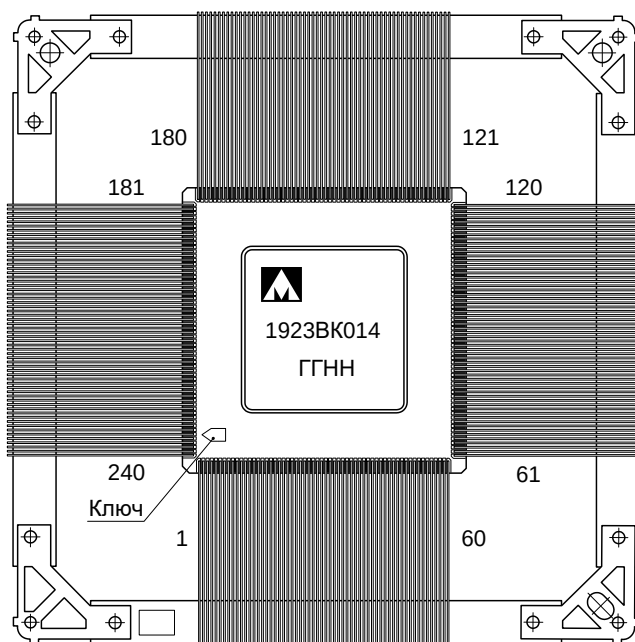




Микросхема контроллера памяти 1923BK014, K1923BK01H4



Основные параметры микросхемы:

- Напряжение источника питания от 3,0 до 5,5 В;
- 32-х разрядная RISC-архитектура ARM Cortex-M0;
- Динамический ток потребления не более 500 мА;
- Встроенная память ОЗУ программ 32 Кбайт;
- Встроенная память ОЗУ данных 32 Кбайт;

- Температурный диапазон:

Обозначение	Диапазон, °С
1923BK014	от – 60 до 125
K1923BK01H4	от 0 до 70

ГГ – год выпуска

НН – неделя выпуска

Тип корпуса:

- 240-выводной металлокерамический корпус 4245.240-6.01;
- микросхемы K1923BK01H4 поставляются в бескорпусном исполнении.

Общее описание и область применения микросхемы

Микросхемы интегральные 1923BK014, K1923BK01H4 (далее – микросхемы) представляют собой специализированную микросхему контроллера внешней памяти и периферии со встроенным вспомогательным микроконтроллерным ядром. Микросхемы предназначены для использования в системах специального назначения для измерения, контроля, управления и диагностики. Микросхема может использоваться в двух основных режимах, либо одновременно в обоих.

Первый режим работы – микроконтроллер. В данном режиме микросхема работает под управлением встроенного микропроцессорного ядра.

Второй режим работы – контроллер периферии. В данном режиме микросхема подключается по параллельной шине к ведущему процессору и позволяет использовать свои встроенные периферийные блоки как периферию для ведущего процессора. В данном режиме может также использоваться и встроенное процессорное ядро.

Микросхема содержит стартовое ПЗУ, позволяющее ему начать работать в одном из режимов. Исполнение пользовательской программы возможно сразу из внешней памяти или из внутренней ОЗУ памяти программ после первоначальной инициализации. Внутренняя память программ при начале работы должна быть проинициализирована пользовательской программой. Данная программа может быть загружена из внешней памяти по различным последовательным интерфейсам (SPI, UART и т. п.).

Благодаря мощной функциональности внешней шины микросхема может работать с различными конфигурациями внешней памяти. Разрядность шины данных может варьироваться от 8 до 48 бит. При этом данные могут быть защищены с помощью различных корректирующих кодов ECC (Error Code Correction). ECC позволяет исправление одиночной и фиксацию двойной SEC-DED (дополнительные 8 разрядов) или исправление двойной и фиксацию тройной ошибки DEC-TED (дополнительные 16 разрядов). Дополнительные разряды ECC могут располагаться как в дополнительных разрядах шины данных (параллельная организация ECC), так и в более старших адресах самой памяти (последовательная организация ECC).

В качестве внешней памяти могут выступать как обычные микросхемы ОЗУ (например, 1645PY5), однократно-программируемые ПЗУ (например, 1645PT2 или 1645PT3), перепрограммируемые типа Flash (например, 1636PP4) или любые другие микросхемы памяти и периферии, имеющие традиционный интерфейс статической памяти (например, FRAM или MRAM).

Подключение ведущего процессора осуществляется через специальный порт CPUIF. Данный порт реализует интерфейс статического ОЗУ. В данном режиме шина данных может быть 8, 16 или 32 бита. Разрядность шины адреса может варьироваться в зависимости от необходимого для реализации адресного пространства. Запись данных задается сигналом WE, чтение сигналом OE. Длительность транзакции должна выбираться такой, чтобы микросхема успела ее выполнить. Длительности транзакций определяются в зависимости от заданной тактовой частоты микросхемы и режимов работы. Для ускорения выполнения транзакций может использоваться флаг READY, который выставляет микросхема после завершения транзакции, и ведущий процессор по данному сигналу может досрочно завершить транзакцию обращения. Ведущий процессор имеет доступ к периферийным блокам микросхемы, такой же доступ, как и встроенное процессорное ядро.

С использованием микросхемы могут быть построены различные системы, примеры и основные параметры приведены ниже.

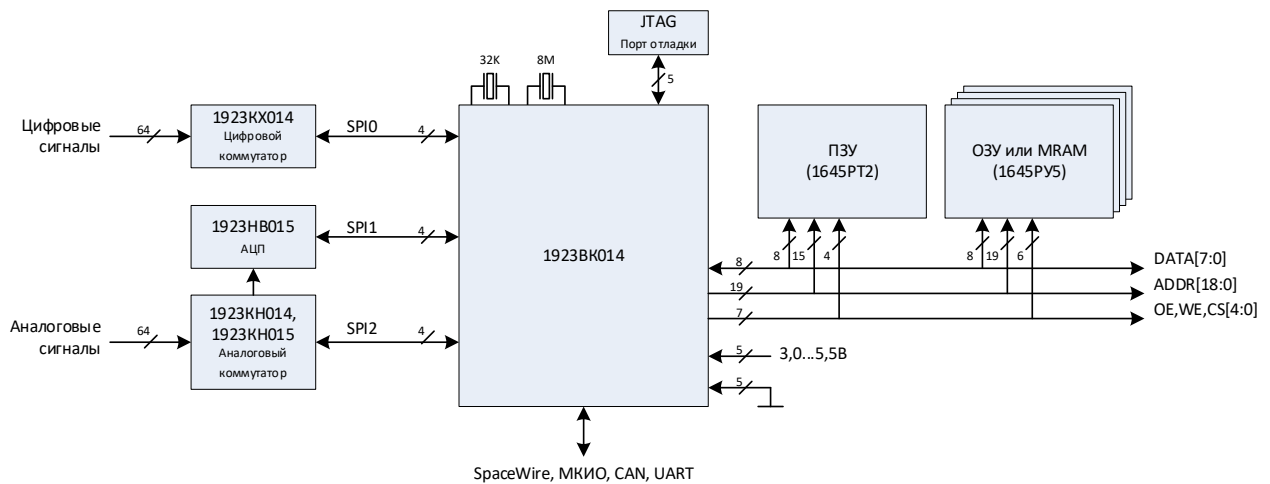


Рисунок 1 – Система сбора и хранения телеметрической информации

Микросхема работает под управлением встроенного микропроцессорного ядра. Запуск программы для ядра осуществляется из внешней ПЗУ по параллельному порту. Микросхема по последовательным интерфейсам SPI осуществляет настройку и управляет приемом и оцифровкой внешних цифровых и аналоговых сигналов. Принятые данные сохраняются в логге телеметрической информации во внешнем ОЗУ. Данные снабжаются контрольной суммой CRC для подтверждения их достоверности и защищаются с помощью ECC для исправления возможных одиночных сбоев. Доступ к подсистеме сбора телеметрической информации с более высокостоящего процессора или устройства съема и передачи может осуществляться по интерфейсам SpaceWire, МКИО, CAN, UART и другим. Для приема телеметрической информации также могут использоваться такие интерфейсы как МКИО или CAN. В качестве памяти накопления информации может выступать энергонезависимые памяти MRAM или Flash.

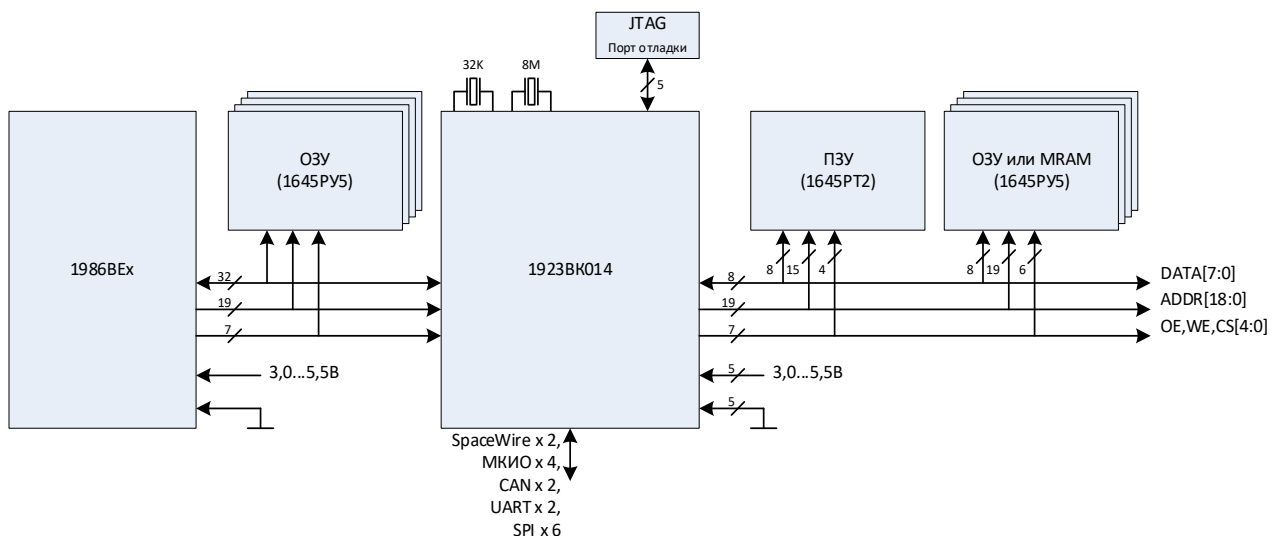


Рисунок 2 – Система расширения функциональности ведущего процессора

Микросхема работает под управлением внешнего процессора, подключение процессора осуществляется по параллельной шине с разрядность 32 бита. Помимо микросхемы на шине ведущего процессора может быть реализована быстрая внешняя память большого объема. На ведущий процессор возложены сложные вычислительные задачи, и для обработки интерфейсов может использоваться ядро в составе микросхемы. Периферийные блоки микросхемы также доступны ведущему процессору и могут полноценно использоваться.

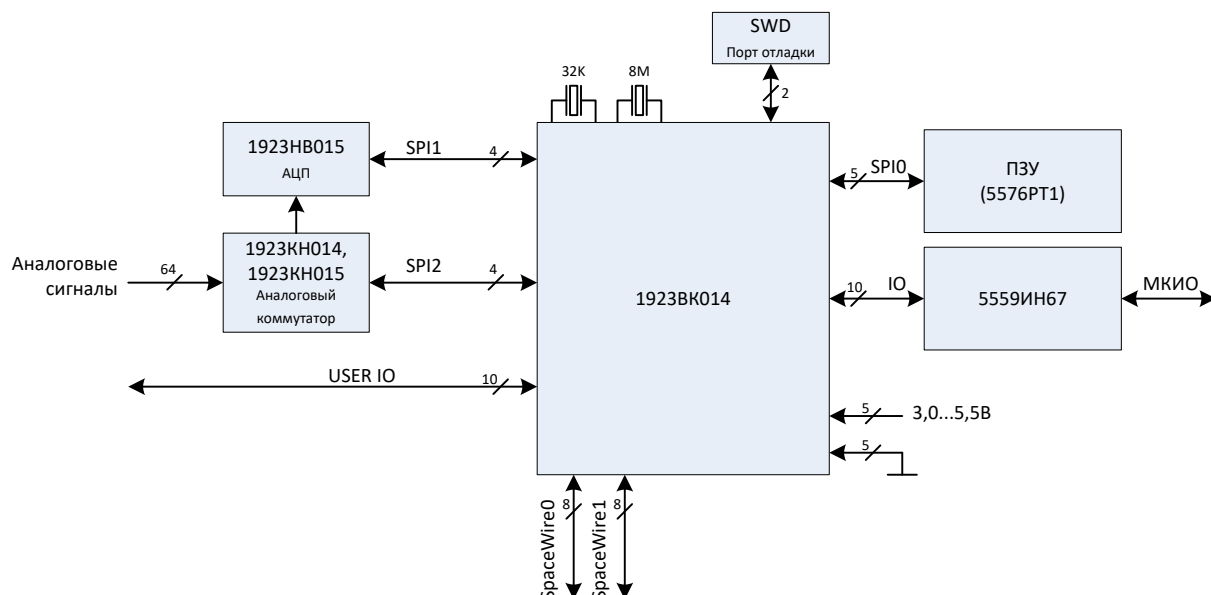


Рисунок 3 – Система микроконтроллерного управления

Микросхема выступает в роли самостоятельного микроконтроллера управления. Пользовательская программа загружается по последовательному порту из внешнего ПЗУ (однократно программируемая ПЗУ 5576PT1). При отладке системы может применяться внешний перепрограммируемый аналог 5576PC1 на базе Flash-памяти. Микроконтроллер может выполнять различные задачи по управлению и обработке данных. Связь с другими элементами системы может осуществляться по различным интерфейсам, таким как МКИО, SpaceWire, SPI, UART и т.п.

Основные характеристики

Ядро:

- ядро ARM Cortex-M0;
- максимальная тактовая частота до 50 МГц;
- блок отладки с поддержкой интерфейсов SWD.

Питание:

- основное питание от 3,0 до 5,5 В;
- встроенные регуляторы для питания ядра;
- аппаратный детектор снижения и превышения допустимого уровня питания;

Тактовые частоты:

- 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 для часов реального времени;
- блока PLL умножения тактовых частот до 150 МГц;
- блоки аппаратной защиты от снижения/увеличения частоты тактирования.

Память:

- 32 Кбайт ОЗУ памяти данных с ECC;
- 32 Кбайт ОЗУ памяти программ с ECC;
- контроллер внешней системной шины с последовательной/параллельной организацией с ECC;
- контроллер внешней шины для подключения к ведущему процессору.

Периферия:

- два контроллера SpaceWire;
- два встроенных приемопередатчика SpaceWirePHY до 50 Мбит/с;
- четыре контроллера МКПД в режимах КШ, ОУ, Монитор;
- стандартная периферия: 2xCAN, 6xSSP, 2xUART, 1xDMA, 4xTimer;
- часы реального времени RTC;
- блок сторожевого таймера WDG;
- блок вычисления ECC по произвольной матрице;
- блок вычисления CRC по произвольному полиному;
- блок приемника и передатчика телеметрической информации по стандарту CCSDS;
- до 160 выводов портов общего назначения.

Стойкость:

- технология 0,18 мкм кремний на изоляторе (КНИ);
- методы аппаратной защиты от одиночных сбоев;
- различные механизмы самодиагностики и защиты от сбоев и отказов.

Содержание

1	Структурная блок-схема микросхемы.....	11
2	Условное графическое обозначение	12
3	Описание выводов	13
4	Указания по применению и эксплуатации	38
5	Архитектура контроллера	39
5.1	Питание	39
5.1.1	Домены питания	39
5.1.2	Сброс при включении питания	39
5.1.3	Схема защиты от перенапряжения POVR.....	40
5.1.4	Перекрытие рабочих диапазонов питания схемами POR и POVR.....	40
5.1.5	Блок монитора основного питания (Ucc).....	41
5.1.6	Управление питанием доменов DUcc.....	41
5.1.7	Блок монитора тока потребления доменов DUcc	41
5.2	Тактовые частоты	41
5.2.1	Встроенный низкоскоростной генератор LSI.....	42
5.2.2	Встроенный высокоскоростной генератор HSI	42
5.2.3	Внешний высокоскоростной генератор HSE0 и HSE1	42
5.2.4	Внешний низкоскоростной часовой генератор LSE	43
5.2.5	Блок умножения тактовой частоты PLL	43
5.2.6	Монитор частоты внешних генераторов и PLL (CLKCNTR)	43
5.3	Сигналы Сброса	44
5.3.1	Сигналы внешнего сброса nRESET0 и nRESET1	44
5.3.2	Сигнал сброса по питанию UccRESET	44
5.3.3	Сигнал сброса по питанию DUccRESET.....	44
5.3.4	Сигнал программного запроса сброса SYS_RESET_REQ.....	45
5.3.5	Сигналы аварийного сброса FT_RESET	45
5.3.6	Сигналы сброса сторожевого таймера IWDG_RESET	45
5.3.7	Сигнал сброса по перенапряжению POVR.....	45
5.3.8	Сигнал сброса отладочного интерфейса TRSTn.....	45
5.4	Помехозащищенное кодирование.....	45
5.4.1	ЕСС кодирование для ПЗУ	46
5.4.2	ЕСС кодирование для RAM.....	48
5.4.3	ЕСС кодирование для внешней шины.....	48
5.4.4	ЕСС кодирование для периферии.....	49
5.4.5	Н-матрица ЕСС по коду Хемминга (72, 64) и (78,64).....	49
5.4.6	Н-матрица ЕСС по коду Хемминга (7, 4)	49
5.4.7	Функция вычисления ЕСС для записи в память	49
5.5	Режимы работы микроконтроллера	51
5.5.1	Режим EXTBUS_8_ECC+JA.....	53
5.5.2	Режим EXTBUS_8_ECC+JB.....	53
5.5.3	Режим EXTBUS_CFG+JA.....	53
5.5.4	Режим EXTBUS_CFG+JB.....	54
5.5.5	Режим SPI0+JB.....	54
5.5.6	Режим SPI1+JA.....	56
5.5.7	Режим SPI2+JB.....	56
5.5.8	Режим SPI3+JA.....	58
5.5.9	Режим UART0+JA.....	58
5.5.10	Режим UART0+JB.....	62
5.5.11	Режим TEST_MODE+JB.....	62
5.5.12	Режим WAIT_BOOT_JA	63

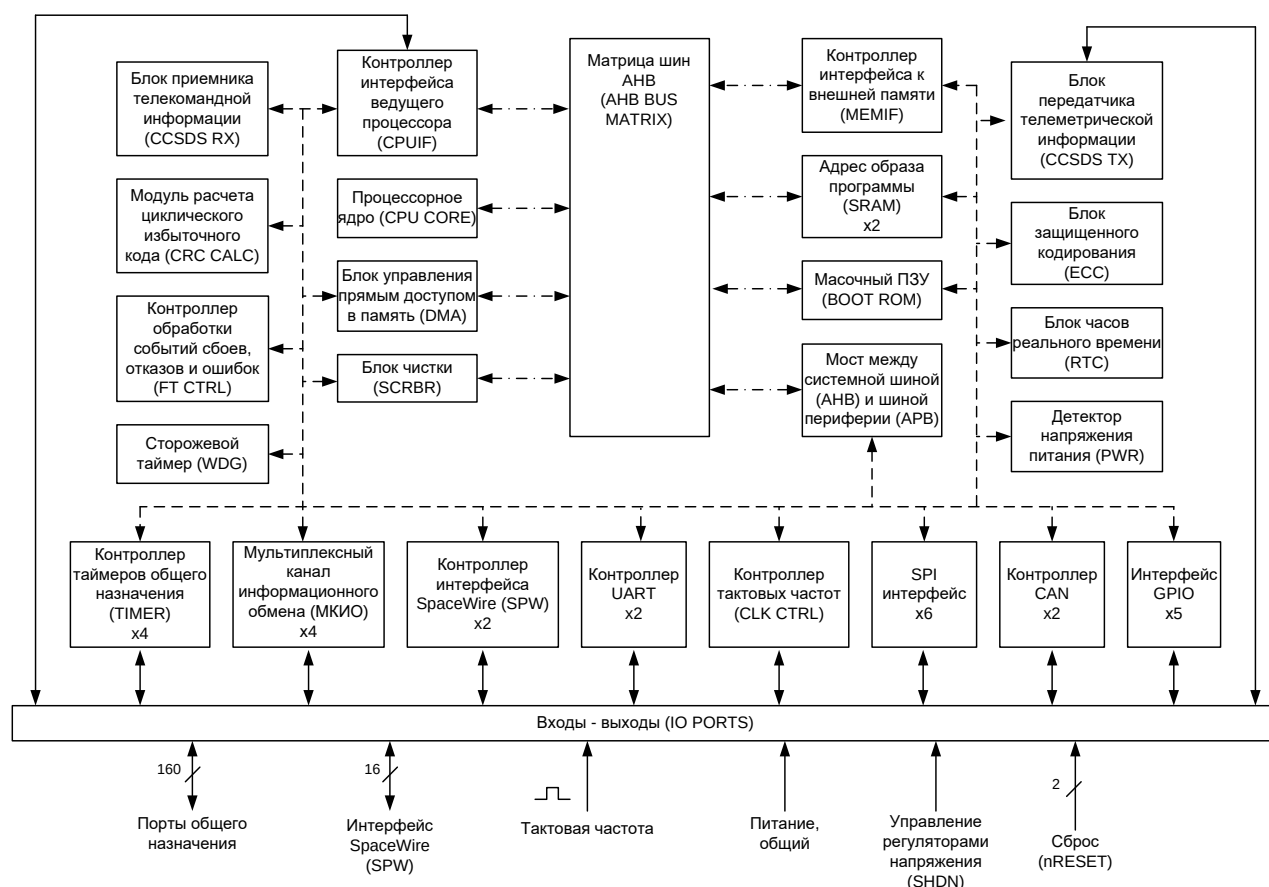
5.5.13	Режим WAIT_BOOT_JB	63
5.5.14	Статус загрузчика.....	63
5.6	Организация памяти.....	64
5.6.1	Распределение областей памяти	66
5.6.2	Распределение областей памяти для регистров периферии.....	67
5.6.3	Организация кэш-памяти.....	69
5.6.4	Организация BOOTROM-памяти.....	70
5.6.5	Организация RAM-памяти	70
5.6.6	Организация доступа к внешней шине MEMIF	71
5.7	Организация доступа DMA.....	80
5.8	Организация доступа со стороны интерфейса CPUIF.....	80
5.8.1	Формирование адреса транзакции	82
5.8.2	Описание регистров моста	83
5.9	Организация доступа к периферии	83
5.10	Периферийные блоки.....	84
5.10.1	Контроллер тактовых частот (CLKCNTR).....	84
5.10.2	Контроллер PWR_BKP.....	88
5.10.3	Контроллер детектора напряжения питания (PWRCNTR)	88
5.10.4	Контроллер обработки событий сбоя, отказов и ошибок (FTCNTR) ..	89
5.10.5	Контроллер сторожевого таймера (WDTCNTR).....	89
5.10.6	Контроллер внешней шины (MEMIF)	89
5.10.7	Контроллер Кэш-памяти (CACHECNTR)	89
5.10.8	Контроллер ПЗУ (ROMCNTR)	89
5.10.9	Контроллер ОЗУ (RAMCCNTR и RAMDCNTR).....	89
5.10.10	Контроллер DMA (DMACNTR)	89
5.10.11	Контроллер портов ввода-вывода (PORTx)	89
5.10.12	Контроллер SpaceWire (SpaceWire)	90
5.10.13	Контроллер таймеров общего назначения (TIMER32x).....	90
5.10.14	Контроллер CAN (CANx)	90
5.10.15	Контроллер SSP (SSPx).....	90
5.10.16	Контроллер UART (UARTx).....	90
5.10.17	Контроллер интерфейса МКПД (MILx).....	90
5.11	Распределение векторов прерываний	481
5.12	Интерфейс отладки	90
5.13	Тестовые режимы работы.....	91
5.13.1	Режим тестирования ОЗУ.....	92
5.13.2	Режим тестирования Масочного ПЗУ (BOOTROM)	93
5.13.3	Режим тестирования RAM_CAN	94
5.13.4	Режим тестирования RAM_MIL	95
5.13.5	Режим тестирования генераторов и PLL.....	96
5.13.6	Режим тестирования приемопередатчиков SPW	97
5.13.7	Режим тестирования PMU (VoltageDetect)	98
6	Программная модель микроконтроллера.....	99
6.1	Контроллер тактовых частот (CLKCNTR)	99
6.1.1	Описание регистров.....	99
6.2	Контроллер батарейного домена (BKPCNTR).....	131
6.2.1	Часы реального времени.....	131
6.2.2	Регистры батарейного домена.....	132
6.2.3	Описание регистров.....	132
6.3	Контроллер детектора напряжения питания (PWR_CNTR).....	145
6.3.1	Описание регистров.....	145
6.4	Контроллер Кэш-памяти (CACHECNTR).....	151

6.4.1	Описание регистров.....	151
6.5	Контроллер обработки событий отказов, сбоев и ошибок (FT_CNTR)	156
6.5.1	Описание регистров.....	156
6.6	Контроллер сторожевых таймеров (WDTCNTR)	188
6.6.1	Описание регистров.....	189
6.7	Контроллер ПЗУ (ROMCNTR)	191
6.7.1	Описание регистров.....	191
6.8	Контроллер ОЗУ (RAMC RAMD)	196
6.8.1	Описание регистров.....	196
6.9	Контроллер DMA (DMACNTR)	200
6.9.1	Основные характеристики контроллера DMA.....	200
6.9.2	Термины и определения.....	200
6.9.3	Функциональное описание	202
6.9.4	Управление DMA.....	204
6.9.5	Структура управляющих данных канала	224
6.9.6	Описание регистров контроллера DMA.....	233
6.10	Контроллер портов ввода-вывода (PORTx).....	251
6.10.1	Описание регистров.....	252
6.11	Контроллер внешней шины	271
6.11.1	Описание регистров.....	271
6.12	Контроллер SpaceWire (SpaceWire)	278
6.12.1	Описание регистров контроллера интерфейса SpaceWire	278
6.13	Контроллер таймеров общего назначения (TIMER32x).....	289
6.13.1	Функционирование	289
6.13.2	Инициализация таймера.....	290
6.13.3	Режим таймера.....	291
6.13.4	Режимы счета	291
6.13.5	Источник событий для счета	294
6.13.6	Режим захвата.....	301
6.13.7	Режим ШИМ.....	302
6.13.8	Примеры	305
6.13.9	Описание регистров.....	307
6.14	Контроллер CAN (CANx)	321
6.14.1	Режимы работы.....	322
6.14.2	Типы пакетов сообщений	323
6.14.3	Структура пакета данных (DataFrame)	324
6.14.4	Структура пакета удаленного запроса данных (RemoteFrame).....	327
6.14.5	Арбитраж на шине.....	327
6.14.6	Инициализация.....	329
6.14.7	Передача сообщений.....	329
6.14.8	Передача сообщений по RemoteTransmitRequest (RTR)	329
6.14.9	Прием сообщений	330
6.14.10	Автоматическая фильтрация принимаемых сообщений.....	330
6.14.11	Перезапись принятых сообщений.....	330
6.14.12	Задание скорости передачи и момента сэмпирования	330
6.14.13	Синхронизация	332
6.14.14	Обработка ошибок.....	332
6.14.15	Прерывания	336
6.14.16	Описание регистров	336
6.15	Контроллер SSP (SSPx)	349
6.15.1	Программируемые параметры.....	350
6.15.2	Общий обзор модуля SSP.....	351

6.15.3	Интерфейс прямого доступа к памяти.....	367
6.15.4	Прерывания	368
6.15.5	Программное управление модулем.....	370
6.15.6	Описание регистров.....	370
6.16	Контроллер UART (UARTx).....	376
6.16.1	Основные характеристики модуля UART	376
6.16.2	Программируемые параметры.....	377
6.16.3	Описание функционирования блока UART	380
6.16.4	Интерфейс прямого доступа к памяти.....	381
6.16.5	Блок и регистры синхронизации	381
6.16.6	Описание функционирования ИК кодека IrDASIR	382
6.16.7	Описание работы UART	383
6.16.8	Работа кодека ИК обмена данными IrDASIR	386
6.16.9	Модуляция данных IrDA	388
6.16.10	Линии управления модемом.....	388
6.16.11	Интерфейс прямого доступа к памяти	390
6.16.12	Прерывания	392
6.16.13	Программное управление модулем	394
6.16.14	Описание регистров	395
6.17	Контроллер интерфейса МКПД (MILx)	410
6.17.1	Режимы работы.....	411
6.17.2	Форматы сообщений.....	412
6.17.3	Формат слов	414
6.17.4	Структурная схема в режиме КШ	416
6.17.5	Структурная схема в режиме ОУ	417
6.17.6	Структурная схема в режиме М	418
6.17.7	Инициализация.....	418
6.17.8	Приём и передача в режиме ОУ	419
6.17.9	Приём и передача в режиме КШ	420
6.17.10	Прерывания	421
6.17.11	Описание регистров	422
6.18	Контроллер приемника телекомандной информации (CCSDS_RX).....	433
6.18.1	Блок приемника телекомандной информации.....	433
6.18.2	Программная модель управления приемником телекомандной информации	435
6.18.3	Описание регистров приёмника телекомандной информации.....	437
6.19	Контроллер передатчика телеметрической информации (CCSDS_TX).....	440
6.19.1	Описание работы блока передатчика телеметрической информации.....	441
6.19.2	Программная модель управления передатчиком телеметрической информации	442
6.19.3	Регистры передатчика телеметрической информации.....	444
6.20	Контроллер блока расчета проверочных кодов (CRC)	449
6.20.1	Общие сведения о циклическом избыточном коде	449
6.20.2	Алгоритм CRC	449
6.20.3	Пример вычисления CRC.....	450
6.20.4	Реализация последовательного алгоритма CRC	451
6.20.5	Реализация алгоритма CRC в микроконтроллере	451
6.20.6	Примеры установки регистров для наиболее популярных протоколов.....	453
6.20.7	Описание регистров.....	457
6.21	Контроллер блока расчета корректирующих кодов (ECC)	462
6.21.1	Реализация алгоритма ECC в микроконтроллере.....	462

6.21.2	Описание регистров управления блока ECC	464
7	Контроллер прерываний NVIC	469
7.1	Логика работы прерываний контроллера NVIC	469
7.1	Регистр разрешения прерываний	472
7.2	Регистр запрета прерываний	473
7.3	Регистр установки состояния ожидания для прерывания	473
7.4	Регистр сброса состояния ожидания для прерывания	474
7.5	Регистры приоритета прерываний	474
7.6	Прерывания, срабатывающие по уровню сигнала	475
7.7	Аппаратное и программное управление прерываниями	475
7.8	Рекомендации по работе с контроллером прерываний	476
8	Прерывания и исключения	478
8.1	Типы исключений	478
8.2	Обработчики исключений	479
8.3	Приоритеты исключений	481
8.4	Вход в обработчик и выход из обработчика	482
9	Типовая схема включения	485
10	Типовые зависимости	486
11	Электрические параметры микросхемы	492
12	Предельно допустимые характеристики микросхемы	494
13	Справочные параметры	496
14	Габаритный чертеж микросхемы	498
15	Информация для заказа	503

1 Структурная блок-схема микросхемы



- - - - - – высокопроизводительная системная шина (AHB);
- - - - - – шина периферии (APB).

Рисунок 4 – Структурная блок-схема микросхемы

2 Условное графическое обозначение

		MCS		
			↔ PA[31:0]	106 – 109, 124 – 130, 159 - 179
			↔ PB[31:0]	68 – 71, 78 - 105
			↔ PC[31:0]	28 – 37, 41 - 67
			↔ PD[31:0]	221, 222, 227 – 232, 234 – 239, 3 – 6, 14 - 27
			↔ PE[31:0]	185 – 192, 197 - 220
9	nRESET0			
8	nRESET1			
147	SPW0_RXDP		SPW0_TXDP	153
148	SPW0_RXDN		SPW0_TXDN	154
149	SPW0_RXSP		SPW0_TXSP	151
150	SPW0_RXSN		SPW0_TXSN	152
138	SPW1_RXDP		SPW1_TXDP	145
139	SPW1_RXDN		SPW1_TXDN	146
140	SPW1_RXSP		SPW1_TXSP	143
141	SPW1_RXSN		SPW1_TXSN	144
7	SHDN			
233	JTAG_EN		TESTI	112
118	OSCI1		OSCO1	119
116	OSCI0		OSCO0	117
184	OSCI2		OSCO2	183
11, 39, 73, 74, 110, 132, 133, 157, 194, 225	VI_LDO			
114, 115	VDO_PLL			
13, 77, 131, 223	VDDO		GND	12, 40, 75, 76, 113, 135, 136, 158, 195, 196, 226
10, 38, 72, 111, 134, 156, 193, 224	VDD		GND_SPW	137
155	VDD_SPW		NC	1, 2, 58 – 62, 120 – 123, 142, 180 – 182, 240

Рисунок 5 – Условное графическое обозначение

3 Описание выводов

Таблица 1 – Описание выводов

Номер вывода корпуса	Номер контактной площадки (КП) кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
1	–	NC	–	Не используется. Рекомендуется подключать к шине «Общий»
2	–	NC	–	Не используется
3	59	PD17	IO 5V	Пользовательский порт D ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
4	60	PD16	IO 5V	
5	61	PD15	IO 5V	
6	62	PD14	IO 5V	
7	63	SHDN	I	Вход выключения встроенных регуляторов напряжения: 0 – регулятор включен; 1 – регулятор выключен
8	64	nRESET1	I 5V	Сброс: 0 – сброс при условии nRESET0 = «0»
9	65	nRESET0	I 5V	Сброс: 0 – сброс при условии nRESET1 = «0»
10	66	VDD	PWR	Питание площадок ввода / вывода
11	67	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
	68			
12	69	GND	GND	Общий
	70			
13	71	VDDO	PWR	Питание цифрового домена, при SHDN = «1»
			AO	Выход встроенного LDO регулятора, при SHDN = «0»
14	72	PD13	IO 5V	Пользовательский порт D ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
15	73	PD12	IO 5V	
16	74	PD11	IO 5V	
17	75	PD10	IO 5V	
18	76	PD9	IO 5V	
19	77	PD8	IO 5V	
20	78	PD7	IO 5V	
21	79	PD6	IO 5V	
22	80	PD5	IO 5V	
23	81	PD4	IO 5V	
24	82	PD3	IO 5V	
25	83	PD2	IO 5V	
26	84	PD1	IO 5V	
27	85	PD0	IO 5V	Пользовательский порт D ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (младший разряд)
28	86	PC31	IO 5V	Пользовательский порт C ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (старший разряд)

Номер вывода корпуса	Номер контактной площадки (КП) кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
29	87	PC30	IO 5V	Пользовательский порт С ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
30	88	PC29	IO 5V	
31	89	PC28	IO 5V	
32	90	PC27	IO 5V	
33	91	PC26	IO 5V	
34	92	PC25	IO 5V	
35	93	PC24	IO 5V	
36	94	PC23	IO 5V	
37	95	PC22	IO 5V	
38	96	VDD	PWR	Питание площадок ввода / вывода
39	97	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
40	98	GND	GND	Общий
41	99	PC21	IO 5V	Пользовательский порт С ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
42	100	PC20	IO 5V	
43	101	PC19	IO 5V	
44	102	PC18	IO 5V	
45	103	PC17	IO 5V	
46	104	PC16	IO 5V	
47	105	PC15	IO 5V	
48	106	PC14	IO 5V	
49	107	PC13	IO 5V	
50	108	PC12	IO 5V	
51	109	PC11	IO 5V	
52	110	PC10	IO 5V	
53	111	PC9	IO 5V	
54	112	PC8	IO 5V	
55	113	PC7	IO 5V	
56	114	PC6	IO 5V	
57	115	PC5	IO 5V	
58	—	NC	—	Не используется
59	—	NC	—	Не используется
60	—	NC	—	Не используется
61	—	NC	—	Не используется
62	—	NC	—	Не используется
63	116	PC4	IO 5V	Пользовательский порт С ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
64	117	PC3	IO 5V	
65	118	PC2	IO 5V	
66	119	PC1	IO 5V	
67	120	PC0	IO 5V	Пользовательский порт С ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (младший разряд)
68	121	PB31	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (старший разряд)
69	122	PB30	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины,
70	123	PB29	IO 5V	

Номер вывода корпуса	Номер контактной площадки (КП) кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
71	124	PB28	IO 5V	интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
72	125	VDD	PWR	Питание площадок ввода / вывода
73	126	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
74	127	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
75	128	GND	GND	Общий
76	129	GND	GND	Общий
77	130	VDDO	PWR	Питание цифрового домена, при SHDN = «1»
			AO	Выход встроенного LDO регулятора, при SHDN = «0»
78	131	PB27	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
79	132	PB26	IO 5V	
80	133	PB25	IO 5V	
81	134	PB24	IO 5V	
82	135	PB23	IO 5V	
83	136	PB22	IO 5V	
84	137	PB21	IO 5V	
85	138	PB20	IO 5V	
86	139	PB19	IO 5V	
87	140	PB18	IO 5V	
88	141	PB17	IO 5V	
89	142	PB16	IO 5V	
90	143	PB15	IO 5V	
91	144	PB14	IO 5V	
92	145	PB13	IO 5V	
93	146	PB12	IO 5V	
94	147	PB11	IO 5V	
95	148	PB10	IO 5V	
96	149	PB9	IO 5V	
97	150	PB8	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_B вывод PB8 совмещен с JTAGB_SWVTDO
98	151	PB7	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_B вывод PB7 совмещен с JTAGB_TDI
99	152	PB6	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_B вывод PB6 совмещен с JTAGB_SWDIOTMS

Номер вывода корпуса	Номер контактной площадки (КП) кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
100	153	PB5	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_B вывод PB5 совмещен с JTAGB_SWCLKTCK
101	154	PB4	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_B вывод PB4 совмещен с JTAGB_nTRST
102	155	PB3	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
103	156	PB2	IO 5V	
104	157	PB1	IO 5V	
105	158	PB0	IO 5V	Пользовательский порт В ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (младший разряд)
106	159	PA31	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (старший разряд)
107	160	PA30	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
108	161	PA29	IO 5V	
109	162	PA28	IO 5V	
110	163	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
111	164	VDD	PWR	Питание площадок ввода / вывода
112	165	TESTI	O	Вывод проверки источников опорных токов. Подключать к U_{CC}
113	166	GND	GND	Общий
114	167	VDO_PLL	PWR	Питание домена PLL, при SHDN = «1»
			AO	Выход встроенного LDO регулятора, при SHDN = «0»
115	168	VDO_PLL	PWR	Питание домена PLL, при SHDN = «1»
			AO	Выход встроенного LDO регулятора, при SHDN = «0»
116	169	OSCI0	A	Вход тактового генератора HSE0, вывод для подключения кварцевого резонатора
117	170	OSCO0	A	Вывод для подключения кварцевого резонатора
118	171	OSCI1	A	Вход тактового генератора HSE1, вывод для подключения кварцевого резонатора
119	172	OSCO1	A	Вывод для подключения кварцевого резонатора
120	—	NC	—	Не используется
121	—	NC	—	Не используется
122	—	NC	—	Не используется

Номер вывода корпуса	Номер контактной площадки (КП) кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
123	—	NC	—	Не используется
124	173	PA27	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
125	174	PA26	IO 5V	
126	175	PA25	IO 5V	
127	176	PA24	IO 5V	
128	177	PA23	IO 5V	
129	178	PA22	IO 5V	
130	179	PA21	IO 5V	
131	180	VDDO	PWR	Питание цифрового домена, при SHDN = «1»
			AO	Выход встроенного LDO регулятора, при SHDN = «0»
132	181	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
133	182	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
134	183	VDD	PWR	Питание площадок ввода / вывода
135	184	GND	GND	Общий
136	185	GND	GND	Общий
137	186	GND_SPW	GND	Общий приемопередатчика SpaceWire PHY
138	187	SPW1_RXDP	AI	Входы приемопередатчика интерфейса SpaceWire1
139	188	SPW1_RXDN	AI	
140	189	SPW1_RXSP	AI	
141	190	SPW1_RXSN	AI	
142	—	NC	—	Не используется
143	191	SPW1_TXSP	AO	Выходы приемопередатчика интерфейса SpaceWire1
144	192	SPW1_TXSN	AO	
145	193	SPW1_TXDP	AO	
146	194	SPW1_TXDN	AO	
147	195	SPW0_RXDP	AI	Входы приемопередатчика интерфейса SpaceWire0
148	196	SPW0_RXDN	AI	
149	197	SPW0_RXSP	AI	
150	198	SPW0_RXSN	AI	
151	199	SPW0_TXSP	AO	Выходы приемопередатчика интерфейса SpaceWire0
152	200	SPW0_TXSN	AO	
153	201	SPW0_TXDP	AO	
154	202	SPW0_TXDN	AO	
155	203	VDD_SPW	PWR	Питание приемопередатчика SpaceWire PHY
156	204	VDD	PWR	Питание площадок ввода / вывода
157	205	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
158	206	GND	GND	Общий
159	207	PA20	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
160	208	PA19	IO 5V	
161	209	PA18	IO 5V	
162	210	PA17	IO 5V	
163	211	PA16	IO 5V	

Номер вывода корпуса	Номер контактной площадки (КП) кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
164	212	PA15	IO 5V	
165	213	PA14	IO 5V	
166	214	PA13	IO 5V	
167	215	PA12	IO 5V	
168	216	PA11	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
169	217	PA10	IO 5V	
170	218	PA9	IO 5V	
171	219	PA8	IO 5V	
172	220	PA7	IO 5V	
173	221	PA6	IO 5V	
174	222	PA5	IO 5V	
175	223	PA4	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_A вывод PA4 совмещен с JTAGA_SWVTDO
176	224	PA3	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_A вывод PA3 совмещен с JTAGA_TDI
177	225	PA2	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_A вывод PA2 совмещен с JTAGA_SWDIOTMS
178	226	PA1	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS. При выборе JTAG_A вывод PA1 совмещен с JTAGA_SWCLKTCK
179	227	PA0	IO 5V	Пользовательский порт А ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (младший разряд). При выборе JTAG_A вывод PA0 совмещен с JTAGA_nTRST
180	—	NC	—	Не используется
181	—	NC	—	Не используется
182	—	NC	—	Не используется
183	1	OSCO2	A	Вывод для подключения кварцевого резонатора
184	2	OSCI2	A	Вход тактового генератора LSE, вывод для подключения кварцевого резонатора
185	3	PE31	IO 5V	Пользовательский порт Е ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (старший разряд)

Номер вывода корпуса	Номер контактной площадки (КП) кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
186	4	PE30	IO 5V	Пользовательский порт E ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
187	5	PE29	IO 5V	
188	6	PE28	IO 5V	
189	7	PE27	IO 5V	
190	8	PE26	IO 5V	
191	9	PE25	IO 5V	
192	10	PE24	IO 5V	
193	11	VDD	PWR	Питание площадок ввода / вывода
194	12	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
	13			
195	14	GND	GND	Общий
196	15	GND	GND	Общий
197	16	PE23	IO 5V	Пользовательский порт E ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
198	17	PE22	IO 5V	
199	18	PE21	IO 5V	
200	19	PE20	IO 5V	
201	20	PE19	IO 5V	
202	21	PE18	IO 5V	
203	22	PE17	IO 5V	
204	23	PE16	IO 5V	
205	24	PE15	IO 5V	
206	25	PE14	IO 5V	
207	26	PE13	IO 5V	
208	27	PE12	IO 5V	
209	28	PE11	IO 5V	
210	29	PE10	IO 5V	
211	30	PE9	IO 5V	
212	31	PE8	IO 5V	
213	32	PE7	IO 5V	
214	33	PE6	IO 5V	
215	34	PE5	IO 5V	
216	35	PE4	IO 5V	
217	36	PE3	IO 5V	
218	37	PE2	IO 5V	
219	38	PE1	IO 5V	
220	39	PE0	IO 5V	Пользовательский порт E ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (младший разряд)
221	40	PD31	IO 5V	Пользовательский порт D ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS (старший разряд)
222	41	PD30	IO 5V	Пользовательский порт D ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSD
223	42	VDDO	PWR	Питание цифрового домена, при SHDN = «1»

Номер вывода корпуса	Номер контактной площадки (КП) кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
			AO	Выход встроенного LDO регулятора, при SHDN = «0»
224	43	VDD	PWR	Питание площадок ввода / вывода
225	44	VI_LDO	AI	Входное напряжение встроенного LDO регулятора для формирования напряжений питания U_{O_LDO} , U_{CC_PLL}
226	45	GND	GND	Общий
227	46	PD29	IO 5V	Пользовательский порт D ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
228	47	PD28	IO 5V	
229	48	PD27	IO 5V	
230	49	PD26	IO 5V	
231	50	PD25	IO 5V	
232	51	PD24	IO 5V	
233	52	JTAG_EN	I	Вход выбора тестового режима: 0 – рабочий режим; 1 – тестовый режим
234	53	PD23	IO 5V	Пользовательский порт D ввода / вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
235	54	PD22	IO 5V	
236	55	PD21	IO 5V	
237	56	PD20	IO 5V	
238	57	PD19	IO 5V	
239	58	PD18	IO 5V	
240	–	NC	–	Не используется
Примечание – Обозначение типов выводов: A – аналоговых вывод; AI – аналоговый вход; AO – аналоговый выход; AIO – цифровой вход / выход совмещенный с аналоговых выводом; I – цифровой вход; O – цифровой выход; IO – цифровой вход / выход; 5V – толерантный к 5 В цифровой вход; PWR – питание; GND – общий.				

Таблица 2 – Описание выводов по блокам

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
Порт А				
179	227	PA0	IO 5V	Пользовательский порт А ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS При выборе JTAG_A Вывод PA0 совмещен с JTAGA_nTRST Вывод PA1 совмещен с JTAGA_SWCLKTCK Вывод PA2 совмещен с JTAGA_SWDIOTMS Вывод PA3 совмещен с JTAGA_TDI Вывод PA4 совмещен с JTAGA_SWVTDO
178	226	PA1	IO 5V	
177	225	PA2	IO 5V	
176	224	PA3	IO 5V	
175	223	PA4	IO 5V	
174	222	PA5	IO 5V	
173	221	PA6	IO 5V	
172	220	PA7	IO 5V	
171	219	PA8	IO 5V	
170	218	PA9	IO 5V	
169	217	PA10	IO 5V	
168	216	PA11	IO 5V	
167	215	PA12	IO 5V	
166	214	PA13	IO 5V	
165	213	PA14	IO 5V	
164	212	PA15	IO 5V	
163	211	PA16	IO 5V	
162	210	PA17	IO 5V	
161	209	PA18	IO 5V	
160	208	PA19	IO 5V	
159	207	PA20	IO 5V	
130	179	PA21	IO 5V	
129	178	PA22	IO 5V	
128	177	PA23	IO 5V	
127	176	PA24	IO 5V	
126	175	PA25	IO 5V	
125	174	PA26	IO 5V	
124	173	PA27	IO 5V	
109	162	PA28	IO 5V	
108	161	PA29	IO 5V	
107	160	PA30	IO 5V	
106	159	PA31	IO 5V	
Порт В				
105	158	PB0	IO 5V	Пользовательский порт В ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS При выборе JTAG_B Вывод PB4 совмещен с JTAGB_nTRST Вывод PB5 совмещен с JTAGB_SWCLKTCK Вывод PB6 совмещен с JTAGB_SWDIOTMS Вывод PB7 совмещен с JTAGB_TDI Вывод PB8 совмещен с JTAGB_SWVTDO
104	157	PB1	IO 5V	
103	156	PB2	IO 5V	
102	155	PB3	IO 5V	
101	154	PB4	IO 5V	
100	153	PB5	IO 5V	
99	152	PB6	IO 5V	
98	151	PB7	IO 5V	
97	150	PB8	IO 5V	
96	149	PB9	IO 5V	
95	148	PB10	IO 5V	
94	147	PB11	IO 5V	
93	146	PB12	IO 5V	
92	145	PB13	IO 5V	
91	144	PB14	IO 5V	

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
90	143	PB15	IO 5V	
89	142	PB16	IO 5V	
88	141	PB17	IO 5V	
87	140	PB18	IO 5V	
86	139	PB19	IO 5V	
85	138	PB20	IO 5V	
84	137	PB21	IO 5V	
83	136	PB22	IO 5V	
82	135	PB23	IO 5V	
81	134	PB24	IO 5V	
80	133	PB25	IO 5V	
79	132	PB26	IO 5V	
78	131	PB27	IO 5V	
71	124	PB28	IO 5V	
70	123	PB29	IO 5V	
69	122	PB30	IO 5V	
68	121	PB31	IO 5V	
Порт С				
67	120	PC0	IO 5V	Пользовательский порт С ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
66	119	PC1	IO 5V	
65	118	PC2	IO 5V	
64	117	PC3	IO 5V	
63	116	PC4	IO 5V	
57	115	PC5	IO 5V	
56	114	PC6	AIO	
55	113	PC7	AIO	
54	112	PC8	AIO	
53	111	PC9	AIO	
52	110	PC10	AIO	
51	109	PC11	AIO	
50	108	PC12	AIO	
49	107	PC13	AIO	
48	106	PC14	AIO	
47	105	PC15	AIO	
46	104	PC16	AIO	
45	103	PC17	AIO	
44	102	PC18	AIO	
43	101	PC19	AIO	
42	100	PC20	AIO	
41	99	PC21	AIO	
37	95	PC22	AIO	
36	94	PC23	AIO	
35	93	PC24	AIO	
34	92	PC25	AIO	
33	91	PC26	AIO	
32	90	PC27	AIO	
31	89	PC28	AIO	
30	88	PC29	AIO	
29	87	PC30	IO 5V	
28	86	PC31	IO 5V	

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
Порт D				
27	85	PD0	IO 5V	Пользовательский порт D ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
26	84	PD1	IO 5V	
25	83	PD2	IO 5V	
24	82	PD3	IO 5V	
23	81	PD4	IO 5V	
22	80	PD5	IO 5V	
21	79	PD6	IO 5V	
20	78	PD7	IO 5V	
19	77	PD8	IO 5V	
18	76	PD9	IO 5V	
17	75	PD10	IO 5V	
16	74	PD11	IO 5V	
15	73	PD12	IO 5V	
14	72	PD13	IO 5V	
6	62	PD14	IO 5V	
5	61	PD15	IO 5V	
4	60	PD16	IO 5V	
3	59	PD17	IO 5V	
239	58	PD18	IO 5V	
238	57	PD19	IO 5V	
237	56	PD20	IO 5V	
236	55	PD21	IO 5V	
235	54	PD22	IO 5V	
234	53	PD23	IO 5V	
232	51	PD24	IO 5V	
231	50	PD25	IO 5V	
230	49	PD26	IO 5V	
229	48	PD27	IO 5V	
228	47	PD28	IO 5V	
227	46	PD29	IO 5V	
222	41	PD30	IO 5V	
221	40	PD31	IO 5V	
Порт E				
220	39	PE0	IO 5V	Пользовательский порт E ввода-вывода, совмещен с функциями системной шины, интерфейсов UART, SSP, CAN, MIL, Timer, CCSDS
219	38	PE1	IO 5V	
218	37	PE2	IO 5V	
217	36	PE3	IO 5V	
216	35	PE4	IO 5V	
215	34	PE5	IO 5V	
214	33	PE6	IO 5V	
213	32	PE7	IO 5V	
212	31	PE8	IO 5V	
211	30	PE9	IO 5V	
210	29	PE10	IO 5V	
209	28	PE11	IO 5V	
208	27	PE12	IO 5V	
207	26	PE13	IO 5V	
206	25	PE14	IO 5V	
205	24	PE15	IO 5V	

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
204	23	PE16	IO 5V	
203	22	PE17	IO 5V	
202	21	PE18	IO 5V	
201	20	PE19	IO 5V	
200	19	PE20	IO 5V	
199	18	PE21	IO 5V	
198	17	PE22	IO 5V	
197	16	PE23	IO 5V	
192	10	PE24	IO 5V	
191	9	PE25	IO 5V	
190	8	PE26	IO 5V	
189	7	PE27	IO 5V	
188	6	PE28	IO 5V	
187	5	PE29	IO 5V	
186	4	PE30	IO 5V	
185	3	PE31	IO 5V	
Интерфейс SpaceWire				
154	202	SPW0_TXDN	AO	Сигналы приемопередатчика интерфейса SpaceWire0
153	201	SPW0_TXDP	AO	
152	200	SPW0_TXSN	AO	
151	199	SPW0_TXSP	AO	
150	198	SPW0_RXSN	AI	
149	197	SPW0_RXSP	AI	
148	196	SPW0_RXDN	AI	
147	195	SPW0_RXDP	AI	Сигналы приемопередатчика интерфейса SpaceWire1
146	194	SPW1_TXDN	AO	
145	193	SPW1_TXDP	AO	
144	192	SPW1_TXSN	AO	
143	191	SPW1_TXSP	AO	
141	190	SPW1_RXSN	AI	
140	189	SPW1_RXSP	AI	
139	188	SPW1_RXDN	AI	
138	187	SPW1_RXDP	AI	
Сигналы управления				
9	65	nRESET0	I 5V	Вход нулевого сигнала сброса 0 – запрос сброса 1 – рабочий режим
8	64	nRESET1	I 5V	Вход первого сигнала сброса 0 – запрос сброса 1 – рабочий режим Сброс микросхемы выполняется при условии запроса сброса по обоим выводам nRESET0 и nRESET1
233	52	JTAG_EN	I	Вход выбора тестового режима 0 – рабочий режим 1 – тестовый режим
7	63	SHDN	I	Вход выключения встроенных регуляторов напряжения 0 – регулятор включен 1 – регулятор выключен

Номер вывода корпуса	Номер КП кристалла	Обозначение вывода	Тип вывода	Функциональное назначение
112		TESTI	O	Вывод проверки источников опорных токов. Подключать к U _{CC}
Сигналы тактирования				
117	170	OSCO0	A	Вывод для подключения кварцевого резонатора к генератору HSE0
116	169	OSCI0	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору
119	172	OSCO1	A	Вывод для подключения кварцевого резонатора к генератору HSE1
118	171	OSCI1	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору
183	1	OSCO2	A	Вывод для подключения кварцевого резонатора к генератору LSE
184	2	OSCI2	A	Вывод для подключения внешнего генератора или кварцевого резонатора к генератору LSE
Вспомогательные сигналы				
13, 77, 131, 223	71, 130, 180, 42	VDDO	PWR	Напряжение питания цифрового домена формируется встроенным LDO регулятором
114, 115	167, 168	VDO_PLL	PWR	Напряжение питания домена PLL формируется встроенным LDO регулятором
10, 38, 72, 111, 134, 156, 193, 224	66, 96, 125, 164, 183, 204, 11, 43	VDD	PWR	Питание площадок ввода-вывода
11, 39, 73, 74, 110, 132, 133, 157, 194, 225	67, 68, 97, 126, 127, 163, 181, 182, 205, 12, 13, 44	VI_LDO	PWR	Питание встроенного регулятора для формирования напряжений питания VDDO, VDO_PLL
155	203	SPW_VDD	PWR	Питание приемопередатчика SpaceWire PHY
12, 40, 75, 76, 113, 135, 136, 158, 195, 196, 226	69, 70, 98, 128, 129, 166, 184, 185, 206, 14, 15, 45	GND	PWR	Общий
137	186	SPW_GND	PWR	Общий сигнал «земля» приемопередатчика SpaceWirePHY
Обозначения: 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				

Таблица 3 – Дополнительные назначения выводов

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
3	59	PD17	–	ADDR[19]	–	CAN1_TX	UART1_TX	SSP2_FS	TMR0_CH2N	TMR0_CH2N	TMR2_ETR	MIL1_RXAP	MIL2_RXAP	–	SSP5_FS
4	60	PD16	–	ADDR[18]	–	CAN0_RX	UART0_RX	SSP2_CLK	TMR0_CH2P	TMR0_CH3P	TMR1_BRK	MIL1_ENA	MIL2_ENA	–	SSP5_CLK
5	61	PD15	–	ADDR[17]	–	CAN0_TX	UART0_TX	SSP2_TX	TMR0_CH1N	TMR0_CH3N	TMR1_CH3N	MIL1_TXAP	MIL2_TXAP	–	SSP5_TX
6	62	PD14	–	ADDR[16]	–	CAN1_RX	UART1_RX	SSP1_RX	TMR0_CH1P	TMR0_BRK	TMR1_CH3P	MIL1_TXAN	MIL2_TXAN	–	TMTD_VAL
14	72	PD13	–	ADDR[15]	–	CAN1_TX	UART1_TX	SSP1_FS	TMR0_CH0N	TMR1_ETR	TMR1_CH2N	MIL0_RXBN	MIL3_RXBN	–	TMTD_DATA
15	73	PD12	–	ADDR[14]	–	CAN0_RX	UART0_RX	SSP1_CLK	TMR0_CH0P	TMR1_CH0P	TMR1_CH2P	MIL0_RXBP	MIL3_RXBP	–	TMTD_CLK
16	74	PD11	–	ADDR[13]	–	CAN0_TX	UART0_TX	SSP1_TX	TMR0_ETR	TMR1_CH0N	TMR1_CH1N	MIL0_ENB	MIL3_ENB	–	–
17	75	PD10	–	ADDR[12]	–	CAN0_RX	UART1_RX	SSP0_RX	TMR3_BRK	TMR1_CH1P	TMR1_CH1P	MIL0_TXBP	MIL3_TXBP	–	SSP4_RX
18	76	PD9	–	ADDR[11]	–	CAN0_TX	UART1_TX	SSP0_FS	TMR3_CH3N	TMR1_CH1N	TMR1_CH0N	MIL0_TXBN	MIL3_TXBN	–	SSP4_FS
19	77	PD8	–	ADDR[10]	–	CAN1_RX	UART0_RX	SSP0_CLK	TMR3_CH3P	TMR1_CH2P	TMR1_CH0P	MIL0_RXAN	MIL3_RXAN	–	SSP4_CLK
20	78	PD7	–	ADDR[9]	–	CAN1_TX	UART0_TX	SSP0_TX	TMR3_CH2N	TMR1_CH2N	TMR1_ETR	MIL0_RXAP	MIL3_RXAP	–	SSP4_TX
21	79	PD6	–	ADDR[8]	–	CAN0_RX	UART1_RX	SSP3_RX	TMR3_CH2P	TMR1_CH3P	TMR0_BRK	MIL0_ENA	MIL3_ENA	–	TMRC_VAL
22	80	PD5	–	ADDR[7]	–	CAN0_TX	UART1_TX	SSP3_FS	TMR3_CH1N	TMR1_CH3N	TMR0_CH3N	MIL0_TXAP	MIL3_TXAP	–	TMRC_DATA
23	81	PD4	–	ADDR[6]	–	CAN1_RX	UART0_RX	SSP3_CLK	TMR3_CH1P	TMR1_BRK	TMR0_CH3P	MIL0_TXAN	MIL3_TXAN	–	TMRC_CLK
24	82	PD3	–	ADDR[5]	–	CAN1_TX	UART0_TX	SSP3_TX	TMR3_CH0N	TMR0_ETR	TMR0_CH2N	MIL1_RXBN	MIL2_RXBN	–	–

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
25	83	PD2	–	ADDR[4]	–	CAN0_RX	UART1_RX	SSP2_RX	TMR3_CH0P	TMR0_CH0P	TMR0_CH2P	MIL1_RXBP	MIL2_RXBP	–	SSP5_RX
26	84	PD1	–	ADDR[3]	–	CAN0_TX	UART1_TX	SSP2_FS	TMR3_ETR	TMR0_CH0N	TMR0_CH1N	MIL1_ENB	MIL2_ENB	–	SSP5_FS
27	85	PD0	–	ADDR[2]	–	CAN0_RX	UART0_RX	SSP2_CLK	TMR2_BRK	TMR0_CH1P	TMR0_CH1P	MIL1_TXBP	MIL2_TXBP	–	SSP5_CLK
28	86	PC31	–	ADDR[1]	ADDR[22]	CAN0_TX	UART0_TX	SSP2_TX	TMR2_CH3N	TMR0_CH1N	TMR0_CH0N	MIL1_TXBN	MIL2_TXBN	–	SSP5_TX
29	87	PC30	–	ADDR[0]	ADDR[21]	CAN1_RX	UART1_RX	SSP1_RX	TMR2_CH3P	TMR0_CH2P	TMR0_CH0P	MIL1_RXAN	MIL2_RXAN	–	TMTD_VAL
30	88	PC29	–	ADDR[31]	CLOCK	CAN1_TX	UART1_TX	SSP1_FS	TMR2_CH2N	TMR0_CH2N	TMR0_ETR	MIL1_RXAP	MIL2_RXAP	–	TMTD_DATA
31	89	PC28	–	–	OCLK	CAN0_RX	UART0_RX	SSP1_CLK	TMR2_CH2P	TMR0_CH3P	TMR1_BRK	MIL1_ENA	MIL2_ENA	–	TMTD_CLK
32	90	PC27	–	ADDR[30]	–	CAN0_TX	UART0_TX	SSP1_TX	TMR2_CH1N	TMR0_CH3N	TMR1_CH3N	MIL1_TXAP	MIL2_TXAP	–	–
33	91	PC26	–	CSn[5]	–	CAN1_RX	UART1_RX	SSP0_RX	TMR2_CH1P	TMR0_BRK	TMR1_CH3P	MIL1_TXAN	MIL2_TXAN	–	SSP4_RX
34	92	PC25	–	–	–	CAN1_TX	UART1_TX	SSP0_FS	TMR2_CH0N	TMR3_ETR	TMR1_CH2N	MIL0_RXBN	MIL3_RXBN	–	SSP4_FS
35	93	PC24	–	CSn[6]	–	CAN0_RX	UART0_RX	SSP0_CLK	TMR2_CH0P	TMR3_CH0P	TMR1_CH2P	MIL0_RXBP	MIL3_RXBP	–	SSP4_CLK
36	94	PC23	–	–	CLOCK	CAN0_TX	UART0_TX	SSP0_TX	TMR2_ETR	TMR3_CH0N	TMR1_CH1N	MIL0_ENB	MIL3_ENB	–	SSP4_TX
37	95	PC22	–	–	OCLK	CAN0_RX	UART1_RX	SSP3_RX	TMR1_BRK	TMR3_CH1P	TMR1_CH1P	MIL0_TXBP	MIL3_TXBP	–	TMRC_VAL
41	99	PC21	–	ADDR[29]	–	CAN0_TX	UART1_TX	SSP3_FS	TMR1_CH3N	TMR3_CH1N	TMR1_CH0N	MIL0_TXBN	MIL3_TXBN	–	TMRC_DATA
42	100	PC20	–	ADDR[28]	–	CAN1_RX	UART0_RX	SSP3_CLK	TMR1_CH3P	TMR3_CH2P	TMR1_CH0P	MIL0_RXAN	MIL3_RXAN	–	TMRC_CLK
43	101	PC19	–	ADDR[27]	–	CAN1_TX	UART0_TX	SSP3_TX	TMR1_CH2N	TMR3_CH2N	TMR1_ETR	MIL0_RXAP	MIL3_RXAP	–	–

