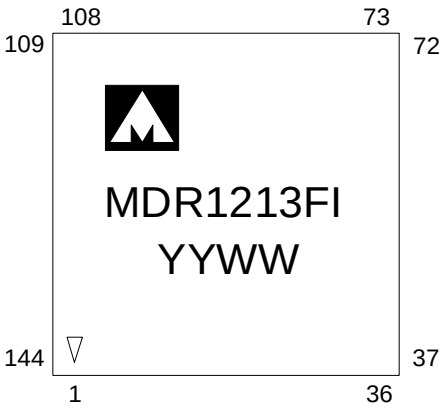
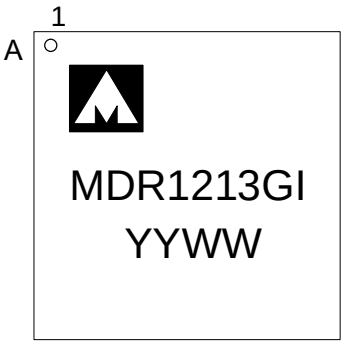
# Микросхема 32-разрядной микро-ЭВМ с Ethernet интерфейсом К1986ВЕ1FI, К1986ВЕ1GI




YY – год выпуска
WW – неделя выпуска

## Основные параметры микросхемы:

- Тактовая частота до 144 МГц;
- 62 однословных инструкции;
- 32×32-битный аппаратный умножитель за три такта;
- Поддержка прямого, косвенного и относительного режимов адресации;
- Аппаратная поддержка интерфейса CAN;
- Восьмиканальный 12-разрядный АЦП;
- Два блока 12-разрядных ЦАП;
- Диапазон напряжения питания от 3,0 до 3,6 В;
- Масса микросхем:
  - К1986ВЕ1FI – не более 1,6 г;
  - К1986ВЕ1GI – не более 1,1 г;
- Уровень чувствительности к влаге MSL:
  - К1986ВЕ1FI – 5a;
  - К1986ВЕ1GI – 3;
- Температурный диапазон: от минус 40 °С до плюс 85 °С.

## Тип корпуса:

– микросхемы К1986ВЕ1FI поставляются в 144-выводном пластмассовом корпусе QFN144 22×22×1,6 (0,5);

– микросхемы К1986ВЕ1GI поставляются в 144-выводном пластмассовом корпусе BGA144 16×16×1,6 (1,27).

## Важно:

1 Микросхемы чувствительны к влажности. Порядок обращения должен соответствовать требованиям, приведеным в ТСКЯ.430106.004Д12.

2 Спецификация действительна совместно с документом К1986ВЕ1FI, К1986ВЕ1GI Errata Notice.

## Основные характеристики микросхем

**Ядро:**

– 32-битное RISC-ядро, тактовая частота до 144 МГц, производительность 0,8 DMIPS/МГц при нулевой задержке памяти;

– умножение 32 × 32 за три цикла.

**Память:**

– встроенная энергонезависимая Flash-память программ размером 128 Кбайт;

– встроенное ОЗУ размером 48 Кбайт;

– контроллер внешней системной шины с поддержкой микросхем памяти СОЗУ, ПЗУ, NAND Flash.

**Питание и тактовая частота:**

– внешнее питание 3,0 ÷ 3,6 В;

– встроенный регулятор напряжения на 1,8 В для питания ядра;

– встроенные схемы контроля питания;

– встроенный домен управления батарейным питанием;

– встроенный подстраиваемый RC-генератор HSI 8 МГц;

– встроенный подстраиваемый RC-генератор LSI 40 кГц;

– внешний осциллятор HSE 2 ÷ 16 МГц;

– внешний осциллятор HSE2 20 ÷ 30 МГц;

– внешний осциллятор LSE 32 кГц;

– встроенный умножитель тактовой частоты PLL для ядра;

– встроенный умножитель тактовой частоты PLL для контроллера USB.

**Режим пониженного энергопотребления:**

– батарейный домен с часами реального времени и регистрами аварийного сохранения.

**Аналоговые модули:**

– 12-разрядный АЦП (до восьми каналов) с амплитудой измеряемых сигналов 0 ÷ 3,6 В;

– двухканальный 12-разрядный ЦАП;

– 32-разрядный ШИМ (до 16 каналов);

– температурный датчик.

**Периферия:**

– контроллер прямого доступа в память с функциями передачи Периферия-Память, Память-Память;

– два контроллера CAN-интерфейса;

– цифровой контроллер интерфейса по ГОСТ 18977-79;

– два цифровых контроллера интерфейса по ГОСТ Р 52070-2003;

– цифровой контроллер интерфейса Ethernet 10/100 и PHY Transceiver;

– контроллер USB интерфейса с функциями работы Device и Host;

– контроллеры интерфейсов 2×UART, 3×SPI;

– до 96 пользовательских линий ввода/вывода.

**Режим отладки:**

– последовательные отладочные интерфейсы SWD и JTAG.

**Температурный диапазон:**

– от минус 40 °С до плюс 85 °С.

# Содержание


1 Введение ... 16
2 Структурная блок-схема ... 17
3 Условное графическое обозначение ... 18
4 Описание выводов ... 20
5 Указания по применению и эксплуатации ... 62
6 Питание микросхемы ... 63
6.1 Схема сброса при включении и выключении основного питания ... 65
7 Организация памяти ... 67
7.1 Секция Code ... 69
7.2 Секция Internal SRAM ... 69
7.3 Секция Peripheral ... 69
7.4 Секция External SRAM и External device ... 69
7.5 Секция Private Peripheral Bus ... 69
7.6 Блок BUS MATRIX ... 70
7.7 Память EEPROM ... 70
7.8 Память SRAM ... 70
7.9 Регионы памяти, типы и атрибуты ... 70
7.10 Последовательность обращений к памяти ... 71
7.11 Поведение обращений к памяти ... 72
7.12 Программное определение последовательности доступа к памяти ... 73
7.13 Базовые адреса процессора ... 74
8 Загрузочное ПЗУ и режимы работы микросхемы ... 76
8.1 UART-загрузчик ... 78
8.1.1 Параметры связи по UART ... 79
8.1.2 Протокол обмена по UART ... 79
8.1.3 Прием параметров команды ... 83
8.1.4 Сообщения об ошибках ... 83
8.2 Тестовый режим работы ... 84
9 Контроллер флэш-памяти программ ... 86
9.1 Работа флэш-памяти программ в обычном режиме ... 86
9.2 Работа флэш-памяти программ в режиме программирования ... 87
9.2.1 Стирание всей памяти ... 88
9.2.2 Стирание банка памяти одной страницы размером 4 Кбайт ... 88
9.2.3 Запись 32-битного слова в память ... 89
9.2.4 Чтение 32-битного слова из памяти ... 90
9.3 Производственная информация ... 91
9.4 Описание регистров управления контроллера флэш-памяти программ ... 92
9.4.1 EEPROM_CMD ... 92
9.4.2 EEPROM_ADR ... 93
9.4.3 EEPROM_DI ... 94
9.4.4 EEPROM_DO ... 94

9.4.5 EEPROM_KEY ..... 94
10 Процессорное ядро ..... 96
10.1 Программная модель ..... 98
10.2 Стек ..... 98
10.3 Регистры ядра ..... 99
10.3.1 Регистры общего назначения R0-R12 ..... 99
10.3.2 Указать стека SP R13 ..... 99
10.3.3 Регистр связи LR R14 ..... 99
10.3.4 Счетчик команд PC R15 ..... 100
10.3.5 Программный регистр состояния PSR ..... 100
10.3.6 Программный регистр состояния приложения APSR ..... 100
10.3.7 Программный регистр состояния прерываний IPSR ..... 101
10.3.8 Программный регистр состояния выполнения EPSR ..... 102
10.3.9 Сохранение битов xPSR ..... 102
10.3.10 Регистр маски приоритетов Priority Mask ..... 102
10.3.11 Контрольный регистр специального назначения CONTROL ..... 103
10.4 Типы данных ..... 103
11 Система команд ..... 104
11.1 Встроенные функции ..... 106
11.2 Описание инструкций ..... 107
11.2.1 Операнды ..... 107
11.2.2 Ограничения на использование PC и SP ..... 107
11.2.3 Операции сдвига ..... 107
11.2.4 Выравнивание адресов ..... 109
11.2.5 Адресация относительно счетчика команд PC ..... 110
11.2.6 Условное исполнение ..... 110
11.3 Команды доступа к памяти ..... 111
11.3.1 ADR ..... 112
11.3.2 LDR и STR, непосредственно заданное смещение ..... 113
11.3.3 LDR и STR, смещение задано в регистре ..... 114
11.3.4 LDR, адресация относительно счетчика команд PC ..... 115
11.3.5 LDM и STM ..... 115
11.3.6 PUSH и POP ..... 117
11.4 Инструкции обработки данных ..... 119
11.4.1 ADD, ADC, SUB, SBC и RSB ..... 119
11.4.2 AND, ORR, EOR, BIC ..... 121
11.4.3 ASR, LSL, LSR, ROR ..... 122
11.4.4 CMP и CMN ..... 124
11.4.5 MOV и MVN ..... 125
11.4.6 MULS ..... 126
11.4.7 REV, REV16, REVSH ..... 126
11.4.8 SXT и UXT ..... 127
11.4.9 TST ..... 128

11.5 Инструкции передачи управления ..... 129
11.5.1 B, BL, BX и BLX ..... 129
11.6 Прочие инструкции ..... 131
11.6.1 BKPT ..... 131
11.6.2 CPS ..... 132
11.6.3 DMB ..... 132
11.6.4 DSB ..... 133
11.6.5 ISB ..... 133
11.6.6 MRS ..... 134
11.6.7 MSR ..... 134
11.6.8 NOP ..... 135
11.6.9 SVC ..... 135
11.6.10 WFE ..... 136
11.6.11 WFI ..... 136
12 Сигналы тактовой частоты ..... 137
12.1 Встроенный RC-генератор HSI ..... 138
12.2 Встроенный RC-генератор LSI ..... 139
12.3 Внешний осциллятор HSE ..... 139
12.4 Внешний осциллятор HSE2 ..... 139
12.5 Внешний осциллятор LSE ..... 139
12.6 Встроенный блок умножения системной тактовой частоты ..... 139
12.7 Встроенный блок умножения тактовой частоты для контроллера USB ..... 140
12.8 Описание регистров блока контроллера тактовой частоты ..... 140
12.8.1 CLOCK_STATUS ..... 141
12.8.2 PLL_CONTROL ..... 141
12.8.3 HS_CONTROL ..... 142
12.8.4 CPU_CLOCK ..... 143
12.8.5 USB_CLOCK ..... 144
12.8.6 ADC_MCO_CLOCK ..... 145
12.8.7 RTC_CLOCK ..... 146
12.8.8 PER_CLOCK ..... 147
12.8.9 CAN_CLOCK ..... 148
12.8.10 TIM_CLOCK ..... 149
12.8.11 UART_CLOCK ..... 150
12.8.12 SSP_CLOCK ..... 151
12.8.13 ETH_CLOCK ..... 152
12.8.14 DMA_DONE_STICK ..... 153
13 Батарейный домен и часы реального времени ..... 154
13.1 Часы реального времени ..... 154
13.2 Регистры аварийного сохранения ..... 155
13.3 Описание регистров блока батарейного домена ..... 156
13.3.1 BKP_REG_[0D…00] ..... 156
13.3.2 BKP_REG_0E ..... 157

13.3.3 BKP_REG_0F ..... 158
13.3.4 RTC_CNT ..... 160
13.3.5 RTC_DIV ..... 161
13.3.6 RTC_PRL ..... 161
13.3.7 RTC_ALRM ..... 161
13.3.8 RTC_CS ..... 162
14 Порты ввода/вывода ..... 164
14.1 Описание регистров портов ввода/вывода ..... 168
14.1.1 PORTx_RXTX ..... 168
14.1.2 PORTx_OE ..... 169
14.1.3 PORTx_FUNC ..... 169
14.1.4 PORTx_ANALOG ..... 169
14.1.5 PORTx_PULL ..... 170
14.1.6 PORTx_PD ..... 170
14.1.7 PORTx_PWR ..... 171
14.1.8 PORTx_GFEN ..... 171
14.1.9 PORTx_SETTX ..... 172
14.1.10 PORTx_CLRTX ..... 172
15 Детектор напряжения питания ..... 173
15.1 PVDCS ..... 174
16 Внешняя системная шина ..... 176
16.1 Работа с внешними статическими ОЗУ, ПЗУ и периферийными устройствами ..... 176
16.1.1 Обмен по внешней системной шине при задании длительности через WS_ACTIVE, WS_SETUP, WS_HOLD ..... 179
16.2 Работа с внешней NAND флэш-памятью ..... 179
16.3 Описание регистров блока контроллера внешней системной шины ..... 183
16.3.1 CONTROL ..... 184
16.3.2 NAND_CYCLES ..... 185
16.3.3 RAM_CYCLESx ..... 186
17 Режим StandAlone ..... 188
18 Контроллер интерфейса USB ..... 191
18.1 Инициализация контроллера при включении ..... 191
18.2 Задание параметров шины USB и события подключения/отключения ..... 192
18.3 Задание адреса и инициализация оконечных точек ..... 192
18.4 Транзакция IN (Usb Device) ..... 192
18.5 Транзакция SETUP/OUT (Usb Device) ..... 195
18.6 Транзакция SETUP/OUT (Usb Host) ..... 197
18.7 Транзакция IN (Usb Host) ..... 197
18.8 Отправка SOF пакетов и отсчет времени (Usb Host) ..... 198
18.9 Описание регистров управления контроллером интерфейса USB ..... 199
18.9.1 USB_HSCR ..... 201
18.9.2 USB_HSVR ..... 202

18.9.3 Регистры режима HOST ..... 202
18.9.4 USB Slave (Device) ..... 212
19 Контроллер интерфейса CAN ..... 221
19.1 Режимы работы ..... 222
19.1.1 Нормальной режим ..... 222
19.1.2 Режим работы только на прием (Receive Only Mode) ..... 222
19.1.3 Режим самотестирования (Self Test Mode) ..... 222
19.2 Типы пакетов сообщений ..... 223
19.3 Структура пакета данных (Data Frame) ..... 224
19.3.1 Начало пакета (Start of frame) ..... 225
19.3.2 Поле арбитража (Arbitration field) ..... 225
19.3.3 Поле управления (Control field) ..... 227
19.3.4 Поле данных (Data field) ..... 227
19.3.5 Поле CRC (CRC field) ..... 227
19.3.6 Поле подтверждения (ACK field) ..... 227
19.3.7 Конец пакета (End of frame) ..... 227
19.3.8 Структура пакета удаленного запроса данных (Remote frame) ..... 227
19.3.9 Арбитраж на шине ..... 228
19.4 Инициализация ..... 229
19.5 Передача сообщений ..... 230
19.6 Передача сообщений по Remote Transmit Request (RTR) ..... 230
19.7 Прием сообщений ..... 230
19.8 Автоматическая фильтрация принимаемых сообщений ..... 231
19.9 Перезапись принятых сообщений ..... 231
19.10 Задание скорости передачи и момента семплирования ..... 231
19.11 Синхронизация ..... 233
19.12 Обработка ошибок ..... 233
19.13 Прерывания ..... 237
19.14 Описание регистров контроллера CAN ..... 238
19.14.1 CANx_CONTROL ..... 239
19.14.2 CANx_STATUS ..... 240
19.14.3 CANx_BITTMNG ..... 242
19.14.4 CANx_INT_EN ..... 243
19.14.5 CANx_OVER ..... 244
19.14.6 CANx_BUF_xx_CON ..... 244
19.14.7 CANx_INT_RX ..... 245
19.14.8 CANx_RX ..... 245
19.14.9 CANx_INT_TX ..... 246
19.14.10 CANx_TX ..... 246
19.14.11 CANx_RXID, CANx_TXID, CANx_BUF_xx_ID ..... 247
19.14.12 CANx_RXDLC, CANx_TXDLC, CANx_BUF_xx_DLC ..... 247
19.14.13 CANx_RXDATAL, CANx_TXDATAL, CANx_BUF_xx_DATAL ..... 248
19.14.14 CANx_RXDATAH, CANx_TXDATAH, CANx_BUF_xx_DATAH ..... 249

20 Контроллер интерфейса по ГОСТ Р 52070-2003 ..... 250
20.1 Режимы работы ..... 251
20.2 Форматы сообщений ..... 252
20.3 Формат слов ..... 253
20.4 Инициализация ..... 258
20.5 Прием и передача в режиме ОУ ..... 259
20.6 Прием и передача в режиме КШ ..... 260
20.7 Прерывания ..... 261
20.8 Описание регистров контроллера ГОСТ Р 52070-2003 ..... 262
20.8.1 CONTROL ..... 262
20.8.2 STATUS ..... 264
20.8.3 ERROR ..... 268
20.8.4 CommandWord1 ..... 269
20.8.5 CommandWord2 ..... 270
20.8.6 ModeData ..... 270
20.8.7 StatusWord1 ..... 271
20.8.8 StatusWord2 ..... 272
20.8.9 INTEN ..... 272
20.8.10 MSG ..... 273
20.8.11 DATA ..... 274
21 Таймеры общего назначения ..... 275
21.1 Основные характеристики ..... 275
21.1.1 Структурная схема ..... 276
21.2 Базовый блок таймера ..... 277
21.2.1 Инициализация тактирования таймера ..... 277
21.2.2 Инициализация основного счетчика таймера ..... 277
21.2.3 Режимы счета ..... 278
21.2.4 Тактовая частота $F_{DTS}$ ..... 280
21.3 Источник событий для счета ..... 280
21.3.1 Внутренний тактовый сигнал (TIM_CLKd) ..... 282
21.3.2 Событие в другом таймере (CNT==ARR) ..... 283
21.3.3 Внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CHyi ..... 284
21.3.4 Внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR ..... 286
21.4 Режим захвата ..... 287
21.5 Режим ШИМ ..... 289
21.5.1 Генератор опорного сигнала REF ..... 289
21.5.2 Генератор «мертвой зоны» ..... 291
21.5.3 Выходные блоки ..... 292
21.6 Блок цифрового фильтра ..... 292
21.7 Флаги состояний, прерывания и запросы DMA ..... 294
21.7.1 Флаги состояний ..... 294

21.7.2 Прерывания ... 294
21.7.3 Запросы DMA ... 295
21.8 Примеры ... 295
21.8.1 Обычный счетчик ... 295
21.8.2 Режим захвата ... 296
21.8.3 Режим ШИМ ... 297
21.9 Описание регистров блока таймера ... 299
21.9.1 CNT ... 300
21.9.2 PSG ... 300
21.9.3 ARR ... 301
21.9.4 CNTRL ... 301
21.9.5 CCRy ... 303
21.9.6 CCRy1 ... 303
21.9.7 CHy_CNTRL ... 303
21.9.8 CHy_CNTRL1 ... 306
21.9.9 CHy_CNTRL2 ... 307
21.9.10 CHy_DTG ... 308
21.9.11 BRKETR_CNTRL ... 308
21.9.12 STATUS ... 309
21.9.13 IE ... 311
21.9.14 DMA_RE, DMA_RE1-DMA_RE4 ... 313
22 Контроллер АЦП ... 315
22.1 Преобразование внешнего канала ... 316
22.2 Последовательное преобразование нескольких каналов ... 316
22.3 Преобразование с контролем границ ... 317
22.4 Датчик опорного напряжения ... 317
22.5 Датчик температуры ... 318
22.6 Время заряда внутренней емкости ... 318
22.7 Описание регистров блока контроллера АЦП ... 320
22.7.1 ADCx_CFG ... 320
22.7.2 ADC1_H_LEVEL ... 322
22.7.3 ADC1_L_LEVEL ... 323
22.7.4 ADC1_RESULT ... 323
22.7.5 ADC1_STATUS ... 324
22.7.6 ADC1_CHSEL ... 324
22.7.7 ADC1_TRIM ... 325
23 Контроллер ЦАП ... 326
23.1 Описание регистров блока контроллера ЦАП ... 326
23.1.1 DAC_CFG ... 326
23.1.2 DAC0_DATA ... 327
23.1.3 DAC1_DATA ... 328
24 Контроллер интерфейса по ГОСТ 18977-79 ... 329
24.1 Формат слова ... 330

24.2 Структурная схема канала передачи ..... 333
24.3 Описание регистров контроллера ГОСТ 18977-79 ..... 334
24.3.1 CONTROL1 ..... 335
24.3.2 CONTROL2 ..... 336
24.3.3 CONTROL3 ..... 337
24.3.4 CONTROL4 ..... 338
24.3.5 CONTROL5 ..... 339
24.3.6 CONTROL8 ..... 339
24.3.7 CONTROL9 ..... 342
24.3.8 CONTROL10 ..... 342
24.3.9 CONTROL11 ..... 343
24.3.10 INTMASK ..... 343
24.3.11 STATUS1 ..... 344
24.3.12 STATUS2 ..... 345
24.3.13 CHANNEL ..... 346
24.3.14 LABEL ..... 346
24.3.15 DATA_R ..... 346
24.3.16 DATA_R1 – DATA_R8 ..... 347
24.3.17 CONTROL1 ..... 347
24.3.18 CONTROL2 ..... 349
24.3.19 CONTROL3 ..... 351
24.3.20 CONTROL4 ..... 351
24.3.21 CONTROL5 ..... 352
24.3.22 STATUS ..... 352
24.3.23 DATA1_T ..... 353
24.3.24 DATA2_T ..... 353
24.3.25 DATA3_T ..... 354
24.3.26 DATA4_T ..... 354
25 Контроллер SSP ..... 355
25.1 Основные характеристики модуля SSP ..... 355
25.2 Программируемые параметры ..... 356
25.3 Характеристики интерфейса SPI ..... 356
25.4 Характеристики интерфейса Microwire ..... 357
25.5 Характеристики интерфейса SSI ..... 357
25.6 Общий обзор модуля SSP ..... 357
25.6.1 Блок формирования тактового сигнала ..... 358
25.6.2 Буфер FIFO передатчика ..... 358
25.6.3 Буфер FIFO приемника ..... 358
25.6.4 Блок приема и передачи данных ..... 358
25.6.5 Блок формирования прерываний ..... 359
25.6.6 Интерфейс прямого доступа к памяти ..... 359
25.6.7 Конфигурирование приемопередатчика ..... 359
25.6.8 Разрешение работы приемопередатчика ..... 360

25.6.9 Соотношения между тактовыми сигналами ..... 360
25.6.10 Программирование регистра управления CR0 ..... 361
25.6.11 Программирование регистра управления CR1 ..... 361
25.6.12 Формирование тактового сигнала обмена данными ..... 362
25.6.13 Формат информационного кадра ..... 362
25.6.14 Формат синхронного обмена SSI фирмы Texas Instruments ..... 363
25.6.15 Формат синхронного обмена SPI фирмы Motorola ..... 364
25.6.16 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0 .. 364
25.6.17 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1 .. 365
25.6.18 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0 .. 366
25.6.19 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1 .. 367
25.6.20 Формат синхронного обмена Microwire фирмы National Semiconductor 368
25.6.21 Примеры конфигурации модуля в ведущем и ведомом режимах ..... 370
25.6.22 Интерфейс прямого доступа к памяти ..... 373
25.7 Программное управление модулем ..... 375
25.7.1 Общая информация ..... 375
25.7.2 Описание регистров контроллера SSP ..... 376
25.8 Прерывания ..... 383
25.8.1 SSPRXINTR ..... 383
25.8.2 SSPTXINTR ..... 383
25.8.3 SSPRORINTR ..... 384
25.8.4 SSPRTINTR ..... 384
25.8.5 SSPINTR ..... 384
26 Контроллер UART ..... 385
26.1 Основные характеристики модуля UART ..... 385
26.2 Программируемые параметры ..... 386
26.3 Отличия от контроллера UART 16C650 ..... 386
26.4 Функциональные возможности ..... 387
26.5 Описание функционирования блока UART ..... 388
26.5.1 Генератор тактового сигнала приемопередатчика ..... 388
26.5.2 Буфер FIFO передатчика ..... 389
26.5.3 Буфер FIFO приемника ..... 389
26.5.4 Блок передатчика ..... 389
26.5.5 Блок приемника ..... 389
26.5.6 Блок формирования прерываний ..... 389
26.5.7 Интерфейс прямого доступа к памяти ..... 390
26.5.8 Блок и регистры синхронизации ..... 390
26.6 Описание функционирования ИК-кодека IrDA SIR ..... 390
26.6.1 Кодер ИК-передатчика ..... 390
26.6.2 Декодер ИК-приемника ..... 391
26.7 Описание работы UART ..... 392
26.7.1 Сброс модуля ..... 392

26.7.2 Тактовые сигналы ... 392
26.7.3 Работа универсального асинхронного приемопередатчика ... 392
26.7.4 Коэффициент деления частоты ... 393
26.7.5 Передача и прием данных ... 393
26.7.6 Биты ошибки ... 394
26.7.7 Бит переполнения буфера ... 394
26.7.8 Запрет буфера FIFO ... 395
26.7.9 Работа кодека ИК-обмена данными IrDA SIR ... 395
26.8 Линии управления модемом ... 397
26.8.1 Аппаратное управление потоком данных ... 397
26.8.2 Управление потоком данных по линии RTS ... 398
26.8.3 Управление потоком данных по линии CTS ... 399
26.9 Интерфейс прямого доступа к памяти ... 399
26.10 Прерывания ... 401
26.10.1 UARTMSINTR ... 402
26.10.2 UARTRXINTR ... 402
26.10.3 UARTTXINTR ... 403
26.10.4 UARTRTINTR ... 403
26.10.5 UARTEINTR ... 403
26.10.6 UARTINTR ... 404
26.11 Программное управление модулем ... 404
26.12 Обобщенные данные о регистрах устройства ... 405
26.12.1 Регистр данных DR ... 406
26.12.2 Регистр состояния приемника / сброса ошибки RSR_ECR ... 407
26.12.3 Регистр флагов FR ... 408
26.12.4 Регистр управления ИК-обменом в режиме пониженного энергопотребления ILPR ... 409
26.12.5 Регистр целой части делителя скорости передачи данных IBRD ... 410
26.12.6 Регистр дробной части делителя скорости передачи данных FBRD ... 410
26.12.7 Регистр управления линией LCR_H ... 412
26.12.8 Регистр управления CR ... 413
26.12.9 Регистр порога прерывания по заполнению буфера FIFO IFLS ... 415
26.12.10 Регистр установки сброса маски прерывания IMSC ... 416
26.12.11 Регистр состояния прерываний RIS ... 417
26.12.12 Регистр маскированного состояния прерываний MIS ... 419
26.12.13 Регистр сброса прерываний ICR ... 420
26.12.14 Регистр управления прямым доступом к памяти DMACR ... 421
26.12.15 Регистр управления тестированием TCR ... 422
27 Контроллер прямого доступа в память DMA ... 423
27.1 Основные свойства контроллера DMA ... 423
27.2 Термины и определения ... 424
27.3 Функциональное описание ... 425
27.3.1 Распределение каналов DMA ... 426

27.3.2 Блок, подключенный к шине APB ..... 427
27.3.3 Блок, подключенный к шине AHB ..... 427
27.3.4 Управляющий блок DMA ..... 427
27.3.5 Типы передач ..... 428
27.3.6 Разрядность передаваемых данных ..... 428
27.3.7 Управление защитой данных ..... 428
27.3.8 Инкремент адреса ..... 429
27.4 Управление DMA ..... 429
27.4.1 Правила обмена данными ..... 429
27.4.2 Диаграммы работы контроллера DMA ..... 432
27.4.3 Правила арбитража DMA ..... 438
27.4.4 Приоритет ..... 438
27.4.5 Типы циклов DMA ..... 441
27.5 Структура управляющих данных канала ..... 453
27.5.1 Разряды управления ..... 458
27.6 Описание регистров контроллера DMA ..... 464
27.6.1 Статусный регистр DMA STATUS ..... 466
27.6.2 Регистр конфигурации DMA CFG ..... 467
27.6.3 Регистр базового адреса управляющих данных каналов CTRL_BASE_PTR ..... 467
27.6.4 Регистр базового адреса альтернативных управляющих данных каналов ALT_CTRL_BASE_PTR ..... 468
27.6.5 Регистр статуса ожидания запроса на обработку каналов WAITONREQ_STATUS ..... 469
27.6.6 Регистр программного запроса на обработку каналов CHNL_SW_REQUEST ..... 469
27.6.7 Регистр установки пакетного обмена каналов CHNL_USEBURST_SET 470
27.6.8 Регистр сброса пакетного обмена каналов CHNL_USEBURST_CLR 471
27.6.9 Регистр маскирования запросов на обслуживание каналов CHNL_REQ_MASK_SET ..... 471
27.6.10 Регистр очистки маскирования запросов на обслуживание каналов CHNL_REQ_MASK_CLR ..... 472
27.6.11 Регистр установки разрешения каналов CHNL_ENABLE_SET ..... 473
27.6.12 Регистр сброса разрешения каналов CHNL_ENABLE_CLR ..... 474
27.6.13 Регистр установки первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_SET ..... 474
27.6.14 Регистр сброса первичной/альтернативной структуры управляющих данных каналов CHNL_PRI_ALT_CLR ..... 476
27.6.15 Регистр установки приоритета каналов CHNL_PRIORITY_SET ..... 477
27.6.16 Регистр сброса приоритета каналов CHNL_PRIORITY_CLR ..... 477
27.6.17 Регистр сброса флага ошибки ERR_CLR ..... 478

28 Контроллер интерфейса Ethernet ..... 479
28.1 Передача пакета ..... 480
28.2 Принцип работы передатчика ..... 480
28.3 Прием пакета ..... 481
28.4 Принцип работы приемника ..... 481
28.5 Линейный режим работы буферов ..... 482
28.6 Автоматический режим работы буферов ..... 482
28.7 Режим FIFO работы буферов ..... 482
28.8 Прерывания ..... 482
28.9 Режим детерминированного времени доставки ..... 483
28.10 Режим КЗ ..... 483
28.11 Регистры ..... 484
28.11.1 Поле управления передачи пакета ..... 485
28.11.2 Поле состояния передачи пакета ..... 485
28.11.3 Поле состояния приема пакета ..... 486
28.11.4 G_CFGh ..... 488
28.11.5 G_CFGl ..... 488
28.11.6 X_CFG ..... 489
28.11.7 R_CFG ..... 490
28.11.8 IMR/IFR ..... 492
28.11.9 STAT ..... 493
28.11.10 MDIO_CTRL ..... 494
28.12 Блок PHY ..... 494
28.12.1 Регистры ..... 497
28.12.2 PHY_Control ..... 497
28.12.3 PHY_Status ..... 498
28.12.4 Основной регистр управления (0) ..... 500
28.12.5 Основной регистр состояния (1) ..... 501
28.12.6 Регистры идентификатора PHY (2, 3) ..... 502
28.12.7 Регистр рекомендаций автоподстройки (4) ..... 502
28.12.8 Регистр возможностей оппонента по автоподстройке (5) ..... 503
28.12.9 Регистр расширенного управления автоподстройкой (6) ..... 503
28.12.10 Регистр расширенного управления режимами (18) ..... 504
28.12.11 Регистр флагов прерываний (29) ..... 504
28.12.12 Регистр маски прерываний (30) ..... 505
28.12.13 Регистр расширенного управления и состояния PHY (31) ..... 505
29 Прерывания и исключения ..... 506
29.1 Типы исключений ..... 506
29.2 Приоритет исключений ..... 508
29.3 Уровни приоритета ..... 508
29.4 Стек ..... 509
29.5 Вход в прерывание ..... 509
29.6 Выход из исключений ..... 511

29.7 Запаздывание (late-arriving) ..... 512
29.8 Передача управления исключению ..... 513
29.9 Уровни активации ..... 513
29.10 Lock-up ..... 515
30 Контроллер прерываний NVIC ..... 517
30.1 Логика работы прерываний контроллера NVIC ..... 517
30.2 Программная модель NVIC ..... 520
30.3 Регистр разрешения прерываний ..... 521
30.4 Регистр запрета прерывания ..... 521
30.5 Регистр перевода прерывания в состояние ожидания обслуживания ..... 522
30.6 Регистр сброса состояния ожидания обслуживания ..... 523
30.7 Регистр приоритета прерываний ..... 523
30.8 Прерывания по уровню и по фронту ..... 524
30.9 Повторная выборка уровня прерываний ..... 524
30.10 Прерывания как входы общего назначения ..... 525
31 Блок управления системой ядра ..... 526
31.1 Вспомогательный управляющий регистр (ACTLR) ..... 527
31.2 Регистр управления и статуса SysTick (CTRL) ..... 527
31.3 Регистр перегружаемого значения SysTick (LOAD) ..... 528
31.4 Регистр текущего значения SysTick (VAL) ..... 528
31.5 Регистр калибровочного значения SysTick (CALIB) ..... 529
31.6 Регистр CPUID ..... 529
31.7 Регистр управления состоянием прерываний (ICSR) ..... 530
31.8 Регистр управления прерываниями и программным сбросом (AIRCR) ..... 532
31.9 Регистр конфигурации и управления (CCR) ..... 533
31.10 Регистры приоритета системных обработчиков ..... 534
31.11 Регистр приоритета системного обработчика 2 (SHP2) ..... 534
31.12 Регистр приоритета системного обработчика 3 (SHP3) ..... 534
31.13 Регистр управления и состояния системного обработчика (SHCSR) ..... 535
32 Сторожевые таймеры ..... 536
32.1 Описание регистров блока сторожевых таймеров ..... 536
32.1.1 Блок сторожевого таймера IWDG ..... 536
32.1.2 Блок сторожевого таймера WWDG ..... 541
33 Электрические параметры ..... 543
34 Предельно-допустимые характеристики ..... 545
35 Справочные данные ..... 547
36 Типовые зависимости ..... 551
37 Габаритный чертеж микросхемы ..... 552
38 Информация для заказа ..... 554

# 1 Введение

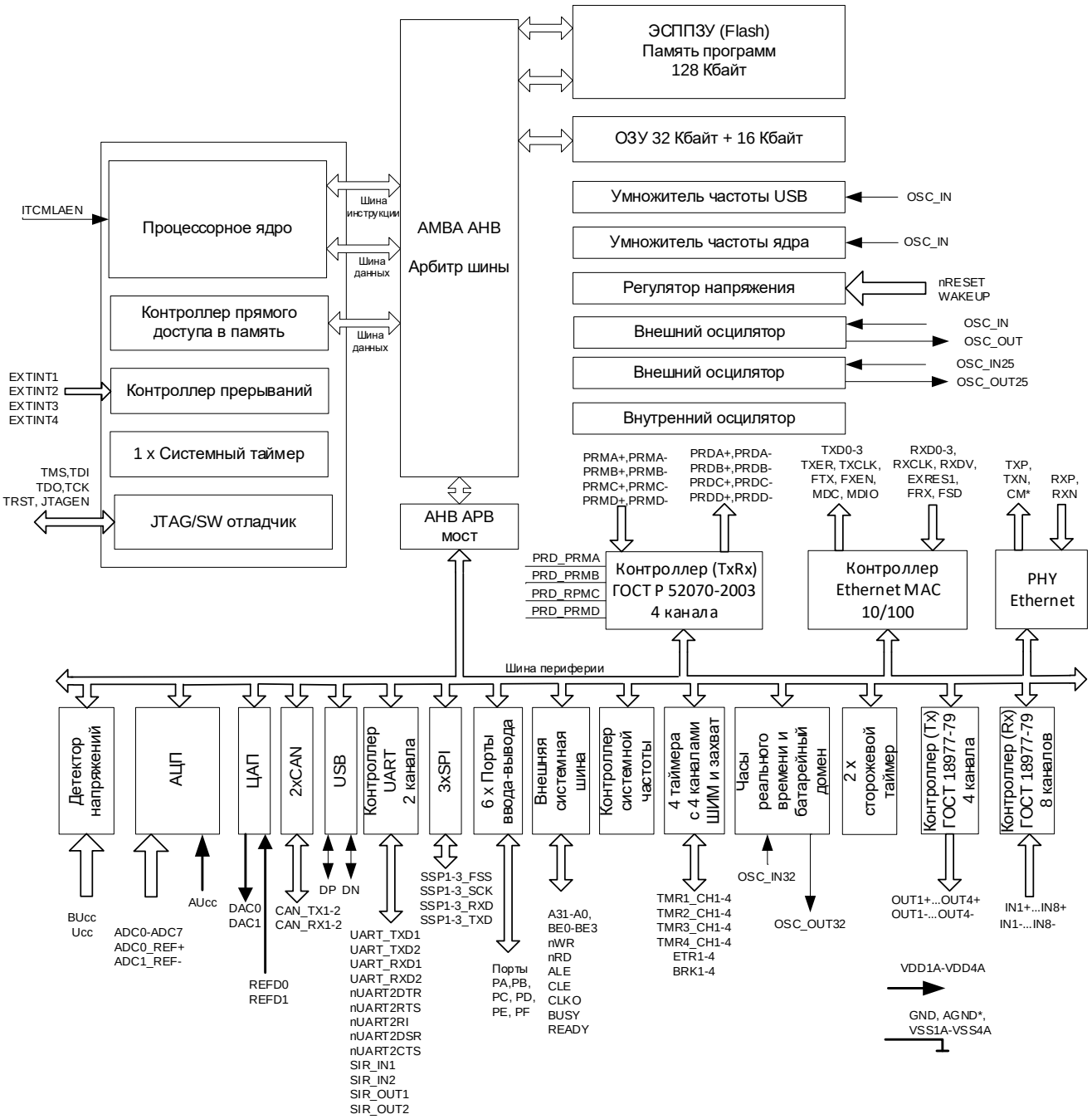
Микросхемы интегральные К1986ВЕ1FI, К1986ВЕ1GI (далее – микросхемы) используются для авиационного применения, имеют встроенную флэш-память программ и построены на базе высокопроизводительного процессорного RISC-ядра. Микросхемы работают на тактовой частоте до 144 МГц и содержат 128 Кбайт флэш-памяти программ и 48 Кбайт ОЗУ. Периферия включает в себя контроллер интерфейса USB со встроенным аналоговым приемопередатчиком со скоростью передачи 12 Мбит/с (Full Speed) и 1,5 Мбит/с (Low Speed), интерфейсы UART, SPI и CAN, авиационные интерфейсы по ГОСТ 18977-79 и ГОСТ Р52070-2003, контроллер внешней системной шины, позволяющий работать с внешними микросхемами статического ОЗУ и ПЗУ, NAND флэш-памятью и другими периферийными устройствами, а также цифровой интерфейс Ethernet со скоростью передачи 10/100 Мбит, интерфейсом MII и со встроенным аналоговым приемопередатчиком. Микросхемы содержат четыре 32-разрядных таймера с четырьмя каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки, системный 24-разрядный таймер и два сторожевых таймера. А также 12-разрядный высокоскоростной (до 512 Квыб/с) АЦП с возможностью оцифровки информации с восьми каналов, встроенного датчика температуры и опорного напряжения и два 12-разрядных ЦАП.

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

Архитектура системы памяти за счет матрицы системных шин позволяет минимизировать возможные конфликты при работе системы и повысить общую производительность. Контроллер DMA позволяет ускорить обмен информацией между ОЗУ и периферией без участия процессорного ядра.

Встроенный регулятор, предназначенный для формирования питания внутренней цифровой части, формирует напряжение 1,8 В и не требует дополнительных внешних элементов. Таким образом, для работы микросхем достаточно одного внешнего напряжения питания в диапазоне от 3,0 до 3,6 В. Также в микросхемах реализован батарейный домен, работающий от внешней батареи, который предназначен для обеспечения функций часов реального времени и сохранения некоторого объема данных при отсутствии основного питания. Встроенный детектор напряжения питания может отслеживать уровень внешнего основного питания и уровень напряжения питания на батарее. Аппаратные схемы сброса по просадке питания позволяют исключить сбои в работе микросхем при выходе уровня напряжения питания за допустимые пределы.

# 2 Структурная блок-схема




* Для микросхем К1986ВЕ1FI

Рисунок 1 – Структурная блок-схема

# 3 Условное графическое обозначение

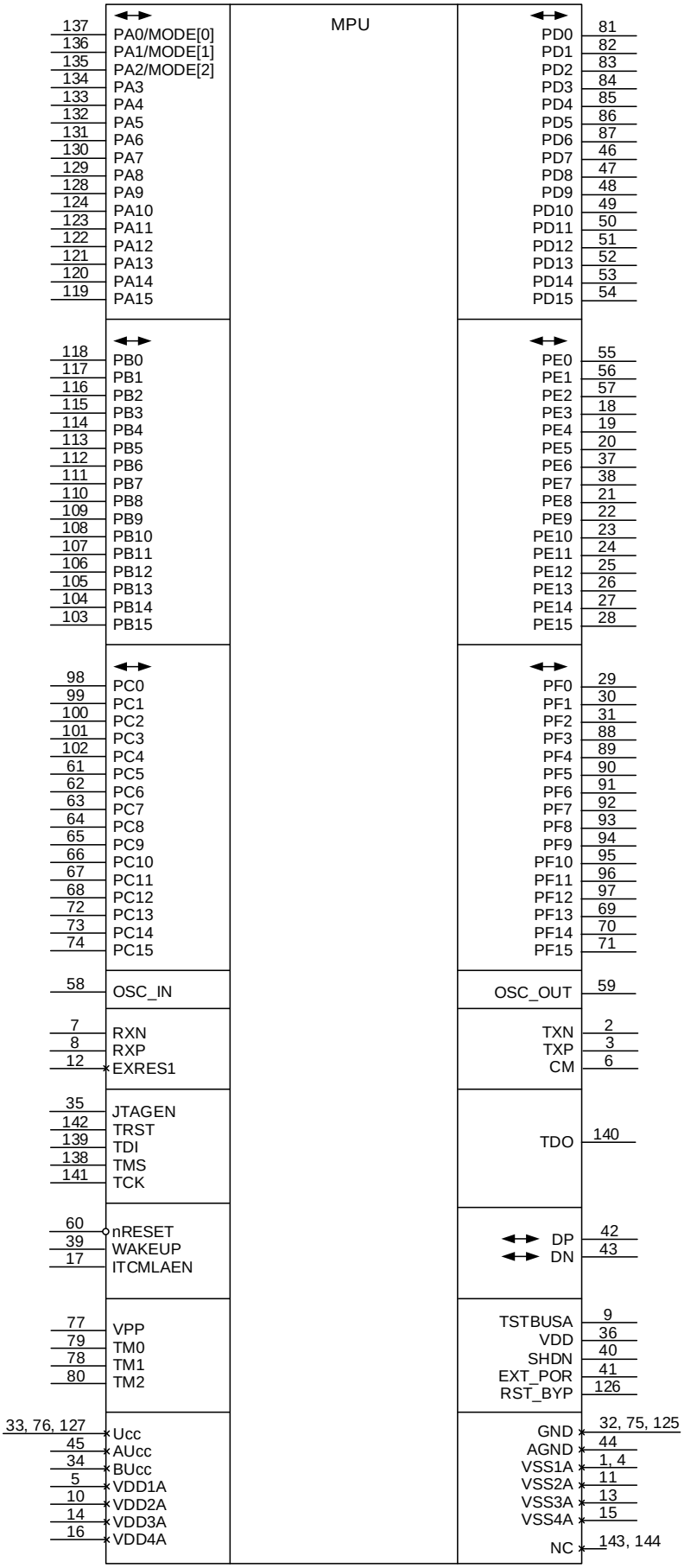



Рисунок 2 – Условное графическое обозначение микросхем К1986ВЕ1FI

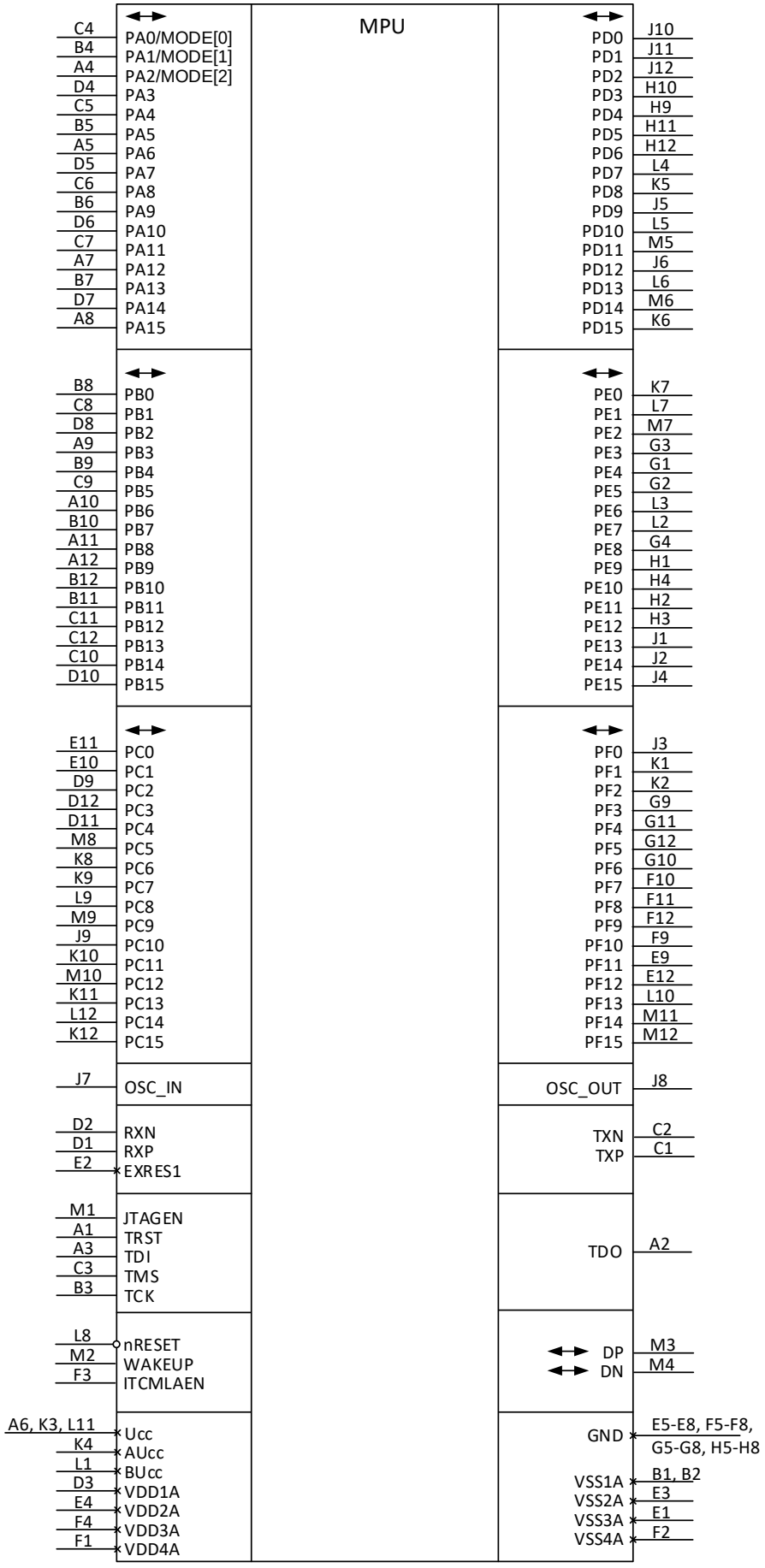



Рисунок 3 – Условное графическое обозначение микросхем К1986ВЕ1GI

# 4 Описание выводов

Таблица 1 – Описание выводов микросхем К1986ВЕ1FI

| Номер вывода | Обозначение вывода | Тип вывода | Функциональное назначение |
|---|---|---|---|
| 1 | VSS1A | GND | Общий PHY |
| 2 | TXN | O | Дифференциальный выход передатчика на трансформаторы |
| 3 | TXP | | |
| 4 | VSS1A | GND | Общий PHY |
| 5 | VDD1A | PWR | Аналоговое питание PHY |
| 6 | CM | O | Выход опорного напряжения приёмника |
| 7 | RXN | I | Дифференциальный вход приёмника с трансформатора |
| 8 | RXP | | |
| 9 | TSTBUSA | T | Тестовый вывод. Рекомендуется не подсоединять |
| 10 | VDD2A | PWR | Аналоговое питание PHY |
| 11 | VSS2A | GND | Общий PHY |
| 12 | EXRES1 | – | Вывод для подключения опорного резистора 12,4 кОм 1 % на VSS2A |
| 13 | VSS3A | GND | Общий PHY |
| 14 | VDD3A | PWR | Аналоговое питание PHY |
| 15 | VSS4A | GND | Общий PHY |
| 16 | VDD4A | PWR | Аналоговое питание PHY |
| 17 | ITCMLAEN | I | Сигнал выбора памяти программ:<br>1 – внутренняя;<br>0 – внешняя |
| 18 | PE3 | IO | Порт E |
| 19 | PE4 | | |
| 20 | PE5 | | |
| 21 | PE8 | | |
| 22 | PE9 | | |
| 23 | PE10 | | |
| 24 | PE11 | | |
| 25 | PE12 | | |
| 26 | PE13 | | |
| 27 | PE14 | | |
| 28 | PE15 | | |
| 29 | PF0 | IO | Порт F |
| 30 | PF1 | | |
| 31 | PF2 | | |
| 32 | GND | GND | Общий |
| 33 | $U_{CC}$ | PWR | Питание |
| 34 | BUcc | | Батарейное питание |
| 35 | JTAGEN | I | Вход разрешения отладочного TAP интерфейса на выводах JTAG |
| 36 | VDD | T | Тестовый вывод. Рекомендуется не подсоединять |
| 37 | PE6 | IO | Порт E |
| 38 | PE7 | | |
| 39 | WAKEUP | I | Сигнал внешнего выхода из режима Standby |

| Номер вывода | Обозначение вывода | Тип вывода | Функциональное назначение |
|---|---|---|---|
| 40 | SHDN | T | Тестовый вывод. Рекомендуется не подсоединять |
| 41 | EXT_POR | | |
| 42 | DP | IO | Шина USB D+ |
| 43 | DN | | Шина USB D- |
| 44 | AGND | GND | Общий АЦП, ЦАП |
| 45 | AUcc | PWR | Аналоговое питание АЦП, ЦАП (должно совпадать с $U_{CC}$) |
| 46 | PD7 | IO | Порт D |
| 47 | PD8 | | |
| 48 | PD9 | | |
| 49 | PD10 | | |
| 50 | PD11 | | |
| 51 | PD12 | | |
| 52 | PD13 | | |
| 53 | PD14 | | |
| 54 | PD15 | | |
| 55 | PE0 | IO | Порт E |
| 56 | PE1 | | |
| 57 | PE2 | | |
| 58 | OSC_IN | I | Генератор HSE |
| 59 | OSC_OUT | O | |
| 60 | nRESET | I | Сигнал внешнего сброса |
| 61 | PC5 | IO | Порт C |
| 62 | PC6 | | |
| 63 | PC7 | | |
| 64 | PC8 | | |
| 65 | PC9 | | |
| 66 | PC10 | | |
| 67 | PC11 | | |
| 68 | PC12 | | |
| 69 | PF13 | IO | Порт F |
| 70 | PF14 | | |
| 71 | PF15 | | |
| 72 | PC13 | IO | Порт C |
| 73 | PC14 | | |
| 74 | PC15 | | |
| 75 | GND | GND | Общий |
| 76 | $U_{CC}$ | PWR | Питание |
| 77 | VPP | T | Тестовый вывод. Рекомендуется не подсоединять |
| 78 | TM1 | | |
| 79 | TM0 | | |
| 80 | TM2 | | |
| 81 | PD0 | IO | Порт D |
| 82 | PD1 | | |
| 83 | PD2 | | |
| 84 | PD3 | | |
| 85 | PD4 | | |

| Номер вывода | Обозначение вывода | Тип вывода | Функциональное назначение |
|---|---|---|---|
| 86 | PD5 | IO | Порт D |
| 87 | PD6 | | |
| 88 | PF3 | IO | Порт F |
| 89 | PF4 | | |
| 90 | PF5 | | |
| 91 | PF6 | | |
| 92 | PF7 | | |
| 93 | PF8 | | |
| 94 | PF9 | | |
| 95 | PF10 | | |
| 96 | PF11 | | |
| 97 | PF12 | | |
| 98 | PC0 | IO | Порт C |
| 99 | PC1 | | |
| 100 | PC2 | | |
| 101 | PC3 | | |
| 102 | PC4 | | |
| 103 | PB15 | IO | Порт B |
| 104 | PB14 | | |
| 105 | PB13 | | |
| 106 | PB12 | | |
| 107 | PB11 | | |
| 108 | PB10 | | |
| 109 | PB9 | | |
| 110 | PB8 | | |
| 111 | PB7 | | |
| 112 | PB6 | | |
| 113 | PB5 | | |
| 114 | PB4 | | |
| 115 | PB3 | | |
| 116 | PB2 | | |
| 117 | PB1 | | |
| 118 | PB0 | | |
| 119 | PA15 | IO | Порт A |
| 120 | PA14 | | |
| 121 | PA13 | | |
| 122 | PA12 | | |
| 123 | PA11 | | |
| 124 | PA10 | | |
| 125 | GND | GND | Общий |
| 126 | RST_BYP | T | Тестовый вывод. Рекомендуется не подсоединять |
| 127 | $U_{CC}$ | PWR | Питание |
| 128 | PA9 | IO | Порт A |
| 129 | PA8 | | |
| 130 | PA7 | | |
| 131 | PA6 | | |

| Номер вывода | Обозначение вывода | Тип вывода | Функциональное назначение |
|---|---|---|---|
| 132 | PA5 | IO | Порт A |
| 133 | PA4 | | |
| 134 | PA3 | | |
| 135 | PA2/MODE[2] | IO | Порт A/Режим работы микросхемы |
| 136 | PA1/MODE[1] | | |
| 137 | PA0/MODE[0] | | |
| 138 | TMS | IO | Выбор режима тестирования - обеспечивает переход схемы в/из режима тестирования и переключение между разными режимами тестирования |
| 139 | TDI | I | Вход тестовых данных - вход последовательных данных периферийного сканирования. Команды и данные вводятся в микросхему с этого вывода по переднему фронту сигнала TCK |
| 140 | TDO | O | Выход тестовых данных - выход последовательных данных. Команды и данные выводятся из микросхемы с этого вывода по заднему фронту сигнала TCK |
| 141 | TCK | IO | Тестовое тактирование - тактирует работу встроенного автомата управления периферийным сканированием |
| 142 | TRST | IO | Тестовый сброс - сигнал сброса встроенного автомата управления периферийным сканированием |
| 143 | NC | - | Не используется. Не подключать |
| 144 | NC | | |

Таблица 2 – Дополнительные функции выводов микросхем К1986ВЕ1FI

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| Порт А/Режим работы микросхемы | | | | |
| 137 | PA0 / MODE[0] | Основ. | D0 | Вход/выход 0 порта А<br>Вход/выход данных 0 системной шины EXT_BUS |
| | | Альтер. | EXTINT1 | Вход внешнего прерывания |
| | | Переопр. | ETR1 | Вход внешнего события таймера |
| 136 | PA1 / MODE[1] | Основ. | D1 | Вход/выход 1 порта А<br>Вход/выход данных 1 системной шины EXT_BUS |
| | | Альтер. | EXTINT2 | Вход внешнего прерывания |
| | | Переопр. | ETR2 | Вход внешнего события таймера 2 |
| 135 | PA2 / MODE[2] | Основ. | D2 | Вход/выход 2 порта А<br>Вход/выход данных 2 системной шины EXT_BUS |
| | | Альтер. | EXTINT3 | Вход внешнего прерывания |
| | | Переопр. | ETR3 | Вход внешнего события таймера 3 |
| Порт А | | | | |
| 134 | PA3 | Основ. | D3 | Вход/выход 3 порта А<br>Вход/выход данных 3 системной шины EXT_BUS |
| | | Альтер. | EXTINT4 | Вход внешнего прерывания |
| | | Переопр. | BRK1 | Вход внешнего события таймера 1 |
| 133 | PA4 | Основ. | D4 | Вход/выход 4 порта А<br>Вход/выход данных 4 системной шины EXT_BUS |
| | | Альтер. | BRK2 | Вход внешнего события таймера 2 |
| | | Переопр. | FRX | Вход приёма с оптического приёмопередатчика Ethernet 1 |
| 132 | PA5 | Основ. | D5 | Вход/выход 5 порта А<br>Вход/выход данных 5 системной шины EXT_BUS |
| | | Альтер. | BRK3 | Вход внешнего события таймера 3 |
| | | Переопр. | FSD | Вход разрешения приёма с оптического приёмопередатчика Ethernet 1 |
| 131 | PA6 | Основ. | D6 | Вход/выход 6 порта А<br>Вход/выход данных 6 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH1 | Вход/выход канала 1 таймера 4 |
| | | Переопр. | FXEN | Выход разрешения передачи в оптический приёмопередатчик Ethernet 1 |
| 130 | PA7 | Основ. | D7 | Вход/выход 7 порта А<br>Вход/выход данных 7 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH1N | Инверсный выход канала 1 таймера 4 |
| | | Переопр. | FTX | Выход передачи в оптический приёмопередатчик Ethernet 1 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 129 | PA8 | Основ. | D8 | Вход/выход 8 порта A<br>Вход/выход данных 8 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH2 | Вход/выход канала 2 таймера 4 |
| | | Переопр. | PRMC+ | Положительный вход приёма канала C по ГОСТ Р 52070-2003 |
| 128 | PA9 | Основ. | D9 | Вход/выход 9 порта A<br>Вход/выход данных 9 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH2N | Инверсный выход канала 2 таймера 4 |
| | | Переопр. | PRMC- | Отрицательный вход приёма канала C по ГОСТ Р 52070-2003 |
| 124 | PA10 | Основ. | D10 | Вход/выход 10 порта A<br>Вход/выход данных 10 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH3 | Вход/выход канала 3 таймера 4 |
| | | Переопр. | PRMD+ | Положительный вход приёма канала D по ГОСТ Р 52070-2003 |
| 123 | PA11 | Основ. | D11 | Вход/выход 11 порта A<br>Вход/выход данных 11 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH3N | Инверсный выход канала 3 таймера 4 |
| | | Переопр. | PRMD- | Отрицательный вход приёма канала D по ГОСТ Р 52070-2003 |
| 122 | PA12 | Основ. | D12 | Вход/выход 12 порта A<br>Вход/выход данных 12 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH4 | Вход/выход канала 4 таймера 4 |
| | | Переопр. | PRDC+ | Положительный выход передачи канала C по ГОСТ Р 52070-2003 |
| 121 | PA13 | Основ. | D13 | Вход/выход 13 порта A<br>Вход/выход данных 13 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH4N | Инверсный выход канала 4 таймера 4 |
| | | Переопр. | PRDC- | Отрицательный выход передачи канала C по ГОСТ Р 52070-2003 |
| 120 | PA14 | Основ. | D14 | Вход/выход 14 порта A<br>Вход/выход данных 14 системной шины EXT_BUS |
| | | Альтер. | BRK4 | Вход внешнего события таймера 4 |
| | | Переопр. | PRDD+ | Положительный выход передачи канала D по ГОСТ Р 52070-2003 |
| 119 | PA15 | Основ. | D15 | Вход/выход 15 порта A<br>Вход/выход данных 15 системной шины EXT_BUS |
| | | Альтер. | ETR4 | Вход внешнего события таймера 4 |
| | | Переопр. | PRDD- | Отрицательный выход передачи канала D по ГОСТ Р 52070-2003 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| Порт B | | | | |
| 118 | PB0 | Основ. | D16 | Вход/выход 0 порта B<br>Вход/выход данных 16 системной шины EXT_BUS |
| | | Альтер. | IN1+ | Положительный вход приёма канала 1 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH1 | Вход/выход канала 1 таймера 3 |
| 117 | PB1 | Основ. | D17 | Вход/выход 1 порта B<br>Вход/выход данных 17 системной шины EXT_BUS |
| | | Альтер. | IN1- | Отрицательный вход приёма канала 1 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH1N | Инверсный выход канала 1 таймера 3 |
| 116 | PB2 | Основ. | D18 | Вход/выход 2 порта B<br>Вход/выход данных 18 системной шины EXT_BUS |
| | | Альтер. | IN2+ | Положительный вход приёма канала 2 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH2 | Вход/выход канала 2 таймера 3 |
| 115 | PB3 | Основ. | D19 | Вход/выход 3 порта B<br>Вход/выход данных 19 системной шины EXT_BUS |
| | | Альтер. | IN2- | Отрицательный вход приёма канала 2 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH2N | Инверсный выход канала 2 таймера 3 |
| 114 | PB4 | Основ. | D20 | Вход/выход 4 порта B<br>Вход/выход данных 20 системной шины EXT_BUS |
| | | Альтер. | IN3+ | Положительный вход приёма канала 3 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH3 | Вход/выход канала 3 таймера 3 |
| 113 | PB5 | Основ. | D21 | Вход/выход 5 порта B<br>Вход/выход данных 21 системной шины EXT_BUS |
| | | Альтер. | IN3- | Отрицательный вход приёма канала 3 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH3N | Инверсный выход канала 3 таймера 3 |
| 112 | PB6 | Основ. | D22 | Вход/выход 6 порта B<br>Вход/выход данных 22 системной шины EXT_BUS |
| | | Альтер. | IN4+ | Положительный вход приёма канала 4 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH4 | Вход/выход канала 4 таймера 3 |
| 111 | PB7 | Основ. | D23 | Вход/выход 7 порта B<br>Вход/выход данных 23 системной шины EXT_BUS |
| | | Альтер. | IN4- | Отрицательный вход приёма канала 4 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH4N | Инверсный выход канала 4 таймера 3 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 110 | PB8 | | | Вход/выход 8 порта B |
| | | Основ. | D24 | Вход/выход данных 24 системной шины EXT_BUS |
| | | Альтер. | IN5+ | Положительный вход приёма канала 5 по ГОСТ 18977-79 |
| | | Переопр. | TMR1_CH1N | Инверсный выход канала 1 таймера 1 |
| 109 | PB9 | | | Вход/выход 9 порта B |
| | | Основ. | D25 | Вход/выход данных 25 системной шины EXT_BUS |
| | | Альтер. | IN5- | Отрицательный вход приёма канала 5 по ГОСТ 18977-79 |
| | | Переопр. | TMR2_CH1N | Инверсный выход канала 1 таймера 2 |
| 108 | PB10 | | | Вход/выход 10 порта B |
| | | Основ. | D26 | Вход/выход данных 26 системной шины EXT_BUS |
| | | Альтер. | IN6+ | Положительный вход приёма канала 6 по ГОСТ 18977-79 |
| | | Переопр. | TMR1_CH2N | Инверсный выход канала 2 таймера 1 |
| 107 | PB11 | | | Вход/выход 11 порта B |
| | | Основ. | D27 | Вход/выход данных 27 системной шины EXT_BUS |
| | | Альтер. | IN6- | Отрицательный вход приёма канала 6 по ГОСТ 18977-79 |
| | | Переопр. | TMR2_CH2N | Инверсный выход канала 2 таймера 2 |
| 106 | PB12 | | | Вход/выход 12 порта B |
| | | Основ. | D28 | Вход/выход данных 28 системной шины EXT_BUS |
| | | Альтер. | IN7+ | Положительный вход приёма канала 7 по ГОСТ 18977-79 |
| | | Переопр. | TMR1_CH3N | Инверсный выход канала 3 таймера 1 |
| 105 | PB13 | | | Вход/выход 13 порта B |
| | | Основ. | D29 | Вход/выход данных 29 системной шины EXT_BUS |
| | | Альтер. | IN7- | Отрицательный вход приёма канала 7 по ГОСТ 18977-79 |
| | | Переопр. | TMR2_CH3N | Инверсный выход канала 3 таймера 2 |
| 104 | PB14 | | | Вход/выход 14 порта B |
| | | Основ. | D30 | Вход/выход данных 30 системной шины EXT_BUS |
| | | Альтер. | IN8+ | Положительный вход приёма канала 8 по ГОСТ 18977-79 |
| | | Переопр. | TMR1_CH4N | Инверсный выход канала 4 таймера 1 |
| 103 | PB15 | | | Вход/выход 15 порта B |
| | | Основ. | D31 | Вход/выход данных 31 системной шины EXT_BUS |
| | | Альтер. | IN8- | Отрицательный вход приёма канала 8 по ГОСТ 18977-79 |
| | | Переопр. | TMR2_CH4N | Инверсный выход канала 4 таймера 2 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| Порт C | | | | |
| 98 | PC0 | Основ. | nWR | Вход/выход 0 порта C<br>Выход разрешения записи системной шины EXT_BUS |
| | | Альтер. | ETR1 | Входы внешнего события таймера 1 |
| | | Переопр. | BRK1 | |
| 99 | PC1 | Основ. | nRD | Вход/выход 1 порта C<br>Выход разрешения чтения системной шины EXT_BUS |
| | | Альтер. | ETR2 | Входы внешнего события таймера 2 |
| | | Переопр. | BRK2 | |
| 100 | PC2 | Основ. | ALE | Вход/выход 2 порта C<br>Выход разрешения защёлкивания адреса системной шины EXT_BUS |
| | | Альтер. | CLKO | Выход синхросигнала системной шины EXT_BUS |
| | | Переопр. | BRK3 | Вход внешнего события таймера 3 |
| 101 | PC3 | Основ. | UART_TXD1 | Вход/выход 3 порта C<br>Выходные данные UART1 |
| | | Альтер. | CLE | Выход разрешения защёлкивания команды системной шины EXT_BUS |
| | | Переопр. | SIR_OUT1 | Выходные данные IRDA UART1 |
| 102 | PC4 | Основ. | UART_RXD1 | Вход/выход 4 порта C<br>Входные данные UART1 |
| | | Альтер. | BUSY | Вход занятости внешнего устройства системной шины EXT_BUS |
| | | Переопр. | SIR_IN1 | Входные данные IRDA UART1 |
| 61 | PC5 | Основ. | EXTINT1 | Вход/выход 5 порта C<br>Вход внешнего прерывания |
| | | Альтер. | SSP1_TXD | Выходные данные SPI 1 |
| | | Переопр. | SSP1_RXD | Входные данные SPI 1 |
| 62 | PC6 | Основ. | EXTINT2 | Вход/выход 6 порта C<br>Вход внешнего прерывания |
| | | Альтер. | SSP1_RXD | Входные данные SPI 1 |
| | | Переопр. | SSP1_TXD | Выходные данные SPI 1 |
| 63 | PC7 | Основ. | EXTINT3 | Вход/выход 5 порта C<br>Вход внешнего прерывания |
| | | Альтер. | SSP1_SCK | Вход/выход синхросигнала SPI 1 |
| | | Переопр. | FXEN | Выход разрешения передачи в оптический приёмопередатчик Ethernet 1 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 64 | PC8 | Основ. | EXTINT4 | Вход/выход 8 порта C<br>Вход внешнего прерывания |
| | | Альтер. | SSP1_FSS | Вход/выход разрешения SPI 1 |
| | | Переопр. | FTX | Выход передачи в оптический приёмопередатчик Ethernet 1 |
| 65 | PC9 | Основ. | SSP2_TXD | Вход/выход 9 порта C<br>Выходные данные SPI 2 |
| | | Альтер. | BE0 | Выход разрешения байта 0 в 32 разрядном слове |
| | | Переопр. | CAN_RX1 | Входные данные интерфейса CAN1 |
| 66 | PC10 | Основ. | SSP2_RXD | Вход/выход 10 порта C<br>Входные данные SPI 2 |
| | | Альтер. | BE1 | Выход разрешения байта 1 в 32 разрядном слове |
| | | Переопр. | CAN_TX1 | Выходные данные интерфейса CAN1 |
| 67 | PC11 | Основ. | SSP2_SCK | Вход/выход 11 порта C<br>Вход/выход синхросигнала SPI 2 |
| | | Альтер. | BE2 | Выход разрешения байта 2 в 32 разрядном слове |
| | | Переопр. | CAN_RX2 | Входные данные интерфейса CAN2 |
| 68 | PC12 | Основ. | SSP2_FSS | Вход/выход 12 порта C<br>Вход/выход разрешения SPI 2 |
| | | Альтер. | BE3 | Выход разрешения байта 3 в 32 разрядном слове |
| | | Переопр. | CAN_TX2 | Выходные данные интерфейса CAN2 |
| 72 | PC13 | Основ. | PRMA+ | Вход/выход 13 порта C<br>Положительный вход приёма канала A по ГОСТ Р 52070-2003 |
| | | Альтер. | A30 | Выход адреса 30 системной шины EXT_BUS |
| | | Переопр. | UART_TXD2 | Выходные данные UART2 |
| 73 | PC14 | Основ. | PRMA- | Вход/выход 14 порта C<br>Отрицательный вход приёма канала A по ГОСТ Р 52070-2003 |
| | | Альтер. | A31 | Выход адреса 31 системной шины EXT_BUS |
| | | Переопр. | UART_RXD2 | Входные данные UART2 |
| 74 | PC15 | Основ. | PRMB+ | Вход/выход 15 порта C<br>Положительный вход приёма канала B по ГОСТ Р 52070-2003 |
| | | Альтер. | BUSY | Вход занятости внешнего устройства системной шины EXT_BUS |
| | | Переопр. | TMR2_CH1 | Выход канала 1 таймера 2 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| Порт D | | | | |
| 81 | PD0 | Основ. | PRMB- | Вход/выход 0 порта D<br>Отрицательный вход приёма канала В по ГОСТ Р 52070-2003 |
| | | Альтер. | ALE | Выход разрешения защёлкивания адреса системной шины EXT_BUS |
| | | Переопр. | A16 | Выход адреса 16 системной шины EXT_BUS |
| 82 | PD1 | Основ. | PRDA+ | Вход/выход 1 порта D<br>Положительный выход передачи канала А по ГОСТ Р 52070-2003 |
| | | Альтер. | CLE | Выход разрешения защёлкивания команды системной шины EXT_BUS |
| | | Переопр. | A15 | Выход адреса 15 системной шины EXT_BUS |
| 83 | PD2 | Основ. | PRDA- | Вход/выход 2 порта D<br>Отрицательный выход передачи канала А по ГОСТ Р 52070-2003 |
| | | Альтер. | SSP1_TXD | Выходные данные SPI 1 |
| | | Переопр. | A14 | Выход адреса 14 системной шины EXT_BUS |
| 84 | PD3 | Основ. | PRDB+ | Вход/выход 3 порта D<br>Положительный выход передачи канала В по ГОСТ Р 52070-2003 |
| | | Альтер. | SSP1_RXD | Входные данные SPI 1 |
| | | Переопр. | A13 | Выход адреса 13 системной шины EXT_BUS |
| 85 | PD4 | Основ. | PRDB- | Вход/выход 4 порта D<br>Отрицательный выход передачи канала В по ГОСТ Р 52070-2003 |
| | | Альтер. | SSP1_SCK | Вход/выход синхросигнала SPI 1 |
| | | Переопр. | A7 | Выход адреса 7 системной шины EXT_BUS |
| 86 | PD5 | Основ. | PRD_PRMA | Вход/выход 5 порта D<br>Выход разрешения приёма/передачи канала А по ГОСТ Р 52070-2003 |
| | | Альтер. | SSP1_FSS | Вход/выход разрешения SPI 1 |
| | | Переопр. | A6 | Выход адреса 6 системной шины EXT_BUS |
| 87 | PD6 | Основ. | PRD_PRMB | Вход/выход 6 порта D<br>Выход разрешения приёма/передачи канала В по ГОСТ Р 52070-2003 |
| | | Альтер. | nUART2RI | Сигнал управления UART2 |
| | | Переопр. | A5 | Выход адреса 5 системной шины EXT_BUS |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 46 | PD7 | Аналог. | ADC0_REF+ | Вход/выход 7 порта D<br>Положительное опорное напряжение / Входной канал 0 SAR АЦП |
| | | Основ. | SSP2_TXD | Выходные данные SPI 2 |
| | | Альтер. | nUART2DCD | Сигнал управления UART2 |
| | | Переопр. | A4 | Выход адреса 4 системной шины EXT_BUS |
| 47 | PD8 | Аналог. | ADC1_REF- | Вход/выход 8 порта D<br>Отрицательное опорное напряжение / Входной канал 1 SAR АЦП |
| | | Основ. | SSP2_RXD | Входные данные SPI 2 |
| | | Альтер. | nUART2DTR | Сигнал управления UART2 |
| | | Переопр. | A3 | Выход адреса 3 системной шины EXT_BUS |
| 48 | PD9 | Аналог. | ADC2 | Вход/выход 9 порта D<br>Входной канал 2 SAR АЦП |
| | | Основ. | SSP2_SCK | Вход/выход синхросигнала SPI 2 |
| | | Альтер. | nUART2DSR | Сигнал управления UART2 |
| | | Переопр. | A2 | Выход адреса 2 системной шины EXT_BUS |
| 49 | PD10 | Аналог. | ADC3 | Вход/выход 10 порта D<br>Входной канал 3 SAR АЦП |
| | | Основ. | SSP2_FSS | Вход/выход разрешения SPI 2 |
| | | Альтер. | nUART2RTS | Сигнал управления UART2 |
| | | Переопр. | A1 | Выход адреса 1 системной шины EXT_BUS |
| 50 | PD11 | Аналог. | ADC4 | Вход/выход 11 порта D<br>Входной канал 4 SAR АЦП |
| | | Основ. | A0 | Выход адреса 0 системной шины EXT_BUS |
| | | Альтер. | nUART2CTS | Сигнал управления UART2 |
| | | Переопр. | FRX | Вход приёма с оптического приёмопередатчика Ethernet |
| 51 | PD12 | Аналог. | ADC5 | Вход/выход 12 порта D<br>Входной канал 5 SAR АЦП |
| | | Основ. | SSP3_TXD | Выходные данные SPI 3 |
| | | Альтер. | ETR3 | Вход внешнего события таймера 3 |
| | | Переопр. | SSP3_RXD | Входные данные SPI 3 |
| 52 | PD13 | Аналог. | ADC6 | Вход/выход 13 порта D<br>Входной канал 6 SAR АЦП |
| | | Основ. | UART_TXD2 | Выходные данные UART2 |
| | | Альтер. | OUT1+ | Положительный выход передачи канала 1 по ГОСТ 18977-79 |
| | | Переопр. | SIR_OUT2 | Выходные данные IRDA UART2 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 53 | PD14 | Аналог. | ADC7 | Вход/выход 14 порта D<br>Входной канал 7 SAR АЦП |
| | | Основ. | UART_RXD2 | Входные данные UART2 |
| | | Альтер. | OUT1- | Отрицательный выход передачи канала 1 по ГОСТ 18977-79 |
| | | Переопр. | SIR_IN2 | Входные данные IRDA UART1 |
| 54 | PD15 | Аналог. | REFD0 | Вход/выход 15 порта D<br>Положительное опорное напряжение ЦАП 0 |
| | | Основ. | OUT3+ | Положительный выход передачи канала 3 по ГОСТ 18977-79 |
| | | Альтер. | A13 | Выход адреса 13 системной шины EXT_BUS |
| | | Переопр. | FSD | Вход разрешения приёма с оптического приёмопередатчика Ethernet 1 |
| Порт E | | | | |
| 55 | PE0 | Аналог. | REFD1 | Вход/выход 0 порта E<br>Положительное опорное напряжение ЦАП 1 |
| | | Основ. | OUT4+ | Положительный выход передачи канала 4 по ГОСТ 18977-79 |
| | | Альтер. | A14 | Выход адреса 14 системной шины EXT_BUS |
| | | Переопр. | MDC | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| 56 | PE1 | Аналог. | DAC0 | Вход/выход 1 порта E<br>Аналоговый выход ЦАП 0 |
| | | Основ. | OUT3- | Отрицательный выход передачи канала 3 по ГОСТ 18977-79 |
| | | Альтер. | A15 | Выход адреса 15 системной шины EXT_BUS |
| | | Переопр. | nUART2RI | Сигнал управления UART1 |
| 57 | PE2 | Аналог. | DAC1 | Вход/выход 2 порта E<br>Аналоговый выход ЦАП 1 |
| | | Основ. | OUT4- | Отрицательный выход передачи канала 4 по ГОСТ 18977-79 |
| | | Альтер. | A16 | Выход адреса 16 системной шины EXT_BUS |
| | | Переопр. | MDIO | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| 18 | PE3 | Основ. | TMR1_CH1 | Вход/выход 3 порта E<br>Вход/выход канала 1 таймера 1 |
| | | Альтер. | A17 | Выход адреса 17 системной шины EXT_BUS |
| | | Переопр. | TXD[0] | Цифровой выход 0 передатчика Ethernet |
| 19 | PE4 | Основ. | TMR1_CH2 | Вход/выход 4 порта E<br>Вход/выход канала 2 таймера 1 |
| | | Альтер. | A18 | Выход адреса 18 системной шины EXT_BUS |
| | | Переопр. | TXD[1] | Цифровой выход 1 передатчика Ethernet |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 20 | PE5 | | | Вход/выход 5 порта E |
| | | Основ. | TMR1_CH3 | Вход/выход канала 3 таймера 1 |
| | | Альтер. | A19 | Выход адреса 19 системной шины EXT_BUS |
| | | Переопр. | TXD[2] | Цифровой выход 2 передатчика Ethernet |
| 37 | PE6 | | | Вход/выход 6 порта E |
| | | Аналог. | OSC_IN32 | Вход осциллятора 32 кГц |
| | | Основ. | TMR1_CH4 | Вход/выход канала 4 таймера 1 |
| | | Альтер. | A20 | Выход адреса 20 системной шины EXT_BUS |
| | | Переопр. | TXD[3] | Цифровой выход 3 передатчика Ethernet |
| 38 | PE7 | | | Вход/выход 7 порта E |
| | | Аналог. | OSC_OUT32 | Выход осциллятора 32 кГц |
| | | Основ. | TMR2_CH1 | Вход/выход канала 1 таймера 2 |
| | | Альтер. | A21 | Выход адреса 21 системной шины EXT_BUS |
| | | Переопр. | RXD[0] | Цифровой вход 0 передатчика Ethernet |
| 21 | PE8 | | | Вход/выход 8 порта E |
| | | Основ. | TMR2_CH2 | Вход/выход канала 2 таймера 2 |
| | | Альтер. | A22 | Выход адреса 22 системной шины EXT_BUS |
| | | Переопр. | RXD[1] | Цифровой вход 1 передатчика Ethernet |
| 22 | PE9 | | | Вход/выход 9 порта E |
| | | Основ. | TMR2_CH3 | Вход/выход канала 3 таймера 2 |
| | | Альтер. | A23 | Выход адреса 23 системной шины EXT_BUS |
| | | Переопр. | RXD[2] | Цифровой вход 2 передатчика Ethernet |
| 23 | PE10 | | | Вход/выход 10 порта E |
| | | Основ. | TMR2_CH4 | Вход/выход канала 2 таймера 4 |
| | | Альтер. | A24 | Выход адреса 24 системной шины EXT_BUS |
| | | Переопр. | RXD[3] | Цифровой вход 3 передатчика Ethernet |
| 24 | PE11 | | | Вход/выход 11 порта E |
| | | Основ. | CAN_RX1 | Входные данные интерфейса CAN1 |
| | | Альтер. | A25 | Выход адреса 25 системной шины EXT_BUS |
| | | Переопр. | TXEN | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY. При этом внутренний PHY должен находиться в состоянии сброса |
| 25 | PE12 | | | Вход/выход 12 порта E |
| | | Основ. | CAN_TX1 | Выходные данные интерфейса CAN1 |
| | | Альтер. | A26 | Выход адреса 26 системной шины EXT_BUS |
| | | Переопр. | TXER | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY. При этом внутренний PHY должен находиться в состоянии сброса |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 26 | PE13 | Основ. | CAN_RX2 | Вход/выход 13 порта E<br>Входные данные интерфейса CAN2 |
| | | Альтер. | A27 | Выход адреса 27 системной шины EXT_BUS |
| | | Переопр. | TXCLK | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| 27 | PE14 | Основ. | CAN_TX2 | Вход/выход 14 порта E<br>Выходные данные интерфейса CAN2 |
| | | Альтер. | A28 | Выход адреса 28 системной шины EXT_BUS |
| | | Переопр. | RXCLK | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| 28 | PE15 | Основ. | PRD_PRMD | Вход/выход 15 порта E<br>Выход разрешения приёма/передачи канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A29 | Выход адреса 29 системной шины EXT_BUS |
| | | Переопр. | RXDV | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| Порт F | | | | |
| 29 | PF0 | Аналог. | OSC_IN25 | Вход/выход 0 порта F<br>Вход осциллятора 25 МГц |
| | | Основ. | PRD_PRMA | Выход разрешения приёма/передачи канала A по ГОСТ Р 52070-2003 |
| | | Альтер. | READY | Вход готовности внешнего устройства системной шины EXT_BUS |
| | | Переопр. | RXER | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| 30 | PF1 | Аналог. | OSC_OUT25 | Вход/выход 1 порта F<br>Выход осциллятора 25 МГц |
| | | Основ. | PRD_PRMB | Выход разрешения приёма/передачи канала B по ГОСТ Р 52070-2003 |
| | | Альтер. | A30 | Выход адреса 30 системной шины EXT_BUS |
| | | Переопр. | CRS | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| 31 | PF2 | Основ. | READY /PRD_PRMC | Вход/выход 2 порта F<br>Вход готовности внешнего устройства системной шины EXT_BUS/<br>Выход разрешения приёма/передачи канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A31 | Выход адреса 31 системной шины EXT_BUS |
| | | Переопр. | COL | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 88 | PF3 | Основ. | PRMC+ | Вход/выход 3 порта F3<br>Положительный вход приёма канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A0 | Выход адреса 0 системной шины EXT_BUS |
| | | Переопр. | TMR1_CH1 | Вход/выход канала 1 таймера 1 |
| 89 | PF4 | Основ. | PRMC- | Вход/выход 4 порта F<br>Отрицательный вход приёма канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A1 | Выход адреса 1 системной шины EXT_BUS |
| | | Переопр. | TMR1_CH2 | Вход/выход канала 2 таймера 1 |
| 90 | PF5 | Основ. | PRMD+ | Вход/выход 5 порта FВход/выход канала 3 таймера 1<br>Положительный вход приёма канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A2 | Выход адреса 2 системной шины EXT_BUS |
| | | Переопр. | TMR1_CH3 | Вход/выход канала 3 таймера 1 |
| 91 | PF6 | Основ. | PRMD- | Вход/выход 6 порта F<br>Отрицательный вход приёма канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A3 | Выход адреса 3 системной шины EXT_BUS |
| | | Переопр. | TMR1_CH4 | Вход/выход канала 4 таймера 1 |
| 92 | PF7 | Основ. | PRDC+ | Вход/выход 7 порта F<br>Положительный выход передачи канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A4 | Выход адреса 4 системной шины EXT_BUS |
| | | Переопр. | OUT4+ | Положительный выход передачи канала 4 по ГОСТ 18977-79 |
| 93 | PF8 | Основ. | PRDC- | Вход/выход 8 порта F<br>Отрицательный выход передачи канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A5 | Выход адреса 5 системной шины EXT_BUS |
| | | Переопр. | OUT4- | Отрицательный выход передачи канала 4 по ГОСТ 18977-79 |
| 94 | PF9 | Основ. | PRDD+ | Вход/выход 9 порта F<br>Положительный выход передачи канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A6 | Выход адреса 6 системной шины EXT_BUS |
| | | Переопр. | OUT3+ | Положительный выход передачи канала 3 по ГОСТ 18977-79 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| 95 | PF10 | Основ. | PRDD- | Вход/выход 10 порта F<br>Отрицательный выход передачи канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A7 | Выход адреса 7 системной шины EXT_BUS |
| | | Переопр. | OUT3- | Отрицательный выход передачи канала 3 по ГОСТ 18977-79 |
| 96 | PF11 | Основ. | PRD_PRMC | Вход/выход 11 порта F<br>Выход разрешения приёма/передачи канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A8 | Выход адреса 8 системной шины EXT_BUS |
| | | Переопр. | OUT2+ | Положительный выход передачи канала 2 по ГОСТ 18977-79 |
| 97 | PF12 | Основ. | PRD_PRMD | Вход/выход 12 порта F<br>Выход разрешения приёма/передачи канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A9 | Выход адреса 9 системной шины EXT_BUS |
| | | Переопр. | OUT2- | Отрицательный выход передачи канала 2 по ГОСТ 18977-79 |
| 69 | PF13 | Основ. | OUT2+ | Вход/выход 13 порта F<br>Положительный выход передачи канала 2 по ГОСТ 18977-79 |
| | | Альтер. | A10 | Выход адреса 10 системной шины EXT_BUS |
| | | Переопр. | SSP3_FSS | Вход/выход разрешения SPI 3 |
| 70 | PF14 | Основ. | OUT2- | Вход/выход 14 порта F<br>Отрицательный выход передачи канала 2 по ГОСТ 18977-79 |
| | | Альтер. | A11 | Выход адреса 11 системной шины EXT_BUS |
| | | Переопр. | SSP3_SCK | Вход/выход синхросигнала SPI 3 |
| 71 | PF15 | Основ. | SSP3_RXD | Вход/выход 15 порта F<br>Входные данные SPI 3 |
| | | Альтер. | A12 | Выход адреса 12 системной шины EXT_BUS |
| | | Переопр. | SSP3_TXD | Выходные данные SPI 3 |
| Системное управление | | | | |
| 60 | nRESET | – | – | Сигнал внешнего сброса |
| 58 | OSC_IN | – | – | Вход генератора HSE |
| 59 | OSC_OUT | – | – | Выход генератора HSE |
| 39 | WAKEUP | – | – | Сигнал внешнего выхода из режима Standby |
| 17 | ITCMLAEN | – | – | Сигнал выбора памяти программ:<br>1- внутренняя;<br>0- внешняя |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| USB интерфейс | | | | |
| 42 | DP | – | – | Шина USB D+ |
| 43 | DN | – | – | Шина USB D- |
| PHY Ethernet интерфейс | | | | |
| 3 | TXP | – | – | Дифференциальный выход передатчика на трансформаторы |
| 2 | TXN | – | – | |
| 8 | RXP | – | – | Дифференциальный вход приёмника с трансформатора |
| 7 | RXN | – | – | |
| 12 | EXRES1 | – | – | Вывод для подключения опорного резистора 12,4 кОм 1 % на VSS2A |
| 6 | CM | – | – | Выход опорного напряжения приёмника |
| Питание | | | | |
| 33, 76, 127 | Ucc | – | – | Питание |
| 45 | AUcc | – | – | Аналоговое питание АЦП, ЦАП (должно совпадать с $U_{CC}$) |
| 5, 10, 14, 16 | VDD1A-VDD4A | – | – | Аналоговое питание PHY |
| 34 | BUcc | – | – | Батарейное питание |
| 32, 75, 125 | GND | – | – | Общий |
| 44 | AGND | – | – | Общий АЦП, ЦАП |
| 1, 4, 11, 13, 15 | VSS1A-VSS4A | – | – | Общий PHY |
| JTAG интерфейс | | | | |
| 138 | TMS | – | – | Выбор режима тестирования - обеспечивает переход схемы в/из режима тестирования и переключение между разными режимами тестирования |
| 139 | TDI | – | – | Вход тестовых данных - вход последовательных данных периферийного сканирования. Команды и данные вводятся в микросхему с этого вывода по переднему фронту сигнала TCK |
| 140 | TDO | – | – | Выход тестовых данных - выход последовательных данных. Команды и данные выводятся из микросхемы с этого вывода по заднему фронту сигнала TCK |
| 141 | TCK | – | – | Тестовое тактирование - тактирует работу встроенного автомата управления периферийным сканированием |
| 142 | TRST | – | – | Тестовый сброс - сигнал сброса встроенного автомата управления периферийным сканированием |
| 35 | JTAGEN | – | – | Вход разрешения отладочного TAP интерфейса на выводах JTAG |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| Тестовые выводы | | | | |
| 9 | TSTBUSA | – | – | Рекомендуется не подсоединять |
| 36 | VDD | – | – | Рекомендуется не подсоединять |
| 40 | SHDN | – | – | Рекомендуется не подсоединять |
| 41 | EXT_POR | – | – | Рекомендуется не подсоединять |
| 77 | VPP | – | – | Рекомендуется не подсоединять |
| 78 | TM1 | – | – | Рекомендуется не подсоединять |
| 79 | TM0 | – | – | Рекомендуется не подсоединять |
| 80 | TM2 | – | – | Рекомендуется не подсоединять |
| 126 | RST_BYP | – | – | Рекомендуется не подсоединять |

Примечания

1 Используемые обозначения функций:

альтер. – альтернативная;

аналог. – аналоговая;

основ. – основная;

переопр. – переопределенная;

2 При использовании внешнего Ethernet PHY, посредством портов PE[15:2], PE[0] и PF[2:0], внутренний Ethernet PHY должен находиться в состоянии сброса (см. описание Блока PHY);

3 При использовании режима работы с ITCMLAEN = 0 аппаратно (программно не модифицируются) порты A и B настраиваются как двунаправленная 32-разрядная шина данных, порт C[2:0] – как сигналы CLKO, nRD, nWR, а порты PF[15:3], PD[15], PE[2:0] – как шина адреса A[16:0]. Можно расширить программно адресную шину до требуемого количества разрядов;

4 Выводы в функции EXTINT1-EXTINT4 являются входами прерываний по уровню. Т.е. пока на вывод EXTINTx подан сигнал высокого уровня, генерируется прерывание. Если на входе сигнал низкого уровня, то прерывание не генерируется

Таблица 3 – Описание выводов микросхем К1986ВЕ1GI

| Номер вывода | Обозначение вывода | Тип вывода | Функциональное назначение |
|---|---|---|---|
| A1 | TRST | IO | Тестовый сброс – сигнал сброса встроенного автомата управления периферийным сканированием |
| A2 | TDO | O | Выход тестовых данных – выход последовательных данных. Команды и данные выводятся из микросхемы с этого вывода по заднему фронту сигнала TCK |
| A3 | TDI | I | Вход тестовых данных – вход последовательных данных периферийного сканирования. Команды и данные вводятся в микросхему с этого вывода по переднему фронту сигнала TCK |
| A4 | PA2/MODE[2] | IO | Порт A/Режим работы микросхемы |
| A5 | PA6 | IO | Порт A |
| A6 | $U_{CC}$ | PWR | Питание |
| A7 | PA12 | IO | Порт A |
| A8 | PA15 | | |
| A9 | PB3 | IO | Порт B |
| A10 | PB6 | | |
| A11 | PB8 | | |
| A12 | PB9 | | |
| B1, B2 | VSS1A | GND | Общий PHY |
| B3 | TCK | IO | Тестовое тактирование – тактирует работу встроенного автомата управления периферийным сканированием |
| B4 | PA1/MODE[1] | IO | Порт A/Режим работы микросхемы |
| B5 | PA5 | IO | Порт A |
| B6 | PA9 | | |
| B7 | PA13 | | |
| B8 | PB0 | IO | Порт B |
| B9 | PB4 | | |
| B10 | PB7 | | |
| B11 | PB11 | | |
| B12 | PB10 | | |
| C1 | TXP | O | Дифференциальный выход передатчика на трансформаторы |
| C2 | TXN | | |
| C3 | TMS | IO | Выбор режима тестирования - обеспечивает переход схемы в/из режима тестирования и переключение между разными режимами тестирования |
| C4 | PA0/MODE[0] | IO | Порт A/Режим работы микросхемы |
| C5 | PA4 | IO | Порт A |
| C6 | PA8 | | |
| C7 | PA11 | | |

| Номер вывода | Обозначение вывода | Тип вывода | Функциональное назначение |
|---|---|---|---|
| C8 | PB1 | IO | Порт B |
| C9 | PB5 | | |
| C10 | PB14 | | |
| C11 | PB12 | | |
| C12 | PB13 | | |
| D1 | RXP | I | Дифференциальный вход приёмника с трансформатора |
| D2 | RXN | | |
| D3 | VDD1A | PWR | Аналоговое питание PHY |
| D4 | PA3 | IO | Порт A |
| D5 | PA7 | | |
| D6 | PA10 | | |
| D7 | PA14 | | |
| D8 | PB2 | IO | Порт B |
| D9 | PC2 | IO | Порт C |
| D10 | PB15 | IO | Порт B |
| D11 | PC4 | IO | Порт C |
| D12 | PC3 | | |
| E1 | VSS3A | GND | Общий PHY |
| E2 | EXRES1 | – | Вывод для подключения опорного резистора 12,4 кОм 1 % на VSS2A |
| E3 | VSS2A | GND | Общий PHY |
| E4 | VDD2A | PWR | Аналоговое питание PHY |
| E5 – E8 | GND | GND | Общий |
| E9 | PF11 | IO | Порт F |
| E10 | PC1 | IO | Порт C |
| E11 | PC0 | | |
| E12 | PF12 | IO | Порт F |
| F1 | VDD4A | PWR | Аналоговое питание PHY |
| F2 | VSS4A | GND | Общий PHY |
| F3 | ITCMLAEN | I | Сигнал выбора памяти программ:<br>1 – внутренняя;<br>0 – внешняя |
| F4 | VDD3A | PWR | Аналоговое питание PHY |
| F5 – F8 | GND | GND | Общий |
| F9 | PF10 | IO | Порт F |
| F10 | PF7 | | |
| F11 | PF8 | | |
| F12 | PF9 | | |

| Номер вывода | Обозначение вывода | Тип вывода | Функциональное назначение |
|---|---|---|---|
| G1 | PE4 | IO | Порт E |
| G2 | PE5 | | |
| G3 | PE3 | | |
| G4 | PE8 | | |
| G5 – G8 | GND | GND | Общий |
| G9 | PF3 | IO | Порт F |
| G10 | PF6 | | |
| G11 | PF4 | | |
| G12 | PF5 | | |
| H1 | PE9 | IO | Порт E |
| H2 | PE11 | | |
| H3 | PE12 | | |
| H4 | PE10 | | |
| H5 – H8 | GND | GND | Общий |
| H9 | PD4 | IO | Порт D |
| H10 | PD3 | | |
| H11 | PD5 | | |
| H12 | PD6 | | |
| J1 | PE13 | IO | Порт E |
| J2 | PE14 | | |
| J3 | PF0 | IO | Порт F |
| J4 | PE15 | IO | Порт E |
| J5 | PD9 | IO | Порт D |
| J6 | PD12 | | |
| J7 | OSC_IN | I | Генератор HSE |
| J8 | OSC_OUT | O | |
| J9 | PC10 | IO | Порт C |
| J10 | PD0 | IO | Порт D |
| J11 | PD1 | | |
| J12 | PD2 | | |
| K1 | PF1 | IO | Порт F |
| K2 | PF2 | | |
| K3 | $U_{CC}$ | PWR | Питание |
| K4 | AUcc | PWR | Аналоговое питание АЦП, ЦАП (должно совпадать с $U_{CC}$) |
| K5 | PD8 | IO | Порт D |
| K6 | PD15 | | |
| K7 | PE0 | IO | Порт E |

| Номер вывода | Обозначение вывода | Тип вывода | Функциональное назначение |
|---|---|---|---|
| K8 | PC6 | IO | Порт C |
| K9 | PC7 | | |
| K10 | PC11 | | |
| K11 | PC13 | | |
| K12 | PC15 | | |
| L1 | BUcc | PWR | Батарейное питание |
| L2 | PE7 | IO | Порт E |
| L3 | PE6 | | |
| L4 | PD7 | IO | Порт D |
| L5 | PD10 | | |
| L6 | PD13 | | |
| L7 | PE1 | IO | Порт E |
| L8 | nRESET | I | Сигнал внешнего сброса |
| L9 | PC8 | IO | Порт C |
| L10 | PF13 | IO | Порт F |
| L11 | $U_{CC}$ | PWR | Питание |
| L12 | PC14 | IO | Порт C |
| M1 | JTAGEN | I | Вход разрешения отладочного TAP интерфейса на выводах JTAG |
| M2 | WAKEUP | I | Сигнал внешнего выхода из режима Standby |
| M3 | DP | IO | Шина USB D+ |
| M4 | DN | | Шина USB D- |
| M5 | PD11 | IO | Порт D |
| M6 | PD14 | | |
| M7 | PE2 | IO | Порт E |
| M8 | PC5 | IO | Порт C |
| M9 | PC9 | | |
| M10 | PC12 | | |
| M11 | PF14 | IO | Порт F |
| M12 | PF15 | | |

Примечание – Обозначение типа выводов:

IO – цифровой вход/выход;

I – цифровой вход;

O – цифровой выход

Таблица 4 – Дополнительные функции выводов микросхем К1986ВЕ1GI

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| Порт А/Режим работы микросхемы | | | | |
| C4 | PA0 / MODE[0] | Основ. | D0 | Вход/выход 0 порта А<br>Вход/выход данных 0 системной шины EXT_BUS |
| | | Альтер. | EXTINT1 | Вход внешнего прерывания |
| | | Переопр. | ETR1 | Вход внешнего события таймера |
| B4 | PA1 / MODE[1] | Основ. | D1 | Вход/выход 1 порта А<br>Вход/выход данных 1 системной шины EXT_BUS |
| | | Альтер. | EXTINT2 | Вход внешнего прерывания |
| | | Переопр. | ETR2 | Вход внешнего события таймера 2 |
| A4 | PA2 / MODE[2] | Основ. | D2 | Вход/выход 2 порта А<br>Вход/выход данных 2 системной шины EXT_BUS |
| | | Альтер. | EXTINT3 | Вход внешнего прерывания |
| | | Переопр. | ETR3 | Вход внешнего события таймера 3 |
| Порт А | | | | |
| D4 | PA3 | Основ. | D3 | Вход/выход 3 порта А<br>Вход/выход данных 3 системной шины EXT_BUS |
| | | Альтер. | EXTINT4 | Вход внешнего прерывания |
| | | Переопр. | BRK1 | Вход внешнего события таймера 1 |
| C5 | PA4 | Основ. | D4 | Вход/выход 4 порта А<br>Вход/выход данных 4 системной шины EXT_BUS |
| | | Альтер. | BRK2 | Вход внешнего события таймера 2 |
| | | Переопр. | FRX | Вход приёма с оптического приёмопередатчика Ethernet 1 |
| B5 | PA5 | Основ. | D5 | Вход/выход 5 порта А<br>Вход/выход данных 5 системной шины EXT_BUS |
| | | Альтер. | BRK3 | Вход внешнего события таймера 3 |
| | | Переопр. | FSD | Вход разрешения приёма с оптического приёмопередатчика Ethernet 1 |
| A5 | PA6 | Основ. | D6 | Вход/выход 6 порта А<br>Вход/выход данных 6 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH1 | Вход/выход канала 1 таймера 4 |
| | | Переопр. | FXEN | Выход разрешения передачи в оптический приёмопередатчик Ethernet 1 |
| D5 | PA7 | Основ. | D7 | Вход/выход 7 порта А<br>Вход/выход данных 7 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH1N | Инверсный выход канала 1 таймера 4 |
| | | Переопр. | FTX | Выход передачи в оптический приёмопередатчик Ethernet 1 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| C6 | PA8 | Основ. | D8 | Вход/выход 8 порта A<br>Вход/выход данных 8 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH2 | Вход/выход канала 2 таймера 4 |
| | | Переопр. | PRMC+ | Положительный вход приёма канала C по ГОСТ Р 52070-2003 |
| B6 | PA9 | Основ. | D9 | Вход/выход 9 порта A<br>Вход/выход данных 9 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH2N | Инверсный выход канала 2 таймера 4 |
| | | Переопр. | PRMC- | Отрицательный вход приёма канала C по ГОСТ Р 52070-2003 |
| D6 | PA10 | Основ. | D10 | Вход/выход 10 порта A<br>Вход/выход данных 10 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH3 | Вход/выход канала 3 таймера 4 |
| | | Переопр. | PRMD+ | Положительный вход приёма канала D по ГОСТ Р 52070-2003 |
| C7 | PA11 | Основ. | D11 | Вход/выход 11 порта A<br>Вход/выход данных 11 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH3N | Инверсный выход канала 3 таймера 4 |
| | | Переопр. | PRMD- | Отрицательный вход приёма канала D по ГОСТ Р 52070-2003 |
| A7 | PA12 | Основ. | D12 | Вход/выход 12 порта A<br>Вход/выход данных 12 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH4 | Вход/выход канала 4 таймера 4 |
| | | Переопр. | PRDC+ | Положительный выход передачи канала C по ГОСТ Р 52070-2003 |
| B7 | PA13 | Основ. | D13 | Вход/выход 13 порта A<br>Вход/выход данных 13 системной шины EXT_BUS |
| | | Альтер. | TMR4_CH4N | Инверсный выход канала 4 таймера 4 |
| | | Переопр. | PRDC- | Отрицательный выход передачи канала C по ГОСТ Р 52070-2003 |
| D7 | PA14 | Основ. | D14 | Вход/выход 14 порта A<br>Вход/выход данных 14 системной шины EXT_BUS |
| | | Альтер. | BRK4 | Вход внешнего события таймера 4 |
| | | Переопр. | PRDD+ | Положительный выход передачи канала D по ГОСТ Р 52070-2003 |
| A8 | PA15 | Основ. | D15 | Вход/выход 15 порта A<br>Вход/выход данных 15 системной шины EXT_BUS |
| | | Альтер. | ETR4 | Вход внешнего события таймера 4 |
| | | Переопр. | PRDD- | Отрицательный выход передачи канала D по ГОСТ Р 52070-2003 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| Порт B | | | | |
| B8 | PB0 | | | Вход/выход 0 порта B |
| | | Основ. | D16 | Вход/выход данных 16 системной шины EXT_BUS |
| | | Альтер. | IN1+ | Положительный вход приёма канала 1 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH1 | Вход/выход канала 1 таймера 3 |
| C8 | PB1 | | | Вход/выход 1 порта B |
| | | Основ. | D17 | Вход/выход данных 17 системной шины EXT_BUS |
| | | Альтер. | IN1- | Отрицательный вход приёма канала 1 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH1N | Инверсный выход канала 1 таймера 3 |
| D8 | PB2 | | | Вход/выход 2 порта B |
| | | Основ. | D18 | Вход/выход данных 18 системной шины EXT_BUS |
| | | Альтер. | IN2+ | Положительный вход приёма канала 2 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH2 | Вход/выход канала 2 таймера 3 |
| A9 | PB3 | | | Вход/выход 3 порта B |
| | | Основ. | D19 | Вход/выход данных 19 системной шины EXT_BUS |
| | | Альтер. | IN2- | Отрицательный вход приёма канала 2 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH2N | Инверсный выход канала 2 таймера 3 |
| B9 | PB4 | | | Вход/выход 4 порта B |
| | | Основ. | D20 | Вход/выход данных 20 системной шины EXT_BUS |
| | | Альтер. | IN3+ | Положительный вход приёма канала 3 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH3 | Вход/выход канала 3 таймера 3 |
| C9 | PB5 | | | Вход/выход 5 порта B |
| | | Основ. | D21 | Вход/выход данных 21 системной шины EXT_BUS |
| | | Альтер. | IN3- | Отрицательный вход приёма канала 3 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH3N | Инверсный выход канала 3 таймера 3 |
| A10 | PB6 | | | Вход/выход 6 порта B |
| | | Основ. | D22 | Вход/выход данных 22 системной шины EXT_BUS |
| | | Альтер. | IN4+ | Положительный вход приёма канала 4 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH4 | Вход/выход канала 4 таймера 3 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| B10 | PB7 | Основ. | D23 | Вход/выход 7 порта B<br>Вход/выход данных 23 системной шины EXT_BUS |
| | | Альтер. | IN4- | Отрицательный вход приёма канала 4 по ГОСТ 18977-79 |
| | | Переопр. | TMR3_CH4N | Инверсный выход канала 4 таймера 3 |
| A11 | PB8 | Основ. | D24 | Вход/выход 8 порта B<br>Вход/выход данных 24 системной шины EXT_BUS |
| | | Альтер. | IN5+ | Положительный вход приёма канала 5 по ГОСТ 18977-79 |
| | | Переопр. | TMR1_CH1N | Инверсный выход канала 1 таймера 1 |
| A12 | PB9 | Основ. | D25 | Вход/выход 9 порта B<br>Вход/выход данных 25 системной шины EXT_BUS |
| | | Альтер. | IN5- | Отрицательный вход приёма канала 5 по ГОСТ 18977-79 |
| | | Переопр. | TMR2_CH1N | Инверсный выход канала 1 таймера 2 |
| B12 | PB10 | Основ. | D26 | Вход/выход 10 порта B<br>Вход/выход данных 26 системной шины EXT_BUS |
| | | Альтер. | IN6+ | Положительный вход приёма канала 6 по ГОСТ 18977-79 |
| | | Переопр. | TMR1_CH2N | Инверсный выход канала 2 таймера 1 |
| B11 | PB11 | Основ. | D27 | Вход/выход 11 порта B<br>Вход/выход данных 27 системной шины EXT_BUS |
| | | Альтер. | IN6- | Отрицательный вход приёма канала 6 по ГОСТ 18977-79 |
| | | Переопр. | TMR2_CH2N | Инверсный выход канала 2 таймера 2 |
| C11 | PB12 | Основ. | D28 | Вход/выход 12 порта B<br>Вход/выход данных 28 системной шины EXT_BUS |
| | | Альтер. | IN7+ | Положительный вход приёма канала 7 по ГОСТ 18977-79 |
| | | Переопр. | TMR1_CH3N | Инверсный выход канала 3 таймера 1 |
| C12 | PB13 | Основ. | D29 | Вход/выход 13 порта B<br>Вход/выход данных 29 системной шины EXT_BUS |
| | | Альтер. | IN7- | Отрицательный вход приёма канала 7 по ГОСТ 18977-79 |
| | | Переопр. | TMR2_CH3N | Инверсный выход канала 3 таймера 2 |
| C10 | PB14 | Основ. | D30 | Вход/выход 14 порта B<br>Вход/выход данных 30 системной шины EXT_BUS |
| | | Альтер. | IN8+ | Положительный вход приёма канала 8 по ГОСТ 18977-79 |
| | | Переопр. | TMR1_CH4N | Инверсный выход канала 4 таймера 1 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| D10 | PB15 | Основ. | D31 | Вход/выход 15 порта B<br>Вход/выход данных 31 системной шины EXT_BUS |
| | | Альтер. | IN8- | Отрицательный вход приёма канала 8 по ГОСТ 18977-79 |
| | | Переопр. | TMR2_CH4N | Инверсный выход канала 4 таймера 2 |
| Порт C | | | | |
| E11 | PC0 | Основ. | nWR | Вход/выход 0 порта C<br>Выход разрешения записи системной шины EXT_BUS |
| | | Альтер. | ETR1 | Входы внешнего события таймера 1 |
| | | Переопр. | BRK1 | |
| E10 | PC1 | Основ. | nRD | Вход/выход 1 порта C<br>Выход разрешения чтения системной шины EXT_BUS |
| | | Альтер. | ETR2 | Входы внешнего события таймера 2 |
| | | Переопр. | BRK2 | |
| D9 | PC2 | Основ. | ALE | Вход/выход 2 порта C<br>Выход разрешения защёлкивания адреса системной шины EXT_BUS |
| | | Альтер. | CLKO | Выход синхросигнала системной шины EXT_BUS |
| | | Переопр. | BRK3 | Вход внешнего события таймера 3 |
| D12 | PC3 | Основ. | UART_TXD1 | Вход/выход 3 порта C<br>Выходные данные UART1 |
| | | Альтер. | CLE | Выход разрешения защёлкивания команды системной шины EXT_BUS |
| | | Переопр. | SIR_OUT1 | Выходные данные IRDA UART1 |
| D11 | PC4 | Основ. | UART_RXD1 | Вход/выход 4 порта C<br>Входные данные UART1 |
| | | Альтер. | BUSY | Вход занятости внешнего устройства системной шины EXT_BUS |
| | | Переопр. | SIR_IN1 | Входные данные IRDA UART1 |
| M8 | PC5 | Основ. | EXTINT1 | Вход/выход 5 порта C<br>Вход внешнего прерывания |
| | | Альтер. | SSP1_TXD | Выходные данные SPI 1 |
| | | Переопр. | SSP1_RXD | Входные данные SPI 1 |
| K8 | PC6 | Основ. | EXTINT2 | Вход/выход 6 порта C<br>Вход внешнего прерывания |
| | | Альтер. | SSP1_RXD | Входные данные SPI 1 |
| | | Переопр. | SSP1_TXD | Выходные данные SPI 1 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| K9 | PC7 | Основ. | EXTINT3 | Вход/выход 5 порта C<br>Вход внешнего прерывания |
| | | Альтер. | SSP1_SCK | Вход/выход синхросигнала SPI 1 |
| | | Переопр. | FXEN | Выход разрешения передачи в оптический приёмопередатчик Ethernet 1 |
| L9 | PC8 | Основ. | EXTINT4 | Вход/выход 8 порта C<br>Вход внешнего прерывания |
| | | Альтер. | SSP1_FSS | Вход/выход разрешения SPI 1 |
| | | Переопр. | FTX | Выход передачи в оптический приёмопередатчик Ethernet 1 |
| M9 | PC9 | Основ. | SSP2_TXD | Вход/выход 9 порта C<br>Выходные данные SPI 2 |
| | | Альтер. | BE0 | Выход разрешения байта 0 в 32 разрядном слове |
| | | Переопр. | CAN_RX1 | Входные данные интерфейса CAN1 |
| J9 | PC10 | Основ. | SSP2_RXD | Вход/выход 10 порта C<br>Входные данные SPI 2 |
| | | Альтер. | BE1 | Выход разрешения байта 1 в 32 разрядном слове |
| | | Переопр. | CAN_TX1 | Выходные данные интерфейса CAN1 |
| K10 | PC11 | Основ. | SSP2_SCK | Вход/выход 11 порта C<br>Вход/выход синхросигнала SPI 2 |
| | | Альтер. | BE2 | Выход разрешения байта 2 в 32 разрядном слове |
| | | Переопр. | CAN_RX2 | Входные данные интерфейса CAN2 |
| M10 | PC12 | Основ. | SSP2_FSS | Вход/выход 12 порта C<br>Вход/выход разрешения SPI 2 |
| | | Альтер. | BE3 | Выход разрешения байта 3 в 32 разрядном слове |
| | | Переопр. | CAN_TX2 | Выходные данные интерфейса CAN2 |
| K11 | PC13 | Основ. | PRMA+ | Вход/выход 13 порта C<br>Положительный вход приёма канала А по ГОСТ Р 52070-2003 |
| | | Альтер. | A30 | Выход адреса 30 системной шины EXT_BUS |
| | | Переопр. | UART_TXD2 | Выходные данные UART2 |
| L12 | PC14 | Основ. | PRMA- | Вход/выход 14 порта C<br>Отрицательный вход приёма канала А по ГОСТ Р 52070-2003 |
| | | Альтер. | A31 | Выход адреса 31 системной шины EXT_BUS |
| | | Переопр. | UART_RXD2 | Входные данные UART2 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| K12 | PC15 | Основ. | PRMB+ | Вход/выход 15 порта C<br>Положительный вход приёма канала B по ГОСТ Р 52070-2003 |
| | | Альтер. | BUSY | Вход занятости внешнего устройства системной шины EXT_BUS |
| | | Переопр. | TMR2_CH1 | Выход канала 1 таймера 2 |
| Порт D | | | | |
| J10 | PD0 | Основ. | PRMB- | Вход/выход 0 порта D<br>Отрицательный вход приёма канала B по ГОСТ Р 52070-2003 |
| | | Альтер. | ALE | Выход разрешения защёлкивания адреса системной шины EXT_BUS |
| | | Переопр. | A16 | Выход адреса 16 системной шины EXT_BUS |
| J11 | PD1 | Основ. | PRDA+ | Вход/выход 1 порта D<br>Положительный выход передачи канала A по ГОСТ Р 52070-2003 |
| | | Альтер. | CLE | Выход разрешения защёлкивания команды системной шины EXT_BUS |
| | | Переопр. | A15 | Выход адреса 15 системной шины EXT_BUS |
| J12 | PD2 | Основ. | PRDA- | Вход/выход 2 порта D<br>Отрицательный выход передачи канала A по ГОСТ Р 52070-2003 |
| | | Альтер. | SSP1_TXD | Выходные данные SPI 1 |
| | | Переопр. | A14 | Выход адреса 14 системной шины EXT_BUS |
| H10 | PD3 | Основ. | PRDB+ | Вход/выход 3 порта D<br>Положительный выход передачи канала B по ГОСТ Р 52070-2003 |
| | | Альтер. | SSP1_RXD | Входные данные SPI 1 |
| | | Переопр. | A13 | Выход адреса 13 системной шины EXT_BUS |
| H9 | PD4 | Основ. | PRDB- | Вход/выход 4 порта D<br>Отрицательный выход передачи канала B по ГОСТ Р 52070-2003 |
| | | Альтер. | SSP1_SCK | Вход/выход синхросигнала SPI 1 |
| | | Переопр. | A7 | Выход адреса 7 системной шины EXT_BUS |
| H11 | PD5 | Основ. | PRD_PRMA | Вход/выход 5 порта D<br>Выход разрешения приёма/передачи канала A по ГОСТ Р 52070-2003 |
| | | Альтер. | SSP1_FSS | Вход/выход разрешения SPI 1 |
| | | Переопр. | A6 | Выход адреса 6 системной шины EXT_BUS |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| H12 | PD6 | Основ. | PRD_PRMB | Вход/выход 6 порта D<br>Выход разрешения приёма/передачи канала В по ГОСТ Р 52070-2003 |
| | | Альтер. | nUART2RI | Сигнал управления UART2 |
| | | Переопр. | A5 | Выход адреса 5 системной шины EXT_BUS |
| L4 | PD7 | Аналог. | ADC0_REF+ | Вход/выход 7 порта D<br>Положительное опорное напряжение / Входной канал 0 SAR АЦП |
| | | Основ. | SSP2_TXD | Выходные данные SPI 2 |
| | | Альтер. | nUART2DCD | Сигнал управления UART2 |
| | | Переопр. | A4 | Выход адреса 4 системной шины EXT_BUS |
| K5 | PD8 | Аналог. | ADC1_REF- | Вход/выход 8 порта D<br>Отрицательное опорное напряжение / Входной канал 1 SAR АЦП |
| | | Основ. | SSP2_RXD | Входные данные SPI 2 |
| | | Альтер. | nUART2DTR | Сигнал управления UART2 |
| | | Переопр. | A3 | Выход адреса 3 системной шины EXT_BUS |
| J5 | PD9 | Аналог. | ADC2 | Вход/выход 9 порта D<br>Входной канал 2 SAR АЦП |
| | | Основ. | SSP2_SCK | Вход/выход синхросигнала SPI 2 |
| | | Альтер. | nUART2DSR | Сигнал управления UART2 |
| | | Переопр. | A2 | Выход адреса 2 системной шины EXT_BUS |
| L5 | PD10 | Аналог. | ADC3 | Вход/выход 10 порта D<br>Входной канал 3 SAR АЦП |
| | | Основ. | SSP2_FSS | Вход/выход разрешения SPI 2 |
| | | Альтер. | nUART2RTS | Сигнал управления UART2 |
| | | Переопр. | A1 | Выход адреса 1 системной шины EXT_BUS |
| M5 | PD11 | Аналог. | ADC4 | Вход/выход 11 порта D<br>Входной канал 4 SAR АЦП |
| | | Основ. | A0 | Выход адреса 0 системной шины EXT_BUS |
| | | Альтер. | nUART2CTS | Сигнал управления UART2 |
| | | Переопр. | FRX | Вход приёма с оптического приёмопередатчика Ethernet |
| J6 | PD12 | Аналог. | ADC5 | Вход/выход 12 порта D<br>Входной канал 5 SAR АЦП |
| | | Основ. | SSP3_TXD | Выходные данные SPI 3 |
| | | Альтер. | ETR3 | Вход внешнего события таймера 3 |
| | | Переопр. | SSP3_RXD | Входные данные SPI 3 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| L6 | PD13 | | | Вход/выход 13 порта D |
| | | Аналог. | ADC6 | Входной канал 6 SAR АЦП |
| | | Основ. | UART_TXD2 | Выходные данные UART2 |
| | | Альтер. | OUT1+ | Положительный выход передачи канала 1 по ГОСТ 18977-79 |
| | | Переопр. | SIR_OUT2 | Выходные данные IRDA UART2 |
| M6 | PD14 | | | Вход/выход 14 порта D |
| | | Аналог. | ADC7 | Входной канал 7 SAR АЦП |
| | | Основ. | UART_RXD2 | Входные данные UART2 |
| | | Альтер. | OUT1- | Отрицательный выход передачи канала 1 по ГОСТ 18977-79 |
| | | Переопр. | SIR_IN2 | Входные данные IRDA UART1 |
| K6 | PD15 | | | Вход/выход 15 порта D |
| | | Аналог. | REFD0 | Положительное опорное напряжение ЦАП 0 |
| | | Основ. | OUT3+ | Положительный выход передачи канала 3 по ГОСТ 18977-79 |
| | | Альтер. | A13 | Выход адреса 13 системной шины EXT_BUS |
| | | Переопр. | FSD | Вход разрешения приёма с оптического приёмопередатчика Ethernet 1 |
| Порт E | | | | |
| K7 | PE0 | | | Вход/выход 0 порта E |
| | | Аналог. | REFD1 | Положительное опорное напряжение ЦАП 1 |
| | | Основ. | OUT4+ | Положительный выход передачи канала 4 по ГОСТ 18977-79 |
| | | Альтер. | A14 | Выход адреса 14 системной шины EXT_BUS |
| | | Переопр. | MDC | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| L7 | PE1 | | | Вход/выход 1 порта E |
| | | Аналог. | DAC0 | Аналоговый выход ЦАП 0 |
| | | Основ. | OUT3- | Отрицательный выход передачи канала 3 по ГОСТ 18977-79 |
| | | Альтер. | A15 | Выход адреса 15 системной шины EXT_BUS |
| | | Переопр. | nUART2RI | Сигнал управления UART1 |
| M7 | PE2 | | | Вход/выход 2 порта E |
| | | Аналог. | DAC1 | Аналоговый выход ЦАП 1 |
| | | Основ. | OUT4- | Отрицательный выход передачи канала 4 по ГОСТ 18977-79 |
| | | Альтер. | A16 | Выход адреса 16 системной шины EXT_BUS |
| | | Переопр. | MDIO | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| G3 | PE3 | | | Вход/выход 3 порта E |
| | | Основ. | TMR1_CH1 | Вход/выход канала 1 таймера 1 |
| | | Альтер. | A17 | Выход адреса 17 системной шины EXT_BUS |
| | | Переопр. | TXD[0] | Цифровой выход 0 передатчика Ethernet |
| G1 | PE4 | | | Вход/выход 4 порта E |
| | | Основ. | TMR1_CH2 | Вход/выход канала 2 таймера 1 |
| | | Альтер. | A18 | Выход адреса 18 системной шины EXT_BUS |
| | | Переопр. | TXD[1] | Цифровой выход 1 передатчика Ethernet |
| G2 | PE5 | | | Вход/выход 5 порта E |
| | | Основ. | TMR1_CH3 | Вход/выход канала 3 таймера 1 |
| | | Альтер. | A19 | Выход адреса 19 системной шины EXT_BUS |
| | | Переопр. | TXD[2] | Цифровой выход 2 передатчика Ethernet |
| L3 | PE6 | | | Вход/выход 6 порта E |
| | | Аналог. | OSC_IN32 | Вход осциллятора 32 кГц |
| | | Основ. | TMR1_CH4 | Вход/выход канала 4 таймера 1 |
| | | Альтер. | A20 | Выход адреса 20 системной шины EXT_BUS |
| | | Переопр. | TXD[3] | Цифровой выход 3 передатчика Ethernet |
| L2 | PE7 | | | Вход/выход 7 порта E |
| | | Аналог. | OSC_OUT32 | Выход осциллятора 32 кГц |
| | | Основ. | TMR2_CH1 | Вход/выход канала 1 таймера 2 |
| | | Альтер. | A21 | Выход адреса 21 системной шины EXT_BUS |
| | | Переопр. | RXD[0] | Цифровой вход 0 передатчика Ethernet |
| G4 | PE8 | | | Вход/выход 8 порта E |
| | | Основ. | TMR2_CH2 | Вход/выход канала 2 таймера 2 |
| | | Альтер. | A22 | Выход адреса 22 системной шины EXT_BUS |
| | | Переопр. | RXD[1] | Цифровой вход 1 передатчика Ethernet |
| H1 | PE9 | | | Вход/выход 9 порта E |
| | | Основ. | TMR2_CH3 | Вход/выход канала 3 таймера 2 |
| | | Альтер. | A23 | Выход адреса 23 системной шины EXT_BUS |
| | | Переопр. | RXD[2] | Цифровой вход 2 передатчика Ethernet |
| H4 | PE10 | | | Вход/выход 10 порта E |
| | | Основ. | TMR2_CH4 | Вход/выход канала 2 таймера 4 |
| | | Альтер. | A24 | Выход адреса 24 системной шины EXT_BUS |
| | | Переопр. | RXD[3] | Цифровой вход 3 передатчика Ethernet |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| H2 | PE11 | Основ. | CAN_RX1 | Вход/выход 11 порта E<br>Входные данные интерфейса CAN1 |
| | | Альтер. | A25 | Выход адреса 25 системной шины EXT_BUS |
| | | Переопр. | TXEN | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY. При этом внутренний PHY должен находиться в состоянии сброса |
| H3 | PE12 | Основ. | CAN_TX1 | Вход/выход 12 порта E<br>Выходные данные интерфейса CAN1 |
| | | Альтер. | A26 | Выход адреса 26 системной шины EXT_BUS |
| | | Переопр. | TXER | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY. При этом внутренний PHY должен находиться в состоянии сброса |
| J1 | PE13 | Основ. | CAN_RX2 | Вход/выход 13 порта E<br>Входные данные интерфейса CAN2 |
| | | Альтер. | A27 | Выход адреса 27 системной шины EXT_BUS |
| | | Переопр. | TXCLK | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| J2 | PE14 | Основ. | CAN_TX2 | Вход/выход 14 порта E<br>Выходные данные интерфейса CAN2 |
| | | Альтер. | A28 | Выход адреса 28 системной шины EXT_BUS |
| | | Переопр. | RXCLK | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| J4 | PE15 | Основ. | PRD_PRMD | Вход/выход 15 порта E<br>Выход разрешения приёма/передачи канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A29 | Выход адреса 29 системной шины EXT_BUS |
| | | Переопр. | RXDV | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| Порт F | | | | |
| J3 | PF0 | Аналог. | OSC_IN25 | Вход/выход 0 порта F<br>Вход осциллятора 25 МГц |
| | | Основ. | PRD_PRMA | Выход разрешения приёма/передачи канала A по ГОСТ Р 52070-2003 |
| | | Альтер. | READY | Вход готовности внешнего устройства системной шины EXT_BUS |
| | | Переопр. | RXER | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| K1 | PF1 | | | Вход/выход 1 порта F |
| | | Аналог. | OSC_OUT25 | Выход осциллятора 25 МГц |
| | | Основ. | PRD_PRMB | Выход разрешения приёма/передачи канала B по ГОСТ Р 52070-2003 |
| | | Альтер. | A30 | Выход адреса 30 системной шины EXT_BUS |
| | | Переопр. | CRS | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| K2 | PF2 | | | Вход/выход 2 порта F |
| | | Основ. | READY /PRD_PRMC | Вход готовности внешнего устройства системной шины EXT_BUS/ Выход разрешения приёма/передачи канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A31 | Выход адреса 31 системной шины EXT_BUS |
| | | Переопр. | COL | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| G9 | PF3 | | | Вход/выход 3 порта F3 |
| | | Основ. | PRMC+ | Положительный вход приёма канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A0 | Выход адреса 0 системной шины EXT_BUS |
| | | Переопр. | TMR1_CH1 | Вход/выход канала 1 таймера 1 |
| G11 | PF4 | | | Вход/выход 4 порта F |
| | | Основ. | PRMC- | Отрицательный вход приёма канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A1 | Выход адреса 1 системной шины EXT_BUS |
| | | Переопр. | TMR1_CH2 | Вход/выход канала 2 таймера 1 |
| G12 | PF5 | | | Вход/выход 5 порта FBВход/выход канала 3 таймера 1 |
| | | Основ. | PRMD+ | Положительный вход приёма канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A2 | Выход адреса 2 системной шины EXT_BUS |
| | | Переопр. | TMR1_CH3 | Вход/выход канала 3 таймера 1 |
| G10 | PF6 | | | Вход/выход 6 порта F |
| | | Основ. | PRMD- | Отрицательный вход приёма канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A3 | Выход адреса 3 системной шины EXT_BUS |
| | | Переопр. | TMR1_CH4 | Вход/выход канала 4 таймера 1 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| F10 | PF7 | Основ. | PRDC+ | Вход/выход 7 порта F<br>Положительный выход передачи канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A4 | Выход адреса 4 системной шины EXT_BUS |
| | | Переопр. | OUT4+ | Положительный выход передачи канала 4 по ГОСТ 18977-79 |
| F11 | PF8 | Основ. | PRDC- | Вход/выход 8 порта F<br>Отрицательный выход передачи канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A5 | Выход адреса 5 системной шины EXT_BUS |
| | | Переопр. | OUT4- | Отрицательный выход передачи канала 4 по ГОСТ 18977-79 |
| F12 | PF9 | Основ. | PRDD+ | Вход/выход 9 порта F<br>Положительный выход передачи канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A6 | Выход адреса 6 системной шины EXT_BUS |
| | | Переопр. | OUT3+ | Положительный выход передачи канала 3 по ГОСТ 18977-79 |
| F9 | PF10 | Основ. | PRDD- | Вход/выход 10 порта F<br>Отрицательный выход передачи канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A7 | Выход адреса 7 системной шины EXT_BUS |
| | | Переопр. | OUT3- | Отрицательный выход передачи канала 3 по ГОСТ 18977-79 |
| E9 | PF11 | Основ. | PRD_PRMC | Вход/выход 11 порта F<br>Выход разрешения приёма/передачи канала C по ГОСТ Р 52070-2003 |
| | | Альтер. | A8 | Выход адреса 8 системной шины EXT_BUS |
| | | Переопр. | OUT2+ | Положительный выход передачи канала 2 по ГОСТ 18977-79 |
| E12 | PF12 | Основ. | PRD_PRMD | Вход/выход 12 порта F<br>Выход разрешения приёма/передачи канала D по ГОСТ Р 52070-2003 |
| | | Альтер. | A9 | Выход адреса 9 системной шины EXT_BUS |
| | | Переопр. | OUT2- | Отрицательный выход передачи канала 2 по ГОСТ 18977-79 |
| L10 | PF13 | Основ. | OUT2+ | Вход/выход 13 порта F<br>Положительный выход передачи канала 2 по ГОСТ 18977-79 |
| | | Альтер. | A10 | Выход адреса 10 системной шины EXT_BUS |
| | | Переопр. | SSP3_FSS | Вход/выход разрешения SPI 3 |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| M11 | PF14 | Основ. | OUT2- | Вход/выход 14 порта F<br>Отрицательный выход передачи канала 2 по ГОСТ 18977-79 |
| | | Альтер. | A11 | Выход адреса 11 системной шины EXT_BUS |
| | | Переопр. | SSP3_SCK | Вход/выход синхросигнала SPI 3 |
| M12 | PF15 | Основ. | SSP3_RXD | Вход/выход 15 порта F<br>Входные данные SPI 3 |
| | | Альтер. | A12 | Выход адреса 12 системной шины EXT_BUS |
| | | Переопр. | SSP3_TXD | Выходные данные SPI 3 |
| Системное управление | | | | |
| L8 | nRESET | – | – | Сигнал внешнего сброса |
| J7 | OSC_IN | – | – | Вход генератора HSE |
| J8 | OSC_OUT | – | – | Выход генератора HSE |
| M2 | WAKEUP | – | – | Сигнал внешнего выхода из режима Standby |
| F3 | ITCMLAEN | – | – | Сигнал выбора памяти программ:<br>1- внутренняя;<br>0- внешняя |
| USB интерфейс | | | | |
| M3 | DP | – | – | Шина USB D+ |
| M4 | DN | – | – | Шина USB D- |
| PHY Ethernet интерфейс | | | | |
| C1 | TXP | – | – | Дифференциальный выход передатчика на трансформаторы |
| C2 | TXN | – | – | |
| D1 | RXP | – | – | Дифференциальный вход приёмника с трансформатора |
| D2 | RXN | – | – | |
| E2 | EXRES1 | – | – | Вывод для подключения опорного резистора 12,4 кОм 1 % на VSS2A |
| Питание | | | | |
| A6, K3, L11 | Ucc | – | – | Питание |
| K4 | AUcc | – | – | Аналоговое питание АЦП, ЦАП<br>(должно совпадать с $U_{CC}$) |
| D3, E4, F1, F4 | VDD1A – VDD4A | – | – | Аналоговое питание PHY |
| L1 | BUcc | – | – | Батарейное питание |
| E5 – E8, F5 – F8, G5 – G8, H5 – H8 | GND | – | – | Общий |

| Номер вывода | Обозначение вывода | Дополнительные функции вывода | | Функциональное назначение |
|---|---|---|---|---|
| B1, B2, E1, E3, F2 | VSS1A – VSS4A | – | – | Общий PHY |
| JTAG интерфейс | | | | |
| C3 | TMS | – | – | Выбор режима тестирования – обеспечивает переход схемы в/из режима тестирования и переключение между разными режимами тестирования |
| A3 | TDI | – | – | Вход тестовых данных – вход последовательных данных периферийного сканирования. Команды и данные вводятся в микросхему с этого вывода по переднему фронту сигнала TCK |
| A2 | TDO | – | – | Выход тестовых данных – выход последовательных данных. Команды и данные выводятся из микросхемы с этого вывода по заднему фронту сигнала TCK |
| B3 | TCK | – | – | Тестовое тактирование – тактирует работу встроенного автомата управления периферийным сканированием |
| A1 | TRST | – | – | Тестовый сброс – сигнал сброса встроенного автомата управления периферийным сканированием |
| M1 | JTAGEN | – | – | Вход разрешения отладочного TAP интерфейса на выводах JTAG |

Примечания

1 Используемые обозначения функций:

альтер. – альтернативная;

аналог. – аналоговая;

основ. – основная;

переопр. – переопределенная;

1 При использовании внешнего Ethernet PHY, посредством портов PE[15:2], PE[0] и PF[2:0], внутренний Ethernet PHY должен находиться в состоянии сброса (см. описание Блока PHY);

2 При использовании режима работы с ITCMLAEN = 0 аппаратно (программно не модифицируются) порты A и B настраиваются как двунаправленная 32-разрядная шина данных, порт C[2:0] – как сигналы CLKO, nRD, nWR, а порты PF[15:3], PD[15], PE[2:0] – как шина адреса A[16:0]. Можно расширить программно адресную шину до требуемого количества разрядов;

3 Выводы в функции EXTINT1-EXTINT4 являются входами прерываний по уровню. Т.е. пока на вывод EXTINTx подан сигнал высокого уровня, генерируется прерывание. Если на входе сигнал низкого уровня, то прерывание не генерируется

Таблица 5 – Описание выводов микросхем в режиме StandAlone

| Обозначение вывода | Номер вывода | | Функции вывода |
|---|---|---|---|
| | K1986BE1FI | K1986BE1GI | |
| DATA0 | 137 | C4 | Двунаправленная шина данных |
| DATA1 | 136 | B4 | |
| DATA2 | 135 | A4 | |
| DATA3 | 134 | D4 | |
| DATA4 | 133 | C5 | |
| DATA5 | 132 | B5 | |
| DATA6 | 131 | A5 | |
| DATA7 | 130 | D5 | |
| DATA8 | 129 | C6 | |
| DATA9 | 128 | B6 | |
| DATA10 | 124 | D6 | |
| DATA11 | 123 | C7 | |
| DATA12 | 122 | A7 | |
| DATA13 | 121 | B7 | |
| DATA14 | 120 | D7 | |
| DATA15 | 119 | A8 | |
| ADDR0 | 118 | B8 | Шина адреса |
| ADDR1 | 117 | C8 | |
| ADDR2 | 116 | D8 | |
| ADDR3 | 115 | A9 | |
| ADDR4 | 114 | B9 | |
| ADDR5 | 113 | C9 | |
| ADDR6 | 112 | A10 | |
| ADDR7 | 111 | B10 | |
| ADDR8 | 110 | A11 | |
| ADDR9 | 109 | A12 | |
| ADDR10 | 108 | B12 | |
| ADDR11 | 107 | B11 | |
| nBE_ETH0 | 106 | C11 | Сигнал byte enable0 для Ethernet контроллера |
| nBE_ETH1 | 105 | C12 | Сигнал byte enable1 для Ethernet контроллера |
| nCE1 | 104 | C10 | Сигнал chip enable 1 для выбора области регистров Ethernet контроллера или выбора первого контроллера ГОСТ Р 52070-2003 |
| nCE2 | 103 | D10 | Сигнал chip enable 2 для выбора области данных Ethernet контроллера или выбора второго контроллера ГОСТ Р 52070-2003 |
| nWE | 98 | E11 | Сигнал записи данных |
| nOE | 99 | E10 | Сигнал чтения данных |
| RDY_ETH | 100 | D9 | Не используется |
| ADDR12 | 101 | D12 | Старший разряд шины адреса |

| Обозначение вывода | Номер вывода | | Функции вывода |
|---|---|---|---|
| | К1986ВЕ1FI | К1986ВЕ1GI | |
| DATA16 | 102 | D11 | Двунаправленная шина данных только для контроллеров ГОСТ Р 52070-2033 |
| DATA17 | 61 | M8 | |
| DATA18 | 62 | K8 | |
| FXEN / INT1 | 63 | K9 | Сигнал разрешения оптического выхода PHY или выход прерывания от первого контроллера ГОСТ Р 52070-2003.<br>Выход прерываний неактивен (равен нулю) при неактивном состоянии сигналов chip enable (nCE1 = nCE2 = 1) |
| FTX / INT2 | 64 | L9 | Сигнал оптического выхода PHY или выход прерывания от второго контроллера ГОСТ Р 52070-2003.<br>Выход прерываний неактивен (равен нулю) при неактивном состоянии сигналов chip enable (nCE1 = nCE2 = 1) |
| - | 65 | M9 | |
| - | 66 | J9 | |
| - | 67 | K10 | |
| - | 68 | M10 | |
| PRMA+ | 72 | K11 | Дифференциальный вход приёмника основного канала первого контроллера ГОСТ Р 52070-2003 |
| PRMA- | 73 | L12 | |
| PRMB+ | 74 | K12 | Дифференциальный вход приёмника резервного канала первого контроллера ГОСТ Р 52070-2003 |
| PRMB- | 81 | J10 | Дифференциальный вход приёмника резервного канала первого контроллера ГОСТ Р 52070-2003 |
| PRDA+ | 82 | J11 | Дифференциальный выход передатчика основного канала первого контроллера ГОСТ Р 52070-2003 |
| PRDA- | 83 | J12 | |
| PRDB+ | 84 | H10 | Дифференциальный выход передатчика резервного канала первого контроллера ГОСТ Р 52070-2003 |
| PRDB- | 85 | H9 | |
| PRD_PRMA | 86 | H11 | Сигнал разрешения передачи основного канала первого контроллера ГОСТ Р 52070-2003 |
| PRD_PRMB | 87 | H12 | Сигнал разрешения передачи резервного канала первого контроллера ГОСТ Р 52070-2003 |
| - | 46 | L4 | |
| - | 47 | K5 | |
| - | 48 | J5 | |
| - | 49 | L5 | |
| FRX | 50 | M5 | Сигнал оптического входа PHY |
| FXEN | 51 | J6 | Сигнал разрешения оптического выхода PHY |
| FTX | 52 | L6 | Сигнал оптического выхода PHY |
| - | 53 | M6 | |
| FSD | 54 | K6 | Сигнал валидности данных на оптическом входе PHY |

| Обозначение вывода | Номер вывода | | Функции вывода |
|---|---|---|---|
| | К1986ВЕ1FI | К1986ВЕ1GI | |
| MDC | 55 | K7 | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| ETH_INT | 56 | L7 | Выход прерывания от Ethernet контроллера |
| MDIO | 57 | M7 | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY. При этом внутренний PHY должен находиться в состоянии сброса |
| TXD[0] | 18 | G3 | |
| TXD[1] | 19 | G1 | |
| TXD[2] | 20 | G2 | |
| TXD[3] | 37 | L3 | |
| RXD[0] | 38 | L2 | |
| RXD[1] | 21 | G4 | |
| RXD[2] | 22 | H1 | |
| RXD[3] | 23 | H4 | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY. При этом внутренний PHY должен находиться в состоянии сброса |
| TXEN | 24 | H2 | |
| TXER | 25 | H3 | |
| TXCLK | 26 | J1 | |
| RXCLK | 27 | J2 | |
| RXDV | 28 | J4 | |
| RXER | 29 | J3 | Интерфейс MII Ethernet контроллера, используется только при подключении внешнего PHY |
| CRS | 30 | K1 | |
| COL | 31 | K2 | |
| PRMC+ | 88 | G9 | Дифференциальный вход приёмника основного канала второго контроллера ГОСТ Р 52070-2003 |
| PRMC- | 89 | G11 | |
| PRMD+ | 90 | G12 | Дифференциальный вход приёмника резервного канала второго контроллера ГОСТ Р 52070-2003 |
| PRMD- | 91 | G10 | |
| PRDC+ | 92 | F10 | Дифференциальный выход передатчика основного канала второго контроллера ГОСТ Р 52070-2003 |
| PRDC- | 93 | F11 | |
| PRDD+ | 94 | F12 | Дифференциальный выход передатчика резервного канала второго контроллера ГОСТ Р 52070-2003 |
| PRDD- | 95 | F9 | |
| PRD_PRMC | 96 | E9 | Сигнал разрешения передачи основного канала второго контроллера ГОСТ Р 52070-2003 |
| PRD_PRMD | 97 | E12 | Сигнал разрешения передачи резервного канала второго контроллера ГОСТ Р 52070-2003 |
| - | 69 | L10 | |
| - | 70 | M11 | |
| - | 71 | M12 | |
| nRESET | 60 | L8 | Сигнал внешнего сброса |
| OSC_IN | 58 | J7 | Вход генератора HSE |
| OSC_OUT | 59 | J8 | Выход генератора HSE |
| WAKEUP | 39 | M2 | Сигнал внешнего выхода из режима Standby |
| ITCMLAEN | 17 | F3 | Сигнал выбора доступа к контроллеру Ethernet, ГОСТ Р 52070-2003 |

| Обозначение вывода | Номер вывода | | Функции вывода |
|---|---|---|---|
| | К1986ВЕ1FI | К1986ВЕ1GI | |
| TXP* | 3 | C1 | Дифференциальный выход передатчика на трансформаторы |
| TXN* | 2 | C2 | |
| RXP* | 8 | D1 | Дифференциальный вход приёмника с трансформатора |
| RXN* | 7 | D2 | |
| EXRES1 | 12 | E2 | Вход для подключения опорного резистора 12,4 кОм 1 % на VSS2A |
| CM* | 6 | – | Выход опорного напряжения приёмника |
| $U_{CC}$ | 33, 76, 127 | A6, K3, L11 | Питание 3,0…3,6 В |
| $AU_{CC}$ | 45 | K4 | Аналоговое питание АЦП, ЦАП 3,0…3,6 В (должно совпадать с Ucc) |
| VDD1A-VDD4A | 5, 10, 14, 16 | D3, E4, F4, F1 | Аналоговое питание PHY 3,0…3,6 В |
| BUcc | 34 | L1 | Батарейное питание 1,8…3,6 В |
| GND | 3, 75, 125 | E5 – E8, F5 – F8, G5 – G8, H5 – H8 | Общий |
| AGND | 44 | - | Общий |
| VSS1A-VSS4A | 1, 4, 11, 13, 15 | B1, B2, E1, E3, F2 | Общий |

* Подключение выводов приведено в подразделе 28.12 «Блок PHY» для микросхем с функцией Ethernet на рисунке 133, для микросхем без использования функции Ethernet – на рисунке 134.

