении любого канала в блоке Fx, включаются FIFO всех остальных каналов (FIFO FxIDAT) в блоке Fx		

13.1.2 **ADCUI_CTRL2**

Таблица 45 – Регистр ADCUI_CTRL2

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

Таблица 46 – Описание бит регистра ADCUI_CTRL2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	–	Зарезервировано
30	DCHPENI3	0 – чоппер канала I3 выключен; 1 – чоппер канала I3 включен*
29	DCHPENI2	0 – чоппер канала I2 выключен; 1 – чоппер канала I2 включен*
28	DCHPENI1	0 – чоппер канала I1 выключен; 1 – чоппер канала I1 включен*
27	DCHPENI0	0 – чоппер канала I0 выключен; 1 – чоппер канала I0 включен*
26…0	–	Зарезервировано
* Рекомендуется использовать при измерении постоянных сигналов. При использовании внешнего дельта-сигма модулятора должен быть выключен		

13.1.3 **ADCUI_F0CTR**

Таблица 47 – Регистр ADCUI_F0CTR

Номер	31…20	19, 18	17…8	7, 6	5…2	1	0
Доступ	–	R/W	–	R/W	–	R/W	R/W
Сброс	–	00	–	0	–	0	0
	–	F0I3GAIN	–	F0I0GAIN	–	F0I3NTEN	F0I0NTEN

Таблица 48 – Описание бит регистра ADCUI_F0CTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...20	–	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
19, 18	F0I3GAIN	Предусилитель в канале АЦП I3: 00 – 6 дБ (x2); 01 – 12 дБ (x4); 10 – 18 дБ (x8); 11 – 24 дБ (x16)
17...8	–	Зарезервировано
7, 6	F0I0GAIN	Предусилитель в канале АЦП I0: 00 – 6 дБ (x2); 01 – 12 дБ (x4); 10 – 18 дБ (x8); 11 – 24 дБ (x16)
5…2	–	Зарезервировано
1	F0I3NTEN	Отключение интегратора в канале АПЦ I3: 0 – интегратор включен; 1 – интегратор отключен. Корректная работа интегратора гарантируется только при частоте дискретизации 4 или 8 кГц – при условии, что значение CHPF_COEFF обеспечивает частоту среза фильтров HPF на уровне 1 Гц для выбранной частоты дискретизации. На частоте дискретизации 16 кГц корректная работа интегратора не гарантируется. Например: – при частоте дискретизации 4 кГц значение CHPF_COEFF должно быть 0x803; – при частоте дискретизации 8 кГц значение CHPF_COEFF должно быть 0x802 (значение по умолчанию). Примечание – Для обычного применения, где функционал с аппаратным сдвигом фазы в канале не требуется, интегратор рекомендуется отключать
0	F0I0NTEN	Отключение интегратора в канале АПЦ I0: 0 – интегратор включен; 1 – интегратор отключен. Корректная работа интегратора гарантируется только при частоте дискретизации 4 или 8 кГц – при условии, что значение CHPF_COEFF обеспечивает частоту среза фильтров HPF на уровне 1 Гц для выбранной частоты дискретизации. На частоте дискретизации 16 кГц корректная работа интегратора не гарантируется. Например: – при частоте дискретизации 4 кГц значение CHPF_COEFF должно быть 0x803; – при частоте дискретизации 8 кГц значение CHPF_COEFF должно быть 0x802 (значение по умолчанию). Примечание – Для обычного применения, где функционал с аппаратным сдвигом фазы в канале не требуется, интегратор рекомендуется отключать

13.1.4 **ADCUI_F0AC**

Таблица 49 – Регистр ADCUI_F0AC

Номер	31…13	12	11…0
Доступ	–	R/W	–
Сброс	–	0	–
	–	F0I3SEL	–

Таблица 50 – Описание бит регистра ADCUI_F0AC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...13	–	Зарезервировано
12	F0I3SEL	Выбор источника сигнала для FIFO канала АПЦ I3 (регистр ADCUI_F0I3DAT): 0 – отсчеты после фильтра высоких частот; 1 – отсчеты до фильтра высоких частот
11...0	–	Зарезервировано

13.1.5 **ADCUI_F0I0DAT**

Таблица 51 – Регистр ADCUI_F0I0DAT

Номер	31…24	23…0
Доступ	–	RO
Сброс	–	000000h
	–	F0I0DAT

Таблица 52 – 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 АЦП канала I0 FIFO включается при установке любого из битов I3EN или I0EN

13.1.6 **ADCUI_F0I3DAT**

Таблица 53 – Регистр ADCUI_F0I3DAT

Номер	31…24	23…0
Доступ	–	RO
Сброс	–	000000h
	–	F0I3DAT

Таблица 54 – 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 АЦП канала I3. FIFO включается при установке любого из битов I3EN или I0EN

13.1.7 **ADCUI_F0STAT**

Таблица 55 – Регистр ADCUI_F0STAT

Номер	31…25	24	23	22	21…6	5
Доступ	–	R/W	RO	RO	–	R/W
Сброс	–	0	0	0	–	0
	–	C3IF_OVR	C3IF_FLL	C3IF_EMP	–	F0IF_OVR

Номер	4	3	2…0
Доступ	RO	RO	–
Сброс	0	0	–
	F0IF_FLL	F0IF_EMP	–

Таблица 56 – Описание бит регистра ADCUI_F0STAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…25	–	Зарезервировано
24	C3IF_OVR	Флаг, что произошло переполнения FIFO F0I3DAT. Запись «1» сбрасывает этот флаг
23	C3IF_FLL	Флаг, что FIFO F0I3DAT заполнено
22	C3IF_EMP	Флаг, что FIFO F0I3DAT пусто
21…6	–	Зарезервировано
5	F0IF_OVR	Флаг, что произошло переполнение FIFO F0IDAT. Запись «1» сбрасывает этот флаг
4	F0IF_FLL	Флаг, что FIFO F0IDAT заполнено
3	F0IF_EMP	Флаг, что FIFO F0IDAT пусто
2…0	–	Зарезервировано

13.1.8 **ADCUI_F0MASK**

Таблица 57 – Регистр ADCUI_F0MASK

Номер	31…25	24	23	22	21…6	5
Доступ	–	R/W	R/W	R/W	–	R/W
Сброс	–	0	0	0	–	0
	–	C3IF_OVRM	C3IF_FLLM	C3IF_EMPM	–	F0IF_OVRM

Номер	4	3	2…0
Доступ	RO	RO	–
Сброс	0	0	–
	F0IF_FLLM	F0IF_EMPM	–

Таблица 58 – Описание бит регистра ADCUI_F0MASK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...25	–	Должнен быть 0 для правильного функционирования

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
24	C3IF_OVRM	Маска бита C3IF_OVR: 0 – прерывание запрещено; 1 – прерывание разрешено
23	C3IF_FLLM	Маска бита C3IF_FLL: 0 – прерывание запрещено; 1 – прерывание разрешено
22	C3IF_EMPM	Маска бита C3IF_EMP: 0 – прерывание запрещено; 1 – прерывание разрешено
21…6	–	Должнен быть 0 для правильного функционирования
5	F0IF_OVRM	Маска бита F0IF_OVR: 0 – прерывание запрещено; 1 – прерывание разрешено
4	F0IF_FLLM	Маска бита F0IF_FLL: 0 – прерывание запрещено; 1 – прерывание разрешено
3	F0IF_EMPM	Маска бита F0IF_EMP: 0 – прерывание запрещено; 1 – прерывание разрешено
2…0	–	Должнен быть 0 для правильного функционирования

13.1.9 **ADCUI_F1CTR**

Таблица 59 – Регистр ADCUI_F1CTR

Номер	31…8	7, 6	5…1	0
Доступ	–	R/W	–	R/W
Сброс	–	0	–	0
	–	F1IGAIN	–	F1INTEN

Таблица 60 – Описание бит регистра ADCUI_F1CTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	–	Зарезервировано
7, 6	F1IGAIN	Предусилитель в канале АЦП I1: 00 – 6 дБ (x2); 01 – 12 дБ (x4); 10 – 18 дБ (x8); 11 – 24 дБ (x16)
5…1	–	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
0	F1INTEN	Отключение интегратора в канале АПЦ I1: 0 – интегратор включен; 1 – интегратор отключен. Корректная работа интегратора гарантируется только при частоте дискретизации 4 или 8 кГц – при условии, что значение CHPF_COEFF обеспечивает частоту среза фильтров HPF на уровне 1 Гц для выбранной частоты дискретизации. На частоте дискретизации 16 кГц корректная работа интегратора не гарантируется. Например: – при частоте дискретизации 4 кГц значение CHPF_COEFF должно быть 0x803; – при частоте дискретизации 8 кГц значение CHPF_COEFF должно быть 0x802 (значение по умолчанию). Примечание – Для обычного применения, где функционал с аппаратным сдвигом фазы в канале не требуется, интегратор рекомендуется отключать

13.1.10 **ADCUI_F1IDAT**

Таблица 61 – Регистр ADCUI_F1IDAT

Номер	31…24	23…0
Доступ	–	RO
Сброс	–	000000h
	–	F1IDAT

Таблица 62 – 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 АЦП канала I1. FIFO включается при установке I1EN

13.1.11 **ADCUI_F1STAT**

Таблица 63 – Регистр ADCUI_F1STAT

Номер	31...6	5	4	3	2…0
Доступ	–	R/W	RO	RO	–
Сброс	–	0	0	0	–
	–	F1IF_OVR	F1IF_FLL	F1IF_EMP	–

Таблица 64 – Описание бит регистра ADCUI_F1STAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...6	–	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5	F1IF_OVR	Флаг, что произошло переполнение FIFO F1IDAT. Запись «1» сбрасывает этот флаг
4	F1IF_FLL	Флаг, что FIFO F1IDAT заполнено
3	F1IF_EMP	Флаг, что FIFO F1IDAT пусто
2…0	–	Зарезервировано

13.1.12 **ADCUI_F1MASK**

Таблица 65 – Регистр ADCUI_F1MASK

Номер	31...6	5	4	3	2…0
Доступ	–	R/W	R/W	R/W	–
Сброс	–	0	0	0	–
	–	F1IF_OVRM	F1IF_FLLM	F1IF_EMPM	–

Таблица 66 – Описание бит регистра ADCUI_F1MASK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...6	–	Должнен быть 0 для правильного функционирования
5	F1IF_OVRM	Маска бита F1IF_OVR: 0 – прерывание запрещено; 1 – прерывание разрешено
4	F1IF_FLLM	Маска бита F1IF_FLL: 0 – прерывание запрещено; 1 – прерывание разрешено
3	F1IF_EMPM	Маска бита F1IF_EMP: 0 – прерывание запрещено; 1 – прерывание разрешено
2…0	–	Должнен быть 0 для правильного функционирования

13.1.13 **ADCUI_F2CTR**

Таблица 67 – Регистр ADCUI_F2CTR

Номер	31…8	7, 6	5…1	0
Доступ	–	R/W	–	R/W
Сброс	–	0	–	0
	–	F2IGAIN	–	F2INTEN

Таблица 68 – Описание бит регистра ADCUI_F2CTR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	–	Зарезервировано
7, 6	F2IGAIN	Предусилитель в канале АПЦ I2: 00 – 6 дБ (x2); 01 – 12 дБ (x4); 10 – 18 дБ (x8); 11 – 24 дБ (x16)

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5…1	–	Зарезервировано
0	F2INTEN	Отключение интегратора в канале АПЦ I2: 0 – интегратор включен; 1 – интегратор отключен. Корректная работа интегратора гарантируется только при частоте дискретизации 4 или 8 кГц – при условии, что значение CHPF_COEFF обеспечивает частоту среза фильтров HPF на уровне 1 Гц для выбранной частоты дискретизации. На частоте дискретизации 16 кГц корректная работа интегратора не гарантируется. Например: – при частоте дискретизации 4 кГц значение CHPF_COEFF должно быть 0x803; – при частоте дискретизации 8 кГц значение CHPF_COEFF должно быть 0x802 (значение по умолчанию). Примечание – Для обычного применения, где функционал с аппаратным сдвигом фазы в канале не требуется, интегратор рекомендуется отключать

13.1.14 **ADCUI_F2IDAT**

Таблица 69 – Регистр ADCUI_F2IDAT

Номер	31…24	23…0
Доступ	–	R/W
Сброс	–	000000h
	–	F2IDAT

Таблица 70 – 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 АЦП канала I2. FIFO включается при установке I2EN

13.1.15 **ADCUI_F2STAT**

Таблица 71 – Регистр ADCUI_F2STAT

Номер	31…6	5	4	3	2…0
Доступ	–	R/W	RO	RO	–
Сброс	–	0	0	0	–
	–	F2IF_OVR	F2IF_FLL	F2IF_EMP	–

Таблица 72 – Описание бит регистра ADCUI_F2STAT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…6	–	Зарезервировано
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…0	–	Зарезервировано

13.1.16 **ADCUI_F2MASK**

Таблица 73 – Регистр ADCUI_F2MASK

Номер	31…6	5	4	3	2…0
Доступ	–	R/W	R/W	R/W	–
Сброс	–	0	0	0	–
	–	F2IF_OVRM	F2IF_FLLM	F2IF_EMPM	–

Таблица 74 – Описание бит регистра ADCUI_F2MASK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...6	–	Должнен быть 0 для правильного функционирования
5	F2IF_OVRM	Маска бита F2IF_OVR: 0 – прерывание запрещено; 1 – прерывание разрешено
4	F2IF_FLLM	Маска бита F2IF_FLL: 0 – прерывание запрещено; 1 – прерывание разрешено
3	F2IF_EMPM	Маска бита F2IF_EMP: 0 – прерывание запрещено; 1 – прерывание разрешено
2…0	–	Должнен быть 0 для правильного функционирования

13.1.17 **ADCUI_CCAL1**

Таблица 75 – Регистр ADCUI_CCAL1

Номер	31…16	15…0
Доступ	R/W	–
Сброс	0000	–
	I0BGAIN	–

Таблица 76 – Описание бит регистра ADCUI_CCAL1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	I0BGAIN	Калибровочный коэффициент канала АЦП I0
15...0	–	Зарезервировано

13.1.18 **ADCUI_CCAL2**

Таблица 77 – Регистр ADCUI_CCAL2

Номер	31…16	15…0
Доступ	R/W	–
Сброс	0000	–
	I1BGAIN	–

Таблица 78 – Описание бит регистра ADCUI_CCAL2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	I1BGAIN	Калибровочный коэффициент канала АЦП I1
15...0	–	Зарезервировано

13.1.19 **ADCUI_CCAL3**

Таблица 79 – Регистр ADCUI_CCAL3

Номер	31…16	15…0
Доступ	R/W	–
Сброс	0000	–
	I2BGAIN	–

Таблица 80 – Описание бит регистра ADCUI_CCAL3

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	I2BGAIN	Калибровочный коэффициент канала АЦП I2
15...0	–	Зарезервировано

13.1.20 **ADCUI_CCAL4**

Таблица 81 – Регистр ADCUI_CCAL4

Номер	31…16	15…0
Доступ	–	R/W
Сброс	–	0000
	–	I3BGAIN

Таблица 82 – Описание бит регистра ADCUI_CCAL4

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

13.1.21 **ADCUI_MSC**

Таблица 83 – Регистр ADCUI_MSC

Номер	31…29	28	27	26	25…14	13…7	6…0
Доступ	–	R/W	R/W	R/W	–	R/W	R/W
Сброс	–	0	0	0	–	00	00
	–	F2ISEL_HPF	F1ISEL_HPF	F0ISEL_HPF	–	DMA_EN_RQ_S	DMA_EN_RQ_B

Таблица 84 – Описание бит регистра ADCUI_MSC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…29	–	Зарезервировано
28	F2ISEL_HPF	В FIFO F2IDAT записываются данные: 0 – взятые после фильтра HPF; 1 – взятые до фильтра HPF
27	F1ISEL_HPF	В FIFO F1IDAT записываются данные: 0 – взятые после фильтра HPF 1 – взятые до фильтра HPF
26	F0ISEL_HPF	В FIFO F0IDAT записываются данные: 0 – взятые после фильтра HPF; 1 – взятые до фильтра HPF
25…14	–	Зарезервировано
13…7	DMA_EN_RQ_S	Разрешение формирования запросов к DMA (одиночные транзакции) от каналов 7...1 ΔΣ АЦП. Запрос формируется, если FIFO содержит 1 или более отсчетов
6…0	DMA_EN_RQ_B	Разрешение формирования запросов к DMA (блочные транзакции) от каналов 7...1 ΔΣ АЦП. Запрос формируется, если FIFO содержит 4 или более отсчетов

13.1.22 **ADCUI_FILTERCFG**

Таблица 85 – Регистр ADCUI_FILTERCFG

Номер	31...26	25…24	23…12	11…0
Доступ	–	R/W	–	R/W
Сброс	–	0	–	802h
	–	CHP_ADC_CFG	–	CHPF_COEFF

Таблица 86 – Описание бит регистра ADCUI_FILTERCFG

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…26	–	Зарезервировано
25…24	CHP_ADC_CFG	Настройка периода чоппера АЦП всех каналов: 0 – 256 периодов синхросигнала АЦП (рекомендуется при OSR = 256); 1 – 512 периодов синхросигнала АЦП (рекомендуется при OSR = 512); 2 – 1024 периода синхросигнала АЦП (рекомендуется при OSR = 1024)
23…12	–	Зарезервировано
11…0	CHPF_COEFF	Коэффициент для фильтров HPF во всех каналах АЦП, определяющий частоту среза. Значение по сбросу «-2046», что соответствует частоте среза 0,5 Гц по уровню «-3дБ» при частоте дискретизации 4 кГц. Рекомендованное значение «-2016». Значение "-2048" (частота среза 0 Гц) использовать не допускается. Формула расчета $$CHPF_COEFF = \frac{2^{11} \times \left(\tan\left(\pi \times \left(\frac{freq_hpf}{samplerate_adc}\right)\right) - 1\right)}{\tan\left(\pi \times \left(\frac{freq_hpf}{samplerate_adc}\right)\right) + 1},$$ где freq_hpf – необходимая частота среза фильтра, samplerate_adc – частота дискретизации АЦП. Результат необходимо округлить до целого числа со знаком. Код для вычисления в САПР: `CHPF_COEFF=max(-2047, round(2^11*(tan(pi*(freq_hpf/samplerate_adc))-1)/(tan(pi*(freq_hpf/samplerate_adc))+1)));`

13.1.23 **ADCUI_CLKPHASE1**

Таблица 87 – Регистр ADCUI_CLKPHASE1

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

Таблица 88 – Описание бит регистра ADCUI_CLKPHASE1

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…30	CLC_ADC_CFG	Настройка частоты тактирования АЦП для всех каналов: 00 – 4 МГц; 01 – 2 МГц; 10 – 1 МГц; 11 – зарезервировано. Эта частота также выводится на PD14 в режиме использования внешних дельта-сигма модуляторов (при настройке вывода в функцию CLK_SD_EXT)
29…20	I2PHASE[9:0]	Фазовый сдвиг канала АЦП I2. От 0 · (такт АЦП) до +1023 · (такт АЦП). «1000000000» соответствует дополнительной задержке на 512 тактов АЦП. При использовании DCHPENI2 = 1 сдвигать фазу рекомендуется с шагом 32
19…10	I1PHASE[9:0]	Фазовый сдвиг канала АЦП I1. От 0 · (такт АЦП) до +1023 · (такт АЦП). «1000000000» соответствует дополнительной задержке на 512 тактов АЦП. При использовании DCHPENI1 = 1 сдвигать фазу рекомендуется с шагом 32
9…0	I0PHASE[9:0]	Фазовый сдвиг канала АЦП I0. От 0 · (такт АЦП) до +1023 · (такт АЦП). «1000000000» соответствует дополнительной задержке на 512 тактов АЦП. При использовании DCHPENI0 = 1 сдвигать фазу рекомендуется с шагом 32

13.1.24 **ADCUI_CLKPHASE2**

Таблица 89 – Регистр ADCUI_CLKPHASE2

Номер	31…26	9…0
Доступ	–	R/W
Сброс	–	200h
	–	I3PHASE

Таблица 90 – Описание бит регистра ADCUI_CLKPHASE2

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…26	–	Зарезервировано
9...0	I3PHASE[9:0]	Фазовый сдвиг канала АЦП I3. От 0 · (такт АЦП) до +1023 · (такт АЦП). «1000000000» соответствует дополнительной задержке на 512 тактов АЦП. При использовании DCHPENI3 = 1 сдвигать фазу рекомендуется с шагом 32

14 Соответствие отчетов АЦП внешним сигналам

На рисунках 21, 22 приведены два вида включения АЦП: полностью дифференциальное и недифференциальное включение.

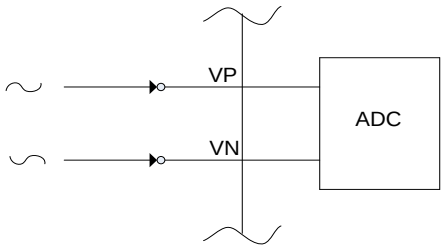


Рисунок 21 – Дифференциальное включение

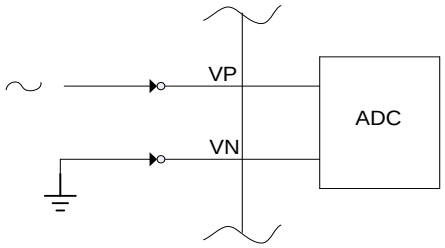


Рисунок 22 – Недифференциальное включение

В каналах АЦП после децемирующих фильтров по умолчанию включен высокочастотный фильтр HPF. Он убирает постоянную составляющую сигнала. Частота среза фильтра по умолчанию равна 1 Гц и может быть настроена с помощью CHPF_COEFF. Отключается фильтр с помощью FxISEL_HPF.

Значения отсчетов из FIFO АЦП можно перевести в напряжения на входе по формулам, приведенным в таблице 91. Отсчеты, записанные в FIFO, представлены в двоичном формате с дополнением до 2.

Таблица 91 – Формулы расчета входного напряжения

Режим	Входное напряжение АЦП, В
24-битный	$\frac{FxIDAT \cdot UREFO}{FxIyGAIN \cdot 2^{24}}$
16-битный	$\frac{FxIDAT \cdot UREFO}{FxIyGAIN \cdot 2^{16}}$

Каждый из каналов АЦП может быть откалиброван с помощью коэффициента IxBGAIN в соответствии с формулой (2). Значение IxBGAIN записывается в двоичном формате с дополнением до 2.

$$FxIDAT_{cor} = FxIDAT_{ADC} \cdot \left(1 + \frac{IxBGAIN}{2^{15}}\right). \quad (2)$$

Для коррекции фазового сдвига в системе присутствует конфигурируемая линия задержки как показано на рисунке 23.

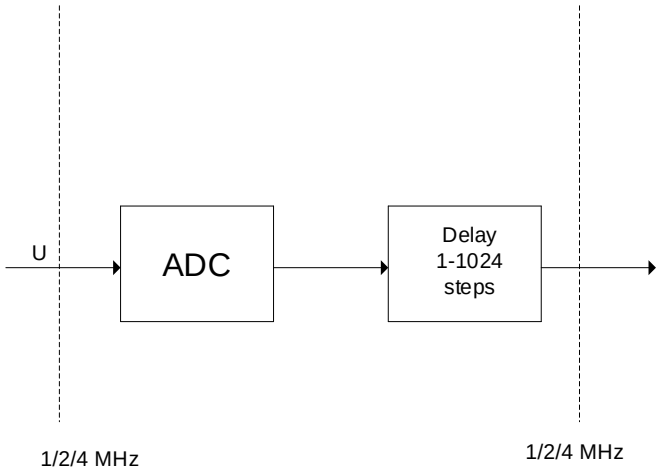


Рисунок 23 – Контролируемый фазовый сдвиг в каналах АЦП

Изменяя линию задержки в канале АЦП с помощью IxPHASE, можно регулировать временной сдвиг одного канала относительно другого. Так как частота отсчетов после модулятора равна 1/2/4 МГц, то один шаг при частоте входного сигнала, например, 50 Гц, равен 0,018 °/0,009 °/0,0045 °. Необходимо иметь в виду, что в этой системе сдвиг осуществляется во временной области, поэтому фазовый сдвиг в градусах зависит от частоты входного сигнала.

В качестве децимирующего фильтра используется фильтр со структурой SINC3, его характеристики приведены на рисунках 24, 25.

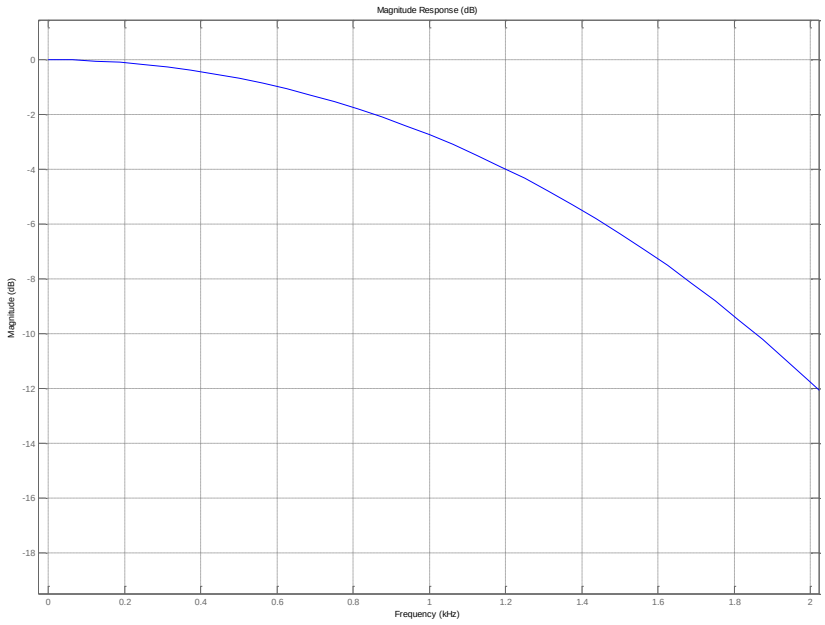


Рисунок 24 – Характеристика децимирующего фильтра в полосе 2 кГц

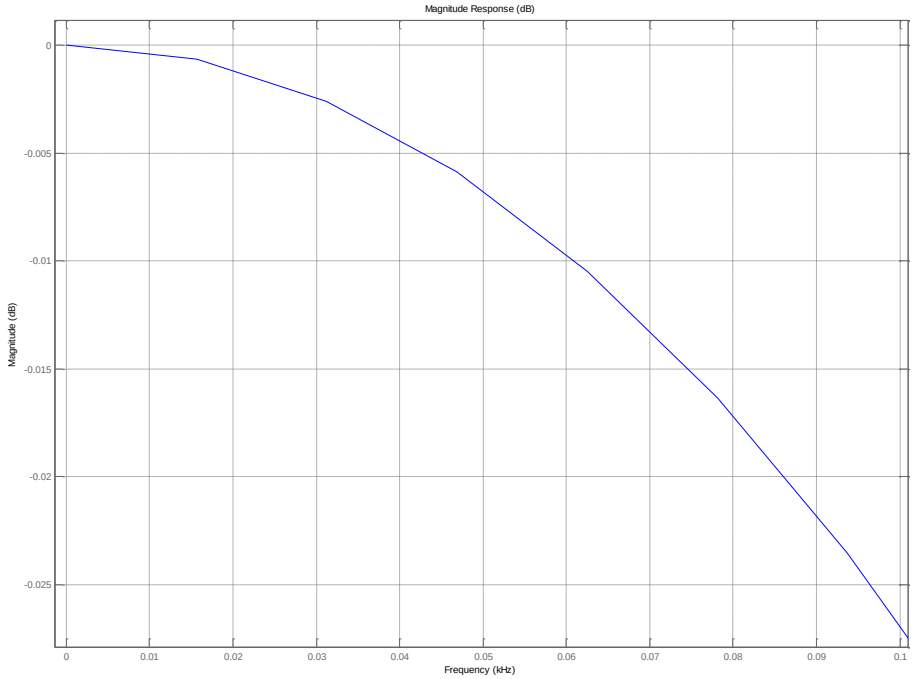


Рисунок 25 – Характеристика децимирующего фильтра в полосе 100 Гц

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

15 Аппаратный блок вычисления CRC

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

15.1 Описание регистров управления блока CRC

Таблица 92 – Описание регистров управления блока CRC

Базовый Адрес	Название	Описание
0x5009_8000	CRC	Контроллер CRC
Смещение		
0x00	CRC_CTRL	Общее управление для контроллера CRC
0x04	CRC_STAT	Статус CRC блока
0x08	CRC_DATAI	Регистр FIFO входных данных
0x0C	CRC_VAL	Регистр подсчитанного CRC
0x10	CRC_POL	Полином для расчета CRC

15.1.1 CRC_CTRL

Таблица 93 – 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 94 – Описание бит регистра CRC_CTRL

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

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

15.1.2 **CRC_STAT**

Таблица 95 – 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 96 – 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 крайне мере одну заполненную ячейку; 1 – FIFO пусто
1	FIFOFULL	FIFO заполнено: 0 – FIFO имеет по крайне мере одну свободную ячейку; 1 – FIFO не имеет свободных ячеек
0	CONVCOMP	Завершение расчета CRC: 0 – расчет идет; 1 – расчет слова данных завершен или блок отключен
* Сброс бита происходит записью «1» в разряд [3]		

15.1.3 **CRC_DATAI**

Таблица 97 – Регистр CRC_DATAI

Номер	31…0
Доступ	W
Сброс	
	DATA_IN

Таблица 98 – Описание бит регистра CRC_DATAI

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

15.1.4 **CRC_VAL**

Таблица 99 – Регистр CRC_VAL

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

Таблица 100 – Описание бит регистра CRC_VAL

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

15.1.5 **CRC_POL**

Таблица 101 – Регистр CRC_POL

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

Таблица 102 – Описание бит регистра CRC_POL

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

Результирующий полином:

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

16 Сигналы тактовой частоты

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

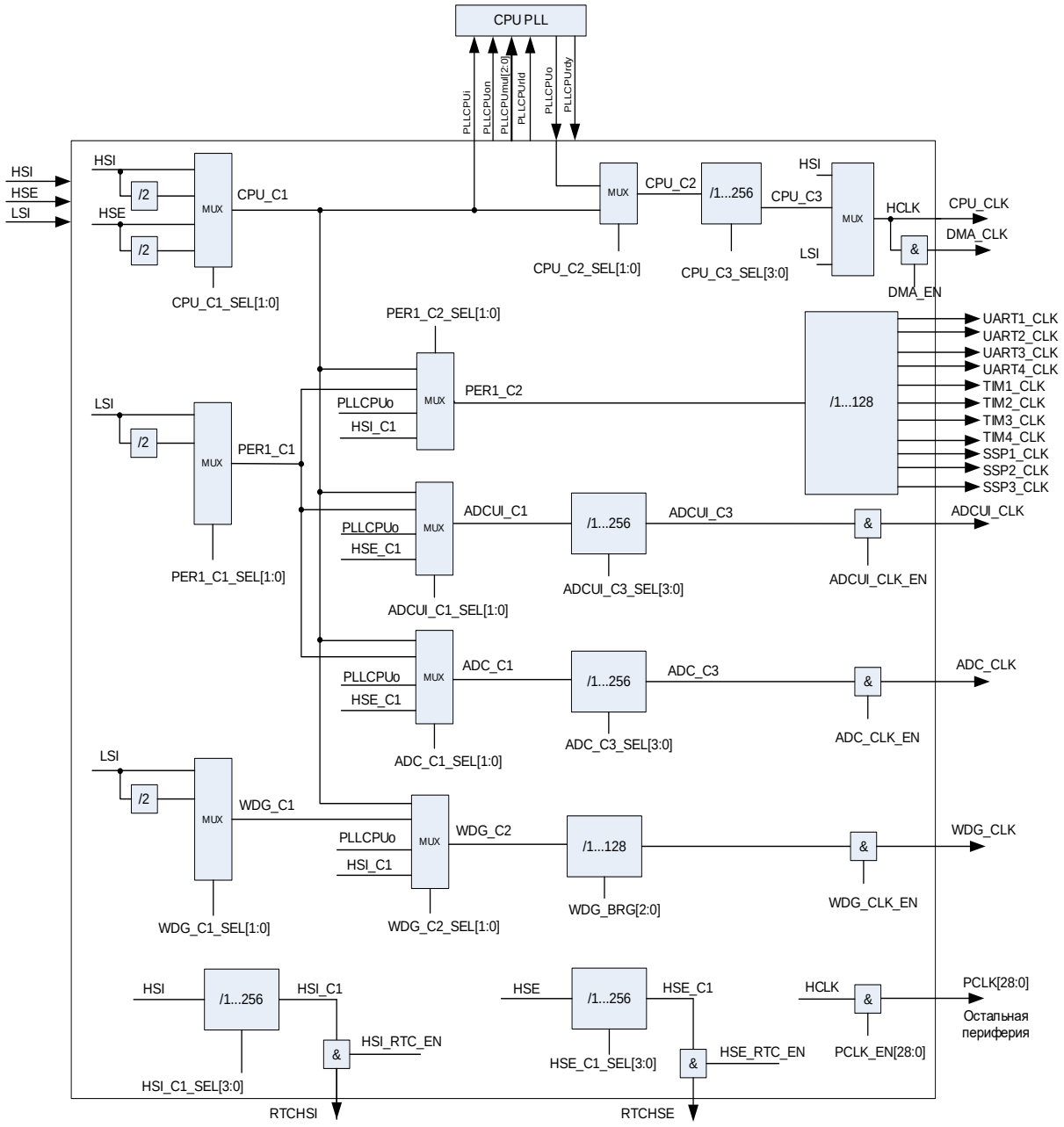


Рисунок 26 – Структурная блок-схема формирования тактовой частоты

Управление тактовыми частотами ведется через периферийный блок RST_CLK. При включении питания микросхема запускается на частоте HSI/16 генератора. Выдача тактовых сигналов синхронизации для всех периферийных блоков кроме RST_CLK отключена. Для начала работы с нужным периферийным блоком необходимо включить его тактовую частоту в регистре PER2_CLOCK. Некоторые контроллеры интерфейсов (UART, SSP, Таймеры) могут работать на частотах отличных от частоты процессорного ядра, поэтому в соответствующих регистрах (PER1_CLOCK, UART_CLOCK, SSP_CLOCK, TIM_CLOCK) могут быть заданы их скорости работы. Для изменения

тактовой частоты ядра можно перейти на другой генератор и/или воспользоваться блоком умножения тактовой частоты. Для корректной смены тактовой частоты сначала должны быть сформированы необходимые тактовые частоты и затем осуществлено переключение на них на соответствующих мультиплексорах управляемом регистре CPU_CLOCK. Регистры блока контроллера тактовой частоты не сбрасываются внутренним сигналом сброса ndmreset, сформированным отладчиком.

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

При смене тактовой частоты процессорного ядра следует настроить число тактов паузы для доступа к Flash-памяти, задаваемое полем Delay в регистре FLASH_CMD контроллера Flash-памяти FLASH_CNTRL. Если настраиваемая частота процессорного ядра выше текущей, то данную настройку нужно проводить до смены частоты, в противном случае – после её смены.

16.1 Встроенный RC-Генератор HSI

Генератор HSI вырабатывает тактовую частоту 8 МГц. Генератор автоматически запускается при появлении питания U_{CC} и при выходе в нормальный режим работы вырабатывает сигнал HSI_RDY в регистре батарейного домена BKP_CLK. Первоначально процессорное ядро запускается на тактовой частоте HSI/16. При дальнейшей работе генератор HSI может быть отключен при помощи сигнала HSI_ON в регистре BKP_CLK. Также генератор может быть подстроен при помощи сигнала HSI_LSI_TRIM в регистре BKP_CLK.

16.2 Встроенный RC-генератор LSI

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

Также генератор может быть подстроен при помощи сигнала HSI_LSI_TRIM в регистре BKP_CLK. Частота генератора LSI равна HSI/256.

16.3 Внешний осциллятор HSE

Осциллятор HSE предназначен для выработки тактовой частоты f_{C_HSE} в диапазоне 8…16 МГц с помощью внешнего резонатора. Рекомендуемая суммарная нагрузочная емкость на выводе микросхемы (паразитная емкость платы + номинал припаиваемой емкости) – 24 и 16 пФ для резонаторов на 8 и 16 МГц соответственно. Осциллятор запускается при появлении питания U_{CC} и сигнала разрешения HSEON в регистре HS_CONTROL. Когда значение дифференциальной амплитуды на резонаторе достигнет уровня, минимально необходимого для появления тактовой частоты на выходе блока HSE внутри микросхемы, появляется сигнал HSERDY в регистре CLOCK_STATUS. Появление бита HSERDY не означает полное завершение переходных процессов, связанных с амплитудой на резонаторе. А частота/скважность на выходе блока HSE может немного отличаться от номинальной. Для большинства применений

этого достаточно. Время окончательного выхода на номинальный режим работы генератора зависит в первую очередь от характеристик кварцевого резонатора и его можно оценить по формуле

$$t_{start} = \mathrm{K} \cdot \frac{Q \cdot R_M}{2 \cdot \pi \cdot quartz_freg \cdot \left(\frac{4 \cdot g_m \cdot C_L^2}{(g_m \cdot C_s')^2 + 16 \cdot \omega^2 \cdot (C_L^2 + C_s' \cdot C_L)^2} - R_M \right)}. \quad (3)$$

Код для вычисления в САПР:

*tstart = K * ((Q * Rm) / (2 * pi * quartz_freq * ((4 * gm * (CL)^2) / ((gm*Cs)^2 + 16* (2 * pi * quartz_freq)^2 * ((CL)^2 + Cs * CL)^2) - Rm))*

Расчет для типового кварцевого резонатора KX-7T от Geyer на 8 МГц, устанавливаемого в рамках отладочных плат разработки компании Миландр, с характеристиками Rm = 250 Ом (500 Ом максимально, для расчёта берется среднее значение), Cs = 3,5 пФ, Cl = 24 пФ дает приблизительное время старта t_{start} = 5,7 мс (добротность Q взята равной 75000).

Параметры и способ их получения, необходимые для расчета по формуле (3), приведены в таблице 104.

Таблица 103 – Параметры для расчета времени старта t_{start} по формуле (3)

Параметр формулы	Буквенное обозначение	Способ получения значения параметра
Коэффицент пропорциональности	K	12
Добротность кварцевого резонатора	Q	Если нет данных в спецификации на резонатор, то принять равным 50000 – 100000
Динамическое сопротивление кварцевого резонатора	R_M	Из спецификации резонатора
Частота настройки кварцевого резонатора	quartz_freq	Из спецификации резонатора
Крутизна усиливающего транзистора	g_m	8,24 мА/В
Нагрузочная емкость кварцевого резонатора	C_L	Исходя из номиналов конденсаторов, установленных на плату
Шунтирующая емкость кварцевого резонатора	C_s'	$C_s' \approx C_0$ где C_0 – из спецификации на кварцевый резонатор
Циклическая частота	ω	$\omega = 2 \cdot \pi \cdot quartz_freg$

Осциллятор также может работать в режиме HSEBYP, когда входная тактовая частота с входа OSC_IN проходит напрямую на выход HSE. Выход OSC_OUT находится в этом режиме в третьем состоянии. Допустимый диапазон частот f_{C_HSE} в режиме HSEBYP – от 0 до 48 МГц.

16.4 Встроенный блок умножения системной тактовой частоты

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 8, задаваемый полем PLLCPUMUL[2:0] в регистре PLL_CONTROL. Входная частота блока умножителя должна быть в диапазоне 8…16 МГц, выходная до 50 МГц. При выходе блока умножителя тактовой частоты в расчетный режим вырабатывается сигнал PLLCPURDY в регистре CLOCK_STATUS. Блок включается с помощью сигнала PLLCPUON в регистре PLL_CONTROL. Выходная частота может быть использована как основная частота процессора и периферии.

16.5 Описание регистров блока контроллера тактовой частоты

Таблица 104 – Описание регистров блока контроллера тактовой частоты

Базовый Адрес	Название	Описание
0x5002_0000	RST_CLK	Контроллер тактовой частоты
Смещение		
0x00	CLOCK_STATUS	Регистр состояния блока управления тактовой частотой
0x04	PLL_CONTROL	Регистр управления блоками умножения частоты
0x08	HS_CONTROL	Регистр управления высокочастотным генератором и осциллятором
0x0C	CPU_CLOCK	Регистр управления тактовой частотой процессорного ядра
0x10	PER1_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x14	ADC_CLOCK	Регистр управления тактовой частотой АЦП и ΔΣ АЦП
0x18	RTC_CLOCK	Регистр управления формированием высокочастотных тактовых сигналов блока RTC
0x1C	PER2_CLOCK	Регистр управления тактовой частотой периферийных блоков
0x20	DMA_DONE_STICK	Регистр фиксации сигналов окончания работы DMA
0x24	TIM_CLOCK	Регистр управления тактовой частотой TIMER
0x28	UART_CLOCK	Регистр управления тактовой частотой UART
0x2C	SSP_CLOCK	Регистр управления тактовой частотой SSP
0x30	DIV_SYS_TIM	Регистр выбора делителя частоты для системного таймера ядра

16.5.1 CLOCK_STATUS

Таблица 105 – 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 106 – Описание бит регистра CLOCK_STATUS

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

16.5.2 **PLL_CONTROL**

Таблица 107 – Регистр PLL_CONTROL

Номер	31…11	10…8	7…4	3	2	1	0
Доступ	U	R/W	U	R/W	R/W	U	U
Сброс	0	000	0000	0	0	0	0
	-	PLL_CPU_MUL[2:0]	-	PLL_CPU_SEL	PLL_CPU_ON	-	-

Таблица 108 – Описание бит регистра PLL_CONTROL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...11	-	Зарезервировано
10...8	PLL_CPU_MUL[2:0]	Коэффициент умножения для CPU PLL: PLLCPUo = PLLCPUi x (PLLCPUMUL+1)
7...4	-	Зарезервировано
3	PLL_CPU_SEL	Бит перезапуска выбора входной частоты PLL: 0 – частота с генератора HSE; 1 – частота задаётся с помощью бит CPU_C1_SEL[1:0] регистра CPU_CLOCK
2	PLL_CPU_ON	Бит включения PLL: 0 – PLL выключена; 1 – PLL включена
1	-	Зарезервировано
0	-	Зарезервировано

16.5.3 **HS_CONTROL**

Таблица 109 – Регистр HS_CONTROL

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

Таблица 110 – Описание битов регистра HS_CONTROL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	-	Зарезервировано
1	HSEBYP	Бит управления HSE осциллятором: 0 – режим осциллятора; 1 – режим внешнего генератора. Для работы в режиме внешнего генератора (bypass) бит hseon должен быть равен 1
0	HSEON	Бит управления HSE осциллятором: 0 – выключен; 1 – включен

16.5.4 **CPU_CLOCK**

Таблица 111 – 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	01	1011	0	0	00
	-	HCLK_SEL[1:0]	CPU_C3_SEL[3:0]	-	CPU_C2_SEL	CPU_C1_SEL[1:0]

Таблица 112 – 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 – зарезервировано; 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

16.5.5 **PER1_CLOCK**

Таблица 113 – Регистр PER1_CLOCK

Номер	31…14	13	12…10	9, 8
Доступ	U	R/W	R/W	R/W
Сброс	0	0	000	00
	-	WDG_CLK_EN	WDG_BRG_[2:0]	WDG_C2_SEL[1:0]

Номер	7, 6	5	4	3, 2	1, 0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	00	1	0	00	00
	WDG_C1_SEL[1:0]	DMA_EN	KeyResetProg	PER1_C2_SEL	PER1_C1_SEL[1:0]

Таблица 114 – Описание бит регистра PER1_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…14	-	Зарезервировано
13	WDG_CLK_EN	Разрешение тактовой частоты на WDG: 0 – нет частоты; 1 – есть частота
12…10	WDG_BRG_[2:0]	Делитель тактовой частоты WDG: 000 – WDG_CLK == WDG_C2; 001 – WDG_CLK == WDG_C2/2; 010 – WDG_CLK == WDG_C2/4; … 111 – WDG_CLK == WDG_C2/128
9, 8	WDG_C2_SEL	Биты выбора источника для WDG_C2: 00 – CPU_C1; 01 – WDG_C1; 10 – PLLCPUo; 11 – HSI_C1
7, 6	WDG_C1_SEL[1:0]	Биты выбора источника для WDG_C1: 00 – LSI; 01 – LSI/2; 10 – недопустимо; 11 – недопустимо
5	DMA_EN	Бит разрешения тактирования DMA контроллера
4	KeyResetProg	Бит сброса памяти криптографических ключей
3, 2	PER1_C2_SEL	Биты выбора источника для PER1_C2: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSI_C1
1, 0	PER1_C1_SEL[1:0]	Биты выбора источника для PER1_C1: 00 – LSI; 01 – LSI/2; 10 – недопустимо; 11 – недопустимо

16.5.6 **ADC_CLOCK**

Таблица 115 – Регистр ADC_CLOCK

Номер	31…14	13	12	11…8
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0000
	-	ADC_CLK_EN	ADCUI_CLK_EN	ADC_C3_SEL[3:0]

Номер	7…4	3, 2	1, 0
Доступ	R/W	R/W	R/W
Сброс	0000	00	00
	ADCUI_C3_SEL[3:0]	ADCUI_C1_SEL[1:0]	ADC_C1_SEL[1:0]

Таблица 116 – 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 – HSE_C1 В качестве частоты синхронизации блока ADCUI рекомендуется выбирать источник HSE = 8,192 МГц без дополнительных делителей
1, 0	ADC_C1_SEL[1:0]	Биты выбора источника для ADC_C1: 00 – CPU_C1; 01 – PER1_C1; 10 – PLLCPUo; 11 – HSE_C1

16.5.7 **RTC_CLOCK**

Таблица 117 – Регистр RTC_CLOCK

Номер	31…10	9	8	7…4	3…0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0000	0000
	-	HSI_RTC_EN	HSE_RTC_EN	HSI_SEL[1:0]	HSE_SEL[1:0]

Таблица 118 – Описание бит регистра RTC_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…10	-	Зарезервировано
9	HSI_RTC_EN	Бит разрешения HSI RTC: 0 – запрещен; 1 – разрешен
8	HSE_RTC_EN	Бит разрешения HSE RTC: 0 – запрещен; 1 – разрешен
7…4	HSI_SEL[3:0]	Биты выбора делителя для HSI_C1: 0xxx – HSI_C1 = HSI; 1000 – HSI_C1 = HSI / 2; 1001 – HSI_C1 = HSI / 4; 1010 – HSI_C1 = HSI / 8; … 1111 – HSI_C1 = HSI / 256
3…0	HSE_SEL[3:0]	Биты выбора делителя для HSE_C1: 0xxx – HSE_C1 = HSE; 1000 – HSE_C1 = HSE / 2; 1001 – HSE_C1 = HSE / 4; 1010 – HSE_C1 = HSE / 8; … 1111 – HSE_C1 = HSE / 256

16.5.8 **PER2_CLOCK**

Таблица 119 – Регистр PER2_CLOCK

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

Таблица 120 – Описание бит регистра PER2_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
30…0	PCLK_EN[29:0]	Биты разрешения тактирования периферийных блоков: 0 – запрещено; 1 – разрешено. PCLK[0] – SSP1 PCLK[1] – UART1 PCLK[2] – UART2 PCLK[3] – FLASH PCLK[4] – RST_CLK PCLK[5] – DMA PCLK[6] – I2C PCLK[7] – UART3 PCLK[8] – ADC PCLK[9] – WWDG PCLK[10] – IWDG PCLK[11] – POWER PCLK[12] – BKP PCLK[13] – ADCUI PCLK[14] – TIMER1 PCLK[15] – TIMER2 PCLK[16] – PORTA PCLK[17] – PORTB PCLK[18] – PORTC PCLK[19] – CRC PCLK[20] – зарезервировано PCLK[21] – зарезервировано PCLK[22] – зарезервировано PCLK[23] – RANDOM PCLK[24] – ISO7816 PCLK[25] – SSP2 PCLK[26] – SSP3 PCLK[27] – TIMER3 PCLK[28] – TIMER4 PCLK[29] – UART4 PCLK[30] – PORTD

16.5.9 **DMA_DONE_STICK**

Таблица 121 – Регистр DMA_DONE_STICK

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

Таблица 122 – Описание бит регистра DMA_DONE_STICK

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

16.5.10 **TIM_CLOCK**

Таблица 123 – Регистр TIM_CLOCK

Номер	27	26	25	24	23…19	18…16	15…8	7…0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	00000	000	00000000	00000000
	TIM4_CLK_EN	TIM3_CLK_EN	TIM2_CLK_EN	TIM1_CLK_EN	TIM4_BRG[4:0]	TIM3_BRG[2:0]	TIM2_BRG[7:0]	TIM1_BRG[7:0]

Таблица 124 – Описание бит регистра TIM_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…28	-	Зарезервировано
27	TIM4_CLK_EN	Разрешение тактовой частоты на TIM4: 0 – нет частоты; 1 – есть частота
26	TIM3_CLK_EN	Разрешение тактовой частоты на TIM3: 0 – нет частоты; 1 – есть частота
25	TIM2_CLK_EN	Разрешение тактовой частоты на TIM2: 0 – нет частоты; 1 – есть частота
24	TIM1_CLK_EN	Разрешение тактовой частоты на TIM1: 0 – нет частоты; 1 – есть частота

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
23…19	TIM_BRG[4:0]*	Делитель тактовой частоты TIM4: xx000 – TIM4_CLK == PER1_C2; xx001 – TIM4_CLK == PER1_C2/2; xx010 – TIM4_CLK == PER1_C2/4; … xx111 – TIM4_CLK == PER1_C2/128
18…16	TIM3_BRG[2:0]*	Делитель тактовой частоты TIM3: 000 – TIM3_CLK == PER1_C2; 001 – TIM3_CLK == PER1_C2/2; 010 – TIM3_CLK == PER1_C2/4; … 111 – TIM3_CLK == PER1_C2/128
15…8	TIM2_BRG[7:0]*	Делитель тактовой частоты TIM2: xxxxx000 – TIM2_CLK == PER1_C2; xxxxx001 – TIM2_CLK == PER1_C2/2; xxxxx010 – TIM2_CLK == PER1_C2/4; … xxxxx111 – TIM2_CLK == PER1_C2/128
7…0	TIM1_BRG[7:0]*	Делитель тактовой частоты TIM1: xxxxx000 – TIM1_CLK == PER1_C2; xxxxx001 – TIM1_CLK == PER1_C2/2; xxxxx010 – TIM1_CLK == PER1_C2/4; … xxxxx111 – TIM1_CLK == PER1_C2/128
* Частота тактирования таймера TIMx_CLK должна быть менее или равна частоте тактирования ядра CPU_CLK		

16.5.11 **UART_CLOCK**

Таблица 125 – Регистр UART_CLOCK

Номер	30	29..27	26	25	24	23…16	15…8	7…0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	000	0	0	0	00000000	00000000	00000000
	UART4_CLK_EN	UART4_BRG[2:0]	UART3_CLK_EN	UART2_CLK_EN	UART1_CLK_EN	UART3_BRG[7:0]	UART2_BRG[7:0]	UART1_BRG[7:0]

Таблица 126 – Описание бит регистра UART_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	-	Зарезервировано
30	UART4_CLK_EN	Разрешение тактовой частоты на UART4: 0 – нет частоты; 1 – есть частота

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
29…27	UART4_BRG[2:0]	Делитель тактовой частоты UART4: 000 – UART4_CLK == PER1_C2; 001 – UART4_CLK == PER1_C2/2; 010 – UART4_CLK == PER1_C2/4; … 111 – UART4_CLK == PER1_C2/128
26	UART3_CLK_EN	Разрешение тактовой частоты на UART3: 0 – нет частоты; 1 – есть частота
25	UART2_CLK_EN	Разрешение тактовой частоты на UART2: 0 – нет частоты; 1 – есть частота
24	UART1_CLK_EN	Разрешение тактовой частоты на UART1: 0 – нет частоты; 1 – есть частота
23…16	UART3_BRG[7:0]	Делитель тактовой частоты UART3: xxxxx000 – UART3_CLK == PER1_C2; xxxxx001 – UART3_CLK == PER1_C2/2; xxxxx010 – UART3_CLK == PER1_C2/4; … xxxxx111 – UART3_CLK == PER1_C2/128
15…8	UART2_BRG[7:0]	Делитель тактовой частоты UART2: xxxxx000 – UART2_CLK == PER1_C2; xxxxx001 – UART2_CLK == PER1_C2/2; xxxxx010 – UART2_CLK == PER1_C2/4; … xxxxx111 – UART2_CLK == PER1_C2/128
7…0	UART1_BRG[7:0]	Делитель тактовой частоты UART: xxxxx000 – UART1_CLK == PER1_C2; xxxxx001 – UART1_CLK == PER1_C2/2; xxxxx010 – UART1_CLK == PER1_C2/4; … xxxxx111 – UART1_CLK == PER1_C2/128

16.5.12 **SSP_CLOCK**

Таблица 127 – Регистр SSP_CLOCK

Номер	26	25	24	23…16	15…8	7…0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	00000000	00000000	00000000
	SSP3_CLK_EN	SSP2_CLK_EN	SSP1_CLK_EN	SSP3_BRG[7:0]	SSP2_BRG[7:0]	SSP1_BRG[7:0]

Таблица 128 – Описание бит регистра SSP_CLOCK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…27	-	Зарезервировано
26	SSP3_CLK_EN	Разрешение тактовой частоты на SSP3: 0 – нет частоты; 1 – есть частота
25	SSP2_CLK_EN	Разрешение тактовой частоты на SSP2: 0 – нет частоты; 1 – есть частота
24	SSP1_CLK_EN	Разрешение тактовой частоты на SSP1: 0 – нет частоты; 1 – есть частота
23…16	SSP3_BRG[7:0]	Делитель тактовой частоты SSP3: xxxxx000 – SSP3_CLK == PER1_C2; xxxxx001 – SSP3_CLK == PER1_C2/2; xxxxx010 – SSP3_CLK == PER1_C2/4; … xxxxx111 – SSP3_CLK == PER1_C2/128
15…8	SSP2_BRG[7:0]	Делитель тактовой частоты SSP2: xxxxx000 – SSP2_CLK == PER1_C2; xxxxx001 – SSP2_CLK == PER1_C2/2; xxxxx010 – SSP2_CLK == PER1_C2/4; … xxxxx111 – SSP2_CLK == PER1_C2/128
7…0	SSP1_BRG[7:0]	Делитель тактовой частоты SSP1: xxxxx000 – SSP1_CLK == PER1_C2; xxxxx001 – SSP1_CLK == PER1_C2/2; xxxxx010 – SSP1_CLK == PER1_C2/4; … xxxxx111 – SSP1_CLK == PER1_C2/128

16.5.13 **DIV_SYS_TIM**

Таблица 129 – Регистр DIV_SYS_TIM

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

Таблица 130 – Описание бит регистра DIV_SYS_TIM

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

17 Батарейный домен и часы реального времени

Блок батарейного домена предназначен для обеспечения функций календаря и часов реального времени, сохранения некоторого набора пользовательских данных при отключении основного источника питания. При снижении питания U_{CC} регулятор напряжения батарейного домена выполняет автоматическое переключение питания батарейного домена с U_{CC} на U_{CCB}. Если на U_{CCB} имеется отдельный источник питания (батарейка), то батарейный домен остается включенным и может выполнять свои функции.

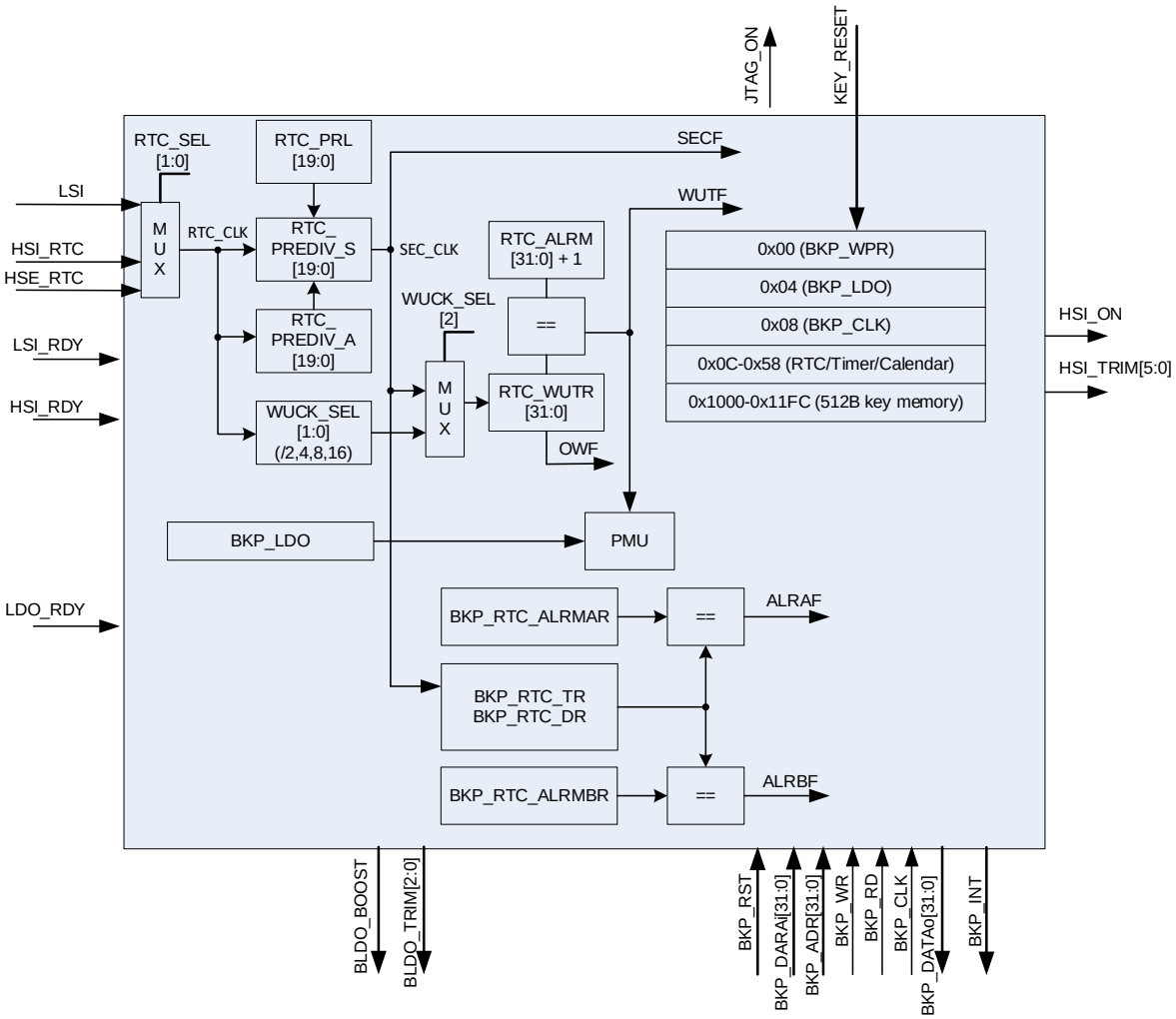


Рисунок 27 – Структурная блок-схема батарейного домена и часов реального времени

17.1 Часы реального времени

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

цифрового ядра, формируемого встроенным регулятором напряжения. Частота LSI доступна только при наличии питания U_{CC}. Выбор между источниками осуществляется битами RTC_SEL. При отключении основного источника питания U_{CC} часы останавливаются.

Для отсчета секунд в часах реального времени применяется 20-битный предварительный делитель входной тактовой частоты RTC_CLK, на выходе которого формируется тактовый сигнал SEC_CLK. Регистр BKP_RTC_PREDIV_S выступает в качестве счетчика предварительного делителя, который тактируется на частоте RTC_CLK. Регистр BKP_RTC_PRL задает коэффициент деления предварительного делителя, при этом счетчик BKP_RTC_PREDIV_S инкрементируется в интервале от 0 до BKP_RTC_PRL. Коэффициент деления в регистре BKP_RTC_PRL должен быть задан таким образом, чтобы частота SEC_CLK составляла 1 Гц. Счетчик BKP_RTC_PREDIV_S после достижения значения BKP_RTC_PRL устанавливается в 0, при этом установка флага SECF в регистре BKP_RTC_CS и изменение регистров BKP_RTC_TR, BKP_RTC_DR и BKP_RTC_WUTR (при тактировании таймера BKP_RTC_WUT от SEC_CLK) выполняются через 1 такт частоты RTC_CLK.

Частота SEC_CLK при разрешенной работе часов реального времени формируется на выходе блока BKP как тактовый сигнал RTCCLK_1Hz, при этом для вывода сигнала RTCCLK_1Hz значение регистра BKP_RTC_PRL, задающего коэффициент деления, должно быть отлично от нуля. Длительность импульса высокого уровня сигнала RTCCLK_1Hz составляет 1 период частоты RTC_CLK.

Для калибровки тактовой частоты RTC_CLK используются биты RTC_CAL[7:0] (см. описание ошибки 0013 в документе MDR12065 Errata Notice). Значение RTC_CAL[7:0] определяет, какое число тактов RTC_CLK из 2^{20} будет замаскировано с помощью дополнительного счетчика BKP_RTC_PREDIV_A. Данный способ калибровки заключается в том, что один период BKP_RTC_PREDIV_S, который приходится на начало интервала 2^{20}, удлиняется на RTC_CAL[7:0] тактов частоты RTC_CLK, или на RTC_CAL[7:0]/RTC_CLK секунд. Таким образом, с помощью бит RTC_CAL[7:0] производится замедление хода часов, при этом единица замедления равна $1/2^{20}$, или ~0,95 ppm. Для ускорения хода часов необходимо задать меньшее, чем требуется, значение регистра BKP_RTC_PRL, а затем произвести замедление с помощью бит RTC_CAL[7:0]. Изменение значения бит RTC_CAL[7:0] может быть осуществлено в ходе работы часов реального времени.

Основу модуля часов реального времени составляет двоично-десятичный таймер/счетчик. Результат счета отображается в двух 32-разрядных регистрах. Первый из них BKP_RTC_TR содержит информацию о времени, второй – BKP_RTC_DR представляет собой календарь, включающий год, месяц, день недели и дату. Все данные представлены в BCD формате, что позволяет их сразу же использовать для отображения на различных индикаторах. Модуль календаря поддерживает автоматическое определение високосных лет, а также количества дней в текущем месяце. Запись в

регистры BKP_RTC_TR и BKP_RTC_DR выполняется, только когда часы реального времени выключены (RTC_EN = 0).

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

Сторожевой таймер BKP_RTC_WUT может работать либо на частоте SEC_CLK, которая используется для работы счетчика/календаря, либо на независимой частоте, задаваемой с помощью бит WUCK_SEL[2:0]. Регистр BKP_RTC_ALRM предназначен для задания времени сторожевого таймера, при превышении которого устанавливается флаг прерывания WUTF.

Процесс выполнения записи регистров BKP_RTC_WUTR, BKP_RTC_PREDIV_S, BKP_RTC_DR и BKP_RTC_TR отображается с помощью флага WEC в регистре BKP_RTC_CS. На время выполнения записи флаг WEC устанавливается в «1», по окончании записи флаг WEC сбрасывается в «0». Запись в один из данных регистров может быть выполнена только когда флаг WEC равен «0». Пока флаг WEC установлен в «1», новая запись в данные регистры выполнена не будет.

Запись в регистры BKP_RTC_PRL, BKP_RTC_ALRM и поле RTC_CAL[7:0] регистра BKP_RTC_CR выполняется за 2 такта частоты RTC_CLK. Если после записи данных регистров выполняется запись регистра BKP_RTC_WUTR, BKP_RTC_PREDIV_S, BKP_RTC_DR или BKP_RTC_TR, то окончание записи всех регистров можно отследить по флагу WEC в регистре BKP_RTC_CS.

17.2 Память регистров и криптографический ключей

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

17.3 Описание регистров блока батарейного домена

Таблица 131 – Описание регистров блока батарейного домена

Базовый Адрес	Название	Описание
0x5006_0000	BKP	Контроллер батарейного домена и часов реального времени
Смещение		
0x00	BKP_WPR	Регистр ключа для разрешения работы с регистрами блока.
0x04	BKP_LDO	Регистр управления блоком LDO
0x08	BKP_CLK	Регистр управления блоками HSI, LSI
0x0C	BKP_RTC_CR	Регистр управления блоком RTC
0x10	BKP_RTC_WUTR	Регистр счетчика сторожевого таймера
0x14	BKP_RTC_PREDIV_S	Регистр счетчика предварительного делителя счетчика/календаря
0x18	BKP_RTC_PRL	Регистр коэффициента деления предварительного делителя счетчика/календаря

Базовый Адрес	Название	Описание
0x1C	BKP_RTC_ALRM	Регистр значения для сравнения со значением счетчика сторожевого таймера BKP_RTC_WUTR и выработки сигнала WUTF
0x20	BKP_RTC_CS	Регистр управления и состояния флагов часов реального времени
0x24	BKP_RTC_TR	Регистр времени счетчика/календаря в BCD формате
0x28	BKP_RTC_DR	Регистр даты счетчика/календаря в BCD формате
0x2C	BKP_RTC_ALRMAR	Регистр задания события A – будильника с календарным временем
0x30	BKP_RTC_ALRMBR	Регистр задания события B – будильника с календарным временем
0x34-0xFFF	–	Зарезервировано
0x1000-0x11FC	BKP_MEM	Память ОЗУ 512 байт

17.3.1 **BKP_MEM (128x32)**

Таблица 132 – Память BKP_MEM

Номер	31…0
Доступ	R/W
Сброс	X
	BKP_MEM[31:0]

Таблица 133 – Описание бит регистра BKP_MEM

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

17.3.2 **BKP_WPR**

Таблица 134 – Память BKP_WPR

Номер	31…0
Доступ	R/W
Сброс	0
	BKP_WPR[31:0]

Таблица 135 – Описание бит регистра BKP_WPR

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

17.3.3 **BKP_LDO**

Таблица 136 – Регистр BKP_LDO

Номер	31…29	28, 27	26	25	24…7	6
Доступ	U	R/W	U	R/W	U	R/W
Сброс	0	00	0	0	0	1
	–	MODE[1:0]	–	FPOR	–	JTAG_ON

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

Таблица 137 – Описание бит регистра BKP_LDO

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...29	–	Зарезервировано
28, 27	MODE[1:0]	При использовании загрузочной программы, описанной в разделе 9: режим запуска микросхемы, полученный с входов MODE[1:0] (PC[0], PC[1]). При использовании загрузочной программы, отличной от описанной в разделе 9, данное поле может применяться по усмотрению пользователя
26	–	Зарезервировано
25	FPOR	При использовании загрузочной программы, описанной в разделе 9: флаг срабатывания POR. При сбросе по питанию устанавливается в 0. Может использоваться расположенной в области памяти BOOT загрузочной программой, описанной в разделе 9, как программный флаг для определения типа сброса. Пример алгоритма анализа бита FPOR: – FPOR = 0: идет выполнение программы после сброса по питанию, установить FPOR в 1; – FPOR = 1: идет выполнение программы после системного сброса, не изменять состояние FPOR. При использовании загрузочной программы, отличной от описанной в разделе 9, данное поле может применяться по усмотрению пользователя
24...7	–	Зарезервировано
6	JTAG_ON	Разрешение работы порта JTAG: 0 – запрещен; 1 – разрешен
5	–	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
4, 3	LDO_BOOST[1:0]	Биты активации boost режима работы основного регулятора: 00 – типовое значение выхода регулятора; 01 – увеличение напряжения регулятора; 10 – уменьшение напряжения регулятора; 11 – уменьшение напряжения регулятора. Рекомендуется устанавливать только прописанные в TRIM_LDO значения
2	BLDO_BOOST	Вход активации boost режима работы регулятора батарейного домена: 0 – типовое значение выхода регулятора; 1 – увеличение выходного значения регулятора напряжения с помощью BLDO_TRIM[1:0]
1, 0	BLDO_TRIM[1:0]	Подстройка регулятора питания батарейного домена: 00 – типовое значение; 01…11 – увеличение опорного напряжения

17.3.4 **BKP_CLK**

Таблица 138 – Регистр BKP_CLK

Номер	31…21	20…15	14	13	12	11…6	5	4…0
Доступ	U	R/W	RO	R/W	RO	U	R/W	U
Сброс	0	000000	1	1	0	0	0	0
	–	HSI_LSI_TRIM[5:0]	HSI_RDY	HSI_ON	LSI_RDY	–	LSI_ON	–

Таблица 139 – Описание бит регистра BKP_CLK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..21	-	Зарезервировано
20...15	HSI_LSI_TRIM[5:0]	Коэффициент подстройки частот генераторов HSI и LSI. Рекомендуется использовать значение бит TRIM_HSI, записанных во Flash-памяти при производстве
14	HSI_RDY	Флаг выхода генератора HSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
13	HSI_ON	Бит управления генератором HSI: 1 – генератор включен; 0 – генератор выключен. Аппаратно устанавливается в 1 только при сбросе по включению батарейного питания
12	LSI_RDY	Флаг выхода генератора LSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
11…6	–	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5	LSI_ON	Бит управления генератором LSI: 0 – генератор выключен; 1 – генератор включен
4…0	–	Зарезервировано

17.3.5 **BKP_RTC_CR**

Таблица 140 – Регистр BKP_RTC_CR

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

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

Таблица 141 – Описание бит регистра BKP_RTC_CR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…26	-	Зарезервировано
25	FMT	Формат времени: 0 – 24 часа в сутках; 1 – формат времени AM/PM
24…15	-	Зарезервировано
14	WUCK_SEL[2]	Биты выбора независимой синхронизации для сторожевого таймера: WUCK_SEL[2] = 1 – общее тактирование с счетчиком/ календарем; WUCK_SEL[2] = 0 – независимое тактирование с выбранного делителя
13...12	WUCK_SEL[1:0]	Биты выбора делителя частоты для сторожевого таймера: WUCK_SEL[1:0] = 3 – делитель на 2 WUCK_SEL[1:0] = 2 – делитель на 4 WUCK_SEL[1:0] = 1 – делитель на 8 WUCK_SEL[1:0] = 0 – делитель на 16
11	RTC_RESET	Сброс часов реального времени: 0 – часы не сбрасываются; 1 – часы сбрасываются Выполняет сброс RTC, WUT, TAMPER и TIMESTAMP, а также сброс регистров BKP_RTC_WUTR, BKP_RTC_PREDIV_S, BKP_RTC_DR, BKP_RTC_TR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
10...3	RTC_CAL[7:0]	Коэффициент подстройки тактовой частоты RTC_CLK. Из каждых 2^{20} тактов RTC_CLK будет замаскировано RTC_CAL тактов: 00000000 – 0 тактов; 00000001 – 1 такт; …. 11111111 – 255 тактов. Таким образом, если исходная частота RTC_CLK равна 32768,00000 Гц, то после подстройки средняя частота тактирования составляет: RTC_CLK_C = 32768,00000 Гц при RTC_CAL = 0; RTC_CLK_C = 32767,96875 Гц при RTC_CAL = 1; … RTC_CLK_C = 32760,03125 Гц при RTC_CAL = 255. В общем случае средняя частота тактирования RTC_CLK после подстройки составляет RTC_CLK_C = RTC_CLK • (1 – RTC_CAL[7:0] / 2^{20}). Единица подстройки (замедления) равна $1/2^{20}$, или ~0,95 ppm. Подстроенная частота RTC_CLK_C используется только для формирования частоты SEC_CLK
2	RTC_EN	Бит разрешения работы часов реального времени: 0 – работа запрещена; 1 – работа разрешена. Разрешает работу RTC, WUT и TIMESTAMP
1, 0	RTC_SEL[1:0]	Биты выбора источника тактовой синхронизации часов реального времени: 00 – LSI; 01 – зарезервировано; 10 – HSI_RTC; 11 – HSE_RTC Для переключения частоты RTC_CLK с одного источника на другой необходимо, чтобы оба источника частоты были включены и находились в рабочем режиме. Исключением является первое переключение с LSI на другой источник после сброса по питанию – в этом случае включение частоты LSI не нужно. Валидная частота RTC_CLK формируется после 3 периодов частоты, с которой происходит переключение, плюс 3 периода частоты, на которую происходит переключение. Использование частоты RTC_CLK возможно только после появления валидной частоты. До формирования валидной частоты RTC_CLK работа RTC должна быть запрещена или же RTC должен находиться в сбросе

17.3.6 **BKP_RTC_WUTR**

Таблица 142 – Регистр BKP_RTC_WUTR

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

Таблица 143 – Описание бит регистра BKP_RTC_WUTR

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

17.3.7 **BKP_RTC_PREDIV_S**

Таблица 144 – Регистр BKP_RTC_PREDIV_S

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

Таблица 145 – Описание бит регистра BKP_RTC_PREDIV_S

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

17.3.8 **BKP_RTC_PRL**

Таблица 146 – Регистр BKP_RTC_PRL

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

Таблица 147 – Описание бит регистра BKP_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 для формирования частоты SEC_CLK: SEC_CLK = RTC_CLK / (RTC_PRL[19:0] + 1), где RTC_CLK – частота тактирования с учетом подстройки RTC_CAL[7:0]

17.3.9 **BKP_RTC_ALRM**

Таблица 148 – Регистр BKP_RTC_ALRM

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

Таблица 149 – Описание бит регистра BKP_RTC_ALRM

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	RTC_ALRM[31:0]	Значения для сравнения со счетчиком сторожевого таймера BKP_RTC_WUTR и выработки сигнала WUTF. Сигнал WUTF вырабатывается в момент превышения счетчиком BKP_RTC_WUTR значения в регистре BKP_RTC_ALRM: RTC_WUTR[31:0] == RTC_ALRM[31:0] + 1

17.3.10 **BKP_RTC_CS**

Таблица 150 – Регистр BKP_RTC_CS

Номер	31…13	12	11	10	9	8	7
Доступ	U	R/W	R/W	R/C1	R/C1	R/W	R/W
Сброс	0	0	0	0	0	0	0
	-	ALRBF_IE	ALRAF_IE	ALRBF	ALRAF	ALRB_EN	ALRA_EN

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

Таблица 151 – Описание бит регистра BKP_RTC_CS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…13	-	Зарезервировано
12	ALRBF_IE	Разрешение прерывания по событию ALRBF: 0 – запрещено; 1 – разрешено
11	ALRAF_IE	Разрешение прерывания по событию ALRAF: 0 – запрещено; 1 – разрешено
10	ALRBF	Флаг установки времени события B. Сброс флага осуществляется записью единицы
9	ALRAF	Флаг установки времени события A. Сброс флага осуществляется записью единицы

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8	ALRB_EN	Разрешение события B. Совпадение BKP_RTC_ALRMBR и текущего календарного времени: 0 – запрещено; 1 – разрешено
7	ALRA_EN	Разрешение события A. Совпадение BKP_RTC_ALRMAR и текущего календарного времени: 0 – запрещено; 1 – разрешено
6	WEC	Флаг выполнения записи в регистры RTC: 0 – можно записывать в один из регистров BKP_RTC_WUTR, BKP_RTC_PREDIV_S, BKP_RTC_DR, BKP_RTC_TR; 1 – идет запись в один из регистров BKP_RTC_WUTR, BKP_RTC_PREDIV_S, BKP_RTC_DR, BKP_RTC_TR, запись в регистры запрещена. После записи каждого из регистров BKP_RTC_WUTR, BKP_RTC_PREDIV_S, BKP_RTC_DR, BKP_RTC_TR необходимо дождаться окончания записи
5	WUTF_IE	Разрешение прерывания по событию WUTF: 0 – запрещено; 1 – разрешено
4	SECF_IE	Разрешение прерывания по событию SECF: 0 – запрещено; 1 – разрешено
3	OWF_IE	Разрешение прерывания по событию OWF: 0 – запрет прерывания по переполнению; 1 – разрешение прерывания по переполнению
2	WUTF	Флаг превышения счетчиком сторожевого таймера BKP_RTC_WUTR значения BKP_RTC_ALRM: 0 – нет события превышения; 1 – было событие превышения. Флаг устанавливается при RTC_WUTR[31:0] == RTC_ALRM[31:0] + 1. Сброс флага осуществляется записью единицы
1	SECF	Флаг совпадения значений регистров BKP_RTC_PREDIV_S и BKP_RTC_PRL: 0 – не было совпадения; 1 – было совпадение. По событиям совпадения BKP_RTC_PREDIV_S и BKP_RTC_PRL формируется частота SEC_CLK. Таким образом флаг SECF устанавливается на каждом такте частоты SEC_CLK. Сброс флага осуществляется записью единицы
0	OWF	Флаг переполнения сторожевого таймера BKP_RTC_WUT: 0 – нет переполнения; 1 – было переполнение. Сброс флага осуществляется записью единицы

17.3.11 **BKP_RTC_TR**

Таблица 152 – Регистр BKP_RTC_TR

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

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

Таблица 153 – Описание бит регистра BKP_RTC_TR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…23	-	Зарезервировано
22	PM	AM/PM значение: 0 – AM или 24-часовой формат; 1 – PM
21, 20	HT[1:0]	Десятки часов в BCD формате
19..16	HU[3:0]	Единицы часов в BCD формате
15	-	Зарезервировано
14..12	MNT[2:0]	Десятки минут в BCD формате
11..8	MNU[3:0]	Единицы минут в BCD формате
7	-	Зарезервировано
6..4	ST[2:0]	Десятки секунд в BCD формате
3..0	SU[3:0]	Единицы секунд в BCD формате

17.3.12 **BKP_RTC_DR**

Таблица 154 – Регистр BKP_RTC_DR

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

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

Таблица 155 – Описание бит регистра BKP_RTC_DR

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

17.3.13 **BKP_RTC_ALRMAR, BKP_RTC_ALRMBR**

Таблица 156 – Регистры BKP_RTC_ALRMAR, BKP_RTC_ALRMBR

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

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

Таблица 157 – Описание бит регистров BKP_RTC_ALRMAR, BKP_RTC_ALRMBR

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	MSK4	Маска даты: 0 – дата учитывается; 1 – дата не учитывается
30	WDSEL	Выбор дня недели: 0 – используется дата DU[3:0], DT[1:0]; 1 – DU[3:0] определяет день недели, DT[1:0] не используется
29, 28	DT[1:0]	Десятки даты в BCD формате
27...24	DU[3:0]	Единицы даты в BCD формате
23	MSK3	Маска часов: 0 – часы учитываются; 1 – часы не учитываются
22	PM	AM/PM значение: 0 – AM или 24-часовой формат; 1 – PM

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
21, 20	HT[1:0]	Десятки часов в BCD формате
19…16	HU[3:0]	Единицы часов в BCD формате
15	MSK2	Маска минут: 0 – минуты учитываются; 1 – минуты не учитываются
14…12	MNT[2:0]	Десятки минут в BCD формате
11…8	MNU[3:0]	Единицы минут в BCD формате
7	MSK1	Маска секунд: 0 – секунды учитываются; 1 – секунды не учитываются
6…4	ST[2:0]	Десятки секунд в BCD формате
3…0	SU[3:0]	Единицы секунд в BCD формате

18 Порты ввода-вывода

Микросхема имеет четыре порта ввода-вывода (см. таблицу 158). Порт A – 16-разрядный, порт B – 15-разрядный, порт C – 8-разрядный, порт D – 16-разрядный и их выводы мультиплексируются между различными функциональными блоками, управление для каждого вывода порта отдельное. Для того, чтобы выводы порта перешли под управление того или иного периферийного блока, необходимо задать для нужных выводов выполняемую функцию и настройки.

Таблица 158 – Порты ввода-вывода

Вывод	Аналоговая функция MODE=00 ANALOG_EN=0 OE=0		Цифровая функция						
			Порт IO MODE=00 ANALOG_EN=0/1 OE=0/1	Основная MODE=01 ANALOG_EN=0/1 OE=X		Альтернативная MODE=10 ANALOG_EN=0/1 OE=X		Переопределённая MODE=11 ANALOG_EN=0/1 OE=X	
Порт A									
PA0	-		PA0	TMR1_CH1	1	SSP2_FSS	9	-	
PA1	-		PA1	TMR1_CH1N		SSP2_CLK		-	
PA2	-		PA2	TMR1_CH2		SSP2_RXD		-	
PA3	-		PA3	TMR1_CH2N		SSP2_TXD		-	
PA6*	-		PA6/TCK	TMR1_CH4		-		-	
PA7*	-		PA7/TDO	TMR1_CH4N		-		-	
PA8*	-		PA8/TMS	TMR1_ETR		-		-	
PA9*	-		PA9/TDI	TMR1_BRK		-		-	
Порт B									
PB0	-		PB0	UART1_TXD	4	UART7816_TXD	11	TMR3_CH1	17
PB1	-		PB1	UART1_RXD		UART7816_RXD		TMR3_CH1N	
PB2	-		PB2	nSIROUT1		UART7816_CLK		TMR3_CH2	
PB3	-		PB3	nSIRIN1		UART7816_CTS		TMR3_CH2N	
PB5	-		PB5	nUART1RTS		UART7816_RTS	12	TMR3_CH3N	
PB10	-		PB10	TMR2_CH2	5	UART4_RXD	20	-	
PB11	-		PB11	TMR2_CH2N		UART4_TXD		-	
PB13	-		PB13	TMR2_CH3N		UART2_TXD	6	-	
PB14	-		PB14	TMR2_CH4		UART2_RXD		-	
Порт C									
PC0	-		PC0/MODE0	TMR4_ETR	19	-		TMR4_CH4	18
PC1	-	3	PC1/MODE1	TMR4_BRK		-		TMR4_CH4N	
PC2	-		PC2	TMR2_CH1	7	SSP3_FSS	14	I2C1_SCL	15
PC3	-		PC3	TMR2_CH1N		SSP3_CLK		I2C1_SDA	
PC4	-		PC4	EXT_INT2		SSP3_RXD		-	
PC5	-		PC5	EXT_INT3		SSP3_TXD		-	
Порт D									
PD2	-		PD2	SSP2_RXD		SD_V2_EXT		-	
PD4	-		PD4	UART2_TXD	6	-		-	
PD5	-		PD5	UART2_RXD		-		-	
PD6	-		PD6	UART3_RXD	10	-		-	
PD7	-		PD7	UART3_TXD		-		-	
PD8	-		PD8	UART4_RXD	20	-		-	
PD10	-		PD10	UART7816_TXD	11	SD_I0_EXT		-	
PD11	-		PD11	UART7816_RXD		SD_I1_EXT		-	
PD12	-		PD12	UART7816_CLK		SD_I2_EXT		-	
PD13	-		PD13	UART7816_CTS		SD_I3_EXT		-	

Для работы вывода в аналоговой функции необходимо установить соответствующие поля регистров в значения: MODEx[1:0] = 00, ANALOG_EN[x] = 0, OE[x] = 0. Для работы вывода в цифровой функции порта IO необходимо задать режим работы приемника в регистре PORT_ANALOG, режим работы передатчика – в регистре PORT_OE. Для работы вывода в цифровой функции, кроме порта IO, необходимо задать режим работы приемника в регистре PORT_ANALOG, режим работы передатчика задается автоматически в соответствии с заданной функцией.