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
44	102	PC18	–	ADDR[26]	–	CAN0_RX	UART1_RX	SSP2_RX	TMR1_CH2P	TMR3_CH3P	TMR0_BRK	MIL0_ENA	MIL3_ENA	–	SSP5_RX
45	103	PC17	–	ADDR[25]	–	CAN0_TX	UART1_TX	SSP2_FS	TMR1_CH1N	TMR3_CH3N	TMR0_CH3N	MIL0_TXAP	MIL3_TXAP	–	SSP5_FS
46	104	PC16	–	ADDR[24]	–	CAN1_RX	UART0_RX	SSP2_CLK	TMR1_CH1P	TMR3_BRK	TMR0_CH3P	MIL0_TXAN	MIL3_TXAN	–	SSP5_CLK
47	105	PC15	–	ADDR[23]	–	CAN1_TX	UART0_TX	SSP2_TX	TMR1_CH0N	TMR2_ETR	TMR0_CH2N	MIL1_RXBN	MIL2_RXBN	–	SSP5_TX
48	106	PC14	–	ADDR[22]	–	CAN0_RX	UART1_RX	SSP1_RX	TMR1_CH0P	TMR2_CH0P	TMR0_CH2P	MIL1_RXBP	MIL2_RXBP	–	TMTD_VAL
49	107	PC13	–	ADDR[21]	–	CAN0_TX	UART1_TX	SSP1_FS	TMR1_ETR	TMR2_CH0N	TMR0_CH1N	MIL1_ENB	MIL2_ENB	–	TMTD_DATA
50	108	PC12	–	CSn[7]	–	CAN0_RX	UART0_RX	SSP1_CLK	TMR0_BRK	TMR2_CH1P	TMR0_CH1P	MIL1_TXBP	MIL2_TXBP	–	TMTD_CLK
51	109	PC11	–	–	–	CAN0_TX	UART0_TX	SSP1_TX	TMR0_CH3N	TMR2_CH1N	TMR0_CH0N	MIL1_TXBN	MIL2_TXBN	–	–
52	110	PC10	–	–	–	CAN1_RX	UART1_RX	SSP0_RX	TMR0_CH3P	TMR2_CH2P	TMR0_CH0P	MIL1_RXAN	MIL2_RXAN	–	SSP4_RX
53	111	PC9	–	–	–	CAN1_TX	UART1_TX	SSP0_FS	TMR0_CH2N	TMR2_CH2N	TMR0_ETR	MIL1_RXAP	MIL2_RXAP	–	SSP4_FS
54	112	PC8	–	–	–	CAN0_RX	UART0_RX	SSP0_CLK	TMR0_CH2P	TMR2_CH3P	TMR3_BRK	MIL1_ENA	MIL2_ENA	–	SSP4_CLK
55	113	PC7	–	–	–	CAN0_TX	UART0_TX	SSP0_TX	TMR0_CH1N	TMR2_CH3N	TMR3_CH3N	MIL1_TXAP	MIL2_TXAP	–	SSP4_TX
56	114	PC6	–	–	–	CAN1_RX	UART1_RX	SSP3_RX	TMR0_CH1P	TMR2_BRK	TMR3_CH3P	MIL1_TXAN	MIL2_TXAN	–	TMRC_VAL
57	115	PC5	–	OEn[0]	–	CAN1_TX	UART1_TX	SSP3_FS	TMR0_CH0N	TMR1_ETR	TMR3_CH2N	MIL0_RXBN	MIL3_RXBN	–	–
63	116	PC4	–	WEn[0]	–	CAN0_RX	UART0_RX	SSP3_CLK	TMR0_CH0P	TMR1_CH0P	TMR3_CH2P	MIL0_RXBP	MIL3_RXBP	–	TMRC_CLK
64	117	PC3	–	BWEn[2]	BEn[2]	CAN0_TX	UART0_TX	SSP3_TX	TMR0_ETR	TMR1_CH0N	TMR3_CH1N	MIL0_ENB	MIL3_ENB	–	–

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
65	118	PC2	–	BWEn[3]	BEn[3]	CAN0_RX	UART1_RX	SSP2_RX	TMR1_BRK	TMR1_CH1P	TMR3_CH1P	MIL0_TXBP	MIL3_TXBP	–	SSP5_RX
66	119	PC1	–	BWEn[0]	BEn[0]	CAN0_TX	UART1_TX	SSP2_FS	TMR1_CH3N	TMR1_CH1N	TMR3_CH0N	MIL0_TXBN	MIL3_TXBN	–	SSP5_FS
67	120	PC0	–	BWEn[1]	BEn[1]	CAN1_RX	UART0_RX	SSP2_CLK	TMR1_CH3P	TMR1_CH2P	TMR3_CH0P	MIL0_RXAN	MIL3_RXAN	–	SSP5_CLK
68	121	PB31	–	BWEn[3]	BEn[3]	CAN1_TX	UART0_TX	SSP2_TX	TMR1_CH2N	TMR1_CH2N	TMR3_ETR	MIL0_RXAP	MIL3_RXAP	–	SSP5_TX
69	122	PB30	–	CS[4]	–	CAN0_RX	UART1_RX	SSP1_RX	TMR1_CH2P	TMR1_CH3P	TMR2_BRK	MIL0_ENA	MIL3_ENA	–	TMTD_VAL
70	123	PB29	–	CS[5]	–	CAN0_TX	UART1_TX	SSP1_FS	TMR1_CH1N	TMR1_CH3N	TMR2_CH3N	MIL0_TXAP	MIL3_TXAP	–	–
71	124	PB28	–	CS[6]	CLOCK	CAN1_RX	UART0_RX	SSP1_CLK	TMR1_CH1P	TMR1_BRK	TMR2_CH3P	MIL0_TXAN	MIL3_TXAN	–	TMTD_CLK
78	131	PB27	–	CS[7]	OCLK	CAN1_TX	UART0_TX	SSP1_TX	TMR1_CH0N	TMR0_ETR	TMR2_CH2N	MIL1_RXBN	MIL2_RXBN	–	–
79	132	PB26	CADDR[2]	DATA[16]	–	CAN0_RX	UART1_RX	SSP0_RX	TMR1_CH0P	TMR0_CH0P	TMR2_CH2P	MIL1_RXBP	MIL2_RXBP	–	SSP4_RX
80	133	PB25	CADDR[3]	DATA[17]	–	CAN0_TX	UART1_TX	SSP0_FS	TMR1_ETR	TMR0_CH0N	TMR2_CH1N	MIL1_ENB	MIL2_ENB	–	SSP4_FS
81	134	PB24	CADDR[4]	DATA[18]	–	CAN0_RX	UART0_RX	SSP0_CLK	TMR0_BRK	TMR0_CH1P	TMR2_CH1P	MIL1_TXBP	MIL2_TXBP	–	SSP4_CLK
82	135	PB23	CADDR[5]	DATA[19]	–	CAN0_TX	UART0_TX	SSP0_TX	TMR0_CH3N	TMR0_CH1N	TMR2_CH0N	MIL1_TXBN	MIL2_TXBN	–	SSP4_TX

№ вы-вода корпуса	№ КП кри-стал-ла	Обозна-чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
83	136	PB22	CADDR[6]	DATA[20]	–	CAN1_RX	UART1_RX	SSP3_RX	TMR0_CH3P	TMR0_CH2P	TMR2_CH0P	MIL1_RXAN	MIL2_RXAN	–	TMRC_VAL
84	137	PB21	CADDR[7]	DATA[21]	–	CAN1_TX	UART1_TX	SSP3_FS	TMR0_CH2N	TMR0_CH2N	TMR2_ETR	MIL1_RXAP	MIL2_RXAP	–	TMRC_DATA
85	138	PB20	CADDR[8]	DATA[22]	–	CAN0_RX	UART0_RX	SSP3_CLK	TMR0_CH2P	TMR0_CH3P	TMR1_BRK	MIL1_ENA	MIL2_ENA	–	TMRC_CLK
86	139	PB19	CADDR[9]	DATA[23]	–	CAN0_TX	UART0_TX	SSP3_TX	TMR0_CH1N	TMR0_CH3N	TMR1_CH3N	MIL1_TXAP	MIL2_TXAP	–	–
87	140	PB18	CADDR [10]	DATA[24]	–	CAN1_RX	UART1_RX	SSP2_RX	TMR0_CH1P	TMR0_BRK	TMR1_CH3P	MIL1_TXAN	MIL2_TXAN	–	SSP5_RX
88	141	PB17	CADDR [11]	DATA[25]	–	CAN1_TX	UART1_TX	SSP2_FS	TMR0_CH0N	TMR1_ETR	TMR1_CH2N	MIL0_RXBN	MIL3_RXBN	–	SSP5_FS
89	142	PB16	CADDR [12]	DATA[26]	–	CAN0_RX	UART0_RX	SSP2_CLK	TMR0_CH0P	TMR1_CH0P	TMR1_CH2P	MIL0_RXBP	MIL3_RXBP	–	SSP5_CLK
90	143	PB15	CADDR [13]	DATA[27]	–	CAN0_TX	UART0_TX	SSP2_TX	TMR0_ETR	TMR1_CH0N	TMR1_CH1N	MIL0_ENB	MIL3_ENB	–	SSP5_TX
91	144	PB14	CADDR [14]	DATA[28]	–	CAN0_RX	UART1_RX	SSP1_RX	TMR3_BRK	TMR1_CH1P	TMR1_CH1P	MIL0_TXBP	MIL3_TXBP	–	TMTD_VAL
92	145	PB13	CADDR [15]	DATA[29]	–	CAN0_TX	UART1_TX	SSP1_FS	TMR3_CH3N	TMR1_CH1N	TMR1_CH0N	MIL0_TXBN	MIL3_TXBN	–	TMTD_DATA
93	146	PB12	CADDR [16]	DATA[30]	–	CAN1_RX	UART0_RX	SSP1_CLK	TMR3_CH3P	TMR1_CH2P	TMR1_CH0P	MIL0_RXAN	MIL3_RXAN	–	TMTD_CLK

№ вы-вода корпуса	№ КП кри-стал-ла	Обозна-чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
94	147	PB11	CADDR [17]	DATA[31]	–	CAN1_TX	UART0_TX	SSP1_TX	TMR3_CH2N	TMR1_CH2N	TMR1_ETR	MIL0_RXAP	MIL3_RXAP	–	–
95	148	PB10	CDATA [31]	–	CLOCK	CAN0_RX	UART1_RX	SSP0_RX	TMR3_CH2P	TMR1_CH3P	TMR0_BRK	MIL0_ENA	MIL3_ENA	–	SSP4_RX
96	149	PB9	CDATA [30]	–	OCLK	CAN0_TX	UART1_TX	SSP0_FS	TMR3_CH1N	TMR1_CH3N	TMR0_CH3N	MIL0_TXAP	MIL3_TXAP	–	SSP4_FS
97	150	PB8	CDATA [29]	–	–	CAN1_RX	UART0_RX	SSP0_CLK	TMR3_CH1P	TMR1_BRK	TMR0_CH3P	MIL0_TXAN	MIL3_TXAN	–	SSP4_CLK
98	151	PB7	CDATA [28]	–	–	CAN1_TX	UART0_TX	SSP0_TX	TMR3_CH0N	TMR0_ETR	TMR0_CH2N	MIL1_RXBN	MIL2_RXBN	SPW1_RXS	SSP4_TX
99	152	PB6	CDATA [27]	–	–	CAN0_RX	UART1_RX	SSP3_RX	TMR3_CH0P	TMR0_CH0P	TMR0_CH2P	MIL1_RXBP	MIL2_RXBP	SPW1_RXD	TMRC_VAL
100	153	PB5	CDATA [26]	–	–	CAN0_TX	UART1_TX	SSP3_FS	TMR3_ETR	TMR0_CH0N	TMR0_CH1N	MIL1_ENB	MIL2_ENB	SPW1_TXS	TMRC_DATA
101	154	PB4	CDATA [25]	–	–	CAN0_RX	UART0_RX	SSP3_CLK	TMR2_BRK	TMR0_CH1P	TMR0_CH1P	MIL1_TXBP	MIL2_TXBP	SPW1_TXD	TMRC_CLK
102	155	PB3	CDATA [24]	–	–	CAN0_TX	UART0_TX	SSP3_TX	TMR2_CH3N	TMR0_CH1N	TMR0_CH0N	MIL1_TXBN	MIL2_TXBN	SPW0_RXS	–
103	156	PB2	CDATA [23]	–	CSn[3]	CAN1_RX	UART1_RX	SSP2_RX	TMR2_CH3P	TMR0_CH2P	TMR0_CH0P	MIL1_RXAN	MIL2_RXAN	SPW0_RXD	SSP5_RX
104	157	PB1	CDATA [22]	–	CSn[2]	CAN1_TX	UART1_TX	SSP2_FS	TMR2_CH2N	TMR0_CH2N	TMR0_ETR	MIL1_RXAP	MIL2_RXAP	SPW0_TXS	SSP5_FS
105	158	PB0	CDATA [21]	–	CSn[1]	CAN0_RX	UART0_RX	SSP2_CLK	TMR2_CH2P	TMR0_CH3P	TMR1_BRK	MIL1_ENA	MIL2_ENA	SPW0_TXD	SSP5_CLK

№ вы-вода корпуса	№ КП кри-стал-ла	Обозна-чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
106	159	PA31	CDATA [20]	–	CSn[0]	CAN0_TX	UART0_TX	SSP2_TX	TMR2_CH1N	TMR0_CH3N	TMR1_CH3N	MIL1_TXAP	MIL2_TXAP	–	SSP5_TX
107	160	PA30	CNBE[3]	–	READY[7]	CAN1_RX	UART1_RX	SSP1_RX	TMR2_CH1P	TMR0_BRK	TMR1_CH3P	MIL1_TXAN	MIL2_TXAN	–	TMTD_VAL
108	161	PA29	CNBE[2]	–	READY[6]	CAN1_TX	UART1_TX	SSP1_FS	TMR2_CH0N	TMR3_ETR	TMR1_CH2N	MIL0_RXBN	MIL3_RXBN	–	TMTD_DATA
109	162	PA28	CNBE[1]	–	READY[5]	CAN0_RX	UART0_RX	SSP1_CLK	TMR2_CH0P	TMR3_CH0P	TMR1_CH2P	MIL0_RXBP	MIL3_RXBP	–	TMTD_CLK
124	173	PA27	CDATA [19]	–	READY[4]	CAN0_TX	UART0_TX	SSP1_TX	TMR2_ETR	TMR3_CH0N	TMR1_CH1N	MIL0_ENB	MIL3_ENB	–	–
125	174	PA26	CDATA [18]	–	READY[3]	CAN0_RX	UART1_RX	SSP0_RX	TMR1_BRK	TMR3_CH1P	TMR1_CH1P	MIL0_TXBP	MIL3_TXBP	–	SSP4_RX
126	175	PA25	CDATA [17]	–	READY[2]	CAN0_TX	UART1_TX	SSP0_FS	TMR1_CH3N	TMR3_CH1N	TMR1_CH0N	MIL0_TXBN	MIL3_TXBN	–	SSP4_FS
127	176	PA24	CDATA [16]	–	–	CAN1_RX	UART0_RX	SSP0_CLK	TMR1_CH3P	TMR3_CH2P	TMR1_CH0P	MIL0_RXAN	MIL3_RXAN	–	SSP4_CLK
128	177	PA23	CNBE[0]	–	–	CAN1_TX	UART0_TX	SSP0_TX	TMR1_CH2N	TMR3_CH2N	TMR1_ETR	MIL0_RXAP	MIL3_RXAP	–	SSP4_TX
129	178	PA22	CADDR [18]	–	–	CAN0_RX	UART1_RX	SSP3_RX	TMR1_CH2P	TMR3_CH3P	TMR0_BRK	MIL0_ENA	MIL3_ENA	–	TMRC_VAL
130	179	PA21	CADDR [19]	–	–	CAN0_TX	UART1_TX	SSP3_FS	TMR1_CH1N	TMR3_CH3N	TMR0_CH3N	MIL0_TXAP	MIL3_TXAP	–	TMRC_DATA
159	207	PA20	CADDR [20]	–	–	CAN1_RX	UART0_RX	SSP3_CLK	TMR1_CH1P	TMR3_BRK	TMR0_CH3P	MIL0_TXAN	MIL3_TXAN	–	TMRC_CLK
160	208	PA19	CADDR [21]	–	–	CAN1_TX	UART0_TX	SSP3_TX	TMR1_CH0N	TMR2_ETR	TMR0_CH2N	MIL1_RXBN	MIL2_RXBN	–	–
161	209	PA18	CADDR [22]	READY[0]	READY[2]	CAN0_RX	UART1_RX	SSP2_RX	TMR1_CH0P	TMR2_CH0P	TMR0_CH2P	MIL1_RXBP	MIL2_RXBP	–	SSP5_RX

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
162	210	PA17	CADDR [23]	READY[1]	READY[3]	CAN0_TX	UART1_ TX	SSP2_FS	TMR1_ ETR	TMR2_ CH0N	TMR0_ CH1N	MIL1_ ENB	MIL2_ ENB	–	SSP5_FS
163	211	PA16	CADDR [24]	READY[2]	–	CAN0_RX	UART0_ RX	SSP2_ CLK	TMR0_ BRK	TMR2_ CH1P	TMR0_ CH1P	MIL1_ TXBP	MIL2_ TXBP	–	SSP5_ CLK
164	212	PA15	CADDR [25]	READY[3]	–	CAN0_TX	UART0_ TX	SSP2_TX	TMR0_ CH3N	TMR2_ CH1N	TMR0_ CH0N	MIL1_ TXBN	MIL2_ TXBN	–	SSP5_TX
165	213	PA14	CADDR [26]	READY[4]	–	CAN1_RX	UART1_ RX	SSP1_RX	TMR0_ CH3P	TMR2_ CH2P	TMR0_ CH0P	MIL1_ RXAN	MIL2_ RXAN	–	TMTD_ VAL
166	214	PA13	CADDR [27]	READY[5]	–	CAN1_TX	UART1_ TX	SSP1_FS	TMR0_ CH2N	TMR2_ CH2N	TMR0_ ETR	MIL1_ RXAP	MIL2_ RXAP	–	TMTD_ DATA
167	215	PA12	CADDR [28]	READY[6]	CLOCK	CAN0_RX	UART0_ RX	SSP1_ CLK	TMR0_ CH2P	TMR2_ CH3P	TMR1_ BRK	MIL1_ ENA	MIL2_ ENA	–	TMTD_ CLK
168	216	PA11	CADDR [29]	READY[7]	OCLK	CAN0_TX	UART0_ TX	SSP1_TX	TMR0_ CH1N	TMR2_ CH3N	TMR1_ CH3N	MIL1_ TXAP	MIL2_ TXAP	–	–
169	217	PA10	CADDR [30]	CSn[0]	–	CAN1_RX	UART1_ RX	SSP0_RX	TMR0_ CH1P	TMR2_ BRK	TMR1_ CH3P	MIL1_ TXAN	MIL2_ TXAN	–	SSP4_RX
170	218	PA9	CADDR [31]	CSn[1]	–	CAN1_TX	UART1_ TX	SSP0_FS	TMR0_ CH0N	TMR1_ ETR	TMR1_ CH2N	MIL0_ RXBN	MIL3_ RXBN	–	SSP4_FS
171	219	PA8	CNWE	CSn[2]	–	CAN0_RX	UART0_ RX	SSP0_ CLK	TMR0_ CH0P	TMR1_ CH0P	TMR1_ CH2P	MIL0_ RXBP	MIL3_ RXBP	–	SSP4_ CLK
172	220	PA7	CNOE	CSn[3]	–	CAN0_TX	UART0_ TX	SSP0_TX	TMR0_ ETR	TMR1_ CH0N	TMR1_ CH1N	MIL0_ ENB	MIL3_ ENB	–	SSP4_TX
173	221	PA6	CADDR[7]	CSn[4]	–	CAN0_RX	UART1_ RX	SSP3_RX	TMR3_ BRK	TMR1_ CH1P	TMR1_ CH1P	MIL0_ TXBP	MIL3_ TXBP	–	TMRC_ VAL
174	222	PA5	CADDR[6]	DATA[47]	–	CAN0_TX	UART1_ TX	SSP3_FS	TMR3_ CH3N	TMR1_ CH1N	TMR1_ CH0N	MIL0_ TXBN	MIL3_ TXBN	–	TMRC_ DATA
175	223	PA4	CNCS	DATA[46]	–	CAN1_RX	UART0_ RX	SSP3_ CLK	TMR3_ CH3P	TMR1_ CH2P	TMR1_ CH0P	MIL0_ RXAN	MIL3_ RXAN	–	TMRC_ CLK
176	224	PA3	CNCS_ CFG	DATA[45]	–	CAN1_TX	UART0_ TX	SSP3_TX	TMR3_ CH2N	TMR1_ CH2N	TMR1_ ETR	MIL0_ RXAP	MIL3_ RXAP	–	–

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
177	225	PA2	CADDR[0]	DATA[44]	–	CAN0_RX	UART1_ RX	SSP2_RX	TMR3_ CH2P	TMR1_ CH3P	TMR0_ BRK	MIL0_ ENA	MIL3_ ENA	–	SSP5_RX
178	226	PA1	CADDR[1]	DATA[43]	–	CAN0_TX	UART1_ TX	SSP2_FS	TMR3_ CH1N	TMR1_ CH3N	TMR0_ CH3N	MIL0_ TXAP	MIL3_ TXAP	–	SSP5_FS
179	227	PA0	CREADY	DATA[42]	–	CAN1_RX	UART0_ RX	SSP2_ CLK	TMR3_ CH1P	TMR1_ BRK	TMR0_ CH3P	MIL0_ TXAN	MIL3_ TXAN	–	SSP5_ CLK
185	3	PE31	CDATA[7]	DATA[41]	CSn[4]	CAN1_TX	UART0_ TX	SSP2_TX	TMR3_ CH0N	TMR0_ ETR	TMR0_ CH2N	MIL1_ RXBN	MIL2_ RXBN	–	SSP5_TX
186	4	PE30	CDATA[6]	DATA[40]	CSn[5]	CAN0_RX	UART1_ RX	SSP1_RX	TMR3_ CH0P	TMR0_ CH0P	TMR0_ CH2P	MIL1_ RXBP	MIL2_ RXBP	–	TMTD_ CLK
187	5	PE29	CDATA[5]	DATA[39]	CSn[6]	CAN0_TX	UART1_ TX	SSP1_FS	TMR3_ ETR	TMR0_ CH0N	TMR0_ CH1N	MIL1_ ENB	MIL2_ ENB	–	TMTD_ DATA
188	6	PE28	CDATA[4]	DATA[38]	CSn[7]	CAN0_RX	UART0_ RX	SSP1_ CLK	TMR2_ BRK	TMR0_ CH1P	TMR0_ CH1P	MIL1_ TXBP	MIL2_ TXBP	–	TMTD_ VAL
189	7	PE27	CDATA[3]	DATA[37]	ADDR[31]	CAN0_TX	UART0_ TX	SSP1_TX	TMR2_ CH3N	TMR0_ CH1N	TMR0_ CH0N	MIL1_ TXBN	MIL2_ TXBN	–	–
190	8	PE26	CDATA[2]	DATA[36]	ADDR[30]	CAN1_RX	UART1_ RX	SSP0_RX	TMR2_ CH3P	TMR0_ CH2P	TMR0_ CH0P	MIL1_ RXAN	MIL2_ RXAN	–	SSP4_RX
191	9	PE25	CDATA[1]	DATA[35]	ADDR[29]	CAN1_TX	UART1_ TX	SSP0_FS	TMR2_ CH2N	TMR0_ CH2N	TMR0_ ETR	MIL1_ RXAP	MIL2_ RXAP	–	SSP4_FS
192	10	PE24	CDATA[0]	DATA[34]	ADDR[28]	CAN0_RX	UART0_ RX	SSP0_ CLK	TMR2_ CH2P	TMR0_ CH3P	TMR1_ BRK	MIL1_ ENA	MIL2_ ENA	–	SSP4_ CLK
197	16	PE23	CDATA[8]	DATA[33]	ADDR[27]	CAN0_TX	UART0_ TX	SSP0_TX	TMR2_ CH1N	TMR0_ CH3N	TMR1_ CH3N	MIL1_ TXAP	MIL2_ TXAP	–	SSP4_TX
198	17	PE22	CDATA[9]	DATA[32]	ADDR[26]	CAN1_RX	UART1_ RX	SSP3_RX	TMR2_ CH1P	TMR0_ BRK	TMR1_ CH3P	MIL1_ TXAN	MIL2_ TXAN	–	TMTD_ CLK
199	18	PE21	CDATA [10]	–	ADDR[25]	CAN1_TX	UART1_ TX	SSP3_FS	TMR2_ CH0N	TMR3_ ETR	TMR1_ CH2N	MIL0_ RXBN	MIL3_ RXBN	–	TMTD_ DATA

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
200	19	PE20	CDATA [11]	–	ADDR[24]	CAN0_RX	UART0_ RX	SSP3_ CLK	TMR2_ CH0P	TMR3_ CH0P	TMR1_ CH2P	MIL0_ RXBP	MIL3_ RXBP	–	TMTD_ VAL
201	20	PE19	CDATA [12]	–	ADDR[23]	CAN0_TX	UART0_ TX	SSP3_TX	TMR2_ ETR	TMR3_ CH0N	TMR1_ CH1N	MIL0_ ENB	MIL3_ ENB	–	–
202	21	PE18	CDATA [13]	–	ADDR[22]	CAN0_RX	UART1_ RX	SSP2_RX	TMR1_ BRK	TMR3_ CH1P	TMR1_ CH1P	MIL0_ TXBP	MIL3_ TXBP	–	SSP5_RX
203	22	PE17	CDATA [14]	–	ADDR[21]	CAN0_TX	UART1_ TX	SSP2_FS	TMR1_ CH3N	TMR3_ CH1N	TMR1_ CH0N	MIL0_ TXBN	MIL3_ TXBN	–	SSP5_FS
204	23	PE16	CDATA [15]	–	CLOCK	CAN1_RX	UART0_ RX	SSP2_ CLK	TMR1_ CH3P	TMR3_ CH2P	TMR1_ CH0P	MIL0_ RXAN	MIL3_ RXAN	–	SSP5_ CLK
205	24	PE15	–	–	OCLK	CAN1_TX	UART0_ TX	SSP2_TX	TMR1_ CH2N	TMR3_ CH2N	TMR1_ ETR	MIL0_ RXAP	MIL3_ RXAP	–	SSP5_TX
206	25	PE14	–	–	–	CAN0_RX	UART1_ RX	SSP1_RX	TMR1_ CH2P	TMR3_ CH3P	TMR0_ BRK	MIL0_ ENA	MIL3_ ENA	–	–
207	26	PE13	–	DATA[15]	–	CAN0_TX	UART1_ TX	SSP1_FS	TMR1_ CH1N	TMR3_ CH3N	TMR0_ CH3N	MIL0_ TXAP	MIL3_ TXAP	–	TMRC_ CLK
208	27	PE12	–	DATA[14]	–	CAN1_RX	UART0_ RX	SSP1_ CLK	TMR1_ CH1P	TMR3_ BRK	TMR0_ CH3P	MIL0_ TXAN	MIL3_ TXAN	–	TMRC_ DATA
209	28	PE11	–	DATA[13]	–	CAN1_TX	UART0_ TX	SSP1_TX	TMR1_ CH0N	TMR2_ ETR	TMR0_ CH2N	MIL1_ RXBN	MIL2_ RXBN	–	TMRC_ VAL
210	29	PE10	–	DATA[12]	–	CAN0_RX	UART1_ RX	SSP0_RX	TMR1_ CH0P	TMR2_ CH0P	TMR0_ CH2P	MIL1_ RXBP	MIL2_ RXBP	–	SSP4_RX
211	30	PE9	–	DATA[11]	–	CAN0_TX	UART1_ TX	SSP0_FS	TMR1_ ETR	TMR2_ CH0N	TMR0_ CH1N	MIL1_ ENB	MIL2_ ENB	–	SSP4_FS
212	31	PE8	–	DATA[10]	–	CAN0_RX	UART0_ RX	SSP0_ CLK	TMR0_ BRK	TMR2_ CH1P	TMR0_ CH1P	MIL1_ TXBP	MIL2_ TXBP	–	SSP4_ CLK
213	32	PE7	–	DATA[9]	–	CAN0_TX	UART0_ TX	SSP0_TX	TMR0_ CH3N	TMR2_ CH1N	TMR0_ CH0N	MIL1_ TXBN	MIL2_ TXBN	–	SSP4_TX
214	33	PE6	–	DATA[8]	–	CAN1_RX	UART1_ RX	SSP3_RX	TMR0_ CH3P	TMR2_ CH2P	TMR0_ CH0P	MIL1_ RXAN	MIL2_ RXAN	–	–
215	34	PE5	–	DATA[7]	–	CAN1_TX	UART1_ TX	SSP3_FS	TMR0_ CH2N	TMR2_ CH2N	TMR0_ ETR	MIL1_ RXAP	MIL2_ RXAP	–	TMRC_ CLK

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
216	35	PE4	–	DATA[6]	–	CAN0_RX	UART0_RX	SSP3_CLK	TMR0_CH2P	TMR2_CH3P	TMR3_BRK	MIL1_ENA	MIL2_ENA	–	TMRC_DATA
217	36	PE3	–	DATA[5]	–	CAN0_TX	UART0_TX	SSP3_TX	TMR0_CH1N	TMR2_CH3N	TMR3_CH3N	MIL1_TXAP	MIL2_TXAP	–	TMRC_VAL
218	37	PE2	–	DATA[4]	–	CAN1_RX	UART1_RX	SSP2_RX	TMR0_CH1P	TMR2_BRK	TMR3_CH3P	MIL1_TXAN	MIL2_TXAN	–	SSP5_RX
219	38	PE1	–	DATA[3]	–	CAN1_TX	UART1_TX	SSP2_FS	TMR0_CH0N	TMR1_ETR	TMR3_CH2N	MIL0_RXBN	MIL3_RXBN	–	SSP5_FS
220	39	PE0	–	DATA[2]	–	CAN0_RX	UART0_RX	SSP2_CLK	TMR0_CH0P	TMR1_CH0P	TMR3_CH2P	MIL0_RXBP	MIL3_RXBP	–	SSP5_CLK
221	40	PD31	–	DATA[1]	CLOCK	CAN0_TX	UART0_TX	SSP2_TX	TMR0_ETR	TMR1_CH0N	TMR3_CH1N	MIL0_ENB	MIL3_ENB	–	SSP5_TX
222	41	PD30	–	DATA[0]	OCLK	CAN0_RX	UART1_RX	SSP1_RX	TMR1_BRK	TMR1_CH1P	TMR3_CH1P	MIL0_TXBP	MIL3_TXBP	–	TMTD_VAL
227	46	PD29	–	BWEn[2]	BEn[2]	CAN0_TX	UART1_TX	SSP1_FS	TMR1_CH3N	TMR1_CH1N	TMR3_CH0N	MIL0_TXBN	MIL3_TXBN		TMTD_DATA
228	47	PD28	–	BWEn[5]	BEn[5]	CAN1_RX	UART0_RX	SSP1_CLK	TMR1_CH3P	TMR1_CH2P	TMR3_CH0P	MIL0_RXAN	MIL3_RXAN		TMTD_CLK
229	48	PD27		BWEn[4]	BEn[4]	CAN1_TX	UART0_TX	SSP1_TX	TMR1_CH2N	TMR1_CH2N	TMR3_ETR	MIL0_RXAP	MIL3_RXAP		
230	49	PD26		BWEn[1]	BEn[1]	CAN0_RX	UART1_RX	SSP0_RX	TMR1_CH2P	TMR1_CH3P	TMR2_BRK	MIL0_ENA	MIL3_ENA		SSP4_RX
231	50	PD25	–	BWEn[0]	BEn[0]	CAN0_TX	UART1_TX	SSP0_FS	TMR1_CH1N	TMR1_CH3N	TMR2_CH3N	MIL0_TXAP	MIL3_TXAP	–	SSP4_FS
232	51	PD24	–	WEn[0]	–	CAN1_RX	UART0_RX	SSP0_CLK	TMR1_CH1P	TMR1_BRK	TMR2_CH3P	MIL0_TXAN	MIL3_TXAN	–	SSP4_CLK
234	53	PD23	–	OEn[0]	–	CAN1_TX	UART0_TX	SSP0_TX	TMR1_CH0N	TMR0_ETR	TMR2_CH2N	MIL1_RXBN	MIL2_RXBN	–	SSP4_TX
235	54	PD22	–	CS[3]	CLOCK	CAN0_RX	UART1_RX	SSP3_RX	TMR1_CH0P	TMR0_CH0P	TMR2_CH2P	MIL1_RXBP	MIL2_RXBP	–	–
236	55	PD21	–	CS[2]	OCLK	CAN0_TX	UART1_TX	SSP3_FS	TMR1_ETR	TMR0_CH0N	TMR2_CH1N	MIL1_ENB	MIL2_ENB	–	TMRC_DATA

№ вы- вода кор- пуса	№ КП кри- стал- ла	Обозна- чение вывода	Дополнительное назначение вывода												
			1	2	3	4	5	6	7	8	9	12	13	14	15
237	56	PD20	–	CS[1]	–	CAN0_RX	UART0_ RX	SSP3_ CLK	TMR0_ BRK	TMR0_ CH1P	TMR2_ CH1P	MIL1_ TXBP	MIL2_ TXBP	–	TMRC_ VAL
238	57	PD19	–	CS[0]	–	CAN0_TX	UART0_ TX	SSP3_TX	TMR0_ CH3N	TMR0_ CH1N	TMR2_ CH0N	MIL1_ TXBN	MIL2_ TXBN	–	TMRC_ CLK
239	58	PD18	–	ADDR[20]	–	CAN1_RX	UART1_ RX	SSP2_RX	TMR0_ CH3P	TMR0_ CH2P	TMR2_ CH0P	MIL1_ RXAN	MIL2_ RXAN	–	SSP5_RX

4 Указания по применению и эксплуатации

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

Типовая схема включения микросхем приведена на рисунке 150.

Вывод 1 корпуса микросхем электрически соединен с обратной стороной кристалла. Рекомендуется вывод 1 соединять с шиной «Общий».

Вывод TEST1 подключать к U_{CC} .

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

- подача (включение микросхем) – «Общий», напряжение источника питания (U_{CC}), напряжение питания SpaceWire (U_{CC_SPW}), входное напряжение встроенного регулятора LDO (U_{I_LDO}), входные сигналы или одновременно;

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

5 Архитектура контроллера

5.1 Питание

5.1.1 Домены питания

Микросхема имеет несколько доменов питания, описание которых приведено в таблице 4.

Таблица 4 – Домены питания

Обозначение	Допустимые значения, В		Описание
	не менее	не более	
Ucc	3,0	5,5	Основное питание микросхемы. Используется для питания портов ввода-вывода
Ucc1	2,5	5,5	Входное питание встроенных LDO регуляторов, формирующих питание для цифровой части. На вход питания Ucc1 может подаваться основное питание Ucc или питание ниже, чем Ucc, но не менее 2,5 В и не более Ucc
SPWUcc	3,0	5,5	Питание физических приемопередатчиков SpaceWire. Должно совпадать с питанием Ucc ($\pm 0,2$ В). Используется для снижения шумов по Ucc
DUcc_PLL	1,62	1,98	Питание, сформированное с помощью внутреннего LDO регулятора для питания батарейного домена. Входным напряжением LDO регулятора является Ucc1
DUcc	1,62	1,98	Питание основной цифровой части, сформированное с помощью внутреннего LDO регулятора

5.1.2 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 Ucc не превысил значение U_{POR} , сигнал сброса UccRESET будет нулевым (сброс). После превышения Ucc уровня U_{POR} сигнал UccRESET в течение времени $T_{PORESETn}$ удерживается в нулевом состоянии, затем переключается в единицу. Задержка $T_{PORESETn}$ формируется на счетчике, работающем с тактовой частотой от генератора LSI. Если по каким-либо причинам произошел отказ генератора LSI, схема сформирует сигнал UccRESET с задержкой много меньшей, чем $T_{PORESETn}$. Кроме того, схема POR контролирует состояния CLAMP (шунтирующие диоды защиты от статического заряда) по питаниям Ucc, которые при резком включении питания кратковременно закорачивают питание Ucc на GND.

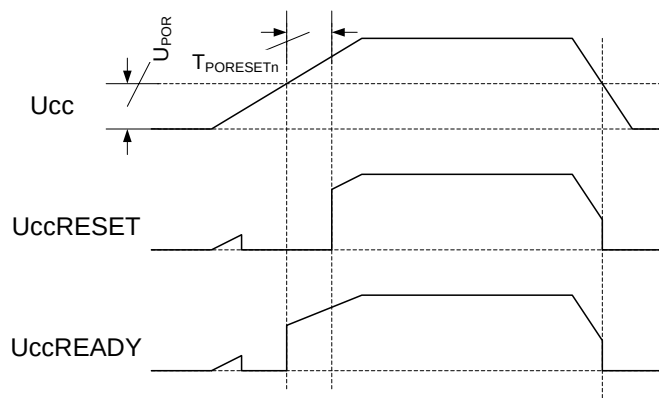


Рисунок 6 – Сигнал сброса при включении основного питания или просадке его ниже критического уровня

При снижении уровня U_{cc} ниже значения U_{POR} сигнал $U_{ccRESET}$ вырабатывается с минимальной задержкой.

5.1.3 Схема защиты от перенапряжения POVR

При превышении напряжением питания U_{cc} уровня U_{OVER} (не менее 5,5 В и не более 5,7 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 блока PWR_BKP и может быть проанализирован при последующем запуске системы.

5.1.4 Перекрывание рабочих диапазонов питания схемами POR и POVR

Чтобы исключить возможность продолжения работы схемы в диапазоне ниже или выше допустимого уровня напряжения питания, обеспечивается перекрывание рабочих диапазонов питания и уровней срабатывания схем POR и POVR. При отбракованном 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.

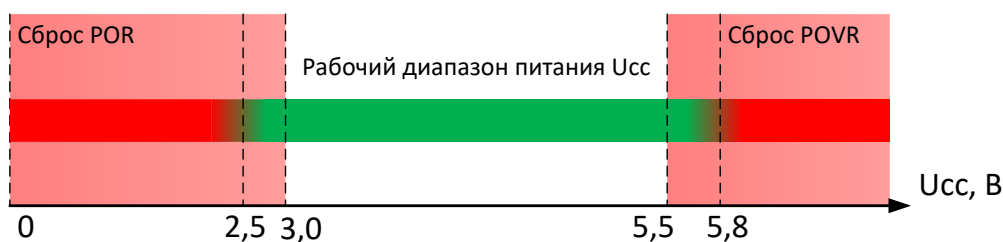


Рисунок 7 – Рабочий диапазон питания

5.1.5 Блок монитора основного питания (Ucc)

Для определения абсолютных значений уровня напряжения питания Ucc используется блок PVD, обеспечивающий определение уровня напряжения с 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.

5.1.6 Управление питанием доменов DUcc

Микроконтроллер содержит два домена питания DUcc (см. таблицу 5), которые включают в себя основные блоки микроконтроллера.

Т а б л и ц а 5 – Домены питания DUcc

Название	Предельно-допустимые значения, В		Описание
	не менее	не более	
DUcc_PLL	1,62	1,98	Питание, сформированное с помощью внутреннего LDO регулятора для блока умножителей частоты и генераторов
DUcc	1,62	1,98	Питание, сформированное с помощью внутреннего LDO регулятора для основной цифровой части (процессорное ядро, память, периферия)

Управление питанием доменов осуществляется в регистрах блока PWR_BKP.

Для каждого домена реализован встроенный регулятор напряжения DUcc. Также может быть отключены все регуляторы питания DUcc сигналом SHDN, при этом питание должно подаваться извне от внешнего источника. Биты управления питанием реализованы в блоке PWR_BKP.

5.1.7 Блок монитора тока потребления доменов DUcc

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 входом CUREN.

5.2 Тактовые частоты

Структурная схема формирования тактовых частот представлена на рисунке 8.

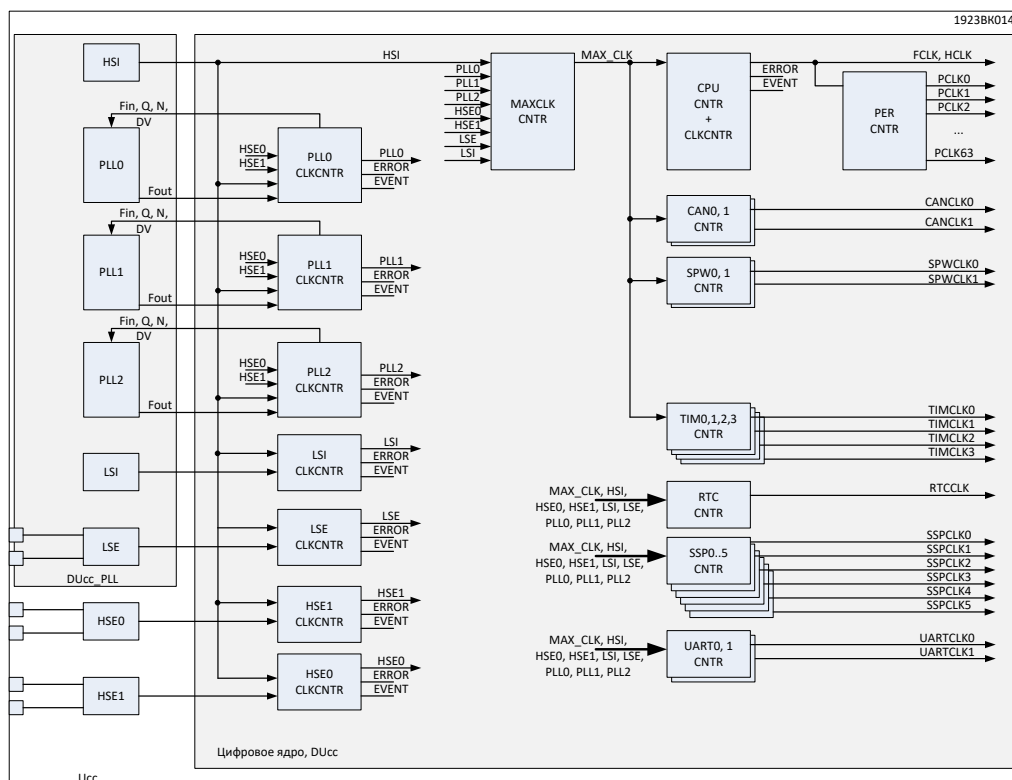


Рисунок 8 – Формирование тактовых частот

5.2.1 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_{PORSTn} .

5.2.2 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.

5.2.3 Внешний высокоскоростной генератор HSE0 и HSE1

Для формирования внешней стабильной высокой частоты (от 1 до 30 МГц) 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 HSI-генератор и программное восстановление работы системы на оставшемся работающем генераторе.

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.

Генератор HSE имеет флаг готовности, который вырабатывается при включении генератора и выхода его в нормальный режим работы.

5.2.4 Внешний низкоскоростной часовой генератор LSE

Для формирования внешней стабильной низкой частоты (от 20 до 40 КГц) применяется генератор LSE. В микроконтроллере нет второго дублирующего генератора LSE. Отказ генератора LSE может быть обнаружен с помощью блока монитора тактовых частот. При отказе генератора может быть выполнен аварийный переход на генератор HSI и программное восстановление работы системы. Также генератор LSE имеет встроенные фильтры помех, исключающие прохождение коротких импульсов или «просечек». Генератор LSE имеет флаг готовности, который вырабатывается при включении генератора и выхода его в нормальный режим работы.

5.2.5 Блок умножения тактовой частоты PLL

Для формирования высоких тактовых частот (от 1 до 150 МГц) используются схемы умножения тактовых частот. В качестве источников опорной частоты для PLL выступают генераторы HSI и HSEx. С помощью PLL формируется частота

$$F_{OUT} = F_{in} \times \frac{N}{(Q + 1) \times (DV + 1)}. \quad (1)$$

На выходе PLL имеются встроенные фильтры помех, исключающие прохождение коротких импульсов или «просечек». PLL имеет флаг готовности, который вырабатывается при включении схемы умножителя и выходе её в нормальный режим работы. Смену коэффициентов PLL необходимо производить в выключенном состоянии. 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 HSI генератор и программно восстановить работу системы на других PLL или генераторах.

5.2.6 Монитор частоты внешних генераторов и PLL (CLKCNTR)

Блок монитора частоты внешних генераторов и PLL предназначен для определения значения тактовых частот по отношению друг к другу, определения следующие события 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 микроконтроллера 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 MAX_CLK также

реализован блок монитора частоты, но он не имеет возможности аппаратного переключения на частоту HSI, и позволяет только обнаруживать факты сбоя или отказа.

5.3 Сигналы Сброса

Структурная схема формирования сигналов сброса представлена на рисунке 9.

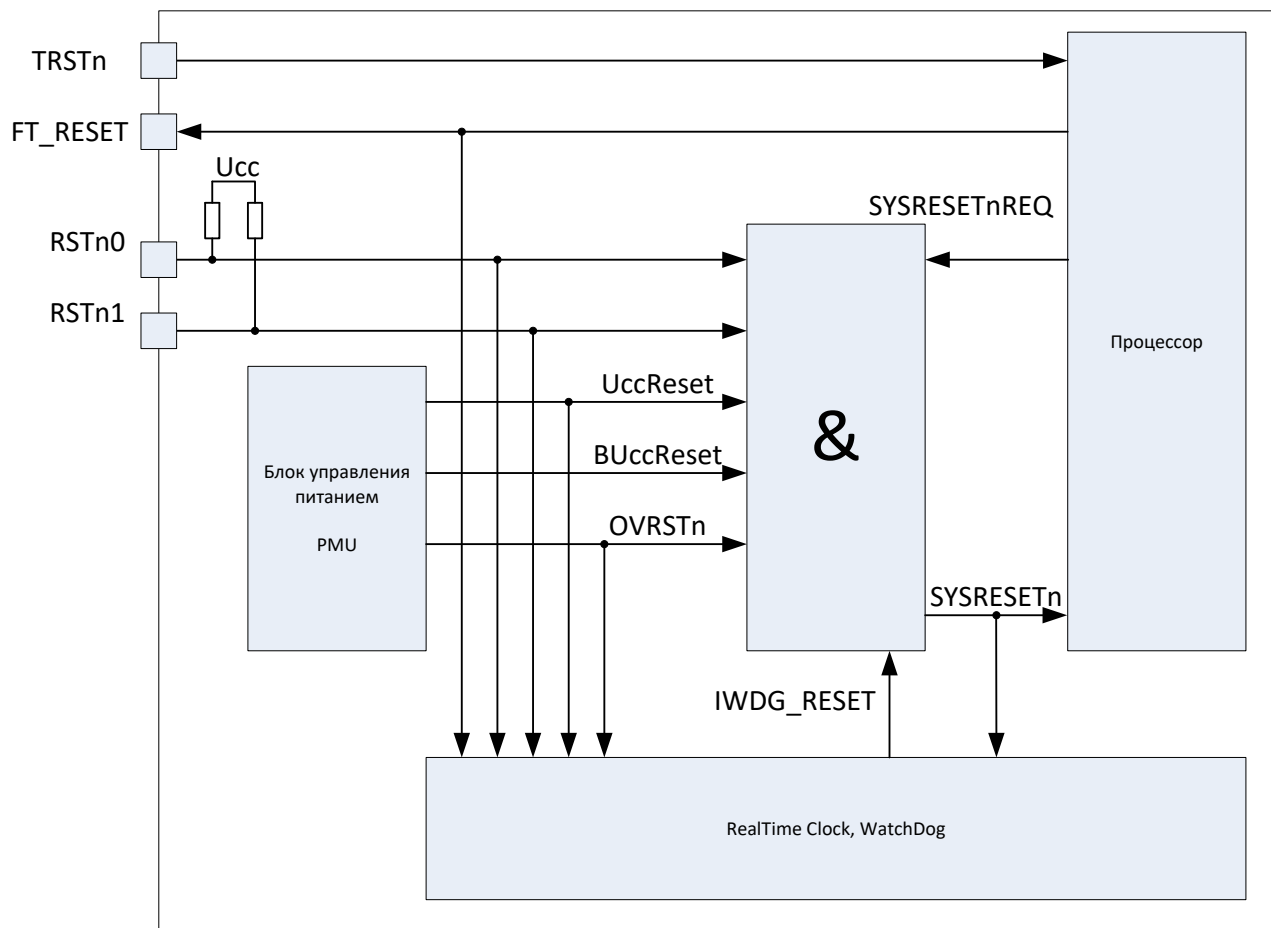


Рисунок 9 – Формирование сигналов сброса

5.3.1 Сигналы внешнего сброса nRESET0 и nRESET1

Микросхема может быть сброшена внешними сигналами с выводов nRESET0 и nRESET1. Сброс осуществляется при одновременной подаче низкого уровня на оба входа. Входы nRESET0 и nRESET1 имеет встроенную подтяжку к питанию с эквивалентным сопротивлением 50 кОм. Факт появления каждого сигнала nRESET0 и nRESET1 фиксируется в батарейном домене.

5.3.2 Сигнал сброса по питанию UccRESET

Начиная с включения питания Ucc и до превышения уровня U_{POR} в течение времени $T_{PORESETn}$, а также при снижении Ucc ниже уровня U_{POR} вырабатывается сигнал сброса UccRESET. Факт появления сигнала UccRESET фиксируется в батарейном домене.

5.3.3 Сигнал сброса по питанию DUccRESET

Домены цифрового питания DUcc сбрасываются сигналом готовности DUCCREADY при отключении регуляторов. При включении регуляторов сигнал DUCCREADY доопределяет в ноль сигнал SYSRESETn.

5.3.4 Сигнал программного запроса сброса **SYS_RESET_REQ**

Микросхема может быть сброшена программным запросом со стороны процессорного ядра.

5.3.5 Сигналы аварийного сброса **FT_RESET**

Микросхема может быть сброшена блоком управления сбоями при обнаружении различных событий, также ряд событий всегда вызывает сброс микросхемы. При этом факт появления сброса от внутреннего сигнала **FT_RESETn** фиксируется в батарейном домене в бите **S_FT_RESET**.

5.3.6 Сигналы сброса сторожевого таймера **IWDG_RESET**

Микросхема может быть сброшена сторожевым таймером **IWDG**. Факт появления сигнала **IWDG_RESET** фиксируется в регистрах блока **PWR_BKP**.

5.3.7 Сигнал сброса по перенапряжению **POVR**

При превышении питанием **Ucc** 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 микроконтроллер может быть аппаратно сброшен.

Схема **POVR** при проведении отбраковочного тестирования может быть заблокирована сигналом **EXTPOVR**. В этом случае сигнал **OVRSTn** всегда равен **Ucc**, 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 блока **PWR_BKP** и может быть проанализирован при последующем запуске системы.

5.3.8 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 **UccRESET**.

5.4 Помехозащищенное кодирование

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

- кодом Хемминга (72,64) – для внутренних памятей и памяти на внешней системной шине;
- кодом Хемминга (7,4) – для задания режима работы.

Код Хемминга (7,4) применяется для задания режима работы микроконтроллера, подробнее см. подраздел 5.5 «Режимы работы микроконтроллера».

Кодом Хемминга (72,64) всегда защищаются внутренние памяти, включая кэш-память. ECC кодирование 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

ЕСС_Т. 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).

Также контроллер внешней шины можно перевести в режим DEC-TED кодирования (исправления двойных и обнаружения тройных ошибок). Для кодирования используется специализированный 80-64 BCH-код, H-матрица которого представлена в таблице 6.

5.4.1 ECC кодирование 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 матрицей ПЗУ. После чего, корректные данные с контрольной суммой передаются в процессорные ядра.

ЕСС биты загрузочной программы вычислены на этапе создания программы и зашиты в ПЗУ микроконтроллера на этапе разработки.

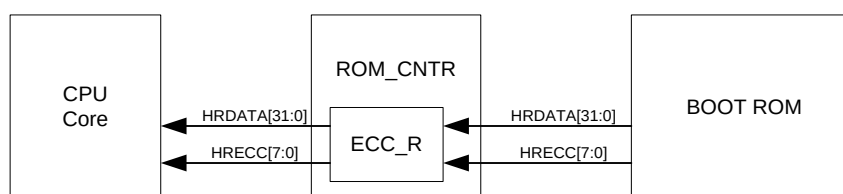


Рисунок 10 – Схема контроля ECC на пути от ПЗУ

Таблица 6 – H-матрица DEC-TED кода

78	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
76	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
75	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
74	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
73	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
72	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
71	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
70	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
69	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
68	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
67	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
66	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
65	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
64	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
63	1	0	0	0	1	0	0	0	1	1	1	0	1	1	1
62	1	1	0	1	0	0	1	1	0	0	0	1	0	0	1
61	1	1	1	0	1	1	0	0	0	1	0	0	1	0	0
60	1	1	1	1	1	1	1	0	1	0	0	0	1	1	1
59	1	1	0	0	1	0	0	1	0	1	1	1	1	1	0
58	1	1	0	1	1	0	0	0	1	0	1	1	0	1	1
57	1	1	1	0	1	1	1	1	0	0	1	0	1	1	1
56	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1
55	1	1	0	0	0	0	1	0	0	1	1	1	1	1	1
54	1	1	0	0	1	0	0	0	1	0	0	1	1	0	1
53	1	1	1	1	1	1	0	0	0	0	1	0	1	0	1
52	1	1	1	0	1	0	0	0	1	1	1	0	1	0	1
51	1	1	0	1	0	1	0	0	1	1	1	1	0	0	1
50	1	1	0	1	1	0	1	0	1	1	0	1	1	0	0
49	1	1	1	0	0	1	1	0	1	0	1	0	1	0	1
48	1	1	1	1	0	0	1	0	1	1	0	0	1	1	0
47	1	1	0	0	0	1	1	0	0	0	0	1	0	1	1
46	1	1	1	1	0	1	1	0	0	1	0	1	1	1	1
45	1	1	0	1	0	1	0	1	0	1	1	0	1	0	1
44	1	1	0	0	1	1	0	0	0	0	1	0	1	1	0
43	1	1	1	0	0	1	1	1	1	0	1	1	0	0	0
42	1	1	0	1	0	1	0	1	0	0	0	0	1	1	0
41	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1
40	1	1	1	0	0	1	1	1	0	0	0	1	1	0	1
39	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0
38	1	1	0	0	0	1	0	1	1	1	1	0	1	1	0
37	1	1	1	0	1	1	0	1	0	0	1	1	0	0	1
36	1	1	1	1	0	0	1	0	0	1	1	1	0	0	1
35	1	1	0	1	0	0	1	0	0	0	0	0	0	1	0
34	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
33	1	1	0	1	1	1	1	1	0	1	0	0	0	1	0
32	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
31	1	1	1	0	0	1	0	0	0	0	0	1	0	1	1
30	1	0	0	0	0	1	0	0	0	0	0	1	1	1	0
29	1	0	0	0	0	1	1	1	1	0	1	1	1	0	1
28	1	0	1	1	0	0	0	0	1	0	1	0	1	0	1
27	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0
26	1	0	0	0	0	1	0	0	1	1	1	1	0	1	0
25	1	0	0	0	1	1	1	1	1	0	0	1	0	0	0
24	1	0	1	1	1	0	0	0	0	0	1	0	0	1	1
23	1	0	1	1	1	0	1	1	1	1	1	0	1	0	1
22	1	0	1	1	1	0	0	1	1	1	0	0	1	1	0
21	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0
20	1	0	0	0	0	0	0	0	1	0	0	1	0	1	1
19	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1
18	1	0	1	0	0	0	0	0	1	1	0	0	0	0	1
17	1	0	1	0	0	0	0	0	0	1	0	0	1	1	0
16	1	0	0	1	0	0	0	1	1	1	1	1	1	1	0
15	1	0	0	1	1	0	0	1	1	1	0	1	1	0	1
14	1	0	0	1	0	1	1	0	1	0	0	0	1	0	1
13	1	0	0	0	1	0	0	1	0	1	0	0	0	0	1
12	1	0	1	1	0	1	0	1	1	1	0	1	0	0	1
11	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1
10	1	0	1	1	1	1	1	0	1	0	0	1	0	1	0
9	1	0	1	0	1	0	0	1	1	0	0	1	1	1	1
8	1	0	0	1	0	1	0	1	1	0	1	0	1	0	0
7	1	0	0	0	1	0	1	0	0	0	0	0	1	0	1
6	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0
5	1	0	0	1	0	1	1	1	0	1	0	0	0	0	1
4	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0
3	1	0	1	1	0	0	0	1	0	1	0	1	1	1	1
2	1	0	1	1	0	0	1	0	1	0	0	1	1	0	0
1	1	0	1	0	1	1	1	0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0

5.4.2 ECC кодирование для RAM

При обращении в область RAM генерация и проверка ECC происходит на границе контроллера памяти и непосредственно самой матрицы.

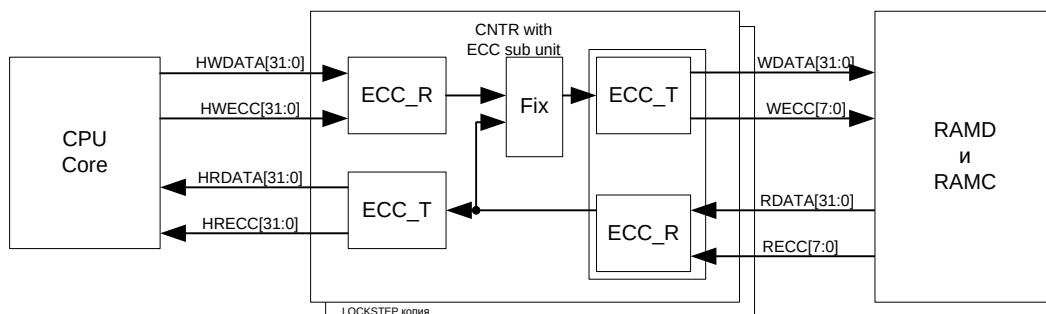


Рисунок 11 – Схема контроля ECC для RAM

В ряде случаев при записи в память сначала выполняется чтение, это связано с необходимостью обработки байтов и полуслов при работе с памятью, имеющей 32-битную организацию с защитой ECC. Так, например, при записи байта в RAMD необходимо считать из памяти ранее записанное слово с ECC, проверить его ECC, модифицировать в проверенном слове записываемый байт, сгенерировать новое ECC и записать его обратно в память. Также необходимо иметь в виду, что после включения питания микроконтроллера память содержит произвольные значения и требует проведение первоначальной процедуры инициализации.

5.4.3 ECC кодирование для внешней шины

При обращении в область MEMIF существует два варианта организации ECC:

- параллельный PECC (аналогичный организации ECC для RAM);
- последовательный SECC (когда ECC биты расположены в старших адресах самой области памяти).

При работе с внешней памятью, в зависимости от режима ECC, биты считываются одновременно с данными (параллельная организация), либо в два этапа (последовательная организация: сначала считываются данные, затем ECC), и затем производится проверка ECC

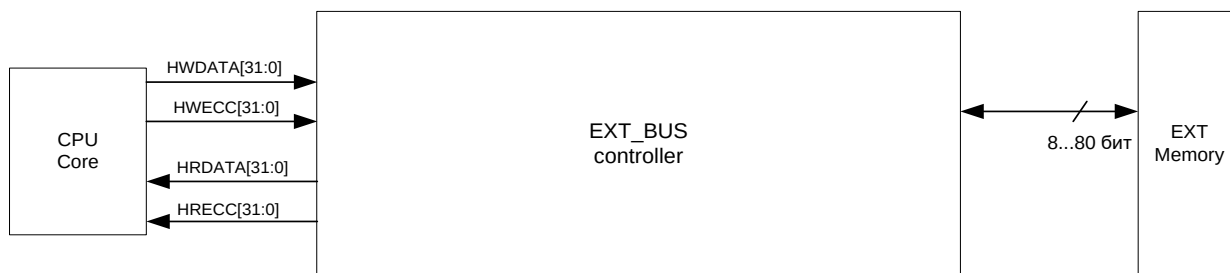


Рисунок 12 – Схема контроля ECC для внешней шины

Данные, считанные по внешней шине могут быть закэшированы в кэш. Для этого при возвращении данных процессору к ним снова генерируется ECC для записи в кэш память. При работе кэш-памяти когерентность закэшированных данных не обеспечивается. Также необходимо иметь в виду, что после включения питания внешняя память может содержать произвольные значения и требует дополнительной процедуры инициализации.

5.4.4 ЕСС кодирование для периферии

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

5.4.5 Н-матрица ЕСС по коду Хемминга (72, 64) и (78, 64)

Н-матрица для кодирования данных представлена далее (см. таблицу 8).

5.4.6 Н-матрица ЕСС по коду Хемминга (7, 4)

Т а б л и ц а 7 – Матрица для кода Хемминга (7,4)

	ecc			data			
	0	1	2	0	1	2	3
S0	1	0	0	1	1	1	0
S1	0	1	0	1	0	1	1
S2	0	0	1	1	1	0	1

5.4.7 Функция вычисления ЕСС для записи в память

Вычисление 8-битной ЕСС суммы для 32-битного слова в памяти осуществляется на основании самого слова (unsigned int DATA) и адреса (unsigned int ADR) расположения слова.

