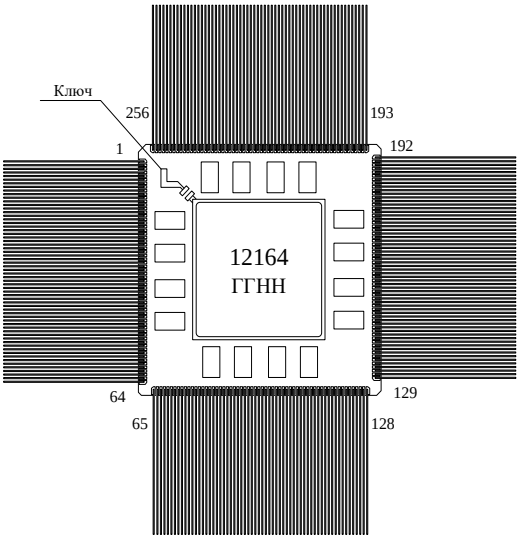
Микросхема 32-разрядного микроконтроллера с однократно программируемой памятью программ К1986ВЕ8Т



ГГ – год выпуска
НН – неделя выпуска

Основные параметры микросхемы

- Напряжение источника питания от 3,0 до 5,5 В;
- Максимальная тактовая частота до 100 МГц;
- 32-разрядная архитектура RISC ARM Cortex-M4F;
- Однократно программируемая встроенная память программ 128 Кбайт;
- Встроенная память данных 32 Кбайт;
- Тепловое сопротивление кристалл-корпус не более 1,6 °С/Вт;
- Масса микросхем с изолирующей рамкой не более 23 г;
- Температурный диапазон от минус 60 °С до плюс 125 °С.

Тип корпуса:

- 256-выводной металлокерамический корпус 4244.256-3;

Общее описание и области применения микросхемы

Микросхемы интегральные К1986ВЕ8Т (далее – микросхемы) представляют собой 32-разрядный микроконтроллер с однократно программируемой памятью программ и с измерительно-информационными каналами для чувствительных элементов (датчиков). Микросхемы предназначены для использования в системах для измерения, контроля, управления и диагностики.

Важно: спецификация действительна совместно с документом К1986ВЕ8Т Errata Notice.

Основные характеристики

Ядро:

– ядро ARM Cortex-M4F;
– производительность до 2,1 CoreMark/МГц;
– поддержка на уровне ядра операций ЦОС;
– аппаратное выполнение операций с плавающей точкой;
– блок аппаратной защиты регионов памяти;
– блок отладки с поддержкой трассы инструкций ETM и интерфейсами JTAG и SWD.

Питание:

– напряжение источника питания U_{CC} от 3,0 до 5,5 В;
– встроенные регуляторы для питания ядра;
– батарейный домен, с автоматическим переключением на питание от батарейки;
– аппаратный детектор снижения и превышения допустимого уровня питания;
– изолированные от помех питания для АЦП и ЦАП.

Тактовые частоты:

– тактовая частота ядра f_{C_CPUCLK} до 64 МГц;
– тактовая частота синхронных модулей f_{C_CLK} до 100 МГц;
– встроенные высоконадежные RC-генераторы HSI (~8 МГц) и LSI (~40 кГц);
– внешние генераторы HSE0 и HSE1 от 1 до 30 МГц в режиме резонатора;
– внешний часовой генератор LSE в батарейном домене;
– три блока PLL умножения тактовых частот f_{O_PLL} до 150 МГц;
– блоки аппаратной защиты от снижения/увеличения частоты тактирования.

Память:

– 32 Кбайт СОЗУ памяти данных с ECC (SEC-DED);
– 128 Кбайт однократно программируемой памяти программ с ECC (SEC-DED);
– контроллер внешней системной шины с последовательной/параллельной организацией с ECC (SEC-DED).

Периферия:

– контроллер EthernetMAC 10/100 Мбит/с;
– встроенный приемопередатчик EthernetPHY 10 Мбит/с;
– контроллер SpaceWire;
– встроенный приемопередатчик SpaceWirePHY до 100 Мбит/с;
– контроллер ARINC 429 (восемь приемников, четыре передатчика);
– два контроллера МКПД в режимах КШ, ОУ, Монитор;
– стандартная периферия: 1×CAN, 1×SSP, 2×UART, 1×DMA, 4×Timer;
– часы реального времени RTC;
– сторожевой таймер WDG;
– два АЦП с 16 внешними каналами с разрешающей способностью 12 бит;
– два ЦАП с разрешающей способностью 12 бит;
– до 160 выводов портов общего назначения.

Содержание

1 Структурные блок-схемы 11
2 Условно-графическое обозначение 12
3 Описание выводов 13
4 Расположение выводов микросхемы 30
5 Указания по применению и эксплуатации 35
6 Архитектура микросхемы 36
6.1 Питание 36
6.1.1 Домены питания 36
6.1.2 Сброс при включении питания 37
6.1.3 Схема защиты от перенапряжения POVR 38
6.1.4 Перекрытие рабочих диапазонов питания схемами POR и POVR 38
6.1.5 Схема переключения основного и батарейного питаний (U_{CC} и BU_{CC}) 38
6.1.6 Блок монитора основного и батарейного питаний (U_{CC} и BU_{CC}) 39
6.1.7 Управление питанием доменов DU_{CC} 39
6.1.8 Блок монитора тока потребления доменов DU_{CC} 39
6.2 Тактовые частоты 40
6.2.1 Встроенный низкоскоростной генератор LSI 40
6.2.2 Встроенный высокоскоростной генератор HSI 40
6.2.3 Внешние высокоскоростные генераторы HSE0 и HSE1 40
6.2.4 Внешний низкоскоростной часовой генератор LSE 41
6.2.5 Блок умножения тактовой частоты PLL0, PLL1 и PLL2 41
6.2.6 Монитор частоты внешних генераторов и PLL (в блоке CLKCNTR) 42
6.3 Сигналы Сброса 43
6.3.1 Сигналы внешнего сброса RSTn0 и RSTn1 43
6.3.2 Сигнал сброса по питанию UccRESET 43
6.3.3 Сигнал сброса по питанию DUccRESET 44
6.3.4 Сигнал программного запроса сброса SYSRESETnREQ 44
6.3.5 Сигналы аварийного сброса FT_RESET 44
6.3.6 Сигналы сброса сторожевого таймера IWDG_RESET 44
6.3.7 Сигнал сброса по перенапряжению POVR 44
6.3.8 Сигнал сброса отладочного интерфейса TRSTn 44
6.4 Режимы работы микросхемы 45
6.4.1 Режим OTP+JA 47
6.4.2 Режим OTP+JB 47
6.4.3 Режим EXTBUS_8_ECC+JA 48
6.4.4 Режим EXTBUS_8_ECC+JB 48
6.4.5 Режим EXTBUS_CFG+JA 48
6.4.6 Режим EXTBUS_CFG+JB 49
6.4.7 Режим SPI0+JB 49
6.4.8 Режим SPI1+JA 51

6.4.9 Режим SPI2+JB 52
6.4.10 Режим SPI3+JA 54
6.4.11 Режим UART0+JA 54
6.4.12 Режим UART0+JB 59
6.4.13 Режим TEST_MODE+JB 59
6.4.14 Режим WAIT_BOOT_JA 59
6.4.15 Режим WAIT_BOOT_JB 59
6.4.16 Статус загрузчика 59
6.5 Организация памяти 61
6.5.1 Распределение областей памяти 63
6.5.2 Распределение областей памяти для регистров периферии 65
6.5.3 Распределение областей памяти для регистров периферии ядра 66
6.6 Помехозащищенное кодирование 67
6.6.1 Описание работы кода Хэмминга HSIA70a 67
6.6.2 Кодирование ECC для ПЗУ 68
6.6.3 Кодирование ECC для памяти OTP программ 68
6.6.4 Кодирование ECC для периферии 69
6.6.5 H-матрица ECC по коду Хэмминга (72, 64) 69
6.6.6 H-матрица ECC по коду Хэмминга (7, 4) с битом четности 70
6.6.7 Функция вычисления ECC для записи в память 70
6.7 Система команд процессорного ядра 71
6.8 Интерфейс отладки 71
6.9 Тестовый режим работы 71
6.9.1 Режим тестирования ОЗУ 73
6.9.2 Режим тестирования Масочного ПЗУ (BOOTROM) 73
6.9.3 Режим тестирования OTP 74
6.9.4 Режим тестирования CAN_RAM 75
6.9.5 Режим тестирования ETH_RAM 75
6.9.6 Режим тестирования MIL_RAM 76
6.9.7 Режим тестирования ARC_RAM 76
6.9.8 Режим тестирования генераторов и PLL 77
6.9.9 Режим тестирования АЦП 78
6.9.10 Режим тестирования ЦАП 78
6.9.11 Режим тестирования приемопередатчиков SPW 79
6.9.12 Режим тестирования ETHPHY 79
6.9.13 Режим тестирования PMU (VoltageDetect) 80
6.9.14 Режим тестирования BKP 81
7 Программная модель микросхемы 82
7.1 Идентификация микросхемы 82
7.1.1 Описание полей идентификации 82
7.2 Контроллер тактовых частот (CLKCNTR) 83
7.2.1 Формирование и управление тактовыми сигналами 83

7.2.2 Схема контроля тактовых частот 85
7.2.3 Описание регистров 90
7.3 Контроллер батарейного домена (BKPCNTR) 129
7.3.1 Часы реального времени 130
7.3.2 Описание регистров 131
7.4 Контроллер детектора напряжения питания (PWRCNTR) 147
7.4.1 Описание регистров 147
7.5 Контроллер обработки событий отказов, сбоев и ошибок (FTCNTR) 154
7.5.1 Описание регистров 155
7.6 Контроллер сторожевых таймеров (WDGCNTR) 189
7.6.1 Описание регистров 190
7.7 Контроллер внешней шины (EXTBUSCNTR) 193
7.7.1 Организация доступа к внешней шине EXTBUS 193
7.7.2 Описание регистров 209
7.8 Контроллеры кэш-памяти (ICACHECNTR, DCACHECNTR и RAMDCACHECNTR) 216
7.8.1 Описание регистров 217
7.9 Контроллер OTP-памяти программ (OTPCNTR) 223
7.9.1 Методика программирования OTP 223
7.9.2 Параметры контроллера OTP 232
7.9.3 Описание регистров 233
7.10 Контроллер ПЗУ (ROMCNTR) 239
7.10.1 Описание регистров 240
7.11 Контроллер ОЗУ (RAMCNTR) 243
7.11.1 Описание регистров 244
7.12 Контроллер DMA (DMACNTR) 249
7.12.1 Основные характеристики контроллера DMA 249
7.12.2 Термины и определения 249
7.12.3 Функциональное описание 251
7.12.4 Управление DMA 253
7.12.5 Структура управляющих данных канала 275
7.12.6 Описание регистров контроллера DMA 285
7.13 Контроллер портов ввода-вывода (PORTx) 305
7.13.1 Описание регистров 306
7.14 Контроллер Ethernet (EthernetMAC) 329
7.14.1 Передача пакета 330
7.14.2 Принцип работы передатчика 331
7.14.3 Прием пакета 331
7.14.4 Принцип работы приемника 332
7.14.5 Линейный режим работы буферов 333
7.14.6 Автоматический режим работы буферов 333
7.14.7 Режим FIFO работы буферов 333

7.14.8 События приемника и передатчика 333
7.14.9 Прерывания 334
7.14.10 Режим детерминированного времени доставки 334
7.14.11 Режим КЗ 334
7.14.12 Блок PHY 335
7.14.13 Описание регистров 335
7.15 Контроллер SpaceWire (SpaceWire) 351
7.15.1 Кодирование данных 351
7.15.2 Отказоустойчивая работа приемопередатчика SpaceWire PHY 351
7.15.3 Скорость интерфейса 352
7.15.4 Представление символов 352
7.15.5 Диаграмма состояний 354
7.15.6 Инициализация 357
7.15.7 Run (Рабочий режим) 357
7.15.8 Передача сообщений 359
7.15.9 Прием сообщений 359
7.15.10 Передача и прием маркера времени 359
7.15.11 Описание регистров 360
7.16 Контроллер таймеров общего назначения (TIMER32x) 373
7.16.1 Основные характеристики 373
7.16.2 Базовый блок таймера 374
7.16.3 Источники событий для счета 379
7.16.4 Режим захвата 385
7.16.5 Режим ШИМ 386
7.16.6 Блок цифрового фильтра 390
7.16.7 Флаги состояний, прерывания и запросы DMA 392
7.16.8 Примеры 393
7.16.9 Описание регистров блока таймера 396
7.17 Контроллер CAN 412
7.17.1 Режимы работы 412
7.17.2 Типы пакетов сообщений 414
7.17.3 Структура пакета данных (DataFrame) 415
7.17.4 Структура пакета удаленного запроса данных (Remoteframe) 418
7.17.5 Арбитраж на шине 418
7.17.6 Инициализация 420
7.17.7 Передача сообщений 420
7.17.8 Передача сообщений по RemoteTransmitRequest (RTR) 421
7.17.9 Прием сообщений 421
7.17.10 Автоматическая фильтрация принимаемых сообщений 421
7.17.11 Задание скорости передачи и момента сэмплирования 422
7.17.12 Перезапись принятых сообщений 423
7.17.13 Синхронизация 424

7.17.14 Обработка ошибок 424
7.17.15 Прерывания 428
7.17.16 Описание регистров 429
7.18 Контроллер SSP 443
7.18.1 Программируемые параметры 443
7.18.2 Общий обзор модуля SSP 445
7.18.3 Интерфейс прямого доступа к памяти 461
7.18.4 Прерывания 463
7.18.5 Программное управление модулем 464
7.18.6 Описание регистров 465
7.19 Контроллер UART (UARTx) 471
7.19.1 Основные сведения 471
7.19.2 Основные характеристики модуля UART 471
7.19.3 Программируемые параметры 472
7.19.4 Описание функционирования блока UART 474
7.19.5 Интерфейс прямого доступа к памяти 475
7.19.6 Блок и регистры синхронизации 476
7.19.7 Описание работы UART 476
7.19.8 Интерфейс прямого доступа к памяти 480
7.19.9 Прерывания 482
7.19.10 Программное управление модулем 484
7.19.11 Описание регистров 485
7.20 Контроллер интерфейса ARINC 499
7.20.1 Формат слова 500
7.20.2 Структурная схема канала приема 502
7.20.3 Структурная схема канала передачи 503
7.20.4 Описание регистров приемника 505
7.20.5 Описание регистров передатчика 516
7.21 Контроллер интерфейса МКПД (MILx) 524
7.21.1 Режимы работы 525
7.21.2 Форматы сообщений 526
7.21.3 Формат слов 528
7.21.4 Структурная схема в режиме КШ 530
7.21.5 Структурная схема в режиме ОУ 531
7.21.6 Структурная схема в режиме М 531
7.21.7 Инициализация 531
7.21.8 Прием и передача в режиме ОУ 532
7.21.9 Прием и передача в режиме КШ 534
7.21.10 Прерывания 535
7.21.11 Описание регистров 535
7.22 Контроллер АЦП (ADCx) 547
7.22.1 Преобразуемые сигналы 549

7.22.2 Источники опорного напряжения 549
7.22.3 Синхронизация АЦП 550
7.22.4 Запуск преобразования 550
7.22.5 Непрерывное преобразование 551
7.22.6 Последовательный опрос каналов АЦП 552
7.22.7 Получение результатов преобразования 552
7.22.8 Обработка результатов с помощью DMA 552
7.22.9 Преобразование с контролем границ 552
7.22.10 Диаграмма работы АЦП 553
7.22.11 Схема входного каскада АЦП 553
7.22.12 Описание регистров 555
7.23 Контроллер ЦАП (DACx) 572
7.23.1 Источники опорного напряжения 572
7.23.2 Синхронизация ЦАП 572
7.23.3 Очередь данных 574
7.23.4 Описание регистров 574
7.24 Векторы прерываний и регистры управления ядра 581
7.24.1 Распределение векторов прерываний 581
7.24.2 Описание регистров управления ядра (SCS) 584
7.25 Контроллер блока защиты памяти (MPU) 588
7.26 Контроллер блока вычисления с плавающей запятой (FPU) 588
7.27 Контроллер порта трассировки (TPIU) 588
7.28 Контроллер блока расширенной трассировки (ETM) 588
8 Системный таймер SysTick 589
8.1 Описание регистров системного таймера SysTick 589
8.1.1 SysTick->CTRL 589
8.1.2 SysTick->LOAD 590
8.1.3 SysTick->VAL 591
8.1.4 SysTick->CAL 591
8.2 Советы и особенности при применении системного таймера 591
9 Прерывания и исключения 592
9.1 Типы исключений 592
9.1.1 RESET 592
9.1.2 NON MASKABLE INTERRUPT (NMI) 592
9.1.3 Hard Fault 592
9.1.4 Memory Management fault 592
9.1.5 Bus Fault 593
9.1.6 Usage Fault 593
9.1.7 SVCall 593
9.1.8 PendSV 593
9.1.9 SysTick 593
9.2 Прерывания (IRQ) 593

9.3 Обработчики исключений 594
9.4 Таблица векторов 595
9.5 Приоритеты исключений 595
9.5.1 Группировка приоритетов прерываний 596
9.6 Вход в обработчик и выход из обработчика 596
9.6.1 Приоритетное прерывание 596
9.6.2 Возврат 597
9.6.3 Передача управления без восстановления контекста (tail-chaining) 597
9.6.4 Запоздавшее исключение (late-arriving exception) 597
9.6.5 Вход в процедуру обработки исключения 597
9.6.6 Возврат из обработчика исключения 598
9.7 Обработка отказов 599
9.7.1 Типы отказов 599
9.7.2 Эскалация отказов и тяжелые отказы 601
9.7.3 Регистры состояния и адреса отказа 601
9.7.4 Блокировка 602
9.8 Управление электропитанием 602
9.8.1 Переход в режим пониженного энергопотребления 602
9.8.2 Ожидание прерывания 602
9.8.3 Ожидание события 602
9.8.4 Переход в режим ожидания по выходу из обработчика исключения (режим Sleep) 603
9.8.5 Выход из состояния ожидания 603
9.8.6 Рекомендации по программированию режима энергопотребления 604
10 Контроллер прерываний NVIC 605
10.1 Логика работы прерываний контроллера NVIC 606
10.2 Упрощенный доступ к регистрам контроллера прерываний 609
10.2.1 NVIC->ISER[x] 610
10.2.2 NVIC->ICER[x] 610
10.2.3 NVIC->ISPR[x] 611
10.2.4 NVIC->ICPR[x] 611
10.2.5 NVIC->IABR[x] 612
10.2.6 NVIC->IP[x] 612
10.2.7 NVIC->STIR 613
10.3 Прерывания, срабатывающие по уровню сигнала 613
10.4 Аппаратное и программное управление прерываниями 614
10.5 Рекомендации по работе с контроллером прерываний 614
11 Блок управления системой 616
11.1 Упрощенный доступ к регистрам блока управления системой 617
11.1.1 InterrupType->ACTLR 617
11.1.2 SCB->CPUID 618
11.1.3 SCB->ICSR 618

11.1.4 SCB->VTOR 620
11.1.5 SCB->AIRCR 621
11.1.6 SCB->SCR 622
11.1.7 SCB->CCR 623
11.1.8 SCB->SHP[x] 624
11.1.9 SCB->SHCSR 626
11.1.10 SCB->CFSR 627
11.1.11 SCB->HFSR 631
11.1.12 SCB->MMFAR 632
11.1.13 SCB->BFAR 633
11.1.14 Рекомендации по программированию блока управления системой. 633
12 Типовая схема включения 634
13 Основные зависимости 635
14 Электрические параметры микросхемы 636
15 Предельно-допустимые характеристики микросхемы 639
16 Справочные данные 642
17 Габаритный чертеж микросхемы 645
18 Инструкция по распаковыванию микросхем с формованными выводами 647
19 Информация для заказа 649

1 Структурные блок-схемы

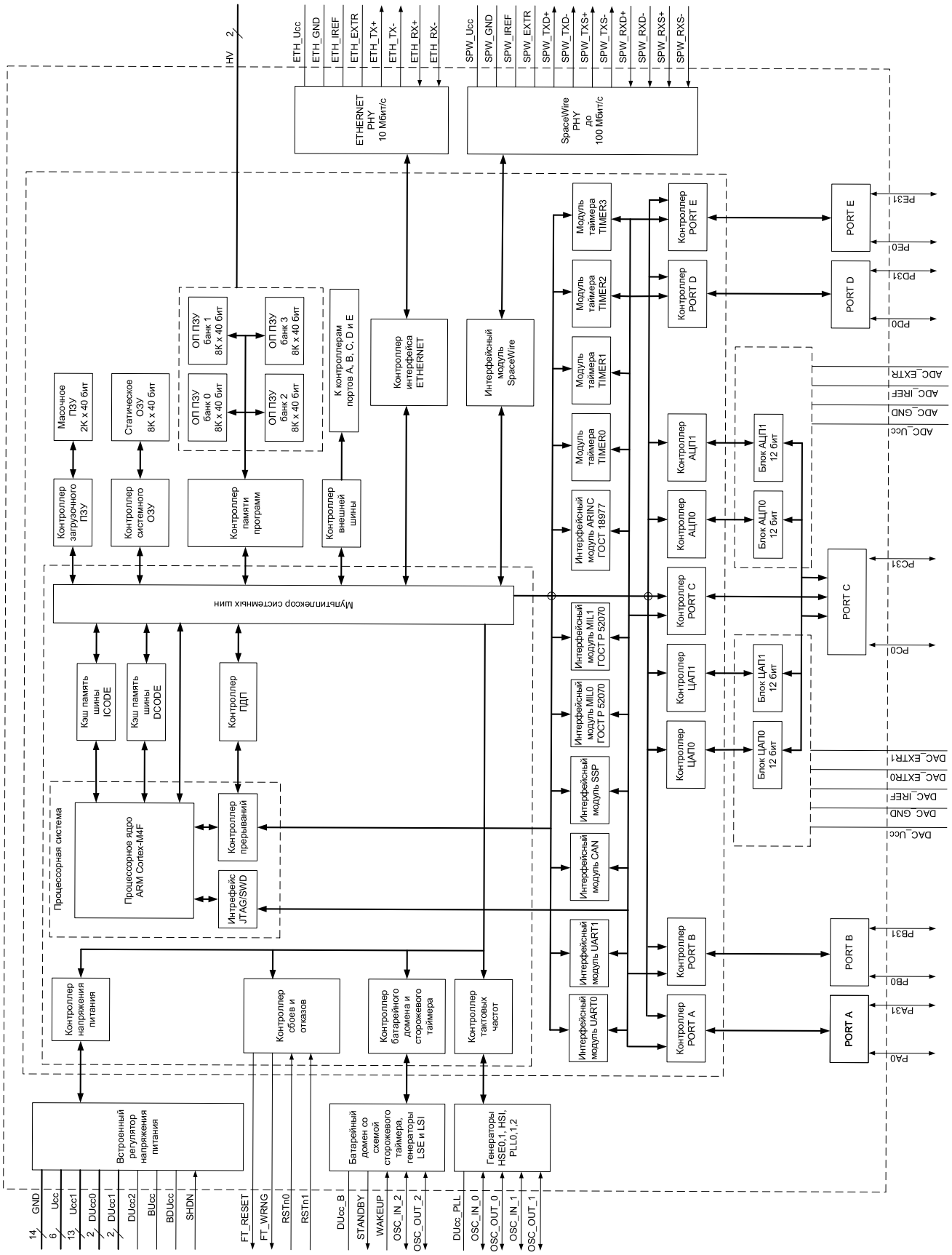


Рисунок 1 – Структурная блок-схема

2 Условно-графическое обозначение

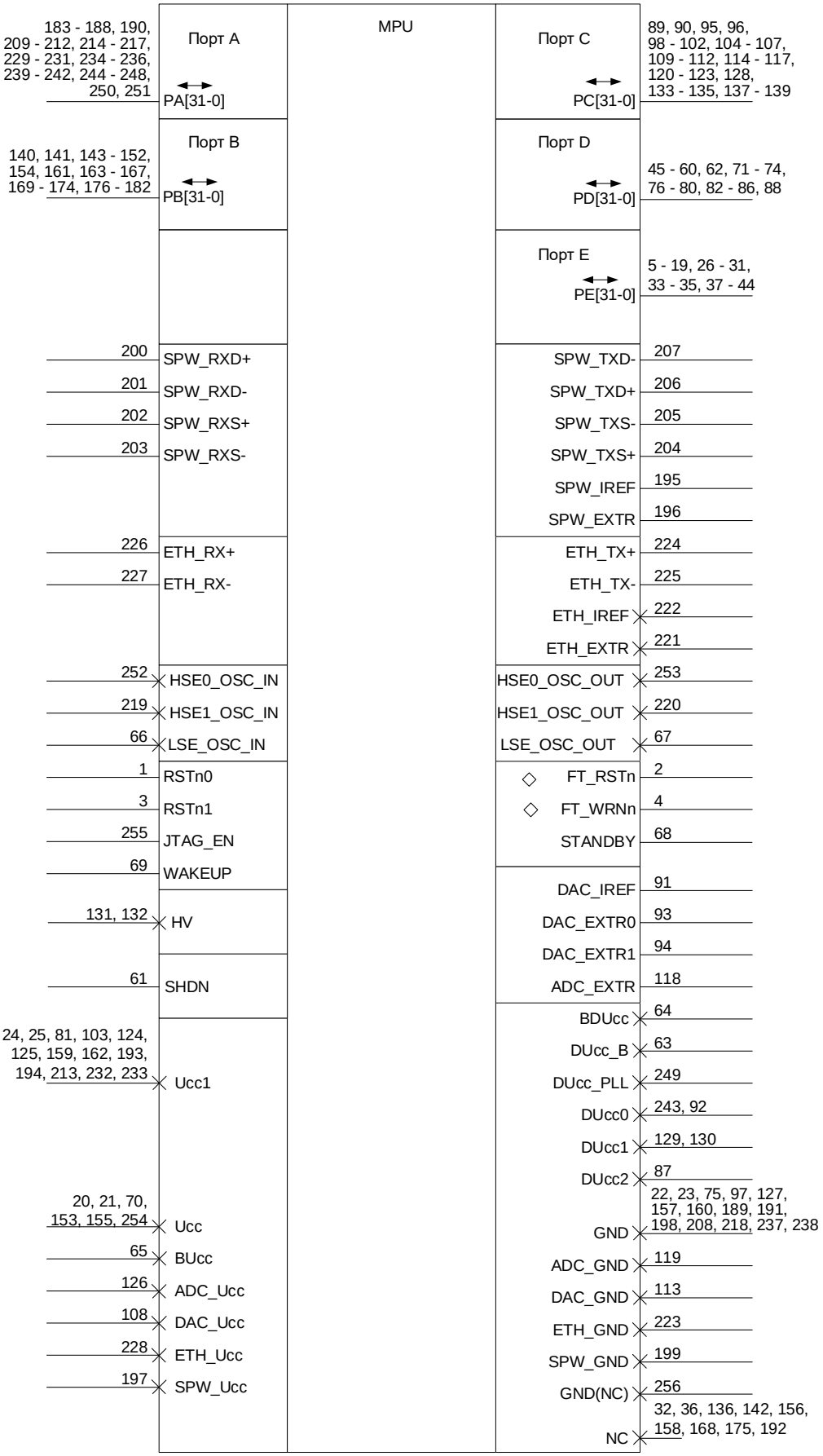


Рисунок 2 – Условно-графическое обозначение

3 Описание выводов

Таблица 1 – Описание выводов

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода
1	RSTn0	I 5V	Вход нулевого сигнала сброса: 0 – запрос сброса; 1 – рабочий режим
2	FT_RSTn	O 5V	Выход с открытым стоком сигнала запроса сброса от блока: 0 – сигнал запроса сброса
3	RSTn1	I 5V	Вход первого сигнала сброса: 0 – запрос сброса; 1 – рабочий режим. Сброс микросхемы выполняется при условии запроса сброса по обоим выводам RSTn0 и RSTn1
4	FT_WRNn	O 5V	Выход с открытым стоком сигнала предупреждения о сбое от блока контроллера сбоев: 0 – предупреждение о сбое
5	PE[31]	IO 5V	Вход-выход 31 порта E
6	PE[30]	IO 5V	Вход-выход 30 порта E
7	PE[29]	IO 5V	Вход-выход 29 порта E
8	PE[28]	IO 5V	Вход-выход 28 порта E
9	PE[27]	IO 5V	Вход-выход 27 порта E
10	PE[26]	IO 5V	Вход-выход 26 порта E
11	PE[25]	IO 5V	Вход-выход 25 порта E
12	PE[24]	IO 5V	Вход-выход 24 порта E
13	PE[23]	IO 5V	Вход-выход 23 порта E
14	PE[22]	IO 5V	Вход-выход 22 порта E
15	PE[21]	IO 5V	Вход-выход 21 порта E
16	PE[20]	IO 5V	Вход-выход 20 порта E
17	PE[19]	IO 5V	Вход-выход 19 порта E
18	PE[18]	IO 5V	Вход-выход 18 порта E
19	PE[17]	IO 5V	Вход-выход 17 порта E
20 21	U_{CC}	PWR	Питание площадок ввода-вывода
22 23	GND	PWR	Общий
24 25	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
26	PE[16]	IO 5V	Вход-выход 16 порта E
27	PE[15]	IO 5V	Вход-выход 15 порта E
28	PE[14]	IO 5V	Вход-выход 14 порта E
29	PE[13]	IO 5V	Вход-выход 13 порта E

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода
30	PE[12]	IO 5V	Вход-выход 12 порта E
31	PE[11]	IO 5V	Вход-выход 11 порта E
32	NC	–	Не подключен
33	PE[10]	IO 5V	Вход-выход 10 порта E
34	PE[9]	IO 5V	Вход-выход 9 порта E
35	PE[8]	IO 5V	Вход-выход 8 порта E
36	NC	–	Не подключен
37	PE[7]	IO 5V	Вход-выход 7 порта E
38	PE[6]	IO 5V	Вход-выход 6 порта E
39	PE[5]	IO 5V	Вход-выход 5 порта E
40	PE[4]	IO 5V	Вход-выход 4 порта E
41	PE[3]	IO 5V	Вход-выход 3 порта E
42	PE[2]	IO 5V	Вход-выход 2 порта E
43	PE[1]	IO 5V	Вход-выход 1 порта E
44	PE[0]	IO 5V	Вход-выход 0 порта E
45	PD[31]	IO 5V	Вход-выход 31 порта D
46	PD[30]	IO 5V	Вход-выход 30 порта D
47	PD[29]	IO 5V	Вход-выход 29 порта D
48	PD[28]	IO 5V	Вход-выход 28 порта D
49	PD[27]	IO 5V	Вход-выход 27 порта D
50	PD[26]	IO 5V	Вход-выход 26 порта D
51	PD[25]	IO 5V	Вход-выход 25 порта D
52	PD[24]	IO 5V	Вход-выход 24 порта D
53	PD[23]	IO 5V	Вход-выход 23 порта D
54	PD[22]	IO 5V	Вход-выход 22 порта D
55	PD[21]	IO 5V	Вход-выход 21 порта D
56	PD[20]	IO 5V	Вход-выход 20 порта D
57	PD[19]	IO 5V	Вход-выход 19 порта D
58	PD[18]	IO 5V	Вход-выход 18 порта D
59	PD[17]	IO 5V	Вход-выход 17 порта D
60	PD[16]	IO 5V	Вход-выход 16 порта D
61	SHDN	I	Вход выключения встроенных регуляторов напряжения: 0 – регуляторы включены; 1 – регуляторы выключены
62	PD[15]	IO 5V	Вход-выход 15 порта D
63	DU_{CC}_B	PWR	Питание 1,8 В цифровой части батарейного домена при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода
64	BDU_{CC}	PWR	Вывод подключения конденсатора к питанию LDO регулятора батарейного домена
65	BU_{CC}	PWR	Батарейное питание
66	LSE_OSC_IN	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору LSE
67	LSE_OSC_OUT	A	Вывод для подключения кварцевого резонатора к генератору LSE
68	STANDBY	O	Выход состояния встроенного регулятора: 0 – регулятор включен; 1 – регулятор программно выключен
69	WAKEUP	I 5V	Вход сигнала запроса включения регулятора: 1 – нет запроса включения; 0 – есть запрос включения встроенного регулятора
70	U_{CC}	PWR	Питание площадок ввода-вывода
71	PD[14]	IO 5V	Вход-выход 14 порта D
72	PD[13]	IO 5V	Вход-выход 13 порта D
73	PD[12]	IO 5V	Вход-выход 12 порта D
74	PD[11]	IO 5V	Вход-выход 11 порта D
75	GND	PWR	Общий
76	PD[10]	IO 5V	Вход-выход 10 порта D
77	PD[9]	IO 5V	Вход-выход 9 порта D
78	PD[8]	IO 5V	Вход-выход 8 порта D
79	PD[7]	IO 5V	Вход-выход 7 порта D
80	PD[6]	IO 5V	Вход-выход 6 порта D
81	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
82	PD[5]	IO 5V	Вход-выход 5 порта D
83	PD[4]	IO 5V	Вход-выход 4 порта D
84	PD[3]	IO 5V	Вход-выход 3 порта D
85	PD[2]	IO 5V	Вход-выход 2 порта D
86	PD[1]	IO 5V	Вход-выход 1 порта D
87	$DU_{CC}2$	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
88	PD[0]	IO 5V	Вход-выход 0 порта D
89	PC[31]	IO 5V	Вход-выход 31 порта C
90	PC[30]	IO 5V	Вход-выход 30 порта C
91	DAC_IREF	A	Контрольный вывод проверки источника опорного тока

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода
92	$DU_{CC}0$	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
93	DAC_EXTR0	A	Выводы для подключения внешних токозадающих резисторов для формирования опорных токов R = 120 кОм ± 5 %
94	DAC_EXTR1	A	
95	PC[29]	AIO	Вход-выход 29 порта C. Совмещен с сигналом DAC1_REF+
96	PC[28]	AIO	Вход-выход 28 порта C. Совмещен с сигналом DAC1_OUT
97	GND	PWR	Общий
98	PC[27]	AIO	Вход-выход 27 порта C. Совмещен с сигналом DAC1_REF-
99	PC[26]	AIO	Вход-выход 26 порта C. Совмещен с сигналом DAC0_REF-
100	PC[25]	AIO	Вход-выход 25 порта C. Совмещен с сигналом DAC0_OUT
101	PC[24]	AIO	Вход-выход 24 порта C. Совмещен с сигналом DAC0_REF+
102	PC[23]	AIO	Вход-выход 23 порта C. Совмещен с сигналом ADC17_REF1-
103	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
104	PC[22]	AIO	Вход-выход 22 порта C. Совмещен с сигналом ADC16_REF1+
105	PC[21]	AIO	Вход-выход 21 порта C. Совмещен с сигналом ADC15
106	PC[20]	AIO	Вход-выход 20 порта C. Совмещен с сигналом ADC14
107	PC[19]	AIO	Вход-выход 19 порта C. Совмещен с сигналом ADC13
108	DAC_U_{CC}	PWR	Питание аналоговых блоков ЦАП
109	PC[18]	AIO	Вход-выход 18 порта C. Совмещен с сигналом ADC12
110	PC[17]	AIO	Вход-выход 17 порта C. Совмещен с сигналом ADC11
111	PC[16]	AIO	Вход-выход 16 порта C. Совмещен с сигналом ADC10
112	PC[15]	AIO	Вход-выход 15 порта C. Совмещен с сигналом ADC9
113	DAC_GND	PWR	Общий аналоговых блоков ЦАП
114	PC[14]	AIO	Вход-выход 14 порта C. Совмещен с сигналом ADC8
115	PC[13]	AIO	Вход-выход 13 порта C. Совмещен с сигналом ADC7
116	PC[12]	AIO	Вход-выход 12 порта C. Совмещен с сигналом ADC6
117	PC[11]	AIO	Вход-выход 11 порта C. Совмещен с сигналом ADC5
118	ADC_EXTR	A	Вывод для подключения внешнего токозадающего резистора для формирования опорного тока R = 120 кОм ± 5 %
119	ADC_GND	PWR	Общий аналоговых блоков АЦП
120	PC[10]	AIO	Вход-выход 10 порта C. Совмещен с сигналом ADC4
121	PC[9]	AIO	Вход-выход 9 порта C. Совмещен с сигналом ADC3
122	PC[8]	AIO	Вход-выход 8 порта C. Совмещен с сигналом ADC2
123	PC[7]	AIO	Вход-выход 7 порта C. Совмещен с сигналом ADC1_REF0-

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода
124	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
125			
126	ADC_U_{CC}	PWR	Питание аналоговых блоков АЦП
127	GND	PWR	Общий
128	PC[6]	AIO	Вход-выход 6 порта С. Совмещен с сигналом ADC0_REF0+
129	$DU_{CC}1$	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
130			
131	HV	PWR	Высоковольтный вход напряжения программирования однократно программируемой памяти программ. В рабочем режиме подключить к U_{CC}
132			
133	PC[5]	IO 5V	Вход-выход 5 порта С
134	PC[4]	IO 5V	Вход-выход 4 порта С
135	PC[3]	IO 5V	Вход-выход 3 порта С
136	NC	–	Не подключен
137	PC[2]	IO 5V	Вход-выход 2 порта С
138	PC[1]	IO 5V	Вход-выход 1 порта С
139	PC[0]	IO 5V	Вход-выход 0 порта С
140	PB[31]	IO 5V	Вход-выход 31 порта В
141	PB[30]	IO 5V	Вход-выход 30 порта В
142	NC	–	Не подключен
143	PB[29]	IO 5V	Вход-выход 29 порта В
144	PB[28]	IO 5V	Вход-выход 28 порта В
145	PB[27]	IO 5V	Вход-выход 27 порта В
146	PB[26]	IO 5V	Вход-выход 26 порта В
147	PB[25]	IO 5V	Вход-выход 25 порта В
148	PB[24]	IO 5V	Вход-выход 24 порта В
149	PB[23]	IO 5V	Вход-выход 23 порта В
150	PB[22]	IO 5V	Вход-выход 22 порта В
151	PB[21]	IO 5V	Вход-выход 21 порта В
152	PB[20]	IO 5V	Вход-выход 20 порта В
153	U_{CC}	PWR	Питание площадок ввода-вывода
154	PB[19]	IO 5V	Вход-выход 19 порта В
155	U_{CC}	PWR	Питание площадок ввода-вывода
156	NC	–	Не подключен
157	GND	PWR	Общий
158	NC	–	Не подключен
159	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
160	GND	PWR	Общий

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода
161	PB[18]	IO 5V	Вход-выход 18 порта B
162	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
163	PB[17]	IO 5V	Вход-выход 17 порта B
164	PB[16]	IO 5V	Вход-выход 16 порта B
165	PB[15]	IO 5V	Вход-выход 15 порта B
166	PB[14]	IO 5V	Вход-выход 14 порта B
167	PB[13]	IO 5V	Вход-выход 13 порта B
168	NC	–	Не подключен
169	PB[12]	IO 5V	Вход-выход 12 порта B
170	PB[11]	IO 5V	Вход-выход 11 порта B
171	PB[10]	IO 5V	Вход-выход 10 порта B
172	PB[9]	IO 5V	Вход-выход 9 порта B
173	PB[8]	IO 5V	Вход-выход 8 порта B. При выборе JTAG_B совмещен с JTAGB_SWVTDO
174	PB[7]	IO 5V	Вход-выход 7 порта B. При выборе JTAG_B совмещен с JTAGB_TDI
175	NC	–	Не подключен
176	PB[6]	IO 5V	Вход-выход 6 порта B. При выборе JTAG_B совмещен с JTAGB_SWDIOTMS
177	PB[5]	IO 5V	Вход-выход 5 порта B. При выборе JTAG_B совмещен с JTAGB_SWCLKTCK
178	PB[4]	IO 5V	Вход-выход 4 порта B. При выборе JTAG_B совмещен с JTAGB_nTRST
179	PB[3]	IO 5V	Вход-выход 3 порта B
180	PB[2]	IO 5V	Вход-выход 2 порта B
181	PB[1]	IO 5V	Вход-выход 1 порта B
182	PB[0]	IO 5V	Вход-выход 0 порта B
183	PA[31]	IO 5V	Вход-выход 31 порта A
184	PA[30]	IO 5V	Вход-выход 30 порта A
185	PA[29]	IO 5V	Вход-выход 29 порта A
186	PA[28]	IO 5V	Вход-выход 28 порта A
187	PA[27]	IO 5V	Вход-выход 27 порта A
188	PA[26]	IO 5V	Вход-выход 26 порта A
189	GND	PWR	Общий
190	PA[25]	IO 5V	Вход-выход 25 порта A
191	GND	PWR	Общий
192	NC	–	Не подключен
193	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
194			
195	SPW_IREF	A	Контрольный вывод проверки источника опорного тока

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода
196	SPW_EXTR	A	Вывод для подключения внешнего токозадающего резистора для формирования опорного тока R = 120 кОм ± 5 %
197	SPW_U_{CC}	PWR	Питание приемопередатчика SpaceWire PHY
198	GND	PWR	Общий
199	SPW_GND	PWR	Общий приемопередатчика SpaceWirePHY
200	SPW_RXD+	AI	Сигналы приемопередатчика интерфейса SpaceWire
201	SPW_RXD-	AI	
202	SPW_RXS+	AI	
203	SPW_RXS-	AI	
204	SPW_TXS+	AO	
205	SPW_TXS-	AO	
206	SPW_TXD+	AO	
207	SPW_TXD-	AO	
208	GND	PWR	Общий
209	PA[24]	IO 5V	Вход-выход 24 порта А
210	PA[23]	IO 5V	Вход-выход 23 порта А
211	PA[22]	IO 5V	Вход-выход 22 порта А
212	PA[21]	IO 5V	Вход-выход 21 порта А
213	U_{CC}1	PWR	Питание встроенного регулятора LDO для формирования напряжений питания DU_{CC}0, DU_{CC}1, DU_{CC}2, DU_{CC}_PLL
214	PA[20]	IO 5V	Вход-выход 20 порта А
215	PA[19]	IO 5V	Вход-выход 19 порта А
216	PA[18]	IO 5V	Вход-выход 18 порта А
217	PA[17]	IO 5V	Вход-выход 17 порта А
218	GND	PWR	Общий
219	HSE1_OSC_IN	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору HSE1
220	HSE1_OSC_OUT	A	Вывод для подключения кварцевого резонатора к генератору HSE1
221	ETH_EXTR	A	Вывод для подключения внешнего токозадающего резистора для формирования опорного тока R = 120 кОм ± 5 %
222	ETH_IREF	A	Контрольный вывод проверки источника опорного тока
223	ETH_GND	PWR	Общий приемопередатчика EthernetPHY
224	ETH_TX+	AO	Сигналы приемопередатчика интерфейса Ethernet
225	ETH_TX-	AO	
226	ETH_RX+	AI	
227	ETH_RX-	AI	
228	ETH_U_{CC}	PWR	Питание приемопередатчика EthernetPHY
229	PA[16]	IO 5V	Вход-выход 16 порта А

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода
230	PA[15]	IO 5V	Вход-выход 15 порта А
231	PA[14]	IO 5V	Вход-выход 14 порта А
232	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
233			
234	PA[13]	IO 5V	Вход-выход 13 порта А
235	PA[12]	IO 5V	Вход-выход 12 порта А
236	PA[11]	IO 5V	Вход-выход 11 порта А
237	GND	PWR	Общий
238			
239	PA[10]	IO 5V	Вход-выход 10 порта А
240	PA[9]	IO 5V	Вход-выход 9 порта А
241	PA[8]	IO 5V	Вход-выход 8 порта А
242	PA[7]	IO 5V	Вход-выход 7 порта А
243	$DU_{CC}0$	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
244	PA[6]	IO 5V	Вход-выход 6 порта А
245	PA[5]	IO 5V	Вход-выход 5 порта А
246	PA[4]	IO 5V	Вход-выход 4 порта А. Совмещен с JTAGA_SWVTDO
247	PA[3]	IO 5V	Вход-выход 3 порта А. Совмещен с JTAGA_TDI
248	PA[2]	IO 5V	Вход-выход 2 порта А. Совмещен с JTAGA_SWDIOTMS
249	DU_{CC}_PLL	PWR	Питание 1,8 В домена PLL при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
250	PA[1]	IO 5V	Вход-выход 1 порта А. Совмещен с JTAGA_SWCLKTCK
251	PA[0]	IO 5V	Вход-выход 0 порта А. Совмещен с JTAGA_nTRST
252	HSE0_OSC_IN	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору HSE0
253	HSE0_OSC_OUT	A	Вывод для подключения кварцевого резонатора к генератору HSE0
254	U_{CC}	PWR	Питание площадок ввода-вывода
255	JTAG_EN	I	Вход выбора тестового режима 0 – рабочий режим 1 – тестовый режим
256	GND(NC)	PWR	Общий подложки кристалла

Номер вывода корпуса	Обозначение вывода	Тип вывода	Назначение и функции вывода

Примечания

1 Пользовательский порт A ввода-вывода совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL и Ethernet MAC, Timer, TRACE.

2 Пользовательский порт B ввода-вывода совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL и SpaceWire MAC, Ethernet MAC, Timer.

3 Пользовательский порт C ввода-вывода совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer.

4 Пользовательский порт D ввода-вывода совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer.

5 Пользовательский порт E ввода-вывода совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer.

6 Типы выводов:

PWR – вывод питания;
I – цифровой вход;
O – цифровой выход;
IO – цифровой вход-выход;
A – аналоговый вывод;
AI – аналоговый вход;
AO – аналоговый выход;
AIO – цифровой вход-выход совмещенный с аналоговых выводом;
5V – толерантный к 5 В цифровой вход

Таблица 2 – Описание выводов по блокам

Номер вывода	Обозначение вывода	Тип вывода	Назначение и функции вывода
Порт А			
251	PA[0]	IO 5V	Пользовательский порт А ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL и Ethernet MAC, Timer, TRACE
250	PA[1]	IO 5V	
248	PA[2]	IO 5V	
247	PA[3]	IO 5V	
246	PA[4]	IO 5V	При выборе JTAG_A
245	PA[5]	IO 5V	Вывод PA[0] совмещен с JTAGA_nTRST
244	PA[6]	IO 5V	Вывод PA[1] совмещен с JTAGA_SWCLKTCK
242	PA[7]	IO 5V	Вывод PA[2] совмещен с JTAGA_SWDIOTMS
241	PA[8]	IO 5V	Вывод PA[3] совмещен с JTAGA_TDI
240	PA[9]	IO 5V	Вывод PA[4] совмещен с JTAGA_SWVTDO
239	PA[10]	IO 5V	
236	PA[11]	IO 5V	
235	PA[12]	IO 5V	
234	PA[13]	IO 5V	
231	PA[14]	IO 5V	
230	PA[15]	IO 5V	
229	PA[16]	IO 5V	
217	PA[17]	IO 5V	
216	PA[18]	IO 5V	
215	PA[19]	IO 5V	
214	PA[20]	IO 5V	
212	PA[21]	IO 5V	
211	PA[22]	IO 5V	
210	PA[23]	IO 5V	
209	PA[24]	IO 5V	
190	PA[25]	IO 5V	
188	PA[26]	IO 5V	
187	PA[27]	IO 5V	
186	PA[28]	IO 5V	
185	PA[29]	IO 5V	
184	PA[30]	IO 5V	
183	PA[31]	IO 5V	

Номер вывода	Обозначение вывода	Тип вывода	Назначение и функции вывода
Порт B			
182	PB[0]	IO 5V	Пользовательский порт B ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL и SpaceWire MAC, Ethernet MAC, Timer При выборе JTAG_B Вывод PB[4] совмещен с JTAGB_nTRST Вывод PB[5] совмещен с JTAGB_SWCLKTCK Вывод PB[6] совмещен с JTAGB_SWDIOTMS Вывод PB[7] совмещен с JTAGB_TDI Вывод PB[8] совмещен с JTAGB_SWVTDO
181	PB[1]	IO 5V	
180	PB[2]	IO 5V	
179	PB[3]	IO 5V	
178	PB[4]	IO 5V	
177	PB[5]	IO 5V	
176	PB[6]	IO 5V	
174	PB[7]	IO 5V	
173	PB[8]	IO 5V	
172	PB[9]	IO 5V	
171	PB[10]	IO 5V	
170	PB[11]	IO 5V	
169	PB[12]	IO 5V	
167	PB[13]	IO 5V	
166	PB[14]	IO 5V	
165	PB[15]	IO 5V	
164	PB[16]	IO 5V	
163	PB[17]	IO 5V	
161	PB[18]	IO 5V	
154	PB[19]	IO 5V	
152	PB[20]	IO 5V	
151	PB[21]	IO 5V	
150	PB[22]	IO 5V	
149	PB[23]	IO 5V	
148	PB[24]	IO 5V	
147	PB[25]	IO 5V	
146	PB[26]	IO 5V	
145	PB[27]	IO 5V	
144	PB[28]	IO 5V	
143	PB[29]	IO 5V	
141	PB[30]	IO 5V	
140	PB[31]	IO 5V	

Номер вывода	Обозначение вывода	Тип вывода	Назначение и функции вывода
Порт С			
139	PC[0]	IO 5V	Пользовательский порт С ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer
138	PC[1]	IO 5V	
137	PC[2]	IO 5V	
135	PC[3]	IO 5V	
134	PC[4]	IO 5V	
133	PC[5]	IO 5V	
128	PC[6]	AIO	Вывод PC[6] совмещен с сигналом ADC0_REF0+
123	PC[7]	AIO	Вывод PC[7] совмещен с сигналом ADC1_REF0-
122	PC[8]	AIO	Вывод PC[8] совмещен с сигналом ADC2
121	PC[9]	AIO	Вывод PC[9] совмещен с сигналом ADC3
120	PC[10]	AIO	Вывод PC[10] совмещен с сигналом ADC4
117	PC[11]	AIO	Вывод PC[11] совмещен с сигналом ADC5
116	PC[12]	AIO	Вывод PC[12] совмещен с сигналом ADC6
115	PC[13]	AIO	Вывод PC[13] совмещен с сигналом ADC7
114	PC[14]	AIO	Вывод PC[14] совмещен с сигналом ADC8
112	PC[15]	AIO	Вывод PC[15] совмещен с сигналом ADC9
111	PC[16]	AIO	Вывод PC[16] совмещен с сигналом ADC10
110	PC[17]	AIO	Вывод PC[17] совмещен с сигналом ADC11
109	PC[18]	AIO	Вывод PC[18] совмещен с сигналом ADC12
107	PC[19]	AIO	Вывод PC[19] совмещен с сигналом ADC13
106	PC[20]	AIO	Вывод PC[20] совмещен с сигналом ADC14
105	PC[21]	AIO	Вывод PC[21] совмещен с сигналом ADC15
104	PC[22]	AIO	Вывод PC[22] совмещен с сигналом ADC16_REF1+
102	PC[23]	AIO	Вывод PC[23] совмещен с сигналом ADC17_REF1-
101	PC[24]	AIO	Вывод PC[24] совмещен с сигналом DAC0_REF+
100	PC[25]	AIO	Вывод PC[25] совмещен с сигналом DAC0_OUT
99	PC[26]	AIO	Вывод PC[26] совмещен с сигналом DAC0_REF-
98	PC[27]	AIO	Вывод PC[27] совмещен с сигналом DAC1_REF-
96	PC[28]	AIO	Вывод PC[28] совмещен с сигналом DAC1_OUT
95	PC[29]	AIO	Вывод PC[29] совмещен с сигналом DAC1_REF+
90	PC[30]	IO 5V	
89	PC[31]	IO 5V	

Номер вывода	Обозначение вывода	Тип вывода	Назначение и функции вывода
Порт D			
88	PD[0]	IO 5V	Пользовательский порт D ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer
86	PD[1]	IO 5V	
85	PD[2]	IO 5V	
84	PD[3]	IO 5V	
83	PD[4]	IO 5V	
82	PD[5]	IO 5V	
80	PD[6]	IO 5V	
79	PD[7]	IO 5V	
78	PD[8]	IO 5V	
77	PD[9]	IO 5V	
76	PD[10]	IO 5V	
74	PD[11]	IO 5V	
73	PD[12]	IO 5V	
72	PD[13]	IO 5V	
71	PD[14]	IO 5V	
62	PD[15]	IO 5V	
60	PD[16]	IO 5V	
59	PD[17]	IO 5V	
58	PD[18]	IO 5V	
57	PD[19]	IO 5V	
56	PD[20]	IO 5V	
55	PD[21]	IO 5V	
54	PD[22]	IO 5V	
53	PD[23]	IO 5V	
52	PD[24]	IO 5V	
51	PD[25]	IO 5V	
50	PD[26]	IO 5V	
49	PD[27]	IO 5V	
48	PD[28]	IO 5V	
47	PD[29]	IO 5V	
46	PD[30]	IO 5V	
45	PD[31]	IO 5V	

Номер вывода	Обозначение вывода	Тип вывода	Назначение и функции вывода
Порт E			
44	PE[0]	IO 5V	Пользовательский порт E ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, ARINC, MIL, Timer
43	PE[1]	IO 5V	
42	PE[2]	IO 5V	
41	PE[3]	IO 5V	
40	PE[4]	IO 5V	
39	PE[5]	IO 5V	
38	PE[6]	IO 5V	
37	PE[7]	IO 5V	
35	PE[8]	IO 5V	
34	PE[9]	IO 5V	
33	PE[10]	IO 5V	
31	PE[11]	IO 5V	
30	PE[12]	IO 5V	
29	PE[13]	IO 5V	
28	PE[14]	IO 5V	
27	PE[15]	IO 5V	
26	PE[16]	IO 5V	
19	PE[17]	IO 5V	
18	PE[18]	IO 5V	
17	PE[19]	IO 5V	
16	PE[20]	IO 5V	
15	PE[21]	IO 5V	
14	PE[22]	IO 5V	
13	PE[23]	IO 5V	
12	PE[24]	IO 5V	
11	PE[25]	IO 5V	
10	PE[26]	IO 5V	
9	PE[27]	IO 5V	
8	PE[28]	IO 5V	
7	PE[29]	IO 5V	
6	PE[30]	IO 5V	
5	PE[31]	IO 5V	

Номер вывода	Обозначение вывода	Тип вывода	Назначение и функции вывода
Интерфейс SpaceWire			
207	SPW_TXD-	AO	Сигналы приемопередатчика интерфейса SpaceWire
206	SPW_TXD+	AO	
205	SPW_TXS-	AO	
204	SPW_TXS+	AO	
203	SPW_RXS-	AI	
202	SPW_RXS+	AI	
201	SPW_RXD-	AI	
200	SPW_RXD+	AI	
Интерфейс Ethernet			
224	ETH_TX+	AO	Сигналы приемопередатчика интерфейса Ethernet
225	ETH_TX-	AO	
226	ETH_RX+	AI	
227	ETH_RX-	AI	
Сигналы управления			
1	RSTn0	I 5V	Вход нулевого сигнала сброса: 0 – запрос сброса; 1 – рабочий режим
3	RSTn1	I 5V	Вход первого сигнала сброса: 0 – запрос сброса; 1 – рабочий режим. Сброс микросхемы выполняется при условии запроса сброса по обоим выводам RSTn0 и RSTn1
4	FT_WRNn	O 5V	Выход с открытым стоком сигнала предупреждения об сбое от блока контроллера сбоев: 0 – предупреждение о сбое
2	FT_RSTn	O 5V	Выход с открытым стоком сигнала запроса сброса от блока: 0 – сигнал запроса сброса
255	JTAG_EN	I	Вход выбора тестового режима: 0 – рабочий режим; 1 – тестовый режим
68	STANDBY	O	Выход состояния встроенного регулятора: 0 – регулятор включен; 1 – регулятор программно выключен
69	WAKEUP	I 5V	Вход сигнала запроса включения регулятора: 1 – нет запроса включения; 0 – есть запрос включения встроенного регулятора
61	SHDN	I	Вход выключения встроенных регуляторов напряжения: 0 – регуляторы включены; 1 – регуляторы выключены

Номер вывода	Обозначение вывода	Тип вывода	Назначение и функции вывода
Сигналы тактирования			
253	HSE0_OSC_OUT	A	Вывод для подключения внешнего кварцевого резонатора к генератору HSE0
252	HSE0_OSC_IN	A	Вывод для подключения кварцевого резонатора или генератора к генератору HSE0
220	HSE1_OSC_OUT	A	Вывод для подключения внешнего кварцевого резонатора к генератору HSE1
219	HSE1_OSC_IN	A	Вывод для подключения кварцевого резонатора или генератора к генератору HSE1
66	LSE_OSC_IN	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору LSE
67	LSE_OSC_OUT	A	Вывод для подключения кварцевого резонатора к генератору LSE
Вспомогательные сигналы			
222	ETH_IREF	A	Контрольные вывода проверки источников опорных токов
91	DAC_IREF	A	
195	SPW_IREF	A	
196	SPW_EXTR	A	Выводы для подключения внешних токозадающих резисторов для формирования опорных токов R = 120 кОм ± 5 %
118	ADC_EXTR	A	
93	DAC_EXTR0	A	
94	DAC_EXTR1	A	
221	ETH_EXTR	A	
Питание			
131, 132	HV	PWR	Высоковольтный вход напряжения программирования однократно программируемой памяти программ. В рабочем режиме подключить к U_{CC}
92, 243	$DU_{CC}0$	PWR	Питание 1,8 В цифрового домена при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
129, 130	$DU_{CC}1$	PWR	
87	$DU_{CC}2$	PWR	
249	DU_{CC}_PLL	PWR	Питание 1,8 В домена PLL при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ
64	BDU_{CC}	PWR	Вывод подключения конденсатора к питанию регулятора LDO батарейного домена
65	BU_{CC}	PWR	Батарейное питание
63	DU_{CC}_B	PWR	Питание 1,8 В цифровой части батарейного домена при SHDN = 1. Выход регулятора LDO при SHDN = 0. Рекомендуется подключать к конденсатору емкостью 10 мкФ

Номер вывода	Обозначение вывода	Тип вывода	Назначение и функции вывода
20, 21, 70, 153, 155, 254	U_{CC}	PWR	Питание площадок ввода-вывода
24, 25, 81, 103, 124, 125, 159, 162, 193, 194, 213, 232, 233	$U_{CC}1$	PWR	Питание встроенного регулятора LDO для формирования напряжений питания $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL
108	DAC_U_{CC}	PWR	Питание аналоговых блоков ЦАП
126	ADC_U_{CC}	PWR	Питание аналоговых блоков АЦП
197	SPW_U_{CC}	PWR	Питание приемопередатчика SpaceWire PHY
228	ETH_U_{CC}	PWR	Питание приемопередатчика Ethernet PHY
22, 23, 75, 97, 127, 157, 160, 189, 191, 198, 208, 218, 237, 238	GND	PWR	Общий
113	DAC_GND	PWR	Общий аналоговых блоков ЦАП
119	ADC_GND	PWR	Общий аналоговых блоков АЦП
199	SPW_GND	PWR	Общий приемопередатчика SpaceWirePHY
223	ETH_GND	PWR	Общий приемопередатчика EthernetPHY
256	GND(NC)	PWR	Общий подложки кристалла
32, 36, 136, 142, 156, 158, 168, 175, 192	NC	–	Не подключены

Примечание – Типы выводов:

PWR – вывод питания;
I – цифровой вход;
O – цифровой выход;
IO – цифровой вход-выход;
A – аналоговых вывод;
AI – аналоговый вход;
AO – аналоговый выход;
AIO – цифровой вход-выход совмещенный с аналоговых выводом;
5V – толерантный к 5 В цифровой вход

4 Расположение выводов микросхемы

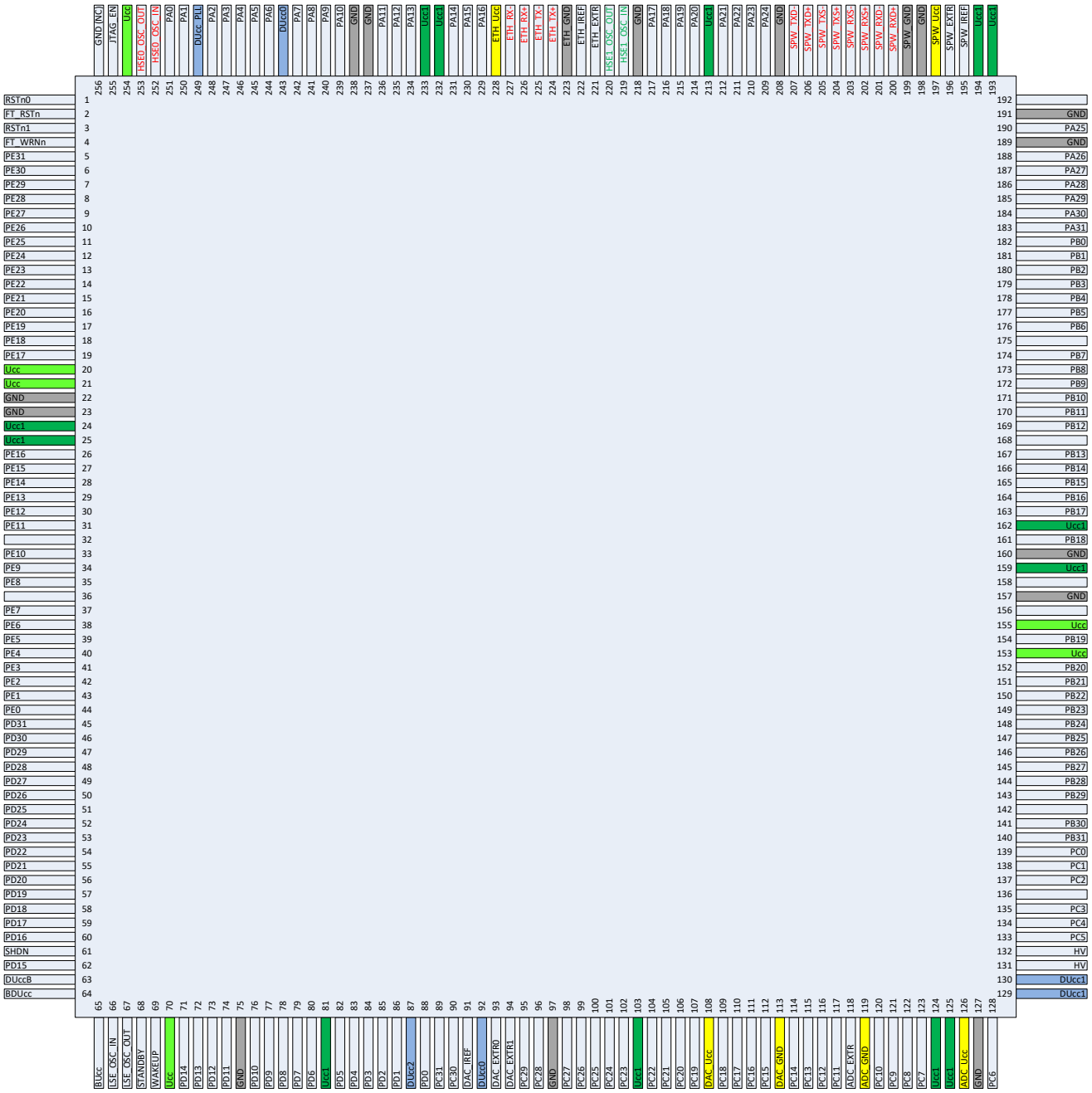


Рисунок 3 – Расположение выводов микросхемы

FUNC 15	FUNC 14	FUNC 13	FUNC 12	FUNC 11	FUNC 10	FUNC 9	FUNC 8	FUNC 7	FUNC 6	FUNC 5	FUNC 4	FUNC 3	FUNC 2	FUNC 1	FUNC 0	ANALOG	PIN	
																	RSTn0	1
																	FT_RSTn	2
																	RSTn1	3
																	FT_WRNn	4
		MIL0_RXBN	MIL1_RXBN		ARC0_IN6P	TMR0_CH3N		TMR3_CH1N		UART0_TX		CS[4]	DATA[41]		PE31		MODE[7]	5
		MIL0_RXBP	MIL1_RXBP		ARC0_IN6N	TMR0_CH3P		TMR3_CH1P			CAN0_RX	CS[5]	DATA[40]		PE30		MODE[6]	6
		MIL0_ENB	MIL1_ENB		ARC0_IN5P	TMR0_CH2N		TMR3_ETR			CAN0_TX	CS[6]	DATA[39]		PE29		MODE[5]	7
		MIL0_TXBP	MIL1_TXBP		ARC0_IN5N	TMR0_CH2P		TMR2_BRK				CS[7]	DATA[38]		PE28		MODE[4]	8
		MIL0_TXBN	MIL1_TXBN		ARC0_OUT2P	TMR0_CH1N		TMR2_CH4N				ADDR[31]	DATA[37]		PE27		MODE[3]	9
		MIL0_RXAN	MIL1_RXAN		ARC0_OUT2N	TMR0_CH1P		TMR2_CH4P	SSP0_RX	UART1_RX		ADDR[30]	DATA[36]		PE26		MODE[2]	10
		MIL0_RXAP	MIL1_RXAP		ARC0_IN4P	TMR0_ETR		TMR2_CH3N	SSP0_FS	UART1_TX		ADDR[29]	DATA[35]		PE25		MODE[1]	11
		MIL0_ENA	MIL1_ENA		ARC0_IN4N			TMR2_CH3P	SSP0_CLK	UART0_RX		ADDR[28]	DATA[34]		PE24		MODE[0]	12
		MIL0_TXAP	MIL1_TXAP		ARC0_IN3P			TMR2_CH2N	SSP0_TX	UART0_TX		ADDR[27]	DATA[33]		PE23			13
		MIL0_TXAN	MIL1_TXAN		ARC0_IN3N			TMR2_CH2P				ADDR[26]	DATA[32]		PE22			14
		MIL1_RXBN	MIL0_RXBN		ARC0_OUT1P		TMR3_ETR	TMR2_CH1N				ADDR[25]	DATA[71]		PE21			15
		MIL1_RXBP	MIL0_RXBP		ARC0_OUT1N		TMR3_CH1P	TMR2_CH1P			CAN0_RX	ADDR[24]	DATA[70]		PE20			16
		MIL1_ENB	MIL0_ENB		ARC0_IN2P		TMR3_CH1N	TMR2_ETR			CAN0_TX	ADDR[23]	DATA[69]		PE19			17
		MIL1_TXBP	MIL0_TXBP		ARC0_IN2N		TMR3_CH2P	TMR1_BRK		UART1_RX		ADDR[22]	DATA[68]		PE18			18
		MIL1_TXBN	MIL0_TXBN		ARC0_IN1P		TMR3_CH2N	TMR1_CH4N		UART1_TX		ADDR[21]	DATA[67]		PE17			19
																	Ucc	20
																	Ucc	21
																	GND	22
																	GND	23
																	Ucc1	24
																	Ucc1	25
		MIL1_RXAN	MIL0_RXAN		ARC0_IN1N		TMR3_CH3P	TMR1_CH4P		UART0_RX		CLOCK	DATA[66]		PE16			26
		MIL1_RXAP	MIL0_RXAP	ARC0_OUT4P			TMR3_CH3N	TMR1_CH3N		UART0_TX		OCLK	DATA[65]		PE15			27
		MIL1_ENA	MIL0_ENA	ARC0_OUT4N			TMR3_CH4P	TMR1_CH3P					DATA[64]		PE14			28
		MIL1_TXAP	MIL0_TXAP	ARC0_IN8P			TMR3_CH4N	TMR1_CH2N					DATA[15]		PE13			29
		MIL1_TXAN	MIL0_TXAN	ARC0_IN8N			TMR3_BRK	TMR1_CH2P					DATA[14]		PE12			30
		MIL0_RXBN	MIL1_RXBN	ARC0_IN7P			TMR2_ETR	TMR1_CH1N					DATA[13]		PE11			31
																		32
		MIL0_RXBP	MIL1_RXBP	ARC0_IN7N			TMR2_CH1P	TMR1_CH1P	SSP0_RX	UART1_RX	CAN0_RX		DATA[12]		PE10			33
		MIL0_ENB	MIL1_ENB	ARC0_OUT3P			TMR2_CH1N	TMR1_ETR	SSP0_FS	UART1_TX	CAN0_TX		DATA[11]		PE9			34
		MIL0_TXBP	MIL1_TXBP	ARC0_OUT3N			TMR2_CH2P	TMR0_BRK	SSP0_CLK	UART0_RX			DATA[10]		PE8			35
																		36
		MIL0_TXBN	MIL1_TXBN	ARC0_IN6P			TMR2_CH2N	TMR0_CH4N	SSP0_TX	UART0_TX			DATA[9]		PE7			37
		MIL0_RXAN	MIL1_RXAN	ARC0_IN6N			TMR2_CH3P	TMR0_CH4P					DATA[8]		PE6			38
		MIL0_RXAP	MIL1_RXAP	ARC0_IN5P			TMR2_CH3N	TMR0_CH3N					DATA[7]		PE5			39
		MIL0_ENA	MIL1_ENA	ARC0_IN5N		TMR3_BRK	TMR2_CH4P	TMR0_CH3P					DATA[6]		PE4			40
		MIL0_TXAP	MIL1_TXAP	ARC0_OUT2P		TMR3_CH4N	TMR2_CH4N	TMR0_CH2N					DATA[5]		PE3			41
		MIL0_TXAN	MIL1_TXAN	ARC0_OUT2N		TMR3_CH4P	TMR2_BRK	TMR0_CH2P		UART1_RX			DATA[4]		PE2			42
		MIL1_RXBN	MIL0_RXBN	ARC0_IN4P		TMR3_CH3N	TMR1_ETR	TMR0_CH1N		UART1_TX			DATA[3]		PE1			43
		MIL1_RXBP	MIL0_RXBP	ARC0_IN4N		TMR3_CH3P	TMR1_CH1P	TMR0_CH1P		UART0_RX	CAN0_RX		DATA[2]		PE0			44
		MIL1_ENB	MIL0_ENB	ARC0_IN3P		TMR3_CH2N	TMR1_CH1N	TMR0_ETR		UART0_TX	CAN0_TX	CLOCK	DATA[1]		PD31			45
		MIL1_TXBP	MIL0_TXBP	ARC0_IN3N		TMR3_CH2P	TMR1_CH2P					OCLK	DATA[0]		PD30			46
		MIL1_TXBN	MIL0_TXBN	ARC0_OUT1P		TMR3_CH1N	TMR1_CH2N					BEn[8]	BWEn[8]		PD29			47
		MIL1_RXAN	MIL0_RXAN	ARC0_OUT1N		TMR3_CH1P	TMR1_CH3P					BEn[5]	BWEn[5]		PD28			48
		MIL1_RXAP	MIL0_RXAP	ARC0_IN2P		TMR3_ETR	TMR1_CH3N					BEn[4]	BWEn[4]		PD27			49
		MIL1_ENA	MIL0_ENA	ARC0_IN2N		TMR2_BRK	TMR1_CH4P		SSP0_RX	UART1_RX		BEn[1]	BWEn[1]		PD26			50
		MIL1_TXAP	MIL0_TXAP	ARC0_IN1P		TMR2_CH4N	TMR1_CH4N		SSP0_FS	UART1_TX		BEn[0]	BWEn[0]		PD25			51
		MIL1_TXAN	MIL0_TXAN	ARC0_IN1N		TMR2_CH4P	TMR1_BRK		SSP0_CLK	UART0_RX			WEn[0]		PD24			52
		MIL0_RXBN	MIL1_RXBN		ARC0_OUT4P	TMR2_CH3N	TMR0_ETR		SSP0_TX	UART0_TX			OEn[0]		PD23			53
		MIL0_RXBP	MIL1_RXBP		ARC0_OUT4N	TMR2_CH3P	TMR0_CH1P				CAN0_RX	CLOCK	CSn[3]		PD22			54
		MIL0_ENB	MIL1_ENB		ARC0_IN8P	TMR2_CH2N	TMR0_CH1N				CAN0_TX	OCLK	CSn[2]		PD21			55
		MIL0_TXBP	MIL1_TXBP		ARC0_IN8N	TMR2_CH2P	TMR0_CH2P						CSn[1]		PD20			56
		MIL0_TXBN	MIL1_TXBN		ARC0_IN7P	TMR2_CH1N	TMR0_CH2N						CSn[0]		PD19			57
		MIL0_RXAN	MIL1_RXAN		ARC0_IN7N	TMR2_CH1P	TMR0_CH3P			UART1_RX			ADDR[20]		PD18			58
		MIL0_RXAP	MIL1_RXAP		ARC0_OUT3P	TMR2_ETR	TMR0_CH3N			UART1_TX			ADDR[19]		PD17			59
		MIL0_ENA	MIL1_ENA		ARC0_OUT3N	TMR1_BRK	TMR0_CH4P			UART0_RX			ADDR[18]		PD16			60
																	SHDN	61
		MIL0_TXAP	MIL1_TXAP		ARC0_IN6P	TMR1_CH3N	TMR0_CH4N			UART0_TX			ADDR[17]		PD15			62
																	DUccB	63
																	BDUcc	64
15 FUNC	14 FUNC	13 FUNC	12 FUNC	11 FUNC	10 FUNC	9 FUNC	8 FUNC	7 FUNC	6 FUNC	5 FUNC	4 FUNC	3 FUNC	2 FUNC	1 FUNC	0 FUNC	ANALOG	PIN	

Рисунок 4 – Функции выводов 1 ÷ 64 в различных режимах

FUNC 15	FUNC 14	FUNC 13	FUNC 12	FUNC 11	FUNC 10	FUNC 9	FUNC 8	FUNC 7	FUNC 6	FUNC 5	FUNC 4	FUNC 3	FUNC 2	FUNC 1	FUNC 0	ANALOG	PIN	
																	BUcc	65
																	LSE_OSC_IN	66
																	LSE_OSC_OUT	67
																	STANDBY	68
																	WAKEUP	69
																	Ucc	70
		MIL0_TXAN	MIL1_TXAN		ARC0_IN6N	TMR1_CH4P	TMR0_BRK						ADDR[16]		PD14			71
		MIL1_RXBN	MIL0_RXBN		ARC0_IN5P	TMR1_CH3N							ADDR[15]		PD13			72
		MIL1_RXBP	MIL0_RXBP		ARC0_IN5N	TMR1_CH3P					CAN0_RX		ADDR[14]		PD12			73
		MIL1_ENB	MIL0_ENB		ARC0_OUT2P	TMR1_CH2N					CAN0_TX		ADDR[13]		PD11			74
																	GND	75
		MIL1_TXBP	MIL0_TXBP		ARC0_OUT2N	TMR1_CH2P		TMR3_BRK	SSP0_RX	UART1_RX			ADDR[12]		PD10			76
		MIL1_TXBN	MIL0_TXBN		ARC0_IN4P	TMR1_CH1N		TMR3_CH4N	SSP0_FS	UART1_TX			ADDR[11]		PD9			77
		MIL1_RXAN	MIL0_RXAN		ARC0_IN4N	TMR1_CH1P		TMR3_CH4P	SSP0_CLK	UART0_RX			ADDR[10]		PD8			78
		MIL1_RXAP	MIL0_RXAP		ARC0_IN3P	TMR1_ETR		TMR3_CH3N	SSP0_TX	UART0_TX			ADDR[9]		PD7			79
		MIL1_ENA	MIL0_ENA		ARC0_IN3N	TMR0_BRK		TMR3_CH3P					ADDR[8]		PD6			80
																	Ucc1	81
		MIL1_TXAP	MIL0_TXAP		ARC0_OUT1P	TMR0_CH4N		TMR3_CH2N					ADDR[7]		PD5			82
		MIL1_TXAN	MIL0_TXAN		ARC0_OUT1N	TMR0_CH4P		TMR3_CH2P					ADDR[6]		PD4			83
		MIL0_RXBN	MIL1_RXBN		ARC0_IN2P	TMR0_CH3N		TMR3_CH1N					ADDR[5]		PD3			84
		MIL0_RXBP	MIL1_RXBP		ARC0_IN2N	TMR0_CH3P		TMR3_CH1P		UART1_RX	CAN0_RX		ADDR[4]		PD2			85
		MIL0_ENB	MIL1_ENB		ARC0_IN1P	TMR0_CH2N		TMR3_ETR		UART1_TX	CAN0_TX		ADDR[3]		PD1			86
																	DUcc2	87
		MIL0_TXBP	MIL1_TXBP		ARC0_IN1N	TMR0_CH2P		TMR2_BRK		UART0_RX			ADDR[2]		PD0			88
		MIL0_TXBN	MIL1_TXBN	ARC0_OUT4P		TMR0_CH1N		TMR2_CH4N		UART0_TX			ADDR[1]		PC31			89
		MIL0_RXAN	MIL1_RXAN	ARC0_OUT4N		TMR0_CH1P		TMR2_CH4P					ADDR[0]		PC30			90
																	DAC_IREF	91
																	DUcc0	92
																	DAC_EXTR0	93
																	DAC_EXTR1	94
		MIL0_RXAP	MIL1_RXAP	ARC0_IN8P		TMR0_ETR		TMR2_CH3N				CLOCK	ADDR[31]		PC29	DAC1_REF+		95
		MIL0_ENA	MIL1_ENA	ARC0_IN8N				TMR2_CH3P				OCLK			PC28	DAC1_OUT		96
																	GND	97
		MIL0_TXAP	MIL1_TXAP	ARC0_IN7P				TMR2_CH2N					ADDR[30]		PC27	DAC1_REF-		98
		MIL0_TXAN	MIL1_TXAN	ARC0_IN7N				TMR2_CH2P	SSP0_RX	UART1_RX			CS[5]		PC26	DAC0_REF-		99
		MIL1_RXBN	MIL0_RXBN	ARC0_OUT3P			TMR3_ETR	TMR2_CH1N	SSP0_FS	UART1_TX					PC25	DAC0_OUT		100
		MIL1_RXBP	MIL0_RXBP	ARC0_OUT3N			TMR3_CH1P	TMR2_CH1P	SSP0_CLK	UART0_RX	CAN0_RX		CS[6]		PC24	DAC0_REF+		101
		MIL1_ENB	MIL0_ENB	ARC0_IN6P			TMR3_CH1N	TMR2_ETR	SSP0_TX	UART0_TX	CAN0_TX	CLOCK			PC23	ADC17_REF1-		102
																	Ucc1	103
		MIL1_TXBP	MIL0_TXBP	ARC0_IN6N			TMR3_CH2P	TMR1_BRK				OCLK			PC22	ADC16_REF1+		104
		MIL1_TXBN	MIL0_TXBN	ARC0_IN5P			TMR3_CH2N	TMR1_CH4N					ADDR[29]		PC21	ADC15		105
		MIL1_RXAN	MIL0_RXAN	ARC0_IN5N			TMR3_CH3P	TMR1_CH4P					ADDR[28]		PC20	ADC14		106
		MIL1_RXAP	MIL0_RXAP	ARC0_OUT2P			TMR3_CH3N	TMR1_CH3N					ADDR[27]		PC19	ADC13		107
																	DAC_Ucc	108
		MIL1_ENA	MIL0_ENA	ARC0_OUT2N			TMR3_CH4P	TMR1_CH3P		UART1_RX			ADDR[26]		PC18	ADC12		109
		MIL1_TXAP	MIL0_TXAP	ARC0_IN4P			TMR3_CH4N	TMR1_CH2N		UART1_TX			ADDR[25]		PC17	ADC11		110
		MIL1_TXAN	MIL0_TXAN	ARC0_IN4N			TMR3_BRK	TMR1_CH2P		UART0_RX			ADDR[24]		PC16	ADC10		111
		MIL0_RXBN	MIL1_RXBN	ARC0_IN3P			TMR2_ETR	TMR1_CH1N		UART0_TX			ADDR[23]		PC15	ADC9		112
																	DAC_GND	113
		MIL0_RXBP	MIL1_RXBP	ARC0_IN3N			TMR2_CH1P	TMR1_CH1P			CAN0_RX		ADDR[22]		PC14	ADC8		114
		MIL0_ENB	MIL1_ENB	ARC0_OUT1P			TMR2_CH1N	TMR1_ETR			CAN0_TX		ADDR[21]		PC13	ADC7		115
		MIL0_TXBP	MIL1_TXBP	ARC0_OUT1N			TMR2_CH2P	TMR0_BRK					CS[7]		PC12	ADC6		116
		MIL0_TXBN	MIL1_TXBN	ARC0_IN2P			TMR2_CH2N	TMR0_CH4N							PC11	ADC5		117
																	ADC_EXTR	118
																	ADC_GND	119
		MIL0_RXAN	MIL1_RXAN	ARC0_IN2N			TMR2_CH3P	TMR0_CH4P	SSP0_RX	UART1_RX					PC10	ADC4		120
		MIL0_RXAP	MIL1_RXAP	ARC0_IN1P			TMR2_CH3N	TMR0_CH3N	SSP0_FS	UART1_TX					PC9	ADC3		121
		MIL0_ENA	MIL1_ENA	ARC0_IN1N		TMR3_BRK	TMR2_CH4P	TMR0_CH3P	SSP0_CLK	UART0_RX					PC8	ADC2		122
		MIL0_TXAP	MIL1_TXAP		ARC0_OUT4P	TMR3_CH4N	TMR2_CH4N	TMR0_CH2N	SSP0_TX	UART0_TX					PC7	ADC1_REF0-		123
																	Ucc1	124
																	Ucc1	125
																	ADC_Ucc	126
																	GND	127
		MIL0_TXAN	MIL1_TXAN		ARC0_OUT4N	TMR3_CH4P	TMR2_BRK	TMR0_CH2P							PC6	ADC0_REF0+		128
15 FUNC	14 FUNC	13 FUNC	12 FUNC	11 FUNC	10 FUNC	9 FUNC	8 FUNC	7 FUNC	6 FUNC	5 FUNC	4 FUNC	3 FUNC	2 FUNC	1 FUNC	0 FUNC	ANALOG	PIN	

Рисунок 5 – Функции выводов 65 ÷ 128 в различных режимах

	PIN	ANALOG	FUNC 0	FUNC 1	FUNC 2	FUNC 3	FUNC 4	FUNC 5	FUNC 6	FUNC 7	FUNC 8	FUNC 9	FUNC 10	FUNC 11	FUNC 12	FUNC 13	FUNC 14	FUNC 15
192																		
191	GND																	
190			PA25		DATA[57]	READY[2]		UART1_TX	SSP0_FS	TMR1_CH4N	TMR3_CH2N			ARC0_IN1P	MIL0_TXBN	MIL1_TXBN	MAC0_TXD2	
189	GND																	
188			PA26		DATA[56]	READY[3]		UART1_RX	SSP0_RX	TMR1_BRK	TMR3_CH2P			ARC0_IN2N	MIL0_TXBP	MIL1_TXBP	MAC0_TXD3	
187			PA27		DATA[55]	READY[4]	CAN0_TX			TMR2_ETR	TMR3_CH1N			ARC0_IN2P	MIL0_ENB	MIL1_ENB	MAC0_TXEN	
186			PA28		DATA[54]	READY[5]	CAN0_RX			TMR2_CH1P	TMR3_CH1P			ARC0_OUT1N	MIL0_RXBP	MIL1_RXBP	MAC0_TXERR	
185			PA29		DATA[53]	READY[6]				TMR2_CH1N	TMR3_ETR			ARC0_OUT1P	MIL0_RXBN	MIL1_RXBN	MAC0_TXCLK	
184			PA30		DATA[52]	READY[7]				TMR2_CH2P				ARC0_IN3N	MIL1_TXAN	MIL0_TXAN	MAC0_COL	
183			PA31		DATA[51]	CS[0]		UART0_TX		TMR2_CH2N				ARC0_IN3P	MIL1_TXAP	MIL0_TXAP	MAC0_CRS	
182			PB0		DATA[50]	CS[1]		UART0_RX		TMR2_CH3P				ARC0_IN4N	MIL1_ENA	MIL0_ENA	MAC0_TXERR	
181			PB1		DATA[49]	CS[2]		UART1_TX		TMR2_CH3N		TMR0_ETR		ARC0_IN4P	MIL1_RXAP	MIL0_RXAP		
180			PB2		DATA[48]	CS[3]		UART1_RX		TMR2_CH4P		TMR0_CH1P		ARC0_OUT2N	MIL1_RXAN	MIL0_RXAN	SPW0_RXD	
179			PB3		DATA[79]					TMR2_CH4N		TMR0_CH1N		ARC0_OUT2P	MIL1_TXBN	MIL0_TXBN	SPW0_RXS	
178	JB_NTRST		PB4		DATA[78]					TMR2_BRK		TMR0_CH2P		ARC0_IN5N	MIL1_TXBP	MIL0_TXBP	SPW0_TXD	
177	JB_SWCLKTCK		PB5		DATA[77]		CAN0_TX			TMR3_ETR		TMR0_CH2N		ARC0_IN5P	MIL1_ENB	MIL0_ENB	SPW0_TXS	
176	JB_SWDIOTMS		PB6		DATA[76]		CAN0_RX			TMR3_CH1P		TMR0_CH3P		ARC0_IN6N	MIL1_RXBP	MIL0_RXBP		
175																		
174	JB_TDI		PB7		DATA[75]			UART0_TX	SSP0_TX	TMR3_CH1N		TMR0_CH3N		ARC0_IN6P	MIL1_RXBN	MIL0_RXBN		
173	JB_SWVTDO		PB8		DATA[74]			UART0_RX	SSP0_CLK	TMR3_CH2P		TMR0_CH4P		ARC0_OUT3N	MIL0_TXAN	MIL1_TXAN		
172			PB9		DATA[73]	OCLK		UART1_TX	SSP0_FS	TMR3_CH2N		TMR0_CH4N		ARC0_OUT3P	MIL0_TXAP	MIL1_TXAP		
171			PB10		DATA[72]	CLOCK		UART1_RX	SSP0_RX	TMR3_CH3P		TMR0_BRK		ARC0_IN7N	MIL0_ENA	MIL1_ENA		
170			PB11		DATA[31]					TMR3_CH3N		TMR1_ETR		ARC0_IN7P	MIL0_RXAP	MIL1_RXAP		
169			PB12		DATA[30]					TMR3_CH4P		TMR1_CH1P		ARC0_IN8N	MIL0_RXAN	MIL1_RXAN		
168																		
167			PB13		DATA[29]					TMR3_CH4N		TMR1_CH1N		ARC0_IN8P	MIL0_TXBN	MIL1_TXBN		
166			PB14		DATA[28]					TMR3_BRK		TMR1_CH2P		ARC0_OUT4N	MIL0_TXBP	MIL1_TXBP		
165			PB15		DATA[27]		CAN0_TX	UART0_TX				TMR1_CH2N		ARC0_OUT4P	MIL0_ENB	MIL1_ENB		
164			PB16		DATA[26]		CAN0_RX	UART0_RX				TMR1_CH3P	ARC0_IN1N		MIL0_RXBP	MIL1_RXBP		
163			PB17		DATA[25]			UART1_TX				TMR1_CH3N	ARC0_IN1P		MIL0_RXBN	MIL1_RXBN		
162	Ucc1																	
161			PB18		DATA[24]			UART1_RX			TMR0_BRK	TMR1_CH4P	ARC0_IN2N		MIL1_TXAN	MIL0_TXAN		
160	GND																	
159	Ucc1																	
158																		
157	GND																	
156																		
155	Ucc																	
154			PB19		DATA[23]						TMR0_CH4N	TMR1_CH4N	ARC0_IN2P		MIL1_TXAP	MIL0_TXAP		
153	Ucc																	
152			PB20		DATA[22]						TMR0_CH4P	TMR1_BRK	ARC0_OUT1N		MIL1_ENA	MIL0_ENA		
151			PB21		DATA[21]						TMR0_CH3N	TMR2_ETR	ARC0_OUT1P		MIL1_RXAP	MIL0_RXAP		
150			PB22		DATA[20]						TMR0_CH3P	TMR2_CH1P	ARC0_IN3N		MIL1_RXAN	MIL0_RXAN		
149			PB23		DATA[19]			UART0_TX	SSP0_TX		TMR0_CH2N	TMR2_CH1N	ARC0_IN3P		MIL1_TXBN	MIL0_TXBN		
148			PB24		DATA[18]			UART0_RX	SSP0_CLK		TMR0_CH2P	TMR2_CH2P	ARC0_IN4N		MIL1_TXBP	MIL0_TXBP		
147			PB25		DATA[17]		CAN0_TX	UART1_TX	SSP0_FS		TMR0_CH1N	TMR2_CH2N	ARC0_IN4P		MIL1_ENB	MIL0_ENB		
146			PB26		DATA[16]		CAN0_RX	UART1_RX	SSP0_RX		TMR0_CH1P	TMR2_CH3P	ARC0_OUT2N		MIL1_RXBP	MIL0_RXBP		
145			PB27		CSn[7]	OCLK					TMR0_ETR	TMR2_CH3N	ARC0_OUT2P		MIL1_RXBN	MIL0_RXBN		
144			PB28		CSn[6]	CLOCK					TMR1_BRK	TMR2_CH4P	ARC0_IN5N		MIL0_TXAN	MIL1_TXAN		
143			PB29		CSn[5]						TMR1_CH4N	TMR2_CH4N	ARC0_IN5P		MIL0_TXAP	MIL1_TXAP		
142																		
141			PB30		CSn[4]						TMR1_CH4P	TMR2_BRK	ARC0_IN6N		MIL0_ENA	MIL1_ENA		
140			PB31		BWEn[9]	BEn[9]		UART0_TX			TMR1_CH3N	TMR3_ETR	ARC0_IN6P		MIL0_RXAP	MIL1_RXAP		
139			PC0		BWEn[7]	BEn[7]		UART0_RX			TMR1_CH3P	TMR3_CH1P	ARC0_OUT3N		MIL0_RXAN	MIL1_RXAN		
138			PC1		BWEn[6]	BEn[6]		UART1_TX			TMR1_CH2N	TMR3_CH1N	ARC0_OUT3P		MIL0_TXBN	MIL1_TXBN		
137			PC2		BWEn[3]	BEn[3]		UART1_RX			TMR1_CH2P	TMR3_CH2P	ARC0_IN7N		MIL0_TXBP	MIL1_TXBP		
136																		
135			PC3		BWEn[2]	BEn[2]	CAN0_TX			TMR0_ETR	TMR1_CH1N	TMR3_CH2N	ARC0_IN7P		MIL0_ENB	MIL1_ENB		
134			PC4		WEn[1]		CAN0_RX			TMR0_CH1P	TMR1_CH1P	TMR3_CH3P	ARC0_IN8N		MIL0_RXBP	MIL1_RXBP		
133			PC5		OEn[1]					TMR0_CH1N	TMR1_ETR	TMR3_CH3N	ARC0_IN8P		MIL0_RXBN	MIL1_RXBN		
132	HV																	
131	HV																	
130	DUcc1																	
129	DUcc1																	
	PIN	ANALOG	0 FUNC	1 FUNC	2 FUNC	3 FUNC	4 FUNC	5 FUNC	6 FUNC	7 FUNC	8 FUNC	9 FUNC	10 FUNC	11 FUNC	12 FUNC	13 FUNC	14 FUNC	15 FUNC

Рисунок 6 – Функции выводов 129 ÷ 192 в различных режимах

	PIN	ANALOG	FUNC 0	FUNC 1	FUNC 2	FUNC 3	FUNC 4	FUNC 5	FUNC 6	FUNC 7	FUNC 8	FUNC 9	FUNC 10	FUNC 11	FUNC 12	FUNC 13	FUNC 14	FUNC 15
256																		
255	JTAG_EN																	
254	Ucc																	
253	HSE0_OSC_OUT																	
252	HSE0_OSC_IN																	
251	JA_NTRST		PA0		DATA[42]			UART0_RX		TMR3_CH2P		TMR0_CH4P	ARC0_IN1N		MIL0_TXAN	MIL1_TXAN		
250	JA_SWCLKTCK		PA1		DATA[43]			UART1_TX		TMR3_CH2N		TMR0_CH4N	ARC0_IN1P		MIL0_TXAP	MIL1_TXAP		
249	DUcc_PLL																	
248	JA_SWDIOTMS		PA2		DATA[44]			UART1_RX		TMR3_CH3P		TMR0_BRK	ARC0_IN2N		MIL0_ENA	MIL1_ENA		
247	JA_TDI		PA3		DATA[45]					TMR3_CH3N		TMR1_ETR	ARC0_IN2P		MIL0_RXAP	MIL1_RXAP		
246	JA_SWVTDO		PA4		DATA[46]					TMR3_CH4P		TMR1_CH1P	ARC0_OUT1N		MIL0_RXAN	MIL1_RXAN		
245			PA5		DATA[47]					TMR3_CH4N		TMR1_CH1N	ARC0_OUT1P		MIL0_TXBN	MIL1_TXBN		
244			PA6		CS[4]					TMR3_BRK		TMR1_CH2P	ARC0_IN3N		MIL0_TXBP	MIL1_TXBP		
243	DUcc0																	
242			PA7		CS[3]		CAN0_TX	UART0_TX	SSP0_TX	TMR0_ETR		TMR1_CH2N	ARC0_IN3P		MIL0_ENB	MIL1_ENB	MAC0_MDC	
241			PA8		CS[2]		CAN0_RX	UART0_RX	SSP0_CLK	TMR0_CH1P		TMR1_CH3P	ARC0_IN4N		MIL0_RXBP	MIL1_RXBP	MAC0_MDCTR	
240			PA9		CS[1]			UART1_TX	SSP0_FS	TMR0_CH2N		TMR1_CH3N	ARC0_IN4P		MIL0_RXBN	MIL1_RXBN	MAC0_MDCOE	
239			PA10		CS[0]			UART1_RX	SSP0_RX	TMR0_CH2P	TMR2_BRK	TMR1_CH4P	ARC0_OUT2N		MIL1_TXAN	MIL0_TXAN	MAC0_MDI	
238	GND																	
237	GND																	
236			PA11		READY[7]	OCLK				TMR0_CH2N	TMR2_CH4N	TMR1_CH4N	ARC0_OUT2P		MIL1_TXAP	MIL0_TXAP	MAC0_MDO	
235			PA12		READY[6]	CLOCK				TMR0_CH3P	TMR2_CH4P	TMR1_BRK	ARC0_IN5N		MIL1_ENA	MIL0_ENA	MAC0_MDOTR	TRACE_DATA3
234			PA13		READY[5]					TMR0_CH3N	TMR2_CH3N		ARC0_IN5P		MIL1_RXAP	MIL0_RXAP	MAC0_MDOOE	TRACE_DATA2
233	Ucc1																	
232	Ucc1																	
231			PA14		READY[4]					TMR0_CH4P	TMR2_CH3P		ARC0_IN6N		MIL1_RXAN	MIL0_RXAN	MAC0_MDINT	TRACE_DATA1
230			PA15		READY[3]			UART0_TX		TMR0_CH4N	TMR2_CH2N		ARC0_IN6P		MIL1_TXBN	MIL0_TXBN		TRACE_DATA0
229			PA16		READY[2]			UART0_RX		TMR0_BRK	TMR2_CH2P		ARC0_OUT3N		MIL1_TXBP	MIL0_TXBP	MAC0_RXDV	TRACE_CLK
228	ETH_Ucc																	
227	ETH_RX-																	
226	ETH_RX+																	
225	ETH_TX-																	
224	ETH_TX+																	
223	ETH_GND																	
222	ETH_IREF																	
221	ETH_EXTR																	
220	HSE1_OSC_OUT																	
219	HSE1_OSC_IN																	
218	GND																	
217			PA17		READY[1]	READY[3]	CAN0_TX	UART1_TX		TMR1_ETR	TMR2_CH1N		ARC0_OUT3P		MIL1_ENB	MIL0_ENB	MAC0_RXD0	
216			PA18		READY[0]	READY[2]	CAN0_RX	UART1_RX		TMR1_CH1P	TMR2_CH1P		ARC0_IN7N		MIL1_RXBP	MIL0_RXBP	MAC0_RXD1	
215			PA19		DATA[63]					TMR1_CH1N	TMR2_ETR		ARC0_IN7P		MIL1_RXBN	MIL0_RXBN	MAC0_RXD2	TRACE_CLK
214			PA20		DATA[62]					TMR1_CH2P	TMR3_BRK		ARC0_IN8N		MIL0_TXAN	MIL1_TXAN	MAC0_RXD3	TRACE_DATA0
213	Ucc1																	
212			PA21		DATA[61]					TMR1_CH2N	TMR3_CH4N		ARC0_IN8P		MIL0_TXAP	MIL1_TXAP	MAC0_RXERR	TRACE_DATA1
211			PA22		DATA[60]					TMR1_CH3P	TMR3_CH4P		ARC0_OUT4N		MIL0_ENA	MIL1_ENA	MAC0_RXCLK	TRACE_DATA2
210			PA23		DATA[59]			UART0_TX	SSP0_TX	TMR1_CH3N	TMR3_CH3N		ARC0_OUT4P		MIL0_RXAP	MIL1_RXAP	MAC0_TXD0	TRACE_DATA3
209			PA24		DATA[58]			UART0_RX	SSP0_CLK	TMR1_CH4P	TMR3_CH3P			ARC0_IN1N	MIL0_RXAN	MIL1_RXAN	MAC0_TXD1	
208	GND																	
207	SPW_TXD-																	
206	SPW_TXD+																	
205	SPW_TXS-																	
204	SPW_TXS+																	
203	SPW_RXS-																	
202	SPW_RXS+																	
201	SPW_RXD-																	
200	SPW_RXD+																	
199	SPW_GND																	
198	GND																	
197	SPW_Ucc																	
196	SPW_EXTR																	
195	SPW_IREF																	
194	Ucc1																	
193	Ucc1																	
	PIN	ANALOG	0 FUNC	1 FUNC	2 FUNC	3 FUNC	4 FUNC	5 FUNC	6 FUNC	7 FUNC	8 FUNC	9 FUNC	10 FUNC	11 FUNC	12 FUNC	13 FUNC	14 FUNC	15 FUNC

Рисунок 7 – Функции выводов 193 ÷ 256 в различных режимах

5 Указания по применению и эксплуатации

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

Инструмент для пайки (сварки) и монтажа не должен иметь потенциал, превышающий 0,3 В относительно шины общий.

Типовая схема включения микросхем приведена на рисунке 168.

Металлизированные площадки основания корпуса A1 – A8 электрически соединены с выводами питания 81, 103, 124, 213, 233.

Металлизированные площадки основания корпуса B1 – B8 электрически соединены с выводами общий 75, 97, 198, 218, 238.

Крышка корпуса электрически изолирована от выводов микросхемы.

Вывод 256 корпуса микросхемы электрически соединен с подложкой кристалла. Вывод 256 соединять с шиной общий.

В зависимости от рабочей частоты микроконтроллера для правильной работы регулятора LDO должно быть настроено поле SRILOW[2:0] в регистрах REG_62_TMRx блока батарейного домена (при переходе на более высокую рабочую частоту настройка полей выполняется до перехода, при переходе с высокой частоты на низкую настройка полей выполняется после смены частоты).

Неиспользуемые выводы, настроенные или имеющие по умолчанию функцию цифрового входа, должны быть доопределены до допустимых логических уровней напряжения через резисторы доопределения.

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

– подача (включение микросхем) – общий, питание батарейного домена U_{CCB}, напряжение питания U_{CC}, U_{CC1}, входные сигналы или одновременно;

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

Программирование микросхем проводят в нормальных условиях при $U_{CC} = U_{CC1} = U_{CCB} = 3{,}0$ В.

После программирования проверку электрических параметров и ФК проводят при нормальной, повышенной и пониженной рабочей температуре среды.

Микросхемы, поставляемые с формованными выводами, извлекать из индивидуальной упаковки (контейнера) согласно инструкции, приведенной в разделе 18 «Инструкция по распаковыванию микросхем с формованными выводами».

Рекомендуемые размеры формовки выводов микросхем, поставляемых с неформованными выводами, приведены на рисунке 170.

6 Архитектура микросхемы

6.1 Питание

6.1.1 Домены питания

Микросхема имеет несколько доменов питания, описание которых приведено в таблице 3.

Таблица 3 – Домены питания

Обозначение	Допустимые значения, В		Описание
	Не менее	Не более	
BU_{CC}	2,0	5,5	Батарейное питание микросхемы. Используется для питания батарейного домена при отсутствии основного питания U_{CC}. Питание изолировано от U_{CC}. Переключение между BU_{CC} и U_{CC} происходит автоматически. Если в системе батарейное питание не требуется, вывод BU_{CC} должен быть объединен с U_{CC}
U_{CC}	3,0	5,5	Основное питание микросхемы. Используется для питания портов ввода-вывода
$U_{CC}1$	2,2	5,5	Входное питание встроенных регуляторов LDO, формирующих питание для цифровой части. На вход питания $U_{CC}1$ может подаваться основное питание U_{CC}, либо питание ниже, чем U_{CC}, но не менее 2,2 В и не более U_{CC}
$ETHU_{CC}$	3,0	5,5	Питание физических приемо-передатчиков Ethernet. Должно совпадать с питанием по уровню с $U_{CC} \pm 0{,}2$ В, не выходя за допустимые значения. Используется для снижения шумов по U_{CC}
$SPWU_{CC}$	3,0	5,5	Питание физических приемо-передатчиков SpaceWire. Должно совпадать с питанием по уровню с $U_{CC} \pm 0{,}2$ В, не выходя за допустимые значения. Используется для снижения шумов по U_{CC}
ADC_U_{CC}	3,0	5,5	Основное питание аналоговых блоков АЦП. Должно совпадать с питанием по уровню с $U_{CC} \pm 0{,}2$ В, не выходя за допустимые значения. Используется для снижения шумов по U_{CC}
DAC_U_{CC}	3,0	5,5	Основное питание аналоговых блоков ЦАП. Должно совпадать с питанием по уровню с $U_{CC} \pm 0{,}2$ В, не выходя за допустимые значения. Используется для снижения шумов по U_{CC}
BDU_{CC}	-	-	Питание батарейного домена. Выбирается из BU_{CC} или U_{CC} (не требуется подавать извне): при отсутствии U_{CC} выбирается BU_{CC}; при появлении $U_{CC} > Upor$ спустя время $T_{PORRSTn}$ переключается на U_{CC}; при снижении $U_{CC} < Upor$ переключается на BU_{CC} с минимальной задержкой

Обозначение	Допустимые значения, В		Описание
	Не менее	Не более	
DU_{CC}_B	1,62	1,98	Питание, сформированное с помощью внутреннего регулятора LDO для питания батарейного домена. Входным напряжением регулятора LDO является BDU_{CC}
DU_{CC}_PLL	1,62	1,98	Питание, сформированное с помощью внутреннего регулятора LDO для блока умножителей частоты и генераторов
DU_{CC}0,1,2	1,62	1,98	Питание основной цифровой части, сформированное с помощью внутреннего регулятора LDO
HV	2,4	U_{CC} + 0,2	Высоковольтное напряжение программирования U_{HVPROG} однократно программируемой памяти OTP. При обычной работе (не программирование) должно совпадать с питанием по уровню с U_{CC}+0,2 В, но не превышать допустимые значения
	7,6	8,0	Высоковольтное напряжение программирования U_{HVPROG} однократно программируемой памяти OTP. При программировании памяти

6.1.2 Сброс при включении питания

Блок сброса (схема POR – Power-on-Reset) предназначен для выработки аппаратного сброса при включении основного питания или просадке его ниже критического уровня. При включении питания, пока уровень U_{CC} не превысил значение U_{POR} (типовое значение 2,75 В), сигнал сброса U_{CC}RESET будет нулевым (сброс). После превышения U_{CC} уровня U_{POR} сигнал U_{CC}RESET в течение времени $T_{PORRSTn}$ удерживается в нулевом состоянии, затем переключается в единицу. Задержка $T_{PORRSTn}$ формируется на счетчике, работающем с тактовой частотой от генератора LSI. Если по каким-либо причинам произошел отказ генератора LSI, схема сформирует сигнал U_{CC}RESET с задержкой много меньшей, чем $T_{PORRSTn}$.

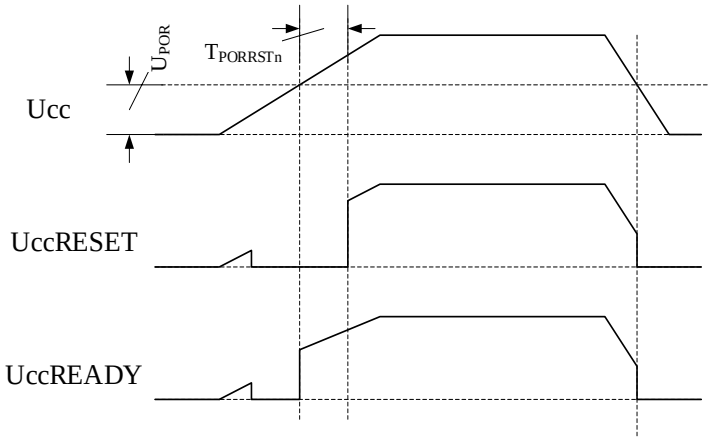


Рисунок 8 – Сигнал сброса при включении основного питания или просадке его ниже критического уровня

При снижении уровня U_{CC} ниже значения U_{POR}, сигнал U_{CC}RESET вырабатывается с минимальной задержкой.

6.1.3 Схема защиты от перенапряжения POVR

При превышении напряжением питания U_{CC} уровня U_{OVER} (типовое значение 5,75 В) формируется сигнал сброса OVRSTn. Таким образом, при превышении напряжением питания предельно-допустимого уровня происходит аппаратный сброс.

Схема POVR может быть заблокирована сигналом OVREN. В этом случае, сигнал OVRSTn всегда равен U_{CC}, Факт срабатывания схемы POVR фиксируется в регистре REG_60 батарейного домена и может быть проанализирован при последующем запуске системы.

6.1.4 Перекрытие рабочих диапазонов питания схемами POR и POVR

Чтобы исключить возможность продолжения работы схемы в диапазоне ниже или выше допустимого уровня напряжения питания, обеспечивается перекрытие рабочих диапазонов питания и уровней срабатывания схем POR и POVR. При отбраковачном тестировании схемы POR и POVR отключаются, и тестирование выполняется на уровнях ниже нижней границы срабатывания схемы POR и выше верхней границы срабатывания схемы POVR. После чего проводится проверка работы схем POR и POVR, подтверждающая, что их области срабатывания укладываются в допустимые диапазоны. Таким образом, при любых технологических отклонениях либо произойдет сброс, либо микросхема будет работать в допустимых режимах использования. При этом нельзя рассчитывать на применение микросхемы в условиях отличных от предельно-допустимых.

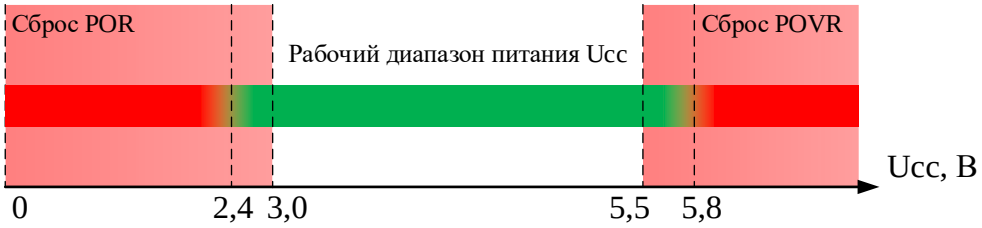


Рисунок 9 – Рабочий диапазон питания

6.1.5 Схема переключения основного и батарейного питаний (U_{CC} и BU_{CC})

Сигнал U_{CC}RESET является ключом для переключения питания батарейного домена. Таким образом, при отсутствии основного питания U_{CC} батарейный домен запитан от вывода BU_{CC}. При включении основного питания и превышении им уровня U_{POR} через время $T_{PORRSTn}$ питание батарейного домена переключится на основное питание. При снижении основного питания ниже уровня U_{POR} питание батарейного домена автоматически переключится на BU_{CC}. Так как переключение с U_{CC} на BU_{CC} происходит не мгновенно, возникает короткий момент времени, при котором на батарейный домен подается питание с уровнем ниже U_{POR}. Но диапазон допустимых

питаний батарейного домена BU_{CC} (не менее 2,0 и не более 5,5 В) перекрывает диапазон значений U_{POR}.

6.1.6 Блок монитора основного и батарейного питаний (U_{CC} и BU_{CC})

Для определения абсолютных значений уровней напряжения питания U_{CC} и BU_{CC} используется блок PVD, обеспечивающий определение уровня напряжения с типовой точностью ±100 мВ. Результат определения уровней напряжения отображается в блоке PWR_CNTR. При этом блок PWR_CNTR может быть настроен таким образом, чтобы при превышении или снижении некоторого заданного уровня напряжения вырабатывался сигнал прерывания для программной обработки данного события. При определении значения напряжения BU_{CC} потребление осуществляется от источника BU_{CC}, не зависимо от наличия или отсутствия U_{CC}. Таким образом, анализ BU_{CC} желательно не делать непрерывным, а осуществлять периодически с коротко-временным включением/отключением анализа BU_{CC}.

6.1.7 Управление питанием доменов DU_{CC}

Микросхема содержит несколько доменов DU_{CC}, которые включают в себя основные блоки микросхемы (см. таблицу 4).

Таблица 4 – Домены питания DU_{CC}

Название	Предельно-допустимые значения, В		Описание
	не менее	не более	
DU_{CC}_B	1,62	1,98	Питание, сформированное с помощью внутреннего регулятора LDO для батарейного домена
DU_{CC}_PLL	1,62	1,98	Питание, сформированное с помощью внутреннего регулятора LDO для блока умножителей частоты и генераторов
DU_{CC}0,1,2	1,62	1,98	Питание, сформированное с помощью внутреннего регулятора LDO для основной цифровой части (процессорное ядро, память, периферия)

Управление питанием доменов осуществляется в регистре батарейного домена. Также может быть отключено все питание DU_{CC}, кроме батарейного домена, переводом схемы в режим Standby. Биты управления питанием реализованы в батарейном домене.

6.1.8 Блок монитора тока потребления доменов DU_{CC}

В каждом блоке регулятора может быть включен блок датчика превышения тока. Датчик превышения тока вырабатывает сигнал CURFLG при превышении током уровня, задаваемого входом CURDUCC[2:0]. Флаг CURFLG формируется только на время превышения тока. Блок датчика превышения тока разрешается битами PDID_EN. Кроме того, регулятор формирует выходной аналоговый сигнал CURLVL, обозначающий

текущий уровень тока потребления. Данный сигнал заводится на отдельный канал АЦП и может быть преобразован в цифровое значение, соответствующее току потребления.

6.2 Тактовые частоты

Микросхема имеет несколько первичных источников тактирования. При включении питания автоматически запускаются генераторы LSI и HSI. Процессорное ядро начинает выполнять загрузочную программу, а затем и пользовательскую на частоте HSI. Переключение на другие источники тактирования выполняется программно в ходе выполнения пользовательской программой в соответствии с заложенным пользователем алгоритмом.

6.2.1 Встроенный низкоскоростной генератор LSI

При включении питания запускается встроенный RC-генератор LSI с частотой в диапазоне от 20 до 60 кГц. На основании этого генератора высчитывается задержка $T_{PORRSTn}$.

Сам генератор и его управление расположены в батарейном домене. Генератор может продолжать работать в STANDBY режиме, в том числе и при наличии только питания BU_{CC}.

6.2.2 Встроенный высокоскоростной генератор HSI

При включении питания запускается встроенный RC-генератор HSI с частотой в диапазоне от 6 до 10 МГц. На данной частоте запускается процессорное ядро.

6.2.3 Внешние высокоскоростные генераторы HSE0 и HSE1

Для формирования внешней стабильной высокой частоты применяются генераторы HSE0 и HSE1. Наличие двух генераторов обеспечивает возможность стабильной работы при отказе одного из генераторов. С помощью блока монитора тактовых частот может быть определен факт отказа одного из генераторов, выполнен аварийный переход на генератор HSI и программное восстановление работы системы на оставшемся работающем генераторе.

Генераторы HSE могут работать в одном из следующих режимов:

– режим работы с внешним резонатором (HSEn_BYP = «0»), допустимая частота f_{C_HSE} составляет от 1 до 30 МГц;

– режим работы с внешним генератором (HSEn_BYP = «1»), допустимая частота f_{C_HSE} составляет не более 100 МГц.

Выбор режима работы (HSEn_BYP), а также включение генераторов (HSEn_ON), осуществляется в регистре HSEn_CLK. В режиме работы с внешним генератором входная тактовая частота с входа HSEn_OSC_IN проходит напрямую на выход генератора HSE, при этом выход HSEn_OSC_OUT находится в третьем состоянии.

Также генераторы HSE имеют встроенные фильтры помех, исключающие прохождение коротких импульсов или «просечек». Данный фильтр предназначен для исключения кратковременных единичных высокоскоростных импульсов, которые могут

быть не обнаружены схемой монитора частоты, которая имеет большее время реакции. Разрешение работы фильтра помех осуществляется с помощью бита HSE_FILTER_EN в регистре HSEn_CLK.

Каждый генератор HSE имеет флаг готовности (бит HSEn_RDY в регистре HSEn_STAT), который вырабатывается при включении генератора и выходе его в нормальный режим работы.

6.2.4 Внешний низкоскоростной часовой генератор LSE

Для формирования внешней стабильной низкой частоты применяется генератор LSE. Данный генератор может работать в одном из следующих режимов:

– режим работы с внешним резонатором (LSE_BYP = «0»), допустимая частота f_{C_LSE} составляет от 20 до 40 кГц;

– режим работы с внешним генератором (LSE_BYP = «1»), допустимая частота f_{C_LSE} составляет не более 1 МГц.

Выбор режима работы (LSE_BYP), а также включение генератора (LSE_ON), осуществляется в регистрах REG_63_TMRx. В режиме работы с внешним генератором входная тактовая частота с входа LSE_OSC_IN проходит напрямую на выход генератора LSE, при этом выход LSE_OSC_OUT находится в третьем состоянии.

В микросхеме нет второго (дублирующего) генератора LSE. Отказ генератора LSE может быть обнаружен с помощью блока монитора тактовых частот, при этом дополнительно может быть выполнен аварийный переход на генератор HSI. Далее работа системы может быть восстановлена путем формирования низкой частоты с помощью оставшихся работающих генераторов.

Генератор LSE имеет встроенный фильтр помех, исключающий прохождение коротких импульсов или «просечек». Разрешение работы фильтра помех осуществляется с помощью бита FILTER_EN в регистрах REG_63_TMRx. Также генератор LSE имеет флаг готовности (бит LSE_RDY в регистрах REG_63_TMRx), который вырабатывается при включении генератора и выходе его в нормальный режим работы.

Генератор LSE может продолжать работать в режиме STANDBY, в том числе и при наличии только питания BU_{CC}.

6.2.5 Блок умножения тактовой частоты PLL0, PLL1 и PLL2

Для формирования высоких тактовых частот (от 40 МГц до 100 МГц) используются схемы умножения тактовых частот. В качестве источников опорной частоты для PLL выступают генераторы HSI, HSE0, HSE1 и их производные. С помощью PLL формируется частота $FOUT = FIN \cdot K_{NPLL}/((K_{QPLL}) \cdot (DV + 1))$ (подробнее см. рисунок 29). PLL имеет флаг готовности, который вырабатывается при включении схемы умножителя и выходе ее в нормальный режим работы, либо при перезапуске PLL с новыми коэффициентами умножения частоты. С помощью блока монитора частоты, можно определить значение частоты по отношению к другим источникам тактирования, обнаружить отказ PLL (снижение или отсутствие частоты, и превышение выходной частоты), выполнить аварийный переход на генератор HSI и программно восстановить

работу системы на других PLL или генераторах. Для одновременной работы двух и более PLL необходимо отключить ограничение по току для LDO, формирующее напряжение питания DU_{CC}_PLL, в регистрах батарейного домена REG_60_TMRx, бит DUCCLIM_EN[5].

6.2.6 Монитор частоты внешних генераторов и PLL (в блоке CLKCNTR)

Блок монитора частоты внешних генераторов и PLL предназначен для определения значения тактовых частот по отношению друг к другу, определения следующих событий сбоев в работе генераторов и блоков PLL:

- снижение частоты ниже предела;
- пропадание частоты (снижение ниже более жесткого предела);
- превышение частоты выше предела.

Монитор, в зависимости от настроек, может по-разному реагировать на различные события в тактовых частотах:

- вырабатывать сигнал предупреждения;
- аппаратно переключать схему тактирования на другой источник тактирования;
- осуществлять сброс микросхемы через блок FT_CNTR.

Определение соотношения частот между собой происходит путем сравнения значений счетчиков, считающих на разных частотах. Сравнивая это соотношение с различными пределами, определяется сбой. Для обнаружения факта сбоя, в зависимости от реального соотношения частот и заданных пределов их допустимого соотношения, может потребоваться некоторое время. Таким образом, при возникновении сбоя (EVENT) или отказа (ERROR) блоку монитора потребуется некоторое время для реакции на это событие. Для тактовой частоты CPUCLK также реализован блок монитора частоты, но он не имеет возможности аппаратного переключения на частоту HSI, и позволяет только обнаруживать факты сбоя или отказа.

6.3 Сигналы Сброса

Структурная схема формирования сигналов сброса представлена на рисунке 10.

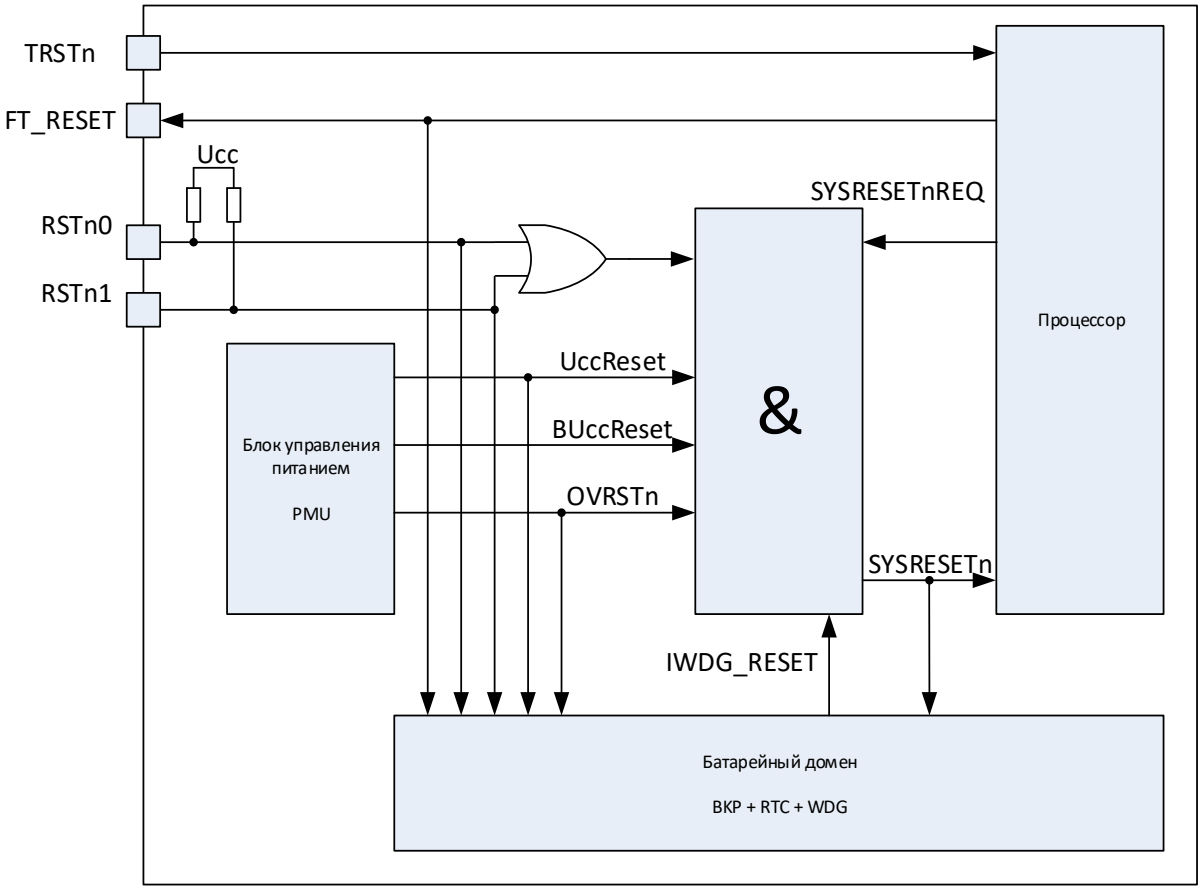


Рисунок 10 – Формирование сигналов сброса

6.3.1 Сигналы внешнего сброса RSTn0 и RSTn1

Микросхема может быть сброшена внешними сигналами с выводов RSTn0 и RSTn1. Сброс осуществляется при одновременной подаче низкого уровня на оба входа. Входы RSTn0 и RSTn1 имеет встроенную подтяжку к питанию с эквивалентным сопротивлением 50 кОм. Факт появления каждого сигнала RSTn0 и RSTn1 фиксируется в батарейном домене в битах S_RESET0n и S_RESET1n.

6.3.2 Сигнал сброса по питанию UccRESET

Начиная с включения питания U_{CC} и до превышения уровня U_{POR} в течение времени $T_{PORRSTn}$, а также при снижении U_{CC} ниже уровня U_{POR} вырабатывается сигнал сброса U_{CC}RESET. Этот сигнал сбрасывает микросхему в исходное состояние, за исключением режима STANDBY. В этом режиме микросхема может быть сброшена только сбросом BDU_{CC}RESET или сигналами пробуждения из режима STANDBY (сигнал WAKEUP или флаг ALRF будильника часов реального времени). Факт появления сигнала U_{CC}RESET фиксируется в батарейном домене в бите S_U_{CC}RESET.

6.3.3 Сигнал сброса по питанию DUccRESET

Домены цифрового питания DU_{CC} сбрасываются сигналом готовности DU_{CC}READY при отключении регуляторов. При выключении регуляторов сигнал DU_{CC}READY доопределяет в ноль сигнал SYSRESETn. Факт появления сигнала DU_{CC}RESET фиксируется в батарейном домене в бите S_SYSRSTn.

6.3.4 Сигнал программного запроса сброса SYSRESETnREQ

Микросхема может быть сброшена программным запросом со стороны процессорного ядра. Факт появления сигнала SYSRESETnREQ фиксируется в батарейном домене в бите S_SYSRSTn.

6.3.5 Сигналы аварийного сброса FT_RESET

Микросхема может быть сброшена блоком управления сбоями при обнаружении различных событий, также ряд событий всегда вызывает сброс микросхемы. Для этого необходимо вывод FT_RESET соединить с выводами внешнего сброса RSTn0 и RSTn1 вне микросхемы. При этом факт появления сигнала FT_RESETn фиксируется в батарейном домене в бите S_FT_RESET.

6.3.6 Сигналы сброса сторожевого таймера IWDG_RESET

Микросхема может быть сброшена сторожевым таймером WDG. Факт появления сигнала IWDG_RESET фиксируется в батарейном домене в бите S_WDG_RESET.

6.3.7 Сигнал сброса по перенапряжению POVR

При превышении питанием U_{CC} уровня U_{OVER} (не менее 5,5 и не более 5,7 В) формируется сигнал сброса OVRSTn. Таким образом, при превышении напряжением питания предельно-допустимых границ микросхема может быть аппаратно сброшена.

Схема POVR при проведении отбраковочного тестирования может быть заблокирована сигналом EXTPOVR. В этом случае сигнал OVRSTn всегда равен U_{CC}, что позволяет протестировать схему при питании выше уровня U_{OVER}. Отключать схему POVR в рабочем режиме не рекомендуется.

Факт срабатывания схемы POVR фиксируется в регистре батарейного домена в бите S_OVRSTn и может быть проанализирован при последующем запуске системы.

6.3.8 Сигнал сброса отладочного интерфейса TRSTn

Сигнал TRSTn служит для сброса отладочного интерфейса JTAG. Если отладочный интерфейс не используется данный вывод должен быть подтянут к логической единице внешним резистором. Установка в первоначальное значение отладочного интерфейса так же происходит по сигналу U_{CC}RESET.

6.4 Режимы работы микросхемы

Выбор режима запуска микросхемы определяется при включении или после любого сигнала сброса. Режим запуска задается комбинацией сигналов на входах MODE[7:0]. При этом режим закодирован кодом Хэмминга (7,4), таким образом, биты MODE[3:0] задают режим, а биты MODE[7:4] задают проверочные биты ECC. При выборе режима запуска автоматически исправляются одиночные ошибки в комбинации режима запуска, при обнаружении двойных ошибок микроксхема остается в режиме ожидания в загрузочной программе. Для определения режима запуска микросхема конфигурирует выводы MODE[7:0] (PE[31:24]) в цифровой режим на вход, включает внутренние резисторы подтяжки ~50 кОм на землю. После чего выдерживается пауза в 300 мкс для переопределения потенциалов на выводе MODE, и выполняется тройное считывание значений выводов MODE для выполнения мажоритарного контроля. Затем выполняется проверка ECC смажорированного значения MODE. И результирующее значение записывается в биты MODE регистров BKP_REG_60_TMRx. Проводится контроль записанного значения (отсутствие разницы в регистрах TMR0, 1, 2 батарейного домена), и на основании конечного значения MODE осуществляется переход в выбранный режим. При обнаружении неисправимой ошибки микросхема остается в режиме ожидания в загрузочной программе.

Таблица 5 – Режимы работы микросхемы

Биты ECC MODE[7:4]	Биты режима MODE[3:0]	Режим	Краткое описание
0000	0000	WAIT_BOOT_JA	Ожидание в бесконечном цикле с включенным интерфейсом отладки через выводы JTAG_A
0111	0001	OTP+JB	Запуск из OTP (0x0100_0000) с включенным интерфейсом отладки через выводы JTAG_B
1011	0010	OTP+JA	Запуск из OTP (0x0100_0000) с включенным интерфейсом отладки через выводы JTAG_A
1100	0011	EXTBUS_8_ECC+JB	Запуск из внешней памяти с (0x1000_0000) сконфигурированной в минимальный режим: – шина данных D[7:0] (PE[5:0]+PD[31:30]); – шина адреса A[15:0] (PD[13:0]+PC[31:30]); – сигнал nOE (PD[23]) с последовательной организацией ECC с базового адреса 0x1000_9000 с включенным интерфейсом отладки через выводы JTAG_B
1101	0100	EXTBUS_8_ECC+JA	Запуск из внешней памяти с (0x1000_0000) сконфигурированной в минимальный режим: – шина данных D[7:0] (PE[5:0]+PD[31:30]); – шина адреса A[15:0] (PD[13:0]+PC[31:30]); – сигнал nOE (PD[23]) с последовательной организацией ECC с базового адреса 0x1000_9000 с включенным интерфейсом отладки через выводы JTAG_A

Биты ECC MODE[7:4]	Биты режима MODE[3:0]	Режим	Краткое описание
1010	0101	EXTBUS_CFG+JB	Запуск из внешней памяти с (0x1000_0000) с чтением конфигурации в режиме: – шина данных D[7:0] (PE[5:0]+PD[31:30]); – шина адреса A[10:3] (PD[8:1]); – сигнал nOE (PD[23]); – сигнал nWE (PD[24]). Читается DATA[7:0] = ECC[3:0]+CFGx[3:0], где: – CFG0 = Режим запуска (8, 16, 32, 64); – CFG1 = Режим ECC (нет, послед., параллельн.); – CFG2 = ECCBASE[3:0]; – CFG3 = ECCBASE [7:4]; … – CFG9 = ECCBASE [31:28] с включенным интерфейсом отладки через выводы JTAG_B
0110	0110	EXTBUS_CFG+JA	Запуск из внешней памяти с (0x1000_0000) с чтением конфигурации в режиме: – шина данных D[7:0] (PE[5:0]+PD[31:30]); – шина адреса A[10:3] (PD[8:1]); – сигнал nOE (PD[23]); – сигнал nWE (PD[24]). Читается DATA[7:0] = ECC[3:0]+CFGx[3:0], где: – CFG0 = Режим запуска (8, 16, 32, 64); – CFG1 = Режим ECC (нет, послед., параллельн.); – CFG2 = ECCBASE [3:0]; – CFG3 = ECCBASE [7:4]; … – CFG9 = ECCBASE [31:28] с включенным интерфейсом отладки через выводы JTAG_A
0001	0111	SPI0+JB	Загрузка последовательно из внешней памяти по интерфейсу SPI с включенным интерфейсом отладки через выводы JTAG_B
1110	1000	SPI1+JA	Загрузка последовательно из внешней памяти по интерфейсу SPI с включенным интерфейсом отладки через выводы JTAG_A
1001	1001	SPI2+JB	Загрузка последовательно из внешней памяти по интерфейсу SPI с включенным интерфейсом отладки через выводы JTAG_B
0101	1010	SPI3+JA	Загрузка последовательно из внешней памяти по интерфейсу SPI с включенным интерфейсом отладки через выводы JTAG_A

Биты ECC MODE[7:4]	Биты режима MODE[3:0]	Режим	Краткое описание
0010	1011	UART0+JB	Загрузка последовательно из внешней памяти по интерфейсу UART с включенным интерфейсом отладки через выводы JTAG_B
0011	1100	UART0+JA	Загрузка последовательно из внешней памяти по интерфейсу UART с включенным интерфейсом отладки через выводы JTAG_A
0100	1101	ETT_K7_TEST	Тест предназначен для кратковременных и длительных испытаний на безотказность
1000	1110	-	Резерв
1111	1111	TEST_MODE+JB	Тестовый режим микросхемы для отбраковочного тестирования с включенным интерфейсом отладки через выводы JTAG_B
-	Двойная ошибка	WAIT_BOOT_JB	Ожидание в бесконечном цикле с включенным интерфейсом отладки через выводы JTAG_B

Выбор выводов для отладки определяется значением в троированных регистрах батарейного домена REG_60_TMRx битами MODE[0] и битом DISABLE_JTAG. При выборе того или иного интерфейса данные выводы жестко переопределяются на выполнение функции JTAG.

Загрузочная программа задействует для работы часть памяти ОЗУ по адресам 0x2000_7E00-0x2000_7FFF. При загрузке пользовательской программы в ОЗУ в режимах SPI0-3+JB/JA и UART0+JB/JA не следует использовать данные области.

6.4.1 Режим OTP+JA

После определения данного режима загрузочная программа выполняет следующие действия:

– устанавливает адрес указателя стека в значение, считанное из ячейки с адресом 0x0100_0000 (SP);

– устанавливает адрес таблицы векторов в значение 0x0100_0000 (VTOR);

– осуществляет безусловный переход по адресу, считанному из ячейки 0x0100_0004 (RESET_HANDLER).

В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении загрузочной программы.

В версии микросхемы с памятью ОЗУ вместо OTP предполагается, что ОЗУ памяти программ проинициализировано кодом заранее.

Отладка выполняется через выводы JTAG_A.

6.4.2 Режим OTP+JB

Аналогично OTP_JA, за исключением того, что выполняется отладка через выводы JTAG_B.

6.4.3 Режим EXTBUS_8_ECC+JA

После определения данного режима загрузочная программа конфигурирует внешнюю системную шину в режим:

- шина данных D[7:0] (PE[5:0]+PD[31:30]);
- шина адреса A[15:0] (PD[13:0]+PC[31:30]);
- сигнал nOE (PD[23]).

С последовательной организацией ECC с базового адреса 0x1000_9000 с включенным интерфейсом отладки через выводы JTAG_A.

После определения данного режима загрузочная программа выполняет следующие действия:

- устанавливает адрес указателя стека в значение, считанное из ячейки с адресом 0x1000_0000 (SP);
- устанавливает адрес таблицы векторов в значение 0x1000_0000 (VTOR);
- осуществляет безусловный переход по адресу, считанному из ячейки 0x1000_0004 (RESET_HANDLER).

В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении загрузочной программы.

6.4.4 Режим EXTBUS_8_ECC+JB

Аналогично режиму EXTBUS_8_ECC+JA, за исключением того, что отладка выполняется через выводы JTAG_B.

6.4.5 Режим EXTBUS_CFG+JA

После определения данного режима загрузочная программа конфигурирует внешнюю системную шину в режим:

- шина данных D[7:0] (PE[5:0]+PD[31:30]);
- шина адреса A[10:3] (PD[8:1]);
- сигнал nOE (PD[23]);
- Сигнал nWE (PD[24]).

С включенным интерфейсом отладки через выводы JTAG_A.

После этого из ячеек по адресам 0x1000_0400, 0x1000_0408, … 0x1000_0448 читается DATA[7:0] = ECC[3:0] + CFGx[3:0], и выполняется контроль ECC. При обнаружении одиночной ошибки загрузочная программа исправляет ее, при обнаружении двойной ошибки – формирует сигнал сброса.

После проверки, загрузочная программа конфигурирует внешнюю системную шину в зависимости от CFGx:

- шина данных (в зависимости от CFG0):
  - CFG0 = 1 – 8-битная шина;
  - CFG0 = 2 – 16-битная шина;
  - CFG0 = 3 – 32-битная шина;
  - CFG0 = 4 – 64-битная шина.

- шина адреса A[20:0] (PD[18:0]+PC[31:30])
- сигнал nOE (PD[23])
- контроль ECC (в зависимости от CFG1):
  - CFG1 = 1 – без ECC;
  - CFG1 = 2 – с последовательной ECC. Базовый адрес ECC устанавливается из CFG9-CFG2:
    - CFG2 = ECCBASE [3:0];
    - CFG3 = ECCBASE [7:4];
    - …
    - CFG9 = ECCBASE [31:28].
  - CFG1 = 3 – с параллельной ECC.

С включенным интерфейсом отладки через выводы JTAG_A.

После переопределения внешней системной шины загрузочная программа выполняет следующие действия:

– устанавливает адрес указателя стека в значение, считанное из ячейки с адресом 0x1000_0000 (SP);

– устанавливает адрес таблицы векторов в значение 0x1000_0000 (VTOR);

– осуществляет безусловный переход по адресу, считанному из ячейки 0x1000_0004 (RESET_HANDLER).

В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении загрузочной программы.

6.4.6 Режим EXTBUS_CFG+JB

Аналогично режиму EXTBUS_CFG+JA, за исключением того, что выполняется отладка через выводы JTAG_B.

6.4.7 Режим SPI0+JB

После определения данного режима загрузочная программа конфигурирует контроллер SSP0 для работы с микросхемами памяти по последовательному интерфейсу для конфигурирования ПЛИС (5576РТ1У, 5576РС1У и аналогичные):

- PA [7] – nSTATUS
- PA [8] – DCLK
- PA [9] – CONF_DONE
- PA [10] – DATA

и отладкой через выводы JTAG_B. В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении загрузочной программы.

Режим SPI0+JB необходим для записи в ОЗУ какой-либо программы (в частности программатора Flash-памяти), верификации ее и запуска на выполнение.

В качестве источника тактовой частоты SSP0 используется внутренний RC-генератор HSI с частотой 8 МГц.

Все данные EEPROM имеют порядок байт «little-endian», все байты имеют обратный порядок бит.

6.4.7.1 Параметры связи по SPI

Для связи по SPI выбраны следующие параметры канала связи:

– частота сигнала DCLK – 100 кГц;
– количество бит данных – 8;
– формат синхронного обмена – SPI фирмы Motorola, SPO = 1, SPH = 0.

6.4.7.2 Протокол обмена по SPI

После конфигурирования выводов загрузочная программа ожидает высокого уровня на выводе nSTATUS, что свидетельствует о готовности к работе внешней EEPROM. После обнаружения высокого логического уровня из EEPROM последовательно считывается информационный заголовок:

Информационный заголовок				
32 бит	32 бит	32 бит	16 бит	16 бит
OFFS*	OFFS + 4	OFFS + 8	OFFS + 12	OFFS + 14
SRAM ADDR	START ADDR	PROG LEN	CRC CNT LEN	CRC 16
* В некоторых EEPROM начальные адреса зарезервированы для конфигурации				

– 32-разрядный адрес (SRAM ADDR), по которому будет расположен образ копируемой программы. С учетом выравнивания таблицы векторов прерываний доступная область ОЗУ для размещения программы начинается с адреса 0x2000_0000.

– 32-разрядный адрес (START ADDR), которому будет передано управление после копирования программы. Как правило, это адрес обработчика RESET_HANDLER, расположенный в таблице векторов прерываний по смещению 0x4. Младший бит START ADDR должен быть установлен в «1».

– 32-разрядная длина (PROG LEN) образа программы в словах (32 бит) без учета CRC16.

– 16-разрядная длина данных в словах (CRC CNT LEN), по которым вычисляется CRC16 в образе программы.

– CRC16 информационного заголовка без учета поля CRC16.

После чтения информационного заголовка верифицируются считанное и рассчитанное значение CRC16. При несовпадении формируется сигнал сброса. После успешной проверки CRC16 загрузочная программа последовательно считывает образ программы:

Образ программы			
CRC CNT LEN *(32 бит)	16 бит	CRC CNT LEN *(32 бит)	16 бит
OFFS+16			
DATA	CRC16	DATA	CRC16

– CRC CNT LEN-слов данных программы и записывает их, начиная с адреса образа программы (SRAM ADDR).

– CRC16 и верифицирует с рассчитанным значением (при несовпадении CRC16 формируется сигнал сброса).

– CRC CNT LEN-слов данных программы и записывает их.

– CRC16 и верифицирует с рассчитанным значением (при несовпадении CRC16 формируется сигнал сброса).

– и т.д., пока внутренний счетчик не станет равным PROG LEN.

В данном случае, образ программы соответствует ситуации, когда половина данных программы участвует в подсчете CRC16 (CRC CNT LEN = PROG LEN / 2). Таким образом, в зависимости от CRC CNT LEN устанавливается необходимая избыточность информационных данных (CRC16).

Все данные EEPROM имеют порядок байт «little-endian», все байты имеют обратный порядок бит.

После загрузки образа программы во внутреннее ОЗУ загрузочная программа осуществляет безусловный переход по адресу START ADDR. Адрес указателя стека (SP) и адрес таблицы векторов (VTOR) не конфигурируются и должны быть установлены в пользовательской программе.

6.4.7.3 Вычисление CRC16

Для вычисления CRC16 используется полином вида: x^16 + x^15 + x^2 + 1 (0xA001) с начальным значением CRC16 0xFFFF:

```
uint16_t crc16_update (uint16_t crc, uint8_t a)
{
    int i;

    crc ^= a;
    for (i = 0; i < 8; ++i)
    {
      if (crc & 1)
        crc = (crc >> 1) ^ 0xA001;
      else
        crc = (crc >> 1);
    }
    return crc;
}
```

6.4.8 Режим SPI1+JA

Аналогично режиму SPI0+JB, за исключением того, что выполняется отладка через выводы JTAG_A.

6.4.9 Режим SPI2+JB

После определения данного режима загрузочная программа конфигурирует контроллер SSP0:

- PA [7] – TX
- PA [8] – CLK
- PA [9] – FS
- PA [10] – RX

и отладкой через выводы JTAG_B. В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении загрузочной программы.

Данный режим необходим для записи в ОЗУ какой-либо программы (в частности программатора Flash-памяти), верификации ее и запуска на выполнение.

В качестве источника тактовой частоты SSP0 используется внутренний RC-генератор HSI с частотой 8 МГц.

Все данные EEPROM имеют порядок байт «little-endian».

6.4.9.1 Параметры связи по SPI

Для связи по SPI выбраны следующие параметры канала связи:

- частота сигнала CLK – 100 кГц;
- количество бит данных – 8;
- формат синхронного обмена – SPI фирмы Motorola, SPO = 0, SPH = 0.

6.4.9.2 Протокол обмена по SPI

После конфигурирования выводов загрузочная программа из внешней EEPROM по адресу 0x000_000 считывает 32-разрядный адрес смещения информационного заголовка OFFS. Если смещение отлично от нуля, то операция чтения прерывается и возобновляется по адресу OFFS. Далее последовательно считывается информационный заголовок:

Информационный заголовок					
32 бит	32 бит	32 бит	32 бит	16 бит	16 бит
0x000_000	OFFS	OFFS + 4	OFFS + 8	OFFS + 12	OFFS + 14
OFFS	SRAM ADDR	START ADDR	PROG LEN	CRC CNT LEN	CRC 16

– 32-разрядный адрес (SRAM ADDR), по которому будет расположен образ копируемой программы. С учетом выравнивания таблицы векторов прерываний доступная область ОЗУ для размещения программы начинается с адреса 0x2000_0000.

– 32-разрядный адрес (START ADDR), которому будет передано управление после копирования программы. Как правило, это адрес обработчика RESET_HANDLER, расположенный в таблице векторов прерываний по смещению 0x4. Младший бит START ADDR должен быть установлен в «1».

– 32-разрядная длина (PROG LEN) образа программы в словах (32 бит) без учета CRC16.

– 16-разрядная длина данных в словах (CRC CNT LEN), по которым вычисляется CRC16 в образе программы.

– CRC16 информационного заголовка, включая смещение информационного заголовка (OFFS), без учета поля CRC16.

После чтения информационного заголовка верифицируются считанное и рассчитанное значение CRC16. При несовпадении формируется сигнал сброса. После успешной проверки CRC16 загрузочная программа последовательно считывает образ программы:

Образ программы			
CRC CNT LEN*(32 бит)	16 бит	CRC CNT LEN *(32 бит)	16 бит
OFFS+16			
DATA	CRC16	DATA	CRC16

– CRC CNT LEN-слов данных программы и записывает их, начиная с адреса образа программы (SRAM_ADDR).

– CRC16 и верифицирует с рассчитанным значением (при несовпадении CRC16 формируется сигнал сброса).

– CRC CNT LEN-слов данных программы и записывает их.

– CRC16 и верифицирует с рассчитанным значением (при несовпадении CRC16 формируется сигнал сброса).

– И т.д., пока внутренний счетчик не станет равным PROG_LEN.

В данном случае образ программы соответствует ситуации, когда половина данных программы участвует в подсчете CRC16 (CRC CNT LEN = PROG LEN / 2). Таким образом, в зависимости от CRC CNT LEN устанавливается необходимая избыточность информационных данных (CRC16).

Все данные EEPROM имеют порядок байт «little-endian».

После загрузки образа программы во внутреннее ОЗУ загрузочная программа осуществляет безусловный переход по адресу START ADDR. Адрес указателя стека (SP) и адрес таблицы векторов (VTOR) не конфигурируются и должны быть установлены в пользовательской программе.

6.4.9.3 **Вычисление CRC16**

Для вычисления CRC16 используется полином вида: x^16 + x^15 + x^2 + 1 (0xA001) и начальным значением CRC16 0xFFFF:

```
uint16_t crc16_update (uint16_t crc, uint8_t a)
{
    int i;
```

```
        crc ^= a;
        for (i = 0; i < 8; ++i)
        {
            if (crc & 1)
                crc = (crc >> 1) ^ 0xA001;
            else
                crc = (crc >> 1);
        }
        return crc;
    }
```

6.4.10 Режим SPI3+JA

Аналогично режиму SPI2+JB за исключением того, что выполняется отладка через выводы JTAG_A.

6.4.11 Режим UART0+JA

После определения данного режима загрузочная программа конфигурирует контроллер UART0 TX(PE[15]), RX(PE[16]) с отладкой через выводы JTAG_A.

В специальной структуре в ОЗУ передаются флаги об ошибках при выполнении загрузочной программы.

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

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

6.4.11.1 **Параметры связи по UART**

Для связи по UART выбраны следующие параметры канала связи:

– начальная скорость – 9600 бод;
– количество бит данных – 8;
– четность – нет;
– количество Stop бит – 1;
– загрузчик не использует FIFO UART0;
– загрузчик всегда выступает в качестве Slave, а внешнее устройство, подающее команды – в качестве Master;
– данные передаются младшим битом вперед.

6.4.11.2 **Протокол обмена по UART**

После определения режима UART необходима синхронизация с устройством Master. После синхронизации устройству Master доступны команды, приведенные в таблице 6.

6.4.11.3 **Синхронизация с внешним устройством**

Начальные условия для дальнейшего обмена по UART.

Master постоянно посылает в канал символ синхронизации 0x00. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ошибки дальнейшего 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 0x00 и теперь может подавать команды согласно протоколу обмена.

6.4.11.4 **Команды UART-загрузчика**

Таблица 6 – Список команд UART-загрузчика

Команда	Код	Символ ASCII	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

6.4.11.5 **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.

6.4.11.6 **Команда CMD_SYNC**

Пустая команда.

Загрузчик (Slave) ее принимает, но ничего по ней не делает.

Таблица 7 – Команда CMD_SYNC

Код команды	**CMD_SYNC = 0x00**
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0

Формат команды:	
Master: Выдает код команды CMD_SYNC	Slave: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды

6.4.11.7 **Команда CMD_CR**

Выдача приглашения устройству Master от микросхемы.

Таблица 8 – Команда CMD_CR

Код команды	**CMD_CR = 0x0D**
ASCII символ, соответствующий коду команды	нет
Количество параметров команды	0
Формат команды:	
Master: Выдает код команды CMD_CR	Slave: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды. Иначе выдает 3 байта: – код команды CMD_CR; – код 0x0A; – код 0x3E (ASCII символ '>')

6.4.11.8 **Команда CMD_BAUD**

Установка скорости обмена.

Таблица 9 – Команда CMD_BAUD

Код команды	**CMD_BAUD = 0x42**
ASCII символ, соответствующий коду команды	'B'
Количество параметров команды	1
Параметр	Новое значение скорости обмена [бод]
Формат команды:	
Master: Выдает код команды CMD_BAUD	Slave: Если команда принята с ошибками, выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр	Slave: Если параметр принят с ошибками, выдает код ошибки ERR_CHN или ERR_BAUD и завершает обработку текущей команды. Иначе: – выдает код команды CMD_BAUD; – после передачи кода команды CMD_BAUD устанавливает новое значение скорости обмена

6.4.11.9 **Команда CMD_LOAD**

Загрузка массива байт в память микросхемы.

Таблица 10 – Команда CMD_LOAD

Код команды	**CMD_LOAD = 0x4C**
ASCII символ, соответствующий коду команды	'L'
Количество параметров команды	2
Параметр 1	Адрес памяти приемника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master: Выдает код команды CMD_LOAD	Slave: Если команда принята с ошибками, выдает код ошибки ERR_CHN или ERR_CMD 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, выдает код ошибки ERR_CHN 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, выдает код ошибки ERR_CHN и завершает обработку текущей команды, не дожидаясь окончания принятия всего массива. По окончании приема массива выдает код ответа REPLY_OK = 0x4B ('K')
Примечание – С учетом выравнивания таблицы векторов прерываний, доступная область ОЗУ для размещения пользовательской программы начинается с адреса 0x2000_0000	

6.4.11.10 **Команда CMD_VFY**

Выдача массива байт из памяти микросхемы.

Таблица 11 – Команда CMD_VFY

Код команды	**CMD_VFY = 0x59**
ASCII символ, соответствующий коду команды	'Y'
Количество параметров команды	2
Параметр 1	Адрес памяти источника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master: Выдает код команды CMD_VFY	Slave: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD 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 код ошибки ERR_CHN и завершает обработку текущей команды. Иначе:

	– выдает код команды CMD_VFY; – выдает массив байт младшим байтом вперед; – по окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K')

6.4.11.11 **Команда CMD_RUN**

Запуск программы на выполнение.

Таблица 12 – Команда CMD_RUN

Код команды	**CMD_RUN = 0x52**
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр	Адрес первой команды загруженной программы (младший бит адреса необходимо установить в 1). Как правило, это адрес обработчика RESET_HANDLER, расположенный в таблице векторов прерываний по смещению 0x4
Формат команды:	
Master: Выдает код команды CMD_RUN	Slave: Если команда принята с ошибками, то выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр	Slave: Если параметр принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды. Иначе: – выдает код команды CMD_RUN; – устанавливает значение PC согласно принятому адресу (MSP и NVIC не перепрограммируются) и, таким образом, Slave завершает свое выполнение. Передача управления загруженной программе происходит, не дожидаясь окончания отправки кода команды CMD_RUN

6.4.11.12 **Сообщения об ошибках**

Сообщения об ошибках – это двухбайтные последовательности символов. Первый символ всегда 0x45 ('E'). Второй символ определяет тип ошибки.

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.

Возможны следующие сообщения об ошибках: ERR_CHN, ERR_CMD, ERR_BAUD.