Примечание – Максимальная частота работы SRAM-интерфейса не более 50 МГц

# 5 Указания по применению и эксплуатации

Указания по режимам и условиям монтажа микросхем согласно ТСКЯ.430106.004Д12.

Выводы микросхем обеспечивают способность их к пайке при температуре 220 °С.

Микросхемы выдерживают воздействие тепла, возникающего при температуре пайки (220 + 5) °С в течение не более (10 ± 1) с.

Материал шариковых выводов микросхем в корпусах типа BGA – Sn63Pb37.

Неиспользуемые пользовательские выводы микросхемы (выводы портов PA – PF) должны быть переведены в аналоговый режим. Неиспользуемые пользовательские входы в цифровом режиме (выводы портов PA – PF, nRESET, WAKEUP) должны быть доопределены до допустимых логических уровней напряжения (высокого или низкого) с помощью внутренних программируемых резисторов доопределения или через внешние резисторы номиналом (1 – 100) кОм.

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

Для обеспечения корректной работы микроконтроллера значения Trim[2:0], HSI_TRIM[5:0] и LSI_TRIM[4:0] должны быть считаны из информационной памяти и записаны в регистры REG_0E и REG_0F.

Запрещается подведение каких-либо электрических сигналов (в том числе шин питание и общий) к неиспользуемым выводам.

Микросхемы после снятия с эксплуатации подлежат утилизации в порядке и методами, устанавливаемыми в контракте на поставку или в соответствии с порядком, установленным у потребителя.

Порядок подачи и снятия напряжении питания и входных сигналов на микросхемы должен быть следующим:

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

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

# 6 Питание микросхемы

Микросхема имеет несколько типов выводов питания:

**$U_{CC}$** – основное питание микросхемы включает питание пользовательских выводов, встроенного регулятора напряжения, USB PHY, генераторов и PLL. Входное напряжение должно быть в пределах от 3,0 до 3,6 В.

**$BU_{CC}$** – питание батарейного домена используется при отсутствии основного питания Ucc для питания батарейного домена и генератора LSE. Переключение с основного питания на батарейное происходит автоматически при снижении уровня $U_{CC}$ ниже 2,0 В. Переключение с батарейного питания на основное происходит автоматически спустя примерно 4 мс после превышения уровнем $U_{CC}$ порога в 2,0 В. Входное напряжение должно быть в пределах от 1,8 до 3,6 В. Если в системе не требуется батарейное питание, то вывод $BU_{CC}$ должен быть объединен с $U_{CC}$.

**$AU_{CC}$** – питание аналоговых блоков АЦП, ЦАП выведено на отдельные выводы для уменьшения помех, создаваемых работой других блоков. Если на данные выводы подается напряжение с того же источника, что и $U_{CC}$, на печатной плате должны быть применены меры по снижению помех. Если для питания АЦП и ЦАП используется отдельный источник, его выходное напряжение не должно отличаться от $U_{CC}$ более чем на ±0,2 В. Для корректной работы АЦП входное напряжение должно быть в пределах от 3,0 до 3,6 В.

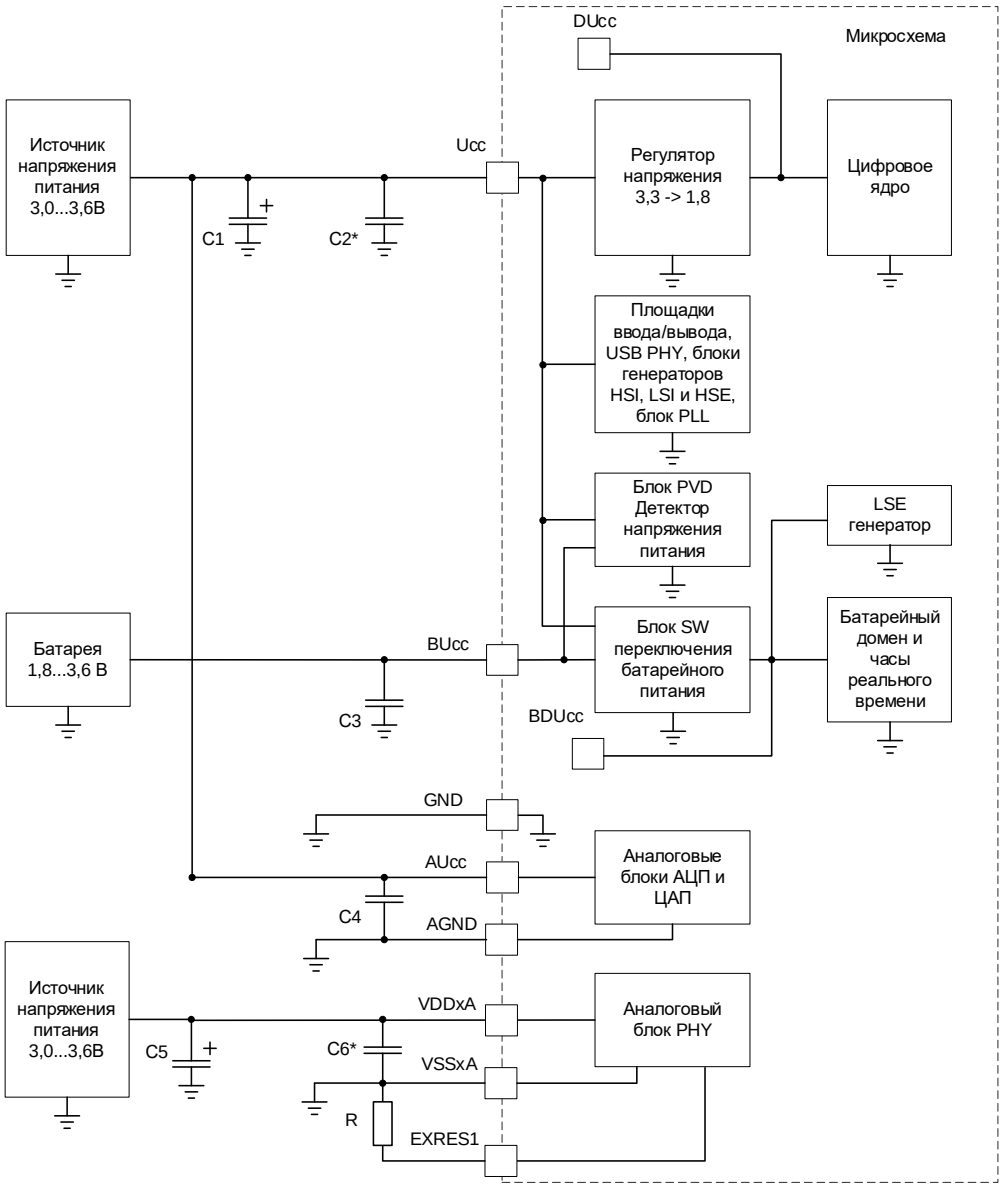
**VDDxA** – питание аналогового блока PHY выведено на отдельные выводы для уменьшения помех, создаваемых работой других блоков. Входное напряжение должно быть в пределах от 3,0 до 3, 6 В. Если на данные выводы подается напряжение с того же источника, что и Ucc, на печатной плате должны быть применены меры по снижению помех.

**GND** – «земля» цифровой части микросхемы. Для микросхем К1986ВЕ1GI на печатной плате должны быть применены меры по снижению помех.

**AGND** – «земля» аналоговой части микросхем К1986ВЕ1FI, данный вывод должен соединяться с GND, но при этом на печатной плате должны быть применены меры по снижению помех. В микросхемах К1986ВЕ1GI контактная площадка AGND кристалла имеет внутреннее соединение с выводами GND, поэтому на печатной плате меры по снижению помех должны быть применены к GND.

**VSSxA** – «земля» аналогового блока PHY. Данные выводы должны соединяться с GND, но при этом на печатной плате должны быть применены меры по снижению помех.

Структурная блок-схема подачи питания приведена на рисунке 4.




* Конденсаторы должны быть установлены у каждого вывода питания

Рисунок 4 – Структурная блок-схема подачи питания

Примечания

1 Конденсатор C1 = C5 = 22 мкФ ± 5 %, C2 = C3 = C4 = C6 = 0,1 мкФ ± 5 %;

2 Резистор R = 12,4 кОм ± 1 %;

3 Если не используется интерфейс Ethernet, вывод VDDxA должен быть объединен с $U_{CC}$;

4 Если не используется батарейное питание, вывод BUcc должен быть объединен с $U_{CC}$;

5 Допускается использование отдельного источника для питания АЦП и ЦАП, но при этом его выходное напряжение не должно отличаться от $U_{CC}$ более чем на ± 0,2 В.

Микросхема имеет встроенный детектор напряжения питания (см. раздел 15 «Детектор напряжения питания»).

## 6.1 Схема сброса при включении и выключении основного питания

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

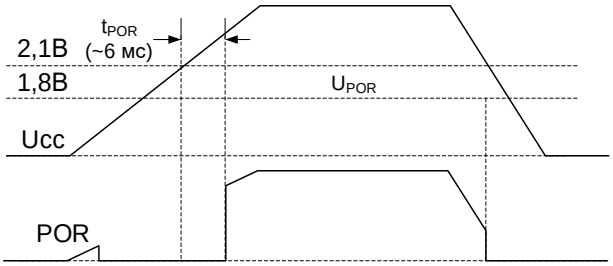



Рисунок 5 – Сигнал сброса при включении и выключении основного напряжения питания

При снижении напряжения питания $U_{CC}$ ниже уровня ~1,8 В сигнал POR вырабатывается без задержки.

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

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

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

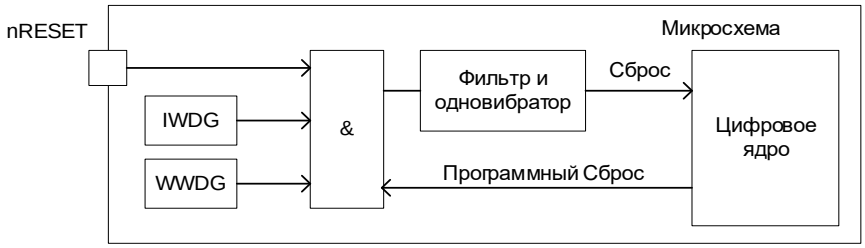



Рисунок 6 – Структурная блок-схема сброса

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

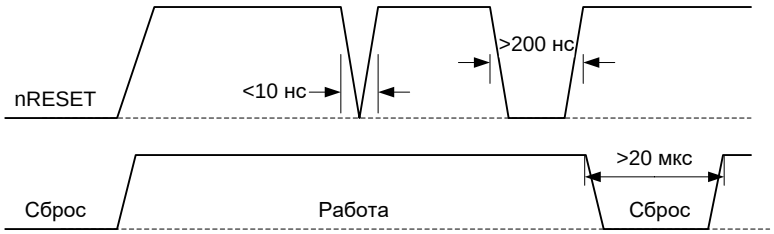



Рисунок 7 – Формирование сигнала сброса

# 7 Организация памяти

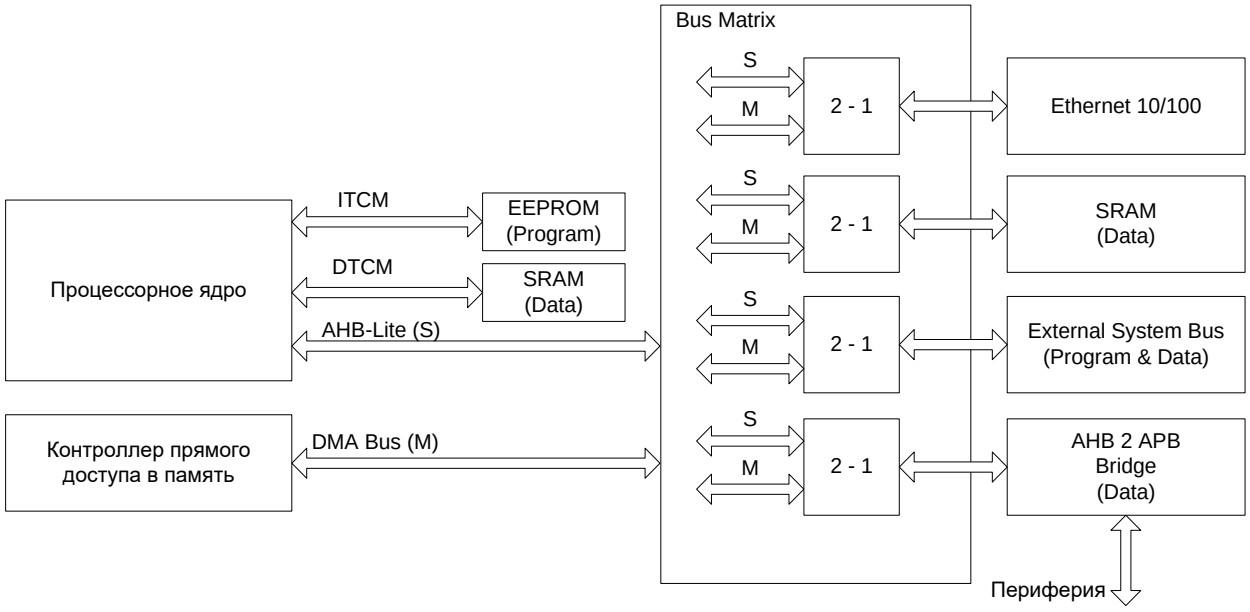



Рисунок 8 – Структурная схема организации памяти

Процессорное ядро имеет три системные шины:

– ITCM Bus – шина выборки инструкций и данных, расположенных в коде программы;

– DTCM Bus – шина выборки данных, расположенных в области ОЗУ;

– AHB-Lite – шина выборки инструкций и данных из внешнего адресного пространства.

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

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

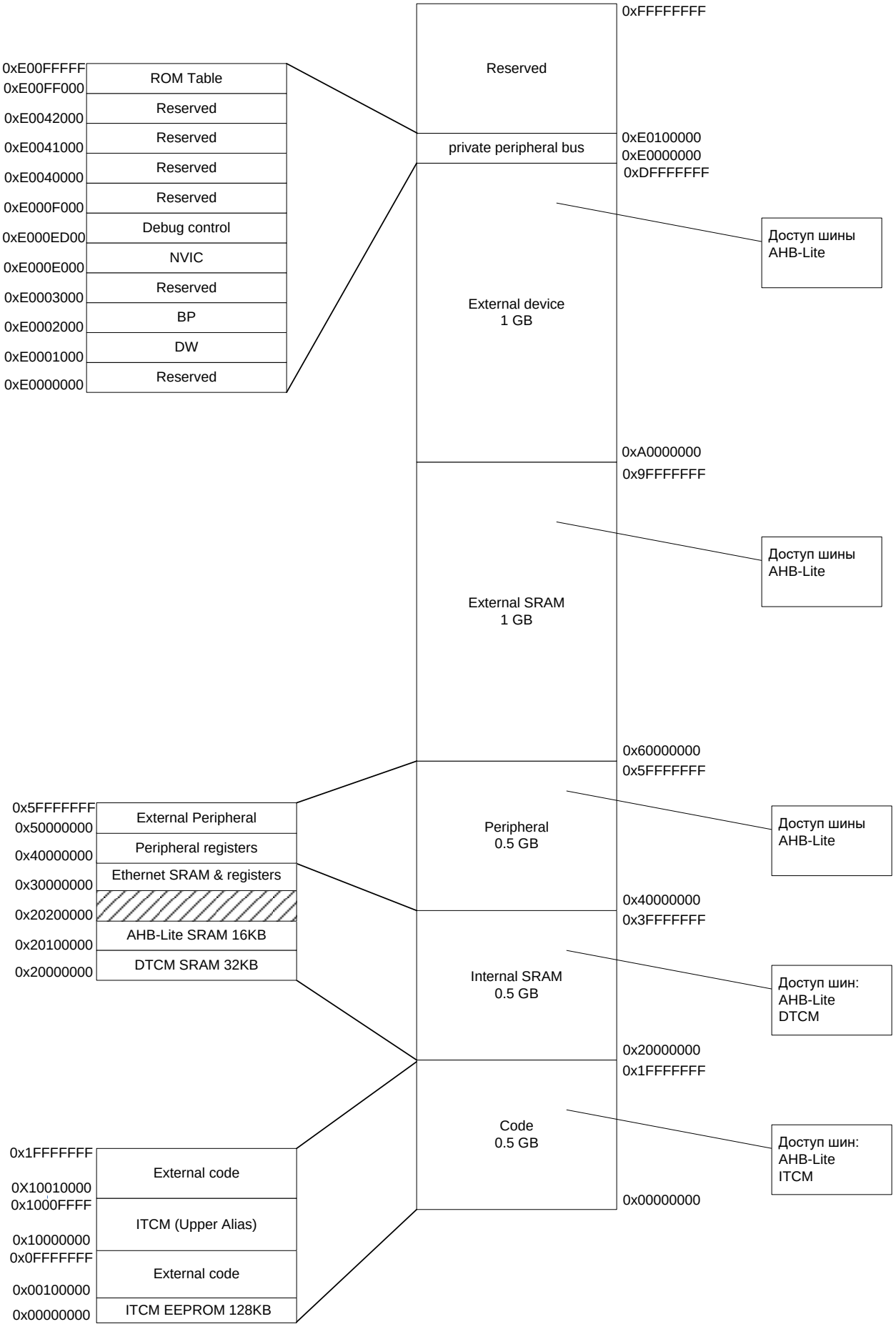



Рисунок 9 – Карта распределения основных областей памяти

## 7.1 Секция Code

Область ITCM EEPROM:

Основная область энергонезависимой памяти программы, доступной для перепрограммирования пользователем. Память предназначена для хранения основной рабочей программы на кристалле.

Область External code:

Область отображения шины AHB-Lite в адресное пространство области программы. Предназначена для хранения кода программ во внешних микросхемах памяти, подсоединенных к внешней системной шине.

## 7.2 Секция Internal SRAM

Область DTCM SRAM:

Основная область ОЗУ, предназначенная для хранения данных программы. В данной области также располагается стек (stack) и «куча» (heap) программы. Адресные диапазоны стека и «кучи» задаются пользователем при написании программы.

Область AHB-Lite SRAM:

Область отображения шины AHB-Lite в адресное пространство области данных. Предназначена для хранения данных в памяти микросхемы, подсоединенной к шине AHB-Lite.

Область Ethernet SRAM & registers:

Область отображения шины AHB-Lite в адресное пространство регистров и памяти контроллера Ethernet. Предназначена для хранения данных в памяти микросхемы, подсоединенной к шине AHB-Lite.

## 7.3 Секция Peripheral

Область Peripheral registers:

Область отображения регистров периферии в общее адресное пространство памяти.

Область External Peripheral:

Область отображения внешней системной шины в адресное пространство области периферии. Предназначена для хранения данных во внешних микросхемах памяти или работы с периферийными устройствами, подсоединенными к внешней системной шине.

## 7.4 Секция External SRAM и External device

Область отображения шины AHB-Lite в адресное пространство области внешней памяти и внешних устройств. Предназначена для хранения данных во внешних микросхемах памяти или работы с внешними устройствами, подсоединенными к внешней системной шине.

## 7.5 Секция Private Peripheral Bus

Предназначена для отображения системных регистров ядра и системной периферии.

## 7.6 Блок BUS MATRIX

Блок BUS MATRIX предназначен для переключения шин AHB-Lite и DMA Bus между различными областями памяти. Переключение производится автоматически на основании адреса запроса каждой конкретной шины. Если адреса запросов не пересекаются, то они могут быть выполнены одновременно. Если адреса запросов пересекаются, то они выполняются в порядке приоритета. Приоритеты обращений заданы аппаратно. Наивысшим приоритетом обладает запрос по шине AHB-Lite, а наименьшим приоритетом обладает запрос DMA Bus. Если два запроса пришли одновременно, то выполняется запрос с большим приоритетом, а запрос с более низким приоритетом задерживается до окончания обработки запроса с более высоким приоритетом. При переключении между шинами возникает дополнительная задержка в один цикл. Если запросы идут непосредственно друг за другом, то дополнительных задержек не возникает.

## 7.7 Память EEPROM

Память области EEPROM реализована в виде перепрограммируемой энергонезависимой памяти. Скорость доступа к памяти EEPROM – порядка 40 нс. При работе микросхемы на частоте до 100 МГц задержка доступа к памяти может составлять до пяти циклов системной частоты. При последовательной выборке за счет упреждающего чтения задержка может быть сокращена до одного цикла системной частоты. Более подробная информация о работе контроллера EEPROM-памяти программ представлена в разделе 7.7 «Контроллер флэш-памяти программ».

## 7.8 Память SRAM

Память области SRAM реализована в виде блока статической памяти. Скорость доступа к памяти SRAM – один цикл системной частоты.

## 7.9 Регионы памяти, типы и атрибуты

Отображение памяти разбивает все адресное пространство на регионы. Каждый регион имеет определенный тип памяти, а некоторые регионы имеют дополнительные атрибуты. Тип памяти и атрибуты определяют поведение системы при доступе к этим регионам.

По отношению к порядку выполнения обращений к памяти различаются следующие типы памяти:

- Normal;
- Device;
- Strongly-ordered («Строго упорядоченная»).

Normal

Процессор может переопределить последовательность обращений для большей эффективности или чтобы выполнить опережающее чтение.

Device

Процессор сохраняет последовательность обращений по отношению к другим обращениям к памяти типа Device или Strongly-ordered.

Strongly-ordered

Процессор сохраняет последовательность обращений по отношению ко всем другим обращениям.

Отличием в последовательности обращения к памяти типа Device и Strongly-ordered является то, что запись в память типа Device система памяти может буферизировать, но никогда не буферизирует запись в память типа Strongly-ordered.

Дополнительные атрибуты памяти:

- Shareable («Допускающая совместное использование»);
- Execute Never или XN («Не выполнять»).

Shareable

Для регионов с атрибутом Shareable система памяти обеспечивает синхронизацию между различными устройствами управления передачей данных по шине при наличии нескольких таких устройств, например, процессор и контроллер DMA.

Память типа Strongly-ordered всегда имеет атрибут Shareable.

Если несколько устройств управления передачей данных по шине могут обращаться к региону, не обладающему атрибутом Shareable, непротиворечивость данных между такими устройствами должна быть гарантирована программным обеспечением.

Execute Never (XN)

Атрибут указывает на запрет обращения к командам (инструкциям). Любая попытка извлечь инструкцию из XN региона приведет к исключению типа Hard fault.

## 7.10 Последовательность обращений к памяти

Для большинства обращений к памяти, инициируемых явно командами обращения к памяти, если только это не затрагивает поведения командной последовательности, система памяти не гарантирует, что порядок, в котором выполняются эти обращения, совпадает с порядком, в котором соответствующие команды следуют в программе.

Обычно, если правильное выполнение программы требует, чтобы два обращения в память произошли в порядке, заданном программой, между командами обращения к памяти в программе должна быть вставлена инструкция барьерной синхронизации (memory barrier instruction), см. подраздел 7.12 «Программное определение последовательности доступа к памяти».

Однако арбитр памяти гарантирует однозначную последовательность доступа в регионы памяти Device и Strongly-ordered. Для двух команд обращения к памяти A1 и A2, если A1 выполняется перед A2 в коде программы, последовательность обращений к памяти будет такой, как приведено в таблице 6.

Таблица 6 – Последовательность обращений инструкций к памяти

| A1 | A2 | | | |
|---|---|---|---|---|
| | Normal | Device | | Strongly-ordered |
| | | non-shareable | shareable | |
| Normal | - | - | - | - |
| Device, non-shareable | - | < | - | < |
| Device, shareable | - | - | < | < |
| Strongly-ordered | - | < | < | < |

Обозначения

- – система памяти не гарантирует последовательность выполнения обращений;

< – обращение к памяти вследствие инструкции A1 всегда будет выполнено перед обращением вследствие инструкции A2

## 7.11 Поведение обращений к памяти

Особенности доступа к разным областям памяти приведены в таблице 7.

Таблица 7 – Особенности доступа к разным областям памяти

| Адресный диапазон | Регион памяти | Тип памяти | XN | Описание |
|---|---|---|---|---|
| 0x00000000-0x000FFFFF | Code, ITCM, Lower Alias | Normal | - | Область памяти шины ITCM для кода программы и данных, если ITCMLAEN установлен в единицу. Если ITCMLAEN равен нулю, то область памяти внешней системной шины |
| 0x00100000-0x0FFFFFFF | Code, external | Normal | - | Область памяти внешней системной шины для кода программы и данных |
| 0x10000000-0x1000FFFF | Code, ITCM, Upper Alias | Normal | - | Область памяти шины ITCM для кода программы и данных, если ITCMUAEN установлен в единицу. Если ITCMUAEN равен нулю, то область памяти внешней системной шины |
| 0x10010000-0x1FFFFFFF | Code, external | Normal | - | Область памяти внешней системной шины для кода программы и данных |
| 0x20000000-0x200FFFFF | SRAM, DTCM | Normal | XN | Область памяти шины DTCM для данных. Доступ за инструкцией в эту область приводит к ошибке |
| 0x20100000-0x3FFFFFFF | SRAM | Normal | | Область памяти внешней системной шины для кода программы и данных |

| Адресный диапазон | Регион памяти | Тип памяти | XN | Описание |
|---|---|---|---|---|
| 0x40000000-0x5FFFFFFF | Peripheral | Device | XN | Область памяти внешней системной шины для данных. Доступ за инструкцией в эту область приводит к ошибке |
| 0x60000000-0x9FFFFFFF | External SRAM | Normal | - | Область памяти внешней системной шины для кода программы и данных. |
| 0xA0000000-0xDFFFFFFF | External Device | Device | XN | Область памяти внешней системной шины для внешних устройств. Доступ за инструкцией в эту область приводит к ошибке |
| 0xE0000000-0xE00FFFFF | Private Peripheral Bus | Strongly-ordered | XN | Этот регион содержит регистры NVIC, system timer и регистры блока управления ядра |
| 0xE0100000-0xFFFFFFFF | Зарезервировано | - | XN | Зарезервировано |

## 7.12 Программное определение последовательности доступа к памяти

Последовательность инструкций в потоке программы не всегда гарантирует последовательность соответствующих обращений в память, это происходит потому что:

– процессор может изменить последовательность обращений для увеличения производительности, но при этом не изменяется общее поведение программы;

– процессор имеет несколько интерфейсов для обращений к памяти;

– память или устройства могут иметь различные скорости доступа;

– для некоторых обращений к памяти имеет место буферизация или упреждающее выполнение.

Если порядок обращений к памяти критичен, программный код должен содержать инструкции барьерной синхронизации:

– инструкция Data Memory Barrier (DMB) позволяет быть уверенным, что выполняемая инструкция транзакции в память будет завершена до следующей транзакции в память (подробнее см. в пункте 11.6.3 «DMB»);

– инструкция Data Synchronization Barrier (DSB) позволяет быть уверенным, что выполняемая инструкция транзакции в память будет завершена до начала выполнения следующей инструкции (подробнее см. в пункте 11.6.4 «DSB»);

– инструкция Instruction Synchronization Barrier (ISB) позволяет быть уверенным, что эффект всех выполняемых транзакций в память полностью соответствует инструкциям (подробнее см. в пункте 11.6.5 «ISB»).

Инструкции барьерной синхронизации используются, например, в следующих случаях:

– самомодифицируемый код. Если программа содержит самомодифицируемый код, используйте инструкцию ISB сразу после модификации кода программы. Это гарантирует, что после этого будет выполняться уже модифицированный код;

– переключение карты памяти. Если система содержит механизм переключения карты памяти, то используйте инструкцию DSB после переключения карты памяти в программе. Это гарантирует, что дальнейшее выполнение инструкций будет идти с новой картой памяти;

– динамическое изменение приоритетов исключений. Когда приоритеты исключений изменяются при обработке исключительной ситуации, используйте инструкцию DSB после изменения. Это гарантирует, что изменение произойдет при завершении инструкции DSB;

– использование семафоров в системе с несколькими устройствами управления передачей данных по шине. Если система содержит несколько таких устройств управления, например, другой процессор, то оба процессора должны использовать инструкции DMB после каждой инструкции работы с семафорами. Это гарантирует, что другой мастер будет видеть обращения к памяти в той последовательности, в которой они выполняются.

При обращениях в память Strongly-ordered, такую как системный блок управления ядра (NVIC, System Timer и так далее) не требуется использовать инструкции DMB.

## 7.13 Базовые адреса процессора

Таблица 8 – Базовые адреса процессора

| Адрес | Размер | Блок | Примечание |
|---|---|---|---|
| Память программ | | | |
| 0x0000_0000 | | EEPROM<br>BOOT ROM | Область флэш-памяти программ с пользовательской программой |
| 0x0010_0000 | | External code | Область доступа к внешней системной шине |
| Память данных | | | |
| 0x2000_0000 | | DTCM SRAM | Область внутреннего ОЗУ |
| 0x2010_0000 | | AHB-Lite SRAM | Область внутреннего ОЗУ |
| 0x3000_0000 | | Ethernet | Область внутреннего ОЗУ и регистров модуля Ethernet |
| Периферия | | | |
| 0x4000_0000 | | CAN1 | Регистры контроллера интерфейса CAN1 |
| 0x4000_8000 | | CAN2 | Регистры контроллера интерфейса CAN2 |
| 0x4001_0000 | | USB | Регистры контроллера интерфейса USB |
| 0x4001_8000 | | EEPROM_CNTRL | Регистры контроллера флэш-памяти программ |
| 0x4002_0000 | | RST_CLK | Регистры контроллера сигналов тактовой частоты |
| 0x4002_8000 | | DMA | Регистры контроллера прямого доступа в память |
| 0x4003_0000 | | UART1 | Регистры контроллера интерфейса UART1 |

| Адрес | Размер | Блок | Примечание |
|---|---|---|---|
| 0x4003_8000 | | UART2 | Регистры контроллера интерфейса UART2 |
| 0x4004_0000 | | SSP1 | Регистры контроллера интерфейса SSP1 |
| 0x4004_8000 | | MIL-STD-1553B1 | Регистры контроллера интерфейса MIL-STD-1553B канал 1 |
| 0x4005_0000 | | MIL-STD-1553B2 | Регистры контроллера интерфейса MIL-STD-1553B канал 2 |
| 0x4005_8000 | | POWER | Регистры детектора напряжения питания |
| 0x4006_0000 | | WWDG | Регистры контроллера сторожевого таймера WWDG |
| 0x4006_8000 | | IWDG | Регистры контроллера сторожевого таймера IWDG |
| 0x4007_0000 | | TIMER1 | Регистры управления Таймер 1 |
| 0x4007_8000 | | TIMER2 | Регистры управления Таймер 2 |
| 0x4008_0000 | | TIMER3 | Регистры управления Таймер 3 |
| 0x4008_8000 | | ADC | Регистры управления АЦП |
| 0x4009_0000 | | DAC | Регистры управления ЦАП |
| 0x4009_8000 | | TIMER4 | Регистры управления Таймер 4 |
| 0x400A_0000 | | SSP2 | Регистры контроллера интерфейса SSP2 |
| 0x400A_8000 | | PORTA | Регистры управления порта A |
| 0x400B_0000 | | PORTB | Регистры управления порта B |
| 0x400B_8000 | | PORTC | Регистры управления порта C |
| 0x400C_0000 | | PORTD | Регистры управления порта D |
| 0x400C_8000 | | PORTE | Регистры управления порта E |
| 0x400D_0000 | | ARINC429R | Регистры контроллера интерфейса приемников ARINC429 |
| 0x400D_8000 | | BKP | Регистры доступа и управления батарейным доменом |
| 0x400E_0000 | | ARINC429T | Регистры контроллера интерфейса передатчиков ARINC429 |
| 0x400E_8000 | | PORTF | Регистры управления порта F |
| 0x400F_0000 | | EXT_BUS_CNTRL | Область доступа к внешней системной шине |
| 0x400F_8000 | | SSP3 | Регистры контроллера интерфейса SSP3 |
| | | | |
| 0x5000_0000 | | External peripheral | Область доступа к внешней системной шине |
| Внешняя системная шина | | | |
| 0x6000_0000 | | External SRAM | Область доступа к внешней системной шине |
| 0xA000_0000 | | External device | Область доступа к внешней системной шине |
| Private Peripheral Bus | | | |
| 0xE000_0000 | | | Системные регистры процессора |

# 8 Загрузочное ПЗУ и режимы работы микросхемы

После включения питания и снятия внутренних (POR) и внешних (RESET) сигналов сброса, микросхема начинает выполнять программу из загрузочной области ПЗУ BOOT ROM (4 Кбайт). В загрузочной программе микросхема определяет, в каком из режимов она будет функционировать, и переходит в этот режим. Режим функционирования определяется внешними выводами MODE[2:0] (PA[2:0]). Также устанавливается бит FPOR в регистре BKP_REG_0E, который может быть сброшен только при отключении основного питания $U_{CC}$. Далее микросхема сбрасывается. После перезапуска микросхемы уровни на выводах MODE[2:0] не влияют на режим функционирования микросхемы, если установлен бит FPOR. В пользовательской программе выводы PA[2:0] могут использоваться пользователем.

Таблица 9 – Режимы первоначального запуска микросхемы

| MODE[2:0] | Режим | Стартовый адрес/ таблица векторов прерываний | Описание |
|---|---|---|---|
| 000 | Микроконтроллер с режимом отладки | 0x0000_0000 | Процессор начинает выполнять программу из внутренней флэш-памяти программ. При этом разрешается работа отладочного интерфейса JTAG/SW.<br>Сигнал выбора памяти программ:<br>ITCMLAEN = 1 – внутренняя память;<br>ITCMLAEN = 0 – внешняя память.<br>При использовании режима работы с ITCMLAEN = 0 аппаратно (программно не модифицируются) порты A и B настраиваются как двунаправленная 32-разрядная шина данных, порт C[2:0] – как сигналы CLKO, nRD, nWR, а порты PF[15:3], PD[15], PE[2:0] – как шина адреса A[16:0]. Можно расширить программно адресную шину до требуемого количества разрядов |
| 001 | Режим Stand alone1 | 0x0000_0000 | Процессор конфигурирует прямой доступ к контроллеру Ethernet с помощью внешней системной шины и переходит в режим сброса. Частота задается внешним генератором HSE/2, умноженная на четыре с помощью PLL.<br>Адрес – {PC[3], PB[11:0]}<br>Данные – PA[15:0]<br>Byte enable – PB[13:12]<br>Chip enable – PB[15:14]<br>Write enable – PC[0]<br>Output enable – PC[1]<br>Вход ITCMLAEN = 1 |

| MODE[2:0] | Режим | Стартовый адрес/ таблица векторов прерываний | Описание |
|---|---|---|---|
| 010 | Режим Stand alone2 | 0x0000_0000 | Процессор конфигурирует прямой доступ к контроллеру интерфейса ГОСТ Р 52070-2003 с помощью внешней системной шины и переходит в режим сброса.<br>Частота задается внешним осциллятором HSE, умноженная на 10 с помощью PLL.<br>Адрес – {PC[3],PB[11:0]}<br>Данные – {PC[6:4],PA[15:0]}<br>Chip enable – PB[15:14]<br>Write enable – PC[0]<br>Output enable – PC[1]<br>Вход ITCMLAEN = 0 |
| 011 | Режим Stand alone3 | 0x0000_0000 | Процессор конфигурирует прямой доступ к контроллерам Ethernet и интерфейса ГОСТ Р 52070-2003 с помощью внешней системной шины и переходит в режим сброса.<br>Частота задается внешним генератором HSE/2, умноженная на четыре с помощью PLL.<br>Вход ITCMLAEN = 1 – доступ к контроллеру Ethernet.<br>Вход ITCMLAEN = 0 – доступ к контроллеру интерфейса ГОСТ Р 52070-2003 |
| 100-110 | UART-загрузчик | 0x0000_0000 | Микросхема через интерфейс UART1 на выводах PC[4:3] получает код программы в ОЗУ для исполнения. При этом разрешается работа отладочного интерфейса JTAG/SW |
| 111 | Зарезервировано | - | - |

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

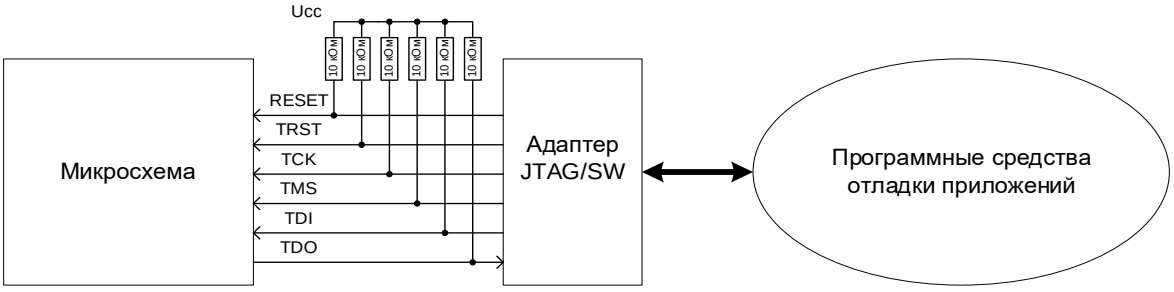



Рисунок 10 – Схема работы в режиме отладки

В режиме отладки можно:

– стирать, записывать, считывать внутреннюю флэш-память программ;

- считывать и записывать содержимое ОЗУ, периферии;
- выполнять программу в пошаговом режиме;
- запускать программу в нормальном режиме;
- останавливать выполнение программы на точках останова;
- просматривать переменные выполняемой программы;
- проводить трассировку хода выполнения программного обеспечения.

В зависимости от режима работы выводы интерфейса JTAG/SW переопределяются на различные выводы микросхемы, как показано в таблице 10.

Таблица 10 – Переопределение выводов интерфейса JTAG/SW

| Вывод JTAG/SW | Вывод микросхемы | Описание |
|---|---|---|
| JTAGEN | JTAGEN | Необходимо доопределить до логического «0» для работы с отладочным интерфейсом JTAG/SW |
| JTAG | | |
| TRST | TRST | В качестве выводов интерфейса используются выделенные выводы микросхемы |
| TCK | TCK | |
| TMS | TMS | |
| TDI | TDI | |
| TDO | TDO | |
| SW | | |
| SWCLKTCK | TCK | Вход тактовой частоты |
| SWDITMS | TMS | Двунаправленные данные |

## 8.1 UART-загрузчик

В режиме UART-загрузчика связь осуществляется через периферийный модуль UART1 на выводах PC3 (в функции UART_TXD1) и PC4 (в функции UART_RXD1).

Данный режим предоставляет достаточный набор операций, необходимых для записи в ОЗУ какой-либо программы (в частности программатора флэш-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 (BOOT ROM, 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 значений делителя частоты UART1 для синхронизации с внешним устройством.

При загрузке данных в ОЗУ необходимо учитывать, что диапазон адресов 0x2000_0000-0x2000_05FF используется загрузочной программой для работы со стеком и глобальными переменными.

### 8.1.1 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.

### 8.1.2 Протокол обмена по UART

После синхронизации с внешним устройством, подающим команды (Master), загрузчик переходит в диспетчер команд.

Таким образом, внешнему устройству доступны команды, приведенные в таблице 11.

Таблица 11 – 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 устройству Master |
| 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 |

#### 8.1.2.1 *Синхронизация с внешним устройством*

Начальные условия.

На этапе синхронизации с внешним устройством (Master) вывод Rx используется как вход. Master постоянно посылает в канал синхросимвол 0x00. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ошибки обмена. Как только загрузчик настроил скорость, он переходит в диспетчер команд и выдает приглашение (три байта 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>'),) устройству Master.

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

#### 8.1.2.2 *Команда CMD_SYNC*

Пустая команда.

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

Таблица 12 – 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 |

#### 8.1.2.3 *Команда CMD_CR*

Выдача приглашения Master-у.

Таблица 13 – 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.<br>Иначе выдает 3 байта:<br>– код команды CMD_CR;<br>– код 0x0A;<br>– код 0x3E (ASCII символ '>') |

#### 8.1.2.4 *Команда CMD_BAUD*

Установка скорости обмена.

Таблица 14 – Команда CMD_BAUD

| **Код команды** | **CMD_BAUD = 0x42** |
|---|---|
| ASCII символ, соответствующий коду команды | 'B' |
| Количество параметров команды | 1 |
| Параметр | Новое значение скорости обмена [бод] |
| **Формат команды:** | |
| Master: выдает код команды CMD_BAUD | Slave: если команда принята с ошибками, то выдает сообщение об ошибке 0x45 (‘E’) и завершает обработку текущей команды |
| Master: выдает параметр | Slave: если параметр принят с ошибками, то выдает сообщение об ошибке 0x45 (‘E’) и завершает обработку текущей команды.<br>Иначе:<br>– выдает код команды CMD_BAUD; |

| | |
|---|---|
| | – устанавливает новое значение скорости обмена во время отправки кода команды CMD_BAUD |

#### 8.1.2.5 *Команда CMD_LOAD*

Загрузка массива байт в память микросхемы.

Таблица 15 – 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, 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 и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончанию принятия массива выдает код ответа REPLY_OK = 0x4B ('K') |

#### 8.1.2.6 *Команда CMD_VFY*

Выдача массива байтов из памяти микросхемы.

С помощью данной команды (0x59 0xFF 0x0F 0x00 0x00 0x01 0x00 0x00 0x00) можно прочитать код даты изготовления микросхемы, хранящийся в четырех старших разрядах байта по адресу 0xFFF.

Коды дат изготовления микросхемы:

– 0x6D – микросхемы с 2332.

Таблица 16 – 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:<br>– выдает код команды CMD_VFY;<br>– выдает массив байт младшим байтом вперед;<br>– по окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K') |

#### 8.1.2.7 *Команда CMD_RUN*

Запуск программы на выполнение.

Таблица 17 – Команда CMD_RUN

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

### 8.1.3 Прием параметров команды

Параметры команд – это четырехбайтные числа.

Параметры передаются младшим байтом вперед.

В качестве значения параметра запрещено использовать число 0xFFFFFFFF.

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

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

### 8.1.4 Сообщения об ошибках

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

После выдачи сообщения об ошибке загрузчик переходит в режим ожидания следующей команды, поэтому Master после получения такого сообщения должен прекратить передачу байт, относящихся к текущей команде.

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

**Ошибка ERR_CHN**

Аппаратная ошибка UART.

Код ошибки 0x69 ('i').

Выдается диспетчером команд, если UART установил в «1» один из аппаратных флагов ошибки при приеме очередного байта.

Примечание – Не реализовано в загрузчике.

**Ошибка ERR_CMD**

Принята неизвестная команда.

Код ошибки 0x63 ('c').

Выдается диспетчером команд, если принят неизвестный код команды.

Примечание – Не реализовано в загрузчике.

**Ошибка ERR_BAUD**

Принята неизвестная команда.

Код ошибки 0x62 ('b').

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

Примечание – Не реализовано в загрузчике.

## 8.2 Тестовый режим работы

Микросхема переводится в тестовый режим при установке высокого уровня на выводе JTAGEN. При переходе в тестовый режим работа микросхемы в функциональном режиме останавливается.

В тестовом режиме к выводам JTAG подключается тестовый TAP-контроллер. Регистр инструкций (IR) тестового TAP-контроллера имеет длину 8 бит. Загрузка инструкций в регистр IR переводит микросхему в тестовые режимы, приведенные в таблице 18.

Таблица 18 – Инструкции тестового TAP-контроллера

| IR[7:0] | Название инструкции | Описание |
|---|---|---|
| 0x00 | EXTEST | Переводит микросхему в режим внешнего граничного тестирования. В качестве регистра данных выбирается регистр граничного сканирования |
| 0x01 | SAMPLE | Позволяет считать регистр граничного сканирования, который выбирается в качестве регистра данных |
| 0x02 | IDCODE | В качестве регистра данных выбирается регистр ID, который имеет значение 0x6DC00FDF |
| 0x06 | PRELOAD | Позволяет загрузить регистр граничного сканирования, который выбирается в качестве регистра данных |

| | | |
|---|---|---|
| 0xFF | BYPASS | В качестве регистра данных выбирается 1-битный регистр. Используется для обеспечения последовательного пути минимальной длины между выводами TDI и TDO |

# 9 Контроллер флэш-памяти программ

Микросхема содержит встроенную флэш-память программ с объемом 128 Кбайт основной памяти программ и 4 Кбайт информационной памяти. В обычном режиме (бит CON = 0, регистр EEPROM_CMD) доступна основная память программ через системную шину ITCM для выборки инструкций и данных кода программы. В режиме программирования (бит CON = 1, регистр EEPROM_CMD) основная и информационная память доступны как периферийное устройство и могут быть использованы для нужд разработчика приложения. В режиме программирования программный код должен выполняться из области шины AHB-Lite или ОЗУ. Выполнение программного кода из флэш-памяти программ в режиме программирования невозможно.

## 9.1 Работа флэш-памяти программ в обычном режиме

Скорость доступа во флэш-память ограничена и составляет порядка 40 нс, в результате выдача новых значений из флэш-памяти может происходить с частотой не более 25 МГц. Для того чтобы процессорное ядро могло получать новые инструкции на больших частотах, в микросхеме реализуется флэш-память с физической организацией 8К на 128 разрядов. Таким образом, за 40 нс из флэш-памяти извлекается 16 байтов, в которых может быть закодировано от четырех до восьми инструкций процессора. И пока ядро выполняет эти инструкции, из памяти извлекается следующая порция данных. Таким образом, тактовая частота может превышать частоты извлечения данных из памяти в несколько раз при линейном выполнении программы. При возникновении переходов, когда из памяти программ не выбраны нужные инструкции, в выполняемой программе возникает пауза в несколько тактов процессора для того, чтобы данные успели считаться из флэш-памяти. Число тактов паузы зависит от тактовой частоты процессора, так при работе с частотой ниже 25 МГц пауза не требуется, так как флэш-память успевает выдать новые данные за один такт, при частоте от 25 до 50 МГц требуется один такт паузы и так далее. Число тактов паузы задается в регистре EEPROM_CMD битами Delay[2:0]. В таблице 19 приведены характеристики необходимой паузы для работы флэш-памяти программ.

Таблица 19 – Дополнительная пауза для работы флэш-памяти

| Delay[2:0] | Тактов паузы | Тактовая частота, МГц | Примечание |
|---|---|---|---|
| 0x00 | 0 | до 25 | |
| 0x01 | 1 | до 50 | |
| 0x02 | 2 | до 75 | |
| 0x03 | 3 | до 100 | |
| 0x04 | 4 | до 125 | Установлено по умолчанию после сброса |
| 0x05 | 5 | до 150 | Работа с частотой более 144 МГц не гарантируется |
| 0x06 | 6 | до 175 | |
| 0x07 | 7 | до 200 | |

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

## 9.2 Работа флэш-памяти программ в режиме программирования

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

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

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

Структура флеш-памяти приведена на рисунке 11.

| | | | | |
|---|---|---|---|---|
| Страница 31<br><br>256 x 128<br>4K x 8 | 0x0001_FFFC<br>…<br>0x0001_F00C | 0x0001_FFF8<br>…<br>0x0001_F008 | 0x0001_FFF4<br>…<br>0x0001_F004 | 0x0001_FFF0<br>…<br>0x0001_F000 |
| | … | … | … | … |
| Страница 1<br><br>256 x 128<br>4K x 8 | 0x0000_1FFC<br>…<br>0x0000_100C | 0x0000_1FF8<br>…<br>0x0000_1008 | 0x0000_1FF4<br>…<br>0x0000_1004 | 0x0000_1FF0<br>…<br>0x0000_1000 |
| Страница 0<br><br>256 x 128<br>4K x 8 | 0x0000_0FFC<br>…<br>0x0000_001C<br>0x0000_000C | 0x0000_0FF8<br>…<br>0x0000_0018<br>0x0000_0008 | 0x0000_0FF4<br>…<br>0x0000_0014<br>0x0000_0004 | 0x0000_0FF0<br>…<br>0x0000_0010<br>0x0000_0000 |
| | Sector_D<br>256 x 32<br>1K x 8 | Sector_C<br>256 x 32<br>1K x 8 | Sector_B<br>256 x 32<br>1K x 8 | Sector_A<br>256 x 32<br>1K x 8 |

Основная память (IFREN=0)

| | | | | |
|---|---|---|---|---|
| Страница 0<br><br>256 x 128<br>4K x 8 | 0x0000_0FFC<br>…<br>0x0000_001C<br>0x0000_000C | 0x0000_0FF8<br>…<br>0x0000_0018<br>0x0000_0008 | 0x0000_0FF4<br>…<br>0x0000_0014<br>0x0000_0004 | 0x0000_0FF0<br>…<br>0x0000_0010<br>0x0000_0000 |
| | Sector_D<br>256 x 32<br>1K x 8 | Sector_C<br>256 x 32<br>1K x 8 | Sector_B<br>256 x 32<br>1K x 8 | Sector_A<br>256 x 32<br>1K x 8 |

Информационная память (IFREN=1)

Рисунок 11 – Структура флеш-памяти

### 9.2.1 Стирание всей памяти

Стирание всей памяти выполняется в четыре этапа:

– первый этап – стирание Sector_A для всей памяти;
– второй этап – стирание Sector_B для всей памяти;
– третий этап – стирание Sector_C для всей памяти;
– четвертый этап – стирание Sector_D для всей памяти.

Стирание одного сектора памяти возможно только в режиме программирования. Для стирания одного сектора памяти надо установить необходимое значение в бит IFREN (1 – для основной и информационной памяти и 0 – для основной памяти) и номер сектора EEPROM_ADR[3:2] (00 – Sector_A, 01 – Sector_B, 10 – Sector_C и 11 – Sector_D), затем установить биты XE, MAS1 и ERASE в «1», и спустя время $t_{nvs}$ (не менее 5 мкс) установить бит NVSTR в «1». Полное стирание памяти длится время $t_{me}$ (от 20 мс до 40 мс). Спустя это время необходимо очистить бит ERASE, и спустя время $t_{nvh1}$ (не менее 100 мкс) очистить биты XE, MAS1 и NVSTR. Последующие операции с память можно выполнять спустя время $t_{rcv}$ (не менее 1 мкс). Временная диаграмма стирания памяти представлена далее (см. рисунок 12). При стирании информационной области, автоматически стирается и основная. Для стирания всей памяти необходимо осуществить стирание всех секторов памяти.

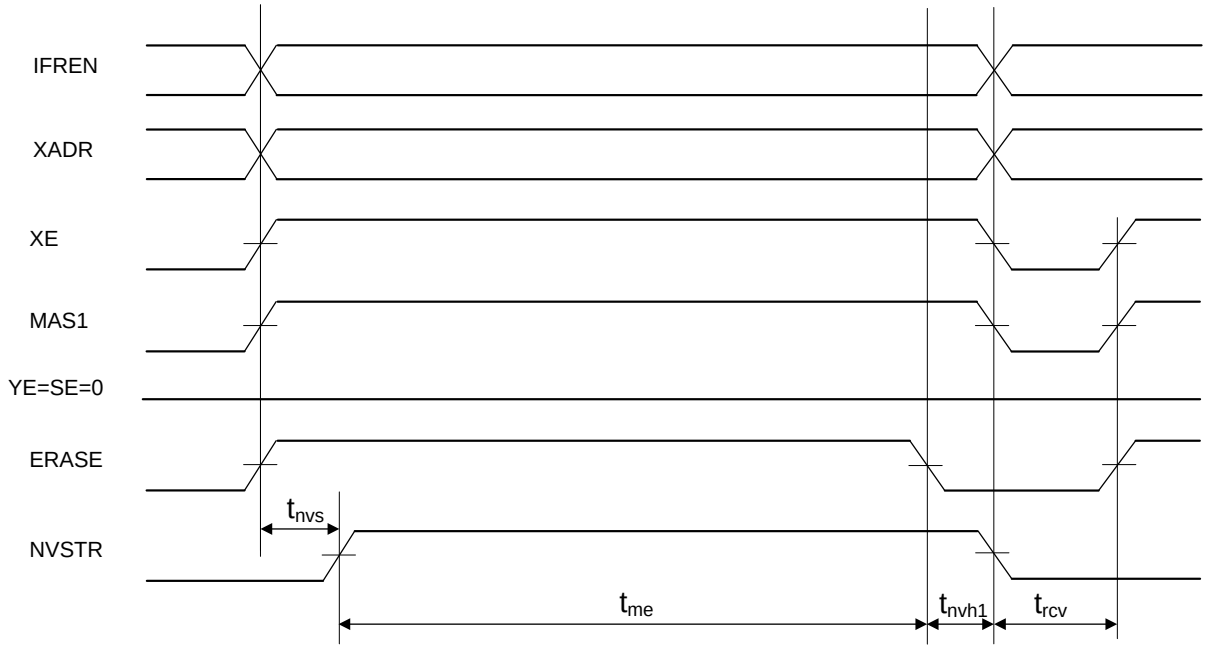



Рисунок 12 – Временная диаграмма стирания памяти

### 9.2.2 Стирание банка памяти одной страницы размером 4 Кбайт

Стирание одной страницы выполняется в четыре этапа:

– первый этап – стирание Sector_A для одной страницы;
– второй этап – стирание Sector_B для одной страницы;
– третий этап – стирание Sector_C для одной страницы;
– четвертый этап – стирание Sector_D для одной страницы.

Стирание одного сектора страницы памяти возможно только в режиме программирования. Для стирания одного сектора страницы памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес стираемой страницы в регистре EEPROM_ADR[16:12] и номер сектора EEPROM_ADR[3:2] (00 – Sector_A, 01 – Sector_B, 10 – Sector_C и 11 – Sector_D). Далее установить биты XE и ERASE в «1», и спустя время $t_{nvs}$ (не менее 5 мкс) установить бит NVSTR в «1». Стирание страницы памяти длится время $t_{erase}$ (от 20 мс до 40 мс). Спустя это время необходимо очистить бит ERASE, и спустя время $t_{nvh}$ (не менее 5 мкс) очистить биты XE и NVSTR. Последующие операции с памятью можно выполнять спустя время $t_{rcv}$ (не менее 1 мкс). Временная диаграмма стирания страницы памяти представлена далее (см. рисунок 13). Для стирания всей страницы необходимо осуществить стирание всех секторов страницы.

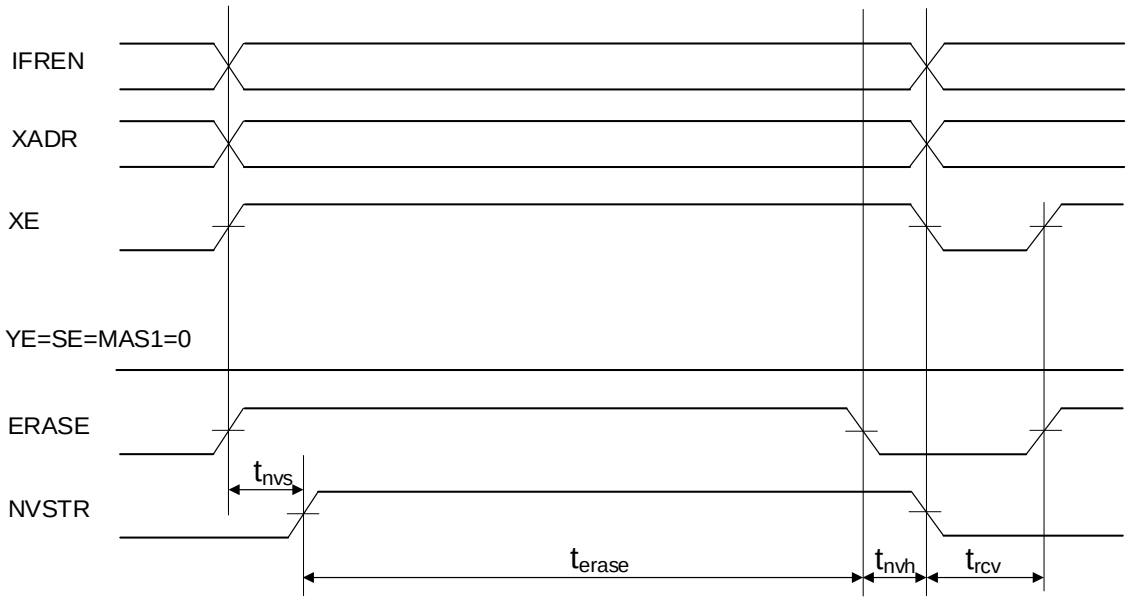



Рисунок 13 – Временная диаграмма стирания банка памяти

### 9.2.3 Запись 32-битного слова в память

Запись в память возможна только в режиме программирования. Для записи в память надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес, по которому производится запись в регистре EEPROM_ADR, в регистр EEPROM_DI поместить записываемое в память слово, установить биты XE и PROG в «1», и спустя время $t_{nvs}$ (не менее 5 мкс) установить бит NVSTR в «1». Спустя время $t_{pgs}$ (не менее 10 мкс) установить бит YE в «1». Запись в память длиться время $t_{prog}$ (от 20 мкс до 40 мкс). Спустя это время необходимо очистить бит YE, и спустя время $t_{adh}$ (не менее 20 нс) установить новый адрес и значение для записи в другую ячейку памяти. И спустя $t_{ads}$ (не менее 20 нс) установить бит YE в «1» и записать следующее слово.

Примечания

1 Перед программированием ячейки необходимо выполнить ее стирание.

2 В одном цикле допускается запись в пределах одного сектора одной страницы.

Если запись больше не требуется, то спустя время $t_{pgh}$ (не менее 20 нс) после очистки бита YE необходимо очистить бит PROG и спустя время $t_{nvh}$ (не менее 5 мкс) очистить биты XE и NVSTR. Последующие операции с памятью можно выполнять спустя время $t_{rcv}$ (не менее 1 мкс). Временная диаграмма записи памяти представлена на рисунке 14.

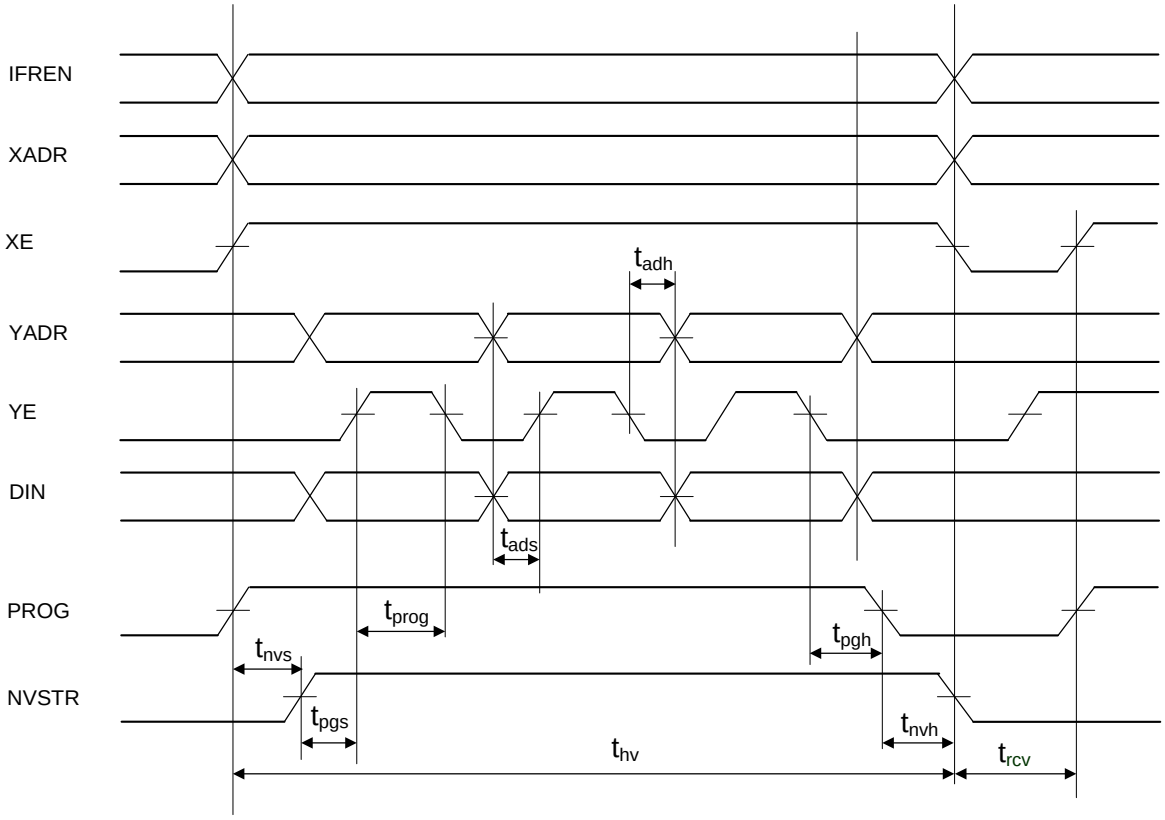



Рисунок 14 – Временная диаграмма записи памяти

### 9.2.4 Чтение 32-битного слова из памяти

В обычном режиме работы для чтения доступна только основная память. Для этого необходимо просто считать требуемый адрес памяти. В режиме программирования для чтения доступна и основная, и информационная память. Для чтения из памяти надо установить необходимое значение в бит IFREN (1 – для информационной памяти и 0 – для основной памяти), затем установить адрес, из которого необходимо считать данные в регистре EEPROM_ADR, установить биты XE, YE и SE в «1», и спустя время $t_{xa}$, $t_{ya}$ (не менее 30 нс) из регистра EEPROM_DO можно считать данные. Если необходимо считать следующее слово, то в регистр EEPROM_ADR необходимо записать новый адрес и спустя время $t_{xa}$, $t_{ya}$ (не менее 30 нс) из регистра EEPROM_DO можно считать следующие данные. Если чтение больше не требуется, то можно очистить все биты управления. Временная диаграмма чтения памяти представлена на рисунке 15.

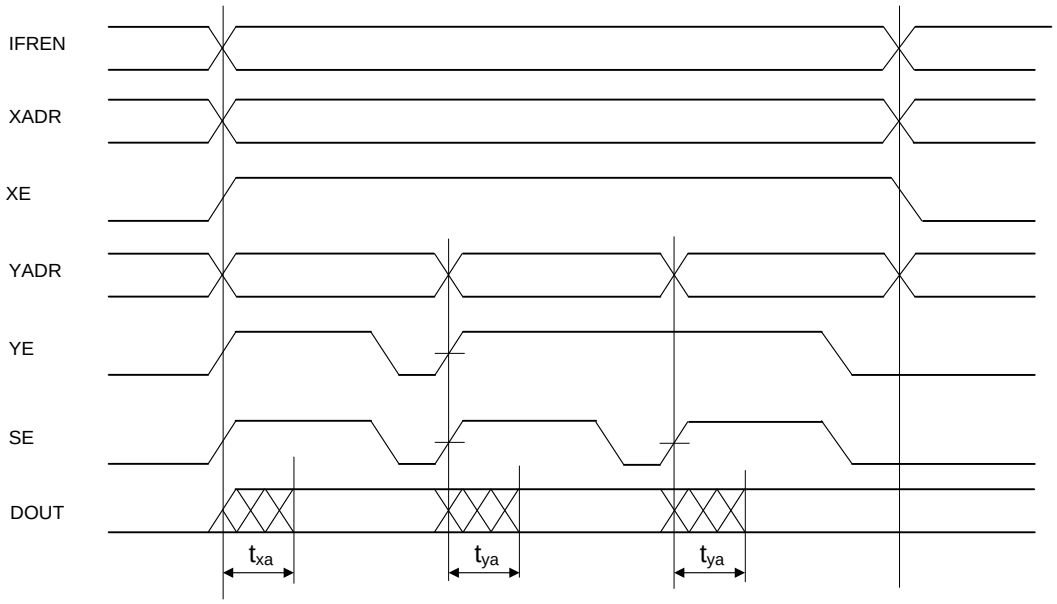



Рисунок 15 – Временная диаграмма чтения памяти

Примечание – Выдача данных начинается каждый раз при передаче фронта любого сигнала – IFREN, XADR, YADR, XE, YE, SE, спустя время $t_{xa}$ или $t_{ya}$.

Состояние линии DOUT будет не определено («X»), если сигналы XE, YE, SE не установлены в «1».

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

## 9.3 Производственная информация

В области информационной памяти (Sector_A) по адресам, приведенным в таблице 20, записана производственная информация, которая содержит идентификационные номера и коэффициенты подстройки блоков.

Таблица 20 – Описание области производственной информации

| Адрес | Название | Описание |
|---|---|---|
| 0x00000FF0<br>0x00000FE0<br>0x00000FD0<br>0x00000FC0<br>0x00000FB0 | UNIQ_ID[159:0] | Уникальный идентификационный номер |
| 0x00000FA0 | LDO_TRIM[2:0] | Биты 2-0:<br>коэффициент настройки опорного напряжения LDO, 1,8 В |
| 0x00000F90 | HSI_TRIM[5:0] | Биты 5-0:<br>коэффициент подстройки частоты генератора HSI, 8 МГц |
| 0x00000F80 | LSI_TRIM[4:0] | Биты 4-0:<br>коэффициент подстройки частоты генератора LSI, 40 кГц |

Для уникальной идентификации микросхемы необходимо использовать все поля UNIQ_ID.

Для обеспечения корректной работы микросхемы значения коэффициентов подстройки LDO_TRIM[2:0], HSI_TRIM[5:0] и LSI_TRIM[4:0] должны быть считаны из информационной памяти и записаны в регистры REG_0E и REG_0F.

## 9.4 Описание регистров управления контроллера флэш-памяти программ

В таблице 21 приведен перечень регистров управления контроллера флэш-памяти программ.

Таблица 21 – Регистры управления контроллера флэш-памяти программ

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

### 9.4.1 EEPROM_CMD

Таблица 22 – Регистр команды EEPROM_CMD

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

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

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

Таблица 23 – Описание бит регистра EEPROM_CMD

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...14 | - | Зарезервировано |
| 13 | NVSTR | Операции записи или стирания:<br>0 – при чтении;<br>1 – при записи или стирании |
| 12 | PROG | Записать данные по ADR[16:2] из регистра EEPROM_DI:<br>0 – нет записи;<br>1 – есть запись |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 11 | MAS1 | Стереть весь блок, при ERASE =1:<br>0 – стирается страница с адресом ADR[16:12] в секторе с адресом ADR[3:2];<br>1 – стирается весь сектор с адресом ADR[3:2] |
| 10 | ERASE | Стереть строку с адресом ADR[16:9], ADR[8:0] значения не имеет:<br>0 – нет стирания;<br>1 – стирание |
| 9 | IFREN | Работа с блоком информации:<br>0 – основная память;<br>1 – информационный блок |
| 8 | SE | Усилитель считывания:<br>0 – не включен;<br>1 – включен |
| 7 | YE | Выдача адреса ADR[8:2]:<br>0 – не разрешено;<br>1 – разрешено |
| 6 | XE | Выдача адреса ADR[16:9]:<br>0 – не разрешено;<br>1 – разрешено |
| 5…3 | Delay[2:0] | Задержка памяти программ при чтении в циклах (в рабочем режиме):<br>000 – ноль циклов;<br>001 – один цикл;<br>…<br>111 – семь циклов |
| 2, 1 | - | Зарезервировано |
| 0 | CON | Переключение контроллера памяти EEPROM на регистровое управление не может производиться при выполнении программы из области EEPROM.<br>0 – управление EEPROM от ядра, рабочий режим;<br>1 – управление от регистров, режим программирования |

### 9.4.2 EEPROM_ADR

Таблица 24 – Регистр адреса EEPROM_ADR

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

Таблица 25 – Описание бит регистра адреса EEPROM_ADR

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

### 9.4.3 EEPROM_DI

Таблица 26 – Регистр записываемых данных EEPROM_DI

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

Таблица 27 – Описание бит регистра записываемых данных EEPROM_DI

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

### 9.4.4 EEPROM_DO

Таблица 28 – Регистр считываемых данных EEPROM_DO

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

Таблица 29 – Описание бит регистра считываемых данных EEPROM_DO

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

### 9.4.5 EEPROM_KEY

Таблица 30 – Регистр ключа EEPROM_KEY

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

Таблица 31 – Описание бит регистра ключа EEPROM_KEY

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…0 | KEY[31:0] | Ключ для разрешения доступа к флэш-памяти через регистровый доступ.<br>Перед переводом памяти в режим программирования необходимо в регистр EEPROM_KEY записать комбинацию 0x8AAA5551 |

# 10 Процессорное ядро

Процессорное ядро с минимизированным количеством вентилей обладает следующими характеристиками:

– имеет в своем составе трехуровневый конвейер;

– имеется набор инструкций архитектуры ARM v6-M, включающий 32-битные инструкции Thumb-2, такие как BL, MRS, MSR, ISB, DSB и DMB;

– 32-разрядный аппаратный умножитель;

– имеется возможность запуска операционной системы и доступные для этого режима работы инструкции SVC, групповой регистр указателя стека и интегрированный системный таймер;

– имеется системная модель исключительных ситуаций;

– имеются режимы Handler и Thread;

– имеется два указателя стека;

– имеется возможность работы только в режиме Thumb;

– отсутствует аппаратная поддержка невыровненного доступа;

– содержит 13x32 разрядных регистра общего назначения, link регистр (LR), счетчик команд (PC), программный регистр статуса xPSR, и два групповых регистра указателя стека (SP).

Контроллер прерываний NVIC обладает следующими характеристиками:

– контроллер интегрирован в процессор для уменьшения задержек в процессе прерываний;

– контроллер поддерживает до 32 внешних прерываний;

– имеется два бита приоритета, обеспечивающие четырехуровневый приоритет прерываний;

– состояние процессора автоматически сохраняется при входе в прерывание и восстанавливается при выходе, что не вызывает потерь на выполнение инструкций.

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

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

Полный набор отладочных модулей:

– полный доступ в режиме останова ко всей памяти и регистрам;

– отладочный порт DAP;

– модуль точек останова BPU;

– модуль наблюдения данных DW.

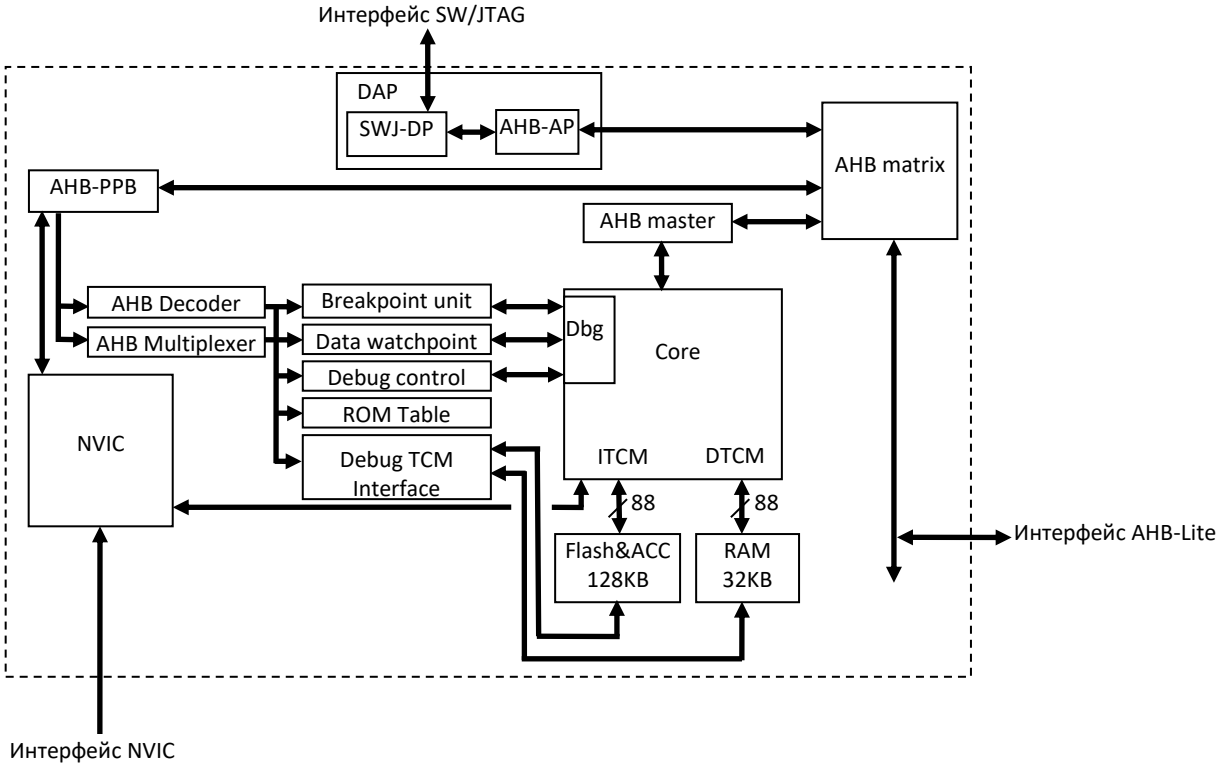



Рисунок 16 – Структурная блок-схема процессорного ядра

Периферийными блоками ядра являются:

– контроллер прерываний NVIC – реализует высокоскоростную обработку прерываний;

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

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

– AHB decoder – дешифрирует адреса AHB шины для выработки сигналов выбора для периферии системы отладки;

– AHB multiplexer – объединяет все ответы ведомых для отладочных блоков;

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

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

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

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

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

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

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

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

## 10.1 Программная модель

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

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

Процессор может функционировать в режимах:

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

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

Процессор может функционировать в одном из состояний:

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

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

## 10.2 Стек

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

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

## 10.3 Регистры ядра

Процессор содержит следующие 32-разрядные регистры:

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

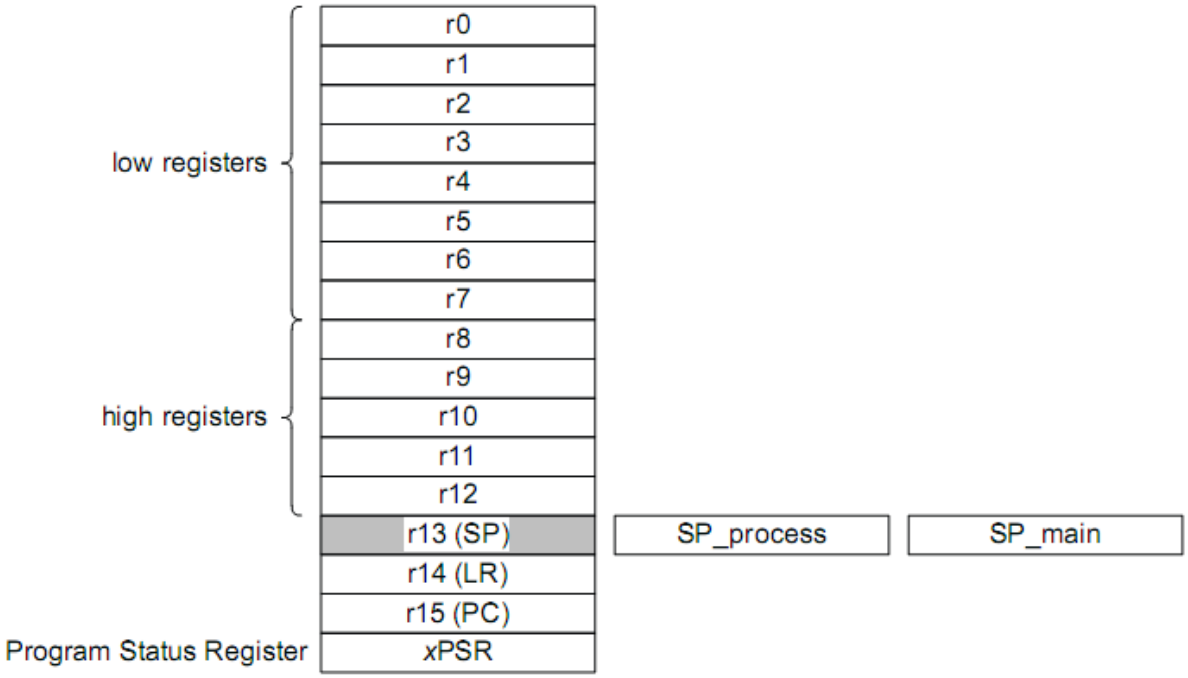



Рисунок 17 – Регистры ядра

### 10.3.1 Регистры общего назначения R0-R12

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

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

### 10.3.2 Указать стека SP R13

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

### 10.3.3 Регистр связи LR R14

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

### 10.3.4 Счетчик команд PC R15

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

### 10.3.5 Программный регистр состояния PSR

Регистр Program Status Register (PSR) объединяет:

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

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

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

Таблица 32 – Комбинация PSR и их атрибуты

| Регистр | Тип | Комбинация |
|---|---|---|
| XPSR | RW(1),(2) | APSR, EPSR и IPSR |
| IEPSR | RO | EPSR и IPSR |
| IAPSR | RW(1) | APSR и IPSR |
| EAPSR | RW(2) | APSR и EPSR |

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

### 10.3.6 Программный регистр состояния приложения APSR

Регистр APSR содержит текущие флаги состояния выполнения предыдущей инструкции.

Таблица 33 – Регистр APSR

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

Таблица 34 – Описание бит регистра APSR

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31 | N | Negative:<br>0 – результат операции положительный, нулевой, больше чем, равен;<br>1 – результат операции отрицательный или меньше чем |
| 30 | Z | Zero:<br>0 – результат операции не нулевой;<br>1 – результат операции нулевой |
| 29 | C | Carry:<br>0 – при суммировании не было переноса, при вычитании не было заема;<br>1 – при суммировании был перенос, при вычитании был заем |
| 28 | V | Overflow:<br>0 – в результате операции не было переполнения;<br>1 – в результате операции было переполнение |
| 27…0 | - | Зарезервировано |

### 10.3.7 Программный регистр состояния прерываний IPSR

Регистр IPSR содержит номер типа исключения для текущего обработчика прерывания.

Таблица 35 – Регистр IPSR

| Номер | 31…6 | 5…0 |
|---|---|---|
| Доступ | | |
| Сброс | | |
| | - | ISR_NUMBER |

Таблица 36 – Описание бит регистра IPSR

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

### 10.3.8 Программный регистр состояния выполнения EPSR

Регистр EPSR содержит бит состояния инструкции Thumb.

Таблица 37 – Регистр EPSR

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

Таблица 38 – Описание бит регистра EPSR

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

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

### 10.3.9 Сохранение битов xPSR

При входе в прерывание процессор сохраняет сгруппированные данные из трех регистров статуса в стек. Бит 9, помещенных в стек регистров xPSR, содержит статус выравнивания активного SP, когда начинается процесс обработки прерывания.

### 10.3.10 Регистр маски приоритетов Priority Mask

Регистр PRIMASK используется для повышения приоритета.

Таблица 39 – Регистр PRIMASK

| Номер | 31…1 | 0 |
|---|---|---|
| Доступ | | |
| Сброс | | |
| | - | PRIMASK |

Таблица 40 – Описание бит регистра PRIMASK

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

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

### 10.3.11 Контрольный регистр специального назначения CONTROL

Регистр определяет текущий указатель стека.

Таблица 41 – Регистр CONTROL

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

Таблица 42 – Описание бит регистра CONTROL

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

## 10.4 Типы данных

Процессор поддерживает следующие типы данных:

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

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

# 11 Система команд

В процессоре реализована версия системы команд Thumb. Поддерживаемые команды приведены в таблице 43.

В таблице используются следующие обозначения:

– в угловых скобках <> записываются альтернативные формы представления операндов;

– в фигурных скобках {} указываются необязательные операнды;

– информация в столбце «операнды» может быть неполной.

Таблица 43 – Система команд процессора

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

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

## 11.1 Встроенные функции

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

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

Таблица 44 – Встроенные функции CMSIS, позволяющие генерировать некоторые инструкции процессора

| Мнемокод команды процессора | Описание встроенной функции |
|---|---|
| CPSIE I | void __enable_irq(void) |
| CPSID I | void __disable_irq(void) |
| ISB | void __ISB(void) |
| DSB | void __DSB(void) |
| DMB | void __DMB(void) |
| NOP | void __NOP(void) |
| REV | uint32_t __REV(uint32_t int value) |
| REV16 | uint32_t __REV16(uint32_t int value) |
| REVSH | uint32_t __REVSH(uint32_t int value) |
| WFE | void __WFE(void) |
| WFI | void __WFI(void) |

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

Таблица 45 – Встроенные функции CMSIS для доступа к специальным регистрам процессора

| Наименование специального регистра | Режим доступа | Описание встроенной функции |
|---|---|---|
| PRIMASK | Чтение | uint32_t __get_PRIMASK (void) |
| | Запись | void __set_PRIMASK (uint32_t value) |
| CONTROL | Чтение | uint32_t __get_CONTROL (void) |
| | Запись | void __set_CONTROL (uint32_t value) |
| MSP | Чтение | uint32_t __get_MSP (void) |
| | Запись | void __set_MSP (uint32_t TopOfMainStack) |
| PSP | Чтение | uint32_t __get_PSP (void) |
| | Запись | void __set_PSP (uint32_t TopOfProcStack) |

## 11.2 Описание инструкций

В разделе представлена подробная информация об инструкциях процессора:

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

### 11.2.1 Операнды

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

### 11.2.2 Ограничения на использование PC и SP

Многие инструкции не позволяют использовать регистры счетчика команд (PC) и указателя стека (SP) в качестве регистра-получателя. Подробная информация содержится в описании конкретных инструкций.

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

### 11.2.3 Операции сдвига

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

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

ASR

Арифметический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. Бит [31] исходного значения регистра записывается в n крайних слева бит результата (см. рисунок 18).

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

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

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

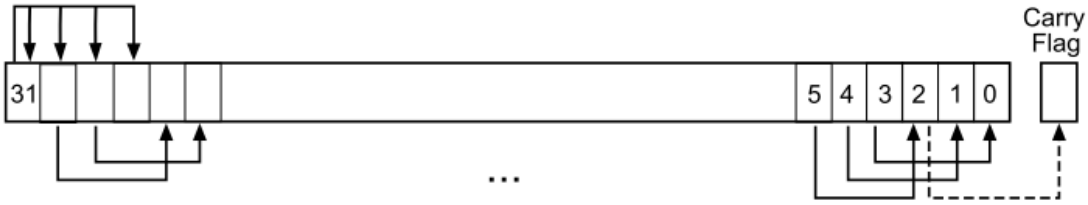



Рисунок 18 – Инструкция ASR # 3

LSR

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

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

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

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

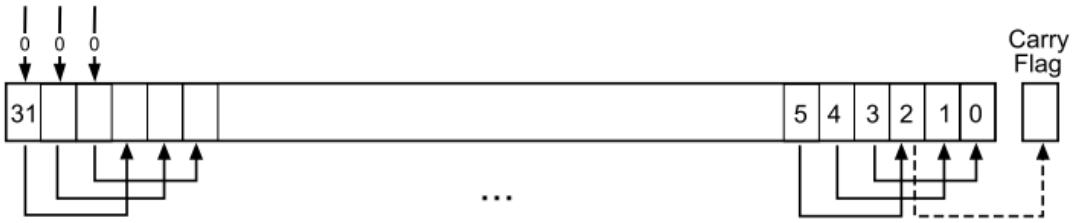



Рисунок 19 – Инструкция LSR # 3

LSL

Логический сдвиг влево на n бит переносит крайние справа 32-n бит регистра Rm влево на n позиций, то есть на место крайних слева 32-n. При этом в n крайних слева бит результата записывается 0 (см. рисунок 20).

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

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

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

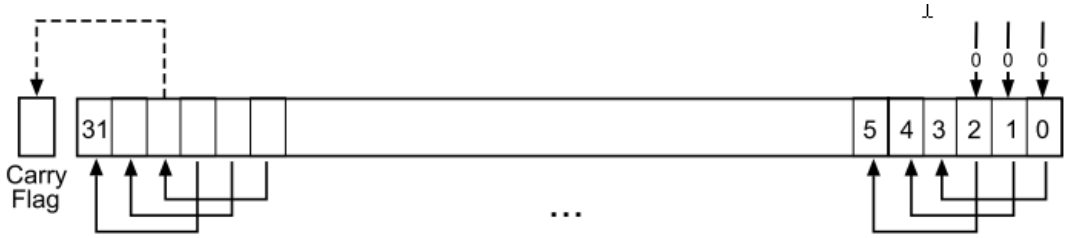



Рисунок 20 – Инструкция LSL # 3

ROR

Циклический сдвиг вправо на n бит переносит крайние слева 32-n бит регистра Rm вправо на n позиций, то есть на место крайних справа 32-n. При этом n крайних справа разрядов регистра переносятся в крайние n слева разрядов результата (см. рисунок 21).

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

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

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

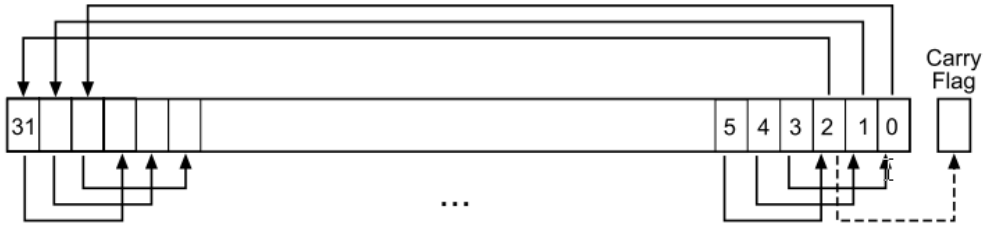



Рисунок 21 – Инструкция ROR # 3

### 11.2.4 Выравнивание адресов

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

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

При попытке доступа по невыровненному адресу процессором формируется исключение HardFault, что указано выставленным битом UNALIGN_TRP регистра конфигурации и управления CCR (см. подраздел 31.9 «Регистр конфигурации и управления (CCR)»).

### 11.2.5 Адресация относительно счетчика команд PC

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

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

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

### 11.2.6 Условное исполнение

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

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

Возможность исполнения или неисполнения команды, в зависимости от значения флагов условий, сформированных ранее, может быть достигнута за счет использования условных переходов. Условный переход может быть выполнен:

При наличии одного из указанных суффиксов процессор проверяет значение флагов на соответствие заданному условию. Если условие не выполняется, то инструкция:

– сразу после команды, которая обновляет флаги условий;

– после любого количества промежуточных команд, которые не обновляют флаги условий.

Процессорное ядро поддерживает только одну инструкцию условного перехода B<c> (Branch), где <c> один из суффиксов условного исполнения.

Ниже в разделе рассматриваются:

– флаги условий;

– суффиксы условного исполнения.

Флаги условий

Регистр состояния прикладной программы APSR содержит следующие флаги:

– N=1 в случае, если результат операции меньше нуля, 0 в противном случае;

– Z=1 в случае, если результат равен нулю, 0 в противном случае;

– C=1 в случае, если при выполнении операции возник перенос, 0 в противном случае;

– V=1 в случае, если при выполнении операции возникло переполнение, 0 в противном случае.

Перенос возникает в следующих случаях:

– результат сложения оказался больше или равен $2^{32}$;

– результат вычитания больше или равен нулю;

– в результате работы внутренней логики процессора при операциях загрузки данных и логических операций.

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

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

Суффиксы условного исполнения

Условный переход в описании синтаксиса это обозначается как B{cond}. Команда перехода с кодом условия выполняется только в том случае, если флаги регистра APSR соответствуют указанному условию, в противном случае команда перехода игнорируется. В таблице 46 приведены доступные коды условий и соответствующие им флаги условий N, Z, C, V.

Таблица 46 – Суффиксы условного исполнения

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

## 11.3 Команды доступа к памяти

Обобщенные данные о командах доступа к памяти демонстрирует таблице 47.

Таблица 47 – Команды доступа к памяти

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

| Мнемокод | Краткое описание |
|---|---|
| PUSH | Загрузка регистров в стек |
| STM | Сохранение множества регистров |
| STR{type} | Сохранение регистра, непосредственно указанное смещение |
| STR{type} | Сохранение регистра, смещение указано в регистре |

### 11.3.1 ADR

Загрузка адреса, заданного относительно счетчика команд.

Синтаксис
ADR Rd, label

где Rd – регистр-получатель;

Label – относительный адрес, см. пункт 11.2.5 «Адресация относительно счетчика команд PC».

Описание

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

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

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

Ограничения

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

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

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

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

### 11.3.2 LDR и STR, непосредственно заданное смещение

Загрузка или сохранение регистра в режиме адресации со смещением.

Синтаксис

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

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

Rn – регистр, содержащий базовый адрес памяти;

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

Описание

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

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

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

Ограничения

Для данных команд:

- в качестве Rt и Rn можно использовать только регистры R0-R7;
- смещение imm должно быть задано:
  - целым числом от 0 до 1020, кратным 4, для инструкций LDR и STR с использованием SP в качестве регистра, содержащий базовый адрес памяти;
  - целым числом от 0 до 124, кратным 4, для инструкций LDR и STR с использованием R0-R7 в качестве регистра, содержащий базовый адрес памяти;
  - целым числом от 0 до 62, кратным 2, для инструкций LDRH и STRH;
  - целым числом от 0 до 31 для инструкций LDRB и STRB;
- вычисленный адрес должен делиться без остатка на количество байт в транзакции, см. пункт 11.2.4 «Выравнивание адресов».

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

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

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

### 11.3.3 LDR и STR, смещение задано в регистре

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

Синтаксис

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

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

Rn – регистр, содержащий базовый адрес памяти;

Rm – регистр, содержащий смещение относительно базового адреса.

Описание

LDR – загружает регистра значением из памяти.

STR – сохраняет значение регистра в памяти.

Адрес области памяти, в которую будет производиться обращение, вычисляется на основании значения базового адреса в регистре Rn и смещения. Смещение определяется значением регистра Rm и параметром сдвига влево значения этого регистра.

Считываемое или записываемое значение может иметь размер байта, полуслова или слова. При загрузке данных из памяти байты и полуслова могут интерпретироваться либо как числа со знаком, либо как беззнаковые. См. пункт 11.2.4 «Выравнивание адресов».

Ограничения

Для данных команд:

– в качестве операндов Rt, Rn и Rm можно использовать только регистры R0-R7;

– вычисленный адрес должен делиться без остатка на количество байт в транзакции, см. пункт 11.2.4 «Выравнивание адресов».

В случае если в команде загрузки слова в качестве регистра Rt используется счетчик команд PC. Бит [0] загружаемого значения должен быть равен 1, передача управления при этом осуществляется по выровненному по границе полуслова адресу.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

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

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

### 11.3.4 LDR, адресация относительно счетчика команд PC

Загрузка регистра из памяти.

Синтаксис

LDR Rt, label

где Rt – регистр, в который должна производиться загрузка;

Label – относительный адрес, см. пункт 11.2.5 «Адресация относительно счетчика команд PC».

Описание

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

Ограничения

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

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

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

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

### 11.3.5 LDM и STM

Загрузка или сохранение множества регистров.

Синтаксис

LDM Rn{!}, reglist

STM Rn!, reglist

где Rn – регистр, содержащий базовый адрес памяти.

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

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

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

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

Описание

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

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

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

Ограничения

В описываемых в разделе командах:

- в списке reglist и в качестве Rn можно использовать только регистры R0-R7;
- суффикс обратной записи должен использоваться всегда. Исключение составляет команда LDM, в которой в списке reglist содержится регистр Rn. В этом случае суффикс обратной записи использовать нельзя;
- значение адреса в регистре Rn должно быть выровнено на границу слова, см. пункт 11.2.4 «Выравнивание адресов»;
- если в команде STM регистр Rn указан в списке reglist, то в таком случае Rn должен быть первым регистром в списке, т.е. иметь наименьший номер.

Флаги

Данные команды не влияют на состояние флагов.

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

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

### 11.3.6 PUSH и POP

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

Синтаксис
PUSH reglist
POP reglist

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

Описание

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

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

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

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

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

Ограничения

В данных командах:

– в списке регистров reglist можно использовать только регистры R0-R7;

– исключением является регистр LR для команды PUSH, а также регистр PC для команды POP.

Флаги

Данные команды не влияют на состояние флагов.

Примеры

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

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

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

## 11.4 Инструкции обработки данных

Таблица 48 показывает инструкции обработки данных.

Таблица 48 – Команды обработки данных

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

### 11.4.1 ADD, ADC, SUB, SBC и RSB

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

Примечание

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

Синтаксис

ADD{S} {Rd,} Rn, <Rm|#imm>
ADCS {Rd,} Rn, Rm
SUB{S} {Rd,} Rn, <Rm|#imm>
SBCS {Rd,} Rn, Rm
RSBS {Rd,} Rn, #0

где S – необязательный суффикс для инструкций ADD и SUB. Если он указан, результат выполнения операции приводит к обновлению флагов, см. пункт 11.2.6 «Условное исполнение»;

Rd – регистр-получатель результата;

Rn – регистр, содержащий значение первого операнда;

Rm – регистр, содержащий значение второго операнда;

imm – определяет непосредственное значение константы.

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

Описание

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

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

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

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

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

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

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

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

См. также описание команды «ADR».

Ограничения

В таблице 49 представлены допустимые комбинации регистров и значений констант imm, которые можно использовать применительно к указанным инструкциям.

Таблица 49 – Ограничения команд ADC, ADD, RSB, SBC и SUB

| Инструкция | Rd | Rn | Rm | imm | Ограничения |
|---|---|---|---|---|---|
| ADCS | R0-R7 | R0-R7 | R0-R7 | - | *Rd* и *Rn* должны задавать один и тот же регистр |
| ADD | R0-R15 | R0-R15 | R0-R15 | - | *Rd* и *Rn* должны задавать один и тот же регистр<br>*Rn* и *Rm* не должны оба задавать PC (R15) |
| | R0-R7 | SP или PC | - | 0-1020 | Значение *imm* должно быть целым числом кратным 4 |
| | SP | SP | - | 0-508 | Значение *imm* должно быть целым числом кратным 4 |

| Инструкция | Rd | Rn | Rm | imm | Ограничения |
|---|---|---|---|---|---|
| ADDS | R0-R7 | R0-R7 | - | 0-7 | - |
| | R0-R7 | R0-R7 | - | 0-255 | *Rd* и *Rn* должны задавать один и тот же регистр |
| | R0-R7 | R0-R7 | R0-R7 | - | - |
| RSBS | R0-R7 | R0-R7 | - | - | - |
| SBCS | R0-R7 | R0-R7 | R0-R7 | - | *Rd* и *Rn* должны задавать один и тот же регистр |
| SUB | SP | SP | - | 0-508 | Значение *imm* должно быть целым числом кратным 4 |
| SUBS | R0-R7 | R0-R7 | - | 0-7 | - |
| | R0-R7 | R0-R7 | - | 0-255 | *Rd* и *Rn* должны задавать один и тот же регистр |
| | R0-R7 | R0-R7 | R0-R7 | - | - |

Примеры

64-разрядное сложение

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

```
ADDS R0, R0, R2 ; сложить младшие значащие слова
ADCS R1, R1, R3 ; сложить старшие значащие слова с учетом флага переноса
```

96-разрядное вычитание

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

```
SUBS R4, R4, R1 – вычитание младших значащих слов
SBCS R5, R5, R2 – вычитание средних значащих слов с учетом флага переноса
SBCS R6, R6, R3 – вычитание старших значащих слов с учетом флага переноса
```

Во фрагменте кода ниже приведен пример использования команды RSBS.

```
RSBS R7, R7, #0 – вычитание R7 из нуля.
```

### 11.4.2 AND, ORR, EOR, BIC

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

Примечание

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

Синтаксис

```
ANDS {Rd,} Rn, Rm
```

ORRS {Rd,} Rn, Rm
EORS {Rd,} Rn, Rm
BICS {Rd,} Rn, Rm

где Rd – регистр назначения;

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

Rm – второй регистр.

Описание

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

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

Ограничения

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

Флаги

Данные инструкции:

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

Примеры

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

### 11.4.3 ASR, LSL, LSR, ROR

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

Синтаксис

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

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

Rm – регистр, значение которого должно быть подвергнуто сдвигу;

Rs – регистр, содержащий величину сдвига значения регистра Rm;

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

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

Примечание

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

Описание

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

Детальное описание операций сдвига представлено в разделе 11.2.3 «Операции сдвига».

Ограничения

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

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

Флаги

- Данные инструкции обновляют флаги N и Z в соответствии с результатом.
- Флаг C обновляется до значения последнего сдвинутого бита, кроме случая, когда длина сдвига равна нулю, см. пункт 11.2.3 «Операции сдвига». Флаг V не изменяется.

Примеры

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

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

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

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

### 11.4.4 CMP и CMN

Сравнение и сравнение с противоположным знаком.

Синтаксис

CMN Rn, Rm
CMP Rn, #imm
CMP Rn, Rm

где Rn – регистр, хранящий значение первого операнда;
Rm – регистр, с которым нужно проводить сравнение;
imm – значение, с которым нужно проводить сравнение.

Описание

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

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

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

Ограничения

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

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

Флаги

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

Примеры

CMP R2, R9
CMN R0, R2.

### 11.4.5 MOV и MVN

Загрузка в регистр прямого или инверсного значения

Синтаксис

MOV{S} Rd, Rm
MOVS Rd, #imm
MVNS Rd, Rm

где S – необязательный суффикс. Если он указан, результат выполнения операции приводит к обновлению соответствующих флагов, см. пункт 11.2.6 «Условное исполнение»;

Rd – регистр-получатель результата;

Rm – регистр-источник данных;

imm – любое значение в диапазоне 0-255.

Описание

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

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

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

Ограничения

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

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

– бит [0] значения, загружаемого в PC, игнорируется;

– передача управления осуществляется по адресу, соответствующему загруженному значению с битом [0], принудительно установленным в 0. T-бит не модифицируется.

Примечание

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

Флаги

В случае, если указан суффикс S, то инструкция:

– обновляет флаги N и Z в соответствии с результатом выполнения операции;

– не изменяет значения флагов C и V.

Примеры

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

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

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

MOVS R3, #23 – Записать значение 23 в R3
MOV R8, SP – Записать значение указателя стека в регистр R8
MVNS R2, R0 – Записать инверсное значение R0 в R2 и обновить флаги.

### 11.4.6 MULS

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

Синтаксис
MULS Rd, Rn, Rm

где Rd – регистр-получатель результата;
Rn, Rm – регистры, содержащие значения, которые нужно перемножить.

Описание
Инструкция MULS производит умножение значений регистров Rn и Rm и помещает младшие 32 бита результата в регистр Rd. Флаги обновляются в соответствии с результатом выполнения операции, см. пункт 11.2.6 «Условное исполнение».
Результат выполнения операции не зависит от того, используются ли знаковые или беззнаковые операнды.

Ограничения
В качестве операндов Rd, Rn, Rm можно использовать только регистры R0 – R7.
Rd должен быть тем же регистром, что и Rm.

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

Примеры
MULS R0, R2, R0 – умножение с обновлением флагов, R0 = R0 x R2.

### 11.4.7 REV, REV16, REVSH

Изменение порядка байтов в слове.

Синтаксис
REV Rd, Rn
REV16 Rd, Rn
REVSH Rd, Rn

где Rd – регистр-получатель результата;

Rn – регистр-источник данных.

Описание

Инструкции предназначены для изменения формата представления (endianness) данных:

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

Ограничения

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

Флаги

Данные инструкции не влияют на состояние флагов.

Примеры

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

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

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

### 11.4.8 SXT и UXT

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

Синтаксис

```
SXTB Rd, Rm
SXTH Rd, Rm
UXTB Rd, Rm
UXTH Rd, Rm
```

где Rd – регистр-получатель результата;

Rm – регистр, содержащий значение, которое нужно дополнить.

Описание

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

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

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

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

Ограничения

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

Флаги

Данные инструкции не влияют на состояние флагов.

Примеры

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

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

### 11.4.9 TST

Проверить значение битов по маске.

Синтаксис

TST Rn, Rm

где Rn – регистр, содержащий первый операнд;

Rm – регистр, использующийся для проверки соответствия.

Описание

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

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

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

Ограничения

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

Флаги

Данная инструкция:

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

Примеры

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

## 11.5 Инструкции передачи управления

Таблица 50 показывает список инструкций передачи управления.

Таблица 50 – Инструкции передачи управления

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

### 11.5.1 B, BL, BX и BLX

Команды ветвления.

Синтаксис

B {cond} label
BL label
BX Rm
BLX Rm

где cond – необязательный код условия, см. пункт 11.2.6 «Условное исполнение»;

label – относительный адрес, см. пункт 11.2.5 «Адресация относительно счетчика команд PC»;

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

Описание

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

– команды BL и BLX записывают адрес следующей инструкции в регистр связи LR (R14);

– команды BX и BLX формируют отказ (Hard fault) в случае, если бит [0] регистра Rm равен 0.

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

Таблица 51 показывает диапазон адресуемых переходов для различных команд ветвления.

Таблица 51 – Диапазон адресуемых переходов для команд ветвления

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

Ограничения

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

Флаги

Данные инструкции не влияют на состояние флагов.

Примеры

B loopA – передача управления по адресу, обозначенному меткой loopA

BL funC – переход со связью (вызов функции) в funC, адрес возврата будет записан в регистр LR

BX LR – возврат из функции

BLX R0 – переход со связью (вызов функции) по адресу, записанному в R0

BEQ labelD – условный переход на метку labelD, если последняя инструкция, изменяющая флаги в регистре APSR, установила флаг Z, иначе переход выполнен не будет.

## 11.6 Прочие инструкции

В таблице 52 приведены инструкции процессора, не рассмотренные в предыдущих разделах.

Таблица 52 – Прочие инструкции

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

### 11.6.1 BKPT

Точка останова.

Синтаксис

BKPT #imm

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

Описание

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

Ограничения

Данная инструкция не имеет ограничений.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

BKPT #0 – Точка останова со значением 0x00

### 11.6.2 CPS

Изменить состояние процессора.

Синтаксис
CPSID i
CPSIE i

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

Ограничения
Данная инструкция не имеет ограничений.

Флаги
Данная инструкция не влияет на состояние флагов.

Примеры
CPSID i – Запретить все прерывания, кроме NMI (установить PRIMASK)
CPSIE i – Разрешить прерывания (сбросить PRIMASK)

### 11.6.3 DMB

Барьер доступа к памяти данных.

Синтаксис
DMB

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

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

Ограничения
Данная инструкция не имеет ограничений.

Флаги
Данная инструкция не влияет на состояние флагов.

Примеры

DMB – Барьер доступа к памяти данных

### 11.6.4 DSB

Барьер синхронизации доступа к памяти данных.

Синтаксис
DSB

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

Ограничения
Данная инструкция не имеет ограничений.

Флаги
Данная инструкция не влияет на состояние флагов.

Примеры
DSB – Барьер синхронизации доступа к памяти данных

### 11.6.5 ISB

Барьер синхронизации доступа к инструкциям.

Синтаксис
ISB

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

Ограничения
Данная инструкция не имеет ограничений.

Флаги
Данная инструкция не влияет на состояние флагов.

Примеры
ISB – Барьер синхронизации доступа к инструкциям

### 11.6.6 MRS

Считать содержимое специального регистра в регистр общего назначения.

Синтаксис

MRS Rd, spec_reg

где Rd – регистр-получатель результата;

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

Описание

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

См. также описание инструкции «MSR».

Ограничения

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

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

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

### 11.6.7 MSR

Записать регистр общего назначения в специальный регистр.

Синтаксис

MSR spec_reg, Rn

где Rn – регистр-источник данных;

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

Описание

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

См. также описание инструкции «MRS».

Ограничения

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

Флаги

Данная инструкция обновляет флаги на основе значения в регистре Rn.

Примеры

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

### 11.6.8 NOP

Нет операции.

Синтаксис

NOP

Описание

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

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

Ограничения

Данная инструкция не имеет ограничений.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

NOP – Нет операции.

### 11.6.9 SVC

Вызов супервизора.

Синтаксис

SVC #imm

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

Описание

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

Ограничения

Данная инструкция не имеет ограничений.

Флаги

Данная инструкция не влияет на состояние флагов.

Примеры

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

### 11.6.10 WFE

Выполняется как команда NOP. Управление режимом пониженного энергопотребления осуществляется в регистре ETH_CLOCK, бит SLEEP.

### 11.6.11 WFI

Выполняется как команда NOP. Управление режимом пониженного энергопотребления осуществляется в регистре ETH_CLOCK, бит SLEEP.

# 12 Сигналы тактовой частоты

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

Управление тактовыми частотами ведется через периферийный блок RST_CLK. При включении питания микросхема запускается на частоте генератора HSI. Выдача тактовых сигналов синхронизации для всех периферийных блоков кроме RST_CLK отключена. Для начала работы с нужным периферийным блоком необходимо включить его тактовую частоту в регистре PER_CLOCK. Некоторые контроллеры интерфейсов (UART, CAN, USB, Таймеры) могут работать на частотах отличных от частоты процессорного ядра, поэтому в соответствующих регистрах (UART_CLOCK, CAN_CLOCK, USB_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 с помощью соответствующих мультиплексоров, управляемых регистрами CPU_CLOCK и USB_CLOCK.

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

При смене тактовой частоты процессорного ядра следует:

1. Настроить число тактов паузы для доступа к Flash-памяти, задаваемое полем Delay в регистре EEPROM_CMD контроллера флэш-памяти EEPROM_CNTRL.

2. Настроить режим работы встроенного регулятора 1,8 В, задаваемый полем LOW регистра REG_0E батарейного домена BKP, совместно с дополнительной нагрузкой для регулятора 1,8 В, задаваемой полем SelectRI регистра REG_0E батарейного домена BKP.

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

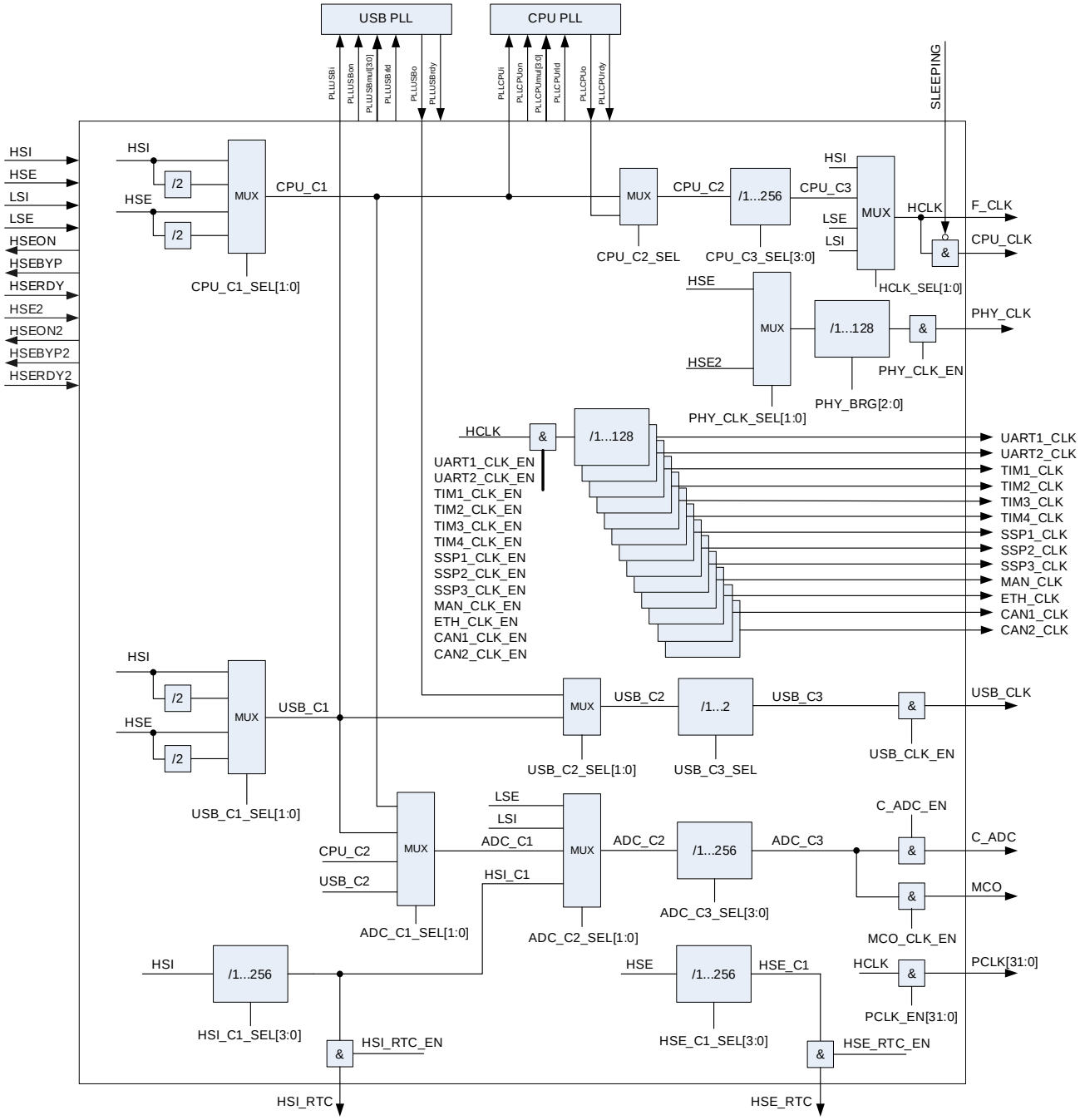



Рисунок 22 – Структурная блок-схема формирования тактовой частоты

## 12.1 Встроенный RC-генератор HSI

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

### 12.2 Встроенный RC-генератор LSI

Генератор LSI вырабатывает тактовую частоту $f_{O_LSI}$ с типовым значением 40 КГц. Генератор автоматически запускается при появлении питания $U_{CC}$ и при выходе в нормальный режим работы вырабатывает сигнал LSIRDY в регистре BKP_REG_0F. Первоначально тактовая частота генератор LSI используется для формирования дополнительной задержки 4 мс. При дальнейшей работе генератор LSI может быть отключен при помощи сигнала LSION в регистре BKP_REG_0F.

### 12.3 Внешний осциллятор HSE

Осциллятор HSE предназначен для выработки тактовой частоты 2…16 МГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания Ucc и сигнала разрешения HSEON в регистре HS_CONTROL. При выходе в нормальный режим работы вырабатывает сигнал HSERDY в регистре CLOCK_STATUS. Также осциллятор может работать в режиме HSEBYP, когда входная тактовая частота с входа OSC_IN проходит напрямую на выход HSE, выход OSC_OUT находится в этом режиме в третьем состоянии.

### 12.4 Внешний осциллятор HSE2

Осциллятор HSE2 предназначен для выработки тактовой частоты 20…30 МГц с помощью внешнего резонатора на выводах PF0 и PF1. Осциллятор запускается при появлении питания Ucc и сигнала разрешения HSEON2 в регистре HS_CONTROL. При выходе в нормальный режим работы вырабатывает сигнал HSERDY2 в регистре CLOCK_STATUS. Также осциллятор может работать в режиме HSEBYP2, когда входная тактовая частота с входа OSC_IN25 проходит напрямую на выход HSE2, выход OSC_OUT25 находится в этом режиме в третьем состоянии.

### 12.5 Внешний осциллятор LSE

Осциллятор LSE предназначен для выработки тактовой частоты 32 КГц с помощью внешнего резонатора. Осциллятор запускается при появлении питания BDUcc и сигнала разрешения LSEON в регистре BKP_REG_0F. При выходе в нормальный режим работы вырабатывает сигнал LSERDY в регистре BKP_REG_0F. Также осциллятор может работать в режиме LSEBYP, когда входная тактовая частота с входа OSC_IN32 проходит напрямую на выход LSE. Выход OSC_OUT32 находится в этом режиме в третьем состоянии. Так как генератор LSE питается от напряжения питания BDUcc и его регистр управления BKP_REG_0F расположен в батарейном домене, то генератор может продолжать работать при пропадании основного питания Ucc. Генератор LSE используется для работы часов реального времени.

### 12.6 Встроенный блок умножения системной тактовой частоты

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 4 до 16, задаваемый на входе PLLCPUMUL[3:0] в регистре PLL_CONTROL. Входная частота блока умножителя должна быть в диапазоне от 6 до 16 МГц, выходная – до 144 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 используется как основная частота процессора и периферии.

### 12.7 Встроенный блок умножения тактовой частоты для контроллера USB

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 4 до 16, задаваемый на входе PLLUSBMUL[3:0] в регистре PLL_CONTROL. Входная частота блока умножителя должна быть в диапазоне от 6 до 16 МГц, выходная частота должна составлять 48 МГц. При выходе блока умножителя тактовой частоты в расчетный режим вырабатывается сигнал PLLUSBRDY в регистре CLOCK_STATUS. Блок включается с помощью сигнала PLLUSBON в регистре PLL_CONTROL. Выходная частота используется как основная частота протокольной части интерфейса USB.