18.1 Описание регистров портов ввода-вывода

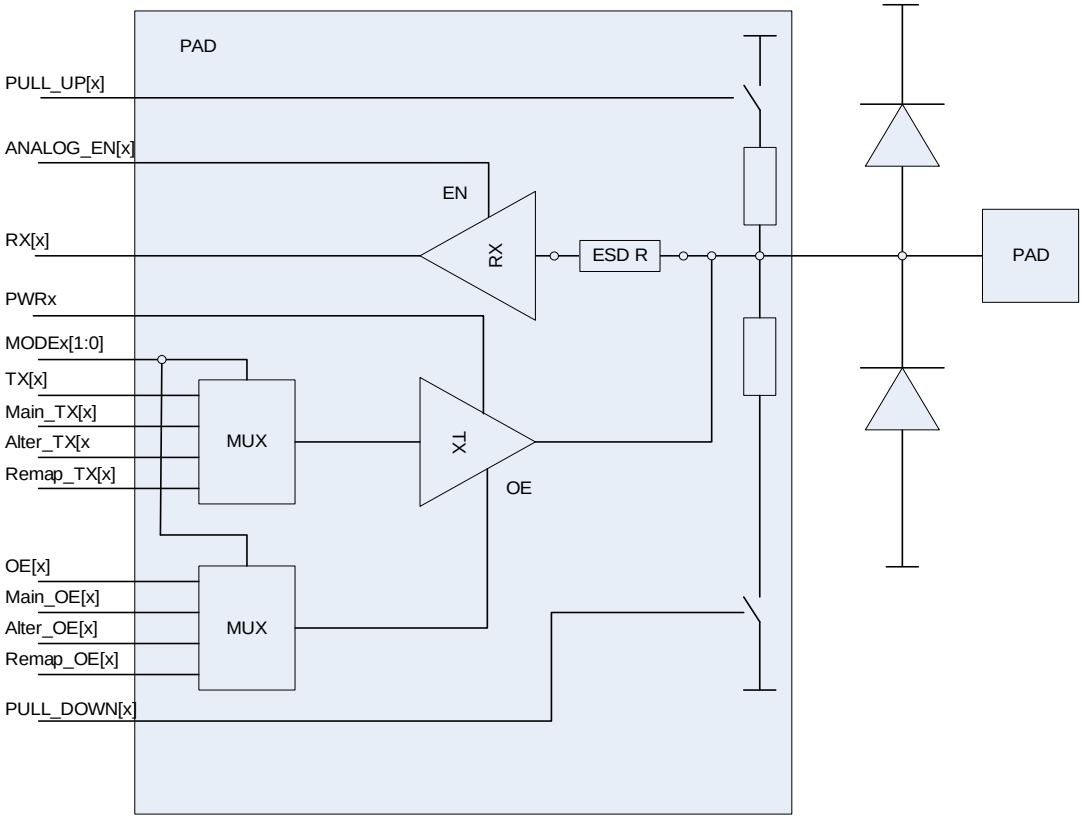


Рисунок 28 – Блок-схема разряда порта ввода-вывода

Таблица 159 – Описание регистров портов ввода-вывода

Базовый Адрес	Название	Описание
0x5008_0000	PORTA	Контроллер порта A
0x5008_8000	PORTB	Контроллер порта B
0x5009_0000	PORTC	Контроллер порта C
0x500F_0000	PORTD	Контроллер порта D
Смещение		
0x00	PORT_RXTX[15:0]	Регистр данных порта
0x04	PORT_OE[15:0]	Регистр режима работы передатчика порта
0x08	PORT_FUNC[31:0]	Регистр цифровой функции порта
0x0C	PORT_ANALOG[15:0]	Регистр режима работы приемника порта
0x10	PORT_PULL[31:0]	Регистр подтяжки порта
0x14	-	Зарезервировано
0x18	PORT_PWR[31:0]	Регистр настройки мощности передатчика порта

Базовый Адрес	Название	Описание
0x1C	-	Зарезервировано
0x20	PORT_SETTX[15:0]	Регистр установки выходных данных передатчика порта
0x24	PORT_CLRTX[15:0]	Регистр сброса выходных данных передатчика порта
0x28	PORT_RDTX[15:0]	Регистр чтения выходных данных передатчика порта

18.1.1 **PORT_RXTX**

Таблица 160 – Регистр PORT_RXTX

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

Таблица 161 – Описание бит регистра PORT_RXTX

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15…0	RXTX[15:0]	Регистр прямого доступа к данным порта. При чтении отображаются данные приемника порта: – ANALOG_EN=0: «0»; – ANALOG_EN=1: состояние на выводе порта. При записи осуществляется запись в регистр выходных данных передатчика порта

Примечание – При разрешенной работе порта JTAG бит 6 регистра PORT_RXTX на порту A всегда равен «0».

18.1.2 **PORT_OE**

Таблица 162 – Регистр PORT_OE

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

Таблица 163 – Описание бит регистра PORT_OE

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…16	-	Зарезервировано
15…0	OE[15:0]	Режим работы передатчика порта: 0 – передатчик выключен (аналоговый режим); 1 – передатчик работает на выход (цифровой режим)

18.1.3 **PORT_FUNC**

Таблица 164 – Регистр PORT_FUNC

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

Таблица 165 – Описание бит регистра PORT_FUNC

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2	MODEx[1:0]	Аналогично MODE0[1:0] для остальных битов порта
1…0	MODE0[1:0]	Функция, выполняемая выводом порта: 00 – порт IO; 01 – основная функция; 10 – альтернативная функция; 11 – переопределенная функция. При выборе функции MODEx[1:0] != 00 управление передатчиком (режим работы и выходное значение) задается автоматически в соответствии с выбранной функцией

18.1.4 **PORT_ANALOG**

Таблица 166 – Регистр PORT_ANALOG

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

Таблица 167 – Описание бит регистра PORT_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 приемника порта: 0 – приемник выключен (аналоговый режим), приемник всегда выдает значение «0». В данном режиме допускается подача на вывод порта произвольного напряжения (от 0 до U_{CC}), потребление микросхемы при этом не увеличивается. Управление передатчиком в данном режиме остается активным; 1 – приемник включен (цифровой режим), приемник выдает состояние на выводе порта в соответствии с входными уровнями U_{IL} и U_{IH} (таблица 438). Дополнительное потребление микросхемы в зависимости от поданного напряжения на вывод порта приведено на рисунке 125

18.1.5 **PORT_PULL**

Таблица 168 – Регистр PORT_PULL

Номер	31…16	15…0
Доступ	R/W	R/W
Сброс	0	0
	PULL_UP[15:0]	PULL_DOWN[15:0]

Таблица 169 – Описание бит регистра PORT_PULL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	PULL_UP[15:0]	Разрешение подтяжки к питанию вывода порта: 0 – подтяжка к питанию U_{CC} выключена; 1 – подтяжка к питанию U_{CC} включена
15…0	PULL_DOWN[15:0]	Разрешение подтяжки к земле вывода порта: 0 – подтяжка к земле выключена; 1 – подтяжка к земле включена

18.1.6 **PORT_PWR**

Таблица 170 – Регистр PORT_PWR

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

Таблица 171 – Описание бит регистра PORT_PWR

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

18.1.7 **PORT_SETTX**

Таблица 172 – Регистр PORT_SETTX

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

Таблица 173 – Описание бит регистра PORT_SETTX

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15…0	SETTX[15:0]	Регистр установки выходных данных передатчика порта. При записи «1» производится установка в «1» соответствующего бита регистра выходных данных порта, при записи «0» значение регистра выходных данных порта не изменяется. При чтении возвращается значение, записанное в регистр выходных данных порта. **Внимание!** Считанное из регистра значение может отличаться от реального значения на соответствующем выводе порта, например, если передатчик выключен

18.1.8 **PORT_CLRTX**

Таблица 174 – Регистр PORT_CLRTX

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

Таблица 175 – Описание бит регистра PORT_CLRTX

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15…0	CLRTX[15:0]	Регистр сброса выходных данных передатчика порта. При записи «1» производится сброс в «0» соответствующего бита регистра выходных данных порта, при записи «0» значение регистра выходных данных порта не изменяется. При чтении возвращается значение, записанное в регистр выходных данных порта. **Внимание!** Считанное из регистра значение может отличаться от реального значения на соответствующем выводе порта, например, если передатчик выключен

18.1.9 **PORT_RDTX**

Таблица 176 – Регистр PORT_RDTX

Номер	31…16	15…0
Доступ	U	RO
Сброс	0	0
	-	RDTX[15:0]

Таблица 177 – Описание бит регистра PORT_RDTX

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15…0	RDTX[15:0]	Выходные данные передатчика порта. **Внимание!** Считанное из регистра значение может отличаться от реального значения на соответствующем выводе порта, например, если передатчик выключен

19 Детектор напряжения питания

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

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

Таблица 178 – Параметры формирования флагов PVD и PVDB

Наименование параметра, единица измерения	Норма параметра		
	не менее	типовое	не более
Напряжение питания, В	2,3	-	3,6
Напряжение питания батарейного домена, В	1,8	-	3,6
Уровень срабатывания PVD, В при PLS = “000”		2,3	
Уровень срабатывания PVDB, В при PLSB = “000”		1,8	
Уровень срабатывания PVD, В при PLS = “001”		2,4	
Уровень срабатывания PVDB, В при PLSB = “001”		2,0	
Уровень срабатывания PVD, В при PLS = “010”		2,5	
Уровень срабатывания PVDB, В при PLSB = “010”		2,2	
Уровень срабатывания PVD, В при PLS = “011”		2,6	
Уровень срабатывания PVDB, В при PLSB = “011”		2,4	
Уровень срабатывания PVD, В при PLS = “100”		2,7	
Уровень срабатывания PVDB, В при PLSB = “100”		2,6	
Уровень срабатывания PVD, В при PLS = “101”		2,8	
Уровень срабатывания PVDB, В при PLSB = “101”		2,8	
Уровень срабатывания PVD, В при PLS = “110”		2,9	
Уровень срабатывания PVDB, В при PLSB = “110”		3,0	
Уровень срабатывания PVD, В при PLS = “111”		3,0	
Уровень срабатывания PVDB, В при PLSB = “111”		3,2	

19.1 Описание регистров блока PVD

Таблица 179 – Описание регистров блока PVD

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

19.1.1 PVDCS

Таблица 180 – Регистр PVDCS

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

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

Таблица 181 – Описание бит регистра PVDCS

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...27	-	Зарезервировано
26	INVB	Флаг инверсии выхода от датчика PVDB: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то он выставляется при превышении заданного уровня. Если инвертируется, то – при снижении ниже заданного уровня
25	-	Зарезервировано
24	INV	Флаг инверсии выхода от датчика PVD: 0 – нет инверсии; 1 – есть инверсия. Если флаг не инвертируется, то он выставляется при превышении заданного уровня. Если инвертируется, то – при снижении ниже заданного уровня
23	-	Всегда записывать «0»
22	IEPVD	Флаг разрешения прерывания от датчика PVD: 0 – прерывание запрещено; 1 – прерывание разрешено
21	IEPVDB	Флаг разрешения прерывания от датчика PVDB: 0 – прерывание запрещено; 1 – прерывание разрешено

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

20 Таймеры общего назначения

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

– подсчет циклов частоты TIM_CLK или каких-либо внешних событий;

– формирование прерываний и запросов DMA по заданным событиям;

– захват входных сигналов, в том числе измерение длительности импульсов входных сигналов;

– генерацию различных форм выходных сигналов.

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

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

Каждый из таймеров позволяет генерировать прерывания и запросы DMA.

20.1 Основные характеристики

Основные характеристики блока таймера:

– 32-битный счетчик: счет прямой, обратный или двунаправленный;

– 32-разрядный предварительный делитель частоты TIM_CLK;

– схема выбора источника тактирования основного счетчика от внешних сигналов или от других таймеров;

– четыре независимых канала, каждый канал может работать в одном из режимов:

  – режим захвата: позволяет захватить (сохранить) текущее значение счетчика при изменении некоторого входного сигнала;
  – режим ШИМ: позволяет осуществлять непрерывное сравнение заданных значений со значением счетчика для формирования выходных сигналов;

– формирование выходных сигналов в режиме ШИМ:
  – сброс в НИЗКИЙ уровень при совпадении;
  – установка в ВЫСОКИЙ уровень при совпадении;
  – переключение (инвертирование) при совпадении;
  – переключение при некотором условии;

– формирование прерываний и запросов DMA по событиям:
  – обновление счетчика;
  – захват;
  – сравнение;
  – внешние события по входам ETR и BRK.

20.1.1 **Структурная схема**

Структурная схема блока «Таймер» представлена на рисунке 29.

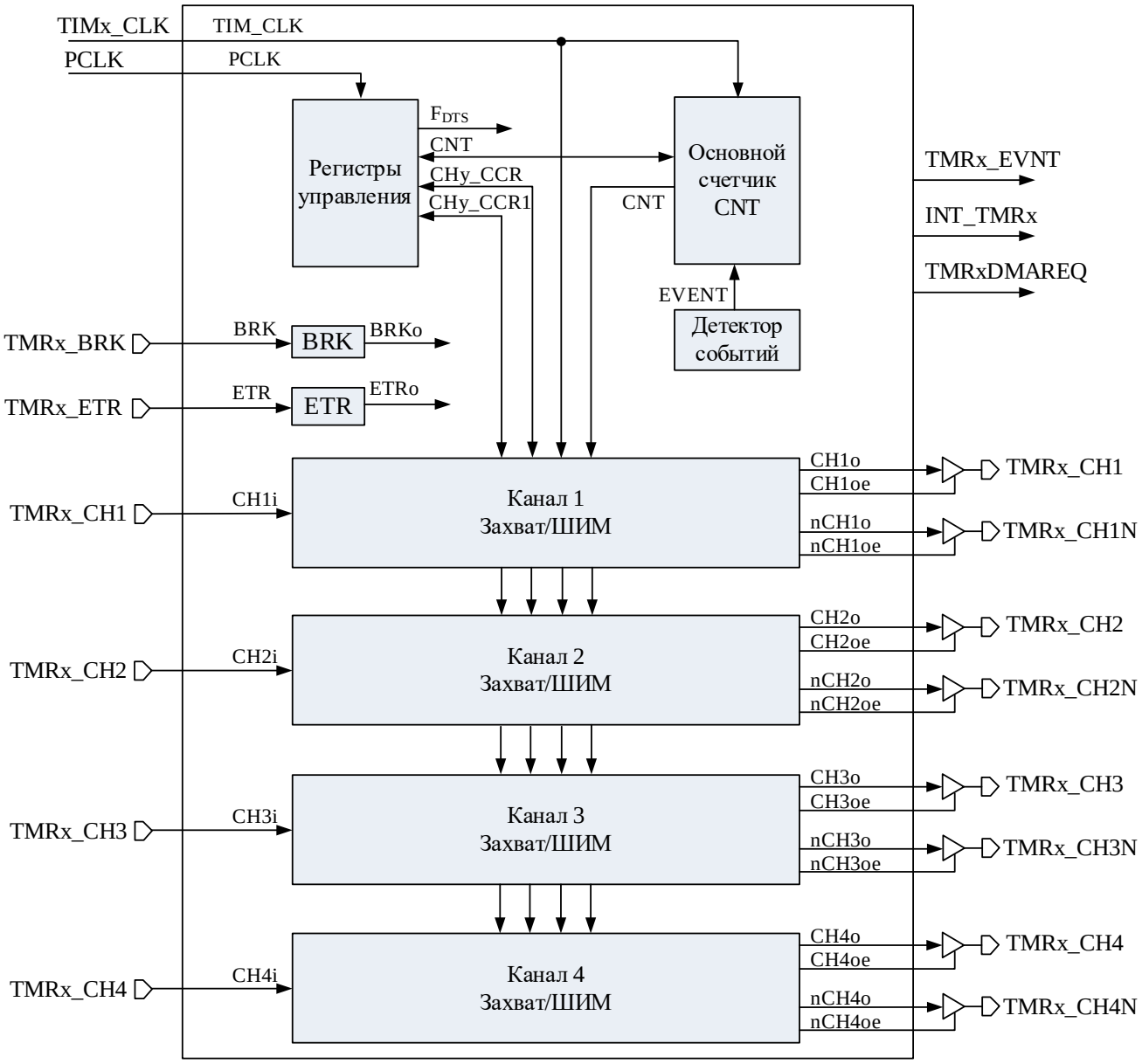


Рисунок 29 – Структурная схема блока «Таймер»

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

Таймер позволяет работать в режимах:

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

20.2 Базовый блок таймера

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

– основной счетчик таймера (CNT);

– основание счета (максимальное значение) основного счетчика (ARR);

– делитель частоты TIM_CLK (PSG), используемый для тактирования основного счетчика;

– регистр управления основным счетчиком (CNTRL).

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

20.2.1 Инициализация тактирования таймера

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

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

Для разрешения тактовой частоты PCLK необходимо установить бит тактирования блока в регистре PER2_CLOCK: бит 14 для таймера 1, бит 15 для таймера 2, бит 27 для таймера 3, бит 28 для таймера 4. Задание тактовой частоты TIM_CLK осуществляется в регистре TIM_CLOCK: в поле TIMxBRG устанавливается коэффициент деления тактовой частоты PER1_C2 для формирования частоты TIM_CLK, разрешение подачи частоты TIM_CLK на блок таймера управляется битом TIMxCLKEN.

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

После подачи тактовых сигналов можно приступать к работе с таймером.

20.2.2 Инициализация основного счетчика таймера

Чтобы запустить работу основного счетчика необходимо задать:

– начальное значение основного счетчика таймера в регистре CNT;

– значение основания счета для основного счетчика в регистре ARR;

– режим работы счетчика в регистре CNTRL:

  – выбрать источник события переключения счетчика EVNT_SEL[3:0];

– режим счета основного счетчика CNT_MODE[1:0]:

  – значения 00 и 01 при тактировании внутренней частотой;

  – значения 10 при тактировании внешними сигналами;

  – направление счета основного счетчика DIR;

– при тактировании внутренней частотой установить значение предварительного делителя в регистре PSG, основной счетчик при этом будет считать на частоте TIM_CLKd = TIM_CLK / (PSG + 1);

– разрешить работу счетчика CNT_EN.

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

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

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

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

20.2.3 Режимы счета

20.2.3.1 Счет прямой: CNT_MODE[1:0] = 00, DIR = 0

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

//Разрешение работы таймера
TIMERx->CNTRL = 0x00000001;//Счет прямой по TIM_CLK
```

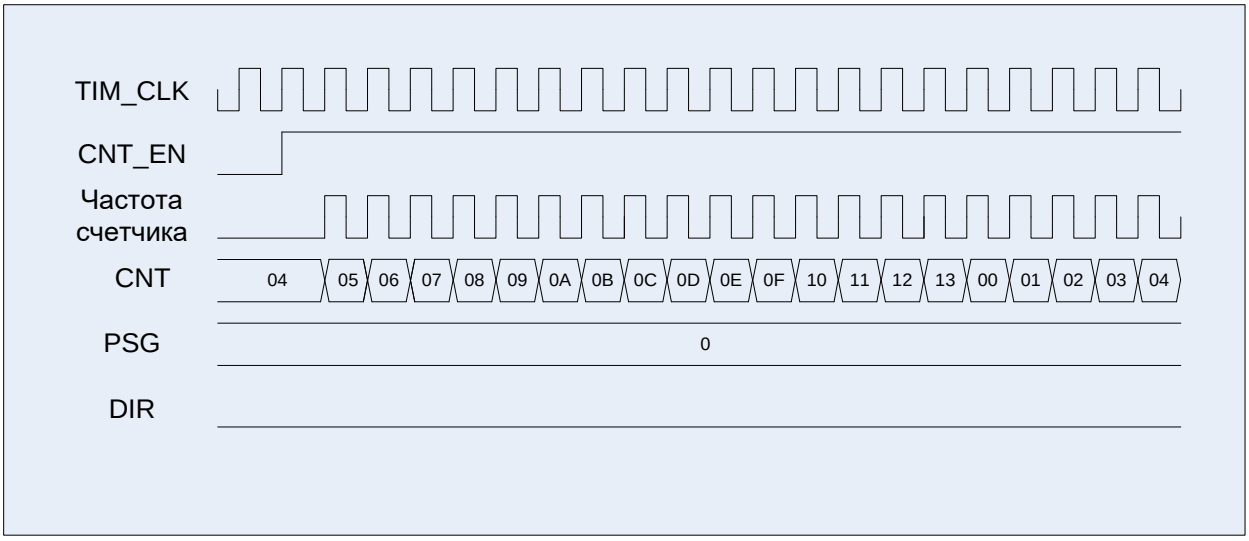


Рисунок 30 – Диаграммы работы таймера, счет прямой от 0 до 0x13, стартовое значение 0x04

20.2.3.2 Счет обратный: CNT_MODE[1:0] = 00, DIR = 1

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

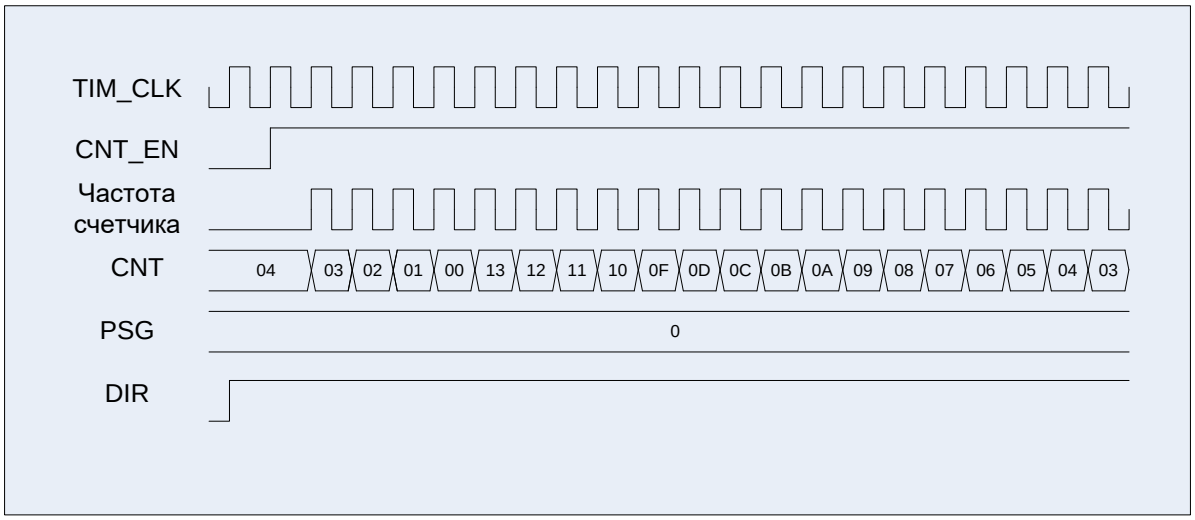


Рисунок 31 – Диаграммы работы таймера, счет обратный от 0x13 до 0, стартовое значение 0x04

20.2.3.3 Счет двунаправленный: CNT_MODE = 01, DIR = 0

```
TIMERx->CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMERx->CNT = 0x00000004;//Начальное значение счетчика
TIMERx->PSG = 0x00000000;//Предделитель частоты TIM_CLK
TIMERx->ARR = 0x00000013;//Основание счета
//Разрешение работы таймера.
TIMERx->CNTRL = 0x00000041;//Счет двунаправленный по TIM_CLK
```

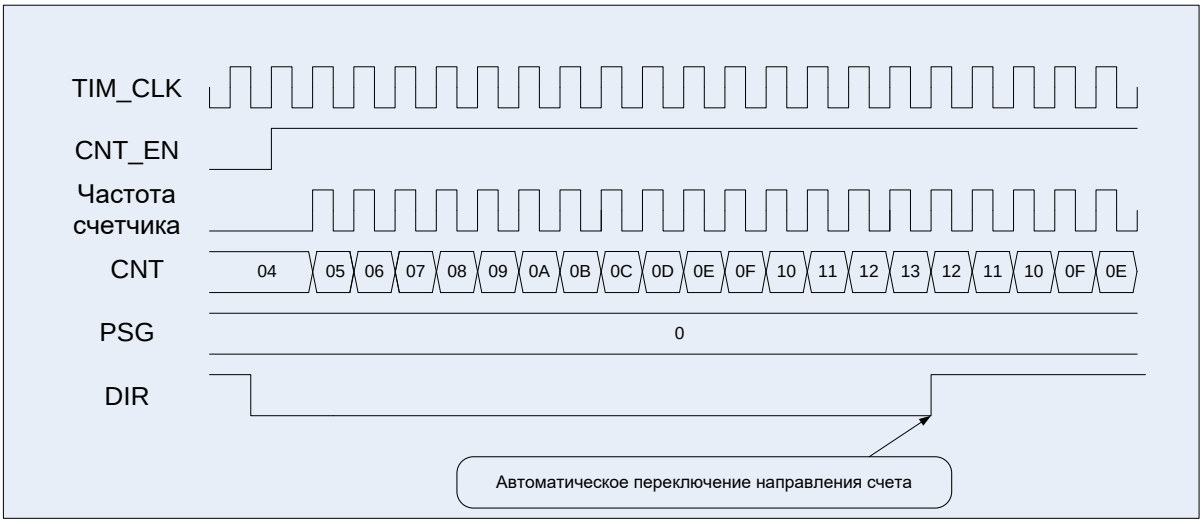


Рисунок 32 – Диаграммы работы таймера, счет двунаправленный, сначала прямой

20.2.3.4 Счет двунаправленный: CNT_MODE = 01, DIR = 1

```
TIMERx->CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMERx->CNT = 0x00000004;//Начальное значение счетчика
TIMERx->PSG = 0x00000000;//Предделитель частоты TIM_CLK
TIMERx->ARR = 0x00000013;//Основание счета
//Разрешение работы таймера.
TIMERx->CNTRL = 0x00000049;//Счет двунаправленный по TIM_CLK
```

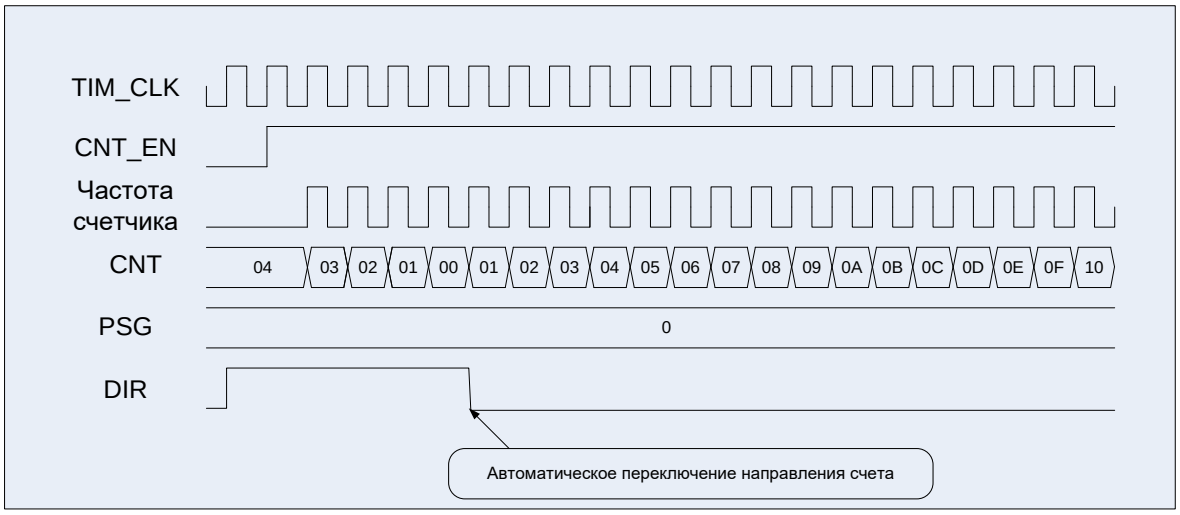


Рисунок 33 – Диаграммы работы таймера, счет двунаправленный, сначала обратный

20.2.4 Тактовая частота F_{DTS}

В блоке таймера предусмотрено формирование дополнительной тактовой частоты F_{DTS}, которая может использоваться для работы генератора «мертвой зоны» и цифровых фильтров на входах ETR и CHyi.

Тактовая частота F_{DTS} формируется из частоты TIM_CLK путём прореживания на заданный коэффициент (1, 2, 3 или 4). Настройка частоты F_{DTS} осуществляется в регистре CNTRL, поле FDTS[1:0].

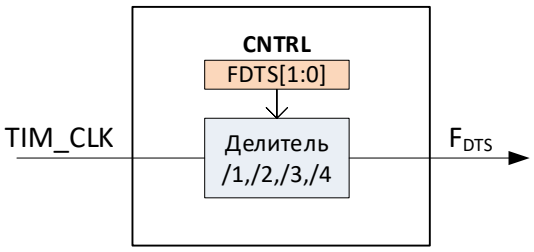


Рисунок 34 – Схема формирования тактовой частоты F_{DTS}

Диаграмма возможных частот F_{DTS} приведена на рисунке 35.

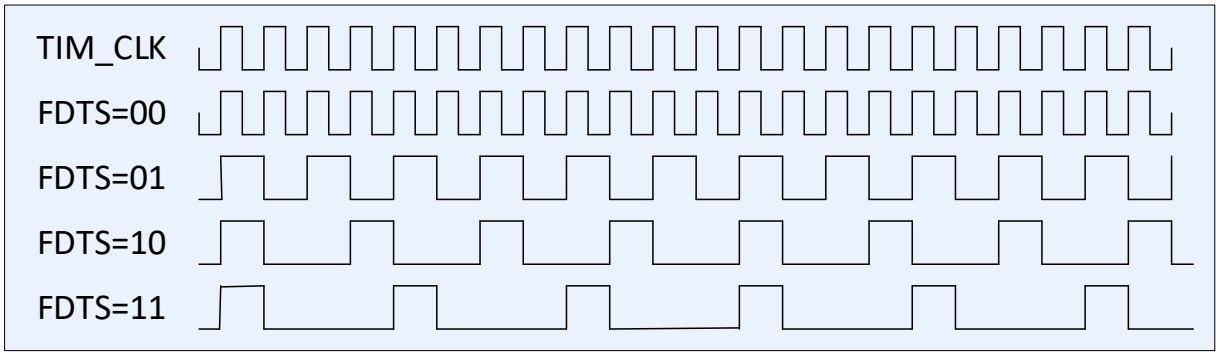


Рисунок 35 – Диаграмма тактовой частоты F_{DTS} в зависимости от значения FDTS[1:0] в регистре CNTRL

20.3 Источники событий для счета

Тактирование основного счетчика таймера может осуществляться от следующих источников:

– внутренний тактовый сигнал (TIM_CLKd);

– событие в другом таймере (CNT==ARR);

– внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CHyi;

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

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

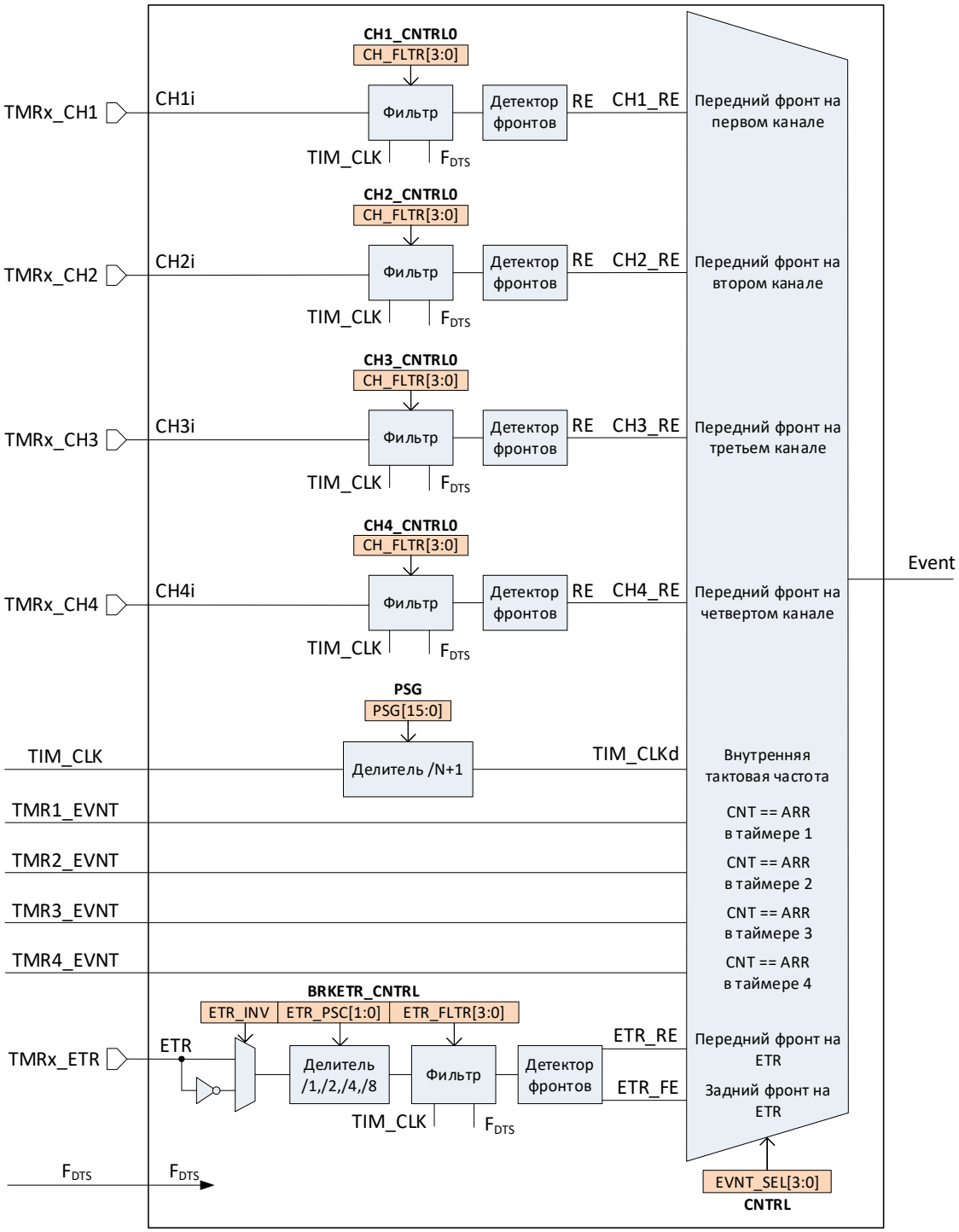


Рисунок 36 – Структурная схема формирования события для счета

20.3.1 **Внутренний тактовый сигнал (TIM_CLKd)**

Данный режим выбирается, когда EVNT_SEL[3:0] = 0000 и CNT_MODE[1:0] = 0x в регистре CNTRL. Основной счетчик таймера тактируется от внутренней частоты TIM_CLKd, которая формируется путем деления частоты TIM_CLK в соответствии с коэффициентом деления, записанным в регистре PSG.

Если значение предварительного делителя основного счетчика (PSG) неравно нулю, то счетный регистр делителя будет инкрементироваться по каждому импульсу сигнала TIM_CLK до тех пор, пока не достигнет значения, находящегося в регистре

делителя. Далее счетный регистр делителя сбрасывается в ноль, содержимое основного счетчика таймера изменяется на 1 и счет начинается заново. Таким образом выходная частота предварительного делителя составляет

$$\mathrm{TIM_CLKd} = \frac{\mathrm{TIM_CLK}}{\mathrm{PSG} + 1} \tag{4}$$

Значение регистра PSG можно изменять даже во время работы счетчика, новое значение предделителя вступит в силу сразу после записи. На рисунках 37, 38 приведены диаграммы работы счетчика при обновлении значения PSG.

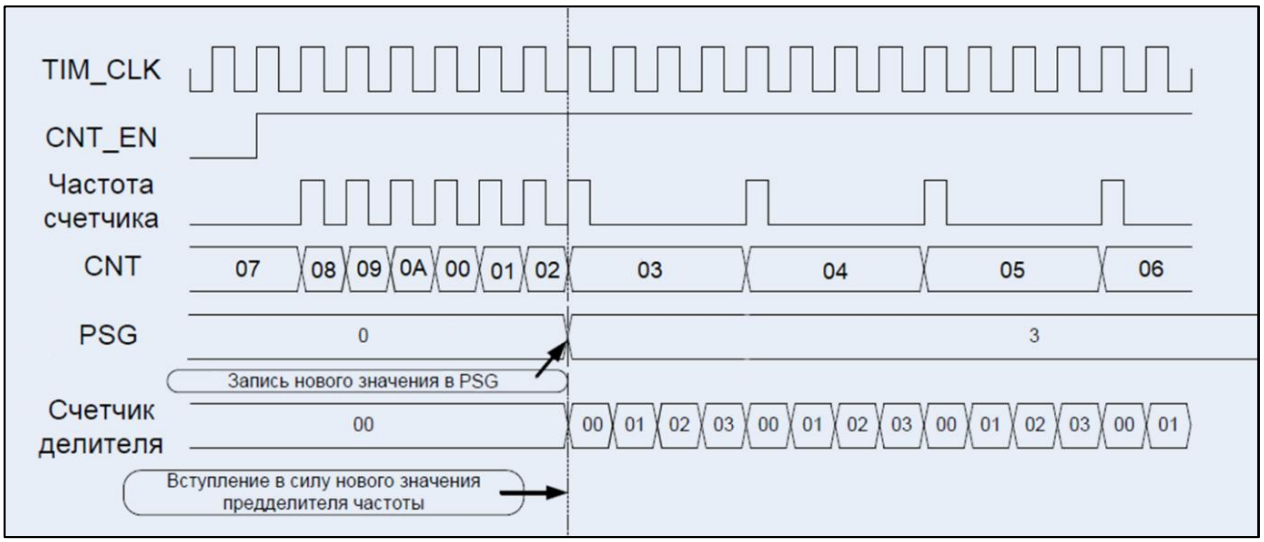


Рисунок 37 – Диаграмма работы счетчика: счет прямой
(CNT_MODE[1:0] = 00, EVNT_SEL[3:0] = 0000, DIR = 0)

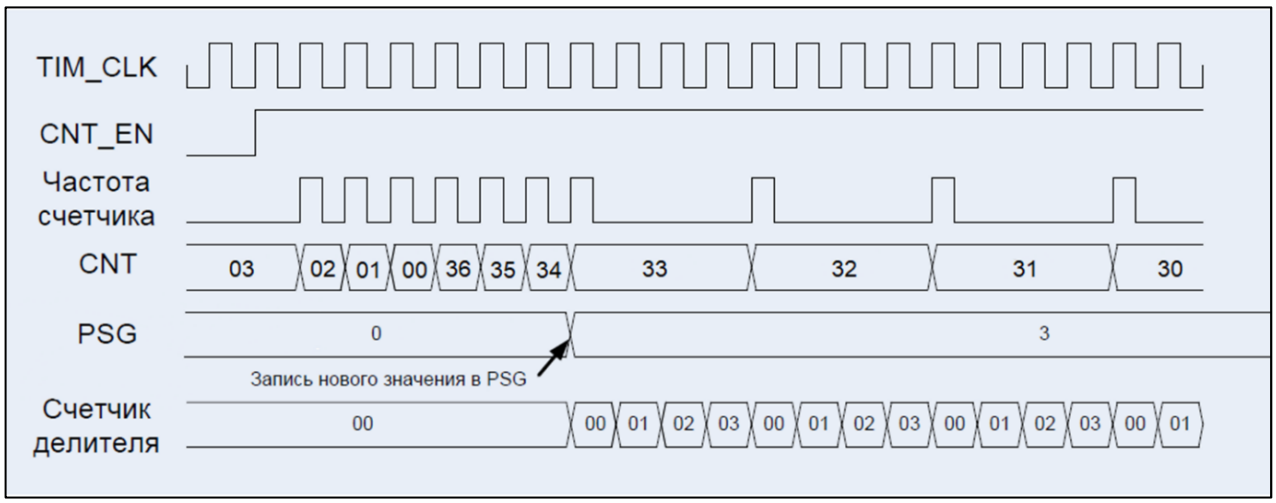


Рисунок 38 – Диаграмма работы счетчика: счет обратный
(CNT_MODE[1:0] = 00, EVNT_SEL[3:0] = 0000, DIR = 1)

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

Все таймеры полностью независимы друг от друга, но при этом у них предусмотрена возможность синхронизированной работы. Это позволяет создавать более сложные массивы таймеров, которые работают полностью автономно и не требуют написания какого-либо кода программы для выполнения сложных временных функций.

У каждого таймера имеется выход запуска TMRx_EVNT, который соединен с входами других таймеров. Тактирование от другого таймера выбирается, когда EVNT_SEL[3:0] = 0001 – 0011 или 1010, а также CNT_MODE[1:0] = 10 в регистре CNTRL. Основной счетчик таймера тактируется от другого таймера по сигналу TMRx_EVNT, который устанавливается при CNT == ARR. Пересинхронизация сигнала TMRx_EVNT (CNT == ARR) с одного таймера на другой происходит с задержкой один такт частоты TIM_CLK.

Синхронизация таймеров возможна в различных режимах. На рисунке 39 показан пример каскадного соединения таймеров, диаграммы работы данных таймеров приведены на рисунке 40.

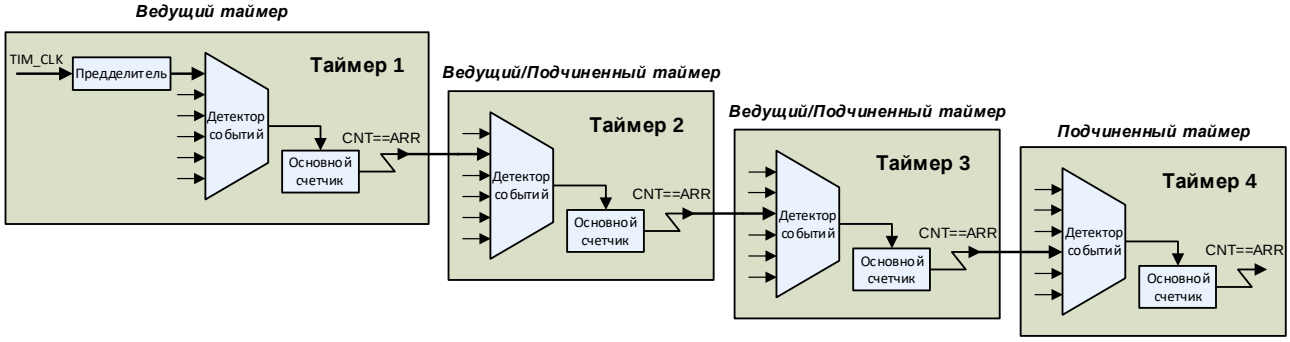


Рисунок 39 – Пример каскадного соединения таймеров

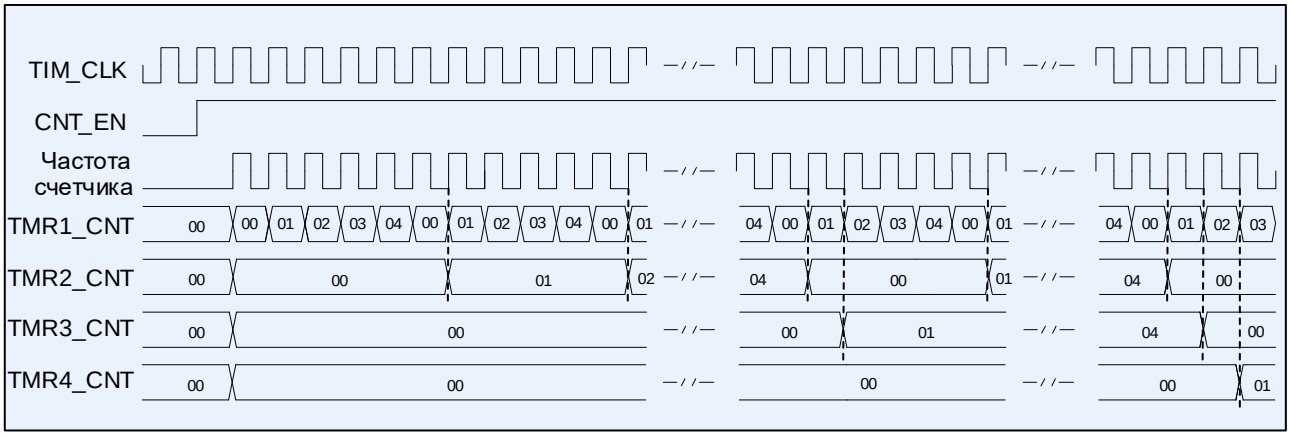


Рисунок 40 – Диаграммы работы четырех таймеров в каскаде
TIMER1: DIR = 0, EVNT_SEL[3:0] = 0000, CNT_MODE[1:0] = 00;
TIMER2: DIR = 0, EVNT_SEL[3:0] = 0001, CNT_MODE[1:0] = 10;
TIMER3: DIR = 0, EVNT_SEL[3:0] = 0010, CNT_MODE[1:0] = 10;
TIMER4: DIR = 0, EVNT_SEL[3:0] = 0011, CNT_MODE[1:0] = 10.

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

Данный режим выбирается, когда EVNT_SEL[3:0] = 01xx и CNT_MODE[1:0] = 10. Основной счетчик таймера считает по переднему фронту внешнего сигнала, поступающего на вход канала CHyi. Биты CH_SEL[1:0] регистра CHy_CNTRL0 не оказывают влияния, так как они применяются для работы канала таймера только в режиме захвата.

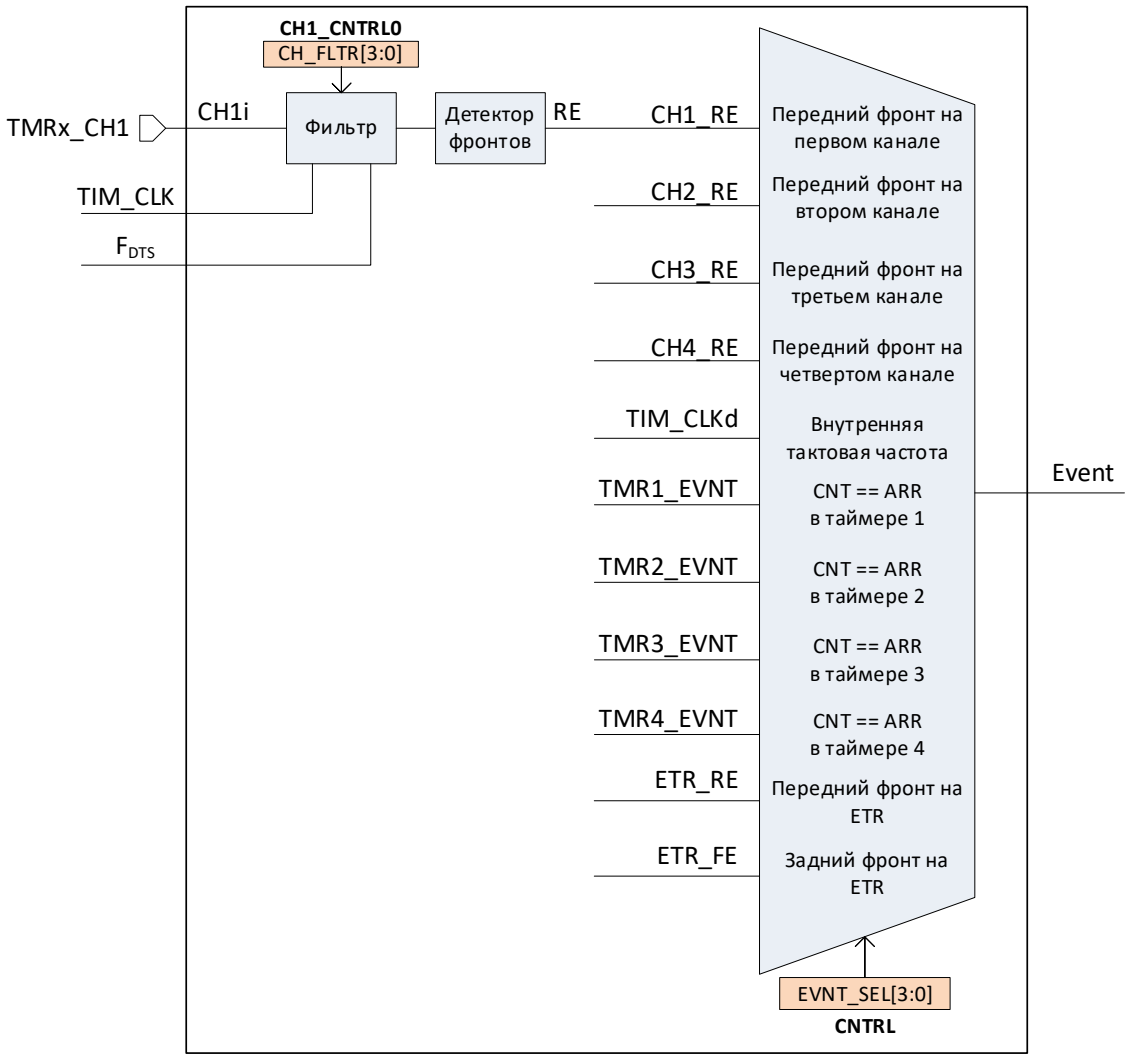


Рисунок 41 – Схема тактирования сигналом со входа первого канала

Со входа CHyi внешний тактовый сигнал поступает в блок цифрового фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. «Блок цифрового фильтра»). Настройки фильтра задаются в поле CH_FLTR[3:0] регистра CHy_CNTRL0.

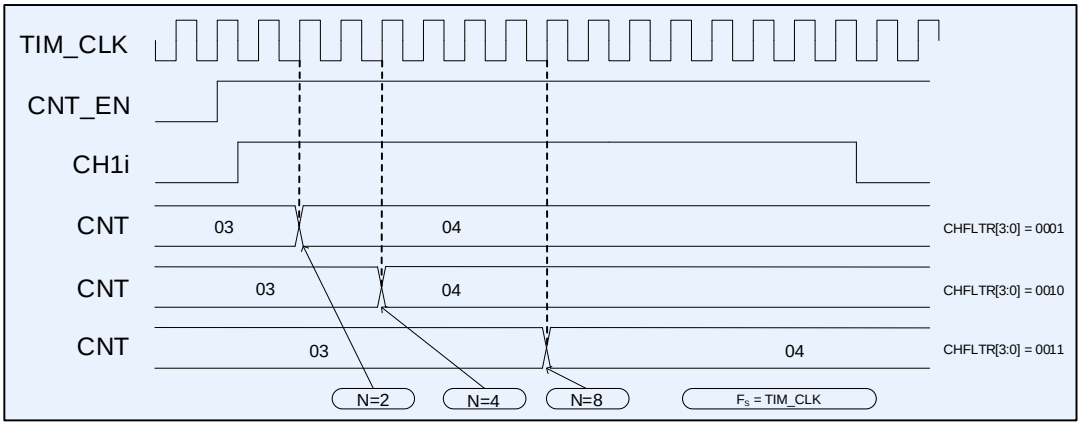


Рисунок 42 – Диаграмма внешнего тактирования с разными вариантами фильтра

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

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

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

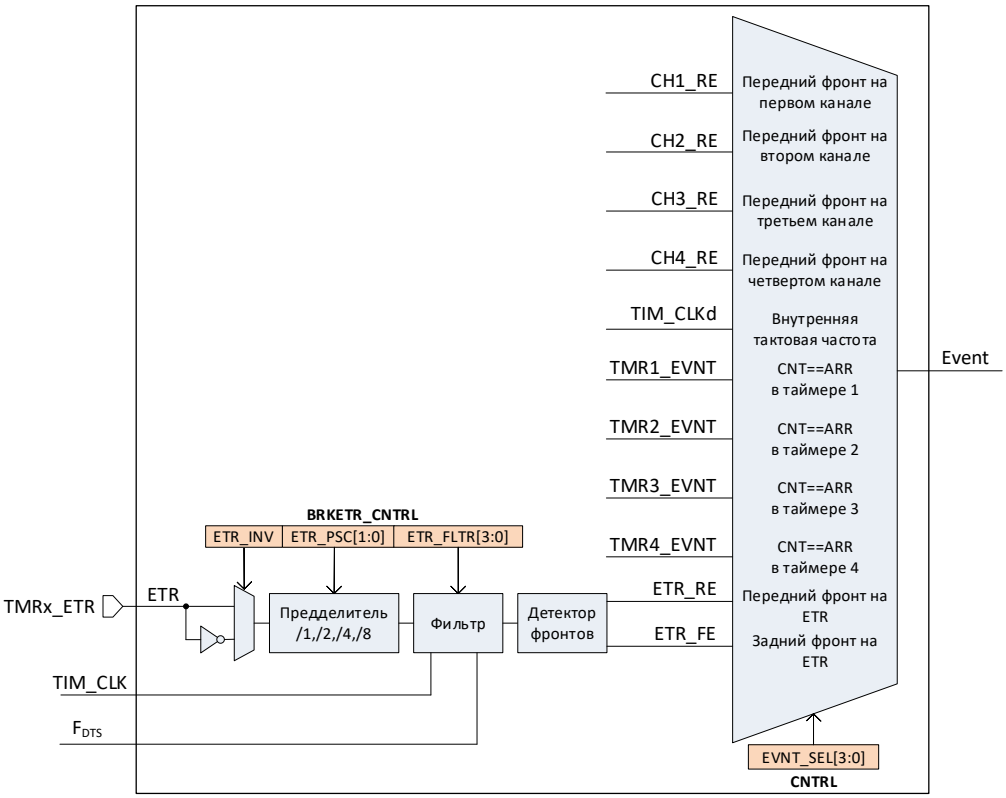


Рисунок 43 – Схема тактирования сигналом со входа ETR

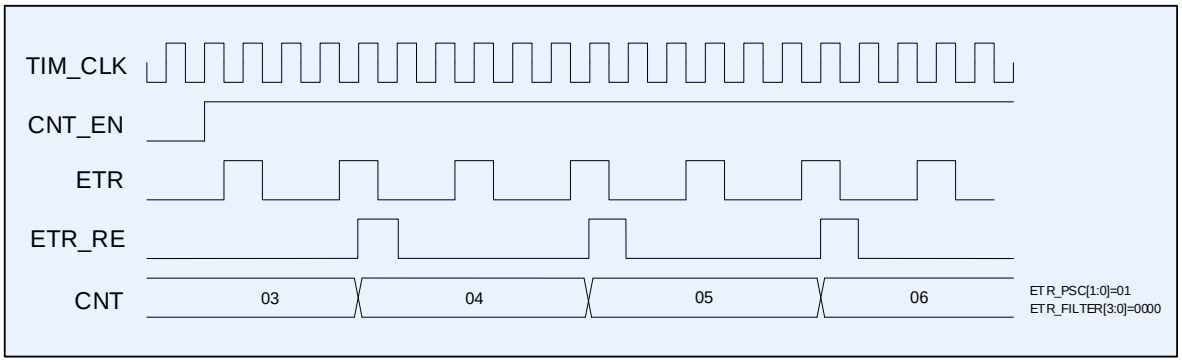


Рисунок 44 – Диаграмма тактирования со входа ETR, EVNT_SEL[3:0] = 1000

20.4 **Режим захвата**

Каждый канал таймера может быть независимо переведен в режим захвата. В режиме захвата по событию от внешнего входного сигнала происходит фиксация значения основного счетчика CNT в регистры CHy_CCR и CHy_CCR1. Регистрация событий осуществляется только по входам CHyi (выводы, настроенные в функцию TMRx_CHy).

Структурная схема блока захвата представлена на рисунке 45.

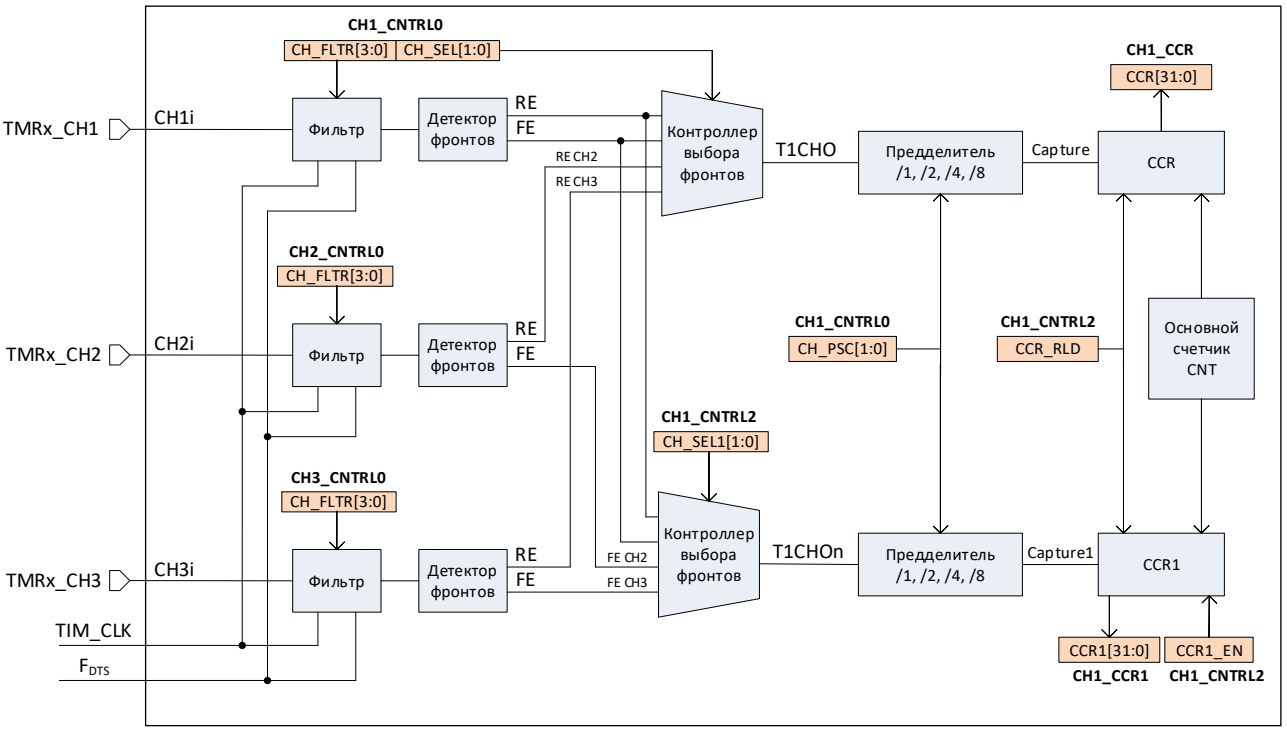


Рисунок 45 – Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо записать «1» в бит CAP_NPWM регистра управления каналом CHy_CNTRL0. Для использования регистра CHy_CCR1 необходимо записать «1» в бит CCR1_EN регистра CHy_CNTRL2.

Внешний сигнал со входа CHyi сначала поступает в блок фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. «Блок цифрового фильтра»). Настройки фильтра задаются в поле CH_FLTR[3:0] регистра CHy_CNTRL0.

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

В блоке «Контроллер выбора фронтов» производится выбор используемого для захвата сигнала между положительным фронтом канала, отрицательным фронтом канала и положительными и отрицательными фронтами сигналов от других каналов. Настройка блока «Контроллер выбора фронтов» для регистра CHy_CCR осуществляется в поле CH_SEL[1:0] регистра CHy_CNTRL0, а для регистра CHy_CCR1 – в поле CH_SEL1[1:0] регистра CHy_CNTRL2. Выбранный для захвата сигнал поступает в предварительный

делитель, который в зависимости от значения в поле CH_PSC[1:0] регистра CHy_CNTRL0 позволяет фиксировать все события, либо каждое второе, каждое четвертое или каждое восьмое событие.

Предварительный делитель для регистра CHy_CCR формирует сигнал Capture, а предварительный делитель для регистра CHy_CCR1 формирует сигнал Capture1. По сигналам Capture и Capture1 выполняется запись текущего значения основного счетчика CNT в регистры CHy_CCR и CHy_CCR1.

На рисунке X18 показан пример захвата значения основного счетчика CNT в регистр CHy_CCR по положительному фронту на входе канала, а в регистр CHy_CCR1 – по отрицательному фронту на входе канала.

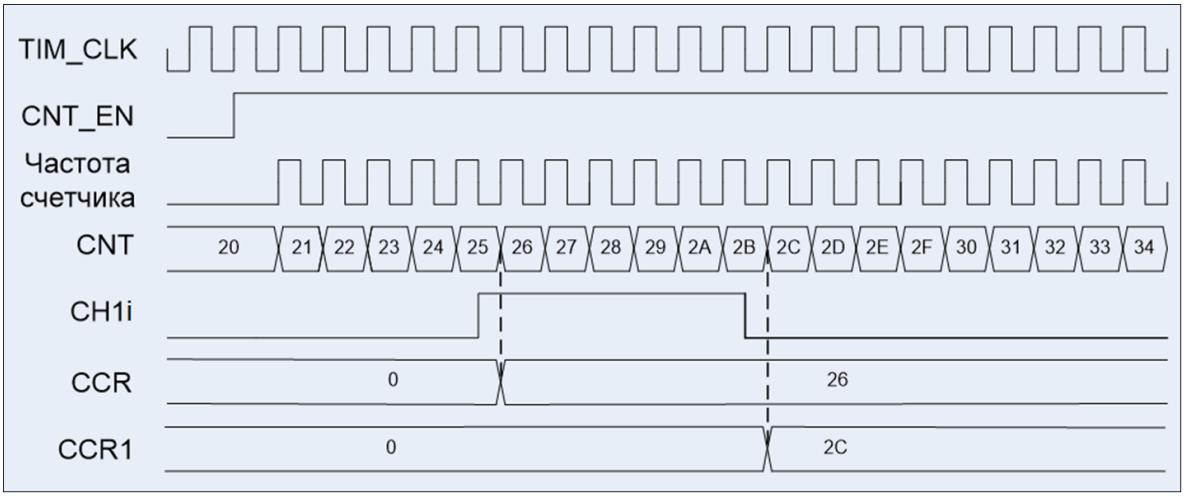


Рисунок 46 – Диаграмма захвата события со входа канала 1

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

Между формированием события захвата и записью текущего значения основного счетчика CNT в регистры CHy_CCR и CHy_CCR1 может быть установлена задержка с помощью бита EVNT_DLY в регистре CHy_CNTRL2. Если бит EVNT_DLY равен «0», то сначала формируется событие захвата, а затем через один такт сигнала синхронизации TIM_CLK выполняется запись CNT в регистры CHy_CCR и CHy_CCR1. Если бит EVNT_DLY равен «1», то обновление информации в регистрах CHy_CCR и CHy_CCR1 происходит синхронно с событием захвата. При реализации чтения регистров CHy_CCR и CHy_CCR1 по событию захвата рекомендуется устанавливать бит EVNT_DLY в «1».

20.5 **Режим ШИМ**

Каждый канал таймера может быть независимо переведен в режим ШИМ для формирования выходных сигналов с возможностью задания «мертвой зоны». Структурная схема блока формирования ШИМ представлена на рисунке 47.

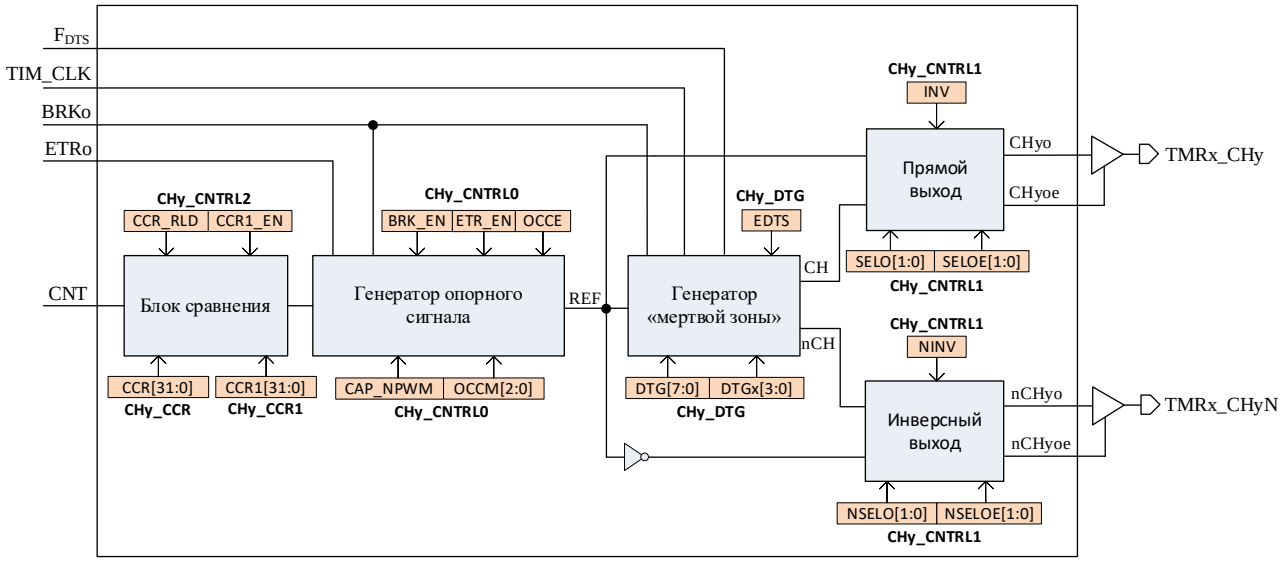


Рисунок 47 – Структурная схема блока формирования ШИМ

Для включения режима ШИМ для определенного канала необходимо в регистре управления каналом CHy_CNTRL0 записать «0» в бит CAP_NPWM.

20.5.1 **Генератор опорного сигнала REF**

При работе в режиме ШИМ блок генератора опорного сигнала формирует сигнал REF. Данный сигнал формируется на основании сравнения значения в регистрах CHy_CCR, CHy_CCR1 и основного счетчика CNT. Формат выработки сигнала REF устанавливается в регистре управления каналом таймера CHy_CNTRL0, поле OCCM[2:0].

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

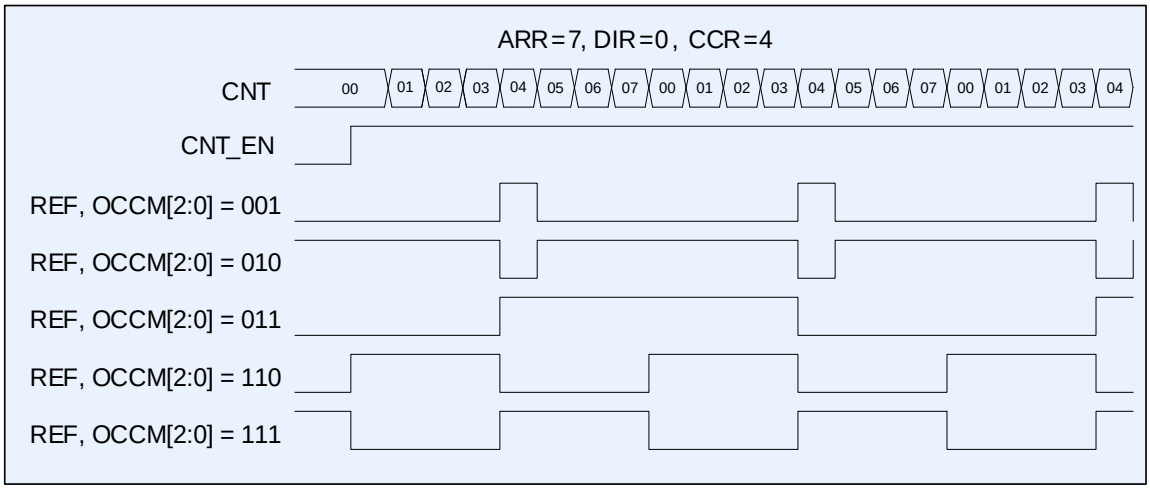


Рисунок 48 – Диаграмма работы в режиме ШИМ, CCR1_EN=0

Если в регистре CHy_CNTRL2 бит CCR1_EN = 1, то для формирования сигнала REF задействуются оба результата сравнения значения в регистрах CHy_CCR, CHy_CCR1 и основного счетчика CNT.

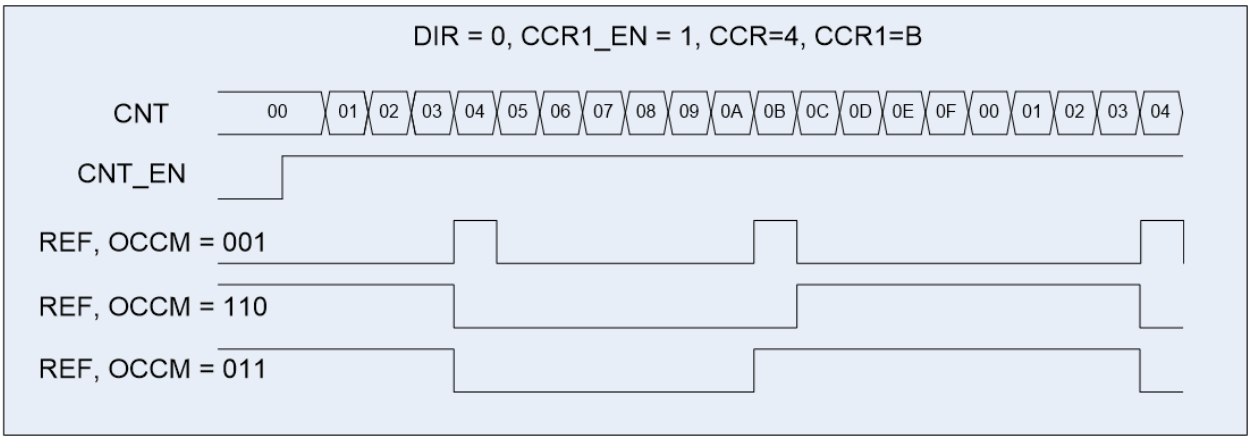


Рисунок 49 – Диаграмма работы в режиме ШИМ, CCR1_EN = 1

Запись новых значений в регистры CHy_CCR и CHy_CCR1 осуществляется немедленно, если в регистре CHy_CNTRL2 бит CCR_RLD установлен в «0», иначе регистры CHy_CCR и CHy_CCR1 получат новые значения только при CNT == 0. Процесс обновления значений в регистрах CHy_CCR и CHy_CCR1 обозначается в регистре CHy_CNTRL0 с помощью флагов WR_CMPL и WR_CMPL1, соответственно. На время выполнения записи флаг WR_CMPL/WR_CMPL1 устанавливается в «1», по окончании записи флаг WR_CMPL/WR_CMPL1 сбрасывается в «0».

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

Для разрешения сброса сигнала REF по входу ETR необходимо установить бит ETR_EN и OCCE в регистре CHy_CNTRL0. Активный уровень на входе ETR сбрасывает

сигнал REF в «0». После снятия активного уровня на входе ETR сигнал REF остается в «0» до следующего события установки REF в «1» (см. рисунок 50).

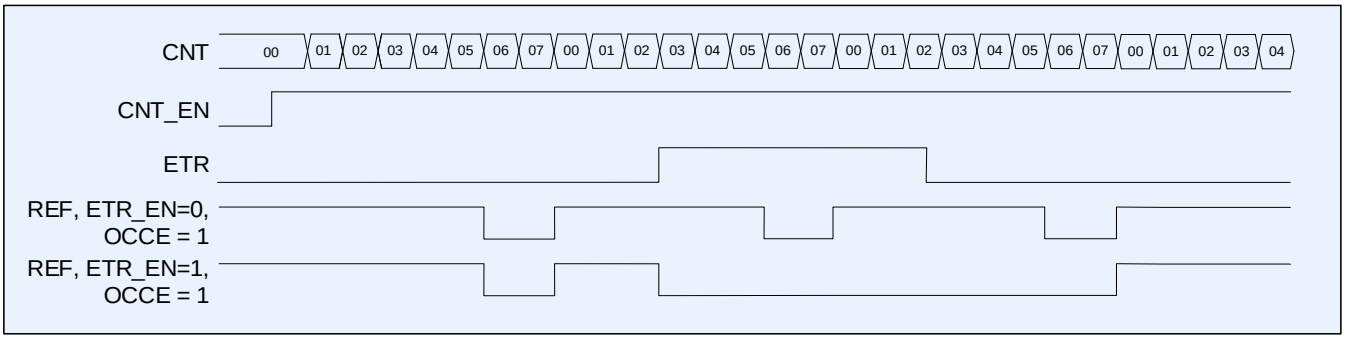


Рисунок 50 – Диаграмма сброса сигнала REF по выводу ETR

Для разрешения сброса сигнала REF по входу BRK необходимо установить бит BRK_EN в регистре CHy_CNTRL0. Активный уровень на входе BRK сбрасывает сигнал REF в «0» путем маскирования. После снятия активного уровня на входе BRK генерация сигнала REF сразу же восстанавливается (см. рисунок 51).

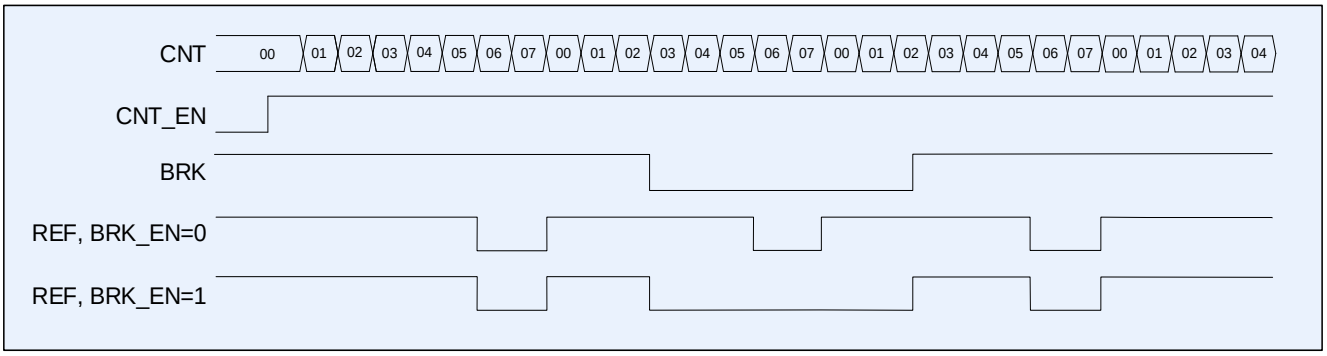


Рисунок 51 – Диаграмма сброса сигнала REF по выводу BRK

20.5.2 **Генератор «мертвой зоны»**

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

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

– сигнал на инверсном выходе (nCHyo, nCHyoe) представляет собой сигнал REF, в котором передний фронт задержан на величину DTGdel относительно переднего фронта опорного сигнала REF.

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

$$\mathrm{DTGdel} = \mathrm{DTG} \cdot (\mathrm{DTGx} + 1)\,, \tag{5}$$

где DTGx – предварительный делитель частоты;

DTG – основной делитель частоты.

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

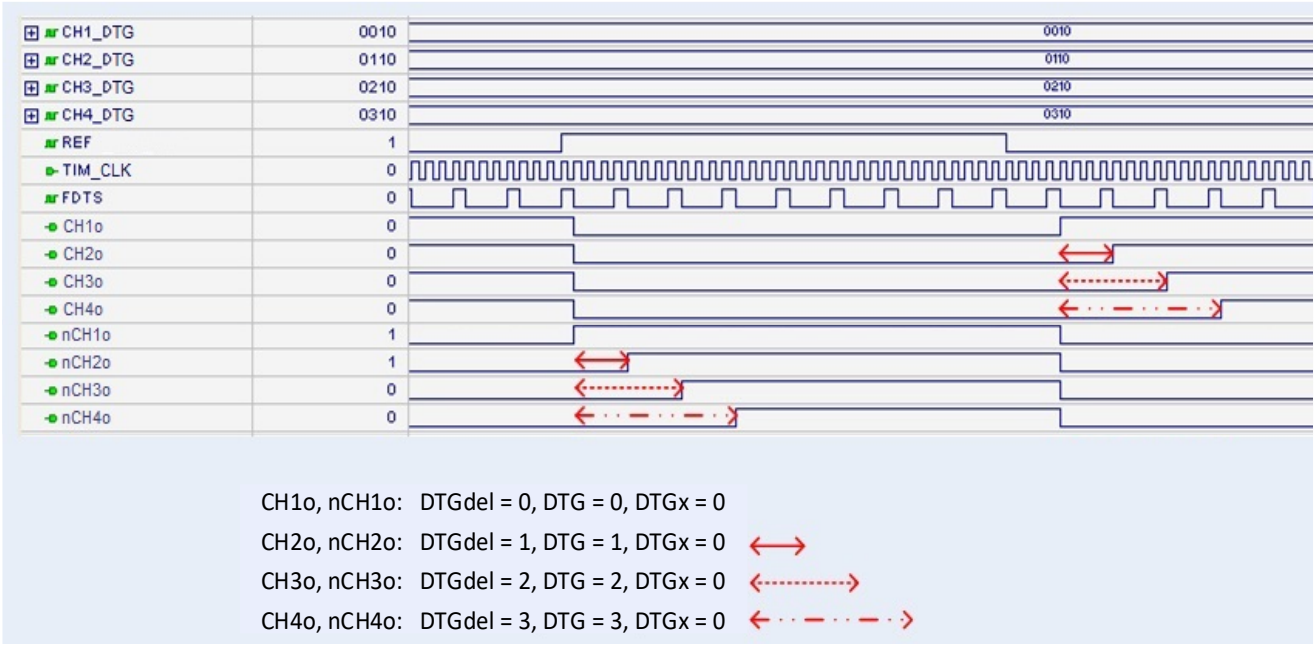


Рисунок 52 – Диаграмма работы блока DTG

Выходные сигналы блока DTG могут быть принудительно установлены в «0» с использованием внешнего сигнала сброса, поступающего со входа BRK (низкий активный уровень). Активный уровень на входе BRK может быть изменён с помощью инверсии входного сигнала, бит BRK_INV в регистре BRKETR_CNTRL. Для разрешения сброса выходных сигналов блока DTG по входу BRK необходимо установить бит BRK_EN в регистре CHy_CNTRL0. Активный уровень на входе BRK сбрасывает выходные сигналы блока DTG в «0» путем их маскирования.

20.5.3 **Выходные блоки**

Каждый канал таймера имеет два выходных блока – прямой и инверсный. Каждый выходной блок формирует как сигнал выдачи (CHyo, nCHyo), так и сигнал разрешения выдачи (CHyoe, nCHyoe). В качестве сигналов для прямого (CHyo, CHyoe) и инверсного (nCHyo, nCHyoe) выходов в регистре CHy_CNTRL1 могут быть заданы либо постоянные уровни (0 или 1), либо сигналы, формируемые на основе сигнала REF. К таким сигналам относится сам сигнал REF, а также сигналы, формируемые блоком DTG.

Выбор источника сигнала выдачи для прямого (CHyo) и инверсного (nCHyo) выходов задается в полях SELO[1:0] и NSELO[1:0] регистра CHy_CNTRL1. Дополнительно каждый сигнал выдачи для прямого (CHyo) и инверсного (nCHyo) выходов может быть инвертирован путем установки битов INV и NINV в регистре CHy_CNTRL1. Выбор источника сигнала разрешения выдачи для прямого (CHyoe) и инверсного (nCHyoe) выходов задается в полях SELOE[1:0] и NSELOE[1:0]

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

20.6 Блок цифрового фильтра

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

Конфигурация фильтра для входа ETR выполняется в поле ETR_FLTR[3:0] регистра BRKETR_CNTRL, для входов каналов CHyi – в поле CH_FLTR[3:0] регистра CHy_CNTRL0. Значение в данных полях позволяет настроить два параметра фильтра:

– частота выборки F_S, на которой входной сигнал захватывается в сдвиговый регистр для накопления. В качестве частоты F_S может использоваться частота TIM_CLK или F_{DTS};

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

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

Таким образом, настраивая частоту F_S и количество выборок N, задается минимальная длительность импульсов входного сигнала, которые не будут отфильтрованы. Диаграмма работы фильтра при использовании частоты TIM_CLK приведена на рисунке 53, частота F_S = TIM_CLK, количество выборок N = 4.

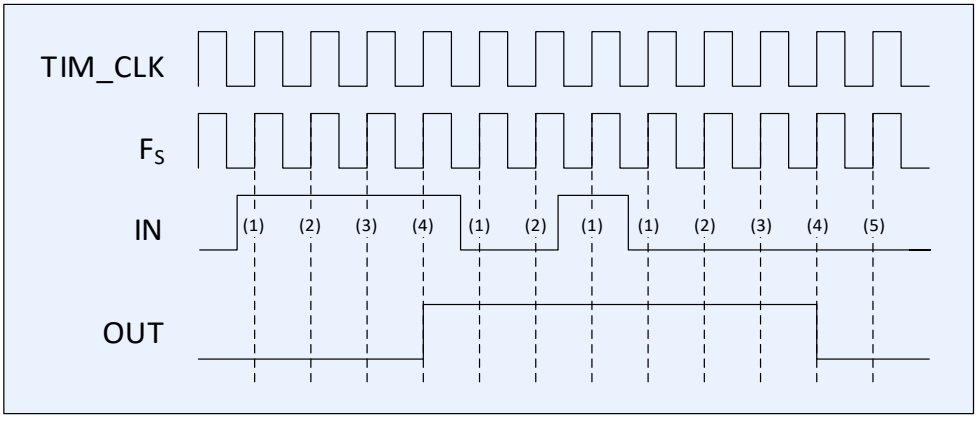


Рисунок 53 – Диаграмма работы фильтра, F_S = TIM_CLK, N = 4

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

Диаграмма работы фильтра при использовании частоты F_{DTS} приведена на рисунке 54, частота выборки $F_S = F_{DTS}/2$, количество выборок N = 6.

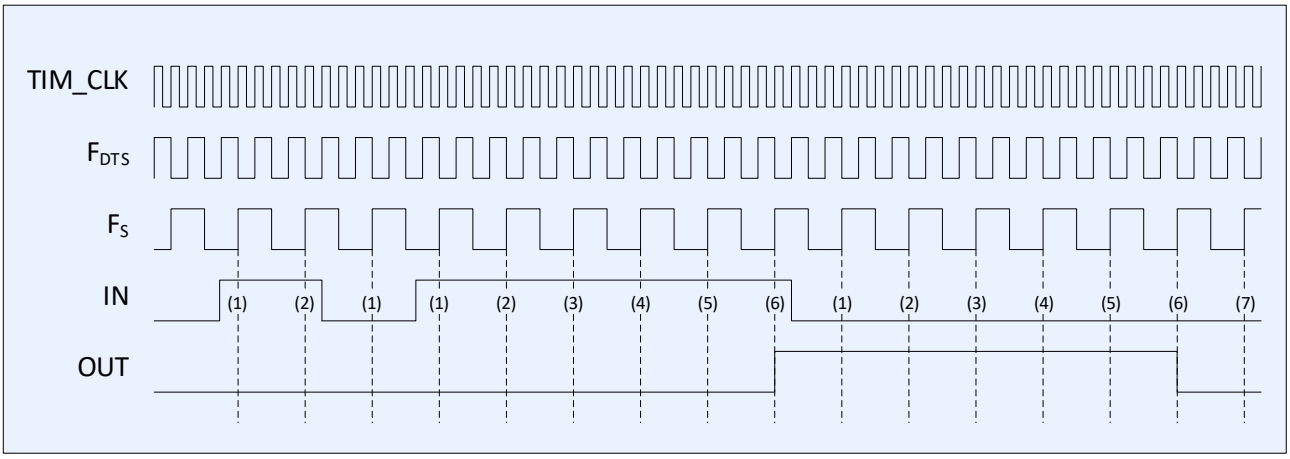


Рисунок 54 – Диаграмма работы фильтра, $F_S = F_{DTS}/2$, N = 6

Возможные варианты настройки блока фильтра приведены в таблице 182.

Таблица 182 – Возможные конфигурации фильтра

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

20.7 Флаги состояний, прерывания и запросы DMA

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

– CNT_ZERO_EVNT – совпадение значения счетчика CNT с нулем;

– CNT_ARR_EVNT – совпадение значения счетчика CNT со значением в регистре ARR;

– ETR_RE_EVNT – фиксация переднего фронта на входе ETR;

– ETR_FE_EVNT – фиксация заднего фронта на входе ETR;

– BRK_EVNT – фиксация высокого уровня на входе BRK;

– CCR_CAP_EVNT[3:0] – запись значения счетчика CNT в регистр CHy_CCR по захвату настроенного фронта на входе канала CHyi, события формируются индивидуально для каждого канала;

– CCR_REF_EVNT[3:0] – фиксация переднего фронта на выходе генератора опорного сигнала REF, события формируются индивидуально для каждого канала;

– CCR_CAP1_EVNT[3:0] – запись значения счетчика CNT в регистр CHy_CCR1 по захвату настроенного фронта на входе канала CHyi, события формируются индивидуально для каждого канала.

20.7.1 Флаги состояний

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

20.7.2 Прерывания

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

20.7.3 Запросы DMA

На основе отслеживаемых событий блок таймера формирует сигнал запроса DMA TMRxDMAREQ. Выбор событий, формирующих запрос DMA, осуществляется через регистр DMA_RE. Запросы к блоку DMA после конфигурации регистра DMA_RE формируются в том числе, когда блок выключен (обратить внимание на событие CNT_ZERO_EVNT: по умолчанию CNT = 0).

20.8 **Примеры**

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

20.8.1 **Обычный счетчик**

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

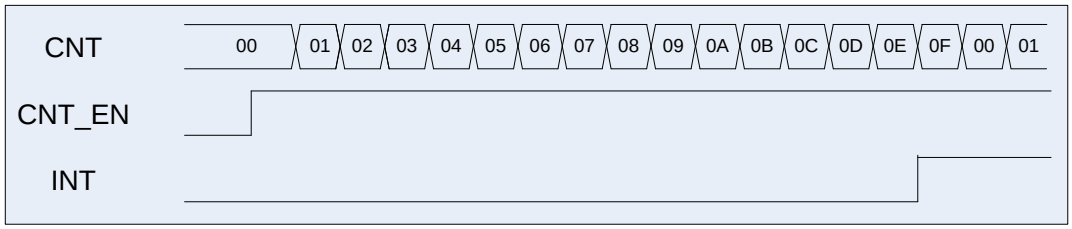


Рисунок 55 – Режим обычного счетчика

20.8.2 **Режим захвата**