Ошибка ERR_CHN

Аппаратная ошибка UART.

Код ошибки 0x69 ('i').

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

Ошибка ERR_CMD

Принята неизвестная команда.

Код ошибки 0x63 ('c').

Выдается диспетчером команд, если принят неизвестный код команды.

Ошибка ERR_BAUD

Принята неизвестная команда.

Код ошибки 0x62 ('b').

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

6.4.12 Режим UART0+JB

Аналогично режиму UART0+JA, за исключением того, что используются выводы TX(PA[7]), RX(PA[8]) и выполняется отладка через выводы JTAG_B.

6.4.13 Режим TEST_MODE+JB

В данном режиме на выводы JTAG_B вместо TAP контроллера процессорных ядер подключится технологический TAP контроллер, позволяющий перевести микросхему в тестовые режимы.

6.4.14 Режим WAIT_BOOT_JA

Ожидание в бесконечном цикле с включенным интерфейсом отладки через выводы JTAG_A.

6.4.15 Режим WAIT_BOOT_JB

Ожидание в бесконечном цикле с включенным интерфейсом отладки через выводы JTAG_B.

6.4.16 Статус загрузчика

Последние четыре слова в ОЗУ используются для передачи ошибок, которые могли возникнуть в процессе загрузки контроллера. Таким образом, после завершения работы загрузочной программы и передачи управления, ошибки могут быть считаны из ОЗУ для корректирования режима и минимизации ошибок при последующих запусках.

Таблица 13 – Поля статуса загрузчика

Адрес ОЗУ	Размер, бит	Обозначение	Описание
0x2000_7FE4	32	REG_60_TMR0	Значение регистра REG_60_TMR0 после сброса
0x2000_7FE8	32	REG_60_TMR1	Значение регистра REG_60_TMR1 после сброса
0x2000_7FEC	32	REG_60_TMR2	Значение регистра REG_60_TMR2 после сброса
0x2000_7FF0	8	INIT_ERR_CNTR	Число повторных инициализаций выводов MODE[7:0]. Значение INIT_ERR_CNTR > 2 говорит об ошибке в контроллере порта
0x2000_7FF1	8	MODE_0	Считанное значение MODE[0]
0x2000_7FF2	8	MODE_1	Считанное значение MODE[1]
0x2000_7FF3	8	MODE_2	Считанное значение MODE[2]
0x2000_7FF4	8	MODE_MAJ	Значение MODE после мажоритарного контроля MODE[0], MODE[1] и MODE[2]
0x2000_7FF5	8	MODE_MAJ_ERR	Ошибка мажоритарного контроля: 0 – Нет различия между MODE[0], MODE[1] и MODE[2] 1 – Есть различия между MODE[0], MODE[1] и MODE[2]
0x2000_7FF6	8	SYNDROM	Синдром: 1 – ошибка в MODE [5]; 2 – ошибка в MODE [6]; 3 – ошибка в MODE [0]; 4 – ошибка в MODE [7]; 5 – ошибка в MODE [1]; 6 – ошибка в MODE [2]; 7 – ошибка в MODE [3]
0x2000_7FF7	8	PARITY_BIT	Четность MODE
0x2000_7FF8	8	ECC_ERR_TYPE	Тип ошибки при контроле ECC: 0 – нет ошибки; 1 – одиночная ошибка; 2 – двойная ошибка; 3 – другие ошибки нечетной кратности
0x2000_7FF9	8	MODE_CORR	Скорректированное значение MODE. При обнаружении двойных или других ошибок нечетной кратности, MODE_CORR совпадает с MODE_MAJ
0x2000_7FFA	8	MODE_EXTBUS	Исправленное значение конфигурации (ECC+CFG0) в режиме EXTBUS_CFG+JB или EXTBUS_CFG+JA

Адрес ОЗУ	Размер, бит	Обозначение	Описание
0x2000_7FFB	8	MODE_ECC_EXTBUS	Исправленное значение конфигурации (ECC+CFG1) в режиме EXTBUS_CFG+JB или EXTBUS_CFG+JA
0x2000_7FFC	32	ECC_BASE_EXTBUS	Исправленное значение базового адреса ECC CFG(9-2) в режиме EXTBUS_CFG+JB или EXTBUS_CFG+JA

6.5 Организация памяти

Микросхема имеет Фон-Неймановскую архитектуру памяти, то есть все адресное пространство (память программ и память данных) отображены в общее адресное пространство размером 4 Гбайт. Но при этом память разделена на две секции: CODESECTION (от 0x0000_0000 до 0x1FFF_FFFF) для кода программ и DATASECTION (от 0x2000_0000 до 0xDFFF_FFFF) для данных. Доступ к CODESECTION осуществляется через шины IBUS и DBUS. Доступ к DATASECTION осуществляется через шину SBUS. Блок прямого доступа к памяти DMA имеет выделенную шину MBUS и может обращаться к памяти в диапазоне DATASECTION и областям памяти EXTBUS. При этом код программы может располагаться в DATASECTION, но при его выполнении снижается производительность из-за возникающих конфликтов по доступу за инструкциями и данными. Данные могут располагаться в CODESECTION в областях RAMD или Внешней памяти ОЗУ, отображаемой в эту секцию.

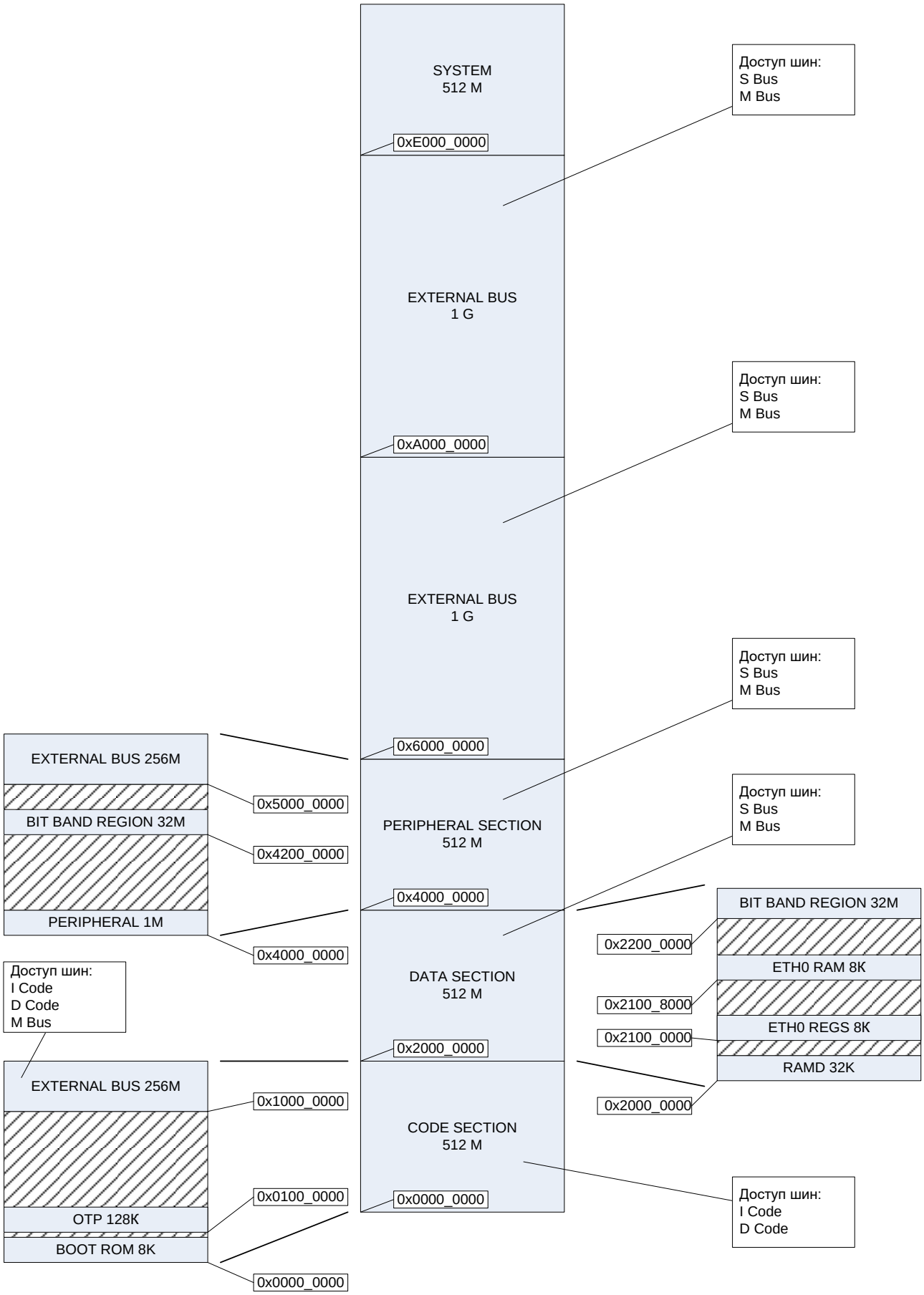


Рисунок 11 – Распределение адресного пространства

Возможная коммутация агентов в блоке BUSMATRIX приведена на рисунке 12.

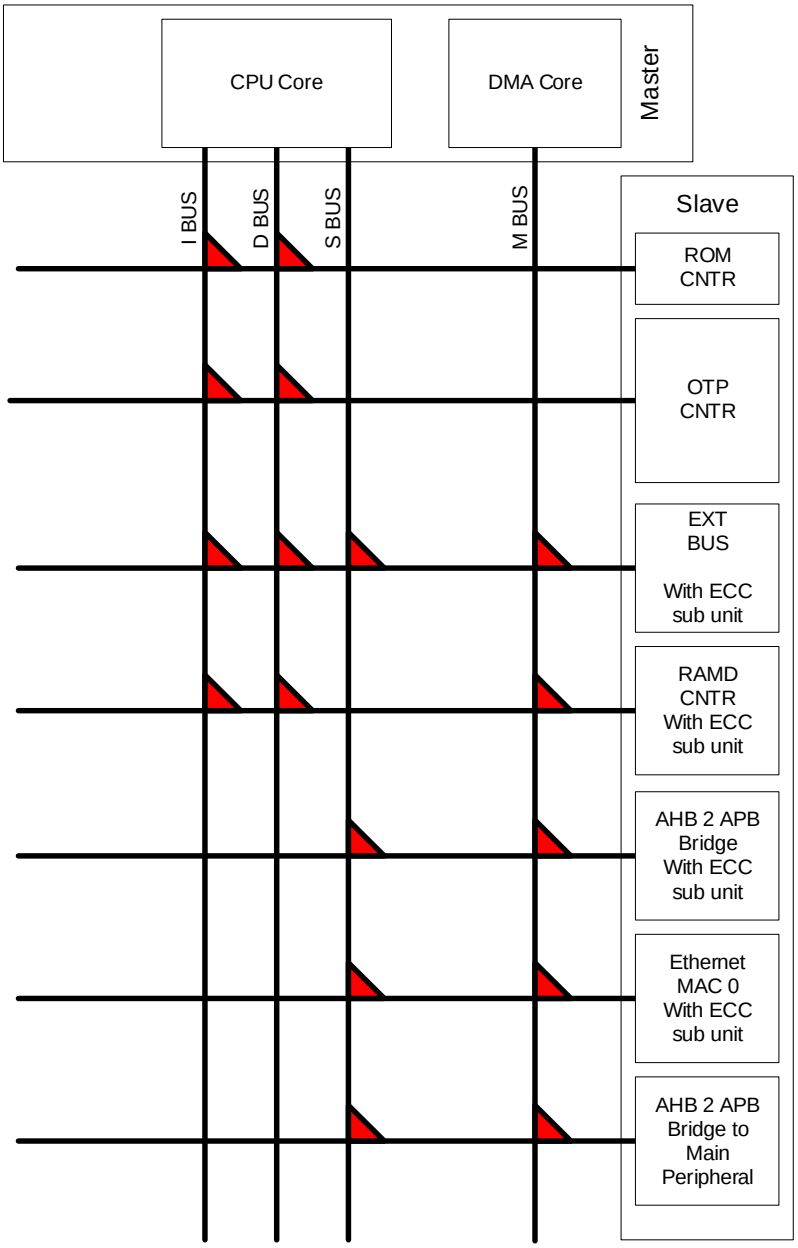


Рисунок 12 – Схема внутренних межсоединений

6.5.1 Распределение областей памяти

Таблица 14 – Распределение адресного пространства

Базовый адрес	Размер, байт	Обозначение	Доступ по шинам	Описание
0x0000_0000	8K	BOOT ROM	I, D	Область памяти загрузочного ПЗУ. Из этой памяти микросхема начинает работу, определяет дальнейший режим функционирования и выполняет необходимые настройки

Базовый адрес	Размер, байт	Обозначение	Доступ по шинам	Описание
0x0100_0000	128K	OTP	I, D	Однократно программируемая память программ, выполненная по технологии anti-fuse. Основная память программ для пользовательской программы
0x0200_0000	32K	-	-	-
0x0300_0000	128M			
0x1000_0000	256M	EXT BUS	I, D, M	Область памяти для отображения внешней системной шины
0x2000_0000	32K	RAMD	S, M	Статическое ОЗУ в DATASECTION, предназначено для данных
0x2100_0000	8K	ETH0 REGS	S, M	Регистры блока контроллера ETHERNETMAC 0
0x2100_8000	8K	ETH0 RAM	S, M	Статическое ОЗУ блока контроллера ETHERNETMAC 0
0x2101_0000		-	-	-
0x2101_8000		-	-	-
0x2102_0000				
0x2103_0000				
0x2200_0000	32M	BIT BAND REGION RAMD	S	Функционально выделенная область памяти на уровне процессорного ядра, позволяющая упростить битовые манипуляции в RAMD
0x4000_0000	1M	PERIPHERAL	S, M	Область отображения регистров периферии шины APB. Подробнее см. таблицу 15
0x4200_0000	32M	BIT BAND REGION PERIPHERAL	S	Функционально выделенная область памяти на уровне процессорного ядра, позволяющая упростить битовые манипуляции в PERIPHERAL
0x5000_0000	256M	EXT BUS	S, M	Область памяти для отображения внешней системной шины
0x6000_0000	1G	EXT BUS	S, M	Область памяти для отображения внешней системной шины
0xA000_0000	1G	EXT BUS	S, M	Область памяти для отображения внешней системной шины
0xE000_0000	512M	SYSTEM	-	Отображение внутренних регистров процессорного ядра. Подробнее см. таблицу 16

6.5.2 Распределение областей памяти для регистров периферии

Таблица 15 – Распределение адресного пространства периферии

	Базовый адрес	Обозначение	Доступ по шинам	Описание
0	0x4000_0000	MDR_CLK	S, M	Контроллер тактовых частот
1	0x4000_1000	MDR_BKP		Контроллер батарейного домена и часов реального времени
2	0x4000_2000	MDR_PWR		Контроллер блока управления питанием
3	0x4000_3000	MDR_FTMODE		Контроллер блока диагностики ошибок
4	0x4000_4000	MDR_WDG		Контроллер сторожевого таймера
5	0x4000_5000	MDR_EBC		Контроллер внешней шины
6	0x4000_6000	MDR_OTP		Контроллер памяти OTP
7	0x4000_7000	-		Зарезервировано
8	0x4000_8000	MDR_RAMD		Контроллер ОЗУ в области DATA
9	0x4000_9000	MDR_ROM		Контроллер ПЗУ
10	0x4000_A000	-		Зарезервировано
11	0x4000_B000	MDR_RAMDCACHE		Контроллер кэш памяти RAMD
12	0x4000_C000	-		Зарезервировано
13	0x4008_0000	MDR_PORTA		Контроллер порта A
14	0x4008_1000	MDR_PORTB		Контроллер порта B
15	0x4008_2000	MDR_PORTC		Контроллер порта C
16	0x4008_3000	MDR_PORTD		Контроллер порта D
17	0x4008_4000	MDR_PORTE		Контроллер порта E
18	0x4008_5000	-		Зарезервировано
19	0x4008_6000	-		Зарезервировано
20	0x4008_7000	-		Зарезервировано
21	0x4008_8000	MDR_SPW0		Контроллер SpaceWire0
22	0x4008_9000	-		Зарезервировано
23	0x4008_A000	MDR_TMR0		Контроллер Таймера 0
24	0x4008_B000	MDR_TMR1		Контроллер Таймера 1
25	0x4008_C000	MDR_TMR2		Контроллер Таймера 2
26	0x4008_D000	MDR_TMR3		Контроллер Таймера 3
27	0x4008_E000	-		Зарезервировано
28	0x4008_F000	-		Зарезервировано
29	0x4009_0000	MDR_CAN0		Контроллер CAN0
30	0x4009_1000	-		Зарезервировано
31	0x4009_2000	-		Зарезервировано
32	0x4009_3000	-		Зарезервировано
33	0x4009_4000	-		Зарезервировано
34	0x4009_5000	MDR_SSP0		Контроллер SSP0
35	0x4009_6000	-		Зарезервировано
36	0x4009_7000	-		Зарезервировано
37	0x4009_8000	-		Зарезервировано

	Базовый адрес	Обозначение	Доступ по шинам	Описание
38	0x4009_9000	MDR_UART0	S, M	Контроллер UART0
39	0x4009_A000	MDR_UART1		Контроллер UART1
40	0x4009_B000	-		Зарезервировано
41	0x4009_C000	-		Зарезервировано
42	0x4009_D000	-		Зарезервировано
43	0x4009_E000	MDR_ARCRX0		Контроллер ARINC0 RX
44	0x4009_F000	MDR_ARCRX0		DATA
45	0x400A_0000	MDR_ARCTX0		Контроллер ARINC0 TX
46	0x400A_1000	MDR_ARCTX0		DATA
47	0x400A_2000	-		Зарезервировано
48	0x400A_3000	-		Зарезервировано
49	0x400A_4000	-		Зарезервировано
50	0x400A_5000	-		Зарезервировано
51	0x400A_6000	MDR_MILDAT0		Контроллер MIL0 данные
52	0x400A_7000	MDR_MILCNR0		Контроллер MIL0 управление
53	0x400A_8000	MDR_MILDAT1		Контроллер MIL1 данные
54	0x400A_9000	MDR_MILCNR1		Контроллер MIL1 управление
55	0x400A_A000	MDR_ADC0		Контроллер ADC0
56	0x400A_B000	MDR_ADC1		Контроллер ADC1
57	0x400A_C000	MDR_DAC0		Контроллер DAC0
58	0x400A_D000	MDR_DAC1		Контроллер DAC1
59	0x400A_E000	-		Зарезервировано
60	0x400A_F000	-		Зарезервировано
61	0x400B_0000	-		Зарезервировано
62	0x400B_1000	-		Зарезервировано
63	0x400B_2000	-		Зарезервировано

6.5.3 Распределение областей памяти для регистров периферии ядра

Таблица 16 – Распределение внутреннего адресного пространства ядра

Базовый адрес	Размер, байт	Обозначение	Описание
0xE000_E008	-	System Control Block	Системные регистры
0xE000_E010		System Timer	Регистры системного таймера
0xE000_E100		NVIC	Регистры контроллера прерываний
0xE000_ED00		System Control Block	Системные регистры
0xE000_ED90		MPU	Регистры блока защиты памяти
0xE000_EF00		NVIC	Регистры контроллера прерываний
0xE000_EF30		FPU	Регистры блока вычислений с плавающей запятой
0xE004_0000		TPIU	Блок управления портом трассировки TPIU
0xE004_1000		ETM	Блок формирования трассировки ядра ETM
0xE004_2000		MDR_DMA	Блок прямого доступа в память

Базовый адрес	Размер, байт	Обозначение	Описание
0xE004_3000		-	Зарезервировано
0xE004_4000		MDR_ICACHE	Блок управления I-кэш памятью
0xE004_5000		MDR_DCACHE	Блок управления D-кэш памятью
0xE00F_F000		ROM TABLE	Таблица описания периферии ядра

Описание блоков, не представленных в данном документе, можно найти в Cortex-M4 Technical Reference Manual.

6.6 Помехозащищенное кодирование

В микросхеме применяется помехозащищенное кодирование SEC-DED (исправление одиночных и обнаружение двойных ошибок):

– кодом Хэмминга (72,64) – для внутренних памятей и памяти на внешней системной шине;

– кодом Хэмминга (7,4) – для задания режима работы.

Код Хэмминга (7,4) применяется для задания режима работы микросхемы, подробнее см. подраздел 6.4 «Режимы работы микросхемы».

Кодом Хэмминга (72,64) всегда защищаются внутренние памяти, включая кэш-память. Кодирование ECC может использоваться и для защиты внешней памяти. С помощью ECC кодируются как непосредственно сами данные, так и адрес расположения данных. В результате автоматически исправляются одиночные ошибки в шине данных, обнаруживаются двойные ошибки в поле данных, а также одиночные или двойные ошибки в поле адреса. Одиночные ошибки в поле адреса не исправляются и считаются неустранимыми ошибками. Кодирование производится при операциях записи в блоках ECC_T. На основании записываемых данных и адреса вычисляется поле HWECC[7:0] и передается совместно с полем данных (далее по тексту генерация ECC). При считывании на основании данных, адреса и поля HRECC[7:0] в блоке ECC_R проверяется полученная информация, и при необходимости исправляется одиночная ошибка в поле данных, либо обнаруживаются одиночные и двойные ошибки в поле адреса и двойные в поле данных (далее по тексту проверка ECC).

6.6.1 Описание работы кода Хэмминга HSIA70a

Алгоритм исправления единичных ошибок и обнаружения многократных ошибок посредством использования вектора синдрома S {S7, ..., S0} следующий:

1 Если вектор синдрома S {S7, ..., S0} равен 0, то кодовое слово не содержит ошибок.

2 Если вектор синдрома S {S7, ..., S0} отличен от нуля, и при этом совпадает с одним из столбцов проверочной матрицы (H-матрица кода ECC HSIA70a, таблица 17), то диагностируется и исправляется одиночная ошибка в разряде, соответствующем номеру столбца, если ошибка зарегистрирована в поле данных (столбцы с 0 по 31), и диагностируется, но не исправляется и вызывает Hardfault, если обнаружена в поле

адреса. При этом увеличивается счетчик единичных ошибок SEEC_CNT, регистр ECCCS[31...16].

3 Если вектор синдрома S {S7, ..., S0} имеет отличные от нуля компоненты, и при этом вес вектора синдрома четный (сумма всех компонентов вектора по модулю 2 или XOR равна 0), то диагностируется двойная ошибка, которая вызывает Hardfault или Busfault. При этом увеличивается счетчик двойных ошибок DEEC_CNT, регистр ECCCS[15...8].

4 Если вектор синдрома S {S7, ..., S0} имеет отличные от нуля компоненты, и при этом вес вектора синдрома нечетный (сумма всех компонентов вектора по модулю 2 или XOR равна 1) и значение этого вектора не соответствует ни одному из столбцов проверочной матрицы (H-матрица кода ECC HSIA70a, таблица 17), то это означает, что были обнаружены нечетные ошибки порядка, большего или равного 3. Детектирование таких многократных ошибок носит вероятностный характер, но при этом также вызывает увеличение счетчика одиночных ошибок SEEC_CNT, регистр ECCCS[31...16].

6.6.2 Кодирование ECC для ПЗУ

При обращении в область загрузочного ПЗУ памяти проверка и, при необходимости, исправление по ECC происходит на границе контроллера памяти ПЗУ и непосредственно самой матрицы ПЗУ. После чего, корректные данные с контрольной суммой передаются процессорному ядру, где они еще раз проверяются по ECC, но уже без исправления, только с целью обеспечения достоверности.

Биты ECC загрузочной программы вычислены на этапе создания программы и зашиты в ПЗУ микросхемына этапе разработки.

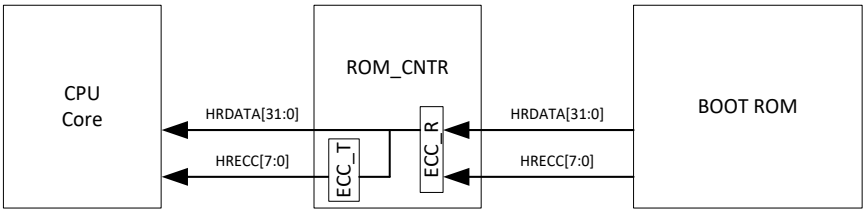


Рисунок 13 – Схема контроля ECC на пути от ПЗУ

6.6.3 Кодирование ECC для памяти OTP программ

При обращении в область однократно программируемой памяти OTP памяти программ проверка ECC происходит на границе контроллера OTP и непосредственно самой матрицы памяти. При этом из матрицы извлекается строка с размером в общей сложности 160 бит (128 бит данных и 32 бита ECC). Для всех слов строки проводится проверка и при необходимости исправление данных по ECC. Затем данные передаются в кэш-данные (DCACHE) или кэш-инструкции (ICACHE) и сохраняются в них с ECC. Одновременно данные передаются в процессорное ядро. Возвращаемые процессорному ядру данные снабжены суммой ECC, которая используется для обеспечения достоверности передаваемых данных.

Генерация бит ECC при программировании памяти OTP осуществляется программными средствами разработки программ и вычисляется при программировании памяти.

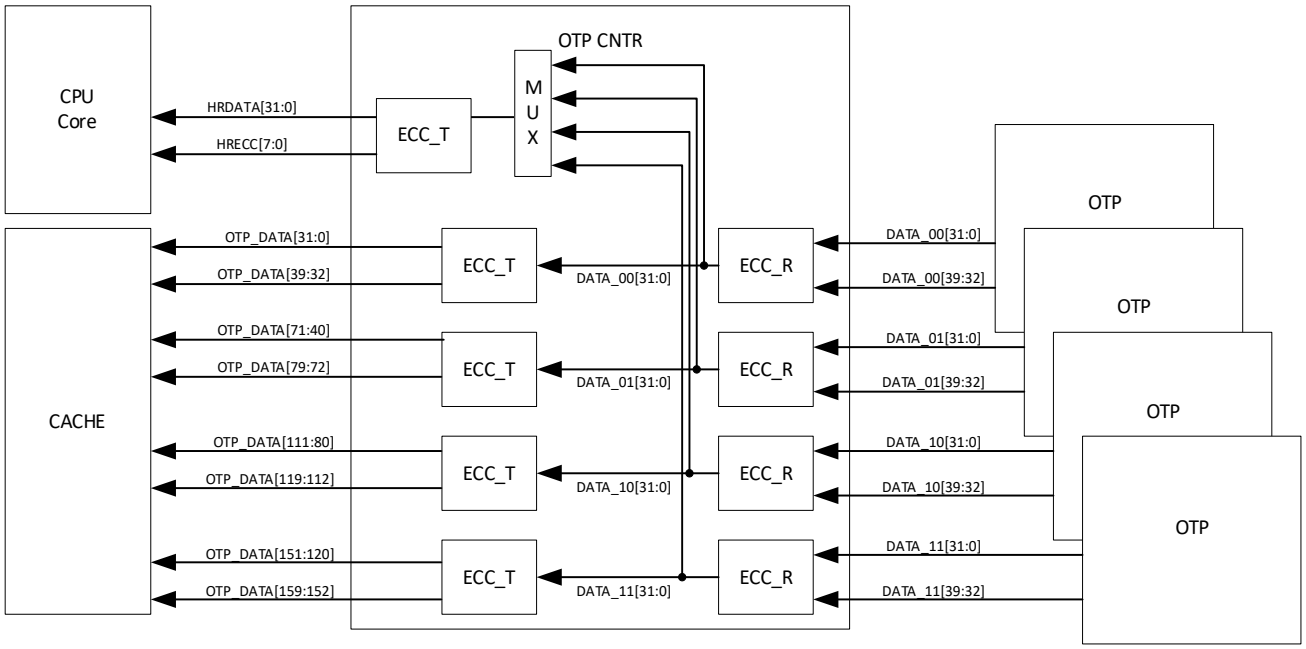


Рисунок 14 – Схема контроля ECC на пути от памяти OTP программ

6.6.4 Кодирование ECC для периферии

Периферийные блоки, имеющие в своем составе большие массивы памяти, защищены ECC. Периферийные блоки также могут иметь собственные механизмы исправления и обнаружения ошибок.

6.6.5 H-матрица ECC по коду Хэмминга (72, 64)

Таблица 17 – H-матрица кода ECC HSIA70a

	Проверочные биты								Байты данных																															
									0								1								2								3							
Bits	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
S0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0
S1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1
S2	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0
S3	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1
S4	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0
S5	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0
S6	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1
S7	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

	Проверочные биты								Байты данных																															
									4								5								6								7							
Bits	0	1	2	3	4	5	6	7	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
S0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0
S1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0
S2	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1
S3	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
S4	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0
S5	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1
S6	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0
S7	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1

6.6.6 H-матрица ECC по коду Хэмминга (7, 4) с битом четности

Таблица 18 – Матрица для кода Хэмминга (8, 4) (расширение кода Хэмминга (7,4) битом четности)

MODE[ ]	7	6	5	4	3	2	1	0
S1	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**
S2	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**
S3	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**
S0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

S0 – бит четности. Соответствующий ему MODE[4] вырезается при сохранении в поле MODE[6:0] регистров BKP REG_60_TMRx.

6.6.7 Функция вычисления ECC для записи в память

Вычисление восьмибитной суммы ECC для 32-битного слова в памяти осуществляется на основании самого слова (unsigned int DATA) и адреса (unsigned int ADR) расположения слова.

```
SECC = getecc ((((unsigned long long)ADR)<<32) | DATA);
```

Тело функции getecc:

```
unsigned int getecc(unsigned long long inw)
{
unsigned char ecc = 0;
unsigned long long  inputw;
unsigned long long H[8];
int i,j;
unsigned char res = 0;

H[0] = (unsigned long long) 0x0738C808099264FF;
H[1] = (unsigned long long) 0x38C808099264FF07;
H[2] = (unsigned long long) 0xC808099264FF0738;
H[3] = (unsigned long long) 0x08099264FF0738C8;
H[4] = (unsigned long long) 0x099264FF0738C808;
H[5] = (unsigned long long) 0x9264FF0738C80809;
H[6] = (unsigned long long) 0x64FF0738C8080992;
H[7] = (unsigned long long) 0xFF0738C808099264;

for (i=0; i < 8; i++)
        {
        inputw = H[i] & inw;
        res = 0;
        for (j=0; j < 64; j++)
                {
                res = res ^ ((inputw>>j)&0x0000000000000001);
                }
        ecc = ecc | res << i;
        }
return ecc;
}
```

6.7 Система команд процессорного ядра

Инструкция	Выполняемое действие	Изменяемые флаги	Примечание
Вызов и возврат из функций			
BL label	PC← адрес label; LR← адрес возврата;	-	Используется для вызова функций
BX Rn	PC← Rn;	-	Используется для перехода по адресу из регистра Rn. Может использоваться для возврата из функции если Rn = LR
Загрузка констант			
ADR Rd,lable	Rd← адрес label;	-	
LDR Rd,=constant	Rd← constant;	-	Псевдоинструкция, аналогична MOV, MVN
MOV Rd, constant	Rd← constant;	NZC	0 ≤ constat ≤ 255
MVN Rd, constant	Rd← ~ constant;	NZC	0 ≤ constat ≤ 255
LDRB Rd,[address mode]	Rd← ;		

6.8 Интерфейс отладки

Для отладки приложений используется интерфейс JTAG/SWD.

При выборе отладочного интерфейса выводы JTAG_A или JTAG_B, мультиплексированные с выводами отладочного интерфейса, переопределяются в выводы JTAG не зависимо от выполняемых ими функций.

Таблица 19 – Сигналы интерфейсов отладки

Сигнал JTAG	Сигналы SWD	JTAG_A=1	JTAG_B=1
nTRST	-	PA[0]	PB[4]
TCK	SWCLK	PA[1]	PB[5]
TMS	SWDIO	PA[2]	PB[6]
TDI	-	PA[3]	PB[7]
TDO	SWV	PA[4]	PB[8]

6.9 Тестовый режим работы

Микросхема переводится в тестовый режим:

– при установке высокого уровня на выводе JTAG_EN;

– при определении загрузочной программой режима MODE[7:0] = 1111 1111;

– при установке поля MODE[3:0] = 1111 в регистрах REG_60_TMRx контроллера батарейного домена.

При переходе в тестовый режим работа микросхемы в функциональном режиме останавливается.

В тестовом режиме к выводам JTAG_A подключается тестовый контроллер TAP. Регистр инструкций (IR) тестового контроллера TAP имеет длину пять бит. Загрузка инструкций в регистр IR переводит микросхему в тестовые режимы, указанные в таблице 20.

Таблица 20 – Инструкция тестового контроллера TAP

IR[4:0]	TYPE	TTMODE	Название инструкции	Описание
0x14			IDCODE	В качестве регистра данных выбирается регистр ID, который имеет значение 0xAAA00FDF
0x1C			EXTEST	Переводит микросхему в режим внешнего граничного тестирования. В качестве регистра данных выбирается регистр граничного сканирования
0x1D			SAMPLE	Позволяет считать регистр граничного сканирования, который выбирается в качестве регистра данных
0x1E			PRELOAD	Позволяет загрузить регистр граничного сканирования, который выбирается в качестве регистра данных
0x1F			BYPASS	В качестве регистра данных выбирается 1-битный регистр. Используется для обеспечения последовательного пути минимальной длины между выводами TDI и TDO
0x13	SCAN	0	MAIN_ATPG	Режим тестирования основной логики МК
0x02		1	ADC0_OPEN	Режим раскрытия ADC0
0x03		2	BKP_OPEN	Режим раскрытия BKP домена и тестирования LSE, LSI, WDG, BKP
0x04		3	HSE_OPEN	Режим раскрытия HSE, PLL
0x05		4	ADC1_OPEN	Режим раскрытия ADC1
0x06		5	DAC_OPEN	Режим раскрытия DAC
0x07	MEM	6	OTP_OPEN	Режим раскрытия OTP
0x09	MEM	8	RAMD_OPEN	Режим раскрытия RAMD
0x0A	MEM	9	ROM_OPEN	Режим раскрытия ROM
0x0B		10	ETH_OPEN	Режим раскрытия ETH PHY
0x0C		11	SPW_OPEN	Режим раскрытия SPW
0x0D	MEM	12	ETH_MEM_OPEN	Режим раскрытия памяти ETH
0x0E	MEM	13	ARC_MEM_OPEN	Режим раскрытия памяти ARC
0x0F	MEM	14	MIL_MEM_OPEN	Режим раскрытия памяти MIL
0x01		16	LDO_OPEN	Режим раскрытия LDO
0x17	MEM	17	CAN_MEM_OPEN	Режим раскрытия памяти CAN

6.9.1 Режим тестирования ОЗУ

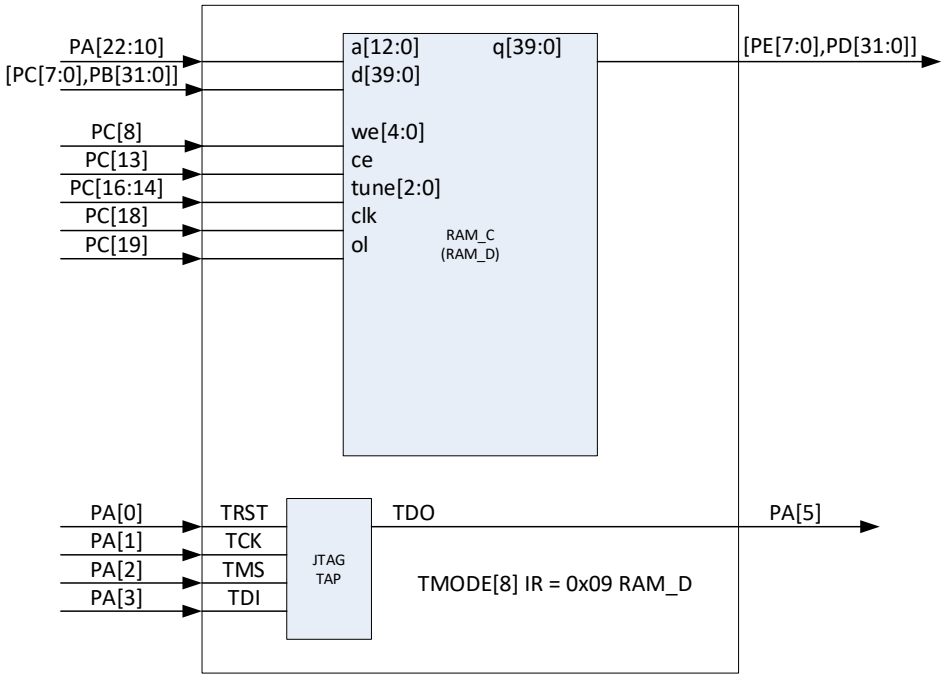


Рисунок 15 – Схема режима раскрытия блоков памяти данных RAM_D

6.9.2 Режим тестирования Масочного ПЗУ (BOOTROM)

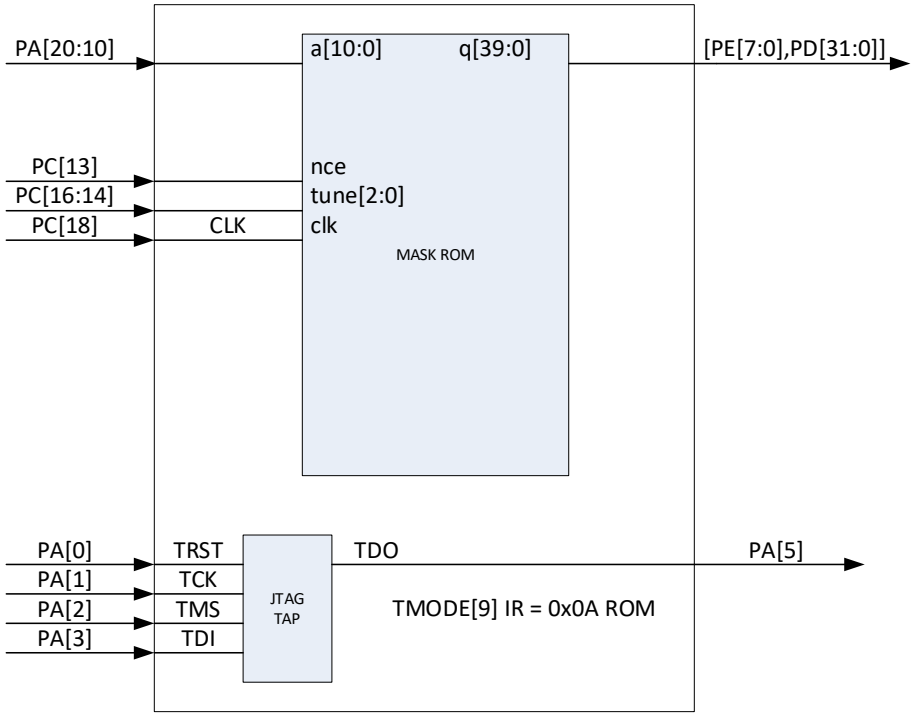


Рисунок 16 – Схема режима раскрытия блоков памяти ROM

6.9.3 Режим тестирования OTP

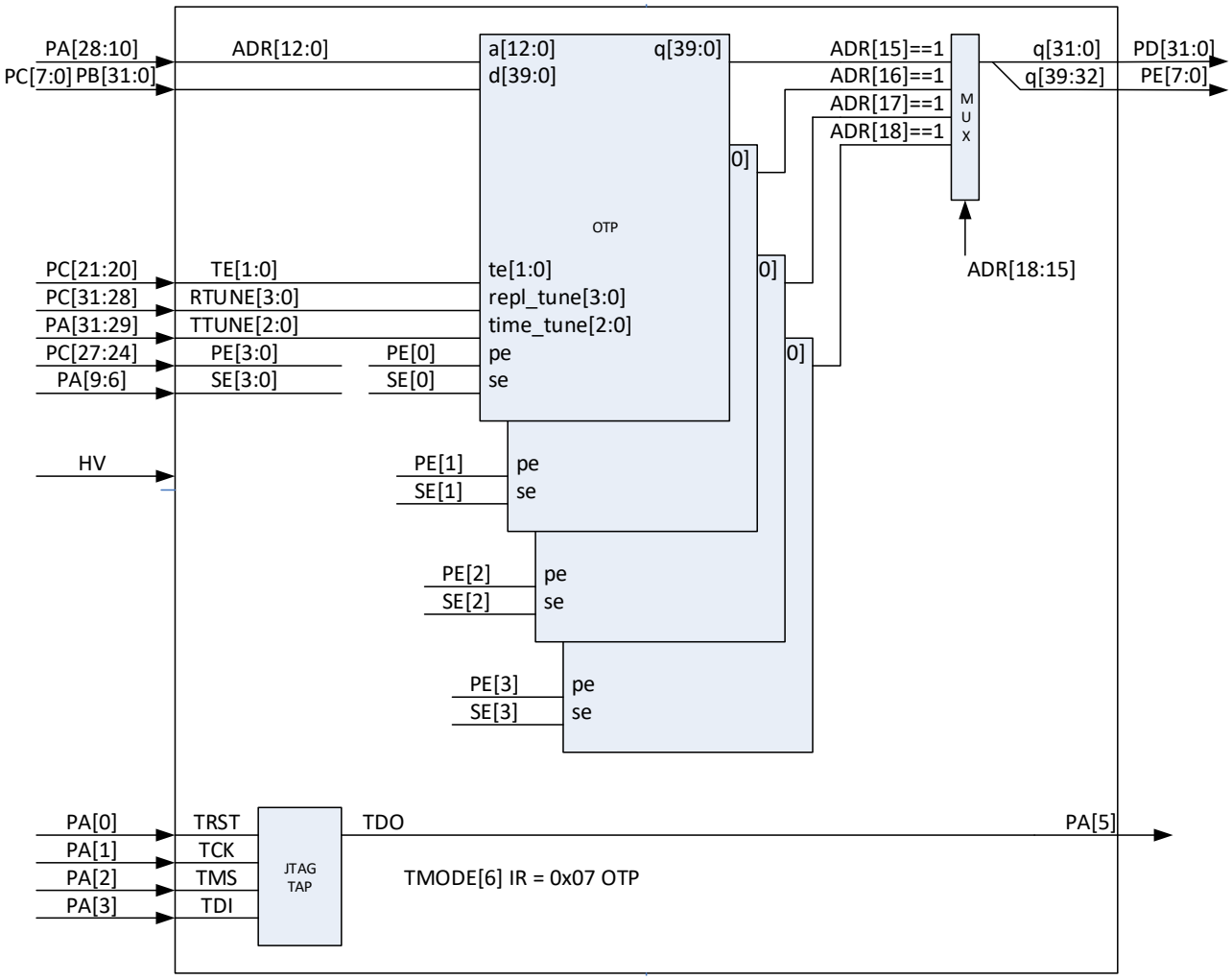


Рисунок 17 – Схема режима раскрытия блоков памяти OTP программ

6.9.4 Режим тестирования CAN_RAM

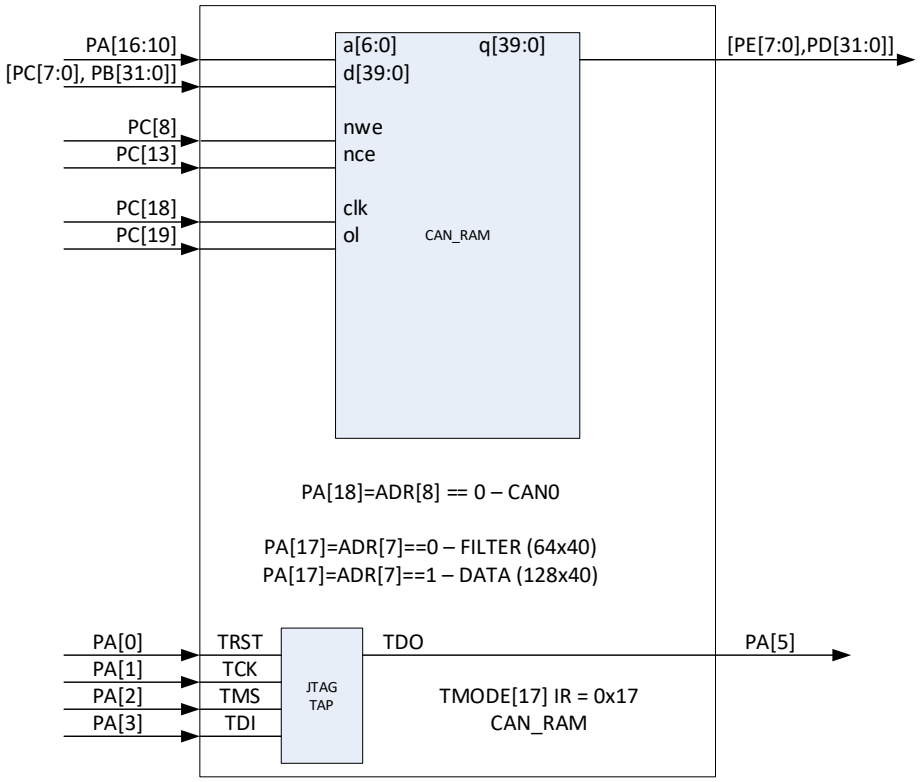


Рисунок 18 – Схема режима раскрытия блока памяти CAN

6.9.5 Режим тестирования ETH_RAM

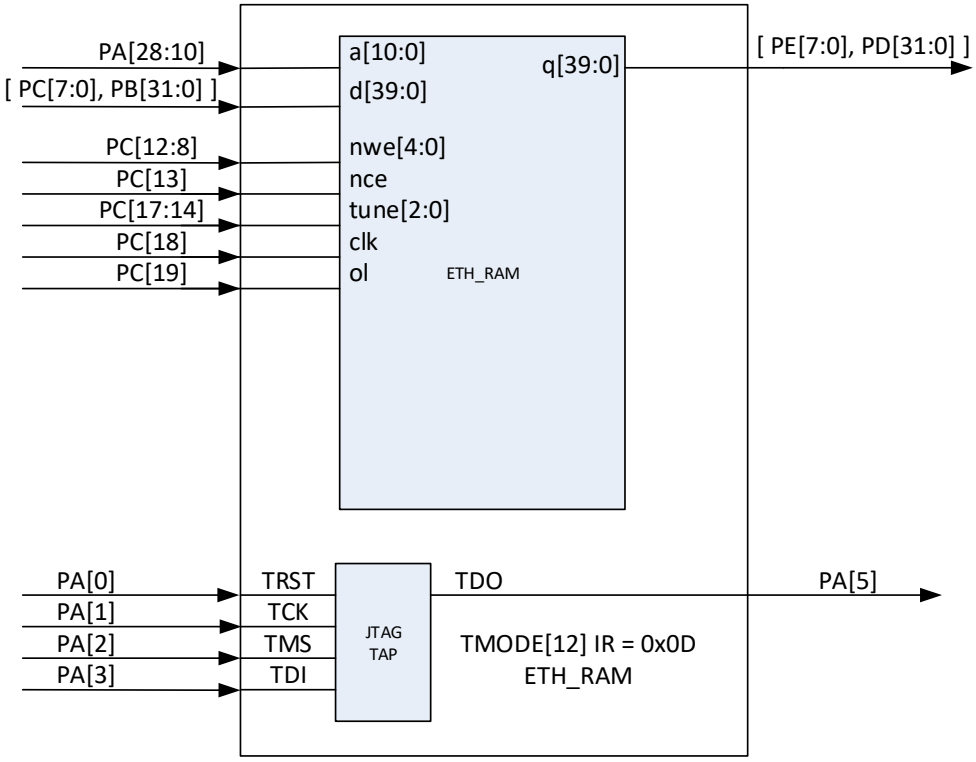


Рисунок 19 – Схема режима раскрытия блоков памяти Ethernet

6.9.6 Режим тестирования MIL_RAM

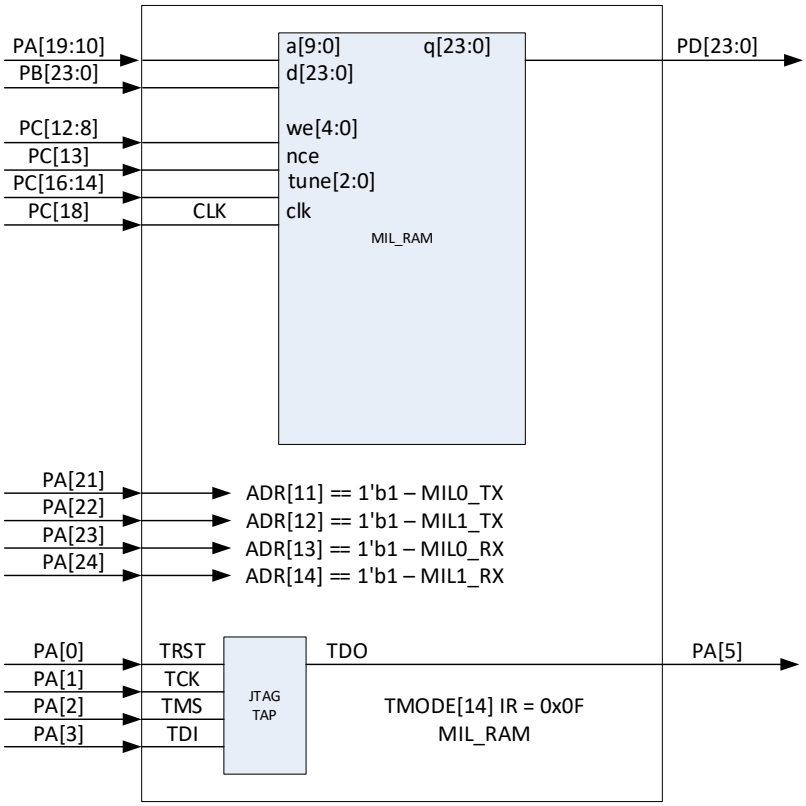


Рисунок 20 – Схема режима тестового раскрытия блоков памяти MIL

6.9.7 Режим тестирования ARC_RAM

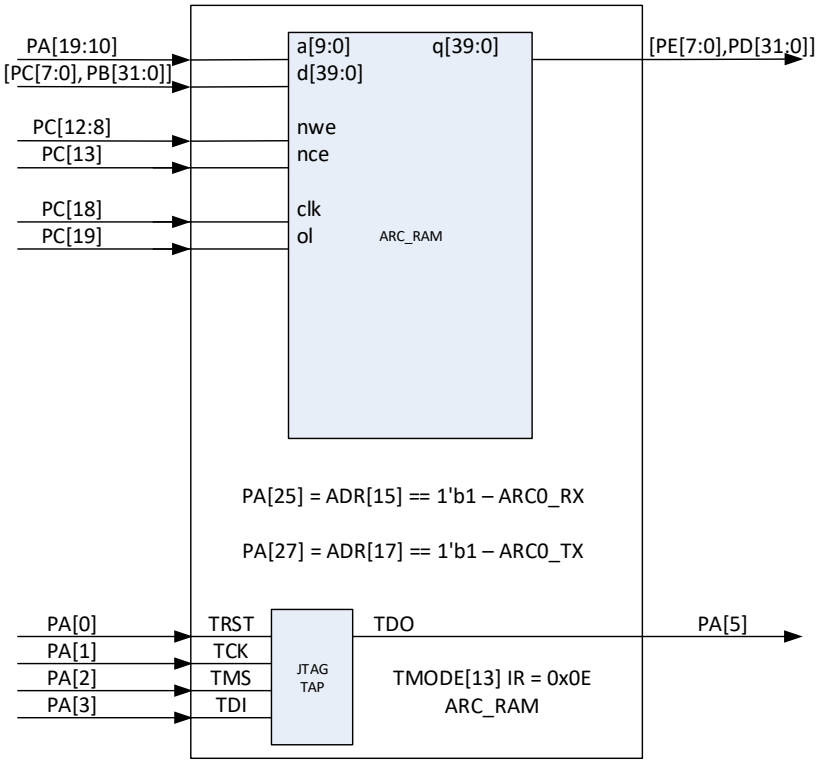


Рисунок 21 – Схема режима раскрытия блоков памяти ARINC

6.9.8 Режим тестирования генераторов и PLL

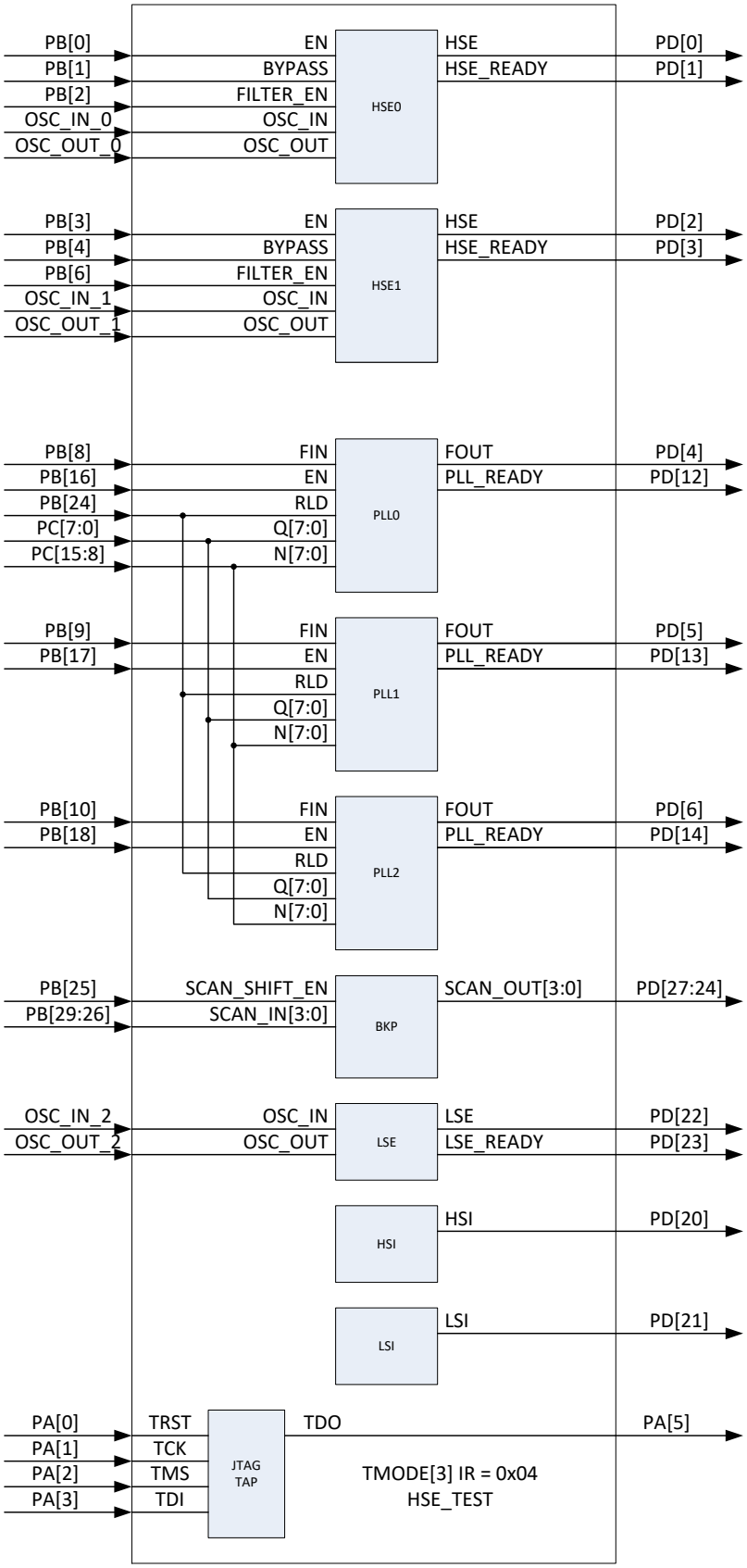


Рисунок 22 – Схема режима раскрытия блоков PLL и генераторов

6.9.9 Режим тестирования АЦП

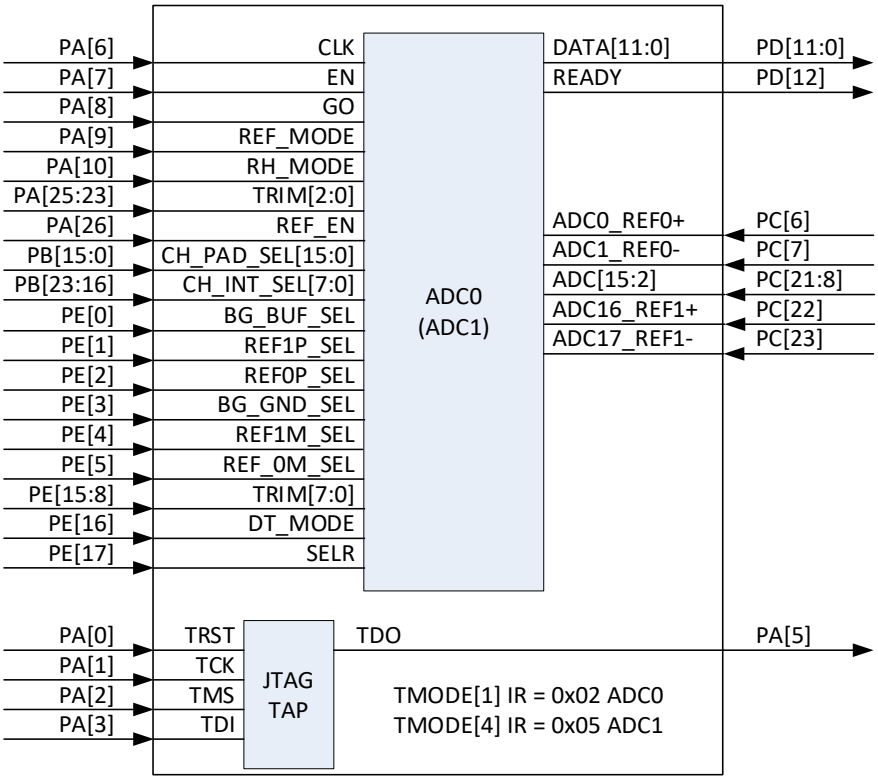


Рисунок 23 – Схема режима раскрытия блоков АЦП

6.9.10 Режим тестирования ЦАП

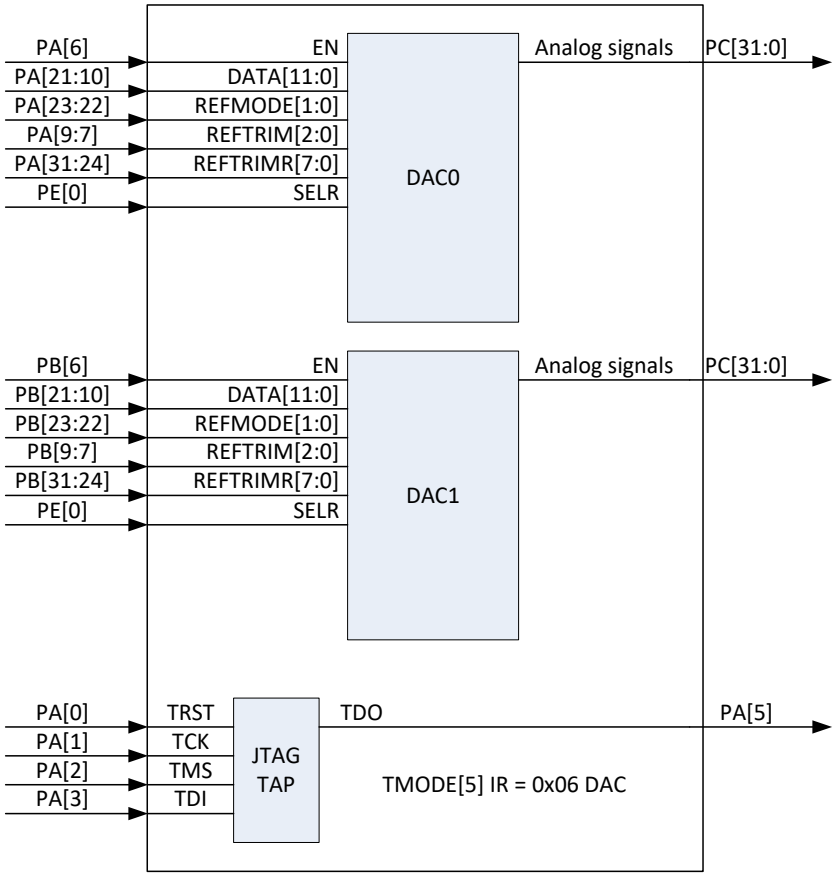


Рисунок 24 – Схема режима раскрытия блоков ЦАП

6.9.11 Режим тестирования приемопередатчиков SPW

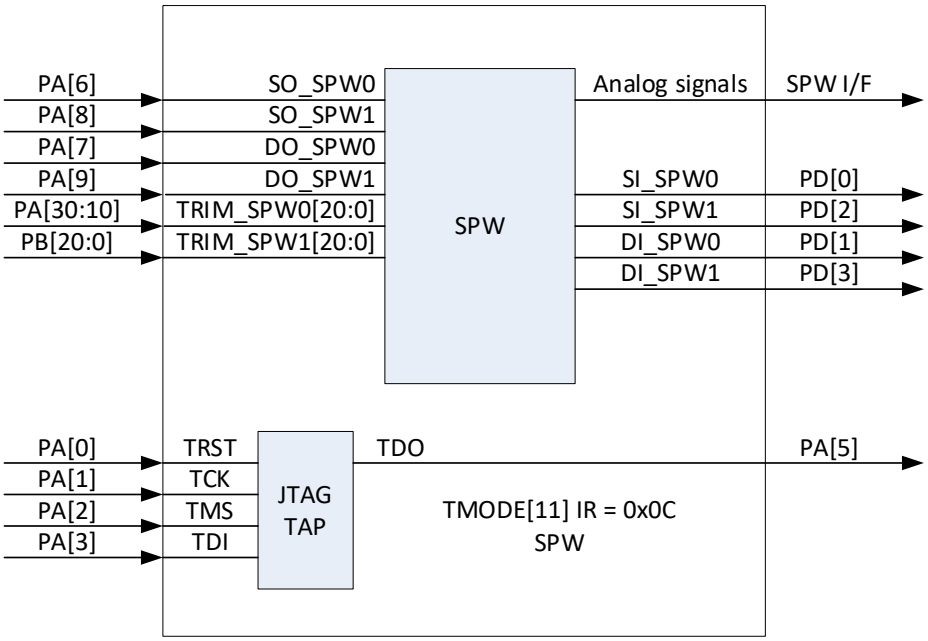


Рисунок 25 – Схема режима раскрытия блоков передатчиков SpaceWire

6.9.12 Режим тестирования ETHPHY

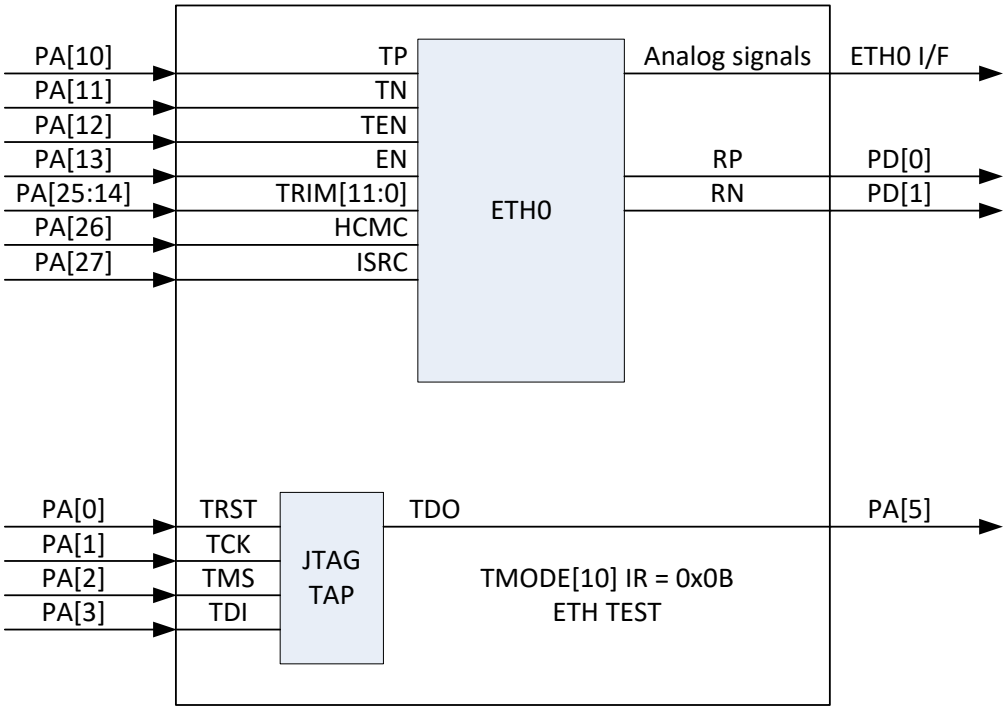


Рисунок 26 – Схема режима тестового раскрытия блоков передатчиков Ethernet

6.9.13 Режим тестирования PMU (VoltageDetect)

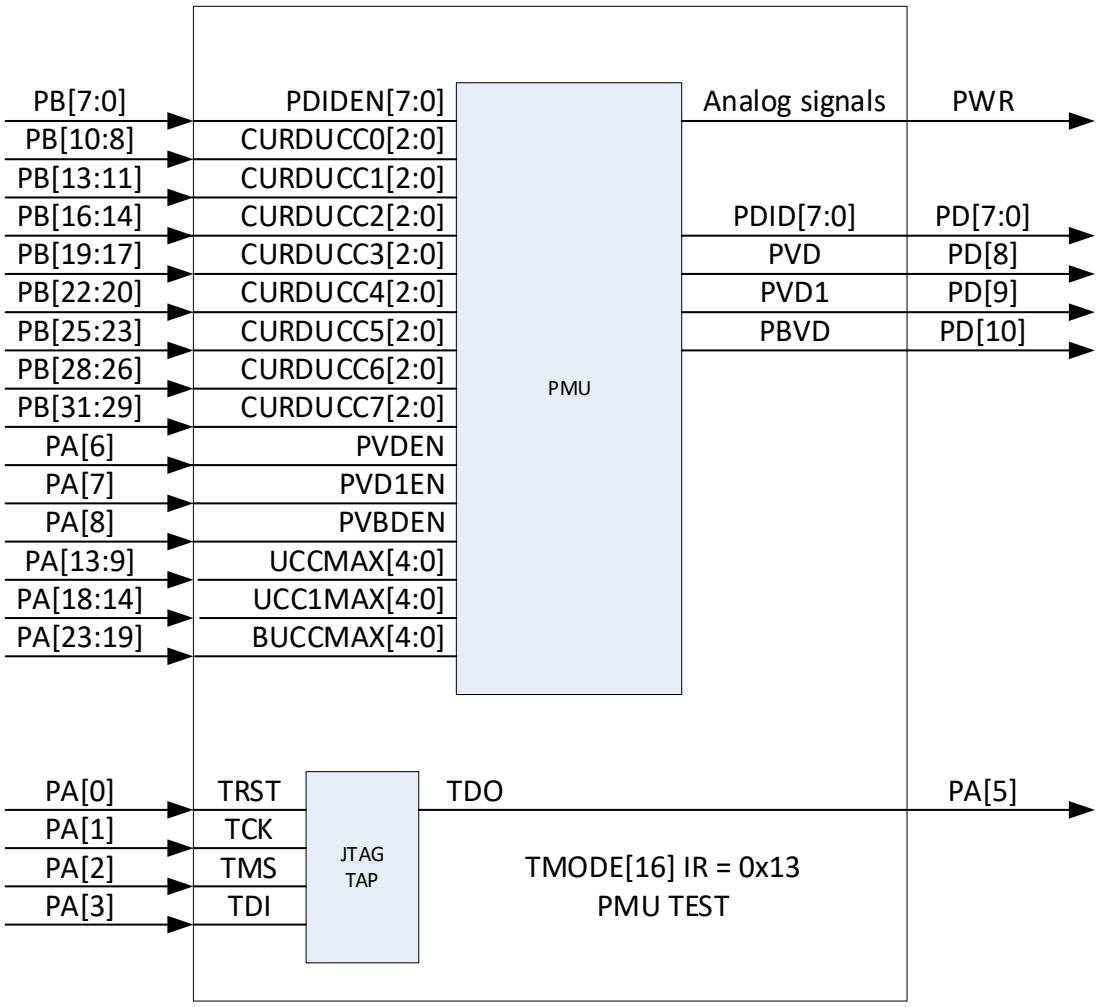


Рисунок 27 – Схема режима тестового раскрытия блок управления питанием

6.9.14 Режим тестирования ВКР

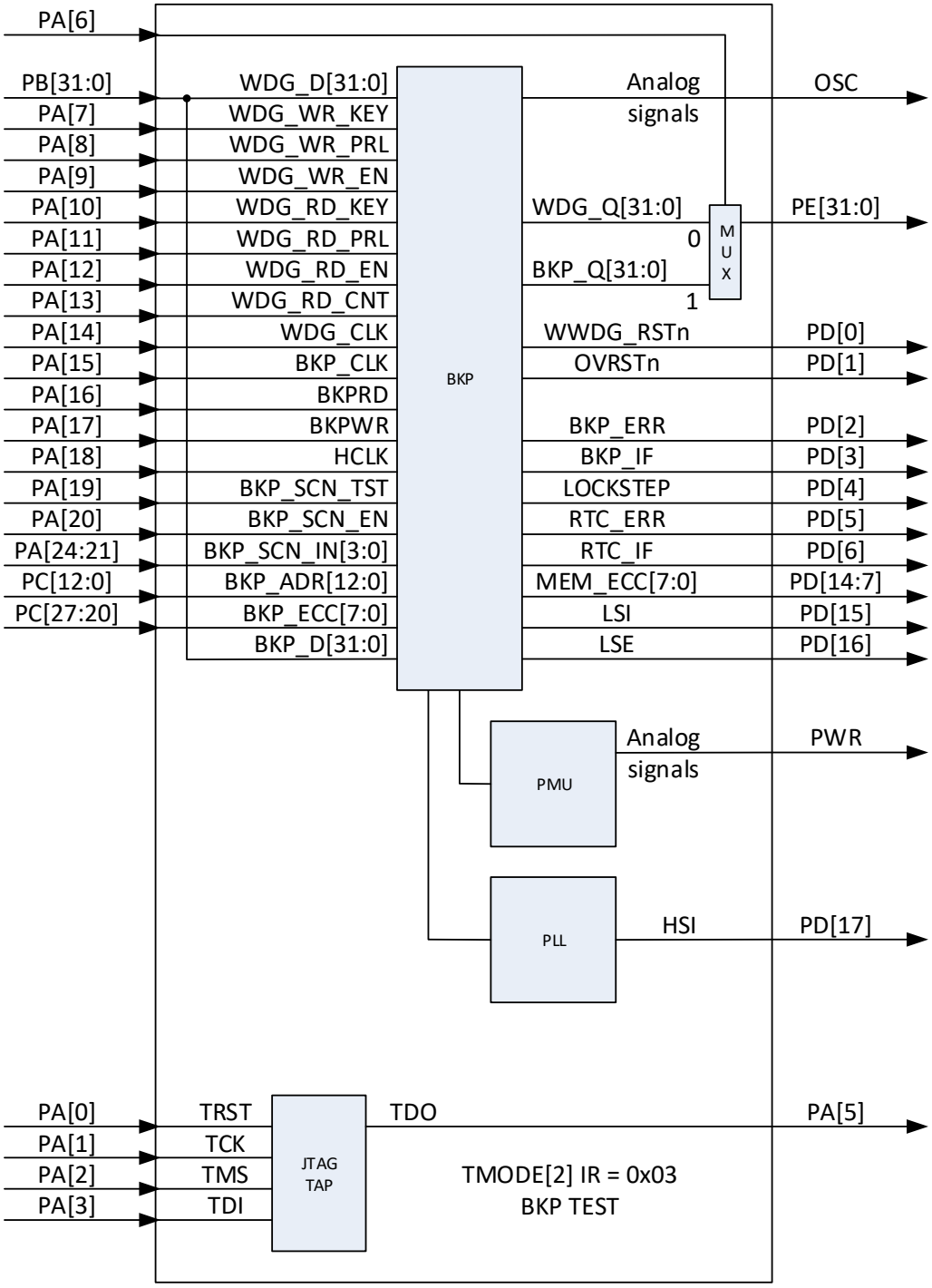


Рисунок 28 – Схема режима тестового раскрытия блока батарейного домена

7 Программная модель микросхемы

В данном разделе приводятся принципы программирования и описание регистров микросхемы.

7.1 Идентификация микросхемы

Для идентификации микросхем в последние 16 слов памяти OTP записана идентификационная информация.

Ячейки памяти с адресом 0x0100_0000 и смещениями с 0x0001_FFC0 по 0x0001_FFFF зарезервированы и не могут использоваться для хранения пользовательской программы. Эти регистры содержат идентификационную информацию.

7.1.1 Описание полей идентификации

Таблица 21 – Описание регистров

Базовый адрес	Смещение	Название	Описание
0x0100_0000	0x0001_FFC0 0x0001_FFC4 0x0001_FFC8 0x0001_FFCC	WAFER_INFO	Номер партии пластин, номер пластины в партии
	0x0001_FFD0	PACKAGE_INFO	Технологический номер партии микросхем, номер микросхемы
	0x0001_FFD4 0x0001_FFD8 0x0001_FFDC 0x0001_FFE0 0x0001_FFE4 0x0001_FFE8 0x0001_FFEC 0x0001_FFF0 0x0001_FFF4 0x0001_FFF8 0x0001_FFFC	–	Резерв, может содержать произвольную информацию, но корректную по ECC

WAFER_INFO (4 слова = 16 байт) состоит из WAFER_LOT_ID (15 байт) и WAFER_ID (1 байт). Запись в эти поля производится на этапе измерения пластины:

- WAFER_LOT_ID – номер партии пластин, 15 символов в ASCII кодировке;
- WAFER_ID – номер пластины в партии пластин, натуральное число от 1 до 25.

PACKAGE_INFO (1 слово = 4 байта) состоит из PACKAGE_LOT_ID (2 байта) и PACKAGE_ID (2 байта). Запись в эти поля производится на этапе измерения микросхемы в корпусе:

- PACKAGE_LOT_ID – технологический номер партии микросхем, натуральное число от 0 до 99;

– PACKAGE_ID – номер микросхемы в партии микросхем, натуральное число от 0 до 4000.

Таблица 22 – Описание полей идентификации

Обозначение поля	WAFER_INFO																PACKAGE_INFO			
	WAFER_LOT_ID															WAFER_ID	PACKAGE_LOT_ID		PACKAGE_ID	
Адрес	0x0101_FFC0	0x0101_FFC1	0x0101_FFC2	0x0101_FFC3	0x0101_FFC4	0x0101_FFC5	0x0101_FFC6	0x0101_FFC7	0x0101_FFC8	0x0101_FFC9	0x0101_FFCA	0x0101_FFCB	0x0101_FFCC	0x0101_FFCD	0x0101_FFCE	0x0101_FFCF	0x0101_FFD0	0x0101_FFD1	0x0101_FFD2	0x0101_FFD3
Кодировка	ASCII String[15]															unsigned char	unsigned short		unsigned short	
Пример	«M12345»															0x13	0x0059		0x1234	

Расшифровка данных примера: микросхема из партии пластин «M12345», из пластины № 13, из производственной партии № 59, микросхема с номером № 1234.

Для уникальной идентификации микросхемы необходимо использовать все поля.

7.2 Контроллер тактовых частот (CLKCNTR)

7.2.1 Формирование и управление тактовыми сигналами

Блок управления тактовыми сигналами является важным элементом обеспечения работоспособности микросхемы, так как при отсутствии тактовых сигналов микросхема не может выполнять возложенные на него функции. Блок CLKCNTR обеспечивает управление всеми тактовыми сигналами микросхемы. Структурная схема формирования тактовых частот представлена на рисунке 29.

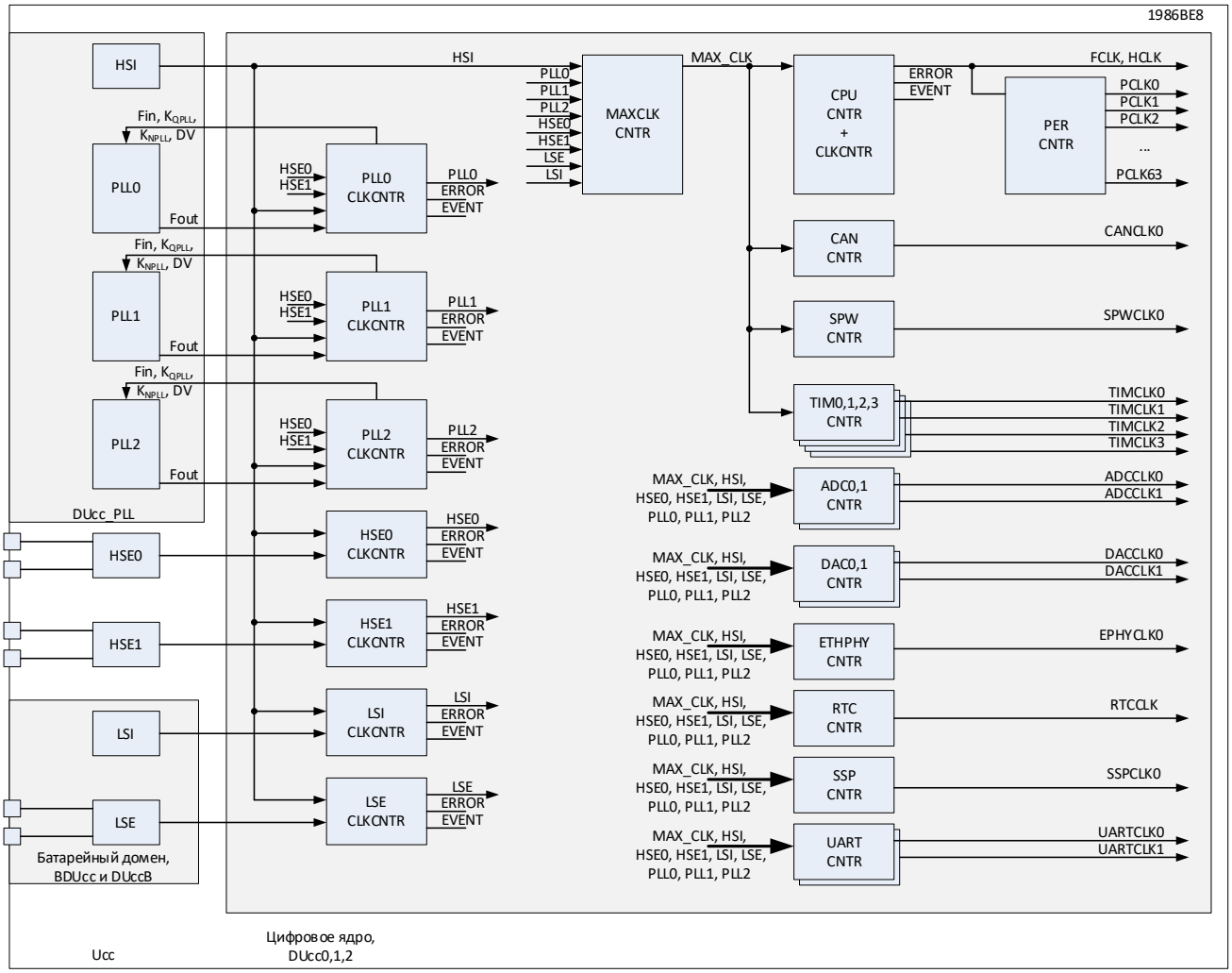


Рисунок 29 – Формирование тактовых частот

При выполнении загрузочной программы и при запуске пользовательской программы микросхема тактируется встроенным генератором HSI. Генератор HSI считается надежным и стабилизированным, отказ генератора HSI приводит к полному отказу микросхемы. Включение и переключение на другие источники тактирования, а также разрешение тактирования периферийных блоков осуществляется программно в пользовательской программе. Система тактовой синхронизации построена по принципу полной синхронности основных элементов, связанных с процессорными ядрами (блоки памяти ОЗУ, контроллер DMA, мосты, и интерфейсные по отношению к процессору части всех периферийных блоков). То есть эти части микросхемы всегда работают на одной тактовой частоте. Ряд периферийных блоков (CAN, TIMER, MIL, SpaceWire) могут быть синхронными процессорному ядру, но при этом могут работать как на большей, так и на меньшей тактовой частоте. В этом случае не требуется дополнительной логики по пересинхронизации. Ряд периферийных блоков (SSP, UART, ADC, DAC, RTC) могут работать на несинхронных частотах. Для обеспечения корректной работы всех синхронных модулей формируется частота MAXCLK. Далее синхронные блоки могут работать на частоте MAXCLK или на прореженной частоте MAXCLK. При этом процессорные ядра и связанные с ним блоки также могут работать на прореженной

частоте (все на одной и той же). Принцип формирования и соотношения тактовых частот в микросхеме приведен на рисунке 30.

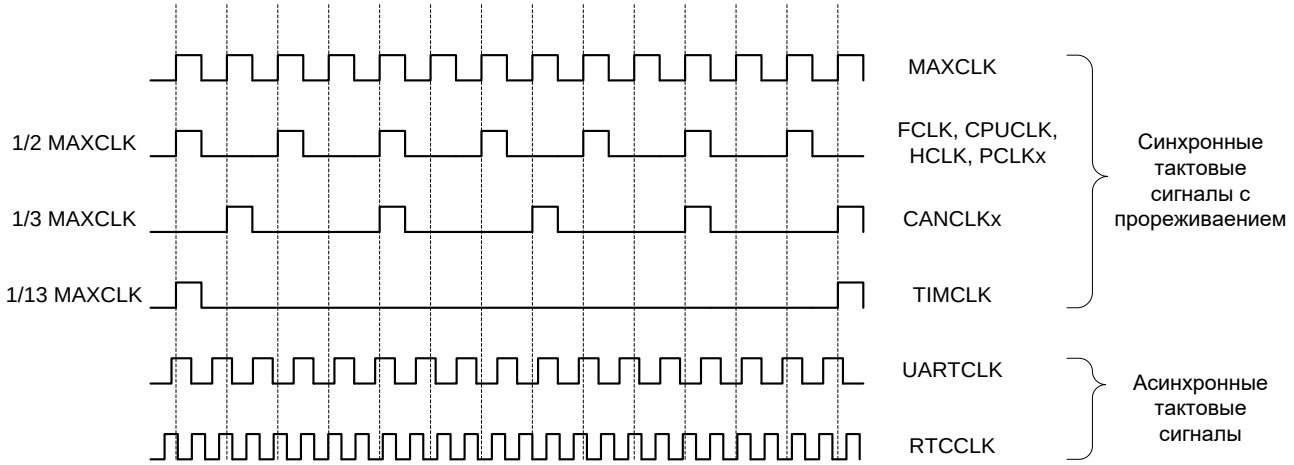


Рисунок 30 – Пример тактовых частот в микросхеме

В ряде периферийных блоков (SSP, UART) вводятся ограничения на соотношения рабочих частот этих блоков и частоты PCLK интерфейса с процессорным ядром. Сигнал CPUCLK может совпадать с MAXCLK, но не должен превышать 64 МГц.

7.2.2 Схема контроля тактовых частот

Генератор HSI может использоваться как аварийная замена других источников тактирования. Для этого частота HSI и проверяемая частота от другого источника подается на блок Checker, в котором они сравниваются, и, в зависимости от настроенных критериев, определяются четыре события:

- незначительное увеличение тактовой частоты;
- значительное увеличение тактовой частоты;
- незначительное снижение тактовой частоты;
- значительно снижение (исчезновение) тактовой частоты.

В зависимости от настроек в данных ситуациях может быть выполнено автоматическое переключение на генератор HSI.

Генератор HSI может быть программно отключен (не рекомендуется) для снижения энергопотребления. Отключение генератора HSI выполняется в регистрах управления батарейного домена. Отключение генератора HSI делает невозможным автоматический контроль наличия тактовой частоты от других источников. Включение генератора HSI может быть выполнено программно либо по сигналу сброса по питанию U_{CC}.

Структурная схема блока Checker представлена на рисунке 31.

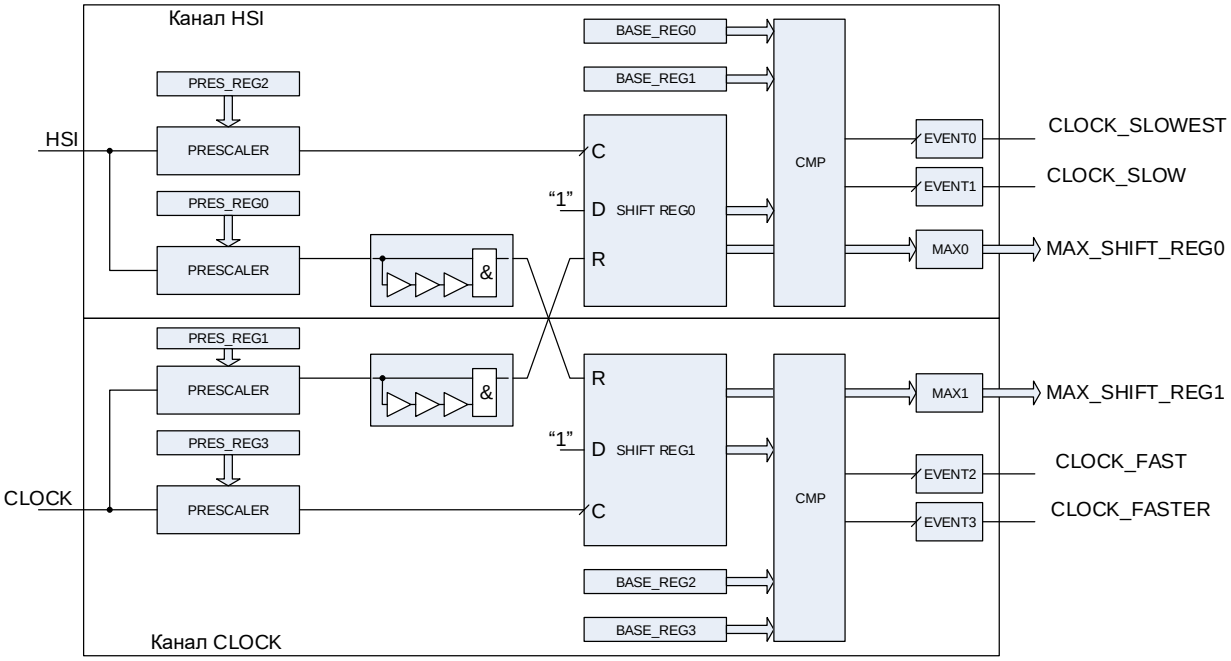


Рисунок 31 – Схема блока контроля тактовых частот

При настройке блока необходимо выбрать предделители сигналов HSI и CLOCK такими, чтобы соотношение полученных частот укладывалось в диапазон от 1 до 15.

$$\text{если } f_{HSI} < f_{CLOCK}, \begin{cases} \dfrac{\dfrac{f_{HSI}}{PRES_REG2+1}}{\dfrac{f_{CLOCK}}{PRES_REG1+1}} = \text{K} \\ \dfrac{\dfrac{f_{CLOCK}}{PRES_REG3+1}}{\dfrac{f_{HSI}}{PRES_REG0}} = \text{M} \end{cases}, \qquad (1)$$

$$\text{если } f_{HSI} > f_{CLOCK}, \begin{cases} \dfrac{\dfrac{f_{HSI}}{PRES_REG2+1}}{\dfrac{f_{CLOCK}}{PRES_REG1}} = K \\ \dfrac{\dfrac{f_{CLOCK}}{PRES_REG3+1}}{\dfrac{f_{HSI}}{PRES_REG0+1}} = \text{M} \end{cases}, \qquad (2)$$

где f_{HSI} – частота сигналов HSI;

f_{CLOCK} – частота сигналов CLOCK;

К ϵ *[1..16]* – фактическое соотношение частот, вычисляемое в блоке HSI. На схеме обозначено как SHIFT_REG0;

М ϵ *[1..16]* – фактическое соотношение частот, вычисляемое в блоке CLOCK. На схеме обозначено как SHIFT_REG1.

К0 – расчетное соотношение частот К в блоке HSI.

М0 – расчетное соотношение частот М в блоке CLOCK.

При отклонении частоты CLOCK от расчетного значения коэффициенты К и М отклоняются от своих расчетных значений К0 и М0.

При *снижении частоты CLOCK* значение К изменяется от К0 до 15. Максимальное значение К сохраняется в поле MAX_CHK_SHIFT_REG0 соответствующего регистра ..._STAT.

При достижении К пороговых значений, заданных полями BASE_REG0 и BASE_REG1, вырабатываются события EVENT0 и EVENT1 соответственно.

$$\begin{cases} f_{CLOCK} < \dfrac{f_{HSI} \cdot PRES_REG1^{*}}{(PRES_REG2 + 1) \cdot BASE_REG0} \Rightarrow EVENT0 \\ f_{CLOCK} < \dfrac{f_{HSI} \cdot PRES_REG1^{*}}{(PRES_REG2 + 1) \cdot BASE_REG1} \Rightarrow EVENT1 \end{cases} \quad (3)$$

* $PRES_REG1$ – при $f_{HSI} > f_{CLOCK}$,
$PRES_REG1 + 1$ – при $f_{HSI} < f_{CLOCK}$.

При *увеличении частоты CLOCK* значение М изменяется от М0 до 15. Максимальное значение М сохраняется в поле MAX_CHK_SHIFT_REG1 соответствующего регистра ..._STAT.

При достижении М пороговых значений, заданных полями BASE_REG2 и BASE_REG3, вырабатываются события EVENT2 и EVENT3 соответственно.

$$\begin{cases} f_{CLOCK} > \dfrac{f_{HSI} \cdot (PRES_REG3 + 1) \cdot BASE_REG2}{PRES_REG0^{*}} \Rightarrow EVENT2 \\ f_{CLOCK} > \dfrac{f_{HSI} \cdot (PRES_REG3 + 1) \cdot BASE_REG3}{PRES_REG0^{*}} \Rightarrow EVENT3 \end{cases} \quad (4)$$

* $PRES_REG0$ – при $f_{HSI} < f_{CLOCK}$,
$PRES_REG0 + 1$ – при $f_{HSI} > f_{CLOCK}$.

Непосредственно в блоке Checker на возникающие события может быть настроено аппаратное переключение на тактовую частоту генератора HSI.

Возникающие события также обрабатываются в блоке FT_CNTR, и на эти события могут быть настроены следующие действия:

- сигнал предупреждения;
- аппаратный сброс микросхемы.

Для иллюстрации принципов работы блока возможные комбинации настроек приведены в таблице 23.

Таблица 23 – Примеры работы блока Checker

Частота HSI	Ожидаемая частота	Реальная частота CLOCK	PRES REG2	PRES REG0	PRES REG1	PRES REG3	BASE REG0	BASE REG1	BASE REG2	BASE REG3	Примеры*
8 МГц	80 МГц	80 МГц	1	2	199	1	12	15	12	15	K0 = 10 M0 = 10 Нет событий
8 МГц	80 МГц	100 МГц	1	2	199	1	12	15	12	15	K0 = 10 M0 = 10 K = 8 M = 13 EVENT2 (Fast Clock)
8 МГц	80 МГц	120 МГц	1	2	199	1	12	15	12	15	K0 = 10 M0 = 10 K = 6 M = 15 EVENT2 (Fast Clock) EVENT3 (Faster Clock)
8 МГц	80 МГц	60 МГц	1	2	199	1	12	15	12	15	K0 = 10 M0 = 10 K = 13 M = 8 EVENT0 (Slow Clock)
8 МГц	80 МГц	0 МГц (нет частоты)	1	2	199	1	12	15	12	15	K0 = 10 M0 = 10 K = 16 (∞) M = 0 EVENT0 (Slow Clock) EVENT1 (Slowest Clock)
8 МГц	2,5 МГц	2,5 МГц	5	89	20	2	12	15	12	15	K0 = 11 M0 = 10 Нет событий
8 МГц	2,5 МГц	3,33 МГц	5	89	20	2	12	15	12	15	K0 = 11 M0 = 10 K = 8 M = 13 EVENT2 (Fast Clock)

Частота HSI	Ожидаемая частота	Реальная частота CLOCK	PRES REG2	PRES REG0	PRES REG1	PRES REG3	BASE REG0	BASE REG1	BASE REG2	BASE REG3	Примеры*
8 МГц	2,5 МГц	5 МГц	5	89	20	2	12	15	12	15	K0 = 11 M0 = 10 K = 6 M = 16 EVENT2 (Fast Clock) EVENT3 (Faster Clock)
8 МГц	2,5 МГц	2 МГц	5	89	20	2	12	15	12	15	K0 = 11 M0 = 10 K = 10 M = 8 EVENT0 (Slow Clock)
8 МГц	2,5 МГц	0 МГц (нет частоты)	5	89	20	2	12	15	12	15	K0 = 11 M0 = 10 K = 16 (∞) M = 0 EVENT0 (Slow Clock) EVENT1 (Slowest Clock)
* Фактические значения коэффициентов K и M могут незначительно отличаться от коэффициентов по формулам (1)-(2) в зависимости от частот и делителей											

Максимальное достигнутое значение регистра *SHIFT_REG* каждого из каналов можно считать программно. При чтении регистра *SHIFT_REG* считывается теневой регистр, в котором отражается максимальное достигнутое регистром *SHIFT_REG* значение за период с последнего сброса теневого регистра. Сброс теневого регистра осуществляется программно, записью управляющего бита CLR_CHK_SHIFT_REG. После разрешения работы блока Checker необходимо осуществить сброс теневых регистров, а также флагов возникновения событий, с задержкой после разрешения работы. Задержка после разрешения работы блока (бит EN_CHK) до сброса регистров блока Checker должна быть не менее 2 периодов наименьшей из двух частот, поступающих на блок Checker. Это обусловлено наличием двух частотных доменов. При этом сигнал разрешения пересинхронизируется в оба частотных домена, длительность процесса синхронизации блока - не более 2 периодов наименьшей из двух поступающих на блок частот. Нормальная работа блока Checker возможна только после того, как сигнал

разрешения будет установлен в обоих доменах, поскольку сигналы из одного тактового домена используются в качестве асинхронных сбросов регистров в другом

Блок Checker установлен на входах тактовых частот от внешних источников HSE0, HSE1 и LSE, встроенного генератора LSI и всех PLL. Также отдельно стоит блок Checker на выходе схемы формирования тактовых сигналов процессорных ядер.

Блоки Checker проверяют наличие и значения частот тактовых сигналов, поступающих только в цифровую часть микросхемы. Блок часов реального времени RTC может быть сконфигурирован на использование тактовой частоты от генератора LSE или LSI, при этом частота на RTC поступает напрямую с генератора и не контролируется блоком Checker. В результате исчезновение частоты LSE приведет к остановке часов реального времени RTC. Этот аспект особенно важен при работе микросхемы в режиме STANDBY с пробуждением от RTC. Так как в этом случае при отказе генератора и остановке часов пробуждение будет невозможно.

Частота генератора LSI поступает напрямую на схему формирования задержки $T_{PORRSTn}$, но в блоке формирования сигнала U_{CC}RESET аппаратно контролируется формирование этой задержки, и в случае отсутствия частоты сигнал U_{CC}RESET будет сгенерирован, но с много меньшей задержкой.

7.2.3 Описание регистров

Таблица 24 – Описание регистров

Базовый адрес	Обозначение	Описание
0x4000_0000	MDR_CLK	Контроллер тактовых частот
Смещение		
0x0000	KEY	Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	MAX_CLK	Регистр настройки частоты MAX_CLK
Тактирование процессорного ядра		
0x0000_0008	CPU_CLK	Регистр управления тактовой частотой процессорных ядер
0x0000_000C	PER0_CLK	Регистр управления тактированием периферии
0x0000_0010	PER1_CLK	Регистр управления тактированием периферии
0x0000_0014	CPU_CHK0	Регистр настройки монитора процессорной частоты CPU_CLK, регистр PRES_REG0, PRES_REG1
0x0000_0018	CPU_CHK1	Регистр настройки монитора процессорной частоты CPU_CLK, регистр PRES_REG2, PRES_REG3
0x0000_001C	CPU_CHK2	Регистр настройки монитора процессорной частоты CPU_CLK, регистр BASE_REG0,1,2,3
0x0000_0020	CPU_STAT	Регистр состояния тактовой частоты процессорных ядер

Базовый адрес	Обозначение	Описание
Генератор LSI		
0x0000_0024	LSI_CLK	Регистр настройки частоты LSI
0x0000_0028	LSI_CHK0	Регистр настройки монитора частоты LSI, регистр PRES_REG0
0x0000_002C	LSI_CHK1	Регистр настройки монитора частоты LSI, регистр PRES_REG1
0x0000_0030	LSI_CHK2	Регистр настройки монитора частоты LSI, регистр BASE_REG0
0x0000_0034	LSI_STAT	Регистр состояния тактовой частоты LSI
Генератор HSI		
0x0000_0038	HSI_STAT	Регистр состояния тактовой частоты HSI
Генератор LSE		
0x0000_003C	LSE_CLK	Регистр настройки частоты LSE
0x0000_0040	LSE_CHK0	Регистр настройки монитора частоты LSE, регистр PRES_REG0
0x0000_0044	LSE_CHK1	Регистр настройки монитора частоты LSE, регистр PRES_REG1
0x0000_0048	LSE_CHK2	Регистр настройки монитора частоты LSE, регистр BASE_REG0
0x0000_004C	LSE_STAT	Регистр состояния тактовой частоты LSE
Генератор HSE0		
0x0000_0050	HSE0_CLK	Регистр настройки генератора HSE0
0x0000_0054	HSE0_CHK0	Регистр настройки монитора частоты HSE0, регистр PRES_REG0
0x0000_0058	HSE0_CHK1	Регистр настройки монитора частоты HSE0, регистр PRES_REG1
0x0000_005C	HSE0_CHK2	Регистр настройки монитора частоты HSE0, регистр BASE_REG0
0x0000_0060	HSE0_STAT	Регистр состояния тактовой частоты HSE0
Генератор HSE1		
0x0000_0064	HSE1_CLK	Регистр настройки генератора HSE1
0x0000_0068	HSE1_CHK0	Регистр настройки монитора частоты HSE1, регистр PRES_REG0
0x0000_006C	HSE1_CHK1	Регистр настройки монитора частоты HSE1, регистр PRES_REG1
0x0000_0070	HSE1_CHK2	Регистр настройки монитора частоты HSE1, регистр BASE_REG0
0x0000_0074	HSE1_STAT	Регистр состояния тактовой частоты HSE1

Базовый адрес	Обозначение	Описание
Блок умножения PLL0		
0x0000_0078	PLL0_CLK	Регистр настройки PLL0
0x0000_007C	PLL0_CHK0	Регистр настройки монитора частоты PLL0, регистр PRES_REG0
0x0000_0080	PLL0_CHK1	Регистр настройки монитора частоты PLL0, регистр PRES_REG1
0x0000_0084	PLL0_CHK2	Регистр настройки монитора частоты PLL0, регистр BASE_REG0
0x0000_0088	PLL0_STAT	Регистр состояния тактовой частоты PLL0
Блок умножения PLL1		
0x0000_008C	PLL1_CLK	Регистр настройки PLL1
0x0000_0090	PLL1_CHK0	Регистр настройки монитора частоты PLL1, регистр PRES_REG0
0x0000_0094	PLL1_CHK1	Регистр настройки монитора частоты PLL1, регистр PRES_REG1
0x0000_0098	PLL1_CHK2	Регистр настройки монитора частоты PLL1, регистр BASE_REG0
0x0000_009C	PLL1_STAT	Регистр состояния тактовой частоты PLL1
Блок умножения PLL2		
0x0000_00A0	PLL2_CLK	Регистр настройки PLL2
0x0000_00A4	PLL2_CHK0	Регистр настройки монитора частоты PLL2, регистр PRES_REG0
0x0000_00A8	PLL2_CHK1	Регистр настройки монитора частоты PLL2, регистр PRES_REG1
0x0000_00AC	PLL2_CHK2	Регистр настройки монитора частоты PLL2, регистр BASE_REG0
0x0000_00B0	PLL2_STAT	Регистр состояния тактовой частоты PLL2
0x00B4 – 0x0114		Зарезервировано
Тактирование CAN		
0x0000_0118	CAN0_CLK	Управление тактированием контроллера CAN0
0x0000_011C – 0x0000_012C		Зарезервировано
Тактирование TIMER		
0x0000_0130	TIM0_CLK	Управление тактированием контроллера TIMER0
0x0000_0134	TIM1_CLK	Управление тактированием контроллера TIMER1
0x0000_0138	TIM2_CLK	Управление тактированием контроллера TIMER2

Базовый адрес	Обозначение	Описание
0x0000_013C	TIM3_CLK	Управление тактированием контроллера TIMER3
0x0000_0140 – 0x0000_0144		Зарезервировано
Тактирование MIL (МКПД)		
0x0000_0148 – 0x0000_0154		Зарезервировано
Тактирование ARINC-249		
0x0000_0158 – 0x0000_0164		Зарезервировано
Тактирование ETHERNET		
0x0000_0168 – 0x0000_016C		Зарезервировано
0x0000_0170	EPHY0_CLK	Управление тактированием контроллера EthernetPHY0
0x0000_0174		Зарезервировано
Тактирование SPACEWIRE		
0x0000_0178	SPW0_CLK	Управление тактированием контроллера SpaceWire0
0x0000_017C – 0x0184		Зарезервировано
Тактирование UART		
0x0000_0188	UART0_CLK	Управление тактированием контроллера UART0
0x0000_018C	UART1_CLK	Управление тактированием контроллера UART1
0x0000_0190 – 0x019C		Зарезервировано
Тактирование SSP		
0x0000_01A0	SSP0_CLK	Управление тактированием контроллера SSP0
0x01B8		Зарезервировано
Тактирование ADC		
0x0000_01BC	ADC0_CLK	Управление тактированием контроллера ADC0
0x0000_01C0	ADC1_CLK	Управление тактированием контроллера ADC1
Тактирование DAC		
0x0000_01C4	DAC0_CLK	Управление тактированием контроллера DAC0
0x0000_01C8	DAC1_CLK	Управление тактированием контроллера DAC1
Тактирование RTC		
0x0000_01CC	RTC_CLK	Управление тактированием контроллера RTC

7.2.3.1 **Регистр KEY**

Base ADDR=		0x4000_0000			Offset=		0x0000_0000								
REG Name:		KEY													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 25 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока CLKCNTR

7.2.3.2 **Регистр MAX_CLK**

Base ADDR=		0x4000_0000			Offset=		0x0000_0004								
REG Name:		MAX_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-												SELECT[3:0]			

Таблица 26 – Описание бит регистра MAX_CLK

Бит	Имя	Значение	Описание
31…4	-		Зарезервировано
3…0	SELECT[3:0]		Номер источника тактовой частоты: 4’b 00000 – Генератор HSI; 4’b 00001 – Генератор HSI/2; 4’b 00010 – Генератор HSE0; 4’b 00011 – Генератор HSE0/2; 4’b 00100 – Генератор HSE1; 4’b 00101 – Генератор HSE1/2; 4’b 00110 – Генератор LSI; 4’b 00111 – Генератор LSE; 4’b 01000 – Генератор PLL0; 4’b 01001 – Генератор PLL1; 4’b 01010 – Генератор PLL2; 4’b 01011 – Зарезервировано; 4’b 01100 – Зарезервировано; 4’b 01101 – Зарезервировано; 4’b 01110 – Зарезервировано; 4’b 01111 – Зарезервировано

7.2.3.3 **Регистр CPU_CLK**

Base ADDR=		0x4000_0000			Offset=		0x0000_0008								
REG Name:		CPU_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-					CLR_CHK_EVENT3	CLR_CHK_EVENT2	CLR_CHK_EVENT1	CLR_CHK_EVENT0	CLR_CHK_ SHIFT_REG1	CLR_CHK_ SHIFT_REG0	EN_CHK	EN_CHK_EVENT3	EN_CHK_EVENT2	EN_CHK_EVENT1	EN_CHK_EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 27 – Описание бит регистра CPU_CLK

Бит	Имя	Значение	Описание
31…27	-		Зарезервировано
26	CLR_CHK_EVENT3		Бит сброса флага EVENT3: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2		Бит сброса флага EVENT2: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1		Бит сброса флага EVENT1: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0		Бит сброса флага EVENT0: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1: 1 – сброс регистра 0 – нет сброса Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита

Бит	Имя	Значение	Описание
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
20	EN_CHK		Бит разрешения контроля частоты FCLK блоком CLK_CHECKER: 0 – частота не контролируется; 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота): 0 – ничего не делать при событии EVENT3; 1 – перейти на частоту HSI при событии EVENT3. Имеет смысл только при EN_CHK = 1. Для FCLK аварийное переключение не предусмотрено
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота): 0 – ничего не делать при событии EVENT2; 1 – перейти на частоту HSI при событии EVENT2. Имеет смысл только при EN_CHK = 1. Для FCLK аварийное переключение не предусмотрено
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота): 0 – ничего не делать при событии EVENT1; 1 – перейти на частоту HSI при событии EVENT1. Имеет смысл только при EN_CHK = 1. Для FCLK аварийное переключение не предусмотрено
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты): 0 – ничего не делать при событии EVENT0; 1 – перейти на частоту HSI при событии EVENT0. Имеет смысл только при EN_CHK = 1. Для FCLK аварийное переключение не предусмотрено
15…0	DIV[15:0]		Делитель частоты MAX_CLK для формирования частоты FCLK(CPU_CLK). FCLK = MAX_CLK/(DIV+1)

7.2.3.4 Регистр PER0_CLK

Base ADDR=		0x4000_0000			Offset=		0x0000_000C								
REG Name:		PER0_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		MDR_ CAN0_EN	-		MDR_ TMR3_E	MDR_ TMR2_E	MDR_ TMR1_E	MDR_ TMR0_E	-	MDR_ SPW0_EN	-			MDR_ PORTE_E	MDR_ PORTD_E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDR_ PORTC_E	MDR_ PORTB_E	MDR_ PORTA_E	-												

Таблица 28 – Описание бит регистра PER0_CLK

Бит	Имя	Значение	Описание
31, 30	-		Зарезервировано
29	MDR_CAN0_EN		Бит разрешения тактирования частотой CAN0: 0 – тактирование запрещено; 1 – тактирование разрешено
28, 27	-		Зарезервировано
26	MDR_TMR3_EN		Бит разрешения тактирования частотой TMR3: 0 – тактирование запрещено; 1 – тактирование разрешено
25	MDR_TMR2_EN		Бит разрешения тактирования частотой TMR2: 0 – тактирование запрещено; 1 – тактирование разрешено
24	MDR_TMR1_EN		Бит разрешения тактирования частотой TMR1: 0 – тактирование запрещено; 1 – тактирование разрешено
23	MDR_TMR0_EN		Бит разрешения тактирования частотой TMR0: 0 – тактирование запрещено; 1 – тактирование разрешено
22	-		Зарезервировано
21	MDR_SPW0_EN		Бит разрешения тактирования частотой SpaceWire: 0 – тактирование запрещено; 1 – тактирование разрешено
20…18	-		Зарезервировано
17	MDR_PORTE_EN		Бит разрешения тактирования частотой PORTE: 0 – тактирование запрещено; 1 – тактирование разрешено

Бит	Имя	Значение	Описание
16	MDR_PORTD_EN		Бит разрешения тактирования частотой PORTD: 0 – тактирование запрещено; 1 – тактирование разрешено
15	MDR_PORTC_EN		Бит разрешения тактирования частотой PORTC: 0 – тактирование запрещено; 1 – тактирование разрешено
14	MDR_PORTB_EN		Бит разрешения тактирования частотой PORTB: 0 – тактирование запрещено; 1 – тактирование разрешено
13	MDR_PORTA_EN		Бит разрешения тактирования частотой PORTA: 0 – тактирование запрещено; 1 – тактирование разрешено
12…0	-		Зарезервировано

7.2.3.5 **Регистр PER1_CLK**

Base ADDR=		0x4000_0000			Offset=		0x0000_0010								
REG Name:		PER1_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-					MDR_DAC1_EN	MDR_DAC0_EN	MDR_ADC1_EN	MDR_ADC0_EN	-	MDR_MIL1_EN	-	MDR_MIL0_EN	-		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		MDR_ARCTX0_E	-	MDR_ARCRX0_E	-			MDR_UART1_EN	MDR_UART0_EN	-			MDR_SSP0_EN	-	

Таблица 29 – Описание бит регистра PER1_CLK

Бит	Имя	Значение	Описание
31…27	-		Зарезервировано
26	MDR_DAC1_EN		Бит разрешения тактирования частотой DAC1: 0 – тактирование запрещено; 1 – тактирование разрешено
25	MDR_DAC0_EN		Бит разрешения тактирования частотой DAC0: 0 – тактирование запрещено; 1 – тактирование разрешено
24	MDR_ADC1_EN		Бит разрешения тактирования частотой ADC1: 0 – тактирование запрещено; 1 – тактирование разрешено

Бит	Имя	Значение	Описание
23	MDR_ADC0_EN		Бит разрешения тактирования частотой ADC0: 0 – тактирование запрещено; 1 – тактирование разрешено
22	-		Зарезервировано
21	MDR_MIL1_EN		Бит разрешения тактирования частотой MIL1: 0 – тактирование запрещено; 1 – тактирование разрешено
20	-		Зарезервировано
19	MDR_MIL0_EN		Бит разрешения тактирования частотой MIL0: 0 – тактирование запрещено; 1 – тактирование разрешено
18…14	-		Зарезервировано
13	MDR_ARCTX0_EN		Бит разрешения тактирования частотой ARCTX0: 0 – тактирование запрещено; 1 – тактирование разрешено
12	-		Зарезервировано
11	MDR_ARCRX0_EN		Бит разрешения тактирования частотой ARCRX0: 0 – тактирование запрещено; 1 – тактирование разрешено
10…8	-		Зарезервировано
7	MDR_UART1_EN		Бит разрешения тактирования частотой UART1: 0 – тактирование запрещено; 1 – тактирование разрешено
6	MDR_UART0_EN		Бит разрешения тактирования частотой UART0: 0 – тактирование запрещено; 1 – тактирование разрешено
5…3	-		Зарезервировано
2	MDR_SSP0_EN		Бит разрешения тактирования частотой SSP0: 0 – тактирование запрещено; 1 – тактирование разрешено
1, 0	-		Зарезервировано

7.2.3.6 **Регистр CPU_CHK0**

Base ADDR=		0x4000_0000			Offset=		0x0000_0014								
REG Name:		CPU_CHK0													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Таблица 30 – Описание бит регистра CPU_CHK0

Бит	Имя	Значение	Описание
31…16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1: RST(SHIFT_REG1) = HSI / (PRES_REG0+1)
15…0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0: CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)

7.2.3.7 Регистр CPU_CHK1

Base ADDR=		0x4000_0000			Offset=		0x0000_0018								
REG Name:		CPU_CHK1													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Таблица 31 – Описание бит регистра CPU_CHK1

Бит	Имя	Значение	Описание
31…16	PRES_REG3[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG0 RST(SHIFT_REG0) = FCLK / (PRES_REG3+1)
15…0	PRES_REG1[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG1 CLK(SHIFT_REG1) = FCLK / (PRES_REG1+1)

7.2.3.8 Регистр CPU_CHK2

Base ADDR=		0x4000_0000			Offset=		0x0000_001C								
REG Name:		CPU_CHK2													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
–				BASE_REG3[3:0]				–				BASE_REG2[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–				BASE_REG1[3:0]				–				BASE_REG0[3:0]			

Таблица 32 – Описание бит регистра CPU_CHK2

Бит	Имя	Значение	Описание
31…28	–		Зарезервировано

27…24	BASE_REG3[3:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23…20	–		Зарезервировано
19…16	BASE_REG2[3:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15…12	–		Зарезервировано
11…8	BASE_REG1[3:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7…4	–		Зарезервировано
3…0	BASE_REG0[3:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.9 **Регистр CPU_STAT**

Base ADDR=		0x4000_0000			Offset=		0x0000_0020								
REG Name:		CPU_STAT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-												EVENT3	EVENT2	EVENT1	EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–			MAX_CHK_SHIFT_REG1[4:0]					–			MAX_CHK_SHIFT_REG0[4:0]				

Таблица 33 – Описание бит регистра CPU_STAT

Бит	Имя	Значение	Описание
31…20	-		Зарезервировано
19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота): 0 – нет события; 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2 (высокая частота): 0 – нет события; 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота): 0 – нет события; 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты): 0 – нет события; 1 – есть событие
15…13	–		Зарезервировано

Бит	Имя	Значение	Описание
12…8	MAX_CHK_ SHIFT_REG1[4:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7…5	–		Зарезервировано
4…0	MAX_CHK_ SHIFT_REG0[4:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.10 **Регистр LSI_CLK***

Base ADDR=	0x4000_0000			Offset=		0x0000_0024					
REG Name:	LSI_CLK										
31…27	26	25	24	23	22	21	20	19	18	17	16
-	LSI_READY	-		CLR_CHK_ EVENT	CLR_CHK_ SHIFT_REG1	CLR_CHK_ SHIFT_REG0	EN_CHK	EN_CHK_ EVENT3	EN_CHK_ EVENT2	EN_CHK_ EVENT1	EN_CHK_ EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Таблица 34 – Описание бит регистра LSI_CLK

Бит	Имя	Значение	Описание
31…27	-		Зарезервировано
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита

* Управление генератором LSI осуществляется через регистры батарейного домена.

Бит	Имя	Значение	Описание
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
20	EN_CHK		Бит разрешения контроля частоты LSI: 0 – частота не контролируется; 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота): 0 – ничего не делать при событии EVENT3; 1 – перейти на частоту HSI при событии EVENT3. Имеет смысл только при EN_CHK = 1
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота): 0 – ничего не делать при событии EVENT2; 1 – перейти на частоту HSI при событии EVENT2. Имеет смысл только при EN_CHK = 1
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота): 0 – ничего не делать при событии EVENT1; 1 – перейти на частоту HSI при событии EVENT1. Имеет смысл только при EN_CHK = 1
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты): 0 – ничего не делать при событии EVENT0; 1 – перейти на частоту HSI при событии EVENT0. Имеет смысл только при EN_CHK = 1

Бит	Имя	Значение	Описание
15…0	-		Зарезервировано

7.2.3.11 **Регистр LSI_CHK0**

Base ADDR=		0x4000_0000			Offset=		0x0000_0028								
REG Name:		LSI_CHK0													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Таблица 35 – Описание бит регистра LSI_CHK0

Бит	Имя	Значение	Описание
31…16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1: RST(SHIFT_REG1) = HSI / (PRES_REG0+1)
15…0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0: CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)

7.2.3.12 **Регистр LSI_CHK1**

Base ADDR=		0x4000_0000			Offset=		0x0000_002C								
REG Name:		LSI_CHK1													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Таблица 36 – Описание бит регистра LSI_CHK1

Бит	Имя	Значение	Описание
31…16	PRES_REG3[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG0: RST(SHIFT_REG0) = LSI / (PRES_REG3+1)
15…0	PRES_REG1[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG1: CLK(SHIFT_REG1) = LSI / (PRES_REG1+1)

7.2.3.13 **Регистр LSI_CHK2**

Base ADDR=		0x4000_0000			Offset=		0x0000_0030								
REG Name:		LSI_CHK2													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
–				BASE_REG3[3:0]				–				BASE_REG2[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–				BASE_REG1[3:0]				–				BASE_REG0[3:0]			

Таблица 37 – Описание бит регистра LSI_CHK2

Бит	Имя	Значение	Описание
31…28	–		Зарезервировано
27…24	BASE_REG3[3:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23…20	–		Зарезервировано
19…16	BASE_REG2[3:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15…12	–		Зарезервировано
11…8	BASE_REG1[3:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7…4	–		Зарезервировано
3…0	BASE_REG0[3:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.14 **Регистр LSI_STAT**

Base ADDR=		0x4000_0000			Offset=		0x0000_0034								
REG Name:		LSI_STAT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											LSI_RDY	EVENT3	EVENT2	EVENT1	EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–			MAX_CHK_SHIFT_REG1[4:0]					–			MAX_CHK_SHIFT_REG0[4:0]				

Таблица 38 – Описание бит регистра LSI_STAT

Бит	Имя	Значение	Описание
31…21	-		Зарезервировано
20	LSI_READY	-	Флаг готовности генератора LSI: 0 – генератор не готов или выключен; 1 – генератор готов

19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота): 0 – нет события; 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2(высокая частота): 0 – нет события; 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота): 0 – нет события; 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты): 0 – нет события; 1 – есть событие
15…13	–		Зарезервировано
12…8	MAX_CHK_ SHIFT_REG1[4:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7…5	–		Зарезервировано
4…0	MAX_CHK_ SHIFT_REG0[4:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.15 **Регистр HSI_STAT***

Base ADDR=		0x4000_0000			Offset=		0x0000_0038								
REG Name:		HSI_STAT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-											HSI_RDY	-			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Таблица 39 – Описание бит регистра HSI_STAT

Бит	Имя	Значение	Описание
31…21	-		Зарезервировано
20	HSI_READY	-	Флаг готовности генератора HSI: 0 – генератор не готов или выключен; 1 – генератор готов
19…0	-		Зарезервировано

* Управление генератором HSI осуществляется через регистры батарейного домена.

7.2.3.16 **Регистр LSE_CLK***

Base ADDR=					0x4000_0000				Offset=		0x0000_003C				
REG Name:					LSE_CLK										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-					LSE_READY	-		CLR_CHK_EVENT	CLR_CHK_SHIFT_REG1	CLR_CHK_SHIFT_REG0	EN_CHK	EN_CHK_EVENT3	EN_CHK_EVENT2	EN_CHK_EVENT1	EN_CHK_EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Таблица 40 – Описание бит регистра LSE_CLK

Бит	Имя	Значение	Описание
31…27	-		Зарезервировано
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита

* Управление генератором LSE осуществляется через регистры батарейного домена.

Бит	Имя	Значение	Описание
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0 Запись 1 сбрасывает теневой регистр
20	EN_CHK		Бит разрешения контроля частоты LSE: 0 – частота не контролируется; 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота): 0 – ничего не делать при событии EVENT3; 1 – перейти на частоту HSI при событии EVENT3. Имеет смысл только при EN_CHK = 1
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота): 0 – ничего не делать при событии EVENT2; 1 – перейти на частоту HSI при событии EVENT2. Имеет смысл только при EN_CHK = 1
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота): 0 – ничего не делать при событии EVENT1; 1 – перейти на частоту HSI при событии EVENT1. Имеет смысл только при EN_CHK = 1
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты): 0 – ничего не делать при событии EVENT0; 1 – перейти на частоту HSI при событии EVENT0. Имеет смысл только при EN_CHK = 1
15…0	-		Зарезервировано

7.2.3.17 **Регистр LSE_CHK0**

Base ADDR=		0x4000_0000			Offset=		0x0000_0040								
REG Name:		LSE_CHK0													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Таблица 41 – Описание бит регистра LSE_CHK0

Бит	Имя	Значение	Описание

31…16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1: RST(SHIFT_REG1) = HSI / (PRES_REG0+1)
15…0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0: CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)

7.2.3.18 **Регистр LSE_CHK1**

Base ADDR=		0x4000_0000			Offset=		0x0000_0044								
REG Name:		LSE_CHK1													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Таблица 42 – Описание бит регистра LSE_CHK1

Бит	Имя	Значение	Описание
31…16	PRES_REG3[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG0: RST(SHIFT_REG0) = LSE / (PRES_REG3+1)
15…0	PRES_REG1[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG1: CLK(SHIFT_REG1) = LSE / (PRES_REG1+1)

7.2.3.19 **Регистр LSE_CHK2**

Base ADDR=		0x4000_0000			Offset=		0x0000_0048								
REG Name:		LSE_CHK2													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
–				BASE_REG3[3:0]				–				BASE_REG2[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–				BASE_REG1[3:0]				–				BASE_REG0[3:0]			

Таблица 43 – Описание бит регистра LSE_CHK2

Бит	Имя	Значение	Описание
31…28	–		Зарезервировано
27…24	BASE_REG3[3:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23…20	–		Зарезервировано
19…16	BASE_REG2[3:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)

15…12	–		Зарезервировано
11…8	BASE_REG1[3:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7…4	–		Зарезервировано
3…0	BASE_REG0[3:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.20 **Регистр LSE_STAT**

Base ADDR=		0x4000_0000				Offset=		0x0000_004C							
REG Name:		LSE_STAT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-											LSE_RDY	EVENT3	EVENT2	EVENT1	EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–			MAX_CHK_SHIFT_REG1[4:0]					–			MAX_CHK_SHIFT_REG0[4:0]				

Таблица 44 – Описание бит регистра LSE_STAT

Бит	Имя	Значение	Описание
31…21	-		Зарезервировано
20	LSE_READY	-	Флаг готовности генератора LSE: 0 – генератор не готов или выключен; 1 – генератор готов
19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота): 0 – нет события; 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2(высокая частота): 0 – нет события; 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота): 0 – нет события; 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты): 0 – нет события; 1 – есть событие
13…15	–		Зарезервировано
12…8	MAX_CHK_ SHIFT_REG1[4:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра

7…5	–		Зарезервировано
4…0	MAX_CHK_ SHIFT_REG0[4:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.21 **Регистр HSEn_CLK**

Base ADDR=		0x4000_0000					Offset=		0x0000_0050 0x0000_0064						
REG Name:		HSEn_CLK													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		HSE_FILTER_EN	HSEn_BYP	HSEn_ON	CLR_CHK_EVENT3	CLR_CHK_EVENT2	CLR_CHK_EVENT1	CLR_CHK_EVENT0	CLR_CHK_SHIFT_REG1	CLR_CHK_SHIFT_REG0	EN_CHK	EN_CHK_EVENT3	EN_CHK_EVENT2	EN_CHK_EVENT1	EN_CHK_EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Таблица 45 – Описание бит регистра HSEn_CLK

Бит	Имя	Значение	Описание
31…30	-		Зарезервировано
29	HSE_FILTER_EN	-	Бит разрешения встроенного фильтра высоких частот: 0 – фильтр отключен; 1 – фильтр включен
28	HSEn_BYP	-	Бит выбора режима работы генератора HSEn: 0 – режим работы с внешним резонатором; 1 – режим работы с внешним генератором. Устанавливать совместно с HSEn_ON = 1
27	HSEn_ON	-	Бит разрешения работы генератора HSEn: 0 – генератор выключен; 1 – генератор включен
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1: 1 – сброс регистра;

Бит	Имя	Значение	Описание
			0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
20	EN_CHK		Бит разрешения контроля частоты HSEn: 0 – частота не контролируется; 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота): 0 – ничего не делать при событии EVENT3; 1 – перейти на частоту HSI при событии EVENT3. Имеет смысл только при EN_CHK = 1
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота): 0 – ничего не делать при событии EVENT2; 1 – перейти на частоту HSI при событии EVENT2. Имеет смысл только при EN_CHK = 1
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота): 0 – ничего не делать при событии EVENT1; 1 – перейти на частоту HSI при событии EVENT1. Имеет смысл только при EN_CHK = 1
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты): 0 – ничего не делать при событии EVENT0; 1 – перейти на частоту HSI при событии EVENT0.

Бит	Имя	Значение	Описание
			Имеет смысл только при EN_CHK = 1
15…0	-		Зарезервировано

7.2.3.22 **Регистр HSEn_CHK0**

Base ADDR=		0x4000_0000			Offset=		0x0000_0054 0x0000_0068								
REG Name:		HSEn_CHK0													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Таблица 46 – Описание бит регистра HSEn_CHK0

Бит	Имя	Значение	Описание
31…16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1: RST(SHIFT_REG1) = HSI / (PRES_REG0+1)
15…0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0: CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)

7.2.3.23 **Регистр HSEn_CHK1**

Base ADDR=		0x4000_0000			Offset=		0x0000_0058 0x0000_006C								
REG Name:		HSEn_CHK1													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Таблица 47 – Описание бит регистра HSEn_CHK1

Бит	Имя	Значение	Описание
31…16	PRES_REG3[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG0: RST(SHIFT_REG0) = HSEn / (PRES_REG3+1)
15…0	PRES_REG1[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG1: CLK(SHIFT_REG1) = HSEn / (PRES_REG1+1)

7.2.3.24 **Регистр HSEn_CHK2**

Base ADDR=		0x4000_0000			Offset=		0x0000_005C 0x0000_0070								
REG Name:		HSEn_CHK2													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
–				BASE_REG3[3:0]				–				BASE_REG2[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–				BASE_REG1[3:0]				–				BASE_REG0[3:0]			

Таблица 48 – Описание бит регистра HSEn_CHK2

Бит	Имя	Значение	Описание
31…28	–		Зарезервировано
27…24	BASE_REG3[3:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23…20	–		Зарезервировано
19…16	BASE_REG2[3:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15…12	–		Зарезервировано
11…8	BASE_REG1[3:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7…4	–		Зарезервировано
3…0	BASE_REG0[3:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.25 **Регистр HSEn_STAT**

Base ADDR=			0x4000_0000			Offset=		0x0000_0060 0x0000_0074							
REG Name:			HSEn_STAT												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											HSEn_RDY	EVENT3	EVENT2	EVENT1	EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–			MAX_CHK_SHIFT_REG1[4:0]					–			MAX_CHK_SHIFT_REG0[4:0]				

Таблица 49 – Описание бит регистра HSEn_STAT

Бит	Имя	Значение	Описание
31…21	-		Зарезервировано
20	HSEn_RDY	-	Флаг готовности генератора HSEn: 0 – генератор не готов или выключен; 1 – генератор готов
19	EVENT3		Флаг возникновения события EVENT3(очень высокая частота): 0 – нет события; 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2(высокая частота): 0 – нет события; 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота): 0 – нет события; 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты): 0 – нет события; 1 – есть событие
15…13	–		Зарезервировано
12…8	MAX_CHK_ SHIFT_REG1[4:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7…5	–		Зарезервировано
4…0	MAX_CHK_ SHIFT_REG0[4:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.26 **Регистр PLLn_CLK**

Base ADDR=				0x4000_0000			Offset=		0x0000_0078 0x0000_008C 0x0000_00A0						
REG Name:				PLLn_CLK											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[2:0]			PLLn_RLD	PLLn_ON	CLR_CHK_ EVENT3	CLR_CHK_ EVENT2	CLR_CHK_ EVENT1	CLR_CHK_ EVENT0	CLR_CHK_ SHIFT_REG1	CLR_CHK_ SHIFT_REG0	EN_CHK	EN_CHK_ EVENT3	EN_CHK_ EVENT2	EN_CHK_ EVENT1	EN_CHK_ EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	PLLn_N[6:0]							-			DV	PLLn_Q[3:0]			

Таблица 50 – Описание бит регистра PLLn_CLK

Бит	Имя	Значение	Описание
31…29	SELECT[2:0]		Номер источника тактовой частоты: 3’b 000 – Генератор HSI; 3’b 001 – Генератор HSI/2; 3’b 010 – Генератор HSE0; 3’b 011 – Генератор HSE0/2; 3’b 100 – Генератор HSE1; 3’b 101 – Генератор HSE1/2; 3’b 110 и 111 – Зарезервировано. Переключение частоты необходимо выполнять с включенного источника
28	PLLn_RLD	-	Бит перезапуска работы PLLn. При смене коэффициентов умножения/ деления запись 1 в этот бит перезапускает PLLn, при этом сбрасывается флаг готовности PLLn_RDY и только после выхода на режим флаг взводится обратно
27	PLLn_ON	-	Бит разрешения работы генератора PLLn: 0 – PLLn выключена; 1 – PLLn включена
26	CLR_CHK_EVENT3	-	Бит сброса флага EVENT3: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
25	CLR_CHK_EVENT2	-	Бит сброса флага EVENT2: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
24	CLR_CHK_EVENT1	-	Бит сброса флага EVENT1: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
23	CLR_CHK_EVENT0	-	Бит сброса флага EVENT0: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита

Бит	Имя	Значение	Описание
22	CLR_CHK_SHIFT_REG1		Бит сброса теневого регистра максимального значения регистра SHIFT_REG1: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
21	CLR_CHK_SHIFT_REG0		Бит сброса теневого регистра максимального значения регистра SHIFT_REG0: 1 – сброс регистра; 0 – нет сброса. Устанавливается в 0 автоматически, запись 1 имеет смысл только при 0 значении бита
20	EN_CHK		Бит разрешения контроля частоты PLLn: 0 – частота не контролируется; 1 – монитор работает
19	EN_CHK_EVENT3		Бит разрешения аварийного переключения на частоту HSI (очень высокая частота): 0 – ничего не делать при событии EVENT3; 1 – перейти на частоту HSI при событии EVENT3. Имеет смысл только при EN_CHK = 1
18	EN_CHK_EVENT2		Бит разрешения аварийного переключения на частоту HSI (высокая частота): 0 – ничего не делать при событии EVENT2; 1 – перейти на частоту HSI при событии EVENT2. Имеет смысл только при EN_CHK = 1
17	EN_CHK_EVENT1		Бит разрешения аварийного переключения на частоту HSI (низкая частота): 0 – ничего не делать при событии EVENT1; 1 – перейти на частоту HSI при событии EVENT1. Имеет смысл только при EN_CHK = 1
16	EN_CHK_EVENT0		Бит разрешения аварийного переключения на частоту HSI (нет частоты): 0 – ничего не делать при событии EVENT0; 1 – перейти на частоту HS при событии EVENT0. Имеет смысл только при EN_CHK = 1
15	-		Зарезервировано
14…8	PLLn_N[6:0]		Коэффициент умножения тактовой частоты PLLn_N = K_{NPLL} FINT = FIN • (K_{NPLL}) / (K_{QPLL}) Значение PLLn_N (K_{NPLL}) может быть в диапазоне от 4 до 75. Частота FINT, полученная после умножения FIN на K_{NPLL} и деления на K_{QPLL}, должна быть в диапазоне от 64 до 150 МГц

Бит	Имя	Значение	Описание
7…5	-		Зарезервировано
4	DV		Бит дополнительного деления PLL: FOUT = FINT/(DV+1)
3…0	PLLn_Q[3:0]		Коэффициент деления тактовой частоты PLLn_Q + 1 = K_{QPLL} FINT = FIN • (K_{NPLL}) / (K_{QPLL}). Значение PLLn_Q может быть в диапазоне от 0 до 15. Значение K_{QPLL} может быть в диапазоне от 1 до 16. Частота FINT, полученная после умножения FIN на K_{NPLL} и деления на K_{QPLL}, должна быть в диапазоне от 64 до 150 МГц

7.2.3.27 **Регистр PLLn_CHK0**

Base ADDR=		0x4000_0000			Offset=		0x0000_007C 0x0000_0090 0x0000_00A4								
REG Name:		PLLn_CHK0													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG0[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG2[15:0]															

Таблица 51 – Описание бит регистра PLLn_CHK0

Бит	Имя	Значение	Описание
31…16	PRES_REG0[15:0]		Значение регистра делителя частоты сброса для счетчика SHIFT_REG1: RST(SHIFT_REG1) = HSI / (PRES_REG0+1)
15…0	PRES_REG2[15:0]		Значение регистра делителя частоты HSI для счетчика на SHIFT_REG0: CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)

7.2.3.28 **Регистр PLLn_CHK1**

Base ADDR=		0x4000_0000			Offset=		0x0000_0080 0x0000_0094 0x0000_00A8								
REG Name:		PLLn_CHK1													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRES_REG3[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRES_REG1[15:0]															

Таблица 52 – Описание бит регистра PLLn_CHK1

Бит	Имя	Значение	Описание
31…16	PRES_REG3[15:0]		Значение регистра делителя частоты PLLn сброса для счетчика SHIFT_REG0: RST(SHIFT_REG0) = PLLn / (PRES_REG3+1)
15…0	PRES_REG1[15:0]		Значение регистра делителя частоты PLLn для счетчика на SHIFT_REG1: CLK(SHIFT_REG1) = PLLn / (PRES_REG1+1)

7.2.3.29 **Регистр PLLn_CHK2**

Base ADDR=				0x4000_0000				Offset=	0x0000_0084 0x0000_0098 0x0000_00AC						
REG Name:				PLLn_CHK2											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-				BASE_REG3[3:0]				-				BASE_REG2[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				BASE_REG1[3:0]				-				BASE_REG0[3:0]			

Таблица 53 – Описание бит регистра PLLn_CHK2

Бит	Имя	Значение	Описание
27…24	BASE_REG3[3:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23…20	-		Зарезервировано
19…16	BASE_REG2[3:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15…12	-		Зарезервировано
11…8	BASE_REG1[3:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7…4	-		Зарезервировано
3…0	BASE_REG0[3:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

7.2.3.30 **Регистр PLLn_STAT**

Base ADDR=				0x4000_0000				Offset=	0x0000_0088 0x0000_009C 0x0000_00B0						
REG Name:				PLLn_STAT											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

											PLLn_RDY	EVENT3	EVENT2	EVENT1	EVENT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–			MAX_CHK_SHIFT_REG1[4:0]					–			MAX_CHK_SHIFT_REG0[4:0]				

Таблица 54 – Описание бит регистра PLLn_STAT

Бит	Имя	Значение	Описание
31…21	-		Зарезервировано
20	PLLn_READY	-	Флаг готовности генератора PLLn: 0 – генератор не готов или выключен; 1 – генератор готов
19	EVENT3		Флаг возникновения события EVENT3 (очень высокая частота): 0 – нет события; 1 – есть событие
18	EVENT2		Флаг возникновения события EVENT2 (высокая частота): 0 – нет события; 1 – есть событие
17	EVENT1		Флаг возникновения события EVENT1 (низкая частота): 0 – нет события; 1 – есть событие
16	EVENT0		Флаг возникновения события EVENT0 (нет частоты): 0 – нет события; 1 – есть событие
15…13	–		Зарезервировано
12…8	MAX_CHK_ SHIFT_REG1[4:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7…5	–		Зарезервировано
4…0	MAX_CHK_ SHIFT_REG0[4:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

7.2.3.31 **Регистр CANn_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_0118							
REG Name:			CANn_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 55 – Описание бит регистра CANn_CLK

Бит	Имя	Значение	Описание
31…17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты CANCLKn: 0 – нет частоты; 1 – разрешена выдача частоты
15…0	DIV[15:0]		Делитель частоты MAX_CLK для формирования частоты CANCLKn: CANCLKn = MAX_CLK/(DIV+1)

7.2.3.32 **Регистр TIMn_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_0130 0x0000_0134 0x0000_0138 0x0000_013C							
REG Name:			TIMn_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 56 – Описание бит регистра TIMn_CLK

Бит	Имя	Значение	Описание
31…17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для TIMCLKn: 0 – нет частоты; 1 – разрешена выдача частоты
15…0	DIV[15:0]		Делитель частоты MAX_CLK для формирования частоты TIMCLKn: TIMCLKn = MAX_CLK/(DIV+1)
Примечание – Частота тактирования таймера TIMCLKn должна быть меньше или равна частоте тактирования ядра CPU_CLK			

7.2.3.33 **Регистр EPHY_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_0170							
REG Name:			EPHY_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]*				-										DIV2	EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 57 – Описание бит регистра EPHY_CLK

Бит	Имя	Значение	Описание
31…28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK: 4’b 00000 – Генератор HSI; 4’b 00001 – Генератор HSE0; 4’b 00010 – Генератор HSE1; 4’b 00011 – Генератор LSI; 4’b 00100 – Генератор LSE; 4’b 00101 – Генератор PLL0; 4’b 00110 – Генератор PLL1; 4’b 00111 – Генератор PLL2; 4’b 01101 – MAX_CLK
27…18	-	-	Зарезервировано
17	DIV2	0	Дополнительный делитель частоты SELCLK для формирования частоты EPHYCLK после делителя DIV[15:0]: 0 – деление на 2 не осуществляется; 1 – осуществляется деление на 2 после делителя DIV[15:0]. Коэффициент заполнения частоты после делителя DIV2 составляет 50%
16	EN_CLK		Разрешение формирования тактовой частоты для EthernetPHY: 0 – нет частоты; 1 – разрешена выдача частоты
15…0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты EPHYCLK: EPHYCLK = SELCLK/(DIV+1)
* См. примечание регистра RTC_CLK			

7.2.3.34 **Регистр SPW0_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_0178							
REG Name:			SPW0_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 58 – Описание бит регистра SPW0_CLK

Бит	Имя	Значение	Описание
31…17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для SPWCLK0: 0 – нет частоты; 1 – разрешена выдача частоты
15…0	DIV[15:0]		Делитель частоты MAX_CLK для формирования частоты SPWCLK0: SPWCLK0 = MAX_CLK/(DIV+1)

7.2.3.35 **Регистр UARTn_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_0188 0x0000_018C							
REG Name:			UARTn_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]*														DIV2	EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 59 – Описание бит регистра UARTn_CLK

Бит	Имя	Значение	Описание
31…28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK: 4’b 00000 – Генератор HSI; 4’b 00001 – Генератор HSE0; 4’b 00010 – Генератор HSE1; 4’b 00011 – Генератор LSI; 4’b 00100 – Генератор LSE; 4’b 00101 – Генератор PLL0; 4’b 00110 – Генератор PLL1; 4’b 00111 – Генератор PLL2; 4’b 01101 – MAX_CLK
27…18	-	-	Зарезервировано
17	DIV2	0	Дополнительный делитель частоты SELCLK для формирования частоты UARTCLKn после делителя DIV[15:0]: 0 – деление на 2 не осуществляется; 1 – осуществляется деление на 2 после делителя DIV[15:0]. Коэффициент заполнения частоты после делителя DIV2 составляет 50%

Бит	Имя	Значение	Описание
16	EN_CLK		Разрешение формирования тактовой частоты для UARTCLKn: 0 – нет частоты; 1 – разрешена выдача частоты
15…0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты UARTCLKn: UARTCLKn = SELCLK/(DIV+1)
* См. примечание регистра RTC_CLK			

7.2.3.36 **Регистр SSP0_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_01A0							
REG Name:			SSP0_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]*														DIV2	EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 60 – Описание бит регистра SSP0_CLK

Бит	Имя	Значение	Описание
31…28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK: 4’b 00000 – Генератор HSI; 4’b 00001 – Генератор HSE0; 4’b 00010 – Генератор HSE1; 4’b 00011 – Генератор LSI; 4’b 00100 – Генератор LSE; 4’b 00101 – Генератор PLL0; 4’b 00110 – Генератор PLL1; 4’b 00111 – Генератор PLL2; 4’b 01101 – MAX_CLK
27…18	-	-	Зарезервировано
17	DIV2	0	Дополнительный делитель частоты SELCLK для формирования частоты SSPCLK0 после делителя DIV[15:0]: 0 – деление на 2 не осуществляется; 1 – осуществляется деление на 2 после делителя DIV[15:0]. Коэффициент заполнения частоты после делителя DIV2 составляет 50%
16	EN_CLK		Разрешение формирования тактовой частоты для SSPCLK0: 0 – нет частоты; 1 – разрешена выдача частоты

Бит	Имя	Значение	Описание
15…0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты SSPCLK0: SSPCLK0 = SELCLK/(DIV+1)
* См. примечание регистра RTC_CLK			

7.2.3.37 **Регистр ADCn_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_01BC 0x0000_01C0							
REG Name:			ADCn_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]*														DIV2	EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 61 – Описание бит регистра ADCn_CLK

Бит	Имя	Значение	Описание
31…28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK: 4’b 00000 – Генератор HSI; 4’b 00001 – Генератор HSE0; 4’b 00010 – Генератор HSE1; 4’b 00011 – Генератор LSI; 4’b 00100 – Генератор LSE; 4’b 00101 – Генератор PLL0; 4’b 00110 – Генератор PLL1; 4’b 00111 – Генератор PLL2; 4’b 01101 – MAX_CLK
27…18	-	-	Зарезервировано
17	DIV2	0	Дополнительный делитель частоты SELCLK для формирования частоты ADCCLKn после делителя DIV[15:0]: 0 – деление на 2 не осуществляется; 1 – осуществляется деление на 2 после делителя DIV[15:0]. Коэффициент заполнения частоты после делителя DIV2 составляет 50%. Если частота SELCLK выше предельно-допустимой f_{C_ADC}, необходимо устанавливать бит DIV2 в 1 вне зависимости от выбранного значения поля DIV

Бит	Имя	Значение	Описание
16	EN_CLK		Разрешение формирования тактовой частоты для ADCCLKn: 0 – нет частоты; 1 – разрешена выдача частоты
15…0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты ADCCLKn: ADCCLKn = SELCLK/(DIV+1). Деление частоты выполняется путем «прореживания» тактовых импульсов. Необходимо учитывать требование к минимальной длительности тактового импульса
* См. примечание регистра RTC_CLK			

7.2.3.38 **Регистр DACn_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_01C4 0x0000_01C8							
REG Name:			DACn_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]*														DIV2	EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 62 – Описание бит регистра DACn_CLK

Бит	Имя	Значение	Описание
31…28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK: 4’b 00000 – Генератор HSI; 4’b 00001 – Генератор HSE0; 4’b 00010 – Генератор HSE1; 4’b 00011 – Генератор LSI; 4’b 00100 – Генератор LSE; 4’b 00101 – Генератор PLL0; 4’b 00110 – Генератор PLL1; 4’b 00111 – Генератор PLL2; 4’b 01101 – MAX_CLK
27…18	-	-	Зарезервировано

Бит	Имя	Значение	Описание
17	DIV2	0	Дополнительный делитель частоты SELCLK для формирования частоты DACCLKn после делителя DIV[15:0]: 0 – деление на 2 не осуществляется; 1 – осуществляется деление на 2 после делителя DIV[15:0]. Коэффициент заполнения частоты после делителя DIV2 составляет 50%
16	EN_CLK		Разрешение формирования тактовой частоты для DACCLKn: 0 – нет частоты; 1 – разрешена выдача частоты
15…0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты DACCLKn: DACCLKn = SELCLK/(DIV+1)
* См. примечание регистра RTC_CLK			

7.2.3.39 **Регистр RTC_CLK**

Base ADDR=			0x4000_0000			Offset=		0x0000_01CC							
REG Name:			RTC_CLK												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]*														DIV2	EN_CLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

Таблица 63 – Описание бит регистра RTC_CLK

Бит	Имя	Значение	Описание
31…28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK: 4’b 00000 – Генератор HSI; 4’b 00001 – Генератор HSE0; 4’b 00010 – Генератор HSE1; 4’b 00011 – Генератор LSI; 4’b 00100 – Генератор LSE; 4’b 00101 – Генератор PLL0; 4’b 00110 – Генератор PLL1; 4’b 00111 – Генератор PLL2; 4’b 01101 – MAX_CLK
27…18	-	-	Зарезервировано

Бит	Имя	Значение	Описание
17	DIV2	0	Дополнительный делитель частоты SELCLK для формирования частоты RTCCLK после делителя DIV[15:0]: 0 – деление на 2 не осуществляется; 1 – осуществляется деление на 2 после делителя DIV[15:0]. Коэффициент заполнения частоты после делителя DIV2 составляет 50%
16	EN_CLK		Разрешение формирования тактовой частоты для RTCCLK: 0 – нет частоты; 1 – разрешена выдача частоты
15…0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты RTCCLK: RTCCLK = SELCLK/(DIV+1)
* Переключение частоты в регистрах EPHY_CLK, UARTn_CLK, SSP0_CLK, ADCn_CLK, DACn_CLK, RTC_CLK необходимо выполнять с включенного источника			

7.3 Контроллер батарейного домена (BKPCNTR)

В микросхеме реализован батарейный домен, включающий в себя:

– 240 байт пользовательской памяти BKP_MEM,
– четыре регистра управления,
– часы реального времени RTC,
– генератор LSE.

Регистры управления BKP_MEM предназначены для хранения и управления режимами работы микросхемы, в том числе управление питанием и т.п. При отсутствии питания U_{CC} питание батарейного домена автоматически переключается на питание от вывода BU_{CC}, и также автоматически переключается на питание U_{CC} при его появлении.

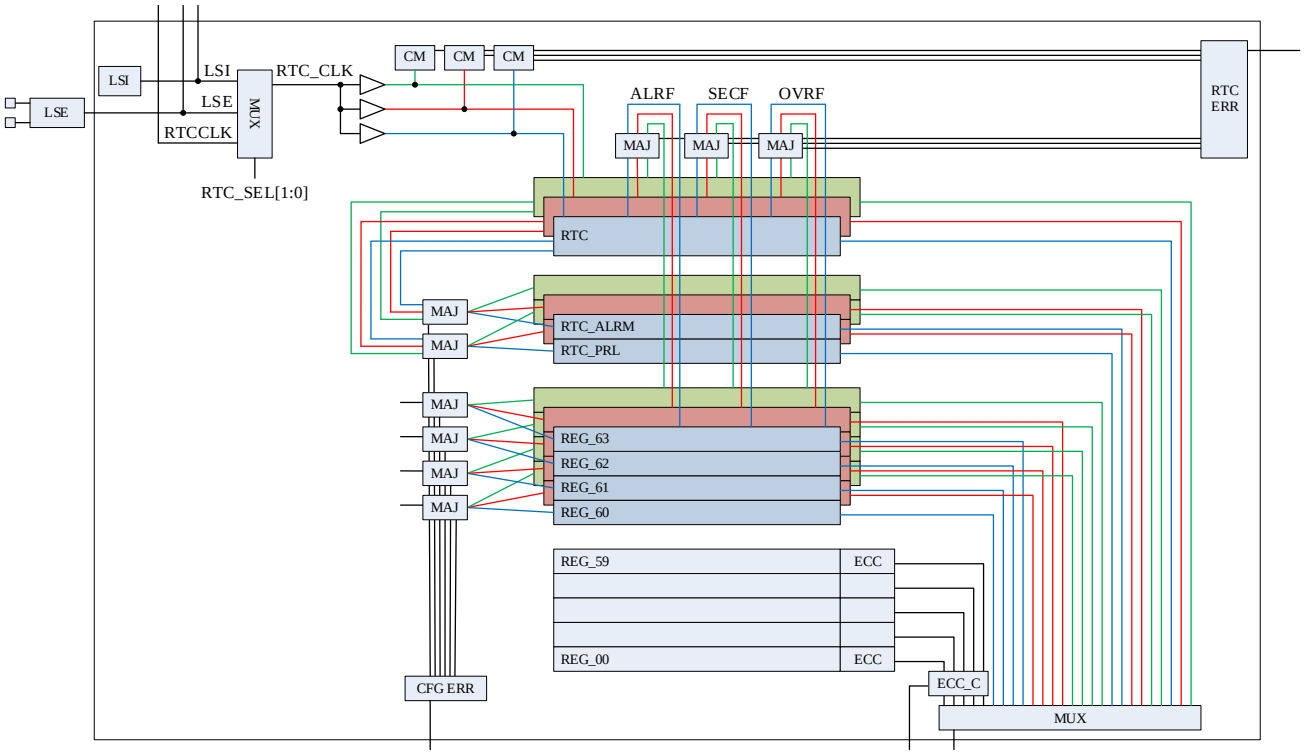


Рисунок 32 – Блок-схема контроллера батарейного домена

В батарейном домене применяются различные механизмы обнаружения и исправления сбоев. Пользовательская память использует биты ECC для автоматического исправления одиночных ошибок и обнаружения двойных. Основная конфигурация и часы реального времени выполнены на троированной логике с непрерывным контролем возникновения ошибок в схемах мажорирования. Мажорирование не исправляет сбоев непосредственно в том канале, в котором он возник, а исправляет выход схемы, влияющий на поведение микросхемы в целом. Таким образом, при возникновении сбоя в одном из каналов схема мажорирования исправит его влияние на систему в целом и будет его обозначать флагом ошибки, вплоть до его исправления программными средствами. Кроме того, для некоторых битов управления, например, STANDBY, действует не мажорированная логика (два из трех), а логика три из трех. То есть для того, чтобы перевести микросхему в режим STANDBY необходимо, чтобы бит STANDBY был взведен во всех трех каналах. Если хотя бы в одном из регистров бит не будет взведен

или ошибочно сброшен в ноль, то микросхема автоматически выходит из режима STANDBY.

Примечание – Переход в режим Standby возможен только, если FILTER_EN=1.