### 12.8 Описание регистров блока контроллера тактовой частоты

Таблица 53 – Описание регистров блока контроллера тактовой частоты

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x4002_0000 | RST_CLK | Контроллер тактовой частоты |
| Смещение | | |
| 0x00 | CLOCK_STATUS | Регистр состояния блока управления тактовой частотой |
| 0x04 | PLL_CONTROL | Регистр управления блоками умножения частоты |
| 0x08 | HS_CONTROL | Регистр управления высокочастотным генератором и осциллятором |
| 0x0C | CPU_CLOCK | Регистр управления тактовой частотой процессорного ядра |
| 0x10 | USB_CLOCK | Регистр управления тактовой частотой контроллера USB |
| 0x14 | ADC_MCO_CLOCK | Регистр управления тактовой частотой АЦП |
| 0x18 | RTC_CLOCK | Регистр управления формированием высокочастотных тактовых сигналов блока RTC |
| 0x1C | PER_CLOCK | Регистр управления тактовой частотой периферийных блоков |
| 0x20 | CAN_CLOCK | Регистр управления тактовой частотой CAN |
| 0x24 | TIM_CLOCK | Регистр управления тактовой частотой TIMER |
| 0x28 | UART_CLOCK | Регистр управления тактовой частотой UART |
| 0x2C | SSP_CLOCK | Регистр управления тактовой частотой SSP |
| 0x34 | ETH_CLOCK | Регистр управления тактовой частотой Ethernet и ГОСТР52070-2003 |
| 0x38 | DMA_DONE_STICK | Регистр фиксации сигналов окончания работы DMA |

### 12.8.1 CLOCK_STATUS

Таблица 54 – Регистр CLOCK_STATUS

| Номер | 31…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | U | RO | RO | RO | RO |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | HSE RDY2 | HSE RDY | PLL CPU RDY | PLL USB RDY |

Таблица 55 – Описание бит регистра CLOCK_STATUS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…4 | - | Зарезервировано |
| 3 | HSE RDY2 | Флаг выхода в рабочий режим осциллятора HSE2:<br>0 – осциллятор не запущен или не стабилен;<br>1 – осциллятор запущен и стабилен |
| 2 | HSE RDY | Флаг выхода в рабочий режим осциллятора HSE:<br>0 – осциллятор не запущен или не стабилен;<br>1 – осциллятор запущен и стабилен |
| 1 | PLL CPU RDY | Флаг выхода в рабочий режим CPU PLL:<br>0 – PLL не запущена или не стабильна;<br>1 – PLL запущена и стабильна.<br>Для обновления состояния PLL_CPU_RDY при PLL_CPU_RDY = 1 перед изменением PLL_CPU_ON необходимо установить PLL_CPU_RLD = 1, а затем одновременно с изменением PLL_CPU_ON записать 0 в PLL_CPU_RLD |
| 0 | PLL USB RDY | Флаг выхода в рабочий режим USB PLL:<br>0 – PLL не запущена или не стабильна;<br>1 – PLL запущена и стабильна.<br>Для обновления состояния PLL_USB_RDY при PLL_USB_RDY = 1 перед изменением PLL_USB_ON необходимо установить PLL_USB_RLD = 1, а затем одновременно с изменением PLL_USB_ON записать 0 в PLL_USB_RLD |

### 12.8.2 PLL_CONTROL

Таблица 56 – Регистр PLL_CONTROL

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

Таблица 57 – Описание бит регистра PLL_CONTROL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…12 | - | Зарезервировано |
| 11….8 | PLL CPU MUL[3:0]* | Коэффициент умножения для CPU PLL<br>PLLCPUo = PLLCPUi • (PLLCPUMUL+1) |
| 7…4 | PLL USB MUL[3:0]* | Коэффициент умножения для USB PLL<br>PLLUSBo = PLLUSBi • (PLLUSBMUL+1) |
| 3 | PLL CPU PLD | Бит перезапуска PLL CPU.<br>При смене коэффициента умножения в рабочем режиме необходимо задать равным 1, а после этого сбросить в ноль |
| 2 | PLL CPU ON | Бит включения PLL CPU:<br>0 – PLL выключена;<br>1 – PLL включена |
| 1 | PLL USB RLD | Бит перезапуска PLL USB.<br>При смене коэффициента умножения в рабочем режиме необходимо задать равным 1, а после этого сбросить в ноль |
| 0 | PLL USB ON | Бит включения PLL USB:<br>0 – PLL выключена;<br>1 – PLL включена |

* Запрещено использование коэффициентов умножения частот PLLCPUi и PLLUSBi, равных 2 и 3, то есть значения битов PLLCPUMUL[3:0] и PLLUSBMUL[3:0] регистра PLL_CONTROL не должны быть равны 1 и 2. Для получения эквивалентных значений частот CPU_CLK и USB_CLK рекомендуется устанавливать PLLCPUMUL[3:0] и PLLUSBMUL[3:0] в значение 3 или 5 с последующим делением PLLCPUo и PLLUSBo на 2 в регистрах CPU_CLOCK и USB_CLOCK (биты CPU C3 SEL[3:0] и USB C3 SEL) соответственно

### 12.8.3 HS_CONTROL

Таблица 58 – Регистра HS_CONTROL

| Номер | 31…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | HSE BYP2 | HSE ON2 | HSE BYP | HSE ON |

Таблица 59 – Описание бит регистра HS_CONTROL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | - | Зарезервировано |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 3 | HSE BYP2 | Бит управления HSE2 осциллятором:<br>0 – режим осциллятора;<br>1 – режим внешнего генератора.<br>Устанавливать совместно с HSE_ON2 = 1 |
| 2 | HSE ON2 | Бит управления HSE2 осциллятором:<br>0 – выключен;<br>1 – включен |
| 1 | HSE BYP | Бит управления HSE осциллятором:<br>0 – режим осциллятора;<br>1 – режим внешнего генератора.<br>Устанавливать совместно с HSE_ON = 1 |
| 0 | HSE ON | Бит управления HSE осциллятором:<br>0 – выключен;<br>1 – включен |

### 12.8.4 CPU_CLOCK

Таблица 60 – Регистр CPU_CLOCK

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

Таблица 61 – Описание бит регистра CPU_CLOCK

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

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 2 | CPU C2 SEL | Биты выбора источника для CPU_C2:<br>0 – CPU_C1;<br>1 – PLLCPUo |
| 1…0 | CPU C1 SEL[1:0] | Биты выбора источника для CPU_C1:<br>00 – HSI;<br>01 – HSI/2;<br>10 – HSE;<br>11 – HSE/2 |

### 12.8.5 USB_CLOCK

Таблица 62 – Регистр USB_CLOCK

| Номер | 31…9 | 8 | 7…5 | 4 | 3 | 2 | 1…0 |
|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | U | R/W | U | R/W | R/W |
| Сброс | 0 | 0 | 000 | 0 | 0 | 0 | 00 |
| | - | USB CLK EN | - | USB C3 SEL | - | USB C2 SEL | USB C1 SEL[1:0] |

Таблица 63 – Описание бит регистра USB_CLOCK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…9 | - | Зарезервировано |
| 8 | USB CLK EN | Бит разрешения тактирования USB:<br>0 – нет тактовой частоты;<br>1 – есть тактовая частота |
| 7…5 | - | Зарезервировано |
| 4 | USB C3 SEL | Биты выбора делителя для USB_C3<br>USB_C3 = USB_C2/(USB_C3_SEL+1) |
| 3 | - | Зарезервировано |
| 2 | USB C2 SEL | Биты выбора источника для USB_C2:<br>0 – USB_C1;<br>1 – PLLUSBo |
| 1…0 | USB C1 SEL[1:0] | Биты выбора источника для USB_C1:<br>00 – HSI;<br>01 – HSI/2;<br>10 – HSE;<br>11 – HSE/2 |

### 12.8.6 ADC_MCO_CLOCK

Таблица 64 – Регистр ADC_MCO_CLOCK

| Номер | 31…14 | 13 | 12 | 11…8 | 7, 6 | 5, 4 | 3, 2 | 1, 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | U | R/W | U | R/W | U | R/W |
| Сброс | 0 | 0 | 0 | 0000 | 00 | 00 | 00 | 00 |
| | - | ADC CLK EN | - | ADC C3 SEL[3:0] | - | ADC C2 SEL[1:0] | - | ADC C1 SEL[1:0] |

Таблица 65 – Описание бит регистра ADC_MCO_CLOCK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…14 | - | Зарезервировано |
| 13 | ADC CLK EN | Бит разрешения выдачи тактовой частоты ADC CLK:<br>0 – запрещен;<br>1 – разрешен |
| 12 | - | Зарезервировано |
| 11…8 | ADC C3 SEL[3:0] | Биты выбора делителя для ADC_C3:<br>0xxx – ADC_C3 = ADC_C2;<br>1000 – ADC_C3 = ADC_C2 / 2;<br>1001 – ADC_C3 = ADC_C2 / 4;<br>1010 – ADC_C3 = ADC_C2 / 8;<br>…<br>1111 – ADC_C3 = ADC_C2 / 256 |
| 7, 6 | - | Зарезервировано |
| 5, 4 | ADC C2 SEL[1:0] | Биты выбора источника для ADC_C2:<br>00 – LSE;<br>01 – LSI;<br>10 – ADC_C1;<br>11 – HSI_C1 |
| 3, 2 | - | Зарезервировано |
| 1, 0 | ADC C1 SEL[1:0] | Биты выбора источника для ADC_C1:<br>00 – CPU_C1;<br>01 – USB_C1;<br>10 – CPU_C2;<br>11 – USB_C2 |

### 12.8.7 RTC_CLOCK

Таблица 66 – Регистр RTC_CLOCK

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

Таблица 67 – Описание бит регистра RTC_CLOCK

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

### 12.8.8 PER_CLOCK

Таблица 68 – Регистр PER_CLOCK

| Номер | 31…5 | 4 | 3…0 |
|---|---|---|---|
| Доступ | R/W | R/W | R/W |
| Сброс | 0 | 1 | 0 |
| | PCLK_EN[31:5] | PCLK_EN[4] | PCLK_EN[3:0] |

Таблица 69 – Описание бит регистра PER_CLOCK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…0 | PCLK EN[31:0] | Биты разрешения тактирования периферийных блоков:<br>0 – запрещено;<br>1 – разрешено.<br>PCLK[0] – CAN1;<br>PCLK[1] – CAN2;<br>PCLK[2] – USB;<br>PCLK[3] – EEPROM_CNTRL;<br>PCLK[4] – RST_CLK. После сброса в состоянии 1;<br>PCLK[5] – DMA;<br>PCLK[6] – UART1;<br>PCLK[7] – UART2;<br>PCLK[8] – SSP1;<br>PCLK[9] – MIL-STD-1553B1;<br>PCLK[10] – MIL-STD-1553B2;<br>PCLK[11] – POWER;<br>PCLK[12] – WWDG;<br>PCLK[13] – IWDG;<br>PCLK[14] – TIMER1;<br>PCLK[15] – TIMER2;<br>PCLK[16] – TIMER3;<br>PCLK[17] – ADC;<br>PCLK[18] – DAC;<br>PCLK[19] – TIMER4;<br>PCLK[20] – SSP2;<br>PCLK[21] – PORTA;<br>PCLK[22] – PORTB;<br>PCLK[23] – PORTC;<br>PCLK[24] – PORTD;<br>PCLK[25] – PORTE;<br>PCLK[26] – ARINC429R;<br>PCLK[27] – BKP;<br>PCLK[28] – ARINC429T;<br>PCLK[29] – PORTF;<br>PCLK[30] – EXT_BUS_CNTRL;<br>PCLK[31] – SSP3 |

### 12.8.9 CAN_CLOCK

Таблица 70 – Регистр CAN_CLOCK

| Номер | 31…26 | 25 | 24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | U | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 00000000 | 00000000 |
| | - | CAN2 CLK EN | CAN1 CLK EN | - | CAN2 BRG [7:0] | CAN1 BRG [7:0] |

Таблица 71 – Описание бит регистра CAN_CLOCK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...26 | - | Зарезервировано |
| 25 | CAN2 CLK EN | Разрешение тактовой частоты на CAN2:<br>0 – нет частоты;<br>1 – есть частота |
| 24 | CAN1 CLK EN | Разрешение тактовой частоты на CAN1:<br>0 – нет частоты;<br>1 – есть частота |
| 23...16 | - | Зарезервировано |
| 15…8 | CAN2 BRG [7:0] | Делитель тактовой частоты CAN2:<br>xxxxx000 – CAN2_CLK == HCLK;<br>xxxxx001 – CAN2_CLK == HCLK/2;<br>xxxxx010 – CAN2_CLK == HCLK/4;<br>…<br>xxxxx111 – CAN2_CLK == HCLK/128 |
| 7…0 | CAN1 BRG [7:0] | Делитель тактовой частоты CAN1:<br>xxxxx000 – CAN1_CLK == HCLK;<br>xxxxx001 – CAN1_CLK == HCLK/2;<br>xxxxx010 – CAN1_CLK == HCLK/4;<br>…<br>xxxxx111 – CAN1_CLK == HCLK/128 |

### 12.8.10 TIM_CLOCK

Таблица 72 – Регистр TIM_CLOCK

| Номер | 31…27 | 26 | 25 | 24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 00000000 | 00000000 |
| | - | TIM3 CLK EN | TIM2 CLK EN | TIM1 CLK EN | TIM3 BRG [7:0] | TIM2 BRG [7:0] | TIM1 BRG [7:0] |

Таблица 73 – Описание бит регистра TIM_CLOCK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...27 | - | Зарезервировано |
| 26 | TIM3 CLK EN | Разрешение тактовой частоты TIM3:<br>0 – нет частоты;<br>1 – есть частота |
| 25 | TIM2 CLK EN | Разрешение тактовой частоты TIM2:<br>0 – нет частоты;<br>1 – есть частота |
| 24 | TIM1 CLK EN | Разрешение тактовой частоты TIM1:<br>0 – нет частоты;<br>1 – есть частота |
| 23...16 | TIM3 BRG [7:0] | Делитель тактовой частоты TIM3:<br>xxxxx000 – TIM3_CLK == HCLK;<br>xxxxx001 – TIM3_CLK == HCLK/2;<br>xxxxx010 – TIM3_CLK == HCLK/4;<br>…<br>xxxxx111 – TIM3_CLK == HCLK/128 |
| 15…8 | TIM2 BRG [7:0] | Делитель тактовой частоты TIM2:<br>xxxxx000 – TIM2_CLK == HCLK;<br>xxxxx001 – TIM2_CLK == HCLK/2;<br>xxxxx010 – TIM2_CLK == HCLK/4;<br>…<br>xxxxx111 – TIM2_CLK == HCLK/128 |
| 7…0 | TIM1 BRG [7:0] | Делитель тактовой частоты TIM1:<br>xxxxx000 – TIM1_CLK == HCLK;<br>xxxxx001 – TIM1_CLK == HCLK/2;<br>xxxxx010 – TIM1_CLK == HCLK/4;<br>…<br>xxxxx111 – TIM1_CLK == HCLK/128 |

### 12.8.11 UART_CLOCK

Таблица 74 – Регистр UART_CLOCK

| Номер | 31…27 | 26 | 25 | 24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 00000000 | 00000000 |
| | - | TIM4 CLK EN | UART2 CLK EN | UART1 CLK EN | TIM4 BRG [7:0] | UART2 BRG [7:0] | UART1 BRG [7:0] |

Таблица 75 – Описание бит регистра UART_CLOCK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...27 | - | Зарезервировано |
| 26 | TIM4 CLK EN | Разрешение тактовой частоты TIM4:<br>0 – нет частоты;<br>1 – есть частота |
| 25 | UART2 CLK EN | Разрешение тактовой частоты UART2:<br>0 – нет частоты;<br>1 – есть частота |
| 24 | UART1 CLK EN | Разрешение тактовой частоты на UART1:<br>0 – нет частоты;<br>1 – есть частота |
| 23...16 | TIM4 BRG [7:0] | Делитель тактовой частоты TIM4:<br>xxxxx000 – TIM4_CLK == HCLK;<br>xxxxx001 – TIM4_CLK == HCLK/2;<br>xxxxx010 – TIM4_CLK == HCLK/4;<br>…<br>xxxxx111 – TIM4_CLK == HCLK/128 |
| 15…8 | UART2 BRG [7:0] | Делитель тактовой частоты UART2:<br>xxxxx000 – UART2_CLK == HCLK;<br>xxxxx001 – UART2_CLK == HCLK/2;<br>xxxxx010 – UART2_CLK == HCLK/4;<br>…<br>xxxxx111 – UART2_CLK == HCLK/128 |
| 7…0 | UART1 BRG [7:0] | Делитель тактовой частоты UART1:<br>xxxxx000 – UART1_CLK == HCLK;<br>xxxxx001 – UART1_CLK == HCLK/2;<br>xxxxx010 – UART1_CLK == HCLK/4;<br>…<br>xxxxx111 – UART1_CLK == HCLK/128 |

### 12.8.12 SSP_CLOCK

Таблица 76 – Регистр SSP_CLOCK

| Номер | 31…27 | 26 | 25 | 24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 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 77 – Описание бит регистра SSP_CLOCK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...27 | - | Зарезервировано |
| 26 | SSP3 CLK EN | Разрешение тактовой частоты SSP3:<br>0 – нет частоты;<br>1 – есть частота |
| 25 | SSP2 CLK EN | Разрешение тактовой частоты SSP2:<br>0 – нет частоты;<br>1 – есть частота |
| 24 | SSP1 CLK EN | Разрешение тактовой частоты SSP1:<br>0 – нет частоты;<br>1 – есть частота |
| 23...16 | SSP3 BRG [7:0] | Делитель тактовой частоты SSP3:<br>xxxxx000 – SSP3_CLK == HCLK;<br>xxxxx001 – SSP3_CLK == HCLK/2;<br>xxxxx010 – SSP3_CLK == HCLK/4;<br>…<br>xxxxx111 – SSP3_CLK == HCLK/128 |
| 15…8 | SSP2 BRG [7:0] | Делитель тактовой частоты SSP2:<br>xxxxx000 – SSP2_CLK == HCLK;<br>xxxxx001 – SSP2_CLK == HCLK/2;<br>xxxxx010 – SSP2_CLK == HCLK/4;<br>…<br>xxxxx111 – SSP2_CLK == HCLK/128 |
| 7…0 | SSP1 BRG [7:0] | Делитель тактовой частоты SSP1:<br>xxxxx000 – SSP1_CLK == HCLK;<br>xxxxx001 – SSP1_CLK == HCLK/2;<br>xxxxx010 – SSP1_CLK == HCLK/4;<br>…<br>xxxxx111 – SSP1_CLK == HCLK/128 |

### 12.8.13 ETH_CLOCK

Таблица 78 – Регистр ETH_CLOCK

| Номер | 31...30 | 29…28 | 27 | 26 | 25 | 24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | | R/W | | R/W | R/W | R/W | R/W |
| Сброс | 0 | 00 | 0 | 0 | 0 | 0 | 00000000 | 00000000 | 00000000 |
| | - | PHY CLK SEL | PHY CLK EN | SLEEP | MAN CLK EN | ETH CLK EN | PHY BRG [7:0] | MAN BRG [7:0] | ETH BRG [7:0] |

Таблица 79 – Описание бит регистра ETH_CLOCK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…30 | Зарезервировано | |
| 29...28 | PHY CLK SEL[1:0] | Биты выбора источника частоты для Ethernet PHY:<br>00 – не использовать;<br>01 – HSE;<br>10 – не использовать;<br>11 – HSE2.<br>Примечание – Частота тактирования блока PHY должна быть 25 МГц с джиттером менее 100 пс и длительностью фронтов менее 3 нс |
| 27 | PHY CLK EN | Разрешение тактовой частоты Ethernet PHY:<br>0 – нет частоты;<br>1 – есть частота |
| 26 | SLEEP | Перевод ядра контроллера в режим пониженного электропотребления:<br>0 – рабочий режим;<br>1 – режим пониженного электропотребления.<br>В режиме пониженного электропотребления тактовая частота поступает только на выбранные периферийные блоки, прерывание от которых возобновляет подачу тактовой частоты на ядро.<br>Для корректного перевода ядра в режим пониженного электропотребления необходимо после записи в бит SLEEP значения «1» выдержать паузу не менее двух тактов, добавив в код программы две инструкции «nop» или инструкцию барьерной синхронизации «DSB» или «DMB» |
| 25 | MAN CLK EN | Разрешение тактовой частоты на контроллер ГОСТ Р 52070-2003:<br>0 – нет частоты;<br>1 – есть частота |
| 24 | ETH CLK EN | Разрешение тактовой частоты на Ethernet MAC:<br>0 – нет частоты;<br>1 – есть частота |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 23...16 | PHY BRG [7:0] | Делитель тактовой частоты PHY:<br>xxxxx000 – PHY_CLK == PHY1_CLK;<br>xxxxx001 – PHY_CLK == PHY1_CLK/2;<br>xxxxx010 – PHY_CLK == PHY1_CLK/4;<br>xxxxx011 – PHY_CLK == PHY1_CLK/8;<br>…<br>xxxxx111 – PHY_CLK == PHY1_CLK/128 |
| 15…8 | MAN BRG [7:0] | Делитель тактовой частоты контроллера ГОСТ Р 52070-2003:<br>xxxxx000 – MAN_CLK == HCLK;<br>xxxxx001 – MAN_CLK == HCLK/2;<br>xxxxx010 – MAN_CLK == HCLK/4;<br>…<br>xxxxx111 – MAN_CLK == HCLK/128 |
| 7…0 | ETH BRG [7:0] | Всегда задавать 0 |

### 12.8.14 DMA_DONE_STICK

Таблица 80 – Регистр DMA_DONE_STICK

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

Таблица 81 – Описание бит регистра DMA_DONE_STICK

| Разряды | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...0 | DMA_DONE_STICK | Биты регистра устанавливаются в единицу при окончании работы соответствующего номера 0 – 31 канала DMA. Сбрасываются только записью нуля |

# 13 Батарейный домен и часы реального времени

Блок батарейного домена предназначен для обеспечения функций часов реального времени и сохранения некоторого набора пользовательских данных при отключении основного источника питания. При снижении питания Ucc в блоке SW происходит автоматическое переключение питания BDUcc с Ucc на BUcc. Если на BUcc имеется отдельный источник питания (батарейка), то батарейный домен остается включенным и может выполнять свои функции.

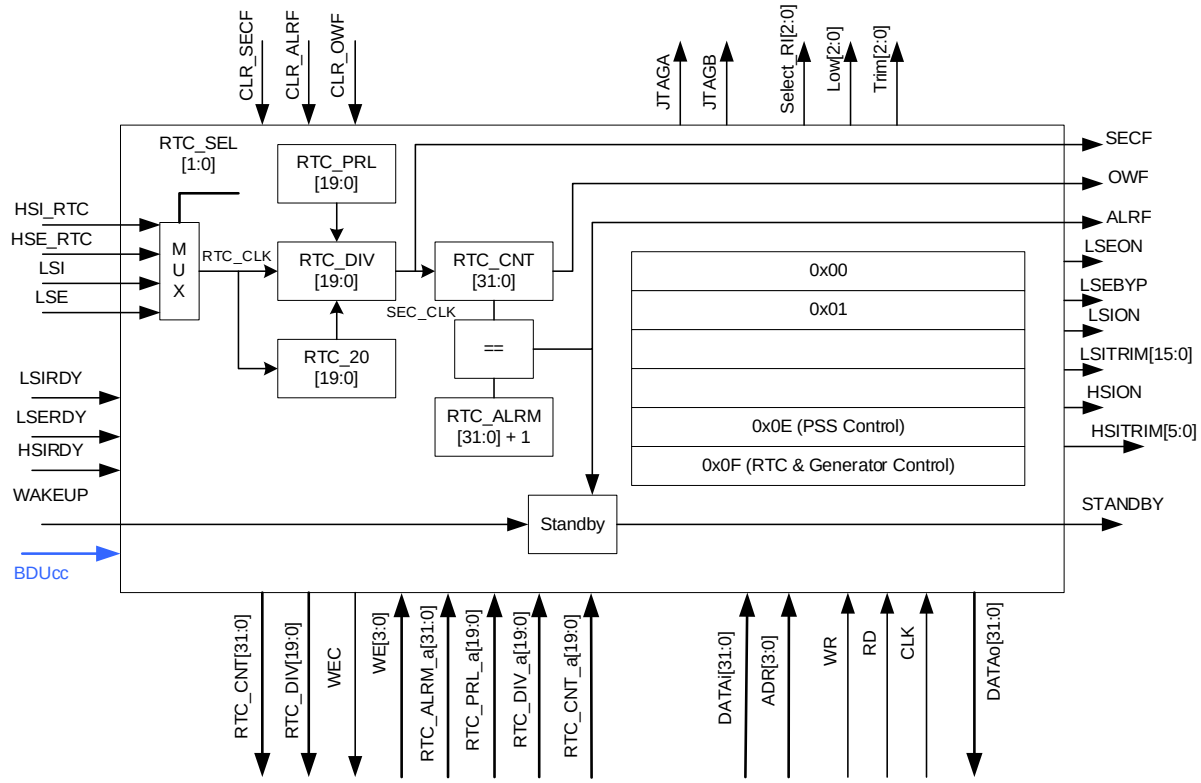



Рисунок 23 – Структурная блок-схема батарейного домена и часов реального времени

## 13.1 Часы реального времени

Часы реального времени позволяют организовать механизм отсчета времени в кристалле, в том числе при отключении основного источника питания. Включение часов реального времени осуществляется битом RTC_EN. В качестве источника тактовой частоты часов реального времени RTC_CLK может выступать генератор LSI или LSE, а также генератор HSI или HSE с дополнительным делителем до 256. Частоты HSI_RTC и HSE_RTC формируются в блоке управления тактовыми частотами и могут быть выбраны только при наличии питания $DU_{CC}$. Частота LSI может быть выбрана при наличии питания $U_{CC}$, LSE – при наличии $U_{CC}$ или $BU_{CC}$. Выбор между источниками осуществляется битами RTC_SEL. При возможном отключении основного источника питания Ucc в качестве источника тактовой частоты RTC_CLK должен использоваться осциллятор LSE, так как он также имеет питание $BDU_{CC}$. Биты управления осциллятором LSE расположены в батарейном домене и, таким образом, при отключении

основного питания они не сбрасываются. При этом при первоначальном включении эти биты так же не определены и могут принять любое значение.

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

Для калибровки тактовой частоты RTC_CLK используются биты RTC_CAL[7:0] (см. описание ошибки 0036 в документе К1986ВЕ1FI, К1986ВЕ1GI Errata Notice). Значение RTC_CAL[7:0] определяет, какое число тактов RTC_CLK из $2^{20}$ будет замаскировано с помощью дополнительного счетчика RTC_20. Данный способ калибровки заключается в том, что один период RTC_DIV, 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 RTC_PRL, а затем произвести замедление с помощью бит RTC_CAL[7:0]. Изменение значения бит RTC_CAL[7:0] может быть осуществлено в ходе работы часов реального времени.

Регистр RTC_CNT предназначен для отсчета времени в секундах и работает на выходной частоте предварительного делителя SEC_CLK. Регистр RTC_ALRM предназначен для задания времени, при превышении которого вырабатывается флаг прерывания и пробуждения процессора ALRF. Таким образом, бит STANBY, отключающий внутренний регулятор напряжения, автоматически сбрасывается при превышении RTC_CNT значения RTC_ALRM. Бит STANDBY так же может быть сброшен с помощью вывода WAKEUP.

Процесс выполнения записи регистров RTC_CNT, RTC_DIV, RTC_PRL и RTC_ALRM отображается с помощью флага WEC в регистре RTC_CS. На время выполнения записи флаг WEC устанавливается в «1», по окончании записи флаг WEC сбрасывается в «0». Пока флаг WEC установлен в «1», новая запись в регистры RTC, а также регистры BKP, запрещена.

## 13.2 Регистры аварийного сохранения

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

## 13.3 Описание регистров блока батарейного домена

Таблица 82 – Описание регистров блока батарейного домена

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x400D_8000 | BKP | Контроллер батарейного домена и часов реального времени |
| Смещение | | |
| 0x00 | BKP_REG_00 | Регистр аварийного сохранения 0 |
| … | | |
| 0x38 | BKP_REG_0E | Регистр аварийного сохранения 14 |
| 0x3C | BKP_REG_0F | Регистр аварийного сохранения 15 и управления блоками RTC, LSE, LSI и HSI |
| 0x40 | RTC_CNT | Регистр основного счетчика часов реального времени |
| 0x44 | RTC_DIV | Регистр счетчика предварительного делителя |
| 0x48 | RTC_PRL | Регистр коэффициента деления предварительного делителя |
| 0x4C | RTC_ALRM | Регистр значения для сравнения с основным счетчиком RTC_CNT и выработки сигнала ALRF |
| 0x50 | RTC_CS | Регистр управления и состояния флагов часов реального времени |

### 13.3.1 BKP_REG_[0D…00]

BKP_REG_00
BKP_REG_01
BKP_REG_02
BKP_REG_03
BKP_REG_04
BKP_REG_05
BKP_REG_06
BKP_REG_07
BKP_REG_08
BKP_REG_09
BKP_REG_0A
BKP_REG_0B
BKP_REG_0C
BKP_REG_0D

Таблица 83 – Регистры REG_[0D…00]

| Номер | 31…0 |
|---|---|
| Доступ | R/W |
| Сброс | U |
| | BKP REG[31:0] |

Таблица 84 – Описание бит регистров REG_[0D…00]

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…0 | BKP REG[31:0] | Регистр аварийного сохранения |

### 13.3.2 BKP_REG_0E

Таблица 85 – Регистр REG_0E

| Номер | 31…16 | 15 | 14 | 13…12 | 11 | 10…8 | 7 | 6 | 5…3 | 2…0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | U | R/W | R/S1 | R/W | U | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 00 | 0 | 000 | 0 | 0 | 000 | 000 |
| | - | ilimen | - | Trim[4:3] | FPOR | Trim[2:0] | - | Stand_ Alone | SelectRI [2:0] | LOW [2:0] |

Таблица 86 – Описание бит регистра REG_0E

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15 | ilimen | Бит разрешения защиты ограничения регулятора по току 150 мА:<br>1 – разрешено;<br>0 – запрещено |
| 14 | - | Зарезервировано |
| 13…12 | Trim[4:3] | Значение опорного напряжения регулятора:<br>00 – 1,8 В.<br>Использование других значений запрещено |
| 11 | FPOR | Флаг срабатывания POR.<br>Устанавливается в 1 загрузочным ПЗУ после сброса по питанию, при сбросе по питанию устанавливается в 0. Служит для анализа загрузочным ПЗУ, что сейчас идет выполнение программы после системного или программного сброса, либо после сброса по питанию |
| 10…8 | Trim[2:0] | Коэффициент настройки опорного напряжения встроенного регулятора напряжения $DU_{CC}$.<br>Устанавливать только значение LDO_TRIM[2:0], считанное из информационной памяти (см. таблицу 20) |
| 7 | - | Зарезервировано |
| 6 | Stand_Alone | 1 – выбор режима StandAlone;<br>0 – обычный режим работы |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 5…3 | SelectRI[2:0] | Выбор дополнительной нагрузки для регулятора 1,8 В:<br>000 – ~6 кОм (дополнительный ток потребления 300 мкА);<br>001 – ~270 кОм (дополнительный ток потребления 6,6 мкА);<br>010 – ~90 кОм (дополнительный ток потребления 20 мкА);<br>011 – ~24 кОм (дополнительный ток потребления 80 мкА);<br>100 – ~900 кОм (собственное потребление 2 мкА);<br>101 – ~2 кОм (дополнительный ток потребления 900 мкА);<br>110 – ~400 Ом (дополнительный ток потребления 4,4 мА);<br>111 – ~100 Ом (дополнительный ток потребления 19 мА) |
| 2…0 | LOW[2:0] | Выбор режима работы регулятора 1,8 В.<br>Значение LOW должно совпадать со значением SelectRI и выставляться в зависимости от тактовой частоты микросхемы:<br>000 – частота до 10 МГц;<br>001 – частота до 200 кГц;<br>010 – частота до 500 кГц;<br>011 – частота до 1 МГц;<br>100 – при выключении всех генераторов;<br>101 – частота до 40 МГц;<br>110 – частота до 80 МГц;<br>111 – частота до 144 МГц |

### 13.3.3 BKP_REG_0F

Таблица 87 – Регистр REG_0F

| Номер | 31 | 30 | 29…24 | 23 | 22 | 21 | 20…16 |
|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | RO | R/W | RO | R/W |
| Сброс | 0 | 0 | 100000 | 1 | 1 | 1 | 10000 |
| | RTC RESET | STANDBY | HSI TRIM[5:0] | HSI RDY | HSI ON | LSI RDY | LSI TRIM[4:0] |

| Номер | 15 | 14 | 13 | 12…5 | 4 | 3..2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | U | RO | R/W | R/W | R/W | R/W | R/W |
| Сброс | 1 | 0 | 0 | 0000000 | 0 | 00 | 0 | 0 |
| | LSI ON | - | LSE RDY | RTC CAL[7:0] | RTC EN | RTC SEL[1:0] | LSE BYP | LSE ON |

Таблица 88 – Описание бит регистра REG_0F

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31 | RTC RESET | Сброс часов реального времени:<br>0 – часы не сбрасываются;<br>1 – часы сбрасываются |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 30 | STANDBY | Режим отключения регулятора DUcc на 1,8 В:<br>0 – регулятор включен и выдает напряжение;<br>1 – выключение регулятора.<br>Триггер сбрасывается по событию ALRF или по низкому уровню на выводе WAKEUP.<br>Рекомендуется оставлять 0 |
| 29….24 | HSI TRIM[5:0] | Коэффициент подстройки частоты генератора HSI.<br>Устанавливать только значение HSI_TRIM[5:0], считанное из информационной памяти (см. таблицу 20).<br>Зависимость частоты HSI от значения HSI_TRIM см. на рисунке 144 |
| 23 | HSI RDY | Флаг выхода генератора HSI в рабочий режим:<br>0 – генератор не запущен или не вышел в режим;<br>1 – генератор работает в рабочем режиме |
| 22 | HSI ON | Бит управления генератором HSI:<br>0 – генератор выключен;<br>1 – генератор включен |
| 21 | LSI RDY | Флаг выхода генератора LSI в рабочий режим:<br>0 – генератор не запущен или не вышел в режим;<br>1 – генератор работает в рабочем режиме |
| 20…16 | LSI TRIM[4:0] | Коэффициент подстройки частоты генератора LSI.<br>Устанавливать только значение LSI_TRIM[4:0], считанное из информационной памяти (см. таблицу 20).<br>Зависимость частоты LSI от значения LSI_TRIM см. на рисунке 143 |
| 15 | LSI ON | Бит управления генератором LSI:<br>0 – генератор выключен;<br>1 – генератор включен |
| 14 | - | Зарезервировано |
| 13 | LSE RDY | Флаг выхода генератора LSE в рабочий режим:<br>0 – генератор не запущен или не вышел в режим;<br>1 – генератор работает в рабочем режиме |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 12…5 | RTC CAL[7:0] | Коэффициент подстройки тактовой частоты часов реального времени RTC_CLK. Из каждых $2^{20}$ тактов RTC_CLK будет замаскировано RTC_CAL тактов:<br>00000000 – 0 тактов;<br>00000001 – 1 такт;<br>….<br>11111111 – 255 тактов.<br>Таким образом, если исходная частота RTC_CLK равна 32768,000000 Гц, то после подстройки средняя частота тактирования составляет:<br>RTC_CLK_C = 32768,00000 Гц при RTC_CAL = 0;<br>RTC_CLK_C = 32767,96875 Гц при RTC_CAL = 1;<br>…<br>RTC_CLK_C = 32760,03125 Гц при RTC_CAL = 255.<br>В общем случае средняя тактовая частота RTC_CLK после подстройки составляет:<br>RTC_CLK_C = RTC_CLK · (1 – RTC_CAL[7:0] / $2^{20}$).<br>Единица подстройки (замедления) равна $1/2^{20}$, или ~0,95 ppm |
| 4 | RTC EN | Бит разрешения работы часов реального времени:<br>0 – работа запрещена;<br>1 – работа разрешена.<br>После записи задержка изменения бита составляет 1 такт частоты RTC_CLK |
| 3…2 | RTC SEL[1:0] | Биты выбора источника тактовой синхронизации часов реального времени RTC_CLK:<br>00 – LSI;<br>01 – LSE;<br>10 – HSI_RTC (формируется в блоке RST_CLK);<br>11 – HSE_RTC (формируется в блоке RST_CLK) |
| 1 | LSE BYP | Бит управления генератором LSE:<br>0 – режим осциллятора;<br>1 – режим работы на проход (внешний генератор). Устанавливать совместно с LSE_ON = 1 |
| 0 | LSE ON | Бит управления генератором LSE:<br>0 – генератор выключен;<br>1 – генератор включен |

### 13.3.4 RTC_CNT

Таблица 89 – Регистр RTC_CNT

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

Таблица 90 – Описание бит регистра RTC_CNT

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

### 13.3.5 RTC_DIV

Таблица 91 – Регистр RTC_DIV

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

Таблица 92 – Описание бит регистра RTC_DIV

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

### 13.3.6 RTC_PRL

Таблица 93 – Регистр RTC_PRL

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

Таблица 94 – Описание бит регистра RTC_PRL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…20 | - | Зарезервировано |
| 19…0 | RTC_PRL[19:0] | Коэффициент деления тактовой частоты RTC_CLK:<br>SEC_CLK = RTC_CLK / (RTC_PRL + 1),<br>где RTC_CLK – частота тактирования с учетом подстройки RTC_CAL[7:0] |

### 13.3.7 RTC_ALRM

Таблица 95 – Регистр RTC_ALRM

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

Таблица 96 – Описание бит регистра RTC_ALRM

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…0 | RTC_ALRM[31:0] | Значение для сравнения с основным счетчиком RTC_CNT и выработки сигнала ALRF.<br>Сигнал ALRF вырабатывается в момент превышения основным счетчиком RTC_CNT значения в регистре RTC_ALRM: RTC_CNT[31:0] == RTC_ALRM[31:0] + 1 |

### 13.3.8 RTC_CS

Таблица 97 – Регистр RTC_CS

| Номер | 31…7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | RO | R/W | R/W | R/W | R/C1 | R/C1 | R/C1 |
| Сброс | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | WEC | ALRF_IE | SECF_IE | OWF_IE | ALRF | SECF | OWF |

Таблица 98 – Описание бит регистра RTC_CS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…7 | - | Зарезервировано |
| 6 | WEC | Флаг выполнения записи в регистры RTC (RTC_CNT, RTC_DIV, RTC_PRL, RTC_ALRM):<br>0 – нет записи в регистр RTC;<br>1 – идет запись в регистр RTC, новая запись в регистры RTC и BKP запрещена |
| 5 | ALRF_IE | Флаг разрешения прерывания по событию ALRF:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено |
| 4 | SECF_IE | Флаг разрешения прерывания по событию SECF:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено |
| 3 | OWF_IE | Флаг разрешения прерывания по событию OWF<br>0 – прерывание запрещено;<br>1 – прерывание разрешено |
| 2 | ALRF | Флаг превышения основным счетчиком RTC_CNT значения RTC_ALRM:<br>0 – нет события превышения;<br>1 – было событие превышения.<br>Флаг устанавливается при RTC_CNT[31:0] == RTC_ALRM[31:0] + 1.<br>Сброс флага осуществляется записью 1 |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 1 | SECF | Флаг совпадения значений регистров RTC_DIV и RTC_PRL:<br>0 – не было совпадения;<br>1 – было совпадение<br>По событиям совпадения RTC_DIV и RTC_PRL формируется частота SEC_CLK. Таким образом флаг SECF устанавливается на каждом такте частоты SEC_CLK.<br>Сброс флага осуществляется записью 1 |
| 0 | OWF | Флаг переполнения основного счетчика RTC_CNT:<br>0 – нет переполнения;<br>1 – было переполнение.<br>Сброс флага осуществляется записью 1 |

# 14 Порты ввода/вывода

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

Для использования порта ввода/вывода необходимо разрешить тактирование этого порта в регистре PER_CLOCK (см. раздел 12 «Сигналы тактовой частоты»).

Таблица 99 – Порты ввода/вывода

| Вывод | Аналоговая функция | | Цифровая функция | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Порт IO | Основная | | Альтернативная | | Переопределенная | |
| | ANALOG_EN=0 | | MODE[1:0]=00 ANALOG_EN=1 | MODE[1:0]=01 ANALOG_EN=1 | | MODE[1:0]=10 ANALOG_EN=1 | | MODE[1:0]=11 ANALOG_EN=1 | |
| Порт A | | | | | | | | | |
| PA0 | - | | PA0 | D0 | 1 | EXTINT1 | 8 | ETR1 | 3 |
| PA1 | - | | PA1 | D1 | | EXTINT2 | | ETR2 | 13 |
| PA2 | - | | PA2 | D2 | | EXTINT3 | | ETR3 | 10 |
| PA3 | - | | PA3 | D3 | | EXTINT4 | | BRK1 | 3 |
| PA4 | - | | PA4 | D4 | | BRK2 | 13 | FRX | 15 |
| PA5 | - | | PA5 | D5 | | BRK3 | 10 | FSD | |
| PA6 | - | | PA6 | D6 | | TMR4_CH1 | 16 | FXEN | |
| PA7 | - | | PA7 | D7 | | TMR4_CH1N | | FTX | |
| PA8 | - | | PA8 | D8 | | TMR4_CH2 | | PRMC+ | 4 |
| PA9 | - | | PA9 | D9 | | TMR4_CH2N | | PRMC- | |
| PA10 | - | | PA10 | D10 | | TMR4_CH3 | | PRMD+ | |
| PA11 | - | | PA11 | D11 | | TMR4_CH3N | | PRMD- | |
| PA12 | - | | PA12 | D12 | | TMR4_CH4 | | PRDC+ | |
| PA13 | - | | PA13 | D13 | | TMR4_CH4N | | PRDC- | |
| PA14 | - | | PA14 | D14 | | BRK4 | | PRDD+ | |
| PA15 | - | | PA15 | D15 | | ETR4 | | PRDD- | |
| Порт B | | | | | | | | | |
| PB0 | - | | PB0 | D16 | 1 | IN1+ | 2 | TMR3_CH1 | 10 |
| PB1 | - | | PB1 | D17 | | IN1- | | TMR3_CH1N | |
| PB2 | - | | PB2 | D18 | | IN2+ | | TMR3_CH2 | |
| PB3 | - | | PB3 | D19 | | IN2- | | TMR3_CH2N | |
| PB4 | - | | PB4 | D20 | | IN3+ | | TMR3_CH3 | |
| PB5 | - | | PB5 | D21 | | IN3- | | TMR3_CH3N | |
| PB6 | - | | PB6 | D22 | | IN4+ | | TMR3_CH4 | |
| PB7 | - | | PB7 | D23 | | IN4- | | TMR3_CH4N | |
| PB8 | - | | PB8 | D24 | | IN5+ | | TMR1_CH1N | 3 |
| PB9 | - | | PB9 | D25 | | IN5- | | TMR2_CH1N | 13 |
| PB10 | - | | PB10 | D26 | | IN6+ | | TMR1_CH2N | 3 |

| Вывод | Аналоговая функция<br>ANALOG_EN=0 | | Цифровая функция<br>Порт IO<br>MODE[1:0]=00<br>ANALOG_EN=1 | Основная<br>MODE[1:0]=01<br>ANALOG_EN=1 | | Альтернативная<br>MODE[1:0]=10<br>ANALOG_EN=1 | | Переопределенная<br>MODE[1:0]=11<br>ANALOG_EN=1 | |
|---|---|---|---|---|---|---|---|---|---|
| PB11 | - | | PB11 | D27 | | IN6- | | TMR2_CH2N | 13 |
| PB12 | - | | PB12 | D28 | | IN7+ | | TMR1_CH3N | 3 |
| PB13 | - | | PB13 | D29 | | IN7- | | TMR2_CH3N | 13 |
| PB14 | - | | PB14 | D30 | | IN8+ | | TMR1_CH4N | 3 |
| PB15 | - | | PB15 | D31 | | IN8- | | TMR2_CH4N | 13 |
| Порт C | | | | | | | | | |
| PC0 | - | | PC0 | nWR | 1 | ETR1 | 3 | BRK1 | 3 |
| PC1 | - | | PC1 | nRD | | ETR2 | 13 | BRK2 | 13 |
| PC2 | - | | PC2 | ALE | | CLKO | 1 | BRK3 | 10 |
| PC3 | - | | PC3 | UART_TXD1 | 9 | CLE | | SIROUT1 | 9 |
| PC4 | - | | PC4 | UART_RXD1 | | BUSY | | SIRIN1 | |
| PC5 | - | | PC5 | EXTINT1 | 8 | SSP1_TXD | 14 | SSP1_RXD | 14 |
| PC6 | - | | PC6 | EXTINT2 | | SSP1_RXD | | SSP1_TXD | |
| PC7 | - | | PC7 | EXTINT3 | | SSP1_SCK | | FXEN | 15 |
| PC8 | - | | PC8 | EXTINT4 | | SSP1_FSS | | FTX | |
| PC9 | - | | PC9 | SSP2_TXD | 11 | BE0 | 1 | CAN_RX1 | 17 |
| PC10 | - | | PC10 | SSP2_RXD | | BE1 | | CAN_TX1 | |
| PC11 | - | | PC11 | SSP2_SCK | | BE2 | | CAN_RX2 | 18 |
| PC12 | - | | PC12 | SSP2_FSS | | BE3 | | CAN_TX2 | |
| PC13 | - | | PC13 | PRMA+ | 4 | A30 | 1 | UART_TXD2 | 12 |
| PC14 | - | | PC14 | PRMA- | | A31 | | UART_RXD2 | |
| PC15 | - | | PC15 | PRMB+ | | BUSY | | TMR2_CH1 | 13 |
| Порт D | | | | | | | | | |
| PD0 | - | | PD0 | PRMB- | 4 | ALE | 1 | A16 | 1 |
| PD1 | - | | PD1 | PRDA+ | | CLE | | A15 | |
| PD2 | - | | PD2 | PRDA- | | SSP1_TXD | 14 | A14 | |
| PD3 | - | | PD3 | PRDB+ | | SSP1_RXD | | A13 | |
| PD4 | - | | PD4 | PRDB- | | SSP1_SCK | | A7 | |
| PD5 | - | | PD5 | PRD_PRMA | | SSP1_FSS | | A6 | |
| PD6 | - | | PD6 | PRD_PRMB | | nUART2RI | 12 | A5 | |
| PD7 | ADC0_REF+ | 5 | PD7 | SSP2_TXD | 11 | nUART2DCD | | A4 | |
| PD8 | ADC1_REF- | | PD8 | SSP2_RXD | | nUART2DTR | | A3 | |
| PD9 | ADC2 | | PD9 | SSP2_SCK | | nUART2DSR | | A2 | |
| PD10 | ADC3 | | PD10 | SSP2_FSS | | nUART2RTS | | A1 | |
| PD11 | ADC4 | | PD11 | A0 | 1 | nUART2CTS | | FRX | 15 |
| PD12 | ADC5 | | PD12 | SSP3_TXD | 19 | ETR3 | 10 | SSP3_RXD | 19 |
| PD13 | ADC6 | | PD13 | UART_TXD2 | 12 | OUT1+ | 2 | SIROUT2 | 12 |
| PD14 | ADC7 | | PD14 | UART_RXD2 | | OUT1- | | SIRIN2 | |
| PD15 | REFD0 | 6 | PD15 | OUT3+ | 2 | A13 | 1 | FSD | 15 |

| Вывод | Аналоговая функция<br><br>ANALOG_EN=0 | | Цифровая функция | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Порт IO<br><br>MODE[1:0]=00<br>ANALOG_EN=1 | Основная<br><br>MODE[1:0]=01<br>ANALOG_EN=1 | | Альтернативная<br><br>MODE[1:0]=10<br>ANALOG_EN=1 | | Переопределенная<br><br>MODE[1:0]=11<br>ANALOG_EN=1 | |
| Порт E | | | | | | | | | |
| PE0 | REFD1 | 6 | PE0 | OUT4+ | 2 | A14 | 1 | MDC | 15 |
| PE1 | DAC0 | | PE1 | OUT3- | | A15 | | nUART2RI | 12 |
| PE2 | DAC1 | | PE2 | OUT4- | | A16 | | MDIO | 15 |
| PE3 | - | | PE3 | TMR1_CH1 | 3 | A17 | | TXD[0] | |
| PE4 | - | | PE4 | TMR1_CH2 | | A18 | | TXD[1] | |
| PE5 | - | | PE5 | TMR1_CH3 | | A19 | | TXD[2] | |
| PE6 | OSC_IN32 | 7 | PE6 | TMR1_CH4 | | A20 | | TXD[3] | |
| PE7 | OSC_OUT32 | | PE7 | TMR2_CH1 | 13 | A21 | | RXD[0] | |
| PE8 | - | | PE8 | TMR2_CH2 | | A22 | | RXD[1] | |
| PE9 | - | | PE9 | TMR2_CH3 | | A23 | | RXD[2] | |
| PE10 | - | | PE10 | TMR2_CH4 | | A24 | | RXD[3] | |
| PE11 | - | | PE11 | CAN_RX1 | 17 | A25 | 1 | TXEN | 15 |
| PE12 | - | | PE12 | CAN_TX1 | | A26 | | TXER | |
| PE13 | - | | PE13 | CAN_RX2 | 18 | A27 | | TXCLK | |
| PE14 | - | | PE14 | CAN_TX2 | | A28 | | RXCLK | |
| PE15 | - | | PE15 | PRD_PRMD | 4 | A29 | | RXDV | |
| Порт F | | | | | | | | | |
| PF0 | OSC_IN25 | | PF0 | PRD_PRMA | 4 | READY | 1 | RXER | 15 |
| PF1 | OSC_OUT25 | | PF1 | PRD_PRMB | | A30 | | CRS | |
| PF2 | - | | PF2 | READY/<br>PRD_PRMC | 1/4 | A31 | | COL | |
| PF3 | - | | PF3 | PRMC+ | 4 | A0 | | TMR1_CH1 | 3 |
| PF4 | - | | PF4 | PRMC- | | A1 | | TMR1_CH2 | |
| PF5 | - | | PF5 | PRMD+ | | A2 | | TMR1_CH3 | |
| PF6 | - | | PF6 | PRMD- | | A3 | | TMR1_CH4 | |
| PF7 | - | | PF7 | PRDC+ | | A4 | | OUT4+ | 2 |
| PF8 | - | | PF8 | PRDC- | | A5 | | OUT4- | |
| PF9 | - | | PF9 | PRDD+ | | A6 | | OUT3+ | |
| PF10 | - | | PF10 | PRDD- | | A7 | | OUT3- | |
| PF11 | - | | PF11 | PRD_PRMC | | A8 | | OUT2+ | |
| PF12 | - | | PF12 | PRD_PRMD | | A9 | | OUT2- | |
| PF13 | - | | PF13 | OUT2+ | 2 | A10 | | SSP3_FSS | 19 |
| PF14 | - | | PF14 | OUT2- | | A11 | | SSP3_SCK | |
| PF15 | - | | PF15 | SSP3_RXD | 19 | A12 | | SSP3_TXD | |

Примечания

1 – выводы управляются системной шиной EXT_BUS;

2 – выводы управляются контроллером интерфейса по ГОСТ 18977-79;

3 – выводы управляются Таймером 1;

4 – выводы управляются контроллером интерфейса по ГОСТ Р 52070-2003;

| Вывод | Аналоговая функция<br><br>ANALOG_EN=0 | Цифровая функция | | | |
|---|---|---|---|---|---|
| | | Порт IO<br><br>MODE[1:0]=00<br>ANALOG_EN=1 | Основная<br><br>MODE[1:0]=01<br>ANALOG_EN=1 | Альтернативная<br><br>MODE[1:0]=10<br>ANALOG_EN=1 | Переопределенная<br><br>MODE[1:0]=11<br>ANALOG_EN=1 |
| 5 – выводы используются АЦП;<br>6 – выводы используются ЦАП;<br>7 – выводы используются генератором LSE;<br>8 – выводы используются контроллером прерываний;<br>9 – выводы управляются контроллером интерфейса UART1;<br>10 – выводы управляются Таймером 3;<br>11 – выводы управляются контроллером интерфейса SSP2;<br>12 – выводы управляются контроллером интерфейса UART2;<br>13 – выводы управляются Таймером 2;<br>14 – выводы управляются контроллером интерфейса SSP1;<br>15 – выводы управляются контроллером интерфейса Ethernet 10/100;<br>16 – выводы управляются Таймером 4;<br>17 – выводы управляются контроллером интерфейса CAN1;<br>18 – выводы управляются контроллером интерфейса CAN2;<br>19 – выводы управляются контроллером интерфейса SSP3 | | | | | |

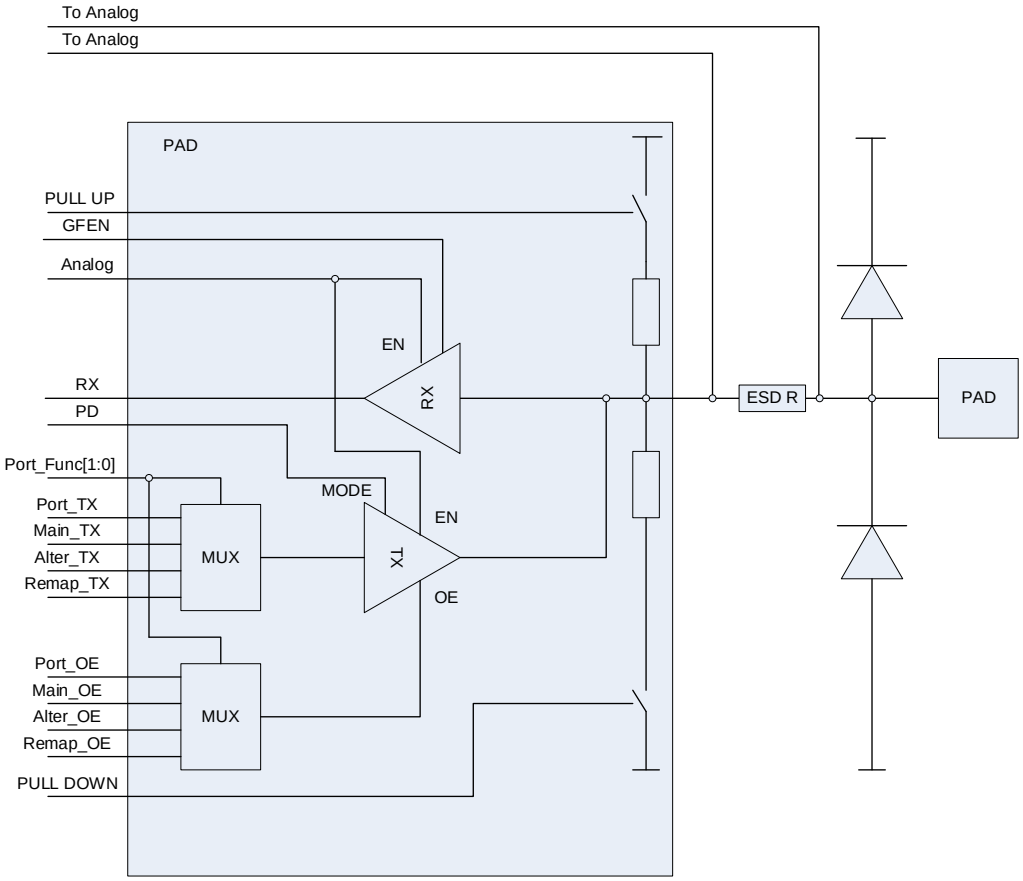



Рисунок 24 – Порты ввода/вывода

## 14.1 Описание регистров портов ввода/вывода

Таблица 100 – Описание регистров портов ввода-вывода

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x400A_8000 | GPIO1 | Порт A |
| 0x400B_0000 | GPIO2 | Порт B |
| 0x400B_8000 | GPIO3 | Порт C |
| 0x400C_0000 | GPIO4 | Порт D |
| 0x400C_8000 | GPIO5 | Порт E |
| 0x400E_8000 | GPIO6 | Порт F |
| Смещение | | |
| 0x00 | PORT_RXTX[15:0] | Данные порта |
| 0x04 | PORT_OE[15:0] | Направление порта |
| 0x08 | PORT_FUNC[31:0] | Режим работы порта |
| 0x0C | PORT_ANALOG[15:0] | Режим работы выводов порта (аналоговый/цифровой) |
| 0x10 | PORT_PULL[31:0] | Подтяжка порта |
| 0x14 | PORT_PD[31:0] | Режим работы выходного драйвера |
| 0x18 | PORT_PWR[31:0] | Режим мощности передатчика |
| 0x1C | PORT_GFEN[15:0] | Режим работы входного фильтра |
| 0x20 | PORT_SETTX[15:0] | Регистр SET_TX записью 1 устанавливает «1» в регистре PORT_RXTX |
| 0x24 | PORT_CLRTX[15:0] | Регистр CLR_TX записью 1 устанавливает «0» в регистре RXTX |
| 0x28 | PORT_RDTX | Регистр позволяет читать то, что записано в выходной регистр порта |

### 14.1.1 PORTx_RXTX

Таблица 101 – Регистр RXTX

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

Таблица 102 – Описание бит регистра RXTX

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…0 | PORT RXTX[15:0] | Режим работы вывода порта.<br>Данные для выдачи на выводы порта и для чтения.<br>Биты читаются как 1, если соответствующие им выводы настроены в аналоговый режим работы |

### 14.1.2 PORTx_OE

Таблица 103 – Регистр OE

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

Таблица 104 – Описание бит регистра OE

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

### 14.1.3 PORTx_FUNC

Таблица 105 – Регистр FUNC

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

Таблица 106 – Описание бит регистра FUNC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…2 | MODEx | Аналогично MODE0 для остальных разрядов порта |
| 1…0 | MODE0[1:0] | Режим работы вывода порта:<br>00 – порт;<br>01 – основная функция;<br>10 – альтернативная функция;<br>11 – переопределенная функция |

### 14.1.4 PORTx_ANALOG

Таблица 107 – Регистр ANALOG

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

Таблица 108 – Описание бит регистра ANALOG

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…0 | ANALOG EN[15:0] | Режим работы вывода порта:<br>0 – аналоговый;<br>1 – цифровой |

### 14.1.5 PORTx_PULL

Таблица 109 – Регистр PULL

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

Таблица 110 – Описание бит регистра PULL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | PULL UP[15:0] | Режим работы вывода порта.<br>Разрешение подтяжки вверх:<br>0 – подтяжка к питанию выключена;<br>1 – подтяжка к питанию включена (есть подтяжка) |
| 15…0 | PULL DOWN[15:0] | Режим работы вывода порта.<br>Разрешение подтяжки вниз:<br>0 – подтяжка к нулю выключена;<br>1 – подтяжка к нулю включена (есть подтяжка) |

### 14.1.6 PORTx_PD

Таблица 111 – Регистр PD

| Номер | 31…16 | 15…0 |
|---|---|---|
| Доступ | R/W | R/W |
| Сброс | 0 | 0 |
| | PORT SHM[15:0] | PORT PD[15:0] |

Таблица 112 – Описание бит регистра PD

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | PORT SHM[15:0] | Режим работы триггера Шмитта на входе:<br>0 – гистерезис 200 мВ;<br>1 – гистерезис 400 мВ |
| 15…0 | PORT PD[15:0] | Режим работы вывода порта.<br>Режим работы выхода:<br>0 – управляемый драйвер;<br>1 – открытый сток |

### 14.1.7 PORTx_PWR

Таблица 113 – Регистр PWR

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

Таблица 114 – Описание бит регистра PWR

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

### 14.1.8 PORTx_GFEN

Таблица 115 – Регистр GFEN

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

Таблица 116 – Описание бит регистра GFEN

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

### 14.1.9 PORTx_SETTX

Таблица 117 – Регистр SETTX

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

Таблица 118 – Описание бит регистра SETTX

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…0 | SETTX[15:0] | Регистр индивидуальной установки выхода порта:<br>1 – устанавливает в единицу соответствующий разряд выхода порта PORTx_RXTX;<br>0 – не влияет на состояние соответствующего выхода порта PORTx_RXTX.<br>Читается ранее записанное в регистр значение, а не состояние соответствующих входов порта PORT_RXTX |

### 14.1.10 PORTx_CLRTX

Таблица 119 – Регистр CLRTX

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

Таблица 120 – Описание бит регистра CLRTX

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…0 | CLRTX[15:0] | Регистр индивидуального сброса выхода порта:<br>1 – сбрасывает в ноль соответствующий разряд выхода порта PORTx_RXTX;<br>0 – не влияет на состояние соответствующего выхода порта PORTx_RXTX.<br>Читается ранее записанное в регистр значение, а не состояние соответствующих входов порта PORTx_RXTX |

# 15 Детектор напряжения питания

Блок детектора напряжения питания PVD предназначен для контроля питания $U_{CC}$ и $BU_{CC}$ при работе микросхемы. Блок PVD позволяет сравнивать внешние уровни напряжения с внутренними опорными уровнями и в случае превышения или снижения ниже опорного уровня выработать сигнал или прерывание для программной обработки.

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

При использовании блока PVD необходима его калибровка в условиях эксплуатации конечного изделия и в расширенном диапазоне напряжения питания $U_{CC}$, так как представленные в таблице 121 типовые значения уровней срабатывания PVD носят справочный характер и варьируются из-за технологического разброса.

Таблица 121 – Типовые уровни напряжений детектора питания

| Наименование параметра, единица измерения | Норма параметра | | |
|---|---|---|---|
| | не менее | типовое | не более |
| Входное напряжение, $U_{CC}$, В | 2,0 | – | 3,6 |
| Входное напряжение, $BU_{CC}$, В | 1,8 | – | 3,6 |
| Уровень срабатывания PVD от $U_{CC}$, при PLS = «000», В | – | 2,0 | – |
| Уровень срабатывания PVD от $U_{CC}$, при PLS = «001», В | – | 2,2 | – |
| Уровень срабатывания PVD от $U_{CC}$, при PLS = «010», В | – | 2,4 | – |
| Уровень срабатывания PVD от $U_{CC}$, при PLS = «011», В | – | 2,6 | – |
| Уровень срабатывания PVD от $U_{CC}$, при PLS = «100», В | – | 2,8 | – |
| Уровень срабатывания PVD от $U_{CC}$, при PLS = «101», В | – | 3,0 | – |
| Уровень срабатывания PVD от $U_{CC}$, при PLS = «110», В | – | 3,2 | – |
| Уровень срабатывания PVD от $U_{CC}$, при PLS = «111», В | – | 3,4 | – |
| Уровень срабатывания PVBD от $BU_{CC}$, при PBLS = «00», В | – | 1,8 | – |
| Уровень срабатывания PVBD от $BU_{CC}$, при PBLS = «01», В | – | 2,2 | – |
| Уровень срабатывания PVBD от $BU_{CC}$, при PBLS = «10», В | – | 2,6 | – |
| Уровень срабатывания PVBD от $BU_{CC}$, при PBLS = «11», В | – | 3,0 | – |

Таблица 122 – Описание регистров блока PVD

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

## 15.1 PVDCS

Таблица 123 – Регистр PVDCS

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

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

Таблица 124 – Описание бит регистра PVDCS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…13 | - | Зарезервировано |
| 12 | PVDBEN | Бит разрешения работы блока датчика напряжения питания $BU_{CC}$:<br>0 – датчик отключен;<br>1 – датчик включен |
| 11 | INV | Флаг инверсии выхода от датчика PVD:<br>0 – нет инверсии;<br>1 – есть инверсия.<br>Если флаг не инвертируется, то флаг выставляется при превышении заданного уровня, если инвертируется, то при снижении ниже заданного уровня |
| 10 | INVB | Флаг инверсии выхода от датчика PVBD:<br>0 – нет инверсии;<br>1 – есть инверсия.<br>Если флаг не инвертируется, то флаг выставляется при превышении заданного уровня, если инвертируется, то при снижении ниже заданного уровня |
| 9 | IEPVD | Флаг разрешения прерывания от датчика PVD:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено.<br>Очищается записью 0, если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен |
| 8 | IEPVBD | Флаг разрешения прерывания от датчика PVBD:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено.<br>Очищается записью 0, если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 7 | PVD | Результат сравнения напряжения основного питания:<br>0 – напряжение питания меньше чем уровень, задаваемый PLS;<br>1 – напряжение питания больше чем уровень, задаваемый PLS.<br>Очищается программно записью 0, если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен.<br>Примечание – Сброс флага необходимо проводить с подтверждением – сбрасывать дважды |
| 6 | PVBD | Результат сравнения напряжения батарейного питания:<br>0 – напряжение питания меньше чем уровень, задаваемый PBLS;<br>1 – напряжение питания больше чем уровень, задаваемый PBLS.<br>Очищается программно записью 0, если при очистке датчик продолжает выдавать сигнал, то флаг не будет очищен.<br>Примечание – Сброс флага необходимо проводить с подтверждением – сбрасывать дважды |
| 5…3 | PLS[2:0] | Уровень напряжения для сравнения с напряжением основного питания:<br>000 – 2,0 В;<br>001 – 2,2 В;<br>010 – 2,4 В;<br>011 – 2,6 В;<br>100 – 2,8 В;<br>101 – 3,0 В;<br>110 – 3,2 В;<br>111 – 3,4 В |
| 2…1 | PBLS[1:0] | Уровень напряжения для сравнения с напряжением батарейного питания:<br>00 – 1,8 В;<br>01 – 2,2 В;<br>10 – 2,6 В;<br>11 – 3,0 В |
| 0 | PVDEN | Бит разрешения работы блока датчика напряжения питания Ucc:<br>0 – датчик отключен;<br>1 – датчик включен |

# 16 Внешняя системная шина

Внешняя системная шина позволяет работать с внешними микросхемами памяти и периферийными устройствами. В таблице 125 приведены области адресного пространства микросхемы, предназначенные для работы с внешней системной шиной.

Таблица 125 – Адресные диапазоны внешней системной шины

| Адресный диапазон | Размер | Описание |
|---|---|---|
| 0x0010_0000 – 0x1FFF_FFFF ITCMLAEN=1<br>0x0000_0000 – 0x1FFF_FFFF ITCMLAEN=0 | 256 Мбайт | Область памяти секции Code отображаемая на внешнюю системную шину с доступом через шину AHB-Lite |
| 0x5000_0000 – 0xDFFF_FFFF | 2 Гбайт | Область памяти секции Peripheral и External SRAM отображаемая на внешнюю системную шину с доступом через шину AHB-Lite. К этой области имеет доступ контроллер DMA |

Контроллер внешней системной шины во всех режимах не формирует сигналов выборки чипа CE. При работе с внешними статическими ОЗУ, ПЗУ и периферийными устройствами в качестве сигнала выборки чипа можно использовать старшие линии шины адреса, не используемые для непосредственной адресации, либо использовать программно управляемые выводы портов для формирования сигналов CE.

Максимальная частота переключения выводов внешней системной шины составляет не более 50 МГц.

## 16.1 Работа с внешними статическими ОЗУ, ПЗУ и периферийными устройствами

Для работы контроллера внешней системной шины с внешними микросхемами статического ОЗУ, ПЗУ или внешними периферийными устройствами необходимо задать режим работы через регистр CONTROL и/или RAM_CYCLESx. Бит RAM разрешает работу с внешними ОЗУ, бит ROM разрешает только чтение внешних ОЗУ или ПЗУ. В зависимость от скорости работы ядра микросхемы и внешних устройств необходимо задать времена транзакции на внешней системной шине через биты WAIT_STATE[3:0]. После этого все обращения в область памяти отображаемой на внешнюю системную шину будут транслироваться на выводы внешней системной шины A, D и сигналы управления nRD, nWR, BE[3:0] и сигнал синхронизации CLKO.

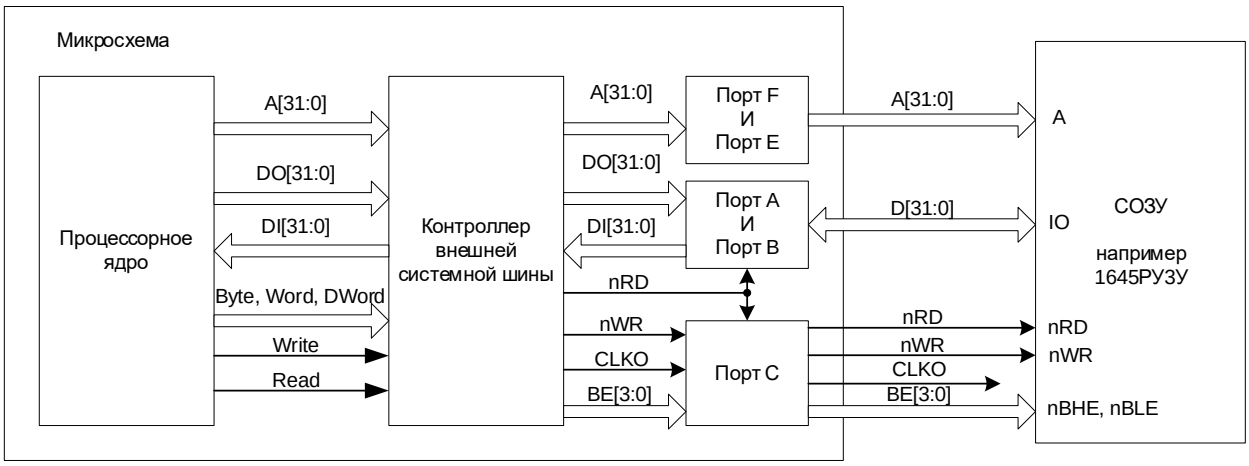



Рисунок 25 – Обмен по внешней системной шине

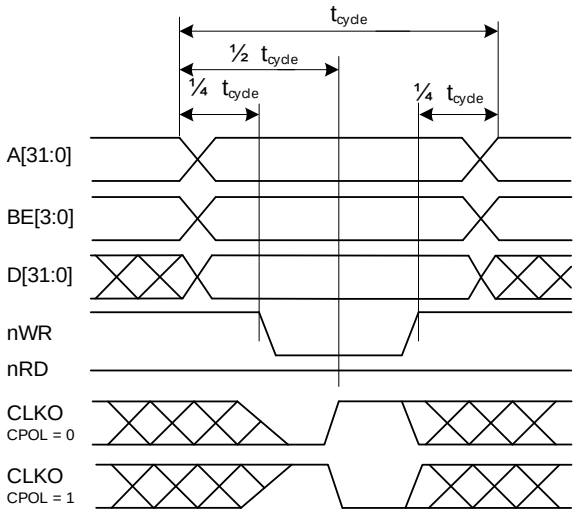



Рисунок 26 – Временная диаграмма записи

Время цикла записи tcycle задается битами WAIT_STATE[3:0]. Активный уровень сигналов nWR, nRD, BE[3:0] низкий. Если сигнал CLKO не требуется, он может не использоваться.

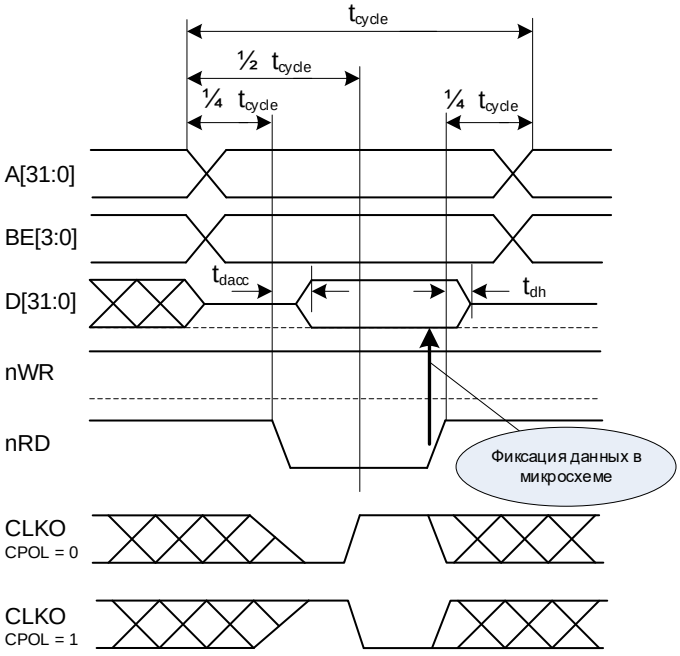



Рисунок 27 – Временная диаграмма чтения

При чтении по внешней системной шине необходимо выбрать такую длительность времени $t_{cycle}$, чтобы выполнялось время скорости доступа к памяти. Время $t_{dh}$ для микросхемы равно нулю.

Для уменьшения задержек в работе контроллера внешней системной шины направление шины данных вне цикла чтения/записи (неактивное состояние) зависит от последнего обращения:

– если выполнялась запись, то после цикла записи шина данных продолжает работать на выход, значение D[31:0] неопределено;

– если выполнялось чтение, то после цикла чтения шина данных продолжает работать на вход.

Шина адреса вне цикла чтения/записи работает на выход, значение A[31:0] = 0xFFFF_FFFF.

Таблица 126 – Длительность фаз обращения в тактах процессора при ENABLE_TUNE = 0 для соответствующего региона

| WAIT_STATE | Длительность nWR или nRD | Предустановка адреса и данных перед сигналом nWR или nRD | Удержание адреса после сигнала nWR или nRD (не менее) |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 1 | 1 |
| 3 | 2 | 1 | 1 |
| 4 | 2 | 2 | 1 |
| 5 | 3 | 2 | 1 |
| 6 | 3 | 2 | 2 |
| 7 | 4 | 2 | 2 |
| 8 | 4 | 3 | 2 |
| 9 | 5 | 3 | 2 |
| 10 | 5 | 3 | 3 |
| 11 | 6 | 3 | 3 |
| 12 | 6 | 4 | 3 |
| 13 | 7 | 4 | 3 |
| 14 | 7 | 4 | 4 |
| 15 | 8 | 4 | 4 |

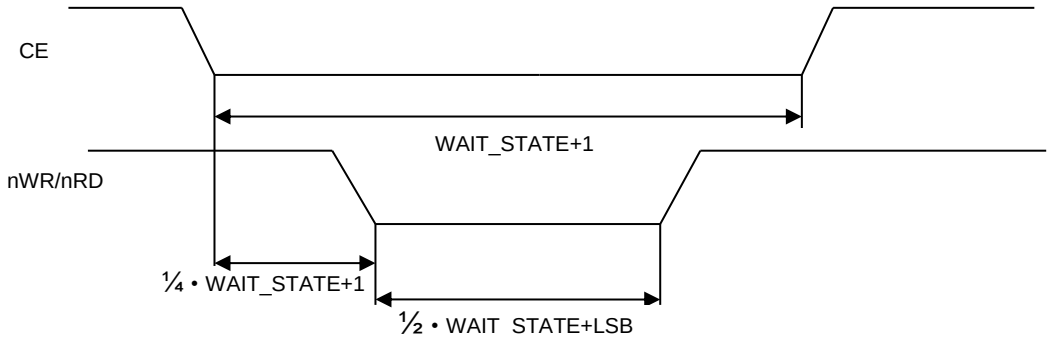



Рисунок 28 – Длительность фаз обращения в тактах процессора при ENABLE_TUNE = 0 для соответствующего региона

### 16.1.1 Обмен по внешней системной шине при задании длительности через WS_ACTIVE, WS_SETUP, WS_HOLD

Каждый адресный диапазон имеет собственный регистр конфигурации контроллера внешней шины RAM_CYCLESx, содержащий следующие основные поля:

– WS_SETUP [2:0] – задает время предустановки сигналов nWR/nRD в цикле записи/чтения, выраженное в количестве тактов системной частоты (WS_SETUP + 1) в диапазоне от 1 до 8;

– WS_ACTIVE [6:0] – задает длительность низкого уровня сигналов nWR/nRD в цикле записи/чтения, выраженное в количестве тактов системной частоты (WS_ACTIVE + 1) в диапазоне от 1 до 128;

– WS_HOLD [2:0] – задает время удержания сигналов nWR/nRD в цикле записи/чтения, выраженное в количестве тактов системной частоты (WS_HOLD + 1) в диапазоне от 1 до 8;

– ENABLE_TUNE – бит разрешения использования собственных настроек длительности:

- 0 – для определения длительности используется биты WAIT_STATE регистра CONTROL;
- 1 – для определения длительности используется биты WS_ACTIVE, WS_SETUP и WS_HOLD регистра RAM_CYCLESx.

## 16.2 Работа с внешней NAND флэш-памятью

Для работы контроллера внешней системной шины с внешними микросхемами NAND флэш-памяти необходимо задать режим работы через регистр CONTROL. В зависимости от скорости работы ядра микросхемы и внешних устройств необходимо задать времена выполнения различных этапов работы NAND флэш-памяти через регистр NAND_CYCLES. После этого обращения в область памяти отображаемой на внешнюю системную шину будут перекодироваться в командные, адресные и обмена данными циклы обращения с NAND флэш-памятью через выводы внешней системной шины D[7:0], ALE, CLE и BUSY.

Схема подключения внешней NAND флэш-памяти приведена на рисунке 29

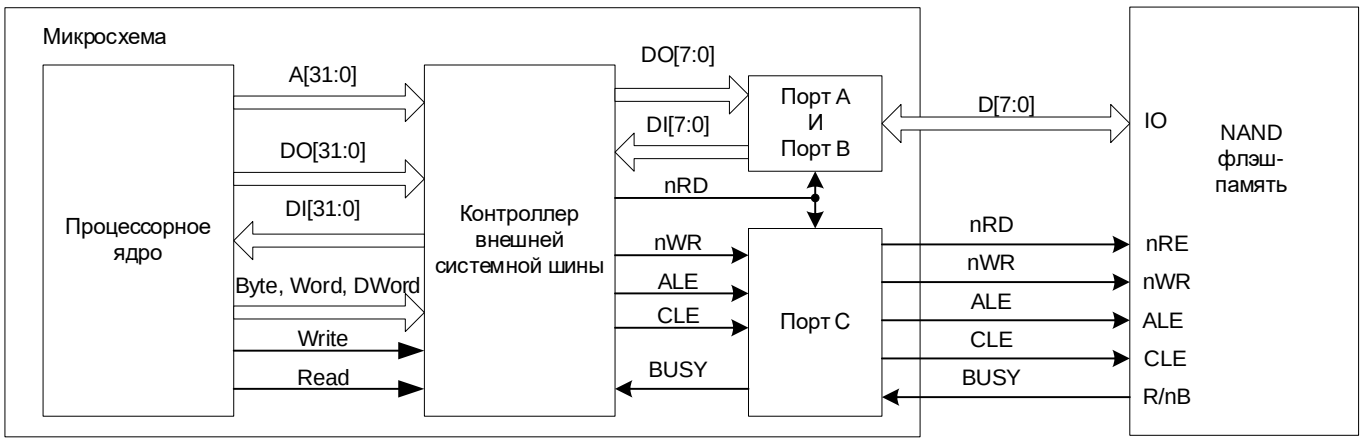



Рисунок 29 – Подключение внешней NAND флэш-памяти

Контроллер внешней системной шины имеет сигнал BUSY для подключения соответствующего вывода внешней NAND флэш-памяти. Сигналы BUSY от различных микросхем внешней NAND флэш-памяти объединяются по логическому И на выводе контроллера и формируют общий сигнал BUSY. При работе с NAND флэш-памятью тип выполняемой операции кодируется адресом обращения, а данные и адрес передаются данными при записи и чтении памяти. Формат кодирования адреса обращения представлен в таблице 127.

Таблица 127 – Формат кодирования адреса обращения

| Адрес обращения | Фаза команды | Фаза данных |
|---|---|---|
| A[31:24] | Не имеет значения, но должно попадать в адресные диапазоны внешней системной шины:<br>0x10…0x1F;<br>0x30…0x3F;<br>0x50…0xCF | |
| A[23:21] | ADR_CYCLES[2:0]<br>000 – 0 циклов;<br>001 – 1 цикл;<br>…<br>111 – 7 циклов | A[23:22] не имеют значения |
| | | A[21] сигнализирует о необходимости снятия сигнала выбора кристалла по завершении текущей команды, если этот бит в нуле, то устройство остается выбранным по завершении команды |
| A[20] | Выполнение завершающей команды:<br>0 – не выполнять;<br>1 – выполнять | |
| A[19] | Всегда 0 | Всегда 1 |
| A[18:11] | Код завершающей команды:<br>ECMD[7:0];<br>0x10/0x11 Page Program;<br>0xD0 Block Erase | |
| A[10:3] | Код начальной команды:<br>SCMD[7:0];<br>0x00/0x01 – Read1;<br>0x50 – Read2;<br>0x90 – Read ID;<br>0xFF – Reset;<br>0x80 – Page Program;<br>0x60 – Block Erase;<br>0x70 – Read Status | Не имеет значения |
| A[2:0] | Не имеет значения | |

Более подробную информацию о командах NAND флэш-памяти см. в документации на этот тип микросхем.

Пример работы с NAND флэш-памятью:

```
        //
==========================================================
        // Инициализация контроллера внешней системной шины для работы с NAND Flash
        //
==========================================================
        NAND_CYCLES = 0x02A63466;
        // время t_rr = 2 цикла HCLK или 20 нс при частоте HCLK 100 МГц
        // время t_alea = 10 циклов
        // время t_whr = 6 циклов
        // время t_wp = 3 цикла
        // время t_rea = 4 цикла
        // время t_wc = 6 циклов
        // время t_rc = 6 циклов

        CONTROL = 0x00000004;
        // NAND = 1;

        //
==========================================================
        // Чтение ID микросхемы
        //
==========================================================

        unsigned char IDH;
        unsigned char IDL;

        // Фаза команды
        *((volatile unsigned char *) (0x77200480)) = 0x00;
        // ADR_CYCLE = 1
        // SCMD = 0x90 (READ)
        // Address 1 cycle = 0x00

        // Фаза данных
        IDL = *((volatile unsigned char *)(0x77080000));
        IDH = *((volatile unsigned char *)(0x77080000));

        //
==========================================================
        // Стирание блока памяти
        //
==========================================================

        // Фаза команды
        *((volatile unsigned char *)(0x70768300))=0x11;
        *((volatile unsigned char *)(0x70768301))=0x22;
        *((volatile unsigned char *)(0x70768302))=0x33;
        // ADR_CYCLE = 3
        // выполнять завершающую команду
```

```
        // ECMD= 0xD0
        // SCMD = 0x60
        // Address 1 cycle = 0x11
        // Address 2 cycle = 0x22
        // Address 1 cycle = 0x33
        while (CONTROL!=0x080 ) {};
        // Ждем R/nB

        // Фаза команды
        *((volatile unsigned char *)(0x70000380+addon))=0x00;
        // ADR_CYCLE = 0
        // SCMD = 0x70
        // Фаза данных
        IDL = *((volatile unsigned char *)(0x77080000));
        If (IDL & 0x01==0x01) Error ();
        // Если бит IO0==1 то стирание не выполнено

        //
==============================================================
        // Запись страницы
        //
==============================================================

        // Фаза команды
        *((volatile unsigned char *)(0x70800400))=0x11;
        *((volatile unsigned char *)(0x70800400))=0x22;
        *((volatile unsigned char *)(0x70800400))=0x33;
        *((volatile unsigned char *)(0x70800400))=0x44;
        // ADR_CYCLE = 4
        // SCMD = 0x80

        // Фаза данных
        *((volatile unsigned char *)(0x70088000+addon))=0xBB;
        *((volatile unsigned char *)(0x70088000+addon))=0xCC;
        *((volatile unsigned char *)(0x70088000+addon))=0xDD;
        // не выполнять завершающую команду
        // ECMD= 0x10
        ...
        *((volatile unsigned char *)(0x70188000+addon))=0xEE;
        // не выполнять завершающую команду
        // ECMD= 0x10
        // Дынные 0 – 0xBB, 1 – 0xCC,... N – 0xEE
        // N от 1 до 528
        while (CONTROL!=0x080 ) {};
        // Ждем R/nB

        // Фаза команды
```

```
*((volatile unsigned char *)(0x70000380+addon))=0x00;
// ADR_CYCLE = 0
// SCMD = 0x70
// Фаза данных
IDL = *((volatile unsigned char *)(0x77080000));
If (IDL & 0x01==0x01) Error ();
// Если бит IO0==1 то запись не выполнена

//
============================================================
// Чтение страницы
//
============================================================

// Фаза команды
*((volatile unsigned char *)(0x70800000))=0x11;
*((volatile unsigned char *)(0x70800000))=0x22;
*((volatile unsigned char *)(0x70800000))=0x33;
*((volatile unsigned char *)(0x70800000))=0x44;
// ADR_CYCLE = 4
// SCMD = 0x00
while (CONTROL!=0x080 ) {};
// Ждем R/nB

// Фаза данных
IDL=*((volatile unsigned char *)(0x70080000));
IDH=*((volatile unsigned char *)(0x70080000));
If (IDL != 0xBB || IDH != 0xCC)  Error ();
// Если считали не то, что записали, то ошибка
```

## 16.3 Описание регистров блока контроллера внешней системной шины

Таблица 128 – Описание регистров блока контроллера внешней системной шины

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x400F_0000 | EXT_BUS | Контроллер внешней системной шины |
| Смещение | | |
| 0x50 | NAND_CYCLES | Регистр управления работой с NAND флэш-памятью |
| 0x54 | CONTROL | Регистр управления внешней системной шиной |
| 0x58 | RAM_CYCLES1 | Регистр индивидуальной настройки параметров обмена для адресного пространства 0x10000000-0x1FFFFFFF |
| 0x5C | RAM_CYCLES2 | Регистр индивидуальной настройки параметров обмена для адресного пространства 0x50000000-0x5FFFFFFF |
| 0x60 | RAM_CYCLES3 | Регистр индивидуальной настройки параметров обмена для адресного пространства 0x60000000-0x6FFFFFFF |

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x400F_0000 | EXT_BUS | Контроллер внешней системной шины |
| Смещение | | |
| 0x64 | RAM_CYCLES4 | Регистр индивидуальной настройки параметров обмена для адресного пространства 0x70000000-0xDFFFFFFF |

### 16.3.1 CONTROL

Таблица 129 – Регистр CONTROL

| Номер | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | U | U | U | U |
| Сброс | 1 | 1 | 1 | 1 | | | | |
| | WAIT_STATE[3:0] | | | | - | | | |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | RO | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 1 | 0 | 0 | 0 | | 0 | 0 | 1 |
| | BUSY | LOW16 | LOW8 | ENDIAN | CPOL | NAND | RAM | ROM |

Таблица 130 – Описание бит регистра CONTROL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…12 | WAIT STATE[3:0] | Количество тактов шины AHB, необходимых для стандартного цикла записи/чтения. Сигналы nRD/nWR устанавливаются в момент времени ¼ WAIT_STATE, снимаются в момент времени ¾ WAIT_STATE |
| 11…8 | - | Зарезервировано |
| 7 | BUSY | Сигнал занятости NAND флэш-памяти:<br>1 – операция завершена;<br>0 – операция не завершена |
| 6 | LOW16 | Выравнивание данных по 16 младшим разрядам внешней системной шины:<br>1 – данные записываются и читаются всегда с 16 младших разрядов данных;<br>0 – обычный режим работы шины.<br>Запись и чтение 32-разрядных данных при LOW16 = 1 происходит автоматически за два обращения на внешнюю шину.<br>Чтение/запись 8 битов не выполняется |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 5 | LOW8 | Выравнивание данных по 8 младшим разрядам внешней системной шины:<br>1 – данные записываются и читаются всегда с 8 младших разрядов данных;<br>0 – обычный режим работы шины.<br>Запись и чтение 32-разрядных данных при LOW8 = 1 происходит автоматически за четыре обращения на внешнюю шину.<br>Чтение/запись 16 битов не выполняется |
| 4 | ENDIAN | Всегда записывать ноль |
| 3 | CPOL | Бит задания полярности сигнала CLKO:<br>0 – положительная полярность;<br>1 – отрицательная полярность |
| 2 | NAND | Бит глобального разрешения с внешней NAND флэш-памятью:<br>1 – выбрана внешняя NAND флэш-память;<br>0 – внешняя NAND флэш-память не выбрана.<br>Одновременная установка нескольких бит 2..0 не допустима, в этом случае запрещается работа со всей памятью |
| 1 | RAM | Бит глобального разрешения памяти RAM:<br>1 – выбрана память RAM;<br>0 – память RAM не выбрана |
| 0 | ROM | Бит глобального разрешения памяти ROM:<br>1 – выбрана память ROM;<br>0 – память ROM не выбрана |

### 16.3.2 NAND_CYCLES

Таблица 131 – Регистр NAND_CYCLES

| Номер | 31…28 | 27…24 | 23…20 | 19…16 | 15…12 | 11…8 | 7…4 | 3…0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | t_rr | t_alea | t_whr | t_wp | t_rea | t_wc | t_rc |

Таблица 132 – Описание бит регистра NAND_CYCLES

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…28 | - | Зарезервировано |
| 27...24 | t_rr[3:0] | Время от снятия сигнала BUSY до операции чтения:<br>0000 – 0 HCLK циклов;<br>0001 – 1 HCLK цикл;<br>….<br>1111 – 15 HCLK циклов.<br>Типовое значение для NAND флэш-памяти составляет 20 нс |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 23...20 | t_alea[3:0] | Время доступа к регистрам ID (аналогично t_rr).<br>Типовое значение для NAND флэш-памяти составляет 100 нс |
| 19…16 | t_whr[3:0] | Время доступа к регистру статуса (аналогично t_rr).<br>Типовое значение для NAND флэш-памяти составляет 60 нс |
| 15…12 | t_wp[3:0] | Время доступа по записи (аналогично t_rr).<br>Типовое значение для NAND флэш-памяти составляет 25 нс |
| 11…8 | t_rea[3:0] | Время доступа по чтению (аналогично t_rr).<br>Типовое значение для NAND флэш-памяти составляет 35 нс |
| 7...4 | t_wc[3:0] | Время цикла записи (аналогично t_rr).<br>Типовое значение для NAND флэш-памяти составляет 60 нс |
| 3…0 | t_rc[3:0] | Время цикла чтения (аналогично t_rr).<br>Типовое значение для NAND флэш-памяти составляет 60 нс |

### 16.3.3 RAM_CYCLESx

Таблица 133 – Регистр RAM_CYCLESx

| Номер | 31...15 | 14 | 13…11 | 10...8 | 7…1 | 0 |
|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W |
| Сброс | | 0 | 0 | 0 | 0 | 0 |
| | - | USE_READY | WS_HOLD | WS_SETUP | WS_ACTIVE | ENABLE_TUNE |

Таблица 134 – Описание бит регистра RAM_CYCLESx

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...15 | | Зарезервировано |
| 14 | USE_READY | Разрешение опроса внешнего сигнала READY на выводе PF[2], динамически определяющего аппаратные состояния ожидания в цикле обмена по внешней системной шине:<br>1 – опрашивается;<br>0 – не опрашивается.<br>Опрос сигнала READY производится на последнем такте фазы ACTIVE, если READY находится в активном состоянии – осуществляется переход к фазе HOLD и завершение цикла обмена, в противном случае повторяется опрос на каждом последующем такте, пока общее число тактов трех фаз: SETUP, ACTIVE и HOLD – не превысит 256 тактов системной частоты (максимально допустимое время цикла обращения). После этого обмен завершается |
| 13...11 | WS_HOLD | Время удержания сигналов nWR/nRD в цикле записи/чтения, выраженное в количестве тактов системной частоты (WS_HOLD + 1) в диапазоне от 1 до 8 |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 10...8 | WS_SETUP | Время предустановки сигналов nWR/nRD в цикле записи/чтения, выраженное в количестве тактов системной частоты (WS_SETUP + 1) в диапазоне от 1 до 8 |
| 7...1 | WS_ACTIVE | Длительность низкого уровня сигналов nWR/nRD в цикле записи/чтения, выраженная в количестве тактов системной частоты (WS_ACTIVE + 1) в диапазоне от 1 до 128 |
| 0 | ENABLE_TUNE | Разрешение настройки параметров обмена соответствующего диапазона адресов:<br>1 – разрешена;<br>0 – запрещена |

# 17 Режим StandAlone

Данный режим предназначен для прямого доступа к контроллерам интерфейса Ethernet (для микросхем с функцией Ethernet) и интерфейса по ГОСТ Р 52070-2003. При этом ядро и все остальные блоки находятся в состоянии сброса за исключением генератора тактовой частоты, который обеспечивает тактирование контроллеров. Для увеличения частоты работы контроллеров можно использовать PLL, предварительно настроив коэффициент умножения и включив блок. Для осуществления перехода в этот режим можно воспользоваться функциями загрузочного ПЗУ, описанного выше. Либо провести инициализацию этих функций программно, установив режим загрузочного ПЗУ – микроконтроллер с режимом отладки, как описано ниже.

Режим StandAlone 1 (доступ только к контроллеру Ethernet):

```
RST_CLK->PLL_CONTROL=0x304;
RST_CLK->HS_CONTROL= 3;
RST_CLK->CPU_CLOCK=0x00000107;
BKP->REG_0E &= ~(1<<7);
RST_CLK->ETH_CLOCK=0x19000000;
BKP->REG_0E |= 1<<6;
```

Режим StandAlone 2 (доступ только к контроллеру по ГОСТ Р 52070-2003):

```
RST_CLK->PLL_CONTROL=0x904;
RST_CLK->HS_CONTROL=1 или 3; (3 – генератор; 1 – осциллятор)
RST_CLK->CPU_CLOCK=0x00000106;
RST_CLK->PER_CLOCK|=1<<4 | 1<<9| 1<<10 | 1<<27;
BKP->REG_0E &= ~(1<<7);
RST_CLK->ETH_CLOCK=0x02000000;
BKP->REG_0E |= 1<<6;
```

Режим StandAlone 3 (доступ к контроллерам ГОСТ Р 52070-2003 и Ethernet):

```
RST_CLK->PLL_CONTROL=0x304;
RST_CLK->HS_CONTROL=3;
RST_CLK->CPU_CLOCK=0x00000107;
RST_CLK->PER_CLOCK|=1<<4 | 1<<9| 1<<10 | 1<<27;
BKP->REG_0E &= ~(1<<7);
RST_CLK->ETH_CLOCK=0x1B000000;
BKP->REG_0E |= 1<<6;
```

Для режима Stand Alone 2 необходимо установить внешний осциллятор 8 МГц на входы OSC_IN и OSC_OUT. В режимах Stand Alone 1 и 3 внешняя частота на вход OSC_IN должна быть 25 МГц и подаваться с внешнего генератора. После перехода в режим StandAlone обмен данными с контроллером осуществляется посредством параллельного асинхронного интерфейса. Временная диаграмма циклов работы асинхронного

интерфейса приведена на рисунках 30, 31. Для приведенных временных диаграмм тактовая частота контроллера составляет 50 МГц. Если частоту увеличить в два раза, то все временные параметры на диаграмме можно уменьшить в два раза. Дальнейшее увеличение частоты не даст прироста пропускной способности, так как для параллельного асинхронного интерфейса она ограничена частотой 50 МГц.

Примечание – Назначение выводов микросхемы в режиме StandAlone приведено в таблице 5 раздела 4 «Описание выводов».

При доступе к контроллеру интерфейса по ГОСТ Р 52070-2003 (ITCMLAEN = 0) адресация системной шины микросхемы всегда словная (32 бита), т.е. единица адреса интерфейса системной шины соответствует 32 битам.

При доступе к контроллеру интерфейса Ethernet (ITCMLAEN = 1) адресация системной шины микросхемы всегда байтовая (8 бит), т.е. единица адреса интерфейса системной шины соответствует 8 битам. После каждой записи в буферную память Ethernet необходимо выполнять верификацию записанных данных и, в случае неудачи, повторять процедуру записи. При аппаратном и программном переходах в режим Stand Alone, с доступом к контроллеру Ethernet, рекомендуется использовать один внешний источник синхросигнала для устройства мастера параллельной шины и микросхемы в ведомом режиме Stand Alone Ethernet, при этом внутренние частоты мастера и ведомого должны быть равными.

Адреса для всех областей памяти и регистров внутри микросхемы в карте памяти и карте регистров указаны байтовые (8 бит). Это необходимо учитывать при подключении внешнего устройства.

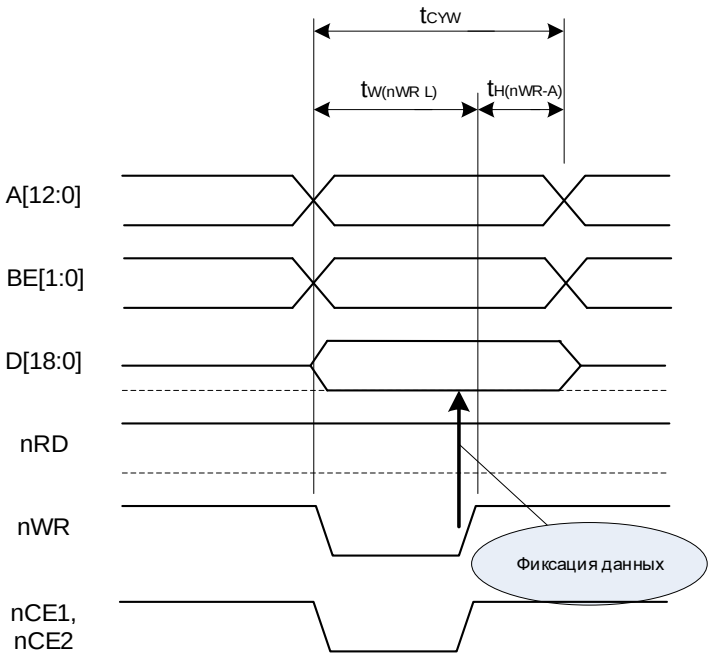



Рисунок 30 – Временная диаграмма цикла записи

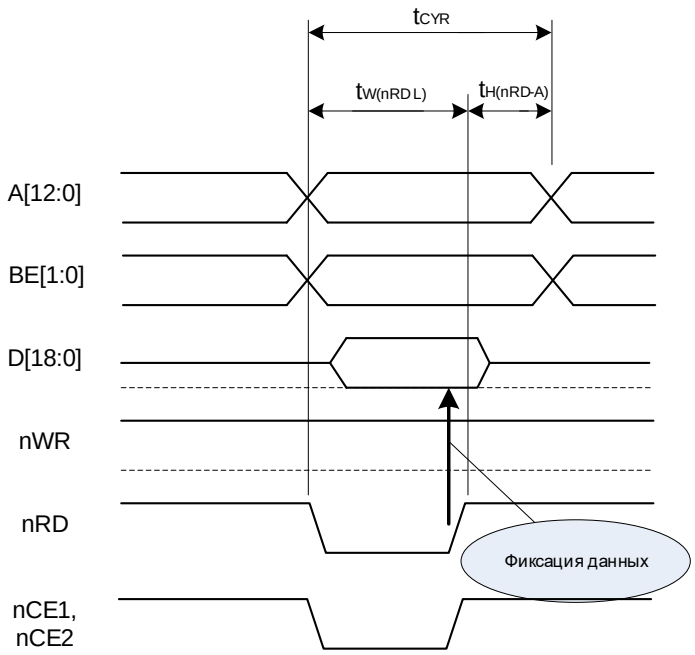



Рисунок 31 – Временная диаграмма цикла чтения

Таблица 135 – Временные характеристики цикла чтения и записи при тактовой частоте микросхемы 50 МГц

| Наименование параметра, единица измерения | Буквенное обозначение параметра | Норма параметра | |
|---|---|---|---|
| | | не менее | не более |
| Время цикла записи, нс | $t_{CYW}$ | 100 | - |
| Длительность сигнала низкого уровня nWR, нс | $t_{W(nWR\ L)}$ | 40 | - |
| Время удержания сигналов адреса, данных и выборки байта относительно сигнала nWR, нс | $t_{H(nWR-A)}$ | 60 | - |
| Время цикла чтения, нс | $t_{CYR}$ | 100 | - |
| Длительность сигнала низкого уровня nRD, нс | $t_{W(nRD\ L)}$ | 100 | - |
| Время удержания сигналов адреса, данных и выборки байта относительно сигнала nRD, нс | $t_{H(nRD-A)}$ | 0 | - |

Активный уровень сигналов nWR, nRD, nCE1, nCE2, BE[1:0] низкий. Сигналы A[12:0], D[18:0], BE[1:0], nCE1, nCE2 должны быть установлены внешним устройством до начала цикла обмена и не должны изменять своего состояния в течение всего цикла обмена. Допускается удерживать сигнал nCE1 или nCE2 в состоянии логического нуля на протяжении нескольких последовательных циклов обмена.

# 18 Контроллер интерфейса USB

Контролер интерфейса USB реализует функции контроллера функционального устройства (Device) и управляющего устройства (Host) в соответствии со спецификацией USB 1.0.

Контроллер интерфейса USB поддерживает: режимы работы Full Speed (12 Мбит/с) и Low Speed (1,5 Мбит/с), контроль ошибок с помощью циклического избыточного кода (CRC), NRZI код приема/передачи, управляющие передачи (Control), сплошные передачи (Bulk), изохронные передачи (Isochronous) и передачи по прерываниям (Interrupt). Также поддерживаются конфигурирование USB Device от одной до четырех оконечных точек, автоматическая отправка SOF пакетов, вычисление оставшегося во фрейме времени. Каждая оконечная точка USB Device имеет собственный буфер FIFO размером 64 байта. USB Host поддерживает до 16 оконечных точек. USB Host имеет буфер FIFO размером 64 байта.

## 18.1 Инициализация контроллера при включении

При включении питания в первую очередь должны быть заданы параметры тактового сигнала блока USB. Параметры задаются в блоке «Сигналы сброса и тактовой частоты». Источником тактового сигнала для блока USB может быть либо встроенный высокочастотный генератор (HSI) или внешний осциллятор (HSE). Блок USB функционирует на частоте 48 МГц. Требуемая частота может быть получена умножением частоты одного из двух генераторов до требуемого значения. Умножение выполняется встроенным блоком PLLUSB.

Блок умножения позволяет провести умножение входной тактовой частоты на коэффициент от 2 до 16, задаваемый в поле PLLUSBMUL регистра PLL_CONTROL. Входная частота блока умножителя должна быть в диапазоне 2…16 МГц, а выходная должна составлять 48 МГц. При выходе блока умножителя тактовой частоты в расчетный режим вырабатывается сигнал PLLRDY. Блок включается с помощью сигнала PLLUSBON. Выходная частота используется как основная частота протокольной части интерфейса USB.

Для задания тактовой частоты блока необходимо соблюдать следующий порядок работы:

1 В регистре PER_CLOCK установить бит 2 для разрешения тактирования блока;

2 В регистре USB_CLOCK установить бит USBCLKEN, задать источник тактового сигнала в полях USBC1SEL и USBC2SEL;

3 В регистре PLL_CONTROL установить бит PLLUSBON и задать коэффициент умножения в поле PLLUSBMUL, если используется USBPLL.

После подачи тактового сигнала на блок USB необходимо выполнить сброс контроллера. Сброс выполняется установкой бита RESET_CORE в регистре USB_HSCR. Сигнал сброса необходимо удерживать как минимум 10 циклов тактовой частоты. После этого могут быть заданы параметры шины USB (скорость, полярность, наличие подтяжек).

### 18.2 Задание параметров шины USB и события подключения/отключения

Контроллер USB может быть сконфигурирован как USB Host или как USB Device. Конфигурация задается битом CORE_MODE в регистре USB_HSCR (0 – режим Device, 1 – режим Host). Прием/передача через физический интерфейс USB разрешается установкой бит EN_RX и EN_TX в этом же регистре. В режиме приема имеется возможность отключить передатчик в целях экономии потребления (EN_TX=0). Отключение всего блока в целом осуществляется при EN_RX=0.

В режиме Device параметры шины задаются в регистре USB_SC. Скорость задается битом SCFSR (0 – 1,5 Мбит/с, 1 – 12 Мбит/с), полярность битом SCFSP (0 – Low speed, 1 – Full speed) этого регистра.

В режиме HOST необходимо прямое подключение (без USB HUB) LOW SPEED устройства.

В режиме Host параметры шины задаются в регистре USB_HTXLC. Скорость задается битом FSLR (0 – 1,5 Мбит/с, 1 – 12 Мбит/с), полярность битом FSPL (0 – Low speed, 1 – Full speed) этого регистра.

В режиме Host контроллер автоматически определяет подключение или отключение устройства к шине. Бит CONEV регистра USB_HIS устанавливается в 1 при возникновении одного из событий.

### 18.3 Задание адреса и инициализация оконечных точек

Функциональный адрес устройства USB задается в регистре USB_SA.

Для инициализации конечной точки в первую очередь необходимо установить бит глобального разрешения всех оконечных точек (SCGEN = 1 в регистре USB_SC). Биты EPEN в регистрах USB_SEPx.CTRL должны быть установлены, чтобы разрешить соответствующую оконечную точку. Если предполагается использовать изохронной тип передачи оконечной точки, то необходимо установить бит EPISOEN в соответствующем регистре USB_SEPx.CTRL.

### 18.4 Транзакция IN (Usb Device)

Если на шине появляется IN пакет и адрес совпадает с заданным в регистре USB_SA, бит SCTDONE регистра USB_SIS устанавливается в «1».

Если оконечная точка не готова (бит EPRDY = 0 в регистре USB_SEPx.CTRL), то контроллер отправляет NAK пакет (рисунок 32а). Бит SCNAKSENT регистра USB_SEPx.STS устанавливается в «1».

Если оконечная точка готова и установлен бит EPSSTALL в регистре USB_SEPx.CTRL, то контроллер отправляет STALL пакет (рисунок 32б). Бит SCSTALLSENT регистра USB_SEPx.STS устанавливается в «1».

Если оконечная точка готова (рисунок 32в), биты SCTTYPE[1:0] в регистре USB_SEPx.TS устанавливаются в «1» для конечной точки с номером, содержащимся в поле пакета. Контроллер может передавать пакет данных. Пакет данных формируется записью в регистр USB_SEPx.TXFD побайтно в FIFO оконечной точки. Запись «1» в USB_SEPx.TXFDC сбрасывает указатель FIFO передачи в «0».

Максимальный размер передаваемого пакета составляет 64 байта. Попытка записи более 64 байт подряд приведет к переполнению FIFO. Перед началом формирования очередного пакета необходимо выполнять сброс указателя FIFO.

Если в ответ на переданные данные хост отправляет ACK пакет, то бит SCACKRXED в регистре USB_SEPx.STS устанавливается в «1». Для отправки следующего пакета необходимо инвертировать бит EPDATASEQ в регистре USB_SEPx.CTRL, чтобы соблюдалась очередность оправки пакетов DATA0, DATA1.

После окончания транзакции бит SCTDONE регистра USB_SIS должен быть очищен записью «1».

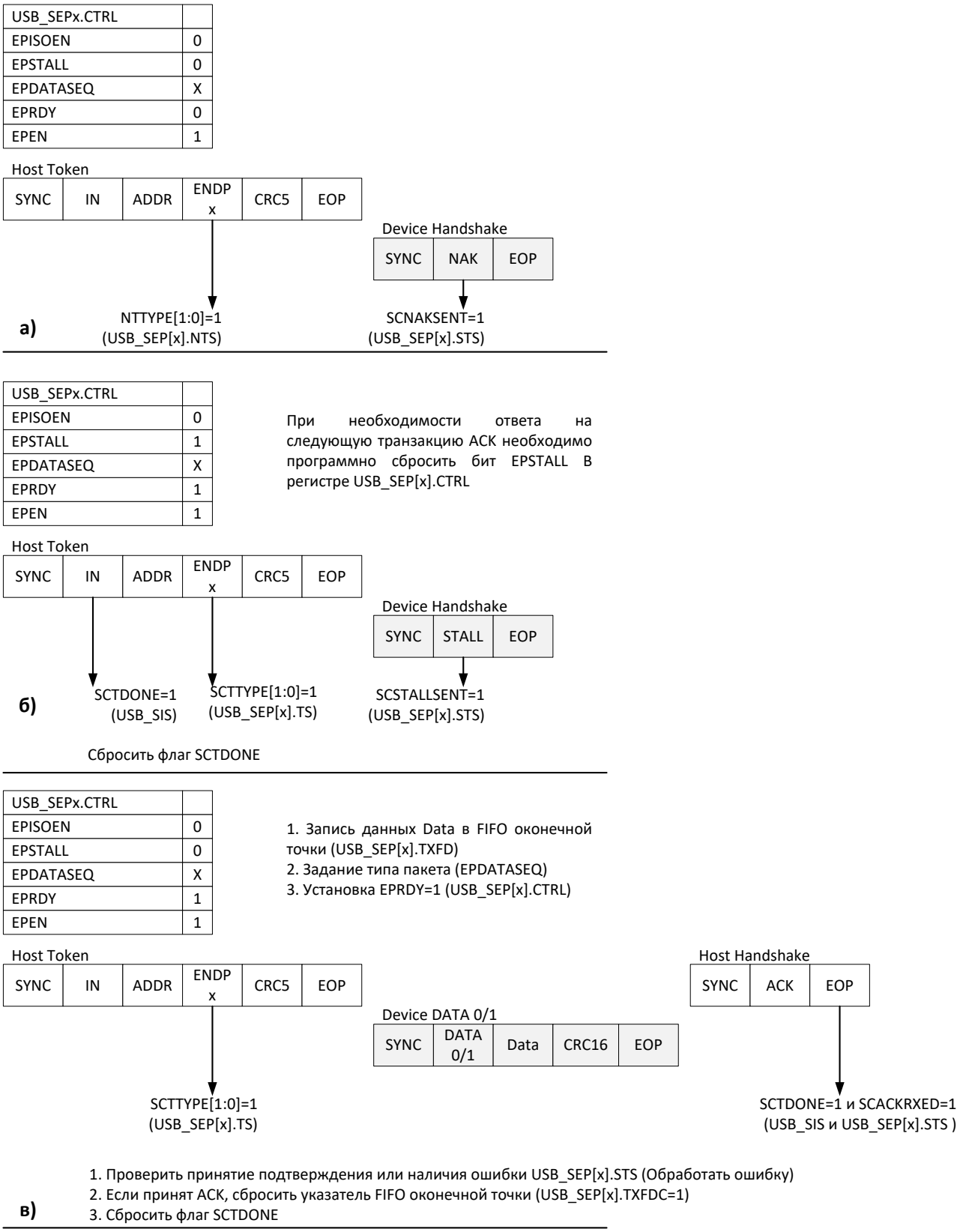



Рисунок 32 – Транзакция IN (USB Device)
а – оконечная точка не готова;
б – установлен бит EPSTALL;
в – оконечная точка готова

## 18.5 Транзакция SETUP/OUT (Usb Device)

Если на шине появляется SETUP/OUT пакет, адрес совпадает с заданным в регистре USB_SA и оконечная точка готова (бит EP_READY = 1 в регистре ENDPOINTx_CONTROL), то бит SCTDONE регистра USB_SIS устанавливается в «1».

Если оконечная точка не готова (бит EPRDY = 0 в регистре USB_SEPx.CTRL), то контроллер отправляет NAK пакет (рисунок 33а). Бит SCNAKSENT регистра USB_SEPx.STS устанавливается в «1».

Если оконечная точка готова и установлен бит EPSSTALL в регистре USB_SEPx.CTRL, то контроллер отправляет STALL пакет (рисунок 33б). Бит SCSTALLSENT регистра USB_SEPx.STS устанавливается в «1».

Если оконечная точка готова (рисунок 33в) и на шине был пакет SETUP, то биты SCTTYPE[1:0] в регистре USB_SEPx.TS устанавливаются в значение 00 для конечной точки с номером, содержащимся в поле пакета. Если пакет OUT, то значение SCTTYPE[1:0] = 2.

Когда на шине появляется DATA0/DATA1 пакет, то данные начинают записываться побайтно в FIFO приема соответствующей оконечной точки. После записи каждого байта увеличивается на единицу счетчик принятых байт. Принятые байты считываются через регистр USB_SEPx.RXFD. Количество принятых байт содержится в регистре USB_SEPx.RXFDC. После приема очередного пакета необходимо выполнять сброс указателя FIFO приема записью «1» в регистр USB_SEPx.RXFC.

После окончания транзакции бит SCTDONE регистра USB_SIS должен быть очищен записью «1».

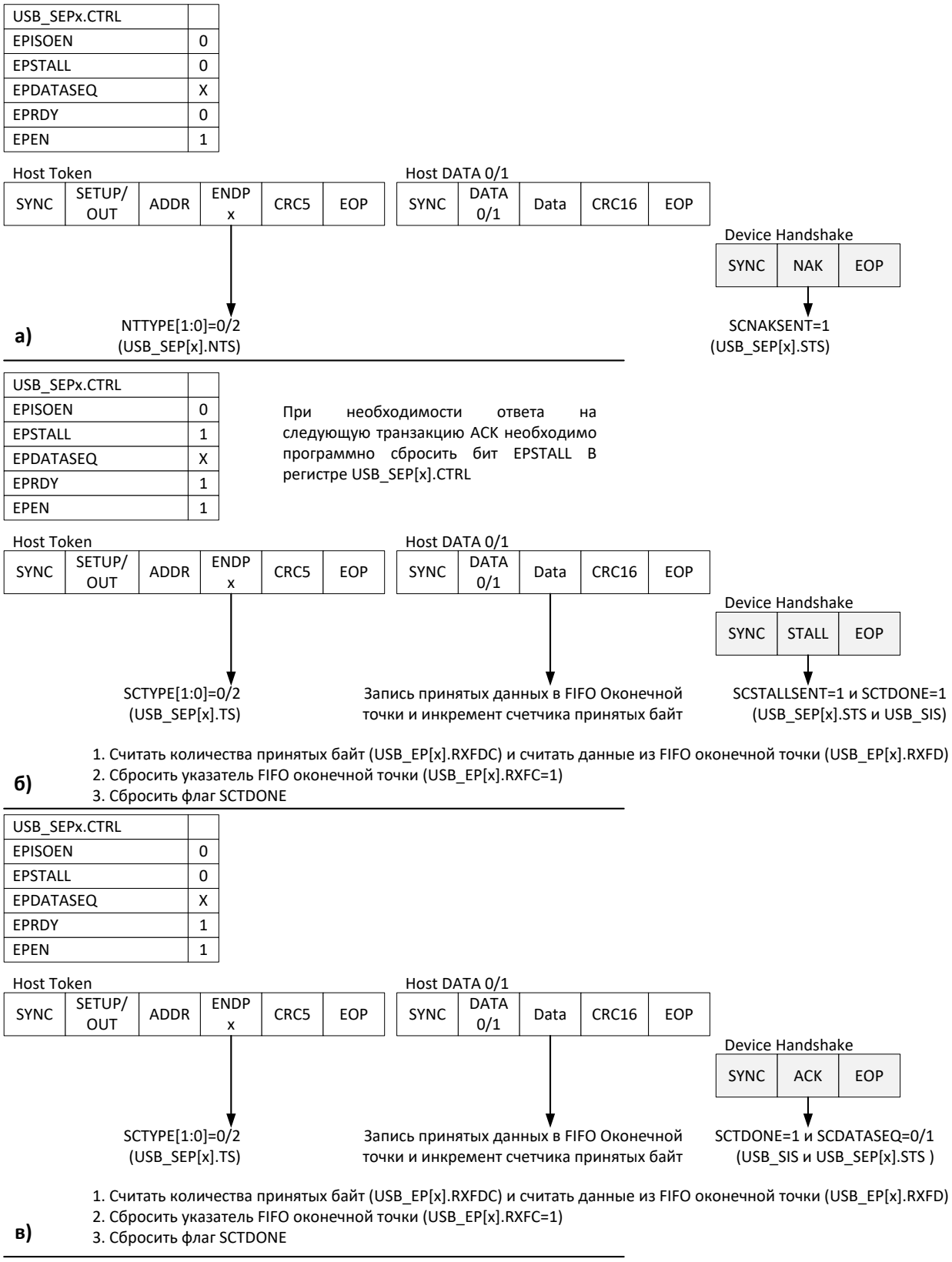



Рисунок 33 – Транзакция SETUP/OUT (USB Device)
а – оконечная точка не готова;
б – установлен бит EPSTALL;
в – оконечная точка готова

## 18.6 Транзакция SETUP/OUT (Usb Host)

Для начала транзакции должны быть заданы адрес устройства (регистр USB_HTXA), оконечная точка (регистр USB_HTXE) и тип token пакета (регистр USB_HTXT). Данные записываются побайтно в регистр USB_HTXFD. Максимальный размер передаваемого пакета составляет 64 байт. Попытка записи более 64 байт подряд приведет к переполнению FIFO. Запись «1» в USB_HTXFDC сбрасывает указатель FIFO передачи в «0». Перед началом формирования очередного пакета необходимо выполнять сброс указателя FIFO. Транзакция запускается при установке бита TREQ регистра USB_HTXC. Host отправляет пакет Setup/Out и пакет данных.

После окончания транзакции бит TDONE = 1 (регистр USB_HIS). Этот бит перед началом каждой транзакции должен быть очищен записью «1». PID принятого пакета записывается в регистре USB_HRXP.

Если в ответ получен пакет NAK (рисунок 34а), то бит NAKRXED = 1 (регистр USB_HRXS).

Если в ответ получен пакет STALL (рисунок 34б), то бит STALLRXED = 1 (регистр USB_HRXS).

Если в ответ получен пакет ACK (рисунок 34в), то бит ACKRXED = 1 (регистр USB_HRXS).

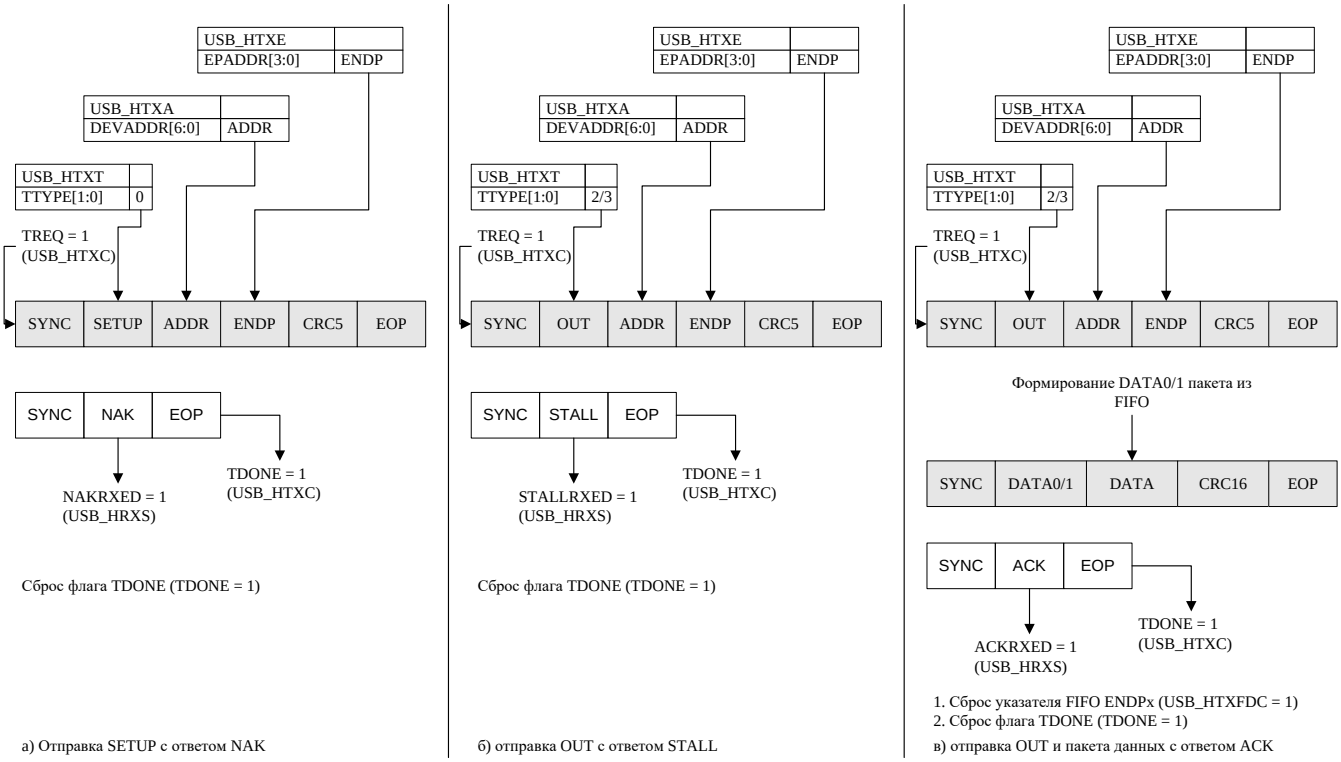



Рисунок 34 – Транзакция SETUP/OUT (USB Host)

## 18.7 Транзакция IN (Usb Host)

Для начала транзакции должны быть заданы адрес устройства (регистр USB_HTXA), оконечная точка (регистр USB_HTXE) и тип token пакета (регистр USB_HTXT). Транзакция запускается при установке бита TREQ регистра USB_HTXC. Host отправляет IN пакет.

После окончания транзакции бит TDONE = 1 (регистр USB_HIS). Этот бит перед началом каждой транзакцией должен быть очищен записью «1». PID принятого пакета записывается в регистре USB_HRXP.

Если в ответ получен пакет NAK (рисунок 35а), то бит NAKRXED = 1 (регистр USB_HRXS).

Если в ответ получен пакет STALL (рисунок 35б), то бит STALLRXED = 1 (регистр USB_HRXS).

Если приходит DATA0/DATA1 пакет (рисунок 35в), то данные начинают записываться побайтно в FIFO приема. После записи каждого байта увеличивается на единицу счетчик принятых байт. Принятые байты считываются через регистр USB_HRXFD. Количество принятых байт содержится в регистре USB_HRXFDC. После приема очередного пакета необходимо выполнять сброс указателя FIFO приема записью «1» в регистр USB_HRXFC. Бит DATASEQ регистра USB_HRXS отображает тип принятого пакета данных (0 – DATA0, 1 – DATA1).

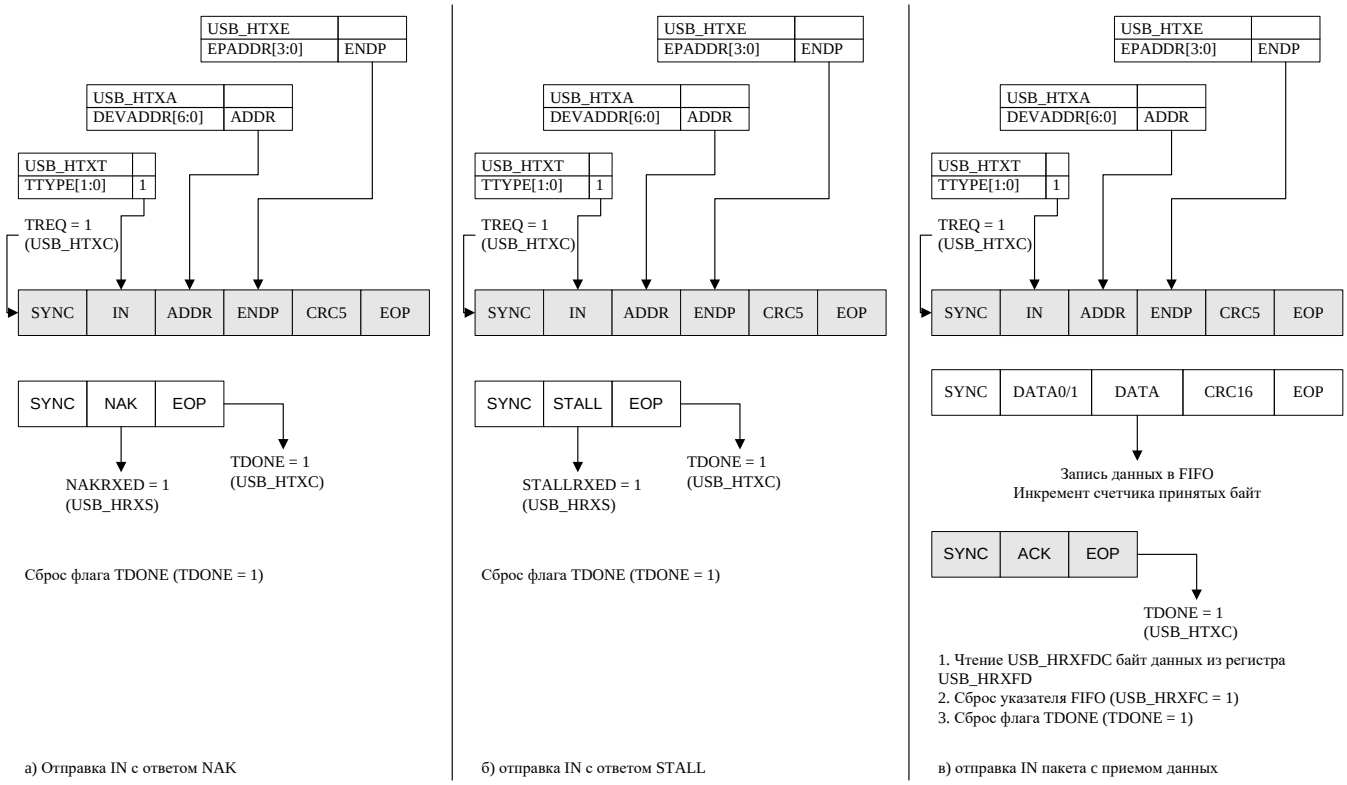



Рисунок 35 – Транзакция IN (USB Host)

## 18.8 Отправка SOF пакетов и отсчет времени (Usb Host)

Для того чтобы контроллер автоматически отправлял SOF пакеты на Full speed необходимо установить SOFEN в регистре USB_HTXSE. Если FSPL = 1 (регистр USB_HTXLC), то SOF будет автоматически отсылаться каждые 1 мс. Если FSPL = 0, то автоматически будет отправляться EOP каждые 1 мс.

После отправки SOF пакета бит SOFS = 1 (регистр USB_HIS). Этот бит должен быть очищен записью «1».

Контроллер ведет счет времени во фрейме таймером. Таймер увеличивается на частоте 48 МГц и имеет 48000 тактов в 1 мс фрейме. Старший байт таймера содержится в

регистре USB_HSTM. Этот регистр может быть использован для вычисления времени, оставшегося во фрейме.