```
RST_CLK->PER2_CLOCK |= 0x00004000; //Разрешение частоты PCLK для таймера 1
RST_CLK->TIM_CLOCK = 0x01000000;    //Настройка и подача частоты TIM_CLK
                                    //для таймера 1 (TIM_CLK = PER1_C2)
TIMER1->CNTRL = 0x00000000;  //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER1->CNT = 0x00000000;   //Начальное значение счетчика
TIMER1->PSG = 0x00000000;   //Предделитель частоты TIM_CLK
TIMER1->ARR = 0x000000FF;  //Основание счета
TIMER1->IE = 0x000001E0;      //Разрешение генерировать прерывание по событию
                              //настроенного фронта на входах CH1i-CH4i
TIMER1->CH1_CNTRL0 = 0x00008000;   //Захват по положительному фронту сигнала
                                   //на входе CH1i, фильтрация отключена
TIMER1->CH2_CNTRL0 = 0x00008010;   //Захват по отрицательному фронту сигнала
                                   //на входе CH2i, фильтрация отключена
TIMER1->CH3_CNTRL0 = 0x00008001;   //Захват по положительному фронту сигнала
                                   //на входе CH3i, фильтрация выполняется
                                  //по 2 выборкам на частоте TIM_CLK
TIMER1->CH4_CNTRL0 = 0x00008011;   //Захват по отрицательному фронту сигнала
                                   //на входе CH4i, фильтрация выполняется
                                  //по 2 выборкам на частоте TIM_CLK
//Режим работы выхода канала – канал работает на вход
TIMER1->CH1_CNTRL1= 0x00000000;
TIMER1->CH2_CNTRL1= 0x00000000;
```

```
TIMER1->CH3_CNTRL1= 0x00000000;
TIMER1->CH4_CNTRL1= 0x00000000;
//Разрешение работы таймера
TIMER1->CNTRL = 0x00000001;   //Счет прямой по TIM_CLK
```

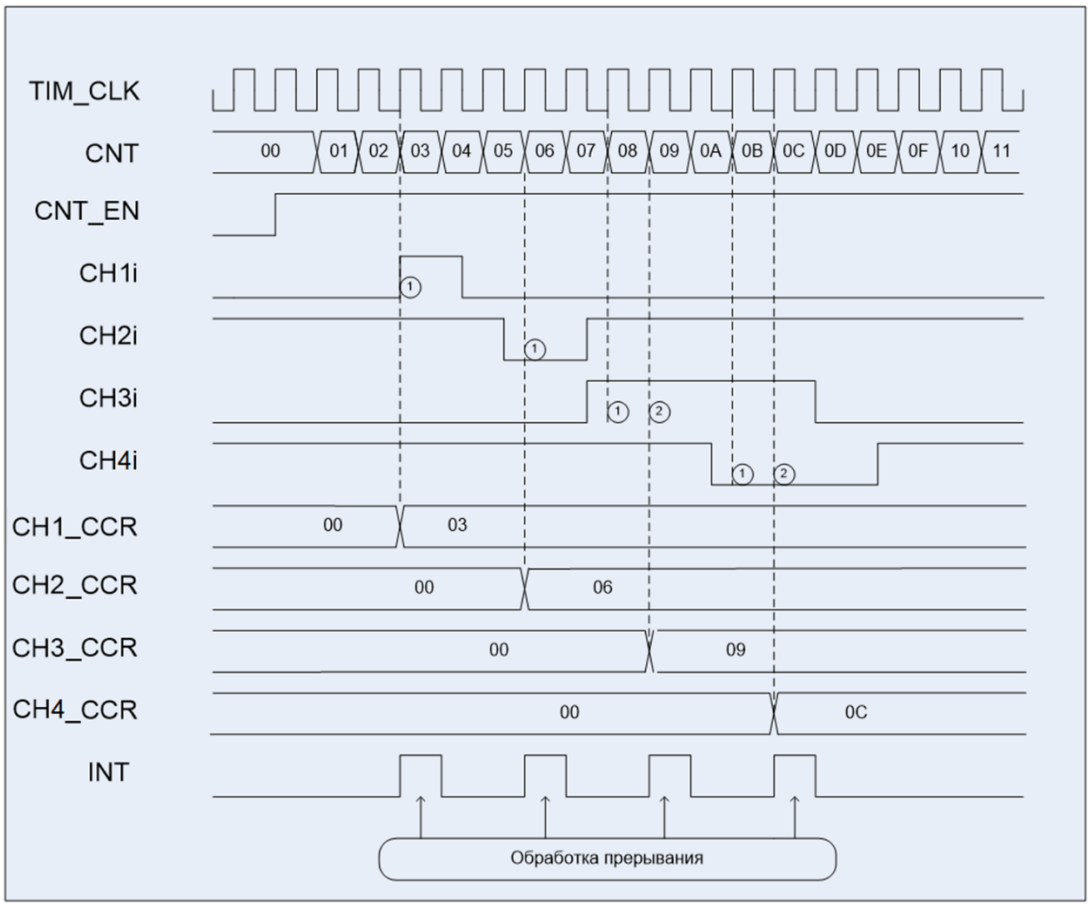


Рисунок 56 – Диаграммы примера работы в режиме захвата

20.8.3 **Режим ШИМ**

```
RST_CLK->PER2_CLOCK |= 0x00004000; //Разрешение частоты PCLK для таймера 1
RST_CLK->TIM_CLOCK = 0x01000000;   //Настройка и подача частоты TIM_CLK
                                   //для таймера 1 (TIM_CLK = PER1_C2)
TIMER1->CNTRL = 0x00000000;   //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER1->CNT = 0x00000000;     //Начальное значение счетчика
TIMER1->PSG = 0x00000000;     //Предделитель частоты TIM_CLK
TIMER1->ARR = 0x00000010;     //Основание счета
TIMER1->IE  = 0x00001E00;     //Разрешение генерировать прерывание по событию
                              //переднего фронта на выходе REF для всех каналов
//Режим работы каналов – ШИМ
TIMER1->CH1_CNTRL0 = 0x00000200;   //REF = 1, если CNT == CCR
TIMER1->CH2_CNTRL0 = 0x00000200;   //REF = 1, если CNT == CCR
TIMER1->CH3_CNTRL0 = 0x00000400;   //REF = 0, если CNT == CCR
TIMER1->CH4_CNTRL0 = 0x00000600;   //Переключение REF, если CNT == CCR
//Режим работы выхода канала – канал работает на выход,
//на выходы канала выдается сигнал REF
TIMER1->CH1_CNTRL1= 0x00000909;
```