7.3.1 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 RTC_CLK часов реального времени могут выступать генераторы LSI или LSE, либо сигнал RTCCLK, который формируется в блоке управления тактовыми частотами и может быть выбран только при наличии питания DU_{CC}. LSI может быть выбран при наличии питания U_{CC}, LSE может быть выбран при наличии U_{CC} или BU_{CC}. Выбор между источниками осуществляется битами RTC_SEL. При возможном отключении основного источника питания U_{CC} в качестве источника тактовой частоты RTC_CLK должен использоваться осциллятор LSE, так как он имеет батарейное питание. Биты управления осциллятором LSE расположены в батарейном домене и, таким образом, при отключении основного питания они не сбрасываются.

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 0100 в документе К1986ВЕ8Т Errata Notice). Значение RTC_CAL[7:0] определяет, какое число тактов RTC_CLK из 2^{20} будет замаскировано с помощью дополнительного счётчика 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. 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. Бит STANDBY, отключающий все внутренние регуляторы напряжения, автоматически сбрасывается при превышении RTC_CNT значения в регистре RTC_ALRM.

Процесс выполнения записи регистра RTC_CNT 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 регистр RTC_CNT (во все регистры RTC_CNT_TMR1-3) запрещена.

7.3.2 Описание регистров

Таблица 64 – Описание регистров

Базовый адрес	Название	Описание
0x4000_1000	MRD_BKP	Контроллер батарейного домена и часов реального времени
Смещение		
0x0000_00F0	KEY	Регистр ключа, разрешающего модификацию остальных регистров
Память батарейного домена		
0x0000_0000	REG_00	Регистр батарейной памяти 00
0x0000_0004	REG_01	Регистр батарейной памяти 01
0x0000_00EC	REG_59	Регистр батарейной памяти 59
0x0000_0100	REG_60_TMR0	Регистр батарейной памяти 60 с функциями управления
0x0000_0104	REG_61_TMR0	Регистр батарейной памяти 61 с функциями управления
0x0000_0108	REG_62_TMR0	Регистр батарейной памяти 62 с функциями управления
0x0000_010C	REG_63_TMR0	Регистр батарейной памяти 63 с функциями управления
0x0000_0110	REG_60_TMR1	Регистр батарейной памяти 60 с функциями управления
0x0000_0114	REG_61_TMR1	Регистр батарейной памяти 61 с функциями управления
0x0000_0118	REG_62_TMR1	Регистр батарейной памяти 62 с функциями управления
0x0000_011C	REG_63_TMR1	Регистр батарейной памяти 63 с функциями управления
0x0000_0120	REG_60_TMR2	Регистр батарейной памяти 60 с функциями управления
0x0000_0124	REG_61_TMR2	Регистр батарейной памяти 61 с функциями управления
0x0000_0128	REG_62_TMR2	Регистр батарейной памяти 62 с функциями управления
0x0000_012C	REG_63_TMR2	Регистр батарейной памяти 63 с функциями управления
Регистры часов реального времени		
0x0000_0130	RTC_CNT_TMR0	Регистр основного счетчика часов реального времени
0x0000_0134	RTC_DIV_TMR0	Регистр счетчика предварительного делителя
0x0000_0138	RTC_PRL_TMR0	Регистр коэффициента деления предварительного делителя
0x0000_013C	RTC_ALRM_TMR0	Регистр значения для сравнения с основным счетчиком RTC_CNT и выработки сигнала ALRF
0x0000_0140	RTC_CS_TMR0	Регистр управления и состояния часов реального времени
0x0000_0150	RTC_CNT_TMR1	Регистр основного счетчика часов реального времени
0x0000_0154	RTC_DIV_TMR1	Регистр счетчика предварительного делителя

Базовый адрес	Название	Описание
0x0000_0158	RTC_PRL_TMR1	Регистр коэффициента деления предварительного делителя
0x0000_015C	RTC_ALRM_TMR1	Регистр значения для сравнения с основным счетчиком RTC_CNT и выработки сигнала ALRF
0x0000_0160	RTC_CS_TMR1	Регистр управления и состояния часов реального времени
0x0000_0170	RTC_CNT_TMR2	Регистр основного счетчика часов реального времени
0x0000_0174	RTC_DIV_TMR2	Регистр счетчика предварительного делителя
0x0000_0178	RTC_PRL_TMR2	Регистр коэффициента деления предварительного делителя
0x0000_017C	RTC_ALRM_TMR2	Регистр значения для сравнения с основным счетчиком RTC_CNT и выработки сигнала ALRF
0x0000_0180	RTC_CS_TMR2	Регистр управления и состояния часов реального времени

7.3.2.1 **Регистр KEY**

Base ADDR=			0x4000_1000			Offset=		0x0000_00F0							
REG Name:			KEY												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 65 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока BKP_CNTR

7.3.2.2 **REG_00…REG_59**

Base ADDR=			0x4000_1000		Offset=		0x0000_0000 0x0000_0004 .. 0x0000_00EC								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Таблица 66 – Описание бит регистров REG_00…REG_59

Бит	Имя	Значение	Описание
31…0	DATA[31:0]		Данные памяти батарейного домена

7.3.2.3 **REG_60_TMRx**

Base ADDR=			0x4000_1000		Offset=		0x0000_0100 0x0000_0110 0x0000_0120								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR_IE	CLR_ERR	OVRSTnDSBL	PORSTnDSBL	ERR_63	ERR_LDO	ERR_60	-	DUCCLIM_EN[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISABLE_JTAG	S_SYSRST	S_OVRSTn	S_RESET1n	S_WDG_RESET	S_FT_RESET	S_RESET0n	S_UccRESET	FPOR	MODE[6:0]						

Таблица 67 – Описание бит регистров REG_60_TMRx

Бит	Имя	Значение	Описание
31	ERR_IE	0	Разрешение прерывания BKP_IF по ошибкам ERR: 0 –прерывание запрещено; 1 – прерывание разрешено. Сигнал формируется по правилам TMR
30	CLR_ERR	0	Бит сброса флагов ошибки: 0 – нет эффекта; 1 – сброс флагов ERR_63, ERR_LDO, ERR_60. Флаги не сбрасываются, если в момент их сброса событие сбоя продолжается. Сигнал сброса вырабатывается, если во всех трех регистрах TRMx записаны единицы. После сброса флагов ошибок бит CLR_ERR необходимо программно сбросить в 0

Бит	Имя	Значение	Описание
29	OVRSTnDSBL	1	Флаг запрещения сброса по превышению питания OVRSTn: 0 – схема сброса работает; 1 – схема сброса заблокирована. Схема POVR выключается, если во всех трех каналах она заблокирована, если хотя бы в одном канала она разрешена, то она работает
28	PORSTnDSBL	0	Флаг запрещения сброса по превышению питания PORSTn: 0 – схема сброса работает; 1 – схема сброса заблокирована. Схема POR выключается, если во всех трех каналах она заблокирована, если хотя бы в одном канала она разрешена, то она работает
27	ERR_63	0	Сбой конфигурации регистров REG_63: 0 – нет сбоя; 1 – есть сбой
26	ERR_LDO	0	Сбой конфигурации регистров REG_61 и REG_62: 0 – нет сбоя; 1 – есть сбой
25	ERR_60	0	Сбой конфигурации регистров REG_60: 0 – нет сбоя; 1 – есть сбой
24	-	1	Зарезервировано Взводится в 1 при любом сбросе
23…16	DUCCLIM_EN[7:0]	00000000	Флаг разрешения работы ограничения тока в доменах DU_{CC}: DUCCLIM_EN[0] – LDO0 основного питания DU_{CC}0,1,2; DUCCLIM_EN[1] – LDO1 основного питания DU_{CC}0,1,2; DUCCLIM_EN[2] – LDO2 основного питания DU_{CC}0,1,2; DUCCLIM_EN[3] – LDO3 основного питания DU_{CC}0,1,2. 0 – есть ограничение по току (~300 мА на один LDO); 1 – нет ограничения по току. DUCCLIM_EN[4] – LDO батарейного домена DU_{CC}_B; DUCCLIM_EN[5] – LDO умножителей частоты DU_{CC}_PLL. 0 – есть ограничение по току (~20 мА на один LDO); 1 – нет ограничения по току

Бит	Имя	Значение	Описание
15	DISABLE_JTAG	0	Запрещение отладочного интерфейса JTAG: 0 – JTAG_A и/или JTAG_B разрешены. Значение при сбросе; 1– JTAG_A и/или JTAG_B запрещены
14	S_SYSRSTn	x*	Флаг события сброса с вывода SYSRSTn. Данный флаг устанавливается, если был установлен любой из флагов события сброса. См. подраздел 6.3 «Сигналы Сброса»: 0 – не было сброса по SYSRSTn; 1 – был сброс по SYSRSTn. Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
13	S_OVRSTn	0	Флаг события сброса с вывода OVRSTn: 0 – не было сброса по OVRSTn; 1 – был сброс по OVRSTn. Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
12	S_RESET1n	x*	Флаг события сброса с вывода RSTn1: 0 – не было сброса по RSTn1; 1 – был сброс по RSTn1. Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
11	S_WDG_RESET	0	Флаг события сброса с вывода WDG_RESET: 0 – не было сброса по WDG_RESET; 1 – был сброс по WDG_RESET. Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
10	S_FT_RESET	0	Флаг события сброса с вывода FT_RESET: 0 – не было сброса по FT_RESET; 1 – был сброс по FT_RESET. Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
9	S_RESET0n	x*	Флаг события сброса с вывода RSTn0: 0 – не было сброса по RSTn0; 1 – был сброс по RSTn0. Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен

Бит	Имя	Значение	Описание
8	S_U_{CC}RESET	x*	Флаг события сброса U_{CC}RESET: 0 – не было сброса по U_{CC}RESET; 1 – был сброс по U_{CC}RESET. Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
7	FPOR	0	Флаг успешного завершения работы загрузчика: 0 – не передано управление основному ПО; 1 – передано управление основному ПО. Следует отметить, что данный флаг взводится загрузчиком до момента фактической передачи управления (не превышает 150 мкс). Возможна только запись 1. Сброс по любому сигналу сброса
6…0	MODE[6:0]	xxxxxx	Режим запуска микросхемы с выводов {MODE[7:5], MODE[3:0]} (с исключенным битом четности). См. подраздел 6.4 «Режимы работы микросхемы». JTAG_A = (MODE[0]==0) & ~DISABLE_JTAG; JTAG_B = (MODE[0]==1) & ~DISABLE_JTAG; JTAG_EN = (MODE[3:0]==4'b1111)
* Зависит от порядка подачи напряжения питаний и внешних сигналов сброса			

7.3.2.4 **REG_61_TMRx**

Base ADDR=		0x4000_1000			Offset=		0x0000_0104 0x0000_0114 0x0000_0124								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DUCC3_ RDY	EN_DUCC3	TRIM3[2:0]			SRILOW3 [2:0]*			DUCC2_ RDY	EN_DUCC0	TRIM2[2:0]			SRILOW2 [2:0]*		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUCC1_ RDY	EN_DUCC1	TRIM1[2:0]			SRILOW1 [2:0]*			DUCC0_ RDY	EN_DUCC0	TRIM0[2:0]			SRILOW0 [2:0]*		

Таблица 68 – Описание бит регистров REG_61_TMRx

Бит	Имя	Значение	Описание
31	DUCC3_RDY		Флаг готовности напряжения регулятора LDO3: 0 – регулятор LDO3 не готов; 1 – регулятор LDO3 в норме
30	EN_DUCC3		Аналогично SRILOW0, TRIM0 и EN_DUCC0 для регулятора LDO3
29…27	TRIM3[2:0]		
26…24	SRILOW3[2:0]*		
23	DUCC2_RDY		Флаг готовности напряжения регулятора LDO2: 0 – регулятор LDO2 не готов; 1 – регулятор LDO2 в норме
22	EN_DUCC2		Аналогично SRILOW0, TRIM0 и EN_DUCC0 для регулятора LDO2
21…19	TRIM2[2:0]		
18…16	SRILOW2[2:0]*		
15	DUCC1_RDY		Флаг готовности напряжения регулятора LDO1: 0 – регулятор LDO1 не готов; 1 – регулятор LDO1 в норме
14	EN_DUCC1		Аналогично SRILOW0, TRIM0 и EN_DUCC0 для регулятора LDO1
13…11	TRIM1[2:0]		
10…8	SRILOW1[2:0]*		
7	DUCC0_RDY		Флаг готовности напряжения любого из регуляторов LDO0 – LDO3: 0 – все регуляторы LDO0 – LDO3 не готовы; 1 – один и более из регуляторов LDO0 – LDO3 находится в рабочем режиме
6	EN_DUCC0		Разрешение выдачи напряжения регулятором LDO0: 0 – разрешена работа регулятора LDO0; 1 – запрещена работа регулятора LDO0. Бит предустанавливается сигналами WAKEUP, ALRF, UCCRESET. Схема LDO выключается, если во всех трех каналах она заблокирована, если хотя бы в одном канала она разрешена, то она работает
5…3	TRIM0[2:0]		Подстройка выходного напряжение DU_{CC} регулятора LDO0: 3’b0 000 = +0,01В (типовое значение); 3’b0 001 = +0,05В; 3’b0 010 = +0,10В; 3’b0 011 = +0,15В; 3’b0 100 = - 0,01В; 3’b0 101 = - 0,05В; 3’b0 110 = - 0,10В; 3’b0 111 = - 0,15В. Сигнал формируется по правилам TMR

Бит	Имя	Значение	Описание
2…0	SRILOW0[2:0]*		Настройка режима работы регулятора LDO0: 3’b0 011 – Частота до 100 МГц (доп. ток ~650 мкА); 3’b0 010 – Частота до 50 МГц (доп. ток ~550 мкА); 3’b0 001 – Частота до 30 МГц (доп. ток ~450мкА); 3’b0 000 – Частота до 10 МГц (доп. ток ~400 мкА) (типовое значение); 3’b0 100 – Частота до 1МГц (доп. ток ~300мкА); 3’b0 101 – Частота до 500 кГц (доп. ток ~200 мкА); 3’b0 110 – Частота до 100 кГц (доп. ток ~100мкА); 3’b0 111 – Частота более 10 кГц (доп. ток ~50 мкА); Сигнал формируется по правилам TMR
* Рекомендуется выбирать значение 111 при работе на частотах выше 40 МГц для случаев, когда микросхема установлена в контактное устройство			

7.3.2.5 **REG_62_TMRx**

Base ADDR=			0x4000_1000			Offset=		0x0000_0108 0x0000_0118 0x0000_0128							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	TRIM_REF[2:0]		-				TRIMR[6]	-	TRIMR[7]	TRIMR[5:3]			TRIMR[2:0]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUCC5_RDY	EN_DUCC5	TRIM5 [2:0]			SRILOW5 [2:0]*			DUCC4_RDY	EN_DUCC4	TRIM4 [2:0]			SRILOW4 [2:0]*		

Таблица 69 – Описание бит регистров REG_62_TMRx

Бит	Имя	Значение	Описание
31	-		Зарезервировано
30…28	TRIM_REF[2:0]		TRIM_REF[2:0] – подстройка источника опорного напряжения LDO: 100 – типовое значение. Рекомендуется не изменять типовое значение
27…25	-		Зарезервировано
24	TRIMR[6]		TRIMR[7:0] – подстройка источника опорного тока LDO: 10000000 – типовое значение. Рекомендуется не изменять типовое значение
23	-		
22	TRIMR[7]		
21…19	TRIMR[5:3]		
18…16	TRIMR[2:0]		
15	DUCC5_RDY		Флаг готовности напряжения для домена $DU_{CC}B$: 0 – нет напряжения DU_{CC} в домене $DU_{CC}B$; 1 – есть напряжение DU_{CC} в домене $DU_{CC}B$

Бит	Имя	Значение	Описание
14	EN_DUCC5		0 – выдавать напряжение в домен DU_{CC}_PLL; 1 – не выдавать напряжение в домен DU_{CC}_PLL
13…11	TRIM5[2:0]		Подстройка напряжение DU_{CC} в домене DU_{CC}_PLL: 3’b0 000 = +0,01В (типовое значение); 3’b0 001 = +0,05В; 3’b0 010 = +0,10В; 3’b0 011 = +0,15В; 3’b0 100 = - 0,01В; 3’b0 101 = - 0,05В; 3’b0 110 = - 0,10В; 3’b0 111 = - 0,15В. Сигнал формируется по правилам TMR
10…8	SRILOW5[2:0]*		Настройка режима работы домена DU_{CC}_PLL: 3’b0 011 – Частота до 100 МГц (доп. ток ~650 мкА); 3’b0 010 – Частота до 50 МГц (доп. ток ~550 мкА); 3’b0 001 – Частота до 30 МГц (доп. ток ~450 мкА); 3’b0 000 – Частота до 10 МГц (доп. ток ~400 мкА) (типовое значение); 3’b0 100 – Частота до 1МГц (доп. ток ~300 мкА); 3’b0 101 – Частота до 500 кГц (доп. ток ~200 мкА); 3’b0 110 – Частота до 100 кГц (доп. ток ~100 мкА); 3’b0 111 – Частота более 10 кГц (доп. ток ~50 мкА); Сигнал формируется по правилам TMR
7	DUCC4_RDY		Флаг готовности напряжения для домена DU_{CC}_PLL: 0 – нет напряжения DU_{CC} в домене DU_{CC}_PLL; 1 – есть напряжение DU_{CC} в домене DU_{CC}_PLL
6	EN_DUCC4		Не имеет значения, батарейный домен DU_{CC}B всегда включен. Выключен может быть только сигналом SHDN. Значение данного бита должно совпадать во всех регистрах
5…3	TRIM4[2:0]		Подстройка напряжение DU_{CC} в домене DU_{CC}B: 3’b0 000 = +0,01В (типовое значение); 3’b0 001 = +0,05В; 3’b0 010 = +0,10В; 3’b0 011 = +0,15В; 3’b0 100 = - 0,01В; 3’b0 101 = - 0,05В; 3’b0 110 = - 0,10В; 3’b0 111 = - 0,15В. Сигнал формируется по правилам TMR

Бит	Имя	Значение	Описание
2…0	SRILOW4[2:0]*		Настройка режима работы домена $DU_{CC}B$: 3’b0 011 – Частота до 100 МГц (доп. ток ~650 мкА); 3’b0 010 – Частота до 50 МГц (доп. ток ~550 мкА); 3’b0 001 – Частота до 30 МГц (доп. ток ~450 мкА); 3’b0 000 – Частота до 10 МГц (доп. ток ~400 мкА) (типовое значение); 3’b0 100 – Частота до 1МГц (доп. ток ~300 мкА); 3’b0 101 – Частота до 500 кГц (доп. ток ~200 мкА); 3’b0 110 – Частота до 100 кГц (доп. ток ~100 мкА); 3’b0 111 – Частота более 10 кГц (доп. ток ~50 мкА). Сигнал формируется по правилам TMR
* Рекомендуется выбирать значение 111 при работе на частотах выше 40 МГц для случаев, когда микросхема установлена в контактное устройство			

7.3.2.6 **REG_63_TMRx**

Base ADDR=		0x4000_1000			Offset=		0x0000_010C 0x0000_011C 0x0000_012C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC_RST	STANDBY	HSI_TRIM[5:0]						HSI_RDY	HSI_ON	LSI_RDY	LSI_TRIM[4:0]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSI_ON	FILTER_EN	LSE_RDY	RTC_CAL[7:0]								RTC_EN	RTC_SEL[1:0]		LSE_BYP	LSE_ON

Таблица 70 – Описание бит регистров REG_63_TMRx

Бит	Имя	Значение	Описание
31	RTC_RST	1	Сброс часов реального времени: 0 – часы не сбрасываются; 1 – часы сбрасываются. Сигнал формируется по правилам TMR
30	STANDBY	0	Режим отключения всех регуляторов DUCC: 0 – регуляторы включены и выдают напряжение; 1 – регулятор выключен Триггер сбрасывается флагом ALRF или по низкому уровню на выводе WAKEUP и сигналом сброса BDUCC. Схемы LDO выключаются, если во всех трех каналах они выключены, если хотя бы в одном канале они разрешены, то работают

Бит	Имя	Значение	Описание
29…24	HSI_TRIM[5:0]	32	Коэффициент подстройки частоты генератора HSI. 0 – минимальная частота; … 32 – типовое значение частоты; … 63 – максимальна частота. Сигнал формируется по правилам TMR
23	HSI_RDY	1	Флаг выхода генератора HSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
22	HSI_ON	1	Бит управления генератором HSI: 0 – генератор выключен; 1 – генератор включен. Схемы HSI выключаются, если во всех трех каналах она выключена, если хотя бы в одном канале она разрешена, то работает
21	LSI_RDY	1	Флаг выхода генератора LSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор находится в рабочем режиме
20…16	LSI_TRIM[4:0]	32	Коэффициент подстройки частоты генератора LSI: 0 – максимальная частота; … 8 – типовое значение частоты; … 15 и более – минимальная частота. Сигнал формируется по правилам TMR
15	LSI_ON	1	Бит управления генератором LSI: 0 – генератор выключен; 1 – генератор включен. Схемы LSI выключаются, если во всех трех каналах она выключена, если хотя бы в одном канале она разрешена, то работает
14	FILTER_EN	1	Фильтрация входных сигналов генератора LSE: 0 – фильтр выключен (переход в режим Standby невозможен); 1 – фильтр включен
13	LSE_RDY	0	Флаг выхода генератора LSE в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме

Бит	Имя	Значение	Описание
12…5	RTC_CAL[7:0]	2	Коэффициент подстройки тактовой частоты часов реального времени RTC_CLK. Из каждых 2^{20} тактов RTC_CLK будет замаскировано RTC_CAL тактов: 00000000 – 0 тактов; 00000001 – 1 такт; … 11111111 – 255 тактов. Таким образом, если исходная частота RTC_CLK равна 32768,00000 Гц, то после подстройки средняя частота тактирования составляет: RTC_CLK_C = 32768,00000 Гц при RTC_CAL = 0; RTC_CLK_C = 32767,96875 Гц при RTC_CAL = 1; … RTC_CLK_C = 32760,03125 Гц при RTC_CAL = 255. В общем случае средняя тактовая частота RTC_CLK после подстройки составляет: RTC_CLK_C = RTC_CLK · (1 – RTC_CAL[7:0] / 2^{20}). Единица подстройки (замедления) равна $1/2^{20}$, или ~0,95 ppm. Сигнал формируется по правилам TMR
4	RTC_EN	0	Бит разрешения работы часов реального времени: 0 – работа запрещена; 1 – работа разрешена. После записи задержка включения/выключения RTC составляет 2 такта частоты RTC_CLK. Сигнал формируется по правилам TMR
3, 2	RTC_SEL[1:0]	00	Биты выбора источника тактовой синхронизации часов реального времени RTC_CLK: 00 – LSI; 01 – LSE; 10 – RTCCLK (формируется в блоке CLKCNTR); 11 – зарезервировано. Сигнал формируется по правилам TMR. Для переключения частоты RTC_CLK с одного источника на другой необходимо, чтобы оба источника частоты были включены и находились в рабочем режиме. Валидная частота RTC_CLK формируется после 3 периодов частоты, с которой происходит переключение, плюс 3 периода частоты, на которую происходит переключение. Использование частоты RTC_CLK возможно только после появления валидной частоты. До формирования валидной частоты RTC_CLK работа RTC должна быть запрещена или же RTC должен находиться в сбросе
1	LSE_BYP	0	Бит управления генератором LSE: 0 – режим работы с внешним резонатором; 1 – режим работы с внешним генератором. Устанавливать совместно с LSE_ON = 1. Сигнал формируется по правилам TMR

Бит	Имя	Значение	Описание
0	LSE_ON	0	Бит разрешения работы генератора LSE: 0 – генератор выключен; 1 – генератор включен. Сигнал формируется по правилам TMR

7.3.2.7 **RTC_CNT_TMRx**

Base ADDR=		0x4000_1000			Offset=		0x0000_0130 0x0000_0150 0x0000_0170								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC_CNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_CNT[15:0]															

Таблица 71 – Описание бит регистров RTC_CNT_TMRx

Бит	Имя	Значение	Описание
31…0	RTC_CNT[31:0]		Значение основного счетчика часов реального времени

7.3.2.8 **RTC_DIV_TMRx**

Base ADDR=		0x4000_1000			Offset=		0x0000_0134 0x0000_0154 0x0000_0174								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-												RTC_DIV[19:16]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_DIV[15:0]															

Таблица 72 – Описание бит регистров RTC_DIV_TMRx

Бит	Имя	Значение	Описание
31…20	-		Зарезервировано
19…0	RTC_DIV[19:0]		Значение счетчика предварительного делителя часов реального времени

7.3.2.9 **RTC_PRL_TMRx**

Base ADDR=		0x4000_1000			Offset=		0x0000_0138 0x0000_0158 0x0000_0178								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-												RTC_PRL[19:16]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_PRL[15:0]															

Таблица 73 – Описание бит регистров RTC_PRL_TMRx

Бит	Имя	Значение	Описание
31…20	-		Зарезервировано
19…0	RTC_PRL[19:0]		Коэффициент деления тактовой частоты RTC_CLK: SEC_CLK = RTC_CLK / (RTC_PRL + 1), где RTC_CLK – частота тактирования с учетом подстройки RTC_CAL[7:0]

7.3.2.10 **RTC_ALRM_TMRx**

Base ADDR=		0x4000_1000			Offset=		0x0000_013C 0x0000_015C 0x0000_017C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC_ALRM[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_ALRM[15:0]															

Таблица 74 – Описание бит регистров RTC_ALRM_TMRx

Бит	Имя	Значение	Описание
31…0	RTC_ALRM[31:0]		Значение для сравнения с основным счетчиком RTC_CNT и выработки сигнала ALRF. Сигнал ALRF вырабатывается в момент превышения основным счетчиком RTC_CNT значения в регистре RTC_ALRM RTC_CNT[31:0] == RTC_ALRM[31:0] + 1

7.3.2.11 **RTC_CS_TMRx**

Base ADDR=		0x4000_1000			Offset=		0x0000_0140 0x0000_0160 0x0000_0180								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	ERR_CMx	ERR_DIVx	ERR_CNTx	ERR_CC	ERR_CM	ERR_DIV	ERR_CNT	CLR_ERRx	WEC	ALRF_IE	SECF_IE	OWF_IE	ALRF	SECF	OWF

Таблица 75 – Описание бит регистров RTC_CS_TMRx

Бит	Имя	Значение	Описание
31…15	-		Зарезервировано
15	ERR_IE		Флаг разрешения прерывания RTC_IF по событиям сбоя: 0 – прерывание запрещено; 1 – прерывание разрешено
14	ERR_CMx		Сбой в данном канале тактовой частоты RTC: 0 – нет сбоя; 1 – есть сбой
13	ERR_RTCx		Сбой в данном канале DIV часов RTC: 0 – нет сбоя; 1 – есть сбой
12	ERR_CNTx		Сбой в данном канале CNT часов RTC: 0 – нет сбоя; 1 – есть сбой
11	ERR_CC		Отказ источника тактовой частоты RTC: 0 – нет сбоя; 1 – есть сбой. При отказе источника тактовой частоты, RTC автоматически переключается на генератор LSI
10	ERR_CM		Сбой в одном из каналов тактовой частоты RTC превышения максимальной частоты: 0 – нет сбоя; 1 – есть сбой
9	ERR_DIV		Сбой в одном из каналов DIV часов RTC: 0 – нет сбоя; 1 – есть сбой

Бит	Имя	Значение	Описание
8	ERR_CNT		Сбой в одном из каналов CNT часов RTC и конфигурации RTC: 0 – нет сбоя; 1 – есть сбой
7	CLR_ERRx		Бит сброса флагов ошибки в данном канале: 0 – нет сброса; 1 – сброс флагов ERR_RTCx, ERR_CNTx, ERR_CMx и сброса ошибок конфигурации RTC (кроме PRL и ALRM). Флаги не сбрасываются, если в момент их сброса событие сбоя продолжается. После сброса флагов ошибок бит CLR_ERRx необходимо программно сбросить в 0. Для сброса флага ERR_CC необходима запись единиц во все CLR_ERRx
6	WEC		Флаг выполнения записи в регистр RTC_CNT: 0 – можно записывать в регистр RTC_CNT; 1 – запись в регистр RTC_CNT (RTC_CNT_TMR1-3) запрещена, идет запись
5	ALRF_IE		Флаг разрешения прерывания RTC_IF по событию ALRF: 0 – прерывание запрещено; 1 – прерывание разрешено
4	SECF_IE		Флаг разрешения прерывания RTC_IF по событию SECF: 0 – прерывание запрещено; 1 – прерывание разрешено
3	OWF_IE		Флаг разрешения прерывания RTC_IF по событию OWF: 0 – прерывание запрещено; 1 – прерывание разрешено
2	ALRF		Флаг превышения основным счетчиком RTC_CNT значения RTC_ALRM: 0 – нет события превышения; 1 – было событие превышения. Флаг устанавливается при RTC_CNT[31:0] == RTC_ALRM[31:0] + 1. Сбрасывается записью 1
1	SECF		Флаг совпадения значений регистров RTC_DIV и RTC_PRL: 0 – не было совпадения; 1 – было совпадение. По событиям совпадения RTC_DIV и RTC_PRL формируется частота SEC_CLK. Таким образом флаг SECF устанавливается на каждом такте частоты SEC_CLK. Сбрасывается записью 1

Бит	Имя	Значение	Описание
0	OWF		Флаг переполнения основного счетчика RTC_CNT: 0 – нет переполнения; 1 – было переполнение. Сбрасывается записью 1

7.4 Контроллер детектора напряжения питания (PWRCNTR)

Блок детектора напряжений питания позволяет оценить напряжение внешних источников питания U_{CC} (питание IO, приемопередатчиков и т.п.), $U_{CC}1$ (питание LDO) и BU_{CC} (питание батарейного домена), а также определить ток потребления доменов питания DU_{CC}. Определение значения источников напряжения осуществляется путем их сравнения с источником опорного напряжения в блоке PVD и самих LDO. При этом обеспечиваемая точность измерения значения напряжения источников питания составляет ±100 мВ. Точность измерения токов потребления составляет ±5 мА для LDO0-LDO3 и ±0,5 мА для LDO батарейного домена и блока умножителей частоты PLL.

В блоке PWR_CNTR могут быть настроены условия для выработки событий превышения или снижение ниже других значений, при возникновении которых формируются прерывания, либо события для блока управления сбоями, ошибками и отказами (FT_CNTR), приводящими к сбросу микросхемы.

7.4.1 Описание регистров

Таблица 76 – Описание регистров блока детектора напряжений питания

Базовый адрес	Название	Состояние после сброса	Описание
0x4000_2000	MDR_PWR		Контроллер блока управления питанием
Смещение			
0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	CNTR0		Регистр управления PWR_CNTR0
0x0000_0008	CNTR1		Регистр управления PWR_CNTR1
0x0000_000C	CNTR2		Регистр управления PWR_CNTR2
0x0000_0010	STAT		Регистр статуса PWR_CNTR
0x0000_0014	ULIMIT		Регистр настройки ограничения напряжений PWR_CNTR
0x0000_0018	CLIMIT		Регистр настройки ограничения токов PWR_CNTR

7.4.1.1 **KEY**

Base ADDR=			0x4000_2000		Offset=		0x0000_0000								
REG Name:			KEY												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 77 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока PWR_CNTR

7.4.1.2 **CNTR0**

Base ADDR=			0x4000_2000		Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-							PDID_EN[5:0]						PBVD_EN	PVD1_EN	PVD_EN

Таблица 78 – Описание бит регистра CNTR0

Бит	Имя	Значение	Описание
31…9	-		Зарезервировано
8…3	PDID_EN[5:0]		Бит разрешения контроля тока потребления через LDO регуляторы: PDID_EN[0] – регулятор LDO0; PDID_EN[1] – регулятор LDO1; PDID_EN[2] – регулятор LDO2; PDID_EN[3] – регулятор LDO3; PDID_EN[4] – регулятор LDO батарейного домена; PDID_EN[5] – регулятор LDO умножителей частоты. 0 – отключен контроль; 1 – включен контроль
2	PBVD_EN		Бит разрешения контроля уровня напряжения BU_{CC} 0 – отключен; 1 – включен. **Внимание!** При PBVD_EN = 1 идет разрядка батарейного источника питания, независимо от наличия основного питания.
1	PVD1_EN		Бит разрешения контроля уровня напряжения $U_{CC}1$: 0 – отключен; 1 – включен
0	PVD_EN		Бит разрешения контроля уровня напряжения U_{CC}: 0 – отключен; 1 – включен

7.4.1.3 **CNTR1**

Base ADDR=			0x4000_2000		Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-							PDID_IE[5:0]						PBVD_IE	PVD1_IE	PVD_IE

Таблица 79 – Описание бит регистра CNTR1

Бит	Имя	Значение	Описание
31…9	-		Зарезервировано
8…3	PDID_IE[5:0]		Бит разрешения прерывания по событию тока потребления LDO регуляторов: PDID_IE[0] – регулятор LDO0; PDID_IE[1] – регулятор LDO1; PDID_IE[2] – регулятор LDO2; PDID_IE[3] – регулятор LDO3; PDID_IE[4] – регулятор LDO батарейного домена; PDID_IE[5] – регулятор LDO умножителей частоты. 0 – запрещено прерывание; 1 – разрешено прерывание
2	PBVDP_IE		Бит разрешения прерывания по событию уровня напряжения BU_{CC}: 0 – запрещено прерывание; 1 – разрешено прерывание
1	PVD1P_IE		Бит разрешения прерывания по событию уровня напряжения $U_{CC}1$: 0 – запрещено прерывание; 1 – разрешено прерывание
0	PVDP_IE		Бит разрешения прерывания по событию уровня напряжения U_{CC}: 0 – запрещено прерывание; 1 – разрешено прерывание

7.4.1.4 **CNTR2**

Base ADDR=			0x4000_2000		Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-							PDIDN[5:0]						PBVDN	PVD1N	PVDN

Таблица 80 – Описание бит регистра CNTR2

Бит	Имя	Значение	Описание
31…9	-		Зарезервировано
8…3	PDIDN[5:0]		Бит инвертирования сигнала от датчика тока LDO для формирования события: PDIDN[0] – регулятор LDO0; PDIDN[1] – регулятор LDO1; PDIDN[2] – регулятор LDO2; PDIDN[3] – регулятор LDO3; PDIDN[4] – регулятор LDO батарейного домена; PDIDN[5] – регулятор LDO умножителей частоты. 0 – не инвертировать (флаг по превышению тока); 1 – инвертировать (флаг по снижению тока)
2	PBVDN		Бит инвертирования сигнала от датчика уровня напряжения BU_{CC} для формирования события: 0 – не инвертировать (флаг по превышению); 1 – инвертировать (флаг по снижению)
1	PVD1N		Бит инвертирования сигнала от датчика уровня напряжения $U_{CC}1$ для формирования события: 0 – не инвертировать (флаг по превышению); 1 – инвертировать (флаг по снижению)
0	PVDN		Бит инвертирования сигнала от датчика уровня напряжения U_{CC} для формирования события: 0 – не инвертировать (флаг по превышению); 1 – инвертировать (флаг по снижению)

7.4.1.5 **STAT**

Base ADDR=		0x4000_2000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		PDID[5:0]						-					PBVDP	PVDP1	PVDP

Таблица 81 – Описание бит регистра STAT

Бит	Имя	Значение	Описание
31…14	-		Зарезервировано
13…8	PDID[5:0]		Флаги события по току LDO: PDID[0] – регулятор LDO0; PDID[1] – регулятор LDO1; PDID[2] – регулятор LDO2; PDID[3] – регулятор LDO3; PDID[4] – регулятор LDO батарейного домена; PDID[5] – регулятор LDO умножителей частоты. 0 – нет события; 1 – есть событие, ток больше допустимого CURDUCC при PDIDN=0, или ток ниже допустимого CURDUCC при PDIDN=1. Сбрасывается записью 1
7…3	-		Зарезервировано
2	PBVDP		Флаг события по уровню напряжения BU_{CC}: 0 – нет события; 1 – есть событие, напряжение больше допустимого BUCCMAX при PVBDN=0, или напряжение ниже допустимого BUCCMAX при PVBDN=1. Сбрасывается записью 1
1	PVDP1		Флаг события по уровню напряжения $U_{CC}1$: 0 – нет события; 1 – есть событие, напряжение больше допустимого UCC1MAX при PVD1N=0, или напряжение ниже допустимого UCC1MAX при PVD1N=1. Сбрасывается записью 1
0	PVDP		Флаг события по уровню напряжения U_{CC}: 0 – нет события; 1 – есть событие, напряжение больше допустимого UCCMAX при PVDN=0, или напряжение ниже допустимого UCCMAX при PVDN=1. Сбрасывается записью 1
Примечание - Для организации надежного сброса статусных флагов прерываний необходимо учитывать требования, изложенные в подразделе 9.3			

7.4.1.6 **ULIMIT**

Base ADDR=		0x4000_2000			Offset=		0x0000_0014								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-											BUCCMAX[4:0]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-			UCC1MAX[4:0]					-			UCCMAX[4:0]				

Таблица 82 – Описание бит регистра ULIMIT

Бит	Имя	Значение	Описание
31…21	-		Зарезервировано
20…16	BUCCMAX[4:0]		Значение максимально допустимого напряжения BU_{CC}, при превышении которого происходит выработка флага PBVDP и EVEN_PVBD: 5’b 00000 – более 1,2В (ref); 5’b 00001 – более 1,4В; 5’b 00010 – более 1,6В; 5’b 00011 – более 1,8В; … 5’b 11111 – более 7,4В
15…13	-		Зарезервировано
12…8	UCC1MAX[4:0]		Значение максимально допустимого напряжения $U_{CC}1$, при превышении или снижении которого происходит выработка флага PVDP1 и EVENT_PVD1: 5’b 00000 – более 1,2В (ref); 5’b 00001 – более 1,4В; 5’b 00010 – более 1,6В; 5’b 00011 – более 1,8В; … 5’b 11111 – более 7,4В
7…5	-		Зарезервировано
4…0	UCCMAX[4:0]		Значение максимально допустимого напряжения U_{CC}, при превышении которого происходит выработка флага PVDP и EVENT_PVD: 5’b 00000 – более 1,2В (ref); 5’b 00001 – более 1,4В; 5’b 00010 – более 1,6В; 5’b 00011 – более 1,8В; … 5’b 11111 – более 7,4В

7.4.1.7 **CLIMIT**

Base ADDR=		0x4000_2000			Offset=		0x0000_0018								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-														CURDUCC5[2:0]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURDUCC5 [2:0]	CURDUCC4 [2:0]			CURDUCC3 [2:0]			CURDUCC2 [2:0]			CURDUCC1 [2:0]			CURDUCC0 [2:0]		

Таблица 83 – Описание бит регистра CLIMIT

Бит	Имя	Значение	Описание
31….18	-		Зарезервировано
17…15	CURDUCC5[2:0]		Уровень тока для выработки события по току LDO умножителей частоты (блок PLL) и события EVENT_CUR[5]. Аналогично CURDUCC4[2:0]
14…12	CURDUCC4[2:0]		Уровень тока для выработки события по току LDO батарейного домена и формирования флага PDID[4] и события EVENT_CUR[4]: 3’b 000 – ток более 0,5 мА; 3’b 001 – ток более 1,5 мА; 3’b 010 – ток более 2,5 мА; 3’b 011 – ток более 3,5 мА; 3’b 100 – ток более 4,5 мА; 3’b 101 – ток более 5,5 мА; 3’b 110 – ток более 6,5 мА; 3’b 111 – ток более 7,5 мА
11…9	CURDUCC3[2:0]		Уровень тока для выработки события по току LDO3 и формирования флага PDID[3] и события EVENT_CUR[3]. Аналогично CURDUCC0[2:0]
8...6	CURDUCC2[2:0]		Уровень тока для выработки события по току LDO2 и формирования флага PDID[2] и события EVENT_CUR[2]. Аналогично CURDUCC0[2:0]
5…3	CURDUCC1[2:0]		Уровень тока для выработки события по току LDO1 и формирования флага PDID[1] и события EVENT_CUR[1]. Аналогично CURDUCC0[2:0]

Бит	Имя	Значение	Описание
2…0	CURDUCC0[2:0]		Уровень тока для выработки события по току LDO0 и формирования флага PDID[0] и события EVENT_CUR[0]: 3’b 000 – ток более 5 мА; 3’b 001 – ток более 15 мА; 3’b 010 – ток более 25 мА; 3’b 011 – ток более 35 мА; 3’b 100 – ток более 45 мА; 3’b 101 – ток более 55 мА; 3’b 110 – ток более 65 мА; 3’b 111 – ток более 75 мА

7.5 Контроллер обработки событий отказов, сбоев и ошибок (FTCNTR)

Все события, критические для функционирования и безопасности выполнения пользовательской программы, обрабатываются в контроллере обработки сбоев FT_CNTR. В зависимости от настроек блока FT_CNTR возникновение события может вызвать сигнал прерывания или сигналы предупреждения FT_WRNn или сигнал запроса сброса FT_RSTn микросхемы.

Для каждого события в блоке FT_CNTR возможна настройка реакции:

– игнорирование;

– прерывание:

– FT_IF0 – по возникновению событий из группы с высокой важностью (EVENT0-EVENT4), для которых разрешен отложенный сброс;

– FT_IF1 – по возникновению любого события из группы с высокой важностью (EVENT0-EVENT4);

– FT_IF2 – по возникновению разрешенных событий из группы со средней важностью (EVENT5-EVENT8);

– FT_IF3 – по возникновению разрешенных событий из группы с низкой важностью (EVENT9-EVENT12);

– формирование отложенного сброса микросхемы FT_WRNn и FT_RSTn.

Формирование событий ошибок ECC в массиве памяти ОЗУ, ПЗУ, памяти OTP программ и кэш-памяти разрешается в соответствующем контроллере (RAMCNTR, ROMCNTR, OTPCNTR, CACHECNTR), регистр ECCCS.

При возникновении события из группы с высокой важностью (EVENT0 – EVENT4), для которого разрешен отложенный сброс, в контроллере FT_CNTR автоматически запускается таймер. Период времени таймера задается программно и формирует временной интервал для выполнения действий по парированию возникшей ошибки. В момент возникновения ошибки формируется сигнал FT_WRNn, а по завершении счета таймера формируется сигнал FT_RSTn. Таким образом период таймера дает время, за которое можно успеть исправить программными средствами

ситуацию, вызвавшую сбой. Если к моменту окончания счета таймера сбойная ситуация разрешилась, то сброс таймера приведет к снятию сигнала FT_WRNn.

Сигналы FT_WRNn и FT_RSTn являются выходами с открытым стоком, и имеют встроенную подтяжку к питанию с номиналом 50 кОм.

7.5.1 Описание регистров

Таблица 84 – Описание регистров контроллера FT_CNTR

Базовый адрес	Название	Описание
0x4000_3000	MDR_FTMODE	Контроллер блока диагностики ошибок
Смещение		
0x0000_0000	KEY	Регистр ключа
0x0000_0004	CONTROL	Регистр управления блоком
0x0000_0008	STATUS	Регистр статуса блока
0x0000_000C	TIMEOUT	Регистр времени отложенного сброса
0x0000_0010	TICKCNT	Общий счетчик генератора HSI
0x0000_0014	FIRSTEVENT	Значение TICKCOUNT на момент возникновения первого разрешенного события сбоя
0x0000_0018	LASTEVENT	Значение TICKCOUNT на момент возникновения последнего разрешенного события сбоя
0x0000_001C	TIMEOUTCNT	Текущее значение счетчика отложенного сброса
0x0000_0020	EVENT0	Флаги событий сбоев
0x0000_0024	EVENT1	Флаги событий сбоев
0x0000_0028	EVENT2	Флаги событий сбоев
0x0000_002C	EVENT3	Флаги событий сбоев
0x0000_0030	EVENT4	Флаги событий сбоев
0x0000_0034	EVENT5	Флаги событий сбоев
0x0000_0038	EVENT6	Флаги событий сбоев
0x0000_003C	EVENT7	Флаги событий сбоев
0x0000_0040	EVENT8	Флаги событий сбоев
0x0000_0044	EVENT9	Флаги событий сбоев
0x0000_0048	EVENT10	Флаги событий сбоев
0x0000_004C	EVENT11	Флаги событий сбоев
0x0000_0050	EVENT12	Флаги событий сбоев
0x0000_0054	RESET_EVENT0	Биты разрешения отложенного сброса от событий сбоев
0x0000_0058	RESET_EVENT1	Биты разрешения отложенного сброса от событий сбоев
0x0000_005C	RESET_EVENT2	Биты разрешения отложенного сброса от событий сбоев
0x0000_0060	RESET_EVENT3	Биты разрешения отложенного сброса от событий сбоев
0x0000_0064	RESET_EVENT4	Биты разрешения отложенного сброса от событий сбоев
0x0000_0068	IE_EVENT5	Биты разрешения прерывания от событий сбоев
0x0000_006C	IE_EVENT6	Биты разрешения прерывания от событий сбоев
0x0000_0070	IE_EVENT7	Биты разрешения прерывания от событий сбоев
0x0000_0074	IE_EVENT8	Биты разрешения прерывания от событий сбоев
0x0000_0078	IE_EVENT9	Биты разрешения прерывания от событий сбоев
0x0000_007C	IE_EVENT10	Биты разрешения прерывания от событий сбоев

Базовый адрес	Название	Описание
0x0000_0080	IE_EVENT11	Биты разрешения прерывания от событий сбоев
0x0000_0084	IE_EVENT12	Биты разрешения прерывания от событий сбоев

7.5.1.1 **KEY**

Base ADDR=			0x4000_3000				Offset=			0x0000_0000						
REG Name:			KEY													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 85 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока FT_CNTR

7.5.1.2 **CONTROL**

Base ADDR=		0x4000_3000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-													RESET_TIMEOUTCNT	TICKOVER_IE	-

Таблица 86 – Описание бит регистра CONTROL

Бит	Имя	Значение	Описание
31…3	-		Зарезервировано
2	RESET_TIMEOUTCNT	0	Бит сброса регистра TIMEOUTCNT в состояние ожидания: 0 – ничего; 1 – сброс, при условии отсутствия каких-либо флагов события, для которых разрешен вызов отложенного сброса
1	TICKOVER_IE	0	Бит разрешения формирования сигнала предупреждения FT_WRNn при возникновении переполнения счетчика TICKCNT. Счетчик TICKCNT переполняется раз в ~9 минут

0	-	1	Зарезервировано

7.5.1.3 STATUS

Base ADDR=		0x4000_3000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-									FT_IF3	FT_IF2	FT_IF1	FT_IF0	-	-	TICKOVER

Таблица 87 – Описание бит регистра STATUS

Бит	Имя	Значение	Описание
31…7	-		Зарезервировано
6	FT_IF3	0	Флаг запроса прерывания от возникновения разрешенных событий с низкой важностью (EVENT9-EVENT12): 0 – нет запроса прерывания; 1 – есть запрос прерывания
5	FT_IF2	0	Флаг запроса прерывания от возникновения разрешенных событий со средней важностью (EVENT5-EVENT8): 0 – нет запроса прерывания; 1 – есть запрос прерывания
4	FT_IF1	0	Флаг запроса прерывания от возникновения любых событий с высокой важностью (EVENT0-EVENT4): 0 – нет запроса прерывания; 1 – есть запрос прерывания
3	FT_IF0	0	Флаг запроса прерывания от возникновения событий с высокой важностью, для которых разрешен отложенный сброс (EVENT0-EVENT4): 0 – нет запроса прерывания при запуске отложенного сброса; 1 – есть запрос прерывания при запуске отложенного сброса
2, 1	-		Зарезервировано
0	TICKOVER	0	Флаг переполнения счетчика TICKCNT: 0 – нет переполнения; 1 – было переполнение. Флаг активен при TICKOVER_IE = 1: - устанавливается в момент переполнения счетчика TICKCNT; - сбрасывается записью 1, если в момент сброса происходит очередное переполнение, флаг не сбрасывается. При TICKOVER_IE = 0 флаг удерживается в 0

7.5.1.4 **TIMEOUT**

Base ADDR=		0x4000_3000			Offset=		0x0000_000C								
REG Name:		TIMEOUT													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT[15:0]															

Таблица 88 – Описание бит регистра TIMEOUT

Бит	Имя	Значение	Описание
31…16	-		Зарезервировано
15…0	TIMEOUT[15:0]	16’HFFFF	Число тактов генератора HSI от момента возникновения сбоя до генерации сигнала сброса. При возникновении события сброса данное значение перезаписывается в регистр TIMEOUTCNT, который после этого начинает отсчет данного времени

7.5.1.5 **TICKCNT**

BaseADDR=		0x4000_3000			Offset=		0x0000_0010								
REGName:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TICKCNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TICKCNT[15:0]															

Таблица 89 – Описание бит регистра TICKCNT

Бит	Имя	Значение	Описание
31…0	TICKCNT[31:0]		Счетчик тактов генератора HSI с момента последнего сброса. При переполнении данного счетчика взводится бит TICKOVER

7.5.1.6 **FIRSTEVENT**

BaseADDR=		0x4000_3000			Offset=		0x0000_0014								
REGName:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIRSTEVENT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRSTEVENT[15:0]															

Таблица 90 – Описание бит регистра FIRSTEVENT

Бит	Имя	Значение	Описание
31…0	FIRSTEVENT[31:0]		Значение счетчика TICKCNT на момент возникновения первого события сбоя, для которого разрешен отложенный сброс (регистр RESET_EVENTx) или формирование прерывания (регистр IE_EVENTx). Сбрасывается только сигналом сброса

7.5.1.7 **LASTEVENT**

Base ADDR=		0x4000_3000			Offset=		0x0000_0018								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LASTEVENT[31:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LASTEVENT[15:0]															

Таблица 91 – Описание бит регистра LASTEVENT

Бит	Имя	Значение	Описание
31…0	LASTEVENT[31:0]		Значение счетчика TICKCNT на момент возникновения последнего события сбоя, для которого разрешен отложенный сброс (регистр RESET_EVENTx) или формирование прерывания (регистр IE_EVENTx). Сбрасывается только сигналом сброса

7.5.1.8 **TIMEOUTCNT**

Base ADDR=			0x4000_3000				Offset=		0x0000_001C						
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															TIMEOUTCNT[16]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUTCNT[15:0]															

Таблица 92 – Описание бит регистра TIMEOUTCNT

Бит	Имя	Значение	Описание
31…17	-		Зарезервировано
16…0	TIMEOUTCNT[16:0]	17`H1FFFF	Значение счетчика времени отложенного сброса

В момент возникновения события, для которого разрешено формирование отложенного сброса, в счетчик TIMEOUTCNT переписывается значение регистра TIMEOUT. Счетчик начинает считать вниз до нуля и при достижении нулевого значения формируется сигнал сброса FT_RSTn.

Счетчик TIMEOUTCNT имеет 17 разрядов, при сбросе устанавливается в состояние всех единиц. При возникновении события устанавливается в состояние 16-ти битного регистра TIMEOUT, при этом 17-й разряд сбрасывается в 0. Это условие начала счета для данного счетчика.

Счетчик выполнен по троированной логике.

Для прекращения счета отложенного сброса необходимо:

– снять все флаги событий, разрешающих формирование сигнала запроса сброса FT_RSTn;

– через запись 1 в бит RESET_TIMEOUTCNT установить счетчик в исходное состояние.

Если в момент сброса счетчика TIMEOUTCNT сохраняются условия для его счета, то сброс счетчика не произойдет.

7.5.1.9 **EVENT0 (HIGH)**

Примечание – Функционал не действителен, так как не реализован режим LOCKSTEP.

BaseADDR=		0x4000_3000			Offset=		0x0000_0020								
REGName:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		EXTBUS_COMP_REG_ERR	AHB_EXTBUS_COMP_ER	-		RAMD_COMP_REG_ERR	AHB_RAMD_COMP_ER	OTP_COMP REG_ERR	AHB_OTP_COMP_ER	-					

Таблица 93 – Описание бит регистра EVENT0 (HIGH)

Бит	Имя	Значение	Описание
31…14	-		Зарезервировано
13	EXTBUS_COMP_REG_ERR	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера EXTBUS: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
12	AHB_EXTBUS_COMP_ERR	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера EXTBUS: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
11, 10	-	2’b00	Зарезервировано
9	RAMD_COMP_REG_ERR	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера RAMD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
8	AHB_RAMD_COMP_ERR	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера RAMD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
7	OTP_COMP_REG_ERR	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера OTP: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
6	AHB_OTP_COMP_ERR	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера OTP: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
5…0	-		Зарезервировано

7.5.1.10 **EVENT1 (HIGH)**

Base ADDR=		0x4000_3000			Offset=		0x0000_0024								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-													ECCERR_ EXTBUS0	-	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	ECCERR_RAMD0	-	ECCERR_OTP0	-	ECCERR_ROM	-						ECCERRM_0	ECCERRS_0	ECCERRD_0	ECCERRI_0

Таблица 94 – Описание бит регистра EVENT1 (HIGH)

Бит	Имя	Значение	Описание
31…19	-		Зарезервировано
18	ECCERR_EXTBUS0*	0	Ошибка ECC на шине AHB на EXT_BUS0: 0 – нет ошибки; 1 – есть ошибка
17, 16	-	2'b00	Зарезервировано
15	-	0	Зарезервировано
14	ECCERR_RAMD0*	0	Ошибка ECC на шине AHB на RAMD0: 0 – нет ошибки; 1 – есть ошибка
13	ECCERR_OTP1*	0	Ошибка ECC на шине AHB на OTP1: 0 – нет ошибки; 1 – есть ошибка
12	ECCERR_OTP0*	0	Ошибка ECC на шине AHB на OTP0: 0 – нет ошибки; 1 – есть ошибка.
11	-	-	Зарезервировано
10	ECCERR_ROM*	0	Ошибка ECC на шине AHB на ROM: 0 – нет ошибки; 1 – есть ошибка
9…4	-		Зарезервировано
3	ECCERRM_0*	0	Ошибка ECC на шине M на DMA: 0 – нет ошибки; 1 – есть ошибка
2	ECCERRS_0*	0	Ошибка ECC на шине S на CPU: 0 – нет ошибки; 1 – есть ошибка
1	ECCERRD_0*	0	Ошибка ECC на шине D на CPU: 0 – нет ошибки; 1 – есть ошибка
0	ECCERRI_0*	0	Ошибка ECC на шине I на CPU: 0 – нет ошибки; 1 – есть ошибка
* Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается			

7.5.1.11 **EVENT2 (HIGH)**

Base ADDR=			0x4000_3000		Offset=		0x0000_0028								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					-	ECC_CACHE_RAMD_DE		ECC_CACHE_D_0_DE		ECC_CACHE_I_0_DE	ECC_EXTBUS_DE	-	ECC_RAMD_DE	ECC_OTP_DE	ECC_ROM_DE

Таблица 95 – Описание бит регистра EVENT2 (HIGH)

Бит	Имя	Значение	Описание
31…10	-		Зарезервировано
9	ECC_CACHE_RAMD_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE в RAMD: 0 – нет ошибки; 1 – есть ошибка
8	-		Зарезервировано
7	ECC_CACHE_D_0_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_D в CPU: 0 – нет ошибки; 1 – есть ошибка
6	-		Зарезервировано
5	ECC_CACHE_I_0_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_I в CPU: 0 – нет ошибки; 1 – есть ошибка
4	ECC_EXTBUS_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти EXT_BUS: 0 – нет ошибки; 1 – есть ошибка
3	-		Зарезервировано
2	ECC_RAMD_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти RAMD: 0 – нет ошибки; 1 – есть ошибка
1	ECC_OTP_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти OTP: 0 – нет ошибки; 1 – есть ошибка
0	ECC_ROM_DE*	0	Двойная неисправимая ошибка ECC в массиве памяти ROM: 0 – нет ошибки; 1 – есть ошибка
* Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается			

7.5.1.12 **EVENT3 (HIGH)**

Base ADDR=		0x4000_3000			Offset=		0x0000_002C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-												CLK_ MONITOR3	CLK_ MONITOR2		CLK_ MONITOR0

Таблица 96 – Описание бит регистра EVENT3 (HIGH)

Бит	Имя	Значение	Описание
31…4	-		Зарезервировано
3	CLK_MONITOR3*	0	Ошибка превышения максимальной тактовой частоты RAMD: 0 – нет ошибки; 1 – есть ошибка
2	CLK_MONITOR2*	0	Ошибка превышения максимальной тактовой частоты OTP: 0 – нет ошибки; 1 – есть ошибка
1	-		Зарезервировано
0	CLK_MONITOR0*	0	Ошибка превышения максимальной тактовой частоты CPU: 0 – нет ошибки; 1 – есть ошибка
* Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается			

7.5.1.13 **EVENT4 (HIGH)**

Base ADDR=		0x4000_3000			Offset=		0x0000_0030								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-							ROM_REG_ DE	-	RAMD_REG_ DE	OTP_REG_ DE	FT_REG_ DE	FTCNTR_ERR	BKP_MEM_ DE	BKP_ERR	RTC_ERR

Таблица 97 – Описание бит регистра EVENT4 (HIGH)

Бит	Имя	Значение	Описание
31…9			
8	ROM_REG_DE*		Двойная неисправимая ошибка в регистрах управления ROM_CNTR: 0 – нет ошибки; 1 – есть ошибка
7	-		Зарезервировано
6	RAMD_REG_DE*		Двойная неисправимая ошибка в регистрах управления RAMD_CNTR: 0 – нет ошибки; 1 – есть ошибка
5	OTP_REG_DE*	0	Двойная неисправимая ошибка в регистрах управления OTP_CNTR 0 – нет ошибки 1 – есть ошибка
4	FT_REG_DE*	0	Двойная неисправимая ошибка в регистрах управления FT_CNTR: 0 – нет ошибки; 1 – есть ошибка
3	FTCNTR_ERR*	0	Ошибка в счетчике TIMEOUTCNT отложенного сброса FT_CNTR: 0 – нет ошибки; 1 – есть ошибка
2	BKP_MEM_DE*	0	Двойная ошибка в массиве памяти BKP (00-59): 0 – нет ошибки; 1 – есть ошибка
1	BKP_ERR*	0	Ошибка в конфигурационных регистрах BKP (60-63): 0 – нет ошибки; 1 – есть ошибка
0	RTC_ERR*	0	Ошибка в часах реального времени RTC: 0 – нет ошибки; 1 – есть ошибка
* Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается			

7.5.1.14 **EVENT5 (MEDIUM)**

Base ADDR=				0x4000_3000		Offset=			0x0000_0034						
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-			RTC_CLK_DE	DAC_CLK_DE	ADC_CLK_DE	SSP_CLK_DE	UART_CLK_DE		SPW_CLK_DE	EPHY_CLK_DE	-			TIM_CLK_DE	CAN_CLK_DE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-					PLL2_CLK_DE	PLL1_CLK_DE	PLL0_CLK_DE	HSE1_CLK_DE	HSE0_CLK_DE	LSE_CLK_DE	LSI_CLK_DE	PER_CLK_DE	CPU_CLK_DE	MAX_CLK_DE	KEY_DE

Таблица 98 – Описание бит регистра EVENT5 (MEDIUM)

Бит	Имя	Значение	Описание
31…29	-		Зарезервировано
28	RTC_CLK_DE		Двойная неисправимая ошибка в регистре RTC_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
27	DAC_CLK_DE		Двойная неисправимая ошибка в регистре DAC_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
26	ADC_CLK_DE		Двойная неисправимая ошибка в регистре ADC_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
25	SSP_CLK_DE		Двойная неисправимая ошибка в регистре SSP_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
24	UART_CLK_DE		Двойная неисправимая ошибка в регистре UART_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
23			Зарезервировано
22	SPW_CLK_DE		Двойная неисправимая ошибка в регистре SPW_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
21	EPHY_CLK_DE		Двойная неисправимая ошибка в регистре EPHY_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
20, 18	-		Зарезервировано
17	TIM_CLK_DE		Двойная неисправимая ошибка в регистре TIM_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
16	CAN_CLK_DE		Двойная неисправимая ошибка в регистре CAN_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
15…11			Зарезервировано

Бит	Имя	Значение	Описание
10	PLL2_CLK_DE		Двойная неисправимая ошибка в регистре PLL2_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
9	PLL1_CLK_DE		Двойная неисправимая ошибка в регистре PLL1_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
8	PLL0_CLK_DE		Двойная неисправимая ошибка в регистре PLL0_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
7	HSE1_CLK_DE		Двойная неисправимая ошибка в регистре HSE1_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
6	HSE0_CLK_DE		Двойная неисправимая ошибка в регистре HSE0_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
5	LSE_CLK_DE		Двойная неисправимая ошибка в регистре LSE_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
4	LSI_CLK_DE		Двойная неисправимая ошибка в регистре LSI_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
3	PER_CLK_DE		Двойная неисправимая ошибка в регистре PER_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
2	CPU_CLK_DE		Двойная неисправимая ошибка в регистре CPU_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
1	MAX_CLK_DE		Двойная неисправимая ошибка в регистре MAX_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается
0	KEY_DE		Двойная неисправимая ошибка в регистре KEY блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, но, если в момент сброса происходит очередное событие, флаг не сбрасывается

7.5.1.15 **EVENT6 (MEDIUM)**

Base ADDR=		0x4000_3000			Offset=		0x0000_0038								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								PLL2_CLOCK_ERROR	PLL1_CLOCK_ERROR	PLL0_CLOCK_ERROR	LSE_CLOCK_ERROR	LSI_CLOCK_ERROR	HSE1_CLOCK_ERROR	HSE0_CLOCK_ERROR	CPU_CLOCK_ERROR

Таблица 99 – Описание бит регистра EVENT6 (MEDIUM)

Бит	Имя	Значение	Описание
31…8	-		Зарезервировано
7	PLL2_CLOCK_ERROR		Ошибка при формировании частоты PLL2_CLK в блоке CLK_CHECKER в блоке CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
6	PLL1_CLOCK_ERROR		Ошибка при формировании частоты PLL1_CLK в блоке CLK_CHECKER в блоке CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
5	PLL0_CLOCK_ERROR		Ошибка при формировании частоты PLL0_CLK в блоке CLK_CHECKER в блоке CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
4	LSE_CLOCK_ERROR		Ошибка при формировании частоты LSE_CLK в блоке CLK_CHECKER в блоке CLKCNTR 0 – нет ошибки 1 – есть ошибка Флаг сбрасываются записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
3	LSI_CLOCK_ERROR		Ошибка при формировании частоты LSI_CLK в блоке CLK_CHECKER в блоке CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
2	HSE1_CLOCK_ERROR		Ошибка при формировании частоты HSE1_CLK в блоке CLK_CHECKER в блоке CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
1	HSE0_CLOCK_ERROR		Ошибка при формировании частоты HSE0_CLK в блоке CLK_CHECKER в блоке CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
0	CPU_CLOCK_ERROR		Ошибка при формировании частоты CPU_CLK в блоке CLK_CHECKER в блоке CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

7.5.1.16 **EVENT7 (MEDIUM)**

Base ADDR=		0x4000_3000			Offset=		0x0000_003C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	RAMD_ CACHE_SE	ROM_ REG_SE	-	RAMD_ REG_SE	OTP_ REG_SE	FT_ REG_SE	BKP_ MEM_SE	-	RAMD_SE	-	ECC_CACHE_ D_0_SE	-	ECC_CACHE_ I_0_SE	ROM_SE	OTP_SE

Таблица 100 – Описание бит регистра EVENT7 (MEDIUM)

Бит	Имя	Значение	Описание
31…15	-		Зарезервировано
14	RAMD_CACHE_SE		Одиночная ошибка в массиве кэш-памяти RAMD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
13	ROM_REG_SE		Одиночная ошибка в регистрах управления блока ROMCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
12	-		Зарезервировано
11	RAMD_REG_SE		Одиночная ошибка в регистрах управления блока RAMD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
10	OTP_REG_SE	0	Одиночная ошибка в регистрах управления блока OTPCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
9	FT_REG_SE	0	Одиночная ошибка в регистрах управления блока FTCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
8	BKP_MEM_SE	0	Одиночная ошибка в массиве памяти BKP (00-59): 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
7	-		Зарезервировано
6	RAMD_SE	0	Одиночная ошибка в массиве памяти RAMD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
5	-		Зарезервировано

Бит	Имя	Значение	Описание
4	ECC_CACHE_D_0_SE	0	Одиночная ошибка в массиве памяти CACHEDCPU: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
3	-		Зарезервировано
2	ECC_CACHE_I_0_SE	0	Одиночная ошибка в массиве памяти CACHEICPU: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
1	ROM_SE	0	Одиночная ошибка в массиве памяти ROM: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
0	OTP_SE	0	Одиночная ошибка в массиве памяти OTP: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

7.5.1.17 **EVENT8 (MEDIUM)**

Base ADDR=		0x4000_3000			Offset=		0x0000_0040								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-							Ducc_PLL_CUR_EVENT	DuccB_CUR_EVENT	LDO3_CUR_EVENT	LDO2_CUR_EVENT	LDO1_CUR_EVENT	LDO0_CUR_EVENT	PVD_BUCC_EVENT	PVD_UCC1_EVENT	PVD_UCC_EVENT

Таблица 101 – Описание бит регистра EVENT8 (MEDIUM)

Бит	Имя	Значение	Описание
31…9		-	Зарезервировано
8	Ducc_PLL_CUR_EVENT	0	Ошибка по току потребления домена множителей частоты PLL от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
7	DuccB_CUR_EVENT	0	Ошибка по току потребления батарейного домена от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
6	LDO3_CUR_EVENT	0	Ошибка по току потребления LDO3 от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
5	LDO2_CUR_EVENT	0	Ошибка по току потребления LDO2 от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
4	LDO1_CUR_EVENT	0	Ошибка по току потребления LDO1 от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
3	LDO0_CUR_EVENT	0	Ошибка по току потребления LDO0 от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
2	PVD_BUCC_EVENT	0	Ошибка по питанию BU_{CC} от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
1	PVD_UCC1_EVENT	0	Ошибка по питанию $U_{CC}1$ от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
0	PVD_UCC_EVENT	0	Ошибка по питанию U_{CC} от блока PVD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

7.5.1.18 **EVENT9 (LOW)**

Base ADDR=		0x4000_3000			Offset=		0x0000_0044								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-			RTC_CLK_SE	DAC_CLK_SE	ADC_CLK_SE	SSP_CLK_SE	UART_CLK_SE		SPW_CLK_SE	EPHY_CLK_SE	-			TIM_CLK_SE	CAN_CLK_SE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-					PLL2_CLK_SE	PLL1_CLK_SE	PLL0_CLK_SE	HSE1_CLK_SE	HSE0_CLK_SE	LSE_CLK_DE	LSI_CLK_DE	PER_CLK_SE	CPU_CLK_SE	MAX_CLK_SE	KEY_SE

Таблица 102 – Описание бит регистра EVENT9 (MEDIUM)

Бит	Имя	Значение	Описание
31…29	-		Зарезервировано
28	RTC_CLK_SE		Одиночная исправимая ошибка в регистре RTC_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
27	DAC_CLK_SE		Одиночная исправимая ошибка в регистре DAC_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
26	ADC_CLK_SE		Одиночная исправимая ошибка в регистре ADC_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
25	SSP_CLK_SE		Одиночная исправимая ошибка в регистре SSP_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
24	UART_CLK_SE		Одиночная исправимая ошибка в регистре UART_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
23	-		Зарезервировано
22	SPW_CLK_SE		Одиночная исправимая ошибка в регистре SPW_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
21	EPHY_CLK_SE		Одиночная исправимая ошибка в регистре EPHY_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
20…18	-		Зарезервировано
17	TIM_CLK_SE		Одиночная исправимая ошибка в регистре TIM_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
16	CAN_CLK_SE		Одиночная исправимая ошибка в регистре CAN_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
15…11	-		Зарезервировано
10	PLL2_CLK_SE		Одиночная исправимая ошибка в регистре PLL2_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
9	PLL1_CLK_SE		Одиночная исправимая ошибка в регистре PLL1_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
8	PLL0_CLK_SE		Одиночная исправимая ошибка в регистре PLL0_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
7	HSE1_CLK_SE		Одиночная исправимая ошибка в регистре HSE1_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
6	HSE0_CLK_SE		Одиночная исправимая ошибка в регистре HSE0_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
5	LSE_CLK_SE		Одиночная исправимая ошибка в регистре LSE_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
4	LSI_CLK_SE		Одиночная исправимая ошибка в регистре LSI_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
3	PER_CLK_SE		Одиночная исправимая ошибка в регистре PER_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
2	CPU_CLK_SE		Одиночная исправимая ошибка в регистре CPU_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
1	MAX_CLK_SE		Одиночная исправимая ошибка в регистре MAX_CLK блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
0	KEY_SE		Одиночная исправимая ошибка в регистре KEY блока CLKCNTR: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

7.5.1.19 **EVENT10 (LOW)**

Base ADDR=		0x4000_3000			Offset=		0x0000_0048								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-										PORTF_CURLIM	PORTE_CURLIM	PORTD_CURLIM	PORTC_CURLIM	PORTB_CURLIM	PORTA_CURLIM

Таблица 103 – Описание бит регистра EVENT10 (MEDIUM)

Бит	Имя	Значение	Описание
31…6	-		Зарезервировано
5	PORTF_CURLIM		Ошибка превышения выходного тока в PORTF: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
4	PORTE_CURLIM		Ошибка превышения выходного тока в PORTE: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

Бит	Имя	Значение	Описание
3	PORTD_CURLIM		Ошибка превышения выходного тока в PORTD: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
2	PORTC_CURLIM		Ошибка превышения выходного тока в PORTC: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
1	PORTB_CURLIM		Ошибка превышения выходного тока в PORTB: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
0	PORTA_CURLIM		Ошибка превышения выходного тока в PORTA: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

7.5.1.20 **EVENT11 (LOW)**

Base ADDR=		0x4000_3000			Offset=		0x0000_004C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	EMAC0_DE	-		ARC1_DE	ARC0_DE	-		MIL1_DE	MIL0_DE	-					CAN0_DE

Таблица 104 – Описание бит регистра EVENT11 (LOW)

Бит	Имя	Значение	Описание
31…15	-		Зарезервировано
14	EMAC0_DE	0	Двойная неисправимая ошибка в массивах памяти ETHERNETMAC0: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
13, 12	-		Зарезервировано

Бит	Имя	Значение	Описание
11	ARC1_DE	0	Двойная неисправимая ошибка в массивах памяти ARC1: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
10	ARC0_DE	0	Двойная неисправимая ошибка в массивах памяти ARC0: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
9, 8	-		Зарезервировано
7	MIL1_DE	0	Двойная неисправимая ошибка в массивах памяти MIL1: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
6	MIL0_DE	0	Двойная неисправимая ошибка в массивах памяти MIL0: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
5…1	-		Зарезервировано
0	CAN0_DE	0	Двойная неисправимая ошибка в массивах памяти CAN0: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

7.5.1.21 **EVENT12 (LOW)**

Base ADDR=		0x4000_3000			Offset=		0x0000_0050								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMAC0_DE	-		ARC1_DE	ARC0_DE	-		MIL1_DE	MIL0_DE	-					CAN0_DE

Таблица 105 – Описание бит регистра EVENT12 (LOW)

Бит	Имя	Значение	Описание
31…15	-		Зарезервировано
14	EMAC0_SE	0	Одиночная неисправимая ошибка в массивах памяти ETHERNETMAC0: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
13, 12	-		Зарезервировано
11	ARC1_SE	0	Одиночная неисправимая ошибка в массивах памяти ARC1: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
10	ARC0_SE	0	Одиночная неисправимая ошибка в массивах памяти ARC0: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
9, 8	-		Зарезервировано
7	MIL1_SE	0	Одиночная неисправимая ошибка в массивах памяти MIL1: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
6	MIL0_SE	0	Одиночная неисправимая ошибка в массивах памяти MIL0: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается
5…1	-		Зарезервировано
0	CAN0_SE	0	Одиночная неисправимая ошибка в массивах памяти CAN0: 0 – нет ошибки; 1 – есть ошибка. Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается

7.5.1.22 **RESET_EVENT0**

Base ADDR=		0x4000_3000			Offset=		0x0000_0054								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		RESET_EN[13:0]													

Таблица 106 – Описание бит регистра RESET_EVENT0

Бит	Имя	Значение	Описание
31…14	-		Зарезервировано
13…0	RESET_EN[13:0]	14’h3FFF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT0: 0 – сброс не формируется; 1 – сброс формируется

7.5.1.23 **RESET_EVENT1**

Base ADDR=		0x4000_3000			Offset=		0x0000_0058								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-												RESET_EN[19:16]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET_EN[15:0]															

Таблица 107 – Описание бит регистра RESET_EVENT1

Бит	Имя	Значение	Описание
31…20	-		Зарезервировано
19…0	RESET_EN[19:0]	20’h1FFFFF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT1: 0 – сброс не формируется; 1 – сброс формируется

7.5.1.24 **RESET_EVENT2**

Base ADDR=		0x4000_3000			Offset=		0x0000_005C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-						RESET_EN[9:0]									

Таблица 108 – Описание бит регистра RESET_EVENT2

Бит	Имя	Значение	Описание
31…10	-		Зарезервировано
9…0	RESET_EN[9:0]	10’h3FF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT2: 0 – сброс не формируется; 1 – сброс формируется

7.5.1.25 **RESET_EVENT3**

Base ADDR=		0x4000_3000			Offset=		0x0000_0060								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-											RESET_EN[3:0]				

Таблица 109 – Описание бит регистра RESET_EVENT3

Бит	Имя	Значение	Описание
31…4	-		Зарезервировано
3…0	RESET_EN[3:0]	4’hF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT3: 0 – сброс не формируется; 1 – сброс формируется

7.5.1.26 **RESET_EVENT4**

Base ADDR=		0x4000_3000			Offset=		0x0000_0064								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-							RESET_EN[8:0]								

Таблица 110 – Описание бит регистра RESET_EVENT4

Бит	Имя	Значение	Описание
31…9	-		Зарезервировано
8…0	RESET_EN[8:0]	9’h1FF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT4: 0 – сброс не формируется; 1 – сброс формируется

7.5.1.27 **IE_EVENT5**

Base ADDR=		0x4000_3000			Offset=		0x0000_0068								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	INT_EN[28]	INT_EN[27]	INT_EN[26]	INT_EN[25]	INT_EN[24]	-	INT_EN[22]	INT_EN[21]	-	-	-	INT_EN[17]	INT_EN[16]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	INT_EN[10]	INT_EN[9]	INT_EN[8]	INT_EN[7]	INT_EN[6]	INT_EN[5]	INT_EN[4]	INT_EN[3]	INT_EN[2]	INT_EN[1]	INT_EN[0]

Таблица 111 – Описание бит регистра IE_EVENT5

Бит	Имя	Значение	Описание
31…29	-		Зарезервировано
28…0	INT_EN[28:0]	29’h0	Биты разрешения запроса прерывания FT_IF2 и формирования флага FT_WRNn по событиям в EVENT5: 0 – запрос прерывания не формируется; 1 – запрос прерывания формируется

7.5.1.28 **IE_EVENT6**

Base ADDR=		0x4000_3000			Offset=		0x0000_006C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								INT_EN[7:0]							

Таблица 112 – Описание бит регистра IE_EVENT6

Бит	Имя	Значение	Описание
31…8	-		Зарезервировано
7…0	INT_EN[7:0]	8’h00	Биты разрешения запроса прерывания FT_IF2 и формирования флага FT_WRNn по событиям в EVENT6: 0 – запрос прерывания не формируется; 1 – запрос прерывания формируется

7.5.1.29 **IE_EVENT7**

Base ADDR=		0x4000_3000			Offset=		0x0000_0070								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	INT_EN[14:0]														

Таблица 113 – Описание бит регистра IE_EVENT7

Бит	Имя	Значение	Описание
31…15	-		Зарезервировано
14…0	INT_EN[14:0]	15’h0000	Биты разрешения запроса прерывания FT_IF2 и формирования флага FT_WRNn по событиям в EVENT7: 0 – запрос прерывания не формируется; 1 – запрос прерывания формируется

7.5.1.30 **IE_EVENT8**

Base ADDR=		0x4000_3000			Offset=		0x0000_0074								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-					INT_EN[10:0]										

Таблица 114 – Описание бит регистра IE_EVENT8

Бит	Имя	Значение	Описание
31…11	-		Зарезервировано
10…0	INT_EN[10:0]	11’h0000	Биты разрешения запроса прерывания FT_IF2 и формирования флага FT_WRNn по событиям в EVENT8: 0 – запрос прерывания не формируется; 1 – запрос прерывания формируется

7.5.1.31 **IE_EVENT9**

Base ADDR=		0x4000_3000			Offset=		0x0000_0078								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-			INT_EN[28:16]												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_EN[15:0]															

Таблица 115 – Описание бит регистра IE_EVENT9

Бит	Имя	Значение	Описание
31…29	-		Зарезервировано
28…0	INT_EN[28:0]	29’h0000	Биты разрешения запроса прерывания FT_IF3 и формирования флага FT_WRNn по событиям в EVENT9: 0 – запрос прерывания не формируется; 1 – запрос прерывания формируется

7.5.1.32 **IE_EVENT10**

Base ADDR=		0x4000_3000			Offset=		0x0000_007C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-										INT_EN[5:0]					

Таблица 116 – Описание бит регистра IE_EVENT10

Бит	Имя	Значение	Описание
31…6	-		Зарезервировано
5…0	INT_EN[5:0]	6’h0000	Биты разрешения запроса прерывания FT_IF3 и формирования флага FT_WRNn по событиям в EVENT10: 0 – запрос прерывания не формируется; 1 – запрос прерывания формируется

7.5.1.33 **IE_EVENT11**

Base ADDR=		0x4000_3000			Offset=		0x0000_0080								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	INT_EN[14:0]														

Таблица 117 – Описание бит регистра IE_EVENT11

Бит	Имя	Значение	Описание
31…15	-		Зарезервировано
14…0	INT_EN[14:0]	15’h0000	Биты разрешения запроса прерывания FT_IF3 и формирования флага FT_WRNn по событиям в EVENT11: 0 – запрос прерывания не формируется; 1 – запрос прерывания формируется

7.5.1.34 **IE_EVENT12**

Base ADDR=		0x4000_3000			Offset=		0x0000_0084								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	INT_EN[14:0]														

Таблица 118 – Описание бит регистра IE_EVENT12

Бит	Имя	Значение	Описание
31…15	-		Зарезервировано
14…0	INT_EN[14:0]	15’h0000	Биты разрешения запроса прерывания FT_IF3 и формирования флага FT_WRNn по событиям в EVENT12: 0 – запрос прерывания не формируется; 1 – запрос прерывания формируется

7.6 Контроллер сторожевых таймеров (WDGCNTR)

В микросхеме реализован сторожевой таймер, при использовании которого, микросхема может быть сброшена при несоблюдении временных интервалов обязательного периода сброса.

Сторожевой таймер предназначен для аварийного перезапуска микросхемы, когда за период счета сторожевого таймера не будет выполнена хотя бы одна операция по его перезапуску. Сторожевой таймер выполнен в виде сбоеустойчивой логики, то есть при возникновении одиночного сбоя, таймер продолжит выполнять поставленную задачу. Для последующего устранения последствий сбоя требуется программная переинициализация таймера. Конфигурация блока защищена с помощью кодирования ECC, а сам счетчик построен по троированной схеме.

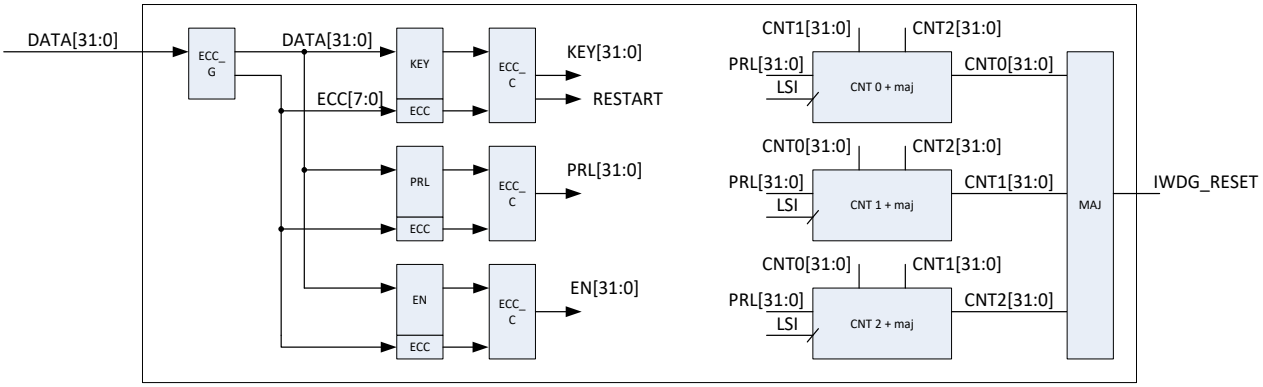


Рисунок 33 – Структурная схема контроллера троированного сторожевого таймера

Для инициализации сторожевого таймера необходимо:

1 В регистр KEY записать 0x05555;

2 Считать регистр и убедится, что записано верное значение;

3 В регистр PRL записать требуемое основание счета;

4 Считать регистр и убедится, что записано верное значение;

5 Записать в регистр KEY значение 0x0AAAA.

Через время не менее 200 мкс значение PRL должно быть автоматически переписано в регистр CNT.

1 Считать регистр CNT и убедится, что в него записано верное значение из регистра PRL;

2 В регистр KEY записать 0x0CCCC;

3 Считать регистр и убедится, что записано верное значение;

4 В регистр EN записать значение 0x03333. После чего сторожевой таймер начнет работу;

5 При возникновении сбоев провести операции по переинициализации сторожевого таймера;

6 С периодом меньшим заданного через регистр PRL проводить перезапуск сторожевого таймера;

7 Контролировать, что генератор LSI работает.

Для перезапуска сторожевого таймера необходимо:

1 При необходимости считать значение CNT и проверить, что оно находится в ожидаемом диапазоне;

2 Записать в регистр KEY значение 0x0AAAA;

Для переинициализации сторожевого таймера необходимо:

1 В регистр KEY записать 0x0CCCC;

2 Считать регистр и убедится, что записано верное значение;

3 В регистр EN записать значение отличное от 0x03333. После чего будет остановлен счет;

4 Выполнить операции по инициализации сторожевого таймера.

Примечание – Не выключать генератор HSI при использовании сброса от WDGCNTR, см. описание ошибки 0097 в документе «К1986ВЕ8Т Errata Notice».

7.6.1 Описание регистров

Таблица 119 – Описание регистров контроллера сторожевых таймеров

Базовый Адрес	Название	Описание
0x4000_4000	MDR_WDG	Котнроллер сторожевого таймера
Смещение		
0x0000_0000	KEY	Регистр Ключей
0x0000_0004	PRL	Регистр периода счета сторожевого таймера
0x0000_0008	EN	Регистр разрешения работы
0x0000_000C	CNT	Значение счетчика сторожевого таймера

7.6.1.1 **KEY**

Base ADDR=		0x4000_4000			Offset=		0x0000_0000								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 120 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0	Регистр управляющих ключей блока сторожевого таймера. 0x05555 – значение, разрешающее запись в регистр периода PRL счета сторожевого таймера. 0x0CCCC – значение, разрешающее запись в регистр разрешения работы EN сторожевого таймера. Запись 0x0AAAA – перезапускает счет сторожевого таймера. Значения, отличные от указанных выше, не имеют значения

7.6.1.2 **PRL**

Base ADDR=		0x4000_4000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRL[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRL[15:0]															

Таблица 121 – Описание бит регистра PRL

Бит	Имя	Значение	Описание
31…0	PRL[31:0]	0	Регистр периода PRL счета сторожевого таймера. Определяет период работы сторожевого таймера. Счетчик сторожевого таймера CNT считает от значения PRL до 0. При достижении нулевого значения в счетчике CNT формируется сигнал сброса IWDG_RESET. Счетчик CNT считает на частое генератора LSI. При значении PRL = 0xFF должен перезапускаться не реже чем раз в 2,5 мс. При значении PRL = 0xFFFF должен перезапускаться не реже чем раз в 655 мс. При значении PRL = 0xFFFFFFFF должен перезапускаться не реже чем раз в 10 часов

7.6.1.3 **EN**

Base ADDR=		0x4000_4000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN[15:0]															

Таблица 122 – Описание бит регистра EN

Бит	Имя	Значение	Описание
31…0	EN[31:0]	0	Значение разрешения работы счетчика: 0x03333 – счетчик работает. Любые отличные значения останавливают счетчик

7.6.1.4 **CNT**

Base ADDR=		0x4000_4000			Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Таблица 123 – Описание бит регистра CNT

Бит	Имя	Значение	Описание
31…0	CNT[31:0]		Текущее значение счетчика CNT. При чтении надо учитывать, что счетчик CNT работает на частоте LSI (~40 кГц), и при высокой частоте работы ядра может потребоваться дополнительное время для обновления счетчика

7.7 Контроллер внешней шины (EXTBUSCNTR)

С помощью контроллера внешней шины EXTBUS можно организовать до восьми регионов для подключения внешних устройств с отображением в адресное пространство микросхемы по диапазонам, приведенным в таблице 124.

Таблица 124 – Базовые адреса внешней системной шины

Номер	Начальный адрес	Конечный адрес	Шина доступа
REGION0	0x1000_0000	0x17FF_FFFF	I, D, M
REGION1	0x1800_0000	0x1FFF_FFFF	I, D, M
REGION2	0x5000_0000	0x57FF_FFFF	S, M
REGION3	0x5800_0000	0x5FFF_FFFF	S, M
REGION4	0x6000_0000	0x67FF_FFFF	S, M
REGION5	0x6800_0000	0x6FFF_FFFF	S, M
REGION6	0x7000_0000	0x7FFF_FFFF	S, M
REGION7	0x8000_0000	0xAFFF_FFFF	S, M

Для каждого региона может быть задан свой режим работы функционирования. При отображении выводов контроллера внешней шины на пользовательские выводы этим выводам необходимо задать функцию внешней шины. Если выводу не задана функция внешней шины, то с него считывается ноль.

7.7.1 Организация доступа к внешней шине EXTBUS

Таблица 125 – Интерфейс контроллера EXT_BUS

Имя входа	Направление	Описание
ADDR[31:0]	Выход	Шина адреса
DATA[31:0] DATA[63:32]	Вход/Выход	Шина данных
DATA[71:64] DATA[79:72]	Вход/Выход	Шина данных ECC
BEn[9:0]	Выход	Выход ByteEnable: 0 – разрешение байта; 1 – запрет байта. BEn[0] – DATAo[7:0]; BEn[1] – DATAo[15:8]; … BEn[7] – DATAo[63:54]; BEn[8] – DATAo[71:64]; BEn[9] – DATAo[79:72]
CSn[7:0]	Выход	Выход разрешения региона: 0 – регион выбран; 1 – регион не выбран

Имя входа	Направление	Описание
CS[7:0]	Выход	Выход разрешения региона: 0 – регион не выбран; 1 – регион выбран
WEn[1:0]	Выход	Выход сигнала записи: 0 – запись; 1 – нет записи. WEn[0] – DATAo[31:0] + DATAo[71:64]; WEn[1] – DATAo[63:32] + DATAo[79:72]
OEn[1:0]	Выход	Сигнал чтения данных с внешних выводов: 0 – есть чтение; 1 – нет чтения. OEn[0] - DATAi[31:0] + DATAi[71:64]; OEn[1] - DATAi[63:32] + DATAi[79:72]
BWEn[9:0]	Выход	Выход сигнала записи с сигналом ByteEnable: BWEn[0] = WEn[0] or BEn[0]; BWEn[1] = WEn[0] or BEn[1]; … BWEn[7] = WEn[1] or BEn[7]; BWEn[8] = WEn[0] or BEn[8]; BWEn[9] = WEn[1] or BEn[9]
READY[7:0]	Вход	Вход готовности от блоков памяти с сигналом готовности: 0 – не готов; 1 – готов
CLOCK	Выход	Выход сигнала синхронизации внутри транзакции
OCLK	Выход	Выход тактового сигнала, формируемого контроллером внешней системной шины

Внешняя шина может иметь 8-, 16-, 32- и 64-битную организацию шины данных (без учета проверочных бит ECC при параллельной организации). При этом процессорное ядро может оперировать только 8-, 16- и 32-битными данными на внешней шине. Реализация 64-битной шины возможна и имеет смысл только для выполнения программного кода из внешней памяти. В этом случае из внешней памяти извлекается сразу два последовательных 32-разрядных слова (запрошенное четное и следующее за ним или запрошенное нечетное и предыдущее перед ним). Это позволяет повысить производительность при исполнении кода из внешней памяти при последовательной выборке, так как в момент обращения к следующему слову оно уже считано и хранится в контроллере, и чтение на внешней шине не выполняется. Реализация 64-битной шины для работы с какой-либо внешней периферией недопустима, так как упреждающее чтение может повлиять на информацию критичную к чтению (буфера FIFO, регистры данных различных контроллеров интерфейсов и т.п.) Внешняя шина имеет восемь программно управляемых диапазонов. Для каждого диапазона настраивается собственный режим работы, включая временные характеристики транзакции на шине,

разрядность шины, способ обращения с ожиданием или без сигнала готовности, и способ организации ECC.

Диаграммы работы контроллера при записи и чтении представлены на рисунках 34 и 35 соответственно.

Важно отметить, что при выполнении операции записи шиной данных управляет контроллер внешней шины, а при выполнении операции чтения шиной управляет внешнее подключенное устройство, выставляя свои данные, линии контроллера же находятся в высокоимпедансном состоянии.

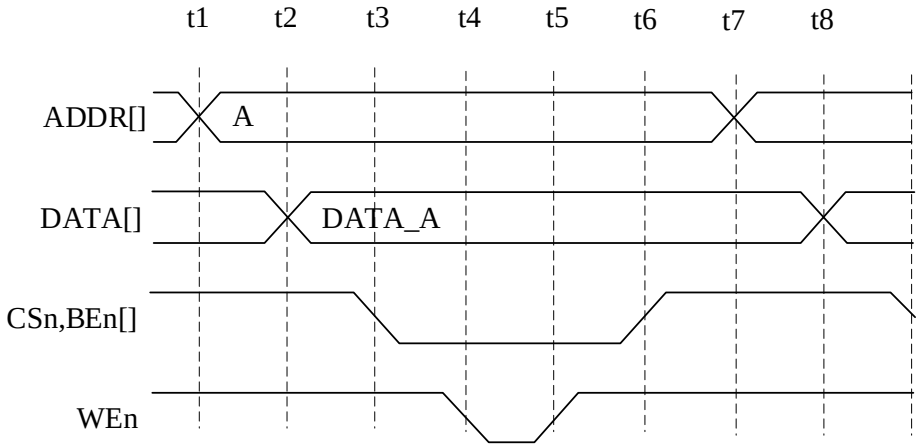


Рисунок 34 – Диаграмма записи на внешней шине

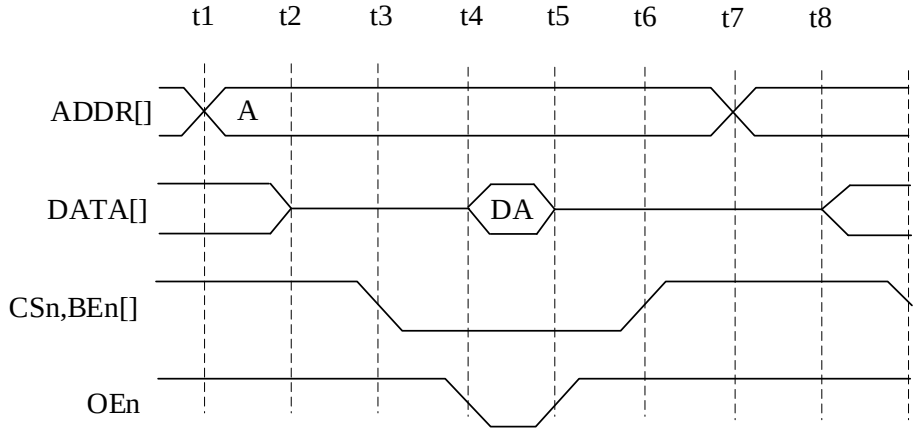


Рисунок 35 – Диаграмма чтения на внешней шине

В момент времени t_1 начинается транзакция.

$t_2 - t_1 = t_{HCLK}$;

$t_3 - t_2 = t_{HCLK}$;

$t_4 - t_3 = t_{HCLK} * (WS_SETUP + 1)$; – Время переустановки сигналов выборки и адреса/данных – фаза переустановки

$t_5 - t_4 = t_{HCLK} * (WS_ACTIVE + 1)$; – Фаза записи/чтения

$t_6 - t_5 = t_{HCLK} * (WS_HOLD + 1)$; – Время удержания адреса/данных – фаза удержания

$t_7 - t_6 = t_{HCLK}$;

$t_8 - t_7 = t_{HCLK}$.

В момент времени t_7 может быть начата следующая транзакция.

Значения параметров WS_SETUP, WS_ACTIVE и WS_HOLD настраиваются в регистре REGION[n].CNTRL.

7.7.1.1 Организация ECC для внешней системной шины

Для каждого из диапазонов внешней системной шины может быть задан режим работы с проверкой ECC или без проверки. Если проводится проверка ECC, то она организована по коду Хэмминга (72,64). При этом проверочные биты ECC могут располагаться как в дополнительных разрядах шины данных (32 данные + 8 ECC – параллельная организация), так и непосредственно в самой 32-х разрядной памяти, начиная с программно задаваемого базового адреса расположения ECC для каждого региона.

7.7.1.2 Параллельная организация ECC

Параллельная организация ECC для внешней системной шины возможна только при организации 64- и 32-разрядной шины данных. При этом дополнительно к каждым 32 разрядам данных добавляется 8 разрядов ECC, то есть при 64-битной шине данных общая разрядность составляет 80 бит, а при 32-разрядной – 40 бит. При настройке 16- или 8-битной шины с параллельной организацией ECC обращение к внешней шине будет приводить к исключению BUSERROR. При 64-х битной организации процессор может за одно обращение считать максимум только 32-битное число, при этом второе 32-битное число может быть «закэшировано» (только если CACHEBLE = 1) внутри контроллера внешней системной шины. При последующем обращении по адресу слова, оно будет считано не из внешней памяти, а из внутреннего регистра. Это позволяет увеличить скорость чтения. Кэширование в контроллере внешней шины возможно только при 64-тбитной организации, при установке бита CACHEBLE = 1 данного региона и при условии, что кэшируемое слово считано без ошибок (нет ни двойных, не одинарных ошибок). Если при чтении «кэшируемого» слова была обнаружена одинарная или двойная ошибка, это слово не будет «закэшировано». В регистр счетчика ошибок, регистр адреса и данных ошибочного слова не заносится информация об ошибках, возникших при чтении кэшируемого слова. В регистры счетчика ошибок, регистр адреса и данных ошибочного слова заносится информация только того слова, которое было запрошено.

Кеширование данных в контроллере внешней шины при 64-битной организации никак не связано с кешированием в блоках кэш памяти DCACHE и ICACHE. Это два различных механизма.

При записи 8- и 16-разрядных слов во внешнюю системную шину обращение выполняется в два этапа:

1 Выполняется считывание из внешней системной шины по адресу обращения данных с ECC, проверка ECC считанных данных;

2 Модификация 8- или 16-разрядного слова в считанном и проверенном по ECC 32-разрядном слове, генерация ECC для модифицированного слова и запись в память.

При записи 32-разрядного слова за один этап проводится генерация ECC и запись в память.

Запись 64-разрядного слова не возможна, либо выполняется двумя записями 32-разрядных слов.

При чтении 8- или 16-разрядных слов из памяти всегда извлекается 32-разрядное слово с ECC (либо два 32-разрядных слова при 64-битной организации). В контроллере внешней шины проверяется ECC, генерируется новое ECC и возвращается процессору, на входе которого ECC проверяется еще раз.

Таблица 126 – Число транзакций на шине EXT_BUS в зависимости от разрядности операции

Тип операции	Разрядность шины	Разрядность данных	Число транзакций на шине
Чтение	64	32,16,8	1 чтение (64-бит DATA и 16-бит ECC)
Чтение	32	32,16,8	1 чтение (32-бит DATA и 8-бит ECC)
Запись	64	32	1 запись (32-бит DATA и 8-бит ECC)
Запись	64	16,8	1 чтение (64-бит DATA и 16-бит ECC) + 1 запись (32-бит DATA и 8-бит ECC)
Запись	32	32	1 запись (32-бит DATA и 8-бит ECC)
Запись	32	16,8	1 чтение (32-бит DATA и 8-бит ECC) + 1 запись (32-бит DATA и 8-бит ECC)

7.7.1.3 **Последовательная организация ECC**

Последовательная организация ECC для внешней системной шины возможна для всех разрядностей шины данных, но при этом контроллер внешней системной шины все равно манипулирует 32-разрядными словами. При 64-битной организации шины, второе 32-разрядное слово выбирается на случай повторного последовательного обращения (если это разрешено в контроллере CACHEBLE = 1). Таким образом, независимо от реализации памяти на внешней системной шине для процессора отображается 32-разрядная память. Контрольные биты ECC для памяти располагаются в старших адресах каждого диапазона, начиная с программно задаваемого базового адреса ECCBASE (младшие четыре бита адреса ECCBASE игнорируются и считаются равными нулю). В зависимости от организации шины их расположение различно. Запись контрольных бит ECC выполняется 8-разрядным словом.

Для 8-, 16- и 32-битных шин данных принцип расположения данных представлен на рисунке 36.

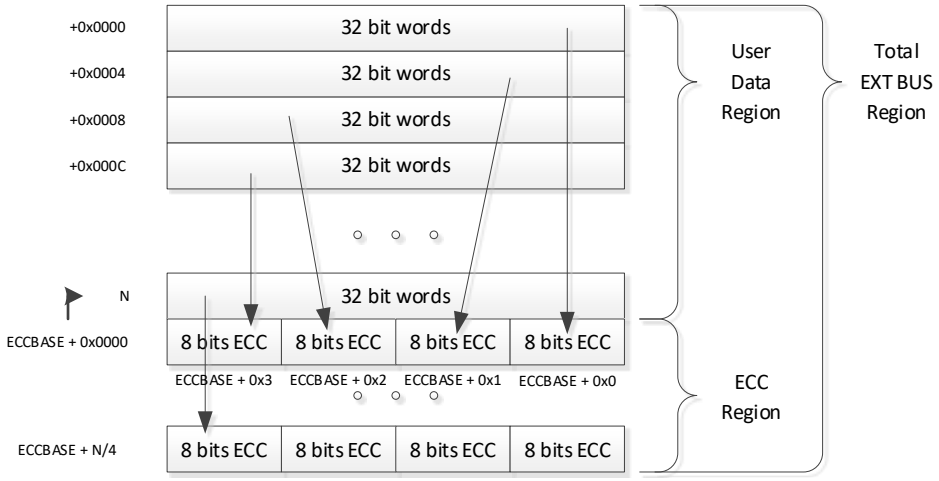


Рисунок 36 – Расположение данных в памяти на шине EXT_BUS
для 32-,16-,8-битных шин

Для 64-битной организации принцип расположения представлен на рисунке 37.

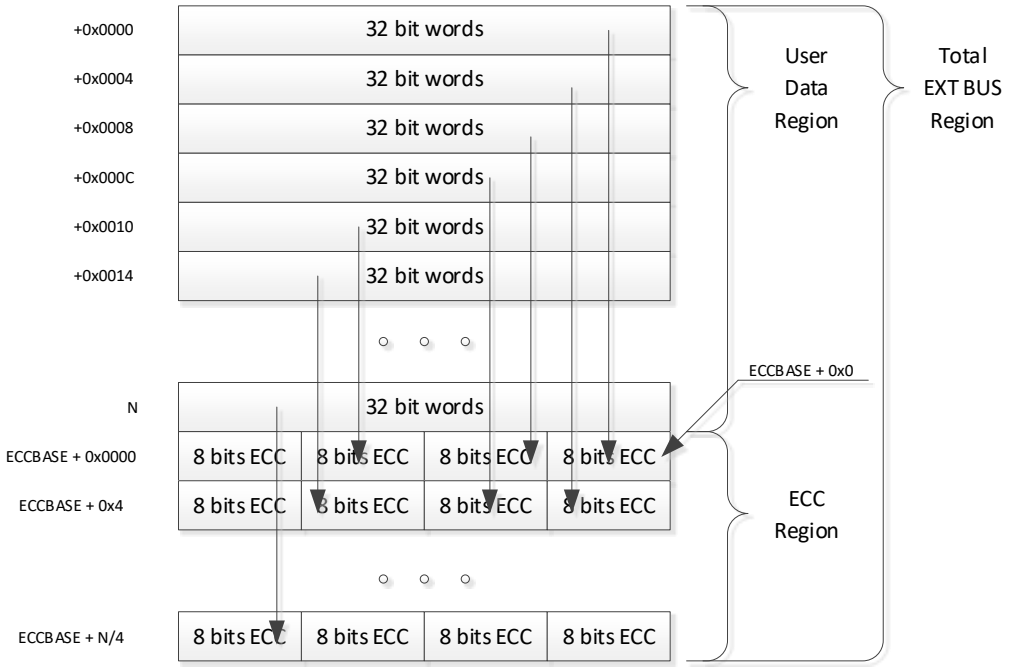


Рисунок 37 – Расположение данных в памяти на шине EXT_BUS при 64-битной шине

Внимание! Биты ECC при последовательной организации располагаются последовательно, начиная с адреса ECCBASE. Биты ECC для четных слов расположены последовательно, но в четных ячейках, начиная с ECCBASE, для нечетных слов – последовательно, но в нечетных ячейках начиная с ECCBASE+0x4. Это необходимо для минимизации повторных обращений при чтении ECC в случае, если проверочные биты ECC от считанных слов попадут в разные ячейки.

Такая организация памяти требует:

– два обращения для считывания четырех 32-разрядных слов при 64-битной организации;

– два обращения для считывания 32-х данных с ECC при 32-разрядной шине данных;

– три обращения при 16-разрядной шине данных;
– пять обращений при 8-разрядной шине данных.

Число обращений для выполнения операций чтения и записи данных различной разрядности при различной организации шины данных представлено в таблице 127

Таблица 127 – Число транзакций на шине EXT_BUS при последовательном ECC

Тип операции	Разрядность шины	Разрядность данных	Число транзакций на шине
чтение	64	32,16,8	1 чтение (64-бит ECC) + 1 чтение (64-бит DATA)
чтение	32	32,16,8	1 чтение (32-бит ECC) + 1 чтение (32-бит DATA)
чтение	16	32,16,8	1 чтение (16-бит ECC) + 2 чтения (16-бит DATA)
чтение	8	32,16,8	1 чтение (8-бит ECC) + 4 чтения (8-бит DATA)
запись	64	32	1 чтение (64-бит ECC) + 1 запись (32-бит DATA) + 1 запись (8-бит ECC)
запись	64	16,8	1 чтение (64-бит ECC) + 1 чтение (64-бит DATA) + 1 запись (32-бит DATA) + 1 запись (8-бит ECC)
запись	32	32	1 чтение (32-бит ECC) + 1 запись (32-бит DATA) + 1 запись (8-бит ECC)
запись	32	16,8	1 чтение (32-бит ECC) + 1 чтение (32-бит DATA) + 1 запись (32-бит DATA) + 1 запись (8-бит ECC)
запись	16	32,16,8	1 чтение (16-бит ECC) + 2 чтения (16-бит DATA) + 2 записи (16-бит DATA) + 1 запись (8-бит ECC)
запись	8	32,16,8	1 чтение (8-бит ECC) + 4 чтения (8-бит DATA) + 4 записи (8-бит DATA) + 1 запись (8-бит ECC)

Область выше ECCBASE, предназначенная для хранения ECC может быть считана/записана как обычные данные. Для обращения в эту область необходимо предварительно перейти в режим работы без контроля ECC. Во избежание порчи данных в этой области, программно может быть установлена защита от записи по адресам выше ECCBASE.

7.7.1.4 Вычисление адреса расположения битов ECC

При параллельной организации биты ECC располагаются в расширении шины данных и считываются одновременно со словом. При последовательной организации биты ECC хранятся в том же массиве памяти, что и проверяемые слова, но начиная с адреса ECCBASE. И для их считывания выполняются дополнительные циклы чтения или записи. При последовательной организации адрес битов ECC вычисляется согласно формулам

ADDRECC[31:0] = ECCBASE[31:0] + {7'b0,ADDR[26:2]};
при 8-, 16-, 32-битной шине данных

ADDRECC[31:0] = ECCBASE[31:0] + {7'b0,ADDR[26:5],ADDR[2],ADDR[4:3]};
при 64-битной шине данных

7.7.1.5 Организация шин без контроля ECC

При отсутствии возможности или необходимости организации контроля ECC для диапазона внешней памяти может быть установлен режим работы без контроля ECC для определенного диапазона. В этом случае данные считываются из внешней памяти без контроля ECC. При этом для возможности кэширования данных для них в контроллере внешней шины будет сгенерирован ECC при возвращении данных процессору. В остальном данный режим работает аналогично режиму с параллельной организацией ECC. При необходимости выполняются дополнительные циклы чтения для формирования 32-разрядных данных, что необходимо учитывать при работе с внешними периферийными блоками, критичными к чтению (FIFO, и т. п.).

Таблица 128 – Число транзакций на шине EXT_BUS в режиме работы без контроля ECC

Тип операции	Разрядность шины	Разрядность данных	Число транзакций на шине
Чтение	64	32,16,8	1 чтение (64-бит DATA)
Чтение	32	32,16,8	1 чтение (32-бит DATA)
Чтение	16	32	2 чтения (16-бит DATA)
Чтение	16	16,8	READ32=0: 1 чтение (16-бит DATA) READ32=1: 2 чтения (16-бит DATA)
Чтение	8	32	4 чтения (8-бит DATA)
Чтение	8	16	READ32=0: 2 чтения (8-бит DATA) READ32=1: 4 чтения (8-бит DATA)
Чтение	8	8	READ32=0: 1 чтение (8-бит DATA) READ32=1: 4 чтения (8-бит DATA)
Запись	64	32,16,8	1 запись (32/16/8-бит DATA)
Запись	32	32,16,8	1 запись (32/16/8-бит DATA)
Запись	16	32	2 записи (16-бит DATA)
Запись	16	16,8	1 запись (16/8-бит DATA)
Запись	8	32	4 записи (8-бит DATA)

Тип операции	Разрядность шины	Разрядность данных	Число транзакций на шине
Запись	8	16	2 записи (8-бит DATA)
Запись	8	8	1 запись (8-бит DATA)

7.7.1.6 **Организация 64-, 32-, 16- и 8-разрядной внешней системной шины**

Для внешней системной шины может быть организована шина данных различной разрядности. При параллельной организации ECC допускается только 64- и 32-разрядная организация шины данных (64 разряда данных + 16 разрядов ECC и 32 разряда данных + 8 разрядов ECC). При последовательной организации ECC может быть задана разрядность шины данных 64, 32, 16 и 8 разрядов. При организации без контроля ECC в контроллере внешней системной шины может быть задана разрядность шины данных 64, 32, 16 и 8 разрядов, но при назначении функций портов микросхемы она может быть сокращена вплоть до 0 бит. В этом случае разряды шины, для которых не задана функция выводов шины данных, внешней системной шины всегда читаются как НОЛЬ.

7.7.1.7 Примеры организаций 64-, 32-, 16- и 8-разрядной внешней системной шины

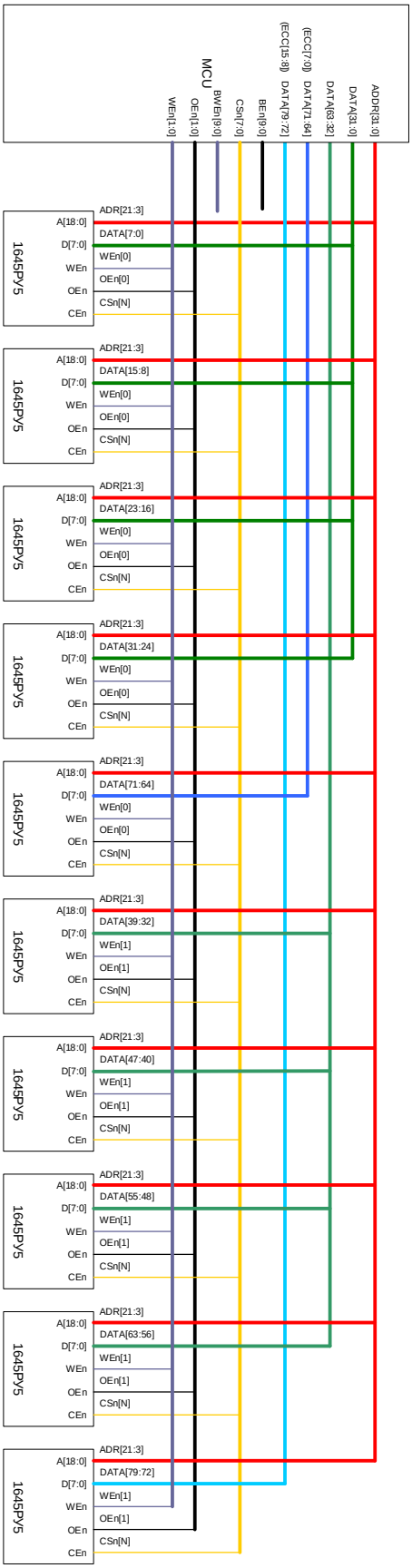


Рисунок 38 – Организация 64-битной шины с параллельной организацией ECC на базе ОЗУ 1645РУ5 (объем пользовательских данных до 4 Мбайт)

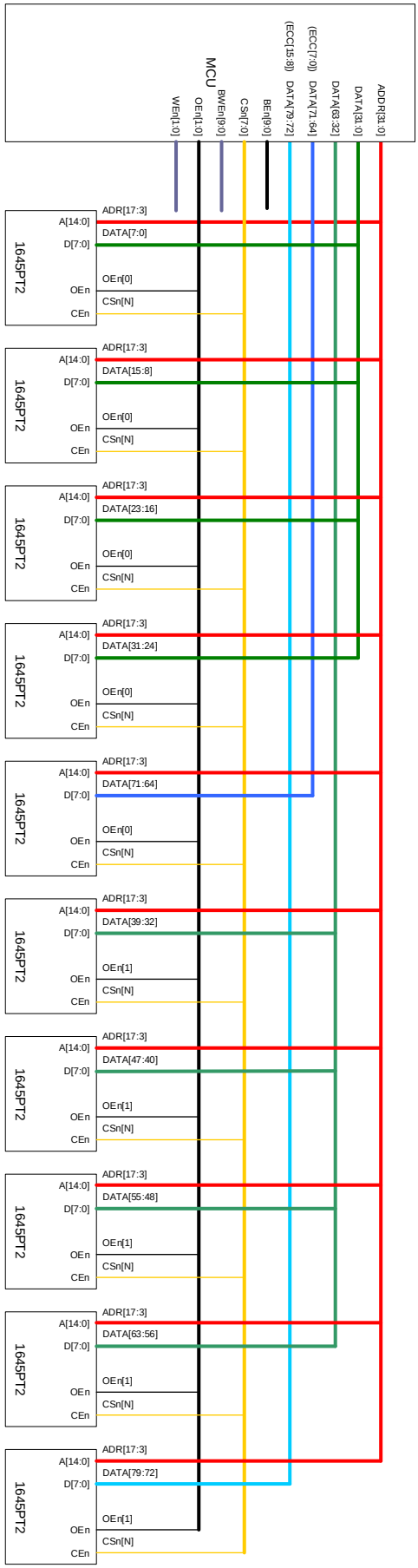


Рисунок 39 – Организация 64-битной шины с параллельной организацией ECC на базе ПЗУ 1645РТ2 (объем пользовательских данных до 256 Кбайт)

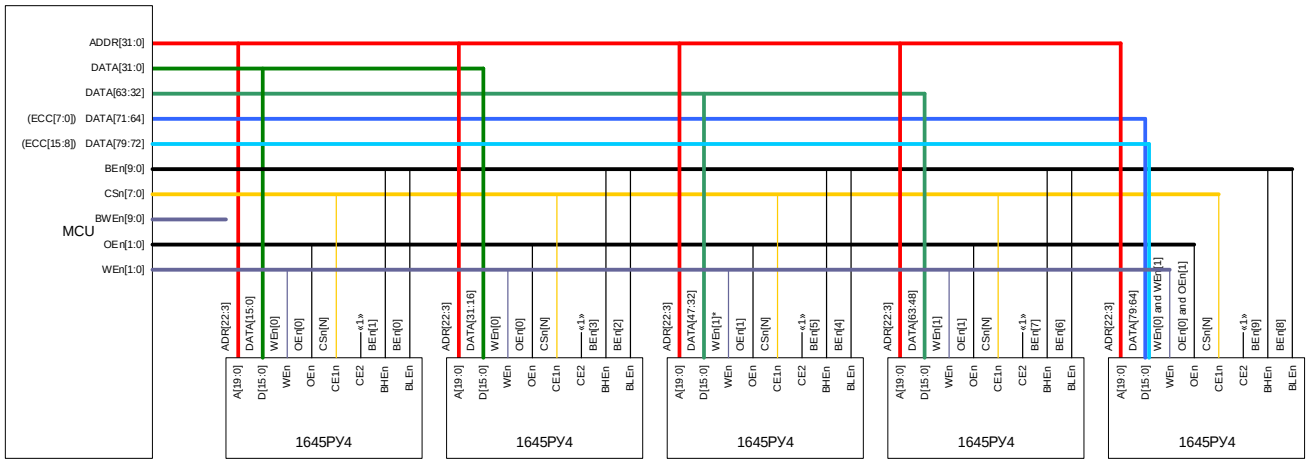


Рисунок 40 – Организация 64-битной шины с параллельной организацией ECC на базе ОЗУ 1645РУ4 (объем пользовательских данных до 8 Мбайт)

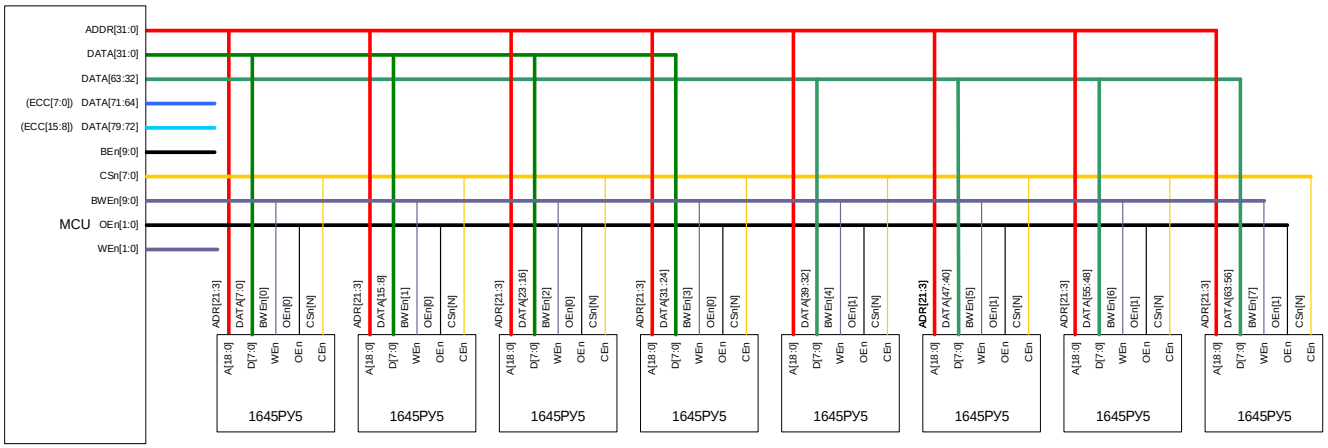


Рисунок 41 – Организация 64-битной шины с последовательной организацией ECC на базе ОЗУ 1645РУ5 (объем пользовательских данных до 3,2 Мбайт и 0,8 Мбайт ECC)

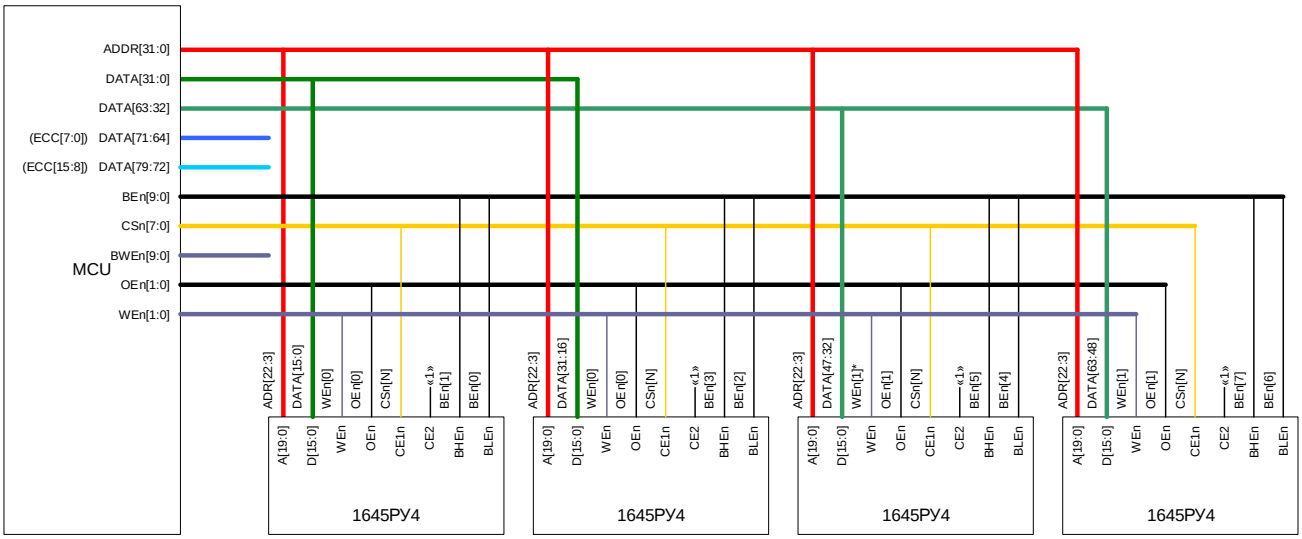


Рисунок 42 – Организации 64-битной шины с последовательной организацией ECC на базе ОЗУ 1645РУ4 (объем пользовательских данных до 6,4 Мбайт и 1,6 Мбайт ECC)

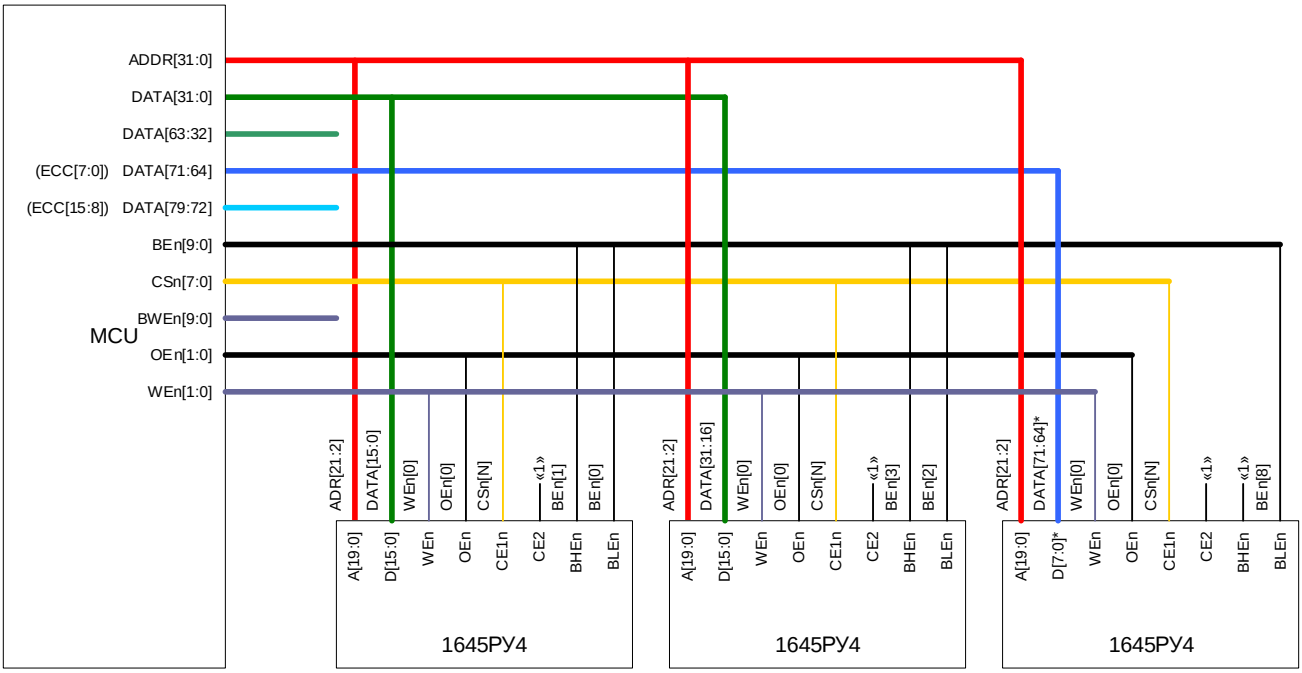


Рисунок 43 – Организация 32-битной шины с параллельной организацией ECC на базе ОЗУ 1645РУ4 (объем пользовательских данных до 4 Мбайт)

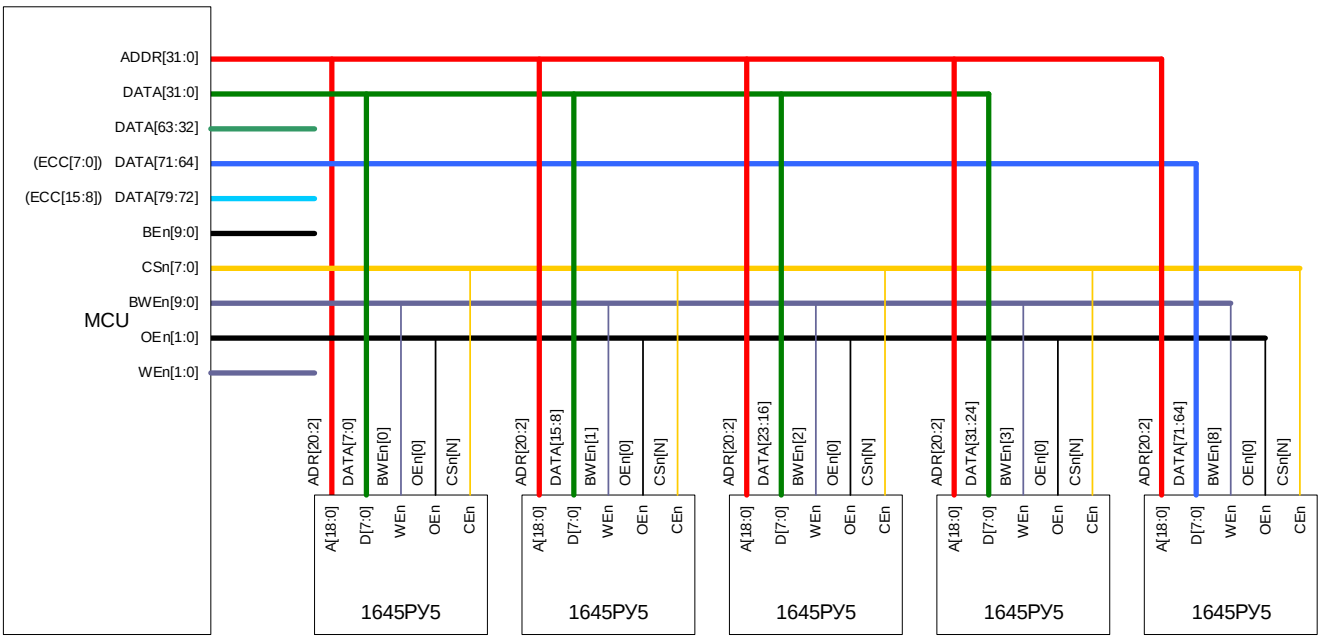


Рисунок 44 – Организация 32-битной шины с параллельной организацией ECC на базе ОЗУ 1645РУ5 (объем пользовательских данных до 2 Мбайт)

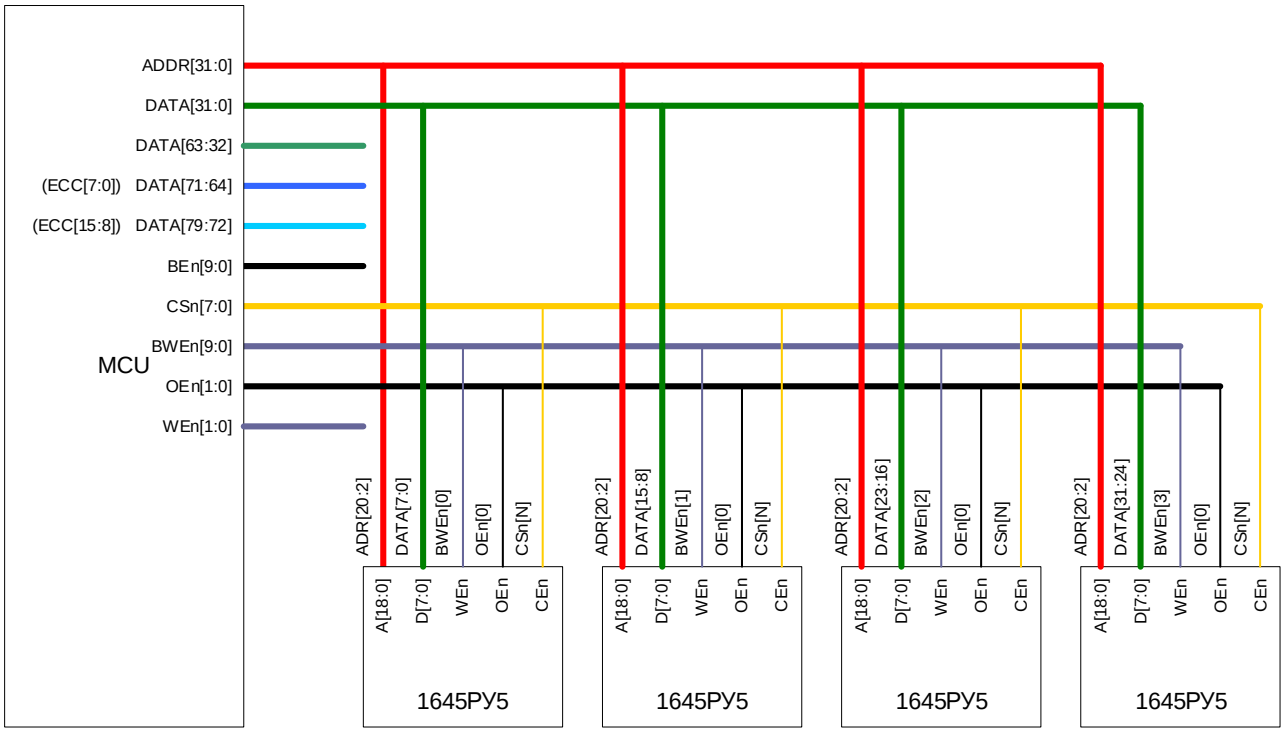


Рисунок 45 – Организация 32-битной шины с последовательной организацией ECC или без ECC на базе ОЗУ 1645РУ5 (объем пользовательских данных до 1,6 Мбайт с последовательной ECC и до 2 Мбайт без ECC)

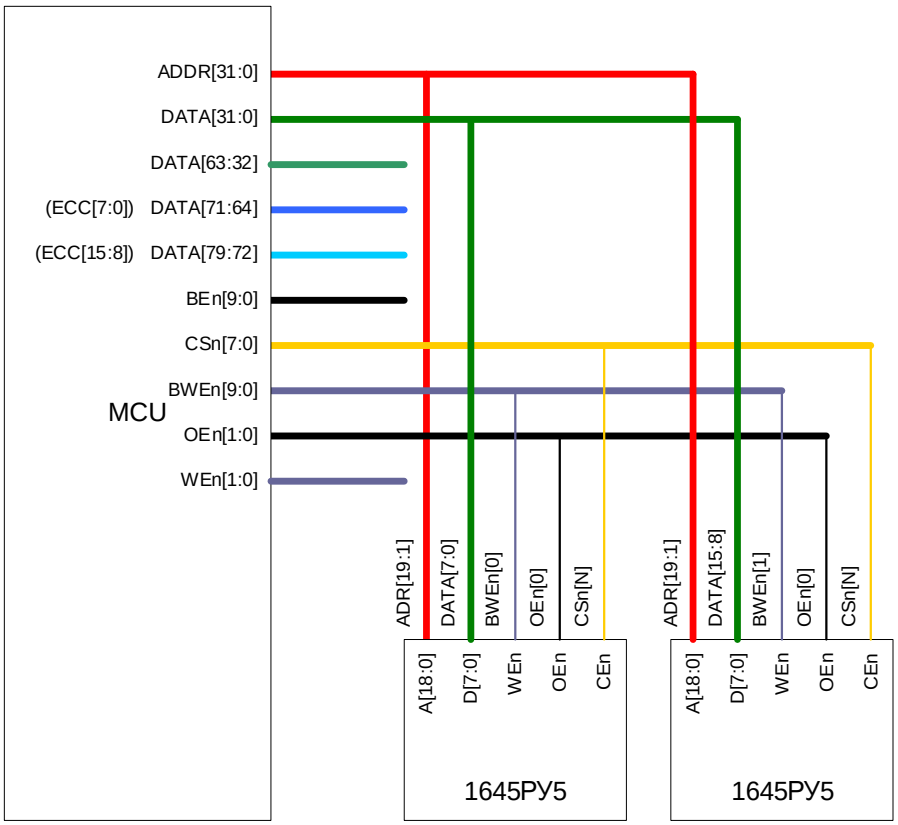


Рисунок 46 – Организация 16-ти битной шины с последовательной организацией ECC или без ECC на базе ОЗУ 1645РУ5 (объем пользовательских данных до 0,8 Мбайт с последовательной ECC и до 1 Мбайт без ECC)

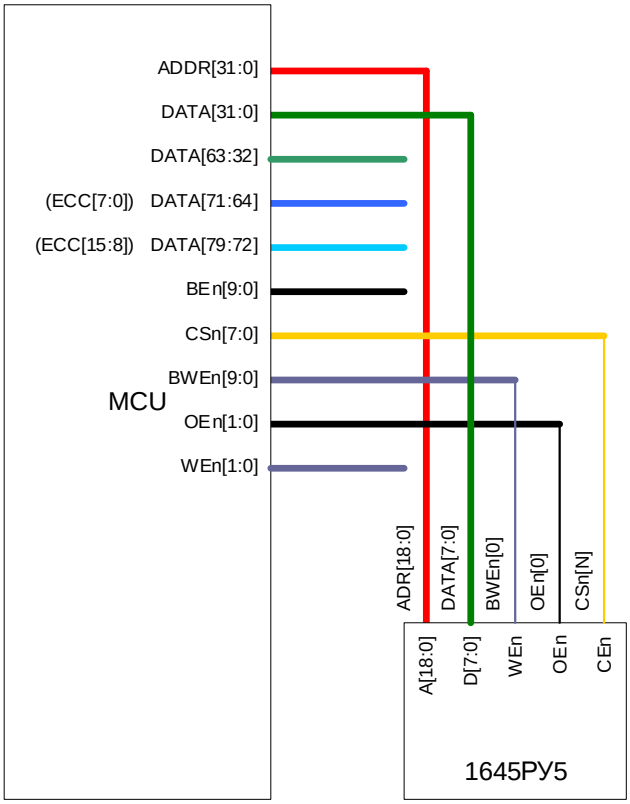


Рисунок 47 – Организация 8-битной шины с последовательной организацией ECC или без ECC на базе ОЗУ 1645РУ5 (объем пользовательских данных до 0,4 Мбайт с последовательной ECC и до 0,5 Мбайт без ECC)

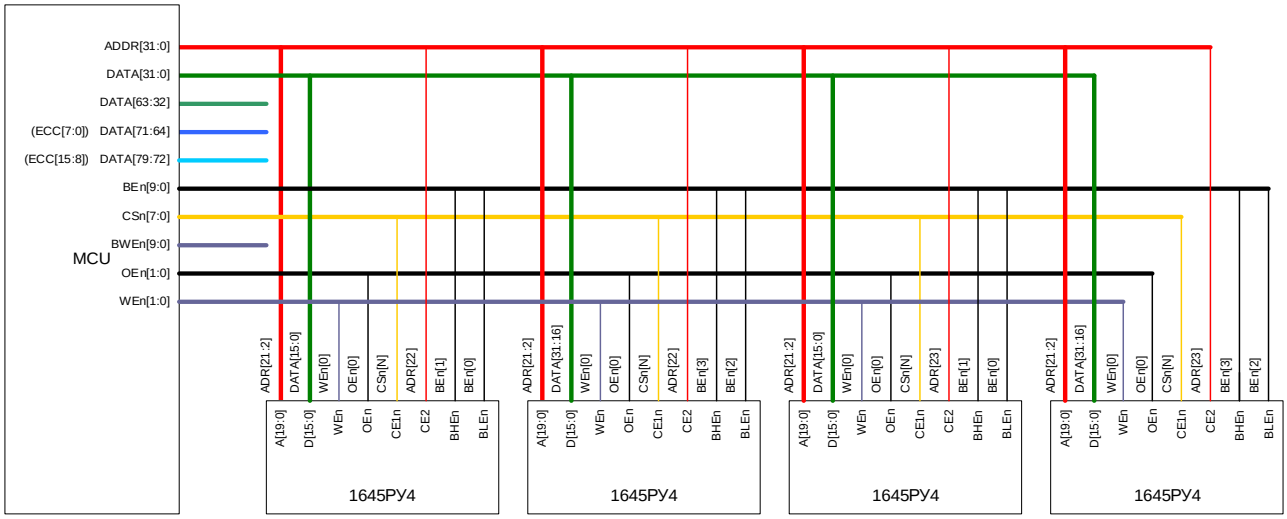


Рисунок 48 – Организация 32-битной шины с последовательной организацией ECC или без ECC на базе ОЗУ 1645РУ4 (объем пользовательских данных до 6,4 Мбайт с ECC или до 8 Мбайт без ECC)

7.7.1.8 **Организация внешней системной шины с ожиданием сигнала готовности READY**

Сигнал готовности READY является входом и позволяет управлять длительностью активной фазы транзакций по внешней системной шине: пассивное состояние это высокий логический уровень, активное состояние - низкий логический уровень. Транзакция на внешней системной шине может быть начата только при пассивном (шина готова) состоянии сигнала READY. При этом будут выполнены фазы предустановки сигналов выборки (SETUP), адреса/данных перед сигналами записи/чтения и вход в фазу записи/чтения (ACTIVE). Переход из фазы записи/чтения в фазу удержания адреса/данных (HOLD) будет выполнен только при наличии пассивного состояния сигнала READY. При наличии активного состояния сигнала READY (шина не готова) контроллер внешней системной шины будет находиться в фазе записи/чтения. В режиме работы с ожиданием сигнала готовности в любом случае для всех фаз выполняются временные настройки их длительности (настройка длительности каждой из трёх фаз выполняется в регистре REGION[n].CNTRL), сигнал готовности при этом может увеличить только длительность фазы записи/чтения. Максимальное время ожидания неактивного уровня сигнала готовности (максимальная длительность активной фазы записи/чтения) составляет 256 тактов частоты HCLK. Если транзакция успешно не выполняется (без учета корректности ECC) за максимальное время ожидания, то транзакция со стороны процессора завершается с флагом неготовности (исключение BusFault). При работе внешней системной шины в режиме без ожидания сигнала готовности сигнал READY всегда считается находящимся в пассивном состоянии.

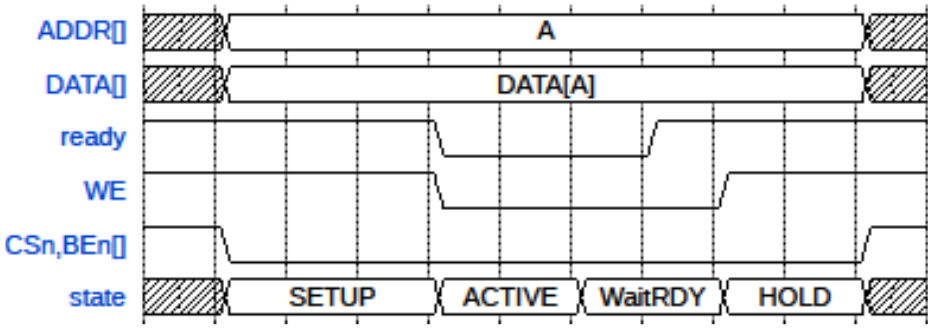


Рисунок 49 – Диаграмма записи

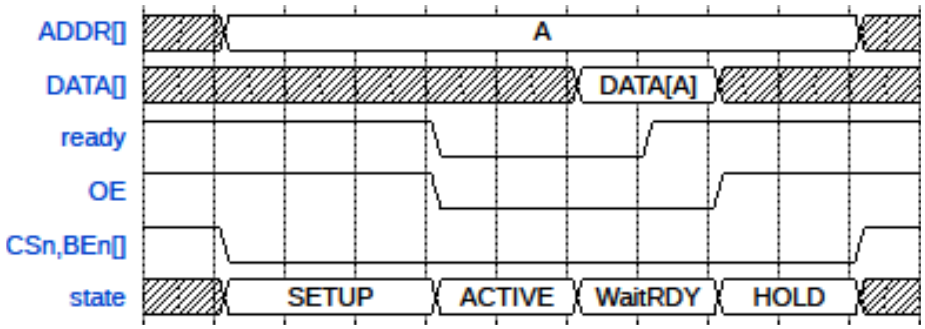


Рисунок 50 – Диаграмма чтения

7.7.2 Описание регистров

Таблица 129 – Описание регистров контроллера внешней шины

Базовый адрес	Название	Состояние после сброса	Описание
0x4000_5000	MDR_EBC		Контроллер внешней шины
Смещение			
0x0000_0000	KEY	0x0000 0000	Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	REGION[0].CNTRL	0x0000 0000	Регистр настройки региона 0
0x0000_0024	REGION[0].ECCBASE	0x0000 0000	Регистр базового адреса таблицы ECC региона 0
0x0000_0044	REGION[0].ECCCS	0x0000 0000	Регистр статуса ошибок ECC региона 0
0x0000_0008	REGION[1].CNTRL	0x0000 0000	Регистр настройки региона 1
0x0000_0028	REGION[1].ECCBASE	0x0000 0000	Регистр базового адреса таблицы ECC региона 1
0x0000_0048	REGION[1].ECCCS	0x0000 0000	Регистр статуса ошибок ECC региона 1
0x0000_000C	REGION[2].CNTRL	0x0000 0000	Регистр настройки региона 2
0x0000_002C	REGION[2].ECCBASE	0x0000 0000	Регистр базового адреса таблицы ECC региона 2
0x0000_004C	REGION[2].ECCCS	0x0000 0000	Регистр статуса ошибок ECC региона 2
0x0000_0010	REGION[3].CNTRL	0x0000 0000	Регистр настройки региона 3
0x0000_0030	REGION[3].ECCBASE	0x0000 0000	Регистр базового адреса таблицы ECC региона 3
0x0000_0050	REGION[3].ECCCS	0x0000 0000	Регистр статуса ошибок ECC региона 3
0x0000_0014	REGION[4].CNTRL	0x0000 0000	Регистр настройки региона 4
0x0000_0034	REGION[4].ECCBASE	0x0000 0000	Регистр базового адреса таблицы ECC региона 4
0x0000_0054	REGION[4].ECCCS	0x0000 0000	Регистр статуса ошибок ECC региона 4
0x0000_0018	REGION[5].CNTRL	0x0000 0000	Регистр настройки региона 5
0x0000_0038	REGION[5].ECCBASE	0x0000 0000	Регистр базового адреса таблицы ECC региона 5
0x0000_0058	REGION[5].ECCCS	0x0000 0000	Регистр статуса ошибок ECC региона 5
0x0000_001C	REGION[6].CNTRL	0x0000 0000	Регистр настройки региона 6
0x0000_003C	REGION[6].ECCBASE	0x0000 0000	Регистр базового адреса таблицы ECC региона 6
0x0000_005C	REGION[6].ECCCS	0x0000 0000	Регистр статуса ошибок ECC региона 6
0x0000_0020	REGION[7].CNTRL	0x0000 0000	Регистр настройки региона 7
0x0000_0040	REGION[7].ECCBASE	0x0000 0000	Регистр базового адреса таблицы ECC региона 7
0x0000_0060	REGION[7].ECCCS	0x0000 0000	Регистр статуса ошибок ECC региона 7
0x0000_0094	ECCADR	0x0000 0000	Регистр адреса последней ошибки ECC
0x0000_0098	ECCDATA	0x0000 0000	Регистр данных последней ошибки ECC
0x0000_009C	ECCECC	0x0000 0000	Регистр ECC последней ошибки ECC

7.7.2.1 **KEY**

Базовый адрес: 0x4000_5000.

Смещение: 0x0000_0000.

Таблица 130 – Регистр ключа, разрешающего модификацию остальных регистров KEY

Номер	31…0
Доступ	RW
Сброс	0
	KEY[31:0]

Таблица 131 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока EXT_BUS_CNTR

7.7.2.2 **REGION[n].CNTRL**

Базовый адрес: 0x4000_5000.

Смещение: 0x0000_0004, 0x0000_0008, 0x0000_000C, 0x0000_0010, 0x0000_0014, 0x0000_0018, 0x0000_001C, 0x0000_0020.

Таблица 132 – Регистр настройки региона [n]

Номер	31…28	27	26	25	24	23…20	19…16
Доступ	RW	RW	RW	RW	RW	RW	RW
Сброс	0	0	0	0	0	0	0
	DIVOCLK[3:0]	ENOCLK		CACHEBLE	-	WS_HOLD[2:0]	WS_SETUP[2:0]

Номер	15…8	7	6	5, 4	3	2	1	0
Доступ	RW	RW	RW	RW	RW	RW	RW	RW
Сброс	0	0	0	0	0	0	0	0
	WS_ACTIVE[7:0]	CPOL	ROM	MODE[1:0]	RDYWAIT	ECCMODE	ECCEN	EN

Таблица 133 – Описание бит регистра REGION[n].CNTRL

Бит	Имя	Описание
31…28	DIVOCLK[3:0]	Биты задания коэффициент деления частоты OCLK: FOCLK = HCLK / ($2^{DIVOCLK+1}$). Примечания 1 Поле реализовано только в регистре REGION0; 2 Значение DIVOCLK может быть в диапазоне от 0 до 14

Бит	Имя	Описание
27	ENOCLK	Бит разрешения выдачи синхросигнала OCLK: 0 – OCLK не формируется; 1 – OCLK формируется. Примечание – Бит реализован только в регистре REGION0
26	READ32	Разрешение считывания 32-разрядных слов при 16- или 8-битной шине (MODE = 01 или 10) в режиме без контроля ECC (ECCEN = 0): 0 – при чтении 8- или 16-битных слов читается только 8- или 16-битные слова. И при возврате процессору ECC не формируется. 1 – при чтении 8- или 16-битных слов всегда читается только 32-битное слов. И при возврате процессору ECC формируется корректно
25	CACHEBLE	Разрешение кэширования внутри контроллера внешней шины при 64-битной организации MODE=11: 0 – кэширование запрещено; 1 – кэширование разрешено (при считывании 64 бит данных, если следующее обращение попадает в выбранное ранее, то транзакции на шине не происходит, и процессору с минимальной задержкой возвращаются запрошенные данные). Примечание – Кэширование доступно только в режиме работы ROM
24	-	Зарезервировано. Записывать всегда 0.
23…20	WS_HOLD[3:0]	Длительность фазы удержания, число тактов: HCLK • (WS_HOLD + 1)
19…16	WS_SETUP[3:0]	Длительность фазы предустановки, число тактов: HCLK • (WS_SETUP + 1)
15…8	WS_ACTIVE[7:0]	Длительность активной фазы, число тактов: HCLK • (WS_ACTIVE + 1)
7	CPOL	Бит полярности сигнала CLOCK в фазе ACTIVE: 0 – передний фронт; 1 – задний фронт
6	ROM	Режим работы региона: 0 – RAM; 1 – ROM. При записи в регион при ROM=1 транзакция завершается с флагом «ошибка на шине»
5…4	MODE[1:0]	Режим организации шины данных: 00 – 32-разрядная шина данных (+8 разрядов при ECCMODE=0 и ECCEN=1); 01 – 16-разрядная шина данных (при ECCEN=0 или при ECCMODE=1и ECCEN=1); 10 – 8-разрядная шина данных (при ECCEN=0 или при ECCMODE=1и ECCEN=1; 11 – 64-разрядная шина данных (+16 разрядов при ECCMODE=0 и ECCEN=1)

Бит	Имя	Описание
3	RDYWAIT	Режим работы с ожиданием флага готовности READY: 0 – режим работы без ожидания готовности, длительность активной фазы определяется битами WS_ACTIVE; 1 – режим работы с ожиданием сигнала готовности, длительность активной фазы не меньше, чем задана битами WS_ACTIVE, и завершается при наличии высокого уровня на линии READY. Если сигнал READY не сформирован за 256 тактов, транзакция завершается с флагом «ошибка на шине» Значение WS_ACTIVE должно быть в диапазоне от 2 до 255, см. описание ошибки 0098 в документе «К1986ВЕ8Т Errata Notice»
2	ECCMODE	Режим ECC: 0 – параллельная организация ECC: – если используется 32-разрядная шина данных, то проверочные 8 бит будут расположены в разрядах шины данных DATA[71:64]; – если используется 64-разрядная шина данных, то проверочные 16 бит будут расположены в разрядах шины данных DATA[79:64]. 1 – последовательная организация ECC (проверочные биты расположены в верхних адресах, начиная с адреса ECCBASE)
1	ECCEN	Бит разрешения контроля ECC: 0 – контроль ECC не выполняется; 1 – контроль ECC выполняется
0	EN	Бит разрешения работы региона: 0 – регион не работает; 1 – регион работает. При обращении в адресное пространство неработающего региона транзакция завершается с флагом «ошибка на шине»
Примечание – Код программы, изменяющий настройки региона внешней системной шины, должен выполняться либо из области внутренней памяти RAMD или памяти OTP, либо из адресного пространства другого региона внешней системной шины. Для применения настроек внешней системной шины необходимо дождаться окончания записи новой конфигурации в регистры REGION[n].CNTRL и REGION[n].ECCBASE с помощью команды барьерной синхронизации данных DSB		

7.7.2.3 **REGION[n].ECCBASE**

Базовый адрес: 0x4000_5000.

Смещение: 0x0000_0024, 0x0000_0028, 0x0000_002C, 0x0000_0030, 0x0000_0034, 0x0000_0038, 0x0000_003C, 0x0000_0040.

Таблица 134 – Регистр базового адреса таблицы ECC региона [n]

Номер	31…0
Доступ	RW
Сброс	0
	ECCBASE[31:0]

Таблица 135 – Описание бит регистра REGION[n].ECCBASE

Бит	Имя	Описание
31…0	ECCBASE[31:0]	Базовый адрес расположения таблицы ECC в регионе при ECCMODE=1 и ECCEN=1, иначе не имеет значения. Базовый адрес должен быть выровнен по границе равной ширине шины данных

7.7.2.4 **REGION[n].ECCCS**

Базовый адрес: 0x4000_5000.

Смещение: 0x0000_0044, 0x0000_0048, 0x0000_004C, 0x0000_0050, 0x0000_0054, 0x0000_0058, 0x0000_005C, 0x0000_0060.