## 18.9 Описание регистров управления контроллером интерфейса USB

Таблица 136 – Описание регистров управления контроллером интерфейса USB

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x4001_0000 | USB | Контроллер интерфейса USB |
| Смещение | | |
| 0x380 | USB_HSCR | Общее управление для контроллера интерфейса USB |
| 0x384 | USB_HSVR | Версия аппаратного контроллера интерфейса USB |
| | | |
| Контроллер HOST | | |
| 0x00 | USB_HTXC | Регистр управления передачей пакетов со стороны хоста |
| 0x04 | USB_HTXT | Регистр задания типа передаваемых пакетов со стороны хоста |
| 0x08 | USB_HTXLC | Регистр управления линиями шины USB |
| 0x0C | USB_HTXSE | Регистр управление автоматической отправки SOF |
| 0x10 | USB_HTXA | Регистр задания адреса устройства для отправки пакета |
| 0x14 | USB_HTXE | Регистр задания номера оконечной точки для отправки пакета |
| | | |
| 0x18, 0x1C | USB_HFN_H USB_HFN_L | Регистр задания номера фрейма для отправки SOF |
| | | |
| 0x20 | USB_HIS | Регистр флагов событий контроллера хост. |
| 0x24 | USB_HIM | Регистра флагов разрешения прерываний по событиям контролера хоста |
| | | |
| 0x28 | USB_HRXS | Регистр состояния очереди приема данных хоста |
| 0x2C | USB_HRXP | Регистр отображения PID принятого пакета |
| 0x30 | USB_HRXA | Регистр отображения адреса устройства, от которого принят пакет |
| 0x34 | USB_HRXE | Регистр отображения номер оконечной точки, от которой принят пакет |
| 0x38 | USB_HRXCS | Регистр отображения состояния подсоединения устройства |
| | | |
| 0x3C | USB_HSTM | Регистр расчета времени фрейма |
| | | |
| 0x80 | USB_HRXFD | Данные очереди приема |
| 0x8C | USB_HRXFDC | Число принятых данных в очереди |
| 0x90 | USB_HRXFC | Управление очередью приема |
| | | |
| 0xC0 | USB_HTXFD | Данные для передачи |
| 0xD0 | USB_HTXFDC | Управление очередью передачи |

| Базовый Адрес | Название | Описание |
|---|---|---|
| | | |
| Контроллер SLAVE | | |
| 0x100<br>0x110<br>0x120<br>0x130 | USB_SEP0.CTRL<br>USB_SEP1.CTRL<br>USB_SEP2.CTRL<br>USB_SEP3.CTRL | Управление очередью нулевой оконечной точки |
| 0x104<br>0x114<br>0x124<br>0x134 | USB_SEP0.STS<br>USB_SEP1.STS<br>USB_SEP2.STS<br>USB_SEP3.STS | Состояние оконечной точки |
| 0x108<br>0x118<br>0x128<br>0x138 | USB_SEP0.TS<br>USB_SEP1.TS<br>USB_SEP2.TS<br>USB_SEP3.TS | Состояние типа передачи оконечной точки |
| 0x10C<br>0x11C<br>0x12C<br>0x13C | USB_SEP0.NTS<br>USB_SEP1.NTS<br>USB_SEP2.NTS<br>USB_SEP3.NTS | Состояние передачи NAK оконечной точки |
| | | |
| 0x140 | USB_SC | Управление контроллеров SLAVE |
| 0x144 | USB_SLS | Отображение состояния линий USB шины |
| 0x148 | USB_SIS | Флаги событий контроллера SLAVE |
| 0x14C | USB_SIM | Флаги разрешения прерываний от контроллера SLAVE |
| 0x150 | USB_SA | Функциональный адрес контроллера |
| 0x154,<br>0x158 | USB_SFN_H<br>USB_SFN_L | Номер фрейма |
| | | |
| 0x180<br>0x200<br>0x280<br>0x300 | USB_SEP0.RXFD<br>USB_SEP1.RXFD<br>USB_SEP2.RXFD<br>USB_SEP3.RXFD | Принятые данные оконечной точки |
| 0x18C,<br>0x20C,<br>0x28C,<br>0x30C | USB_SEP0.RXFDC<br>USB_SEP1.RXFDC<br>USB_SEP2.RXFDC<br>USB_SEP3.RXFDC | Число данных в оконечной точке |
| 0x190<br>0x210<br>0x290<br>0x310 | USB_SEP0.RXFC<br>USB_SEP1.RXFC<br>USB_SEP2.RXFC<br>USB_SEP3.RXFC | Управление очередью приема оконечной точки |
| 0x1C0<br>0x240<br>0x2C0<br>0x340 | USB_SEP0.TXFD<br>USB_SEP1.TXFD<br>USB_SEP2.TXFD<br>USB_SEP3.TXFD | Данные для передачи через оконечную точку |
| 0x1D0<br>0x250<br>0x2D0 | USB_SEP0.TXFDC<br>USB_SEP1.TXFDC<br>USB_SEP2.TXFDC | Управление очередью передачи оконечной точки |

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x350 | USB_SEP3.TXFDC | |

### 18.9.1 USB_HSCR

Таблица 137 – Регистр USB_HSCR

| Номер | 31…8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | D- PULL DOWN | D- PULL UP | D+ PULL DOWN | D+ PULL UP | EN RX | EN TX | RESET CORE | HOST MODE |

Таблица 138 – Описание бит регистра USB_HSCR

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7 | D- PULLDOWN | Управление встроенной подтяжкой линии D-:<br>0 – нет подтяжки вниз;<br>1 – есть подтяжка вниз |
| 6 | D- PULLUP | Управление встроенной подтяжкой линии D-:<br>0 – нет подтяжки вверх;<br>1 – есть подтяжка вверх |
| 5 | D+ PULLDOWN | Управление встроенной подтяжкой линии D+:<br>0 – нет подтяжки вниз;<br>1 – есть подтяжка вниз |
| 4 | D+ PULLUP | Управление встроенной подтяжкой линии D+:<br>0 – нет подтяжки вверх;<br>1 – есть подтяжка вверх |
| 3 | EN_RX | Разрешение работы приемника USB:<br>0 – запрещен;<br>1 – разрешен.<br>Может использоваться в энергосберегающих целях |
| 2 | EN_TX | Разрешение работы передатчика USB:<br>0 – запрещен;<br>1 – разрешен.<br>Может использоваться в энергосберегающих целях |
| 1 | RESET_CORE | Программный сброс контроллера:<br>1 – сброс контроллера (удерживать минимум 10 циклов USBCLK);<br>0 – рабочий режим |
| 0 | HOST_MODE | Режим работы вывода порта:<br>1 – режим HOST;<br>0 – режим Device |

### 18.9.2 USB_HSVR

Таблица 139 – Регистр USB_HSVR

| Номер | 31…8 | 7…4 | 3…0 |
|---|---|---|---|
| Доступ | U | RO | RO |
| Сброс | 0 | 0 | 0 |
| | - | REVISION | VERSION |

Таблица 140 – Описание бит регистра USB_HSVR

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7…4 | REVISION | Номер Ревизии |
| 3…0 | VERSION | Номер Версии |

### 18.9.3 Регистры режима HOST

#### 18.9.3.1 *USB_HTXC*

Таблица 141 – Регистр USB_HTXC

| Номер | 31…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | ISOEN | PREEN | SOFS | TREQ |

Таблица 142 – Описание бит регистра USB_HTXC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | - | Зарезервировано |
| 3 | ISOEN | Флаг разрешения изохронного режима:<br>1 – разрешение изохронного режима, ACK не посылается и не принимается. Необходимо, чтобы TRANS_TYPE_REG был установлен в IN_TRANS или OUTDATA0_TRANS. Изохронный режим не применим с другими типами передачи;<br>0 – запрещение изохронного режима |
| 2 | PREEN | Рекомендуется оставлять 0 |
| 1 | SOFS | Флаг задания синхронизации передачи с SOF:<br>1 – синхронизировать передачу с окончанием SOF. Передача будет запущена сразу за передачей SOF;<br>0 – передача не синхронизирована |
| 0 | TREQ | Флаг запроса передачи данных:<br>1 – запрос разрешения передачи данных, автоматически сбрасывается после передачи;<br>0 – запрещена передача |

#### 18.9.3.2 *USB_HTXT*

Таблица 143 – Регистр USB_HTXT

| Номер | 31…2 | 1 | 0 |
|---|---|---|---|
| Доступ | U | R/W | R/W |
| Сброс | 0 | 0 | 0 |
| | - | TTYPE | |

Таблица 144 – Описание бит регистра USB_HTXT

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…2 | - | Зарезервировано |
| 1…0 | TTYPE | Тип передачи:<br>00 – setup_trans;<br>01 – in_trans;<br>10 – outdata0_trans;<br>01 – outdata1_trans |

#### 18.9.3.3 *USB_HTXLC*

Таблица 145 – Регистр USB_HTXLC

| 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 |
| | - | FSLR | FSLP | DC | TXLS[1:0] | |

Таблица 146 – Описание бит регистра USB_HTXLC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…5 | - | Зарезервировано |
| 4 | FSLR | 1 – 12 Мбит/с;<br>0 – 1,5 Мбит/с |
| 3 | FSPL | 1 – FULL SPEED полярность шины USB;<br>0 – LOW SPEED полярность шины USB.<br>Если host работает с full speed устройством, full speed полярность должна быть установлена. Если работа ведется с low speed устройством на прямую, то должна быть установлена low speed полярность. Работа с low speed через hub не поддерживается |
| 2 | DC | Режим управления линиями шины USB:<br>1 – разрешение прямого управления состоянием линий USB шины;<br>0 – нормальный режим работы |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 1…0 | TXLC[1:0] | Если установлен бит DIRECT_CONTROL_BIT, то отображается состояние шины USB.<br>TX_LINE_STATE[0] = D-<br>TX_LINE_STATE[1] = D+ |

#### 18.9.3.4 *USB_HTXSE*

Таблица 147 – Регистр USB_HTXSE

| Номер | 31…1 | 0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | SOFEN |

Таблица 148 – Описание бит регистра USB_HTXSE

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…1 | - | Зарезервировано |
| 0 | SOFEN | 1 – Если FULL_SPEED_LINE_POLARITY_BIT установлен, то SOF будет автоматически отсылаться каждые 1 мс. SOF отправляется на full speed не зависимо от состояния FULL_SPEED_LINE_RATE_BIT.<br>Если FULL_SPEED_LINE_POLARITY_BIT не установлен, то автоматически будет отправляться EOP каждые 1 мс. Это необходимо при работе с low speed устройством напрямую (не через хаб).<br>0 – запрет автоматической отправки SOF/EOP и позволяет подсоединенным устройствам перейти в suspend режим |

#### 18.9.3.5 *USB_HTXA*

Таблица 149 – Регистр USB_HTXA

| Номер | 31…7 | 6…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | DEVADDR[6:0] |

Таблица 150 – Описание бит регистра USB_HTXA

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…7 | - | Зарезервировано |
| 6…0 | DEVADDR[6:0] | USB Device address. Адрес устройства для обращения |

#### 18.9.3.6 *USB_HTXE*

Таблица 151 – Регистр USB_HTXE

| Номер | 31…4 | 3…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | EPADDR[3:0] |

Таблица 152 – Описание бит регистра USB_HTXE

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | - | Зарезервировано |
| 3…0 | EPADDR[3:0] | Endpoint address. Номер оконченной точки устройства для обращения |

#### 18.9.3.7 *USB_HFN*

**USB_HFN_H**

Таблица 153 – Регистр USB_HFN_H

| Номер | 31…3 | 2…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FNUM HIGH[2:0] |

Таблица 154 – Описание бит регистра USB_HFN_H

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…3 | - | Зарезервировано |
| 2…0 | FNUM HIGH[2:0] | Старшая часть номера фрейма |

**USB_HFN_L**

Таблица 155 – Регистр USB_HFN_L

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

Таблица 156 – Описание бит регистра USB_HFN_L

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7…0 | FNUM LOW[7:0] | Младшая часть номера фрейма |

#### 18.9.3.8 *USB_HIS*

Таблица 157 – Регистр USB_HIS

| Номер | 31…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | SOFS | CONEV | RESUME | TDONE |

Таблица 158 – Описание бит регистра USB_HIS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | - | Зарезервировано |
| 3 | SOFS | 1 – автоматически устанавливается, когда SOF был отправлен. Должен быть очищен записью 1.<br>0 – не было SOF |
| 2 | CONEV | 1 – автоматически устанавливается, когда происходит подсоединение или отсоединение. Должно быть очищено записью 1.<br>0 – события не было |
| 1 | RESUME | 1 – автоматически устанавливается, когда возникает состояние повтора. Должен быть очищен записью 1.<br>0 – не было повтора |
| 0 | TDONE | 1 – автоматически устанавливается, когда передача закончена. Должен быть очищен записью 1.<br>0 – передача не закончена или ее нет |

#### 18.9.3.9 *USB_HIM*

Таблица 159 – Регистр USB_HIM

| Номер | 31…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | SOFS IE | CONEVIE | RESUMEIE | TDONE IE |

Таблица 160 – Описание бит регистра USB_HIM

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | - | Зарезервировано |
| 3 | SOFIE | 1 – разрешение выработки прерывания при передаче SOF;<br>0 – запрещение выработки прерывания |
| 2 | CONEVIE | 1 – разрешение выработки прерывания при подсоединении или отсоединении;<br>0 – запрещение выработки прерывания |

| | | |
|---|---|---|
| 1 | RESUMEIE | 1 – разрешение выработки прерывания при повторе передачи;<br>0 – запрещение выработки прерывания |
| 0 | TDONEIE | 1 – разрешение выработки прерывания при окончании передачи;<br>0 – запрещение выработки прерывания |

#### 18.9.3.10 *USB_HRXS*

Таблица 161 – Регистр USB_HRXS

| Номер | 31…8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | DATASEQ | ACK RXED | STALL RXED | NAK RXED | RX TO | RXOF | BSERR | CRCER |

Таблица 162 – Описание бит регистра USB_HRXS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7 | DATASEQ | Если последняя транзакция была типа IN_TRANS, этот бит указывает номер последнего принятого пакета:<br>DATA0 = 0;<br>DATA1 = 1 |
| 6 | ACK RXED | 1 – получен ACK;<br>0 – не получен ACK |
| 5 | STALL RXED | 1 – получен STALL;<br>0 – не получен STALL |
| 4 | NAK RXED | 1 – получен NAK от устройства;<br>0 – не получен NAK |
| 3 | RXTO | 1 – превышение времени ожидания ответа от устройства;<br>0 – нет превышения времени |
| 2 | RXOF | 1 – обнаружена ошибка переполнения FIFO при приеме пакета;<br>0 – не было переполнения |
| 1 | BSERR | 1 – обнаружена ошибка stuff при последней передаче;<br>0 – ошибки stuff не было |
| 0 | CRCERR | 1 – обнаружена ошибка CRC при последней передаче;<br>0 – ошибки CRC не было |

#### 18.9.3.11 *USB_HRXP*

Таблица 163 – Регистр USB_HRXP

| Номер | 31…4 | 3…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | RPID[3:0] |

Таблица 164 – Описание бит регистра USB_HRXP

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | - | Зарезервировано |
| 3…0 | RPID[3:0] | Packet identifier последнего принятого пакета |

### 18.9.3.12 *USB_HRXA*

Таблица 165 – Регистр USB_HRXA

| Номер | 31…7 | 6…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | RADDR[6:0] |

Таблица 166 – Описание бит регистра USB_HRXA

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…7 | - | Зарезервировано |
| 6…0 | RADDR[6:0] | Адрес устройства, от которого принят последний пакет |

### 18.9.3.13 *USB_HRXE*

Таблица 167 – Регистр USB_HRXE

| Номер | 31…4 | 3…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | RXENDP[3:0] |

Таблица 168 – Описание бит регистра USB_HRXE

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | - | Зарезервировано |
| 3…0 | RXENDP[3:0] | Номер оконечной точки, от которой принят последний пакет |

### 18.9.3.14 *USB_HRXCS*

Таблица 169 – Регистр USB_HRXCS

| Номер | 31…2 | 1…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | RXLS[1:0] |

Таблица 170 – Описание бит регистра USB_HRXCS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…2 | - | Зарезервировано |
| 1…0 | RXLS[1:0] | Состояние линий шины USB:<br>DISCONNECT = 0;<br>LOW_SPEED_CONNECT = 1;<br>FULL_SPEED_CONNECT = 2 |

#### 18.9.3.15 *USB_HSTM*

Таблица 171 – Регистр USB_HSTM

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

Таблица 172 – Описание бит регистра USB_HSTM

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7…0 | HSTM[7:0] | Старший байт таймера SOF, используемого для передачи SOF. Таймер увеличивается на частоте 48 МГц и имеет 48000 тактов в 1 мс фрейме. Этот регистр может быть использован для вычисления времени, оставшегося во фрейме |

#### 18.9.3.16 *USB_HRXFD*

Таблица 173 – Регистр USB_HRXFD

| Номер | 31…8 | 7…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | RX FIFO DATA[7:0] |

Таблица 174 – Описание бит регистра USB_HRXFD

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7…0 | RX FIFO DATA[7:0] | Если последняя транзакция была IN_TRANS, то в буфере содержатся принятые данные, и они могут быть считаны через этот регистр |

#### 18.9.3.17 *USB_HRXDC*

Таблица 175 – Регистр USB_HRXDC

| Номер | 31…16 | 15..0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FIFO DATA COUNT[15:0] |

Таблица 176 – Описание бит регистра USB_HRXDC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…0 | FIFO DATA COUNT[15:0] | Счетчик байтов, записанных в буфер |

#### 18.9.3.18 *USB_HRXFC*

Таблица 177 – Регистр USB_HRXFC

| Номер | 31…1 | 0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FIFO FORCE EMPTY |

Таблица 178 – Описание бит регистра USB_HRXFC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…1 | - | Зарезервировано |
| 0 | FIFO FORCE EMPTY | Запись 1 принудительно сбрасывает указатель FIFO |

#### 18.9.3.19 *USB_HTXFD*

Таблица 179 – Регистр USB_HTXFD

| Номер | 31…8 | 7..0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | TX FIFO DATA[7:0] |

Таблица 180 – Описание бит регистра USB_HTXFD

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7…0 | TX FIFO DATA[7:0] | При запросах передачи OUTDATA0_TRANS или OUTDATA1_TRANS, через данный регистр должны быть загружены данные для отправки |

#### 18.9.3.20 *USB_HTXFDC*

Таблица 181 – Регистр USB_HTXFDC

| Номер | 31…1 | 0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FIFO FORCE EMPTY |

Таблица 182 – Описание бит регистра USB_HTXFDC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…1 | - | Зарезервировано |
| 0 | FIFO FORCE EMPTY | Запись 1 принудительно сбрасывает указатель FIFO |

### 18.9.4 USB Slave (Device)

#### 18.9.4.1 *USB_SEPx.CTRL*

Таблица 183 – Регистр SEP[x].CTRL

| 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 |
| | - | EPISOEN | EPSSTALL | EPDATASEQ | EPRDY | EPEN |

Таблица 184 – Описание бит регистра USB_SEPx.CTRL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…5 | - | Зарезервировано |
| 4 | EPISOEN | 1 – изохронный режим передачи;<br>0 – не изохронный режим передачи.<br>В изохронном режиме не отсылаются какие-либо подтверждения передачи |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 3 | EPSSTALL | 1 – если точка разрешена, готова и не в изохронном режиме, то на запрос хоста будет отвечен STALL;<br>0 – не отвечать STALL |
| 2 | EPDATASEQ | 1 – отвечать на IN запрос от хоста с DATA1;<br>0 – отвечать на IN запрос от хоста с DATA0 |
| 1 | EPRDY | 1 – оконечная точка готова;<br>0 – оконечная точка не готова или закончила передачу.<br>Если точка разрешена и готова, то она может ответить на инициализированную хостом передачу. Бит автоматически сбрасывается в 0 после успешного окончания передачи |
| 0 | EPEN | 1 – оконечная точка разрешена;<br>0 – оконечная точка запрещена.<br>Если точка запрещена, она не отвечает на транзакции. Если точка разрешена, но не готова и не находится в изохронном режиме, то она отвечает NAK |

#### 18.9.4.2 *USB_SEPx.STS*

Таблица 185 – Регистр SEP[x].STS

| Номер | 31…8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | SC DATA SEQ | SC ACK RXED | SC STALL SENT | NAK SENT | SC RXTO | SC RXOF | SC BS ERR | SC CRC ERR |

Таблица 186 – Описание бит регистра USB_SEPx.STS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7 | SC DATA SEQ | Если предыдущий тип передачи был OUT_TRANS, то этот бит определяет тип принятого пакета:<br>DATA0 = 0;<br>DATA1= 1 |
| 6 | SC ACK RXED | 1 – получен ACK от хоста на переданные данные;<br>0 – нет подтверждения |
| 5 | SC STALL SENT | 1 – обозначает отправку STALL;<br>0 – не было STALL |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 4 | SC<br>NAK<br>SENT | 1 – обозначает отправку NAK ответа;<br>0 – не было NAK |
| 3 | SC<br>RXTO | 1 – обозначает возникновение ошибки времени ожидания ответа от хоста;<br>0 – нет ошибки |
| 2 | SC<br>RXOF | 1 – обозначает возникновение переполнения буфера FIFO при приеме последнего пакета;<br>0 – нет переполнения |
| 1 | SC<br>BS<br>ERR | 1 – обозначает возникновение STUFF ошибки в последней передаче;<br>0 – нет ошибки |
| 0 | SC<br>CRC<br>ERR | 1 – обозначает возникновение CRC ошибки в последней передаче;<br>0 – нет ошибки |

#### 18.9.4.3 *USB_SEPx.TS*

Таблица 187 – Регистр SEP[x].TS

| Номер | 31…2 | 1 | 0 |
|---|---|---|---|
| Доступ | U | R/W | R/W |
| Сброс | 0 | 0 | 0 |
| | - | SCTTYPE[1:0] | |

Таблица 188 – Описание бит регистра SEP[x].TS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…2 | - | Зарезервировано |
| 1…0 | SCTTYPE[1:0] | Отображает тип последней передачи, перед тем как ENDPOINT_READY_BIT был изменен с 1 на 0:<br>SC_SETUP_TRANS = 0;<br>SC_IN_TRANS = 1;<br>SC_OUTDATA_TRANS = 2 |

#### 18.9.4.4 *USB_SEPx.NTS*

Таблица 189 – Регистр SEP[x].NTS

| Номер | 31…2 | 1 | 0 |
|---|---|---|---|
| Доступ | U | R/W | R/W |
| Сброс | 0 | 0 | 0 |
| | - | NTTYPE[1:0] | |

Таблица 190 – Описание бит регистра USB_SEPx.NTS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…2 | - | Зарезервировано |
| 1…0 | NTTYPE[1:0] | Тип последней передачи, в результате которой на хост был послан NAK:<br>SC_SETUP_TRANS = 0;<br>SC_IN_TRANS = 1;<br>SC_OUTDATA_TRANS = 2 |

#### 18.9.4.5 *USB_SC*

Таблица 191 – Регистр USB_SC

| Номер | 31…6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | SCFSR | SCFSP | SCDC | SCTXLS[1:0] | | SCGEN |

Таблица 192 – Описание бит регистра USB_SC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…6 | - | Зарезервировано |
| 5 | SCFSR | Флаг управления скоростью работы:<br>1 – 12 Мбит/с;<br>0 – 1,5 Мбит/с |
| 4 | SCFSP | Флаг выбора полярности линий USB шины:<br>1 – FULL SPEED;<br>0 – LOW SPEED |
| 3 | SCDC | Флаг прямого управления линиями USB шины:<br>1 – разрешено прямое управление;<br>0 – запрещено прямое управление |
| 2…1 | SCTXL[1:0] | Если установлен бит SC_DIRECT_CONTROL_BIT, то через SC_TX_LINE_STATE осуществляется прямое управление состоянием линий USB шины:<br>SC_TX_LINE_STATE [1] = D+;<br>SC_TX_LINE_STATE [0] = D- |
| 0 | SCGEN | 1 – разрешение для работы с разрешенных оконечных точек;<br>0 – все оконечные точки запрещены |

#### 18.9.4.6 *USB_SLS*

Таблица 193 – Регистр SLS

| Номер | 31…2 | 1 | 0 |
|---|---|---|---|
| Доступ | U | R/W | R/W |
| Сброс | 0 | 0 | 0 |
| | - | SCRXLS[1:0] | |

Таблица 194 – Описание бит регистра SLS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…2 | - | Зарезервировано |
| 1…0 | SCRXLS[1:0] | Отображает состояние подсоединения на шине USB:<br>RESET = 0;<br>LOW_SPEED_CONNECT = 1;<br>FULL_SPEED_CONNECT = 2 |

#### 18.9.4.7 ***USB_SIS***

Таблица 195 – Регистр SIS

| Номер | 31…6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Доступ | U | U | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | - | - | SC<br>NAK<br>SENT | SC<br>SOF<br>REC | SC<br>RESET<br>EV | SC<br>RESUME | SC<br>TDONE |

Таблица 196 – Описание бит регистра USB_SIS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…6 | - | Зарезервировано |
| 5 | - | Зарезервировано |
| 4 | SC<br>NAK<br>SENT | Устанавливается в 1, когда отвечен NAK. Очищается записью 1 |
| 3 | SC<br>SOF<br>REC | Устанавливается в 1, когда принят пакет SOF.<br>Очищается записью 1 |
| 2 | SC<br>RESET<br>EV | Устанавливается в 1, когда обнаруживается состояние сброса на шине USB. Очищается записью 1 |
| 1 | SC<br>RESUME | Устанавливается в 1, когда обнаруживается состояние повтора. Очищается записью 1 |
| 0 | SC<br>TDONE | Устанавливается в 1 после успешного выполнения передачи. Очищается записью 1 |

#### 18.9.4.8 *USB_SIM*

Таблица 197 – Регистр SIM

| 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 |
| | - | SC<br>NAK<br>SENT<br>IE | SC<br>SOF<br>RECIE | SC<br>RESET<br>EVIE | SC<br>RESUME<br>IE | SC<br>TDONE<br>IE |

Таблица 198 – Описание бит регистра B_SIM

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…5 | - | Зарезервировано |
| 4 | SC<br>NAK<br>SENT<br>IE | Бит разрешения прерывания при отправке NAK:<br>1 – разрешено прерывание;<br>0 – запрещено прерывание |
| 3 | SC<br>SOF<br>RECIE | Бит разрешения прерывания при приеме SOF:<br>1 – разрешено прерывание;<br>0 – запрещено прерывание |
| 2 | SC<br>RESET<br>EVIE | Бит разрешения прерывания при состоянии сброса на шине:<br>1 – разрешено прерывание;<br>0 – запрещено прерывание |
| 1 | SC<br>RESUME<br>IE | Бит разрешения прерывания при состоянии повтора:<br>1 – разрешено прерывание;<br>0 – запрещено прерывание |
| 0 | SC<br>TDONE<br>IE | Бит разрешения прерывания при окончании передачи:<br>1 – разрешено прерывание;<br>0 – запрещено прерывание |

#### 18.9.4.9 *USB_SA*

Таблица 199 – Регистр SA

| Номер | 31…7 | 6…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | SDEVADDR[6:0] |

Таблица 200 – Описание бит регистра SA

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…7 | - | Зарезервировано |
| 6…0 | SDEVADDR[6:0] | Функциональный адрес USB Device |

#### 18.9.4.10 *USB_SFN*

**USB_SFN_H**

Таблица 201 – Регистр SFN_H

| Номер | 31…3 | 2…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FRAME<br>NUM HIGH[2:0] |

Таблица 202 – Описание бит регистра SFN_H

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…3 | - | Зарезервировано |
| 2…0 | FRAME<br>NUM HIGH[2:0] | Старшая часть номера фрейма, принятого в последнем SOF |

**USB_SFN_L**

Таблица 203 – Регистр SFN_L

| Номер | 31…8 | 7…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FRAME<br>NUM LOW[7:0] |

Таблица 204 – Описание бит регистра SFN_H

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7…0 | FRAME<br>NUM LOW[7:0] | Младшая часть номера фрейма, принятого в последнем SOF |

#### 18.9.4.11 *USB_SEPx.RXFD*

Таблица 205 – Регистр SEP[x].RXFD

| Номер | 31…8 | 7…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | RX<br>FIFO<br>DATA[7:0] |

Таблица 206 – Описание бит регистра SEP[x].RXFD

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7…0 | RX FIFO DATA[7:0] | После приема OUTDATA_TRANS или SETUP_TRANS пакета, принятые данные читаются из регистра RX_FIFO_DATA |

### 18.9.4.12 *USB_SEPx.RXFDC*

Таблица 207 – Регистр SEP[x].RXFDC

| Номер | 31…16 | 15…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FIFO DATA COUNT[15:0] |

Таблица 208 – Описание бит регистра SEP[x].RXFDC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…0 | FIFO DATA COUNT[15:0] | Отображает число байт, записанных в буфер FIFO |

### 18.9.4.13 *USB_SEPx.RXFC*

Таблица 209 – Регистр SEP[x].RXFC

| Номер | 31…1 | 0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FIFO FORCE EMPTY |

Таблица 210 – Описание бит регистра SEP[x].RXFC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…1 | - | Зарезервировано |
| 0 | FIFO FORCE EMPTY | Запись 1 очищает указатель буфера FIFO |

#### 18.9.4.14 *USB_SEPx.TXFD*

Таблица 211 – Регистр SEP[x].TXFD

| Номер | 31…8 | 7…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | TX<br>FIFO<br>DATA[7:0] |

Таблица 212 – Описание бит регистра SEP[x].TXFD

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7...0 | TX<br>FIFO<br>DATA [7:0] | Перед приемом IN_TRANS в буфер FIFO записываются данные для отправки |

#### 18.9.4.15 *USB_SEPx.TXFDC*

Таблица 213 – Регистр SEP[x].TXFDC

| Номер | 31…1 | 0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | FIFO<br>FORCE<br>EMPTY |

Таблица 214 – Описание бит регистра SEP[x].TXFDC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…1 | - | Зарезервировано |
| 0 | FIFO<br>FORCE<br>EMPTY | Запись 1 очищает указатель буфера FIFO |

# 19 Контроллер интерфейса CAN

В микросхеме реализовано два независимых цифровых контроллера интерфейса CAN. Они являются полнофункциональными узлами CAN, отвечающими требованиям к активным и пассивным устройствам CAN 2.0A и 2.0B и поддерживающими передачу данных на скорости до 1 Мбит/с.

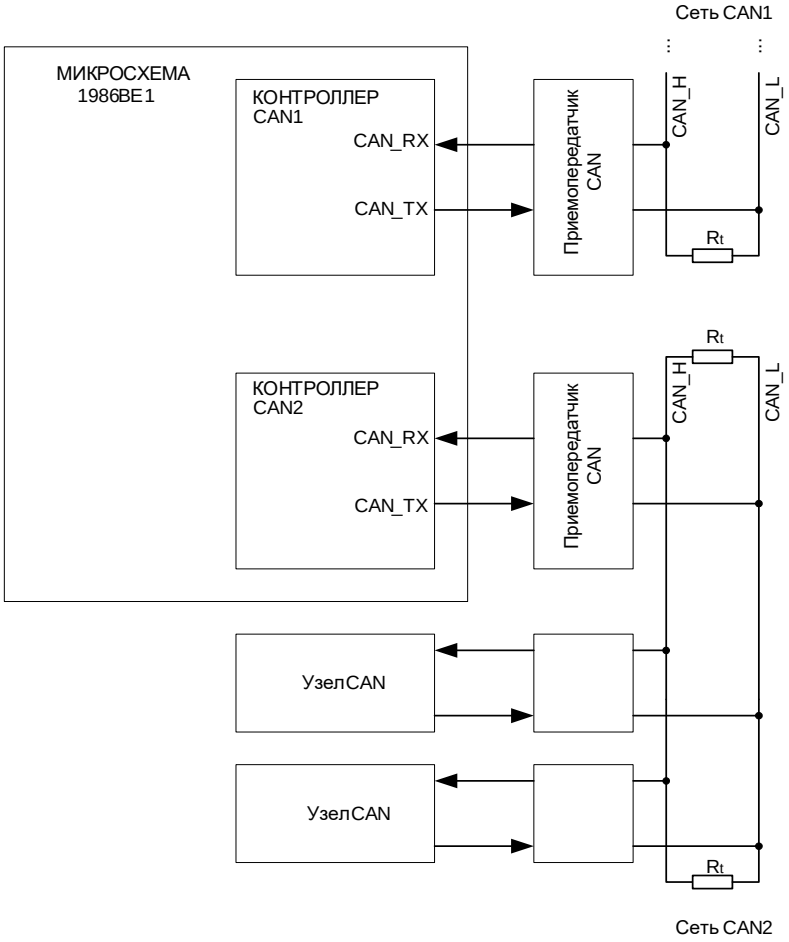



Рисунок 36 – Структурная блок – схема организации сети CAN

Интерфейс CAN позволяет обмениваться сообщениями в сети равноправных устройств. При передаче сообщения в сети CAN все узлы сети получают это сообщение. В сообщении передается уникальный идентификатор узла и данные. Все сообщения в протоколе CAN довольно короткие и могут содержать не более восьми байт данных. При возникновении коллизий (одновременная передача сообщений различными узлами) при передаче идентификатора происходит арбитраж, и узел с большим номером идентификатора уступает сеть узлу с меньшим номером идентификатора.

Особенности:

- поддержка CAN протокола версии CAN 2.0 A и B;
- скорость передачи до 1 Мбит/с;
- 32 буфера приема/передачи;
- поддержка приоритетов сообщений;
- 32 фильтра приема;
- маскирование прерываний.

## 19.1 Режимы работы

Контроллер CAN поддерживает несколько режимов работы: нормальный режим для приема и передачи пакетов сообщений, режим работы только на прием, режим самотестирования и режим инициализации для задания параметров связи.

### 19.1.1 Нормальной режим

Нормальный режим выбирается установкой в регистре CAN_CONTROL битов ROM = 0, STM = 0.

Выводы CAN_TX и CAN_RX подключены к шине.

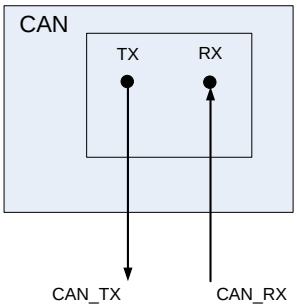



Рисунок 37 – Режим нормальной передачи

В этом режиме можно установить флаги разрешения приема своих пакетов и разрешения подтверждения своих пакетов посылкой ACK (регистр CAN_CONTROL биты SAP и ROP).

### 19.1.2 Режим работы только на прием (Receive Only Mode)

Режим работы только на прием выбирается установкой в регистре CAN_CONTROL битов ROM = 1, STM = 0.

Контроллер интерфейса CAN принимает, но не посылает никакой информации, т.е. линия TX всегда в «1», но внутри контроллера все управляющие сигналы проходят.

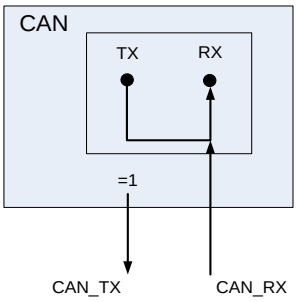



Рисунок 38 – Режим работы только на прием (Receive Only Mode)

### 19.1.3 Режим самотестирования (Self Test Mode)

Режим самотестирования выбирается установкой в регистре CAN_CONTROL битов STM = 1, ROM = 0.

Выводы CAN_TX и CAN_RX отключены, вся передаваемая информация видна только внутри контроллера.

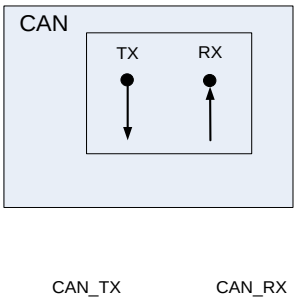



Рисунок 39 – Режим самотестирования (Self Test Mode)

Для успешного приема своих сообщений необходимо установить флаги разрешения приема своих пакетов и разрешения подтверждения своих пакетов посылкой ACK (регистр CAN_CONTROL поля SAP и ROP). В этом режиме передаваемые сообщения сразу же принимаются в приемный буфер. Режим самотестирования полезен в период отладки кода программы.

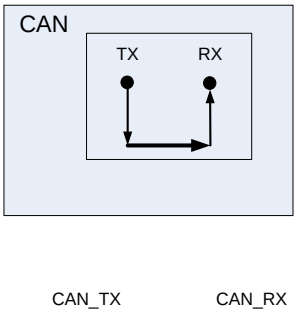



Рисунок 40 – Режим инициализации для задания параметров связи

Еще одна важная функция контроллера CAN – фильтрация получаемых сообщений. Поскольку CAN является широковещательной шиной каждое переданное сообщение принимается всеми узлами шины. В шине CAN любой разумной степени сложности передается достаточно большое число сообщений. Задачей каждого подключенного к узлу CAN ЦПУ является реагирование на сообщения CAN. Таким образом, чтобы избавить контроллер CAN от проблемы приема в буфер нежелательных сообщений, необходима их фильтрация. У контроллера CAN микросхемы имеется 32 регистра фильтров и 32 регистра масок, которые можно использовать для блокировки всех сообщений CAN, кроме избранных сообщений или групп сообщений.

## 19.2 Типы пакетов сообщений

Информация на шине представлена в виде фиксированных сообщений различной, но ограниченной длины. Когда шина свободна, любой подключенный узел может начать передавать новое сообщение. При передаче информации с помощью протокола CAN используется четыре типа пакетов:

– **пакет удаленного запроса данных** передается узлом, чтобы запросить передачу пакета данных с тем же самым идентификатором;

– **пакет ошибки** передается любым узлом при обнаружении ошибочного состояния на шине. Пакет ошибки передается сразу же после обнаружения ошибки и накладывается на передаваемый пакет так, чтобы испортить его окончательно. Таким

образом, если один из узлов обнаружил ошибку, он усиливает ошибку для того, чтобы ее обнаружили и другие узлы;

– **пакет перегрузки** используется для обеспечения дополнительной задержки между предшествующим и последующим кадрами данных или кадрами удаленного запроса данных. Он передается в редких случаях, подробнее можно прочесть в стандарте ISO 11898-1. Контроллер интерфейса CAN отсылает пакет перегрузки в соответствии со стандартом;

– основными пакетами на шине CAN являются **пакеты данных**. Пакет данных передает данные от передатчика приемнику. Пакеты могут быть стандартными и расширенными. Отличие пакетов заключается в размере полей идентификатора. Пакеты с 11-разрядным идентификатором – называются стандартными пакетами, пакеты, содержащие 29 разрядные идентификаторы, называются расширенными пакетами. При передаче идентификационной информации происходит автоматический арбитраж на шине CAN таким образом, чтобы пакет с меньшим значением поля ID остался на шине. На шине не допускается наличие двух или более узлов с одним и тем же идентификатором. Размер передаваемых данных кодируется в поле DLC и может составлять от 0 до 8 байт. После передачи поля данных контроллер автоматически передает рассчитанное значение CRC. Если хотя бы один из узлов принял пакет, то он выставляет ACK подтверждение на шине, если хотя бы один из узлов обнаружит ошибку, то на шину будет выставлен пакет ошибки. Таким образом, обеспечивается гарантированность доставки сообщений.

Пакеты данных и пакеты удаленного запроса данных отделяются от предшествующих пакетов межкадровым пространством.

## 19.3 Структура пакета данных (Data Frame)

Пакет данных состоит из семи различных полей:

– «начало пакета» (SOF-start of frame);
– «поле арбитража» (arbitration field);
– «поле контроля» (control field);
– «поле данных» (data field);
– «поле CRC» (CRC field);
– «поле подтверждения» (ACK field);
– «конец пакета» (end of frame).

Поле данных может иметь нулевую длину.

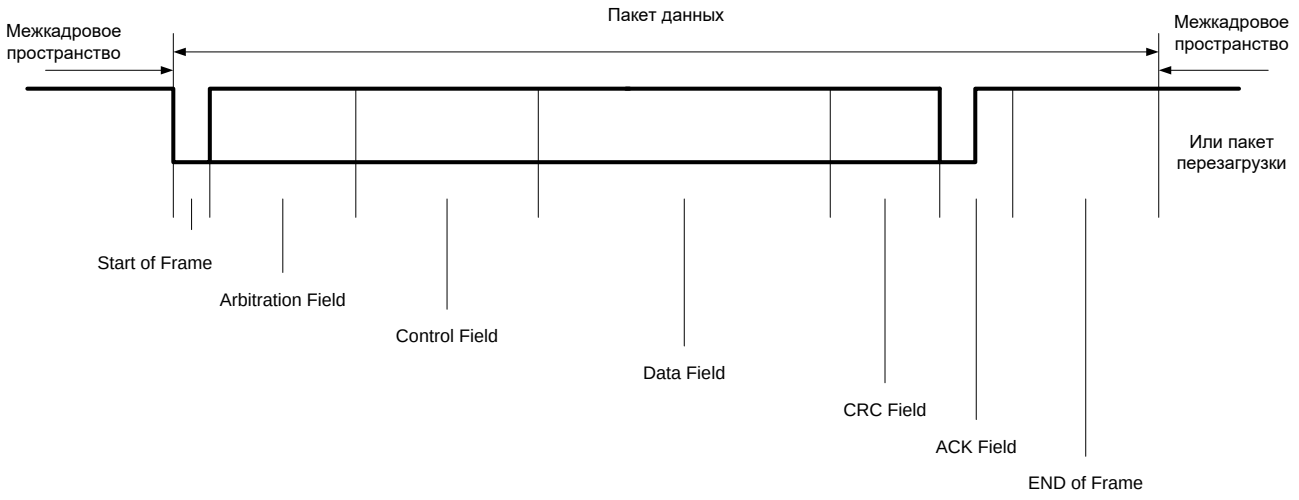



Рисунок 41 – Пакет сообщения CAN

В терминах протокола CAN логическая единица называется рецессивным битом, а логический ноль называется доминантным битом. Во всех случаях доминантный бит будет затирать рецессивный. То есть, если несколько узлов выставят на шину рецессивный бит, а один – доминантный, то обратно всеми узлами будет считан доминантный бит.

### 19.3.1 Начало пакета (Start of frame)

Начало пакета отмечает начало пакета данных или пакета удаленного запроса данных. Это поле состоит из одиночного доминантного бита. Узлу разрешено начать передачу, когда шина свободна. Все узлы должны синхронизироваться по фронту, вызванному передачей поля «начало пакета» узла, начавшего передачу первым.

### 19.3.2 Поле арбитража (Arbitration field)

Формат поля арбитража отличается для стандартного и расширенного форматов:

– в стандартном формате поле арбитража состоит из 11-разрядного идентификатора и RTR-бита.

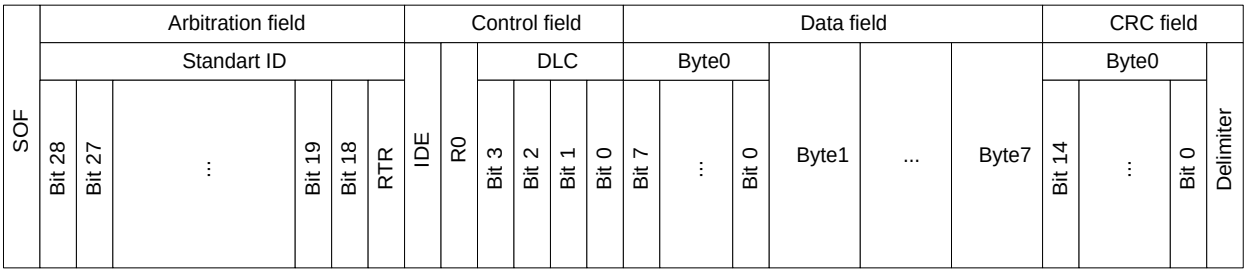



Рисунок 42 – Структура стандартного пакета данных

– в расширенном формате поле арбитража состоит из 29-разрядного идентификатора, SRR-бита, IDE-бита и RTR-бита.

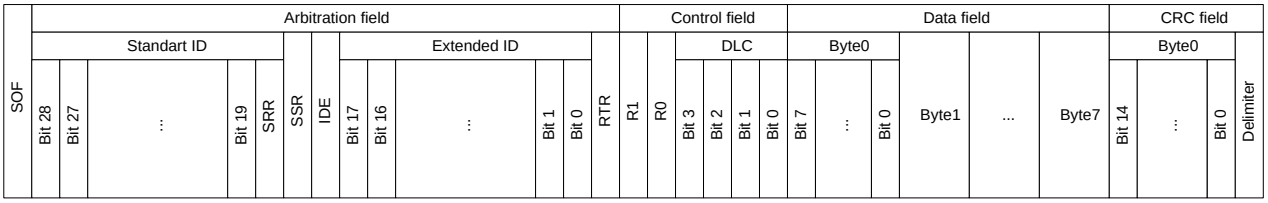



Рисунок 43 – Структура расширенного пакета данных

Идентификатор

Идентификатор – стандартный формат. Длина идентификатора – 11 бит и соответствует Standart ID в расширенном формате. Эти биты передаются в порядке Bit28 … Bit18. Самый младший бит – Bit18. Семь старших битов (Bit28 – Bit 22) не должны быть все единичными битами.

Идентификатор – расширенный формат. В отличие от стандартного идентификатора, расширенный идентификатор состоит из 29 бит. Его формат содержит две секции:

- Standart ID – 11 бит;
- Extended ID – 18 бит.

Standart ID состоит из 11 бит. Эта секция передается в порядке от Bit28 … Bit18. Это эквивалентно формату стандартного идентификатора. Standart ID определяет базовый приоритет расширенного пакета.

Extended ID состоит из 18 бит. Эта секция передается в порядке от Bit17 до Bit0. В стандартном пакете идентификатор сопровождается битом RTR.

Бит RTR

Бит запроса удаленной передачи. В пакетах данных RTR бит должен быть передан нулевым уровнем. Внутри пакета удаленного запроса данных RTR бит должен быть единичным. В расширенном пакете сначала передается Standart ID, с последующими битами IDE и SRR. Extended ID передается после SRR бита.

Бит SRR (расширенный формат)

Заменитель бита удаленного запроса. SRR – единичный бит. Он передается в расширенных пакетах в позиции RTR бита. Таким образом, он заменяет RTR – бит стандартного пакета.

Следовательно, при одновременной передаче стандартного пакета и расширенного пакета, Standart ID которого совпадает с идентификатором стандартного пакета, стандартный пакет преобладает над расширенным пакетом.

Бит IDE (расширенный формат)

Бит расширения идентификатора

Бит IDE принадлежит:

- полю арбитража для расширенного формата;
- полю управления для стандартного формата.

Бит IDE в стандартном формате передается нулевым уровнем, в расширенном формате бит IDE – единичный уровень.

### 19.3.3 Поле управления (Control field)

Поле управления состоит из шести битов. Формат поля управления отличается для стандартного и расширенного формата.

Пакеты в стандартном формате включают: код длины данных (DLC), бит IDE, который передается нулевым уровнем (см. выше), и зарезервированный бит r0.

Пакеты в расширенном формате включают код длины данных и два зарезервированных бита r1 и r0. Зарезервированные биты должны быть посланы нулевым уровнем, но приемники принимают единичные и нулевые уровни биты во всех комбинациях.

Код длины данных (Data length code)

Число байт в поле данных обозначается кодом длины данных. Этот код длины данных, размером четыре бита, передается внутри поля управления. Допустимое число байт данных: {0,1, …., 7,8}. Другие величины использоваться не могут.

### 19.3.4 Поле данных (Data field)

Поле данных состоит из данных, которые будут переданы внутри пакета данных. Оно может содержать от нуля до восьми байт, каждый содержит восемь бит, которые передаются, начиная со старшего значащего бита.

### 19.3.5 Поле CRC (CRC field)

Содержит последовательность CRC и CRC – разделитель. При вычислении 15-битного CRC кода используется последовательность бит, состоящая из полей: «начало пакета», «поле арбитража», «управляющее поле», «поле данных» (если есть). Последовательность CRC сопровождается разделителем CRC, который состоит из одного единичного бита.

### 19.3.6 Поле подтверждения (ACK field)

Поле подтверждения имеет длину два бита и содержит: «область подтверждения» и разделитель подтверждения. В поле подтверждения передающий узел посылает два бита с единичным уровнем. Приемник, который получил сообщение правильно (CRC соответствует), сообщает об этом передатчику, посылая бит с нулевым уровнем в течение приема поля «область подтверждения».

### 19.3.7 Конец пакета (End of frame)

Каждый пакет данных и пакет удаленного запроса данных ограничен последовательностью флагов, состоящей из семи единичных бит.

### 19.3.8 Структура пакета удаленного запроса данных (Remote frame)

Узел, действующий как приемник некоторых данных, может инициировать передачу соответствующих данных узлами-источниками, посылая пакет удаленного запроса данных. Пакет удаленного запроса данных существует и в стандартном формате, и в расширенном формате. В обоих случаях он состоит из шести битовых полей:

– «начало пакета» (Start of frame);

- «поле арбитража» (Arbitration field);
- «управляющее поле» (Control field);
- «поле CRC» (CRC – field);
- «поле подтверждения» (ACK field);
- «конец пакета» (End of frame).

В отличие от обычного пакета данных, RTR бит пакета удаленного запроса данных – единичный. В этом пакете отсутствует поле данных. При этом значение кода длины данных может принимать любое значение в пределах допустимого диапазона от 0 до 8. Значение кода длины данных соответствует коду длины данных кадра данных. Бит RTR указывает, является ли переданный кадр кадром данных.

### 19.3.9 Арбитраж на шине

Арбитраж сообщений гарантирует, что наиболее важное сообщение захватит шину и будет передано без задержки. Затем будут переданы приостановленные сообщения согласно их приоритетам: сообщение с наименьшим идентификатором передается первым.

Если планируется передача сообщения, и шина свободна, то сообщение будет передано и сможет быть принято любым заинтересованным в нем узлом. Если передача сообщения запланирована, а шина активна, то прежде чем приступить к передаче сообщения, необходимо дождаться освобождения шины. Если запланирована передача нескольких сообщений, то при освобождении шины они начнут передаваться одновременно, синхронизируясь по признаку начала пакета. В этом случае на шине начнется процесс арбитража, задача которого – определить, какое именно из сообщений захватит шину и будет передано.

Арбитраж сообщений на шине CAN осуществляется методом, который называется «неразрушающий побитовый арбитраж».

На рисунке 44 изображены три сообщения, ожидающие передачи. После освобождения шины и синхронизации пакетов сообщений по старт-биту на шину начинают выдаваться все три идентификатора. При передаче первых двух битов все три узла выставляют на шину одинаковые логические уровни и соответственно считывают те же значения, поэтому они все продолжают передачу. Однако при передаче третьего бита узлы A и C выставляют на шину доминантный бит, а узел B выставляет рецессивный бит, но при этом считывает с шины доминантный. В результате узел B освобождает шину и начинает следить за ее состоянием. Узлы A и C продолжают передачу, пока ситуация не повторится; теперь узел C выдает рецессивный бит, а узел A – доминантный. При этом узел C прекращает передачу и начинает следить за состоянием шины. С этого момента шина захватывается узлом A. После передачи сообщения узлом A узлы B и C начинают передачу, причем узел C захватит шину и передает свое сообщение. Если бы узлу A снова надо было передавать сообщение, он снова захватил бы шину. Таким образом, первым на шине CAN передается сообщение с наименьшим идентификатором.

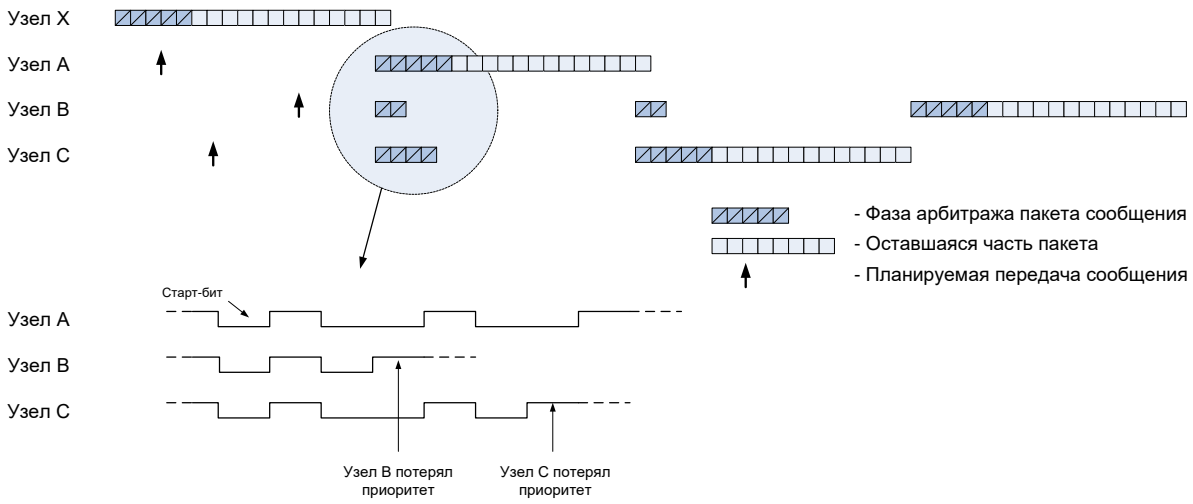



Рисунок 44 – Арбитраж на шине CAN

В случае «проигрыша» арбитража в регистре статуса контроллера CAN будет установлен флаг ID_LOWER.

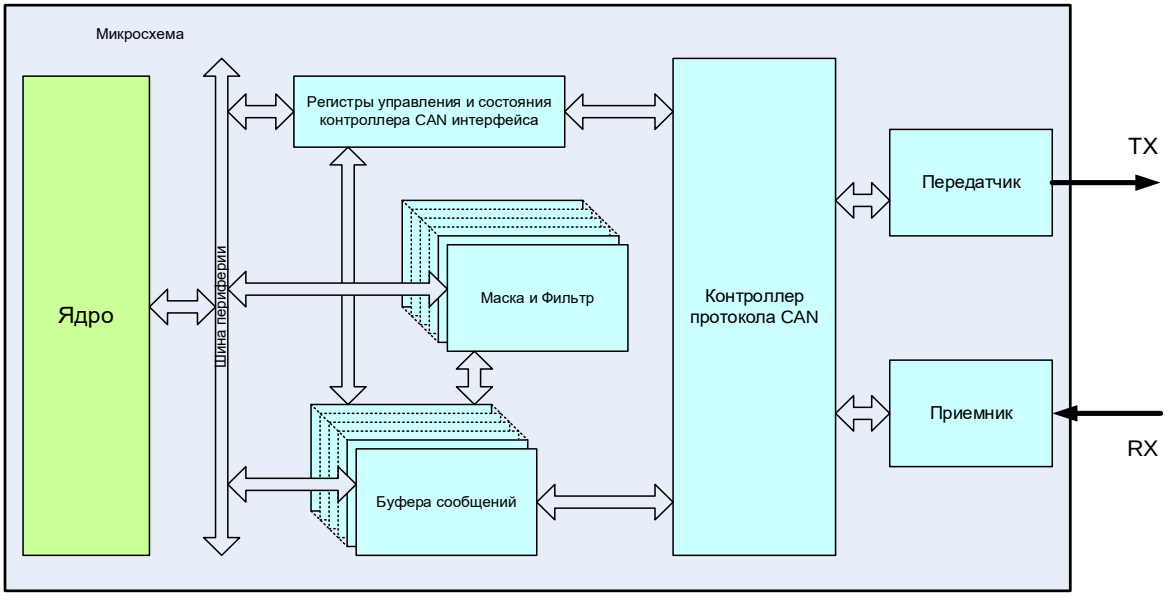



Рисунок 45 – Структурная блок-схема контроллера CAN

## 19.4 Инициализация

Перед началом работы с контроллерами CAN в первую очередь должны быть заданы параметры их тактового сигнала. Параметры задаются в блоке «Сигналы тактовой частоты».

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 0 для CAN1, бит 1 для CAN2 регистра PER_CLOCK). В регистре CAN_CLOCK установить бит CANyCLKEN, чтобы разрешить тактовую частоту для определенного контроллера CAN, задать коэффициент деления тактовой частоты HCLK для каждого контроллера CAN.

После подачи тактового сигнала на контроллер CAN, можно приступать к работе с ним.

Для работы контроллера шины CAN, он должен быть настроен на соответствующую скорость шины CAN. Для этого должны быть заданы соответствующим

образом поля SB, SJW, SEG2, SEG1, PSEG и BRP в регистре CAN_BITTMNG. После этого должны быть заданы работающие буфера сообщений путем задания битов EN (разрешение работы) RXTXn (1 – прием, 0 – передача) в регистре BUF_xx_CON. После этого должен быть выдан общий сигнал разрешения работы контроллера через задание бита CANEN в регистре CONTROL. После этого контроллер CAN начинает работу.

## 19.5 Передача сообщений

Для передачи сообщения необходимо в разрешенный для работы и конфигурируемый на передачу буфер записать сообщение для передачи (задать значения регистрам CAN_BUF[x].ID, CAN_BUF[x].DLC, CAN_BUF[x].DATAL и CAN_BUF[x].DATAH), после чего установить бит TX_REQ. После установки этого бита сообщение будет поставлено в очередь на отправку. После отправки сообщения бит TX_REQ будет автоматически сброшен. Если в нескольких буферах есть сообщения на отправку, то порядок отправки определяется по полю PRIOR_0. Если у сообщения бит PRIOR_0 выставлен в ноль, то оно отправляется в первую очередь. Если есть несколько сообщений с одинаковым приоритетом, то порядок отправки определяется порядковым номером буфера, буфер с меньшим порядковым номером имеет больший приоритет. Значение полей ID для выбора порядка отправки в рамках контроллера CAN (одного узла) значения не имеет. По ID выбирается приоритет между различными узлами.

## 19.6 Передача сообщений по Remote Transmit Request (RTR)

Для автоматической отправки сообщения по запросу Remote Transmit Request необходимо задать режим маскирования для данного буфера таким образом, чтобы он принимал только сообщения от устройства, которое может выслать запрос RTR. В регистре INT_TX при необходимости настроить генерацию прерывания передачи для соответствующего буфера. В регистре управления этим буфером (BUFF_CON[x]) проверить, что флаг TX_REQ = 0, задать приоритет отправляемого сообщения PRIOR_0, установить разрешение ответа при приеме RTR в буфер (RTR_EN=1), задать RX_TX = 0 для разрешения отправки сообщения и задать EN = 1 для разрешения работы буфера. В регистре идентификации задать необходимые SID и EID, в регистре BUF_xx_DLC указать формат пакета (расширенный или стандартный) и указать длину передаваемых данных в поле DLC. В регистрах данных CAN_BUF[x].DATAL и CAN_BUF[x].DATAH задать необходимые для отправки данные. Далее можно переходить к выполнению остальной части программы с отправкой сообщений CAN. Отправка сообщения буфером будет произведена по запросу RTR, удовлетворяющему механизму фильтрации для принимаемых сообщений, который выбран для данного буфера. Если запрос RTR удовлетворяет механизму фильтрации нескольких буферов, то ответы будут отправлены из каждого из этих буферов.

## 19.7 Прием сообщений

Для приема сообщений необходимо иметь свободные и разрешенные для работы буфера, сконфигурированные на прием сообщений. При этом если по шине CAN будут передаваться сообщения от других узлов, они будут сохраняться в этих буферах.

### 19.8 Автоматическая фильтрация принимаемых сообщений

Для уменьшения затрат процессорного ядра на обработку принимаемых сообщений, контроллер интерфейса CAN может автоматически фильтровать принимаемые сообщения. Для каждого буфера могут быть заданы маска (CAN_BUF_FILTER[x].MASK) и фильтр (CAN_BUF_FILTER[x].FILTER) таким образом, что в этот буфер будут приниматься только те сообщения, для которых выполняется условие:

ID & CAN_BUF_FILTER[x].MASK == CAN_BUF_FILTER[x].FILTER

Если принимаемое сообщение не может быть помещено ни в один из буферов, то оно будет проигнорировано. Если сообщение может быть принято более чем одним буфером, то оно будет помещено в буфер с меньшим порядковым номером. При инициализации после включения питания или сброса CAN_BUF_FILTER[x].MASK и CAN_BUF_FILTER[x].FILTER для всех буферов имеют произвольное значение, таким образом, необходимо перед началом работы их проинициализировать. Для приема всех сообщений без фильтрации необходимо задать им нулевое значение. Специального бита для включения или выключения фильтрации нет.

### 19.9 Перезапись принятых сообщений

В буфере может быть включено разрешение перезаписи принятого сообщения. Если принимаемое сообщение не может быть сохранено в свободный буфер, то оно может быть сохранено в буфер с ранее полученным сообщением, если для него выставлен бит OVER_EN. При этом выставляется флаг OVER_WR. Таким образом, если у буфера разрешена перезапись принятых сообщений, после прочтения сообщения необходимо проверить флаг OVER_WR. Если он выставлен в «1», то необходимо сбросить OVER_WR (не сбрасывая флаг RX_FULL), затем еще раз прочесть сообщение, после чего снова проверить флаг OVER_WR и, если он не выставлен повторно, то сбросить флаг RX_FULL. И считанное значение считать корректным.

Прибегать к помощи механизма перезаписи принятых сообщений можно только в случае, когда допустима потеря сообщений, работа с перезаписью сообщений не гарантирует прием всех сообщений, а только позволяет принять сообщение корректно, так как момент чтения сообщения может совпасть с моментом сохранения нового сообщения. При этом первая часть считанного процессорным ядром сообщения будет от первого сообщения, вторая от второго. Если же между сбросом флага OVER_WR, чтением сообщения и при следующей проверке OVER_WR он оказался не выставлен, это означает, что в момент чтения сообщения из буфера в него не сохранялось новое сообщение.

### 19.10 Задание скорости передачи и момента семплирования

Все узлы шины CAN должны работать на одной скорости. Протокол CAN использует кодирование без возврата в ноль (NRZ). Также при передаче не передаются тактовые сигналы. Таким образом, приемники должны засинхронизироваться под тактовый сигнал передатчика. Поскольку все узлы имеют свои индивидуальные тактовые генераторы, все приемники имеют специальный блок синхронизации DPLL.

Максимальная скорость передачи CAN 1 Мбит/сек. Время битового интервала Nominal Bit Time определяется как

$$TBIT = \frac{1}{\text{Скорость передачи}}. \quad (1)$$

Блок DPLL разбивает битовый интервал на интервалы Time Quanta (TQ). Битовый интервал состоит из четырех частей:

- Synchronization Segment (Sync_Seg);
- Propagation Time Segment (PSEG);
- Phase Buffer Segment 1 (SEG1);
- Phase Buffer Segment 2 (SEG2).

По определению Nominal Bit Time программируется длительностью от 8 до 25 TQ. В этом случае:

$$Nominal\ Bit\ Time = TQ \cdot (Sync_Seg + PSEG + SEG1 + SEG2)\ . \quad (2)$$

Время TQ фиксировано и определяется периодом генератора и программируемым прескалером BRP со значением от 1 до 65536:

$$TQ\ (\mu s) = \frac{BRP + 1}{CANx_CLK\ (MHz)} \quad (3)$$

или

$$TQ\ (\mu s) = (BRP + 1) \cdot Tclk\ (\mu s)\ . \quad (4)$$

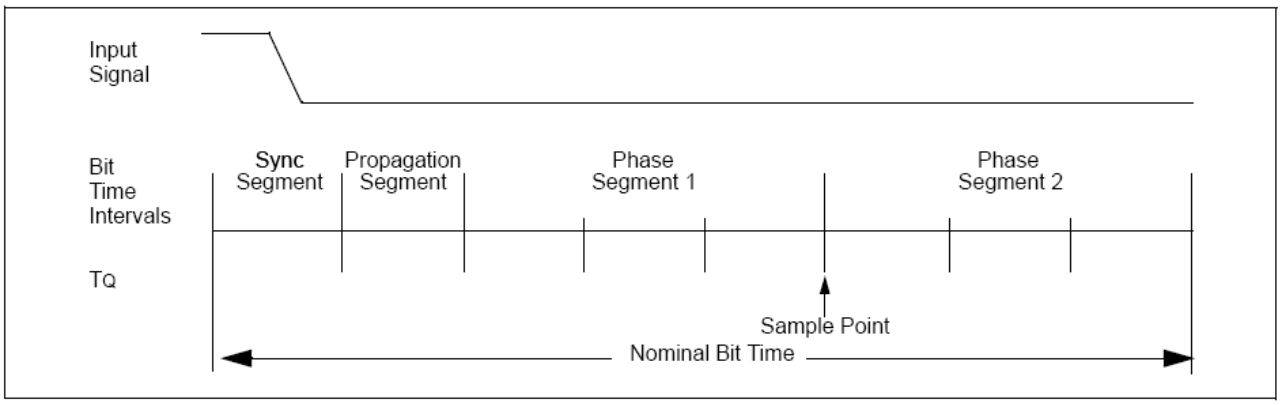



Рисунок 46 – Структура битового интервала

Synchronization Segment

Эта часть битового интервала, в которой должно происходить переключение сигнала. Длительность этого интервала 1 TQ. Если переключение происходит в этой области, то приемник засинхронизирован с передатчиком.

Propagation Time Segment

Эта часть предназначена, чтобы компенсировать физические задержки времени распространения сигнала в шине и внутренние задержки в узлах. Длительность этого интервала может быть запрограммирована от 1 до 8 TQ

Phase Buffer Segments

Эти интервалы предназначены для более точной установки точки семплирования, которая располагается между ними. Длительности этих интервалов могут быть запрограммированы между 1 и 8 TQ.

## 19.11 Синхронизация

При обнаружении фронта принимаемого сигнала этот момент принимается как граница между битовыми интервалами; в зависимости от того, на какой интервал приходится фронт, DPLL выполняет различного рода действия по подсинхронизации данных.

Hard Synchronization

Жесткая синхронизация выполняется однократно во время начала приема сообщения. Независимо от того, в каком состоянии находился DPLL при возникновении фронта, он переводится в Sync_Seg.

Resynchronization

Если фронт принимаемого сигнала отклоняется от Sync_Seg, длительность Phase Segment 1 может быть увеличена, а Phase Segment 2 уменьшена, чтобы в следующий раз фронт прошел в нужном месте. Величина изменения Phase Segment 1 и Phase Segment 2 варьируется в зависимости от значения отклонения фронта, но не превышает значения Synchronization Jump Width (SJW).

## 19.12 Обработка ошибок

В спецификации протокола CAN определено пять методов ограничения распространения ошибок, реализованных на аппаратном уровне. При обнаружении любой ошибки передающее устройство повторяет посылку пакета, поэтому ядру не нужно вмешиваться до тех пор, пока не возникнет грубая ошибка. Предусмотрено три метода обнаружения ошибок на уровне пакетов (контроль формата, CRC и подтверждение) и два метода на уровне бит (контроль бит и битстаффинг). Для реализации этих методов используется несколько полей, добавляемых к основному сообщению. При приеме осуществляется проверка, все ли поля присутствуют в сообщении. Если нет, то сообщение игнорируется, генерируется кадр ошибки и в регистре статуса контроллера STATUS устанавливается флаг ошибки формата пакета FRAME_ERR.

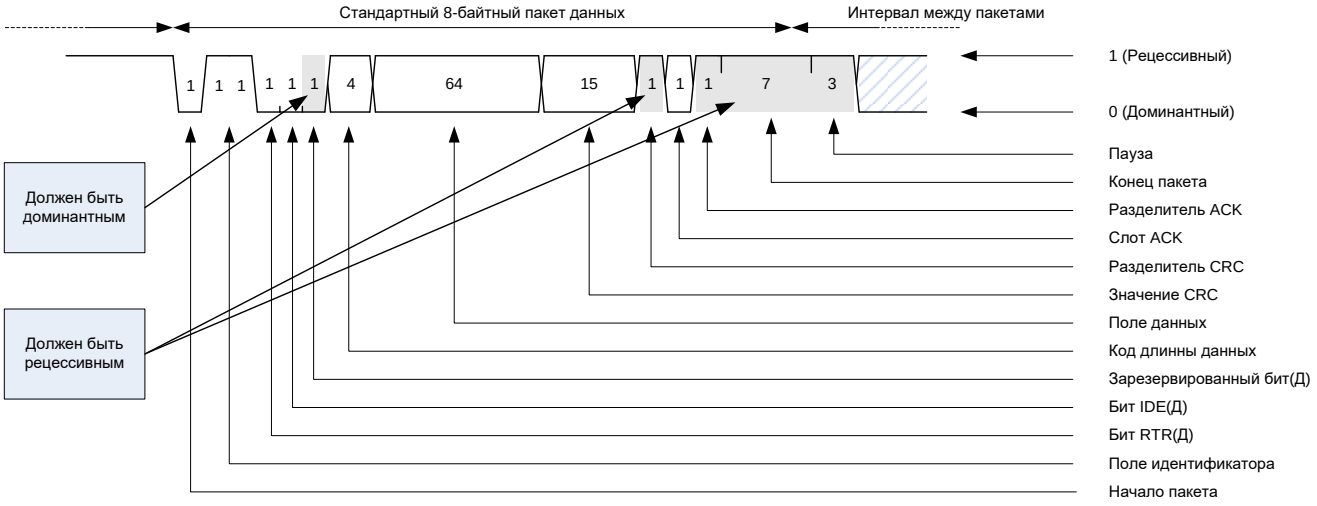



Рисунок 47 – Контроль формата пакета

Каждое сообщение должно подтверждаться вставкой доминантного бита в поле подтверждения. Если подтверждения нет, передающий узел будет передавать сообщение до тех пор, пока не получит подтверждение, при этом в регистре статуса контроллера STATUS будет установлен флаг ошибки подтверждения ACK_ERR.

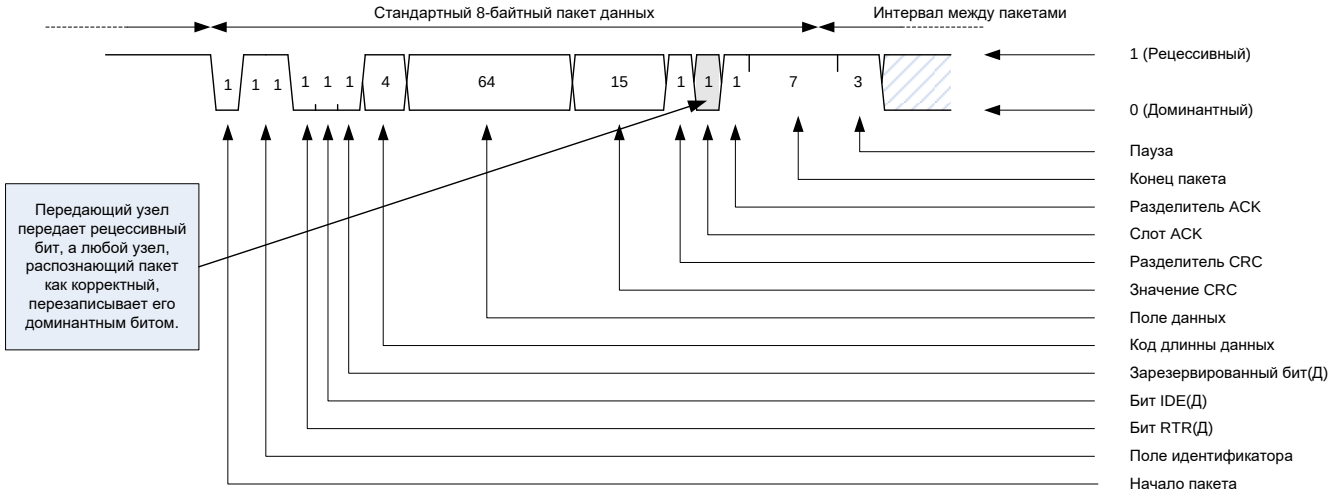



Рисунок 48 – Контроль подтверждения

Пакет сообщения CAN содержит 15-битовое значение CRC, которое автоматически генерируется передатчиком и проверяется приемником. С помощью этого кода можно обнаружить и исправить ошибку в четырех битах сообщения от начала кадра до начала поля CRC. Если CRC неверен и сообщение игнорируется, то передается кадр ошибки, и в регистре статуса контроллера STATUS будет установлен флаг ошибки контрольной суммы пакета CRC_ERR.

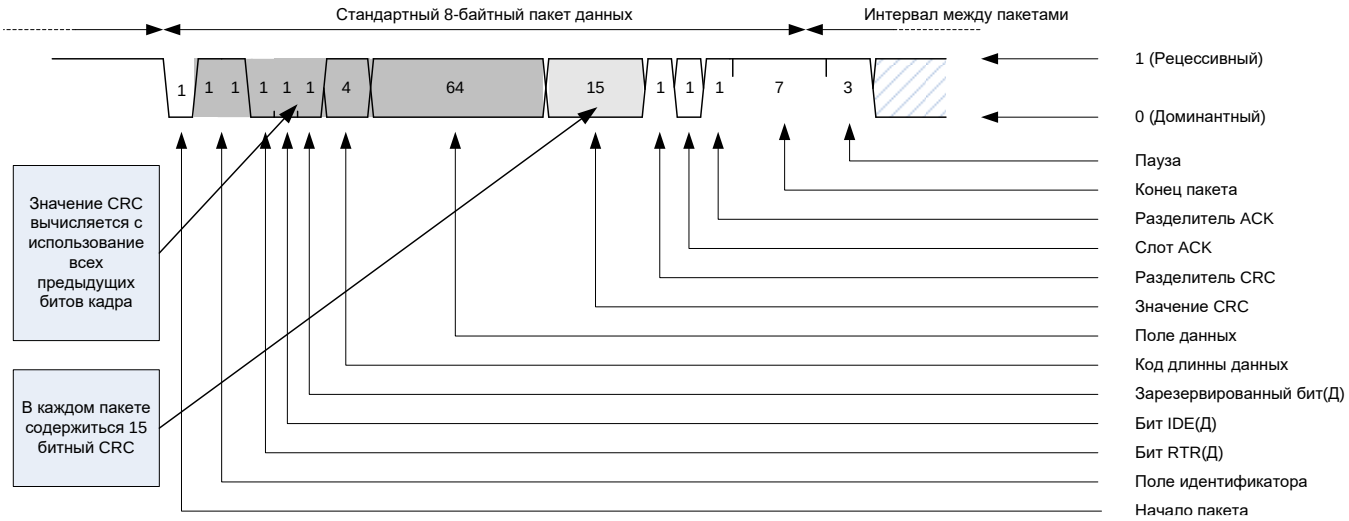



Рисунок 49 – Контроль CRC

После того, как узел выиграет арбитраж, он начинает передачу своего сообщения по шине. Как и во время арбитража, контроллер CAN считывает обратно каждый бит, выдаваемый им на шину. Поскольку узел уже выиграл арбитраж, больше никто не должен передавать данные на шине, поэтому значение каждого выданного на шину бита должно соответствовать значению, считанному обратно с шины. Если считано неверное значение, передатчик генерирует кадр ошибки, в регистре статуса контроллера STATUS устанавливается флаг ошибки передаваемых битов пакета BIT_ERR, и сообщение снова ставится в очередь. Это сообщение будет послано в следующем слоте сообщений, однако, при этом оно должно пройти через процесс арбитража с другими запланированными сообщениями.

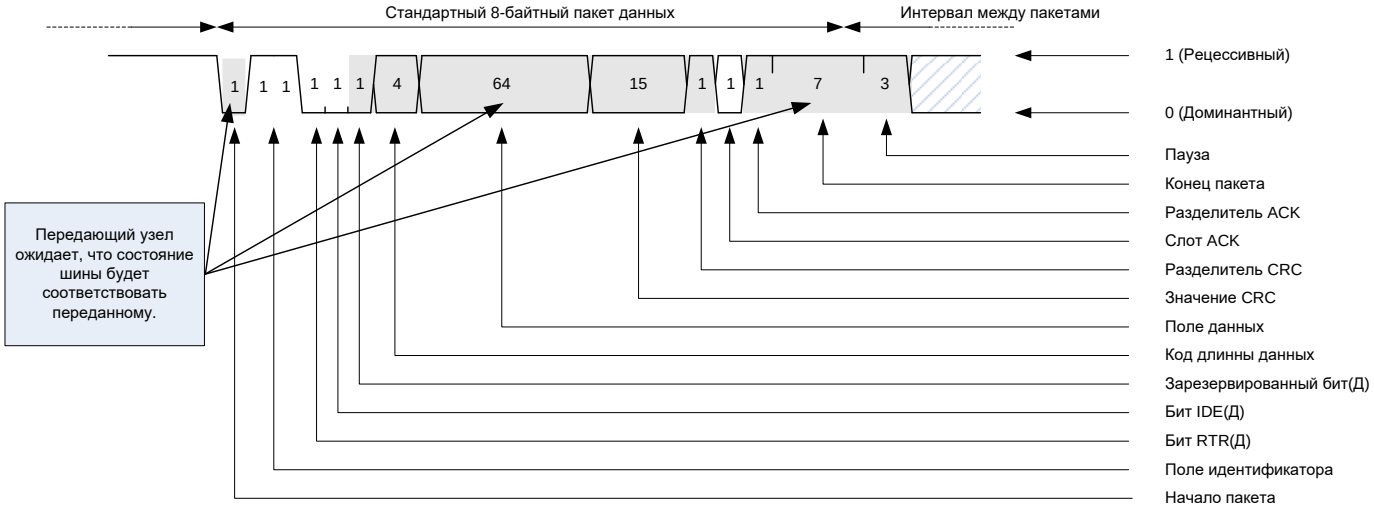



Рисунок 50 – Контроль передаваемых бит

На уровне бит в протоколе CAN реализован также метод вставки бита (битстаффинг). После каждой последовательности из пяти доминантных бит вставляется рецессивный бит; если рецессивный бит не обнаружен, в регистре статуса устанавливается флаг ошибки вставленных битов пакета BIT_STUF_ERR. Этот метод позволяет предотвратить появление на шине постоянных уровней и обеспечивает наличие в потоке бит достаточного количества переходов, используемых для повторной синхронизации. Кадр ошибки в протоколе CAN представляет собой простую последовательность из шести

доминантных битов. Это позволяет любому контроллеру CAN формировать на шине сообщение об ошибке сразу после ее обнаружения, не дожидаясь конца сообщения.

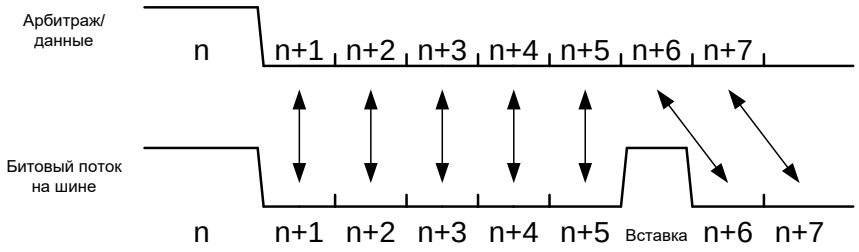



Рисунок 51 – Битстаффинг

В каждом контроллере CAN имеется два счетчика. Этими счетчиками являются счетчик ошибок приема (регистр STATUS, поле RX_ERR_CNT) и счетчик ошибок передачи (регистр STATUS, поле TX_ERR_CNT). Изменение состояния этих счетчиков происходит при приеме или передаче кадра ошибки. Когда любой счетчик достигает значения 128, контроллер CAN переходит в режим «error passive». В этом режиме он продолжает отзываться на кадры ошибки, однако при генерации кадра ошибки он вместо доминантных битов выставляет на шину рецессивные. Если счетчик ошибок передачи достигает значения 255, то контроллер CAN переходит в режим «bus-off» и больше не принимает участия в обмене по шине. Для возобновления обмена (переход контроллера CAN в состояние Error Active) необходимо либо вмешательство ядра микросхемы, которое повторно инициализирует контроллер CAN, либо на шине CAN не должно быть активности (шина в рецессивном состоянии) в течение времени, равного времени передачи 128 x 11 = 1408 бит. Текущий статус состояния контроллера можно посмотреть в регистре статуса контроллера STATUS. При успешном приеме/передаче кадров в режиме «error passive» счетчики ошибок RX_ERR_CNT и TX_ERR_CNT декрементируются, и модуль может перейти в режим «error active».

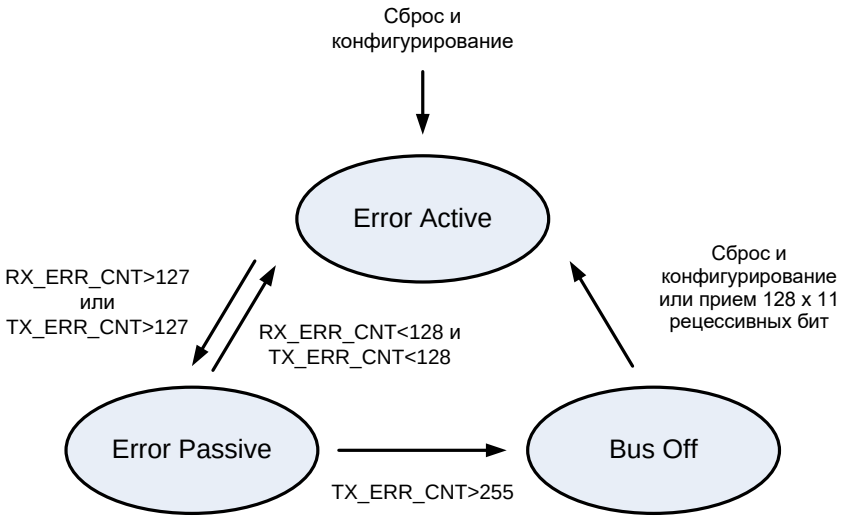



Рисунок 52 – Счетчики ошибок

Контроллер CAN имеет несколько механизмов обнаружения ошибок. Во-первых, из регистра состояния контроллера CAN_STATUS можно считать текущее состояние счетчиков ошибок приема и передачи. Также в этом регистре содержится флаг превышения счетчиками ошибок порогового значения ERROR_OVER. Это значение

произвольно и записывается в регистр CAN_OVER. Как и регистры синхронизации, регистр CAN_OVER можно изменять только при нахождении контроллера в состоянии сброса.

### 19.13 Прерывания

В контроллере CAN в качестве источников прерывания выступают буфера сообщений. Генерируемые прерывания делятся на три группы:

– прерывания передачи (по одному для каждого буфера);
– прерывания приема (по одному для каждого буфера);
– прерывания ошибки.

При возникновении какого-либо прерывания и наличии сигналов разрешения этих прерываний, буфер вырабатывает прерывание. Контроллер CAN объединяет прерывания приема, передачи и ошибки в каждом буфере и вырабатывает прерывание, отображаемое в регистре прерываний периферии. Если прерывание разрешено в регистре прерываний периферии, процессор выполняет переход на обработчик прерываний. Обработчик прерываний должен выполнить действия по обработке прерывания и снять его выставление. Прерывание передачи/приема для каждого буфера может быть замаскировано путем установления соответствующего бита в регистрах CAN_INT_TX/CAN_INT_RX. Также есть возможность группового маскирования прерываний по приему, по передаче и по ошибке (см. регистр CAN_INT_EN).

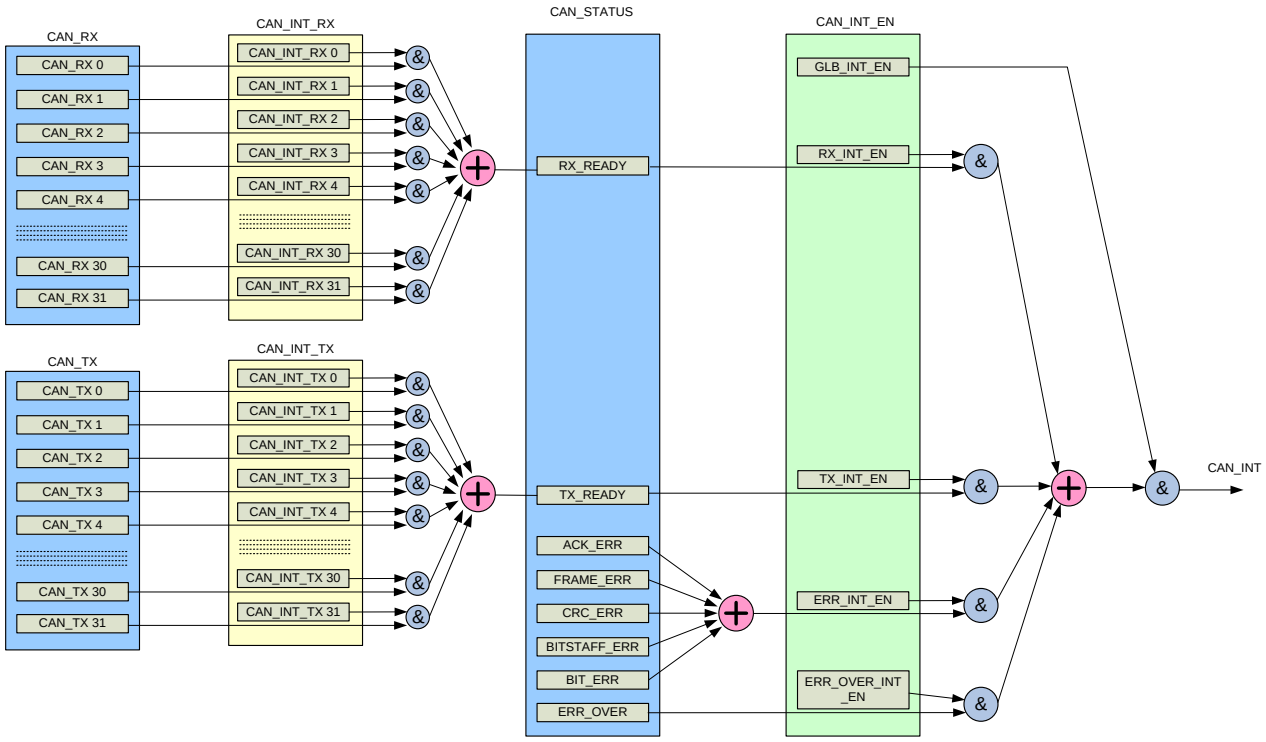



Рисунок 53 – Схема формирования прерывания блока CAN

## 19.14 Описание регистров контроллера CAN

Таблица 215– Описание регистров контроллера CAN

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x4000_0000 | CAN1 | Контроллер интерфейса CAN1 |
| 0x4000_8000 | CAN2 | Контроллер интерфейса CAN2 |
| Смещение | | |
| 0x00 | CAN_CONTROL | Регистр управления контроллером CAN |
| 0x04 | CAN_STATUS | Регистр состояния контроллера CAN |
| 0x08 | CAN_BITTMNG | Регистр задания скорости работы |
| 0x10 | CAN_INT_EN | Регистр разрешения прерываний контроллера |
| | | |
| 0x1C | CAN_OVER | Регистр границы счетчика ошибок |
| | | |
| 0x20 | CAN_RXID | Регистр принятого ID сообщения |
| 0x24 | CAN_RXDLC | Регистр принятого DLC сообщения |
| 0x28 | CAN_RXDATAL | Регистр принятых данных |
| 0x2C | CAN_RXDATAH | Регистр принятых данных |
| | | |
| 0x30 | CAN_TXID | Регистр передаваемого ID сообщения |
| 0x34 | CAN_TXDLC | Регистр передаваемого DLC сообщения |
| 0x38 | CAN_DATAL | Регистр передаваемых данных |
| 0x3C | CAN_DATAH | Регистр передаваемых данных |
| | | |
| 0x40 | CAN_BUF_01_CON | Регистр управления буфером 01 |
| | … | |
| 0xBC | CAN_BUF_32_CON | Регистр управления буфером 32 |
| | | |
| 0xC0 | CAN_INT_RX | Флаги разрешения прерываний от приемных буферов |
| 0xC4 | CAN_RX | Флаги RX_FULL от приемных буферов |
| 0xC8 | CAN_INT_TX | Флаги разрешения прерываний от передающих буферов |
| 0xCC | CAN_TX | Флаги ~TX_REQ от передающих буферов |
| | | |
| 0x200 | BUF_01_ID | ID сообщения буфера 01 |
| 0x204 | BUF_01_DLC | DLC сообщения буфера 01 |
| 0x208 | BUF_01_DATAL | Данные сообщения буфера 01 |
| 0x20C | BUF_01_DATAH | Данные сообщения буфера 01 |
| 0x210 | BUF_02_ID | ID сообщения буфера 02 |
| | … | |
| 0x3FC | BUF_32_DATAH | Данные сообщения буфера 32 |
| | | |
| 0x500 | BUF_01_MASK | Маска для приема сообщения в буфер 01 |
| 0x504 | BUF_01_FILTER | Фильтр для приема сообщения в буфер 01 |

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x508 | BUF_02_MASK | Маска для приема сообщения в буфер 02 |
| | … | |
| 0x5FC | BUF_32_FILTER | Фильтр для приема сообщения в буфер 32 |
| Примечание – Регистры CAN_RXID, CAN_RXDLC, CAN_RXDATAL/H (0x20 – 0x2C) и CAN_TXID, CAN_TXDLC, CAN_DATAL/H (0x30 – 0x3C) предназначены для временного хранения данных принятого или передаваемого пакета, соответственно. Использовать только для отладки | | |

### 19.14.1 CANx_CONTROL

Таблица 216 – Регистр управления контроллером CONTROL

| 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 |
| | - | ROP | SAP | STM | ROM | CAN EN |

Таблица 217 – Описание бит регистра CONTROL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…5 | - | Зарезервировано |
| 4 | ROP | Receive own packets:<br>1 – контроллер принимает собственные пакеты;<br>0 – контроллер принимает только чужие пакеты |
| 3 | SAP | Send ACK on own packets:<br>1 – контроллер подтверждает прием собственных пакетов;<br>0 – контроллер подтверждает прием только чужих пакетов |
| 2 | STM | Self Test Mode:<br>1 – контроллер работает в режиме самотестирования;<br>0 – контроллер работает в нормальном режиме |
| 1 | ROM | Read Only Mode:<br>1 – контроллер работает только на прием;<br>0 – контроллер работает в нормальном режиме |
| 0 | CAN_EN | Режим работы контроллера CAN:<br>1 – разрешение работы;<br>0 – сброс |

### 19.14.2 CANx_STATUS

Таблица 218 – Регистр состояния контроллера STATUS

| Номер | 31…24 | 23...16 |
|---|---|---|
| Доступ | RO | RO |
| Сброс | 0 | 0 |
| | TX<br>ERR<br>CNT [7:0] | RX<br>ERR<br>CNT [7:0] |

| Номер | 15...13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Доступ | R/W | RO | RO | RO | RO | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | TX<br>ERR<br>CNT8 | RX<br>ERR<br>CNT8 | ERR<br>STATUS[1:0] | | ID<br>LOWER |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | RO | RO |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ACK<br>ERR | FRAME<br>ERR | CRC<br>ERR | BIT<br>STUFF<br>ERR | BIT<br>ERR | ERROR<br>OVER | TX<br>READY | RX<br>READY |

Таблица 219 – Описание бит регистра STATUS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…24 | TX<br>ERR<br>CNT [7:0] | Счетчик ошибок передатчика TEC, биты [7:0]<br>TEC > 127, ERROR PASSIVE |
| 23…16 | RX<br>ERR<br>CNT [7:0] | Счетчик ошибок приемника REC, биты [7:0]<br>REC > 127, ERROR PASSIVE |
| 15…13 | - | |
| 12 | TX<br>ERR<br>CNT8 | Счетчик ошибок передатчика TEC, бит 8:<br>0 – TEC менее 255;<br>1 – TEC более 255 |
| 11 | RX<br>ERR<br>CNT8 | Счетчик ошибок приемника REC, бит 8:<br>0 – REC менее 255;<br>1 – REC более 255 |
| 10…9 | ERR<br>STATUS[1:0] | Статус состояния контроллера CAN:<br>00 – ERROR ACTIVE, при возникновении ошибки отсылается флаг активной ошибки;<br>01 – ERROR PASSIVE, при возникновении ошибки отсылается флаг пассивной ошибки;<br>1x – BUS OFF, ожидается восстановление шины |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 8 | ID LOWER | Флаг «проигрыша» арбитража:<br>0 – при передаче не было проигрыша арбитража;<br>1 – при передаче был проигран арбитраж |
| 7 | ACK ERR | Флаг ошибки подтверждения приема:<br>0 – нет ошибки;<br>1 – есть ошибка.<br>Примечание – сбрасывается записью нуля |
| 6 | FRAME ERR | Флаг ошибки формата пакета:<br>0 – нет ошибки;<br>1 – есть ошибка.<br>Примечание – сбрасывается записью нуля |
| 5 | CRC ERR | Флаг ошибки контрольной суммы пакета<br>0 – нет ошибки<br>1 – есть ошибка<br>Примечание – сбрасывается записью нуля |
| 4 | BIT STUFF ERR | Флаг ошибки вставленных битов пакета:<br>0 – нет ошибки;<br>1 – есть ошибка.<br>Примечание – сбрасывается записью нуля |
| 3 | BIT ERR | Флаг ошибки передаваемых битов пакета:<br>0 – нет ошибки:<br>1 – есть ошибка.<br>Примечание – сбрасывается записью нуля |
| 2 | ERROR OVER | Флаг превышения TEC и REC уровня, заданного ERROR_MAX:<br>0 – ERROR_MAX ≤ TEC и REC;<br>1 – ERROR_MAX > TEC или REC.<br>Примечание – сбрасывается записью нуля |
| 1 | TX READY | Флаг наличия данных в буфере для отправки:<br>0 – буферы пусты, нет сообщений готовых к передаче;<br>1 – есть не пустой буфер, в котором сообщение готово к передаче.<br>Примечание – биты TX READY регистра STATUS выставляются при условии, когда установлен бит TX_INT_EN = 1 (регистр CAN_INT_EN) и выставлен флаг разрешения прерывания от выполняющего отправку/прием буфера в регистрах CAN_INT_TX |
| 0 | RX READY | Флаг наличия принятых сообщений:<br>0 – буферы пусты, нет принятых сообщений;<br>1 – есть буфер, содержащий принятые сообщения.<br>Примечание – биты RX READY регистра STATUS выставляются при условии, когда установлен бит RX_INT_EN = 1 (регистр CAN_INT_EN) и выставлен флаг разрешения прерывания от выполняющего отправку/прием буфера в регистрах CAN_INT_RX |

### 19.14.3 CANx_BITTMNG

Таблица 220 – Регистр задания скорости работы BITTMNG

| Номер | 31…28 | 27 | 26…25 | 24…22 | 21…19 | 18…16 | 15…0 |
|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | SB | SJW [1:0] | SEG2 [2:0] | SEG1 [2:0] | PSEG [2:0] | BRP [15:0] |

Таблица 221 – Описание бит регистра BITTMNG

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…28 | - | Зарезервировано |
| 27 | SB | Семплирование:<br>0 – однократное;<br>1 – трехкратное с мажоритарным контролем |
| 26…25 | SJW [1:0] | Значение размера фазы SJW:<br>11 = Synchronization jump width time = 4 • TQ;<br>10 = Synchronization jump width time = 3 • TQ;<br>01 = Synchronization jump width time = 2 • TQ;<br>00 = Synchronization jump width time = 1 • TQ.<br>SJW – это максимальное значение, на которое происходит подстройка приема и передачи при работе на шине CAN. Приемник подстраивается на значение ошибки, но не более чем SJW |
| 24…22 | SEG2 [2:0] | Значение размера фазы SEG2:<br>111 = Phase Segment 2 time = 8 • TQ;<br>110 = Phase Segment 2 time = 7 • TQ;<br>101 = Phase Segment 2 time = 6 • TQ;<br>100 = Phase Segment 2 time = 5 • TQ;<br>011 = Phase Segment 2 time = 4 • TQ;<br>010 = Phase Segment 2 time = 3 • TQ;<br>001 = Phase Segment 2 time = 2 • TQ;<br>000 = Phase Segment 2 time = 1 • TQ.<br>SEG2 – это время, используемое для сокращения битового интервала при подстройке |
| 21…19 | SEG1 [2:0] | Значение размера фазы SEG1:<br>111 = Phase Segment 1 time = 8 • TQ;<br>110 = Phase Segment 1 time = 7 • TQ;<br>101 = Phase Segment 1 time = 6 • TQ;<br>100 = Phase Segment 1 time = 5 • TQ;<br>011 = Phase Segment 1 time = 4 • TQ;<br>010 = Phase Segment 1 time = 3 • TQ;<br>001 = Phase Segment 1 time = 2 • TQ;<br>000 = Phase Segment 1 time = 1 • TQ.<br>SEG1 – это время, используемое для увеличения битового интервала при подстройке |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 18…16 | PSEG[2:0] | Значение размера фазы PSEG:<br>111 = Propagation time = 8 • TQ;<br>110 = Propagation time = 7 • TQ;<br>101 = Propagation time = 6 • TQ;<br>100 = Propagation time = 5 • TQ;<br>011 = Propagation time = 4 • TQ;<br>010 = Propagation time = 3 • TQ;<br>001 = Propagation time = 2 • TQ;<br>000 = Propagation time = 1 • TQ.<br>PSEG – это время, компенсирующее задержку распространения сигналов в шине CAN |
| 15…0 | BRP [15:0] | Предделитель системной частоты<br>CLK = CANx_CLK/(BRP + 1);<br>TQ(us) = 1/CLK (MHz) = (BRP + 1)/CANx_CLK (MHz) |

### 19.14.4 CANx_INT_EN

Таблица 222 – Регистр разрешения прерываний INT_EN

| Номер | 31…5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Доступ | U | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | ERR OVER INT EN | ERR INT EN | TX INT EN | RX INT EN | GLB INT EN |

Таблица 223 – Описание бит регистра INT_EN

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…5 | - | Зарезервировано |
| 4 | ERR OVER INT EN | Флаг разрешения прерывания по превышению TEC или REC допустимого значения в ERROR_MAX:<br>0 – запрещено прерывание;<br>1 – разрешено прерывание |
| 3 | ERR INT EN | Флаг разрешения прерывания по возникновению ошибки:<br>0 – запрещено прерывание;<br>1 – разрешено прерывание |
| 2 | TX INT EN | Флаг разрешения прерывания по возможности передачи:<br>0 – запрещено прерывание;<br>1 – разрешено прерывание |
| 1 | RX INT EN | Флаг разрешения прерывания по приему сообщений:<br>0 – запрещено прерывание;<br>1 – разрешено прерывание |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 0 | GLB INT EN | Общий флаг разрешения прерывания блока CAN:<br>0 – запрещено прерывание;<br>1 – разрешено прерывание |

### 19.14.5 CANx_OVER

Таблица 224 – Регистр границы счета ошибок OVER

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

Таблица 225 – Описание бит регистра OVER

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7…0 | ERROR MAX [7:0] | Регистр границы счетчика ошибок.<br>Допустимое значение счетчиков ошибок TEC и REC, при превышении которого вырабатывается флаг ERROR_OVER |

### 19.14.6 CANx_BUF_xx_CON

Таблица 226 – Регистр управления буфером BUF_CON[x]

| Номер | 31…8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | OVER WR | RX FULL | TX REQ | PRIOR 0 | RTR EN | OVER EN | RX TXn | EN |

Таблица 227 – Описание бит регистра BUF_CON[x]

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…8 | - | Зарезервировано |
| 7 | OVER_WR | Флаг перезаписи принятого сообщения:<br>0 – не было перезаписи;<br>1 – была перезапись принятого сообщения |
| 6 | RX_FULL | Флаг готовности приема:<br>0 – нет принятого сообщения;<br>1 – принятое сообщение в буфере |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 5 | TX_REQ | Запрос на отправку сообщения:<br>0 – нет запроса или отправлено;<br>1 – запрос на отправку |
| 4 | PRIOR_0 | Приоритет при отправке:<br>0 – приоритет;<br>1 – нет приоритета |
| 3 | RTR_EN | Режим ответа на RTR:<br>1 – ответить при приеме RTR в буфер;<br>0 – не отвечать при приеме RTR |
| 2 | OVER_EN | Разрешение перезаписи принятого сообщения:<br>1 – разрешена перезапись сообщения;<br>0 – не разрешена перезапись |
| 1 | RX_TXn | Режим работы буфера:<br>1 – на прием;<br>0 – не передачу |
| 0 | EN | Разрешение работы буфера:<br>1 – работает;<br>0 – отключен |

### 19.14.7 CANx_INT_RX

Таблица 228 – Регистр разрешения прерываний от приемных буферов INT_RX

| Номер | 31…0 |
|---|---|
| Доступ | R/W |
| Сброс | 0 |
| | CAN_INT_RX[31:0] |

Таблица 229 – Описание бит регистра INT_RX

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…0 | CAN_INT_RX [31:0] | Флаги разрешения прерываний от буферов по приему сообщений:<br>CAN_INT_RX[0] – для первого буфера;<br>CAN_INT_RX[1] – для второго буфера и так далее |

### 19.14.8 CANx_RX

Таблица 230 – Регистр RX флагов RX_FULL от приемных буферов

| Номер | 31…0 |
|---|---|
| Доступ | RO |
| Сброс | 0 |
| | CAN_RX[31:0] |

Таблица 231 – Описание бит регистра RX

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…0 | CAN_RX[31:0] | Флаги RX_FULL разрешенных на прием буферов:<br>CAN_RX[0] – флаг RX_FULL от первого буфера;<br>CAN_RX[1] – флаг RX_FULL от второго буфера и так далее, доступны только на чтение |

### 19.14.9 CANx_INT_TX

Таблица 232 – Регистр разрешения прерываний от передающих буферов INT_TX

| Номер | 31…0 |
|---|---|
| Доступ | R/W |
| Сброс | 0 |
| | CAN_INT_TX[31:0] |

Таблица 233 – Описание бит регистра INT_TX

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…0 | CAN_INT_TX [31:0] | Флаги разрешения прерываний от буферов по передаче сообщений:<br>CAN_INT_TX[0] – для первого буфера;<br>CAN_INT_TX[1] – для второго буфера и так далее |

### 19.14.10 CANx_TX

Таблица 234 – Регистр TX флагов ~TX_REQ от передающих буферов

| Номер | 31..0 |
|---|---|
| Доступ | RO |
| Сброс | 0 |
| | CAN_TX[31:0] |

Таблица 235 – Описание бит TX

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…0 | CAN_TX[31:0] | Флаги ~TX_REQ разрешенных на передачу буферов:<br>CAN_TX[0] – флаг ~TX_REQ от первого буфера;<br>CAN_TX[1] – флаг ~TX_REQ от второго буфера и так далее, доступны только на чтение |

### 19.14.11 CANx_RXID, CANx_TXID, CANx_BUF_xx_ID

Таблица 236 – Регистры RXID, TXID и CAN_BUF[x].ID идентификаторов

| Номер | 31…29 | 28…18 | 17…0 |
|---|---|---|---|
| Доступ | U | R/W | R/W |
| Сброс | 0 | 0 | 0 |
| | - | SID [10:0] | EID [17:0] |

Таблица 237 – Описание бит регистров RXID, TXID и CAN_BUF[x].ID

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…29 | - | Зарезервировано |
| 28…18 | SID [10:0] | Поле SID.<br>Для стандартного и расширенного пакетов CAN.<br>Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже |
| 17…0 | EID [17:0] | Поле EID.<br>Для расширенных пакетов CAN.<br>Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже |

### 19.14.12 CANx_RXDLC, CANx_TXDLC, CANx_BUF_xx_DLC

Таблица 238 – Регистры RXDLC, TXDLC и CANx_BUF_xx_DLC сообщения

| Номер | 31…13 | 12 | 11 | 10 | 9 | 8 | 7…4 | 3…0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | IDE | SRR | R0 | R1 | RTR | - | DLC [3:0] |

Таблица 239 – Описание бит регистров RXDLC, TXDLC и CANx_BUF_xx_DLC

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…13 | - | Зарезервировано |
| 12 | IDE | Поле IDE.<br>Поле, обозначающее формат пакета:<br>1 – расширенный пакет;<br>0 – стандартный пакет |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 11 | SRR | Поле SRR, расширенного формата.<br>Всегда должен быть «1» |
| 10 | R0 | Поле R0.<br>Всегда должен быть «0» |
| 9 | R1 | Поле R1, расширенного формата.<br>Всегда должен быть «0» (в соответствии со стандартом) |
| 8 | RTR | Поле RTR, запроса обратного ответа:<br>0 – нет запроса;<br>1 – есть запрос.<br>Если узел получил пакет с запросом обратного ответа, он должен ответить |
| 7…4 | - | Зарезервировано |
| 3…0 | DLC[3:0] | Поле DLC, длина передаваемых данных в пакете:<br>0000 – нет данных;<br>0001 – 1 байт;<br>0010 – 2 байт;<br>0011 – 3 байт;<br>0100 – 4 байт;<br>0101 – 5 байт;<br>0110 – 6 байт;<br>0111 – 7 байт;<br>1000 – 8 байт;<br>1xxx – недопустимо |

### 19.14.13 CANx_RXDATAL, CANx_TXDATAL, CANx_BUF_xx_DATAL

Таблица 240 – Регистры RXDATAL, TXDATAL и CANx_BUF_xx_DATAL данных сообщения

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | DB3[7:0] | DB2[7:0] | DB1[7:0] | DB0[7:0] |

Таблица 241 – Описание бит регистров RXDATAL, TXDATAL и CANx_BUF_xx_DATAL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…24 | DB3[7:0] | Поле DB3, четвертый байт, передаваемый в пакете |
| 23…16 | DB2[7:0] | Поле DB2, третий байт, передаваемый в пакете |
| 15…8 | DB1[7:0] | Поле DB1, второй байт, передаваемый в пакете |

| | | |
|---|---|---|
| 7…0 | DB0[7:0] | Поле DB0, первый байт, передаваемый в пакете |

### 19.14.14 CANx_RXDATAH, CANx_TXDATAH, CANx_BUF_xx_DATAH

Таблица 242 – Регистры RXDATAH, TXDATAH и CANx_BUF_xx_DATAH данных сообщения

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | DB7[7:0] | DB6[7:0] | DB5[7:0] | DB4[7:0] |

Таблица 243 – Описание бит регистров RXDATAH, TXDATAH и CANx_BUF_xx_DATAH

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…24 | DB7[7:0] | Поле DB7, восьмой байт, передаваемый в пакете |
| 23…16 | DB6[7:0] | Поле DB6, седьмой байт, передаваемый в пакете |
| 15…8 | DB5[7:0] | Поле DB5, шестой байт, передаваемый в пакете |
| 7…0 | DB4[7:0] | Поле DB4, пятый байт, передаваемый в пакете |

# 20 Контроллер интерфейса по ГОСТ Р 52070-2003

В микросхеме имеется два независимых контроллера интерфейса по ГОСТ Р 52070-2003 (далее 1553), каждый из которых содержит необходимую логику и память для обработки и хранения командных слов и слов данных одного полного сообщения 1553. Каждый контроллер содержит два канала для приема/передачи сообщений 1553: основной и резервный. В один момент времени может работать только один из каналов – основной или резервный. Одновременная работа двух каналов не предусмотрена. Контроллер может работать как в режиме контролера шины, так и в режиме оконечного устройства. Для хранения входящих и исходящих командных и статусных слов, а также команд управления используются 16-разрядные регистры. Для хранения данных используется 16-разрядная двухпортовая память, в которой данные хранятся в области памяти, соответствующей подадресу командного слова. В каждом подадресе можно хранить только одно полное сообщение 1553. При передаче сообщения данные в память можно заносить как на «лету», так и до начала передачи. При приеме сообщения, данные можно считывать из памяти, как на «лету», так и после установки флага VALMESS.

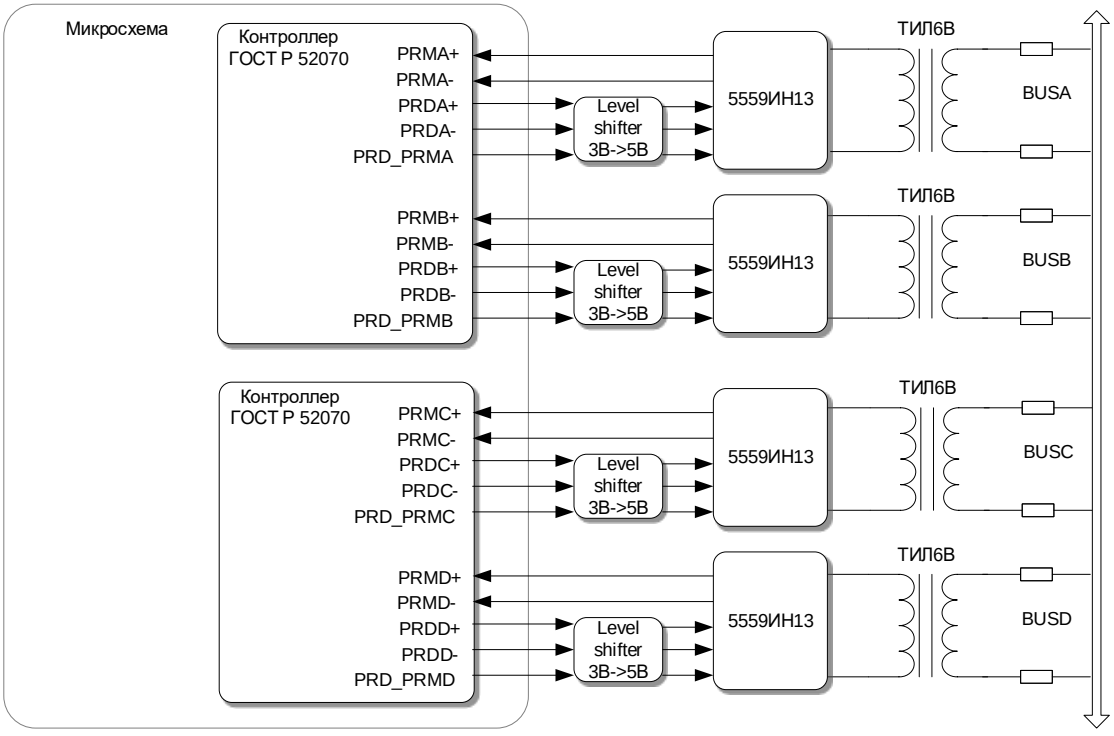



Рисунок 54 – Структурная схема контроллера интерфейса по ГОСТ Р 52070-2003

Особенности:

– поддержка основных (формат 1- формат 6) и групповых (формат 7 – формат 10) форматов сообщений;

– поддержка режимов работы: контроллер шины, оконечное устройство, монитор;

– скорость передачи данных 1 Мбит/с в полудуплексном режиме;

– поддержка двух каналов связи: основного и резервного;

– двухпортовая память принимаемых данных 1К×16;

– двухпортовая память передаваемых данных 1К×16;

– возможность формирования прерываний при успешном приеме и при возникновении ошибок на шине;

– маскирование прерываний.

## 20.1 Режимы работы

Контроллер поддерживает три режима работы: контроллера шины (КШ), оконечного устройства (ОУ) и неадресуемого монитора (М).

Контроллер шины

В этом режиме контроллер передает команды в магистраль, участвует в пересылке слов данных, принимает и контролирует ответную информацию о состоянии ОУ. Помимо этого, КШ реализует все команды управления. Для того чтобы реализовать передачу командного слова в магистраль, используется регистр CommandWord1. А для сообщений формата 3 и 8, помимо этого, применяется регистр CommandWord2. Ответная информация о состоянии ОУ после приема из магистрали хранится в регистре StatusWord1. А для сообщений формата 3 и 8, помимо этого, применяется регистр StatusWord2. Для передачи и приема слов данных команд управления (КУ), форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой бита BCMODE и сбросом RTMODE.

Оконечное устройство

В этом режиме контроллер осуществляет проверку достоверности командных слов, поступающих к нему от КШ. Командное слово считается достоверным, если не возникло ошибок в магистрали при его приеме, или если поле «Адрес ОУ» соответствует коду собственного адреса ОУ или коду 11111 (групповая команда). Если командное слово определено как достоверное, то ОУ посылает в линию ответное слово (ОС) и в зависимости от поля «Прием/Передача» принимает или передает число данных, соответствующее полю «Число СД/Код КУ». Если же происходит прием от КШ команды управления, то ОУ реагирует в соответствии с форматами сообщений команд управления. Принятое из магистрали командное слово помещается в регистр CommandWord1, а для сообщений формата 3 и 8 принятое второе командное слово помещается в регистр CommandWord2. Ответное слово ОУ для передачи в магистраль помещается в регистр StatusWord1. Помимо этого, для сообщения формата 3, этот регистр содержит принятое ответное слово от другого ОУ. Для передачи и приема слов данных команд управления, форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой бита RTMODE и сбросом BCMODE.

Монитор

В этом режиме осуществляется прослушивание магистрали и отбор необходимой информации для проведения: технического обслуживания, регистрации эксплуатационных параметров, анализа решаемых задач или обеспечения информацией резервного КШ. Монитор пассивно прослушивает выбранную шину и захватывает весь

трафик на шине, но никогда не передает информацию на шину. Принятое из магистрали командное слово помещается в регистр CommandWord1, а для сообщений формата 3 и 8 принятое второе командное слово помещается в регистр CommandWord2. Ответное слово ОУ, принятое из магистрали, помещается в регистр StatusWord1. А для сообщений формата 3 и 8, помимо этого, применяется регистр StatusWord2. Для приема слов данных команд управления, форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой битов RTMODE и BCMODE. Для быстрой расшифровки сообщений можно применить регистр MSG. Каждому формату сообщения на магистрали соответствует определенный код в этом регистре.

## 20.2 Форматы сообщений

Сообщения, передаваемые по информационной магистрали, имеют формат, соответствующий форматам основных или групповых сообщений. Любые другие типы сообщений, не соответствующие ГОСТ Р 52070-2003, не поддерживаются.

Форматы основных сообщений, приведенные на рисунке 55, используются для передачи информации, предназначенной одному ОУ, и предусматривают выдачу ОС. В данном случае КС – командное слово, СД – слово данных, ОС – ответное слово. Времена $t_1$ и $t_2$ формируются аппаратно и не могут быть изменены программно. Пауза $t_2$ между сообщениями, формируемая КШ, – не менее 4 мкс, а пауза перед передачей ОС, формируемая ОУ, – в пределах от 4 до 12 мкс. Если после ожидания 14 мкс так и не поступило ОС от ОУ, то фиксируется отсутствие ОС от ОУ и формируется соответствующий признак ошибки.

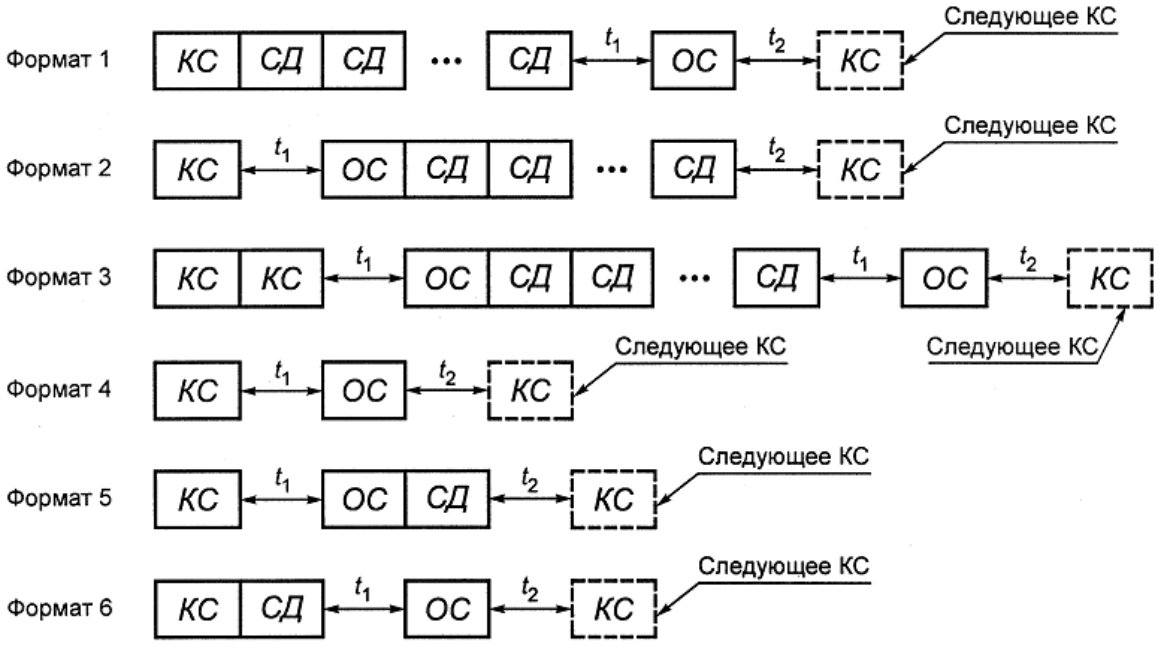



Рисунок 55 – Форматы сообщений

Время непрерывной передачи данных в линию не превышает 660 мкс, что соответствует командному слову и 32-м словам данных.

Формат 1 – передача данных от КШ к ОУ.

Формат 2 – передача данных от ОУ к КШ.

Формат 3 – передача данных от ОУ к ОУ.

Формат 4 – передача КУ.

Формат 5 – передача КУ и прием СД от ОУ.

Формат 6 – передача КУ и СД оконечному устройству.

Групповые сообщения, приведенные на рисунке 56, начинающиеся с передачи КШ групповой команды с кодом адреса 11111, используются для передачи информации одновременно нескольким ОУ без выдачи ими ОС.

Формат 7 – передача данных (в групповом сообщении) от КШ к оконечным устройствам.

Формат 8 – передача данных (в групповом сообщении) от оконечного устройства к оконечным устройствам.

Формат 9 – передача групповой команды управления.

Формат 10 – передача групповой команды управления со словом данных.

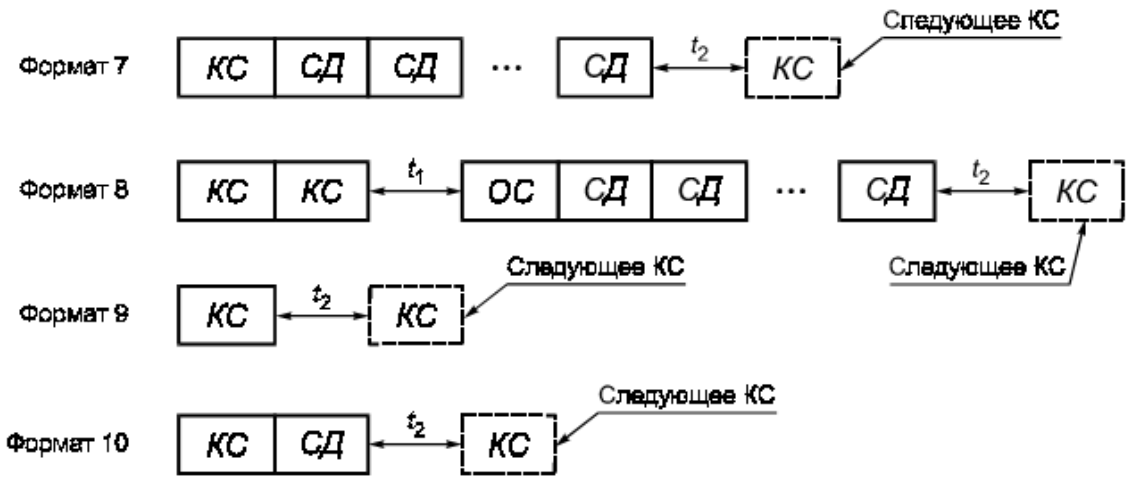



Рисунок 56 – Форматы групповых сообщений

Если ОУ в формате сообщения ОУ-ОУ получило достоверное командное слово на прием информации, то первое СД должно быть им принято через паузу не более (57±3) мкс, в противном случае формируется соответствующий признак ошибки.

## 20.3 Формат слов

Каждое слово начинается с сигнала пословной синхронизации (с синхросигнала) и имеет 17 информационных разрядов, включая разряд контроля по четности. Форматы слов приведены на рисунке 57.

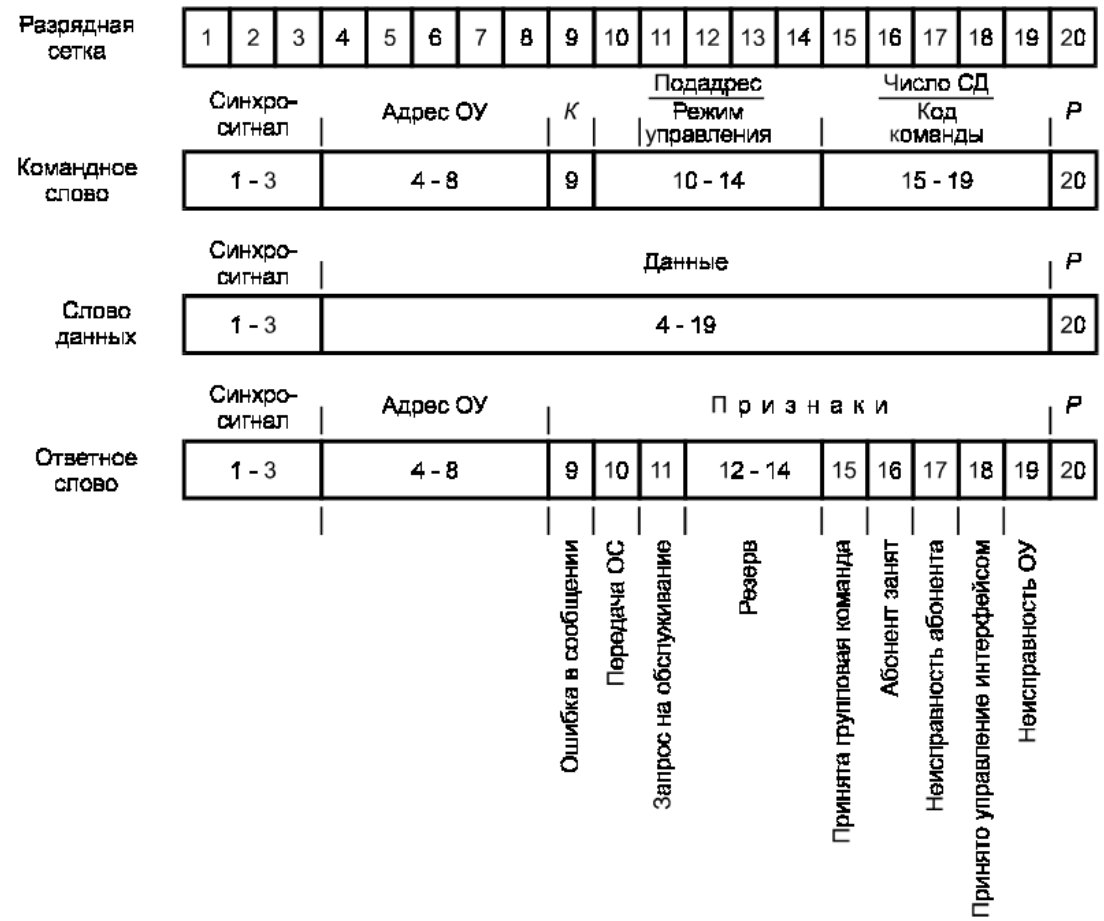



Рисунок 57 – Форматы слов

Командное слово содержит:

- синхросигнал;
- поле «Адрес ОУ»;
- разряд «Прием/Передача»;
- поле «Подадрес/Режим управления»;
- поле «Число СД/Код КУ»;
- разряд контроля по четности.

Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала положительная, а второй – отрицательная.

Адрес ОУ содержит код адреса из диапазона кодов 00000 – 11110, которому предназначено КС. КС с кодом адреса 11111 называется групповой командой, а сообщение, содержащее групповую команду – групповым.

Разряд «Прием/Передача» указывает на действие, которое должно выполнить ОУ (принимать или передавать СД). Логический «0» означает, что ОУ должно принимать СД, а логическая единица – передавать СД.

Поле «Подадрес/Режим управления» содержит код подадреса ОУ или код признака режима управления 00000 или 11111.

Поле «Число СД/Код КУ» содержит код числа слов данных, которые должны быть переданы или приняты ОУ в связи с приемом адресованного ему КС, или код КУ. В одном

сообщении может быть передано не более 32 СД. Числовое значение двоичных кодов, обозначающих число СД, соответствует их десятичным эквивалентам, за исключением кода 00000, который соответствует числу 32.

Разряд контроля по четности используется для контроля по четности предшествующих ему 16 разрядов КС. Разряд принимает такое значение, чтобы сумма значений всех 17 информационных разрядов слова (включая контрольный разряд) была нечетной.

Слово данных содержит:

- синхросигнал;
- данные;
- разряд контроля по четности.

Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала отрицательная, а второй – положительная.

Поле данных содержит передаваемые данные, а разряд контроля по четности формируется так же, как в командном слове.

Ответное слово содержит:

- синхросигнал;
- поле «Адрес ОУ»;
- поле разрядов признаков состояния: ошибка в сообщении, передача ОС, запрос на обслуживание, принята групповая команда, абонент занят, неисправность абонента, принято управление интерфейсом, неисправность ОУ;
- разряд контроля по четности.

Синхросигнал аналогичен синхросигналу КС. Поле «Адрес ОУ» содержит собственный адрес ОУ. Поле разрядов признаков состояния ОУ отображает текущее состояние ОУ. Разряд контроля по четности формируется так же, как в командном слове.

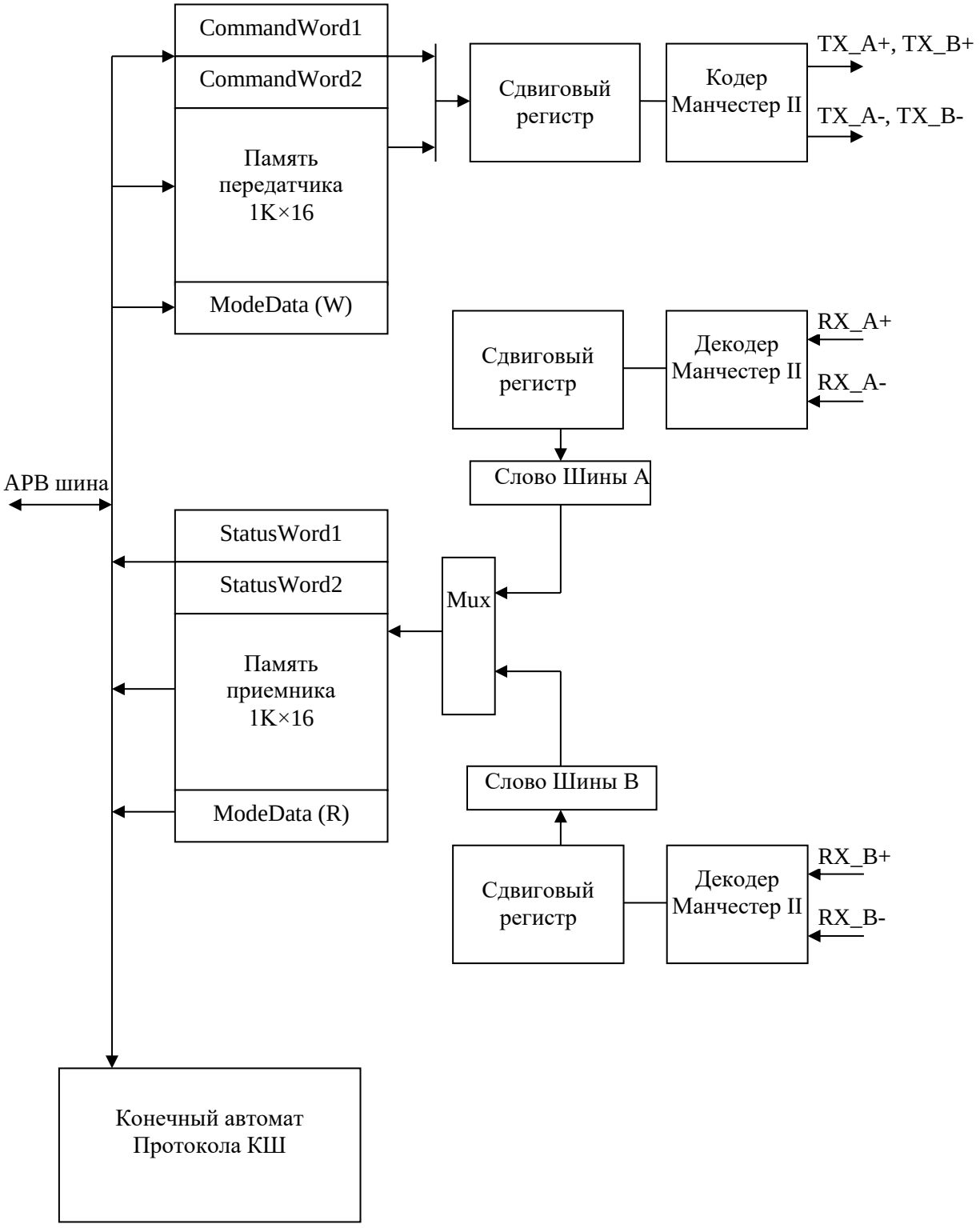



Рисунок 58 – Структурная схема работы в режиме КШ

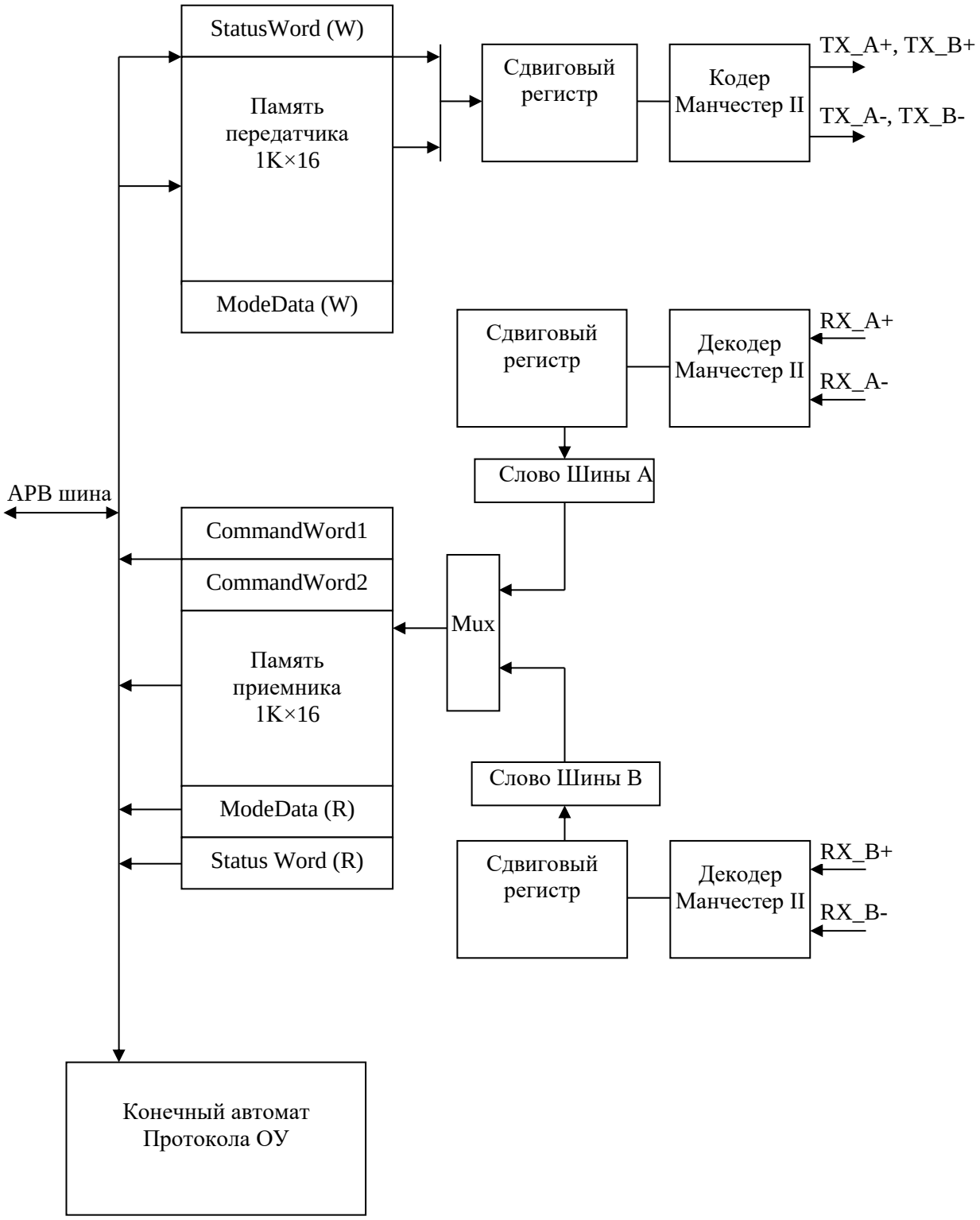



Рисунок 59 – Структурная схема работы в режиме ОУ

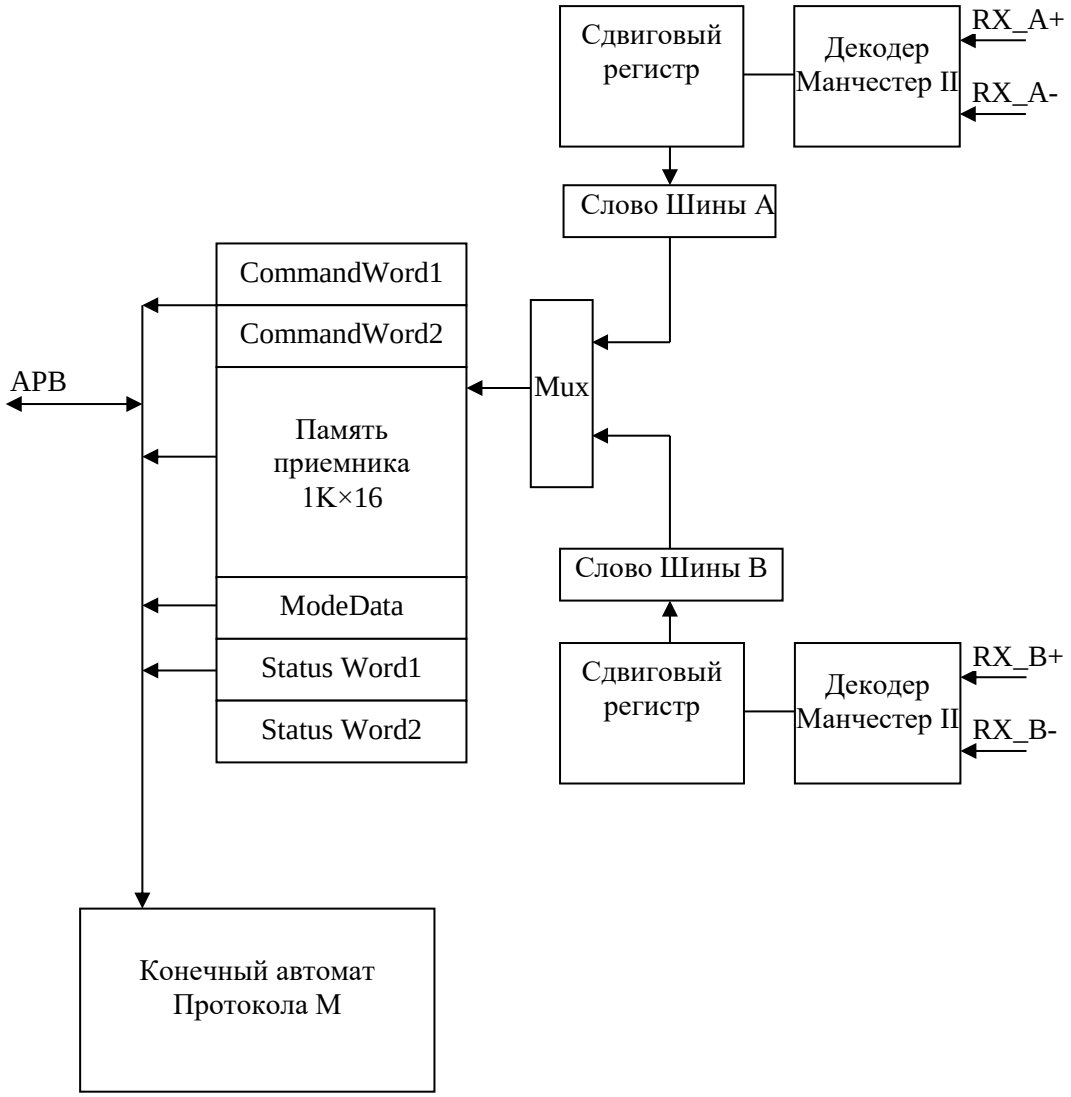



Рисунок 60 – Структурная схема работы в режиме M

## 20.4 Инициализация

Перед началом работы с контроллером в первую очередь необходимо сбросить контроллер, чтобы очистить все регистры сообщений. Это достигается установкой бита MR регистра CONTROL в логическую единицу. Затем бит необходимо сбросить в «0». Далее нужно задать в регистре CONTROL значение делителя частоты DIV таким образом, чтобы при делении частоты MAN_CLK на это значение получить опорную частоту блока контроллера 1 МГц. После этого с помощью бит RTMODE и BCMODE выбирается соответствующий режим работы ОУ или КШ.

Для того чтобы выбрать, какой канал будет использован для передачи данных, основной или резервный, устанавливается соответствующий бит (TRA – основной канал, TRB – резервный канал). В режиме КШ командные слова будут передаваться только по тому каналу, который выбран. В режиме ОУ необходимо установить оба бита, так как ОУ не может выбирать, по какому каналу ему передавать СД и ОС, и поэтому их передача происходит по тому каналу, по которому было принято КС.

Для режима ОУ в битах RTA4 – RTA0 регистра CONTROL задается адрес ОУ, который должен соответствовать адресу в поле «Адрес ОУ» командного слова, если идет обращение к этому ОУ.

*Пример инициализации ОУ*
*MIL_STD_1553B1->CONTROL=0x00000001; //Установка бита MR=1*
*MIL_STD_1553B1->CONTROL= 0x00014078;*
*//RTMODE=1, TRB=TRA=1, RTA=1, DIV=40*

*Пример инициализации КШ*
*MIL_STD_1553B1->CONTROL=0x00000001; //Установка бита MR=1*
*MIL_STD_1553B1->CONTROL= 0x00014014;*
*//BCMODE=1, TRA=1, TRB=0, RTA=0, DIV=40*

В обоих случаях значения делителя частоты DIV = 40, что соответствует частоте ядра 40 МГц, и для получения опорной частоты контроллера необходимо 40 МГц/DIV = 1 МГц.

## 20.5 Прием и передача в режиме ОУ

Для того чтобы настроить контроллер в режиме ОУ, необходимо выполнить все пункты, описанные в параграфе Инициализация. После этого необходимо задать ответное слово для КШ с помощью регистра StatusWord1. В режиме ОУ регистр по записи содержит предназначенное для передачи КШ ответное слово, а по чтению содержит ответное слово, полученное от передающего ОУ в транзакции ОУ-ОУ.

*Пример записи ответного слова ОУ*
*MIL_STD_1553B1->StatusWord1=0x00000800;*

В данном случае в регистр заносятся только старшие пять разрядов, соответствующие адресу ОУ. Остальные разряды признаки состояния ОУ можно оставить в нуле. Но в процессе работы может возникать необходимость изменять эти биты. Для этого необходимо программно устанавливать и сбрасывать эти биты, так как аппаратно они не изменяются.

Для того чтобы обеспечить формат сообщения 5, необходимо задавать слово данных, передаваемое КШ в команде управления. Для этих целей используется регистр ModeData.

*Пример записи слова данных команды управления*
*MIL_STD_1553B1->ModeData=0x000055AA;*

После того как проведена инициализация, заданы ответное слово и слово данных команды управления, ОУ сразу готово к работе и может отвечать на все возможные форматы сообщений.

Так как в процессе работы ОУ в каждый момент времени требуется передача определенных СД и СД команды управления, то программно необходимо обновлять те области памяти, которые содержат эти данные. Если эти области не обновляются, то при запросе данных КШ будут переданы те данные, которые были последний раз записаны в эти области памяти. Поэтому при написании программы следует помнить и обновлять данные и слова данных команды управления.

Для хранения СД применяется адресное пространство 0x000-0xFFC (относительно базового адреса периферийного блока). Данные шестнадцатиразрядные, но обращение к

ним должно быть выровнено по границе 32-разрядного слова. То есть два младших разряда не участвуют в формировании адреса.

```
Пример инициализации данных для подадреса 1
addon=0x80;
addr_DATA=(unsigned int)&MIL_STD_1553B1->DATA[0];
for(i=1;i<=32;i++)
{
   *((volatile unsigned int *)(addr_DATA+addon)) = i;
   addon=addon+4;
}
```

Из примера становится ясно, что стартовый адрес памяти СД для подадреса 1 – 0x80, для последующих подадресов: n*0x80, где n- номер подадреса (n=1-31). Запись СД в память подадреса необходимо выполнять до начала фактической передачи записываемого СД от ОУ к КШ.

При приеме СД или слова данных команды управления, признаком обновления их значений является флаг VALMESS. После того как флаг установлен, можно считать новые данные или слово данных команды управления. Но следует учитывать то, что этот флаг автоматически сбрасывается через 4 мкс после его установки, поэтому желательно применять прерывания по установке сигнала VALMESS.

```
Пример чтения слова данных команды управления
i=MIL_STD_1553B1->ModeData;
В переменную i будет прочитано значение слова данных команды управления.

Пример чтения данных для подадреса 1
addon=0x80;
addr_DATA=(unsigned int)&MIL_STD_1553B1->DATA[0];
for(i=0;i<32;i++)
{
   mas[i]=*((volatile unsigned int *)(addr_DATA+addon));
   addon=addon+4;
}
```

Для упрощения декодирования команд управления КШ в режиме ОУ доступен регистр кодов MSG полученных сообщений. Каждому формату сообщения на магистрали соответствует определенный код в этом регистре. Чтение и последующая дешифрация этого кода упрощает процедуру декодирования сообщения и уменьшает время обработки сообщений. При использовании регистра экономится время на чтение двух командных регистров и разбор значений бит этих регистров.

## 20.6 Прием и передача в режиме КШ

В отличие от режима ОУ, в режиме КШ необходимо задавать не ответное слово, а командное слово или два командных слова в режиме работы ОУ-ОУ. Помимо этого, нужно инициировать процедуру приема или передачи данных установкой бита BCSTART. После

завершения транзакции на шине этот бит автоматически сбрасывается в ноль. Поэтому для инициирования новой транзакции нужно повторно устанавливать этот бит.

*Пример записи командных слов и бита BCSTART*
*MIL_STD_1553B1->CommandWord1=0x00000820; //Командное слово 1*
*MIL_STD_1553B1->CommandWord2=0x00000000 //Командное слово 2*
*MIL_STD_1553B1->CONTROL=0x00014016; //Регистр управления*

Как видно из примера, в командном слове 1 задается код слов данных 00000, что соответствует 32 СД. Данные будут передаваться от контроллера шины оконечному устройству с адресом 1 из подадреса 1. Второе командное слово задается равным нулю и никак не влияет на транзакцию. В регистре управления устанавливается бит BCMODE, что соответствует режиму работы КШ, а также устанавливается бит BCSTART, что инициирует начало транзакции, выбирается канал А для передачи (TRA=1), а также устанавливается делитель частоты 40, что соответствует частоте работы ядра 40 МГц.

Для того чтобы инициировать прием в этом примере, необходимо только установить бит 10 равным единице в командном слове 1.

Если транзакция завершена успешно (признак VALMESS установился в единицу), то полученные СД или слово данных команды управления могут быть прочитаны. В противном случае устанавливается один из флагов ошибки. Сброс этих флагов осуществляется либо установкой битом MR, либо инициированием новой транзакции битом BCSTART.

В режиме работы ОУ-ОУ, форматы сообщений 3 и 8, КШ принимает из магистрали СД в подадрес, указанный во втором командном слове, регистр CommandWord2.

После приема ОС необходимо выполнить программный анализ бит признаков состояния ОУ. Аппаратно контроллер выполняет анализ бит «Абонент занят» и «Ошибка в сообщении». Если в принятом ОС установлен бит «Абонент занят» или бит «Ошибка в сообщении», то в регистре STATUS устанавливается бит ERR и выполняется переход в состояние обработки паузы t2. В формате 5 при приеме ОС с битом «Абонент занят», равным нулю, бит «Ошибка в сообщении» аппаратно не анализируется.

## 20.7 Прерывания

Для уменьшения потерь времени программы на опрос флагов контроллера, введено одно прерывание, генерируемое при установке любого из флагов контроллера. Прерывание может генерировать установка одного из четырех флагов:

- флаг ошибки;
- флаг успешного завершения транзакции в канале;
- флаг приема достоверного КС, ОС или слова данных команды управления;
- флаг неактивности контроллера.

Каждый из флагов может быть маскирован битами разрешения прерывания по какому-либо флагу.

Разрешенные в регистре INTEN сигналы флагов контроллера по фронту формируют независимые внутренние импульсные сигналы INT[3:0]. Сигналы INT[3:0] объединяются по схеме ИЛИ и формируют общий импульсный запрос прерывания,

который реализует прерывание по фронту. Сброс прерывания выполняется автоматическии ввиду импульсного запроса прерывания.