$SECC = \text{getecc}(((\text{unsigned long long})\text{ADR}) \ll 32) \mid \text{DATA});$

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;
}
```

Таблица 8 – Н-матрица ECC кода (72,64) ECC кода (78,64)

		Bits	S0	S1	S2	S3	S4	S5	S6	S7
Проверочные биты		0	1	0	0	0	0	0	0	0
		1	0	1	0	0	0	0	0	0
		2	0	0	1	0	0	0	0	0
		3	0	0	0	1	0	0	0	0
		4	0	0	0	0	1	0	0	0
		5	0	0	0	0	0	1	0	0
		6	0	0	0	0	0	0	1	0
		7	0	0	0	0	0	0	0	1
Биты данных	0	0	1	1	0	0	0	1	0	0
		1	1	1	0	0	0	0	1	0
		2	1	1	0	0	0	0	0	1
		3	1	0	1	1	1	1	0	0
		4	1	0	1	0	0	0	1	0
		5	1	0	1	0	0	0	0	1
		6	1	0	0	1	0	0	0	1
	1	7	1	0	0	1	0	0	1	0
		8	0	1	1	0	0	0	1	0
		9	0	1	1	0	0	0	0	1
		10	1	1	1	0	0	0	0	0
		11	0	1	0	1	1	1	1	0
		12	0	1	0	1	0	0	0	1
		13	1	1	0	1	0	0	0	0
	2	14	1	1	0	0	1	0	0	0
		15	0	1	0	0	1	0	0	1
		16	0	0	1	1	0	0	0	1
		17	1	0	1	1	0	0	0	0
		18	0	1	1	1	0	0	0	0
		19	0	0	1	0	1	1	1	1
		20	1	0	1	0	1	0	0	0
	3	21	0	1	1	0	1	0	0	0
		22	0	1	1	0	0	1	0	0
		23	1	0	1	0	0	1	0	0
		24	1	0	0	1	1	0	0	0
		25	0	1	0	1	1	0	0	0
		26	0	0	1	1	1	0	0	0
		27	1	0	0	1	0	1	1	1
	4	28	0	1	0	1	0	1	0	0
		29	0	0	1	1	0	1	0	0
		30	0	0	1	1	0	0	1	0
		31	0	1	0	1	0	0	1	0
		32	0	1	0	0	1	1	0	0
		33	0	0	1	0	1	1	0	0
		34	0	0	0	1	1	1	0	0
	5	35	1	1	0	0	1	0	1	1
		36	0	0	1	0	1	0	1	0
		37	0	0	0	1	1	0	1	0
		38	0	0	0	1	1	0	0	1
		39	0	0	1	0	1	0	0	1
		40	0	0	1	0	0	1	1	0
		41	0	0	0	1	0	1	1	0
	6	42	0	0	0	0	1	1	1	0
		43	1	1	1	0	0	1	0	1
		44	0	0	0	1	0	1	0	1
		45	0	0	0	0	1	1	0	1
		46	1	0	0	0	1	1	0	0
		47	1	0	0	1	0	1	0	0
		48	0	0	0	1	0	0	1	1
	7	49	0	0	0	0	1	0	1	1
		50	0	0	0	0	0	1	1	1
		51	1	1	1	1	0	0	1	0
		52	1	0	0	0	1	0	1	0
		53	1	0	0	0	0	1	1	0
		54	0	1	0	0	0	1	1	0
		55	0	1	0	0	1	0	1	0
		56	1	0	0	0	1	0	0	1
		57	1	0	0	0	0	1	0	1
		58	1	0	0	0	0	0	1	1
		59	0	1	1	1	1	0	0	1
		60	0	1	0	0	0	1	0	1
		61	0	1	0	0	0	0	1	1
		62	0	0	1	0	0	0	1	1
		63	0	0	1	0	0	1	0	1

	Bits	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
Проверочные биты	0	1													
	1		1												
	2			1											
	3				1										
	4					1									
	5						1								
	6							1							
	7								1						
	8									1					
	9										1				
	10											1			
	11												1		
	12													1	
	13														1
Биты данных	0	0	1	1	1		1	1	1		1	1			
		1		1	1	1		1	1	1		1			
		2			1	1	1	1	1	1		1	1		
		3				1	1	1	1	1	1	1	1		
		4					1	1	1	1	1	1	1	1	1
		5	1	1	1		1			1	1		1	1	1
		6	1			1	1		1				1	1	
		7		1			1	1		1				1	1
	1	8	1	1			1				1				1
		9	1			1		1		1	1	1			
		10		1			1		1		1	1	1		
		11			1			1		1		1	1	1	
		12				1			1		1		1	1	1
		13	1	1	1			1	1			1	1		1
		14	1			1	1	1		1	1	1	1	1	1
		15	1		1							1	1	1	
	2	16		1		1							1	1	1
		17	1	1				1	1		1	1		1	1
		18	1				1	1		1	1		1		1
		19	1		1		1						1		
		20		1		1		1						1	
		21			1		1		1						1
		22	1	1	1	1	1		1	1	1	1			
		23		1	1	1	1	1		1	1	1	1		
	3	24			1	1	1	1	1	1		1	1	1	
		25				1	1	1	1	1		1	1	1	
		26					1	1	1	1	1		1	1	1
		27	1	1	1		1		1				1	1	1
		28	1			1	1				1			1	1
		29	1		1			1	1	1	1	1			1
		30	1		1	1	1	1	1	1			1	1	
		31		1		1	1	1	1	1	1			1	1
	4	32			1		1	1	1	1	1	1		1	1
		33	1	1	1	1	1		1			1			1
		34	1			1			1			1		1	
		35		1			1			1			1		
		36			1			1					1		1
		37	1	1	1	1	1	1	1		1			1	
		38		1	1	1	1	1			1				1
		39	1	1		1				1	1	1	1		
	5	40		1	1		1				1	1	1	1	
		41			1	1		1				1	1	1	1
		42				1	1		1				1	1	1
		43	1	1	1				1	1	1	1		1	1
		44	1			1	1	1	1	1			1		1
		45	1		1					1			1		1
		46	1		1	1	1	1	1			1	1		1
		47		1		1	1	1	1	1			1	1	1
	6	48	1	1					1			1		1	1
		49		1	1					1			1		1
		50	1	1		1	1	1	1				1		1
		51	1							1	1				1
		52		1							1	1			1
		53	1	1			1	1	1			1			
		54		1	1			1	1	1		1		1	
		55			1	1			1	1	1		1		1

5.5 Режимы работы микроконтроллера

Выбор режима запуска микроконтроллера определяется при включении микроконтроллера при выполнении программы стартового загрузчика из масочного ПЗУ BOOT ROM. Режим запуска задается комбинацией сигналов на входе MODE[7:0]. При этом режим закодирован кодом Хемминга (7,4), таким образом, биты MODE[3:0] задают режим, а биты MODE[7:4] задают проверочные ECC биты. При выборе режима запуска автоматически исправляются одиночные ошибки в комбинации режима запуска, при обнаружении двойных ошибок микроконтроллер остается в режиме ожидания в загрузочной программе. Для определения режима запуска микроконтроллер конфигурирует выводы MODE[7:0] ({PE[5:0], PD[31:30]}) 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 смасжорированного значения MODE. И результирующее значение записывается в биты MODE регистров BKP_REG_60_TMRx блока PWR_BKP. Проводится контроль записанного значения (отсутствие разницы в регистрах TMR0,1,2 блока BKP_PWR), и на основании конечного значения MODE осуществляется переход к выполнению основной программы в выбранном режиме. При обнаружении неисправимой ошибки микроконтроллер остается в режиме ожидания в бутовой программе.

Таблица 9 – Режимы работы микроконтроллера

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	-	Резерв
1011	0010	-	Резерв
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) – 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) – 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 SPI0 интерфейсу с включенным интерфейсом отладки через выводы JTAG_B
1110	1000	SPI1+JA	Загрузка последовательно из внешней памяти по SPI0 интерфейсу с включенным интерфейсом отладки через выводы JTAG_A
1001	1001	SPI2+JB	Загрузка последовательно из внешней памяти по SPI0 интерфейсу с включенным интерфейсом отладки через выводы JTAG_B
0101	1010	SPI3+JA	Загрузка последовательно из внешней памяти по SPI0 интерфейсу с включенным интерфейсом отладки через выводы JTAG_A
0010	1011	UART0+JB	Загрузка последовательно из внешней памяти по UART0 интерфейсу с включенным интерфейсом отладки через выводы JTAG_B

Биты ECC MODE[7:4]	Биты режима MODE[3:0]	Режим	Краткое описание
0011	1100	UART0+JA	Загрузка последовательно из внешней памяти по UART0 интерфейсу с включенным интерфейсом отладки через выводы JTAG_A
0100	1101	-	Резерв
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 блока PWR_BKP REG_60_TMRx битами MODE[0] и битом DISABLE_JTAG. При выборе того или иного интерфейса данные выводы жестко переопределяются на выполнение функции JTAG.

5.5.1 Режим EXTBUS_8_ECC+JA

После определения данного режима бутовая 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 бутовая программа выполняет следующие действия:

- устанавливает адрес указателя стека в значение, считанное из ячейки с адресом 0x1000_0000 (SP);
- копирует из внешней памяти таблицу ветров прерываний в память RAMC
- осуществляет безусловный переход по адресу, считанному из ячейки 0x1000_0004 (RESET_HANDLER).

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

5.5.2 Режим EXTBUS_8_ECC+JB

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

5.5.3 Режим EXTBUS_CFG+JA

После определения данного режима бутовая 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 бутовая программа исправляет её, при обнаружении двойной ошибки – формирует сигнал сброса.

После проверки, бутовая 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.
- 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 бутовая программа выполняет следующие действия:

- устанавливает адрес указателя стека в значение, считанное из ячейки с адресом 0x1000_0000 (SP);
- копирует из внешней памяти таблицу векторов прерываний в память RAMC;
- осуществляет безусловный переход по адресу, считанному из ячейки 0x1000_0004 (RESET_HANDLER).

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

5.5.4 Режим EXTBUS_CFG+JB

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

5.5.5 Режим SPI0+JB

После определения данного режима бутовая программа конфигурирует контроллер SSP0 для работы с микросхемами памяти по последовательному интерфейсу для конфигурирования ПЛИС (5576PT1Y, 5576PC1Y 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 бутовой программы.

Режим SPI0+JB необходим для записи в ОЗУ какой-либо программы, 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.

5.5.5.1 Параметры связи по SPI

Для связи по SPI выбраны следующие параметры канала связи:

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

5.5.5.2 Протокол обмена по SPI

После конфигурирования выводов, бутровая 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 (см. таблицу 10).

Таблица 10 – Информационный заголовок, считываемый из EEPROM

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 конфигурирования.				

- 32-разрядный адрес (SRAM ADDR), по которому будет расположен образ копируемой программы.
- 32-разрядный адрес (START ADDR), которому будет передано управление после копирования программы.
- 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 бутровая программа последовательно считывает образ программы (см. таблицу 11).

Таблица 11 – Считываемый 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.

5.5.5.3 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;
}
```

5.5.6 Режим SPI1+JA

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

5.5.7 Режим SPI2+JB

После определения данного режима бутовая 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 бутовой программы.

Данный режим необходим для записи в ОЗУ какой-либо программы, 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".

5.5.7.1 Параметры связи по SPI

Для связи по SPI выбраны следующие параметры канала связи:

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

5.5.7.2 Протокол обмена по SPI

После конфигурирования выводов бутовая программа из внешней EEPROM по адресу 0x000_000 считывает 32-х 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 (см. таблицу 12).

Таблица 12 – Считываемый 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.
- 32-разрядный адрес (START ADDR), которому будет передано управление после копирования программы.
- 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 бутовая программа последовательно считывает образ программы (см. таблицу 13).

Таблица 13 – Считываемый 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".