```
TIMER1->CH2_CNTRL1= 0x00000909;
TIMER1->CH3_CNTRL1= 0x00000909;
TIMER1->CH4_CNTRL1= 0x00000909;
//Установка значений CCR, с которыми сравнивается CNT при работе в режиме ШИМ
TIMER1->CH1_CCR = 0x00000003;
TIMER1->CH2_CCR = 0x00000006;
TIMER1->CH3_CCR = 0x00000009;
TIMER1->CH4_CCR = 0x0000000F;
//Разрешение работы таймера
TIMER1->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
```

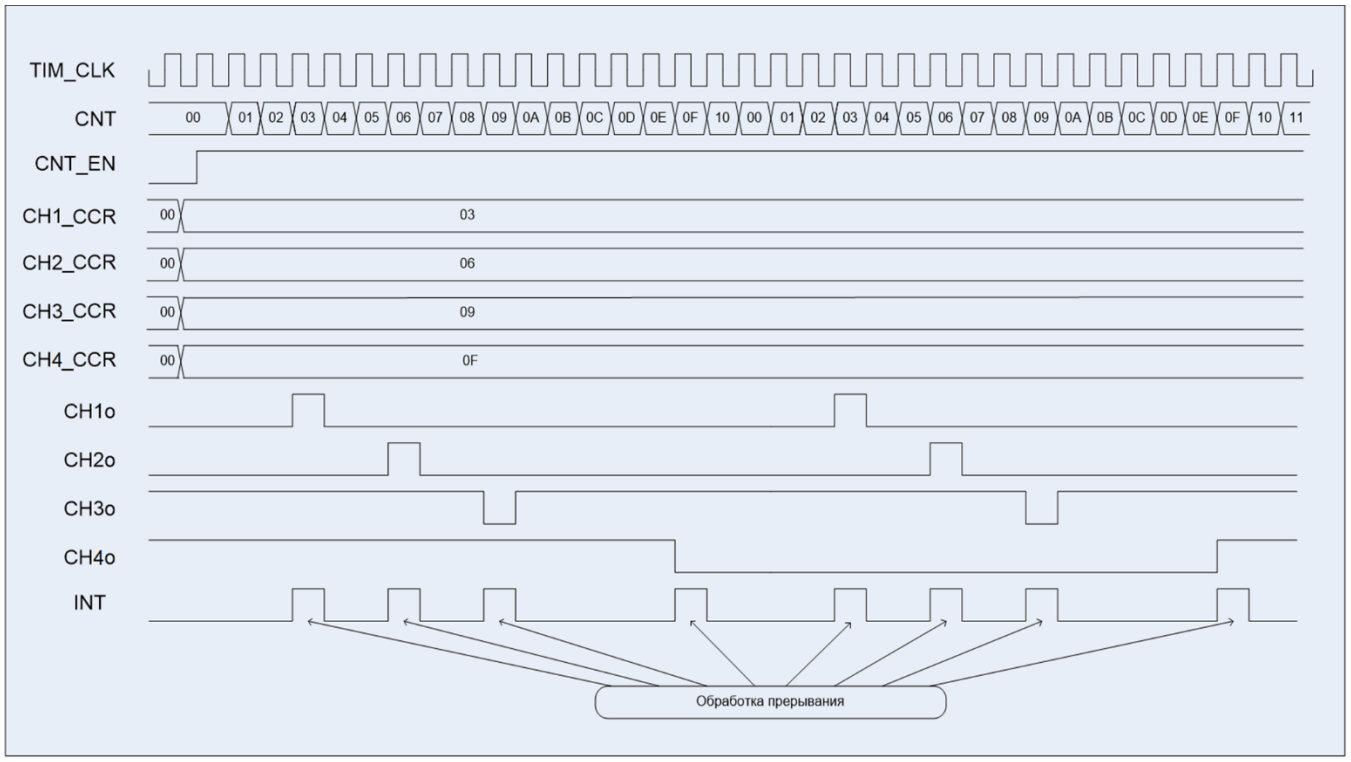


Рисунок 57 – Диаграммы примера работы в режиме ШИМ

20.9 **Описание регистров блока таймера**

Таблица 183 – Базовые адреса и смещения регистров управления таймера

Адрес	Название	Описание
0x5007_0000	TIMER1	Контроллер TIMER1
0x5007_8000	TIMER2	Контроллер TIMER2
0x500D_8000	TIMER3	Контроллер TIMER3
0x500E_0000	TIMER4	Контроллер TIMER4
Смещение		
0x00	CNT[31:0]	Основной счетчик таймера
0x04	PSG[31:0]	Делитель частоты TIM_CLK для тактирования основного счетчика
0x08	ARR[31:0]	Основание счета основного счетчика
0x0C	CNTRL[31:0]	Регистр управления основным счетчиком
0x50	BRKETR_CNTRL[31:0]	Регистр управления входом BRK и ETR
0x54	STATUS[31:0]	Регистр статуса таймера
0x58	IE[31:0]	Регистр разрешения прерываний
0x5C	DMA_RE[31:0]	Регистр разрешения формирования запроса DMA
Канал 1		
0x10	CH1_CCR[31:0]	Регистр сравнения/захвата для 1 канала таймера
0x20	CH1_CNTRL0[31:0]	Регистр управления 0 для 1 канала таймера
0x30	CH1_CNTRL1[31:0]	Регистр управления 1 для 1 канала таймера
0x40	CH1_DTG[31:0]	Регистр управления DTG для 1 канала таймера
0x60	CH1_CNTRL2[31:0]	Регистр управления 2 для 1 канала таймера
0x70	CH1_CCR1[31:0]	Регистр сравнения/захвата 1 для 1 канала таймера
Канал 2		
0x14	CH2_CCR[31:0]	Регистр сравнения/захвата для 2 канала таймера
0x24	CH2_CNTRL0[31:0]	Регистр управления 0 для 2 канала таймера
0x34	CH2_CNTRL1[31:0]	Регистр управления 1 для 2 канала таймера
0x44	CH2_DTG[31:0]	Регистр управления DTG для 2 канала таймера
0x64	CH2_CNTRL2[31:0]	Регистр управления 2 для 2 канала таймера
0x74	CH2_CCR1[31:0]	Регистр сравнения/захвата 1 для 2 канала таймера
Канал 3		
0x18	CH3_CCR[31:0]	Регистр сравнения/захвата для 3 канала таймера
0x28	CH3_CNTRL0[31:0]	Регистр управления 0 для 3 канала таймера
0x38	CH3_CNTRL1[31:0]	Регистр управления 1 для 3 канала таймера
0x48	CH3_DTG[31:0]	Регистр управления DTG для 3 канала таймера
0x68	CH3_CNTRL2[31:0]	Регистр управления 2 для 3 канала таймера
0x78	CH3_CCR1[31:0]	Регистр сравнения/захвата 1 для 3 канала таймера
Канал 4		
0x1C	CH4_CCR[31:0]	Регистр сравнения/захвата для 4 канала таймера
0x2C	CH4_CNTRL0[31:0]	Регистр управления 0 для 4 канала таймера
0x3C	CH4_CNTRL1[31:0]	Регистр управления 1 для 4 канала таймера
0x4C	CH4_DTG[31:0]	Регистр управления DTG для 4 канала таймера
0x6C	CH4_CNTRL2[31:0]	Регистр управления 2 для 4 канала таймера
0x7C	CH4_CCR1[31:0]	Регистр сравнения/захвата 1 для 4 канала таймера

20.9.1 **CNT**

Таблица 184 – Основной счетчик таймера CNT

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

Таблица 185 – Описание бит регистра CNT

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

20.9.2 **PSG**

Таблица 186 – Делитель частоты TIM_CLK для счета основного счетчика PSG

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

Таблица 187 – Описание бит регистра PSG

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

20.9.3 **ARR**

Таблица 188 – Основание счета основного счетчика ARR

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

Таблица 189 – Описание бит регистра ARR

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

20.9.4 **CNTRL**

Таблица 190 – Регистр управления основным счетчиком CNTRL

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

Номер	7…6	5…9	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	00	00	0	0	0	0
	CNT_MODE[1:0]	FDTS[1:0]	DIR	WR_CMPL	ARRB_EN	CNT_EN

Таблица 191 – Описание бит регистра CNTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…12	-	Зарезервировано.
11...8	EVNT_SEL[3:0]	Биты выбора источника событий: 0000 – внутренняя тактовая частота TIM_CLKd (формируется путем деления частоты TIM_CLK); 0001 – CNT == ARR в таймере 1; 0010 – CNT == ARR в таймере 2; 0011 – CNT == ARR в таймере 3; 0100 – событие переднего фронта на канале 1, «Режим 1»; 0101 – событие переднего фронта на канале 2, «Режим 1»; 0110 – событие переднего фронта на канале 3, «Режим 1»; 0111 – событие переднего фронта на канале 4, «Режим 1»; 1000 – событие переднего фронта на ETR, «Режим 2»; 1001 – событие заднего фронта на ETR, «Режим 2»; 1010 – CNT == ARR в таймере 4; 1011 – 1111 – зарезервировано
7..6	CNT_MODE[1:0]	Режим счета основного счетчика: 00 – счетчик прямой при DIR = 0; счетчик обратный при DIR = 1; 01 – счетчик двунаправленный с автоматическим изменением DIR при CNT == 0 или CNT == ARR; 10 – счетчик прямой при DIR = 0; счетчик обратный при DIR = 1; 11 – зарезервировано. Режим счета CNT_MODE[1:0] необходимо устанавливать в соответствии со значением в поле EVNT_SEL[3:0]: – EVNT_SEL[3:0] = 0000: CNT_MODE[1:0] = 00 или 01; – EVNT_SEL[3:0] != 0000: CNT_MODE[1:0] = 10
5...4	FDTS[1:0]	Делитель тактовой частоты F_{DTS}: 00 – F_{DTS} = TIM_CLK; 01 – F_{DTS} = TIM_CLK/2; 10 – F_{DTS} = TIM_CLK/3; 11 – F_{DTS} = TIM_CLK/4

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	DIR	Направление счета основного счетчика: 0 – прямой, от 0 до ARR; 1 – обратный, от ARR до 0
2	WR_CMPL	Флаг выполнения записи нового значения в регистры CNT, PSG и ARR: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
1	ARRB_EN	Режим обновления регистра ARR: 0 – ARR будет перезаписан в момент записи в ARR; 1 – ARR будет перезаписан при CNT == ARR
0	CNT_EN	Разрешение работы таймера: 0 – таймер отключен; 1 – таймер включен

20.9.5 **CHy_CCR**

Таблица 192 – Регистр сравнения/захвата для ‘y’ канала таймера CHy_CCR

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

Таблица 193 – Описание бит регистра CHy_CCR

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

20.9.6 **CHy_CCR1**

Таблица 194 – Регистр сравнения/захвата для ‘y’ канала таймера CHy_CCR1

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

Таблица 195 – Описание бит регистра CHy_CCR1

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

20.9.7 **CHy_CNTRL0**

Таблица 196 – Регистр управления для ‘y’ канала таймера CHy_CNTRL0

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

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

Таблица 197 – Описание бит регистра CHy_CNTRL0

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16	WR_CMPL1	Флаг выполнения записи нового значения в регистр CHy_CCR1: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
15	CAP_NPWM	Режим работы канала: 0 – канал работает в режиме ШИМ; 1 – канал работает в режиме захвата
14	WR_CMPL	Флаг выполнения записи нового значения в регистр CHy_CCR: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
13	ETR_EN	Разрешение сброса сигнала REF в «0» при высоком уровне на входе ETR: 0 – запрещен; 1 – разрешен
12	BRK_EN	Разрешение сброса сигналов REF и DTG в «0» при низком уровне на входе BRK: 0 – запрещен; 1 – разрешен
11…9	OCCM[2:0]	Формат выработки сигнала REF в режиме ШИМ: Если CCR1_EN=0: 000 – всегда 0; 001 – 1, если CNT==CCR; 010 – 0, если CNT==CCR; 011 – переключение REF, если CNT==CCR; 100 – всегда 0; 101 – всегда 1; 110 – 1, если DIR=0 (счет прямой), CNT<CCR, иначе 0; 0, если DIR=1 (счет обратный), CNT>CCR, иначе 1; 111 – 0, если DIR=0 (счет прямой), CNT<CCR, иначе 1; 1, если DIR=1 (счет обратный), CNT>CCR, иначе 0. Если CCR1_EN=1: 000 – всегда 0; 001 – 1, если CNT==CCR или CNT==CCR1;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		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<CN <CCR1, иначе 0; Необходимо соблюдать условие CCR<CCR1
8	OCCE	Разрешение работы ETR: 0 – запрещен; 1 – разрешен
7…6	CH_PSC[1:0]	Предварительный делитель входного канала: 00 – нет деления; 01 – /2; 10 – /4; 11 – /8
5…4	CH_SEL[1:0]	Выбор события по входному каналу CHyi для фиксации значения основного счетчика (регистр CNT) в регистр CHy_CCR: 00 – положительный фронт на входном канале CHyi; 01 – отрицательный фронт на входном канале CHyi; 10 – положительный фронт от других каналов: – для первого канала от второго канала; – для второго канала от третьего канала; – для третьего канала от четвертого канала; – для четвертого канала от первого канала; 11 – положительный фронт от других каналов: – для первого канала от третьего канала; – для второго канала от четвертого канала; – для третьего канала от первого канала; – для четвертого канала от второго канала
3…0	CH_FLTR[3:0]	Конфигурация фильтра на входе канала ‘y’. Выбор частоты выборки Fs 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

20.9.8 **CHy_CNTRL1**

Таблица 198 – Регистр управления 1 для ‘y’ 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 199 – 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 nCHyo выдается 0; 01 – на nCHyo выдается 1; 10 – на nCHyo выдается сигнал nREF; 11 – на nCHyo выдается сигнал с DTG
9…8	NSELOE[1:0]	Режим работы инверсного выхода nCHy: 00 – на nCHyoe выдается 0; 01 – на nCHyoe выдается 1; 10 – на nCHyoe выдается сигнал nREF; 11 – на nCHyoe выдается сигнал с DTG. При nCHyoe = 0 вывод канала в третьем состоянии, при nCHyoe = 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 CHyo выдается 0; 01 – на CHyo выдается 1; 10 – на CHyo выдается сигнал REF; 11 – на CHyo выдается сигнал с DTG
1…0	SELOE[1:0]	Режим работы прямого выхода CHy: 00 – на CHyoe выдается 0; 01 – на CHyoe выдается 1; 10 – на CHyoe выдается сигнал REF; 11 – на CHyoe выдается сигнал с DTG. При CHyoe = 0 вывод канала работает в режиме входа, при CHyoe = 1 вывод канала работает в режиме выхода

20.9.9 **CHy_CNTRL2**

Таблица 200 – Регистр управления 2 для ‘y’ канала таймера CHy_CNTRL2

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

Таблица 201 – Описание бит регистра CHy_CNTRL2

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

20.9.10 **CHy_DTG**

Таблица 202 – 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 203 – 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 DTGdel = DTG • (DTGx + 1)
7…5	-	Зарезервировано.
4	EDTS	Частота работы DTG: 0 – TIM_CLK; 1 – F_{DTS}
3…0	DTGx[3:0]	Предварительный делитель частоты DTG

20.9.11 **BRKETR_CNTRL**

Таблица 204 – Регистр BRKETR_CNTRL управления входом BRK и ETR

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

Таблица 205 – Описание бит регистра BRKETR_CNTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7…4	ETR_FLTR[3:0]	Конфигурация фильтра на входе ETR. Выбор частоты выборки Fs 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

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

20.9.12 **STATUS**

Таблица 206 – Регистр статуса таймера STATUS

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

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

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

Таблица 207 – Описание бит регистра STATUS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…17	-	Зарезервировано
16…13	CCR_CAP1_EVNT[3:0]	Событие записи значения счетчика CNT в регистр CHy_CCR1 по захвату настроенного фронта на входе канала CHyi: 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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
12…9	CCR_REF_EVNT[3:0]	Событие переднего фронта на выходе генератора опорного сигнала REF: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал; Бит 3 – четвертый канал
8…5	CCR_CAP_EVNT[3:0]	Событие записи значения счетчика CNT в регистр CHy_CCR по захвату настроенного фронта на входе канала CHyi: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал; Бит 3 – четвертый канал
4	BRK_EVNT	Событие высокого уровня на входе BRK: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», при условии наличия низкого уровня на входе BRK
3	ETR_FE_EVNT	Событие заднего фронта на входе ETR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
2	ETR_RE_EVNT	Событие переднего фронта на входе ETR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
1	CNT_ARR_EVNT	Событие совпадения CNT с ARR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса флага регистры CNT и ARR не изменили состояния, то флаг повторно не взводится

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
0	CNT_ZERO_EVNT	Событие совпадения CNT с нулем: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса флага регистр CNT не изменил состояния, то флаг повторно не взводится

20.9.13 **IE**

Таблица 208 – Регистр разрешения прерываний таймера IE

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

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

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

Таблица 209 – Описание бит регистра IE

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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
12…9	CCR_REF_EVNT_IE[3:0]	Флаг разрешения прерывания по событию CCR_REF_EVNT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
8…5	CCR_CAP_EVNT_IE[3:0]	Флаг разрешения прерывания по событию CCR_CAP_EVNT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
4	BRK_EVNT_IE	Флаг разрешения прерывания по событию BRK_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
3	ETR_FE_EVNT_IE	Флаг разрешения прерывания по событию ETR_FE_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
2	ETR_RE_EVNT_IE	Флаг разрешения прерывания по событию ETR_RE_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
1	CNT_ARR_EVNT_IE	Флаг разрешения прерывания по событию CNT_ARR_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
0	CNT_ZERO_EVNT_IE	Флаг разрешения прерывания по событию CNT_ZERO_EVNT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено

20.9.14 **DMA_RE**

Таблица 210 – Регистр DMA_RE разрешения запроса DMA

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

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

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

Таблица 211 – Описание бит регистра DMA_RE

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

21 Контроллер SAR АЦП

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

Контроллер АЦП позволяет:

– оцифровать внешний канал;

– оцифровать значение встроенного датчика температуры;

– оцифровать значение встроенного источника опорного напряжения;

– оцифровать значение напряжения батарейного домена;

– осуществить автоматический опрос заданных каналов;

– выработать прерывание при выходе оцифрованного значения за заданные пределы.

Для осуществления преобразования требуется не менее 25 тактов синхронизации C_ADCS. В качестве синхросигнала может выступать поделенная частота периферийного блока PCLKd или частота ADC_CLK, формируемая в блоке «Сигналы тактовой частоты». Выбор частоты осуществляется с помощью бита Cfg_REG_CLKS. Для получения частоты PCLKd в контроллере АЦП частота PCLK может быть поделена с помощью битов Cfg_REG_DIVCLK[3:0]. Максимальная частота C_ADCS не может превышать 28 МГц.

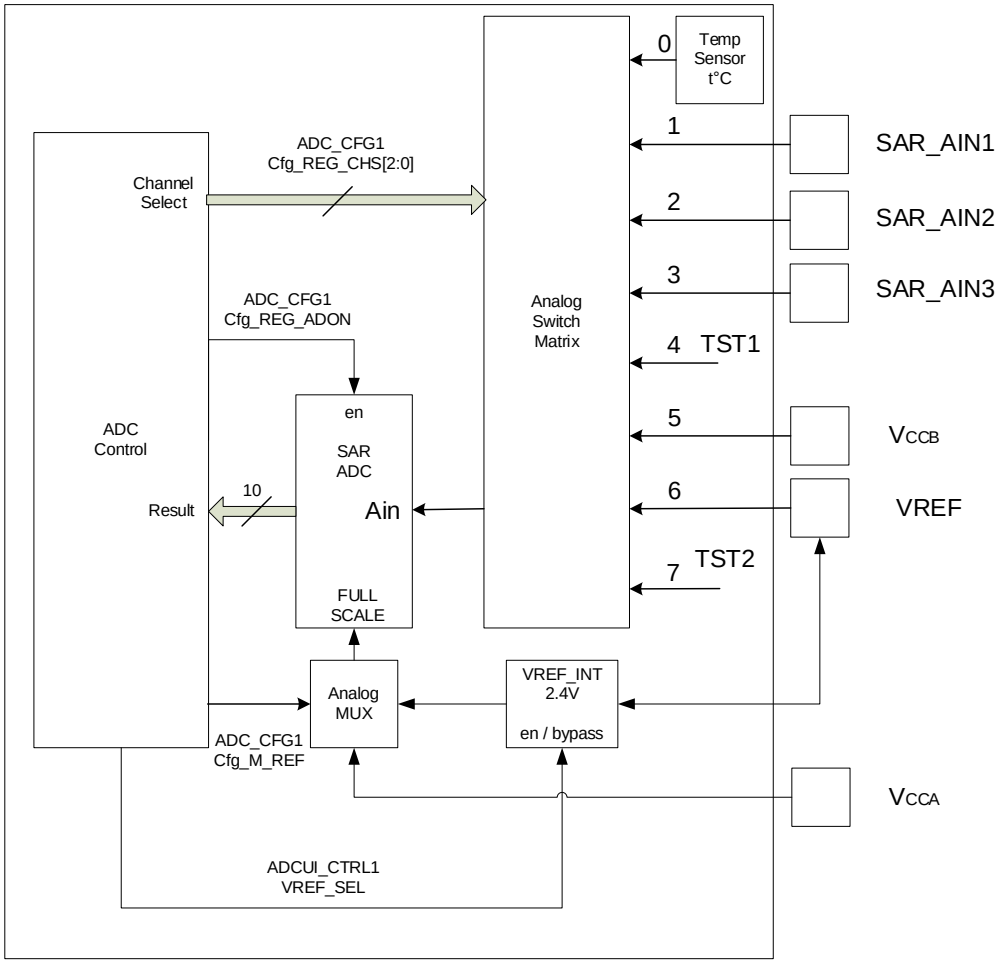


Рисунок 58 – Структурная схема АЦП

Для включения АЦП необходимо установить бит Cfg_REG_ADON.

21.1 **Преобразование внешнего канала**

В регистре ADC1_CFG в битах Cfg_REG_CHS[2:0] необходимо задать соответствующий выводу номер канала. Преобразование может осуществляться при опоре равной U_{CCA} (бит Cfg_M_REF = 0) и внешней (Cfg_M_REF = 1), в этом случая опора берется с вывода VREF. Биты Cfg_REG_CHCH, Cfg_REG_RNGC, Cfg_REG_SAMPLE должны быть сброшены.

Для начала преобразования необходимо записать 1 в бит Cfg_REG_GO. После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования. После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата 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.

21.2 Последовательное преобразование нескольких каналов

Для автоматического последовательного преобразования нескольких каналов или одного канала в регистре ADC1_CHSEL необходимо установить единицы в битах, соответствующих необходимым для преобразования каналам. Выставление данных бит необходимо обеспечить до установки конфигурации АЦП, то есть до записи в регистр ADC1_CFG. Преобразование может осуществляться при внутренней опоре (бит Cfg_M_REF = 0) и внешней (Cfg_M_REF = 1), в этом случая опора берется с выводов VREF. Бит Cfg_REG_RNGC должен быть сброшен, а Cfg_REG_SAMPLE и Cfg_REG_CHCH должны быть установлены. С помощью бит DelayGO можно задать паузу между преобразованиями при переборе каналов. При включенном АЦП (Cfg_REG_ADON = 1) и установленном бите Cfg_REG_SAMPLE = 1 процесс преобразования запускается автоматически и значение бита Cfg_REG_GO контролируется аппаратно.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования. После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Флаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.

Для последовательного преобразования одного и того же канала можно в регистре ADC1_CHSEL выбрать только один канал и установить бит Cfg_REG_CHCH в 1, либо установить номер канала в битах Cfg_REG_CHS[2:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. Последовательное преобразование значения датчика температуры и источника опорного напряжения могут выполняться только в режиме последовательного преобразования одного канала.

21.3 Преобразование с контролем границ

При необходимости отслеживать нахождение оцифрованных значений в допустимых пределах можно задать нижнюю и верхнюю допустимые границы в регистрах ADC1_L_LEVEL и ADC1_H_LEVEL. При этом, если установлен бит Cfg_REG_RNGC, то в случае, когда результат преобразования выходит за границы выставляется флаг Flg_REG_AWOIFEN. А в регистре результата будет полученное значение.

Результат преобразования вычисляется по формуле

$$\text{напряжение на входе АЦП} = \frac{\text{напряжение полной шкалы АЦП} \cdot \text{значение регистра } ADC1_RESULT}{1024}. \quad (6)$$

21.4 Датчик температуры

С помощью АЦП можно осуществить преобразования датчика температуры. Для выбора датчика температуры в качестве источника для преобразования необходимо в битах Cfg_REG_CHS установить значение 0 канала. После чего можно запустить процесс преобразования. Для начала преобразования необходимо записать 1 в бит Cfg_REG_GO.

После завершения преобразования будет взведен бит Flg_REG_EOCIF в регистре ADC1_STATUS. А в регистре ADC1_RESULT будет результат преобразования. После считывания результата бит Flg_REG_EOCIF сбросится.

Если после первого преобразования результат не был считан и было выполнено второе преобразование, то в регистре результата ADC1_RESULT будет значение от последнего преобразования, а помимо бита Flg_REG_EOCIF будет взведен бит Flg_REG_OVERWRITE. Флаг Flg_REG_OVERWRITE может быть сброшен только записью в регистр ADC1_STATUS.

Для последовательного преобразования только датчика температуры можно в регистре ADC1_CHSEL выбрать только 0 канал и установить бит Cfg_REG_CHCH в 1, либо установить номер 0-го канала в битах Cfg_REG_CHS[2:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала.

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

21.4.1 Формула расчета температуры

Температура кристалла вычисляется по формуле

$$TEMP = 25 - \frac{\frac{\text{напряжение полной шкалы АЦП} \cdot \text{значение регистра } ADC1_RESULT}{1024} - 1{,}403}{0{,}0037}. \qquad (7)$$

Например, при опросе 0 канала АЦП выдает код = 625, в качестве полной шкалы при этом выбран внешний источник 2,5 В, подключенный ко входу микросхемы VREF. Тогда температура кристалла равна

$$TEMP = 25 - \frac{\frac{2{,}5 \cdot 625}{1024} - 1{,}403}{0{,}0037} = -8{,}2\ ℃. \qquad (8)$$

21.5 Время заряда внутренней емкости

Процесс преобразования состоит из двух этапов: сначала происходит заряд внутренней емкости до уровня внешнего сигнала, и затем происходит преобразование уровня заряда внутренней емкости в цифровой вид. Таким образом, для точного преобразования внешнего сигнала в цифровой вид, за время первого этапа внутренняя

емкость должна зарядиться до уровня внешнего сигнала. Это время определяется соотношением номинальной внутренней емкости, входным сопротивлением тракта АЦП и выходным сопротивлением источника сигнала. Приведенная ниже формула позволяет определить максимальное выходное сопротивление источника R_{AIN} для обеспечения качественного преобразования.

$$R_{AIN} < \frac{T_{track}}{C_{ADC} \cdot \ln(2^N)} - R_{ADC}\,, \tag{9}$$

где C_{ADC} – внутренняя емкость АЦП (~ 2 – 3 пФ);
N – требуемая точность в разрядах;
R_{ADC} – входное сопротивление тракта АЦП (~1500 Ом);
T_{track} – время заряда внутренней емкости в тактах, определяется по формуле

$$T_{track} = 4 \cdot T_{C_ADCS} + N_{PCLKd} \cdot T_{PCLKd} = \frac{4}{f_{C_ADCS}} + \frac{(ConvDelay)}{f_{PCLKd}}\,, \tag{10}$$

где ConvDelay – дополнительная задержка перед началом преобразования, определяется битами DelayGo[2:0] исходя из таблиц 212 – 214;
f_{C_ADCS} – рабочая частота SAR АЦП;
f_{PCLKd} – определяется формулой

$$f_{PCLKd} = \frac{f_{PCLK}}{2^{Cfg\ REG\ DIVCLK}}\,. \tag{11}$$

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 таблицах 212 – 214. Время заряда T_{track} должно составлять не менее 1 мкс.

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

Таблица 212 – Время заряда внутренней емкости АЦП и время преобразования. C_ADCS = ADC_CLK/PCLKd (Cfg_REG_CLKS = 1/0), ADC_CLK = PCLKd = PCLK

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

Таблица 213 – Время заряда внутренней емкости АЦП и время преобразования. C_ADCS = PCLKd (Cfg_REG_CLKS = 0)

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

Таблица 214 – Время заряда внутренней емкости АЦП и время преобразования. C_ADCS = ADC_CLK (Cfg_REG_CLKS = 1)

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

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

Для корректного задания режимов работы АЦП в регистре ADC1_CFG необходимо сделать до задания бита Cfg_REG_GO, иначе новая конфигурация будет действовать со следующего преобразования.

21.7 **Описание регистров блока контроллера АЦП**

Таблица 215 – Описание регистров блока контроллера АЦП

Базовый Адрес	Название	Описание
0x5004_0000	ADC	Контроллер ADC
Смещение		
0x00	ADC1_CFG	Регистр управления ADC
0x08	ADC1_H_LEVEL	Регистр верхней границы ADC
0x10	ADC1_L_LEVEL	Регистр нижней границы ADC
0x18	ADC1_RESULT	Регистр результата ADC

Базовый Адрес	Название	Описание
0x5004_0000	ADC	Контроллер ADC
Смещение		
0x20	ADC1_STATUS	Регистр статуса ADC
0x28	ADC1_CHSEL	Регистр выбора каналов перебора ADC
0x30	ADC_TRIM	Регистр настройки ИОН

21.7.1 **ADC1_CFG**

Таблица 216 – Регистр ADC1_CFG

Номер	31…28	27…25	24…19
Доступ	U	R/W	U
Сброс	0	0	0
	-	DelayGo[2:0]	-

Номер	18..16	15…12	11	10	9
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	-	Cfg_REG_DIVCLK [3:0]	Cfg_M_ REF	Cfg_REG_RNGC	Cfg_REG_CHC H

Номер	8, 7	6…4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	-	Cfg_REG_ CHS[2:0]	Cfg_REG_ SAMPLE	Cfg_REG_ CLKS	Cfg_REG_ GO	Cfg_REG_ ADON

Таблица 217 – Описание бит регистра ADC1_CFG

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...28	-	Зарезервировано
27…25	DelayGo[2:0]	Задержка перед началом следующего преобразования после завершения предыдущего при последовательном переборе каналов: см. таблицы 212 – 214
24...19	-	Должны быть в нуле для правильного функционирования
18…16	-	Должно быть 010 для правильного функционирования
15...12	Cfg_REG_DIVCLK[3:0]	Выбор коэффициента деления частоты периферийного блока: 0000 – PCLKd = PCLK; 0001 – PCLKd = PCLK/2; 0010 – PCLKd = PCLK/4; 0011 – PCLKd = PCLK/8; … 1011 – PCLKd = PCLK/2048. Остальные значения: PCLKd = PCLK

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
11	Cfg_M_REF	Выбор шкалы АЦП: 0 – шкала АЦП от 0 до напряжения U_{CCA}; 1 – шкала АЦП от 0 до напряжения VREF
10	Cfg_REG_RNGC	Разрешение автоматического контролирования уровней: 1 – разрешено: выработка прерывания при выходе за диапазон в регистрах границы обработки; 0 – не разрешено
9	Cfg_REG_CHCH	Выбор переключения каналов: 1 – переключение включено (перебираются каналы, выбранные в регистре выбора канала Sl_Ch_Ch_REF); 0 – используется только выбранный канал
8, 7	-	Должны быть в нуле для правильного функционирования
6…4	Cfg_REG_CHS[2:0]	Выбор аналогового канала, по которому поступает сигнал для преобразования: 111 – 0 канал (термодатчик); 110 – 1 канал (подключение к SAR_AIN1); 101 – 2 канал (подключение к SAR_AIN2); 100 – 3 канал (подключение к SAR_AIN3); 011 – 4 канал (тестовый режим); 010 – 5 канал (подключение к VDD_BD); 001 – 6 канал (подключение к VREF); 000 – 7 канал (тестовый режим)
3	Cfg_REG_SAMPLE	Выбор способа запуска АЦП: 1 – последовательный: автоматический запуск после завершения предыдущего преобразования; 0 – одиночный
2	Cfg_REG_CLKS	Выбор источника синхросигнала CLK работы ADC: 1 – ACLK (формируется от выбранной частоты, подробнее см. раздел «Сигналы тактовой частоты»); 0 – PCLKd (формируется от частоты ядра, подробнее см. формулу (11))
1	Cfg_REG_GO	Начало преобразования. Запись «1» начинает процесс преобразования, сбрасывается автоматически. Запись при Cfg_REG_SAMPLE = 1 не влияет на работу АЦП, бит устанавливается и сбрасывается автоматически
0	Cfg_REG_ADON	Включение АЦП: 1 – включено; 0 – выключено

21.7.2 **ADC1_H_LEVEL**

Таблица 218 – Регистр ADC1_H_LEVEL

Номер	31…10	9…0
Доступ	U	R/W
Сброс	0	0
	-	REG_H_LEVEL[9:0]

Таблица 219 – Описание бит регистра ADC1_H_LEVEL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…10	-	Зарезервировано
9…0	REG_H_LEVEL[9:0]	Верхняя граница зоны допуска

21.7.3 **ADC1_L_LEVEL**

Таблица 220 – Регистр ADC1_L_LEVEL

Номер	31…10	9…0
Доступ	U	R/W
Сброс	0	0
		REG_L_LEVEL[9:0]

Таблица 221 – Описание бит регистра ADC1_L_LEVEL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..10	-	Зарезервировано
9…0	REG_L_LEVEL[9:0]	Нижняя граница зоны допуска

21.7.4 **ADC1_RESULT**

Таблица 222 – Регистр ADC1_RESULT

Номер	31…19	18…16	15…10	9…0
Доступ	U	RO	U	RO
Сброс	0	0	0	0
	-	CHANNEL[2:0]	-	RESULT[9:0]

Таблица 223 – Описание бит регистра ADC1_RESULT

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...19	-	Зарезервировано

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
18...16	CHANNEL[2:0]	Канал результата преобразования: 111 – 0 канал (термодатчик); 110 – 1 канал (подключение к SAR_AIN1); 101 – 2 канал (подключение к SAR_AIN2); 100 – 3 канал (подключение к SAR_AIN3); 011 – 4 канал (тестовый режим); 010 – 5 канал (подключение к VDD_BD); 001 – 6 канал (подключение к VREF); 000 – 7 канал (тестовый режим)
15...10	-	Зарезервировано
9…0	RESULT[9:0]	Значение результата преобразования

21.7.5 **ADC1_STATUS**

Таблица 224 – Регистр ADC1_STATUS

Номер	31…5	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	-	ECOIF_IE	AWOIF_IE	Flg_REG_EOCIF	Flg_REG_AWOIFEN	Flg_REG_OVERWRITE

Таблица 225 – 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 ADC1_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. Сбрасывается только при записи нуля в данный бит. 0 – не было события перезаписи не считанного результата; 1 – был результат преобразования, который не был считан

21.7.6 **ADC1_CHSEL**

Таблица 226 – Регистр ADC1_CHSEL

Номер	31…8	7…0
Доступ	U	R/W
Сброс	0	0
	-	Sl_Ch_Ch_REF[7:0]

Таблица 227 – Описание бит регистра ADC1_CHSEL

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…8	-	Должны быть в нуле для правильного функционирования
7...0	Sl_Ch_Ch_REF[7:0]	Выбор каналов автоматического перебора: 0 – в соответствующем бите канал не участвует в переборе; 1 – в соответствующем бите канал участвует в переборе. Номера каналов, участвующих в переборе, соответствует номерам установленных бит в регистре

21.7.7 **ADC_TRIM**

Таблица 228 – Регистр ADC_TRIM

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

Таблица 229 – Описание бит регистра ADC_TRIM

Разряды	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…6	-	Зарезервировано
5…1	BG_TRIM[4:0]	Подстройка опорного напряжения bandgap. Рекомендуется использовать значение бит TRIM_BG записанных во Flash-памяти при производстве
0	-	Зарезервировано

21.8 **Выводы SAR_AINx**

В микросхеме реализованы специализированные выводы SAR_AINx, отличающиеся от остальных отстутствием защитных диодов на какое-либо питание. Допустимо задавать на них напряжения выше напряжения питания, но не выше указанных в таблице 438.

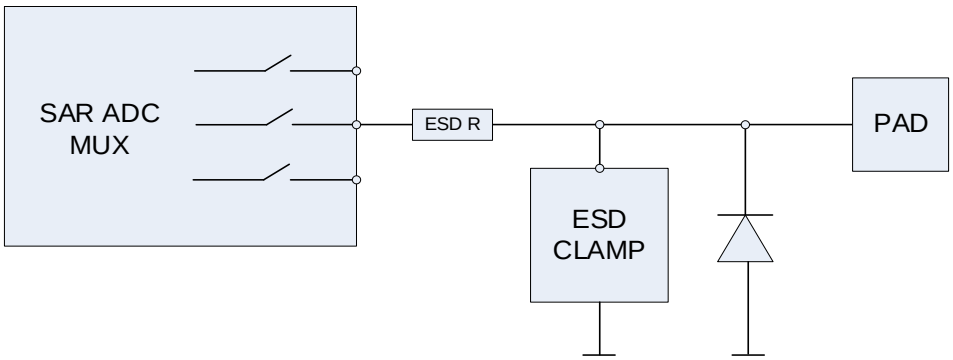


Рисунок 59 – Структурная схема выводов SAR_AINx

22 Контроллер интерфейса I2C

Интерфейс I2C является двухпроводным, двунаправленным последовательным каналом связи с простым и эффективным методом обмена данными между устройствами.

Контроллер интерфейса I2C в микросхеме работает только в режиме ведущего («мастера»). Реализована совместимость с многомастерными системами: предусмотрены механизмы детектирования коллизий и арбитража шины, предотвращающие потерю данных при обмене, когда два или более мастера пытаются осуществить передачу одновременно. Поддерживается работа с ведомыми устройствами, применяющими растяжение тактовой частоты SCL (clock stretching) для замедления скорости обмена данными.

22.1 Конфигурация контроллера I2C

Контроллер I2C формирует тактовый сигнал обмена данными SCL с использованием внутреннего предделителя частоты DIV, коэффициент деления которого задается в регистрах PRL (младшая часть) и PRH (старшая часть). Минимальное значение DIV равно 1. Если рассчитанное значение DIV не использует старшую часть, то регистр PRH должен быть установлен в 0. Настройка DIV должна выполняться только при выключенном контроллере I2C (EN_I2C=0).

Расчетная скорость обмена данными по интерфейсу I2C определяется по формуле:

$$F_{scl} = \frac{\mathrm{HCLK}}{5 \cdot (\mathrm{DIV} + 1)} \ . \qquad (12)$$

Необходимо учитывать, что контроллер I2C отслеживает фактическое состояние линии SCL. Если после завершения формирования низкого уровня контроллер отпускает линию SCL, но она остаётся в состоянии логического нуля (например, из-за растяжения SCL ведомым устройством или медленного фронта нарастания сигнала SCL), то начало следующего тактового цикла откладывается до перехода SCL в высокий уровень. Таким образом фактическая скорость обмена может оказаться ниже расчётной, определяемой формулой (12).

Разрешение работы контроллера I2C осуществляется установкой бита EN_I2C в регистре CMD. Выключение контроллера I2C (сброс бита EN_I2C) допустимо только после завершения обмена на шине, то есть после окончания передачи сигнала STOP. Нарушение данного требования может привести к блокировке шины I2C.

Запись в регистр управления CMD возможна только при разрешенной работе контроллера I2C (EN_I2C=1). Попытка записи в регистр CMD при неактивном состоянии контроллера I2C (EN_I2C=0) игнорируется.

22.2 Конфигурация системы

I2C-системы используют последовательную линию данных SDA и линию тактового сигнала SCL. Все устройства, подсоединенные к этим двум линиям, должны работать в режиме открытого стока, обеспечивая тем самым создание на линии «проводного И» за счет внешних резисторов подтяжки обоих линий к питанию.

Передача данных между мастером и ведомым осуществляется по линии SDA и синхронизируется по линии SCL. После завершения передачи информации осуществляется передача в обратную сторону одного бита подтверждения. Каждый принимаемый бит фиксируется принимающей стороной при высоком уровне SCL и может изменяться передатчиком при низком уровне. Изменение линии SDA при высоком уровне SCL является командным состоянием (см. «Сигнал START» и «Сигнал STOP»).

22.3 Протокол I2C

Типовая передача по интерфейсу I2C содержит четыре фазы:

– сигнал START;
– передача адреса;
– передача данных;
– сигнал STOP.

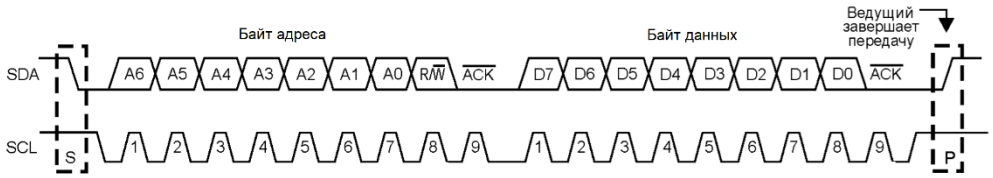


Рисунок 60 – Прием/передача по I2C в 7-битном режиме

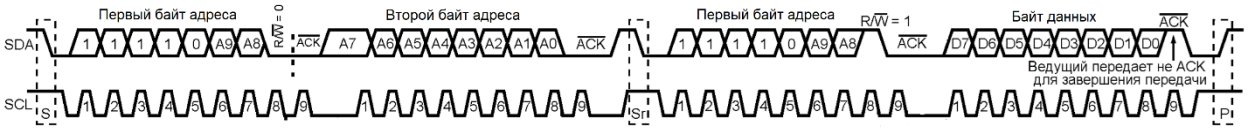


Рисунок 61 – Прием по I2C в 10-битном режиме

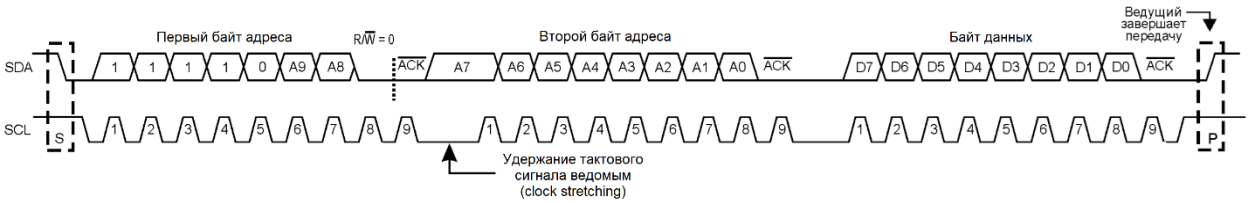


Рисунок 62 – Передача по I2C в 10-битном режиме

22.3.1 **Сигнал START**

Когда шина находится в свободном состоянии, то есть ни одно из устройств не осуществляет передачи (на линиях SCL и SDA высокий уровень), мастер может инициализировать процесс передачи через создание сигнала START на линии. Сигнал START или S-бит задается, когда уровень на линии SDA переходит из высокого в низкий при высоком уровне на линии SCL. Появление сигнала START не означает начала передачи данных.

Повторный сигнал START является обычным сигналом START, но без предварительно сгенерированного до этого сигнала STOP. Мастер может использовать метод для начала соединения с другим ведомым или с тем же ведомым, но с изменением режима работы (например, чтение после записи, или, наоборот) без перевода шины в свободное состояние.

Контроллер интерфейса генерирует сигнал START при записи единицы в бит START регистра CMD при установленных битах WR или RD. Для начала стандартного обмена необходимо установить биты START и WR в регистре CMD, что приведет к последовательной выдаче на линию сигнала START, а также адреса ведомого устройства и бита RW, которые записаны в регистр TXD. В зависимости от состояния линии SCL генерируется либо сигнал START, либо повторный сигнал START.

22.3.2 **Передача адреса**

Первым байтом данных, передаваемым мастером сразу после сигнала START, является адрес ведомого. Это 7-битный адрес и следующий за ним бит RW. Бит RW определяет дальнейшее направление передачи данных: 0 – запись данных в ведомое устройство, 1 – чтение данных из ведомого устройства. В системе на одной шине не может быть несколько ведомых устройств с одним адресом. Ведомое устройство, у которого совпадает адрес с адресом в сообщении, подтверждает прием, выставляя ACK и опуская линию SDA в низкий уровень на 9-й тактовый импульс SCL.

Процесс выдачи 7-битного адреса выполняется совместно с выдачей сигнала START. Необходимо записать адрес ведомого и бит RW в регистр TXD, после чего установить биты START и WR в регистре CMD. Контроллер осуществит передачу сигнала START и адреса в линию.

Контроллер также поддерживает передачу 10-битного адреса ведомого путем генерации нескольких циклов передачи адреса (см. рисунки 61 и 62). Сначала необходимо выполнить передачу сигнала START и первого байта адреса ведомого с битом RW равным 0 (запись в ведомое устройство), как указано выше. После завершения передачи необходимо записать второй байт адреса ведомого в регистр TXD, после чего установить бит WR в регистре CMD. Если требуется смена направления передачи на RW = 1 (чтение из ведомого устройства), то после завершения передачи второго байта адреса необходимо выполнить повторную передачу сигнала START и первого байта адреса ведомого с битом RW равным 1.

22.3.3 **Передача данных**

После успешного подтверждения приема адреса одним ведомым устройством может быть начата передача данных в направлении, задаваемым битом RW в посылке мастера. Каждый передаваемый бит подтверждается ACK на 9-й тактовый импульс SCL. Если ведомое устройство выдало NACK (нет подтверждения), то мастер может сгенерировать либо сигнал STOP для прекращения передачи, либо повторный сигнал START для начала нового цикла передачи.

Если мастер является принимающим устройством и выдает NACK, то ведомое устройство отпускает линию SDA и мастер может сгенерировать сигнал STOP или повторный сигнал START.

Для записи данных в ведомое устройство нужно записать данные в регистр TXD и установить бит WR в регистре CMD. Для чтения данных из устройства нужно установить бит RD в регистре CMD, при этом одновременно записав нужное значение в бит ACK. На время выполнения передачи контроллер интерфейса выставляет флаг TR_PROG в регистре STA. Когда передача завершена, этот флаг снимается и устанавливается флаг INT. Если при этом установлен бит разрешения EN_INT в регистре CTR, то генерируется прерывание контроллеру прерываний. Регистр RXD содержит корректные принятые данные после сброса флага TR_PROG или установки флага INT. Новый цикл чтения или записи можно начать только тогда, когда флаг TR_PROG сброшен.

22.3.4 **Сигнал STOP**

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

Сигнал STOP генерируется при записи единицы в бит STOP регистра CMD.

22.4 **Описание регистров контроллера I2C**

Таблица 230 – Описание регистров контроллера I2C

Базовый Адрес	Название	Описание
0x5003_0000	I2C	Контроллер I2C
Смещение		
0x00	PRL	Младшая часть предделителя частоты
0x04	PRH	Старшая часть предделителя частоты
0x08	CTR	Управление контроллером I2C
0x0C	RXD	Принятые данные по I2C
0x10	STA	Статус I2C
0x14	TXD	Передаваемые данные по I2C
0x18	CMD	Управление I2C

22.4.1 **PRL**

Таблица 231 – Регистр PRL

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

Таблица 232 – Описание бит регистра PRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…8	-	Зарезервировано
7…0	PR[7:0]	Младшая часть предделителя

22.4.2 **PRH**

Таблица 233 – Регистр PRH

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

Таблица 234 – Описание бит регистра PRH

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7…0	PR[15:8]	Старшая часть предделителя

22.4.3 **CTR**

Таблица 235 – Регистр CTR

Номер	31…8	7	6	5…0
Доступ	U	R/W	R/W	U
Сброс	0	0	0	0
	-	EN_I2C	EN_INT	-

Таблица 236 – Описание бит регистра CTR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7	EN_I2C	Разрешение работы контроллера I2C: 0 – выключен; 1 – включен

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
6	EN_INT	Разрешение прерывания от I2C: 0 – запрещено; 1 – разрешено
5…0	-	Зарезервировано

22.4.4 **RXD**

Таблица 237 – Регистр RXD

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

Таблица 238 – Описание бит регистра RXD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7…0	RXD[7:0]	Последний полученный по I2C байт

22.4.5 **STA**

Таблица 239 – Регистр STA

Номер	31…8	7	6	5	4…2	1	0
Доступ	U	R/O	R/O	R/O	U	R/O	R/O
Сброс	0	0	0	0	0	0	0
	-	RX_ACK	BUSY	LOST_ARB	-	TR_PROG	INT

Таблица 240 – Описание бит регистра STA

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…8	-	Зарезервировано
7	RX_ACK	Полученный от ведомого ACK/NACK: 0 – получен ACK; 1 – получен NACK. Бит устанавливается в 0 при потере арбитража
6	BUSY	Состояние шины I2C: 0 – после получения сигнала STOP; 1 – после получения состояния сигнала START
5	LOST_ARB	Потеря арбитража: 0 – нет потери арбитража; 1 – потерян арбитраж. Этот бит выставляется, если:

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		– получен сигнал STOP, но он не был инициирован этим контроллером; – контроллер пытается выставить SDA в высокий уровень, но SDA остается в низком уровне. Бит устанавливается в 0 после получения состояния сигнала START
4…2	-	Зарезервировано
1	TR_PROG	Процесс передачи: 0 – передача завершена; 1 – передаются данные. Бит является объединением битов RD и WR регистра CMD
0	INT	Флаг прерывания. 0 – нет прерывания; 1 – есть прерывание. Флаг выставляется если: – завершена передача байта; – был потерян арбитраж. Прерывание для процессора выставляется, если установлен бит EN_INT

22.4.6 **TXD**

Таблица 241 – Регистр TXD

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

Таблица 242 – Описание бит регистра TXD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7…0	TXD[7:0]	Байт для отправки по I2C. При передаче первого бита адреса нулевой бит определяет режим передачи: 0 – запись в ведомое устройство; 1 – чтение из ведомого устройства

22.4.7 **CMD**

Таблица 243 – Регистр CMD

Номер	31…8	7	6	5	4	3	2, 1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W	U	R/W
Сброс	0	0	0	0	0	0	0	0
	-	START	STOP	RD	WR	ACK	-	CLR_INT

Таблица 244 – Описание бит регистра CMD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Зарезервировано
7	START	Отправить сигнал START. 0 – нет действия; 1 – начать отправку сигнала START. Передача START выполняется только при установленных битах WR или RD. Сбрасывается автоматически при потере арбитража и после завершения передачи
6	STOP	Отправить сигнал STOP. 0 – нет действия; 1 – начать отправку сигнала STOP. Сбрасывается автоматически при потере арбитража и после завершения передачи
5	RD	Чтение из ведомого устройства: 0 – нет действия; 1 – начать чтение. Сбрасывается автоматически при потере арбитража и после завершения передачи
4	WR	Запись в ведомое устройство: 0 – нет действия; 1 – начать запись. Сбрасывается автоматически при потере арбитража и после завершения передачи
3	ACK	Отправить ACK: 0 – отправить ACK; 1 – отправить NACK. После завершения отправки NACK автоматически в 0 не сбрасывается, а очищается записью нуля.
2, 1	-	Зарезервировано
0	CLR_INT	Очистить прерывание INT. Запись 1 очищает прерывание

23 Контроллер SSP

Модуль порта синхронной последовательной связи (SSP – Synchronous Serial Port) выполняет функции интерфейса последовательной синхронной связи в режиме ведущего и ведомого устройства и обеспечивает обмен данными с подключенным ведомым или ведущим периферийным устройством в соответствии с одним из протоколов:

– интерфейс SPI фирмы Motorola;

– интерфейс SSI фирмы Texas Instruments;

– интерфейс Microwire фирмы National Semiconductor.

Как в ведущем, так и в ведомом режиме работы модуль SSP обеспечивает:

– преобразование данных, размещенных во внутреннем буфере FIFO передатчика (восемь 32-разрядных ячеек данных), из параллельного в последовательный формат;

– преобразование данных из последовательного в параллельный формат и их запись в аналогичный буфер FIFO приемника (восемь 32-разрядный ячеек данных).

Модуль формирует сигналы прерываний по следующим событиям:

– необходимость обслуживания буферов FIFO приемника или передатчика;

– переполнение буфера FIFO приемника;

– наличие данных в буфере FIFO приемника по истечении времени таймаута;

– наличие данных в FIFO приемника;

– отсутствие данных в FIFO передатчика;

– отсутствие данных в сдвиговом регистре передатчика.

Основные сведения о модуле представлены в следующих подразделах:

– Характеристики интерфейса SPI;

– Характеристики интерфейса Microwire;

– Характеристики интерфейса SSI.

23.1 Основные характеристики модуля SSP

– Функционирует как в ведущем, так и в ведомом режиме;

– Программное управление скоростью обмена;

– Состоит из независимых буферов приема и передачи (восемь ячеек по 32 бита) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);

– Программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;

– Программируемая длительность информационного кадра от 4 до 32 бит;

– Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по переполнению буфера приемника, по наличию данных в FIFO приемника, по отсутствию данных в FIFO передатчика, а также по отсутствию данных в сдвиговом регистре передатчика;

– Доступна возможность тестирования по шлейфу, соединяющему вход с выходом;

– Поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена на рисунке 63.

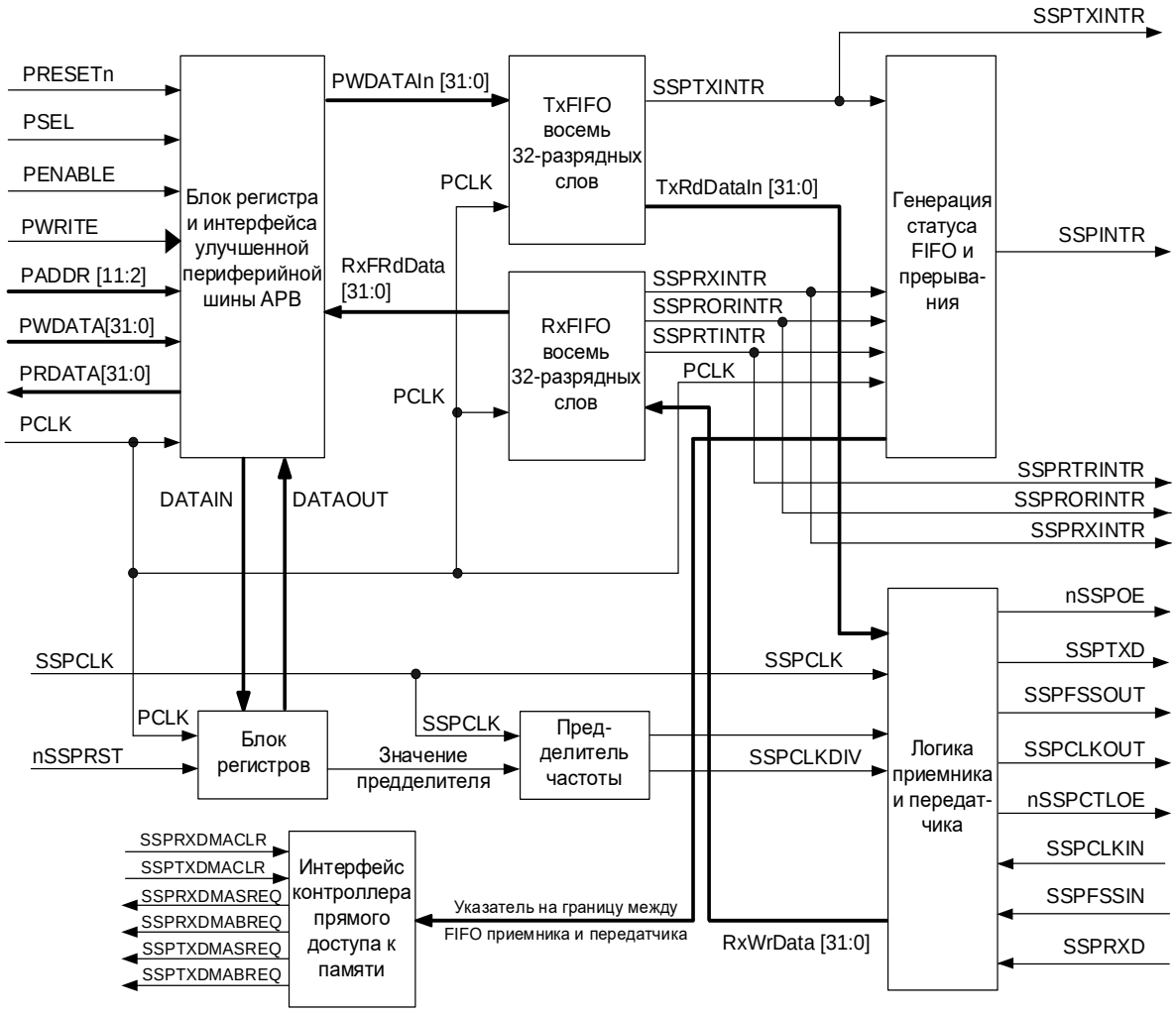


Рисунок 63 – Структурная схема модуля SSP

23.1.1 **Программируемые параметры**

Следующие ключевые параметры могут быть заданы программно:

– режим функционирования периферийного устройства – ведущее или ведомое;
– разрешение или запрещение функционирования;
– формат информационного кадра;
– скорость передачи данных;
– фаза и полярность тактового сигнала;
– размер блока данных – от 4 до 32 бит;
– сброс FIFO передатчика;
– маскирование прерываний.

23.1.2 **Характеристики интерфейса SPI**

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:

– полнодуплексный обмен данными по четырехпроводной линии;
– программное задание фазы и полярности тактового сигнала.

23.1.3 Характеристики интерфейса Microwire

Интерфейс Microwire фирмы National Semiconductor обеспечивает:

– полудуплексный обмен данными с использованием 8-битных управляющих последовательностей.

23.1.4 Характеристики интерфейса SSI

Интерфейс SSI фирмы Texas Instruments обеспечивает:

– полнодуплексный обмен данными по четырехпроводной линии;

– возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

23.2 Общий обзор модуля SSP

Модуль SSP представляет собой интерфейс синхронного последовательного обмена данными способный функционировать в качестве ведущего или ведомого устройства, поддерживающий протоколы передачи данных SPI фирмы Motorola, Microwire фирмы National Semiconductor, а также SSI фирмы Texas Instruments.

Модуль выполняет следующие функции:

– преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;

– преобразование данных, передаваемых на периферийное устройство, из параллельной в последовательную форму;

– центральный процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии;

– прием и передача данных буферизуются с помощью буферов FIFO и обеспечивают хранение до восьми слов данных шириной до 32 бит независимо для режимов приема и передачи.

Последовательные данные передаются по линии SSPTXD и принимаются с линии SSPRXD.

Модуль SSP содержит программируемые делители частоты, формирующие тактовый сигнал обмена данными SSPCLKOUT (сигнал, поступающий на линию SSPSCKOUT и предназначенный для синхронизации приема и передачи данных) из сигнала, поступающего на линию SSPCLK (сигнал, который подается на модуль SSP с блока формирования тактовых частот). Скорость передачи данных может быть более 2 МГц, в зависимости от частоты SSPCLK и характеристик подключенного периферийного устройства.

Режим обмена данными, формат информационного кадра и количество бит данных задаются программно с помощью регистров управления CR0 и CR1.

Модуль формирует следующие независимо маскируемые прерывания:

– SSPTXINTR – запрос на обслуживание буфера передатчика;

– SSPRXINTR – запрос на обслуживание буфера приемника;

– SSPRORINTR – переполнение приемного буфера FIFO;

– SSPRTINTR – таймаут ожидания чтения данных из приемного FIFO;
– SSPRNEINTR – наличие данных в буфере FIFO приемника;
– SSPTFEINTR – отсутствие данных в буфере FIFO передатчика;
– SSPTNBSYINTR – отсутствие данных в сдвиговом регистре передатчика.

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

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

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

23.2.1 **Блок формирования тактового сигнала**

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSPCLK с помощью внутреннего делителя частоты, состоящего из двух последовательно соединенных счетчиков без цепи сброса.

Коэффициент предварительного деления частоты в диапазоне от 2 до 254 с шагом 2 можно задать путем записи значения в регистр CPSR. Так как младший значащий разряд коэффициента деления не используется, то исключается возможность деления частоты на нечетный коэффициент деления. Это, в свою очередь, гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровней).

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

Коэффициент деления второго делителя задается программно в диапазоне от 1 до 256 путем записи соответствующего значения в регистр управления CR0.

23.2.2 **Буфер FIFO передатчика**

Буфер передатчика имеет ширину 32 бита, глубину 8 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Данные от центрального процессора сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных.

В случае работы контроллера SPI в режиме ведомого устройства, чтение пустого FIFO приводит к выдаче некорректных данных.

23.2.3 **Буфер FIFO приемника**

Буфер приемника имеет ширину 32 бита, глубину 8 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Принятые от периферийного

устройства данные сохраняются в этом буфере блоком приема данных до тех пор, пока не будут считаны центральным процессором.

23.2.4 Блок приема и передачи данных

23.2.4.1 Режим ведущего устройства

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

Блок передатчика последовательно считывает данные из буфера FIFO передатчика и производит их преобразование из параллельной формы в последовательную. Далее поток последовательных данных и элементов кадровой синхронизации, тактированный сигналом SSPCLKOUT, передаётся по линии SSPTXD к подключенным ведомым устройствам.

Блок приемника выполняет преобразование данных, поступающих синхронно с линии SSPRXD, из последовательной в параллельную форму. После этого загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

23.2.4.2 Режим ведомого устройства

В режиме ведомого устройства тактовый сигнал обмена данными формируется одним из подключенных к модулю периферийных устройств и поступает по линии SSPCLKIN.

При этом блок передатчика, тактируемый этим внешним сигналом, считывает данные из буфера FIFO, преобразует их из параллельной формы в последовательную. После этого выдает поток последовательных данных и элементов кадровой синхронизации в линию SSPTXD.

Аналогично, блок приемника выполняет преобразование данных, поступающих с линии SSPRXD синхронно с сигналом SSPCLKIN, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны процессором.

23.2.5 Блок формирования прерываний

Модуль SSP генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения указанных независимых прерываний по схеме ИЛИ.

Комбинированный сигнал прерывания подаётся на контроллер прерываний CLIC, при этом появляется дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

23.3 Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

23.4 Конфигурирование приемопередатчика

После сброса работа блоков приемопередатчика запрещается до выполнения процедуры задания конфигурации.

Для этого необходимо выбрать ведущий или ведомый режим работы устройства, а также используемый протокол передачи данных (SPI фирмы Motorola, SSI фирмы Texas Instruments, либо Microwave фирмы National Semiconductor), после чего записать необходимую информацию в регистры управления CR0 и CR1.

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

23.5 Разрешение работы приемопередатчика

Разрешение осуществляется путем установки бита SSE регистра управления CR1. Буфер FIFO передатчика может быть либо проинициализирован путем записи в него до восьми 32-разрядных слов заблаговременно перед установкой этого бита, либо может заполняться передаваемыми данными в процедуре обслуживания прерывания.

После разрешения работы модуля приемопередатчик начинает обмен данными по линиям SSPTXD и SSPRXD.

23.6 Соотношения между тактовыми сигналами

В модуле имеется ограничение на соотношение между частотами тактовых сигналов CPU_CLK и SSPCLK. Частота SSPCLK должна быть меньше или равна частоте CPU_CLK

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

Выполнение этого требования гарантирует синхронизацию сигналов управления, передаваемых из зоны действия тактового сигнала SSPCLK в зону действия сигнала CPU_CLK в течение времени, меньшего продолжительности передачи одного информационного кадра.

В режиме ведомого устройства сигнал SSPCLKIN от ведущего внешнего устройства поступает на схемы синхронизации, задержки и обнаружения фронта. Для того чтобы обнаружить фронт сигнала SSPCLKIN, необходимо три такта сигнала SSPCLK. Сигнал SSPTXD имеет меньшее время установки по отношению к заднему фронту SSPCLKIN, по которому и происходит считывание данных из линии. Время установки и удержания сигнала SSPRXD по отношению к сигналу SSPCLKIN должно выбираться с запасом, гарантирующим правильное считывание данных. Для обеспечения корректной работы устройства необходимо, чтобы частота SSPCLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSPCLKIN.

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

Так в режиме ведущего устройства для обеспечения максимальной скорости обмена 1,8432 Мбит/с частота сигнала SSPCLK должна составлять не менее 3,6864 МГц. В этом случае в регистр CPSR должно быть записано значение 2, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

В режиме ведомого устройства для обеспечения той же информационной скорости необходимо использовать тактовый сигнал SSPCLK с частотой не менее 22,12 МГц. При этом в регистр CPSR должно быть записано значение 12, а поле SCR[7:0] регистра CR0 должно быть установлено в 0.

Соотношение между максимальной частотой сигнала SSPCLK и минимальной частотой SSPCLKOUT составляет 254 × 256.

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

$$\begin{cases} F_{SSPCLK}\ (min) \geq 2 \times F_{SSPCLKOUT}\ (\text{max})\ [\text{режим ведущего}]\ , \\ F_{SSPCLK}\ (min) \geq 12x \times F_{SSPCLKIN}\ (\text{max})\ [\text{режим ведомого}]\ . \end{cases}$$

При активации сигнала SSPFRX в регистре управления CR0 в режиме SPI это соотношение приобретает вид

$$\begin{cases} F_{SSPCLK}\ (min) \geq 2 \times F_{SSPCLKOUT}\ (\text{max})\ [\text{режим ведущего}]\ , \\ F_{SSPCLK}\ (min) \geq 4 \times F_{SSPCLKIN}\ (\text{max})\ [\text{режим ведомого}]\ . \end{cases}$$

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

$$\begin{cases} F_{SSPCLK}\ (max) \leq 254 \times 256 \times F_{SSPCLKOUT}\ (\text{min})\ [\text{режим ведущего}]\ , \\ F_{SSPCLK}\ (max) \leq 254 \times 256 \times F_{SSPCLKIN}\ (\text{min})\ [\text{режим ведомого}]\ . \end{cases}$$

23.7 Программирование регистра управления CR0

Регистр CR0 предназначен для:

- установки скорости информационного обмена;
- выбора одного из трех протоколов обмена данными;
- выбора размера слова данных;
- активации быстрого режима работы ведомого устройства SPI.

Скорость информационного обмена зависит от частоты внешнего тактового сигнала SSPCLK и коэффициента деления блока формирования тактового сигнала. Последний задается совместно значением поля SCR (Serial Clock Rate – скорость информационного обмена) регистра CR0 и значением поля CPSDVSR (clock prescale divisor value – коэффициент деления тактового сигнала) регистра CPSR.

Формат информационного кадра задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS регистра CR0.

Сигнал SSPFRX используется для активации у ведомого модуля SPI быстрого режима работы и включения синхронизации сигнала SSPRXD.

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

23.8 Программирование регистра управления CR1

Регистр CR1 предназначен для:

– выбора ведущего или ведомого режима функционирования приемопередатчика;

– включения режима проверки канала по шлейфу;

– разрешения или запрещения работы модуля;

– программного сброса FIFO передатчика.

Выбор ведущего режима осуществляется путем записи 0 в поле MS регистра CR1 (это значение устанавливается после сброса автоматически).

Запись 1 в поле MS переводит приемопередатчик в режим ведомого устройства. В этом режиме разрешение или запрещение формирования сигнала передатчика SSPTXD осуществляется путем установки бита SOD (slave mode SSPTXD output disable – запрет линии SSPTXD для ведомого режима) регистра CR1. Указанная функция полезна при подключении к одной линии нескольких подчиненных устройств.

Для того чтобы разрешить функционирование приемопередатчика, необходимо установить в 1 бит SSE (Synchronous Serial Port Enable – разрешение последовательного синхронного порта).

Для программного сброса FIFO передатчика необходимо установить в 1 бит RESTxFIFO. После сброса сигнал RESTxFIFO автоматически сбросится в 0.

23.9 Формирование тактового сигнала обмена данными

Тактовый сигнал обмена данными формируется путем деления частоты тактового сигнала SSPCLK. На первом этапе формирования частота этого сигнала делится на четный коэффициент CPSDVSR, лежащий в диапазоне от 2 до 254, доступный для программирования через регистр CPSR. Сформированный сигнал далее поступает на делитель частоты с коэффициентом (1 + SCR) от 1 до 256, где значение SCR доступно для программирования через CR0.

Частота выходного тактового сигнала обмена данными SSPCLK определяется следующим соотношением

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \cdot (1 + SCR)} \quad (13)$$

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

23.10 Формат информационного кадра

Каждый информационный кадр содержит в зависимости от запрограммированного значения от 4 до 32 бит данных. Передача данных начинается со старшего значащего разряда. Есть возможность выбрать три базовых структуры построения кадра:

– SSI фирмы Texas Instruments;

– SPI фирмы Motorola;
– Microwire фирмы National Semiconductor.

Во всех режимах построения кадра тактовый сигнал SSPCLK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SSPCLK в неактивное состояние используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

В режимах SPI и Microwire выходной сигнал кадровой синхронизации передатчика SSPFSS имеет активный низкий уровень и поддерживается в низком уровне в течение всего периода передачи информационного кадра.

В режиме построения кадра SSI фирмы Texas Instruments перед началом каждого информационного кадра на выходе SSPFSS формируется импульс с длительностью, равной одному тактовому интервалу обмена данными. В этом режиме приемопередатчик SSP, равно как и ведомые периферийные устройства, передаёт данные в линию по переднему фронту сигнала SSPCLK, а считывает данные из линии по заднему фронту этого сигнала.

В отличие от полнодуплексных режимов передачи данных SSI и SPI, режим Microwire фирмы National Semiconductor использует специальный способ обмена данными между ведущим и ведомым устройством, функционирующий в режиме полудуплекса. В указанном режиме на внешнее ведомое устройство перед началом передачи информационного кадра посылается специальная 8-битная управляющая последовательность. В течение всего времени передачи этой последовательности приемник не обрабатывает каких-либо входных данных. После того как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 32 бит, таким образом общая длительность информационного кадра составляет от 13 до 41 бит.

23.10.1 **Формат синхронного обмена SSI фирмы Texas Instruments**

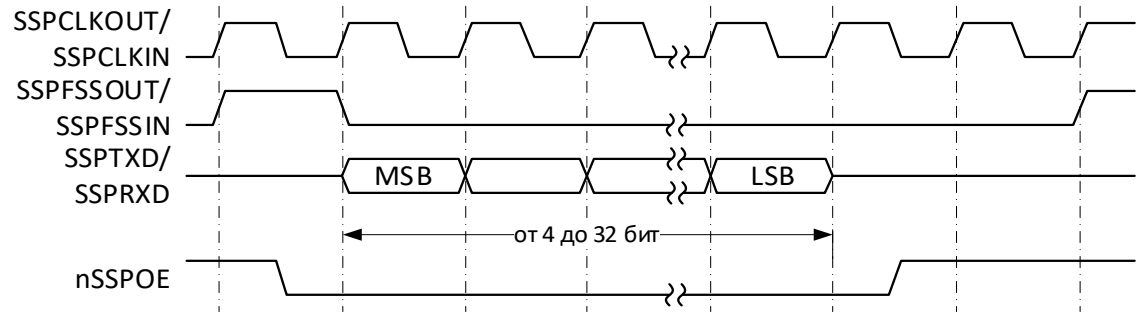


Рисунок 64 – Формат синхронного обмена протокола SSI (единичный обмен)

В данном режиме при неактивном приемопередатчике SSP сигналы SSPCLK и SSPFSS переводятся в низкий логический уровень, а линия передачи данных SSPTXD поддерживается в третьем состоянии.

После появления хотя бы одного элемента в буфере FIFO передатчика сигнал SSPFSSOUT переводится в высокий логический уровень на время, соответствующее одному периоду сигнала SSPCLK. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSPCLK старший значащий разряд информационного кадра (2 – 32 бит данных) выдается на выход линии SSPTXD и т.д.

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

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments привдены на рисунке 64 (передача единичного информационно кадра) и рисунке 65 (передача последовательности кадров).

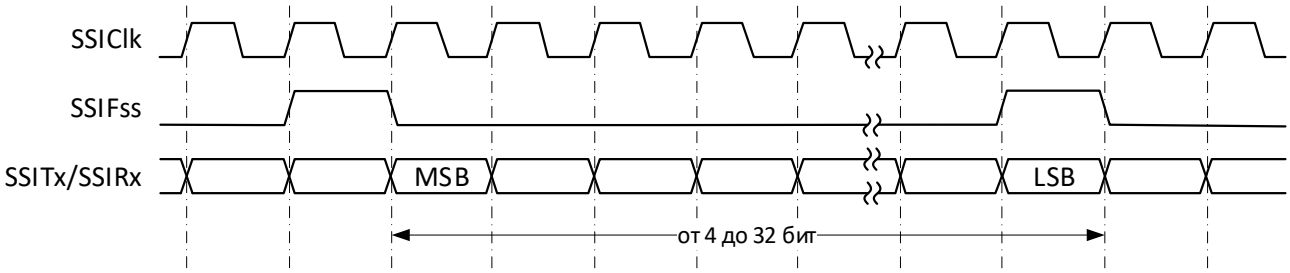


Рисунок 65 – Формат синхронного обмена протокола SSI (непрерывный обмен)

23.10.2 **Формат синхронного обмена SPI фирмы Motorola**

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSPFSS выполняет функцию выбора ведомого устройства. Главной особенностью протокола SPI является возможность выбора состояния и фазы сигнала SSPCLK в режиме ожидания (неактивном приемопередатчике) путем задания значений бит SPO и SPH регистра управления CR0.

23.10.2.1 **Выбор полярности тактового сигнала – бит SPO**

Если бит SPO равен 0, то в режиме ожидания линия SSPCLK переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SSPCLK переводится в высокий логический уровень.

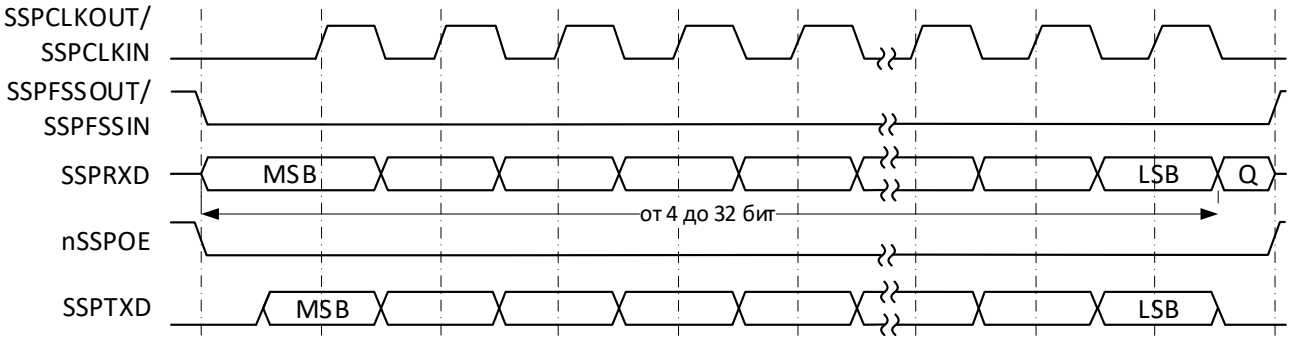
23.10.2.2 **Выбор фазы тактового сигнала – бит SPH**

Значение бита SPH определяет фронт тактового сигнала, по которому осуществляется выборка данных и изменение состояния на выходе линии.

В случае если бит SPH установлен в 0, регистрация данных приемником осуществляется по нарастающему фронта тактового сигнала, в противном случае по спадающему фронту.

23.10.3 **Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0**

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 0, SPH = 0 приведены на рисунке 66 (одиночный обмен) и рисунке 67 (непрерывный обмен).



Q – сигнал с неопределенным уровнем.

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

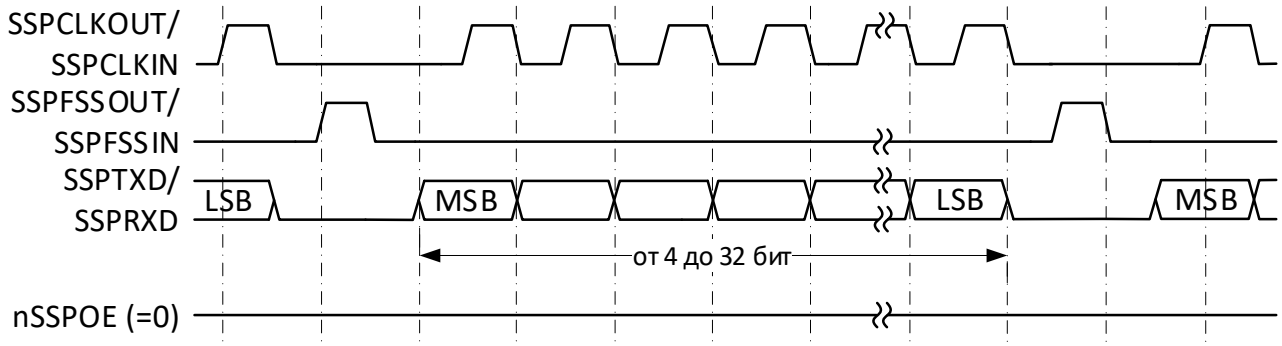


Рисунок 67 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

– сигнал SSPCLK имеет низкий логический уровень;

– сигнал SSPFSS имеет высокий логический уровень;

– сигнал SSPTXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. При этом линия передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

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

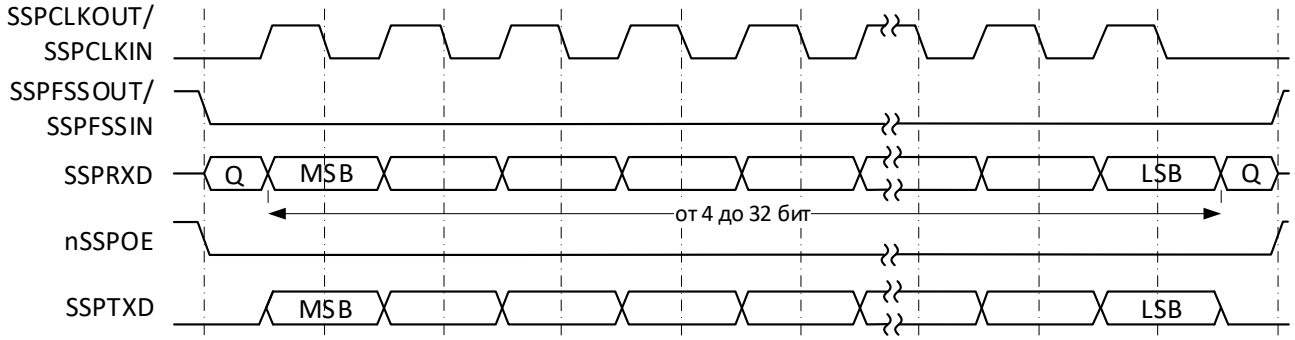
Далее данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSPCLK.

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

В режиме непрерывной передачи данных на линии SSPFSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSPFSS в высокий уровень по окончании передачи каждого кадра, разрешая запись новых данных. По окончании приема последнего бита блока данных линия SSPFSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLK.

23.10.4 **Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1**

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=1 приведены на рисунке 68 (одиночный и непрерывный обмен).



Q – сигнал с неопределенным уровнем.

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

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLK имеет низкий логический уровень;
- сигнал SSPFSS имеет высокий логический уровень;
- сигнал SSPTXD переводится в высокоимпедансное состояние.

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

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

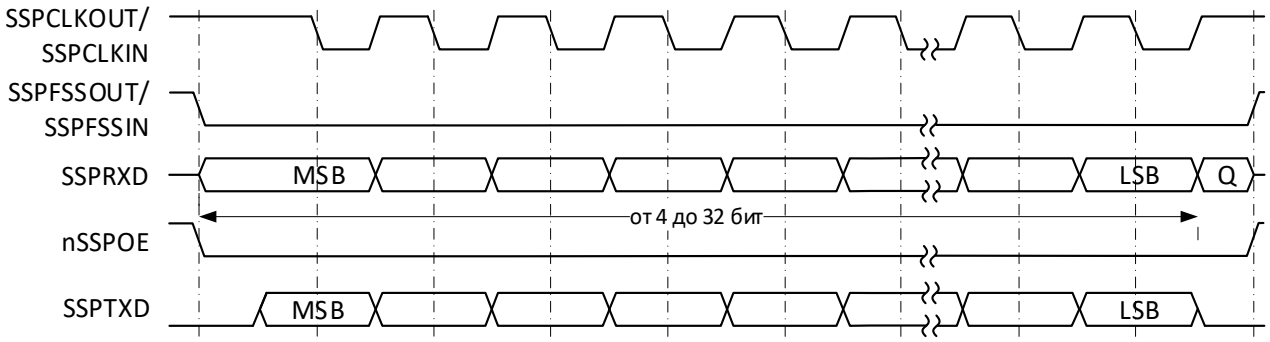
Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSPCLK.

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

В режиме непрерывной передачи данных линия SSPFSS постоянно находится в низком логическом уровне, и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

23.10.5 **Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0**

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 1, SPH = 0 приведены на рисунке 69 (одиночный обмен) и рисунке 70 (непрерывный обмен).



Q – сигнал с неопределенным уровнем.

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

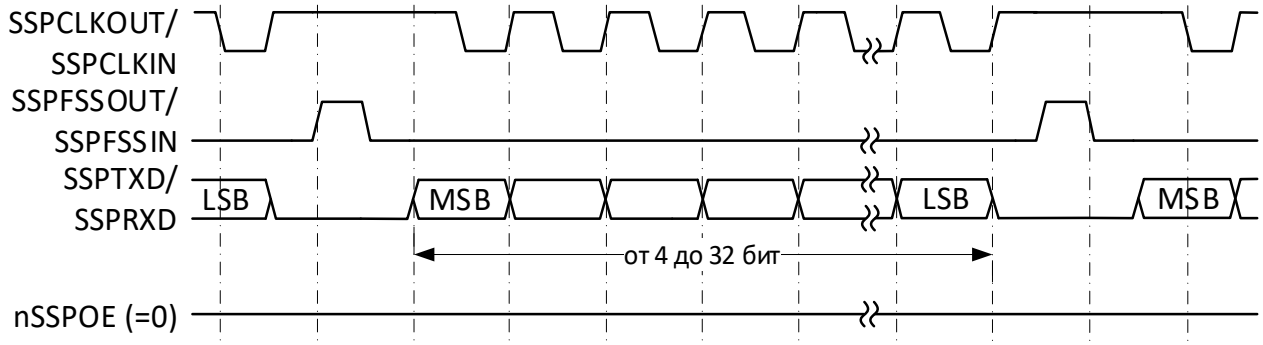


Рисунок 70 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

– сигнал SSPCLK имеет высокий логический уровень;

– сигнал SSPFSS имеет высокий логический уровень;

– сигнал SSPTXD переводится в высокоимпедансное состояние.

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

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

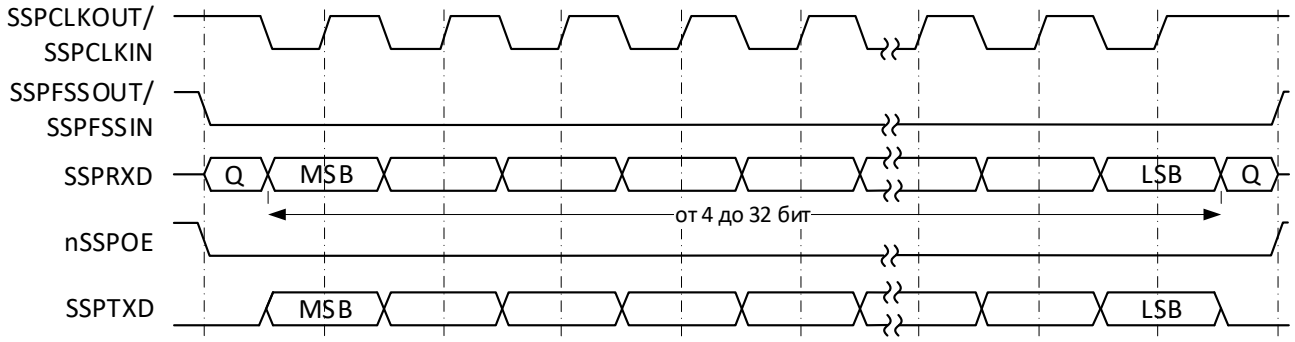
Далее данные регистрируются по заднему фронту и выдаются в линию по переднему фронту сигнала SSPCLK.

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

В режиме непрерывной передачи данных на линии SSPFSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSPFSS в высокий уровень по окончании передачи каждого кадра, разрешая запись новых данных. По окончании приема последнего бита блока данных линия SSPFSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLK.

23.10.6 **Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1**

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 1, SPH = 1 приведены на рисунке 71 (одиночный и непрерывный обмен).



Q – сигнал с неопределенным уровнем.

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

В данном режиме во время ожидания приемопередатчика:

– сигнал SSPCLK имеет высокий логический уровень;

– сигнал SSPFSS имеет высокий логический уровень;

– сигнал SSPTXD переводится в высокоимпедансное состояние.

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

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

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

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

В режиме непрерывной передачи данных линия SSPFSS постоянно находится в низком логическом уровне и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

23.10.7 Формат синхронного обмена Microwire фирмы National Semiconductor

Временные диаграммы последовательного синхронного обмена в режиме Microwire приведены на рисунке 72 (одиночный обмен) и рисунке 73 (непрерывный обмен).

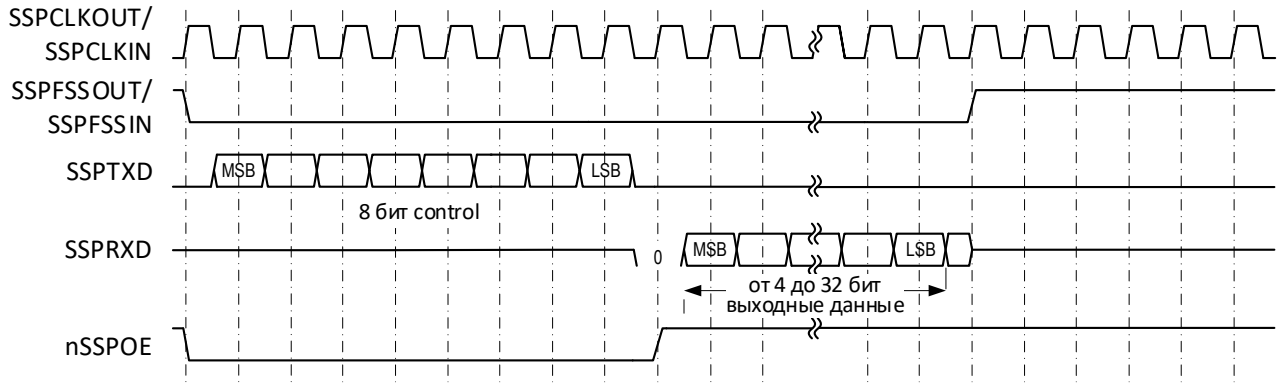


Рисунок 72 – Формат синхронного обмена протокола Microwire (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен в нем осуществляется в полудуплексном режиме, с использованием служебных последовательностей. Каждая информационный обмен начинается с передачи ведущим устройством специальной 8-битной управляющей последовательности. В течение всего времени ее передачи приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 32 бит, таким образом, общая длительность информационного кадра составляет от 13 до 41 бит.

В данном режиме во время ожидания приемопередатчика:

– сигнал SSPCLK имеет низкий логический уровень;
– сигнал SSPFSS имеет высокий логический уровень;
– сигнал SSPTXD переводится в высокоимпедансное состояние.

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

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSPCLK. По окончании приема последнего бита управляющей последовательности она декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля SSP. Биты данных выдаются в линию SSPRXD по заднему фронту сигнала SSPCLK. Ведущее устройство в свою очередь регистрирует их по переднему фронту этого тактового сигнала. В случае одиночного информационного обмена по окончании приема последнего бита слова данных сигнал SSPFSS переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных из регистра сдвига в буфер FIFO приемника.

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

Непрерывный обмен данными начинается и заканчивается так же, как и одиночный обмен. Однако линия SSPFSS удерживается в низком логическом уровне в течение всего сеанса передачи данных. Управляющий байт следующего информационного кадра передается сразу же после приема младшего значащего разряда текущего кадра. Данные из сдвигового регистра передаются в буфер приемника после регистрации младшего разряда очередного слова по заднему фронту сигнала SSPCLK.

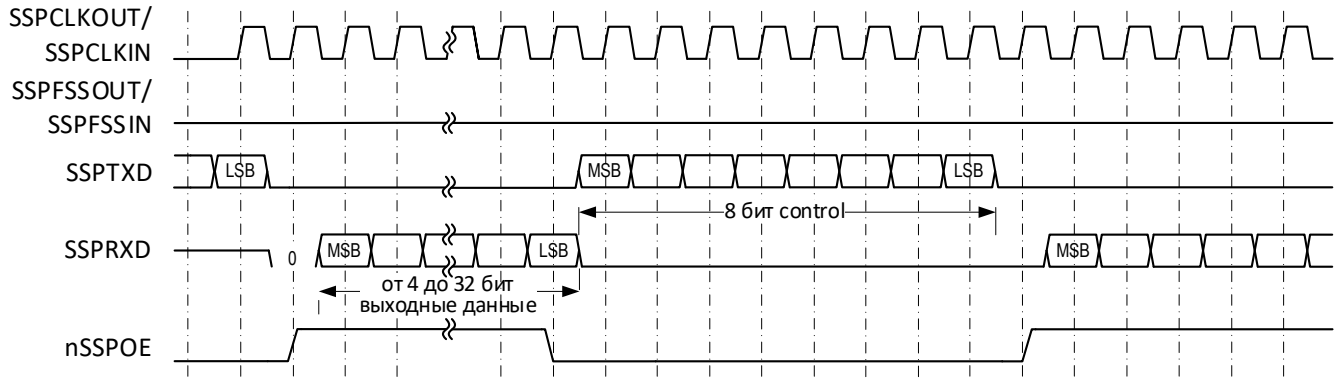


Рисунок 73 – Формат синхронного обмена протокола Microwire (непрерывный обмен)

23.10.7.1 Требования к временным параметрам сигнала SSPFSS относительно тактового сигнала SSPCLK в режиме Microwire

Модуль SSP, работающий в режиме Microwire как ведомое устройство, регистрирует данные по переднему фронту сигнала SSPCLK после установки сигнала SSPFSS в низкий логический уровень. Ведущие устройства, формирующие

сигнал SSPCKL, должны гарантировать достаточное время установки и удержания сигнала SSPFSS по отношению к переднему фронту сигнала SSPCLK.

Данные требования приведены на рисунке 74. По отношению к переднему фронту сигнала SSPCLK, по которому осуществляется регистрация данных в приемнике ведомого модуля SSP, время установки сигнала SSPFSS должно быть, как минимум в два раза больше периода SSPCLK, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSPCLK должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.

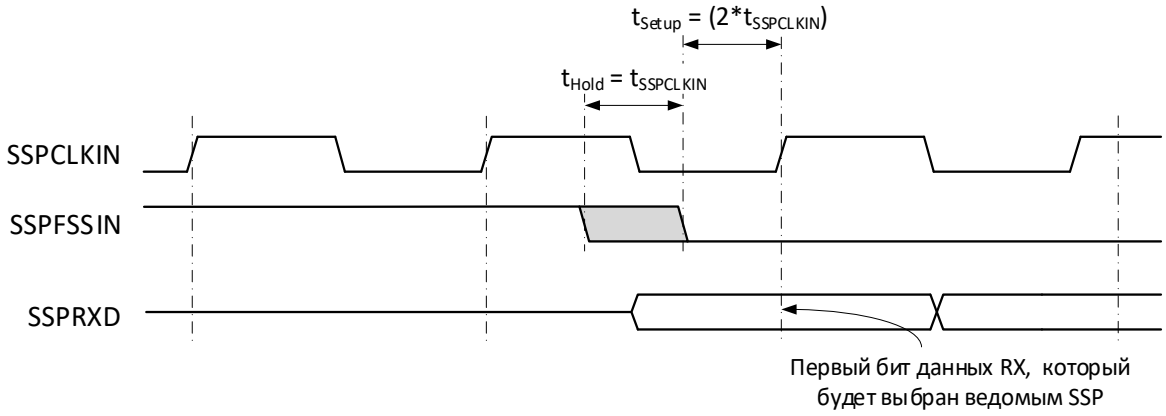


Рисунок 74 – Формат Microwire, требования к времени установки и удержания сигнала SSPFSSIN

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

На рисунках 75 – 77 приведены 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.

На рисунке 75 приведена 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 разрешает прохождение сигнала от своей линии SSPTXD на вход SSPRXD ведущего.

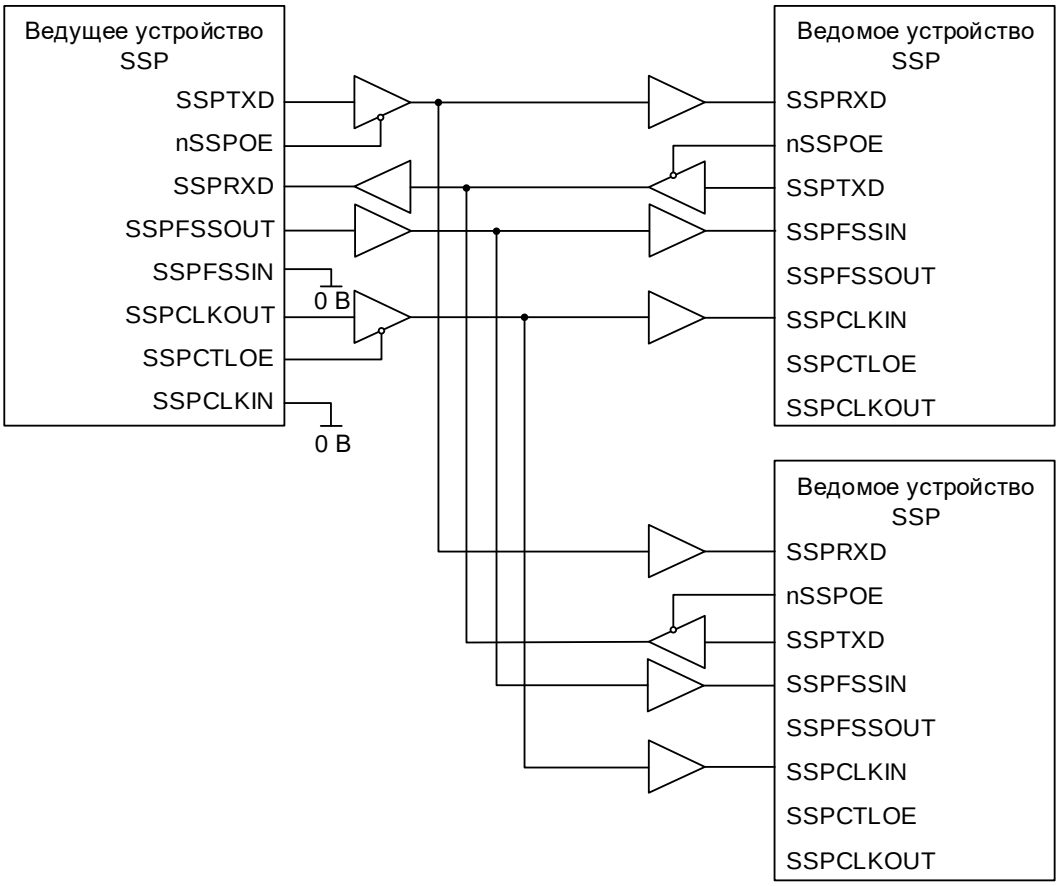


Рисунок 75 – Ведущее устройство SSP подключено к двум ведомым

На рисунке 76 приведено подключение модуля SSP, сконфигурированного как ведущее устройство, к двум ведомым устройствам, поддерживающим протокол SPI фирмы Motorola. Внешние устройства сконфигурированы как ведомые путем установки в низкий логический уровень сигнала выбора ведомого устройства Slave Select (SS). Как и в предыдущем примере, ведущее устройство способно передавать данные в адрес ведомых циркулярно по линии SSPTXD. Ответная передача данных на входную линию SSPRXD ведущего устройства одновременно осуществляется только одним из ведомых по соответствующей линии MISO.

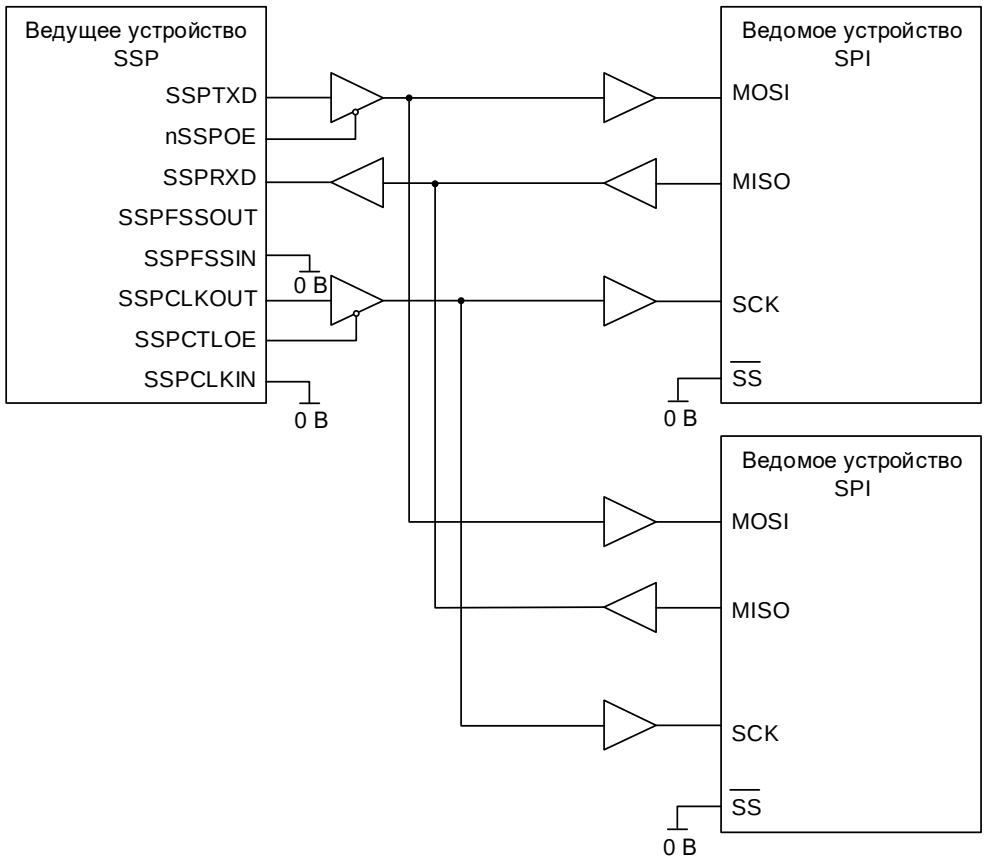


Рисунок 76 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим SPI

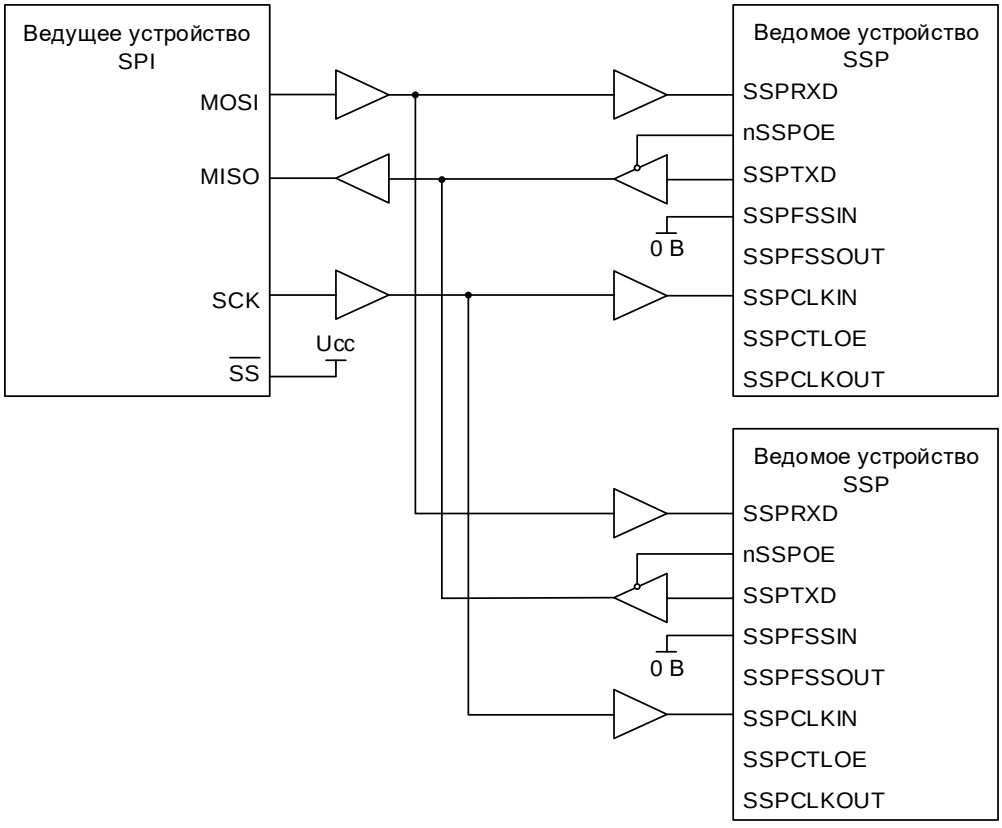


Рисунок 77 – Ведущее устройство, протокол SPI, подключено к двум ведомым модулям SSP

На рисунке 77 приведено ведущее устройство, поддерживающее протокол SPI фирмы Motorola, соединенное с двумя модулями SSP, сконфигурированными для работы в ведомом режиме. Линия Slave Select (SS) ведущего устройства в этом случае установлена в высокий логический уровень. Ведущее устройство осуществляет передачу данных по линии MOSI циркулярно в адрес двух ведомых модулей.

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

23.12 **Интерфейс прямого доступа к памяти**

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

Интерфейс DMA включает в себя следующие сигналы:

- для приема:
  - SSPRXDMASREQ – запрос передачи отдельного слова, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит, по меньшей мере, одно слово;
  - SSPRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если буфер FIFO приемника содержит четыре или более слов;
  - SSPRXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего слова данных в блоке.
- для передачи:
  - SSPTXDMASREQ – запрос передачи отдельного слова, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку;
  - SSPTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит четыре или менее слов;
  - SSPTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего слова данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимоисключающими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение четыре, формируются как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае если количество данных в буфере приема меньше

порогового значения, формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно принять 19 символов. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

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

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

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

В таблице 245 приведены значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 245 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
½	4	4

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

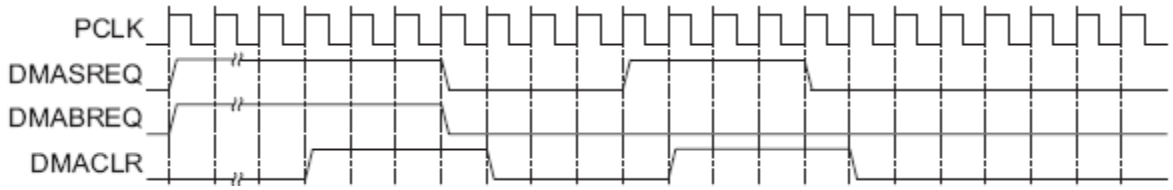


Рисунок 78 – Временные диаграммы обмена в режиме DMA

23.13 **Прерывания**

В модуле предусмотрено семь маскируемых линий запроса на прерывание с выводом на один общий сигнал, представляющий собой комбинацию независимых по схеме ИЛИ.

Сигналы запроса на прерывание:

- SSPRXINTR – запрос на обслуживание буфера FIFO приемника;
- SSPTXINTR – запрос на обслуживание буфера FIFO передатчика;
- SSPRORINTR – переполнение буфера FIFO приемника;
- SSPRTINTR – таймаут приемника;

– SSPRNEINTR – наличие данных в FIFO приемника;

– SSPTFEINTR – отсутствие данных в FIFO передатчика;

– SSPTNBSYINTR – отсутствие данных в сдвиговом регистре передатчика;

– SSPINTR – логическое ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR, SSPRORINTR, SSPRNEINTR, SSPTFEINTR и SSPTNBSYINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSPIMSC. Установка бита в 1 разрешает соответствующее прерывание, а в 0 – запрещает.

Доступность индивидуальных линий, и общей линии запроса, позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика SSPRXINTR и SSPTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать данные сигналы запроса для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний SSPRIS, либо из маскированного регистра прерываний SSPMIS.

23.13.1 **SSPRXINTR**

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

23.13.2 **SSPTXINTR**

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

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

23.13.3 **SSPRORINTR**

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

23.13.4 **SSPRTINTR**

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

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

23.13.5 **SSPRNEINTR**

Прерывание по наличию данных в FIFO приемника. Сигнал формируется при записи хотя бы одного слова в FIFO приемника и сохраняется до тех пор, пока все данные не будут считаны.

23.13.6 **SSPTFEINTR**

Прерывание по отсутствию данных в FIFO передатчика. Находится в активном уровне до тех пор, пока в FIFO не будет записано хотя бы одно слово. После передачи всех данных возвращается в активный уровень.

23.13.7 **SSPTNBSYINTR**

Прерывание по отсутствию данных в сдвиговом регистре передатчика.

23.13.8 **SSPINTR**

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRNEINTR, SSPTNBSYINTR, SSPTBSYINTR, SSPRTINTR и SSPRORINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерываний, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

23.14 **Программное управление модулем**

23.14.1 **Общая информация**

Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:

– адреса со смещениями в диапазоне +0x028 … +0x07C и +0xFD0 … +0xFDC зарезервированы для перспективных расширений возможностей модуля;

– адреса со смещениями в диапазоне +0x080 … +0x088 зарезервированы для тестирования.

23.14.2 **Описание регистров контроллера SSP**

Данные о регистрах контроллера SSP приведены в таблице 246.

Таблица 246 – Обобщенные данные о регистрах модуля SSP

Базовый адрес	Наименование				Описание
0x5000_0000	SSP1				Регистры контроллера интерфейса SSP1
0x500C_8000	SSP2				Регистры контроллера интерфейса SSP2
0x500D_0000	SSP3				Регистры контроллера интерфейса SSP3
Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x000	CR0	RW	0x0000	18	Регистр управления 0
0x004	CR1	RW	0x0	5	Регистр управления 1
0x008	DR	RW	0x----	32	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись)
0x00C	SR	RO	0x03	5	Регистр состояния
0x010	CPSR	RW	0x00	8	Регистр делителя тактовой частоты
0x014	IMSC	RW	0x0	7	Регистр маски прерывания
0x018	RIS	RO	0x8	7	Регистр состояния прерываний без учета маскирования
0x01C	MIS	RO	0x0	7	Регистр состояния прерываний с учетом маскирования
0x020	ICR	WO	0x0	2	Регистр сброса прерывания
0x024	DMACR	RW	0x0	2	Регистр управления прямым доступом к памяти

Примечание – В графе Тип указан вид доступа к регистру:
RW – чтение и запись;
RO – только чтение;
WO – только запись

23.14.2.1 **Регистр CR0**

Регистр управления 0

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

Назначение бит регистра CR0 представлено в таблице 247.

Таблица 247 – Формат регистра CR0

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…18	-	Зарезервировано
17	SSPFRX	Для ведомого модуля включает быстрый режим протокола SPI и активирует синхронизацию сигнала RXD: 1 – активация быстрого режима работы ведомого; 0 – нормальный режим работы

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
16	DSS[4]	Старший разряд размера слова данных: 0 – DSS[3:0]; 1 – DSS[3:0] + 16 бит
15…8	SCR	Скорость последовательного обмена. Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению $\frac{F_{SSPCLK}}{\left(CPSDVR * (1 + SCR)\right)}$, где CPSDVR – четное число в диапазоне от 2 до 254 (см. Регистр CPSR), SCR – число от 0 до 255
7	SPH	Фаза сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. подраздел «Формат синхронного обмена SPI фирмы Motorola»
6	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. подраздел «Формат синхронного обмена SPI фирмы Motorola»
5…4	FRF	Формат информационного кадра: 00 – протокол SPI фирмы Motorola; 01 – протокол SSI фирмы Texas Instruments; 10 – протокол Microwire фирмы National Semiconductor; 11 – резерв
3…0	DSS[3:0]	Младшие 4 разряда размера слова данных. При DSS[4] = 0: 0000 – резерв 0001 – резерв 0010 – резерв 0011 – 4 бита 0100 – 5 бит 0101 – 6 бит … 1110 – 15 бит 1111 – 16 бит При DSS[4] = 1: 0000 – 17 бит 0001 – 18 бит 0010 – 19 бит 0011 – 20 бит 0100 – 21 бит 0101 – 22 бита … 1110 – 31 бит 1111 – 32 бита

23.14.2.2 **Регистр CR1**

Регистр управления 1

Регистр CR1 содержит пять битовых полей, предназначенных для управления блоками модуля SSP.

Назначение бит регистра CR1 представлено в таблице 248.

Таблица 248 – Регистр CR1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…5		Резерв, при чтении результат не определен. При записи следует устанавливать в 0
4	RESTxFIFO	Программный сброс FIFO передатчика: 0 – нормальный режим работы FIFO передатчика; 1 – активация сброса FIFO передатчика. Сбрасывается в ноль автоматически по завершении сброса
3	SOD	Запрет выходных линий в режиме ведомого устройства. Бит используется только в режиме ведомого устройства (MS =1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. Бит SOD следует установить в случае, если данный ведомый модуль SSP не должен в настоящее время осуществлять передачу данных в линию SSP_TXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSP_TXD в ведомом режиме разрешено; 1 – управление линией SSP_TXD в ведомом режиме запрещено
2	MS	Выбор ведущего или ведомого режима работы: 0 – ведущий модуль (устанавливается по умолчанию); 1 – ведомый модуль
1	SSE	Разрешение работы приемопередатчика: 0 – работа запрещена; 1 – работа разрешена
0	LBM	Тестирование по шлейфу: 0 – нормальный режим работы приемопередатчика; 1 – выход регистра сдвига передатчика соединен со входом регистра сдвига приемника

23.14.2.3 **Регистр DR**

Регистр данных

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

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

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

В случае если выбран размер информационного слова менее 32 бит, перед записью в регистр SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника.

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

Назначение бит регистра DR представлено в таблице 252.

Таблица 249 – Формат регистра DR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	DATA	Принимаемые данные 32 бита (чтение). Передаваемые данные 32 бита (запись). В случае, если выбран размер информационного слова менее 32 бит, перед записью в регистр SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника

23.14.2.4 **Регистр SR**

Регистр состояния

Регистр SR доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика, и занятости модуля SSP.

Назначение бит регистра SR представлено в таблице 250.

Таблица 250 – Регистр SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…5	-	Зарезервировано. При чтении результат не определен. При записи следует заполнить нулями
4	BSY	Флаг занятости модуля: 0 – модуль SSP неактивен; 1 – модуль SSP в настоящее время передает и/или принимает данные, либо буфер FIFO передатчика не пуст

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	RFF	Буфер FIFO приемника заполнен: 0 – не заполнен; 1 – заполнен
2	RNE	Буфер FIFO приемника не пуст: 0 – пуст; 1 – не пуст
1	TNF	Буфер FIFO передатчика не заполнен: 0 – заполнен; 1 – не заполнен
0	TFE	Буфер FIFO передатчика пуст: 0 – не пуст; 1 – пуст

23.14.2.5 **Регистр CPSR**

Регистр делителя таковой частоты

Регистр CPSR используется для установки параметров делителя тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль. Если записать в регистр CPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Назначение бит регистра CPSR представлено в таблице 251.

Таблица 251 – Регистр CPSR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…8	-	Зарезервировано. При чтении результат не определен. При записи следует заполнить нулями
7…0	CPSDVSR	Коэффициент деления тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль

23.14.2.6 **Регистр IMSC**

Регистр установки и сброса маски прерывания

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание. При этом запись 1 в разряд разрешает соответствующее прерывание, запись 0 – запрещает.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение бит регистра IMSC представлено в таблице 252.

Таблица 252 – Регистр IMSC

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…7		Резерв. Не модифицируйте. При чтении выдаются нули
6	TNBSYIM	Маска прерывания по пустоте сдвигового регистра передатчика: 1 – не маскирована; 0 – маскирована
5	TFEIM	Маска прерывания по пустоте FIFO передатчика: 1 – не маскирована; 0 – маскирована
4	RNEIM	Маска прерывания по наличию данных в FIFO приемника: 1 – не маскирована; 0 – маскирована
3	TXIM	Маска прерывания по заполнению на 50 % и менее буфера FIFO передатчика: 1 – не маскирована; 0 – маскирована
2	RXIM	Маска прерывания по заполнению на 50 % и менее буфера FIFO приемника: 1 – не маскирована; 0 – маскирована
1	RTIM	Маска прерывания по таймауту приемника (буфер FIFO приемника не пуст и не было попуток его чтения в течение времени таймаута): 1 – не маскирована; 0 – маскирована
0	RORIM	Маска прерывания по переполнению буфера приемника: 1 – не маскирована; 0 – маскирована

23.14.2.7 **Регистр RIS**

Регистр состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит регистра RIS представлено в таблице 253.

Таблица 253 – Регистр RIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…7		Резерв. Не модифицируйте. При чтении выдаются нули
6	TNBSYRIS	Состояние до маскирования прерывания SSPTNBSYINTR
5	TFERIS	Состояние до маскирования прерывания SSPTFEINTR
4	RNERIS	Состояние до маскирования прерывания SSPRNEINTR
3	TXRIS	Состояние до маскирования прерывания SSPTXINTR
2	RXRIS	Состояние до маскирования прерывания SSPRXINTR
1	RTRIS	Состояние до маскирования прерывания SSPRTINTR
0	RORRIS	Состояние до маскирования прерывания SSPRORINTR

23.14.2.8 **Регистр MIS**

Регистр маскированного состояния прерываний

Регистр MIS доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит регистра MIS представлено в таблице 254.

Таблица 254 – Регистр MIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…7		Резерв. Не модифицируйте. При чтении выдаются нули
6	TNBSYMIS	Состояние маскированного прерывания SSPTNBSYINTR
5	TFEMIS	Состояние маскированного прерывания SSPTFEINTR
4	RNEMIS	Состояние маскированного прерывания SSPRNEINTR
3	TXMIS	Состояние маскированного прерывания SSPTXINTR
2	RXMIS	Состояние маскированного прерывания SSPRXINTR
1	RTMIS	Состояние маскированного прерывания SSPRTINTR
0	RORMIS	Состояние маскированного прерывания SSPRORINTR

23.14.2.9 **Регистр ICR**

Регистр сброса прерываний

Регистр ICR доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись в любой из разрядов регистра 0 игнорируется.

Назначение бит регистра ICR представлено в таблице 255.

Таблица 255 – Регистр ICR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…2	-	Зарезервировано. При записи следует заполнить нулями
1	RTIC	Сброс прерывания SSPRTINTR
0	RORIC	Сброс прерывания SSPRORINTR

23.14.2.10 **Регистр DMACR**

Регистр управления прямым доступом

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