Таблица 136 – Регистр статуса ошибок ECC региона [n]

Номер	31…16
Доступ	RO
Сброс	0
	SECC_CNT[15:0]

Номер	15…8	7	6	5	4	3	2	1	0
Доступ	RO	W1C	W1C	RW	RW	RW	RW	R_W1C	R_W1C
Сброс	0	0	0	0	0	0	0	0	0
	DECC_ CNT[7:0]	CLR_ DCNT	CLR_ SCNT	FIX_ DECC	FIX_ SECC	DECC_IE	SECC_IE	DECC	SECC
Примечание – W1C – Write «1» to Clear. Запись «1» стирает бит									

Таблица 137 – Описание бит регистра REGION[n].ECCCS

Бит	Имя	Описание
31…16	SEEC_CNT[15:0]	Счетчик числа одиночных и двойных ошибок: 0 – нет ошибок; 1 – одна ошибка. … 65535 – 65535 или более ошибок
15:8	DEEC_CNT[7:0]	Счетчик числа двойных ошибок: 0 – нет ошибок; 1 – одна ошибка. … 255 – 255 или более ошибок
7	CLR_DCNT	Бит сброса счетчика двойных ошибок. Запись 1 сбрасывает счетчик DEEC_CNT, при этом не сбрасывает двойные ошибки счетчика SEEC_CNT. Всегда читается как 0
6	CLR_SCNT	Бит сброса счетчика одиночных и двойных ошибок Запись 1 сбрасывает счетчик SECC_CNT. Всегда читается как 0

Бит	Имя	Описание
5	FIX_DECC	Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при двойной ошибке: 0 – запрещено; 1 – разрешено
4	FIX_SECC	Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке: 0 – запрещено; 1 – разрешено
3	DECC_IE	Бит разрешения прерывания при возникновении двойной ошибки ECC: 0 – прерывание запрещено; 1 – прерывание разрешено
2	SECC_IE	Бит разрешения прерывания при возникновении одиночной ошибки ECC: 0 – прерывание запрещено; 1 – прерывание разрешено
1	DECC	Флаг возникновения двойной ошибки: 0 – ошибок не было; 1 – ошибка была. Сбрасывается записью 1
0	SECC	Флаг возникновения одиночной ошибки: 0 – ошибок не было; 1 – ошибка была. Сбрасывается записью 1

7.7.2.5 **ECCADR**

Базовый адрес: 0x4000_5000.

Смещение: 0x0000_0094.

Таблица 138 – Регистр адреса последней ошибки ECC

Номер	31…0
Доступ	RO
Сброс	0
	ECCADR[31:0]

Таблица 139 – Описание бит регистра ECCADR

Бит	Имя	Описание
31…0	ECCADR[31:0]	Адрес последней двойной или одинарной ошибки. Предварительно необходимо разрешить фиксацию данных значений в регистре REGION[n].ECCCS, биты FIX_SECC и FIX_DECC

7.7.2.6 **ECCDATA**

Базовый адрес: 0x4000_5000.

Смещение: 0x0000_0098.

Таблица 140 – Регистр данных последней ошибки ECC

Номер	31…0
Доступ	RO
Сброс	0
	ECCDATA[31:0]

Таблица 141 – Описание бит регистра

Бит	Имя	Описание
31…0	ECCDATA[31:0]	Считанные данные при последней двойной или одинарной ошибке, без корректировки ECC. Предварительно необходимо разрешить фиксацию данных значений в регистре REGION[n].ECCCS, биты FIX_SECC и FIX_DECC

7.7.2.7 **ECCECC**

Базовый адрес: 0x4000_5000.

Смещение: 0x0000_009C.

Таблица 142 – Регистр ECC последней ошибки ECC

Номер	31…8	7…0
Доступ		RO
Сброс		0
	-	ECC[7:0]

Таблица 143 – Описание бит регистра ECC

Бит	Имя	Описание
31…8	-	Зарезервировано
7…0	ECC[7:0]	Считанные биты ECC при последней двойной или одинарной ошибке, без корректировки ECC. Предварительно необходимо разрешить фиксацию данных значений в регистре REGION[n].ECCCS, биты FIX_SECC и FIX_DECC

7.8 Контроллеры кэш-памяти (ICACHECNTR, DCACHECNTR и RAMDCACHECNTR)

В микросхеме для процессора реализованы кэш шины инструкций (I-CACHE) размером 1 Кбайт, и кэш шины данных (D-CACHE) размером 128 байт, и кэш данных памяти RAMD (RAMD-CACHE) размером 128 байт.

Кэш-память включается программно. При выключении питания кэш-память обнуляется. С помощью обращения к регистрам кэш-контроллера, кэш-память может быть сброшена. Кэш-память заполняется автоматически по ходу выполнения программы, алгоритм обновления данных в кэш-памяти построен по принципу FIFO в I-CACHE и PLRU в D-CACHE и RAMD-CACHE. Тактирование кэш-памяти осуществляется всегда с частотой ядра HCLK.

В кэш-памяти I-CACHE и D-CACHE кэшируются данные из адресного пространства OTP и внешней шины EXT_BUS, обращения к которым осуществляется через шины I-Bus и D-Bus, соответственно. В кэш-памяти RAMD-CACHE кэшируются данные из адресного пространства RAMD. При этом не обеспечивается аппаратная поддержка когерентности закэшированных данных, которые могут быть изменены в ходе работы. Обеспечение когерентности этого региона реализуется пользовательской программой путем перезапуска кэш-памяти.

Кэш-память состоит из записей тега адреса и четырех слов данных с кодами ECC.

Тэг Адреса + V + P	Данные 32 + 8ECC бита		Данные 32 + 8ECC бита		Данные 32 + 8ECC бита		Данные 32 + 8ECC бита	
ADDR0	DATA[ADDR0+0x0]	ECC	DATA[ADDR0+0x4]	ECC	DATA[ADDR0+0x8]	ECC	DATA[ADDR0+0xC]	ECC
ADDR1	DATA[ADDR1+0x0]	ECC	DATA[ADDR1+0x4]	ECC	DATA[ADDR1+0x8]	ECC	DATA[ADDR1+0xC]	ECC
ADDR2	DATA[ADDR2+0x0]	ECC	DATA[ADDR2+0x4]	ECC	DATA[ADDR2+0x8]	ECC	DATA[ADDR2+0xC]	ECC
•••	•••		•••		•••		•••	
ADDRm	DATA[ADDRm+0x0]	ECC	DATA[ADDRm+0x4]	ECC	DATA[ADDRm+0x8]	ECC	DATA[ADDRm+0xC]	ECC

Рисунок 51 – Структура записей в кэш памяти

Для I-CACHE m = 64, для D-CACHE и RAMD-CACHE m = 8.

Значение ECC, сохраняемое в кэш-памяти, позволяет в поле данных исправить одинарную ошибку и обнаружить двойную, а также в поле адреса обнаружить одинарную и двойную ошибку.

При исполнении кода программы из OTP в кэш-память записываются строки целиком, то есть из памяти OTP извлекается 160 битное слово (4×32 бита данных и 4×8 бита ECC), которые по отдельному каналу записываются в кэш-память. Данные будут записаны в I- или D-кэш в зависимости от того, по какой шине (I или D) происходит обращение.

При исполнении кода или чтении данных из внешней шины EXT_BUS через шины I-Bus и D-Bus, строки кэш-памяти заполняются последовательно через сами шины I-Bus и D-Bus. При чтении данных из области RAMD строки кэш-памяти RAMD-CACHE также заполняются последовательно. При этом у необновленных слов в строках кэш-памяти не установлен флаг готовности. И в случае обращения по адресу, совпадающему с активным тегом в кэш, но без установленного

флага готовности у необходимого слова, обращение будет пропущено в область внешней шины EXT_BUS или памяти RAMD. Далее при возврате данных они автоматически запишутся в нужную строку кэш-памяти и установят бит готовности. Таким образом, при последующем обращении по этому адресу ответ будет получен уже от кэш-памяти.

Управление кэшированием осуществляется через регистр CNTR соответствующего контроллера кэш-памяти. Бит EN_CACHE разрешает кэширование контроллера кэш-памяти. Для контроллеров I-CACHE и D-CACHE предусмотрены дополнительные биты EN_CACHE_OTP и EN_CACHE_BUS, которые разрешают кэширование соответствующего диапазона памяти. Настройки MPU игнорируются.

7.8.1 Описание регистров

Таблица 144 – Описание регистров контроллера кэш-памяти

Базовый адрес	Название	Состояние после сброса	Описание
0xE004_4000	MDR_ICACHE		Базовый адрес контроллера кэша шины I-Bus
0xE004_5000	MDR_DCACHE		Базовый адрес контроллера кэша шины D-Bus
0x4000_B000	MDR_ RAMDCACHE		Базовый адрес контроллера кэш памяти RAMD
Смещение			
0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	CNTR		Регистр управления контроллером кэш-памяти
0x0000_0008	HIT_CNT		Регистр счетчика числа попаданий Кэш
0x0000_000C	MISS_CNT		Регистр счетчика числа промахов Кэш
0x0000_0010 … 0x0000_001F	-		Зарезервировано
0x0000_0020	ECCCS		Регистр статуса ошибок ECCCACHE
0x0000_0024	ECCADR		Регистр адреса последней ошибки
0x0000_0028	ECCDATA		Регистр данных последней ошибки
0x0000_002C	ECCECC		Регистр контрольной суммы последней ошибки

7.8.1.1 **KEY**

Base ADDR=		0xE004_4000 0xE004_5000 0x4000_B000			Offset=		0x0000_0000								
REG Name:		KEY													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 145 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры контроллера кэш-памяти

7.8.1.2 **CNTR**

Base ADDR=		0xE004_4000 0xE004_5000 0x4000_B000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-										CLR_MISI	CLR_HITI	EN_CACHE_BUS	EN_CACHE_OTP	CLR_CACHE	EN_CACHE

Таблица 146 – Описание бит регистра CNTR

Бит	Имя	Значение	Описание
31…6			Зарезервировано
5	CLR_MIS		Бит сброса регистра счетчика промахов. Сброс выполняется записью 1
4	CLR_HIT		Бит сброса регистра счетчика попаданий. Сброс выполняется записью 1
3	EN_CACHE_BUS		Бит разрешения кэширования данных от EXT_BUS. Имеет значение только при EN_CACHE=1: 0 –данные не кэшируются ; 1 – данные кэшируются. Кэширование осуществляется пословно. Для кэша памяти RAMD должен быть 0
2	EN_CACHE_OTP		Бит разрешения кэширования данных от OTP. Имеет значение только при EN_CACHE=1: 0 –данные не кэшируются ; 1 – данные кэшируются. Кэширование осуществляется строками целиком. Для кэша памяти RAMD должен быть 0
1	CLR_CACHE		Бит сброса кэша. Сброс выполняется записью 1

Бит	Имя	Значение	Описание
0	EN_CACHE		Бит общего разрешения работы кэша: 0 – выключен; 1 – включен

7.8.1.3 **HIT_CNT**

Base ADDR=		0xE004_4000 0xE004_5000 0x4000_B000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Таблица 147 – Описание бит регистра HIT_CNT

Бит	Имя	Значение	Описание
31…0	CNT[31:0]		Счетчик числа попаданий в кэш

7.8.1.4 **MISS_CNT**

Base ADDR=		0xE004_4000 0xE004_5000 0x4000_B000			Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Таблица 148 – Описание бит регистра MISS_CNT

Бит	Имя	Значение	Описание
31…0	CNT[31:0]		Счетчик числа промахов в кэш

7.8.1.5 ECCCS

Base ADDR=		0xE004_4000 0xE004_5000 0x4000_B000			Offset=		0x0000_0020								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECC_CNT[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECC_ CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC

Таблица 149 – Описание бит регистра ECCCS

Бит	Имя	Значение	Описание
31…16	SEEC_CNT[15:0]		Счетчик числа одиночных ошибок: 0 – нет ошибок; 1 – одна ошибка. … 65535 – 65535 или более ошибок
15:8	DEEC_CNT[7:0]		Счетчик числа двойных ошибок: 0 – нет ошибок; 1 – одна ошибка. … 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика двойных ошибок. Запись 1 сбрасывает счетчик DEEC_CNT, при этом не сбрасывает двойные ошибки счетчика SEEC_CNT. Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных и двойных ошибок. Запись 1 сбрасывает счетчик SECC_CNT. Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при двойной ошибке: 0 – запрещено; 1 – разрешено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке. 0 – запрещено; 1 – разрешено

Бит	Имя	Значение	Описание
3	DECC_IE		Бит разрешения формирования запроса к контроллеру FTCNTR при возникновении двойной ошибки ECC: 0 – запрос запрещен; 1 – запрос разрешен
2	SECC_IE		Бит разрешения формирования запроса к контроллеру FTCNTR при возникновении одиночной ошибки ECC: 0 – запрос запрещен; 1 – запрос разрешен
1	DECC		Флаг возникновения двойной ошибки 0 – ошибок не было; 1 – ошибка была. Сбрасывается записью 1
0	SECC		Флаг возникновения одиночной ошибки: 0 – ошибок не было; 1 – ошибка была. Сбрасывается записью 1

7.8.1.6 **ECCADR**

Base ADDR=		0xE004_4000 0xE004_5000 0x4000_B000			Offset=		0x0000_0024								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCADR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCADR[15:0]															

Таблица 150 – Описание бит регистра ECCADR

Бит	Имя	Значение	Описание
31…0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки

7.8.1.7 **ECCDATA**

Base ADDR=		0xE004_4000 0xE004_5000 0x4000_B000			Offset=		0x0000_0028								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCDATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Таблица 151 – Описание бит регистра ECCDATA

Бит	Имя	Значение	Описание
31…0	ECCDATA[31:0]		Считанные данные при последней двойной или одинарной ошибке, без корректировки ECC

7.8.1.8 **ECCECC**

Base ADDR=		0xE004_4000 0xE004_5000 0x4000_B000			Offset=		0x0000_002C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ECC[7:0]							

Таблица 152 – Описание бит регистра ECCECC

Бит	Имя	Значение	Описание
31…8	-		Зарезервировано
7…0	ECC[7:0]		Считанные биты ECC при последней двойной или одинарной ошибке, без корректировки ECC

7.9 Контроллер OTP-памяти программ (OTPCNTR)

В микросхеме реализован один контроллер OTP. Встроенная OTP имеет параллельную организацию кодирования ECC и имеет размерность 4 × 8192 × (32+8ECC) бит. Доступ к шинам и памяти имеет только один контроллер, второй используется только для проверки корректности работы первого. Конфликты между процессорным ядром и DMA контроллера одного номера разрешаются на уровне арбитража коммутатора AHB. Данные OTP могут быть кэшированы в кэш-памяти процессора. Память OTP имеет доступ только на чтение. Попытка записи в OTP вызывает исключение busfaulterror. Из OTP извлекаются четыре слова с адресами ADDR+0x0, ADDR+0x4, ADDR+0x8 и ADDR+0xC. Процессору возвращаются считанные по адресу ADDR данные с ECC. Контроллер OTP различает обращения по шине I и D и в зависимости от того, по какой шине произошло обращение, записывает четыре считанных слова соответственно в I-кэш или D-кэш. После завершения обращения от процессора и отсутствия нового, контроллер начинает упреждающее считывание по адресам ADDR+0x10, ADDR+0x14, ADDR+0x18, ADDR+0x1C. Если в момент считывания приходит обращение в данные адреса, то чтение продолжается до полного выполнения. Если за время упреждающего чтения обращений со стороны процессора не было, то считанные данные фиксируются во внутреннем регистре контроллера OTP, и память OTP переводится в режим пониженного потребления. Если при наличии упреждающе считанных данных приходит обращение по этим адресам, то данные без чтения из OTP выставляются процессору, и начинается новое упреждающее чтение. В зависимости от частоты процессора и времени чтения из памяти OTP, обращение может занимать от одного до восьми тактов.

В старших словах памяти OTP записана уникальная идентификационная информация микросхемы. Данная информация записывается на этапе тестирования и предназначена для идентификации микросхем.

Память OTP имеет топологически распределенную структуру, то есть биты одного слова (32 данных + 8 ECC) максимально разнесены по площади блока. Это позволяет при возникновении единичного точечного воздействия (ТЗЧ) не потерять сразу несколько бит одного слова, а максимум получить единичную ошибку, но во многих словах, вместо множественной ошибки в одном слове.

7.9.1 Методика программирования OTP

Однократнопрограммируемая память программ может быть запрограммирована в двух режимах:

- режим тестирования OTP;
- режим регистрового доступа со стороны процессорного ядра.

В обоих случаях алгоритм программирования одинаков, отличаются только способы формирования сигналов управления при программировании. В режиме тестирования OTP сигналы управления формируются напрямую на выводах микросхемы и отображаются во внутренние сигналы управления блоками памяти OTP. В режиме

регистрового доступа внутренние сигналы управления памяти OTP формируются путем записи соответствующих значений в регистры управления блока OTPCNTR.

Программирование однократнопрограммируемой памяти выполняется в три этапа. Общая блок-схема процесса программирования представлена на рисунке 52.

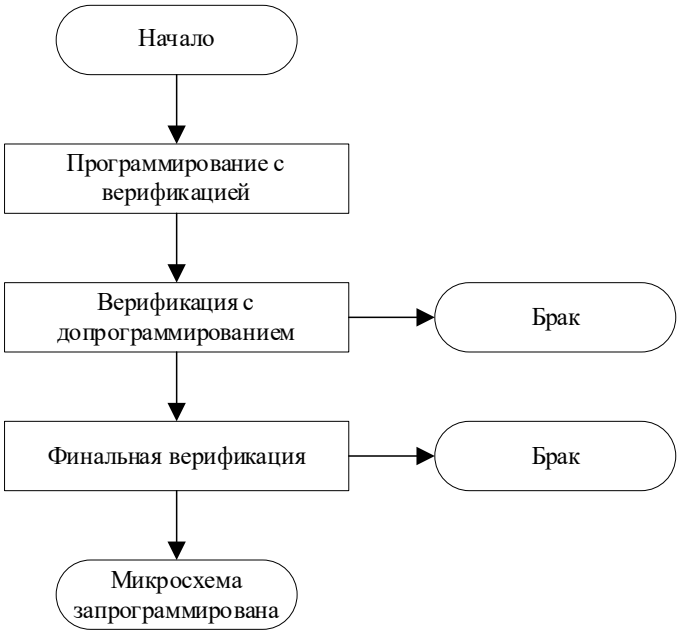


Рисунок 52 – Общая блок-схема процесса программирования

7.9.1.1 **Общие требования к процессу программирования OTP**

В процессе программирования должны быть выполнены требования, приведенные ниже.

При программировании памяти программ в режиме регистрового доступа невозможно исполнение программного кода из памяти OTP. Исполняемый код в этом случае можно расположить, например, в ОЗУ.

Программирование памяти OTP должно проводится в лабораторных условиях при температуре от 15 °С до 45 °С.

Проведение электротермотренировки после программирования микросхем не требуется.

Гарантируется программируемость с коэффициентом программируемости N_{PR}, что необходимо учитывать при выборе метода программирования и монтажа микросхем.

Минимальной программируемой ячейкой памяти является одно 40-битное слово (32 информационных бита и 8 бит корректирующего кода ECC).

При программировании информационные биты и корректирующие биты ECC программируются одинаковым методом. Расположение информационных битов и битов ECC в шине данных блока памяти при программировании в тестовом режиме OTP следующее:

– биты ECC[7:0] соответствуют битам шины данных для записи d[7:0] и q[7:0] шины данных для чтения;

– информационные биты DATA[31:0] соответствуют битам шины данных для записи d[39:8] и q[39:8] шины данных для чтения;

– в режиме регистрового доступа биты ECC[7:0] при записи необходимо устанавливать в поле WECC регистра CNTR блока OTPCNTR;

– в режиме регистрового доступа информационные биты DATA[31:0] при записи необходимо устанавливать в поле DATA регистра WDATA блока OTPCNTR.

Значение восьми бит корректирующего кода вычисляется на основании значения информационных бит и адреса их расположения в соответствии с пунктом 6.6.7 «Функция вычисления ECC для записи в память».

При программировании памяти программ возможно последующее добавление записи в ранее не запрограммированную область. При этом финальная верификация должна выполняться для всех областей, в том числе для ранее запрограммированных.

При программировании памяти OTP программ должны быть запрограммированы также не менее восьми последующих 40-битных слов памяти за последним словом программируемого массива. Это необходимо для исключения возможного возникновения ошибок при упреждающем чтении памяти при выборке инструкций в последнем слове программируемого массива.

7.9.1.2 Этап «Программирование с верификацией»

Этап «Программирование с верификацией» предназначен для программирования основного массива памяти. Алгоритм этапа представлен на рисунке 53.

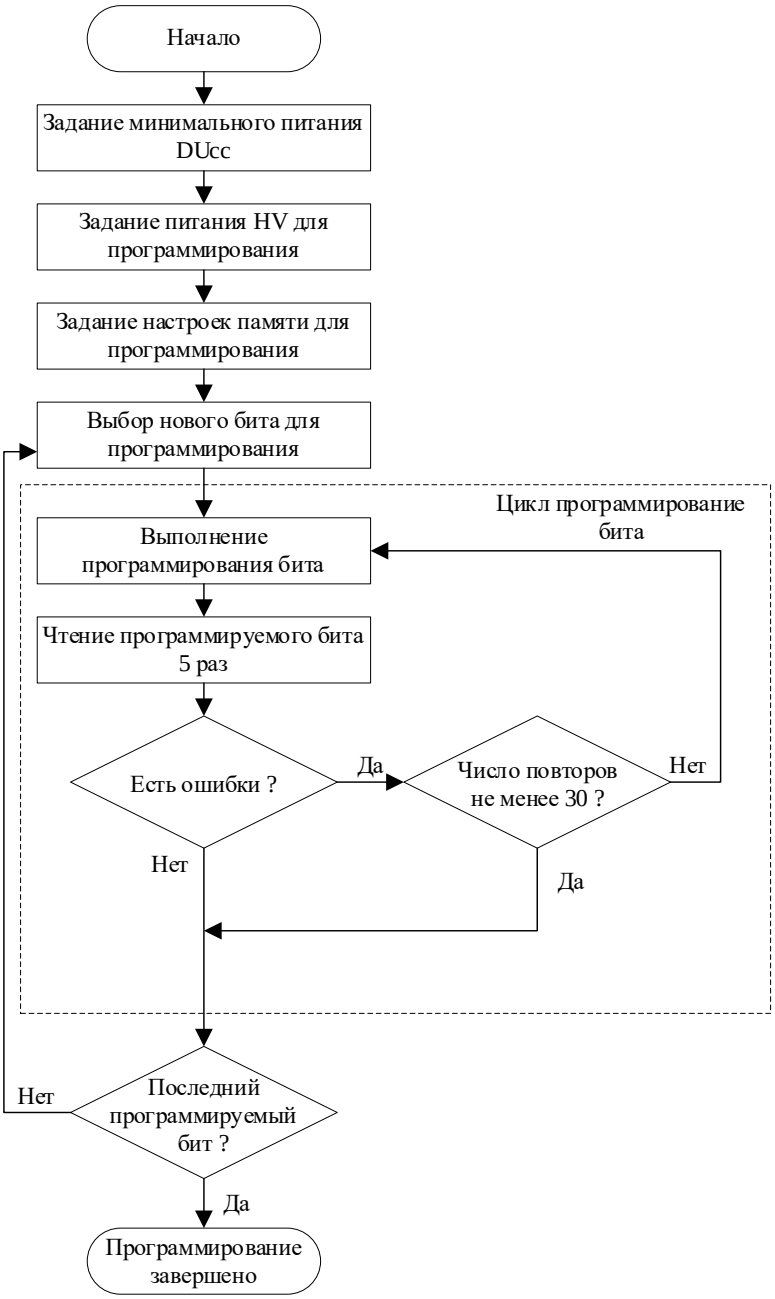


Рисунок 53 – Алгоритм «Программирование с верификацией»

Подэтап «Задание минимального питания DU_{CC}» предназначен для формирования пониженного уровня напряжения DU_{CC}. Снижение напряжения питания может быть осуществлено одним из двух способов:

- перестройкой встроенных регуляторов;
- заданием внешнего питания при отключении встроенных регуляторов.

Перестройка встроенных регуляторов напряжения осуществляется записью значения 3’b0111 в поля TRIM0, TRIM1, TRIM2 и TRIM3 регистров REG_61_TMR0,1,2 батарейного домена BKPCNTR.

Задание внешнего питания при отключении встроенных регуляторов осуществляется путем подачи на выводы микросхемы DU_{CC}0,1,2 и DU_{CC}_B и DU_{CC}_PLL от внешнего источника напряжения питания 1,62 В (с погрешностью задания не ниже 1,60 В и не выше 1,66 В) с последующей подачей на вывод SHDN высокого уровня.

Подэтап «Задание питания HV для программирования» предназначен для формирования высокого уровня напряжения на выводе HV для программирования. Высокое напряжения U_{HVPROG} на выводе HV должно быть сформировано внешним источником питания со значением от 7,6 до 8,0 В. Типовое значение составляет 7,8 В. Схема подачи высокого напряжения программирования на выводе HV должна быть реализована таким образом, чтобы при отсутствии высокого напряжения на выводе формировался уровень напряжения от 2,4 до U_{CC} + 0,2 В.

Подэтап «Задание настроек памяти для программирования» предназначен для перевода памяти в режим, при котором «усложнено» чтение запрограммированных бит, что позволяет более качественно проверить успешность их программирования. Для задания настроек необходимо сформировать на входах repl_tune[3:0] и time_tune[2:0] блоков памяти OTP значения в соответствии с таблицей 153.

Таблица 153 – Значения на входах repl_tune[3:0] и time_tune[2:0]

Настройка	При программировании	При допрограммировании	При финальной верификации	При чтении (типовое значение)
repl_tune[3:0]	4’b 1100	4’b 1100	4’b 1000	4'b 0100 (рекомендуемое значение устанавливается при включении питания)
time_tune[2:0]	3’b 010	3’b 010	3'b 011	3’b 011 (рекомендуемое значение устанавливается при включении питания)

В тестовом режиме OTP данные значения (см. таблицу 153) задаются на выводы PC[31:28] и PA[31:29], в режиме регистрового доступа данные задаются в регистре TEST_TUNING блока OTPCNTR.

Подэтап «Выбор нового бита программирования» предназначен для задания очередного бита, программируемого из 0 в 1. Незапрограммированные биты памяти читаются как 0. Программируются только биты, для которых необходимо установить значение 1. Программирование осуществляется побитно. То есть в каждый момент времени может программироваться только один бит. Для ранее запрограммированных битов необходимо устанавливать нулевое значение при программировании других битов в одном слове.

Подэтап «Выполнение программирования бита» предназначен для формирования диаграммы сигналов управления для программирования одного бита. Диаграмма программирования представлена на рисунке 54.

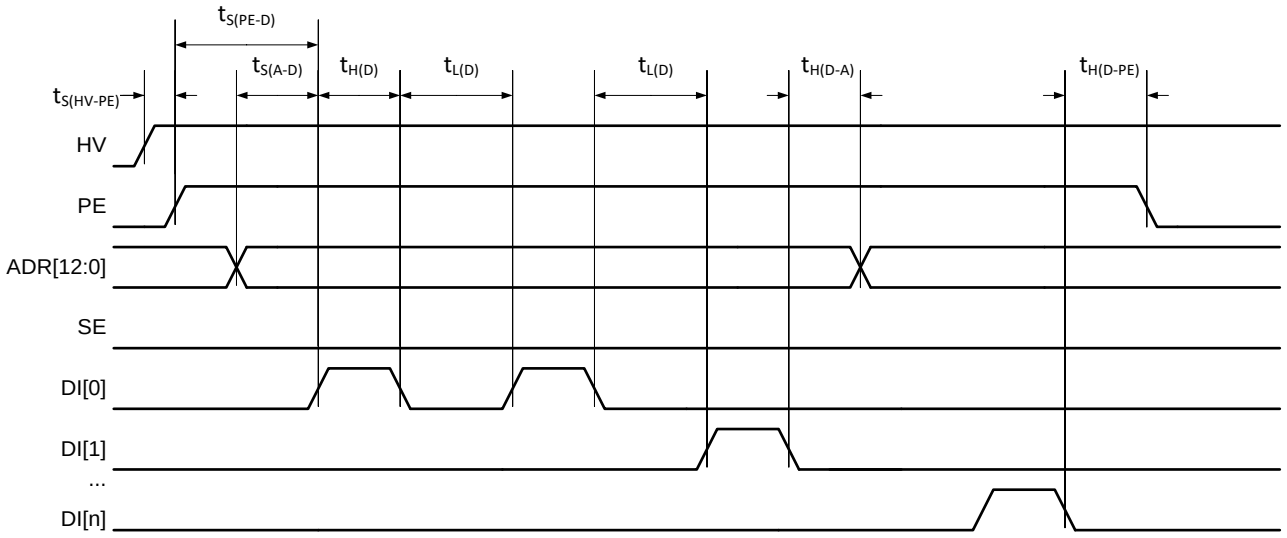


Рисунок 54 – Временная диаграмма процесса программирования памяти OTP

Подэтап «Чтение программируемого бита 5 раз» предназначен для определения результата выполнения программирования бита. Диаграмма чтения представлена на рисунке 55.

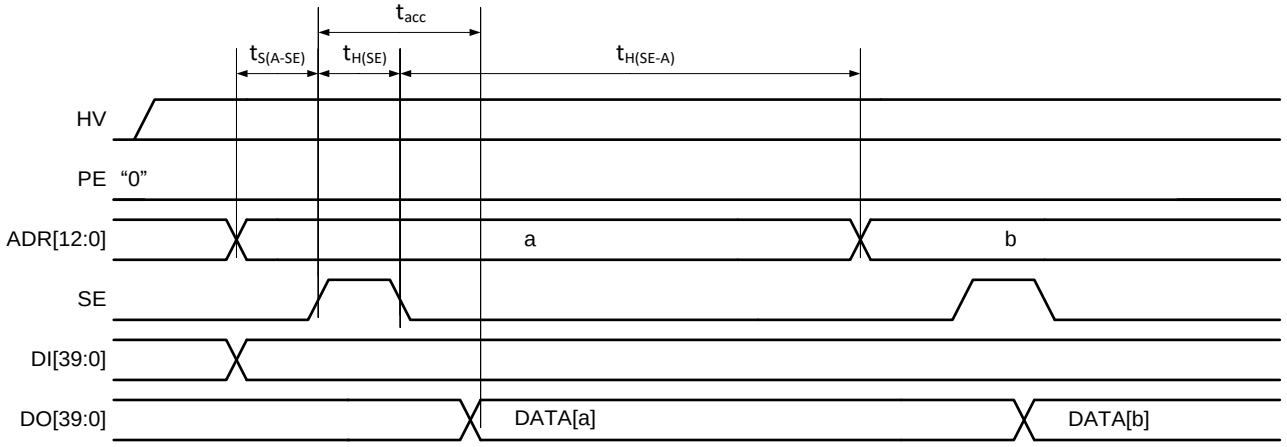


Рисунок 55 – Диаграмма чтения

Чтение должно быть выполнено не менее пяти раз. Если при этом хотя бы один раз запрограммированный бит считался с ошибкой (контролироваться должен только программируемый бит), бит считается не запрограммированным. Необходимо провести повторное программирование бита. При превышении 30 раз числа повторов программирования одного бита можно перейти к программированию следующего бита, не смотря на результат программирования данного бита.

Этап считается выполненным успешно после выполнения всех подэтапов.

7.9.1.3 **Этап «Верификация и допрограммирование»**

Этап «Верификация и допрограммирование» предназначен для проверки запрограммированной информации и допрограммирования «проблемных» битов. При проверке контролируются все биты, в том числе нулевые. Алгоритм процесса представлен на рисунке 56.

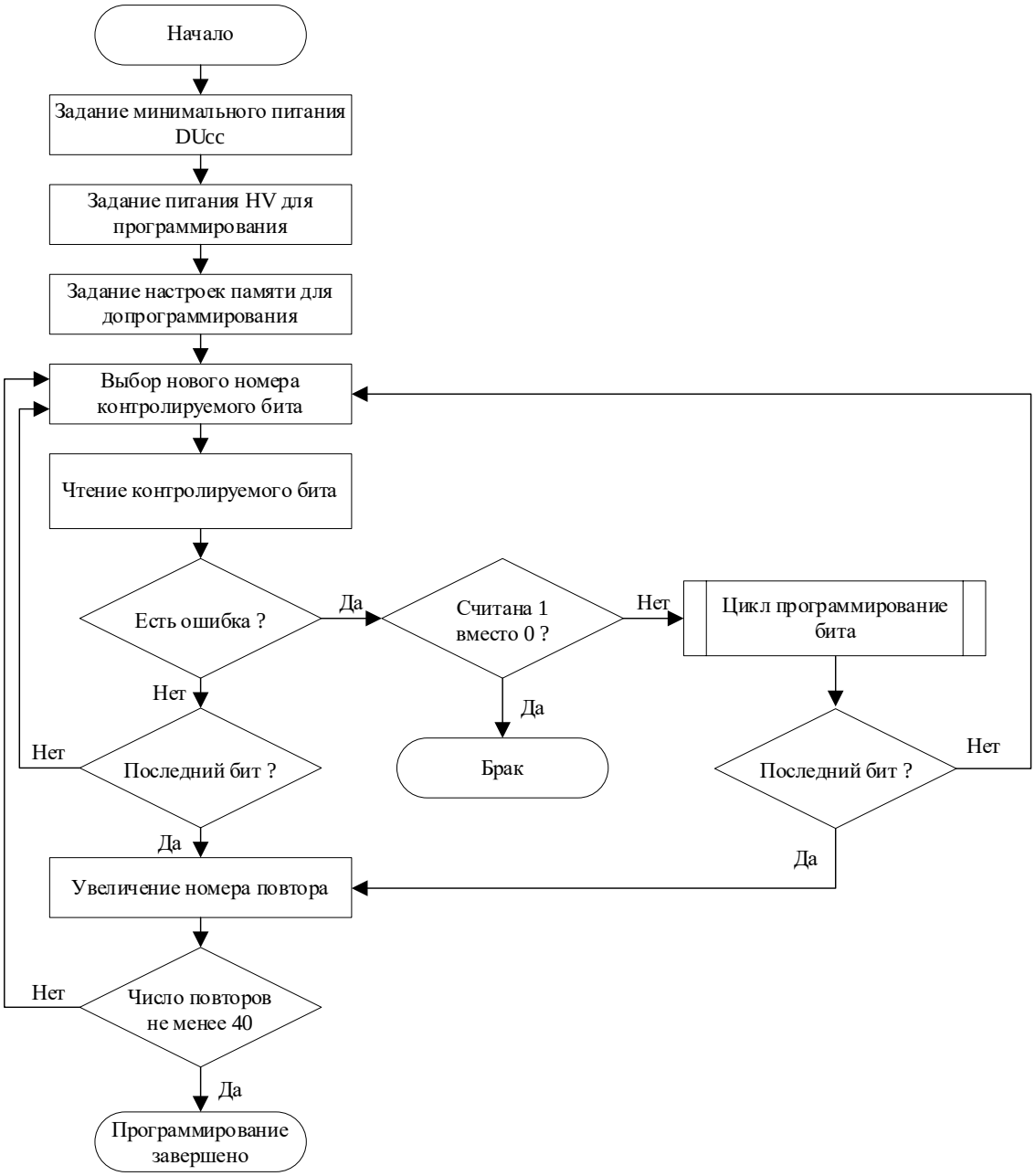


Рисунок 56 – Алгоритм этапа «Верификация и допрограммирование»

Подэтапы «Задание минимального напряжения питания DUcc» и «Задание питания HV для программирования» аналогичны соответствующим подэтапам, рассмотренным в подразделе 7.9.1.2 «Этап «Программирование с верификацией».

Подэтап «Задание настроек памяти для допрограммирования» предназначен для перевода памяти в режим, при котором «усложнено» чтение запрограммированных бит, что позволяет более качественно проверить успешность их программирования. Для задания настроек необходимо сформировать на входах repl_tune[3:0] и time_tune[2:0] блоков памяти OTP значения в соответствии с таблицей 153.

Подэтап «Выбор нового номера контролируемого бита» предназначен для выбора очередного контролируемого бита ранее запрограммированной информации.

Подэтап «Чтение контролируемого бита» предназначен для чтения ранее запрограммированной информации, содержащей выбранный контролируемый бит. Для

контроля битов одной ячейки памяти допускается выполнение однократного чтения ячейки. Если в результате чтения возникла ошибка, и вместо ожидаемого «0» считана «1», процесс программирования бракуется. Если вместо ожидаемой «1» считан «0», для данного бита должен быть выполнен цикл программирования бита, аналогичный циклу, рассмотренному в подразделе 7.9.1.2 «Этап «Программирование с верификацией».

Подэтап «Увеличение номера повтора» предназначен для увеличения номера повтора цикла контроля всей запрограммированной информации.

Этап считается выполненным успешно, если число повторов превысило значение 40.

7.9.1.4 **Этап «Финальная верификация»**

Этап «Финальная верификация» предназначен для проверки запрограммированной информации с учетом времени выборки из памяти. При чтении данных из памяти ОТР, кроме самой информации, должно контролироваться время выборки данных.

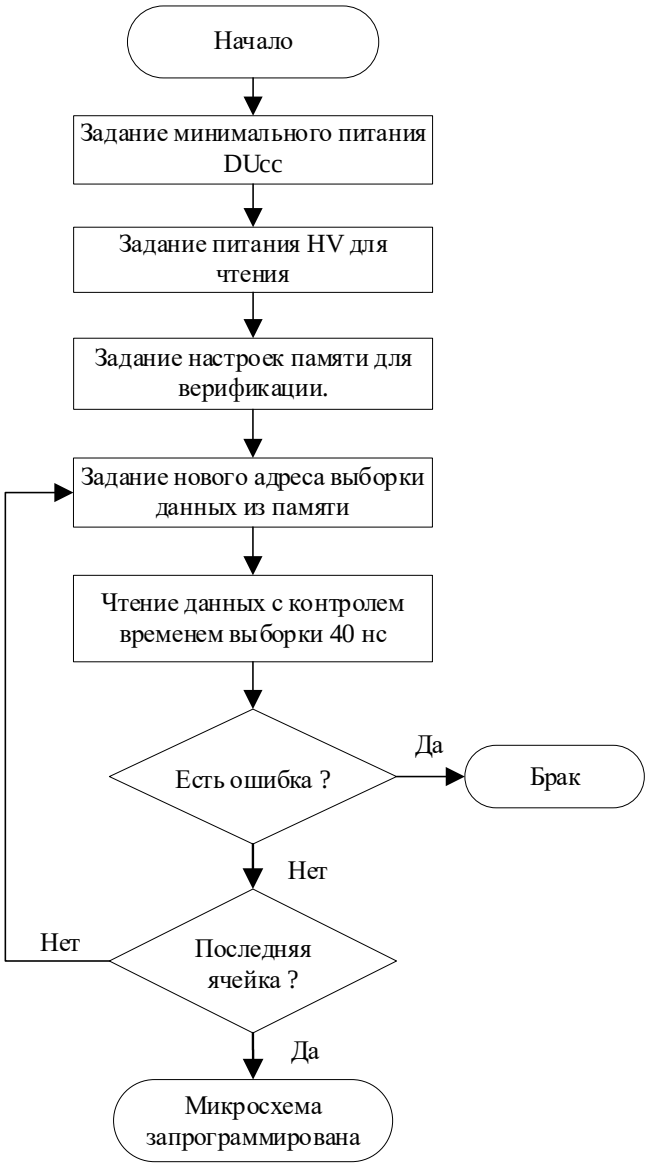


Рисунок 57 – Алгоритм этапа «Финальная верификация»

Подэтап «Задание минимального питания DUcc» предназначен для задания пониженного уровня питания DUcc. В случае использования встроенных регуляторов напряжения должна быть выполнена перестройка встроенных регуляторов напряжения, которая осуществляется записью значения 3’b0111 в поля TRIM0, TRIM1, TRIM2 и TRIM3 регистров REG_61_TMR0,1,2 батарейного домена BKPCNTR.

Подэтап «Задание питания HV для чтения» предназначен для задания на выводе HV микросхемы напряжения в диапазоне от 2,4 до U_{CC} + 0,2 В.

Подэтап «Задание настроек памяти для верификации» предназначен для перевода памяти в режим, при котором обеспечивается «усложненное» чтение запрограммированных бит, по сравнению с чтением при реальной работе. Для задания настроек необходимо сформировать на входах repl_tune[3:0] и time_tune[2:0] блоков памяти OTP значения в соответствии с таблицей 153. Микросхема считается запрограммированной, когда верификация прошла успешно. В противном случае, микросхема попадает в группу непрограммируемых микросхем согласно коэффициенту программируемости Npr.

Подэтап «Задание нового адреса выборки данных из памяти» предназначен для выбора очередного контролируемого слова в памяти.

Подэтап «Чтение данных с контролем времени выборки 40 нс» предназначен для чтения данных из памяти с контролем времени выборки. Данный режим чтения обеспечивается в тестовом режиме OTP путем задания момента стробирования считанных данных с шины данных в 55 нс от момента выдачи высокого уровня SE (55 нс включают время выборки 40 нс из OTP и задержку на выводах микросхемы). Либо путем чтения запрограммированной памяти процессорным ядром при 25 МГц (40 нс) с последующим контролем информации. При этом число тактов ожидания WAITCYCLE должно быть установлено в 1. В тестовом режиме работы памяти OTP возможно контролировать все биты информации, включая биты ECC. В режиме чтения памяти со стороны процессорного ядра будет выполняться автоматическое исправление одиночных ошибок и обнаружение двойных. Таким образом, считанная информация считается верной, если информационные биты совпадают с требуемым значением, и при чтении контроллером OTP не было обнаружено одиночных и двойных ошибок в результате контроля данных по ECC.

Микросхема считается успешно запрограммированной, если вся ранее запрограммированная информация считана без ошибок.

При исполнении программного кода из памяти OTP программ должны быть выставлены типовые настройки уровня питания DU_{CC} и значения на входах repl_tune[3:0] и time_tune[2:0] в соответствии с таблицей 153.

Таблица 154 – Параметры при программировании OTP

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Значение параметра	
		не менее	не более
Напряжение на выводах HV при программировании, В	U_{HVPROG}	7,6	8,0
Напряжение на выводах HV при хранении и чтении, В	U_{HVREAD}	2,4	U_{CC} + 0,2
Время предустановки напряжения на выводах HV перед сигналом PE, мс	$t_{S(HV-PE)}$	10	–
Время предустановки сигнала PE перед данными, мкс	$t_{S(PE-D)}$	300	–
Время предустановки адреса перед данными, мкс	$t_{S(A-D)}$	300	–
Время удержания адреса после данных, мкс	$t_{H(D-A)}$	5	–
Длительность программирования бита данных, мс	$t_{H(D)}$	3	7
Время между программируемыми битами данных, мкс	$t_{L(D)}$	5	–
Время удержания сигнала PE после данных, мкс	$t_{H(D-PE)}$	0	–
Время предустановки адреса перед SE, нс	$t_{S(A-SE)}$	5	–
Время удержания адреса после SE, нс	$t_{H(SE-A)}$	0	–
Длительность сигнала SE, нс	$t_{H(SE)}$	10	–

7.9.2 Параметры контроллера OTP

Контроллер OTP может работать в двух режимах:

– нормальный режим;
– режим регистрового доступа.

Выбор режима осуществляется с помощью бита REG_ACCESS регистра CNTR.

Для корректной работы контроллера в нормальном режиме необходимо установить параметр WAITCYCL регистра CNTR в значение, соответствующее частоте работы CPU_CLK. Изменения данного параметра необходимо производить только в отсутствие обращений к OTP со стороны контроллера. То есть если код выполняется из OTP, процедура установки WAITCYCL должна быть скопирована в ОЗУ и выполнена оттуда. Диаграмма работы контроллера в нормальном режиме представлена на рисунке 58.

Перевод контроллера в регистровый режим позволяет управлять входными сигналами OTP по отдельности и считывать состояние выходных сигналов блоков OTP.

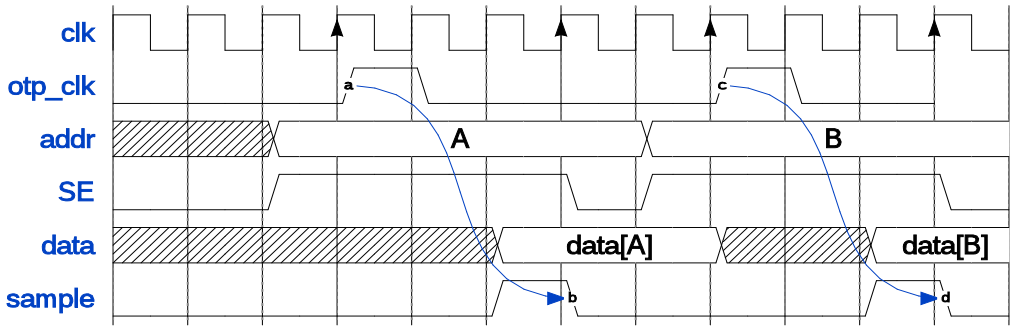


Рисунок 58 – Диаграмма работы контроллера OTP с WAITCYCL=3

7.9.3 Описание регистров

Таблица 155 – Описание регистров контроллера памяти OTP

Базовый адрес	Название	Описание
0x4000_6000	MDR_OTP	Контроллер памяти OTP
Смещение		
0x0000_0000	KEY	Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	CNTR	Регистр управления памяти OTP
0x0000_0008	ADR	Регистр адреса при программировании
0x0000_000C	WDATA	Регистр данных при программировании
0x0000_0010	RDATA	Регистр данных при регистровом считывании
0x0000_0014	ECCCS	Регистр статуса ошибок ECCOTP
0x0000_0018	ECCADR	Регистр адреса последней ошибки
0x0000_001C	ECCDATA	Регистр данных последней ошибки
0x0000_0020	ECCECC	Регистр данных ECC последней ошибки
0x0000_0024	TEST_TUNING	Регистр подстройки параметров блока

7.9.3.1 **KEY**

Base ADDR=		0x4000_6000			Offset=		0x0000_0000								
REG Name:		KEY													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 156 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока OTPCNTR

7.9.3.2 **CNTR**

Base ADDR=		0x4000_6000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RECC[7:0]								WECC[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-									PE	SE	REG_ACCESS	WAITCYCL[3:0]			

Таблица 157 – Описание бит регистра CNTR

Бит	Имя	Значение	Описание
31…24	RECC[7:0]	0x00	Биты ECC при регистровом чтении памяти
23…16	WECC[7:0]	0x00	Биты ECC при программировании WDATA
15…7	-		Зарезервировано
6	PE	0	Сигнал PE для OTP: 0 – хранение; 1 – программирование
5	SE	0	Сигнал SE для OTP: 0 – хранение; 1 – чтение
4	REG_ACCESS	0	Бит перевода OTP память в регистровый доступ для программирования: 0 – доступ осуществляется через I и D шины (выполнение кода); 1 – доступ через регистры периферии по шине S (программирование). При переводе OTP в регистровый доступ исполнение программного кода из OTP невозможно
3…0	WAITCYCL[3:0]	0x3	Регистр задания числа дополнительных циклов ожидания при чтении из памяти программ: 0 – не допустимо; 1 – один дополнительный цикл ожидания (до 20,83 МГц); 2 – два дополнительных цикла ожидания (до 41,66 МГц); 3 – три дополнительных цикла ожидания (до 62,5 МГц); 4 – четыре дополнительных цикла ожидания (до 83,33 МГц); 5 – пять дополнительных циклов ожидания (до 100 МГц). Изменение этого поля во время операции чтения OTP может привести к возврату некорректных данных. Таким образом код производящий модификацию данного поля должен выполняться из другой области памяти – RAM, EXT_BUS, ROM

7.9.3.3 **ADR**

Base ADDR=	0x4000_6000	Offset=	0x0000_0008						
REG Name:									

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADR[15:0]															

Таблица 158 – Описание бит регистра ADR

Бит	Имя	Значение	Описание
31…0	ADR[31:0]		Адрес записываемого слова

7.9.3.4 **WDATA**

Base ADDR=		0x4000_6000			Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Таблица 159 – Описание бит регистра WDATA

Бит	Имя	Значение	Описание
31…0	DATA[31:0]		Данные для записи. При записи в память к ним добавляется поле WECC из регистра CNTR

7.9.3.5 **RDATA**

Base ADDR=		0x4000_6000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Таблица 160 – Описание бит регистра RDATA

Бит	Имя	Значение	Описание
31…0	DATA[31:0]		Данные при регистровом чтении. Проверка ECC для данных не выполняется, поле ECC считанных данных отображено в битах RECC регистра CNTR

7.9.3.6 **ECCCS**

BaseADDR=	0x4000_6000	Offset=	0x0000_0014						

REGName:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECC_CNT[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECC_ CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC

Таблица 161 – Описание бит регистра ECCCS

Бит	Имя	Значение	Описание
31…16	SEEC_CNT[15:0]		Счетчик числа одиночных ошибок: 0 – нет ошибок; 1 – одна ошибка; … 65535 – 65535 или более ошибок. Примечание - Значение счетчика также увеличивается при обнаружении многократной нечетной ошибки. См. пункт 4 в алгоритме работы кода Хэмминга HSIA70a. Ошибка, обнаруженная в поле данных, при этом не исправляется
15…8	DEEC_CNT[7:0]		Счетчик числа двойных ошибок: 0 – нет ошибок; 1 – одна ошибка; … 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика двойных ошибок. Запись 1 сбрасывает счетчик DEEC_CNT, при этом не сбрасывает двойные ошибки счетчика SEEC_CNT. Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных и двойных ошибок Запись 1 сбрасывает счетчик SECC_CNT. Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных последней ошибки при двойной ошибке: 0 – запрещено; 1 – разрешено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных последней ошибки при одинарной ошибке: 0 – запрещено; 1 – разрешено

Бит	Имя	Значение	Описание
3	DECC_IE		Бит разрешения формирования запроса к контроллеру FTCNTR при возникновении двойной ошибки ECC: 0 – запрос запрещен; 1 – запрос разрешен
2	SECC_IE		Бит разрешения формирования запроса к контроллеру FTCNTR при возникновении одиночной ошибки ECC: 0 – запрос запрещен; 1 – запрос разрешен
1	DECC		Флаг возникновения двойной ошибки: 0 – ошибок не было; 1 – ошибка была. Сбрасывается записью 1
0	SECC		Флаг возникновения одиночной ошибки: 0 – ошибок не было; 1 – ошибка была. Сбрасывается записью 1

7.9.3.7 **ECCADR**

Base ADDR=		0x4000_6000			Offset=		0x0000_0018								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCADR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCADR[15:0]															

Таблица 162 – Описание бит регистра ECCADR

Бит	Имя	Значение	Описание
31…0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки. Предварительно необходимо разрешить фиксацию данных значений в регистре ECCCS, биты FIX_SECC и FIX_DECC

7.9.3.8 **ECCDATA**

Base ADDR=		0x4000_6000			Offset=		0x0000_001C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCDATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Таблица 163 – Описание бит регистра ECCDATA

Бит	Имя	Значение	Описание
31…0	ECCDATA[31:0]		Считанные данные при последней двойной или одинарной ошибке, без корректировки ECC. Предварительно необходимо разрешить фиксацию данных значений в регистре ECCCS, биты FIX_SECC и FIX_DECC

7.9.3.9 **ECCECC**

Base ADDR=		0x4000_6000			Offset=		0x0000_0020								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ECC[7:0]							

Таблица 164 – Описание бит регистра ECCECC

Бит	Имя	Значение	Описание
31…8	-		Зарезервировано
7…0	ECC[7:0]		Считанные биты ECC при последней двойной или одинарной ошибке, без корректировки ECC. Предварительно необходимо разрешить фиксацию данных значений в регистре ECCCS, биты FIX_SECC и FIX_DECC

7.9.3.10 **TEST_TUNING**

Base ADDR=		0x4000_6000			Offset=		0x0000_0024								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-												REPL_TUNING[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-					TIME_TUNING[2:0]			-			TE[1:0]		TUNE[2:0]		

Таблица 165 – Описание бит регистра TEST_TUNING

Бит	Имя	Значение	Описание
31…20	-		Зарезервировано
19…16	REPL_TUNING[3:0]	4’b0010	Регистр подстройки OTP: 0000 – Больший запас по чтению «1» и меньший по чтению «0»; 0001 0010 – Значение устанавливается по включению питания; 0100 – Рекомендуемое значение. Устанавливается программно; … 1100 1101 1110 1111 – Меньший запас по чтению «1» и больший по чтению «0»
15…11	-		Зарезервировано
10…8	TIME_TUNING[2:0]	3’b011	Регистр подстройки OTP 000 – Меньше время выборки, меньше общий запас по чтению и «0» и «1»; 001 010 011 – Типовое значение; 100 101 110 111 – Больше время выборки, больше общий запас по чтению и «0» и «1»
7…5	-		Зарезервировано
4, 3	TE[1:0]	2’b00	Биты выбора тестового режима при отбраковочном тестировании
2…0	-		Зарезервировано

7.10 Контроллер ПЗУ (ROMCNTR)

В микросхеме реализован один контроллер ПЗУ. Контроллер ПЗУ не имеет каких-либо функций по управлению блоком ПЗУ. Данный блок используются всегда, и контроллер предоставляет центральному процессору только информацию об обнаруженных ошибках при работе с ПЗУ.

Встроенная загрузочная ПЗУ имеет параллельную организацию кодирования ECC и имеет размерность 2048 x (32+8ECC). Память BOOTROM имеет топологически распределенную структуру, то есть биты одного слова (32 данных + 8 ECC) максимально разнесены по площади блока. Это позволяет при

возникновении единичного точечного воздействия (ТЗЧ) не потерять сразу несколько бит одного слова, а максимум получить единичную ошибку, но во многих словах, вместо множественной ошибки в одном слове.

7.10.1 Описание регистров

Таблица 166 – Описание регистров контроллера ПЗУ

Базовый адрес	Название	Состояние после сброса	Описание
0x4000_9000	MDR_ROM		Контроллер ПЗУ
Смещение			
0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	ECCCS		Регистр статуса ошибок ECCROM
0x0000_0008	ECCADR		Регистр адреса последней ошибки
0x0000_000C	ECCDATA		Регистр данных последней ошибки
0x0000_0010	ECCECC		Регистр данных ECC
0x0000_0014	TEST_TUNING		Регистр подстройки параметров блока

7.10.1.1 **KEY**

Base ADDR=		0x4000_9000			Offset=		0x0000_0000								
REG Name:		KEY													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 167 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока ROM_CNTR

7.10.1.2 **ECCCS**

Base ADDR=		0x4000_9000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECC_CNT[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECC_CNT[7:0]								CLR_ DCNT	CLR_ SCNT	FIX_ DECC	FIX_ SECC	DECC_ IE	SECC_ IE	DECC	SECC

Таблица 168 – Описание бит регистра ECCCS

Бит	Имя	Значение	Описание
31…16	SEEC_CNT[15:0]		Счетчик числа одиночных ошибок: 0 – нет ошибок; 1 – одна ошибка; … 65535 – 65535 или более ошибок
15:8	DEEC_CNT[7:0]		Счетчик числа двойных ошибок: 0 – нет ошибок; 1 – одна ошибка. … 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика двойных ошибок. Запись 1 сбрасывает счетчик DEEC_CNT, при этом не сбрасывает двойные ошибки счетчика SEEC_CNT Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных и двойных ошибок Запись 1 сбрасывает счетчик SECC_CNT. Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при двойной ошибке: 0 – запрещено; 1 – разрешено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке: 0 – запрещено; 1 – разрешено
3	DECC_IE		Бит разрешения формирования запроса к контроллеру FTCNTR при возникновении двойной ошибки ECC: 0 – запрос запрещен; 1 – запрос разрешен
2	SECC_IE		Бит разрешения формирования запроса к контроллеру FTCNTR при возникновении одиночной ошибки ECC: 0 – запрос запрещен; 1 – запрос разрешен
1	DECC		Флаг возникновения двойной ошибки: 0 – ошибок не было; 1 – ошибка была Сбрасывается записью 1.
0	SECC		Флаг возникновения одиночной ошибки: 0 – ошибок не было;

Бит	Имя	Значение	Описание
			1 – ошибка была Сбрасывается записью 1.

7.10.1.3 **ECCADR**

Base ADDR=		0x4000_9000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCADR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCADR[15:0]															

Таблица 169 – Описание бит регистра ECCADR

Бит	Имя	Значение	Описание
31…0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки

7.10.1.4 **ECCDATA**

Base ADDR=		0x4000_9000			Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCDATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Таблица 170 – Описание бит регистра ECCDATA

Бит	Имя	Значение	Описание
31…0	ECCDATA[31:0]		Данные, считанные при последней двойной или одинарной ошибке, без корректировки ECC

7.10.1.5 **ECCECC**

Base ADDR=		0x4000_9000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ECC[7:0]							

Таблица 171 – Описание бит регистра ECCECC

Бит	Имя	Значение	Описание
31…8	-		Зарезервировано
7…0	ECC[7:0]		Считанные биты ECC при последней двойной или одинарной ошибке, без корректировки ECC

7.10.1.6 **TEST_TUNING**

Base ADDR=		0x4000_9000			Offset=		0x0000_0014								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TUNE[2:0]		

Таблица 172 – Описание бит регистра TEST_TUNING

Бит	Имя	Значение	Описание
31…3	-		Зарезервировано
2…0	TUNE	3’b010	Регистр подстройки ПЗУ: 000 – Минимальное время считывания. Минимальный запас по считыванию; 001 010 – Типовое значение; 011 100 101 110 111 – Максимальное время считывания. Максимальный запас по считыванию

7.11 Контроллер ОЗУ (RAMCNTR)

В микросхеме реализован один контроллер RAM. Встроенная статическая ОЗУ RAM имеет размерность 32768 × (32 + 8 ECC) и параллельную организацию кодирования ECC. При записи байта и полуслова в память RAM в контроллере на основании полученных адреса обращения, данных для записи и поля HWECC[7:0] проверяется их корректность. Затем выполняется чтение (32 + 8 ECC) разрядной ячейки, расположенной по адресу обращения, а также проверка ECC, считанных из памяти данных. В корректные считанные данные «вставляется» записываемый байт или полуслово, выполняется генерация ECC для модифицированного 32-разрядного слова и выполняется его запись по адресу обращения. Таким образом, запись байт и полуслов выполняются в два этапа:

1 при обращении на запись 32-разрядного слова в контроллере проверяется ECC записываемого слова;

2 генерируется новое ECC и записывается в память.

Запись 32-разрядного слова выполняются в один этап с проверкой/генерацией ECC. При выполнении операции считывания из встроенного ОЗУ всегда извлекается 32-разрядное слово с восемью битами ECC. В контроллере ОЗУ выполняется проверка считанных данных, которые возвращаются процессору, на входе которого осуществляется еще раз проверка ECC.

Память RAM имеет топологически распределенную структуру, то есть биты одного слова (32 данных + 8 ECC) максимально разнесены по площади блока. Это позволяет при возникновении единичного точечного воздействия (ТЗЧ) не потерять сразу несколько бит одного слова, а получить единичную ошибку во многих словах, вместо множественной ошибки в одном слове.

7.11.1 Описание регистров

Таблица 173 – Описание регистров контроллера ОЗУ

Базовый адрес	Название	Описание
0x4000_8000	MDR_RAMD	Контроллер ОЗУ в области DATA
Смещение		
0x0000_0000	KEY	Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	ECCCS	Регистр статуса ошибок ECCROM
0x0000_0008	ECCADR	Регистр статуса ошибок ECCCRAM
0x0000_000C	ECCDATA	Регистр статуса ошибок ECCDRAM
0x0000_0010	ECCECC	Регистр данных ECC последней ошибки
0x0000_0014	TEST_TUNING	Регистр подстройки параметров блока

7.11.1.1 **KEY**

Base ADDR=			0x4000_8000		Offset=		0x0000_0000								
REG Name:			KEY												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 174 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока RAM_CNTR

7.11.1.2 **ECCCS**

Base ADDR=		0x4000_8000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECC_CNT[15:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECC_CNT[7:0]								CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC

Таблица 175 – Описание бит регистра ECCCS

Бит	Имя	Значение	Описание
31…16	SEEC_CNT[15:0]		Счетчик числа одиночных ошибок: 0 – нет ошибок; 1 – одна ошибка; … 65535 – 65535 или более ошибок
15:8	DEEC_CNT[7:0]		Счетчик числа двойных ошибок: 0 – нет ошибок; 1 – одна ошибка; … 255 – 255 или более ошибок
7	CLR_DCNT		Бит сброса счетчика двойных ошибок. Запись 1 сбрасывает счетчик DEEC_CNT, при этом не сбрасывает двойные ошибки счетчика SEEC_CNT. Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных и двойных ошибок. Запись 1 сбрасывает счетчик SECC_CNT. Всегда читается как 0
5	FIX_DECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при двойной ошибке: 0 – запрещено; 1 – разрешено

Бит	Имя	Значение	Описание
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке: 0 – запрещено; 1 – разрешено
3	DECC_IE		Бит разрешения формирования запроса к контроллеру FTCNTR при возникновении двойной ошибки ECC: 0 – запрос запрещен; 1 – запрос разрешен
2	SECC_IE		Бит разрешения формирования запроса к контроллеру FTCNTR при возникновении одиночной ошибки ECC: 0 – запрос запрещен; 1 – запрос разрешен
1	DECC		Флаг возникновения двойной ошибки: 0 – ошибок не было; 1 – ошибка была. Сбрасывается записью 1
0	SECC		Флаг возникновения одиночной ошибки: 0 – ошибок не было; 1 – ошибка была. Сбрасывается записью 1

7.11.1.3 **ECCADR**

Base ADDR=		0x4000_8000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCADR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCADR[15:0]															

Таблица 176 – Описание бит регистра ECCADR

Бит	Имя	Значение	Описание
31…0	ECCADR[31:0]		Адрес последней двойной или одинарной ошибки

7.11.1.4 **ECCDATA**

Base ADDR=		0x4000_8000			Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCDATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCDATA[15:0]															

Таблица 177 – Описание бит регистра ECCDATA

Бит	Имя	Значение	Описание
31…0	ECCDATA[31:0]		Данные, считанные при последней двойной или одинарной ошибке, без корректировки ECC

7.11.1.5 **ECCECC**

Base ADDR=		0x4000_8000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ECC[7:0]							

Таблица 178 – Описание бит регистра ECCECC

Бит	Имя	Значение	Описание
31…8	-		Зарезервировано
7…0	ECC[7:0]		Считанные биты ECC при последней двойной или одинарной ошибке, без корректировки ECC

7.11.1.6 **TEST_TUNING**

Base ADDR=		0x4000_8000			Offset=		0x0000_0014								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-													TUNE[2:0]		

Таблица 179 – Описание бит регистра TEST_TUNING

Бит	Имя	Значение	Описание
31…3	-		Зарезервировано
2…0	TUNE	3’b010	Регистр подстройки ОЗУ: 000 – Минимальное время считывания/записи. Минимальный запас по считыванию/по записи;

			001 010 – Типовое значение; 011 100 101 110 111 – Максимальное время считывания/записи. Максимальный запас по считыванию/по записи

7.12 Контроллер DMA (DMACNTR)

7.12.1 Основные характеристики контроллера DMA

Основные характеристики и отличительные особенности контроллера DMA:

– 32 канала DMA;

– каждый канал DMA имеет свои сигналы управления передачей данных;

– каждый канал DMA имеет программируемый уровень приоритета;

– каждый уровень приоритета обрабатывается, исходя из уровня приоритета, определяемого номером канала DMA;

– поддержка различного типа передачи данных:

– память – память;

– память – периферия;

– периферия – память;

– поддержка различных типов DMA циклов;

– поддержка передачи данных различной разрядности;

– каждому каналу DMA доступна первичная и альтернативная структура управляющих данных канала;

– все управляющие данные канала хранятся в системной памяти;

– разрядность данных приемника равна разрядности данных передатчика;

– количество передач в одном цикле DMA может программироваться от 1 до 1024;

– инкремент адреса передачи может быть больше чем разрядность данных.

7.12.2 Термины и определения

При описании контроллера DMA используются следующие термины:

Таблица 180 – Термины и определения

Термины/определения	Описание
Альтернативная	Альтернативная структура управляющих данных канала. Для изменения типа структуры данных можно установить соответствующий регистр (см. подраздел 7.12.5 «Структура управляющих данных канала»)
C	Идентификатор номера канала прямого доступа. Например: C=1 – канал DMA 1; C=23 – канал DMA 23
Канал	Возможны конфигурации контроллера с числом каналов до 32. Каждый канал содержит независимые сигналы управления передачей данных, которые могут инициировать передачу данных по каналу DMA
Управляющие данные канала	Структура данных находится в системной памяти. Можно программировать структуру данных так, чтобы контроллер выполнял передачу данных по каналу DMA в желаемом режиме.

	Контроллер должен иметь доступ к области системной памяти, где находится эта информация. Примечание – Любое упоминание в данном разделе структуры данных означает управляющие данные канала
Цикл DMA	Все передачи DMA, которые контроллер должен выполнить для передачи N пакетов данных
Передача DMA	Акция пересылки одного байта, полуслова или слова. Общее количество передач DMA, которые контроллер выполняет для канала
Пинг-понг	Режим работы для выбранного канала, при котором контроллер получает начальный запрос и затем выполняет цикл DMA, используя первичную или альтернативную структуру данных. После завершения этого цикла DMA контроллер начинает выполнять новый цикл DMA, используя другую 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, устанавливающая число передач DMA, которые могут произойти перед сменой арбитража. Количество передач DMA программируется в диапазоне от 1 до 1024 двоичными шагами от 2 в степени 0 до 2 в степени 10
Исполнение с изменением конфигурации	Режим работы для выбранного канала, при котором контроллер получает запрос от периферии и выполняет 4 DMA передачи, используя первичную структуру управляющих данных, которые настраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. После окончания цикла, в случае, если периферия устанавливает новый запрос на обслуживание, контроллер выполняет снова 4 DMA передачи, используя первичную структуру управляющих данных, которые опять перенастраивают альтернативную структуру управляющих данных. Затем контроллер начинает цикл DMA, используя альтернативную структуру данных. Контроллер будет продолжать работать вышеописанным способом до тех пор, пока не прочитает неправильную структуру данных или процессор не установит альтернативную структуру данных для обычного цикла. Контроллер устанавливает флаг dma_done, если окончание подобного режима работы происходит после выполнения обычного цикла

7.12.3 Функциональное описание

Ниже показана упрощенная структурная схема контроллера DMA (Рисунок 59).

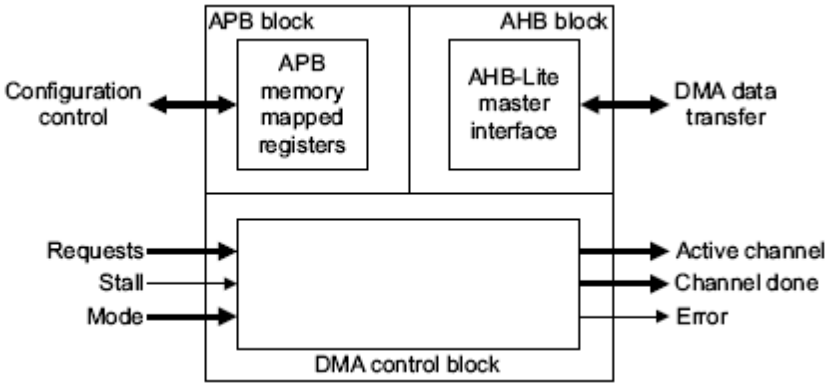


Рисунок 59 – Структурная схема контроллера DMA

Контроллер состоит из следующих основных функциональных блоков:

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

7.12.3.1 **Типы транзакций AHB**

Контроллер интерфейса не поддерживает пакетные (burst) транзакции на шине AHB, выполняет только одиночные (non-sequence) транзакции. Отсутствие возможности осуществлять пакетные транзакции оказывает минимальное влияние на производительность системы, так как пакетные транзакции более эффективны в одноуровневых системах с шиной AHB, где блоки должны «захватывать» шину или обращаться к внешней памяти. В микросхеме контроллер DMA подключен к многоуровневой системе с шиной AHB, включающей встроенную память. Стоит обратить внимание, что количество DMA-передач до проведения арбитража, определяемое числом R, не связано с типом транзакций на шине AHB. Как уже было сказано, контроллер DMA всегда использует одиночные транзакции AHB.

7.12.3.2 **Разрядность передач данных**

Контроллер интерфейса предоставляет возможность осуществлять передачу 8, 16 и 32-разрядных данных. Таблица 181 перечисляет значения комбинаций шины HSIZE.

Таблица 181 – Комбинации шины HSIZE

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

* Сигнал постоянно удерживается в состоянии логический ноль;
** Запрещенная комбинация

Контроллер всегда использует передачи 32-х разрядными данными при обращении к управляющим данным канала. Необходимо устанавливать разрядность данных источника, соответствующую разрядности данных приемника.

7.12.3.3 **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 182 перечисляет значения комбинаций шины HPROT.

Таблица 182 – Режимы защиты данных

HPROT[3] Кэширование	HPROT[2] буферизация	HPROT[1] Привилегированный	HPROT[0] Данные/команда	Описание
-	-	-	1*	Доступ к данным
-	-	0	-	Пользовательский доступ
-	-	1	-	Привилегированный доступ
-	0	-	-	Без буферизации
-	1	-	-	Буферизированный
0	-	-	-	Без кэширования
1	-	-	-	Кэшированный
* Контроллер удерживает HPROT[0] в состоянии логической единицы, чтобы обозначить доступ к данным				

Для каждого цикла DMA возможен выбор режимов защиты данных передач источника и приемника (подробнее см. в пункте 7.12.5 «Структура управляющих данных канала»).

Для каждого канала DMA также возможен выбор режима защиты данных (подробнее см. в пункте 7.12.4 «Управление DMA»).

7.12.3.4 **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 183 перечислены возможные комбинации.

Таблица 183 – Инкремент адреса

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

Минимальная величина инкремента адреса всегда соответствует разрядности передаваемых данных. Максимальная величина инкремента адреса, осуществляемая контроллером, – одно слово. Более подробно о настройке инкремента адреса написано в пункте 7.12.5 «Структура управляющих данных канала». Этот раздел описывает разряды управления величиной инкремента адреса в управляющих данных канала.

Примечание – Если необходимо оставлять адрес неизменным при чтении или записи данных, для примера, при работе с FIFO, можно соответствующим образом настроить контроллер на работу с фиксированным адресом (см. пункт 7.12.5 «Структура управляющих данных канала»).

7.12.4 Управление DMA

7.12.4.1 Правила обмена данными

Контроллер использует правила обмена данными, перечисленные в таблице 184, при соблюдении следующих условий:

– канал DMA включен, что выполняется установкой в состояние логической единицы разрядов управления chnl_enable_set[C] и master_enable;

– флаги запроса dma_req[C] и dma_sreq[C] не замаскированы, что выполняется установкой в состояние логического нуля разряда управления chnl_req_mask_set[C];

– контроллер находится не в тестовом режиме, что выполняется установкой в состояние логического нуля разряда управления int_test_en bit[C].

Таблица 184 – Правила, при которых передача данных по каналам разрешена, и запросы не маскируются

Правило	Описание
1	Если dma_active[C] установлен в 0, то установка в 1 dma_req[C] или dma_sreq[C] на один или более тактов сигнала hclk, следующих или не следующих друг за другом, инициирует передачу по каналу номер C
2	Контроллер осуществляет установку в 1 только одного разряда dma_active[C]
3	Контроллер устанавливает в 1 dma_active[C] в момент начала передачи по каналу C
4	Для типов циклов DMA, отличных от периферийного «Исполнение с изменением конфигурации», dma_active[C] остается в состоянии 1 до тех пор, пока контроллер не окончит передачи с номерами меньше, чем значение 2^R или чем число передач, указанное в регистре n_minus_1. В периферийном режиме «Исполнение с изменением конфигурации», dma_active[C] остается в состоянии 1 в течение каждой пары DMA передач, с использованием первичной и альтернативной структур управляющих данных. Таким образом, контроллер выполняет 2^R передач, используя первичную структуру управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных. По окончании последней передачи dma_active[C] сбрасывается в 0

Правило	Описание
5	Контроллер устанавливает dma_active[C] в 0 на, как минимум, один такт сигнала hclk перед тем, как снова установит dma_active[C] или dma_active[ ] в 1
6	Для каналов, по которым разрешена передача, контроллер осуществляет установку в 1 только одного dma_done[ ]
7	Если dma_req[C] устанавливается в состояние 1 в момент, когда dma_active[C] или dma_stall также в состоянии 1, то это означает, что контроллер обнаружил запрос
8	Если разряды cycle_ctrl для канала установлены в состояние 3’b100, 3’b101, 3’b110, 3’b111, то dma_done[C] никогда не будет установлен в 1
9	Если все передачи по каналу завершены, и разряды cycle_ctrl позволяют удержание dma_done[C], то по срезу сигнала dma_active[ ] произойдут события: – если dma_stall в состоянии 0, контроллер устанавливает dma_done[ ] в состояние 1 продолжительностью один такт hclk; – если dma_stall в состоянии 1, работа контроллера приостановлена. После того, как dma_stall будет установлен в 0, контроллер устанавливает dma_done[ ] в состояние 1 продолжительностью один такт hclk
10	Состояние dma_waitonreq[C] можно изменять только при выключенном канале
11	Если dma_waitonreq[C] в состоянии 1, то сигнал dma_active[C] не перейдет в состояние 0 до тех пор, пока: – контроллер завершит 2^R передач (или число передач, указанное в регистре n_minus_1); – dma_req[C] будет установлен в 0; – dma_sreq[C] будет установлен в 0
12	Если за один такт сигнала hclk перед установкой dma_active[C] в 0 dma_stall устанавливается в 1, то: контроллер установит dma_active[C] в 0 на следующем такте сигнала hclk; передача по каналу C не завершится, пока не будет сброшен в 0 dma_stall
13	Контроллер игнорирует dma_sreq[C], если dma_waitonreq[C] в состоянии 0
14	Контроллер игнорирует dma_sreq[C], если chnl_useburst_set[C] в состоянии 1*
15	Для циклов DMA, отличных по типу от периферийного режима «Исполнение с изменением конфигурации», по окончании 2^R передач контроллер устанавливает значение chnl_useburst_set[C] в состояние 0, если количество оставшихся передач меньше, чем 2^R. В периферийном режиме «Исполнение с изменением конфигурации» контроллер устанавливает значение chnl_useburst_set[C] в состояние 0 только, если количество оставшихся передач с использованием альтернативной структуры управляющих данных меньше, чем 2^R.

Правило	Описание
16	Для типов циклов DMA, отличных от периферийного режима «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, то контроллер выполняет одну DMA передачу. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_waitonreq[C] установлены в 1 и dma_req[C] установлен в 0, контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных. Затем без осуществления арбитража выполняет одну передачу, используя альтернативную структуру управляющих данных
17	Для типов циклов DMA, отличных от периферийного режима «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1, а dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c], и контроллер выполняет 2^R (или число передач, указанное в регистре n_minus_1) DMA передач. В периферийном режиме «Исполнение с изменением конфигурации», если за один такт hclk до установки dma_active[C] в 1 dma_sreq[C] и dma_req[C] установлены в 1, то приоритет предоставляется dma_req[c], и контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных
18	Когда chnl_req_mask_set[C] установлен в 1, контроллер игнорирует запросы по dma_sreq[C] и dma_req[C]
* Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре n_minus_1 меньше, чем значение 2R, то контроллер не очистит разряды chnl_useburst_set, и запросы по dma_sreq[C] будут маскированы. Если периферия не устанавливает dma_req[C] в состояние 1, то контроллер никогда не выполнит необходимых передач	

При отключении канала контролер осуществляет DMA передачи согласно правилам, представленным в таблице 185.

Таблица 185 – Правила, при которых передача данных по каналам разрешена, и запросы не маскируются

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

7.12.4.2 **Диаграммы работы контроллера DMA**

Данный раздел описывает примеры функционирования контроллера с использованием правил обмена данными, представленных в таблице 184:

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

Примечание – Все диаграммы, показанные далее на рисунках в этом подразделе (рисунки 60 – 64), подразумевают следующее:

– hready находится в состоянии 1;
– AHB «ведомый» всегда дает ответ «OKAY».

7.12.4.3 **Импульсный запрос на обработку**

Рисунок 60 показывает временную диаграмму работы контроллера DMA при получении импульсного запроса от периферии.

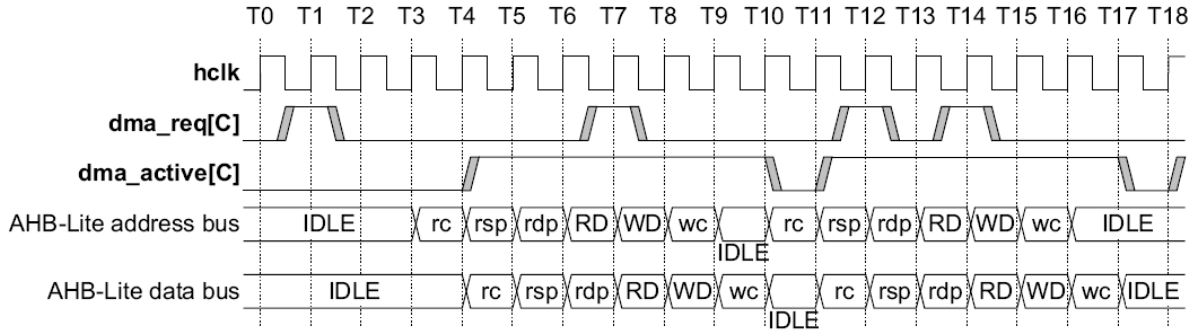


Рисунок 60 – Диаграмма работы при получении импульсного запроса

Пояснения к диаграмме на рисунке 60 приведены в таблице 186.

Таблица 186 – Пояснения к диаграмме работы при получении импульсного запроса

T1	Контроллер обнаружил запрос на обработку по каналу C (таблица 184, правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющую данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7	При установленном dma_active[C] в 1 и при условии, что chnl_req_mask_set[C] находится в состоянии 0, контроллер обнаруживает импульс запроса на обработки по каналу C (см. правило 7). Контроллер обработает этот запрос в течение следующего арбитража

T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)
T10-T11	Контроллер удерживает dma_active[C] на, как минимум, 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 обработки 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)

7.12.4.4 **Запрос на обработку по уровню**

Рисунок 61 показывает временную диаграмму работы контроллера DMA при получении от периферии запроса на обработку по уровню.

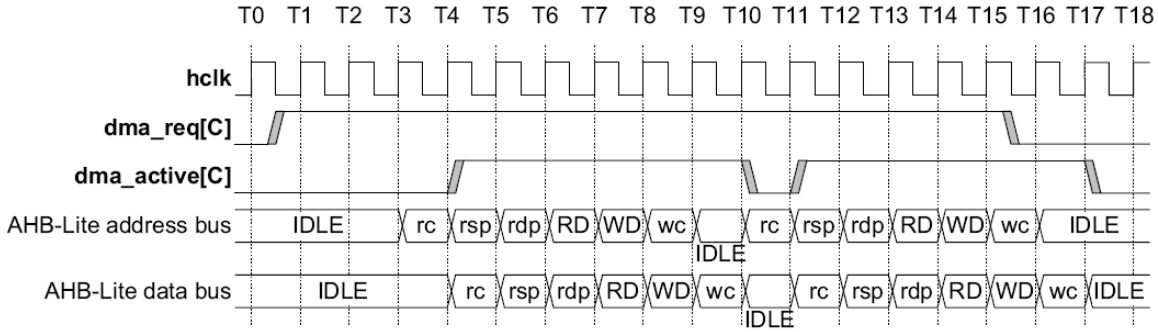


Рисунок 61 – Диаграмма работы при получении запроса на обработку по уровню

Пояснения к диаграмме на рисунке 61 приведены в таблице 187.

Таблица 187 – Пояснения к диаграмме работы при получении запроса на обработку по уровню

T1	Контроллер обнаружил запрос на обработку по каналу C (таблица 184, правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp - чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных
T9-T10	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T10	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4). Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18).
T10-T11	Контроллер удерживает dma_active[C] на как минимум один такт hclk (см. правило 5)
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает dma_active[C] и начинает вторую DMA передачу по каналу C
T11-T14	Контроллер считывает управляющие данные канала
T14-T16	Контроллер выполняет передачу DMA по каналу C
T15-T16	Периферийный блок обнаруживает, что передача DMA началась и сбрасывает dma_req[C]
T16-T17	Контроллер осуществляет запись настроек канала channel_cfg
T17	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 4)

При использовании запроса на обработку по уровню периферийный блок может не обладать достаточным быстродействием, чтобы вовремя снять сигнал запроса, в этом случае он должен установить сигнал dma_stall. Установка сигнала dma_stall предотвращает повторение выполненной передачи.

7.12.4.5 **Флаги завершения**

Рисунок 62 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.

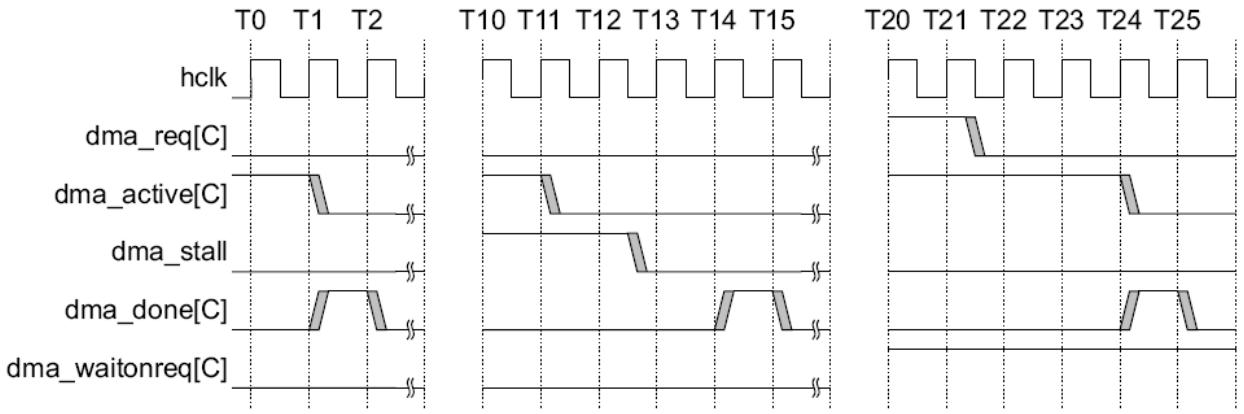


Рисунок 62 – Диаграммы функционирования dma_done

Пояснения к диаграмме на рисунке 62 приведены в таблицах 188, 189, 190.

Таблица 188 – Пояснения функционирования dma_done, такты от T0 до T2

T1	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (таблица 184, 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 189 – 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 190 – 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)

7.12.4.6 **Флаги ожидания запроса на обработку**

На рисунках ниже приведены 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.

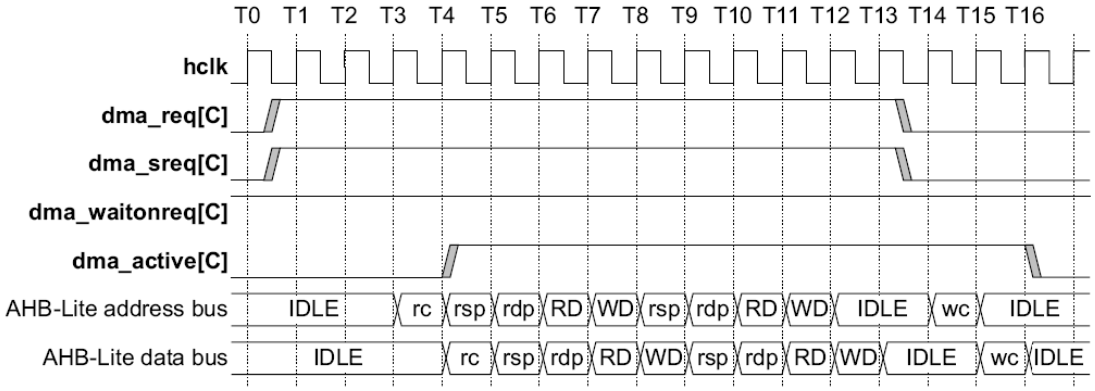


Рисунок 63 – Диаграмма работы контроллера DMA при использовании dma_waitonreq

Пояснения к диаграмме на рисунке 63 приведены в таблице 191.

Таблица 191 – Пояснения работы контроллера DMA при использовании dma_waitonreq

T0-T16	Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10)
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_req_mask_set[C] находится в состоянии 0 (см. правило 18)
T3-T4	Периферийный блок удерживает dma_req[C] и dma_sreq[C] в 1. Контроллер игнорирует dma_sreq[C] запрос и отвечает на dma_req[C] запрос (см. правила 16 и 17)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных
T9-T11	Контроллер считывает 2 указателя адреса окончания данных rsp и rdp
T11-T13	Периферийный блок сбрасывает сигналы dma_req[C] и dma_sreq[C]
T15-T16	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T16	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11). Контроллер устанавливает значение по чтению регистра chnl_useburst_set[C] в 0, если количество оставшихся передач менее 2R (см. правило 15)

На рисунке 64 приведена работа контроллера DMA при установке dma_waitonreq в «1» и выполнении одиночной DMA передачи.

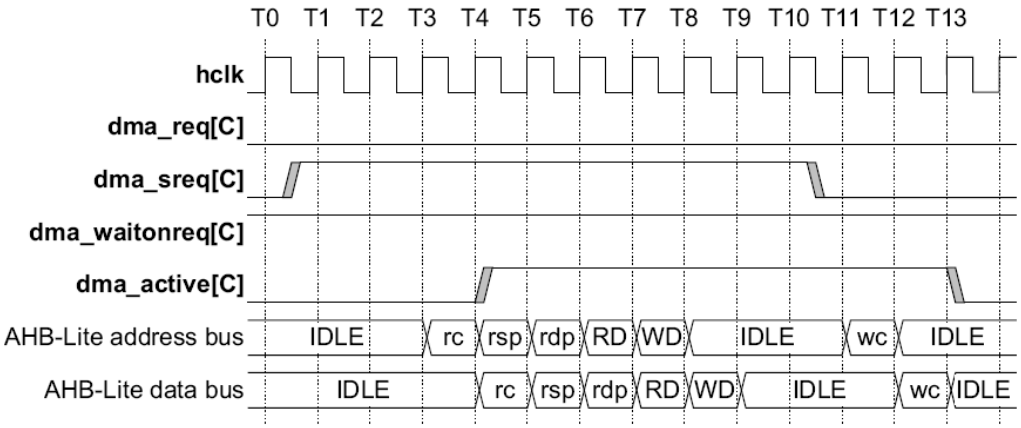


Рисунок 64 – Работа DMA при использовании dma_waitonreq совместно с dma_sreq

Пояснения к диаграмме на рисунке 64 приведены в таблице 192.

Таблица 192 – Пояснения работы DMA при использовании dma_waitonreq совместно с dma_sreq

T0-T13	Периферийный блок должен оставлять состояние dma_waitonreq[C] постоянно (см. правило 10)
T0-T1	Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что chnl_useburst_set[C] находится в состоянии 0 (см. правила 13 и 14)
T3-T4	Контроллер отвечает на dma_sreq[C] запрос (см. правила 16)
T4	Контроллер устанавливает dma_active[C] (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, channel_cfg; rsp – чтение указателя адреса окончания данных источника, src_data_end_ptr; rdp – чтение указателя адреса окончания данных приемника, dst_data_end_ptr
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных; WD – запись данных. Это запрос в ответ на dma_sreq[], таким образом, R=0 и, следовательно, контроллер исполнит 1 DMA передачу
T10-T11	Периферийный блок сбрасывает сигнал dma_sreq[C]
T12_T13	Контроллер осуществляет запись настроек канала, channel_cfg, где wc – запись настроек канала, channel_cfg
T13	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (см. правило 11)

7.12.4.7 **Правила арбитража DMA**

Контроллер имеет возможность настройки момента арбитража при передачах DMA. Эта возможность позволяет уменьшить время отклика при обслуживании каналов с высоким приоритетом.

Контроллер имеет четыре разряда, которые определяют количество транзакций по шине AHB до повторения арбитража. Эти разряды задают степень R числа 2; изменение R напрямую устанавливает периодичность арбитража как 2 в степени R. Для примера, если R равно 4, то арбитраж будет проводиться через каждые 16 передач DMA.

Таблица 193 показывает возможную периодичность арбитража.

Таблица 193 – Периодичность арбитража в единицах передач по шине AHB

Значение R	Периодичность арбитража каждые x передач DMA
b0000	1
b0001	2
b0010	4
b0011	8
b0100	16
b0101	32
b0110	64
b0111	128
b1000	256
b1001	512
b1010-b1111	1024

Внимание! 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 в случае, 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 оставшиеся N передач в конце цикла DMA.

Разряды степени R числа 2 находятся в структуре управляющих данных канала. Местонахождение этих разрядов описано в пункте 7.12.5 «Структура управляющих данных канала».

7.12.4.8 **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, и уровень приоритета снижается с увеличением номера канала. В таблице 194 приведен уровень приоритета каналов DMA в порядке его уменьшения.

Таблица 194 – Уровень приоритета каналов DMA

Уровень приоритета в порядке его уменьшения	Номер канала	Уровень приоритета, установленный битом chnl_priority_set
Наивысший уровень приоритета	0	Высокий
-	1	Высокий
-	2	Высокий
…….	……	……
-	30	Высокий
-	31	Высокий
-	0	По умолчанию (низкий)
-	1	По умолчанию (низкий)
-	2	По умолчанию (низкий)
…….	……	……
-	30	По умолчанию (низкий)
Низший уровень приоритета	31	По умолчанию (низкий)

После окончания цикла DMA контроллер выбирает следующий для обслуживания канал из всех включенных каналов DMA. На рисунке 65 приведен процесс выбора следующего канала для обслуживания.

Рисунок 65 – Алгоритм выбора следующего канала для обслуживания

7.12.4.9 **Типы циклов DMA**

Разряды cycle_ctrl определяют, как контроллер будет выполнять циклы DMA. Описание значений этих разрядов приведено в таблице 195.

Таблица 195 – Типы циклов DMA

cycle_ctrl	Описание
b000	Структура управляющих данных канала в запрещенном состоянии
b001	Обычный цикл DMA
b010	Автозапрос
b011	Режим «пинг-понг»
b100	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
b101	Работа с памятью в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала
b110	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием первичных управляющих данных канала
b111	Работа с периферией в режиме «Исполнение с изменением конфигурации» с использованием альтернативных управляющих данных канала
Примечание – Разряды cycle_ctrl находятся в области памяти, отведенной под channel_cfg – см. пункт 7.12.5 «Структура управляющих данных канала»	

Для всех типов циклов DMA повторный арбитраж происходит после 2^R передач DMA. Если установить длинный период арбитража на низкоприоритетном канале, то это заблокирует все запросы на обработку от других каналов до тех пор, пока не будут выполнены 2^R передач DMA по данному каналу. Поэтому, устанавливая значение R, необходимо учитывать, что это может привести к повышенному времени отклика на запрос на обработку от высокоприоритетных каналов.

Данный раздел описывает следующие **типы циклов DMA**:

- недействительный;
- основной;
- автозапрос;
- «пинг-понг»;
- работа с памятью в режиме «исполнение с изменением конфигурации»;
- работа с периферией в режиме «исполнение с изменением конфигурации».

Недействительный

После окончания цикла DMA контроллер устанавливает тип цикла в значение «недействительный» для предотвращения повтора выполненного цикла DMA.

Основной

В этом режиме контроллер работает только с основными или альтернативными управляющими данными канала. После того, как разрешена работа канала, и контроллер получил запрос на обработку, цикл DMA выглядит следующим образом:

1 Контроллер выполняет 2^R передач. Если число оставшихся передач 0, контроллер переходит к шагу 3.

2 Осуществление арбитража:

если высокоприоритетный канал выдает 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.

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.

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 66 демонстрирует пример функционирования контроллера в режиме «пинг-понг».

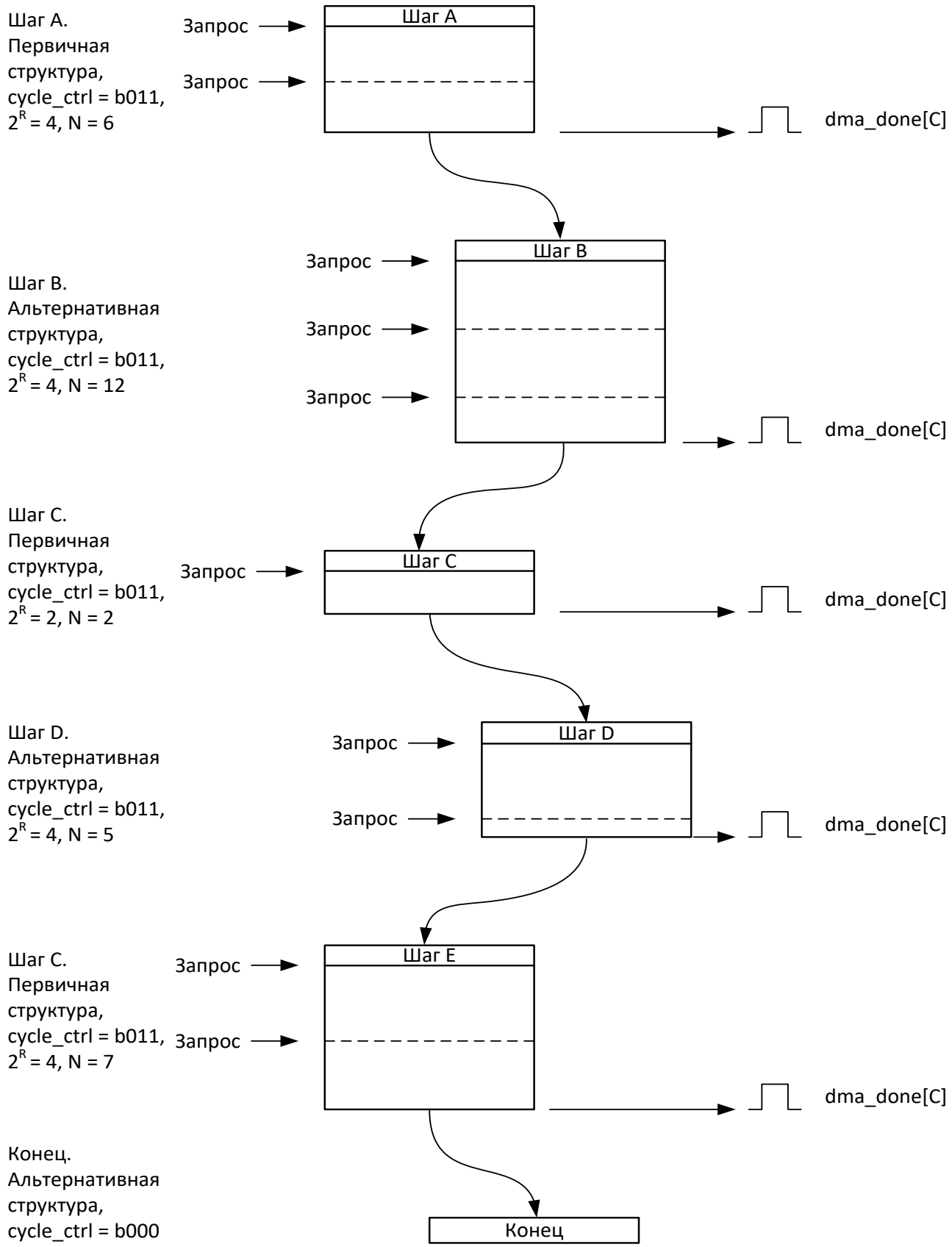


Рисунок 66 – Пример функционирования контроллера в режиме «пинг-понг»

Пояснения к схеме на рисунке 66:

Шаг A Процессор устанавливает первичную структуру управляющих данных для шага A.
Процессор устанавливает альтернативную структуру управляющих данных для шага B. Это позволит контроллеру преступить 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 3’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 DM» путем установки cycle_ctrl в 3’b001.

Режим работы с памятью «исполнение с изменением конфигурации»

В данном режиме контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные. Затем контроллер выполняет еще 4 передачи DMA, используя первичные управляющие данные. Контроллер продолжает выполнять циклы ПДА, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

– процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;

– контроллер считает «неправильную» структуру управляющих данных.

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

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

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных. В таблице 196 приведены области памяти channel_cfg, как те, которые должны быть определены константами, так и те, значения которых определяются пользователем.

Таблица 196 – Channel_cfg для первичной структуры управляющих данных в режиме работы с памятью «исполнение с изменением конфигурации»

Номер бита	Обозначение	Значение	Описание
Области с константными значениями			
31…30	dst_inc	b’10	Контроллер производит инкремент адреса пословно
29…28	dst_size	b’10	Контроллер осуществляет передачу пословно
27…26	src_inc	b’10	Контроллер производит инкремент адреса пословно
25…24	src_size	b’10	Контроллер осуществляет передачу пословно
17…14	R_power	b’0010	Контроллер выполняет четыре передачи DMA
3	next_useburst	b’0	Для данного режима этот разряд должен быть равен 0
2…0	cycle_ctrl	b’100	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 R_power задает значение 4, то необходимо задавать значение N, кратное 4. Число, равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные			

Рисунок 67 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=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

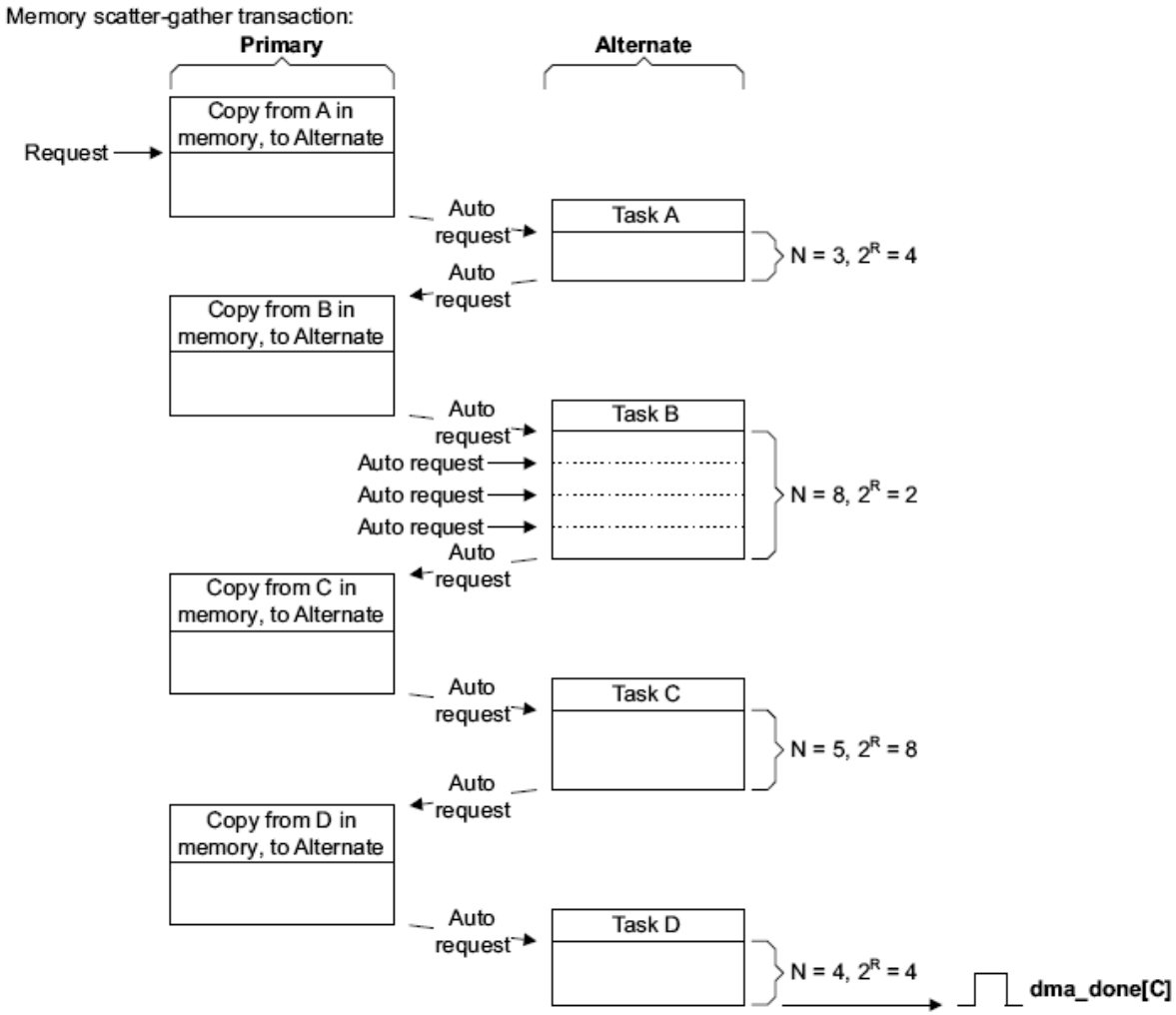


Рисунок 67 – Пример работы DMA в режиме
с «Исполнением с изменением конфигурации»

Пояснения к схеме на рисунке 67:

Инициализация:

1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с памятью «исполнение с изменением конфигурации» путем установки cycle_ctrl в b100. Так как управляющие данные канала состоят из четырех слов, необходимо установить 2^R в 4. В этом примере количество задач равно четырем и поэтому N установлен в 16.

2 Процессор записывает управляющие данные для 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 получении 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 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.

Режим работы с периферией «исполнение с изменением конфигурации»

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

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

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

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

– процессор переведет контроллер в режим «Основной» во время цикла с альтернативной структурой;

– контроллер считает «неправильную» структуру управляющих данных.

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

Контроллер устанавливает флаг dma_done[C] в этом режиме работы только тогда, когда передача DMA заканчивается с использованием основного цикла. В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных.

В таблице 197 приведены области памяти channel_cfg, как те, которые должны быть определены константами, так и те, значения которых определяются пользователем.

Таблица 197 – Channel_cfg для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации»

Номер бита	Обозначение	Значение	Описание
Области с константными значениями			
31…30	dst_inc	b’10	Контроллер производит инкремент адреса пословно
29…28	dst_size	b’10	Контроллер осуществляет передачу пословно
27…26	src_inc	b’10	Контроллер производит инкремент адреса пословно
25…24	src_size	b’10	Контроллер осуществляет передачу пословно
17…14	R_power	b’0010	Контроллер выполняет четыре передачи DMA
2…0	cycle_ctrl	b’110	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 R_power задает значение 4, то необходимо задавать значение N, кратное 4. Число, равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные			

На рисунке 68 приведен 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=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

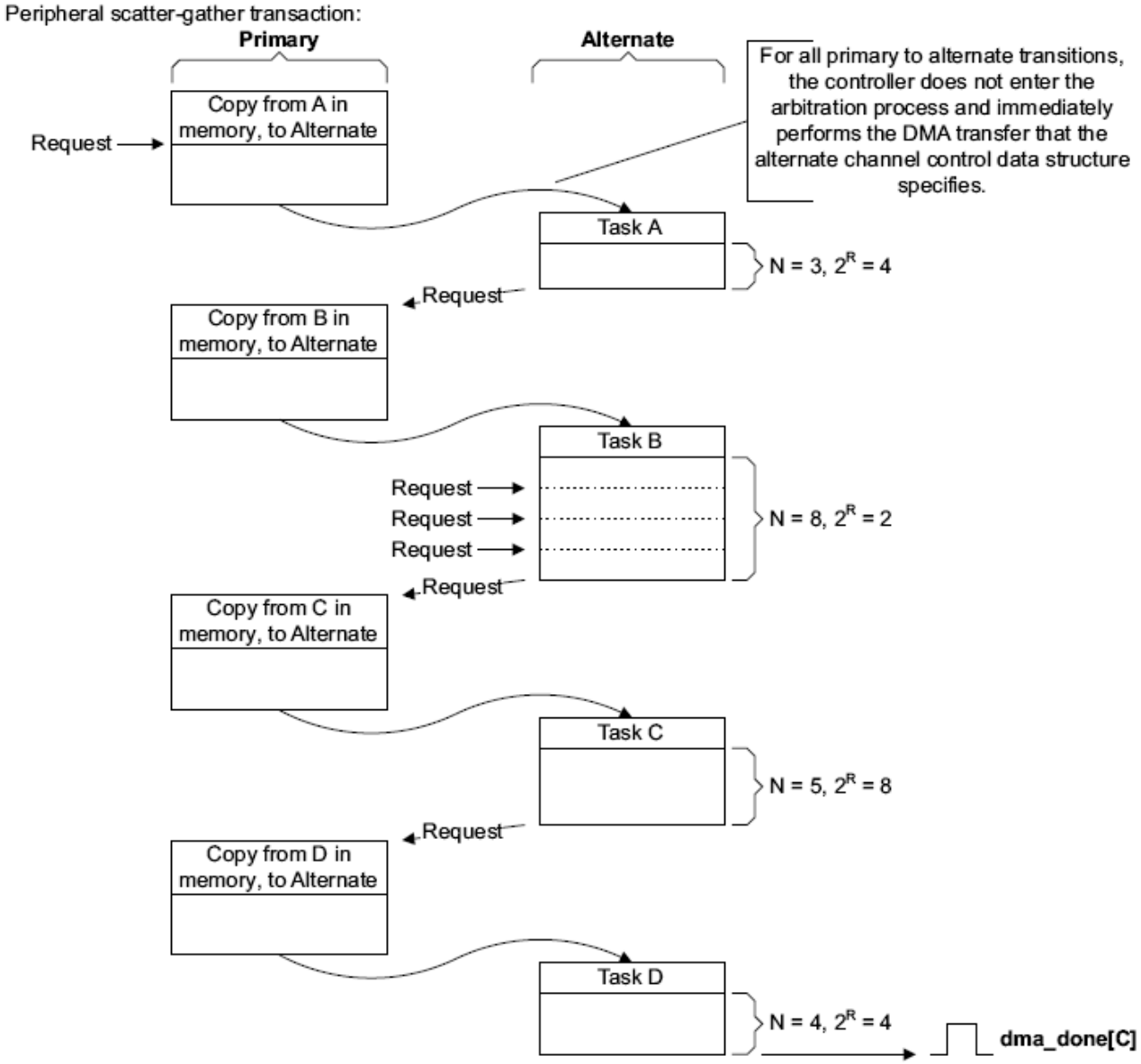


Рисунок 68 – Пример работы DMA в режиме
с «Исполнением с изменением конфигурации»

Пояснения к схеме на рисунке 68:

Инициализация:

1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации» путем установки cycle_ctrl в b110. Так как управляющие данные канала состоят из четырех слов, необходимо установить 2^R в 4. В этом примере количество задач равно 4 и поэтому N установлен в 16.

2 Процессор записывает управляющие данные для 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 получении запроса на обслуживание, контроллер выполняет четыре 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 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.

7.12.4.10 **Индикация ошибок**

При получении контроллером по шине AHB ответа об ошибке контроллер 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.

7.12.5 Структура управляющих данных канала

В системной памяти должна быть отведена область для хранения управляющих данных каналов. Системная память должна:

– предоставлять смежную область системной памяти, к которой контроллер и процессор имеют доступ;

– иметь базовый адрес, который целочисленно кратен общему размеру структуры управляющих данных канала.

На рисунке 69 приведена область памяти, необходимая контроллеру для структур управляющих данных канала, при использовании всех 32 каналов и опциональной альтернативной структуры управляющих данных.

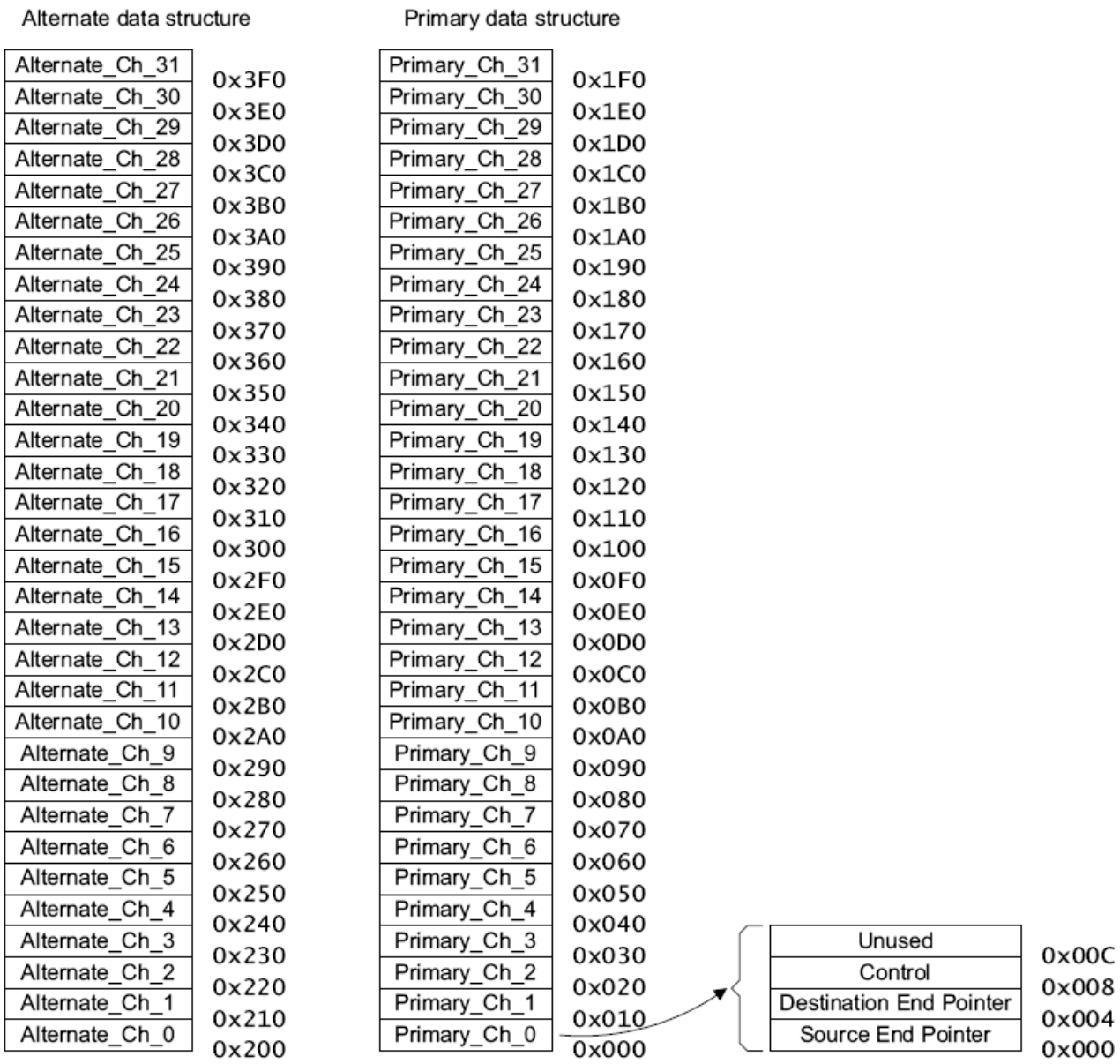


Рисунок 69 – Карта памяти для 32-х каналов, включая альтернативную структуру

Карта памяти, приведенная на рисунке 69, использует 1 Кбайт системной памяти. Контроллер использует младшие 10 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть выровнен по границе 1 Кбайт – 0xXXXXX000, далее 0xXXXXX400, далее 0xXXXXX800, далее 0xXXXXXC00.

Существует возможность установить базовый адрес для первичной структуры управляющих данных путем записи соответствующего значения в регистр ctrl_base_ptr.

Необходимый размер области системной памяти зависит:

– от количества каналов, используемых в контроллере;
– от того, используется или нет альтернативная структура управляющих данных.

В таблице 198 приведены разряды адреса, обеспечивающие контроллеру доступ к различным элементам структуры управляющих данных.

Таблица 198 – Разряды адреса, соответствующие элементам структуры управляющих данных

	Разряды адреса						
Количество каналов, используемых в контроллере	[9]	[8]	[7]	[6]	[5]	[4]	[3:0]
32	A	C[4]	C[3]	C[2]	C[1]	C[0]	0x0 0x4 0x8

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 – Базовый адрес альтернативной структуры управляющих данных вычислять не обязательно, так как регистр alt_ctrl_base_ptr содержит эту информацию.

В таблице 199 приведены все разрешенные значения базового адреса для первичной структуры управляющих данных в зависимости от количества каналов DMA, использованных в контроллере.

Таблица 199 – Разрешенные базовые адреса

Количество каналов DMA	Разрешенные значения базового адреса для первичной структуры управляющих данных
32	0xXXXXX000, 0xXXXXX400, 0xXXXXX800, 0xXXXXXC00

Контроллер использует системную память для доступа к двум указателям адреса конца данных и разрядам управления каждого канала. Эти 32-разрядные области памяти и процедуру вычисления контроллером адреса передачи DMA описывают следующие подразделы:

- указатель конца данных источника;
- указатель конца данных приемника;
- разряды управления;
- вычисление адреса.

7.12.5.1 **Указатель конца данных источника**

Область памяти src_data_end_ptr содержит указатель на последний адрес месторасположения данных источника.

Таблица 200 – Значения разрядов src_data_end_ptr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	src_data_end_ptr	Указатель последнего адреса 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.

7.12.5.2 **Указатель конца данных приемника**

Область памяти dst_data_end_ptr содержит указатель на последний адрес месторасположения данных приемника.

Таблица 201 – Значения разрядов dst_data_end_ptr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	dst_data_end_ptr	Указатель на последний адрес данных приемника

Перед тем, как контроллер выполнит передачу DMA, необходимо определить эту область памяти. Контроллер считывает значение этой области перед началом 2^R передачи DMA.

Примечание – Контроллер не имеет доступа по записи в эту область памяти.

7.12.5.3 **Разряды управления**

Область памяти channel_cfg обеспечивает управление каждой передачей DMA.

Таблица 202 – Значения разрядов dst_data_end_ptr

Номер	31	30	29	28	27	26	25	24	23...21	20...18	17...14	13...4	3	2...0
Доступ														
Сброс														
	dst_inc[1:0]		dst_size[1:0]		src_inc[1:0]		src__size[1:0]		dst_prot_crtl[2:0]	Src_prot_ctrl[2:0]	R_power[3:0]	n_minus_1[9:0]	next_useburst	cycle_ctrl[2:0]

В таблице 203 приведены назначения разрядов этой области памяти.

Таблица 203 – Назначение разрядов channel_cfg

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31, 30	dst_inc[1:0]	Шаг инкремента адреса приемника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: b00 = байт; b01 = полуслово (16 разрядов); b10 = слово (32 разряда); b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr. Разрядность данных источника = полуслово: b00 = зарезервировано; b01 = полуслово; b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr. Разрядность данных источника = слово: b00 = зарезервировано; b01 = зарезервировано; b10 = слово (32 разряда); b11 = нет инкремента. Адрес остается равным значению области памяти dst_data_end_ptr
29, 28	dst_size[1:0]	Размерность данных приемника Примечание – Значение этого поля должно быть равно значению поля src_size
27, 26	src_inc[1:0]	Шаг инкремента адреса источника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: b00 = байт; b01 = полуслово; b10 = слово (32 разряда); b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr. Разрядность данных источника = полуслово: b00 = зарезервировано b01 = полуслово b10 = слово b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr. Разрядность данных источника = слово: b00 = зарезервировано; b01 = зарезервировано;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти src_data_end_ptr
25, 24	src_size[1:0]	Задает размерность данных источника: b00 = байт; b01 = полуслово; b10 = слово (32 разряда); b11 = зарезервировано
23…21	dst_prot_ctrl[2:0]	Задает состояние HPROT[3:1], когда контроллер записывает данные в приемник. Разряд 23 управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд 22 управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд 21 управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ непривилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный
20…18	src_prot_ctrl[2:0]	Задает состояние HPROT[3:1], когда контроллер считывает данные из источника. Разряд 20 управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд 19 управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд 18 управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ непривилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный
17…14	R_power[3:0]	Задает количество передач DMA до выполнения контроллером процедуры арбитража. Возможные значения: b0000 – арбитраж производится после каждой передачи DMA; b0001 – арбитраж производится после 2 передач DMA; b0010 – арбитраж производится после 4 передач DMA; b0011 – арбитраж производится после 8 передач DMA; b0100 – арбитраж производится после 16 передач DMA; b0101 – арбитраж производится после 32 передач DMA; b0110 – арбитраж производится после 64 передач DMA; b0111 – арбитраж производится после 128 передач DMA; b1000 – арбитраж производится после 256 передач DMA; b1001 – арбитраж производится после 512 передач DMA;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		b1010 ÷ 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[9:0]	Перед выполнением цикла DMA эти разряды указывают общее количество передач DMA, из которых состоит цикл DMA. Необходимо установить эти разряды в значение, соответствующее размеру желаемого цикла DMA. 10-разрядное число плюс 1 задает количество передач DMA. Возможные значения: b0000000000 = 1 передача DMA; b0000000001 = 2 передачи DMA; b0000000010 = 3 передачи DMA; b0000000011 = 4 передачи DMA; b0000000100 = 5 передач DMA; b0000000101 = 6 передач DMA; …. b1111111111 = 1024 передачи DMA. Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач DMA до завершения цикла DMA
3	next_useburst	Контролирует, не установлен ли chnl_useburst_set[C] в состояние 1, если контроллер работает в режиме работы с периферией «Исполнение с изменением конфигурации» и, если контроллер завершает цикл DMA, используя альтернативные управляющие данные. Примечание – Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер устанавливает chnl_useburst_set[C] в значение 0, если количество оставшихся передач DMA меньше, чем 2^R. Установка next_useburst разряда определяет, будет ли контроллер дополнительно переопределять разряд chnl_useburst_set[C]. Если контроллер выполняет цикл DMA в режиме работы с периферией «Исполнение с изменением конфигурации», то после окончания цикла, использующего альтернативные управляющие данные, происходит следующее в зависимости от состояния next_useburst: 0 – контроллер не изменяет значение chnl_useburst_set[C]. Если chnl_useburst_set[C] установлен в 0, то для всех оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер отвечает на запросы по dma_req[] и dma_sreq[], при выполнении циклов DMA он использует альтернативные управляющие данные.

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		1 – контроллер изменяет значение chnl_useburst_set[C] в состояние 1. Поэтому для оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер реагирует только на запросы по dma_req[], при выполнении циклов DMA он использует альтернативные управляющие данные
2…0	cycle_ctrl[2:0]	Режим работы при выполнении цикла DMA: b000 – **Стоп**. Означает, что структура управляющих данных является «неправильной»; b001 – Основной. Контроллер должен получить новый запрос для окончания цикла DMA, перед этим он должен выполнить процедуру арбитража; b010 – Авто-запрос. Контроллер автоматически осуществляет запрос на обработку посоответствующему каналу в течение процедуры арбитража. Это означает, что начального запроса наобработку достаточно для выполнения цикла DMA; b011 – Пинг-понг. Контроллер выполняет цикл DMA использую одну из структур управляющих данных. По окончании выполнения цикла DMA, контроллервыполняет следующий цикл DMA, используя другую структуру. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя процессору перенастраивать неактивную структуру данных. Контроллер продолжает выполнять циклы DMA, до тех пор, пока он не прочитает «неправильную» структуру данных или пока процессор не изменит cycle_ctrl поле в состояние b001 или b 010; 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 b100; 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 b101; 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 b110; 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 b111

В начале цикла DMA или 2^R передачи DMA контроллер считывает значение channel_cfg из системной памяти. После выполнения 2R 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 b000, делая тем самым channel_cfg данные «неправильными». Это позволяет избежать повторения выполненной передачи DMA.

7.12.5.4 **Вычисление адреса**

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

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

src_inc=b00 and dst_inc=b00
– адрес источника = src_data_end_ptr - n_minus_1
– адрес приемника = dst_data_end_ptr - n_minus_1.

src_inc=b01 and dst_inc=b01
– адрес источника = src_data_end_ptr – (n_minus_1<<1)
– адрес приемника = dst_data_end_ptr – (n_minus_1<<1).

src_inc=b01 and dst_inc=b10
– адрес источника = src_data_end_ptr – (n_minus_1<<2)
– адрес приемника = dst_data_end_ptr – (n_minus_1<<2).

src_inc=b11 and dst_inc=b11
– адрес источника = src_data_end_ptr
– адрес приемника = dst_data_end_ptr.

В таблице 204 приведены адреса приемника цикла DMA для шести слов.

Таблица 204 – Цикла DMA для шести слов с пословным инкрементом

Начальные значения channel_cfg перед циклом DMA				
src_size=b10, dst_inc=b10, n_minus_1=b101, cycle_ctrl=1				
DMA передачи	Указатель конца данных	Счетчик	Отличие*)	Адрес
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения channel_cfg после цикла DMA				
src_size=b10, dst_inc=b10, n_minus_1=0, cycle_ctrl=0				
* Значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc				

В таблице 205 приведены адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.

Таблица 205 – Цикла DMA для 12 байт с «полусловным» инкрементом

Начальные значения channel_cfg перед циклом DMA				
src_size=b00, dst_inc=b01, n_minus_1=b1011, cycle_ctrl=1, R_power=b11				
DMA передачи	Указатель конца данных	Счетчик	Отличие*)	Адрес
	0x5E7	11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3
	0x5E7	9	0x12	0x5D5
	0x5E7	8	0x10	0x5D7
	0x5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
	0x5E7	4	0x8	0x5DF
Значения channel_cfg после 2^R передач DMA				
src_size=b00, dst_inc=b01, n_minus_1=b011, cycle_ctrl=1, R_power=b11				
DMA передачи	0x5E7	3	0x6	0x5E1
	0x5E7	2	0x4	0x5E3
	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7
Конечные значения channel_cfg после цикла DMA				
src_size=b00, dst_inc=b01, n_minus_1=0, cycle_ctrl=0**), R_power=b11				
* Значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc. ** После окончания цикла DMA контроллер делает channel_cfg «неправильным», сбрасывая в 0 поле cycle_ctrl				

7.12.6 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 работы с 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 206 – Регистры контроллера DMA

Базовый адрес	Наименование	Тип	Значение по сбросу	Описание
0xE004_2000	MDR_DMA			Контроллер DMA
Смещение				
0x000	STATUS	RO	0x101F0000	Статусный регистр DMA
0x004	CFG	WO	-	Регистр конфигурации DMA
0x008	CTRL_BASE_PTR	R/W	0x00000000	Регистр базового адреса управляющих данных каналов
0x00C	ALT_CTRL_BASE_PTR	RO	0x00000200	Регистр базового адреса альтернативных управляющих данных каналов
0x010	WAITONREQ_STATUS	RO	0xFFFFFFFF	Регистр статуса ожидания запроса на обработку каналов
0x014	CHNL_SW_REQUEST	WO	-	Регистр программного запроса на обработку каналов
0x018	CHNL_USEBURST_SET	R/W	0x00000000	Регистр установки пакетного обмена каналов
0x01C	CHNL_USEBURST_CLR	WO	-	Регистр сброса пакетного обмена каналов

Базовый адрес	Наименование	Тип	Значение по сбросу	Описание
0x020	CHNL_REQ_MASK_SET	R/W	0x00000000	Регистр маскирования запросов на обслуживание каналов
0x024	CHNL_REQ_MASK_CLR	WO	-	Регистр очистки маскирования запросов на обслуживание каналов
0x028	CHNL_ENABLE_SET	R/W	0x00000000	Регистр установки разрешения каналов
0x02C	CHNL_ENABLE_CLR	WO	-	Регистр сброса разрешения каналов
0x030	CHNL_PRI_ALT_SET	R/W	0x00000000	Регистр установки первичной/альтернативной структуры управляющих данных каналов
0x034	CHNL_PRI_ALT_CLR	WO	-	Регистр сброса первичной/альтернативной структуры управляющих данных каналов
0x038	CHNL_PRIORITY_SET	R/W	0x00000000	Регистр установки приоритета каналов
0x03C	CHNL_PRIORITY_CLR	WO	-	Регистр сброса приоритета каналов
0x040-0x048	-		-	Зарезервировано
0x04C	ERR_CLR	R/W	0x00000000	Регистр сброса флага ошибки
0x050	CHMUX0	R/W	0x00000000	Регистр назначения каналов 0,1,2,3
0x054	CHMUX1	R/W	0x00000000	Регистр назначения каналов 4,5,6,7
0x058	CHMUX2	R/W	0x00000000	Регистр назначения каналов 8,9,10,11
0x05C	CHMUX3	R/W	0x00000000	Регистр назначения каналов 12,13,14,15
0x060	CHMUX4	R/W	0x00000000	Регистр назначения каналов 16,17,18,19
0x064	CHMUX5	R/W	0x00000000	Регистр назначения каналов 20,21,22,23
0x068	CHMUX6	R/W	0x00000000	Регистр назначения каналов 24,25,26,27
0x06C	CHMUX7	R/W	0x00000000	Регистр назначения каналов 28,29,30,31
0x070-0xDFC	-	-		Зарезервировано

7.12.6.1 **STATUS**

Регистр STATUS является статусным регистром контроллера DMA.

Данный регистр имеет доступ только на чтение. При чтении регистр возвращает состояние контроллера. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

В таблице 208 приведены назначения разрядов регистра.

Таблица 207 – 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	1	0	11111	0	0	0	0
	test_status	-	chnls_minus1	-	state	-	master_enable

Таблица 208 – 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: 0x0 = контроллер не имеет интегрированной схемы тестирования; 0x1 = контроллер имеет интегрированную схему тестирования; 0x2 – 0xF = не определено
27...21	-	Не определено
20...16	chnls_minus1	Количество доступных каналов DMA минус 1. b11111 = контроллер имеет 32 канала DMA
15...8	-	Не определено
7...4	state	Текущее состояние автомата управления контроллера. Состояние может быть одним из следующих: b0000 = в покое; b0001 = чтение управляющих данных канала; b0010 = чтение указателя конца данных источника; b0011 = чтение указателя конца данных приемника; b0100 = чтение данных источника; b0101 = запись данных в приемник; b0110 = ожидание запроса на выполнение DMA; b0111 = запись управляющих данных канала; b1000 = приостановлен; b1001 = выполнен; b1010 = режим работы с периферией «Исполнение с изменением конфигурации»; b1011-b1111 = не определено
3...1	-	Не определено
0	master_enable	Состояние контроллера: 0 = работа контроллера запрещена;

		1 = работа контроллера разрешена

7.12.6.2 **CFG**

Регистр CFG является регистром конфигурации контроллера DMA.

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера.

В таблице 210 приведены назначения разрядов регистра.

Таблица 209 – 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 210 – Назначение разрядов регистра dma_cfg

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...8	-	Не определено, следует записывать 0.
7...5	chnl_prot_ctrl	Определяет уровни индикации сигналов HPROT[3:1] защиты шины AHB-Lite: Разряд 7 управляет сигналом HPROT[3], с целью индикации о появлении доступа с кэшированием; Разряд 6 управляет сигналом HPROT[2], с целью индикации о появлении доступа с буферизацией; Разряд 5 управляет сигналом HPROT[1], с целью индикации о появлении привилегированного доступа. Примечания 1 Если разряд[n] = 1, то соответствующий сигнал HPROT в состоянии 1; 2 Если разряд[n] = 0, то соответствующий сигнал HPROT в состоянии 0
4...1	-	Не определено. Следует записывать 0.
0	master_enable	Определяет состояние контроллера: 0 – запрещает работу контроллера; 1 – разрешает работу контроллера

7.12.6.3 **CTRL_BASE_PTR**

Регистр CTRL_BASE_PTR является регистром базового адреса управляющих данных каналов.

Данный регистр имеет доступ на запись и чтение. Регистр определяет базовый адрес системной памяти размещения управляющих данных каналов.

Примечание – Контроллер не содержит внутреннюю память для хранения управляющих данных каналов.

Размер системной памяти, предназначенной контроллеру, зависит от количества каналов DMA, использующихся контроллером, а также от возможности использования альтернативных управляющих данных каналов.

Значение, записываемое в регистр, должно быть выровнено по границе 1 Кбайт. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

В таблице 212 приведены назначения разрядов регистра ctrl_base_ptr.

Таблица 211 – Регистр базового адреса управляющих данных каналов

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

Таблица 212 – 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

7.12.6.4 **ALT_CTRL_BASE_PTR**

Регистр ALT_CTRL_BASE_PTR является регистром базового адреса альтернативных управляющих данных каналов.

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении указатель базового адреса альтернативных управляющих данных каналов. Если контроллер находится в состоянии сброса, то чтение регистра запрещено. Этот регистр позволяет не производить вычисления базового адреса альтернативных управляющих данных каналов.

В таблице 214 приведены назначения разрядов регистра.

Таблица 213 – Регистр базового адреса альтернативных управляющих данных каналов

Номер	31... 0
Доступ	RO
Сброс	0x00000200
	Alt_ctrl_base_ptr

Таблица 214 – 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

7.12.6.5 **WAITONREQ_STATUS**

Регистр WAITONREQ_STATUS является регистром статуса ожидания запроса на обработку каналов.

Данный регистр имеет доступ только на чтение. Регистр возвращает при чтении состояние сигналов dma_waitonreq[]. Если контроллер находится в состоянии сброса, то чтение регистра запрещено.

В таблице 216 приведены назначения разрядов регистра.

Таблица 215 – Регистр статуса ожидания запроса на обработку каналов

Номер	31	.....	2	1	0
Номер	RO	.....	RO	RO	RO
Доступ	1	.....	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 216 – Назначение разрядов регистра dma_waitonreq_status

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	dma_waitonreq_status	Состояние сигналов ожидания запроса на обработку каналов DMA. При чтении: Разряд [C] =0 означает, что dma_waitonreq[C] в состоянии 0; Разряд [C] =1 означает, что dma_waitonreq[C] в состоянии 1

7.12.6.6 **CHNL_SW_REQUEST**

Регистр CHNL_SW_REQUEST является регистром программного запроса на обработку каналов.

Данный регистр имеет доступ только на запись. Регистр позволяет устанавливать программно запрос на выполнение цикла DMA.

В таблице 218 приведены назначения разрядов регистра.

Таблица 217 – 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 218 – Назначение разрядов регистра chnl_sw_request

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_sw_request	Устанавливает соответствующий разряд для генерации программного запроса на выполнение цикла DMA по соответствующему каналу DMA. При записи: Разряд [C] = 0 означает, что запрос на выполнение цикла DMA по каналу C не будет установлен; Разряд [C] =1 означает, что запрос на выполнение цикла DMA по каналу C будет установлен. Запись разряда, соответствующего нереализованному каналу, означает, что запрос на выполнение цикла DMA не будет установлен

7.12.6.7 **CHNL_USEBURST_SET**

Регистр CHNL_USEBURST_SET является регистром установки пакетного обмена каналов.

Данный регистр имеет доступ на чтение и запись. Регистр отключает выполнение одиночных запросов по установке dma_sreq[] и поэтому будут обрабатываться и исполняться только запросы по dma_req[]. Регистр возвращает при чтении состояние установок пакетного обмена

В таблице 220 приведены назначения разрядов регистра.

Таблица 219 – 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 220 – Назначение разрядов регистра chnl_useburst_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_useburst_set	Отключает обработку запросов на выполнение циклов DMA от dma_sreq[] и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные от dma_sreq[] и dma_req[]. Контроллер выполняет одиночные передачи или 2^R передач.

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		Разряд [C] = 1 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные только от dma_req[]. Контроллер выполняет 2^R передач. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_clr регистр и установить соответствующий разряд C в 0; Разряд [C] = 1 отключает возможность обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Контроллер выполняет 2^R передач. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

После выполнения предпоследней передачи из 2^R передач, в том случае, если число оставшихся передач (N) меньше чем 2^R, контроллер сбрасывает разряд chnl_useburst_set в 0. Это позволяет выполнять оставшиеся передачи, используя dma_sreq[] и dma_req[].

Примечание – При программировании channel_cfg значением N меньшим, чем 2^R, запрещена установка соответствующего разряда chnl_useburst_set в случае, если периферийный блок не поддерживает сигнал dma_req[].

В режиме работы с периферией «исполнение с изменением конфигурации», если разряд next_useburst установлен в channel_cfg, то контроллер устанавливает chnl_useburst_set [C] в 1 после окончания цикла DMA, использующего альтернативные управляющие данные.

7.12.6.8 **CHNL_USEBURST_CLR**

Регистр CHNL_USEBURST_CLR является регистром сброса пакетного обмена каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает выполнение одиночных запросов по установке dma_sreq[].

В таблице 222 приведены назначения разрядов регистра chnl_useburst_clr.

Таблица 221 – 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 222 – Назначение разрядов регистра chnl_useburst_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_useburst_clr	Установка соответствующего разряда разрешает обработку запросов на выполнение циклов DMA от dma_sreq[]. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_useburst_set регистр для отключения обработки запросов от dma_sreq[]; Разряд [C] = 1 разрешает обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.12.6.9 **CHNL_REQ_MASK_SET**

Регистр CHNL_REQ_MASK_SET является регистром маскирования запросов на обслуживание каналов

Данный регистр имеет доступ на чтение и запись. Регистр отключает установку запросов на выполнение циклов DMA на dma_sreq[] и dma_req[]. Регистр возвращает при чтении состояние установок маскирования запросов от dma_sreq[] и dma_req[] на обслуживание каналов.

В таблице 224 приведены назначения разрядов регистра chnl_req_mask_set.

Таблица 223 – Регистр маскирования запросов на обслуживание каналов

Номер	31	.....	2	1	0
Доступ	R/W	.....	R/W	R/W	R/W
Сброс	0	.....	0	0	0
	chnl_reg_mask_set for dma_reg [31] and dma_sreg [31]	.....	chnl_reg_mask_set for dma_reg [2] and dma_sreg [2]	chnl_reg_mask_set for dma_reg [1] and dma_sreg [1]	chnl_reg_mask_set for dma_reg [0] and dma_sreg [0]

Таблица 224 – Назначение разрядов регистра chnl_req_mask_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_req_mask_set	Отключает обработку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C выполняет циклы DMA в ответ на поступающие запросы; Разряд [C] = 1 означает, что канал DMA C не выполняет циклы DMA в ответ на поступающие запросы.

		При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_req_mask_clr регистр для разрешения установки запросов; Разряд [C] = 1 отключает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.12.6.10 **CHNL_REQ_MASK_CLR**

Регистр CHNL_REQ_MASK_CLR является регистром очистки маскирования запросов на обслуживание каналов.

Данный регистр имеет доступ только на запись. Регистр разрешает установку запросов на выполнение циклов DMA на dma_sreq[] и dma_req[].

В таблице 226 приведены назначения разрядов регистра chnl_req_mask_clr.

Таблица 225 – Регистр очистки маскирования запросов на обслуживание каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
Сброс	0	.....	0	0	0
	chnl_reg_mask_clr for dma_reg [31] and dma_sreg [31]	.....	chnl_reg_mask_clr for dma_reg [2] and dma_sreg [2]	chnl_reg_mask_clr for dma_reg [1] and dma_sreg [1]	chnl_reg_mask_clr for dma_reg [0] and dma_sreg [0]

Таблица 226 – Назначение разрядов регистра chnl_req_mask_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_req_mask_clr	Установка соответствующего разряда разрешает установку запросов по dma_sreq[] и dma_req[] на выполнение циклов DMA от каналов. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_req_mask_set регистр для отключения установки запросов; Разряд [C] =1 разрешает установку запросов на выполнение циклов DMA, по dma_sreq[] и dma_req[]. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.12.6.11 **CHNL_ENABLE_SET**

Регистр CHNL_ENABLE_SET является регистром установки разрешения каналов.

Данный регистр имеет доступ на чтение и запись. Регистр разрешает работу каналов DMA. Регистр возвращает при чтении состояние разрешений работы каналов DMA.

В таблице 228 приведены назначения разрядов регистра chnl_enable_set.

Таблица 227 – Регистр установки разрешения каналов

Номер	31	.....	2	1	0
Доступ	WO	.....	WO	WO	WO
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 228 – Назначение разрядов регистра chnl_enable_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_enable_set	Разрешает работу каналов DMA и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C отключен; Разряд [C] = 1 означает, что работа канала DMA C разрешена. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_enable_clr регистр для отключения канала; Разряд [C] = 1 разрешает работу канала DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.12.6.12 **CHNL_ENABLE_CLR**

Регистр CHNL_ENABLE_CLR является регистром сброса разрешения каналов.

Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов DMA.

В таблице 230 приведены назначения разрядов регистра chnl_enable_clr.

Таблица 229 – 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 230 – Назначение разрядов регистра chnl_enable_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_enable_clr	Установка соответствующего разряда запрещает работу соответствующего канала DMA. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_enable_set регистр для разрешения работы канала; Разряд [C] = 1 запрещает работу канала DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. Примечание – Контроллер может отключить канал DMA, установив соответствующий разряд в следующих случаях: – при завершении цикла DMA; – при чтении из channel_cfg с полем cycle_ctrl установленным в b000; – при появлении ошибки на шине AHB-Lite

7.12.6.13 **CHNL_PRI_ALT_SET**

Регистр CHNL_PRI_ALT_SET является регистром установки первичной/альтернативной структуры управляющих данных каналов

Данный регистр имеет доступ на запись и чтение. Регистр разрешает работу канала DMA с использованием альтернативной структуры управляющих данных. Чтение регистра возвращает состояние каналов DMA (какую структуру управляющих данных использует каждый канал DMA).

В таблице 232 приведены назначение разрядов регистра chnl_pri_alt_set.

Таблица 231 – 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 232 – Назначение разрядов регистра chnl_pri_alt_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_pri_alt_set	Установка соответствующего разряда подключает использование альтернативных управляющих данных для соответствующего канала DMA, чтение возвращает состояние этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C использует первичную структуру управляющих данных; Разряд [C] = 1 означает, что канал DMA C использует альтернативную структуру управляющих данных. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_clr регистр для сброса разряда [C] в 0; Разряд [C] = 1 подключает использование альтернативной структуры управляющих данных каналом DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. Примечание – Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях: – при завершении четырех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; – при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «Пинг-понг»; – при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: – «пинг-понг»; – работа с памятью «Исполнение с изменением конфигурации»; – работа с периферией «Исполнение с изменением конфигурации

7.12.6.14 **CHNL_PRI_ALT_CLR**

Регистр CHNL_PRI_ALT_CLR является регистром сброса первичной/альтернативной структуры управляющих данных каналов

Данный регистр имеет доступ только на запись. Регистр разрешает работу канала DMA с использованием первичной структуры управляющих данных.

В таблице 234 приведены назначения разрядов регистра chnl_pri_alt_clr.

Таблица 233 – 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 234 – Назначение разрядов регистра chnl_pri_alt_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_pri_alt_clr	Установка соответствующего разряда подключает использование первичных управляющих данных для соответствующего канала DMA. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_set регистр для выбора альтернативных управляющих данных; Разряд [C] = 1 подключает использование первичной структуры управляющих данных каналом DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. Примечание – Контроллер может переключить значение разряда chnl_pri_alt_clr[C] в следующих случаях: – при завершении четырех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; – при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «пинг-понг»; – при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: – «пинг-понг»; – работа с памятью «Исполнение с изменением конфигурации»; – работа с периферией «Исполнение с изменением конфигурации»

7.12.6.15 **CHNL_PRIORITY_SET**

Регистр CHNL_PRIORITY_SET является регистром установки приоритета каналов.

Данный регистр имеет доступ на запись и чтение. Регистр позволяет присвоить высокий приоритет каналу DMA. Чтение регистра возвращает состояние приоритета каналов DMA.

В таблице 236 приведены назначения разрядов регистра chnl_priority_set.

Таблица 235 – 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 236 – Назначение разрядов регистра chnl_priority_set

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	chnl_priority_set	Установка высокого приоритета каналу DMA, чтение возвращает состояние приоритета каналов DMA. При чтении: Разряд [C] = 0 означает, что каналу DMA с присвоен уровень приоритета по умолчанию; Разряд [C] = 1 означает, что каналу DMA C присвоен высокий уровень приоритета. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_clr регистр для установки каналу C уровня приоритета по умолчанию; Разряд [C] = 1 устанавливает каналу DMA C высокий уровень приоритета. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

7.12.6.16 **CHNL_PRIORITY_CLR**

Регистр CHNL_PRIORITY_CLR является регистром сброса приоритета каналов.

Данный регистр имеет доступ только на запись. Регистр позволяет присвоить каналу DMA уровень приоритета по умолчанию.

В таблице 238 приведены назначения разрядов регистра chnl_priority_clr.

Таблица 237 – 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 238 – Назначение разрядов регистра chnl_priority_clr

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

7.12.6.17 **ERR_CLR**

Регистр ERR_CLR является регистром сброса флага ошибки.

Данный регистр имеет доступ на запись и чтение. Регистр позволяет сбрасывать сигнал dma_err в 0. Чтение регистра возвращает состояние сигнала dma_err.

В таблице 240 приведены назначение разрядов регистра err_clr.

Таблица 239 – Регистр сброса флага ошибки

Номер	31…1	0
Доступ	U	R/W
Сброс	0	0
	-	**err_clr**

Таблица 240 – Назначение разрядов регистра err_clr

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…1	-	Не определено. Следует записывать 0

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

7.12.6.18 **CHMUX[x]**

Base ADDR=		STATUS			Offset=		0x050 0x054 0x058 0x05C 0x060 0x064 0x068 0x06C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHANNEL[x•4+3]								CHANNEL[x•4+2]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANNEL[x•4+1]								CHANNEL[x•4+0]							

Таблица 241 – Описание бит регистров CHMUX[x]

Бит	Имя	Значение	Описание
31…24	CHANNEL[x•4+3]	0	Номер виртуального канала запроса DMA, назначенного на реальный канал x•4+3
23…16	CHANNEL[x•4+2]	0	Номер виртуального канала запроса DMA, назначенного на реальный канал x•4+2
15…8	CHANNEL[x•4+1]	0	Номер виртуального канала запроса DMA, назначенного на реальный канал x•4+1
7…0	CHANNEL[x•4+0]	0	Номер виртуального канала запроса DMA, назначенного на реальный канал x•4+0

Таблица 242 – Виртуальные номера запросов DMA

Номера виртуальных каналов DMA	Тип SREQ	Тип REQ	Описание
0	ETH0_EVENT[0]	-	Запрос от ETH0
1	ETH0_EVENT[1]	-	Запрос от ETH0
2	-	-	Зарезервировано
3	-	-	Зарезервировано
4	SPW0_RX_SREQ	SPW0_RX_BREQ	Запрос приемника SPW0
5	-	-	Зарезервировано
6	SPW0_TX_SREQ	SPW0_TX_BREQ	Запрос передатчика SPW0
7	-	-	Зарезервировано
8	TMR0_REQ	-	Запрос REQ от Таймера 0
9	TMR1_REQ	-	Запрос REQ от Таймера 1
10	TMR2_REQ	-	Запрос REQ от Таймера 2
11	TMR3_REQ	-	Запрос REQ от Таймера 3
12	-	-	Зарезервировано
13	-	-	Зарезервировано
14	TMR0_REQ1	-	Запрос REQ1 от Таймера 0
15	TMR1_REQ1	-	Запрос REQ1 от Таймера 1
16	TMR2_REQ1	-	Запрос REQ1 от Таймера 2
17	TMR3_REQ1	-	Запрос REQ1 от Таймера 3
18	-	-	Зарезервировано
19	-	-	Зарезервировано
20	TMR0_REQ2	-	Запрос REQ2 от Таймера 0
21	TMR1_REQ2	-	Запрос REQ2 от Таймера 1
22	TMR2_REQ2	-	Запрос REQ2 от Таймера 2
23	TMR3_REQ2	-	Запрос REQ2 от Таймера 3
24	-	-	Зарезервировано
25	-	-	Зарезервировано
26	TMR0_REQ3	-	Запрос REQ3 от Таймера 0
27	TMR1_REQ3	-	Запрос REQ3 от Таймера 1
28	TMR2_REQ3	-	Запрос REQ3 от Таймера 2
29	TMR3_REQ3	-	Запрос REQ3 от Таймера 3
30	-	-	Зарезервировано
31	-	-	Зарезервировано
32	TMR0_REQ4	-	Запрос REQ4 от Таймера 0
33	TMR1_REQ4	-	Запрос REQ4 от Таймера 1
34	TMR2_REQ4	-	Запрос REQ4 от Таймера 2
35	TMR3_REQ4	-	Запрос REQ4 от Таймера 3
36	-	-	Зарезервировано
37	-	-	Зарезервировано
38	SSP0_TX_SREQ	SSP0_TX_BREQ	Запрос передатчика SSP0
39	-	-	Зарезервировано

Номера виртуальных каналов DMA	Тип SREQ	Тип REQ	Описание
40	-	-	Зарезервировано
41	-	-	Зарезервировано
42	SSP0_RX_SREQ	SSP0_RX_BREQ	Запрос приемника SSP0
43	-	-	Зарезервировано
44	-	-	Зарезервировано
45	-	-	Зарезервировано
46	UART0_TX_SREQ	UART0_TX_BREQ	Запрос передатчика UART0
47	UART1_TX_SREQ	UART1_TX_BREQ	Запрос передатчика UART1
48	-	-	Зарезервировано
49	-	-	Зарезервировано
50	UART0_RX_SREQ	UART0_RX_BREQ	Запрос приемника UART0
51	UART1_RX_SREQ	UART1_RX_BREQ	Запрос приемника UART1
52	-	-	Зарезервировано
53	-	-	Зарезервировано
54	ADC0_SREQ0	-	Запрос REQ0 от ADC0
55	ADC0_SREQ1	-	Запрос REQ1от ADC0
56	ADC0_SREQ2	-	Запрос REQ2от ADC0
57	ADC0_SREQ3	-	Запрос REQ3от ADC0
58	ADC0_SREQ4	-	Запрос REQ4от ADC0
59	ADC0_SREQ5	ADC0_BREQ5	Запрос REQ5от ADC0
60	ADC1_SREQ0	-	Запрос REQ0 от ADC1
61	ADC1_SREQ1	-	Запрос REQ1от ADC1
62	ADC1_SREQ2	-	Запрос REQ2от ADC1
63	ADC1_SREQ3	-	Запрос REQ3от ADC1
64	ADC1_SREQ4	-	Запрос REQ4от ADC1
65	ADC1_SREQ5	ADC1_BREQ5	Запрос REQ5от ADC1
66	DAC0_SREQ	DAC0_BREQ	Запрос REQ от DAC0
67	DAC1_SREQ	DAC1_BREQ	Запрос REQ от DAC1
68	-	-	Зарезервировано
69	-	-	Зарезервировано
70	-	-	Зарезервировано
71	-	-	Зарезервировано
72	-	-	Зарезервировано
73	-	-	Зарезервировано
74	-	-	Зарезервировано
75	-	-	Зарезервировано
76	-	-	Зарезервировано
77	-	-	Зарезервировано
78	-	-	Зарезервировано
79	-	-	Зарезервировано
80	-	-	Зарезервировано

Номера виртуальных каналов DMA	Тип SREQ	Тип REQ	Описание
81	-	-	Зарезервировано
82	-	-	Зарезервировано
83	-	-	Зарезервировано
84	-	-	Зарезервировано
85	-	-	Зарезервировано
86	-	-	Зарезервировано
87	-	-	Зарезервировано
88	-	-	Зарезервировано
89	-	-	Зарезервировано
90	-	-	Зарезервировано
91	-	-	Зарезервировано
92	-	-	Зарезервировано
93	-	-	Зарезервировано
94	-	-	Зарезервировано
95	-	-	Зарезервировано

7.13 Контроллер портов ввода-вывода (PORTx)

Порты ввода-вывода предназначены для приема и формирования внешних электрических сигналов. Для каждого вывода микросхемы может быть назначена одна из 16 функций. В качестве функции вывода выступает использование данного вывода как ввода-вывода сигналов периферийных контроллеров, таких как внешняя шина, UART, SSP и так далее, либо полностью программное управление – пользовательский порт. Для некоторых выводов может быть назначена аналоговая функция, например, канал ввода аналогового сигнала в АЦП или выдачи сигнала с ЦАП.

Порты ввода-вывода могут использоваться пользовательскими задачами без дополнительного программного драйвера. Для того, чтобы избежать взаимовлияния работы различных задач друг на друга, например, при возникновении перехода с одной задачи на другую в момент выполнения операции чтения-модификации-записи, все регистры порта имеют виртуальный регистр установки битов в 1 и виртуальный регистр сброса битов в 0. Таким образом, чтобы установить какой-либо бит регистра в 1 нет необходимости выполнять операции чтения и модификации, достаточно записать 1 в соответствующий бит регистра с префиксом «S», и это взведет соответствующий бит. Для сброса этого бита необходимо записать 1 в регистр с префиксом «C». Запись 0 в биты никакого эффекта не несет. При чтении из всех регистров, кроме RXTX, возвращается состояние триггеров. При чтении регистра RXTX возвращается значение с выводов микросхемы.