5.5.7.3 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;
}
```

5.5.8 Режим SPI3+JA

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

5.5.9 Режим UART0+JA

После определения данного режима бутовая 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 бутовой программы.

Данный режимы предоставляет достаточный набор операций, необходимых для записи в ОЗУ какой-либо программы, 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.

5.5.9.1 Параметры связи по UART

Для связи по UART выбраны следующие параметры канала связи:

- начальная скорость – 9600 бод;
- количество данных – 8 бит;

- 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.

5.5.9.2 Протокол обмена по UART

После синхронизации с Master загрузчик переходит в диспетчер команд. Таким образом, устройству Master доступны следующие команды:

5.5.9.3 Команды UART-загрузчика

Список команд UART-загрузчика приведен в таблице 14.

Таблица 14 – Список команд UART-загрузчика

Команда	Код	ASCII Символ	Описание
CMD_SYNC	0x00		Пустая команда. Загрузчик ее принимает, но ничего по ней не делает
CMD_CR	0x0D		Выдача приглашения устройству Master
CMD_BAUD	0x42	'B'	Установка скорости обмена
CMD_LOAD	0x4C	'L'	Загрузка массива байт
CMD_VFY	0x59	'Y'	Выдача массива байт
CMD_RUN	0x52	'R'	Запуск программы на выполнение

5.5.9.4 Синхронизация с внешним устройством

Начальные условия.

На этапе синхронизации с внешним устройством (Master) вывод Rx используется как вход. Master постоянно посылает в канал синхросимвол – 0. Загрузчик подстраивает свою скорость таким образом, чтобы минимизировать ошибки обмена. Как только Загрузчик настроил скорость, он переходит в диспетчер команд и выдает приглашение (3 байта 0x0D (перевод строки), 0x0A (возврат каретки), 0x3E ('>')) Master-у.

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

5.5.9.5 Команда CMD_SYNC

Пустая команда.

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

Таблица 15 – 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, выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды

5.5.9.6 Команда CMD_CR

Выдача приглашения устройству Master.

Таблица 16 – 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, выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды. Выдает код команды CMD_CR. Выдает код 0x0A Выдает код 0x3E (ASCII символ '>')

5.5.9.7 Команда CMD_BAUD

Установка скорости обмена.

Таблица 17 – 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	Если параметр принят с ошибками, выдает код ошибки ERR_CHN или ERR_BAUD и завершает обработку текущей команды. Выдает код команды CMD_BAUD. Устанавливает новое значение скорости обмена

5.5.9.8 Команда CMD_LOAD

Загрузка массива байт в память микроконтроллера.

Таблица 18 – 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: Если хотя бы один из параметров принят с ошибками, выдает код ошибки ERR_CHN и завершает обработку текущей команды
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. 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')
--	---

5.5.9.9 Команда CMD_VFY

Выдача массива байт из памяти микроконтроллера.

Таблица 19 – Команда CMD_VFY

Код команды	CMD_VFY = 0x59
ASCII символ, соответствующий коду команды	'Y'
Количество параметров команды	2
Параметр 1	Адрес памяти источника данных
Параметр 2	Размер массива в байтах
Формат команды:	
Master: Выдает код команды CMD_VFY	Slave: Если команда принята с ошибками, выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Master: Выдает параметр 1	Slave: Если хотя бы один из параметров принят с ошибками, то выдает код ошибки ERR_CHN и завершает обработку текущей команды
Master: Выдает параметр 2	Slave: Если хотя бы один из параметров принят с ошибками, выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_VFY. Выдает массив байт младшим байтом вперед. По окончании передачи массива выдает код ответа REPLY_OK = 0x4B ('K')

5.5.9.10 Команда CMD_RUN

Запуск программы на выполнение.

Таблица 20 – Команда CMD_RUN

Код команды	CMD_RUN = 0x52
ASCII символ, соответствующий коду команды	'R'
Количество параметров команды	1
Параметр	Адрес таблицы векторов загруженной программы
Формат команды:	
Master: Выдает код команды CMD_RUN	Slave: Если команда принята с ошибками, выдает код ошибки ERR_CHN или ERR_CMD и завершает обработку текущей команды
Выдает параметр	Если параметр принят с ошибками, выдает код ошибки ERR_CHN и завершает обработку текущей команды. Выдает код команды CMD_RUN. Устанавливает значение MSP и PC

	согласно таблице векторов (NVIC не перепрограммируется) и, таким образом, Slave завершает свое выполнение
--	---

5.5.9.11 Прием параметров команды

Параметры команд – это 4-байтные числа.

Параметры передаются младшим байтом вперед.

В качестве значения параметра запрещено использовать число 0xFFFFFFFF.

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

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

5.5.9.12 Сообщения об ошибках

Сообщения об ошибках – это 2-байтные 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

5.5.9.13 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.

5.5.9.14 Ошибка ERR_CMD

Принята неизвестная команда.

Код ошибки 0x63 ('c').

Выдается диспетчером команд, если принят неизвестный код команды.

5.5.9.15 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.

5.5.10 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.

5.5.11 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.

5.5.12 Режим WAIT_BOOT_JA

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

5.5.13 Режим WAIT_BOOT_JB

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

5.5.14 Статус загрузчика

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

Таблица 21 – Поля статуса загрузчика

Адрес ОЗУ	Размер, бит	Обозначение	Описание
0x2000_7FF0	8	INIT_ERR_CNTR	Число повторных инициализаций выводов MODE[7:0]. Значение INIT_ERR_CNTR >2 свидетельствует 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: – нет ошибки; – одиночная ошибка; – двойная ошибка; – 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

Адрес ОЗУ	Размер, бит	Обозначение	Описание
0x2000_7FFA	8	MODE_EXTBUS	Исправленное значение конфигурации (ECC+CFG0) в режиме EXTBUS_CFG+JB или EXTBUS_CFG+JA
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

5.6 Организация памяти

Микроконтроллер имеет Фон-Неймановскую архитектуру памяти, т.е. 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 0x00000000 до 0x1FFFFFFF) для кода программ и DATASECTION (от 0x20000000 до 0xDFFFFFFF) для данных. Доступ к CODESECTION и DATASECTION осуществляется через шину SBUS. Доступ DMA-контроллера осуществляется через шину MBUS и может быть выполнен в диапазоне DATASECTION. 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 RAMC или внешней памяти ОЗУ, отображаемой в эту секцию.

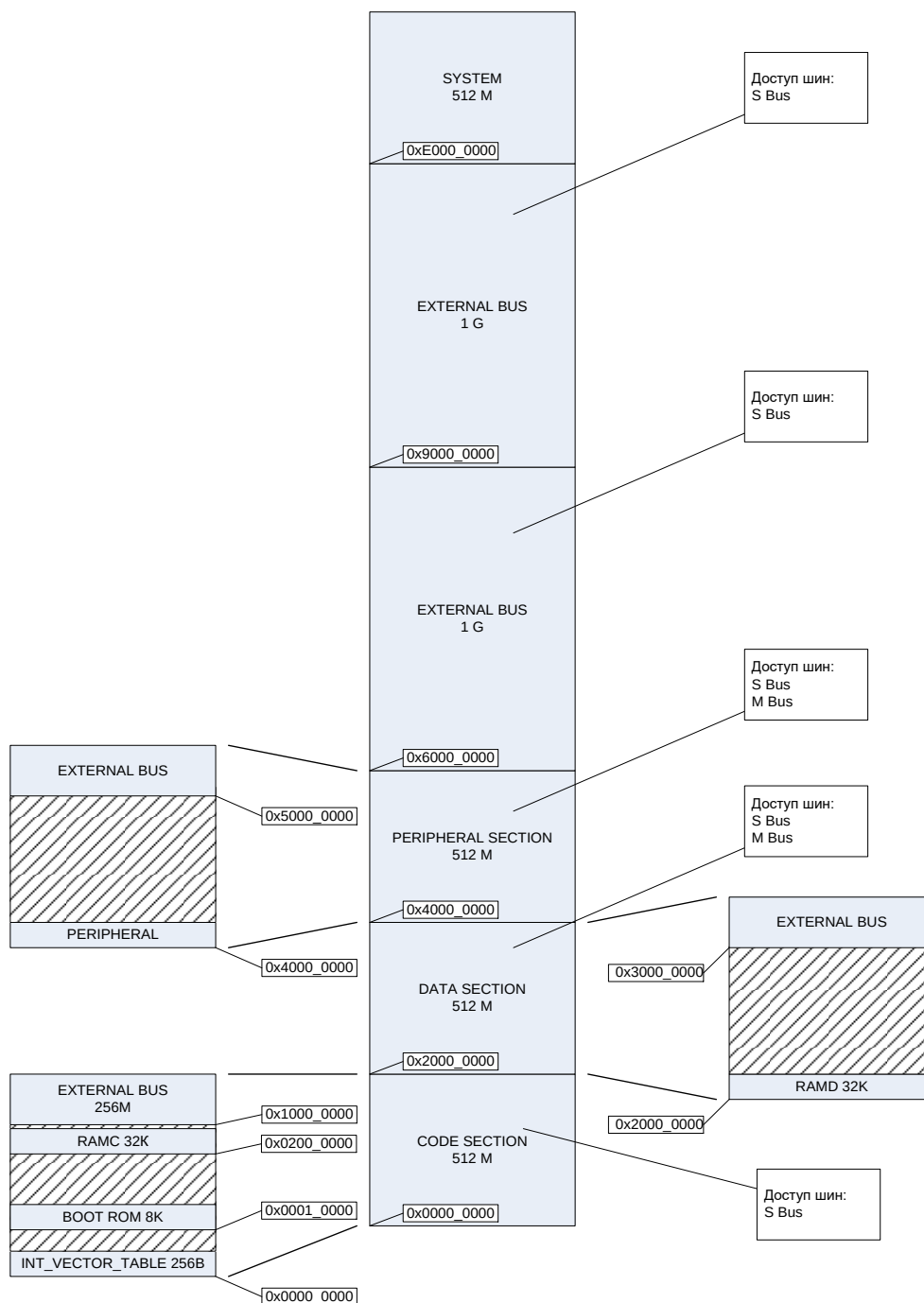


Рисунок 13 – Распределение адресного пространства

Возможная коммутация агентов в блоке BUSMATRIX представлена на рисунке 14. Арбитраж между мастерами на каждой Slave шине осуществляется по принципу round-robin, т.е. равного приоритета.

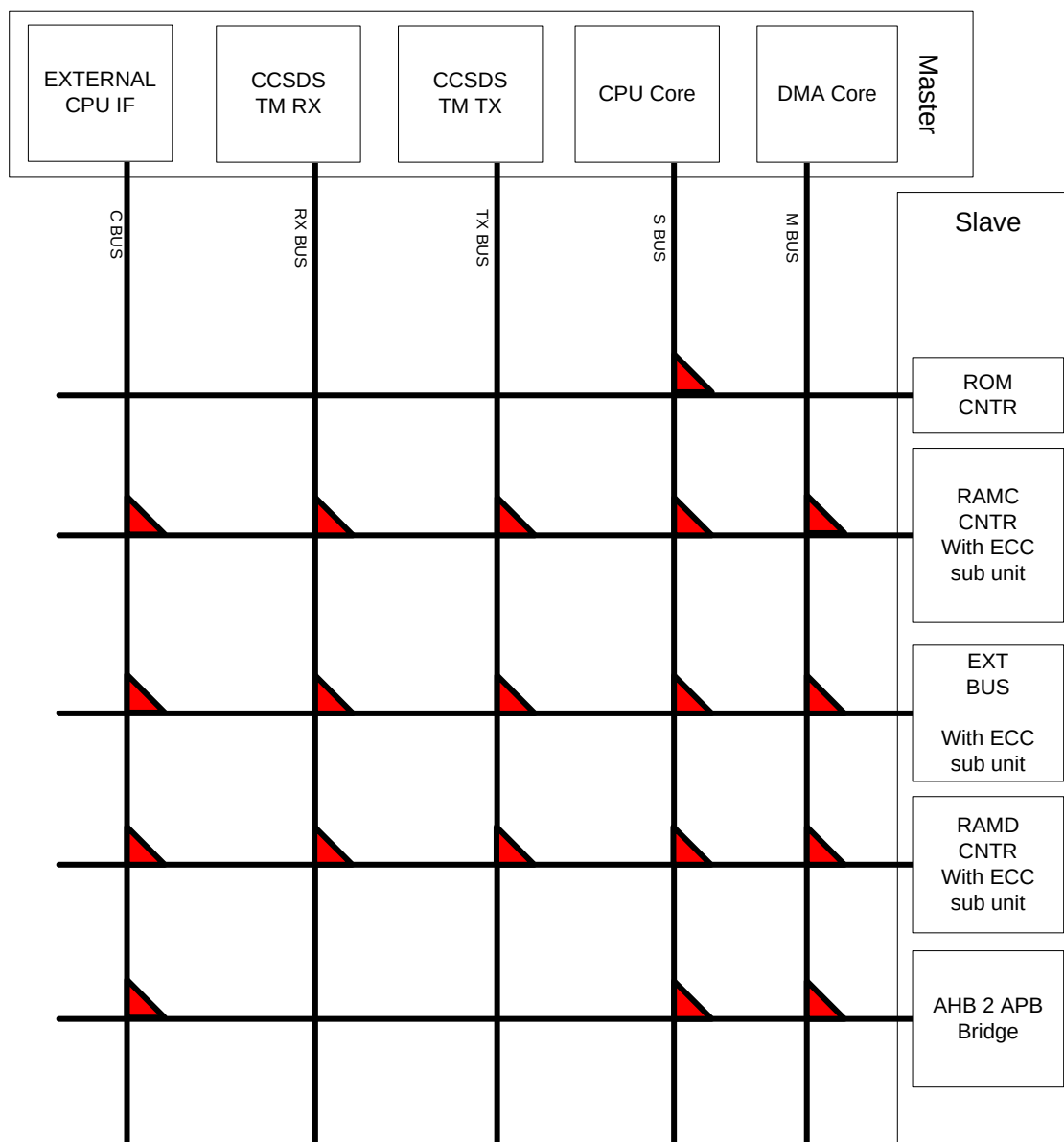


Рисунок 14 – Схема внутренних межсоединений

5.6.1 Распределение областей памяти

Таблица 22 – Распределение адресного пространства

Базовый адрес	Размер, байт	Обозначение	Доступ по шинам	Описание
0x0000_0000 (0x0001_0000)	8K	BOOT ROM	S	Область памяти загрузочного ПЗУ. Из этой памяти микроконтроллер начинает работу, определяет дальнейший режим функционирования и выполняет необходимые настройки (перед передачей управления пользовательской программе производит переключение области отображения на RAMC)

Базовый адрес	Размер, байт	Обозначение	Доступ по шинам	Описание
0x0000_2000 (0x0001_2000)		-	-	-
0x0200_0000 (0x0000_0000)	32K	RAMC	S, M, RX, TX, C	Статическое ОЗУ в CODESECTION, предназначено для программ
0x0200_8000 (0x0000_8000)		-	-	-
0x0300_0000		-	-	-
0x1000_0000	256M	EXT BUS	S, M, C	Область памяти для отображения внешней системной шины
0x2000_0000	32K	RAMD	S, M, RX, TX, C	Статическое ОЗУ в DATASECTION, предназначено для данных
0x2100_0000		-	-	-
0x2100_8000		-	-	-
0x2101_0000		-	-	-
0x2101_8000		-	-	-
0x2102_0000		-	-	-
0x2103_0000		-	-	-
0x2200_0000		-	-	-
0x4000_0000	1M	PERIPHERAL	S, M, C	Область отображения регистров периферии шины APB. Подробнее см. таблицу 23
0x4200_0000	32M	-	-	-
0x5000_0000	256M	EXT BUS	S, M, C	Область памяти для отображения внешней системной шины
0x6000_0000	1G	EXT BUS	S, M, RX, TX, C	Область памяти для отображения внешней системной шины
0xA000_0000	1G	EXTBUS	S, M, C	Область памяти для отображения внешней системной шины
0xE000_0000	512M	SYSTEM	-	Отображение внутренних регистров процессорного ядра

5.6.2 Распределение областей памяти для регистров периферии

Таблица 23 – Распределение адресного пространства периферии

	Базовый адрес	Размер, байт	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

	Базовый адрес	Размер, байт	Обозначение	Доступ по шинам	Описание
3	0x4000_3000		MDR_FTMODE		Контроллер блока диагностики ошибок
4	0x4000_4000		MDR_WDT		Контроллер сторожевых таймеров
5	0x4000_5000		MDR_EBC		Контроллер внешней шины
6	0x4000_6000		MDR_DMA		Контроллер DMA
7	0x4000_7000		MDR_RAMC		Контроллер ОЗУ в области CODE
8	0x4000_8000		MDR_RAMD		Контроллер ОЗУ в области DATA
9	0x4000_9000		MDR_ROM		Контроллер ПЗУ
10	0x4000_A000		MDR_CCSDSRX		Приемник телеметрической информации CCSRS
11	0x4000_B000		MDR_CCSDSTX		Передатчик телеметрической информации CCSRS
12	0x4000_C000		MDR_CACHE		Контроллер кэш памяти
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		MDR_SPW1		Контроллер SpaceWire1
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		MDR_CAN1		Контроллер CAN1
31	0x4009_2000		-		Зарезервировано
32	0x4009_3000		-		Зарезервировано
33	0x4009_4000		-		Зарезервировано
34	0x4009_5000		MDR_SSP0		Контроллер SSP0
35	0x4009_6000		MDR_SSP1		Контроллер SSP1
36	0x4009_7000		MDR_SSP2		Контроллер SSP2
37	0x4009_8000		MDR_SSP3		Контроллер SSP3
38	0x4009_9000		MDR_UART0		Контроллер UART0
39	0x4009_A000		MDR_UART1		Контроллер UART1
40	0x4009_B000		-		Зарезервировано
41	0x4009_C000		-		Зарезервировано
42	0x4009_D000				
43	0x4009_E000		-		Зарезервировано

	Базовый адрес	Размер, байт	Обозначение	Доступ по шинам	Описание
44	0x4009_F000		-		Зарезервировано
45	0x400A_0000		-		Зарезервировано
46	0x400A_1000		-		Зарезервировано
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_MILDAT2		Контроллер MIL2 данные
56	0x400A_B000		MDR_MILCNR2		Контроллер MIL2 управление
57	0x400A_C000		MDR_MILDAT3		Контроллер MIL3 данные
58	0x400A_D000		MDR_MILCNR3		Контроллер MIL3 управление
59	0x400A_E000				
60	0x400A_F000		MDR_CRC		Блок вычисления CRC
61	0x400B_0000		MDR_ECC		Блок вычисления ECC
62	0x400B_1000		MDR_SSP4		Контроллер SSP4
63	0x400B_2000		MDR_SSP5		Контроллер SSP5

5.6.3 Организация кэш-памяти

В микроконтроллере для встроенного процессорного ядра реализован блок кэш MDR_MCUCACHE размером 1 Кбит. 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, замещение данных происходит в соответствии с алгоритмом PLRU. В кэш-памяти кэшируются код и данные из разрешенных для кэширования диапазонов внешней шины MEMIF. При этом не обеспечивается аппаратная поддержка когерентности закэшированных данных, которые могут быть изменены в ходе работы. Обеспечение когерентности реализуется пользовательской программой путем перезапуска кэш-памяти. Кэш-память состоит из записей тега адреса и четырех слов данных с ECC кодами.

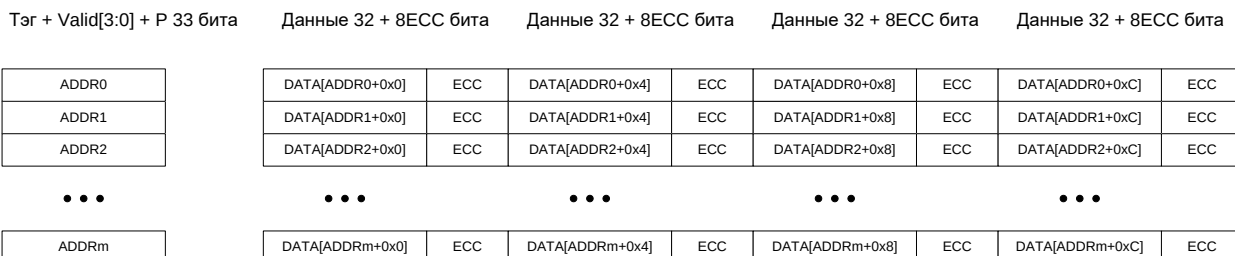


Рисунок 15 – Структура записей в кэш памяти

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 или чтении данных из внешней шины MEMIF, строки кэш-памяти 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 во внешнюю шину. 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.

5.6.4 Организация BOOTROM-памяти

Встроенная загрузочная ПЗУ имеет параллельную организацию ECC кодирования и имеет размерность 2048 x (32+8ECC). Конфликты между контроллером CPUIF, процессорным ядром и DMA контроллера разрешаются на уровне арбитража коммутатора АНВ. Загрузочное ПЗУ имеет доступ только на чтение. Попытка записи в загрузочное ПЗУ вызывает исключение busfaulterror.

Память BOOTROM имеет топологически распределенную структуру, т.е. 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.

5.6.5 Организация RAM-памяти

Встроенная статическая ОЗУ RAM имеет размерность 32768 × (32 + 8 ECC) и параллельную организацию ECC кодирования. В микроконтроллере реализовано два контроллера RAM, работающих в режиме LockStep, для проверки корректности исполнения запросов. 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-х разрядного слова в контроллере проверяется ECC записываемого слова;
- 2 генерируется новое ECC и записывается в память.

Запись 32-разрядного слова выполняется в один этап с проверкой/генерацией ECC. При выполнении операции считывания из встроенного ОЗУ всегда извлекается 32-разрядное слово с 8-ью 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, т.е. 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.

5.6.6 Организация доступа к внешней шине MEMIF

Таблица 24 – Интерфейсы контроллера MEMIF

Наименование входа	Направление	Логич. уровень	Описание
			Выводы для микросхемы
ADR[31:0]	Out	-	Выход шины адреса
DATAo[31:0]	Out	-	Выход шины данных
DATAo[47:32]	Out	-	Выход шины данных ECC
BEn[5:0]	Out	-	Выход ByteEnable 0 – разрешение байта 1 – запрет байта BEn[0] – DATAo[7:0] BEn[1] – DATAo[15:8] BEn[2] – DATAo[23:16] BEn[3] – DATAo[31:24] BEn[4] – DATAo[30:32] BEn[5] – DATAo[47:40]
CSn[7:0]	Out	-	Выход разрешения региона (активный 0) 0 – регион выбран 1 – регион не выбран
CS[7:0]	Out	-	Выход разрешения региона (активная 1) 1 – регион выбран 0 – регион не выбран
WEn	Out	-	Выход сигнала записи 0 – запись 1 – нет записи
OEn	Out		Сигнал чтения данных с внешних выводов 0 – есть чтение 1 – нет чтения
BWEn[5:0]	Out	-	Выход сигнала записи с сигналом ByteEnable BWEn[0] = WEn or BEn[0] BWEn[1] = WEn or BEn[1] BWEn[2] = WEn or BEn[2] BWEn[3] = WEn or BEn[3] BWEn[4] = WEn or BEn[4] BWEn[5] = WEn or BEn[5]
DATAo[31:0]	In	-	Вход шины данных
DATAo[47:32]	In	-	Вход шины данных ECC
Ready[7:0]	In	-	Вход готовности от блоков памяти с сигналом готовности 0 – не готов 1 – готов

Внешняя шина может иметь 8-, 16- и 32-битную организацию шины данных без учета проверочных бит ECC при параллельной организации (см. таблицу 25). При этом оперировать можно только 8-, 16- и 32-битными данными. 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.

Т а б л и ц а 25 – Организация ECC в зависимости от конфигурации внешней шины

Режим				Максимальный пользовательский объем	Примечание
Разрядность данных	Разрядность адреса	Разрядность ЕСС	Организация ЕСС		
8	32	-	нет	4 Гбайт	
16	31	-	нет	4 Гбайт	
32	30	-	нет	4 Гбайт	
8	32	8	послед. ЕСС	$4/5$ 4 Гбайт	SECC8
8	32	16	послед. ЕСС	$2/3$ 4 Гбайт	SECC16
16	31	8	послед. ЕСС	$4/5$ 4 Гбайт	SECC8
16	31	16	послед. ЕСС	$2/3$ 4 Гбайт	SECC16
32	30	8	послед. ЕСС	$4/5$ 4 Гбайт	SECC8
32	30	16	послед. ЕСС	$2/3$ 4 Гбайт	SECC16
32	30	8	парал. ЕСС	4 Гбайт	PECC8 DATA[39:32]
32	30	16	парал. ЕСС	4 Гбайт	PECC16 DATA[47:32]

5.6.6.1 Организация ЕСС для шины MEMIF

Для всех диапазонов внешней системной шины может быть задан режим работы с проверкой ЕСС или без проверки. Если проводится проверка ЕСС, она может быть организована по коду ЕСС8 (8 проверочных бит на 32 информационных) или ЕСС16 (16 проверочных бит на 32 информационных). При этом проверочные биты ЕСС могут располагаться как в дополнительных разрядах шины данных (32 данные + 8 или 16 ЕСС – параллельная организация РЕСС8 или РЕСС16), так и непосредственно в самой 32-разрядной памяти, начиная с программно задаваемого базового адреса расположения ЕСС для каждого региона (SECC8 или SECC16).

Режим ЕСС8 для 32-битного слова позволяет:

- обнаруживать и исправлять произвольные одиночные ошибки;
- обнаруживать любые двойные ошибки.

Режим ЕСС16 для 32-битного слова позволяет:

- обнаруживать и исправлять произвольные одиночные и двойные ошибки;
- обнаруживать любые тройные ошибки.

5.6.6.2 Параллельная организация ЕСС (РЕСС8 или РЕСС16)

Параллельная организация ЕСС для внешней системной шины возможна только при организации 32-разрядной шины данных. При этом дополнительно к каждому 32-м разрядам данных добавляется 8 или 16 разрядов ЕСС, т.е. общая разрядность шины данных будет 40 или 48 бит. При настройке разрядности шины данных в 16- или 8-битный режим и выбор параллельной организацией ЕСС обращение к внешней шине будет приводить к ошибке.

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 ЕСС, проверка ЕСС считанных данных;
- 2 модификация 8- или 16-разрядного слова в считанном и проверенном по ЕСС 32-разрядном слове, генерация ЕСС для модифицированного слова и запись в память.

При записи 32-разрядного слова за один этап проводится генерация ECC и запись в память. При чтении 8- или 16-разрядных слов из памяти всегда извлекается 32-разрядное слово с ECC. В контроллере внешней шины проверяется ECC, генерируется новое ECC и возвращается процессору, на входе которого ECC проверяется еще раз.

Т а б л и ц а 26 – Число транзакций на шине MEMIF в зависимости от разрядности операции при PECC8 и PECC16

Тип операции	Разрядность данных	Разрядность шины	Число транзакций на шине
Чтение	32,16,8	32	1 чтение
Запись	32	32	1 запись
Запись	16,8	32	1 чтение + 1 запись

5.6.6.3 Последовательная организация ECC (SECC8 или SECC16)

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. Таким образом, независимо от реализации шины данных памяти на внешней системной шине для процессора отображается 32-разрядная память. Контрольные ECC биты для памяти располагаются в старших адресах каждого диапазона, начиная с программно задаваемого базового адреса ECCBASE (младшие 4 бита адреса ECCBASE игнорируются и считаются равными нулю).

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

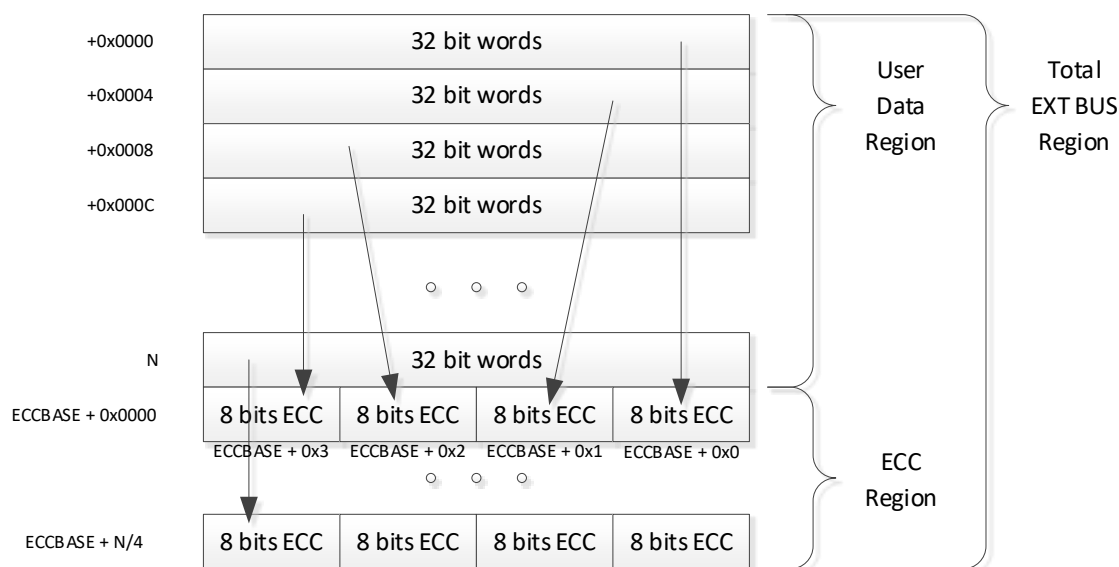


Рисунок 16 – Расположение данных в памяти на шине MEMIF для 32-, 16-, 8-битных шин при SECC8

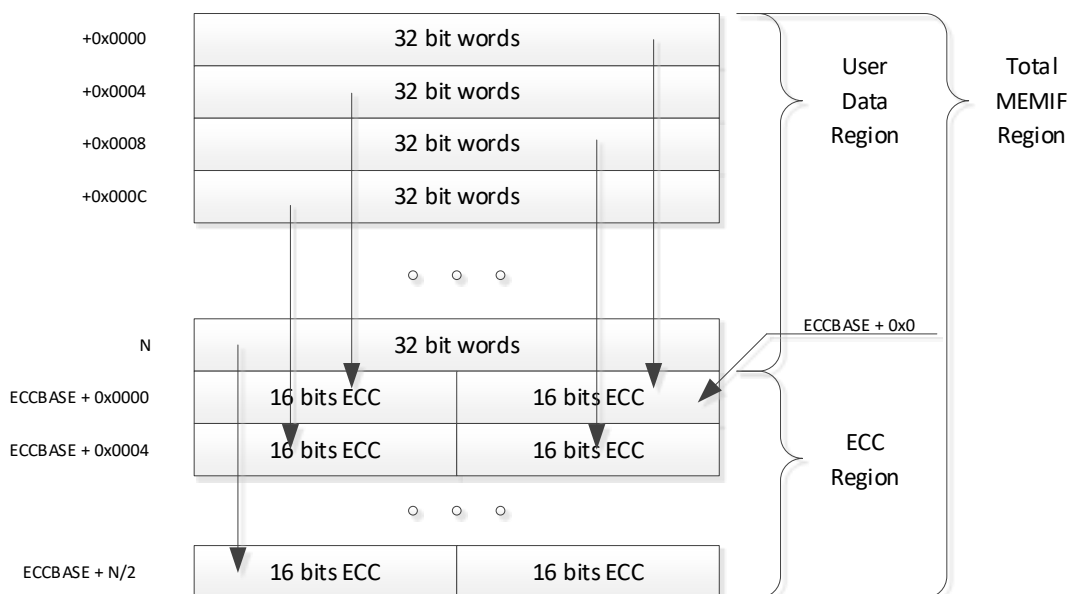


Рисунок 17 – Расположение данных в памяти на шине MEMIF для 32-, 16-, 8-битных шин при SECC16

Такая организация памяти требует:

- два обращения для считывания 32-разрядных данных с ECC при 32-разрядной шине данных;
- три обращения при 16-разрядной шине данных;
- пять обращений при 8-разрядной шине данных при SECC8.
- шесть обращений при 8-разрядной шине данных при SECC16.

Число обращений для выполнения операций чтения и записи данных различной разрядности при различной организации шины данных представлено в таблице 27.

Таблица 27 – Число транзакций на шине MEMIF при последовательной организации ECC

Тип операции	Разрядность данных	Разрядность шины	Тип ECC	Число транзакций на шине
Чтение	32,16,8	32	SECC8 и SECC16	2 чтения
Чтение	32,16,8	16	SECC8 и SECC16	3 чтения
Чтение	32,16,8	8	SECC8	5 чтений
Чтение	32,16,8	8	SECC16	6 чтений
Запись	32	32	SECC8 и SECC16	2 записи
Запись	32	16	SECC8 и SECC16	3 записи
Запись	32	8	SECC8	5 записей
Запись	32	8	SECC16	6 записей
Запись	16,8	32	SECC8 и SECC16	2 чтения + 2 записи
Запись	16,8	16	SECC8 и SECC16	3 чтения + 2 записи
Запись	16	8	SECC8	5 чтений + 3 записи
Запись	16	8	SECC16	6 чтений + 4 записи
Запись	8	8	SECC8	5 чтений + 2 записи
Запись	8	8	SECC16	6 чтений + 3 записи

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.

5.6.6.4 Вычисление адреса расположения ECC-бит

При параллельной организации ECC проверочные биты располагаются в расширении шины данных и считываются одновременно со словом. При последовательной организации ECC проверочные биты 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 ECC-бит вычисляется согласно формулам

$$\text{ADDR}_{\text{ECC}}[31:0] = \text{ECCBASE}[31:0] + \text{OFFSET}(\text{ADDR}[31:2]), \quad (2)$$

при 8-, 16-, 32-битной шине данных при ECC8.

$$\text{ADDRECC}[31:0] = \text{ECCBASE}[31:0] + \text{OFFSET}(\text{ADDR}[31:2]) < 1, \quad (3)$$

при 8-, 16-, 32-битной шине данных при ECC16.

где $\text{OFFSET}(A) = A - \text{REGIONBASE}(A)$;

ECCBASE – для каждого сегмента должен лежать внутри адресного пространства сегмента.

5.6.6.5 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 (см. таблицу 28). 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.).

Т а б л и ц а 28 – Число транзакций на шине MEMIF в режиме работы без контроля ECC

Тип операции	Разрядность данных	Разрядность шины	Число транзакций на шине
Чтение	32, 16, 8	32	1 чтение
Чтение	32	16	2 чтения
Чтение	32	8	4 чтения
Чтение	16, 8	16	1 чтение
Чтение	16	8	2 чтения
Чтение	8	8	1 чтение
Запись	32	32	1 запись
Запись	32	16	2 записи
Запись	32	8	4 записи
Запись	16, 8	32	1 чтение + 1 запись
Запись	16	16	1 запись
Запись	16	8	2 записи
Запись	8	16	1 чтение + 1 запись
Запись	8	8	1 запись

С помощью контроллера внешней шины MEMIF можно организовать до 8-ми регионов с подключением внешних устройств с отображением по адресам, приведенным в таблице 29. Базовые адреса регионов фиксированы и не могут изменяться.

Таблица 29 – Базовые адреса внешней системной шины

Номер	Начальный адрес	Конечный адрес	Шина доступа	Примечание
REGION0	0x1000_0000	0x17FF_FFFF		
REGION1	0x1800_0000	0x1FFF_FFFF		
REGION2	0x5000_0000	0x57FF_FFFF		
REGION3	0x5800_0000	0x5FFF_FFFF		
REGION4	0x6000_0000	0x67FF_FFFF		
REGION5	0x6800_0000	0x6FFF_FFFF		
REGION6	0x7000_0000	0x7FFF_FFFF		
REGION7	0x8000_0000	0xDFFF_FFFF		

Для каждого региона может быть задан свой режим работы. При отображении выводов контроллера внешней шины на пользовательские выводы этим выводам необходимо задать функцию внешней шины. Если выводу не задана функция внешней шины, с него считывается ноль. Если выводы внешней шины отображаются на выделенные выводы для многокристальной сборки, они выполняют только эту функцию, и их настройка выполняется в регистрах контроллера внешней шины. Для различных регионов могут быть выбраны различные способы отображения – на пользовательские выводы, либо на выводы для микросборки.

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

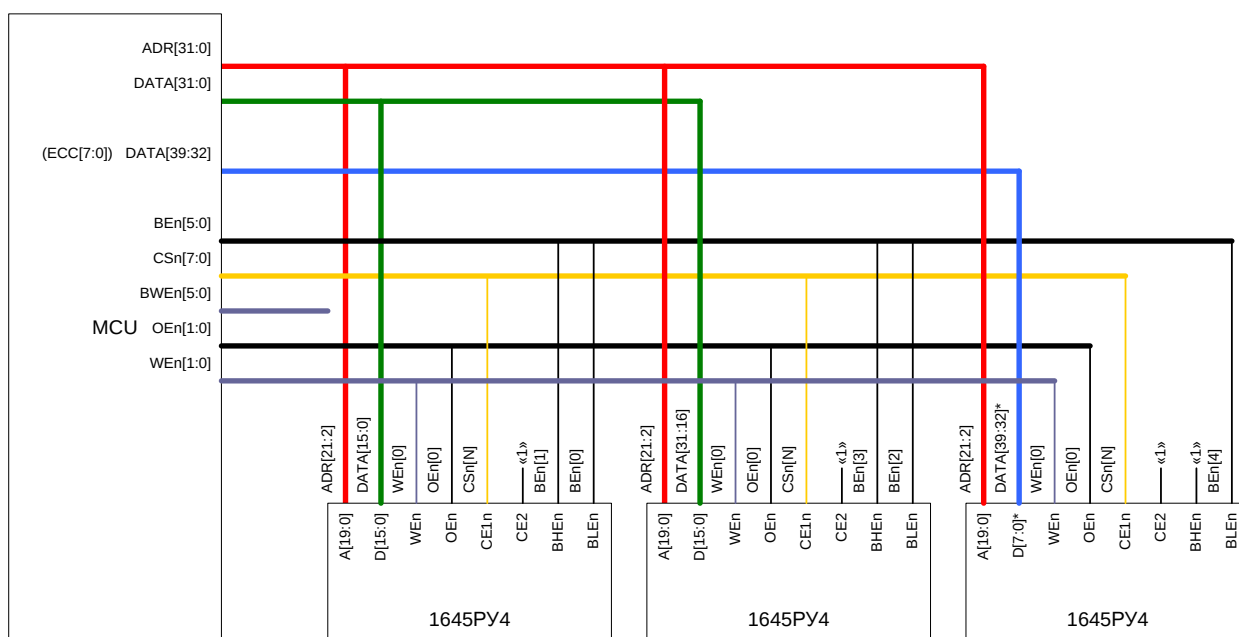


Рисунок 18 – Организация 32-битной шины с параллельной PECC8 на базе ОЗУ 1645PY4 (объем пользовательских данных до 4 Мбайт)

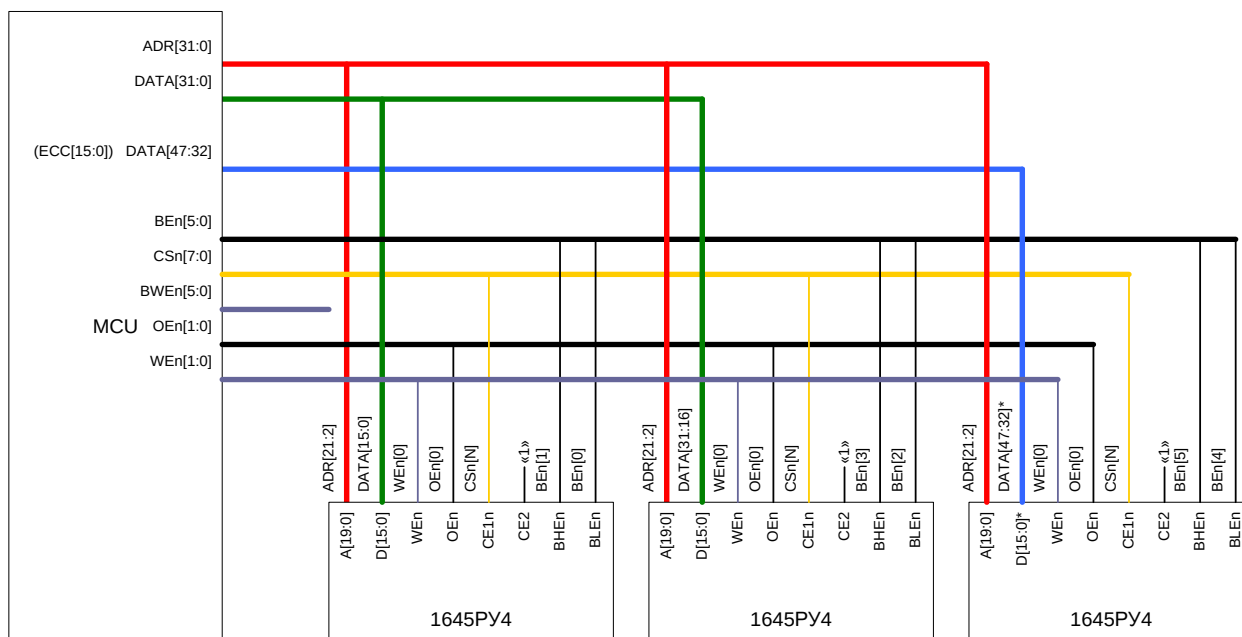


Рисунок 19 – Организация 32-битной шины с параллельной РЕСС16 на базе ОЗУ 1645PY4 (объем пользовательских данных до 4 Мбайт)

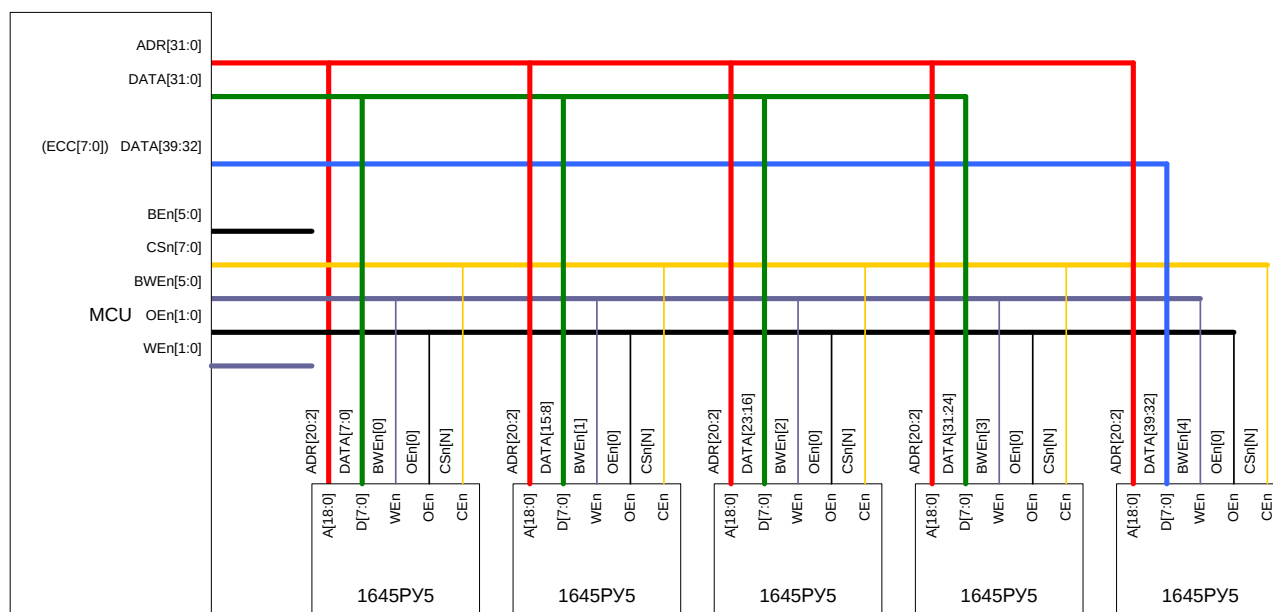


Рисунок 20 – Организация 32-битной шины с параллельной РЕСС8 на базе ОЗУ 1645PY5 (объем пользовательских данных до 2 Мбайт)

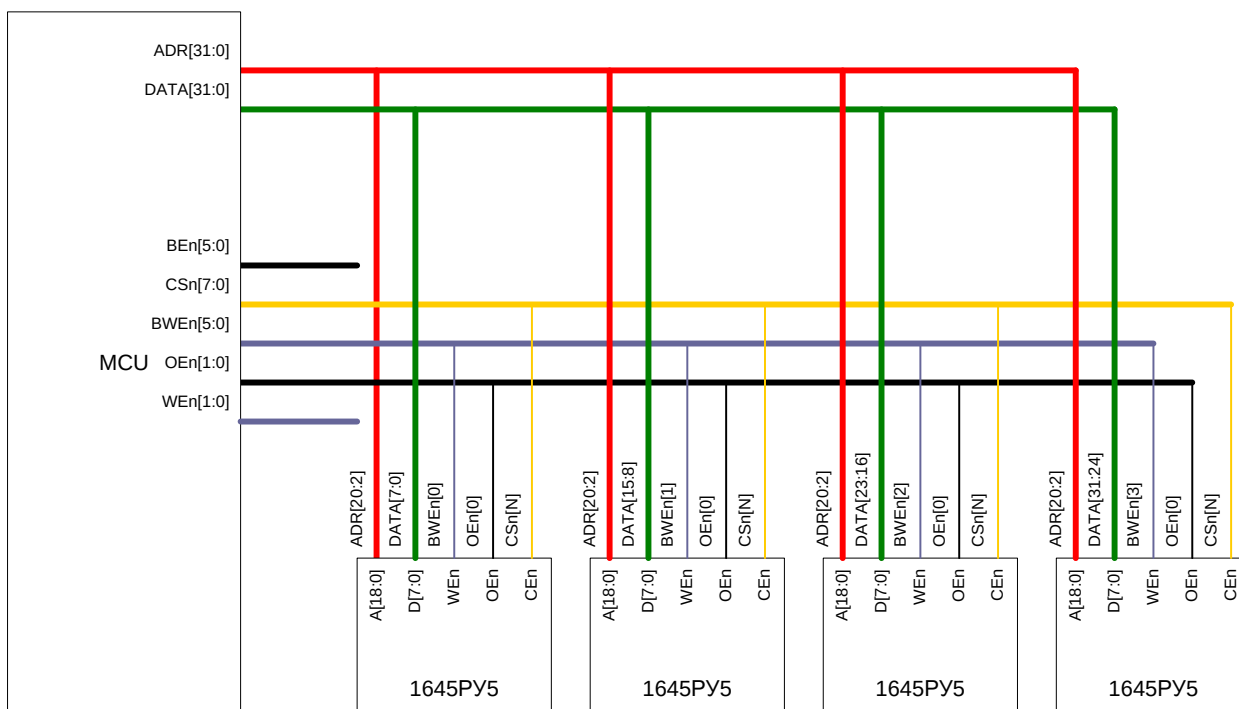


Рисунок 21 – Организация 32-битной шины с последовательной организацией SECC8 или SECC16 или без ECC на базе ОЗУ 1645PY5
(объем пользовательских данных до 1,6 Мбайт с последовательной SECC8, до 1,3 Мбайт с последовательной SECC16 и до 2 Мбайт без ECC)

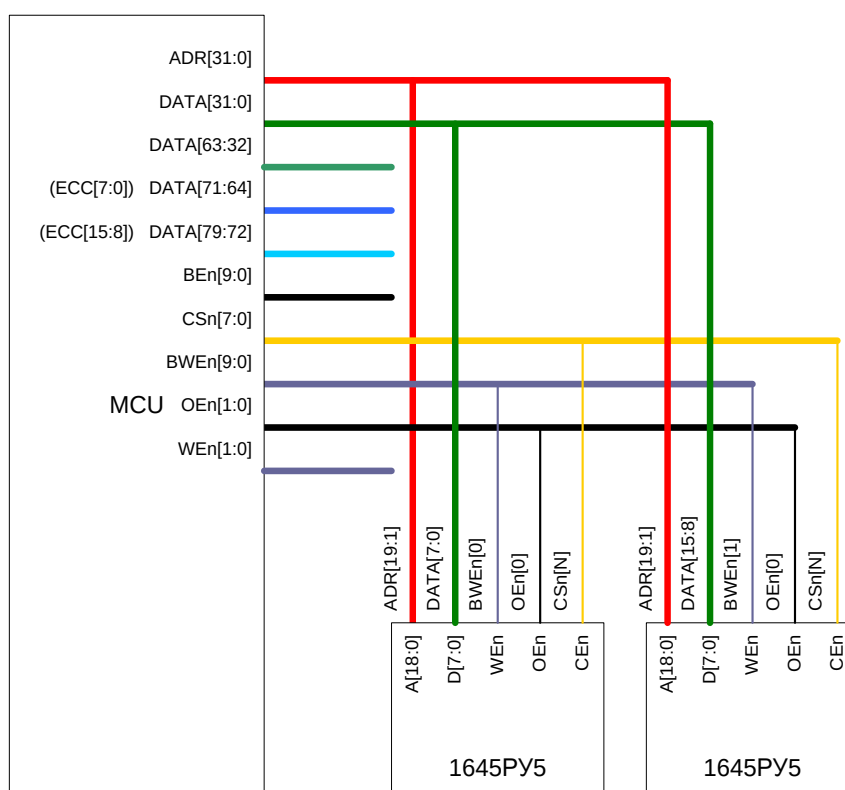


Рисунок 22 – Организация 16-ти битной шины с последовательной организацией SECC8 или SECC16 или без ECC на базе ОЗУ 1645PY5
(объем пользовательских данных до 0,8 Мбайт с последовательной SECC8, до 0,6 Мбайт с последовательной SECC16 и до 1 Мбайт без ECC)

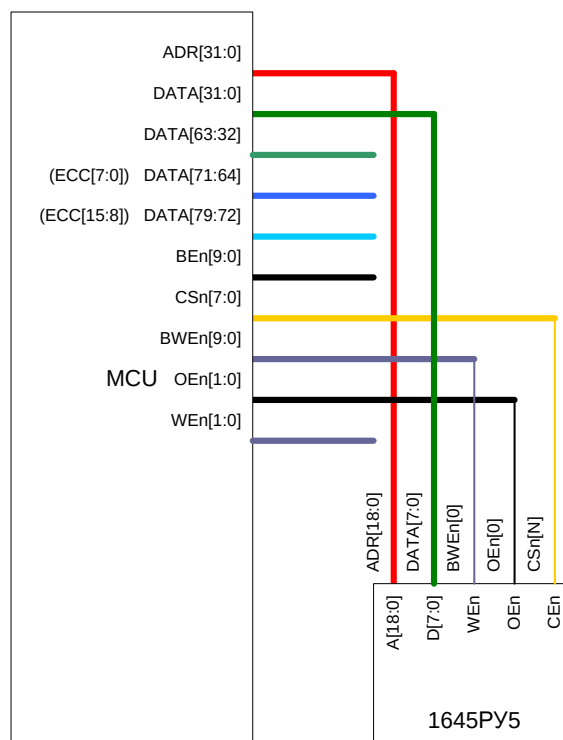


Рисунок 23 – Организация 8-ми битной шины с последовательной организацией SECC8 или SECC16 или без ECC на базе ОЗУ 1645PY5 (объем пользовательских данных до 0,4 Мбайт с последовательной SECC8, до 0,33 Мбайт с последовательной SECC16 и до 0,5 Мбайт без ECC)

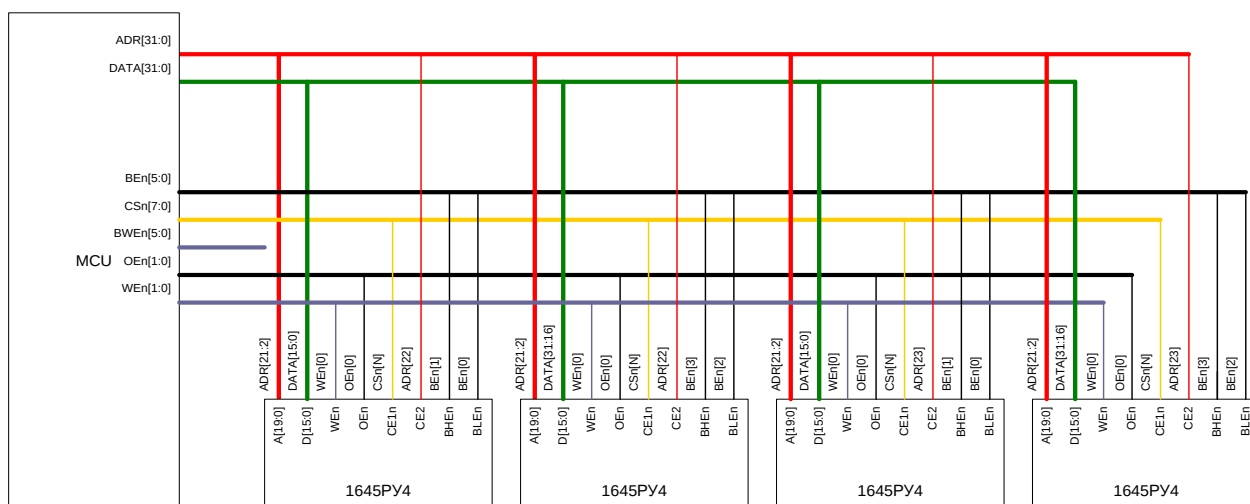


Рисунок 24 – Организация 32-битной шины с последовательной SECC8 или SECC16 или без ECC на базе ОЗУ 1645PY4 (объем пользовательских данных до 6,4 Мбайт с SECC8, до 5,3 Мбайт с SECC16 или до 8 Мбайт без ECC)

5.6.6.7 Организация внешней системной шины с ожиданием сигнала готовности

Для каждого диапазона внешней системной шины может быть настроен режим работы с ожиданием сигнала готовности READY. В этом случае транзакция на внешней системной шине может быть начата только при пассивном (шина готова) состоянии сигнала READY. При этом будут выполнены фазы предустановки сигналов выборки, адреса/данных перед сигналами записи/чтения и вход в фазу записи/чтения. Переход из фазы записи/чтения в фазу удержания адреса/данных 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, сигналом готовности может быть только увеличена фаза записи/чтения. Для всех диапазонов, работающих с ожиданием внешнего сигнала, может быть задана только одна полярность пассивного/активного состояния. Для каждого диапазона, работающего с ожиданием внешнего сигнала готовности, может быть программно задана максимальная длительность времени ожидания сигнала готовности (перед началом транзакции и нахождения в фазе записи/чтения). Если транзакция успешно не выполняется (без учета корректности ECC) за максимальное время ожидания, 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.

5.7 Организация доступа DMA

Контроллер прямого доступа в память (DMA) является инициатором транзакций на шине AHB. Контроллер DMA формирует шину MBUS. На входе/выходе контроллера DMA генерируется и проверяется ECC, аналогично шинам процессорного ядра и контроллера CPUIF. При работе с другими блоками системы по чтению и записи данных DMA контроллер функционирует аналогично процессорному ядру. Настройка и управление контроллера DMA осуществляется в регистрах периферии.

5.8 Организация доступа со стороны интерфейса CPUIF

Микросхема может быть подключена к внешнему процессору через интерфейс CPUIF. Внешний процессор может осуществлять обращения ко всем ресурсам микросхемы (см. рисунок 12). Также через контроллер внешней шины MEMIF внешний процессор может осуществлять обращения к внешней памяти. Интерфейс CPUIF может быть включен при загрузке микросхемы после сброса в режимах MODE = 1 и 2, либо включен в ходе выполнения пользовательской программы на встроенном микроконтроллерном ядре.

При выборе CS_CFGn при обращении по параллельной шине возможен доступ к регистрам управления микросхемой для изменения режима работы и обработки различных сбойных ситуаций. При выборе CS осуществляется доступ через микросхему в массив памяти.

Время Transaction должно быть выбрано таким образом, чтобы за это время на MEMIF были выполнены все обращения необходимые для завершения транзакции (см. рисунок 25).



При подключении к интерфейсу CPUIF возможно три варианта ширины шины данных: 32, 16 и 8 бит.

Для каждой ширины шины данных биты CDATE маски CNBE должны быть установлены в значение, соответствующее активным байтам для определения размера транзакции на шине.

Таблица 31 – Разрешенные комбинации битов маски CNBE

CNBE	Ширина шины		
	32	16	8
4'b0000	о	х	х
4'b1100	о	о	х
4'b0011	о	х	х
4'b1110	о	о	о
4'b1101	о	о	х
4'b1011	о	х	х
4'b0111	о	х	х
Обозначения в таблице: о – комбинация разрешена; х – комбинация запрещена.			

5.8.1 Формирование адреса транзакции

5.8.1.1 Формирование адреса транзакции для микросхем первой ревизии

При прямом и косвенном обращении финальный адрес транзакции определяется содержимым регистра ABASE.

BASE	ANMASK
[31:5]	[4:0]

Поле ANMASK содержит номер самого старшего немаскируемого бита адреса. Не может быть установлено в значение меньше 4. Т.е сигналы CADDR[4:0] должны всегда управляться при обращениях к интерфейсу.

Пример:

В регистр ABASE произведена запись 0xABCDEF0B. При обращениях через мост результирующий адрес обращения будет {0xABCDE,CADDR[11:0]}.

5.8.1.2 Формирование адреса транзакции для микросхем, начиная со второй ревизии

При прямом и косвенном обращении финальный адрес транзакции определяется содержимым регистра ABASE и AMASK.

Адрес обращения на шине AHB haddr определяется содержимым регистров ABASE и AMASK и вычисляется из адреса addr обращения на асинхронной шине по формуле:

$$haddr[i] = (AMASK[i]) ? ABASE[i] : addr[i], i \in [31..2]$$

Младшие разряды адреса haddr[1:0] рассчитываются, исходя из активных сигналов nbe и младших разрядов адреса addr[1:0], но также могут быть замаскированы

AMASK и выбраны из ABASE. Маскирование разрядов haddr[1:0] обязательно в косвенном режиме

BASE
[31:2]

5.8.2 Описание регистров моста

При активном уровне сигнала CNCS_CFG происходит обращение к регистрам моста.

Таблица 32 – Описание регистров моста

CADDR[3:2]	Имя регистра	Описание
2'b00	ABASE	Регистр базы для трансляции адресов на шину AHB
2'b01	AMASK	Битовая маска для регистра ABASE Примечание – регистр доступен, начиная с ревизии 2
2'b10	ADATA	Регистр данных для модификации ширины обращений 8-, 16-разрядной шины в 32 бит обращения по шине AHB

Описание регистра ABASE см. в подразделе 5.8.1.

5.8.2.1 Регистр ADATA

Регистр ADATA предназначен для модификации ширины обращений через 8-, 16-разрядную шину к 32-битным устройствам на шине AHB (см. рисунок 26).

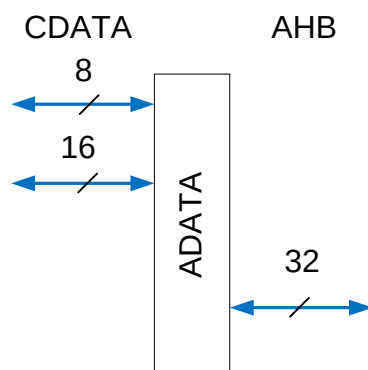


Рисунок 26 – Режим тестового раскрытия блоков основной памяти ROM

Регистр позволяет последовательно загрузить/выгрузить по 8-битным байтам и/или 16-битным словам одно 32-битное слово. Обращение в младший байт или 16-битное слово генерирует транзакцию на шине AHB, используя регистр ADATA в качестве буфера для данных.

5.9 Организация доступа к периферии

При работе с периферией на шине APB осуществляется проверка ECC, но не все периферийные блоки ее поддерживают в полном объеме. Проверка ECC проводится для блоков с большой внутренней памятью, таких как CAN.

5.10 Периферийные блоки

Периферийные блоки содержат элементы диагностики, обнаружения, исправления сбоев и обнаружения отказов. В зависимости от функции, выполняемой блоком, система защиты в блоке реализуется по-разному. В ряде блоков для защиты используются механизмы защиты, реализованные на уровне стандарта.

5.10.1 Контроллер тактовых частот (CLKCNTR)

Блок управления тактовыми сигналами является важным элементом обеспечения работоспособности микроконтроллера, так как при отсутствии тактовых сигналов микроконтроллер не может выполнять возложенные на него функции. При выполнении загрузочной программы из встроенной ПЗУ и при запуске пользовательской программы, микроконтроллер тактируется встроенным HSI-генератором. HSI-генератор считается надежным и стабильным – отказ HSI-генератора приводит к полному отказу микроконтроллера. HSI-генератор используется для определения значения частоты 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.

HSI-генератор может быть программно отключен для снижения энергопотребления. Программное отключение HSI-генератора не контролируется блоком Checker и автоматически ведет сначала к незначительному, а затем к значительному увеличению тактовой частоты контролируемых источников. Если по какому-либо из этих событий настроено аппаратное переключение на HSI-генератор, то это приведет к остановке тактирования, а если настроен аппаратный сброс, то и к сбросу микроконтроллера.

Структурная схема блока Checker представлена на рисунке 27.

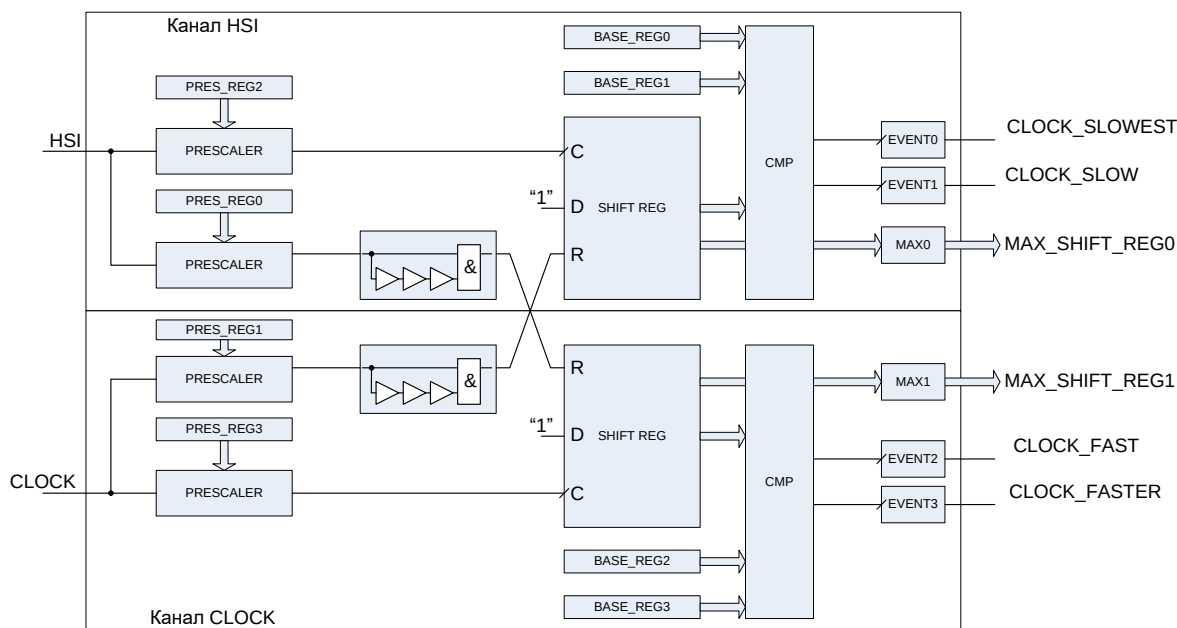


Рисунок 27 – Схема блока Checker для контроля тактовых частот

При настройке блока, исходя из ожидаемых значений частот сигналов HSI и CLOCK, необходимо задать значения предварительных делителей в регистрах **PRES_REGx** так, чтобы:

При $f_{HSI} < f_{CLOCK}$:

$$\begin{cases} \frac{f_{HSI}}{PRES_REG2 + 1} = K_0, \text{ где } K_0 \in [1..15] \\ \frac{f_{CLOCK}}{PRES_REG1 + 1} \\ \frac{f_{CLOCK}}{PRES_REG3 + 1} = K_1, \text{ где } K_1 \in [1..15] \\ \frac{f_{HSI}}{PRES_REG0} \end{cases} \quad (4)$$

При $f_{HSI} > f_{CLOCK}$:

$$\begin{cases} \frac{f_{HSI}}{PRES_REG2 + 1} = K_0, \text{ где } K_0 \in [1..15] \\ \frac{f_{CLOCK}}{PRES_REG1} \\ \frac{f_{CLOCK}}{PRES_REG3 + 1} = K_1, \text{ где } K_1 \in [1..15] \\ \frac{f_{HSI}}{PRES_REG0 + 1} \end{cases} \quad (5)$$

Тогда в регистре **SHIFTREG** канала HSI будут возникать значения от 0 до K_0 , а в канале CLOCK – от 0 до K_1 .

При снижении частоты CLOCK:

- в регистре канала HSI будут возникать значения от 0 до K_2 , причем $K_2 > K_0$;
- в регистре канала CLOCK будут возникать значения от 0 до K_3 , причем $K_3 < K_1$.

Если значение K_2 превысит значение **BASE_REG0**, возникнет событие EVENT0, а если K_3 превысит значение **BASE_REG1** – событие EVENT1.

$$\begin{cases} f_{CLOCK} < \frac{f_{HSI} \times PRES_REG1^*}{PRES_REG2 \times BASE_REG0} \Rightarrow EVENT0 \\ f_{CLOCK} < \frac{f_{HSI} \times PRES_REG1^*}{PRES_REG2 \times BASE_REG1} \Rightarrow EVENT1 \end{cases} \quad (6)$$

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

При увеличении частоты CLOCK:

- в регистре канала HSI будет возникать значения от 0 до K_2 , причем $K_2 < K_0$;
- в регистре канала CLOCK будет возникать значения от 0 до K_3 , причем $K_3 > K_1$.

Если значение K_3 превысит значение **BASE_REG2**, возникнет событие EVENT2, а если K_3 превысит значение **BASE_REG3** – событие EVENT3.

$$\begin{cases} f_{CLOCK} > \frac{f_{HSI} \times PRES_REG3 \times BASE_REG2}{PRES_REG0^*} \Rightarrow EVENT2 \\ f_{CLOCK} > \frac{f_{HSI} \times PRES_REG3 \times BASE_REG3}{PRES_REG0^*} \Rightarrow EVENT3 \end{cases} \quad (7)$$

* $PRES_REG0$ 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 HSI-генератора.

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

- сигнал предупреждения;
- аппаратный сброс микроконтроллера.

Для нормальной работы блока контроля частоты необходимо после установки бита разрешения контроля произвести сброс всех событий и содержимого сдвиговых регистров, поскольку сразу после включения сдвиговые регистры принимают некорректные значения.

Для иллюстрации принципов работы блока возможные комбинации настроек приведены в таблице 33.

Таблица 33 – Примеры работы блока Checker

Частота HSI	Ожидаемая частота CLOCK	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	1	100	1	12	15	12	15	K0 = 10 K1 = 10 Нет событий
8 МГц	80 МГц	100 МГц	1	1	100	1	12	15	12	15	K0 = 10 K1 = 10 K2 = 8 K3 = 12 EVENT2Fast Clock
8 МГц	80 МГц	120 МГц	1	1	100	1	12	15	12	15	K0 = 10 K1 = 10 K2 = 6 K3 = 15 EVENT2Fast Clock EVENT3Faster Clock
8 МГц	80 МГц	66 МГц	1	1	100	1	12	15	12	15	K0 = 10 K1 = 10 K2 = 12 K3 = 7 EVENT0Slow Clock
8 МГц	80 МГц	0 МГц (нет частоты)	1	1	100	1	12	15	12	15	K0 = 10 K1 = 10 K2 = 15(∞) K3 = 0 EVENT0Slow Clock EVENT1Slowest Clock

Частота HSI	Ожидаемая частота CLOCK	Реальная частота CLOCK	PRES REG2	PRES REG0	PRES REG1	PRES REG3	BASE REG0	BASE REG1	BASE REG2	BASE REG3	Примеры
8 МГц	32 кГц	32 кГц	48	1600	1	1	12	15	12	15	K0 = 5 K1 = 6 Нет событий
8 МГц	32 кГц	60 кГц	48	1600	1	1	12	15	12	15	K0 = 5 K1 = 6 K2 = 3 K3 = 12 EVENT 2Fast Clock
8 МГц	32 кГц	75 кГц	48	1600	1	1	12	15	12	15	K0 = 5 K1 = 6 K2 = 2 K3 = 15 EVENT 2Fast Clock EVENT 3Faster Clock
8 МГц	32 кГц	15 кГц	48	1600	1	1	12	15	12	15	K0 = 5 K1 = 6 K2 = 11 K3 = 3 EVENT 0Slow Clock
8 МГц	32 кГц	0 кГц (нет частоты)	48	1600	1	1	12	15	12	15	K0 = 5 K1 = 6 K2 = 15(∞) K3 = 0 EVENT0Slow Clock EVENT1Slowest Clock

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.

Блок Checker установлен на входах тактовых частот от внешних источников HSE0, HSE1 и LSE, встроенного LSI генератора и всех PLL.

Блоки Checker проверяют наличие и значения частот тактовых сигналов, поступающих только в цифровую часть микроконтроллера. Блок часов реального времени RTC может быть сконфигурирован на использование тактовой частоты от LSE или LSI генератора, 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.

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

Блок CLK_CNTR формирует тактовые частоты для процессорных ядер и периферии. Система тактовой синхронизации построена по принципу полной синхронности основных элементов, связанных с процессорным ядром (блоки памяти

ОЗУ, DMA контроллер, мосты, и интерфейсные по отношению к процессору части всех периферийных блоков). Т.е. 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 процессорное ядро и связанные с ним блоки также могут работать на прореженной частоте (все на одной и той же). Принцип формирования и соотношения тактовых частот в микроконтроллере приведен на рисунке 28.

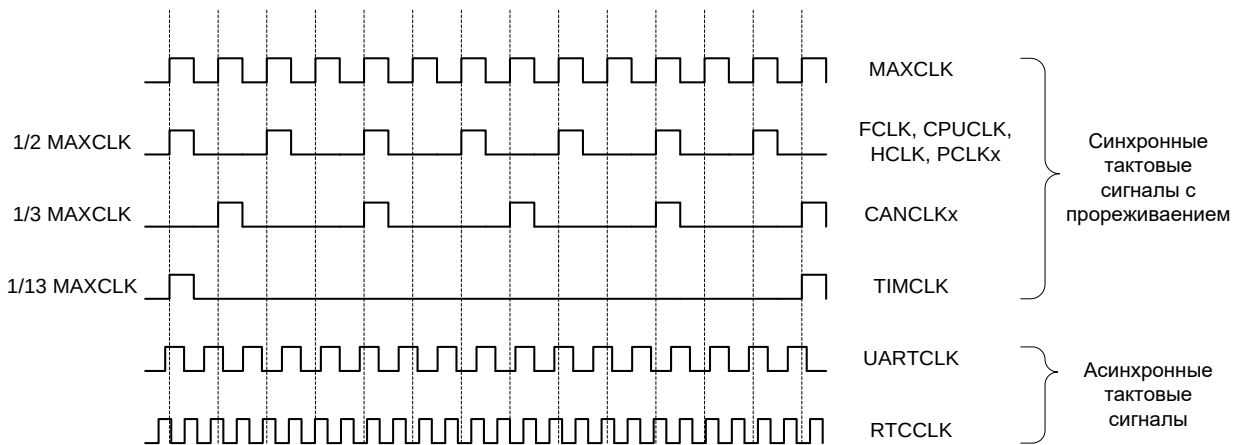


Рисунок 28 – Пример тактовых частот в микроконтроллере

В ряде периферийных блоков (SSP, UART) вводятся ограничения на соотношения рабочих частот этих блоков и частоты PCLK интерфейса с процессорным ядром. CPUCLK может совпадать с MAXCLK, но не должен превышать 50 МГц.

5.10.2 Контроллер PWR_BKP

В микроконтроллере реализован блок PWR_BKP, включающий в себя 256 байт памяти BKP_MEM, часы реального времени RTC и генератор LSE. 16 байт памяти BKP_MEM предназначены для хранения и управления работой микроконтроллера. В том числе управление питанием микросхемы, режимами работы и т.п.

5.10.3 Контроллер детектора напряжения питания (PWRCNTR)

В состав микроконтроллера входит блок детектора напряжения питания (блок PWR_CNTR). С помощью блока PWR_CNTR можно определить уровень напряжения питания U_{cc} с точностью ± 100 мВ, а также определить ток потребления доменов питания DU_{cc} с точностью ± 5 мА.

Для определения напряжения питания U_{cc} в блоке задается по два уровня сравнения для каждого из питаний. И при превышении первого заданного уровня или снижении ниже второго заданного уровня вырабатываются события.

Значение тока потребления доменов питания DU_{cc} отображается в регистре DUCCURR для каждого LDO регулятора. Для каждого LDO регулятора в контроллере PWR_CNTR может быть настроен максимально допустимый уровень потребления, при превышении которого вырабатывается событие.

В микроконтроллере реализован аппаратный блок защиты от перенапряжения POVR, который вырабатывает сигнал сброса OVRSTn при превышении питанием U_{ss} уровня U_{POVR} .

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

Все события, критические для функционирования и безопасности выполнения пользовательской программы, обрабатываются в контроллере обработки сбоев FT_CNTR. В зависимости от настроек контроллера FT_CNTR возникновение события может вызвать сигнал прерывания или сброс микроконтроллера.

Для каждого события в блоке FT_CNTR возможна настройка реакции:

- игнорирование;
- прерывание (для группы с высокой важностью FT_IF0, средней FT_IF1 и низкой FT_IF2);
- отложенный сброс микроконтроллера.

При возникновении события, вызывающего отложенный сброс микроконтроллера, в контроллере FT_CNTR автоматически запускается таймер отложенного сброса. Период времени таймера отложенного сброса задается программно и формирует временной интервал для выполнения действий по парированию возникшей ошибки.

5.10.5 Контроллер сторожевого таймера (WDTCNTR)

В микроконтроллере реализован сторожевой таймер, при использовании которых, микроконтроллер может быть сброшен при несоблюдении временных интервалов обязательного сброса сторожевых таймеров.

5.10.6 Контроллер внешней шины (MEMIF)

Контроллер внешней шины имеет встроенные механизмы генерации и проверки ECC.

5.10.7 Контроллер Кэш-памяти (CACHCNTR)

Кэш-память защищена ECC.

5.10.8 Контроллер ПЗУ (ROMCNTR)

Не имеет функций управления, только регистр учета ошибок ECC.

5.10.9 Контроллер ОЗУ (RAMCCNTR и RAMDCNTR)

Не имеет функций управления, только регистр учета ошибок ECC.

5.10.10 Контроллер DMA (DMACNTR)

При работе контроллера DMA все транзакции защищены ECC.

5.10.11 Контроллер портов ввода-вывода (PORTx)

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 генерируется

прерывание от порта. Для исправления ситуации необходимо программно перевести вывод в третье состояние или, в случае, если перегрузки кратковременны, сбросить флаг записью 1 в соответствующий бит регистра HCUR.

Каждый вывод имеет до 16-ти различных функциональных назначений. Функциональное назначение может задаваться индивидуально для каждого вывода в регистрах FUNC0, FUNC1, FUNC2 и FUNC3.

5.10.12 Контроллер SpaceWire (SpaceWire)

Память блока контроллера SpaceWire имеет встроенную ECC-защиту. Встроенный PHY обеспечивает работу на скорости до 100 Мбит/с.

5.10.13 Контроллер таймеров общего назначения (TIMER32x)

Таймер позволяет формировать различные временные интервалы и контролировать длительность внешних сигналов.

5.10.14 Контроллер CAN (CANx)

Контроллер CAN позволяет передавать данные согласно стандарту ISO 11898.

5.10.15 Контроллер SSP (SSPx)

Контроллер позволяет последовательно передавать данные в ведомом и ведущем синхронном режиме.

5.10.16 Контроллер UART (UARTx)

Контроллер позволяет последовательно передавать данные в асинхронном режиме.

5.10.17 Контроллер интерфейса МКПД (MILx)

Контроллер MIL позволяет передавать данные согласно стандарту ГОСТ Р 52070-2003.

5.11 Интерфейс отладки

Для отладки приложений используется интерфейс JTAG/SWD.

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

Т а б л и ц а 34 – 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]

5.12 Тестовые режимы работы

Микросхема переключается в тестовый режим при высоком уровне на выводе JTAG_EN или MODE = 1111.

В тестовом режиме к выводам JTAG_A подключается тестовый TAP-контроллер. Загрузкой инструкций IR в TAP-контроллер микросхема переводится в тестовые режимы, указанные в таблице 35. При переходе в тестовый режим, работа микроконтроллера останавливается. Все сигналы интерфейсов определённого тестового режима должны находиться в неактивном состоянии до момента включения соответствующего тестового режима.

Таблица 35 – Тестовые режимы работы

IR[7:0]	TYPE	TTMODE	Название тестового режима	Описание
0x00			Boundary Scan	
			IDCODE	
0x7F				
0x01	SCAN	0	MAIN_ATPG	Режим тестирования основной логики МК
0x02		1	-	
0x03		2	LS_OPEN	Режим раскрытия и тестирования LSE, LSI
0x04		3	HSE_OPEN	Режим раскрытия HSE, PLL
0x05		4	-	
0x06		5	-	
0x07	MEM	6	-	
0x08	MEM	7	SRANC_OPEN	Режим раскрытия SRAMC
0x09	MEM	8	SRAMD_OPEN	Режим раскрытия SRAMD
0x0A	MEM	9	ROM_OPEN	Режим раскрытия ROM
0x0B		10		
0x0C		11	SPW_USB_OPEN	Режим раскрытия SPW и USBPHY
0x0D	MEM	12		
0x8D	MEM	13		
0x0E	MEM	14	MIL_MEM_OPEN	Режим раскрытия MIL памяти
0x11		15		
0x13		16	LDO_OPEN	Режим раскрытия LDO
0x17	MEM	17	CAN_MEM_OPEN	Режим раскрытия CAN памяти
0x18		18		
			Резерв	

5.12.1 Режим тестирования ОЗУ

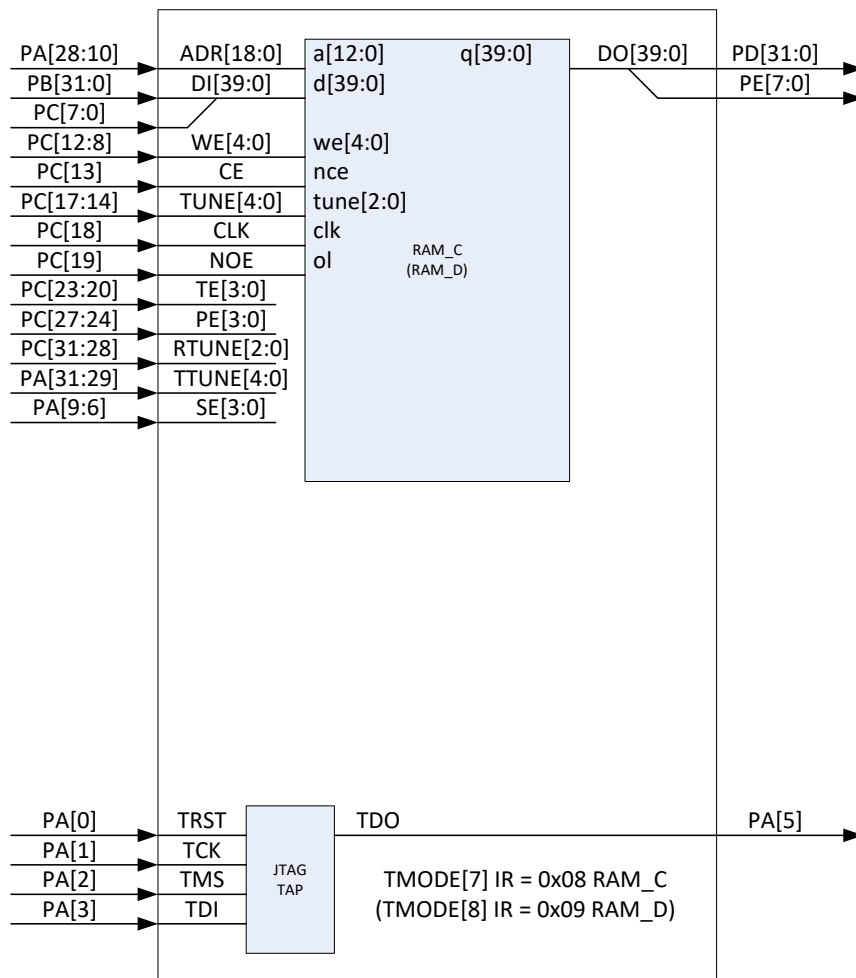


Рисунок 29 – Режим тестового раскрытия блоков основной памяти RAM

5.12.2 Режим тестирования Масочного ПЗУ (BOOTROM)

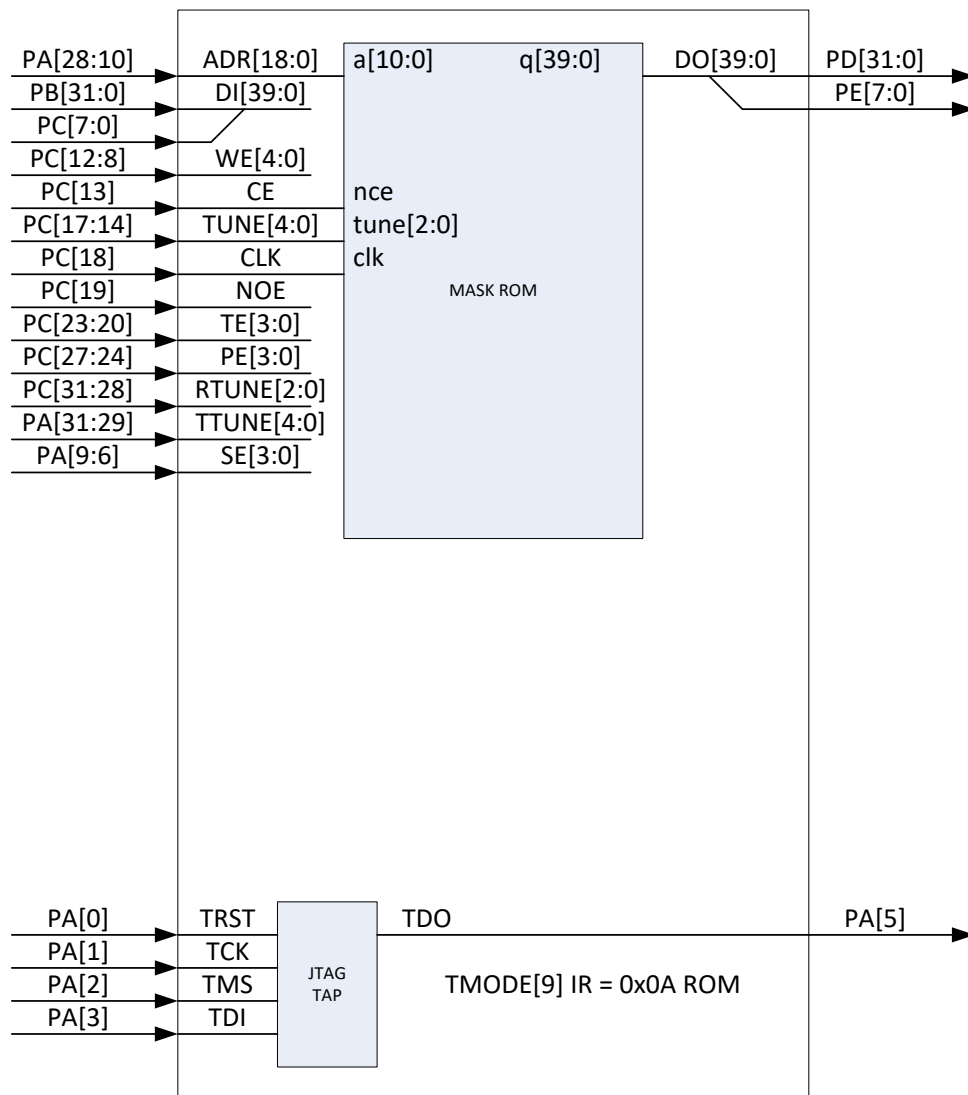


Рисунок 30 – Режим тестового раскрытия блоков основной памяти ROM

5.12.3 Режим тестирования RAM_CAN

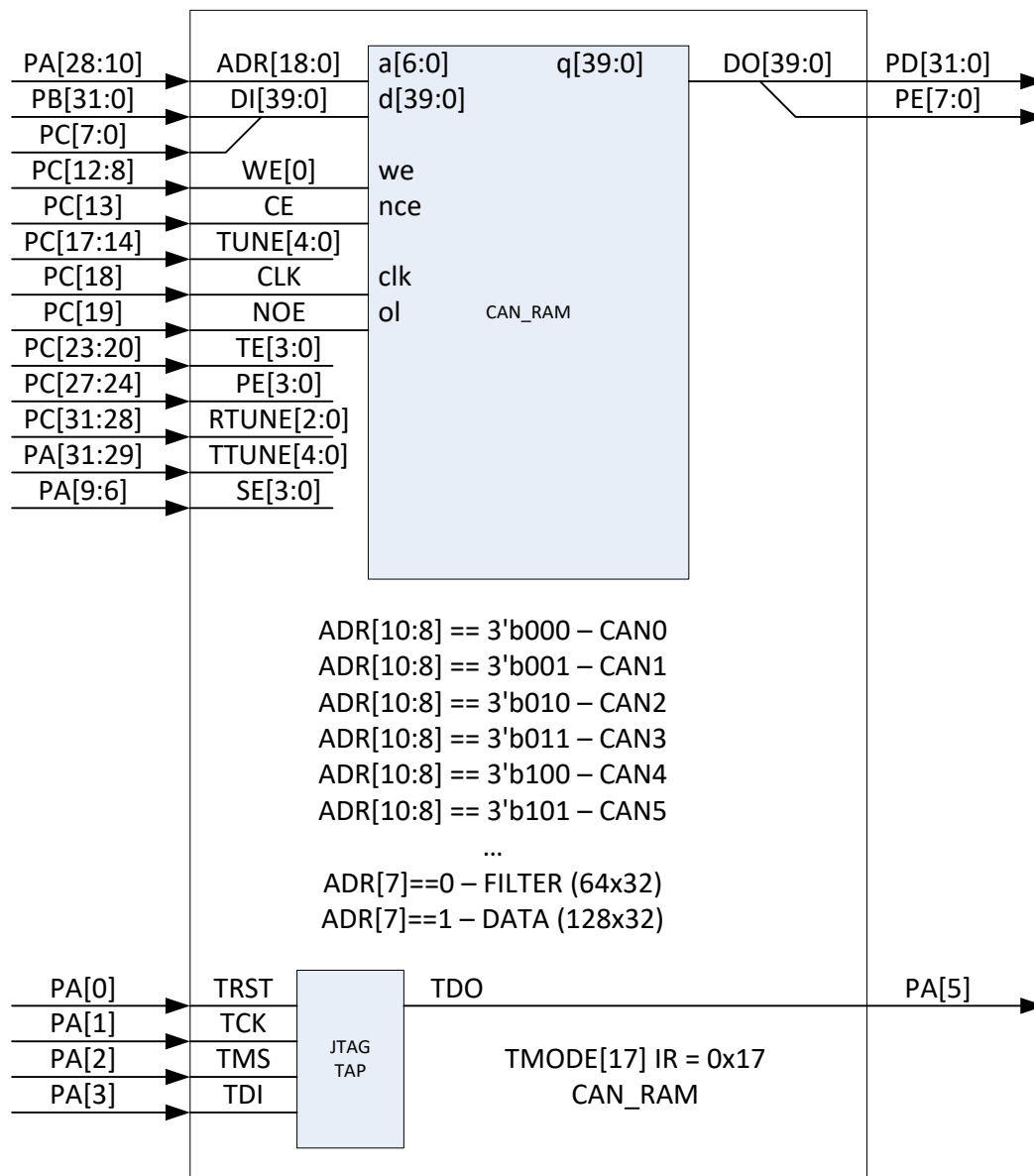


Рисунок 31 – Режим тестового раскрытия блоков памяти CAN

5.12.4 Режим тестирования RAM_MIL

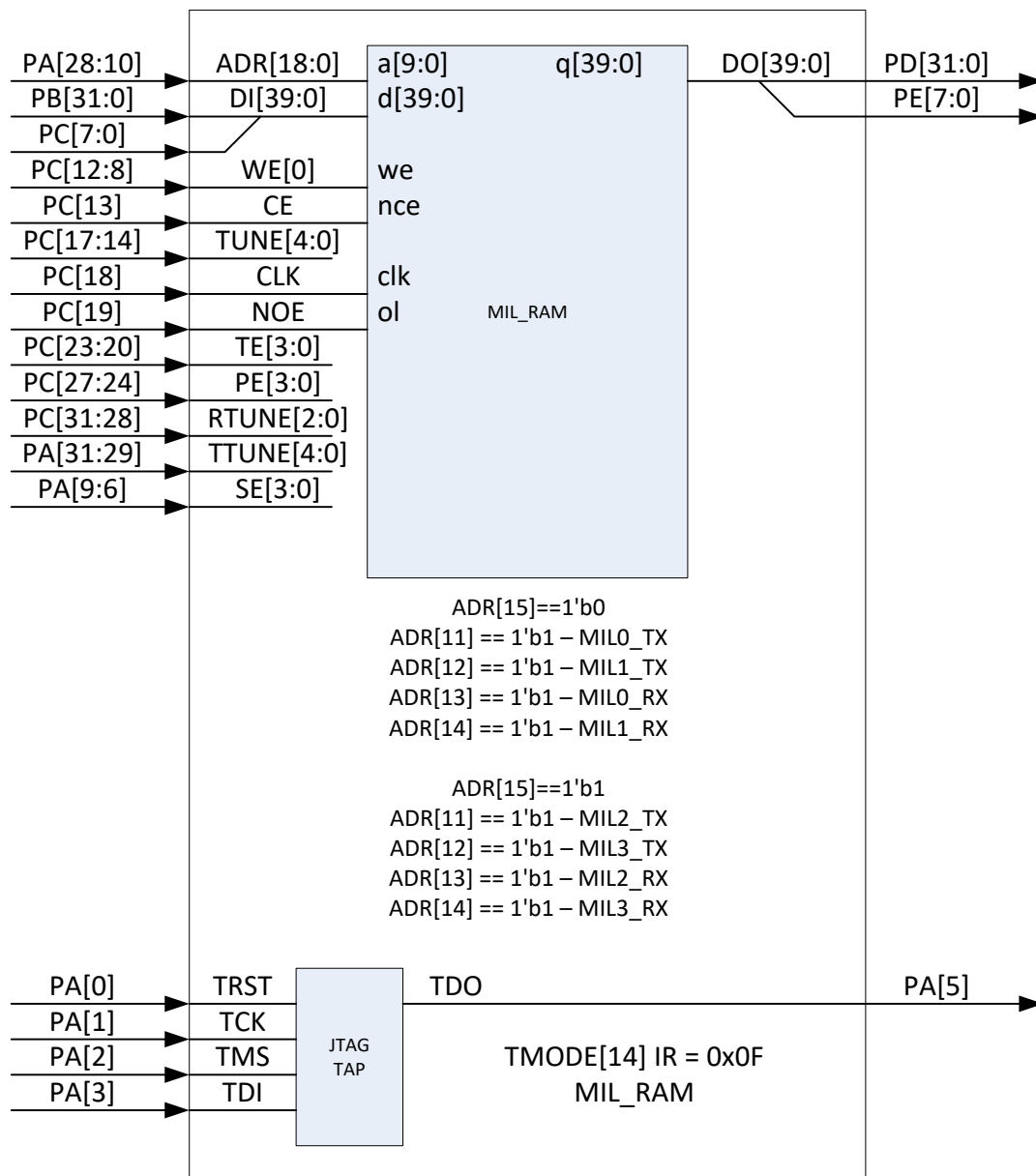


Рисунок 32 – Режим тестового раскрытия блоков памяти MIL

5.12.5 Режим тестирования генераторов и PLL

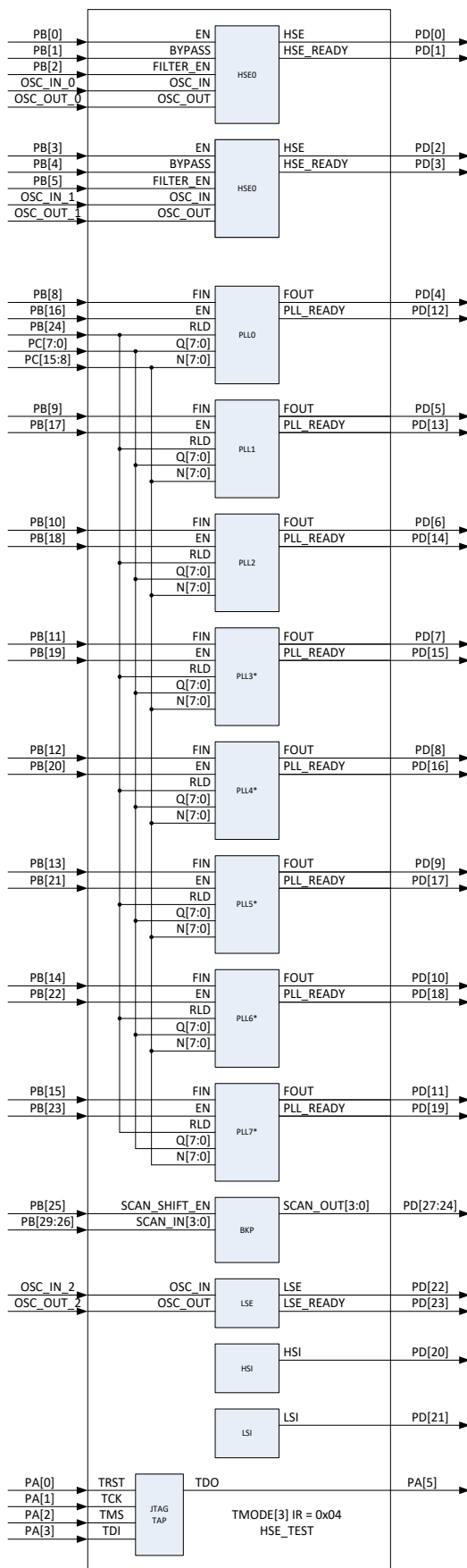


Рисунок 33 – Режим тестового раскрытия блоков PLL и генераторов

5.12.6 Режим тестирования приемопередатчиков SPW

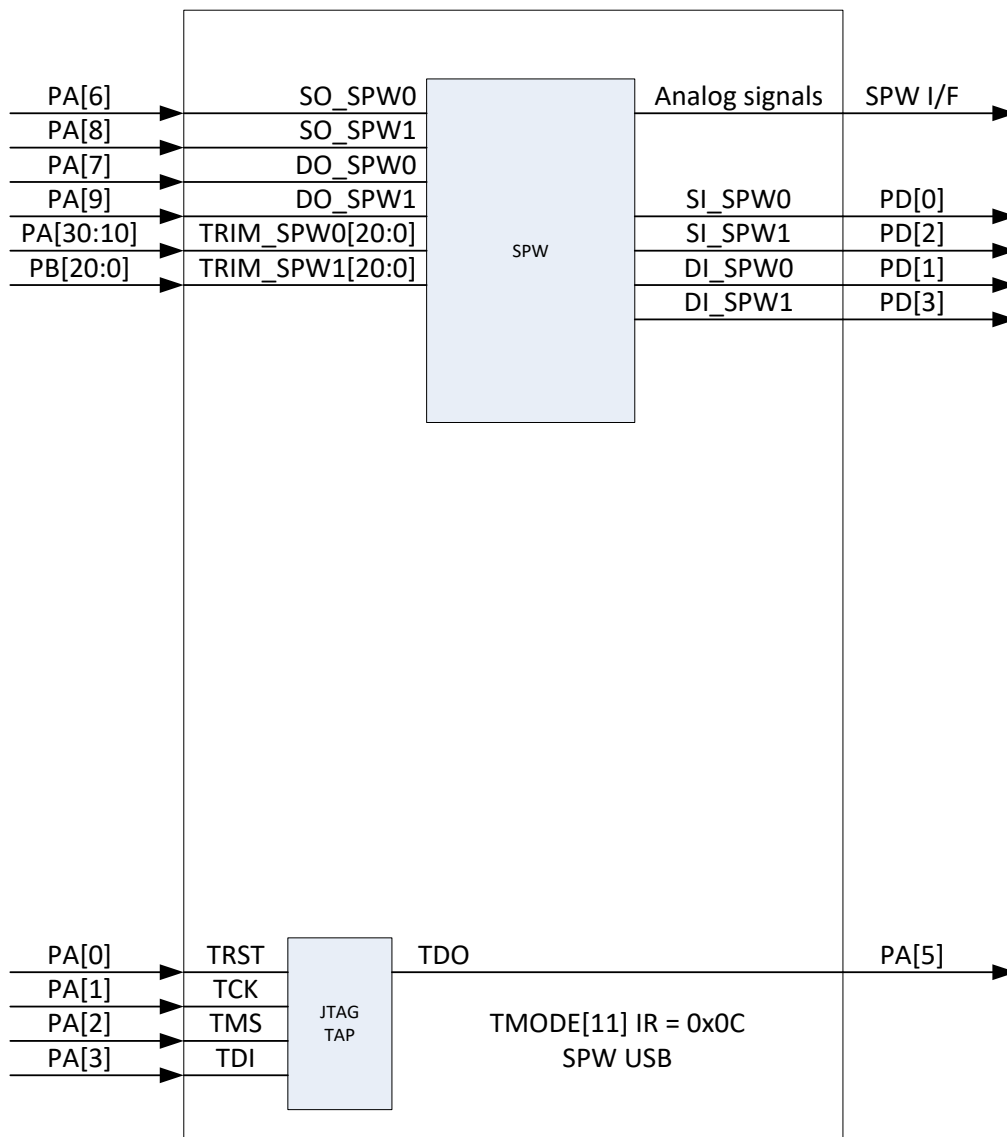


Рисунок 34 – Режим тестового раскрытия блоков передатчиков SpaceWire

5.12.7 Режим тестирования PMU (VoltageDetect)

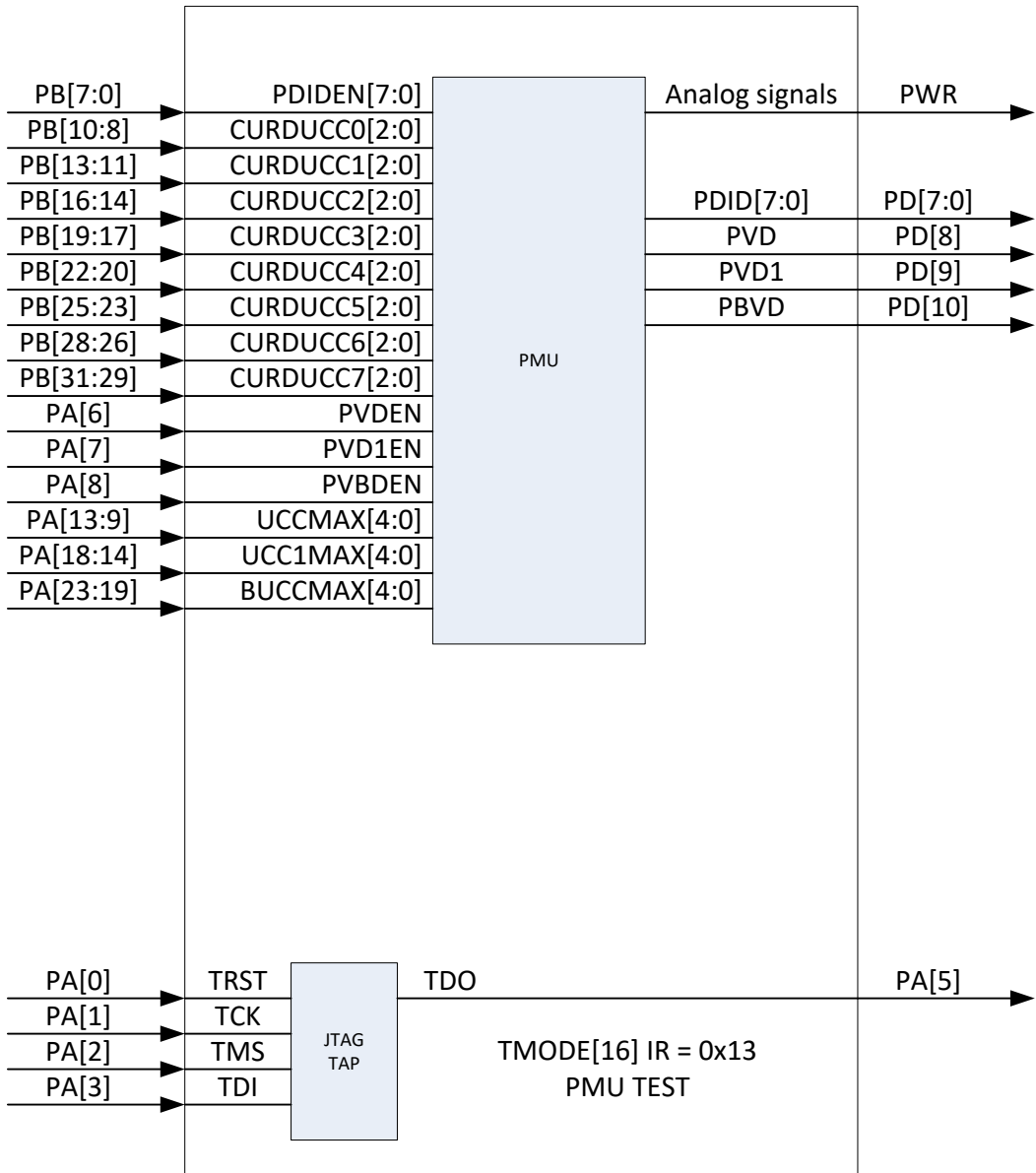


Рисунок 35 – Режим тестового раскрытия блок управления питанием

6 Программная модель микроконтроллера

6.1 Контроллер тактовых частот (CLKCNTR)

6.1.1 Описание регистров

Таблица 36 – Описание регистров

Базовый адрес	Смещение	Обозначение	Состояние после сброса	Описание
0x4000_0000	0x0000_0000	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_002	CPU_STAT		Регистр состояния тактовой частоты процессорных ядер
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

Базовый адрес	Смещение	Обозначение	Состояние после сброса	Описание
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
Блок умножения PLL0				
	0x0000_0078	PLL0_CLK		Регистр настройки PLL0

Базовый адрес	Смещение	Обозначение	Состояние после сброса	Описание
	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
Тактирование CAN				
	0x0000_0118	CAN0_CLK		Управление тактированием контроллера CAN0
	0x0000_011C	CAN1_CLK		Управление тактированием контроллера CAN1
Тактирование TIMER				
	0x0000_0130	TIM0_CLK		Управление тактированием контроллера TIMER0
	0x0000_0134	TIM1_CLK		Управление тактированием контроллера TIMER1

Базовый адрес	Смещение	Обозначение	Состояние после сброса	Описание
	0x0000_0138	TIM2_CLK		Управление тактированием контроллера TIMER2
	0x0000_013C	TIM3_CLK		Управление тактированием контроллера TIMER3
Тактирование MIL (МКПД)				
	0x0000_0148	MIL0_CLK		Зарезервировано
	0x0000_014C	MIL1_CLK		Зарезервировано
	0x0000_0150	MIL2_CLK		Зарезервировано
	0x0000_0154	MIL3_CLK		Зарезервировано
Тактирование SPACE WIRE				
	0x0000_0178	SPW0_CLK		Управление тактированием контроллера SpaceWire 0
	0x0000_017C	SPW1_CLK		Управление тактированием контроллера SpaceWire 1
Тактирование UART				
	0x0000_0188	UART0_CLK		Управление тактированием контроллера UART 0
	0x0000_018C	UART1_CLK		Управление тактированием контроллера UART 1
Тактирование SSP				
	0x0000_01A0	SSP0_CLK		Управление тактированием контроллера SSP 0
	0x0000_01A4	SSP1_CLK		Управление тактированием контроллера SSP 1
	0x0000_01A8	SSP2_CLK		Управление тактированием контроллера SSP 2
	0x0000_01AC	SSP3_CLK		Управление тактированием контроллера SSP 3
	0x0000_01B0	SSP4_CLK		Управление тактированием контроллера SSP 4
	0x0000_01B4	SSP5_CLK		Управление тактированием контроллера SSP 5
Тактирование RTC				
	0x0000_01CC	RTC_CLK		Управление тактированием контроллера RTC

6.1.1.1 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]															

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

6.1.1.2 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
Зарезервировано															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Зарезервировано												SELECT[3:0]			

Бит	Имя	Значение	Описание
31...4	-	-	Зарезервировано
3...0	SELECT[3:0]		Номер источника тактовой частоты 00000 – Генератор HIS 00001 – Генератор HIS/2 00010 – Генератор HSE0 00011 – Генератор HSE0/2 00100 – Генератор HSE1 00101 – Генератор HSE1/2 00110 – Генератор LSI 00111 – Генератор LSE 01000 – Генератор PLL0 01001 – Генератор PLL1 01010 – Генератор PLL2 01011 – Генератор PLL3 01100 – Генератор PLL4 01101 – Генератор PLL5 01110 – Генератор PLL6 01111 – Генератор PLL7

6.1.1.3 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]															

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
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

Бит	Имя	Значение	Описание
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 FCLK = MAX_CLK/(DIV+1)

6.1.1.4 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_CAN1_EN	MDR_CAN0_EN			MDR_TMR3_EN	MDR_TMR2_EN	MDR_TMR1_EN	MDR_TMR0_EN	MDR_SPW1_EN	MDR_SPW0_EN				MDR_PORTE_EN	MDR_PORTD_EN

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

Бит	Имя	Значение	Описание
31	-		
30	MDR_CAN1_EN		Бит разрешения тактирования частотой CAN1 0 – тактирование запрещено 1 – тактирование разрешено
29	MDR_CAN0_EN		Бит разрешения тактирования частотой CAN0 0 – тактирование запрещено 1 – тактирование разрешено
28, 27	-		

Бит	Имя	Значение	Описание
26	MDR_TMR3_EN		Бит разрешения тактирования частотой TMR4 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	MDR_SPW1_EN		Бит разрешения тактирования частотой SpaceWire 1 0 – тактирование запрещено 1 – тактирование разрешено
21	MDR_SPW0_EN		Бит разрешения тактирования частотой SpaceWire 0 0 – тактирование запрещено 1 – тактирование разрешено
20...18	-		
17	MDR_PORTE_EN		Бит разрешения тактирования частотой PORTE 0 – тактирование запрещено 1 – тактирование разрешено
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

6.1.1.5 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_SSP5_EN	MDR_SSP4_EN	MDR_ECC_EN	MDR_CRC_EN			MDR_MIL3_EN		MDR_MIL2_EN		MDR_MIL1_EN		MDR_MIL0_EN																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								MDR_UART1_EN	MDR_UART0_EN	MDR_SSP3_EN	MDR_SSP2_EN	MDR_SSP1_EN	MDR_SSP0_EN		

Бит	Имя	Значение	Описание
31	MDR_SSP5_EN		Бит разрешения тактирования частотой SSP5 0 – тактирование запрещено 1 – тактирование разрешено
30	MDR_SSP4_EN		Бит разрешения тактирования частотой SSP4 0 – тактирование запрещено 1 – тактирование разрешено
29	MDR_ECC_EN		Бит разрешения тактирования частотой ECC 0 – тактирование запрещено 1 – тактирование разрешено
28	MDR_CRC_EN		Бит разрешения тактирования частотой CRC 0 – тактирование запрещено 1 – тактирование разрешено
27, 26	-		
25	MDR_MIL3_EN		Бит разрешения тактирования частотой MIL3 0 – тактирование запрещено 1 – тактирование разрешено
24	-		
23	MDR_MIL2_EN		Бит разрешения тактирования частотой MIL2 0 – тактирование запрещено 1 – тактирование разрешено
22	-		
21	MDR_MIL1_EN		Бит разрешения тактирования частотой MIL1 0 – тактирование запрещено 1 – тактирование разрешено
20	-		
19	MDR_MIL0_EN		Бит разрешения тактирования частотой MIL0 0 – тактирование запрещено 1 – тактирование разрешено
18...8	-		
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	MDR_SSP3_EN		Бит разрешения тактирования частотой SSP3 0 – тактирование запрещено 1 – тактирование разрешено
4	MDR_SSP2_EN		Бит разрешения тактирования частотой SSP2 0 – тактирование запрещено 1 – тактирование разрешено
3	MDR_SSP1_EN		Бит разрешения тактирования частотой SSP1 0 – тактирование запрещено 1 – тактирование разрешено

Бит	Имя	Значение	Описание
2	MDR_SSP0_EN		Бит разрешения тактирования частотой SSP0 0 – тактирование запрещено 1 – тактирование разрешено
1, 0	-		

6.1.1.6 Регистр CPU_CHK0

Base ADDR=	0x4000_0000	Offset=	0x0000_0014												
REG Name:															
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]															

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 $CLK(SHIFT_REG0) = HSI / (PRES_REG2+1)$

6.1.1.7 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]															

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 $CLK(SHIFT_REG1) = FCLK / (PRES_REG1+1)$

6.1.1.8 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[7:0]								BASE_REG2[7:0]							

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

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

6.1.1.9 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[7:0]								MAX_CHK_SHIFT_REG0[7:0]							

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...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

6.1.1.10 Регистр LSI_CLK¹

Base ADDR=		0x4000_0000				Offset=		0x0000_0024										
REG Name:		LSI_CLK																
31	30	29	28	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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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
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

¹ Управление генератором LSI осуществляется через регистры батарейного домена.

Бит	Имя	Значение	Описание
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

6.1.1.11 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]															

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 $CLK(SHIFT_REG0) = HSI / (PRES_REG2 + 1)$

6.1.1.12 Регистр LSI_CHK1

Base ADDR=		0x4000_0000				Offset=		0x0000_002C							
REG Name:		CHIPID													
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]															

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 $CLK(SHIFT_REG1) = LSI / (PRES_REG1 + 1)$

6.1.1.13 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[7:0]								BASE_REG2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_REG1[7:0]								BASE_REG0[7:0]							

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

6.1.1.14 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_ERR	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[7:0]								MAX_CHK_SHIFT_REG0[7:0]							

Бит	Имя	Значение	Описание
31...22	-		Зарезервировано
21	LSI_ERR	-	Выход флага ошибки генератора 0 – нет сбоя 1 – есть сбой Внутренний RC элемент задержки при включенном LSION=1 и LSI_RDY=1 постоянно сбрасывается выходной частотой. Если по какой-либо причине частота пропадет или станет слишком малой, RC элемент успеет зарядиться до максимума и выдаст флаг ошибки генератора.
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...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

6.1.1.15 Регистр HSI_STAT¹

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_ERR	HSI_RDY					

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

Бит	Имя	Значение	Описание
31...22	-		Зарезервировано
21	HSI_ERR	-	Выход флага ошибки генератора 0 – нет сбоя 1 – есть сбой Внутренний RC элемент задержки при включенном HSION=1 и HSI_RDY=1 постоянно сбрасывается выходной частотой. Если по какой-либо причине частота пропадет или станет слишком малой, RC элемент успеет зарядиться до максимума и выдаст флаг ошибки генератора.
20	HSI_READY	-	Флаг готовности генератора HSI 0 – генератор не готов или выключен 1 – генератор готов
19...0	-	-	Зарезервировано

6.1.1.16 Регистр LSE_CLK¹

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
-															

¹ Управление генератором HSI осуществляется через регистры батарейного домена.

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
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