Назначение бит регистра DMACR представлено в таблице 256.

Таблица 256 – Регистр DMACR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…2	-	Зарезервировано. При чтении выдаются нули. При записи следует заполнить нулями
1	TXDMAE	Использование DMA при передаче: 1 – разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика; 0 – запрещено формирование запросов DMA для обслуживания буфера FIFO передатчика
0	RXDMAE	Использование DMA при приеме: 1 – разрешено формирование запросов DMA для обслуживания буфера FIFO приемника; 0 – запрещено формирование запросов DMA для обслуживания буфера FIFO приемника

24 Контроллер UART

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

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

24.1 Основные характеристики модуля UART

Модуль UART может быть запрограммирован для использования, как в качестве универсального асинхронного приемопередатчика, так и для инфракрасного обмена данными (SIR).

Модуль содержит независимые буферы приема (16 слов по 13 бит) и передачи (16 слов по 9 бит) типа FIFO (First In First Out – первый вошел, первый вышел), что позволяет снизить интенсивность прерываний центрального процессора.

Программное отключение FIFO позволяет ограничить размер буфера одним словом.

Есть возможность программно настраивать скорость обмена данными, путем деления тактовой частоты опорного генератора в диапазоне (1x16 – 65535x16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц.

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

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

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

Также модуль обеспечивает обнаружение ложных стартовых бит.

Формирование и обнаружения сигнала разрыва линии.

Возможность организации аппаратного управления потоком данных.

Полностью программируемый UART имеет следующие характеристики:

– данные длиной 5, 6, 7, 8 или 9 бит;

– формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение, либо не передается);

– формирование 1 или 2 стоповых бит;

– скорость передачи данных – до UART_CLK/16 Бод.

Кодек ИК-обмена данными IrDA SIR обеспечивает:

– программный выбор обмена данными по линиям асинхронного приемопередатчика либо кодека ИК связи IrDA SIR;

– поддержку функционирования с информационной скоростью до 115200 бит/с в режиме полудуплекса;

– поддержку длительности бит для нормального режима (3/16) и для режима пониженного энергопотребления (1,41 – 2,23 мкс);

– программируемое деление опорной частоты UART_CLK для получения заданной длительности бит в режиме пониженного энергопотребления.

24.2 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

– скорость передачи данных – целая и дробная часть числа;

– количество бит данных;

– количество стоповых бит;

– режим контроля четности;

– разрешение или запрет использования буферов FIFO (глубина очереди данных – 16 элементов или один элемент, соответственно);

– порог срабатывания прерывания по заполнению буферов FIFO (1/8, 1/4, 1/2, 3/4 и 7/8);

– частота внутреннего тактового генератора (номинальное значение – 1,8432 МГц) может быть задана в диапазоне от 1,42 до 2,12 МГц для обеспечения возможности формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления (для ИК-обмена);

– режим аппаратного управления потоком данных.

24.3 Отличия от контроллера UART 16C650

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

– пороги срабатывания прерывания по заполнению буфера FIFO приемника – 1/8, 1/4, 1/2, 3/4 и 7/8;

– пороги срабатывания прерывания по заполнению буфера FIFO передатчика – 1/8, 1/4, 1/2, 3/4 и 7/8;

– отличается распределение адресов внутренних регистров и назначение бит в регистрах;

– недоступны изменения сигналов состояния модема.

Следующие возможности контроллера 16C650 не поддерживаются:

– полуторная длительность стопового бита (поддерживается только 1 или 2 стоповых бита);

– независимое задание тактовой частоты приемника и передатчика.

24.4 Функциональные возможности

Устройство выполняет следующие функции:

– преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму;

– преобразование данных, передаваемых на периферийное устройство, из параллельной в последовательную форму.

Процессор читает и записывает данные, а также управляющую информацию и информацию о состоянии модуля. Прием и передача данных буферизуются с помощью внутренней памяти FIFO, позволяющей сохранить до 16 слов по 13 бит для режима приема и до 16 слов по 9 бит для режима передачи.

Модуль приемопередатчика:

– содержит программируемый генератор, формирующий тактовый сигнал одновременно для передачи и для приема данных на основе внутреннего тактового сигнала UART_CLK;

– обеспечивает возможности, сходные с возможностями индустриального стандарта – контроллера UART 16C650.

Режим работы приемопередатчика и скорость обмена данными контролируются регистром управления линией LCR_H регистрами делителя скорости передачи данных – целой части (IBRD) и дробной части (FBRD).

Устройство может формировать следующие сигналы:

– независимые маскируемые прерывания от приемника (в том числе по таймауту), передатчика, а также по изменению состояния модема и в случае обнаружения ошибки;

– общее прерывание, возникающее в случае, если возникло одно из независимых немаскированных прерываний;

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

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

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

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

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

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

Блок последовательного интерфейса инфракрасной передачи данных в соответствии с протоколом IrDA SIR реализует протокол обмена данными ENDEC. В случае его активизации обмен информацией осуществляется не с помощью сигналов UARTTXD и UARTRXD, а посредством сигналов nSIROUT и SIRIN.

В этом случае устройство переводит линию UARTTXD в пассивное состояние (высокий уровень), и перестает реагировать на изменение состояния модема, а также сигнала на линии UARTRXD. Протокол SIR ENDEC обеспечивает возможность обмена данными исключительно в режиме полудуплекса, то есть он не может передавать во время приема данных и принимать во время передачи данных.

В соответствии со спецификацией физического уровня протокола IrDA SIR, задержка между передачей и приемом должна составлять не менее 10 мс.

24.5 **Описание функционирования блока UART**

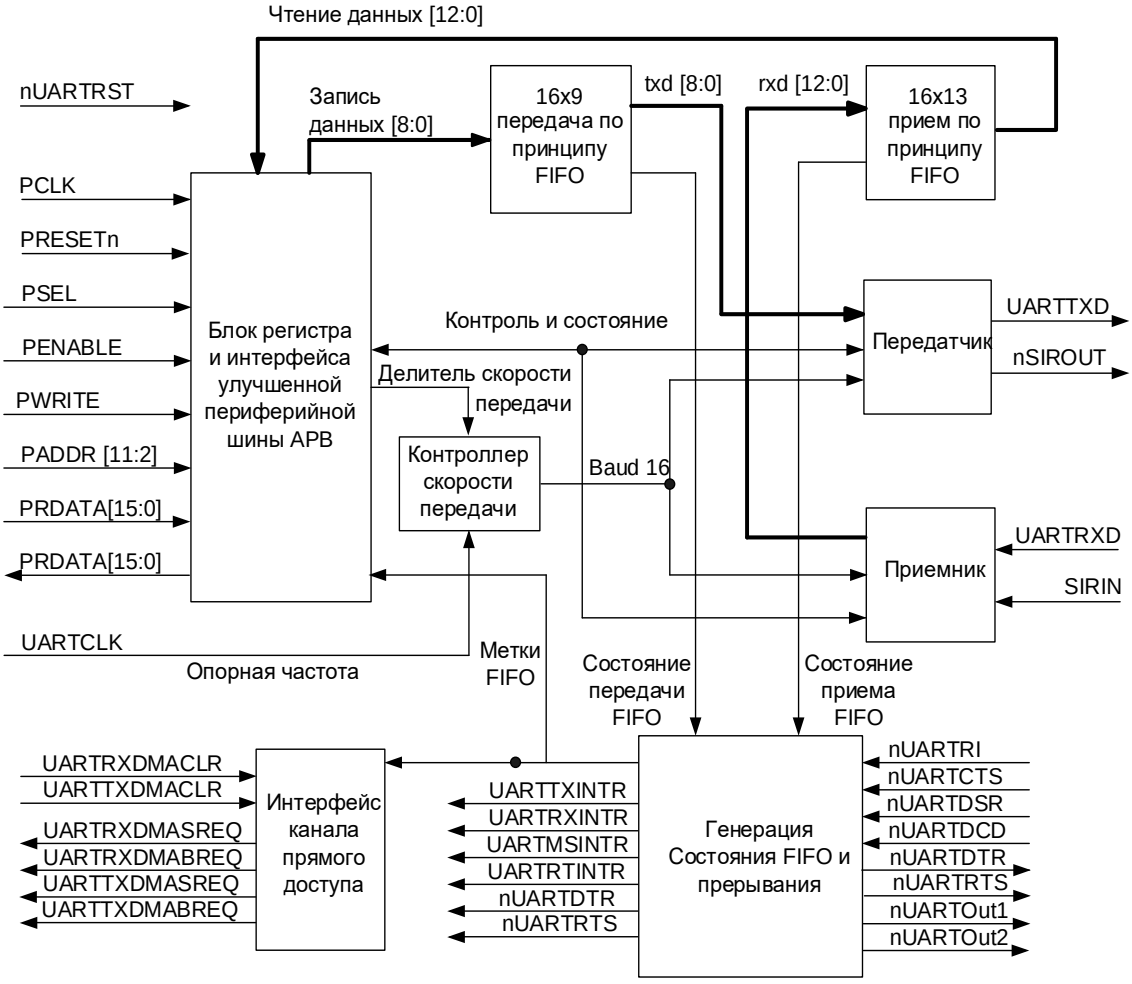


Рисунок 79 – Блок-схема универсального асинхронного приёмопередатчика (UART)

24.5.1 Генератор тактового сигнала приемопередатчика

Генератор содержит счетчики без цепи сброса, формирующие внутренние тактовые сигналы Baud16 и IrLPBaud16.

Сигнал Baud16 используется для синхронизации схем управления приемником и передатчиком последовательного обмена данными. Он представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UART_CLK и частотой, в 16 раз выше скорости передачи данных.

Сигнал IrLPBaud16 предназначен для синхронизации схемы формирования импульсов с длительностью, требуемой для ИК обмена данными в режиме с пониженным энергопотреблением.

24.5.2 Буфер FIFO передатчика

Буфер передатчика имеет ширину 9 бит, глубину 16 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Данные от центрального процессора, записанные через шину APB, сохраняются в буфере до тех пор, пока не будут считаны логической схемой передачи данных. Существует возможность запретить буфер FIFO передатчика, в этом случае он будет функционировать как однобайтовый буферный регистр.

24.5.3 Буфер FIFO приемника

Буфер приемника имеет ширину 13 бит, глубину 16 слов, схему организации доступа типа FIFO («первый вошел, первый вышел»). Принятые от периферийного устройства данные и соответствующие коды ошибок сохраняются логикой приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину APB. Буфер FIFO приемника может быть запрещен, в этом случае он будет действовать как однобайтовый буферный регистр.

24.5.4 Блок передатчика

Логические схемы передатчика осуществляют преобразование данных, считанных из буфера передатчика, из параллельной в последовательную форму. Управляющая логика выдает последовательный поток бит в порядке: стартовый бит, биты данных, начиная с младшего значащего разряда, бит проверки на четность, и, наконец, стоповые биты, в соответствии с конфигурацией, записанной в регистре управления.

24.5.5 Блок приемника

Логические схемы приемника преобразуют данные, полученные от периферийного устройства, из последовательной в параллельную форму после обнаружения корректного стартового импульса. Кроме того, производятся проверки на: переполнение буфера, ошибки контроля четности, ошибки в структуре сигнала и на разрыв линии. Признаки обнаружения этих ошибок также сохраняются в выходном буфере.

24.5.6 Блок формирования прерываний

Контроллер генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения указанных независимых прерываний по схеме ИЛИ.

Комбинированный сигнал прерывания может быть подан на внешний контроллер прерываний системы, при этом появится дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

Другой подход состоит в подаче на системный контроллер прерываний независимых сигналов запроса на прерывание от приемопередатчика. В этом случае процедура обработки сможет одновременно считать информацию обо всех источниках прерываний. Данный подход привлекателен в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

Для более подробной информации см. подраздел «Прерывания».

24.5.7 Интерфейс прямого доступа к памяти

Модуль обеспечивает интерфейс с контроллером DMA согласно схеме взаимодействия приемопередатчика и контроллера DMA.

24.5.8 Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов CPU_CLK и UART_CLK. Регистры синхронизации и логика квитирования постоянно находятся в активном состоянии. Это практически не отражается на характеристиках устройства. Синхронизация сигналов управления осуществляется в обоих направлениях потока данных, то есть как из области действия CPU_CLK в область действия UART_CLK, так и наоборот.

24.6 Описание функционирования ИК кодека IrDA SIR

Структурная схема кодека представлена на рисунке 80.

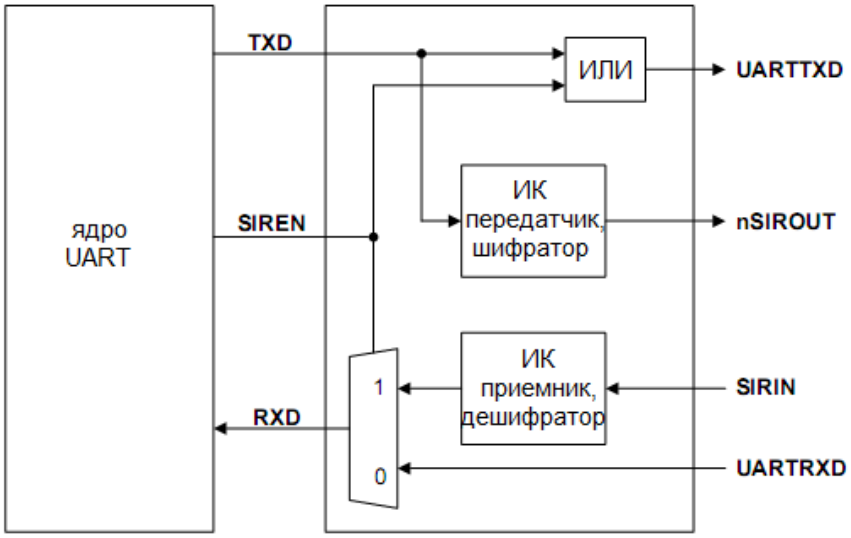


Рисунок 80 – Структурная схема кодека IrDA

24.6.1 **Кодер ИК передатчика**

Кодер преобразует поток данных с выхода асинхронного передатчика, сформированный по закону модуляции без возврата к нулю (NRZ). Спецификация физического уровня протокола IrDA SIR подразумевает использование модуляции с возвратом к нулю и инверсией (RZI), в соответствии с которой передача логического нуля соответствует излучению одного светового ИК импульса. Сформированный выходной поток импульсов подается на усилитель и далее на ИК светодиод.

Длительность импульса в режиме IrDA составляет, согласно спецификации, 3 периода внутреннего тактового генератора с частотой Baud16, то есть 3/16 периода времени, выделенного на передачу одного бита.

В режиме IrDA с пониженным энергопотреблением ширина импульса задана как 3/16 периода, выделенного на передачу бита, при скорости передачи данных 115200 бит/с. Данное требование реализуется за счет формирования трех периодов тактового сигнала IrLPBaud16 с номинальной частотой 1,8432 МГц, в свою очередь, формируемого путем деления частоты UART_CLK. Значение частоты IrLPBaud16 задается путем записи соответствующего коэффициента деления частоты в регистр ILPR.

Выход кодера имеет активное низкое состояние. При передаче логической единицы выход кодера остается в низком состоянии, при передаче логического нуля – формируется импульс, при этом выход кратковременно переводится в высокое состояние.

Как в нормальном режиме, так и в режиме пониженного энергопотребления, использование нецелых значений коэффициента деления скорости передачи данных увеличивает джиттер («дребезжание») фронтов импульсов данных. Наличие джиттера в случае использования дробных коэффициентов деления связано с тем, что интервалы между тактовыми импульсами Baud16 будут нерегулярными – период сигнала Baud16 в разное время будет содержать различное количество периодов сигнала UART_CLK. Можно показать, что в наихудшем случае величина джиттера в потоке ИК импульсов может достигать трех периодов UART_CLK. В соответствии со спецификацией стандарта IrDA SIR, джиттер не должен превышать величины 13 %. В случае, если частота сигнала UART_CLK составляет более 3,6834 МГц, а скорость передачи данных меньше или равна 115200 бит/с, величина джиттера не превышает 9 %. Таким образом, требования стандарта выполняются.

24.6.2 **Декодер ИК приемника**

Декодер преобразует поток данных, сформированных по закону возврата к нулю, полученного от приемника ИК сигнала, и выдает поток данных без возврата к нулю на вход приемника UART. В неактивном состоянии вход декодера находится в высоком состоянии. Выходной сигнал кодера имеет полярность, противоположную полярности входа декодера.

Обнаружение стартового бита осуществляется при низком уровне сигнала на входе декодера.

Примечание – Для того чтобы исключить ложные срабатывания UART от импульсных помех, на входе SIRIN игнорируются импульсы с длительностью менее, чем:

- 3/16 длительности Baud16 в режиме IrDA;
- 3/16 длительности IrLPBaud16 в режиме IrDA с пониженным энергопотреблением.

24.7 Описание работы UART

24.7.1 Сброс модуля

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

24.7.2 Тактовые сигналы

Частота тактового сигнала F_{UART_CLK} должна обеспечивать поддержку требуемого диапазона скоростей передачи данных:

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

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

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота F_{UART_CLK} должна находиться в интервале от 7,3728 до 115,34 МГц.

Частота F_{UART_CLK}, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами F_{CPU_CLK} и F_{UART_CLK}. Частота F_{UART_CLK} должна быть не более чем в 5/3 раз выше частоты F_{CPU_CLK}.

$$F_{UART_CLK} \leq \frac{5}{3} \cdot F_{CPU_CLK}.$$

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

24.7.3 Работа универсального асинхронного приемопередатчика

Управляющая информация хранится в регистре управления линией LCR. Этот регистр имеет внутреннюю ширину 30 бит, однако внешний доступ по шине APB к нему осуществляется через следующие регистры:

– LCR_H – определяет:
  – параметры передачи данных;
  – длину слова;
  – режим буферизации;
  – количество передаваемых стоповых бит;
  – режим контроля четности;
  – формирование сигнала разрыва линии;
– IBRD – определяет целую часть коэффициента деления для скорости передачи данных;
– FBRD – определяет дробную часть коэффициента деления для скорости передачи данных.

Примечание – Изменение значений трех регистров можно осуществить корректно двумя способами:

– запись IBRD, запись FBRD, запись LCR_H;
– запись FBRD, запись IBRD, запись LCR_H.

Чтобы изменить значение лишь одного из регистров (IBRD или FBRD), необходимо выполнить следующие шаги:

– запись IBRD (или FBRD), запись LCR_H.

24.7.4 **Коэффициент деления частоты**

Коэффициент деления для формирования скорости передачи данных состоит из 22 бит, при этом 16 бит выделено для представления его целой части, а 6 бит – дробной части. Возможность задания нецелых коэффициентов деления позволяет осуществлять обмен данными со стандартными информационными скоростями, не превышающими 230400, при этом используя в качестве UART_CLK тактовый сигнал с произвольной частотой более 3,6864 МГц.

Целая часть коэффициента деления записывается в 16-битный регистр IBRD. Шестиразрядная дробная часть записывается в регистр FBRD. Значение коэффициента деления связано с содержимым указанных регистров следующим образом

$$\text{Коэффициент деления} = \frac{F_{UART_CLK}}{16 \cdot \text{скорость передачи данных}} = IBRD + FBRD\ , \quad (14)$$

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

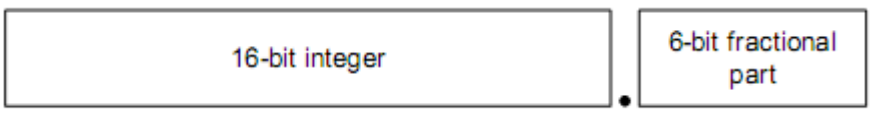


Рисунок 81 – Коэффициент деления

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

$$m = \text{integer}(\text{FBRD} \cdot 2^n + 0{,}5\,, \qquad (15)$$

где *integer* – операция отсечения дробной части числа, n = 6.

В модуле формируется внутренний сигнал Baud16, представляющий собой последовательность импульсов с длительностью, равной периоду сигнала UART_CLK и средней частотой, в 16 раз большей требуемой скорости обмена данными.

24.7.5 **Передача и прием данных**

Принятые или передаваемые данные заносятся в 16-элементные буферы FIFO, при этом каждый элемент приемного буфера FIFO кроме байта данных хранит также четыре бита информации о состоянии модема.

Данные для передачи заносятся в буфер FIFO передатчика. Если работа приемопередатчика разрешена, начинается передача информационного кадра с параметрами, указанными в регистре управления линией LCR_H. Передача данных продолжается до опустошения буфера FIFO передатчика. После записи элемента в буфер FIFO передатчика сигнал BUSY переходит в высокое состояние. Это состояние сохраняется в течение всего времени передачи данных. В низкое состояние сигнал BUSY переходит только после того, как буфер FIFO передатчика станет пуст, а последний бит данных (включая стоповые биты) будет передан. Сигнал BUSY может находиться в высоком состоянии даже в случае, если приемопередатчик будет переведен из разрешенного состояния в запрещенное.

Для каждого бита данных (в приемной линии) производится три измерения уровня, решение принимается по мажоритарному принципу.

В случае если приемник находился в неактивном состоянии (на линии входного сигнала UARTRXD постоянно присутствует единица) и произошел переход входного сигнала из высокого в низкий логический уровень (обнаружен стартовый бит), включается счетчик, тактируемый сигналом Baud16, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов (в режиме асинхронного приемопередатчика) или каждые четыре такта (в режиме ИК обмена данными) сигнала Baud16. Более частая выборка данных в режиме ИК обмена связана с необходимостью корректной обработки импульсов данных согласно протоколу SIR IrDA.

Стартовый бит считается достоверным в случае, если сигнал на линии UARTRXD сохраняет низкий логический уровень в течение восьми отсчетов сигнала Baud16 с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

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

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

24.7.6 **Биты ошибки**

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся на позиции [11:9] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO, расположенный на позиции 12-го слова данных.

Таблица 257 показывает назначение всех битов слова данных в FIFO буфере приемника.

Таблица 257 – Назначение бит слова данных в FIFO-буфере приемника

Бит буфера FIFO	Назначение
12	Признак переполнения буфера
11	Ошибка – разрыв линии
10	Ошибка проверки на четность
09	Ошибка формирования кадра
08…00	Принятые данные

24.7.7 **Бит переполнения буфера**

Бит переполнения непосредственно не связан с конкретным символом в буфере приемника. Признак переполнения фиксируется в случае, если буфер FIFO заполнен к моменту, когда очередной символ данных полностью принят (находится в регистре сдвига). При этом данные из регистра сдвига не попадают в буфер приемника и теряются с началом приема очередного символа. Как только в буфере приемника появляется свободное место, очередной принятый символ данных заносится в буфер FIFO вместе с текущим значением признака переполнения. После успешной записи данных в буфер признак переполнения сбрасывается.

24.7.8 **Запрет буфера FIFO**

Предусмотрена возможность отключения FIFO буферов приемника и передатчика. В этом случае приемная и передающая сторона контроллера UART располагают лишь однобайтными буферными регистрами. Бит переполнения буфера

устанавливается при этом тогда, когда очередной символ данных уже принят, однако предыдущий еще не был считан.

В настоящей реализации модуля буферы FIFO физически не отключаются, необходимая функциональность достигается за счет логических манипуляций с флагами. При этом в случае, если буфер FIFO отключен, а сдвиговый регистр передатчика пуст (не используется), запись байта данных происходит непосредственно в регистр сдвига, минуя буферный регистр.

24.7.8.1 **Проверка по шлейфу**

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

24.7.9 **Работа кодека ИК обмена данными IrDA SIR**

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

Предусмотрено два режима работы:

– Режим IrDA – уровень логического нуля передается на линию nSIROUT в виде импульса с высоким логическим уровнем и длительностью 3/16 от выбранного периода следования бит данных. Логическая единица при этом передается в виде постоянного низкого уровня сигнала. Сформированный выходной сигнал далее подается на передатчик ИК-сигнала, обеспечивая излучение светового импульса всякий раз при передаче нулевого бита. На приемной стороне световые импульсы воздействуют на базу фототранзистора ИК приемника, который в результате формирует низкий логический уровень. Это, в свою очередь, обуславливает низкий уровень на входе SIRIN.

– Режим IrDA с пониженным энергопотреблением – длительность передаваемых импульсов ИК излучения устанавливается в три раза больше длительности импульсов внутреннего опорного сигнала IrLPBaud16 (равной 1,63 мкс при номинальной частоте 1,8432 МГц). Данный режим активизируется путем установки бита SIR_LP в регистре управления CR.

Как в режиме IrDA, так и в режиме IrDA с пониженным энергопотреблением:

– кодирование осуществляется на основе бит данных, сформированных асинхронным передатчиком модуля;

– в ходе приема данных декодированные биты далее обрабатываются блоком асинхронного приема.

В соответствии со спецификацией физического уровня протокола IrDA SIR, обмен данными должен осуществляться в режиме полудуплекса, при этом задержка между передачей и приемом данных должна составлять не менее 10 мс. Эта задержка должна формироваться программно. Необходимость ее введения обусловлена тем, что воздействие передающего ИК светодиода на находящийся рядом ИК приемник может

привести к искажению принимаемого сигнала или даже ввести приемный тракт в состояние насыщения. Задержка между окончанием передачи и началом приема данных именуется латентность, или время установки (готовности) приемника.

Сигнал IrLPBaud16 формируется путем деления частоты сигнала UART_CLK в соответствии с коэффициентом деления, записанным в регистре ILPR.

Коэффициент деления вычисляется по формуле

$$\text{Коэффициент деления} = \frac{F_{UART_CLK}}{IrLPBaud16}, \quad (16)$$

где номинальное значение *IrLPBaud16* составляет 1,8432 МГц. Коэффициент деления должен быть выбран так, чтобы выполнялось соотношение

$$1{,}48\ \text{МГц} < IrLPBaud16 < 2{,}12\ \text{МГц}\,. \quad (17)$$

24.7.9.1 **Проверка по шлейфу**

Проверка по шлейфу выполняется после установки в 1 бита LBE регистра управления контроллером CR с одновременной установкой в 1 бита SIRTEST регистра управления тестированием TCR.

В этом режиме данные, передаваемые на выход nSIROUT, должны подаваться на вход SIRIN.

Примечание – Это единственный случай использования тестового регистра в нормальном режиме функционирования модуля.

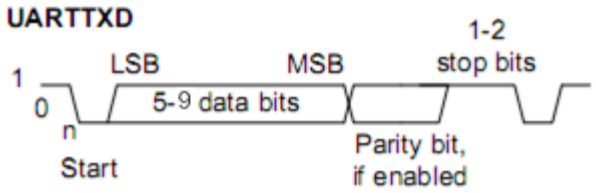


Рисунок 82 – Кадр передачи данных

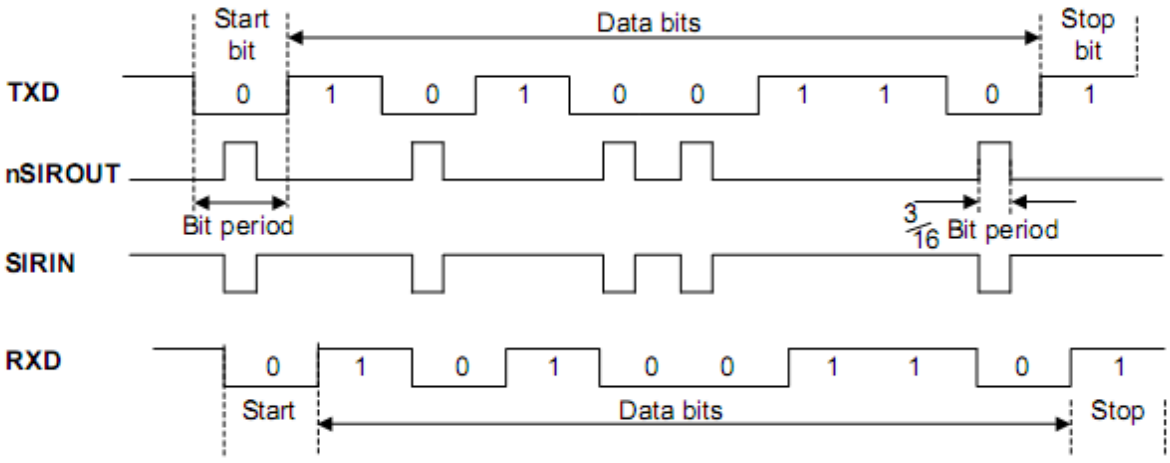


Рисунок 83 – Модуляция данных IrDA

24.8 Линии управления модемом

Модуль универсального асинхронного приемопередатчика может использоваться как в режиме оконечного оборудования (DTE), так и в режиме оборудования передачи данных (DCE).

Таблица 258 – Назначение управления модемом в режимах DTE и DCE

Сигнал	Назначение	
	Режим оконечного оборудования	Режим оборудования передачи данных
nUARTCTS	Готов к передаче данных	Запрос передачи данных
nUARTDSR	Источник данных готов	Приемник данных готов
nUARTDCD	Обнаружен информационный сигнал	-
nUARTRI	Индикатор вызова	-
nUARTRTS	Запрос передачи данных	Готов к передаче данных
nUARTDTR	Приемник данных готов	Источник данных готов
nUARTOut1	-	Обнаружен информационный сигнал
nUARTOut2	-	Индикатор вызова

24.8.1 Аппаратное управление потоком данных

Программно активизируемый режим аппаратного управления потоком данных позволяет контролировать (приостанавливать и возобновлять) информационный обмен с помощью сигналов nUARTRTS и nUARTCTS. Иллюстрация взаимодействия двух устройств последовательной связи с аппаратным управлением потоком данных представлена на рисунке 84.

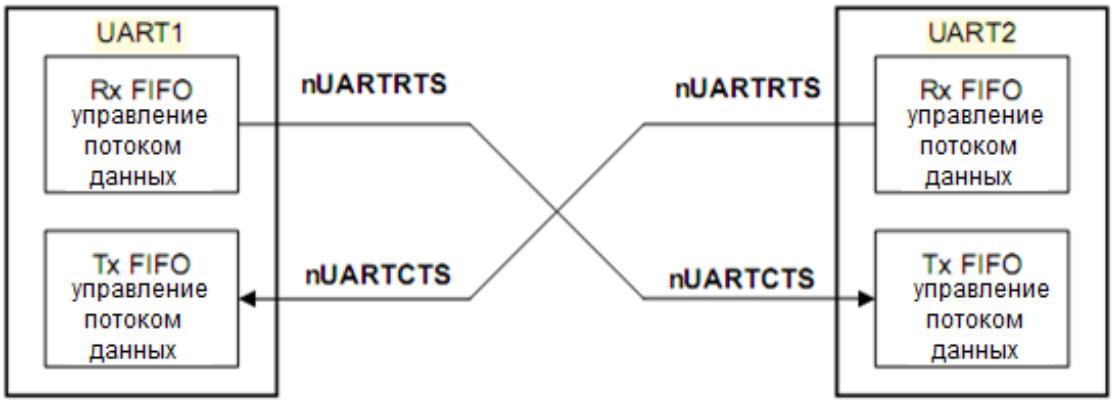


Рисунок 84 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

Если разрешено управление потоком данных по сигналу RTS, линия nUARTRTS переводится в активное состояние только после того, как в FIFO буфере приема появляется заданное количество свободных элементов.

Если разрешено управление потоком данных по сигналу CTS, передача данных осуществляется только после перевода линии nUARTCTS в активное состояние.

Режим аппаратного управления потоком данных задается путем установки значений бит RTSEn и CTSEn в регистре управления CR. В таблице 259 показаны необходимые установки для различных режимов управления потоком данных.

Таблица 259 – Режимы управления потоком данных

CTSEn	RTSEn	Описание
1	1	Разрешено управление потоком данных по CTS и RTS
1	0	Управления потоком данных осуществляется по линии CTS
0	1	Управления потоком данных осуществляется по линии RTS
0	0	Управления потоком данных запрещено

Примечание – В случае если выбран режим управления потоком данных по RTS, программное обеспечение не может использовать бит RTS регистра CR для проверки состоянии линии RTS.

24.8.2 **Управление потоком данных по линии RTS**

Логика управления потоком данных по RTS использует данные о превышении пороговых уровней заполнения буфера FIFO приемника. В случае выбора режимов с управлением по RTS, сигнал на линии nUARTRTS переводится в активное состояние только после того, как в FIFO буфере приема появляется заданное количество свободных элементов. После достижения порогового уровня заполнения буфера приемника сигнал nUARTRTS снимается (переводится в пассивное состояние), указывая, таким образом, на отсутствие свободного места для сохранения принятых данных. При этом дальнейшая передача данных должна быть прекращена по завершении передачи текущего символа.

Обратно в активное состояние сигнал nUARTRTS переводится после считывания данных из приемного буфера FIFO в количестве, достаточном для того, чтобы заполнение буфера оказалось ниже порогового уровня.

В случае если управление потоком данных по RTS запрещено, однако работа приемопередатчика UART разрешена, прием будет осуществляться до полного заполнения буфера FIFO, либо до завершения передачи данных.

24.8.3 **Управление потоком данных по линии CTS**

В случае выбора одного из режимов с управлением потоком данных по CTS передатчик осуществляет проверку состояния линии nUARTCTS перед началом передачи очередного байта данных. Передача осуществляется только в случае, если данная линия активна, и продолжается до тех пор, пока активное состояние линии сохраняется и буфер передатчика не пуст.

При переходе линии nUARTCTS в неактивное состояние модуль завершает выдачу текущего передаваемого символа, после чего передача данных прекращается.

Если управление потоком данных по CTS запрещено, и при этом работа приемопередатчика UART разрешена – данные будут выдаваться до опустошения буфера FIFO передатчика.

24.9 **Интерфейс прямого доступа к памяти**

Модуль универсального асинхронного приемопередатчика оснащен интерфейсом подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMA DMACR.

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

– UARTRXDMASREQ – запрос передачи отдельного символа, инициируется контроллером UART. Размер символа в режиме приема данных – до 13 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ.

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

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

Для передачи:

– UARTTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Размер символа в режиме передачи данных – до 9 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку.

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

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

Сигналы блочного и одноэлементного обмена данными не являются взаимно исключающими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение, формируется как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае если количество данных в буфере приема меньше порогового значения формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока. Пусть, например, нужно

принять 19 символов, а порог заполнения буфера FIFO установлен равным четырем. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов.

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

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

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

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

Когда буферы FIFO включены, обмен данными может производиться в ходе как одноэлементных, так и блочных передач данных, в зависимости от установленной величины порога заполнения буферов и их фактического заполнения. Таблица 260 показывает значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

Таблица 260 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/8	14	2
¼	12	4
½	8	8
¾	4	12
7/8	2	14

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

Рисунок 85 показывает временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть

синхронизированы с CPU_CLK. В интересах ясности изложения предполагается, что синхронизация сигналов запроса DMA в контроллере DMA не производится.

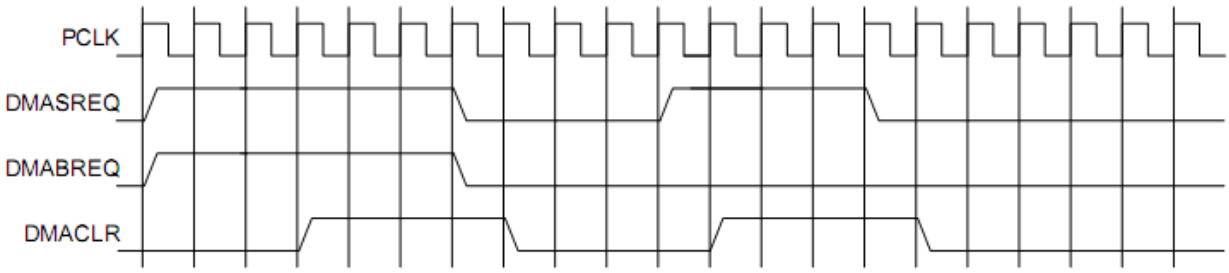


Рисунок 85 – Временные диаграммы одноэлементного и блочного запросов DMA

24.10 **Прерывания**

В модуле предусмотрено 11 маскируемых источников прерывания. В результате формируется один общий сигнал, представляющий собой комбинацию независимых сигналов, объединенных по схеме ИЛИ.

Сигналы запроса на прерывание:

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

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски UARTIMSC. Установка бита в 1 разрешает соответствующее прерывание, в 0 – запрещает.

Доступность, как индивидуальных линий, так и общей линии запроса позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика UARTRXINTR и UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

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

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

Прерывание UARTRNEINTR возникает в случае, если FIFO приемника получает хотя бы одно слово данных, то есть становится не пусто.

Прерывание UARTTFEINTR выставляется, если FIFO передатчика не имеет никаких данных, то есть пусто в данный момент.

UARTTNBSYINTR выставляется в том случае, если сдвиговый регистр передатчика пуст.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний UARTRIS, либо из маскированного регистра прерываний UARTMIS.

24.10.1 **UARTMSINTR**

Прерывание по состоянию модема возникает в случае изменения любой из линий состояний модема (nUARTCTS, nUARTDCD, nUARTDSR, nUARTRI). Сброс прерывания осуществляется путем записи 1 в соответствующий (в зависимости от линии состояния модема, вызвавшей прерывание) разряд регистра сброса прерывания ICR.

24.10.2 **UARTRXINTR**

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

– буфер FIFO разрешён и его заполнение достигло заданного порогового значения. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения данных из буфера приемника до тех пор, пока его заполнение не станет меньше порога, либо после сброса прерывания;

– буфер FIFO запрещен (имеет размер один символ), принят один символ данных. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения одного байта данных, либо после сброса прерывания.

24.10.3 **UARTTXINTR**

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

– буфер FIFO разрешён и его заполнение меньше или равно заданному пороговому значению. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи данных в буфера передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания;

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

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

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

24.10.4 **UARTRTINTR**

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения (или считывания одного байта в случае, если буфер FIFO запрещен), либо путем записи 1 в соответствующий бит регистра сброса прерывания ICR.

24.10.5 **UARTEINTR**

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

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

Причину возникновения прерывания можно определить, прочитав содержимое регистра прерываний RIS, либо содержимое маскированного регистра прерываний MIS.

Сброс прерывания осуществляется путем записи соответствующих бит в регистр сброса прерывания ICR. За прерываниями по обнаружению ошибки закреплены биты с 7 по 10.

24.10.6 **UARTINTR**

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR, UARTEINTR, UARTRNEINTR, UARTTFEINTR и UARTTNBSYINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

24.11 **Программное управление модулем**

Следующая информация применима ко всем регистрам контроллера:

– Базовый адрес контроллера фиксирован. Смещение каждого регистра относительно базового адреса постоянно.

– Не следует пытаться получить доступ к зарезервированным или неиспользуемым адресам. Это может привести к непредсказуемому поведению модуля.

– За исключением специально оговоренных в настоящей спецификации случаев:
  – не следует изменять значения не определенных в спецификации разрядов регистров;
  – не следует использовать значения не определенных в спецификации разрядов регистров;
  – все биты регистров (за исключением специально оговоренных случаев) устанавливаются в значение 0 после сброса по включению питания или системного сброса.

– Столбец «Тип» таблицы 261 определяет режим доступа к регистру в соответствии с обозначениями:
  – RW – чтение и запись;
  – RO – только чтение;
  – WO – только запись.

24.12 **Описание регистров контроллера UART**

Данные о регистрах контроллера UART приведены в таблице 261.

Таблица 261 – Обобщенные данные о регистрах устройства

Базовый Адрес	Название				Описание
0x50008000	UART1				Регистры контроллера интерфейса UART1
0x50010000	UART2				Регистры контроллера интерфейса UART2
0x50038000	UART3				Регистры контроллера интерфейса UART3
0x500E8000	UART4				Регистры контроллера интерфейса UART4
Смещение	Наимено-вание	Тип	Значение после сброса	Размер, бит	Описание
0x000	DR	RW	0x---	13/9	Регистр данных
0x004	RSR_ECR	RW	0x0	4/0	Регистра состояния приемника / Сброс ошибки приемника
0x008-0x014					Резерв
0x018	FR	RO	0b-10010---	9	Регистр флагов
0x01C					Резерв
0x020	ILPR	RW	0x00	8	Регистр управления ИК обменом в режиме пониженного энергопотребления
0x024	IBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	FBRD	RW	0x00	6	Дробная часть делителя скорости обмена данными
0x02C	LCR_H	RW	0x00	9	Регистр управления линией
0x030	CR	RW	0x0300	16	Регистр управления
0x034	IFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	IMSC	RW	0x000	14	Регистр маски прерывания
0x03C	RIS	RO	0x300-	14	Регистр состояния прерываний
0x040	MIS	RO	0x00-	14	Регистр состояния прерываний с маскированием
0x044	ICR	WO	-	11	Регистр сброса прерывания
0x048	DMACR	RW	0x00	3	Регистр управления DMA
0x080	TCR	RW	0x00	3	Регистр управления тестированием

24.12.1 **Регистр данных DR**

В ходе передачи данных:

Если буфер FIFO передатчика разрешен, то слово данных, записанное в рассматриваемый регистр, направляется в буфер FIFO передатчика.

В противном случае, записанное слово фиксируется в буферный регистр передатчика (последний элемент буфера FIFO).

Операция записи в регистр инициирует передачу данных. Слово данных предваряется стартовым битом, дополняется битом контроля четности (если режим контроля четности включен) и стоповым битом. Сформированное слово отправляется в линию передачи данных.

В ходе приема данных:

Если буфер FIFO приемника разрешен, байт данных и четыре бита состояния (разрыв, ошибка формирования кадра, четность, переполнение) сохраняются в 12-битном буфере.

В противном случае байт данных и биты состояния записываются в буферный регистр (последний элемент буфера FIFO).

Полученные из линии связи байты данных считывается путем чтения из регистра UART_DR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра RSR_ECR (таблица 262).

Таблица 262 – Формат регистра UARTDR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…13	-	Зарезервировано
12	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время как буфер заполнен. Сбрасывается в 0 после того, как в буфере появится свободное место
11	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита
10	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией LCR_H. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер
9	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер
8…0	DATA	Принимаемые данные (чтение) Передаваемые данные (запись)

Примечание – Необходимо запрещать работу приемопередатчика перед любым перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние во время передачи или приема символа, то перед остановкой он завершает выполняемую операцию.

24.12.2 **Регистр состояния приемника / сброса ошибки RSR_ECR**

Состояние приемника также может быть считано из регистра RSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра относится к последнему символу, считанному из регистра данных DR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним, считанным из регистра DR, байтом данных).

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

Назначения битов регистра RSR_ECR приведены в таблице 263.

Таблица 263 – Регистр RSR_ECR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7…4	-	Зарезервировано. При чтении результат не определен
3	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время как буфер заполнен. Сбрасывается в 0 после записи в регистр ECR. Содержимое буфера остается верным, так как перезаписан был только регистр сдвига. Центральный процессор должен считать данные для того, чтобы освободить буфер FIFO
2	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). Бит сбрасывается в 0 после записи в регистр ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящемся на вершине буфера. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита
1	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией LCR_H. Бит сбрасывается в 0 после записи в регистр ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера
0	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). Бит сбрасывается в 0 после записи в регистр ECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера

Примечание – Перед чтением регистра состояния RSR необходимо считать данные, принятые из линии, путем обращения к регистру данных DR. Противоположная последовательность действий не допускается, так как регистр RSR обновляет свое состояние

только после чтения регистра DR. Вместе с тем, информация о состоянии приемника может быть получена непосредственно из регистра данных DR.

24.12.3 **Регистр флагов FR**

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. Назначения битов регистра приведены в таблице 264.

Таблица 264 – Регистр FR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…9		Резерв. Не модифицируйте. При чтении заполняются нулями
8	RI	Инверсия линии nUARTRI
7	TXFE	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр передатчика пуст. В противном случае он равен 1, если пуст буфер FIFO передатчика. Данный бит не дает никакой информации о наличии данных в регистре сдвига передатчика
6	RXFF	Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника занят. В противном случае он равен 1, если заполнен буфер FIFO приемника
5	TXFF	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит равен 1, когда буферный регистр передатчика занят. В противном случае он равен 1, если заполнен буфер FIFO передатчика
4	RXFE	Буфер FIFO приемника пуст. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника пуст. В противном случае он равен 1, если пуст буфер FIFO приемника
3	BUSY	UART занят. Бит устанавливается в 1 в случае, если контроллер передает в линию данные. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Кроме того, бит занятости устанавливается в 1 при наличии данных в буфере FIFO передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен)
2	DCD	Инверсия линии nUARTDCD
1	DSR	Инверсия линии nUARTDSR
0	CTS	Инверсия линии nUARTCTS

24.12.4 **Регистр управления ИК обменом в режиме пониженного энергопотребления ILPR**

Этот восьмиразрядный регистр, доступный для чтения и записи, содержит значение коэффициента деления частоты UART_CLK, для формирования тактового сигнала IrLPBaud16. Назначение разрядов регистра показано в таблице 265.

Требуемое значение коэффициента деления для формирования сигнала IrLPBaud16 вычисляется по формуле

$$\text{ILPDVSR} = \frac{F_{UART_CLK}}{F_{IrLPBaud16}}, \quad (18)$$

где номинальное значение частоты $F_{IrLPBaud16}$ составляет 1,8432 МГц.

Коэффициент деления должен быть установлен таким образом, чтобы выполнялось соотношение

$$1{,}48 \text{ МГц} < F_{IrLPBaud16} < 2{,}12 \text{ МГц}, \quad (19)$$

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

Таблица 265 – Регистр ILPR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7…0	ILPDVSR	Коэффициент деления частоты UART_CLK, для формирования тактового сигнала IrLPBaud16. После сброса устанавливается в 0. Примечание – Коэффициент 0 – запрещенное значение. В случае его установки импульсы IrLPBaud16 формироваться не будут

Примечание – В интересах подавления помех, при работе в режиме IrDA с пониженным энергопотреблением кодек игнорирует поступающие на вход SIRIN импульсы с длительностью, меньшей трех периодов сигнала IrLPBaud16.

24.12.5 **Регистр целой части делителя скорости передачи данных IBRD**

Назначение бит регистра представлено в таблице 266.

Таблица 266 – Регистр IBRD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

24.12.6 **Регистр дробной части делителя скорости передачи данных FBRD**

Назначение бит регистра представлено в таблице ниже.

Таблица 267 – Регистр BFRD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5…0	BAUDDIV_FRAC	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

Коэффициент деления вычисляется по формуле

$$\text{BAUDDIV} = \frac{F_{UART_CLK}}{16 \cdot Baud_rate}, \quad (20)$$

где FUART_CLK – тактовая частота контроллера UART;

Baud_rate – требуемая скорость передачи данных (в бит/с).

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUDDIV_INT и BAUDDIV_FRAC, соответственно.

Примечание – Изменение содержимого регистров IBRD и FBRD вступают в силу только после завершения передачи и приема текущего символа данных.

Минимальный допустимый коэффициент деления – 1, максимальный – 65535 ($2^{16} – 1$). Таким образом, значение IBRD, равное 0 является недопустимым, при этом значение регистра FBRD игнорируется.

Аналогично, при IBRD равном 65535 (0xFFFF) значение FBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

Пример. Вычисление коэффициента деления

Пусть требуемая скорость передачи данных составляет 230400 бит/с, частота тактового сигнала F_{UART_CLK} = 4 МГц. Тогда

$$\text{Коэффициент деления} = \frac{4 \cdot 10^6}{16 \cdot 230400} = 1{,}085\,.$$

Таким образом, BRDI = 1, BRDF = 0,085.

Следовательно, значение, записываемое в регистр BFRD, равно

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

Реальное значение коэффициента деления = 1+5/64 = 1,078.

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

Ошибка установки скорости = (231911-230400)/230400 • 100% = 0,656 %.

Максимальная ошибка установки скорости передачи данных с использованием шестиразрядного регистра BFRD = 1/64 • 100 % = 1,56 %. Такая ошибка возникает в случае m = 1, при этом разница накапливается в течение 64 тактовых интервалов.

Таблица 268 содержит значения коэффициента деления для типичных скоростей передачи данных при частоте F_{UART_CLK} = 7,3728 МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр FBRD должен быть записан ноль.

Таблица 268 – Коэффициенты деления при частоте F_{UART_CLK} = 7,3728 МГц

Коэффициент деления	Скорость передачи данных
0x0001	460800
0x0002	230400
0x0004	115200
0x0006	76800
0x0008	57600
0x000C	38400
0x0018	19200
0x0020	14400
0x0030	9600
0x00C0	2400
0x0180	1200
0x105D	110

В таблице 269 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте F_{UART_CLK} = 4 МГц.

Таблица 269 – Коэффициенты деления при частоте F_{UART_CLK} = 4 МГц

Целая часть	Дробная часть	Требуемая скорость	Реальная скорость	Ошибка, %
0x001	0x05	230400	231911	0.656
0x002	0x0B	115200	115101	0.086
0x003	0x10	76800	76923	0.160
0x006	0x21	38400	38369	0.081
0x011	0x17	14400	14401	0.007
0x01A	0x03	9600	9598	0.021
0x068	0x0B	2400	2400	~0

24.12.7 **Регистр управления линией LCR_H**

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

Назначение битов регистра описано в таблице 270.

Таблица 270 – Регистр LCR_H

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…9		Зарезервировано. Не модифицируйте. При чтении выдаются нули
8	WLEN[2]	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0 – длина слова определяется WLEN[1:0]; 1 – 9 бит

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	SPS	Передача бита четности с фиксированным значением: 0 – запрещена; 1 – на месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. (При EPS=0 на месте бита четности передается 1, при EPS=1 – передается 0). Значение бита SPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено (таблица 271)
6…5	WLEN[1:0]	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0b11 – 8 бит, 0b10 – 7 бит, 0b01 – 6 бит, 0b00 – 5 бит
4	FEN	Разрешение работы буфера FIFO приемника и передатчика: 0 – запрещено; 1 – разрешено
3	STP2	Режим передачи двух стоповых битов: 0 – один стоповый бит; 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре
2	EPS	Четность/нечетность: 0 – бит четности дополняет количество единиц в информационной части кадра до нечетного; 1 – до четного числа. Значение бита EPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено (Таблица 271)
1	PEN	Разрешение проверки четности: 0 – кадр не содержит бита четности; 1 – бит четности передается в кадре и проверяется при приеме данных (таблица 271)
0	BRK	Разрыв линии. Если этот бит установлен в 1, то по завершении передачи текущего символа на выходе линии UART_TXD устанавливается низкий уровень сигнала. Для правильного выполнения этой операции программное обеспечение должно обеспечить передачу сигнала разрыва в течение, как минимум, времени передачи двух информационных кадров. В нормальном режиме функционирования бит должен быть установлен в 0

Содержимое регистров LCR_H, IBRD и FBRD совместно образует общий 31-разрядный регистр LCR, который обновляется по стробу, формируемому при записи в LCR_H. Таким образом, для того, чтобы изменение параметров коэффициента деления частоты обмена данными вступило в силу, после их изменения значения регистров IBRD и/или FBRD необходимо осуществить запись данных в регистр LCR_H.

Таблица 271 содержит данные об истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией LCR_H.

Таблица 271 – Управление режимом контроля четности

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

Примечания

– 1 Регистры LCR_H, IBRD и FBRD не должны изменяться:
  – при разрешенной работе приемопередатчика;
  – во время завершения приема или передачи данных в процессе остановки (перевода в запрещенное состояние) приемопередатчика.
– 2 Целостность данных в буферах FIFO не гарантируется в следующих случаях:
  – после установки бита разрыва линии BRK;
  – если программное обеспечение произвело остановку приемопередатчика при наличии данных в буферах FIFO, после его повторного перевода в разрешенное состояние.

24.12.8 **Регистр управления CR**

После сброса все биты регистра управления, за исключением битов 9 и 8 устанавливаются в состояние логического «0». Биты 9 и 8 устанавливаются в состояние логической «1».

Назначение битов регистра управления показано в таблице 272.

Таблица 272 – Регистр управления CR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15	CTSEn	Разрешение управления потоком данных по CTS: 1 – разрешено, данные передаются в линию только при активном значении сигнала nUARTCTS; 0 – запрещено
14	RTSEn	Разрешение управления потоком данных по RTS: 1 – разрешено, запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере FIFO приемника; 0 – запрещено
13	Out2	Инверсия сигнала на линии состояния модема nUARTOut2. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «сигнал вызова» (RI). 1 – сигнал разрешен; 0 – сигнал запрещен
12	Out1	Инверсия сигнала на линии состояния модема nUARTOut1. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «обнаружен информационный сигнал» (DCD). 1 – сигнал разрешен; 0 – сигнал запрещен

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
11	RTS	Инверсия сигнала на линии состояния модема nUARTRTS: 1 – сигнал разрешен; 0 – сигнал запрещен
10	DTR	Инверсия сигнала на линии состояния модема nUARTDTR: 1 – сигнал разрешен; 0 – сигнал запрещен
9	RXE	Разрешение приема. 1 – работа приемника разрешена. Прием данных осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчика в запрещенное состояние в ходе приема данных, он завершает прием текущего символа перед остановкой. 0 – работа приемника запрещена
8	TXE	Разрешение передачи. 1 – работа передатчика разрешена. Передача осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчик в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой. 0 – работа передатчика запрещена
7	LBE	Режим тестирования по шлейфу: 1 – шлейф разрешен; 0 – запрещен. В режиме разрешенного шлейфа: Если установлены бит SIREN=1 и бит регистра управления тестированием TCR SIRTEST=1, то сигнал с выхода кодека nSIROUT инвертируется и подается на вход кодека SIRIN. Бит SIRTEST устанавливается в 1 для того, чтобы вывести устройство из полудуплексного режима, характерного для интерфейса SIR. После окончания тестирования по шлейфу бит SIRTEST должен быть установлен в 0. Если бит SIRTEST=0, то выходная линия передатчика UART_TXDx коммутируется на вход приемника UART_RXDx. Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии. После сброса бит устанавливается в 0
6…3		Резерв. Не модифицируйте. При чтении выдаются нули
2	SIRLP	Выбор режима ИК обмена с пониженным энергопотреблением: 1 – длительность импульсов данных равна трем тактам сигнала IrLPBaud16 вне зависимости от выбранной скорости передачи данных. Выбор этого режима снижает энергопотребление, однако может привести к уменьшению дальности связи; 0 – длительность импульсов данных равна 3/16 длительности передачи бита

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1	SIREN	Разрешение работы кодека ИК передачи данных IrDA SIR: 1 – разрешена работа кодека ИК. Данные передаются на выход nSIROUT и принимаются с входа SIRIN. Линия UART_TXDx находится в высоком состоянии. Данные на входе UART_RXDx и линиях состояния модема не обрабатываются. В случае если UARTEN=0 значение бита не играет роли. 0 – запрещен. Сигнал nSIROUT находится в низком состоянии, данные на входе SIRIN не обрабатываются
0	UARTEN	Разрешение работы приемопередатчика: 0 – работа запрещена. Перед остановкой завершается прием и/или передача обрабатываемого в текущий момент символа; 1 – работа разрешена. Производится обмен данными либо по линиям асинхронного обмена, либо по линиям ИК обмена SIR, в зависимости от состояния бита SIREN

Примечание – Для того, чтобы разрешить передачу данных, необходимо установить в 1 биты TXE и UARTEN. Аналогично, для разрешения приема данных необходимо установить в 1 биты RXE и UARTEN.

Для программирования регистров управления рекомендуется следующая последовательность действий:

1. Остановите работу приемопередатчика;
2. Дождаться окончания приема и/или передачи текущего символа данных;
3. Сбросьте буфер передатчика путем установки бита FEN регистра LCR_H в 0;
4. Изменить настройки регистра CR;
5. Возобновите работу приемопередатчика.

24.12.9 **Регистр порога прерывания по заполнению буфера FIFO IFLS**

Данный регистр используется для установки порогового значения заполнения буферов передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода величины заполнения буфера через заданное значение.

После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра и значения его битов приведены в таблице 273.

Таблица 273 – Регистр IFLS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…6		Резерв. Не модифицируйте. При чтении выдаются нули
5…3	RXIFLSEL	Порог прерывания по заполнению буфера приемника: b000 = Буфер заполнен на 1/8; b001 = Буфер заполнен на 1/4; b010 = Буфер заполнен на 1/2; b011 = Буфер заполнен на 3/4; b100 = Буфер заполнен на 7/8; b101-b111 = резерв

2…0	TXIFLSEL	Порог прерывания по заполнению буфера передатчика: b000 = Буфер заполнен на 1/8; b001 = Буфер заполнен на 1/4; b010 = Буфер заполнен на 1/2; b011 = Буфер заполнен на 3/4; b100 = Буфер заполнен на 7/8; b101-b111 = резерв. Также стоит помнить, что в случае, когда сдвиговый регистр передатчика пуст, то слово, записанное в FIFO, будет сразу же переписано в сдвиговый регистр. Следовательно, для генерирования события прерывания от передатчика блока UART необходимо произвести запись в FIFO такого количества слов, которое превысит установленный порог хотя бы на одно слово с учетом описанного случая

24.12.10 **Регистр установки сброса маски прерывания IMSC**

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение битов регистра IMSC приведено в таблице 274.

Таблица 274 – Регистр IMSC

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…14		Резерв. Не модифицируйте. При чтении выдаются нули
13	TNBSYIM	Маска прерывания при отсутствии данных в сдвиговом регистре передатчика UARTTNBSYINTR: 1 – установлена; 0 – сброшена
12	TFEIM	Маска прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR: 1 – установлена; 0 – сброшена
11	RNEIM	Маска прерывания по наличию данных в FIFO приемника UARTRNEINTR: 1 – установлена; 0 – сброшена
10	OEIM	Маска прерывания по переполнению буфера UARTOEINTR: 1 – установлена; 0 – сброшена
9	BEIM	Маска прерывания по разрыву линии UARTBEINTR: 1 – установлена; 0 – сброшена
8	PEIM	Маска прерывания по ошибке контроля четности UARTPEINTR: 1 – установлена; 0 – сброшена

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	FEIM	Маска прерывания по ошибке в структуре кадра UARTFEINTR: 1 – установлена; 0 – сброшена
6	RTIM	Маска прерывания по таймауту приема данных UARTRTINTR: 1 – установлена; 0 – сброшена
5	TXIM	Маска прерывания от передатчика UARTTXINTR: 1 – установлена; 0 – сброшена
4	RXIM	Маска прерывания от приемника UARTRXINTR: 1 – установлена; 0 – сброшена
3	DSRMIM	Маска прерывания UARTDSRINTR по изменению состояния линии nUARTDSR: 1 – установлена; 0 – сброшена
2	DCDMIM	Маска прерывания UARTDCDINTR по изменению состояния линии nUARTDCD: 1 – установлена; 0 – сброшена
1	CTSMIM	Маска прерывания UARTCTSINTR по изменению состояния линии nUARTCTS: 1 – установлена; 0 – сброшена
0	RIMIM	Маска прерывания UARTRIINTR по изменению состояния линии nUARTRI: 1 – установлена; 0 – сброшена

24.12.11 **Регистр состояния прерываний RIS**

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Предупреждение. После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено.

Назначение битов регистра RIS приведены в таблице 275.

Таблица 275 – Регистр RIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…14		Резерв. Не модифицируйте. При чтении выдаются нули
13	TNBSYRIS	Состояние прерывания при отсутствии данных в сдвиговом регистре передатчика UARTTNBSYINTR
12	TFERIS	Состояние прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
11	RNERIS	Состояние прерывания по наличию данных в FIFO приемника UARTRNEINTR
10	OERIS	Состояние прерывания по переполнению буфера UARTOEINTR: 1 – буфер приемника переполнен; 0 – буфер приемника не переполнен
9	BERIS	Состояние прерывания по разрыву линии UARTBEINTR: 1 – произошел разрыв линии приема; 0 – разрыва линии приема не происходило
8	PERIS	Состояние прерывания по ошибке контроля четности UARTPEINTR: 1 – возникла ошибка контроля четности; 0 – ошибки контроля четности не возникало
7	FERIS	Состояние прерывания по ошибке в структуре кадра UARTFEINTR: 1 – возникла ошибка в структуре кадра; 0 – ошибки в структуре кадра не возникало
6	RTRIS	Состояние прерывания по таймауту приема данных UARTRTINTR*: 1 – вышло время таймаута приема данных; 0 – время таймаута приема данных не вышло
5	TXRIS	Состояние прерывания от передатчика UARTTXINTR: 1 – возникло прерывание от передатчика; 0 – прерывания от передатчика нет
4	RXRIS	Состояние прерывания от приемника UARTRXINTR: 1 – возникло прерывание от приемника; 0 – прерывание от приемника не возникало
3	DSRRMIS	Состояние прерывания UARTDSRINTR по изменению линии nUARTDSR: 1 – возникло прерывание; 0 – прерывание не возникало
2	DCDRMIS	Состояние прерывания UARTDCDINTR по изменению линии nUARTDCD: 1 – возникло прерывание; 0 – прерывание не возникало
1	CTSRMIS	Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS: 1 – возникло прерывание; 0 – прерывание не возникало
0	RIRMIS	Состояние прерывания UARTRIINTR по изменению линии nUARTRI: 1 – возникло прерывание; 0 – прерывание не возникало

* Сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Поэтому чтение состояния прерывания по таймауту из регистров MIS и RIS даст одинаковый результат

24.12.12 **Регистр маскированного состояния прерываний MIS**

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено.

Назначение битов регистра MIS приведены в таблице 276.

Таблица 276 – Регистр MIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…14		Резерв. Не модифицируйте. При чтении выдаются нули
13	TNBSYMIS	Маскированное состояние прерывания при отсутствии данных в сдвиговом регистре передатчика UARTTNBSYINTR
12	TFEMIS	Маскированное состояние прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR
11	RNEMIS	Маскированное состояние прерывания по наличию данных в FIFO приемника UARTRNEINTR
10	OEMIS	Маскированное состояние прерывания по переполнению буфера UARTOEINTR: 1 – буфер приемника переполнен; 0 – буфер приемника не переполнен
9	BEMIS	Маскированное состояние прерывания по разрыву линии UARTBEINTR: 1 – произошел разрыв линии приема; 0 – разрыва линии приема не происходило
8	PEMIS	Маскированное состояние прерывания по ошибке контроля четности UARTPEINTR: 1 – возникла ошибка контроля четности; 0 – ошибки контроля четности не возникало
7	FEMIS	Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR: 1 – возникла ошибка в структуре кадра; 0 – ошибки в структуре кадра не возникало
6	RTMIS	Маскированное состояние прерывания по таймауту приема данных UARTRTINTR: 1 – вышло время таймаута приема данных; 0 – время таймаута приема данных не вышло
5	TXMIS	Маскированное состояние прерывания от передатчика UARTTXINTR: 1 – возникло прерывание от передатчика; 0 – прерывания от передатчика нет
4	RXMIS	Маскированное состояние прерывания от приемника UARTRXINTR: 1 – возникло прерывание от приемника; 0 – прерывание от приемника не возникало

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	DSRMMIS	Маскированное состояние прерывания UARTDSRINTR по изменению линии nUARTDSR: 1 – возникло прерывание; 0 – прерывание не возникало
2	DCDMMIS	Маскированное состояние прерывания UARTDCDINTR по изменению линии nUARTDCD: 1 – возникло прерывание; 0 – прерывание не возникало
1	CTSMMIS	Маскированное состояние прерывания UARTCTSINTR по изменению линии nUARTCTS: 1 – возникло прерывание; 0 – прерывание не возникало
0	RIMMIS	Маскированное состояние прерывания UARTRIINTR по изменению линии nUARTRI. 1 – возникло прерывание; 0 – прерывание не возникало

24.12.13 **Регистр сброса прерываний ICR**

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись 0 в любой из разрядов регистра игнорируется.

Назначение битов регистра ICR приведено в таблице 277.

Таблица 277 – Регистр ICR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…11		Резерв. Не модифицируйте. При чтении выдаются нули
10	OEIC	Сброс прерывания по переполнению буфера UARTOEINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
9	BEIC	Сброс прерывания по разрыву линии UARTBEINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
8	PEIC	Сброс прерывания по ошибке контроля четности UARTPEINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
7	FEIC	Сброс прерывания по ошибке в структуре кадра UARTFEINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
6	RTIC	Сброс прерывания по таймауту приема данных UARTRTINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
5	TXIC	Сброс прерывания от передатчика UARTTXINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
4	RXIC	Сброс прерывания от приемника UARTRXINTR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
3	DSRMIC	Сброс прерывания UARTDSRINTR по изменению линии nUARTDSR: 1 – сброс прерывания; 0 – не влияет на состояние регистра
2	DCDMIC	Сброс прерывания UARTDCDINTR по изменению линии nUARTDCD: 1 – сброс прерывания; 0 – не влияет на состояние регистра
1	CTSMIC	Сброс прерывания UARTCTSINTR по изменению линии nUARTCTS: 1 – сброс прерывания; 0 – не влияет на состояние регистра.
0	RIMIC	Сброс прерывания UARTRIINTR по изменению линии nUARTRI: 1 – сброс прерывания; 0 – не влияет на состояние регистра

24.12.14 **Регистр управления прямым доступом к памяти DMACR**

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

Назначение битов регистра DMACR приведено в таблице 278.

Таблица 278 – Регистр DMACR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…3		Резерв. Не модифицируйте. При чтении выдаются нули
2	DMAONERR	Блокирование DMA запросов при приеме при возникновении ошибок на линии: 1 – в случае возникновения прерывания по обнаружению ошибки блокируются запросы DMA от приемника UARTRXDMASREQ и UARTRXDMABREQ; 0 – DMA запросы не блокируются
1	TXDMAE	Использование DMA при передаче: 1 – разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика; 0 – запрещено формирование DMA запросов
0	RXDMAE	Использование DMA при приеме: 1 – разрешено формирование запросов DMA для обслуживания буфера FIFO приемника; 0 – запрещено формирование DMA запросов

24.12.15 **Регистр управления тестированием TCR**

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

Назначение битов регистра TCR приведено в таблице 279.

Таблица 279 – Регистр TCR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…13		Резерв. При чтении значение непредсказуемо
12…3	-	Зарезервировано
2	SIRTEST	Разрешение приёма данных в кольцевом режиме с выхода IrDA передатчика: 1 – разрешено; 0 – запрещено. Используется совместно с установкой бита LBE в регистре CR
1	TESTFIFO	Разрешение чтения данных из FIFO передатчика и запись в FIFO приёмника: 1 – разрешено; 0 – запрещено
0	ITEN	Перевод контроллера UART в тестовый режим: 1 – тестовый режим разрешён; 0 – тестовый режим запрещён

25 Контроллер прямого доступа в память DMA

25.1 Основные свойства контроллера DMA

Основные свойства и отличительные особенности:

– 32 канала DMA;

– каждый канал DMA имеет свои сигналы управления передачей данных;

– каждый канал DMA имеет программируемый уровень приоритета;

– каждый уровень приоритета обрабатывается, исходя из уровня приоритета, определяемого номером канала DMA;

– поддержка различного типа передачи данных:

  – память – память;

  – память – периферия;

  – периферия – память;

– поддержка различных типов DMA циклов;

– поддержка передачи данных различной разрядности;

– каждому каналу DMA доступна первичная и альтернативная структура управляющих данных канала;

– все управляющие данные канала хранятся в системной памяти;

– разрядность данных приемника равна разрядности данных передатчика;

– количество передач в одном цикле DMA может программироваться от 1 до 1024;

– инкремент адреса передачи может быть больше, чем разрядность данных.

25.2 Термины и определения

Таблица 280 – Термины и определения

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

Передача DMA	Акция пересылки одного байта, полуслова или слова
N	Общее количество передач DMA, которые контроллер выполняет для канала
Пинг-понг	Режим работы для выбранного канала, при котором контроллер получает начальный запрос и затем выполняет цикл DMA, используя первичную или альтернативную структуру данных. После завершения этого цикла DMA контроллер начинает выполнять новый цикл DMA, используя другую (первичную или альтернативную) структуру данных. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя главному процессору перенастраивать неактивную структуру данных. Контроллер продолжает переключаться от первичной к альтернативной структуре данных и обратно до тех пор, пока он не прочитает «неправильную» структуру данных, или пока он не завершит цикл без переключения к другой структуре
Первичная	Первичная структура управляющих данных канала. Контроллер использует эту структуру данных, если соответствующий разряд в регистре chnl_pri_alt_set установлен в 0
R	Степень числа 2, устанавливающее число передач DMA, которые могут произойти перед сменой арбитража. Количество передач DMA программируется в диапазоне от 1 до 1024 двоичными шагами от 2 в степени 0 до 2 в степени 10
Исполнение с изменением конфигурации	Режим работы для выбранного канала, при котором контроллер получает запрос от периферии и выполняет 4 DMA передачи, используя первичную структуру управляющих данных, которые настраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. После того, как цикл закончится и, если периферия устанавливает новый запрос на обслуживание, контроллер выполняет снова 4 DMA передачи, используя первичную структуру управляющих данных, которые опять перенастраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. Контроллер будет продолжать работать вышеописанным способом до тех пор, пока не прочитает неправильную структуру данных или процессор не установит альтернативную структуру данных для обычного цикла. Контроллер устанавливает флаг dma_done, если окончание подобного режима работы происходит после выполнения обычного цикла

25.3 **Функциональное описание**

На рисунке 86 показана упрощенная структурная схема контроллера.

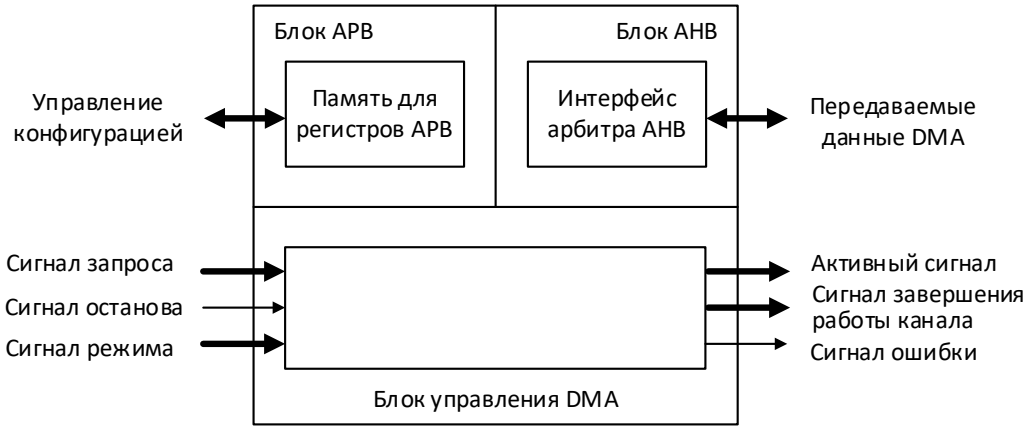


Рисунок 86 – Структурная схема контроллера

Контроллер состоит из следующих основных функциональных блоков:

– блок, подключенный к шине APB;
– блок, подключенный к шине AHB;
– блок управления DMA.

25.3.1 **Распределение каналов DMA**

Таблица 281 – Распределение каналов DMA

Номер запроса	dma_req	dma_sreq	Описание
0	UART1TXDMABREQ	UART1TXDMASREQ	Запрос от передатчика UART1
1	UART1RXDMABREQ	UART1RXDMASREQ	Запрос от приёмника UART1
2	UART2TXDMABREQ	UART2TXDMASREQ	Запрос от передатчика UART2
3	UART2RXDMABREQ	UART2RXDMASREQ	Запрос от приёмника UART2
4	SSP1TXDMABREQ	SSP1TXDMASREQ	Запрос от передатчика SSP1
5	SSP1RXDMABREQ	SSP1RXDMASREQ	Запрос от приёмника SSP1
6	CRCDMABREQ	CRCDMASREQ	Запрос от аппаратного блока вычисления CRC
7	UART3TXDMABREQ	UART3TXDMASREQ	Запрос от передатчика UART3
8	UART3RXDMABREQ	UART3RXDMASREQ	Запрос от приёмника UART3
9	TMR3DMAREQ	TMR3DMAREQ	Запрос от таймера общего назначения TIMER3
10	TMR1DMAREQ	TMR1DMAREQ	Запрос от таймера общего назначения TIMER1
11	TMR2DMAREQ	TMR2DMAREQ	Запрос от таймера общего назначения TIMER2
12	ADCUIDMABREQ1 (I0)	ADCUIDMASREQ1 (I0)	Запросы от блока ΔΣ АЦП
13	–	–	
14	ADCUIDMABREQ3 (I1)	ADCUIDMASREQ3 (I1)	
15	–	–	
16	ADCUIDMABREQ5 (I2)	ADCUIDMASREQ5 (I2)	
17	–	–	
18	ADCUIDMABREQ7 (I3)	ADCUIDMASREQ7 (I3)	

Номер запроса	dma_req	dma_sreq	Описание
19	SSP2TXDMABREQ	SSP2TXDMASREQ	Запрос от передатчика SSP2
20	SSP2RXDMABREQ	SSP2RXDMASREQ	Запрос от приёмника SSP2
21	SSP3TXDMABREQ	SSP3TXDMASREQ	Запрос от передатчика SSP3
22	SSP3RXDMABREQ	SSP3RXDMASREQ	Запрос от приёмника SSP3
23	TMR4DMAREQ	TMR4DMAREQ	Запрос от таймера общего назначения TIMER4
24	UART4TXDMABREQ	UART4TXDMASREQ	Запрос от передатчика UART4
25	UART4RXDMABREQ	UART4RXDMASREQ	Запрос от приёмника UART4
26	-	-	-
27	-	-	-
28	-	-	-
29	-	-	-
30	-	ADCDMASREQ	Запрос от АЦП последовательных приближений
31	-	-	-

25.3.2 **Блок, подключенный к шине APB**

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

25.3.3 **Блок, подключенный к шине AHB**

Контроллер содержит один блок типа «ведущий» шины DMA Bus, который позволяет, используя 32-разрядную шину, передавать данные от источника к приемнику. Источник и приемник являются ведомыми шины AHB.

25.3.4 **Управляющий блок DMA**

Этот блок содержит схему управления, позволяющую реализовать следующие функции:

- осуществление арбитража поступающих запросов;
- индикацию активного канала;
- индикацию завершения обмена по каналу;
- индикацию состояния ошибки обмена по шине DMA Bus;
- разрешение медленным устройствам приостанавливать исполнение цикла DMA;
- ожидание запроса на очистку до завершения цикла DMA;
- осуществление одиночных или множественных передач DMA для каждого запроса;
- осуществление следующих типов DMA передач:
  - память – память;
  - память – периферия;
  - периферия – память.

25.3.5 **Типы передач**

Контроллер интерфейса не поддерживает пакетные передачи. Контроллер выполняет одиночные передачи. Отсутствие возможности осуществлять пакетные передачи оказывает минимальное влияние на производительность системы, так как пакетные передачи более эффективны в одноуровневых системах с шиной AHB, где блоки должны «захватывать» шину или обращаться к внешней памяти. В тоже время контроллер DMA предназначен для использования в многоуровневых системах с шиной AHB, включающих встроенную память.

25.3.6 **Разрядность передаваемых данных**

Контроллер интерфейса предоставляет возможность осуществлять передачу 8, 16 и 32 разрядных данных. Значения комбинаций шины HSIZE приведены в таблице 282.

Таблица 282 – Комбинации шины HSIZE

HSIZE[2]*	HSIZE[1]	HSIZE[0]	Разрядность данных (бит)
0	0	0	8
0	0	1	16
0	1	0	32
0	1	1	**
* Сигнал постоянно удерживается в состоянии 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.

25.3.7 **Управление защитой данных**

Контроллер позволяет устанавливать режимы защиты данных протокола AHB-Lite, определяемые шиной HPROT[3:1]. Возможен выбор следующих режимов защиты:

- кэширование;
- буферизация;
- привилегированный.

В таблице 283 приведены значения комбинаций шины HPROT.

Таблица 283 – Режимы защиты данных

HPROT[3] кэширование	HPROT[2] буферизация	HPROT[1] привилегиро-ванный	HPROT[0] данные/команда	Описание
-	-	-	1*	Доступ к данным
-	-	0	-	Пользовательский доступ
-	-	1	-	Привилегированный доступ

-	0	-	-	Без буферизации
-	1	-	-	Буферизованный
0	-	-	-	Без кэширования
1	-	-	-	Кэшированный

* Контроллер удерживает HPROT[0] в состоянии логической «1», чтобы обозначить доступ к данным.

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

Для каждого канала DMA также возможен выбор режима защиты данных. Более подробно см. в подразделе «Управление DMA»

25.3.8 **Инкремент адреса**

Контроллер позволяет управлять инкрементом адреса при чтении данных из источника и при записи данных в приемник. Инкремент адреса зависит от разрядности передаваемых данных. В таблице 284 приведены возможные комбинации.

Таблица 284 – 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 приведено в подразделе 25.5 «Структура управляющих данных канала». Этот раздел описывает разряды управления величиной инкремента адреса в управляющих данных канала.

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

25.4 **Управление DMA**

25.4.1 **Правила обмена данными**

Контроллер использует правила обмена данными (см. таблицу 285), при соблюдении следующих условий:

– канал DMA включен, что выполняется установкой в состояние логической единицы разрядов управления chnl_enable_set[C] и master_enable;

– флаги запроса dma_req[C] и dma_sreq[C] не замаскированы, что выполняется установкой в состояние логического нуля разряда управления chnl_req_mask_set [C];

Таблица 285 – Правила, при которых передача данных по каналам разрешена, и запросы не маскируются

Правило	Описание
1	Если dma_active[C] установлен в 0, то установка в 1 dma_req[C] или dma_sreq[C] на один или более тактов сигнала HCLK, следующих или не следующих друг за другом, инициирует передачу по каналу номер C
2	Контроллер осуществляет установку в 1 только одного разряда dma_active[C]
3	Контроллер устанавливает в 1 dma_active[C] в момент начала передачи по каналу C
4	Для типов циклов DMA, отличных от периферийного «Исполнение с изменением конфигурации», dma_active[C] остается в состоянии 1 до тех пор, пока контроллер не окончит передачи с номерами меньше, чем значение 2^R или чем число передач, указанное в регистре n_minus_1. В периферийном режиме «Исполнение с изменением конфигурации», dma_active[C] остается в состоянии 1 в течение каждой пары DMA передач, с использованием первичной и альтернативной структур управляющих данных. Таким образом, контроллер выполняет 2^R передач, используя первичную структуру управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных. По окончании последней передачи dma_active[C] сбрасывается в 0
5	Контроллер устанавливает dma_active[C] в 0 как минимум на один такт сигнала HCLK перед тем, как снова установит dma_active[C] или dma_active[] в 1
6	Для каналов, по которым разрешена передача, контроллер осуществляет установку в 1 только одного dma_done[]
7	Если dma_req[C] устанавливается в состояние 1 в момент, когда dma_active[C] или dma_stall также в состоянии 1, то это означает, что контроллер обнаружил запрос
8	Если разряды cycle_ctrl для канала установлены в состояние 3’b100, 3’b101, 3’b110, 3’b111, то dma_done[C] никогда не будет установлен в 1
9	Если все передачи по каналу завершены, и разряды cycle_ctrl позволяют удержание dma_done[C], то по срезу сигнала dma_active[] произойдут события: – если dma_stall в состоянии 0, контроллер устанавливает dma_done[] в состояние 1 продолжительностью один такт HCLK – если dma_stall в состоянии 1, работа контроллера приостановлена. После того, как dma_stall будет установлен в 0, контроллер устанавливает dma_done[] в состояние 1 продолжительностью один такт HCLK
10	Состояние dma_waitonreq[C] можно изменять только при выключенном канале
11	Если dma_waitonreq[C] в состоянии 1, то сигнал dma_active[C] не перейдет в состояние 0 до тех пор, пока: контроллер завершит 2^R передач (или число передач, указанное в регистре n_minus_1); dma_req[C] будет установлен в 0; dma_sreq[C] будет установлен в 0
12	Если за один такт сигнала HCLK перед установкой dma_active[C] в 0 dma_stall устанавливается в 1, то контроллер установит dma_active[C] в 0 на следующем такте сигнала HCLK; передача по каналу C не завершится, пока не будет сброшен в 0 dma_stall
13	Контроллер игнорирует dma_sreq[C], если dma_waitonreq[C] в состоянии 0
14	Контроллер игнорирует dma_sreq[C], если chnl_useburst_set[C] в состоянии 1*)

Правило	Описание
15	Для циклов DMA, отличных по типу от периферийного режима «Исполнение с изменением конфигурации», по окончании 2^R передач контроллер устанавливает значение chnl_useburst_set[C] в состояние 0, если количество оставшихся передач меньше, чем 2^R. В периферийном режиме «Исполнение с изменением конфигурации» контроллер устанавливает значение chnl_useburst_set[C] в состояние 0 только, если количество оставшихся передач с использованием альтернативной структуры управляющих данных меньше, чем 2^R
16	Для типов циклов DMA, отличных от периферийного режима «Исполнение с изменением конфигурации», если за один такт HCLK до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, то контроллер выполняет одну DMA передачу. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт HCLK до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных. Затем без осуществления арбитража выполняет одну передачу, используя альтернативную структуру управляющих данных
17	Для типов циклов DMA, отличных от периферийного режима «Исполнение с изменением конфигурации», если за один такт HCLK до установки dma_active[C] в 1, а dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c], и контроллер выполняет 2^R (или число передач, указанное в регистре n_minus_1) DMA передач. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт HCLK до установки dma_active[C] в 1 dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c], и контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных
18	Когда chnl_req_mask_set[C] установлен в 1, контроллер игнорирует запросы по dma_sreq[C] и dma_req[C]
*) Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре n_minus_1 меньше, чем значение 2^R, то контроллер не очистит разряды chnl_useburst_set и поэтому запросы по dma_sreq[C] будут маскированы. Если периферия не устанавливает dma_req[C] в состояние 1, то контроллер никогда не выполнит необходимых передач	

При отключении канала контролер осуществляет DMA-передачи согласно правилам, приведенным в таблице 286.

Таблица 286 – Правила осуществления DMA передач при «запрещенных» каналах

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

25.4.2 Диаграммы работы контроллера DMA

Данный раздел описывает следующие примеры функционирования контроллера с использованием правил обмена данными (см. таблицу 285):

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

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

– hready находится в состоянии 1;
– AHB «ведомый» всегда дает ответ «OKAY».

25.4.2.1 Импульсный запрос на обработку

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

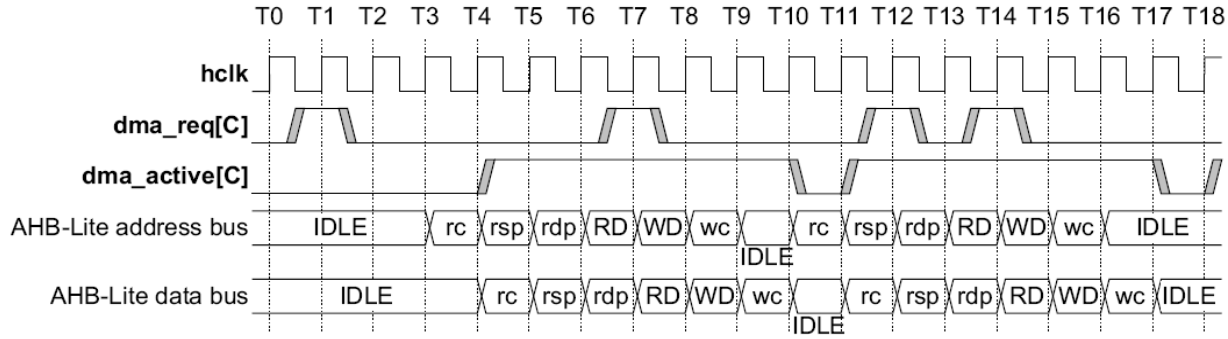


Рисунок 87 – Диаграмма работы при получении импульсного запроса от периферийного блока

Таблица 287 – Пояснения к диаграмме работы при получении импульсного запроса

T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7	При установленном dma_active[C] в 1 и при условии, что chnl_req_mask_set[C] находится в состоянии 0, контроллер обнаруживает импульс запроса на обработки по каналу C (см. правило 7). Контроллер обработает этот запрос в течение следующего арбитража

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

25.4.2.2 **Запрос на обработку по уровню.**

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

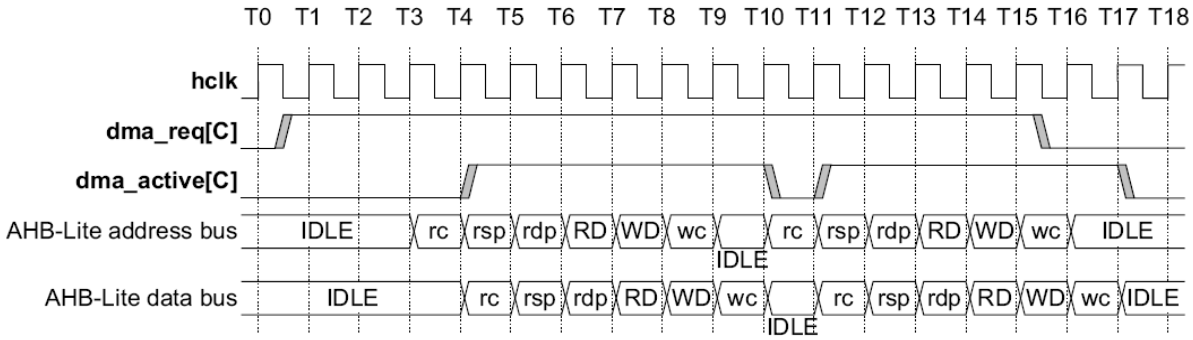


Рисунок 88 – Диаграмма работы при получении запроса на обработку по уровню

Таблица 288 – Пояснения к диаграмме работы при получении запроса на обработку по уровню

T1	Контроллер обнаружил запрос на обработку по каналу C (Таблица 285, правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr.
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных.
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4). Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
T10-T11	Контроллер удерживает dma_active[C] в 0 как минимум на один такт HCLK (см. правило 5)
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C] 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 вовремя снять сигнал запроса, в этом случае он должен установить сигнал dma_stall. Установка сигнала dma_stall предотвращает повторение выполненной передачи.

25.4.2.3 **Флаги завершения**

На рисунке 89 приведена диаграмма функционирования сигнала (флага) dma_done[] при следующих условиях:

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

Выставление сигнала dma_done транслируется в регистр DMA_DONE_STICK блока контроллера тактовой частоты (см. пункт 16.5.9 «DMA_DONE_STICK»).

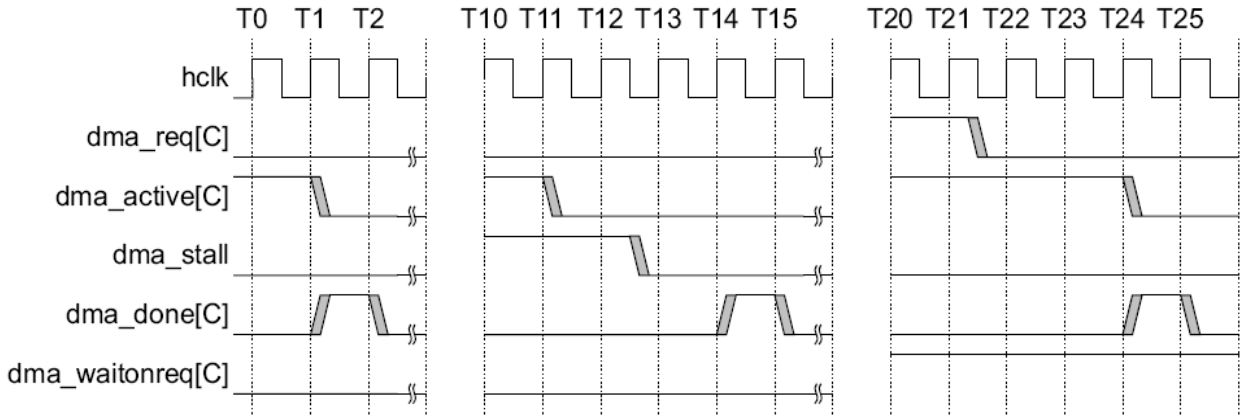


Рисунок 89 – Диаграммы функционирования dma_done

Таблица 289 – Пояснения функционирования dma_done, такты от T0 до T2

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

Таблица 290 – Пояснения функционирования dma_done, такты от T10 до T15

T11*	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T12-T13	Периферийный блок сбрасывает сигнал dma_stall
T14-T15	Контроллер завершает цикл DMA, и если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт HCLK (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6)
* Контроллер не устанавливает сигнал dma_done[C], так как сигнал dma_stall установлен в 1 в предшествующем такте HCLK (см. правила 9 и 12)	

Таблица 291 – Пояснения функционирования dma_done, такты от T20 до T25

T20	Контроллер выполнил передачу DMA, но из-за установленного в 1 dma_waitonreq[C] он должен ожидать сброса в 0 сигнала dma_req[C], перед тем как сбросить dma_active[C] (см. правило 11) и установить dma_done[C] (см. правило 9)
T21-T25	Периферийный блок сбрасывает dma_req[C]
T24	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T24-T25	Контроллер завершает цикл DMA и, если cycle_ctrl[2] установлен в 0, то устанавливает в 1 dma_done[C] на один такт HCLK (см. правила 8 и 9). Для других разрешенных каналов сигнал dma_done[C] останется в состоянии 0 (см. правило 6)

25.4.2.4 **Флаги ожидания запроса на обработку**

Рисунки 90, 91 демонстрируют примеры использования флагов ожидания запроса на обработку при выполнении 2^R передач и одиночных передач:

– диаграмма работы контроллера DMA при использовании периферией dma_waitonreq;

– диаграмма работы контроллера DMA при использовании периферией dma_waitonreq совместно с dma_sreq.

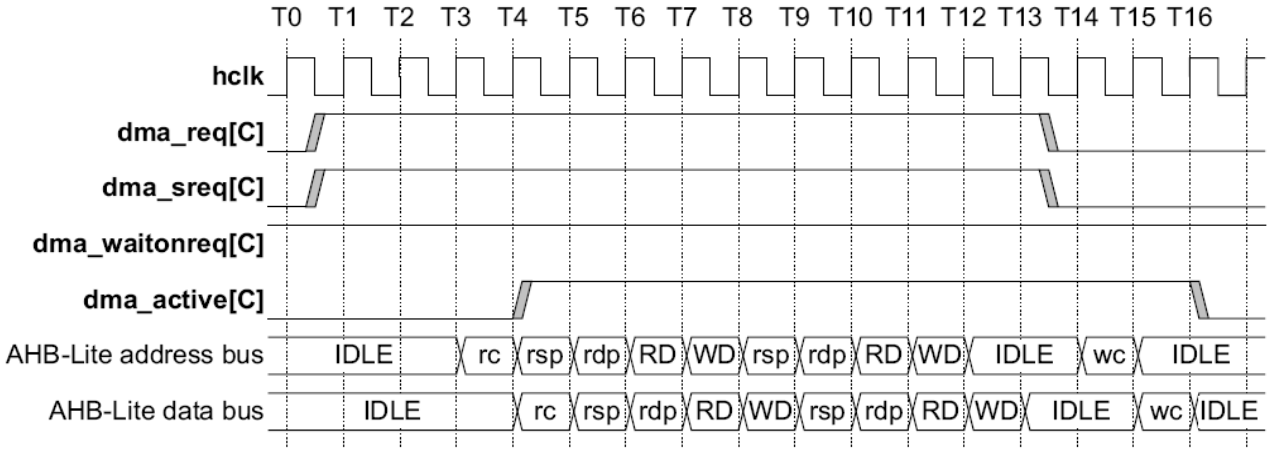


Рисунок 90 – Диаграмма работы контроллера DMA при использовании dma_waitonreq

Таблица 292 – Пояснения работы контроллера DMA при использовании dma_waitonreq

T0-T16	Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10)
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T3-T4	Периферийный блок удерживает dma_req[C] и dma_sreq[C] в 1. Контроллер игнорирует dma_sreq[C] запрос и отвечает на dma_req[C] запрос (см. правила 16 и 17)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr

T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных
T9-T11	Контроллер считывает 2 указателя адреса окончания данных rsp и rdp
T11-T13	Периферийный блок сбрасывает сигналы dma_req[C] и dma_sreq[C]
T15-T16	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T16	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11). Контроллер устанавливает значение по чтению регистра chnl_useburst_set[C] в 0, если количество оставшихся передач менее 2^R (см. правило 15)

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

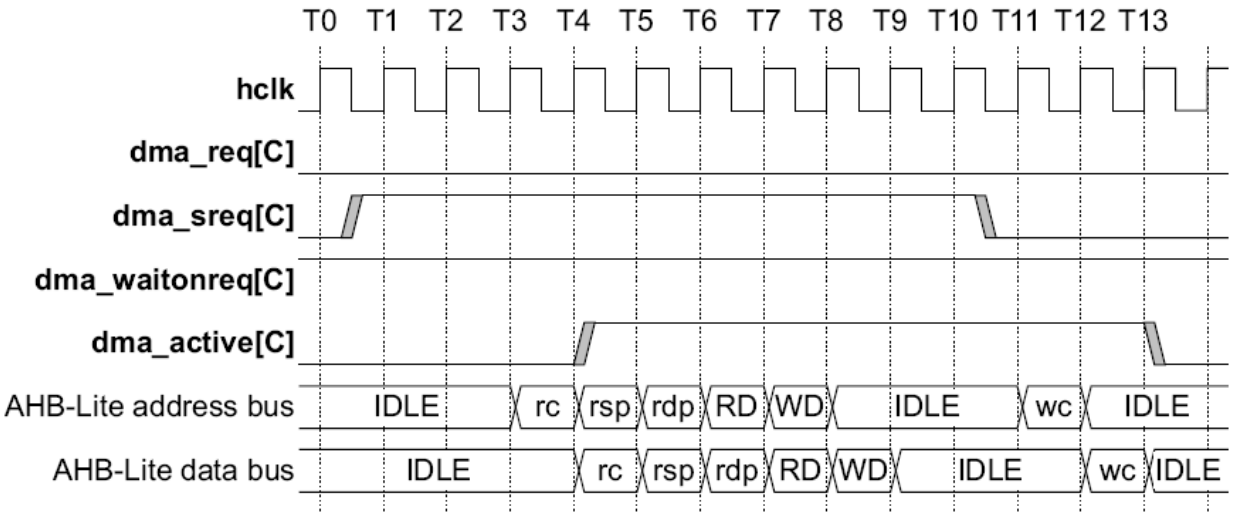