Каждый вывод порта может служить источником прерываний по уровню от портов. Время удержания активного уровня на входе порта, для формирования запроса на прерывание, должно быть не менее двух тактов ядра. Схема формирования запроса прерывания представлена на рисунке 70.

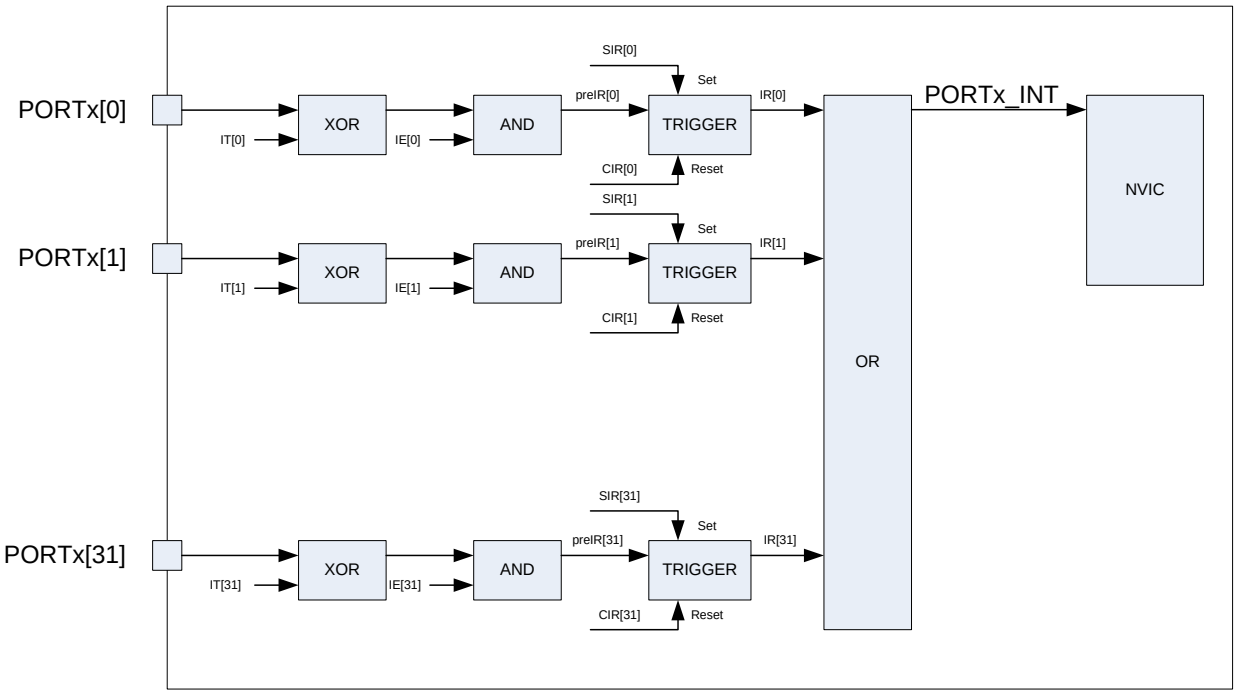


Рисунок 70 – Схема формирования запроса прерывания

В регистре IR запроса прерывания порта факт прерывания фиксируется только при разрешении запроса. При этом запрещение запроса прерывания IE не сбрасывает флаг IR, если он появился до запрета. Сброс запроса прерывания возможен только сбросом соответствующего разряда регистра IR. Формирование прерываний, их обработка и сброс возможны только при включенном тактировании порта.

Для портов ввода-вывода реализована защита от «перегрузки». Если вывод выдает логическую «1», но на линии уровень сигнала соответствует логическому “0”, вырабатывается флаг защиты от перегрузки HCUR. При выдаче логического “0” механизм защиты аналогичен. При возникновении флага защиты может быть сгенерировано прерывание от порта. Для исправления ситуации необходимо программно перевести вывод в третье состояние или, в случае, когда перегрузки кратковременны, сбросить флаг записью 1 в соответствующий бит регистра HCUR.

7.13.1 Описание регистров

Таблица 243 – Описание регистров контроллера портов ввода-вывода

Базовый адрес	Обозначение	Состояние после сброса	Описание
0x4008_0000	MDR_PORTA		Контроллер порта A
0x4008_1000	MDR_PORTB		Контроллер порта B
0x4008_2000	MDR_PORTC		Контроллер порта C
0x4008_3000	MDR_PORTD		Контроллер порта D
0x4008_4000	MDR_PORTE		Контроллер порта E
Смещение			
0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	RXTX		Регистр данных порта
0x0000_0008	SRXTX		Регистр установки порта
0x0000_000C	CRXTX		Регистр сброса порта
0x0000_0010	SOE		Регистр установки направления вывода порта
0x0000_0014	COE		Регистр сброса направления вывода порта
0x0000_0018	SFUNC0		Регистр установки функции вывода порта (7-0)
0x0000_001C	SFUNC1		Регистр установки функции вывода порта (15-8)
0x0000_0020	SFUNC2		Регистр установки функции вывода порта (23-16)
0x0000_0024	SFUNC3		Регистр установки функции вывода порта (31-24)
0x0000_0028	CFUNC0		Регистр сброса функции вывода порта (7-0)
0x0000_002C	CFUNC1		Регистр сброса функции вывода порта (15-8)
0x0000_0030	CFUNC2		Регистр сброса функции вывода порта (23-16)
0x0000_0034	CFUNC3		Регистр сброса функции вывода порта (31-24)
0x0000_0038	SANALOG		Регистр установки типа вывода порта
0x0000_003C	CANALOG		Регистр сброса типа вывода порта
0x0000_0040	SPULLUP		Регистр установки подтяжки к питанию вывода порта
0x0000_0044	CPULLUP		Регистр сброса подтяжки к питанию вывода порта

Базовый адрес	Обозначение	Состояние после сброса	Описание
0x0000_0048	SPULLDOWN		Регистр установки подтяжки к земле вывода порта
0x0000_004C	CPULLDOWN		Регистр сброса подтяжки к земле вывода порта
0x0000_0050	SPD		Регистр установки типа драйвера вывода порта
0x0000_0054	CPD		Регистр сброса типа драйвера вывода порта
0x0000_0058	SPWR0		Регистр установки скорости вывода порта (15-0)
0x0000_005C	SPWR1		Регистр установки скорости вывода порта (31-16)
0x0000_0060	CPWR0		Регистр сброса скорости вывода порта (15-0)
0x0000_0064	CPWR1		Регистр сброса скорости вывода порта (31-16)
0x0000_0068	SCL		Регистр установки ограничения по току вывода порта
0x0000_006C	CCL		Регистр сброса ограничения по току вывода порта
0x0000_0070	SIE		Регистр установки разрешения прерывания от вывода
0x0000_0074	CIE		Регистр сброса разрешения прерывания от вывода
0x0000_0078	SIT		Регистр установки активного уровня прерывания
0x0000_007C	CIT		Регистр сброса активного уровня прерывания
0x0000_0080	SIR		Регистр установки флага прерывания
0x0000_0084	CIR		Регистр сброса флага прерывания
0x0000_0088	HCUR		Регистр флага превышения выходного тока

7.13.1.1 **KEY**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0000								
REG Name:		KEY													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 244 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока

7.13.1.2 **RXTX**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXTX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXTX[15:0]															

Таблица 245 – Описание бит регистра RXTX

Бит	Имя	Значение	Описание
31…0	RXTX	0x00000000	Регистр прямого доступа данных порта. При чтении отображает состояние на выводах порта. При записи осуществляет запись в выходные триггера порта: RXTX[0] соответствует выводу порта с номером 0; RXTX[1] соответствует выводу порта с номером 1 и так далее. **Внимание!** При выполнении операции чтение-модификация-запись возможно возникновения ошибок, например: – вывод порта настроен на выдачу высокого уровня (в выходном триггере 1), но при этом нагрузка на выводе такая, что уровень стал ниже нижней границы высокого уровня, в этом случае при чтении регистра RXTX может бить считан 0. И при его последующей записи измениться значение в выходном триггере с 1 на 0. – при выполнении операции чтение-модификации-записи произошло переключение задачи, либо уход на обработку прерывания сразу после выполнения чтения. В другой задаче или обработчике прерывания изменяется значение в выходных триггерах и осуществляется возврат в текущую задачу или выход из обработчика. И выполняется вторая часть – модификация и запись, – но при этом эти действия выполняются над данными, полученными до ухода, и по их завершении могут быть изменены данные, записанные в другой задаче либо обработчике прерываний. Для избегания такого рода проблем рекомендуется использовать биты установки и сброса значений, например, SRXTX и CRXTX и другие аналогичные

7.13.1.3 **SRXTX**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRXTX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRXTX[15:0]															

Таблица 246 – Описание бит регистра SRXTX

Бит	Имя	Значение	Описание
31…0	SRXTX	0x00000000	Регистр установки порта При чтении отображает состояние выходного триггеров порта. При записи осуществляет запись в выходные триггера порта: SRXTX[0] соответствует выводу порта с номером 0; SRXTX[1] соответствует выводу порта с номером 1 и так далее. При записи 1 производится установка в 1 выходного триггера порта, при записи 0, значение выходного триггера не изменяется. **Внимание!** При чтении из регистра возвращается значение, записанное в выходные триггера, но при этом, если вывод, например, не настроен на выход, считанное значение может отличаться от реального значения на соответствующем выводе порта

7.13.1.4 **CRXTX**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRXTX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRXTX[15:0]															

Таблица 247 – Описание бит регистра CRXTX

Бит	Имя	Значение	Описание
31…0	CRXTX	0x00000000	Регистр сброса порта При чтении отображает состояние выходного триггеров порта. При записи осуществляет запись в выходные триггера порта CRXTX[0] соответствует выводу порта с номером 0 CRXTX[1] соответствует выводу порта с номером 1 и так далее. При записи 1 производится установка в 0 выходного триггера порта, при записи 0, значение выходного триггера не изменяется. **Внимание!** При чтении из регистра возвращается значение, записанное в выходные триггера, но при этом, если вывод, например, не настроен на выход, считанное значение может отличаться от реального значения на соответствующем выводе порта

7.13.1.5 **SOE**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOE[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE[15:0]															

Таблица 248 – Описание бит регистра SOE

Бит	Имя	Значение	Описание
31…0	SOE[31:0]	0x00000000	Регистр установки направления вывода порта. При записи 1 производится установка в 1 сигнала направления порта, при записи 0, значение выходного триггера не изменяется. SOE[0] соответствует выводу порта с номером 0; SOE[1] соответствует выводу порта с номером 1 и так далее. 0 – вывод работает на вход; 1 – вывод работает на выход. Значение направления, задаваемое регистром SOE и COE, имеет смысл при функционировании порта в режиме пользовательского вывода FUNC= 0000. При выборе других функций для вывода, его направление задается в соответствии с выбранной функцией автоматически

7.13.1.6 **COE**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0014								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COE[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COE[15:0]															

Таблица 249 – Описание бит регистра COE

Бит	Имя	Значение	Описание
31…0	COE[31:0]	0x00000000	Регистр установки направления вывода порта При записи 1 производится установка в 0 сигнала направления порта, при записи 0, значение выходного триггера не изменяется. COE[0] соответствует выводу порта с номером 0; COE[1] соответствует выводу порта с номером 1 и так далее. 0 – вывод работает на вход; 1 – вывод работает на выход. Значение направления, задаваемое регистром SOE и COE, имеет смысл при функционировании порта в режиме пользовательского вывода FUNC= 0000. При выборе других функций для вывода, его направление задается в соответствии с выбранной функцией автоматически

7.13.1.7 **SFUNCx**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0018 0x0000_001C 0x0000_0020 0x0000_0024								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FUNC7[3:0]				FUNC6[3:0]				FUNC5[3:0]				FUNC4[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUNC3[3:0]				FUNC2[3:0]				FUNC1[3:0]				FUNC0[3:0]			

Таблица 250 – Описание бит регистров SFUNCx

Бит	Имя	Значение	Описание
31…28	FUNC7[3:0]	0x0	Функция, выполняемая выводом При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется: 0000 – пользовательский вывод; 0001 – Функция 1; … 1111 – Функция 15. Конкретное назначение функции смотрите в описании выводов микросхемы. Выбор аналоговой функции приоритетен. SFUNC0[3…0] – вывод 0; SFUNC0[7…4] – вывод 1; SFUNC0[11…8] – вывод 2; SFUNC0[15…12] – вывод 3; SFUNC0[19…16] – вывод 4; SFUNC0[23…20] – вывод 5; SFUNC0[27…24] – вывод 6; SFUNC0[31…28] – вывод 7; SFUNC1[3…0] – вывод 8; SFUNC1[7…4] – вывод 9; SFUNC1[11…8] – вывод 10; SFUNC1[15…12] – вывод 11; SFUNC1[19…16] – вывод 12; SFUNC1[23…20] – вывод 13; SFUNC1[27…24] – вывод 14; SFUNC1[31…28] – вывод 15; SFUNC2[3…0] – вывод 16; SFUNC2[7…4] – вывод 17; SFUNC2[11…8] – вывод 18; SFUNC2[15…12] – вывод 19; SFUNC2[19…16] – вывод 20; SFUNC2[23…20] – вывод 21; SFUNC2[27…24] – вывод 22; SFUNC2[31…28] – вывод 23; SFUNC3[3…0] – вывод 24; SFUNC3[7…4] – вывод 25; SFUNC3[11…8] – вывод 26; SFUNC3[15…12] – вывод 27; SFUNC3[19…16] – вывод 28; SFUNC3[23…20] – вывод 29; SFUNC3[27…24] – вывод 30; SFUNC3[31…28] – вывод 31
27…24	FUNC6[3:0]	0x0	
23…20	FUNC5[3:0]	0x0	
19…16	FUNC4[3:0]	0x0	
15…12	FUNC3[3:0]	0x0	
11…8	FUNC2[3:0]	0x0	
7…4	FUNC1[3:0]	0x0	
3…0	FUNC0[3:0]	0x0	

7.13.1.8 **CFUNCx**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0028 0x0000_002C 0x0000_0030 0x0000_0034								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FUNC7[3:0]				FUNC6[3:0]				FUNC5[3:0]				FUNC4[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUNC3[3:0]				FUNC2[3:0]				FUNC1[3:0]				FUNC0[3:0]			

Таблица 251 – Описание бит регистров CFUNCx

Бит	Имя	Значение	Описание
31…28	FUNC7[3:0]	0x0	Функция, выполняемая выводом. При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. 0000 – пользовательский вывод; 0001 – Функция 1; … 1111 – Функция 15. Конкретное назначение функции смотрите в описании выводов микросхемы: CFUNC0[3…0] – вывод 0; CFUNC0[7…4] – вывод 1; CFUNC0[11…8] – вывод 2; CFUNC0[15…12] – вывод 3; CFUNC0[19…16] – вывод 4; CFUNC0[23…20] – вывод 5; CFUNC0[27…24] – вывод 6; CFUNC0[31…28] – вывод 7; CFUNC1[3…0] – вывод 8; CFUNC1[7…4] – вывод 9; CFUNC1[11…8] – вывод 10; CFUNC1[15…12] – вывод 11; CFUNC1[19…16] – вывод 12; CFUNC1[23…20] – вывод 13; CFUNC1[27…24] – вывод 14; CFUNC1[31…28] – вывод 15; CFUNC2[3…0] – вывод 16;
27…24	FUNC6[3:0]	0x0	
23…20	FUNC5[3:0]	0x0	
19…16	FUNC4[3:0]	0x0	
15…12	FUNC3[3:0]	0x0	
11…8	FUNC2[3:0]	0x0	
7…4	FUNC1[3:0]	0x0	
3…0	FUNC0[3:0]	0x0	

Бит	Имя	Значение	Описание
			CFUNC2[7…4] – вывод 17; CFUNC2[11…8] – вывод 18; CFUNC2[15…12] – вывод 19; CFUNC2[19…16] – вывод 20; CFUNC2[23…20] – вывод 21; CFUNC2[27…24] – вывод 22; CFUNC2[31…28] – вывод 23; CFUNC3[3…0] – вывод 24; CFUNC3[7…4] – вывод 25; CFUNC3[11…8] – вывод 26; CFUNC3[15…12] – вывод 27; CFUNC3[19…16] – вывод 28; CFUNC3[23…20] – вывод 29; CFUNC3[27…24] – вывод 30; CFUNC3[31…28] – вывод 31

7.13.1.9 **SANALOG**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0038								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SANALOG[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SANALOG [15:0]															

Таблица 252 – Описание бит регистра SANALOG

Бит	Имя	Значение	Описание
31…0	SANALOG[31:0]	0x00000000	Регистр установки типа вывода порта. При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. SANALOG[0] соответствует выводу порта с номером 0; SANALOG[1] соответствует выводу порта с номером 1 и так далее. 0 – вывод выполняет аналоговую функцию; 1 – вывод выполняет цифровую функцию. При выборе аналоговой функции выключается вход вывода микросхемы, и из RXTX всегда будет считана 1 в соответствующем разряде

7.13.1.10 **CANALOG**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_003C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CANALOG[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CANALOG[15:0]															

Таблица 253 – Описание бит регистра CANALOG

Бит	Имя	Значение	Описание
31…0	CANALOG[31:0]	0x00000000	Регистр сброса типа вывода порта При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется: CANALOG[0] соответствует выводу порта с номером 0; CANALOG[1] соответствует выводу порта с номером 1 и так далее. 0 – вывод выполняет аналоговую функцию; 1 – вывод выполняет цифровую функцию. При выборе аналоговой функции выключается вход вывода микросхемы, и из RXTX всегда будет считана 1 в соответствующем разряде

7.13.1.11 **SPULLUP**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0040								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPULLUP[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPULLUP [15:0]															

Таблица 254 – Описание бит регистра SPULLUP

Бит	Имя	Значение	Описание
31…0	SPULLUP[31:0]	0x00000000	Регистр установки подтяжки к питанию вывода порта При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. SPULLUP[0] соответствует выводу порта с номером 0; SPULLUP[1] соответствует выводу порта с номером 1 и так далее. 0 – нет подтяжки; 1 –есть подтяжка к питанию Ucc через резистор R_{PULLUP} с типовым значением 50 кОм

7.13.1.12 **CPULLUP**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0044								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPULLUP[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPULLUP [15:0]															

Таблица 255 – Описание бит регистра CPULLUP

Бит	Имя	Значение	Описание
31…0	CPULLUP[31:0]	0x00000000	Регистр сброса подтяжки к питанию вывода порта При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. SPULLUP[0] соответствует выводу порта с номером 0; SPULLUP[1] соответствует выводу порта с номером 1 и так далее. 0 – нет подтяжки; 1 –есть подтяжка к питанию Ucc через резистор R_{PULLUP} с типовым значением 50 кОм

7.13.1.13 **SPULLDOWN**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0048								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPULLDOWN[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPULLDOWN[15:0]															

Таблица 256 – Описание бит регистра SPULLDOWN

Бит	Имя	Значение	Описание
31…0	SPULLDOWN [31:0]	0x00000000	Регистр установки подтяжки к земле вывода порта. При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. SPULLDOWN[0] соответствует выводу порта с номером 0; SPULLDOWN[1] соответствует выводу порта с номером 1 и так далее. 0 – нет подтяжки; 1 –есть подтяжка к земле через резистор $R_{PULLDOWN}$ с типовым значением 50 кОм

7.13.1.14 **CPULLDOWN**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_004C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPULLDOWN[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPULLDOWN [15:0]															

Таблица 257 – Описание бит регистра CPULLDOWN

Бит	Имя	Значение	Описание
31…0	CPULLDOWN[31:0]	0x00000000	Регистр сброса подтяжки к земле вывода порта При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. SPULLDOWN[0] соответствует выводу порта с номером 0; SPULLDOWN[1] соответствует выводу порта с номером 1 и так далее. 0 – нет подтяжки; 1 –есть подтяжка к земле через резистор $R_{PULLDOWN}$ с типовым значением 50 кОм

7.13.1.15 **SPD**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0050								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPD[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPD [15:0]															

Таблица 258 – Описание бит регистра SPD

Бит	Имя	Значение	Описание
31…0	SPD[31:0]	0x00000000	Регистр установки типа драйвера вывода порта При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. SPD[0] соответствует выводу порта с номером 0; SPD[1] соответствует выводу порта с номером 1 и так далее. 0 – драйвер с активным высоким и низким уровнями; 1 – драйвер с активным низким уровнем (открытый сток)

7.13.1.16 **CPD**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0054								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPD[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPD [15:0]															

Таблица 259 – Описание бит регистра CPD

Бит	Имя	Значение	Описание
31…0	CPD[31:0]	0x00000000	Регистр сброса типа драйвера вывода порта. При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. CPD[0] соответствует выводу порта с номером 0; CPD[1] соответствует выводу порта с номером 1 и так далее. 0 – драйвер с активным высоким и низким уровнями; 1 –драйвер с активным низким уровнем (открытый сток)

7.13.1.17 **SPWRx**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0058 0x0000_005C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWR15		PWR14		PWR13		PWR12		PWR11		PWR10		PWR9		PWR8	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWR7		PWR6		PWR5		PWR4		PWR3		PWR2		PWR1		PWR0	

Таблица 260 – Описание бит регистров SPWRx

Бит	Имя	Значение	Описание
31…30	PWR15[1:0]	0x0	Регистр задания скорости драйвера. При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется; 00 – драйвер выключен; 01 – медленный фронт 300 нс; 10 – средний фронт 100 нс; 11 – быстрый фронт 10 нс. SPWR0[1:0] – вывод 0; SPWR0[3:2] – вывод 1; SPWR0[5:4] – вывод 2; SPWR0[7:6] – вывод 3; SPWR0[9:8] – вывод 4; SPWR0[11:10] – вывод 5; SPWR0[13:12] – вывод 6; SPWR0[15:14] – вывод 7; SPWR0[17:16] – вывод 8; SPWR0[19:18] – вывод 9; SPWR0[21:20] – вывод 10; SPWR0[23:22] – вывод 11; SPWR0[25:24] – вывод 12; SPWR0[27:26] – вывод 13; SPWR0[29:28] – вывод 14; SPWR0[31:30] – вывод 15; SPWR1[1:0] – вывод 16; SPWR1[3:2] – вывод 17; SPWR1[5:4] – вывод 18; SPWR1[7:6] – вывод 19; SPWR1[9:8] – вывод 20; SPWR1[11:10] – вывод 21; SPWR1[13:12] – вывод 22; SPWR1[15:14] – вывод 23; SPWR1[17:16] – вывод 24; SPWR1[19:18] – вывод 25; SPWR1[21:20] – вывод 26; SPWR1[23:22] – вывод 27; SPWR1[25:24] – вывод 28; SPWR1[27:26] – вывод 29; SPWR1[29:28] – вывод 30; SPWR1[31:30] – вывод 31
29…28	PWR14[1:0]	0x0	
27…26	PWR13[1:0]	0x0	
25…24	PWR12[1:0]	0x0	
23…22	PWR11[1:0]	0x0	
21…20	PWR10[1:0]	0x0	
19…18	PWR9[1:0]	0x0	
17…16	PWR8[1:0]	0x0	
15…14	PWR7[1:0]	0x0	
13…12	PWR6[1:0]	0x0	
11…10	PWR5[1:0]	0x0	
9…8	PWR4[1:0]	0x0	
7…6	PWR3[1:0]	0x0	
5...4	PWR2[1:0]	0x0	
3…2	PWR1[1:0]	0x0	
1…0	PWR0[1:0]	0x0	

7.13.1.18 **CPWRx**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0060 0x0000_0064								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWR15		PWR14		PWR13		PWR12		PWR11		PWR10		PWR9		PWR8	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWR7		PWR6		PWR5		PWR4		PWR3		PWR2		PWR1		PWR0	

Таблица 261 – Описание бит регистров CPWRx

Бит	Имя	Значение	Описание
31…30	PWR15[1:0]	0x0	Регистр сброса скорости драйвера. При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется: 00– драйвер выключен; 01 – медленный фронт 300 нс; 10 – средний фронт 100 нс; 11 – быстрый фронт 10 нс. CPWR0[1:0] – вывод 0; CPWR0[3:2] – вывод 1; CPWR0[5:4] – вывод 2; CPWR0[7:6] – вывод 3; CPWR0[9:8] – вывод 4; CPWR0[11:10] – вывод 5; CPWR0[13:12] – вывод 6; CPWR0[15:14] – вывод 7; CPWR0[17:16] – вывод 8; CPWR0[19:18] – вывод 9; CPWR0[21:20] – вывод 10; CPWR0[23:22] – вывод 11; CPWR0[25:24] – вывод 12; CPWR0[27:26] – вывод 13; CPWR0[29:28] – вывод 14; CPWR0[31:30] – вывод 15; CPWR1[1:0] – вывод 16; CPWR1[3:2] – вывод 17; CPWR1[5:4] – вывод 18; CPWR1[7:6] – вывод 19; CPWR1[9:8] – вывод 20;
29…28	PWR14[1:0]	0x0	
27…26	PWR13[1:0]	0x0	
25…24	PWR12[1:0]	0x0	
23…22	PWR11[1:0]	0x0	
21…20	PWR10[1:0]	0x0	
19…18	PWR9[1:0]	0x0	
17…16	PWR8[1:0]	0x0	
15…14	PWR7[1:0]	0x0	
13…12	PWR6[1:0]	0x0	
11…10	PWR5[1:0]	0x0	
9…8	PWR4[1:0]	0x0	
7…6	PWR3[1:0]	0x0	
5...4	PWR2[1:0]	0x0	
3…2	PWR1[1:0]	0x0	
1…0	PWR0[1:0]	0x0	

Бит	Имя	Значение	Описание
			CPWR1[11:10] – вывод 21; CPWR1[13:12] – вывод 22; CPWR1[15:14] – вывод 23; CPWR1[17:16] – вывод 24; CPWR1[19:18] – вывод 25; CPWR1[21:20] – вывод 26; CPWR1[23:22] – вывод 27; CPWR1[25:24] – вывод 28; CPWR1[27:26] – вывод 29; CPWR1[29:28] – вывод 30; CPWR1[31:30] – вывод 31

7.13.1.19 **SCL**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0068								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CL[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CL[15:0]															

Таблица 262 – Описание бит регистра SCL

Бит	Имя	Значение	Описание
31…0	CL[31:0]	0x00000000	Регистр установки разрешения ограничения по току для вывода. При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется: 0 – нет контроля ограничения по току; 1 – есть контроль ограничения по току. CL[0] соответствует выводу порта с номером 0; CL[1] соответствует выводу порта с номером 1 и так далее

7.13.1.20 **CCL**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_006C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CL[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CL[15:0]															

Таблица 263 – Описание бит регистра CCL

Бит	Имя	Значение	Описание
31…0	CL[31:0]	0x00000000	Регистр сброса разрешения ограничения по току для вывода При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. 0 – нет контроля ограничения по току; 1 – есть контроль ограничения по току. CL[0] соответствует выводу порта с номером 0; CL[1] соответствует выводу порта с номером 1 и так далее

7.13.1.21 **SIE**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0070								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IE[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE[15:0]															

Таблица 264 – Описание бит регистра SIE

Бит	Имя	Значение	Описание
31…0	IE[31:0]	0x00000000	Регистр установки разрешения прерывания по выводу. При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется: 0 – прерывание и фиксация прерывания запрещены; 1 – прерывание и фиксация прерывания разрешены. IE[0] соответствует выводу порта с номером 0; IE[1] соответствует выводу порта с номером 1 и так далее

7.13.1.22 **CIE**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0074								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IE[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE[15:0]															

Таблица 265 – Описание бит регистра CIE

Бит	Имя	Значение	Описание
31…0	IE[31:0]	0x00000000	Регистр сброса разрешения прерывания по выводу. При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется: 0 – прерывание и фиксация прерывания запрещены; 1 – прерывание и фиксация прерывания разрешены. IE[0] соответствует выводу порта с номером 0; IE[1] соответствует выводу порта с номером 1 и так далее

7.13.1.23 **SIT**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0078								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT[15:0]															

Таблица 266 – Описание бит регистра SIT

Бит	Имя	Значение	Описание
31…0	IT[31:0]	0x00000000	Регистр установки рабочего уровня прерывания по выводу. При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется: 0 – прерывание и фиксация прерывания по низкому уровню; 1 – прерывание и фиксация прерывания по высокому уровню. Имеет смысл только при разрешении прерывания по соответствующему выводу в регистре IE: IT[0] соответствует выводу порта с номером 0; IT[1] соответствует выводу порта с номером 1 и так далее

7.13.1.24 **CIT**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_007C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IT[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT[15:0]															

Таблица 267 – Описание бит регистра CIT

Бит	Имя	Значение	Описание
31…0	IT[31:0]	0x00000000	Регистр сброса рабочего уровня прерывания по выводу При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. 0 – прерывание и фиксация прерывания по низкому уровню; 1 – прерывание и фиксация прерывания по высокому уровню. Имеет смысл только при разрешении прерывания по соответствующему выводу в регистре IE: IT[0] соответствует выводу порта с номером 0; IT[1] соответствует выводу порта с номером 1 и так далее

7.13.1.25 **SIR**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0080								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR[15:0]															

Таблица 268 – Описание бит регистра SIR

Бит	Имя	Значение	Описание
31…0	IR[31:0]	0x00000000	Регистр установки запроса прерывания по выводу. При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется: 0 – нет запроса прерывания; 1 – есть запрос прерывания. Имеет смысл только при разрешении прерывания по соответствующему выводу в регистре IE: IR[0] соответствует выводу порта с номером 0; IR[1] соответствует выводу порта с номером 1 и так далее

7.13.1.26 **CIR**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0084								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR[15:0]															

Таблица 269 – Описание бит регистра CIR

Бит	Имя	Значение	Описание
31…0	IR[31:0]	0x00000000	Регистр сброса запроса прерывания по выводу. При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется. 0 – нет запроса прерывания; 1 – есть запрос прерывания. Имеет смысл только при разрешении прерывания по соответствующему выводу в регистре IE: IR[0] соответствует выводу порта с номером 0; IR[1] соответствует выводу порта с номером 1 и так далее

7.13.1.27 **HCUR**

Base ADDR=		0x4008_0000 0x4008_1000 0x4008_2000 0x4008_3000 0x4008_4000			Offset=		0x0000_0088								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HCUR[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCUR[15:0]															

Таблица 270 – Описание бит регистра HCUR

Бит	Имя	Значение	Описание
31…0	HCUR[31:0]	0x00000000	Регистр флага превышения по току по выводу. При записи 1 производится сброс флага, если при этом закончилось событие превышения: 0 – нет превышения по току; 1 – есть превышение по току. HCUR[0] соответствует выводу порта с номером 0; HCUR [1] соответствует выводу порта с номером 1 и так далее

7.14 Контроллер Ethernet (EthernetMAC)

Память блока контроллера Ethernet имеет встроенную защиту ECC. Встроенный PHY обеспечивает работу на скорости до 10 Мбит/с. Контроллер MAC поддерживает режимы работы 10 и 100 Мбит/с. При режиме 100 Мбит/с необходимо применение внешнего PHY, подключаемого по интерфейсу MII, а также установление максимальных частот FCLK = HCLK = 62,5 МГц. В самом протоколе Ethernet на уровне пакетов используется защита данных по CRC.

Встроенная память контроллера EthernetMAC отображается в адресное пространство на уровне арбитража шины AHB, при этом регистры управление контроллером Ethernet отображены в адресное пространство периферии. Отображаемая память EthernetMAC снабжена защитой ECC аналогично памяти RAMD. Передаваемые и принимаемые данные по сети Ethernet защищены контрольными суммами CRC согласно протоколу стандарта Ethernet. При приеме по сети Ethernet данные формируются в 32-разрядные слова, для которых генерируется ECC и записываются в память. При считывании со стороны процессора проверка ECC производится на входе в процессор. При записи со стороны процессора данные в память записываются с проверкой ECC и генерацией новой ECC, аналогично работе памяти RAMD.

Объем памяти контроллера Ethernet составляет 8 Кбайт.

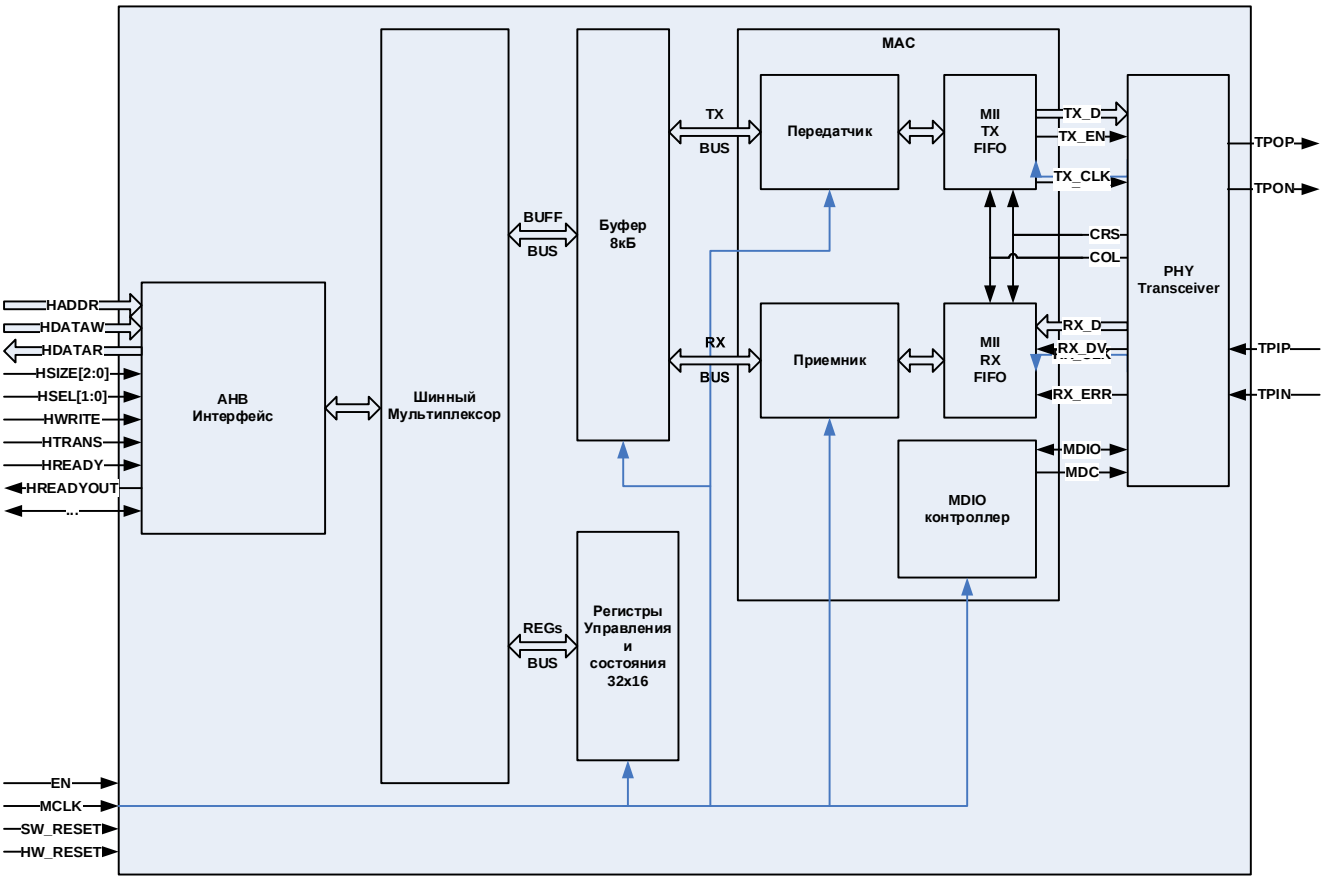


Рисунок 71 – Структурная схема контроллера Ethernet

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.

Работа блока возможна в полно- и полудуплексном режимах. Переключение режима осуществляется битом G_CFG.HD_EN: («1» – полудуплексный режим работы).

Также в общее управление блоком входит разрешение обработки пакета PAUSE (G_CFG.PAUSE_EN), управление размером окна распознаваний коллизий (G_CFG.ColWnd) и режимом работы буферов приемника и передатчика (G_CFG.BUFF_MODE).

Режимы работы буферов:

– линейный (BUFF_MODE=00b);
– автоматический (BUFF_MODE=01b);
– режим FIFO (BUFF_MODE=10b).

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 последнего чтения или записи в соответствующие области;

– в режиме FIFO границы определяются автоматически, при этом обращение к приемнику производится через адрес 0x0000, а к передатчику через адрес 0x0004. Размер области памяти контроллера Ethernet выделяемой под буфер приемника и буфер передатчика можно менять с помощью регистра DELIMITER. По сбросу граница между буфером приемника и передатчика устанавливается посередине, т.е в 0x1000.

7.14.1 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);
– выбор события при передаче, выводимого на вывод EVNT[1] (поле EVNT_MODE);
– 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;
– собственно, данные пакета уровня звена данных;
– поле состояния передачи пакета.

Поле управления содержит количество передаваемых данных пакета в байтах. 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 прочее.

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.

Во время отправки пакета или по его завершении выставляется событие, запрограммированное в поле X_CFG.EVNT_MODE.

7.14.2 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 (X_CFG.EN=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.

7.14.3 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);
– выбор события при передаче, выводимого на вывод EVNT[0] (поле EVNT_MODE);
– 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 принимаемого пакета.

Принятый пакет содержит два поля (все поля выровнены по границе слова буферного ОЗУ):

- поле состояния приема пакета;
- собственно, данные пакета уровня звена данных.

Поле состояния заполняется по успешном завершении процедуры приема пакета и содержит количество байт в пакете (включая заголовок пакет 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.

Во время приема пакета или по его завершении выставляется событие, запрограммированное в поле R_CFG.EVNT_MODE.

7.14.4 Принцип работы приемника

Приемник начинает работать сразу же после разрешения работы приемника в регистре R_CFG (R_CFG.EN=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. Затем переходит в режим ожидания данных на входе и после поступления данных – в режим приема. По завершении приема в слово, следующее за последним словом данных, записывается статусная информация по приему пакета. Пакеты, отброшенные по причине ошибок в них или не прошедшие фильтрацию по MAC-адресу, переводят приемник в режим ожидания нового пакета. Таким образом, не изменяется общее состояние приемника, изменяется лишь состояние регистра флагов прерываний.

7.14.5 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.

7.14.6 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, а также 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 режиме отладки необходимо активировать биты G_CFG.DBG_XF_EN и G_CFG.DBG_RF_EN.

7.14.7 Режим FIFO работы буферов

Режим FIFO отличается от предыдущих режимов полностью автоматическим отслеживанием данных в буфере. В данный режим модель 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 режиме отладки необходимо активировать биты G_CFG.DBG_XF_EN и G_CFG.DBG_RF_EN. Данный режим позволяет работать на максимальной скорости, если инструментальные средства управляющего контроллера не обеспечивают режимы адресации с автоинкрементом и циклической буферизацией.

7.14.8 События приемника и передатчика

В блоке MAC присутствуют два вывода индикации событий: события передатчика (EVNT[1]) и приемника (EVNT[0]). Основным назначением этих выводов является информирование управляющего процессора или DMA-контроллера о наличии данных для перемещения. Выводы EVNT программируемые. В качестве источника события могут быть выбраны:

- начало приема/передачи пакета;
- завершение приема/передачи пакета;
- перемещение слова из/в буфера.

7.14.9 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 MDIO интерфейса информируют о завершении затребованной операции по MDIO интерфейсу.

Прерывания передатчика показывают состояние отправки пакетов, включая информацию об успешной отправке или наличии ошибок.

Прерывания приемника отражают состояние приема пакета, включая информацию о приеме пакета без ошибок или наличии ошибок при приеме.

Все флаги прерываний кумулятивные. Сброс флагов производится чтением регистра, если бит RCLR_EN регистра G_CFGl установлен (G_CFGl. RCLR_EN = 1), или записью 1 в соответствующий разряд регистра IFR.

7.14.10 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 (G_CFGl.DTRM_EN = 1). Данный режим может использоваться только в полнодуплексном режиме работы (значение бита G_CFGl.HD_EN = 1 блокирует данный режим).

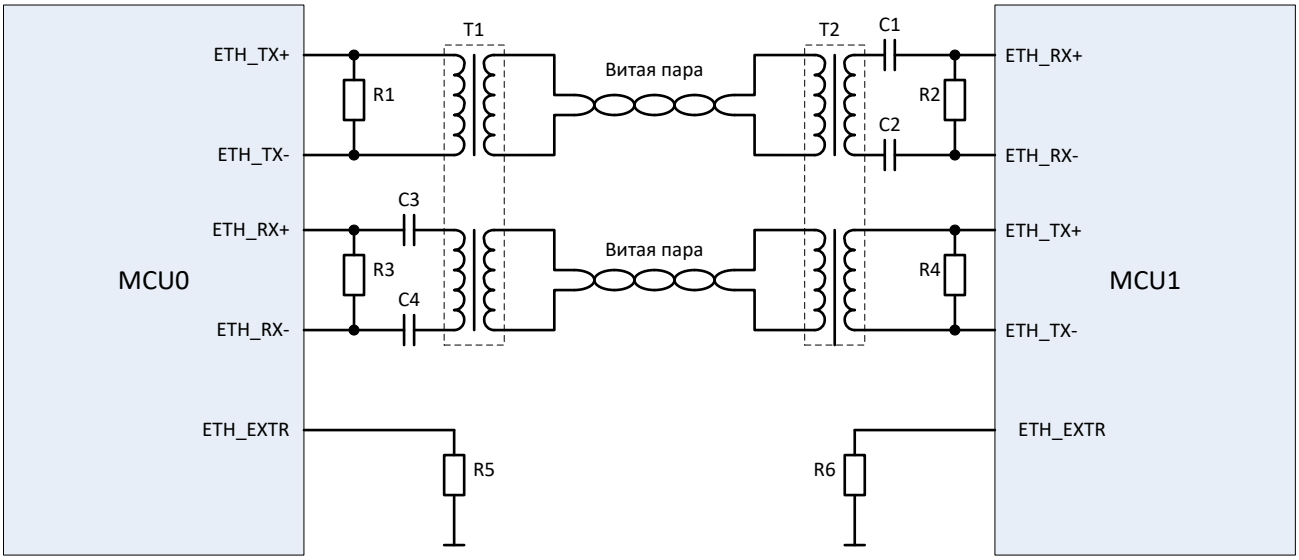
В данном режиме для начала передачи пакета выделяется интервал размером, задаваемым регистром JitterWnd (размер окна = JitterWnd + 1), с периодом задаваемым регистром BAG (период = BAG + 1). Единица измерения периода и размера джиттера задается регистром PSC в тактах основной частоты работы блока (размер единицы = PSC + 1).

7.14.11 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.

7.14.12 Блок PHY

Схема включения интерфейса Ethernet с использованием встроенных приемопередатчиков приведена на рисунке 72.



C1, C2, C3, C4 – конденсаторы емкостью 100 нФ;
R1, R2, R3, R4 – резисторы сопротивлением100 Ом ± 1 %;
R5, R6 – резисторы сопротивлением 120 кОм ± 1 %;
T1, T2 – трансформаторы для Ethernet линии.

Рисунок 72 – Схема подключения к линии

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. Для корректной работы блока необходимо настроить частоту тактирования EPHYCLK равной 80 МГц через регистр EPHY_CLK блока CLKCNTR.

7.14.13 Описание регистров

Таблица 271 – Описание регистров контроллера Ethernet

Базовый адрес	Название	Состояние после сброса	Описание
0x2100_8000	ETH0 RAM		Буфер данных контроллера Ethernet 0
0x2100_0000	ETH0 REGS		Регистры управления контроллера Ethernet 0
Смещение			
0x0000_0000	DELIMITER	RW, 0x1000	Регистр границы буферов приемника и передатчика
0x0000_0002	MAC_ADDRESS		Регистр индивидуального MAC-адреса
0x0000_0002	MAC_T	RW, 0x78AB	Младшая* часть индивидуального MAC-адреса

Базовый адрес	Название	Состояние после сброса	Описание
0x0000_0004	MAC_M	RW, 0x3456	Средняя часть индивидуального MAC-адреса
0x0000_0006	MAC_H	RW, 0x0012	Старшая часть индивидуального MAC-адреса
0x0000_0008	HASH		HASH-таблица групповых адресов
0x0000_0008	HASH0	RW, 0x0000	Младшая часть HASH-таблицы
0x0000_000A	HASH1	RW, 0x0000	Средняя часть HASH-таблицы
0x0000_000C	HASH2	RW, 0x0000	Средняя часть HASH-таблицы
0x0000_000E	HASH3	RW, 0x8000	Старшая часть HASH-таблицы
0x0000_0010	IPG	RW, 0x0060	Регистр задания межпакетного интервала для полнодуплексного режима
0x0000_0012	PSC	RW, 0x0050	Регистр задания предделителя шага изменения значений BAG и JitterWnd (1 мкс при частоте 50 МГц)
0x0000_0014	BAG	RW, 0x0200	Регистр задания периода следования пакетов (100 мкс при частоте 50 МГц)
0x0000_0016	JITTERWND	RW, 0x0005	Регистр задания джиттера при передаче пакетов (5 мкс при частоте 50 МГц)
0x0000_0018	R_CFG	RW, 0x0507	Регистр управления приемника
0x0000_001A	X_CFG	RW, 0x01FA	Регистр управления передатчика
0x0000_001C	G_CFGl	RW, 0x4880	Регистр общего управления блоком, младшее полуслово
0x0000_001E	G_CFGh	RW, 0x3000	Регистр общего управления блоком, старшее полуслово
0x0000_0020	IMR	RW, 0x0000	Регистр маски прерываний
0x0000_0022	IFR	RW, 0x0000	Регистр флагов прерываний
0x0000_0024	MDIO_CTRL	RW, 0x0000	Регистр управления канала MDIO интерфейса MII
0x0000_0026	MDIO_DATA	RW, 0x0000	Регистр данных канала MDIO интерфейса MII
0x0000_0028	R_Head	RW, 0x0000	Указатель начала области действительных данных приемника (указывает на первое непустое слово)
0x0000_002A	X_TAIL	RW, 0x1000	Указатель конца области действительных данных передатчика (указывает на первое пустое слово)
0x0000_002C	R_TAIL	R, 0x0000	Указатель начала области действительных данных приемников (указывает на первое непустое слово)
0x0000_002E	X_HEAD	R, 0x1000	Указатель начала области действительных данных передатчика (указывает на первое непустое слово)
0x0000_0030	STAT	R, 0x0303	Регистр статуса

Базовый адрес	Название	Состояние после сброса	Описание
0x0000_0032	RCOUNTER	R, 0x0000	Счетчик количества принятых пакетов (циклический, без насыщения)
0x0000_0034	PHY_CTRL	RW	Регистр управления PHY
0x0000_0036	PHY_STATUS	R	Регистр флагов статуса PHY
0x0000_0038	PHY_CONTROL_A	RW	Регистр управления аналогового PHY
* 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)			

7.14.13.1 **Поле управления передачей пакета**

Base ADDR=					Offset=										
REG Name:		CHIPID													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Length[15:0]															

Таблица 272 – Описание бит поля управления передачи пакета

Бит	Имя	Значение	Описание
31…16	-		Зарезервировано
15…0	Length[15:0]		Количество байт в пакете

7.14.13.2 **Поле состояния передачи пакета**

Base ADDR=					Offset=										
REG Name:		CHIPID													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-									-	LC	RL	RCOUNT[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Таблица 273 – Описание бит поля состояния передачи пакета

Бит	Имя	Значение	Описание
31…23	-		Зарезервировано
22	-	0	Зарезервировано
21	LC		Флаг индикации Latecollision во время передачи пакета 1 – произошла Latecollision во время передачи пакета; 0 – Latecollision во время передачи пакета не происходила
20	RL		Флаг исчерпания попыток передачи пакета 1 – превышено разрешенное количество попыток передачи пакета; 0 – количество попыток передачи пакета не превысило разрешенного значения
19…16	RCOUNT[3:0]		Число попыток передачи пакета
15…0	-		Зарезервировано

7.14.13.3 **Поле состояния приема пакета**

Base ADDR=					Offset=										
REG Name:		CHIPID													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-					UCA	BCA	MCA	SMB_ERR	CRC_ERR	DN_ERR	LEN_ERR	SF_ERR	LF_ERR	CF_ERR	PF_ERR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Length[15:0]															

Таблица 274 – Описание бит поля состояния приема пакета

Бит	Имя	Значение	Описание
31…27			Зарезервировано
26	UCA		Признак индивидуального пакета (MAC-адрес соответствует установленному): 1 – MAC-адрес принятого пакета совпадает с MAC-адресом Ethernet-контроллера; 0 – MAC-адрес принятого пакета не совпадает с MAC-адресом Ethernet-контроллера
25	BCA		Признак широковещательного пакета (MAC = FF_FF_FF): 1 – принят широковещательный пакет; 0 – широковещательный пакет не принят

Бит	Имя	Значение	Описание
24	MCA		Признак группового пакета (MAC соответствует HASH): 1 – принят пакет, удовлетворяющий фильтрации по HASH-таблице; 0 – принятый пакет не удовлетворяет фильтрации по HASH-таблице или фильтрация отключена
23	SMB_ERR		Признак наличия в пакете ошибочных nibbles: 1 – наличие; 0 – отсутствие
22	CRC_ERR		Признак несоответствия CRC пакета: 1 – произошла ошибка сравнения CRC-пакета с вычисленной CRC; 0 – CRC-пакета и вычисленной CRC совпадают
21	DN_ERR		Количество бит в пакете не кратно 8: 1 – не кратно 8; 0 – кратно 8
20	LEN_ERR		Признак несоответствия между реальной длиной и длиной, указанной в поле длины – 13,14 октеты: 1 – несоответствие; 0 – соответствие
19	SF_ERR		Признак недостаточной длины пакета 64 октетов: 1 – ошибочная длина; 0 – корректная длина
18	LF_ERR		Признак превышение длины пакета 1518 октетов: 1 – превышение; 0 – норма
17	CF_ERR		Признак пакета управления (фильтрация по специальным MAC и тэгам в поле длины – 13,14 – октеты): 1 – пакет управления; 0 – другой пакет
16	PF_ERR		Признак пакета PAUSE: 1 – пакет PAUSE; 0 – другой пакет
15…0	Length[15:0]		Количество байт в пакете, включая заголовок и CRC

7.14.13.4 **DELIMITER**

Base ADDR=		0x2100_0000			Offset=		0x0000_0000								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELIMITER[15:0]															

Таблица 275 – Описание бит регистра DELIMITER

Бит	Имя	Значение	Описание
15…0	DELIMITER[15:0]	1000	Смещение границы между буферами приемника и передатчика относительно базового адреса буфера данных контроллера (не путать с базовым адресом регистров контроллера)

7.14.13.5 **MAC_T**

Base ADDR=		0x2100_0000			Offset=		0x0000_0002								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_T[15:0]															

Таблица 276 – Описание бит регистра MAC_T

Бит	Имя	Значение	Описание
15…0	MAC_T[15:0]	78AB	Младшая часть индивидуального MAC-адреса

7.14.13.6 **MAC_M**

Base ADDR=		0x2100_0000			Offset=		0x0000_0004								
REG Name:		KEY													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_M[15:0]															

Таблица 277 – Описание бит регистра MAC_M

Бит	Имя	Значение	Описание
15…0	MAC_M[15:0]	3456	Средняя часть индивидуального MAC-адреса

7.14.13.7 **MAC_H**

Base ADDR=		0x2100_0000			Offset=		0x0000_0006								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_H[15:0]															

Таблица 278 – Описание бит регистра MAC_H

Бит	Имя	Значение	Описание
15…0	MAC_H[15:0]	0012	Старшая часть MAC-адреса

7.14.13.8 **R_CFG**

Base ADDR=		0x2100_0000			Offset=		0x0000_0018								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	-	BE	MSB1st	-	EVNT_ MODE			SF_EN	LF_EN	CF_EN	EF_EN	AC_EN	UCA_EN	BCA_EN	MCA_EN

Таблица 279 – Описание бит регистра R_CFG

Бит	Имя	Значение	Описание
15	EN		Разрешение работы приемника: 0 – приемник остановлен; 1 – разрешена работа
14	-		Зарезервировано
13	BE		Порядок следования байт в слове: 0 – LittleEndian; 1 – BigEndian
12	MSB1st		Порядок следования бит при приеме байтов данных: 0 – первым принимается LSB; 1 – первым принимается MSB
11	-		Зарезервировано
10…8	EVNT_MODE[2:0]		Выбор режима работы вывода EVNT[0]: 3’b000 – зарезервировано; 3’b001 – зарезервировано; 3’b010 – зарезервировано; 3’b011 – зарезервировано; 3’b100 – зарезервировано; 3’b101 – прием пакета завершен; 3’b110 – приемник положил данные в буфер; 3’b111 – приемник отбросил пакет
7	SF_EN		Разрешение приема пакетов длиной меньше минимальной. 0 – выключено; 1 – включено
6	LF_EN		Разрешение приема пакетов длиной больше максимальной: 0 – выключено; 1 – включено
5	CF_EN		Разрешение приема управляющих пакетов: 0 – выключено; 1 – включено

Бит	Имя	Значение	Описание
4	EF_EN		Разрешение приема пакетов с ошибками: 0 – выключено; 1 – включено
3	AC_EN		Прием пакетов без фильтрации MAC-адреса: 0 – выключен; 1 – включен
2	UCA_EN		Прием пакетов с MAC-адресом, указанным в регистре MAC_Address: 0 – выключен; 1 – включен
1	BCA_EN		Прием пакетов с широковещательным MAC-адресом: 0 – выключен; 1 – включен
0	MCA_EN		Прием пакетов с групповым MAC-адресом с фильтрацией по HASH-таблице: 0 – выключен; 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			

7.14.13.9 **X_CFG**

Base ADDR=		0x2100_0000			Offset=		0x0000_001A								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	-	BE	MSB1st	-	EVNT_ MODE			PAD_EN	PRE_EN	CRC_EN	IPG_EN	RtryCnt			

Таблица 280 – Описание бит регистра X_CFG

Бит	Имя	Значение	Описание
15	EN		Разрешение работы передатчика: 0 – остановлен; 1 – разрешена работа
14	-		Зарезервировано
13	BE		Порядок следования байт в слове передатчика: 0 – LittleEndian; 1 – BigEndian

Бит	Имя	Значение	Описание
12	MSB1st		Порядок следования бит при передаче байтов данных: 0 – первым передается LSB; 1 – первым передается MSB
11	-		Зарезервировано
10…8	EVNT_MODE[2:0]		Выбор режима работы вывода EVNT[1]: 3’b000 – зарезервировано; 3’b001 – зарезервировано; 3’b010 – зарезервировано; 3’b011 – зарезервировано; 3’b100 – зарезервировано; 3’b101 – отправка пакета завершена; 3’b110 – передатчик считал слово данных из буфера; 3’b111 – передатчик начал очередную попытку передачи пакета
7	PAD_EN		Дополнение пакета до минимальной длины PAD-ами: 0 – выключено; 1 – включено
6	PRE_EN		Дополнение пакета преамбулой: 0 – выключено; 1 – включено
5	CRC_EN		Дополнение пакета автоматически высчитанным CRC: 0 – выключено; 1 – включено
4	IPG_EN		Режим выдержки паузы между отправкой пакетов. 0 – выключен: 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			

7.14.13.10 **G_CFGl**

Base ADDR=		0x2100_0000			Offset=		0x0000_001C								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	RCLR_EN	BUFF_MODE [1:0]		EXT_EN	HD_EN	DTRM_EN	PAUSE_EN	ColWnd [7:0]							

Таблица 281 – Описание бит регистра G_CFGI

Бит	Имя	Значение	Описание
15	-		Зарезервировано
14	RCLR_EN		Сброс регистров статуса: 0 – производиться записью в регистры статуса; 1 – регистры статуса сбрасываются при чтении
13, 12	BUFF_MODE[1:0]		Режим работы буфера: 2’b00 – линейный режим; 2’b 01 – режим с автоматическим изменением указателей; 2’b 10 – режим FIFO; 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): 0 – выключен; 1 – включен
10	HD_EN		Полудуплексный режим работы: 0 – выключен; 1 – включен
9	DTRM_EN		Режим детерминированного времени доставки: 0 – выключен; 1 – включен
8	PAUSE_EN		Режим автоматической обработки пакета PAUSE: 0 – выключен; 1 – включен
7…0	ColWnd[7:0]		Размер «окна коллизий»

7.14.13.11 G_CFGh

Base ADDR=		0x2100_0000	Offset=	0x0000_001E											
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		DBGXF_EN	DBGRF_EN	-									DLB	RRST	XRST

Таблица 282 – Описание бит регистра G_CFGh

Бит	Имя	Значение	Описание
15…14	-		Зарезервировано
13	DBGXF_EN		Разрешение автоматического изменения указателей FIFO передатчика: 0 – запрещено; 1 – разрешено

Бит	Имя	Значение	Описание
12	DBGRF_EN		Разрешение автоматического изменения указателей FIFO приемника: 0 – запрещено; 1 – разрешено
11...3	-		Зарезервировано
2	DLB		Режим КЗ: 0 – выключен; 1 – включен
1	RRST		Сброс приемника: 0 – работает; 1 – сброшен
0	XRST		Сброс передатчика: 0 – работает; 1 – сброшен

7.14.13.12 **IMR/IFR**

Base ADDR=		0x2100_0000			Offset=		0x0000_0020 0x0000_0022								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MII_RDY	MDIO_INT	-	CRS_LOST	LC	UNDF	XF_ERR	XF_OK	SF	LF	CF	CRC_ERR	SMB_ERR	OVF	MISSED_F	RF_OK

Таблица 283 – Описание бит регистра IMR/IFR

Бит	Имя	Значение	Описание
31…16	-		Зарезервировано
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 длиной менее минимальной

Бит	Имя	Значение	Описание
6	LF		Индикатор приема пакета длиной более 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			

7.14.13.13 MDIO_CTRL

Base ADDR=		0x2100_0000			Offset=		0x0000_0024								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDY	PRE_EN	OP	PHY_A					DIV			RG_A				

Таблица 284 – Описание бит регистра MDIO_CTRL

Бит	Имя	Значение	Описание
15	RDY		Управление/индикатор обмена по MDIO. После записи команды необходимо установить в единицу 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: 1 – с передачей преамбулы (32 бита «1»); 0 – без передачи преамбулы
13	OP		Операция: 1 – чтение; 0 – запись
12…8	PHY_A[4:0]		Адрес модуля PHY
7…5	DIV[2:0]		Коэффициент деления основной частоты для работы MDIO интерфейса: MDC= ETH_CLK / [(DIV+1)•16]
4…0	RG_A[4:0]		Номер регистра PHY

7.14.13.14 STAT

Base ADDR=	0x2100_0000	Offset=	0x0000_0030						

REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			X_FULL	X_AFULL	X_HALF	X_AEMPY	X_EMPY	RCOUNT			R_FULL	R_AFULL	R_HALF	R_AEMPY	R_EMPY

Таблица 285 – Описание бит регистра STAT

Бит	Имя	Значение	Описание
15…13	-		Зарезервировано
12	X_FULL		1 – буфер передатчика полон; 0 – буфер передатчика не полон
11	X_AFULL		1 – буфер передатчика почти полон; 0 – буфер передатчика не в состоянии почти полон
10	X_HALF		1 – буфер передатчика полуполон; 0 – буфер передатчика не полуполон
9	X_AEMPTY		1 – буфер передатчика почти пуст; 0 – буфер передатчика не в состоянии почти пуст
8	X_EMPTY		1 – буфер передатчика пуст; 0 – буфер передатчика не пуст
7…5	R_COUNT		Количество принятых, но не считанных пакетов: 0…6 – количество пакетов; 7 – количество несчитанных пакетов ≥ 7. 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; 0 – буфер приемника не полон
3	R_AFULL		1 – буфер приемника почти полон; 0 – буфер приемника не в состоянии почти полон
2	R_HALF		1 – буфер приемника полуполон; 0 – буфер приемника не полуполон
1	R_AEMPTY		1 – буфер приемника почти пуст; 0 – буфер приемника не в состоянии почти пуст
0	R_EMPTY		1 – буфер приемника пуст; 0 – буфер приемника не пуст

7.14.13.15 **PHY_CTRL**

Base ADDR=		0x2100_0000			Offset=		0x0000_0034								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RST	EXT_EN	-	LINK_ PERIOD	BASE_2	DIR	EARLY_DV	HALFD	DLB	LB

Таблица 286 – Описание бит регистра PHY_CTRL

Бит	Имя	Значение	Описание
15	RST		Сброс встроенного контроллера уровня PHY: 0 – штатный режим; 1 – сброс модуля PHY
14	EXT_EN		Переключение на работу с внешним контроллером уровня PHY: 0 – внутренний блок PHY; 1 – внешний блок PHY (интерфейс MII)
13…12	-		Зарезервировано
11…6	LINK_PERIOD		Период следования LINK-импульсов -1. Период ожидания LINK-импульсов вдвое больше заданного периода следования LINK-импульсов (диапазон 1...64 мс)
5	BASE_2		Переключение на работу с коаксиальным кабелем в режиме полудуплексного приема-передачи: 0 – подключение по витой паре; 1 – подключение по коаксиальному кабелю
4	DIR		Порядок передачи битов в полубайте: 1 – инверсный; 0 – прямой
3	EARLY_DV		Включение формирования сигнала RxDV одновременно с сигналом CRS (зарезервировано)
2	HALFD		Включение режима полудуплексного приема-передачи: 0 – полнодуплексный режим; 1 – полудуплексный режим
1	DLB		Включение тестового замыкания ПРД на ПРМ на входе контроллера уровня PHY: 0 – штатный режим; 1 – режим КЗ
0	LB		Включение тестового замыкания ПРД на ПРМ на выходе контроллера PHY-уровня до аналоговой части ПРМ/ПРД: 0 – штатный режим; 1 – режим КЗ

7.14.13.16 PHY_STATUS

Base ADDR=		0x2100_0000			Offset=		0x0000_0036								
REG Name:															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_JAM	INT_JAB	-	INT_LOSS	LED_STAT	LED_LINK	-									

Таблица 287 – Описание бит регистра PHY_STATUS

Бит	Имя	Значение	Описание
15	INT_JAM		Индикатор встроенного контроллера PHY-уровня о передаче JAM последовательности в случае коллизии (для тестирования): 0 – штатный режим; 1 – осуществляется передача JAM-последовательности
14	INT_JAB		Индикатор встроенного контроллера PHY-уровня о превышении времени передачи максимально разрешенной (для тестирования): 0 – штатный режим; 1 – передача текущего пакета ведется более максимально допустимого интервала
13	-		Зарезервировано
12	INT_LOSS		Индикатор встроенного контроллера PHY-уровня о наличии подключения в линии: 0 – линия подключена; 1 – потеря синхронизации
11	LED_STAT		Индикация режима работы блока PHY: 0 – режим работы full-duplex; 1 – режим работы half-duplex
10	LED_LINK		Индикатор встроенного контроллера PHY-уровня о наличии несущей в линии (для тестирования): 0 – соединение отсутствует; 1 – установлено соединение (при условии, что идет обмен данными CRS=1)
9...0	-		Зарезервировано

7.14.13.17 PHY_CONTROL_A

Base ADDR=		0x2100_0000			Offset=		0x0000_0038								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN_PHY	ISRC	-		TRIMR[7:0]								TRIM[2:0]			SELR

Таблица 288 – Описание бит регистра PHY_CONTROL_A

Бит	Имя	Значение	Описание
31…16	-		Зарезервировано
15	EN_PHY	0	Бит разрешения работы аналогового PHY: 1 – разрешен; 0 – запрещен
14	ISRC	0	Бит увеличение рабочего тока передатчика при низком напряжении питания: 1 – должен быть установлен при питании ниже 4,0 В; 0 – должен быть сброшен при питании выше 4,0 В
13…12	-		Зарезервировано
11…4	TRIMR[7:0]	0000_0000	Вход подстройки внутреннего резистора источника опорного тока: 1000_0000 Типовое значение опорного тока (10 мкА); 0000_0000 Минимальный опорный ток (5 мкА); 1111_1111 Максимальный опорный ток (15 мкА)
3…1	TRIM[2:0]	000	Вход цифрового тримминга источника опорного напряжения: Типовое значение 100 (1,2 В); 000 – максимальное напряжение (1,25 В); 111 – минимальное напряжение (1,15 В)
0	SELR	0	Вход выбора резистора источника опорного тока: 0 – используется внутренний подстраиваемый резистор (вывод EXTR в третьем состоянии и толерантен); 1 – используется внешний стабильный резистор на выводе ETH_EXTR на землю

7.15 Контроллер SpaceWire (SpaceWire)

7.15.1 Кодирование данных

Каналы SpaceWire – это последовательные высокоскоростные (от 2 до 400 Мб/с) полнодуплексные каналы (DS-линки) типа «точка-точка». Стандарт SpaceWire использует Data-Strobe (DS)-кодирование для передачи данных, которое изображено на рисункерисунке 73.

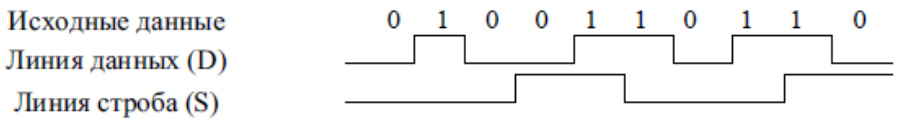


Рисунок 73 – Data-Strobe (DS)-кодирование

Данная схема позволяет кодировать синхросигнал передачи в линиях данных таким образом, что синхросигнал может быть восстановлен операцией исключающее ИЛИ двух линий D и S. Линия D повторяет значение бита данных, то есть имеет высокий потенциал для бита логической «1» и низкий потенциал для логического «0». Линия S изменяет свое состояние каждый раз, когда состояние линии D остается неизменным для следующего бита.

Канал SpaceWire включает две пары дифференциальных сигналов, одна пара D и S передающая, другая пара D и S принимающая. В результате для двунаправленного канала требуется восемь однонаправленных линий передачи данных.

7.15.2 Отказоустойчивая работа приемопередатчика SpaceWire PHY

По стандарту SpaceWire при возникновении любого из следующих условий выходы приемника SpaceWire PHY должны быть зафиксированы в высоком логическом уровне в том случае, если порог шума на входе приемника не превышает 10 мВ:

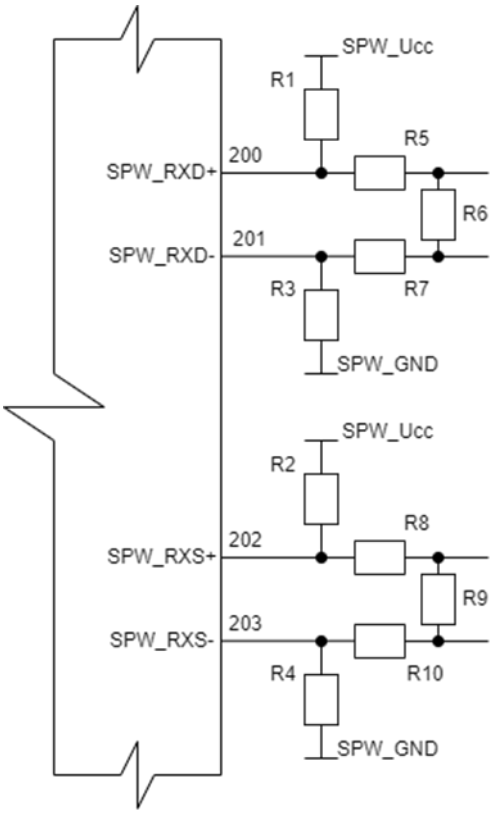
– драйвер не запитан;

– драйвер отключен (то есть выходы драйвера находятся в высокоомном состоянии);

– входы приемника разомкнуты (то есть отсоединен кабель или провод в кабеле).

Дифференциальные линии приемника должны быть терминированы резистором номинала 100 Ом ± 1 %.

Для обеспечения безотказной работы или повышения устойчивости к помехам на входе приемника используются внешние резисторы смещения, при этом обеспечиваемый ими ток смещения должен быть менее 0,25 мА. На рисунке 74 приведена рекомендуемая схема подключения выводов приемника SpaceWire PHY с учетом терминирующих резисторов.



R1–R2 – резисторы сопротивлением 18 кОм ± 1 %;
R3–R4 – резисторы сопротивлением 5,6 кОм ± 1 %;
R5–R10 – резисторы сопротивлением 100 Ом ± 1 %.

Рисунок 74 – Рекомендуемая схема подключения выводов приемника SpaceWire PHY

7.15.3 Скорость интерфейса

Скорость работы интерфейса на передачу определяется частотой синхронизации SPWCLK, которая должна быть равна частоте контроллера SpaceWire(=CPUCLK) или ее удвоенному значению. Каналы приема и передачи в интерфейсе асинхронны. Скорость интерфейса на прием задается внешним передатчиком. Для корректной работы приемника скорость в канале приема не должна превышать частоту CPUCLK более чем в два раза. Например, если частота CPUCLK равна 40 МГц, максимально возможная скорость интерфейса не будет превышать 80 Мбит/с.

7.15.4 Представление символов

Битовый поток состоит из символов данных и управления. Тип символа определяется в зависимости от значения флага данные/управление. Символы данных содержат восьмиразрядные значения данных, передаваемых младшими разрядами вперед. Каждый символ данных состоит из бита паритета, контрольного флага и восьми бит данных. Контрольный флаг для символа данных равен нулю. Структура символа данных показана на рисунке 75.

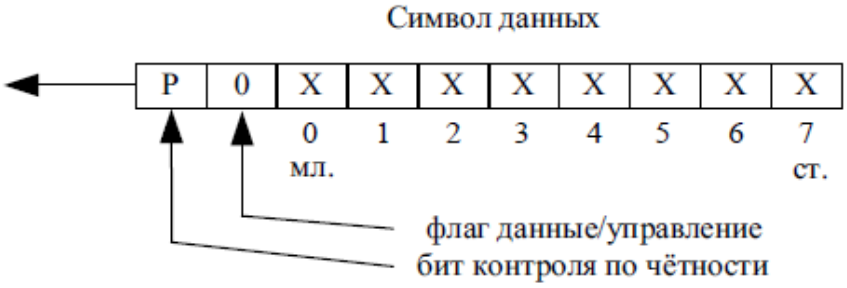


Рисунок 75 – Символ данных SpaceWire

Управляющий символ состоит из бита контроля по четности, флага данные/управление и двух бит кода управления. Значение флага данные/управление равное 1 указывает, что данный символ является символом управления. Возможные управляющие символы показаны на рисунке 76. Символ расширения ESC используется для создания дополнительных управляющих символов – маркеров. В спецификации SpaceWire символ ESC используется исключительно для формирования NULL-маркера или маркера времени. Все другие применения этого символа считаются некорректными и рассматриваются как ошибка расширения. NULL-маркер состоит из символа расширения ESC и следующим за ним символом управления потоком FCT. Бит контроля по четности P в середине маркера NULL равен 0. NULL-маркер передается всегда при отсутствии передач символов данных и символов управления в канале. Это позволяет сохранить активность канала и определить ошибку рассоединения.

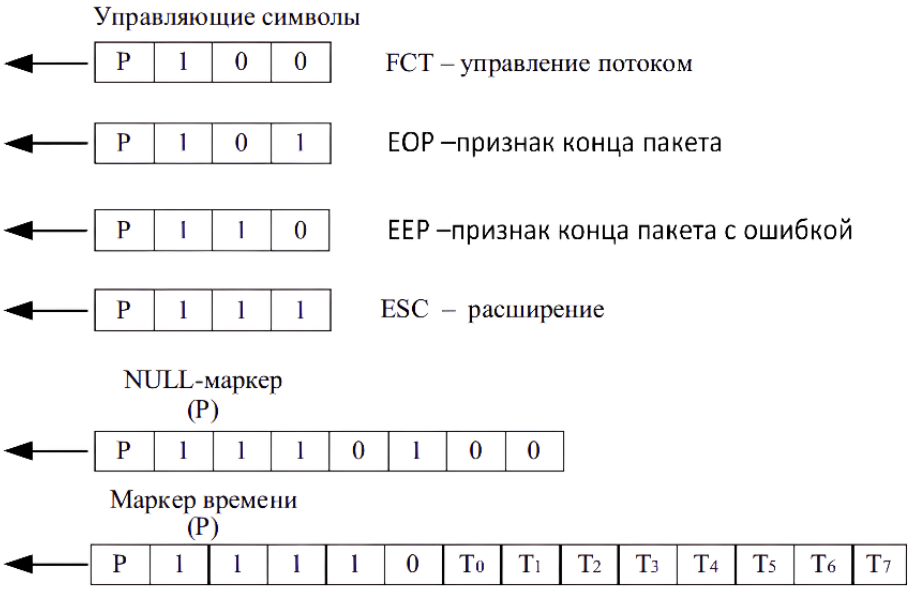


Рисунок 76 – Управляющие символы SpaceWire

Маркер времени состоит из символа расширения ESC и символа данных. Бит контроля по четности P в середине маркера времени равен 1. Этот маркер используется для поддержания единого системного времени в сети.

Бит контроля по четности добавляется в символы данных или управления для определения ошибок передачи по каналу (см. рисунок 77).

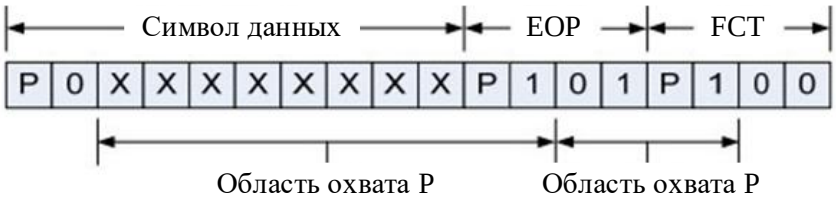


Рисунок 77 – Область, контролируемая битом паритета

Бит контроля по четности охватывает восемь бит данных предыдущего символа данных или два бита управляющего кода предыдущего символа управления и флаг управления текущего символа, как показано на рисунке 77. Бит P используется для проверки нечетности, то есть количество единичных битов в области охвата бита контроля по четности (включая сам бит P) должно быть нечетным.

7.15.5 Диаграмма состояний

Инициализация канала SpaceWire представляет собой сложный процесс синхронного запуска обеих его сторон. При запуске стороны обмениваются служебными символами, информируя друг друга о степени готовности к нормальной работе. При этом используется алгоритм «обмена подтверждениями» и прохождение через определенные состояния интерфейса канала. При запуске интерфейс канала последовательно проходит через следующие состояния (рисунок 83):

- ErrorReset (Сброс);
- ErrorWait (Ожидание);
- Ready (Готовность);
- Started (Запуск);
- Connecting (Соединение);
- Run (Рабочий режим).

Для того чтобы контролировать состояния канала интерфейса, используются биты STATE регистра STATUS. Также возможно формирование прерывания при изменении статуса соединения установкой бита INTENLINK в регистре CONTROL.

В случае возникновения ошибок из любого состояния интерфейс канала сбрасывается и переходит в состояние ErrorReset (Сброс). Ошибки делятся на четыре типа:

- прием символов, не ожидаемых в данном состоянии;
- ошибки таймаута;
- ошибки приема (ошибки расширения символов, ошибки контроля четности, ошибки рассоединения);
- ошибки кредитования.

Для того чтобы контролировать эти ошибки, используются флаги ошибок PERR, DSCERR, ESCERR, CREDERR в регистре STATUS. Также возможно формирование прерывания по любому из вышеперечисленных флагов установкой бита INTENERR в регистре CONTROL.

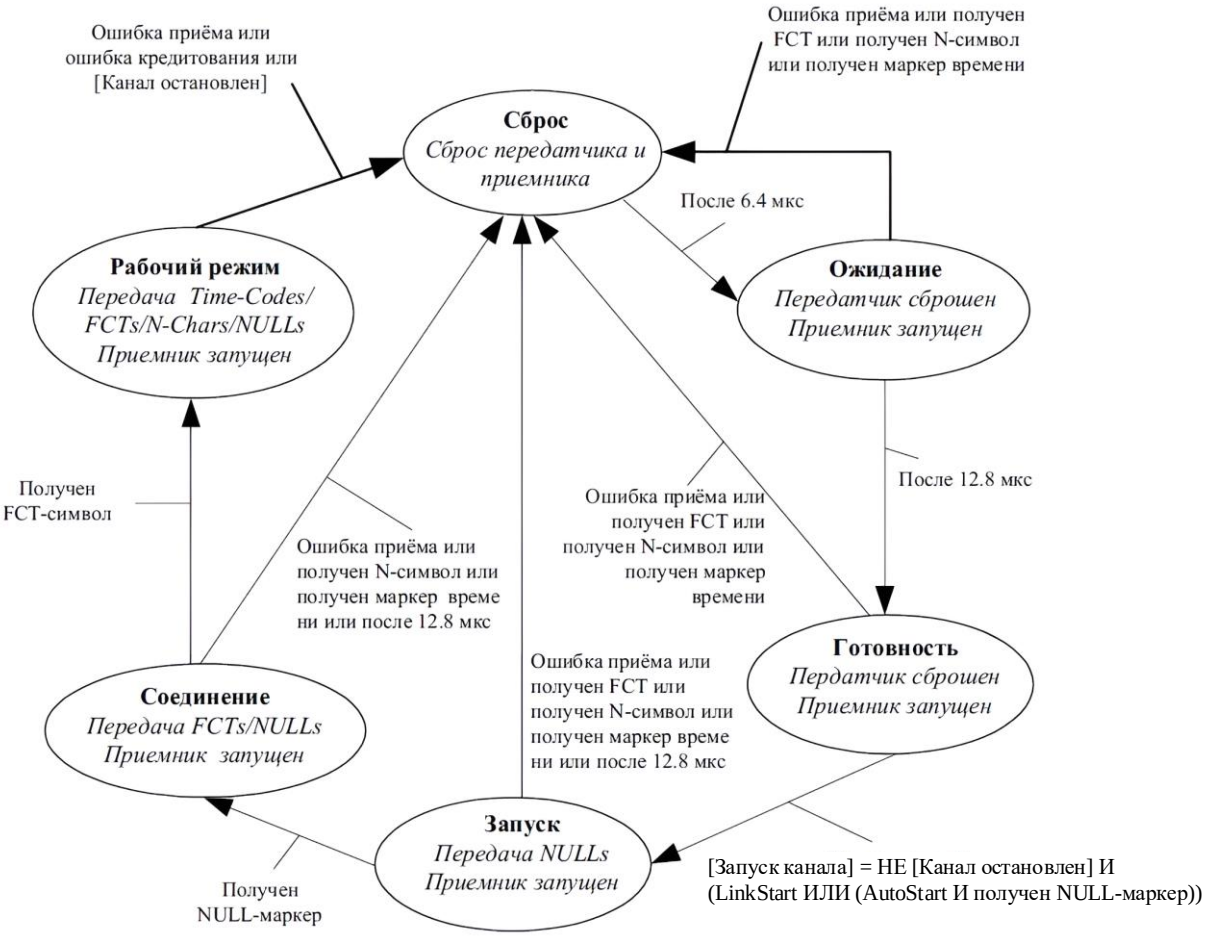


Рисунок 78 – Диаграмма состояния интерфейса канала SpaceWire

Определение ***ошибки рассоединения*** возможно только после приема первого бита после состояния ErrorReset (Сброс). Рассоединение определяется в случае, если на линиях данных и строба не было изменения сигнала в течение 850 нс. Для измерения этого интервала времени необходимо в регистре PAUSE с помощью бит DSCTIME задать количество тактов частоты синхронизации, необходимое на интервал работы счетчика 850 нс.

Ошибка кредитования возникает в случае приема символов данных и отсутствия свободного места в буфере приемника. Первоначально интерфейс канала находится в состоянии ErrorReset (Сброса). Приемник и передатчик канала сбрасываются. Через 6,4 мкс интерфейс канала переходит в состояние ErrorWait (Ожидания). Запускается приемник канала. Через 12,8 мкс интерфейс канала переходит в состояние Ready (Готовности). В состоянии Ready (Готовности) интерфейс канала ожидает команду от конечного узла на запуск канала. Для разрешения перехода из состояния Ready (Готовности) в состояние Started (Запуска) после получения команды на запуск необходимо установить бит LINKSTART или AUTOSTART в регистре CONTROL. Установленный бит LINKSTART разрешает переход в состояние Started (Запуска). При установленном бите AUTOSTART для перехода в состояние Started (Запуска) интерфейс канала ожидает получение NULL-маркера, при приеме которого устанавливается признак «получен NULL-маркер».

После перехода в состояние Started (Запуска) запускается передатчик канала, который начинает передавать NULL-маркеры. В состоянии Started (Запуска) интерфейс канала находится до тех пор, пока не будет установлен признак «получен NULL-маркер» (этот признак устанавливается в состояниях ErrorWait (Ожидание), Ready (Готовность) и Started (Запуск) при приеме NULL-маркера). Если в течение 12,8 мкс признак «получен NULL-маркер» не был установлен, канал сбрасывается. Также канал сбрасывается при наличии ошибки приема, получении символа FCT или N или маркера времени. При установленном бите AUTOSTART в регистре CONTROL признак «получен NULL-маркер» устанавливается при получении NULL-маркера интерфейсом канала в состоянии Ready (Запуска), поэтому сброс канала из Started (Запуска) в ErrorReset (Сброса) по таймеру через 12,8 мкс не происходит.

При установленном признаке приема NULL-маркера интерфейс канала переходит в состояние Connecting (Соединения) и начинает передавать символы FCT. Если в течение 12,8 мкс символ FCT не был получен, канал также сбрасывается.

При приеме символа FCT интерфейс канала переходит в состояние Run (Рабочего режима) и готов нормально функционировать.

Следует отметить, что до перехода в состояние Run интерфейс работает на скорости 10 Мбит/с и использует биты DIVCNTDEF регистра DIV для ее обеспечения. После перехода в состояние Run интерфейс работает на скорости, установленной с помощью бит DIVCNT регистра DIV.

Определение ошибок контроля по четности и расширения символов управления, получение информационных символов и символа FCT разрешено только после приема первого NULL-маркера. Определение первого NULL-маркера включает в себя проверку трех бит контроля по четности (см. рисунок 84).

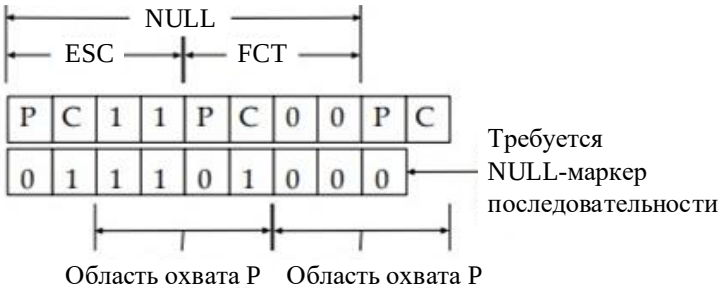


Рисунок 79 – Декодирование NULL-последовательности

Проверяются:

- бит P, охватывающий флаг управления символа ESC;
- бит P, охватывающий символ ESC;
- бит P, охватывающий символ FCT.

Следовательно, NULL-маркер может быть принят, и условие «принят NULL-маркер» установлено, если принята последовательность бит 011101000, как показано на рисунке 77. При включении канала символ, следующий за первым NULL-маркером – управляющий символ (NULL-маркер или символ FCT). Следовательно, последний бит контроля по четности NULL-маркера равен 0. Если

ошибка контроля по четности произошла в первом NULL-маркере, то NULL-маркер не будет получен, и признак «принят NULL-маркер» не будет установлен.

7.15.6 Инициализация

Для работы с контроллером SpaceWire должны быть заданы параметры тактового сигнала.

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 21 для SpaceWire регистра PER0_CLK). В регистре SPW0_CLK установить бит EN_CLK, разрешающий выдачу частоты SPWCLK для контроллера SpaceWire, задать коэффициент деления DIV тактовой частоты MAXCLK для задания SPWCLK.

Далее в регистр DIV записать делители SPWCLK для задания скорости работы контроллера до перехода в состояние Run = 10 Мбит/с (DIVCNTDEF), а также скорости передачи данных (DIVCNT).

Для измерения интервала времени 850 нс необходимо в регистре PAUSE битами DSCTIME задать количество тактов частоты синхронизации необходимое на интервал работы счетчика 850 нс.

Для измерения интервалов времени 6,4 мкс, а также 12,8 мкс необходимо в регистре PAUSE с помощью бит RESTIME задать количество тактов частоты синхронизации необходимое на интервал работы счетчика 6,4 мкс.

При работе через встроенный приемопередатчик необходимо установить биты EN_BNG, OUT_EN и EN_PHY в регистре PHY_CNTR. При этом рекомендуется устанавливать бит OUT_EN регистра PHY_CNTR через 50 мкс после установки бита EN_PHY в этом регистре. Если биты OUT_EN и EN_PHY установлены в состояние логической «1» одновременно, то в течение первых 50 мкс достоверность передаваемых и принимаемых данных не гарантируется.

В регистр CONTROL записать биты разрешения перехода из состояния Ready в Started (LINKSTART или AUTOSTART), а также разрешить прерывания по необходимым событиям. Включить интерфейс битом SPWEN.

После прохождения процедуры установления связи между устройствами и перехода контроллера в режим нормального функционирования Run можно принимать и передавать данные.

7.15.7 Run (Рабочий режим)

В рабочем режиме добавляется возможность приема/передачи двух типов символов, передача которых недоступна и ошибочна в других состояниях, а именно Time-Codes (маркеры времени) и N-Chars (символы данных).

В стандарте SpaceWire предусмотрен механизм обмена метками времени между узлами. Источник меток времени в сети SpaceWire должен быть один. Посылка времени приведена на рисунке 76 и на уровне обмена представляет собой слово данных, следующее за командой ESC. Посылка времени имеет наивысший приоритет и отправляется сразу же после окончания отправки текущей команды. Для отправки

посылок времени в контроллере предусмотрен регистр TIME, в котором биты TIMEIN содержат маркер времени для передачи. При установке бита TICKIN регистра TIME инициируется процесс передачи содержимого TIMEIN в канал. После чего TICKIN автоматически сбрасывается в ноль. Использование механизма обмена времени в соответствии с протоколом предусматривает наличие 6-ти разрядного счетчика времени (биты TIMEOUT в регистре TIME) в каждом из узлов сети. Оставшиеся два бита из восьми являются командными флагами и зарезервированы для будущего использования (биты CTRLOUT в регистре TIME). В узле источнике времени оба контрольных флага должны быть установлены в 0 (биты CTRLIN в регистре TIME). При получении маркера времени в регистре STATUS устанавливается флаг GOTTIME. Может быть сгенерировано прерывание в момент установки этого флага разрешением маски INTENTIME в регистре CONTROL.

Каждому элементу сети SpaceWire (терминальным узлам и маршрутизаторам) рекомендуется иметь буфер данных (не менее 56 байт). В нашем случае размер буфера данных 64 байта. Узел (маршрутизатор) является ответственным за управление буфером данных. Буфер может быть реализован несколькими методами. Наиболее подходящие для стандарта SpaceWire – FIFO-буферы. Данные из FIFO-буфера передающей стороны канала могут быть переданы только в случае получения FCT-символа (одного или нескольких) с противоположной стороны. Принимающая сторона принимает данные до тех пор, пока есть свободное место в ее FIFO-буфере. Как только в FIFO-буфере принимающей стороны освобождается место для восьми информационных символов, информация об этом должна немедленно попасть передающей стороне канала в виде символа FCT. Каждый FCT-символ символизирует о восьми свободных байтах с противоположной стороны. Непрерывно может быть передано не более семи символов FCT подряд. О состоянии FIFO сигнализируют флаги регистра STATUS, а именно TXF и TXE для FIFO передачи, RXF и RXFNE для FIFO приема. Может быть сгенерировано прерывания по событию установки любого из вышеописанных флагов, если они разрешены масками прерываний INTETXF, INTETXE, INTERXF, INTENRXNE в регистре CONTROL.

Для передачи данных в/из памяти конечного узла может использоваться прямой доступ в память (DMA – Direct Memory Address). Если передача DMA уже подготовлена, возможна передача сразу нескольких FCT-символов для повышения скорости передачи данных. Для разрешения этого режима работы используются биты DMARXEN и DMATXEN регистра CONTROL. Для чтения принятых данных и записи передаваемых данных используются регистры FIFORX и FIFOTX соответственно.

Передача и прием FCT-символов контроллером происходит автоматически независимо от режима работы с DMA или без него. Контроллер имеет счетчик числа символов, разрешенных к передаче (счетчик передачи), и счетчик числа символов, для приема которых есть свободное пространство в буфере приемника (счетчик приема).

Так как символы N-Chars образуют пакеты, ограниченные символами EOP либо EEP, требуется хранить информацию о длине пакетов и типах конца пакетов. Для этих целей служат регистры дескрипторов приема RX_DESC0-RXDESC15 и дескрипторов

передачи TXDESC0-TXDESC15. Старший 31 разряд регистров дескрипторов является признаком того, что дескриптор содержит действительные данные. В случае если 31 разряд любого из регистров дескрипторов равен нулю, то этот регистр не содержит достоверной информации. Данные пакета, соответствующие дескрипторам, поступают в FIFO, и так как 64 байта недостаточно для хранения всего пакета, то данные пакета необходимо переносить в/из внешней памяти SRAM большего размера посредством DMA или регистрового чтения FIFORX и FIFOTX. В случае завершения приема или передачи пакета происходит установка флагов TXDESC или RXDESC регистра STATUS. Может быть сгенерировано прерывания по событию установки любого из вышеописанных флагов, если они разрешены масками прерываний INTENRXDES или INTENTXDES в регистре CONTROL.

Имеется возможность формирования статистики принятых пакетов нулевой и ненулевой длины посредством регистров счетчиков принятых пакетов CNTRX0_PACK и CNTRX_PACK соответственно.

7.15.8 Передача сообщений

Для передачи пакета необходимо:

1 Определить номер по порядку NUM_TXDESC дескриптора TXDESC[i], определяющего передаваемый пакет;

2 В TXDESC[i] задать тип передаваемого пакета EEP или EOP битами 29…30;

3 Задать размер битами 0...24;

4 Инициировать передачу установкой бита 31;

5 Если пакет ненулевой заполнить данными FIFOTX.

7.15.9 Прием сообщений

Для приема пакета необходимо:

1 Вычислить дескриптор RXDESC[i], определяющий принятый пакет путем проверки бита 31;

2 Определить тип принимаемого пакета EEP или EOP, определяемого битами 29...30;

3 Определить длину пакета, определяемого битами 0...24;

4 Если пакет ненулевой вычитать данные из FIFORX.

7.15.10 Передача и прием маркера времени

Значение принятого маркера времени определяется битами 8...13 регистра TIME, значение передаваемого – битами 0...5. Для передачи маркера необходимо взвести бит 16, прием осуществляется по флагу GOTTIME регистра STATUS.

7.15.11 Описание регистров

Таблица 289 – Описание регистров контроллера интерфейса SpaceWire

Базовый адрес	Название	Описание
0x4008_8000	SPW	Контроллер интерфейса SpaceWire
Смещение		
0x0000_0000	CONTROL	Регистр управление контроллером
0x0000_0004	STATUS	Регистр состояния контроллера
0x0000_0008	DIV	Регистр делителей частоты контроллера
0x0000_000C	TIME	Регистр маркеров времени
0x0000_0010	PAUSE	Регистр задержек установления соединения
0x0000_0014	FIFORX	FIFO данных приемника
0x0000_0018	FIFOTX	FIFO данных передатчика
0x0000_001C	CNTRX_PACK	Регистр счетчика принятых пакетов ненулевой длины
0x0000_0020	CNTRX0_PACK	Регистр счетчика принятых пакетов нулевой длины
0x0000_0024	NUM_TXDESC	Регистр номера дескриптора передатчика, планируемого к обработке
0x0000_0028	NUM_RXDESC	Регистр номера дескриптора приемника, планируемого к заполнению
0x0000_002C	PHY_CNTR	Регистр управления аналоговым приемопередатчиком
0x0000_0030	FSC	Регистр задания бесперебойной работы приемника контроллера SpaceWire
0x0000_0040-0x0000_007C	RXDESC0-SPW_RXDESC15	Память дескрипторов приемника
0x0000_0080-0x0000_00BC	TXDESC0-SPW_TXDESC15	Память дескрипторов передатчика

7.15.11.1 **CONTROL**

Base ADDR=		0x4008_8000			Offset=		0x0000_0000								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERXF34	INTEN TXDES	INTEN RXDES	INTE TXF	INTE TXE	INTE RXF	INTEN RXNE	DMARXEN	DMATXEN	INTEN LINK	INTEN ERR	INTEN TIME	AUTO START	LINK START	LINK DIS	SPWEN

Таблица 290 – Описание бит регистра CONTROL

Бит	Имя	Значение	Описание
31...16			Зарезервировано
15	INTERXF34		Разрешение прерывания при заполнении 3/4 FIFO приемника: 0 – прерывание запрещено; 1 – прерывание разрешено
14	INTENTXDES		Разрешение прерывания по завершении передачи пакета данных с признаком конца пакета: 0 – прерывание запрещено; 1 – прерывание разрешено
13	INTENRXDES		Разрешение прерывания по завершении приема пакета данных с признаком конца пакета: 0 – прерывание запрещено; 1 – прерывание разрешено
12	INTETXF		Разрешение прерывания при полном заполнении FIFO передатчика: 0 – прерывание запрещено; 1 – прерывание разрешено
11	INTETXE		Разрешение прерывания при отсутствии данных в FIFO передатчика: 0 – прерывание запрещено; 1 – прерывание разрешено
10	INTERXF		Разрешение прерывания при полном заполнении FIFO приемника: 0 – прерывание запрещено; 1 – прерывание разрешено
9	INTENRXNE		Разрешение прерывание при наличии в FIFO приемника хотя бы одного 32-разрядного слова данных: 0 – прерывание запрещено; 1 – прерывание разрешено
8	DMARXEN		Разрешение запроса DMA при заполнении FIFO приемника: 0 –запрос запрещен; 1 – запрос разрешен
7	DMATXEN		Разрешение запроса DMA при опустошении FIFO передатчика: 0 –запрос запрещен; 1 – запрос разрешен
6	INTENLINK		Разрешение прерывания при изменении статуса соединения (соединение установлено/соединение разорвано): 0 – прерывание запрещено; 1 – прерывание разрешено

Бит	Имя	Значение	Описание
5	INTENERR		Разрешение прерывания при возникновении одной из ошибок: DSCERR, PERR, ESCERR или CREDERR: 0 – прерывание запрещено; 1 – прерывание разрешено
4	INTENTIME		Разрешение прерывания при приеме маркера времени: 0 – прерывание запрещено; 1 – прерывание разрешено
3	AUTOSTART		Управляет параметром AutoStart стандарта ECSS-E-ST-50-12C. Разрешение перехода блока Linkinterface из состояния Ready в состояние Started с ожиданием символа NULL (согласно Figure 8-2 стандарта ECSS-E-ST-50-12C): 0 – переход запрещен; 1 – переход разрешен (с ожиданием символа NULL). В отличие от LinkStart, при AutoStart не происходит переход по таймеру из состояния Started в состояние ErrorReset, так как признак «получен NULL-маркер» устанавливается при приеме NULL-маркера интерфейсом канала в состоянии Ready (Готовности)
2	LINKSTART		Управляет параметром LinkStart стандарта ECSS-E-ST-50-12C. Разрешение перехода блока Linkinterface из состояния Ready в состояние Started (согласно Figure 8-2 стандарта ECSS-E-ST-50-12C): 0 – переход запрещен; 1 – переход разрешен
1	LINKDIS		Перевод блока Linkinterface из состояния Run в состояние ErrorReset (LinkDisable): 0 – нет перевода в состояние ErrorReset. Разрешение выполнение команд LinkStart, AutoStart; 1 - перевод в состояние ErrorReset
0	SPWEN		Разрешение работы интерфейса SpaceWire: 1 – работа интерфейса разрешена; 0 – интерфейс находится в состоянии сброса. Установка бита в единицу разрешается только после инициализации регистров SPW_DIV и SPW_PAUSE.

7.15.11.2 **STATUS**

Base ADDR=		0x4008_8000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

-	RXF34	TXDESC	RXDESC	TXF	TXE	RXF	RXNE	GOT TIME	CRED ERR	ESC ERR	DSC ERR	PERR	STATE

Таблица 291 – Описание бит регистра STATUS

Бит	Имя	Значение	Описание
31…15			Зарезервировано
14	RXF34		Флаг заполненности 3/4 FIFO приемника: 0 – FIFO не полно; 1– FIFO полно
13	TXDESC		Флаг завершения передачи пакета данных с признаком конца пакета: 0 – передача пакета не завершена; 1 – передача пакета завершена. При выполнении операции записи 1 в этот разряд он устанавливается в 0
12	RXDESC		Флаг завершения приема пакета данных с признаком конца пакета: 0 – прием пакета не завершен; 1 – прием пакета завершен. При выполнении операции записи 1 в этот разряд он устанавливается в 0
11	TXF		Флаг полной заполненности FIFO передатчика: 0 – FIFO не полно; 1 – FIFO полно
10	TXE		Флаг отсутствия данных в FIFO передатчика: 0 – FIFO не пусто; 1 – FIFO пусто
9	RXF		Флаг полной заполненности FIFO приемника: 0 – FIFO не полно; 1 – FIFO полно
8	RXNE		Флаг наличия данных в FIFO приемника: 0– FIFO пусто; 1 – FIFO содержит данные
7	GOTTIME		Признак приема маркера времени из канала связи SpaceWire: 1 – принят маркер времени; 0 – нет принятого маркера времени. При выполнении операции записи 1 в этот разряд он устанавливается в 0
6	CREDERR		Признак ошибки кредитования: 1 – произошла ошибка; 0 – нет ошибки. При выполнении операции записи 1 в этот разряд он устанавливается в 0

Бит	Имя	Значение	Описание
5	ESCERR		Признак ошибки в ESC последовательности (получен неверный символ (пп. 7.3 и 8.9.2.3 стандарта ECSS-E-ST-50-12C): 1 – произошла ошибка; 0 – нет ошибки. При выполнении операции записи 1 в этот разряд он устанавливается в 0
4	DSCERR		Признак рассоединения с каналом связи SpaceWire: 1 – произошло рассоединение; 0 – нет рассоединения. При выполнении операции записи 1 в этот разряд он устанавливается в 0
3	PERR		Признак ошибки четности: 1 – произошла ошибка; 0 – нет ошибки. При выполнении операции записи 1 в этот разряд он устанавливается в 0
2…0	STATE[2:0]		Состояние блока Linkinterface (согласно Figure 8-2стандарта ECSS-E-ST-50-12C): 000 – Error Reset (исходное состояние); 001 – Error Wait; 010 – Ready; 011 – Started; 100 – Connecting; 101 – Run

Примечания

1. Для чтения данных из FIFORX рекомендуется использовать событие установки флага RXF34 вместо RXF.

2. Для организации надежного сброса статусных флагов прерываний необходимо учитывать требования, изложенные в подразделе 9.3

7.15.11.3 **DIV**

Base ADDR=		0x4008_8000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVCNTDEF								DIVCNT							

Таблица 292 – Описание бит регистра DIV

Бит	Имя	Значение	Описание
31…16			Зарезервировано
15…8	DIVCNTDEF		Делитель для задания скорости передачи данных в канал связи SpaceWire равной 10 Мбит/с: скорость передачи $= SPWCLK/DIVCNTDEF$, DIVCNTDEF не равно 0, где $SPWCLK$ – частота тактирования передатчика. Частота $SPWCLK$ должна быть равна частоте контроллера либо быть больше ее в два раза. До перехода контроллера в состояние Run скорость передачи данных 10 Мбит/с. В состоянии Run скорость передачи данных определяется делителем DIVCNT
7…0	DIVCNT		Делитель для задания скорости передачи данных в канал связи SpaceWire скорость передачи $= SPWCLK/DIVCNT$, DIVCNT не равно 0, где $SPWCLK$- частота тактирования передатчика. Частота $SPWCLK$ должна быть равна частоте контроллера либо быть больше ее в два раза

7.15.11.4 **TIME**

Base ADDR=		0x4008_8000			Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															TICKIN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRLOUT		TIMEOUT						CTRLIN		TIMEIN					

Таблица 293 – Описание бит регистра TIME

Бит	Имя	Значение	Описание
31…17	-		Зарезервировано
16	TICKIN		При установке этого бита инициируется процесс передачи маркера времени с контрольными разрядами. После этого TICKIN автоматически сбрасывается в ноль
15…14	CTRLOUT		Значение контрольных бит, принятых из сети последними
13…8	TIMEOUT		Значение маркера времени, принятого из сети последним
7…6	CTRLIN		Значение контрольных бит для передачи при установке бита TICKIN
5…0	TIMEIN		Значение маркера времени для передачи при установке бита TICKIN

7.15.11.5 **PAUSE**

Base ADDR=		0x4008_8000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-													RESTIME		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTIME								DSCTIME							

Таблица 294 – Описание бит регистра PAUSE

Бит	Имя	Значение	Описание
31…19			Зарезервировано
18…8	RESTIME		Количество тактов частоты процессорного ядра необходимое на интервал работы счетчика 6.4 мкс
7…0	DSCTIME		Количество тактов частоты процессорного ядра необходимое на интервал работы счетчика 850 нс

7.15.11.6 **FIFORX**

Base ADDR=		0x4008_8000			Offset=		0x0000_0014								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Таблица 295 – Описание бит регистра FIFORX

Бит	Имя	Значение	Описание
31…0	DATA		В блоке приема из байтов данных формируются слова разрядности 32. При формировании слов первый поступивший байт размещается в разрядах 7:0, второй – в разрядах 15:8, третий – в разрядах 23:16 и четвертый – в разрядах 31:24. Распределение символов данных по разрядам слова данных производится по счетчику байт. В случае получения кода конца пакета данные выравниваются по границам 32-разрядных слов путем добавления «лишних» байт. FIFO размером 16 слов

7.15.11.7 **FIFOTX**

Base ADDR=		0x4008_8000			Offset=		0x0000_0018								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Таблица 296 – Описание бит регистра FIFOTX

Бит	Имя	Значение	Описание
31…0	DATA		Блок передачи разбивает слова на отдельные байты. При этом из последовательности байтов в соответствии с информацией, содержащейся в дескрипторе, удаляются «лишние» байты – байты, добавленные для выравнивания пакетов по границам слов, и вставляются символы концов пакетов EOP или EEP. Блок передачи вначале передает в Linkinterface байт данных, находящийся в разрядах 7:0 слова, затем байт, находящийся в разрядах 15:8, затем байт, находящийся в разрядах 23:15, затем байт из разрядов 31:24 тридцатидвухразрядного слова. FIFO размером 16 слов

7.15.11.8 **CNTRX_PACK**

Base ADDR=		0x4008_8000			Offset=		0x0000_001C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOTNULLCNT															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTNULLCNT															

Таблица 297 – Описание бит регистра CNTRX_PACK

Бит	Имя	Значение	Описание
31…0	NOTNULLCNT		Содержимое 32-разрядного регистра SPW_CNTRX_PACK увеличивается на 1 каждый раз, когда из канала связи SpaceWire принимается символ конца пакета EOP, если ему предшествовал хотя бы один символ данных, что означает принятие пакета ненулевой длины. При записи (любых данных), значение регистра обнуляется

7.15.11.9 **CNTRX0_PACK**

Base ADDR=		0x4008_8000			Offset=		0x0000_0020								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NULLCNT															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NULLCNT															

Таблица 298 – Описание бит регистра CNTRX0_PACK

Бит	Имя	Значение	Описание
31…0	NULLCNT		Содержимое 32-разрядного регистра SPW_CNTRX0_PACK увеличивается на 1 при приеме символа EOP вслед за символом EOP, что эквивалентно принятию пакета нулевой длины. При записи (любых данных), значение регистра обнуляется

7.15.11.10 **NUM_TXDESC**

Base ADDR=		0x4008_8000			Offset=		0x0000_0024								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-												NUM_TXDESC			

Таблица 299 – Описание бит регистра NUM_TXDESC

Бит	Имя	Значение	Описание
31…4	-		Зарезервировано
3…0	NUM_TXDESC		Дескрипторы передатчика SPW_TXDESC0-SPW_TXDESC15 вычитываются из памяти, начиная с номера 0 и заканчивая номером 15, после этого процесс повторяется. Если дескриптор в 31-ом разряде не содержит единицу, то после передачи пакета, определяемого предыдущим дескриптором, процесс увеличения номера дескриптора останавливается. Процесс возобновляется после установки в единицу 31-го разряда дескриптора с номером, содержащимся в регистре SPW_NUM_TXDESC. После инициирования передачи пакета, 31-й разряд дескриптора автоматически

			сбрасывается в ноль, а значение регистра SPW_NUM_TXDESC увеличивается на единицу

7.15.11.11 NUM_RXDESC

Base ADDR=		0x4008_8000			Offset=		0x0000_0028								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-												NUM_RXDESC			

Таблица 300 – Описание бит регистра NUM_RXDESC

Бит	Имя	Значение	Описание
31…4	-		Зарезервировано
3…0	NUM_RXDESC		Дескрипторы приемника SPW_RXDESC0-SPW_RXDESC15 заполняются в памяти, начиная с номера 0 и заканчивая номером 15, после этого процесс повторяется

7.15.11.12 PHY_CNTR

Base ADDR=			0x4008_8000			Offset=		0x0000_002C							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-											TST[2]	TST[1]	TST[0]	TRIMR[7:6]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIMR [5:0]						EN_ BNG	TRIM [2:0]			SELR	OUT_ EN	-	EN_ PHY	IR_ UP	IR_ DOWN

Таблица 301 – Описание бит регистра PHY_CNTR

Бит	Имя	Значение	Описание
31...21	-	0	Зарезервировано
20...18	TST[2:0]	000	Вход задания тестового режима: TST[0] = 0 – выключен режим петли приемника на передатчик; TST[0] = 1 – включен режим петли приемника на передатчик; TST[1] – зарезервировано; TST[2] = 0 – выключен гистерезис приемника; TST[2] = 1 – включен гистерезис приемника
17…10	TRIMR[7:0]	0	Зарезервировано

Бит	Имя	Значение	Описание
9	EN_BNG	0	Зарезервировано
8…6	TRIM[2:0]	0	Вход цифрового тримминга источника опорного напряжения. Типовое значение 100 (1,2 В). 000 – максимальное напряжение (1,25 В); 111 – минимальное напряжение (1,15 В)
5	SELR	0	Зарезервировано
4	OUT_EN	0	Разрешение передачи приёмопередатчика: 0 – выключен; 1 – включен
3	-		Зарезервировано
2	EN_PHY	0	Разрешение работы приёмопередатчика 0 – выключен; 1 – включен
1	IR_UP	0	Увеличение тока: 0 – по умолчанию; 1 – увеличить на 12,5 %
0	IR_DOWN	0	Уменьшение тока: 0 – по умолчанию; 1 – снизить на 12,5 %

7.15.11.13 **FSC**

Base ADDR=			0x4008_8000			Offset=		0x0000_0030							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															FSC_EN

Таблица 302 – Описание бит регистра FSC

Бит	Имя	Значение	Описание
31...1	-	0	Зарезервировано
0	FSC_EN	0	Задание режима бесперебойной работы: 0 – формирование высокого логического уровня на выходе приемника при отключении или обрыве входов приемника; 1 – формирование высокого логического уровня на выходе приемника при отключении или обрыве входов приемника, при выключенном передатчике или если на передатчик не подается питание

7.15.11.14 **RXDESCx**

Base ADDR=		0x4008_8000			Offset=		0x0000_0040 .. 0x0000_007C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXDES															
RW0	R0	R0	R0												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDES															

Таблица 303 – Описание бит регистров RXDESCx

Бит	Имя	Значение	Описание
31…0	RXDES		Дескриптор пакета имеет следующую структуру: – бит 31 – признак заполнения дескриптора действительными данными. До запуска приема, все 31-е биты дескрипторов области приема должны быть обнулены программно, записью единицы; контроллер не обнуляет 31-е биты, а только записывает «1» в 31-е биты принятых дескрипторов; – биты 30…29 – тип конца пакета: 01 – EOP, 10 – EEP; – биты 28…25 – не используются (0000); – биты 24…0 – размер пакета в байтах. Дескриптор пакета формируется в счетчике принятых символов N-Char. При поступлении символа данных N-Char счетчик увеличивается на 1, при поступлении символа конца пакета значение счетчика переписывается в разряды 24…0 текущего дескриптора, а сам счетчик сбрасывается в 0. Всего 16 регистров

7.15.11.15 **TXDESCx**

Base ADDR=		0x4008_8000			Offset=		0x0000_0080 .. 0x0000_00BC								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXDES															
RW0	RW0	R0	R0												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TXDES

Таблица 304 – Описание бит регистров TXDESCx

Бит	Имя	Значение	Описание
31…0	TXDES		Дескриптор пакета имеет следующую структуру: – бит 31 – признак заполнения дескриптора действительными данными. Бит 31 автоматически сбрасывается в ноль после инициирования передачи пакета. – биты 30…29 – тип конца пакета: 01 – EOP, 10 – EEP; – биты 28…25 – не используются (0000); – биты 24…0 – размер пакета в байтах. Данные из SPW_FIFOTX в блок Linkinterface выдаются побайтно. Преобразование 32-разрядных слов в байты осуществляется в блоке преобразования под управлением счетчика переданных байт. В счетчик заносится размер пакета из разрядов 24…0 дескриптора передаваемого пакета. После передачи каждого байта этот счетчик уменьшается на 1. По достижении счетчиком значения 0, в поток передаваемых данных вставляется символ конца пакета EOP или EEP в зависимости от значения разрядов 30...29 дескриптора, а в счетчик заносится размер следующего передаваемого пакета из следующего дескриптора. При установке размера пакета в байтах равного нулю происходит передача пакета нулевой длины. Максимально возможный размер пакета в байтах 0x1FFFFFE. Всего 16 регистров

7.16 Контроллер таймеров общего назначения (TIMER32x)

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

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

– формирование прерываний и запросов DMA по заданным событиям;

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

– генерацию различных форм выходных сигналов.

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

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

Каждый из таймеров позволяет генерировать прерывания и запросы DMA.

7.16.1 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.

7.16.1.1 **Структурная схема**

Структурная схема блока «Таймер» представлена на рисунке 80.

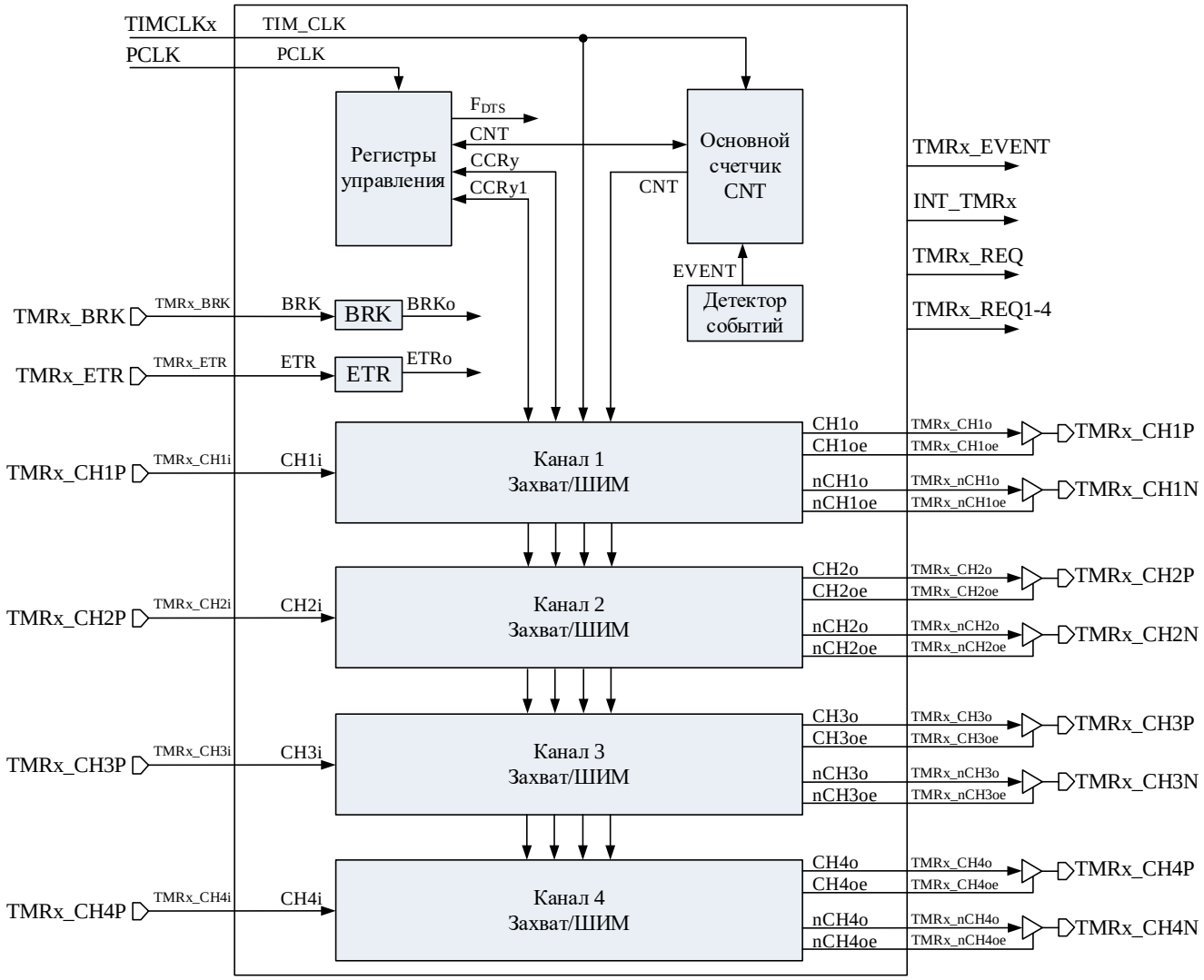


Рисунок 80 – Структурная схема блока «Таймер»

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.

7.16.2 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. пункт 7.16.3 «Источники событий для счета»).

7.16.2.1 **Инициализация тактирования таймера**

Перед началом работы с таймером в первую очередь должны быть включены тактовые сигналы. Параметры задаются в блоке «Контроллер тактовых частот (CLKCNTR)».

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

Для разрешения тактовой частоты PCLK необходимо установить бит тактирования блока в регистре PER0_CLK: бит 23 для таймера 0, бит 24 для таймера 1, бит 25 для таймера 2, бит 26 для таймера 3. Задание тактовой частоты TIM_CLK осуществляется в регистре TIMn_CLK: в поле DIV[15:0] устанавливается коэффициент деления тактовой частоты MAX_CLK для формирования частоты TIM_CLK, разрешение подачи частоты TIM_CLK на блок таймера управляется битом EN_CLK. Тактовая частота TIM_CLK должна быть меньше или равна тактовой частоте PCLK (CPU_CLK).

После подачи тактовых сигналов можно приступать к работе с таймером.

7.16.2.2 **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.

7.16.2.3 **Режимы счета**

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
```

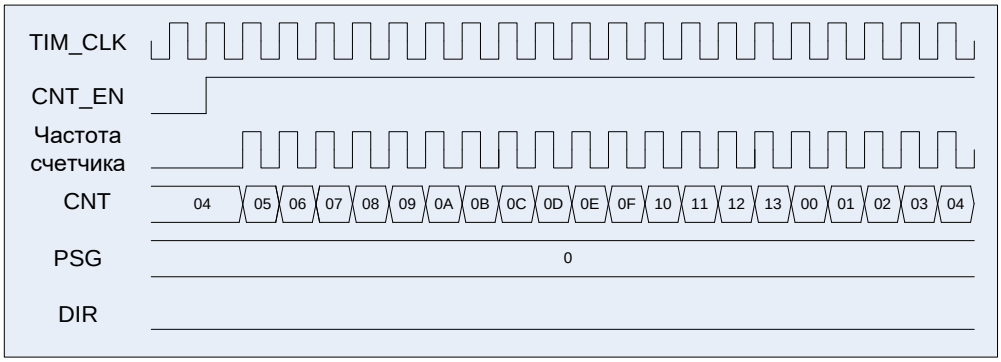


Рисунок 81 – Диаграммы работы таймера, счет прямой от 0 до 0x13, стартовое значение 0x04

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
```

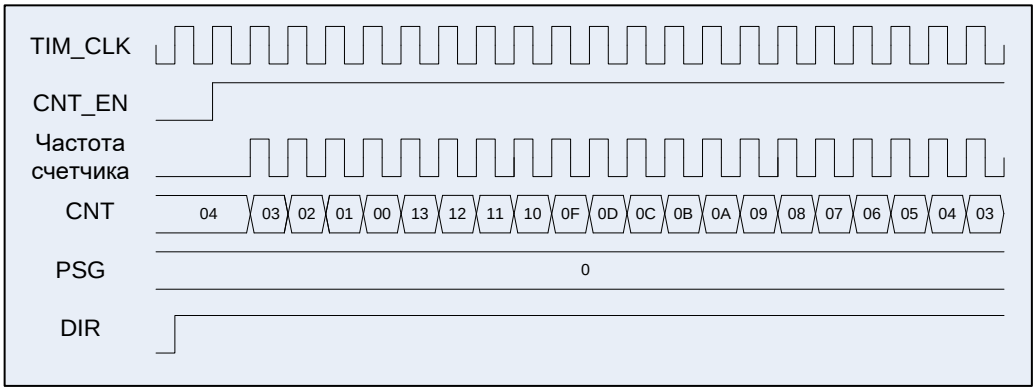


Рисунок 82 – Диаграммы работы таймера, счет обратный от 0x13 до 0, стартовое значение 0x04

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
```

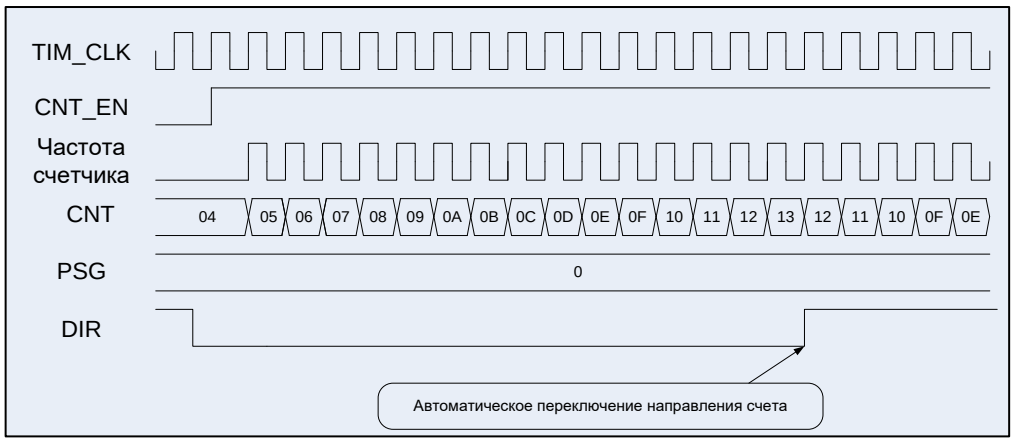


Рисунок 83 – Диаграммы работы таймера, счет двунаправленный, сначала прямой

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
```

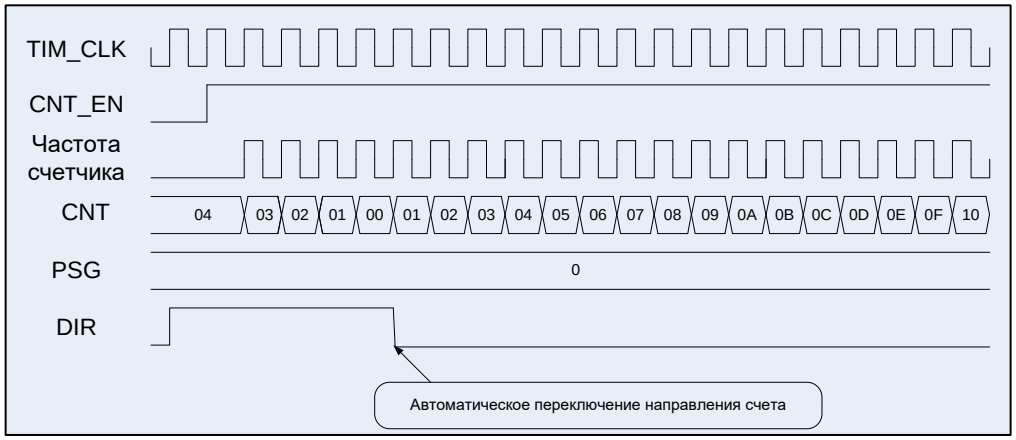


Рисунок 84 – Диаграммы работы таймера, счет двунаправленный, сначала обратный

7.16.2.4 **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 путем прореживания на заданный коэффициент (1, 2, 3 или 4). Настройка частоты F_{DTS} осуществляется в поле FDTS[1:0] регистра CNTRL,.

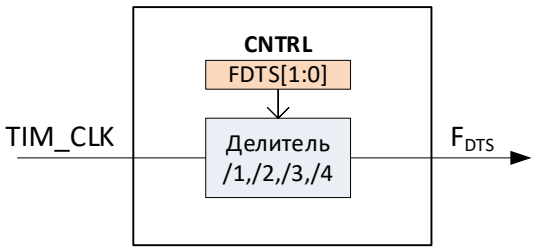


Рисунок 85 – Схема формирования тактовой частоты F_{DTS}

Диаграмма возможных частот F_{DTS} приведена на рисунке 86.

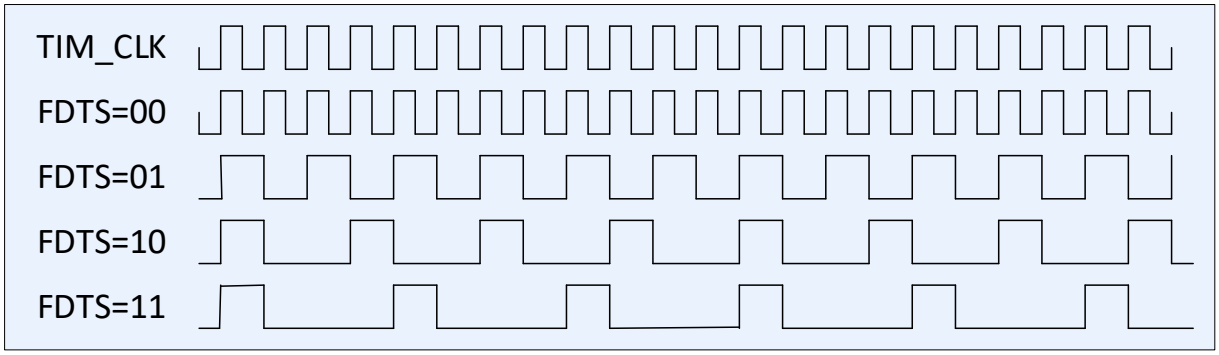


Рисунок 86 – Диаграмма тактовой частоты F_{DTS} в зависимости от значения FDTS[1:0] в регистре CNTRL

7.16.3 Источники событий для счета

Тактирование основного счетчика таймера может осуществляться от следующих источников:

– внутренний тактовый сигнал (TIM_CLKd);

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

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

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

Выбор источника тактирования основного счетчика осуществляется в регистре CNTRL, поле 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.

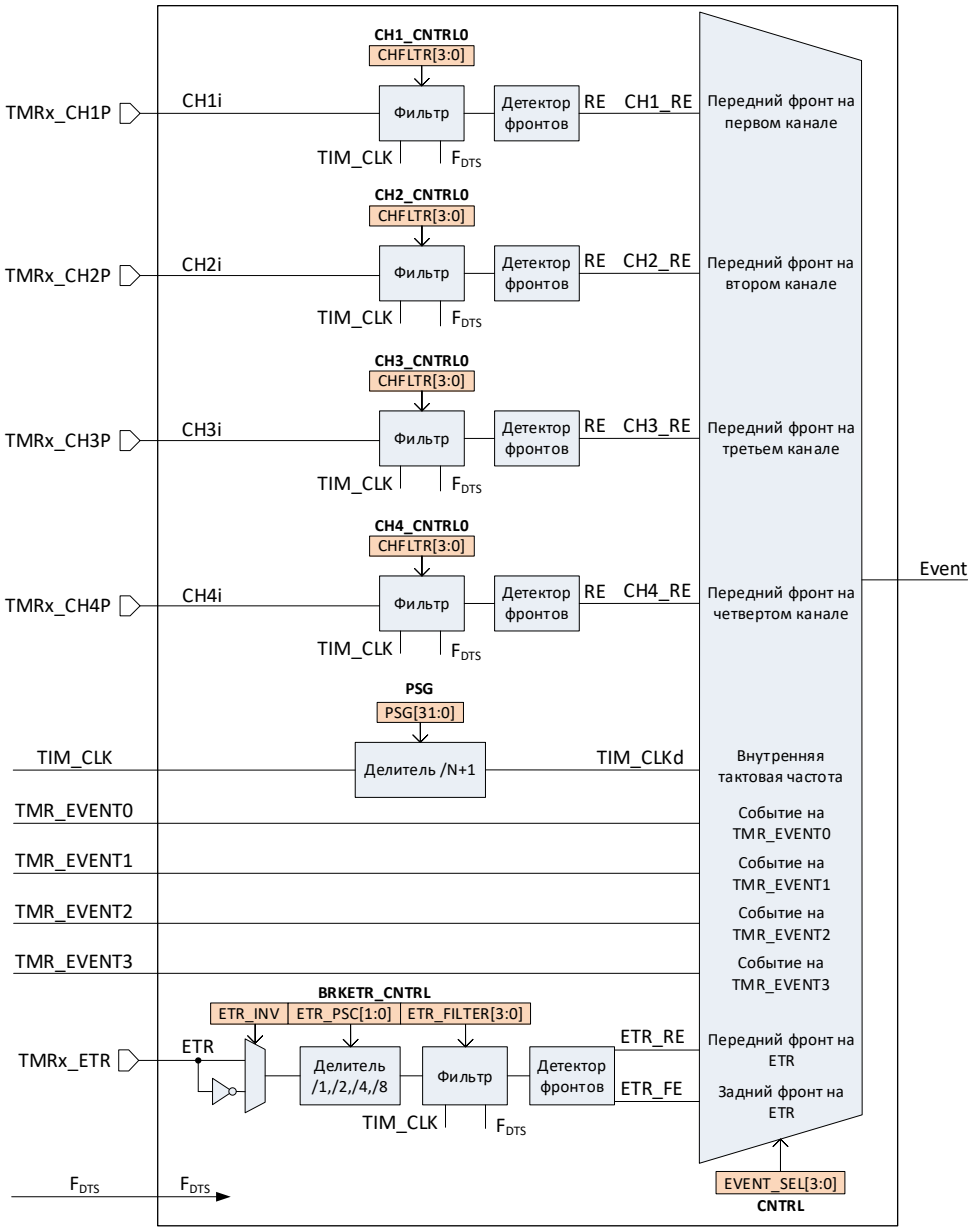


Рисунок 87 – Структурная схема формирования события для счета

7.16.3.1 **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

$$\mathrm{TIM_CLKd} = \frac{\mathrm{TIM_CLK}}{\mathrm{PSG} + 1} \tag{5}$$

Значение регистра PSG можно изменять даже во время работы счетчика, новое значение предделителя вступит в силу сразу после записи. На рисунках 88 и 89 приведены диаграммы работы счетчика при обновлении значения PSG.

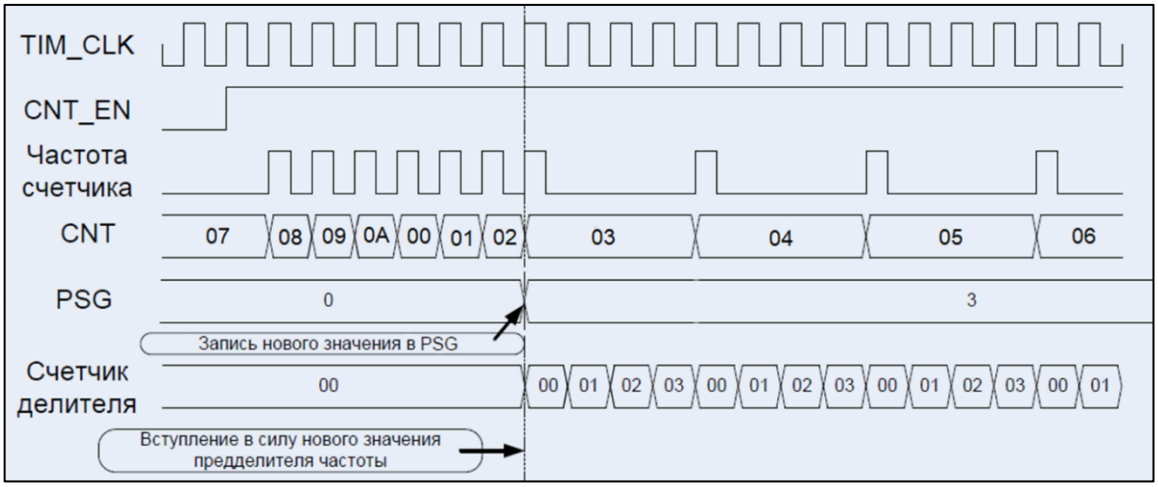


Рисунок 88 – Диаграмма работы счетчика: счет прямой
(CNT_MODE[1:0] = 00, EVENT_SEL[3:0] = 0000, DIR = 0)

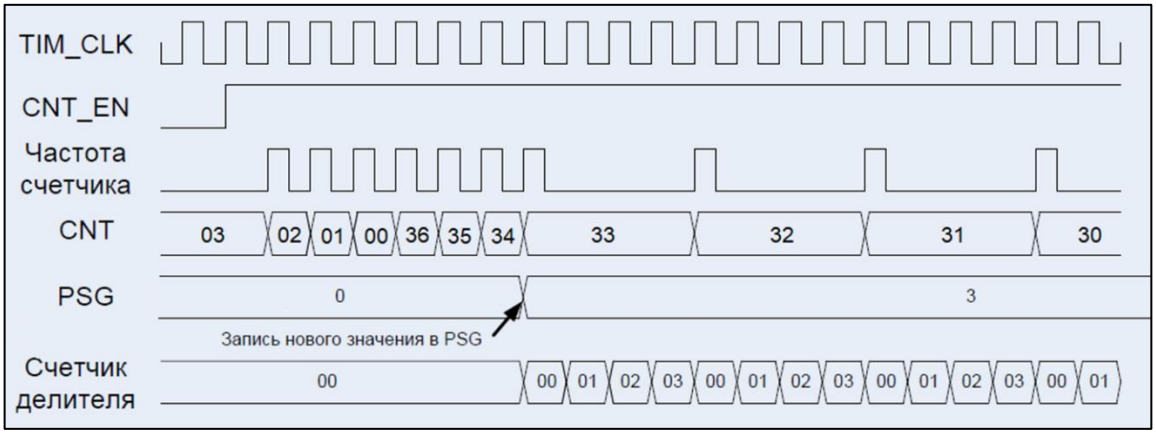


Рисунок 89 – Диаграмма работы счетчика: счет обратный
(CNT_MODE[1:0] = 00, EVENT_SEL[3:0] = 0000, DIR = 1)

7.16.3.2 **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. Подключение выходов запуска TMRx_EVENT к входам других таймеров приведено на рисунке X11.

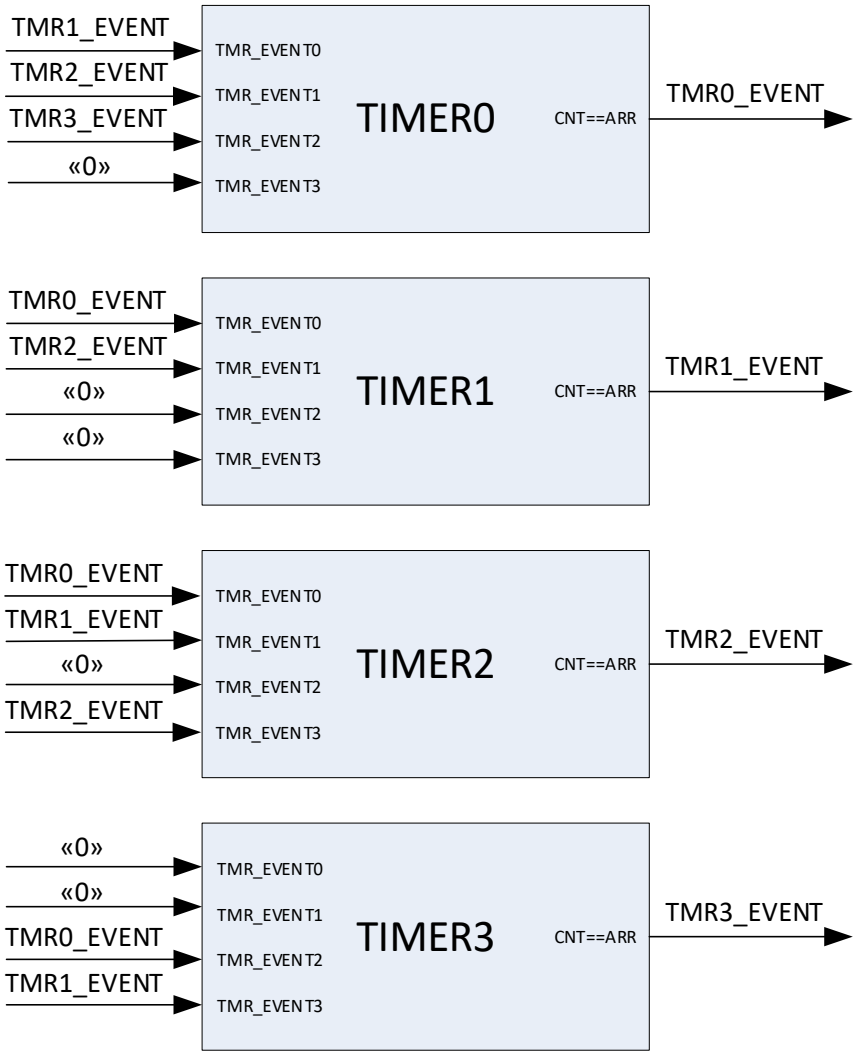


Рисунок 90 – Распределение выходов запуска по блокам таймеров

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. На рисунке 91 показан пример каскадного соединения таймеров, диаграммы работы данных таймеров приведены на рисунке 92.

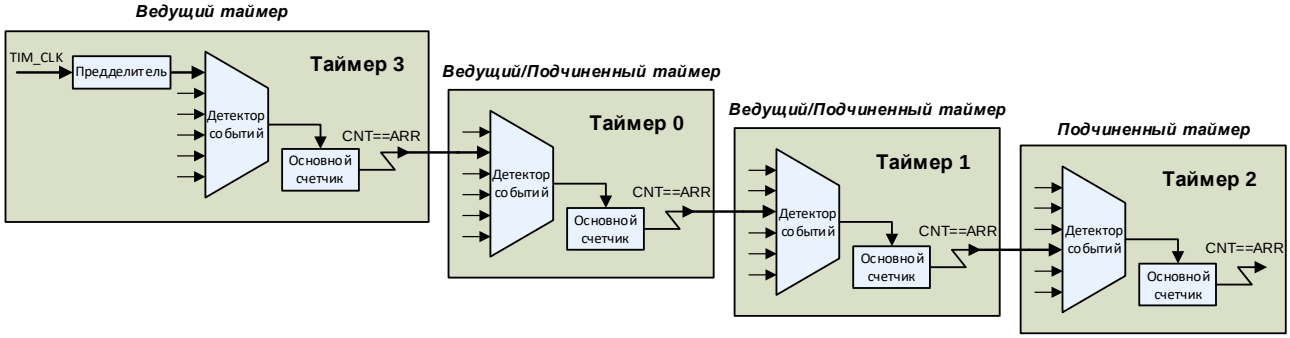


Рисунок 91 – Пример каскадного соединения таймеров

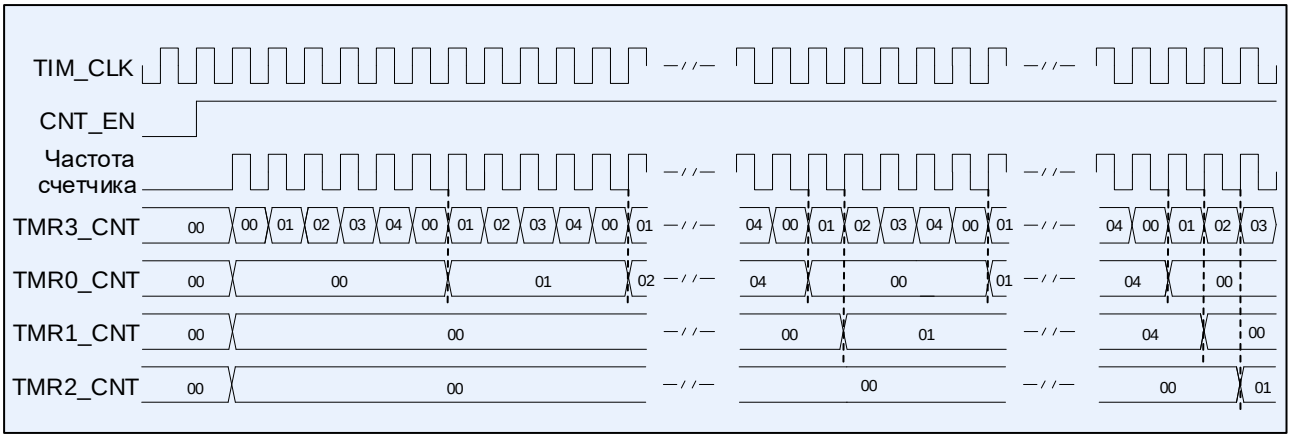


Рисунок 92 – Диаграммы работы четырех таймеров в каскаде
TIMER3: DIR = 0, EVENT_SEL[3:0] = 0000, CNT_MODE[1:0] = 00;
TIMER0: DIR = 0, EVENT_SEL[3:0] = 0011, CNT_MODE[1:0] = 10;
TIMER1: DIR = 0, EVENT_SEL[3:0] = 0001, CNT_MODE[1:0] = 10;
TIMER2: DIR = 0, EVENT_SEL[3:0] = 0010, CNT_MODE[1:0] = 10

7.16.3.3 **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.

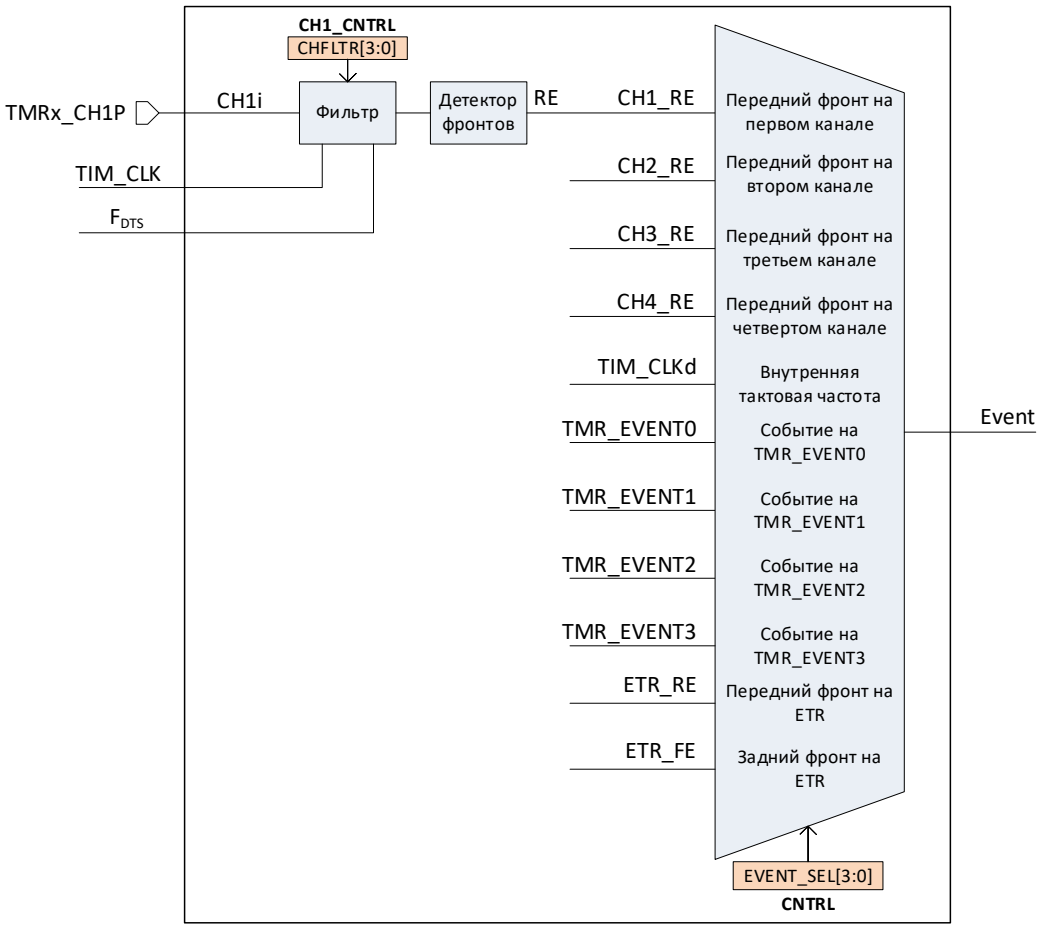


Рисунок 93 – Схема тактирования сигналом со входа первого канала

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. пункт 7.16.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле CHFLTR[3:0] регистра CHy_CNTRL.

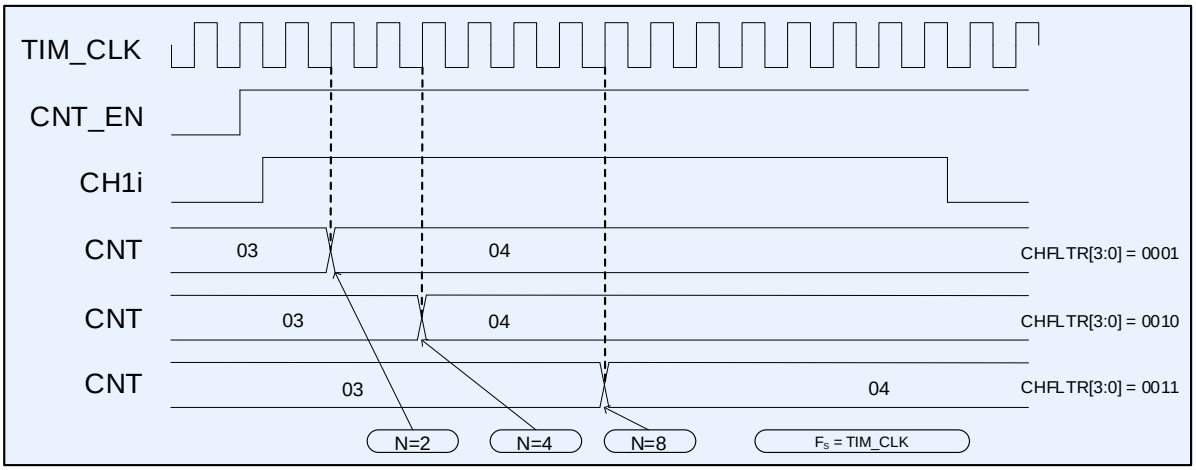


Рисунок 94 – Диаграмма внешнего тактирования с разными вариантами фильтра

7.16.3.4 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. пункт 7.16.6 «Блок цифрового фильтра»). Настройки фильтра задаются в поле ETR_FILTER[3:0].

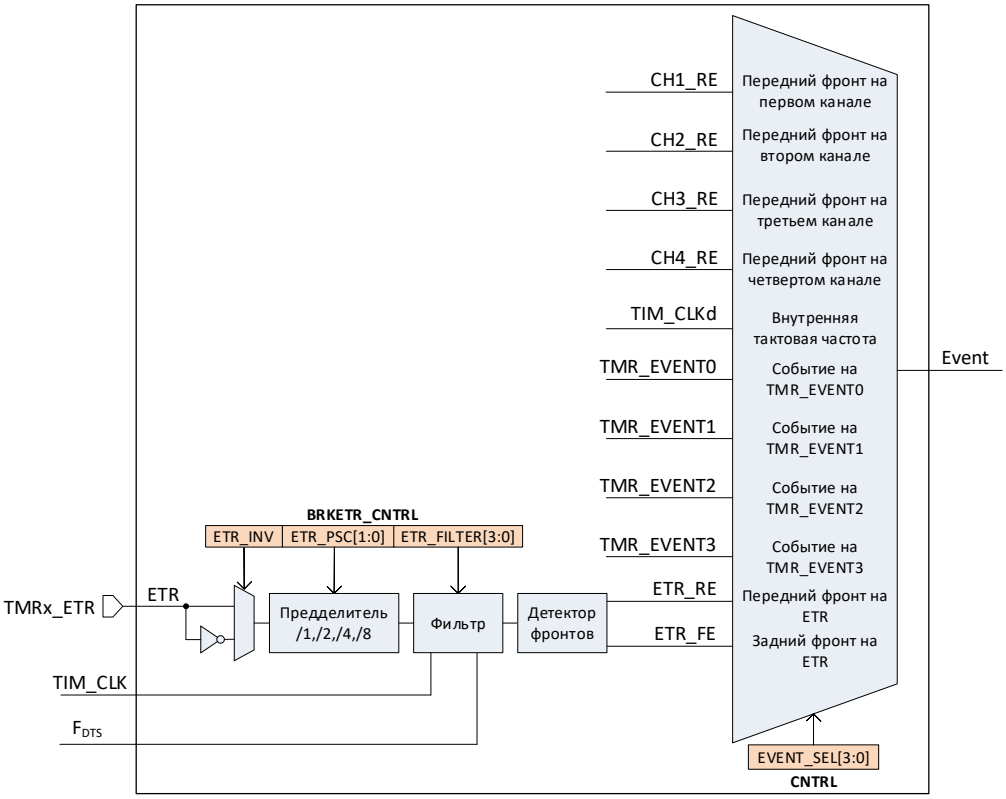


Рисунок 95 – Схема тактирования сигналом со входа ETR

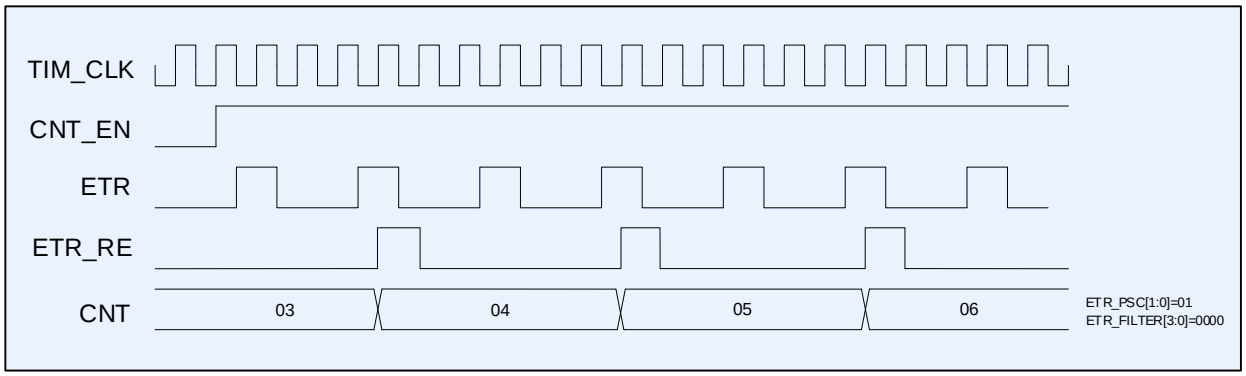


Рисунок 96 – Диаграмма тактирования со входа ETR, EVENT_SEL[3:0] = 1000

7.16.4 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_CHyP).

Структурная схема блока захвата представлена на рисунке 97.