6.1.1.17 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]															

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 $CLK(SHIFT_REG0) = HSI / (PRES_REG2 + 1)$

6.1.1.18 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]															

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 $CLK(SHIFT_REG1) = LSE / (PRES_REG1 + 1)$

6.1.1.19 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[7:0]								BASE_REG2[7:0]							

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

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

6.1.1.20 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[7:0]								MAX_CHK_SHIFT_REG0[7:0]							

Бит	Имя	Значение	Описание
31...20	-		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
15...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

6.1.1.21 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
-															

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 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 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 Имеет смысл только при EN_CHK = 1
15...0	-		Зарезервировано

6.1.1.22 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]															

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 $CLK(SHIFT_REG0) = HSI / (PRES_REG2 + 1)$

6.1.1.23 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]															

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 $CLK(SHIFT_REG1) = HSEn / (PRES_REG1+1)$

6.1.1.24 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[7:0]								BASE_REG2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_REG1[7:0]								BASE_REG0[7:0]							

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

6.1.1.25 Регистр HSEn_STAT

Base ADDR=		0x4000_0000					Offset=		0x0000_0060							
REG Name:		HSEn_STAT							0x0000_0074							
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[7:0]								MAX_CHK_SHIFT_REG0[7:0]								

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...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

6.1.1.26 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_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]				

Бит	Имя	Значение	Описание
31...29	SELECT[2:0]		Номер источника тактовой частоты 000 – Генератор HSI 001 – Генератор HSI/2 010 – Генератор HSE0 011 – Генератор HSE0/2 100 – Генератор HSE1 101 – Генератор HSE1/2 110 и 111 – Зарезервировано
28	-		Зарезервировано
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 $FINT = FIN * (PLLn_N) / (PLLn_Q + 1)$ Значение PLLn_N может быть в диапазоне от 3 до 75 Полученная частота FINT после умножения на N и деления на Q+1 должна быть в диапазоне от 75 до 150 МГц. При PLLn_N = 0 соответствует умножению на 2
7...5	-		Зарезервировано
4	DV		Бит дополнительного деления PLL $FOUT = FINT / (DV + 1)$
3...0	PLLn_Q[3:0]		Коэффициент деления тактовой частоты PLLn $FINT = FIN * (PLLn_N) / (PLLn_Q + 1)$ Значение PLLn_Q может быть в диапазоне от 0 до 15

6.1.1.27 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]															

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 $CLK(SHIFT_REG0) = HSI / (PRES_REG2 + 1)$

6.1.1.28 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]															

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 $CLK(SHIFT_REG1) = PLLn / (PRES_REG1 + 1)$

6.1.1.29 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[7:0]								BASE_REG2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_REG1[7:0]								BASE_REG0[7:0]							

Бит	Имя	Значение	Описание
31...24	BASE_REG3[7:0]		Регистр сравнения частоты для определения сильного увеличения частоты (EVENT3)
23...16	BASE_REG2[7:0]		Регистр сравнения частоты для определения увеличения частоты (EVENT2)
15...8	BASE_REG1[7:0]		Регистр сравнения частоты для определения снижения частоты (EVENT1)
7...0	BASE_REG0[7:0]		Регистр сравнения частоты для определения исчезновения частоты (EVENT0)

6.1.1.30 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[7:0]								MAX_CHK_SHIFT_REG0[7:0]							

Бит	Имя	Значение	Описание
31...21	-	-	Зарезервировано
20	PLL _n _READY	-	Флаг готовности генератора PLL _n 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...8	MAX_CHK_SHIFT_REG1[7:0]		Максимальное значение регистра SHIFT_REG1 с момента последнего сброса теневого регистра
7...0	MAX_CHK_SHIFT_REG0[7:0]		Максимальное значение регистра SHIFT_REG0 с момента последнего сброса теневого регистра

6.1.1.31 Регистр CAN_n_CLK

Base ADDR=		0x4000_0000		Offset=		0x0000_0118 0x0000_011C									
REG Name:		CAN _n _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]															

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

6.1.1.32 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]															

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 $TIMCLKn = MAX_CLK / (DIV + 1)$

6.1.1.33 Регистр MILn_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0148 0x0000_014C 0x0000_0150 0x0000_0154							
REG Name:		CHIPID													
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]															

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

Регистр не реализован в микроконтроллере.

6.1.1.34 Регистр SPWn_CLK

Base ADDR=		0x4000_0000				Offset=		0x0000_0178 0x0000_017C							
REG Name:		SPWn_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]															

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

6.1.1.35 Регистр UARTn_CLK

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

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

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 00000 – Генератор HSI 00001 – Генератор HSE0 00010 – Генератор HSE1 00011 – Генератор LSI 00100 – Генератор LSE 00101 – Генератор PLL0 00110 – Генератор PLL1 00111 – Генератор PLL2 01000 – Генератор PLL3 01001 – Генератор PLL4 01010 – Генератор PLL6 01011 – Генератор PLL7 01101 – MAX_CLK
27...17	-	-	Зарезервировано
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 $UARTCLKn = SELCLK / (DIV + 1)$

6.1.1.36 Регистр SSPn_CLK

Base ADDR=	0x4000_0000	Offset=	0x0000_01A0 0x0000_01A4 0x0000_01A8 0x0000_01AC 0x0000_01B0 0x0000_01B4												
REG Name:	CHIPID														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SELECT[3:0]															EN_CLK

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

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 00000 – Генератор HSI 00001 – Генератор HSE0 00010 – Генератор HSE1 00011 – Генератор LSI 00100 – Генератор LSE 00101 – Генератор PLL0 00110 – Генератор PLL1 00111 – Генератор PLL2 01000 – Генератор PLL3 01001 – Генератор PLL4 01010 – Генератор PLL6 01011 – Генератор PLL7 01101 – MAX_CLK
27...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для SSPCLKn 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты SSPCLKn SSPCLKn = SELCLK/(DIV+1)

6.1.1.37 Регистр RTC_CLK

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

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

Бит	Имя	Значение	Описание
31...28	SELECT[3:0]		Номер источника тактовой частоты для формирования частоты SELCLK 00000 – Генератор HSI 00001 – Генератор HSE0 00010 – Генератор HSE1 00011 – Генератор LSI 00100 – Генератор LSE 00101 – Генератор PLL0 00110 – Генератор PLL1 00111 – Генератор PLL2 01000 – Генератор PLL3 01001 – Генератор PLL4 01010 – Генератор PLL6 01011 – Генератор PLL7 01101 – MAX_CLK
27...17	-	-	Зарезервировано
16	EN_CLK		Разрешение формирования тактовой частоты для RTCCLKn 0 – нет частоты 1 – разрешена выдача частоты
15...0	DIV[15:0]		Делитель частоты SELCLK для формирования частоты RTCCLKn $RTCCLKn = SELCLK / (DIV + 1)$

6.2 Контроллер батарейного домена (BKPCNTR)

Блок батарейного домена предназначен для хранения основной конфигурации микроконтроллера, обеспечения функций часов реального времени и сохранения некоторого набора пользовательских данных при сбросах микроконтроллера кроме сброса по питанию.

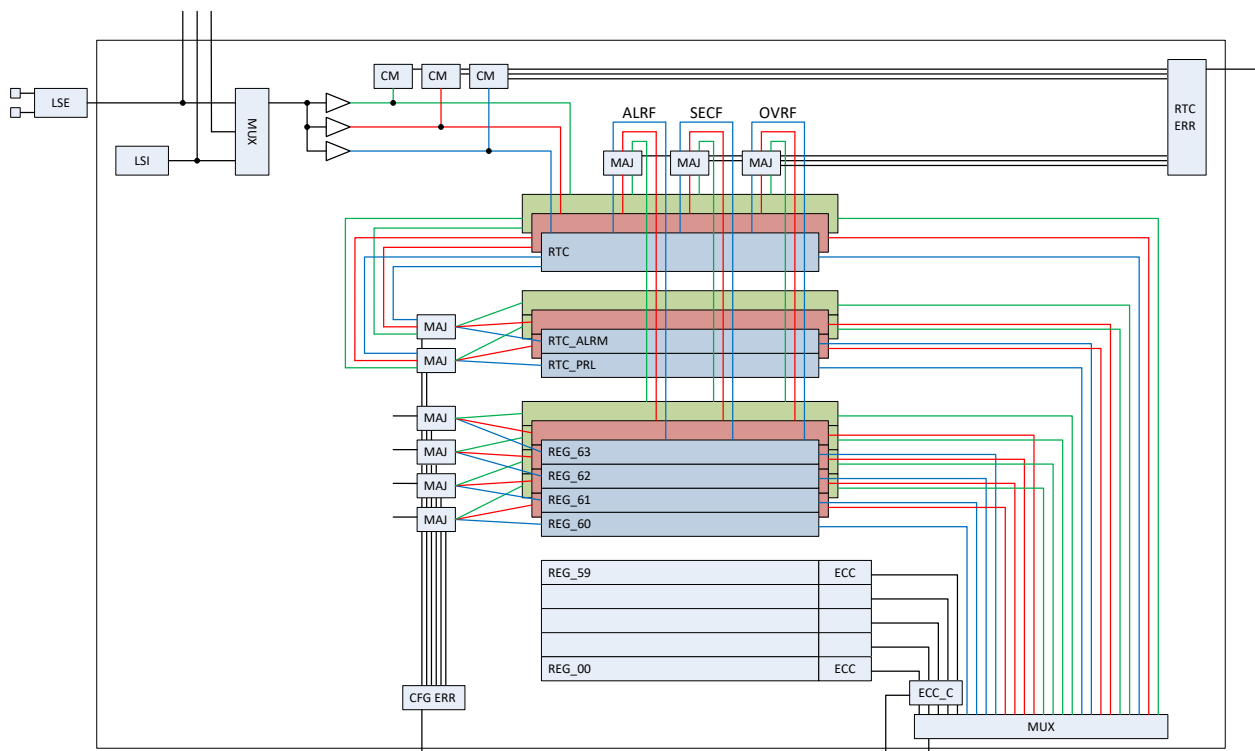


Рисунок 36 – Блок-схема контроллера батарейного домена

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, PORSTn_DSBL, действует не мажорированная логика (2 из 3), а логика 3 из 3. Т.е. для того, чтобы отключить сброс по недостаточному напряжению питания необходимо, чтобы бит PORSTn_DSBL был взведен во всех трех каналах. Если хотя бы в одном из регистров бит не будет взведен или ошибочно сброшен в ноль, то схема POR будет работать.

6.2.1 Часы реального времени

Часы реального времени позволяют организовать механизм отсчета времени в кристалле, в том числе при отключении основного источника питания. Включение часов реального времени осуществляется битом RTC_EN. В качестве источника тактовой частоты часов реального времени могут выступать генераторы LSI или LSE или сигнал RTC_CLK, который формируется в блоке управления тактовыми частотами и может быть выбран только при наличии питания DUcc. Выбор между источниками осуществляется

битами RTC_SEL. Биты управления осциллятором LSE расположены в батарейном домене.

Для калибровки тактовой частоты используются биты CAL[6:0]. Значение CAL определяет, какое число тактов из 2^{20} будет замаскировано. Таким образом, с помощью бит CAL[6:0] производится замедление хода часов. Изменение значения бит CAL может быть осуществлено в ходе работы часов реального времени.

Регистр RTC_DIV выступает в роли 20-ти битного предварительного делителя входной тактовой частоты, таким образом, чтобы на его выходе была тактовая частота в 1 Гц. Для задания коэффициента деления регистра RTC_DIV используется регистр RTC_PRL.

Регистр RTC_CNT предназначен для отсчета времени в секундах и работает на выходной частоте делителя RTC_DIV. Регистр RTC_ALR предназначен для задания времени, при совпадении с которым вырабатывается флаг прерывания и пробуждения процессора. Бит STANDBY, отключающий все внутренние регуляторы напряжения, автоматически сбрасывается при превышении RTC_CNT значения в регистре RTC_ALRM.

6.2.2 Регистры батарейного домена

Батарейный домен имеет 64 встроенных 32-разрядных регистра. Регистры с 60 по 64 служат для хранения битов управления батарейным доменом, микроконтроллером и часами реального времени, остальные 60 регистров могут быть использованы разработчиком программы.

6.2.3 Описание регистров

Таблица 37 – Описание регистров

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4000_1000	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

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
	0x0000_011C	REG_63_TMR1		Регистр батарейной памяти 63с 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 значения предварительного делителя часов реального времени
	0x0000_0138	RTC_PRL_TMR0		Основание счета предварительного делителя часов реального времени
	0x0000_013C	RTC_ALARM_TMR0		Регистр значения будильник часов реального времени
	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 значения предварительного делителя часов реального времени
	0x0000_0158	RTC_PRL_TMR1		Основание счета предварительного делителя часов реального времени
	0x0000_015C	RTC_ALARM_TMR1		Регистр значения будильник часов реального времени
	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 значения предварительного делителя часов реального времени
	0x0000_0178	RTC_PRL_TMR2		Основание счета предварительного делителя часов реального времени
	0x0000_017C	RTC_ALARM_TMR2		Регистр значения будильник часов реального времени

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
	0x0000_0180	RTC_CS_TMR2		Регистр управления и состояния часов реального времени

6.2.3.1 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]															

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

6.2.3.2 Регистр 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]															

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

6.2.3.3 Регистр 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	LOCKSTEP_EN	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]						

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 Запись 0 – нет эффекта Запись 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
29	OVRSTnDSBL	0	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 – она работает
28	PORSTnDSBL	0	Флаг запрещения сброса по недостаточному напряжению 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 – 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

Бит	Имя	Значение	Описание
24	LOCKSTEP_EN	1	Флаг разрешения работы в режиме LOCKSTEP 0 – режим LOCKSTEP отключен 1 – режим LOCKSTEP включен Взводится в 1 при любом сбросе. Режим LOCKSTEP выключается, если во всех трех каналах записан 0, если хотя бы в одном канале остается 1, выбран режим LOCKSTEP
23...16	DUCCLIM_EN	11111111	Флаг разрешения работы ограничения тока в доменах DU_{CC} DUCCLIM_EN[0] = 0 – нет ограничения по току в DU_{CC0} DUCCLIM_EN[0] = 1 – есть ограничения по току (>300 мА) в DU_{CC0} DUCCLIM_EN[1] = 0 – нет ограничения по току в DU_{CC1} DUCCLIM_EN[1] = 1 – есть ограничения по току (>300 мА) в DU_{CC1} ... DUCCLIM_EN[7] = 0 – нет ограничения по току в DU_{CC7} DUCCLIM_EN[7] = 1 – есть ограничения по току (>300 мА) в DU_{CC7} Сигнал формируется по правилам TMR
15	DISABLE_JTAG	0	Запрещение отладочного интерфейса JTAG 0 – JTAG_A и/или JTAG_B разрешены 1 – JTAG_A и/или JTAG_B запрещены
14	S_SYSRSTn	0	Флаг события сброса с вывода SYSRSTn 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	0	Флаг события сброса с вывода RESET1n 0 – не было сброса по RESET1n 1 – был сброс по RESET1n 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

Бит	Имя	Значение	Описание
9	S_RESET0n	0	Флаг события сброса с вывода RESET0n 0 – не было сброса по RESET0n 1 – был сброс по RESET0n Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
8	S_UccRESET	1	Флаг события сброса UccRESET 0 – не было сброса по UccRESET 1 – был сброс по UccRESET Флаг сбрасывается записью 1, если в момент очистки флага происходит очередное событие, флаг не будет очищен
7	FPOR	0	Флаг успешного завершения работы загрузочной бутовой программы 0 – бутовая программа не завершена 1 – бутовая программа завершена Возможна только запись 1. Сброс по любому сигналу сброса
6...0	MODE[6:0]	xxxxxx	Режим запуска микроконтроллера с выводов MODE. См. подраздел 5.5 «Режимы работы микроконтроллера». Можно переписать программно, начинает действовать со следующего запуска

6.2.3.4 Регистр 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_DUCC2	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]		

Бит	Имя	Значение	Описание
31	DUCC3_RDY		Аналогично SRILOW0, TRIM0, EN_DUCC0 и DUCC0_RDY для LDO3 регулятора
30	EN_DUCC3		
29:27	TRIM3[2:0]		
26:24	SRILOW3[2:0]		
23	DUCC2_RDY		Аналогично SRILOW0, TRIM0, EN_DUCC0 и DUCC0_RDY для LDO2 регулятора
22	EN_DUCC2		
21:19	TRIM2[2:0]		
18:16	SRILOW2[2:0]		
15	DUCC1_RDY		Аналогично SRILOW0, TRIM0, EN_DUCC0 и DUCC0_RDY для LDO1 регулятора
14	EN_DUCC1		
13:11	TRIM1[2:0]		
10:8	SRILOW1[2:0]		

Бит	Имя	Значение	Описание
7	DUCC0_RDY		Флаг готовности напряжения LDO0 регулятора 0 – регулятор LDO0 не готов 1 – регулятор LDO0 в норме
6	EN_DUCC0		Разрешение выдачи напряжения LDO0 регулятором 0 – разрешена работа LDO0 регулятора 1 – запрещена работа LDO0 регулятора Бит предустанавливается сигналами ALRF, UCCRESET Схема LDO выключается, если во всех трех каналах она заблокирована. Если хотя бы в одном канале схема LDO разрешена, она работает. Работа регулятора LDO0 всегда разрешена
5:3	TRIM0[2:0]		Подстройка напряжение DUcc в домене DUcc0 000 = +0,10 В 001 = +0,06 В 010 = +0,04 В 011 = +0,01 В 100 = - 0,01 В 101 = - 0,04 В 110 = - 0,06 В 111 = - 0,10 В Сигнал формируется по правилам TMR
2:0	SRILOW0[2:0]		Настройка режима работы домена DUcc0: 000 – Частота до 10 МГц (доп. ток ~300 мкА) 001 – Частота до 200 КГц (доп. ток ~7 мкА) 010 – Частота до 500 КГц (доп. ток ~20 мкА) 011 – Частота до 1 МГц (доп. ток ~80 мкА) 100 – При выключении всех генераторов (доп. ток ~2 мкА) 101 – Частота до 40 МГц (доп. ток ~900 мкА) 110 – Частота до 80 МГц (доп. ток ~5 мА) 111 – Частота более 80 МГц (доп. ток ~20 мА) Сигнал формируется по правилам TMR

6.2.3.5 Регистр 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]			-	-	REMAP	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	-						

Бит	Имя	Значение	Описание
31	-		TRIM_REF[2:0] – подстройка источника опорного напряжения LDO 100 – типовое значение
30:28	TRIM_REF[2]		
27,26	-		
25	REMAP		Переключение карты памяти контроллера 0 – BOOTROM расположен по адресу 0x00000000 1 – RAMC расположен по адресу 0x00000000
24	TRIMR[6]		TRIMR[7:0] – подстройка источника опорного тока LDO 10000000 – типовое значение
23			
22	TRIMR[7]		
21:19	TRIMR[5:3]		
18:16	TRIMR[2:0]		
15	DUCC5_RDY		Флаг готовности напряжения для домена DUCC_PLL 0 – нет напряжения DUCC в домене DUCC_PLL 1 – есть напряжение DUCC в домене DUCC_PLL
14	EN_DUCC5		Разрешение выдачи напряжения домен DUCC_PLL 0 – выдавать напряжение в домен DUCC_PLL 1 – не выдавать напряжение в домен DUCC_PLL Бит предустанавливается сигналами ALRF, UCCRESET Схема LDO выключается, если во всех трех каналах она заблокирована. Если хотя бы в одном канала она разрешена, она работает
13:11	TRIM5[2:0]		Подстройка напряжение DUCC в домене DUCC_PLL 000 = +0,10 В 001 = +0,06 В 010 = +0,04 В 011 = +0,01 В 100 = - 0,01 В 101 = - 0,04 В 110 = - 0,06 В 111 = - 0,10 В Сигнал формируется по правилам TMR
10:8	SRILOW5[2:0]		Настройка режима работы домена DUCC_PLL: 000 – Частота до 10 МГц (доп. ток ~300 мкА) 001 – Частота до 200 КГц (доп. ток ~7 мкА) 010 – Частота до 500 КГц (доп. ток ~20 мкА) 011 – Частота до 1 МГц (доп. ток ~80 мкА) 100 – При выключении всех генераторов (доп. ток ~2 мкА) 101 – Частота до 40 МГц (доп. ток ~900 мкА) 110 – Частота до 80 МГц (доп. ток ~5 мА) 111 – Частота более 80 МГц (доп. ток ~20 мА) Сигнал формируется по правилам TMR
7	DUCC4_RDY		Флаг готовности напряжения для домена DUCCB (не используется) 0 – нет напряжения DUCC в домене DUCCB 1 – есть напряжение DUCC в домене DUCCB Выключен может быть только сигналом SHDN
6	-		Зарезервирован Значение данного бита должно совпадать во всех регистрах
5...0	-		Зарезервирован

6.2.3.6 Регистр 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		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		LSE_RDY	CAL[7:0]								RTC_EN	RTC_SEL[1:0]		LSE_BYP	LSE_ON

Бит	Имя	Значение	Описание
31	RTC_RST		Сброс часов реального времени: 0 – часы не сбрасываются; 1 – часы сбрасываются Сигнал формируется по правилам TMR
30	-	0	Зарезервирован
29...24	HSI_TRIM[5:0]		Коэффициент подстройки частоты генератора HSI
23	HSI_RDY		Флаг выхода генератора HSI в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме
22	HSI_ON		Бит управления генератором HSI: 0 – генератор выключен; 1 – генератор включен Схемы HSI выключаются, если во всех трех каналах она выключена. Если хотя бы в одном канале она разрешена, работает
21	LSI_RDY		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]		Коэффициент подстройки частоты генератора LSI. Сигнал формируется по правилам TMR
15	LSI_ON		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, она работает
14	FILTER_EN		Фильтрация входных сигналов генератора LSE 0 – фильтр выключен 1 – фильтр включен
13	LSE_RDY		Флаг выхода генератора LSE в рабочий режим: 0 – генератор не запущен или не вышел в режим; 1 – генератор работает в рабочем режиме

Бит	Имя	Значение	Описание
12...5	CAL[7:0]		Коэффициент подстройки тактовой частоты часов реального времени, из каждых 2^{20} тактов будет замаскировано CAL тактов: 00000000 – 0 тактов 00000001 – 1 такт 11111111 – 256 тактов Таким образом, при частоте 32768.00000 Гц при CAL = 0 тактовая частота = 32768,00000 Гц при CAL = 1 тактовая частота = 32767,96875 Гц; ... при CAL = 256 тактовая частота = 32760,00000 Гц Сигнал формируется по правилам TMR
4	RTC_EN		Бит разрешения работы часов реального времени: 0 – работа запрещена; 1 – работа разрешена Сигнал формируется по правилам TMR
3...2	RTC_SEL[1:0]		Биты выбора источника тактовой синхронизации часов реального времени: 00 – LSI 01 – LSE 10 – RTCCLK (формируется в блоке CLK_CNTR) 11 – зарезервировано Сигнал формируется по правилам TMR
1	LSE_BYP		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
0	LSE_ON		Бит разрешения работы LSE генератора 0 – генератор выключен 1 – генератор включен Сигнал формируется по правилам TMR

6.2.3.7 Регистр 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]															

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

6.2.3.8 Регистр 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]															

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

6.2.3.9 Регистр 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]															

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19...0	RTC_PRL[19:0]		Значение основания для счетчика предварительного делителя часов реального времени

6.2.3.10 Регистр RTC_ALARM_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_ALARM[31:16]															

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

Бит	Имя	Значение	Описание
31...0	RTC_ALARM[31:0]		Значение, при превышении которого основным счетчиком часов реального времени будет выработан сигнал ALRF

6.2.3.11 Регистр 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_ERR	WEC	ALRF_IE	SECF_IE	OWF_IE	ALRF	SECF	OWF

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 LSI генератор.
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
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 сброс ошибок конфигурации RTC (кроме PRL и ALRM). Флаги не сбрасываются, если в момент их сброса событие сбоя продолжается. Для сброса флага ERR_CC необходима запись единиц во все CLR_ERRx

Бит	Имя	Значение	Описание
6	WEC		Запись завершена: 0 – можно записывать в регистры часов реального времени 1 – запись в регистры запрещена
5	ALRF_IE		Флаг разрешения прерывания RTC_IF по событию ALRF 0 – не генерировать прерывание по событию ALRF 1 – генерировать прерывание по событию ALRF
4	SECF_IE		Флаг разрешения прерывания RTC_IF по событию SECF 0 – не генерировать прерывание по событию 1 – генерировать прерывание по событию
3	OWF_IE		Флаг разрешения прерывания RTC_IF по событию OWF 0 – не генерировать прерывание по событию 1 – генерировать прерывание по событию
2	ALRF		Флаг возникновения события превышения основным счетчиком RTC_CNT значения в регистре RTC_ALRM 0 – нет события 1 – есть событие Сбрасывается записью 1
1	SECF		Флаг возникновения события равенства предварительного счетчика RTC_DIV и значения в регистре RTC_PRL 0 – нет события 1 – есть событие Сбрасывается записью 1
0	OWF		Флаг возникновения события переполнения основного счетчика RTC_CNT 0 – нет события 1 – есть событие Сбрасывается записью 1

6.3 Контроллер детектора напряжения питания (PWR_CNTR)

Блок детектора напряжений питания позволяет оценить напряжение внешних источников питания U_{сс} (питание IO, приемопередатчиков и т.п.), U_{сс1} (питание LDO), а также определить ток потребления доменов питания DU_{сс}. 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 ± 10 мА.

В микроконтроллере реализованы аппаратные сбросы при превышении уровня U_{сс} предельно-допустимых значений.

В блоке PWR_CNTR могут быть настроены условия для выработки событий превышения или снижения ниже других значений, при возникновении которых формируются прерывания или события для блока управления сбоями, ошибками и отказами (FT_CNTR), приводящими к сбросу микроконтроллера. Для каждого домена питания DU_{сс} реализована аппаратная защита по току потребления на уровне 300 мА.

6.3.1 Описание регистров

Т а б л и ц а 38 – Описание регистров блока детектора напряжений питания

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4000_2000	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

6.3.1.1 Регистр 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]															

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

6.3.1.2 Регистр 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[7:0]								PBVD_EN		PVD1_EN	PVD_EN

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10...3	PDID_EN[7:0]		Бит разрешения контроля тока потребления доменов питания Ducc PDID_EN[0] = 0 – отключен контроль для домена Ducc0 PDID_EN[0] = 1 – включен контроль для домена Ducc0 PDID_EN[7] = 0 – отключен контроль для домена Ducc7 PDID_EN[7] = 1 – включен контроль для домена Ducc7
2	PBVD_EN		Бит разрешения контроля уровня напряжения Висс

			0 – выключен 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 Uсс1
0	PVD_EN		Бит разрешения контроля уровня напряжения Uсс 0 – выключен 1 – включен

6.3.1.3 Регистр 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
														PVD1_IE	PVD_IE

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10...3	PDID_IE[7:0]		Бит разрешения прерывания по событию тока потребления доменов питания Ducc PDID_EN[0] = 0 – запрещено прерывание Ducc0 PDID_EN[0] = 1 – разрешено прерывание Ducc0 PDID_EN[7] = 0 – запрещено прерывание Ducc7 PDID_EN[7] = 1 – разрешено прерывание Ducc7
2	-	0	Зарезервирован
1	PVD1P_IE		Бит разрешения прерывания по событию уровня напряжения Uсс1 0 – запрещено прерывание 1 – разрешено прерывание
0	PVDP_IE		Бит разрешения прерывания по событию уровня напряжения Uсс 0 – запрещено прерывание 1 – разрешено прерывание

6.3.1.4 Регистр 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

[illegible]

Base ADDR=	0x1000_0000	Offset=	0x0000_0010						
------------	-------------	---------	-------------	--	--	--	--	--	--

[illegible]

--	--	--	--	--	--

Бит	Имя	Значение	Описание
			Сбрасывается записью 1

6.3.1.6 Регистр 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
-															

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

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...8	UCC1MAX[4:0]		Значение максимально допустимого напряжения U_{CC1} , при превышении или снижении которого происходит выработка флага PVDP1 и EVENT_PVD1 00000 – более 1,2В (ref) 00001 – более 1,4В 00010 – более 1,6В 00011 – более 1,8В ... 11111 – более 7,4В
7...0	UCCMAX[4:0]		Значение максимально допустимого напряжения U_{CC} , при превышении которого происходит выработка флага PVDP и EVENT_PVD

6.3.1.7 Регистр 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
								CURDUCC7[2:0]			CURDUCC6[2:0]			CURDUCC5[2:1]	

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

Бит	Имя	Значение	Описание
31...24	-		Зарезервировано
23...21	CURDUCC7[2:0]		Уровень тока для выработки события по току DUcc0 и формирования флага PDID[7] и события EVENT_CUR[7]
20...18	CURDUCC6[2:0]		Уровень тока для выработки события по току DUcc0 и формирования флага PDID[6] и события EVENT_CUR[6]
17...15	CURDUCC5[2:0]		Уровень тока для выработки события по току DUcc0 и формирования флага PDID[5] и события EVENT_CUR[5]
14...12	CURDUCC4[2:0]		Уровень тока для выработки события по току DUcc0 и формирования флага PDID[4] и события EVENT_CUR[4]
11...9	CURDUCC3[2:0]		Уровень тока для выработки события по току DUcc0 и формирования флага PDID[3] и события EVENT_CUR[3]
8...6	CURDUCC2[2:0]		Уровень тока для выработки события по току DUcc0 и формирования флага PDID[2] и события EVENT_CUR[2]
5...3	CURDUCC1[2:0]		Уровень тока для выработки события по току DUcc0 и формирования флага PDID[1] и события EVENT_CUR[1]
2...0	CURDUCC0[2:0]		Уровень тока для выработки события по току DUcc0 и формирования флага PDID[0] и события EVENT_CUR[0] 000 – ток более 5 мА 001 – ток более 15 мА 010 – ток более 25 мА 011 – ток более 35 мА 100 – ток более 45 мА 101 – ток более 55 мА 110 – ток более 65 мА 111 – ток более 75 мА

6.4 Контроллер Кэш-памяти (CACHCNTR)

6.4.1 Описание регистров

Таблица 39 – Описание регистров контроллера Кэш-памяти

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4000C000				Базовый адрес контроллера кэша шины S-Bus
	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_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

6.4.1.1 Регистр KEY

Base ADDR=	0x4000C000	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]															

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

6.4.1.2 Регистр CNTR

Base ADDR=		0x4000C000					Offset=		0x0000_0004						
REG Name:		s													
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_MIS	CLR_HIT	EN_CACHE_DATA	EN_CACHE_CODE	CLR_CACHE
															EN_CACHE

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_DATA		Бит разрешения кэширования данных программ Имеет значение только при EN_CACHE=1 0 – данные не кэшируются 1 – данные кэшируются
2	EN_CACHE_CODE		Бит разрешения кэширования кода программ Имеет значение только при EN_CACHE=1 0 – код не кэшируется 1 – код кэшируется
1	CLR_CACHE		Бит сброса кэша Сброс выполняется записью 1
0	EN_CACHE		Бит общего разрешения работы кэша 0 – выключен 1 – включен

6.4.1.3 Регистр HIT_CNT

Base ADDR=		0x4000C000					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]															

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

6.4.1.4 Регистр MISS_CNT

Base ADDR=	0x4000C000	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]															
Бит	Имя	Значение	Описание												
31...0	CNT[31:0]		Счетчик числа промахов в кэш												

6.4.1.5 Регистр ECCCS

Base ADDR=	0x4000C000	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

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 одиночных ошибок Запись 1 сбрасывает счетчик SECC_CNT Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных 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 – разрешено 1 – запрещено

Бит	Имя	Значение	Описание
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке 0 – разрешено 1 – запрещено
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

6.4.1.6 Регистр ECCADR

Base ADDR=	0x4000C000	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]															

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

6.4.1.7 Регистр ECCDATA

Base ADDR=	0x4000C000	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]															

Бит	Имя	Значение	Описание
-----	-----	----------	----------

31...0	ECCDATA[31:0]		Считанные данные при последней двойной или одинарной ошибке без корректировки ECC
--------	---------------	--	---

6.4.1.8 Регистр ECCECC

Base ADDR=		0x4000C000		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]							

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]		Считанные ECC биты при последней двойной или одинарной ошибке без корректировки ECC

6.5 Контроллер обработки событий отказов, сбоев и ошибок (FT_CNTR)

Таблица 40 – Интерфейсы FT_CNTR

Обозначение входа	Направление	Логич. уровень	Описание
APB	IO		Шина APB
PCLK	I		Тактовый сигнал APB
PRESETn	I		Сброс APB 0 – сброс 1 – работа
HSI	I		Тактовый сигнал HIS
EVENT0[31:0]	I		Сигналы запросов событий сбоев 0 – нет запроса 1 – есть запрос
...			
EVENT12[31:0]	I		Сигналы запросов событий сбоев 0 – нет запроса 1 – есть запрос
CANADR[5:0]	I		Адрес для выбора регистров FT_CNTR 0 – KEY 1 – CONTROL ... 34 – FT_STAT0 35 – FT_STAT1
CANSTAT[1:0]	I		Статус передатчика CAN (отображен в STATUS)
CONTROL[31:0]	O		Отображение регистра CONTROL
FT_RESET	O		Выход сигнала сброса После того как произошло событие сбоя, для которого разрешен отложенный сброс, и счетчик досчитал до конца, вырабатывается сигнал сброса. 1 – сброс 0 – нет сброса
FT_IF0	O		Сигнал запроса прерывания по всем событиям с высокой важностью, для которых разрешен отложенный сброс
FT_IF1	O		Сигнал запроса прерывания по всем событиям с высокой важностью
FT_IF2	O		Сигнал запроса прерывания для событий средней важности, для которых он разрешен
FT_IF3	O		Сигнал запроса прерывания для событий низкой важности, для которых он разрешен
FT_REG_SE	O		Сигнал обнаружения одиночной ошибки в регистрах FT_CNTR
FT_REG_DE	O		Сигнал обнаружения двойной ошибки в регистрах FT_CNTR
FT_CNT_ERR	O		Обнаружение ошибки мажорирования в регистре счетчика отложенного сброса

6.5.1 Описание регистров

Таблица 41 – Описание регистров контроллера FT_CNTR

Базовый адрес	Название	Описание
0x4000_3000	FT_CNTR	Блок обработки сбоев, ошибок и отказов

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 HSI генератора
0x0000_0014	FIRSTEVENT	Значение TICKCOUNT первого зафиксированного события
0x0000_0018	LASTEVENT	Значения TICKCOUNT последнего зафиксированного события
0x0000_001C	TIMEOUTCNT	Текущее значение счетчика отложенного сброса
0x0000_0020	EVENT0	Флаги событий сбоя
0x0000_0024	EVENT1	Флаги событий сбоя
0x0000_0028	EVENT2	Флаги событий сбоя
0x0000_002C	EVENT3	Флаги событий сбоя
0x0000_0030	EVENT4	Флаги событий сбоя
0x0000_0034	EVENT5	Флаги событий сбоя
0x0000_0038	EVENT6	Флаги событий сбоя
0x0000_003C	EVENT7	Флаги событий сбоя
0x0000_0040	EVENT8	Флаги событий сбоя
0x0000_0044	EVENT9	Флаги событий сбоя
0x0000_0048	EVENT10	Флаги событий сбоя
0x0000_004C	EVENT11	Флаги событий сбоя
0x0000_0050	EVENT12	Флаги событий сбоя
0x0000_0054	RESET_EVENT0	Биты разрешения отложенного сброса от событий сбоя
0x0000_0058	RESET_EVENT1	Биты разрешения отложенного сброса от событий сбоя
0x0000_005C	RESET_EVENT2	
0x0000_0060	RESET_EVENT3	Биты разрешения отложенного сброса от событий сбоя
0x0000_0064	RESET_EVENT3	Биты разрешения отложенного сброса от событий сбоя
0x0000_0068	IE_EVENT5	Биты разрешения прерывания от событий сбоя
0x0000_006C	IE_EVENT6	Биты разрешения прерывания от событий сбоя
0x0000_0070	IE_EVENT7	Биты разрешения прерывания от событий сбоя
0x0000_0074	IE_EVENT8	Биты разрешения прерывания от событий сбоя
0x0000_0078	IE_EVENT9	Биты разрешения прерывания от событий сбоя
0x0000_007C	IE_EVENT10	Биты разрешения прерывания от событий сбоя
0x0000_0080	IE_EVENT11	Биты разрешения прерывания от событий сбоя
0x0000_0084	IE_EVENT12	Биты разрешения прерывания от событий сбоя

6.5.1.1 Регистр 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]															

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

6.5.1.2 Регистр 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
												CAN_RESET	CANDATA_VALID	RESET_TIMEOUT_CNT	TICKOVER_IE
															CAN_EN

Бит	Имя	Значение	Описание
	CANID[7:0]	8'h00	ID код отправляемых пакетов CAN После сброса CANID = 0 Контроллер FTCAN отправляет пакеты с ID={b0,CANID, НОМЕР РЕГИСТРА } Где НОМЕР РЕГИСТРА KEY – 0 CONTROL – 1 ... FT_STAT0 – 34 FT_STAT1 – 35
4	CAN_RESET	0	Бит программного сброса блока FT_CAN в начальное состояние Запись 0 – ничего Запись 1 – сброс
2	RESET_TIMEOUT_CNT	0	Бит сброса регистра TIMEOUTCNT в состояние ожидания Запись 0 – ничего Запись 1 – сброс, при условии отсутствия каких-либо флагов события, для которых разрешен вызов отложенного сброса
1	TICKOVER_IE	0	Бит разрешения прерывания FT_IE2 при возникновении переполнения счетчика TICKCOUNT Счетчик TICKCOUNT переполняется раз в ~9 минут
0	CAN_EN	1	Бит разрешения работы встроенного контроллера CAN по отправке статуса состояния микроконтроллера 0 – CAN запрещен 1 – CAN разрешен (после сброса)

[illegible]

Бит	Имя	Значение	Описание
31...10	–	–	Зарезервировано
9, 8	CAN_STAT[1:0]		Состояние CAN интерфейса 00 – активный 01 – пассивная ошибка 10 и 11 – выключенная шина
7			
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	LAST_EVENT	0	Флаг возникновения последнего события сбоя 0 – нет события или был сброшен 1 – есть событие Флаг сбрасывается записью 1, если в момент сброса происходит очередное событие, флаг не сбрасывается.
1	FIRST_EVENT	0	Флаг возникновения первого события сбоя 0 – нет события 1 – есть событие Данный флаг сбрасывается только сигналом сброса
0	TICKOVER	0	Флаг переполнения счетчика TICKCNT 0 – нет переполнения 1 – было переполнение Флаг сбрасывается записью 1, если в момент сброса происходит очередное переполнение, флаг не сбрасывается.

6.5.1.4 Регистр TIMEOUT

Base ADDR=		0x4000_3000				Offset=		0x0000_000C							
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

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

6.5.1.5 Регистр 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]															

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

6.5.1.6 Регистр 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]															

Бит	Имя	Значение	Описание
31...0	FIRSTEVENT[31:0]		Значение счетчика TICKCNT на момент возникновения первого события сбоя. Сбрасывается только сигналом сброса

6.5.1.7 Регистр 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]															

Бит	Имя	Значение	Описание
31...0	LASTEVENT[31:0]		Значение счетчика TICKCNT на момент возникновения последнего события сбоя. Сбрасывается только сигналом сброса

6.5.1.8 Регистр 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
RESERVED															TIMEOUTCNT [16]

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

Бит	Имя	Значение	Описание
31...17	–	–	Зарезервировано
16...0	TIMEOUTCNT	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_RESET.

Счетчик TIMEOUTCNT имеет 17 разрядов, при сбросе устанавливается в состояние всех единиц. При возникновении события устанавливается в состояние 16-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 отложенный сброс;
- через запись 1 в бит RESET_TIMEOUTCNT установить счетчик в исходное состояние.

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

6.5.1.9 Регистр EVENT0 (HIGH)

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_	AHB_EXTBUS_	SRAMC_COMP_	AHB_SRAMC_	SRAMD_COMP_	AHB_SRAMD_	OTP_COMP_REG_	AHB_OTP_COMP_	AHB_C_	AHB_M_	AHB_S_	AHB_D_	AHB_I_	COMP_		
		REG_ERR	COMP_ERR	REG_ERR	COMP_ERR	REG_ERR	COMP_ERR	ERR	ERR	COMP_ERR	COMP_ERR	COMP_ERR	COMP_ERR	COMP_ERR	REG_ERR		

Бит	Имя	Значение	Описание
31...14	–	–	Зарезервировано
13	EXTBUS_COMP_REG_ERR ¹	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера EXTBUS 0 – нет ошибки 1 – есть ошибка
12	AHB_EXTBUS_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера EXTBUS 0 – нет ошибки 1 – есть ошибка
11	SRAMC_COMP_REG_ERR ¹	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера SRAMC 0 – нет ошибки 1 – есть ошибка
10	AHB_SRAMC_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера SRAMC 0 – нет ошибки 1 – есть ошибка
9	SRAMD_COMP_REG_ERR ¹	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера SRAMD 0 – нет ошибки 1 – есть ошибка
8	AHB_SRAMD_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера SRAMD 0 – нет ошибки 1 – есть ошибка
7	OTP_COMP_REG_ERR ¹	0	Флаг ошибки сравнения сигналов LOCKSTEP контроллера OTP 0 – нет ошибки 1 – есть ошибка
6	AHB_OTP_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов шины AHBLOCKSTEP контроллера OTP 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
5	AHB_C_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов системной шины SCRUBBER процессора CPU 0 и процессора CPU 1 0 – нет ошибки 1 – есть ошибка
4	AHB_M_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов системной шины DMA процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка
3	AHB_S_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов системной шины S процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка
2	AHB_D_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов системной шины D процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка
1	AHB_I_COMP_ERR ¹	0	Флаг ошибки сравнения сигналов системной шины I процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка
0	COMP_REG_ERR ¹	0	Ошибка сравнения вспомогательных сигналов процессорного ядра CPU 0 и процессорного ядра CPU 1 0 – нет ошибки 1 – есть ошибка

6.5.1.10 Регистр 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_EXTBUS1	ECCERR_EXTBUS0	ECCERR_RAMC1	ECCERR_RAMC0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCERR_RAMD1	ECCERR_RAMD0	ECCERR_OTP1	ECCERR_OTP0	.	ECCERR_ROM	ECCERRC_1	ECCERRM_1	ECCERRS_1	ECCERRD_1	ECCERRI_1	ECCERRC_0	ECCERRM_0	ECCERRS_0	ECCERRD_0	ECCERRI_0

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19	ECCERR_EXTBUS1 ¹	0	Ошибка ЕСС на шине АНВ на EXT_BUS1 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
18	ECCERR_EXTBUS0 ¹	0	Ошибка ECC на шине АHB на EXT_BUS0 0 – нет ошибки 1 – есть ошибка
17	ECCERR_RAMC1 ¹	0	Ошибка ECC на шине АHB на RAMC1 0 – нет ошибки 1 – есть ошибка
16	ECCERR_RAMC0 ¹	0	Ошибка ECC на шине АHB на RAMC0 0 – нет ошибки 1 – есть ошибка
15	ECCERR_RAMD1 ¹	0	Ошибка ECC на шине АHB на RAMD1 0 – нет ошибки 1 – есть ошибка
14	ECCERR_RAMD0 ¹	0	Ошибка ECC на шине АHB на RAMD0 0 – нет ошибки 1 – есть ошибка
13	ECCERR_OTP1 ¹	0	Ошибка ECC на шине АHB на OTP1 0 – нет ошибки 1 – есть ошибка
12	ECCERR_OTP0 ¹	0	Ошибка ECC на шине АHB на OTP0 0 – нет ошибки 1 – есть ошибка.
11	-	-	Зарезервировано
10	ECCERR_ROM ¹	0	Ошибка ECC на шине АHB на ROM 0 – нет ошибки 1 – есть ошибка
9	ECCERRC_1 ¹	0	Ошибка ECC на шине С на SCRUBBER1 0 – нет ошибки 1 – есть ошибка
8	ECCERRM_1 ¹	0	Ошибка ECC на шине М на DMA 1 0 – нет ошибки 1 – есть ошибка
7	ECCERRS_1 ¹	0	Ошибка ECC на шине S на CPU 1 0 – нет ошибки 1 – есть ошибка
6	ECCERRD_1 ¹	0	Ошибка ECC на шине D на CPU 1 0 – нет ошибки 1 – есть ошибка
5	ECCERRI_1 ¹	0	Ошибка ECC на шине I на CPU 1 0 – нет ошибки 1 – есть ошибка
4	ECCERRC_0 ¹	0	Ошибка ECC на шине С на SCRUBBER 0 0 – нет ошибки 1 – есть ошибка
3	ECCERRM_0 ¹	0	Ошибка ECC на шине М на DMA 0 0 – нет ошибки 1 – есть ошибка
2	ECCERRS_0 ¹	0	Ошибка ECC на шине S на CPU 0 0 – нет ошибки 1 – есть ошибка
1	ECCERRD_0 ¹	0	Ошибка ECC на шине D на CPU 0 0 – нет ошибки 1 – есть ошибка
0	ECCERRI_0 ¹	0	Ошибка ECC на шине I на CPU 0 0 – нет ошибки 1 – есть ошибка

6.5.1.11 Регистр 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_RAMC_DE	ECC_CACHE_D_1_DE	ECC_CACHE_D_0_DE	ECC_CACHE_I_1_DE	ECC_CACHE_I_0_DE	ECC_EXTBUS_DE	ECC_RAMC_DE	ECC_RAMD_DE	ECC_OTP_DE	ECC_ROM_DE	

Бит	Имя	Значение	Описание
31...9	-		Зарезервировано
10	ECC_CACHE_RAMD_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE в RAMD 0 – нет ошибки 1 – есть ошибка
9	ECC_CACHE_RAMC_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE в RAMC 0 – нет ошибки 1 – есть ошибка
8	ECC_CACHE_D_1_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_D в CPU1 0 – нет ошибки 1 – есть ошибка
7	ECC_CACHE_D_0_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_D в CPU0 0 – нет ошибки 1 – есть ошибка
6	ECC_CACHE_I_1_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_I в CPU1 0 – нет ошибки 1 – есть ошибка
5	ECC_CACHE_I_0_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти CACHE_I в CPU0 0 – нет ошибки 1 – есть ошибка
4	ECC_EXTBUS_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти EXT_BUS 0 – нет ошибки 1 – есть ошибка
3	ECC_RAMC_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти RAMC 0 – нет ошибки 1 – есть ошибка
2	ECC_RAMD_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти RAMD 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
1	ECC_OTP_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти OTP 0 – нет ошибки 1 – есть ошибка
0	ECC_ROM_DE ¹	0	Двойная неисправимая ошибка ECC в массиве памяти ROM 0 – нет ошибки 1 – есть ошибка

6.5.1.12 Регистр 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_MONITOR7	CLK_MONITOR6	CLK_MONITOR5	CLK_MONITOR4	CLK_MONITOR3	CLK_MONITOR2	CLK_MONITOR1	CLK_MONITOR0

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7	CLK_MONITOR7	0	Резерв
6	CLK_MONITOR6	0	Резерв
5	CLK_MONITOR5	0	Резерв
4	CLK_MONITOR4 ¹	0	Ошибка превышения максимальной тактовой частоты RAMC 0 – нет ошибки 1 – есть ошибка
3	CLK_MONITOR3 ¹	0	Ошибка превышения максимальной тактовой частоты RAMD 0 – нет ошибки 1 – есть ошибка.
2	CLK_MONITOR2 ¹	0	Ошибка превышения максимальной тактовой частоты OTP 0 – нет ошибки 1 – есть ошибка
1	CLK_MONITOR1 ¹	0	Ошибка превышения максимальной тактовой частоты CPU1 0 – нет ошибки 1 – есть ошибка
0	CLK_MONITOR0 ¹	0	Ошибка превышения максимальной тактовой частоты CPU0 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

6.5.1.13 Регистр 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	RAMC_REG_DE	RAMD_REG_DE	OTP_REG_DE	FT_REG_DE	FTCNTR_ERR	BKP_MEM_DE	BKP_ERR	RTC_ERR

Бит	Имя	Значение	Описание
31...9			
8	ROM_REG_DE ¹		Двойная не исправимая ошибка в регистрах управления ROM_CNTR 0 – нет ошибки 1 – есть ошибка
7	RAMC_REG_DE ¹		Двойная не исправимая ошибка в регистрах управления RAMC_CNTR 0 – нет ошибки 1 – есть ошибка
6	RAMD_REG_DE ¹		Двойная не исправимая ошибка в регистрах управления RAMD_CNTR 0 – нет ошибки 1 – есть ошибка
5	OTP_REG_DE ¹	0	Двойная не исправимая ошибка в регистрах управления OTP_CNTR 0 – нет ошибки 1 – есть ошибка
4	FT_REG_DE ¹	0	Двойная не исправимая ошибка в регистрах управления FT_CNTR 0 – нет ошибки 1 – есть ошибка
3	FTCNTR_ERR ¹	0	Ошибка в счетчике TIMEOUTCNT отложенного сброса FT_CNTR 0 – нет ошибки 1 – есть ошибка
2	BKP_MEM_DE ¹	0	Двойная ошибка в массиве памяти ВКР (00-59) 0 – нет ошибки 1 – есть ошибка
1	BKP_ERR ¹	0	Ошибка в часах конфигурационных регистрах ВКР (60-63) 0 – нет ошибки 1 – есть ошибка
0	RTC_ERR ¹	0	Ошибка в часах реального времени RTC 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

6.5.1.14 Регистр EVENT5 (MIDDLE)

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	SPHY_CLK_ DE	SPW_CLK_ DE	EPHY_CLK_ DE	EMAC_CLK_ DE	ARC_CLK_ DE	MIL_CLK_D DE	TIM_CLK_D DE	CAN_CLK_ DE	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL7_CLK_ DE	PLL6_CLK_ DE	PLL5_CLK_ DE	PLL4_CLK_ DE	PLL3_CLK_ DE	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

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28	RTC_CLK_DE ¹		Двойная не исправимая ошибка в регистре RTC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
27	DAC_CLK_DE ¹		Двойная не исправимая ошибка в регистре DAC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
26	ADC_CLK_DE ¹		Двойная не исправимая ошибка в регистре ADC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
25	SSP_CLK_DE ¹		Двойная не исправимая ошибка в регистре SSP_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
24	UART_CLK_DE ¹		Двойная не исправимая ошибка в регистре UART_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
23	SPHY_CLK_DE ¹		Двойная не исправимая ошибка в регистре SPHY_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
22	SPW_CLK_DE ¹		Двойная не исправимая ошибка в регистре SPW_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
21	EPHY_CLK_DE ¹		Двойная не исправимая ошибка в регистре EPHY_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
20	EMAC_CLK_DE ¹		Двойная не исправимая ошибка в регистре EMAC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
19	ARC_CLK_DE ¹		Двойная не исправимая ошибка в регистре ARC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
18	MIL_CLK_DE ¹		Двойная не исправимая ошибка в регистре MIL_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
17	TIM_CLK_DE ¹		Двойная не исправимая ошибка в регистре TIM_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
16	CAN_CLK_DE ¹		Двойная не исправимая ошибка в регистре CAN_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
15	PLL7_CLK_DE ¹		Двойная не исправимая ошибка в регистре PLL7_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
14	PLL6_CLK_DE ¹		Двойная не исправимая ошибка в регистре PLL6_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
13	PLL5_CLK_DE ¹		Двойная не исправимая ошибка в регистре PLL5_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
12	PLL4_CLK_DE ¹		Двойная не исправимая ошибка в регистре PLL4_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
11	PLL3_CLK_DE ¹		Двойная не исправимая ошибка в регистре PLL3_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
10	PLL2_CLK_DE ¹		Двойная не исправимая ошибка в регистре PLL2_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
9	PLL1_CLK_DE ¹		Двойная не исправимая ошибка в регистре PLL1_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
8	PLL0_CLK_DE ¹		Двойная не исправимая ошибка в регистре PLL0_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
7	HSE1_CLK_DE ¹		Двойная не исправимая ошибка в регистре HSE1_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
6	HSE0_CLK_DE ¹		Двойная не исправимая ошибка в регистре HSE0_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка

Бит	Имя	Значение	Описание
5	LSE_CLK_DE ¹		Двойная не исправимая ошибка в регистре LSE_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
4	LSI_CLK_DE ¹		Двойная не исправимая ошибка в регистре LSI_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
3	PER_CLK_DE ¹		Двойная не исправимая ошибка в регистре PER_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
2	CPU_CLK_DE ¹		Двойная не исправимая ошибка в регистре CPU_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
1	MAX_CLK_DE ¹		Двойная не исправимая ошибка в регистре MAX_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка.
0	KEY_DE ¹		Двойная не исправимая ошибка в регистре KEY блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка

6.5.1.15 Регистр EVENT6 (MIDLE)

[illegible]

Бит	Имя	Значение	Описание
31...13	-		Зарезервировано
12	PLL7_CLOCK_ERROR ¹		Ошибка при формировании частоты PLL7_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
11	PLL6_CLOCK_ERROR ¹		Ошибка при формировании частоты PLL6_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
10	PLL5_CLOCK_ERROR ¹		Ошибка при формировании частоты PLL5_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
9	PLL4_CLOCK_ERROR ¹		Ошибка при формировании частоты PLL4_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
8	PLL3_CLOCK_ERROR ¹		Ошибка при формировании частоты PLL3_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
7	PLL2_CLOCK_ERROR ¹		Ошибка при формировании частоты PLL2_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
6	PLL1_CLOCK_ERROR ¹		Ошибка при формировании частоты PLL1_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
5	PLL0_CLOCK_ERROR ¹		Ошибка при формировании частоты PLL0_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
4	LSE_CLOCK_ERROR ¹		Ошибка при формировании частоты LSE_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
3	LSI_CLOCK_ERROR ¹		Ошибка при формировании частоты LSI_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
2	HSE1_CLOCK_ERROR ¹		Ошибка при формировании частоты HSE1_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
1	HSE0_CLOCK_ERROR ¹		Ошибка при формировании частоты HSE0_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
0	CPU_CLOCK_ERROR ¹		Ошибка при формировании частоты CPU_CLK в блоке CLK_CHECKER в блоке CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

6.5.1.16 Регистр EVENT7 (MIDDLE)

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		
					OTP_REG_SE	FT_REG_SE	BKP_MEM_SE	SRAMC_SE	SRAMD_SE	ECC_CACHE_D_1_SE	ECC_CACHE_D_0_SE	ECC_CACHE_I_1_SE	ECC_CACHE_I_0_SE	ROM_SE	OTP_SE		

Бит	Имя	Значение	Описание
31...10	–	–	Зарезервировано
15	RAMD_CACHE_SE		
14	RAMC_CACHE_SE		
13	ROM_REG_SE		
12	RAMD_REG_SE		
11	RAMC_REG_SE		
10	OTP_REG_SE ¹	0	Одиночная ошибка в регистрах управления блока OTP_CNTR 0 – нет ошибки 1 – есть ошибка
9	FT_REG_SE ¹	0	Одиночная ошибка в регистрах управления блока FT_CNTR 0 – нет ошибки 1 – есть ошибка
8	BKP_MEM_SE ¹	0	Одиночная ошибка в массиве памяти BKP (00-59) 0 – нет ошибки 1 – есть ошибка
7	SRAMC_SE ¹	0	Одиночная ошибка в массиве памяти RAMC 0 – нет ошибки 1 – есть ошибка
6	SRAMD_SE ¹	0	Одиночная ошибка в массиве памяти RAMD 0 – нет ошибки 1 – есть ошибка
5	ECC_CACHE_D_1_SE ¹	0	Одиночная ошибка в массиве памяти CACHEDCPU1 0 – нет ошибки 1 – есть ошибка
4	ECC_CACHE_D_0_SE ¹	0	Одиночная ошибка в массиве памяти CACHEDCPU0 0 – нет ошибки 1 – есть ошибка
3	ECC_CACHE_I_1_SE ¹	0	Одиночная ошибка в массиве памяти CACHEICPU1 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
2	ECC_CACHE_I_0_SE ¹	0	Одинокная ошибка в массиве памяти CACHEICPU0 0 – нет ошибки 1 – есть ошибка
1	ROM_SE ¹	0	Одинокная ошибка в массиве памяти ROM 0 – нет ошибки 1 – есть ошибка
0	OTP_SE ¹	0	Одинокная ошибка в массиве памяти OTP 0 – нет ошибки 1 – есть ошибка

6.5.1.17 Регистр EVENT8 (MIDLE)

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
					LDO7_CUR_EVENT	LDO6_CUR_EVENT	LDO5_CUR_EVENT	LDO4_CUR_EVENT	LDO3_CUR_EVENT	LDO2_CUR_EVENT	LDO1_CUR_EVENT	LDO0_CUR_EVENT	-	PVD_UCC1_EVENT	PVD_UCC_EVENT

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10	LDO7_CUR_EVENT ¹	0	Ошибка по току потребления LDO7 от блока PVD 0 – нет ошибки 1 – есть ошибка
9	LDO6_CUR_EVENT ¹	0	Ошибка по току потребления LDO6 от блока PVD 0 – нет ошибки 1 – есть ошибка
8	LDO5_CUR_EVENT ¹	0	Ошибка по току потребления LDO5 от блока PVD 0 – нет ошибки 1 – есть ошибка
7	LDO4_CUR_EVENT ¹	0	Ошибка по току потребления LDO4 от блока PVD 0 – нет ошибки 1 – есть ошибка
6	LDO3_CUR_EVENT ¹	0	Ошибка по току потребления LDO3 от блока PVD 0 – нет ошибки 1 – есть ошибка
5	LDO2_CUR_EVENT ¹	0	Ошибка по току потребления LDO2 от блока PVD 0 – нет ошибки 1 – есть ошибка
4	LDO1_CUR_EVENT ¹	0	Ошибка по току потребления LDO1 от блока PVD 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
3	LDO0_CUR_EVENT ¹	0	Ошибка по току потребления LDO0 от блока PVD 0 – нет ошибки 1 – есть ошибка
2	-	0	Зарезервирован
1	PVD_UCC1_EVENT ¹	0	Ошибка по питанию Ucc1 от блока PVD 0 – нет ошибки 1 – есть ошибка
0	PVD_UCC_EVENT ¹	0	Ошибка по питанию Ucc от блока PVD 0 – нет ошибки 1 – есть ошибка