## 20.8 Описание регистров контроллера ГОСТ Р 52070-2003

Таблица 244 – Описание регистров контроллера ГОСТ Р 52070-2003

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x4004_8000 | MIL_STD_1553B1 | Контроллер интерфейса 1553 канал 1 |
| 0x4005_0000 | MIL_STD_1553B2 | Контроллер интерфейса 1553 канал 2 |
| Смещение | | |
| 0x000-0xFFC | DATA | Память принимаемых/передаваемых СД |
| 0x1000 | CONTROL | Регистр управление контроллером |
| 0x1004 | STATUS | Регистр состояния контроллера |
| 0x1008 | ERROR | Регистр ошибок контроллера |
| 0x100C | CommandWord1 | Регистр командного слова 1 |
| | | |
| 0x1010 | CommandWord2 | Регистр командного слова 2 |
| | | |
| 0x1014 | ModeData | Слово данных команды управления |
| 0x1018 | StatusWord1 | Регистр ответного слова 1 |
| 0x101C | StatusWord2 | Регистр ответного слова 2 |
| 0x1020 | INTEN | Регистр разрешения прерываний |
| 0x1024 | MSG | Регистр декодирования сообщений |

### 20.8.1 CONTROL

Таблица 245 – Регистр управления контроллером CONTROL

| Номер | 31….24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | U | R/W | R/W | R/W | R/W | R/W |
| Сброс | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | RT_HW | INPINV | DIV7 | EN_FLT | INVTR | RERR | DIV6 | DIV5 |

| Номер | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | DIV4 | DIV3 | DIV2 | DIV1 | DIV0 | RTA4 | RTA3 | RTA2 |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | RTA1 | RTA0 | TRB | TRA | RTMODE | BCMODE | BCSTART | MR |

Таблица 246 – Описание бит регистра CONTROL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…24 | - | Зарезервировано |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 23 | RT_HW | Бит аппаратной поддержки ГОСТ 51765-2001 для режима ОУ:<br>1 – аппаратная поддержка разрешена;<br>0 – аппаратная поддержка запрещена.<br>При установке этого бита достаточно программной реализации циклического подадреса, установки и сброса признаков абонент занят, неисправность абонента, неисправность ОУ, запрос на обслуживание, а также задание адреса ОУ и бита паритета посредством внешнего соединения |
| 22 | INPINV | Бит инверсии входов приемника PRMx+ и PRMx-:<br>1 – инверсия разрешена;<br>0 – инверсия запрещена.<br>Вход PRMx+ инвертируется и коммутируется на отрицательный вход декодера манчестерского кода.<br>Вход PRMx- инвертируется и коммутируется на положительный вход декодера манчестерского кода.<br>Применение этого бита актуально для приемопередатчиков с принудительной установкой выхода приемника в состояние логическая «1», например, 5559ИН74Т (HI1574) |
| 21 | DIV7 | Делитель частоты MAN_CLK до 1 МГц.<br>Содержит старший разряд делителя, на который необходимо поделить частоту MAN_CLK, чтобы получить 1 МГц |
| 20 | EN_FLT | Включение фильтрации импульсных помех:<br>1 – фильтрация включена;<br>0 – фильтрация выключена.<br>Рекомендуется устанавливать в ноль |
| 19 | INVTR | Разрешение инверсии сигналов PRD_PRMA, PRD_PRMB, PRD_PRMC, PRD_PRMD:<br>1 – инверсия;<br>0 – прямой выход |
| 18 | RERR | Сброс ошибок в режиме ОУ и М:<br>1 – ошибки могут быть сброшены только битом MR;<br>0 – сброс ошибок происходит автоматически, после установки бита IDLE |
| 17…11 | DIV6-DIV0 | Делитель частоты MAN_CLK до 1 МГц.<br>Содержит значение, на которое необходимо поделить частоту MAN_CLK, чтобы получить 1 МГц. Частота MAN_CLK обязательно должна быть не менее 24 МГц, не более 120 МГц и кратна 8. Если MAN_CLK не кратна 8, то DIV[6:3]=(MAN_CLK/8)+1, DIV[2:0]=0, но стабильность приема не гарантируется |
| 10…6 | RTA4-RTA0 | Адрес оконечного устройства.<br>Содержит адрес, который присвоен устройству, если контроллер работает в режиме оконечного устройства:<br>RTMODE = 1; BCMODE = 0 |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 5 | TRB | Блокировка передатчика резервного канала:<br>1 – передатчик разблокирован;<br>0 – передатчик заблокирован |
| 4 | TRA | Блокировка передатчика основного канала:<br>1 – передатчик разблокирован;<br>0 – передатчик заблокирован |
| 3…2 | RTMODE<br>BCMODE | Выбор режима работы контроллера:<br>10 – режим оконечного устройства;<br>01 – режим контроллера шины;<br>11 – режим неадресуемого монитора |
| 1 | BCSTART | Инициирует передачу сообщения в канал в режиме КШ:<br>1 – старт сообщения;<br>0 – стоп сообщения.<br>Сбрасывается в ноль автоматически по завершении сообщения |
| 0 | MR | Сброс контроллера:<br>1 – контроллер сбрасывается в исходное состояние;<br>0 – разрешение работы контроллера |

### 20.8.2 STATUS

Таблица 247 – Регистр состояния STATUS

| Номер | 31…11 | 10 | 9 | 8 | 7 | 6 |
|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W |
| Сброс | | 0 | 0 | 0 | 0 | 0 |
| | - | RCVB_stick | RCVA_stick | ERR_stick | VALMESS_stick | RFLAGN_stick |

| Номер | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Доступ | RO | RO | RO | RO | RO | RO |
| Сброс | 0 | 0 | 0 | 0 | 0 | 1 |
| | RCVB | RCVA | ERR | VALMESS | RFLAGN | IDLE |

Таблица 248 – Описание бит регистра STATUS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...11 | - | Зарезервировано |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 10 | RCVB_stick* | Признак активности резервного канала:<br>0 – канал неактивен;<br>1 – канал активен.<br>Устанавливается при установке бита RCVB. Сбрасывается программной записью нуля, если RCVB = 0.<br>В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету |
| 9 | RCVA_stick[1] | Признак активности основного канала:<br>0 – канал неактивен;<br>1 – канал активен.<br>Устанавливается при установке бита RCVA. Сбрасывается программной записью нуля, если RCVA = 0.<br>В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету |
| 8 | ERR_stick[1] | Ошибка в сообщении:<br>0 – нет ошибок;<br>1 – в последней транзакции возникла ошибка.<br>Устанавливается при установке бита ERR. Сбрасывается программной записью нуля, если ERR = 0.<br>В установленном состоянии не формирует прерывание и не влияет на прием и передачу пакетов. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету |
| 7 | VALMESS_stick[1] | Успешное завершение транзакции в канале:<br>0 – транзакция завершена с ошибкой, либо транзакции нет в канале;<br>1 – транзакция завершена успешно.<br>Устанавливается при установке бита VALMESS. Сбрасывается программной записью нуля, если VALMESS = 0.<br>В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету |

* В режиме StandAlone биты не доступны.

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 6 | RFLAGN_stick* | Получено достоверное слово из канала:<br>0 – нет достоверных слов в канале;<br>1 – в режиме КШ получено достоверное ответное слово;<br>1 – в режиме ОУ получено достоверное командное слово, ответное слово или слово данных в команде управления (сообщение с адресом ОУ равным RTA4-RTA0 или групповое сообщение);<br>1 – в режиме М получено достоверное командное слово, ответное слово или слово данных в команде управления (сообщение с любым адресом ОУ или групповое сообщение).<br>Устанавливается при установке бита RFLAGN. Сбрасывается программной записью нуля, если RFLAGN=0.<br>В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету |
| 5 | RCVB | Признак активности резервного канала:<br>0 – канал неактивен;<br>1 – канал активен |
| 4 | RCVA | Признак активности основного канала:<br>0 – канал неактивен;<br>1 – канал активен |
| 3 | ERR | Ошибка в сообщении:<br>0 – нет ошибок;<br>1 – в последней транзакции возникла ошибка.<br>В режиме ОУ и М, если сброшен бит RERR, автоматически сбрасывается не менее чем через 4 мкс после установки |
| 2 | VALMESS | Успешное завершение транзакции в канале:<br>0 – транзакция завершена с ошибкой, либо транзакции нет в канале;<br>1 – транзакция завершена успешно.<br>В режиме ОУ и М автоматически сбрасывается не менее чем через 4 мкс после установки.<br>В режиме КШ не устанавливается при групповых сообщениях формата 7, 9 и 10 |
| 1 | RFLAGN | Получено достоверное слово из канала:<br>0 – нет достоверных слов в канале;<br>1 – в режиме КШ получено достоверное ответное слово;<br>1 – в режиме ОУ получено достоверное командное слово, ответное слово или слово данных в команде управления (сообщение с адресом ОУ равным RTA4-RTA0 или групповое сообщение);<br>1 – в режиме М получено достоверное командное слово, ответное слово или слово данных в команде управления (сообщение с любым адресом ОУ или групповое сообщение).<br>Между сообщениями бит автоматически сбрасывается в ноль |

* В режиме StandAlone биты не доступны.

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 0 | IDLE | Состояние контроллера:<br>1 – контроллер в неактивном состоянии;<br>0 – контроллер в состоянии обмена сообщениями |

### 20.8.3 ERROR

Таблица 249 – Регистр ошибок ERROR

| Номер | 31...7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | RO | RO | RO | RO | RO | RO | RO |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | PRO ERR | CON ERR | GAP ERR | CSYCERR/ SEQERR | DSYCERR/ SYNCERR | MAN ERR | NORCV |

Таблица 250 – Описание бит регистра ERROR

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...7 | - | Зарезервировано |
| 6 | PROERR | Ошибка в протоколе в режиме КШ:<br>1 – недопустимое слово обнаружено на шине во время обмена сообщениями;<br>0 – нет ошибок |
| 5 | CONERR | Ошибка непрерывности сообщения:<br>1 – передача сообщения не непрерывная;<br>0 – нет ошибок |
| 4 | GAPERR | Недопустимая активность на шине:<br>1 – обнаружена активность на шине в интервале 4 мкс после успешного завершения сообщения.<br>0 – нет ошибок |
| 3 | CSYCERR/ SEQERR | Ошибка синхронизации команды в режиме КШ (CSYCERR):<br>1 – ожидался синхроимпульс команды, а получен синхроимпульс данных;<br>0 – ошибок нет.<br>Ошибка после приема команды в режиме ОУ (SEQERR):<br>1 – обнаружена пауза после приема командного слова с битом 10, равным нулю, или обнаружены слова данных после приема командного слова с битом 10, равным единице;<br>0 – ошибок нет.<br>Ошибка в режиме М (SEQERR):<br>1 –обнаружено отсутствие ожидаемых данных в сообщении или пауза при приеме первого слова;<br>0 – ошибок нет |
| 2 | DSYCERR/ SYNCERR | Ошибка синхронизации данных в режиме КШ (DSYCERR):<br>1 – ожидался синхроимпульс данных, а получен синхроимпульс команды;<br>0 – ошибок нет.<br>Ошибка синхронизации в режиме ОУ и М (SYNCERR):<br>1 – ожидался синхроимпульс команды, а получен синхроимпульс данных или наоборот;<br>0 – ошибок нет |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 1 | MANERR | Ошибка декодирования NRZ кода:<br>1 – ошибка в количестве принятых бит или ошибка в бите контроля четности;<br>0 – ошибок нет |
| 0 | NORCV | Ошибка приема:<br>1 – не получено ответное слово в интервале 14 мкс или не получены ожидаемые данные;<br>0 – ошибок нет |

### 20.8.4 CommandWord1

Таблица 251 – Регистр команды 1 CommandWord1

| Номер | 31...16 | 15…11 | 10 | 9…5 | 4…0 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | Адрес ОУ | Прием/ Передача | Подадрес / Режим управления | Число СД / Код команды |

Таблица 252 – Описание бит регистра CommandWord1

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...16 | - | Зарезервировано |
| 15...11 | Адрес ОУ | Адрес оконечного устройства, которому предназначено командное слово |
| 10 | Прием / передача | Бит приема/передачи:<br>1 – режим работы ОУ-КШ;<br>0 – режим работы КШ-ОУ |
| 9…5 | Подадрес / Режим управления | Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД. В случае передачи команды, содержит код 00000 или 11111 |
| 4…0 | Число СД / Код команды | Содержит количество принимаемых или передаваемых слов данных. В случае передачи команды содержит код команды из таблицы 1 ГОСТ Р 52070-2003 |
| Примечание – В режиме ОУ и М регистр доступен только на чтение, в режиме КШ только на запись | | |

### 20.8.5 CommandWord2

Таблица 253 – Регистр команды 2 CommandWord2

| Номер | 31...16 | 15…11 | 10 | 9…5 | 4…0 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | Адрес ОУ | Прием/ передача | Подадрес / Режим управления | Число СД / Код команды |

Таблица 254 – Описание бит регистра CommandWord2

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...16 | - | Зарезервировано |
| 15...11 | Адрес ОУ | Адрес оконечного устройства, которому предназначено командное слово |
| 10 | Прием/передача | Бит приема/передачи:<br>1 – режим работы ОУ-ОУ;<br>0 – командное слово не используется |
| 9..5 | Подадрес | Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД |
| 4..0 | Число СД | Содержит количество принимаемых или передаваемых слов данных |
| Примечание – в режиме ОУ и М регистр доступен только на чтение и содержит второе командное слово транзакции ОУ-ОУ. В режиме КШ регистр доступен только на запись и используется для транзакции ОУ-ОУ, если установлен в единицу бит 10 | | |

### 20.8.6 ModeData

Таблица 255 – Регистр слова данных команды управления ModeData

| Номер | 31...16 | 15…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | Слово данных команды управления |

Таблица 256 – Описание бит регистра ModeData

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31..16 | - | Зарезервировано |
| 15..0 | ModeData | Содержит принятое или передаваемое слово данных в команде управления |

### 20.8.7 StatusWord1

Таблица 257 – Регистр ответного слова 1 StatusWord1

| Номер | 31...16 | 15...11 | 10 | 9 | 8 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | Адрес ОУ | Ошибка в сообщ. | Пер.ОС | Запр. На обсл. |

| Номер | 7…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 |
| | - | Прин ГК | Абонзан. | Неисп. Абон. | Прин упр. Инт. | Неисп ОУ |

Таблица 258 – Описание бит регистра StatusWord1

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...16 | - | Зарезервировано |
| 15...11 | Адрес ОУ | Адрес ОУ, от которого принято ответное слово в режиме КШ. Адрес ОУ, которое передает ответное слово в режиме ОУ |
| 10 | Ошибка в сообщ. | Ошибка в сообщении |
| 9 | Пер.ОС | Передача ответного слова |
| 8 | Запр. На обсл. | Запрос на обслуживание |
| 7..5 | - | Зарезервировано |
| 4 | Прин. ГК | Принята групповая команды |
| 3 | Абон. Зан. | Абонент занят |
| 2 | Неисп. Абон. | Неисправность абонента |
| 1 | Прин. Упр. Инт. | Принято управление интерфейсом |
| 0 | Неисп. ОУ | Неисправность ОУ |

Примечания

1 Для режима КШ по чтению регистр содержит первое принятое ответное слово, а в случае транзакции ОУ-ОУ по чтению содержит ОС, принятое от принимающего ОУ (второе ОС на шине).

2 Для режима М по чтению регистр содержит первое ОС в транзакции на шине.

3 Для режима ОУ по чтению регистр содержит ОС, принятое от второго ОУ в транзакции ОУ-ОУ (это может быть, как принимающее, так и передающее ОУ)

### 20.8.8 StatusWord2

Таблица 259 – Регистр ответного слова 2 StatusWord2

| Номер | 31…16 | 15…11 | 10 | 9 | 8 | 7..5 |
|---|---|---|---|---|---|---|
| Доступ | U | RO | RO | RO | RO | U |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | Адрес ОУ | Ошибка в сообщ. | Пер. ОС | Запр. На обсл. | - |

| Номер | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | RO | RO | RO | RO | RO |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | ПринГК | Абон зан. | Неисп. абон. | Прин упр. Инт. | Неисп ОУ |

Таблица 260 – Описание бит регистра StatusWord2

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...16 | - | Зарезервировано |
| 15...11 | Адрес ОУ | Адрес ОУ, передающего данные в транзакции ОУ-ОУ |
| 10 | Ошибка в сообщ. | Ошибка в сообщении |
| 9 | Пер. ОС | Передача ответного слова |
| 8 | Запр. На обсл. | Запрос на обслуживание |
| 7…5 | | Зарезервировано |
| 4 | Прин. ГК | Принята групповая команды |
| 3 | Абон. Зан. | Абонент занят |
| 2 | Неисп. Абон. | Неисправность абонента |
| 1 | Прин. Упр. Инт. | Принято управление интерфейсом |
| 0 | Неисп. ОУ | Неисправность ОУ |

Примечания

1 Для режима КШ доступен только на чтение и в случае транзакции ОУ-ОУ содержит ОС, принятое от передающего ОУ (первое ОС на шине);

2 Для режима М доступен только на чтение и содержит второе ОС в транзакции ОУ-ОУ на шине;

3 В режиме ОУ регистр не используется

### 20.8.9 INTEN

Таблица 261 – Регистр разрешения прерываний INTEN

| Номер | 31…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | | 0 | 0 | 0 | 0 |
| | - | ERRIE | VALMESSIE | RFLAGNIE | IDLEIE |

Таблица 262 – Описание бит регистра INTEN

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31..4 | | Зарезервировано |
| 3 | ERRIE | Прерывание при возникновении ошибки в сообщении.<br>0 – прерывание маскировано<br>1 – прерывание разрешено, это позволяет генерировать прерывание при возникновении ошибок в сообщении |
| 2 | VALMESSIE | Прерывание при успешном завершении транзакции в канале.<br>0 – прерывание маскировано<br>1 – прерывание разрешено, это позволяет генерировать прерывание при успешном завершении обмена данными в канале |
| 1 | RFLAGNIE | Прерывание при приеме достоверного слова.<br>0 – прерывание маскировано<br>1 – прерывание разрешено, это позволяет генерировать прерывание при приеме достоверного ОС в режиме КШ или достоверного КС, ОС или слова данных команды управления в режиме ОУ |
| 0 | IDLEIE | Прерывание неактивности контроллера.<br>0 – прерывание маскировано<br>1 – прерывание разрешено, это позволяет генерировать прерывание по переходу контроллера в неактивное состояние |

### 20.8.10 MSG

Таблица 263 – Регистр декодирования сообщений MSG

| Номер | 15,14 | 13…0 |
|---|---|---|
| Доступ | U | RO |
| Сброс | | 0 |
| | - | Код сообщения |
| Примечание – регистр содержит код сообщения, полученного в режиме ОУ или М, и доступен только на чтение. В режиме КШ регистр не используется. Регистр обновляется каждый раз при получении нового достоверного КС | | |

Таблица 264 – Коды сообщений регистра MSG

| Код сообщения | CommandWord1 | CommandWord2 |
|---|---|---|
| *Команды обмена данными* | 15:11 10 9:5 4:0 | 15:11 10 9:5 4:0 |
| 0001 Команда приема КШ-ОУ, не групповая | RTA 0 00001-11110 XXXXX | |
| 0080 Команда приема КШ-ОУ, групповая | 11111 0 00001-11110 XXXXX | |
| 0004 Команда приема ОУ-ОУ, не групповая | RTA 0 00001-11110 XXXXX | XXXXX 1 00001-11110 XXXXX |
| 0100 Команда приема ОУ-ОУ, групповая | 11111 0 00001-11110 XXXXX | не RTA 1 00001-11110 XXXXX |
| 0402 Команда передачи ОУ-КШ | RTA 1 00001-11110 XXXXX | |
| 1008 Команда передачи ОУ-ОУ, не групповая | не F 0 00001-11110 XXXXX | RTA 1 00001-11110 XXXXX |
| 0200 Команда передачи ОУ-ОУ, групповая | 11111 0 00001-11110 XXXXX | RTA 1 00001-11110 XXXXX |
| *Команды управления* | | |
| 0410 Код 0-15 K=1 нет данных, не групповая | RTA 1 00000 11111 0XXXX | |
| 0400 Код 0-15 K=1 нет данных, групповая | 11111 1 00000 11111 0XXXX | |
| 2420 Код 16-31 K=1 с данными, не групповая | RTA 1 00000 11111 1XXXX | |
| 0040 Код 16-31 K=0 с данными, не групповая | RTA 0 00000 11111 1XXXX | |
| 0800 Код 16-31 K=0 с данными, групповая | 1111 0 00000 11111 1XXXX | |

### 20.8.11 DATA

Память принимаемых/передаваемых данных.

Примечание – Данные читаются из памяти или записываются в память в соответствии с подадресом (биты с 9 по 5) достоверного командного слова. Каждому подадресу соответствует 32×16 ячеек памяти на прием и 32×16 ячеек памяти на передачу. Общий объем памяти данных 2K×16.

# 21 Таймеры общего назначения

В микросхеме реализовано четыре блока таймеров общего назначения, каждый из которых может быть использован для широкого спектр 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.

Каждый блок таймера содержит 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.

## 21.1 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.

### 21.1.1 Структурная схема

Структурная схема блока «Таймер» представлена на рисунке 61.

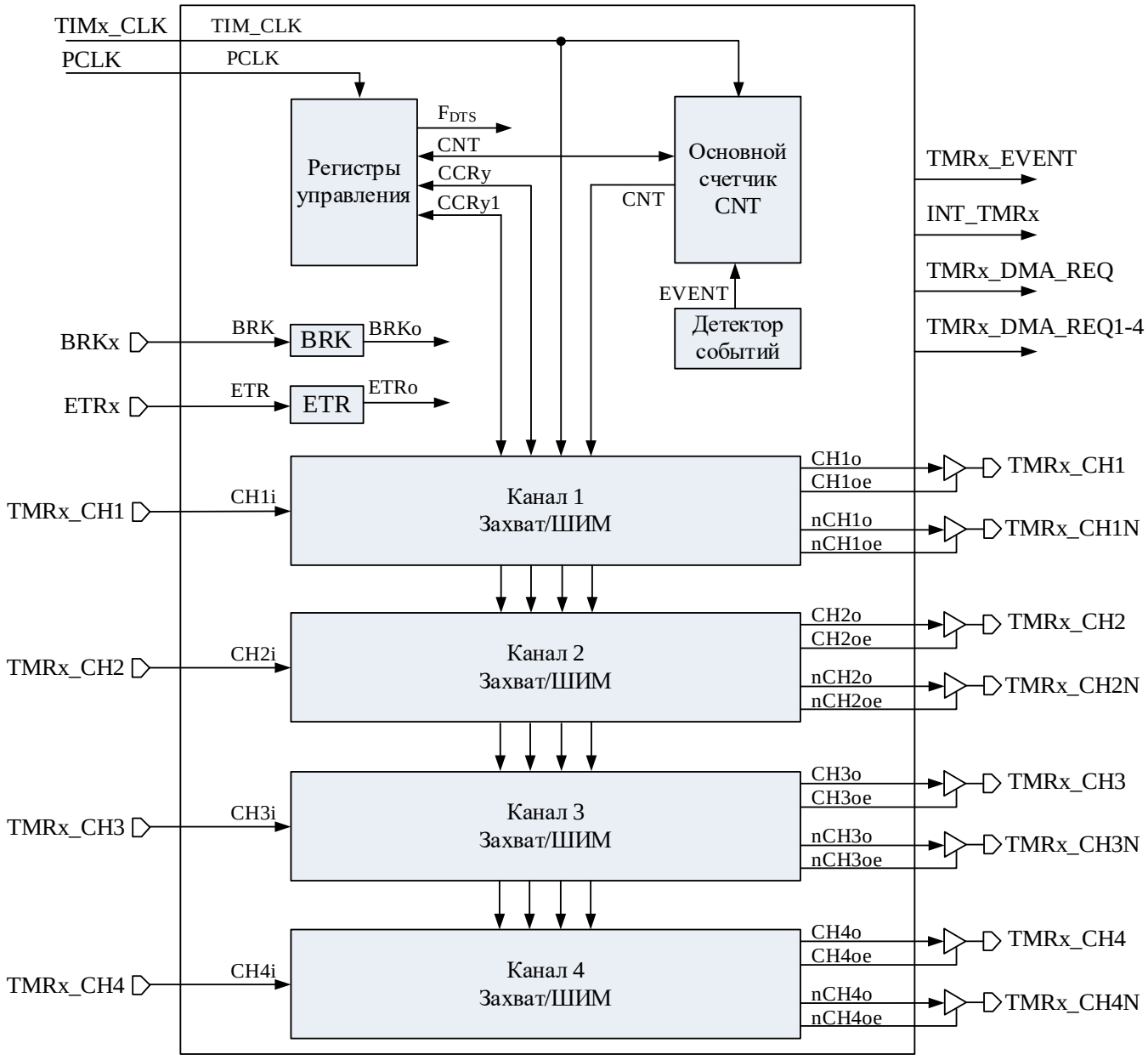



Рисунок 61 – Структурная схема блока Таймер

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.

## 21.2 Базовый блок таймера

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

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

Сигналом для изменения основного счетчика CNT может служить как внутренняя частота TIM_CLK, так и события в других счетчиках, либо внешние входные сигналы (см. подраздел 21.3 «Источник событий для счета»).

### 21.2.1 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 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 PER_CLOCK: бит 14 для таймера 1, бит 15 для таймера 2, бит 16 для таймера 3, бит 19 для таймера 4. Задание тактовой частоты TIM_CLK осуществляется в регистре TIM_CLOCK (для таймера 4 в регистре UART_CLOCK): в поле TIMxBRG[7:0] устанавливается коэффициент деления тактовой частоты HCLK для формирования частоты TIM_CLK, разрешение подачи частоты TIM_CLK на блок таймера управляется битом TIMxCLKEN.

После подачи тактовых сигналов можно приступать к работе с таймером.

### 21.2.2 Инициализация основного счетчика таймера

Чтобы запустить работу основного счетчика необходимо задать:

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

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

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

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

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

### 21.2.3 Режимы счета

#### 21.2.3.1 *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
```

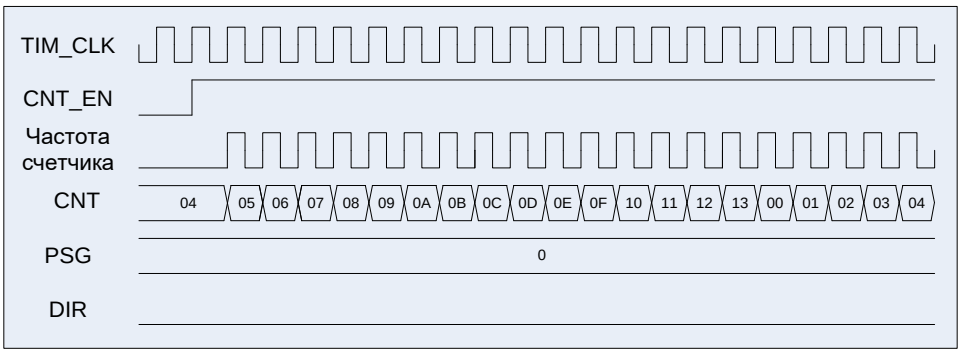



Рисунок 62 – Диаграммы работы таймера, счет прямой от 0 до 0x13, стартовое значение 0x04

#### 21.2.3.2 *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*

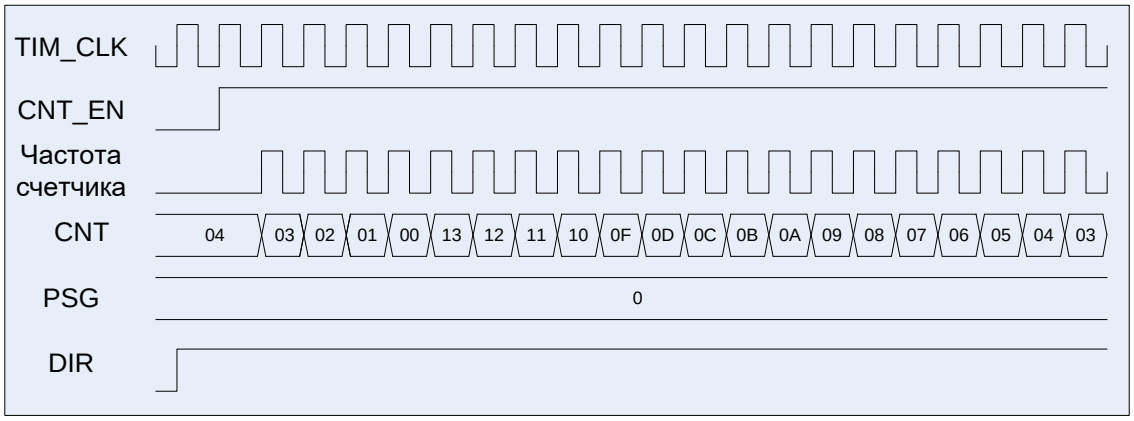



Рисунок 63 – Диаграммы работы таймера, счет обратный от 0x13 до 0, стартовое значение 0x04

### 21.2.3.3 *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*

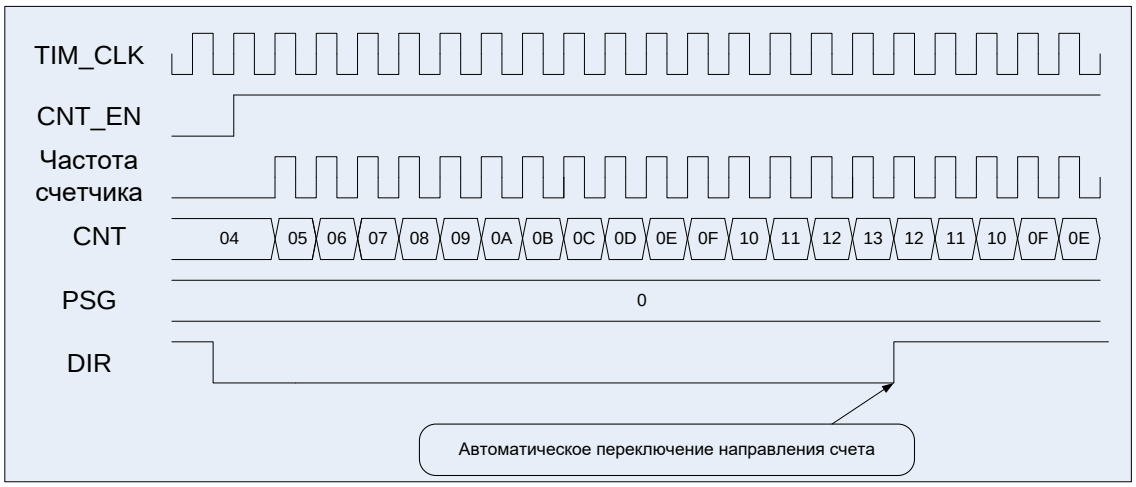



Рисунок 64 – Диаграммы работы таймера, счет двунаправленный, сначала прямой

### 21.2.3.4 *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*

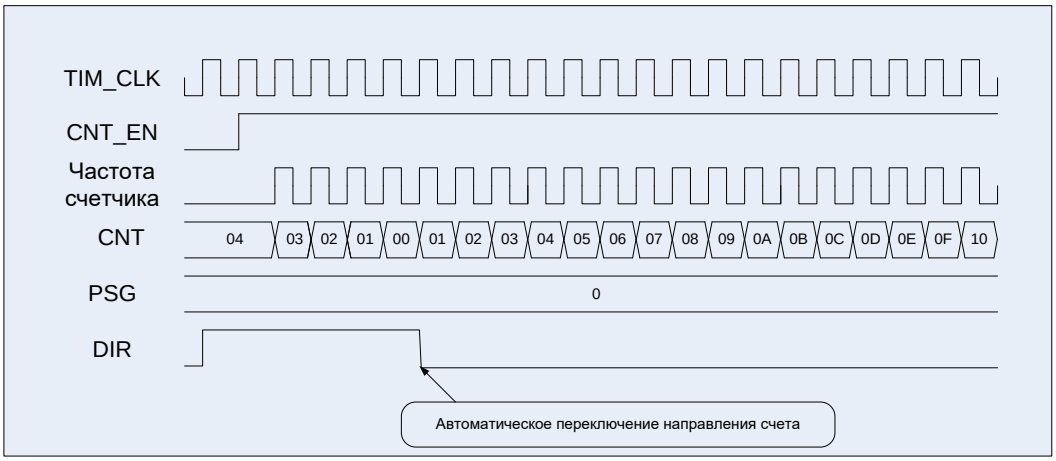



Рисунок 65 – Диаграммы работы таймера, счет двунаправленный, сначала обратный

### 21.2.4 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].

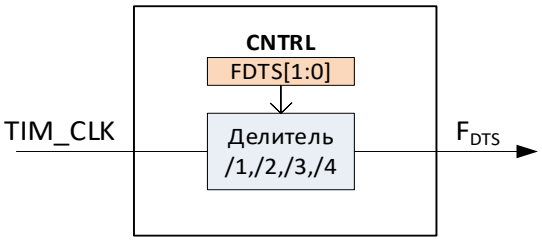



Рисунок 66 – Схема формирования тактовой частоты $F_{DTS}$

Диаграмма возможных частот $F_{DTS}$ приведена на рисунке 67.

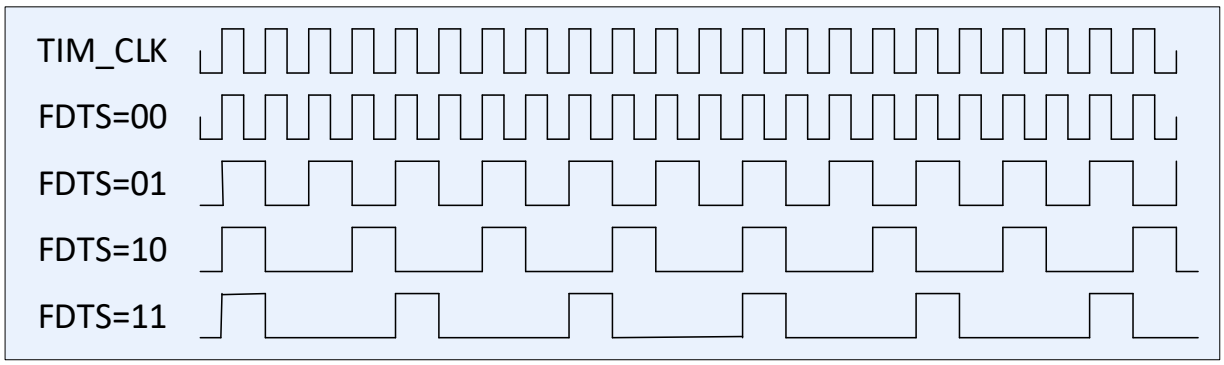



Рисунок 67 – Диаграмма тактовой частоты $F_{DTS}$ в зависимости от значения FDTS[1:0] в регистре CNTRL

## 21.3 Источник событий для счета

Тактирование основного счетчика таймера может осуществляться от следующих источников:

– внутренний тактовый сигнал (TIM_CLKd);
– событие в другом таймере (CNT==ARR);
– внешний тактовый сигнал, «Режим 1»: событие переднего фронта на входе канала CHyi;
– внешний тактовый сигнал, «Режим 2»: событие переднего или заднего фронта на входе ETR.

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

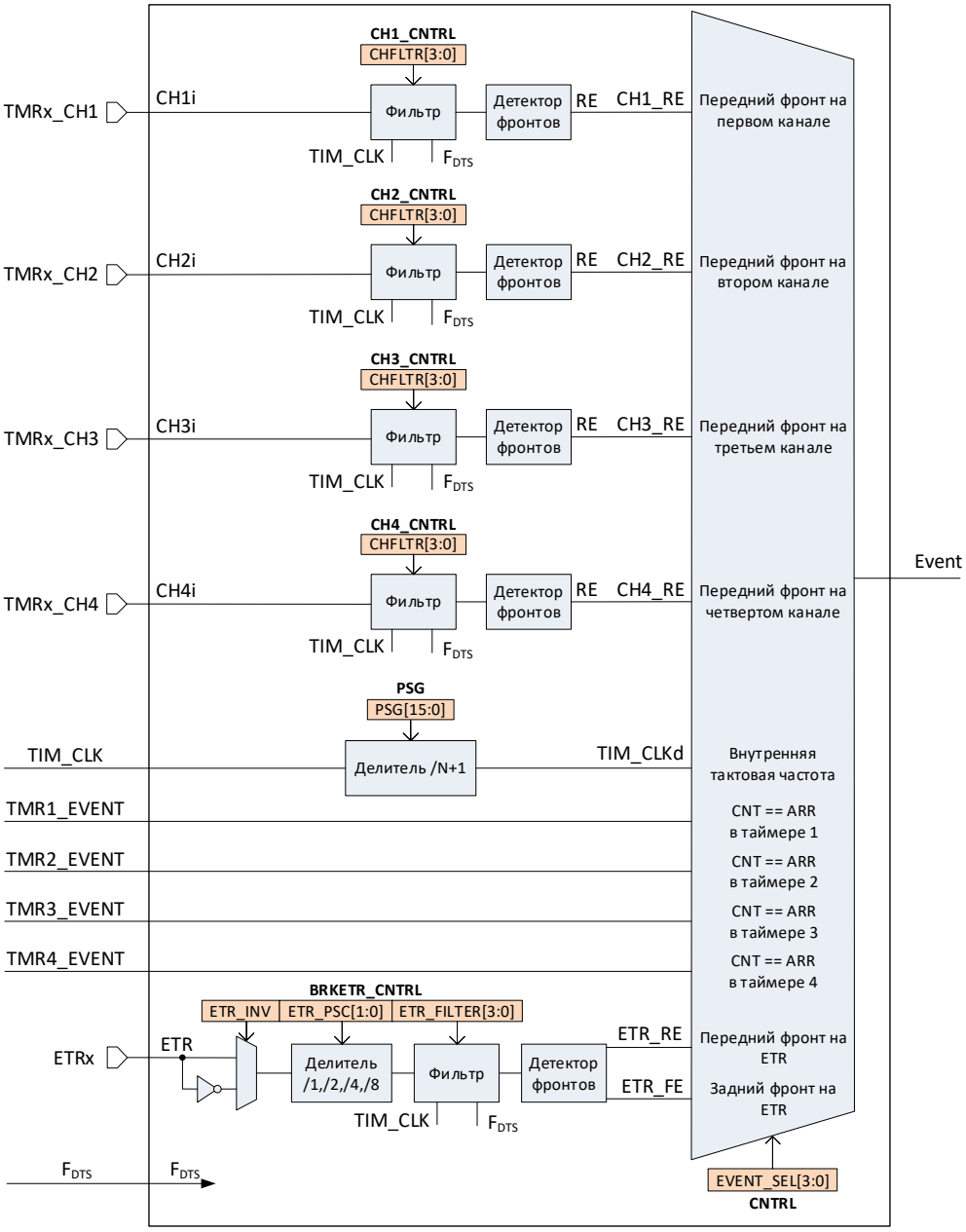



Рисунок 68 – Структурная схема формирования события для счета

### 21.3.1 Внутренний тактовый сигнал (TIM_CLKd)

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

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

$$TIM_CLKd = \frac{TIM_CLK}{PSG + 1} \quad (5)$$

Значение регистра PSG можно изменять даже во время работы счетчика, новое значение предделителя вступит в силу сразу после записи. На рисунках 69 и 70 приведены диаграммы работы счетчика при обновлении значения PSG.

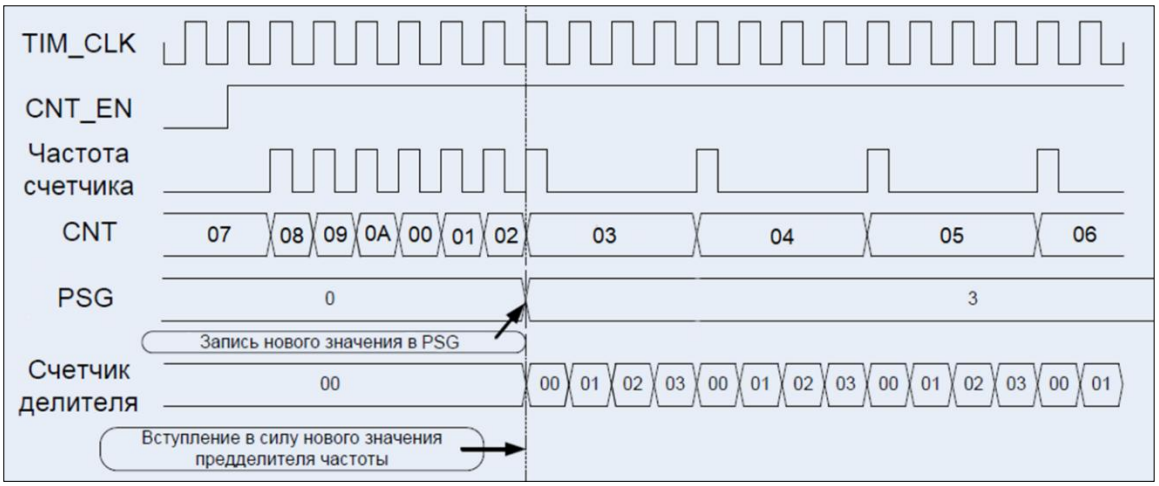



Рисунок 69 – Диаграмма работы счетчика: счет прямой
(CNT_MODE[1:0] = 00, EVENT_SEL[3:0] = 0000, DIR = 0)

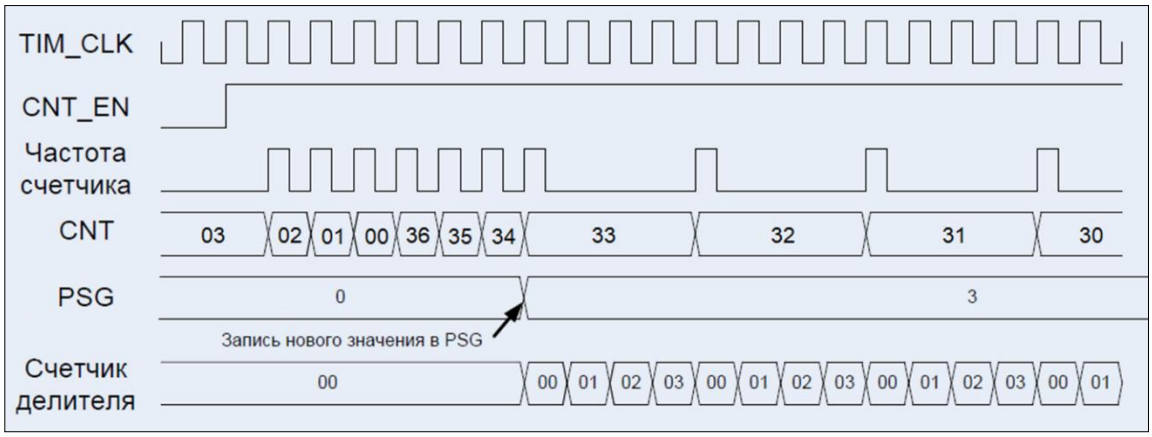



Рисунок 70 – Диаграмма работы счетчика: счет обратный (CNT_MODE[1:0] = 00, EVENT_SEL[3:0] = 0000, DIR = 1)

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

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

У каждого таймера имеется выход запуска TMRx_EVENT, который соединен с входами других таймеров. Тактирование от другого таймера выбирается, когда EVENT_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_EVENT, который устанавливается при CNT == ARR. Пересинхронизация сигнала TMRx_EVENT (CNT == ARR) с одного таймера на другой происходит с задержкой один такт частоты TIM_CLK.

Синхронизация таймеров возможна в различных режимах. Пример каскадного соединения таймеров и диаграммы работы данных таймеров приведены на рисунках 71 и 72 соответственно.

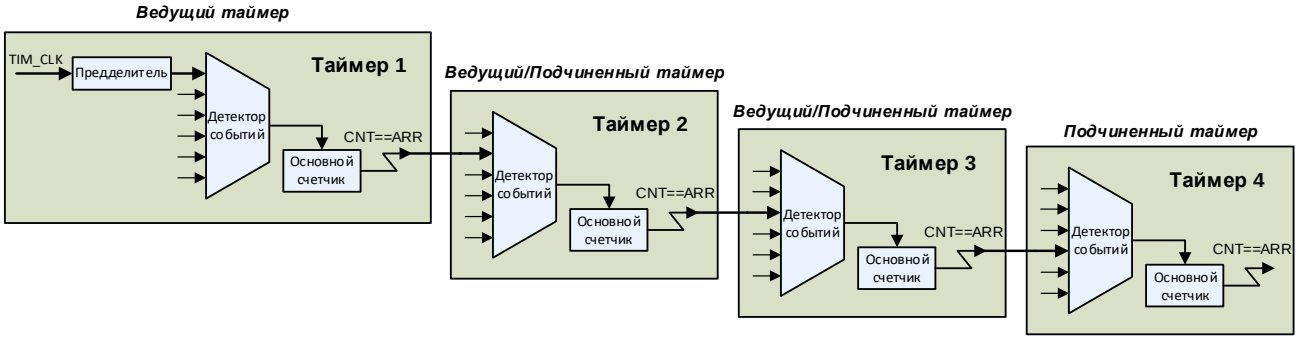



Рисунок 71 – Пример каскадного соединения таймеров

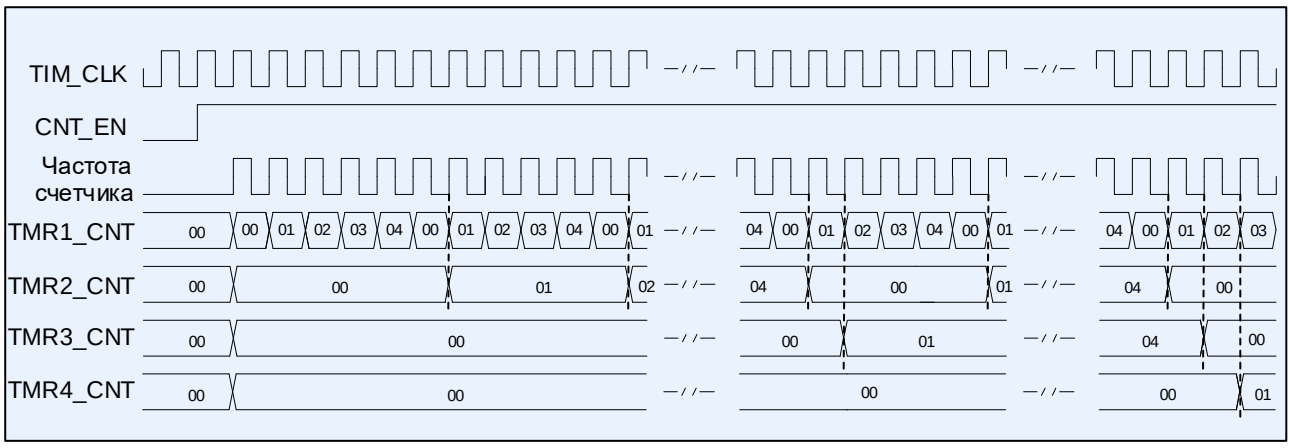



Рисунок 72 – Диаграммы работы четырех таймеров в каскаде
TIMER1: DIR = 0, EVENT_SEL[3:0] = 0000, CNT_MODE[1:0] = 00;
TIMER2: DIR = 0, EVENT_SEL[3:0] = 0001, CNT_MODE[1:0] = 10;
TIMER3: DIR = 0, EVENT_SEL[3:0] = 0010, CNT_MODE[1:0] = 10;
TIMER4: DIR = 0, EVENT_SEL[3:0] = 0011, CNT_MODE[1:0] = 10

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

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

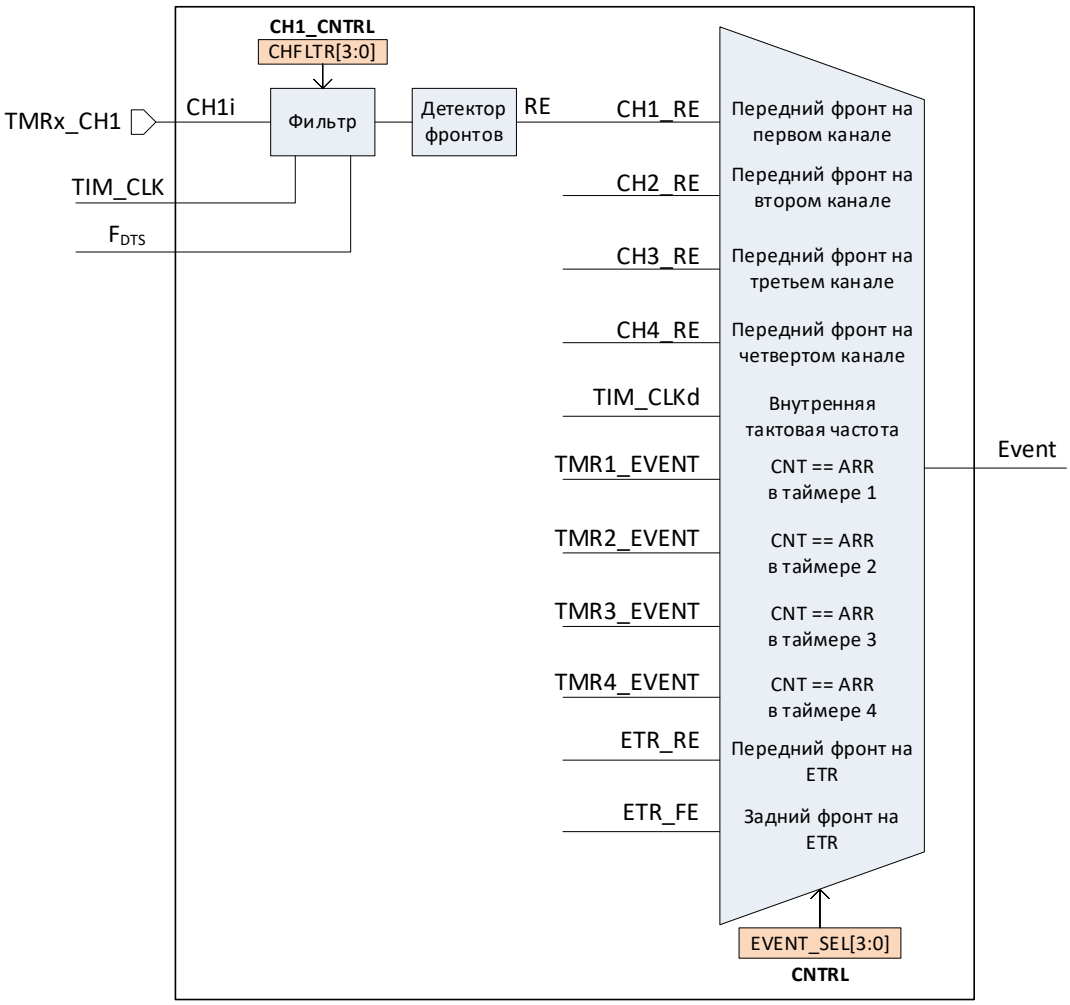



Рисунок 73 – Схема тактирования сигналом со входа первого канала

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 CHFLTR[3:0] регистра CHy_CNTRL.

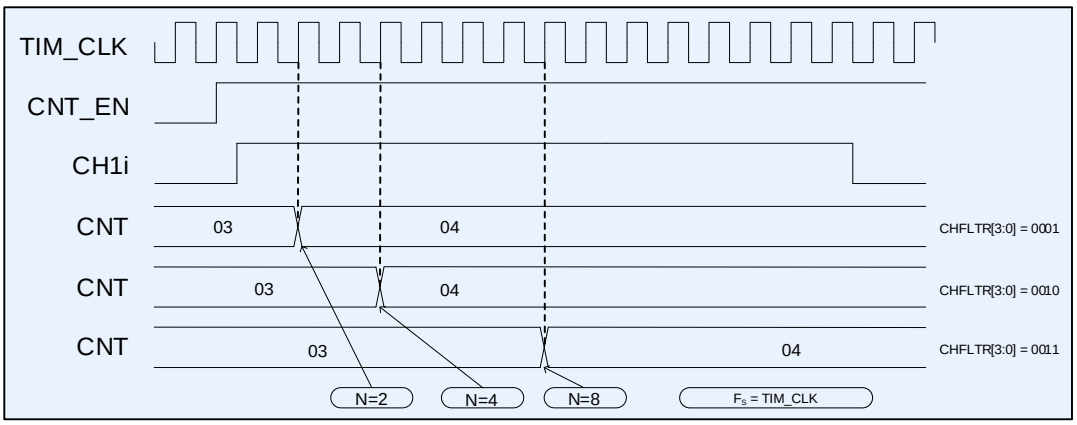



Рисунок 74 – Диаграмма внешнего тактирования с разными вариантами фильтра

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

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

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

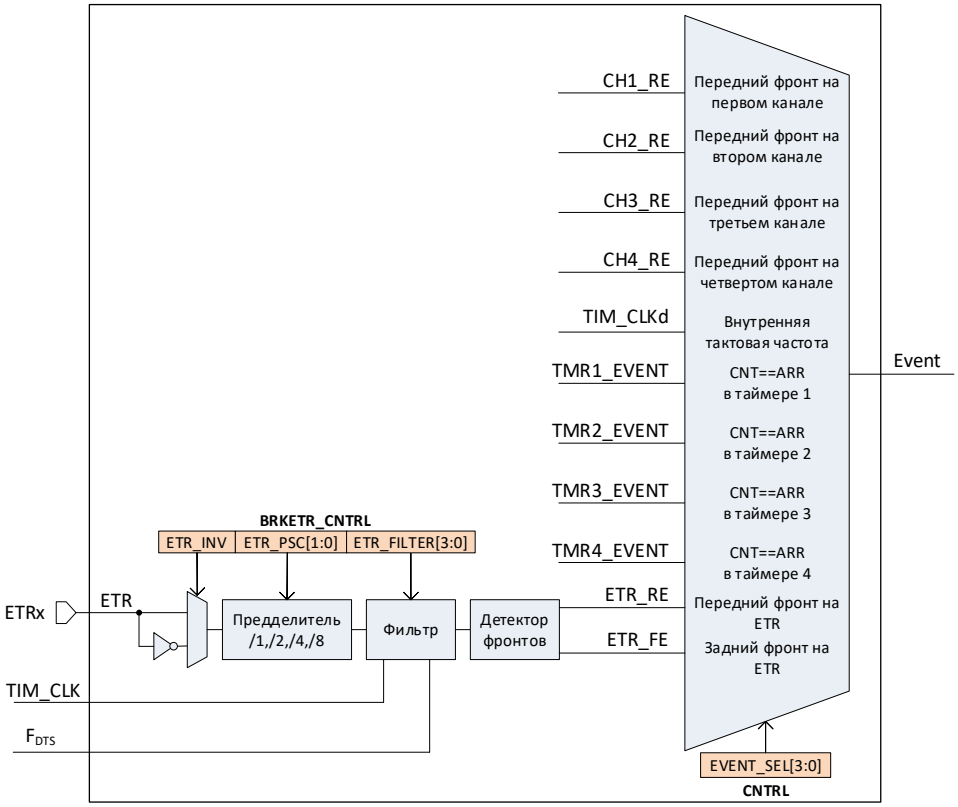



Рисунок 75 – Схема тактирования сигналом со входа ETR

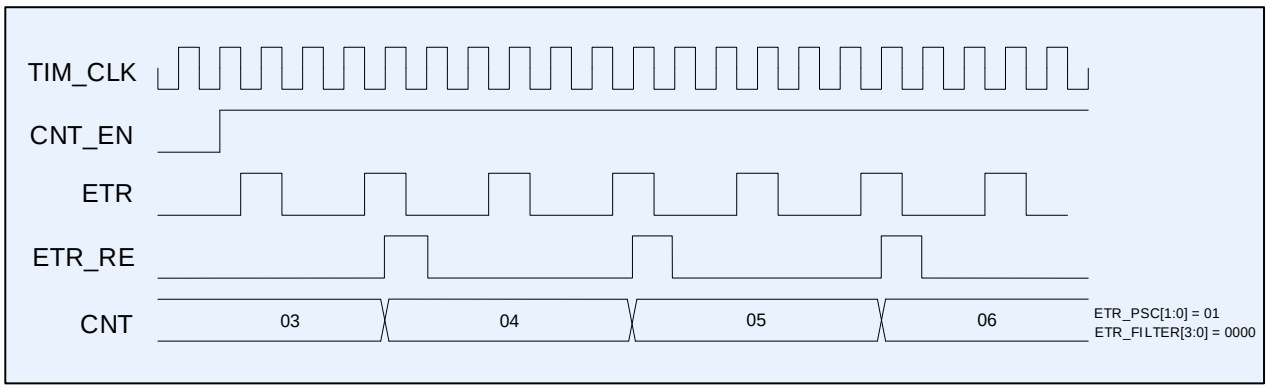



Рисунок 76 – Диаграмма тактирования со входа ETR, EVENT_SEL[3:0] = 1000

## 21.4 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 CCRy (CCR) и CCRy1 (CCR1). Регистрация событий осуществляется только по входам CHyi (выводы, настроенные в функцию TMRx_CHy).

Структурная схема блока захвата представлена на рисунке 77.

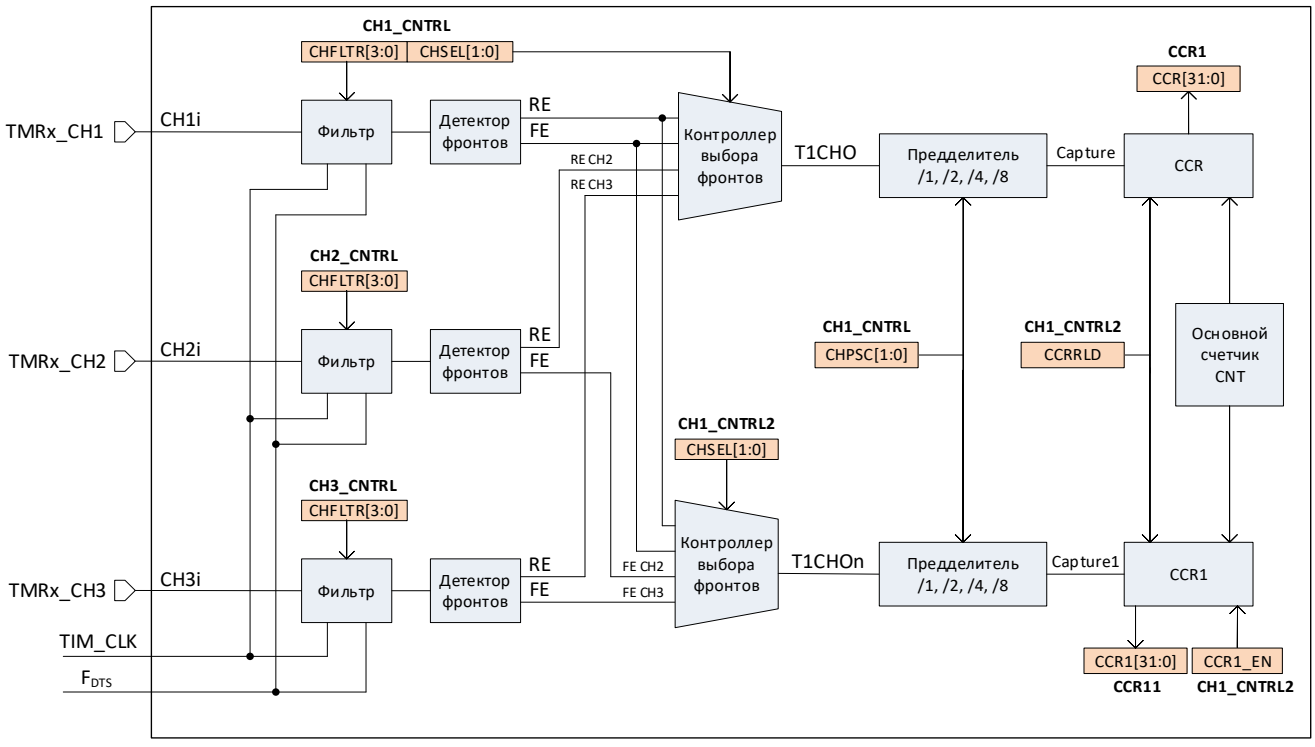



Рисунок 77 – Структурная схема блока захвата на примере канала 1

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

Внешний сигнал со входа CHyi сначала поступает в блок фильтра. Данный блок позволяет отфильтровать входной сигнал с целью устранения импульсов, длительность которых меньше заданного порога (см. подраздел 21.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле CHFLTR[3:0] регистра CHy_CNTRL.

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

В блоке «Контроллер выбора фронтов» производится выбор используемого для захвата сигнала между положительным фронтом канала, отрицательным фронтом канала и положительными и отрицательными фронтами сигналов от других каналов. Настройка блока «Контроллер выбора фронтов» для регистра CCRy осуществляется в поле CHSEL[1:0] регистра CHy_CNTRL, а для регистра CCRy1 – в поле CHSEL[1:0] регистра CHy_CNTRL2. Выбранный для захвата сигнал поступает в предварительный делитель, который в зависимости от значения в поле CHPSC[1:0] регистра CHy_CNTRL позволяет фиксировать все события, либо каждое второе, каждое четвертое или каждое восьмое событие.

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

На рисунке 78 показан пример захвата значения основного счетчика CNT в регистр CCRy по положительному фронту на входе канала, а в регистр CCRy1 – по отрицательному фронту на входе канала.

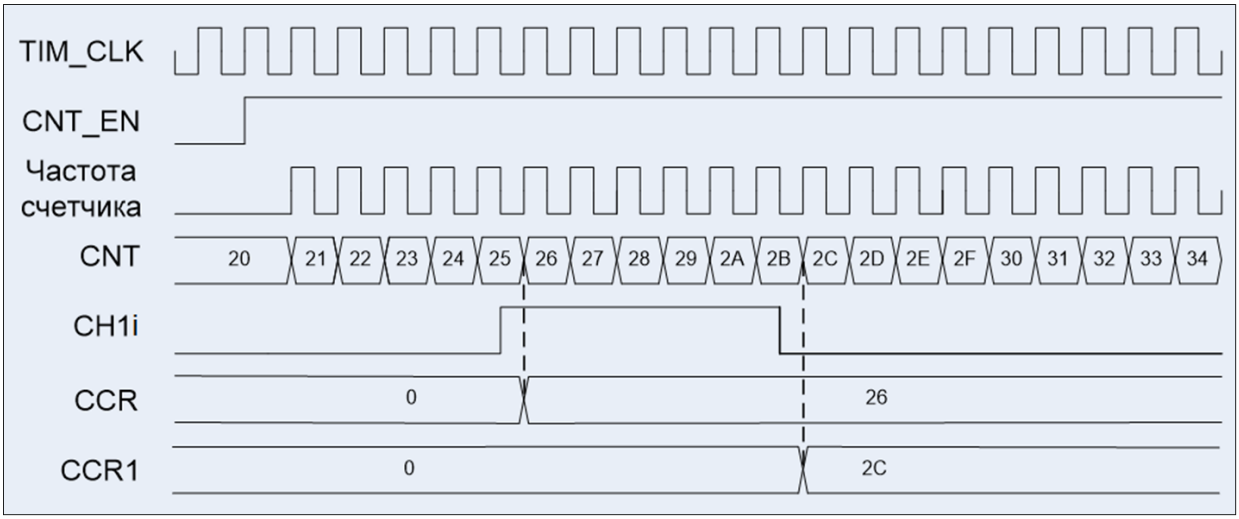



Рисунок 78 – Диаграмма захвата события со входа канала 1

По событию захвата на определенном канале в регистре IE можно разрешить выработку прерываний, а в регистрах DMA_RE и DMA_RE1-DMA_RE4 можно разрешить формирование запросов DMA.

Между формированием события захвата и записью текущего значения основного счетчика CNT в регистры CCRy и CCRy1 может быть установлена задержка с помощью бита EV_DELAY в регистре CHy_CNTRL2. Если бит EV_DELAY равен «0», то сначала формируется событие захвата, а затем через один такт сигнала синхронизации TIM_CLK выполняется запись CNT в регистры CCRy и CCRy1. Если бит EV_DELAY равен «1», то обновление информации в регистрах CCRy и CCRy1 происходит синхронно с событием захвата. При реализации чтения регистров CCRy и CCRy1 по событию захвата рекомендуется устанавливать бит EV_DELAY в «1».

## 21.5 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 79.

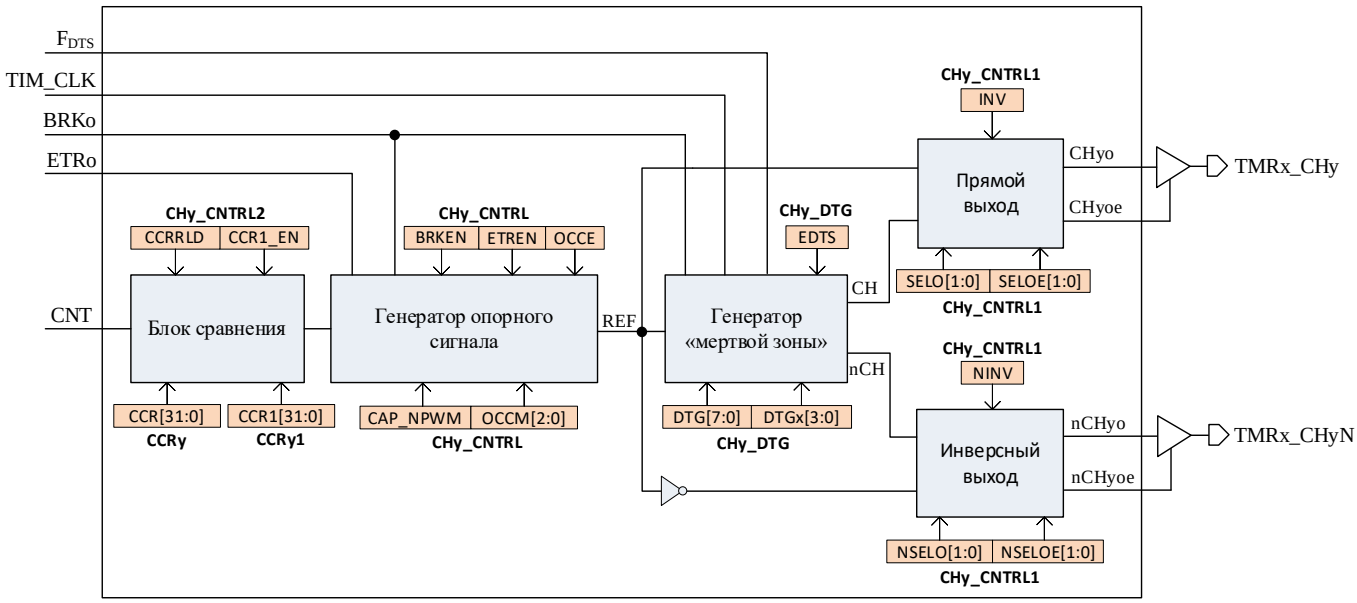



Рисунок 79 – Структурная схема блока формирования ШИМ

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

### 21.5.1 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 CCRy (CCR), CCRy1 (CCR1) и основного счетчика CNT. Формат выработки сигнала REF устанавливается в поле OCCM[2:0] регистра управления каналом таймера CHy_CNTRL.

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

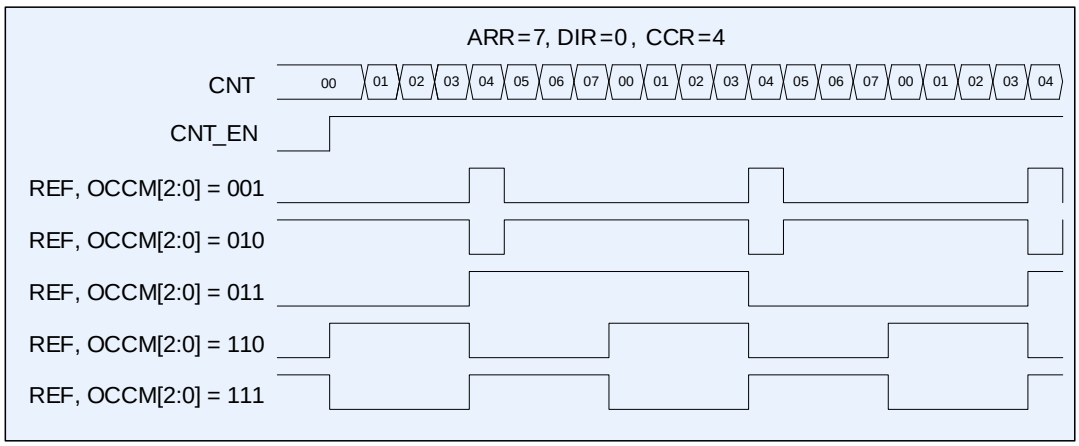



Рисунок 80 – Диаграмма работы в режиме ШИМ, CCR1_EN=0

Если в регистре CHy_CNTRL2 бит CCR1_EN = 1, то для формирования сигнала REF задействуются оба результата сравнения значения в регистрах CCRy, CCRy1 и основного счетчика CNT.

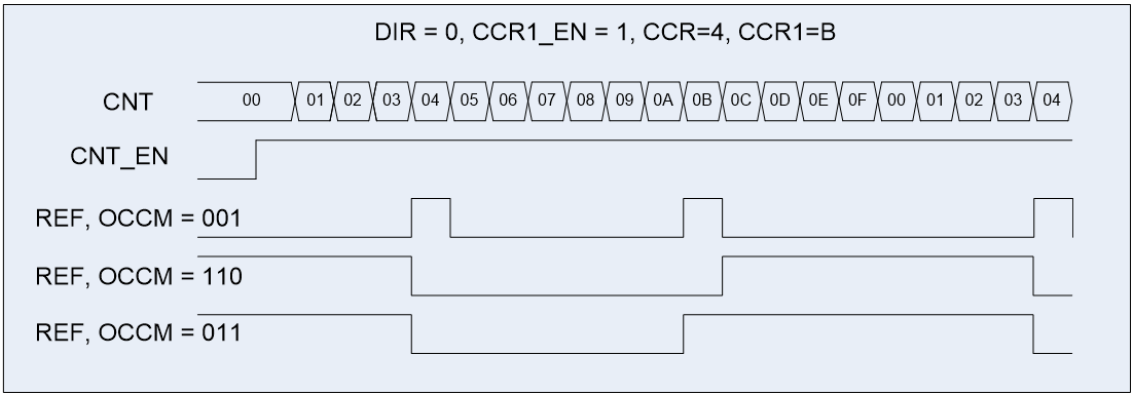



Рисунок 81 – Диаграмма работы в режиме ШИМ, CCR1_EN = 1

Запись новых значений в регистры CCRy и CCRy1 осуществляется немедленно, если в регистре CHy_CNTRL2 бит CCRRLD установлен в «0», иначе регистры CCRy и CCRy1 получат новые значения только при CNT == 0. Процесс обновления значений в регистрах CCRy и CCRy1 обозначается в регистре CHy_CNTRL с помощью флагов WR_CMPL и WR_CMPL1, соответственно. На время выполнения записи флаг WR_CMPL/WR_CMPL1 устанавливается в «1», по окончании записи флаг WR_CMPL/WR_CMPL1 сбрасывается в «0».

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

Для разрешения сброса сигнала REF по входу ETR необходимо установить бит ETREN и OCCE в регистре CHy_CNTRL. Активный уровень на входе ETR сбрасывает сигнал REF в «0». После снятия активного уровня на входе ETR сигнал REF остается в «0» до следующего события установки REF в «1», рисунок 82.

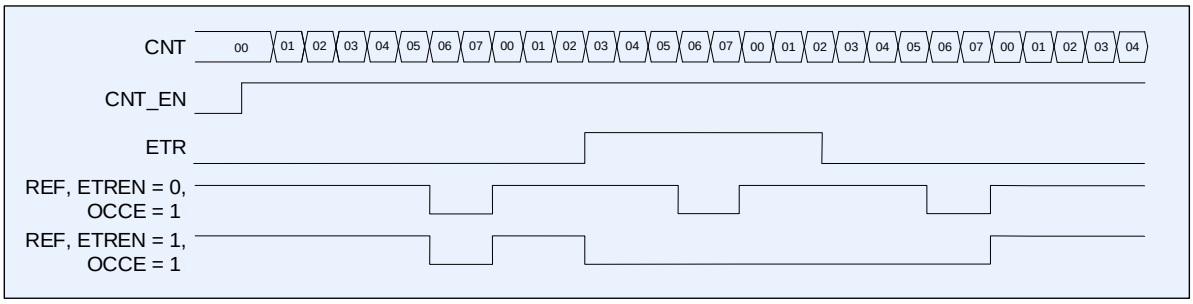



Рисунок 82 – Диаграмма сброса сигнала REF по выводу ETR

Для разрешения сброса сигнала REF по входу BRK необходимо установить бит BRKEN в регистре CHy_CNTRL. Активный уровень на входе BRK сбрасывает сигнал REF в «0» путем маскирования. После снятия активного уровня на входе BRK генерация сигнала REF сразу же восстанавливается, рисунок 83.

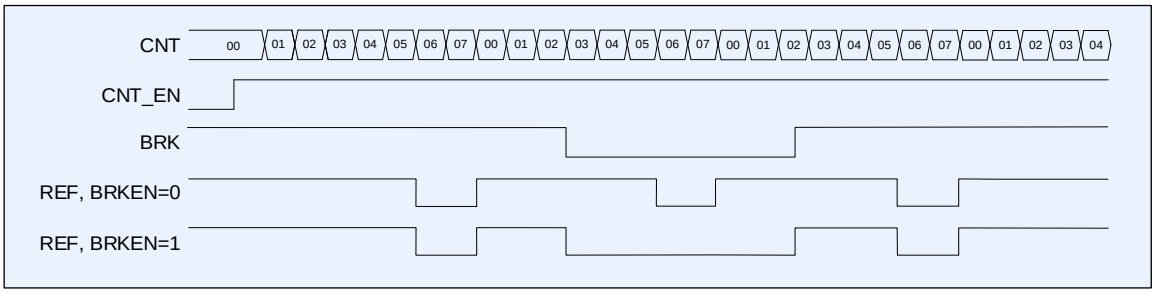



Рисунок 83 – Диаграмма сброса сигнала REF по выводу BRK

### 21.5.2 Генератор «мертвой зоны»

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

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

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

$$DTGdel = DTG \cdot (DTGx + 1)\,, \qquad (6)$$

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.

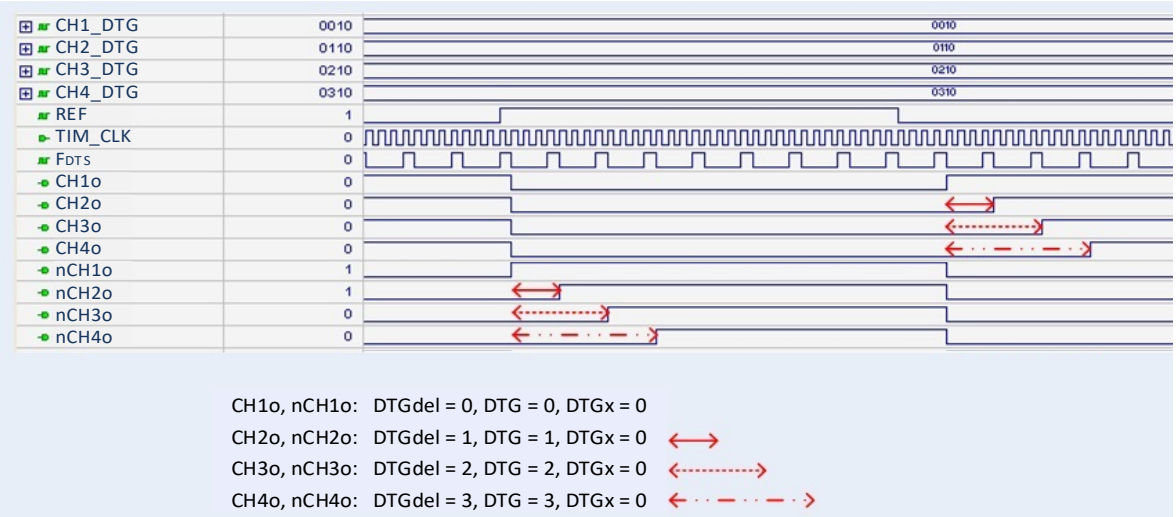



Рисунок 84 – Диаграмма работы блока DTG

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

### 21.5.3 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.

## 21.6 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_FILTER[3:0] регистра BRKETR_CNTRL, для входов каналов CHyi – в поле CHFLTR[3:0] регистра CHy_CNTRL. Значение в данных полях позволяет настроить два параметра фильтра:

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

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

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

Таким образом, настраивая частоту $F_S$ и количество выборок N, задается минимальная длительность импульсов входного сигнала, которые не будут

отфильтрованы. Диаграмма работы фильтра при использовании частоты TIM_CLK приведена на рисунке 85, частота $F_S$ = TIM_CLK, количество выборок N = 4.

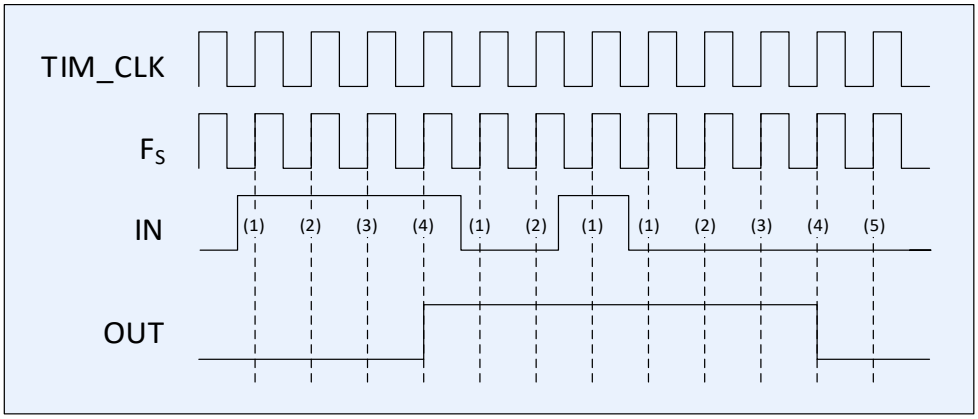



Рисунок 85 – Диаграмма работы фильтра, $F_S$ = TIM_CLK, N = 4

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. пункт 21.2.4 «Тактовая частота $F_{DTS}$»).

Диаграмма работы фильтра при использовании частоты $F_{DTS}$ приведена на рисунке 86, частота выборки $F_S = F_{DTS}/2$, количество выборок N = 6.

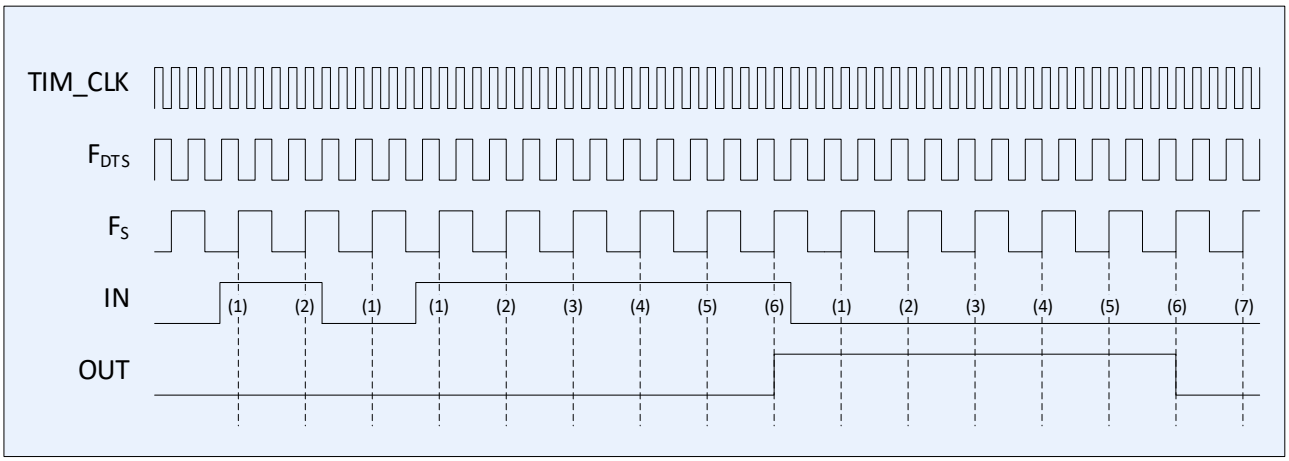



Рисунок 86 – Диаграмма работы фильтра, $F_S = F_{DTS}/2$, N = 6

Возможные варианты настройки блока фильтра приведены в таблице 265.

Таблица 265 – Возможные конфигурации фильтра

| CHFLTR[3:0], ETR_FILTER[3:0] | Частота выборки Fs | Количество выборок N | Минимальная длительность импульсов, которые не будут отфильтрованы |
|---|---|---|---|
| 0000 | $F_{DTS}$ | 1 | - |
| 0001 | TIM_CLK | 2 | $2 \times T_{TIM_CLK}$ |
| 0010 | TIM_CLK | 4 | $4 \times T_{TIM_CLK}$ |
| 0011 | TIM_CLK | 8 | $8 \times T_{TIM_CLK}$ |
| 0100 | $F_{DTS}/2$ | 6 | $12 \times T_{FDTS}$ |
| 0101 | $F_{DTS}/2$ | 8 | $16 \times T_{FDTS}$ |
| 0110 | $F_{DTS}/4$ | 6 | $24 \times T_{FDTS}$ |
| 0111 | $F_{DTS}/4$ | 8 | $32 \times T_{FDTS}$ |

| CHFLTR[3:0], ETR_FILTER[3:0] | Частота выборки Fs | Количество выборок N | Минимальная длительность импульсов, которые не будут отфильтрованы |
|---|---|---|---|
| 1000 | $F_{DTS}/8$ | 6 | $48 \times T_{FDTS}$ |
| 1001 | $F_{DTS}/8$ | 8 | $64 \times T_{FDTS}$ |
| 1010 | $F_{DTS}/16$ | 5 | $80 \times T_{FDTS}$ |
| 1011 | $F_{DTS}/16$ | 6 | $96 \times T_{FDTS}$ |
| 1100 | $F_{DTS}/16$ | 8 | $128 \times T_{FDTS}$ |
| 1101 | $F_{DTS}/32$ | 5 | $160 \times T_{FDTS}$ |
| 1110 | $F_{DTS}/32$ | 6 | $192 \times T_{FDTS}$ |
| 1111 | $F_{DTS}/32$ | 8 | $256 \times T_{FDTS}$ |

## 21.7 Флаги состояний, прерывания и запросы DMA

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

– CNT ZERO EVENT – совпадение значения счетчика CNT с нулем;

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

– ETR RE EVENT – фиксация переднего фронта на входе ETR;

– ETR FE EVENT – фиксация заднего фронта на входе ETR;

– BRK EVENT – фиксация высокого уровня на входе BRK;

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

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

– CCR CAP1 EVENT[3:0] – запись значения счетчика CNT в регистр CCRy1 по захвату настроенного фронта на входе канала CHyi, события формируются индивидуально для каждого канала.

### 21.7.1 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.

### 21.7.2 Прерывания

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

### 21.7.3 Запросы DMA

На основе отслеживаемых событий блок таймера формирует сигналы запросов DMA TMRx_DMA_REQ и TMRx_DMA_REQ1-TMRx_DMA_REQ4. Выбор событий, формирующих запрос DMA, осуществляется через регистры DMA_RE и DMA_RE1-DMA_RE4. Данные регистры имеют одинаковые поля, при этом каждый из регистров отвечает за конфигурацию индивидуального запроса от таймера к соответствующему каналу DMA:

– регистр DMA_RE конфигурирует формирование запроса TMRx_DMA_REQ;

– регистры DMA_RE1-DMA_RE4 конфигурируют формирование запросов TMRx_DMA_REQ1-TMRx_DMA_REQ4, соответственно.

Запросы TMRx_DMA_REQ1-TMRx_DMA_REQ4 не привязаны к определенным каналам таймера, поэтому в полях CCR CAP EVENT RE[3:0], CCR REF EVENT RE[3:0] и CCR CAP1 EVENT RE[3:0] регистров DMA_RE1-DMA_RE4 может быть выбрано событие от любого канала таймера. Например, событие настроенного фронта на входе CH3i третьего канала таймера (CCR CAP EVENT RE[3:0]=0x4) может вызвать запрос TMRx_DMA_REQ1 при конфигурации регистра DMA_RE1 = 0x80.

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).

## 21.8 Примеры

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

### 21.8.1 Обычный счетчик

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

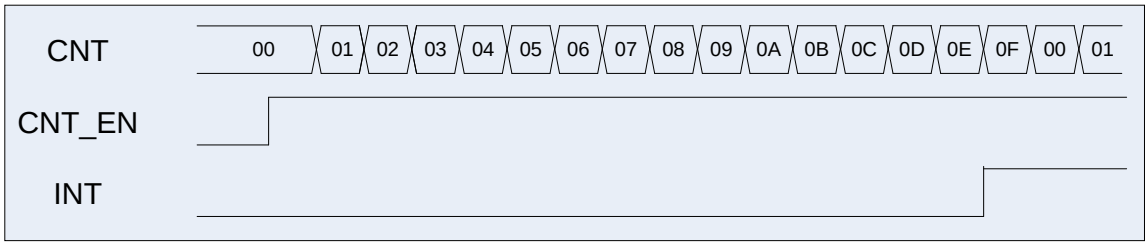



Рисунок 87 – Режим обычного счетчика

### 21.8.2 Режим захвата

```
RST_CLK->PER_CLOCK |= 0x00004000; //Разрешение частоты PCLK для таймера 1
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
                                 //для таймера 1 (TIM_CLK = HCLK)
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_CNTRL = 0x00008000; //Захват по положительному фронту сигнала
                                //на входе CH1i, фильтрация отключена
TIMER1->CH2_CNTRL = 0x00008010; //Захват по отрицательному фронту сигнала
                                //на входе CH2i, фильтрация отключена
TIMER1->CH3_CNTRL = 0x00008001; //Захват по положительному фронту сигнала
                                //на входе CH3i, фильтрация выполняется
                                //по двум выборкам на частоте TIM_CLK
TIMER1->CH4_CNTRL = 0x00008011; //Захват по отрицательному фронту сигнала
                                //на входе CH4i, фильтрация выполняется
                                //по двум выборкам на частоте TIM_CLK
//Режим работы выхода канала – канал работает на вход
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
```

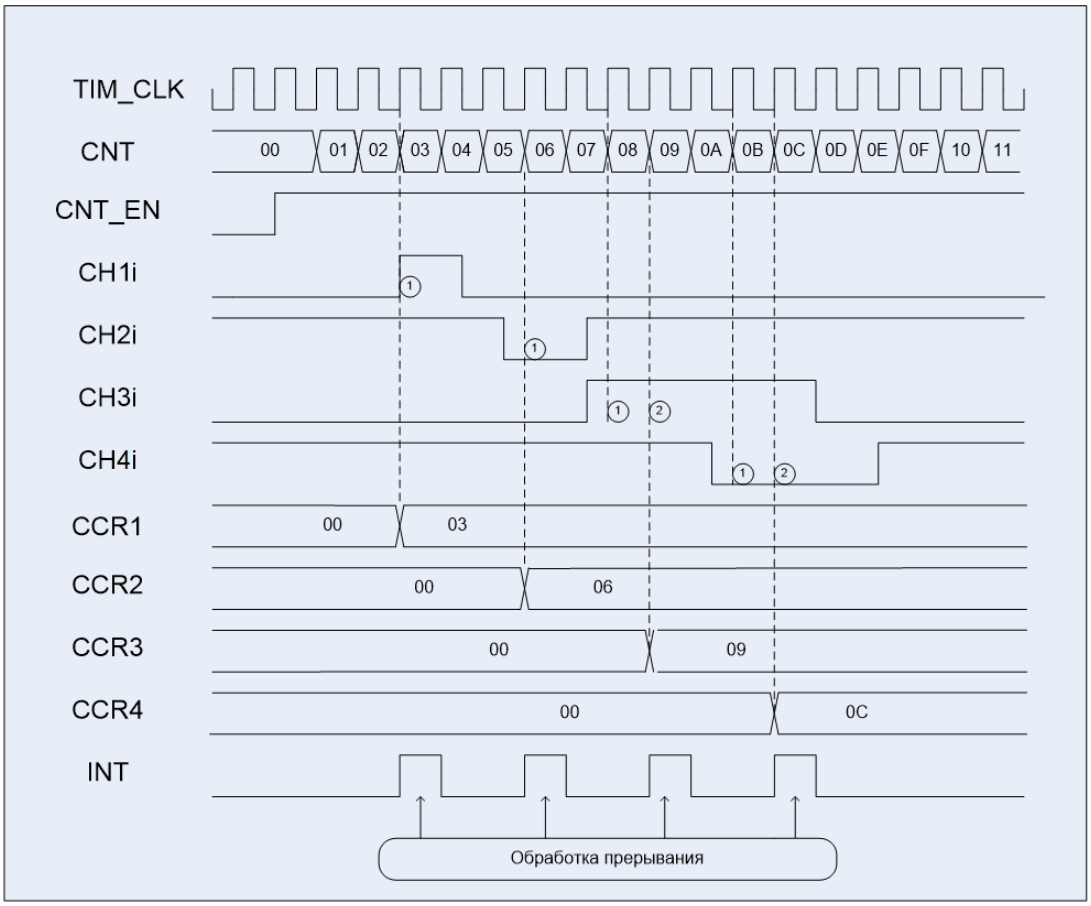



Рисунок 88 – Диаграммы примера работы в режиме захвата

### 21.8.3 Режим ШИМ