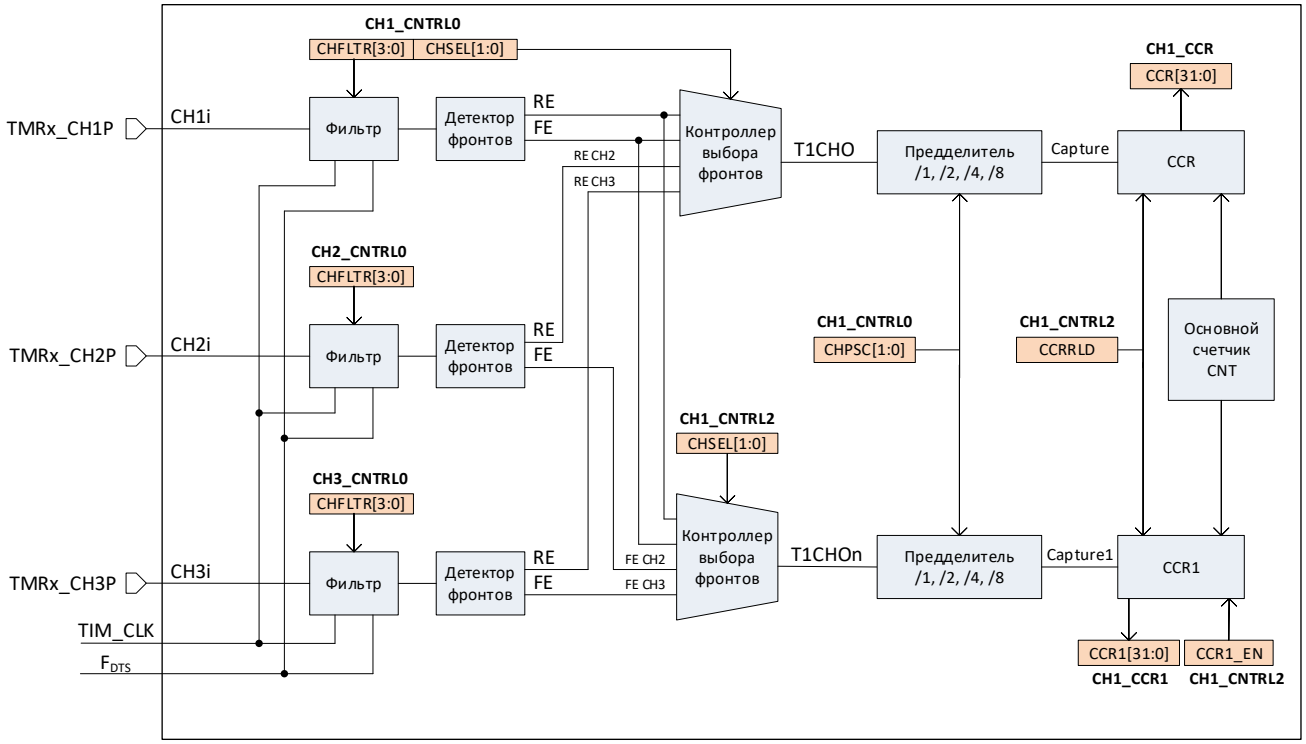


Рисунок 97 – Структурная схема блока захвата на примере канала 1

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. пункт 7.16.6 «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 98 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.

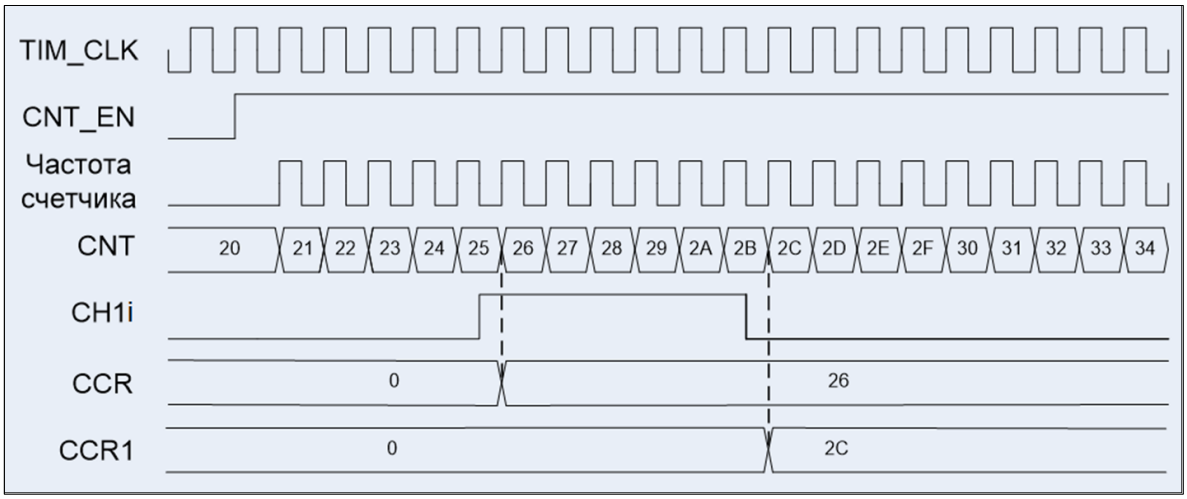


Рисунок 98 – Диаграмма захвата события со входа канала 1

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».

7.16.5 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 99.

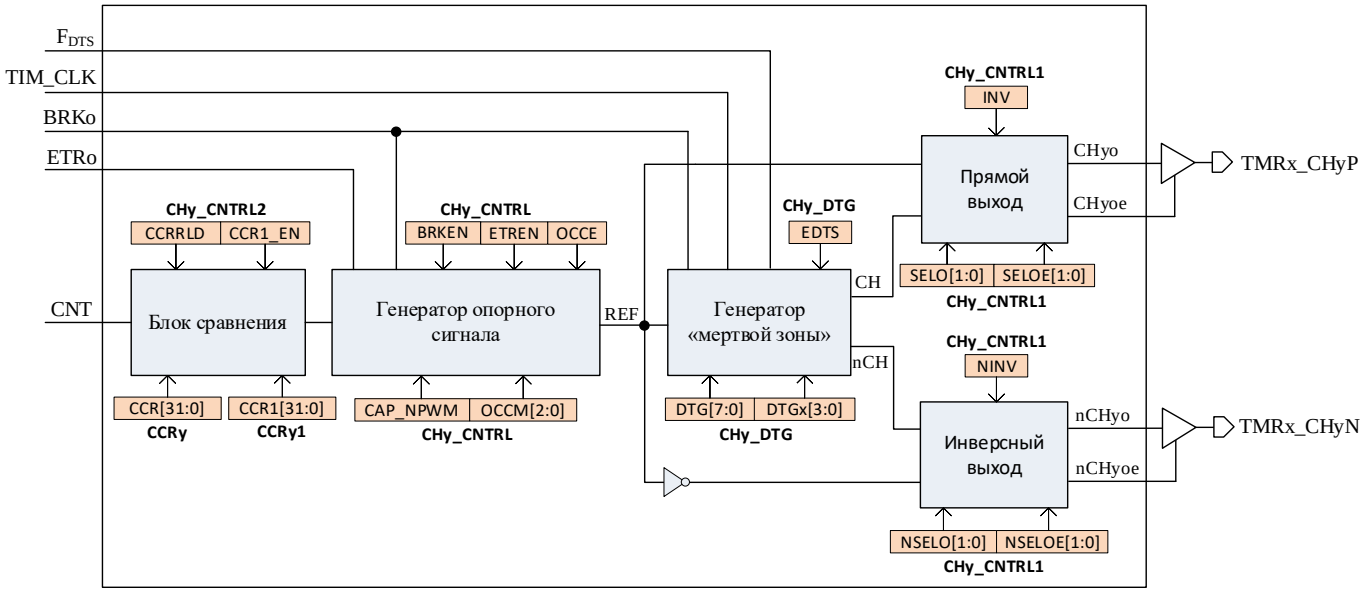


Рисунок 99 – Структурная схема блока формирования ШИМ

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

7.16.5.1 **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.

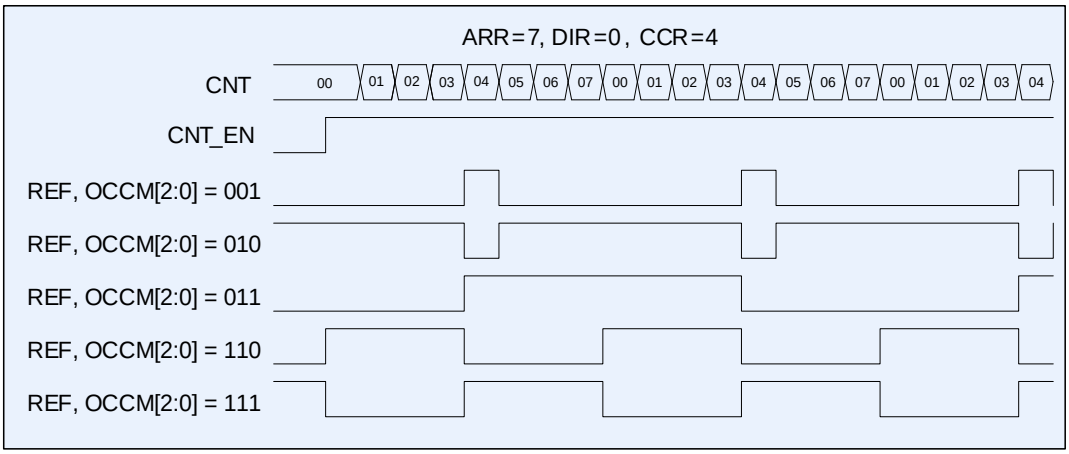


Рисунок 100 – Диаграмма работы в режиме ШИМ, CCR1_EN=0

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.

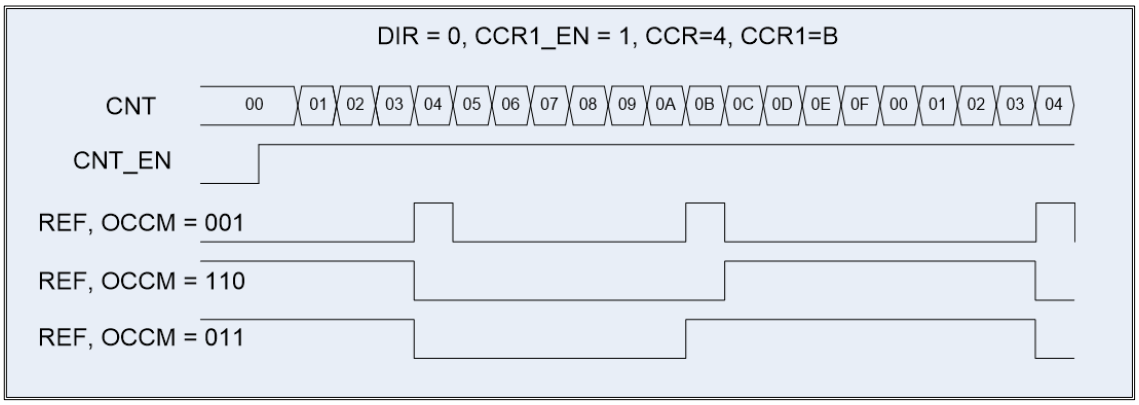


Рисунок 101 – Диаграмма работы в режиме ШИМ, CCR1_EN = 1

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 102.

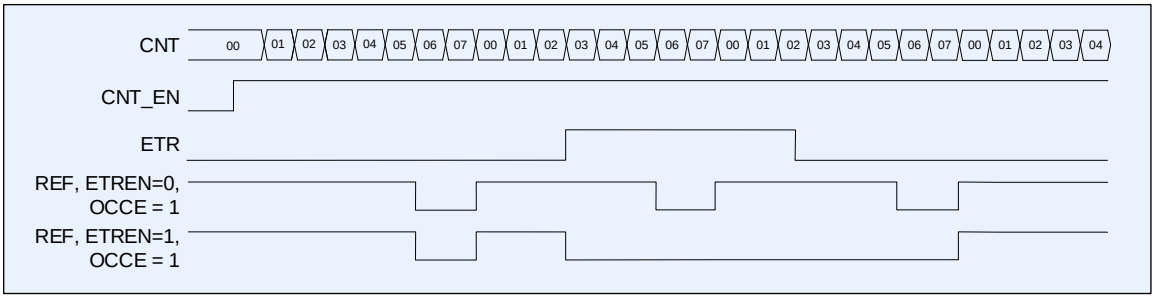


Рисунок 102 – Диаграмма сброса сигнала REF по выводу ETR

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 103.

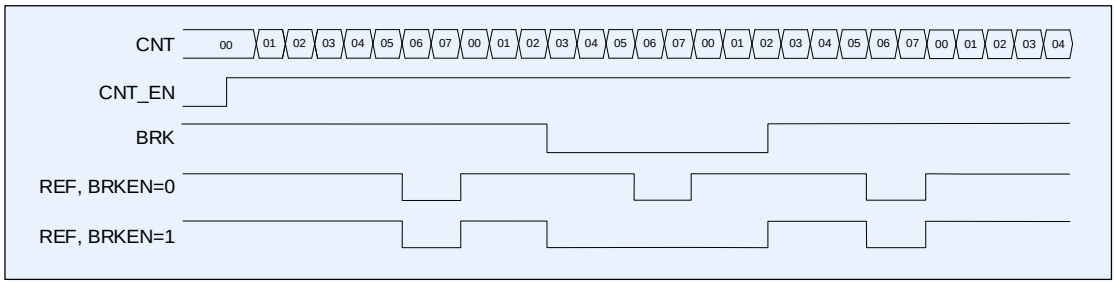


Рисунок 103 – Диаграмма сброса сигнала REF по выводу BRK

7.16.5.2 **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 FDTS по формуле

$$\mathrm{DTGdel} = \mathrm{DTG} \cdot (\mathrm{DTGx} + 1)\,, \qquad (6)$$

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.

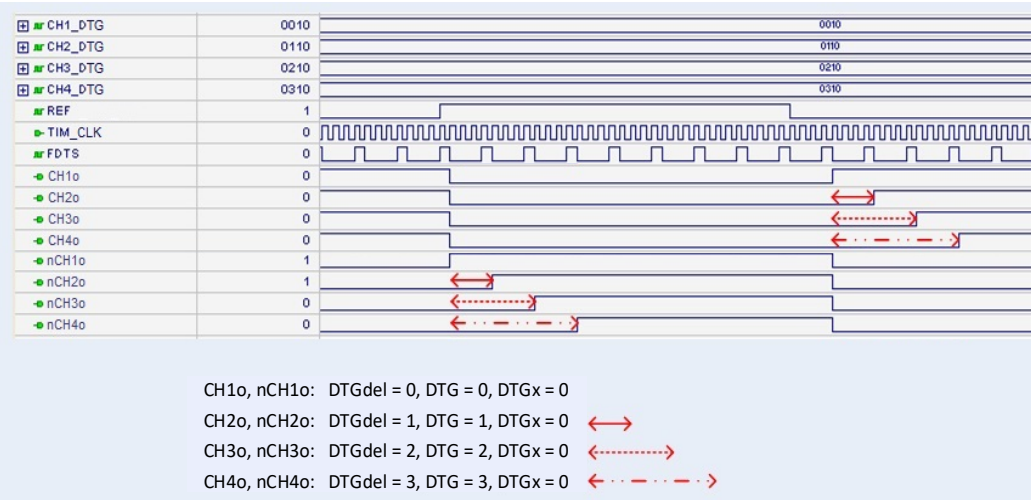


Рисунок 104 – Диаграмма работы блока DTG

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.

7.16.5.3 **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.

7.16.6 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 FS, на которой входной сигнал захватывается в сдвиговый регистр для накопления. В качестве частоты FS может использоваться частота TIM_CLK или FDTS;

– 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 FS 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 FS 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 105, частота FS = TIM_CLK, количество выборок N = 4.

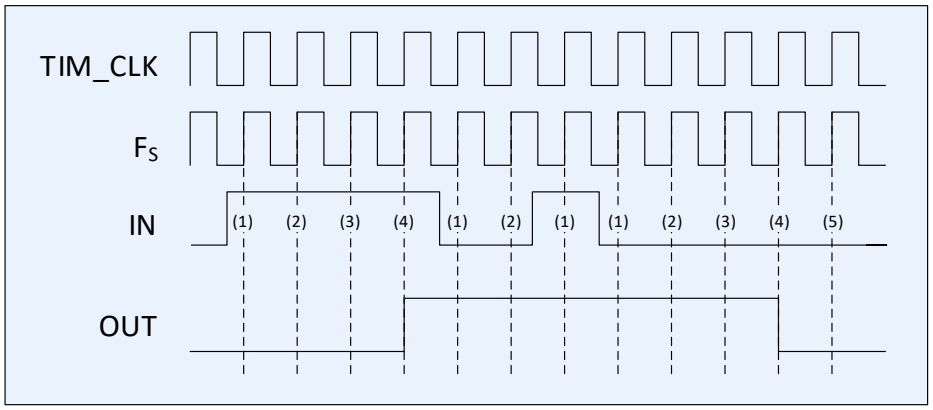


Рисунок 105 – Диаграмма работы фильтра, F_S = TIM_CLK, N = 4

Для задания длительных интервалов накопления входного сигнала имеется возможность использовать частоту F_{DTS}, которая формируется из частоты TIM_CLK путем прореживания на заданный коэффициент (см. подпункт 7.16.2.4 «Тактовая частота F_{DTS}»).

Диаграмма работы фильтра при использовании частоты FDTS приведена на рисунке 106, частота выборки $F_S = F_{DTS}/2$, количество выборок N = 6.

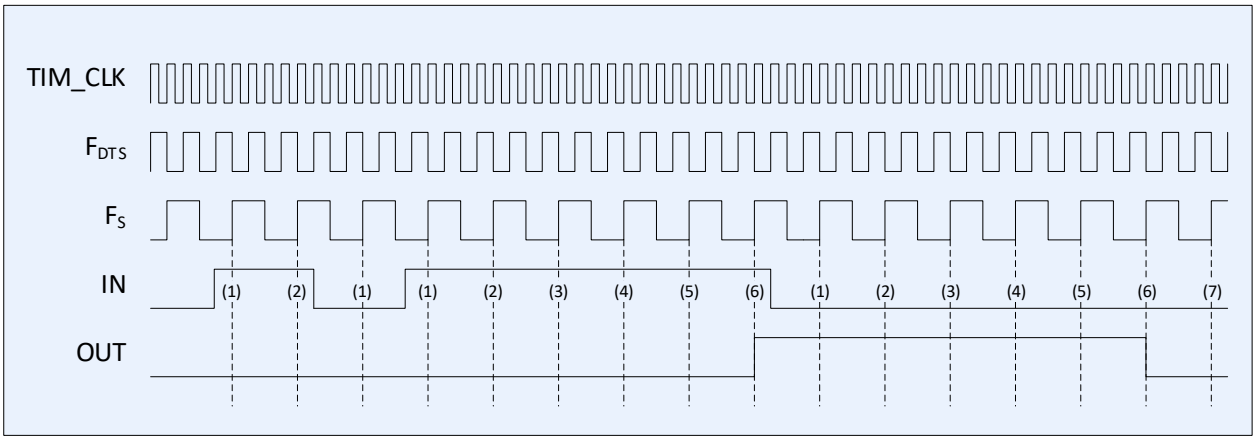


Рисунок 106 – Диаграмма работы фильтра, FS = FDTS/2, N = 6

Возможные варианты настройки блока фильтра приведены в таблице 305.

Таблица 305 – 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	$12 \times T_{FDTS}$

CHFLTR[3:0], ETR_FILTER[3:0]	Частота выборки Fs	Количество выборок N	Минимальная длительность импульсов, которые не будут отфильтрованы
0101	$F_{DTS}/2$	8	$16 \times T_{FDTS}$
0110	$F_{DTS}/4$	6	$24 \times T_{FDTS}$
0111	$F_{DTS}/4$	8	$32 \times T_{FDTS}$
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	$256 \times T_{FDTS}$

7.16.7 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.

7.16.7.1 **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.

7.16.7.2 **Прерывания**

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

7.16.7.3 **Запросы DMA**

На основе отслеживаемых событий блок таймера формирует сигналы запросов DMA TMRx_REQ и TMRx_REQ1-TMRx_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_REQ;

– регистры DMA_RE1-DMA_RE4 конфигурируют формирование запросов TMRx_REQ1-TMRx_REQ4, соответственно.

Запросы TMRx_REQ1-TMRx_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_REQ1 при конфигурации регистра DMA_RE1 = 0x80.

7.16.8 Примеры

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

7.16.8.1 **Обычный счетчик**

```
CLKCNTR->PER0_CLK |= 0x00800000; //Разрешение частоты PCLK для таймера 0
CLKCNTR->TIM0_CLK = 0x00010000; //Настройка и подача частоты TIM_CLK
                                //для таймера 0 (TIM_CLK = MAX_CLK)
TIMER0->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER0->CNT = 0x00000000;    //Начальное значение счетчика
TIMER0->PSG = 0x00000000;    //Предделитель частоты TIM_CLK
TIMER0->ARR = 0x0000000F;    //Основание счета
TIMER0->IE  = 0x00000002;    //Разрешение генерировать прерывание при CNT =
ARR
//Разрешение работы таймера
TIMER0->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
```

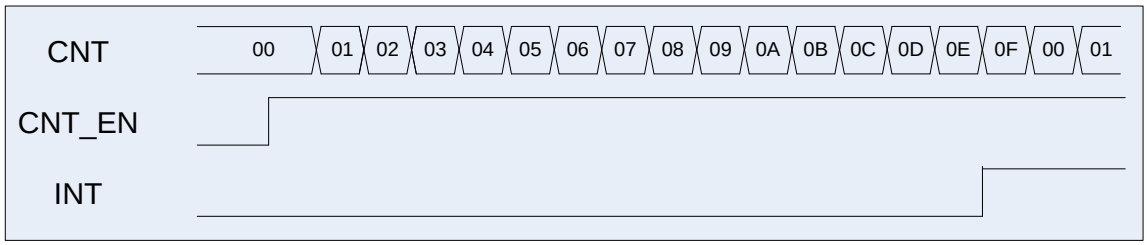


Рисунок 107 – Режим обычного счетчика

7.16.8.2 **Режим захвата**

```
CLKCNTR->PER0_CLK |= 0x00800000; //Разрешение частоты PCLK для таймера 0
CLKCNTR->TIM0_CLK = 0x00010000;  //Настройка и подача частоты TIM_CLK
                                 //для таймера 0 (TIM_CLK = MAX_CLK)
TIMER0->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER0->CNT = 0x00000000;     //Начальное значение счетчика
TIMER0->PSG = 0x00000000;     //Предделитель частоты TIM_CLK
TIMER0->ARR = 0x000000FF;     //Основание счета
TIMER0->IE = 0x000001E0;      //Разрешение генерировать прерывание по событию
                              //настроенного фронта на входах CH1i-CH4i
TIMER0->CH1_CNTRL = 0x00008000;  //Захват по положительному фронту сигнала
                                 //на входе CH1i, фильтрация отключена
TIMER0->CH2_CNTRL = 0x00008010;  //Захват по отрицательному фронту сигнала
                                 //на входе CH2i, фильтрация отключена
TIMER0->CH3_CNTRL = 0x00008001;  //Захват по положительному фронту сигнала
                                 //на входе CH3i, фильтрация выполняется
                                 //по двум выборкам на частоте TIM_CLK
TIMER0->CH4_CNTRL = 0x00008011;  //Захват по отрицательному фронту сигнала
                                 //на входе CH4i, фильтрация выполняется
                                 //по двум выборкам на частоте TIM_CLK
//Режим работы выхода канала – канал работает на вход
TIMER0->CH1_CNTRL1= 0x00000000;
TIMER0->CH2_CNTRL1= 0x00000000;
TIMER0->CH3_CNTRL1= 0x00000000;
TIMER0->CH4_CNTRL1= 0x00000000;
//Разрешение работы таймера
TIMER0->CNTRL = 0x00000001; //Счет прямой по TIM_CLK
```

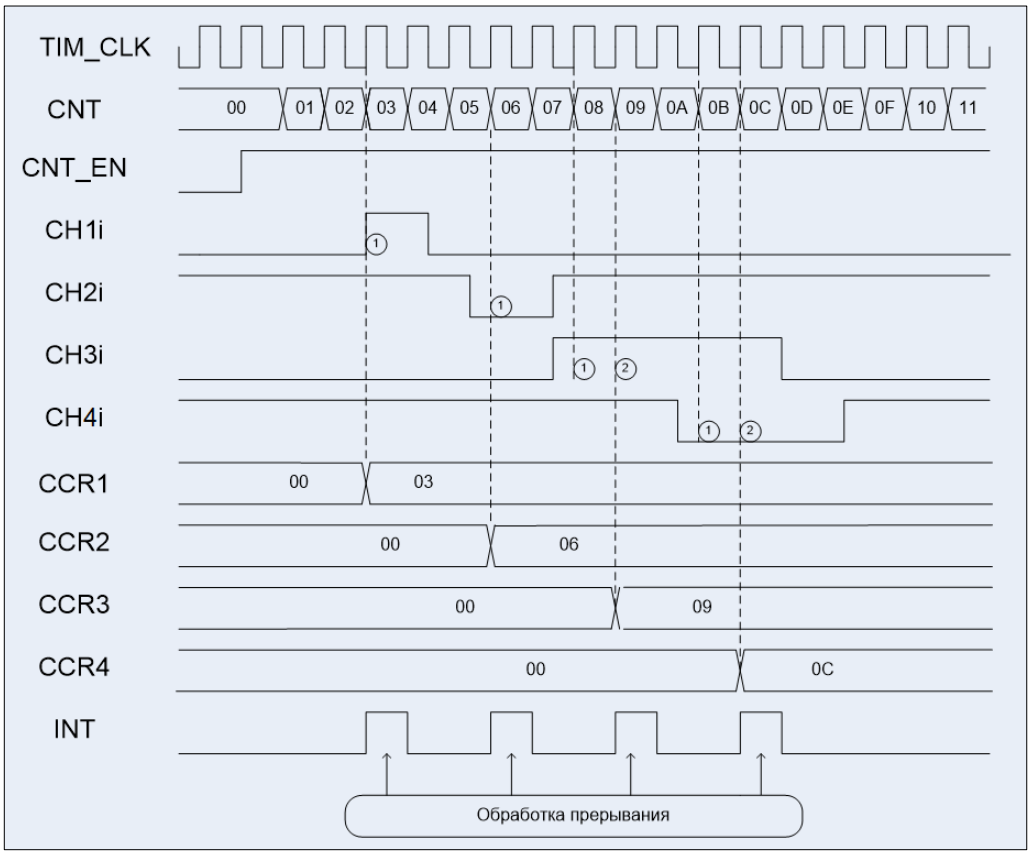


Рисунок 108 – Диаграммы примера работы в режиме захвата

7.16.8.3 **Режим ШИМ**

```
CLKCNTR->PER0_CLK |= 0x00800000; //Разрешение частоты PCLK для таймера 0
CLKCNTR->TIM0_CLK = 0x00010000;  //Настройка и подача частоты TIM_CLK
                                 //для таймера 0 (TIM_CLK = MAX_CLK)
TIMER0->CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMER0->CNT = 0x00000000;     //Начальное значение счетчика
TIMER0->PSG = 0x00000000;     //Предделитель частоты TIM_CLK
TIMER0->ARR = 0x00000010;     //Основание счета
TIMER0->IE  = 0x00001E00;     //Разрешение генерировать прерывание по событию
                              //переднего фронта на выходе REF для всех каналов
//Режим работы каналов – ШИМ
TIMER0->CH1_CNTRL = 0x00000200;  //REF = 1, если CNT == CCR
TIMER0->CH2_CNTRL = 0x00000200;  //REF = 1, если CNT == CCR
TIMER0->CH3_CNTRL = 0x00000400;  //REF = 0, если CNT == CCR
TIMER0->CH4_CNTRL = 0x00000600;  //Переключение REF, если CNT == CCR
//Режим работы выхода канала – канал работает на выход,
//на выходы канала выдается сигнал REF
TIMER0->CH1_CNTRL1= 0x00000909;
TIMER0->CH2_CNTRL1= 0x00000909;
TIMER0->CH3_CNTRL1= 0x00000909;
TIMER0->CH4_CNTRL1= 0x00000909;
```

```
//Установка значений CCR, с которыми сравнивается CNT при работе в режиме ШИМ
TIMER0->CCR1 = 0x00000003;
TIMER0->CCR2 = 0x00000006;
TIMER0->CCR3 = 0x00000009;
TIMER0->CCR4 = 0x0000000F;
//Разрешение работы таймера
TIMER0->CNTRL = 0x00000001;   //Счет прямой по TIM_CLK
```

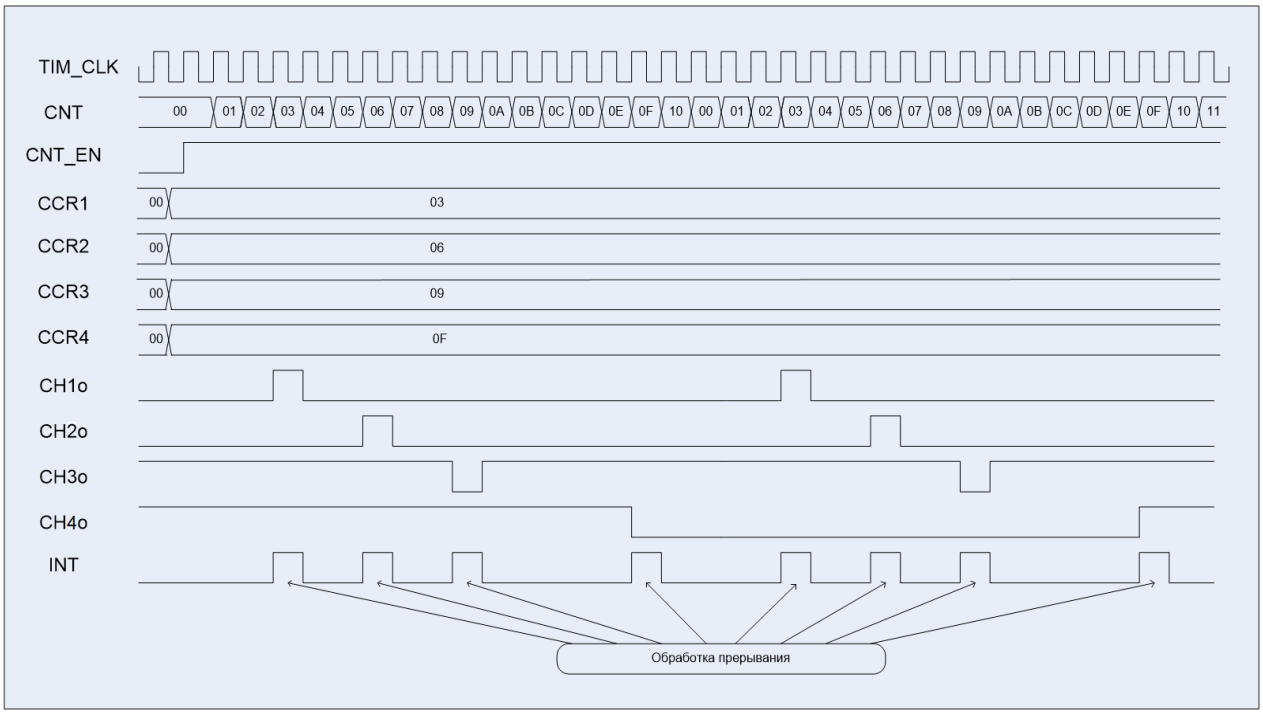


Рисунок 109 – Диаграммы примера работы в режиме ШИМ

7.16.9 Описание регистров блока таймера

Таблица 306 – Базовые адреса и смещения регистров управления таймера

Базовый адрес	Название	Описание
0x4008_A000	MDR_TMR0	Контроллер TIMER0
0x4008_B000	MDR_TMR1	Контроллер TIMER1
0x4008_C000	MDR_TMR2	Контроллер TIMER2
0x4008_D000	MDR_TMR3	Контроллер TIMER3
Смещение		
0x00	CNT	Основной счетчик таймера
0x04	PSG	Делитель частоты TIM_CLK для тактирования основного счетчика
0x08	ARR	Основание счета основного счетчика
0x0C	CNTRL	Регистр управления основным счетчиком
0x50	BRKETR_CNTRL	Регистр управления входом BRK и ETR
0x54	STATUS	Регистр статуса таймера
0x58	IE	Регистр разрешения прерываний
0x5C	DMA_RE	Регистр разрешения формирования запроса DMA – TMRx_REQ

Базовый адрес	Название	Описание
0x80	DMA_RE1	Регистр разрешения формирования запроса DMA – TMRx_REQ1
0x84	DMA_RE2	Регистр разрешения формирования запроса DMA – TMRx_REQ2
0x88	DMA_RE3	Регистр разрешения формирования запроса DMA – TMRx_REQ3
0x8C	DMA_RE4	Регистр разрешения формирования запроса DMA – TMRx_REQ4
Канал 1		
0x10	CCR1	Регистр сравнения/захвата для 1 канала таймера
0x20	CH1_CNTRL	Регистр управления для 1 канала таймера
0x30	CH1_CNTRL1	Регистр управления 1 для 1 канала таймера
0x40	CH1_DTG	Регистр управления DTG для 1 канала таймера
0x60	CH1_CNTRL2	Регистр управления 2 для 1 канала таймера
0x70	CCR11	Регистр сравнения/захвата 1 для 1 канала таймера
Канал 2		
0x14	CCR2	Регистр сравнения/захвата для 2 канала таймера
0x24	CH2_CNTRL	Регистр управления для 2 канала таймера
0x34	CH2_CNTRL1	Регистр управления 1 для 2 канала таймера
0x44	CH2_DTG	Регистр управления DTG для 2 канала таймера
0x64	CH2_CNTRL2	Регистр управления 2 для 2 канала таймера
0x74	CCR21	Регистр сравнения/захвата 1 для 2 канала таймера
Канал 3		
0x18	CCR3	Регистр сравнения/захвата для 3 канала таймера
0x28	CH3_CNTRL	Регистр управления для 3 канала таймера
0x38	CH3_CNTRL1	Регистр управления 1 для 3 канала таймера
0x48	CH3_DTG	Регистр управления DTG для 3 канала таймера
0x68	CH3_CNTRL2	Регистр управления 2 для 3 канала таймера
0x78	CCR31	Регистр сравнения/захвата 1 для 3 канала таймера
Канал 4		
0x1C	CCR4	Регистр сравнения/захвата для 4 канала таймера
0x2C	CH4_CNTRL	Регистр управления для 4 канала таймера
0x3C	CH4_CNTRL1	Регистр управления 1 для 4 канала таймера
0x4C	CH4_DTG	Регистр управления DTG для 4 канала таймера
0x6C	CH4_CNTRL2	Регистр управления 2 для 4 канала таймера
0x7C	CCR41	Регистр сравнения/захвата 1 для 4 канала таймера

7.16.9.1 **CNT**

Таблица 307 – Основной счетчик таймера CNT

Номер	31…0
Доступ	R/W
Сброс	0

	CNT[31:0]

Таблица 308 – 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

7.16.9.2 **PSG**

Таблица 309 – Делитель частоты TIM_CLK для счета основного счетчика PSG

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

Таблица 310 – Описание бит регистра PSG

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

7.16.9.3 **ARR**

Таблица 311 – Основание счета основного счетчика ARR

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

Таблица 312 – 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]

7.16.9.4 **CNTRL**

Таблица 313 – Регистр управления основным счетчиком CNTRL

Номер	31…12	11…8
Доступ	U	R/W
Сброс	0	0

	-	EVENT_SEL[3:0]

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

Таблица 314 – Описание бит регистра CNTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений.
31…12	-	Зарезервировано
11...8	EVENT_SEL[3:0]	Биты выбора источника событий: 0000 – внутренняя тактовая частота TIM_CLKd (формируется путем деления частоты TIM_CLK); 0001 – событие на TMR_EVENT0; 0010 – событие на TMR_EVENT1; 0011 – событие на TMR_EVENT2; 0100 – событие переднего фронта на канале 1, «Режим 1»; 0101 – событие переднего фронта на канале 2, «Режим 1»; 0110 – событие переднего фронта на канале 3, «Режим 1»; 0111 – событие переднего фронта на канале 4, «Режим 1»; 1000 – событие переднего фронта на ETR, «Режим 2»; 1001 – событие заднего фронта на ETR, «Режим 2»; 1010 – событие на TMR_EVENT3; 1011-1111 – зарезервировано. Примечание – подключение входов таймера TMR_EVENTx к выходам запуска других таймеров приведено на рисунке 90
7..6	CNT_MODE[1:0]	Режим счета основного счетчика: 00 – счетчик прямой при DIR = 0; счетчик обратный при DIR = 1; 01 – счетчик двунаправленный с автоматическим изменением DIR при CNT == 0 или CNT == ARR; 10 – счетчик прямой при DIR = 0; счетчик обратный при DIR = 1; 11 – зарезервировано. Режим счета CNT_MODE[1:0] необходимо устанавливать в соответствии со значением в поле EVENT_SEL[3:0]: – EVENT_SEL[3:0] = 0000: CNT_MODE[1:0] = 00 или 01; – EVENT_SEL[3:0] != 0000: CNT_MODE[1:0] = 10

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

7.16.9.5 **CCRy**

Таблица 315 – Регистр сравнения/захвата для ‘y’ канала таймера CCRy

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

Таблица 316 – 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. Режим ШИМ: значение CCR, с которым сравнивается CNT.

7.16.9.6 **CCRy1**

Таблица 317 – Регистр сравнения/захвата для ‘y’ канала таймера CCRy1

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

Таблица 318 – 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. Режим ШИМ: значение CCR1, с которым сравнивается CNT.

7.16.9.7 **CHy_CNTRL**

Таблица 319 – Регистр управления для ‘y’ канала таймера CHy_CNTRL

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

Номер	12	11…9	8	7…6	5…4	3…0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	000	0	00	00	0000
	BRKEN	OCCM[2:0]	OCCE	CHPSC[1:0]	CHSEL[1:0]	CHFLTR[3:0]

Таблица 320 – Описание бит регистра CHy_CNTRL

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16	WR_CMPL1	Флаг выполнения записи нового значения в регистр CCRy1: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
15	CAP_NPWM	Режим работы канала: 0 – канал работает в режиме ШИМ; 1 – канал работает в режиме захвата
14	WR_CMPL	Флаг выполнения записи нового значения в регистр CCRy: 0 – новые данные можно записывать; 1 – данные не записаны и идет запись
13	ETREN	Разрешение сброса сигнала REF в «0» при высоком уровне на входе ETR: 0 – запрещен; 1 – разрешен
12	BRKEN	Разрешение сброса сигналов REF и DTG в «0» при низком уровне на входе BRK: 0 – запрещен; 1 – разрешен
11…9	OCCM[2:0]	Формат выработки сигнала REF в режиме ШИМ: Если CCR1_EN=0: 000 – всегда 0;

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		001 – 1, если CNT==CCR; 010 – 0, если CNT==CCR; 011 – переключение REF, если CNT==CCR; 100 – всегда 0; 101 – всегда 1; 110 – 1, если DIR=0 (счет прямой), CNT<CCR, иначе 0; 0, если DIR=1 (счет обратный), CNT>CCR, иначе 1; 111 – 0, если DIR=0 (счет прямой), CNT<CCR, иначе 1; 1, если DIR=1 (счет обратный), CNT>CCR, иначе 0. Если CCR1_EN=1: 000 – всегда 0; 001 – 1, если CNT==CCR или CNT==CCR1; 010 – 0, если CNT==CCR или CNT==CCR1; 011 – переключение REF, если CNT==CCR или CNT==CCR1; 100 – всегда 0; 101 – всегда 1; 110 – 0, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 1; 0, если DIR=1 (счет обратный), CCR<CNT<CCR1, иначе 1; 111 – 1, если DIR=0 (счет прямой), CCR≤CNT≤CCR1, иначе 0; 1, если DIR=1 (счет обратный), CCR<CN <CCR1, иначе 0; Необходимо соблюдать условие CCR<CCR1
8	OCCE	Разрешение работы ETR: 0 – запрещен; 1 – разрешен
7…6	CHPSC[1:0]	Предварительный делитель входного канала: 00 – нет деления; 01 – /2; 10 – /4; 11 – /8
5…4	CHSEL[1:0]	Выбор события по входному каналу CHyi для фиксации значения основного счетчика (регистр CNT) в регистр CCRy: 00 – положительный фронт на входном канале CHyi; 01 – отрицательный фронт на входном канале CHyi; 10 – положительный фронт от других каналов: для первого канала от второго канала; для второго канала от третьего канала; для третьего канала от четвертого канала; для четвертого канала от первого канала; 11 – положительный фронт от других каналов: для первого канала от третьего канала; для второго канала от четвертого канала; для третьего канала от первого канала; для четвертого канала от второго канала

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3…0	CHFLTR[3:0]	Конфигурация фильтра на входе канала ‘y’. Выбор частоты выборки Fs и количества выборок N: 0000 – нет фильтрации, $F_S = F_{DTS}$; 0001 – F_S = TIM_CLK, N = 2; 0010 – F_S = TIM_CLK, N = 4; 0011 – F_S = TIM_CLK, N = 8; 0100 – $F_S = F_{DTS}/2$, N = 6; 0101 – $F_S = F_{DTS}/2$, N = 8; 0110 – $F_S = F_{DTS}/4$, N = 6; 0111 – $F_S = F_{DTS}/4$, N = 8; 1000 – $F_S = F_{DTS}/8$, N = 6; 1001 – $F_S = F_{DTS}/8$, N = 8; 1010 – $F_S = F_{DTS}/16$, N = 5; 1011 – $F_S = F_{DTS}/16$, N = 6; 1100 – $F_S = F_{DTS}/16$, N = 8; 1101 – $F_S = F_{DTS}/32$, N = 5; 1110 – $F_S = F_{DTS}/32$, N = 6; 1111 – $F_S = F_{DTS}/32$, N = 8

7.16.9.8 **CHy_CNTRL1**

Таблица 321 – 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 322 – Описание бит регистра CHy_CNTRL1

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

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

7.16.9.9 **CHy_CNTRL2**

Таблица 323 – 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 324 – 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: 0 – сигнал события захвата устанавливается в момент обнаружения события, при этом обновление регистров CCRy и CCRy1 выполняется через один такт TIM_CLK; 1 – сигнал события захвата устанавливается синхронно с обновлением информации в регистрах CCRy и CCRy1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	CCRRLD	Режим обновления регистров CCRy и CCRy1: 0 – обновление возможно в любой момент времени; 1 – обновление будет осуществлено только при CNT == 0
2	CCR1_EN	Разрешение работы регистра CCRy1: 0 – CCRy1 не используется; 1 – CCRy1 используется
1…0	CHSEL[1:0]	Выбор события по входному каналу CHyi для фиксации значения основного счетчика (регистр CNT) в регистр CCRy1: 00 – положительный фронт на входном канале CHyi; 01 – отрицательный фронт на входном канале CHyi; 10 – отрицательный фронт от других каналов: для первого канала от второго канала; для второго канала от третьего канала; для третьего канала от четвертого канала; для четвертого канала от первого канала; 11 – отрицательный фронт от других каналов: для первого канала от третьего канала; для второго канала от четвертого канала; для третьего канала от первого канала; для четвертого канала от второго канала

7.16.9.10 **CHy_DTG**

Таблица 325 – Регистр управления генератором «мертвой зоны» CHy_DTG

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

Таблица 326 – Описание бит регистра CHy_DTG

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

7.16.9.11 **BRKETR_CNTRL**

Таблица 327 – Регистр BRKETR_CNTRL управления входом BRK и ETR

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

Таблица 328 – 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: 0000 – нет фильтрации, $F_S = F_{DTS}$; 0001 – F_S = TIM_CLK, N = 2; 0010 – F_S = TIM_CLK, N = 4; 0011 – F_S = TIM_CLK, N = 8; 0100 – $F_S = F_{DTS}/2$, N = 6; 0101 – $F_S = F_{DTS}/2$, N = 8; 0110 – $F_S = F_{DTS}/4$, N = 6; 0111 – $F_S = F_{DTS}/4$, N = 8; 1000 – $F_S = F_{DTS}/8$, N = 6; 1001 – $F_S = F_{DTS}/8$, N = 8; 1010 – $F_S = F_{DTS}/16$, N = 5; 1011 – $F_S = F_{DTS}/16$, N = 6; 1100 – $F_S = F_{DTS}/16$, N = 8; 1101 – $F_S = F_{DTS}/32$, N = 5; 1110 – $F_S = F_{DTS}/32$, N = 6; 1111 – $F_S = F_{DTS}/32$, N = 8
3…2	ETR_PSC[1:0]	Асинхронный предделитель частоты со входа ETR: 00 – без деления; 01 – /2; 10 – /4; 11 – /8
1	ETR_INV	Инверсия входа ETR: 0 – без инверсии; 1 – инверсия
0	BRK_INV	Инверсия входа BRK: 0 – без инверсии; 1 – инверсия

7.16.9.12 **STATUS**

Таблица 329 – Регистр статуса таймера STATUS

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

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

Таблица 330 – Описание бит регистра STATUS

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

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
4	BRK EVENT	Событие высокого уровня на входе BRK: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», при условии наличия низкого уровня на входе BRK
3	ETR FE EVENT	Событие заднего фронта на входе ETR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
2	ETR RE EVENT	Событие переднего фронта на входе ETR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием, приоритет у нового события
1	CNT ARR EVENT	Событие совпадения CNT с ARR: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса флага регистры CNT и ARR не изменили состояния, то флаг повторно не взводится
0	CNT ZERO EVENT	Событие совпадения CNT с нулем: 0 – нет события; 1 – есть событие. Сбрасывается записью «0», если запись происходит одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса флага регистр CNT не изменил состояния, то флаг повторно не взводится
Примечание - Для организации надежного сброса статусных флагов прерываний необходимо учитывать требования, изложенные в подразделе 9.3		

7.16.9.13 **IE**

Таблица 331 – Регистр разрешения прерываний таймера IE

Номер	31...17	16…13	12…9	8…5
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0

	-	CCR CAP1 EVENT IE[3:0]	CCR REF EVENT IE[3:0]	CCR CAP EVENT IE[3:0]

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

Таблица 332 – Описание бит регистра IE

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…17	-	Зарезервировано
16...13	CCR CAP1 EVENT IE[3:0]	Флаг разрешения прерывания по событию CCR CAP1 EVENT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
12…9	CCR REF EVENT IE[3:0]	Флаг разрешения прерывания по событию CCR REF EVENT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
8…5	CCR CAP EVENT IE[3:0]	Флаг разрешения прерывания по событию CCR CAP EVENT[3:0] в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено. Бит 0 – первый канал; Бит 3 – четвертый канал
4	BRK EVENT IE	Флаг разрешения прерывания по событию BRK EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
3	ETR FE EVENT IE	Флаг разрешения прерывания по событию ETR FE EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено
2	ETR RE EVENT IE	Флаг разрешения прерывания по событию ETR RE EVENT в регистре STATUS: 0 – прерывание запрещено; 1 – прерывание разрешено

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

7.16.9.14 **DMA_RE, DMA_RE1-DMA_RE4**

Таблица 333 – 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 334 – 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]: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – первый канал; Бит 3 – четвертый канал
12…9	CCR REF EVENT RE[3:0]	Флаг разрешения запроса DMA по событию CCR REF EVENT[3:0]: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – первый канал; Бит 3 – четвертый канал

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

7.17 Контроллер CAN

В микросхеме реализован контроллер интерфейса CAN, который является полнофункциональным CAN-узлом, отвечающими требованиям к активным и пассивным устройствам CAN 2.0A и 2.0B и поддерживающими передачу данных на скорости не более 1 Мбит/сек.

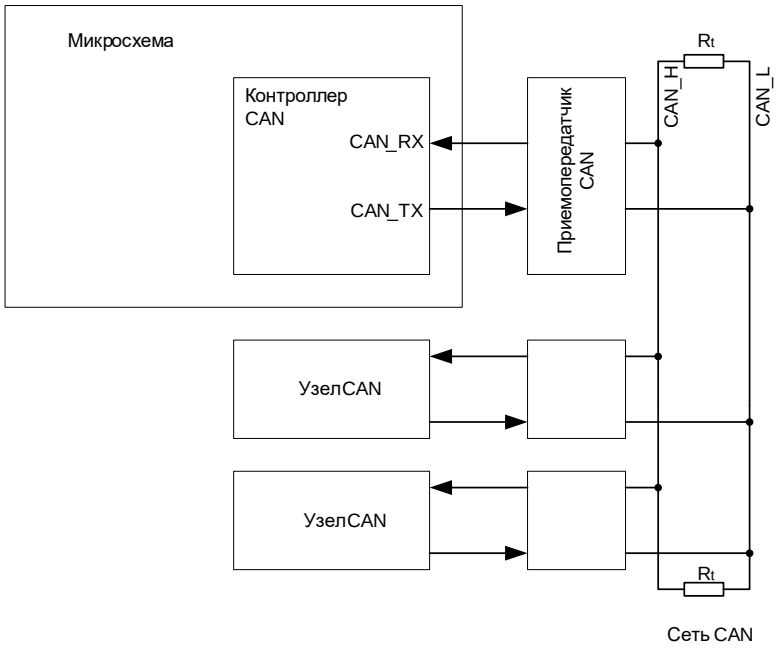


Рисунок 110 – Структурная блок-схема организации сети CAN

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 байтов 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 CAN2.0 A и B;
- скорость передачи до 1 Мбит/с;
- 32 буфера приема/передачи;
- поддержка приоритетов сообщений;
- 32 фильтра приема;
- маскирование прерываний.

7.17.1 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.

Режим нормальной передачи (регистр CAN_STATUS: ROM = 0, STM = 0)

Выводы CAN_TX и CAN_RX подключены к шине.

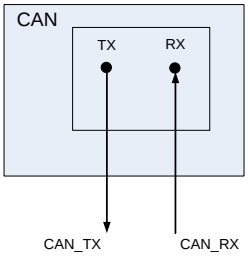


Рисунок 111 – Режим нормальной передачи

В этом режиме можно установить флаги разрешения приема своих пакетов и флаги разрешения подтверждения своих пакетов посылкой ACK (регистр CAN_CONTROL поля SAP и ROP).

Режим работы только на прием – ReceiveOnlyMode (регистр CAN_STATUS: ROM = 1, STM = 0)

Контроллер CAN интерфейса принимает, но не посылает никакой информации, то есть линия TX всегда в «1», но внутри контроллера все управляющие сигналы проходят.

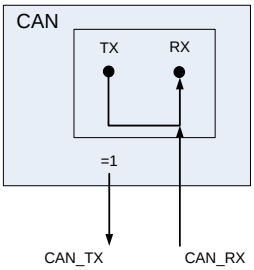


Рисунок 112 – Режим работы только на прием – ReceiveOnlyMode

Режим самотестирования – Self Test Mode (регистр CAN_STATUS: STM = 1, ROM = 0)

Выводы CAN_TX и CAN_RX отключены, вся передаваемая информация видна только внутри контроллера.

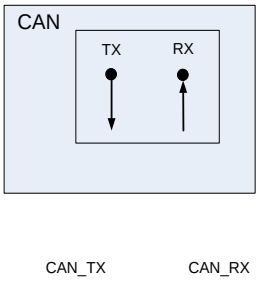


Рисунок 113 – Режим самотестирования – SelfTestMode

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.

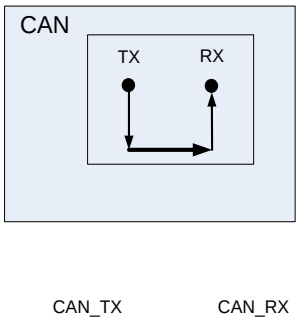


Рисунок 114 – Режим инициализации для задания параметров связи

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 имеется 32 регистра фильтров и 32 регистра масок, которые можно использовать для блокировки всех сообщений CAN, кроме избранных сообщений или групп сообщений.

7.17.2 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, он выставляет ACK подтверждение на шине, если хотя бы один из узлов обнаружит ошибку, 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.

7.17.3 Структура пакета данных (DataFrame)

Пакет данных состоит из семь различных полей:

- «начало пакета» (SOF-startofframe);
- «поле арбитража» (arbitrationfield);
- «поле контроля» (controlfield);
- «поле данных» (datafield);
- «поле CRC» (CRCfield);
- «поле подтверждения» (ACKfield);
- «конец пакета» (endofframe).

Поле данных может иметь нулевую длину.

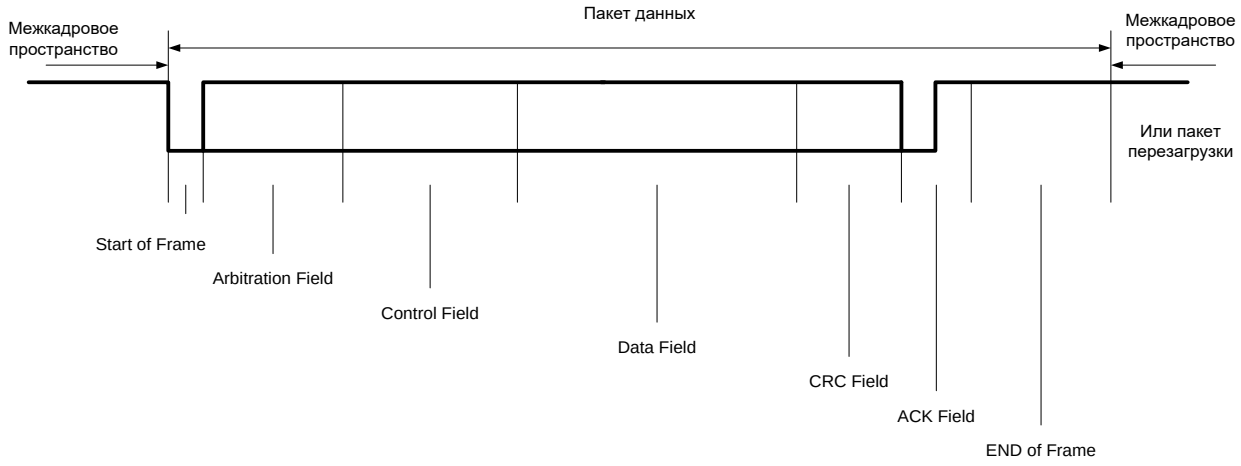


Рисунок 115 – Пакет сообщения CAN

В терминах протокола CAN логическая единица называется рецессивным битом, а логический ноль называется доминантным битом. Во всех случаях доминантный бит будет затираться рецессивным. То есть, если несколько узлов выставят на шину

рецессивный бит, а один – доминантный, то обратно всеми узлами будет считан доминантный бит.

7.17.3.1 **Начало пакета (Startofframe)**

Начало пакета отмечает начало пакета данных или пакета удаленного запроса данных. Это поле состоит их 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.

7.17.3.2 **Поле арбитража (Arbitrationfield)**

Формат поля арбитража отличается для стандартного и расширенного форматов:

– в стандартном формате поле арбитража состоит из 11 разрядного идентификатора и бита RTR;

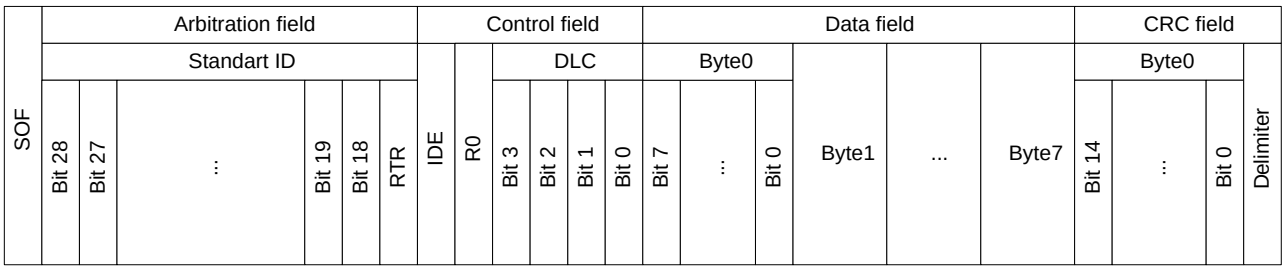


Рисунок 116 – Структура стандартного пакета данных

– в расширенном формате поле арбитража состоит из 29 разрядного идентификатора, бита SRR, бита IDE и бита RTR.

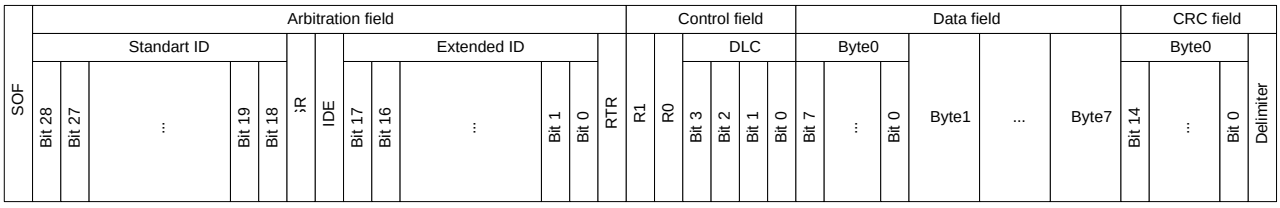


Рисунок 117 – Структура расширенного пакета данных

7.17.3.3 **Идентификатор**

Идентификатор – стандартный формат. Длина идентификатора – 11 бит и соответствует StandartID в расширенном формате. Эти биты передаются в порядке от Bit28 до Bit18. Самый младший бит – Bit18. 7 старших бит (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:

– StandartID – состоит из 11 бит. Эта секция передается в порядке от Bit28 до Bit18. Это эквивалентно формату стандартного идентификатора. StandartID определяет базовый приоритет расширенного пакета;

– ExtendedID – состоит из 18 бит. Эта секция передается в порядке от Bit17 до Bit0. В стандартном пакете идентификатор сопровождается битом RTR.

7.17.3.4 **Бит RTR**

Бит запроса удаленной передачи. В пакетах данных бит RTR должен быть передан нулевым уровнем. Внутри пакета удаленного запроса данных бит RTR должен быть единичным. В расширенном пакете сначала передается **Standart**ID, с последующими битами IDE и SRR. ExtendedID передается после бита SRR.

7.17.3.5 **Бит SRR (расширенный формат)**

Заменитель бита удаленного запроса. SRR – единичный бит. Он передается в расширенных пакетах в позиции бита RTR. Таким образом, он заменяет RTR – бит стандартного пакета.

Следовательно, при одновременной передаче стандартного пакета и расширенного пакета, StandartID которого совпадает с идентификатором стандартного пакета, стандартный пакет преобладает над расширенным пакетом.

7.17.3.6 **Бит IDE (расширенный формат)**

Бит IDE – 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.

7.17.3.7 **Поле управления (Controlfield)**

Поле управления состоит из шести бит. 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.

7.17.3.8 **Код длины данных (Datalengthcode)**

Число байт в поле данных обозначается кодом длины данных. Этот код длины данных, размером 4 бита, передается внутри поля управления. Допустимое число байт данных: {0,1, ..., 7,8}. Другие величины использоваться не могут.

7.17.3.9 **Поле данных (Datafield)**

Поле данных состоит из данных, которые будут переданы внутри пакета данных. Оно может содержать от 0 до 8 байт, каждый содержит 8 бит, которые передаются, начиная со старшего значащего бита.

7.17.3.10 **Поле CRC (CRCfield)**

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.

7.17.3.11 **Поле подтверждения (ACKfield)**

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».

7.17.3.12 **Конец пакета (End of frame)**

Каждый пакет данных и пакет удаленного запроса данных ограничен последовательностью флагов, состоящей из семи единичных бит.

7.17.4 Структура пакета удаленного запроса данных (Remoteframe)

Узел, действующий как приемник некоторых данных, может инициировать передачу соответственных данных исходными узлами, 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, бит RTR 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 [0,8]. Значение кода длины данных соответствует коду длины данных кадра данных. бит RTR указывает, является ли переданный кадр кадром данных.

7.17.5 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 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 бит 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 А и С выставляют на шину доминантный бит, а узел В выставляет рецессивный бит, но при этом считывает с шины доминантный. В результате узел освобождает шину и начинает следить за ее состоянием. Узлы А и С продолжают передачу, пока ситуация не повторится; теперь узел С выдает рецессивный бит, а узел А – доминантный. При этом узел С прекращает передачу и начинает следить за состоянием шины. С этого момента шина захватывается узлом А. После передачи сообщения узлом А узлы В и С начинают передачу, причем узел С захватит шину и передает свое сообщение. Если бы узлу А снова надо было передавать сообщение, он снова захватил бы шину. Таким образом, первым на шине CAN передается сообщение с наименьшим идентификатором.

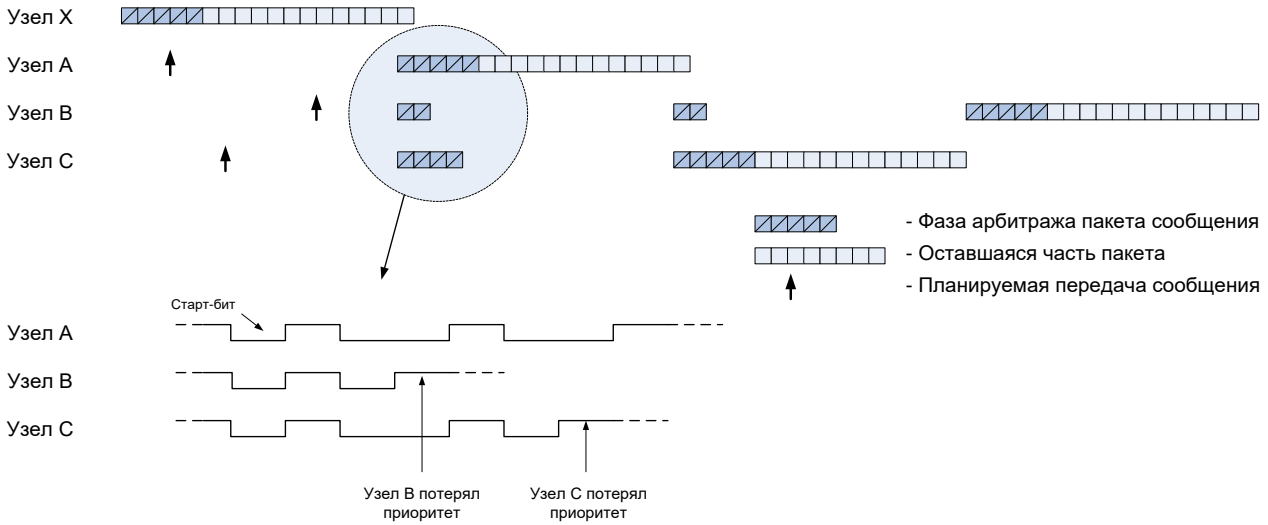


Рисунок 118 – Арбитраж на шине CAN

В случае «проигрыша» арбитража в регистре статуса контроллера CAN будет установлен флаг ID_LOWER.

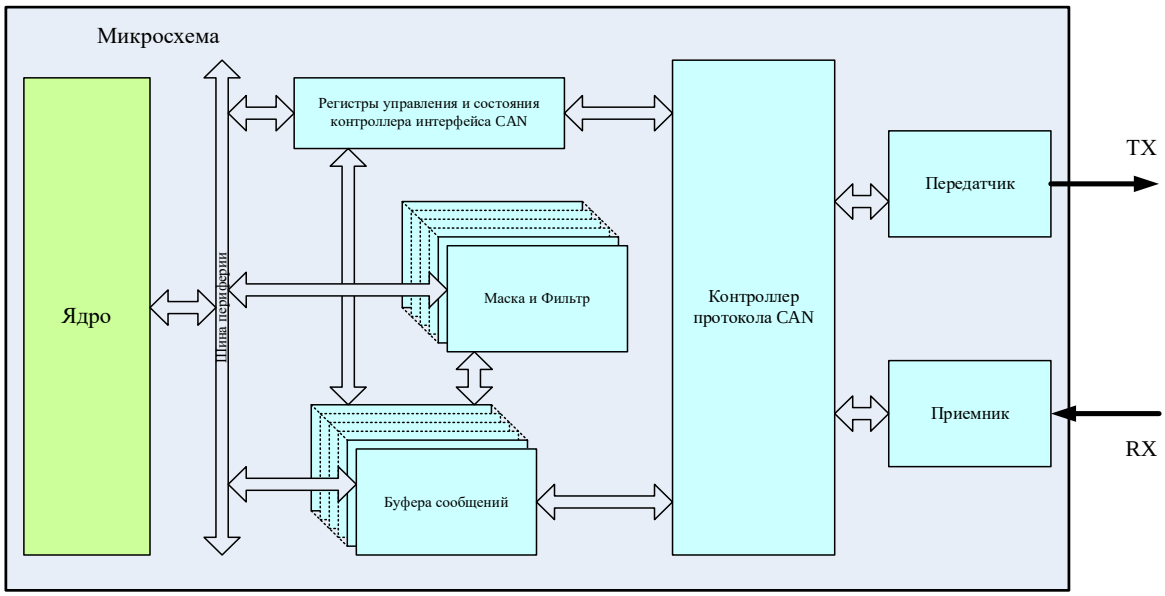


Рисунок 119 – Структурная блок-схема контроллера CAN

7.17.6 Инициализация

Перед началом работы с контроллерами CAN в первую очередь должны быть заданы параметры тактового сигнала.

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 блок таймера 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 бит 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.

7.17.7 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.

7.17.8 Передача сообщений по RemoteTransmitRequest (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. При необходимости в регистре INT_TX 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.

7.17.9 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.

7.17.10 Автоматическая фильтрация принимаемых сообщений

Для уменьшения затрат процессорного ядра на обработку принимаемых сообщений, контроллер CAN интерфейса 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.

7.17.11 Задание скорости передачи и момента сэмплирования

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 с тактовым сигналом 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

$$\mathrm{TBIT} = \frac{1}{\text{Скорость передачи}}. \tag{7}$$

Блок DPLL разбивает битовый интервал на интервалы Time Quanta (TQ). Битовый интервал состоит из четырех частей:

- SynchronizationSegment (Sync_Seg);
- PropagationTimeSegment (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

$$\text{Nominal Bit Time} = \mathrm{TQ} \bullet (\mathrm{Sync_Seg} + \mathrm{PSEG} + \mathrm{SEG1} + \mathrm{SEG2}). \tag{8}$$

Время TQ фиксировано и определяется периодом генератора и программируемым прескалером BRP со значением от 1 до 65536

$$\mathrm{TQ}\ (\mu\mathrm{s}) = \frac{\mathrm{BRP} + 1}{\mathrm{CANCLK0}\ (\text{МГц})} \tag{9}$$

или

$$\mathrm{TQ}\ (\mu\mathrm{s}) = (\mathrm{BRP} + 1) \bullet \mathrm{Tclk}\ (\mu\mathrm{s}). \tag{10}$$

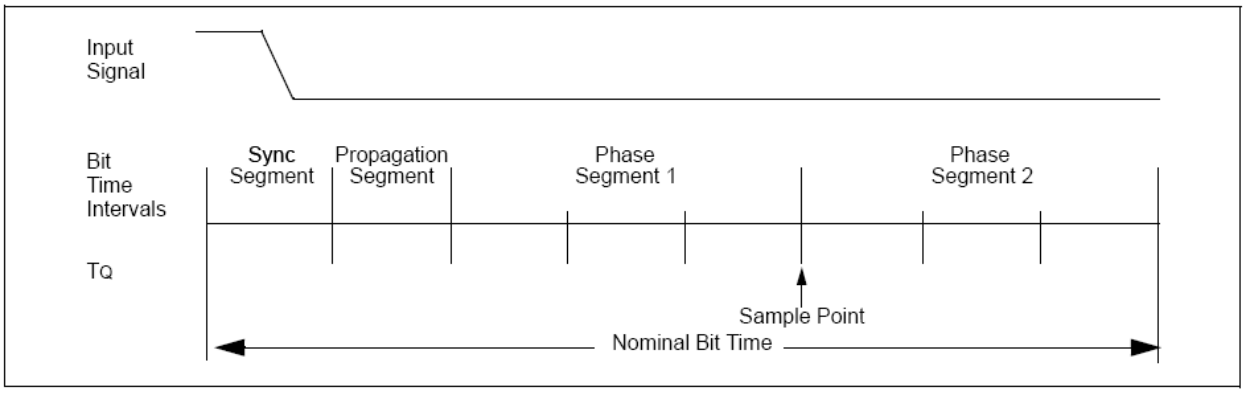


Рисунок 120 – Структура битового интервала

7.17.11.1 **SynchronizationSegment**

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.

7.17.11.2 **PropagationTimeSegment**

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

7.17.11.3 **PhaseBufferSegments**

Эти интервалы предназначены для более точной установки точки сэмплирования, которая располагается между ними. Длительности этих интервалов могут быть запрограммированы между 1 и 8 TQ.

7.17.12 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. 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.

7.17.13 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.

7.17.13.1 **HardSynchronization**

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.

7.17.13.2 **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).

7.17.14 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.

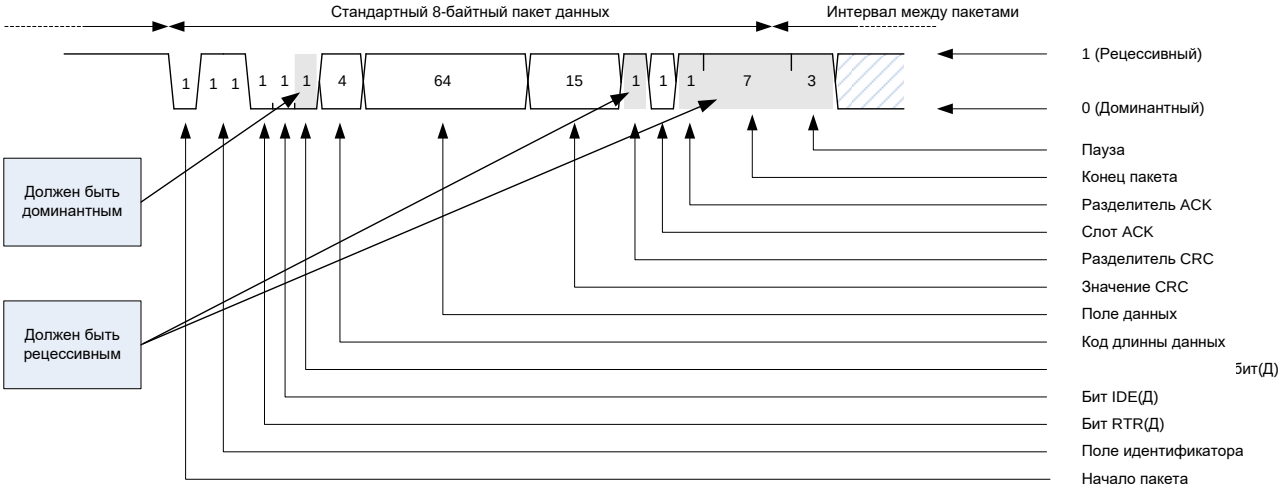


Рисунок 121 – Контроль формата пакета

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.

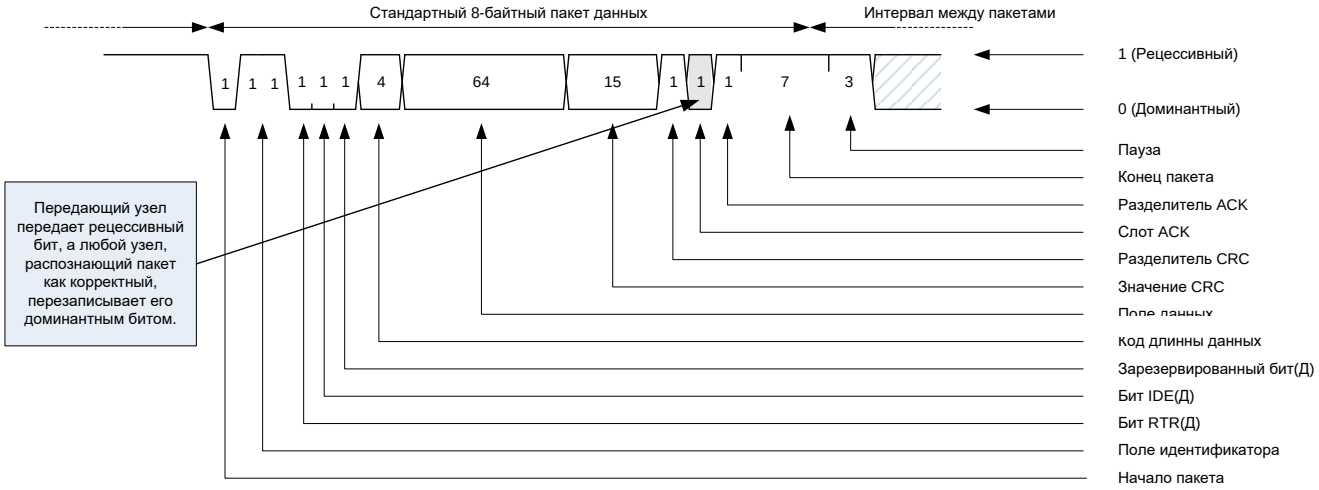


Рисунок 122 – Контроль подтверждения

Пакет сообщения CAN содержит 15-битное 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.

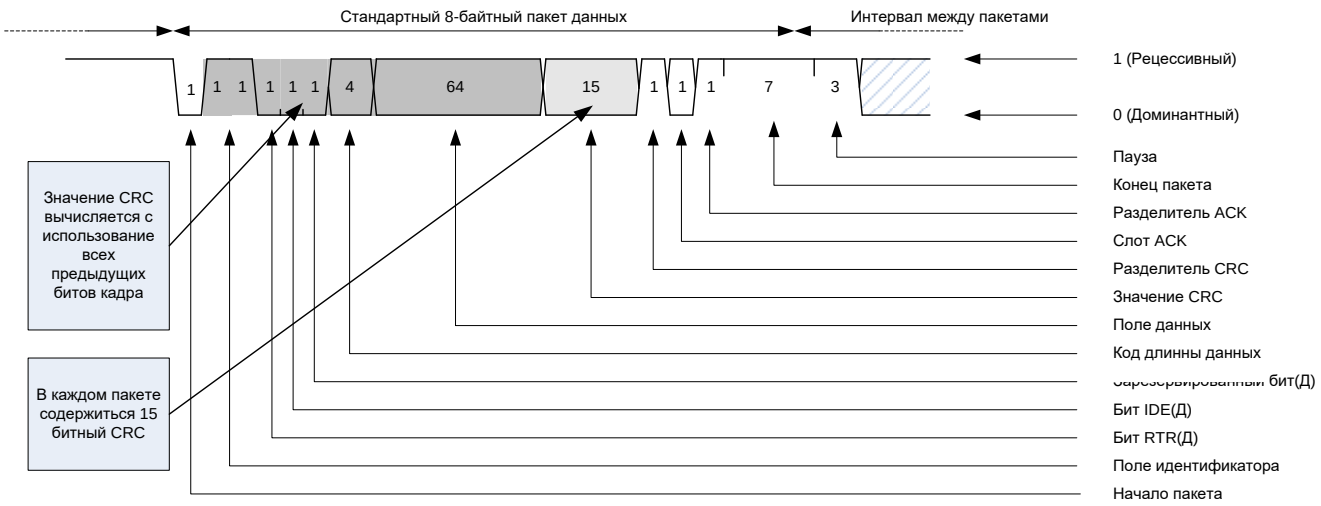


Рисунок 123 – Контроль CRC

После того, как узел выиграет арбитраж, он начинает передачу своего сообщения по шине. Как и во время арбитража, CAN-контроллер 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 устанавливает флаг ошибки передаваемых бит пакета BIT_ERR и сообщение снова ставит в очередь. Это сообщение будет послано в следующем слоте сообщений, однако при этом оно должно пройти через процесс арбитража с другими запланированными сообщениями.

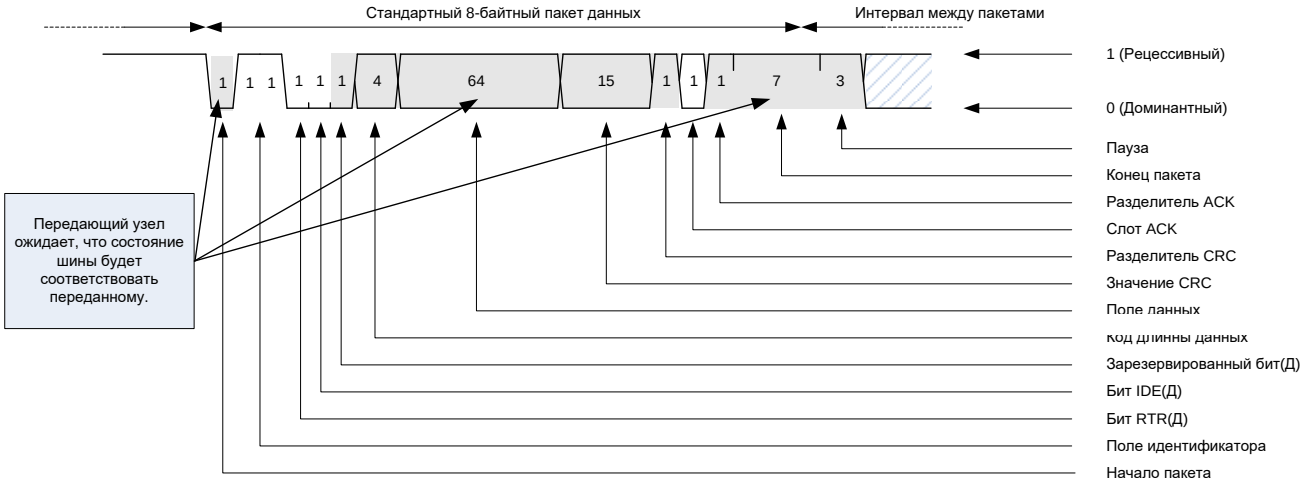


Рисунок 124 – Контроль передаваемых бит

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 бит 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 бит. Это позволяет любому контроллеру CAN формировать на шине сообщение об ошибке сразу после ее обнаружения, не дожидаясь конца сообщения.

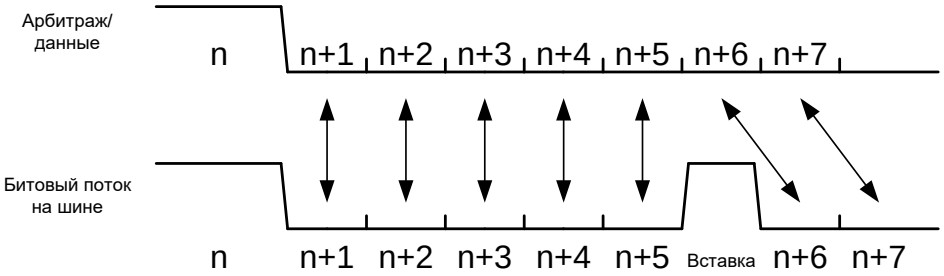


Рисунок 125 – Битстаффинг

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 бит 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 необходимо вмешательство процессора, который повторно инициализирует контроллер и подключает его обратно к шине. Текущий статус состояния контроллера можно посмотреть в регистре статуса контроллера STATUS.

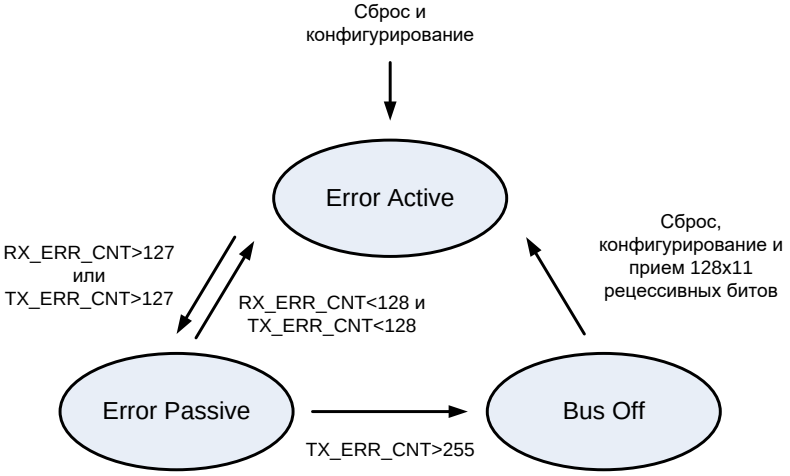


Рисунок 126 – Счетчики ошибок

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.

7.17.15 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, 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.

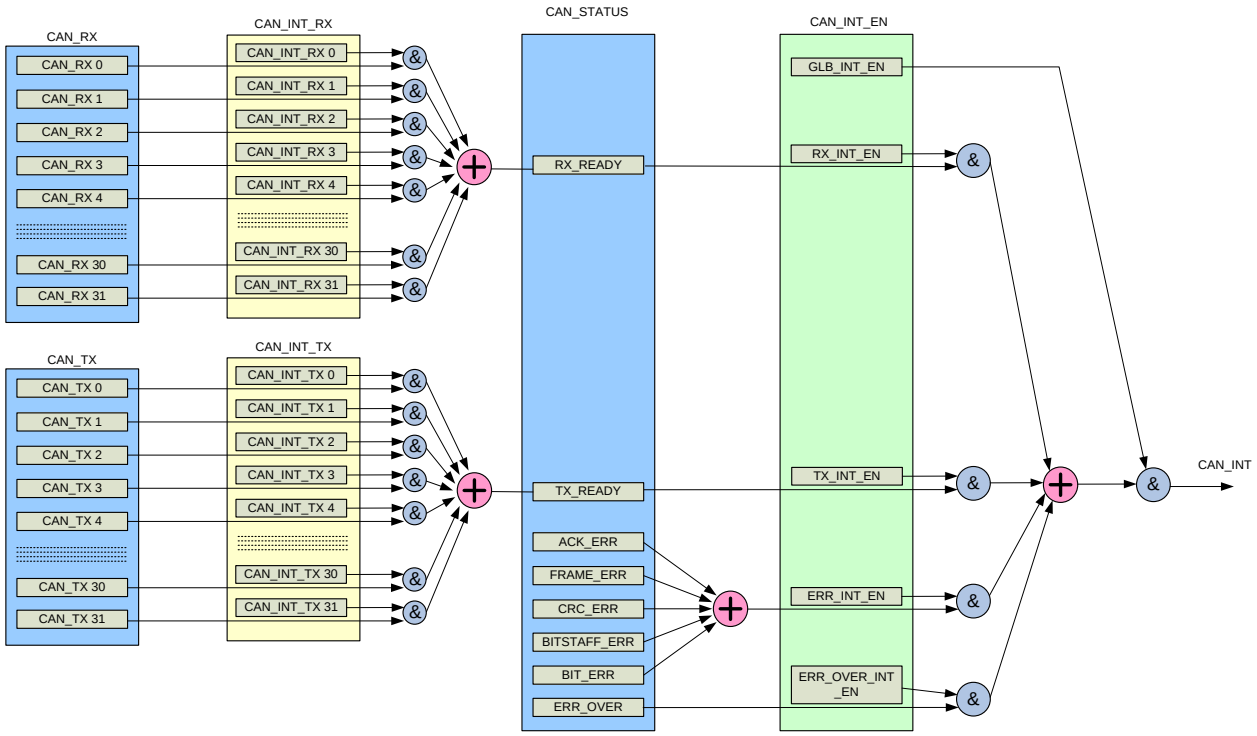


Рисунок 127 – Схема формирования прерывания блока CAN

7.17.16 Описание регистров

Таблица 335 – Описание регистров контроллера CAN

Базовый адрес	Название	Состояние после сброса	Описание
0x4009_0000	MDR_CAN0		Контроллер CAN0
Смещение			
0x0000_0000	CONTROL		Регистр управление контроллером CAN
0x0000_0004	STATUS		Регистр состояния контроллера CAN
0x0000_0008	BITTMNG		Регистр задания скорости работы
0x0000_0010	INT_EN		Регистр разрешения прерываний контроллера
0x0000_001C	OVER		Регистр границы счетчика ошибок
0x0000_0020	RXID		Регистр принятого ID сообщения
0x0000_0024	RXDLC		Регистр принятого DLC сообщения
0x0000_0028	RXDATAL		Регистр принятых данных
0x0000_002C	RXDATAH		Регистр принятых данных
0x0000_0030	TXID		Регистр передаваемого ID сообщения
0x0000_0034	TXDLC		Регистр передаваемого DLC сообщения
0x0000_0038	DATAL		Регистр передаваемых данных
0x0000_003C	DATAH		Регистр передаваемых данных
0x0000_0040	BUF_CON[0]		Регистр управления буфером 00
	…		
0x0000_00BC	BUF_CON[31]		Регистр управления буфером 31
0x0000_00C0	INT_RX		Флаги разрешения прерываний от приемных буферов
0x0000_00C4	RX		Флаги RX_FULL от приемных буферов
0x0000_00C8	INT_TX		Флаги разрешения прерываний от передающих буферов
0x0000_00CC	TX		Флаги ~TX_REQ от передающих буферов
0x0000_0200	BUF[0].ID		ID сообщения буфера 00
0x0000_0204	BUF[0].DLC		DLC сообщения буфера 00
0x0000_0208	BUF[0].DATAL		Данные сообщения буфера 00
0x0000_020C	BUF[0].DATAH		Данные сообщения буфера 00
0x0000_0210	BUF[1].ID		ID сообщения буфера 01
	…		
0x0000_03FC	BUF[31].DATAH		Данные сообщения буфера 31

Базовый адрес	Название	Состояние после сброса	Описание
0x0000_0400-0x0000_07FF			Зарезервировано
0x0000_0500	FILTER[0].MASK		Маска для приема сообщения в буфер 00
0x0000_0504	FILTER[0].FILTER		Фильтр для приема сообщения в буфер 00
0x0000_0508	FILTER[1].MASK		Маска для приема сообщения в буфер 01
	...		
0x0000_05FC	FILTER[31].FILTER		Фильтр для приема сообщения в буфер 31
Примечание – Регистры RXID, RXDLC, RXDATAL/H (0x20-0x2C) и TXID, TXDLC, DATAL/H (0x30 – 0x3C) предназначены для временного хранения данных принятого или передаваемого пакета, соответственно. Использовать только для отладки			

7.17.16.1 **CONTROL**

Base ADDR=		0x4009_0000			Offset=		0x0000_0000								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-											ROP	SAP	STM	ROM	CANEN

Таблица 336 – Описание бит регистра CONTROL

Бит	Имя	Значение	Описание
31...5	-		Зарезервировано
4	ROP	0	Прием собственных пакетов (Receiveownpackets): 1 – контроллер принимает собственные пакеты; 0 – контроллер принимает только чужие пакеты
3	SAP	0	Подтверждение собственных пакетов (SendACKonownpackets): 1 – контроллер подтверждает прием собственных пакетов; 0 – контроллер подтверждает прием только чужих пакетов
2	STM	0	Режим самотестирования (SelfTestMode): 1 – контроллер работает в режиме самотестирования; 0 – контроллер работает в нормальном режиме
1	ROM	0	Режим «Только прием» (ReadOnlyMode): 1 – контроллер работает только на прием; 0 – контроллер работает в нормальном режиме
0	CAN_EN	0	Режим работы контроллера CAN: 1 – разрешение работы; 0 – сброс

7.17.16.2 **STATUS**

Base ADDR=		0x4009_0000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXERRCNT [7:0]								RXERRCNT [7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-			TXERRCNT8	RXERRCNT8	ERRSTATUS [1:0]		IDLOWER	ACKERR	FRAMEERR	CRCERR	BITSTUFF ERR	BITERR	ERROR OVER	TX READY	RX READY

Таблица 337 – Описание бит регистра STATUS

Бит	Имя	Значение	Описание
31...24	TXERRCNT[7:0]	0x00	Счетчик ошибок передатчика TEC, биты [7:0]: TEC > 127, ERROR PASSIVE
23…16	RXERRCNT[7:0]	0x00	Счетчик ошибок приемника REC, биты [7:0]: REC > 127, ERROR PASSIVE
15…13	-		Зарезервировано
12	TXERRCNT8	0	Счетчик ошибок передатчика TEC, бит 8: 0 – TEC менее 255; 1 – TEC более 255
11	RXERRCNT8	0	Счетчик ошибок приемника REC, бит 8: 0 – REC менее 255; 1 – REC более 255
10…9	ERRSTATUS[1:0]	00	Статус состояния контроллера CAN: 00 – ERRORACTIVE, при возникновении ошибки отсылается флаг активной ошибки; 01 – ERRORPASSIVE, при возникновении ошибки отсылается флаг пассивной ошибки; 1x – BUSOFF, ожидается восстановление шины
8	ID LOWER	0	Флаг «проигрыша» арбитража: 0 – при передаче не было проигрыша арбитража; 1 – при передаче был проигран арбитраж
7	ACK ERR	0	Флаг ошибки подтверждения приема: 0 – нет ошибки; 1 – есть ошибка. Сбрасывается записью «0»
6	FRAME ERR	0	Флаг ошибки формата пакета: 0 – нет ошибки; 1 – есть ошибка. Сбрасывается записью «0»

Бит	Имя	Значение	Описание
5	CRC ERR	0	Флаг ошибки контрольной суммы пакета: 0 – нет ошибки; 1 – есть ошибка. Сбрасывается записью «0»
4	BIT STUFF ERR	0	Флаг ошибки вставленных бит пакета: 0 – нет ошибки; 1 – есть ошибка. Сбрасывается записью «0»
3	BIT ERR	0	Флаг ошибки передаваемых бит пакета: 0 – нет ошибки; 1 – есть ошибка. Сбрасывается записью «0»
2	ERROR OVER	0	Флаг превышения TEC и REC уровня, заданного ERROR_MAX: 0 – ERROR_MAX ≤ TEC и REC; 1 – ERROR_MAX > TEC или REC. Сбрасывается записью «0»
1	TX READY	0	Флаг наличия данных в буфере для отправки: 0 – буферы пусты, нет сообщений готовых к передаче; 1 – есть не пустой буфер, в котором сообщение готово к передаче. 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 буфера в регистрах CAN_INT_TX
0	RX READY	0	Флаг наличия принятых сообщений: 0 – буферы пусты, нет принятых сообщений; 1 – есть буфер, содержащий принятые сообщения. 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 прием буфера в регистрах CAN_INT_RX
Примечание - Для организации надежного сброса статусных флагов прерываний необходимо учитывать требования, изложенные в подразделе 9.3			

7.17.16.3 **BITTMNG**

Base ADDR=		0x4009_0000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-				SB	SJW[1:0]		SEG2[2:0]			SEG1[2:0]			PSEG[2:0]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRP[15:0]															

Таблица 338 – Описание бит регистра BITTMNG

Бит	Имя	Значение	Описание
31...28	-		Зарезервировано
27	SB	0	Сэмплирование: 0 – однократное; 1 – трехкратное с мажоритарным контролем
26…25	SJW [1:0]	00	Значение размера фазы SJW: 11 = Synchronization jump width time = 4 × TQ; 10 = Synchronization jump width time = 3 × TQ; 01 = Synchronization jump width time = 2 × TQ; 00 = Synchronization jump width time = 1 × TQ. 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]	000	Значение размера фазы SEG2: 111 = Phase Segment 2 time = 8 × TQ; 110 = Phase Segment 2 time = 7 × TQ; 101 = Phase Segment 2 time = 6 × TQ; 100 = Phase Segment 2 time = 5 × TQ; 011 = Phase Segment 2 time = 4 × TQ; 010 = Phase Segment 2 time = 3 × TQ; 001 = Phase Segment 2 time = 2 × TQ; 000 = Phase Segment 2 time = 1 × TQ. SEG2 – это время, используемое для сокращения битового интервала при подстройке
21…19	SEG1 [2:0]	000	Значение размера фазы SEG1: 111 = Phase Segment 1 time = 8 × TQ; 110 = Phase Segment 1 time = 7 × TQ; 101 = Phase Segment 1 time = 6 × TQ; 100 = Phase Segment 1 time = 5 × TQ; 011 = Phase Segment 1 time = 4 × TQ; 010 = Phase Segment 1 time = 3 × TQ; 001 = Phase Segment 1 time = 2 × TQ; 000 = Phase Segment 1 time = 1 × TQ. SEG1 – это время, используемое для увеличения битового интервала при подстройке

Бит	Имя	Значение	Описание
18…16	PSEG[2:0]	000	Значение размера фазы PSEG: 111 = Propagation time = 8 × TQ; 110 = Propagation time = 7 × TQ; 101 = Propagation time = 6 × TQ; 100 = Propagation time = 5 × TQ; 011 = Propagation time = 4 × TQ; 010 = Propagation time = 3 × TQ; 001 = Propagation time = 2 × TQ; 000 = Propagation time = 1 × TQ. PSEG – это время, компенсирующее задержку распространения сигналов в шине CAN
15…0	BRP [15:0]	0x0000	Предделитель системной частоты: CLK = CANCLK0/(BRP + 1); TQ (μs) = 1/CLK (МГц) = (BRP + 1)/CANCLK0 (МГц)

7.17.16.4 **INT_EN**

Base ADDR=		0x4009_0000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-											ERROVER INTN	ERRINTEN	TXINTEN	RXINTEN	GLBINTEN

Таблица 339 – Описание бит регистра INT_EN

Бит	Имя	Значение	Описание
31...5	-		Зарезервировано
4	ERROVER INTEN	0	Флаг разрешения прерывания по превышению TEC или REC допустимого значения в ERROR_MAX: 0 – запрещено прерывание; 1 – разрешено прерывание
3	ERRINT EN	0	Флаг разрешения прерывания по возникновению ошибки: 0 – запрещено прерывание; 1 – разрешено прерывание
2	TXINT EN	0	Флаг разрешения прерывания по возможности передачи: 0 – запрещено прерывание; 1 – разрешено прерывание
1	RXINT EN	0	Флаг разрешения прерывания по приему сообщений: 0 – запрещено прерывание; 1 – разрешено прерывание
0	GLBINT EN	0	Общий флаг разрешения прерывания блока CAN: 0 – запрещено прерывание; 1 – разрешено прерывание

7.17.16.5 **OVER**

Base ADDR=		0x4009_0000			Offset=		0x0000_001C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ERROR_MAX[7:0]							

Таблица 340 – Описание бит регистра OVER

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7…0	ERRORMAX[7:0]	0x00	Регистр границы счетчика ошибок. Допустимое значение счетчиков ошибок TEC и REC, при превышении которого вырабатывается флаг ERROR_OVER

7.17.16.6 **BUF_CON[x]**

Base ADDR=		0x4009_0000			Offset=		0x0000_0040 .. 0x0000_00BC								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								OVERWR	RXFULL	TXREQ	PRIOR_0	RTREN	OVEREN	RXTXn	EN

Таблица 341 – Описание бит регистров BUF_CON[x]

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7	OVER_WR	0	Флаг перезаписи принятого сообщения: 0 – не было перезаписи; 1 – была перезапись принятого сообщения
6	RX_FULL	0	Флаг готовности приема: 0 – нет принятого сообщения; 1 – принятое сообщение в буфере
5	TX_REQ	0	Запрос на отправку сообщения: 0 – нет запроса или отправлено; 1 – запрос на отправку

Бит	Имя	Значение	Описание
4	PRIOR_0	0	Приоритет при отправке: 0 – приоритет; 1 – нет приоритета
3	RTR_EN	0	Режим ответа на RTR: 0 – не отвечать при приеме RTR; 1 – ответить при приеме RTR в буфер
2	OVER_EN	0	Разрешение перезаписи принятого сообщения: 0 – не разрешена перезапись; 1 – разрешена перезапись сообщения
1	RX_TXn	0	Режим работы буфера: 0 – на передачу; 1 – на прием
0	EN	0	Разрешение работы буфера: 0 – отключен; 1 – работает

7.17.16.7 **INT_RX**

Base ADDR=		0x4009_0000			Offset=		0x0000_00C0								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAN_INT_RX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN_INT_RX[15:0]															

Таблица 342 – Описание бит регистра INT_RX

Бит	Имя	Значение	Описание
31...0	CAN_INT_RX[31:0]	0x00000000	Флаги разрешения прерываний от буферов по приему сообщений: 0 – прерывание запрещено; 1 – прерывание разрешено. CAN_INT_RX[0] – для первого буфера; CAN_INT_RX[1] – для второго буфера и так далее

7.17.16.8 **RX**

Base ADDR=		0x4009_0000			Offset=		0x0000_00C4								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAN_RX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN_RX[15:0]															

Таблица 343 – Описание бит регистра RX

Бит	Имя	Значение	Описание
31...0	CAN_RX[31:0]	0x00000000	Флаги RX_FULL разрешенных на прием буферов: CAN_RX[0] – флаг RX_FULL от первого буфера; CAN_RX[1] – флаг RX_FULL от второго буфера и так далее

7.17.16.9 **INT_TX**

Base ADDR=		0x4009_0000			Offset=		0x0000_00C8								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAN_INT_TX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN_INT_TX[15:0]															

Таблица 344 – Описание бит регистра INT_TX

Бит	Имя	Значение	Описание
31...0	CAN_INT_TX[31:0]	0x00000000	Флаги разрешения прерываний от буферов по передаче сообщений: CAN_INT_TX[0] – для первого буфера; CAN_INT_TX[1] – для второго буфера и так далее

7.17.16.10 **TX**

Base ADDR=		0x4009_0000			Offset=		0x0000_00CC								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAN_TX[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN_TX[15:0]															

Таблица 345 – Описание бит регистра TX

Бит	Имя	Значение	Описание
31...0	CAN_TX[31:0]	0x00000000	Флаги ~TX_REQ разрешенных на передачу буферов: CAN_TX[0] – флаг ~TX_REQ от первого буфера; CAN_TX[1] – флаг ~TX_REQ от второго буфера и так далее, доступны только на чтение

7.17.16.11 **RXID**
TXID

BUF[x].ID
FILTER[x].MASK
FILTER[x].FILTER

Base ADDR=		0x4009_0000			Offset=		0x0000_0020 0x0000_0030 0x0000_0200 0x0000_0210 .. 0x0000_03F0 0x0000_0500 0x0000_0504 .. 0x0000_05FC								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-			SID[10:0]											EID[17:16]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EID[15:0]															

Таблица 346 – Описание бит регистров RXID, TXID, BUF[x].ID, FILTER[x].MASK, FILTER[x].FILTER

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28…18	SID [10:0]	0x000*	Поле SID. Для стандартного и расширенного пакетов CAN. Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже
17…0	EID [17:0]	0x00000*	Поле EID. Для расширенных пакетов CAN. Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже
* Регистры BUF[x].ID FILTER[x].MASK FILTER[x].FILTER не инициализируются сбросом			

7.17.16.12 RXDLC TXDLC BUF[x].DLC

Base ADDR=		0x4009_0000			Offset=		0x0000_0024 0x0000_0034 0x0000_0204 0x0000_0214 .. 0x0000_03F4								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-			IDE	SRR	R0	R1	RTR	-				DLC[3:0]			

Таблица 347 – Описание бит регистров RXDLC, TXDLC, BUF[x].DLC

Бит	Имя	Значение	Описание
31...13	-	0*	Зарезервировано
12	IDE	0*	Поле IDE. Поле, обозначающее формат пакета: 0 – стандартный пакет; 1 – расширенный пакет
11	SRR	0*	Поле SRR, расширенного формата. Всегда должен быть «1»
10	R0	0*	Поле R0. Всегда должен быть «0»
9	R1	0*	Поле R1, расширенного формата. Всегда должен быть «0» (в соответствии со стандартом)
8	RTR	0*	Поле RTR, запроса обратного ответа: 0 – нет запроса; 1 – есть запрос. Если узел получил пакет с запросом обратного ответа, то он должен ответить
7…4	-	0*	Зарезервировано
3…0	DLC[3:0]	0000*	Поле DLC, длина передаваемых данных в пакете: 0000 – нет данных; 0001 – 1 байт; 0010 – 2 байт; 0011 – 3 байт; 0100 – 4 байт; 0101 – 5 байт;

Бит	Имя	Значение	Описание
			0110 – 6 байт; 0111 – 7 байт; 1000 – 8 байт; 1xxx – 8 байт и недопустимо
* Регистры BUF[x].DLC не инициализируются сбросом			

7.17.16.13 RXDATAL TXDATAL BUF[x].DATAL

Base ADDR=		0x4009_0000			Offset=		0x0000_0028 0x0000_0038 0x0000_0208 0x0000_0218 .. 0x0000_03F8								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB3[7:0]								DB2[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							

Таблица 348 – Описание бит регистров RXDATAL, TXDATAL, BUF[x].DATAL

Бит	Имя	Значение	Описание
31…24	DB3[7:0]	0x00*	Поле DB3. Четвертый байт, передаваемый в пакете
23…16	DB2[7:0]	0x00*	Поле DB2. Третий байт, передаваемый в пакете
15…8	DB1[7:0]	0x00*	Поле DB1. Второй байт, передаваемый в пакете
7…0	DB0[7:0]	0x00*	Поле DB0. Первый байт, передаваемый в пакете
* Регистры BUF[x].DATAL не инициализируются сбросом			

7.17.16.14 RXDATAH TXDATAH BUF[x].DATAH

Base ADDR=		0x4009_0000			Offset=		0x0000_002C 0x0000_003C 0x0000_020C 0x0000_021C .. 0x0000_03FC								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB7[7:0]								DB6[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							

Таблица 349 – Описание бит регистров RXDATAH, TXDATAH, BUF[x].DATAH

Бит	Имя	Значение	Описание
31…24	DB7[7:0]	0x00*	Поле DB7. Четвертый байт, передаваемый в пакете
23…16	DB6[7:0]	0x00*	Поле DB5. Третий байт, передаваемый в пакете
15…8	DB5[7:0]	0x00*	Поле DB4. Второй байт, передаваемый в пакете
7…0	DB4[7:0]	0x00*	Поле DB3. Первый байт, передаваемый в пакете
* Регистры BUF[x].DATAH не инициализируются сбросом			

7.17.16.15 BUF[x]

Base ADDR=		0x4009_0000			Offset=		0x0000_0200 0x0000_0210 .. 0x0000_03F0								
REG Name:															
127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
DATAH[31:16]															

111	100	109	108	107	106	105	104	103	102	101	100	99	98	97	96
DATAH[15:0]															

95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
DATAL[31:16]															

79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
DATAL[15:0]															

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
DLC[31:16]															

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DLC[15:0]															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															

Таблица 350 – Описание бит регистров BUF[x]

Бит	Имя	Значение	Описание
127…96	DATAH[31:0]		Данные сообщения
95…64	DATAL[31:0]		Данные сообщения
63…32	DLC[31:0]		DLC сообщения
31…0	ID[31:0]		ID сообщения

7.17.16.16 **FILTER[x]**

Base ADDR=	0x4009_0000	Offset=	0x0000_0500 0x0000_0508 .. 0x0000_05F8						
REG Name:									

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
FILTER[31:16]															

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
FILTER[15:0]															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK[15:0]															

Таблица 351 – Описание бит регистров FILTER[x]

Бит	Имя	Значение	Описание
63…32	FILTER[31:0]		Фильтр для приема сообщения
31…0	MASK[31:0]		Маска для приема сообщения

7.18 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.

Основные характеристики модуля SSP:

– функционирование как в ведущем, так и в ведомом режиме;
– программное управление скоростью обмена;
– состоит из независимых буферов приема и передачи (8 ячеек по 16 бит) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел);
– программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI;
– программируемая длительность информационного кадра от 4 до 16 бит;
– независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по переполнению буфера приемника;
– доступна возможность тестирования по шлейфу, соединяющему вход с выходом;
– поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена на рисунке 128.

7.18.1 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

– режим функционирования периферийного устройства – ведущее или ведомое;
– разрешение или запрещение функционирования;

– формат информационного кадра;
– скорость передачи данных;
– фаза и полярность тактового сигнала;
– размер блока данных – от 4 до 16 бит;
– маскирование прерываний.

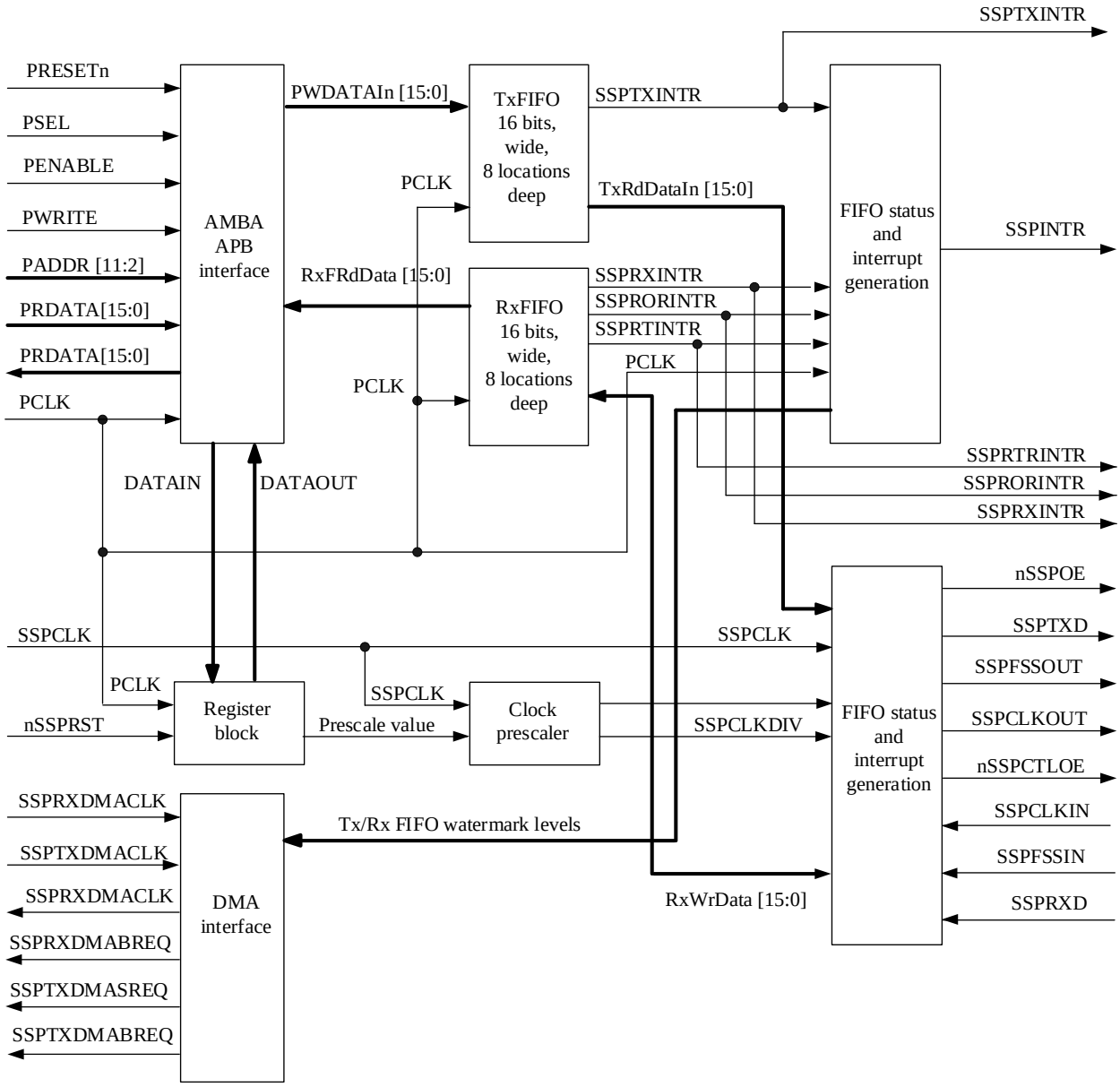


Рисунок 128 – Структурная схема модуля SSP

7.18.1.1 **Характеристики интерфейса SPI**

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:
– полнодуплексный обмен данными по четырехпроводной линии;
– программное задание фазы и полярности тактового сигнала.

7.18.1.2 **Характеристики интерфейса Microwire**

Интерфейс Microwire фирмы National Semiconductor обеспечивает полудуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

7.18.1.3 **Характеристики интерфейса SSI**

Интерфейс SSI фирмы Texas Instruments обеспечивает:

– полнодуплексный обмен данными по четырехпроводной линии;

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

7.18.2 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).

7.18.2.1 **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.

7.18.2.2 **Буфер FIFO передатчика**

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

7.18.2.3 **Буфер FIFO приемника**

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

7.18.2.4 **Блок приема и передачи данных**

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

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

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

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

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

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

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

7.18.2.5 **Блок формирования прерываний**

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

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

7.18.2.6 **Интерфейс прямого доступа к памяти**

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

7.18.2.7 **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.

7.18.2.8 **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.

7.18.2.9 **Соотношения между тактовыми сигналами**

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

$$\mathrm{FSSPCLK} \leq \mathrm{FPCLK}.$$

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:

$$\begin{cases} FSSPCLK_{min} \geq 2 \bullet FSSPCLKOUT_{max}\ (master\ mode) \\ FSSPCLK_{min} \geq 12 \bullet FSSPCLKIN_{max}\ (slave\ mode)\ . \end{cases} \quad (11)$$

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

$$\begin{cases} FSSPCLK_{max} \leq 254 \bullet 256 \bullet FSSPCLKOUT_{min}\ (master\ mode) \\ FSSPCLK_{max} \leq 254 \bullet 256 \bullet FSSPCLKIN_{min}\ (slave\ mode)\ . \end{cases} \quad (12)$$

7.18.2.10 **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).

7.18.2.11 **Программирование регистра управления CR1**

Регистр SSPCR1 предназначен для:

– выбора ведущего или ведомого режима функционирования приемо-передатчика;
– 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).

7.18.2.12 **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

$$FSSPCLKOUT = \frac{FSSPCLK}{CPSDVR \bullet (1 + SCR)}. \quad (13)$$

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.

7.18.2.13 **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.

7.18.2.14 **Формат синхронного обмена SSI фирмы TexasInstruments**

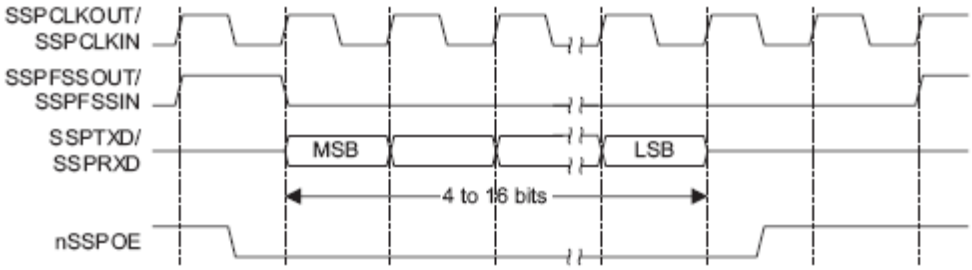


Рисунок 129 – Формат синхронного обмена протокола SSI (единичный обмен)

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. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSP_CLK, старший значащий разряд информационного кадра (4 – 16 бит данных) выдается на выход линии SSP_TXD и т.д.

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

Временные диаграммы последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments представлены на рисунках: рисунок 129 – передача единичного информационного кадра, рисунок 130 – передача последовательности кадров.

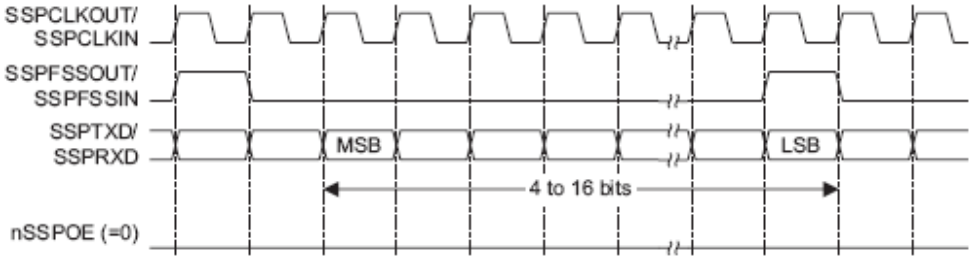


Рисунок 130 – Формат синхронного обмена протокола SSI (непрерывный обмен)

7.18.2.15 **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.

7.18.2.16 **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 131 – одиночный обмен, рисунок 132 – непрерывный обмен.

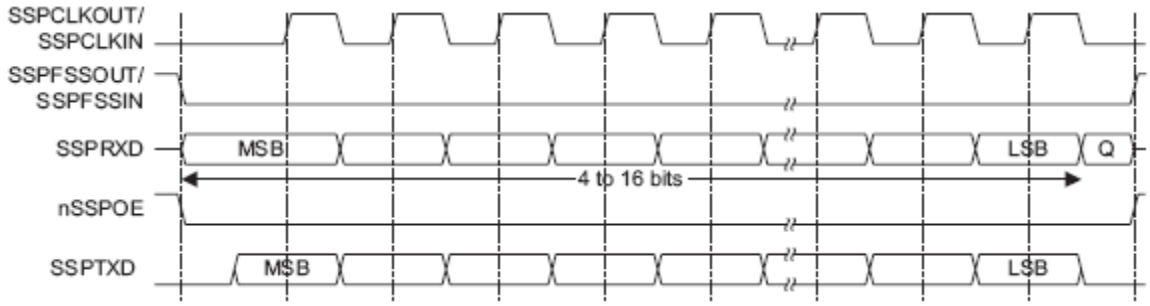


Рисунок 131 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (одиночный обмен)

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

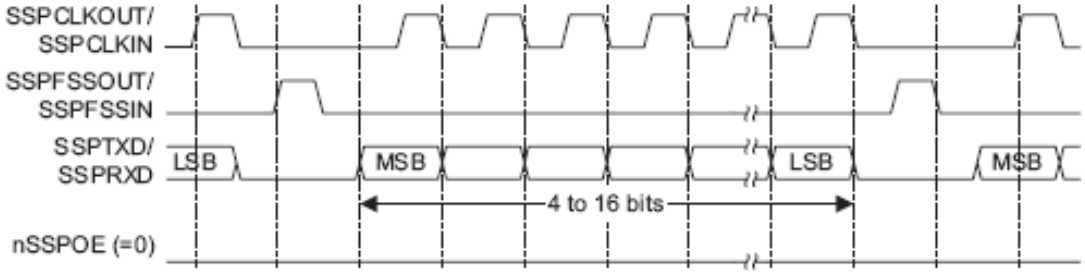


Рисунок 132 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (непрерывный обмен)

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.

7.18.2.17 **Формат синхронного обмена SPI фирмы Motorola, SPO=0, SPH=1**

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=0, SPH=1 показывает рисунок 133 – одиночный и непрерывный обмен.

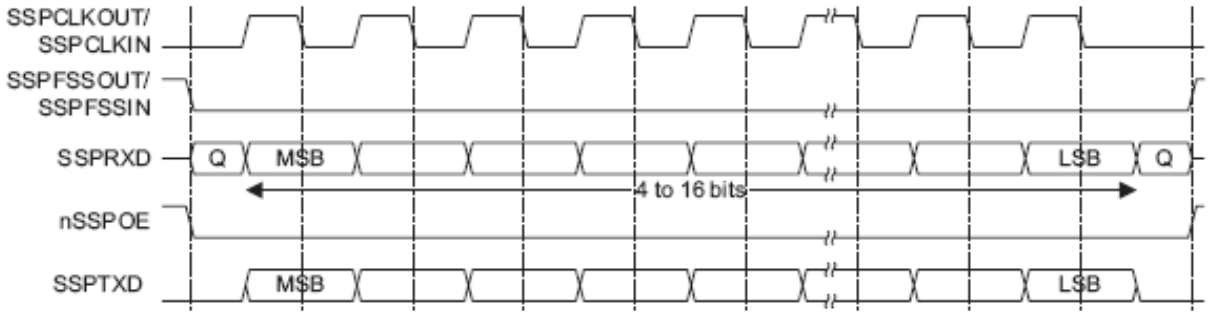


Рисунок 133 – Формат синхронного обмена протокола SPI, SPO=0, SPH=1

Примечание – На рисунке 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.

7.18.2.18 **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 134 – одиночный обмен и рисунок 135 – непрерывный обмен.

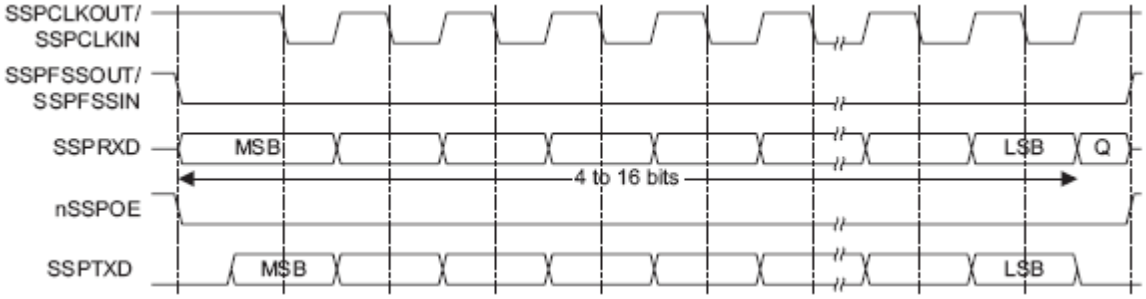


Рисунок 134 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (одиночный обмен)

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

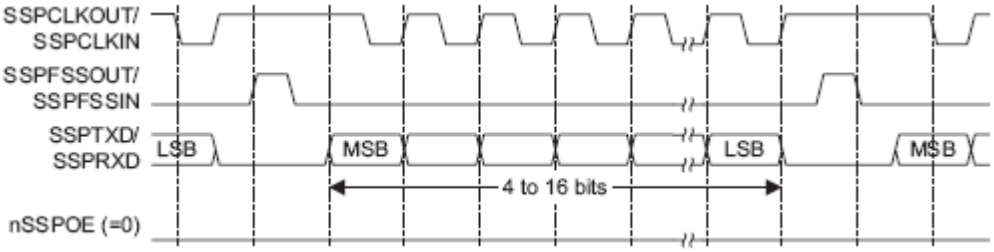


Рисунок 135 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (непрерывный обмен)

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.

7.18.2.19 **Формат синхронного обмена SPI фирмы Motorola, SPO=1, SPH=1**

Временные диаграммы последовательного синхронного обмена в режиме SPI с SPO=1, SPH=1 показывает рисунок 136 – одиночный и непрерывный обмен.

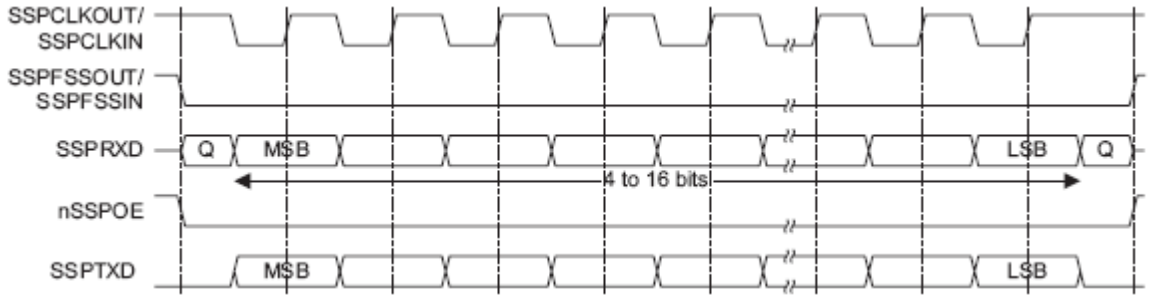


Рисунок 136 – Формат синхронного обмена протокола SPI, SPO=1, SPH=1

Примечание – На рисунке 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 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.

7.18.2.20 **Формат синхронного обмена Microwire фирмы NationalSemiconductor**

Временные диаграммы последовательного синхронного обмена в режиме Microwire показаны на рисунках: рисунок 137 – одиночный обмен, рисунок 138 – непрерывный обмен.

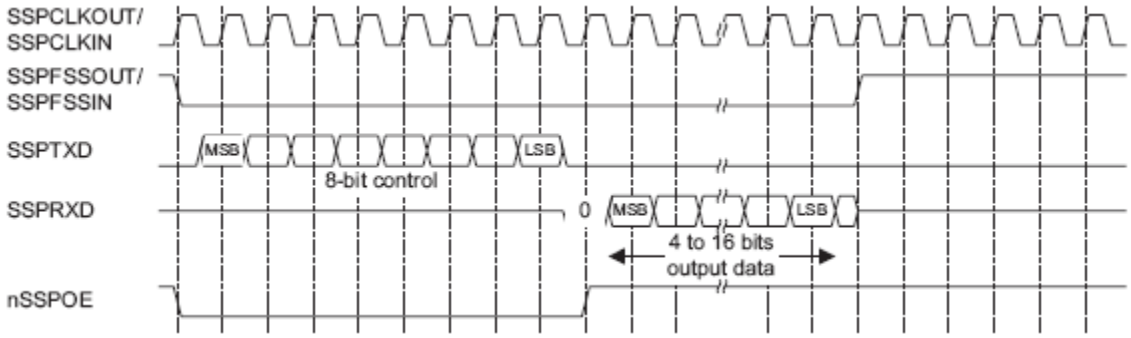


Рисунок 137 – Формат синхронного обмена протокола Microwire (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен в нем осуществляется в полудуплексном режиме, с использованием служебных последовательностей. Каждая информационный обмен начинается с передачи ведущим устройством специальной 8-ми битной управляющей последовательности. В течение всего времени ее передачи приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит, таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

В данном режиме во время ожидания приемопередатчика:

– сигнал 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.

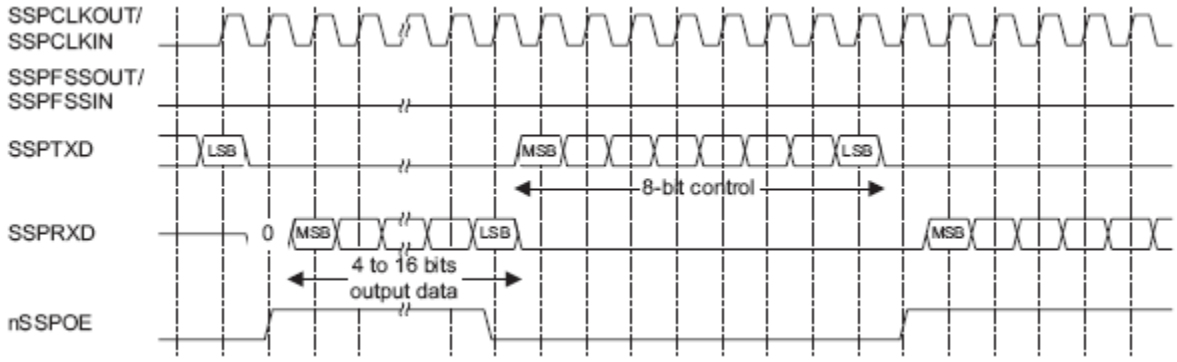


Рисунок 138 – Формат синхронного обмена протокола Microwire (непрерывный обмен)

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 139. 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.

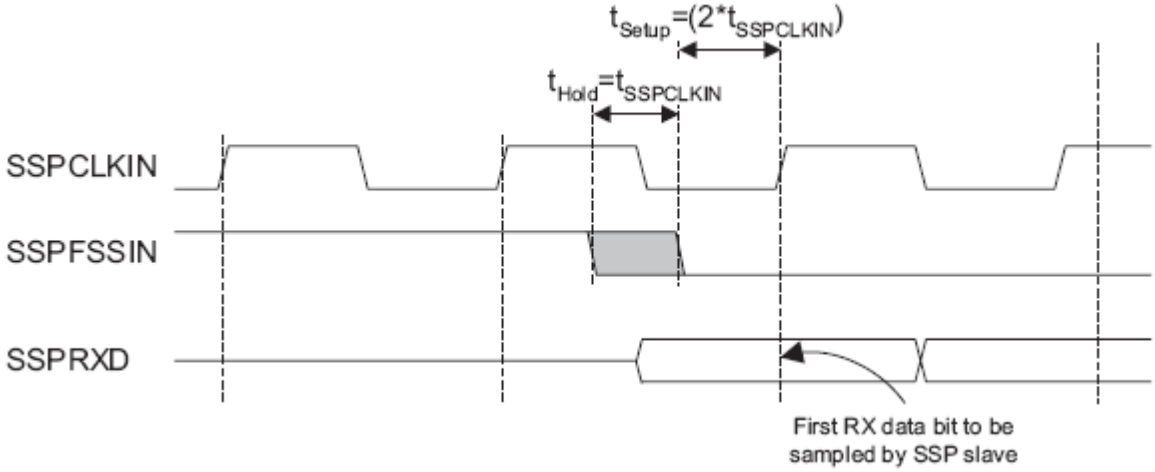


Рисунок 139 – Формат Microwire, требования к времени установки и удержания сигнала

7.18.2.21 **Примеры конфигурации модуля в ведущем и ведомом режимах**

На рисунках 140, 141 и 142 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.

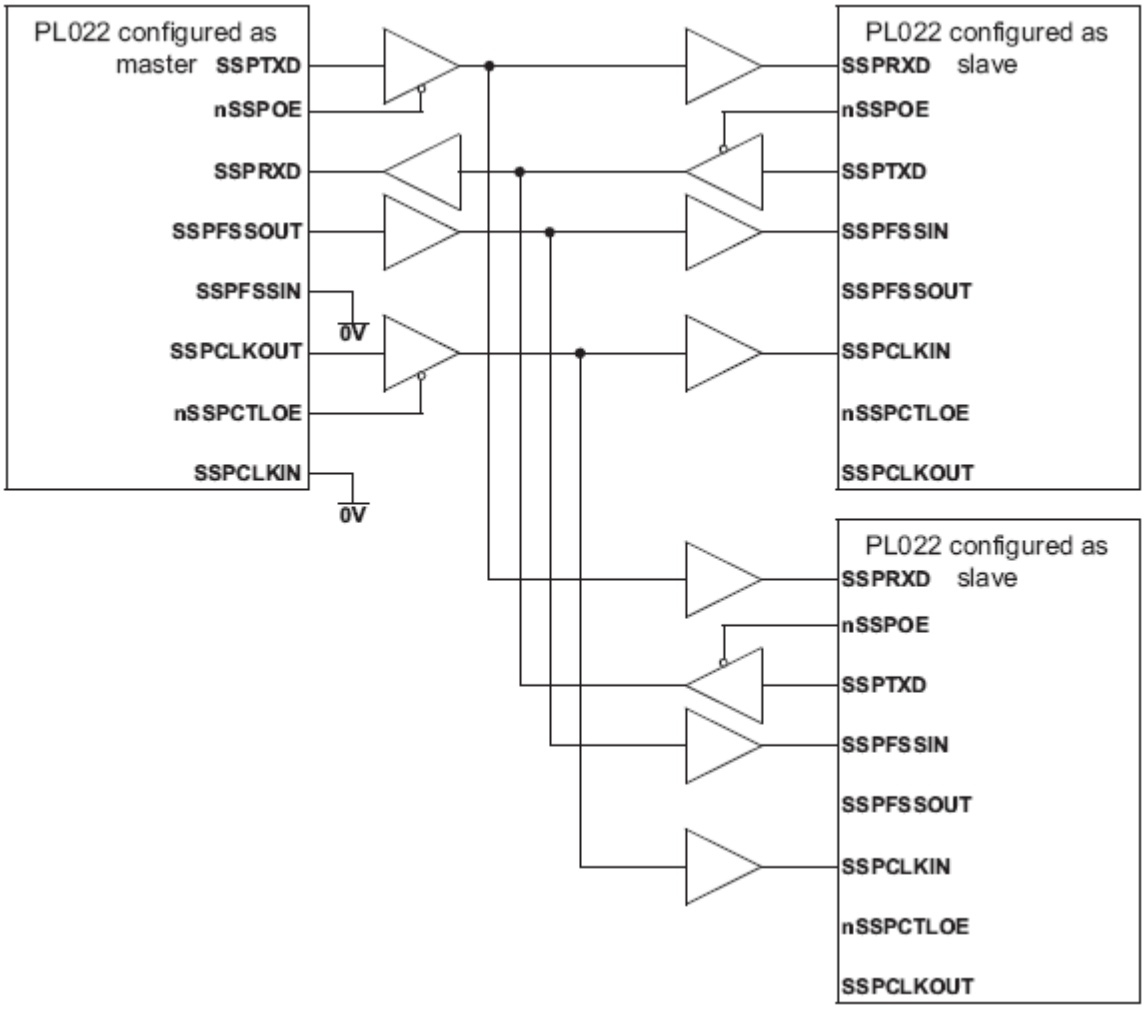


Рисунок 140 – Ведущее устройство SSP подключено к двум ведомым

Рисунок 140 показывает совместную работу трех модулей SSP, один из которых сконфигурирован в качестве ведущего, а два – в качестве ведомых устройств. Ведущее устройство способно передавать данные по кругу 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.

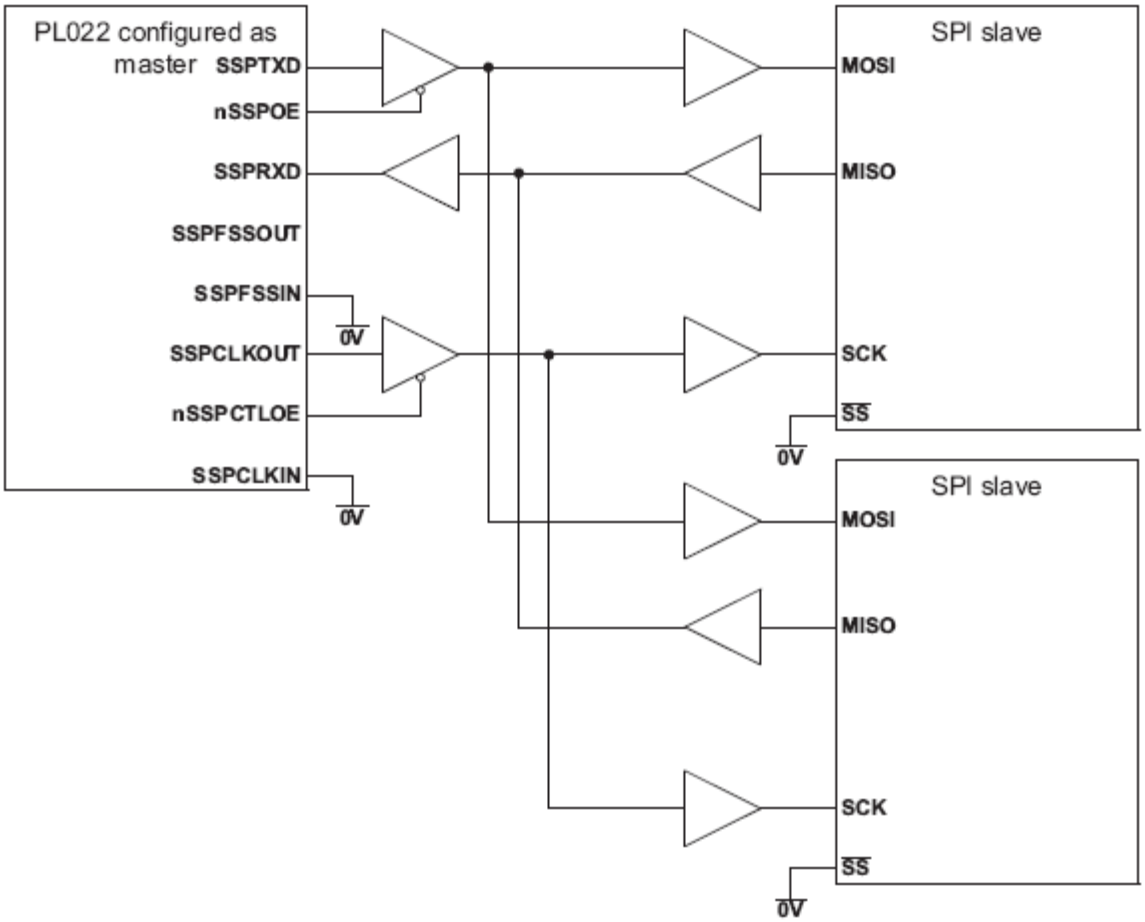


Рисунок 141 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим SPI

Рисунок 141 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 по кругу по линии SSP_TXD. Ответная передача данных на входную линию SSP_RXD ведущего устройства, одновременно осуществляется только одним из ведомых по соответствующей линии MISO.

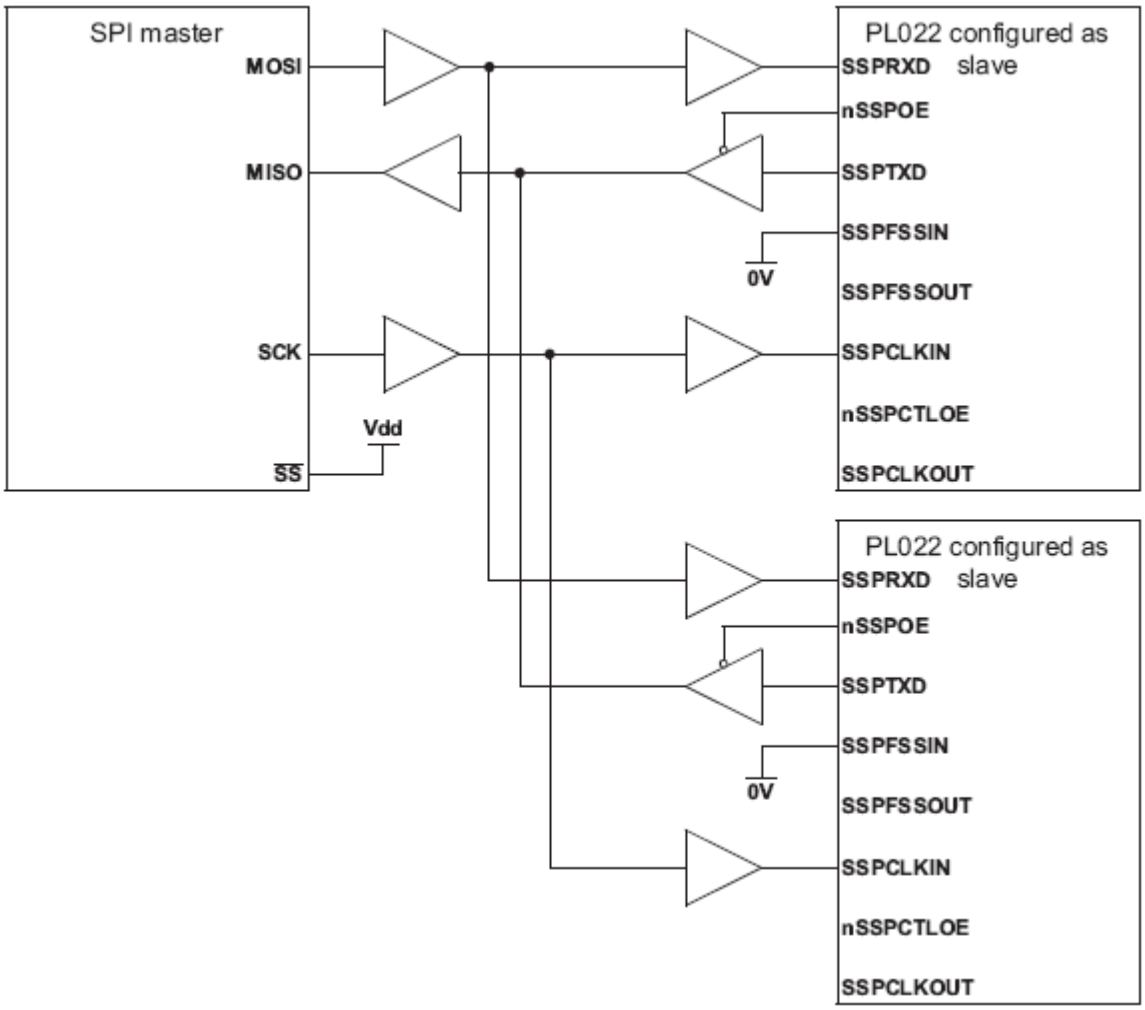


Рисунок 142 – Ведущее устройство, протокол SPI, подключено к двум ведомым модулям SSP

Рисунок 142 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 по кругу 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.

7.18.3 Интерфейс прямого доступа к памяти

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

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.

В таблице 352 приведены значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 352 – 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 143 показывает временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

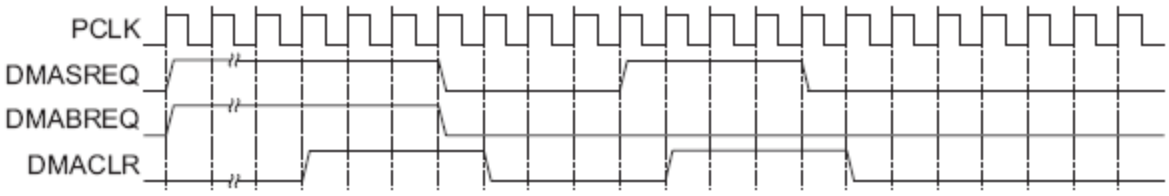


Рисунок 143 – Временные диаграммы обмена в режиме DMA

7.18.4 Прерывания

В модуле предусмотрено четыре маскируемые линии запроса на прерывание с выводом на один общий сигнал, представляющий собой их объединение 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. Общий сигнал запроса прерывания SSPINTR маскируется средствами NVIC ядра контроллера.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний SSPRIS, либо из маскированного регистра прерываний SSPMIS.

7.18.4.1 **SSPRXINTR**

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

7.18.4.2 **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.

7.18.4.3 **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.

7.18.4.4 **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.

7.18.4.5 **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.

7.18.5 Программное управление модулем

7.18.5.1 **Общая информация**

В микросхеме реализован модуль SSP. Базовый адрес регистров управления модулем указан в таблице 353. 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.

7.18.6 Описание регистров

Данные о регистрах модуля SSP приведены в таблице 353.

Таблица 353 – Обобщенные данные о регистрах модуля SSP

Базовый адрес	Наимено-вание	Тип	Значение после сброса	Раз-мер, бит	Описание
0x4009_5000	MDR_SSP0				Контроллер SSP0
Смещение					
0x000	CR0	RW	0x0000	16	Регистр управления 0
0x004	CR1	RW	0x0	4	Регистр управления 1
0x008	DR	RW	0x----	16	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись)
0x00C	SR	RO	0x03	5	Регистр состояния
0x010	CPSR	RW	0x00	8	Регистр делителя тактовой частоты
0x014	IMSC	RW	0x0	4	Регистр маски прерывания
0x018	RIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования
0x01C	MIS	RO	0x0	4	Регистр состояния прерываний с учетом маскирования
0x020	ICR	WO	0x0	2	Регистр сброса прерывания
0x024	DMACR	RW	0x0	2	Регистр управления прямым доступом к памяти

Примечание – В поле «тип» указан вид доступа к регистру:
RW – чтение и запись;
RO – только чтение;
WO – только запись

7.18.6.1 **CR0**

Регистр управления 0

Регистр CR0 содержит пять битовых полей, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 354.

Таблица 354 – Формат регистра CR0

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…16	-	Зарезервировано
15…8	SCR	Скорость последовательного обмена. Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению: F_{SSPCLK} / ($CPSDVSR \cdot (1 + SCR)$),

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		где CPSDVSR – четное число в диапазоне от 2 до 254 (см. регистр SSPCPSR), а SCR – число от 0 до 255
7	SPH	Фаза сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. подпункт 7.18.2.15 «Формат синхронного обмена SPI фирмы Motorola»
6	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. подпункт 7.18.2.15 «Формат синхронного обмена SPI фирмы Motorola»
5…4	FRF	Формат информационного кадра: 00 – протокол SPI фирмы Motorola; 01 – протокол SSI фирмы Texas Instruments; 10 – протокол Microwire фирмы National Semiconductor; 11 – резерв
3…0	DSS	Размер слова данных: 0000 – резерв; 0001 – резерв; 0010 – резерв; 0011 – 4 бита; 0100 – 5 бит; 0101 – 6 бит; 0110 – 7 бит; 0111 – 8 бит; 1000 – 9 бит; 1001 – 10 бит; 1010 – 11 бит; 1011 – 12 бит; 1100 – 13 бит; 1101 – 14 бит; 1110 – 15 бит; 1111 – 16 бит

7.18.6.2 **CR1**

Регистр управления 1

Регистр CR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 355.

Таблица 355 – Регистр CR1

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…4		Резерв, при чтении результат не определен. При записи следует устанавливать в 0
3	SOD	Запрет выходных линий в режиме ведомого устройства.

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. Бит SOD следует установить в случае, если данный ведомый модуль SSP не должен в настоящее время осуществлять передачу данных в линию SSP_TXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSP_TXD в ведомом режиме разрешено; 1 – управление линией SSP_TXD в ведомом режиме запрещено
2	MS	Выбор ведущего или ведомого режима работы: 0 – ведущий модуль (устанавливается по умолчанию); 1 – ведомый модуль
1	SSE	Разрешение работы приемопередатчика: 0 – работа запрещена; 1 – работа разрешена
0	LBM	Тестирование по шлейфу: 0 – нормальный режим работы приемопередатчика; 1 – выход регистра сдвига передатчика соединен со входом регистра сдвига приемника

7.18.6.3 **DR**

Регистр данных

Регистр SSPDR имеет разрядность 16 бит и предназначен для чтения принятых, и записи передаваемых данных.

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

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

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

В режиме обмена данными Microwire фирмы National Semiconductor модуль SSP по умолчанию работает с восьмиразрядными информационными словами (старший значащий байт игнорируется). Размер принимаемых данных задается программно. Буфера FIFO приемника и передатчика автоматически не очищаются даже в случае, если

бит SSE установлен в 0. Это позволяет заполнить буфер передатчика необходимой информацией заблаговременно, перед разрешением работы модуля.

Назначение разрядов регистра SSPDR описано в таблице 356.

Таблица 356– Формат регистра DR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…0	DATA	Принимаемые данные (чтение). Передаваемые данные (запись). В случае, если выбран размер информационного слова менее 16 бит, перед записью в регистр SSPDR необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника

7.18.6.4 **SR**

Регистр состояния

Регистр состояния доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика, а также занятости модуля SSP.

Назначение бит регистра SSPSR представлено в таблице 357.

Таблица 357– Регистр SR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…5		Резерв, при чтении результат не определен
4	BSY	Флаг активности модуля: 0 – модуль SSP не активен; 1 – модуль SSP в настоящее время передает и/или принимает данные, либо буфер FIFO передатчика не пуст
3	RFF	Буфер FIFO приемника заполнен: 0 – не заполнен; 1 – заполнен
2	RNE	Буфер FIFO приемника не пуст: 0 – пуст; 1 – не пуст
1	TNF	Буфер FIFO передатчика не заполнен: 0 – заполнен; 1 – не заполнен
0	TFE	Буфер FIFO передатчика пуст: 0 – не пуст; 1 – пуст

7.18.6.5 **CPSR**

Регистр делителя тактовой частоты

Регистр SSPCPSR используется для установки параметров делителя тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль. Если записать в регистр SSPCPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Таблица 358 отображает назначение бит регистра SSPCPSR.

Таблица 358 – Регистр CPSR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…8	-	Резерв. При чтении результат не определен. При записи следует заполнить нулями
7… 0	CPSDVSR	Коэффициент деления тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль

7.18.6.6 **IMSC**

Регистр установки и сброса маски прерывания

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание. При этом запись 1 в разряд разрешает соответствующее прерывание, запись 0 – запрещает.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение бит регистра IMSC показано в таблице 359.

Таблица 359– Регистр IMSC

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…4		Резерв. Не модифицируйте. При чтении выдаются нули
3	TXIM	Маска прерывания по заполнению на 50% и менее буфера FIFO передатчика. 1 – не маскирована. 0 – маскирована
2	RXIM	Маска прерывания по заполнению на 50% и более буфера FIFO приемника. 1 – не маскирована. 0 – маскирована
1	RTIM	Маска прерывания по таймауту приемника (буфер FIFO приемника не пуст и не было попыток его чтения в течение времени таймаута). 1 – не маскирована. 0 – маскирована
0	RORIM	Маска прерывания по переполнению буфера приемника. 1 – не маскирована. 0 – маскирована

7.18.6.7 **RIS**

Регистр состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Таблица 360 отображает назначение бит в регистре RIS.

Таблица 360 – Регистр RIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31… 4		Резерв. Не модифицируйте. При чтении выдаются нули
3	TXRIS	Состояние до маскирования прерывания SSPTXINTR
2	RXRIS	Состояние до маскирования прерывания SSPRXINTR
1	RTRIS	Состояние до маскирования прерывания SSPRTINTR
0	RORRIS	Состояние до маскирования прерывания SSPRORINTR

7.18.6.8 **MIS**

Регистр маскированного состояния прерываний

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 361.

Таблица 361 – Регистр MIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…4		Резерв. Не модифицируйте. При чтении выдаются нули
3	TXMIS	Состояние маскированного прерывания SSPTXINTR
2	RXMIS	Состояние маскированного прерывания SSPRXINTR
1	RTMIS	Состояние маскированного прерывания SSPRTINTR
0	RORMIS	Состояние маскированного прерывания SSPRORINTR

7.18.6.9 **ICR**

Регистр сброса прерываний

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 362.

Таблица 362 – Регистр ICR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31… 2		Резерв. Не модифицируйте. При чтении выдаются нули
1	RTIC	Сброс прерывания SSPRTINTR
0	RORIC	Сброс прерывания SSPRORINTR

7.18.6.10 **DMACR**

Регистр управления прямым доступом к памяти

Регистр доступен по чтению и записи. После сброса, все биты регистра обнуляются. Назначение бит регистра DMACR представлено в таблице 363.

Таблица 363 – Регистр DMACR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…2		Резерв. Не модифицируйте. При чтении выдаются нули.
1	TXDMAE	Использование DMA при передаче. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика
0	RXDMAE	Использование DMA при приеме. Если бит установлен в 1, разрешено формирование запросов DMA для обслуживания буфера FIFO приемника

7.19 Контроллер UART (UARTx)

Модуль универсального асинхронного приемопередатчика (UART – Universal Asynchronous Receiver Transmitter) является периферийным устройством микросхемы.

7.19.1 Основные сведения

Основные сведения о модуле представлены в следующих разделах:

– основные характеристики;
– программируемые параметры;
– отличия от приемопередатчика 16C650.

7.19.2 Основные характеристики модуля UART

– Содержит независимые буферы приема (16x12) и передачи (16x8) типа FIFO (First In First Out – первый вошел, первый вышел), что позволяет снизить интенсивность прерываний центрального процессора.

– Программное отключение FIFO позволяет ограничить размер буфера одним байтом.

– Программное управление скоростью обмена. Обеспечивается возможность деления тактовой частоты опорного генератора в диапазоне (1x16 – 65535x16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц.

– Поддержка стандартных элементов асинхронного протокола связи – стартового и стопового бит, а также бита контроля четности, которые добавляются перед передачей и удаляются после приема.

– Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по таймауту приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки.

– Поддержка прямого доступа к памяти.
– Обнаружение ложных стартовых бит.
– Формирование и обнаружения сигнала разрыва линии.
– Полностью программируемый асинхронный последовательный интерфейс с характеристиками:
  – данные длиной 5, 6, 7 или 8 бит;
  – формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение, либо не передается);
  – формирование 1 или 2 стоповых бит;
  – скорость передачи данных – от 0 до UARTCLK/16 Бод.
– Наличие идентификационного регистра, однозначно идентифицирующего модуль, что позволяет операционной системе выполнять автоматическую конфигурацию.

7.19.3 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:
– скорость передачи данных – целая и дробная часть числа;
– количество бит данных;
– количество стоповых бит;
– режим контроля четности;
– разрешение или запрет использования буферов FIFO (глубина очереди данных – 16 элементов или один элемент, соответственно);
– порог срабатывания прерывания по заполнению буферов FIFO (1/8, 1/4, 1/2, 3/4 и 7/8);
– частота внутреннего тактового генератора (номинальное значение – 1,8432 МГц) может быть задана в диапазоне 1,42 – 2,12 МГц для обеспечения возможности формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления;
– режим аппаратного управления потоком данных.

7.19.3.1 **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.

7.19.3.2 **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 UARTCLK;

– обеспечивает возможности, сходные с возможностями индустриального стандарта – контроллера UART 16C650.

Режим работы приемопередатчика и скорость обмена данными контролируются регистром управления линией UARTLCR_H и регистрами делителя скорости передачи данных – целой части (UARTIBRD) и дробной части (UARTFBRD).

Устройство может формировать следующие сигналы:

– независимые маскируемые прерывания от приемника (в том числе по таймауту), передатчика, а также по изменению состояния модема и в случае обнаружения ошибки;

– общее прерывание, возникающее в случае, если возникло одно из независимых немаскированных прерываний;

– 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.