Рисунок 91 – Работа DMA при использовании dma_waitonreq совместно с dma_sreq

Таблица 293 – Пояснения работы DMA при использовании dma_waitonreq совместно с dma_sreq

T0-T13	Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10)
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_useburst_set[C] находится в состоянии 0 (см. правила 13 и 14)
T3-T4	Контроллер отвечает на dma_sreq[C] запрос (см. правила 16).
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных. Это запрос в ответ на dma_sreq[], таким образом, R=0 и, следовательно, контроллер исполнит 1 DMA передачу
T10-T11	Периферийный блок сбрасывает сигнал dma_sreq[C]

T12_T13	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T13	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11)

25.4.3 **Правила арбитража DMA**

Контроллер имеет возможность настройки момента арбитража при передачах DMA. Эта возможность позволяет уменьшить время отклика при обслуживании каналов с высоким приоритетом.

Контроллер имеет четыре разряда, которые определяют количество транзакций по шине AHB до повторения арбитража. Эти разряды задают степень R числа 2; изменение R напрямую устанавливает периодичность арбитража как 2 в степени R. Для примера, если R равно 4, то арбитраж будет проводиться через каждые 16 передач DMA.

Таблица 294 показывает возможную периодичность арбитража.

Таблица 294 – Периодичность арбитража в единицах передач по шине AHB

Значение R	Периодичность арбитража каждые x передач DMA
4’b0000	1
4’b0001	2
4’b0010	4
4’b0011	8
4’b0100	16
4’b0101	32
4’b0110	64
4’b0111	128
4’b1000	256
4’b1001	512
4’b1010 – 4’b1111	1024

Примечание – Необходимо с осторожностью устанавливать большие значения R для низкоприоритетных каналов, так как это может привести к невозможности обслуживать запросы по высокоприоритетным каналам.

При $N > 2^R$ (N – номер передачи) и, если результат деления 2^R на N не целое число, контроллер всегда выполняет последовательность из 2^R передач до тех пор, пока не станет верным $N<2^R$. Контроллер выполняет оставшихся N передач в конце цикла DMA.

Разряды степени R числа 2 находятся в структуре управляющих данных канала. Местонахождение этих разрядов описано в подразделе 25.5 «Структура управляющих данных канала».

25.4.4 **Приоритет**

При проведении арбитража определяется канал для обслуживания в следующем цикле DMA. На выбор следующего канала влияют:

- номер канала;
- уровень приоритета, присвоенного каналу.

Каждому каналу может быть присвоен уровень приоритета по умолчанию (низкий) или высокий уровень приоритета. Присвоение уровня приоритета осуществляется установкой или сбросом разряда chnl_priority_set.

Канал номер 0 имеет высший уровень приоритета, уровень приоритета снижается с увеличением номера канала. В таблице 295 приведены уровни приоритета каналов DMA в порядке его уменьшения.

Таблица 295 – Уровень приоритета каналов DMA

Номер канала	Установка уровня приоритета	Уровень приоритета в порядке его уменьшения
0	Высокий	Наивысший уровень приоритета
1	Высокий	-
2	Высокий	-
-	Высокий	-
-	Высокий	-
-	Высокий	-
30	Высокий	-
31	Высокий	-
0	По умолчанию (низкий)	-
1	По умолчанию (низкий)	-
2	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
-	По умолчанию (низкий)	-
30	По умолчанию (низкий)	-
31	По умолчанию (низкий)	Низший уровень приоритета

После окончания цикла DMA контроллер выбирает следующий для обслуживания канал из всех включенных каналов DMA. На рисунке 92 приведен алгоритм выбора следующего канала для обслуживания.

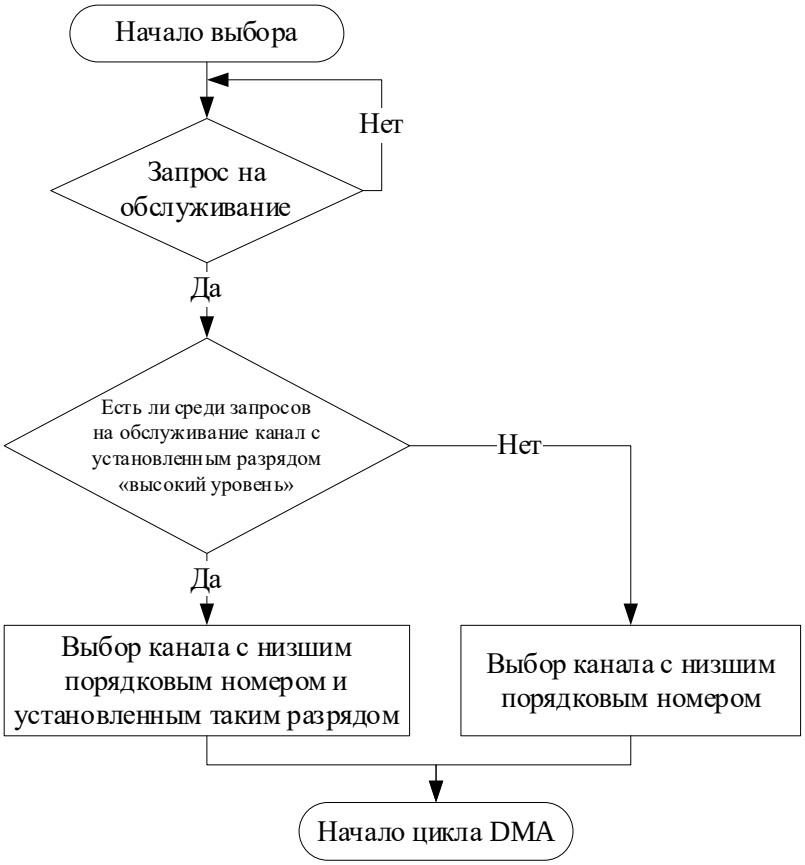


Рисунок 92 – Алгоритм выбора следующего канала для обслуживания

25.4.5 **Типы циклов DMA**

Разряды cycle_ctrl определяют, как контроллер будет выполнять циклы DMA. Описание значений этих разрядов приведено ниже.

Таблица 296 – Типы циклов DMA

cycle_ctrl	Описание
3’b000	Структура управляющих данных канала в запрещенном состоянии
3’b001	Обычный цикл DMA
3’b010	Авто-запрос
3’b011	Режим пинг-понг
3’b100	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
3’b101	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала
3’b110	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
3’b111	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала
Примечание – Разряды cycle_ctrl находятся в области памяти, отведенной под channel_cfg – см. подраздел «Структура управляющих данных канала»	

Для всех типов циклов DMA повторный арбитраж происходит после 2^R передач DMA. Если установить длинный период арбитража на низкоприоритетном канале, то это

заблокирует все запросы на обработку от других каналов до тех пор, пока не будут выполнены 2^R передач DMA по данному каналу. Поэтому, устанавливая значение R, необходимо учитывать, что это может привести к повышенному времени отклика на запрос на обработку от высокоприоритетных каналов.

Данный раздел описывает следующие типы циклов DMA:

– недействительный;
– основной;
– авто-запрос;
– «пинг-понг»;
– работа с памятью в режиме «исполнение с изменением конфигурации»;
– работа с периферией в режиме «исполнение с изменением конфигурации».

25.4.5.1 **Недействительный**

После окончания цикла DMA контроллер устанавливает тип цикла в значение «недействительный» для предотвращения повтора выполненного цикла DMA.

25.4.5.2 **Основной**

В этом режиме контроллер работает только с основными или альтернативными управляющими данными канала. После того, как разрешена работа канала, и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

1. Контроллер выполняет 2^R передач. Если число оставшихся передач 0, контроллер переходит к шагу 3.

2. Осуществление арбитража:

– если высокоприоритетный канал выдает запрос на обработку, то контроллер начинает обслуживание этого канала;

– если периферийный блок или программное обеспечение выдает запрос на обработку (повторный запрос на обработку по каналу), то контроллер переходит к шагу 1.

3. Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала HCLK. Это указывает центральному процессору на завершение цикла DMA.

25.4.5.3 **Авто-запрос**

Функционируя в данном режиме, контроллер ожидает получения одиночного запроса на обработку для разрешения работы и выполнения цикла DMA. Такая работа позволяет выполнять передачу больших пакетов данных без существенного увеличения времени отклика на обслуживание высокоприоритетных запросов и не требует множественных запросов на обработку от процессора или периферийных блоков.

Контроллер позволяет выбрать для использования первичную или альтернативную структуру управляющих данных канала. После того как разрешена работа канала и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

1 Контроллер выполняет 2^R передач для канала C. Если число оставшихся передач 0, контроллер переходит к шагу 3.

2 Контроллер осуществляет арбитраж. Когда канал C становится каналом с самым высоким приоритетом, контроллер переходит к шагу 1.

3 Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала HCLK. Это указывает центральному процессору на завершение цикла DMA.

25.4.5.4 **Пинг-понг**

В данном режиме контроллер выполняет цикл DMA, используя одну из структур управляющих данных, а затем выполняет еще один цикл DMA, используя другую структуру управляющих данных. Контроллер выполняет циклы DMA с переключением структур до тех пор, пока не считает «недействительную» структуру данных или пока процессор не запретит работу канала.

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

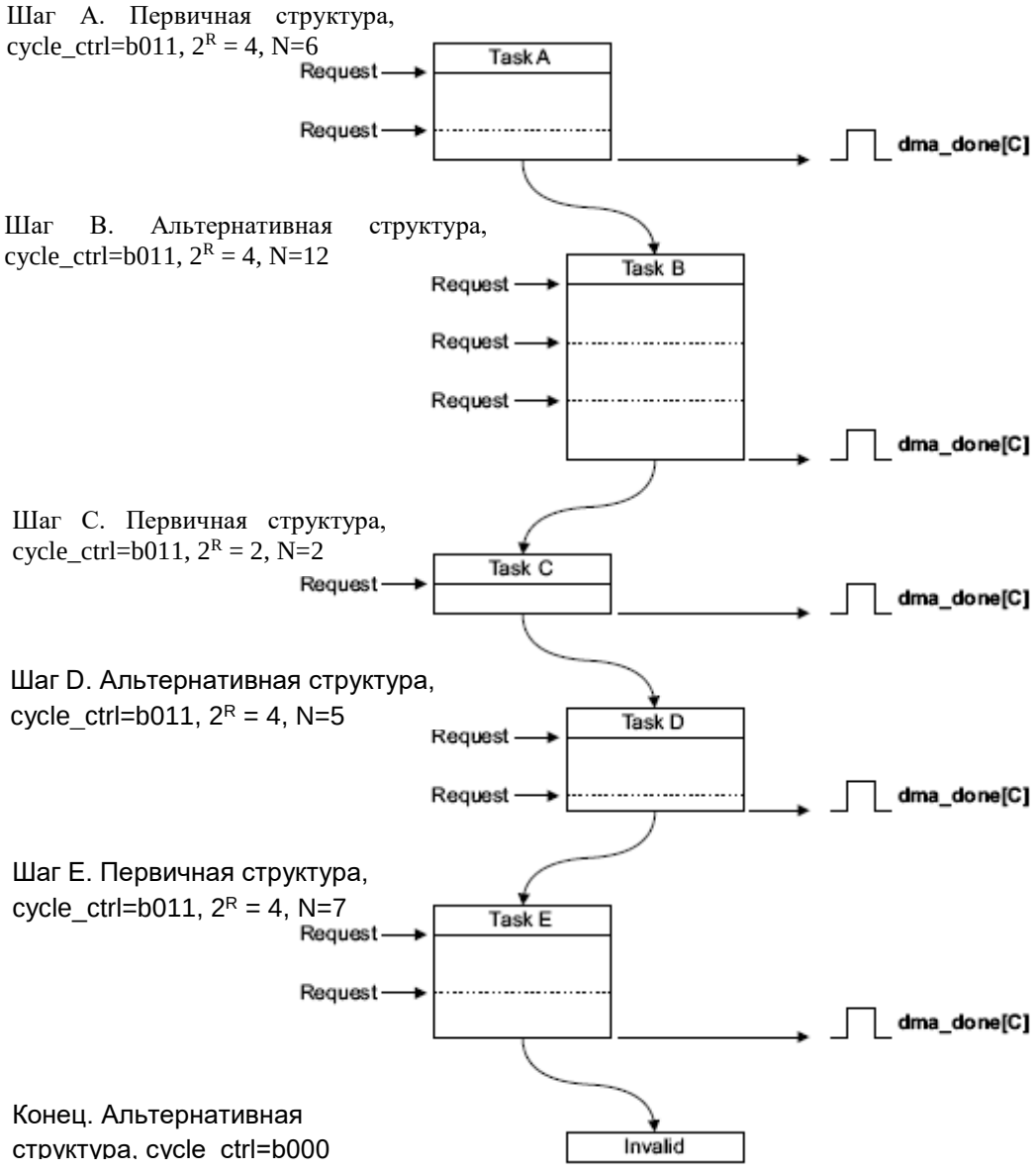


Рисунок 93 – Пример функционирования контроллера в режиме «пинг-понг»

Пояснения к рисунку 93

Шаг A — Процессор устанавливает первичную структуру управляющих данных для шага A.
Процессор устанавливает альтернативную структуру управляющих данных для шага B. Это позволит контроллеру переключиться к шагу B незамедлительно после выполнения шага A, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.
Контроллер получает запрос и выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 2 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража

После выполнения шага A процессор может установить первичные управляющие данные канала для шага C. Это позволит контроллеру переключиться к шагу C незамедлительно после выполнения шага B, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.

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

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

После выполнения шага B процессор может установить альтернативные управляющие данные канала для шага D.

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

Шаг C — Контроллер выполняет 2 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации HCLK и входит в процедуру арбитража.

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

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

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

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

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

Если контроллер получит новый запрос на обработку от данного канала и этот запрос будет самым приоритетным, контроллер предпримет попытку выполнения следующего шага. Однако из-за того, что процессор не установил альтернативные управляющие данные, и по окончании шага D контроллер установил cycle_ctrl в состояние b000, передачи DMA прекращаются.

Примечание – Для прерывания цикла DMA, исполняемого в режиме «пинг-понг», также возможен перевод режима работы контроллера на шаге E в режим «Основной цикл DMA» путем установки cycle_ctrl в 3’b001.

25.4.5.5 Режим работы с памятью «исполнение с изменением конфигурации»

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные. Затем контроллер выполняет еще 4 передачи DMA, используя первичные управляющие данные. Контроллер продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

– процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;

– контроллер считает «неправильную» структуру управляющих данных.

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

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

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. В таблице 297 приведены области памяти channel_cfg, которые должны быть определены константами, а также, значения которых определяются пользователем.

Таблица 297 – Channel_cfg для первичной структуры управляющих данных в режиме работы с памятью «исполнение с изменением конфигурации»

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
31, 30	dst_inc	2’b10	Контроллер производит инкремент адреса пословно
29, 28	dst_size	2’b10	Контроллер осуществляет передачу пословно
27, 26	src_inc	2’b10	Контроллер производит инкремент адреса пословно
25, 24	src_size	2’b10	Контроллер осуществляет передачу пословно
17…14	R_power	4’b0010	Контроллер выполняет 4 передачи DMA
3	next_useburst	1’b0	Для данного режима этот разряд должен быть равен 0
2…0	cycle_ctrl	3’b100	Контролер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями, определяемыми пользователем			
23…21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20…18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13…4	n_minus_1	N*	Настраивает контроллер на выполнение N передач DMA, где N кратно 4
* Так как разряды R_power установлены в состояние 2, необходимо задавать значение N кратное 4. Число равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные			

Рисунок 94 демонстрирует пример функционирования в режиме работы с памятью «исполнение с изменением конфигурации».

Инициализация:

1. Настройка первичных управляющих данных для разрешения копирования A, B, C и D: cycle_ctrl=3’b100, 2^R=4, N=16.

2. Запись первичных данных в память с использованием структуры, показанной в таблице ниже.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	0x0A000000	0x0AE00000	cycle_ctrl = b101, 2^R = 4, N = 3	0xXXXXXXXX
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b101, 2^R = 2, N = 8	0xXXXXXXXX
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b101, 2^R = 8, N = 5	0xXXXXXXXX
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2^R = 4, N = 4	0xXXXXXXXX

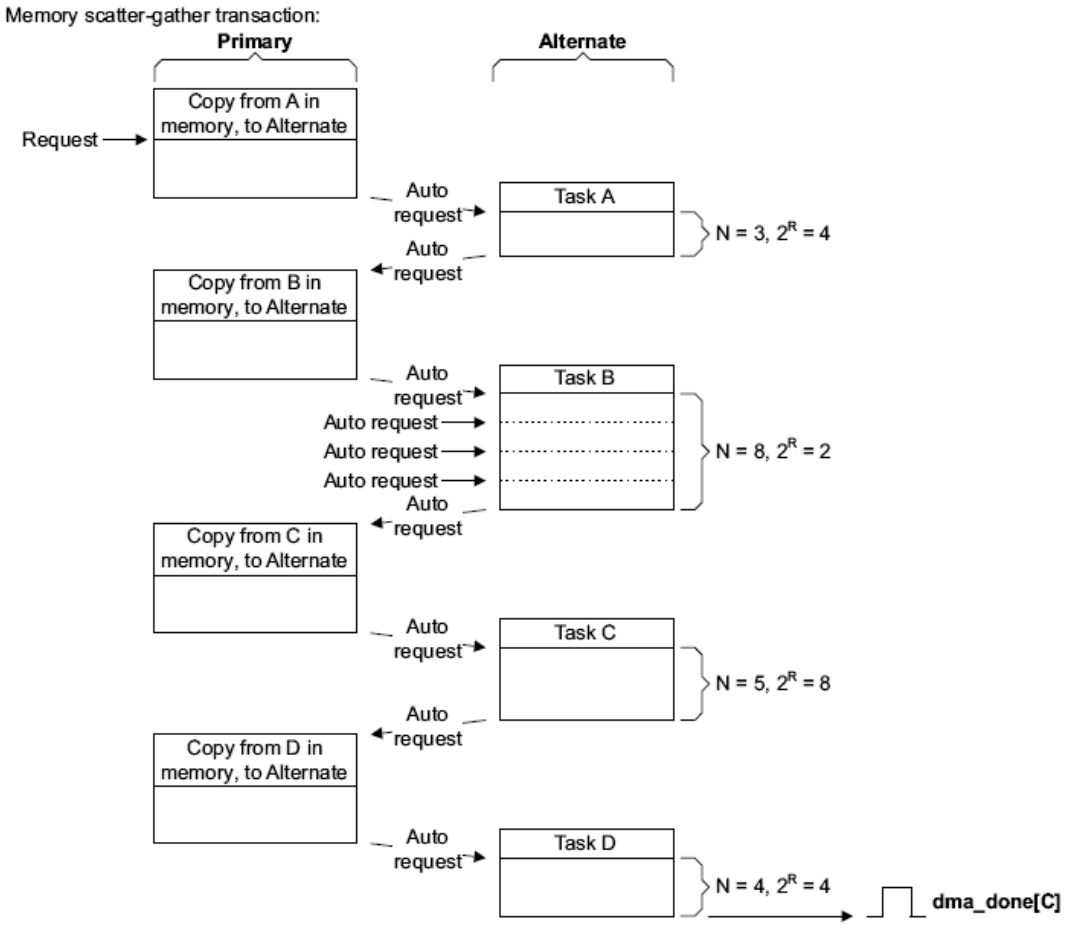


Рисунок 94 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

Инициализация:

1. Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с памятью «исполнение с изменением конфигурации» путем установки cycle_ctrl в 3b’100. Так как управляющие данные канала состоят из 4 слов, необходимо установить 2^R в 4. В этом примере количество передач равно 4 и поэтому N установлен в 16.

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

3. Процессор разрешает работу канала DMA.

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

Первичная, копирование A

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

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

Шаг A

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

Первичная, копирование B

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

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

Шаг B

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

Первичная, копирование C

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

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

Шаг C

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

Первичная, копирование D

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

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

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

Шаг D

Контроллер выполняет шаг D, используя основной цикл DMA.

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

25.4.5.6 Режим работы с периферией «исполнение с изменением конфигурации»

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

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

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

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

– процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;

– контроллер считает «неправильную» структуру управляющих данных.

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

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

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. В таблице 298 приведены области памяти channel_cfg, которые должны быть определены константами, и те области, значения которых определяются пользователем.

Таблица 298 – Channel_cfg для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации»

Разряды	Обозначение	Значение	Описание
Области с константными значениями			
31, 30	dst_inc	2'b10	Контроллер производит инкремент адреса пословно
29, 28	dst_size	2'b10	Контроллер осуществляет передачу пословно
27, 26	src_inc	2'b10	Контроллер производит инкремент адреса пословно
25, 24	src_size	2'b10	Контроллер осуществляет передачу пословно
17…14	R_power	4'b0010	Контроллер выполняет 4 передачи DMA
2…0	cycle_ctrl	3'b110	Контролер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями, определяемыми пользователем			
23…21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20…18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13…4	n_minus_1	N*	Настраивает контроллер на выполнение N передач DMA, где N кратно 4
3	next_useburst	-	При установке в 1, контроллер установит chnl_useburst_set[C] в 1 после выполнения передачи с альтернативной структурой
* Так как разряды R_power установлены в состояние 2, необходимо задавать значение N кратное 4. Число равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные			

Рисунок 95 демонстрирует пример функционирования в режиме работы с периферией «исполнение с изменением конфигурации».

Инициализация:

1. Настройка первичных управляющих данных для разрешения копирования A, B, C и D: cycle_ctrl=3'b110, 2^R=4, N=16.

2. Запись первичных данных в память с использованием структуры, показанной в таблице ниже.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	0x0A000000	0x0AE00000	cycle_ctrl = b111, 2^R = 4, N = 3	0xXXXXXXXX
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b111, 2^R = 2, N = 8	0xXXXXXXXX
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b111, 2^R = 8, N = 5	0xXXXXXXXX
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2^R = 4, N = 4	0xXXXXXXXX

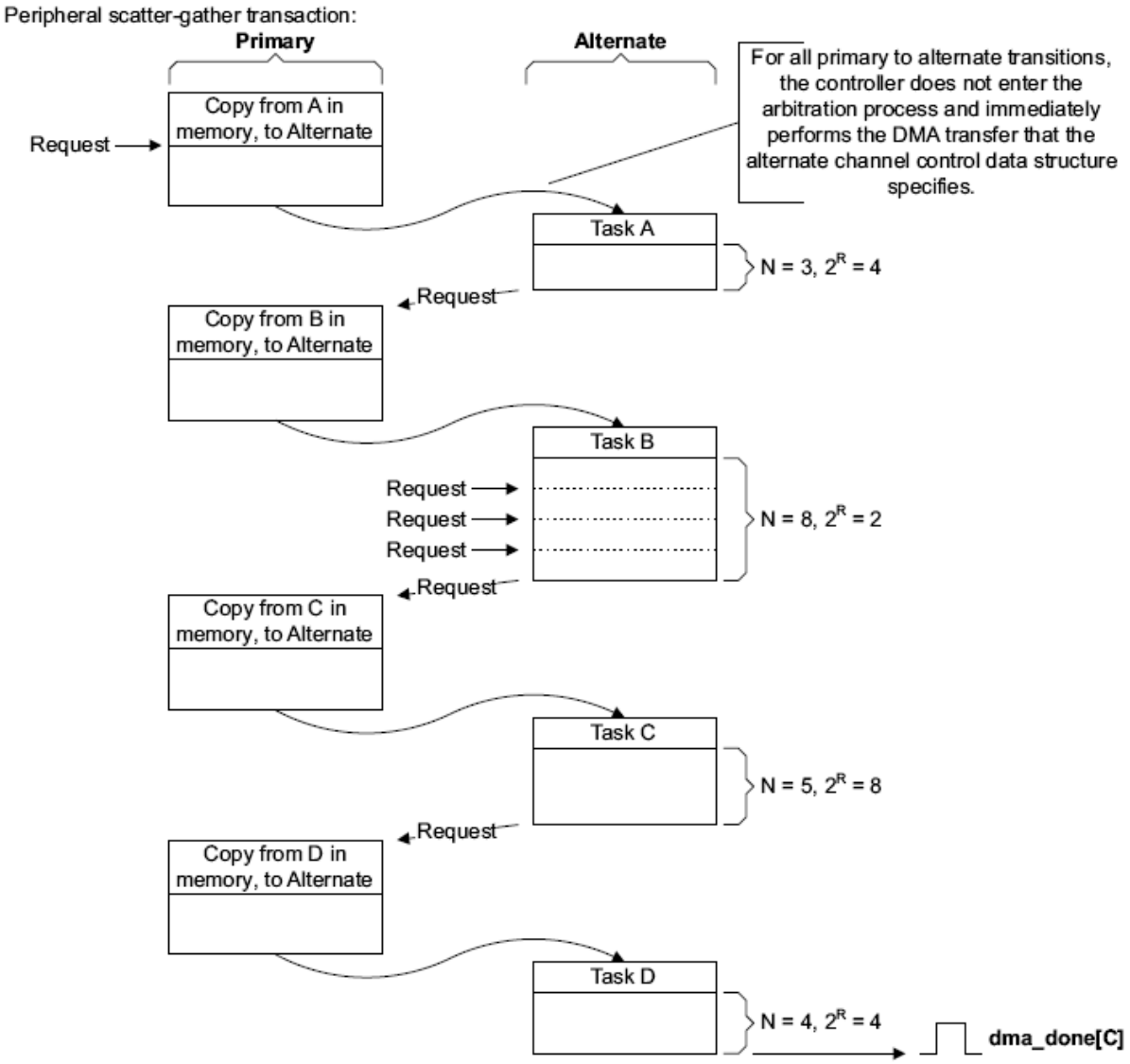


Рисунок 95 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

Инициализация:

1. Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации» путем установки cycle_ctrl в 3’b110. Так как управляющие данные канала состоят из 4 слов, необходимо установить 2^R в 4. В этом примере количество задач равно 4 и поэтому N установлено в 16.

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

3. Процессор разрешает работу канала DMA.

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

Первичная, копирование из области A памяти

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

Шаг A

Контроллер выполняет шаг A.

По окончании контроллер проводит процедуру арбитража.

Первичная, копирование из области B памяти.

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

Шаг B

Контроллер выполняет шаг B. Для завершения задачи периферия должна установить последовательно 3 запроса.

По окончании контроллер проводит процедуру арбитража.

Первичная, копирование из области C памяти.

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

Шаг C

Контроллер выполняет шаг C.

По окончании контроллер проводит процедуру арбитража.

После выставления периферией нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается следующим образом:

Первичная, копирование из области D памяти.

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

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

Шаг D

Контроллер выполняет шаг D, используя основной цикл DMA.

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

25.4.5.7 **Индикация ошибок**

При получении контроллером по шине AHB ответа об ошибке, он выполняет следующие действия:

– отключает канал, связанный с ошибкой;

– устанавливает флаг dma_err в состояние 1.

После обнаружения процессором флага dma_err процессор определяет номер канала, который был активен в момент появления ошибки. Для этого он осуществляет следующее:

1 Чтение регистра chnl_enable_set с целью создания списка отключенных каналов;

2 Если канал установил флаг dma_done[], то контроллер отключает канал. Программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно установили флаги dma_done[]. Выставление сигнала dma_done транслируется в регистр DMA_DONE_STICK блока контроллера тактовой частоты (см. пункт 16.5.9 «DMA_DONE_STICK»);

3 Процессор должен сравнить список выключенных каналов, полученный в шаге 1, с данными о каналах, которые недавно устанавливали флаги dma_done[]. Канал, по которому отсутствуют данные об установке флага dma_done[], это и есть канал, с которым связана ошибка.

25.5 **Структура управляющих данных канала**

В системной памяти должна быть отведена область для хранения управляющих данных каналов. Системная память должна:

– предоставлять смежную область системной памяти, к которой контроллер и процессор имеют доступ;

– иметь базовый адрес, который целочисленно кратен общему размеру структуры управляющих данных канала.

На рисунке 96 приведена область памяти, необходимая контроллеру для структур управляющих данных канала при использовании 32 каналов и опциональной альтернативной структуры управляющих данных.

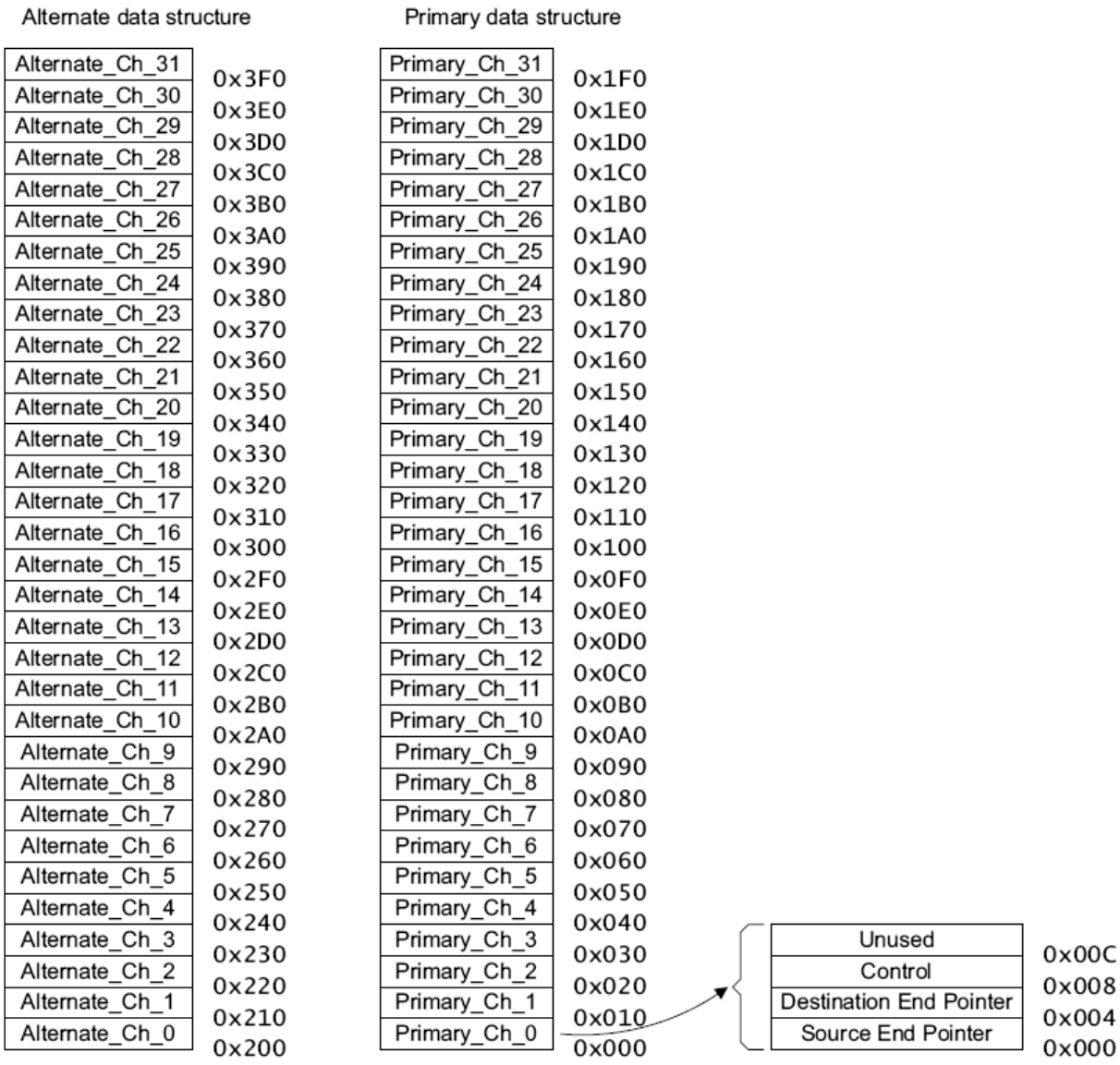


Рисунок 96 – Карта памяти для 32-х каналов, включая альтернативную структуру управляющих данных

Пример использует 1 Кбайт системной памяти. В этом примере контроллер использует младшие 10 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXX000, далее 0xXXXXX400, далее 0xXXXXX800, далее 0xXXXXXC00.

Базовый адрес для первичной структуры управляющих данных устанавливается путем записи соответствующего значения в регистр ctrl_base_ptr.

Необходимый размер области системной памяти зависит от:

- количества каналов, используемых в контроллере;
- от того, используется или нет альтернативная структура управляющих данных.

В таблице 299 приведены разряды адреса, которые используются контроллером при доступе к различным элементам структуры управляющих данных, в зависимости от количества каналов, используемых в контроллере.

Таблица 299 – Разряды адреса, соответствующие элементам структуры управляющих данных

Количество каналов, используемых в контроллере	[9]	[8]	[7]	[6]	[5]	[4]	[3:0]
1						A	0x0 0x4 0x8
2					A	C[0]	
3-4				A	C[1]	C[0]	
5-8			A	C[2]	C[1]	C[0]	
9-16		A	C[3]	C[2]	C[1]	C[0]	
17-32	A	C[4]	C[3]	C[2]	C[1]	C[0]	

где A выбирает одну из структур управляющих данных канала:

A = 0 выбирает первичную структуру управляющих данных;

A = 1 выбирает альтернативную структуру управляющих данных.

C[x:0] выбирает канал DMA.

Address[3:0] выбирает один из управляющих элементов:

0x0 выбирает указатель конца данных источника;

0x4 выбирает указатель конца данных приемника;

0x8 выбирает конфигурацию управляющих данных;

0xC контроллер не имеет доступа к этому адресу.

Если это необходимо, то возможно разрешить процессору использовать эти адреса в качестве системной памяти.

Примечание – Совсем не обязательно вычислять базовый адрес альтернативной структуры управляющих данных, так как регистр alt_ctrl_base_ptr содержит эту информацию.

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

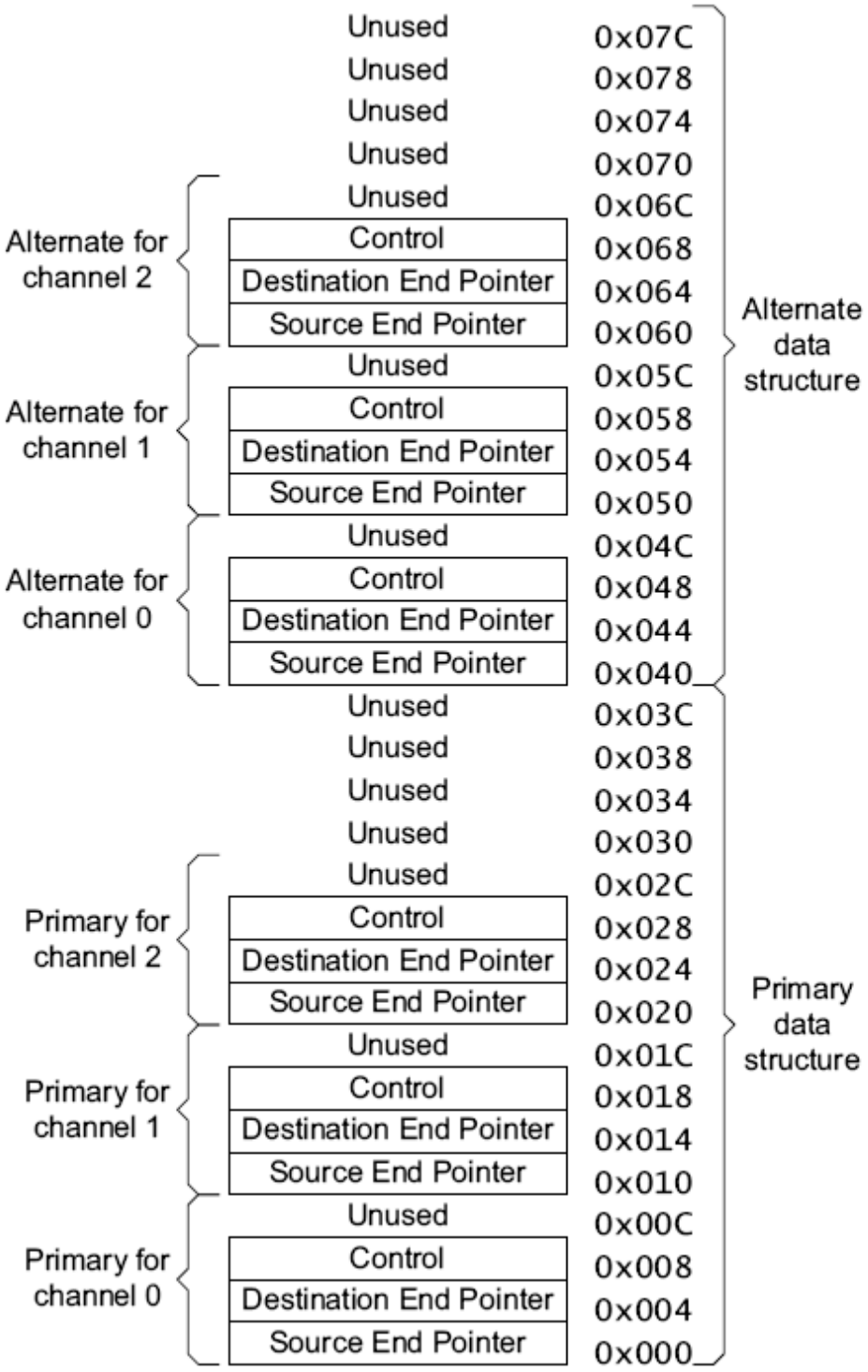


Рисунок 97 – Карта памяти для трех каналов DMA, включая альтернативную структуру управляющих данных (где Destination end pointer – указатель конца данных приемника; Source end pointer – указатель конца данных источника; Control – управление)

Пример структуры управляющих данных использует 128 байт системной памяти. В этом примере контроллер использует младшие 6 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXXX00, далее 0xXXXXXX80.

В таблице 300 приведены разрешенные значения базового адреса для первичной структуры управляющих данных, в зависимости от количества каналов DMA, использованных в контроллере.

Таблица 300 – Разрешенные базовые адреса

Кол-во каналов DMA	Разрешенные значения базового адреса для первичной структуры управляющих данных
1	0xXXXXXX00, 0xXXXXXX20, 0xXXXXXX40, 0xXXXXXX60, 0xXXXXXX80, 0xXXXXXXA0, 0xXXXXXXC0, 0xXXXXXXE0
2	0xXXXXXX00, 0xXXXXXX40, 0xXXXXXX80, 0xXXXXXXC0
3-4	0xXXXXXX00, 0xXXXXXX80
5-8	0xXXXXX000, 0xXXXXX100, 0xXXXXX200, 0xXXXXX300, 0xXXXXX400, 0xXXXXX500, 0xXXXXX600, 0xXXXXX700, 0xXXXXX800, 0xXXXXX900, 0xXXXXXA00, 0xXXXXXB00, 0xXXXXXC00, 0xXXXXXD00, 0xXXXXXE00, 0xXXXXXF00
9-16	0xXXXXX000, 0xXXXXX200, 0xXXXXX400, 0xXXXXX600, 0xXXXXX800, 0xXXXXXA00, 0xXXXXXC00, 0xXXXXXE00
17-32	0xXXXXX000, 0xXXXXX400, 0xXXXXX800, 0xXXXXXC00

Контроллер использует системную память для доступа к двум указателям адреса конца данных и разрядам управления каждого канала. Следующие подразделы описывают эти 32-разрядные области памяти и процедуру вычисления контроллером адреса передачи DMA:

- указатель конца данных источника;
- указатель конца данных приемника;
- разряды управления;
- вычисление адреса.

25.5.1 **Указатель конца данных источника**

Область памяти под названием src_data_end_ptr содержит указатель на последний адрес месторасположения данных источника. Значения битов этой области приведены в таблице 301.

Таблица 301 – Значения разрядов src_data_end_ptr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	src_data_end_ptr	Указатель на последний адрес данных источника

Перед тем как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.

Примечание – Контроллер не имеет доступа по записи в эту область памяти.

25.5.2 **Указатель конца данных приемника**

Область памяти под названием dst_data_end_ptr содержит указатель на последний адрес месторасположения данных приемника. Значения битов этой области приведены в таблице 302.

Таблица 302 – Значения разрядов dst_data_end_ptr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	dst_data_end_ptr	Указатель на последний адрес данных приемника

Перед тем как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.

Примечание – Контроллер не имеет доступа по записи в эту область памяти.

25.5.3 **Разряды управления**

Область памяти под названием channel_cfg обеспечивает управление каждой передачей DMA. Значение битов этой области приведены в таблице 303.

Таблица 303 – Название разрядов области памяти channel_cfg

Номер	31...30	29...28	27...26	25...24	23...21
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	10	10	10	10	-
	dst_inc	dst_size	src_inc	src__size	dst_prot_crtl

Номер	20...18	17...14	13...4	3	2...0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	-	0010	-	0	100
	Src_prot_ctrl	R_power	n_minus_1	next_useburst	cycle_ctrl

Таблица 304 – Назначение разрядов channel_cfg

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	dst_inc	Шаг инкремента адреса приемника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: 2’b00 = байт; 2’b01 = полуслово (16-разрядное слово); 2’b10 = слово (32-разрядное слово); 2’b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr. Разрядность данных источника = полуслово: 2’b00 = зарезервировано; 2’b01 = полуслово;

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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
20…18	src_prot_ctrl	Задает состояние HPROT[3:1], когда контроллер считывает данные из источника. Разряд [20] управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд [19] управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд [18] управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ не привилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный
17…14	R_power	Задает количество передач DMA до выполнения контроллером процедуры арбитража. Возможные значения: 4’b0000 - арбитраж производится после каждой передачи DMA; 4’b0001 – арбитраж производится после 2 передач DMA; 4’b0010 – арбитраж производится после 4 передач DMA; 4’b0011 – арбитраж производится после 8 передач DMA; 4’b0100 – арбитраж производится после 16 передач DMA; 4’b0101 – арбитраж производится после 32 передач DMA; 4’b0110 – арбитраж производится после 64 передач DMA; 4’b0111 – арбитраж производится после 128 передач DMA; 4’b1000 – арбитраж производится после 256 передач DMA; 4’b1001 – арбитраж производится после 512 передач DMA; 4’b1010 – 4’b1111 – арбитраж производится после 1024 передач DMA. Это означает, что арбитраж не производится, так как максимальное количество передач DMA равно 1024
13…4	n_minus_1	Перед выполнением цикла DMA эти разряды указывают общее количество передач DMA, из которых состоит цикл DMA. Необходимо установить эти разряды в значение, соответствующее размеру желаемого цикла DMA. 10-разрядное число плюс 1 задает количество передач DMA. Возможные значения: 10’b0000000000 = 1 передача DMA; 10’b0000000001 = 2 передачи DMA; 10’b0000000010 = 3 передачи DMA; 10’b0000000011 = 4 передачи DMA; 10’b0000000100 = 5 передач DMA; 10’b0000000101 = 6 передач DMA; …. 10’b1111111111 = 1024 передачи DMA. Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач DMA до завершения цикла DMA
3	next_useburst	Контролирует, не установлен ли chnl_useburst_set[C] в состояние 1, если контроллер работает в режиме работы с периферией «Исполнение с изменением конфигурации», и, если контроллер завершает цикл DMA, используя альтернативные управляющие данные.

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		Примечание – Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер устанавливает chnl_useburst_set[C] в значение 0, если количество оставшихся передач DMA меньше, чем 2^R. Установка next_useburst разряда определяет, будет ли контроллер дополнительно переопределять разряд chnl_useburst_set[C]. Если контроллер выполняет цикл DMA в режиме работы с периферией «Исполнение с изменением конфигурации», то после окончания цикла, использующего альтернативные управляющие данные, происходит следующее в зависимости от состояния next_useburst: 0 – контроллер не изменяет значение chnl_useburst_set[C]. Если chnl_useburst_set[C] установлен в 0, то для всех оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер отвечает на запросы по dma_req[] и dma_sreq[], при выполнении циклов DMA он использует альтернативные управляющие данные. 1 – контроллер изменяет значение chnl_useburst_set[C] в состояние 1. Поэтому для оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер реагирует только на запросы по dma_req[], при выполнении циклов DMA он использует альтернативные управляющие данные
2…0	cycle_ctrl	Режим работы при выполнении цикла DMA: 3’b000 Стоп. Означает, что структура управляющих данных является «неправильной»; 3’b001 Основной. Контроллер должен получить новый запрос для окончания цикла DMA, перед этим он должен выполнить процедуру арбитража; 3’b010 Авто-запрос. Контроллер автоматически осуществляет запрос на обработку по соответствующему каналу в течение процедуры арбитража. Это означает, что начального запроса на обработку достаточно для выполнения цикла DMA; 3’b011 Пинг-понг. Контроллер выполняет цикл DMA использую одну из структур управляющих данных. По окончании выполнения цикла DMA, контроллер; выполняет следующий цикл DMA, используя другую структуру. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя процессору перенастраивать неактивную структуру данных. Контроллер продолжает выполнять циклы DMA, до тех пор, пока он не прочитает «неправильную» структуру данных или пока процессор не изменит cycle_ctrl поле в состояние 3’b001 или 3’b010; 3’b100 Режим работы с памятью «Исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть 3’b100; 3’b101 Режим работы с памятью «Исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		альтернативной структуре управляющих данных должно быть 3’b101; 3’b110 Режим работы с периферией «исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в первичной структуре управляющих данных должно быть 3’b110; 3’b111 Режим работы с периферией «исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть 3’b111

В начале цикла DMA или 2^R передачи DMA контроллер считывает значение channel_cfg из системной памяти. После выполнения 2^R или N передач он сохраняет обновленное значение channel_cfg в системную память.

Контроллер не поддерживает значений dst_size, отличных от значений src_size. Если контроллер обнаруживает неравные значения этих полей, он использует значение src_size в качестве размера данных и приемника, и источника и при ближайшем обновлении поля n_minus_1, он также устанавливает значение поля dst_size, равное src_size.

После выполнения контроллером N передач, контроллер устанавливает значение поля cycle_ctrl в 3’b000, делая тем самым channel_cfg данные «неправильными». Это позволяет избежать повторения выполненной передачи DMA.

25.5.3.1 *Вычисление адреса*

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

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

src_inc=b00 and dst_inc=b00

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

Src_inc=b01 and dst_inc=b01

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

Src_inc=b01 and dst_inc=b10

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

Src_inc=b11 and dst_inc=b11

- адрес источника = src_data_end_ptr
- адрес приемника = dst_data_end_ptr.

В таблице 305 приведены адреса приемника цикла DMA для 6 слов.

Таблица 305 – Цикла DMA для 6 слов с пословным инкрементом

Начальные значения channel_cfg перед циклом DMA				
src_size=2'b10, dst_inc=2'b10, n_minus_1=3'b101, cycle_ctrl=1				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения channel_cfg после цикла DMA				
src_size=2'b10, dst_inc=2'b10, n_minus_1=0, cycle_ctrl=0				
* Значение, полученное после сдвига влево значения счетчика на количество разрядов соответствующее dst_inc				

В таблице 306 приведены адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.

Таблица 306 – Цикла DMA для 12 байт с «полусловным» инкрементом

Начальные значения channel_cfg перед циклом DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=4'b1011, cycle_ctrl=1, R_power=2'b11				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x5E7	11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3
	0x5E7	9	0x12	0x5D5
	0x 5E7	8	0x10	0x5D7
	0x 5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
	0x5E7	4	0x8	0x5DF

Значения channel_cfg после 2^R передач DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=3'b011, cycle_ctrl=1, R_power=2'b11				
DMA передачи	0x 5E7	3	0x6	0x5E1
	0x 5E7	2	0x4	0x5E3
	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7
Конечные значения channel_cfg после цикла DMA				
src_size=2'b00, dst_inc=2'b01, n_minus_1=0, cycle_ctrl=0**, R_power=2'b11				
* Значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc. ** После окончания цикла DMA контроллер делает channel_cfg «неправильным», сбрасывая в 0 поле cycle_ctrl				

25.6 **Описание регистров контроллера DMA**

Данный раздел описывает регистры контроллера и управление контроллером через них.

Раздел содержит следующие сведения:

– о регистровой модели контроллера;

– описание регистров.

Основные положения регистровой модели контроллера:

– запрещается обращаться к зарезервированным или неиспользуемым адресам, поскольку это может привести к непредсказуемому поведению контроллера;

– необходимо заполнять неиспользуемые или зарезервированные разряды регистров нулями при записи и игнорировать значения таких разрядов при считывании, кроме случаев, специально описанных в разделе;

– системный сброс или сброс по установке питания сбрасывает все регистры в состояние 0, кроме случаев, специально описанных в разделе;

– все регистры поддерживают доступ по чтению и записи, кроме случаев, специально описанных в разделе. Доступ по записи обновляет содержание регистра, а доступ по чтению возвращает содержимое регистра.

Таблица 307 – Перечень регистров контроллера

Наименование	Смещение относительно базового адреса	Тип	Значение по сбросу	Описание
status	0x000	RO	0x-nn0000*	Статусный регистр DMA
cfg	0x004	WO	-	Регистр конфигурации DMA
ctrl_base_ptr	0x008	R/W	0x00000000	Регистр базового адреса управляющих данных каналов

Наименование	Смещение относительно базового адреса	Тип	Значение по сбросу	Описание
alt_ctrl_base_ptr	0x00C	RO	0x000000nn**	Регистр базового адреса альтернативных управляющих данных каналов
waitonreq_status	0x010	RO	0x43FFFFFF	Регистр статуса ожидания запроса на обработку каналов
chnl_sw_request	0x014	WO	-	Регистр программного запроса на обработку каналов
chnl_useburst_set	0x018	R/W	0x00000000	Регистр установки пакетного обмена каналов
chnl_useburst_clr	0x01C	WO	-	Регистр сброса пакетного обмена каналов
chnl_req_mask_set	0x020	R/W	0x00000000	Регистр маскирования запросов на обслуживание каналов
chnl_req_mask_clr	0x024	WO	-	Регистр очистки маскирования запросов на обслуживание каналов
chnl_enable_set	0x028	R/W	0x00000000	Регистр установки разрешения каналов
chnl_enable_clr	0x02C	WO	-	Регистр сброса разрешения каналов
chnl_pri_alt_set	0x030	R/W	0x00000000	Регистр установки первичной/альтернативной структуры управляющих данных каналов
chnl_pri_alt_clr	0x034	WO	-	Регистр сброса первичной/альтернативной структуры управляющих данных каналов
chnl_priority_set	0x038	R/W	0x00000000	Регистр установки приоритета каналов
chnl_priority_clr	0x03C	WO	-	Регистр сброса приоритета каналов
-	0x040-0x048		-	зарезервировано
err_clr	0x04C	R/W	0x00000000	Регистр сброса флага ошибки
-	0x050-0xDFC	-		зарезервировано
* Значение по сбросу зависит от количества каналов DMA, использованных в контроллере, а также от наличия интегрированной схемы тестирования. ** Значение по сбросу зависит от количества каналов DMA, использованных в контроллере				

25.6.1 **Статусный регистр DMA STATUS**

Данный регистр имеет доступ только на чтение. При чтении регистр возвращает состояние контроллера. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

Таблица 308 – Статусный регистр DMA

Номер	31...28	27...21	20...16	15...8	7...4	3...1	0
Доступ	RO	U	RO	U	RO	U	RO
Сброс	0	0	0	0	0	0	0
	test_status	-	chnls_minus1	-	state	-	master_enable

Таблица 309 – Назначение разрядов регистра dma_status

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…28	test_status	Значение при чтении: 4’b0000 = контроллер не имеет интегрированной схемы тестирования; 4’b0001 = контроллер имеет интегрированную схему тестирования; 4’b0010 – 4’b1111 = не определено
27…21	-	Не определено
20…16	chnls_minus1	Количество доступных каналов DMA минус 1. Например: 5’b00000 = контроллер имеет 1 канал DMA; 5’b00001 = контроллер имеет 2 канала DMA; 5’b00010 = контроллер имеет 3 канала DMA; … 5’b11111 = контроллер имеет 32 канала DMA
15…8	-	Не определено
7…4	state	Текущее состояние автомата управления контроллера. Состояние может быть одним из следующих: 4’b0000 = в покое; 4’b0001 = чтение управляющих данных канала; 4’b0010 = чтение указателя конца данных источника; 4’b0011 = чтение указателя конца данных приемника; 4’b0100 = чтение данных источника; 4’b0101 = запись данных в приемник; 4’b0110 = ожидание запроса на выполнение DMA; 4’b0111 = запись управляющих данных канала; 4’b1000 = приостановлен; 4’b1001 = выполнен; 4’b1010 = режим работы с периферией «Исполнение с изменением конфигурации»; 4’b1011 – 4’b1111 = не определено
3…1	-	Не определено
0	master_enable	Состояние контроллера: 0 = работа контроллера запрещена; 1 = работа контроллера разрешена

25.6.2 **Регистр конфигурации DMA CFG**

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера.

Таблица 310 – Регистр конфигурации DMA

Номер	31...8	7...5	4...1	0
Доступ	U	WO	U	WO
Сброс	0	0	0	0
	-	chnl_prot_ctrl	-	master_enable

Таблица 311 – Назначение разрядов регистра dma_cfg

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…8	-	Не определено, следует записывать 0
7…5	chnl_prot_ctrl	Определяет уровни индикации сигналов HPROT[3:1] защиты шины AHB-Lite: Разряд 7 управляет сигналом HPROT[3], с целью индикации о появлении доступа с кэшированием; Разряд 6 управляет сигналом HPROT[2], с целью индикации о появлении доступа с буферизацией; Разряд 5 управляет сигналом HPROT[1], с целью индикации о появлении привилегированного доступа. Примечание – Если разряд[n] = 1, то соответствующий сигнал HPROT в состоянии 1. Если разряд[n] = 0, то соответствующий сигнал HPROT в состоянии 0
4…1	-	Не определено. Следует записывать 0
0	master_enable	Определяет состояние контроллера: 0 – запрещает работу контроллера; 1 – разрешает работу контроллера

25.6.3 Регистр базового адреса управляющих данных каналов CTRL_BASE_PTR

Данный регистр имеет доступ на запись и чтение. Регистр определяет базовый адрес системной памяти размещения управляющих данных каналов.

Примечание – Контроллер не содержит внутреннюю память для хранения управляющих данных каналов.

Размер системной памяти, предназначенной контроллеру, зависит от количества каналов DMA, использующихся контроллером, а также от возможности использования альтернативных управляющих данных каналов. Поэтому количество разрядов регистра, необходимых для задания базового адреса, варьируется и зависит от варианта построения системы.

Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

Таблица 312 – Регистр базового адреса управляющих данных каналов

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

Таблица 313 – Назначение разрядов регистра ctrl_base_ptr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…10	ctrl_base_ptr	Указатель на базовый адрес первичной структуры управляющих данных. См. соответствующий раздел
9…0	-	Не определено. Следует записывать 0

25.6.4 Регистр базового адреса альтернативных управляющих данных каналов ALT_CTRL_BASE_PTR

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении указатель базового адреса альтернативных управляющих данных каналов. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Этот регистр позволяет не производить вычисления базового адреса альтернативных управляющих данных каналов.

Таблица 314 – Регистр базового адреса альтернативных управляющих данных каналов

Номер	31... 0
Доступ	RO
Сброс	0
	alt_ctrl_base_ptr

Таблица 315 – Назначение разрядов регистра alt_ctrl_base_ptr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	alt_ctrl_base_ptr	Указатель базового адреса альтернативной структуры управляющих данных

25.6.5 Регистр статуса ожидания запроса на обработку каналов WAITONREQ_STATUS

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении состояние сигналов dma_waitonreq[]. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

Таблица 316 – Регистр статуса ожидания запроса на обработку каналов

Номер	31	.....	2	1	0
Доступ	RO	.....	RO	RO	RO
Сброс	0	.....	1	1	1
	dma_waitonreg_ status for dma_waitnreg [31]	.....	dma_waitonreg_ status for dma_waitnreg [2]	dma_waitonreg_ status for dma_waitnreg [1]	dma_waitonreg_status for dma_waitnreg [0]

Таблица 317 – Назначение разрядов регистра dma_waitonreq_status

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	dma_waitonreq_status	Состояние сигналов ожидания запроса на обработку каналов DMA. При чтении: Разряд [C] = 1 означает, что dma_waitonreq[C] в состоянии 1; Разряд [C] = 0 означает, что dma_waitonreq[C] в состоянии 0

25.6.6 Регистр программного запроса на обработку каналов CHNL_SW_REQUEST

Данный регистр имеет доступ только на запись. Регистр позволяет устанавливать программно запрос на выполнение цикла DMA.

Таблица 318 – Регистр программного запроса на обработку каналов

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

Таблица 319 – Назначение разрядов регистра chnl_sw_request

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_sw_request	Устанавливает соответствующий разряд для генерации программного запроса на выполнение цикла DMA по соответствующему каналу DMA. При записи: Разряд [C] = 0 означает, что запрос на выполнение цикла DMA по каналу C не будет установлен; Разряд [C] =1 означает, что запрос на выполнение цикла DMA по каналу C будет установлен. Запись разряда, соответствующего нереализованному каналу, означает, что запрос на выполнение цикла DMA не будет установлен

25.6.7 Регистр установки пакетного обмена каналов CHNL_USEBURST_SET

Данный регистр имеет доступ на чтение и запись. Регистр отключает выполнение одиночных запросов по установке dma_sreq[] и поэтому будут обрабатываться и исполняться только запросы по dma_req[]. Регистр возвращает при чтении состояние установок пакетного обмена.

Таблица 320 – Регистр установки пакетного обмена каналов

Номер	31	...	2	1	0
Доступ	R/W	...	R/W	R/W	R/W
Сброс	0	...	0	0	0
	chnl_useburst_ set for channel [31]	...	chnl_useburst_ set for channel [2]	chnl_useburst_set for channel [1]	chnl_useburst_set for channel [0]

Таблица 321 – Назначение разрядов регистра chnl_useburst_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_useburst_set	Отключает обработку запросов на выполнение циклов DMA от dma_sreq[] и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные от dma_sreq[] и dma_req[]. Контроллер выполняет одиночные передачи или 2^R передач. Разряд [C] = 1 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные только от dma_req[]. Контроллер выполняет 2^R передач. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_clr регистр и установить соответствующий разряд C в 0; Разряд [C] = 1 отключает возможность обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Контроллер выполняет 2^R передач. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

После выполнения предпоследней передачи из 2^R передач, в том случае, если число оставшихся передач (N) меньше, чем 2^R, контроллер сбрасывает разряд chnl_useburst_set в 0. Это позволяет выполнять оставшиеся передачи, используя dma_sreq[] и dma_req[].

Примечание – При программировании channel_cfg значением N меньшим, чем 2^R, запрещена установка соответствующего разряда chnl_useburst_set в случае, если периферийный блок не поддерживает сигнал dma_req[].

В режиме работы с периферией «исполнение с изменением конфигурации», если разряд next_useburst установлен в channel_cfg, то контроллер устанавливает chnl_useburst_set [C] в 1 после окончания цикла DMA, использующего альтернативные управляющие данные.

25.6.8 Регистр сброса пакетного обмена каналов CHNL_USEBURST_CLR

Данный регистр имеет доступ только на запись. Регистр разрешает выполнение одиночных запросов по установке dma_sreq[].

Таблица 322 – Регистр сброса пакетного обмена каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_useburst_ clr for channel [31]	.....	chnl_useburst_ clr for channel [2]	chnl_useburst_clr for channel [1]	chnl_useburst_clr for channel [0]

Таблица 323 – Назначение разрядов регистра chnl_useburst_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_useburst_clr	Установка соответствующего разряда разрешает обработку запросов на выполнение циклов DMA от dma_sreq[]. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_set регистр для отключения обработки запросов от dma_sreq[]; Разряд [C] = 1 разрешает обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.

25.6.9 Регистр маскирования запросов на обслуживание каналов CHNL_REQ_MASK_SET

Данный регистр имеет доступ на чтение и запись. Регистр отключает установку запросов на выполнение циклов DMA на dma_sreq[] и dma_req[]. Регистр возвращает при чтении состояние установок маскирования запросов от dma_sreq[] и dma_req[] на обслуживание каналов.

Таблица 324 – Регистр маскирования запросов на обслуживание каналов

Номер	31	.....	2	1	0
Доступ	R/W	.....	R/W	R/W	R/W
Сброс	0	.....	0	0	0
	chnl_reg_mask_s et for dma_req [31] and dma_sreq [31]	.....	chnl_reg_mask_s et for dma_req [2] and dma_sreq [2]	chnl_reg_mask_s et for dma_req [1] and dma_sreq [1]	chnl_reg_mask_s et for dma_req [0] and dma_sreq [0]

Таблица 325 – Назначение разрядов регистра chnl_req_mask_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_req_mask_set	Отключает обработку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на поступающие запросы; Разряд [C] = 1 означает, что канал DMA C не выполняет циклы DMA в ответ на поступающие запросы. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_req_mask_clr регистр для разрешения установки запросов; Разряд [C] = 1 отключает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

25.6.10 **Регистр очистки маскирования запросов на обслуживание каналов CHNL_REQ_MASK_CLR**

Данный регистр имеет доступ только на запись. Регистр разрешает установку запросов на выполнение циклов DMA на dma_sreq[] и dma_req[].

Таблица 326 – Регистр очистки маскирования запросов на обслуживание каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_reg_mask_clr for dma_req [31] and dma_sreq [31]	.....	chnl_reg_mask_clr for dma_req [2] and dma_sreq [2]	chnl_reg_mask_clr for dma_req [1] and dma_sreq [1]	chnl_reg_mask_clr for dma_req [0] and dma_sreq [0]

Таблица 327 – Назначение разрядов регистра chnl_req_mask_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_req_mask_clr	Установка соответствующего разряда разрешает установку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_req_mask_set регистр для отключения установки запросов; Разряд [C] = 1 разрешает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

25.6.11 **Регистр установки разрешения каналов CHNL_ENABLE_SET**

Данный регистр имеет доступ на чтение и запись. Регистр разрешает работу каналов DMA. Регистр возвращает при чтении состояние разрешений работы каналов DMA.

Таблица 328 – Регистр установки разрешения каналов

Номер	31	.....	2	1	0
Доступ	R/W	.....	R/W	R/W	R/W
Сброс	0	.....	0	0	0
	chnl_enable_set for channel [31]	.....	chnl_enable_set for channel [2]	chnl_enable_set for channel [1]	chnl_enable_set for channel [0]

Таблица 329 – Назначение разрядов регистра chnl_enable_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_enable_set	Разрешает работу каналов DMA и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C отключен; Разряд [C] = 1 означает, что работа канала DMA C разрешена. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_enable_clr регистр для отключения канала; Разряд [C] = 1 разрешает работу канала DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

25.6.12 **Регистр сброса разрешения каналов CHNL_ENABLE_CLR**

Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов DMA.

Таблица 330 – Регистр сброса разрешения каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_enable_clr for channel 31	.....	chnl_enable_clr for channel 2	chnl_enable_clr for channel 1	chnl_enable_clr for channel 0

Таблица 331 – Назначение разрядов регистра chnl_enable_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_enable_clr	Установка соответствующего разряда запрещает работу соответствующего канала DMA. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_enable_set регистр для разрешения работы канала; Разряд [C] = 1 запрещает работу канала DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. Примечание – Контроллер может отключить канал DMA, установив соответствующий разряд в следующих случаях: - при завершении цикла DMA; - при чтении из channel_cfg с полем cycle_ctrl установленным в 3’b000; - при появлении ошибки на шине AHB-Lite

25.6.13 **Регистр установки первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_SET**

Данный регистр имеет доступ на запись и чтение. Регистр разрешает работу канала DMA с использованием альтернативной структуры управляющих данных. Чтение регистра возвращает состояние каналов DMA (какую структуру управляющих данных использует каждый канал DMA).

Таблица 332 – Регистр установки первичной/альтернативной структуры управляющих данных каналов

Номер	31	.....	2	1	0
Доступ	R/W	.....	R/W	R/W	R/W
Сброс	0	.....	0	0	0
	chnl_pri_alt_set for channel [31]	.....	chnl_pri_alt_set for channel [2]	chnl_pri_alt_set for channel [1]	chnl_pri_alt_set for channel [0]

Таблица 333 – Назначение разрядов регистра chnl_pri_alt_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_pri_alt_set	Установка соответствующего разряда подключает использование альтернативных управляющих данных для соответствующего канала DMA, чтение возвращает состояние этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C использует первичную структуру управляющих данных; Разряд [C] = 1 означает, что канал DMA C использует альтернативную структуру управляющих данных. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_clr регистр для сброса разряда [C] в 0; Разряд [C] = 1 подключает использование альтернативной структуры управляющих данных каналом DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. Примечание – Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях: - при завершении 4-х передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; - при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «Пинг-понг»; - при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: - «пинг-понг»; - работа с памятью «Исполнение с изменением конфигурации»; - работа с периферией «Исполнение с изменением конфигурации

25.6.14 **Регистр сброса первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_CLR**

Данный регистр имеет доступ только на запись. Регистр разрешает работу канала DMA с использованием первичной структуры управляющих данных.

Таблица 334 – Регистр сброса первичной/альтернативной структуры управляющих данных каналов

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

Таблица 335 – Назначение разрядов регистра chnl_pri_alt_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_pri_alt_clr	Установка соответствующего разряда подключает использование первичных управляющих данных для соответствующего канала DMA. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_set регистр для выбора альтернативных управляющих данных; Разряд [C] = 1 подключает использование первичной структуры управляющих данных каналом DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. Примечание – Контроллер может переключить значение разряда chnl_pri_alt_clr[C] в следующих случаях: - при завершении 4-х передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; - при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «пинг-понг»; - при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: - «пинг-понг» - работа с памятью «Исполнение с изменением конфигурации» - работа с периферией «Исполнение с изменением конфигурации»

25.6.15 **Регистр установки приоритета каналов CHNL_PRIORITY_SET**

Данный регистр имеет доступ на запись и чтение. Регистр позволяет присвоить высокий приоритет каналу DMA. Чтение регистра возвращает состояние приоритета каналов DMA.

Таблица 336 – Регистр установки приоритета каналов

Номер	31	...	2	1	0
Доступ	R/W	…	R/W	R/W	R/W
Сброс	0	...	0	0	0
	chnl_priorit_set for channel [31]	...	chnl_priority_set for channel [2]	chnl_priority_set for channel [1]	chnl_priority_set for channel [0]

Таблица 337 – Назначение разрядов регистра chnl_priority_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_priority_set	Установка высокого приоритета каналу DMA, чтение возвращает состояние приоритета каналов DMA. При чтении: Разряд [C] = 0 означает, что каналу DMA C присвоен уровень приоритета по умолчанию; Разряд [C] = 1 означает, что каналу DMA C присвоен высокий уровень приоритета. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_clr регистр для установки каналу C уровня приоритета по умолчанию; Разряд [C] = 1 устанавливает каналу DMA C высокий уровень приоритета. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

25.6.16 **Регистр сброса приоритета каналов CHNL_PRIORITY_CLR**

Данный регистр имеет доступ только на запись. Регистр позволяет присвоить каналу DMA уровень приоритета по умолчанию.

Таблица 338 – Регистр сброса приоритета каналов

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

Таблица 339 – Назначение разрядов регистра chnl_priority_clr

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

25.6.17 **Регистр сброса флага ошибки ERR_CLR**

Данный регистр имеет доступ на запись и чтение. Регистр позволяет сбрасывать сигнал dma_err в 0. Чтение регистра возвращает состояние сигнала dma_err.

Таблица 340 – Регистр сброса флага ошибки

Номер	31…1	0
Доступ	U	R/W
Сброс	0	0
	-	err_clr

Таблица 341 – Назначение разрядов регистра err_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…1	-	Не определено. Следует записывать 0
0	err_clr	Установка сигнала в состояние 0, чтение возвращает состояние сигнала (флага) dma_err. При чтении: Разряд [C] = 0 означает, что dma_err находится в состоянии 0; Разряд [C] = 1 означает, что dma_err находится в состоянии 1. При записи: Разряд [C] =0 не дает эффекта. Состояние dma_err останется неизменным; Разряд [C] =1 сбрасывает сигнал (флаг) dma_err в состояние 0. Примечание – При сбросе сигнала dma_err одновременно с появлением ошибки на шине AHB-Lite, приоритет отдается ошибке, и, следовательно, значение регистра (и dma_err) останется неизменным (несброшенным)

26 Прерывания

Согласно спецификации привилегированной системы команд, RISC-V ядро должно обеспечивать поддержку обработки программных прерываний (machine software interrupt) и прерываний таймера (machine timer interrupt). За генерацию запросов (сигналов) этих прерываний отвечает локальный контроллер прерываний (Core Local Interruptor, CLINT). CLINT содержит регистры для управления программными прерываниями и прерываниями таймера. Карта регистров CLINT приведена в таблице 342

Таблица 342 – Регистры CLINT

Адрес	Название	Доступ
0x0200_0000	Регистр запроса/снятия программного прерывания	R/W
0x0200_4000	Регистр сравнения таймера	R/W
0x0200_BFF8	Регистр значения таймера	R/W

Таблица 343 – Регистр запроса/снятия прерывания

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

Таблица 344 – Регистр значения таймера

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

Регистр содержит 64-битное значение таймера. Значение увеличивается по сигналу timer_pulse (передний фронт частоты HCLK деленной на значение, установленное в регистре DIV_SYS_TIM) внешнего интерфейса процессорного комплекса.

Таблица 345 – Регистр сравнения таймера

Биты	Название	Доступ	Описание	Значение
31…0	mtimecmp_lo	R/W	Младшая часть значения регистра сравнения	0
64…32	mtimecmp_hi	R/W	Старшая часть значения регистра сравнения	0

Регистр содержит 64-битное значение сравнения с таймером. Запрос на прерывание таймера установлен, если значение таймера строго больше значения сравнения.

27 Контроллер обработки локальных прерываний CLIC

Контроллер обработки локальных прерываний (контроллер CLIC) получает сигналы прерывания и передаёт ядру прерывание, подлежащее обработке. Контроллер CLIC поддерживает 47 входов прерываний в ядро. Каждый вход прерывания i имеет четыре 8-разрядных регистра управления, отображенных в памяти: бит ожидания прерывания clicintip[i], бит разрешения прерывания clicintie[i], атрибуты прерывания clicintattr[i] для указания режима привилегированности и типа триггера, и биты управления прерыванием для указания уровня и приоритета clicintctl[i]. Первые 16 входов прерываний подключены к сигналам базовых локальных прерываний. Остальные входы прерываний подключены к сигналам прерываний, которые формируются периферийными блоками.

27.1 Вытеснение прерываний

Контроллер CLIC реализует поддержку вытеснения прерываний, основанную на разделении прерываний по уровням. Поддерживается до 16 уровней прерываний для каждого режима привилегированности, при этом уровни прерываний с более высоким номером могут вытеснять уровни прерываний с более низким номером. Уровень прерывания 0 соответствует обычному выполнению вне обработчика прерываний. Уровни 1 – 15 соответствуют уровням обработчика прерываний. Входящие прерывания с более высоким уровнем могут вытеснить активный обработчик прерываний, работающий с более низким уровнем в том же режиме привилегированности, при условии, что прерывания глобально разрешены в этом режиме привилегированности. Аналогично поведение прерываний, при котором входящие прерывания для режима с более высокими привилегиями могут вытеснять активный обработчик прерываний, запущенный в режиме с более низкими привилегиями, независимо от разрешения глобального прерывания в режиме с более низкими привилегиями.

Программная обработка вытеснения прерываний описана в документе «Smclic» Core-Local Interrupt Controller (CLIC) RISC-V Privileged Architecture Extension», версия 0.9-12/21/2021.

27.2 Взаимодействие контроллера CLIC с другими локальными прерываниями

Контроллер CLIC включает в себя функционал базовых локальных прерываний, таким образом биты управления базовыми локальными прерываниями не отображаются в младших 16 битах CSR-регистров mip/mie (uip/uie). Прерывания таймера (mtip/utip), программные прерывания (msip/usip) и входы внешних прерываний (meip/ueip) рассматриваются как дополнительные локальные источники прерываний, где режим привилегированности, уровень прерывания и приоритет могут быть изменены с помощью регистров clicintattr[i] и clicintctl[i]. В режиме CLIC делегирование прерывания для этих сигналов достигается путем изменения режима привилегированности

прерывания в регистре атрибутов прерывания CLIC clicintattr, как и для любого другого входа прерываний CLIC.

27.3 Описание регистров контроллера CLIC

Данные о регистрах контроллера CLIC приведены в таблице 346.

Таблица 346 – Регистры контроллера CLIC

Адрес	Название	Доступ	Описание
0x0D00_0000	CLIC		Контроллер CLIC
Смещение			
0x0000	cliccfg	R/W	Регистр управления
0x0004	clicinfo	R/O	Регистр информации
0x0008 – 0x0FFC	–	–	Зарезервировано
0x1000 + 4 · i*	clicintip[i]	R/W	Регистр наличия прерывания
0x1001 + 4 · i*	clicintie[i]	R/W	Регистр включения прерывания
0x1002 + 4 · i*	clicintattr[i]	R/W	Регистр настройки прерывания
0x1003 + 4 · i*	clicintctl[i]	R/W	Регистр управления уровнем прерывания
* i – номер прерывания от 0 до 46			

Доступ к регистрам cliccfg и clicinfo возможен только из режима привилегированности Machine. При доступе к данным регистрам из режима привилегированности User будет сгенерировано исключение.

Из режима привилегированности Machine возможен доступ ко всем регистрам clicintip[i], clicintie[i], clicintattr[i] и clicintctl[i]. Доступ к регистрам прерывания i из режима привилегированности User осуществляется только в том случае, если соответствующее прерывание i в регистре clicintattr[i], поле mode[1:0], настроено в режим User. Если прерывание i настроено в режим Machine, то из режима привилегированности User чтение регистров прерывания i возвращает нули, запись не осуществляется.

27.3.1 Конфигурационный регистр cliccfg

Единый 8-разрядный регистр глобальной конфигурации cliccfg определяет, сколько поддерживается режимов привилегированности, как регистры clicintctl[i] подразделяются на поля уровня и приоритета, а также поддерживается ли выборочная аппаратная векторизация. Регистр cliccfg имеет три поля:

- 2-битное поле nmbits;
- 4-битное поле nlbits;
- 1-битное поле nvbits.

Таблица 347 – Регистр cliccfg

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

Таблица 348 – Описание бит регистра cliccfg

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	–	Зарезервировано. Рекомендуется не изменять значение поля
6	nmbits[1]	Зарезервировано
5	nmbits[0]	Количество аппаратных бит, реализованных в clicintattr[i].mode для представления режима привилегированности входа i
4…1	nlbits[3:0]	Количество старших бит в регистре clicintctl[i], выделенных для задания уровня прерывания. Важно инициализировать это поле из программного обеспечения
0	nvbits	Наличие функции аппаратной векторизации выборочных прерываний в контроллере CLIC: 1 – функция реализована; 0 – функция не реализована

27.3.1.1 **Определение режима привилегированности прерывания**

Двухразрядное поле cliccfg.nmbits указывает, сколько бит физически реализовано в clicintattr[i].mode для представления режима привилегированности входа i. Хотя поле cliccfg.nmbits всегда имеет ширину в 2 бита, физически реализованных бит в этом поле – один (поддерживаются M/U-режимы привилегированности прерывания).

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

Таблица 349 – Расшифровка уровней привилегированности (mstatus.mpp)

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

Таблица 350 – Режимы привилегированности

Режим привилегированности	nmbits	clicintattr[i].mode	Интерпретация
M/U	0	xx	M-режим прерывание
M/U	1	0x	U-режим прерывание
M/U	1	1x	M-режим прерывание

27.3.1.2 **Определение уровня прерывания**

В регистре clicintctl[i] реализовано 4 старших бита, которые предназначены для настройки уровня и приоритета прерывания. Младшие 4 бита регистра clicintctl[i]

не реализованы и аппаратно установлены в логическую «1». Количество фактически реализованных бит в регистре clicintctl[i] также может быть считано из регистра clicinfo, поле CLICINTCTLBITS.

Четырёхразрядное поле cliccfg.nlbits указывает, сколько старших бит в регистре clicintctl[i] выделено для задания уровня прерывания. Важно инициализировать это поле из программного обеспечения. В текущей реализации данное поле может устанавливаться от 0 до 8. Значение cliccfg.nlbits > CLICINTCTLBITS идентично по реализации cliccfg.nlbits=CLICINTCTLBITS.

В таблице 351 приведено, как кодируются уровни прерывания в зависимости от значения, заданного в cliccfg.nlbits.

Таблица 351 – Режимы привилегированности

Значение в cliccfg.nlbits	Кодирование	Уровни прерываний
0	pppp....	255
1	lppp....	127, 255
2	llpp....	63, 127, 191, 255
3	lllp....	31, 63, 95, 127, 159, 191, 223, 255
4-8	llll....	15, 31, 47, 63, 79, 95, 111, 127, 143, 159, 175, 191, 207, 223, 239, 255
Примечание – Обозначения в графе «Кодирование»: «l» – изменяемые биты, доступные для задания уровня; «p» – изменяемые биты, доступные для задания приоритета; «.» – несуществующие биты для кодирования уровня, предполагается, что установлены в «1»		

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

27.3.1.3 **Определение приоритета прерывания**

Наименее значимые биты в clicintctl[i], которые не настроены как часть уровня прерывания, отвечают за приоритет прерывания. Они используются для определения приоритета среди ожидающих и разрешённых прерываний в том же режиме привилегированности и на том же уровне прерывания. Количество доступных бит приоритета прерывания может быть определено путем вычитания значения cliccfg.nlbits из clicinfo.CLICINTCTLBITS (отрицательное значение бит приоритета, полученное после вычитания, эквивалентно нулю).

Прерывание с наивысшим приоритетом при заданных режиме привилегированности и уровне прерывания выполняется первым. В случае наличия нескольких ожидающих и разрешенных прерываний с одинаковым наивысшим приоритетом, прерывание с наибольшим номером (таблица 380) выполняется первым.

Стоит обратить внимание, что уровень прерывания используется для определения вытеснения (для вложенных прерываний). Напротив, приоритет прерывания не влияет на вытеснение, а используется только в качестве дополнительного ранжирования при наличии нескольких ожидающих прерываний с одинаковым уровнем прерывания.

27.3.1.4 Аппаратная векторизация выборочных прерываний

Функция выборочной аппаратной векторизации предоставляет пользователям гибкость в выборе поведения для каждого прерывания: либо аппаратная векторизация, либо без нее. В результате это позволяет пользователям оптимизировать каждое прерывание и пользоваться преимуществами обоих вариантов поведения. Аппаратная векторизация имеет преимущество в более быстрой реакции на прерывания за счёт небольшого увеличения размера кода (для сохранения/восстановления контекстов). С другой стороны, не векторная имеет преимущество в меньшем размере кода (за счет совместного использования и повторного использования одной копии общего кода для сохранения/восстановления контекстов) за счет более медленного отклика на прерывание.

В регистре cliccfg бит nvbits равен 1, что указывает на наличие функции аппаратной векторизации выборочных прерываний в контроллере CLIC. Бит clicintattr[i].shv управляет поведением векторизации прерывания i. Если clicintattr[i].shv равен 0, то прерывание не является векторным и всегда переходит к общему коду, адрес которого задан в CSR-регистре mtvec (utvec). Если clicintattr[i].shv равен 1, то выполняется переход по адресу функции обработчика прерывания, который аппаратно считывается из таблицы векторов прерываний. Базовый адрес таблицы векторов прерываний устанавливается в CSR-регистре mtvt (utvt). Это позволяет некоторым прерываниям переходить к общему базовому адресу, хранящемуся в mtvec (utvec), в то время как другие обрабатываются аппаратно через таблицу, на которую указывает дополнительный CSR-регистр mtvt (utvt).

27.3.2 Информационный регистр clicinfo

Регистр clicinfo доступен только для чтения и отображает отладочную информацию.

Таблица 352 – Регистр clicinfo

Номер	31	30...25	24…21	20...13	12…0
Доступ	U	R/O	R/O	R/O	R/O
Сброс	0	000000	0100	00000000	00101111
		num_trigger	CLICINTCTLBITS	version	num_interrupt

Таблица 353 – Описание бит регистра clicinfo

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31	-	Зарезервировано
30…25	num_trigger	Максимальное количество триггеров прерывания
24...21	CLICINTCTLBITS	Количество аппаратных бит, реализованных в регистрах clicintctl[i]
20…13	version	Версия реализации CLIC
12…0	num_interrupt	Максимальное количество входных сигналов прерывания

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

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

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

Поле num_trigger указывает максимальное количество триггеров прерывания, поддерживаемых в данной реализации.

27.3.3 **Регистр ожидания прерывания clicintip**

Каждый входной сигнал прерывания имеет выделенный бит ожидания прерывания clicintip[i]. Регистр clicintip занимает один байт в карте памяти. Ожидающий бит расположен в бите 0 байта.

Таблица 354 – Регистр clicintip

Номер	7...1	0
Доступ	U	R/O
Сброс	-	X
		clicintip

Таблица 355 – Описание бит регистра clicintip

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7...1	-	Зарезервировано
0	clicintip	Бит ожидания прерывания: 0 – отсутствие ожидания прерывания; 1 – ожидание прерывания

Программное обеспечение должно рассматривать значение clicintip[i] = 0 как отсутствие ожидания прерывания, а clicintip[i]! = 0 – как ожидание прерывания.

Когда вход сконфигурирован со срабатыванием по уровню, бит clicintip[i] отражает значение входного сигнала в контроллер прерывания после операции условной инверсии, указанной в поле clicintattr[i], а программные записи в бит игнорируются. Программное обеспечение очищает прерывание на устройстве-источнике.

Когда вход сконфигурирован со срабатыванием по фронту, clicintip[i] представляет собой регистр чтения-записи, который может обновляться как с помощью аппаратных прерываний, так и с помощью программного обеспечения. Бит устанавливается аппаратно после того, как на входе прерывания наблюдается фронт соответствующей полярности, как определено полем clicintattr[i]. Соответствующий бит ожидания прерывания очищается аппаратно, когда прерывание обслуживается в векторном режиме. Дополнительные сведения об аппаратной очистке приведены в п. 27.4.1 «Регистры mtvec, utvec». Программные записи могут устанавливать или

очищать срабатывающие по фронту биты ожидания прерывания непосредственно путем записи в регистр, отображенный в память. Срабатывающие по фронту биты ожидания прерывания также могут быть очищены с помощью записи в CSR-регистр mnxti (unxti).

Для повышения производительности, когда выбрано векторное прерывание и обрабатывается, аппаратно очищается соответствующий срабатывающий по фронту бит ожидания, поэтому программному обеспечению не нужно очищать бит ожидания в процедуре обработки.

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

Значение в clicintip[i] не определено при переключении из режима со срабатыванием по уровню в режим со срабатыванием по фронту в clicintattr[i]. Программное обеспечение не может полагаться на сохранённое состояние в битах регистра clicintip[i] режима со срабатыванием по фронту при переходе в режим со срабатыванием по уровню.

27.3.4 **Регистр разрешения прерываний clicintie**

Каждый вход прерывания имеет выделенный бит разрешения прерывания clicintie[i]. Регистр clicintie занимает один байт в карте памяти. Управляющий бит clicintie[0] доступен по чтению-записи и используется для включения/выключения соответствующего прерывания.

Таблица 356 – Регистр clicintie

Номер	7…1	0
Доступ	U	R/W
Сброс	-	0
		clicintie

Таблица 357 – Описание бит регистра clicintie

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7…1	-	Зарезервировано
0	clicintie	Бит разрешения прерывания: 0 – прерывание выключено; 1 – прерывание включено

Программное обеспечение должно рассматривать clicintie[i] = 0 как прерывание выключено, а clicintie[i]! = 0 – как прерывание включено.

Бит clicintie[i] – это индивидуальный бит включения, в то время как mstatus.xie (ustatus.uie) – глобальный бит включения для текущего режима привилегированности. Следовательно, для включения прерывания i в текущем режиме

привилегированности должны быть установлены как clicintie[i], так и mstatus.xie (ustatus.uie).

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

27.3.5 **Регистр атрибутов прерывания clicintattr**

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

Таблица 358 – Регистр clicintattr

Номер	7...6	5…3	2...1	0
Доступ	R/W	U	R/W	R/W
Сброс	11	000	00	X
	mode		trig	shv

Таблица 359 – Описание бит регистра clicintattr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7…6	mode	Режим привилегированности прерывания: 00 – U-режим; 01, 10 – зарезервировано; 11 – M-режим
5...3	-	Зарезервировано
2...1	trig	Определяет тип триггера и полярность для каждого входного сигнала прерывания: 00 – высокий уровень срабатывания прерывания; 01 – положительный фронт срабатывания прерывания; 10 – низкий уровень срабатывания прерывания; 11 – отрицательный фронт срабатывания прерывания
0	shv	Выбор векторного режима прерывания: 0 – прерывание не векторное; 1 – прерывание векторное

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

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

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

Примечания

1 Прерывание по уровню будет взято в обработку, если конвейер подсистемы исполнения процессорного ядра будет свободен. Количество тактов, через которое конвейер подсистемы исполнения будет свободен, зависит от типа команды, которая уже принята к исполнению. Количество тактов:

– для команд DIV, DIVU, REM, REMU: от 2 до 16 тактов (зависит от операндов);
– для команд CLMUL, CLMULH и CLMULR: 8 тактов;
– для команд чтения/записи из TCM региона: 1 такт;
– для команд чтения/записи из IO региона: больше 1 такта (зависит от периферии, в которую идёт обращение (APB, AHB), см. рисунок 9;
– для остальных команд: 1 такт.

2 Для прерываний, которые сбрасывают запрос аппаратно (такие как DMA, EXT_INTx и системный таймер), для предотвращения пропуска прерываний из-за снятия запроса до его взятия в обработку, рекомендуется настраивать формирование по фронту.

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

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

27.3.6 **Регистр управления входом прерывания clicintctl**

Регистр clicintctl[i] – это 8-разрядный регистр управления, доступный по чтению/записи. Он указывает уровень прерывания и приоритет прерывания.

Таблица 360 – Регистр clicintctl

Номер	7..4	3..0
Доступ	R/W	U
Сброс	XXXX	1111
	clicintctl	

Таблица 361 – Описание бит регистра clicintctl

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7…4	clicintctl	Интерпретируются как уровень и приоритет прерывания
3…0	–	Зарезервировано

Количество бит, фактически реализованных в этом регистре, задается фиксированным параметром CLICINTCTLBITS (в clicinfo), который имеет значение от 0 до 8. Реализованные биты сохраняются выровненными по левому краю в старших битах каждого восьмиразрядного регистра clicintctl[i], при этом младшие нереализованные биты равны логической единице. Эти управляющие биты интерпретируются как уровень и приоритет в соответствии с настройками в регистре конфигурации CLIC cliccfg.nlbits.

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

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

27.3.6.1 **Идентификационный номер ввода прерывания**

47-и присваиваются уникальные идентификационные номера (таблица 380), которые используются в CSR-регистре mcause (ucause), поле exccode. Для поддержания обратной совместимости базовые прерывания сохраняют свои первоначальные значения, в то время как новые прерывания нумеруются, начиная с 16.

27.4 **Контрольно-статусные регистры CSR**

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

Список реализованных контрольно-статусных регистров, которые доступны из режима привилегированности Machine, приведен в таблице 362.

Таблица 362 – Контрольно-статусные регистры CSR (Machine)

Адрес	Название	Доступ	Описание
0x305	mtvec	R/W	Адрес общего вектора прерываний
0x307	mtvt	R/W	Адрес таблицы с адресами обработчиков векторных прерываний
0x308	mclicbase	R/O	Базовый адрес контроллера локальных прерываний
0x345	mnxti	R/W	Адрес следующего обработчика прерываний и управление включением прерываний
0x346	mintstatus	R/O	Статус прерывания
0x347	mintthresh	R/W	Управление порогом уровня прерываний
0x348	mscratchcsw	R/W	Scratch регистр с проверкой режима привилегированности
0x349	mscratchcswl	R/W	Scratch регистр с проверкой уровня прерывания

Список реализованных контрольно-статусных регистров, которые доступны из режима привилегированности User, приведен в таблице 363.

Таблица 363 – Контрольно-статусные регистры CSR (User)

Адрес	Название	Доступ	Описание
0x005	utvec	R/W	Адрес общего вектора прерываний
0x007	utvt	R/W	Адрес таблицы с адресами обработчиков векторных прерываний
0x045	unxti	R/W	Адрес следующего обработчика прерываний и управление включением прерываний
0x046	uintstatus	R/O	Статус прерывания
0x047	uintthresh	R/W	Управление порогом уровня прерываний
0x048	uscratchcsw	R/W	Scratch регистр с проверкой режима привилегированности
0x049	uscratchcswl	R/W	Scratch регистр с проверкой уровня привилегированности

27.4.1 **Регистры mtvec, utvec**

32-разрядный регистр, доступный по чтению/записи, содержит базовый адрес общего вектора прерываний, выровненный по границе в 64 байта, а также режим обработки прерываний.

Таблица 364 – Регистр mtvec, utvec

Номер	31...6	5...2	1...0
Доступ	R/W	U	R
Сброс	0	0000	11
	BASE		MODE

Таблица 365 – Описание бит регистров mtvec, utvec

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...6	BASE	Базовый адрес общего вектора прерываний
5…2	-	Зарезервировано
1…0	MODE	Режим обработки прерываний: 00 – невекторный режим обработки прерываний; 01 – векторный режим обработки прерываний; 10 – зарезервировано; 11 – режим обработки прерываний с использованием CLIC

Реализован только режим обработки прерываний с использованием CLIC. В данном режиме к общему вектору прерываний переходят все исключения, а также все прерывания, настроенные в невекторный режим работы (clicintattr[i].shv = 0).

27.4.2 **Регистры mtvt, utvt**

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

Таблица 366 – Регистр mtvt, utvt

Номер	31...6	5...0
Доступ	R/W	U
Сброс	X	000000
	mtvt, utvt	

Таблица 367 – Описание бит регистров mtvt, utvt

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...6	mtvt, utvt	Базовый адрес таблицы векторов прерываний
5…0	-	Зарезервировано

27.4.3 **Регистры mnxti, unxti**

Таблица 368 – Регистр mnxti, unxti

Номер	31…0
Доступ	R/W
Сброс	0
	mnxti, unxti

Таблица 369 – Описание бит регистров mnxti,unxti

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	mnxti,unxti	Указатель на запись в таблице обработчиков прерываний

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

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

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

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

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

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

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

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

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

27.4.4 **Регистры mintstatus, uintstatus**

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

Таблица 370 – Регистр mintstatus

Номер	31...24	23...8	7...0
Доступ	R/O	U	R/O
Сброс	0	-	0
	mil		uil

Таблица 371 – Описание бит регистра mintstatus

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...24	mil	Уровень активного прерывания для уровня привилегированности Machine
23...8	-	Зарезервировано
7...0	uil	Уровень активного прерывания для уровня привилегированности User

Регистр uintstatus предоставляет ограниченный доступ к mintstatus.

Таблица 372 – Регистр uintstatus

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

Таблица 373 – Описание бит регистра uintstatus

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…8	-	Зарезервировано
7…0	uil	Уровень активного прерывания для уровня привилегированности User

27.4.5 **Регистры mintthresh, uintthresh**

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

Таблица 374 – Регистр mintthresh, uintthresh

Номер	31...8	7...0
Доступ	U	R/W
Сброс	0	0
		th

Таблица 375 – Описание бит регистров mintthresh, uintthresh

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…8	-	Зарезервировано
7...0	th	Уровень порога прерывания

Обычно уровень порога прерывания используется для реализации критических разделов. Текущий обработчик может временно повысить свой эффективный уровень

прерывания, чтобы реализовать критическую секцию среди подмножества уровней, в то же время позволяя более высоким уровням прерывания выполнять вытеснение текущего прерывания.

Эффективный уровень прерывания вычисляется как:

effective_level = max(mintstatus.xil, mintthresh.th),

effective_level = max(uintstatus.xil, uintthresh.th).

Значение max используется для предотвращения падения ниже его первоначального уровня, что привело бы к нарушению прав, а также упростило программному обеспечению удаление порога, не зная его собственного уровня, просто записав ноль.

Пороговое значение уровня прерывания допустимо только при запуске в режиме связанной привилегированности, но не в других режимах. Это связано с тем, что прерывания для режимов с более низкими привилегиями всегда отключены, в то время как прерывания для режимов с более высокими привилегиями всегда включены. Например, прерывания в машинном режиме не будут маскироваться установкой порога в пользовательском режиме. Это аналогично тому, как работает mstatus.mie – не маскирует прерывания машинного режима при запуске в режимах с более низкими привилегиями.

Такое поведение значительно упрощает аппаратную реализацию, поскольку требуется выбрать только одно глобальное максимальное прерывание и сравнить с пороговым значением соответствующего режима привилегированности (игнорируя пороговые значения в других режимах). В противном случае аппаратному обеспечению пришлось бы выбирать несколько максимальных прерываний (по одному на режим привилегированности), сравнить и сопоставить с соответствующими пороговыми значениями, а затем выбрать максимальное прерывание с режимом наивысшей привилегированности.

27.4.6 **Регистр mclicbase**

Регистр mclicbase – 32-разрядный регистр, доступный только для чтения, предоставляющий базовый адрес регистров, отображенных в память CLIC. Его значение должно быть сконфигурировано или настроено на аппаратном уровне платформы для указания начального адреса регистров, отображенных в память CLIC.

Таблица 376 – Регистр mclicbase

Номер	31…0
Доступ	R/O
Сброс	0xD000000
	mclicbase

Таблица 377 – Описание бит регистра mclicbase

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	mclicbase	Базовый адрес регистров CLIC, отображенных в память

Поскольку карта памяти CLIC должна быть выровнена по границе 4 Кбайт, в регистре mclicbase двенадцать младших разрядов доопределены в логический «0». Регистр используется программным обеспечением для определения местоположения отображаемых в памяти регистров CLIC.

27.4.7 **Регистры mscratchcswl, uscratchcswl**

Регистр добавлен для быстрого обмена указателем стека между выполняемым кодом прерывания и не прерывания в одинаковом режиме привилегированности.

Таблица 378 – Регистр mscratchcswl, uscratchcswl

Номер	31...0
Доступ	R/W
Сброс	X
	mscratchcswl, uscratchcswl

Таблица 379 – Описание бит регистров mscratchcswl, uscratchcswl

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	mscratchcswl, uscratchcswl	Значение указателя стека

Пример:

```
csrrw rd, mscratchcswl, rs1
// Pseudocode operation.
if ( (mcause.pil==0) != (mintstatus.mil==0) ) then {
        t = rs1;
        rd = mscratch;
        mscratch = t;
} else {
        rd = rs1; // mscratch unchanged.
}
// Usual use: csrrw sp, mscratchcswl, sp
```

Регистр mscratchcswl (uscratchcswl) работает аналогично регистру mscratchcsw (uscratchcsw), за исключением того, что условие обмена истинно, когда прерыватель и прерванный не являются одновременно прикладными задачами или оба не являются обработчиками прерываний.

Как и для mscratchcsw (uscratchcsw) для mscratchcswl (uscratchcswl) определена только операция чтения-изменения-записи (swap/CSRRW).

Реализация вариантов без CSRRW (когда rd или rs1 – это x0, либо rs1 является непосредственным операндом) в mscratchcswl (uscratchcswl) не определена/зарезервирована.

27.5 Карта локальных прерываний

Таблица 380 – Прерывания, формируемые периферийными блоками

Номер	Прерывания	Блок	Принцип формирования
0	usip	Ядро	Программное прерывание для User режима привилегированности
1	-	-	Зарезервировано
2	-	-	Зарезервировано
3	msip	Ядро	Программное прерывание для Machine режима привилегированности
4	utip	Системный таймер	Прерывание таймера для User режима привилегированности
5	-	-	Зарезервировано
6	-	-	Зарезервировано
7	mtip	Системный таймер	Прерывание таймера для Machine режима привилегированности
8	-	-	Зарезервировано
9	-	-	Зарезервировано
10	-	-	Зарезервировано
11	-	-	Зарезервировано
12	csip	Ядро	Программное прерывание CLIC
13	-	-	Зарезервировано
14	-	-	Зарезервировано
15	-	-	Зарезервировано
16	int_local0	DMA	Прерывания от DMA DMA_ERR или DMA_DONE. Обработка прерываний от DMA в соответствии с подпунктом 25.4.5.7 «Индикация ошибок»
17	int_local1	UART1	Сигнал UARTINTR
18	int_local2	UART2	Сигнал UARTINTR
19	int_local3	SSP1	Сигнал SSPINTR
20	int_local4	POWER	Сигнал прерывания от POWER Detector
21	int_local5	WWDG	Сигнал прерывания от WWDG
22	int_local6	Timer1	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
23	int_local7	Timer2	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
24	int_local8	ADC	Сигналы прерываний от АЦП EOCIF_1 или AWOIF_1 или EOCIF_2 или AWOIF_2
25	int_local9	I2C	Сигнал I2C_INT
26	int_local10	BACKUP	Прерывание от BKP и часов реального времени
27	int_local11	Внешнее прерывание 1	Сигнал EXT_INT1 Выводы PA[10] и PD[15] в основном режиме
28	int_local12	Внешнее прерывание 2	Сигнал EXT_INT2 Выводы PB[6] в альтернативном режиме и PC[4] в основном режиме
29	int_local13	Внешнее прерывание 3	Сигнал EXT_INT3 Выводы PB[7] в альтернативном режиме и PC[5] в основном режиме

Номер	Прерывания	Блок	Принцип формирования
30	int_local14	ADCUI	Прерывание от блока F0 сигма-дельта АЦП для измерения напряжений и токов
31	int_local15	ADCUI	Прерывание от блока F1 сигма-дельта АЦП для измерения напряжений и токов
32	int_local16	ADCUI	Прерывание от блока F2 сигма-дельта АЦП для измерения напряжений и токов
33	int_local17	-	Программное прерывание
34	int_local18	-	Программное прерывание
35	int_local19	-	Программное прерывание
36	int_local20	-	Программное прерывание
37	int_local21	RANDOM	Прерывание блока генератора случайных чисел
38	int_local22	ISO7816	Прерывание блока UART ISO7816
39	int_local23	UART3	Сигнал UARTINTR
40	int_local24	SSP2	Сигнал SSPINTR
41	int_local25	SSP3	Сигнал SSPINTR
42	int_local26	Timer3	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
43	int_local27	Timer4	Сигнал прерывания от Таймера TIM_STATUS и TIM_IE
44	int_local28	UART4	Сигнал UARTINTR
45	int_local29	-	Программное прерывание
46	int_local30	-	Программное прерывание

28 Подсистема отладки

Подсистема отладки ядра соответствует спецификации RISC-V External Debug Support Version 0.13. Поддерживаются следующие опции:

– чтение/запись регистров общего назначения с помощью абстрактных команд;
– Read/Write программный буфер 64 байта;
– четыре регистра абстрактных данных;
– память, доступная только по чтению (Debug ROM).

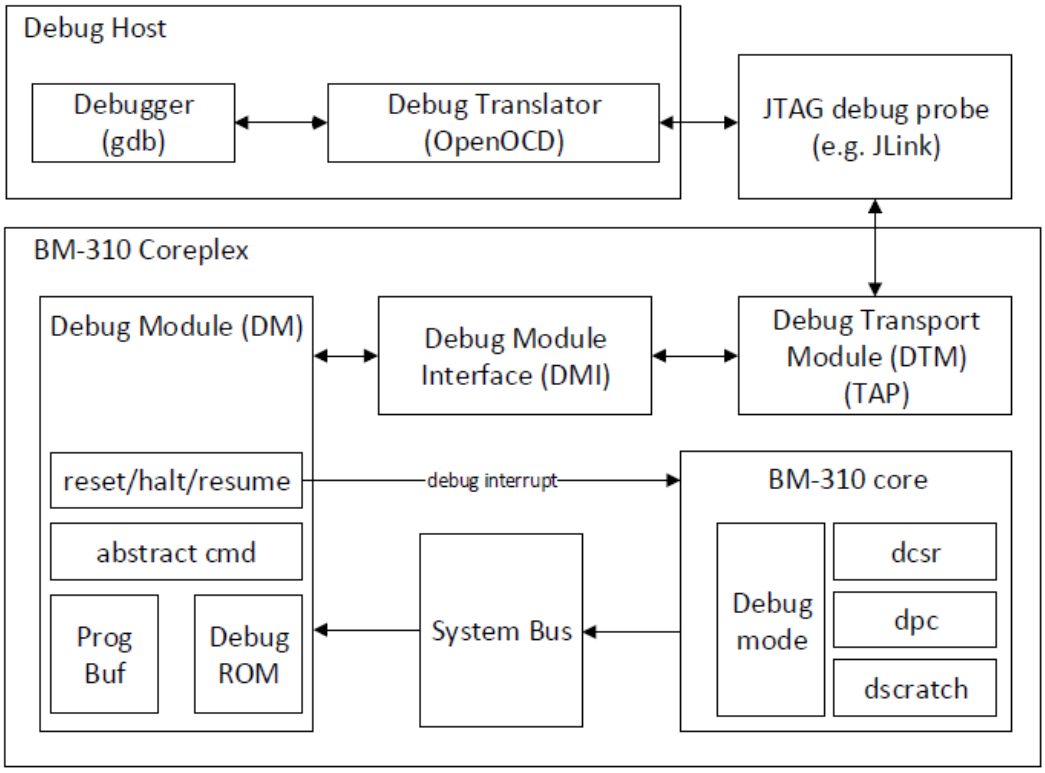


Рисунок 98 – Архитектура системы отладки

Подсистема отладки состоит из блока DTM (Debug Transport Module) и блока DM (Debug Module). DTM связан с DM через блок DMI (Debug Module Interface).

28.1 Блок DTM (Debug Transport Module)

DTM содержит TAP (Test Access Port), соответствующий спецификации интерфейса JTAG. Размер IR – 5 бит. Доступ к DM через JTAG осуществляется через регистр DMI. Список доступных по JTAG регистров приведен в таблице ниже. Общая схема цепочки отладки процессорного ядра представлена на рисунке 99. Переход из области синхросигнала JTAG в область синхросигнала ядра происходит в блоке DMI.

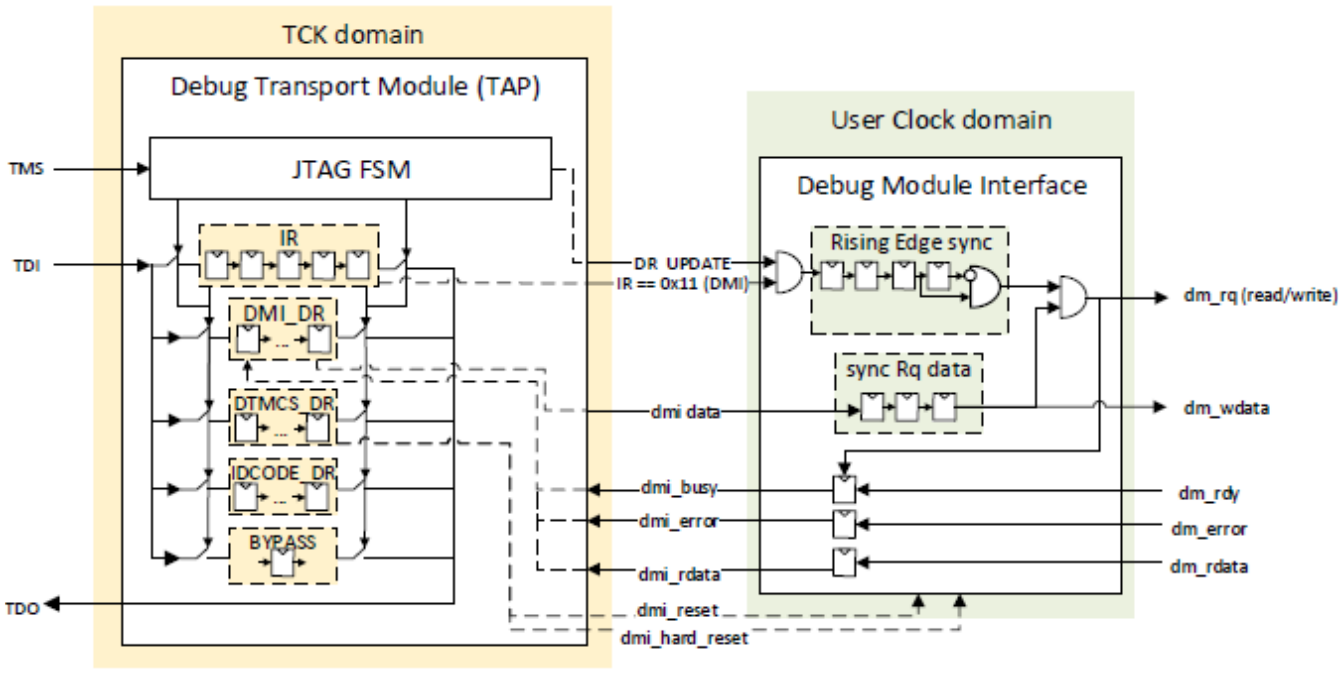


Рисунок 99 – Интерфейс DM

28.1.1 **Регистры DTM**

Таблица 381 – Регистры DTM

Адрес	Название	Доступ
0x00	BYPASS	R
0x01	Регистр идентификатора производителя микросхемы (IDCODE)	R
0x10	Регистр статуса и управления DTM (DTMCS)	R/W
0x11	Регистр доступа к DMI	R/W
0x12-0x1A	BYPASS	R/W

Таблица 382 – Регистр статуса и управления DTM (DTMCS)

Биты	Название	Доступ	Описание	Значение
17	dmihardreset	W1	Запись 1 сбрасывает DMI	
16	dmireset	W1	Запись 1 сбрасывает ошибку в DMI	
14…12	idle	R	Количество циклов, которое отладчик должен провести в состоянии IDLE, между запросом и чтением результата из DMI, для того, чтобы избежать состояния busy	5
11, 10	dmistat	R	Состояние DMI: 0 – ошибок нет; 1 – см. 2; 2 – ошибка выполнения операции; 3 – операция ещё не выполнена (состояние busy)	
9…4	abits	R	Размер поля адреса регистра DMI	7
3…0	version	R	Версия реализованной спецификации: 0 – 0.11; 1 – 0.13	1

Таблица 383 – Регистр доступа к DMI

Биты	Название	Доступ	Описание	Значение
abits+33…34	address	R/W	Адрес запроса на DMI	
33…2	data	R/W	Данные для записи, отправляемые в DM во время состояния Update-DR, либо данные, считанные из DM	
1…0	op	R/W	Код операции. При записи: 0 – пустой запрос. В DM ничего не посылается; 1 – чтение по адресу; 2 – запись по адресу. При чтении: 0 – предыдущий запрос завершен успешно; 2 – предыдущий запрос вернул ошибку; 3 – предыдущий запрос ещё не выполнен. Чтобы сбросить это состояние необходим сброс DMI	

Взаимодействие отладчика с DM происходит через DMI путем чтения и записи регистров DM. Для этого используется регистр DMI в DTM. Значение поля abits равно 7. Список доступных по DMI регистров приведен в таблице 384.

Таблица 384 – Регистры отладчика DM

Адрес	Название	Доступ
0x04-0x07	Abstract Data 0-3	R/W
0x10	DM control	R/W
0x11	DM status	R/W
0x12	Hart info	R/W
0x16	Abstract Control and Status	R/W
0x17	Abstract Command	R/W
0x18	Abstract Command Autoexec	R/W
0x20-0x2F	Program Buffer 0-15	R/W

28.2 **Блок DM (Debug Module)**

Взаимодействие процессорного ядра с DM происходит с помощью чтения/записи отображенных в памяти регистров (DM MMRs (Memory Mapped Registers)). Карта памяти DM представлена в таблице 385.

Таблица 385 – Карта памяти DM

Адрес	Название	Доступ
0x0000_0000	Safe zero	R/W
0x0000_0100	Halted ack	W
0x0000_0104	Going ack	W
0x0000_0108	Resume ack	W
0x0000_010C	Exception	W
0x0000_0300	Where to	R
0x0000_0338	Abstract command 0	R
0x0000_033C	Abstract command 1	R
0x0000_0340-0x37C	Program buf	R/W
0x0000_0380-0x38C	Abstract data	R/W
0x0000_0400	CPU flags	R
0x0000_0800-0xFFF	Debug ROM	R

Примечание – Обращения к неиспользуемым адресам запрещены, могут привести к непредсказуемому поведению.

Таблица 386 – CPU flags

Биты	Название	Доступ	Описание	Значение
0	Going	R	Запрос на выполнение абстрактной команды. Находясь в режиме отладки, ядро считывает данное поле и делает переход по адресу первой абстрактной команды (0x338)	
1	Resumereq	R	Запрос на выход из режима отладки. Находясь в режиме отладки, ядро считывает данное поле и выходит из режима отладки, если его значение равно 1. Исполнение программы продолжается со значения PC, сохраненного в регистре dpc	

28.2.1 **Регистры DM**

Для взаимодействия с внешним отладчиком внутри процессорного ядра реализован специальный режим исполнения: режим отладки. В таблице 387 представлен список контрольных регистров режима отладки. Процессор имеет право читать и писать контрольные регистры отладки только в режиме отладки. В любом другом режиме попытка доступа к данным регистрам вызовет исключение. Исполнение в режиме отладки имеет следующие свойства:

– операции обращения к контрольным регистрам и к памяти происходят также как в режиме привилегированности M;

– любые прерывания не обрабатываются;

– исключения не обновляют значения контрольных регистров;

– все инструкции смены уровня привилегированности не обрабатываются. Исключение – инструкция *ebreak*. При исполнении инструкции *ebreak* в режиме отладки процессор выполняет переход на начало Debug ROM, без обновления dpc и dcsr.

Таблица 387 – Контрольные регистры режима отладки

Адрес	Название	Доступ
0x7B0	Регистр управления режимом отладки dcsr	R/W
0x7B1	Регистр счетчика команд dpc	R/W
0x7B2	Дополнительный регистр режима отладки dscratch	R/W

Таблица 388 – Регистр управления режимом отладки dcsr

Биты	Название	Доступ	Описание	Значение
31…28	version	R/W	Версия: 0 – Режим отладки не поддерживается 4 – Режим отладки поддерживается и соответствует спецификации версии 0.13	4
15	ebreakm	R/W	Значение 1 – при исполнении инструкции ebreak в режиме M процессор переходит в режим отладки	0
12	ebreaku	R/W	Значение 1 – при исполнении инструкции ebreak в режиме U процессор переходит в режим отладки	0
8…6	cause	R	Причина перехода в режим отладки: 1 – Выполнена инструкция ebreak 3 – Прерывание от DM 4 – Исполнена очередная инструкция в режиме пошагового исполнения	0
2	step	R/W	Значение 1 – режим пошагового исполнения включен. В данном режиме после выхода из режима отладки процессор исполняет очередную инструкцию пользовательского приложения и снова переключается в режим отладки. При пошаговом исполнении прерывания выключены	0
1, 0	prv	R/W	Режим привилегированности, в котором находилось ядро перед переключением в режим отладки. Отладчик может изменить значение этого поля, чтобы изменить режим процессора после выхода из режима отладки	0

Таблица 389 – Регистр счетчика команд dpc

Биты	Название	Доступ	Описание	Значение
31…0	dpc	R/W	Сохраненное значение счетчика команд. При входе в режим отладки в данный регистр сохраняется текущее значение счетчика команд. Запись в данный регистр возможна только в режиме отладки. При выходе из режима отладки исполнение продолжается с адреса, сохраненного в данном регистре	

Таблица 390 – Дополнительный регистр режима отладки dscratch

Биты	Название	Доступ	Описание	Значение
31…0	dscratch	R/W	Дополнительный регистр, используемый DM для сохранения промежуточной информации	

После входа в режим отладки процессорное ядро выполняет переход по адресу начала Debug ROM (адрес 0x800). Программный код, сохраненный в Debug ROM, осуществляет опрос флагов DM. При определении запроса на исполнение абстрактной команды (флаг “Going” в регистре 0x500) происходит переход исполнения на адрес первой абстрактной команды (регистр по адресу 0x338). В случае возникновения исключения во время исполнения абстрактной команды или кода из программного буфера ядро переходит по адресу внутри Debug ROM, где сохранен код обработчика исключений. В данном обработчике находится инструкция записи по адресу 0x10c. После чего отладчик может считать значение регистра Abstract Control and Status и определить факт возникновения исключения в процессе выполнения абстрактной команды. Если при опросе считывается запрос на продолжение исполнения пользовательского приложения (флаг “Resumereq”), то исполняется код, осуществляющий переход по адресу, сохраненному в регистре dpc.

Любые операции отладчика: считывание/запись регистров, считывание/запись памяти, установка точек остановки в коде, последовательное исполнение – осуществляются с помощью модификации кода в программном буфере и исполнения абстрактных команд.

28.3 Подключение адаптера JTAG к микросхеме

Если отладочный интерфейс не заблокирован, к микросхеме может быть подключен адаптер JTAG, с помощью которого программные средства разработки позволяют работать с микросхемой в отладочном режиме. Линии JTAG должны быть подтянуты к питанию резисторами сопротивлением от 4,7 до 10 кОм с учетом того, чтобы эти подтяжки не влияли на работу системы (см рисунок 100).

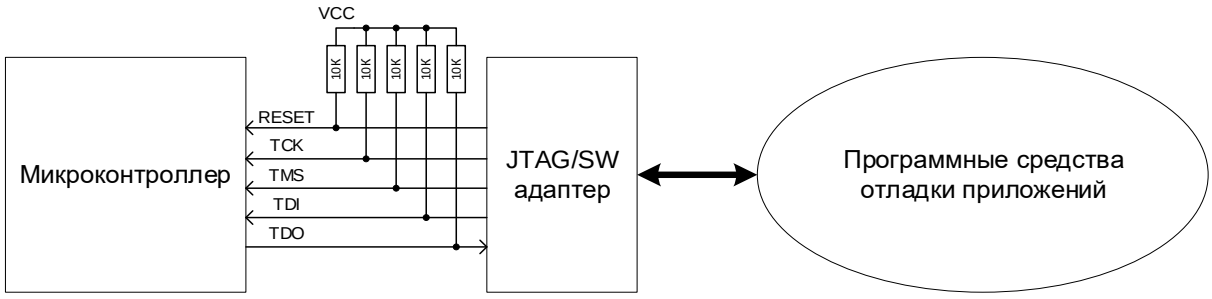


Рисунок 100 – Схема работы в режиме отладки

Отладочный интерфейс JTAG расположен на выводах порта A (см. таблицу 391). В отладочном режиме следует учитывать информацию, приведенную в разделе 18 «Порты ввода-вывода».

Таблица 391 – Переопределение выводов интерфейса JTAG

Вывод JTAG	Вывод микросхемы	Описание
TCK	PA6/TMR1_CH4/TCK	В качестве выводов интерфейса JTAG используются выводы порта A, совмещенные с выводами Таймера 1. Запрещено использование совмещенных с JTAG выводов Таймера 1 при разрешенной работе порта JTAG
TDO	PA7/TMR1_CH4N/TDO	
TMS	PA8/TMR1_ETR/TMS	
TDI	PA9/TMR1_BRK/TDI	

29 Сторожевые таймеры

29.1 Регистры блока сторожевых таймеров

Таблица 392 – Описание регистров блока сторожевого таймера IWDG

Базовый Адрес	Название	Описание
0x5005_0000	IWDG	Сторожевой таймер IWDG
Смещение		
0x00	IWDG_KR[15:0]	Регистр ключа
0x04	IWDG_PR[2:0]	Делитель частоты сторожевого таймера
0x08	IWDG_RLR[11:0]	Регистр основания счета сторожевого таймера
0x0C	IWDG_SR[1:0]	Регистр статуса сторожевого таймера

Таблица 393 – Описание регистров блока сторожевого таймера WWDG

Базовый Адрес	Название	Описание
0x5004_8000	WWDG	Сторожевой таймер WWDG
Смещение		
0x00	WWDG_CR[7:0]	Регистр управления
0x04	WWDG_CFR[9:0]	Регистр конфигурации
0x08	WWDG_SR[0]	Регистр статуса

29.1.1 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 0xAAAA 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 101

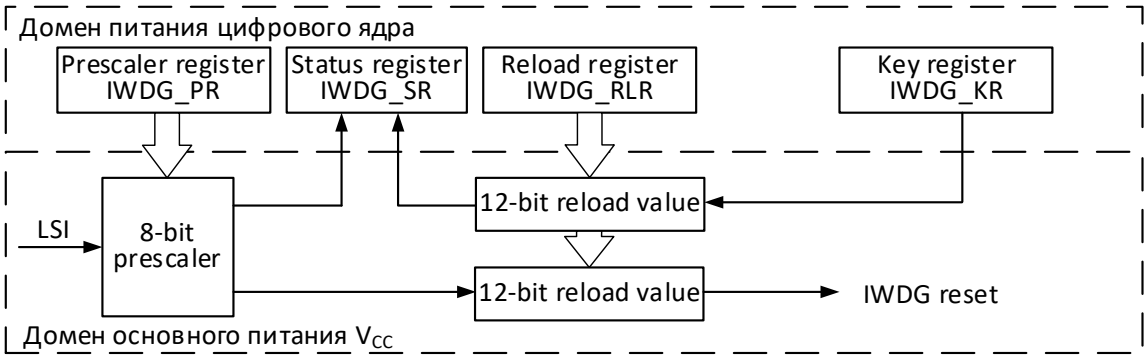


Рисунок 101 – Блок-схема независимого сторожевого таймера

Структурно IWDG состоит из двух частей. Часть с регистрами управления расположена в домене питания цифрового ядра, доступна по APB, тактируется частотой PCLK и сбрасывается по всем типам сброса. Часть непосредственно с независимым сторожевым таймером расположена в домене основного питания микросхемы V_{CC} и тактируется от генератора LSI – поэтому для фактической записи в регистры IWDG в домене основного питания V_{CC} генератор LSI должен находиться в рабочем режиме. Часть сторожевого таймера в домене основного питания V_{CC} 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 0xAAAA в регистр ключа. Для возможности записи в регистры IWDG_PR и IWDG_RLR предварительно в регистр ключа должно быть записано ключевое значение 0x5555 для разрешения доступа по записи в них. При записи в регистр IWDG_PR значения предделителя и ключа 0xAAAA в регистр IWDG_KR формируются запросы на обновление соответствующих параметров (предделителя и основания счета), которые передаются IWDG в домене основного питания. Запросы на обновления не захватываются (не сохраняются) частью в домене основного питания, то есть при любом типе сброса запросы на обновление от IWDG в домене питания ядра сбрасываются. Наличие запроса обновления от IWDG в домене питания ядра и/или информация о процессе фактического обновления на стороне IWDG в домене основного питания транслируются флагами PVU (для предделителя) и RVU (для основания счета). Фактическое обновление значений (со сбросом соответствующих флагов после обновления) осуществляется только при разрешенной работе IWDG и наличии частоты LSI. При фактическом обновлении предделителя и основания счета недоступна запись в регистры IWDG_PR и IWDG_RLR соответственно. После фактического обновления снимаются запросы обновления со стороны IWDG в домене питания ядра V_{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 0x5555 в регистр IWDG_KR.

3 Запись значения предделителя в регистр IWDG_PR.

4 Разрешение работы IWDG записью значения 0xCCCC в регистр IWDG_KR.

5 Ожидание обновления значения предделителя (сброса флага PVU в 0).

6 Разрешение доступа к регистрам IWDG_PR и IWDG_RLR записью значения 0x5555 в регистр IWDG_KR.

7 Проверка флага RVU = 0 перед записью в IWDG_RLR.

8 Запись значения перезагрузки в регистр IWDG_RLR.

9 Периодическая перезагрузка сторожевого таймера в процессе выполнения программы значением регистра IWDG_RLR с помощью записи ключевого значения 0xAAAA в регистр IWDG_KR.

29.1.1.1 **IWDG_KR**

Таблица 394 – Регистр IWDG_KR

Номер	31…16	15…0
Доступ	U	W
Сброс	0	0
	-	KEY[15:0]

Таблица 395 – Описание бит регистра IWDG_KR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…16	–	Зарезервировано
15...0	KEY[15:0]	Значение ключа (только запись, читается 0x0000). При записи: – 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)

29.1.1.2 **IWDG_PR**

Таблица 396 – Регистр IWDG_PR

Номер	31…3	2…0
Доступ	U	R/W
Сброс	0	0
	-	PR[2:0]

Таблица 397 – Описание бит регистра IWDG_PR

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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
2...0	PR[2:0]	Делитель частоты сторожевого таймера. 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 Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR. При записи в регистр формируется запрос на обновление значения делителя частоты в часть IWDG в домене основного питания (см. рисунок 101). Запись в регистр IWDG_PR недоступна при фактической записи предделителя в часть IWDG в домене основного питания V_{CC} (см. рисунок 101). При чтении возвращает записанное в регистр значение, а не значение предделителя IWDG в домене основного питания

29.1.1.3 **IWDG_RLR**

Таблица 398 – Регистр IWDG_RLR

Номер	31…12	11…0
Доступ	U	R/W
Сброс	0	0xFFF
	-	RLR[11:0]

Таблица 399 – Описание бит регистра IWDG_RLR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...12	–	Зарезервировано
11...0	RLR[11:0]	Значение перезагрузки (основания счета) сторожевого таймера. Сторожевой таймер декрементируется, начиная с этого значения. Определяет значение, загружаемое в сторожевой таймер при записи значения 0xAAAA в регистр IWDG_KR. Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR. Запись в регистр IWDG_RLR недоступна при фактической записи основания счета в часть IWDG в домене основного питания V_{CC} (см. рисунок 101). При чтении возвращает записанное в регистр значение, а не значение перезагрузки IWDG в домене основного питания

29.1.1.4 **IWDG_SR**

Таблица 400 – Регистр IWDG_SR

Номер	31…2	1	0
Доступ	U	R	R
Сброс	0	0	0
	-	RVU	PVU

Таблица 401 – Описание бит регистра IWDG_SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…2	–	Зарезервировано
1	RVU	Флаг обновления значения сторожевого таймера. Устанавливается аппаратно и служит признаком того, что обновляется значение сторожевого таймера IWDG в домене основного питания V_{CC} из регистра перезагрузки IWDG_RLR (см. Рисунок 101). **Работа IWDG разрешена и LSI работает:** Отображает запрос от регистров управления на обновление значения сторожевого таймера значением регистра IWDG_RLR и/или фактическое обновление значения сторожевого таймера в домене основного питания V_{CC}. Сбрасывается при завершении фактического обновления значения перезагрузки. Значение регистра перезагрузки IWDG_RLR может быть обновлено только, если этот бит равен нулю. **Работа IWDG не разрешена или LSI не работает:** Отображает запрос от регистров управления на обновление значения сторожевого таймера в домене основного питания V_{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
0	PVU	Флаг обновления делителя частоты сторожевого таймера. Устанавливается аппаратно и служит признаком того, что обновляется значение делителя частоты IWDG в домене основного питания V_{CC} из регистра предделителя IWDG_PR (см. Рисунок 101). **Работа IWDG разрешена и LSI работает:** Отображает запрос от регистров управления на обновление делителя частоты сторожевого таймера значением регистра IWDG_PR и/или фактическое обновление значения делителя в домене основного питания V_{CC}. Сбрасывается при завершении фактического обновления делителя. Значение регистра предделителя IWDG_PR может быть обновлено только, если этот бит равен нулю. **Работа IWDG не разрешена или LSI не работает:** Отображает запрос от регистров управления на обновление значения сторожевого таймера в домене основного питания V_{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

29.1.2 **Регистры сторожевого таймера WWDG**

WWDG в данном контроллере – это оконный сторожевой таймер, счётчик которого считает сверху вниз, начиная со значения, которое мы вносим в регистр WWDG_CR (биты T[6:0]), но не до нуля, а до 63 (0x3F). Только кроме значения, с которого считает WWDG, и значения 0x63, до которого считает данный сторожевой таймер, существует ещё одно промежуточное значение, находящееся между данными величинами. Оно также является задаваемой величиной и, таким образом, образуется окно между данной величиной и значением 0x3F, поэтому данный сторожевой таймер и является оконным. Подобно IWDG, сторожевой таймер WWDG также, досчитав до 0x3F, даст команду на перезагрузку системы, если же мы, конечно, не перезагрузим данный оконный сторожевой таймер своевременно. Только перезагрузить мы должны его не только не позже, чем его счётчик досчитает вниз до 0x3F, но ещё и не раньше, чем он досчитает до промежуточной величины, которая также заносится в регистр WWDG_CFR (биты W[6:0]). Данная величина поэтому должна находиться в интервале от числа 0x3F до заданного значения счётчика, с которого он начинает обратный отсчёт. Таким образом, мы получаем возможность проверки определённых временных интервалов хода нашей программы.

Блок-схема оконного сторожевого таймера приведена на рисунке 102.

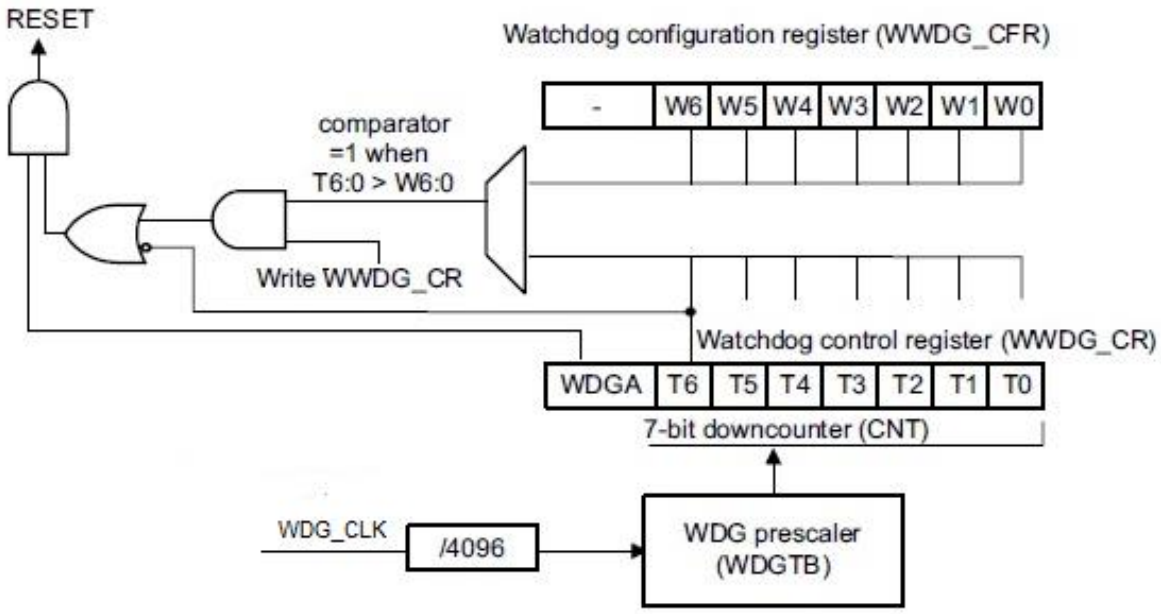


Рисунок 102 – Блок-схема оконного сторожевого таймера

Тактовые сигналы счётчика идут через первый предделитель с постоянным коэффициентом деления 4096 с шины WDG_CLK на второй, регулируемый предделитель, значение коэффициента деления которого мы можем устанавливать с помощью битов регистра WGTB. После второго предделителя тактовые сигналы попадают на счётчик обратного отсчёта.

Число, которое находится в битах T6:T0 регистра WWDG_CR после старта оконного таймера начинает инкрементироваться, как только оно достигнет значения 0x3F, то есть как только бит T6 очистится, на один из мультиплексоров пойдёт

сигнал о перезагрузке системы. Но на то он и мультиплексор, что сигнал о перезагрузке он может получить не только от очистки бита T6. Аналогичный сигнал он может получить от компаратора, в котором сравниваются значения битов T6:T0 регистра WWDG_CR со значениями битов W6:W0 регистра WWDG_CFR. И, если первое число больше второго, то есть если счётчик обратного отсчёта ещё не досчитал до значения первой границы, а в это время мы пытались перезагрузить оконный сторожевой таймер, то мультиплексор и получит данный сигнал, который также заставит систему перезагрузиться.

Также процесс работы оконного сторожевого таймера показан на графике на рисунке 103.

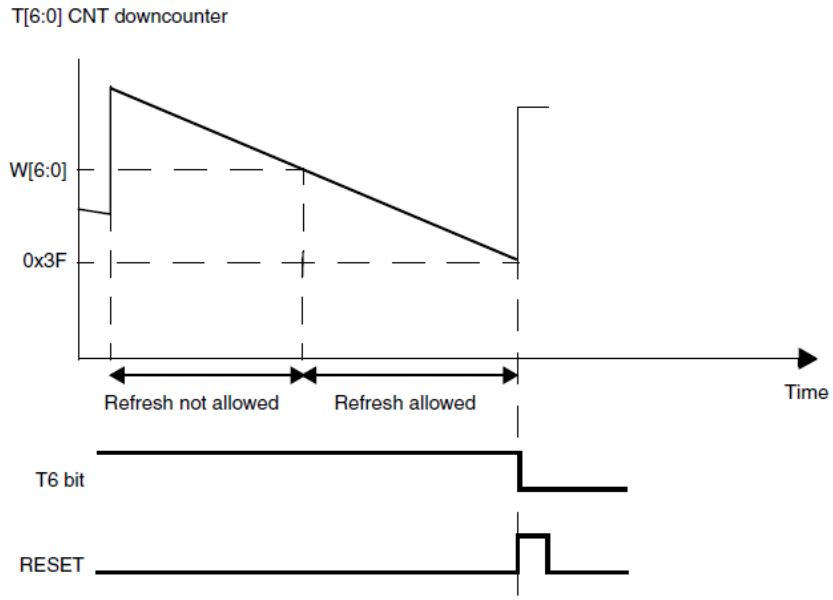


Рисунок 103 – Временная диаграмма оконного сторожевого таймера

Можно рассчитать время работы WWDG от старта до окончания счёта и перезагрузки по следующей формуле

$$t_{WWDG} = t_{WDG_CLK} \cdot 4096 \cdot 2^{WGTB} \cdot (T[5:0] + 1)\text{, мс} \quad (21)$$

где t_{WWDG} – таймаут;

t_{WDG_CLK} – период синхросигнала APB шины в мс.

29.1.2.1 **WWDG_CR**

Таблица 402 – Регистр WWDG_CR

Номер	31…8	7	6…0
Доступ	U	R/S	R/W
Сброс		0	1
	-	WDGA	T6…T0

Таблица 403 – Описание бит регистра WWDG_CR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...18		Зарезервировано
7	WDGA	Бит активации. Этот бит устанавливается программно и очищается только аппаратно при сбросе. Когда WDGA=1, сторожевой таймер может генерировать сброс. 0 – сторожевой таймер отключен 1 - сторожевой таймер включен
6…0	T[6:0]	Значение семиразрядного счётчика (от старших разрядов к младшим) Эти биты содержат значение сторожевого таймера, который декрементируется каждые $4096x2^{WDGTB}$ циклов частоты WDG_CLK периферийной шины APB

29.1.2.2 **WWDG_CFR**

Таблица 404 – Регистр WWDG_CFR

Номер	31…10	9	8	7	6…0
Доступ	U	R/S	R/W	R/W	R/W
Сброс		0	0	0	1
	-	EWI	WDGTB1	WDGTB0	W6…W0

Таблица 405 – Описание бит регистра WWDG_CFR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...10		Зарезервировано
9	EWI	Раннее предупреждающее прерывание. Если бит установлен, то разрешается генерация прерывания при достижении сторожевым таймером значении 40h. Прерывание запрещается только аппаратным сбросом
8, 7	WGTB[1:0]	Делитель частоты сторожевого таймера: 00 – частота таймера (WDG_CLK / 4096) /1; 01 – частота таймера (WDG_CLK / 4096) /2; 10 – частота таймера (WDG_CLK / 4096) /4; 11 – частота таймера (WDG_CLK / 4096) /8
6..0	W[6:0]	Значение окна. Эти биты содержат значение окна, в пределах которого возможна инициализация битов T[6:0] значением в пределах 40h-7Fh. Если происходит инициализация битов в момент T>W, то формируется сброс на выходе RESET. Если таймер достигнет значения T=3Fh, то также формируется сброс

29.1.2.3 **WWDG_SR**

Таблица 406 – Регистр WWDG_SR

Номер	31…2	1	0
Доступ	U	R	R/C
Сброс		1	0
	-	WDG_WEC	EWIF

Таблица 407 – Описание бит регистра WWDG_SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...2		Зарезервировано
1	WDG_WEC	Флаг окончания записи в регистры WWDG_CR и WWDG_CFR: 1 – запись завершена; 0 – запись не завершена
0	EWIF	Флаг раннего предупреждающего прерывания. Этот бит устанавливается аппаратно, когда сторожевой таймер достигает значения 40h. Бит очищается программно записью нуля. Запись единицы не влияет. Этот бит также устанавливается, если прерывание запрещено EWI=0

30 Генератор случайных чисел RANDOM

30.1 Назначение и свойства

Модуль служит для управления двумя кольцевыми генераторами, их включения, тестирования, а также формирования случайного 32-битного числа.

Модуль имеет следующие особенности:

– содержит два генератора;

– позволяет включать и выключать генераторы группами по 1;

– позволяет задать паузу после включения генераторов, перед началом сбора случайного числа;

– позволяет включать генераторы по биту регистра управления;

– имеет несколько режимов запуска сбора случайного числа:

  – одиночный;

  – постоянный.

– формирует сигнал прерывания по окончании сбора случайного числа.

Датчик случайных чисел представляет собой механизм получения непредсказуемой последовательности чисел, основанный на физическом недетерминированном процессе.

Для увеличения количества случайных чисел и улучшения их статистических характеристик, пользователь может воспользоваться аппаратно-программным механизмом генерации псевдослучайных чисел. В этом случае выход генератора случайных чисел является рандомизирующим фактором для выполнения операции шифрования с использованием стойкого шифра.

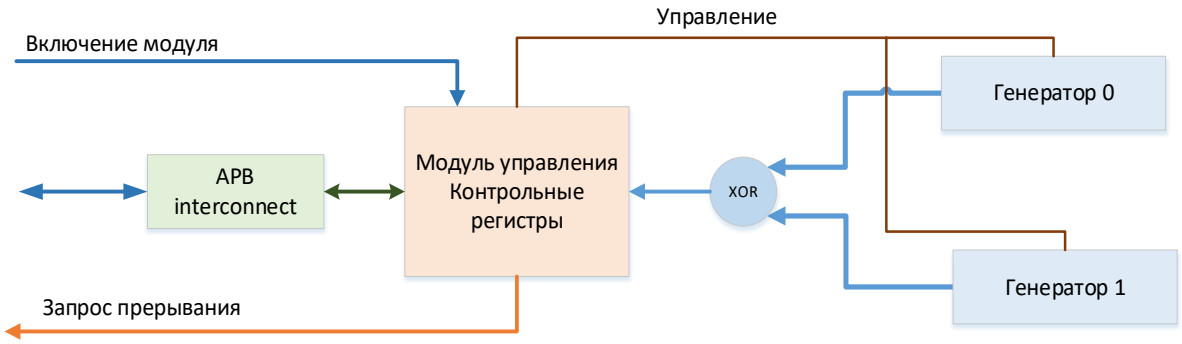


Рисунок 104 – Структурная схема блока генератора случайных чисел

30.2 Регистры модуля

Таблица 408 – Регистры модуля

Базовый Адрес	Название	Описание
0x500B_8000	RANDOM	Блок генератора случайных чисел
Смещение		
0x00	STAT_CTRL	Регистр статуса и управления
0x04	INT_CTRL	Регистр управления прерыванием

0x08	CLK_DIV	Регистр делителя синхросигнала генератора
0x0C	PAUSE	Регистр паузы включения
0x10	OUTPUT	Регистр случайного значения
0x14	PAUSE_CNT	Регистр счётчика паузы
0x18	TEMP	Регистр сбора случайного числа

30.2.1 **Регистр статуса и управления STAT_CTRL**

Относительный адрес: 0x00, начальное значение 0x000000F0.

Таблица 409 – Регистр STAT_CTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
0	–	Зарезервировано. Записывать всегда 0	RW
1	EN_STATE	Генераторы включены, данный бит устанавливается в 1, если есть хоть одна группа включенных генераторов (значение MASK не нулевое) и вышла пауза после включения (*см. PAUSE)*	RO
2	BUSY	Генератор занят, идет сбор случайного числа	RO
3	UNREAD_DATA	Флаг наличия несчитанных данных. После сбора очередного случайного числа в данном бите появляется 1, бит сбрасывается при чтении случайного числа (*см. OUTPUT)*	RO
5, 4	MASK	Маска выбора активного генератора: 4 бит – генератор 0; 5 бит – генератор 1	RW
7, 6	–	Зарезервировано	RO
9, 8	WORK_MODE	Режим сбора случайного числа: 00 – сбор случайных чисел остановлен; 01 – однократный запуск сбора случайного числа (после начала сбора поле сбрасывается в 00 автоматически) 10 – сбор нового случайного числа начинается, если нет непрочитанного случайного числа (UNREAD_DATA равен 0) 11 – постоянный сбор случайных чисел, сбор нового случайного числа начинается сразу по окончании сбора очередного числа	RW
11, 10	–	Зарезервировано	RO
17…12	BIT_CN	Счетчик оставшихся бит до сбора нового случайного числа	RO
31…18	–	Зарезервировано	RO

30.2.2 **Регистр управления прерыванием INT_CTRL**

Относительный адрес: 0x04.

Таблица 410 – Регистр INT_CTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
0	INT_EN	Разрешение прерывания. При наличии 1 в этом бите и 1 в поле UNREAD_DATA, на выходе interrupt появляется 1	RW
1	UNREAD_DATA	Флаг наличия несчитанных данных. После сбора очередного случайного числа в данном бите появляется 1, бит сбрасывается при чтении случайного числа (см. OUTPUT)	RO
31…2	-	Зарезервировано	RO

30.2.3 **Регистр делителя частоты генератора CLK_DIV**

Относительный адрес: 0x08, начальное значение 7.

Таблица 411 – Регистр CLK_DIV

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
15…0	DIV	Для формирования частоты защелки данных генератора используется входная частота модуля, деленная на 2 • (**DIV** + 1)	RW
31…16	-	Зарезервировано	RO

30.2.4 **Регистр паузы включения PAUSE**

Относительный адрес: 0x0C.

Таблица 412 – Регистр PAUSE

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31…0	PAUSE	После включения генераторов через внешний сигнал или бит управляющего регистра, при включении любого блока через маску, при добавлении новых включенных блоков к работающим, сбор случайного числа может быть начат не ранее, чем закончится данная пауза. Если включение происходит во время сбора случайного числа, сбор приостанавливается до тех пор, пока не завершится данная пауза	RW

30.2.5 **Регистр случайного значения OUTPUT**

Относительный адрес: 0x10.

Таблица 413 – Регистр OUTPUT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31…0	DATA	Последнее собранное случайное число. Во время сбора очередного случайного числа значение данного регистра неизменно. Обновление данных происходит по окончании сбора случайного числа	RO

30.2.6 **Регистр счетчика паузы PAUSE_CNT**

Относительный адрес: 0x14.

Таблица 414 – Регистр PAUSE_CNT

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31…0	CUR_PAUSE	Текущее значение счетчика паузы включения генераторов (значение для справки). Нулевое значение означает что пауза выдержана	RO

30.2.7 **Регистр сбора случайного числа TEMP**

Относительный адрес: 0x18.

Таблица 415 – Регистр TEMP

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений	Доступ
31…0	TEMP_DATA	Текущее значение сдвигового регистра сбора случайного числа (значение для справок)	RO

31 Контроллер USART (ISO7816)

31.1 Введение

Универсальный синхронно-асинхронный приемопередатчик (USART) предлагает гибкие средства полнодуплексного обмена с внешним оборудованием, требующим стандартного формата кодирования данных NRZ. Модуль USART поддерживает широкий диапазон скорости передачи данных с использованием генератора дробной скорости передачи данных.

Модуль поддерживает синхронную одностороннюю передачу данных и полудуплексную однопроводную передачу данных. Также поддерживается протокол Smartcard, и операции модема (CTS/RTS).

31.2 Особенности модуля USART

Модуль USART имеет следующие особенности:

- полный дуплекс, асинхронный обмен данными;
- стандартный формат данных NRZ (Mark/Space);
- конфигурируемый метод передискретизации на 16 или 8 для обеспечения гибкости между скоростью и допуском по тактовой частоте;
- системы генератора дробной скорости передачи данных:
  - программируемая скорость передачи и приема данных (значение скорости передачи данных при максимальной частоте шины APB см. в спецификациях);
- программируемая длина слова данных (8 или 9 бит);
- поддержка 2 стоп-бит;
- тактовый выход передатчика для синхронной передачи данных;
- возможность эмуляции смарт-карты (режим Smartcard):
  - интерфейс Smartcard поддерживает асинхронный протокол для смарт-карт в соответствии со стандартом ISO 7816-3;
  - стоп-биты 0,5; 1,5 для интерфейса Smartcard без возможности применения back2back-кадров в режиме 0,5 стоп-бит. Подробнее см. пункт 31.3.1 «Описание символов USART»;
- однопроводная полудуплексная передача данных;
- отдельные биты разрешения (enable) для передатчика и приемника;
- флаги обнаружения передачи:
  - буфер приемника полон;
  - буфер передатчика пуст;
  - флаг завершения передачи;
- проверка четности:
  - передача бита четности;
  - проверка четности принятого байта данных;

– флаги обнаружения ошибки:
  – переполнение;
  – обнаружение шума;
  – ошибка фрейма;
  – ошибка четности;
– прерывания с флагами:
  – изменение состояния CTS;
  – регистр данных передатчика пуст;
  – передача завершена;
  – регистр данных приемника заполнен;
  – ошибка переполнения;
  – ошибка фрейма;
  – обнаружение шума;
  – ошибка четности.

31.3 Функциональное описание USART

Интерфейс подключается к внешним приборам при помощи трех выводов (см. рисунок 105). Любая двунаправленная передача данных USART требует минимум два вывода: вход для данных (RX) и выход для передачи данных (TX):

– **RX**: вход для последовательных принимаемых данных. Для восстановления данных используется техника передискретизации, чтобы отделить нужные входящие данные от шума;

– **TX**: выход для передачи данных. Когда передатчик запрещен, вывод возвращается в состояние, заданное конфигурацией порта ввода-вывода (I/O).

Когда передатчик разрешен, но никакие данные не передаются, на выводе TX устанавливается высокий уровень (логическая 1). В однопроводном режиме и режиме смарт-карты, данный порт ввода-вывода используется для передачи и приема данных (для вывода должен быть включен приемник (регистр PORT_ANALOG) и передатчик (регистр PORT_OE)).

При помощи данных выводов происходит передача и прием последовательных данных в нормальном режиме работы модуля USART, в виде фреймов, содержащих:

– сигнал ожидания линии (Idle) до передачи или приема;

– стартовый бит;

– слово данных (8 или 9 бит), LSB передается первым;

– 0,5, 1,5, 2 стоп-бита, показывающие, что фрейм завершен;

– данный интерфейс использует дробный генератор скорости передачи данных – с 12-битной мантиссой и четырехбитной дробной частью;

– регистр статуса (USART_SR);

– регистр данных (USART_DR);

– регистр скорости передачи (USART_BRR) - 12-битная мантисса и четырехбитная дробная часть;

– регистр защитного интервала (Guardtime register) (USART_GTPR) в случае использования режима Smartcard.

Описание регистров и их бит приведено в подразделе 31.5 «Регистры USART».

Для работы в синхронном режиме требуется дополнительный вывод:

– **CK:** Выход тактов передатчика. На этот вывод выдаются такты данных для синхронной передачи, соответствующей режиму ведущий SPI (нет тактовых импульсов в стартовых и стоп-битах, и программная опция отправки тактового импульса на последнем бите данных). Параллельно данные могут синхронно приниматься по выводу RX. Это можно использовать для управления внешними периферийными устройствами, у которых есть регистры сдвига (например, LCD драйверы). Фаза и полярность тактов выбирается программно. В режиме Smartcard, вывод CK обеспечивает такты для смарт-карты.

В режиме аппаратного управления потоком данных требуются дополнительно выводы:

– **CTS:** сигнал Clear To Send блокирует передачу данных по окончании текущей передачи при высоком уровне сигнала;

– **RTS:** сигнал Request to send показывает готовность USART к приему данных (при низком уровне сигнала).

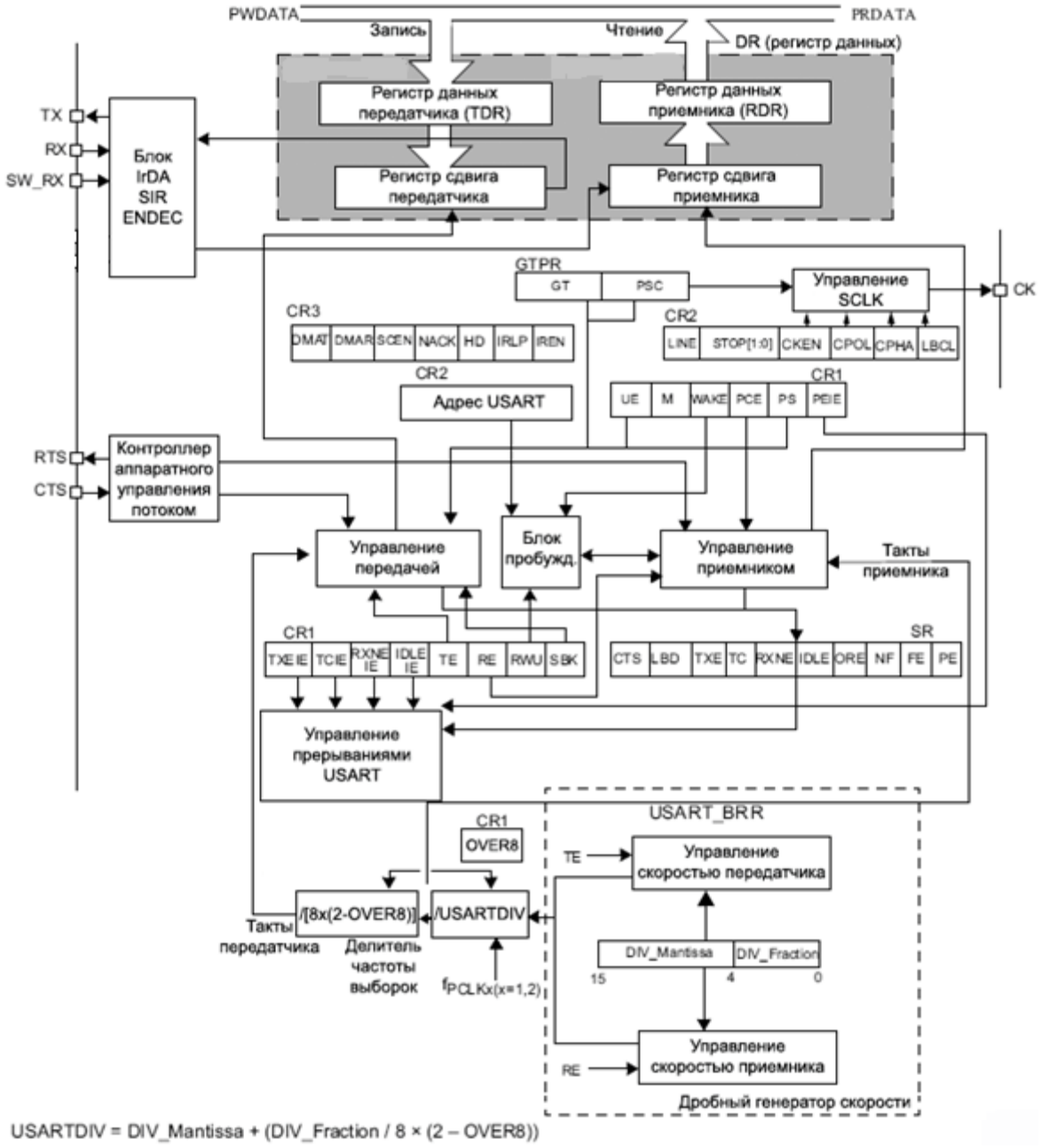


Рисунок 105 – Блок схема USART

31.3.1 **Описание символов USART**

Для передачи можно выбрать длину слова 8 или 9 бит путем программирования бита M в регистре USART_CR1 (см. рисунок 106).

Вывод TX находится в состоянии логического «0» во время передачи стартового бита, и в состоянии логической «1» во время передачи стоп-бита.

Передача и прием управляются генератором скорости передачи. Такты для передатчика и приемника генерируются, когда установлен соответствующий бит разрешения.

Ниже подробно описан каждый блок.

Примечание – При выборе режима работы 0,5 стоп-бит не применять back2back-кадры (кадры, время паузы между которыми меньше одного такта), так как модуль USART не успевает принять кадр из-за ожидания в один такт между кадрами (режим T=1, ISO7816-3).

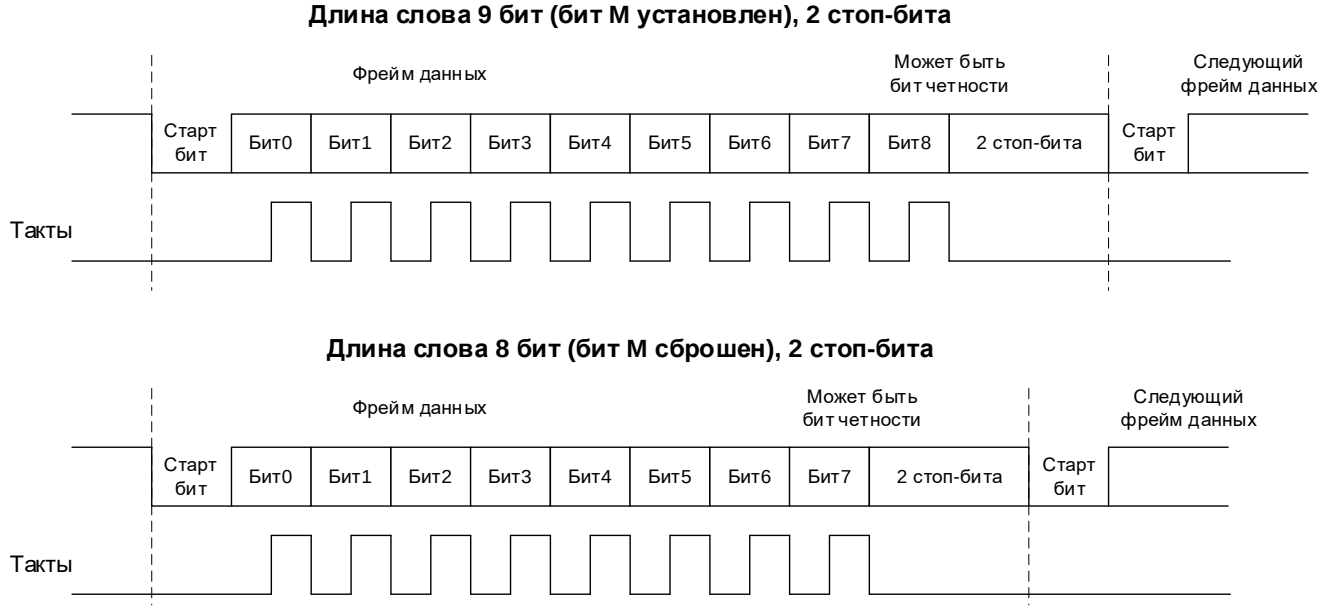


Рисунок 106 – Программирование длины слова

31.3.2 **Передатчик**

Передатчик может отправлять слова данных по 8 или 9 бит, в зависимости от статуса бита M.

Когда установлен бит разрешения передачи (TE), данные из регистра сдвига выводятся на вывод TX, и соответствующие тактовые импульсы выводятся на вывод CK.

31.3.2.1 **Передача символа**

Во время передачи USART данные сдвигаются на вывод TX, младший значащий бит (LSB) идет первым. В данном режиме, регистр USART_DR состоит из буфера (TDR) между внутренней шиной и регистром сдвига передачи (см. рисунок 105).

Каждому символу предшествует стартовый бит низкого уровня (логический «0») в течение длительности одного бита. Символ завершается конфигурируемым количеством стоп-битов.

USART модуль поддерживает следующие стоп-биты: 0.5, 1.5 и 2 стоп-бита.

Примечание – бит TE не должен сбрасываться по время передачи данных. Сброс бита TE во время передачи приведет к повреждению данных на выводе TX, так как счетчики генератора скорости остановятся. Текущие передаваемые данные будут потеряны.

После разрешения бита TE будет отправлен фрейм idle.

31.3.2.2 **Конфигурируемые стоп-биты**

Количество передаваемых стоп-бит с каждым символом может быть запрограммировано в регистре USART_CR2 битами 13, 12.

– *2 стоп-бита*: значение поддерживается в обычном режиме USART, однопроводном режиме и режиме модема;

– *0,5 стоп-бит*: для использования при приеме данных в режиме Smartcard;

– *1,5 стоп-бит*: для использования при передаче и приеме данных в режиме Smartcard.

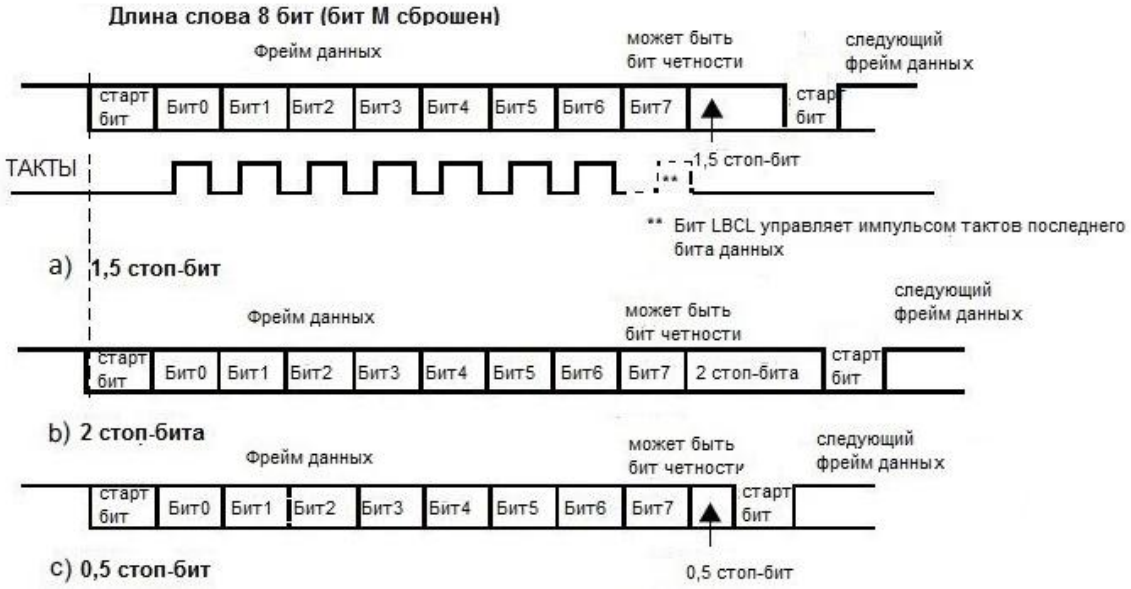


Рисунок 107 – Конфигурируемые стоп-биты

Процедура программирования:

1. Разрешить USART записью в 1 бита UE в регистре USART_CR1;
2. Запрограммировать бит M в регистре USART_CR1, чтобы определить длину слова данных;
3. Запрограммировать количество стоп-бит в регистре USART_CR2;
4. Выбрать необходимую скорость передачи при помощи регистра USART_BRR;
5. Установить бит TE в регистре USART_CR1;
6. Записать данные для передачи в регистр USART_DR (это сбросит бит TXE). Повторить данную операцию для каждого передаваемого символа;
7. После записи последних данных в регистр USART_DR, подождать, пока TC станет равным 1. Это покажет, что передача последнего фрейма завершена. Например, это требуется, когда модуль USART отключен или переходит в режиме Halt, чтобы избежать повреждения последней передачи.

31.3.2.3 **Однобайтный обмен**

Бит TXE всегда сбрасывается при записи в регистр данных.

Бит TXE устанавливается аппаратно, и указывает, что:

– данные были перемещены из регистра TDR в регистр сдвига, и началась передача;

– регистр TDR пуст;

– следующие данные могут быть записаны в регистр USART_DR без риска перезаписи предыдущих данных.

Данный флаг генерирует прерывание, если установлен бит TXE.

Во время передачи данных команда записи в регистр USART_DR помещает данные в регистр TDR, откуда затем данные копируются в регистр сдвига при завершении текущей передачи.

Если передачи данных не происходит, команда записи в регистр USART_DR помещает данные напрямую в регистр сдвига, начинается передача данных, и бит TXE незамедлительно устанавливается.

Если передается фрейм (после стоп-бита) и бит TXE установлен, бит TC устанавливается в высокий уровень (логическая 1). Генерируется прерывание, если бит TCIE установлен в регистре USART_CR1.

После записи последних данных в регистр USART_DR важно дождаться, когда TC установится в 1, перед запретом USART или переходом микросхемы в режим пониженного потребления (см. рисунок 108).

Бит TC сбрасывается следующей программной последовательностью:

1. Чтение из регистра USART_SR;
2. Запись в регистр USART_DR.

Примечание – Бит TC может быть также сброшен при помощи записи «0».

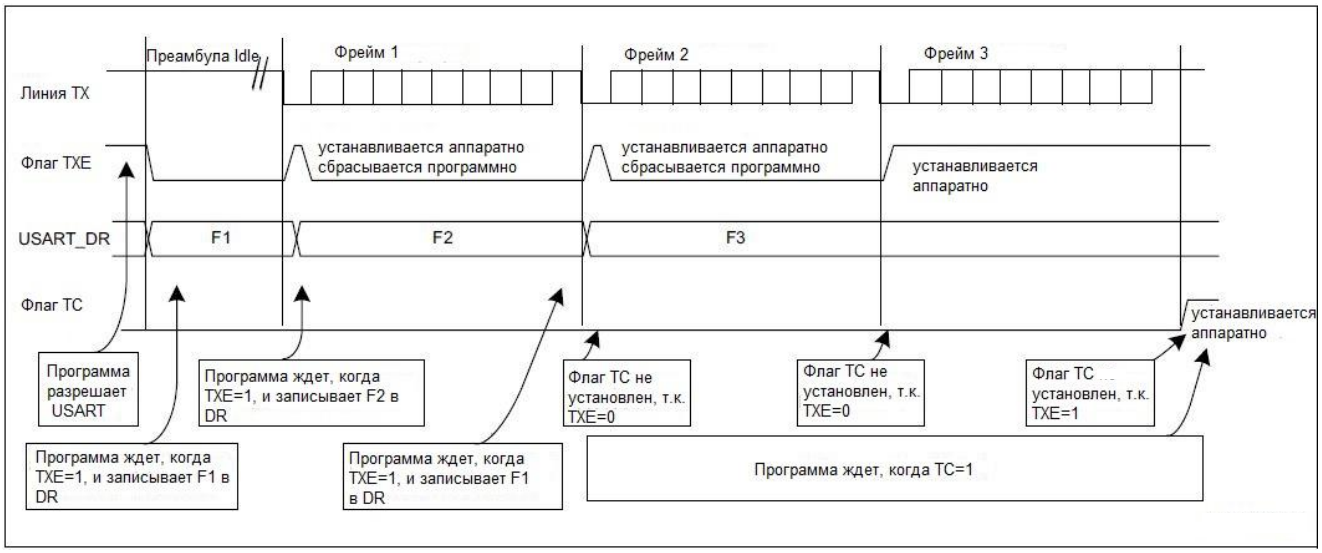


Рисунок 108 – Состояние TC/TXE во время передачи

31.3.3 **Приемник**

USART может принимать слово данных разрядностью 8 или 9 бит, в зависимости от значения бита M в регистре USART_CR1.

31.3.3.1 **Обнаружение стартового бита**

Последовательность обнаружения стартового бита одинаковая при передискретизации сигнала с кратностью 16 или 8.

В приемопередатчике USART стартовый бит детектируется, когда распознана определенная последовательностью выборки сигнала: 1110X0X0X0000.

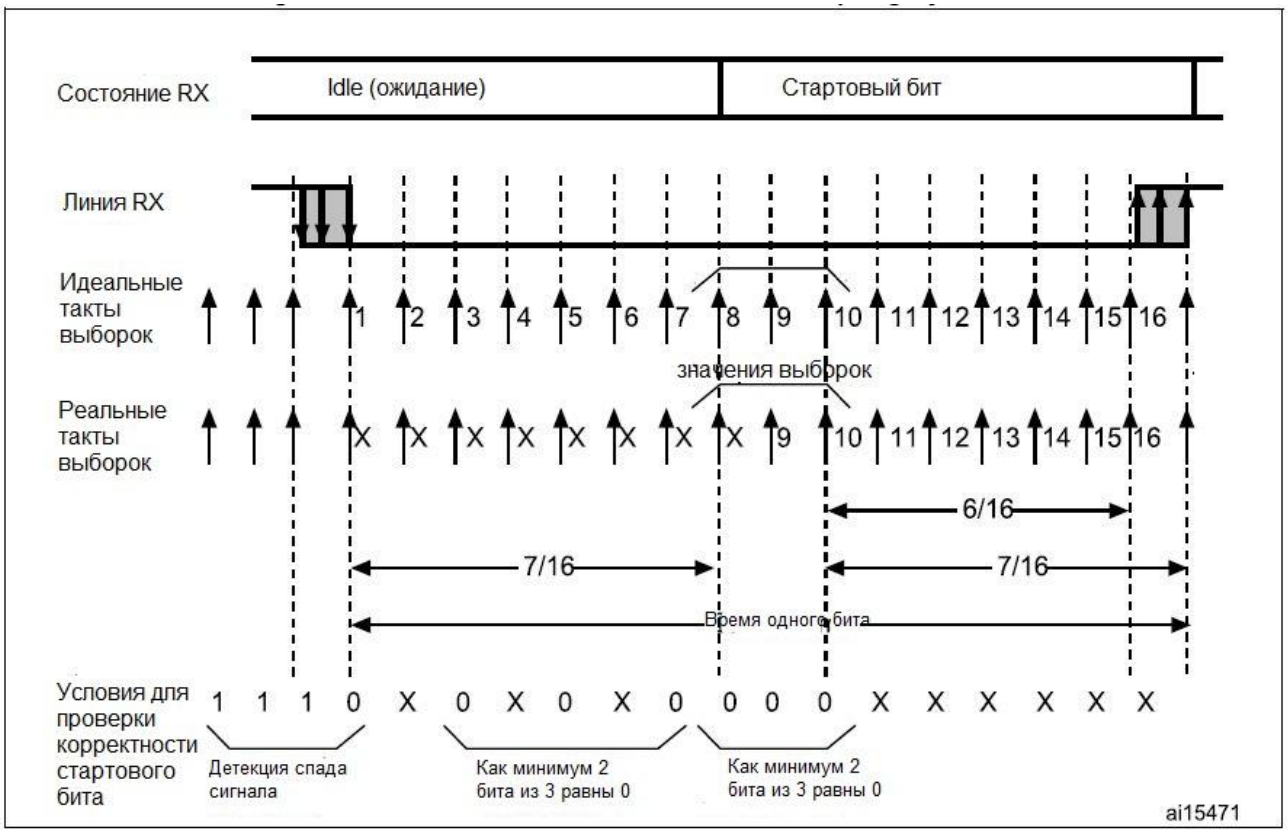


Рисунок 109 – Обнаружение стартового бита при передискретизации на 16 и 8

Примечание – Если последовательность не завершена, обнаружение стартового бита прекращается, и приемник переходит в состояние ожидания (idle) (никакие флаги не установлены) и ожидает заднего фронта.

Стартовый бит подтвержден (флаг RXNE установлен, генерируется прерывание при RXNEIE=1), если 3 бита выборки равны 0 (первая выборка на 3, 5 и 7 бите находит три бита в логическом 0, а вторая выборка на 8, 9 и 10 битах также находит три бита в 0).

Стартовый бит подтвержден (флаг RXNE установлен, генерируется прерывание при RXNEIE=1), но устанавливается флаг NF, если при обеих выборках как минимум 2 из 3 бит установлены в 0 (выборка на 3, 5 и 7 битах и выборка на 8, 9 и 10 битах). Если данное условие не выполняется, не происходит обнаружение стартового бита, и приемник возвращается в состояние ожидания (idle) (нет установленных флагов).

Если при одной из выборок (выборка на 3, 5 и 7 битах или выборка на 8, 9 и 10 битах) 2 из 3 бит находятся в логическом 0, стартовый бит подтверждается, но устанавливается флаг NE.

31.3.3.2 **Прием символа**

При приеме данных данные перемещаются на вывод RX, начиная с LSB. В данном режиме регистр USART_DR состоит из буфера (RDR) между внутренней шиной и регистром сдвига приемника.

Процедура программирования:

1. Разрешить USART путем записи в 1 бита UE в регистре USART_CR1.
2. Запрограммировать бит M в регистре USART_CR1 для определения длины слова данных.
3. Запрограммировать количество стоп-бит в регистре USART_CR2.
4. Выбрать необходимую скорость передачи данных при помощи регистра USART_BRR
5. Установить бит RE в регистре USART_CR1. Это активирует приемник, и он начнет обнаружение стартового бита.

Когда принят символ:

– устанавливается бит RXNE. Он показывает, что содержимое сдвигового регистра передано в RDR. Другими словами, данные были получены и могут быть прочитаны (а также связанные с ними флаги ошибок);

– генерируется прерывание, если установлен бит RXNEIE;

– флаг ошибки устанавливается, если во время приема была обнаружена ошибка фрейма, шума или переполнения;

– бит RXNE сбрасывается программно при чтении регистра USART_DR. Бит RXNE может быть также сброшен записью 1. Бит RXNE должен быть сброшен перед окончанием приема следующего символа, чтобы избежать ошибки переполнения.

Примечание – бит RE не должен сбрасываться при приеме данных. Если бит RE запрещен во время приема, прием текущего байта будет прерван.

31.3.3.3 **Ошибка переполнения (overrun)**

Ошибка переполнения возникает, когда получен символ, но бит RXNE не сброшен. Данные не могут быть переданы из регистра сдвига в регистр RDR, пока бит RXNE не будет сброшен.

Флаг RXNE устанавливается после каждого полученного байта. Ошибка переполнения возникает, если флаг RXNE установлен, когда приняты следующие данные. При появлении ошибки переполнения:

– бит ORE будет установлен;

– содержимое регистра RDR не будет потеряно. Предыдущие данные будут доступны при чтении регистра USART_DR;

– регистр сдвига будет перезаписан. В этот момент любые данные полученные во время ошибки переполнения будут потеряны;

– генерируется прерывание, если установлен бит RXNEIE;

– бит ORE сбрасывается чтением регистра USART_SR, за которым следует операция чтения регистра USART_DR.

Примечание – установленный бит ORE показывает, что как минимум один элемент данных потерян. Существует два варианта:

– если RXNE=1, то последние корректные данные сохраняются в регистр RDR и могут быть прочитаны;

– если RXNE=0, это означает, что последние корректные данные уже были прочитаны, поэтому нечего считывать из регистра RDR. Это может произойти, когда последние корректные данные были прочитаны из регистра RDR одновременно с получением новых данных. Также это может произойти, когда новые данные приняты во время последовательности чтения (между доступом на чтение регистра USART_SR и доступом на чтение регистра USART_DR).

31.3.3.4 **Выбор правильного метода передискретизации**

Приемник использует разные техники передискретизации, конфигурируемые пользователем (кроме синхронного режима), для восстановления данных путем отделения друг от друга входящих данных и шума.

Метод передискретизации можно выбрать путем программирования бита OVER8 в регистре USART_CR1, передискретизация может быть или 16- или 8-кратная тактам скорости (рисунки 110 и 111).

В зависимости от применения:

– выберите передискретизацию на 8 (OVER8=1) для достижения увеличенной скорости (до fPCLK/8). В данном случае максимальный допуск на отклонение тактов для приемника снижается (см. Допуск отклонения тактовой частоты USART приемника);

– выберите передискретизацию на 16 (OVER8=0) для увеличения допуска приемника к отклонению тактов. В данном случае максимальная скорость ограничена fPCLK/16.

Программирование бита ONEBIT в регистре USART_CR3 выбирает метод, используемый для оценки логического уровня. Есть две опции:

– мажоритарная выборка из 3 выборок по центру принятого бита. В данном случае, если эти 3 выборки не одинаковые, установится бит NF;

– одна выборка в центре принятого бита.

В зависимости от применения:

– выберите метод мажоритарности из 3-х выборок (ONEBIT=0) при работе в условиях шума и отклоните данные, если обнаружен шум, потому что это показывает, что во время выборки произошел сбой.

– выберите метод одиночной выборки (ONEBIT=1), если нет шумов на линии, чтобы увеличить допуск на отклонение тактов приемника (см. Допуск на отклонения тактовой частоты USART приемника). В данном случае бит NF не будет установлен.

Когда во фрейме обнаружен шум:

– бит NF устанавливается одновременно с битом RXNE;

– некорректные данные передаются из регистра сдвига в регистр USART_DR;

Бит NF сбрасывается чтением регистра USART_SR, после чего следует чтение регистра USART_DR.

Примечание – передискретизация на 8 недоступна в режимах Smartcard. В данном режиме бит OVER8 аппаратно сбрасывается в 0.

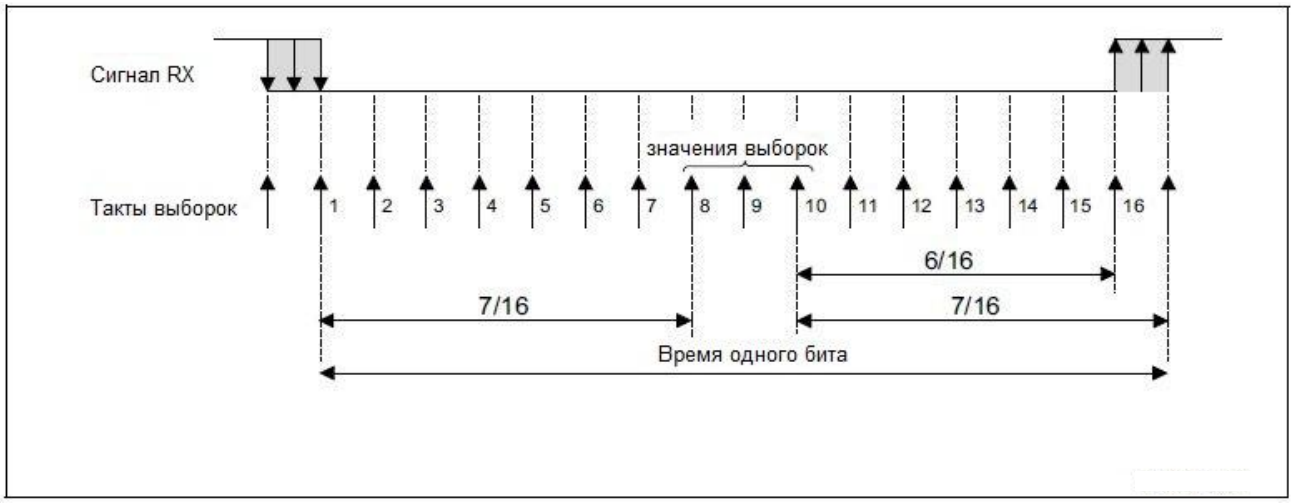


Рисунок 110 – Выборка данных при передискретизации на 16

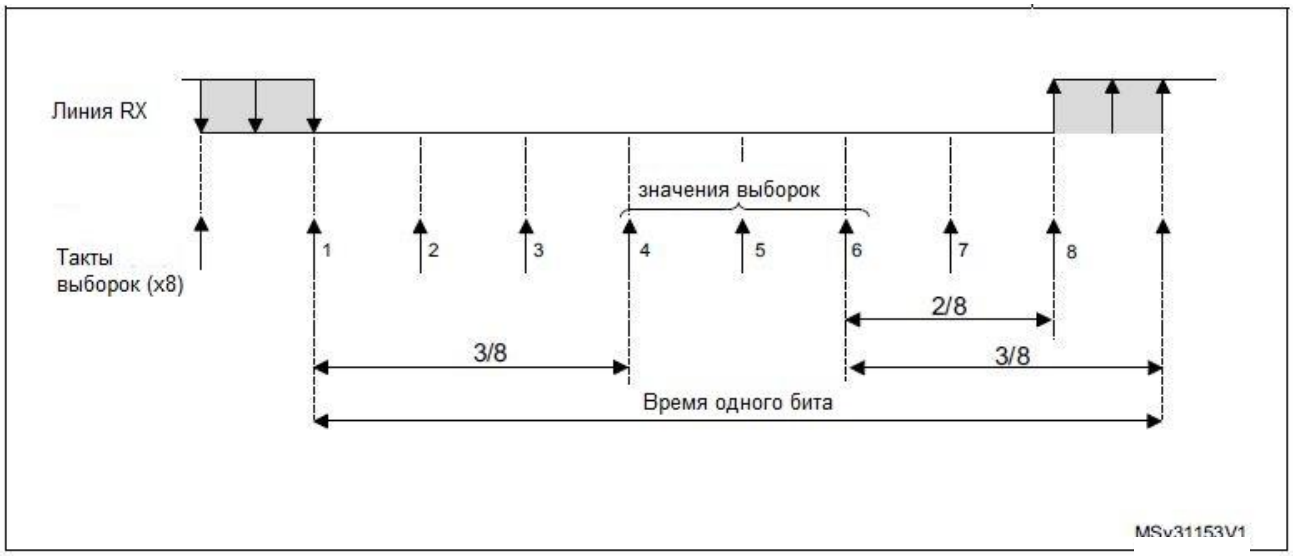


Рисунок 111 – Выборка данных при передискретизации на 8

Таблица 416 – Обнаружение шумов для считанных данных

Значение	NF статус	Значение принятого бита
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

31.3.3.5 **Ошибка фрейма (framing error)**

Ошибка фрейма обнаруживается, когда стоп-бит не распознан во время приема данных в ожидаемый промежуток времени из-за рассинхронизации или из-за чрезмерного шума.

Когда обнаружена ошибка фрейма:

– бит FE устанавливается одновременно с битом RXNE;

– некорректные данные передаются из регистра сдвига в регистр USART_DR.

Бит FE сбрасывается чтением регистра USART_SR, после чего следует чтение регистра USART_DR.

31.3.3.6 **Конфигурируемые стоп-биты во время приема**

Количество принимаемых стоп-бит можно конфигурировать в Регистре Управления 2. Это может быть 2 бита в нормальном режиме и 0,5 или 1,5 в режиме Smartcard.

1 ***0,5 стоп-бит (прием в режиме Smartcard)***: для данного режиме выборки не делаются. Как следствие, при выборе 0,5 стоп-бита ошибка фрейма или сигнал break не могут быть обнаружены.

3 ***1,5 стоп-бита (режим Smartcard)***: при передаче данных в режиме smartcard, прибор должен проверить корректность отправляемых данных. Таким образом, блок приемника должен быть разрешен (RE=1 в регистре USART_CR1), а стоп-бит проверяется на наличие ошибки четности. В случае наличия ошибки четности, смарт-карта подтягивает сигнал данных в логический 0 во время выборки (сигнал NACK), который опознается как ошибка фрейма. После чего флаг FE устанавливается с RXNE по окончании 1,5 стоп-бита. Анализ уровня для 1.5 стоп-бита осуществляется на 16, 17 и 18 выборках (1 период скорости после начала стоп-бита). 1,5 стоп-бит может быть разложен на 2 части: одна часть из 0,5 периода скорости, во время которой ничего не происходит, и вторая часть – 1 период нормального стоп-бита, во время которого происходит анализ сигнала. См. Режим Smartcard.

4 ***2 стоп-бита***: Анализ для 2 стоп-бит осуществляется на 8, 9 и 10 выборках первого стоп-бита. Если была обнаружена ошибка фрейма во время первого стоп-бита, устанавливается флаг ошибки. Второй стоп-бит не проверяется на наличие ошибки фрейма. Флаг RXNE устанавливается по окончании первого стоп-бита.

31.3.4 **Дробный генератор скорости**

Скорость обмена для приемника и передатчика (Rx и Tx) устанавливается в одинаковое значение, программируемое коэффициентами Mantissa (целая часть) и Fraction (дробная часть) делителя USARTDIV.

Скорость для стандартного USART (включая режим SPI) вычисляется по формуле

$$\text{Tx/Rx baud} = \frac{f_{CK}}{8 \cdot (2 - OVER8) \cdot USARTDIV}. \quad (22)$$

Скорость для режимов Smartcard вычисляется по формуле

$$\text{Tx/Rx baud} = \frac{f_{CK}}{16 \times USARTDIV}, \quad (23)$$

где USARTDIV – число с фиксированной запятой без знака, запрограммированное в регистре USART_BRR.

– Если OVER8=0, дробная часть кодируется 4 битами и программируется битами DIV_fraction [3:0] в регистре USART_BRR