```
RST_CLK->PER_CLOCK |= 0x00004000; //Разрешение частоты PCLK для таймера 1
RST_CLK->TIM_CLOCK = 0x01000000; //Настройка и подача частоты TIM_CLK
                                 //для таймера 1 (TIM_CLK = HCLK)
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_CNTRL = 0x00000200;  //REF = 1, если CNT == CCR
TIMER1->CH2_CNTRL = 0x00000200;  //REF = 1, если CNT == CCR
TIMER1->CH3_CNTRL = 0x00000400;  //REF = 0, если CNT == CCR
TIMER1->CH4_CNTRL = 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->CCR1 = 0x00000003;
TIMER1->CCR2 = 0x00000006;
TIMER1->CCR3 = 0x00000009;
TIMER1->CCR4 = 0x0000000F;
//Разрешение работы таймера
TIMER1->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
```

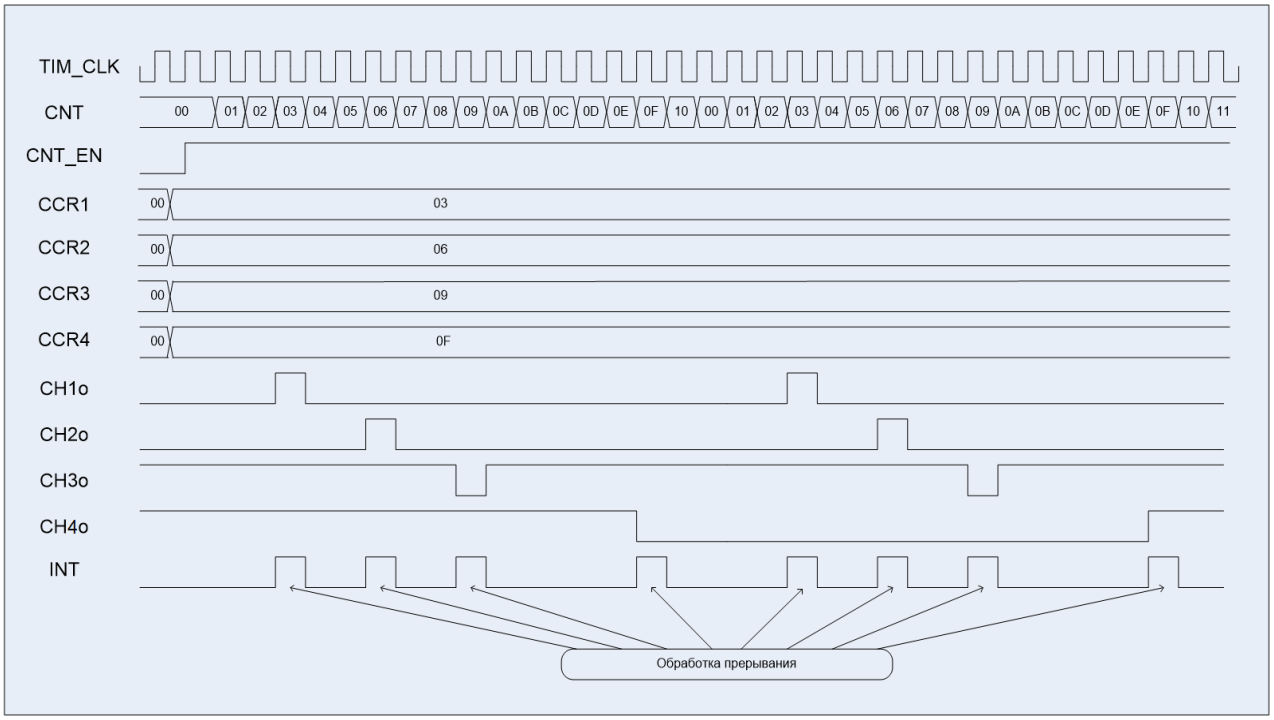



Рисунок 89 – Диаграммы примера работы в режиме ШИМ

## 21.9 Описание регистров блока таймера

Таблица 266– Базовые адреса и смещения регистров управления таймера

| Адрес | Название | Описание |
|---|---|---|
| 0x4007_0000 | TIMER1 | Контроллер TIMER 1 |
| 0x4007_8000 | TIMER2 | Контроллер TIMER 2 |
| 0x4008_0000 | TIMER3 | Контроллер TIMER 3 |
| 0x4009_8000 | TIMER4 | Контроллер TIMER 4 |
| 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[15:0] | Регистр управления основным счетчиком |
| 0x50 | BRKETR_CNTRL[15:0] | Регистр управления входом BRK и ETR |
| 0x54 | STATUS[31:0] | Регистр статуса таймера |
| 0x58 | IE[31:0] | Регистр разрешения прерываний таймера |
| 0x5C | DMA_RE[31:0] | Регистр разрешения формирования запроса DMA – TMRx_DMA_REQ |
| 0x80 | DMA_RE1[31:0] | Регистр разрешения формирования запроса DMA – TMRx_DMA_REQ1 |
| 0x84 | DMA_RE2[31:0] | Регистр разрешения формирования запроса DMA – TMRx_DMA_REQ2 |
| 0x88 | DMA_RE3[31:0] | Регистр разрешения формирования запроса DMA – TMRx_DMA_REQ3 |
| 0x8C | DMA_RE4[31:0] | Регистр разрешения формирования запроса DMA – TMRx_DMA_REQ4 |
| Канал 1 | | |
| 0x10 | CCR1[31:0] | Регистр сравнения/захвата для 1 канала таймера |
| 0x20 | CH1_CNTRL[15:0] | Регистр управления для 1 канала таймера |
| 0x30 | CH1_CNTRL1[15:0] | Регистр управления 1 для 1 канала таймера |
| 0x40 | CH1_DTG[15:0] | Регистр управления DTG для 1 канала таймера |
| 0x60 | CH1_CNTRL2[15:0] | Регистр управления 2 для 1 канала таймера |
| 0x70 | CCR11[31:0] | Регистр сравнения/захвата 1 для 1 канала таймера |
| Канал 2 | | |
| 0x14 | CCR2[31:0] | Регистр сравнения/захвата для 2 канала таймера |
| 0x24 | CH2_CNTRL[15:0] | Регистр управления для 2 канала таймера |
| 0x34 | CH2_CNTRL1[15:0] | Регистр управления 1 для 2 канала таймера |
| 0x44 | CH2_DTG[15:0] | Регистр управления DTG для 2 канала таймера |
| 0x64 | CH2_CNTRL2[15:0] | Регистр управления 2 для 2 канала таймера |
| 0x74 | CCR21[31:0] | Регистр сравнения/захвата 1 для 2 канала таймера |
| Канал 3 | | |
| 0x18 | CCR3[31:0] | Регистр сравнения/захвата для 3 канала таймера |
| 0x28 | CH3_CNTRL[15:0] | Регистр управления для 3 канала таймера |
| 0x38 | CH3_CNTRL1[15:0] | Регистр управления 1 для 3 канала таймера |

| Адрес | Название | Описание |
|---|---|---|
| 0x48 | CH3_DTG[15:0] | Регистр управления DTG для 3 канала таймера |
| 0x68 | CH3_CNTRL2[15:0] | Регистр управления 2 для 3 канала таймера |
| 0x78 | CCR31[31:0] | Регистр сравнения/захвата 1 для 3 канала таймера |
| Канал 4 | | |
| 0x1C | CCR4[31:0] | Регистр сравнения/захвата для 4 канала таймера |
| 0x2C | CH4_CNTRL[15:0] | Регистр управления для 4 канала таймера |
| 0x3C | CH4_CNTRL1[15:0] | Регистр управления 1 для 4 канала таймера |
| 0x4C | CH4_DTG[15:0] | Регистр управления DTG для 4 канала таймера |
| 0x6C | CH4_CNTRL2[15:0] | Регистр управления 2 для 4 канала таймера |
| 0x7C | CCR41[31:0] | Основной счетчик таймера |

### 21.9.1 CNT

Таблица 267 – Основной счетчик таймера CNT

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

Таблица 268 – 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 |

### 21.9.2 PSG

Таблица 269 – Делитель частоты при счете основного счетчика PSG

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

Таблица 270 – 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.<br>Основной счетчик считает на частоте<br>TIM_CLKd = TIM_CLK/(PSG+1) |

### 21.9.3 ARR

Таблица 271 – Основание счета основного счетчика ARR

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

Таблица 272 – 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] |

### 21.9.4 CNTRL

Таблица 273 – Регистр управления основного счетчика CNTRL

| Номер | 31…12 | 11…8 | 7…6 | 5…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0000 | 00 | 00 | 0 | 0 | 0 | 0 |
| | - | EVENT_ SEL[3:0] | CNT_ MODE[1:0] | FDTS[1:0] | DIR | WR_ CMPL | ARRB_ EN | CNT_ EN |

Таблица 274 – Описание бит регистра CNTRL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 31…12 | - | Зарезервировано |
| 11...8 | EVENT_SEL[3:0] | Биты выбора источника событий:<br>0000 – внутренняя тактовая частота TIM_CLKd (формируется путем деления частоты TIM_CLK);<br>0001 – CNT == ARR в таймере 1;<br>0010 – CNT == ARR в таймере 2;<br>0011 – CNT == ARR в таймере 3;<br>0100 – событие переднего фронта на канале 1, «Режим 1»;<br>0101 – событие переднего фронта на канале 2, «Режим 1»;<br>0110 – событие переднего фронта на канале 3, «Режим 1»;<br>0111 – событие переднего фронта на канале 4, «Режим 1»;<br>1000 – событие переднего фронта на ETR, «Режим 2»;<br>1001 – событие заднего фронта на ETR, «Режим 2»;<br>1010 – CNT == ARR в таймере 4;<br>1011-1111 – зарезервировано |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 7, 6 | CNT_MODE[1:0] | Режим счета основного счетчика:<br>00 – счетчик прямой при DIR = 0;<br>счетчик обратный при DIR = 1;<br>01 – счетчик двунаправленный с автоматическим изменением DIR при CNT == 0 или CNT == ARR;<br>10 – счетчик прямой при DIR = 0;<br>счетчик обратный при DIR = 1;<br>11 – зарезервировано.<br>Режим счета CNT_MODE[1:0] необходимо устанавливать в соответствии со значением в поле EVENT_SEL[3:0]:<br>– EVENT_SEL[3:0] = 0000: CNT_MODE[1:0] = 00 или 01;<br>– EVENT_SEL[3:0] != 0000: CNT_MODE[1:0] = 10 |
| 5, 4 | FDTS[1:0] | Делитель тактовой частоты $F_{DTS}$:<br>00 – $F_{DTS}$ = TIM_CLK;<br>01 – $F_{DTS}$ = TIM_CLK/2;<br>10 – $F_{DTS}$ = TIM_CLK/3;<br>11 – $F_{DTS}$ = TIM_CLK/4 |
| 3 | DIR | Направление счета основного счетчика:<br>0 – прямой, от 0 до ARR;<br>1 – обратный, от ARR до 0 |
| 2 | WR_CMPL | Флаг выполнения записи нового значения в регистры CNT, PSG и ARR:<br>0 – новые данные можно записывать;<br>1 – данные не записаны и идет запись |
| 1 | ARRB_EN | Режим обновления регистра ARR:<br>0 – ARR будет перезаписан в момент записи в ARR;<br>1 – ARR будет перезаписан при CNT == ARR |
| 0 | CNT_EN | Разрешение работы таймера:<br>0 – таймер отключен;<br>1 – таймер включен |

### 21.9.5 CCRy

Таблица 275 – Регистр сравнения/захвата для ‘y’ канала таймера CCRy

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

Таблица 276 – Описание бит регистра CCRy

| 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.<br>Режим ШИМ: значение CCR, с которым сравнивается CNT |

### 21.9.6 CCRy1

Таблица 277 – Регистр сравнения/захвата для ‘y’ канала таймера CCRy1

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

Таблица 278 – Описание бит регистра CCRy1

| 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.<br>Режим ШИМ: значение CCR1, с которым сравнивается CNT |

### 21.9.7 CHy_CNTRL

Таблица 279 – Регистр управления для ‘y’ канала таймера CHy_CNTRL

| Номер | 31…17 | 16 | 15 | 14 | 13 | 12 |
|---|---|---|---|---|---|---|
| Доступ | U | RO | R/W | RO | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | WR_CMPL1 | CAP_NPWM | WR_CMPL | ETREN | BRKEN |

| Номер | 11…9 | 8 | 7, 6 | 5, 4 | 3…0 |
|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W |
| Сброс | 000 | 0 | 00 | 00 | 0000 |
| | OCCM[2:0] | OCCE | CHPSC[1:0] | CHSEL[1:0] | CHFLTR[3:0] |

Таблица 280 – Описание бит регистра CHy_CNTRL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...17 | - | Зарезервировано |
| 16 | WR_CMPL1 | Флаг выполнения записи нового значения в регистр CCRy1:<br>0 – новые данные можно записывать;<br>1 – данные не записаны и идет запись |
| 15 | CAP_NPWM | Режим работы канала:<br>0 – канал работает в режиме ШИМ;<br>1 – канал работает в режиме захвата |
| 14 | WR_CMPL | Флаг выполнения записи нового значения в регистр CCRy:<br>0 – новые данные можно записывать;<br>1 – данные не записаны и идет запись |
| 13 | ETREN | Разрешение сброса сигнала REF в «0» при высоком уровне на входе ETR:<br>0 – запрещен;<br>1 – разрешен |
| 12 | BRKEN | Разрешение сброса сигналов REF и DTG в «0» при низком уровне на входе BRK:<br>0 – запрещен;<br>1 – разрешен |
| 11…9 | OCCM[2:0] | Формат выработки сигнала REF в режиме ШИМ:<br>Если CCR1_EN=0:<br>000 – всегда 0;<br>001 – 1, если CNT==CCR;<br>010 – 0, если CNT==CCR;<br>011 – переключение REF, если CNT==CCR;<br>100 – всегда 0;<br>101 – всегда 1;<br>110 – 1, если DIR=0 (счет прямой), CNT<CCR, иначе 0;<br>0, если DIR=1 (счет обратный), CNT>CCR, иначе 1;<br>111 – 0, если DIR=0 (счет прямой), CNT<CCR, иначе 1;<br>1, если DIR=1 (счет обратный), CNT>CCR, иначе 0.<br>Если CCR1_EN=1:<br>000 – всегда 0;<br>001 – 1, если CNT==CCR или CNT==CCR1;<br>010 – 0, если CNT==CCR или CNT==CCR1;<br>011 – переключение REF, если CNT==CCR или CNT==CCR1;<br>100 – всегда 0;<br>101 – всегда 1;<br>110 – 0, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 1;<br>0, если DIR=1 (счет обратный), CCR<CNT<CCR1, иначе 1;<br>111 – 1, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 0;<br>1, если DIR=1 (счет обратный), CCR<CN <CCR1, иначе 0;<br>Необходимо соблюдать условие CCR<CCR1 |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 8 | OCCE | Разрешение работы ETR:<br>0 – запрещен;<br>1 – разрешен |
| 7, 6 | CHPSC[1:0] | Предварительный делитель входного канала:<br>00 – нет деления;<br>01 – /2;<br>10 – /4;<br>11 – /8 |
| 5, 4 | CHSEL[1:0] | Выбор события по входному каналу CHyi для фиксации значения основного счетчика (регистр CNT) в регистр CCRy:<br>00 – положительный фронт на входном канале CHyi;<br>01 – отрицательный фронт на входном канале CHyi;<br>10 – положительный фронт от других каналов:<br>для первого канала от второго канала;<br>для второго канала от третьего канала;<br>для третьего канала от четвертого канала;<br>для четвертого канала от первого канала;<br>11 – положительный фронт от других каналов:<br>для первого канала от третьего канала;<br>для второго канала от четвертого канала;<br>для третьего канала от первого канала;<br>для четвертого канала от второго канала |
| 3…0 | CHFLTR[3:0] | Конфигурация фильтра на входе канала ‘y’. Выбор частоты выборки Fs и количества выборок N:<br>0000 – нет фильтрации, $F_S = F_{DTS}$;<br>0001 – $F_S$ = TIM_CLK, N = 2;<br>0010 – $F_S$ = TIM_CLK, N = 4;<br>0011 – $F_S$ = TIM_CLK, N = 8;<br>0100 – $F_S = F_{DTS}/2$, N = 6;<br>0101 – $F_S = F_{DTS}/2$, N = 8;<br>0110 – $F_S = F_{DTS}/4$, N = 6;<br>0111 – $F_S = F_{DTS}/4$, N = 8;<br>1000 – $F_S = F_{DTS}/8$, N = 6;<br>1001 – $F_S = F_{DTS}/8$, N = 8;<br>1010 – $F_S = F_{DTS}/16$, N = 5;<br>1011 – $F_S = F_{DTS}/16$, N = 6;<br>1100 – $F_S = F_{DTS}/16$, N = 8;<br>1101 – $F_S = F_{DTS}/32$, N = 5;<br>1110 – $F_S = F_{DTS}/32$, N = 6;<br>1111 – $F_S = F_{DTS}/32$, N = 8 |

### 21.9.8 CHy_CNTRL1

Таблица 281 – 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 282 – 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:<br>0 – выход не инвертируется;<br>1 – выход инвертируется |
| 11, 10 | NSELO[1:0] | Выбор источника сигнала для инверсного выхода nCHy:<br>00 – на nCHyo выдается 0;<br>01 – на nCHyo выдается 1;<br>10 – на nCHyo выдается сигнал nREF;<br>11 – на nCHyo выдается сигнал с DTG |
| 9, 8 | NSELOE[1:0] | Режим работы инверсного выхода nCHy:<br>00 – на nCHyoe выдается 0;<br>01 – на nCHyoe выдается 1;<br>10 – на nCHyoe выдается сигнал nREF;<br>11 – на nCHyoe выдается сигнал с DTG.<br>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:<br>0 – выход не инвертируется;<br>1 – выход инвертируется |
| 3, 2 | SELO[1:0] | Выбор источника сигнала для прямого выхода CHy:<br>00 – на CHyo выдается 0;<br>01 – на CHyo выдается 1;<br>10 – на CHyo выдается сигнал REF;<br>11 – на CHyo выдается сигнал с DTG |
| 1, 0 | SELOE[1:0] | Режим работы прямого выхода CHy:<br>00 – на CHyoe выдается 0;<br>01 – на CHyoe выдается 1;<br>10 – на CHyoe выдается сигнал REF;<br>11 – на CHyoe выдается сигнал с DTG.<br>При CHyoe = 0 вывод канала работает в режиме входа, при CHyoe = 1 вывод канала работает в режиме выхода |

### 21.9.9 CHy_CNTRL2

Таблица 283 – 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 |
| | - | EV_DELAY | CCRRLD | CCR1_EN | CHSEL[1:0] |

Таблица 284 – Описание бит регистра CHy_CNTRL2

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...5 | - | Зарезервировано |
| 4 | EV_DELAY | Задержка события захвата до обновления регистров CCRy и CCRy1:<br>0 – сигнал события захвата устанавливается в момент обнаружения события, при этом обновление регистров CCRy и CCRy1 выполняется через один такт TIM_CLK;<br>1 – сигнал события захвата устанавливается синхронно с обновлением информации в регистрах CCRy и CCRy1 |
| 3 | CCRRLD | Режим обновления регистров CCRy и CCRy1:<br>0 – обновление возможно в любой момент времени;<br>1 – обновление будет осуществлено только при CNT == 0 |
| 2 | CCR1_EN | Разрешение работы регистра CCRy1:<br>0 – CCRy1 не используется;<br>1 – CCRy1 используется |
| 1, 0 | CHSEL[1:0] | Выбор события по входному каналу CHyi для фиксации значения основного счетчика (регистр CNT) в регистр CCRy1:<br>00 – положительный фронт на входном канале CHyi;<br>01 – отрицательный фронт на входном канале CHyi;<br>10 – отрицательный фронт от других каналов:<br>для первого канала от второго канала;<br>для второго канала от третьего канала;<br>для третьего канала от четвертого канала;<br>для четвертого канала от первого канала;<br>11 – отрицательный фронт от других каналов:<br>для первого канала от третьего канала;<br>для второго канала от четвертого канала;<br>для третьего канала от первого канала;<br>для четвертого канала от второго канала |

### 21.9.10 CHy_DTG

Таблица 285 – Регистр CHy_DTG управления 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 286 – 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.<br>Задержка DTGdel = DTG • (DTGx + 1). |
| 7…5 | - | Зарезервировано |
| 4 | EDTS | Частота работы DTG<br>0 – TIM_CLK;<br>1 – $F_{DTS}$ |
| 3…0 | DTGx[3:0] | Предварительный делитель частоты DTG |

### 21.9.11 BRKETR_CNTRL

Таблица 287 – 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 | | 0000 | 00 | 0 | 0 |
| | - | ETR_FILTER[3:0] | ETR_PSC[1:0] | ETR_INV | BRK_INV |

Таблица 288 – Описание бит регистра BRKETR_CNTRL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...8 | - | Зарезервировано |
| 7…4 | ETR_FILTER[3:0] | Конфигурация фильтра на входе ETR. Выбор частоты выборки Fs и количества выборок N:<br>0000 – нет фильтрации, $F_S = F_{DTS}$;<br>0001 – $F_S$ = TIM_CLK, N = 2;<br>0010 – $F_S$ = TIM_CLK, N = 4;<br>0011 – $F_S$ = TIM_CLK, N = 8;<br>0100 – $F_S = F_{DTS}/2$, N = 6;<br>0101 – $F_S = F_{DTS}/2$, N = 8;<br>0110 – $F_S = F_{DTS}/4$, N = 6;<br>0111 – $F_S = F_{DTS}/4$, N = 8;<br>1000 – $F_S = F_{DTS}/8$, N = 6;<br>1001 – $F_S = F_{DTS}/8$, N = 8; |

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

### 21.9.12 STATUS

Таблица 289 – Регистр статуса таймера STATUS

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

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

Таблица 290 – Описание бит регистра STATUS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…17 | - | Зарезервировано |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 16…13 | CCR CAP1 EVENT[3:0] | Событие записи значения счетчика CNT в регистр CCRy1 по захвату настроенного фронта на входе канала CHyi:<br>0 – нет события;<br>1 – есть событие.<br>Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события.<br>Бит 0 – первый канал;<br>Бит 3 – четвертый канал |
| 12…9 | CCR REF EVENT[3:0] | Событие переднего фронта на выходе генератора опорного сигнала REF:<br>0 – нет события;<br>1 – есть событие.<br>Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события.<br>Бит 0 – первый канал;<br>Бит 3 – четвертый канал |
| 8…5 | CCR CAP EVENT[3:0] | Событие записи значения счетчика CNT в регистр CCRy по захвату настроенного фронта на входе канала CHyi:<br>0 – нет события;<br>1 – есть событие.<br>Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события.<br>Бит 0 – первый канал;<br>Бит 3 – четвертый канал |
| 4 | BRK EVENT | Событие высокого уровня на входе BRK:<br>0 – нет события;<br>1 – есть событие.<br>Сбрасывается записью «0», при условии наличия низкого уровня на входе BRK |
| 3 | ETR FE EVENT | Событие заднего фронта на входе ETR:<br>0 – нет события;<br>1 – есть событие.<br>Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события |
| 2 | ETR RE EVENT | Событие переднего фронта на входе ETR:<br>0 – нет события;<br>1 – есть событие.<br>Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события |

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

### 21.9.13 IE

Таблица 291 – Регистр разрешения прерывания таймера IE

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

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

Таблица 292 – Описание бит регистра IE

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…17 | - | Зарезервировано |
| 16...13 | CCR CAP1 EVENT IE[3:0] | Флаг разрешения прерывания по событию CCR CAP1 EVENT[3:0] в регистре STATUS:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено.<br>Бит 0 – первый канал;<br>Бит 3 – четвертый канал |
| 12…9 | CCR REF EVENT IE[3:0] | Флаг разрешения прерывания по событию CCR REF EVENT[3:0] в регистре STATUS:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено.<br>Бит 0 – первый канал;<br>Бит 3 – четвертый канал |
| 8…5 | CCR CAP EVENT IE [3:0] | Флаг разрешения прерывания по событию CCR CAP EVENT[3:0] в регистре STATUS:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено.<br>Бит 0 – первый канал;<br>Бит 3 – четвертый канал |
| 4 | BRK EVENT IE | Флаг разрешения прерывания по событию BRK EVENT в регистре STATUS:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено |
| 3 | ETR FE EVENT IE | Флаг разрешения прерывания по событию ETR FE EVENT в регистре STATUS:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено |
| 2 | ETR RE EVENT IE | Флаг разрешения прерывания по событию ETR RE EVENT в регистре STATUS:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено |
| 1 | CNT ARR EVENT IE | Флаг разрешения прерывания по событию CNT ARR EVENT в регистре STATUS:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено |
| 0 | CNT ZERO EVENT IE | Флаг разрешения прерывания по событию CNT ZERO EVENT в регистре STATUS:<br>0 – прерывание запрещено;<br>1 – прерывание разрешено |

### 21.9.14 DMA_RE, DMA_RE1-DMA_RE4

Таблица 293 – Регистры DMA_RE, DMA_RE1-DMA_RE4 разрешения запросов DMA

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

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

Таблица 294 – Описание бит регистров DMA_RE, DMA_RE1-DMA_RE4

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

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 2 | ETR RE EVENT RE | Флаг разрешения запроса DMA по событию ETR RE EVENT:<br>0 – запрос DMA запрещен;<br>1 – запрос DMA разрешен |
| 1 | CNT ARR EVENT RE | Флаг разрешения запроса DMA по событию CNT ARR EVENT:<br>0 – запрос DMA запрещен;<br>1 – запрос DMA разрешен |
| 0 | CNT ZERO EVENT RE | Флаг разрешения запроса DMA по событию CNT ZERO EVENT:<br>0 – запрос DMA запрещен;<br>1 – запрос DMA разрешен |

# 22 Контроллер АЦП

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

Контроллер АЦП позволяет:

– оцифровать один из восьми внешних каналов;

– оцифровать значение встроенного датчика температуры;

– оцифровать значение встроенного источника опорного напряжения;

– осуществить автоматический опрос заданных каналов;

– выработать сигнал прерывания при выходе оцифрованного значения за заданные пределы.

Для осуществления преобразования требуется 28 тактов синхронизации C_ADC. В качестве синхросигнала может выступать частота процессора PCLKd либо 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_ADC не может превышать 14 МГц.

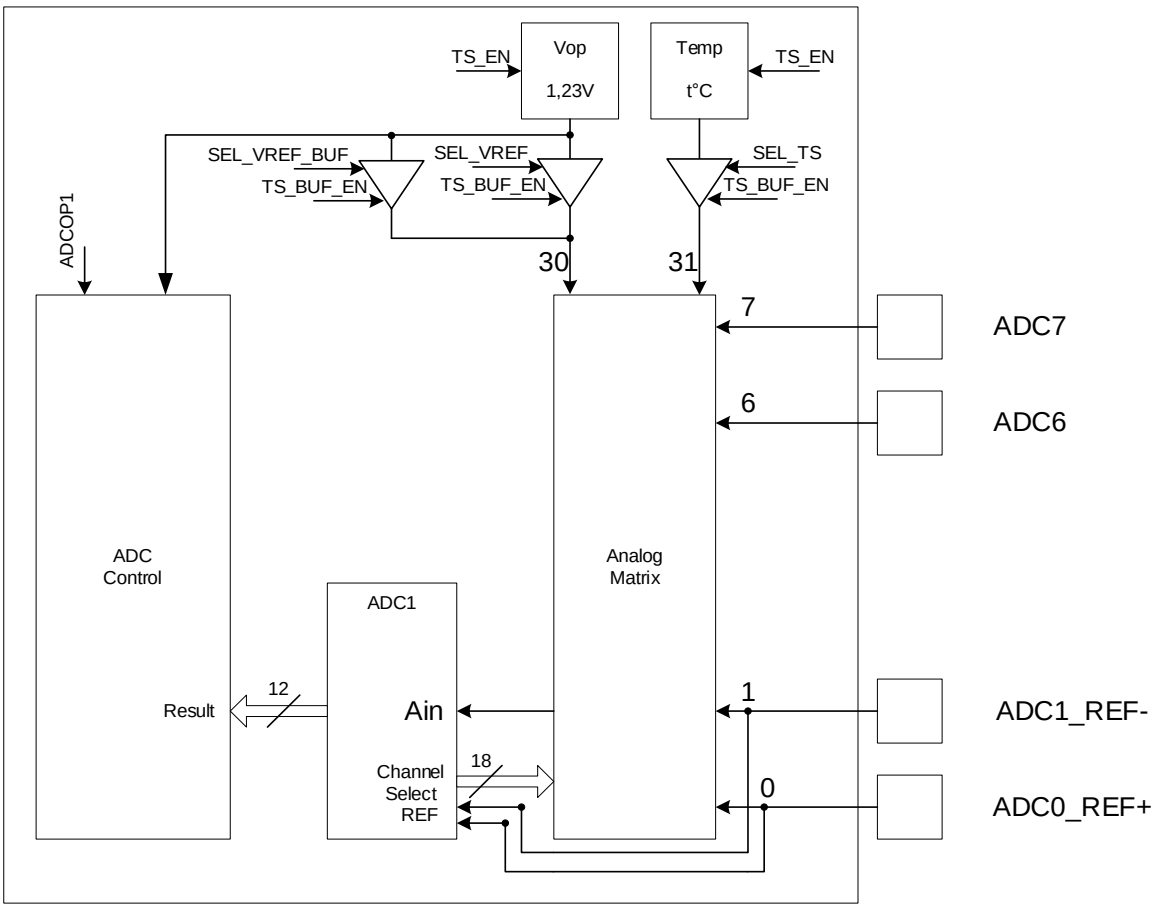



Рисунок 90 – Структурная схема контроллера АЦП

Для включения АЦП необходимо установить бит Cfg_REG_ADON. Для снижения тока потребления вместо собственного источника опорного напряжения в АЦП может использоваться источник датчика температуры. Для этого необходимо включить блок датчика температуры и источник опорного напряжения, установив бит TS_EN в «1». После включения можно использовать источник опорного напряжения для АЦП вместо его собственного. Для этого необходимо установить биты ADC1_OP в «1». Для преобразования необходимо, чтобы выводы, используемые АЦП в порте D, были сконфигурированы как аналоговые, и были отключены какие-либо внутренние подтяжки.

## 22.1 Преобразование внешнего канала

В регистре ADC1_CFG в битах Cfg_REG_CHS[4:0] необходимо задать соответствующий выводу номер канала. Диапазон измеряемых напряжений определяется состоянием 11-го бита CFG M_REF, где выбирается либо внутреннее опорное напряжение ($AU_{CC}$-AGND) при Cfg M_REF = 0, либо внешнее, подаваемое на ADC0_REF+ / ADC1_REF- (т.е. точное внешнее) при Cfg M_REF = 1. Биты Cfg_REG_CHCH, Cfg_REG_RNGC, Cfg_REG_SAMPLE, TS_BUF_EN, SEL_VREF, SEL_VREF_BUF, SEL_TS и Cfg_Sync_Conver должны быть сброшены.

Для начала преобразования необходимо записать «1» в бит Cfg_REG_GO.

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

После считывания результата бит Flg_REG_EOCIF сбросится.

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

## 22.2 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 битов необходимо обеспечить до установки конфигурации АЦП, то есть до записи в регистр ADCx_CFG. Преобразование может осуществляться при внутреннем опорном напряжении (бит Cfg_M_REF = 0) или при внешнем (Cfg_M_REF = 1), в этом случае опорное напряжение берется с выводов ADC0_REF+ и ADC1_REF-. Биты Cfg_REG_RNGC, TS_BUF_EN, SEL_VREF, SEL_VREF_BUF, SEL_TS и Cfg_Sync_Conver должны быть сброшены, а Cfg_REG_SAMPLE и Cfg_REG_CHCH должны быть установлены. С помощью битов Delay_GO можно задать паузу между преобразованиями при переборе каналов. Для начала преобразования необходимо записать «1» в бит Cfg_REG_GO.

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

После считывания результата бит Flg_REG_EOCIF сбросится.

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

Для последовательного преобразования одного и того же канала можно в регистре ADC1_CHSEL выбрать только один канал и установить бит Cfg_REG_CHCH в «1», либо установить номер канала в битах Cfg_REG_CHS[4:0] и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. Последовательное преобразование значения датчика температуры и источника опорного напряжения могут выполняться только в режиме последовательного преобразования одного канала.

## 22.3 Преобразование с контролем границ

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

## 22.4 Датчик опорного напряжения

С помощью АЦП можно осуществить оцифровку напряжения с источника опорного напряжения Vop 1,23 В. Для этого необходимо включить блок датчика температуры и источника опорного напряжения, установив бит TS_EN в «1». Для выбора источника опорного напряжения в качестве источника для преобразования необходимо в битах Cfg_REG_CHS установить значение 30 канала, установить бит SEL_VREF_BUF или SEL_VREF, а также бит TS_BUF_EN, 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.

Примечание – При настройке преобразования сигналов с датчика температуры или источника опорного напряжения допустимо устанавливать в логическую «1» только один из битов: SEL_VREF, SEL_TS или SEL_VREF_BUF (запрещено одновременно устанавливать в «1» более одного из этих разрядов). Если необходимо в последствии проводить преобразование сигналов со внешних каналов АЦП, то предварительно необходимо сбросить в «0» биты SEL_VREF, SEL_TS и SEL_VREF_BUF регистров ADC1_CFG, ADC1_TRIM.

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

## 22.5 Датчик температуры

С помощью первого АЦП можно осуществить оцифровку напряжения с датчика температуры. Для этого необходимо включить блок датчика температуры и источник опорного напряжения, установив бит TS_EN в «1». После включения можно использовать источник опорного напряжения для работы АЦП вместо его собственного, что позволяет снизить ток потребления. Для этого необходимо установить биты ADC1_OP в «1». Для выбора датчика температуры в качестве источника для преобразования необходимо в битах Cfg_REG_CHS установить значение 31 канала. Установить биты TS_BUF_EN и SEL_TS, после чего можно запустить процесс преобразования. Для начала преобразования необходимо записать «1» в бит Cfg_REG_GO.

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

После считывания результата бит Flg_REG_EOCIF сбросится.

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

Для последовательного преобразования только датчика температуры можно в регистре ADC1_CHSEL выбрать только 31 канал и установить бит Cfg_REG_CHCH в «1», либо установить номер 31-го канала в битах Cfg_REG_CHS[4:0], и сбросить бит Cfg_REG_CHCH в 0. В этом случае процесс последовательного преобразования будет выполняться только для данного канала. При этом должны быть также установлены биты TS_BUF_EN и SEL_TS.

Примечание – При настройке преобразования сигналов с датчика температуры или источника опорного напряжения допустимо устанавливать в логическую «1» только один из битов: SEL_VREF, SEL_TS или SEL_VREF_BUF (запрещено одновременно устанавливать в «1» более одного из этих разрядов). Если необходимо в последствии проводить преобразование сигналов со внешних каналов АЦП, то предварительно необходимо сбросить в «0» биты SEL_VREF, SEL_TS и SEL_VREF_BUF регистров ADC1_CFG, ADC1_TRIM.

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

## 22.6 Время заряда внутренней емкости

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

преобразования внешнего сигнала в цифровой вид, за время первого этапа внутренняя емкость должна зарядиться до уровня внешнего сигнала. Это время определяется соотношением номинальной внутренней емкости, входным сопротивлением тракта АЦП и выходным сопротивлением источника сигнала. Максимальное выходное сопротивление источника $R_{AIN}$ для обеспечения качественного преобразования определяется по формуле

$$R_{AIN} < \frac{T_{track}}{C_{ADC} \cdot ln(2^N)} - R_{ADC}\,, \tag{7}$$

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

N – требуемая точность в разрядах;

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

Ttrack – время заряда внутренней емкости, с. Определяется по формуле

$$T_{track} = 4 \cdot T_{C_ADC} + N_{PCLKd} \cdot T_{PCLKd} = \frac{4}{f_{C_ADC}} + \frac{(DelayGo + 1)}{f_{PCLKd}}, \tag{8}$$

где $f_{C_ADC}$ – рабочая частота АЦП, $c^{-1}$ (определяется Cfg REG CLKS в регистре ADC1_CFG);

$f_{PCLKd}$ – определяется по формуле

$$f_{PCLKd} = \frac{f_{PCLK}}{2^{Cfg\ REG\ DIVCLK}}. \tag{9}$$

Время заряда внутренней емкости можно изменять с помощью битов DelayGo[2:0].

Если необходимо обеспечить преобразование с точностью 12 разрядов ± 1/4 LSB, то N = 14. Если необходимо обеспечить преобразование с точностью 10 разрядов ± 1 LSB, то N = 10.

Таблица 295 – Время заряда внутренней емкости АЦП и время преобразования

| DelayGo[2:0] | Дополнительная задержка перед началом преобразования | Общее время $T_{track}$ заряда емкости АЦП перед началом преобразования | Общее время преобразования АЦП |
|---|---|---|---|
| 000 | 1 • PCLKd | 4 • C_ADC + 1 • PCLKd | 28 • C_ADC + 1 • PCLKd |
| 001 | 2 • PCLKd | 4 • C_ADC + 2 • PCLKd | 28 • C_ADC + 2 • PCLKd |
| 010 | 3 • PCLKd | 4 • C_ADC + 3 • PCLKd | 28 • C_ADC + 3 • PCLKd |
| 011 | 4 • PCLKd | 4 • C_ADC + 4 • PCLKd | 28 • C_ADC + 4 • PCLKd |
| 100 | 5 • PCLKd | 4 • C_ADC + 5 • PCLKd | 28 • C_ADC + 5 • PCLKd |
| 101 | 6 • PCLKd | 4 • C_ADC + 6 • PCLKd | 28 • C_ADC + 6 • PCLKd |
| 110 | 7 • PCLKd | 4 • C_ADC + 7 • PCLKd | 28 • C_ADC + 7 • PCLKd |
| 111 | 8 • PCLKd | 4 • C_ADC + 8 • PCLKd | 28 • C_ADC + 8 • PCLKd |

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

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

## 22.7 Описание регистров блока контроллера АЦП

Таблица 296 – Описание регистров блока контроллера АЦП

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x4008_8000 | ADC | Контроллер ADC |
| Смещение | | |
| 0x00 | ADC1_CFG | Регистр управления ADC |
| 0x04 | ADC2_CFG | Регистр управления ADC |
| 0x08 | ADC1_H_LEVEL | Регистр верхней границы ADC |
| 0x10 | ADC1_L_LEVEL | Регистр нижней границы ADC |
| 0x18 | ADC1_RESULT | Регистр результата ADC |
| 0x20 | ADC1_STATUS | Регистр статуса ADC |
| 0x28 | ADC1_CHSEL | Регистр выбора каналов перебора ADC |
| 0x30 | ADC1_TRIM | Регистр настройки термодатчика |

### 22.7.1 ADCx_CFG

Таблица 297 – Регистр ADCx_CFG

| Номер | 31…28 | 27…25 | 24…21 | 20 | 19 | 18 | 17 | 16 | 15…12 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | U | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | Delay Go [2:0] | - | SEL VREF | SEL TS | TS_BUF EN | TS_EN / ADC1 OP | Cfg Sync Conver | Cfg REG DIVCLK [3:0] |

| Номер | 11 | 10 | 9 | 8…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Cfg M_REF | Cfg REG RNGC | Cfg REG CHCH | Cfg REG CHS [4:0] | Cfg REG SAMPLE | Cfg REG CLKS | Cfg REG GO | Cfg REG ADON |

Таблица 298 – Описание бит регистра ADCx_CFG

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…28 | - | Зарезервировано |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 27…25 | Delay Go [2:0] | Задержка перед началом следующего преобразования после завершения предыдущего при последовательном переборе каналов:<br>000 – 0 тактов PCLKd;<br>001 – 1 такта PCLKd;<br>…<br>111 – 7 тактов PCLKd |
| 24…21 | - | Зарезервировано |
| 20 | SEL VREF | Выбор для оцифровки источника опорного напряжения на 1,23 В:<br>0 – не выбран;<br>1 – выбран.<br>Должен использоваться совместно с выбором канала Cfg_REG_CHS = 30 |
| 19 | SEL TS | Выбор для оцифровки датчика температуры:<br>0 – не выбран;<br>1 – выбран.<br>Должен использоваться совместно с выбором канала Cfg_REG_CHS = 31 |
| 18 | TS BUF EN | Включение выходного усилителя для датчика температуры:<br>0 – выключен;<br>1 – включен.<br>Используется при TS_EN = 1. Для уменьшения тока потребления |
| 17 | TS EN | В регистре ADC1_CFG.<br>Включение датчика температуры и источника опорного напряжения:<br>0 – выключен;<br>1 – включен.<br>При включении датчика температуры и источника опорного напряжения выходной сигнал стабилизируется в течение 1 мс |
| 17 | ADC1 OP | В регистре ADC2_CFG.<br>Выбор источника для формирования внутренней рабочей точки АЦП:<br>0 – внутренний (неточный);<br>1 – от датчика температуры (точный) |
| 16 | Cfg Sync Conver | Записывать всегда ноль |
| 15…12 | Cfg REG DIVCLK [3:0] | Выбор коэффициента деления частоты процессора:<br>0000 – PCLKd = PCLK;<br>0001 – PCLKd = PCLK /2;<br>0010 – PCLKd = PCLK /4;<br>0011 – PCLKd = PCLK/8;<br>…<br>1011 – PCLKd = PCLK/2048;<br>Остальные комбинации – PCLKd = PCLK |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 11 | Cfg<br>M_REF | Выбор источника опорных напряжений:<br>0 – внутренне опорное напряжение (от AUcc и AGND);<br>1 – внешнее опорное напряжение (от Uref+ и Uref-) |
| 10 | Cfg<br>REG<br>RNGC | Разрешение автоматического контролирования уровней:<br>1 – разрешена выработка сигнала прерывания при выходе напряжения за диапазон, указанный в регистрах границы зоны допуска;<br>0 – запрещено |
| 9 | Cfg<br>REG<br>CHCH | Выбор переключения каналов:<br>1 – переключение включено (перебираются каналы, выбранные в регистре выбора канала);<br>0 – используется только выбранный канал |
| 8…4 | Cfg<br>REG<br>CHS<br>[4:0] | Выбор аналогового канала, по которому поступает сигнал для преобразования:<br>00000 – 0 канал;<br>00001 – 1 канал;<br>…<br>11111 – 31 канал |
| 3 | Cfg<br>REG<br>SAMPLE | Выбор способа запуска АЦП:<br>1 – последовательное, автоматический запуск после завершения предыдущего преобразования;<br>0 – одиночное |
| 2 | Cfg<br>REG<br>CLKS | Выбор источника синхросигнала C_ADC работы ADC:<br>0 – PCLKd (определяется по формуле (9));<br>1 – ADC_CLK |
| 1 | Cfg<br>REG<br>GO | Начало преобразования.<br>Запись «1» начинает процесс преобразования, сбрасывается автоматически |
| 0 | Cfg<br>REG<br>ADON | Включение АЦП:<br>1 – включен;<br>0 – выключен |

### 22.7.2 ADC1_H_LEVEL

Таблица 299 – Регистр ADC1_H_LEVEL

| Номер | 31…12 | 11…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | REG H<br>LEVEL [11:0] |

Таблица 300 – Описание бит регистра ADC1_H_LEVEL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…12 | - | Зарезервировано |
| 11…0 | REG H LEVEL [11:0] | Верхняя граница зоны допуска |

### 22.7.3 ADC1_L_LEVEL

Таблица 301 – Регистр ADC1_L_LEVEL

| Номер | 31…12 | 11…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | REG L LEVEL [11:0] |

Таблица 302 – Описание бит регистра ADC1_L_LEVEL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…12 | - | Зарезервировано |
| 11…0 | REG L LEVEL [11:0] | Нижняя граница зоны допуска |

### 22.7.4 ADC1_RESULT

Таблица 303 – Регистр ADC1_RESULT

| Номер | 31…21 | 20…16 | 15…12 | 11…0 |
|---|---|---|---|---|
| Доступ | U | RO | U | RO |
| Сброс | 0 | 0 | 0 | 0 |
| | - | CHANNEL [4:0] | - | RESULT [11:0] |

Таблица 304 – Описание бит регистра ADC1_RESULT

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…21 | - | Зарезервировано |
| 20…16 | CHANNEL [4:0] | Канал результата преобразования |
| 15…12 | - | Зарезервировано |
| 11…0 | RESULT [11:0] | Значение результата преобразования |

### 22.7.5 ADC1_STATUS

Таблица 305 – Регистр ADC1_STATUS

| Номер | 31...5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | ECOIF IE | AWOIFIE | Flg REG EOCIF | Flg REG AWOIFEN | Flg REG OVERWRITE |

Таблица 306 – Описание бит регистра ADC1_STATUS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…5 | - | Зарезервировано |
| 4 | ECOIF_IE | Флаг разрешения генерирования прерывания по событию Flg_REG_ECOIF:<br>0 – прерывания не генерируется;<br>1 – прерывание генерируется |
| 3 | AWOIF_IE | Флаг разрешения генерирования прерывания по событию Flg_REG_AWOIFEN:<br>0 – прерывания не генерируется;<br>1 – прерывание генерируется |
| 2 | Flg REG EOCIF | Флаг выставляется, когда закончено преобразования, и данные еще не считаны. Очищается считыванием результата из регистра ADCx_RESULT:<br>1 – есть готовый результат преобразования;<br>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:<br>0 – результат в допустимой зоне;<br>1 – вне допустимой зоны |
| 0 | Flg REG OVERWRITE | Данные в регистре результата были перезаписаны, данный флаг сбрасывается только при записи в регистр флагов:<br>0 – не было события перезаписи несчитанного результата;<br>1 – был результат преобразования, который не был считан |

### 22.7.6 ADC1_CHSEL

Таблица 307 – Регистр ADC1_CHSEL

| Номер | 31…0 |
|---|---|
| Доступ | R/W |
| Сброс | 0 |
| | Sl_Ch_Ch_REF[31:0] |

Таблица 308 – Описание бит регистра ADC1_CHSEL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…0 | Sl_Ch_Ch_REF[31:0] | Выбор каналов автоматического перебора:<br>0 – в соответствующем бите канал не участвует в переборе;<br>1 – канал участвует в переборе |

### 22.7.7 ADC1_TRIM

Таблица 309 – Регистр ADC1_TRIM

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

Таблица 310 – Описание бит регистра ADC1_TRIM

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

# 23 Контроллер ЦАП

В микросхеме реализован ЦАП. Для включения ЦАП необходимо установить бит Cfg_ON_DACx в «1», используемые выводы ЦАП в портах D и E сконфигурировать как аналоговые, и отключить какие-либо внутренние подтяжки этих выводов. При работе ЦАП после записи данных в регистр данных DACx_DATA на выходе DACx_OUT формируется уровень напряжения, соответствующий записанному значению. ЦАП может работать от внутреннего (Cfg_M_REFx = 0) или внешнего (Cfg_M_REFx = 1) опорного напряжения. Если используется внутреннее опорное напряжение, то ЦАП формирует выходной сигнал в диапазоне от 0 до напряжения питания AUcc. В режиме работы с внешним опорным напряжением ЦАП формирует выходное напряжение в диапазоне от 0 до значения DACx_REF.

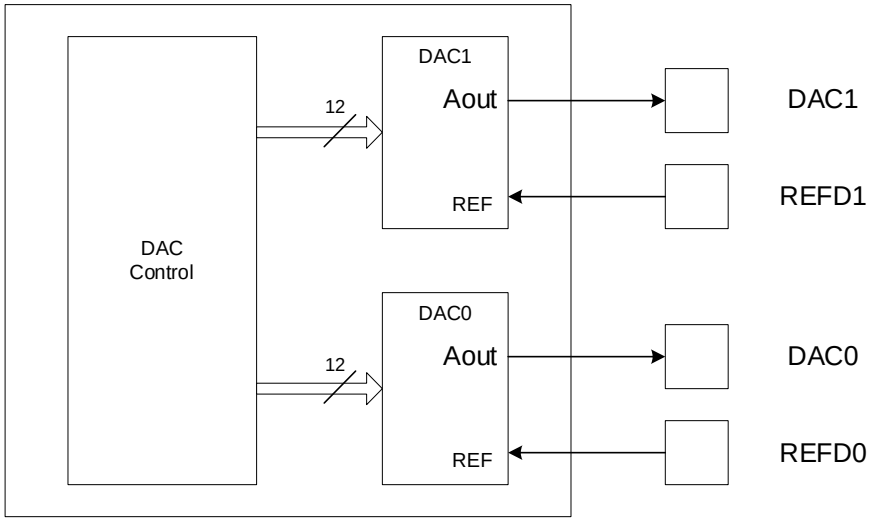



Рисунок 91 – Структурная схема контроллера ЦАП

## 23.1 Описание регистров блока контроллера ЦАП

Таблица 311 – Описание регистров блока контроллера ЦАП

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x4009_0000 | DAC | Контроллер DAC |
| Смещение | | |
| 0x00 | DAC_CFG | Регистр управления DAC |
| 0x04 | DAC0_DATA | Регистр данных DAC0 |
| 0x08 | DAC1_DATA | Регистр данных DAC1 |

### 23.1.1 DAC_CFG

Таблица 312 – Регистр CFG

| 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 |
| | | Cfg_ SYNC_A | Cfg_ON_ DAC1 | Cfg_ON_ DAC0 | Cfg_M_ REF1 | Cfg_M_ REF0 |

Таблица 313 – Описание бит регистра CFG

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…5 | - | Зарезервировано |
| 4 | Cfg_SYNC_A | Синхронизация DAC0 и DAC1:<br>0 – асинхронные;<br>1 – синхронные |
| 3 | Cfg_ON_DAC1 | Включение DAC1:<br>1 – включен;<br>0 – выключен |
| 2 | Cfg_ON_DAC0 | Включение DAC0:<br>1 – включен;<br>0 – выключен |
| 1 | Cfg_M_REF1 | Выбор источника опорного напряжения DAC1:<br>0 – в качестве опорного напряжения используется напряжение питания с вывода AUcc;<br>1 – в качестве опорного напряжения используется напряжение на входе DACx_REF |
| 0 | Cfg_M_REF0 | Выбор источника опорного напряжения DAC0, аналогично Cfg_M_REF1 |

### 23.1.2 DAC0_DATA

Таблица 314 – Регистр DAC0_DATA

| Номер | 31…28 | 27…16 | 15…12 | 11…0 |
|---|---|---|---|---|
| Доступ | U | R/W | U | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | - | DAC1_DATA[11:0] | - | DAC0_DATA[11:0] |

Таблица 315 – Описание бит регистра DAC0_DATA

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…28 | - | Зарезервировано |
| 27…16 | DAC1 DATA[11:0] | Данные DAC1 при Cfg_SYNC_A = 1.<br>При чтении всегда равны нулю. Читать из DAC1_DATA |
| 15…12 | - | Зарезервировано |
| 11…0 | DAC0 DATA[11:0] | Данные DAC0 |

### 23.1.3 DAC1_DATA

Таблица 316 – Регистр DAC1_DATA

| Номер | 31…28 | 27…16 | 15…12 | 11…0 |
|---|---|---|---|---|
| Доступ | U | R/W | U | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | - | DAC0_DATA[11:0] | - | DAC1_DATA[11:0] |

Таблица 317 – Описание бит регистра DAC1_DATA

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…28 | - | Зарезервировано |
| 27…16 | DAC0 DATA[11:0] | Данные DAC0 при Cfg_SYNC_A = 1.<br>При чтении всегда равны нулю. Читать из DAC0_DATA |
| 15…12 | - | Зарезервировано |
| 11…0 | DAC1 DATA[11:0] | Данные DAC1 |
| Примечание – Если бит конфигурации Cfg_SYNC_A регистра CFG установлен в «1», то данные для DAC0 и DAC1 задаются записью в один из регистров DACx_DATA | | |

# 24 Контроллер интерфейса по ГОСТ 18977-79

Контроллер содержит в своем составе восемь приемников и четыре передатчика по ГОСТ 18977-79 (далее ARINC). Каждый приемник поддерживает функцию распознавания меток (или адресов). Для каждого приемника может быть запрограммировано до 32 восьмиразрядных меток. Помимо этого, фильтрация входных данных может осуществляться не только на базе меток, но и на базе двух бит Источник/Приемник. Каждый передатчик поддерживает однонаправленную передачу 32-разрядных слов по двухпроводной витой паре, используя формат кодирования RZ. Доступна возможность запрограммировать 32-й бит либо как данные, либо как бит паритета. В случае формирования бита паритета, программируется его четность или нечетность. Каждый приемник и передатчик использует собственный буфер FIFO для хранения данных. Размеры буфера FIFO варьируются от 32×32 до 256×32. Статус наполненности FIFO определяется на основе соответствующих бит статуса для каждого FIFO. Контроллер поддерживает различные скорости приема и передачи данных. Работа контроллера осуществляется на базовой частоте 1 МГц, что позволяет обнаруживать ошибки в скорости приема/передачи данных, а также в паузах между сообщениями.

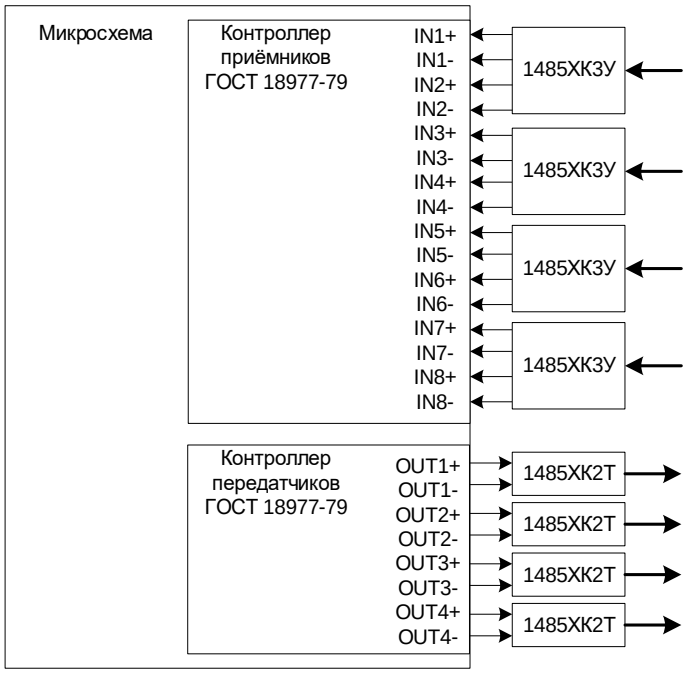



Рисунок 92 – Структура блока контроллера интерфейса по ГОСТ 18977-79

Особенности:

– симплексный режим приема/передачи со скоростями 12,5 кГц или 100 кГц;

– фильтрация входных данных на базе меток 32×8 и двух бит Источник/Приемник для каждого приемника;

– возможность передачи 32 бита, как данных, так и паритета;

– выбор четности/нечетности бита паритета;

– размеры буферов FIFO передатчиков: одно 256×32, три 64×32;

– размеры буферов FIFO приемников: два 256×32, четыре 64×32, два 32×32;

– возможность формирования прерываний при разных статусах наполненности буферов FIFO и при возникновении ошибок скорости передачи слова и паузы между словами;

– маскирование прерываний.

## 24.1 Формат слова

Слова в интерфейсе ARINC всегда 32-разрядные, и включают в себя пять полей: паритет, SSM, данные, источник/приемник, метка. Биты передаются младшими разрядами вперед, за исключением метки, которая передается старшими разрядами вперед. В результате можно описать порядок следования бит по шине ARINC следующим образом:

8, 7, 6, 5, 4, 3, 2, 1, 9, 10, 11, 12, 13…32.

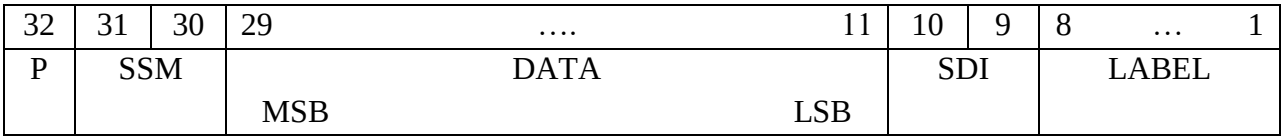



Рисунок 93 – Формат слова

Старший разряд всегда бит паритета. Стандартом установлено, что бит паритета должен дополнять слово до нечетного. Таким образом, количество единиц в 32-разрядном слове должно быть нечетным. Например, если биты 1-31 содержат четное количество единиц, то бит паритета должен быть установлен в единицу, с другой стороны, если биты 1-31 содержат нечетное количество единиц, то бит паритета должен быть сброшен в ноль.

Биты 31 и 30 содержат знак или статус. В контроллере эти биты рассматриваются как обычные данные и помещаются в FIFO вместе с полем данных без изменений и дополнительной обработки.

Как пример биты 31 и 30 могут кодировать следующие характеристики, представленные в таблице 318.

Таблица 318 – Биты 30, 31

| Бит | | Значение |
|---|---|---|
| 31 | 30 | |
| 0 | 0 | плюс, север, восток, справа, к, выше |
| 0 | 1 | не вычислительные данные |
| 1 | 0 | функциональный тест |
| 1 | 1 | минус, юг, запад, слева, от, ниже |

Биты 10 и 9 позволяют распознать Источник/Приемник данных. Это применяется при нескольких приемниках на шине ARINC, чтобы определить, для кого из них предназначаются данные. В системе со сложной структурой эти биты могут также использоваться, чтобы определить источник передачи. В остальных случаях эти разряды используются как данные. Следует отметить, что в интерфейсе ARINC на одной витой паре может быть один передатчик и до 20 приемников. Если включена функция проверки этих

бит, то при их несовпадении с битами, заданными программно в контроллере, сообщение не будет помещено в FIFO.

Биты с первого по восьмой позволяют идентифицировать тип данных оставшейся части слова, следовательно, методы преобразования, применяемые к данным. Помимо этого, в контроллере метки используются для фильтрации входных данных, то есть если метка в принятом сообщении не соответствует ни одной из меток определенной в памяти меток приемного канала, то данные не помещаются в FIFO. Это может служить аналогом того, что приемник не может интерпретировать метод обработки этих данных, следовательно, эти данные предназначены для другого приемника.

В случае если приемник принимает данные с неправильным битом паритета, они также не будут помещены в FIFO.

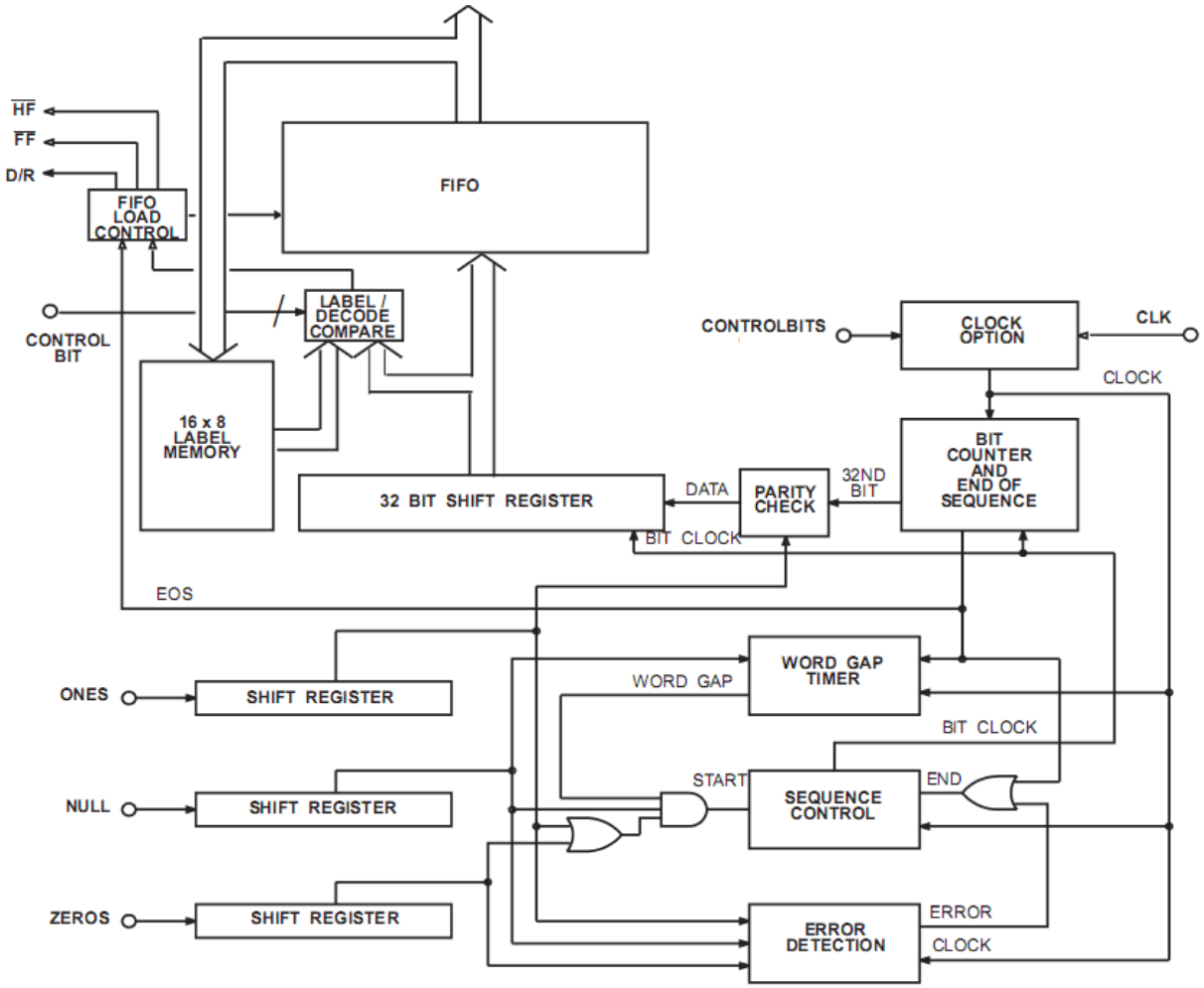



Рисунок 94 – Структурная схема канала приема

Представленная на рисунке 94 схема работает на частоте CLK = 1 МГц, в этом случае ошибка обнаружения бита в линии не будет составлять более 0,1 %.

Сдвиговые регистры длиной 10 бит, предназначенные для обнаружения в линии трех последовательностей единиц (Ones), нулей (Zeros) и отсутствие сигнала (Null), позволяют считать данные действительными. В дополнении к этому для бит данных, One или Zero в верхних битах сдвигового регистра должны сопровождаться Null в нижних битах в переделах битового интервала. В пределах паузы между сообщениями, три последовательных бита Null должны быть сэмплированы в верхней и нижней части

сдвигового регистра Null. В этом случае гарантируется минимальная ширина импульса данных.

Каждый бит данных должен быть обнаружен в пределах от 8 до 12 сэмплов. В этом случае скорость передачи считается верной.

Таймер паузы между сообщениями сэмплирует сдвиговый регистр Null каждые 10 входных тактов (или 80 тактов для скорости 12,5 кГц) после последнего полученного бита данных. Если Null обнаружен, то таймер инкрементируется. Значение таймера равное трем разрешает следующий прием.

Схема паритета считает количество принятых единиц, включая бит паритета. Если результат нечетный, то на выходе схемы формируется сигнал равный нулю.

После того как приняты все 32 бита логика приемника формирует сигнал конец последовательности (EOS). В зависимости от состояния бит LB_EN, SD_EN, SDI1, SDI2 регистра управления принимается решение о загрузке принятых данных в FIFO. Если в принятом слове биты 9 и 10 не соответствует правилам или не совпала метка, то слово не загружается в FIFO. В таблице 319 показано, в каком случае происходит загрузка FIFO принятыми данными.

Таблица 319 – Загрузка FIFO принятыми данными

| LB_EN | Результат сравнения слова ARINC с меткой | SD_EN | Результат сравнения бит 9,10 слова ARINC с SDI1, SDI2 | FIFO |
|---|---|---|---|---|
| 0 | X | 0 | X | Загружается |
| 1 | не совпала | 0 | X | Игнорируются |
| 1 | совпала | 0 | X | Загружается |
| 0 | X | 1 | не совпали | Игнорируются |
| 0 | X | 1 | совпали | Загружается |
| 1 | совпала | 1 | не совпали | Игнорируются |
| 1 | не совпала | 1 | совпали | Игнорируются |
| 1 | не совпала | 1 | не совпали | Игнорируются |
| 1 | совпала | 1 | совпали | Загружается |

Если хотя бы одно слово загружено в FIFO, то устанавливается в единицу сигнал DR, что отражается в регистре статуса контроллера. Флаг остается в неизменном состоянии, пока последнее слово не будет прочитано из FIFO, и оно не будет пустым. Применяются еще два сигнала, характеризующие состояние FIFO, а именно HF означает, что FIFO наполовину полно и FF означает, что FIFO полно. Установка этих сигналов также отражается в регистре статуса. Каждый из этих флагов может быть источником прерывания, в случае если оно разрешено соответствующим битом маскирования регистра управления.

Фактически пауза 4T между словами начинает измеряться контроллером от последнего среза импульса на любой из шин (+ или -) первого слова и до среза первого импульса (корректной длительности) на любой из шин (+ или -) следующего слова. Таким образом, если между словами, передаваемыми друг за другом, пауза более 3,5T, такие слова будут приняты без формирования флага ошибки. По стандарту (ГОСТ 18977-79)

пауза 4T между смежными словами измеряется от последнего среза импульса на любой из шин (+ или -) первого слова и до фронта первого импульса на любой из шин (+ или -) второго слова.

Для корректного приема следующего слова необходимо обязательно программно сбрасывать канал, по которому зафиксирована ошибка, с помощью бита CH_ENx регистра CONTROL1 контроллера интерфейса приемников ARINC-429.

## 24.2 Структурная схема канала передачи

Если флаг TX_R в состоянии логической единицы, это значит, что FIFO пусто, и в него могут быть загружены 31- или 32-битные данные. Количество слов данных определяется размером FIFO для выбранного канала передачи. Если флаг TX_R в состоянии логического нуля, тогда только в доступные в FIFO ячейки можно загрузить данные. Если FIFO заполнено полностью, флаг FFx установлен в единицу, то FIFO игнорирует дальнейшие попытки записи в него. FIFO наполовину полно, если установлен флаг HFx, в этом случае можно загрузить данными оставшуюся половину буфера FIFO.

В нормальном режиме работы 32-ой бит передаваемых данных является битом паритета. Четность или нечетность выбирается битом ODD регистра управления. Если бит разрешения паритета (EN_PAR) сброшен в ноль, то тогда 32-ой бит передается, как бит данных из FIFO.

Если бит CH_EN установлен в единицу и FIFO передачи не пусто, то начинается передача слов данных из FIFO до тех пор, пока FIFO не будет пусто или не будет сброшен бит CH_EN.

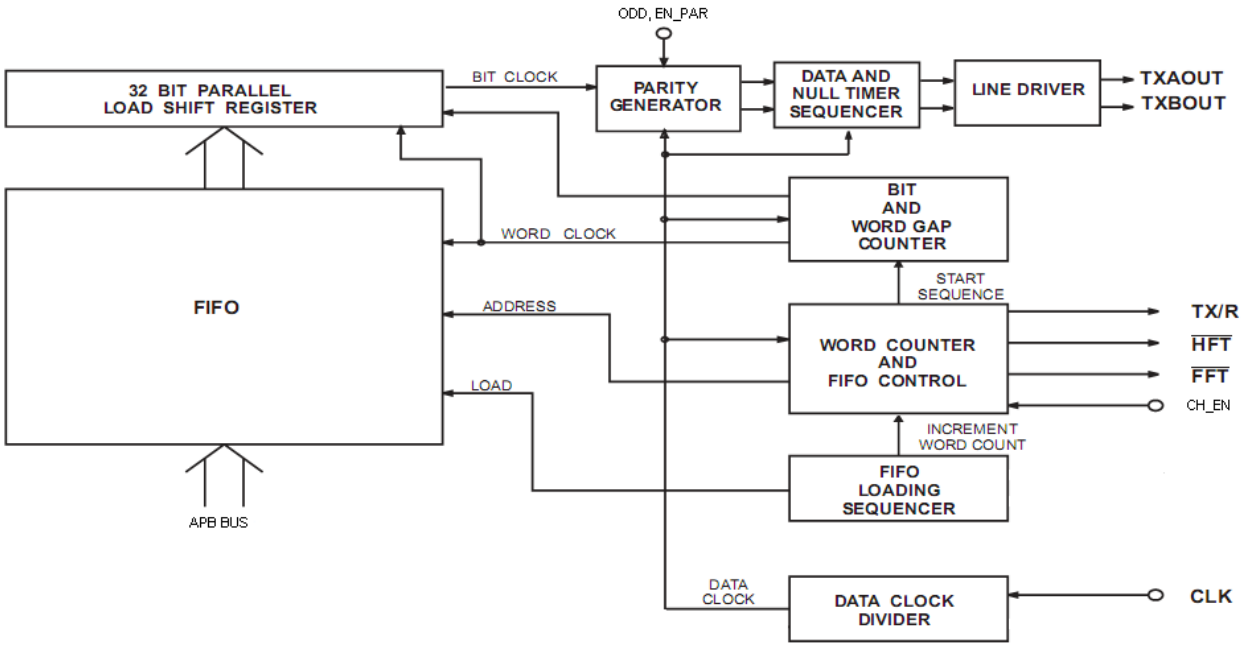



Рисунок 95 – Структурная схема канала передачи

## 24.3 Описание регистров контроллера ГОСТ 18977-79

Таблица 320 – Описание регистров контроллера ГОСТ 18977-79

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x400D_0000 | ARINC429R | Контроллер интерфейса приемников ARINC-429 |
| Смещение | | |
| 0x0000 | CONTROL1 | Регистр управления 1 приемником |
| 0x0004 | CONTROL2 | Регистр управления 2 приемником |
| 0x0008 | CONTROL3 | Регистр управления 3 приемником |
| 0x000C | STATUS1 | Регистр состояния 1 приемника |
| 0x0010 | STATUS2 | Регистр состояния 2 приемника |
| 0x0014 | CONTROL4 | Регистр настройки индивидуального делителя частоты каналов 1-4 |
| 0x0018 | CONTROL5 | Регистр настройки индивидуального делителя частоты каналов 5-8 |
| 0x001C | CHANNEL | Регистр номера канала приемников |
| 0x0020 | LABEL | FIFO меток |
| 0x0024 | DATA_R | FIFO принимаемых данных |
| 0x0030 | DATA_R1 | FIFO принимаемых данных канала 1 при CHANNEL=14 |
| 0x0034 | DATA_R2 | FIFO принимаемых данных канала 2 при CHANNEL=14 |
| 0x0038 | DATA_R3 | FIFO принимаемых данных канала 3 при CHANNEL=14 |
| 0x003C | DATA_R4 | FIFO принимаемых данных канала 4 при CHANNEL=14 |
| 0x0040 | DATA_R5 | FIFO принимаемых данных канала 5 при CHANNEL=14 |
| 0x0044 | DATA_R6 | FIFO принимаемых данных канала 6 при CHANNEL=14 |
| 0x0048 | DATA_R7 | FIFO принимаемых данных канала 7 при CHANNEL=14 |
| 0x004C | DATA_R8 | FIFO принимаемых данных канала 8 при CHANNEL=14 |
| 0x0068 | INTMASK | Регистр индивидуальной настройки разрешения прерывания по заполненности FIFO каждого канала |
| 0x0070 | CONTROL8 | Регистр управления 8 приемником |
| 0x0074 | CONTROL9 | Регистр управления 9 приемником |
| 0x0078 | CONTROL10 | Регистр управления 10 приемником |
| 0x007C | CONTROL11 | Регистр управления 11 приемником |
| 0x400E_0000 | ARINC429T | Контроллер интерфейса передатчиков ARINC-429 |
| Смещение | | |
| 0x0000 | CONTROL1 | Регистр управления 1 передатчиком |
| 0x0004 | CONTROL2 | Регистр управления 2 передатчиком |
| 0x0008 | STATUS | Регистр состояния передатчиков |
| 0x000C | DATA1_T | Регистр передаваемых данных канала 1 |
| 0x0010 | DATA2_T | Регистр передаваемых данных канала 2 |
| 0x0014 | DATA3_T | Регистр передаваемых данных канала 3 |
| 0x0018 | DATA4_T | Регистр передаваемых данных канала 4 |
| 0x001C | CONTROL3 | Регистр настройки индивидуального делителя частоты каналов |
| 0x0020 | CONTROL4 | Регистр управления 4 передатчиком |
| 0x0024 | CONTROL5 | Регистр управления 5 передатчиком |

### 24.3.1 CONTROL1

Таблица 321 – Регистр управления 1 приемником CONTROL1

| Номер | 31 | 30 | 29 | 28 | 27…22 |
|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | U |
| Сброс | 0 | 0 | 0 | 0 | |
| | DIV3 | DIV2 | DIV1 | DIV0 | - |

| Номер | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13..8 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | U |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | CLK8 | CLK7 | CLK6 | CLK5 | CLK4 | CLK3 | CLK2 | CLK1 | - |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | CH_EN8 | CH_EN7 | CH_EN6 | CH_EN5 | CH_EN4 | CH_EN3 | CH_EN2 | CH_EN1 |

Таблица 322 – Описание бит регистра CONTROL1

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…28 | DIV[3:0] | Делитель частоты ядра до 1 МГц.<br>Содержит младшие четыре разряда значения, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц. Значение частоты не может быть более 125 МГц |
| 27…22 | - | Зарезервировано |
| 21…14 | CLK8-CLK1 | Скорость приема данных:<br>1 – частота приема данных= опорная частота/80<br>(12,5 кГц, если DIV не равен 0);<br>0 – частота приема данных= опорная частота/10<br>(100 кГц, если DIV не равен 0).<br>Опорная частота для каждого канала задается делителем в регистрах CONTROL4 и CONTROL5, если DIV=0 |
| 13…8 | - | Зарезервировано |
| 7…0 | CH_EN8-CH_EN1 | Разрешение работы канала:<br>1 – прием по каналу разрешен;<br>0 – канал приема находится в состоянии сброса |

### 24.3.2 CONTROL2

Таблица 323 – Регистр управления 2 приемником CONTROL2

| Номер | 31 | 30…25 | 24 | 23 | 22 | 21 |
|---|---|---|---|---|---|---|
| Доступ | R/W | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 |
| | DA | - | SD_EN8 | SD_EN7 | SD_EN6 | SD_EN5 |

| Номер | 20 | 19 | 18 | 17 | 16…11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | U | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SD_EN4 | SD_EN3 | SD_EN2 | SD_EN1 | - | LB_EN8 | LB_EN7 | LB_EN6 |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | LB_EN5 | LB_EN4 | LB_EN3 | LB_EN2 | LB_EN1 | DIV6 | DIV5 | DIV4 |

Таблица 324 – Описание бит регистра CONTROL2

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31 | DA | Бит прямого доступа в FIFO1 и FIFO2:<br>1 – память приема каналов 1 и 2 работает не в режиме FIFO (биты состояния наличия данных в FIFO: DRx, HFx, FFx становятся неактивными), доступ к ней осуществляется в диапазоне адресов:<br>0x400D1000 – 0x400D13FC для FIFO1;<br>0x400D1400 – 0x400D17FC для FIFO2;<br>0 – обычный режим работы FIFO.<br>Адрес должен быть кратен четырем, обращения только по 32-разрядным словам.<br>CHANNEL = 0 или 1 в зависимости от канала.<br>При приеме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения.<br><br>Особенность работы в режиме прямого доступа к FIFO.<br>В асинхронных системах возможна ситуация, когда одновременно выполнятся запись и чтение одной ячейки. Из-за того, что у операции записи в двухпортовой памяти буфера FIFO над операцией чтения приоритет, при такой ситуации на стороне приемника из буфера может быть считано некорректное значение. Список рекомендаций для того, чтобы этого избежать:<br>- использовать при работе с приемным каналом режим FIFO;<br>- разграничить во времени операции чтения/записи буфера приемника;<br>- в программном коде выполнять два чтения одной и той же ячейки приемного буфера подряд с запретом работы блока DMA и |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| | | прерываний на время этих чтений. Программный код, реализующий данный способ обхода:<br><br>while( (MDR_DMA->STATUS & 0x70) != 0); // Перед выполнением двух чтений из приемного буфера ожидаем, что все транзакции по DMA завершены<br>MDR_DMA->CFG &= ~ (1 << 0); // Запрет работы DMA на момент двух чтений из приемного буфера<br>NVIC_DisableIRQ(interrupts); // Выключение обработки прерываний на уровне NVIC<br>real = (*(uint32_t*)(0x400D1000+addr); // Первое чтение из приемного буфера (в данном случае используется базовый адрес буфера первого приемника) в переменную<br>real = (*(uint32_t*)(0x400D1000+addr); // Второе чтение из приемного буфера в переменную<br>NVIC_EnableIRQ(interrupts); // Включение обработки прерываний на уровне NVIC. Если за время двух чтений был запрос на прерывание, то после исполнения данной строчки кода вызовется его обработчик<br>MDR_DMA->CFG \|= (1 << 0); // Обратное разрешение работы DMA |
| 30...25 | - | Зарезервировано |
| 24...17 | SD_EN8 – SD_EN1 | Разрешение декодирования бит данных 9 и 10:<br>1 – разрешено сравнение бит данных 9 и 10 со значением бит SDI1 и SDI2 соответствующего канала;<br>0 – декодирование отключено, все принятые данные помещаются в FIFO |
| 16...11 | | Зарезервировано |
| 10...3 | LB_EN8 – LB_EN1 | Разрешение обнаружения меток:<br>1 – разрешено обнаружение меток в первых восьми принятых битах;<br>0 – обнаружение отключено, все принятые данные помещаются в FIFO |
| 2...0 | DIV[6:4] | Делитель частоты ядра до 1 МГц.<br>Содержит старшие три разряда значения, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц |

### 24.3.3 CONTROL3

Таблица 325 – Регистр управления 3 приемником CONTROL3

| Номер | 31 | 30 | 29 | 28 | 27…22 | 21 |
|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | U | R/W |
| Сброс | | 0 | 0 | 0 | 0 | 0 |
| | INTEHF | INTEFF | INTEER | INTEDR | - | SDI2_8 |

| Номер | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13…8 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | U |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SDI2_7 | SDI2_6 | SDI2_5 | SDI2_4 | SDI2_3 | SDI2_2 | SDI2_1 | - |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SDI1_8 | SDI1_7 | SDI1_6 | SDI1_5 | SDI1_4 | SDI1_3 | SDI1_2 | SDI1_1 |

Таблица 326 – Описание бит регистра CONTROL3

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31 | INTEHF | Глобальное разрешение прерывания FIFO наполовину полно:<br>1 – разрешено прерывание, если FIFO наполовину полно;<br>0 – прерывание запрещено |
| 30 | INTEFF | Глобальное разрешение прерывания FIFO полно:<br>1 – разрешено прерывание при переполнении FIFO данных;<br>0 – прерывание запрещено |
| 29 | INTEER | Глобальное разрешение прерывания ошибка приема:<br>1 – разрешено прерывания при возникновении ошибки в скорости приема или во времени паузы 4T между сообщениями (для сброса ошибки необходимо сбросить канал битом CH_EN);<br>0 – прерывание запрещено |
| 28 | INTEDR | Глобальное разрешение прерывания наличие данных в FIFO:<br>1 – разрешено прерывание, если FIFO приема данных не пусто;<br>0 – прерывание запрещено |
| 27...22 | - | Зарезервировано |
| 21...14 | SDI2_1–<br>SDI2_8 | Бит сравнения SDI2.<br>Значение бита сравнивается с битом 10 принимаемых данных, если установлен бит SD_EN соответствующего канала |
| 13...8 | - | Зарезервировано |
| 7...0 | SDI1_1 –<br>SDI1_8 | Бит сравнения SDI1.<br>Значение бита сравнивается с битом 9 принимаемых данных, если установлен бит SD_EN соответствующего канала |

### 24.3.4 CONTROL4

Таблица 327 – Регистр управления 4 приемником CONTROL4

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | DIV_CH4 | DIV_CH3 | DIV_CH2 | DIV_CH1 |

Таблица 328 – Описание бит регистра CONTROL4

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...24 | DIV_CH4 | Делитель частоты ядра для получения опорной частоты канала 4 |
| 23...16 | DIV_CH3 | Делитель частоты ядра для получения опорной частоты канала 3 |
| 15...8 | DIV_CH2 | Делитель частоты ядра для получения опорной частоты канала 2 |
| 7...0 | DIV_CH1 | Делитель частоты ядра для получения опорной частоты канала 1 |

### 24.3.5 CONTROL5

Таблица 329 – Регистр управления 5 приемником CONTROL5

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | DIV_CH8 | DIV_CH7 | DIV_CH6 | DIV_CH5 |

Таблица 330 – Описание бит регистра CONTROL5

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...24 | DIV_CH8 | Делитель частоты ядра для получения опорной частоты канала 8 |
| 23…16 | DIV_CH7 | Делитель частоты ядра для получения опорной частоты канала 7 |
| 15…8 | DIV_CH6 | Делитель частоты ядра для получения опорной частоты канала 6 |
| 7...0 | DIV_CH5 | Делитель частоты ядра для получения опорной частоты канала 5 |

### 24.3.6 CONTROL8

Таблица 331 – Регистр управления 8 приемником CONTROL8

| Номер | 31…30 | 29 | 28 | 27…22 | 21…14 | 13…8 | 7…0 |
|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | U | R/W | U | R/W |
| Сброс | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| | - | DA1 | DA0 | - | ODD | - | ENPAR |

Таблица 332 – Описание бит регистра CONTROL8

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...30 | - | Зарезервировано |
| 29 | DA1 | Бит индивидуального разрешения прямого доступа в FIFO2 (используется при DA = 0 в регистре CONTROL2):<br>1 – память приема канала 2 работает не в режиме FIFO (биты состояния наличия данных в FIFO: DR2, HF2, FF2 становятся неактивными), доступ к ней осуществляется в диапазоне адресов 0x400D1400 – 0x400D17FC; |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| | | 0 – обычный режим работы FIFO.<br>Адрес должен быть кратен четырем, обращения только по 32-разрядным словам.<br>CHANNEL = 1 или 14<br>При приеме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения.<br><br>Особенность работы в режиме прямого доступа к FIFO.<br>В асинхронных системах возможна ситуация, когда одновременно выполнятся запись и чтение одной ячейки. Из-за того, что у операции записи в двухпортовой памяти буфера FIFO над операцией чтения приоритет, при такой ситуации на стороне приемника из буфера может быть считано некорректное значение. Список рекомендаций для того, чтобы этого избежать:<br>- использовать при работе с приемным каналом режим FIFO;<br>- разграничить во времени операции чтения/записи буфера приемника;<br>- в программном коде выполнять два чтения одной и той же ячейки приемного буфера подряд с запретом работы блока DMA и прерываний на время этих чтений. Программный код, реализующий данный способ обхода:<br><br>`while( (MDR_DMA->STATUS & 0x70) != 0);` // Перед выполнением двух чтений из приемного буфера ожидаем, что все транзакции по DMA завершены<br>`MDR_DMA->CFG &= ~(1 << 0);` // Запрет работы DMA на момент двух чтений из приемного буфера<br>`NVIC_DisableIRQ(interrupts);` // Выключение обработки прерываний на уровне NVIC<br>`real = (*(uint32_t*)(0x400D1000+addr);` // Первое чтение из приемного буфера (в данном случае используется базовый адрес буфера первого приемника) в переменную<br>`real = (*(uint32_t*)(0x400D1000+addr);` // Второе чтение из приемного буфера в переменную<br>`NVIC_EnableIRQ(interrupts);` // Включение обработки прерываний на уровне NVIC. Если за время двух чтений был запрос на прерывание, то после исполнения данной строчки кода вызовется его обработчик<br>`MDR_DMA->CFG |= (1 << 0);` // Обратное разрешение работы DMA |

| | | |
|---|---|---|
| 28 | DA0 | Бит индивидуального разрешения прямого доступа в FIFO1 (используется при DA=0 в регистре CONTROL2):<br>1 – память приема канала 1 работает не в режиме FIFO (биты состояния наличия данных в FIFO: DR1, HF1, FF1 становятся неактивными), доступ к ней осуществляется в диапазоне адресов 0x400D1000 – 0x400D13FC;<br>0 – обычный режим работы FIFO.<br>Адрес должен быть кратен четырем, обращения только по 32-разрядным словам.<br>CHANNEL = 0 или 14.<br>При приеме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения.<br><br>Особенность работы в режиме прямого доступа к FIFO.<br>В асинхронных системах возможна ситуация, когда одновременно выполнятся запись и чтение одной ячейки. Из-за того, что у операции записи в двухпортовой памяти буфера FIFO над операцией чтения приоритет, при такой ситуации на стороне приемника из буфера может быть считано некорректное значение. Список рекомендаций для того, чтобы этого избежать:<br>- использовать при работе с приемным каналом режим FIFO;<br>- разграничить во времени операции чтения/записи буфера приемника;<br>- в программном коде выполнять два чтения одной и той же ячейки приемного буфера подряд с запретом работы блока DMA и прерываний на время этих чтений. Программный код, реализующий данный способ обхода:<br><br>`while( (MDR_DMA->STATUS & 0x70) != 0);` // Перед выполнением двух чтений из приемного буфера ожидаем, что все транзакции по DMA завершены<br>`MDR_DMA->CFG &= ~(1 << 0);` // Запрет работы DMA на момент двух чтений из приемного буфера<br>`NVIC_DisableIRQ(interrupts);` // Выключение обработки прерываний на уровне NVIC<br>`real = (*(uint32_t*)(0x400D1000+addr);` // Первое чтение из приемного буфера (в данном случае используется базовый адрес буфера первого приемника) в переменную<br>`real = (*(uint32_t*)(0x400D1000+addr);` // Второе чтение из приемного буфера в переменную<br>`NVIC_EnableIRQ(interrupts);` // Включение обработки прерываний на уровне NVIC. Если за время двух чтений был запрос на прерывание, то после исполнения данной строчки кода вызовется его обработчик<br>`MDR_DMA->CFG |= (1 << 0);` // Обратное разрешение работы DMA |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 27…22 | - | Зарезервировано |
| 21...14 | ODD | Выбор четности или нечетности бита паритета для каналов 1-8:<br>1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в 1, в противном случае в 0);<br>0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в 1, в противном случае в 0) |
| 13…8 | - | Зарезервировано |
| 7...0 | ENPAR | Разрешение 32 бита паритета для каналов 1-8:<br>1 – разрешена передача 32-м битом бита паритета;<br>0 – разрешена передача 32-м битом бита данных.<br>Запрещено сбрасывать этот бит в ноль в штатном режиме работы контроллера |

### 24.3.7 CONTROL9

Таблица 333 – Регистр управления 9 приемником CONTROL9

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | - | - | - | ENSYNC |

Таблица 334 – Описание бит регистра CONTROL9

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...8 | - | Зарезервировано |
| 7...0 | ENSYNC[7:0] | Разрешение работы входов приемника в режиме данных и синхросигнала:<br>1 – разрешено;<br>0 – запрещено.<br>При установленном бите ENSYNC для соответствующего канала вход IN_A работает как данные (D), вход IN_B как синхросигнал (SYN) |

### 24.3.8 CONTROL10

Таблица 335 – Регистр управления 10 приёмником CONTROL10

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 3 | 3 | 3 | 3 |
| | R_PAUSE4 | R_PAUSE3 | R_PAUSE2 | R_PAUSE1 |

Таблица 336 – Описание бит регистра CONTROL10

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...24 | R_PAUSE4 | Значение паузы (R_PAUSE4+1) между сообщениями для канала приёма 4 в периодах скорости |
| 23...16 | R_PAUSE3 | Значение паузы (R_PAUSE3+1) между сообщениями для канала приёма 3 в в периодах скорости |
| 15...8 | R_PAUSE2 | Значение паузы (R_PAUSE2+1) между сообщениями для канала приёма 2 в в периодах скорости |
| 7...0 | R_PAUSE1 | Значение паузы (R_PAUSE1+1) между сообщениями для канала приёма 1 в периодах скорости |

### 24.3.9 CONTROL11

Таблица 337 – Регистр управления 11 приёмником CONTROL11

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 3 | 3 | 3 | 3 |
| | R_PAUSE8 | R_PAUSE7 | R_PAUSE6 | R_PAUSE5 |

Таблица 338 – Описание бит регистра CONTROL11

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...24 | R_PAUSE8 | Значение паузы (R_PAUSE8+1) между сообщениями для канала приёма 8 в периодах скорости |
| 23...16 | R_PAUSE7 | Значение паузы (R_PAUSE7+1) между сообщениями для канала приёма 7 в периодах скорости |
| 15...8 | R_PAUSE6 | Значение паузы (R_PAUSE6+1) между сообщениями для канала приёма 6 в периодах скорости |
| 7...0 | R_PAUSE5 | Значение паузы (R_PAUSE5+1) между сообщениями для канала приёма 5 в периодах скорости |

### 24.3.10 INTMASK

Таблица 339 – Регистр индивидульного разрешения прерываний каналов INTMASK

| Номер | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | IEHF8 | IEFF8 | IEER8 | IEDR8 | IEHF7 | IEFF7 | IEER7 | IEDR7 |

| Номер | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | IEHF6 | IEFF6 | IEER6 | IEDR6 | IEHF5 | IEFF5 | IEER5 | IEDR5 |

| Номер | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | IEHF4 | IEFF4 | IEER4 | IEDR4 | IEHF3 | IEFF3 | IEER3 | IEDR3 |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | IEHF2 | IEFF2 | IEER2 | IEDR2 | IEHF1 | IEFF1 | IEER1 | IEDR1 |

Значения битов регистра INTMASK учитываются только в случае, если соответствующий бит в регистре CONTROL3 (INTEHF, INTEFF, INTEER, INTEDR) установлен в «0».

Таблица 340 – Описание бит регистра INTMASK

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…28 | | Для канала 8 |
| 27…24 | | Для канала 7 |
| 23…20 | | Для канала 6 |
| 19…16 | | Для канала 5 |
| 15…12 | | Для канала 4 |
| 11…8 | | Для канала 3 |
| 7…5 | | Для канала 2 |
| 3…0 | | Для канала 1.<br>IEDR1:<br>1 – разрешено прерывание, если FIFO приема данных не пусто;<br>0 – прерывание запрещено.<br>IEER1:<br>1 – разрешено прерывания при возникновении ошибки в скорости приема или во времени паузы 4T между сообщениями (для сброса ошибки необходимо сбросить канал битом CH_EN);<br>0 – прерывание запрещено.<br>IEFF1:<br>1 – разрешено прерывание при переполнении FIFO данных;<br>0 – прерывание запрещено.<br>IEHF1:<br>1 – разрешено прерывание, если FIFO наполовину полно;<br>0 – прерывание запрещено |

### 24.3.11 STATUS1

Таблица 341 – Регистр состояния 1 приемника STATUS1

| Номер | 31…22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | ERR8 | ERR7 | ERR6 | ERR5 | ERR4 | ERR3 | ERR2 | ERR1 |

| Номер | 13…8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | DR8 | DR7 | DR6 | DR5 | DR4 | DR3 | DR2 | DR1 |

Таблица 342 – Описание бит регистра STATUS1

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…22 | - | Зарезервировано |
| 21…14 | ERR8 – ERR1 | Бит ошибки:<br>0 – нет ошибок;<br>1 – возникла ошибка приема |
| 13…8 | - | Зарезервировано |
| 7…0 | DR8 – DR1 | Бит наличия данных в FIFO:<br>0 – FIFO пусто;<br>1 – FIFO содержит данные |

### 24.3.12 STATUS2

Таблица 343 – Регистр состояния 2 приемника STATUS2

| Номер | 31….22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | HF8 | HF7 | HF6 | HF5 | HF4 | HF3 | HF2 | HF1 |

| Номер | 13…8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | FF8 | FF7 | FF6 | FF5 | FF4 | FF3 | FF2 | FF1 |

Таблица 344 – Описание бит регистра STATUS2

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…22 | - | Зарезервировано |
| 21…14 | HF8 – HF1 | Бит наполненности FIFO:<br>0 – FIFO не наполнено до половины;<br>1 – FIFO наполнено до половины |
| 13…8 | - | Зарезервировано |
| 7…0 | FF8 – FF1 | Бит полноты FIFO:<br>0 – FIFO не полно;<br>1 – FIFO полно |

### 24.3.13 CHANNEL

Таблица 345 – Регистр номера канала приемников CHANNEL

| Номер | 31…4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 |
| | - | CHAN3 | CHAN2 | CHAN1 | CHAN0 |

Таблица 346 – Описание бит регистра CHANNEL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | | Зарезервировано |
| 3…0 | CHAN[3:0] | Биты выбора канала.<br>Значение этих бит определяет к данным и меткам, какого канала будет осуществляться доступ:<br>0000 – канал 1;<br>0001 – канал 2;<br>0010 – канал 3;<br>0011 – канал 4;<br>0100 – канал 5;<br>0101 – канал 6;<br>0110 – канал 7;<br>0111 – канал 8;<br>1110 – прямой доступ в память каналов 1 и 2 при DA=1, либо доступ по индивидуальным адресам для каждого FIFO |

### 24.3.14 LABEL

FIFO меток, с которыми сравниваются первые восемь принимаемых бит, если установлен LB_EN бит соответствующего канала. Размер FIFO для каждого канала 32×8. Выбор необходимого FIFO осуществляется переключением канала в регистре CHANNEL. При записи или чтении FIFO указатель FIFO инкрементируется. Для возврата в начало FIFO необходимо осуществить запись в регистр CHANNEL.

### 24.3.15 DATA_R

FIFO принимаемых данных.

В FIFO помещаются 32-разрядные данные, принимаемые из соответствующего канала. Размер FIFO для каждого канала разный:

- канал 1 – 256×32;
- канал 2 – 256×32;
- канал 3 – 64×32;
- канал 4 – 64×32;
- канал 5 – 64×32;
- канал 6 – 64×32;

– канал 7 – 32×32;
– канал 8 – 32×32.

Выбор необходимого FIFO осуществляется переключением канала в регистре CHANNEL. Наличие или отсутствие данных в FIFO контролируется битами статуса DR, HF, FF соответствующего канала.

### 24.3.16 DATA_R1 – DATA_R8

FIFO принимаемых данных в случае записи в регистр CHANNEL значения 14.

### 24.3.17 CONTROL1

Таблица 347 – Регистр управления 1 передатчиком CONTROL1

| Номер | 31...21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 |
| | - | INTE_HFT2 | INTE_TXR2 | INTE_FFT2 | INTE_HFT1 | INTE_TXR1 |

| Номер | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | INTE_FFT1 | DIV6 | DIV5 | DIV4 | DIV3 | DIV2 | DIV1 | DIV0 |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ODD2 | EN_PAR2 | CLK2 | CH_EN2 | ODD1 | EN_PAR1 | CLK1 | CH_EN1 |

Таблица 348 – Описание бит регистра CONTROL1

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31….21 | - | Зарезервировано |
| 20 | INTE_HFT2 | Разрешение прерывания при заполнении наполовину буфера FIFO канала 2:<br>1 – разрешено прерывание, FIFO наполовину полно;<br>0 – прерывание запрещено |
| 19 | INTE_TXR2 | Разрешение прерывания при опустошении буфера FIFO канала 2:<br>1 – разрешено прерывание, буфер FIFO пуст;<br>0 – прерывание запрещено |
| 18 | INTE_FFT2 | Разрешение прерывания при полном заполнении буфера FIFO канала 2:<br>1 – разрешено прерывание при полном заполнении FIFO данных;<br>0 – прерывание запрещено |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 17 | INTE_HFT1 | Разрешение прерывания при заполнении наполовину буфера FIFO канала 1:<br>1 – разрешено прерывание FIFO наполовину полно;<br>0 –прерывание запрещено |
| 16 | INTE_TXR1 | Разрешение прерывания при опустошении буфера FIFO канала 1:<br>1 – разрешено прерывание FIFO передачи данных пусто;<br>0 – прерывание запрещено |
| 15 | INTE_FFT1 | Разрешение прерывания при полном заполнении буфера FIFO канала 1:<br>1 – разрешено прерывание при полном заполнении FIFO данных;<br>0 – прерывание запрещено |
| 14…8 | DIV[6:0] | Делитель частоты ядра до 1 МГц.<br>Содержит значение, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц |
| 7 | ODD2 | Выбор четности или нечетности бита паритета для канала 2:<br>1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в «1», в противном случае в «0»);<br>0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в «1», в противном случае в «0») |
| 6 | EN_PAR2 | Разрешение 32 бита паритета для канала 2:<br>1 – разрешена передача 32-м битом бита паритета;<br>0 – разрешена передача 32-м битом бита данных |
| 5 | CLK2 | Скорость передачи данных по 2 каналу:<br>1 – частота передаваемых данных = опорная частота/80<br>(12,5 кГц, если DIV не равен нулю);<br>0 – частота передаваемых данных = опорная частота/10<br>(100 кГц, если DIV не равен нулю) |
| 4 | CH_EN2 | Разрешение работы канала 2:<br>1 – передача по каналу разрешена;<br>0 – канал передачи находится в состоянии сброса |
| 3 | ODD1 | Выбор четности или нечетности бита паритета для канала 1:<br>1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в «1», в противном случае в «0»);<br>0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в «1», в противном случае в «0») |
| 2 | EN_PAR1 | Разрешение 32 бита паритета для канала 1:<br>1 – разрешена передача 32-м битом бита паритета;<br>0 – разрешена передача 32-м битом бита данных |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 1 | CLK1 | Скорость передачи данных по 1 каналу:<br>1 – частота передаваемых данных = опорная частота/80<br>(12,5 кГц, если DIV не равен нулю);<br>0 – частота передаваемых данных = опорная частота/10<br>(100 кГц, если DIV не равен нулю) |
| 0 | CH_EN1 | Разрешение работы канала 1:<br>1 – передача по каналу разрешена;<br>0 – канал передачи находится в состоянии сброса |

### 24.3.18 CONTROL2

Таблица 349 – Регистр управления 2 передатчиком CONTROL2

| Номер | 31...21 | 20 | 19 | 18 | 17 | 16 | 15 | 14…8 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | U |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | - | INTE_ HFT4 | INTE_ TXR4 | INTE_ FFT4 | INTE_ HFT3 | INTE_ TXR3 | INTE_ FFT3 | - |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ODD4 | EN_PAR4 | CLK4 | CH_EN4 | ODD3 | EN_PAR3 | CLK3 | CH_EN3 |

Таблица 350 – Описание бит регистра CONTROL2

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…21 | - | Зарезервировано |
| 20 | INTE_HFT4 | Разрешение прерывания при заполнении наполовину буфера FIFO канала 4:<br>1 – разрешено прерывание, буфер FIFO наполовину полон;<br>0 – прерывание запрещено |
| 19 | INTE_TXR4 | Разрешение прерывания при опустошении буфера FIFO канала 4:<br>1 – разрешено прерывание, буфер FIFO пуст;<br>0 – прерывание запрещено |
| 18 | INTE_FFT4 | Разрешение прерывания при полном заполнении буфера FIFO канала 4:<br>1 – разрешено прерывание при полном заполнении FIFO данных;<br>0 – прерывание запрещено |
| 17 | INTE_HFT3 | Разрешение прерывания при заполнении наполовину буфера FIFO канала 3:<br>1 – разрешено прерывание, буфер FIFO наполовину полон;<br>0 –прерывание запрещено |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 16 | INTE_TXR3 | Разрешение прерывания при опустошении буфера FIFO канала 3:<br>1 – разрешено прерывание, буфер FIFO пуст;<br>0 – прерывание запрещено |
| 15 | INTE_FFT3 | Разрешение прерывания при полном заполнении буфера FIFO канала 3:<br>1 – разрешено прерывание при полном заполнении FIFO данных;<br>0 – прерывание запрещено |
| 14…8 | - | Зарезервировано |
| 7 | ODD4 | Выбор четности или нечетности бита паритета для канала 4:<br>1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в «1», в противном случае в «0»);<br>0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в «1», в противном случае в «0») |
| 6 | EN_PAR4 | Разрешение 32 бита паритета для канала 4:<br>1 – разрешена передача 32-м битом бита паритета;<br>0 – разрешена передача 32-м битом бита данных |
| 5 | CLK4 | Скорость передачи данных по 4 каналу:<br>1 – частота передаваемых данных = опорная частота/80<br>(12,5 кГц, если DIV не равен нулю);<br>0 – частота передаваемых данных = опорная частота/10<br>(100 кГц, если DIV не равен нулю) |
| 4 | CH_EN4 | Разрешение работы канала 4:<br>1 – передача по каналу разрешена;<br>0 – канал передачи находится в состоянии сброса |
| 3 | ODD3 | Выбор четности или нечетности бита паритета для канала 3:<br>1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, то бит паритета устанавливается в «1», в противном случае в «0»);<br>0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, то бит паритета устанавливается в «1», в противном случае в «0»). |
| 2 | EN_PAR3 | Разрешение 32 бита паритета для канала 3:<br>1 – разрешена передача 32-м битом бита паритета;<br>0 – разрешена передача 32-м битом бита данных |
| 1 | CLK3 | Скорость передачи данных по 3 каналу:<br>1 – частота передаваемых данных = опорная частота/80<br>(12,5 кГц, если DIV не равен нулю);<br>0 – частота передаваемых данных = опорная частота/10<br>(100 кГц, если DIV не равен нулю) |
| 0 | CH_EN3 | Разрешение работы канала 3:<br>1 – передача по каналу разрешена;<br>0 – канал передачи находится в состоянии сброса |

### 24.3.19 CONTROL3

Таблица 351 – Регистр управления 3 передатчиком CONTROL3

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | DIV_CH4 | DIV_CH3 | DIV_CH2 | DIV_CH1 |

Таблица 352 – Описание бит регистра CONTROL3

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…24 | DIV_CH4 | Делитель частоты ядра для получения опорной частоты канала 4 |
| 23…16 | DIV_CH3 | Делитель частоты ядра для получения опорной частоты канала 3 |
| 15…8 | DIV_CH2 | Делитель частоты ядра для получения опорной частоты канала 2 |
| 7…0 | DIV_CH1 | Делитель частоты ядра для получения опорной частоты канала 1 |

### 24.3.20 CONTROL4

Таблица 353 – Регистр управления 4 передатчиком CONTROL4

| Номер | 31…24 | 23…16 | 15…4 | 3…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | - | - | - | ENSYNC |

Таблица 354 – Описание бит регистра CONTROL4

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…4 | - | Зарезервировано |
| 3…0 | ENSYNC[3:0] | Разрешение работы выходов передатчика в режиме данных и синхросигнала:<br>1 – разрешено;<br>0 – запрещено.<br>При установленном бите ENSYNC для соответствующего канала выход OUT_A работает как данные (D), выход OUT_B как синхросигнал (SYN) |

### 24.3.21 CONTROL5

Таблица 355 – Регистр управления 5 передатчиком CONTROL5

| Номер | 31…24 | 23…16 | 15…8 | 7…0 |
|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W |
| Сброс | 3 | 3 | 3 | 3 |
| | T_PAUSE4 | T_PAUSE3 | T_PAUSE2 | T_PAUSE1 |

Таблица 356 – Описание бит регистра CONTROL5

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…24 | T_PAUSE4 | Значение паузы (T_PAUSE4+1) между сообщениями для канала передачи 1 в периодах скорости |
| 23…16 | T_PAUSE3 | Значение паузы (T_PAUSE3+1) между сообщениями для канала передачи 1 в периодах скорости |
| 15…8 | T_PAUSE2 | Значение паузы (T_PAUSE2+1) между сообщениями для канала передачи 1 в периодах скорости |
| 7…0 | T_PAUSE1 | Значение паузы (T_PAUSE1+1) между сообщениями для канала передачи 1 в периодах скорости |

### 24.3.22 STATUS

Таблица 357 – Регистр состояния передатчиков STATUS

| Номер | 31…14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| | - | HFT4 | FFT4 | TX_R4 | HFT3 | FFT3 | TX_R3 |

| Номер | 7…6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| | - | HFT2 | FFT2 | TX_R2 | HFT1 | FFT1 | TX_R1 |

Таблица 358 – Описание бит регистра STATUS

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…14 | - | Зарезервировано |
| 13 | HFT4 | Флаг наполненности FIFO канала 4:<br>1 – FIFO наполнено до половины;<br>0 – FIFO не наполнено до половины |
| 12 | FFT4 | Флаг полноты FIFO канала 4:<br>1 – FIFO полно;<br>0 – FIFO не полно |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 11 | TX_R4 | Флаг наличия данных в FIFO канала 4:<br>1 – FIFO пусто;<br>0 – FIFO содержит данные |
| 10 | HFT3 | Флаг наполненности FIFO канала 3:<br>1 – FIFO наполнено до половины;<br>0 – FIFO не наполнено до половины |
| 9 | FFT3 | Флаг полноты FIFO канала 3:<br>1 – FIFO полно;<br>0 – FIFO не полно |
| 8 | TX_R3 | Флаг наличия данных в FIFO канала 3:<br>1 – FIFO пусто;<br>0 – FIFO содержит данные |
| 7…6 | - | Зарезервировано |
| 5 | HFT2 | Флаг наполненности FIFO канала 2:<br>1 – FIFO наполнено до половины;<br>0 – FIFO не наполнено до половины |
| 4 | FFT2 | Флаг полноты FIFO канала 2:<br>1 – FIFO полно;<br>0 – FIFO не полно |
| 3 | TX_R2 | Флаг наличия данных в FIFO канала 2:<br>1 – FIFO пусто;<br>0 – FIFO содержит данные |
| 2 | HFT1 | Флаг наполненности FIFO канала 1:<br>1 – FIFO наполнено до половины;<br>0 – FIFO не наполнено до половины |
| 1 | FFT1 | Флаг полноты FIFO канала 1:<br>1 – FIFO полно;<br>0 – FIFO не полно |
| 0 | TX_R1 | Флаг наличия данных в FIFO канала 1:<br>1 – FIFO пусто;<br>0 – FIFO содержит данные |

### 24.3.23 DATA1_T

FIFO передаваемых данных канала 1.

FIFO может содержать данные объемом 256×32 для передачи по каналу 1. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R1, HFT1, FFT1.

### 24.3.24 DATA2_T

FIFO передаваемых данных канала 2.

FIFO может содержать данные объемом 64×32 для передачи по каналу 2. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R2, HFT2, FFT2.

### 24.3.25 DATA3_T

FIFO передаваемых данных канала 3.

FIFO может содержать данные объемом 64×32 для передачи по каналу 3. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R3, HFT3, FFT3.

### 24.3.26 DATA4_T

FIFO передаваемых данных канала 4.

FIFO может содержать данные объемом 64×32 для передачи по каналу 4. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R4, HFT4, FFT4.

# 25 Контроллер SSP

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

- интерфейс SPI фирмы Motorola;
- интерфейс SSI фирмы Texas Instruments;
- интерфейс Microwire фирмы National Semiconductor.

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

- преобразование данных, размещенных во внутреннем буфере FIFO передатчика (восемь 16-разрядных ячеек данных) из параллельного в последовательный формат;
- преобразование данных из последовательного в параллельный формат и их запись в аналогичный буфер FIFO приемника (восемь 16-разрядный ячеек данных).

Модуль формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов FIFO приемника и передатчика;
- переполнение буфера FIFO приемника;
- наличие данных в буфере FIFO приемника по истечении времени таймаута.

Основные сведения о модуле представлены в следующих разделах:

- характеристики интерфейса SPI;
- характеристики интерфейса Microwire;
- характеристики интерфейса SSI.

## 25.1 Основные характеристики модуля SSP

- может функционировать как в ведущем, так и в ведомом режиме;
- программное управление скоростью обмена;
- состоит из независимых буферов приема и передачи (восемь ячеек по 16 бит) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);
- программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;
- программируемая длительность информационного кадра от 4 до 16 бит;
- независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по переполнению буфера приемника;
- доступна возможность тестирования по шлейфу, соединяющему вход с выходом;
- поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена на рисунке 96.

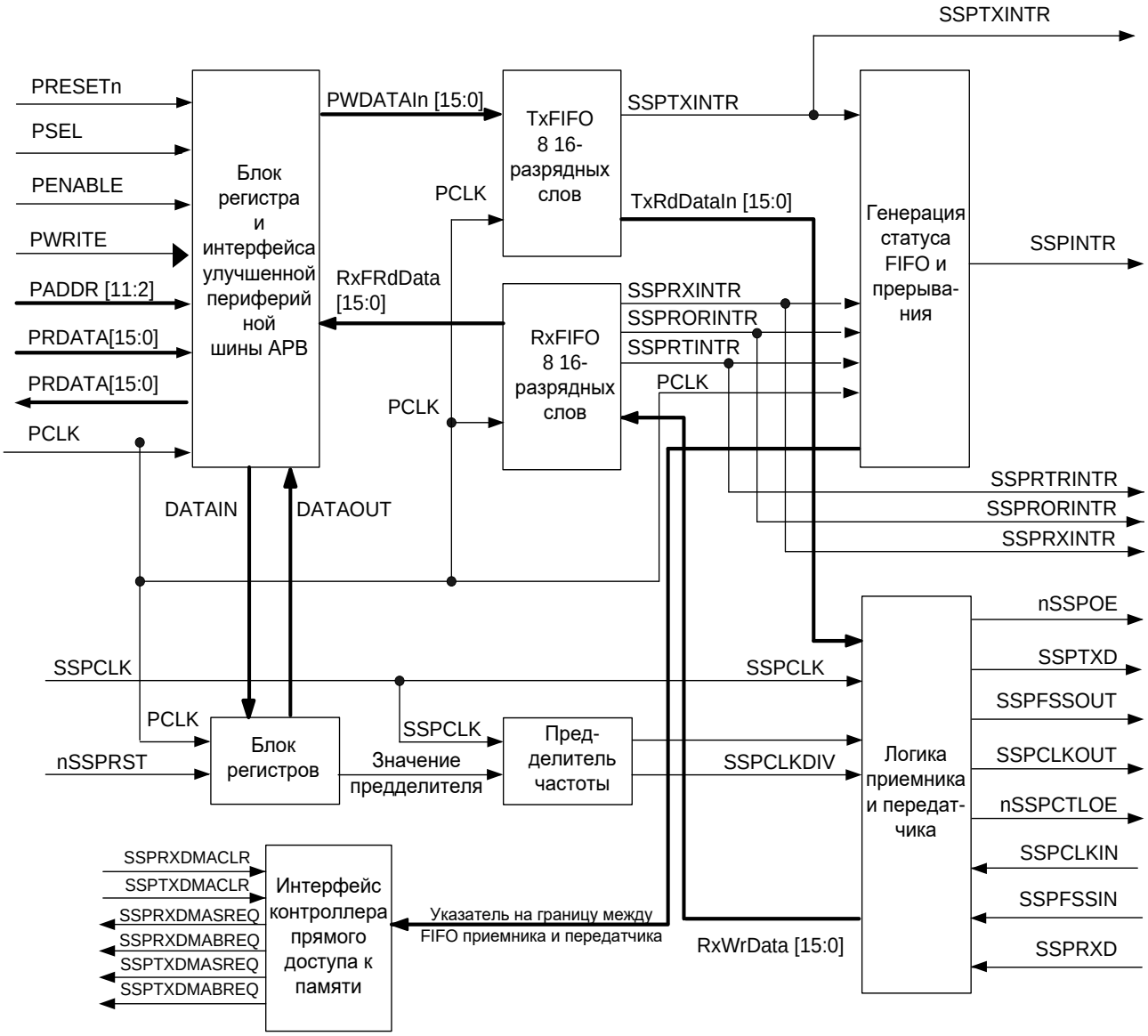



Рисунок 96 – Структурная схема модуля SSP

## 25.2 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

- режим функционирования периферийного устройства – ведущее или ведомое;
- разрешение или запрещение функционирования;
- формат информационного кадра;
- скорость передачи данных;
- фаза и полярность тактового сигнала;
- размер блока данных – от 4 до 16 бит;
- маскирование прерываний.

## 25.3 Характеристики интерфейса SPI

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:

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

## 25.4 Характеристики интерфейса Microwire

Интерфейс Microwire фирмы National Semiconductor обеспечивает:

– полудуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

## 25.5 Характеристики интерфейса SSI

Интерфейс SSI фирмы Texas Instruments обеспечивает:

– полнодуплексный обмен данными по четырехпроводной линии;

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

## 25.6 Общий обзор модуля SSP

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

Модуль выполняет следующие функции:

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

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

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

– прием и передача данных буферизуются с помощью буферов FIFO, обеспечивающих хранение до восьми слов данных шириной 16 бит независимо для режимов приема и передачи.

Последовательные данные передаются по линии SSP_TXD и принимаются с линии SSP_RXD.

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

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

Модуль формирует четыре независимо маскируемых прерывания:

– SSPTXINTR – запрос на обслуживание буфера передатчика;

– SSPRXINTR – запрос на обслуживание буфера приемника;

– SSPRORINTR – переполнение приемного буфера FIFO;

– SSPRTINTR – таймаут ожидания чтения данных из приемного FIFO.

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

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).

### 25.6.1 Блок формирования тактового сигнала

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

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

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

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

### 25.6.2 Буфер FIFO передатчика

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

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

### 25.6.3 Буфер FIFO приемника

Буфер приемника имеет ширину 16 бит, глубину восемь слов, схему организации доступа типа FIFO – «первый вошел, первый вышел». Принятые от периферийного устройства данные сохраняются в этом буфере блоком приема данных в до тех пор, пока не будут считаны центральным процессором.

### 25.6.4 Блок приема и передачи данных

Режим ведущего устройства

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

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

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

Режим ведомого устройства

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

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

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

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

### 25.6.5 Блок формирования прерываний

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

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

### 25.6.6 Интерфейс прямого доступа к памяти

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

### 25.6.7 Конфигурирование приемопередатчика

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

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

### 25.6.8 Разрешение работы приемопередатчика

Разрешение осуществляется путем установки бита SSE регистра управления CR1. Буфер FIFO передатчика может быть либо проинициализирован путем записи в него до восьми 16-разрядных слов заблаговременно перед установкой этого бита, либо может заполняться передаваемыми данными в процедуре обслуживания прерывания.

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

### 25.6.9 Соотношения между тактовыми сигналами

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

FSSPCLK <= FPCLK.

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

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

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

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

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

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

FSSPCLK(min) => 2 × FSSPCLKOUT(max) [for master mode],
FSSPCLK(min) => 12 × FSSPCLKIN(max) [for slave mode].

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

FSSPCLK(max) <= 254 × 256 × FSSPCLKOUT(min) [for master mode],
FSSPCLK(max) <= 254 × 256 × FSSPCLKIN(min) [for slave mode].

### 25.6.10 Программирование регистра управления CR0

Регистр CR0 предназначен для:

– установки скорости информационного обмена;
– выбора одного из трех протоколов обмена данными;
– выбора размера слова данных.

Скорость информационного обмена зависит от частоты внешнего тактового сигнала SSPCLK и коэффициента деления блока формирования тактового сигнала. Последний задается совместно значением поля SCR (Serial Clock Rate – скорость информационного обмена) регистра SSPCR0 и значением поля CPSDVSR (clock prescale divisor value – коэффициент деления тактового сигнала) регистра SSPCPSR.

Формат информационного кадра задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS регистра SSPCR0.

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

### 25.6.11 Программирование регистра управления CR1

Регистр SSPCR1 предназначен для:

– выбора ведущего или ведомого режима функционирования приемопередатчика;
– включения режима проверки канала по шлейфу;
– разрешения или запрещения работы модуля.

Выбор ведущего режима осуществляется путем записи 0 в поле MS регистра SSPCR1 (это значение устанавливается после сброса автоматически).

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

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

### 25.6.12 Формирование тактового сигнала обмена данными

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

Частота выходного тактового сигнала обмена данными SSP_CLK определяется следующим соотношением

FSSPCLKOUT = FSSPCLK / (CPSDVR • (1+SCR)).

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

### 25.6.13 Формат информационного кадра

Каждый информационный кадр содержит в зависимости от запрограммированного значения от 4 до 16 бит данных. Передача данных начинается со старшего значащего разряда. Можно выбрать три базовых структуры построения кадра:

- SSI фирмы Texas Instruments;
- SPI фирмы Motorola;
- Microwire фирмы National Semiconductor.

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

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

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

В отличие от полнодуплексных режимов передачи данных SSI и SPI, режим Microwire фирмы National Semiconductor использует специальный способ обмена данными между ведущим и ведомым устройством, функционирующий в режиме полудуплекса. В указанном режиме на внешнее ведомое устройство перед началом передачи информационного кадра посылается специальная восьмибитная управляющая последовательность. В течение всего времени передачи этой последовательности приемник не обрабатывает каких-либо входных данных. После того как сигнал передан и

декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом общая длительность информационного кадра составляет от 13 до 25 бит.

### 25.6.14 Формат синхронного обмена SSI фирмы Texas Instruments

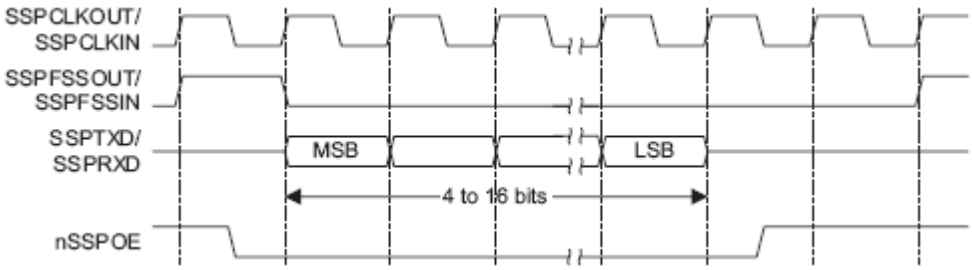



Рисунок 97 – Формат синхронного обмена протокола SSI (единичный обмен)

В данном режиме при неактивном приемопередатчике SSP сигналы SSP_CLK и SSP_FSS переводятся в низкий логический уровень, а линия передачи данных SSP_TXD поддерживается в третьем состоянии.

После появления хотя бы одного элемента в буфере FIFO передатчика сигнал SSP_FSS переводится в высокий логический уровень на время, соответствующее одному периоду сигнала SSP_CLK. При этом значение из буфера FIFO переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSP_CLK старший значащий разряд информационного кадра (4 – 16 бит данных) выдается на выход линии SSP_TXD и т.д.

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

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments представлены на рисунках: рисунок 97 – передача единичного информационно кадра) и рисунок 98 – передача последовательности кадров.

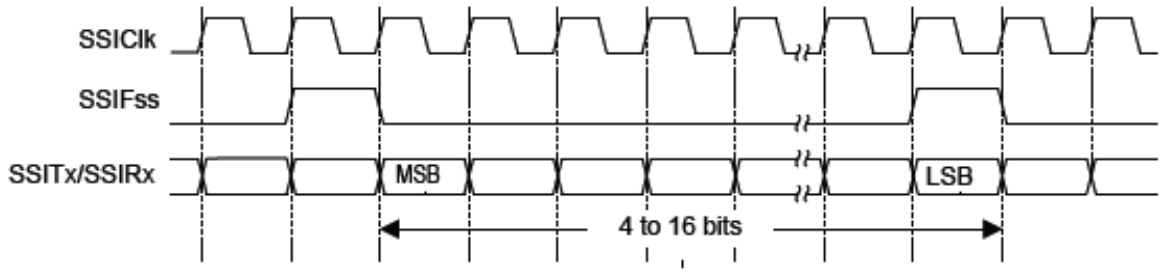



Рисунок 98 – Формат синхронного обмена протокола SSI (непрерывный обмен)

### 25.6.15 Формат синхронного обмена SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSP_FSS выполняет функцию выбора ведомого устройства. Главной особенностью протокола SPI является возможность выбора состояния и фазы сигнала SSP_CLK в режиме ожидания (неактивном приемопередатчике) путем задания значений бит SPO и SPH регистра управления SSPSCR0.

***Выбор полярности тактового сигнала – бит SPO***

Если бит SPO равен 0, то в режиме ожидания линия SSP_CLK переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SSP_CLK переводится в высокий логический уровень.

***Выбор фазы тактового сигнала – бит SPH***

Значение бита SPH определяет фронт тактового сигнала, по которому осуществляется выборка данных и изменение состояния на выходе линии.

В случае если бит SPH установлен в 0, регистрация данных приемником осуществляется после первого обнаружения фронта тактового сигнала, в противном случае – после второго.

### 25.6.16 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 0, SPH = 0 показаны на рисунках: рисунок 99 – одиночный обмен и рисунок 100 – непрерывный обмен.

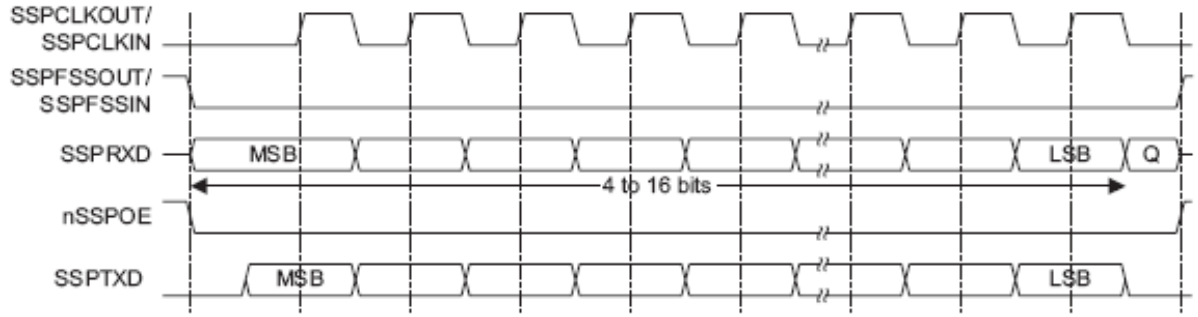



Рисунок 99 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (одиночный обмен)

Примечание – На рисунке 99 буквой Q обозначен сигнал с неопределенным уровнем.

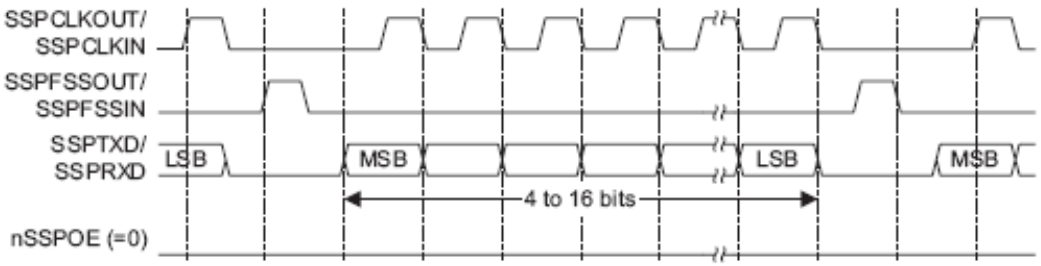



Рисунок 100 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

– сигнал SSP_CLK имеет низкий логический уровень;
– сигнал SSP_FSS имеет высокий логический уровень;
– сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Контакт передатчика SSPTXD переходит из высокоимпедансного в активное состояние.

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

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

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

В режиме непрерывной передачи данных на линии SSP_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP_FSS в высокий уровень по окончании передачи каждого кадра, разрешая, таким образом, запись новых данных. По окончании приема последнего бита блока данных линия SSP_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP_CLK.

### 25.6.17 Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=1 показаны на рисунке 101 (одиночный и непрерывный обмен).

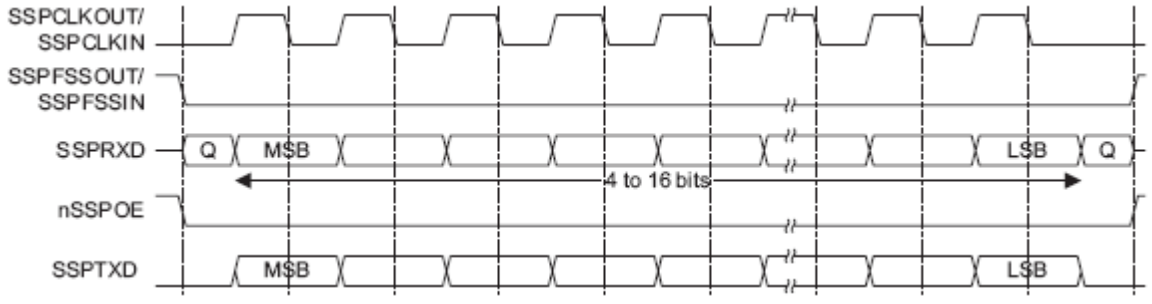



Рисунок 101 – Формат синхронного обмена протокола SPI, SPO=0, SPH=1

Примечание – На рисунке 101 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:
– сигнал SSP_CLK имеет низкий логический уровень;

– сигнал SSP_FSS имеет высокий логический уровень;
– сигнал SSP_TXD переводится в высокоимпедансное состояние.

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

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

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

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

В режиме непрерывной передачи данных линия SSP_FSS постоянно находится в низком логическом уровне, и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

### 25.6.18 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=0

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 1, SPH = 0 показаны на рисунках: рисунок 102 – одиночный обмен и рисунок 103 – непрерывный обмен.

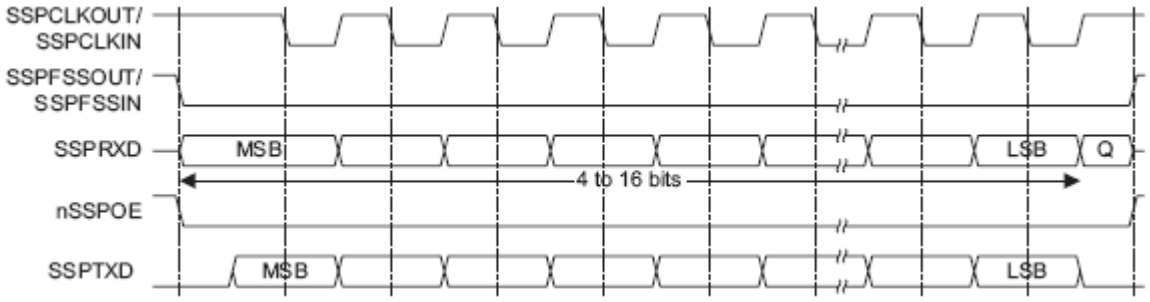



Рисунок 102 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (одиночный обмен)

Примечание – На рисунке 102 буквой Q обозначен сигнал с неопределенным уровнем.

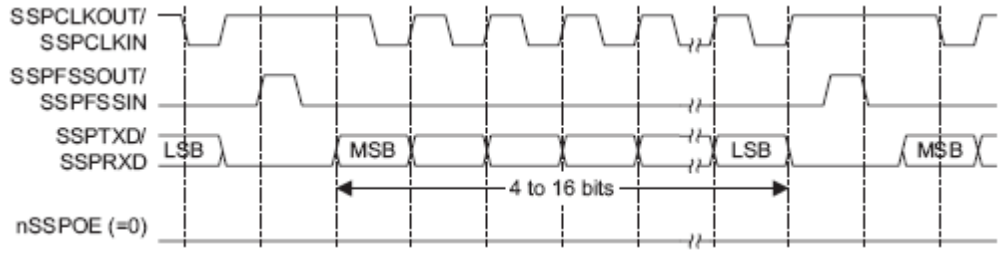



Рисунок 103 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

– сигнал SSP_CLK имеет высокий логический уровень;
– сигнал SSP_FSS имеет высокий логический уровень;
– сигнал SSP_TXD переводится в высокоимпедансное состояние.

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

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

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

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

В режиме непрерывной передачи данных на линии SSP_FSS должны формироваться импульсы высокого логического уровня между передачами каждого из слов данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSP_FSS в высокий уровень по окончании передачи каждого кадра, разрешая, таким образом, запись новых данных. По окончании приема последнего бита блока данных линия SSP_FSS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSP_CLK.

### 25.6.19 Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO = 1, SPH = 1 показаны на рисунке 104 (одиночный и непрерывный обмен).

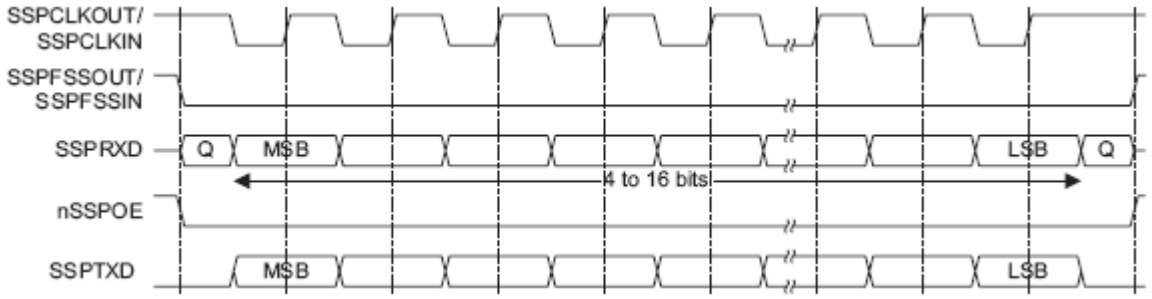



Рисунок 104 – Формат синхронного обмена протокола SPI, SPO=1, SPH=1

Примечание – На рисунке 103 буквой Q обозначен сигнал с неопределенным уровнем.

В данном режиме во время ожидания приемопередатчика:

– сигнал SSP_CLK имеет высокий логический уровень;
– сигнал SSP_FSS имеет высокий логический уровень;
– сигнал SSP_TXD переводится в высокоимпедансное состояние.

Если работа модуля разрешена и в буфере FIFO передатчика содержатся корректные данные, сигнал SSP_FSS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSP_RXD ведущего. Выходной контакт передатчика SSP_TXD переходит из высокоимпедансного в активное состояние.

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

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

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

В режиме непрерывной передачи данных линия SSP_FSS постоянно находится в низком логическом уровне и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

### 25.6.20 Формат синхронного обмена Microwire фирмы National Semiconductor

Временные диаграммы последовательного синхронного обмена в режиме Microwire показаны на рисунках: рисунок 105 – одиночный обмен и рисунок 106 – непрерывный обмен.

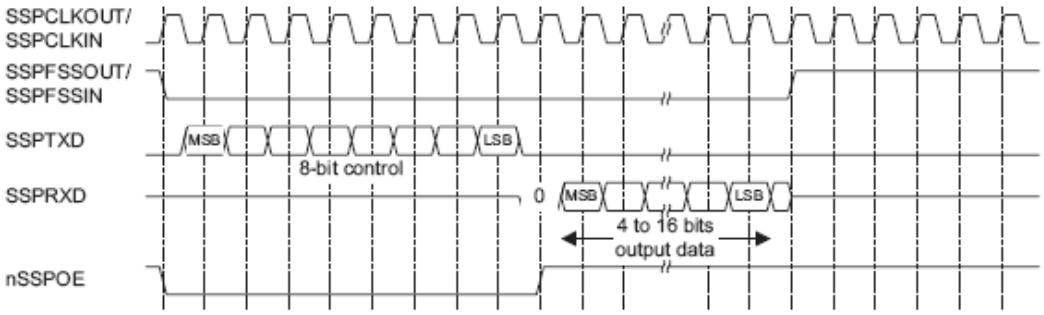



Рисунок 105 – Формат синхронного обмена протокола Microwire (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен в нем осуществляется в полудуплексном режиме, с использованием служебных последовательностей. Каждая информационный обмен начинается с передачи ведущим устройством специальной восьмибитной управляющей последовательности. В течение всего времени ее передачи приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего

устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

В данном режиме во время ожидания приемопередатчика:

– сигнал SSP_CLK имеет низкий логический уровень;
– сигнал SSP_FSS имеет высокий логический уровень;
– сигнал SSP_TXD переводится в высокоимпедансное состояние.

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

Внешнее ведомое устройство осуществляет прием бит данных по переднему фронту сигнала SSP_CLK. По окончании приема последнего бита управляющей последовательности она декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля SSP. Биты данных выдаются в линию SSP_RXD по заднему фронту сигнала SSP_CLK. Ведущее устройство, в свою очередь, регистрирует их по переднему фронту этого тактового сигнала. В случае одиночного информационного обмена по окончании приема последнего бита слова данных сигнал SSP_FSS переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных из регистра сдвига в буфер FIFO приемника.

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

Непрерывный обмен данными начинается и заканчивается так же, как и одиночный обмен. Однако линия SSP_FSS удерживается в низком логическом уровне в течение всего сеанса передачи данных. Управляющий байт следующего информационного кадра передается сразу же после приема младшего значащего разряда текущего кадра. Данные из сдвигового регистра передаются в буфер приемника после регистрации младшего разряда очередного слова по заднему фронту сигнала SSP_CLK.

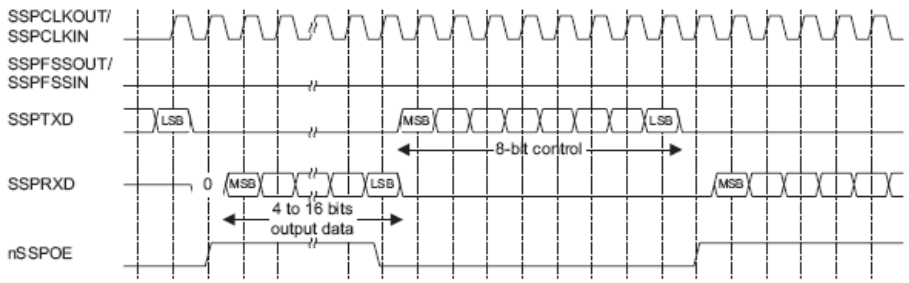



Рисунок 106 – Формат синхронного обмена протокола Microwire (непрерывный обмен)

***Требования к временным параметрам сигнала SSP_FSS относительно тактового сигнала SSP_CLK в режиме Microwire***

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

Данные требования иллюстрирует рисунок 107. По отношению к переднему фронту сигнала SSP_CLK, по которому осуществляется регистрация данных в приемнике ведомого модуля SSP, время установки сигнала SSP_FSS должно быть, как минимум в два раза больше периода SSP_CLK, на котором работает модуль. По отношению к предыдущему переднему фронту сигнала SSP_CLK должно обеспечиваться время удержания не менее одного периода этого тактового сигнала.

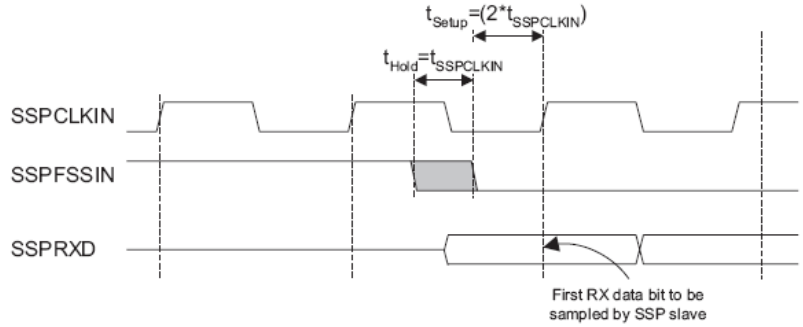



Рисунок 107 – Формат Microwire, требования к времени установки и удержания сигнала SSPFSSIN

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

На рисунках 108, 109, 110 показаны варианты подключения модуля SSP к периферийным устройствам, работающим в ведущем или ведомом режиме.

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

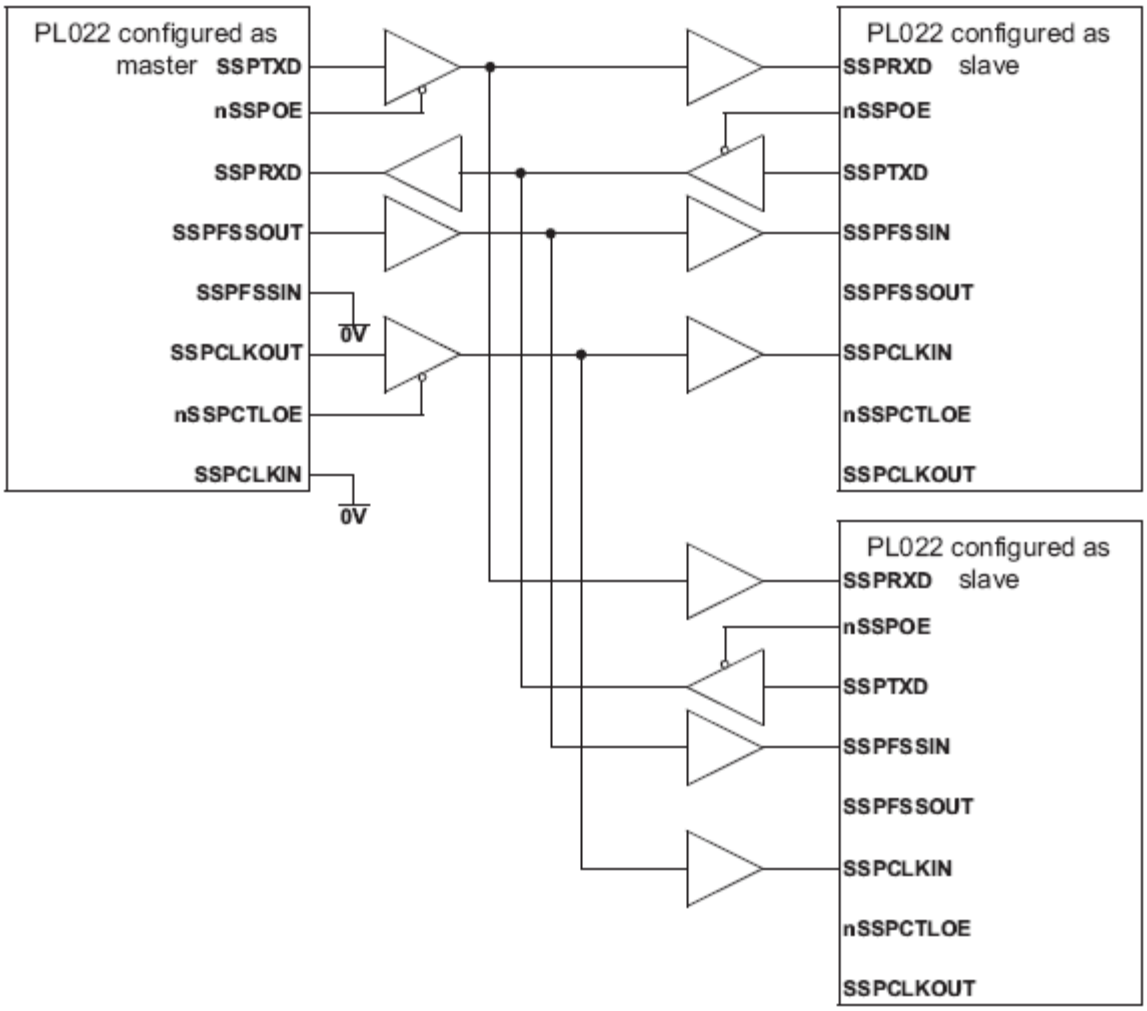



Рисунок 108 – Ведущее устройство SSP подключено к двум ведомым

Рисунок 108 показывает совместную работу трех модулей SSP, один из которых сконфигурирован в качестве ведущего, а два – в качестве ведомых устройств. Ведущее устройство способно передавать данные циркулярно в адрес двух ведомых по линии SSP_TXD.

Для ответной передачи данных один из ведомых модулей разрешает прохождение сигнала от своей линии SSP_TXD на вход SSP_RXD ведущего.

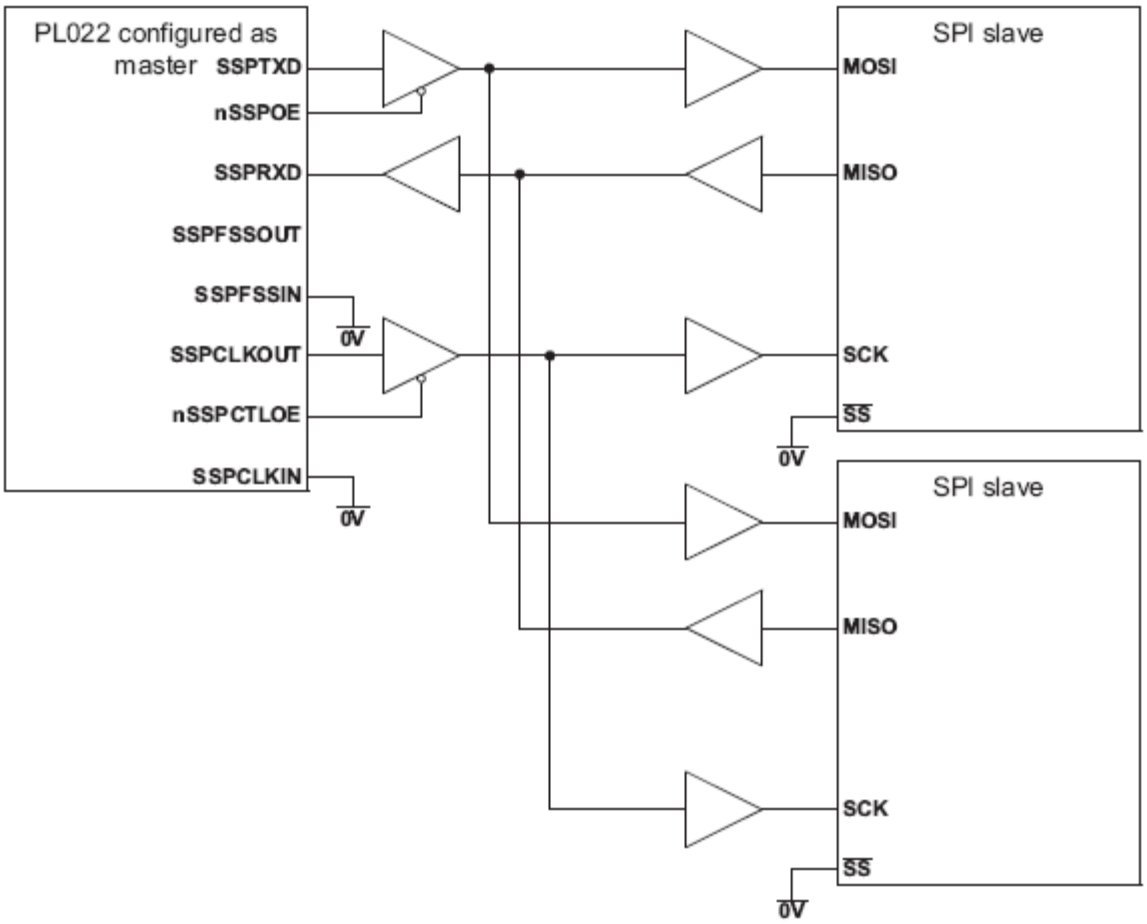



Рисунок 109 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим SPI

Рисунок 109 показывает подключение модуля SSP, сконфигурированного как ведущее устройство, к двум ведомым устройствам, поддерживающим протокол SPI фирмы Motorola. Внешние устройства сконфигурированы как ведомые путем установки в низкий логический уровень сигнала выбора ведомого устройства Slave Select (SS). Как и в предыдущем примере, ведущее устройство способно передавать данные в адрес ведомых циркулярно по линии SSP_TXD. Ответная передача данных на входную линию SSP_RXD ведущего устройства одновременно осуществляется только одним из ведомых по соответствующей линии MISO.

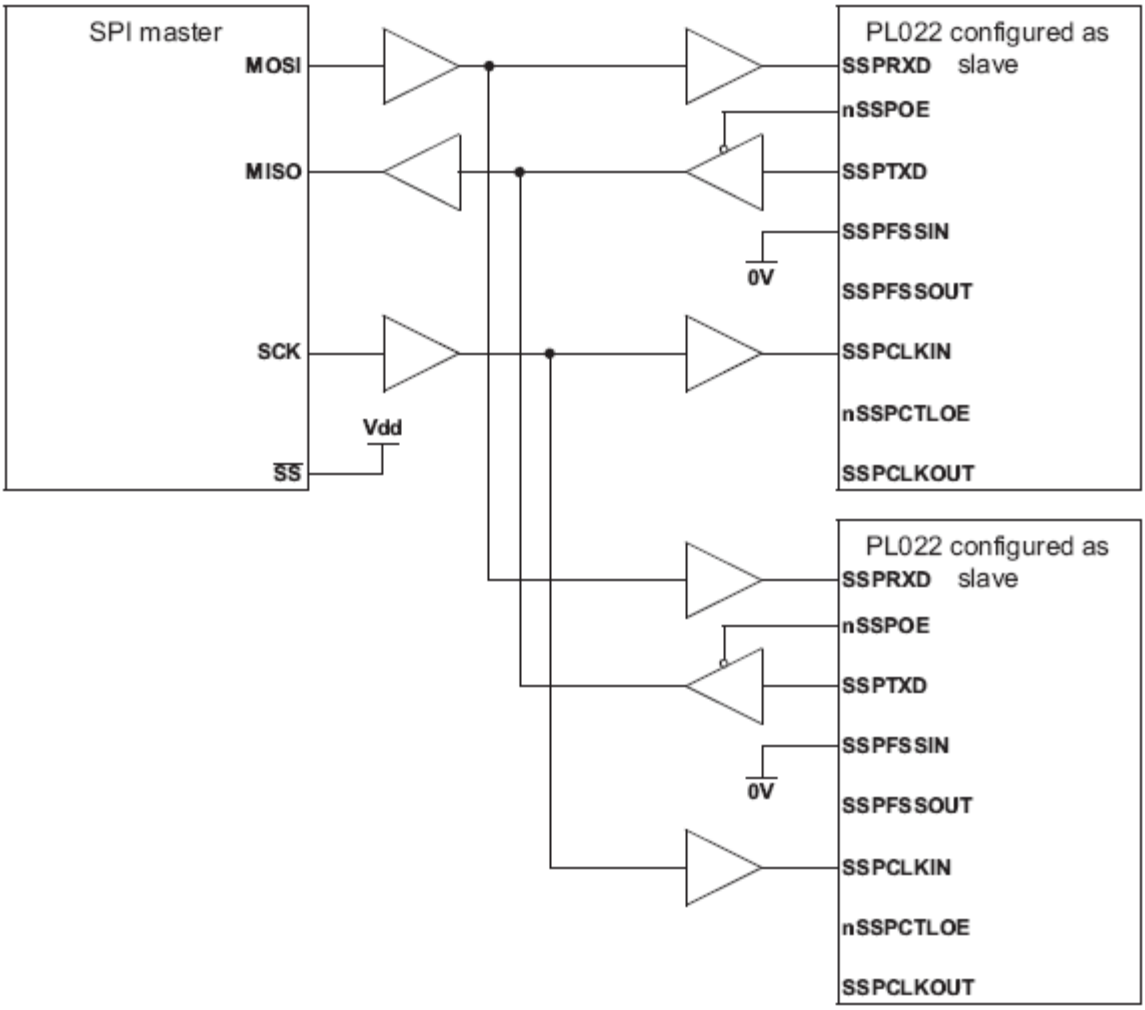



Рисунок 110 – Ведущее устройство, протокол SPI, подключено к двум ведомым модулям SSP

Рисунок 110 показывает ведущее устройство, поддерживающее протокол SPI фирмы Motorola, соединенное с двумя модулями SSP, сконфигурированными для работы в ведомом режиме. Линия Slave Select (SS) ведущего устройства в этом случае установлена в высокий логический уровень. Ведущее устройство осуществляет передачу данных по линии MOSI циркулярно в адрес двух ведомых модулей.

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

### 25.6.22 Интерфейс прямого доступа к памяти

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

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

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

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

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

Для передачи:

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

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

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

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

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

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

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

Таблица 359 показывает значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 359 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

| Пороговый уровень | Длина блока обмена данными | |
|---|---|---|
| | Буфер передатчика (количество незаполненных ячеек) | Буфер приемника (количество заполненных ячеек) |
| ½ | 4 | 4 |

Рисунок 111 показывает временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

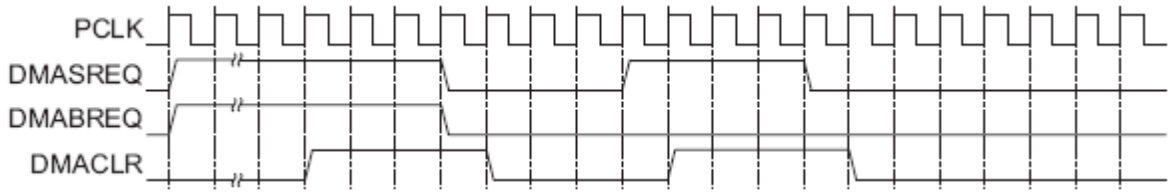



Рисунок 111 – Временные диаграммы обмена в режиме DMA

## 25.7 Программное управление модулем

### 25.7.1 Общая информация

В микросхеме реализовано два модуля SSP, базовые адреса каждого модуля указаны в таблице 360. Смещение каждого регистра относительно базового адреса постоянно. Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:

– адреса со смещениями в диапазоне +0x028 ... +0x07C и +0xFD0 ... +0xFDC зарезервированы для перспективных расширений возможностей модуля;

– адреса со смещениями в диапазоне +0x080 ... +0x088 зарезервированы для тестирования.

### 25.7.2 Описание регистров контроллера SSP

Данные о регистрах модуля SSP приведены в таблице 360.

Таблица 360 – Обобщенные данные о регистрах модуля SSP

| Базовый Адрес | Наименование | | | | Описание |
|---|---|---|---|---|---|
| 0x4004_0000 | SSP1 | | | | Регистры контроллера интерфейса SSP1 |
| 0x400A_0000 | SSP2 | | | | Регистры контроллера интерфейса SSP2 |
| 0x400F_8000 | SSP3 | | | | Регистры контроллера интерфейса SSP3 |
| Смещение | Наименование | Тип | Значение после сброса | Размер, бит | Описание |
| 0x000 | CR0 | RW | 0x0000 | 16 | Регистр управления 0 |
| 0x004 | CR1 | RW | 0x0 | 4 | Регистр управления 1 |
| 0x008 | DR | RW | 0x---- | 16 | Буфера FIFO приемника (чтение)<br>Буфер FIFO передатчика (запись) |
| 0x00C | SR | RO | 0x03 | 3 | Регистр состояния |
| 0x010 | CPSR | RW | 0x00 | 8 | Регистр делителя тактовой частоты |
| 0x014 | IMSC | RW | 0x0 | 4 | Регистр маски прерывания |
| 0x018 | RIS | RO | 0x8 | 4 | Регистр состояния прерываний без учета маскирования |
| 0x01C | MIS | RO | 0x0 | 4 | Регистр состояния прерываний с учетом маскирования |
| 0x020 | ICR | WO | 0x0 | 4 | Регистр сброса прерывания |
| 0x024 | DMACR | RW | 0x0 | 2 | Регистр управления прямым доступом к памяти |

Примечание – В поле «тип» указан вид доступа к регистру:
RW – чтение и запись;
RO – только чтение;
WO – только запись

#### 25.7.2.1 *Регистр управления 0 CR0*

Регистр CR0 содержит пять битовых полей, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 361.

Таблица 361 – Формат регистра CR0

| Бит | Наименование | Назначение |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…8 | SCR | Скорость последовательного обмена.<br>Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению<br>F_SSPCLK / (CPSDVR • (1 + SCR)),<br>где CPSDVR – четное число в диапазоне от 2 до 254 (см. подраздел «Регистр делителя тактовой частоты CPSR»);<br>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:<br>00 – протокол SPI фирмы Motorola;<br>01 – протокол SSI фирмы Texas Instruments;<br>10 – протокол Microwire фирмы National Semiconductor;<br>11 – резерв |