7.19.4 Описание функционирования блока UART

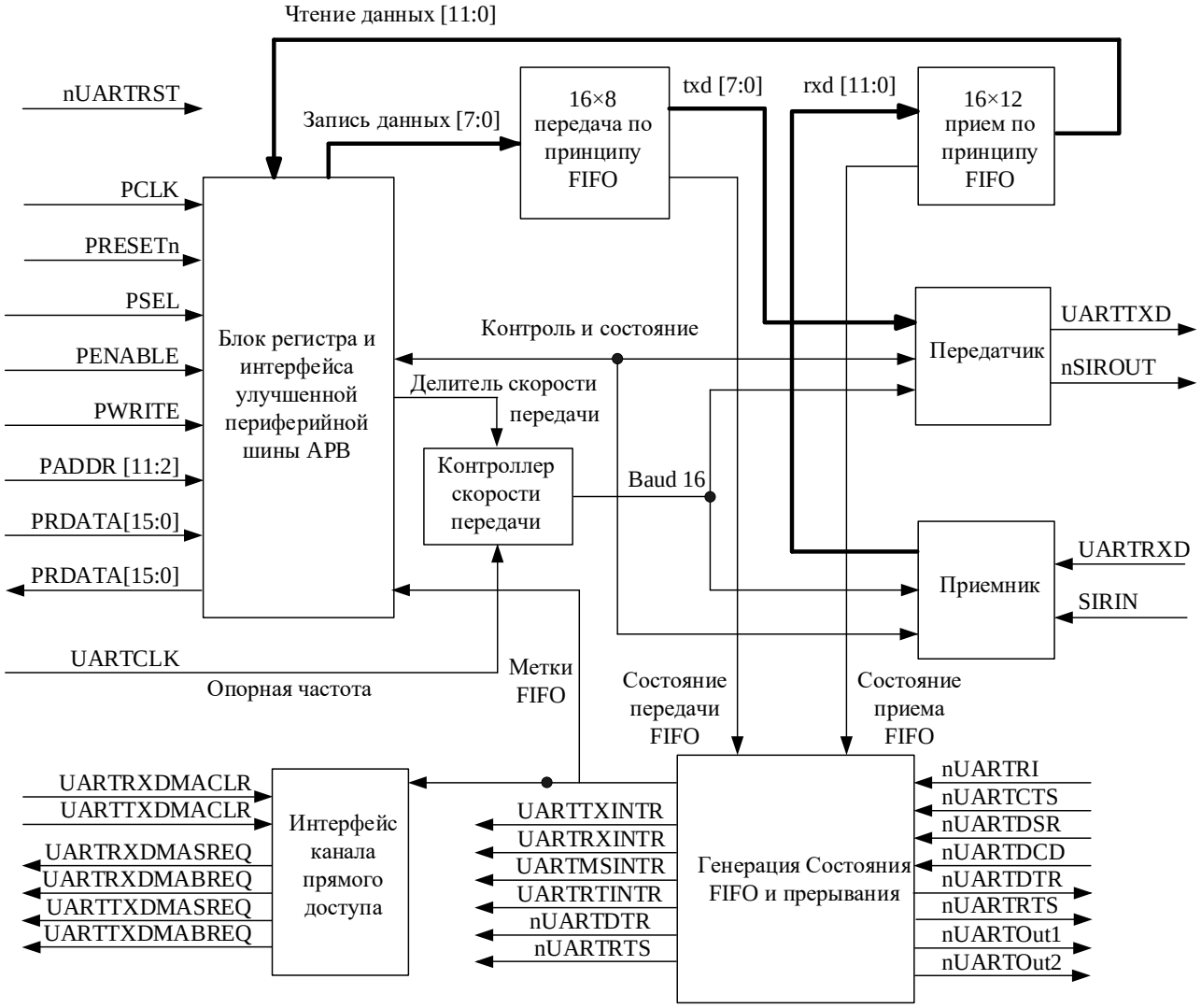


Рисунок 144 – Блок-схема универсального асинхронного приемопередатчика (UART)

7.19.4.1 **Генератор тактового сигнала приемопередатчика**

Генератор содержит счетчики без цепи сброса, формирующие внутренние тактовые сигналы Baud16.

Сигнал Baud16 используется для синхронизации схем управления приемником и передатчиком последовательного обмена данными. Он представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UARTCLK и частотой, в 16 раз выше скорости передачи данных.

7.19.4.2 **Буфер FIFO передатчика**

Буфер передатчика имеет ширину 8 бит, глубину 16 слов, схему организации доступа типа «первый вошел, первый вышел». Данные от центрального процессора, записанные через шину APB, сохраняются в буфере до тех пор, пока не будут считаны логической схемой передачи данных. Существует возможность запретить буфер FIFO передатчика, в этом случае он будет функционировать как однобайтовый буферный регистр.

7.19.4.3 **Буфер FIFO приемника**

Буфер приемника имеет ширину 12 бит, глубину 16 слов, схему организации доступа типа «первый вошел, первый вышел». Принятые от периферийного устройства данные и соответствующие кодов ошибки сохраняются логикой приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину APB. Буфер FIFO приемника может быть запрещен, в этом случае он будет действовать как однобайтовый буферный регистр.

7.19.4.4 **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.

7.19.4.5 **Блок приемника**

Логические схемы приемника преобразуют данные, полученные от периферийного устройства, из последовательной в параллельную форму после обнаружения корректного стартового импульса. Кроме того, производятся проверки переполнения буфера, проверки на ошибки контроля четности, на ошибки в структуре сигнала, а также на разрыв линии. Признаки обнаружения этих ошибок также сохраняются в выходном буфере.

7.19.4.6 **Блок формирования прерываний**

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

Комбинированный сигнал прерывания может быть подан на внешний контроллер прерываний системы, при этом появится дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

Другой подход состоит в подаче на системный контроллер прерываний независимых линий запроса на прерывание от приемопередатчика. В этом случае процедура обработки сможет одновременно считать информацию обо всех источниках прерывания. Данный подход привлекателен в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

Для более подробной информации см. пункт 7.19.9 «Прерывания».

7.19.5 Интерфейс прямого доступа к памяти

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

7.19.6 Блок и регистры синхронизации

Контроллер поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов CPU_CLK и UARTCLK. Регистры синхронизации и логика квитирования постоянно находятся в активном состоянии. Это практически не отражается на характеристиках устройства и занимаемой площади. Синхронизация сигналов управления осуществляется в обоих направлениях потока данных, то есть как из области действия CPU_CLK в область действия UARTCLK, так и наоборот.

7.19.7 Описание работы UART

7.19.7.1 **Сброс модуля**

Приемопередатчик и кодек могут быть сброшены общим сигналом сброса процессора. Значения регистров после сброса описаны в пункте 7.19.10 «Программное управление модулем».

7.19.7.2 **Тактовые сигналы**

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

$$F_{UARTCLK}(min) \geq 16 \cdot baud_rate_{max}\,; \tag{14}$$

$$F_{UARTCLK}(max) \leq 16 \cdot 65535 \cdot baud_rate_{min}\,. \tag{15}$$

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота UARTCLK должна находиться в интервале от 7,3728 МГц до 115,34 МГц.

Частота UARTCLK, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами CPU_CLK и UARTCLK. Частота UARTCLK должна быть не более, чем в 5/3 раз выше частоты CPU_CLK.

$$F_{UARTCLK} \leq \frac{5}{3} \cdot F_{CPU_CLK}\,. \tag{16}$$

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

7.19.7.3 **Работа универсального асинхронного приемопередатчика**

Управляющая информация хранится в регистре управления линией UARTLCR. Этот регистр имеет внутреннюю ширину 30 бит, однако внешний доступ по шине APB к нему осуществляется через следующие регистры:

- UARTLCR_H – определяет:
  - параметры передачи данных;
  - длину слова;
  - режим буферизации;
  - количество передаваемых стоповых бит;
  - режим контроля четности;
  - формирование сигнала разрыва линии;
- UARTIBRD – определяет целую часть коэффициента деления для скорости передачи данных;
- UARTFBRD – определяет дробную часть коэффициента деления для скорости передачи данных.

7.19.7.4 **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 UARTCLK тактовый сигнал с произвольной частотой более 3,6864 МГц.

Целая часть коэффициента деления записывается в 16-битный регистр UARTIBRD. Шестиразрядная дробная часть записывается в регистр UARTFBRD. Значение коэффициента деления связано с содержимым указанных регистров следующим образом

$$\text{Коэффициент деления} = \frac{\text{UARTCLK}}{(16 \cdot \text{скорость передачи данных})} = \text{BRD_I} + \text{BRD_F}, \qquad (17)$$

где BRD_I – целая часть;

BRD_F – дробная часть коэффициента деления.

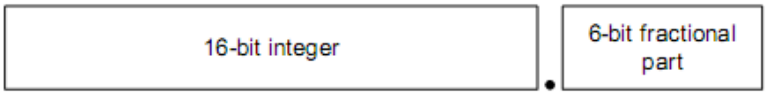


Рисунок 145 – Коэффициент деления

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

$$M = \lfloor BRD_F \cdot 2^n + 0{,}5 \rfloor, \qquad (18)$$

где $\lfloor \quad \rfloor$ – операция отсечения дробной части числа, n = 6.

В модуле формируется внутренний сигнал Baud16, представляющий собой последовательность импульсов с длительностью, равной периоду сигнала UARTCLK и средней частотой, в 16 раз большей требуемой скорости обмена данными.

7.19.7.5 **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 UARTLCR_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. Сигнал BUSY переходит в низкое состояние 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 UART_RXD постоянно присутствовала единица), и произошел переход входного сигнала из высокого в низкий логический уровень (обнаружен стартовый бит), то включается счетчик, тактируемый сигналом Baud16. После чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов (в режиме асинхронного приемопередатчика) или каждые четыре такта (в режиме ИК обмена данными) сигнала Baud16. Более частая выборка данных в режиме ИК обмена связана с необходимостью корректной обработки импульсов данных согласно протоколу SIR IrDA.

Стартовый бит считается достоверным в случае, если сигнал на линии UART_RXD сохраняет низкий логический уровень в течение восьми отсчетов сигнала Baud16 с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

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

Наконец, производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UART_RXD). В случае, если последнее условие не выполняется, устанавливается признак ошибки формирования кадра. После того, как слово данных принято полностью, оно заносится в буфер FIFO приемника, наряду с четырьмя битами признаков ошибки, связанных с принятым словом (см. таблицу 364).

7.19.7.6 **Биты ошибки**

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся в разряды [10…8] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO в разряде 11 слова данных.

Таблица 364 описывает назначение всех бит слова данных в FIFO-буфере приемника.

Таблица 364 – Назначение бит слова данных в FIFO-буфере приемника

Бит буфера FIFO	Назначение
11	Признак переполнения буфера
10	Ошибка – «разрыв линии»
9	Ошибка проверки на четность
8	Ошибка формирования кадра
7…0	Принятые данные

7.19.7.7 **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.

7.19.7.8 **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.

7.19.7.9 **Проверка по шлейфу**

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

7.19.8 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 UARTDMACR.

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

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

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

– 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 UARTIFLS.

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

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

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

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

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

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

Когда буферы FIFO включены, обмен данными может производиться в ходе как одноэлементных, так и блочных передач данных, в зависимости от установленной величины порога заполнения буферов и их фактического заполнения. Таблица 365 показывает значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

Таблица 365 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

Пороговый уровень	Длина блока обмена данными	
	Буфер передатчика (количество незаполненных ячеек)	Буфер приемника (количество заполненных ячеек)
1/8	14	2
1/4	12	4
1/2	8	8
3/4	4	12
7/8	2	14

В регистре управления DMA UARTDMACR 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.

На рисунке 146 показаны 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.

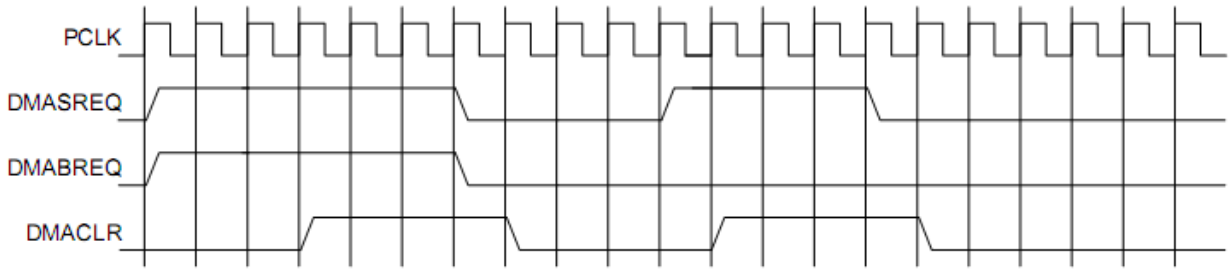


Рисунок 146 – Временные диаграммы одноэлементного и блочного запросов DMA

7.19.9 Прерывания

В модуле предусмотрено семь 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.
- 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, и UARTEINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски UARTIMSC. Установка бита в 1 разрешает соответствующее прерывание, в 0 – запрещает.

Доступность, как индивидуальных линий, так и общей линии запроса позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика UARTRXINTR и UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

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

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

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний UARTRIS, либо из маскированного регистра прерываний UARTMIS.

7.19.9.1 **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.

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

7.19.9.3 **UARTRTINTR**

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения (или считывания одного байта в случае, если буфер FIFO запрещен), либо путем записи 1 в соответствующий бит регистра сброса прерывания UARTICR.

7.19.9.4 **UARTEINTR**

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

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

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

Сброс прерывания осуществляется путем записи соответствующих бит в регистр сброса прерывания UARTICR. За прерываниями по обнаружению ошибки закреплены биты с 7 по 10.

7.19.9.5 **UARTINTR**

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR и UARTEINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерываний, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

7.19.10 Программное управление модулем

7.19.10.1 **Общая информация**

Следующая информация применима ко всем регистрам контроллера:

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

– Не следует пытаться получить доступ к зарезервированным или неиспользуемым адресам. Это может привести к непредсказуемому поведению модуля.

– За исключением специально оговоренных в настоящем документе случаев:

– не следует изменять значения не определенных в документе разрядов регистров;

– не следует использовать значения не определенных в документе разрядов регистров;

– все биты регистров (за исключением специально оговоренных случаев) устанавливаются в значение 0 после сброса по включению питания или системного сброса.

– Столбец «Тип» (см. таблицу 366) определяет режим доступа к регистру в соответствии с обозначениями:

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

7.19.11 Описание регистров

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в таблице 366.

Таблица 366 – Обобщенные данные о регистрах устройства

Адрес	Наимено-вание	Тип	Значение после сброса	Размер, бит	Описание
0x4009_9000	MDR_UART1				Контроллер UART1
0x4009_A000	MDR_UART2				Контроллер UART2
Смещение					
0x000	DR	RW	0x---	12/8	Регистр данных
0x004	RSR_ECR	RW	0x0	4/0	Регистр состояния приемника / Сброс ошибки приемника
0x008-0x014					Зарезервировано
0x018	FR	RO	0b-10010---	9	Регистр флагов
0x01C					Зарезервировано
0x020					Зарезервировано
0x024	IBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	FBRD	RW	0x00	6	Дробная часть делителя скорости обмена данными
0x02C	LCR_H	RW	0x00	8	Регистр управления линией
0x030	CR	RW	0x0300	16	Регистр управления
0x034	IFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	IMSC	RW	0x000	11	Регистр маски прерывания
0x03C	RIS	RO	0x00-	11	Регистр состояния прерываний
0x040	MIS	RO	0x00-	11	Регистр состояния прерываний с маскированием
0x044	ICR	WO	-	11	Регистр сброса прерывания
0x048	DMACR	RW	0x00	3	Регистр управления DMA
0x04C-FDC					Зарезервировано
0xFE0	PerID0	RO	0x11	8	Идентификационный регистр 0
0xFE4	PerID1	RO	0x10	8	Идентификационный регистр 1
0xFE8	PerID2	RO	0x14	8	Идентификационный регистр 2
0xFEC	PerID3	RO	0x00	8	Идентификационный регистр 3
0xFF0	PerID4	RO	0x0D	8	Идентификационный регистр 4
0xFF4	PerID5	RO	0xF0	8	Идентификационный регистр 5
0xFF8	PerID6	RO	0x05	8	Идентификационный регистр 6
0xFFC	PerID7	RO	0xB1	8	Идентификационный регистр 7

7.19.11.1 **DR**

Регистр данных

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 UARTDR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра UARTRSR/UARTECR (см. таблицу 367).

Таблица 367 – Формат регистра UARTDR

Номер бита	Сигнал	Назначение
15…12		Резерв
11	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время, как буфер заполнен. Сбрасывается в 0 после того, как в буфере появится свободное место
10	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита
9	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам бит EPS и SPS в регистре управления линией UARTLCR_H. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер

Номер бита	Сигнал	Назначение
8	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер
7…0	DATA	Принимаемые данные (чтение). Передаваемые данные (запись)
Примечание – Необходимо запрещать работу приемопередатчика перед любым перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние во время передачи или приема символа, то перед остановкой он завершает выполняемую операцию		

7.19.11.2 **RSR_ECR**

Регистр состояния приемника / сброса ошибки

Состояние приемника также может быть считано из регистра UARTRSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра относится к последнему символу, считанному из регистра данных UARTDR. С другой стороны, признак переполнения буфера устанавливается немедленно после возникновения этого состояния (и не связан с последним, считанным из регистра UARTDR, байтом данных).

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

В таблице 368 приведены назначения бит регистра UARTRSR/UARTECR.

Таблица 368 – Регистр UARTRSR/UARTECR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7…4		Резерв, при чтении результат не определен
3	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время как буфер заполнен. Сбрасывается в 0 после записи в регистр UARTECR. Содержимое буфера остается верным, так как был перезаписан только регистр сдвига. Центральный процессор должен считать данные для того, чтобы освободить буфер FIFO
2	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). Бит сбрасывается в 0 после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с символом,

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		находящемся на вершине буфера. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита
1	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам бит EPS и SPS в регистре управления линией UARTLCR_H (стр. 3-12). Бит сбрасывается в 0 после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера
0	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). Бит сбрасывается в 0 после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с символом, находящимся на вершине буфера
Примечания 1 Перед чтением регистра состояния UARTRSR необходимо считать данные, принятые из линии, путем обращения к регистру данных UARTDR. Противоположная последовательность действий не допускается, так как регистр UARTRSR обновляет свое состояние только после чтения регистра UARTDR. Вместе с тем, информация о состоянии приемника может быть получена непосредственно из регистра данных UARTDR. 2 Запись в регистр UARTRSR/UARTECR любого кода сбрасывает признаки ошибок формирования кадра, проверки на четность, разрыва линии и переполнения буфера		

7.19.11.3 **FR**

Регистр флагов

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. В таблице 369 приведена информация о назначении бит регистра UARTFR.

Таблица 369 – Регистр UARTFR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…9		Резерв. Не модифицируйте. При чтении заполняются нулями
8	-	Зарезервировано
7	TXFE	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр передатчика пуст. В противном случае он равен 1, если пуст буфер FIFO передатчика. Данный бит не дает

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		никакой информации о наличии данных в регистре сдвига передатчика
6	RXFF	Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника занят. В противном случае он равен 1, если заполнен буфер FIFO приемника
5	TXFF	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит равен 1, когда буферный регистр передатчика занят. В противном случае он равен 1, если заполнен буфер FIFO передатчика
4	RXFE	Буфер FIFO приемника пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника пуст. В противном случае он равен 1, если пуст буфер FIFO приемника
3	BUSY	UART занят. Бит равен 1 в случае, если контроллер передает в линию данные. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Кроме того, бит занятости устанавливается в 1 при наличии данных в буфере FIFO передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен)
2	-	Зарезервировано
1	-	Зарезервировано
0	-	Зарезервировано

7.19.11.4 **IBRD**

Регистр целой части делителя скорости передачи данных

Назначение бит регистра UARTIBRD показано ниже.

Таблица 370 – Регистр UARTIBRD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…0	BAUDDIV_INT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

7.19.11.5 **FBRD**

Регистр дробной части делителя скорости передачи данных

В таблице 371 приведены назначение бит регистра.

Таблица 371 – Регистр UARTFBRD

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
5…0	BAUDDIV_FRAC	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в 0

Коэффициент деления вычисляется по формуле

$$BAUDDIV = \frac{F_{UARTCLK}}{16 \cdot Baud_rate}, \quad (19)$$

где $F_{UARTCLK}$ – тактовая частота контроллера UART, Baud_rate – требуемая скорость передачи данных.

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUDDIV_INT и BAUDDIV_FRAC, соответственно.

Примечания

1 Изменения содержимого регистров UARTIBRD и UARTFBRD вступают в силу только после завершения передачи и приема текущего символа данных.

2 Минимальный допустимый коэффициент деления – 1, максимальный 65535 (2^{16} - 1). Таким образом, значение UARTIBRD, равное 0, является недопустимым, при этом значение регистра UARTFBRD игнорируется.

3 Аналогично, при UARTIBRD равном 65535 (0xFFFF), значение UARTFBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

Далее приведен пример вычисления коэффициента деления.

Пример. Вычисление коэффициента деления

Пусть требуемая скорость передачи данных составляет 230400 бит/с, частота тактового сигнала UARTCLK равна 4 МГц. Тогда

$$\text{Коэффициент деления} = (4 \cdot 10^6)/(16 \cdot 230400) = 1{,}085.$$

Таким образом, BRDI = 1, BRDF = 0,085.

Следовательно, значение, записываемое в регистр UARTBFRD, равно

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

Реальное значение коэффициента деления равно

$$1 + 5/64 = 1{,}078.$$

Реальная скорость передачи данных равна

$$(4 \cdot 10^6)/(16 \cdot 1.078) = 231911 \text{ бит/с}.$$

Ошибка установки скорости равна

$$\frac{231911-230400}{230400} \cdot 100\% = 0{,}656\%.$$

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

В таблице 372 представлены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 7,3728 МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр UARTFBRD должен быть записан ноль.

Таблица 372 – Коэффициенты деления при частоте UARTCLK = 7,3728 МГц

Коэффициент деления	Скорость передачи данных
0x0001	460800
0x0002	230400
0x0004	115200
0x0006	76800
0x0008	57600
0x000C	38400
0x0018	19200
0x0020	14400
0x0030	9600
0x00C0	2400
0x0180	1200
0x105D	110

В таблице 373 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц.

Таблица 373 – Коэффициенты деления при частоте UARTCLK = 4 МГц

Целая часть	Дробная часть	Требуемая скорость	Реальная скорость	Ошибка, %
0x001	0x05	230400	231911	0.656
0x002	0x0B	115200	115101	0.086
0x003	0x10	76800	76923	0.160
0x006	0x21	38400	38369	0.081
0x011	0x17	14400	14401	0.007
0x068	0x0B	2400	2400	~ 0
0x8E0	0x2F	110	110	~ 0

7.19.11.6 **LCR_H**

Регистр управления линией

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

Таблица 374 показывает назначение разрядов регистра UARTLCR_H.

Таблица 374 – Регистр UARTLCR_H

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…8		Резерв. Не модифицируйте. При чтении выдаются нули
7	SPS	Передача бита четности с фиксированным значением: 0 – запрещена; 1 – на месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. (При EPS=0 на месте бита четности передается 1, при EPS=1 – передается 0). Значение бита SPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещены
6…5	WLEN	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0b11 – 8 бит; 0b10 – 7 бит; 0b01 – 6 бит; 0b00 – 5 бит
4	FEN	Разрешение работы буфера FIFO приемника и передатчика: 0 – запрещено; 1 – разрешено
3	STP2	Режим передачи двух стоповых бит: 0 – один стоповый бит; 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре
2	EPS	Четность/нечетность: 0 – бит четности дополняет количество единиц в информационной части кадра до нечетного; 1 – бит четности дополняет количество единиц в информационной части кадра до четного числа. Значение бита EPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещены
1	PEN	Разрешение проверки четности: 0 – кадр не содержит бита четности; 1 – бит четности передается в кадре и проверяется при приеме данных

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
0	BRK	Разрыв линии. Если этот бит установлен в 1, то по завершении передачи текущего символа на выходе UARTTXD устанавливается низкий уровень сигнала. Для правильного выполнения этой операции программное обеспечение должно обеспечить передачу сигнала разрыва в течение, как минимум, времени передачи двух информационных кадров. В нормальном режиме функционирования бит должен быть установлен в 0

Содержимое регистров UARTLCR_H, UARTIBRD и UARTFBRD совместно образует общий 30-разрядный регистр UARTLCR, который обновляется по стробу, формируемому при записи в UARTLCR_H. Для того, чтобы изменение параметров коэффициента деления частоты обмена данными вступило в силу, после изменения значения регистров UARTIBRD и/или UARTFBRD необходимо осуществить запись данных в регистр UARTLCR_H.

Примечания

1 Изменение значений трех регистров можно осуществить корректно двумя способами:

– запись UARTIBRD, запись UARTFBRD, запись UARTLCR_H;

– запись UARTFBRD, запись UARTIBRD, запись UARTLCR_H.

2 Для того чтобы изменить значение лишь одного из регистров (UARTIBRD или UARTFBRD), необходимо выполнить следующий шаг:

– запись UARTIBRD (или UARTFBRD), запись UARTLCR_H.

Таблица 375 показывает таблицу истинности для бит управления контролем четности PEN, EPS и SPS регистра управления линией UARTLCR_H.

Таблица 375 – Управление режимом контроля четности

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

Примечания

1 Регистры UARTLCR_H, UARTIBRD и UARTFBRD не должны изменяться:

– при разрешенной работе приемопередатчика;

– во время завершения приема или передачи данных в процессе остановки (перевода в запрещенное состояние) приемопередатчика.

2 Целостность данных в буферах FIFO не гарантируется в следующих случаях:

– после установки бита разрыва линии BRK;

– если программное обеспечение произвело остановку приемопередатчика при наличии данных в буферах FIFO после его повторного перевода в разрешенное состояние.

7.19.11.7 **CR**

Регистр управления

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

Назначение разрядов регистра управления показано в таблице 376.

Таблица 376 – Регистр управления UARTCR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15…10	-	Зарезервировано
9	RXE	Прием разрешен. Установка бита в 1 разрешает работу приемника. В случае перевода приемопередатчика в запрещенное состояние в ходе приема данных, он завершает прием текущего символа перед остановкой
8	TXE	Передача разрешена. Установка бита в 1 разрешает работу передатчика. В случае перевода приемопередатчика в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой
7	LBE	0 – запрещено; 1 – шлейф разрешен. Выходная линия передатчика UARTTXD коммутируется на вход приемника UARTRXD. После сброса бит устанавливается в 0
6…3	-	Резерв. Не модифицируйте. При чтении выдаются нули
2, 1	-	Зарезервировано
0	UARTEN	Разрешение работы приемопередатчика: 0 – работа запрещена. Перед остановкой завершается прием и/или передача обрабатываемого в текущий момент символа. 1 – работа разрешена. Производится обмен данными либо по линиям асинхронного обмена, либо по линиям ИК обмена SIR, в зависимости от состояния бита SIREN
Примечание – Для того, чтобы разрешить передачу данных, биты TXE и UARTEN необходимо установить в логическую 1. Аналогично, для разрешения приема данных необходимо установить в 1 биты RXE и UARTEN		

Примечание – Рекомендуется следующая последовательность действий для программирования регистров управления:

1 остановите работу приемопередатчика;
2 дождитесь окончания приема и/или передачи текущего символа данных;
3 сбросьте буфер передатчика путем установки бита FEN регистра UARTLCR_H в 0;
4 измените настройки регистра UARTCR;
5 возобновите работу приемопередатчика.

7.19.11.8 **IFLS**

Регистр порога прерывания по заполнению буфера FIFO

Данный регистр используется для установки порогового значения заполнения буферов передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода величины заполнения буфера через заданное значение.

После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра UARTIFLS и значения его бит представлены в таблице 377.

Таблица 377 – Регистр UARTIFLS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…6		Резерв. Не модифицируйте. При чтении выдаются нули
5…3	RXIFLSEL	Порог прерывания по заполнению буфера приемника: b000 = буфер заполнен на 1/8; b001 = буфер заполнен на 1/4; b010 = буфер заполнен на 1/2; b011 = буфер заполнен на 3/4; b100 = буфер заполнен на 7/8; b101-b111 = резерв
2…0	TXIFLSEL	Порог прерывания по заполнению буфера передатчика: b000 = буфер заполнен на 1/8; b001 = буфер заполнен на 1/4; b010 = буфер заполнен на 1/2; b011 = буфер заполнен на 3/4; b100 = буфер заполнен на 7/8; b101-b111 = резерв. Примечание – если сдвиговый регистр передатчика пуст, 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

7.19.11.9 **IMSC**

Регистр установки сброса маски прерывания

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание.

После сброса все биты регистра маски устанавливаются в нулевое состояние. Назначение бит регистра UARTIMSC показано в таблице 378.

Таблица 378 – Регистр UARTIMSC

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…11		Зарезервировано. Не модифицируйте. При чтении выдаются нули
10	OEIM	Маска прерывания по переполнению буфера UARTOEINTR: 1 – установлена; 0 – сброшена
9	BEIM	Маска прерывания по разрыву линии UARTBEINTR: 1 – установлена; 0 – сброшена
8	PEIM	Маска прерывания по ошибке контроля четности UARTPEINTR: 1 – установлена; 0 – сброшена
7	FEIM	Маска прерывания по ошибке в структуре кадра UARTFEINTR: 1 – установлена; 0 – сброшена
6	RTIM	Маска прерывания по таймауту приема данных UARTRTINTR: 1 – установлена; 0 – сброшена
5	TXIM	Маска прерывания от передатчика UARTTXINTR. 1 – установлена; 0 – сброшена
4	RXIM	Маска прерывания от приемника UARTRXINTR. 1 – установлена; 0 – сброшена
3…0	-	Зарезервировано

7.19.11.10 **RIS**

Регистр состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Предупреждение. После сброса все биты регистра, за исключением бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит регистра UARTRIS представлено в таблице 379.

Таблица 379 – Регистр UARTRIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…11		Зарезервировано. Не модифицируйте. При чтении выдаются нули
10	OERIS	Состояние прерывания по переполнению буфера UARTOEINTR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
9	BERIS	Состояние прерывания по разрыву линии UARTBEINTR
8	PERIS	Состояние прерывания по ошибке контроля четности UARTPEINTR
7	FERIS	Состояние прерывания по ошибке в структуре кадра UARTFEINTR
6	RTRIS	Состояние прерывания по таймауту приема данных UARTRTINTR. Примечание – Бит RTRIS может быть установлен только при установленной маске прерывания по таймауту приема данных UARTRTINTR в регистре UARTIMSC. Чтение состояния прерывания по таймауту из регистров UARTMIS и UARTRIS приводит к одинаковым результатам
5	TXRIS	Состояние прерывания от передатчика UARTTXINTR
4	RXRIS	Состояние прерывания от приемника UARTRXINTR
3	DSRRMIS	Состояние прерывания UARTDSRINTR по изменению линии nUARTDSR
2	DCDRMIS	Состояние прерывания UARTDCDINTR по изменению линии nUARTDCD
1	CTSRMIS	Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS
0	RIRMIS	Состояние прерывания UARTRIINTR по изменению линии nUARTRI

7.19.11.11 **MIS**

Регистр маскированного состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит регистра UARTMIS представлено в таблице 380.

Таблица 380 – Регистр UARTMIS

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…11		Зарезервировано. Не модифицируйте. При чтении выдаются нули
10	OEMIS	Маскированное состояние прерывания по переполнению буфера UARTOEINTR
9	BEMIS	Маскированное состояние прерывания по разрыву линии UARTBEINTR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8	PEMIS	Маскированное состояние прерывания по ошибке контроля четности UARTPEINTR
7	FEMIS	Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR
6	RTMIS	Маскированное состояние прерывания по таймауту приема данных UARTRTINTR
5	TXMIS	Маскированное состояние прерывания от передатчика UARTTXINTR
4	RXMIS	Маскированное состояние прерывания от приемника UARTRXINTR
3…0	-	Зарезервировано

7.19.11.12 **ICR**

Регистр сброса прерываний

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись нуля в любой из разрядов регистра игнорируется.

Назначение бит регистра UARTICR представлено в таблице 381.

Таблица 381 – Регистр UARTICR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…11		Зарезервировано. Не модифицируйте. При чтении выдаются нули
10	OEIC	Сброс прерывания по переполнению буфера UARTOEINTR
9	BEIC	Сброс прерывания по разрыву линии UARTBEINTR
8	PEIC	Сброс прерывания по ошибке контроля четности UARTPEINTR
7	FEIC	Сброс прерывания по ошибке в структуре кадра UARTFEINTR
6	RTIC	Сброс прерывания по таймауту приема данных UARTRTINTR
5	TXIC	Сброс прерывания от передатчика UARTTXINTR
4	RXIC	Сброс прерывания от приемника UARTRXINTR
3…0	-	Зарезервировано
Примечание - Для организации надежного сброса статусных флагов прерываний необходимо учитывать требования, изложенные в подразделе 9.3		

7.19.11.13 **DMACR**

Регистр управления прямым доступом к памяти

Регистр доступен по чтению и записи. После сброса, все биты регистра обнуляются. Назначение бит регистра UARTDMACR представлено в таблице 382.

Таблица 382 – Регистр UARTDMACR

Номер бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…3		Зарезервировано. Не модифицируйте. При чтении выдаются нули
2	DMAONERR	Если бит установлен в 1, то в случае возникновения прерывания по обнаружению ошибки блокируются запросы DMA от приемника UARTRXDMASREQ и UARTRXDMABREQ
1	TXDMAE	Использование DMA при передаче. Если бит установлен в 1, то разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика
0	RXDMAE	Использование DMA при приеме. Если бит установлен в 1, то разрешено формирование запросов DMA для обслуживания буфера FIFO приемника

7.20 Контроллер интерфейса ARINC

Контроллер интерфейса ARINC состоит из восьми приемников и четырех передатчиков по ГОСТ 18977-79 (далее ARINC). Каждый приемник поддерживает функцию распознавания меток (или адресов). Для каждого приемника может быть запрограммировано до 32 8-разрядных 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 форма т 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 32x32 до 256x32. 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.

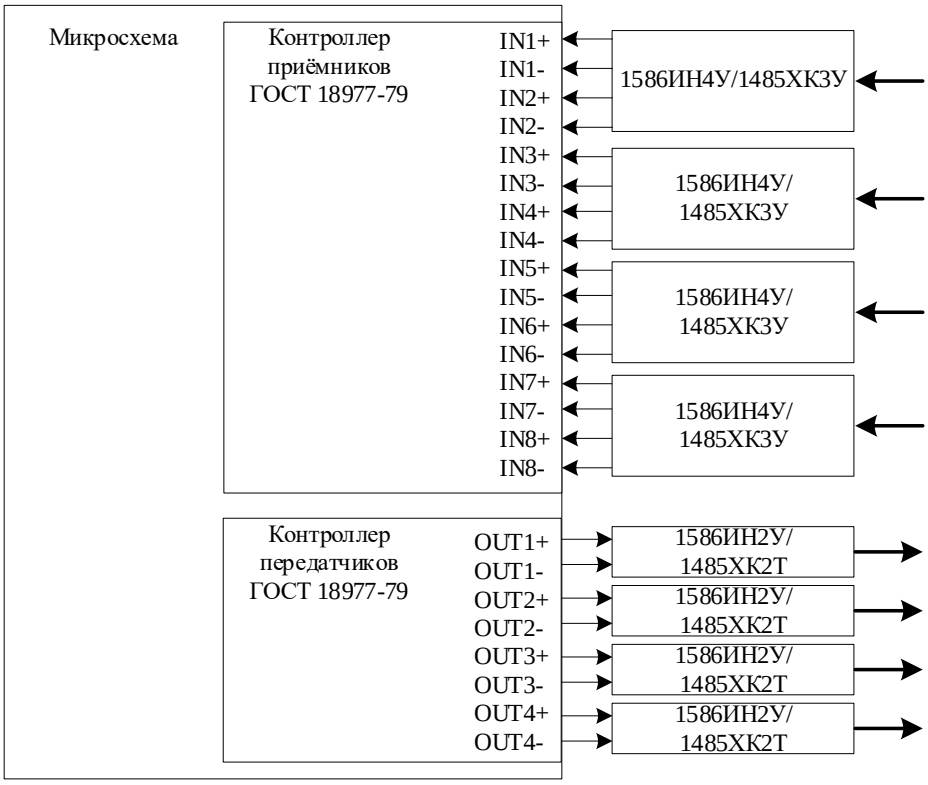


Рисунок 147 – Структурная схема контроллера интерфейса ARINC

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 256х×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.

7.20.1 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.

32	31	30	29 … 11	10	9	8 … 1
P	SSM		DATA MSB … LSB	SDI		LABEL

Старший разряд всегда бит паритета. Стандартом установлено, что бит паритета должен дополнять слово до нечетного. Таким образом, количество единиц в 32-х 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 383.

Таблица 383 – Значения битов 31 и 30

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 1 по 8 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.

7.20.2 Структурная схема канала приема

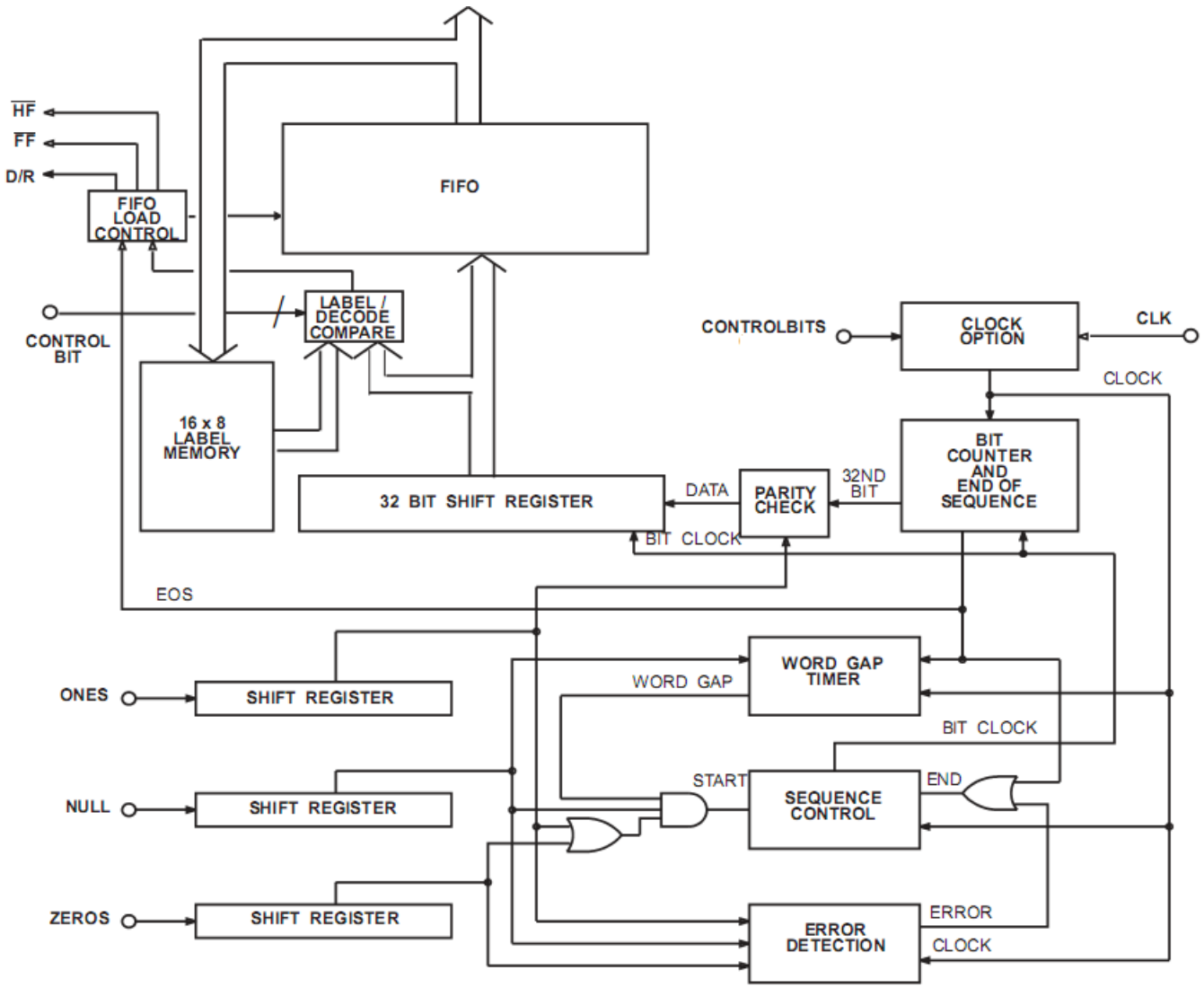


Рисунок 148 – Схема канала приема ARINC

Представленная на рисунке схема (рисунок 148) работает на частоте CLK = 1 МГц, в этом случае ошибка обнаружения бита в линии будет составлять не 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. Ниже приведены случаи, когда происходит загрузка FIFO принятыми данными (см. таблицу 384).

Таблица 384 – 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. Помимо этого, 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.

7.20.3 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 данные можно загрузить только в доступные в FIFO ячейки. Если FIFO заполнено полностью, флаг FFT установлен в единицу, то FIFO игнорирует дальнейшие попытки записи в него. FIFO наполовину полно, если установлен флаг HFT, 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.

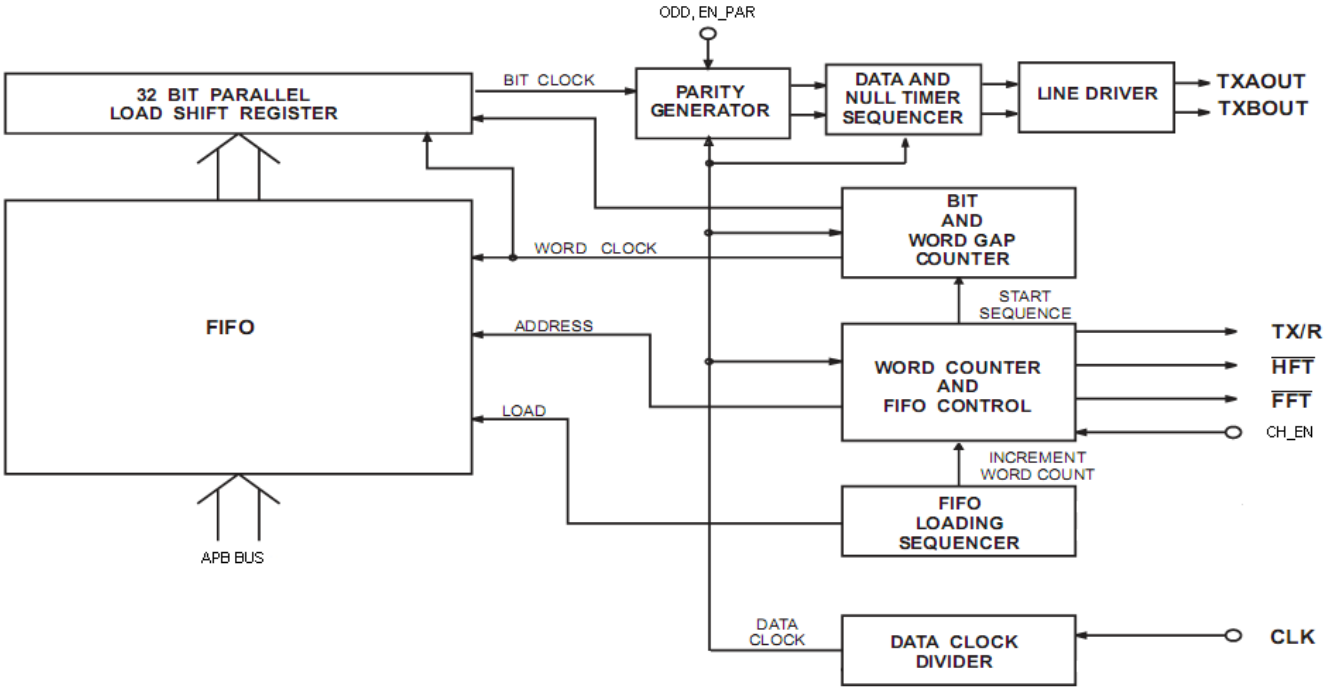


Рисунок 149 – Структурная схема канала передачи

7.20.4 Описание регистров приемника

Таблица 385 – Описание регистров приемника

Базовый адрес	Название	Состояние после сброса	Описание
0x4009_E000	ARINC429R0		Контроллер интерфейса приемников ARINC429
Смещение			
0x0000_0000	CONTROL1		Регистр управления приемников 1
0x0000_0004	CONTROL2		Регистр управления приемников 2
0x0000_0008	CONTROL3		Регистр управления приемников 3
0x0000_000C	STATUS1		Регистр состояния приемников 1
0x0000_0010	STATUS2		Регистр состояния приемников 2
0x0000_0014	CONTROL4		Регистр настройки индивидуального делителя частоты каналов 1–4
0x0000_0018	CONTROL5		Регистр настройки индивидуального делителя частоты каналов 5–8
0x0000_001C	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
0x0000_0068	INTMASK		Регистр индивидуальной настройки разрешения прерывания по заполненности FIFO каждого канала
0x0000_0070	CONTROL8		Регистр управления приемников 8
0x0000_0074	CONTROL9		Регистр управления приемников 9
0x1000-0x2000	Память приемников		При CHANNEL=0xF прямой доступ в память приемников

7.20.4.1 **CONTROL1**

Регистр управления приемников 1

Base ADDR=		0x4009_E000			Offset=		0x0000_0000								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV[3:0]				-						CLK[7:2]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK[1:0]		-						CH_EN[7:0]							

Таблица 386 – Описание бит регистра CONTROL1

Бит	Имя	Значение	Описание
31..28	DIV[3:0]		Делитель частоты ядра до 1 МГц 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	CLK[7:0]		Скорость приема данных: Бит 0 – канал 1, бит 7 – канал 8. 1 – частота приема данных = опорная частота/80 (12,5 кГц, если DIV не равен 0); 0 – частота приема данных = опорная частота/10 (100 кГц, если DIV не равен 0); Опорная частота для каждого канала задается делителем в регистрах CONTROL4 и CONTROL5, если DIV=0
13..8			Зарезервировано
7..0	CH_EN[7:0]		Разрешение работы канала: Бит 0 – канал 1, бит 7 – канал 8. 1 – прием по каналу разрешен; 0 – канал приема находится в состоянии сброса

7.20.4.2 **CONTROL2**

Регистр управления приемников 2

Base ADDR=		0x4009_E000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DA	-						SD_EN[7:0]								-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-					LB_EN[7:0]								DIV[6:4]		

Таблица 387 – Описание бит регистра CONTROL2

Бит	Имя	Значение	Описание
31	DA		Бит прямого доступа в FIFO1 и FIFO2: 1 – память приема каналов 1 и 2 работает не в режиме FIFO, доступ к ней осуществляется в диапазоне адресов: 0x400D1000 – 0x400D13FC для FIFO1; 0x400D1400 – 0x400D14FC для FIFO2. 0 – обычный режим работы FIFO; Адрес должен быть кратен четырем, обращения только по 32 разрядным словам. CHANNEL = 0 или 1 в зависимости от канала. При приеме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения
30..25			Зарезервировано
24..17	SD_EN[7:0]		Разрешение декодирования бит данных 9 и 10: Бит 0 – канал 1, бит 7 – канал 8. 1 – разрешено сравнение бит данных 9 и 10 со значением бит SDI1 и SDI2 соответствующего канала; 0 – декодирование отключено, все принятые данные помещаются в FIFO
16..11			Зарезервировано
10..3	LB_EN[7:0]		Разрешение обнаружения меток: Бит 0 – канал 1, бит 7 – канал 8. 1 – разрешено обнаружение меток в первых 8 принятых битах; 0 – обнаружение отключено, все принятые данные помещаются в FIFO
2..0	DIV[6:4]		Делитель частоты ядра до 1 МГц. Содержит старшие три разряда значения, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц

7.20.4.3 **CONTROL3**

Регистр управления приемников 3

Base ADDR=		0x4009_E000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTEHF	INTEFF	INTEER	INTEDR	-						SDI2_1					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDI2_1		-						SDI1_1							

Таблица 388 – Описание бит регистра CONTROL3

Бит	Имя	Значение	Описание
31	INTEHF		Разрешение прерывания FIFO наполовину полно: 1 – разрешено прерывание, если FIFO наполовину полно; 0 – прерывание запрещено
30	INTEFF		Разрешение прерывания FIFO полно: 1 – разрешено прерывание при переполнении FIFO данных; 0 – прерывание запрещено
29	INTEER		Разрешение прерывания ошибка приема: 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); 0 – прерывание запрещено
28	INTEDR		Разрешение прерывания наличие данных в FIFO: 1 – разрешено прерывание, если FIFO приема данных не пусто; 0 – прерывание запрещено
27..22			Зарезервировано
21..14	SDI2[7:0]		Бит сравнения SDI2: Значение бита сравнивается с битом 10 принимаемых данных, если установлен бит SD_EN соответствующего канала. Бит 0 – канал 1, бит 7 – канал 8
13..8			Зарезервировано
7..0	SDI1[7:0]		Бит сравнения SDI1: Значение бита сравнивается с битом 9 принимаемых данных, если установлен бит SD_EN соответствующего канала. Бит 0 – канал 1, бит 7 – канал 8

7.20.4.4 **CONTROL4**

Регистр управления приемников 4

Base ADDR=		0x4009_E000			Offset=		0x0000_0014								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_CH4[7:0]								DIV_CH3[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_CH2[7:0]								DIV_CH1[7:0]							

Таблица 389 – Описание бит регистра CONTROL4

Бит	Имя	Значение	Описание
31..24	DIV_CH4[7:0]		Делитель частоты ядра для получения опорной частоты канала 4
23..16	DIV_CH3[7:0]		Делитель частоты ядра для получения опорной частоты канала 3
15..8	DIV_CH2[7:0]		Делитель частоты ядра для получения опорной частоты канала 2
7..0	DIV_CH1[7:0]		Делитель частоты ядра для получения опорной частоты канала 1

7.20.4.5 **CONTROL5**

Регистр управления приемников 5

Base ADDR=		0x4009_E000			Offset=		0x0000_0018								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_CH8[7:0]								DIV_CH7[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_CH6[7:0]								DIV_CH5[7:0]							

Таблица 390 – Описание бит регистра CONTROL5

Бит	Имя	Значение	Описание
31..24	DIV_CH8[7:0]		Делитель частоты ядра для получения опорной частоты канала 8
23..16	DIV_CH7[7:0]		Делитель частоты ядра для получения опорной частоты канала 7
15..8	DIV_CH6[7:0]		Делитель частоты ядра для получения опорной частоты канала 6
7..0	DIV_CH5[7:0]		Делитель частоты ядра для получения опорной частоты канала 5

7.20.4.6 **CONTROL8**

Регистр управления приемников 8

Base ADDR=		0x4009_E000			Offset=		0x0000_0070								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		DA2	DA1	-						ODD[7:0]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODD[7:0]		-						ENPAR[7:0]							

Таблица 391 – Описание бит регистра CONTROL8

Бит	Имя	Значение	Описание
31	-	-	Зарезервировано
30	-	-	Зарезервировано
29	DA2	0	Бит прямого доступа в FIFO2: 1 – память приема канала 2 работает не в режиме FIFO, доступ к ней осуществляется в диапазоне адресов: 0x400D1400 – 0x400D14FC для FIFO2. 0 – обычный режим работы FIFO. Адрес должен быть кратен четырем, обращения только по 32 разрядным словам. CHANNEL = 1. При приеме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения. Бит DA регистра CONTROL2 должен быть в нуле
28	DA1	0	Бит прямого доступа в FIFO1: 1 – память приема канала 1 работает не в режиме FIFO, доступ к ней осуществляется в диапазоне адресов: 0x400D1000 – 0x400D13FC для FIFO1. 0 – обычный режим работы FIFO. Адрес должен быть кратен четырем, обращения только по 32-х разрядным словам. CHANNEL = 0. При приеме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения. Бит DA регистра CONTROL2 должен быть в нуле
27…22	-	-	Зарезервировано

Бит	Имя	Значение	Описание
21..14	ODD[7:0]	0xFF	Выбор четности или нечетности бита паритета для каналов 1–8: 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); 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[7:0]	0xFF	Разрешение 32 бита паритета для каналов 1–8: 1 – разрешена передача 32-м битом бита паритета; 0 – разрешена передача 32-м битом бита данных. Запрещено сбрасывать этот бит в ноль в штатном режиме работы контроллера

7.20.4.7 **CONTROL9**

Регистр управления приемников 9

Base ADDR=		0x4009_E000			Offset=		0x0000_0074								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								ENSYNC[7:0]							

Таблица 392 – Описание бит регистра CONTROL9

Бит	Имя	Значение	Описание
31..24	-		Зарезервировано
23..16	-		Зарезервировано
15..8	-		Зарезервировано
7..0	ENSYNC[7:0]		Разрешение работы входов приемника в режиме данных и синхросигнала: 1 – разрешено; 0 – запрещено. При установленном бите ENSYNC для соответствующего канала вход IN_A работает как данные (D), вход IN_B как синхросигнал (SYN)

7.20.4.8 **INTMASK**

Регистр индивидуального разрешения прерываний каналов

Base ADDR=		0x4009_E000			Offset=		0x0000_0068								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IEHF8	IEFF8	IEER8	IEDR8	IEHF7	IEFF7	IEER7	IEDR7	IEHF6	IEFF6	IEER6	IEDR6	IEHF5	IEFF5	IEER5	IEDR5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEHF4	IEFF4	IEER4	IEDR4	IEHF3	IEFF3	IEER3	IEDR3	IEHF2	IEFF2	IEER2	IEDR2	IEHF1	IEFF1	IEER1	IEDR1

Таблица 393 – Описание бит регистра INTMASK

Бит	Имя	Значение	Описание
31..28	IEDR8 IEER8 IEFF8 IEHF8	0000	Для канала 8
27..24	IEDR7 IEER7 IEFF7 IEHF7	0000	Для канала 7
23..20	IEDR6 IEER6 IEFF6 IEHF6	0000	Для канала 6
19..16	IEDR5 IEER5 IEFF5 IEHF5	0000	Для канала 5
15..12	IEDR4 IEER4 IEFF4 IEHF4	0000	Для канала 4
11..8	IEDR3 IEER3 IEFF3 IEHF3	0000	Для канала 3
7..5	IEDR2 IEER2 IEFF2 IEHF2	0000	Для канала 2

Бит	Имя	Значение	Описание
3..0	IEDR1 IEER1 IEFF1 IEHF1	0000	Для канала 1: IEDR1: 1 – разрешено прерывание, если FIFO приема данных не пусто; 0 – прерывание запрещено. IEER1: 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); 0 – прерывание запрещено. IEFF1: 1 – разрешено прерывание при переполнении FIFO данных; 0 – прерывание запрещено. IEHF1: 1 – разрешено прерывание, если FIFO наполовину полно; 0 – прерывание запрещено

7.20.4.9 **STATUS1**

Регистр состояния приемников 1

Base ADDR=		0x4009_E000			Offset=		0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-										ERR[7:2]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR[1:0]		-						DR[7:0]							

Таблица 394 – Описание бит регистра STATUS1

Бит	Имя	Значение	Описание
31..22	-		Зарезервировано
21..14	ERR[7:0]		Бит ошибки: Бит 0 – канал 1, бит 7 – канал 8. 0 – нет ошибок; 1 – возникла ошибка приема
13..8	-		Зарезервировано
7..0	DR[7:0]		Бит наличия данных в FIFO: Бит 0 – канал 1, бит 7 – канал 8. 0 – FIFO пусто; 1 – FIFO содержит данные

7.20.4.10 **STATUS2**

Регистр состояния приемников 2

Base ADDR=		0x4009_E000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-										HF[7:2]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HF[1:0]		-						FF[7:0]							

Таблица 395 – Описание бит регистра STATUS2

Бит	Имя	Значение	Описание
31..22			Зарезервировано
21..14	HF[7:0]		Бит наполненности FIFO: Бит 0 – канал 1, бит 7 – канал 8. 0 – FIFO не наполнено до половины; 1 – FIFO наполнено до половины
13..8			Зарезервировано
7..0	FF[7:0]		Бит полноты FIFO: Бит 0 – канал 1, бит 7 – канал 8. 0 – FIFO не полно; 1 – FIFO полно

7.20.4.11 **CHANNEL**

Регистр номера канала приемников

Base ADDR=		0x4009_E000			Offset=		0x0000_001C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-												CHAN[3:0]			

Таблица 396 – Описание бит регистра STATUS3

Бит	Имя	Значение	Описание
31…4	-		Зарезервировано
3…0	CHAN[3:0]		Биты выбора канала. Значение этих бит определяет к данным и меткам, какого канала будет осуществляться доступ: 0000 – канал 1; 0001 – канал 2 ; 0010 – канал 3; 0011 – канал 4; 0100 – канал 5; 0101 – канал 6; 0110 – канал 7; 0111 – канал 8; 1110 – прямой доступ в память каналов 1 и 2 при DR=1, либо доступ по индивидуальным адресам для каждого FIFO

7.20.4.12 **LABEL**

FIFO меток

FIFO меток, с которыми сравниваются первые восемь принимаемых бит, если установлен LB_EN бит соответствующего канала. Размер FIFO для каждого канала 16×8, либо 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.

7.20.4.13 **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.

7.20.4.14 **DATA_R1**
DATA_R2
DATA_R3
DATA_R4
DATA_R5
DATA_R6
DATA_R7
DATA_R8

FIFO принимаемых данных в случае записи в регистр CHANNEL значения 14.

7.20.5 Описание регистров передатчика

Таблица 397 – Описание регистров передатчика

Базовый адрес	Название	Состояние после сброса	Описание
0x400A_0000	ARINC429T0		Контроллер интерфейса передатчиков ARINC429
Смещение			
0x0000_0000	CONTROL1		Регистр управления передатчиков 1
0x0000_0004	CONTROL2		Регистр управления передатчиков 2
0x0000_0008	STATUS		Регистр состояния передатчиков
0x0000_000C	DATA1_T		Регистр передаваемых данных канала 1
0x0000_0010	DATA2_T		Регистр передаваемых данных канала 2
0x0000_0014	DATA3_T		Регистр передаваемых данных канала 3
0x0000_0018	DATA4_T		Регистр передаваемых данных канала 4
0x0000_001C	CONTROL3		Регистр настройки индивидуального делителя частоты каналов
0x0000_0020	CONTROL4		Регистр управления передатчиков 4
0x0000_1000-0x0000_2000	Память передатчиков		Прямой доступ в память передатчиков

7.20.5.1 **CONTROL1**

Регистр управления передатчиками 1

Base ADDR=		0x400A_0000			Offset=		0x0000_0000								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-											INTE_HFT2	INTE_TXR2	INTE_FFT2	INTE_HFT1	INTE_TXR1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTE_FFT1	DIV[6:0]							ODD2	EN_PAR2	CLK2	CH_EN2	ODD1	EN_PAR1	CLK1	CH_EN1

Таблица 398 – Описание бит регистра CONTROL1

Бит	Имя	Значение	Описание
31..21			Зарезервировано
20	INTE_HFT2		Разрешение прерывания при заполнении наполовину буфера FIFO канала 2: 1 – разрешено прерывание, FIFO наполовину полно; 0 – прерывание запрещено
19	INTE_TXR2		Разрешение прерывания при опустошении буфера FIFO канала 2: 1 – разрешено прерывание, буфер FIFO пуст; 0 – прерывание запрещено
18	INTE_FFT2		Разрешение прерывания при полном заполнении буфера FIFO канала 2: 1 – разрешено прерывание при полном заполнении FIFO данных; 0 – прерывание запрещено
17	INTE_HFT1		Разрешение прерывания при заполнении наполовину буфера FIFO канала 1: 1 – разрешено прерывание FIFO наполовину полно; 0 –прерывание запрещено
16	INTE_TXR1		Разрешение прерывания при опустошении буфера FIFO канала 1: 1 – разрешено прерывание FIFO передачи данных пусто; 0 – прерывание запрещено
15	INTE_FFT1		Разрешение прерывания при полном заполнении буфера FIFO канала 1: 1 – разрешено прерывание при полном заполнении FIFO данных; 0 – прерывание запрещено
14..8	DIV[6:0]		Делитель частоты ядра до 1 МГц. Содержит значение, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц
7	ODD2		Выбор четности или нечетности бита паритета для канала 2 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); 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: 1 – разрешена передача 32-м битом бита паритета; 0 – разрешена передача 32-м битом бита данных
5	CLK2		Скорость передачи данных по 2 каналу: 1 – частота передаваемых данных= опорная частота/80 (12,5 кГц если DIV не равен нулю); 0 – частота передаваемых данных= опорная частота/10 (100 кГц если DIV не равен нулю)

Бит	Имя	Значение	Описание
4	CH_EN2		Разрешение работы канала 2: 1 – передача по каналу разрешена; 0 – канал передачи находится в состоянии сброса
3	ODD1		Выбор четности или нечетности бита паритета для канала : 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); 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: 1 – разрешена передача 32-м битом бита паритета; 0 – разрешена передача 32-м битом бита данных
1	CLK1		Скорость передачи данных по 1 каналу: 1 – частота передаваемых данных= опорная частота/80 (12,5 кГц если DIV не равен нулю); 0 – частота передаваемых данных =опорная частота/10 (100 кГц если DIV не равен нулю)
0	CH_EN1		Разрешение работы канала 1: 1 – передача по каналу разрешена; 0 – канал передачи находится в состоянии сброса

7.20.5.2 **CONTROL2**

Регистр управления передатчиков 2

Base ADDR=		0x400A_0000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-											INTE_HFT4	INTE_TXR4	INTE_FFT4	INTE_HFT3	INTE_TXR3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTE_FFT3	-							ODD4	EN_PAR4	CLK4	CH_EN4	ODD3	EN_PAR3	CLK3	CH_EN3

Таблица 399 – Описание бит регистра CONTROL2

Бит	Имя	Значение	Описание
31..21			Зарезервировано
20	INTE_HFT4		Разрешение прерывания при заполнении наполовину буфера FIFO канала 4: 1 – разрешено прерывание, FIFO наполовину полно; 0 – прерывание запрещено
19	INTE_TXR4		Разрешение прерывания при опустошении буфера FIFO канала 4: 1 – разрешено прерывание, буфер FIFO пуст; 0 – прерывание запрещено
18	INTE_FFT4		Разрешение прерывания при полном заполнении буфера FIFO канала 4: 1 – разрешено прерывание при полном заполнении FIFO данных; 0 – прерывание запрещено
17	INTE_HFT3		Разрешение прерывания при заполнении наполовину буфера FIFO канала 3: 1 – разрешено прерывание FIFO наполовину полно; 0 –прерывание запрещено
16	INTE_TXR3		Разрешение прерывания при опустошении буфера FIFO канала 3: 1 – разрешено прерывание FIFO передачи данных пусто; 0 – прерывание запрещено
15	INTE_FFT3		Разрешение прерывания при полном заполнении буфера FIFO канала 3: 1 – разрешено прерывание при полном заполнении FIFO данных; 0 – прерывание запрещено
14..8	-		Зарезервировано
7	ODD4		Выбор четности или нечетности бита паритета для канала 4: 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); 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: 1 – разрешена передача 32-м битом бита паритета; 0 – разрешена передача 32-м битом бита данных
5	CLK4		Скорость передачи данных по 4 каналу: 1 – частота передаваемых данных= опорная частота/80 (12,5 кГц если DIV не равен нулю); 0 – частота передаваемых данных= опорная частота/10 (100 кГц если DIV не равен нулю)

Бит	Имя	Значение	Описание
4	CH_EN4		Разрешение работы канала 4: 1 – передача по каналу разрешена; 0 – канал передачи находится в состоянии сброса
3	ODD3		Выбор четности или нечетности бита паритета для канала 3: 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); 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: 1 – разрешена передача 32-м битом бита паритета; 0 – разрешена передача 32-м битом бита данных
1	CLK3		Скорость передачи данных по 3 каналу: 1 – частота передаваемых данных= опорная частота/80 (12,5 кГц если DIV не равен нулю); 0 – частота передаваемых данных =опорная частота/10 (100 кГц если DIV не равен нулю)
0	CH_EN3		Разрешение работы канала 3: 1 – передача по каналу разрешена; 0 – канал передачи находится в состоянии сброса

7.20.5.3 **CONTROL3**

Регистр управления передатчиков 3

Base ADDR=		0x400A_0000			Offset=		0x0000_001C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_CH4[7:0]								DIV_CH3[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_CH2[7:0]								DIV_CH1[7:0]							

Таблица 400 – Описание бит регистра CONTROL3

Бит	Имя	Значение	Описание
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

7.20.5.4 **CONTROL4**

Регистр управления передатчиков 4

Base ADDR=		0x400A_0000			Offset=		0x0000_0020								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ENSYNC[3:0]			

Таблица 401 – Описание бит регистра CONTROL4

Бит	Имя	Значение	Описание
31..24	-		
23..16	-		
15..4	-		
3..0	ENSYNC[3:0]		Разрешение работы выходов передатчика в режиме данных и синхросигнала: 1 – разрешено; 0 – запрещено. При установленном бите ENSYNC для соответствующего канала выход OUT_A работает как данные (D), выход OUT_B как синхросигнал(SYN)

7.20.5.5 **STATUS**

Регистр состояния передатчиков

Base ADDR=		0x400A_0000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		HFT4	FFT4	TX_R4	HFT3	FFT3	TX_R3	-		HFT2	FFT2	TX_R2	HFT1	FFT1	TX_R1

Таблица 402 – Описание бит регистра STATUS

Бит	Имя	Значение	Описание
31..14			Зарезервировано
13	HFT4		Флаг наполненности FIFO канала 4: 1 – FIFO наполнено до половины; 0 – FIFO не наполнено до половины

Бит	Имя	Значение	Описание
12	FFT4		Флаг полноты FIFO канала 4: 1 – FIFO полно; 0 – FIFO неполно
11	TX_R4		Флаг наличия данных в FIFO канала 4: 1 – FIFO пусто; 0 – FIFO содержит данные
10	HFT3		Флаг наполненности FIFO канала 3: 1 – FIFO наполнено до половины; 0 – FIFO не наполнено до половины
9	FFT3		Флаг полноты FIFO канала 3: 1 – FIFO полно; 0 – FIFO неполно
8	TX_R3		Флаг наличия данных в FIFO канала 3: 1 – FIFO пусто; 0 – FIFO содержит данные
7..6			Зарезервировано
5	HFT2		Флаг наполненности FIFO канала 2: 1 – FIFO наполнено до половины; 0 – FIFO не наполнено до половины
4	FFT2		Флаг полноты FIFO канала 2: 1 – FIFO полно; 0 – FIFO неполно
3	TX_R2		Флаг наличия данных в FIFO канала 2: 1 – FIFO пусто; 0 – FIFO содержит данные
2	HFT1		Флаг наполненности FIFO канала 1: 1 – FIFO наполнено до половины; 0 – FIFO не наполнено до половины
1	FFT1		Флаг полноты FIFO канала 1: 1 – FIFO полно; 0 – FIFO неполно
0	TX_R1		Флаг наличия данных в FIFO канала 1: 1 – FIFO пусто; 0 – FIFO содержит данные

7.20.5.6 **DATA1_T**

FIFO передаваемых данных канала 1

FIFO может содержать данные объемом 256x32 для передачи по каналу 1. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R1, HFT1, FFT1.

7.20.5.7 **DATA2_T**

FIFO передаваемых данных канала 2

FIFO может содержать данные объемом 64x32 для передачи по каналу 2. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R2, HFT2, FFT2.

7.20.5.8 **DATA3_T**

FIFO передаваемых данных канала 3

FIFO может содержать данные объемом 64x32 для передачи по каналу 3. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R3, HFT3, FFT3.

7.20.5.9 **DATA4_T**

FIFO передаваемых данных канала 4

FIFO может содержать данные объемом 64x32 для передачи по каналу 4. Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R4, HFT4, FFT4.

7.21 Контроллер интерфейса МКПД (MILx)

В микросхеме имеется два независимых контроллера интерфейса мультиплексного канала передачи данных по ГОСТ Р 52070-2003 (далее МКПД). Каждый из которых, содержит необходимую логику и память для обработки и хранения командных слов и слов данных одного полного сообщения МКПД. Каждый контроллер содержит два канала для приема/передачи сообщений МКПД: 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 шестнадцатиразрядная двухпортовая память, в которой данные хранятся в области памяти, соответствующей подадресу командного слова. В каждом подадресе можно хранить только одно полное сообщение МКПД. 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.

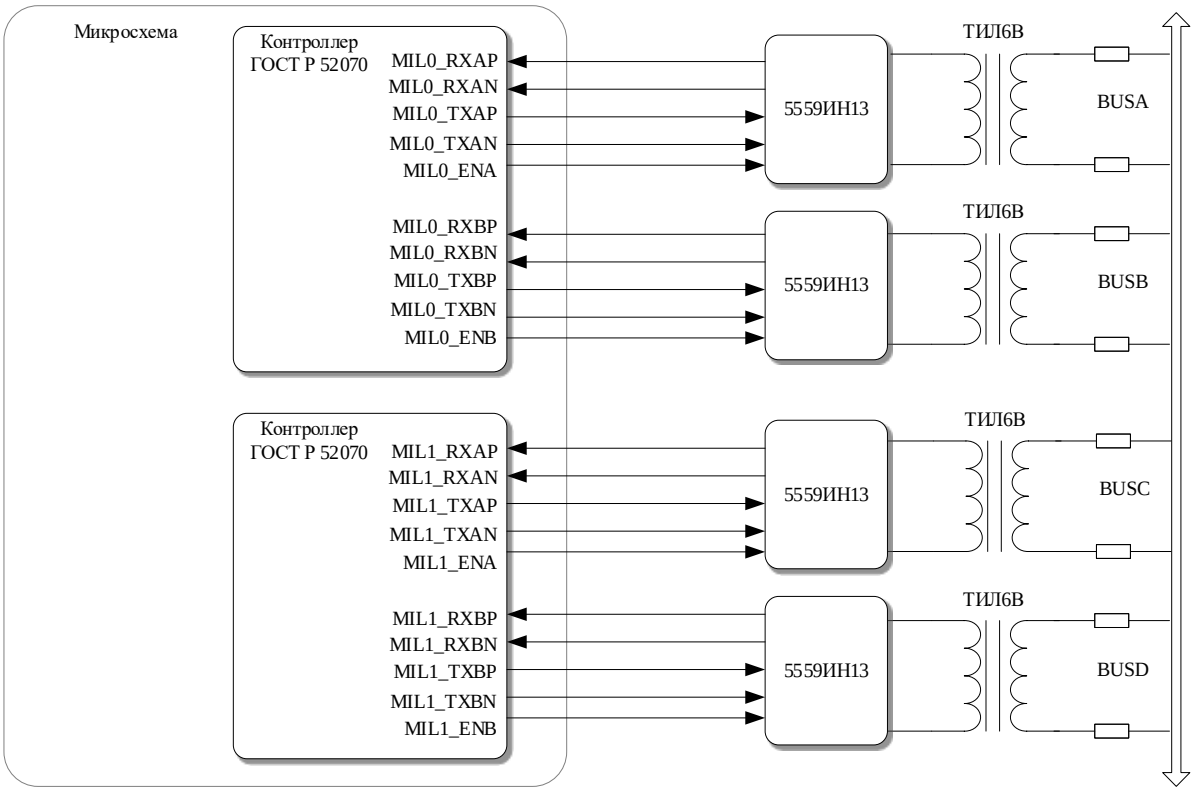


Рисунок 150 – Структурная схема контроллера интерфейса МКПД

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.

7.21.1 Режимы работы

Контроллер поддерживает три режима работы: контроллера шина (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.

7.21.2 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 151, 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.

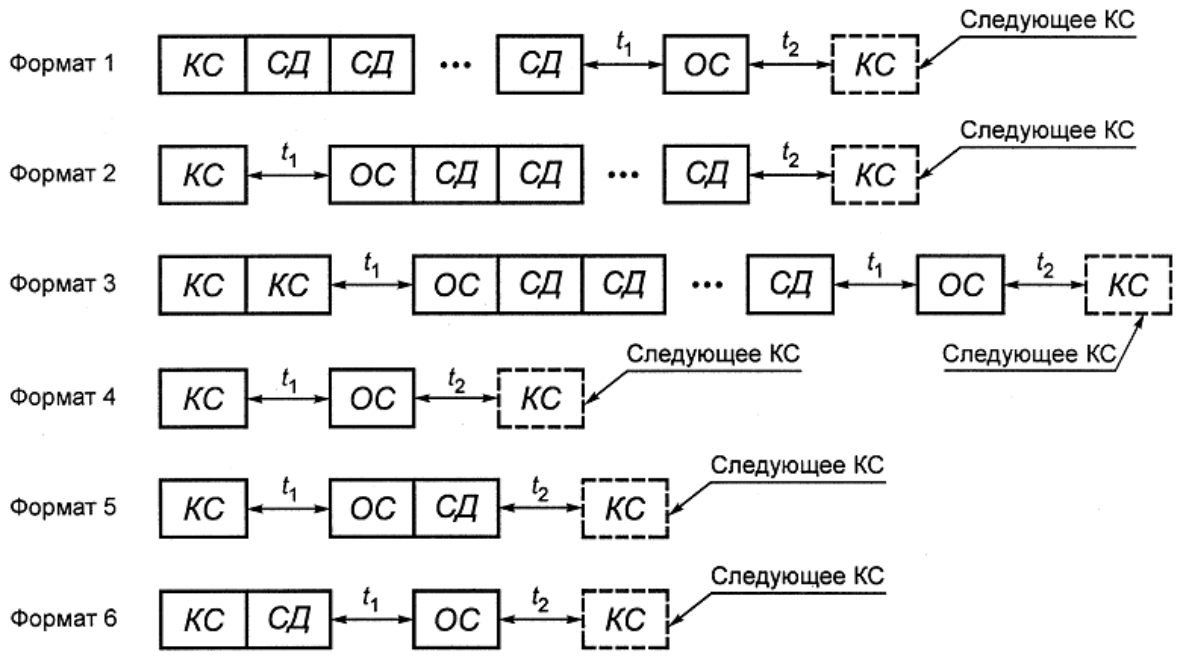


Рисунок 151 – Форматы сообщений (часть 1)

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 152, 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

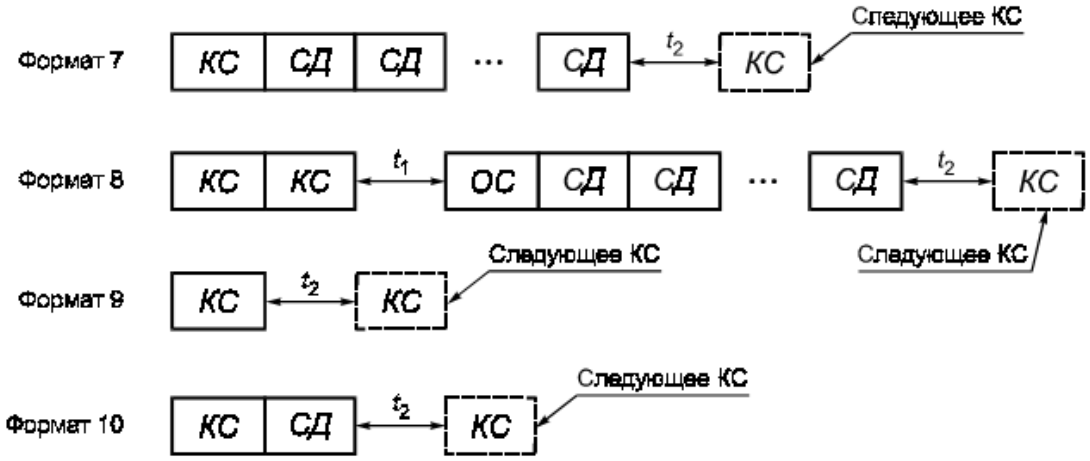


Рисунок 152 – Форматы сообщений (часть 2)

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.

7.21.3 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 ниже (рисунок 153).

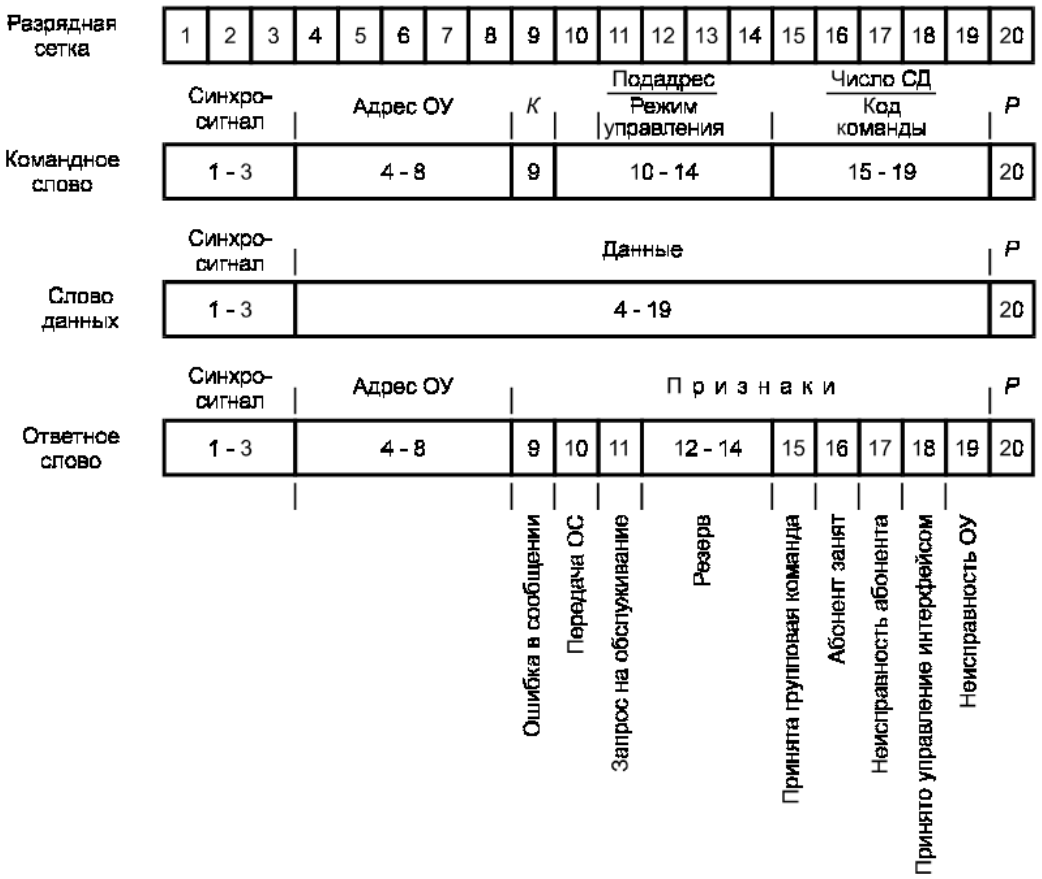


Рисунок 153 – Форматы слов

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 вторая 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 нуль 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 вторая 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.

7.21.4 Структурная схема в режиме КШ

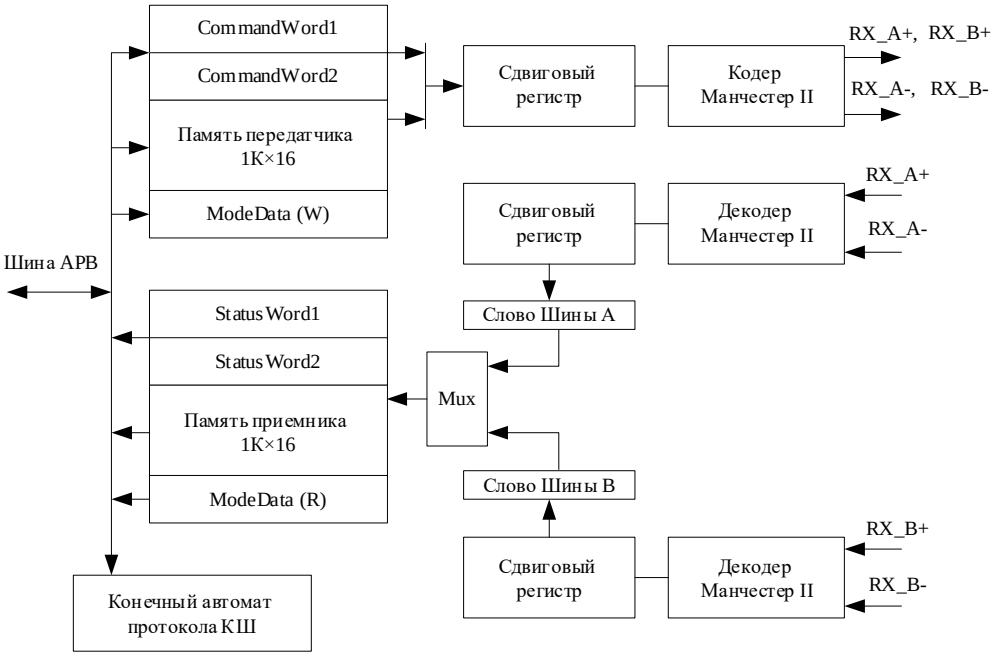


Рисунок 154 – Структурная схема в режиме КШ

7.21.5 Структурная схема в режиме ОУ

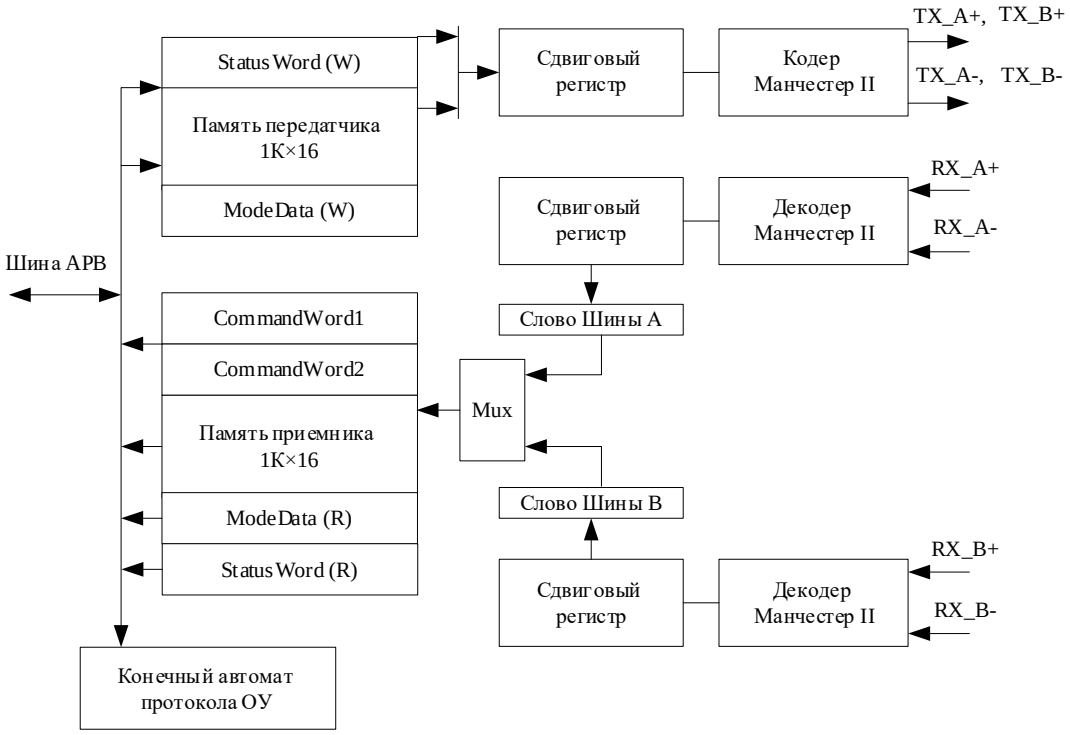


Рисунок 155– Структурная схема в режиме ОУ

7.21.6 Структурная схема в режиме М

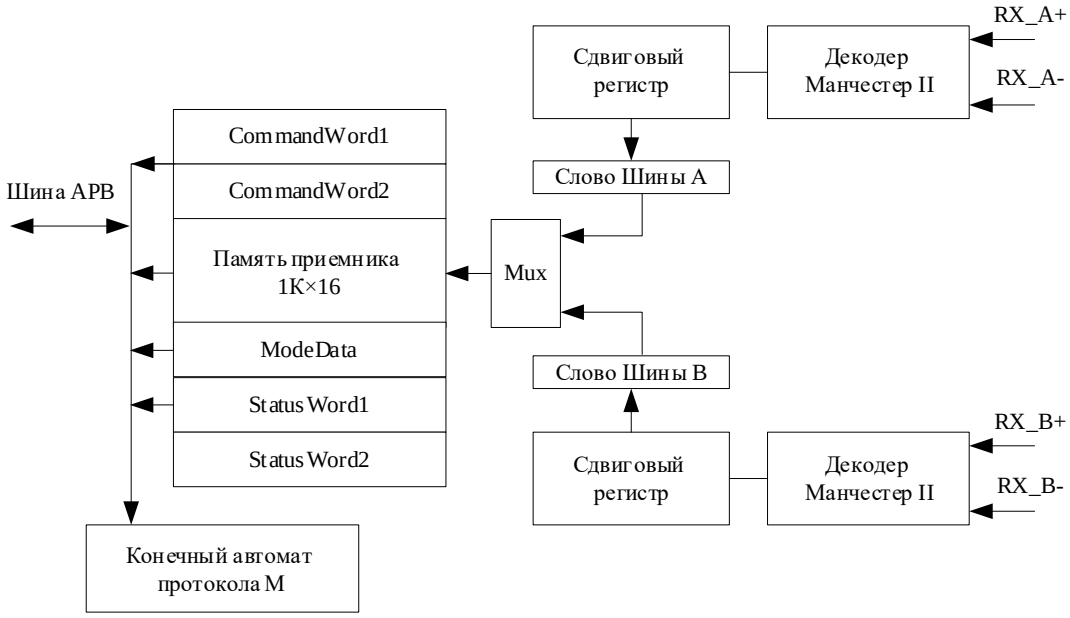


Рисунок 156 – Структурная схема в режиме М

7.21.7 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 нуль. Далее нужно задать в регистре CONTROL значение делителя частоты DIV таким

образом, чтобы при делении частоты ядра HCLK 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. Одновременная установка двух этих бит запрещена.

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

```
*((volatile unsigned int *)(0x400A7000))=0x00000001; //установка бита MR=1
*((volatile unsigned int *)(0x400A7000))=0x00014078;
//RTMODE=1, TRB=TRA=1, RTA=1, DIV=40
```

Пример инициализации КШ

```
*((volatile unsigned int *)(0x400A7000))=0x00000001; //установка бита MR=1
*((volatile unsigned int *)(0x400A7000))=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.

7.21.8 Прием и передача в режиме ОУ

Для того, чтобы настроить контроллер в режиме ОУ, необходимо выполнить все пункты, описанные в пункте 7.21.7 «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

```
*((volatile unsigned int *)(0x400A7018))=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

```
*((volatile unsigned int *)(0x400A7014))=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. Без обновления при запросе данных КШ будут переданы последние записанные в эти области памяти данные. 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 16-тразрядные, 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;
for (i=1;i<=32;i++)
{
*((volatile unsigned int *)(0x400A6000+addon))=i;
addon = addon +4;
}
```

Из примера видно, что стартовый адрес памяти СД для подадреса 1 – 0x80, для последующих подадресов: n*0x80, где n-номер подадреса (n=1-31).

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=*((volatile unsigned int *)(0x400A7014));
```

В переменную *i* будет прочитано значение слова данных команды управления.

Пример чтения данных для подадреса 1

```
addon=0x80;
for (i=1;i<=32;i++)
{
mas[i] = *((volatile unsigned int *)(0x400A6000+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.