– Если OVER8=1, дробная часть кодируется 3 битами и программируется битами DIV_fraction [2:0] в регистре USART_BRR, бит DIV_fraction [3] должен быть сброшен.

Примечание – Счетчики скорости обновляются в регистрах скорости после операции записи в регистр USART_BRR. Следовательно, значение регистра скорости не должно изменяться во время активного обмена данными.

31.3.4.1 Примеры расчета USARTDIV для значений регистра USART_BRR при OVER8=0

Пример 1:

Если DIV_Mantissa = 0d27 и DIV_Fraction = 0d12 (USART_BRR = 0x1BC), тогда
Mantissa (USARTDIV) = 0d27
Fraction (USARTDIV) = 12/16 = 0d0.75
Следовательно, USARTDIV = 0d27.75

Пример 2:

Для программирования USARTDIV = 0d25.62
получается:
DIV_Fraction = 16 • 0d0.62 = 0d9.92
Ближайшее действительное число 0d10 = 0xA
DIV_Mantissa = mantissa (0d25.620) = 0d25 = 0x19
Тогда, USART_BRR = 0x19A, следовательно, USARTDIV = 0d25.625

Пример 3:

Для программирования USARTDIV = 0d50.99
получается:
DIV_Fraction = 16 • 0d0.99 = 0d15.84
Ближайшее действительное число 0d16 = 0x10 => переполнение DIV_frac[3:0] => должен быть добавлен перенос (carry) к мантиссе
DIV_Mantissa = mantissa (0d50.990 + carry) = 0d51 = 0x33

Тогда, USART_BRR = 0x330, следовательно, USARTDIV = 0d51.000

31.3.4.2 Примеры расчета USARTDIV для значений регистра USART_BRR при OVER8=1

Пример 1:

Если DIV_Mantissa = 0x27 и DIV_Fraction[2:0]= 0d6 (USART_BRR = 0x1B6), тогда

Mantissa (USARTDIV) = 0d27
Fraction (USARTDIV) = 6/8 = 0d0.75
Therefore USARTDIV = 0d27.75

Пример 2:

Для программирования USARTDIV = 0d25.62
получается:
DIV_Fraction = 8 • 0d0.62 = 0d4.96
Ближайшее действительное число 0d5 = 0x5
DIV_Mantissa = mantissa (0d25.620) = 0d25 = 0x19
Тогда, USART_BRR = 0x195 => USARTDIV = 0d25.625

Пример 3:

Для программирования USARTDIV = 0d50.99

Получается:

$$DIV_Fraction = 8 \cdot 0d0.99 = 0d7.92$$

Ближайшее действительное число 0d8 = 0x8 => переполнение DIV_frac[2:0] => должен быть добавлен перенос (carry) к мантиссе

DIV_Mantissa = mantissa (0d50.990 + carry) = 0d51 = 0x33
Тогда, USART_BRR = 0x0330 => USARTDIV = 0d51.000

Таблица 417 – Определение погрешности для запрограммированных скоростей передачи при f_{PCLK} = 60 МГц(1)

№	Требуемая скорость	Передискретизация на 16 (OVER8=0)			Передискретизация на 8 (OVER8=1)		
		Реальная скорость	Значение в регистре скорости передачи USART_BRR	Ошибка, %	Реальная скорость	Значение в регистре скорости передачи USART_BRR	Ошибка, %
1	2,4	2,4	1562,5	0	2,4	3125	0

	Кбит/с	Кбит/с			Кбит/с		
2	9,6 Кбит/с	9,6 Кбит/с	390,625	0	9,6 Кбит/с	781,25	0
3	19,2 Кбит/с	19,2 Кбит/с	195,3125	0	19,2 Кбит/с	390,625	0
4	57,6 Кбит/с	57,582 Кбит/с	65,125	0,03	57,582 Кбит/с	130,25	0,03
5	115,2 Кбит/с	115,163 Кбит/с	32,5625	0,03	115,163 Кбит/с	65,125	0,03
6	230,4 Кбит/с	230,769 Кбит/с	16,25	0,16	230,769 Кбит/с	32,5	0,16
7	460,8 Кбит/с	461,538 Кбит/с	8,125	0,16	461,538 Кбит/с	16,25	0,16
8	896 Кбит/с	895,522 Кбит/с	4,1875	0,05	895,522 Кбит/с	8,375	0,05
9	921,6 Кбит/с	923,077 Кбит/с	4,0625	0,16	923,077 Кбит/с	8,125	0,16
10	2 Мбит/с	2 Мбит/с	1,875	0	2 Мбит/с	3,75	0
11	3 Мбит/с	3 Мбит/с	1,25	0	3 Мбит/с	2,5	0
12	4 Мбит/с	NA	NA	NA	4 Мбит/с	1,875	0
13	5 Мбит/с	NA	NA	NA	5 Мбит/с	1,5	0
14	6 Мбит/с	NA	NA	NA	6 Мбит/с	1,25	0

(1) Чем ниже тактовая частота центрального процессора, тем меньше точность для конкретного значения скорости передачи. Верхний предел достижимой скорости передачи может быть зафиксирован с этими данными.

31.3.5 Допуск ухода тактовой частоты для приемника USART

Асинхронный приемник USART корректно работает, только если общее отклонение тактовой частоты меньше, чем допуск приемника USART. Причины, которые способствуют общему отклонению:

– DTRA: отклонение из-за ошибки передатчика (что также включает отклонение от локального тактового генератора передатчика);

– DQUANT: ошибка квантования скорости приемника;

– DREC: отклонение локального генератора тактов приемника;

– DTCL: отклонение из-за линии передачи (обычно из-за того, что приемопередатчики могут давать ассиметричные перепады от 0 к 1 по сравнению с перепадами от 1 к 0).

$$DTRA + DQUANT + DREC + DTCL < \text{допуск приемника } USART$$

Допуск приемника USART для правильного приема данных равен максимально допустимому отклонению и зависит от следующих параметров:

– длина символа 10 или 11 бит, что определяется битом M в регистре USART_CR1

– передискретизация на 8 или 16, что определяется битом OVER8 в регистре USART_CR1

– используется или нет дробная установка скорости

– используется 1 или 3 бита для оцифровки данных, в зависимости от бита ONEBIT в регистре USART_CR3.

Таблица 418 – Допуск приемника USART при DIV_fraction = 0

Бит M	OVER8 = 0		OVER8 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3,75%	4,375%	2,50%	3,75%
1	3,41%	3,97%	2,27%	3,41%

Таблица 419 – Допуск приемника USART при DIV_Fraction, не равным 0

Бит M	OVER8 = 0		OVER8 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3,33%	3,88%	2%	3%
1	3,03%	3,53%	1,82%	2,73%

31.3.6 **Многопроцессорный обмен**

Существует возможность многопроцессорного обмена данными через USART (несколько приемопередатчиков USART, объединенных в одну сеть). Например, один USART может быть ведущим устройством, его выход TX подключается ко входу RX другого приемопередатчика USART. Другие приемопередатчики являются ведомыми, их соответствующие выходы TX логически объединены операцией «логическое И» и подсоединены ко входу RX ведущего устройства.

В многопроцессорных конфигурациях предпочтительно, чтобы только один получатель сообщения активно принимал все сообщение целиком, чтобы уменьшить обработку избыточных данных для приемников, которым эти данные не адресованы.

Неадресованные устройства могут быть переведены в режим «молчания» (mute mode) при помощи функции приостановки. В данном режиме:

– никакой статус бит приема не может быть установлен;

– все прерывания приема запрещены;

– бит RWU в регистре USART_CR1 установлен в 1. Бит RWU может управляться автоматически аппаратно или может быть записан программой при определенных условиях.

USART может войти в режим молчания или выйти из него при помощи одного из двух методов, в зависимости от значения бита WAKE в регистре USART_CR1:

– обнаружение линии Idle, если бит WAKE сброшен;

– обнаружение сигнала метки адреса (Address Mark), если бит WAKE установлен.

31.3.6.1 Обнаружение линии Idle (WAKE=0)

USART переключится в режим молчания, если бит RWU установлен в 1.

Приемопередатчик выйдет из режима молчания, когда будет обнаружен фрейм Idle. После чего бит RWU сбрасывается аппаратной частью, но бит IDLE в регистре USART_SR не устанавливается. Бит RWU может быть записан в 0 программно.

На рисунке 112 показан пример поведения устройства в режиме молчания с использованием обнаружения линии Idle.

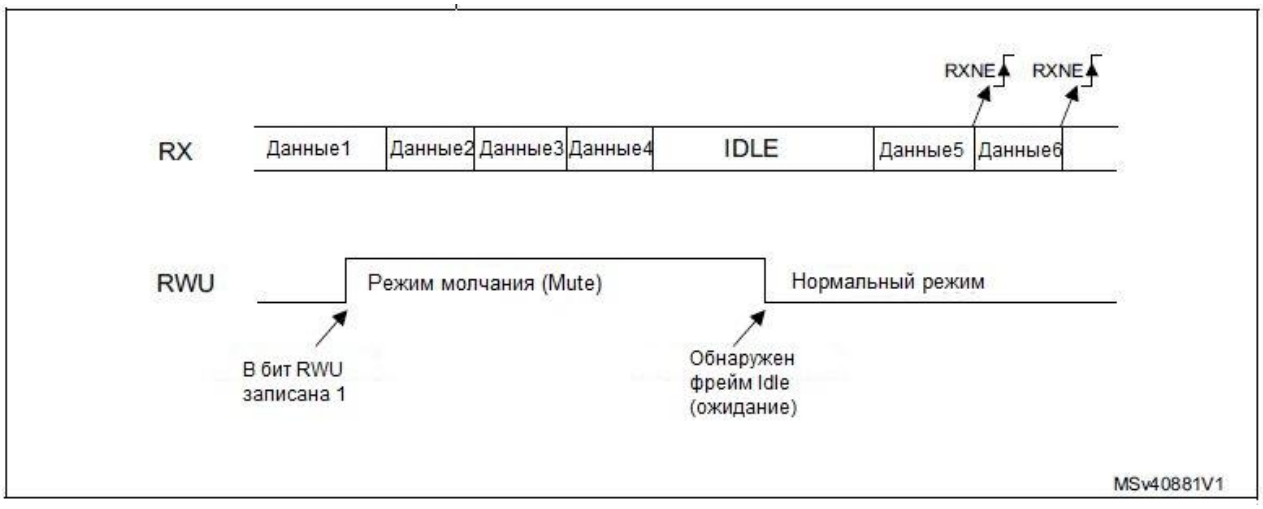


Рисунок 112 – Режим молчания (Mute) с обнаружением линии Idle

31.3.6.2 Обнаружение метки адреса (Address mark) (WAKE=1)

В данном режиме, байты распознаются как адреса, если MSB равен 1, в другом случае они распознаются как данные. В байте адреса адрес приемника помещается в 4 LSB. Приемник сравнивает это 4-битное слово с собственным адресом, запрограммированным в битах ADD в регистре USART_CR2.

USART переключится в режим молчания, когда будет получен символ адреса, который не соответствует запрограммированному режиму. В данном случае, бит RWU устанавливается аппаратно. Флаг RXNE не установится для данного байта адреса, и не будет сгенерировано прерывание, так как приемопередатчик перейдет в режим молчания.

Приемопередатчик выйдет из режима молчания, когда будет принят символ адреса, совпадающий с запрограммированным адресом. После чего бит RWU будет сброшен и последующие биты будут приняты. Символ адреса будет установлен в бите RXNE, так как бит RWU был сброшен.

Бит RWU может быть записан в 0 или 1, когда буфер приемника не содержит данных (RXNE=0 в регистре USART_SR). Иначе попытка записи будет проигнорирована.

Пример поведения приемопередатчика в режиме молчания с использованием режима обнаружения метки адреса приведен на рисунке 113.

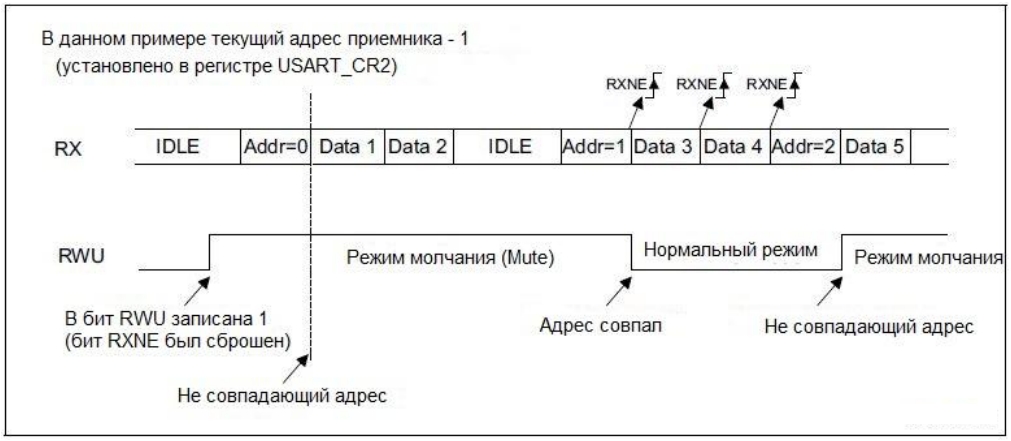


Рисунок 113 – Режим молчания (Mute) с обнаружением метки адреса

31.3.7 **Контроль четности**

Контроль четности бит (генерация бита четности при передаче и контроль бита четности при приеме) разрешается установкой бита PCE в регистре USART_CR1. В зависимости от длины фрейма, определяемой битом M, возможные форматы фрейма приемопередатчика USART приведены в таблице 420.

Таблица 420 – Форматы фрейма

Бит M	Бит PCE	Фрейм USART (1)
0	0	| SB | 8 бит данных | STB |
0	1	| SB | 7 бит данных | PB | STB |
1	0	| SB | 9 бит данных | STB |
1	1	| SB | 8 бит данных | PB | STB |
(1) SB: стартовый бит, STB: стоп-бит, PB: бит четности		

31.3.7.1 **Проверка на четность**

Бит четности вычисляется так, чтобы получить четную сумму всех «1» во фрейме из 7 или 8 бит (в зависимости от значения бита M) и бита четности.

Например, data=00110101; установлено 4 бита => бит четности = 0, если выбрана проверка на четность (бит PS в USART_CR1 = 0).

31.3.7.2 **Проверка на нечетность**

Бит четности вычисляется так, чтобы получить нечетную сумму всех «1» во фрейме из 7 или 8 бит (в зависимости от значения бита M) и бита четности.

Например, data=00110101; установлено 4 бита => бит четности = 1, если выбрана проверка на нечетность (бит PS в USART_CR1 = 1).

31.3.7.3 **Проверка четности при приеме**

Если проверка четности показала ошибку данных, устанавливается флаг PE в регистре USART_SR, и генерируется прерывание, если в регистре USART_CR1

установлен бит PEIE. Флаг PE сбрасывается программной последовательностью (чтение регистра статуса, за которым следует чтение или запись регистра данных USART_DR).

Примечание – в случае пробуждения по метке адреса: MSB используется для идентификации адреса, а не бит четности. Приемник не проверяет бит четности данных адреса (бит PE не устанавливается в случае ошибки четности).

31.3.7.4 **Генерация бита четности при передаче**

Если в регистре USART_CR1 установлен бит PCE, то MSB данных, записанных в регистре данных, будет передан, но он будет изменен битом четности (четное количество «1», если выбрана проверка на четность (PS=0) или нечетное количество «1», если выбрана проверка на нечетность (PS=1)).

Примечание – Программная часть, которая контролирует передачу, может активировать программную последовательность для сброса флага PE (чтение регистра статуса, за которым следует доступ на чтение или запись регистра данных). При работе в полудуплексном режиме, в зависимости от программной части, это может привести к неожиданному сбросу флага PE.

31.3.8 **Синхронный режим USART**

Синхронный режим выбирается записью бита CLKEN в «1» в регистре USART_CR2.

В синхронном режиме следующие биты должны быть сброшены:

– SCEN и HDSEL в регистре USART_CR3.

USART дает возможность пользователю управлять двунаправленным синхронным последовательным обменом данных в режиме ведущего приемопередатчика. Вывод CK является выходом тактов передатчика USART. Во время стартового и стоп-битов никакие тактовые импульсы не посылаются на вывод CK. В зависимости от состояния бита LBCL в регистре USART_CR2 тактовые импульсы будут или не будут генерироваться во время последнего достоверного бита данных (маркер адреса). Бит CPOL в регистре USART_CR2 позволяет пользователю выбрать полярность тактов, а бит CPHA в регистре USART_CR2 позволяет выбрать фазу внешних тактов (см. рисунки 114, 115 и 116)

Во время состояния ожидания (Idle), преамбулы и отправки символа break, внешний тактовый сигнал CK не активируется.

В синхронном режиме, передатчик работает точно также, как и в асинхронном режиме. Но так как сигнал CK синхронизируется с сигналом TX (в зависимости от CPOL и CPHA), данные на TX синхронные.

Приемник в данном режиме работает не так, как в асинхронном режиме. Если RE=1, данные тактируются синхронно с сигналом CK (передний и задний фронт, в зависимости от значений CPOL и CPHA), без какой-либо передискретизации. Время установки и удержания сигнала должно соблюдаться (зависит от скорости обмена: 1/16 от времени бита).

Примечание – Вывод CK работает совместно с выводом TX. Таким образом, тактирование обеспечивается только, если разрешен передатчик (TE=1) и передаются данные (регистр данных USART_DR записан). Это означает, что невозможно принимать синхронные данные без передачи данных.

Должны быть выбраны биты LBCL, CPOL и CPHA, когда передатчик и приемник отключены (TE=RE=0) для обеспечения корректной работы тактовых импульсов. Значения данных бит не должны изменяться во время включения передатчика или приемника.

Рекомендуется устанавливать биты TE и RE одной командой с целью минимизации времени установки и удержания приемника.

Приемопередатчик USART поддерживает только режим ведущего устройства: он не может принимать или передавать данные, связанные со входом тактирования (CK всегда работает как выход).

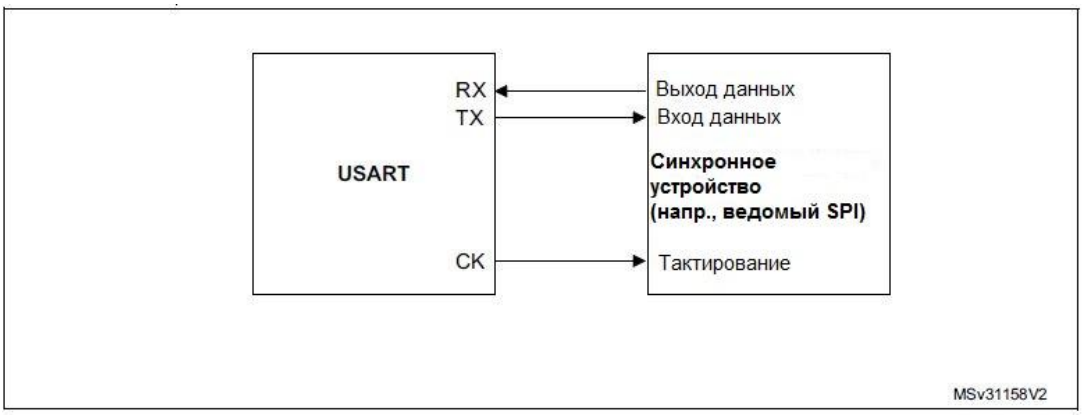


Рисунок 114 – Пример синхронной передачи USART

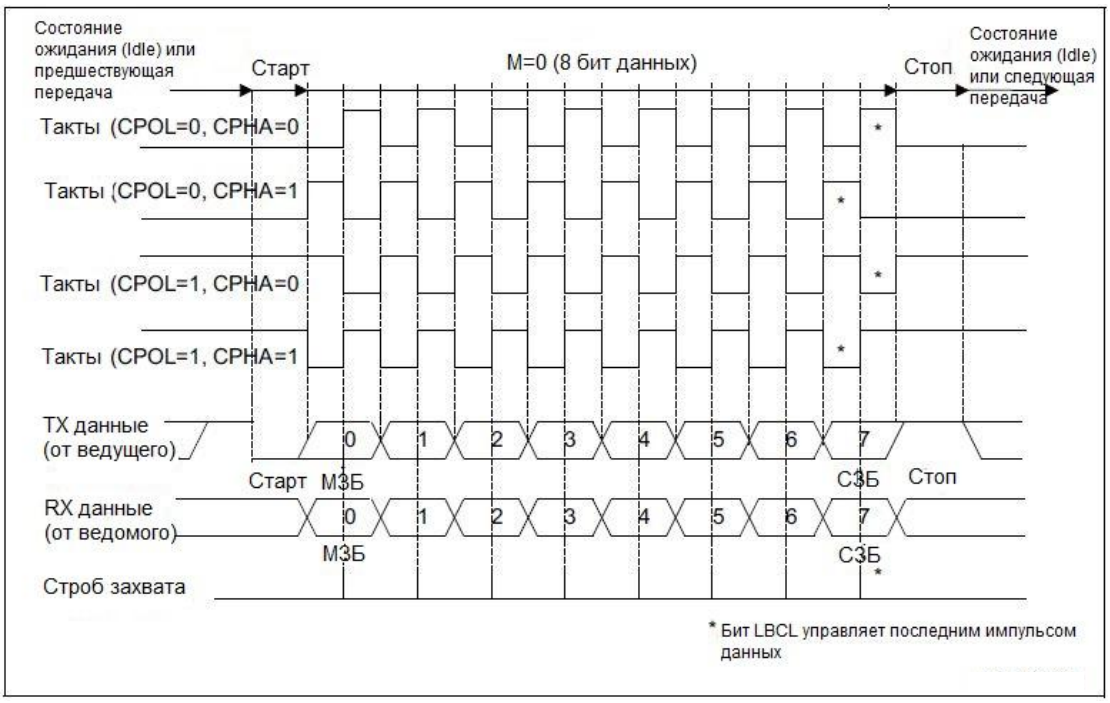


Рисунок 115 – Диаграмма тактирования данных USART (M=0)

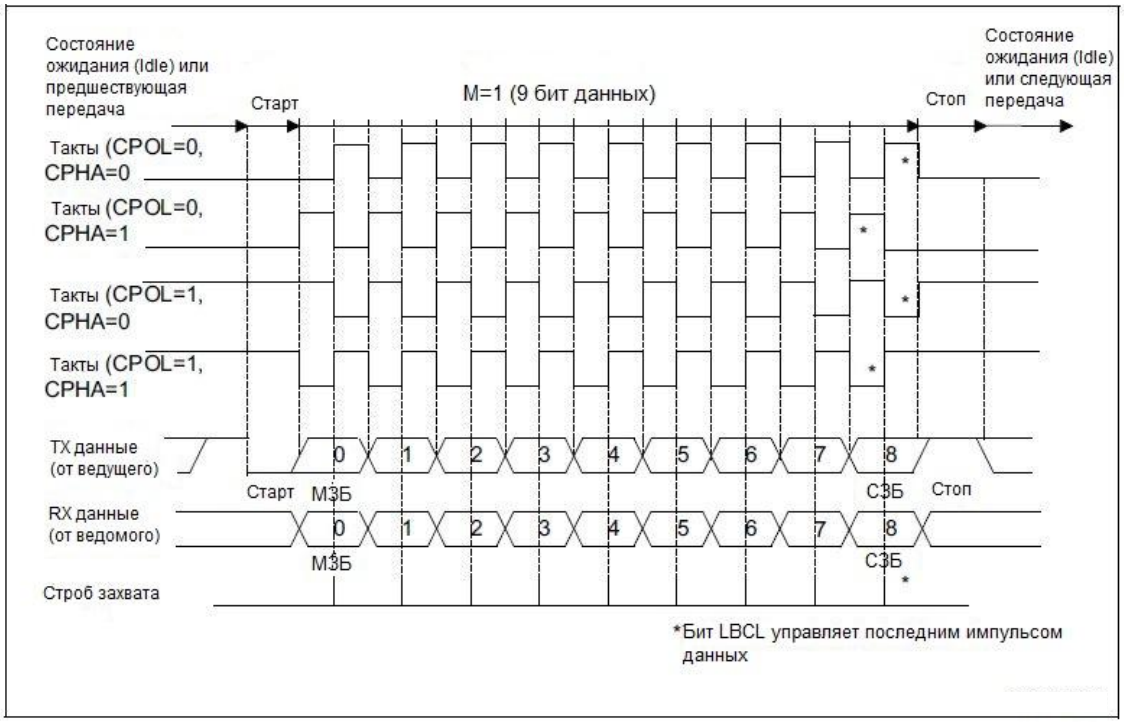


Рисунок 116 – Диаграмма тактирования данных USART (M=1)

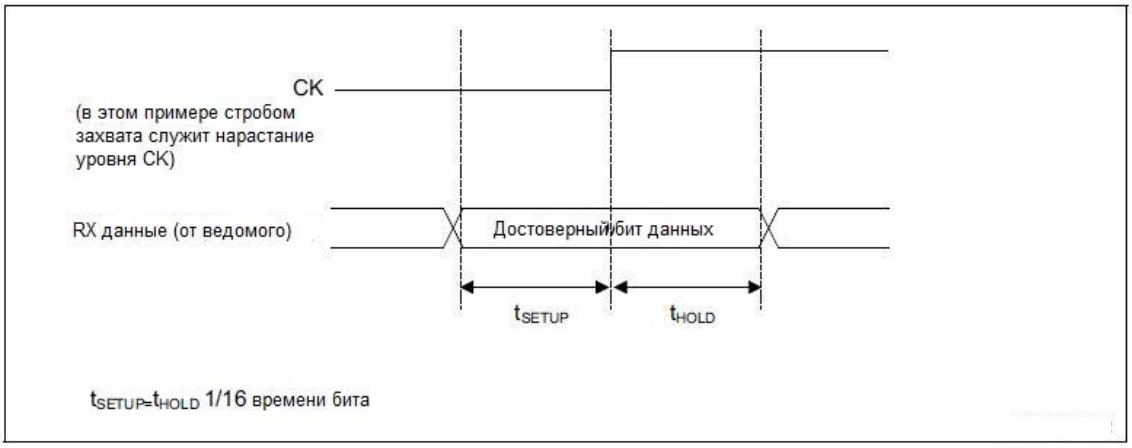


Рисунок 117 – Время установки/удержания RX

Примечание – Функция тактирования CK отличается в режиме Smartcard (См. далее описание режима Smartcard).

31.3.9 **Однопроводной полудуплексный обмен данными**

Однопроводной полудуплексный режим обмена данными (single-wire half-duplex mode) выбирается путем установки бита HDSEL в регистре USART_CR3. В данном режиме должны быть сброшены следующие биты:

– SCEN в регистре USART_CR3.

USART может быть сконфигурирован для соответствия однопроводному полудуплексному протоколу, где линии сигналов TX и RX имеют внутреннее соединение. Выбор между полудуплексным или полнодуплексным режимом передачи данных осуществляется управляющим битом ‘HALF DUPLEX SEL’ (HDSEL в регистре USART_CR3).

Как только бит HDSEL устанавливается в «1»:

– линии сигналов TX и RX соединяются внутри;

– вывод RX больше не используется;

– вывод TX всегда свободен, если данные не передаются. Таким образом, он используется как стандартный ввод/вывод в режиме ожидания или при приеме данных. Это означает, что данный ввод/вывод должен быть сконфигурирован так, чтобы он был высокоомным входом (или выходом с открытым стоком), когда он не управляется USART.

Кроме этого, обмен данными происходит так же, как и обычном режиме передачи данных USART.

Конфликты на линии связи должны контролироваться программно (например, с использованием централизованного арбитра шины). В частности, передача данных никогда не блокируется аппаратной частью и будет продолжаться, как только данные будут записаны в регистр данных, пока установлен бит TE.

31.3.10 **Режим Smartcard**

Режим смарт-карты (Smartcard mode) выбирается путем установки бита SCEN в регистре USART_CR3. В режиме Smartcard следующие биты должны быть сброшены:

– бит HDSEL в регистре USART_CR3.

Более того, бит CLKEN может быть установлен с целью тактирования смарт-карты.

Интерфейс режима Smartcard разработан для поддержки асинхронного протокола в соответствии со стандартом ISO 7816-3. USART должен быть сконфигурирован следующим образом:

– 8 бит + бит четности: где M=1 и PCE=1 в регистре USART_CR1

– 1,5 стоп-бита при передаче и приеме: где STOP=11 в регистре USART_CR2.

Примечание – Также можно выбрать 0,5 стоп-бит для приема, но рекомендуется использовать 1,5 стоп-бита для передачи и приема данных, чтобы избежать переключения между двумя конфигурациями.

Рисунок 118 показывает пример, как выглядит сигнал данных с ошибкой четности и без ошибки четности.

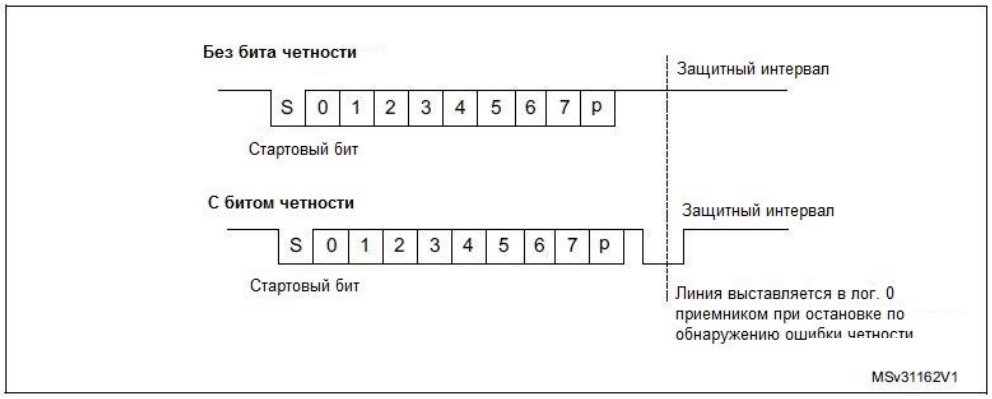


Рисунок 118 – Асинхронный протокол ISO 7816-3

При подключении к смарт-карте, выход TX приемопередатчика USART управляет двунаправленной линией, которая также управляется и смарт-картой. Вывод TX должен быть сконфигурирован как открытый сток.

Smartcard – это однопроводный полудуплексный протокол передачи данных.

– Передача данных из регистра сдвига передатчика осуществляется с гарантированной задержкой минимум ½ такта скорости. При нормальном режиме работы полная передача из регистра сдвига начнется сдвигом на следующем перепаде тактов скорости. В режиме Smartcard эта передача дополнительно задерживается на ½ такта скорости.

– Если обнаружена ошибка четности при приеме фрейма, запрограммированного на 0,5 или 1,5 стоп-бита, линия передачи подтягивается в 0 на период такта скорости после завершения приема фрейма. Это показывает смарт-карте, что данные, переданные на приемопередатчик USART, не были корректно приняты. Данный сигнал NACK (подтягивание линии передачи в 0 на период 1 такта скорости) сгенерирует ошибку фрейма на стороне передатчика (конфигурация 1,5 стоп-бита). Приложение должно обработать эту ситуацию повторной отправкой данных в соответствии с протоколом. Ошибка четности не подтверждается приемником (сигнал NACK), если установлен управляющий бит NACK, иначе сигнал NACK не передается.

– Установка флага TC может быть задержана путем программирования регистра защитного интервала времени (Guard Time). При нормальной работе флаг TC устанавливается, когда регистр сдвига передатчика пуст и нет никаких запросов на передачу. В режиме смарт-карты пустой регистр сдвига передатчика запускает счетчик защитного интервала для счета запрограммированного значения в регистре защитного интервала времени (Guard Time register). Флаг TC подтягивается в 0. Когда счетчик достигает запрограммированного значения, устанавливается флаг TC.

– На снятие флага TC режим смарт-карты не влияет.

– Если обнаружена ошибка фрейма на стороне передатчика (ответ NACK от приемника), сигнал NACK не будет обнаружен как стартовый бит блоком приема передатчика. В соответствии со стандартом ISO, длительность принятого сигнала NACK может составлять 1 или 2 периода тактов скорости.

– На стороне приемника, если была обнаружена ошибка четности, и был передан сигнал NACK, приемник не обработает сигнал NACK как стартовый бит.

Примечание – Символ break не имеет значения в режиме Smartcard. Данные 0x00 с ошибкой фрейма будут обработаны как данные, а не как символ break.

Фрейм Idle не передается при переключении бита TE. Фрейм Idle (как определено для других конфигураций) не определен в протоколе ISO.

Рисунок 119 показывает, как обрабатывается сигнал NACK. В данном примере USART передает данные и сконфигурирован на 1.5 стоп-бита. Блок приемника USART разрешен для проверки целостности данных и сигнала NACK.

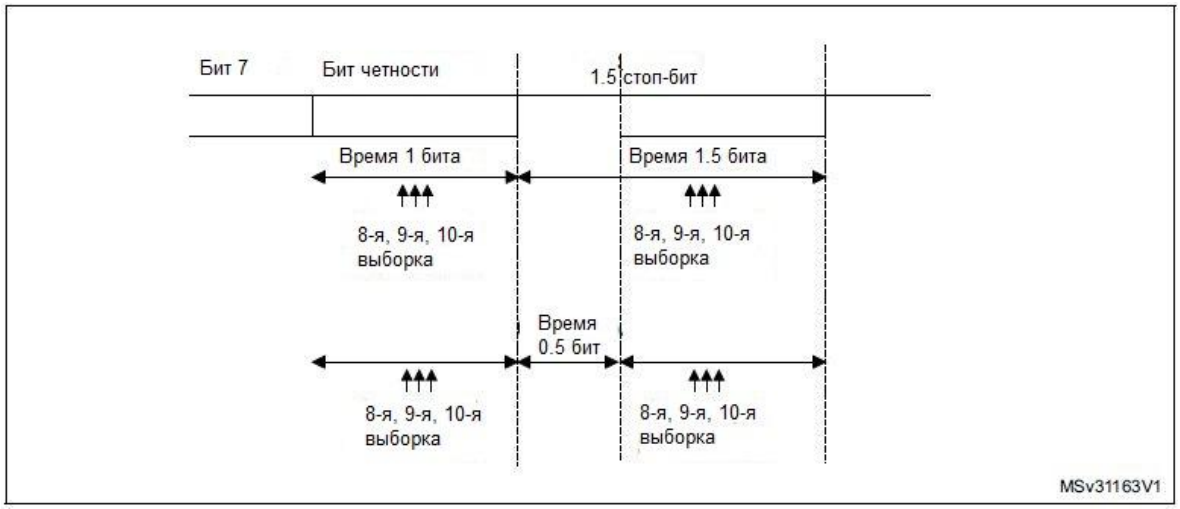


Рисунок 119 – Обнаружение ошибки четности с использованием1,5 стоп-бит

USART может тактировать смарт-карту при помощи выхода CK. В режиме смарт-карты, выход CK не связан с обменом данными, он просто генерирует тактовую частоту, полученную из внутреннего входа тактирования периферии при помощи 5-битного предделителя. Коэффициент деления конфигурируется в регистре предделителя USART_GTPR. Частота CK программируется от $f_{CK}/2$ до $f_{CK}/62$, где f_{CK} – частота тактов входа периферии.

31.3.11 **Аппаратное управление потоком**

Можно управлять потоком последовательных данных между двумя устройствами, используя вход CTS и выход RTS. Рисунок 120 показывает схему соединения двух устройств для данного режима:

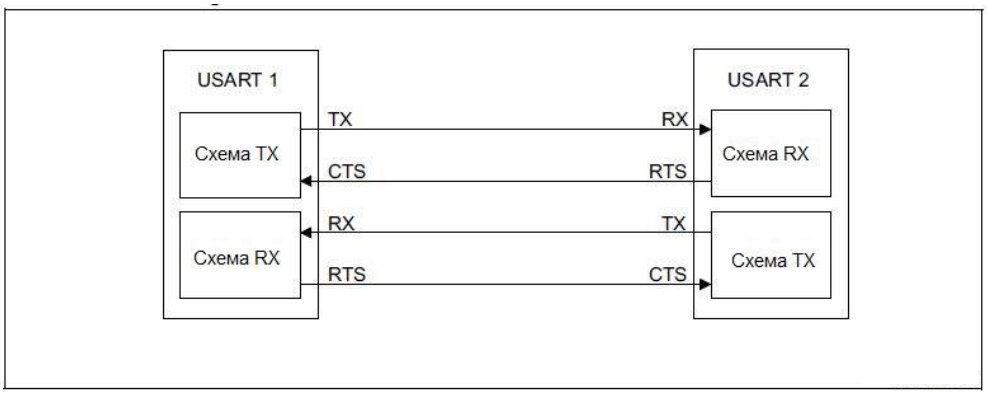


Рисунок 120 – Аппаратное управление потоком данных между двумя устройствами USART

Управление потоком RTS и CTS можно разрешить независимо записью битов RTSE и CTSE соответственно в 1 (в регистре USART_CR3).

31.3.11.1 **Управление потоком RTS**

Если разрешено управление потоком RTS (RTSE=1), выставляется сигнал RTS (логический «0»), когда приемник USART готов принимать новые данные. Если регистр приема заполнен, то сигнал RTS снимается, тем самым показывая, что ожидается остановка передачи в конце текущего фрейма.

Рисунок 121 показывает пример обмена с разрешенным управлением потоком RTS.

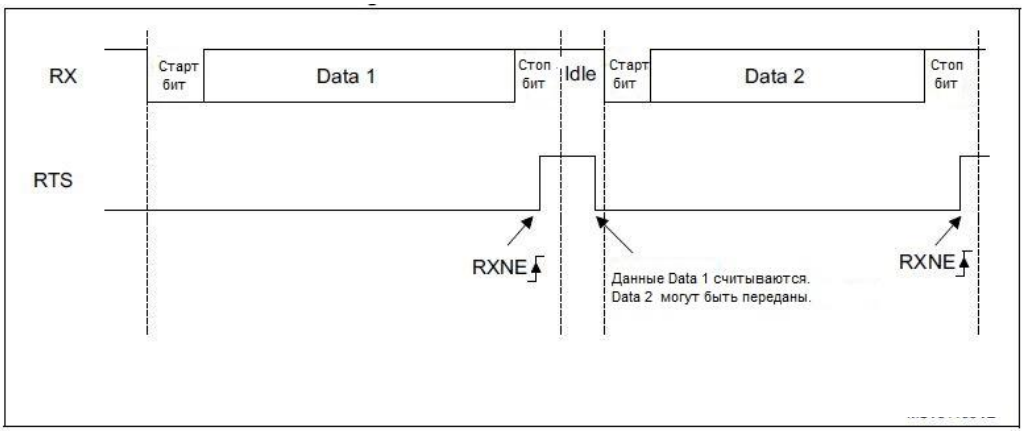


Рисунок 121 – Управление потоком RTS

31.3.11.2 **Управление потоком CTS**

Если управление потоком CTS разрешено (CTSE=1), передатчик проверяет вход CTS перед передачей следующего фрейма. Если устанавливается сигнал CTS (лог. 0), тогда начинается передача следующих данных (подразумевается, что есть данные для передачи, то есть, другими словами, TXE=0), в ином случае передача не произойдет. Когда снимается сигнал CTS во время передачи, текущая передача завершится перед остановкой передатчика.

Если CTSE=1, статус CTSIF будет установлен автоматически аппаратурой, как только вход CTS переключится. Это показывает, когда приемник будет готов к обмену данными. Прерывание генерируется, если установлен бит CTSIE в регистре USART_CR3. Рисунок 122 показывает пример обмена с разрешенным управлением потоком CTS.

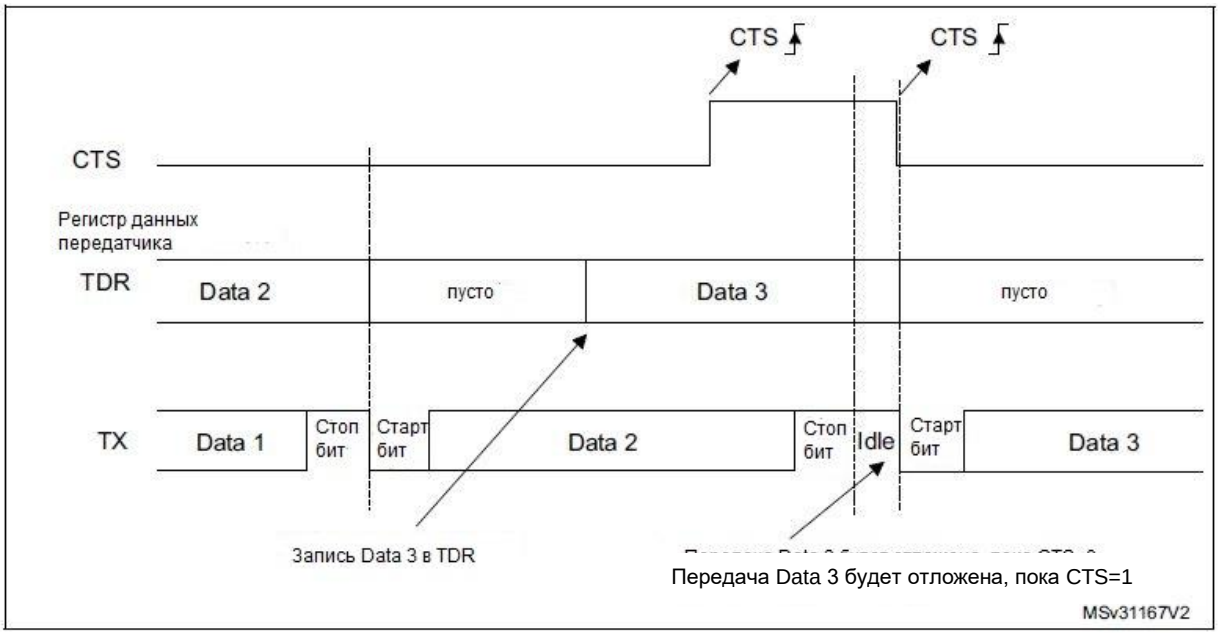


Рисунок 122 – Управление потоком CTS

Примечание – Особое поведение фреймов break: при разрешенном управлении потоком CTS, передатчик не проверяет статус входа CTS для отправки сигнала break

31.4 **Прерывания USART**

Таблица 421 – Запрос на прерывание USART

Событие	Флаг события	Бит разрешения
Регистр данных передачи пуст	TXE	TXEIE
Флаг CTS	CTS	CTSIE
Передача завершена	TC	TCIE
Принятые данные готовы для чтения	RXNE	RXNEIE
Обнаружена ошибка переполнения	ORE	
Обнаружена линия Idle	IDLE	IDLEIE
Ошибка четности	PE	PEIE

События прерываний подсоединены к одному и тому же вектору прерывания (см. Рисунок 123).

– Во время передачи: Передача завершена (Transmission Complete), Линия пуста для начала передачи (Clear to Send) или Регистр данных передачи пуст (Transmit Data Register Empty).

– Во время приема: Обнаружение линии Idle, ошибка переполнения, Регистр данных приема не пуст (Receive Data register not empty), Ошибка четности.

Все эти события генерируют прерывания, если соответствующий бит разрешения прерывания установлен.

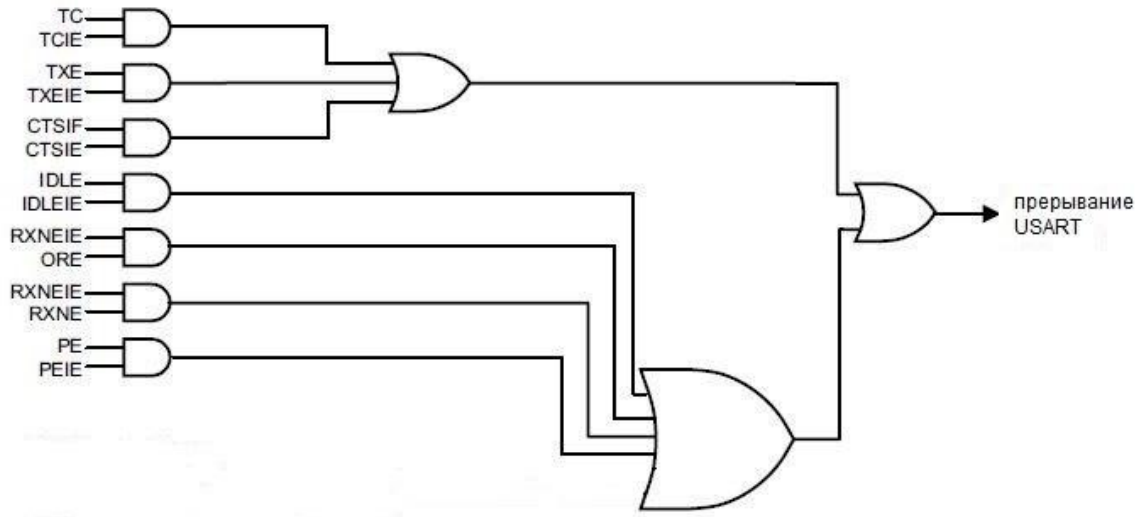


Рисунок 123 – Диаграмма отображений событий USART на прерывание

31.5 **Регистры USART**

Доступ к регистрам периферии осуществляется по 16-битным полусловам или 32-битными словам. Таблицы 423 – 435 показывают карту памяти регистров USART и их значения после сброса.

Таблица 422 – Регистры блока USART_CNTR

Базовый Адрес		Название	Описание
0x500C_0000		USART_CNTR	Блок контроллера USART (ISO7816)
Смещение			
0x0000	0	USART_SR	Регистр статуса
0x0004	1	USART_DR	Регистр данных
0x0008	2	USART_BRR	Регистр настройки скорости передачи данных
0x000c	3	USART_CR1	Регистр управления 1
0x0010	4	USART_CR2	Регистр управления 2
0x0014	5	USART_CR3	Регистр управления 3
0x0018	6	USART_GTPR	Регистр защищенного интервала предделителя

31.5.1 **USART_SR**

Таблица 423 –Регистр USART_SR

Base ADDR=			0x500C_0000			Offset=		0x0000_0000			Reset=		0x0000_00C0		
REG Name:			USART_SR												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

.-						CTS	-	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						RW		R	RW	RW	R	R	R	R	R

Таблица 424 – Описание бит регистра USART_SR

Бит	Имя	Значение	Описание
31…10	-	0	Зарезервировано
9	CTS	0	Флаг CTS. Данный бит устанавливается аппаратно, когда на входе CTS меняется уровень, если установлен бит CTSE. Бит сбрасывается программно записью 1. Прерывание генерируется, если CTSIE=1 в регистре USART_CR3. 0 – не было изменения статуса линии CTS; 1 – на линии CTS обнаружено изменение уровня
8	-	0	Зарезервировано
7	TXE	1	Регистр данных передатчика пуст. Данный бит устанавливается аппаратно, когда содержимое регистра TDR было перемещено в регистр смещения. Прерывание генерируется, если TXEIE=1 в регистре USART_CR1. Бит сбрасывается записью регистра USART_DR. 0 – данные не были переданы в регистр сдвига; 1 – данные переданы в регистр сдвига
6	TC	1	Передача завершена. Данный бит устанавливается аппаратно, когда передача фрейма, содержащего данные, завершена, и бит TXE установлен. Прерывание генерируется, если TCIE=1 в регистре USART_CR1. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем запись регистра USART_DR). Бит TC может быть также сброшен записью 1. 0 – передача не завершена; 1 – передача завершена
5	RXNE	0	Регистр данных приема не пуст. Данный бит устанавливается аппаратно, когда содержимое регистра сдвига RDR было передано в регистр USART_DR. Прерывание генерируется, если RXNEIE=1 в регистре USART_CR1. Бит сбрасывается чтением регистра USART_DR. Флаг RXNE может быть также сброшен записью 1. 0 – данные не получены; 1 – полученные данные готовы для чтения

Бит	Имя	Значение	Описание
4	IDLE	0	Обнаружение линии IDLE (ожидание). Данный бит устанавливается аппаратно, когда обнаружена линия Idle. Прерывание генерируется, если IDLEIE=1 в регистре USART_CR1. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем чтения регистра USART_DR). 0 – не обнаружена линия Idle; 1 – линия Idle обнаружена. Примечание – Бит IDLE не будет снова установлен, пока не установится бит RXNE (то есть пока не будет обнаружено состояние ожидания на линии)
3	ORE	0	Ошибка переполнения. Данный бит устанавливается аппаратно, когда принятое в настоящий момент слово в регистре сдвига готово к передаче в регистр RDR, если RXNE=1. Прерывание генерируется, если RXNEIE=1 в регистре USART_CR1. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем чтение регистра USART_DR). 0 – нет ошибки переполнения; 1 – обнаружена ошибка переполнения. Примечание – Когда установлен этот бит, содержимое регистра RDR не будет потеряно, но регистр сдвига будет перезаписан
2	NF	0	Флаг обнаружения шума. Данный бит устанавливается аппаратно, когда в принятом фрейме обнаружен шум. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем чтение регистра USART_DR). 0 – шум не обнаружен; 1 – шум обнаружен. Примечания 1 Данный бит не генерирует прерывание, так как он появляется одновременно с битом RXNE, который сам генерирует прерывание. 2 Если шумов на линии нет, флаг NF может быть отключен установкой бита ONEBIT в 1 для увеличения допуска на отклонения приемопередатчика (см. п. 31.3.5«Допуск ухода тактовой частоты для приемника USART»)

Бит	Имя	Значение	Описание
1	FE	0	Ошибка фрейма. Данный бит устанавливается аппаратно, когда произошла рассинхронизация, чрезмерный шум или был получен символ break. Бит сбрасывается программной последовательностью (чтение регистра USART_SR, затем чтение регистра USART_DR). 0 – ошибки фрейма не обнаружено; 1 – обнаружена ошибка фрейма или символ break. Примечания 1 Данный бит не генерирует прерывание, так как он появляется одновременно с битом RXNE, который сам генерирует прерывание. 2 Если принимаемое в данный момент слово вызывает как ошибку фрейма (FE), так и ошибку переполнения (ORE), то будет установлен только бит ORE
0	PE	0	Ошибка четности. Данный бит устанавливается аппаратно, когда в режиме приема обнаружена ошибка четности. Бит сбрасывается программной последовательностью (чтение регистра статуса, затем чтение или запись регистра данных USART_DR). Программа должна дождаться, пока установится флаг RXNE, перед тем как сбрасывать бит PE. Прерывание генерируется, если PEIE = 1 в регистре USART_CR1. Бит PE устанавливается одновременно со всеми остальными флагами RX (RXNE, ORE, NF, FE), поэтому данный бит может сгенерировать запрос прерывания только по окончании приёма. 0 – нет ошибки четности; 1 – обнаружена ошибка четности.

31.5.2 **USART_DR**

Таблица 425 – Регистр USART_DR

Base ADDR=			0x500C_0000			Offset=		0x0000_0004			Reset=		0xXXXX_XXXX		
REG Name:			USART_DR												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8…0
-							DR
							RW

Таблица 426 – Описание бит регистра USART_DR

Бит	Имя	Значение	Описание
31…9	-	0	Зарезервировано
8…0	DR [8:0]	X	Значение данных. Содержит принятый или передаваемый символ данных в зависимости от операции с регистром – чтение или запись. Регистр данных выполняет двойную функцию (чтение и запись), так как состоит из двух регистров, один для передачи данных (TDR), другой для приема данных (RDR). Регистр TDR обеспечивает параллельный интерфейс между внутренней шиной и выходным регистром сдвига. Регистр RDR обеспечивает параллельный интерфейс между входным регистром сдвига и внутренней шиной. Когда разрешена передача с генерацией бита четности (бит PCE установлен в 1 в регистре USART_CR1), значение, записываемое в MSB (бит 7 или бит 8 в зависимости от длины данных) не играет значения, поскольку он заменяется битов четности. Когда на приеме разрешен контроль четности, значение, считываемое из MSB, является принятым битом четности

31.5.3 **USART_BRR**

Примечание – Счетчики скорости останавливают счет, если биты TE или RE запрещены, соответственно.

Таблица 427 –Регистр USART_BRR

Base ADDR=			0x500C_0000			Offset=		0x0000_0008			Reset=		0x0000_0000		
REG Name:			USART_BRR												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Mantissa[11:0]												DIV_Fraction[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Таблица 428 – Описание бит регистра USART_BRR

Бит	Имя	Значение	Описание
31…16	-	0	Зарезервировано
15…4	DIV_Mantissa [11:0]	0	Мантисса USARTDIV. Эти 12 бит определяют мантиссу делителя USART (USARTDIV)
3…0	DIV_Fraction [3:0]	0	Дробная часть USARTDIV. Эти 4 бита определяют дробную часть делителя USART (USARTDIV). Если OVER8=1, бит DIV_Fraction3 не учитывается и должен быть сброшен

31.5.4 **USART_CR1**

Таблица 429 –Регистр USART_CR1

Base ADDR=			0x500C_0000			Offset=		0x0000_000C			Reset=		0x0000_0000		
REG Name:			USART_CR1												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	-	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Таблица 430 – Описание бит регистра USART_CR1

Бит	Имя	Значение	Описание
31…16	-	0	Зарезервировано
15	OVER8	0	Режим передискретизации: 0 – передискретизация на 16; 1 – передискретизация на 8. Примечание – Передискретизация на 8 недоступна в режиме Smartcard: если SCEN=1, тогда OVER8 принудительно сбрасывается аппаратурой в 0
14	-	0	Зарезервировано
13	UE	0	Разрешение USART. Сброс данного бита в 0 моментально отключает TX, RX и генератор CK. При этом текущие передачи по TX и RX обрываются (если они идут). Данный бит устанавливается и сбрасывается программно. 0 – предделитель и выходы USART запрещены; 1 – USART разрешен
12	M	0	Длина слова. Данный бит определяет длину слова. Данный бит устанавливается и сбрасывается программно. 0 – 1 стартовый бит, 8 бит данных, n стоп-бит; 1 – 1 стартовый бит, 9 бит данных, n стоп-бит. Примечание – Бит M не должен быть модифицирован во время передачи данных (и приема и передачи)
11	WAKE	0	Метод пробуждения. Данный бит определяет метод пробуждения USART. Данный бит устанавливается и сбрасывается программно. 0 – линия Idle (состояние ожидания на линии); 1 – метка адреса

Бит	Имя	Значение	Описание
10	PCE	0	Разрешение контроля четности. Данный бит выбирает аппаратную генерацию и обнаружение бита четности. Если разрешен контроль бита четности, вычисленное значение четности вставляется на место MSB (9 бит, если M=1; 8 бит, если M=0) и на приеме этот бит проверяется на четность. Данный бит устанавливается и сбрасывается программно. Как только бит установлен, PCE активируется после текущего байта (при приеме и при передаче): 0 – контроль четности запрещен; 1 – контроль четности разрешен
9	PS	0	Выбор четности/нечетности. Данный бит выбирает, как вычисляется и проверяется бит четности – либо на четность, либо на нечетность (бит PCE установлен). Данный бит устанавливается и сбрасывается программно. Вариант контроля четности/нечетности выбирается после текущего байта. 0 – контроль на четность; 1 – контроль на нечетность
8	PEIE	0	Разрешение прерывание по событию PE. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при PE=1 в регистре USART_SR
7	TXEIE	0	Разрешение прерывание по событию TXE. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при TXE=1 в регистре USART_SR
6	TCIE	0	Разрешение прерывания по завершению передачи. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при TC=1 в регистре USART_SR
5	RXNEIE	0	Разрешение прерывания по событию RXNE. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при ORE=1 или RXNE=1 в регистре USART_SR
4	IDLEIE	0	Разрешение прерывания по событию IDLE. Данный бит устанавливается и сбрасывается программно: 0 – прерывание запрещено; 1 – прерывание генерируется при IDLE=1 в регистре USART_SR

Бит	Имя	Значение	Описание
3	TE	0	Разрешение передатчика. Данные бит разрешает передатчик. Данный бит устанавливается и сбрасывается программно: 0 – передатчик запрещен; 1 – передатчик разрешен. Примечание – Во время передачи, нулевой импульс в бите TE (“0”, за которым следует “1”) передает преамбулу (сигнала ожидания на линии) после текущего слова, кроме режима smartcard. Когда установлен бит TE, существует 1-битная задержка перед началом передачи
2	RE	0	Разрешение приемника. Данный бит разрешает приемник. Данный бит устанавливается и сбрасывается программно: 0 – приемник запрещен; 1 – приемник разрешен, и он начинает искать стартовый бит
1	RWU	0	Пробуждение приемника. Данный бит определяет, находится ли приемопередатчик в режиме молчания или нет. Данный бит устанавливается и сбрасывается программно, а также может быть сброшен аппаратно при распознавании последовательности пробуждения. 0 – приемник в активном режиме; 1 – приемник в режиме молчания. Примечание – Перед выбором режима молчания (Mute) (установка бита RWU) приемопередатчик USART должен сначала получить байт данных, иначе он не сможет работать в режиме молчания с пробуждением по обнаружению сигнала ожидания на линии. При конфигурации пробуждения по метке адреса (WAKE=1) бит RWU не может быть модифицирован программно, пока установлен бит RXNE
0	SBK	0	Отправка сигнала break. Данный бит используется для отправки символов break. Данный бит устанавливается и сбрасывается программно. Он должен быть установлен программой и будет сброшен аппаратно во время стопового бита или символа break. 0 – символ break не передается; 1 – символ Break передается

31.5.5 **USART_CR2**

Таблица 431 – Регистр USART_CR2

Base ADDR=			0x500C_0000			Offset=		0x0000_0010			Reset=		0x0000_0000		
REG Name:			USART_CR2												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	-				ADD [3:0]			
		RW	RW	RW	RW	RW	RW					RW	RW	RW	RW

Таблица 432 – Описание бит регистра USART_CR2

Бит	Имя	Значение	Описание
31…14	-	0	Зарезервировано
13...12	STOP [1:0]	0	Стоповые биты. Данные биты используются для программирования стоп-бит: 00 – зарезервировано; 01 – 0,5 стоп-бит; 10 – 2 стоп-бита; 11 – 1,5 стоп-бита
11	CLKEN	0	Разрешение тактирования. Данный бит позволяет разрешить работу вывода CK: 0 – вывод CK запрещен; 1 – вывод CK разрешен
10	CPOL	0	Полярность тактов. Данный бит позволяет выбрать полярность выхода тактов на выводе CK в синхронном режиме. Совместно с битом CPHA данный бит определяет взаимосвязь тактов и данных. 0 – постоянное значение 0 на выводе CK вне окна передачи; 1 – постоянное значение 1 на выводе CK вне окна передачи
9	CPHA	0	Фаза тактов. Данный бит позволяет выбрать фазу выхода тактов на выводе CK в синхронном режиме. Совместно с битом CPOL данный бит определяет взаимосвязь тактов и данных. 0 – первое изменение сигнала синхронизации является первым перепадом захвата данных; 1 – второе изменение сигнала синхронизации является первым перепадом захвата данных

Бит	Имя	Значение	Описание
8	LBCL	0	Тактовый импульс последнего бита. Бит позволяет выбрать, будет ли тактовый импульс последнего переданного бита (MSB) выдаваться на вывод CK в синхронном режиме. 0 – тактовый импульс последнего бита данных не выводится на вывод CK; 1 – тактовый импульс последнего бита данных выводится на вывод CK. Примечание – 1: последний бит – это 8 или 9 бит данных, в зависимости от выбранного формата, задаваемого битом M в регистре USART_CR1
7…4	-	0	Зарезервировано
3…0	ADD [3:0]	0	Адреса узла USART. Данное битовое поле указывает адрес режима USART. Данные биты используются в многопроцессорном обмене во время режима молчания, для пробуждения приемопередатчика при обнаружении метки адреса. Примечание – Бит CPOL, CPHA и LBCL не должны быть записаны, пока разрешен передатчик

31.5.6 **USART_CR3**

Таблица 433 – Регистр USART_CR3

Base ADDR=			0x500C_0000			Offset=		0x0000_0014			Reset=		0x0000_0000		
REG Name:			USART_CR3												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				ONEBIT	CTSIE	CTSE	RTSE	-		SCEN	NACK	HDSEL	-		-
				RW	RW	RW	RW	RW	RW	RW	RW	RW			RW

Таблица 434 – Описание бит регистра USART_CR3

Бит	Имя	Значение	Описание
31…12	-	0	Зарезервировано
11	ONEBIT	0	Разрешение режима одной выборки на бит. Бит позволяет выбрать метод оцифровки сигнала: 0 – метод трех выборок; 1 – метод одной выборки. Когда выбран метод одной выборки, флаг обнаружения шума (NF) запрещен. Примечание – функция ONEBIT применима только к битам данных. Это не относится к стартовому биту

Бит	Имя	Значение	Описание
10	CTSIE	0	Разрешение прерывания CTS: 0 – прерывание запрещено; 1 – прерывание генерируется каждый раз при CTS=1 в регистре USART_SR
9	CTSE	0	Разрешение CTS: 0 – запрещено аппаратное управление потоком CTS; 1 – режим CTS разрешен, данные передаются только, когда вход CTS имеет логический «0». Если вход CTS установлен на логическую «1» во время передачи данных, тогда передача завершается до остановки. Если данные записаны в регистр данных, когда CTS в логической «1», то передача будет отложена до тех пор, пока на входе CTS не появится логический «0»
8	RTSE	0	Разрешение RTS: 0 – запрещено аппаратное управление потоком RTS; 1 – прерывание RTS разрешено, данные запрашиваются, когда есть место в буфере приемника. Ожидается, что передача данных прекратится после того, как будет передан последний текущий символ данных. Выход RTS имеет логический «0», когда приемопередатчик может принимать данные
7, 6	-	0	Записывать всегда 0
5	SCEN	0	Разрешение режима Smartcard. Данный бит используется для разрешения режима Smartcard. 0 – режим Smartcard запрещен; 1 – режим Smartcard разрешен
4	NACK	0	Разрешение NACK в режиме Smartcard: 0 – ответ NACK в случае ошибки четности запрещен; 1 – ответ NACK в случае ошибки четности разрешен
3	HDSEL	0	Выбор полудуплексного режима. Бит позволяет выбрать однопроводной режим или полудуплексный режим: 0 – полудуплексный режим не выбран; 1 – полудуплексный режим выбран
2, 1	-	0	Зарезервировано
0	-	0	Записывать всегда 0

31.5.7 **USART_GTPR**

Таблица 435 – Регистр USART_GTPR

Base ADDR=			0x500C_0000			Offset=		0x0000_0018			Reset=		0x0000_0000		
REG Name:			USART_GTPR												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT [7:0]								-			PSC [4:0]				

RW	RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW

Таблица 436 – Описание бит регистра USART_GTPR

Бит	Имя	Значение	Описание
31…16	-	0	Зарезервировано
15…8	GT[7:0]	0	Значение защитного интервала. Это битовое поле показывает значение защитного интервала времени в единицах тактов скорости. Биты используются в режиме Smartcard. Флаг завершения передачи устанавливается после этого значения защитного интервала
4…0	PSC[4:0]	0	Значение предделителя в режиме smartcard. Используется для программирования предделителя для деления системной тактовой частоты, чтобы обеспечить тактирование смарт-карты. Значение, заданное в регистре (5 значащих бит), умножается на 2, чтобы получить коэффициент деления исходной тактовой частоты: 00000: зарезервировано – не программировать это значение; 00001: делит источник тактов на 2; 00010: делит источник тактов на 4; 00011: делит источник тактов на 6

32 Электрические параметры микросхем

Таблица 437 – 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 выходах портов PA, PB, PC, PD	U_{OH}	U_{CC} – 0,6	–	25, 125, – 60
Выходное напряжение низкого уровня, В, на выходах портов: PA, PB, PC, PD при:	U_{OL}			25, 125, – 60
– $U_{CC} \geq 3{,}0$ В;		–	0,4	
– $U_{CC} < 3{,}0$ В		–	0,5	
Ток утечки низкого уровня на входе, мкА: на входах портов PA, PB, PC, PD, nRESET, SAR_AIN2	I_{ILL}	– 1,0	1,0	25, 125, – 60
Ток утечки высокого уровня на входе, мкА:	I_{ILH}			25, 125, – 60
– на входах портов PA, PB, PC, PD, nRESET, SAR_AIN2;		– 1,0	1,0	
– на входе OSC_IN		1	20	
Динамический ток потребления по питанию Ucc, мА, при $f_C \leq 50$ МГц	I_{CCO_UCC}	–	20	25, 125, – 60
Динамический ток потребления по питанию AUcc, мА, при $f_C \leq 50$ МГц	I_{CCO_AUCC}	–	7	25, 125, – 60
Динамический ток потребления по питанию BUcc, мкА, при $f_C \leq 50$ МГц	I_{CCO_BUCC}	–	10	25, 125, – 60
Выходная частота HSI RC-генератора, МГц, после выполнение тримминга	f_{O_HSI}	7,5	8,5	25, 125, – 60
Выходная частота LSI RC-генератора, кГц, после выполнение тримминга	f_{O_LSI}	29,3	33,2	
Выходная частота PLL, МГц:	f_{O_PLL}			
– максимальная;		50	–	
– минимальная		–	8	
Параметры ΔΣ АЦП				
Соотношение сигнал/шум, дБ, усиление + 6 дБ, f_{NADCO}[1)] = 4 кГц	SNR_{D6}	82	–	25, 125, – 60

[1)] f_{NADCO} – частота передискретизации на выходе микросхем

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В.

33 Предельно-допустимые и предельные параметры

Таблица 438 – Предельно-допустимые электрические режимы эксплуатации и предельные электрические режимы микросхем

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра			
		предельно-допустимый режим		предельный режим	
		не менее	не более	не менее	не более
Напряжение питания, В, на выводах V_{CC}	U_{CC}	1,8	3,6	–	3,9
Напряжение питания аналоговое, В, на выводах V_{CCA}	U_{CCA}	1,8	3,6	–	3,9
– при использовании АЦП		3,0	3,6	–	3,9
Входное напряжение высокого уровня, В, на входах портов PA, PB, PC, PD при: – $U_{CC} \geq 3{,}0$ В; – $U_{CC} < 3{,}0$ В	U_{IH}	2,0 $0{,}65 \cdot U_{CC}$	U_{CC}	–	$U_{CC} + 0{,}3$
Входное напряжение низкого уровня, В, на входах портов PA, PB, PC, PD при: – $U_{CC} \geq 3{,}0$ В; – $U_{CC} < 3{,}0$ В	U_{IL}		0,8 0,55	– 0,3	–
Входное напряжение высокого уровня, В, на входе OSC_IN при HSE_BYP = 1	$U_{IH_OSC_IN}$	$U_{CC} - 0{,}6$	U_{CC}	–	$U_{CC} + 0{,}3$
Входное напряжение низкого уровня, В, на входе OSC_IN при HSE_BYP = 1	$U_{IL_OSC_IN}$	0	0,4	–0,3	–
Выходной ток высокого уровня, мА: на выходах портов PA, PB, PC, PD	I_{OH}	– 4,0	–	– 6,0	–
Выходной ток низкого уровня, мА, на выходах портов PA, PB, PC, PD	I_{OL}	–	4,0	–	6,0
Выходной ток высокого уровня на выводе OSC_OUT, мА	$I_{OH_OSC_OUT}$	– 0,3	–	– 1	–
Выходной ток низкого уровня на выводе OSC_OUT, мА	$I_{OL_OSC_OUT}$	–	0,3	–	1
Тактовая частота ядра микроконтроллера, МГц	f_C	–	50	–	–
Частота следования импульсов тактовых сигналов PLL, МГц	f_{C_PLL}	8	16	–	–
Частота следования импульсов тактовых сигналов HSE, МГц, на входе OSC_IN при: – BYPASS = «0»;	f_{C_HSE}	8	16	–	–
– BYPASS = «1»		–	50	–	–
Емкость нагрузки, пФ, на выходах портов PA, PB, PC, PD	C_L	–	30	–	–

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра			
		предельно-допустимый режим		предельный режим	
		не менее	не более	не менее	не более
Параметры ΔΣ АЦП					
Опорное напряжение, В, на выводе VREF, при выключенном внутреннем источнике опорного напряжения	U_{REF_ADC}	2,33	2,53	–	–
Входное напряжение дифференциальное, В, на входах I0P – I3P, I0N – I3N	U_{ID_ADCD}	– 1,0	1,0	–	–
Напряжение, В, на входах I0P – I3P, I0N – I3N	U_{I_ADCD}	– 0,5	0,5	– 0,8	U_{CC} + 0,3
Частота следования импульсов тактовых сигналов ΔΣ АЦП, МГц	f_{C_ADCD}	0,9	4,1	–	–
Параметры SAR АЦП					
Входное напряжение SAR АЦП, В, на входе SAR_AIN2 при: − использовании внутреннего источника опорного напряжения; − подаче опорного напряжения на вывод VREF; − без использования опорного напряжения	U_{AIN}	0	U_{REF_ADC}	– 0,3	3,9
		0	U_{CCA}		
Частота следования импульсов тактовых сигналов SAR АЦП, МГц	f_{C_ADCS}	–	28	–	–
Примечание – Не допускается одновременное задание двух предельных режимов					

34 Справочные параметры

Таблица 439 – Справочные параметры микросхем

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра			Температура среды, °С
		не менее	типовое	не более	
Статический ток потребления, мА	I_{CC}	–	0,6	–	25, 125, – 60
Статический ток потребления по питанию аналоговой части, мА	I_{CCA}	–	0,13	–	25, 125, – 60
Время хранения информации в Flash-памяти, лет	t_{SG}	25	–	–	25
		10	–	–	125
Сопротивление внутренних программируемых резисторов доопределения к шине общий, кОм	R_{PD}		76		
Сопротивление внутренних программируемых резисторов доопределения к шине питание, кОм	R_{PU}		76		
Параметры SAR АЦП					
Разрядность АЦП, бит	E_{NADC}	10	–	–	25, 125, – 60
Дифференциальная нелинейность, ЕМР [1], f_{C_ADCS} = 15 МГц, $U_{CC} = U_{CCA} = U_{CCB}$ = 3,0 В	E_{DLADC}				
– максимальная		–	0,26	1,00	
– минимальная		– 1,00	– 0,6	–	
Интегральная нелинейность, ЕМР [1], f_{C_ADCS} = 15 МГц, $U_{CC} = U_{CCA} = U_{CCB}$ = 3,0 В	E_{ILADC}				
– максимальная		–	0,4	1,0	
– минимальная		– 1,0	– 0,4	–	
Ошибка смещения, ЕМР [1], f_{C_ADCS} = 15 МГц, $U_{CC} = U_{CCA} = U_{CCB}$ = 3,0 В	E_{OFFADC}	– 1,00	– 0,25	1,00	
Ошибка усиления, %, f_{C_ADCS} = 15 МГц, $U_{CC} = U_{CCA} = U_{CCB}$ = 3,0 В	$E_{GAINADC}$	– 0,10	– 0,05	0,10	
Параметры SPI					
Частота синхросигнала SSP_CLK, МГц,	f_{OP}				25, 125, – 60
– в режиме master;		–	–	$0{,}5 \cdot f_C$	
– в режиме slave		–	–	$0{,}25 \cdot f_C$	
Длительность сигнала высокого/низкого уровня синхросигнала SSP_CLK, нс	t_{WSSP_CLK}	$\frac{1}{2 \cdot f_{OP}}$	–	–	
Время задержки распространения данных при переходе выхода SSP_TXD из состояния «Выключено» в состояние высокого/низкого уровня по сигналу SSP_FSS, нс	$t_{PZH(SSP_CLK-SSP_TXD)}$ $t_{PZL(SSP_CLK-SSP_TXD)}$	$\frac{1}{2 \cdot f_{OP}}$	–	–	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра			Температура среды, °С
		не менее	типовое	не более	
Время задержки распространения данных при переходе выхода SSP_TXD из состояния высокого /низкого уровня в состояние «Выключено» по сигналу SSP_FSS, нс	$t_{PHZ(SSP_CLK-SSP_TXD)}$ $t_{PLZ(SSP_CLK-SSP_TXD)}$	$\frac{1}{2 \cdot f_{OP}}$	–	–	25, 125, – 60
Время установления входных данных SSP_RXD относительно SSP_CLK, нс	$t_{SU(SSP_RXD–SSP_CLK)}$	2,5	–	–	
Время удержания сигнала входных данных SSP_RXD относительно SSP_CLK, нс	$t_{H(SSP_CLK–SSP_RXD)}$	3,5	–	–	
Время сохранения выходных данных SSP_TXD относительно сигнала SSP_CLK, нс	$t_{V(SSP_CLK-SSP_TXD)}$	0	–	–	
Параметры ΔΣ АЦП					
Напряжение внутреннего источника опорного напряжения, В, на выводе VREF	U_{REFO}	–	2,45	–	25, 125, – 60
Соотношение сигнал / шум, дБ, усиление + 6 дБ, U_{ID_ADCD} = +/-1,0 В, f_{NADCO} = 4кГц	SNR_{D6}	–	91	–	
Соотношение сигнал / шум, дБ усиление + 12 дБ, U_{ID_ADCD} = +/-0,5 В, f_{NADCO} = 4кГц	SNR_{D12}	–	88	–	
Соотношение сигнал / шум, дБ усиление + 18 дБ, U_{ID_ADCD} = +/-0,25 В, f_{NADCO} = 4кГц	SNR_{D18}	–	85	–	
Соотношение сигнал / шум, дБ усиление + 24 дБ, U_{ID_ADCD} = +/-0,125 В, f_{NADCO} = 4кГц	SNR_{D24}	–	82	–	
Ошибка усиления предусилителя, дБ	$GAIN_{ERR}$	– 2,0	– 0,5	2,0	

[1)] EMP, единица младшего разряда – величина дифференциального напряжения, соответствующая для заданного опорного напряжения (U_{REF_ADC}) изменению напряжения на входе АЦП, требующемуся для изменения выходного кода на одну единицу младшего разряда.

1 EMP $= \frac{U_{REF_ADC}}{2^{10}}$ для 10-разрядного идеального АЦП

34.1 Динамические характеристики ΔΣ АЦП

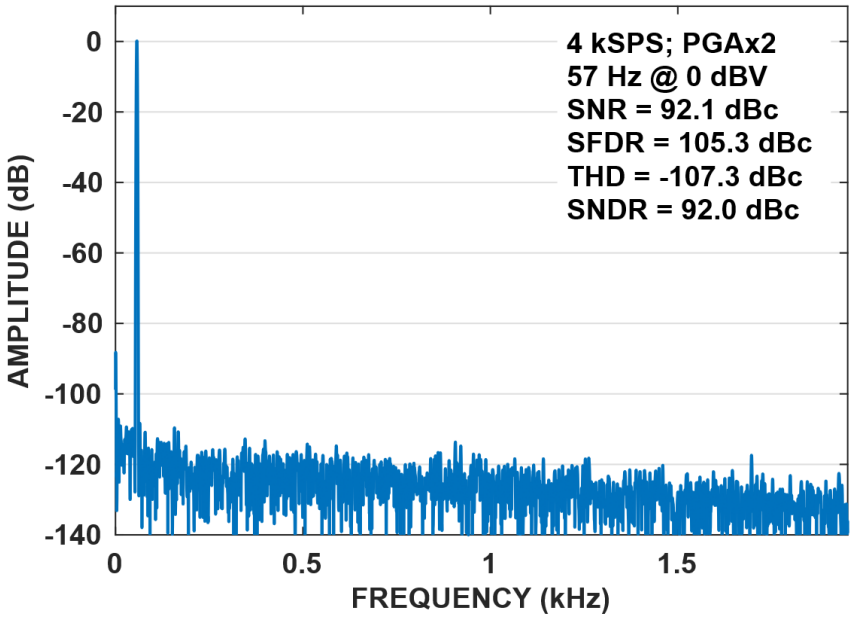


Рисунок 124 – Типовой спектр в НУ для ΔΣ АПЦ
в канале F2IDAT для дифференциального входного сигнала +/- 1 В (0.707 RMS).
Усиление = 6 дБ (F2IGAIN = 00)

34.2 Производительность в тестах

Таблица 440 – Производительность в криптографии

ГОСТ Р 34.11-2012		ГОСТ Р 34.12-2015				AES128	
Стрибог-512		Кузнечик		Магма			
МБ/с	Тактов/байт	МБ/с	Тактов/байт	МБ/с	Тактов/байт	МБ/с	Тактов/байт
0,90	53,3	3,31	14,5	5,48	8,8	3,84	12,5

Таблица 441 – Производительность в Coremark

Выполнение кода из FLASH		Выполнение кода из TCM
Delay[2:0] = 0b001	Delay[2:0] = 0b000	
2,15	3,4	3,7

34.3 Характеристики выводов микросхемы

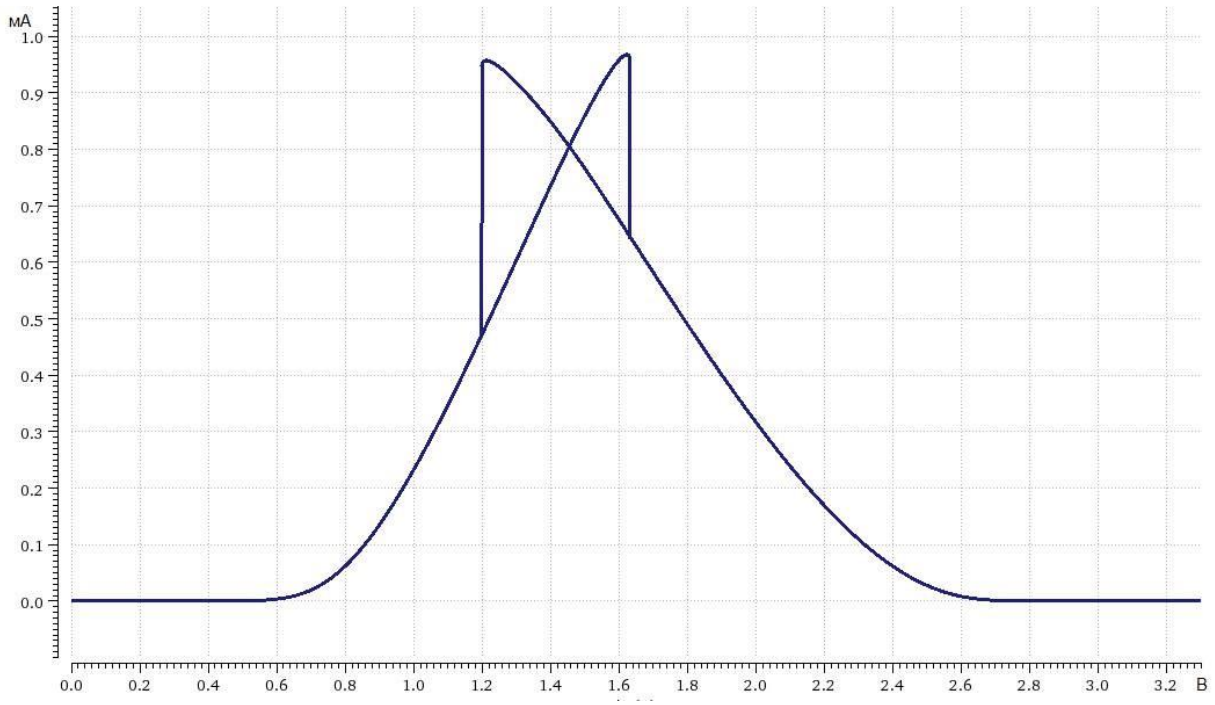


Рисунок 125 – Типовая зависимость дополнительного тока потребления по питанию U_{CC} от напряжения на цифровом входе порта ввода-вывода при U_{CC} = 3,3 В в НУ

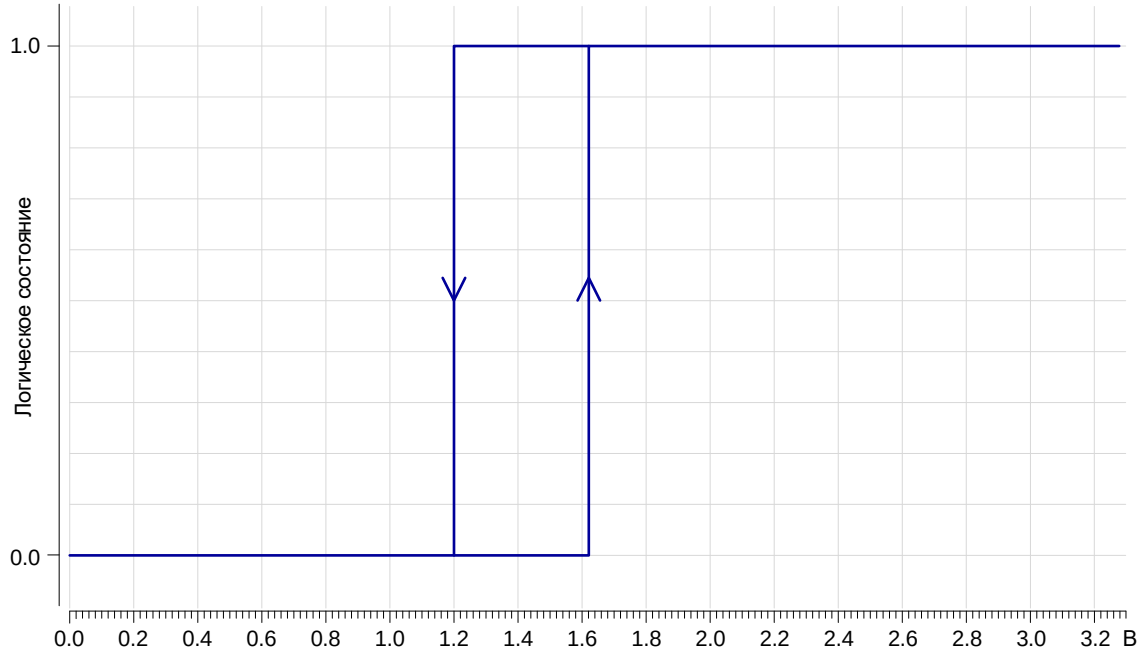


Рисунок 126 – Типовая зависимость логического состояния от напряжения на цифровом входе порта ввода-вывода при U_{CC} = 3,3 В в НУ

35 Типовые зависимости

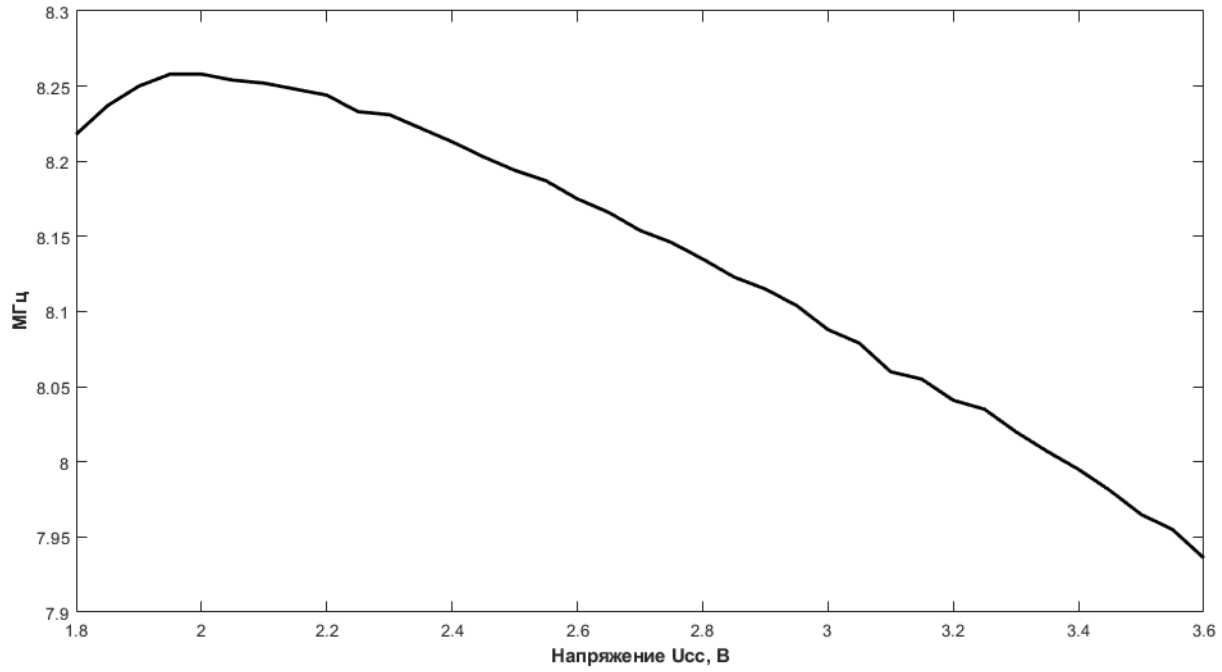


Рисунок 127 – Зависимость частоты RC генератора HSI
от напряжения питания U_{CC} в НУ

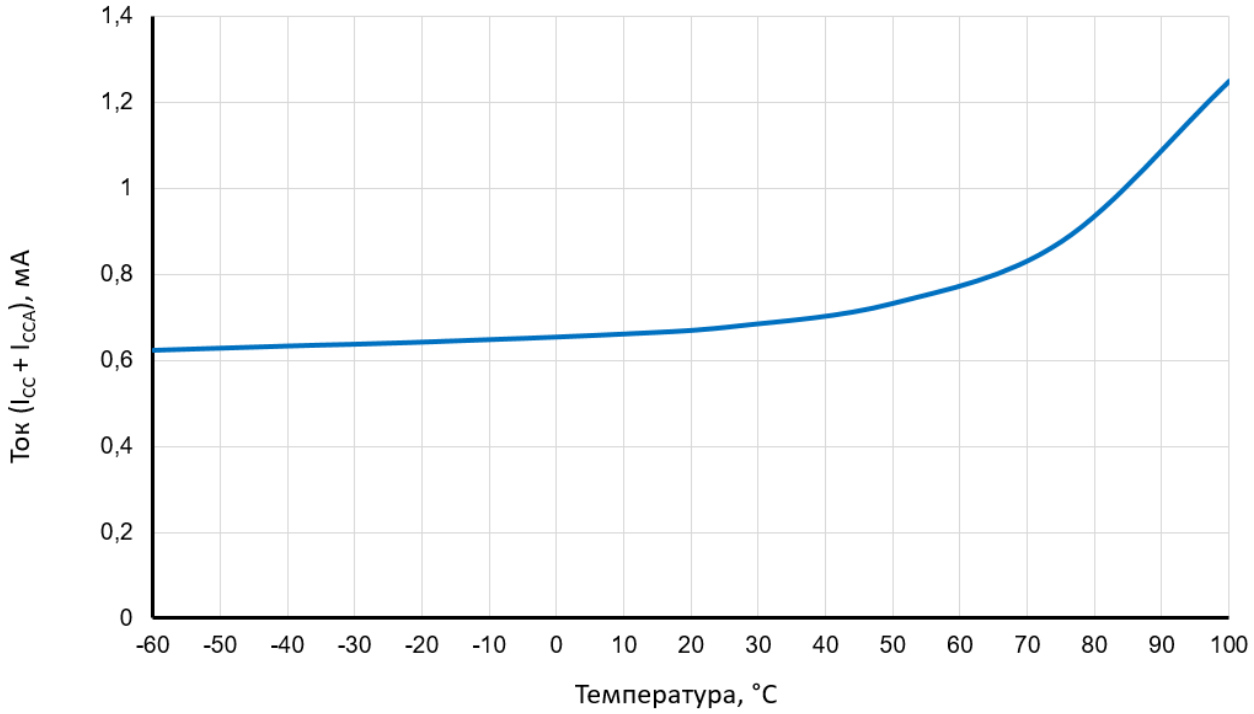


Рисунок 128 – Зависимость суммарного статического тока (nRESET = 0) I_{CC} и I_{CCA}
от температуры при $U_{CC} = U_{CCA} = 3{,}3$ В

36 Габаритный чертеж микросхемы

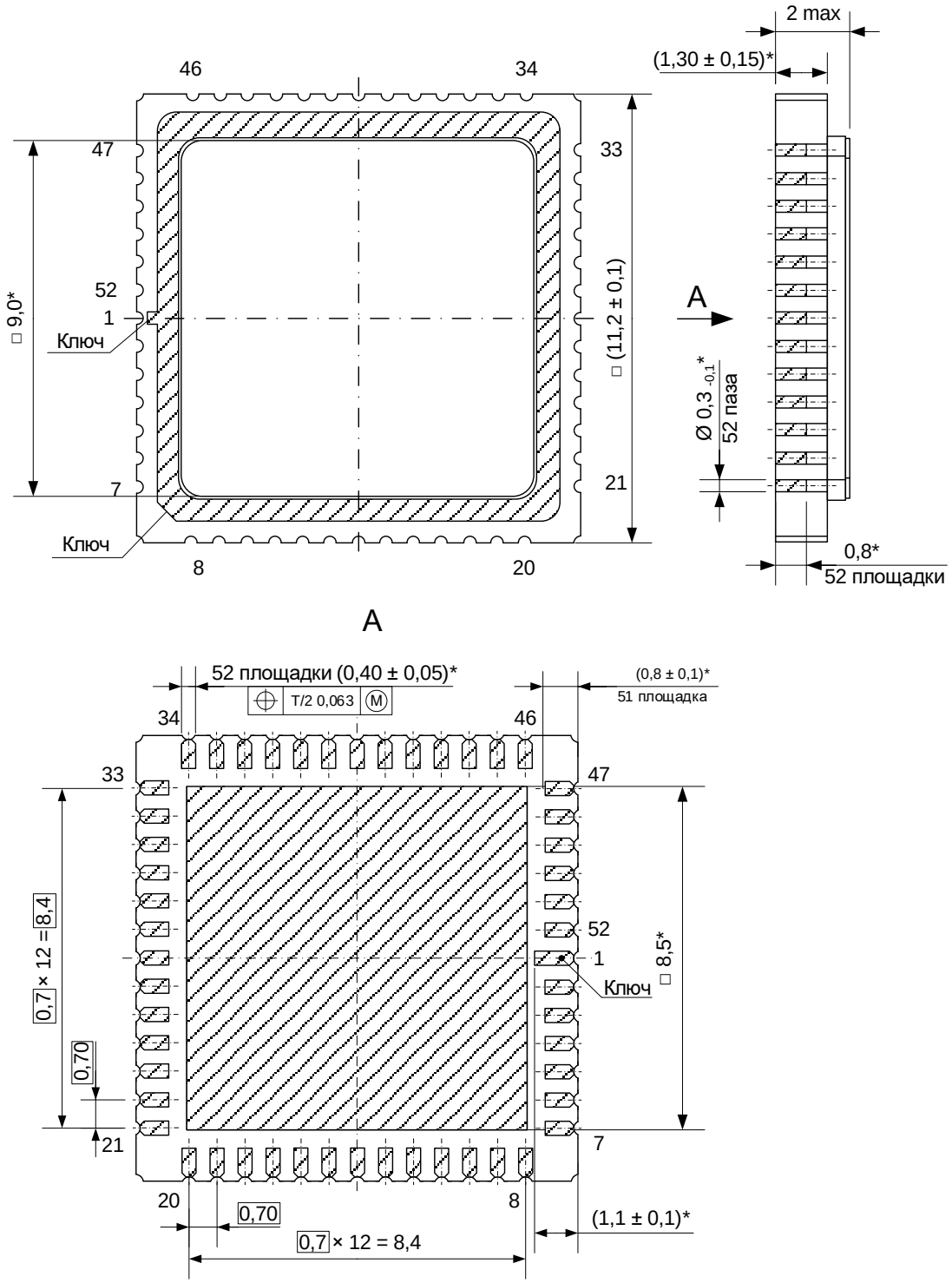


Рисунок 129 – Микросхема в корпусе 5152.52-3 К

37 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон, °С
MDR12065	12065 RISC-V	5152.52-3 К	от – 60 до 125

Условное обозначение микросхем при заказе в договоре на поставку и в конструкторской документации другой продукции должно состоять из:

- наименование изделия – микросхема;
- обозначения типа (типономинала);
- обозначения технических условий ТСКЯ.431000.002ТУ;
- обозначения спецификации ТСКЯ.431296.051СП.

Пример обозначения микросхем:
Микросхема MDR12065 – ТСКЯ.431000.002ТУ, ТСКЯ.431296.051СП.

Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	26.03.2026	0.1.0	Введена впервые	
2	03.08.2026	0.1.1	Таблица 48 – скорректировано описание F0I3NTEN, F0I0NTEN; Таблица 60 – скорректировано описание F1INTEN; Таблица 68 – скорректировано описание F2INTEN; Таблица 307 – скорректировано значение по сбросу для waitonreq_status; Таблица 316 – скорректировано значение по сбросу для бит 2 – 0	65 69 71 289 293
3	30.10.2026	0.1.2	Раздел 13 дополнен; Таблица 84 – дополнено описание DMA_EN_RQ_S, DMA_EN_RQ_B; Рисунок 9 скорректирован; Подраздел 23.7 скорректирован; Раздел 31 скорректирован; Таблица 437 – по результатам измерений исправлена норма параметра I_{CCO_UCC}; Таблица 438 – по результатам исследований исправлено значение U_{IL} при U_{CC} < 3 В; Таблица 439 – добавлены режимы для параметров E_{DLADC}, E_{ILADC}, E_{OFFADC} и $E_{GAINADC}$	60 75 25 187 343 – 379 381 382 384