6.5.1.18 Регистр 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	SPHY_CLK_SE	SPW_CLK_SE	EPHY_CLK_SE	EMAC_CLK_SE	ARC_CLK_SE	MIL_CLK_SE	TIM_CLK_SE	CAN_CLK_SE		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
PLL7_CLK_SE	PLL6_CLK_SE	PLL5_CLK_SE	PLL4_CLK_SE	PLL3_CLK_SE	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		

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28	RTC_CLK_SE ¹		Одиночная исправимая ошибка в регистре RTC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
27	DAC_CLK_SE ¹		Одиночная исправимая ошибка в регистре DAC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
26	ADC_CLK_SE ¹		Одиночная исправимая ошибка в регистре ADC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
25	SSP_CLK_SE ¹		Одиночная исправимая ошибка в регистре SSP_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
24	UART_CLK_SE ¹		Одиночная исправимая ошибка в регистре UART_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
23	SPHY_CLK_SE ¹		Одиночная исправимая ошибка в регистре SPHY_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
22	SPW_CLK_SE ¹		Одиночная исправимая ошибка в регистре SPW_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
21	EPHY_CLK_SE ¹		Одиночная исправимая ошибка в регистре EPHY_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка.
20	EMAC_CLK_SE ¹		Одиночная исправимая ошибка в регистре EMAC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
19	ARC_CLK_SE ¹		Одиночная исправимая ошибка в регистре ARC_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
18	MIL_CLK_SE ¹		Одиночная исправимая ошибка в регистре MIL_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
17	TIM_CLK_SE ¹		Одиночная исправимая ошибка в регистре TIM_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
16	CAN_CLK_SE ¹		Одиночная исправимая ошибка в регистре CAN_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
15	PLL7_CLK_SE ¹		Одиночная исправимая ошибка в регистре PLL7_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
14	PLL6_CLK_SE ¹		Одиночная исправимая ошибка в регистре PLL6_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка.
13	PLL5_CLK_SE ¹		Одиночная исправимая ошибка в регистре PLL5_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
12	PLL4_CLK_SE ¹		Одиночная исправимая ошибка в регистре PLL4_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
11	PLL3_CLK_SE ¹		Одиночная исправимая ошибка в регистре PLL3_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
10	PLL2_CLK_SE ¹		Одиночная исправимая ошибка в регистре PLL2_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка

Бит	Имя	Значение	Описание
9	PLL1_CLK_SE ¹		Одиночная исправимая ошибка в регистре PLL1_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
8	PLL0_CLK_SE ¹		Одиночная исправимая ошибка в регистре PLL0_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
7	HSE1_CLK_SE ¹		Одиночная исправимая ошибка в регистре HSE1_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
6	HSE0_CLK_SE ¹		Одиночная исправимая ошибка в регистре HSE0_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка.
5	LSE_CLK_SE ¹		Одиночная исправимая ошибка в регистре LSE_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
4	LSI_CLK_SE ¹		Одиночная исправимая ошибка в регистре LSI_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
3	PER_CLK_SE ¹		Одиночная исправимая ошибка в регистре PER_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
2	CPU_CLK_SE ¹		Одиночная исправимая ошибка в регистре CPU_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
1	MAX_CLK_SE ¹		Одиночная исправимая ошибка в регистре MAX_CLK блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка
0	KEY_SE ¹		Одиночная исправимая ошибка в регистре KEY блока CLOCK_CNTR 0 – нет ошибки 1 – есть ошибка

6.5.1.19 Регистр 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

Бит	Имя	Значение	Описание
31...6	-		Зарезервировано
5	PORTF_CURLIM ¹		Ошибка превышения выходного тока в PORTF 0 – нет ошибки 1 – есть ошибка
4	PORTE_CURLIM ¹		Ошибка превышения выходного тока в PORTE 0 – нет ошибки 1 – есть ошибка
3	PORTD_CURLIM ¹		Ошибка превышения выходного тока в PORTD 0 – нет ошибки 1 – есть ошибка
2	PORTC_CURLIM ¹		Ошибка превышения выходного тока в PORTC 0 – нет ошибки 1 – есть ошибка
1	PORTB_CURLIM ¹		Ошибка превышения выходного тока в PORTB 0 – нет ошибки 1 – есть ошибка
0	PORTA_CURLIM ¹		Ошибка превышения выходного тока в PORTA 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

6.5.1.20 Регистр 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	EMAC0_DE	ARC3_DE	ARC2_DE	ARC1_DE	ARC0_DE	MIL3_DE	MIL2_DE	MIL1_DE	MIL0_DE	CAN5_DE	CAN4_DE	CAN3_DE	CAN2_DE	CAN1_DE	CAN0_DE

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15	EMAC1_DE ¹	0	Двойная не исправимая ошибка в массивах памяти ETHERNETMAC1 0 – нет ошибки 1 – есть ошибка
14	EMAC0_DE ¹	0	Двойная не исправимая ошибка в массивах памяти ETHERNETMAC0 0 – нет ошибки 1 – есть ошибка
13	ARC3_DE ¹	0	Двойная не исправимая ошибка в массивах памяти ARC3 0 – нет ошибки 1 – есть ошибка
12	ARC2_DE ¹	0	Двойная не исправимая ошибка в массивах памяти ARC2 0 – нет ошибки 1 – есть ошибка
11	ARC1_DE ¹	0	Двойная не исправимая ошибка в массивах памяти ARC1 0 – нет ошибки 1 – есть ошибка
10	ARC0_DE ¹	0	Двойная не исправимая ошибка в массивах памяти ARC0 0 – нет ошибки 1 – есть ошибка
9	MIL3_DE ¹	0	Двойная не исправимая ошибка в массивах памяти MIL3 0 – нет ошибки 1 – есть ошибка
8	MIL2_DE ¹	0	Двойная не исправимая ошибка в массивах памяти MIL2 0 – нет ошибки 1 – есть ошибка
7	MIL1_DE ¹	0	Двойная не исправимая ошибка в массивах памяти MIL1 0 – нет ошибки 1 – есть ошибка
6	MIL0_DE ¹	0	Двойная не исправимая ошибка в массивах памяти MIL0 0 – нет ошибки 1 – есть ошибка
5	CAN5_DE ¹	0	Двойная не исправимая ошибка в массивах памяти CAN5 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
4	CAN4_DE ¹	0	Двойная не исправимая ошибка в массивах памяти CAN4 0 – нет ошибки 1 – есть ошибка
3	CAN3_DE ¹	0	Двойная не исправимая ошибка в массивах памяти CAN3 0 – нет ошибки 1 – есть ошибка
2	CAN2_DE ¹	0	Двойная не исправимая ошибка в массивах памяти CAN2 0 – нет ошибки 1 – есть ошибка
1	CAN1_DE ¹	0	Двойная не исправимая ошибка в массивах памяти CAN1 0 – нет ошибки 1 – есть ошибка
0	CAN0_DE ¹	0	Двойная не исправимая ошибка в массивах памяти CAN0 0 – нет ошибки 1 – есть ошибка

6.5.1.21 Регистр 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	EMAC0_DE	ARC3_DE	ARC2_DE	ARC1_DE	ARC0_DE	MIL3_DE	MIL2_DE	MIL1_DE	MIL0_DE	CAN5_DE	CAN4_DE	CAN3_DE	CAN2_DE	CAN1_DE	CAN0_DE

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15	EMAC1_SE ¹	0	Одиночная не исправимая ошибка в массивах памяти ETHERNETMAC1 0 – нет ошибки 1 – есть ошибка
14	EMAC0_SE ¹	0	Одиночная не исправимая ошибка в массивах памяти ETHERNETMAC0 0 – нет ошибки 1 – есть ошибка
13	ARC3_SE ¹	0	Одиночная не исправимая ошибка в массивах памяти ARC3 0 – нет ошибки 1 – есть ошибка
12	ARC2_SE ¹	0	Одиночная не исправимая ошибка в массивах памяти ARC2 0 – нет ошибки 1 – есть ошибка
11	ARC1_SE ¹	0	Одиночная не исправимая ошибка в массивах памяти ARC1 0 – нет ошибки 1 – есть ошибка

¹ Флаг сбрасывается записью 1. Если в момент сброса происходит очередное событие, флаг не сбрасывается.

Бит	Имя	Значение	Описание
10	ARC0_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти ARC0 0 – нет ошибки 1 – есть ошибка
9	MIL3_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти MIL3 0 – нет ошибки 1 – есть ошибка
8	MIL2_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти MIL2 0 – нет ошибки 1 – есть ошибка
7	MIL1_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти MIL1 0 – нет ошибки 1 – есть ошибка
6	MIL0_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти MIL0 0 – нет ошибки 1 – есть ошибка
5	CAN5_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти CAN5 0 – нет ошибки 1 – есть ошибка
4	CAN4_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти CAN4 0 – нет ошибки 1 – есть ошибка
3	CAN3_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти CAN3 0 – нет ошибки 1 – есть ошибка
2	CAN2_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти CAN2 0 – нет ошибки 1 – есть ошибка
1	CAN1_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти CAN1 0 – нет ошибки 1 – есть ошибка
0	CAN0_SE ¹	0	Одинокная не исправимая ошибка в массивах памяти CAN0 0 – нет ошибки 1 – есть ошибка

6.5.1.22 Регистр 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]													

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

6.5.1.23 Регистр 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]															

Бит	Имя	Значение	Описание
31...20	-		Зарезервировано
19...0	RESET_EN[19:0]	20'h1FFFFFF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT1 0 – сброс не формируется 1 – сброс формируется

6.5.1.24 Регистр 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[8:0]								

Бит	Имя	Значение	Описание
31...9	-		Зарезервировано
8...0	RESET_EN[8:0]	9'h1FF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT2 0 – сброс не формируется 1 – сброс формируется

6.5.1.25 Регистр 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[7:0]							

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	RESET_EN[7:0]	8'hFF	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT3 0 – сброс не формируется 1 – сброс формируется

6.5.1.26 Регистр 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[4:0]				

Бит	Имя	Значение	Описание
31...5	-		Зарезервировано
4...0	RESET_EN[4:0]	5'h1F	Биты разрешения запуска отложенного сброса при возникновении событий в EVENT4 0 – сброс не формируется 1 – сброс формируется

6.5.1.27 Регистр 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[28:16]													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
INT_EN[15:0]																

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28...0	INT_EN[28:0]	29'h0000	Биты разрешения запроса прерывания FT_IF2 по событиям в EVENT5 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

6.5.1.28 Регистр 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[13:0]														

Бит	Имя	Значение	Описание
31...13	-		Зарезервировано
12...0	INT_EN[13:0]	13'h0000	Биты разрешения запроса прерывания FT_IF2 по событиям в EVENT6 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

6.5.1.29 Регистр 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[9:0]									

Бит	Имя	Значение	Описание
31...10	-		Зарезервировано
9...0	INT_EN[9:0]	10'h0000	Биты разрешения запроса прерывания FT_IF2 по событиям в EVENT7 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

6.5.1.30 Регистр 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]										

Бит	Имя	Значение	Описание
31...11	-		Зарезервировано
10...0	INT_EN[10:0]	13'h0000	Биты разрешения запроса прерывания FT_IF2 по событиям в EVENT8 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

6.5.1.31 Регистр 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]															

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28...0	INT_EN[28:0]	29'h0000	Биты разрешения запроса прерывания FT_IF3 по событиям в EVENT9 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

6.5.1.32 Регистр 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]					

Бит	Имя	Значение	Описание
31...6	-		Зарезервировано
5...0	INT_EN[5:0]	6'h0000	Биты разрешения запроса прерывания FT_IF3 по событиям в EVENT10 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

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

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0	INT_EN[15:0]	16'h0000	Биты разрешения запроса прерывания FT_IF3 по событиям в EVENT11 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

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

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0	INT_EN[15:0]	16'h0000	Биты разрешения запроса прерывания FT_IF3 по событиям в EVENT12 0 – запрос прерывания не формируется 1 – запрос прерывания формируется

6.6 Контроллер сторожевых таймеров (WDT_CNTR)

Сторожевой таймер IWDG предназначен для аварийного перезапуска микроконтроллера, когда за период счета сторожевого таймера не будет выполнена хотя бы одна операция по перезапуску. Сторожевой таймер выполнен по сбоеустойчивой технологии, т.е. при возникновении одиночного сбоя таймер продолжит выполнять поставленную задачу. Для последующего устранения последствий сбоя требуется программная переинициализация таймера. Конфигурация блока защищена с помощью ECC кодирования, а сам счетчик построен по троированной схеме (см. рисунок 37).

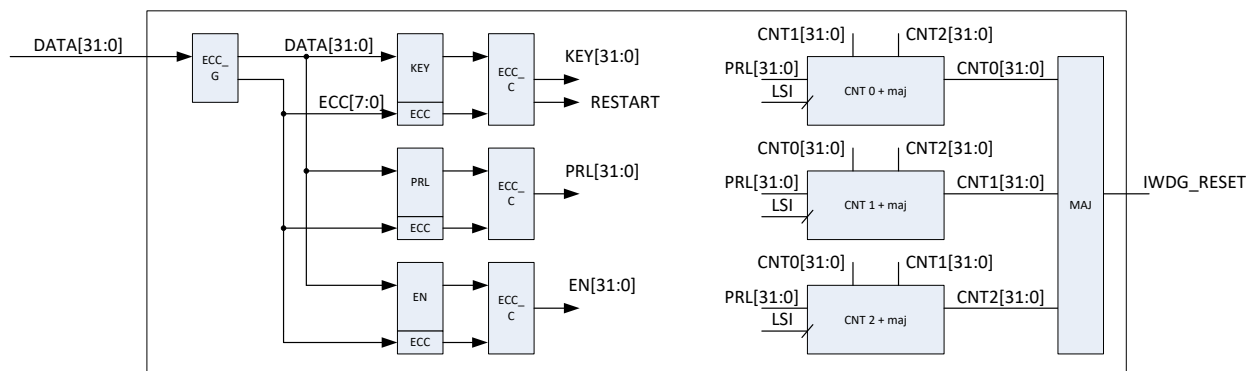


Рисунок 37 – Структурная схема контроллера троированного сторожевого таймера

Для инициализации сторожевого таймера необходимо:

- 1 разрешить тактирование блока WDT_CNTR в блоке управления системой тактирования CLK_CNTR;
- 2 в регистр KEY записать 0x05555;
- 3 считать регистр и убедиться, что записано верное значение;
- 4 в регистр PRL записать требуемое основание счета;
- 5 считать регистр и убедиться, что записано верное значение;
- 6 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 убедиться, что в него записано верное значение из регистра PRL;
- 2 в регистр KEY записать 0x0CCCC;
- 3 считать регистр и убедиться, что записано верное значение;
- 4 в регистр EN записать значение 0x03333. После чего сторожевой таймер начнет работу;
- 5 в блоке FT_CNTR настроить требуемую реакцию на возникновения сбоев в блоке сторожевого таймера;
- 6 при возникновении сбоев провести операции по переинициализации сторожевого таймера;
- 7 с периодом меньшим заданного через регистр PRL проводить перезапуск сторожевого таймера;
- 8 отключить тактирование блока IWDG_CNTR;
- 9 контролировать, что генератор LSI работает.

Для перезапуска сторожевого таймера необходимо:

- 1 разрешить тактирование блока IWDG_CNTR;
- 2 при необходимости считать значение CNT и проверить, что оно находится в ожидаемом диапазоне;

- 3 записать в регистр KEY значение 0x0AAAA;
- 4 в блоке FT_CNTR проверить наличие флагов сбоя в блоке сторожевого таймера;
- 5 отключить тактирование блока IWDG_CNTR.

Для переинициализации сторожевого таймера необходимо:

- 1 разрешить тактирование блока IWDG_CNTR;
- 2 в регистр KEY записать 0x0CCCC;
- 3 считать регистр и убедиться, что записано верное значение;
- 4 в регистр EN записать значение отличное от 0x03333. После чего будет остановлен счет;
- 5 выполнить операции по инициализации сторожевого таймера.

6.6.1 Описание регистров

Таблица 42 – Описание регистров контроллера сторожевых таймеров

Базовый Адрес	Название	Описание
0x4000_4000	IWDG	Сторожевой таймер IWDG
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

6.6.1.1 Регистр 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]															

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 разрешающие запись в регистр разрешения работы EN сторожевого таймера. Запись 0x0AAAA – перезапускает счет сторожевого таймера. Значения, отличные от указанных выше, не имеют значения

6.6.1.2 Регистр 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]															

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 частоте LSI генератора. 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

6.6.1.3 Регистр 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]															

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

6.6.1.4 Регистр 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]															

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

6.7 Контроллер ПЗУ (ROMCNTR)

Блок ПЗУ используются всегда, и контроллер предоставляет центральному процессору информацию об обнаруженных ошибках при работе с ПЗУ. Контроллер ПЗУ не имеет каких-либо функций по управлению данным блоком.

6.7.1 Описание регистров

Т а б л и ц а 43 – Описание регистров контроллера ПЗУ

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x40009000	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	ECCCS0		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

6.7.1.1 Регистр KEY

Base ADDR=	0x40009000	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]															

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

6.7.1.2 Регистр ECCCS

Base ADDR=	0x40009000	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

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

6.7.1.3 Регистр ECCADR

Base ADDR=		0x40009000				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]																	

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

6.7.1.4 Регистр ECCDATA

Base ADDR=		0x40009000				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]																	

Бит	Имя	Значение	Описание
31...0	ECCDATA[31:0]		Данные, считанные при последней двойной или одинарной ошибке, без корректировки ECC

6.7.1.5 Регистр ECCECC

Base ADDR=		0x40009000				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]							

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]		Считанные ECC биты при последней двойной или одинарной ошибке, без корректировки ECC

6.7.1.6 Регистр TEST_TUNING

Base ADDR=		0x40009000				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]	

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

6.8 Контроллер ОЗУ (RAMC RAMD)

6.8.1 Описание регистров

Таблица 44 – Описание регистров контроллера ОЗУ

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x40008000	0x0000_0000	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0004	ECCCS		Регистр статуса ошибок ECCROM
	0x0000_0008	ECCADR		Регистр статуса ошибок ECCDRAM
	0x0000_000C	ECCDATA		Регистр статуса ошибок ECCDRAM
	0x0000_0010	ECCECC		Регистр данных ECC последней ошибки
	0x0000_0014	TEST_TUNING		Регистр подстройки параметров блока

6.8.1.1 Регистр KEY

Base ADDR=	0x40008000	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]															

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

6.8.1.2 Регистр ECCCS

Base ADDR=		0x40008000				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	

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

6.8.1.3 Регистр ECCADR

Base ADDR=		0x40008000				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]																				

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

6.8.1.4 Регистр ECCDATA

Base ADDR=		0x40008000				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]																	

Бит	Имя	Значение	Описание
31...0	ECCDATA[31:0]		Данные, считанные при последней двойной или одинарной ошибке, без корректировки ECC

6.8.1.5 Регистр ECCECC

Base ADDR=		0x40008000					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]														

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ECC[7:0]		Считанные ECC биты при последней двойной или одинарной ошибке, без корректировки ECC

6.8.1.6 Регистр TEST_TUNING

Base ADDR=		0x40008000				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]		

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

6.9 Контроллер DMA (DMACNTR)

6.9.1 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.

6.9.2 Термины и определения

При описании контроллера DMA используются термины, приведенные в таблице 45.

Таблица 45 – Термины и определения

Термины/определения	Описание
Альтернативная	Альтернативная структура управляющих данных канала. Для изменения типа структуры данных можно установить соответствующий регистр (см. подраздел 6.9.5 «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-х. 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.

Термины/определения	Описание
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 <code>chnl_pri_alt_set</code> установлен в 0.
R	Степень числа 2, устанавливающая число передач DMA, которые могут произойти перед сменой арбитража. Количество передач DMA программируется в диапазоне от 1 до 1024 двоичными шагами от 2 в степени 0 до 2 в степени 100
Исполнение с изменением конфигурации	Режим работы для выбранного канала, при котором контроллер получает запрос от периферии и выполняет 4 DMA передачи, используя первичную структуру управляющих данных, которые настраивают альтернативную структуру управляющих данных. После чего контроллер начинает цикл DMA, используя альтернативную структуру данных. После 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 <code>dma_done</code> , если окончание подобного режима работы происходит после выполнения обычного цикла

6.9.3 Функциональное описание

На рисунке 38 показана упрощенная структурная схема контроллера DMA.

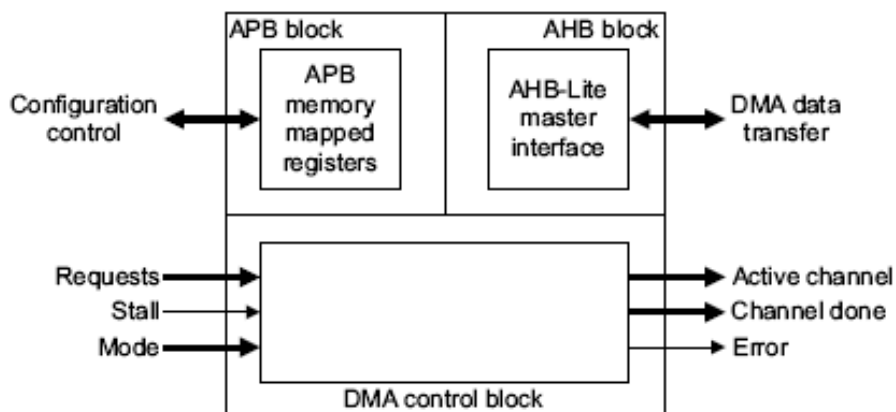


Рисунок 38 – Структурная схема контроллера DMA

Контроллер состоит из следующих основных функциональных блоков:

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

6.9.3.1 Типы передач

Контроллер интерфейса не поддерживает пакетные передачи, выполняет только одиночные передачи. Отсутствие возможности осуществлять пакетные передачи оказывает минимальное влияние на производительность системы, так как пакетные передачи более эффективны в одноуровневых системах с шиной AHB, где блоки должны «захватывать» шину или обращаться к внешней памяти. В тоже время, контроллер DMA предназначен для использования в многоуровневых системах с шиной AHB, включающих встроенную память.

6.9.3.2 Разрядность передач данных

Контроллер интерфейса предоставляет возможность осуществлять передачу 8-, 16- и 32-разрядных данных. В таблице 46 приведены комбинации значений шины HSIZE.

Таблица 46 – 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-разрядными данными при обращении к управляющим данным канала. Необходимо устанавливать разрядность данных источника, соответствующую разрядности данных приемника.

6.9.3.3 Управление защитой данных

Контроллер позволяет устанавливать режимы защиты данных протокола ANB-Lite, определяемые шиной HPROT[3:1]. Возможен выбор следующих режимов защиты:

- кэширование;
- буферизация;
- привилегированный.

В таблице 47 приведены значения комбинаций шины HPROT.

Таблица 47 – 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 подразделе «Структура управляющих данных канала»).

Для каждого канала DMA также возможен выбор режима защиты данных (подробнее см. в подразделе 6.9.4 «Управление DMA»).

1.1.1.1 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 48 перечислены возможные комбинации.

Таблица 48 – 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 подразделе «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. подраздел «Структура управляющих данных канала»).

6.9.4 Управление DMA

6.9.4.1 Правила обмена данными

Контроллер использует правила обмена данными, перечисленные в таблице 49, 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 49 – Правила, при которых передача данных по каналам разрешена, и запросы не маскируются

Правило	Описание
1	Если <code>dma_active[C]</code> установлен в 0, установка в 1 <code>dma_req[C]</code> или <code>dma_sreq[C]</code> на один или более тактов сигнала <code>hclk</code> , следующих или не следующих друг за другом, инициирует передачу по каналу номер C
2	Контроллер осуществляет установку в 1 только одного разряда <code>dma_active[C]</code>
3	Контроллер устанавливает в 1 <code>dma_active[C]</code> в момент начала передачи по каналу C
4	Для типов циклов DMA, отличных от периферийного «Исполнение с изменением конфигурации», <code>dma_active[C]</code> остается в состоянии 1 до тех пор, пока контроллер не окончит передачи с номерами меньше, чем значение 2^R или чем число передач, указанное в регистре <code>n_minus_1</code> . В периферийном режиме «Исполнение с изменением конфигурации», <code>dma_active[C]</code> остается в состоянии 1 в течение каждой пары DMA передач, с использованием первичной и альтернативной структур управляющих данных. 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 <code>n_minus_1</code>), используя альтернативную структуру управляющих данных. По окончании последней передачи <code>dma_active[C]</code> сбрасывается в 0
5	Контроллер устанавливает <code>dma_active[C]</code> в 0 на, как минимум, один такт сигнала <code>hclk</code> перед тем, как снова установит <code>dma_active[C]</code> или <code>dma_active[]</code> в 1
6	Для каналов, по которым разрешена передача, контроллер осуществляет установку в 1 только одного <code>dma_done[]</code>
7	Если <code>dma_req[C]</code> устанавливается в состояние 1 в момент, когда <code>dma_active[C]</code> или <code>dma_stall</code> также в состоянии 1, то это означает, что контроллер обнаружил запрос
8	Если разряды <code>cycle_ctrl</code> для канала установлены в состояние 3'b100, 3'b101, 3'b110, 3'b111, то <code>dma_done[C]</code> никогда не будет установлен в 1
9	Если все передачи по каналу завершены, и разряды <code>cycle_ctrl</code> позволяют удержание <code>dma_done[C]</code> , то по срезу сигнала <code>dma_active[]</code> произойдут события: - если <code>dma_stall</code> в состоянии 0, контроллер устанавливает <code>dma_done[]</code> в состояние 1 продолжительностью один такт <code>hclk</code>; - если <code>dma_stall</code> в состоянии 1, работа контроллера приостановлена. После того, как <code>dma_stall</code> будет установлен в 0, контроллер устанавливает <code>dma_done[]</code> в состояние 1 продолжительностью один такт <code>hclk</code>
10	Состояние <code>dma_waitonreq[C]</code> можно изменять только при выключенном канале

Правило	Описание
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
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, 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, 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, приоритет предоставляется dma_req[c], и контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных, затем без осуществления арбитража выполняет передачи с номерами меньше, чем значение 2^R (или чем число передач, указанное в регистре n_minus_1), используя альтернативную структуру управляющих данных
18	Когда chnl_req_mask_set[C] установлен в 1, контроллер игнорирует запросы по dma_sreq[C] и dma_req[C]
* – Необходимо с осторожностью устанавливать эти разряды. Если значение, указанное в регистре n_minus_1 меньше, чем значение 2^R, 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, контроллер никогда не выполнит необходимых передач	

При отключении канала контроллер осуществляет DMA передачи согласно правилам, представленным в таблице 50.

Т а б л и ц а 50 – 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

6.9.4.2 Диаграммы работы контроллера DMA

Данный раздел описывает примеры функционирования контроллера с использованием правил обмена данными, представленных ранее (см. таблицу 49):

- импульсный запрос на обработку;
- запрос по уровню на обработку;
- флаги завершения;
- флаги ожидания запроса на обработку.

Примечание – Все диаграммы, показанные далее на рисунках 39 – 43 в данном подразделе, подразумевают следующее:

- hready находится в состоянии 1;
- АНВ «ведомый» всегда дает ответ «OKAY».

6.9.4.3 Импульсный запрос на обработку

На рисунке 39 приведена временная диаграмма работы контроллера DMA при получении импульсного запроса от периферии.

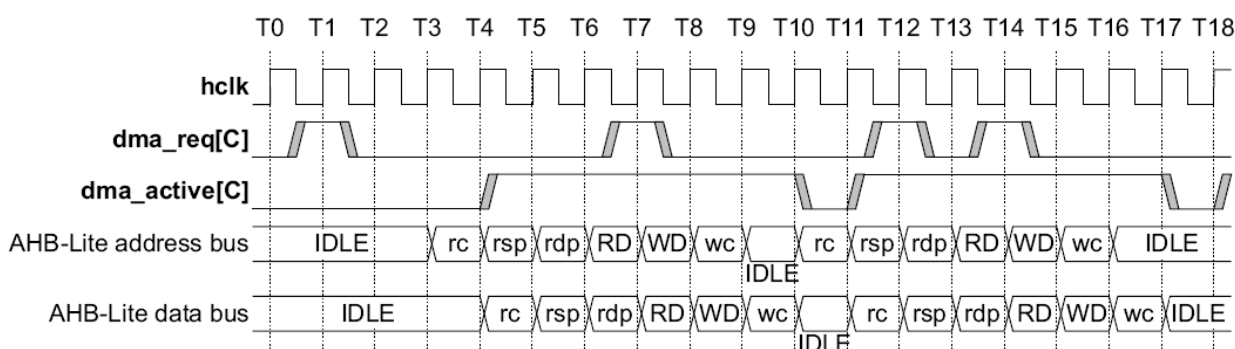


Рисунок 39 – Диаграмма работы при получении импульсного запроса

Пояснения к диаграмме на рисунке 39 приведены ниже в таблице 51.

Таблица 51 – Пояснения к диаграмме работы при получении импульсного запроса

T1	Контроллер обнаружил запрос на обработку по каналу С (Таблица 49, правило 1) при условии, что <code>chnl_req_mask_set[C]</code> находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает <code>dma_active[C]</code> (см. правила 2 и 3) и начинает DMA передачи по каналу С
T4-T7	Контроллер считывает управляющую данные канала, где: <code>rc</code> – чтение настроек канала, <code>channel_cfg</code> ; <code>rsp</code> – чтение указателя адреса окончания данных источника, <code>src_data_end_ptr</code> ; <code>rdp</code> – чтение указателя адреса окончания данных приемника, <code>dst_data_end_ptr</code>
T7	При установленном <code>dma_active[C]</code> в 1 и при условии, что <code>chnl_req_mask_set[C]</code> находится в состоянии 0, контроллер обнаруживает импульс запроса на обработки по каналу С (см. правило 7). Контроллер обрабатывает этот запрос в течение следующего арбитража
T7-T9	Контроллер выполняет передачу DMA по каналу С, где: <code>RD</code> – чтение данных; <code>WD</code> – запись данных
T9-T10	Контроллер осуществляет запись настроек канала, <code>channel_cfg</code> , где <code>ws</code> – запись настроек канала, <code>channel_cfg</code>
T10	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4)
T10-T11	Контроллер удерживает <code>dma_active[C]</code> на, как минимум, один такт <code>hclk</code> (см. правило 5)
T11	Если канал С имеет более высокий приоритет, то контроллер устанавливает <code>dma_active[C]</code> , так как ранее на такте T7 был получен запрос на обработку (см. правила 2 и 3)
T12	При установленном <code>dma_active[C]</code> в 1 и при условии, что <code>chnl_req_mask_set[C]</code> находится в состоянии 0, контроллер обнаруживает импульс запроса на обработки по каналу С (см. правило 7). Контроллер обрабатывает этот запрос в течение следующего арбитража
T14	Контроллер игнорирует запрос по каналу С из-за отложенного запроса, полученного на такте T12
T17	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4)
T17-T18	Контроллер удерживает <code>dma_active[C]</code> , как минимум, на один такт <code>hclk</code> (см. правило 5)
T18	Если канал С имеет более высокий приоритет, то контроллер устанавливает <code>dma_active[C]</code> , так как ранее на такте T12 был получен запрос на обработку (см. правила 2 и 3)

6.9.4.4 Запрос на обработку по уровню

На рисунке 40 показана временная диаграмма работы контроллера DMA при получении от периферии запроса на обработку по уровню.

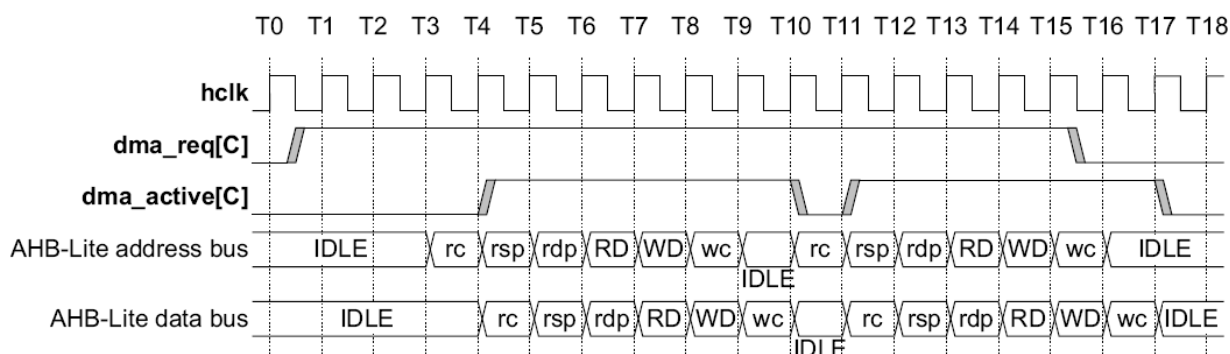


Рисунок 40 – Диаграмма работы при получении запроса на обработку по уровню

Пояснения к диаграмме на рисунке 40 приведены в таблице 52.

Т а б л и ц а 52 – Пояснения к диаграмме работы при получении запроса на обработку по уровню

T1	Контроллер обнаружил запрос на обработку по каналу C (Т а б л и ц а 49, правило 1) при условии, что <code>chnl_req_mask_set[C]</code> находится в состоянии 0 (см. правило 18)
T4	Контроллер устанавливает <code>dma_active[C]</code> (см. правила 2 и 3) и начинает DMA передачи по каналу C
T4-T7	Контроллер считывает управляющие данные канала, где: rc – чтение настроек канала, <code>channel_cfg</code> ; rsp – чтение указателя адреса окончания данных источника, <code>src_data_end_ptr</code> ; rdp – чтение указателя адреса окончания данных приемника, <code>dst_data_end_ptr</code>
T7-T9	Контроллер выполняет передачу DMA по каналу C, где: RD – чтение данных WD – запись данных
T9-T10	Контроллер осуществляет запись настроек канала, <code>channel_cfg</code> , где wc – запись настроек канала, <code>channel_cfg</code>
T10	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , что указывает на окончание передачи DMA (см. правило 4). Контроллер обнаружил запрос на обработку по каналу C (см. правило 1) при условии, что <code>chnl_req_mask_set[C]</code> находится в состоянии 0 (см. правило 18).
T10-T11	Контроллер удерживает <code>dma_active[C]</code> на как минимум один такт <code>hclk</code> (см. правило 5)
T11	Если канал C имеет более высокий приоритет, то контроллер устанавливает <code>dma_active[C]</code> 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 <code>dma_req[C]</code>
T16-T17	Контроллер осуществляет запись настроек канала <code>channel_cfg</code>
T17	Контроллер сбрасывает сигнал <code>dma_active[C]</code> , 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.

6.9.4.5 Флаги завершения

На рисунке 41 продемонстрировано 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.

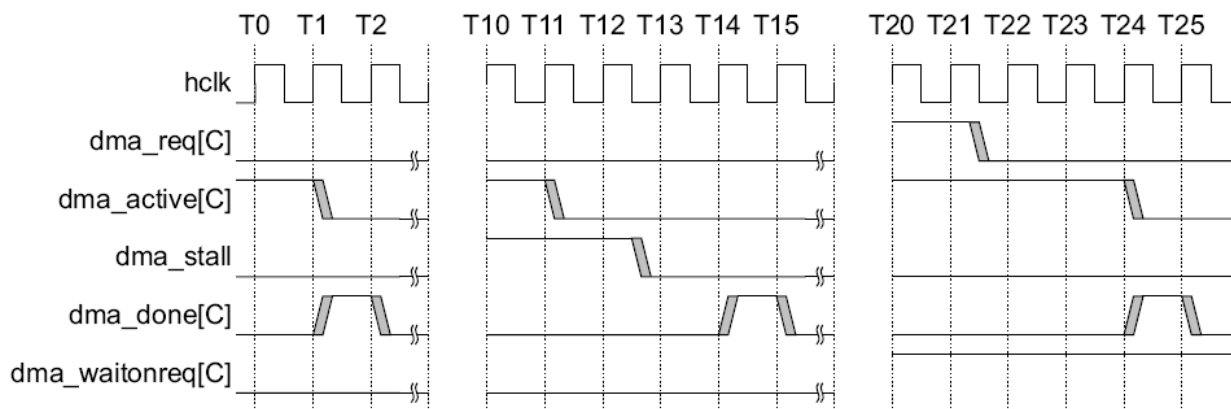


Рисунок 41 – Диаграммы функционирования dma_done

Пояснения к диаграмме на рисунке 41 приведены в таблицах 53 – 55.

Таблица 53 – Пояснения функционирования dma_done, такты от T0 до T2

T1	Контроллер сбрасывает сигнал dma_active[C], что указывает на окончание передачи DMA (Таблица 49, 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 54 – 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 55 – 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)

6.9.4.6 Флаги ожидания запроса на обработку

Ниже приведены рисунки 42 и 43, которые демонстрируют примеры использования флагов ожидания запроса на обработку при выполнении 2^R передач и одиночных передач:

- диаграмма работы контроллера DMA при использовании периферией dma_waitonreq;
- диаграмма работы контроллера DMA при использовании периферией dma_waitonreq совместно с dma_sreq.

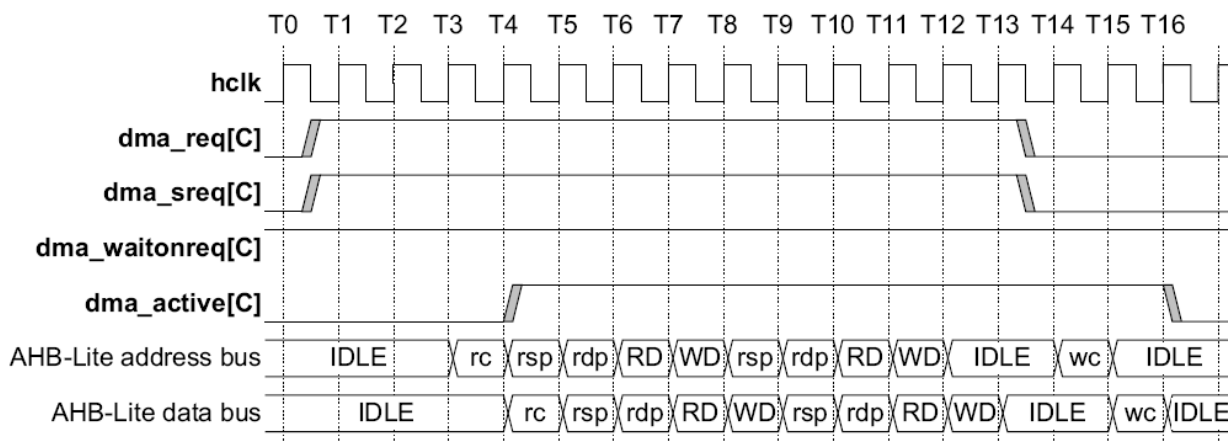


Рисунок 42 – Диаграмма работы контроллера DMA при использовании dma_waitonreq

Пояснения к диаграмме на рисунке 42 приведены в таблице 56.

Таблица 56 – 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 43 показана работа контроллера DMA при установке dma_waitonreq в 1 и выполнении одиночной DMA передачи.

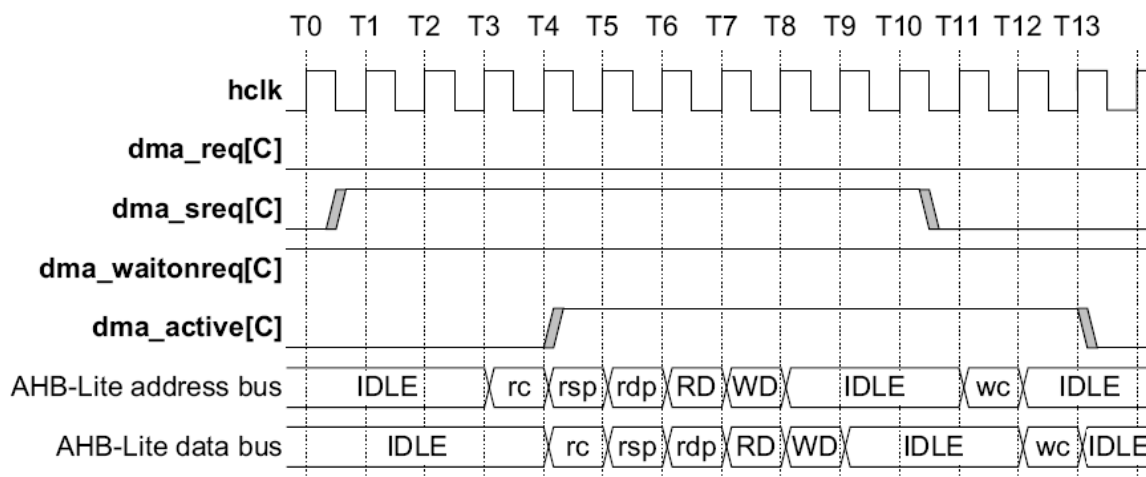


Рисунок 43 – Работа DMA при использовании dma_waitonreq совместно с dma_sreq

Пояснения к диаграмме на рисунке 43 приведены в таблице 57.

Таблица 57 – 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)

6.9.4.7 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 АНВ 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, арбитраж будет проводиться через каждые 16 передач DMA.

Т а б л и ц а 58 показывает возможную периодичность арбитража.

Т а б л и ц а 58 – Периодичность арбитража в единицах передач по шине АНВ

Значение R	Периодичность арбитража каждые x передач DMA
b0000	1
b0001	2
b0010	4
b0011	8
b0100	16
b0101	32
b0110	64
b0111	128
b1000	256
b1001	512
b1010–b1111	1024

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 разделе «Управляющие данные канала».

6.9.4.8 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. Т а б л и ц а 59 показывает уровень приоритета каналов DMA в порядке его уменьшения.

Таблица 59 – Уровень приоритета каналов DMA

Уровень приоритета в порядке его уменьшения	Номер канала	Уровень приоритета 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. Рисунок 44 иллюстрирует процесс выбора следующего канала для обслуживания.



Рисунок 44 – Алгоритм выбора следующего канала для обслуживания

6.9.4.9 Типы циклов DMA

Разряды `cycle_ctrl` определяют, как контроллер будет выполнять циклы DMA. Описание значений этих разрядов приведено в таблице 60.

Таблица 60 – Типы циклов DMA

<code>cycle_ctrl</code>	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. раздел «Настройка управляющих данных канала».

Для всех типов циклов DMA повторный арбитраж происходит после 2^R передач DMA. Если установить длинный период арбитража на низкоприоритетном канале, 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 основном 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, контроллер начинает обслуживание этого канала;

- если периферийный блок или программное обеспечение выдает запрос на обработку (повторный запрос на обработку по каналу), 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 режиме авто-запрос, 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 Осуществление арбитража:
 - если высокоприоритетный канал выдает запрос на обработку, контроллер начинает обслуживание этого канала;
 - если периферийный блок или программное обеспечение выдает запрос на обработку (повторный запрос на обработку по каналу), 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 режиме пинг-понг 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 45 демонстрирует пример функционирования контроллера в режиме «пинг-понг».

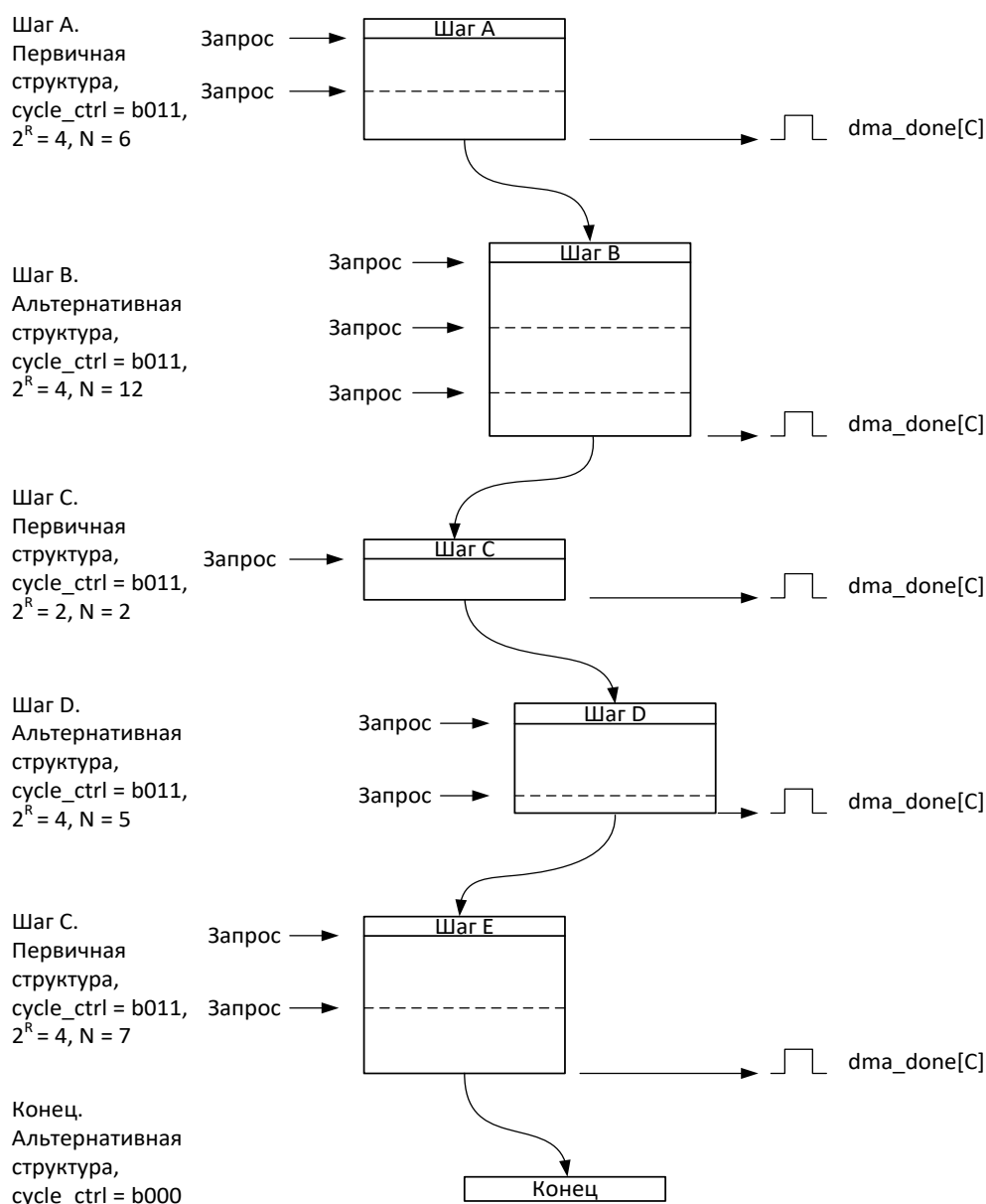


Рисунок 45 – Пример функционирования контроллера в режиме «пинг-понг»

Пояснения к схеме на рисунке 45:

- Шаг А** Процессор устанавливает первичную структуру управляющих данных для шага А. Процессор устанавливает альтернативную структуру управляющих данных для шага В. Это позволит контроллеру преступить к шагу В незамедлительно после выполнения шага А, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала. Контроллер получает запрос и выполняет 4 передачи DMA. Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов. Контроллер выполняет оставшиеся 2 передачи DMA. Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После выполнения шага А процессор может установить первичные управляющие данные канала для шага С. Это позволит контроллеру переключиться к

шагу С незамедлительно после выполнения шага В, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета выполняется шаг В:

Шаг В Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 4 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После выполнения шага В процессор может установить альтернативные управляющие данные канала для шага D.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета выполняется шаг С:

Шаг С Контроллер выполняет 2 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После выполнения шага С процессор может установить первичные управляющие данные канала для шага Е.

После получения нового запроса на обработку от канала при условии его наивысшего приоритета выполняется шаг D:

Шаг D Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов
Контроллер выполняет оставшуюся передачу DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

После получения нового запроса на обработку от канала при условии его наивысшего приоритета выполняется шаг Е:

Шаг Е Контроллер выполняет 4 передачи DMA.
Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.
Контроллер выполняет оставшиеся 3 передачи DMA.
Контроллер устанавливает dma_done[C] в состояние 1 на один такт сигнала синхронизации hclk и входит в процедуру арбитража

Если контроллер получит новый запрос на обработку от данного канала и этот запрос будет самым приоритетным, контроллер предпримет попытку выполнения следующего шага. Однако из-за того, что процессор не установил альтернативные управляющие данные, и по окончании шага D контроллер установил cycle_ctrl в состояние b000, передачи DMA прекращаются.

Примечание – Для прерывания цикла DMA, исполняемого в режиме «пинг-понг», также возможен перевод режима работы контроллера на шаге E в режим «Основной цикл 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 четыре 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. Таблица 61 перечисляет области памяти channel_cfg, как те, которые должны быть определены константами, так и те, значения которых определяются пользователем.

Т а б л и ц а 61 – Channel_cfg для первичной структуры управляющих данных в режиме работы с памятью «исполнение с изменением конфигурации»

№ 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 4 передачи DMA
3	next_useburst	b'0	Для данного режима этот разряд должен быть равен 0
2...0	cycle_ctrl	b'100	Контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями, определяемыми пользователем			
23...21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20...18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13...4	n_minus_1	N*	Настраивает контроллер на выполнение N передач DMA, где N кратно 4
* Так как R_power задает значение 4, необходимо задавать значение N, кратное 4. Число, равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные			

Рисунок 46 демонстрирует пример функционирования в режиме работы с памятью «Исполнение с изменением конфигурации».

Инициализация:

- 1 Настройка первичных управляющих данных для разрешения копирования А, В, С 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	0xFFFFFFFF
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b101, $2^R = 2$, N = 8	0xFFFFFFFF
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b101, $2^R = 8$, N = 5	0xFFFFFFFF
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, $2^R = 4$, N = 4	0xFFFFFFFF

Memory scatter-gather transaction:

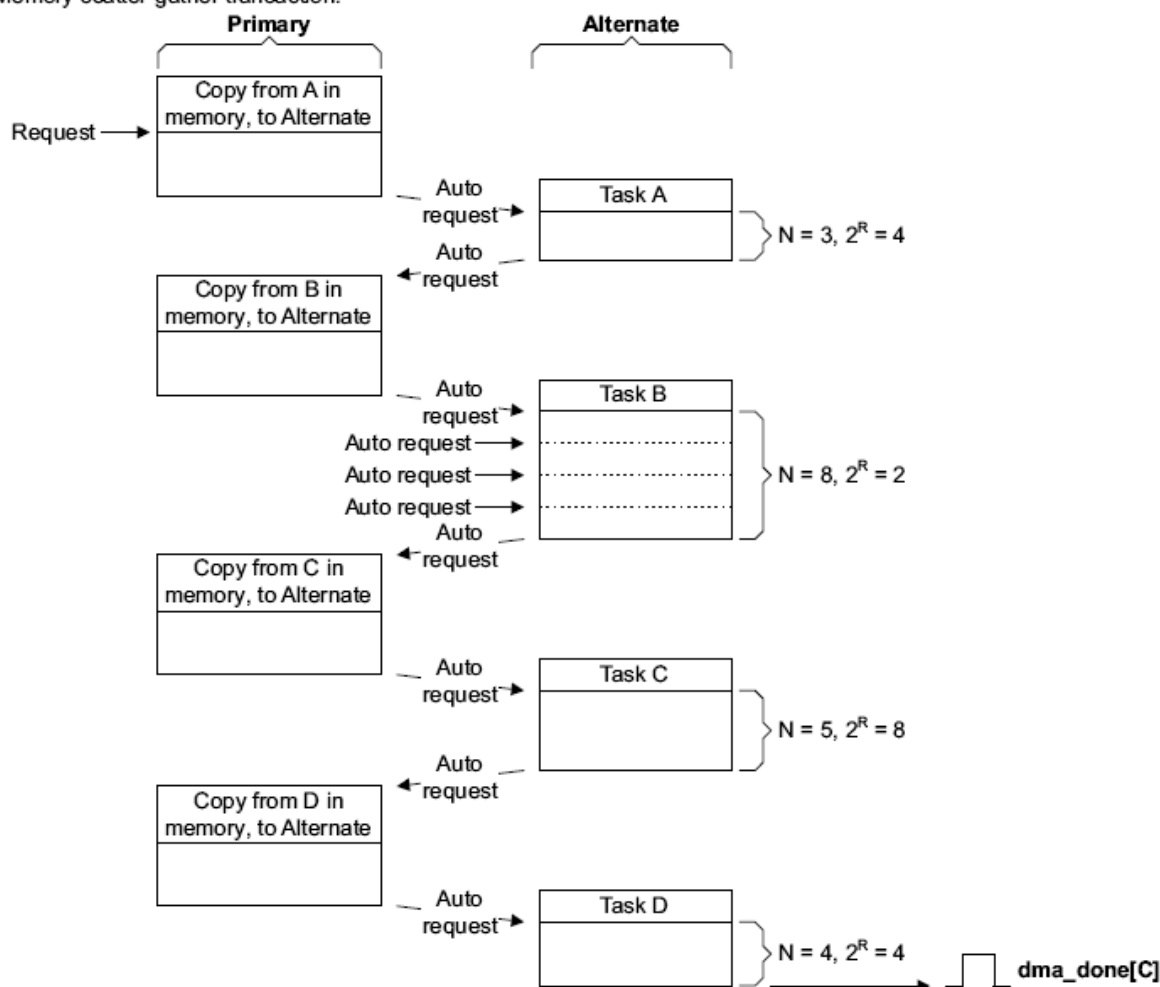


Рисунок 46 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

Пояснения к схеме на рисунке 46:

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 4 слов, необходимо установить 2^R в 4. В этом примере количество задач равно 4 и поэтому N установлен в 16.

2 Процессор записывает управляющие данные для шагов А, В, С, D в область памяти с адресом, указанным в `src_data_end_ptr`.

3 Процессор разрешает работу канала DMA. Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по `dma_req[]` или запроса от процессора. Порядок выполнения следующий:

Первичная, копирование А

По получении запроса на обслуживание контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага А. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг А

Контроллер выполняет шаг А. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование В

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага В. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг В

Контроллер выполняет шаг В. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование С

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага С. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Шаг С

Контроллер выполняет шаг С. По окончании контроллер генерирует автозапрос для канала и проводит процедуру арбитража.

Первичная, копирование D

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D. Контроллер устанавливает `cycle_ctrl` первичных управляющих данных в `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 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные управляющие данные без осуществления арбитража и не устанавливая сигнал `dma_active[C]` в 0.

Примечание – Это единственный случай, при котором контроллер не осуществляет процедуру арбитража после выполнения передачи DMA, используя первичные управляющие данные.

После того, как этот цикл завершился, контроллер выполняет арбитраж и по получении запроса на обслуживание от периферии, имеющего наивысший приоритет, он выполняет еще 4 передачи DMA, используя первичные управляющие данные. По окончании этих передач контроллер начинает цикл DMA, используя альтернативные

управляющие данные без осуществления арбитража и не устанавливая сигнал dma_active[C] в 0.

Контроллер продолжает выполнять циклы 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.

Таблица 62 перечисляет области памяти channel_cfg, как те, которые должны быть определены константами, так и те, значения которых определяются пользователем.

Таблица 62 – Channel_cfg для первичной структуры управляющих данных в режиме работы с периферией «Исполнение с изменением конфигурации»

№ 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 4 передачи DMA
2...0	cycle_ctrl	b'110	Контроллер работает в режиме работы с периферией «исполнение с изменением конфигурации»
Области со значениями, определяемыми пользователем			
23...21	dst_prot_ctrl	-	Определяет состояние HPROT при записи данных в приемник
20...18	src_prot_ctrl	-	Определяет состояние HPROT при чтении данных из источника
13...4	n_minus_1	N ^{*)}	Настраивает контроллер на выполнение N передач DMA, где N кратно 4
3	next_useburst	-	При установке в 1 контроллер установит chnl_useburst_set[C] в 1 после выполнения передачи с альтернативной структурой
* Так как R_power задает значение 4, необходимо задавать значение N, кратное 4. Число, равное N/4 – это количество раз, которое нужно настраивать альтернативные управляющие данные			

Рисунок 47 демонстрирует пример функционирования в режиме работы с периферией «исполнение с изменением конфигурации».

Инициализация:

- 1 Настройка первичных управляющих данных для разрешения копирования A, B, C и D: cycle_ctrl=b'110, 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	0xFFFFFFFF
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b111, $2^R = 2$, N = 8	0xFFFFFFFF
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b111, $2^R = 8$, N = 5	0xFFFFFFFF
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, $2^R = 4$, N = 4	0xFFFFFFFF

Peripheral scatter-gather transaction:

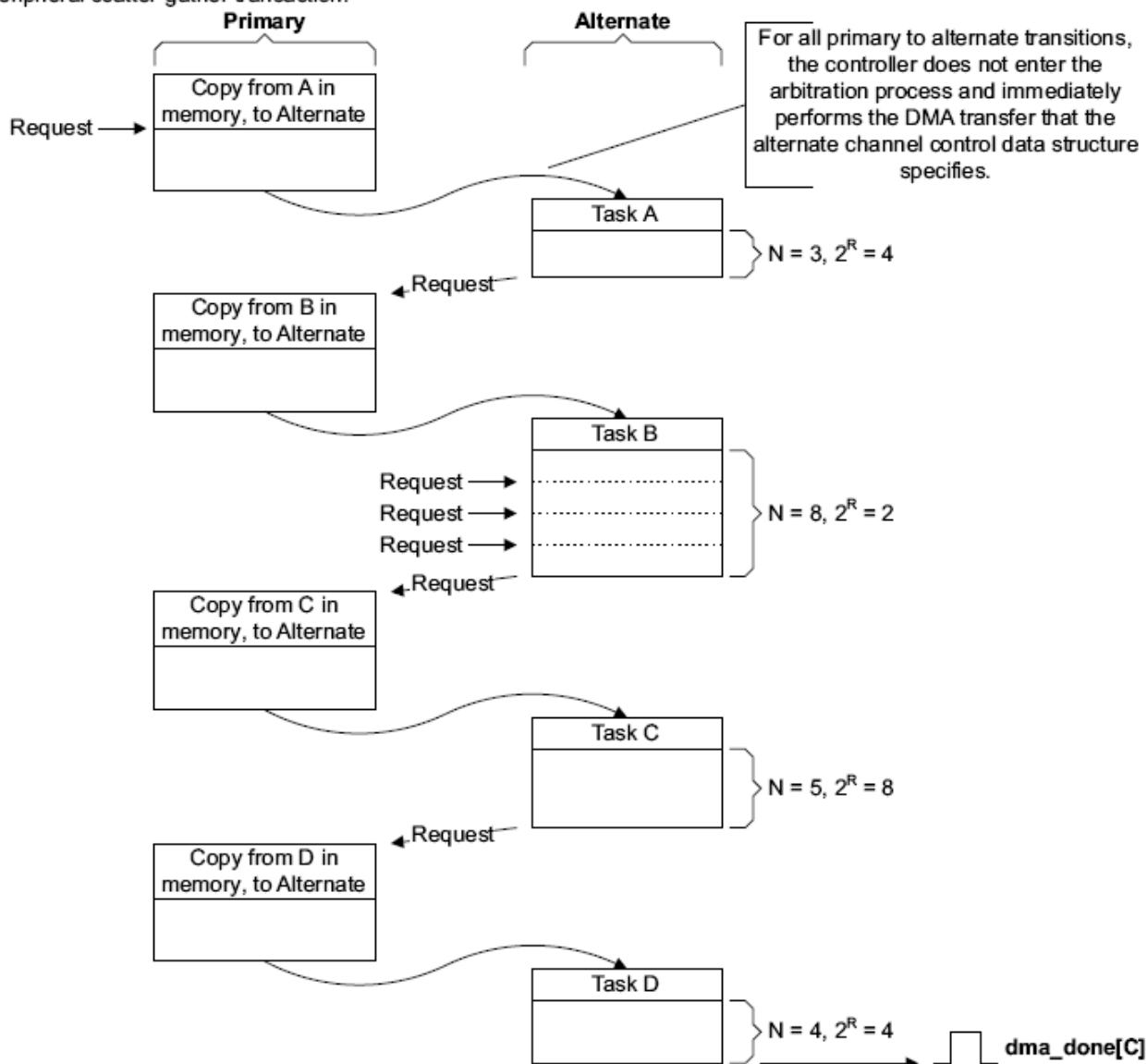


Рисунок 47 – Пример работы DMA в режиме с «Исполнением с изменением конфигурации»

Пояснения к схеме на рисунке 47:

Инициализация:

- 1 Процессор настраивает первичную структуру управляющих данных для работы в режиме работы с периферией «исполнение с изменением конфигурации» путем установки cycle_ctrl в b110. Так как управляющие данные канала состоят из 4 слов, необходимо установить 2^R в 4. В этом примере количество задач равно 4 и поэтому N установлен в 16.

- 2 Процессор записывает управляющие данные для шагов А, В, С, D в область памяти с адресом, указанным в `src_data_end_ptr`.
- 3 Процессор разрешает работу канала DMA.

Передачи в данном режиме начинают исполняться при получении контроллером запроса на обслуживание по `dma_req[]`. Передачи выполняются следующим образом:

Первичная, копирование из области А памяти

По получению запроса на обслуживание, контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага А.

Шаг А

Контроллер выполняет шаг А. По окончании контроллер проводит процедуру арбитража. Первичная, копирование из области В памяти. Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага В.

Шаг В

Контроллер выполняет шаг В. Для завершения задачи периферия должна установить последовательно 3 запроса. По окончании контроллер проводит процедуру арбитража. Первичная, копирование из области С памяти. Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага С.

Шаг С

Контроллер выполняет шаг С. По окончании контроллер проводит процедуру арбитража. После выставления периферией нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается следующим образом:

Первичная, копирование из области D памяти

Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D.

Контроллер устанавливает `cycle_ctrl` первичных управляющих данных в `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.

6.9.4.10 Индикация ошибок

При получении контроллером по шине АHB 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.

6.9.5 Структура управляющих данных канала

В системной памяти должна быть отведена область для хранения управляющих данных каналов. Системная память должна:

- предоставлять смежную область системной памяти, к которой контроллер и процессор имеют доступ;
- иметь базовый адрес, который целочисленно кратен общему размеру структуры управляющих данных канала.

Рисунок 48 показывает область памяти, необходимую контроллеру для структур управляющих данных канала, при использовании всех 32-х каналов и опциональной альтернативной структуры управляющих данных.

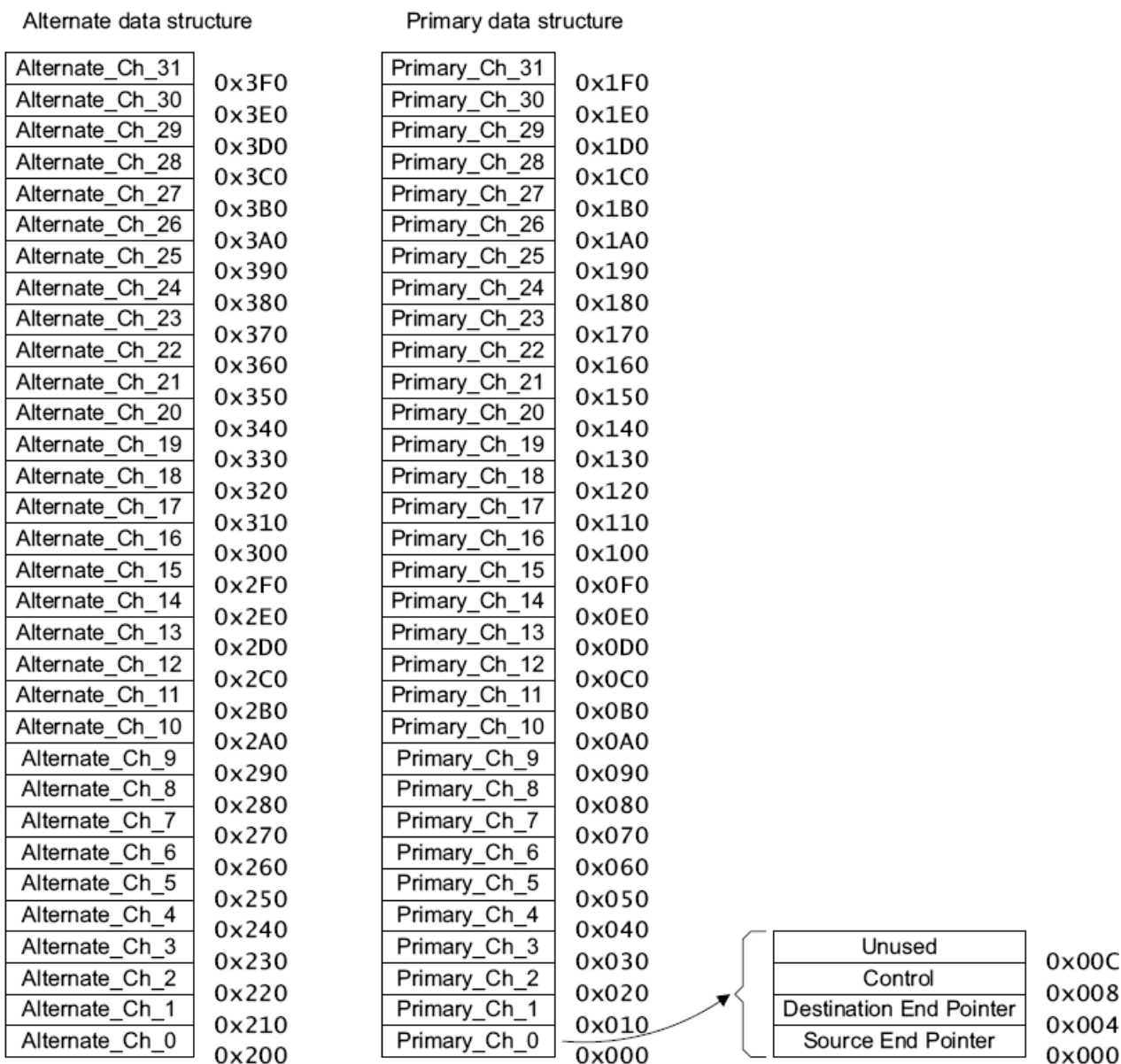


Рисунок 48 – Карта памяти для 32-х каналов, включая альтернативную структуру

Пример, показанный на рисунке, использует 1 Кбайт системной памяти. В этом примере контроллер использует младшие 0x10 разрядов адреса для доступа ко всем элементам структуры управляющих данных, и поэтому базовый адрес структуры должен быть 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.

Таблица 63 перечисляет разряды адреса, обеспечивающие контроллеру доступ к различным элементам структуры управляющих данных, в зависимости от количества каналов, используемых в контроллере.

Т а б л и ц а 63 – Разряды адреса, соответствующие элементам структуры управляющих данных

Количество каналов, используемых в контроллере	Разряды адреса						
	[9]	[8]	[7]	[6]	[5]	[4]	[3:0]
1						A	0x0 0x4 0x8
2					A	C[0]	
3-4				A	C[1]	C[0]	
5-8			A	C[2]	C[1]	C[0]	
9-16		A	C[3]	C[2]	C[1]	C[0]	
17-32	A	C[4]	C[3]	C[2]	C[1]	C[0]	

Где А выбирает одну из структур управляющих данных канала:

- А = 0 выбирает первичную структуру управляющих данных;
- А = 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.

Рисунок 49 демонстрирует пример реализации контроллера с использованием трех каналов DMA и с альтернативной структурой управляющих данных.

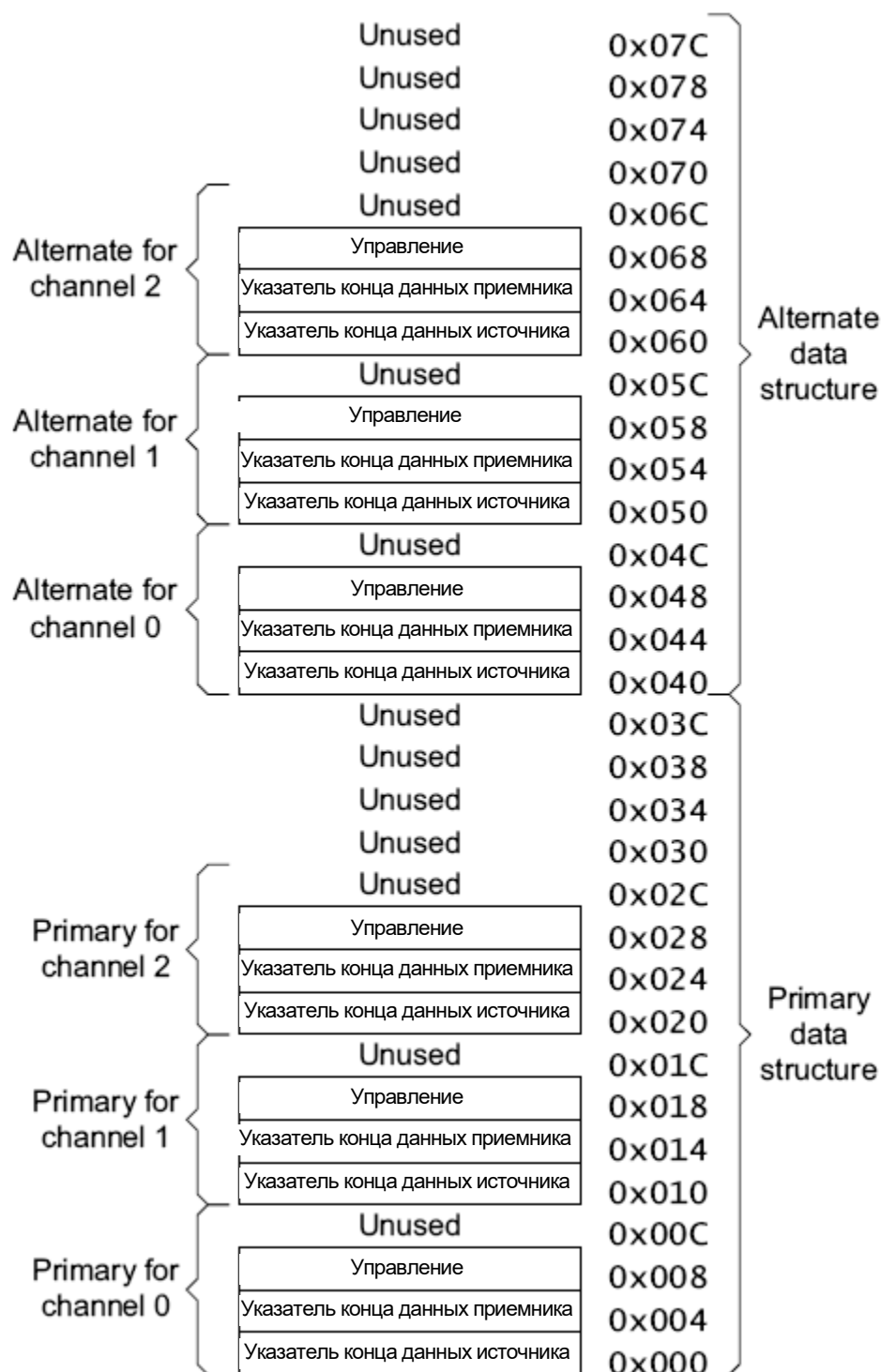


Рисунок 49 – Карта памяти для трех каналов DMA, включая альтернативную структуру

Этот пример структуры управляющих данных использует 128 байт системной памяти. В нем контроллер использует младшие 0x06 разрядов адреса для доступа ко всем элементам структуры управляющих данных. Поэтому базовый адрес структуры должен быть 0xxxxxxx00, далее 0xxxxxxx80.

Таблица 64 перечисляет все разрешенные значения базового адреса для первичной структуры управляющих данных в зависимости от количества каналов DMA, использованных в контроллере.

Т а б л и ц а 64 – Разрешенные базовые адреса

Количество каналов DMA	Разрешенные значения базового адреса для первичной структуры управляющих данных
17-32	0xXXXXXX000, 0xXXXXXX400, 0xXXXXXX800, 0xXXXXXXC00

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.

6.9.5.1 Указатель конца данных источника

Область памяти под названием `src_data_end_ptr` содержит указатель на последний адрес месторасположения данных источника.

Т а б л и ц а 65 – Значения разрядов `src_data_end_ptr`

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	<code>src_data_end_ptr</code>	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.

6.9.5.2 Указатель конца данных приемника

Область памяти под названием `dst_data_end_ptr` содержит указатель на последний адрес месторасположения данных приемника.

Т а б л и ц а 66 – Значения разрядов `dst_data_end_ptr`

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	<code>dst_data_end_ptr</code>	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.

6.9.5.3 Разряды управления

Область памяти под названием `channel_cfg` обеспечивает управление каждой передачей DMA.

Таблица 67 – 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	dst_size	src_inc	src_size	dst_prot_ctrl	Src_prot_ctrl	R_power	n_minus_1	Next_useburst	cycle_ctrl				

Таблица 68 объясняет назначение разрядов этой области памяти.

Таблица 68 – Назначение разрядов channel_cfg

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...30	dst_src	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	Размерность данных приемника Примечание – Значение этого поля должно быть равно значению поля src_size.
27...26	src_inc	Шаг инкремента адреса источника. Шаг инкремента адреса зависит от разрядности данных источника. Разрядность данных источника = байт: 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:

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		b00 = зарезервировано; b01 = зарезервировано; b10 = слово; b11 = нет инкремента. Адрес остается равным значению области памяти <code>src_data_end_ptr</code>
25...24	src_size	Задаёт размерность данных источника: b00 = байт; b01 = полуслово (в русском обычно слово); b10 = слово (в русском обычно двойное слово); b11 = зарезервировано
23...21	dst_prot_ctrl	Задаёт состояние HPROT[3:1], когда контроллер записывает данные в приемник. Разряд 23 управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд 22 управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд 21 управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ непривилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный
20...18	src_prot_ctrl	Задаёт состояние HPROT[3:1], когда контроллер считывает данные из источника. Разряд 20 управляет разрядом HPROT[3]: 0 = HPROT[3] в состоянии 0 и доступ не кэшируется; 1 = HPROT[3] в состоянии 1 и доступ кэшируется. Разряд 19 управляет разрядом HPROT[2]: 0 = HPROT[2] в состоянии 0 и доступ не буферизуется; 1 = HPROT[2] в состоянии 1 и доступ буферизуется. Разряд 18 управляет разрядом HPROT[1]: 0 = HPROT[1] в состоянии 0 и доступ непривилегированный; 1 = HPROT[1] в состоянии 1 и доступ привилегированный
17...14	R_power	Задаёт 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; 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	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.

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		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. 1 – контроллер изменяет значение chnl_useburst_set[C] в состояние 1. Поэтому для оставшихся циклов DMA в режиме работы с периферией «Исполнение с изменением конфигурации», контроллер реагирует только на запросы по dma_req[], при выполнении циклов DMA он использует альтернативные управляющие данные
2...0	cycle_ctrl	Режим работы при выполнении цикла DMA: 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 по соответствующему каналу в течение процедуры арбитража. Это означает, что начального запроса на обработку достаточно для выполнения цикла DMA; b011 Пинг-понг. Контроллер выполняет цикл DMA

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		использую одну из структур управляющих данных. По окончании выполнения цикла DMA, контроллер выполняет следующий цикл DMA, используя другую структуру. Контроллер сигнализирует об окончании каждого цикла DMA, позволяя процессору перенастраивать неактивную структуру данных. Контроллер продолжает выполнять циклы DMA, до тех пор, пока он не прочитает «неправильную» структуру данных или пока процессор не изменит cycle_ctrl поле в состоянии 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.

6.9.5.4 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.
```