7.21.9 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

```
*((volatile unsigned int *)(0x400A700C))=0x00000820; //Командное слово 1
*((volatile unsigned int *)(0x400A7010))=0x00000000; //Командное слово 2
*((volatile unsigned int *)(0x400A7000))=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 делить 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" аппартно не анализируется.

7.21.10 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.

7.21.11 Описание регистров

Таблица 403 – Описание регистров контроллера интерфейса МКПД

Базовый адрес	Название	Описание
0x400A_6000	MIL-STD-1553B0	Контроллер интерфейса 1553 канал 0
0x400A_8000	MIL-STD-1553B1	Контроллер интерфейса 1553 канал 1
Смещение		
0x0000-0x0FFC	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

7.21.11.1 **CONTROL**

Таблица 404 – 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	ENFILTER	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 405 – Описание бит регистра CONTROL

Бит	Имя	Описание
31..24		Зарезервировано
23	RT_HW	Бит аппаратной поддержки ГОСТ 51765-2001 для режима ОУ: 1 – аппаратная поддержка разрешена; 0 – аппаратная поддержка запрещена. При установке этого бита достаточно программной реализация 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 MILx_RXxP и MILx_RXxN: 1 – инверсия разрешена; 0 – инверсия запрещена. Вход MILx_RXxP инвертируется и коммутируется на отрицательный вход декодера манчестерского кода. Вход MILx_RXxN инвертируется и коммутируется на положительны вход декодера манчестерского кода. 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. Содержит старший разряд делителя, на который необходимо поделить частоту MAN_CLK, чтобы получить 1 МГц
20	ENFILTER	Бит разрешения фильтрации потока NRZ: 1 – фильтрация разрешена; 0 – фильтрация запрещена. В случае применения драйверов с некорректной скважностью и длительностью импульсов NRZ кода, таких как 5559ИН67Т, необходимо устанавливать бит в единицу для корректного приема. В этом случае контроль длительностей импульсов NRZ не осуществляется
19	INVTR	Разрешение инверсии сигналов MIL0_ENA, MIL0_ENB, MIL1_ENA, MIL1_ENB: 1 – инверсия; 0 – прямой выход

Бит	Имя	Описание
18	RERR	Сброс ошибок в режиме ОУ: 1 – ошибки могут быть сброшены только битом MR; 0 – сброс ошибок происходит автоматически, после установки бита IDLE
17…11	DIV[6:0]	Делитель частоты PCLK блока МКИО до 1 МГц. Содержит значение, на которое необходимо поделить частоту PCLK, чтобы получить 1 МГц. Частота PCLK обязательно должна быть не менее 24 МГц, не более 120 МГц и кратна 8. Если PCLK не кратна 8, то DIV[6:3]=(PCLK/8)+1, DIV[2:0]=0, но стабильность приема не гарантируется
10..6	RTA4-RTA0	Адрес оконечного устройства. Содержит адрес, который присвоен устройству, если контроллер работает в режиме оконечного устройства RTMODE=1; BCMODE=0
5	TRB	Блокировка передатчика резервного канала: 1 – передатчик разблокирован; 0 – передатчик заблокирован
4	TRA	Блокировка передатчика основного канала: 1 – передатчик разблокирован; 0 – передатчик заблокирован
3..2	RTMODE BCMODE	Выбор режима работы контроллера: 10 – режим оконечного устройства; 01 – режим контроллера шины; 11 – режим неадресуемого монитора
1	BCSTART	Инициирует передачу сообщения в канал в режиме КШ: 1 – старт сообщения; 0 – стоп сообщения. Сбрасывается в ноль автоматически по завершении сообщения
0	MR	Сброс контроллера: 1 – контроллер сбрасывается в исходное состояние; 0 – разрешение работы контроллера
* Дата изготовления микросхемы в формате ГГНН, где ГГ – год изготовления, НН – неделя изготовления указана на крышке корпуса		

7.21.11.2 **STATUS**

Таблица 406 – 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 407 – Описание бит регистра STATUS

Бит	Имя	Описание
31..11	-	Зарезервировано
10	RCVB_stick*	Признак активности резервного канала: 0 – канал неактивен; 1 – канал активен. Устанавливается при установке бита RCVB. Сбрасывается программной записью нуля, если RCVB=0. В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относится не к своему пакету
9	RCVA_stick*	Признак активности основного канала: 0 – канал неактивен; 1 – канал активен. Устанавливается при установке бита RCVA. Сбрасывается программной записью нуля, если RCVA=0. В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относится не к своему пакету
8	ERR_stick*	Ошибка в сообщении: 0 – нет ошибок; 1 – в последней транзакции возникла ошибка. Устанавливается при установке бита ERR. Сбрасывается программной записью нуля, если ERR=0. В установленном состоянии не формирует прерывание и не влияет на прием и передачу пакетов. Необходимо следить за своевременным сбросом, иначе бит может относится не к своему пакету
7	VALMESS_stick*	Успешное завершение транзакции в канале: 0 – транзакция завершена с ошибкой, либо транзакции нет в канале; 1 – транзакция завершена успешно. Устанавливается при установке бита VALMESS. Сбрасывается программной записью нуля, если VALMESS = 0. В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относится не к своему пакету

Бит	Имя	Описание
6	RFLAGN_stick*	Получено достоверное слово из канала: 0 – нет достоверных слов в канале; 1 – в режиме КШ получено достоверное ответное слово; 1 – в режиме ОУ или М получено достоверное командное слово, ответное слово или слово данных в команде управления. Устанавливается при установке бита RFLAGN. Сбрасывается программной записью нуля, если VALMESS = 0. В установленном состоянии не формирует прерывание. Необходимо следить за своевременным сбросом, иначе бит может относится не к своему пакету
5	RCVB	Признак активности резервного канала: 0 – канал неактивен; 1 – канал активен
4	RCVA	Признак активности основного канала: 0 – канал неактивен; 1 – канал активен
3	ERR	Ошибка в сообщении: 0 – нет ошибок; 1 – в последней транзакции возникла ошибка. 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: 0 – транзакция завершена с ошибкой, либо транзакции нет в канале; 1 – транзакция завершена успешно. В режиме ОУ и М автоматически сбрасывается не менее чем через 4 мкс после установки
1	RFLAGN	Получено достоверное слово из канала: 0 – нет достоверных слов в канале; 1 – в режиме КШ получено достоверное ответное слово; 1 – в режиме ОУ или М получено достоверное командное слово, ответное слово или слово данных в команде управления. Между сообщениями бит автоматически сбрасывается в ноль
0	IDLE	Состояние контроллера: 1 – контроллер в неактивном состоянии; 0 – контроллер в состоянии обмена сообщениями

* В режиме StandAlone биты не доступны

Примечание - Для организации надежного сброса статусных флагов прерываний необходимо учитывать требования, изложенные в подразделе 9.3

7.21.11.3 **ERROR**

Таблица 408 – 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 409 – Описание бит регистра ERROR

Бит	Имя	Описание
31...7		Зарезервировано
6	PROERR	Ошибка в протоколе: 1 – недопустимое слово обнаружено на шине во время обмена сообщениями; 0 – нет ошибок
5	CONERR	Ошибка непрерывности сообщения: 1 – передача сообщения не непрерывная; 0 – нет ошибок
4	GAPERR	Недопустимая активность на шине: 1 – обнаружена активность на шине в интервале 4 мкс после успешного завершения сообщения; 0 – нет ошибок
3	CSYCERR/ SEQERR	Ошибка синхронизации команды в режиме КШ (CSYCERR): 1 – ожидался синхроимпульс команды, а получен синхроимпульс данных; 0 – ошибок нет Ошибка после приема команды в режиме ОУ и М (SEQERR): 1 – обнаружена пауза после приема командного слова с битом 10 в регистре команды 1 (CommandWord1), равным нулю, или обнаружены слова данных после приема командного слова с битом 10 (CommandWord1), равным единице; 0 – ошибок нет
2	DSYCERR/ SYNCERR	Ошибка синхронизации данных в режиме КШ (DSYCERR): 1 – ожидался синхроимпульс данных, а получен синхроимпульс команды; 0 – ошибок нет. Ошибка синхронизации в режиме ОУ и М (SYNCERR): 1 – ожидался синхроимпульс команды, а получен синхроимпульс данных или наоборот; 0 – ошибок нет
1	MANERR	Ошибка декодирования кода NRZ: 1 – ошибка в количестве принятых бит или ошибка в бите контроля четности; 0 – ошибок нет
0	NORCV	Ошибка приема: 1 – не получено ответное слово в интервале 14 мкс или не получены ожидаемые данные; 0 – ошибок нет

7.21.11.4 **CommandWord1**

Таблица 410 – 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 411 – Описание бит регистра CommandWord1

Бит	Имя	Описание
31..16	-	Зарезервировано
15..11	Адрес ОУ	Адрес оконечного устройства, которому предназначено командное слово
10	Прием/передача	Бит приема/передачи: 1 – режим работы ОУ-КШ; 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 Р52070-2003
Примечание – В режиме ОУ и М регистр доступен только на чтение, в режиме КШ – только на запись		

7.21.11.5 **CommandWord2**

Таблица 412 – 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 413 – Описание бит регистра CommandWord2

Бит	Имя	Описание
31..16		Зарезервировано
15..11	Адрес ОУ	Адрес оконечного устройства, которому предназначено командное слово
10	Прием/передача	Бит приема/передачи 1 – режим работы ОУ-ОУ 0 – командное слово не используется

Бит	Имя	Описание
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		

7.21.11.6 **ModeData**

Таблица 414 – Регистр слова данных команды управления ModeData

Номер	31...16	15…0
Доступ	U	R/W
Сброс	0	0
	-	Слово данных команды управления

Таблица 415 – Описание бит регистра ModeData

Бит	Имя	Описание
31..16		Зарезервировано
15…0	ModeData	Содержит принятое или передаваемое слово данных в команде управления

7.21.11.7 **StatusWord1**

Таблица 416 – 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 417 – Описание бит регистра StatusWord1

Бит	Имя	Описание
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

Бит	Имя	Описание
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
Примечание – В режиме ОУ регистр по записи содержит предназначенное для передачи КШ ответное слово, а по чтению содержит ОС, полученное от передающего ОУ в транзакции ОУ-ОУ. В режиме КШ и М регистр доступен только на чтение и содержит принятое от ОУ ответное слово		

7.21.11.8 **StatusWord2**

Таблица 418 – 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 419 – Описание бит регистра StatusWord2

Бит	Имя	Описание
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
Примечание – В режиме КШ и М регистр доступен только на чтение и содержит ОС, передающего ОУ в транзакции ОУ-ОУ. В режиме ОУ регистр не используется		

7.21.11.9 **INTEN**

Таблица 420 – 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 421 – Описание бит регистра INTEN

Бит	Имя	Описание
31...4		Зарезервировано
3	ERRIE	Прерывание при возникновении ошибки в сообщении: 0 – прерывание маскировано; 1 – прерывание разрешено, это позволяет генерировать прерывание при возникновении ошибок в сообщении
2	VALMESSIE	Прерывание при успешном завершении транзакции в канале: 0 – прерывание маскировано; 1 – прерывание разрешено, это позволяет генерировать прерывание при успешном завершении обмена данными в канале
1	RFLAGNIE	Прерывание при приеме достоверного слова: 0 – прерывание маскировано; 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: 0 – прерывание маскировано; 1 – прерывание разрешено, это позволяет генерировать прерывание по переходу контроллера в неактивное состояние

7.21.11.10 **MSG**

Регистр декодирования сообщений

Таблица 422 – Регистр декодирования сообщений MSG

Номер	31…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 423 – Описание бит регистра MSG

Бит	Имя	Значение	Описание
31…14	-		Зарезервировано
13…0	Код сообщения	0x0000	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 424 – 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	не F 0 00001-11110 XXXXX
0200 Команда передачи ОУ-ОУ, групповая	11111 0 00001-11110 XXXXX	RTA 1 00001-11110 XXXXX
Команда 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	

7.21.11.11 **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 2К×16.

7.22 Контроллер АЦП (ADCx)

Микросхема содержит два независимых блока АЦП. Каждый АЦП единовременно может осуществлять преобразование от одного из 64-х каналов. Всего в системе 18 внешних аналоговых каналов и 16 внутренних. Структурная схема АЦП представлена на рисунке 157. Распределение каналов по АЦП представлено в таблице 425.

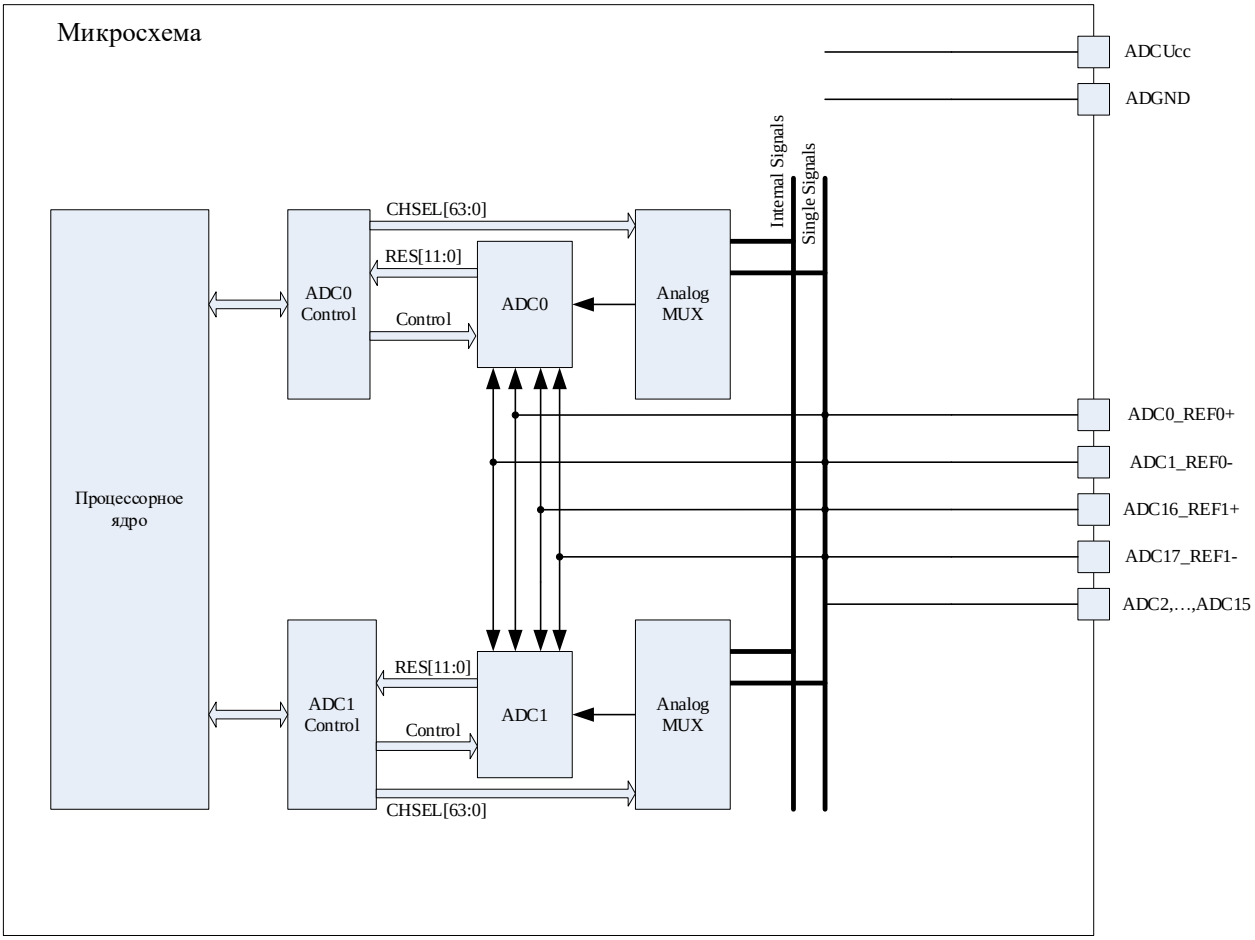


Рисунок 157 – Структурная схема контроллера АЦП

Таблица 425 – Распределение каналов по АЦП

Источник сигнала		ADC0	ADC1	Описание
Тип	Имя			
	ADC0_REF0+		Канал 0	Внешний вход АЦП/ Внешнее опорное напряжение
	ADC1_REF0-		Канал 1	Внешний вход АЦП/ Внешнее опорное напряжение
	ADC2	Канал 0	Канал 2	Внешний вход АЦП
	ADC3	Канал 1	Канал 3	Внешний вход АЦП
	ADC4	Канал 2	Канал 4	Внешний вход АЦП
	ADC5	Канал 3	Канал 5	Внешний вход АЦП
	ADC6	Канал 4	Канал 6	Внешний вход АЦП

Источник сигнала		ADC0	ADC1	Описание
Тип	Имя			
	ADC7	Канал 5	Канал 7	Внешний вход АЦП
	ADC8	Канал 6	Канал 8	Внешний вход АЦП
	ADC9	Канал 7	Канал 9	Внешний вход АЦП
	ADC10	Канал 8	Канал 10	Внешний вход АЦП
	ADC11	Канал 9	Канал 11	Внешний вход АЦП
	ADC12	Канал 10	Канал 12	Внешний вход АЦП
	ADC13	Канал 11	Канал 13	Внешний вход АЦП
	ADC14	Канал 12	Канал 14	Внешний вход АЦП
	ADC15	Канал 13	Канал 15	Внешний вход АЦП
	ADC16_REF1+	Канал 14		Внешний вход АЦП/ Внешнее опорное напряжение
	ADC17_REF1-	Канал 15		Внешний вход АЦП /Внешнее опорное напряжение
	DUCC0	Канал 16		Напряжение питания домена 0
	DUCC1	Канал 17		Напряжение питания домена 1
	DUCC2	Канал 18		Напряжение питания домена 2
	DUCC3	Канал 19		Напряжение питания домена 3
	DAC0_OUT	Канал 20		Выход ЦАП 0
	DAC1_OUT	Канал 21		Выход ЦАП 1
	DUcc_B	Канал 22		Напряжение питания батарейного домена
	TEMP_0	Канал 23		Датчик температуры 0
	DUccPLL		Канал 16	Напряжение питания домена PLL
	CURRDUCC		Канал 17	Ток потребления батарейного домена
	CURRDUCC0		Канал 18	Ток потребления LDO0
	CURRDUCC1		Канал 19	Ток потребления LDO1
	CURRDUCC2		Канал 20	Ток потребления LDO2
	CURRDUCC3		Канал 21	Ток потребления LDO3
	CURRDUCCPLL		Канал 22	Ток потребления домена PLL
	TEMP_1		Канал 23	Датчик температуры 1

Примечание – 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

7.22.1 Преобразуемые сигналы

7.22.1.1 **Внутренние сигналы**

Каждый АЦП может преобразовать внутренние сигналы:

- напряжение питания $DU_{CC}0$ (канал 16 АЦП 0);
- напряжение питания $DU_{CC}1$ (канал 17 АЦП 0);
- напряжение питания $DU_{CC}2$ (канал 18 АЦП 0);
- напряжение питания $DU_{CC}3$ (канал 19 АЦП 0);
- выходное напряжения DAC0_OUT (канал 20 АЦП 0);
- выходное напряжения DAC1_OUT (канал 21 АЦП 0);
- напряжение питания DU_{CC}_B (канал 22 АЦП 0);
- датчик температуры 0 (канал 23 АЦП 0);
- напряжение питания DU_{CC}PLL (канал 16 АЦП 1);
- ток потребления DU_{CC}_B (канал 17 АЦП 1);
- ток потребления $DU_{CC}0$ (канал 18 АЦП 1);
- ток потребления $DU_{CC}1$ (канал 19 АЦП 1);
- ток потребления $DU_{CC}2$ (канал 20 АЦП 1);
- ток потребления $DU_{CC}3$ (канал 21 АЦП 1);
- ток потребления $_{CC}$_PLL (канал 22 АЦП 1);
- датчик температуры 1 (канал 23 АЦП 1).

Примечание – для токов потребления только косвенное измерение. Для каждого образца требуется проведение калибровки шкалы.

7.22.1.2 **Внешние одиночные сигналы**

Внешними сигналами АЦП могут являться до 18-ти выводов микросхемы (выводы ADC0_REF0+…ADC17_REF1-). При этом часть выводов может быть преобразована на обоих АЦП (выводы ADC2…ADC15), часть только на ADC0 (выводы ADC16_REF1+, ADC17_REF1-), и часть только на ADC1 (выводы ADC0_REF0+, ADC1_REF0-). Для проведения преобразования должен быть выбран номер канала, соответствующего выводу, и произведен запуск АЦП.

7.22.2 Источники опорного напряжения

Для проведения преобразования может быть использован следующий способ формирования опорного напряжения:

- REFMODE = 0

В качестве опорного напряжения преобразования выступают напряжение питания $ADCU_{CC}$ и общий ADCGND:

- REFMODE = 1

В качестве опорного напряжения преобразования выступают дополнительные сигналы, задаваемые битами REFSEL[5:0].

7.22.3 Синхронизация АЦП

Работа контроллера АЦП и само преобразование в блоке АЦП выполняется на тактовой частоте ADCCLK, формируемой в блоке CLKCNTR. При этом запись и чтение регистров управления, регистров статуса и результатов осуществляется на частоте PCLK. В общем случае эти частоты могут быть асинхронными, таким образом, если необходимо обеспечить минимальный джиттер начала преобразования по какому-либо внешнему сигналу или программному запуску, необходимо учитывать соотношение рабочей частоты ADCCLK и PCLK.

Если частота SELCLK, из которой формируется частота ADCCLK, превышает предельно-допустимое значение f_{C_ADC}, для обеспечения корректной длительности сигнала высокого уровня частоты ADCCLK необходимо в регистре ADCn_CLK контроллера тактовых частот установить бит DIV2 в 1 (независимо от выбранного значения поля DIV).

7.22.4 Запуск преобразования

Преобразование АЦП могут быть начато по следующим событиям:

– одиночное преобразование по записи бита GO;
– непрерывное преобразование по завершении предыдущего;
– одиночное преобразование по фронту сигнала EXT_GO. В качестве источников сигнала EXT_GO могут выступать:
  – события на внешних выводах;
  – события в таймерах;
  – сигналы ШИМ таймеров;
  – сигнал от второго АЦП.

Описание событий приведено в таблице 426.

Таблица 426 – События запуска преобразований

Бит EXT_GO_SEL	ADC0	Описание	ADC1	Описание
0	TMR0_EVENT	CNT==ARR в TMR0	TMR0_EVENT	CNT==ARR в TMR0
1	TMR1_EVENT	CNT==ARR в TMR1	TMR1_EVENT	CNT==ARR в TMR1
2	TMR2_EVENT	CNT==ARR в TMR2	TMR2_EVENT	CNT==ARR в TMR2
3	TMR3_EVENT	CNT==ARR в TMR3	TMR3_EVENT	CNT==ARR в TMR3
4	–	Зарезервировано	–	Зарезервировано
5	–	Зарезервировано	–	Зарезервировано
6	TMR0_CH1o	Сигнал ШИМ канала 1 в TMR0	TMR0_CH3o	Сигнал ШИМ канала 3 в TMR0
7	TMR1_CH1o	Сигнал ШИМ канала 1 в TMR1	TMR1_CH3o	Сигнал ШИМ канала 3 в TMR1
8	TMR2_CH1o	Сигнал ШИМ канала 1 в TMR2	TMR2_CH3o	Сигнал ШИМ канала 3 в TMR2

Бит EXT_GO_SEL	ADC0	Описание	ADC1	Описание
9	TMR3_CH1o	Сигнал ШИМ канала 1 в TMR3	TMR3_CH3o	Сигнал ШИМ канала 3 в TMR3
10	–	Зарезервировано	–	Зарезервировано
11	–	Зарезервировано	–	Зарезервировано
12	PORTA[0]	Сигнал на выводе PA[0]	PORTA[8]	Сигнал на выводе PA[8]
13	PORTA[16]	Сигнал на выводе PA[16]	PORTA[24]	Сигнал на выводе PA[24]
14	PORTB[0]	Сигнал на выводе PB[0]	PORTB[8]	Сигнал на выводе PB[8]
15	PORTB[16]	Сигнал на выводе PB[16]	PORTB[24]	Сигнал на выводе PB[24]
16	PORTC[0]	Сигнал на выводе PC[0]	PORTC[8]	Сигнал на выводе PC[8]
17	PORTC[16]	Сигнал на выводе PC[16]	PORTC[24]	Сигнал на выводе PC[24]
18	PORTD[0]	Сигнал на выводе PD[0]	PORTD[8]	Сигнал на выводе PD[8]
19	PORTD[16]	Сигнал на выводе PD[16]	PORTD[24]	Сигнал на выводе PD[24]
20	PORTE[0]	Сигнал на выводе PE[0]	PORTE[8]	Сигнал на выводе PE[8]
21	PORTE[16]	Сигнал на выводе PE[16]	PORTE[24]	Сигнал на выводе PE[24]
22	ADC0_IF	Окончание преобразования АЦП 0	ADC0_IF	Окончание преобразования АЦП 0
23	ADC1_IF	Окончание преобразования АЦП 1	ADC1_IF	Окончание преобразования АЦП 1
Примечание – для запуска преобразования по сигналу от внешних выводов необходимо, что бы они были сконфигурированы в цифровой режим, при этом выполняемая функция не имеет значения (вывод может работать как на вход, так и на выход)				

7.22.5 Непрерывное преобразование

Запуск непрерывного преобразования осуществляется выставлением соответствующего режима в поле WORKMODE[2:1] регистра CONFIG0. Сам блок АЦП всегда необходимо включать (ADCEN=1), до выставления режима непрерывного преобразования, поскольку старт преобразования происходит сразу после записи значения 0x2 в поле WORKMODE[2:1]. Каналы, используемые в непрерывном

преобразовании, определяются битом SELMODE регистра CONFIG0. При SELMODE=0 происходит непрерывное преобразование канала, указанного в поле CHANNEL[13:8] регистра CONTROL. При SELMODE=1 происходит цикличное преобразование каналов выставленных в регистрах CHSEL0 и CHSEL1. Переключение каналов происходит от младшего к старшему и затем цикл повторяется. Для непрерывного преобразования с перебором нескольких каналов необходимо сначала включить блок АЦП, выставив бит ADCEN=1. Затем в регистрах CHSEL0 и CHSEL1 установить номера каналов, участвующих в переборе, после чего установить бит SELMODE=1. Последним шагом необходимо запустить преобразование, выставив значение поля WORKMODE[2:1]=0x2.

7.22.6 Последовательный опрос каналов АЦП

АЦП может осуществлять автоматическое переключение каналов при преобразовании:

– SELMODE = 0, автоматическое переключение не производится;

– SELMODE = 1, после завершения преобразования автоматически производится переключение на следующий канал (номера каналов, участвующих в автоматическом переключении, задаются в регистре AUTO_CH_SEL).

7.22.7 Получение результатов преобразования

Результаты преобразования АЦП сохраняются в выходном FIFO размером 16 записей. Каждая запись содержит в себе поля значения результата и поле номера канала преобразования. При наличии данных в FIFO-очереди формируется флаг наличия результатов ADC_NE, который может вызвать соответствующий обработчик прерывания либо опрошен программно. При переполнении FIFO-очереди формируется флаг ошибки ADC_OF. При этом происходит приостановка записи в FIFO. В регистрах RESULTCHx всегда отображается последнее значение, полученное при преобразовании соответствующего канала (не зависимо от настройки контроля уровней). Сброс FIFO флагом FIFOCLR регистра CONFIG0 допустимо производить только при отключенной частоте ADCCLK.

7.22.8 Обработка результатов с помощью DMA

При необходимости результаты работы АЦП могут быть обработаны с помощью DMA. При использовании DMA не допускается программное считывание данных из FIFO. Блок АЦП формирует 5 различных запросов DMA с программируемой привязкой к каналам. Для запросов DMA_REQ0, DMA_REQ1, DMA_REQ2 и DMA_REQ3 задается режим работы с жесткой привязкой к конкретному номеру канала, DMA_REQ4 обрабатывает все остальные каналы. Не должна допускаться обработка одного канала двумя запросами, либо наличия канала без обработки.

7.22.9 Преобразование с контролем границ

Для обработки событий попадания уровня сигнала в заданную область можно применить режим работы с автоматическим контролем границ значения результата:

– контроль границ не выполняется, все результаты попадают в FIFO;

– контроль границ выполняется, и в FIFO попадают результаты, если результирующее значение более или равно MIN_LEVEL и менее или равно MAX_LEVEL;

– контроль границ выполняется, и в FIFO попадают результаты, если результирующее значение менее или равно MIN_LEVEL, или более или равно MAX_LEVEL.

7.22.10 Диаграмма работы АЦП

На рисунке 158 представлена временная диаграмма работы АЦП. Длительность фаз PAUSE и SETUP задается в тактах ADCCLK в соответствующих полях регистра CONFIG1. Любая из фаз или обе фазы могут быть пропущены записью 0 в соответствующее поле регистра CONFIG1. Фаза SETUP позволяет увеличить время заряда внутренней емкости АЦП. Минимальное время заряда внутренней емкости АЦП без использования фазы SETUP – шесть тактов ADCCLK. Следующие 36 тактов, пока АЦП выполняет преобразование, вход отключен от АЦП. Минимальный цикл АЦП в режиме непрерывного преобразования составляет 43 такта ADCCLK.

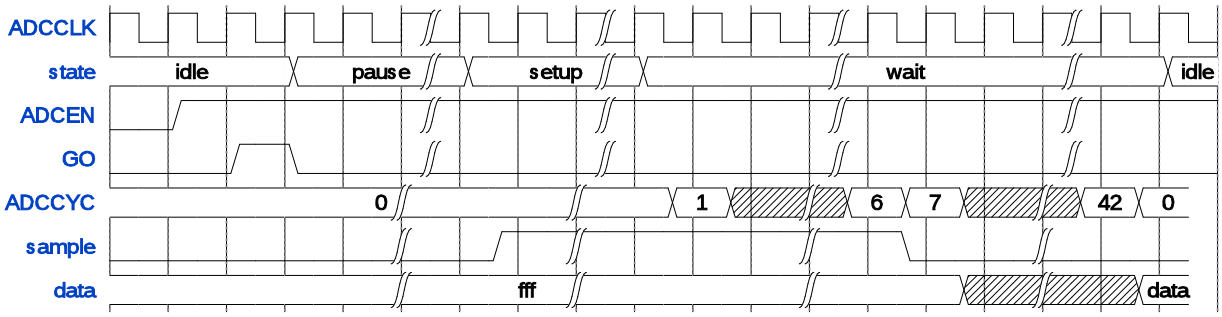


Рисунок 158 – Диаграмма работы АЦП

7.22.11 Схема входного каскада АЦП

На рисунке 159 представлена эквивалентная схема входного каскада АЦП в режиме семплирования входа (соответствует активному уровню сигнала sample на рисунке 158). Значение параметров R0 < 4,5 кОм, C0 = 6,84 пФ необходимо учитывать при выборе тактовой частоты ADCCLK (f_{C_ADC}) и количества циклов фазы SETUP для получения требуемой точности.

Для достижения 12-битной точности должно выполнятся следующее соотношение

$$(Rin + R0) \cdot C0 \leq \frac{t_{setup}}{\ln(2^{12})}, \qquad (20)$$

где Rin – сопротивление источника сигнала;

t_{setup} – длительность фазы SETUP (см. рисункок 158), задаваемой битами SETUP[8:0].

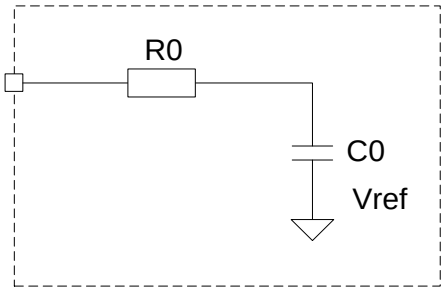


Рисунок 159 – Схема входного каскада АЦП в режиме семплирования входа

В АЦП встроено минимальное число циклов фазы SETUP = 6. Таким образом, например, для частоты преобразования АЦП f_{CONV} = 0,25 МВыб/с с нулевым количеством дополнительных циклов фазы SETUP необходимо задать тактовую частоту ADCCLK равной

$$f_{\text{C_ADC}} = \frac{43}{4\ \text{мкс}} = 10{,}75\ \text{МГц},$$

при этом время зарядки входной емкости будет равно

$$t_{setup} = (0 + 6) \cdot \frac{4\ \text{мкс}}{43} = 558\ \text{нс}.$$

Приведенные выше формула (20) и расчеты справедливы для режима непрерывного преобразования с внешним источником сигнала на одном канале (SELMODE = 0).

При синхронной смене канала и начале преобразования одной записью в регистр CONTROL или при автоматическом переборе каналов (SELMODE = 1) длительность фазы SETUP необходимо выбирать, исходя из максимального требуемого времени зарядки среди выбранных каналов, но не менее 1 мкс в дополнение к встроенным 6 тактам. Причем время 1 мкс соответствует худшему перепаду между каналами «Ucc» – «GND» при лучшем быстродействии (RH_MODE = 1). Для других конфигураций бит минимальное необходимое время оценивается по формуле

$$t_{\text{setup}} = \frac{U_{CH} \cdot 6{,}84\ \text{пФ}}{2 \cdot I_{\text{ИОТ}}}, \qquad (21)$$

где U_{CH} – разница напряжений между соседними каналами;

$I_{ИОТ}$ – ток источника опорного тока варьируется от ~5 до ~15 мкА в зависимости от битов SELR, REF_TRIMR, REFTRIM.

Если биты SELR, REF_TRIMR, REFTRIM равны значениям по сбросу, то время зарядки емкости АЦП, время переключения каналов, время заряда при опросе датчика температуры будут выше номинального. Это можно компенсировать пропорциональным увеличением длительности фазы SETUP. Конкретные примеры приведены ниже.

Пример 1 – если в режиме автоматического перебора каналов в список опрашиваемых каналов включен термодатчик и канал от внешнего источника сигнала с выходным сопротивлением 20 кОм, то согласно формуле (21) для внешнего источника

необходимая длительность фазы SETUP составит 1,4 мкс. Но для термодатчика требуется длительность 5 мкс (при DT_MODE = 0). Значит, SETUP нужно прописать не менее 34 + 6 «встроенных» тактов (для тактовой частоты ADCCLK 8 МГц).

Пример 2 – если в режиме автоматического перебора каналов в список опрашиваемых каналов включено два канала от внешнего источника сигнала с выходным сопротивлением 1 Ом у обоих, то согласно формуле (20) получим время меньше 1 мкс. Но так как идет речь об автоматическом переборе каналов, то SETUP нужно прописать равным не менее восьми без учета шести «встроенных» тактов (для тактовой частоты ADCCLK 8 МГц).

Примечание – При учете рекомендаций по настройке регистров АЦП корректность выбора значения параметра SETUP можно дополнительно проверить эмпирически. Если при увеличении значения SETUP в два раза результат выполнения преобразования не изменится, то выбранное значение можно считать верным.

7.22.12 Описание регистров

Таблица 427 – Описание регистров контроллера АЦП

Базовый адрес	Название	Состояние после сброса	Описание
0x400A_A000	ADC0		Контроллер ADC0
0x400A_B000	ADC1		Контроллер ADC1
Смещение			
0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
0x0000_0004	CONFIG0		Регистр настройки работы контроллера АЦП
0x0000_0008	CONFIG1		Регистр настройки работы контроллера АЦП
0x0000_000C	CONFIG2		Регистр настройки работы контроллера АЦП
0x0000_0010	CONTROL		Регистр управления контроллера АЦП
0x0000_0014	STATUS		Регистр статуса контроллера АЦП, регистр недоступен до записи ключа
0x0000_0018	RESULT		Регистр выхода FIFO результатов преобразования АЦП, регистр недоступен до записи ключа
0x0000_001C	RESULTCH0		Регистр последнего значения, полученного по каналу 0, регистр недоступен до записи ключа
	…		
0x0000_0118	RESULTCH63		Регистр последнего значения, полученного по каналу 63, регистр недоступен до записи ключа
0x0000_011C	MAX_LEVEL		Регистр верхней границы преобразования
0x0000_0120	MIN_LEVEL		Регистр нижней границы преобразования
0x0000_0124	CHSEL0		Регистр задания каналов автоматического перебора

Базовый адрес	Название	Состояние после сброса	Описание
0x0000_0128	CHSEL1		Регистр задания каналов автоматического перебора
0x0000_012C	DMAREQ		Регистр задания запросов DMA
0x0000_0130	STATRCH0		Статус приема данных с АЦП
0x0000_0134	STATRCH1		Статус приема данных с АЦП
0x0000_0138	FIFOEN0		Разрешение записи в FIFO
0x0000_013C	FIFOEN1		Разрешение записи в FIFO

7.22.12.1 **KEY**

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0000								
REG Name:		KEY													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
RW-0															

Таблица 428 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока ADC_CNTR

7.22.12.2 **CONFIG0**

Регистр конфигурации 0

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0004								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXT_GO_SEL[23:8]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXT_GO_SEL [7:0]								EXT_GO_INV	FIFOCLR	SELMODE	RH_MODE	REFMODE	WORKMODE [1:0]		ADCEN
RW-0								RW-0	W-1	RW-0	RW-0	RW-0	RW-0		RW-0

Таблица 429 – Описание бит регистра CONFIG0

Бит	Имя	Значение	Описание
31...8	EXT_GO_SEL[23:0]	0x000000	Биты выбора источника запуска АЦП по внешнему событию EXT_GO: 0000_0000_0000_0000_0000_0000 запрет всех источников (*по умолчанию*); 0000_0000_0000_0000_0000_0001 вывод EXT_GO[0] (TMR0_EVENT); 0000_0000_0000_0000_0000_0010 сигнал EXT_GO[1] (TMR1_EVENT); 0000_0000_0000_0000_0000_0100 сигнал EXT_GO[2] (TMR2_EVENT). Всего 24 источника (см. таблицу 426)
7	EXT_GO_INV	0	Управляемая инверсия сигнала EXT_GO: 0 – сигнал не инвертируется (по умолчанию); 1 – сигнал инвертируется
6	FIFOCLR	0	Запись «1» в поле сбрасывает единовременно FIFO
5	SELMODE	0	Режим автоматического переключения каналов: 0 – номер канала по завершении преобразования не переключается (по умолчанию); 1 – по завершении преобразования производится автоматическое переключение на следующий канал, заданный в CHSEL
4	RH_MODE	0	Выбор режима задания опорного тока для АЦП: 0 – стандартный режим. АЦП использует опорный ток от ИОТ (настраиваемого битами SELR, REFTRIM, REF_TRIMR); 1 – быстрый режим. АЦП использует ток, не зависящий от битов SELR, REFTRIM, REF_TRIMR. Данный режим позволяет добиться максимального быстродействия при автоматическом переборе каналов. Подробнее см. пункт 7.22.11 «Схема входного каскада АЦП». Кроме того, этот режим позволяет получать валидные отчеты не через 20 мкс, а через 2 мкс
3	REFMODE	0	Выбор источника опорного напряжения для преобразования: 0 – REF+ и REF- задаются напряжением питания АЦП (ADCUcc и ADCGND); 1 – REF+ и REF- задаются отдельными сигналами, задаваемыми битами REFSEL. Данный режим позволяет достичь лучших характеристик «ошибка смещения» (E_{OFFADC}) и «ошибка усиления» ($E_{GAINDAC}$)

Бит	Имя	Значение	Описание
2,1	WORKMODE[1:0]	00	Режим работы АЦП: 00 – запуск преобразования по записи бита GO (регистр CONTROL.GO) (по умолчанию); 01 – запуск преобразования по фронту сигнала EXT_GO; 10 – непрерывное преобразование; (по записи значения WORKMODE = 2 предустанавливается первый канал из списка выбранных в CHSEL); 11 – недопустимо
0	ADCEN	0	Включение АЦП: 0 – АЦП выключен (по умолчанию); 1 – АЦП включен (по записи значения ADCEN=1 предустанавливается первый канал из списка выбранных в CHSEL). В течение 20 мкс после включения АЦП полученные отчеты следует считать не валидными

7.22.12.3 **CONFIG1**

Регистр конфигурации 1

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-												PAUSE[7:4]			
RW-0												RW-0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAUSE[3:0]				SETUP[8:0]									REFTRIM[2:0]		
RW-0				RW-0									RW-0		

Таблица 430 – Описание бит регистра CONFIG1

Бит	Имя	Значение	Описание
31…20		-	Зарезервировано
19…12	PAUSE[7:0]	0	Время паузы перед началом процесса преобразования, в тактах ADCCLK: 0 – нет паузы; 1 – пауза равна 1 такт ADCCLK; … 255 – пауза равна 255 тактов ADCCLK

Бит	Имя	Значение	Описание
11…3	SETUP[8:0]	0	Время «зарядки» емкости АЦП в тактах ADCCLK: 0 – нет паузы; 1 – пауза 1 ADCCLK; … 511 – пауза 511 ADCCLK. Подробнее про настройку SETUP см. пункт 7.22.11 «Схема входного каскада АЦП»
2…0	REFTRIM[2:0]	000	Подстройка источника опорного напряжения U_{REF_ADC}: 000 – максимальное значение; … 111 – минимальное значение. Влияет на уровень U_{REF_ADC} и на его температурную зависимость. Настройка также оказывает влияние на шкалу термодатчиков и ток зарядки АЦП совместно с битами SELR, REF_TRIMR. Примечание – Значение данного поля задается для обоих блоков АЦП в регистре CONFIG1 блока ADC0 (0x400A_A008). По адресу регистра CONFIG1 блока ADC1 (0x400A_B008) данное поле зарезервировано

7.22.12.4 **CONFIG2**

Регистр конфигурации 2

Base ADDR=		0x400A_A000 0x400A_B000				Offset=	0x0000_000C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-						SELR	DT_MODE	REF_TRIMR[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		REFSEL[5:0]						LEVLCTRL [1:0]		-	IE_ERFIN	IE_AF	IE_NAE	IE_OF	IE_NE
RW-0								RW-0			RW-0	RW-0	RW-0	RW-0	RW-0

Таблица 431 – Описание бит регистра CONFIG2

Бит	Имя	Значение	Описание
31…26		-	Зарезервировано
25	SELR	0	Выбор резистора источника опорного тока: 0 – используется внутренний подстраиваемый резистор (вывод EXTR в третьем состоянии и толерантен); 1 – используется внешний стабильный резистор на выводе ADC_EXTR на землю. Источник опорного тока используется одновременно для TEMP_0, TEMP_1, ADC0, ADC1, ИОН_0 «2,5В»(U_{REF_ADC}), ИОН_1 «2,5В»(U_{REF_ADC}). При SELR=0 необходимо настраивать REF_TRIMR. В противном случае, например, время зарядки емкости АЦП будет выше номинального. Это можно компенсировать подстройкой времени в поле SETUP[8:0] регистра CONFIG1. Подробнее см. пункт 7.22.11 «Схема входного каскада АЦП». Значение бита также влияет на шкалу термодатчика. Примечание – Установка/сброс данного бита для обоих ADC выполняется в регистре CONFIG2 блока ADC0 (0x400A_A00C). В регистре CONFIG2 блока ADC1 (0x400A_B00C) данный бит зарезервирован
24	DT_MODE	0	Режим работы датчика температуры: 0 – максимальный ток (время установки 5 мкс); 1 – минимальный ток (время установки 30 мкс). Время установки необходимо задать с помощью бит SETUP[8:0] соответствующего АЦП. Значение бита влияет на шкалу термодатчика. Примечание – Установка/сброс данного бита для обоих ADC выполняется в регистре CONFIG2 блока ADC0 (0x400A_A00C). В регистре CONFIG2 блока ADC1 (0x400A_B00C) данный бит зарезервирован

Бит	Имя	Значение	Описание
23…16	REF_TRIMR[7:0]	0x00	Подстройка внутреннего резистора источника опорного тока: 1000_0000 Типовое значение опорного тока (10 мкА); 0000_0000 Минимальный опорный ток (5 мкА); … 1111_1111 Максимальный опорный ток (15 мкА). Функционирует только при SELR = 0. Рекомендуется устанавливать в типовое значение. В противном случае, время зарядки емкости АЦП будет выше номинального. Это можно компенсировать подстройкой времени SETUP[8:0]. Подробнее см. пункт 7.22.11 «Схема входного каскада АЦП». Значение битов влияет на шкалу термодатчика. При SELR = 1 не оказывает влияния, можно использовать значение по сбросу. Примечание – Значение данного поля для обоих ADC задается в регистре CONFIG2 блока ADC0 (0x400A_A00C). В регистре CONFIG2 блока ADC1 (0x400A_B00C) данное поле зарезервировано
15, 14	-	-	Зарезервировано
13…8	REFSEL[5:0]	000000	Биты выбора источников опорного напряжения при REFMODE = 1: REFSEL[0] = 1 – REF+ = U_{REF_ADC} внутреннего источника опорного напряжения; REFSEL[1] = 1 – REF+ = ADC16_REF1+ для ADC0, ADC0_REF0+ для ADC1; REFSEL[2] = 1 – REF+ = ADC0_REF0+ для ADC0, ADC16_REF1+ для ADC1; REFSEL[3] = 1 – REF- = GND внутреннего источника опорного напряжения (совпадает с ADC_GND); REFSEL[4] = 1 – REF- = ADC17_REF1- для ADC0, ADC1_REF0- для ADC1; REFSEL[5] = 1 – REF- = ADC1_REF0- для ADC0, ADC17_REF1- для ADC1. Примечание – Для опорных уровней REF+ и REF- допускается выбрать только один источник опорного напряжения

Бит	Имя	Значение	Описание
7,6	LEVLCNTRL[1:0]	00	Управление заданием контроля границ: 00 – нет контроля границ (по умолчанию); 01 – результаты попадают в FIFO, если результирующее значение более или равно MIN_LEVEL и менее или равно MAX_LEVEL; 10 – результаты попадают в FIFO, если результирующее значение менее или равно MIN_LEVEL, или более или равно MAX_LEVEL; 11 – зарезервировано
5		-	Зарезервировано
4	IE_ERFIN	0	Флаг разрешения запроса прерывания по наличию ошибки в интерфейсе с АЦП: 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено
3	IE_AF	0	Флаг разрешения запроса прерывания по событию ADC_AF (AlmostFull): 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено
2	IE_NAE	0	Флаг разрешения запроса прерывания по событию ADC_NAE (NotAlmostEmpty): 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено
1	IE_OF	0	Флаг разрешения запроса прерывания по событию ADC_OF (OverFlow): 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено
0	IE_NE	0	Флаг разрешения запроса прерывания по событию ADC_NE (NotEmpty): 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено

7.22.12.5 **CONTROL**

Регистр управления

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		CHANNEL[5:0]						-							GO
		RW-0													W-1

Таблица 432 – Описание бит регистра CONTROL

Бит	Имя	Значение	Описание
31…14		-	Зарезервировано
13…8	CHANNEL[5:0]	000000	Номер канала, выбранного для преобразования (используется от 0 до 63)
7…1		-	Зарезервировано
0	GO	0	Бит запуска начала преобразования. Запись 1 начинает процесс преобразования, запись 1 в процессе преобразования игнорируется, запись 0 не имеет эффекта

7.22.12.6 **STATUS**

Регистр статуса

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0014								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-		RTCHVAL						-			ADC_ERFIN	ADC_AF	ADC_NAE	ADC_OF	ADC_NE
		R-0										R-0	R-0	R-0	R-0

Таблица 433 – Описание бит регистра STATUS

Бит	Имя	Значение	Описание
31…14	-		Зарезервировано
13…8	RTCHVAL		Номер самого младшего, не обработанного канала по чтению регистров RESULTCHx, если все каналы обработаны, то сохраняется последнее значение, по умолчанию 0
7…5	-		Зарезервировано
4	ADC_ERFIN		Флаг наличия ошибки в интерфейсе с АЦП: 0 – без ошибки; 1 – Ошибка интерфейса с АЦП – сигнал FIN менее или более 39 тактов ADC_CLK. Флаг сбрасывается записью 1 и при CONFIG0.ADCEN=0

3	ADC_AF		Флаг наличия данных на выходе FIFO (AlmostFull): 0 – в FIFO четыре и более свободных записей; 1 – в FIFO менее четырех свободных записей. Флаг сбрасывается автоматически при считывании из FIFO (регистр RESULT) и уменьшения данных в FIFO менее 13
2	ADC_NAE		Флаг наличия данных на выходе FIFO (NotAlmostEmpty): 0 – в FIFO менее четырех записей; 1 – в FIFO четыре и более записей. Флаг сбрасывается автоматически при считывании из FIFO (регистр RESULT) и уменьшения данных в FIFO менее 4
1	ADC_OF		Флаг переполнения FIFO: 0 – нет переполнения FIFO; 1 – было переполнение FIFO. Флаг сбрасывается записью 1 и при CONFIG0.ADCEN=0
0	ADC_NE		Флаг наличия данных в FIFO: 0 – FIFO пусто; 1 – FIFO содержит данные. Флаг сбрасывается автоматически при считывании из FIFO (регистр RESULT) всех данных
Примечание - Для организации надежного сброса статусных флагов прерываний необходимо учитывать требования, изложенные в подразделе 9.3			

7.22.12.7 **RESULT**

Регистр выхода FIFO результата

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0018								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-										CHANNEL[5:0]					
										R-0					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				RESULT[11:0]											
				R-0											

Таблица 434 – Описание бит регистра RESULT

Бит	Имя	Значение	Описание
31…22	-		Зарезервировано
21…16	CHANNEL[5:0]		Номер канала, по которому получены результаты
15…12	-		Зарезервировано
11…0	RESULT[11:0]		Результат преобразования
Примечание – Размер FIFO составляет 16 записей			

7.22.12.8 **RESULTCHx**

Регистр последнего полученного значения по «X» каналу

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_001C - 0x0000_0118								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-										CHANNEL[5:0]					
										R-x					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				RESULT[11:0]											
				R-0											

Таблица 435 – Описание бит регистров RESULTCHx

Бит	Имя	Значение	Описание
31…22	-		Зарезервировано
21…16	CHANNEL[5:0]		Номер канала: Для RESULTCH0 всегда 0; Для RESULTCH1 всегда 1; … Для RESULTCH63 всегда 63
15…12	-		Зарезервировано
11…0	RESULT[11:0]		Результат преобразования

7.22.12.9 **MAX_LEVEL**

Регистр верхней границы

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_011C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				MAX_LEVEL[11:0]											
				RW-0											

Таблица 436 – Описание бит регистра MAX_LEVEL

Бит	Имя	Значение	Описание
31…12	-		Зарезервировано
11…0	MAX_LEVEL[11:0]		Верхняя граница при контроле границ преобразования

7.22.12.10 MIN_LEVEL

Регистр нижней границы

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0120								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				MIN_LEVEL[11:0]											
				RW-0											

Таблица 437 – Описание бит регистра MIN_LEVEL

Бит	Имя	Значение	Описание
31…12	-		Зарезервировано
11…0	MIN_LEVEL[11:0]		Нижняя граница при контроле границ преобразования

7.22.12.11 CHSEL0

Регистр каналов автоперебора 0

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0124								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHSEL[31:16]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL[15:0]															
RW-0															

Таблица 438 – Описание бит регистра CHSEL0

Бит	Имя	Значение	Описание
31…0	CHSEL[31:0]		Унарный код задания каналов, участвующих в автоматическом переключении каналов: 0 – канал не участвует; 1 – канал участвует. Номер бита соответствует каналу: 0 – 0 каналу; … 31 – 31 каналу

7.22.12.12 **CHSEL1**

Регистр каналов автоперебора 1

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0128								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHSEL[63:48]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL[47:32]															
RW-0															

Таблица 439 – Описание бит регистра CHSEL1

Бит	Имя	Значение	Описание
31…0	CHSEL[63:32]		Унарный код задания каналов, участвующих в автоматическом переключении каналов: 0 – канал не участвует; 1 – канал участвует. Номер бита соответствует каналу: 0 – 32 каналу; … 31 – 63 каналу

7.22.12.13 **DMAREQ**

Регистр управления запросами DMA

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_012C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	DMAREQEN[6:0]							DMAREQ3[5:0]						DMAREQ2[5:4]	
	RW-0							RW-0						RW-0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAREQ2[3:0]				DMAREQ1[5:0]						DMAREQ0[5:0]					
RW-0				RW-0						RW-0					

Таблица 440 – Описание бит регистра DMAREQ

Бит	Имя	Значение	Описание
31	-		Зарезервировано
30…24	DMAREQEN[6:0]		Бит разрешения запросов ADC_DMA_REQ: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен. Бит 0 – ADC_DMA_SREQ0; Бит 1 – ADC_DMA_SREQ1; Бит 2 – ADC_DMA_SREQ2; Бит 3 – ADC_DMA_SREQ3; Бит 4 – ADC_DMA_SREQ4; Бит 5 – ADC_DMA_SREQ5; Бит 6 – ADC_DMA_REQ5. Запрос ADC_DMA_SREQ4 формируется, если он разрешен и есть данные, но нет ни одного, настроенного на данный канал, другого запроса DMA. Запрос ADC_DMA_SREQ5 формируется, если он разрешен и есть хотя бы одно данное в FIFO – условие FIFO_NEMPTY. Запрос ADC_DMA_REQ5 формируется, если он разрешен и есть четыре и более данных в FIFO – условие FIFO_NAEMPTY
23…18	DMASREQ3[5:0]		Номер канала, вызывающего запрос ADC_DMA_SREQ3: 0 – канал 0; 1 – канал 1; … 63 – канал 63
17…12	DMASREQ2[5:0]		Номер канала, вызывающего запрос ADC_DMA_SREQ2: 0 – канал 0; 1 – канал 1; … 63 – канал 63
11…6	DMASREQ1[5:0]		Номер канала, вызывающего запрос ADC_DMA_SREQ1: 0 – канал 0; 1 – канал 1; … 63 – канал 63
5…0	DMASREQ0[5:0]		Номер канала, вызывающего запрос ADC_DMA_SREQ0: 0 – канал 0; 1 – канал 1; … 63 – канал 63

7.22.12.14 STATRCH0

Регистр статуса по приему данных с АЦП

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0130								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STRCH [31:16]															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRCH [15:0]															
R-0															

Таблица 441 – Описание бит регистра STATRCH0

Бит	Имя	Значение	Описание
31…0	STRCH[31:0]		Статус приема данных с АЦП: 0 – данных нет; 1 – есть новые данные в соответствующем регистре RESULTCHx. Сброс происходит при чтении соответствующего регистра RESULTCHx, причем приход новых данных более приоритетен, чем сброс по чтению. Номер бита соответствует каналу: 0 – 0 каналу; … 31 – 31 каналу

7.22.12.15 STATRCH1

Регистр статуса по приему данных с АЦП

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0134								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STRCH[63:48]															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRCH[47:32]															
R-0															

Таблица 442 – Описание бит регистра STATRCH1

Бит	Имя	Значение	Описание
31…0	STRCH[63:32]		Статус приема данных с АЦП: 0 – данных нет; 1 – есть новые данные в соответствующем регистре RESULTCHx. Сброс происходит при чтении соответствующего регистра RESULTCHx, причем приход новых данных более приоритетен, чем сброс по чтению. Номер бита соответствует каналу: 0 – 32 каналу; … 31 – 63 каналу

7.22.12.16 **FIFOEN0**

Регистр разрешения записи результата из АЦП в FIFO

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_0138								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOEN[31:16]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFOEN[15:0]															
RW-0															

Таблица 443 – Описание бит регистра FIFOEN0

Бит	Имя	Значение	Описание
31…0	FIFOEN[31:0]		Разрешение записи в FIFO: 0 – канал не участвует; 1 – канал участвует. Номер бита соответствует каналу: 0 – 0 каналу; … 31 – 31 каналу

7.22.12.17 **FIFOEN1**

Регистр разрешения записи данных из АЦП в FIFO

Base ADDR=		0x400A_A000 0x400A_B000			Offset=		0x0000_013C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOEN[63:48]															
RW-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFOEN[47:32]															
RW-0															

Таблица 444 – Описание бит регистра FIFOEN1

Бит	Имя	Значение	Описание
31…0	FIFOEN[63:32]		Разрешение записи в FIFO: 0 – канал не участвует; 1 – канал участвует. Номер бита соответствует каналу: 0 – 32 каналу; … 31 – 63каналу

7.23 Контроллер ЦАП (DACx)

Микросхема содержит два встроенных ЦАП. Структура ЦАП представлена на рисунке 160.

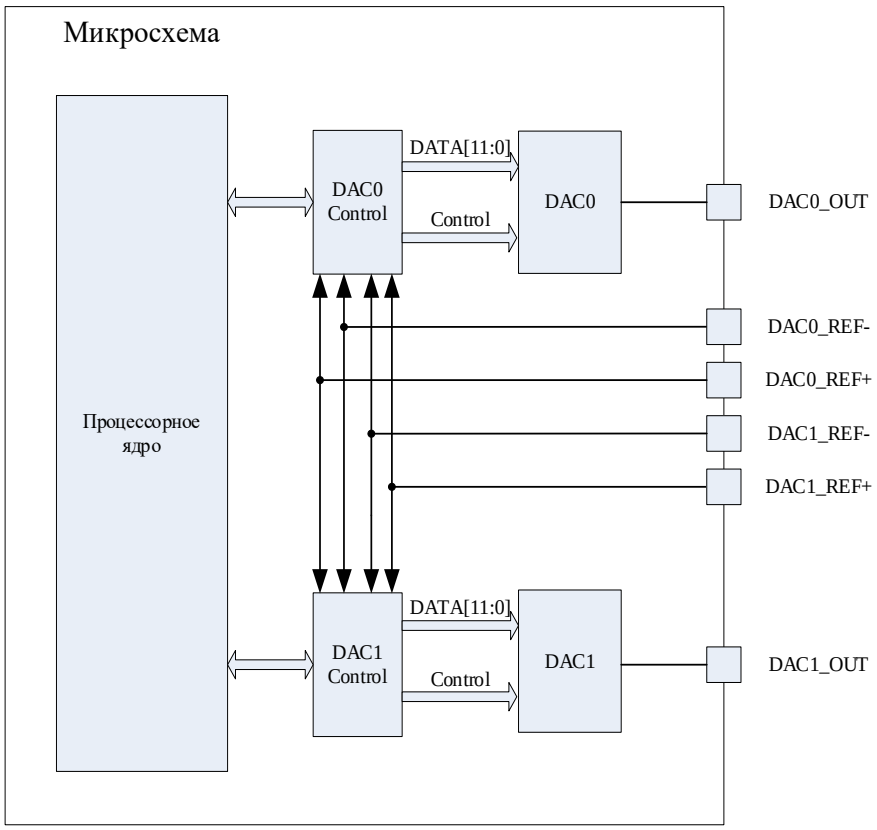


Рисунок 160 – Структурная схема контроллера ЦАП

7.23.1 Источники опорного напряжения

В качестве опорного напряжения ЦАП могут выступать:

– REFMODE = 00 опорное напряжение формируется напряжением питания ЦАП ($AU_{CC}0$ и AGND0 для DAC0 и $AU_{CC}1$и AGND1для DAC1);

– REFMODE = 01 опорное напряжение задается с внешних выходов (DAC0_REF-и DAC0_REF+для DAC0и DAC1_REF-и DAC1_REF+для DAC1);

– REFMODE = 10 опорное напряжение задается с внешних выходов (DAC1_REF-и DAC1_REF+для DAC0и DAC0_REF-и DAC0_REF+для DAC1);

– REFMODE = 11 опорное напряжение формируется встроенным источником опорного напряжения (refDAC0 для DAC0 и refDAC1 для DAC1). Типовое значение 2,5 В можно подстроить с помощью поля REFTRIM регистра CONFIG1.

7.23.2 Синхронизация ЦАП

Обновление значения на выходе DAC происходит на частоте DAC_CLK. При этом могут быть выбраны следующие режимы:

– непрерывное обновление, если в FIFO есть новые данные, то они выставятся на выход на следующий такт DAC_CLK после выставления предыдущих. Если новых данных нет, то на выходе удерживается ранее выставленное значение.

– обновление по событию, на выход DAC данные поступают из FIFO, если они там есть по фронту сигнала EXT_GO. Если есть фронт сигнала EXT_GO, но в FIFO нет новых данных, то на выходе удерживается ранее выставленное значение.

Таблица 445 – Сигналы запуска преобразования ЦАП

Бит EXT_GO_SEL	DAC0	Описание	DAC1	Описание
0	TMR0_EVENT	CNT==ARR в TMR0	TMR0_EVENT	CNT==ARR в TMR0
1	TMR1_EVENT	CNT==ARR в TMR1	TMR1_EVENT	CNT==ARR в TMR1
2	TMR2_EVENT	CNT==ARR в TMR2	TMR2_EVENT	CNT==ARR в TMR2
3	TMR3_EVENT	CNT==ARR в TMR3	TMR3_EVENT	CNT==ARR в TMR3
4	–	Зарезервировано	–	Зарезервировано
5	–	Зарезервировано	–	Зарезервировано
6	TMR0_CH1o	Сигнал ШИМ канала 1 в TMR0	TMR0_CH3o	Сигнал ШИМ канала 3 в TMR0
7	TMR1_CH1o	Сигнал ШИМ канала 1 в TMR1	TMR1_CH3o	Сигнал ШИМ канала 3 в TMR1
8	TMR2_CH1o	Сигнал ШИМ канала 1 в TMR2	TMR2_CH3o	Сигнал ШИМ канала 3 в TMR2
9	TMR3_CH1o	Сигнал ШИМ канала 1 в TMR3	TMR3_CH3o	Сигнал ШИМ канала 3 в TMR3
10	–	Зарезервировано	–	Зарезервировано
11	–	Зарезервировано	–	Зарезервировано
12	PORTA[0]	Сигнал на выводе PA[0]	PORTA[8]	Сигнал на выводе PA[8]
13	PORTA[16]	Сигнал на выводе PA[16]	PORTA[24]	Сигнал на выводе PA[24]
14	PORTB[0]	Сигнал на выводе PB[0]	PORTB[8]	Сигнал на выводе PB[8]
15	PORTB[16]	Сигнал на выводе PB[16]	PORTB[24]	Сигнал на выводе PB[24]
16	PORTC[0]	Сигнал на выводе PC[0]	PORTC[8]	Сигнал на выводе PC[8]
17	PORTC[16]	Сигнал на выводе PC[16]	PORTC[24]	Сигнал на выводе PC[24]
18	PORTD[0]	Сигнал на выводе PD[0]	PORTD[8]	Сигнал на выводе PD[8]
19	PORTD[16]	Сигнал на выводе PD[16]	PORTD[24]	Сигнал на выводе PD[24]
20	PORTE[0]	Сигнал на выводе PE[0]	PORTE[8]	Сигнал на выводе PE[8]
21	PORTE[16]	Сигнал на выводе PE[16]	PORTE[24]	Сигнал на выводе PE[24]
22	ADC0_IF	Окончание преобразования АЦП 0	ADC0_IF	Окончание преобразования АЦП 0

Бит EXT_GO_SEL	DAC0	Описание	DAC1	Описание
23	ADC1_IF	Окончание преобразования АЦП 1	ADC1_IF	Окончание преобразования АЦП 1

7.23.3 Очередь данных

На входе по данным контроллера DAC установлено FIFO размером 16 записей. При переполнении FIFO новые записи игнорируются. Для работы с FIFO формируются сигналы DAC_EMPTY (FIFO пустое), DAC_FULL (FIFO полное), DAC_AEMPTY (в FIFO только четыре или менее записей) и DAC_AFULL (в FIFO только четыре или более свободных записей). Кроме того, контроллер DAC формирует запросы DMA (одиночные и множественные – четыре записи). При переполнении FIFO возникает событие DAC_OF.

7.23.4 Описание регистров

Таблица 446 – Описание регистров контроллера ЦАП

Базовый адрес	Название	Состояние после сброса	Описание
0x400A_C000	DAC0		Контроллер DAC0
0x400A_D000	DAC1		Контроллер DAC1
Смещение			
0x000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
0x004	CONFIG0		Регистр настройки работы контроллера ЦАП
0x008	CONFIG1		Регистр настройки работы контроллера ЦАП
0x00C	DATA		Регистр данных ЦАП, регистр недоступен до записи ключа
0x010	STATUS		Регистр состояние DAC
0x014	IMSK		Регистр маскирования прерываний
0x018	DMAREQ		Регистр управления запросами DMA

7.23.4.1 **KEY**

Base ADDR=		0x400A_C000 0x400A_D000			Offset=		0x000								
REG Name:		KEY													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Таблица 447 – Описание бит регистра KEY

Бит	Имя	Значение	Описание
31…0	KEY[31:0]	0000_0000	При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока DAC контроллера

7.23.4.2 **CONFIG0**

Регистр конфигурации 0

Base ADDR=	0x400A_C000 0x400A_D000	Offset=	0x004						
REG Name:									

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXT_GO_SEL[23:8]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXT_GO_SEL[7:0]								EXT_GO_INV	-		FIFOCLR	REFMODE[1:0]		STMODE	DACEN

Таблица 448 – Описание бит регистра CONFIG0

Бит	Имя	Значение	Описание
31…8	EXT_GO_SEL[23:0]		Биты выбора источника запуска ЦАП по внешнему событию EXT_GO: 0000_0000_0000_0000_0000_0000 запрет всех источников (*по умолчанию*); 0000_0000_0000_0000_0000_0001 вывод EXT_GO[0] (TMR0_EVENT); 0000_0000_0000_0000_0000_0010 сигнал EXT_GO[1] (TMR1_EVENT); 0000_0000_0000_0000_0000_0100 сигнал EXT_GO[2] (TMR2_EVENT). Всего 24 источника (см. таблицу 445)
7	EXT_GO_INV		Управляемая инверсия сигнала EXT_GO: 0 – сигнал не инвертируется (*по умолчанию*); 1 – сигнал инвертируется
6, 5		-	Зарезервировано
4	FIFOCLR		Запись «1» в поле сбрасывает единовременно FIFO, по чтению всегда 0

Бит	Имя	Значение	Описание
3, 2	REFMODE[1:0]		Выбор источника опорного напряжения для преобразования: – REFMODE = 00 опорное напряжение формируется напряжением питания ЦАП ($AU_{CC}0$ и AGND0 для DAC0; $AU_{CC}1$ и AGND1 для DAC1) (*по умолчанию*); – REFMODE = 01 опорное напряжение задается с внешних выходов (DAC0_REF- и DAC0_REF+ для DAC0 и DAC1_REF- и DAC1_REF+ для DAC1); – REFMODE = 10 опорное напряжение задается с внешних выходов (DAC1_REF- и DAC1_REF+ для DAC0 и DAC0_REF- и DAC0_REF+ для DAC1); – REFMODE = 11 опорное напряжение формируется встроенным источником опорного напряжения (refDAC0 для DAC0 и refDAC1 для DAC1)
1	SMODE		Режим работы ЦАП: 0 – обновление по фронту EXT_GO при наличии данных в FIFO (*по умолчанию*); 1 – непрерывное обновление при наличии данных в FIFO по DAC_CLK
0	DACEN		Включение ЦАП: 0 – ЦАП выключен (*по умолчанию*); 1 – ЦАП включен

7.23.4.3 **CONFIG1**

Регистр конфигурации 1

Base ADDR=		0x400A_C000 0x400A_D000			Offset=		0x008								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFTRIMR[7:0]								SELR	-				REFTRIM [2:0]		

Таблица 449 – Описание бит регистра CONFIG1

Бит	Имя	Значение	Описание
31…16	-		Зарезервировано
15…8	REFTRIMR[7:0]		Подстройка внутреннего резистора источника опорного тока: 1000_0000 Типовое значение опорного тока (10 мкА); 0000_0000 Минимальный опорный ток (5 мкА); …

			1111_1111 Максимальный опорный ток (15 мкА). *Для функционирования должно быть задано типовое значение*
7	SELR		Выбор резистора источника опорного тока: 0 – используется внутренний подстраиваемый резистор (вывод EXTR в третьем состоянии и толерантен); 1 – используется внешний стабильный резистор на выводе DAC_EXTR на землю
6…3	-		Зарезервировано
2…0	REFTRIM[2:0]		Подстройка источника опорного напряжения refDAC: 000 – типовое значение (*по умолчанию*); 001… 011 – увеличение опорного напряжения; 100…111 – уменьшение опорного напряжения Значение подстройки определяется в процессе эксплуатации

7.23.4.4 **DATA**

Регистр данных ЦАП

Base ADDR=		0x400A_C000 0x400A_D000			Offset=		0x00C								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-				DATA[11:0]											

Таблица 450 – Описание бит регистра DATA

Бит	Имя	Значение	Описание
31…12			Зарезервировано
11…0	DATA[11:0]		Регистр входных данных FIFO ЦАП При записи, если в FIFO есть свободное место, то записываются данные в FIFO. Если в FIFO нет свободного места, запись игнорируется и взводится флаг переполнения FIFODAC_OF При чтении читается последнее, корректно записанное значение

7.23.4.5 **STATUS**

Регистр состояния ЦАП

Base ADDR=		0x400A_C000 0x400A_D000			Offset=		0x010								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-											DAC_OF	DAC_AFULL	DAC_FULL	DAC_AEMPTY	DAC_EMPTY

Таблица 451 – Описание бит регистра STATUS

Бит	Имя	Значение	Описание
31…5	-		Зарезервировано
4	DAC_OF		Флаг возникновения переполнения FIFO: 0 – нет факта переполнения FIFO; 1 – было переполнения FIFO (запись в DATA при DAC_FULL=1). Очищается записью 1
3	DAC_AFULL		FIFO частично заполнено: 0 – в FIFO четыре и более свободных записей; 1 – в FIFO менее четырех свободных записей
2	DAC_FULL		FIFO полное: 0 – FIFO не заполнено, есть хотя бы одна свободная запись; 1 – FIFO заполнено
1	DAC_AEMPTY		FIFO частично пустое: 0 – в FIFO четыре и более записей; 1 – в FIFO менее четырех записей
0	DAC_EMPTY		FIFO пустое: 0 – FIFO не пусто, есть хотя бы одна запись; 1 – FIFO пустое

7.23.4.6 **IMSK**

Регистр маскирования прерываний

Base ADDR=		0x400A_C000 0x400A_D000			Offset=		0x014								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-											IE_OF	IE_AFULL	IE_FULL	IE_AEMPTY	IE_EMPTY

Таблица 452 – Описание бит регистра IMSK

Бит	Имя	Значение	Описание
31…5			Зарезервировано
4	IE_OF		Флаг разрешения запроса прерывания по событию DAC_OF: 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено
3	IE_AFULL		Флаг разрешения запроса прерывания по событию DAC_AFULL: 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено
2	IE_FULL		Флаг разрешения запроса прерывания по событию DAC_FULL: 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено
1	IE_AEMPTY		Флаг разрешения запроса прерывания по событию DAC_AEMPTY: 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено
0	IE_EMPTY		Флаг разрешения запроса прерывания по событию DAC_EMPTY: 0 – прерывание запрещено (*по умолчанию*); 1 – прерывание разрешено

7.23.4.7 **DMAREQ**

Регистр управления запросами DMA

Base ADDR=		0x400A_C000 0x400A_D000			Offset=		0x018								
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

-														DMAREQEN	DMASREQEN

Таблица 453 – Описание бит регистра DMAREQ

Бит	Имя	Значение	Описание
31…2	-		Зарезервировано
1	DMAREQEN		Бит разрешения запроса DAC_DMA_REQ при DAC_AFULL=0: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен
0	DMASREQEN		Бит разрешения запроса DAC_DMA_SREQ при DAC_FULL=0: 0 – запрос DMA запрещен; 1 – запрос DMA разрешен

7.24 Векторы прерываний и регистры управления ядра

7.24.1 Распределение векторов прерываний

Таблица 454 – Распределение векторов прерываний

Номер исключения	Номер прерывания*	Смещение в таблице векторов	Название	Описание
		0x0000	Initial SP value	
1		0x0004	Reset	
2	-14	0x0008	NMI	Зарезервировано
3	-13	0x000C	Hard fault	
4	-12	0x0010	Memory management fault	
5	-11	0x0014	Bus fault	
6	-10	0x0018	Usage fault	
7	-9	0x001C	Reserved	
8	-8	0x0020	Reserved	
9	-7	0x0024	Reserved	
10	-6	0x0028	Reserved	
11	-5	0x002C	SVCall	
12	-4	0x0030	Reserved for Debug	
13	-3	0x0034	Reserved	
14	-2	0x0038	PendSV	
15	-1	0x003C	Systick	
16	0	0x0040	Reserved	Зарезервировано
17	1	0x0044	Reserved	Зарезервировано
18-46	2-30	0x0048 – 0x00B8	…	Зарезервировано
47	31	0x00BC	Reserved	Зарезервировано
48	32	0x00C0	FT_IF0	Прерывание от событий EVENT0-EVENT4 FTCNTR при разрешенном сбросе по этим событиям
49	33	0x00C4	FT_IF1	Прерывание от событий EVENT0-EVENT4 FTCNTR, независящее от маски сброса
50	34	0x00C8	FT_IF2	Прерывание от событий EVENT5-EVENT8 FTCNTR (маскируемое)

* Нумерация прерываний соответствует стандартной библиотеке CMSIS. Отрицательные номера используются для индексации регистров в SCS для управления системными прерываниями, неотрицательные номера используются для индексации регистров NVIC для управления внешними прерываниями.

Номер исключения	Номер прерывания*	Смещение в таблице векторов	Название	Описание
51	35	0x00CC	FT_IF3	Прерывание от событий EVENT9-EVENT12 FTCNTR (маскируемое)
52	36	0x00D0	CLK_IF	Прерывание контроллера тактовых частот
53	37	0x00D4	PVD_IF	Прерывание от блока детектора напряжений питания
54	38	0x00D8	RTC_IF	Прерывание от блока часов реального времени
55	39	0x00DC	BKP_IF	Прерывание от контроллера батарейного домена
56	40	0x00E0	EXT_INTERROR0	Прерывание контроллера EXT_BUS для региона 0
57	41	0x00E4	EXT_INTERROR1	Прерывание контроллера EXT_BUS для региона 1
58	42	0x00E8	EXT_INTERROR2	Прерывание контроллера EXT_BUS для региона 2
59	43	0x00EC	EXT_INTERROR3	Прерывание контроллера EXT_BUS для региона 3
60	44	0x00F0	EXT_INTERROR4	Прерывание контроллера EXT_BUS для региона 4
61	45	0x00F4	EXT_INTERROR5	Прерывание контроллера EXT_BUS для региона 5
62	46	0x00F8	EXT_INTERROR6	Прерывание контроллера EXT_BUS для региона 6
63	47	0x00FC	EXT_INTERROR7	Прерывание контроллера EXT_BUS для региона 7
64	48	0x0100	-	
65	49	0x0104	-	
66	50	0x0108	DMA_ERR	Прерывание от контроллера DMA по ошибке на шине
67	51	0x010C	DMA_DONE0	Прерывание от контроллера DMA по завершении цикла для канала 0
68	52	0x0110	DMA_DONE1	Прерывание от контроллера DMA по завершении цикла для канала 1
69-97	53-81	0x0114 – 0x0184	…	Прерывание от контроллера DMA по завершении цикла для канала 2-30
98	82	0x0188	DMA_DONE31	Прерывание от контроллера DMA по завершении цикла для канала 31
99	83	0x018C	IRQ_PORTA	Прерывание от контроллера портов ввода/вывода PORTA

Номер исключения	Номер прерывания*	Смещение в таблице векторов	Название	Описание
100	84	0x0190	IRQ_PORTB	Прерывание от контроллера портов ввода/вывода PORTB
101	85	0x0194	IRQ_PORTC	Прерывание от контроллера портов ввода/вывода PORTC
102	86	0x0198	IRQ_PORTD	Прерывание от контроллера портов ввода/вывода PORTD
103	87	0x019C	IRQ_PORTE	Прерывание от контроллера портов ввода/вывода PORTE
104	88	0x01A0	-	
105	89	0x01A4	INT_ETH0	Прерывание от контроллера Ethernet
106	90	0x01A8	-	
107	91	0x01AC	INT_SPW0	Прерывание от контроллера SpaceWire
108	92	0x01B0	-	
109	93	0x01B4	INT_TMR0	Прерывание от контроллера таймера 0
110	94	0x01B8	INT_TMR1	Прерывание от контроллера таймера 1
111	95	0x01BC	INT_TMR2	Прерывание от контроллера таймера 2
112	96	0x01C0	INT_TMR3	Прерывание от контроллера таймера 3
113	97	0x01C4	-	
114	98	0x01C8	-	
115	99	0x01CC	INT_CAN0	Прерывание от контроллера CAN
116	100	0x01D0	-	
117	101	0x01D4	-	
118	102	0x01D8	-	
119	103	0x01DC	-	
120	104	0x01E0	INT_SSP0	Прерывание от контроллера SSP
121	105	0x01E4	-	
122	106	0x01E8	-	
123	107	0x01EC	-	
124	108	0x01F0	INT_UART0	Прерывание от контроллера UART0
125	109	0x01F4	INT_UART1	Прерывание от контроллера UART1
126	110	0x01F8	-	
127	111	0x01FC	-	
128	112	0x0200	-	

Номер исключения	Номер прерывания*	Смещение в таблице векторов	Название	Описание
129	113	0x0204	INT_RX_ARC0	Прерывание от контроллера приемника ARINC
130	114	0x0208	INT_TX_ARC0	Прерывание от контроллера передатчика ARINC
131	115	0x020C	-	
132	116	0x0210	-	
133	117	0x0214	INT_MIL0	Прерывание от контроллера МКПД 0
134	118	0x0218	INT_MIL1	Прерывание от контроллера МКПД 1
135	119	0x021C	INT_ADC0	Прерывание от контроллера АЦП 0
136	120	0x0220	INT_ADC1	Прерывание от контроллера АЦП 1
137	121	0x0224	INT_DAC0	Прерывание от контроллера ЦАП 0
138	122	0x0228	INT_DAC1	Прерывание от контроллера ЦАП 1
139	123	0x022C	-	
140	124	0x0230	-	
141	125	0x0234	-	
142	126	0x0238	-	
143	127	0x023C	-	

7.24.2 Описание регистров управления ядра (SCS)

Подробное описание регистров управления ядра приведено в документе Cortex-M4 Technical Reference Manual.

Таблица 455 – Описание регистров управления ядра (SCS)

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0xE000_0000		ITM		
0xE000_1000		DWT		
	0x000			
0xE000_E000		SC		
	0x004	ICTR		Регистр описания контроллера прерываний
	0x008	ACTLR		Регистр настройки ядра
	0x010	STCSR		Регистр настройки и состояния SysTick

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
	0x014	STRVR		Регистр основания счета SysTick
	0x018	STCVR		Регистр текущего значения SysTick
	0x01C	STCR		Регистр калибровки SysTick
	0x100-0x11C			
	0x180-0x19C			
	0x200-0x21C			
	0x300-0x31C			
	0x400-0x41C			
	0xD00	CPUID		Регистр ID процессорного ядра
	0xD04	ICSR		Регистр настройки и состояния прерываний
	0xD08	VTOR		Регистр смещения таблицы векторов прерывания
	0xD0C	AIRCR		Регистр прерываний приложений и сброса
	0xD10	SCR		Регистр настройки системы
	0xD14	CCR		Регистр настройки и конфигурации
	0xD18	SHPR1		Регистр настройки приоритета 1 обработчиков
	0xD1C	SHPR2		Регистр настройки приоритета 2 обработчиков
	0xD20	SHPR3		Регистр настройки приоритета 3 обработчиков
	0xD24	SHCSR		Регистр настройки и состояния обработчиков

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
	0xD28	CFSR		Регистр конфигурации состояния ошибки
	0xD2C	HFSR		Регистр состояния HardFault
	0xD30	DFSR		Регистр состояния ошибки отладки
	0xD34	MMFAR		Регистр адреса ошибки управления памяти
	0xD38	BFAR		Регистр адреса ошибки BusFault
	0xD3C	AFSR		Регистр состояния ошибки настройки ядра
	0xD40	ID_PFR0		Регистр
	0xD44	ID_PFR1		
	0xD48	ID_DFR0		
	0xD4C	ID_AFR0		
	0xD50	ID_MMFR0		
	0xD54	ID_MMFR1		
	0xD58	ID_MMFR2		
	0xD5C	ID_MMFR3		
	0xD60	ID_IASR0		
	0xD64	ID_IASR1		
	0xD68	ID_IASR2		
	0xD6C	ID_IASR3		
	0xD70	ID_IASR4		
	0xD88	CPACR		
	0xD90	MPU_TYPE		
	0xD94	MPU_CTRL		
	0xD98	MPU_RNR		
	0xD9C	MPU_RBAR		
	0xDA0	MPU_RASR		
	0xDA4	MPU_RBAR_A1		
	0xDA8	MPU_RASR_A1		
	0xDAC	MPU_RBAR_A2		
	0xDB0	MPU_RASR_A2		
	0xDB4	MPU_RBAR_A3		
	0xDB8	MPU_RASR_A3		
	0xDF0	DHCSR		
	0xDF4	DCRSR		

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
	0xDF8	DCRDR		
	0xDFC	DEMCR		
	0xF00	STIR		
	0xF34	FPCCR		
	0xF38	FPCAR		
	0xF3C	FPDSCR		
	0xF40	MVFR0		
	0xF44	MVFR1		
	0xFD0	Peripheral ID4		
	0xFE0	Peripheral ID0		
	0xFE4	Peripheral ID1		
	0xFE8	Peripheral ID2		
	0xFEC	Peripheral ID3		
	0xFF0	Component ID0		
	0xFF4	Component ID1		
	0xFF8	Component ID2		
	0xFFC	Component ID3		
0xE004_0000		TPUI		
0xE00F_F000				
	0x000	SCS		
	0x004	DWT		
	0x008	FPB		
	0x00C	ITM		
	0x010	TPUI		
	0x014	ETM		
	0x018	End Marker		
	0xFFC	SYSTEM ACCESS		

7.25 Контроллер блока защиты памяти (MPU)

Подробное описание блока защиты памяти приведено в документе Cortex-M4 Technical Reference Manual.

7.26 Контроллер блока вычисления с плавающей запятой (FPU)

Подробное описание блока вычислений с плавающей запятой приведено в документе Cortex-M4 Technical Reference Manual.

7.27 Контроллер порта трассировки (TPIU)

Подробное описание регистров управления ядра приведено в документе Cortex-M4 Technical Reference Manual.

7.28 Контроллер блока расширенной трассировки (ETM)

Подробное описание блока расширенной трассировки приведено в CoreSight ETM-M4 Technical Reference Manual.

8 Системный таймер SysTick

Процессор имеет 24-разрядный системный таймер, SysTick, который считает вниз от загруженного в него значения до нуля; перезагрузка (возврат в начало) значения в регистр LOAD происходит по следующему фронту синхросигнала, затем счет продолжается по последующему фронту.

Когда процессор остановлен для отладки, таймер не декрементируется.

8.1 Описание регистров системного таймера SysTick

Таблица 456 – Описание регистров системного таймера SysTick

Адрес	Название	Тип	Доступ	Значение после сброса	Описание
0xE000E010	SysTick				Системный таймер SYSTICK
0xE000E010	CTRL	RW	привилегированный	0x00000004	SysTick->CTRL
0xE000E014	LOAD	RW	привилегированный	0x00000000	SysTick->LOAD
0xE000E018	VAL	RW	привилегированный	0x00000000	SysTick->VAL
0xE000E01C	CALIB	RO	привилегированный	0x00002904*	SysTick->CAL
* Калибровочное значение системного таймера					

8.1.1 SysTick->CTRL

Регистр CTRL разрешает основные функции системного таймера.

Назначение бит:

Таблица 457 – Регистр контроля и статуса CTRL

Номер	31…17	16	15…3	2	1	0
Доступ						
Сброс						
	-	COUNTFLAG	-	CLKSOURCE	TICKINT	ENABLE

8.1.1.1 COUNTFLAG

Возвращает 1, если таймер досчитал до нуля с последнего момента чтения.

8.1.1.2 CLKSOURCE

Указывает источник синхросигнала:

0 – LSI;

1 – HCLK

8.1.1.3 TCKINT

Разрешает запрос на прерывание от системного таймера:

0 – таймер досчитает до нуля и прерывание не возникнет;

1 – таймер досчитывает до нуля и возникает запрос на прерывание.

Программное обеспечение может использовать бит COUNTFLAG, чтобы определить, досчитал таймер до нуля или нет.

8.1.1.4 **ENABLE**

Разрешает работу таймера:

0 – работа таймера запрещена;

1 – работа таймера разрешена.

Когда ENABLE установлен в единицу, таймер загружает значение RELOAD из регистра LOAD и затем начинает декрементироваться. По достижению значения 0 таймер устанавливает бит COUNTFLAG и в зависимости от TCKINT генерирует запрос на прерывание. Затем загружается значение RELOAD и продолжается счет.

8.1.2 SysTick->LOAD

Регистр LOAD устанавливает стартовое значение, загружаемое в регистр VAL.

Таблица 458 – Регистр перегружаемого значения LOAD

Номер	31…24	23…0
Доступ		
Сброс		
	-	RELOAD

8.1.2.1 **RELOAD**

Значение, загружаемое в регистр VAL, когда таймер разрешен и когда достигается значение нуля.

Расчет значения RELOAD

Значение RELOAD может быть любым в диапазоне 0x00000001-0x00FFFFFF. Значение 0 допустимо, но не оказывает эффекта, потому что запрос на прерывание и активизация бита COUNTFLAG происходит только при переходе таймера из состояния 1 в 0.

Расчет значения RELOAD происходит в соответствии с использованием таймера:
Для формирования мультикороткого таймера с периодом N процессорных циклов 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.

8.1.3 SysTick->VAL

Регистр VAL содержит текущее значение системного таймера.

Таблица 459 – Регистр текущего значения таймера VAL

Номер	31…24	23…0
Доступ		
Сброс		
	-	CURRENT

8.1.3.1 CURRENT

Чтение возвращает текущее значение системного таймера.

Запись любого значения очищает регистр в ноль, и также очищает бит COUNTFLAG регистра CTRL.

8.1.4 SysTick->CAL

Регистр CALIB показывает калибровочное значение системного таймера.

Таблица 460 – Регистр калибровочного значения таймера CAL

Номер	31	30	29…24	23…0
Доступ				
Сброс				
	NOREF	SKEW	-	TENMS

8.1.4.1 NOREF

Читается как ноль.

8.1.4.2 SKEW

Читается как ноль.

8.1.4.3 TENMS

Читается как 0x0002904.

Калибровочное значение фиксировано и равно 0x0002904 (10500), что позволяет генерировать базовое время 1 мс с частотой 10,5 МГц (84/8=10,5 МГц).

8.2 Советы и особенности при применении системного таймера

Системный таймер работает от процессорного синхросигнала. Если синхросигнал останавливается в режиме пониженного энергопотребления, то системный таймер останавливается.

Гарантируйте, чтобы программа использовала доступ к регистрам системного таймера, выровненный по словам.

9 Прерывания и исключения

Состояние исключений:

Inactive – исключение не находится в стадии Active или Pending

Pending – исключение находится в состоянии ожидания обработки процессором. Запрос прерывания от периферийных блоков или программы может изменить состояние соответствующего прерывания на состояние Pending

Active – исключение начало обрабатываться процессором, но еще не закончено. Обработчик исключения может быть прерван другим обработчиком исключения. В этом случае оба исключения находятся в состоянии Active

Active и Pending – исключение начало обрабатываться процессором, но появилось новое исключение в состоянии pending от того же источника.

9.1 Типы исключений

Исключения бывают следующих типов:

9.1.1 RESET

RESET вызывается при включении питания и горячем сбросе. Модель исключений трактует RESET как специальную форму исключения. Когда выставляется RESET, работа процессора останавливается потенциально в любой точке инструкций. Когда RESET убирается, выполнение перезапускается с адреса, заданного в таблице векторов для сброса. Выполнение перезапускается с уровнем privileged в thread режиме.

9.1.2 NON MASKABLE INTERRUPT (NMI)

Немаскируемое прерывание (NMI) может быть вызвано периферией или установлено программой. Это самое высокоприоритетное исключение после сброса. Всегда разрешено и имеет фиксированный приоритет - 2.

Примечание – Данное прерывание не реализовано.

NMI не может быть:

- замаскировано или предотвращено от активации из другого исключения;
- прерывает любые исключения, кроме RESET.

9.1.3 Hard Fault

Исключение Hard Fault возникает при ошибке при обработке исключений или потому, что исключение не может быть обработано каким-либо другим механизмом. Hard fault имеет фиксированный приоритет -1, означающий, что оно имеет больший приоритет, чем любое из исключений с конфигурируемым приоритетом.

9.1.4 Memory Management fault

Исключение Memory Management fault возникает при срабатывании по защите памяти. Блок MPU или фиксированные защитные настройки определяют это исключение, как для данных, так и для инструкций. Исключение используется для

прерывания доступа за инструкцией в область EXECUTE NEVER (XN), если блок MPU не используется.

9.1.5 Bus Fault

Исключение возникает при ошибке памяти при выполнении выборки инструкций или обращения за данными. Это может быть при возникновении ошибки на шинах доступа к памяти, например, обращение в несуществующую память.

9.1.6 Usage Fault

Исключение USAGE FAULT возникает при сбоях при выполнении инструкции.

Например:

- выполнение неизвестной инструкции;
- обращение к некорректно выровненным данным;
- некорректное состояние при выполнении инструкции;
- ошибка при возвращении из обработчика.

Данное исключение может быть сконфигурировано и используется для обработки следующих ситуаций:
невыровненный адрес при обращении за полусловами halfword и словами word;
деление на ноль.

9.1.7 SVCall

Исключение Supervisor Call (SVCALL) возникает при выполнении инструкции SVC. В приложениях с использованием Операционных Сред инструкция SVC может использоваться для доступа к функциям ОС и драйверам устройств.

9.1.8 PendSV

Исключение PendSV является прерыванием запросом сервисов системного уровня. В приложениях с использованием ОС PendSV используется для переключения контекстов, когда нет других активных исключений.

9.1.9 SysTick

Исключение SysTick генерируется системным таймером, когда он обнуляется. Программное обеспечение также может генерировать исключение SysTick. В приложениях с использованием ОС процессор может использовать это исключение для подсчета системных циклов

9.2 Прерывания (IRQ)

Прерывания или IRQ – это исключения, вызываемые периферийными устройствами или программными запросами. Все прерывания асинхронны по отношению к выполняемым инструкциям. В системе прерывания используются для коммуникации периферии и процессора

Таблица 461 – Различные типы исключений

Номер исключения	Номер IRQ	Тип	Приоритет	Адрес вектора обработчика (смещение)	Активация
1	-	RESET	-3, наивысший	0x0000_0004	Асинхронный
2	-14	NMI	-2	0x0000_0008	Асинхронный
3	-13	Hard Fault	-1	0x0000_000C	-
4	-12	Memory Management Fault	Конфигурируемый	0x0000_0010	Синхронный
5	-11	Bus Fault	Конфигурируемый	0x0000_0014	Синхронный/ Асинхронный
6	-10	Usage Fault	Конфигурируемый	0x0000_0018	Синхронный
7-10	-	-	-	Зарезервировано	-
11	-5	SVCall	Конфигурируемый	0x0000_002C	Синхронный
12-13	-	-	-	Зарезервировано	-
14	-2	PendSV	Конфигурируемый	0x0000_0038	Асинхронный
15	-1	SysTick	Конфигурируемый	0x0000_003C	Асинхронный
16 и выше	0 и выше	IRQ	Конфигурируемый	0x0000_0040 и выше	Асинхронный

Для асинхронных исключений, кроме RESET, процессор может выполнить другие инструкции между возникновением сигнала исключения и входом в обработчик.

Программа в Privileged режиме может запретить прерывания, имеющие конфигурируемый приоритет.

9.3 Обработчики исключений

Для обработки исключений используются:

– Процедуры обработки прерываний (Interrupt Service Routines – ISRs):

Прерывания с IRQ0 по IRQ127 обрабатываются процедурами ISR.

Если программный код, сбрасывающий флаг прерывания на уровне периферийного блока, находится в конце функции обработчика прерывания, ядро может не успеть выполнить очистку флага в момент выхода из обработчика, из-за чего прерывание будет вызвано повторно. Чтобы этого избежать, необходимо после сброса флага прерывания сделать фиктивное чтение, например, регистра статусных флагов периферийного блока в отдельную переменную.

– Обработчики ошибок (Fault Handlers):

Обрабатывают исключения Hard fault, memory management fault, usage fault и bus fault.

– Системные обработчики (System handlers):

Обрабатывают исключения NMI, PendSV, SVCall и SysTick.

9.4 Таблица векторов

Таблица векторов содержит указатель стека, вектор входа по RESET и стартовые адреса обработчиков, также называемых векторами. Рисунок 161 представляет последовательность векторов в таблице. Младший бит всех векторов должен быть равен 1, указывая на то, что обработчик выполняется в Thumb режиме.

Exception number	IRQ number	Offset	Vector
47	31	0×00BC	IRQ31
.	.	.	
.	.	.	
.	.	.	
		0×004C	
18	2	0×0048	IRQ2
17	1	0×0044	IRQ1
16	0	0×0040	IRQ0
15	-1	0×003C	Systick
14	-2	0×0038	PendSV
13			Reserved
12			Reserved for Debug
11	-5	0×002C	SVCall
10			Reserved
9			
8			
7			
6	-10	0×0018	Usage fault
5	-11	0×0014	Bus fault
4	-12	0×0010	Memory management fault
3	-13	0×000C	Hard fault
2	-14	0×0008	Reserved
1		0×0004	Reset
		0×0000	Initial SP value

Рисунок 161 – Таблица векторов исключений и прерываний

При системном сбросе таблица векторов располагается по фиксированному адресу 0x00000000. Программное обеспечение в privileged режиме может перенести таблицу в другое место памяти через регистр VTOR. Таблица может располагаться в адресах от 0x00000080 до 0x3fffff80. Подробнее в описании регистра VTOR.

9.5 Приоритеты исключений

– Более малое значение приоритета означает больший приоритет.

– Конфигурируемы все приоритеты, кроме RESET и Hard Fault.

– Если программное обеспечение не задает приоритетов, то все они имеют приоритет 0.

– Конфигурируемый приоритет может быть в диапазоне от 0 до 7. Это означает что RESET, Hard Fault и NMI, имеющие отрицательное значение приоритета, всегда имеют больший приоритет.

– Если имеется несколько исключений с одинаковым приоритетом, то больший приоритет имеет исключение с меньшим порядковым номером.

– Если процессор выполняет обработчик исключения и происходит исключение с большим приоритетом, то происходит переход на обработчик исключения с большим приоритетом.

– Если при выполнении обработчика произошло исключение с таким же приоритетом, то это исключение будет выполнено по завершении текущего обработчика, несмотря на порядковый номер исключения.

9.5.1 Группировка приоритетов прерываний

Для увеличения управляемости приоритетов в системах с прерываниями контроллер прерываний NVIC поддерживает группировку приоритетов. Это достигается за счет разбиения регистра приоритета прерывания на две части:
верхняя часть определяет группу приоритетов;
нижняя часть задает подприоритет в группе.

Только приоритет группы определяет последовательность обработки прерываний. Когда процессор выполняет обработку прерывания, другое прерывание с таким же приоритетом группы не прервет обработку первоначального обработчика. При возникновении нескольких прерываний, имеющих одинаковый приоритет группы, подприоритеты определяют последовательность их обработки. При возникновении нескольких прерываний с одинаковым приоритетом группы и подприоритетом первым обрабатывается прерывание с меньшим номером.

9.6 Вход в обработчик и выход из обработчика

При описании используются следующие термины:

9.6.1 Приоритетное прерывание

Выполнение процессором процедуры обработки исключительной ситуации (далее по тексту – исключения), может быть прервано в случае возникновения исключения с приоритетом выше, чем у обрабатываемого. Подробнее данный вопрос рассмотрен в пункте 9.5.1 «Группировка приоритетов прерываний». В случае если внутри обработчика исключения возникает прерывание более высокого приоритета, возникает ситуация, называемая вложенным исключением. Подробнее данный вопрос рассмотрен в пункте 9.6.5 «Вход в процедуру обработки исключения».

9.6.2 Возврат

Возврат из программы-обработчика осуществляется по завершении обработки исключительной ситуации, с одновременным выполнением следующих условий:
в системе отсутствуют необработанные исключения с достаточным приоритетом;
завершенный обработчик не обрабатывал запоздавшее исключение (late-arriving exception).

Процессор обращается к стеку и восстанавливает состояние, имевшее место до вызова обработчика. Более подробная информация дана в пункте 9.6.6 «Возврат из обработчика исключения»

9.6.3 Передача управления без восстановления контекста (tail-chaining)

Данный механизм ускоряет процесс обработки исключений. По завершении выполнения обработчика осуществляется проверка наличия необработанных исключений и в случае, если исключения, требующие вызова обработчика, присутствуют, восстановление состояния процессора из стека не производится, а управление передается непосредственно на новый обработчик.

9.6.4 Запоздавшее исключение (late-arriving exception)

В случае, если во время сохранения состояния при входе в обработчик возникла исключительная ситуация с более высоким приоритетом, процессор передает управление непосредственно высокоприоритетному обработчику.

Подобный способ обработки высокоприоритетного исключения возможен до момента начала выполнения первой инструкции процедуры обработки исключительной ситуации. После возврата из обработчика запоздавшего исключения осуществляется передача управления на прерванный низкоприоритетный обработчик без восстановления контекста.

9.6.5 Вход в процедуру обработки исключения

Вызов процедуры обработки исключения происходит в случае наличия необработанных исключительных ситуаций с достаточным приоритетом и при выполнении одного из следующих условий:

– процессор находится в режиме приложения (thread mode);
– новая исключительная ситуация имеет приоритет выше, чем обрабатываемая в текущий момент времени, что приводит к приоритетному прерыванию выполнения текущего обработчика. В этом случае возникает вложение одного исключения в другое.

При необходимости вызова обработчика, за исключением случаев обработки запоздавшего исключения и передачи управления на обработчик без восстановления контекста, процессор заносит в текущий стек восемь слов данных, называемые далее стековым фреймом. Этот фрейм включает в себя следующие значения:

– регистры R0-R3, R12;
– адрес возврата;
– регистр PSR;

– регистр LR.

Указанная операция далее будет называться сохранением контекста. Непосредственно после ее выполнения указатель стека равен младшему адресу стекового фрейма.

В случае если бит STKALIGN в регистре управления конфигурацией (CCR) установлен в 1, во время сохранения контекста производится выравнивание адреса стека по границе двойного слова.

Стековый фрейм содержит адрес возврата, указывающий на ближайшую невыполненную инструкцию прерванной программы. По завершении процедуры обработки исключения значение адреса возврата заносится в счетчик команд, после чего выполнение программы возобновляется с прерванной точки.

Одновременно с сохранением контекста процессор осуществляет выборку адреса точки входа в процедуру обработки исключения из таблицы векторов исключений. По завершении операции сохранения контекста процессор передает управление на полученный из таблицы адрес.

Одновременно в регистр LR записывается значение EXC_RETURN, позволяющее определить, какой из двух указателей стека соответствует данному стековому фрейму, и в каком режиме находился процессор перед входом в обработчик.

Если во время передачи управления не возникло исключения с более высоким приоритетом, процессор начинает выполнение вызванной процедуры обработки и автоматически изменяет состояние текущего прерывания с ожидающего обработки на активное.

В противном случае процессор передает управление обработчику высокоприоритетной исключительной ситуации без изменения состояния отложенного прерывания в соответствии с правилами, изложенными в пункте 9.6.4 «Запоздавшее исключение (late-arriving exception)».

9.6.6 Возврат из обработчика исключения

Возврат из обработчика исключения осуществляется в случае, если процессор находится в режиме обработчика (handler mode) и выполняет одну из следующих инструкций, позволяющих загрузить значение EXC_RETURN в регистр PC:

– инструкцию POP с аргументом PC;
– инструкцию BX с любым регистром;
– инструкции LDR или LDM с регистром PC в качества приемника.

Значение EXC_RETURN загружается в регистр LR по входу в обработчик исключения. Механизм обработки исключений использует это значение для того, чтобы определить, завершил ли процессор выполнение процедуры обработки исключительной ситуации. Младшие четыре бита EXC_RETURN содержат информацию о состоянии стека и режиме работы процессора. Информация о назначении разрядов EXC_RETURN[3:0] и особенности процесса возврата из обработчика исключения представлены в таблице 462.

Процессор устанавливает биты EXC_RETURN [31:4] в 0xFFFFFFF. Загрузка данного значения в PC указывает на завершение процедуры обработки исключения и заставляет процессор выполнить необходимые действия для возврата из обработчика.

Таблица 462 – Возврат из обработчика исключения

EXC_RETURN[3:0]	Описание
bXXX0	Зарезервирован
b0001	Возврат в режим обработчика. Восстановление контекста осуществляется из стека MSP. Дальнейшая работа осуществляется со стеком MSP
b0011	Зарезервирован
b01X1	Зарезервирован
b1001	Возврат в режим приложения. Восстановление контекста осуществляется из стека MSP. Дальнейшая работа осуществляется со стеком MSP
b1101	Возврат в режим приложения. Восстановление контекста осуществляется из стека PSP. Дальнейшая работа осуществляется со стеком PSP
b1X11	Зарезервирован

9.7 Обработка отказов

Отказы являются частным случаем исключений. Отказы могут возникать по следующим причинам:

– ошибка шины в ходе:

– чтения инструкции или вектора обработчика;

– доступа к данным.

– ошибка, обнаруженная процессором, например, неопределенная инструкция или попытка изменить состояние процессора с помощью команды BX;

– попытка выполнить инструкцию, расположенную в области памяти, помеченной как неисполняемая (Non-Executable – XN);

– отказ блока защиты памяти MPU вследствие нарушения прав доступа или вследствие попытки доступа к неподдерживаемой области адресного пространства.

9.7.1 Типы отказов

В таблице 463 представлены типы отказов, обработчики, вызываемые при их возникновении, соответствующие данному типу отказа регистры состояния, и биты регистра, указывающие на конкретный отказ. Более подробная информация представлена в разделе “Конфигурируемый регистр отказов”.

Таблица 463 – Отказы

Отказ	Обработчик	Наименование бита регистра	Регистр отказа
Ошибка доступа к шине при чтении вектора	Тяжелый отказ	VECTTBL	«Регистр состояния тяжелых отказов»
Эскалация отказа		FORCED	
Ошибка доступа к памяти:	Отказ доступа к памяти	-	«Регистр состояния отказов доступа к памяти», «Регистр адреса отказа доступа к памяти»
- при чтении команды		IACCVIOL	
- при доступе к данным		DACCVIOL	
- при сохранении контекста		MSTKERR	
- при восстановлении контекста		MUNSKERR	
Ошибка шины:	Отказ доступа к шине	-	«Регистр состояния отказа доступа к шине», «Регистр адреса отказа доступа к шине»
- при сохранении контекста		STKERR	
- при восстановлении контекста		UNSTKERR	
- при загрузке инструкции		IBUSERR	
локализованная ошибка шины данных		PRECISERR	
нелокализованная ошибка шины данных		IMPRECISERR	
Попытка доступа к сопроцессору	Отказ, вызванный ошибками программирования	NOCP	«Регистр состояния отказов, вызванных ошибками программирования»
Неизвестная инструкция		UNDEFINSTR	
Попытка выбора неверного набора инструкций*		INVSTATE	
Неверное значение EXC_RETURN		INVPC	
Запись или чтение по неверно выровненному адресу		UNALIGNED	
Деление на 0		DIVBYZERO	
* Попытка выбора набора инструкций, не поддерживаемого процессором			

9.7.2 Эскалация отказов и тяжелые отказы

Всем типам исключительных ситуаций по отказу, за исключением тяжелых отказов (hard fault) можно задать приоритет обработки, см. пункт 11.1.8 «SCB->SHP[x]». Выполнение данных обработчиков можно программно запретить, см. пункт 11.1.9 «SCB->SHCSR».

Как правило, приоритет обработки исключения, наряду со значениями регистров маскирования исключений, определяет, будет ли вызываться данный обработчик отказа, а также – сможет ли он прервать выполнение другого обработчика.

В некоторых ситуациях отказ с конфигурируемым уровнем приоритета рассматривается системой как тяжелый. Такая ситуация именуется эскалацией отказа (escalation). Это возможно в следующих случаях:

– обработчик отказа во время своего выполнения вызвал отказ того же типа. Этот тип эскалации обусловлен тем фактом, что обработчик не может прервать собственное выполнение, так как его приоритет равен текущему;

– обработчик отказа вызвал отказ другого типа с приоритетом, меньшим или равным собственному. В этом случае новый обработчик также не может быть активизирован вследствие недостаточного уровня приоритета;

– обработчик исключительной ситуации вызвал отказ с приоритетом обработки, меньшим или равным текущему;

– возник отказ, обработчик которого не разрешен.

Если отказ обращения к шине возник во время загрузки данных в стек при передаче управления на обработчик отказа доступа к шине - эскалации не происходит. Таким образом, в случае, если отказ возник вследствие разрушения стека, передача управления на обработчик отказа выполняется, несмотря на то, что сохранение контекста не было осуществлено.

Обработка тяжелых отказов имеет фиксированный приоритет. Она может быть прервана только по сигналу сброса Reset или немаскируемого прерывания NMI. Сам обработчик способен прерывать обработку любых исключительных ситуаций, кроме ситуаций сброса Reset, NMI, а также другого тяжелого отказа.

9.7.3 Регистры состояния и адреса отказа

Регистры состояния отказа содержат информацию о причине отказа. Для обработки отказов шины и доступа к памяти предусмотрены регистры адреса отказа, содержащие адрес, по которому произошло обращение, вызвавшее отказ. Подробная информация приведена в таблице 464.

Таблица 464 – Регистры состояния и адреса отказа

Обработчик	Регистр состояния	Регистр адреса	Описание регистров
Тяжелый отказ	HFSR	-	«Регистр состояния тяжелых отказов»
Отказ доступа к памяти	MMFSR	MMFAR	«Регистр состояния отказов доступа к памяти»

			«Регистр адреса отказа доступа к памяти»
Отказ доступа к шине	BFSR	BFAR	«Регистр состояния отказов доступа к шине» «Регистр адреса отказа доступа к шине»
Отказ, вызванный ошибками программирования	UFSR	-	«Регистр состояния отказов, вызванных ошибками программирования»

9.7.4 Блокировка

Процессор переходит в состояние блокировки в случае, если тяжелый отказ возник во время выполнения программы-обработчика тяжелого отказа.

После перехода в состояние блокировки процессор перестает выполнять какие-либо команды. В этом состоянии он будет находиться до момента сброса.

9.8 Управление электропитанием

В процессоре Cortex-M4 предусмотрены следующие режимы ожидания (пониженного энергопотребления):

- Deep Sleep;
- Sleep;
- Standby.

Выбор процессором конкретного режима ожидания определяется значением бита SLEEPDEEP регистра SCR (см. пункт 11.1.6 «Регистр управления системой»).

Далее в разделе описаны механизмы перехода в режим пониженного энергопотребления и условия выхода из этого режима.

9.8.1 Переход в режим пониженного энергопотребления

Система может формировать ложные сигналы событий, выводящие процессор из ожидания. Например, эти сигналы возникают при работе отладчика. Следовательно, программное обеспечение должно быть способным перевести процессор обратно в указанный режим ожидания. Для этого можно, например, организовать в программе пустой цикл.

9.8.2 Ожидание прерывания

Инструкция ожидания прерывания WFI (wait for interrupt) после своего выполнения немедленно переводит процессор в режим пониженного энергопотребления.

9.8.3 Ожидание события

Инструкция ожидания сигнала события WFE (wait for event) переводит или не переводит процессор в режим пониженного энергопотребления в зависимости от результата проверки одноразрядного регистра события. При этом процессор проверяет значение регистра события, и в случае, если он равен 0, приостанавливает дальнейшее

выполнение команд и переходит в состояние ожидания. В случае если он равен 1, процессор записывает в регистр события 0 и продолжает нормальную работу без перехода в режим ожидания.

9.8.4 Переход в режим ожидания по выходу из обработчика исключения (режим Sleep)

В случае если бит SLEEPONEXIT регистра SCR установлен в 1, по завершении выполнения обработчика исключения процессор возвращается в режим приложения, после чего немедленно переходит в состояние пониженного энергопотребления.

Данный механизм рекомендуется использовать в задачах, в которых процессор используется только для обработки исключений.

9.8.5 Выход из состояния ожидания

Условия выхода процессора из режима ожидания зависят от причины, по которой он был переведен в этот режим.

9.8.5.1 Выход из ожидания по команде WFI и в режиме Sleep

Как правило, процессор выходит из режима ожидания только в случае возникновения исключительной ситуации с приоритетом, достаточным для активизации соответствующего обработчика.

В некоторых приложениях может возникнуть необходимость выполнения процедур восстановления системы после выхода процессора из режима пониженного энергопотребления, однако до того, как он начнет выполнять обслуживание прерываний. Для того чтобы добиться этого, достаточно установить бит PRIMASK в 1, а бит FAULTMASK – в 0. В случае возникновения в системе разрешенного прерывания с приоритетом выше текущего приоритета, процессор будет выведен из ожидания, однако не сможет передать управление обработчику прерывания до тех пор, пока бит PRIMASK не будет установлен в 0.

9.8.5.2 Выход из ожидания по команде WFE

Процессор выходит из режима ожидания в случае обнаружения исключительной ситуации с приоритетом, достаточным для активизации обработчика.

Кроме того, в случае установки бита SEVONPEND регистра SCR в 1, любое новое необслуженное прерывание формирует сигнал события и выводит процессор из ожидания, даже если это прерывание запрещено или имеет приоритет, недостаточно высокий для запуска обработчика.

Более подробная информация о регистре SCR представлена в пункте 11.1.6 «Регистр управления системой».

9.8.6 Рекомендации по программированию режима энергопотребления

В стандарте ANSI языка C отсутствует возможность непосредственной генерации инструкций WFI и WFE. В CMSIS предусмотрены встроенные функции, предназначенные для включения в код этих инструкций:

void __WFE(void) // Wait for Event
void __WFI(void) // Wait for Interrupt

10 Контроллер прерываний NVIC

В разделе описан векторный контроллер прерываний с возможностью вложения (NVIC – Nested Vectored Interrupt Controller) и используемые им регистры.

Контроллер обеспечивает следующие возможности:

– программное задание уровня приоритета в диапазоне от 0 до 7 независимо каждому прерыванию. Более высокое значение уровня соответствует меньшему приоритету, таким образом, уровень 0 отвечает наивысшему приоритету прерывания;

– срабатывание сигнала прерывания по импульсу и по уровню;

– динамическое изменение приоритета прерываний;

– разделение исключений по группам с одинаковым приоритетом и по подгруппам внутри одной группы;

– передача управления из одного обработчика исключения в другой без восстановления контекста.

Процессор автоматически сохраняет в стеке свое состояние (контекст) по входу в обработчик прерывания и восстанавливает его по завершении обработчика, без необходимости непосредственного программирования этих операций. Это обеспечивает обработку исключительных ситуаций с малой задержкой.

Назначение регистров контроллера прерываний представлено в таблице 465.

Таблица 465 – Обобщенная информация о регистрах контроллера NVIC

Адрес	Название	Тип	Доступ	Значение после сброса	Описание
0xE000E100	NVIC				Контроллер прерываний NVIC
0x000	ISER[0]	RW	Привилегированный	0x00000000	Регистр разрешения прерываний ISER
…					
0x01C	ISER[7]				
…					
0x080	ICER[0]	RW	Привилегированный	0x00000000	Регистр запрета прерываний ICER
…					
0x09C	ICER[7]				
…					
0x100	ISPR[0]	RW	Привилегированный	0x00000000	Регистр установки состояния ожидания для прерывания ISPR
…					
0x11C	ISPR[7]				
…					
0x180	ICPR[0]	RW	Привилегированный	0x00000000	Регистр сброса состояния ожидания для прерывания ICPR
…					
0x19C	ICPR[7]				
…					
0x200	IABR[0]	RO	Привилегированный	0x00000000	

Адрес	Название	Тип	Доступ	Значение после сброса	Описание
…					Регистр активных прерываний IABR
0x21C	IABR[7]				
…					
0x300	IP[3],IP[2],IP[1],IP[0]	RW	Привилегированный	0x00000000	Регистр приоритета прерываний IP
…					
0x3F0	IP[239], IP[238], IP[237], IP[236]				
…					
0xE00	STIR	WO	В зависимости от конфигурации*	0x00000000	Регистр программного формирования прерываний STIR
* Более подробную информацию см. в описании регистра					

10.1 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

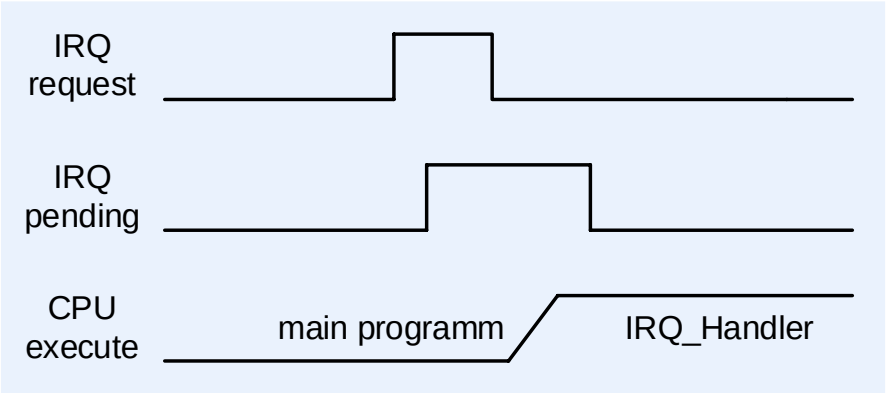


Рисунок 162 – Выставление отложенного запроса на прерывание и последующая его обработка

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 163.

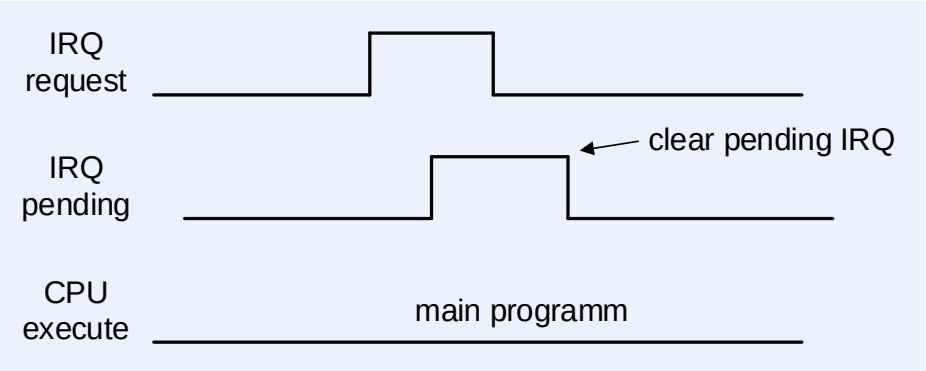


Рисунок 163 – Сброс признака отложенного прерывания, до обработки ядром

Если произойдет снятие запроса IRQ request от источника, «защелкивание» признака отложенного прерывания гарантирует отработку его ядром в соответствии с приоритетом. – рисунок 164. Сам IRQ pending признак снимается автоматически, когда прерывание становится активным, о чем сигнализирует признак IRQ active. Информация об активности соответствующего прерывания содержится в регистре IABR[x].

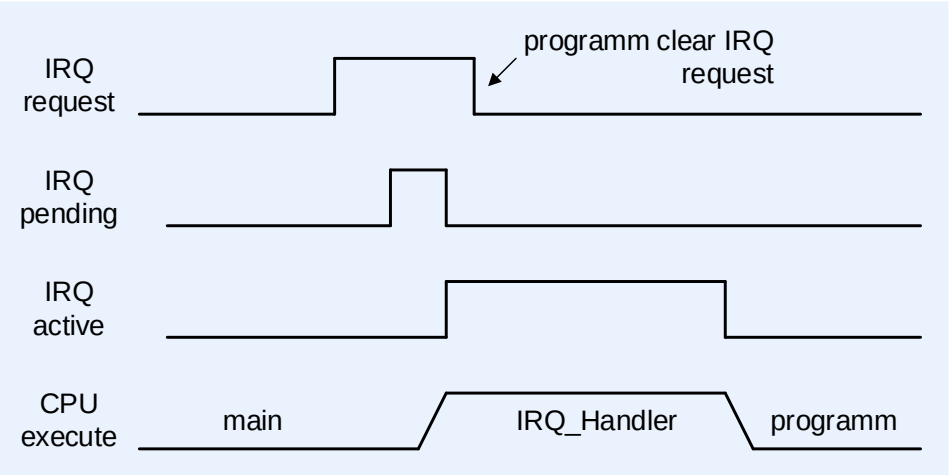


Рисунок 164 – Сброс признака отложенного прерывания, до обработки ядром

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 165.

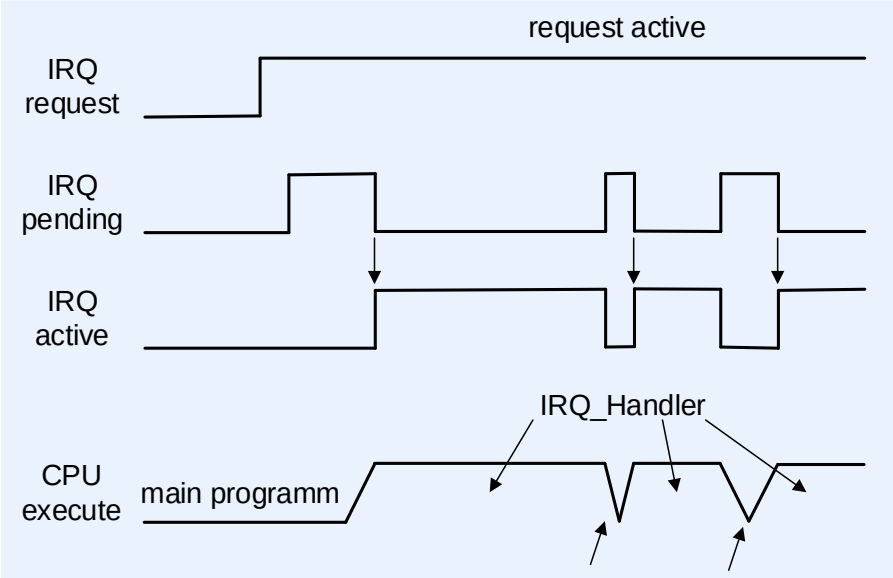


Рисунок 165 – Повторная обработка прерываний при удержании запроса от источника

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 166.

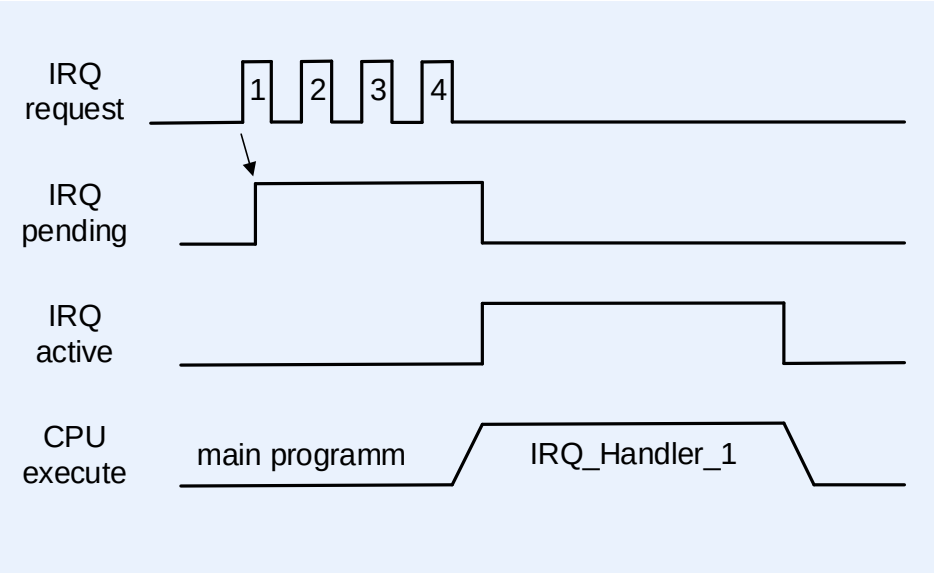


Рисунок 166 – Многократная установка снятие запроса IRQ request

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 5, 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 167.

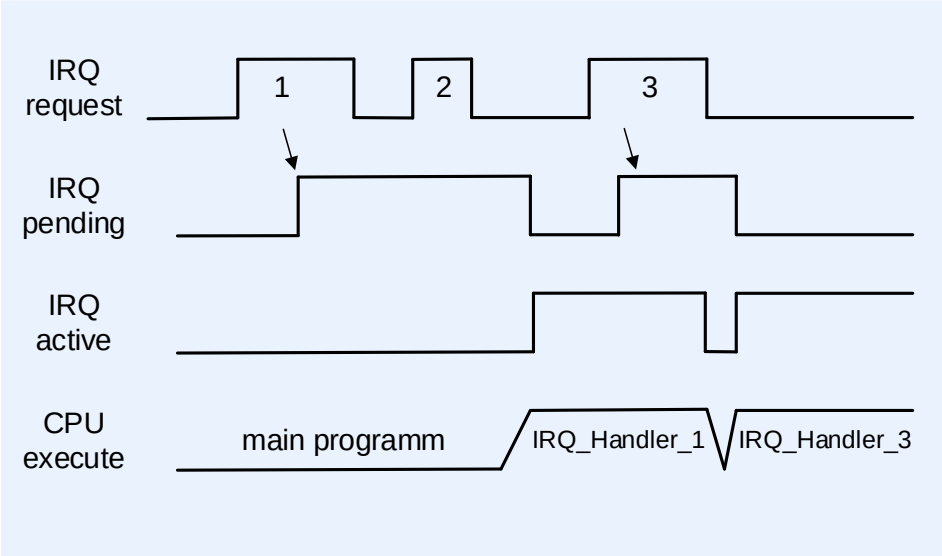


Рисунок 167 – Повторная установка запроса на прерывание
в момент выполнения обработчика исключения

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.

10.2 Упрощенный доступ к регистрам контроллера прерываний

В целях повышения эффективности разработки программного обеспечения в CMSIS предусмотрен упрощенный доступ к регистрам контроллера прерываний NVIC из среды разработки программного обеспечения:

– регистры разрешения, запрета, установки и сброса состояния ожидания прерываний, а также регистр активных прерываний отображаются на массивы 32-разрядных целых чисел, а именно:

- массив ISER[0] соответствует регистру ISER0;
- массив ICER[0] соответствует регистру ICER0;
- массив ISPR[0] соответствует регистру ISPR0;
- массив ICPR[0] соответствует регистру ICPR0;
- массив IABR[0] соответствует регистру IABR0;

– трехбитные поля регистра приоритета прерываний отображаются на массив 3-разрядных целых чисел, а именно:

– массив IP[0]...IP[29] соответствует регистрам IPR0-IPR7, причем элемент массива IP[n] соответствует приоритету прерывания с номером n.

CMSIS генерирует код, гарантированно обеспечивающий в условиях многозадачности корректный непрерываемый (atomic) доступ к регистрам приоритета.

Более подробная информация изложена в описании функции NVIC_SetPriority в разделе «Рекомендации по программированию контроллера прерываний NVIC».

В таблице 466 показано отображение прерываний (номеров запросов IRQ) на регистры прерываний и соответствующие переменные CMSIS, для которых предусмотрено по одному биту на прерывание.

Таблица 466 – Распределение прерываний в переменных прерывания

Номер прерывания	Элементы массивов CMSIS*				
	Разрешение	Запрет	Установка режима ожидания	Сброс режима ожидания	Признак активности
0 – 31	ISER[0]	ICER[0]	ISPR[0]	ICPR[0]	IABR[0]
* Каждый элемент массива соответствует одному регистру контроллера прерываний NVIC, например, элемент ICER[1] соответствует регистру ICER1					

10.2.1 NVIC->ISER[x]

Регистр ISER0 предназначен для разрешения прерываний (запись) и определения, какие из прерываний разрешены (чтение).

Таблица 467 – Регистр разрешения прерываний

Номер	31…0
Доступ	R/W
Сброс	0
	SETENA bits

Назначение бит **SETENA**:

запись: 0 – не влияет,
1 – разрешение прерывания;

чтение: 0 – прерывание запрещено,
1 – прерывание разрешено.

При разрешении прерывания, находящегося в состоянии ожидания обработки, контроллер NVIC активизирует его в зависимости от приоритета. Запрос запрещенного прерывания переводит его в состояние ожидания обработки, однако контроллер NVIC не активизирует его вне зависимости от приоритета.

10.2.2 NVIC->ICER[x]

Регистр запрета прерываний

Регистр ICER0 предназначен для запрета прерываний (запись) и определения, какие из прерываний разрешены (чтение).

Таблица 468 – Регистр запрета прерываний

Номер	31… 0
Доступ	R/W

Сброс	0
	CLRENA

Назначение бит CLRENA:
запись: 0 – не влияет,
1 – запрет прерывания;
чтение: 0 – прерывание запрещено,
1 – прерывание разрешено.

10.2.3 NVIC->ISPR[x]

Регистр установки состояния ожидания для прерывания

Регистр ISPR0 предназначен для принудительного перевода прерываний в состояние ожидания обслуживания (запись) и определения, какие из прерываний находятся в этом состоянии (чтение).

Таблица 469 – Регистр установки состояния ожидания для прерывания

Номер	31…0
Доступ	R/W
Сброс	0
	SETPEND

Назначение бит SETPEND:
запись: 0 – не влияет,
1 – перевод прерывания в состояние ожидания;
чтение: 0 – прерывание не ожидает обслуживания,
1 – прерывание ожидает обслуживания.
Запись 1 в бит регистра ISPR, соответствующий:
- прерыванию, уже ожидающему обслуживания – не влияет на работу системы;
- запрещенному прерыванию – переводит его в состояние ожидания.

10.2.4 NVIC->ICPR[x]

Регистр сброса состояния ожидания для прерывания

Регистр ICPR0 предназначен для принудительного сброса состояния ожидания обслуживания прерывания (запись) и определения, какие из прерываний находятся в состоянии ожидания (чтение).

Таблица 470 – Регистр сброса состояния ожидания для прерывания

Номер	31…0
Доступ	R/W
Сброс	0
	CLRPEND

Назначение бит CLRPEND:
запись: 0 – не влияет,

1 – сброс состояния ожидания;

чтение: 0 – прерывание не ожидает обслуживания,

1 – прерывание ожидает обслуживания.

Запись 1 в разряд регистра ICPR, соответствующий прерыванию в активном состоянии, не влияет на работу системы.

10.2.5 NVIC->IABR[x]

Регистр активных прерываний

Регистр ICPR0 показывает, какие из прерываний находятся в активном состоянии. Этот регистр доступен только для чтения (см. таблицу 471).

Таблица 471 – Регистр активных прерываний

Номер	31…0
Доступ	RO
Сброс	0
	ACTIVE

Назначение бит ACTIVE:

чтение: 0 – прерывание не активно;

1 – прерывание активно и обслуживается, либо активно и ожидает обслуживания.

10.2.6 NVIC->IP[x]

Регистры приоритета прерываний

Регистры IPR0-IPR7 представляют собой набор трехбитных полей, каждое из которых соответствует одному прерыванию. Регистры доступны побайтно.

Каждый из регистров содержит четыре поля приоритета, которые отображаются на четыре элемента массива IP[0] .. IP[29] CMSIS, как показано ниже.

Таблица 472 – Регистры приоритета прерываний

IP

Номер	31...16	15...8	7...0
Доступ	U	R/W	R/W
Сброс	0	0	0
	-	IP[29]	IP[28]

IP

Номер	31...24	23...16	15...8	7...0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	IP[4m+3]	IP[4m+2]	IP[4m+1]	IP[4m]

IP

Номер	31...24	23...16	15...8	7...0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	IP[3]	IP[2]	IP[1]	IP[0]

Каждое поле содержит значение приоритета в диапазоне от 0 до 7, причем меньшие значения соответствуют более высокому приоритету соответствующего прерывания. Процессор обеспечивает доступ только к битам [7:5] приоритета, биты [4:0] при чтении всегда равны нулю, а при записи игнорируются.

Номер регистра IPR и смещение данных в регистре для заданного номера прерывания N определяются следующими соотношениями:

- номер M соответствующего регистра приоритета равен M = N DIV 4;
- смещение данных в регистре в зависимости от значения N MOD 4 равно:
  - 0 – биты регистра [7:0];
  - 1 – биты регистра [15:8];
  - 2 – биты регистра [23:16];
  - 3 – биты регистра [31:24].

10.2.7 NVIC->STIR

Регистр программного формирования прерывания

Запись в регистр STIR приводит к формированию в системе программного прерывания (SGI – Software Generated Interrupt).

В случае если бит USERSETMPEND в регистре SCR установлен в 1, возможен доступ к регистру STIR из непривилегированных приложений (см. пункт 11.1.6 «Регистр управления системой»). Установка этого бита возможна только из привилегированного режима работы процессора.

Таблица 473 – Регистр программного формирования прерывания

Номер	31...9	8...0
Доступ	U	R/W
Сброс	0	0
	-	INTID

INTID – идентификатор формируемого прерывания в диапазоне 0 – 239.

Например: значение b000000011 соответствует прерыванию IRQ3.

10.3 Прерывания, срабатывающие по уровню сигнала

Процессор способен обрабатывать прерывания, сформированные по уровню сигнала. Формирование запроса на прерывание по уровню происходит при условии удержания сигнала не менее двух тактов процессорного ядра.

Прерывание такого типа считается активным до тех пор, пока периферийное устройство не снимет активный уровень сигнала запроса. Как правило, это происходит

после соответствующего обращения процедуры обработки прерывания к периферийному устройству.

После того, как процессор передал управление на обработчик, он автоматически снимает признак ожидания обслуживания прерывания (см. подраздел 10.4 «Аппаратное и программное управление прерываниями»). Если прерывание формируется по уровню сигнала, а сигнал запроса не снят до возврата из обработчика, процессор вновь переведет прерывание в состояние ожидания обслуживания, что, в свою очередь, приведет к повторному вызову его обработчика. Таким образом, периферийное устройство может поддерживать сигнал запроса прерывания в активном состоянии до тех пор, пока не перестанет нуждаться в обслуживании.

10.4 Аппаратное и программное управление прерываниями

Процессор Cortex-M4 регистрирует все поступающие прерывания. Перевод прерывания, сформированного периферийным устройством, в состояние ожидания обслуживания осуществляется в одном из следующих случаев:

контроллер прерываний NVIC обнаруживает, что сигнал запроса имеет высокий логический уровень, а прерывание неактивно;

контроллер прерываний NVIC обнаруживает передний фронт сигнала запроса прерывания;

программное обеспечение осуществляет запись в соответствующий разряд регистра ISPR0 (см. пункт 10.2.3 «NVIC->ISPR[x]») или соответствующего значения в регистр STIR (см. пункт 10.2.7 «NVIC->STIR»).

Прерывание находится в состоянии ожидания до тех пор, пока не произойдет одно из следующих событий:

– процессор передаст управление процедуре обработки прерывания. В этом случае прерывание переходит в активное состояние, после чего:

– по завершении обработки прерывания, срабатывающего по уровню, контроллер NVIC проверяет состояние сигнала запроса на прерывание. Если этот сигнал активен, прерывание вновь переводится в состояние ожидания обслуживания, что приводит к немедленной повторной передаче управления на обработчик. В противном случае прерывание переводится в неактивное состояние;

– если в период выполнения процедуры обработки прерывания, настроенного на срабатывание по фронту, не было зафиксировано импульсов на линии запроса, прерывание переводится в неактивное состояние.

– программное обеспечение осуществляет запись в соответствующий разряд регистра сброса состояния ожидания прерывания.

10.5 Рекомендации по работе с контроллером прерываний

Доступ к регистрам контроллера из программного обеспечения должен осуществляться по корректно выровненным адресам. Процессор не поддерживает

возможность доступа к контроллеру по невыровненным адресам. Требования по выравниванию приведены в описании регистров.

Прерывание может быть переведено в состояние ожидания обслуживания даже в случае, если оно запрещено.

Перед установкой нового адреса таблицы векторов прерывания необходимо убедиться, что элементы новой таблицы корректно проинициализированы адресами обработчиков отказов и всех разрешенных исключений, в частности, прерываний. Более подробная информация представлена в пункте 11.1.4 «SCB->VTOR».

Программное разрешение или запрещение прерываний может осуществляться с помощью инструкций CPSIE I и CPSID I. В CMSIS предусмотрены следующие встроенные функции, генерирующие эти инструкции:

void __disable_irq(void) // Disable Interrupts
void __enable_irq(void) // Enable Interrupts

Кроме того, в CMSIS имеется ряд дополнительных функций, обеспечивающих управление контроллером прерываний NVIC:

Таблица 474 – Функции CMSIS для управления контроллером прерываний

Функция	Описание
void NVIC_SetPriorityGrouping (uint32_t priority_grouping)	Установить группировку приоритетов
void NVIC_EnableIRQ (IRQn_t IRQn)	Разрешить IRQn
void NVIC_DisableIRQ (IRQn_t IRQn)	Запретить IRQn
uint32_t NVIC_GetPendingIRQ (IRQn_t IRQn)	Вернуть TRUE, если прерывание IRQn ожидает обслуживания, FALSE – в противном случае
void NVIC_SetPendingIRQ (IRQn_t IRQn)	Перевести IRQn в состояние ожидания обслуживания
void NVIC_ClearPendingIRQ (IRQn_t IRQn)	Сбросить состояние ожидания обслуживания для IRQn
uint32_t NVIC_GetActive (IRQn_t IRQn)	Вернуть номер IRQ текущего активного прерывания
void NVIC_SetPriority (IRQn_t IRQn, uint32_t priority)	Установить приоритет для IRQn
uint32_t NVIC_GetPriority (IRQn_t IRQn)	Считать приоритет IRQn
void NVIC_SystemReset (void)	Сбросить систему

Более подробная информация отражена в документации по CMSIS.

11 Блок управления системой

Блок управления системой SCB обеспечивает доступ к информации о конфигурации и управление работой системы. Регистры блока управления системой представлены в таблице 475.

Таблица 475 – Обобщенная информация о регистрах блока управления системой

Адрес	Имя	Тип	Доступ	Значение после сброса	Описание
0xE000E000	InterruptType				
0x008	ACTLR	RW	Привилегированный	0x00000000	Дополнительный регистр управления
0xE000ED00	SCB				Блок управления системой
0x000	CPUID	RO	Привилегированный	0x412FC240	Регистр идентификации процессора
0x004	ICSR	RW	Привилегированный	0x00000000	Регистр управления прерываниями
0x008	VTOR	RW	Привилегированный	0x00000000	Регистр смещения таблицы векторов прерываний
0x00C	AIRCR	RW	Привилегированный	0xFA050000	Регистр управления прерываниями и программного сброса
0x010	SCR	RW	Привилегированный	0x00000000	Регистр управления системой
0x014	CCR	RW	Привилегированный	0x00000200	Регистр конфигурации и управления
0x018	SHPR1	RW	Привилегированный	0x00000000	Регистр №1 приоритета системных обработчиков
0x01C	SHPR2	RW	Привилегированный	0x00000000	Регистр №2 приоритета системных обработчиков
0x020	SHPR3	RW	Привилегированный	0x00000000	Регистр №3 приоритета системных обработчиков
0x024	SHCRS	RW	Привилегированный	0x00000000	Регистр управления и состояния системных обработчиков
0x028	CFSR	RW	Привилегированный	0x00000000	Регистр состояния отказов с конфигурируемым приоритетом
0x028	MMSR	RW	Привилегированный	0x00	Регистр состояния отказов доступа к памяти
0x029	BFSR	RW	Привилегированный	0x00	Регистр состояния отказов доступа к шине
0x02A	UFSR	RW	Привилегированный	0x0000	Регистр состояния отказов, вызванных ошибками программирования
0x02C	HFSR	RW	Привилегированный	0x00000000	Регистр состояния тяжелого отказа

Адрес	Имя	Тип	Доступ	Значение после сброса	Описание
0x034	MMAR	RW	Привилеги-рованный	Не определено	Регистр адреса отказа доступа к памяти
0x038	BFAR	RW	Привилеги-рованный	Не определено	Регистр адреса отказа доступа к шине

11.1 Упрощенный доступ к регистрам блока управления системой

В целях повышения эффективности в CMSIS предусмотрен упрощенный доступ к регистрам SCB из среды разработки программного обеспечения, а именно, регистры SHPR1-SHPR3 в CMSIS отображаются на массив байтов SHP[0]...SHP[12].

11.1.1 InterrupType->ACTLR

Дополнительный регистр управления

Регистр ACTLR позволяет разрешить или запретить следующие возможности процессора:

– вложение условных инструкций (IT folding);

– использование буферизации записи в режиме отображения памяти по умолчанию (default memory map);

– прерывание многоэлементных инструкций чтения и записи регистров.

Обобщенные данные о регистре ACTLR представлены далее в таблице 476.

Таблица 476 – Дополнительный регистр управления

Номер	31...3	2	1	0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-	DISFOLD	DISDEFWBUF	DISMCYCINT

DISFOLD – установка разряда в 1 запрещает вложение условных инструкций (IT folding) (см. ниже п. «О вложении условных инструкций»).

DISDEFWBUF – установка в 1 запрещает использование буфера записи при работе в режиме отображения памяти по умолчанию (default memory map). Это обеспечивает возможность локализовать любые отказы шины, однако приводит к снижению производительности системы, так как все операции записи данных в память должны быть завершены до того, как процессор перейдет к выполнению следующей инструкции. Данный бит влияет исключительно на функционирование буферов записи, реализованных в процессоре Cortex-M4.

DISMCYCINT – установка бита в 1 запрещает прерывание многоэлементных инструкций чтения и записи регистров (LDM и STM). Это приводит к увеличению задержки обработки прерываний, вследствие необходимости завершения выполнения

инструкций LDM или STM перед началом сохранения контекста и передачи управления обработчику прерывания.

О вложении условных инструкций

В некоторых случаях процессор может начать выполнение первой инструкции в IT-блоке, все еще выполняя инструкцию IT. Эта возможность, называемая далее вложением условных инструкций (IT folding), позволяет увеличить производительность системы, однако может привести к непостоянству времени выполнения тела цикла программы («джиттеру»). В случае если в разрабатываемом приложении это нежелательно, следует установить бит DISFOLD в 1.

11.1.2 SCB->CPUID

Регистр идентификации процессора.

Регистр CPUID содержит информацию о модели процессора, версии и варианте его реализации. Подробная информация о регистре представлена в таблице 477.

Таблица 477 – Регистр идентификации процессора

Номер	31...24	23...20	19...16	15...4	3...0
Доступ	RO	RO	RO	RO	RO
Сброс	0x41	0x2	0xF	0xC24	0x0
	Implementer	Variant	Constant	PartNo	Revision

Implementer – код разработчика 0x41 = ARM.

Variant – значение r в номере версии rnpn изделия: 0x2 = r2p0;

Constant – постоянное значение 0xF;

PartNo – номер модели процессора: 0xC24 = Cortex-M4;

Revision – значение p в номере версии rnpn изделия: 0x0 = r2p0.

11.1.3 SCB->ICSR

Регистр управления прерываниями.

Регистр ICSR обеспечивает возможность установки и сброса состояния ожидания обслуживания для исключений PendSV и SysTick, а также доступ к следующей информации:

- номер текущего обрабатываемого исключения;
- наличие активных исключений, обработка которых была прервана;
- номер исключения, ожидающего обслуживания, с наивысшим приоритетом;
- наличие прерываний, ожидающих обслуживания.

Таблица 478 – Регистр управления прерываниями

Номер	31...29	28	27	26	25	24	23	22	21...12	11	10, 9	8...0
Доступ	U	R/W	R/W	R/W	R/W	U	R/W	R/W	R/W	R/W	U	R/W
Сброс	0	0	0	0	0	0	0	0	0	0	0	0

	-	PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	-	Reserved for Debug	ISRPENDING	VECTPENDING	RETTOBASE	-	VECTACTIVE

PENDSVSET (RW) – бит установки состояния ожидания обслуживания для исключения PendSV.

Запись 0 – не влияет на работу системы,

1 – переводит исключение PendSV в состояние ожидания обслуживания.

Чтение 0 – исключение PendSV не ожидает обслуживания,

1 – ожидает.

Запись 1 – это единственно возможный способ перевода исключения PendSV в состояние ожидания обслуживания.

PENDSVCLR (WO) – бит сброса состояния ожидания обслуживания для исключения PendSV.

Запись 0 – не влияет на работу системы.

Запись 1 – сбрасывает состояние ожидания обслуживания для исключения PendSV.

PENDSTSET (RW) – бит установки состояния ожидания обслуживания для исключения SysTick.

Запись 0 – не влияет на работу системы.

Запись 1 – переводит исключение SysTick в состояние ожидания обслуживания.

Чтение 0 – исключение SysTick не ожидает обслуживания.

1 – ожидает.

PENDSTCLR (WO) – бит сброса состояния ожидания обслуживания для исключения SysTick.

Запись 0 – не влияет на работу системы.

Запись 1 – сбрасывает состояние ожидания обслуживания для исключения SysTick.

Данный бит доступен только для записи, при чтении результат не определен.

Reserved for Debug use (RO) – этот бит зарезервирован для целей отладки, при чтении вне режима отладки возвращает значение 0.

ISRPENDING (RO) – флаг наличия в системе прерываний (за исключением отказов), ожидающих обслуживания. 0 – ожидающие обслуживания прерывания отсутствуют, 1 – присутствуют.

VECTPENDING (RO) – содержит номер ожидающего обслуживания исключения с наивысшим приоритетом, обработка которого в системе разрешена. **0** – необслуженных исключений нет, другое число – номер ожидающего обслуживания исключения.

Значение данного поля формируется с учетом полей BASEPRI и FAULTMASK, однако не учитывает влияние поля PRIMASK.

RETTOBASE (RO) – показывает наличие в системе активных исключений, обслуживание которых было прервано. 0 – присутствуют, 1 – отсутствуют.

VECTACTIVE (RO) – содержит номер активного исключения. 0 – режим приложения, другое число – номер текущего обслуживаемого исключения. Для получения номера запроса прерывания (IRQ) из значения VECTACTIVE необходимо вычесть 16.

Запись в регистр ICSR может привести к непредсказуемым результатам в случае:
- одновременной установки в 1 бит PENDSVSET и PENDSVCLR;
- одновременной установки в 1 бит PENDSTSET и PENDSTCLR.

11.1.4 SCB->VTOR

Регистр смещения таблицы векторов прерываний

Регистр VTOR содержит смещение базового адреса таблицы векторов прерываний относительно адреса 0x00000000.

Таблица 479 – Регистр смещения таблицы векторов прерываний

Номер	31, 30	29…7	6...0
Доступ	U	R/W	R/W
Сброс	0	0	0
	-	TBLOFF	Reserved

TBLOFF – смещение базового адреса таблицы векторов относительно нижней границы карты распределения памяти. Собственно, смещение хранится в битах [28:7]. Бит [29] определяет, размещена ли таблица в области кода или в области памяти SRAM: 0 – область кода, 1 – SRAM. Бит [29] может также обозначаться как TBLBASE.

При установке значения TBLOFF требуется обеспечить выравнивание базового адреса таблицы векторов. Минимальный размер выравнивания – по границе блока из 32 слов, достаточен для хранения 16 векторов прерываний. Для поддержки большего количества прерываний необходимо увеличить размер выравнивания до ближайшей степени двойки, большей или равной размеру таблицы. Например, для хранения 21 вектора прерываний таблицу следует выровнять по границе блока из 64 слов, так как ее объем составляет 37 слов, а ближайшая степень двойки, большая или равная 37, равна 64.

Учитывая описанные выше требования по выравниванию, разряды [6…0] смещения всегда равны нулю.

11.1.5 SCB->AIRCR

11.1.5.1 **Регистр управления прерываниями и программного сброса**

Регистр AIRCR позволяет группировать исключения по приоритетам, задавать порядок следования байтов в слове (endian) при доступе к данным, а также управлять процессом сброса системы.

Для записи данных в регистр необходимо установить его поле VECTKEY в значение 0x05FA, в противном случае попытка записи будет проигнорирована процессором.

Таблица 480 – Регистр управления прерываниями и программного сброса

Номер	31...16	15	14…11	10…8	7...3	2	1	0
Доступ	R/W	R/W	U	R/W	U	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	On Read: VECTKEYSTAT, On Write: VECTKEY	ENDIANESS	-	PRIGROUP	-	SYSRESETREQ	VECTCLRACTIVE	VECTRESET

VECTKEYSTAT – ключ доступа к регистру. При чтении возвращает 0xFA05.

VECTKEY – ключ доступа к регистру. При записи должен быть равен 0x05FA, в противном случае попытка записи в регистр будет проигнорирована процессором.

ENDIANESS (RO) – порядок следования значащих разрядов при доступе к данным. 0 – младший байт идет первым (little-endian), 1 – старший байт идет первым (big-endian). Значение поля устанавливается, исходя из уровня конфигурационного сигнала BIGEND в момент сброса системы.

PRIGROUP (RW) – группировка приоритетов исключений. Значение данного поля определяет положение двоичной точки, разделяющей поле приоритета на поля номера группы и подгруппы приоритетов.

PRIGROUP – определяет позицию двоичной точки, разделяющей поля PRI_n регистров приоритета прерываний на два подполя – номер группы и номер подгруппы. Зависимость этого разбиения от значения PRIGROUP показана в таблице 481.

Таблица 481 – Группировка приоритетов прерываний

PRIGROUP	Значение приоритета в поле PRI_N[7:0]			Общее количество	
	Положение двоичной точки	Биты номера группы	Биты номера подгруппы	Групп	подгрупп
0b000, 0b001, 0b010, 0b011, 0b100	bxxx.00000	[7:5]	None	8	1

0b101	bxx.y00000	[7:6]	5	4	2
0b110	bx.yy00000	[7]	[6:5]	2	4
0b111	b.yyy00000	None	[7:5]	1	8

SYSRESETREQ (WO) – запрос сброса системы. 0 – не влияет на работу, 1 – инициирует сигнал сброса процессора. При чтении возвращает 0.

VECTCLRACTIVE (WO) – зарезервировано для целей отладки. При чтении возвращает 0. При записи данных в регистр значение поля должно быть равно 0, в противном случае результат непредсказуем.

VECTRESET (WO) – зарезервировано для целей отладки. При чтении возвращает 0. При записи данных в регистр значение поля должно быть равно 0, в противном случае результат непредсказуем.

11.1.6 SCB->SCR

Регистр управления системой

Регистр SCR позволяет определить требования к переходу в режим и выходу из режима пониженного энергопотребления.

Таблица 482 – Регистр управления системой

Номер	31…5	4	3	2	1	0
Доступ	U	R/W	U	R/W	R/W	U
Сброс	0	0	0	0	0	0
	-	SEVONPEND	-	SLEEPDEEP	SLEEONEXIT	-

SEVONPEND – разрешает или запрещает формирование сигнала события при переводе исключения в состояние ожидания обработки. 0 – выход из режима пониженного энергопотребления по прерыванию могут инициировать только разрешенные прерывания или события; 1 – выход может инициироваться разрешенными событиями и любыми, в том числе запрещенными, прерываниями.

Перевод прерывания в состояние ожидания обслуживания формирует событие, что в свою очередь приводит к выходу процессора из режима пониженного потребления, инициированного инструкцией WFE, либо к регистрации факта события, если эта инструкция еще не выполнялась.

Кроме того, процессор может быть выведен из режима пониженного энергопотребления при поступлении внешнего события, а также после выполнения инструкции SEV.

SLEEPDEEP – определяет режим пониженного энергопотребления процессора:

0 – спящий режим (Sleep);

1 – режим глубокого сна (Deep Sleep).

SLEEPONEXIT – разрешает или запрещает перевод процессора в режим пониженного энергопотребления при выходе из обработчика события в режим выполнения прикладной программы: 0 – не переводить, 1 – переводить.

11.1.7 SCB->CCR

Регистр конфигурации и управления

Регистр CCR управляет процессом перехода процессора в режим приложения, а также позволяет запретить или разрешить:

– игнорирование отказов доступа к шине в обработчиках тяжелых отказов и при эскалации отказа по FAULTMASK;

– генерацию исключений при делении на ноль и при доступе по невыровненному адресу;

– доступ к регистру STIR из непривилегированного приложения (см. пункт 10.2.7 «NVIC->STIR»).

Таблица 483 – Регистр конфигурации и управления

Номер	31...10	9	8	7...5	4	3	2	1	0
Доступ	U	R/W	R/W	U	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	0	0	0	0	0
	-	STKALIGN	BFHFNMIGN	-	DIV_O_TRP	UNALIGN_TRP	-	USERSETMPEND	NONBASETHRDENA

STKALIGN определяет режим выравнивания адреса стека при обработке исключений:

0 – выравнивание по границе 4 байт;

1 – выравнивание по границе 8 байт.

При передаче управления на обработчик исключения процессор анализирует бит [9] сохраненного в стеке слова состояния PSR и определяет по нему режим выравнивания стека. При возврате из обработчика процессор использует сохраненный в стеке бит этого слова для восстановления требуемого режима выравнивания.

BFHFNMIGN разрешает обработчикам с уровнем приоритета -1 и -2 игнорировать отказы доступа к шине, вызванные инструкциями чтения и записи. Бит влияет на функционирование обработчиков тяжелых отказов и при эскалации отказов по FAULTMASK:

0 – отказы доступа к шине данных, вызванные инструкциями чтения или записи, приводят к блокировке процессора;

1 – обработчики с уровнем приоритета -1 и -2 игнорируют указанные отказы доступа к шине данных.

Данный бит следует устанавливать лишь в том случае, если обработчик и используемые им данные размещены в абсолютно безопасной области памяти. Как правило, данный бит используется для локализации и исправления проблем доступа к системным устройствам и мостам ввода-вывода.

DIV_0_TRP разрешает процессору формировать отказ или останавливаться в случае деления на ноль при выполнении инструкций SDIV или UDI:

0 – не обрабатывать деление на 0;

1 – обрабатывать.

В случае если бит установлен в 0, при делении на ноль процессор устанавливает частное в 0.

UNALIGN_TRP разрешает процессору формировать отказ при невыровненном доступе к данным:

0 – не обрабатывать невыровненный доступ к словам или полусловам данных;

1 – обрабатывать.

Если бит равен 1, то невыровненный доступ приводит к отказу, вызванному ошибкой программирования (usage fault).

В случае невыровненного доступа по инструкциям LDM, STM, LDRD или STRD отказ формируется всегда, вне зависимости от значения бита UNALIGN_TRP.

USERSETMPEND разрешает доступ к регистру STIR (см. пункт 10.2.7 «NVIC->STIR») из непривилегированного приложения:

0 – доступ запрещен;

1 – разрешен.

NONEBASETHRDENA определяет процедуру перехода процессора в режим приложения (Thread mode): 0 – процессор может перейти в режим приложения только в случае отсутствия активных исключений, 1 – процессор может перейти в режим приложения из обработчика любого уровня, в соответствии со значением слова EXC_RETURN (см. пункт 9.6.6 «Возврат из обработчика исключения»).

11.1.8 SCB->SHP[x]

Регистры приоритета системных обработчиков

Регистры приоритета системных обработчиков SHPR1-SHPR3 позволяют установить уровень приоритета обработки исключений.

Доступ к регистрам осуществляется побайтно.

Поля PRI_N регистров имеют ширину 8 бит, однако в процессоре реализована поддержка доступа только к старшему полубайту [7…4], при чтении данных из младшего полубайта [3…0] процессор возвращает нули, запись в этот полубайт игнорируется.

Таблица 484 – Поля приоритета обработчиков системных отказов

Обработчик отказа	Поле	Описание регистра
Отказ доступа к памяти	SHP[4]	Регистр №1 приоритета системных обработчиков

Обработчик отказа	Поле	Описание регистра
Отказ доступа к шине	SHP[5]	
Ошибка программирования (usage fault)	SHP[6]	
Вызов SVCall	SHP[11]	Регистр №2 приоритета системных обработчиков
Вызов PendSV	SHP[14]	Регистр №3 приоритета системных обработчиков
Вызов SysTick	SHP[15]	

Регистр №1 приоритета системных обработчиков

Таблица 485 – Регистр №1 приоритета системных обработчиков

Номер	31...24	23...16	15...8	7...0
Доступ	R/W	R/W	R/W	R/W
Сброс	0	0	0	0
	PRI_7: Резерв	PRI_6	PRI_5	PRI_4

PRI_7 Резерв.
PRI_6 Приоритет системного обработчика 6, ошибка программирования
PRI_5 Приоритет системного обработчика 5, отказ доступа к шине
PRI_4 Приоритет системного обработчика 4, отказ доступа к памяти

Регистр №2 приоритета системных обработчиков

Таблица 486 – Регистр №2 приоритета системных обработчиков

Номер	31...24	23...0
Доступ	R/W	U
Сброс	0	0
	PRI_11	-

PRI_11 Приоритет системного обработчика 11, вызов SVCall

Регистр №3 приоритета системных обработчиков

Таблица 487 – Регистр №3 приоритета системных обработчиков

Номер	31...24	23...16	15... 0
Доступ	R/W	R/W	U
Сброс	0	0	0
	PRI_15	PRI_14	.

PRI_15 Приоритет системного обработчика 15, вызов SysTick
PRI_14 Приоритет системного обработчика 14, вызов PendSV

11.1.9 SCB->SHCSR

Регистр управления и состояния системных обработчиков.

Регистр SHCSR позволяет разрешить или запретить работу системных обработчиков, а также содержит сведения:
о наличии ожидающих обработки отказов доступа к шине, управления памятью, а также вызов SVCall;
об активных системных обработчиках.

Таблица 488 – Регистр управления и состояния системных обработчиков

Номер	31...19	18	17	16	15	14	13	12	11
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
	-	USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA	SVCALL PENDED	BUSFAULT PENDED	MEMFAUL TPENDED	USGFAULT PENDED	SYSTICK ACT

Номер	10	9	8	7	6...4	3	2	1	0
Доступ	R/W	U	R/W	R/W	U	R/W	U	R/W	R/W
Сброс	0	0	0	0	0	0	0	0	0
	PENDSV ACT	-	MONITOR ACT	CVCALL AVCT	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT

USGFAULTENA разрешение обработки отказов, вызванных ошибками программирования, 1 – разрешено, 0 – запрещено.

BUSFAULTENA разрешение обработки отказа доступа к шине, 1 – разрешено, 0 – запрещено.

MEMFAULTENA разрешение обработки отказа доступа к памяти, 1 – разрешено, 0 – запрещено.

SVCALLPENDED признак ожидания обработки вызова SVC, возвращает 1, если вызов ожидает обработки.

BUSFAULTPENDED признак ожидания обработки отказа доступа к шине, возвращает 1, если отказ ожидает обработки.

MEMFAULTPENDED признак ожидания обработки отказа доступа к памяти, возвращает 1, если отказ ожидает обработки.

USGFAULTPENDED признак ожидания обработки отказа, вызванного ошибками программирования, возвращает 1, если отказ ожидает обработки.

SYSTICKACT признак активности обработчика исключения SysTick, возвращает 1, если обработчик активен.

PENDSVACT признак активности обработчика исключения PendSV, возвращает 1, если обработчик активен.

MONITORACT признак активности монитора отладчика, возвращает 1, если монитор отладчика активен.

SVCALLACT признак активности обработчика вызова SVC, возвращает 1, если обработчик активен.

USGFAULTACT признак активности обработчика отказа, вызванного ошибкой программирования, возвращает 1, если обработчик активен.

BUSFAULTACT признак активности обработчика отказа доступа к шине, возвращает 1, если обработчик активен.

MEMFAULTACT признак активности обработчика отказа доступа к памяти, возвращает 1, если обработчик активен.

Примечания

1 Установка бита разрешения в 1 разрешает обработку исключения, установка в 0 – запрещает.

2 Чтение 1 из бита-признака активности свидетельствует об активности исключения, 0 – о его неактивности. Существует возможность записи значения в данный бит для принудительного перевода исключения в активное состояние, однако при этом следует предпринять меры предосторожности, описанные далее в разделе;

3 Чтение 1 из бита-признака ожидания свидетельствует о том, что исключение находится в состоянии ожидания обработки. Существует возможность принудительного перевода исключения в состояние ожидания путем записи 1 в данный бит.

Если в системе возникло исключение (отказ), обработчик которого запрещен, процессор формирует запрос на обработку тяжелого отказа.

Существует возможность принудительного перевода того или иного системного исключения в состояние ожидания обработки или активное состояние путем записи в соответствующий разряд регистра SHCSR.

Например, ядро операционной системы может осуществлять запись в биты – признаки активности для того, чтобы осуществить переключение контекста со сменой типа обрабатываемого исключения.

Программа, меняющая значение бит – признаков активности исключения, должна обеспечить необходимую корректировку содержимого стека, в противном случае процессор может сгенерировать отказ. Необходимо убедиться, что программа сохраняет и впоследствии корректно восстанавливает текущее значение признаков активности исключений.

После разрешения системных обработчиков все дальнейшие манипуляции с битами регистра необходимо производить, последовательно выполняя операции чтения, модификации и обратной записи, гарантирующие изменение только необходимых разрядов регистра.

11.1.10 SCB->CFSR

Регистр состояния отказов с конфигурируемым уровнем приоритета.

Регистр CFSR содержит информацию о причине возникновения отказов управления памятью, отказов доступа к шине и ошибок программирования (usage fault).

Таблица 489 – Регистр состояния отказов с конфигурируемым уровнем приоритета

Номер	31...16	15...8	7...0
Доступ	RO	RO	RO
Сброс	0	0	0
	Usage Fault Status Register: UFSR	Bus Fault Status Register: BFSR	Memory Management Fault Status Register: MMFSR

Регистр CFSR доступен побайтно. Возможны следующие варианты доступа к регистру CFSR и его отдельным элементам:

- слово по адресу 0xE000ED28 – полный регистр CFSR;
- байт по адресу 0xE000ED28 – регистр MMFSR;
- полуслово по адресу 0xE000ED28 – регистры MMFSR и BFSR;
- байт по адресу 0xE000ED29 – регистр BFSR;
- полуслово по адресу 0xE000ED2A – регистр UFSR.

В последующих подразделах подробно описаны элементы, составляющие регистр CFSR:

- регистр состояния отказов доступа к памяти;
- регистр состояния отказов доступа к шине;
- регистр состояния отказов, вызванных ошибками программирования.

11.1.10.1 **Поле MMFSR**

Регистр состояния отказов доступа к памяти

Регистр MMFSR содержит набор флагов, указывающих на различные причины отказа доступа к памяти.

Таблица 490 – Регистр состояния отказов доступа к памяти

Номер	7	6, 5	4	3	2	1	0
Доступ	RO	U	RO	RO	U	RO	RO
Сброс	0	0	0	0	0	0	0
	MMARVALID	-	MSTKERR	MUNSTKERR	-	DACCVIOL	IACCVIOL

MMARVALID признак корректности значения в регистре адреса отказа доступа к памяти (MMAR): 0 – значение в MMAR не содержит корректный адрес отказа, 1 – содержит.

В случае если произошла эскалация отказа доступа к памяти, обработчик тяжелого отказа должен установить этот бит в 0. В противном случае после возврата в обработчик отказа доступа к памяти возможна его некорректная работы, так как значение регистра MMAR будет изменено.

MSTKERR признак отказа на этапе сохранения в стеке контекста при передаче управления на обработчик исключения: 0 – отсутствует, 1 – попытка сохранения в стеке

контекста при вызове обработчика исключения вызвала одно или несколько нарушений доступа к памяти. В случае, если бит равен 1, значение указателя стека SP по-прежнему корректно, однако содержимое стека может быть неверным. Адрес отказа в регистр MMAR не записывается.

MUNSTKERR признак отказа на этапе восстановления контекста из стека при выходе из обработчика исключения: 0 – отсутствует, 1 – попытка восстановления контекста из стека вызвала одно или несколько нарушений доступа к памяти.

Передача управления на обработчик данного отказа осуществляется без сохранения контекста. Таким образом, в случае, если данный бит равен 1, состояние стека сохраняется, значение указателя стека не меняется, контекст не сохраняется.

Адрес отказа в регистр MMAR не записывается.

DACCVIOL признак нарушения доступа к памяти данных: 0 – отсутствует, 1 – процессор попытался прочесть или записать данные в области, для которой не разрешен такой тип доступа. Если бит равен 1, значение счетчика команд PC, сохраненное в стеке, указывает на инструкцию, вызвавшую отказ. В регистре MMAR содержится адрес, по которому была осуществлена попытка доступа к памяти.

IACCVIOL признак нарушения доступа к памяти команд: 0 – отсутствует, 1 – процессор попытался считать очередную команду из области памяти, для которой не разрешено выполнение. Этот отказ возникает всякий раз при доступе к области, помеченной как неразрешенная для выполнения (XN), даже в случае, если блок защиты памяти MPU не активен (disabled) или отсутствует. Если бит равен 1, значение счетчика команд PC, сохраненное в стеке, указывает на инструкцию, вызвавшую отказ. Адрес отказа в регистр MMAR не записывается.

11.1.10.2 **Поле BFSR**

Регистр состояния отказов доступа к шине

Регистр BFSR содержит набор флагов, указывающих на различные причины отказа доступа к шине:

Таблица 491 – Регистр состояния отказов доступа к шине

Номер	7	6, 5	4	3	2	1	0
Доступ	RO	U	RO	RO	RO	RO	RO
Сброс	0	0	0	0	0	0	0
	BFRVALID		STKERR	UNS TKERR	IMPRECI SERR	PRECI SERR	IBUSERR

BFARVALID признак корректности значения в регистре адреса отказа доступа к шине (BFAR): 0 – значение в BFAR не содержит корректный адрес отказа, 1 – содержит.

Процессор устанавливает этот бит в 1 в случае, если известен адрес, при доступе по которому произошел отказ. Возникновение впоследствии других отказов, например, отказов управления памятью, может сбросить этот бит в 0.

В случае если возникла эскалация отказа, обработчик тяжелого отказа должен установить этот бит в 0. В противном случае после возврата в обработчик отказа доступа

к шине возможна его некорректная работа, так как значение регистра MMAR будет изменено.

STKERR признак отказа на этапе сохранения в стеке контекста при передаче управления на обработчик исключения: 0 – отсутствует, 1 – попытка сохранения в стеке контекста при вызове обработчика исключения вызвала одно или несколько нарушений доступа к шине. В случае если бит равен 1, значение указателя стека SP по-прежнему корректно, однако содержимое стека может быть неверным. Адрес отказа в регистр BFAR не записывается.

UNSTKERR признак отказа на этапе восстановления контекста из стека при выходе из обработчика исключения: 0 – отсутствует, 1 – попытка восстановления контекста из стека вызвала одно или несколько нарушений доступа к шине.

Передача управления на обработчик данного отказа осуществляется без сохранения контекста. Таким образом, в случае, если данный бит равен 1, состояние стека сохраняется, значение указателя стека не меняется, контекст не сохраняется.

Адрес отказа в регистр BFAR не записывается.

IMPRECISERR признак нелокализованной ошибки доступа к шине данных. 0 – отсутствует, 1 – произошла ошибка доступа к шине данных, однако адрес возврата в стековом фрейме не указывает на инструкцию, вызвавшую ошибку. В случае, если процессор установил этот бит в 1, адрес отказа в регистр BFAR не записывается. Данный отказ является асинхронным, таким образом, если он возник внутри процесса, приоритет которого выше, чем приоритет обработки отказа шины, процессор переводит его в состояние ожидания обслуживания до завершения более приоритетных процессов. В случае, если до передачи управления на обработчик возникла также локализованная ошибка доступа к шине, процессор устанавливает оба соответствующих флага.

PRECISERR признак локализованной ошибки доступа к шине данных. 0 – отсутствует, 1 – произошла ошибка доступа к шине данных, при этом адрес возврата в стековом фрейме указывает на инструкцию, вызвавшую ошибку. В случае, если процессор установил этот бит в 1, он также записывает адрес отказа в регистр BFAR.

IBUSERR признак ошибки доступа к шине инструкций. 0 – отсутствует, 1 – произошла ошибка доступа к шине инструкций. Процессор обнаруживает факт ошибки доступа к шине инструкций на этапе выборки очередной команды, однако признак IBUSERR устанавливается только после попытки выполнения этой инструкции. В случае, если процессор установил этот бит в 1, адрес отказа в регистр BFAR не записывается.

11.1.10.3 **Поле UFSR**

Регистр состояния отказов, вызванных ошибками программирования

Регистр UFSR содержит набор флагов, указывающих на различные причины отказа.

Таблица 492 – Регистр состояния отказов, вызванных ошибками программирования

Номер	15...10	9	8	7...4	3	2	1	0
Доступ	U	RO	RO	U	RO	RO	RO	RO

Сброс	0	0	0	0	0	0	0	0
	-	DIV BYZERO	UN ALIGNED	-	NOCP	INV PC	INV STATE	UNDEF INSTR

DIVBYZERO признак деления на ноль: 0 – деления на ноль не было, либо обработка данного типа ошибки запрещена, 1 – процессор выполнил инструкцию SDIV или UDIV с делителем равным 0. Если бит равен 1, значение счетчика команд PC, сохраненное в стеке, указывает на инструкцию, вызвавшую отказ. Разрешить либо запретить обработку деления на ноль можно путем установки в 1 бита DIV_0_TRP регистра CCR (см. пункт 11.1.7 «SCB->CCR»).

UNALIGNED признак доступа к памяти по невыровненному адресу: 0 – не было, либо обработка данного типа ошибки запрещена, 1 – процессор попытался обратиться к памяти по невыровненному адресу. Разрешить либо запретить обработку этой ошибки можно путем установки в 1 бита UNALIGN_TRP регистра CCR (см. пункт 11.1.7 «SCB->CCR»).

Инструкции LDM, STM, LDRD, и STRD, пытающиеся обратиться по невыровненному адресу, вызывают исключение всегда, вне зависимости от значения бита UNALIGN_TRP.

NOCP попытка обращения к сопроцессору. Процессор не поддерживает инструкции, требующие наличия сопроцессора. 0 – не было, 1 – была.

INVPC загрузка неверного значения в счетчик команд PC. 0 – не было, 1 – процессор попытался загрузить в счетчик команд PC неверное значение EXC_RETURN, вследствие неправильного восстановления контекста, либо неверного значения EXC_RETURN. Если бит равен 1, значение счетчика команд PC, сохраненное в стеке, указывает на инструкцию, пытавшуюся загрузить неверное значение в PC.

INVSTATE неверное состояние: 0 – не было, 1 – процессор попытался выполнить инструкцию, связанную с неверным использованием регистра EPSR. Если бит равен 1, значение счетчика команд PC, сохраненное в стеке, указывает на инструкцию, попытавшуюся некорректно использовать регистр EPSR.

UNDEFINSTR попытка выполнения неверной инструкции. 0 – не было, 1 – процессор попытался выполнить неверной инструкцию. Если бит равен 1, значение счетчика команд PC, сохраненное в стеке, указывает на инструкцию, вызвавшую отказ. Под неверной понимается инструкция, которую процессор не смог декодировать.

После установки в 1 биты регистра UFSR сохраняют это значение до тех пор, пока не будут принудительно сброшены путем записи в них 1, либо до сброса системы.

11.1.11 SCB->HFSR

Регистр состояния тяжелого отказа

Регистр HFSR содержит сведения о причинах вызова обработчика тяжелого отказа. Особенностью данного регистра является то, что для сброса в 0 его разрядов необходимо записать в них значение 1.

Таблица 493 – Регистр состояния тяжелого отказа

Номер	31	30	29...2	1	0
Доступ	R/W	R/W	U	R/W	R/W
Сброс	0	0	0	0	0
	DEBUGEVT	FORCED	-	VECTTBL	Reserved

DEBUGEVT бит зарезервирован для отладки. При записи в регистр данный бит должен быть равен 0, в противном случае поведение процессора непредсказуемо.

FORCED признак тяжелого отказа, возникшего вследствие эскалации отказа с конфигурируемым уровнем приоритета, который не может быть обработан (запрещен или имеет недостаточно высокий приоритет): 0 – нет, 1 – да.

Если этот бит равен 1, то для определения причины отказа обработчику следует прочитать значения остальных разрядов регистров HFSR.

VECTTBL признак возникновения отказа шины при попытке доступа к таблице векторов исключений: 0 – не было, 1 – было. Эта ошибка всегда вызывает передачу управления на обработчик тяжелого отказа. Если бит равен 1, значение счетчика команд PC, сохраненное в стеке, указывает на инструкцию, выполнение которой было прервано для обработки исключения.

После установки в 1 биты регистра HFSR сохраняют это значение до тех пор, пока не будут принудительно сброшены путем записи в них 1, либо до сброса системы.

11.1.12 SCB->MMFAR

Регистр адреса отказа доступа к памяти

Регистр MMFAR содержит адрес, при обращении по которому возникла ошибка управления памятью.

Таблица 494 – Регистр адреса отказа доступа к памяти

Номер	31...0
Доступ	RO
Сброс	0
	ADDRESS

ADDRESS если бит MMARVALID регистра MMFSR равен 1, это поле содержит адрес, при обращении по которому возникла ошибка управления памятью. В случае ошибки доступа по невыровненному адресу поле содержит фактическое значение адреса, вызвавшего отказ.

Учитывая, что одна единственная операция чтения или записи может быть разбита процессором на несколько операций доступа по выровненному адресу, в регистре адреса отказа может находиться любое значение в диапазоне адресов, по которым осуществлялась попытка доступа.

Флаги регистра MMFSR содержат информацию о причине отказа, а также сообщают, является ли значение MMFAR корректным. Подробнее см. пункт 9.7.3 «Регистры состояния и адреса отказа».

11.1.13 SCB->BFAR

Регистр адреса отказа доступа к шине

Регистр BFAR содержит адрес, при обращении по которому возникла ошибка доступа к шине.

Таблица 495 – Регистр адреса отказа доступа к шине

Номер	31...0
Доступ	RO
Сброс	0
	ADDRESS

ADDRESS если бит BFARVALID регистра BFSR равен 1, это поле содержит адрес, при обращении по которому возникла ошибка доступа к шине. В случае ошибки доступа по невыровненному адресу поле содержит значение адреса, запрошенного командой процессора, даже если оно не совпадает с адресом, вызвавшем отказ.

Флаги регистра BFSR содержат информацию о причине отказа, а также сообщают, является ли значение BFAR корректным. Подробнее см. пункт 9.7.3 «Регистры состояния и адреса отказа».

11.1.14 Рекомендации по программированию блока управления системой

Необходимо убедиться, что программа использует для обращения к регистрам блока управления системой доступ по корректно выровненным адресам. Обращение ко всем регистрам, за исключением CFSR и SHPR1-SHPR3, должно быть выровнено по границе слова. Регистры CFSR и SHPR1-SHPR3 допускают как побайтный доступ, так и доступ по адресам, выровненным по границе слова или полуслова.

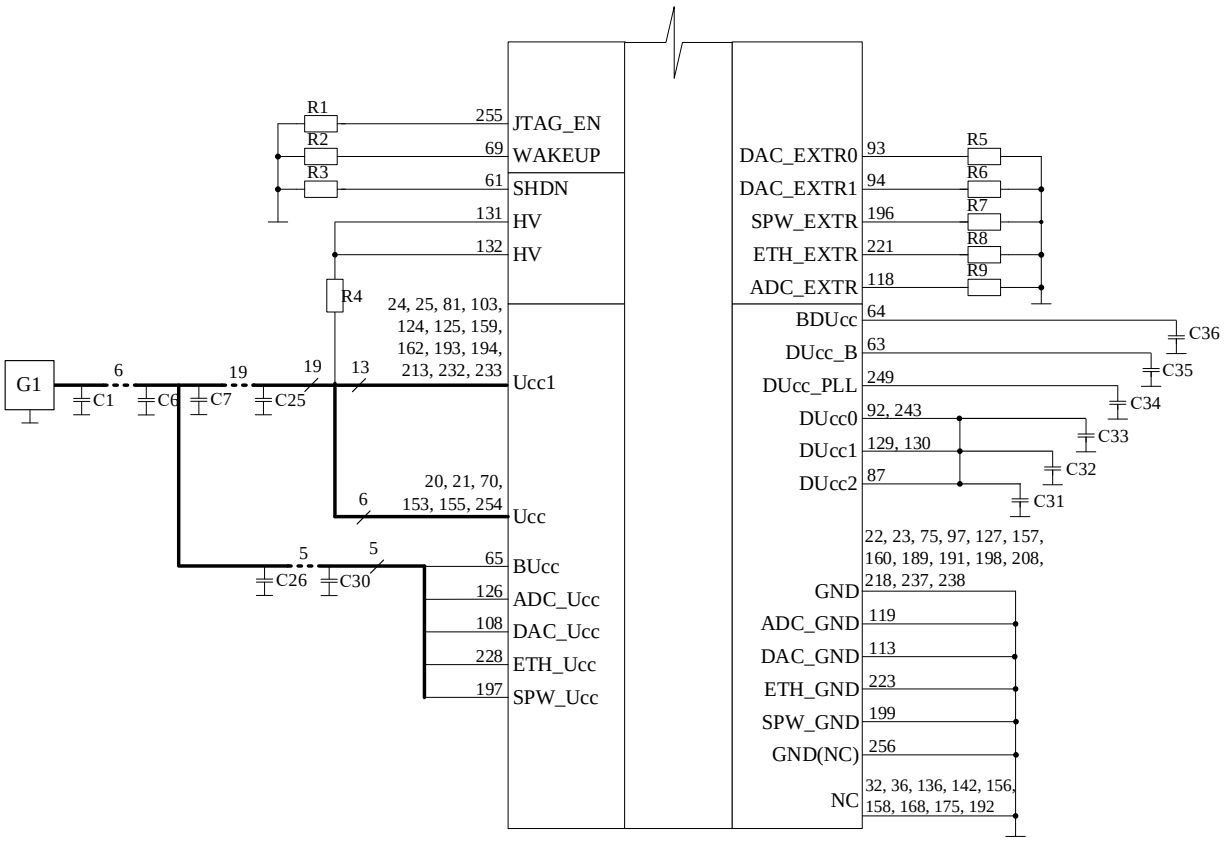
Для того чтобы определить истинный адрес, вызвавший отказ, в обработчике необходимо выполнить следующие действия:

– считать и сохранить значения регистров MMFAR или BFAR;

– проверить значение бита MMARVALID регистра MMFSR, либо бита BFARVALID регистра BFSR. Значения MMFAR или BFAR корректны только в случае, если соответствующие биты равны 1.

Рекомендуется именно такая последовательность операций, так как возникновение исключения с более высоким приоритетом может изменить значения в регистрах MMFAR и BFAR, например, в случае возникновения сбоя в обработчике более высокоприоритетного исключения.

12 Типовая схема включения



C1 – C6	–	конденсаторы емкостью 22 мкФ, номинальным напряжением не менее 10 В. Подключать по одному на каждую группу из 4 выводов: – группа 1: 20, 21, 24, 25; – группа 2: 65, 70, 81, 103; – группа 3: 108, 124, 125, 126; – группа 4: 153, 155, 159, 162; – группа 5: 193, 194, 197, 213; – группа 6: 228, 232, 233, 254;
C7 – C30	–	конденсаторы емкостью 0,1 мкФ, номинальным напряжением не менее 10 В. Подключать по одному на каждый вывод питания;
C31 – C36	–	конденсаторы емкостью 10 мкФ, номинальным напряжением не менее 10 В;
G1	–	источник напряжения питания, от 3,0 до 5,5 В;
R1, R3, R4	–	резисторы сопротивлением 10 кОм ± 10 %;
R2	–	резистор сопротивлением 1 кОм ± 10 %;
R5 – R9	–	резисторы сопротивлением 120 кОм ± 5 %.

Конденсаторы должны располагаться максимально близко к соответствующим выводам питания.

Рисунок 168 – Типовая схема включения микросхемы

13 Основные зависимости

Раздел находится в разработке.

14 Электрические параметры микросхемы

Таблица 496 – Электрические параметры микросхем при приемке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Выходное напряжение высокого уровня, В	U_{OH}	2,4	–	25, 125, – 60
Выходное напряжение низкого уровня, В	U_{OL}	–	0,4	
Напряжение питания формирования сигнала «Сброс» встроенного регулятора, В, – при повышении U_{CC};	U_{POR}	–	3,0	
– при понижении U_{CC}		2,4	–	
Выходное напряжение встроенного регулятора, В, в режиме «Включено»	U_{O_D}	1,62	1,98	
Входной ток высокого уровня, мкА, на выводах SHDN, JTAG_EN	I_{IH1}	10	500	
Входной ток низкого уровня, мкА, на выводах WAKEUP, RSTn0, RSTn	I_{IL1}	– 500	– 10	
Выходной ток низкого уровня в состоянии «Выключено», мкА, на выводах FT_RSTn, FT_WRNn, при U_O = 0 В	I_{OZL}	– 500	– 10	
Входной ток высокого уровня на выводах генератора тактового сигнала, мкА	I_{IH2}	– 100	100	
Входной ток низкого уровня, на выводах генератора тактового сигнала, мкА	I_{IL2}	– 100	100	
Входной ток утечки высокого уровня, мкА, на входах/выходах портов A – E в цифровом режиме; на выводах WAKEUP, RSTn0, RSTn1, при $U_I = U_{CC}$	I_{ILH}	– 10	10	
Выходной ток высокого уровня в состоянии «Выключено», мкА, на выводах FT_RSTn, FT_WRNn, при $U_O = U_{CC}$	I_{OZH}	– 10	10	
Входной ток утечки низкого уровня, мкА, на входах/выходах портов A – E в цифровом режиме; на выводах SHDN, JTAG_EN, при U_I = 0 В	I_{ILL}	– 10	10	
Входной ток, мкА, на входах/выходах портов A – E в аналоговом режиме, при U_I = 2,5 В	I_I	– 500	500	
Статический ток потребления, мА	I_{CC}	–	5	
Динамический ток потребления процессорного ядра, мА, при $U_{CC} = U_{CC1} = U_{CCB}$ = 3,6 В	I_{OCC1}	–	150	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Динамический ток потребления, мА, при $U_{CC} = U_{CC1} = U_{CCB} = 5{,}5$ В	I_{OCC2}	–	750	25, 125, – 60
Выходная частота генератора HSE, МГц, в режиме резонатора – максимальная	f_{O_HSE}	30	–	
– минимальная		–	1	
Максимальная выходная частота генератора HSE, МГц, в режиме генератора	f_{O_HSEmax}	100	–	
Выходная частота генератора LSE, кГц, в режиме резонатора – максимальная	f_{O_LSE}	40	–	
– минимальная		–	20	
Максимальная выходная частота генератора LSE, МГц, в режиме генератора	f_{O_LSEmax}	1	–	
Выходная частота генератора HSI, МГц	f_{O_HSI}	6	10	
Выходная частота генератора LSI, кГц	f_{O_LSI}	20	60	
Выходная частота PLL, МГц – максимальная	f_{O_PLL}	100	–	
– минимальная		–	40	
Коэффициент деления входной частоты, 1	K_{QPLL}	1	16	
Коэффициент деления синтезируемой частоты, 1	K_{NPLL}	4	75	
Параметры АЦП				
Разрядность АЦП, бит	E_{NADC}	12	–	25, 125, – 60
Дифференциальная нелинейность АЦП, ЕМР	E_{DLADC}	– 1	2	
Интегральная нелинейность АЦП, ЕМР	E_{ILADC}	– 3	3	
Ошибка смещения, ЕМР	E_{OFFADC}	– 25	25	
Ошибка усиления, %	$E_{GAINADC}$	– 10	10	
Параметры ЦАП				
Разрядность ЦАП, бит	E_{NDAC}	12	–	25, 125, – 60
Дифференциальная нелинейность ЦАП, ЕМР	E_{DLDAC}	– 1	2	
Интегральная нелинейность ЦАП, ЕМР	E_{ILDAC}	– 10	10	
Ошибка смещения, мВ	E_{OFFDAC}	– 50	50	
Ошибка усиления, %	$E_{GAINDAC}$	– 10	10	
Параметры SpaceWire				
Выходное дифференциальное напряжение, мВ	U_{OD_TXSPW}	250	450	25, 125, – 60
Выходное синфазное напряжение, В	U_{OS_TXSPW}	1,125	1,375	
Входной ток на выводах приемника SpaceWire, мкА	I_{IN_RXSPW}	– 20	20	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Параметры EtherNet				
Выходное дифференциальное напряжение, В				25, 125, – 60
– при U_{CC} = 3,6 В;	U_{OD_TXETH}	2,5	3,5	
– при U_{CC} = (3,0 – 5,5) В		2,0	4,5	
Входной ток на выводах приемника EtherNet, мкА	I_{IN_RXETH}			
– приемник выключен		– 1	1	
– приемник включен		– 2000	2000	

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В.

Коэффициент программируемости микросхем, N_{PR}, не менее 0,7. После программирования ЭТТ не требуется.

15 Предельно-допустимые характеристики микросхемы

Таблица 497 – Предельно-допустимые режимы эксплуатации и предельные режимы микросхем

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение питания, В, – на выводах U_{CC}; – на выводах ADC_U_{CC}; – на выводах DAC_U_{CC}; – на выводах SPW_U_{CC}; – на выводах ETH_U_{CC}	U_{CC}*	3,0	5,5	– 0,3	6,0
Входное напряжение встроенного регулятора, В, на выводах U_{CC}1	U_{CC1}	2,2	U_{CC}	– 0,3	U_{CC} + 0,3
Напряжение питания батарейного домена, В, на выводе BU_{CC}	U_{CCB}	2,0	5,5	– 0,3	6,0
Входное напряжение высокого уровня, В, – на 5 В толерантных выводах: PA[31–0], PB[31–0], PC[5–0], PC[31], PC[30], PD[31-0], PE[31-0]	U_{IH}	2,0	5,5	–	6,0
– на 5 В не толерантных выводах: PC[29–6], JTAG_EN, SHDN		2,0	U_{CC}	–	U_{CC} + 0,3
Входное напряжение низкого уровня, В, – на выводах: PA[31–0], PB[31–0], PC[31–0], PD[31-0], PE[31-0], JTAG_EN, SHDN, RSTn0, RSTn1, WAKEUP	U_{IL}	0	0,8	– 0,3**	–
Выходной ток высокого уровня, мА	I_{OH}	– 6	–	– 15	–
Выходной ток низкого уровня, мА	I_{OL}	–	6	–	15
Частота следования импульсов АЦП, МГц	f_{C_ADC}	–	11	–	–
Длительность тактового импульса АЦП, нс	t_{W_ADC}	45	–	–	–
Частота следования импульсов CLK, МГц	f_{C_CLK}	–	100	–	–
Длительность тактового импульса CLK, нс	t_{W_CLK}	5	–	–	–
Частота следования импульсов CPU CLK, МГц	f_{C_CPUCLK}	–	64	–	–
Частота следования импульсов тактовых сигналов PLL, МГц	f_{C_PLL}	2	16	–	–
Емкость нагрузки на выводах, пФ	C_L	–	50	–	100

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Параметры генератора тактовых сигналов					
Входное напряжение высокого уровня, В, на входах генератора тактового сигнала в режиме BYPASS: HSE0_OSC_IN, HSE1_OSC_IN, LSE_OSC_IN	U_{IH1}	0,8·U_{CC}	U_{CC}	–	U_{CC} + 0,3
Входное напряжение низкого уровня, В, на входах генератора тактового сигнала в режиме BYPASS: HSE0_OSC_IN, HSE1_OSC_IN, LSE_OSC_IN	U_{IL1}	0	0,2·U_{CC}	– 0,3	–
Амплитуда входного сигнала, В, на входах генератора тактового сигнала в режиме OSC: HSE0_OSC_IN, HSE1_OSC_IN, LSE_OSC_IN	U_{A_OSC}	0,05	U_{CC}	–	–
Частота следования импульсов тактовых сигналов HSE, МГц, – в режиме BYPASS	f_{C_HSE}	–	100	–	–
– в режиме OSC		1	30	–	–
Частота следования импульсов тактовых сигналов LSE, – в режиме BYPASS, МГц	f_{C_LSE}	–	1	–	–
– в режиме OSC, кГц		20	40	–	–
Параметры АЦП					
Напряжение верхней границы внешнего опорного напряжения АЦП, В	U_{ADC_REFp}	2,4	U_{CC}	–	–
Напряжение нижней границы внешнего опорного напряжения АЦП, В	U_{ADC_REFn}	0	U_{CC} – 2,4	–	–
Диапазон напряжения внешнего опорного источника АЦП, В, $U_{REF(ADC)} = U_{ADC_REFp} - U_{ADC_REFn}$	$U_{REF(ADC)}$	2,4	U_{CC}	–	–
Параметры ЦАП					
Напряжение верхней границы внешнего опорного напряжения ЦАП, В	U_{DAC_REFp}	2,4	U_{CC}	–	–
Напряжение нижней границы внешнего опорного напряжения ЦАП, В	U_{DAC_REFn}	0	U_{CC} – 2,4	–	–
Диапазон напряжения внешнего опорного источника ЦАП, В, $U_{REF(DAC)} = U_{DAC_REFp} - U_{DAC_REFn}$	$U_{REF(DAC)}$	2,4	U_{CC}	–	–
Сопротивление нагрузки ЦАП, кОм	$R_{LOADDAC}$	10	–	–	–
Емкость нагрузки ЦАП, пФ	$C_{LOADDAC}$	–	100	–	–

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Параметры SpaceWire					
Входное напряжение на выводах приемника SpaceWire, В	U_{IN_RXSPW}	0	2,4	– 0,3	U_{CC} + 0,3
Входное напряжение дифференциальное, В	U_{ID_SPW}	0,1	2,4	–	–
Входное напряжение синфазное, В	U_{IC_SPW}	0,05	2,35	–	–
Параметры EtherNet					
Входное напряжение на выводах приемника EtherNet, В	U_{IN_RXETH}	0	U_{CC}	– 0,3	U_{CC} + 0,3
Входное напряжение дифференциальное, мВ	U_{ID_ETH}	500	U_{CC}	–	–

* Напряжения питания на выводах ADC_U_{CC}, DAC_U_{CC}, SPW_U_{CC}, ETH_U_{CC} должны отличаться от напряжения питания на выводах U_{CC} не более, чем на |±0,2| В.

** Допускается подача отрицательного напряжения без ограничения по уровню при ограничении входного тока не более 1 мА.

Примечание – Не допускается одновременное задание более одного предельного режима

16 Справочные данные

Значение собственной резонансной частоты не менее 5,3 кГц.

Тепловое сопротивление кристалл-корпус 2,8 °С/Вт.

Справочные параметры микросхемы приведены в таблице 498.

Предельная температура p-n перехода кристалла 150 °С.

Максимальное значение входной и выходной емкости вывода микросхемы в корпусе 4244.256-4 на частотах менее 1 МГц не превышает 15,4 пФ.

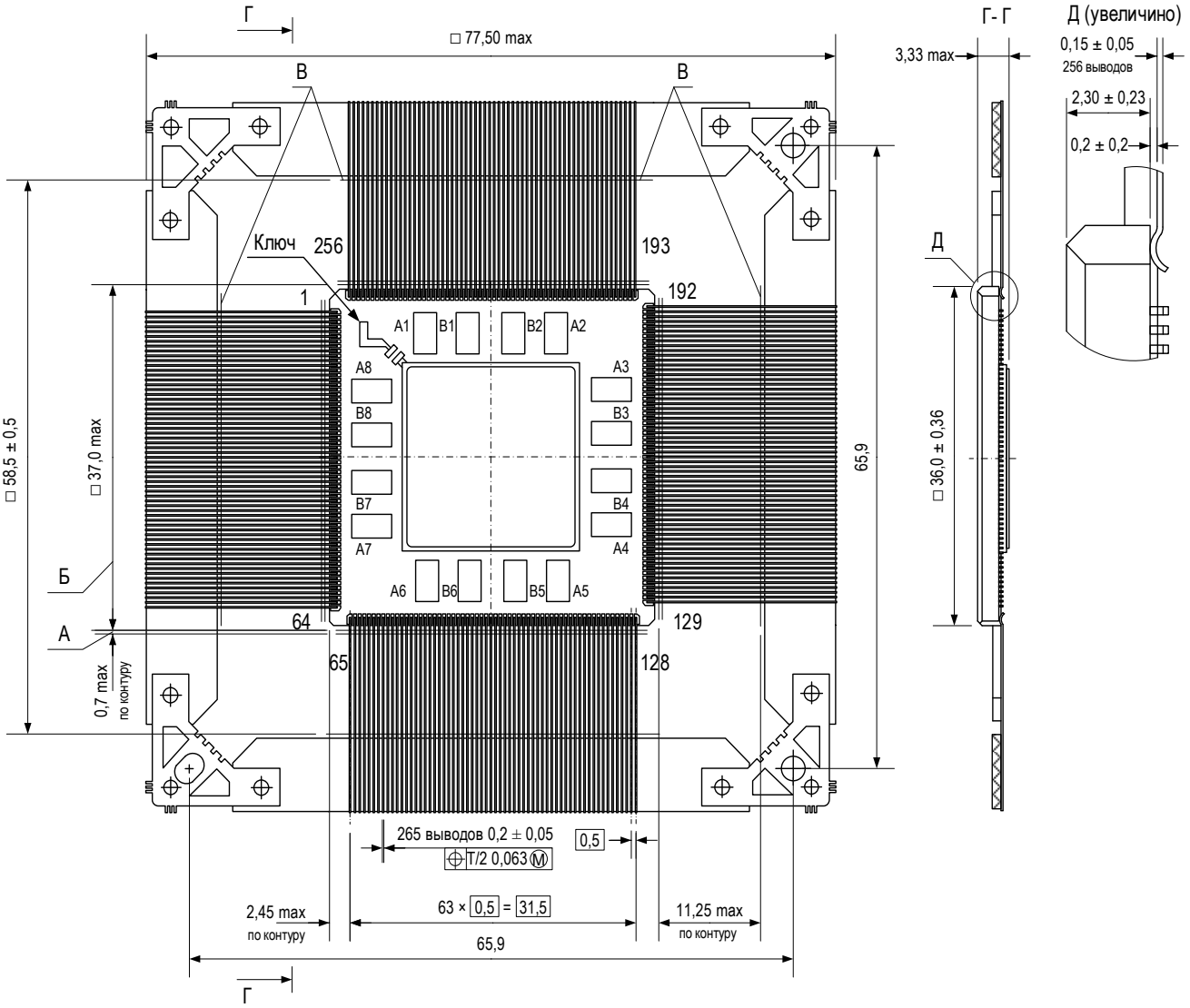
Таблица 498 – Справочные параметры микросхемы

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Питание				
Напряжение питания U_{CC} разрешения работы блока сброса POVR, В	U_{POVR_max}	5,5	–	25, 125, – 60
Напряжение питания U_{CC} запрещения работы блока сброса POVR, В	U_{POVR_min}	–	5,7	
Напряжение питания цифровой части, В, на выводах $DU_{CC}0$, $DU_{CC}1$, $DU_{CC}2$, DU_{CC}_PLL, DU_{CC}_B, SHDN = 1	U_{CCD}	1,70	1,98	
Статический ток потребления в режиме пониженного энергопотребления, мА	I_{CCS}	–	10	
Ток потребления по выводу BU_{CC}, мА	I_{CC1}	–	20	
Ток потребления по выводам $ETHU_{CC}$, мА	I_{CC2}	–	20	
Ток потребления по выводам $ETHU_{CC}$, мА, в режиме замыкания	I_{CC3}	–	150	
Ток потребления по выводам $SPWU_{CC}$, мА	I_{CC4}	–	40	
Ток потребления по выводам $SPWU_{CC}$, мА, в режиме замыкания	I_{CC5}	–	40	
Ток потребления по выводам $DACU_{CC}$, мА	I_{CC6}	–	10	
Ток потребления по выводам DACREF, мА	I_{CC7}	–	10	
Ток потребления по выводам $ADCU_{CC}$, мА	I_{CC8}	–	10	
Ток потребления по выводам ADCREF, мА	I_{CC9}	–	10	
Ток потребления по выводам HV, мА, в режиме программирования	I_{CCHV}	–	10	
Время задержки сигнала сброса от момента появления питания, мс	$T_{PORRSTn}$	2	8	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Порты ввода-вывода				
Входной гистерезис переключения приемника, В	U_{HES}	0,15	0,3	25, 125, – 60
Выходной ток высокого уровня срабатывания схемы защиты от перегрузки, мА	I_{HLOAD}	– 100	– 10	
Выходной ток низкого уровня срабатывания схемы защиты от перегрузки, мА	I_{ILOAD}	10	100	
Сопротивление внутреннего резистора подтяжки, кОм	R_{PULL}	40	60	
Тактовые генераторы				
Время запуска генератора HSE, нс	t_{HSEON}	–	100+512 тактов	25, 125, – 60
Длительность фильтруемых иголок в HSE, нс	$t_{HSEFILTER}$	–	3	
Время запуска генератора LSE, нс, в режиме резонатора	t_{LSEON}	–	100+512 тактов	
Длительность фильтруемых иголок в LSE, нс	$t_{LSEFILTER}$	–	3	
Время запуска PLL, мкс	t_{PLLON}	–	60	
Емкость нагрузки, пФ, на выводах генератора тактового сигнала в режиме OSC: HSE0_OSC_IN и HSE0_OSC_OUT; HSE1_OSC_IN и HSE1_OSC_OUT	C_{L_HSE}	15	30	
Емкость нагрузки, пФ, на выводах генератора тактового сигнала в режиме OSC: LSE_OSC_IN и LSE_OSC_OUT	C_{L_LSE}	40	60	
АЦП				
Число тактов преобразования	N	43	43	25, 125, – 60
Время включения АЦП, мкс	t_{ADCON}	–	10	
Опорное напряжение 2,5 В, В	U_{REF_ADC}	2,3	2,7	
Время зарядки от ref_2,5 В, мкс	t_{REF}	–	2	
Частота преобразования, выборок в секунду	f_{CONW}	–	$0{,}25 \cdot 10^6$	
Входное напряжение АЦП, В, при опоре от вывода ADC_U_{CC}	U_{IN_ADC}	0	U_{CC_ADC}	
Входное напряжение АЦП, В, при опоре от выводов ADCREF	U_{IN_ADC1}	U_{ADC_REFn}	U_{ADC_REFp}	
Входное напряжение АЦП, В, при опоре от источника опорного напряжения 2,4 В	U_{IN_ADC24}	0	2,2	
Сопротивление внешнего токозадающего резистора на выводе ADC_EXTR, кОм	$R_{L_ADC_EXTR}$	114	126	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
SpaceWire				
Ток короткого замыкания, мА	I_{OS_TX}	5,9	20,0	25, 125, – 60
Дифференциальный ток короткого замыкания, мА	I_{OSD_TX}	4,1	8,5	
Входное сопротивление дифференциальное на выводах приемника SpaceWire, кОм	$R_{RX_IN_SPW}$	500	1000	
Выходное сопротивление дифференциальное на выводах приемника SpaceWire, Ом	$R_{TX_OUT_SPW}$	98	224	
Время нарастания/спада выходного сигнала, нс, от 20 % до 80 %	t_{RISE_TX} t_{FALL_TX}	–	0,8	
Скважность выходного сигнала, %	$СК_{_TX}$	45,0	55,0	
Дифференциальное входное напряжение перехода в «1», мВ	U_{TH_RX}	-	100	
Дифференциальное входное напряжение перехода в «0», мВ	U_{TL_RX}	– 100	–	
Ethernet				
Изменение выходного дифференциального напряжения, В, на выводах TXP и TXM	ΔU_{OD_TX}	–	0,8	25, 125, – 60
Выходное синфазное напряжение передатчика, В, выводы TXP и TXM	U_{OS_TX}	1,2	3,5	
Изменение выходного синфазного напряжение передатчика, В	ΔU_{OS_TX}	–	2,2	
Дифференциальное пороговое напряжение приемника, В	U_{OD}	– 0,15	0,15	
Гистерезис приемника, мВ	U_h	5	15	
Время включения передатчика по сигналу TXEN, нс	$t_{TXEN\text{-}TX}$	–	12	
Время задержки передатчика от сигнала TXD, нс	$t_{TXD\text{-}TX}$	–	10	
Длительность фронта TXP и TXN, нс	t_{fTXP} t_{fTXN}	–	8	
Источник опорного напряжения				
Сопротивление внешнего токозадающего резистора на выводе EXTR, кОм	R_{L_EXTR}	118,8	121,2	25, 125, – 60

17 Габаритный чертеж микросхемы



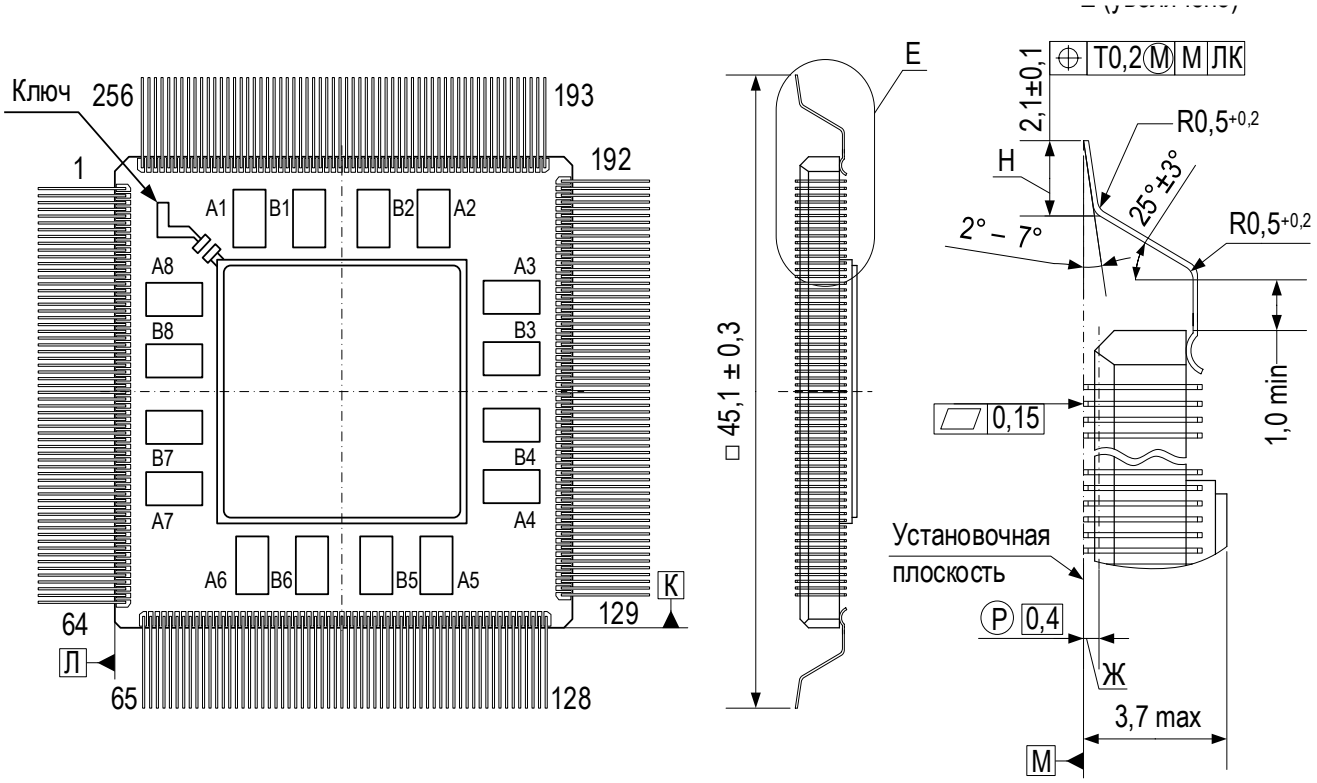
1 А – длина вывода, в пределах которой установлено смещение осей выводов от номинального расположения;

2 Б – Ширина зоны, которая включает действительную шиниру микросхемы и часть выводов, непригодные для монтажа;

3 В – рекомендуемые линии обрубки изолирующей рамки;

4 Нумерация вывододв и металлизированных площадок на корпусе показана условно.

Рисунок 169 – Микросхема в корпусе 4244.256-3



Остальное см. рисунок 169

1 Ж – расстояние от установленной плоскости до плоскости, пересекающей вывод на длине, пригодной для монтажа;

2 Н – длина вывода, пригодная для монтажа.

Рисунок 170 – Микросхема в корпусе 4244.256-3 с формованными выводами

18 Инструкция по распаковыванию микросхем с формованными выводами

Для распаковывания микросхем с формованными выводами необходимо выполнить следующие действия:

1 Открыть картонную коробку с изделиями, извлечь верхние прокладки, развернуть фольгу (см. рисунок 171).

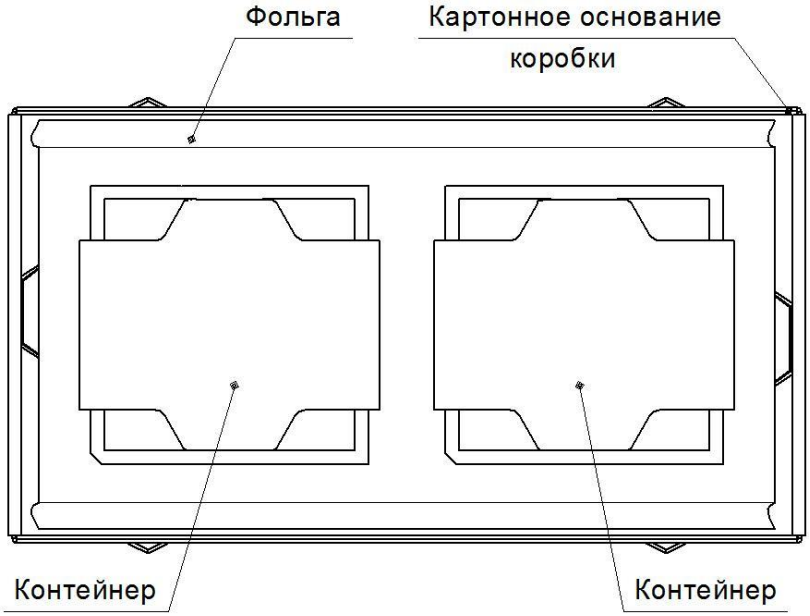


Рисунок 171 – Картонная коробка с изделиями

2 Достать контейнер, положить на горизонтальную поверхность. Снять защелку контейнера, сдвинув ее в сторону (см. рисунок 172). Необходимо проконтролировать, чтобы контейнер не упал и не раскрылся.

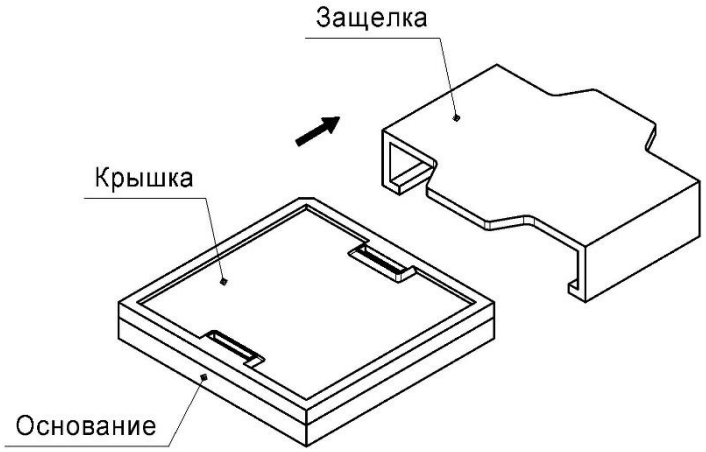


Рисунок 172 – Схема сдвига защелки контейнера

3 Открыть крышку контейнера, удерживая основание на горизонтальной плоскости, согласно рисунку 173.

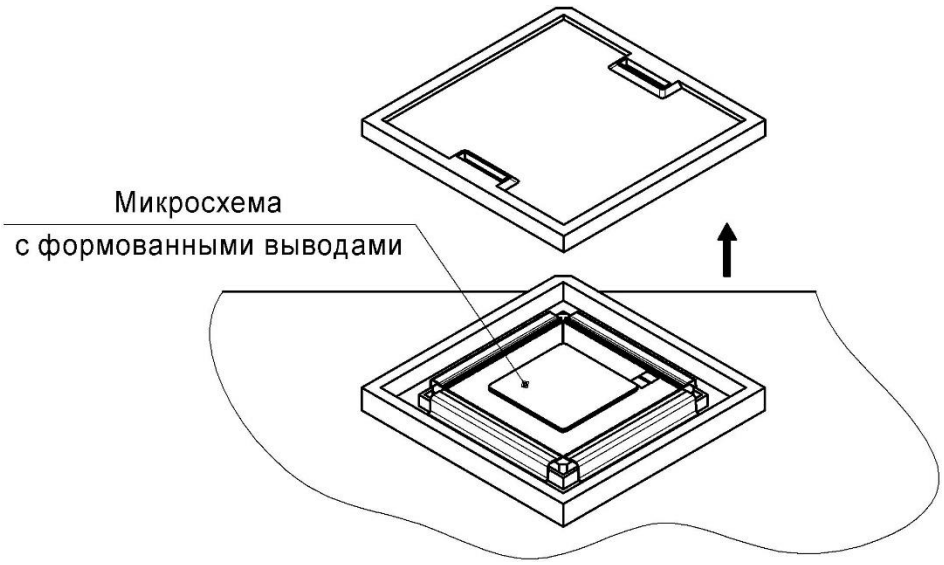


Рисунок 173 – Схема открытия крышки контейнера

4 Извлечь микросхему с формованными выводами вакуумным пинцетом, не задевая стенки основания, и переложить в технологическую тару, исключающую деформацию выводов.

5 Не допускается повторная упаковка микросхемы с формованными выводами в контейнер из-за риска неправильной установки и деформации выводов.

19 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон, °С
К1986ВЕ8Т	12164	4244.256-3	от – 60 до 125
		4244.256-3 с формованными выводами*	
* Необходимость поставки микросхем с формованными выводами указывается в договоре на поставку			

Условное обозначение микросхем при заказе в договоре на поставку и в конструкторской документации другой продукции должно состоять из:

– наименование изделия – микросхема;
– обозначения типа (типономинала);
– обозначения технических условий ТСКЯ.431000.002ТУ;
– обозначения спецификации ТСКЯ.431296.048СП.

Пример обозначения микросхем:
Микросхема К1986ВЕ8Т – ТСКЯ.431000.002ТУ, ТСКЯ.431296.048СП.

Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	27.09.2024	0.1.0	Введена впервые	
2	29.01.2025	0.2.0	Раздел 5 – дополнен; Подпункты 6.4.11.2, 6.4.11.3, 6.4.11.5 – 6.4.11.7 дополнены; Таблица 10 – дополнена; Подпункт 6.4.11.12 – перенесены ошибки; Подраздел 7.5 – дополнен; Таблица 81, таблица 291, таблица 330, таблица 337, таблица 381, таблица 407, таблица 433– дополнены; Подпункт 7.5.1.3 – зарезервированы биты 1 и 2 в таблице регистра STATUS; Таблица 84, таблица 87, таблица 90, таблица 91 – исправлены; Подраздел 7.8 – дополнен; Подпункт 7.18.4.4 – исправлен; Пункт 7.21.9 – дополнен; Таблица 431 – добавлено описание поля REFSEL[5:0]; Подраздел 9.3 – дополнен	35 55, 56 57 58 154 151, 363, 407, 431, 498, 538, 563 157 155, 157, 159, 159 216 464 534 560 594
3	12.02.2025	0.2.1	Заменено "WDT" (WDTCNTR, MDR_WDT) на "WDG" по всей СП	
4	13.05.2025	0.3.0	Исправлены опечатки; Раздел 5 – скорректирован; Подпункт 7.2.3.33 – 7.2.3.39, таблица 57 – 63 – добавлен бит 17 (DIV2); Пункт 7.7.1 – дополнено описание фаз после рисунка 35; Рисунок 34, 35 заменены; Подпункт 7.7.1.3 дополнен; Таблицы 125, 126, 127, 128 – скорректированы; Рисунки 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 – скорректированы; Подпункт 7.7.1.8 – исправлено название подпункта и описание; Пункт 7.21.10 – добавлено описание особенностей формирования прерывания; Пункт 7.22.3 – дополнен	По тексту 35 121 – 127 195 195 197 193, 197, 199, 200 202 – 207 208 535 550
5	11.07.2025	0.4.0	Добавлено тепловое сопротивление; Масса перенесена в основные параметры; Подраздел 6.4 скорректирован;	1 1 47

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
			Подпункты 6.4.7.2, 6.4.9.2, 6.4.11.9 – скорректированы адреса;	50, 52, 57
			Пункт 7.3.1 скорректирован;	130
			Таблица 70 – скорректировано описание бит RTC_CAL[7:0], RTC_EN, RTC_SEL[1:0];	140
			Таблица 73 – скорректировано описание бита RTC_PRL[19:0];	144
			Таблица 74 – скорректировано описание бита RTC_ALRM[31:0];	144
			Таблица 75 – скорректировано описание бит WEC, ALRF, SECF	145