| 3…0 | DSS | Размер слова данных:<br>0000 – резерв;<br>0001 – резерв;<br>0010 – резерв;<br>0011 – 4 бита;<br>0100 – 5 бит;<br>0101 – 6 бит;<br>0110 – 7 бит;<br>0111 – 8 бит;<br>1000 – 9 бит;<br>1001 – 10 бит;<br>1010 – 11 бит;<br>1011 – 12 бит;<br>1100 – 13 бит;<br>1101 – 14 бит;<br>1110 – 15 бит;<br>1111 – 16 бит |

#### 25.7.2.2 *Регистр управления 1 CR1*

Регистр CR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 362.

Таблица 362 – Регистр CR1

| Биты | Наименование | Назначение |
|---|---|---|
| 15…4 | | Зарезервировано.<br>При чтении результат не определен.<br>При записи следует устанавливать в 0 |
| 3 | SOD | Запрет выходных линий в режиме ведомого устройства.<br>Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств.<br>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.<br>0 – управление линией SSP_TXD в ведомом режиме разрешена;<br>1 – управление линией SSP_TXD в ведомом режиме запрещена |
| 2 | MS | Выбор ведущего или ведомого режима работы:<br>0 – ведущий модуль (устанавливается по умолчанию);<br>1 – ведомый модуль |
| 1 | SSE | Разрешение работы приемопередатчика:<br>0 – работа запрещена;<br>1 – работа разрешена |
| 0 | LBM | Тестирование по шлейфу:<br>0 – нормальный режим работы приемопередатчика;<br>1 – выход регистра сдвига передатчика соединен со входом регистра сдвига приемника |

#### 25.7.2.3 *Регистр данных DR*

Регистр SSPDR имеет разрядность 16 бит и предназначен для чтения принятых и записи передаваемых данных.

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

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

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

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

Назначение разрядов регистра SSPDR описано в таблице 363.

Таблица 363 – Формат регистра DR

| Бит | Наименование | Назначение |
|---|---|---|
| 15…0 | DATA | Принимаемые данные (чтение).<br>Передаваемые данные (запись).<br>В случае если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника |

#### 25.7.2.4 *Регистр состояния SR*

Регистр состояния доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика, и занятости модуля SSP.

Таблица 364 показывает назначение бит регистра SSPSR.

Таблица 364 – Регистр SR

| Биты | Наименование | Назначение |
|---|---|---|
| 15…5 | - | Зарезервировано.<br>При чтении результат не определен.<br>При записи следует заполнить нулями. |
| 4 | BSY | Флаг занятости модуля:<br>0 – модуль SSP неактивен;<br>1 – модуль SSP в настоящее время передает и/или принимает данные, либо буфер FIFO передатчика не пуст. |
| 3 | RFF | Буфер FIFO приемника заполнен:<br>0 – не заполнен;<br>1 – заполнен |
| 2 | RNE | Буфер FIFO приемника не пуст:<br>0 – пуст;<br>1 – не пуст |
| 1 | TNF | Буфер FIFO передатчика не заполнен:<br>0 – заполнен;<br>1 – не заполнен |
| 0 | TFE | Буфер FIFO передатчика пуст:<br>0 – не пуст;<br>1 – пуст |

#### 25.7.2.5 *Регистр делителя тактовой частоты CPSR*

Регистр SSPCPSR используется для установки параметров делителя тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль. Если записать в регистр SSPCPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Назначение бит регистра SSPCPSR представлено в таблице 365.

Таблица 365 – Регистр CPSR

| Биты | Наименование | Назначение |
|---|---|---|
| 15…8 | - | Зарезервировано.<br>При чтении результат не определен.<br>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 |

#### 25.7.2.6 *Регистр установки и сброса маски прерывания IMSC*

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание. При этом запись 1 в разряд разрешает соответствующее прерывание, запись 0 – запрещает.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение битов регистра IMSC показано в таблице 366.

Таблица 366 – Регистр IMSC

| Биты | Наименование | Назначение |
|---|---|---|
| 15…4 | - | Зарезервировано.<br>При чтении выдаются нули.<br>При записи следует заполнить нулями. |
| 3 | TXIM | Маска прерывания по заполнению наполовину и менее буфера FIFO передатчика:<br>1 – не маскирована;<br>0 – маскирована |
| 2 | RXIM | Маска прерывания по заполнению наполовину и более буфера FIFO приемника:<br>1 – не маскирована;<br>0 – маскирована |
| 1 | RTIM | Маска прерывания по таймауту приемника (буфер FIFO приемника не пуст и не было попыток его чтения в течение времени таймаута):<br>1 – не маскирована;<br>0 – маскирована |
| 0 | RORIM | Маска прерывания по переполнению буфера приемника:<br>1 – не маскирована;<br>0 – маскирована |

#### 25.7.2.7 *Регистр состояния прерываний RIS*

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре RIS представлено в таблице 367.

Таблица 367 – Регистр RIS

| Биты | Наименование | Назначение |
|---|---|---|
| 15…4 | - | Зарезервировано.<br>При чтении выдаются нули |
| 3 | TXRIS | Состояние до маскирования прерывания SSPTXINTR:<br>1 – буфер FIFO передатчика заполнен наполовину или менее;<br>0 – буфер FIFO передатчика заполнен более чем наполовину |
| 2 | RXRIS | Состояние до маскирования прерывания SSPRXINTR:<br>1 – буфер FIFO приемника заполнен наполовину или более;<br>0 – буфер FIFO приемника заполнен менее чем наполовину |
| 1 | RTRIS | Состояние до маскирования прерывания SSPRTINTR:<br>1 – истекло время таймаута приемника;<br>0 – время таймаута приемника не истекло |
| 0 | RORRIS | Состояние до маскирования прерывания SSPRORINTR:<br>1 – возникло событие переполнения буфера приемника;<br>0 – событие переполнения буфера приемника не возникало |

#### 25.7.2.8 *Регистр маскированного состояния прерываний MIS*

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре SSPMIS представлено в таблице 368.

Таблица 368 – Регистр MIS

| Биты | Наименование | Назначение |
|---|---|---|
| 15…4 | - | Зарезервировано.<br>При чтении выдаются нули |
| 3 | TXMIS | Состояние маскированного прерывания SSPTXINTR:<br>1 – буфер FIFO передатчика заполнен наполовину или менее;<br>0 – буфер FIFO передатчика заполнен более чем наполовину |
| 2 | RXMIS | Состояние маскированного прерывания SSPRXINTR:<br>1 – буфер FIFO приемника заполнен наполовину или более;<br>0 – буфер FIFO приемника заполнен менее чем наполовину |

| Биты | Наименование | Назначение |
|---|---|---|
| 1 | RTMIS | Состояние маскированного прерывания SSPRTINTR:<br>1 – истекло время таймаута приемника;<br>0 – время таймаута приемника не истекло |
| 0 | RORMIS | Состояние маскированного прерывания SSPRORINTR:<br>1 – возникло событие переполнения буфера приемника;<br>0 – событие переполнения буфера приемника не возникало |

#### 25.7.2.9 *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 в любой из разрядов регистра 0 игнорируется.

Назначение бит в регистре SSPICR представлено в таблице 369.

Таблица 369 – Регистр ICR

| Биты | Наименование | Назначение |
|---|---|---|
| 15…2 | - | Зарезервировано.<br>При записи следует заполнить нулями |
| 1 | RTIC | Сброс прерывания SSPRTINTR |
| 0 | RORIC | Сброс прерывания SSPRORINTR |

#### 25.7.2.10 *Регистр управления прямым доступом к памяти DMACR*

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

Назначение бит регистра DMACR представлено таблице 370.

Таблица 370 – Регистр DMACR

| Биты | Наименование | Назначение |
|---|---|---|
| 15…2 | - | Зарезервировано.<br>При чтении выдаются нули.<br>При записи следует заполнить нулями |
| 1 | TXDMAE | Использование DMA при передаче:<br>1 – разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика;<br>0 – запрещено формирование запросов DMA для обслуживания буфера FIFO передатчика |
| 0 | RXDMAE | Использование DMA при приеме:<br>1 – разрешено формирование запросов DMA для обслуживания буфера FIFO приемника;<br>0 – запрещено формирование запросов DMA для обслуживания буфера FIFO приемника |

## 25.8 Прерывания

В модуле предусмотрено пять маскируемых линий запроса на прерывание, в том числе, четыре независимые линии запроса с активным высоким логическим уровнем, а также один общий сигнал, представляющий собой комбинацию независимых по схеме ИЛИ.

Сигналы запроса на прерывание:

- SSPRXINTR – запрос на обслуживание буфера FIFO приемника.
- SSPTXINTR – запрос на обслуживание буфера FIFO передатчика.
- SSPRORINTR – переполнение буфера FIFO приемника.
- SSPRTINTR – таймаут приемника.
- SSPINTR – логическое ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSPIMSC. Установка бита в «1» разрешает соответствующее прерывание, 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.

### 25.8.1 SSPRXINTR

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

### 25.8.2 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.

### 25.8.3 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.

### 25.8.4 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.

### 25.8.5 SSPINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR и SSPRORINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

# 26 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).

## 26.1 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 (16×12) и передачи (16×8) типа FIFO (First In First Out – первый вошел, первый вышел), что позволяет снизить интенсивность прерываний центрального процессора.

Программное отключение FIFO позволяет ограничить размер буфера одним словом.

Есть возможность программно настраивать скорость обмена данными, путем деления тактовой частоты опорного генератора в диапазоне (1×16 – 65535×16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц.

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

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

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

Также модуль обеспечивает обнаружение ложных стартовых бит.

Формирование и обнаружения сигнала разрыва линии.

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

Возможность организации аппаратного управления потоком данных.

Полностью программируемый асинхронный последовательный интерфейс имеет следующие характеристики:

– данные длиной 5, 6, 7 или 8 бит;
– формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение, либо не передается);
– формирование 1 или 2 стоповых бит;
– скорость передачи данных – от 0 до 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.

## 26.2 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 1,42 – 2,12 МГц для обеспечения возможности формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления (для ИК-обмена);

– режим аппаратного управления потоком данных.

## 26.3 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.

## 26.4 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 байтов независимо для режимов приема и 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). Данные функции выводятся на порты микросхемы в микросхемах с даты изготовления 1236.

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.

## 26.5 Описание функционирования блока UART

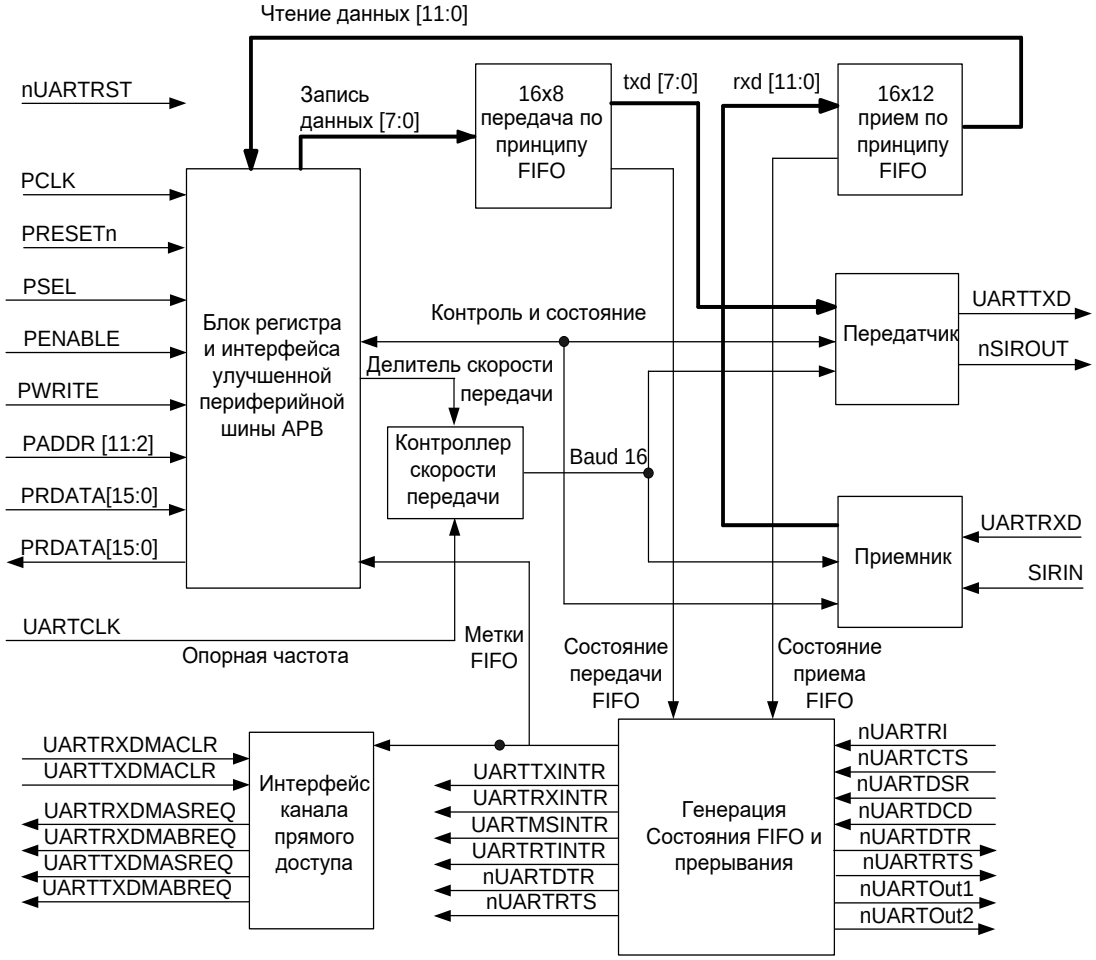



Рисунок 112 – Блок-схема универсального асинхронного приемопередатчика (UART)

### 26.5.1 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.

### 26.5.2 Буфер FIFO передатчика

Буфер передатчика имеет ширину 8 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.

### 26.5.3 Буфер FIFO приемника

Буфер приемника имеет ширину 12 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.

### 26.5.4 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.

### 26.5.5 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.

### 26.5.6 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 26.10 «Прерывания».

### 26.5.7 Интерфейс прямого доступа к памяти

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

### 26.5.8 Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов CPU_CLK и UART_CLK. Регистры синхронизации и логика квитирования постоянно находятся в активном состоянии. Это практически не отражается на характеристиках устройства и занимаемой площади. Синхронизация сигналов управления осуществляется в обоих направлениях потока данных, то есть как из области действия CPU_CLK в область действия UART_CLK, так и наоборот.

## 26.6 Описание функционирования ИК-кодека IrDA SIR

Структурная схема кодека представлена на рисунке 113.

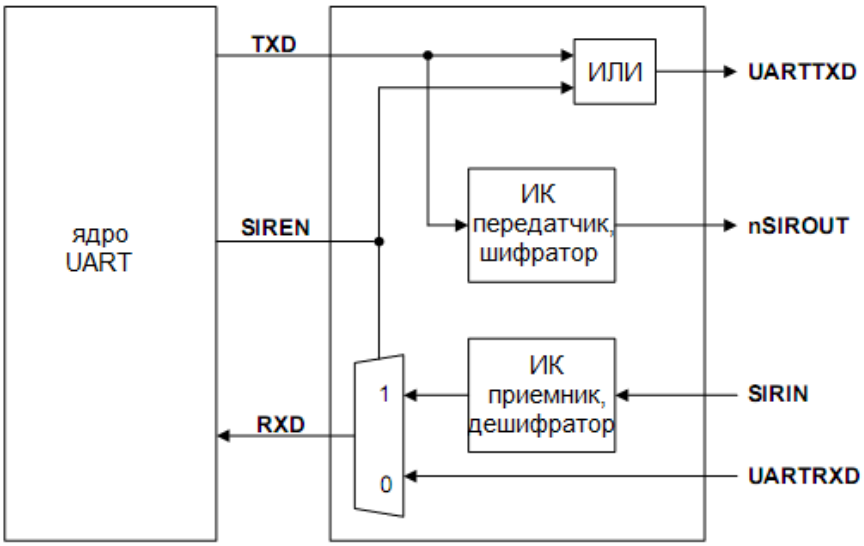



Рисунок 113 – Структурная схема кодека IrDA

### 26.6.1 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 «1» выход кодера остается в низком состоянии, при передаче логического «0» – 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.

### 26.6.2 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.

## 26.7 Описание работы UART

### 26.7.1 Сброс модуля

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса процессора. Значения регистров после сброса описаны в подразделе 26.11 «Программное управление модулем».

### 26.7.2 Тактовые сигналы

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

$F_{UART_CLK}(min) >= 16 \cdot baud_rate_max;$
$F_{UART_CLK}(max) <= 16 \cdot 65535 \cdot baud_rate_min.$

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота $F_{UART_CLK}$ должна находиться в интервале от 7,3728 МГц до 115,34 МГц.

Частота $F_{UART_CLK}$, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

### 26.7.3 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.

### 26.7.4 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, 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:

$$\text{Коэффициент деления} = \frac{F_{UART_CLK}}{16 \cdot \text{скорость передачи данных}} = IBRD + FBRD\ , \qquad (10)$$

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

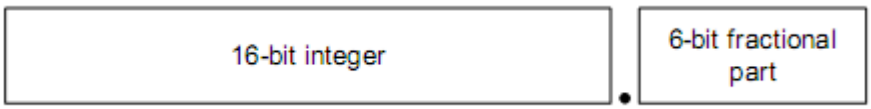



Рисунок 114 – Коэффициент деления

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:

$$M = integer(FBRD \cdot 2^n + 0{,}5), \qquad (11)$$

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.

### 26.7.5 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.

### 26.7.6 Биты ошибки

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся на позиции [10:8] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO, расположенный на позиции 11 слова данных.

Таблица 371 показывает назначение всех битов слова данных в FIFO буфере приемника.

### 26.7.7 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.

Таблица 371 – Назначение бит слова данных в FIFO-буфере приемника

| Бит буфера FIFO | Назначение |
|---|---|
| 11 | Признак переполнения буфера |
| 10 | Ошибка – разрыв линии |
| 09 | Ошибка проверки на четность |
| 08 | Ошибка формирования кадра |
| 07…00 | Принятые данные |

### 26.7.8 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.

Проверка по шлейфу

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

### 26.7.9 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:

В режиме IrDA уровень логического нуля передается на линию nSIROUT в виде импульса с высоким логическим уровнем и длительностью 3/16 от выбранного периода следования бит данных. Логическая единица при этом передается в виде постоянного низкого уровня сигнала. Сформированный выходной сигнал далее подается на передатчик ИК-сигнала, обеспечивая излучение светового импульса всякий раз при передаче нулевого бита. На приемной стороне световые импульсы воздействуют на базу фототранзистора ИК-приемника, который в результате формирует низкий логический уровень. Это, в свою очередь, обуславливает низкий уровень на входе SIRIN.

В режиме IrDA с пониженным энергопотреблением длительность передаваемых импульсов ИК-излучения устанавливается в три раза 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 нормальном режиме, так и в режиме пониженного энергопотребления:

– кодирование осуществляется на основе бит данных, сформированных асинхронным передатчиком модуля;

– в ходе приема данных декодированные биты далее обрабатываются блоком асинхронного приема.

В соответствии со спецификацией физического уровня протокола IrDA SIR, обмен данными должен осуществляться в режиме полудуплекса, при этом задержка между передачей и приемом данных должна составлять не менее 10 мс. Эта задержка должна формироваться программно. Необходимость ее введения обусловлена тем, что воздействие передающего ИК-светодиода на находящийся рядом ИК-приемник может привести к искажению принимаемого сигнала или даже ввести приемный тракт в состояние насыщения. Задержка между окончанием передачи и началом приема данных именуется латентность, или время установки (готовности) приемника.

Сигнал IrLPBaud16 формируется путем деления частоты сигнала UART_CLK в соответствии с коэффициентом деления, записанным в регистре ILPR.

Коэффициент деления вычисляется по формуле:

$$\text{Коэффициент деления} = \frac{F_{UART_CLK}}{IrLPBaud16}, \qquad (12)$$

где номинальное значение IrLPBaud16 составляет 1,8432 МГц.

Коэффициент деления должен быть выбран так, чтобы выполнялось соотношение

$$1{,}42\ \text{Мгц} < IrLPBaud16 < 2{,}12\ \text{МГц}.$$

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.

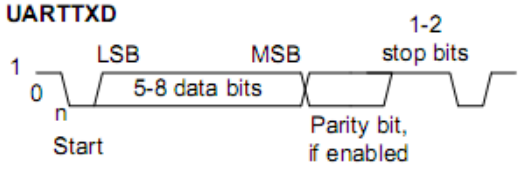



Рисунок 115 – Кадр передачи данных

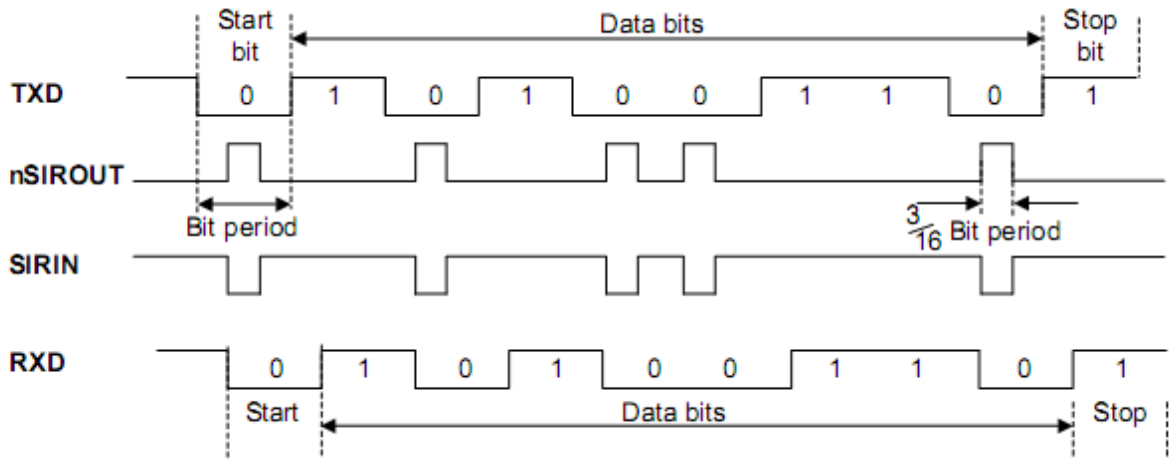



Рисунок 116 – Модуляция данных IrDA

## 26.8 Линии управления модемом

Модуль универсального асинхронного приемопередатчика может использоваться как в режиме оконечного оборудования (DTE), так и в режиме оборудования передачи данных (DCE).

Таблица 372 – 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 |

### 26.8.1 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 117.

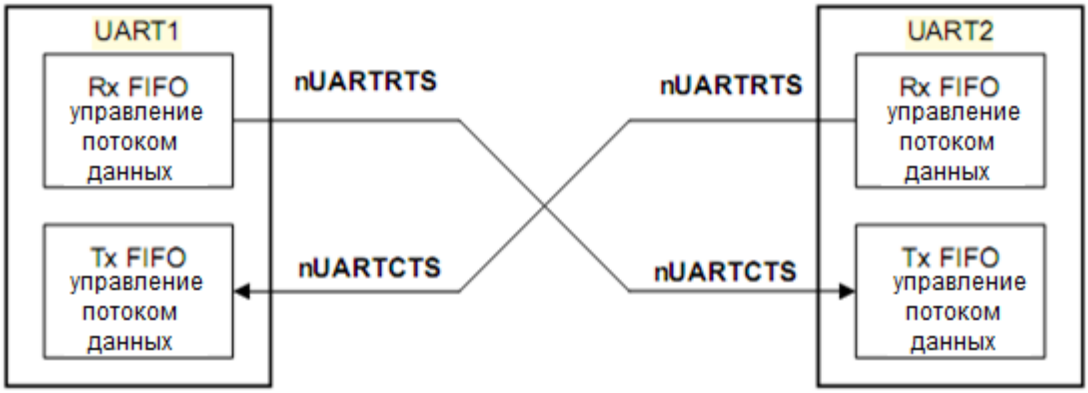



Рисунок 117 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

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 373 показаны необходимые установки для различных режимов управления потоком данных.

Таблица 373 – 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 | | |

### 26.8.2 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.

### 26.8.3 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.

## 26.9 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 12 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ.

UARTRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если заполнение буфера FIFO приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр 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 восьми бит. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит, по меньшей мере, одну свободную ячейку.

UARTTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если заполнение буфера FIFO передатчика ниже заданного порога. Порог программируется индивидуально для каждого буфера FIFO путем записи значения в регистр 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. Значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера показаны в таблице 374.

Таблица 374 – 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 118 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.

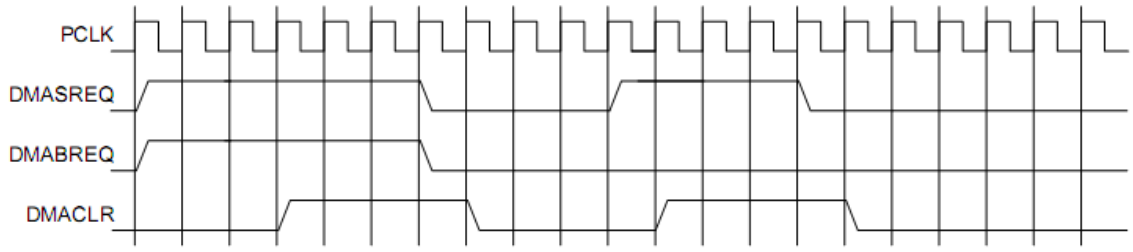



Рисунок 118 – Временные диаграммы одноэлементного и блочного запросов DMA

## 26.10 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;
- 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 и UARTEINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски IMSC. Установка бита в «1» разрешает соответствующее прерывание, в 0 – запрещает.

Доступность, как индивидуальных линий, так и общей линии запроса позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика UARTRXINTR и UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

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

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

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний RIS, либо из маскированного регистра прерываний MIS.

### 26.10.1 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.

### 26.10.2 UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

– буфер FIFO разрешен и его заполнение достигло заданного порогового значения. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения данных из буфера приемника до тех пор, пока его заполнение не станет меньше порога, либо после сброса прерывания;

– буфер FIFO запрещен (имеет размер один символ), принят один символ данных. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения одного байта данных, либо после сброса прерывания.

### 26.10.3 UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

– буфер FIFO разрешен и его заполнение меньше или равно заданному пороговому значению. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи данных в буфера передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания;

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

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

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

### 26.10.4 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.

### 26.10.5 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.

### 26.10.6 UARTINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

## 26.11 Программное управление модулем

**Общая информация**

Следующая информация применима ко всем регистрам контроллера:

– базовый адрес контроллера фиксирован. Смещение каждого регистра относительно базового адреса постоянно.

– не следует пытаться получить доступ к зарезервированным или неиспользуемым адресам. Это может привести к непредсказуемому поведению модуля.

– за исключением специально оговоренных в 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». Таблица 375 определяет режим доступа к регистру в соответствии с обозначениями:

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

## 26.12 Обобщенные данные о регистрах устройства

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в таблице 375.

Таблица 375 – Обобщенные данные о регистрах устройства

| Базовый Адрес | Название | | | | Описание |
|---|---|---|---|---|---|
| 0x40030000 | UART1 | | | | Регистры контроллера интерфейса UART1 |
| 0x40038000 | UART2 | | | | Регистры контроллера интерфейса UART2 |
| Смещение | Наимено-вание | Тип | Значение после сброса | Размер, бит | Описание |
| 0x000 | DR | RW | 0x--- | 12/8 | Регистр данных |
| 0x004 | RSR_ECR | RW | 0x0 | 4/0 | Регистра состояния приемника / Сброс ошибки приемника |
| 0x008-0x014 | | | | | Резерв |
| 0x018 | FR | RO | 0b-10010--- | 9 | Регистр флагов |
| 0x01C | | | | | Резерв |
| 0x020 | ILPR | RW | 0x00 | 8 | Регистр управления ИК-обменом в режиме пониженного энергопотребления |
| 0x024 | IBRD | RW | 0x0000 | 16 | Целая часть делителя скорости обмена данными |
| 0x028 | FBRD | RW | 0x00 | 6 | Дробная часть делителя скорости обмена данными |
| 0x02C | LCR_H | RW | 0x00 | 8 | Регистр управления линией |
| 0x030 | CR | RW | 0x0300 | 16 | Регистр управления |
| 0x034 | IFLS | RW | 0x12 | 6 | Регистр порога прерывания по запол-нению буфера FIFO |
| 0x038 | IMSC | RW | 0x000 | 11 | Регистр маски прерывания |
| 0x03C | RIS | RO | 0x00- | 11 | Регистр состояния прерываний |
| 0x040 | MIS | RO | 0x00- | 11 | Регистр состояния прерываний с маскированием |
| 0x044 | ICR | WO | - | 11 | Регистр сброса прерывания |
| 0x048 | DMACR | RW | 0x00 | 3 | Регистр управления ПДП |
| 0x080 | TCR | RW | 0x00 | 3 | Регистр управления тестированием |

### 26.12.1 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 376).

Таблица 376 – Формат регистра UARTDR

| Бит | Наименование | Назначение |
|---|---|---|
| 15…12 | - | Зарезервировано |
| 11 | OE | Переполнение буфера приемника. Бит устанавливается в «1» в случае, если на вход приемника поступают данные, в то время как буфер заполнен. Сбрасывается в «0» после того, как в буфере появится свободное место |
| 10 | BE | Разрыв линии. Устанавливается в «1» при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую «1» и последующего обнаружения корректного стартового бита |
| 9 | PE | Ошибка контроля четности. Устанавливается в «1» в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией LCR_H. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер |

| Бит | Наименование | Назначение |
|---|---|---|
| 8 | FE | Ошибка в структуре кадра. Устанавливается в «1» в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер |
| 7…0 | DATA | Принимаемые данные (чтение).<br>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 | | |

### 26.12.2 Регистр состояния приемника / сброса ошибки RSR_ECR

Состояние приемника также может быть считано из регистра RSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра относится к последнему символу, считанному из регистра данных DR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним, считанным из регистра DR, байтом данных).

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

Таблица 377 показывает назначение бит регистра RSR_ECR.

Таблица 377 – Регистр RSR_ECR

| Биты | Наименование | Назначение |
|---|---|---|
| 7…4 | - | Зарезервировано.<br>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 |

| Биты | Наименование | Назначение |
|---|---|---|
| | | 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 | | |

### 26.12.3 Регистр флагов FR

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в «», а биты TXFE и RXFE – в «1». Таблица 378 содержит информацию о назначении бит регистра.

Таблица 378 – Регистр FR

| Биты | Наименование | Назначение |
|---|---|---|
| 15…9 | | Резерв. Не модифицируйте. При чтении заполняются нулями |
| 8 | RI | Инверсия линии nUARTRI |
| 7 | TXFE | Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит устанавливается в «1», когда буферный регистр передатчика пуст. В противном случае он равен 1, если пуст буфер FIFO передатчика. Данный бит не дает никакой информации о наличии данных в регистре сдвига передатчика |
| 6 | RXFF | Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией LCR_H. Если буфер FIFO запрещен, бит устанавливается в «1», когда буферный регистр приемника занят. В противном случае он равен 1, если заполнен буфер FIFO приемника |
| 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 |

| Биты | Наименование | Назначение |
|---|---|---|
| | | 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 |

### 26.12.4 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 379.

Требуемое значение коэффициента деления для формирования сигнала IrLPBaud16 вычисляется по формуле

$$ILPDVSR = F_{UART_CLK} / F_{IrLPBaud16},$$

где номинальное значение частоты $F_{IrLPBaud16}$ составляет 1,8432 МГц.

Коэффициент деления должен быть установлен таким образом, чтобы выполнялось соотношение: 1,42 МГц < $F_{IrLPBaud16}$ < 2,12 МГц, что, в свою очередь, гарантирует формирование кодеком импульсов данных с длительностью 1,41 – 2,11 мкс (в три раза длиннее периода сигнала IrLPBaud16).

Таблица 379 – Регистр LPR

| Биты | Наименование | Назначение |
|---|---|---|
| 7…0 | ILPDVSR | Коэффициент деления частоты UART_CLK, для формирования тактового сигнала IrLPBaud16. После сброса устанавливается в «0».<br>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 | | |

### 26.12.5 Регистр целой части делителя скорости передачи данных IBRD

Назначение бит регистра представлено в таблице 380.

Таблица 380 – Регистр IBRD

| Биты | Наименование | Назначение |
|---|---|---|
| 15…0 | BAUDDIV_INT | Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в «0» |

### 26.12.6 Регистр дробной части делителя скорости передачи данных FBRD

Назначение бит регистра представлено в таблице 381.

Таблица 381 – Регистр BFRD

| Биты | Наименование | Назначение |
|---|---|---|
| 5…0 | BAUDDIV_FRAC | Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в «0» |

Коэффициент деления вычисляется по формуле

$$BAUDDIV = \frac{F_{UART_CLK}}{16 \cdot Baud_rate}, \quad (13)$$

где $F_{UART_CLK}$ – тактовая частота контроллера UART, Baud_rate – требуемая скорость передачи данных (в бит/с).

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUDDIV_INT и BAUDDIV_FRAC, соответственно.

Примечание – Изменение содержимого регистров IBRD и FBRD вступают в силу только после завершения передачи и приема текущего символа данных.

Минимальный допустимый коэффициент деления – 1, максимальный 65535 ($2^{16} – 1$). Таким образом, значение IBRD, равное 0 является недопустимым, при этом значение регистра FBRD игнорируется.

Аналогично, при IBRD равном 65535 (0xFFFF), значение FBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

Далее приведен пример вычисления коэффициента деления.

Пример. Вычисление коэффициента деления

Пусть требуемая скорость передачи данных составляет 230400 бит/с, частота тактового сигнала $F_{UART_CLK}$ = 4 МГц. Тогда:

Коэффициент деления = $(4 \cdot 10^6)/(16 \cdot 230400)$ = 1,085.

Таким образом, BRDI = 1, BRDF = 0,085.

Следовательно, значение, записываемое в регистр BFRD, равно
m = integer((0,085 • 64) + 0.5) = 5.
Реальное значение коэффициента деления = 1+5/64 = 1,078.
Реальная скорость передачи данных = $(4 \cdot 10^6) / (16 \cdot 1.078)$ = 231911 бит/с.
Ошибка установки скорости = (231911 - 230400) / 230400 • 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 382 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 382 – 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 |

Таблица 383 содержит значения коэффициента деления для типичных скоростей передачи данных при частоте $F_{UART_CLK}$ = 4 МГц.

Таблица 383 – 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 |
| 0x068 | 0x0B | 2400 | 2400 | ~0 |
| 0x8E0 | 0x2F | 110 | 110 | ~0 |

### 26.12.7 Регистр управления линией LCR_H

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

Назначение разрядов регистра описано в таблице 384.

Таблица 384 – Регистр LCR_H

| Биты | Наименование | Назначение |
|---|---|---|
| 15…8 | | Зарезервировано. Не модифицируйте.<br>При чтении выдаются нули |
| 7 | SPS | Передача бита четности с фиксированным значением:<br>0 – запрещена;<br>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).<br>Значение бита SPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено (Таблица 385) |
| 6, 5 | WLEN | Длина слова – количество передаваемых или принимаемых информационных бит в кадре:<br>0b11 – 8 бит, 0b10 – 7 бит, 0b01 – 6 бит, 0b00 – 5 бит |
| 4 | FEN | Разрешение работы буфера FIFO приемника и передатчика:<br>0 – запрещено;<br>1 – разрешено |
| 3 | STP2 | Режим передачи двух стоповых бит:<br>0 – один стоповый бит;<br>1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре |
| 2 | EPS | Четность/нечетность:<br>0 – бит четности дополняет количество единиц в информационной части кадра до нечетного;<br>1 – до четного числа. Значение бита EPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещено (см. таблицу 385) |
| 1 | PEN | Разрешение проверки четности:<br>0 – кадр не содержит бита четности;<br>1 – бит четности передается в кадре и проверяется при приеме данных (см. таблицу 385) |
| 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 30-разрядный регистр LCR, который обновляется по стробу, формируемому при записи в LCR_H. Таким образом, для того, чтобы изменение параметров коэффициента деления частоты обмена данными вступило в силу, после их изменения значения регистров IBRD и/или FBRD необходимо осуществить запись данных в регистр LCR_H.

Таблица 385 содержит данные об истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией LCR_H.

Таблица 385 – 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.

### 26.12.8 Регистр управления CR

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

Назначение разрядов регистра управления показано в таблице 386.

Таблица 386 – Регистр управления CR

| Биты | Наименование | Назначение |
|---|---|---|
| 15 | CTSEn | Разрешение управления потоком данных по CTS:<br>1 – разрешено, данные передаются в линию только при активном значении сигнала nUARTCTS;<br>0 – запрещено |
| 14 | RTSEn | Разрешение управления потоком данных по RTS:<br>1 – разрешено, запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере FIFO приемника;<br>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).<br>1 – сигнал разрешен;<br>0 – сигнал запрещен |

| Биты | Наименование | Назначение |
|---|---|---|
| 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).<br>1 – сигнал разрешен;<br>0 – сигнал запрещен |
| 11 | RTS | Инверсия сигнала на линии состояния модема nUARTRTS:<br>1 – сигнал разрешен;<br>0 – сигнал запрещен |
| 10 | DTR | Инверсия сигнала на линии состояния модема nUARTDTR:<br>1 – сигнал разрешен;<br>0 – сигнал запрещен |
| 9 | RXE | Разрешение приема:<br>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;<br>0 – работа приемника запрещена |
| 8 | TXE | Разрешение передачи:<br>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;<br>0 – работа передатчика запрещена |
| 7 | LBE | Режим тестирования по шлейфу:<br>1 – шлейф разрешен;<br>0 – запрещен.<br>В режиме разрешенного шлейфа:<br>– 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»;<br>– если бит SIRTEST=0, то выходная линия передатчика UART_TXDx коммутируется на вход приемника UART_RXDx.<br>Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии.<br>После сброса бит устанавливается в «0» |
| 6…3 | | Резерв. Не модифицируйте. При чтении выдаются нули |

| Биты | Наименование | Назначение |
|---|---|---|
| 2 | SIRLP | Выбор режима ИК-обмена с пониженным энергопотреблением:<br>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;<br>0 – длительность импульсов данных равна 3/16 длительности передачи бита |
| 1 | SIREN | Разрешение работы кодека ИК-передачи данных IrDA SIR:<br>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.<br>В случае если UARTEN = 0 значение бита не играет роли.<br>0 – запрещен. Сигнал nSIROUT находится в низком состоянии, данные на входе SIRIN не обрабатываются |
| 0 | UARTEN | Разрешение работы приемопередатчика:<br>0 – работа запрещена. Перед остановкой завершается прием и/или передача обрабатываемого в текущий момент символа;<br>1 – работа разрешена. Производится обмен данными либо по линиям асинхронного обмена, либо по линиям ИК-обмена SIR, в зависимости от состояния бита SIREN |

Примечания

1 Для того чтобы разрешить передачу данных, необходимо установить в «1» биты TXE и UARTEN. Аналогично, для разрешения приема данных необходимо установить в «1» биты RXE и UARTEN.

2 Рекомендуется следующая последовательность действий для программирования регистров управления:

– остановите работу приемопередатчика;
– дождаться окончания приема и/или передачи текущего символа данных;
– сбросьте буфер передатчика путем установки бита FEN регистра LCR_H в «0»;
– изменить настройки регистра CR;
– возобновите работу приемопередатчика

### 26.12.9 Регистр порога прерывания по заполнению буфера FIFO IFLS

Данный регистр используется для установки порогового значения заполнения буферов передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода величины заполнения буфера через заданное значение.

После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра и значения его битов представлены в таблице 387.

Таблица 387 – Регистр IFLS

| Биты | Наименование | Назначение |
|---|---|---|
| 15…6 | | Резерв. Не модифицируйте. При чтении выдаются нули |
| 5…3 | RXIFLSEL | Порог прерывания по заполнению буфера приемника:<br>b000 = Буфер заполнен на 1/8;<br>b001 = Буфер заполнен на 1/4;<br>b010 = Буфер заполнен на 1/2;<br>b011 = Буфер заполнен на 3/4;<br>b100 = Буфер заполнен на 7/8;<br>b101-b111 = резерв |
| 2…0 | TXIFLSEL | Порог прерывания по заполнению буфера передатчика:<br>b000 = Буфер заполнен на 1/8;<br>b001 = Буфер заполнен на 1/4;<br>b010 = Буфер заполнен на 1/2;<br>b011 = Буфер заполнен на 3/4;<br>b100 = Буфер заполнен на 7/8;<br>b101-b111 = резерв.<br>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 |

### 26.12.10 Регистр установки сброса маски прерывания IMSC

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение битов регистра IMSC показано в таблице 388.

Таблица 388 – Регистр IMSC

| Биты | Наименование | Назначение |
|---|---|---|
| 15…11 | | Резерв. Не модифицируйте. При чтении выдаются нули |
| 10 | OEIM | Маска прерывания по переполнению буфера UARTOEINTR:<br>1 – установлена;<br>0 – сброшена |
| 9 | BEIM | Маска прерывания по разрыву линии UARTBEINTR:<br>1 – установлена;<br>0 – сброшена |
| 8 | PEIM | Маска прерывания по ошибке контроля четности UARTPEINTR:<br>1 – установлена;<br>0 – сброшена |

| Биты | Наименование | Назначение |
|---|---|---|
| 7 | FEIM | Маска прерывания по ошибке в структуре кадра UARTFEINTR:<br>1 – установлена;<br>0 – сброшена |
| 6 | RTIM | Маска прерывания по таймауту приема данных UARTRTINTR:<br>1 – установлена;<br>0 – сброшена |
| 5 | TXIM | Маска прерывания от передатчика UARTTXINTR:<br>1 – установлена;<br>0 – сброшена |
| 4 | RXIM | Маска прерывания от приемника UARTRXINTR:<br>1 – установлена;<br>0 – сброшена |
| 3 | DSRMIM | Маска прерывания UARTDSRINTR по изменению состояния линии nUARTDSR:<br>1 – установлена;<br>0 – сброшена |
| 2 | DCDMIM | Маска прерывания UARTDCDINTR по изменению состояния линии nUARTDCD:<br>1 – установлена;<br>0 – сброшена |
| 1 | CTSMIM | Маска прерывания UARTCTSINTR по изменению состояния линии nUARTCTS:<br>1 – установлена;<br>0 – сброшена |
| 0 | RIMIM | Маска прерывания UARTRIINTR по изменению состояния линии nUARTRI:<br>1 – установлена;<br>0 – сброшена |

### 26.12.11 Регистр состояния прерываний RIS

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Предупреждение. После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в «0». Значение битов прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре RIS представлено в таблице 389.

Таблица 389 – Регистр RIS

| Биты | Наименование | Назначение |
|---|---|---|
| 15…11 | | Резерв. Не модифицируйте. При чтении выдаются нули |
| 10 | OERIS | Состояние прерывания по переполнению буфера UARTOEINTR:<br>1 – буфер приемника переполнен;<br>0 – буфер приемника не переполнен |
| 9 | BERIS | Состояние прерывания по разрыву линии UARTBEINTR:<br>1 – произошел разрыв линии приема;<br>0 – разрыва линии приема не происходило |
| 8 | PERIS | Состояние прерывания по ошибке контроля четности UARTPEINTR.<br>1 – возникла ошибка контроля четности;<br>0 – ошибки контроля четности не возникало |
| 7 | FERIS | Состояние прерывания по ошибке в структуре кадра UARTFEINTR:<br>1 – возникла ошибка в структуре кадра;<br>0 – ошибки в структуре кадра не возникало |
| 6 | RTRIS | Состояние прерывания по таймауту приема данных UARTRTINTR*:<br>1 – вышло время таймаута приема данных;<br>0 – время таймаута приема данных не вышло |
| 5 | TXRIS | Состояние прерывания от передатчика UARTTXINTR:<br>1 – возникло прерывание от передатчика;<br>0 – прерывания от передатчика нет |
| 4 | RXRIS | Состояние прерывания от приемника UARTRXINTR:<br>1 – возникло прерывание от приемника;<br>0 – прерывание от приемника не возникало |
| 3 | DSRRMIS | Состояние прерывания UARTDSRINTR по изменению линии nUARTDSR:<br>1 – возникло прерывание;<br>0 – прерывание не возникало |
| 2 | DCDRMIS | Состояние прерывания UARTDCDINTR по изменению линии nUARTDCD:<br>1 – возникло прерывание;<br>0 – прерывание не возникало |
| 1 | CTSRMIS | Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS:<br>1 – возникло прерывание;<br>0 – прерывание не возникало |

* Бит RTRIS может быть установлен только при установленной маске прерывания по таймауту приема данных UARTRTINTR в регистре IMSC. Это вызвано тем, что сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Чтение состояния прерывания по таймауту из регистров MIS и RIS приводит к одинаковым результатам.

| Биты | Наименование | Назначение |
|---|---|---|
| 0 | RIRMIS | Состояние прерывания UARTRIINTR по изменению линии nUARTRI:<br>1 – возникло прерывание;<br>0 – прерывание не возникало |

### 26.12.12 Регистр маскированного состояния прерываний MIS

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в «0». Значение битов прерывания по состоянию модема после сброса не определено.

Назначение бит в регистре MIS представлено в таблице 390.

Таблица 390 – Регистр MIS

| Биты | Наименование | Назначение |
|---|---|---|
| 15…11 | | Резерв. Не модифицируйте. При чтении выдаются нули |
| 10 | OEMIS | Маскированное состояние прерывания по переполнению буфера UARTOEINTR:<br>1 – буфер приемника переполнен;<br>0 – буфер приемника не переполнен |
| 9 | BEMIS | Маскированное состояние прерывания по разрыву линии UARTBEINTR:<br>1 – произошел разрыв линии приема;<br>0 – разрыва линии приема не происходило |
| 8 | PEMIS | Маскированное состояние прерывания по ошибке контроля четности UARTPEINTR:<br>1 – возникла ошибка контроля четности;<br>0 – ошибки контроля четности не возникало |
| 7 | FEMIS | Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR:<br>1 – возникла ошибка в структуре кадра;<br>0 – ошибки в структуре кадра не возникало |
| 6 | RTMIS | Маскированное состояние прерывания по таймауту приема данных UARTRTINTR:<br>1 – вышло время таймаута приема данных;<br>0 – время таймаута приема данных не вышло |
| 5 | TXMIS | Маскированное состояние прерывания от передатчика UARTTXINTR:<br>1 – возникло прерывание от передатчика;<br>0 – прерывания от передатчика нет |
| 4 | RXMIS | Маскированное состояние прерывания от приемника UARTRXINTR:<br>1 – возникло прерывание от приемника;<br>0 – прерывание от приемника не возникало |

| Биты | Наименование | Назначение |
|---|---|---|
| 3 | DSRMMIS | Маскированное состояние прерывания UARTDSRINTR по изменению линии nUARTDSR:<br>1 – возникло прерывание;<br>0 – прерывание не возникало |
| 2 | DCDMMIS | Маскированное состояние прерывания UARTDCDINTR по изменению линии nUARTDCD:<br>1 – возникло прерывание;<br>0 – прерывание не возникало |
| 1 | CTSMMIS | Маскированное состояние прерывания UARTCTSINTR по изменению линии nUARTCTS:<br>1 – возникло прерывание;<br>0 – прерывание не возникало |
| 0 | RIMMIS | Маскированное состояние прерывания UARTRIINTR по изменению линии nUARTRI:<br>1 – возникло прерывание;<br>0 – прерывание не возникало |

### 26.12.13 Регистр сброса прерываний ICR

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи «1» в соответствующий бит. Запись «0» в любой из разрядов регистра игнорируется.

Назначение бит в регистре ICR представлено в таблице 391.

Таблица 391 – Регистр ICR

| Биты | Наименование | Назначение |
|---|---|---|
| 15…11 | | Резерв. Не модифицируйте. При чтении выдаются нули |
| 10 | OEIC | Сброс прерывания по переполнению буфера UARTOEINTR:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 9 | BEIC | Сброс прерывания по разрыву линии UARTBEINTR:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 8 | PEIC | Сброс прерывания по ошибке контроля четности UARTPEINTR:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 7 | FEIC | Сброс прерывания по ошибке в структуре кадра UARTFEINTR:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 6 | RTIC | Сброс прерывания по таймауту приема данных UARTRTINTR:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 5 | TXIC | Сброс прерывания от передатчика UARTTXINTR:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |

| Биты | Наименование | Назначение |
|---|---|---|
| 4 | RXIC | Сброс прерывания от приемника UARTRXINTR:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 3 | DSRMIC | Сброс прерывания UARTDSRINTR по изменению линии nUARTDSR:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 2 | DCDMIC | Сброс прерывания UARTDCDINTR по изменению линии nUARTDCD:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 1 | CTSMIC | Сброс прерывания UARTCTSINTR по изменению линии nUARTCTS:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |
| 0 | RIMIC | Сброс прерывания UARTRIINTR по изменению линии nUARTRI:<br>1 – сброс прерывания;<br>0 – не влияет на состояние регистра |

### 26.12.14 Регистр управления прямым доступом к памяти DMACR

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются. Назначение бит регистра DMACR представлено в таблице 392.

Таблица 392 – Регистр DMACR

| Биты | Наименование | Назначение |
|---|---|---|
| 15…13 | | Резерв. Не модифицируйте. При чтении выдаются нули |
| 12…3 | - | Зарезервировано |
| 2 | DMAONERR | Блокирование запросов DMA при приеме при возникновении ошибок на линии:<br>1 – в случае возникновения прерывания по обнаружению ошибки блокируются запросы DMA от приемника UARTRXDMASREQ и UARTRXDMABREQ;<br>0 – DMA запросы не блокируются |
| 1 | TXDMAE | Использование DMA при передаче:<br>1 – разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика;<br>0 – запрещено формирование запросов DMA |
| 0 | RXDMAE | Использование DMA при приеме:<br>1 – разрешено формирование запросов DMA для обслуживания буфера FIFO приемника;<br>0 – запрещено формирование запросов DMA |

### 26.12.15 Регистр управления тестированием TCR

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются. Назначение бит регистра TCR представлено в таблице 393.

Таблица 393 – Регистр TCR

| Биты | Наименование | Назначение |
|---|---|---|
| 15…13 | | Резерв. При чтении значение непредсказуемо |
| 12…3 | - | Зарезервировано |
| 2 | SIRTEST | Разрешение приема данных в кольцевом режиме с выхода IrDA передатчика:<br>1 – разрешено;<br>0 – запрещено.<br>Используется совместно с установкой бита LBE в регистре CR |
| 1 | TESTFIFO | Разрешение чтения данных из FIFO передатчика и запись в FIFO приемника:<br>1 – разрешено;<br>0 – запрещено |
| 0 | ITEN | Перевод контроллера UART в тестовый режим:<br>1 – тестовый режим разрешен;<br>0 – тестовый режим запрещен |

# 27 Контроллер прямого доступа в память DMA

## 27.1 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 циклов DMA;
- 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.

## 27.2 Термины и определения

Таблица 394 – Термины и определения

| | |
|---|---|
| Альтернативная | Альтернативная структура управляющих данных канала.<br>Вы можете установить соответствующий регистр для изменения типа структуры данных (см. подраздел 27.5 «Структура управляющих данных канала») |
| C | Идентификатор номера канала прямого доступа. Например:<br>C=1 – канал DMA 1<br>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.<br>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.<br>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 контроллер DMA 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 четыре передачи DMA, 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 четыре передачи DMA, используя первичную структуру управляющих данных, которые опять перенастраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных.<br>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 |

## 27.3 Функциональное описание

На рисунке 119 показана упрощенная структурная схема контроллера.

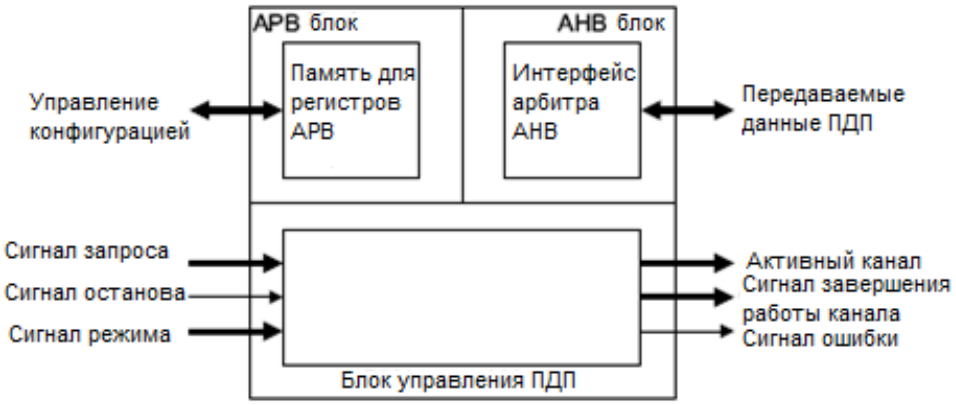



Рисунок 119 – Структурная схема контроллера

Контроллер состоит из следующих основных функциональных блоков:

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

### 27.3.1 Распределение каналов DMA

Таблица 395 – Распределение каналов DMA

| Номер канала | Источник req | Источник sreq | Описание |
|---|---|---|---|
| 0 | UART1_TX_DMA_BREQ | UART1_TX_DMA_SREQ | Запрос от передатчика UART1 |
| 1 | UART1_RX_DMA_BREQ | UART1_RX_DMA_SREQ | Запрос от приемника UART1 |
| 2 | UART2_TX_DMA_BREQ | UART2_TX_DMA_SREQ | Запрос от передатчика UART2 |
| 3 | UART2_RX_DMA_BREQ | UART2_RX_DMA_SREQ | Запрос от приемника UART2 |
| 4 | SSP1_TX_DMA_BREQ | SSP1_TX_DMA_SREQ | Запрос от передатчика SPI1 |
| 5 | SSP1_RX_DMA_BREQ | SSP1_RX_DMA_SREQ | Запрос от приемника SPI1 |
| 6 | SSP2_TX_DMA_BREQ | SSP2_TX_DMA_SREQ | Запрос от передатчика SPI2 |
| 7 | SSP2_RX_DMA_BREQ | SSP2_RX_DMA_SREQ | Запрос от приемника SPI2 |
| 8 | SSP3_TX_DMA_BREQ | SSP3_TX_DMA_SREQ | Запрос от передатчика SPI3 |
| 9 | SSP3_RX_DMA_BREQ | SSP3_RX_DMA_SREQ | Запрос от приемника SPI3 |
| 10 | TMR1_DMA_REQ | TMR1_DMA_REQ | Запрос REQ от таймера общего назначения TIMER1 |
| 11 | TMR2_DMA_REQ | TMR2_DMA_REQ | Запрос REQ от таймера общего назначения TIMER2 |
| 12 | TMR3_DMA_REQ | TMR3_DMA_REQ | Запрос REQ от таймера общего назначения TIMER3 |
| 13 | TMR4_DMA_REQ | TMR4_DMA_REQ | Запрос REQ от таймера общего назначения TIMER4 |
| 14 | - | TMR1_DMA_REQ1 | Запрос REQ1 от таймера общего назначения TIMER1 |
| 15 | - | TMR1_DMA_REQ2 | Запрос REQ2 от таймера общего назначения TIMER1 |
| 16 | - | TMR1_DMA_REQ3 | Запрос REQ3 от таймера общего назначения TIMER1 |
| 17 | - | TMR1_DMA_REQ4 | Запрос REQ4 от таймера общего назначения TIMER1 |
| 18 | - | TMR2_DMA_REQ1 | Запрос REQ1 от таймера общего назначения TIMER2 |
| 19 | - | TMR2_DMA_REQ2 | Запрос REQ2 от таймера общего назначения TIMER2 |
| 20 | - | TMR2_DMA_REQ3 | Запрос REQ3 от таймера общего назначения TIMER2 |
| 21 | - | TMR2_DMA_REQ4 | Запрос REQ4 таймера общего назначения TIMER2 |
| 22 | - | TMR3_DMA_REQ1 | Запрос REQ1 от таймера общего назначения TIMER3 |
| 23 | - | TMR3_DMA_REQ2 | Запрос REQ2 от таймера общего назначения TIMER3 |
| 24 | - | TMR3_DMA_REQ3 | Запрос REQ3 от таймера общего назначения TIMER3 |

| Номер канала | Источник req | Источник sreq | Описание |
|---|---|---|---|
| 25 | - | TMR3_DMA_REQ4 | Запрос REQ4 таймера общего назначения TIMER3 |
| 26 | - | TMR4_DMA_REQ1 | Запрос REQ1 от таймера общего назначения TIMER4 |
| 27 | - | TMR4_DMA_REQ2 | Запрос REQ2 от таймера общего назначения TIMER4 |
| 28 | - | TMR4_DMA_REQ3 | Запрос REQ3 от таймера общего назначения TIMER4 |
| 29 | - | TMR4_DMA_REQ4 | Запрос REQ4 таймера общего назначения TIMER4 |
| 30 | - | ADC_DMA_SREQ | Запрос от АЦП последовательных приближений |
| 31 | - | - | Программный (не вызывает прерывание от DMA) |

### 27.3.2 Блок, подключенный к шине APB

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

### 27.3.3 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.

### 27.3.4 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 передач DMA:
  - память – память;
  - память – периферия;
  - периферия – память.

### 27.3.5 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.

### 27.3.6 Разрядность передаваемых данных

Контроллер интерфейса предоставляет возможность осуществлять передачу 8-, 16- и 32-разрядных данных. Таблица 396 отображает значения комбинаций шины HSIZE.

Таблица 396 – Комбинации шины HSIZE

| HSIZE[2]* | HSIZE[1] | HSIZE[0] | Разрядность данных (бит) |
|---|---|---|---|
| 0 | 0 | 0 | 8 |
| 0 | 0 | 1 | 16 |
| 0 | 1 | 0 | 32 |
| | 1 | 1 | ** |
| * Сигнал постоянно удерживается в состоянии логический ноль.<br>** Запрещенная комбинация | | | |

Контроллер всегда использует передачи 32-разрядными данными при обращении к управляющим данным канала. Необходимо устанавливать разрядность данных источника соответствующую разрядности данных приемника.

### 27.3.7 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.

Таблица 397 перечисляет значения комбинаций шины HPROT.

Таблица 397 – Режимы защиты данных

| HPROT[3] кэширование | HPROT[2] буферизация | HPROT[1] привелигированный | HPROT[0] данные/команда | Описание |
|---|---|---|---|---|
| - | - | - | 1* | Доступ к данным |
| - | - | 0 | - | Пользовательский доступ |
| - | - | 1 | - | Привилегированный доступ |
| - | 0 | - | - | Без буферизации |
| - | 1 | - | - | Буферизованный |
| 0 | - | - | - | Без кэширования |

| | | | | |
|---|---|---|---|---|
| 1 | - | - | - | Кэшированный |

Контроллер удерживает HPROT[0] в состоянии логической «1», чтобы обозначить доступ к данным.

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

Для каждого канала DMA также возможен выбор режима защиты данных. Более подробно это описано в разделе Управление DMA.

### 27.3.8 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 398 перечислены возможные комбинации.

Таблица 398 – Инкремент адреса

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

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

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

## 27.4 Управление DMA

### 27.4.1 Правила обмена данными

Контроллер использует правила обмена данными (Таблица 399), при соблюдении следующих условий:

– канал DMA включен, что выполняется установкой в состояние логической «1» 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 «0» разряда управления chnl_req_mask_set[C].

Таблица 399 – 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.<br>В периферийном режиме «Исполнение с изменением конфигурации», dma_active[C] остается в состоянии 1 в течение каждой пары передач DMA, 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:<br>– если dma_stall в состоянии «0», контроллер устанавливает dma_done[] в состояние «1» продолжительностью один такт HCLK<br>– 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:<br>контроллер завершит $2^R$ передач (или число передач, указанное в регистре n_minus_1);<br>dma_req[C] будет установлен в «0»;<br>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:<br>контроллер установит dma_active[C] в «0» на следующем такте сигнала HCLK;<br>передача по каналу C не завершится, пока не будет сброшен в «0» dma_stall |

| Правило | Описание |
|---|---|
| 13 | Контроллер игнорирует dma_sreq[C], если dma_waitonreq[C] в состоянии «0» |
| 14 | Контроллер игнорирует dma_sreq[C], если chnl_useburst_set[C] в состоянии «1»* |
| 15 | Для циклов DMA, отличных по типу от периферийного режима «Исполнение с изменением конфигурации», по окончании $2^R$ передач контроллер устанавливает значение chnl_useburst_set[C] в состояние «0», если количество оставшихся передач меньше, чем $2^R$.<br>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 передачу DMA.<br>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) передач DMA.<br>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 передачи DMA согласно правилам, представленным в таблице 400.

Таблица 400 – Правила осуществления передач DMA 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» |

### 27.4.2 Диаграммы работы контроллера DMA

Данный раздел описывает следующие примеры функционирования контроллера с использованием правил обмена данными (Таблица 399):

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

Примечание – Все диаграммы, показанные на рисунках 120 – 124, подразумевают следующее:

Импульсный запрос на обработку

Рисунок 120 показывает временную диаграмму работы контроллера DMA при получении импульсного запроса от периферии.

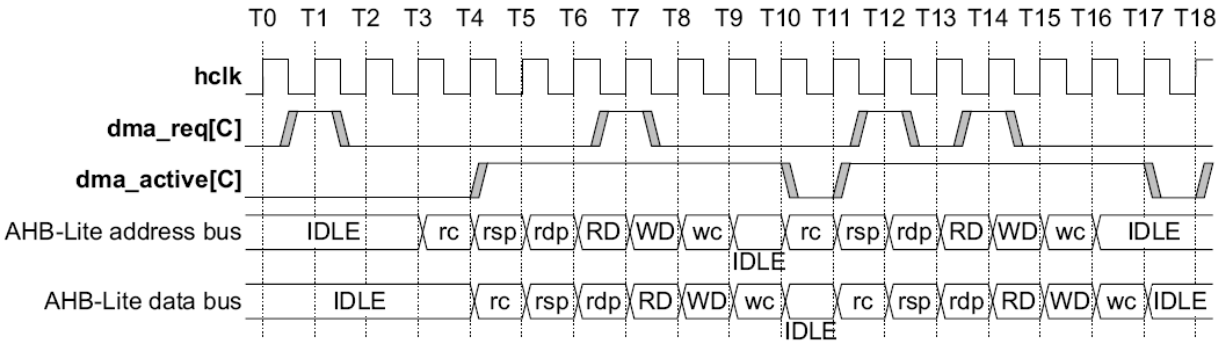



Рисунок 120 – Диаграмма работы при получении импульсного запроса от периферийного блока

Таблица 401 – 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 передачи DMA по каналу C |
| T4-T7 | Контроллер считывает управляющие данные канала, где:<br>rc – чтение настроек канала, channel_cfg;<br>rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr;<br>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:<br>RD – чтение данных;<br>WD – запись данных |
| T9-T10 | Контроллер осуществляет запись настроек канала, channel_cfg, где<br>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) |

Запрос на обработку по уровню.

Рисунок 121 показывает временную диаграмму работы контроллера DMA при получении от периферии запроса на обработку по уровню.

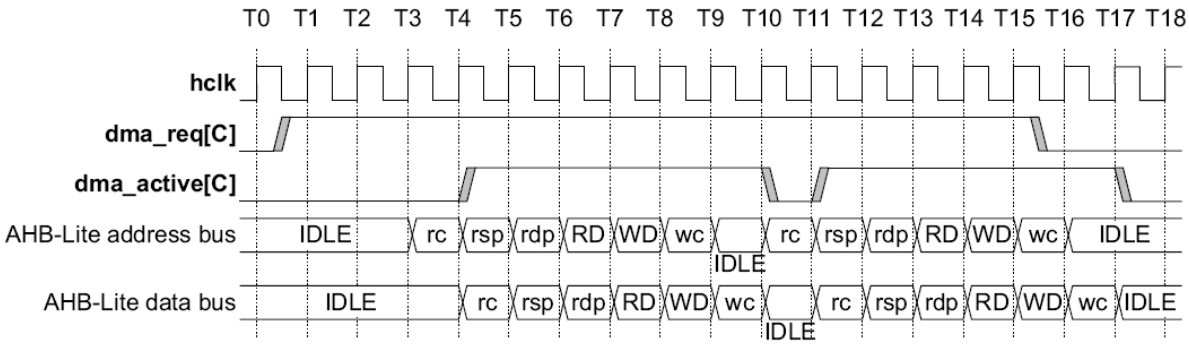



Рисунок 121 – Диаграмма работы при получении запроса на обработку по уровню

Таблица 402 – Пояснения к диаграмме работы при получении запроса на обработку по уровню

| | |
|---|---|
| T1 | Контроллер обнаружил запрос на обработку по каналу C (Таблица 399, 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 передачи DMA по каналу C |
| T4-T7 | Контроллер считывает управляющие данные канала, где:<br>rc – чтение настроек канала, channel_cfg;<br>rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr;<br>rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr. |
| T7-T9 | Контроллер выполняет передачу DMA по каналу C, где:<br>RD – чтение данных;<br>WD – запись данных. |
| T9-T10 | Контроллер осуществляет запись настроек канала, channel_cfg, где<br>wc – запись настроек канала, channel_cfg |
| T10 | Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4).<br>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 передачу DMA 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.

Флаги завершения

Рисунок 122 демонстрирует функционирование сигнала (флага) dma_done[] при следующих условиях:

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

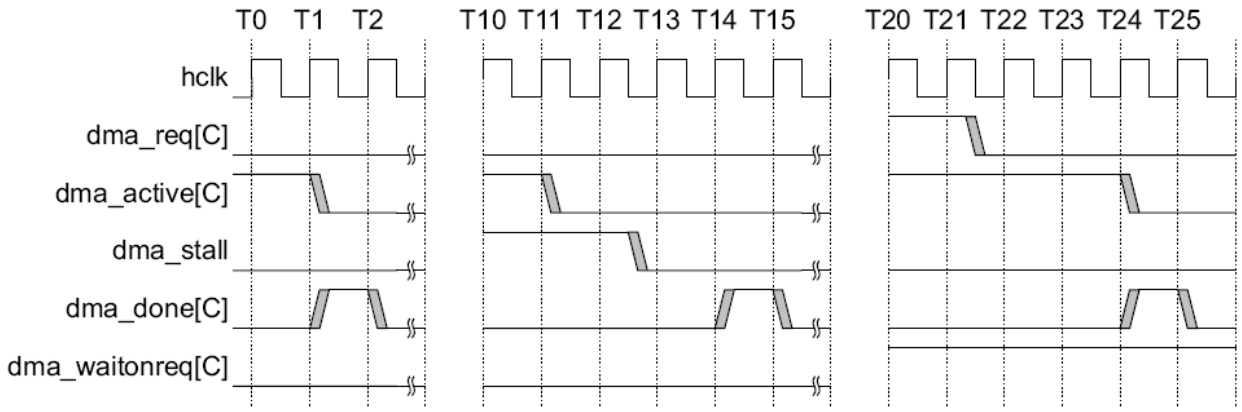



Рисунок 122 – Диаграммы функционирования dma_done

Таблица 403 – 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 404 – 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) |

Примечание к T11 – Контроллер не устанавливает сигнал dma_done[C], так как сигнал dma_stall установлен в 1 в предшествующем такте HCLK (см. правила 9 и 12).

Таблица 405 – 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) |

Флаги ожидания запроса на обработку

Ниже приведены рисунки, которые 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.

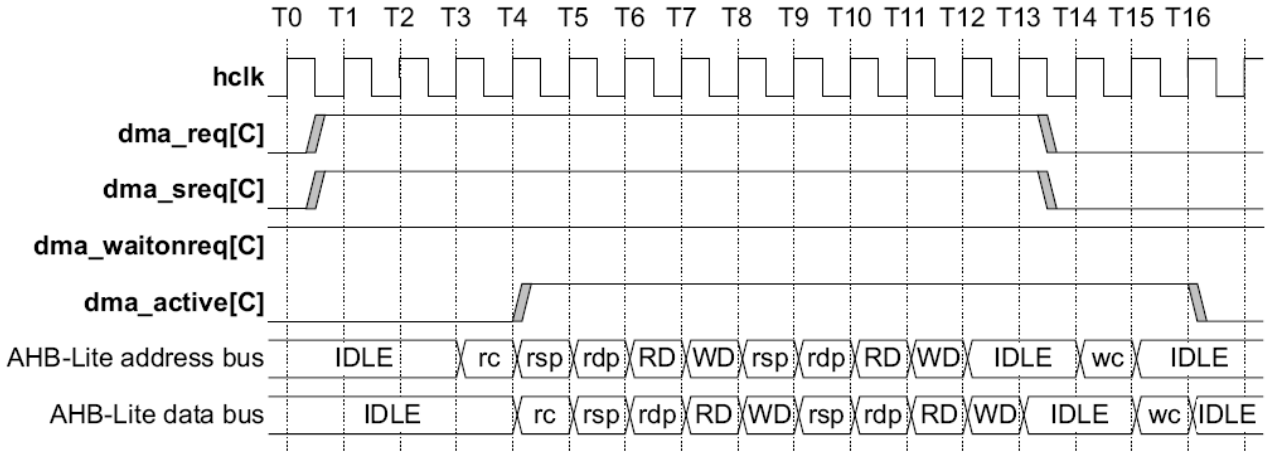



Рисунок 123 – Диаграмма работы контроллера DMA при использовании dma_waitonreq

Таблица 406 – 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».<br>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 передачи DMA по каналу C |
| T4-T7 | Контроллер считывает управляющие данные канала, где:<br>rc – чтение настроек канала, channel_cfg;<br>rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr;<br>rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr |
| T7-T9 | Контроллер выполняет передачу DMA по каналу C, где:<br>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<br>wc – запись настроек канала, channel_cfg |
| T16 | Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11).<br>Контроллер устанавливает значение по чтению регистра chnl_useburst_set[C] в 0, если количество оставшихся передач менее $2^R$ (см. правило 15) |

Рисунок 124 показывает работу контроллера DMA при установке dma_waitonreq в 1 и выполнении одиночной передачи DMA.

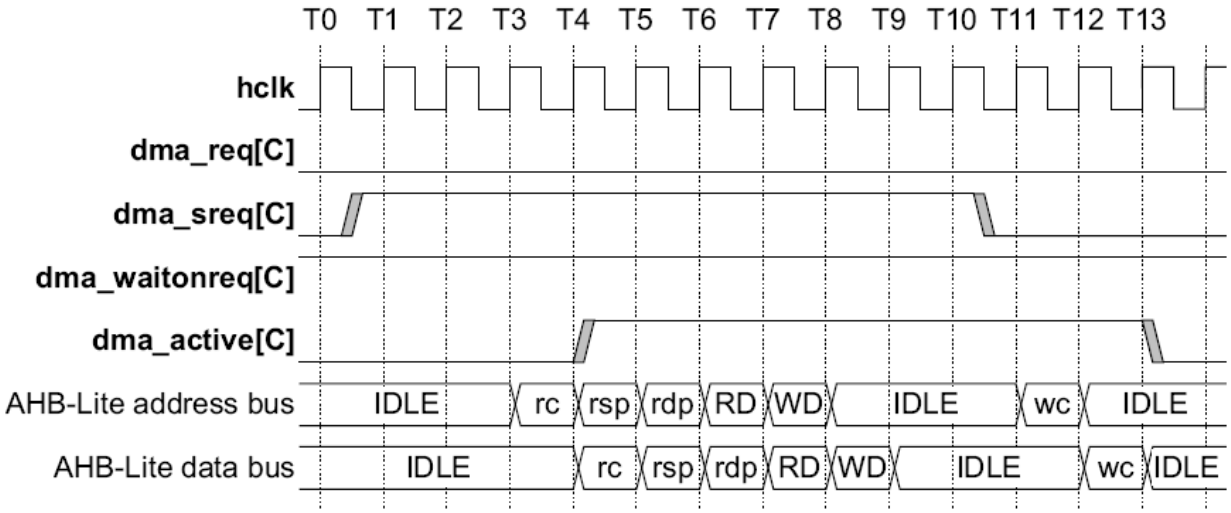



Рисунок 124 – Работа DMA при использовании dma_waitonreq совместно с dma_sreq

Таблица 407 – 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 передачи DMA по каналу C |
| T4-T7 | Контроллер считывает управляющие данные канала, где:<br>rc – чтение настроек канала, channel_cfg;<br>rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr;<br>rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr |
| T7-T9 | Контроллер выполняет передачу DMA по каналу C, где:<br>RD – чтение данных;<br>WD – запись данных. |

| | |
|---|---|
| | Это запрос в ответ на dma_sreq[], таким образом, R=0 и, следовательно, контроллер исполнит 1 передачу DMA |
| T10-T11 | Периферийный блок сбрасывает сигнал dma_sreq[C] |
| T12_T13 | Контроллер осуществляет запись настроек канала, channel_cfg, где<br>wc – запись настроек канала, channel_cfg |
| T13 | Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11) |

### 27.4.3 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 $2^R$. Для примера, если R равно четырем, то арбитраж будет проводиться через каждые 16 передач DMA.

Таблица 408 показывает возможную периодичность арбитража.

Таблица 408 – 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 разделе «Управляющие данные канала».

### 27.4.4 Приоритет

При проведении арбитража определяется канал для обслуживания в следующем цикле DMA. На выбор следующего канала влияют:

– номер канала;
– уровень приоритета, присвоенного каналу.

Каждому каналу может быть присвоен уровень приоритета по умолчанию (низкий) или высокий уровень приоритета. Присвоение уровня приоритета осуществляется установкой или сбросом разряда chnl_priority_set.

Канал номер 0 имеет высший уровень приоритета, уровень приоритета снижается с увеличением номера канала. Таблица 409 показывает уровень приоритета каналов DMA в порядке его уменьшения.

Таблица 409 – 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 контроллер DMA выбирает следующий для обслуживания канал из всех включенных каналов DMA.




Рисунок 125 – Алгоритм выбора следующего канала для обслуживания

Начало выбора.

Есть ли запрос на обслуживание.

Есть ли среди запросов на обслуживание канал с установленным разрядом «высокий уровень».

Выбор канала с низшим порядковым номером и установленным таким разрядом.

Выбор канала с низшим порядковым номером.

Начало цикла DMA.

### 27.4.5 Типы циклов DMA

Разряды cycle_ctrl определяют, как контроллер будет выполнять циклы DMA. Описание значений этих разрядов приведено в таблице 410.

Таблица 410 – 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. раздел «Настройка управляющих данных канала».<br>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».

#### 27.4.5.1 *Недействительный*

После окончания цикла контроллер DMA устанавливает тип цикла в значение «недействительный» для предотвращения повтора выполненного цикла DMA.

#### 27.4.5.2 *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.

#### 27.4.5.3 *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.

#### 27.4.5.4 *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 «неправильную» структуру данных или пока процессор не запретит работу канала.

Рисунок 126 демонстрирует пример функционирования контроллера в режиме «пинг-понг».

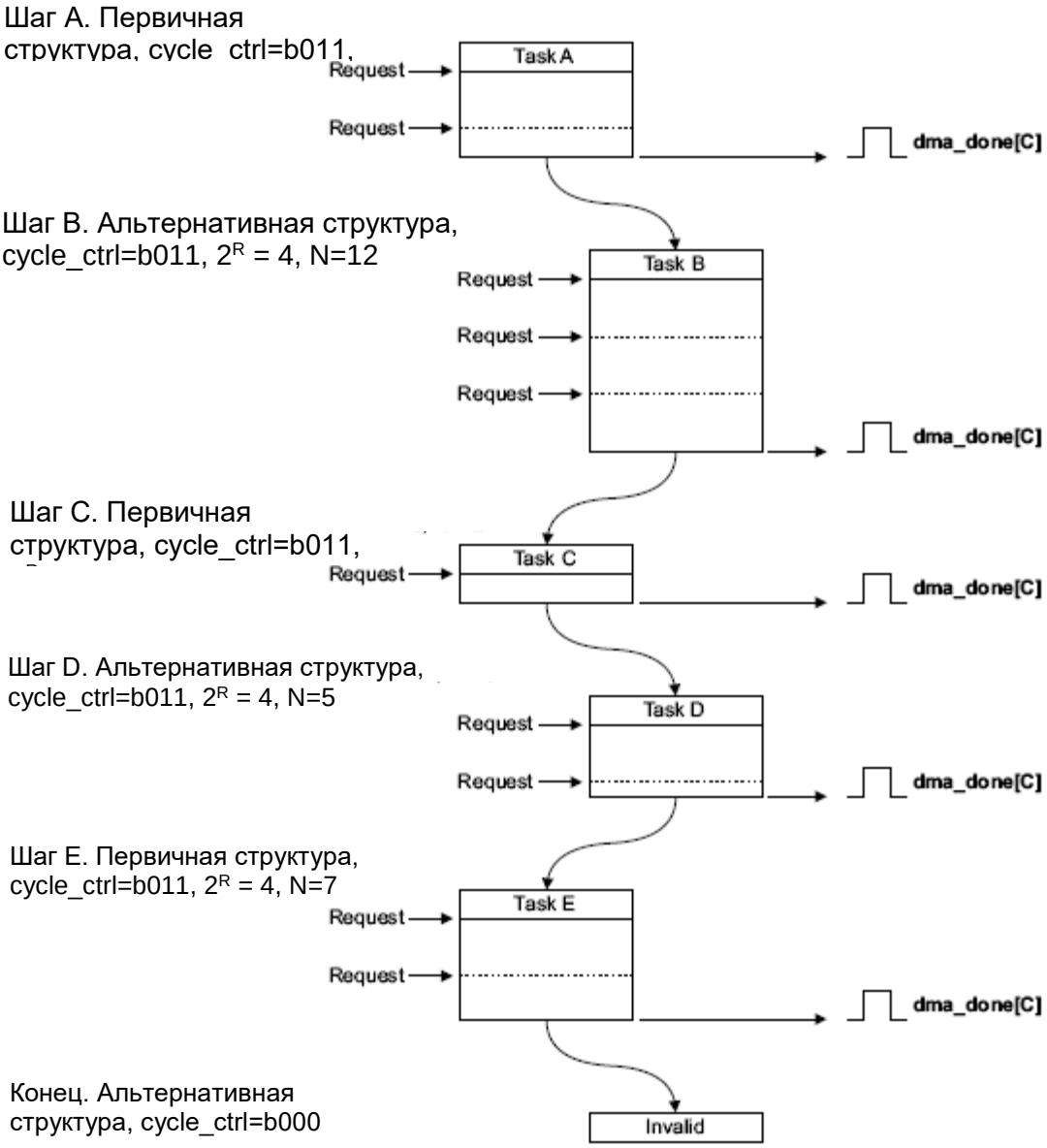



Рисунок 126 – Пример функционирования контроллера в режиме «пинг-понг»

Пояснения к рисунку

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 четыре передачи DMA.

Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.

Контроллер выполняет оставшиеся две передачи DMA.

Контроллер устанавливает dma_done[C] в состояние «1» на один такт сигнала синхронизации HCLK и входит в процедуру арбитража

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

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

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

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

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

Если контроллер получит новый запрос на обработку от данного канала и этот запрос будет самым приоритетным, контроллер предпримет попытку выполнения

следующего шага. Однако из-за того, что процессор не установил альтернативные управляющие данные, и по окончании шага D контроллер установил cycle_ctrl в состояние b000, передачи DMA прекращаются.

Примечание – Для прерывания цикла DMA, исполняемого в режиме «пинг-понг», также возможен перевод режима работы контроллера на шаге E в режим «Основной цикл DMA» путем установки cycle_ctrl в 3’b001.

#### 27.4.5.5 *Режим работы с памятью «исполнение с изменением конфигурации»*

В данном режиме контроллер, получая начальный запрос на обработку, выполняет четыре передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные. Затем контроллер выполняет еще четыре передачи DMA, используя первичные управляющие данные. Контроллер продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

– процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;

– контроллер считает «неправильную» структуру управляющих данных.

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

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

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 411 перечисляет области памяти channel_cfg, которые должны быть определены константами, и значения которых определяются пользователем.

Таблица 411 – Channel_cfg для первичной структуры управляющих данных в режиме работы с памятью «исполнение с изменением конфигурации»

| Разряды | Обозначение | Значение | Описание |
|---|---|---|---|
| Области с константными значениями | | | |
| 31, 30 | dst_inc | 2’b10 | Контроллер производит инкремент адреса пословно |
| 29, 28 | dst_size | 2’b10 | Контроллер осуществляет передачу пословно |
| 27, 26 | src_inc | 2’b10 | Контроллер производит инкремент адреса пословно |
| 25, 24 | src_size | 2’b10 | Контроллер осуществляет передачу пословно |
| 17…14 | R_power | 4’b0010 | Контроллер выполняет четыре передачи DMA |
| 3 | next_useburst | 1’b0 | Для данного режима этот разряд должен быть равен 0 |

| Разряды | Обозначение | Значение | Описание |
|---|---|---|---|
| 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 127 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 |

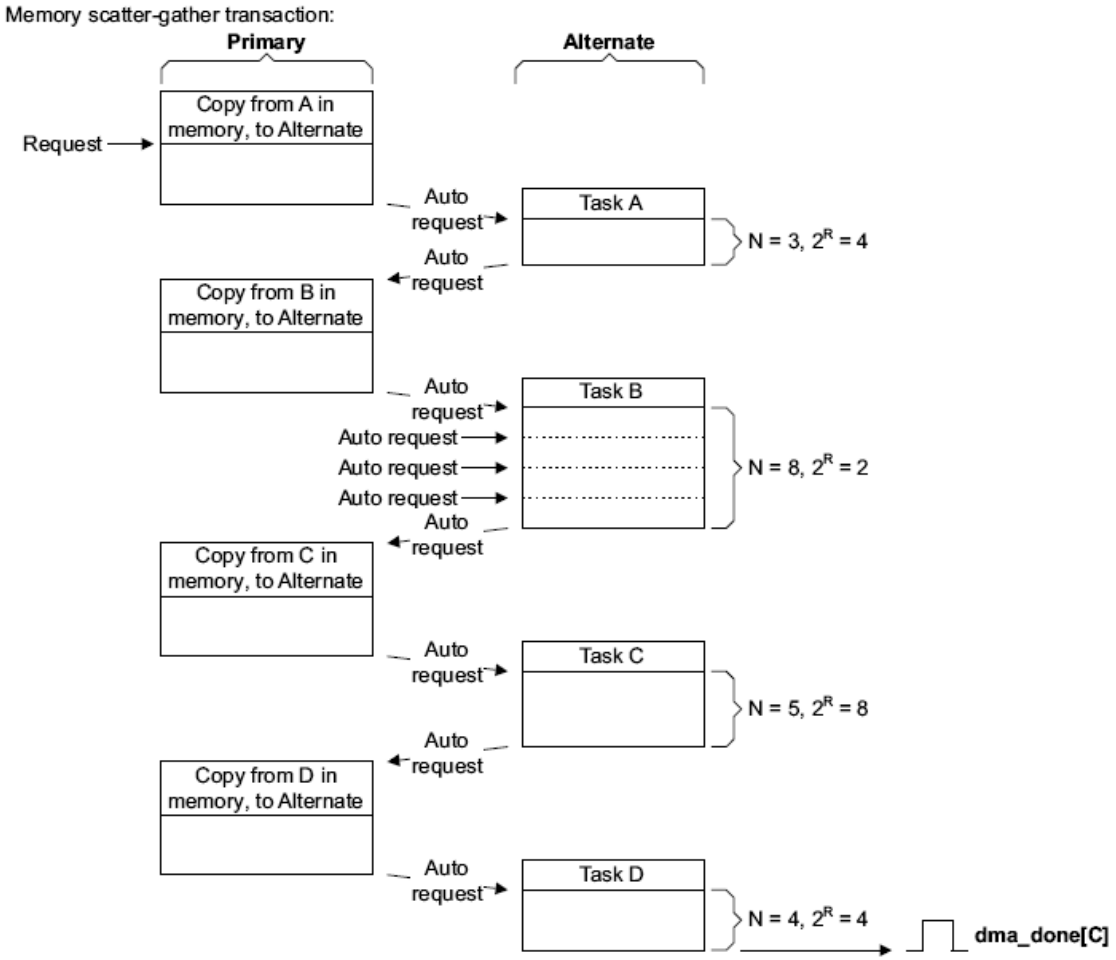



Рисунок 127 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

*Инициализация:*

1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с памятью «исполнение с изменением конфигурации» путем установки cycle_ctrl в 3b’100. Так как управляющие данные канала состоят из четырех слов, необходимо установить $2^R$ в 4. В этом примере количество передач равно четырем и поэтому N установлен в 16.

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

3 Процессор разрешает работу канала DMA.

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

**Первичная, копирование A**

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

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

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

**Шаг D**

Контроллер выполняет шаг D, используя основной цикл DMA.

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

#### 27.4.5.6 *Режим работы с периферией «исполнение с изменением конфигурации»*

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

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

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

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

– процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;

– контроллер считает «неправильную» структуру управляющих данных.

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

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

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. Таблица 412 перечисляет области памяти channel_cfg, которые должны быть определены константами, и те области, значения которых определяются пользователем.

Таблица 412 – Channel_cfg для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации»

| Разряды | Обозначение | Значение | Описание |
|---|---|---|---|
| Области с константными значениями | | | |
| 31, 30 | dst_inc | 2’b10 | Контроллер производит инкремент адреса пословно |
| 29, 28 | dst_size | 2’b10 | Контроллер осуществляет передачу пословно |
| 27, 26 | src_inc | 2’b10 | Контроллер производит инкремент адреса пословно |
| 25, 24 | src_size | 2’b10 | Контроллер осуществляет передачу пословно |
| 17…14 | R_power | 4’b0010 | Контроллер выполняет четыре передачи DMA |
| 2…0 | cycle_ctrl | 3’b110 | Контролер работает в режиме работы с периферией «исполнение с изменением конфигурации» |

| 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 кратно 4. Число равное N/4 это количество раз, которое нужно настраивать альтернативные управляющие данные | | | |

Рисунок 128 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 |

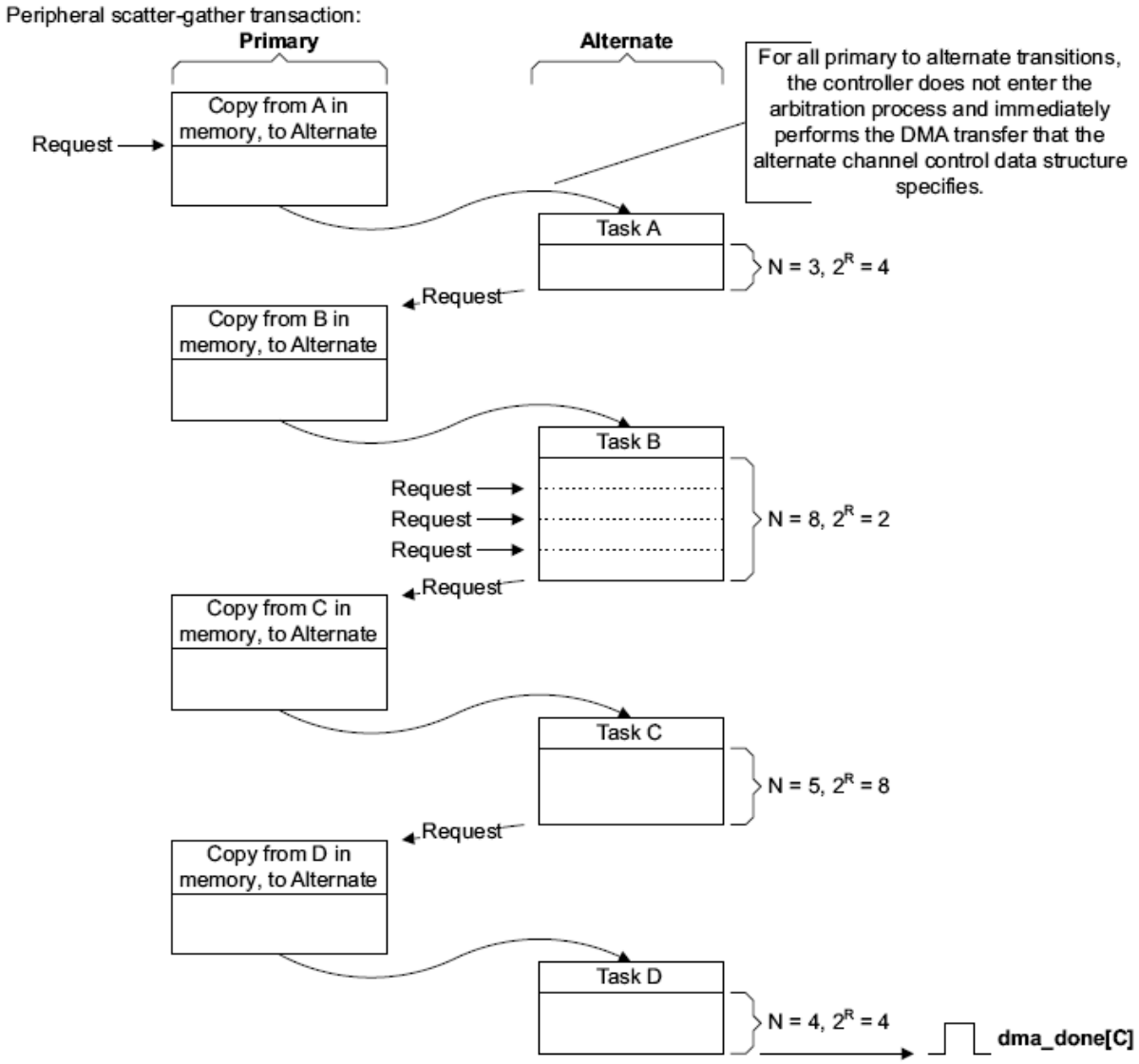



Рисунок 128 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

*Пояснения к рисунку*

*Инициализация:*

1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации» путем установки cycle_ctrl в 3'b110. Так как управляющие данные канала состоят из четырех слов, необходимо установить $2^R$ в 4. В этом примере количество задач равно четырем 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 четыре 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 три запроса.

По окончании контроллер проводит процедуру арбитража.

Первичная, копирование из области C памяти.

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

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

**Шаг D**

Контроллер выполняет шаг D, используя основной цикл DMA.

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

**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:

– чтение регистра chnl_enable_set с целью создания списка отключенных каналов;

– если канал установил флаг dma_done[], то контроллер отключает канал. Программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно установили флаги dma_done[];

– процессор должен сравнить список выключенных каналов, полученный в шаге 1, с данными о каналах, которые недавно устанавливали флаги dma_done[]. Канал, по которому отсутствуют данные об установке флага dma_done[], это и есть канал, с которым связана ошибка.

## 27.5 Структура управляющих данных канала

В системной памяти должна быть отведена область для хранения управляющих данных каналов. Системная память должна:

– предоставлять смежную область системной памяти, к которой контроллер и процессор имеют доступ;

– иметь базовый адрес, который целочисленно кратен общему размеру структуры управляющих данных канала.

Рисунок 129 показывает область памяти необходимую контроллеру для структур управляющих данных канала, при использовании всех 32 каналов и опциональной альтернативной структуры управляющих данных.

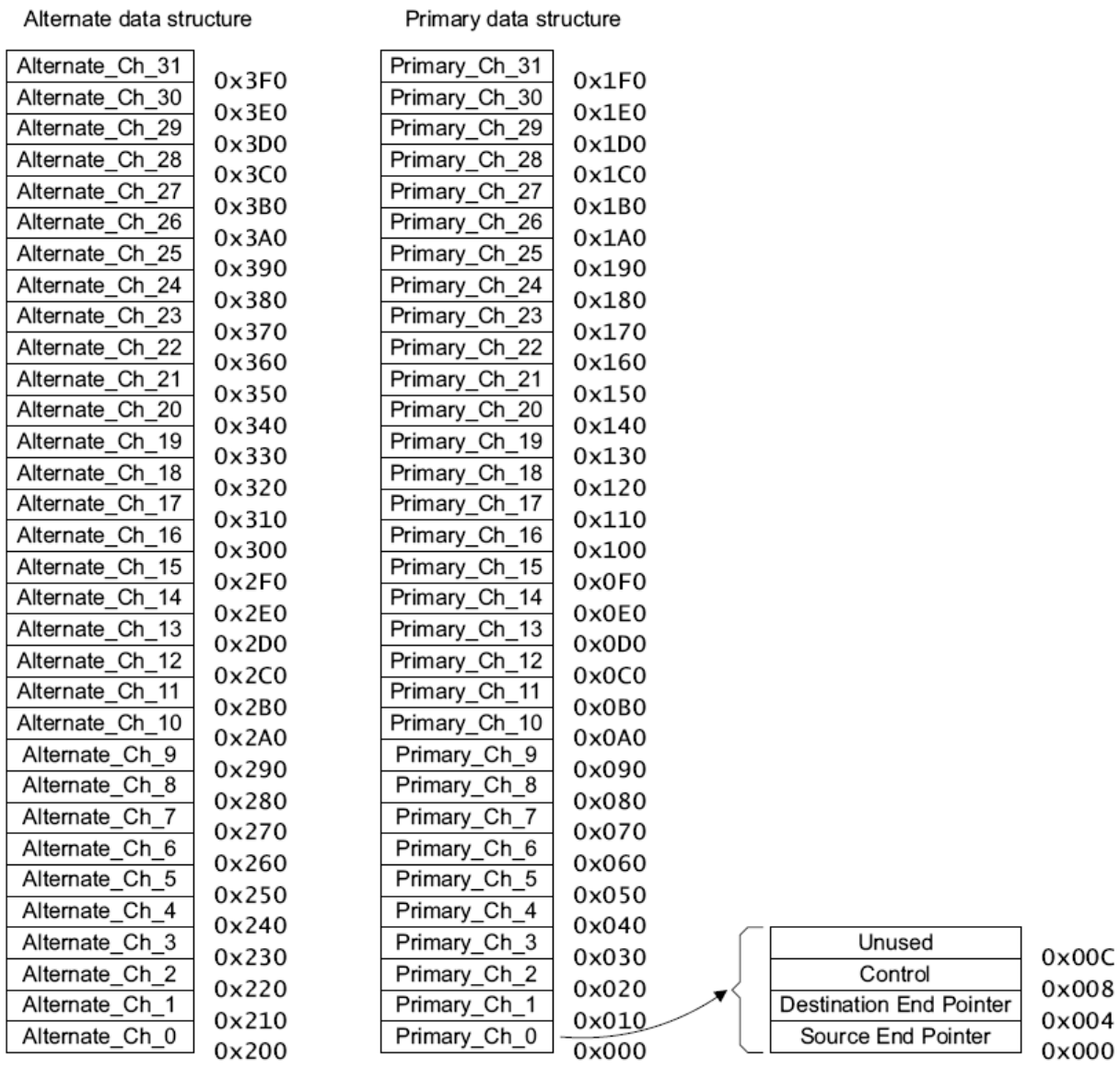



Рисунок 129 – Карта памяти для 32 каналов, включая альтернативную структуру управляющих данных

Пример использует 1 Кбайт системной памяти. В этом примере контроллер использует младшие 10 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXX000, далее 0xXXXXX400, далее 0xXXXXX800, далее 0xXXXXXC00.

Базовый адрес для первичной структуры управляющих данных устанавливается путем записи соответствующего значения в регистр ctrl_base_ptr.

Необходимый размер области системной памяти зависит от:

– количества каналов, используемых в контроллере;
– от того, используется или нет альтернативная структура управляющих данных.

Таблица 413 перечисляет разряды адреса, которые используются контроллером при доступе к различным элементам структуры управляющих данных, в зависимости от количества каналов, используемых в контроллере.

Таблица 413 – Разряды адреса, соответствующие элементам структуры управляющих данных

| Количество каналов, используемых в контроллере | [9] | [8] | [7] | [6] | [5] | [4] | [3:0] |
|---|---|---|---|---|---|---|---|
| 1 | | | | | | A | 0x0<br>0x4<br>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 130 демонстрирует пример реализации контроллера с использованием трех каналов DMA и альтернативной структурой управляющих данных.

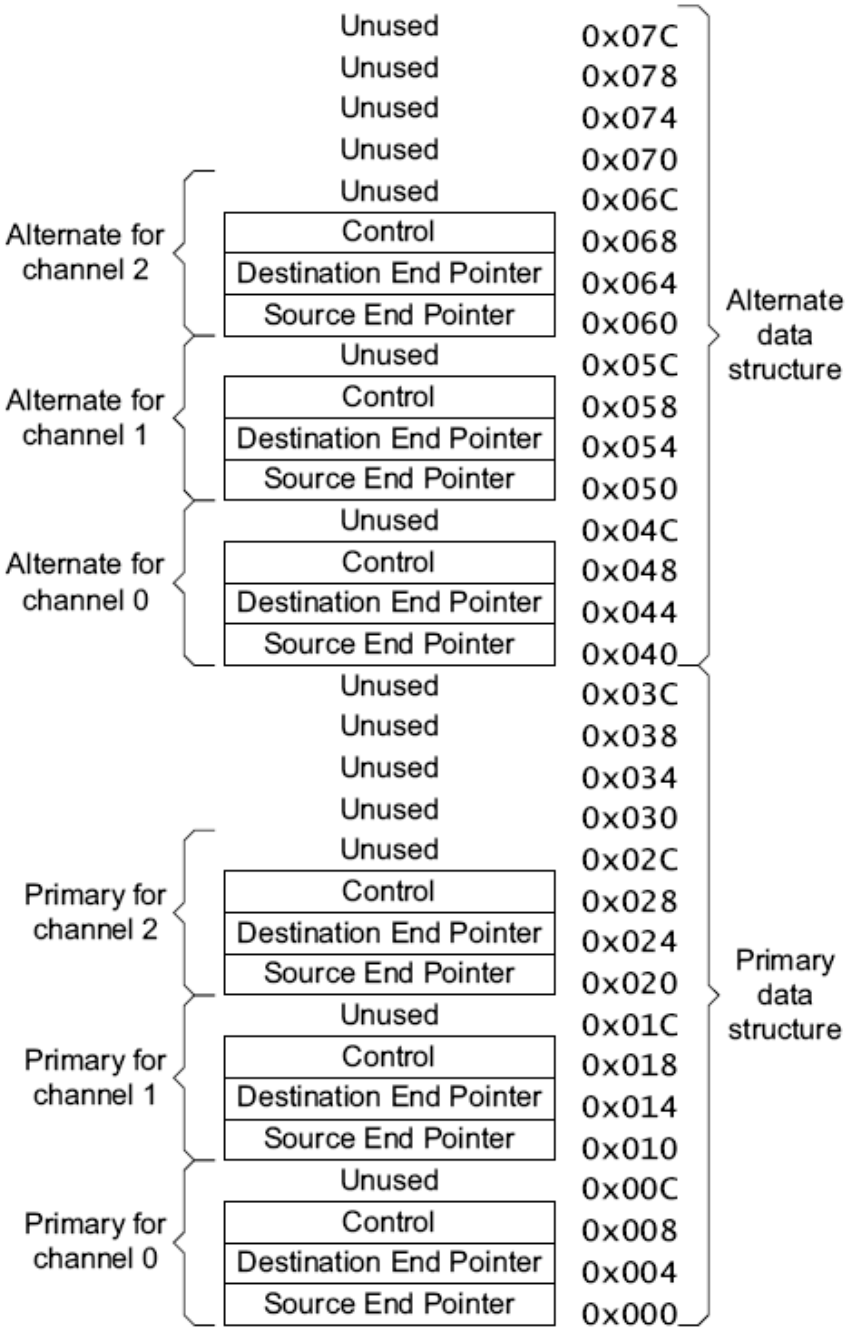



Рисунок 130 – Карта памяти для трех каналов DMA, включая альтернативную структуру управляющих данных (где Destination end pointer – указатель конца данных приемника; Source end pointer – указатель конца данных источника; Control – управление)

Пример структуры управляющих данных использует 128 байт системной памяти. В этом примере контроллер использует младшие 6 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 0xXXXXXX00, далее 0xXXXXXX80.

Таблица 414 перечисляет все разрешенные значения базового адреса для первичной структуры управляющих данных, в зависимости от количества каналов DMA, использованных в контроллере.

Таблица 414 – 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.

Указатель конца данных источника

Область памяти под названием src_data_end_ptr содержит указатель на последний адрес месторасположения данных источника. Таблица 415 перечисляет значения разрядов этой области.

Таблица 415 – Значения разрядов src_data_end_ptr

| Разряд | Имя | Описание |
|---|---|---|
| 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.

Указатель конца данных приемника

Область памяти под названием dst_data_end_ptr содержит указатель на последний адрес месторасположения данных приемника. Таблица 416 перечисляет значения разрядов этой области.

Таблица 416 – Значения разрядов dst_data_end_ptr

| Разряд | Имя | Описание |
|---|---|---|
| 31…0 | dst_data_end_ptr | Указатель на последний адрес данных приемника |

Перед тем как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом $2^R$ передачи DMA.

Примечание – Контроллер не имеет доступа по записи в эту область памяти.

### 27.5.1 Разряды управления

Область памяти под названием channel_cfg обеспечивает управление каждой передачей DMA. Таблица 417 показывает название разрядов этой области.

Таблица 417 – 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 418 – Назначение разрядов channel_cfg

| Разряд | Имя | Описание |
|---|---|---|
| 31…30 | dst_src | Шаг инкремента адреса приемника<br>Шаг инкремента адреса зависит от разрядности данных источника<br>Разрядность данных источника = байт<br>2’b00 = байт;<br>2’b01 = полуслово (16-разрядное слово);<br>2’b10 = слово (32-разрядное слово);<br>2’b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr.<br>Разрядность данных источника = полуслово<br>2’b00 = зарезервировано;<br>2’b01 = полуслово;<br>2’b10 = слово;<br>2’b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr.<br>Разрядность данных источника = слово<br>2’b00 = зарезервировано;<br>2’b01 = зарезервировано; |

| Разряд | Имя | Описание |
|---|---|---|
| | | 2’b10 = слово;<br>2’b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr |
| 29…28 | dst_size | Размерность данных приемника.<br>Примечание – Значение этого поля должно быть равно значению поля src_size |
| 27…26 | src_inc | Шаг инкремента адреса источника.<br>Шаг инкремента адреса зависит от разрядности данных источника.<br>Разрядность данных источника = байт:<br>2’b00 = байт;<br>2’b01 = полуслово (16-разрядное слово);<br>2’b10 = слово (32-разрядное слово);<br>2’b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr.<br>Разрядность данных источника = полуслово:<br>2’b00 = зарезервировано;<br>2’b01 = полуслово;<br>2’b10 = слово;<br>2’b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr.<br>Разрядность данных источника = слово:<br>2’b00 = зарезервировано;<br>2’b01 = зарезервировано;<br>2’b10 = слово;<br>2’b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr |
| 25…24 | src_size | Задает размерность данных источника:<br>2’b00 = байт;<br>2’b01 = полуслово (16-разрядное слово);<br>2’b10 = слово (32-разрядное слово);<br>2’b11 = зарезервировано |
| 23…21 | dst_prot_ctrl | Задает состояние HPROT[3:1], когда контроллер записывает данные в приемник.<br>Разряд [23] управляет разрядом HPROT[3]:<br>0 = HPROT[3] в состоянии «0» и доступ не кэшируется;<br>1 = HPROT[3] в состоянии «1» и доступ кэшируется.<br>Разряд [22] управляет разрядом HPROT[2]:<br>0 = HPROT[2] в состоянии «0» и доступ не буферизуется;<br>1 = HPROT[2] в состоянии «1» и доступ буферизуется.<br>Разряд [21] управляет разрядом HPROT[1]:<br>0 = HPROT[1] в состоянии «0» и доступ не привилегированный;<br>1 = HPROT[1] в состоянии «1» и доступ привилегированный |

| Разряд | Имя | Описание |
|---|---|---|
| 20…18 | src_prot_ctrl | Задает состояние HPROT[3:1], когда контроллер считывает данные из источника<br>Разряд [20] управляет разрядом HPROT[3]:<br>0 = HPROT[3] в состоянии «0» и доступ не кэшируется;<br>1 = HPROT[3] в состоянии «1» и доступ кэшируется.<br>Разряд [19] управляет разрядом HPROT[2]:<br>0 = HPROT[2] в состоянии «0» и доступ не буферизуется;<br>1 = HPROT[2] в состоянии «1» и доступ буферизуется.<br>Разряд [18] управляет разрядом HPROT[1]:<br>0 = HPROT[1] в состоянии 0 и доступ не привилегированный;<br>1 = HPROT[1] в состоянии «1» и доступ привилегированный |
| 17…14 | R_power | Задает количество передач DMA до выполнения контроллером процедуры арбитража.<br>Возможные значения:<br>4’b0000 – арбитраж производится после каждой передачи DMA;<br>4’b0001 – арбитраж производится после 2 передач DMA;<br>4’b0010 – арбитраж производится после 4 передач DMA;<br>4’b0011 – арбитраж производится после 8 передач DMA;<br>4’b0100 – арбитраж производится после 16 передач DMA;<br>4’b0101 – арбитраж производится после 32 передач DMA;<br>4’b0110 – арбитраж производится после 64 передач DMA;<br>4’b0111 – арбитраж производится после 128 передач DMA;<br>4’b1000 – арбитраж производится после 256 передач DMA;<br>4’b1001 – арбитраж производится после 512 передач DMA;<br>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.<br>10-разрядное число плюс 1 задает количество передач DMA.<br>Возможные значения:<br>10’b0000000000 = 1 передача DMA;<br>10’b0000000001 = 2 передачи DMA;<br>10’b0000000010 = 3 передачи DMA;<br>10’b0000000011 = 4 передачи DMA;<br>10’b0000000100 = 5 передач DMA;<br>10’b0000000101 = 6 передач DMA;<br>….<br>10’b1111111111 = 1024 передачи DMA.<br>Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить |

| Разряд | Имя | Описание |
|---|---|---|
| | | 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.<br>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].<br>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:<br>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.<br>1 – 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:<br>3’b000 Стоп. Означает, что структура управляющих данных является «неправильной»;<br>3’b001 Основной. Контроллер должен получить новый запрос для окончания цикла DMA, перед этим он должен выполнить процедуру арбитража;<br>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;<br>3’b011 Пинг-понг. Контроллер выполняет цикл DMA использую одну из структур управляющих данных. |

| Разряд | Имя | Описание |
|---|---|---|
| | | 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;<br>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;<br>3’b101 Режим работы с памятью «Исполнение с изменением конфигурации». Смотрите соответствующий раздел. При работе контроллера в данном режиме значение этого поля в альтернативной структуре управляющих данных должно быть 3’b101;<br>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;<br>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 контроллер DMA 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.

#### 27.5.1.1 *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.

Таблица 419 перечисляет адреса приемника цикла DMA для 6 слов.

Таблица 419 – Цикла DMA для шести 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 | | | | |

* Это значение, полученное после сдвига влево значения счетчика на количество разрядов соответствующее dst_inc.

Таблица 420 перечисляет адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.

Таблица 420 – 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 | | | | |

* Это значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc.

** После окончания цикла контроллер DMA делает channel_cfg «неправильным», сбрасывая в «0» поле cycle_ctrl.

## 27.6 Описание регистров контроллера DMA

Данный раздел описывает регистры контроллера и управление контроллером через них.

Раздел содержит следующие сведения:

– о регистровой модели контроллера;

– описание регистров.

Основные положения регистровой модели контроллера:

– нужно избегать адресации при доступе к зарезервированным или неиспользованным адресам, так как это может привести к непредсказуемым результатам;

– необходимо заполнять неиспользуемые или зарезервированные разряды регистров нулями при записи и игнорировать значения таких разрядов при считывании, кроме случаев, специально описанных в разделе;

– системный сброс или сброс по установке питания сбрасывает все регистры в состояние 0, кроме случаев, специально описанных в разделе;

– все регистры поддерживают доступ по чтению и записи, кроме случаев, специально описанных в разделе. Доступ по записи обновляет содержание регистра, а доступ по чтению возвращает содержимое регистра.

Таблица 421 – Перечень регистров контроллера

| Наименование | Смещение относительно базового адреса | Тип | Значение по сбросу | Описание |
|---|---|---|---|---|
| status | 0x000 | RO | 0x-nn0000* | Статусный регистр DMA |
| cfg | 0x004 | WO | - | Регистр конфигурации DMA |
| ctrl_base_ptr | 0x008 | R/W | 0x00000000 | Регистр базового адреса управляющих данных каналов |
| alt_ctrl_base_ptr | 0x00C | RO | 0x000000nn** | Регистр базового адреса альтернативных управляющих данных каналов |
| waitonreq_status | 0x010 | RO | 0x7FFFFFFF | 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

### 27.6.1 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 422 – 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 423 – 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:<br>4’b0000 = контроллер не имеет интегрированной схемы тестирования;<br>4’b0001 = контроллер имеет интегрированную схему тестирования;<br>4’b0010 – 4’b1111 = не определено |
| 27…21 | - | Не определено |
| 20…16 | chnls_minus1 | Количество доступных каналов DMA минус 1.<br>Например:<br>5’b00000 = контроллер имеет 1 канал DMA;<br>5’b00001 = контроллер имеет 2 канала DMA;<br>5’b00010 = контроллер имеет 3 канала DMA;<br>…<br>5’b11111 = контроллер имеет 32 канала DMA |
| 15…8 | - | Не определено |
| 7…4 | state | Текущее состояние автомата управления контроллера.<br>Состояние может быть одним из следующих:<br>4’b0000 = в покое;<br>4’b0001 = чтение управляющих данных канала;<br>4’b0010 = чтение указателя конца данных источника;<br>4’b0011 = чтение указателя конца данных приемника;<br>4’b0100 = чтение данных источника;<br>4’b0101 = запись данных в приемник;<br>4’b0110 = ожидание запроса на выполнение DMA;<br>4’b0111 = запись управляющих данных канала;<br>4’b1000 = приостановлен;<br>4’b1001 = выполнен;<br>4’b1010 = режим работы с периферией «Исполнение с изменением конфигурации»;<br>4’b1011 – 4’b1111 = не определено |
| 3…1 | - | Не определено |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 0 | master_enable | Состояние контроллера:<br>0 = работа контроллера запрещена;<br>1 = работа контроллера разрешена |

### 27.6.2 Регистр конфигурации DMA CFG

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера.

Таблица 424 – 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 425 – 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:<br>Разряд 7 управляет сигналом HPROT[3], с целью индикации о появлении доступа с кэшированием;<br>Разряд 6 управляет сигналом HPROT[2], с целью индикации о появлении доступа с буферизацией;<br>Разряд 5 управляет сигналом HPROT[1], с целью индикации о появлении привилегированного доступа.<br>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:<br>0 – запрещает работу контроллера;<br>1 – разрешает работу контроллера |

### 27.6.3 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 426 – Регистр базового адреса управляющих данных каналов

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

Таблица 427 – 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» |

### 27.6.4 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 428 – Регистр базового адреса альтернативных управляющих данных каналов

| Номер | 31... 0 |
|---|---|
| Доступ | RO |
| Сброс | 0 |
| | Alt_ctrl_base_ptr |

Таблица 429 – 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 |

### 27.6.5 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 430 – 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 431 – 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.<br>При чтении:<br>Разряд [C] = 1 означает, что dma_waitonreq[C] в состоянии «1»;<br>Разряд [C] = 0 означает, что dma_waitonreq[C] в состоянии «0» |

### 27.6.6 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 432 – 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 433 – 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. |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| | | При записи:<br>Разряд [C] = 0 означает, что запрос на выполнение цикла DMA по каналу C не будет установлен;<br>Разряд [C] =1 означает, что запрос на выполнение цикла DMA по каналу C будет установлен.<br>Запись разряда, соответствующего нереализованному каналу, означает, что запрос на выполнение цикла DMA не будет установлен |

### 27.6.7 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 434 – 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 435 – 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.<br>При чтении:<br>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.<br>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.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_clr регистр и установить соответствующий разряд C в 0;<br>Разряд [C] = 1 отключает возможность обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Контроллер выполняет $2^R$ передач.<br>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.

### 27.6.8 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 436 – 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 437 – 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[].<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_useburst_set для отключения обработки запросов от dma_sreq[];<br>Разряд [C] = 1 разрешает обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[].<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта |

### 27.6.9 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 438 – 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_req_mask_set for dma_req [31] and dma_sreq [31] | ..... | chnl_req_mask_set for dma_req [2] and dma_sreq [2] | chnl_req_mask_set for dma_req [1] and dma_sreq [1] | chnl_req_mask_set for dma_req [0] and dma_sreq [0] |

Таблица 439 – 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.<br>При чтении:<br>Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на поступающие запросы;<br>Разряд [C] = 1 означает, что канал DMA C не выполняет циклы DMA в ответ на поступающие запросы.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_req_mask_clr для разрешения установки запросов;<br>Разряд [C] = 1 отключает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[].<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта |

### 27.6.10 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 440 – Регистр очистки маскирования запросов на обслуживание каналов

| Номер | 31 | ..... | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | WO | ..... | WO | WO | WO |
| Сброс | 0 | ..... | 0 | 0 | 0 |
| | chnl_req_mask_clr for dma_req [31] and dma_sreq [31] | ..... | chnl_req_mask_clr for dma_req [2] and dma_sreq [2] | chnl_req_mask_clr for dma_req [1] and dma_sreq [1] | chnl_req_mask_clr for dma_req [0] and dma_sreq [0] |

Таблица 441 – 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.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_req_mask_set для отключения установки запросов;<br>Разряд [C] = 1 разрешает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[].<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта |

### 27.6.11 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 442 – 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 443 – 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.<br>При чтении:<br>Разряд [C] = 0 означает, что канал DMA C отключен;<br>Разряд [C] = 1 означает, что работа канала DMA C разрешена.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_enable_clr для отключения канала;<br>Разряд [C] = 1 разрешает работу канала DMA C.<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта |

### 27.6.12 Регистр сброса разрешения каналов CHNL_ENABLE_CLR

Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов DMA.

Таблица 444 – 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 445 – 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.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_enable_set для разрешения работы канала;<br>Разряд [C] = 1 запрещает работу канала DMA C.<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.<br>Примечание – Контроллер может отключить канал DMA, установив соответствующий разряд в следующих случаях:<br>– при завершении цикла DMA;<br>– при чтении из channel_cfg с полем cycle_ctrl установленным в 3’b000;<br>– при появлении ошибки на шине AHB-Lite |

### 27.6.13 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 446 – 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 447 – 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.<br>При чтении:<br>Разряд [C] = 0 означает, что канал DMA C использует первичную структуру управляющих данных;<br>Разряд [C] = 1 означает, что канал DMA C использует альтернативную структуру управляющих данных.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_pri_alt_clr для сброса разряда [C] в «0»;<br>Разряд [C] = 1 подключает использование альтернативной структуры управляющих данных каналом DMA C.<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.<br><br>Примечание – Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях:<br>– при завершении четырех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»;<br>– при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «Пинг-понг»;<br>– при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах:<br>– «пинг-понг»;<br>– работа с памятью «Исполнение с изменением конфигурации»;<br>– работа с периферией «Исполнение с изменением конфигурации |

### 27.6.14 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 448 – 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 449 – 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.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_pri_alt_set для выбора альтернативных управляющих данных;<br>Разряд [C] = 1 подключает использование первичной структуры управляющих данных каналом DMA C.<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта.<br><br>Примечание – Контроллер может переключить значение разряда chnl_pri_alt_clr[C] в следующих случаях:<br>– при завершении четырех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»;<br>– при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «пинг-понг»;<br>– при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах:<br>– «пинг-понг»;<br>– работа с памятью «Исполнение с изменением конфигурации»;<br>– работа с периферией «Исполнение с изменением конфигурации» |

### 27.6.15 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 450 – 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 451 – 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.<br>При чтении:<br>Разряд [C] = 0 означает, что каналу DMA C присвоен уровень приоритета по умолчанию;<br>Разряд [C] = 1 означает, что каналу DMA C присвоен высокий уровень приоритета.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_priority_clr для установки каналу C уровня приоритета по умолчанию;<br>Разряд [C] = 1 устанавливает каналу DMA C высокий уровень приоритета.<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта; |

### 27.6.16 Регистр сброса приоритета каналов CHNL_PRIORITY_CLR

Данный регистр имеет доступ только на запись. Регистр позволяет присвоить каналу DMA уровень приоритета по умолчанию.

Таблица 452 – 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 453 – 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.<br>При записи:<br>Разряд [C] = 0 не дает эффекта. Необходимо использовать регистр chnl_priority_set для установки каналу C высокого уровня приоритета.<br>Разряд [C] = 1 устанавливает каналу DMA C уровень приоритета по умолчанию.<br>Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. |

### 27.6.17 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 454 – Регистр сброса флага ошибки

| Номер | 31…1 | 0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0 |
| | - | err_clr |

Таблица 455 – 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.<br>При чтении:<br>Разряд [C] = 0 означает, что dma_err находится в состоянии 0;<br>Разряд [C] = 1 означает, что dma_err находится в состоянии 1.<br><br>При записи:<br>Разряд [C] =0 не дает эффекта. Состояние dma_err останется неизменным;<br>Разряд [C] =1 сбрасывает сигнал (флаг) dma_err в состояние 0.<br>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) |

# 28 Контроллер интерфейса Ethernet

При работе блок MAC выполняет две основные функции:

– формирование пакета уровня звена данных протоколов Ethernet/IEEE802.3 и передача его на физический уровень;

– прием с физического уровня и разбор пакета уровня звена данных протоколов Ethernet/IEEE802.3.

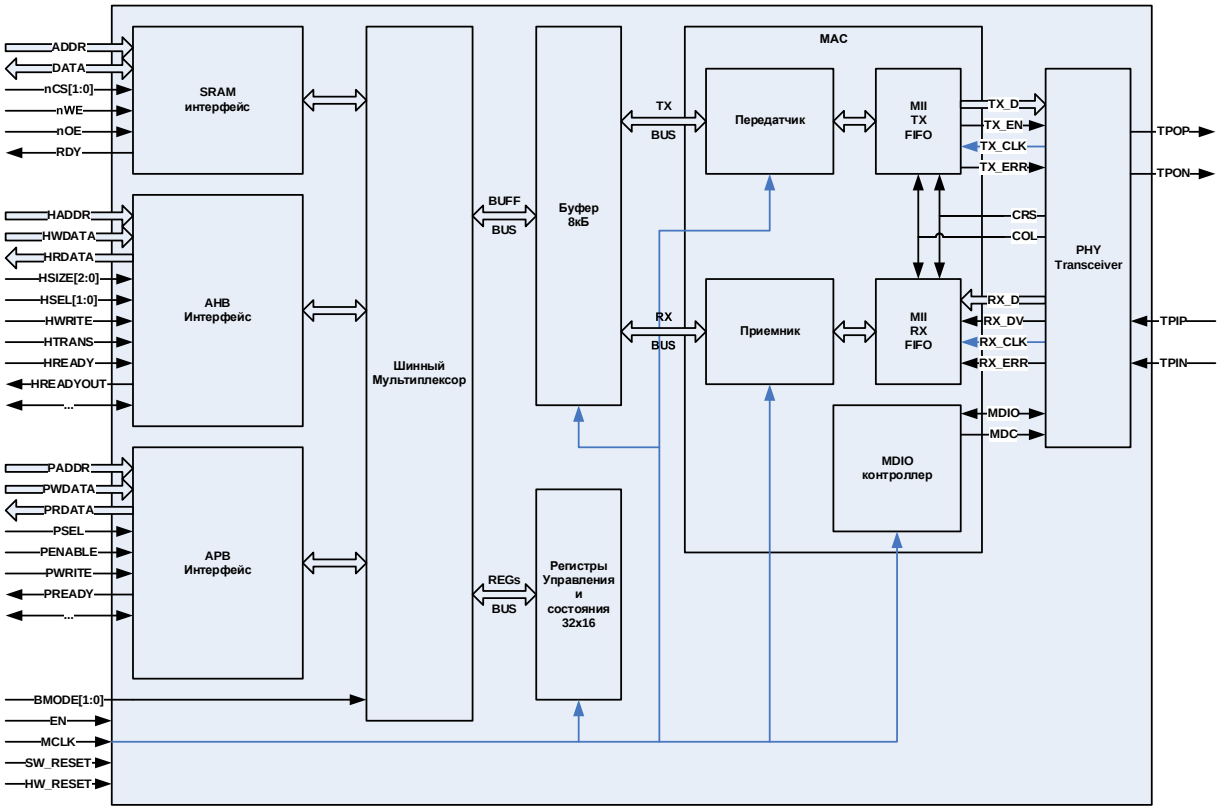



Рисунок 131 – Структурная схема контроллера интерфейса Ethernet

Работа блока возможна в полно- и полудуплексном режимах. Переключение режима осуществляется битом HD_EN регистра G_CFG («1» – полудуплексный режим работы). Также в общее управление блоком входит разрешение обработки пакета PAUSE (бит PAUSE_EN регистра G_CFG), управление размером окна распознаваний коллизий (бит ColWnd регистра G_CFG). И режимом работы буферов приемника и передатчика (бит BUFF_MODE регистра G_CFG).

Режимы работы буферов линейный (BUFF_MODE=2’b00), автоматический (BUFF_MODE=2’01) и режим FIFO (BUFF_MODE=2’10). Режимы различаются способом обработки указателей границ пустых и полных областей буферов приемника и передатчика. В линейном режиме границы начала пустой области буфера приемника (head_R) и передатчика (tail_X) определяются записью в соответствующие регистры управления. В автоматическом режиме эти границы определяются автоматически по адресам последних чтения или записи в соответствующие области. В режиме FIFO границы определяются автоматически, при этом обращение к приемнику производится через адрес 0x0000, а передатчику через адрес 0x0004.

У контроллера Ethernet выводы интерфейса MII доступны по внешним выводам микросхемы, что позволяет подключить внешнюю микросхему PHY-уровня. При этом внутренний PHY должен находиться в состоянии сброса.

## 28.1 Передача пакета

Для передачи пакета(ов) необходима предварительная настройка блока передатчика MAC. Настройка параметров работы передатчика осуществляется посредством регистра управления передатчика X_CFG:

- разрешение работы передатчика (бит EN);
- порядок следования байт в буфере (бит BE);
- порядок следования бит в байте (бит MSB1st);
- управление дополнением пакета до минимальной длины PAD-ами (бит PAD_EN);
- управление дополнением пакета преамбулой (бит PRE_EN), SFD добавляется в любом случае;
- управление дополнением пакета, автоматически подсчитываемым полем CRC (CRC_EN);
- управление интервалом между отправлением пакетов (бит IPG_EN);
- управление максимальным числом повторений (поле RtryCnt).

Далее необходимо записать пакет для передачи в буфер передатчика.

Пакет для передачи содержит три поля (все поля должны быть выровнены по границе слова буферного ОЗУ):

- поле управления передачей пакета
- данные пакета уровня звена данных;
- поле состояния передачи пакета.

Поле управления содержит количество байт пакета в буфере, которые должны быть переданы.

Поле состояния заполняется по завершении процедуры отправки пакета (успешной или нет) и содержит статусную информацию по отправке пакета: о наличии ошибок при его передаче, о количестве попыток передачи пакета и пр.

Если выбран линейный режим работы буферов, то после помещения пакета для передачи необходимо записать в регистр управления tail_X границу пустой области (адрес, следующий за последним словом пакета).

По завершении передачи пакета блок MAC выставит один из флагов прерываний передатчика.

## 28.2 Принцип работы передатчика

Передатчик начинает работать, прочитав ненулевое поле длины из буфера передатчика. Для этого необходимо чтобы буфер передатчика был не пуст (ненулевая разница между значениями head_X и tail_X) и передатчику было разрешено работать (бит EN регистра X_CFG установлен в «1»). Прочитав слово управления, передатчик

перемещает указатель head_X на первое слово пакета данных. При получении управляющего слова в передатчике также фиксируется вся управляющая информация для работы передатчика, препятствуя срыву передачи текущего пакета и позволяя сменить настройки для отправки следующего пакета во время передачи текущего.

По завершении передачи в слово, следующее за последним словом данных, записывается статусная информация отправки пакета.

## 28.3 Прием пакета

Для приема пакета(ов) необходима предварительная настройка блока приемника MAC. Настройка параметров работы приемника осуществляется посредством регистра управления приемника R_CFG:

- разрешение работы приемника (бит EN);
- порядок следования байт в буфере (бит BE);
- порядок следования бит в байте (бит MSB1st);
- управление разрешением приема пакетов:
- длины меньше минимально разрешенной (SF_EN);
- длины больше максимально разрешенной (LF_EN);
- пакетов управления (CF_EN);
- пакетов, содержащих ошибки (EF_EN);
- управление фильтрацией по MAC-адресу:
- разрешение приема пакетов с заданным MAC-адресом (UCA_EN);
- разрешение приема пакетов с широковещательным MAC-адресом (BCA_EN);
- разрешение приема пакетов с групповым MAC-адресом (MCA_EN);
- разрешение приема пакетов с любым MAC-адресом (AC_EN).

Для приема пакета необходимо чтобы в буфере приемника было достаточно пустого места для того чтобы вместить принимаемый пакет.

Принятый пакет содержит два поля (все поля выровнены по границе слова буферного ОЗУ):

- поле состояния приема пакета;
- данные пакета уровня звена данных.

Поле состояния заполняется по успешном завершении процедуры приема пакета и содержит количество байт в пакете (включая заголовок пакета уровня звена дынных), а также статусную информацию по приему пакета, о наличии ошибок при приеме.

Если выбран линейный режим работы буферов, то указание начала свободной для приема данных области указывается в регистре head_R.

По завершении приема пакета блок MAC выставит один из флагов прерываний приемника.

## 28.4 Принцип работы приемника

Приемник начинает работать сразу же после разрешения работы приемника в регистре R_CFG (бит EN установлен в «1»), после обнаружения свободного места в буфере

приемника. Обнаружив наличие свободного места, приемник фиксирует всю управляющую информацию для работы приемника, препятствуя срыву приема изменениями настроек и позволяя сменить настройки для приема следующего пакета, и переходит в режим ожидания данных на входе, после поступления данных – в режим приема. Пакеты, отброшенные по причине ошибок в них или не прошедшие фильтрацию по MAC-адресу, переводят приемник в режим ожидания нового пакета, таким образом, не изменяя общего состояния приемника, лишь изменяя состояние регистра флагов прерываний.

### 28.5 Линейный режим работы буферов

Данный режим включается сбросом поля BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE = 2'b00). В данном режиме все управление границами свободных областей в буферах осуществляется вручную.

### 28.6 Автоматический режим работы буферов

Для включения данного режима необходимо установить значение 1 в поле BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE=2'b01). В данном режиме в буфере автоматически отслеживаются указатели границ достоверных данных для передачи и приема по адресу записи в буфер передатчика и адресу чтения из буфера приемника. В данном режиме нет необходимости ручного управления границей свободного места в приемнике и передатчике через запись в соответствующие регистры. Это позволяет упростить алгоритм запуска передачи и приема и передавать данные одновременно с их помещением в буфер передатчика. Граница достоверных данных в буфере приемника перемещается по завершении приема пакета и, таким образом, данный режим не допускает одновременного приема пакета и его чтения из буфера. Для обеспечения корректной работы в этом режиме необходимо активировать биты DBG_XF_EN и DBG_RF_EN регистра G_CFG.

### 28.7 Режим FIFO работы буферов

Режим FIFO отличается от предыдущих полностью автоматическим отслеживанием данных в буфере. В данный режим модуль переводится установкой значения 2 в поле BUFF_MODE регистра G_CFG (G_CFG.BUFF_MODE = 2'b10). В данном режиме чтение/запись в буферы статусной и управляющей информации, а также данных осуществляются через один адрес – 0x0000 для приемника и 0x0004 для передатчика. Для обеспечения корректной работы в этом режиме необходимо активировать биты DBG_XF_EN и DBG_RF_EN регистра G_CFG. Данный режим позволяет работать на максимальной скорости, если инструментальные средства управляющего контроллера не обеспечивают режимы адресации с автоинкрементом и циклической буферизацией.

### 28.8 Прерывания

Прием и передача пакетов сопровождаются не только формированием событий, но и формированием прерываний, которые служат для непосредственного отражения

оперативной статусной информации о состоянии Ethernet-контроллера. Вывод прерывания один, он обеспечивает общую индикацию наличия флагов в регистре флагов прерываний (IFR), разрешенных регистром маски прерываний (IMR). Все прерывания маскируемые. «1» в бите регистра маски прерываний (IMR) разрешает соответствующее прерывание, 0 – запрещает соответствующее прерывание.

Прерывания делятся на три группы:

- прерывания интерфейса MDIO;
- прерывания передатчика;
- прерывания приемника.

Прерывания интерфейса MDIO информируют о завершении затребованной операции по интерфейсу MDIO.

Прерывания передатчика показывают состояние отправки пакетов, включая информацию об успешной отправке или наличие ошибок.

Прерывания приемника отражают состояние приема пакета, включая информацию о приеме пакета без ошибок или наличие ошибок при приеме.

Все флаги прерываний кумулятивные. Сброс флагов производится чтением регистра, если бит RCLR_EN регистра G_CFGl установлен в «1», или записью «1» в соответствующий разряд регистра IFR.

### 28.9 Режим детерминированного времени доставки

Данный режим является расширением стандарта IEEE 802.3/Ethernet для обеспечения детерминированного времени доставки. Режим включается установкой бита DTRM_EN регистра G_CFGl в «1». Данный режим может использоваться только в полнодуплексном режиме работы (значение бита HD_EN = 1 регистра G_CFGl блокирует данный режим).

В данном режиме для начала передачи пакета выделяется интервал, размером, задаваемым регистром JitterWnd (размер окна = JitterWnd + 1), с периодом задаваемым регистром BAG (период = BAG + 1). Единица измерения периода и размера джиттера задается регистром PSC в тактах основной частоты работы блока (размер единицы = PSC + 1).

### 28.10 Режим КЗ

В блоке MAC для целей тестирования алгоритмов обработки данных предусмотрен режим короткого замыкания (КЗ). В данном режиме выход передатчика переключается на вход приемника. Также в данном режиме блок принудительно переводится в полнодуплексный режим работы.

### 28.11 Регистры

Таблица 456 – Регистры контроллера интерфейса Ethernet

| Базовый адрес | Название | | Описание |
|---|---|---|---|
| 0x38000000 | Ethernet | | Буфер данных контроллера интерфейса Ethernet |
| 0x30000000 | Ethernet | | Регистры контроллера интерфейса Ethernet |
| Смещение (в байтах) | Название | Доступ и значение по сбросу | Описание |
| 0x00 | Delimiter | RW, 0x1000 | Регистр границы буферов приемника и передатчика (размер буфера приемника/начало буфера передатчика) |
| 0x02 | MAC-Address | | Регистр индивидуального MAC-адреса |
| 0x02 | MAC_T | RW, 0x78AB | Младшая* часть индивидуального MAC-адреса |
| 0x04 | MAC_M | RW, 0x3456 | Средняя часть индивидуального MAC-адреса |
| 0x06 | MAC_H | RW, 0x0012 | Старшая часть индивидуального MAC-адреса |
| 0x08 | HASH | | HASH-таблица групповых адресов |
| 0x08 | HASH0 | RW, 0x0000 | Младшая часть HASH-таблицы |
| 0x0A | HASH1 | RW, 0x0000 | Средняя часть HASH-таблицы |
| 0x0C | HASH2 | RW, 0x0000 | Средняя часть HASH-таблицы |
| 0x0E | HASH3 | RW, 0x8000 | Старшая часть HASH-таблицы |
| 0x10 | IPG | RW, 0x0060 | Регистр задания межпакетного интервала для полнодуплекного режима (в битовых интервалах) |
| 0x12 | PSC | RW, 0x0031 | Регистр задания предделителя шага изменения значений BAG и JitterWnd (1 мкс при частоте 50 МГц) |
| 0x14 | BAG | RW, 0x0063 | Регистр задания периода следования пакетов (100 мкс при частоте 50 МГц) |
| 0x16 | JitterWnd | RW, 0x0004 | Регистр задания джиттера при передаче пакетов (5 мкс при частоте 50 МГц) |
| 0x18 | R_CFG | RW, 0x0507 | Регистр управления приемника |
| 0x1A | X_CFG | RW, 0x01FA | Регистр управления передатчика |
| 0x1C | G_CFGl | RW, 0x4880 | Регистр общего управления блоком, младшее полуслово |
| 0x1E | G_CFGh | RW, 0x3000 | Регистр общего управления блоком, старшее полуслово |
| 0x20 | IMR | RW, 0x0000 | Регистр маски прерываний |
| 0x22 | IFR | RW, 0x0000 | Регистр флагов прерываний |
| 0x24 | MDIO_CTRL | RW, 0x0000 | Регистр управления канала MDIO интерфейса MII |
| 0x26 | MDIO_DATA | RW, 0x0000 | Регистр данных канала MDIO интерфейса MII |
| 0x28 | R_Head | RW, 0x0000 | Указатель начала области действительных данных приемника (указывает на первое непустое слово) |

| Базовый адрес | Название | | Описание |
|---|---|---|---|
| 0x38000000 | Ethernet | | Буфер данных контроллера интерфейса Ethernet |
| 0x30000000 | Ethernet | | Регистры контроллера интерфейса Ethernet |
| Смещение (в байтах) | Название | Доступ и значение по сбросу | Описание |
| 0x2A | X_Tail | RW, 0x0800 | Указатель конца области действительных данных передатчика (указывает на первое пустое слово) |
| 0x2C | R_Tail | R, 0x0000 | Указатель конца области действительных данных приемника (указывает на первое пустое слово) |
| 0x2E | X_Head | R, 0x0800 | Указатель начала области действительных данных передатчика (указывает на первое непустое слово) |
| 0x30 | STAT | RW, 0x0303 | Регистр статуса |
| * При значениях регистров MAC_T, MAC_M, MAC_H по умолчанию будут приниматься пакеты, начинающиеся с: 0xAB, 0x78, 0x56, 0x34, 0x12, 0x00… (где 0xAB - первый байт MAC адреса назначения пакета) | | | |

### 28.11.1 Поле управления передачи пакета

Таблица 457 – Поле управления передачи пакета

| Номер | 31…16 | 15…0 |
|---|---|---|
| Доступ | U | R/W |
| | - | Length[15:0] |

Таблица 458 – Описание бит поля управления передачи пакета

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15…0 | Length[15:0] | Количество байт в пакете |

### 28.11.2 Поле состояния передачи пакета

Таблица 459 – Поле состояния передачи пакета

| Номер | 31…23 | 22 | 21 | 20 | 19…16 | 15…0 |
|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | U |
| | - | UR | LC | RL | RCOUNT | - |

Таблица 460 – Описание бит поля состояния передачи пакета

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…23 | - | Зарезервировано |
| 22 | UR | Флаг опустошения буфера передатчика<br>1 – буфер передатчика пуст;<br>0 – буфер передатчика не пуст |
| 21 | LC | Флаг индикации Late collision во время передачи пакета<br>1 – произошла Late collision во время передачи пакета;<br>0 – Late collision во время передачи пакета не происходила |
| 20 | RL | Флаг исчерпания попыток передачи пакета<br>1 – превышено разрешенное количество попыток передачи пакета;<br>0 – количество попыток передачи пакета не превысило разрешенного значения |
| 19…16 | RCOUNT[3:0] | Число попыток передачи пакета |
| 15…0 | - | Зарезервировано |

### 28.11.3 Поле состояния приема пакета

Таблица 461 – Поле состояния приема пакета

| Номер | 31…27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | - | UCA | BCA | MCA | SMB_ERR | CRC_ERR | DN_ERR | LEN_ERR |

| Номер | 19 | 18 | 17 | 16 | 15…0 |
|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W |
| | SF_ERR | LF_ERR | CF_ERR | PF_ERR | Length[15:0] |

Таблица 462 – Описание бит поля состояния приема пакета

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31…27 | - | Зарезервировано |
| 26 | UCA | Признак индивидуального пакета (MAC-адрес соответствует установленному)<br>1 – MAC-адрес принятого пакета совпадает с MAC-адресом Ethernet-контроллера;<br>0 – MAC-адрес принятого пакета не совпадает с MAC-адресом Ethernet-контроллера |
| 25 | BCA | Признак широковещательного пакета (MAC = FF_FF_FF)<br>1 – принят широковещательный пакет;<br>0 – широковещательный пакет не принят |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 24 | MCA | Признак группового пакета (MAC соответствует HASH)<br>1 – принят пакет, удовлетворяющий фильтрации по HASH-таблице;<br>0 – принятый пакет не удовлетворяет фильтрации по HASH-таблице или фильтрация отключена |
| 23 | SMB_ERR | Признак наличия в пакете ошибочных nibbles<br>1 – наличие<br>0 – отсутствие |
| 22 | CRC_ERR | Признак несоответствия CRC пакета<br>1 – произошла ошибка сравнения CRC-пакета с вычисленной CRC;<br>0 – CRC-пакета и вычисленной CRC совпадают |