Таблица 69 перечисляет адреса приемника цикла DMA для шести слов.

Таблица 69 – Цикл DMA для 6 слов с пословным инкрементом

Начальные значения channel_cfg перед циклом DMA				
src_size=b10, dst_inc=b10, n_minus_1=b101, cycle_ctrl=1				
DMA передачи	Указатель конца данных	Счетчик	Отличие*	Адрес
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения channel_cfg после цикла DMA				
src_size=b10, dst_inc=b10, n_minus_1=0, cycle_ctrl=0				
* – это значение, полученное после сдвига влево значения счетчика на количество разрядов, соответствующее dst_inc				

Таблица 70 перечисляет адреса приемника для передач DMA 12 байт с использованием «полусловного» инкремента.

Таблица 70 – Цикл 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				

6.9.6 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.

Т а б л и ц а 71 – Перечень регистров контроллера

Базовый адрес	Наименование	Тип	Значение по сбросу	Описание
0x40006000	MDR_DMA			Контроллер DMA
Смещение				
0x000	STATUS	RO	0x-0nn0000*	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	0x000000nn**	Регистр базового адреса альтернативных управляющих данных каналов
0x010	WAITONREQ_STATUS	RO	0x00000000	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
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
* Значение по сбросу зависит от количества каналов DMA, использованных в контроллере, а также от того, интегрирована ли схема тестирования. ** Значение по сбросу зависит от количества каналов DMA, использованных в контроллере				

6.9.6.1 Регистр 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. Таблица 73 перечисляет назначение разрядов регистра.

Таблица 72 – Статусный регистр DMA

Номер	31...28	27...21	20...16	15...8	7...4	3...1	0
Доступ	RO	U	RO	U	RO	U	RO
Сброс	0	0	0	0	0	0	0
	test_status	-	chnls_minus1	-	state	-	master_enable

Таблица 73 – Назначение разрядов регистра dma_status

№ 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. Например: b00000 = контроллер имеет 1 канал DMA; b00001 = контроллер имеет 2 канала DMA; b00010 = контроллер имеет 3 канала DMA; ... b11111 = контроллер имеет 32 канала DMA
15...8	-	Не определено
7...4	state	Текущее состояние автомата управления контроллера. Состояние может быть одним из следующих: b0000 = в покое; b0001 = чтение управляющих данных канала; b0010 = чтение указателя конца данных источника; b0011 = чтение указателя конца данных приемника; b0100 = чтение данных источника; b0101 = запись данных в приемник; b0110 = ожидание запроса на выполнение DMA; b0111 = запись управляющих данных канала; b1000 = приостановлен; b1001 = выполнен; b1010 = режим работы с периферией «Исполнение с изменением конфигурации»; b1011-b1111 = не определено
3...1	-	Не определено
0	master_enable	Состояние контроллера: 0 = работа контроллера запрещена; 1 = работа контроллера разрешена

6.9.6.2 Регистр CFG

Регистр CFG является регистром конфигурации контроллера DMA.

Данный регистр имеет доступ только на запись. Регистр определяет состояние контроллера.

Таблица 75 перечисляет назначение разрядов регистра.

Таблица 74 – 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 75 – Назначение разрядов регистра dma_cfg

№ 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

6.9.6.3 Регистр 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. Поэтому количество разрядов регистра, необходимых для задания базового адреса, варьируется и зависит от варианта построения системы.

Если контроллер находится в состоянии сброса, чтение регистра запрещено.

Таблица 77 перечисляет назначение разрядов регистра ctrl_base_ptr.

Т а б л и ц а 76 – Регистр базового адреса управляющих данных каналов

Номер	31...10	9...0
Доступ	R/W	U
Сброс	0	0
	ctrl_base_ptr	-

Т а б л и ц а 77 – Назначение разрядов регистра ctrl_base_ptr

№ 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

6.9.6.4 Регистр 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.

Таблица 79 перечисляет назначение разрядов регистра.

Т а б л и ц а 78 – Регистр базового адреса альтернативных управляющих данных каналов

Номер	31... 0
Доступ	RO
Сброс	0
	Alt_ctrl_base_ptr

Т а б л и ц а 79 – Назначение разрядов регистра alt_ctrl_base_ptr

№ 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

6.9.6.5 Регистр 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.

Таблица 81 перечисляет назначение разрядов регистра.

Т а б л и ц а 80 – Регистр статуса ожидания запроса на обработку каналов

Номер	31		2	1	0
Номер	RO		RO	RO	RO
Доступ	0		0	0	0
	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]

Т а б л и ц а 81 – Назначение разрядов регистра dma_waitonreg_status

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	dma_waitonreg_status	Состояние сигналов ожидания запроса на обработку каналов DMA. При чтении: Разряд [C] =0 означает, что dma_waitonreg[C] в состоянии 0 Разряд [C] =1 означает, что dma_waitonreg[C] в состоянии 1

6.9.6.6 Регистр 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.

Таблица 83 перечисляет назначение разрядов регистра.

Т а б л и ц а 82 – 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]

Т а б л и ц а 83 – Назначение разрядов регистра chnl_sw_request

№ 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;

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		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

6.9.6.7 Регистр 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

Таблица 85 перечисляет назначение разрядов регистра.

Таблица 84 – 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 85 – Назначение разрядов регистра chnl_useburst_set

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_useburst_set	Отключает обработку запросов на выполнение циклов DMA от dma_sreq[] и возвращает при чтении состоянии этих настроек. При чтении: Разряд [C] =0 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные от dma_sreq[] и dma_req[]. Контроллер выполняет одиночные передачи или 2 ^R передач. Разряд [C] =1 означает, что канал DMA C выполняет циклы DMA в ответ на запросы, полученные только от dma_req[]. Контроллер выполняет 2 ^R передач. При записи: Разряд [C] =0 не дает эффекта. Необходимо использовать chnl_useburst_clr регистр и установить соответствующий разряд C в 0; Разряд [C] =1 отключает возможность обрабатывать запросы на выполнение циклов DMA, полученные от dma_sreq[]. Контроллер выполняет 2 ^R передач. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

После выполнения предпоследней передачи из 2^R передач, в том случае, если число оставшихся передач (N) меньше чем 2^R , контроллер сбрасывает разряд `chnl_useburst_set` в 0. Это позволяет выполнять оставшиеся передачи, используя `dma_sreq[]` и `dma_req[]`.

Примечание – При программировании `channel_cfg` значением N меньшим, чем 2^R , запрещена установка соответствующего разряда `chnl_useburst_set` в случае, если периферийный блок не поддерживает сигнал `dma_req[]`.

В режиме работы с периферией «исполнение с изменением конфигурации», если разряд `next_useburst` установлен в `channel_cfg`, контроллер устанавливает `chnl_useburst_set [C]` в 1 после окончания цикла DMA, использующего альтернативные управляющие данные.

6.9.6.8 Регистр 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[]`.

Таблица 87 перечисляет назначение разрядов регистра `chnl_useburst_clr`.

Таблица 86 – 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 87 – Назначение разрядов регистра `chnl_useburst_clr`

№ 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 <code>dma_sreq[]</code> . При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать <code>chnl_useburst_set</code> регистр для отключения обработки запросов от <code>dma_sreq[]</code> ; Разряд [C] = 1 разрешает обрабатывать запросы на выполнение циклов DMA, полученные от <code>dma_sreq[]</code> . Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

6.9.6.9 Регистр 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.

Таблица 89 перечисляет назначение разрядов регистра chnl_req_mask_set.

Таблица 88 – Регистр маскирования запросов на обслуживание каналов

Номер	31		2	1	0
Доступ	R/W		R/W	R/W	R/W
Сброс	0		0	0	0
	chnl_req_mask_set for dma_reg [31] and dma_sreg [31]		chnl_req_mask_set for dma_reg [2] and dma_sreg [2]	chnl_req_mask_set for dma_reg [1] and dma_sreg [1]	chnl_req_mask_set for dma_reg [0] and dma_sreg [0]

Таблица 89 – Назначение разрядов регистра chnl_req_mask_set

№ 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

6.9.6.10 Регистр 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[].

Таблица 91 перечисляет назначение разрядов регистра chnl_req_mask_clr.

Таблица 90 – Регистр очистки маскирования запросов на обслуживание каналов

Номер	31		2	1	0
Доступ	WO		WO	WO	WO
Сброс	0		0	0	0
	chnl_req_mask_clr for dma_reg [31] and dma_sreg [31]		chnl_req_mask_clr for dma_reg [2] and dma_sreg [2]	chnl_req_mask_clr for dma_reg [1] and dma_sreg [1]	chnl_req_mask_clr for dma_reg [0] and dma_sreg [0]

Таблица 91 – Назначение разрядов регистра chnl_req_mask_clr

№ 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

6.9.6.11 Регистр 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.

Таблица 93 перечисляет назначение разрядов регистра chnl_enable_set.

Таблица 92 – 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 93 – Назначение разрядов регистра chnl_enable_set

№ 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

6.9.6.12 Регистр CHNL_ENABLE_CLR

Регистр CHNL_ENABLE_CLR является регистром сброса разрешения каналов. Данный регистр имеет доступ только на запись. Регистр запрещает работу каналов DMA.

Таблица 95 перечисляет назначение разрядов регистра chnl_enable_clr.

Т а б л и ц а 94 – 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

Т а б л и ц а 95 – Назначение разрядов регистра chnl_enable_clr

№ 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 С. 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

6.9.6.13 Регистр 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).

Таблица 97 перечисляет назначение разрядов регистра chnl_pri_alt_set.

Т а б л и ц а 96 – 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]

Т а б л и ц а 97 – Назначение разрядов регистра chnl_pri_alt_set

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_pri_alt_set	Установка соответствующего разряда подключает использование альтернативных управляющих данных для соответствующего канала DMA, чтение возвращает состояние этих настроек. При чтении: Разряд [C] = 0 означает, что канал DMA C использует первичную структуру управляющих данных; Разряд [C] = 1 означает, что канал DMA C использует альтернативную структуру управляющих данных. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_pri_alt_clr регистр для сброса разряда [C] в 0; Разряд [C] = 1 подключает использование альтернативной структуры управляющих данных каналом DMA C. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. Примечание – Контроллер может переключить значение разряда chnl_pri_alt_set[C] в следующих случаях: – при завершении 4-х передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; – при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «Пинг-понг»; – при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: – «пинг-понг»; – работа с памятью «Исполнение с изменением конфигурации»; – работа с периферией «Исполнение с изменением конфигурации»;

6.9.6.14 Регистр 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.

Таблица 99 перечисляет назначение разрядов регистра chnl_pri_alt_clr.

Т а б л и ц а 98 – 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]

Т а б л и ц а 99 – Назначение разрядов регистра chnl_pri_alt_clr

№ 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 С. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта. Примечание – Контроллер может переключить значение разряда chnl_pri_alt_clr[C] в следующих случаях: - при завершении 4-х передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «исполнение с изменением конфигурации»; - при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «пинг-понг»; - при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: - «пинг-понг»; - работа с памятью «Исполнение с изменением конфигурации»; - работа с периферией «Исполнение с изменением конфигурации»

6.9.6.15 Регистр 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.

Таблица 101 перечисляет назначение разрядов регистра chnl_priority_set.

Т а б л и ц а 100 – Регистр установки приоритета каналов

Номер	31		2	1	0
Доступ	R/W		R/W	R/W	R/W
Сброс	0		0	0	0
	chnl_prioit_set for channel [31]		chnl_priority_set for channel [2]	chnl_priority_set for channel [1]	chnl_priority_set for channel [0]

Т а б л и ц а 101 – Назначение разрядов регистра chnl_priority_set

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...0	chnl_priority_set	Установка высокого приоритета каналу DMA, чтение возвращает состояние приоритета каналов DMA. При чтении: Разряд [C] = 0 означает, что каналу DMA C присвоен уровень приоритета по умолчанию; Разряд [C] = 1 означает, что каналу DMA C присвоен высокий уровень приоритета. При записи: Разряд [C] = 0 не дает эффекта. Необходимо использовать chnl_priority_clr регистр для установки каналу C уровня приоритета по умолчанию; Разряд [C] = 1 устанавливает каналу DMA C высокий уровень приоритета. Запись разряда, соответствующего нереализованному каналу, не дает никакого эффекта

6.9.6.16 Регистр 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.

Таблица 103 перечисляет назначение разрядов регистра chnl_priority_clr.

Таблица 102 – 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 103 – Назначение разрядов регистра chnl_priority_clr

№ 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

6.9.6.17 Регистр 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.

Таблица 105 перечисляет назначение разрядов регистра err_clr.

Таблица 104 – Регистр сброса флага ошибки

Номер	31...1	0
Доступ	U	R/W
Сброс	0	0
	-	err_clr

Таблица 105 – Назначение разрядов регистра err_clr

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...1	-	Не определено. Следует записывать 0
0	chnl_priority_set	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. Для целей тестирования можно использовать регистр err_set, чтобы установить сигнал dma_err в состояние 1. Примечание – При сбросе сигнала dma_err одновременно с появлением ошибки на шине АНВ-Lite, то приоритет отдается ошибке, и, следовательно, значение регистра (и dma_err) останется неизменным (несброшенным)

6.9.6.18 Регистр CHMUX[x]

Base ADDR=	0x40006000	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
CHANEL[x*4+3]								CHANEL[x*4+2]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANEL[x*4+1]								CHANEL[x*4+0]							

Бит	Имя	Значение	Описание
31...24	CHANEL[x*4+3]	0	Номер реального канала запроса DMA назначенного на виртуальный канал x*4+3
23...16	CHANEL[x*4+2]	0	Номер реального канала запроса DMA назначенного на виртуальный канал x*4+2
15...8	CHANEL[x*4+1]	0	Номер реального канала запроса DMA назначенного на виртуальный канал x*4+1
7...0	CHANEL[x*4+0]	0	Номер реального канала запроса DMA назначенного на виртуальный канал x*4+0

Таблица 106 – Виртуальные номера запросов DMA

Номера реальных каналов DMA	Тип SREQ	Тип REQ	Описание
0			
1			
2			
3			
4	SPW0_RX_SREQ	SPW0_RX_BREQ	Запрос приемника SPW0
5	SPW1_RX_SREQ	SPW1_RX_BREQ	Запрос приемника SPW1
6	SPW0_TX_SREQ	SPW0_TX_BREQ	Запрос передатчика SPW0
7	SPW1_TX_SREQ	SPW1_TX_BREQ	Запрос передатчика SPW1
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			
13			
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			
19			
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			
25			
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	-	-	
31	-	-	
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	-	-	
37	-	-	
38	SSP0_TX_SREQ	SSP0_TX_BREQ	Запрос передатчика SSP0
39	SSP1_TX_SREQ	SSP1_TX_BREQ	Запрос передатчика SSP1
40	SSP2_TX_SREQ	SSP2_TX_BREQ	Запрос передатчика SSP2
41	SSP3_TX_SREQ	SSP3_TX_BREQ	Запрос передатчика SSP3
42	SSP4_TX_SREQ	SSP4_TX_BREQ	Запрос передатчика SSP4
43	SSP5_TX_SREQ	SSP5_TX_BREQ	Запрос передатчика SSP5
44	SSP0_RX_SREQ	SSP0_RX_BREQ	Запрос приемника SSP0
45	SSP1_RX_SREQ	SSP1_RX_BREQ	Запрос приемника SSP1
46	SSP2_RX_SREQ	SSP2_RX_BREQ	Запрос приемника SSP2
47	SSP3_RX_SREQ	SSP3_RX_BREQ	Запрос приемника SSP3
48	SSP4_RX_SREQ	SSP4_RX_BREQ	Запрос приемника SSP4
49	SSP5_RX_SREQ	SSP5_RX_BREQ	Запрос приемника SSP5
50	UART0_TX_SREQ	UART0_TX_BREQ	Запрос передатчика UART0
51	UART1_TX_SREQ	UART1_TX_BREQ	Запрос передатчика UART1

Номера реальных каналов DMA	Тип SREQ	Тип REQ	Описание
52			
53			
54	UART0_RX_SREQ	UART0_RX_BREQ	Запрос приемника UART0
55	UART1_RX_SREQ	UART1_RX_BREQ	Запрос приемника UART1
56	-	-	
57	-	-	
58	CRC_SREQ	-	Запрос REQ от CRC

6.10 Контроллер портов ввода-вывода (PORTx)

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, а необходимо осуществить запись 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” и это сбросит соответствующий бит. 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 2-ух тактов ядра. Схема формирования запроса прерывания представлена на рисунке 50.

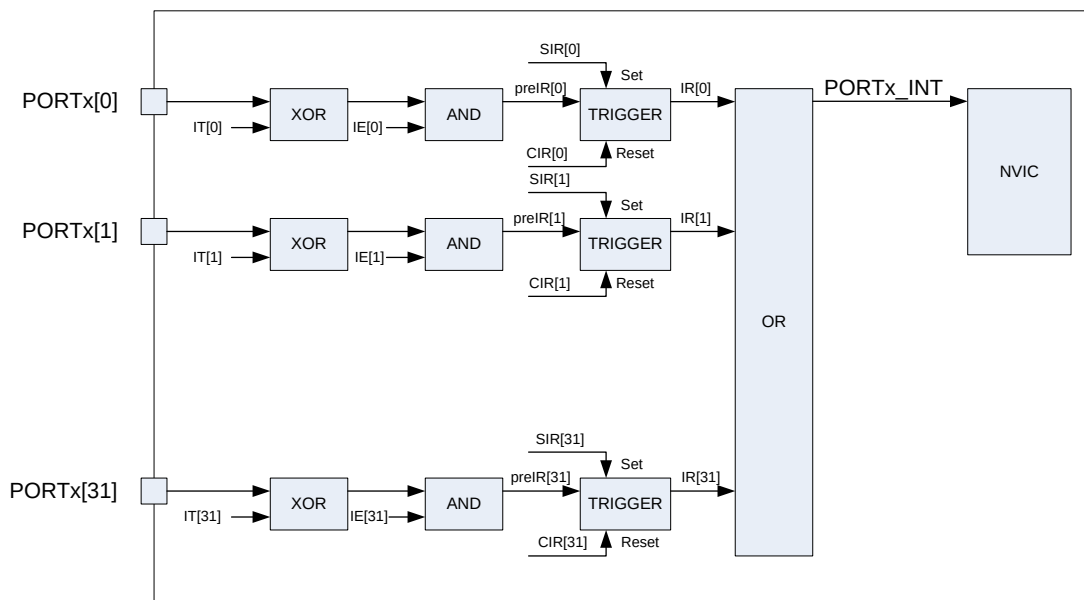


Рисунок 50 – Схема формирования запроса прерывания

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. Формирования запроса прерывания при отключенном тактировании, либо при нахождении микроконтроллера в режиме SLEEP возможно только на выводах, настроенных на специальную функцию EXT_INT.

6.10.1 Описание регистров

Таблица 107 – Описание регистров контроллера портов ввода-вывода

Базовый адрес	Смещение	Обозначение	Состояние после сброса	Описание
0x40080000				
0x40081000				
0x40082000				
0x40083000				
0x40084000				
	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
	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 фильтра 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

6.10.1.1 Регистр KEY

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000					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]															
Бит	Имя		Значение				Описание								
31...0	KEY[31:0]		0000_0000				При записи в регистр значения 0x8555AAA1 открывается возможность записи в другие регистры блока								

6.10.1.2 Регистр RXTX

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000					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]															
Бит	Имя		Значение				Описание								
31...0	RXTX						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 чтения-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 быть 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								

Бит	Имя	Значение	Описание
			прерывания сразу после выполнения чтения. В другой задаче либо обработке прерывания изменяется значение в выходных триггерах и осуществляется возврат в текущую задачу, либо выход из обработчика. И выполняется вторая часть модификация и запись, но при этом эти действия выполняются над данными полученными до ухода и по их завершению могут быть изменены данные записанные в другой задаче, либо обработке прерываний. Для избегания такого рода проблем рекомендуется использовать биты установки и сброса значений, например, SRXTX и CRXTX и другие аналогичные

6.10.1.3 Регистр SRXTX

Base ADDR=	0x40080000 0x40081000 0x40082000 0x40083000 0x40084000	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]															

Бит	Имя	Значение	Описание
31...0	SRXTX		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

6.10.1.4 Регистр CRXTX

Base ADDR=		0x40080000				Offset=		0x0000_000C								
		0x40081000														
		0x40082000														
		0x40083000														
		0x40084000														
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]																

Бит	Имя	Значение	Описание
31...0	CRXTX		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

6.10.1.5 Регистр SOE

Base ADDR=		0x40080000				Offset=		0x0000_0010								
		0x40081000														
		0x40082000														
		0x40083000														
		0x40084000														
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]																

Бит	Имя	Значение	Описание
-----	-----	----------	----------

31...0	SOE[31:0]		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
--------	-----------	--	--

6.10.1.6 Регистр COE

Base ADDR=	0x40080000 0x40081000 0x40082000 0x40083000 0x40084000	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]															

Бит	Имя	Значение	Описание
31...0	COE[31:0]		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.

6.10.1.7 Регистр SFUNCx

Base ADDR=	0x40080000	Offset=	0x0000_0018												
	0x40081000		0x0000_001C												
	0x40082000		0x0000_0020												
	0x40083000		0x0000_0024												
	0x40084000														
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]			

Бит	Имя	Значение	Описание
31...28	FUNC7[3:0]		Функция, выполняемая выводом
27...24	FUNC6[3:0]		При записи 1 производится установка в 1 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.
23...20	FUNC5[3:0]		0000 – пользовательский вывод
19...16	FUNC4[3:0]		0001 – Функция 1
15...12	FUNC3[3:0]		...
11...8	FUNC2[3:0]		1111 – Функция 15
7...4	FUNC1[3:0]		Конкретное назначение функции смотрите в описании выводов микросхемы.
3...0	FUNC0[3:0]		Выбор аналоговой функции приоритета. 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

6.10.1.8 Регистр CFUNCx

Base ADDR=	0x40080000	Offset=	0x0000_0028												
	0x40081000		0x0000_002C												
	0x40082000		0x0000_0030												
	0x40083000		0x0000_0034												
	0x40084000														
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]			

Бит	Имя	Значение	Описание
31...28	FUNC7[3:0]		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
27...24	FUNC6[3:0]		
23...20	FUNC5[3:0]		
19...16	FUNC4[3:0]		
15...12	FUNC3[3:0]		
11...8	FUNC2[3:0]		
7...4	FUNC1[3:0]		
3...0	FUNC0[3:0]		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 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

6.10.1.9 Регистр SANALOG

Base ADDR=		0x40080000				Offset=		0x0000_0038								
		0x40081000														
		0x40082000														
		0x40083000														
		0x40084000														
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]																

Бит	Имя	Значение	Описание
31...0	SANALOG[31:0]		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

6.10.1.10 Регистр CANALOG

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000					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]															

Бит	Имя	Значение	Описание
31...0	CANALOG[31:0]		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.

6.10.1.11 Регистр SPULLUP

Base ADDR=	0x40080000	Offset=	0x0000_0040												
	0x40081000														
	0x40082000														
	0x40083000														
	0x40084000														
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]															

Бит	Имя	Значение	Описание
31...0	SPULLUP [31:0]		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

6.10.1.12 Регистр CPULLUP

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	CPULLUP [31:0]		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

6.10.1.13 Регистр SPULLDOWN

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	SPULLDOWN [31:0]		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

6.10.1.14 Регистр CPULLDOWN

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	CPULLDOWN [31:0]		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

6.10.1.15 Регистр SPD

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	SPD [31:0]		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 и низким уровнем (открытый сток)

6.10.1.16 Регистр CPD

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	CPD [31:0]		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 и низким уровнем (открытый сток)

6.10.1.17 Регистр SPWRx

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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	

Бит	Имя	Значение	Описание
31...30	PWR15[1:0]		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
29...28	PWR14[1:0]		
27...26	PWR13[1:0]		
25...24	PWR12[1:0]		
23...22	PWR11[1:0]		
21...20	PWR10[1:0]		
19...18	PWR9[1:0]		
17...16	PWR8[1:0]		
15...14	PWR7[1:0]		
13...12	PWR6[1:0]		
11...10	PWR5[1:0]		
9...8	PWR4[1:0]		
7...6	PWR3[1:0]		
5...4	PWR2[1:0]		
3...2	PWR1[1:0]		
1...0	PWR0[1:0]		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

6.10.1.18 Регистр CPWRx

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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	

Бит	Имя	Значение	Описание
31...30	PWR0[1:0]		Регистр сброса скорости драйвера При записи 1 производится установка в 0 сигнала настройки порта, при записи 0, значение выходного триггера не изменяется.
29...28	PWR0[1:0]		
27...26	PWR0[1:0]		
25...24	PWR0[1:0]		00– драйвер выключен
23...22	PWR0[1:0]		01 – медленный фронт 300 нс
21...20	PWR0[1:0]		10 – средний фронт 100 нс
19...18	PWR0[1:0]		11 – быстрый фронт 10 нс
17...16	PWR0[1:0]		
15...14	PWR0[1:0]		CPWR0[1:0] – вывод 0
13...12	PWR0[1:0]		CPWR0[3:2] – вывод 1
11...10	PWR0[1:0]		CPWR0[5:4] – вывод 2
9...8	PWR0[1:0]		CPWR0[7:6] – вывод 3
7...6	PWR0[1:0]		CPWR0[9:8] – вывод 4
5...4	PWR0[1:0]		CPWR0[11:10] – вывод 5
3...2	PWR0[1:0]		CPWR0[13:12] – вывод 6
1...0	PWR0[1:0]		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
			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

6.10.1.19 Регистр SCL

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	CL[31:0]		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.

6.10.1.20 Регистр CCL

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	CL[31:0]		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

6.10.1.21 Регистр SIE

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	IE[31:0]		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.

6.10.1.22 Регистр CIE

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	IE[31:0]		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

6.10.1.23 Регистр SIT

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	IT[31:0]		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

6.10.1.24 Регистр CIT

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	IT[31:0]		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

6.10.1.25 Регистр SIR

Base ADDR=	0x40080000 0x40081000 0x40082000 0x40083000 0x40084000	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]															

Бит	Имя	Значение	Описание
31...0	IR[31:0]		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

6.10.1.26 Регистр CIR

Base ADDR=		0x40080000				Offset=		0x0000_0084									
		0x40081000															
		0x40082000															
		0x40083000															
		0x40084000															
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]															

Бит	Имя	Значение	Описание
31...0	IR[31:0]		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

6.10.1.27 Регистр HCUR

Base ADDR=		0x40080000 0x40081000 0x40082000 0x40083000 0x40084000				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]															

Бит	Имя	Значение	Описание
31...0	HCUR[31:0]		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

6.11 Контроллер внешней шины

6.11.1 Описание регистров

Таблица 108 – Описание регистров контроллера внешней шины

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x40005000				
	0x0000_0000	REGION[0].CNTRL		Настройки региона 0
	0x0000_0004	REGION[0].ECCBASE		Базовый адрес таблицы ECC региона 0
	0x0000_0008	REGION[0].ECCCR		Регистр флагов и управления счётчиками ошибок ECC региона 0
	0x0000_000C	REGION[0].ECCST		Регистр счётчиков ошибок ECC региона 0
	...			
	0x0000_0070	REGION[7].CNTRL		Настройки региона 7
	0x0000_0074	REGION[7].ECCBASE		Базовый адрес таблицы ECC региона 7
	0x0000_0078	REGION[7].ECCCR		Регистр флагов и управления счётчиками ошибок ECC региона 7
	0x0000_007C	REGION[7].ECCST		Регистр статуса ошибок ECC региона 7
	0x0000_0080	KEY		Регистр ключа, разрешающего модификацию остальных регистров
	0x0000_0084	ECCADR		Регистр адреса последней ошибки ECC
	0x0000_0088	ECCDATA		Регистр данных последней ошибки ECC
	0x0000_008C	ECCECC		Регистр ECC последней ошибки ECC
	0x0000_0090	OCLKCTL		Регистр управления частотой OCLK

6.11.1.1 Регистр KEY

Base ADDR=	0x40005000	Offset=	0x0000_0080												
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]															

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

6.11.1.2 Регистр REGION[n].CNTRL

Base ADDR=	0x40005000	Offset=	0x0000_0000 0x0000_0010 0x0000_0020 0x0000_0030 0x0000_0040 0x0000_0050 0x0000_0060 0x0000_0070												
REG Name:															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-							ECC8BIT	WS_HOLD[2:0]				WS_SETUP[2:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WS_ACTIVE[7:0]								CPOL	ROM	MODE[1:0]		RDYWAIT	ECCMODE	ECCEN	EN

Бит	Имя	Значение	Описание
31...25	-	0	Зарезервировано
24	ECC8BIT		Режим работы ECC 1 – 8 бит ECC 0 – 16 бит ECC
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-х разрядная шина данных (+8 разрядов при ECCMODE=0 и ECCEN=1) 01 – 16-ти разрядная шина данных (при ECCEN=0 или при ECCMODE=1и ECCEN=1) 10 – 8-ми разрядная шина данных (при ECCEN=0 или при ECCMODE=1и ECCEN=1) 11 – 32-х разрядная шина данных (+16 разрядов при ECCMODE=0 и ECCEN=1)

Бит	Имя	Значение	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 больше чем задана битами WS_ACTIVE и завершается при наличии высокого уровня на линии READY (Необходимо учитывать два такта пересинхронизации сигнала READY на внутренний синхросигнал)
2	ECCMODE		Режим ECC 0 – параллельная организация ECC (проверочные биты расположены в разрядах шины данных DATA[39:32] или DATA[47:32] в зависимости от бита ECC8BIT) 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 не работающего региона транзакция завершается с флагом ошибка на шине

Примечание – Код программы, изменяющий настройки регионов внешней системной шины, должен выполняться из области внутренней памяти RAMC или RAMD. 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.

6.11.1.3 Регистр REGION[n].ECCBASE

Base ADDR=	0x40005000	Offset=	0x0000_0004 0x0000_0014 0x0000_0024 0x0000_0034 0x0000_0044 0x0000_0054 0x0000_0064 0x0000_0074												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCBASE[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCBASE[15:0]															

Бит	Имя	Значение	Описание
31...0	ECCBASE[31:0]		Базовый адрес расположения таблицы ECC в регионе, при ECCMODE=1 и ECCEN=1, иначе не имеет значения

6.11.1.4 Регистр REGION[n].ECCCR

Base ADDR=	0x40005000	Offset=	0x0000_0008 0x0000_0018 0x0000_0028 0x0000_0038 0x0000_0048 0x0000_0058 0x0000_0068 0x0000_0078												
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_TCNT	FIX_TECC	TECC_IE	TECC	CLR_DCNT	CLR_SCNT	FIX_DECC	FIX_SECC	DECC_IE	SECC_IE	DECC	SECC

Бит	Имя	Значение	Описание
31...12			Зарезервировано
11	CLR_TCNT		Сброс счётчика тройных ошибок Запись 1 сбрасывает счётчик TECC_CNT
10	FIX_TECC		Разрешение фиксации в регистрах ECCADR, ECCDATA, ECSECC информации о последнем обращении с тройной ошибкой 1 – разрешено 0 – запрещено
9	TECC_IE		Разрешение прерывания при возникновении тройной ошибки ECC 1 – разрешено 0 – запрещено
8	TECC		Флаг тройной ошибки ECC 1 – ошибка была 0 – ошибок не было Сбрасывается записью 1
7	CLR_DCNT		Бит сброса счетчика двойных ошибок Запись 1 сбрасывает счетчик DECC_CNT Всегда читается как 0
6	CLR_SCNT		Бит сброса счетчика одиночных 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 1 – разрешено 0 – запрещено
4	FIX_SECC		Бит разрешения фиксации в регистрах адреса и данных адреса последней ошибки при одинарной ошибке 1 – разрешено 0 – запрещено
3	DECC_IE		Бит разрешения прерывания при возникновении двойной ошибки ECC 0 – прерывание запрещено 1 – прерывание разрешено

Бит	Имя	Значение	Описание
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

6.11.1.5 Регистр REGION[n].ECCST

Base ADDR=	0x40005000	Offset=	0x0000_000C 0x0000_001C 0x0000_002C 0x0000_003C 0x0000_004C 0x0000_005C 0x0000_006C 0x0000_007C												
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]								TECC_CNT[7:0]							

Бит	Имя	Значение	Описание
31...16	SECC_CNT[15:0]		Счетчик числа одиночных ошибок 0 – нет ошибок 1 – одна ошибка ... 65535 – 65535 ошибок
15:8	DECC_CNT[7:0]		Счетчик числа двойных ошибок 0 – нет ошибок 1 – одна ошибка ... 255 – 255 ошибок
7:0	TECC_CNT[7:0]		Счётчик числа тройных ошибок 0 – нет ошибок 255 – 255 ошибок

6.11.1.6 Регистр ECCADR

Base ADDR=	0x40005000	Offset=	0x0000_0084												
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]															

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

6.11.1.7 Регистр ECCDATA

Base ADDR=	0x40005000	Offset=	0x0000_0088												
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]															

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

6.11.1.8 Регистр ECCECC

Base ADDR=	0x40005000	Offset=	0x0000_008C												
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[15:8]								ECC[7:0]							

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0	ECC[15:0]		Считанные ECC биты при последней тройной, двойной или одинарной ошибке, без корректировки ECC. Предварительно необходимо разрешить фиксацию данных значений в регистре REGION[n].ECCCR, биты FIX_SECC, FIX_DECC, FIX_TECC

6.11.1.9 Регистр OCLKCTL

Base ADDR=	0x40005000	Offset=	0x0000_0090												
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
											ENOCLK				
												DIVOCLK[3:0]			

Бит	Имя	Значение	Описание
4	ENOCLK		Бит разрешения выдачи синхросигнала OCLK 0 – OCLK не формируется 1 – OCLK формируется
3...0	DIVOCLK[3:0]		Биты задания коэффициент деления частоты OCLK $FOCLK = HCLK / (2^{DIVOCLK+1})$

6.12 Контроллер SpaceWire (SpaceWire)

6.12.1 Описание регистров контроллера интерфейса SpaceWire

Базовый Адрес	Название	Описание
0x40088000	SPW0	Контроллер интерфейса SpaceWire
0x40089000	SPW1	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 приёмника
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_CTRL	Регистр управления аналоговым приемопередатчиком
0x0000_0040-0x0000_007C	RXDESC0-SPW_RXDESC15	Память дескрипторов приёмника
0x0000_0080-0x0000_00BC	TXDESC0-SPW_TXDESC15	Память дескрипторов передатчика

6.12.1.1 Регистр CONTROL

Base ADDR=		0x40088000 0x40089000				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	INTENTXDES	INTENRXDES	INTETXF	INTETXE	INTERXF	INTENRXNE	DMARXEN	DMATXEN	INTENLINK	INTENERR	INTENTIME	AUTOSTART	LINKSTART	LINKDIS	SPWEN

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 завершению передачи пакета данных с признаком конца пакета: 0 – прерывание запрещено 1 – прерывание разрешено
13	INTENRXDES		Разрешение прерывания по завершению приёма 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 приёмника: 0 – прерывание запрещено 1 – прерывание разрешено
9	INTENRXNE		Разрешение прерывание при наличии в FIFO приёмника хотя бы одного 32-разрядного слова данных: 0 – прерывание запрещено 1 – прерывание разрешено
8	DMARXEN		Разрешение запроса DMA при заполнении FIFO приёмника: 0 –запрос запрещён 1 – запрос разрешён
7	DMATXEN		Разрешение запроса DMA при опустошении FIFO передатчика: 0 –запрос запрещён 1 – запрос разрешён
6	INTENLINK		Разрешение прерывания при изменении статуса соединения (соединение установлено/соединение разорвано): 0 – прерывание запрещено 1 – прерывание разрешено
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 приёме маркера времени: 0 – прерывание запрещено 1 – прерывание разрешено
3	AUTOSTART		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. В отличие от LinkStart при AutoStart не происходит перехода по таймеру из состояния Started в состояние ErrorReset. Блок Linkinterface переходит из состояния Started в состояние Connecting при получении символа NULL, или переходит в состояние ErrorReset при получении какого-либо другого символа

Бит	Имя	Значение	Описание
2	LINKSTART		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

6.12.1.2 Регистр STATUS

Base ADDR=	0x40088000	Offset=	0x0000_0004												
REG Name:	0x40089000														
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	

Бит	Имя	Значение	Описание
31...15			Зарезервировано
14	RXF34		Флаг заполненности 3/4 FIFO приёмника: 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 приёма пакета данных с признаком конца пакета: 0 – приём пакета не завершён 1 – приём пакета завершён 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

Бит	Имя	Значение	Описание
9	RXF		Флаг полной заполненности FIFO приёмника: 0 – FIFO не полно; 1 – FIFO полно
8	RXNE		Флаг наличия данных в FIFO приёмника: 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
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		Состояние блока Linkinterface (согласно Figure 8-2 стандарта ECSS-E-ST-50-12C): 000 – Error Reset (исходное состояние); 001 – Error Wait; 010 – Ready; 011 – Started; 100 – Connecting; 101 – Run

6.12.1.3 Регистр DIV

Base ADDR=		0x40088000 0x40089000				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							

Бит	Имя	Значение	Описание
31...16			Зарезервировано
15...8	DIVCNTDEF		Делитель для задания скорости передачи данных в канал связи SpceWire равной 10Мбит/с $\text{скорость передачи} = \text{txclk} / \text{DIVCNTDEF}$, DIVCNTDEF не равно 0 где txclk- частота тактирования передатчика. Частота txclk должна быть синхронна по отношению к частоте контроллера и не должна превышать её более чем в два раза, а также не может быть меньше неё. 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 $\text{скорость передачи} = \text{txclk} / \text{DIVCNT}$, DIVCNT не равно 0 где txclk- частота тактирования передатчика. Частота txclk должна быть синхронна по отношению к частоте контроллера и не должна превышать её более чем в два раза, а также не может быть меньше неё

6.12.1.4 Регистр TIME

Base ADDR=	0x40088000	Offset=	0x0000_000C												
REG Name:	0x40089000														
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						

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

6.12.1.5 Регистр PAUSE

Base ADDR=		0x40088000 0x40089000				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							

Бит	Имя	Значение	Описание
31...19			Зарезервировано
18...8	RESTIME		Количество тактов частоты контроллера необходимое на интервал работы счетчика 6,4 мкс
7...0	DSCTIME		Количество тактов частоты контроллера необходимое на интервал работы счетчика 850 нс

6.12.1.6 Регистр FIFORX

Base ADDR=		0x40088000 0x40089000				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]															

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-х разрядных слов путём добавления “лишних” байт. FIFO размером 16 слов

6.12.1.7 Регистр FIFOTX

Base ADDR=	0x40088000	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]															

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

6.12.1.8 Регистр CNTRX_PACK

Base ADDR=	0x40088000	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															

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

6.12.1.9 Регистр CNTRX0_PACK

Base ADDR=		0x40088000 0x40089000				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																	

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

6.12.1.10 Регистр NUM_TXDESC

Base ADDR=		0x40088000 0x40089000				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	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

6.12.1.11 Регистр NUM_RXDESC

Base ADDR=	0x40088000 0x40089000					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			

Бит	Имя	Значение	Описание
31...4	-		Зарезервировано
3...0	NUM_RXDESC		Дескрипторы приёмника SPW_RXDESC0-SPW_RXDESC15 заполняются в памяти, начиная с номера 0 и заканчивая номером 15, после этого процесс повторяется

6.12.1.12 Регистр PHY_CNTR

Base ADDR=	0x40088000 0x40089000					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]	-	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-						EN_BNG	TRIM[2:0]			-	OUT_EN	-	EN_PHY	IR_UP	IR_DOWN

Бит	Имя	Значение	Описание
31...21	-	0	Зарезервировано
20...18	TST[2:0]	000	Вход задания тестового режима 000 – 001 – ... 111 –
17...10	-	0	Зарезервировано
9	EN_BNG	0	Сигнал разрешения работы блока REF для SPW 0 – блок выключен 1 – блок включен
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)

Бит	Имя	Значение	Описание
5	-	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 %

6.12.1.13 Регистр RXDESCx

Base ADDR=	0x40088000 0x40089000	Offset=	0x0000_0040 .. 0x0000_007C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXDES	RXDES														
RW0	R0	R0	R0												

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

Бит	Имя	Значение	Описание
31...0	RXDES		Дескриптор пакета имеет следующую структуру: · бит 31 – признак заполнения дескриптора действительными данными. · До запуска приёма, все 31-е биты дескрипторов области приёма 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

6.12.1.14 Регистр TXDESCx

Base ADDR=	0x40088000 0x40089000	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															

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-хразрядных 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

6.13 Контроллер таймеров общего назначения (TIMER32x)

Все блоки таймеров выполнены на основе 32-битного перезагружаемого счетчика, который синхронизируется с выхода 16-битного предделителя. Перезагружаемое значение хранится в отдельном регистре. Счет может быть прямой, обратный или двунаправленный (сначала прямой до определенного значения, а затем обратный).

Каждый из четырех таймеров микроконтроллера содержит 32-битный счетчик, 16-битный предделитель частоты и 4-канальный блок захвата/сравнения. Их можно синхронизировать системной синхронизацией, внешними сигналами или другими таймерами.

Помимо составляющего основу таймера счетчика, в каждый блок таймера также входит четырехканальный блок захвата/сравнения. Данный блок выполняет, как стандартные функции захвата и сравнения, так и ряд специальных функций. Таймеры имеют в своем составе четыре канала схем захвата, ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Каждый из таймеров может генерировать прерывания и запросы DMA.

Особенности:

- 32-битный вверх, вниз, вверх/вниз счетчик;
- 16-разрядный программируемый предварительный делитель частоты;
- до четырех независимых 32-битных каналов захвата на один таймер. Каждый из каналов захвата может захватить (скопировать) текущее значение таймера при изменении некоторого входного сигнала. В случае захвата имеется дополнительная возможность генерировать прерывание и/или запрос DMA;
- четыре 32-битных регистра сравнения (совпадения), которые позволяют осуществлять непрерывное сравнение, с дополнительной возможностью генерировать прерывание и/или запрос DMA при совпадении;
- имеется до четырёх внешних выводов, соответствующих регистрам совпадения со следующими возможностями:
 - 2 сброс в НИЗКИЙ уровень при совпадении;
 - 3 установка в ВЫСОКИЙ уровень при совпадении;
 - 4 переключение (инвертирование) при совпадении;
 - 5 при совпадении состояние выхода не изменяется;
 - 6 переключение при некотором условии.

6.13.1 Функционирование

Таймер предназначен для того, чтобы подсчитывать циклы периферийной тактовой частоты F_{dts} или какие-либо внешние события и произвольно генерировать прерывания, запросы DMA или выполнять другие действия. Значения таймера, при достижении которых будут выполнены те или иные действия, задаются восьмью регистрами совпадения. Кроме того, в микроконтроллере имеются четыре входа захвата, чтобы захватить значение таймера при изменении некоторого входного сигнала, с возможностью генерировать прерывание или запрос DMA.

Структурная схема блока таймера представлена на рисунке 51.

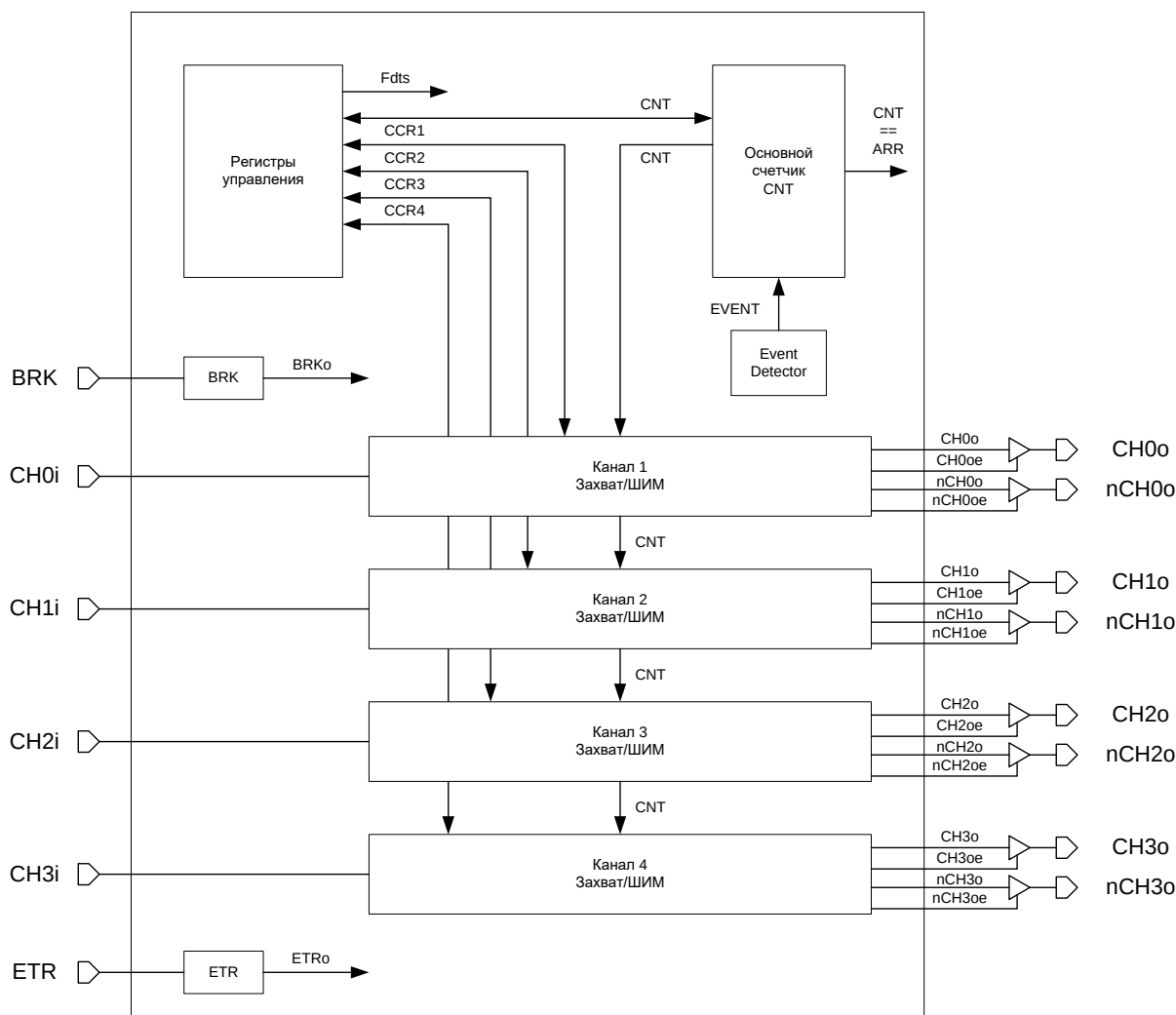


Рисунок 51 – Структурная схема таймера

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.

6.13.2 Инициализация таймера

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

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 14 для таймера 1, бит 15 для таймера 2, бит 16 для таймера 3, бит 19 для таймера 4 регистра PER_CLOCK). В регистре TIM_CLOCK установить бит TIM_CLKEN, чтобы разрешить тактирование определенного таймера, задать коэффициент деления тактовой частоты HCLK для каждого таймера.

После подачи тактового сигнала на блок таймера можно приступить к работе с ним.

6.13.3 Режим таймера

Таймеры построены на базе 32-битного счетчика, объединенного с 16-битным предварительным делителем. Скорость счета таймера зависит от значения, находящегося в регистре делителя.

Счетчик может считать вверх, вниз или сначала вверх и затем вниз.

Базовый блок таймера включает в себя:

- основной счетчик таймера (TIMx_CNT);
- делитель частоты при счете основного счетчика (TIMx_PSC);
- основание счета основного счетчика (TIMx_ARR).
- Сигналом для изменения CNT может служить как внутренняя частота TIM_CLK, так и события в других счетчиках, либо события на линиях TxCH0 данного счетчика.

Чтобы запустить работу основного счетчика необходимо задать:

- начальное значение основного счетчика таймера – TIMx_CNT;
- значение предварительного делителя счетчика – TIMx_PSG, при этом основной счетчик будет считать на частоте $CLK = TIMx_CLK / (PSG + 1)$;
- значение основания счета для основного счетчика TIMx_ARR;
- режим работы счетчика TIMx_CNTRL:
 - выбрать источник события переключения счетчика EVENT_SEL;
- режим счета основного счетчика CNT_MODE (значения 00 и 01 при тактировании внутренней частотой, значения 10 и 11 при тактировании внешними сигналами);
- направление счета основного счетчика DIR;
- разрешить работу счетчика CNT_EN.

По событиям совпадения значения основного счетчика со значением нуля или значением основания счета генерируется прерывание и запроса DMA, которые могут быть замаскированы.

6.13.4 Режимы счета

Счет вверх: CNT_MODE = 00, DIR = 0

TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика

TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика

TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx_ARR = 0x00000013; //Основание счета

//Разрешение работы таймера.

TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK.

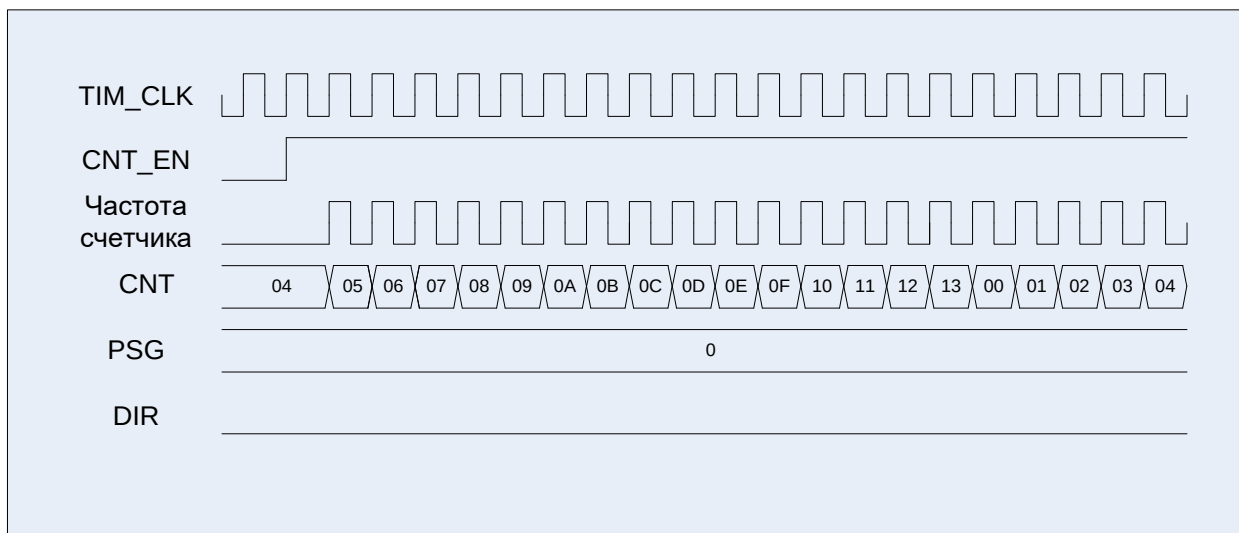


Рисунок 52 – Диаграммы работы таймера, счет вверх от 0 до 0x0013, стартовое значение 0x0004

Счет вниз: CNT_MODE = 00, DIR = 1

TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика

TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика

TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx_ARR = 0x00000013; //Основание счета

//Разрешение работы таймера.

TIMx->TIMx_CNTRL = 0x00000009; //Счет вниз по TIM_CLK.

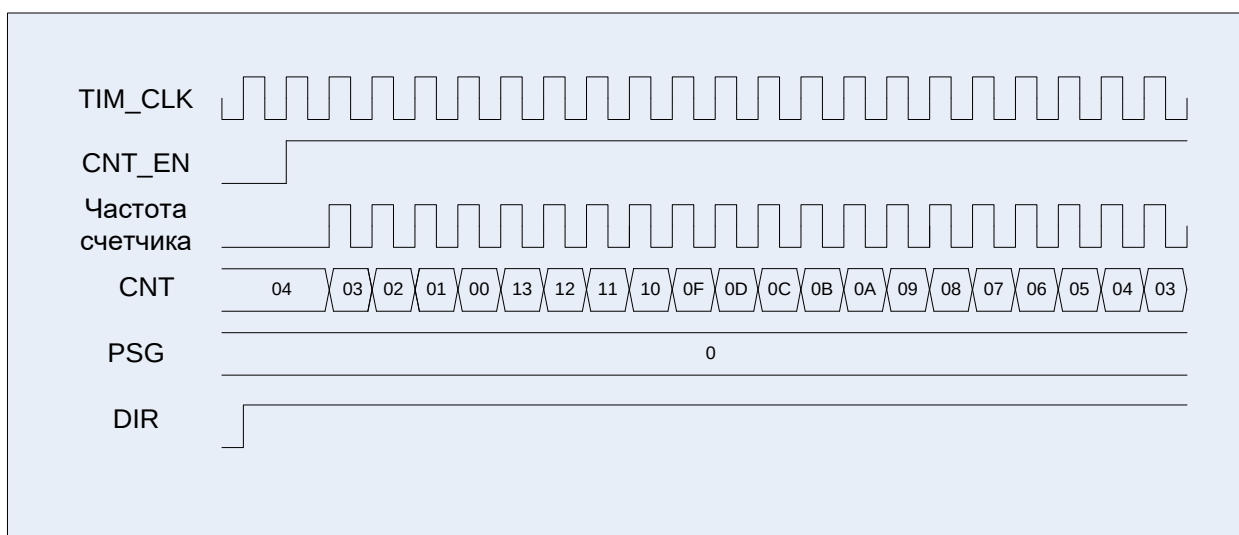


Рисунок 53 – Диаграммы работы таймера, счет вниз от 0x0013 до 0, стартовое значение 0x0004

Счет вверх/вниз: CNT_MODE = 01, DIR = 0

TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера


```
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
TIMx->TIMx_ARR = 0x00000013; //Основание счета
```

```
//Разрешение работы таймера.
TIMx->TIMx_CNTRL = 0x00000041; //Счет вверх/вниз по TIM_CLK.
```

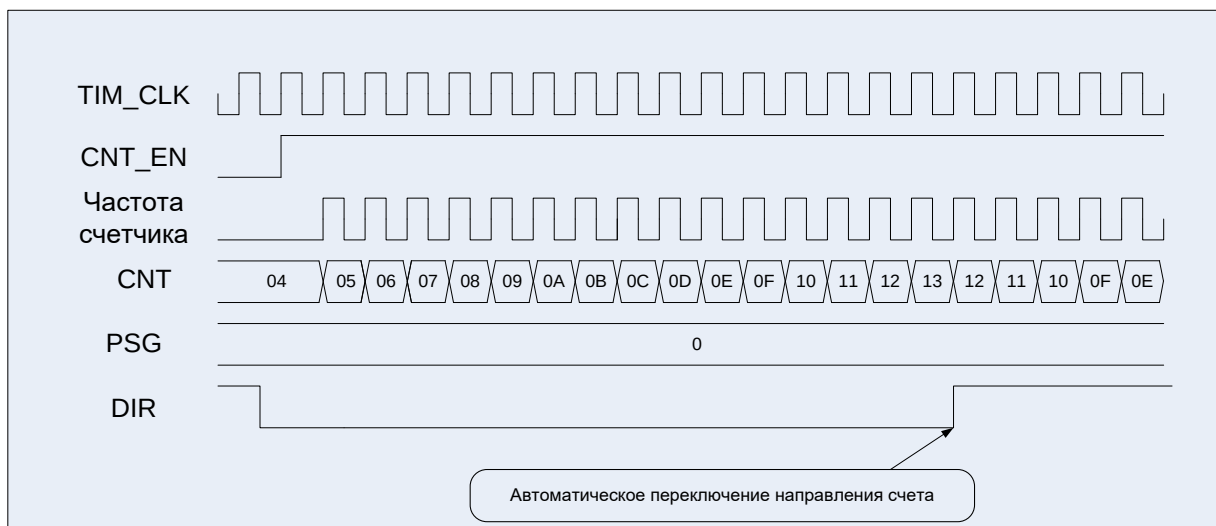


Рисунок 54 – Диаграммы работы таймера, счет вверх/вниз, сначала вверх
Счет вверх/вниз: CNT_MODE = 01, DIR = 1

```
TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000004; //Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
TIMx->TIMx_ARR = 0x00000013; //Основание счета
```

```
//Разрешение работы таймера.
TIMx->TIMx_CNTRL = 0x00000049; //Счет вверх/вниз по TIM_CLK.
```

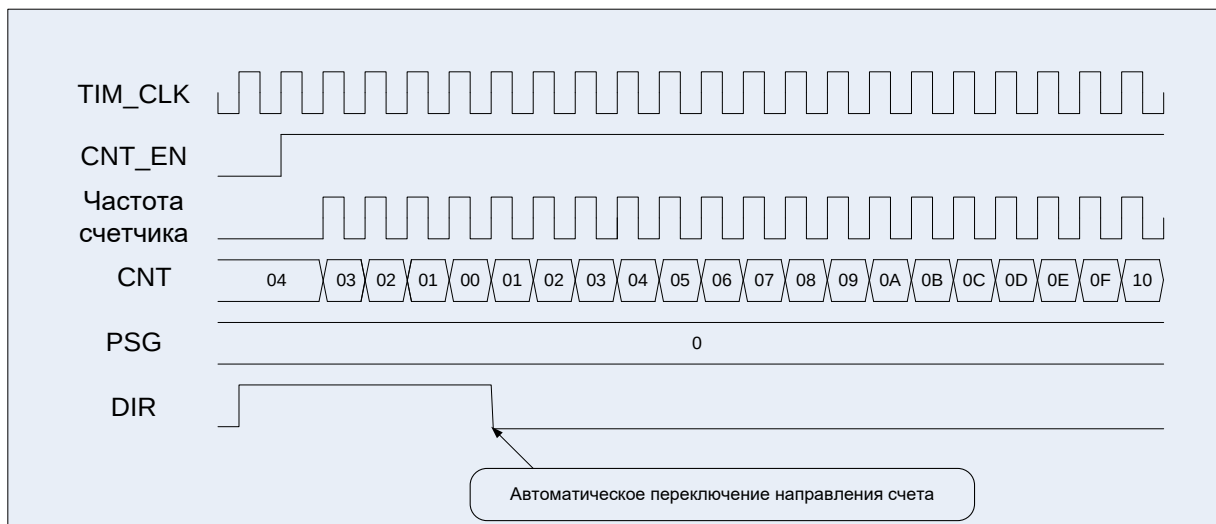


Рисунок 55 – Диаграммы работы таймера, счет вверх/вниз, сначала вниз

6.13.5 Источник событий для счета

Внутренний тактовый сигнал (TIM_CLK).

События в других счетчиках (CNT==ARR в таймере X).

Внешний тактовый сигнал режим 1: События на линиях TxCHO данного счетчика.

Внешний тактовый сигнал режим 2: События на входе ETR данного счетчика.

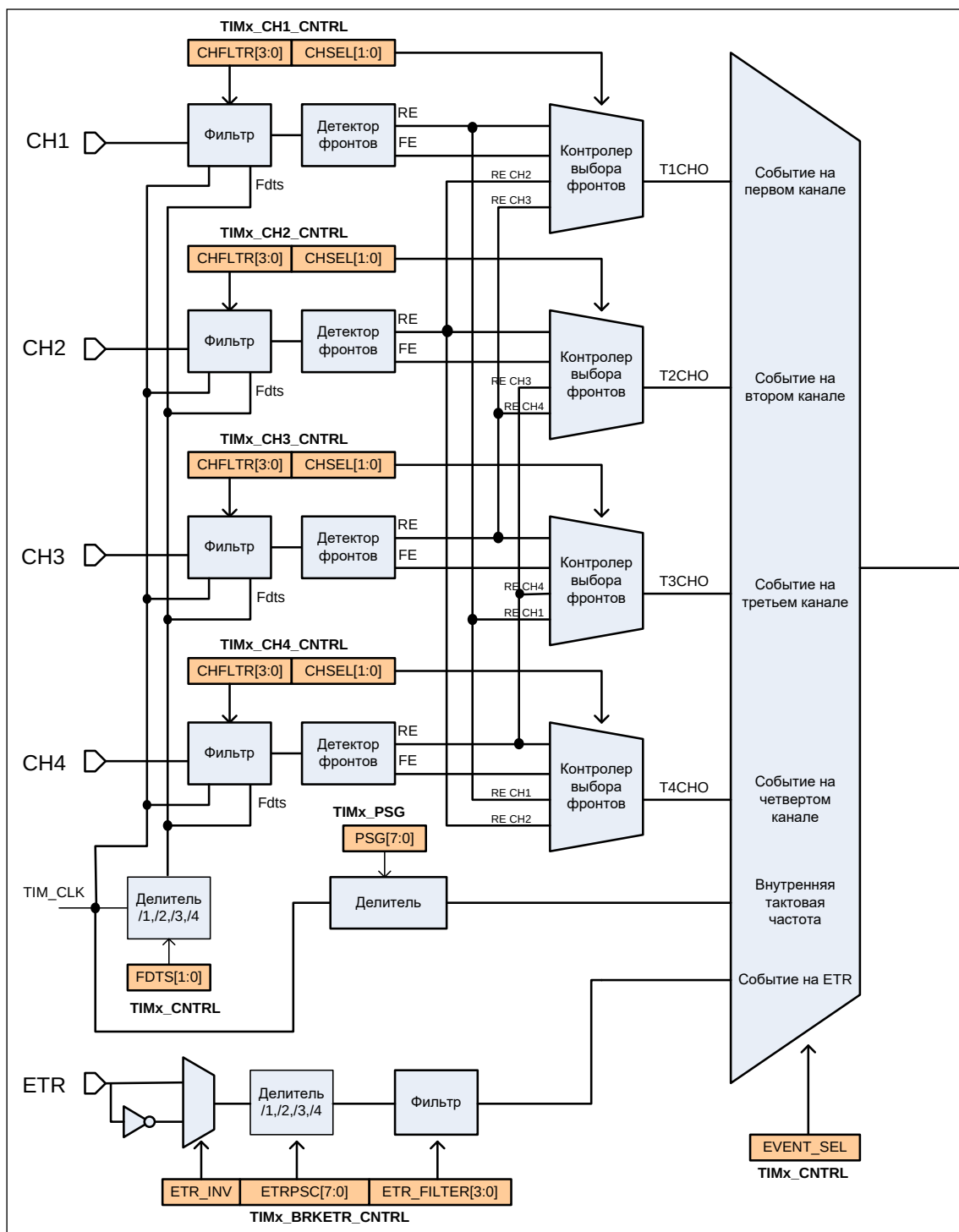


Рисунок 56 – Структурная схема формирования события для счета

6.13.5.1 Внутренний тактовый сигнал (TIM_CLK)

Этот режим выбирается, когда $CNT_MODE = 0x$, $EVENT_SEL = 0000$. Для запуска этого режима необходимо задать начальное значение основного счетчика, значение предварительного делителя основного счетчика, основание счета для основного счетчика и задать режим работы в регистре $TIMx_CNTRL$. Значения регистров $TIMx_CNT$, $TIMx_PSG$ и $TIMx_ARR$ можно изменять даже во время работы счетчика, при этом их значения вступят в силу по $CNT = ARR$ или $CNT = 0$, в зависимости от направления счета. Значение регистра основание счета ($TIMx_ARR$) может вступить в силу мгновенно после его записи в регистр при условии, что сброшен флаг $ARRB_EN = 0$ (регистр $TIMx_CNTRL$). Если значение предварительного делителя основного счетчика не равно нулю, то счетный регистр делителя будет инкрементироваться по каждому импульсу сигнала TIM_CLK до тех пор, пока не достигнет значения, находящегося в регистре делителя. Далее счетный регистр делителя сбрасывается в ноль, содержимое основного счетчика таймера измениться на 1 и снова начинается счет. Поле DIR определяет, в какую сторону будет меняться значение счетчика: $DIR = 0$ – счетчик считает вверх (см. рисунок 57), $DIR = 1$ – счетчик считает вниз (см. рисунок 58). Если $CNT_MODE = 00$, то направление счета определяется полем DIR , если $CNT_MODE = 01$, счетчик считает вверх/вниз с автоматическим изменением DIR (см. рисунок 59).