| 21 | DN_ERR | Количество бит в пакете не кратно 8<br>1 – не кратно 8<br>0 – кратно 8 |
| 20 | LEN_ERR | Признак несоответствия между реальной длиной и длинной указанной в поле длины – 13,14 октеты<br>1 – несоответствие<br>0 – соответствие |
| 19 | SF_ERR | Признак недостаточной длины пакета 64 октетов<br>1 – ошибочная длина<br>0 – корректная длина |
| 18 | LF_ERR | Признак превышение длины пакета 1518 октетов<br>1 – превышение<br>0 – норма |
| 17 | CF_ERR | Признак пакета управления (фильтрация по специальным MAC и тэгам в поле длины – 13, 14 – октеты)<br>1 – пакет управления<br>0 – другой пакет |
| 16 | PF_ERR | Признак пакета PAUSE<br>1 – пакет PAUSE<br>0 – другой пакет |
| 15…0 | Length[15:0] | Количество байт в пакете, включая заголовок и CRC |

### 28.11.4 G_CFGh

Таблица 463 – Регистр G_CFGh

| Номер | 15 | 14 | 13 | 12 | 11…3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W,+0 | R/W,+0 | R/W,+1 | R/W,+1 | U | R/W,+0 | R/W,+0 | R/W,+0 |
| | DBG_mode | | DBG_XF_EN | DBG_RF_EN | - | DLB | RRST | XRST |

Таблица 464 – Описание бит регистра G_CFGh

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 15…14 | DBG_mode | Зарезервировано |
| 13 | DBG_XF_EN | Разрешение автоматического изменения указателей FIFO передатчика.<br>0 – запрещено;<br>1 – разрешено |
| 12 | DBG_RF_EN | Разрешение автоматического изменения указателей FIFO приемника.<br>0 – запрещено;<br>1 – разрешено |
| 11…3 | | Зарезервировано |
| 2 | DLB | Режим КЗ.<br>0 – выключен;<br>1 – включен |
| 1 | RRST | Сброс приемника.<br>0 – работает;<br>1 – сброшен |
| 0 | XRST | Сброс передатчика.<br>0 – работает;<br>1 – сброшен |

### 28.11.5 G_CFGl

Таблица 465 – Регистр G_CFGI

| Номер | 15 | 14 | 13…12 | 11 | 10 | 9 | 8 | 7…0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/W,+0 | R/W,+0 | R/W,+1 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 |
| | - | RCLR_EN | BUFF_MODE | EXT_EN | HD_EN | DTRM_EN | PAUSE_EN | ColWnd |

Таблица 466 – Описание бит регистра G_CFGl

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 15 | | Зарезервировано |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 14 | RCLR_EN | Сброс регистров статуса(IFR)<br>0 – производится записью в регистры статуса;<br>1 – регистры статуса сбрасываются при чтении |
| 13…12 | BUFF_MODE | Режим работы буфера.<br>2’b00 – линейный режим;<br>2’b 01 – режим с автоматическим изменением указателей;<br>2’b 10 – режим FIFO;<br>2’b 11 – зарезервировано (линейный режим) |
| 11 | EXT_EN | Включение режима дополнения коротких пакетов до размера slotTime полем «Extension» (При приеме отбрасывание слова осуществляется по полю length пакета, если оно отражает длину пакета).<br>0 – выключен;<br>1 – включен |
| 10 | HD_EN | Полудуплексный режим работы.<br>0 – выключен;<br>1 – включен |
| 9 | DTRM_EN | Режим детерминированного времени доставки.<br>0 – выключен<br>1 – включен |
| 8 | PAUSE_EN | Режим автоматической обработки пакета PAUSE.<br>0 – выключен;<br>1 – включен |
| 7…0 | ColWnd[7:0] | Размер «окна коллизий». Для распознавания «легальной» коллизии, вызванной задержкой распространения и коллизии, вызванной проблемами в самой сети.<br>(в битовых интервалах x4) |

### 28.11.6 X_CFG

Таблица 467 – Регистр X_CFG

| Номер | 15 | 14 | 13 | 12 | 11…8 |
|---|---|---|---|---|---|
| Доступ | R/W,+1 | U | R/W,+0 | R/W,+0 | U |
| | EN | - | BE | MSB1st | - |

| Номер | 7 | 6 | 5 | 4 | 3…0 |
|---|---|---|---|---|---|
| Доступ | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 |
| | PAD_EN | PRE_EN | CRC_EN | IPG_EN | RtryCnt |

Таблица 468 – Описание бит регистра X_CFG

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 15 | EN | Разрешение работы передатчика.<br>0 – остановлен;<br>1 – разрешена работа |
| 14 | | Зарезервировано |
| 13 | BE | Порядок следования байт в слове передатчика.<br>0 – LittleEndian;<br>1 – BigEndian |
| 12 | MSB1st | Порядок следования бит при передаче байтов данных.<br>0 – первым передается LSB;<br>1 – первым передается MSB |
| 11…8 | | Зарезервировано |
| 7 | PAD_EN | Дополнение пакета до минимальной длины PAD-ами.<br>0 – выключено;<br>1 – включено |
| 6 | PRE_EN | Дополнение пакета преамбулой.<br>0 – выключено;<br>1 – включено |
| 5 | CRC_EN | Дополнение пакета автоматически вычисленным CRC.<br>0 – выключено;<br>1 – включено |
| 4 | IPG_EN | Режим выдержки паузы между отправкой пакетов.<br>0 – выключен;<br>1 – включен |
| 3…0 | RtryCnt[3:0] | Максимальное количество попыток отправки пакета |
| Примечание – Для применения новых управляющих настроек передатчика необходимо произвести сброс передатчика путем последовательного выполнения операций установки и сброса бита XRST в регистре G_CFGh | | |

### 28.11.7 R_CFG

Таблица 469 – Регистр R_CFG

| Номер | 15 | 14 | 13 | 12 | 11…8 |
|---|---|---|---|---|---|
| Доступ | R/W,+1 | U | R/W,+0 | R/W,+0 | U |
| | EN | - | BE | MSB1st | - |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 |
| | SF_EN | LF_EN | CF_EN | EF_EN | AC_EN | UCA_EN | BCA_EN | MCA_EN |

Таблица 470 – Описание бит регистра R_CFG

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 15 | EN | Разрешение работы приемника:<br>0 – приемник остановлен;<br>1 – разрешена работа |
| 14 | | Зарезервировано |
| 13 | BE | Порядок следования байт в слове:<br>0 – LittleEndian;<br>1 – BigEndian |
| 12 | MSB1st | Порядок следования бит при приеме байтов данных:<br>0 – первым принимается LSB;<br>1 – первым принимается MSB |
| 11…8 | | Зарезервировано |
| 7 | SF_EN | Разрешение приема пакетов длиной меньше минимальной:<br>0 – выключено;<br>1 – включено |
| 6 | LF_EN | Разрешение приема пакетов длиной больше максимальной:<br>0 – выключено;<br>1 – включено |
| 5 | CF_EN | Разрешение приема управляющих пакетов:<br>0 – выключено;<br>1 – включено |
| 4 | EF_EN | Разрешение приема пакетов с ошибками:<br>0 – выключено;<br>1 – включено |
| 3 | AC_EN | Прием пакетов без фильтрации MAC-адреса:<br>0 – выключен;<br>1 – включен |
| 2 | UCA_EN | Прием пакетов с MAC-адресом, указанным в регистре MAC_Address.<br>0 – выключен;<br>1 – включен |
| 1 | BCA_EN | Прием пакетов с широковещательным MAC-адресом:<br>0 – выключен;<br>1 – включен |
| 0 | MCA_EN | Прием пакетов с групповым MAC-адресом с фильтрацией по HASH-таблице:<br>0 – выключен;<br>1 – включен |
| Примечание – Для применения новых управляющих настроек приемника необходимо произвести сброс приемника путем последовательного выполнения операций установки и сброса бита RRST в регистре G_CFGh | | |

### 28.11.8 IMR/IFR

Таблица 471 – Регистр IMR/IFR

| Номер | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R,+0 | R,+0 | | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 |
| | MII_RDY | MDIO_INT | - | CRS_LOST | LC | UNDF | XF_ERR | XF_OK |

| Номер | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 |
| | SF | LF | CF | CRC_ERR | SMB_ERR | OVF | MISSED_F | RF_OK |

Таблица 472 – Описание бит регистра IMR/IFR

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 15 | MII_RDY | Индикатор завершения текущей команды обмена по интерфейсу MDIO |
| 14 | MDIO_INT | Индикатор наличия прерывания по интерфейсу MDIO |
| 13 | | Зарезервировано |
| 12 | CRS_LOST | Индикатор потери несущей во время передачи в полудуплексном режиме работы |
| 11 | LC | Индикатор наличия LateCollision в линии |
| 10 | UNDF | Индикатор опустошения буфера передатчика |
| 9 | XF_ERR | Индикатор наличия ошибок при передаче пакета |
| 8 | XF_OK | Индикатор успешной отправки пакета |
| 7 | SF | Индикатор приема пакета длинной меньше минимальной |
| 6 | LF | Индикатор приема пакета длинной больше максимальной |
| 5 | CF | Индикатор приема управляющих пакетов |
| 4 | CRC_ERR | Индикатор наличия несовпадения пакета CRC принятых данных с пакета CRC |
| 3 | SMB_ERR | Индикатор наличия ошибок в данных при приеме пакета |
| 2 | OVF | Индикатор переполнения буфера приемника |
| 1 | MISSED_F | Индикатор потери пакета из-за отсутствия места в буфере приемника |
| 0 | RF_OK | Индикатор успешно принятого пакета |
| Примечание – Индикатор в состоянии единицы означает наличие события, в нуле отсутствие события | | |

### 28.11.9 STAT

Таблица 473 – Регистр STAT

| Номер | 15…13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Доступ | | R,+0 | R,+0 | R,+0 | R,+0 | R,+0 |
| | - | X_FULL | X_AFULL | X_HALF | X_AEMPY | X_EMPY |

| Номер | 7…5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Доступ | | R,+0 | R,+0 | R,+0 | R,+0 | R,+0 |
| | RCOUNT | R_FULL | R_AFULL | R_HALF | R_AEMPY | R_EMPY |

Таблица 474 – Описание бит регистра STAT

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 15…13 | - | Зарезервировано |
| 12 | X_FULL | 1 – буфер передатчика полон<br>0 – буфер передатчика не полон |
| 11 | X_AFULL | 1 – буфер передатчика почти полон<br>0 – буфер передатчика не в состоянии почти полон |
| 10 | X_HALF | 1 – буфер передатчика полуполон<br>0 – буфер передатчика не полуполон |
| 9 | X_AEMPTY | 1 – буфер передатчика почти пуст<br>0 – буфер передатчика не в состоянии почти пуст |
| 8 | X_EMPTY | 1 – буфер передатчика пуст<br>0 – буфер передатчика не пуст |
| 7…5 | R_COUNT | Количество принятых, но не считанных пакетов:<br>0-6 – количество пакетов;<br>7 – количество несчитанных пакетов ≥7.<br>Инкрементируется автоматически при получении нового пакета. Декрементировать необходимо программным путем после считывания пакета. Запись любого значения декрементирует счетчик на 1 |
| 4 | R_FULL | 1 – буфер приемника полон<br>0 – буфер приемника не полон |
| 3 | R_AFULL | 1 – буфер приемника почти полон<br>0 – буфер приемника не в состоянии почти полон |
| 2 | R_HALF | 1 – буфер приемника полуполон<br>0 – буфер приемника не полуполон |
| 1 | R_AEMPTY | 1 – буфер приемника почти пуст<br>0 – буфер приемника не в состоянии почти пуст |
| 0 | R_EMPTY | 1 – буфер приемника пуст;<br>0 – начало приема пакета.<br>Бит сбрасывается в "0" при начале приема пакета и при отсутствии данных в буфере |

### 28.11.10 MDIO_CTRL

Таблица 475 – Регистр MDIO_CTRL

| Номер | 15 | 14 | 13 | 12…8 | 7…5 | 4…0 |
|---|---|---|---|---|---|---|
| Доступ | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 | R/W,+0 |
| | RDY | PRE_EN | OP | PHY_A | DIV | RG_A |

Таблица 476 – Описание бит регистра MDIO_CTRL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 15 | RDY | Управление/индикатор обмена по MDIO<br>После записи команды необходимо установить в «1» для инициирования исполнения команды в регистре MDIO_CTRL после одного такта сбрасывается в ноль и снова устанавливается в единицу после завершения цикла обмена по интерфейсу MDIO |
| 14 | PRE_EN | Режим передачи:<br>1 – с передачей преамбулы (32 бита «1»);<br>0 – без передачи преамбулы |
| 13 | OP | Операция:<br>1 – чтение;<br>0 – запись |
| 12…8 | PHY_A[4:0] | Адрес модуля PHY |
| 7…5 | DIV | Коэффициент деления основной частоты для работы интерфейса MDIO<br>MDC = ETH_CLK / [(DIV + 1) • 16]<br>Частота MDC должна быть не более 2,5 МГц |
| 4…0 | RG_A | Номер регистра PHY |

## 28.12 Блок PHY

Блок PHY реализует физический уровень протоколов Ethernet/IEEE 802.3. Он может функционировать в одном из следующих режимов:

- 10Base-T FD (full duplex);
- 10Base-T HD (half duplex);
- 100Base-T FD (full duplex);
- 100Base-T HD (half duplex);
- 100Base-FX.

Данный блок обладает автоподстройкой (Auto-negotiation) параметров линии и обеспечивает автоматическое их определение для выбора режима работы с наибольшей пропускной способностью.

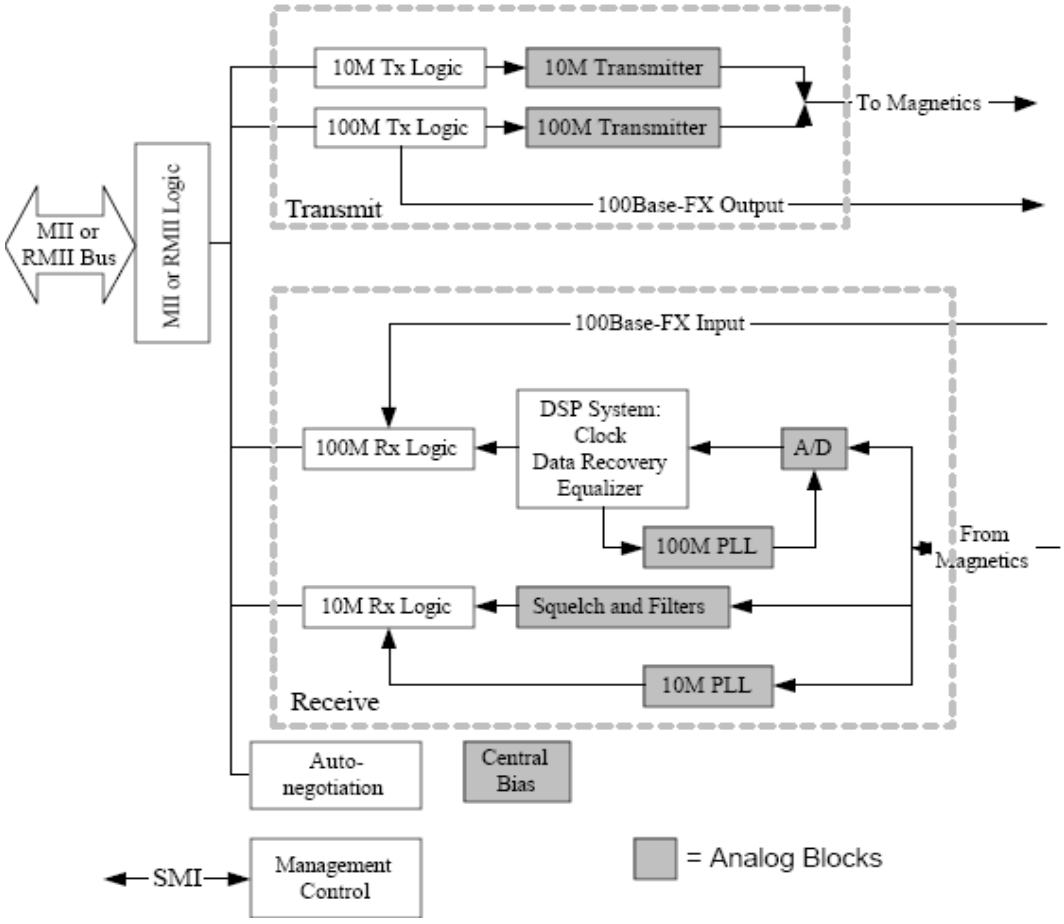


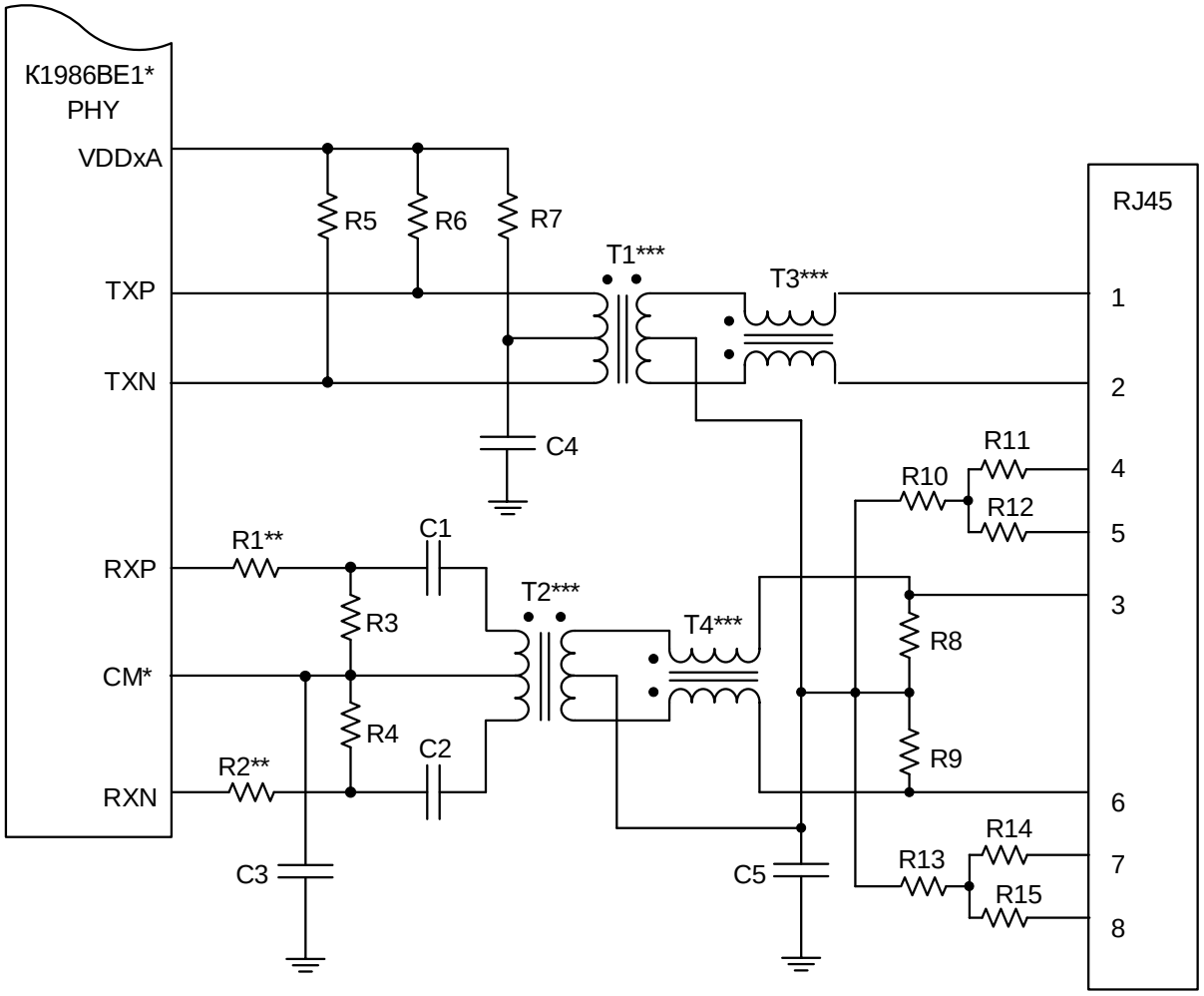

Рисунок 132 – Структурная схема модуля PHY

Блок PHY обеспечивает простое подключение к линии с использованием небольшого количества внешних элементов.

Типовая схема подключения микросхем с функцией Ethernet приведена на рисунке 133.

Типовая схема подключения микросхем без использования функции Ethernet приведена на рисунке 134.

Частота тактирования блока PHY должна быть 25 МГц с джиттером менее 100 пс и длительностью фронтов менее 3 нс.



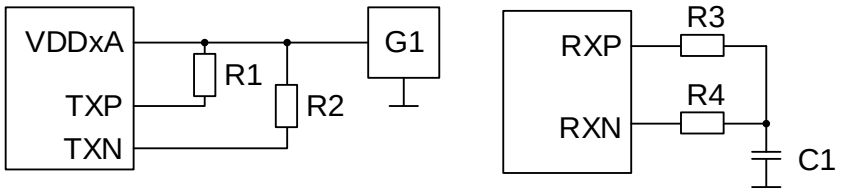

* В микросхемах К1986ВЕ1FI вывод CM подключается согласно схеме. В микросхемах К1986ВЕ1GI вывод CM отсутствует, остальные подключения выполняются согласно схеме.

** Используется только при тестировании.

*** Рекомендованные трансформаторы – Bel Fuse S558-5999-46-F, Pulse-H1102, Pulse-H1089 или аналогичные (расположение катушек и/или ответвлений от средней точки обмотки может отличаться от приведенных на рисунке).

| Компонент | Номинал | Компонент | Номинал |
|---|---|---|---|
| R1 | 0 Ом | R11 | 50 Ом ± 1% |
| R2 | 0 Ом | R12 | 50 Ом ± 1% |
| R3 | 82 Ом ± 1% | R13 | 50 Ом ± 1% |
| R4 | 82 Ом ± 1% | R14 | 50 Ом ± 1% |
| R5 | 50 Ом ± 1% | R15 | 50 Ом ± 1% |
| R6 | 50 Ом ± 1% | C1 | 6,8 нФ ± 10% |
| R7 | 10 Ом ± 1% | C2 | 6,8 нФ ± 10% |
| R8 | 150 Ом ± 1% | C3 | 10 нФ ± 20% |
| R9 | 150 Ом ± 1% | C4 | 22 нФ ± 20% |
| R10 | 50 Ом ± 1% | C5 | 10 нФ, 2 кВ |

Рисунок 133 – Типовая схема подключения к линии Ethernet/IEEE 802.3




C1 – конденсатор, емкостью 0,01 мкФ ± 10 %;
G1 – источник напряжения питания, от 3,0 до 3,6 В;
R1, R2 – резисторы сопротивлением 51 Ом ± 5 %;
R3, R4 – резисторы сопротивлением 82 Ом ± 5 %.

Рисунок 134 – Типовая схема подключения выводов микросхем без использования функции Ethernet

Управление режимами блока осуществляется через регистры PHY_CTRL и PHY_STAT блока MAC.

При этом через регистр PHY_CTRL осуществляется программный сброс блока, а также настройка режимов его работы после сброса.

В регистре PHY_STAT отражается информация о текущем состоянии блока PHY.

Примечание – После аппаратного сброса необходимо выдержать паузу 16 мс для выхода блока PHY в рабочий режим.

### 28.12.1 Регистры

Таблица 477 – Регистры блока PHY

| Базовый адрес | Название | | Описание |
|---|---|---|---|
| 0x30000000 | Ethernet | | Контроллер интерфейса Ethernet |
| Смещение (в байтах) | Название | Доступ и значение по умолчанию | Описание |
| 0x34 | PHY_Control | 0 | Регистр управления PHY |
| 0x36 | PHY_Status | 0xf | Регистр флагов статуса PHY |

### 28.12.2 PHY_Control

Таблица 478 – Регистр PHY_Control

| Номер | 15…11 | 10 | 9 | 8 | 7 | 6…4 | 3…1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W | | R/W | R/W |
| | PHYADD | MDC | MDIO_SEL | MDI | FX_EN | - | MODE | nRST |

Таблица 479 – Описание бит регистра PHY_Control

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 15…11 | PHYADD[4:0] | Адрес PHY, используемый для интерфейса MII и для инициализации скрамблера |
| 10 | MDC | Тактовый сигнал обмена через MII блока PHY (для ручного управления работой через MII) |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 9 | MDIO_SEL | Выбор режима управления обмена данными через интерфейс MII:<br>1 – ручное управление работой через MII;<br>0 – обмен через MII осуществляется через регистры и автомат, встроенные в блок MAC |
| 8 | MDI | Состояние входа данных MII блока PHY<br>(для ручного управления работой через MII) |
| 7 | FX_EN | Выбор режима работы блока PHY 100BaseFX.<br>1 – включен режим 100BaseFX;<br>0 – режим 100BaseFX выключен |
| 6…4 | - | Зарезервировано |
| 3…1 | MODE[2:0] | Режим работы блока PHY:<br>3'b000 – 10BaseT HD без автоподстройки;<br>3'b001 – 10BaseT FD без автоподстройки;<br>3'b010 – 100BaseT HD без автоподстройки;<br>3'b011 – 100BaseT FD без автоподстройки;<br>3'b100 – 100BaseT HD с автоподстройкой;<br>3'b101 – режим повторителя;<br>3'b110 – режим пониженного потребления;<br>3'b111 – полностью автоматический режим.<br>Примечание – Режимы без автоподстройки использовать только в нормальных климатических условиях |
| 0 | nRST | Разрешение работы блока PHY:<br>0 – блок PHY сброшен;<br>1 – блок PHY в штатном режиме |

Примечания

1 Для применения новых управляющих настроек блока PHY необходимо произвести сброс блока PHY с помощью бита nRST в регистре PHY_Control. Бит nRST необходимо удерживать в состоянии «0» (блок PHY сброшен) не менее 100 мкс;

2 В режиме работы 10BaseT HD блок PHY дополнительно возвращает переданные данные обратно блоку MAC. В полудуплексном режиме работы MAC такие пакеты отбрасываются. Для приема переданных данных требуется выключить полудуплексный режим работы MAC (бит HD_EN, регистр G_CFGl), а также разрешить прием пакетов с ошибками (бит EF_EN, регистр R_CFG), так как данные принимаются с ошибкой CRC

### 28.12.3 PHY_Status

Таблица 480 – Регистр PHY_Status

| Номер | 15…11 | 10 | 9 | 8 | 7…6 | 5 | 4 | 3…0 |
|---|---|---|---|---|---|---|---|---|
| Доступ | | RO | RO | RO | RO | RO | RO | RO |
| | - | MDINT | MDO | FX_VALID | COL | CRS | READY | LED[3:0] |

Таблица 481 – Описание бит регистра PHY_Status

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений. |
|---|---|---|
| 15…11 | - | Зарезервировано |
| 10 | MDINT | Флаг запроса прерывания от блока PHY:<br>1 – имеется прерывание от блока PHY;<br>0 – от блока PHY прерывания отсутствуют<br>(дублируется в регистре прерываний блока MAC). |
| 9 | MDO | Состояние выхода данных MII блока PHY<br>(для ручного управления работой через MII) |
| 8 | FX_VALID | Флаг наличия обмена данными в оптоволоконной линии:<br>1 – присутствует обмен в линии FX;<br>0 – линия FX в исходном состоянии |
| 7...6 | COL | Флаг наличия коллизии в линии:<br>1 – в линии присутствует коллизия;<br>0 – в линии коллизия отсутствует. |
| 5 | CRS | Флаг наличия обмена данными по витой паре:<br>1 – в линии идет обмен данными;<br>0 – линия в исходном состоянии |
| 4 | READY | Флаг готовности к работе блока PHY:<br>1 – блок PHY вышел в рабочий режим после аппаратного сброса/отключения;<br>0 – блок PHY не в рабочем режиме |
| 3 | LED3 | Индикация режима работы блока PHY:<br>0 – режим работы full-duplex;<br>1 – режим работы half-duplex |
| 2 | LED2 | Индикация наличия Carrier sense:<br>0 – наличие Carrier sense (CRS);<br>1 – отсутствие Carrier sense (CRS) |
| 1 | LED1 | Индикация наличия Link сигнала:<br>0 – сигнал Link включен;<br>1 – сигнал Link выключен |
| 0 | LED0 | Индикация выбранной скорости обмена данными:<br>0 – выбрана скорость 100 Мбит;<br>1 – выбрана скорость 10 Мбит |

Для доступа к внутренним регистрам блока PHY предназначены регистры MDIO_CTRL и MDIO_DATA блока MAC.

В таблице 482 приведен перечень и описание внутренних регистров блока PHY.

Таблица 482 – Внутренние регистры блока PHY

| Регистр | Описание | Группа |
|---|---|---|
| 0 | Основной регистр управления | основные |
| 1 | Основной регистр состояния | основные |
| 2 | Регистр идентификатора PHY 1 | расширенные |

| Регистр | Описание | Группа |
|---|---|---|
| 3 | Регистр идентификатора PHY 2 | расширенные |
| 4 | Регистр рекомендаций автоподстройки | расширенные |
| 5 | Регистр возможностей оппонента по автоподстройке | расширенные |
| 6 | Регистр расширенного управления автоподстройкой | расширенные |
| 18 | Регистр расширенного управления режимами | производителя |
| 29 | Регистр флагов прерываний | производителя |
| 30 | Регистр маски прерываний | производителя |
| 31 | Регистр расширенного управления и состояния PHY | производителя |

### 28.12.4 Основной регистр управления (0)

Таблица 483 – Основной регистр управления (0)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15 | Reset | Программный сброс блока PHY.<br>1 – программный сброс.<br>Самоочищаемый.<br>Рекомендуется не изменять остальные биты данного регистра во время его установки | RW/SC | 0 |
| 14 | Loopback | Режим КЗ блока PHY:<br>1 – режим КЗ;<br>0 – штатный режим. | RW | 0 |
| 13 | Speed Select | Выбор скорости обмена данными:<br>1 – 100Mbps;<br>0 – 10Mbps.<br>Игнорируется, если установлен бит AutoNegotiation (0.12 = 1) | RW | PHY_CTRL |
| 12 | Auto-Negotiation Enable | Разрешение режима автоподстройки:<br>1 – режим автоподстройки включен;<br>0 – режим автоподстройки отключен | RW | PHY_CTRL |
| 11 | Power Down | Разрешение режима пониженного энергопотребления:<br>1 – режим пониженного потребления;<br>0 – штатный режим | RW | 0 |
| 10 | Isolate | Разрешение отключения интерфейса MII от PHY:<br>1 – PHY отключен от интерфейса MII<br>0 – нормальное функционирование | RW | PHY_CTRL |
| 9 | Restart Auto-Negotiate | Перезапуск автоподстройки:<br>1 – перезапуск автоподстройки;<br>0 – штатный режим.<br>Самоочищаемый | RW/SC | 0 |

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 8 | Duplex Mode | Выбор режима работы блока PHY.<br>1 – полнодуплексный режим;<br>0 – полудуплексный режим.<br>Игнорируется, если установлен бит AutoNegotiation (0.12 = 1) | RW | PHY_CTRL |
| 7 | Collision Test | Выбор режима тестирования Collision Test:<br>1 – включен COL test;<br>0 – COL test отключен | RW | 0 |
| 6…0 | Reserved | | RO | 0 |

### 28.12.5 Основной регистр состояния (1)

Таблица 484 – Основной регистр состояния (1)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15 | 100Base-T4 | 1 – доступен режим 100Base-T4;<br>0 – режим 100Base-T4 не доступен | RO | 0 |
| 14 | 100Base-TX Full Duplex | 1 – возможен режим полного дуплекса 100 Мбит/с;<br>0 – режим полного дуплекса 100 Мбит/с невозможен | RO | 1 |
| 13 | 100Base-TX Half Duplex | 1 – возможен режим полудуплекса 100 Мбит/с;<br>0 – режим полудуплекса 100 Мбит/с невозможен | RO | 1 |
| 12 | 10Base-T Full Duplex | 1 – возможен режим полного дуплекса 10 Мбит/с;<br>0 – режим полного дуплекса 10 Мбит/с невозможен | RO | 1 |
| 11 | 10Base-T Half Duplex | 1 – возможен режим полудуплекса 10 Мбит/с;<br>0 – режим полудуплекса 10 Мбит/с невозможен | RO | 1 |
| 10…6 | Reserved | | RO | 0 |
| 5 | Auto-Negotiate Complete | 1 – автоподстройка завершена;<br>0 – автоподстройка не завершена | RO | 0 |
| 4 | Remote Fault | 1 – обнаружено состояние remote fault;<br>0 – remote fault отсутствует | RO/LH | 0 |
| 3 | Auto-Negotiate Ability | 1 – возможна автоподстройка;<br>0 – автоподстройка не возможна | RO | 1 |
| 2 | Link Status | 1 – линия подключена;<br>0 – линия отключена | RO/LL | 0 |
| 1 | Jabber Detect | 1 – обнаружено состояние jabber;<br>0 – состояние jabber отсутствует | RO/LH | 0 |
| 0 | Extended Capabilities | 1 – поддерживаются расширенные регистры;<br>0 – расширенные регистры не поддерживаются | RO | 1 |

### 28.12.6 Регистры идентификатора PHY (2, 3)

Таблица 485 – Регистры идентификатора PHY (2, 3)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15…0 | PHY ID Number | 32-битный идентификатор модели PHY | RW | |

### 28.12.7 Регистр рекомендаций автоподстройки (4)

Таблица 486 – Регистр рекомендаций автоподстройки (4)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15 | Next Page | 1 – next page поддерживается;<br>0 – next page не поддерживается | RO | 0 |
| 14 | Reserved | | RO | 0 |
| 13 | Remote Fault | 1 – обнаружена remote fault,<br>0 – remote fault не обнаружена | RW | 0 |
| 12 | Reserved | | R/W | 0 |
| 11…10 | Pause Operation | 00 – PAUSE не обрабатывается;<br>01 – Асимметричная обработка PAUSE;<br>10 – Симметричная обработка PAUSE;<br>11 – Возможны и симметричная и асимметричная обработки PAUSE | R/W | 00 |
| 9 | 100Base-T4 | 1 – доступен режим 100Base-T4;<br>0 – режим 100Base-T4 не доступен | RO | 0 |
| 8 | 100Base-TX Full Duplex | 1 – доступен режим 100Base-T с полным дуплексом;<br>0 – режим 100Base-T с полным дуплексом не доступен | RW | PHY_CTRL |
| 7 | 100Base-TX | 1 – доступен режим 100Base-T;<br>0 – режим 100Base-T не доступен | RW | 1 |
| 6 | 10Base-T Full Duplex | 1 – доступен режим 10Base-T с полным дуплексом;<br>0 – режим 10Base-T с полным дуплексом не доступен | RW | PHY_CTRL |
| 5 | 10Base-T | 1 – доступен режим 10Base-T;<br>0 – режим 10Base-T не доступен | RW | PHY_CTRL |
| 4…0 | Selector Field | [00001] – IEEE 802.3 | RW | 00001 |

### 28.12.8 Регистр возможностей оппонента по автоподстройке (5)

Таблица 487 – Регистр возможностей оппонента по автоподстройке (5)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15 | Next Page | 1 – next page поддерживается;<br>0 – next page не поддерживается | RO | 0 |
| 14 | Acknowledge | 1 – получено кодовое слово link;<br>0 – кодовое слово link еще не получено | RO | 0 |
| 13 | Remote Fault | 1 – обнаружена remote fault;<br>0 – remote fault не обнаружена | RO | 0 |
| 12, 11 | Reserved | | RO | 0 |
| 10 | Pause Operation | 1 – обработка PAUSE поддерживаетс;<br>0 – обработка PAUSE не поддерживается | RO | 0 |
| 9 | 100Base-T4 | 1 – доступен режим 100Base-T4;<br>0 – режим 100Base-T4 не доступен | RO | 0 |
| 8 | 100Base-TX Full Duplex | 1 – доступен режим 100Base-T с полным дуплексом;<br>0 – режим 100Base-T с полным дуплексом не доступен | RO | 0 |
| 7 | 100Base-TX | 1 – доступен режим 100Base-T;<br>0 – режим 100Base-T не доступен | RO | 0 |
| 6 | 10Base-T Full Duplex | 1 – доступен режим 10Base-T с полным дуплексом;<br>0 – режим 10Base-T с полным дуплексом не доступен | RO | 0 |
| 5 | 10Base-T | 1 – доступен режим 10Base-T;<br>0 – режим 10Base-T не доступен | RO | 0 |
| 4…0 | Selector Field | [00001] – IEEE 802.3 | RO | 00001 |

### 28.12.9 Регистр расширенного управления автоподстройкой (6)

Таблица 488 – Регистр расширенного управления автоподстройкой (6)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15…5 | Reserved | | RO | 0 |
| 4 | Parallel Detection Fault | 1 – обнаружена ошибка parallel detection logic<br>0 – ошибки отсутствуют | RO/LH | 0 |
| 3 | Link Partner Next Page Able | 1 – оппонент поддерживает next page<br>0 – оппонент не поддерживает next page | RO | 0 |
| 2 | Next Page Able | next page не поддерживается | RO | 0 |
| 1 | Page Received | 1 – получена новая страница<br>0 – новая страница еще не получена | RO/LH | 0 |
| 0 | Link Partner Auto-Negotiation Able | 1 – оппонент имеет возможность автоматического согласования<br>0 – оппонент не имеет возможность автоматического согласования | RO | 0 |

### 28.12.10 Регистр расширенного управления режимами (18)

Таблица 489 – Регистр расширенного управления режимами (18)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15…14 | MIIMODE | Режим работы MII:<br>должен быть установлен в «00» (MII) | RW, NASR | 0 |
| 13 | CLKSELFREQ | Опорная частота<br>должен быть установлен в «0» (25 MHz) | RO, NASR | 0 |
| 12 | DSPBP | Режим обзора DSP.<br>Только для лабораторных тестов | RW, NASR | 0 |
| 11 | SQBP | Режим обхода SQUELCH | RW, NASR | 0 |
| 10 | FXMODE | Разрешение режима 100Base-FX.<br>Если включен, то режим (MODE) должен быть выставлен только «011» (100Base-TX FD) или «010» (100Base-TX FD) | RW, NASR | PHY_CTRL |
| 9 | PLLBP | Режим обхода PLL | RW, NASR | 0 |
| 8 | ADCBP | Режим обхода АЦП | RW, NASR | 0 |
| 7…5 | MODE | Текущий режим работы PHY | RW, NASR | PHY_CTRL |
| 4…0 | PHYADD | PHY Address.<br>Используется для доступа посредством MII, а также в качестве ключа для скремблирования | RW, NASR | 0 |

### 28.12.11 Регистр флагов прерываний (29)

Таблица 490 – Регистр флагов прерываний (29)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15…8 | Reserved | Ignore on read | RO/LH | 0 |
| 7 | INT7 | 1 – ENERGYON<br>0 – нет прерывания | RO/LH | 0 |
| 6 | INT6 | 1 – автоподстройка завершена<br>0 – нет прерывания | RO/LH | 0 |
| 5 | INT5 | 1 – обнаружена Remote Fault Detected<br>0 – нет прерывания | RO/LH | 0 |
| 4 | INT4 | 1 – отсутствует подключение к линии<br>0 – нет прерывания | RO/LH | 0 |
| 3 | INT3 | 1 – подтверждение автоподстройки от оппонента<br>0 – нет прерывания | RO/LH | 0 |
| 2 | INT2 | 1 – Parallel Detection Fault<br>0 – нет прерывания | RO/LH | 0 |
| 1 | INT1 | 1 – получена страница автоподстройки<br>0 – нет прерывания | RO/LH | 0 |
| 0 | Reserved | | RO/LH | 0 |

### 28.12.12 Регистр маски прерываний (30)

Таблица 491 – Регистр маски прерываний (30)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15…8 | Reserved | | RO | 0 |
| 7…0 | Mask Bits | 1 – прерывание разрешено<br>0 – прерывание запрещено | RW | 0 |

### 28.12.13 Регистр расширенного управления и состояния PHY (31)

Таблица 492 – Регистр расширенного управления и состояни PHY (31)

| Бит | Наименование | Описание | Режим | Значение по умолчанию |
|---|---|---|---|---|
| 15, 14 | Reserved | | RW | 0 |
| 13 | | | RO | 0 |
| 12 | Autodone | Индикатор завершения автоподстройки:<br>0 – автоподстройка не завершена или отключена;<br>1 – автоподстройка завершена | RO | 0 |
| 11…7 | Reserved | | RW | 0 |
| 6 | enable 4B5B | 0 – пропустить кодирование/ декодирование;<br>1 – включить кодирование/декодирование 4B5B | RW | 1 |
| 5 | Reserved | | RW | 0 |
| 4…2 | Speed Indication | Значение HCDSPEED:<br>[001] – 10Mbps HD;<br>[101] – 10Mbps FD;<br>[010] – 100Base-TX HD;<br>[110] – 100Base-TX FD | RO | 0 |
| 1 | Reserved | | RW | 0 |
| 0 | Scramble Disable | 0 – скремблирование включено;<br>1 – скремблирование отключено | RW | 0 |

# 29 Прерывания и исключения

Процессор и вложенный векторный контроллер прерываний (NVIC) назначают приоритет и обрабатывают все исключения. Все исключения обрабатываются в режиме Handler. Состояние процессора автоматически сохраняется в стек при возникновении исключения и автоматически восстанавливается из стека по завершении обработки исключения.

Следующие характеристики позволяют увеличить эффективность обработки исключений и уменьшить задержки:

– автоматическое сохранение состояния. Процессор помещает содержимое регистров в стек при входе в исключение и извлекает их при выходе из исключения, при этом не затрачивается дополнительных инструкций (см. подраздел 29.5 «Вход в прерывание»);

– автоматическое считывание таблицы векторов при входе, которая содержит адрес обработчика. Бит 0 значения вектора загружается в T-бит регистра EPSR при входе в исключение. Создание таблицы входа с очищенным битом 0 генерирует аппаратную ошибку Hard Fault на первой инструкции обработчика соответствующему этому вектору;

– тесно связанный интерфейс между процессором и NVIC позволяет эффективно обрабатывать прерывания и поздно поступающие (late-arriving) запросы прерывания с высоким приоритетом;

– два бита конфигурирования приоритета прерываний обеспечивают четыре уровня приоритета;

– разделенный стек для режимов Handler и Thread;

– команды передачи управления исключению соответствуют соглашению C/C++ стандарту ARM Architecture Procedure Call Standard (AAPCS);

– маскирование приоритета для поддержки критических регионов.

## 29.1 Типы исключений

В процессоре существуют различные типы исключений. Ошибка – это исключение, возникшее вследствие ошибочных условий. Ошибки могут генерироваться синхронно или асинхронно с соответствующей инструкцией, которая ее вызвала. Обычно ошибки генерируются синхронно. Ошибки, вызванные записью на внешней шине AHB асинхронные. Синхронные ошибки всегда генерируются совместно с инструкцией, которая ее вызвала. Генерация асинхронных ошибок совместно с вызвавшей ее инструкцией не гарантирована.

В таблице 493 представлены типы исключений, их номера и приоритет. Номер показывает словное смещение векторов исключений относительно стартового адреса таблицы векторов, которая всегда располагается с адреса 0x0. Исключения с наименьшими числами, представленные в столбце приоритета таблицы, имеют наивысший приоритет. Как формируются исключения, асинхронно или синхронно, также показано.

Таблица 493 – Таблица различных типов исключений

| Номер | Тип | Приоритет | Описание | Активация |
|---|---|---|---|---|
| - | - | - | Вершина стека загружается в начало таблицы после сброса | |
| 1 | RESET | -3 (наивысший) | Вызывается при включении питания или горячем сбросе. На первой инструкции в режиме Thread падает до низшего приоритета. | Асинхронный |
| 2 | NMI | -2 | Это исключение не может быть:<br>– маскировано или задержано активизацией любого другого исключения;<br>– заменено любым другим исключением, кроме сброса.<br>Может быть вызвано только программным способом | Асинхронный |
| 3 | Hard Fault | -1 | Все виды ошибок | Синхронный/ Асинхронный |
| 4-10 | - | - | Зарезервировано | - |
| 11 | SVCall | конфигурируемый | Системное обслуживание, вызванное инструкцией SVC | Синхронный |
| 12-13 | - | - | Зарезервировано | - |
| 14 | PendSV | конфигурируемый | Запрос ожидания обработки для обслуживания системы. Ожидание обработки генерируется программным обеспечением. | Асинхронный |
| 15 | SysTick | конфигурируемый | Системный таймер закончил работу | Асинхронный |
| 16-47 | IRQ | конфигурируемый | Запрос устанавливается извне процессора или обусловлен программным обеспечением. | Асинхронный |

## 29.2 Приоритет исключений

В таблице 494 показано влияние приоритетов на то, когда и как процессор обрабатывает исключения.

Таблица 494 – Сценарий исключений

| Сценарий | Описание |
|---|---|
| Приоритетное прерывание обслуживания | Ожидающее обработки исключение может прервать текущее выполнение задачи, если приоритет этого исключения выше, чем приоритет текущего прерывания. Когда одно исключение прерывает обслуживание другого, то появляется вложенность исключений.<br>При входе в исключение процессор автоматически сохраняет свое состояние, помещая его в стек. Выбирается вектор, соответствующий исключению. Выполнение начинается с адреса указанного в таблице векторов. Выполнение первой инструкции начинается, когда состояние процессора сохранено. Сохранение состояния происходит через интерфейсы ITCM, DTCM или AHB-Lite в зависимости от:<br>– значения указателя стека, когда процессор обнаружил исключение;<br>– размера памяти TCM.<br>Выбор вектора происходит по внешнему интерфейсу AHB-Lite или интерфейсу памяти ITCM в зависимости от конфигурации ITCM |
| Возврат | Когда выполняется инструкция возврата, процессор выгружает стек и возвращается к помещенному в стек исключению или в Thread режим.<br>По завершении выполнения обработчика исключения процессор автоматически восстанавливает состояние, выгружая стек и переходит в состояние, предшествующее исключению |
| Запаздывание (Late-arriving) | Это механизм, используемый процессором для ускорения прерывания обслуживания. Если исключение с более высоким приоритетом прибывает во время сохранения состояния предыдущего прерывания обслуживания, то процессор переключается на обработку исключения с более высоким приоритетом вместо выполнения выборки вектора для этого исключения. На сохранение состояния процессора запоздавшее исключение не влияет, потому что сохраняемое состояние аналогично для обоих исключений и поэтому продолжается не прерываясь. Запаздывающие исключения распознаются в момент, когда происходит выборка вектора. Если исключение с более высоким приоритетом распознается слишком поздно, чтобы быть обслужено как запаздывающее, то оно ожидает обработки и позднее прерывает обслуживание исходного обработчика исключения |

## 29.3 Уровни приоритета

NVIC поддерживает программное присвоение уровней приоритета. Можно установить требуемый уровень приоритета прерывания записью значения от 0 до 3 в поле IP_N регистра приоритета прерывания. Приоритет с уровнем 0 считается наивысшим, а с уровнем 3 самым низким. Например, если присвоить уровень приоритета 1 для IRQ[0] и уровень приоритета 0 для IRQ[31], то IRQ[31] имеет приоритет выше IRQ[0].

Программное присвоение приоритетов не влияет на немаскируемое прерывание NMI и исключение Hard Fault. Они всегда имеют приоритет выше, чем внешние прерывания.

Когда несколько исключений имеют одинаковый приоритет, тогда сначала будет обработано исключение с меньшим порядковым номером, а самым последним – с наибольшим номером. Например, если оба IRQ[0] и IRQ[1] имеют приоритет 1, тогда IRQ[0] предшествует IRQ[1].

Новое исключение прерывает обработку текущего исключения, если его приоритет выше. Если новое исключение с таким же приоритетом, как обрабатываемое в текущий момент, то обработка не прерывается независимо от номера прерывания.

## 29.4 Стек

Процессор поддерживает два индивидуальных стека.

Процессорный стек.

Необходимо установить режим Thread, чтобы использовать SP_process или SP_main.

Основной стек.

Режим Handler использует только основной стек.

Когда происходит прерывание обслуживания, контекст автоматически сохраняется в стек, который был активным в момент обнаружения исключения. Если исключение прерывает обслуживание в Thread режиме, то контекст прерванной задачи может быть помещен в стек с применением SP_process или SP_main в зависимости от значения бита Active Stack Pointer регистра CONTROL ядра контроллера.

Если исключение прерывает обслуживание другого исключения, выполняемого в режиме Handler, то прерванный контекст можем быть помещен только в стек SP_main, так как только этот указатель стека может быть активным в режиме Handler.

При возвращении из прерывания значение EXC_RETURN определяет, какой стек использовать для извлечения контекста. Значение EXC_RETURN помещается в регистр R14 в процессе входа в исключение, и соответствующий стек используется для сохранения контекста. Если ваш код обработчика исключения изменяет стек, то вы должны быть уверены, что значение EXC_RETURN для возвращения из исключения корректно.

Все обработчики исключений должны использовать SP_main для их локальных переменных. Инструкциям MSR и MRS доступны оба указателя стека.

## 29.5 Вход в прерывание

Когда процессор обрабатывает исключение, то автоматически помещаются в стек следующие восемь регистров:

- xPSR;
- Адрес возврата;
- Регистр связи (LR);

– R12;
– R3;
– R2;
– R1;
– R0.

SP декрементируется на восемь слов по завершении загрузки стека. На рисунке 135 показано содержимое стека после того, как исключение прервало обслуживание текущего программного потока.




Рисунок 135 – Содержимое стека после того, как исключение прервало обслуживание текущего программного потока

После возвращения из исключения процессор автоматически выгружает восемь регистров из стека. Значение возврата из исключения EXC_RETURN автоматически загружается в LR при входе в исключение, позволяя описывать обработчики прерываний как обычные C/C++ функции.

Таблица 495 описывает шаги процессора, прежде чем происходит вход в исключение.

Таблица 495 – Таблица шагов процессора

| Действие | Описание |
|---|---|
| Сохранение 8 регистров | Сохранение xPSR, Адрес возврата, LR, R12, R3, R2, R1 и R0 в стек. |
| Чтение таблицы векторов | Чтение вектора входа из соответствующего адреса таблицы векторов: (0x0)+(номер исключения*4). Чтение таблицы векторов происходит после того как все восемь регистров помещены в стек. |
| Чтение SP_main из таблицы векторов | SP_main обновляется в таблице векторов только после сброса. Другие исключения не модифицируют SP_main таким способом. |
| Обновление LR | LR устанавливается в соответствии с EXC_RETURN для корректного возврата из исключения. |
| Обновление PC | Обновление PC прочитанными данными из таблицы векторов. Никакие другие запаздывающие прерывания не могут выполняться, пока первая инструкция исключения не начнет исполняться. |
| Загрузка конвейера | Конвейер заполняется последовательностью инструкций из адреса вектора. |

## 29.6 Выход из исключений

Инструкция возврата из исключения загружает PC значением EXC_RETURN, которое было загружено в LR при входе в обработчик прерывания. Это сигнализирует процессору о том, то исключение завершено и процессор инициирует последовательность выхода из исключения.

При выходе их исключения процессор либо возвращается в последнее, помещенное в стек исключение, либо переходит в режим Thread.

Таблица 496 описывает шаги процессора, прежде чем происходит вход в исключение.

Таблица 496 – Таблица шагов процессора

| Действие | Описание |
|---|---|
| Выбор SP | Установка CONTROL[1] в соответствии с EXC_RETURN. |
| Выгрузка 8 регистров из стека | Выгрузка R0, R1, R2, R3, R12, LR, PC и xPSR из стека, выбранного EXC_RETURN. Значение xPSR[5:0] выгружаемое из стека определяет номер исключения, что в свою очередь определяет приоритет задачи, к которой необходимо вернуться.<br>Значение EXC_RETURN определяет, в какой режим вернуться |

Возврат из исключения происходит при выполнении одной из следующих инструкций, выполняемых в режиме Handler и загружающих в PC значение 0xFXXXXXXX:

- POP, которая включает загрузку PC;
- BX с любым регистром.

Если используется такой способ, значение, записываемое в PC, заменяет значение EXC_RETURN.

Таблица описывает поведение при выходе из исключения при различных EXC_RETURN[3:0].

Таблица 497 – Таблица, описывающая поведение при выходе из исключения

| EXC_RETURN[3:0] | Описание |
|---|---|
| 4’bXXX0 | Зарезервировано |
| 4’b0001 | Возврат в режим Handler.<br>Возврат из исключения с получением состояния из основного стека.<br>Далее после выхода из обработчика исключения при исполнении кода использует SP_Main |
| 4’b0011 | Зарезервировано |
| 4’b01X1 | Зарезервировано |
| 4’b1001 | Возврат в режим Thread.<br>Возврат из исключения с получением состояния из стека SP_Main.<br>Далее после выхода из обработчика исключения при исполнении кода использует SP_Main |

| EXC_RETURN[3:0] | Описание |
|---|---|
| 4’b1101 | Возврат в режим Thread.<br>Возврат из исключения с получением состояния из стека Process.<br>Далее после выхода из обработчика исключения при исполнении кода использует SP_Process |
| 4’b1X11 | Зарезервировано |

Если значение EXC_RETURN загружается в PC в режиме Thread или из таблицы векторов, или любой другой инструкцией, значение рассматривается как адрес, а не как специальное значение. Если этот адрес из диапазона адресов имеющий атрибут XN (выполнение запрещено), то возникает аппаратная ошибка Hard Fault.

Обработчик прерывания должен сохранить значение EXC_RETURN[28:4] или записать их как все единицы.

## 29.7 Запаздывание (late-arriving)

Запаздывающее исключение может иметь преимущество в обслуживании по отношению к предыдущему исключению, если выборка вектора не началась и запаздывающее исключение имеет:

– приоритет выше предыдущего исключения;

– одинаковый приоритет, но меньший порядковый номер, чем предыдущее исключение.

Запаздывающее исключение вызывает изменение выборки вектора адреса и предвыборки исключения. Сохранение состояния не выполняется для запаздывающего исключения, потому что это уже было выполнено для исходного исключения. В этом случае, выполнение начинается с вектора запаздывающего исключения, в то время как предыдущее исключение ожидает обслуживания.

Если исключение с высоким приоритетом распознается после выборки вектора исходного исключения, то запаздывающее исключение не может использовать контекст, который помещен в стек для исходного исключения. В этом случае прерывается обслуживание исходного прерывания, и контекст сохраняется в стеке.

## 29.8 Передача управления исключению

Таблица 498 показывает, в соответствии с какими правилами процессор передает управление исключению.

Таблица 498 – Таблица, описывающая, в соответствии с какими правилами процессор передает управление исключению

| Активность процессора и обнаружение исключения | Передача управления исключению |
|---|---|
| Инструкция | Завершается инструкция, и исключение начинается перед выполнением следующей инструкции |
| Вход в исключение | Это классифицируется как запаздывающее исключение. Если новое исключение имеет приоритет выше или такой же, и номер исключения меньше, чем у первого исключения, то ядру необходимо обслужить запаздывающее исключение первым. Если нет, то запаздывающее прерывания ожидает обслуживания и используются обычные правила прерывания обслуживания.<br>Если запаздывающее исключение поступило достаточно рано во время фазы помещения в стек, то оно рассматривается как запаздывающее. В этом случае ядро выбирает вектор для запаздывающего исключения вместо вектора первого исключения.<br>Если запаздывающее прерывание поступает слишком поздно (позже времени фазы помещения контекста в стек), то оно не может обрабатываться как запаздывающее. Вместо этого выбирается вектор первого исключения, начинается выполнение по вектору адреса первого исключения, а запаздывающее прерывание ожидает обслуживания в соответствии с обычными правилами прерывания обслуживания |
| Завершение исключения | Завершается последовательность возврата из исключения, и восстанавливается выполнение задачи возврата. Обычные правила прерывания обслуживания применяются в этом случае |

## 29.9 Уровни активации

Если нет активных исключений, то процессор находится в режиме Thread. Если исключения или Hard Fault активны, то процессор входит в режим Handler.

Таблица 499 – Уровни активации стека

| Активное исключение | Уровень активации | Стек |
|---|---|---|
| Нет | Режим Thread | Основной или стек процессов |
| Исключение активно | Асинхронное прерывание обслуживания | Основной |
| Fault handler активен | Асинхронное или синхронное прерывание обслуживания | Основной |

Таблица 500 – Транзакции исключений

| Активное исключение | Старт события | Тип транзакции | Стек |
|---|---|---|---|
| Сброс | Сигнал сброса | Thread | Основной |
| ISR или NMI[a] | Установка запроса обслуживания программной инструкцией или аппаратным сигналом (актуально для ISR: прерывание NMI не может быть вызвано аппаратно, только программным способом) | Асинхронное прерывание обработки | Основной |
| Hard Fault | Любая ошибка | Синхронное или асинхронное прерывание обработки | Основной |
| SVC[b] | Инструкция SVC | Синхронное прерывание обработки | Основной |
| [a] Немаскируемое прерывание;<br>[b] Вызов супервизора | | | |

Таблица 501 – Подгруппы транзакций исключений

| Группа активации | Старт события | Активация | Приоритет |
|---|---|---|---|
| Thread | Сигнал сброса | Асинхронная | Незамедлительный, thread имеет низкий приоритет |
| Прерывание или NMI | Аппаратный сигнал или установка запроса обслуживания | Асинхронная | Прерывание обслуживания согласно приоритету |
| SVC | Инструкция SVC | Синхронная | Если приоритет для SVCall исключения запрограммирован выше чем для текущего исключения, то выполняется SVCall. Если нет, то SVC вызывает Hard Fault. |
| PendSV | Программный запрос обслуживания | Асинхронная | Прерывание обслуживания согласно приоритету |
| SysTick | Счетчик достиг нуля или установлен запрос обслуживания | Асинхронная | Прерывание обслуживания согласно приоритету |
| Hard Fault | любая ошибка | Синхронная или асинхронная[a] | Выше чем другие за исключением NMI[b] |

| Группа активации | Старт события | Активация | Приоритет |
|---|---|---|---|

[a] Активация зависит от причины вызвавшей ошибку;

[b] Если Hard Fault происходит когда процессор выполняет обработчик NMI или Hard Fault, процессор входит в состояние lock-up

## 29.10 Lock-up

Процессор имеет состояние lock-up, в которое входит, когда случается неисправимая ситуация. Причины неисправимой ситуации могут быть синхронные или асинхронны ошибки, включая конфликтную инструкцию SVC.

Процессор может войти в состояние lock-up с приоритетом -1 или -2. NMI может быть причиной выхода процессора из состояния lock-up, если это будет приоритет -1. Отладчик также может быть причиной выхода процессора из состояния lock-up.

Периферийные блоки формируют прерывания с IRQ0 до IRQ31

Таблица 502 – Принцип формирования прерываний

| Прерывания | Блок | Принцип формирования |
|---|---|---|
| IRQ0 | MIL-STD-1553B2 | Сигнал прерывания от контроллера интерфейса по ГОСТ Р 52070-2003. Канал 2. Сигнал VALMESS, ERR, RFLAGN, IDLE. |
| IRQ1 | MIL-STD-1553B1 | Аналогично. Канал 1 |
| IRQ2 | USB | Прерывания от USB Host при наличии соответствующих флагов разрешения<br>HostSOFSent или HostConnEvent или HostResume или HostTransDone.<br>Прерывания от USB Slave при наличии соответствующих флагов разрешения<br>SlaveNAKSent или SlaveSOFRXed или SlaveResetEvent или SlaveResume или SlaveTransDone. |
| IRQ3 | CAN1 | Сигнал прерывания от блока CAN.<br>Возникает при установленном бите GLB_INT_EN и при сигналах<br>RX_INT_EN[31:0] и RX_INT[31:0] или<br>TX_INT_EN[31:0] и EX_INT[31:0] или<br>ERR_INT_EN и (ACKERR или FRAMEERR или CRCERR или BSERR или BITERR) или<br>ERR_OVER_INT_EN и<br>REC > CAN_ERR_MAX или<br>TEC > CAN_ERR_MAX |
| IRQ4 | CAN2 | Аналогично |

| Прерывания | Блок | Принцип формирования |
|---|---|---|
| IRQ5 | DMA | Прерывания от DMA<br>DMA_ERR или DMA_DONE.<br>Обработка прерываний от DMA в соответствии с разделом «Индикация ошибок» технического описания DMA |
| IRQ6 | UART1 | Сигнал UARTINTR |
| IRQ7 | UART2 | Сигнал UARTINTR |
| IRQ8 | SSP1 | Сигнал SSPINTR |
| IRQ9 | BUSY | Сигнал занятости от NAND флэш |
| IRQ10 | ARINC429R1-ARINC429R8 | Сигнал прерывания от одного из приемников ARINC-429.<br>Сигналы DR, ERROR, FF, HF. |
| IRQ11 | POWER | Сигнал прерывания от POWER Detecor |
| IRQ12 | WWDG | Сигнал прерывания от WWDG |
| IRQ13 | Timer4 | Сигнал прерывания от Таймера<br>TIM_STATUS и TIM_IE. |
| IRQ14 | Timer1 | Аналогично |
| IRQ15 | Timer2 | Аналогично |
| IRQ16 | Timer3 | Аналогично |
| IRQ17 | ADC | Сигналы прерываний от АЦП<br>EOCIF_1 или AWOIF_1 или EOCIF_2 или AWOIF_2. |
| IRQ18 | Ethernet | Сигнал прерывания от контроллера интерфейса Ehternet. |
| IRQ19 | SSP3 | Сигнал SSPINTR |
| IRQ20 | SSP2 | Сигнал SSPINTR |
| IRQ21 | ARINC429T1 | Сигнал прерывания от передатчика по ГОСТ 18977-79.<br>Сигналы FFT, HFT, TX_R. |
| IRQ22 | ARINC429T2 | Аналогично |
| IRQ23 | ARINC429T3 | Аналогично |
| IRQ24 | ARINC429T4 | Аналогично |
| IRQ25…IRQ26 | Зарезервировано | |
| IRQ27 | BACKUP | Прерывание от BKP и часов реального времени. |
| IRQ28 | Внешнее прерывание 1 | Сигнал EXTINT1<br>Вывод PC[5] в основном режиме |
| IRQ29 | Внешнее прерывание 2 | Сигнал EXTINT2<br>Вывод PC[6] в основном режиме |
| IRQ30 | Внешнее прерывание 3 | Сигнал EXTINT3<br>Вывод PC[7] в основном режиме |
| IRQ31 | Внешнее прерывание 4 | Сигнал EXTINT4<br>Вывод PC[8] в основном режиме |

# 30 Контроллер прерываний NVIC

NVIC (Nested Vectored Interrupt Controller) поддерживает прерывания, у которых может быть переопределен приоритет. NVIC и ядро процессора тесно связаны, что позволяет уменьшить задержки обработки прерываний и повысить эффективность обработки запаздывающих прерываний.

Все регистры NVIC доступны только при использовании словных транзакций. Любая попытка записать полуслово или индивидуальный байт вызывает порчу бит регистра. Регистры NVIC используют режим доступа little-endian. Доступ процессора к ним корректно обрабатывается, несмотря на конфигурацию endian процессора. Доступ DAP также должен быть интерпретирован как little-endian.

## 30.1 Логика работы прерываний контроллера NVIC

В данном разделе описывается функционирование контроллера NVIC при поступлении на его вход запросов прерываний IRQ от различных модулей периферии микросхемы.

Первоначальным условием работы прерывания является его разрешение в модуле NVIC. За это отвечают регистры:

ISER – за разрешение прерываний,

ICER – за запрет прерываний.

В случае, когда соответствующий запрос разрешен (при данном условии рассмотрены все диаграммы в разделе), и приходит сигнал активации прерывания – запрос IRQ request, то возникает признак отложенного прерывания IRQ pending. Данный признак переводит прерывание в состояние ожидания его обработки ядром.

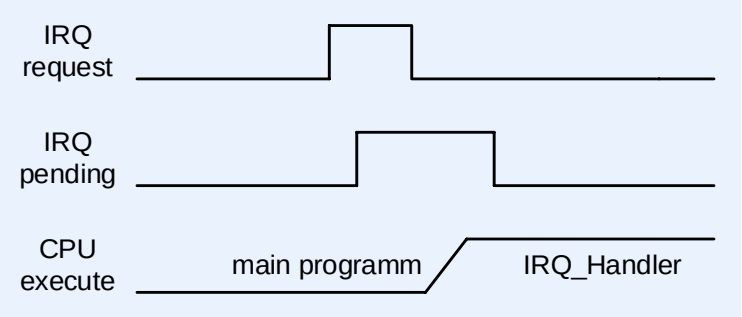



Рисунок 136 – Выставление отложенного запроса на прерывание
и последующая его обработка

Pending биты выставляются в регистрах ISPR/ICPR, которые в свою очередь позволяют программно управлять признаком отложенного прерывания. ISPR – для установки pending бит, ICPR – для сброса соответственно. Если после прихода запроса на прерывание IRQ request, сбросить pending бит в регистре ICPR до того, как ядро приступит к его обработке, то прерывание будет проигнорировано – рисунок 137.

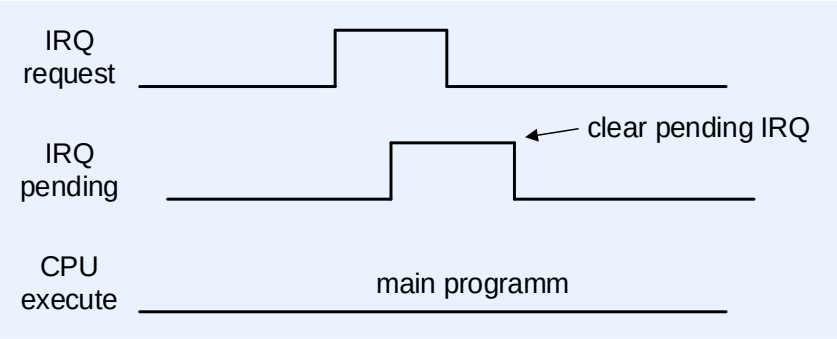



Рисунок 137 – Сброс признака отложенного прерывания, до обработки ядром

Если произойдет снятие запроса IRQ request от источника, «защелкивание» признака отложенного прерывания гарантирует отработку его ядром в соответствии с приоритетом. – рисунок 138. Сам IRQ pending признак снимается автоматически, когда прерывание становится активным, о чем сигнализирует признак IRQ active.

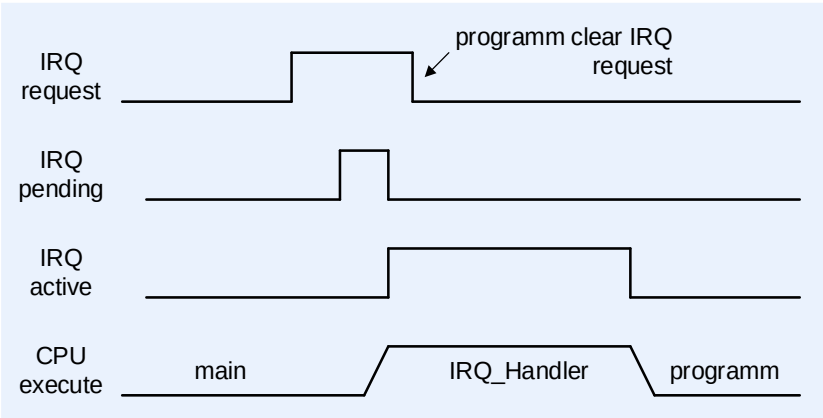



Рисунок 138 – Сброс признака отложенного прерывания, до обработки ядром

После того как прерывание стало активным, повторно запустить обработчик того же прерывания будет невозможно до тех пор, пока не будет завершена процедура обработки прерывания командой выхода из исключения. После выполнения команды выхода происходит сброс признака активности IRQ active.

При удержании источником на входе NVIC запроса на обработку IRQ request, по окончании обработки прерывания и снятия признака активного прерывания IRQ active, происходит повторное выставление признака отложенного прерывания IRQ pending – «защелкивание» pending бита, сброс которого в дальнейшем инициирует повторную активность и обработку того же исключения – рисунок 139.

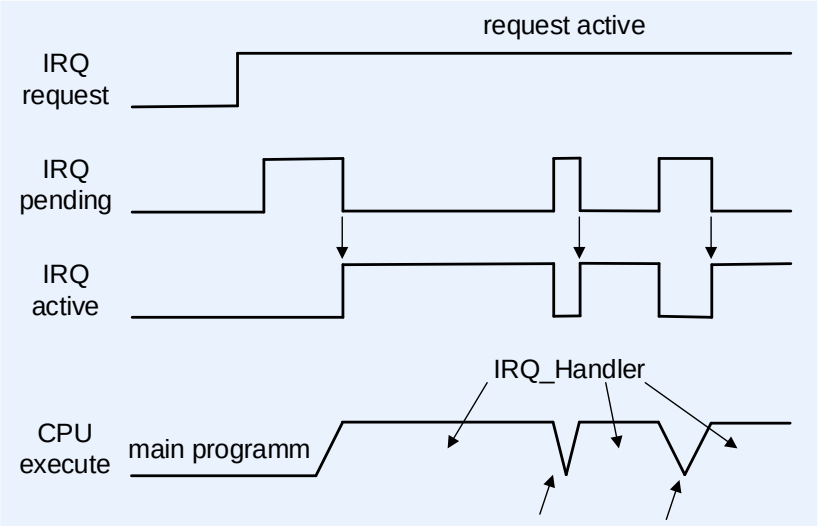



Рисунок 139 – Повторная обработка прерываний
при удержании запроса от источника

Необходимо учитывать, что если источник прерываний выдает многократную установку и снятие запроса IRQ request на входе контроллера NVIC, то в таком случае только первый запрос выставляет признак отложенного прерывания IRQ pending, а остальные запросы до начала процедуры обработки прерывания (в момент активного признака отложенного прерывания) будут проигнорированы ядром – рисунок 140.

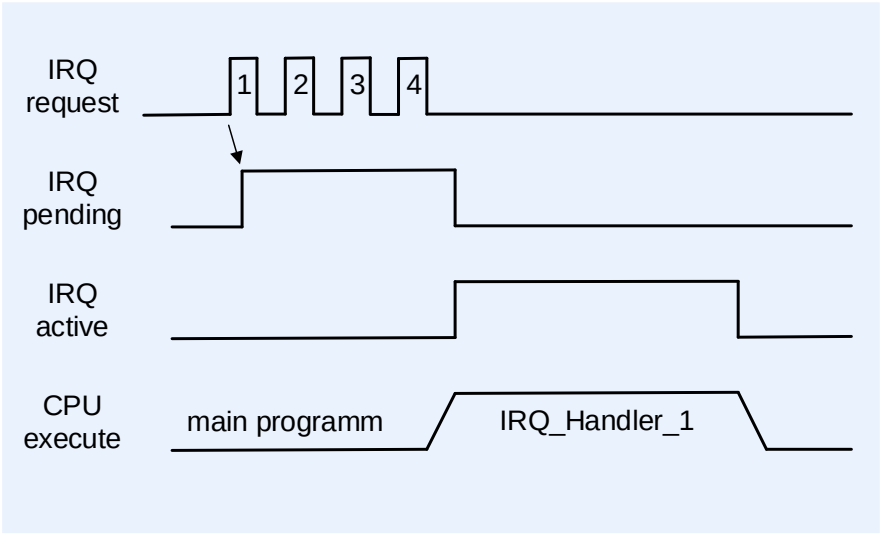



Рисунок 140 – Многократная установка снятие запроса IRQ request

Если запрос на прерывание пришел в момент активного прерывания, то в такой ситуации уже будут отработаны оба запроса на прерывание. В отличие от случая, изображенного на рисунке 140, запрос приходит тогда, когда признак отложенного прерывания IRQ pending уже сброшен, и новый запрос как раз его выставляет, что в дальнейшем позволяет провести повторную обработку прерывания – рисунок 141.

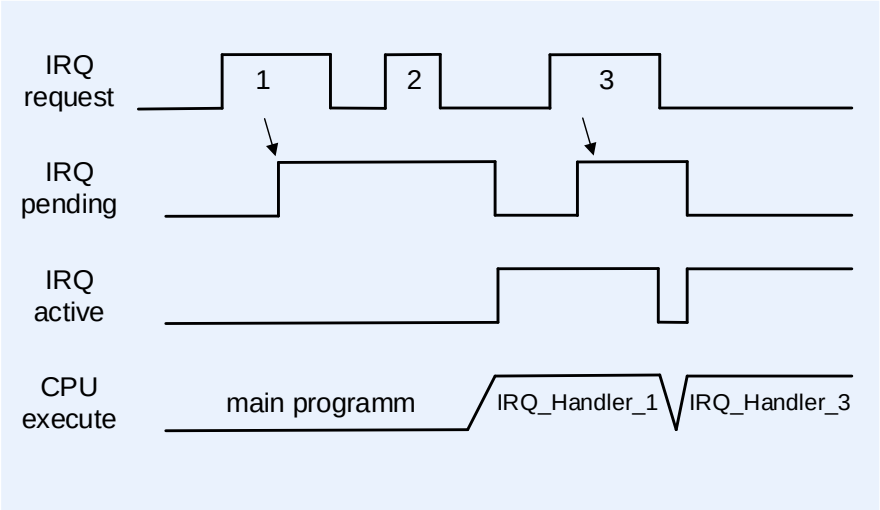



Рисунок 141 – Повторная установка запроса на прерывание
в момент выполнения обработчика исключения

Выставление признака отложенного прерывания возможно даже в тех случаях, когда соответствующее прерывание запрещено. Все отложенные прерывания будут отражены в ISPR/ICPR, и в случае разрешения таких прерываний регистром ISPR, ядро тут же приступит к их обработке. Рекомендуется перед разрешением соответствующего прерывания убедиться в отсутствии признака отложенного запроса и, при необходимости, сбросить его.

## 30.2 Программная модель NVIC

В этом разделе описываются регистры NVIC. Описание содержит:

- карту памяти NVIC;
- описание регистров NVIC.

Таблица 503 – Карта памяти регистров

| Имя регистра | Тип операции | Адрес | Значение после сброса | Описание |
|---|---|---|---|---|
| ISER | R/W | 0xE000E100 | 0x00000000 | Регистр разрешения прерываний |
| ICER | R/W | 0xE000E180 | 0x00000000 | Регистр запрета прерывания |
| ISPR | R/W | 0xE000E200 | 0x00000000 | Регистр перевода прерывания в состояние ожидания обслуживания |
| ICPR | R/W | 0xE000E280 | 0x00000000 | Регистр сброса состояния ожидания обслуживания |
| IPR0 | R/W | 0xE000E400 | 0x00000000 | Регистр приоритета прерываний 0 |
| IPR1 | R/W | 0xE000E404 | 0x00000000 | Регистр приоритета прерываний 1 |
| IPR2 | R/W | 0xE000E408 | 0x00000000 | Регистр приоритета прерываний 2 |
| IPR3 | R/W | 0xE000E40C | 0x00000000 | Регистр приоритета прерываний 3 |
| IPR4 | R/W | 0xE000E410 | 0x00000000 | Регистр приоритета прерываний 4 |
| IPR5 | R/W | 0xE000E414 | 0x00000000 | Регистр приоритета прерываний 5 |
| IPR6 | R/W | 0xE000E418 | 0x00000000 | Регистр приоритета прерываний 6 |
| IPR7 | R/W | 0xE000E41C | 0x00000000 | Регистр приоритета прерываний 7 |

## 30.3 Регистр разрешения прерываний

Этот регистр используется для разрешения прерываний и определения, какие прерывания разрешены. Каждый бит этого регистра соответствует одному из 32 прерываний. Установка бита в этом регистре разрешает соответствующее прерывание.

Когда бит разрешения обслуживания прерывания установлен, процессор активизирует прерывания на основе их приоритета. Когда бит разрешения очищен, установка сигнала запроса обслуживания прерывания не приводит к активации прерывания, несмотря на его приоритет. Следовательно, запрещенное прерывание может служить как защелка бита общего назначения. Вы можете прочитать этот бит или сбросить его, не вызывая прерывания.

Сброс разрешения производится записью соответствующего бита Регистра запрета прерывания. Это также очищает соответствующий бит в Регистре разрешения прерываний.

Адрес регистра: 0xE000E100

Доступ: Чтение/запись

Значение после сброса: 0x00000000

Таблица 504 – Регистр разрешения прерываний

| Биты | Поле | Функция |
|---|---|---|
| 31…0 | SETENA | Биты разрешения прерывания.<br>При записи:<br>1 – разрешение прерывания;<br>0 – не оказывает влияния.<br>При чтении:<br>1 – прерывание разрешено;<br>0 – прерывание запрещено.<br>Запись нуля в SETENA не оказывает влияние. Чтение бита возвращает текущее состояние разрешения прерывания. Сброс очищает поле SETENA. |

## 30.4 Регистр запрета прерывания

Этот регистр используется для запрета прерываний и определения, какие прерывания разрешены. Каждый бит этого регистра соответствует одному из 32 прерываний. Установка бита в этом регистре запрещает соответствующее прерывание.

Адрес регистра: 0xE000E180

Доступ: Чтение/запись

Значение после сброса: 0x00000000

Запись «1» в этот регистр не оказывает влияние на текущее активное прерывание, а только предотвращает новую активацию.

Таблица 505 – Регистр запрета прерываний

| Биты | Поле | Функция |
|---|---|---|
| 31…0 | CLRENA | Биты запрещения прерывания.<br>При записи:<br>1 – запрещает прерывание;<br>0 – не оказывает влияния.<br>При чтении:<br>1 – прерывание разрешено;<br>0 – прерывание запрещено.<br>Запись нуля в CLRENA не оказывает влияние. Чтение бита возвращает текущее состояние разрешения. Сброс очищает поле CLRENA. |

## 30.5 Регистр перевода прерывания в состояние ожидания обслуживания

Этот регистр используется для принудительного перевода прерываний в состояние ожидания обслуживания, а также определения какие прерывания находятся в этом состоянии.

Каждый бит этого регистра соответствует одному из 32 прерываний. Установка бит этого регистра переводит в состояние ожидания обслуживания соответствующего прерывания. Запись «0» в этот регистр не оказывает влияние на состояние соответствующего прерывания.

Сброс бита перевода в состояние ожидания обслуживания производится записью «1» в соответствующий бит Регистра сброса состояния ожидания обслуживания.

Запись в регистр перевода прерывания в состояние ожидания обслуживания не оказывает влияние на прерывание, которое обслуживается.

Адрес регистра: 0xE000E200

Доступ: Чтение/запись

Значение после сброса: 0x00000000

Таблица 506 – Регистр перевода прерывания в состояние ожидания обслуживания

| Биты | Поле | Функция |
|---|---|---|
| 31…0 | SETPEND | При записи:<br>1 – разрешение ожидания обслуживания;<br>0 – не оказывает влияния.<br>При чтении:<br>1 – прерывание в состоянии ожидания обслуживания;<br>0 – прерывание не в состоянии ожидания обслуживания. |

### 30.6 Регистр сброса состояния ожидания обслуживания

Этот регистр используется для сброса состояние ожидания обслуживания прерывания, а также определения какие прерывания находятся в состоянии ожидания обслуживания.

Каждый бит этого регистра соответствует одному из 32 прерываний. Установка бит этого регистра сбрасывает состояние ожидания обслуживания соответствующего прерывания.

Запись в регистр сброса состояния ожидания обслуживания не оказывает влияние на прерывание, которое обслуживается.

Адрес регистра: 0xE000E280

Доступ: Чтение/запись

Значение после сброса: 0x00000000

Таблица 507 – Регистр сброса состояния ожидания обслуживания

| Биты | Поле | Функция |
|---|---|---|
| 31…0 | CLRPEND | При записи:<br>1 – сбрасывает состояние ожидания обслуживания;<br>0 – не оказывает влияния.<br>При чтении:<br>1 – прерывание в состоянии ожидания обслуживания;<br>0 – прерывание не в состоянии ожидания обслуживания. |

### 30.7 Регистр приоритета прерываний

Этот регистр используется для присвоения приоритета от нулевого до третьего каждому из доступных прерываний. Ноль – наивысший приоритет, а три – самый низший. Два бита приоритета хранятся в битах [7:6] каждого байта.

Адрес регистра: 0xE000E400- 0xE000E41C

Доступ: Чтение/запись

Значение после сброса: 0x00000000

Таблица 508 – Назначение бит

| | 31 30 | 29 24 | 23 22 | 21 16 | 15 14 | 13 8 | 7 6 | 5 0 |
|---|---|---|---|---|---|---|---|---|
| E000E400 | IP_3 | | IP_2 | | IP_1 | | IP_0 | |
| E000E404 | IP_7 | | IP_6 | | IP_5 | | IP_4 | |
| E000E408 | IP_11 | | IP_10 | | IP_9 | | IP_8 | |
| E000E40C | IP_15 | | IP_14 | | IP_13 | | IP_12 | |
| E000E410 | IP_19 | Зарезерв. | IP_18 | Зарезерв. | IP_17 | Зарезерв. | IP_16 | Зарезер. |
| E000E414 | IP_23 | | IP_22 | | IP_21 | | IP_20 | |
| E000E418 | IP_27 | | IP_26 | | IP_25 | | IP_24 | |
| E000E41C | IP_31 | | IP_30 | | IP_29 | | IP_28 | |

Таблица 509 – Регистр приоритета прерываний

| Биты | Поле | Функция |
|---|---|---|
| 7…6 | IP_n | Приоритет прерывания n |

## 30.8 Прерывания по уровню и по фронту

Процессор поддерживает два вида прерываний: по уровню и по фронту. Уровень сигнала удерживается установленным до тех пор, пока не будет сброшен программой обработки прерывания (ISR) устройства. Прерывание по фронту, заключается в том, что процессор сэмплирует линию прерывания по переднему фронту синхросигнала. Процессор распознает фронт, когда наблюдается низкий уровень сигнала и высокий уровень сигнала в двух его последовательных выборках по переднему фронту процессорного синхросигнала.

Для прерываний по уровню, если сигнал не снимается, прежде чем происходит возврат из обработчика прерывания, то прерывание повторно активизируется и ожидает обслуживания. Это часто применяется для FIFO и в устройствах, основанных на буферах, потому что гарантирует считывание информации либо одним ISR, либо повторным запросом без дополнительной работы. Это означает, что устройство удерживает сигнал прерывания установленным, пока устройство пусто. Формирование запроса на прерывание по уровню происходит при условии удержания сигнала не менее двух тактов процессорного ядра.

Прерывания по фронту должны быть установлены, по крайней мере, один процессорный такт процессора, чтобы NVIC отследил их.

Прерывания по фронту могут быть заново установлены во время ISR, поэтому прерывания могут быть активными и ожидать обслуживания в одно и то же время. Должно гарантироваться условие, что второй фронт не придет прежде, чем первый фронт, вызвавший прерывание будет активирован. Если второй фронт придет прежде, чем прерывание активируется, то второй фронт не окажет никакого воздействия, так как обслуживание запущено. Когда ISR активирован, бит ожидания обслуживания очищается. Если прерывание вызывается вновь, когда ISR активирован, NVIC защелкивает опять бит ожидания обслуживания.

Прерывания по фронту используются в основном для внешних сигналов и для частотных или повторяющихся сигналов.

## 30.9 Повторная выборка уровня прерываний

ISR может детектировать, происходили или нет прерывания при обработке текущего прерывания, для того чтобы избежать потерь на вход и выход из обработчика прерывания. Эта информация доступна в регистрах ISPR и ICPR.

Для прерываний по фронту бит, который установился в единицу, показывает, что другое прерывание произошло с момента старта ISR.

Если прерывание по уровню было очищено, а затем установлено, то статусный бит, считанный из регистров ISPR и ICPR, установленный в «1», аналогичен ситуации с прерыванием по фронту.

Для прерываний по уровню, у которых линия сигнала может оставаться произвольно долго с момента входа в ISR, необходимо записать соответствующий бит в ISPR или ICPR. Регистр ICPR не очистится, если линия прерывания в состоянии «1», и может быть считан повторно, чтобы определить статус.

### 30.10 Прерывания как входы общего назначения

Вы можете использовать линии прерываний как линии общего назначения. При таком использовании необходимо гарантировать, что прерывания запрещены в регистре ICER.

Вы можете использовать ICPR, чтобы проверить перешел ли вход в состояние «1» с момента последней проверки.

Для проверки текущего состояния, необходимо записать «1» в соответствующий бит ICPR. Значение статусного бита очистится, если состояние линии «0», и ICPR может быть считан вновь, чтобы определить статус.

# 31 Блок управления системой ядра

Блок управления системой (SCB – System control block) обеспечивает доступ к информации о конфигурации и управление работой системы. Регистры блока управления системой представлены в таблице 510.

Таблица 510 – Регистры блока управления системой ядра

| Имя регистра | Тип операции | Адрес | Значение после сброса |
|---|---|---|---|
| Вспомогательный управляющий регистр (ACTLR) | R/W | 0xE000E008 | Нули в старших 28 битах, состояние вывода ITCMLAEN в бите [3]. Нули в младших трех битах. |
| Регистр управления и статуса SysTick (CTRL) | R/W | 0xE000E010 | 0x00000004 |
| Регистр перегружаемого значения SysTick (LOAD) | R/W | 0xE000E014 | 0x00000000 |
| Регистр текущего значения SysTick (VAL) | R/W очистка | 0xE000E018 | 0x00000000 |
| Регистр калибровочного значения SysTick (CALIB) | RO | 0xE000E01C | 0x80000000 |
| Регистр CPUID | RO | 0xE000ED00 | 0x411CC210 |
| Регистр управления состоянием прерываний (ICSR) | a | 0xE000ED04 | 0x00000000 |
| Регистр управления прерываниями и программным сбросом (AIRCR) | B | 0xE000ED0C | 0xFA050000[c]<br>0xFA058000[d] |
| Регистр конфигурации и управления (CCR) | R/W | 0xE000ED14 | 0x00000208 |
| Регистр приоритета системного обработчика 2 (SHP2) | R/W | 0xE000ED1C | 0x00000000 |
| Регистр приоритета системного обработчика 3 (SHP3) | R/W | 0xE000ED20 | 0x00000000 |
| Регистр управления и состояния системного обработчика (SHCSR) | R/W | 0xE000ED24 | 0x00000000 |

[a] Тип доступа зависит от конкретного бита;
[b] Тип доступа зависит от конкретного бита;
[c] Значение сброса для little-endian;
[d] Значение сброса для big-endian

Все регистры SCB доступны только с использованием словных транзакций. Любая попытка записать полуслово или байт вызывает искажение бит регистра.

## 31.1 Вспомогательный управляющий регистр (ACTLR)

Применяется для разрешения дополнительных верхнего и нижнего адресного пространства инструкций ITCM.

Адрес: 0xE000E008

Тип доступа: Чтение/запись

Значение после сброса: Верхние 28 бит нули, состояние вывода ITCMLAEN в бите [3], нули в трех младших битах.

Таблица 511 – Вспомогательный управляющий регистр (ACTLR)

| Биты | Поле | Функция |
|---|---|---|
| 31…5 | - | Зарезервировано |
| 4 | ITCMUAEN | Разрешение верхнего адресного пространства инструкций ITCM |
| 3 | ITCMLAEN | Разрешение нижнего адресного пространства инструкций ITCM |
| 2…0 | - | |

Когда установлен бит ITCMLAEN, все допустимые инструкции и данные считываются с адресного пространства 0x00000000 – 0x0001FFFF (128 КБ флэш-памяти на кристалле) через интерфейс ITCM. Когда бит ITCMLAEN очищен, этот доступ выполняется через внешний интерфейс AHB-Lite с адресным пространством 1 МБ.

Когда установлен бит ITCMUAEN, все допустимые инструкции и данные считываются с адресного пространства 0x10000000-0x1000FFFF через интерфейс ITCM. Когда бит ITCMUAEN сброшен, этот доступ выполняется через внешний интерфейс AHB-Lite.

## 31.2 Регистр управления и статуса SysTick (CTRL)

Адрес: 0xE000E010

Тип доступа: Чтение/запись

Значение после сброса: 0x00000004

Таблица 512 – Регистр управления и статуса SysTick (CTRL)

| Биты | Поле | Функция |
|---|---|---|
| 31…17 | - | Зарезервировано |
| 16 | COUNTFLAG | Возвращает 1, если таймер досчитал до нуля с последнего момента чтения. Очищается при чтении приложением или отладчиком |
| 15….3 | - | Зарезервировано |
| 2 | CLKSOURCE | Всегда читается как единица.<br>1 – синхросигнал процессора.<br>Признак того, что SysTick использует процессорный синхросигнал HCLK |
| 1 | TCKINT | Бит разрешения прерывания от системного таймера:<br>0 – если таймер досчитал до нуля, то прерывание не возникает;<br>1 – если таймер досчитал до нуля, то возникает запрос на прерывание.<br>Программное обеспечение может использовать бит COUNTFLAG, чтобы определить досчитал таймер до нуля или нет |

| Биты | Поле | Функция |
|---|---|---|
| 0 | ENABLE | Разрешение работы таймера:<br>1 – работа счетчика разрешена. Это означает, что счетчик загружает значение Reload и начинает обратный счет. При достижении нуля, устанавливается флаг COUNTFLAG в «1», и дополнительно, в зависимости от TCKINT, формируется запрос на обслуживание прерывания (SysTick_Handler) от системного таймера. Затем загружается значение Reload и опять начинается счет;<br>0 – счетчик отключен |

## 31.3 Регистр перегружаемого значения SysTick (LOAD)

Регистр используется для определения стартового значения, загружаемого в регистр текущего значения SysTick, когда счетчик достигает нуля. Значение Reload может быть любым в диапазоне 0x00000001-0x00FFFFFF. Значение 0 допустимо, но не оказывает эффекта, потому что запрос на прерывание и установка бита COUNTFLAG регистра CTRL происходит только при переходе таймера из состояния 1 в 0.

Расчет значения Reload происходит в соответствии с использованием таймера:

– Для формирования короткого интервала времени с периодом N процессорных тактов, применяется значение RELOAD равное N-1. Например, если требуется прерывание каждые 100 циклов, то устанавливается значение RELOAD равное 99.

– Для формирования одиночного прерывания после задержки в N тактов процессора, используется значение N. Например, если требуется прерывание после 400 тактов процессора, то устанавливается RELOAD равное 400.

Адрес: 0xE000E014

Тип доступа: Чтение/запись

Значение после сброса: 0x00000000

Таблица 513 – Регистр перегружаемого значения SysTick (LOAD)

| Биты | Поле | Функция |
|---|---|---|
| 31…24 | - | Зарезервировано |
| 23…0 | Reload | Значение, загружаемое в регистр текущего значения SysTick, когда счетчик достигает нуля |

## 31.4 Регистр текущего значения SysTick (VAL)

Используется для определения текущего значения таймера SysTick.

Адрес: 0xE000E018

Тип доступа: Чтение/запись очистка

Значение после сброса: 0x00000000

Таблица 514 – Регистр текущего значения SysTick (VAL)

| Биты | Поле | Функция |
|---|---|---|
| 31…24 | - | Зарезервировано |
| 23…0 | Current | Чтение возвращает текущее значение системного таймера. |

| | | |
|---|---|---|
| | | Запись любого значения очищает регистр в ноль и также очищает бит COUNTFLAG регистра CTRL |

## 31.5 Регистр калибровочного значения SysTick (CALIB)

Регистр используется программным обеспечением для масштабирования до любой желаемой скорости, используя деление и умножение.

Адрес: 0xE000E01C

Тип доступа: Чтение

Значение после сброса: 0x80000000

Таблица 515 – Регистр калибровочного значения SysTick (CALIB)

| Биты | Поле | Функция |
|---|---|---|
| 31 | NOREF | Читается как единица. Показывает, что отдельный синхросигнал таймера не поддерживается |
| 30 | SKEW | Читается как ноль. Калибровочное значение с неточностью синхронизации 10 мс, так как неизвестно TENMS |
| 29…24 | | Зарезервировано |
| 23…0 | TENMS | Читается как ноль. Показывает, что калибровочное значение неизвестно |

## 31.6 Регистр CPUID

При чтении регистра можно определить:

- Номер ID процессорного ядра;
- Номер версии процессорного ядра;
- Подробности реализации ядра.

Адрес: 0xE000ED00

Тип доступа: Чтение

Значение после сброса: 0x411CC210

Таблица 516 – Регистр CPUID

| Биты | Поле | Функция |
|---|---|---|
| 31...24 | IMPLEMENTER | Код производителя<br>0x41 – ARM |
| 23...20 | VARIANT | Исполнения определяет номер варианта:<br>0x0 – для r0p0 и r0p1;<br>0x1 – для r1p0 |
| 19…6 | Constant | Читается как 0xC |
| 5...4 | PARTNO | Номер процессора в пределах семейства<br>0xC21 |
| 3…0 | REVISION | Исполнение определяет номер ревизии:<br>0x0 – для r0p0 и r1p0;<br>0x1 – для r0p1 |

## 31.7 Регистр управления состоянием прерываний (ICSR)

Регистр используется для:

– установки состояния ожидания обслуживания NMI;

– установки или сброса состояния ожидания обслуживания для PendSV;

– установки или сброса состояния ожидания обслуживания для SysTick;

– проверки состояния ожидания обслуживания для исключений;

– определения номера вектора исключения наивысшего приоритета, ожидающего обслуживания;

– определения номера вектора активного исключения.

Адрес: 0xE000ED04

Тип доступа: Зависит от индивидуальных бит

Значение после сброса: 0x00000000

Таблица 517 – Регистр управления состоянием прерываний (ICSR)

| Биты | Поле | Тип доступа | Функция |
|---|---|---|---|
| 31 | NMIPENDSET | R/W | При записи:<br>1 – устанавливает запрос обслуживания для NMI;<br>0 – не оказывает влияния.<br>NMIPENDSET ожидает обслуживания и активирует NMI. Так как NMI имеет наивысший приоритет, то его обслуживание начинается сразу, как только обнаружено, за исключением случая, когда процессор имеет приоритет -2.<br>При чтении возвращает информацию о состоянии ожидания обслуживания NMI |
| 30, 29 | - | - | Зарезервировано |
| 28 | PENDSVSET | R/W | Бит установки состояния ожидания обслуживания для исключения PendSV.<br>При записи:<br>0 – не влияет на работу системы;<br>1 – переводит исключение PendSV в состояние ожидания обслуживания.<br>При чтении:<br>0 – исключение PendSV не ожидает обслуживания;<br>1 – исключение PendSV ожидает обслуживания.<br>Запись «1» в разряд PENDSVSET это единственный возможный способ перевода исключения PendSV в состояние ожидания обслуживания |
| 27 | PENDSVCLR | WO | Бит сброса состояния ожидания обслуживания для исключения PendSV.<br>При записи:<br>0 – не влияет на работу системы;<br>1 – сбрасывает состояние ожидания обслуживания для исключения PendSV |

| Биты | Поле | Тип доступа | Функция |
|---|---|---|---|
| 26 | PENDSTSET | R/W | Бит установки состояния ожидания обслуживания для исключения SysTick.<br>При записи:<br>0 – не влияет на работу системы;<br>1 – переводит исключение SysTick в состояние ожидания обслуживания.<br>При чтении:<br>0 – исключение SysTick не ожидает обслуживания;<br>1 – ожидает |
| 25 | PENDSTCLR | WO | Бит сброса состояния ожидания обслуживания для исключения SysTick.<br>При записи:<br>0 – не влияет на работу системы;<br>1 – сбрасывает состояние ожидания обслуживания для исключения SysTick |
| 24 | - | | Зарезервировано |
| 23 | ISRPREEMPT[a] | RO | Этот бит используется в режиме отладки. Бит сигнализирует, что прерывание, ожидающее обслуживания, станет активным при запуске следующего цикла. Если бит C_MASKINTS очищен в регистре управления и статуса отладочного режима останова, то прерывание обслуживается следующим образом:<br>1 – ожидающее обслуживание исключение обслуживается при выходе из состояния останова режима отладки;<br>0 – ожидающее обслуживание исключение не обслуживается |
| 22 | ISRPENDING[a] | RO | Флаг наличия в системе прерываний, ожидающих обслуживания.<br>0 – ожидающие обслуживания прерывания отсутствуют;<br>1 – присутствуют |
| 21…18 | - | | Зарезервировано |
| 17…12 | VECTPENDING | RO | Содержит номер исключения, ожидающего обслуживания, с наивысшим приоритетом, обработка которого в системе разрешена.<br>0 – необслуженных исключений нет;<br>5’bXXXXXX – номер ожидающего обслуживания исключения.<br>Значение данного поля не учитывает влияние поля PRIMASK |
| 11...6 | - | | Зарезервировано |
| 5…0 | VECTACTIVE[b] | RO | Содержит номер активного исключения.<br>0 – режим Thread;<br>6’bXXXXXX – номер[b] текущего обслуживаемого исключения |

| Биты | Поле | Тип доступа | Функция |
|---|---|---|---|
| [a] Только для режима отладки;<br>[b] Это значение аналогично битам [5:0] регистра IPSR | | | |

## 31.8 Регистр управления прерываниями и программным сбросом (AIRCR)

Этот регистр используется для:

– определения порядка следования байт в слове (endianness) при доступе к данным;

– очистка всей информации об активных состояниях из отладочного режима останова;

– запроса сброса системы.

Адрес: 0xE000ED0C

Тип доступа: Зависит от индивидуальных бит

Значение после сброса: 0xFA050000 в случае режима данных little-endian
0xFA058000 в случае режима данных big-endian

Таблица 518 – Регистр управления прерываниями и программным сбросом (AIRCR)

| Биты | Поле | Тип доступа | Функция |
|---|---|---|---|
| 31...16 | VECTKEY | WO | Ключ доступа к регистру. При записи должен быть равен 0x05FA, в противном случае попытка записи в регистр будет проигнорирована процессором |
| 15 | ENDIANNESS | RO | Порядок следования значащих разрядов при доступе к данным.<br>0 – младший байт идет первым (little-endian);<br>1 – старший байт идет первым (big-endian) |
| 14….3 | - | | Зарезервировано |
| 2 | SYSRESETREQ | WO | Запись «1» в этот бит вызовет установку сигнала SYSRESETREQ на выходе системы для запроса сброса. В результате произойдет сброс всей системы и основных компонентов за исключением отладочных. Бит C_HALT в регистре DHCSR очистится как результат запроса на сброс системы. Но отладчик не потеряет связь с устройством |

| Биты | Поле | Тип доступа | Функция |
|---|---|---|---|
| 1 | VECTCLRACTIVE | WO | Очищает всю информацию об активных состояниях фиксированных и конфигурируемых исключениях.<br>Этот бит:<br>– самоочищающийся;<br>– может быть установлен только DAP в режиме останова процессора.<br>Когда этот бит установлен:<br>– очищается статус всех активных исключений процессора;<br>– принудительный возврат в режим Thread;<br>– принудительная установка IPSR в ноль |
| 0 | - | | Зарезервировано |

## 31.9 Регистр конфигурации и управления (CCR)

Это регистр используется для разрешения выравнивания стека и служит причиной Hard Fault в случае невыровненного доступа.

Адрес: 0xE000ED14

Тип доступа: Чтение

Значение после сброса: 0x00000208

Таблица 519 – Регистр конфигурации и управления (CCR)

| Биты | Поле | Функция |
|---|---|---|
| 31...10 | - | Зарезервировано |
| 9 | STKALIGN | Всегда в «1». Вход в любое исключение происходит с восьмибайтовым выравниванием стека и это сохраняется при восстановлении контекста. SP восстанавливается при соответствующем возвращении из исключения |
| 8…4 | - | Зарезервировано |
| 3 | UNALIGN_TRP | Показывает, что любой невыровненный доступ приводит к Hard Fault. Ловушка для невыровненного доступа устанавливается в «1» |
| 2…0 | - | Зарезервировано |

## 31.10 Регистры приоритета системных обработчиков

Системные обработчики – это специальный класс обработчиков исключений, которые могут иметь свой приоритет, установленный в любое значение из уровней приоритета.

Существует два регистра приоритета системных обработчиков для задания приоритета следующим системным обработчикам:

- SVCall;
- SysTick;
- PendSV.

PendSV и SVCall постоянно разрешены. Вы можете разрешить или запретить SysTick записью в регистр управления и статуса SysTick.

## 31.11 Регистр приоритета системного обработчика 2 (SHP2)

Адрес: 0xE000ED1C

Тип доступа: Чтение/Запись

Значение после сброса: 0x00000000

Таблица 520 – Регистр приоритета системного обработчика 2 (SHP2)

| Биты | Поле | Функция |
|---|---|---|
| 31…30 | PRI_11 | Приоритет системного обработчика 11, SVCall |
| 29…0 | - | Зарезервировано |

## 31.12 Регистр приоритета системного обработчика 3 (SHP3)

Адрес: 0xE000ED20

Тип доступа: Чтение/Запись

Значение после сброса: 0x00000000

Таблица 521 – Регистр приоритета системного обработчика 3 (SHP3)

| Биты | Поле | Функция |
|---|---|---|
| 31…30 | PRI_15 | Приоритет системного обработчика 15, SysTick |
| 29…24 | - | Зарезервировано |
| 23…22 | PRI_14 | Приоритет системного обработчика 14, PendSV |
| 21…0 | - | Зарезервировано |

### 31.13 Регистр управления и состояния системного обработчика (SHCSR)

Этот регистр используется для чтения или записи статуса ожидания обслуживания исключения SVCall.

Адрес: 0xE000ED24

Тип доступа: Чтение/Запись

Значение после сброса: 0x00000000

Таблица 522 – Регистр управления и состояния системного обработчика (SHCSR)

| Биты | Поле | Функция |
|---|---|---|
| 31…16 | - | Зарезервировано |
| 15 | SVCALLPENDED | Читается как 1, если SVCall ожидает обслуживания.<br>При записи:<br>1 – установка состояния ожидания обслуживания SVCall;<br>0 – сброс состояния ожидания обслуживания для SVCall. |
| 14...0 | - | Зарезервировано |

Этот регистр доступен только как часть отладки и не доступен через карту памяти.

# 32 Сторожевые таймеры

## 32.1 Описание регистров блока сторожевых таймеров

Таблица 523 – Описание регистров блока сторожевого таймера IWDG

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x4006_8000 | 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 524 – Описание регистров блока сторожевого таймера WWDG

| Базовый Адрес | Название | Описание |
|---|---|---|
| 0x4006_0000 | 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 |

### 32.1.1 Блок сторожевого таймера IWDG

Блок сторожевого таймера 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 142.

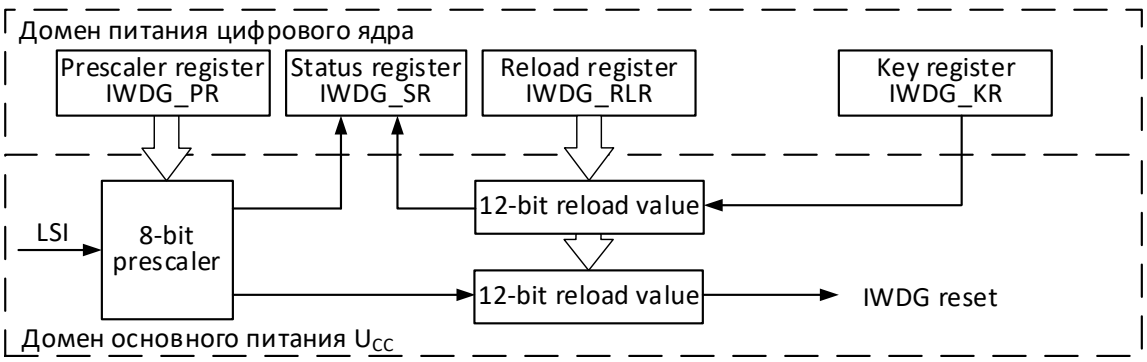



Рисунок 142 – Блок-схема независимого сторожевого таймера

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 $U_{CC}$ и тактируется от генератора LSI, поэтому для фактической записи в регистры IWDG в домене основного питания $U_{CC}$ генератор LSI должен находиться в рабочем режиме. Часть сторожевого таймера в домене основного питания $U_{CC}$ 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 $U_{CC}$.

Пример работы с блоком IWDG:

1 Включение генератора LSI и ожидание его выхода в рабочий режим.

2 Разрешение доступа к регистрам IWDG_PR и IWDG_RLR записью значения 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.

#### 32.1.1.1 *Регистр ключа IWDG_KR*

Таблица 525 – Регистр IWDG_KR

| Номер | 31…16 | 15…0 |
|---|---|---|
| Доступ | U | W |
| Сброс | 0 | 0 |
| | - | KEY[15:0] |

Таблица 526 – 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).<br>При записи:<br>– 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;<br>– 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;<br>– 0xCCCC – разрешение работы сторожевого таймера (если работа сторожевого таймера не разрешена; работа таймера останавливается аппаратно сбросом от IWDG) |

#### 32.1.1.2 *Делитель частоты сторожевого таймера IWDG_PR*

Таблица 527 – Регистр IWDG_PR

| Номер | 31….3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Доступ | U | R/W | R/W | R/W |
| Сброс | 0 | 0 | 0 | 0 |
| | - | PR2 | PR1 | PR0 |

Таблица 528 – 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.<br>000 – делитель на 4: LSI/4;<br>001 – делитель на 8: LSI/8;<br>010 – делитель на 16: LSI/16;<br>011 – делитель на 32: LSI/32;<br>100 – делитель на 64: LSI/64;<br>101 – делитель на 128: LSI/128;<br>110 – делитель на 256: LSI/256;<br>111 – делитель на 256: LSI/256.<br>Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR.<br>При записи в регистр формируется запрос на обновление значения делителя частоты в часть IWDG в домене основного питания (см. Рисунок 142).<br>Запись в регистр IWDG_PR недоступна при фактической записи предделителя в часть IWDG в домене основного питания $U_{CC}$ (см. см. Рисунок 142).<br>При чтении возвращает записанное в регистр значение, а не значение предделителя IWDG в домене основного питания |

#### 32.1.1.3 *Регистр основания счета сторожевого таймера IWDG_PRL*

Таблица 529 – Регистр IWDG_PRL

| Номер | 31...12 | 11…0 |
|---|---|---|
| Доступ | U | R/W |
| Сброс | 0 | 0xFFF |
| | - | RLR[11:0] |

Таблица 530 – Описание бит регистра IWDG_PRL

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...12 | - | Зарезервировано |
| 11...0 | PRL[11:0] | Значение перезагрузки (основания счета) сторожевого таймера.<br>Сторожевой таймер декрементируется, начиная с этого значения.<br>Определяет значение, загружаемое в сторожевой таймер при записи значения 0xAAAA в регистр IWDG_KR.<br>Для возможности записи в регистр должен быть разрешен доступ по записи с помощью ключевого значения 0x5555 в регистре IWDG_KR.<br>Запись в регистр IWDG_RLR недоступна при фактической записи основания счета в часть IWDG в домене основного питания $U_{CC}$ (см. Рисунок 142).<br>При чтении возвращает записанное в регистр значение, а не значение перезагрузки IWDG в домене основного питания |

#### 32.1.1.4 *Регистр статуса сторожевого таймера IWDG_SR*

Таблица 531 – Регистр IWDG_SR

| Номер | 31…2 | 1 | 0 |
|---|---|---|---|
| Доступ | U | R | R |
| Сброс | 0 | 0 | 0 |
| | - | RVU | PVU |

Таблица 532 – 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 | Устанавливается аппаратно и служит признаком того, что обновляется значение сторожевого таймера IWDG в домене основного питания $U_{CC}$ из регистра перезагрузки IWDG_RLR (см. Рисунок 142).<br>**Работа IWDG разрешена и LSI работает:** Отображает запрос от регистров управления на обновление значения сторожевого таймера значением регистра IWDG_RLR и/или фактическое обновление значения сторожевого таймера в домене основного питания $U_{CC}$. Сбрасывается при завершении фактического обновления значения перезагрузки. Значение регистра перезагрузки IWDG_RLR может быть обновлено только, если этот бит равен нулю.<br>**Работа IWDG не разрешена или LSI не работает:** Отображает запрос от регистров управления на обновление значения сторожевого таймера в домене основного питания $U_{CC}$ значением регистра IWDG_RLR. Сбрасывается любым типом сброса. Значение регистра перезагрузки IWDG_RLR может быть обновлено независимо от состояния этого бита – после запуска IWDG будет обновлен последним значением, записанным в IWDG_RLR |
| 0 | PVU | Флаг обновления делителя частоты сторожевого таймера.<br>Устанавливается аппаратно и служит признаком того, что обновляется значение делителя частоты IWDG в домене основного питания $U_{CC}$ из регистра предделителя IWDG_PR (см. Рисунок 142).<br>**Работа IWDG разрешена и LSI работает:** Отображает запрос от регистров управления на обновление делителя частоты сторожевого таймера значением регистра IWDG_PR и/или фактическое обновление значения делителя в домене основного питания $U_{CC}$. Сбрасывается при завершении фактического обновления делителя. Значение регистра предделителя IWDG_PR может быть обновлено только, если этот бит равен нулю.<br>**Работа IWDG не разрешена или LSI не работает:** Отображает запрос от регистров управления на обновление значения сторожевого таймера в домене основного питания $U_{CC}$ значением регистра IWDG_PR. Сбрасывается любым типом сброса. Значение регистра перезагрузки IWDG_PR может быть обновлено независимо от состояния этого бита – после запуска IWDG будет обновлен последним значением, записанным в IWDG_PR |

### 32.1.2 Блок сторожевого таймера WWDG

#### 32.1.2.1 *Регистр управления WWDG_CR*

Таблица 533 – Регистр WWDG_CR

| Номер | 31...8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Доступ | U | R/S | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Сброс | - | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | - | WDGA | T6 | T5 | T4 | T3 | T2 | T1 | T0 |

Таблица 534 – Описание бит регистра WWDG_CR

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...8 | - | Зарезервировано |
| 7 | WDGA | Бит активации.<br>Этот бит устанавливается программно и очищается только аппаратно при сбросе. Когда WDGA=1, сторожевой таймер может генерировать сброс.<br>1 – сторожевой таймер включен;<br>0 – сторожевой таймер отключен |
| 6...0 | T[6:0] | Значение семиразрядного счетчика (от старших разрядов к младшим).<br>Эти биты содержат значение сторожевого таймера, который декрементируется каждые 4096•2•WGTB циклов частоты PCLK периферийной шины APB |

#### 32.1.2.2 *Регистр конфигурации WWDG_CFR*

Таблица 535 – Регистр WWDG_CFR

| Номер | 31…10 | 9 | 8 | 7 | 6 | 5 |
|---|---|---|---|---|---|---|
| Доступ | U | R/S | R/W | R/W | R/W | R/W |
| Сброс | - | 0 | 0 | 0 | 1 | 1 |
| | - | EWI | WDGTB1 | WDGTB0 | W6 | W5 |

| Номер | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Доступ | R/W | R/W | R/W | R/W | R/W |
| Сброс | 1 | 1 | 1 | 1 | 1 |
| | W4 | W3 | W2 | W1 | W0 |

Таблица 536 – Описание бит регистра WWDG_CFR

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...10 | - | Зарезервировано |

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 9 | EWI | Раннее предупреждающее прерывание.<br>Если бит установлен, то разрешается генерация прерывания при достижении сторожевым таймером значении 0x40. Прерывание запрещается только аппаратным сбросом |
| 8..7 | WGTB[1:0] | Делитель частоты сторожевого таймера.<br>2'b00 – частота таймера (PCLK / 4096) /1;<br>2'b01 – частота таймера (PCLK / 4096) /2;<br>2'b10 – частота таймера (PCLK / 4096) /4;<br>2'b11 – частота таймера (PCLK / 4096) /8 |
| 6..0 | W[6:0] | Значение окна.<br>Эти биты содержат значение окна, в пределах которого возможна инициализация битов T[6:0] значением в пределах 0x40 – 0x7F. Если происходит инициализация битов в момент T > W, то формируется сброс на выходе RESET. Если таймер достигнет значения T = 0x3F, то также формируется сброс |

#### 32.1.2.3 *Регистр статуса WWDG_SR*

Таблица 537 – Регистр WWDG_SR

| Номер | 31…1 | 0 |
|---|---|---|
| Доступ | U | R/C |
| Сброс | - | 0 |
| | - | EWIF |

Таблица 538 – Описание бит регистра WWDG_CFR

| Номер бита | Функциональное имя бита | Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений |
|---|---|---|
| 31...1 | | Зарезервировано |
| 0 | EWIF | Флаг раннего предупреждающего прерывания.<br>Этот бит устанавливается аппаратно, когда сторожевой таймер достигает значения 0x40. Бит очищается программно записью нуля. Запись «1» не влияет. Этот бит также устанавливается, если прерывание запрещено EWI = 0 |

# 33 Электрические параметры

Таблица 539 – 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 низкого уровня, В | $U_{OL}$ | – | 0,4 | 25, 85, – 40 |
| Выходное напряжение высокого уровня, В | $U_{OH}$ | 2,4 | – | |
| Уровень напряжения питания для срабатывания схемы генерации сброса, В | $U_{POR}$ | 1,8 | 2,2 | |
| Входной ток утечки низкого уровня, мкА, на выводах: PA, PB, PC, PD, PE, PF, DN, DP, RESET, WAKEUP, ITCMLAEN, JTAGEN, TCK, TDI, TMS, TRST при: $U_{CC} = U_{CCA} = 3{,}6$ В, $U_I = 0$ В | $I_{ILL1}$ | –1,0 | 1,0 | |
| Входной ток утечки низкого уровня, мкА, на выводе: OSC_IN при: $U_{CC} = U_{CCA} = 3{,}6$ В, $U_I = 0$ В | $I_{ILL2}$ | – 1,0 | 1,0 | |
| Входной ток утечки высокого уровня, мкА, на выводах: PD (7-15), PE (0-2, 6-7), DP, DN при: $U_{CC} = U_{CCA} = 3{,}6$ В, $U_I = 3{,}6$ В на выводах: PA, PB, PC, PD (0-6), PE (3-5, 8-15), PF, RESET, WAKEUP, ITCMLAEN, JTAGEN, TCK, TDI, TMS, TRST при: $U_{CC} = U_{CCA} = 3{,}6$ В, $U_I = 5{,}25$ В | $I_{ILH1}$ | – 1,0 | 1,0 | |
| Входной ток утечки высокого уровня, мкА, на выводе: OSC_IN при: $U_{CC} = U_{CCA} = 3{,}6$ В, $U_I = 3{,}6$ В | $I_{ILH2}$ | – 40,0 | 40,0 | |
| Статический ток потребления в режиме сброса, мА | $I_{CCS}$ | – | 15 | |
| Динамический ток потребления, мА, на выводах Ucc, AUcc, BUcc при $U_{CC} = U_{CCA} = U_{CCB} = 3{,}6$ В, $f_C = 144$ МГц | $I_{OCC_MC}$ | – | 360 | |
| Динамический ток потребления, мА, на выводах VDD1A – VDD4A, при $U_{CC} = U_{CCA} = U_{CCB} = 3{,}6$ В | $I_{OCC_PHY}$ | – | 180 | |
| Выходная частота LSI RC-генератора, кГц | $f_{O_LSI}$ | 10 | 60 | |
| Выходная частота HSI RC-генератора, МГц | $f_{O_HSI}$ | 6 | 10 | |
| Выходная частота PLL, МГц максимальная минимальная | $f_{O_PLL}$ | 144 – | – 6 | |
| Параметры АЦП | | | | |
| Разрядность АЦП | $E_{NADC}$ | 12 | – | 25, |

| Наименование параметра, единица измерения, режим измерения | Буквенное обозначение параметра | Норма параметра | | Температура среды, °С |
|---|---|---|---|---|
| | | не менее | не более | |
| Дифференциальная нелинейность АЦП, единица младшего разряда | $E_{DLADC}$ | – 1,0 | 2,0 | 85, – 40 |
| Интегральная нелинейность АЦП, единица младшего разряда | $E_{ILADC}$ | – 3,0 | 3,0 | 25, 85, – 40 |
| Ошибка смещения АЦП, единица младшего разряда | $E_{OFFADC}$ | – 6,0 | 6,0 | |
| Ошибка усиления АЦП, % | $E_{GAINADC}$ | – 1,0 | 1,0 | |
| Параметры ЦАП | | | | |
| Разрядность ЦАП | $E_{NDAC}$ | 12 | – | 25, 85, – 40 |
| Дифференциальная нелинейность ЦАП, единица младшего разряда | $E_{DLDAC}$ | – 1,0 | 2,0 | |
| Интегральная нелинейность ЦАП, единица младшего разряда | $E_{ILDAC}$ | – 6,0 | 6,0 | |
| Ошибка смещения ЦАП, мВ | $E_{OFFDAC}$ | – 40,0 | 40,0 | |
| Ошибка усиления ЦАП, % | $E_{GAINDAC}$ | – 2,0 | 2,0 | |
| Минимальное выходное напряжение ЦАП, В | $U_{O_DAC\,min}$ | – | 0,08 | |
| Максимальное выходное напряжение ЦАП, В | $U_{O_DAC\,max}$ | $U_{REF(DAC)}$-0,08 | – | |
| Примечание – Знак минус перед значением тока указывает на направление – вытекающий ток | | | | |

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В.

# 34 Предельно-допустимые характеристики

Таблица 540 – Предельно-допустимые и предельные режимы эксплуатации

| Наименование параметра, единица измерения | Буквенное обозначение параметра | Норма параметра | | | |
|---|---|---|---|---|---|
| | | Предельно-допустимый режим | | Предельный режим [1] | |
| | | не менее | не более | не менее | не более |
| Напряжение источника питания, В, на выводах $U_{CC}$, VDD1A-VDD4A | $U_{CC}$ | 3,0 | 3,6 | – | 4,0 |
| Напряжение источника питания АЦП, и ЦАП, В, на выводе $AU_{CC}$ | $U_{CCA}$ [2] | 3,0 | 3,6 | – | 4,0 |
| Напряжение источника питания батарейного домена, В, на выводе $BU_{CC}$ | $U_{CCB}$ | 1,8 | 3,6 | – | 4,0 |
| Входное напряжение низкого уровня, В, (при работе в цифровом режиме) на выводах: PA, PB, PC, PD, PE, PF, RESET, WAKEUP, DN, DP | $U_{IL}$ | 0 | 0,8 | – 0,3 | – |
| на выводе: OSC_IN BYPASS = 1 | | 0 | 0,8 | – 0,3 | – |
| Входное напряжение высокого уровня, В, на выводах: PD (7-15), PE (0-2, 6-7), DN, DP | $U_{IH}$ | 2,0 | $U_{CC}$ | – | $U_{CC}$ +0,3 |
| на выводах: PA, PB, PC, PD (0-6), PE (3-5, 8-15), PF, RESET, WAKEUP | | 2,0 | 5,25 | – | 5,3 |
| на выводе: OSC_IN BYPASS = 1 | | 2,0 | $U_{CC}$ | – | $U_{CC}$ +0,3 |
| на выводе: PF0 (в функции OSC_IN25) BYPASS = 1 | | 2,0 | $U_{CC}$ | – | $U_{CC}$ +0,3 |
| Выходной ток низкого уровня, мА, (при работе в цифровом режиме) на выводах: PA, PB, PC, PD, PE, PF, DN, DP | $I_{OL}$ [3, 4] | – 6 | – | – 10 | – |
| Выходной ток высокого уровня, мА, на выводах: PA, PB, PC, PD, PE (0-5, 8-15), PF, DN, DP | $I_{OH}$ [5] | – | 6 | – | 10 |
| на выводах: PE 6, 7 | | – | 3 | – | 10 |
| Частота следования импульсов тактовых сигналов, МГц | $f_C$ | – | 144 | – | – |
| Частота следования импульсов тактовых сигналов АЦП, МГц | $f_{C_ADC}$ | – | 14 | – | – |
| Частота следования импульсов тактовых сигналов HSE, МГц при: BYPASS = 0 | $f_{C_HSE}$ | 2 | 16 | – | – |
| при: BYPASS = 1 | | – | 144 | – | – |

| Наименование параметра, единица измерения | Буквенное обозначение параметра | Норма параметра | | | |
|---|---|---|---|---|---|
| | | Предельно-допустимый режим | | Предельный режим [1] | |
| | | не менее | не более | не менее | не более |
| Частота следования импульсов тактовых сигналов LSE, кГц<br>при: BYPASS = 0 | $f_{C_LSE}$ | 32 | 33 | – | – |
| при: BYPASS = 1 | | – | 1 000 | – | – |
| Частота следования импульсов тактовых сигналов PLL, МГц | $f_{C_PLL}$ | 6 | 16 | – | – |
| Параметры ЦАП | | | | | |
| Напряжение верхней границы опорного напряжения ЦАП, В,<br>на выводе: REFD0, REFD1 | $U_{REF(DAC)}$ | 2,4 | $U_{CCA}$ | – | – |
| Резистивная нагрузка ЦАП, кОм | $R_{LOAD}$ | 10 | – | – | – |
| Емкостная нагрузка ЦАП, пФ | $C_{LOAD}$ | – | 100 | – | – |
| Параметры АЦП | | | | | |
| Напряжение нижней границы внешнего опорного напряжения АЦП, В | $U_{ADC1_REF-}$ [6] | 0 | $U_{CCA}$-2,4 | – | $U_{CC}$ +0,3 |
| Напряжение верхней границы внешнего опорного напряжения АЦП, В | $U_{ADC0_REF+}$ [6] | 2,4 | $U_{CCA}$ | – | $U_{CC}$ +0,3 |
| Диапазон напряжения внешнего опорного источника АЦП, В,<br>$U_{REF(ADC)} = U_{ADC0_REF+} - U_{ADC1_REF-}$ | $U_{REF(ADC)}$ [6] | 2,4 | $U_{CCA}$ | – | – |
| Диапазон напряжения на входе АЦП, В | $U_{AIN(ADC)}$ | $U_{ADC1_REF-}$ | $U_{ADC0_REF+}$ | – 0,3 | $U_{CC}$ +0,3 |
| Емкость нагрузки, пФ,<br>на выводах: PA, PB, PC, PD, PE, PF | $C_L$ | – | 30 | – | – |
| Время хранения информации, лет,<br>при: T = 25 °C | $t_{GS}$ | 25 | – | – | – |
| при: T = 85 °C | | 10 | – | – | – |

[1] Не допускается одновременное воздействие двух и более предельных режимов;

[2] Допускается использование отдельного источника для питания АЦП и ЦАП, но при этом его выходное напряжение не должно отличаться от $U_{CC}$ более чем на ± 0,2 В;

[3] Суммарный выходной ток высокого уровня $I_{OH}$ не должен превышать 100 мА;

[4] Знак минус перед значением тока указывает на направление – вытекающий ток;

[5] Суммарный выходной ток низкого уровня $I_{OL}$ не должен превышать 100 мА;

[6] При использовании внутреннего опорного напряжения для АЦП $U_{ADC1_REF} = GND_A$ и $U_{ADC0_REF+} = U_{CCA}$, для ЦАП $U_{REF(DAC)} = U_{CCA}$

# 35 Справочные данные

Предельная температура p-n перехода кристалла 150 °С.

Значение емкости входа-выхода на частоте 1 МГц не превышает 12,5 пФ.

Если при применении нагрузочных емкостей расчетных значений генератор LSE в режиме осциллятора генерирует частоту на седьмой гармонике, то рекомендуется использовать нагрузочные емкости номиналом 51 пФ на выводах OSC_IN32 и OSC_OUT32.

Таблица 541 – Справочные данные

| Наименование параметра, единица измерения, режим измерения | Буквенное обозначение параметра | Норма параметра | | Температура среды, °С |
|---|---|---|---|---|
| | | не менее | не более | |
| Ток потребления батарейного домена, мкА, при: $U_{CC} = U_{CCA} = 0$ В | $I_{CC_B}$ | – | 5 | 25, 85, – 40 |
| Длительность фронта/спада входного сигнала, нс, на выводе: OSC_IN при: HSE BYPASS = 1 | $\tau_r$ $\tau_f$ | – | 3,5 | |
| Длительность сигнала «Сброс», мкс | $t_{W(СБРОС)}$ | 20 | – | |
| Время запуска после сброса по POR, мс | $t_{POR}$ | – | 6 | |
| Минимальная длительность сигнала низкого уровня на входе nRESET для формирования сигнала «Сброс», нс | $t_{W(nRESET)min}$ | 200 | – | |
| Максимальная длительность сигнала низкого уровня на входе nRESET, при котором не формируется сигнал «Сброс», нс | $t_{W(nRESET)max}$ | – | 10 | |
| Сопротивление внутренних программируемых резисторов доопределения, кОм | $R_{PULL}$ | 20 | 100 | |
| Параметры ЦАП | | | | |
| Время установления сигнала ЦАП, мкс, при: $U_{CC}$ = 3,6 В, $C_l$ = 100 пФ, $R_l$ = 10 кОм | $t_{SU(DAC)}$ | – | 5,2 | 25, 85, – 40 |
| Время включения ЦАП, мкс, при: $U_{CC}$ = 2,4 В | $t_{ON_DAC}$ | – | 10 | |
| Ток потребления по входу опоры, мкА | $I_{DAC0_UREF}$ | – | 500 | |
| Ток потребления по входу опоры, мкА | $I_{DAC1_UREF}$ | – | 500 | |
| Ток потребления ЦАП, мА, при отсутствии нагрузки | $I_{OCCDAC}$ | – | 2 | |
| Параметры АЦП | | | | |
| Время выборки АЦП, нс | $t_{A_ADC}$ | – | $4 \times f_{C_ADC}$ | 25, 85, – 40 |
| Время преобразования АЦП, нс | $t_{AO_ADC}$ | – | $28 \times f_{C_ADC}$ | |
| Ток потребления по входу внешней верхней границы опоры АЦП, мкА | $I_{ADC0_VREF+}$ | – | 50 | |
| Ток потребления по входу внешней нижней опоры АЦП, мкА | $I_{ADC0_VREF-}$ | – 50 | – | |

| Наименование параметра, единица измерения, режим измерения | Буквенное обозначение параметра | Норма параметра | | Температура среды, °С |
|---|---|---|---|---|
| | | не менее | не более | |
| Ток потребления по питанию АЦП, мА при: fadc=14 МГц, $U_{CCA}$=3,6В | $I_{OCCADC}$ | – | 3 | |
| Минимальная частота преобразования АЦП, кГц | $f_{C_ADCMIN}$ | 10 | – | |

| Наименование параметра, единица измерения, режим измерения | Буквенное обозначение параметра | Норма параметра | | Температура среды, °С |
|---|---|---|---|---|
| | | не менее | не более | |
| Параметры PHY Ethernet | | | | |
| Отношение сигнал/шум АЦП, дБ | $N_{N_RF(PHY)}$ | 28 | – | 25, 85, – 40 |
| Дифференциальная нелинейность АЦП, единица младшего разряда | $E_{DLDAC(PHY)}$ | – 1,6 | 1,6 | |
| Интегральная нелинейность ЦАП, единица младшего разряда | $E_{ILDAC(PHY)}$ | – 1 | 2 | |
| Выходное дифференциальное напряжение передатчика, В, при: скорость передачи данных 100 Мбит/с; | $U_{OD(PHY)}$ | 0,8 | – | |
| Выходное дифференциальное напряжение передатчика, В, при: скорость передачи данных 10 Мбит/с | $U_{OD1(PHY)}$ | 4,4 | 5,6 | |
| Время задержки распространения сигнала приемника при включении\ выключении, нс, при: скорость передачи данных 100 Мбит/с | $t_{PLH_R(PHY)}$ $t_{PHL_R(PHY)}$ | 2 | 5 | |
| Разность задержек распространения сигнала приемника, нс, $\lvert t_{PLH_R} - t_{PHL_R}\rvert$ при: скорость передачи данных 100 Мбит/с | $t_{SKEW_R(PHY)}$ | – | 0,5 | |
| Искажение рабочего цикла передатчика, нс, при: скорость передачи данных 100 Мбит/с | $\Delta CY_{R(PHY)}$ | – | 0,5 | |
| Частота следования импульсов тактовых сигналов PLL, МГц при: скорость передачи данных 100 Мбит/с | $f_{C_PLL(PHY)}$ | 124,875 | 125,125 | |
| Частота следования импульсов тактовых сигналов PLL, МГц при: скорость передачи данных 10 Мбит/с | $f_{C_PLL1(PHY)}$ | 19,99 | 20,01 | |
| Дрожание фронта тактового сигнала PLL, нс при: скорость передачи данных 100 Мбит/с | $\Delta\tau_{_PLL(PHY)}$ | – | 1,4 | |
| Дрожание фронта тактового сигнала PLL, нс при: скорость передачи данных 10 Мбит/с | $\Delta\tau_{_PLL1(PHY)}$ | минус 3,5 | 3,5 | |
| Выходное напряжение на выводе EXRES1, В | $U_{O(PHY)}$ | 1,14 | 1,34 | |

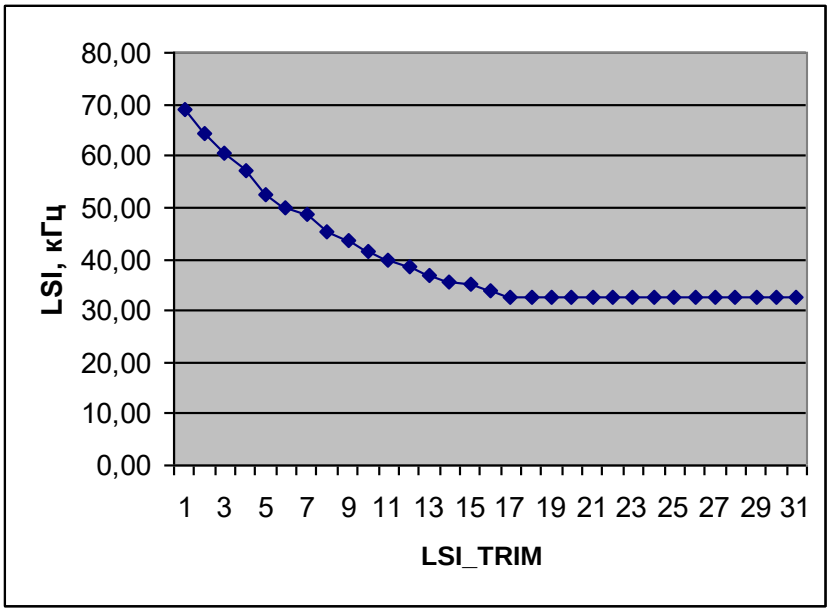



Рисунок 143 – Зависимость частоты LSI от значения LSI_TRIM

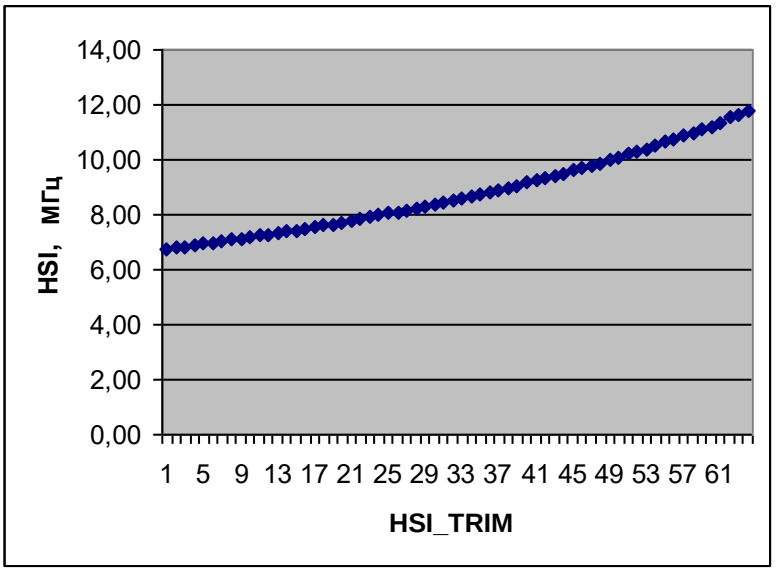
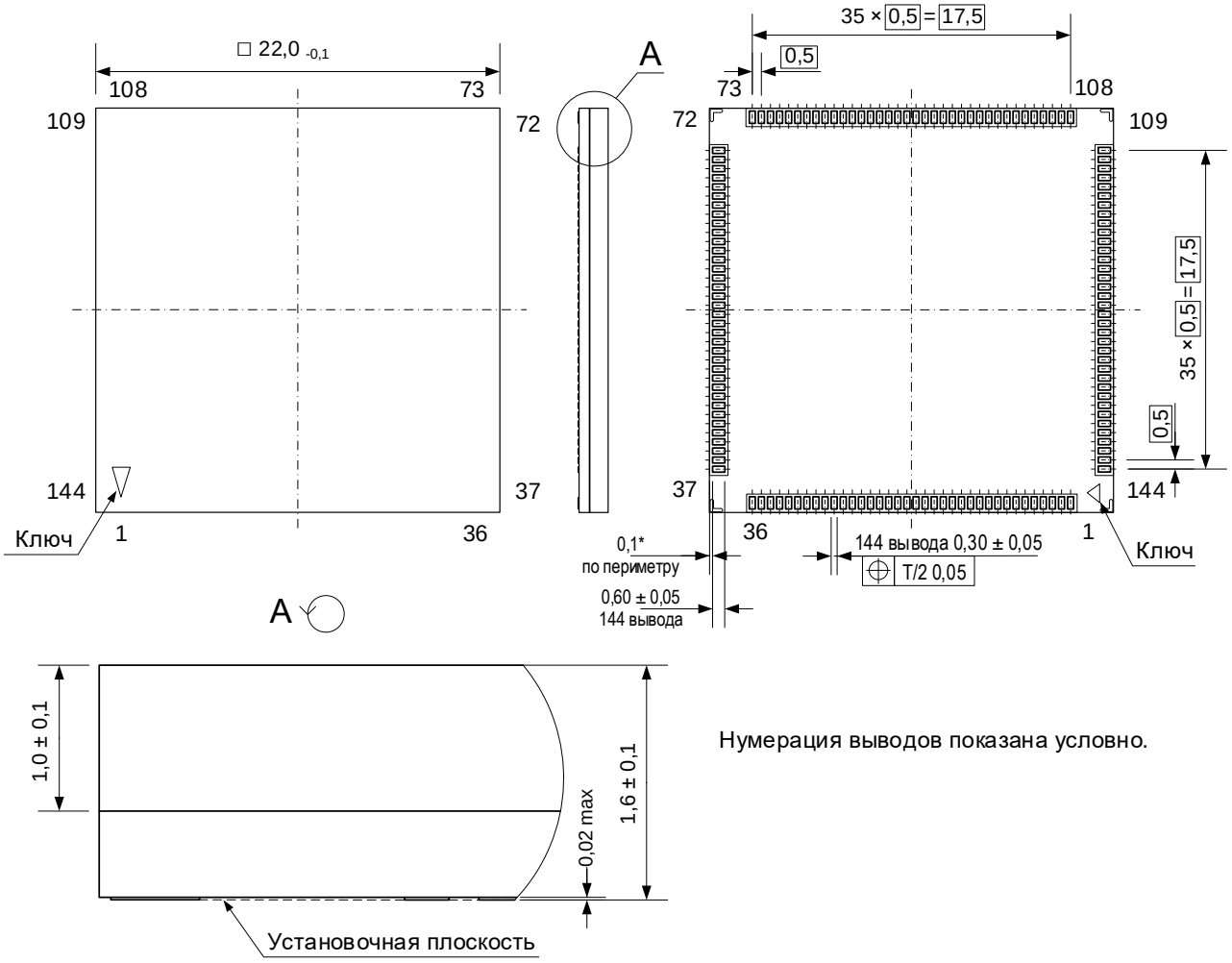



Рисунок 144 – Зависимость частоты HSI от значения HSI_TRIM

# 36 Типовые зависимости

Раздел находится в раработке.

# 37 Габаритный чертеж микросхемы




Нумерация выводов показана условно.

Рисунок 145 – Габаритный чертеж микросхемы К1986ВЕ1FI
в корпусе QFN144 22×22×1,6 (0,5)

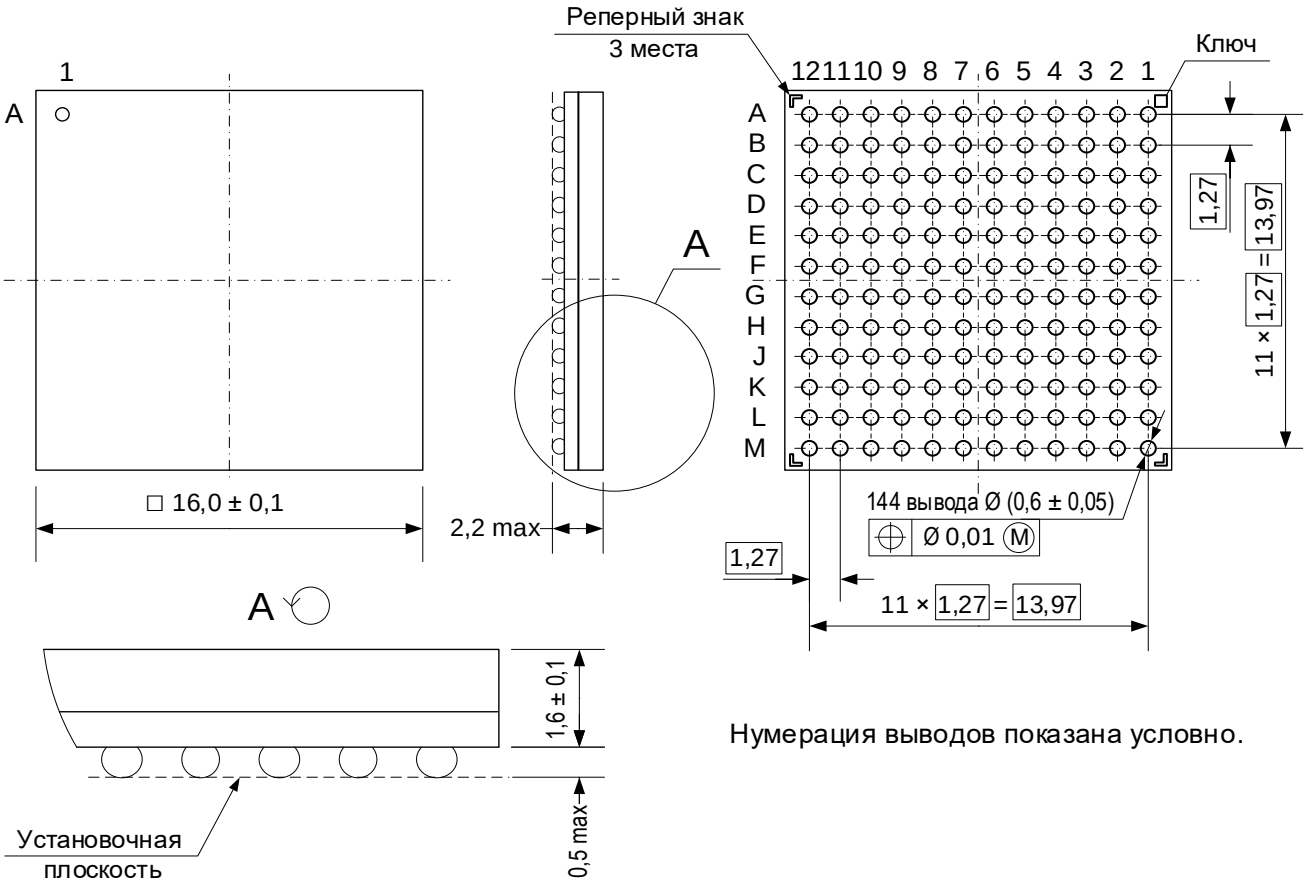



Рисунок 146 – Габаритный чертеж микросхемы К1986ВЕ1GI
в корпусе BGA144 16×16×1,6 (1,27)

# 38 Информация для заказа

| Обозначение | Маркировка | Тип корпуса | Температурный диапазон, °С |
|---|---|---|---|
| К1986ВЕ1FI | MDR1213FI | QFN144 22×22×1,6 (0,5) | от – 40 до 85 |
| К1986ВЕ1GI | MDR1213GI | BGA144 16×16×1,6 (1,27) | от – 40 до 85 |

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.431000.003ТУ;
- обозначения спецификации ТСКЯ.431296.038СП.

Пример обозначения микросхем:
Микросхема К1986ВЕ1FI – ТСКЯ.431000.003ТУ, ТСКЯ.431296.038СП.

# Лист регистрации изменений

| № п/п | Дата | Версия | Краткое содержание изменения | №№ изменяемых листов |
|---|---|---|---|---|
| 1 | 14.11.2023 | 1.0.0 | Введена впервые | |
| 2 | 24.11.2023 | 1.1.0 | Добавлено описание команды CMD_VFY | 82 |
| 3 | 29.11.2023 | 1.2.0 | Исправлены значения коэффициентов умнжения в подразделах Встроенный блок умножения системной тактовой частоты и Встроенный блок умножения тактовой частоты для контроллера USB | 139, 140 |
| 4 | 13.03.2024 | 1.3.0 | Добавлен подраздел 8.2 Тестовый режим работы.<br>Добавлено уточнение в подразделе 22.2.<br>Исправлен габаритный чертеж корпуса QFN144 22×22×1,6 (0,5) | 84<br>316<br>552 |
| 5 | 16.04.2024 | 1.3.1 | Исправлена нумерация выводов для микросхем К1986ВЕ1FI в корпусе QFN144 22×22×1,6 (0,5) | 1, 552 |
| 6 | 17.04.2024 | 1.3.2 | Дополнено описание выводов PA[2:0] на УГО и в таблицах выводов | 18, 19, 24, 24, 39, 43 |
| 7 | 08.05.2024 | 1.4.0 | Дополнен раздел Указания по применению и эксплуатации.<br>Изменения и уточнения в подразделе Схема сброса при включении и выключении основного питания.<br>Изменения и уточнения в подразделе Описание регистров блока сторожевых таймеров.<br>В таблице электрических параметров обозначение параметра $U_{BOR}$ заменено на $U_{POR}$, верхняя граница – на 2,2 В.<br>На УГО для К1986ВЕ1FI (Рисунок 2) исправлена нумерация выводов PA6, PA7.<br>Исправлен ГЧ коруса BGA144 16×16×1,6 (1,27) (Рисунок 146) | 62<br>65<br>536 – 540<br>543<br>18<br>553 |
| 8 | 24.06.2024 | 1.5.0 | Исправлены наименования выводов PE3-PE6 блока Ethernet в таблицах 2, 4.<br>Уточннен алгоритм режима авто-запрос DMA в п. 27.4.5.3.<br>Дополнен алгоритм работы с блоком сторожевого тамера IWDG в п. 32.1.1.<br>Добавлен номер ТУ (обозначение микросхем при заказе) в разделе Информация для заказа | 32, 33, 52<br>442<br>531, 538<br>554 |
| 9 | 19.11.2024 | 1.6.0 | Исправлена масса микросхем<br>Таблица 9 – дополнено описание режима MODE[2:0] = 000;<br>Раздел 12 дополнен;<br>Таблица 99 – исправлено примечание 15;<br>Таблица 102 – дополнено описание PORT RXTX[15:0];<br>Раздел 16 дополнен;<br>Таблица 126, Рисунок 28 – изменены названия и перенесены из пункта 16.3.1 в подраздел 16.1;<br>Подраздел 20.6 – дополнен; | 1<br>76<br>137<br>167<br>168<br>176<br>178<br>260 |

| № п/п | Дата | Версия | Краткое содержание изменения | №№ изменяемых листов |
|---|---|---|---|---|
| | | | Таблица 324, Таблица 332 – адрес «0x400D14FC» исправлен на «0x400D17FC»;<br>Таблица 324 – дополнено описание бита DA;<br>Таблица 332 – дополнено описание битов DA1, DA0 | 336, 339<br>336<br>339 |
| 10 | 04.02.2025 | 1.6.1 | Добавлен уровень чувствительности к влаге в основные параметры;<br>Раздел 5 – дополнен;<br>Раздел 8, подраздел 8.1 – уточнены;<br>Пункт 9.2.3 – добавлено примечание;<br>Рисунок 14, Рисунок 15 – заменены;<br>Таблица 493, таблица 500 – скорректированы | 1<br>62<br>76 – 78<br>89<br>90, 91<br>507, 514 |
| 11 | 28.03.2025 | 1.6.2 | Заменено «IWDT» на «IWDG», «WWDT» на «WWDG»;<br>Обновлен раздел Указания по применению и эксплуатации в соответствии с ТУ;<br>Таблица 8, Таблица 69 – заменено "SPIx" на "SSPx";<br>Подраздел 20.7 – дополнен | По тексту<br>62<br>74, 147<br>261 |
| 12 | 07.07.2025 | 1.7.0 | Изменен уровень MSL для микросхем К1986ВЕ1FI.<br>Обновлен раздел Указания по применению и эксплуатации | 1<br>62 |
| 13 | 16.10.2025 | 1.7.1 | Подпункт 8.1.2.6 – дополнен;<br>Рисунок 15 заменен;<br>Добавлено примечание после рисунка ;<br>Подраздел 13.1 – дополнен;<br>Рисунок 23 скорректирован;<br>Таблица 85, таблица 97 скорректированы;<br>Таблица 88 – скорректировано описание RTC CAL[7:0], RTC EN;<br>Таблица 94 – скорректировано описание RTC_PRL[19:0];<br>Таблица 96 – скорректировано описание RTC_ALRM[31:0];<br>Таблица 98 – скорректировано описание WEC, ALRF, SECF | 82<br>91<br>91<br>154<br>154<br>157, 162<br>158<br>161<br>162<br>162 |
| 14 | 10.11.2025 | 1.7.2 | Раздел 5 – добавлена информация о теплостойкости при пайке;<br>Таблица 136 – скорректированы адреса для USB_SEP0.RXFDC, USB_SEP1.RXFDC, USB_SEP2.RXFDC, USB_SEP3.RXFDC, USB_HRXFDC;<br>Таблица 474 – скорректировано описание R_EMPTY | 62<br>199<br>493 |
| 15 | 10.08.2026 | 1.7.3 | Таблица 2, Таблица 4 – добавлены примечания в конце таблиц; | 24, 43 |

| № п/п | Дата | Версия | Краткое содержание изменения | №№ изменяемых листов |
|---|---|---|---|---|
| | | | Таблица 55 – скорректировано описание PLL_CPU_RDY, PLL_USB_RDY; | 141 |
| | | | Таблица 112 – скорректировано описание PORT SHM[15:0]; | 171 |
| | | | Таблица 248 – дополнено описание бит RFLAGN_stick, RFLAGN; | 264 |
| | | | Таблица 421 – исправлено значение по сбросу для waitonreq_status; | 465 |
| | | | Таблица 430 – исправлено значение по сбросу для бит 2…0; | 469 |
| | | | Раздел 15 дополнен; | 173 |
| | | | Подпункт 18.9.3.7 – добавлено описание USB_HFN_H, USB_HFN_L; | 205 |
| | | | Подпункт 18.9.4.10 – добавлено описание USB_SFN_H, USB_SFN_L; | 218 |
| | | | Пункты 21.7.1, 21.7.3 дополнены; | 294, 295 |
| | | | Подраздел 26.4 скорректирован; | 387 |
| | | | Таблица 539 – скорректированы режимы для параметров $I_{ILL1}$, $I_{ILH1}$ | 543 |
| 16 | 16.11.2026 | 1.7.4 | Рисунок 22 скорректирован; | 138 |
| | | | Таблица 79 – скорректировано описание PHY_CLK_SEL[1:0]; | 152 |
| | | | Таблица 540 – добавлено значение $U_{IH}$ на выводе PF0 | 545 |