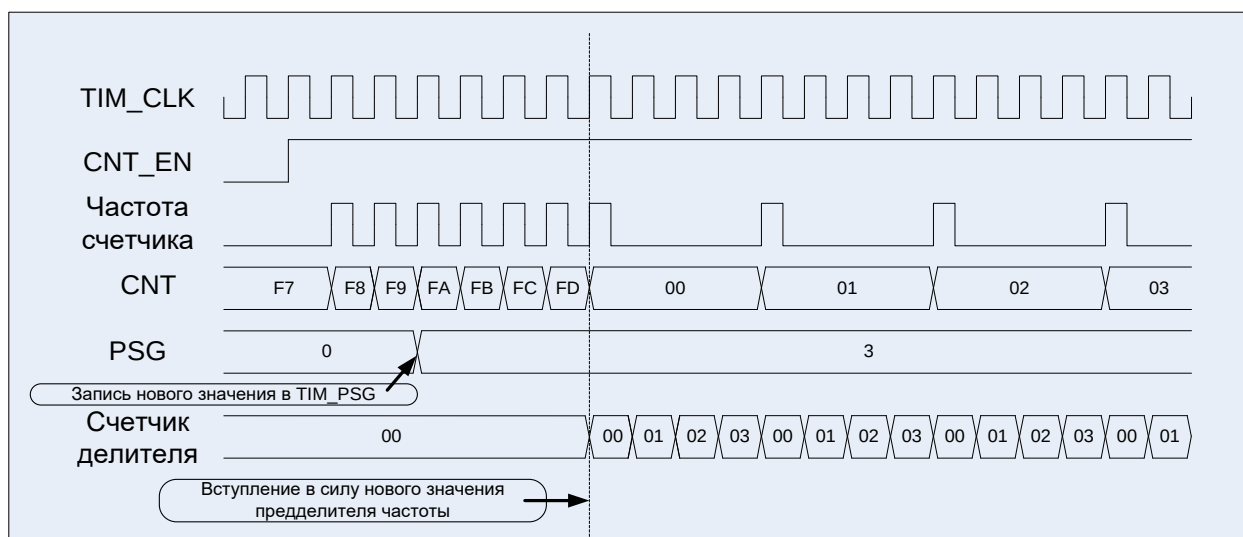


Рисунок 57 – Диаграммы работы счетчика:
счет вверх ($CNT_MODE = 00$, $EVENT_SEL = 0000$, $DIR = 0$)

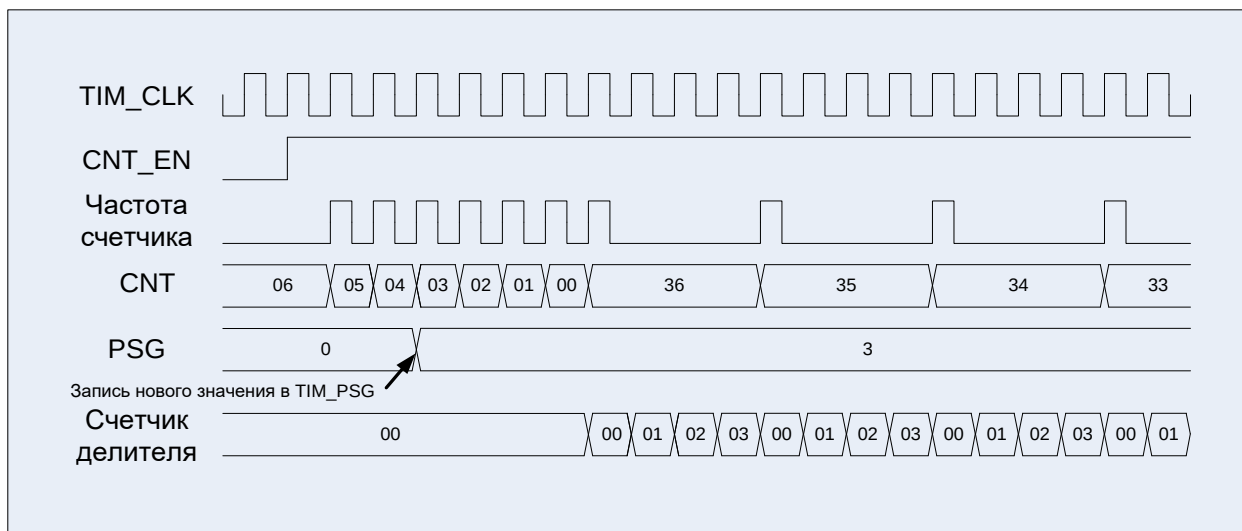


Рисунок 58 – Диаграммы работы счетчика:
счет вниз (CNT_MODE = 00, EVENT_SEL = 0000, DIR = 1)

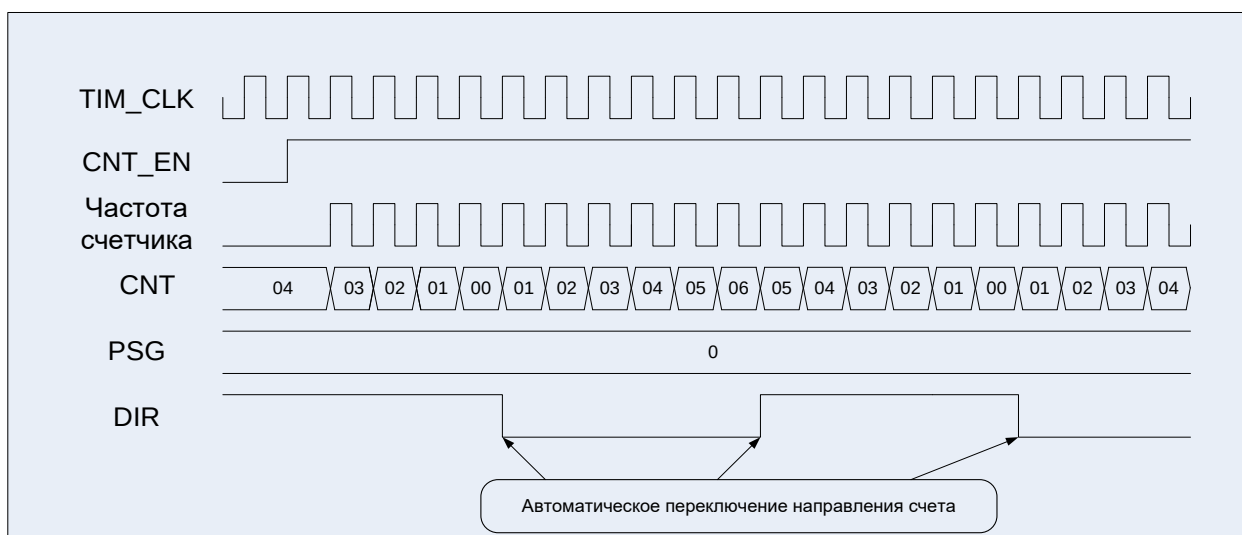


Рисунок 59 – Диаграммы работы счетчика:
счет вниз/вверх (CNT_MODE = 01, EVENT_SEL = 0000, DIR = 1)

6.13.5.2 События в других счетчиках (CNT==ARR в таймере X)

Каждый из блоков таймеров полностью независим друг от друга, но у них предусмотрена возможность синхронизированной друг с другом 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 имеются входы запуска от других трех таймеров, а также внешние входы, связанные с выводами блоков захвата/сравнения.

У каждого из блоков таймеров имеется выход запуска, который соединен с входами других трех таймеров. Синхронизация таймеров возможна в нескольких различных режимах. На рисунке 60 показан пример каскадного соединения счетчиков.

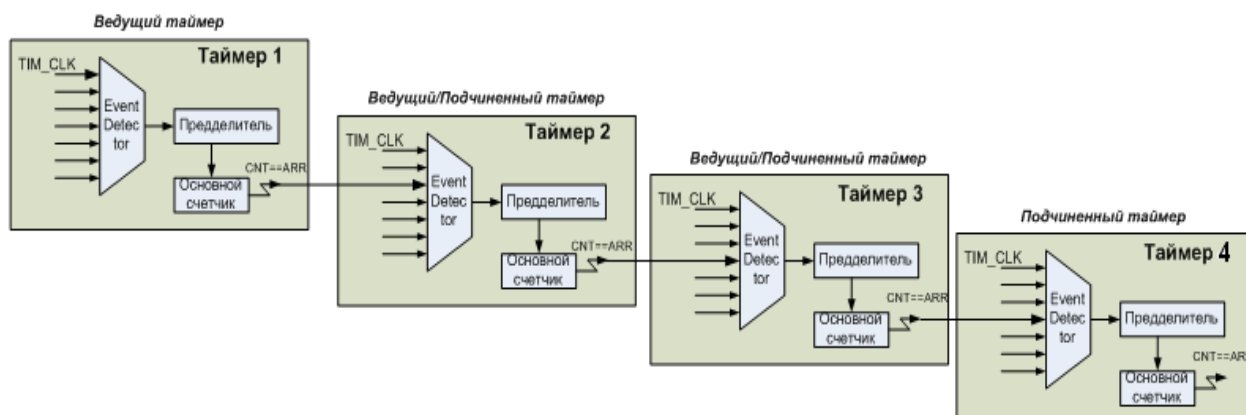


Рисунок 60 – Пример каскадного соединения таймеров

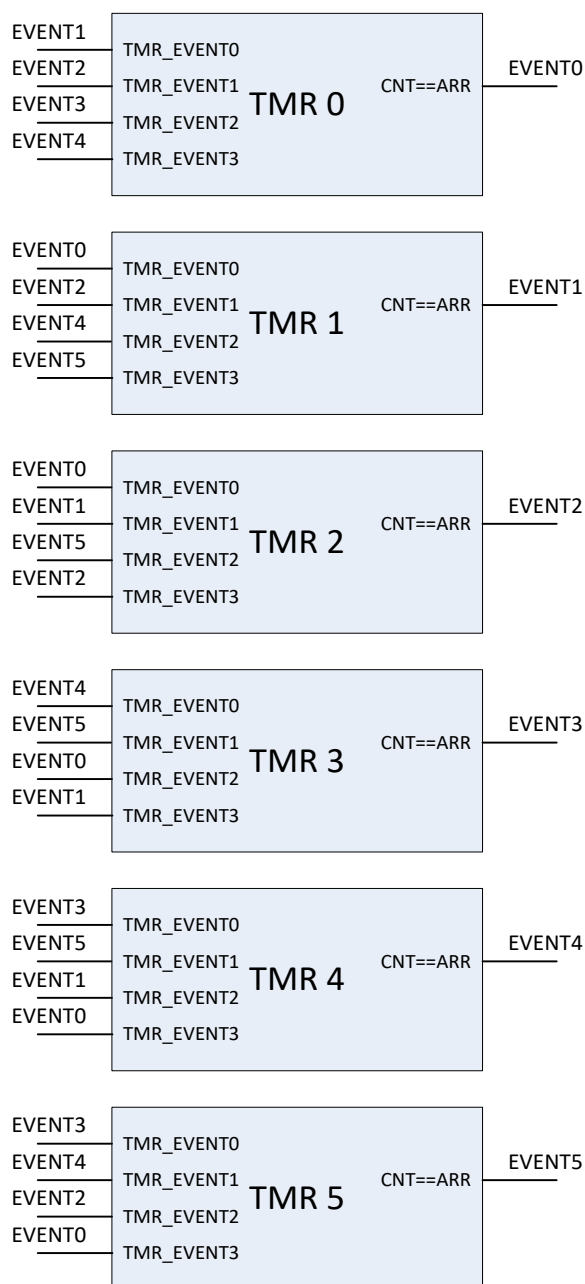


Рисунок 61 – Варианты каскадного объединения таймеров

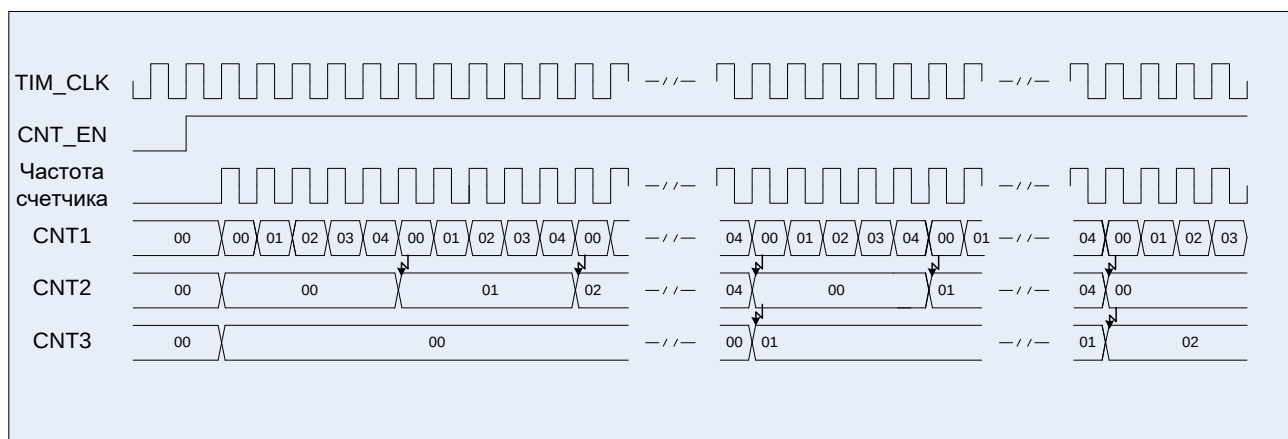


Рисунок 62 – Диаграммы работы трех таймеров в каскаде
 DIR_1, DIR_2, DIR_3 = 0; EVENT_SEL_1 = 0000, EVENT_SEL_2 = 0001, EVENT_SEL_3 = 0010; CNT_MODE_1, CNT_MODE_2, CNT_MODE_3 = 00

6.13.5.3 Внешний тактовый сигнал режим 1. События на линиях TxCH0 данного счетчика

Этот режим выбирается, когда EVENT_SEL = 01xx в регистре TIMx_CNTRL. Счетчик может считать по положительному фронту или по отрицательному фронту на выбранном входе или по положительному фронту на других каналах (см. рис.). На входе сигнала стоит фильтр, с помощью которого можно контролировать длительность сигнала, для фильтрации можно использовать как сигнал TIM_CLK, при этом может быть идентифицированная длительность 1, 2, 4, 8 TIM_CLK, также можно при фильтровании использовать производную от TIM_CLK частоту FDTS. Частота семплирования данных задается в регистре TIMx_CNTRL в поле FDTS.

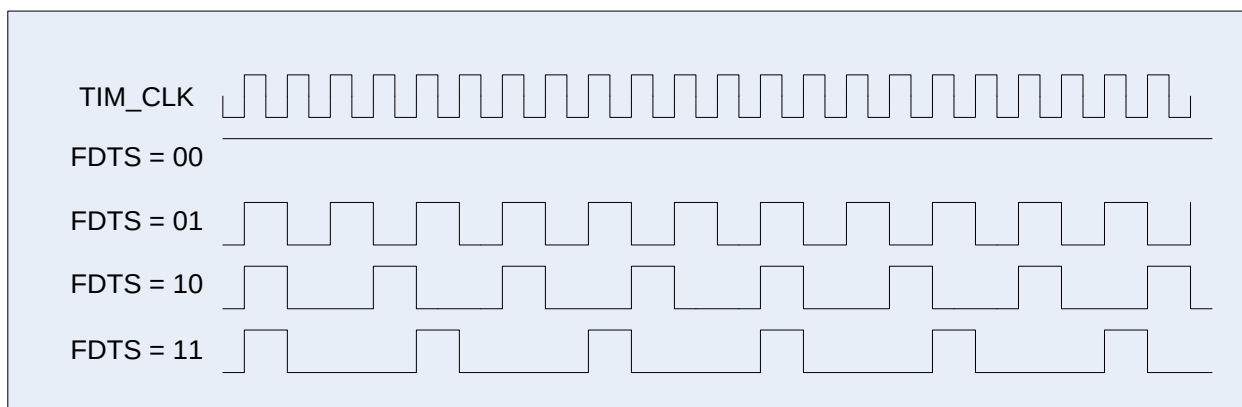


Рисунок 63 – Диаграммы возможных частот семплирования данных (FDTS)

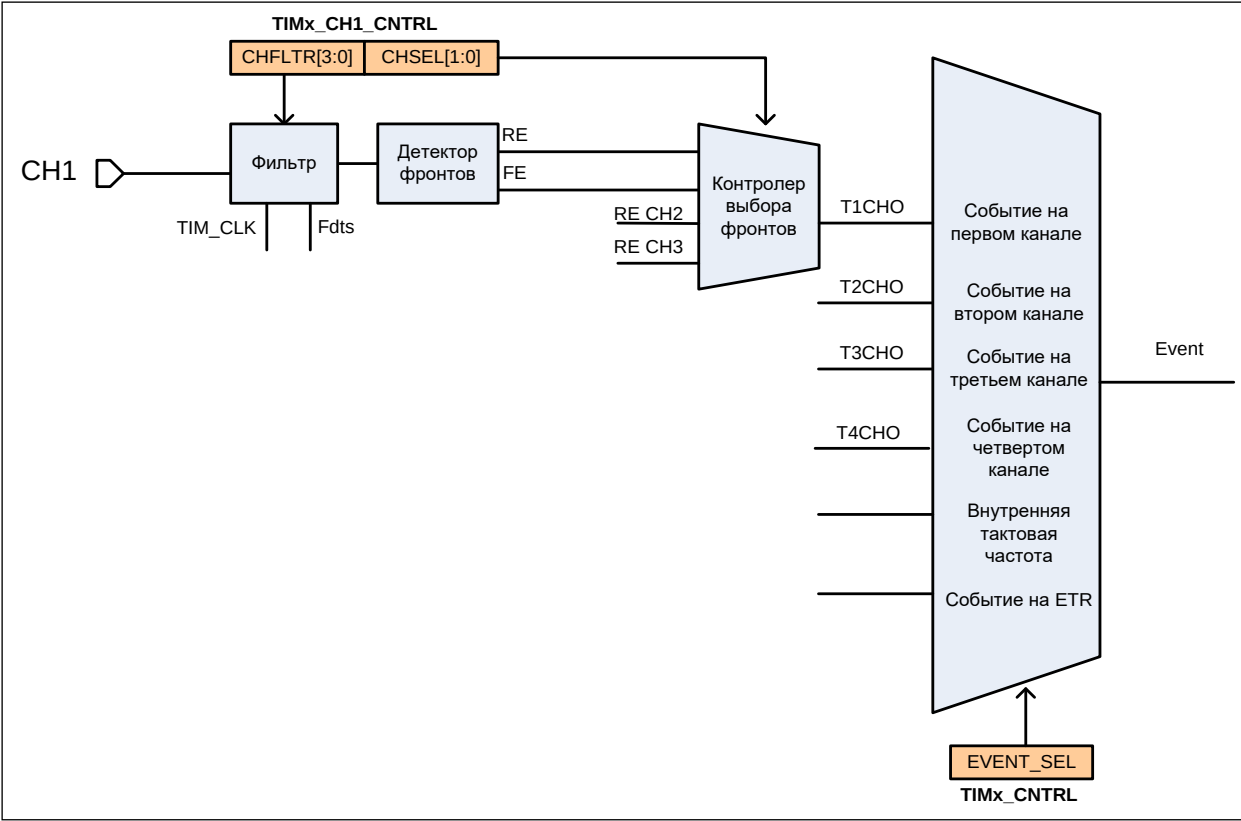


Рисунок 64 – Тактирование с входа первого канала

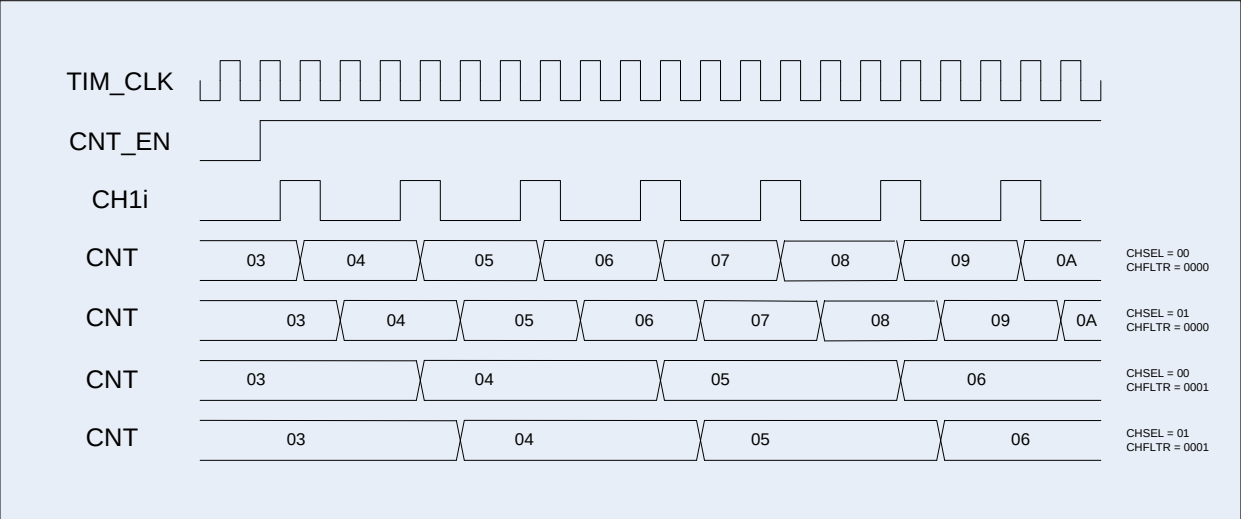


Рисунок 65 – Диаграмма внешнего тактирования с разными вариантами фильтра

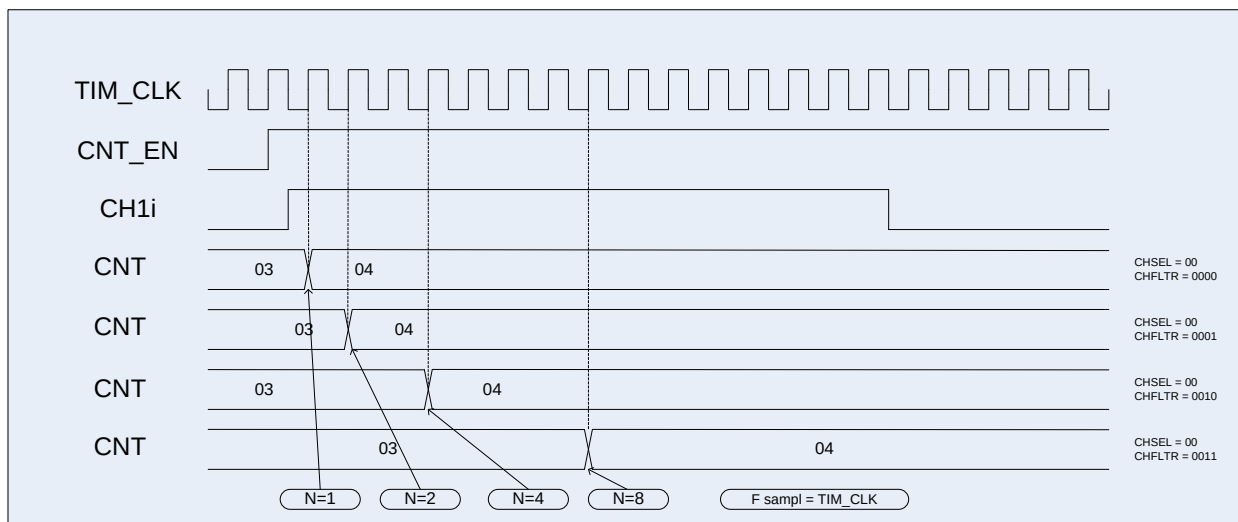


Рисунок 66 – Диаграмма внешнего тактирования с разными вариантами фильтра

6.13.5.4 Внешний тактовый сигнал режим 2. События на входе ETR данного счетчика

Этот режим выбирается, когда EVENT_SEL = 1000 в регистре TIMx_CNTRL. В регистре TIMx_BRKETR_CNTRL можно настроить коэффициент деления 2, 4 или 8 (ETRPSC) данного входа тактовой частоты, а также использовать инверсию входа.

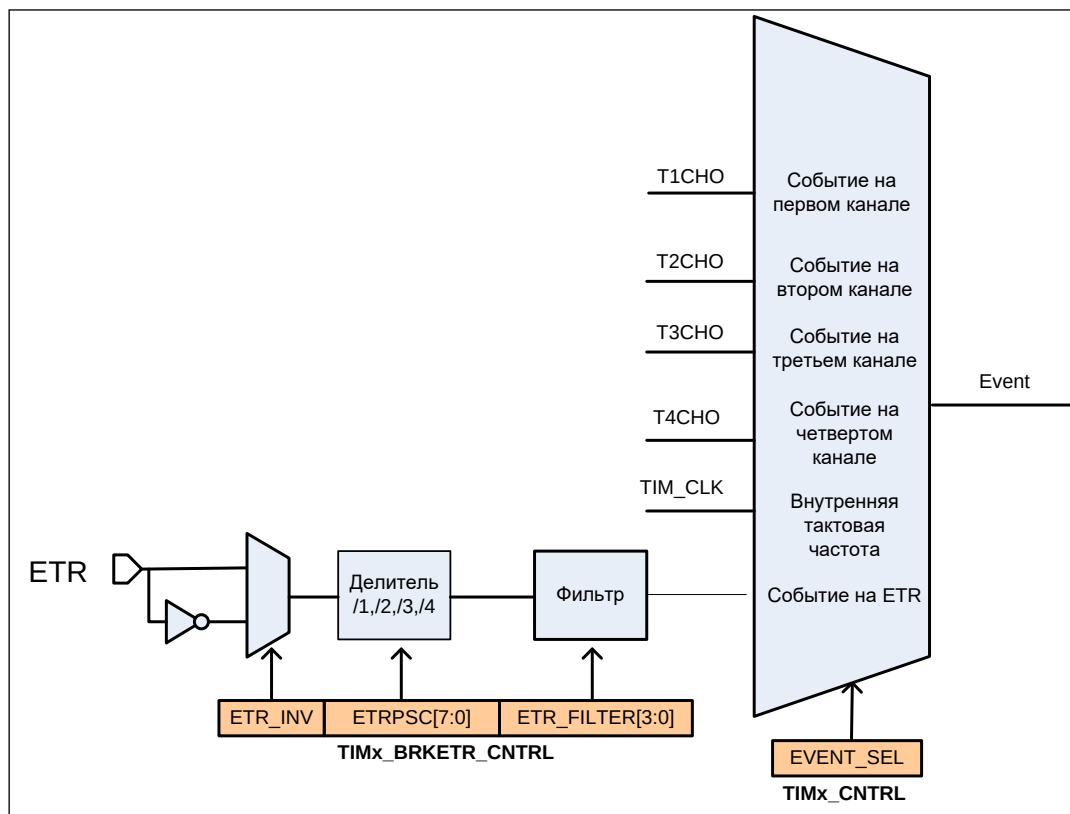


Рисунок 67 – Схема тактирования сигналом с входа ETR

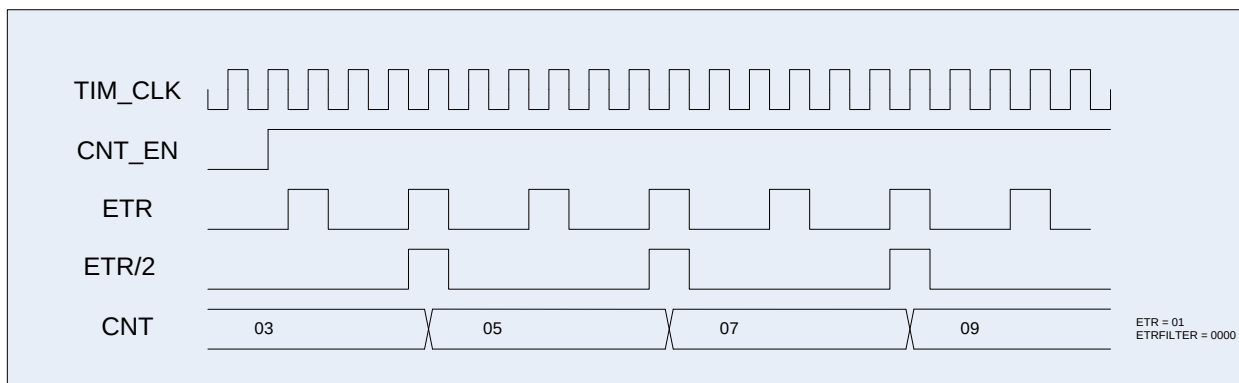


Рисунок 68 – Диаграмма тактирования сигналом с входа ETR

6.13.6 Режим захвата

Структурная схема блока захвата представлена на рисунке 69.

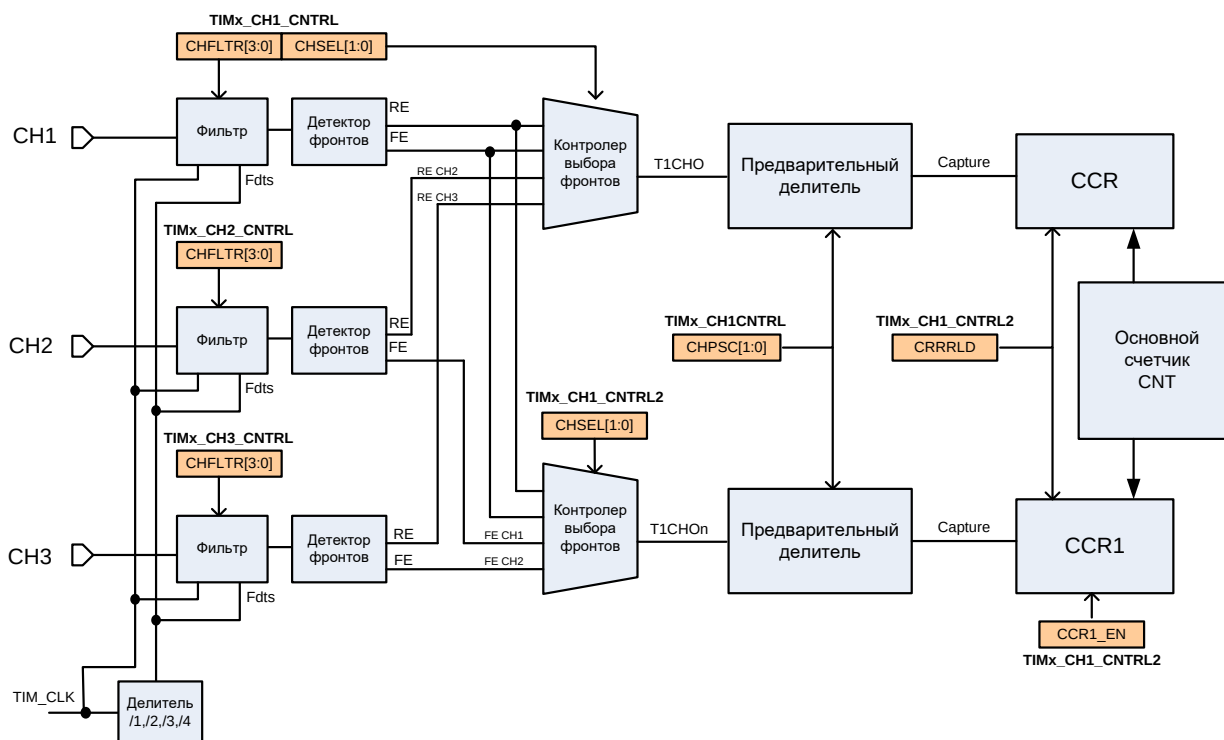


Рисунок 69 – Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо в регистре управления каналом TIMx_CHy_CNTRL записать 1 в поле CAPnPWM. Для регистрации событий по линии CHxi используется схема регистрации событий. Входной сигнал фиксируется в таймере с частотой Fdts, или TIM_CLK. Также вход может быть настроен на прием импульсов заданной длины за счет конфигурирования блока FILTER. На выходе блока Фильтр вырабатывается сигнал положительного перепада и отрицательного перепада. На блоке MUX производится выбор используемого для Захвата сигнала, между положительным фронтом канала, отрицательным фронтом канала и положительными и отрицательными фронтами сигналов от других каналов. После блока MUX предварительный делитель может быть использован для фиксации каждого события, каждого второго, каждого четвертого и каждого восьмого события.

Выход предварительного делителя является сигналом Capture для регистра CCR, и Capture1 для регистра CCR1 при этом в регистры CCR и CCR1 записывается текущее значение основного счетчика CNT.

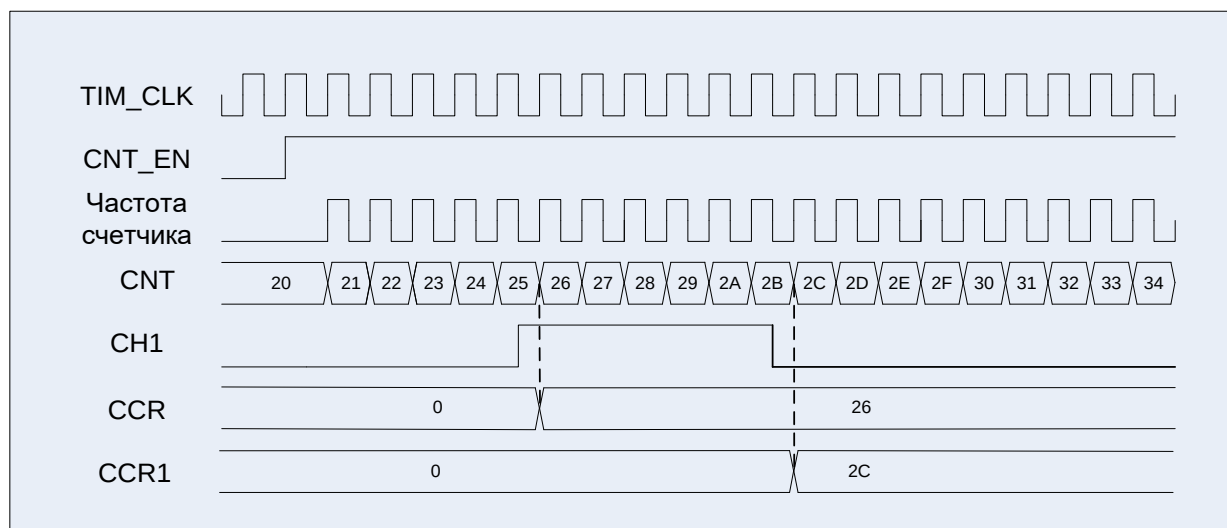


Рисунок 70 – Диаграмма захвата события с входа первого канала

На рисунке 70 показан пример захвата значения основного счетчика в регистр CCR по положительному фронту на входе канала, а в регистр CCR1 по отрицательному фронту на входе канала. В регистре TIMx_IE можно разрешить выработку прерываний по событию захвата на определенном канале, а в регистре TIMx_DMA_RE можно разрешить формирование запросов DMA.

6.13.7 Режим ШИМ

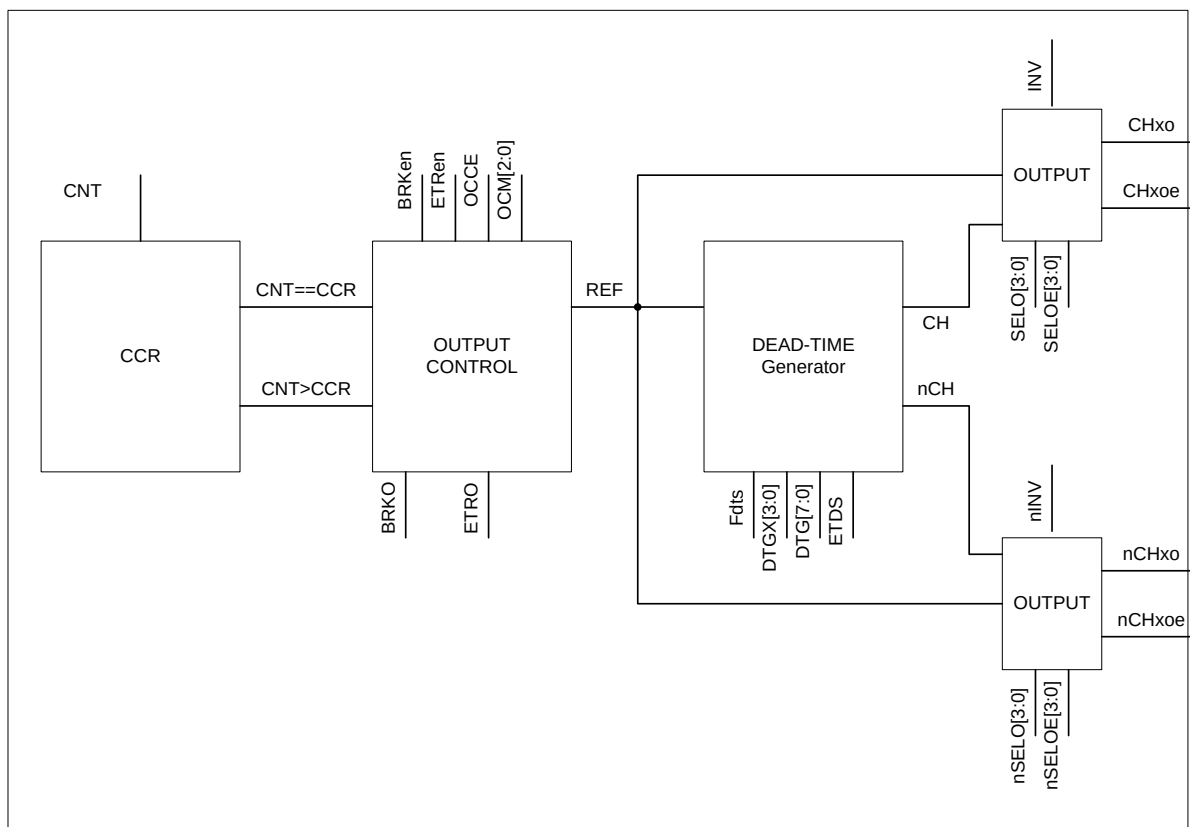


Рисунок 71 – Структурная схема блока сравнения

Для включения режима сравнения для определенного канала необходимо в регистре управления каналом TIMx_CHy_CNTRL записать 0 в поле CAPnPWM. При работе в режиме ШИМ выходной сигнал может формироваться на основании сравнения значения в регистре CCR и основного счетчика CNT или регистров CCR, CCR1 и значения основного счетчика CNT. Полученный сигнал может без изменения выдаваться на выводы CHxO и nCHxO. Либо с применением схемы DEADTIME Generator формируются управляющие сигналы с мертвой зоной. У каждого канала есть два выхода: прямой и инверсный. Для каждого выхода формируется как сигнал для выдачи, так и сигнал разрешения выдачи, т.е. если выход канала должен всегда выдавать тот или иной уровень, то на выводе разрешения выдачи CHxOE (для прямого) и на CHxNOE (для инверсного) должны формироваться "1". Если канал работает на вход (например, режим захвата), то там всегда должен быть "0" для прямого канала. Сигналы OE работают по тем же принципам, что и просто выходные уровни, но у них есть собственные сигналы разрешения вывода SELOE и nSELOE, в которых можно выбрать постоянный уровень, либо формируемый на основании REF.

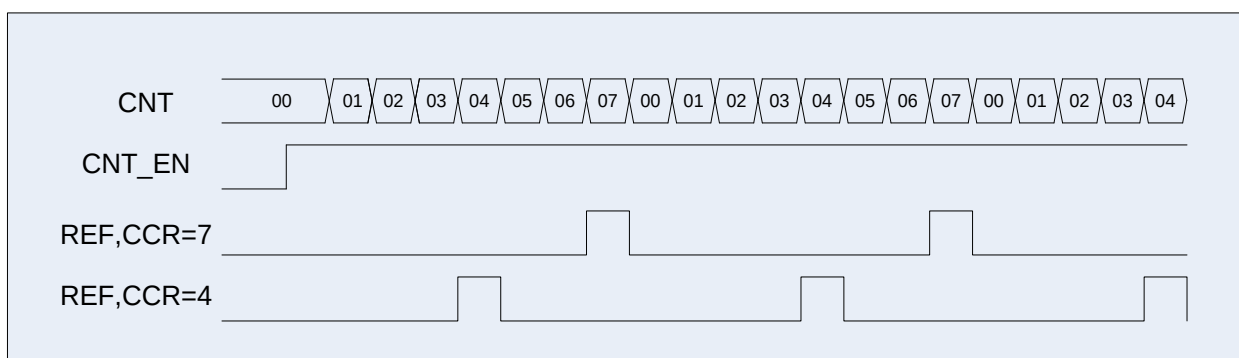


Рисунок 72 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

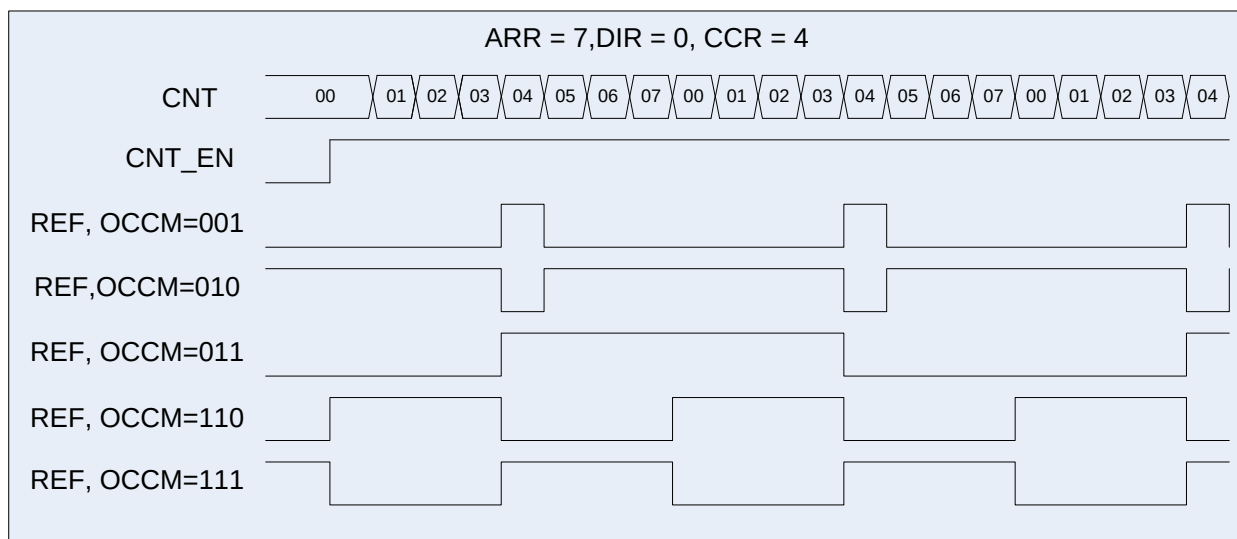


Рисунок 73 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

Сигнал REF может быть очищен с использованием внешнего сигнала с входа ETR или внешнего триггерированного по PCLK сигнала с входа BRK.

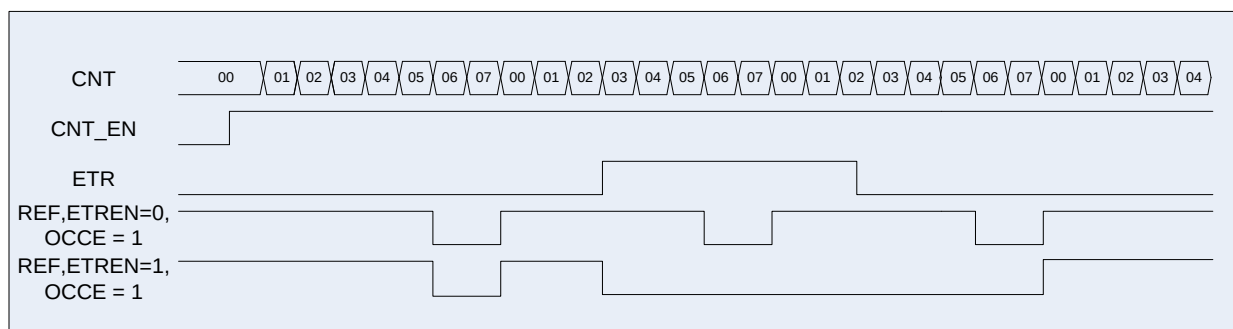


Рисунок 74 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 0

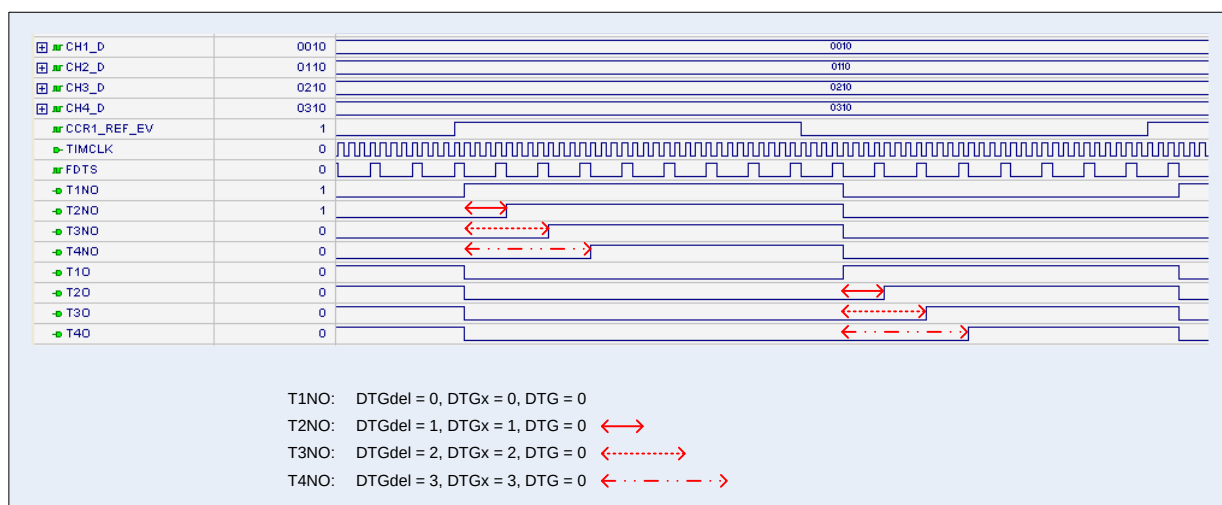


Рисунок 75 – Диаграмма работы схемы DTG

Если CCR1_EN = 1, тогда значение основного счетчика CNT сравнивается со значениями регистров CCR и CCR1, и в зависимости от запрограммированного формата выработки сигнала REF (регистры управления каналами таймера TIMx_CHy_CNTRL поле OCCM) будет формироваться сигнал соответствующей формы.

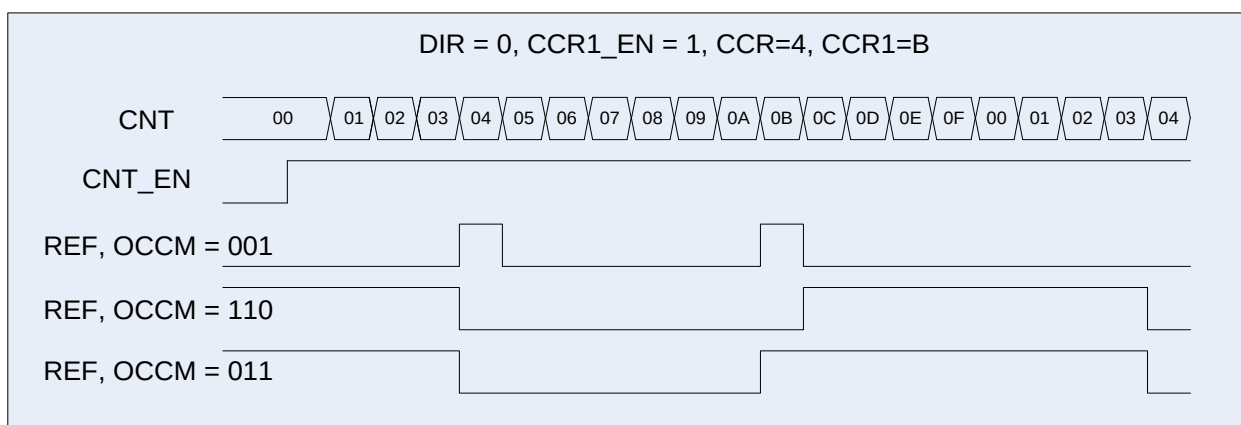


Рисунок 76 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 1

При записи новых значений CCR и CCR1, если установлен бит CRRRLD, то регистры CCR1 и CCR получают новые значения только при CNT = 0, иначе запись осуществляется немедленно. Факт окончания записи обозначается взведением флага WR_CMPL.

6.13.8 Примеры

6.13.8.1 Обычный счетчик

```
RST_CLK->PER_CLOCK = 0xFFFFFFFF;
RST_CLK->TIM_CLOCK = 0x07000000;
TIMx->TIMx_CNTRL = 0x00000000;
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000000; //Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
TIMx->TIMx_ARR = 0x0000000F; //Основание счета

TIMx->TIMx_IE = 0x00000002; //Разрешение генерировать прерывание при CNT = ARR

TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK. Разрешение работы таймера.
```

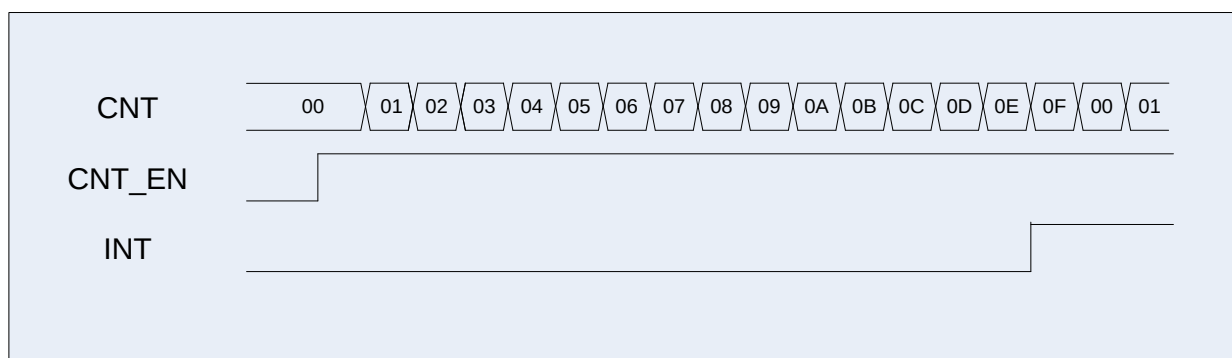


Рисунок 77 – Диаграмма работы счётчика

6.13.8.2 Режим захвата

```
RST_CLK->PER_CLOCK = 0xFFFFFFFF; //Разрешение тактовой частоты таймеров
RST_CLK->TIM_CLOCK = 0x07000000; //Включение тактовой частоты таймеров
TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера
//Настраиваем работу основного счетчика
TIMx->TIMx_CNT = 0x00000000; //Начальное значение счетчика
TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты
TIMx->TIMx_ARR = 0x000000FF; //Основание счета

TIMx->TIMx_IE = 0x00001E00; //Разрешение генерировать прерывание
//по переднему фронту на выходе CAP по всем каналам
//Режим работы каналов - захват
TIMx->TIMx_CHy_CNTRL[0] = 0x00008000;
TIMx->TIMx_CHy_CNTRL[1] = 0x00008002;
TIMx->TIMx_CHy_CNTRL[2] = 0x00008001;
TIMx->TIMx_CHy_CNTRL[3] = 0x00008003;

//Режим работы выхода канала – канал на выход не работает
TIMx->TIMx_CHy_CNTRL1[0] = 0x00000000;
TIMx->TIMx_CHy_CNTRL1[1] = 0x00000000;
TIMx->TIMx_CHy_CNTRL1[2] = 0x00000000;
TIMx->TIMx_CHy_CNTRL1[3] = 0x00000000;

TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK. Разрешение работы таймера.
```

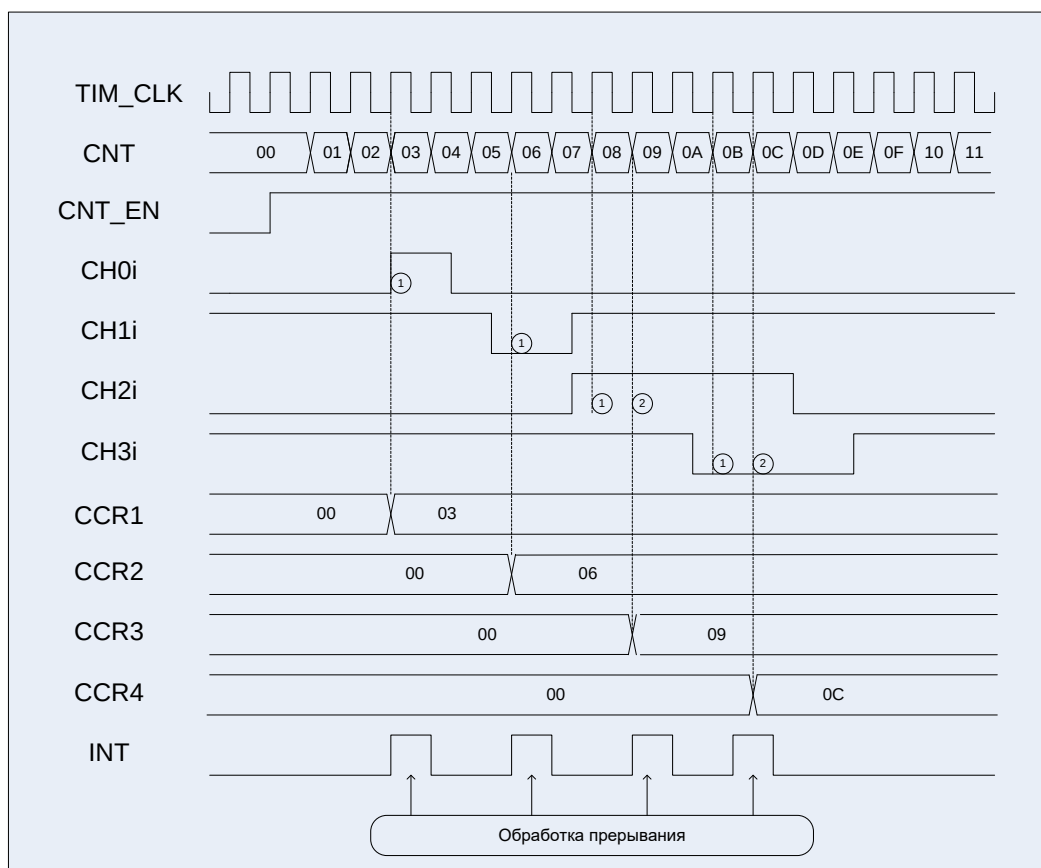


Рисунок 78 – Диаграммы примера работы в режиме захвата

6.13.8.3 Режим ШИМ

RST_CLK->PER_CLOCK = 0xFFFFFFFF; //Разрешение тактовой частоты таймеров

RST_CLK->TIM_CLOCK = 0x07000000; //Включение тактовой частоты таймеров

TIMx->TIMx_CNTRL = 0x00000000; //Режим инициализации таймера

//Настраиваем работу основного счетчика

TIMx->TIMx_CNT = 0x00000000; //Начальное значение счетчика

TIMx->TIMx_PSG = 0x00000000; //Предделитель частоты

TIMx->TIMx_ARR = 0x00000010; //Основание счета

TIMx->TIMx_IE = 0x000001E0; //Разрешение генерировать прерывание

//по переднему фронту на выходе REF по всем каналам

//Режим работы каналов - ШИМ

TIMx->TIMx_CHy_CNTRL[0] = 0x00000200;

TIMx->TIMx_CHy_CNTRL[1] = 0x00000200;

TIMx->TIMx_CHy_CNTRL[2] = 0x00000400;

TIMx->TIMx_CHy_CNTRL[3] = 0x00000600;

//Режим работы выхода канала – канал на выход не работает

TIMx->TIMx_CHy_CNTRL1[0]= 0x00000099;

TIMx->TIMx_CHy_CNTRL1[1]= 0x00000099;

TIMx->TIMx_CHy_CNTRL1[2]= 0x00000099;

TIMx->TIMx_CHy_CNTRL1[3]= 0x00000099;

//Разрешение работы таймера.

TIMx->TIMx_CNTRL = 0x00000001; //Счет вверх по TIM_CLK.

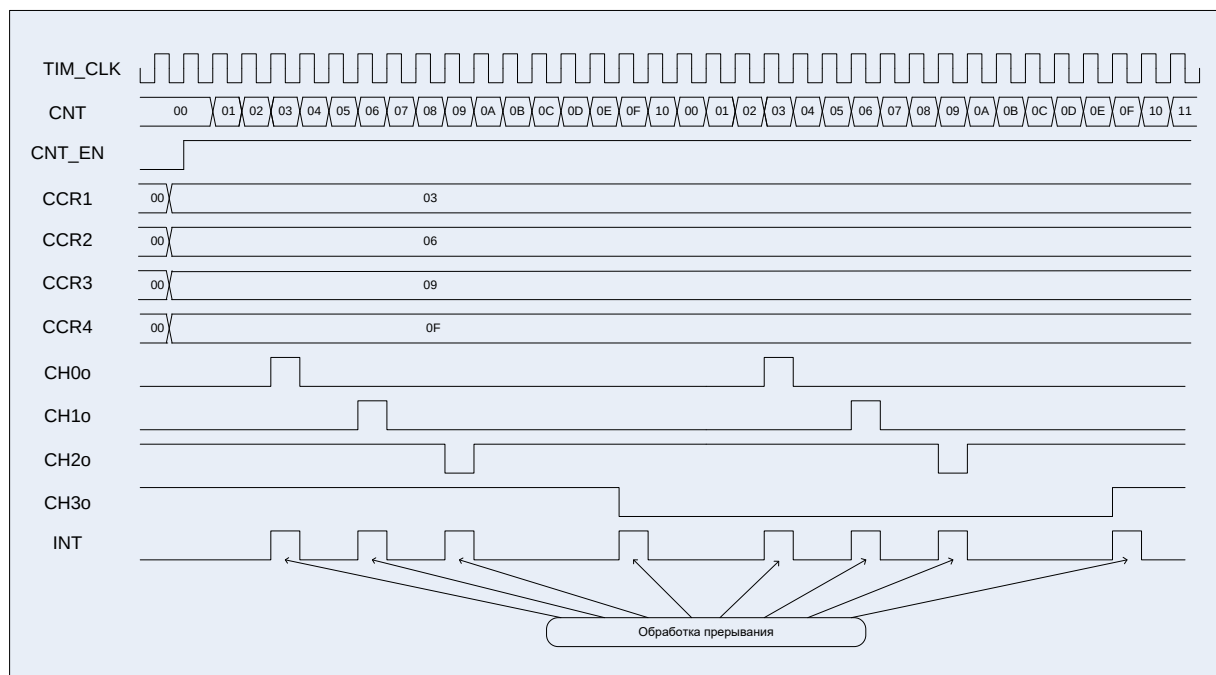


Рисунок 79 – Диаграмма работы в режиме ШИМ

6.13.9 Описание регистров

Таблица 109 – Описание регистров

Базовый адрес	Название	Состояние после сброса	Описание
0x4008A000 0x4008B000 0x4008C000 0x4008D000 0x4008E000 0x4008F000			
Смещение			
0x0000_0000	CNT		Основной счетчик таймера
0x0000_0004	PSG		Делитель частоты при счете основного счетчика
0x0000_0008	ARR		Основание счета основного счетчика
0x0000_000C	CNTRL		Регистр управления основным счетчика

Базовый адрес	Название	Состояние после сброса	Описание
0x0000_0010	CCR1		Регистр сравнения, захвата для 1 канала таймера
0x0000_0014	CCR2		Регистр сравнения, захвата для 2 канала таймера
0x0000_0018	CCR3		Регистр сравнения, захвата для 3 канала таймера
0x0000_001C	CCR4		Регистр сравнения, захвата для 4 канала таймера
0x0000_0020	CH1_CNTRL		Регистр управления для 1 канала таймера
0x0000_0024	CH2_CNTRL		Регистр управления для 2 канала таймера
0x0000_0028	CH3_CNTRL		Регистр управления для 3 канала таймера
0x0000_002C	CH4_CNTRL		Регистр управления для 4 канала таймера
0x0000_0030	CH1_CNTRL1		Регистр управления 1 для 1 канала таймера
0x0000_0034	CH2_CNTRL1		Регистр управления 1 для 2 канала таймера
0x0000_0038	CH3_CNTRL1		Регистр управления 1 для 3 канала таймера
0x0000_003C	CH4_CNTRL1		Регистр управления 1 для 4 канала таймера
0x0000_0040	CH1_DTG		Регистр управления DTG для 1 канала таймера
0x0000_0044	CH2_DTG		Регистр управления DTG для 2 канала таймера
0x0000_0048	CH3_DTG		Регистр управления DTG для 3 канала таймера
0x0000_004C	CH4_DTG		Регистр управления DTG для 4 канала таймера
0x0000_0050	BRKETR_CNTRL		Регистр управления входом BRK и ETR
0x0000_0054	STATUS_		Регистр статуса таймера
0x0000_0058	IE		Регистр разрешения прерывания таймера
0x0000_005C	DMA_RE		Регистр разрешения запросов DMA от прерываний таймера
0x0000_0060	CH1_CNTRL2		Регистр управления 2 для 1 канала таймера
0x0000_0064	CH2_CNTRL2		Регистр управления 2 для 2 канала таймера
0x0000_0068	CH3_CNTRL2		Регистр управления 2 для 3 канала таймера
0x0000_006C	CH4_CNTRL2		Регистр управления 2 для 4 канала таймера
0x0000_0070	CCR11		Регистр сравнения, захвата 1 для 1 канала таймера
0x0000_0074	CCR21		Регистр сравнения, захвата 1 для 2 канала таймера
0x0000_0078	CCR31		Регистр сравнения, захвата 1 для 3 канала таймера
0x0000_007C	CCR41		Регистр сравнения, захвата 1 для 4 канала таймера
0x0000_0080	DMA_RE1		Регистр разрешения запросов DMA от прерываний канала 1 таймера
0x0000_0084	DMA_RE2		Регистр разрешения запросов DMA от прерываний канала 2 таймера
0x0000_0088	DMA_RE3		Регистр разрешения запросов DMA от прерываний канала 3 таймера
0x0000_008C	DMA_RE4		Регистр разрешения запросов DMA от прерываний канала 4 таймера

6.13.9.1 Регистр CNT

Base ADDR=	0x4008A000 0x4008B000 0x4008C000	Offset=	0x0000_0000						
------------	--	---------	-------------	--	--	--	--	--	--

		0x4008D000 0x4008E000 0x4008F000													
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]															

Бит	Имя	Значение	Описание
31...0	CNT[31:0]		Значение основного счетчика таймера

6.13.9.2 Регистр PSG

Base ADDR=		0x4008A000 0x4008B000 0x4008C000 0x4008D000 0x4008E000 0x4008F000				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
PSG[15:0]															

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0	PSG[15:0]		Значение предварительного делителя счетчика Основной счетчик считает на частоте $CLK = TIM_CLK / (PSG + 1)$

6.13.9.3 Регистр TIMx_ARR

Base ADDR=		0x4008A000 0x4008B000 0x4008C000 0x4008D000 0x4008E000 0x4008F000				Offset=		0x0000_0008							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															

Бит	Имя	Значение	Описание
31...0	ARR[31:0]		Основание счета для основного счетчика CNT = [0...ARR]

6.13.9.4 Регистр CNTRL

Base ADDR=	0x4008A000 0x4008B000 0x4008C000 0x4008D000 0x4008E000 0x4008F000	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
				EVENT_SEL[3:0]				CNT_MODE[1:0]		FDTS[1:0]		DIR	WRCMPL	ARBEN	CNT

Бит	Имя	Значение	Описание
31..11	-		Зарезервировано
11..8	EVENT_SEL[3:0]		Биты выбора источника событий 0000 – всегда “0” 0001 – событие на TMR_EVENT0 0010 – событие на TMR_EVENT1 0011 – событие на TMR_EVENT2 (описание TMR_EVENTx в рисунке каскадного объединения) 0100 – событие на первом канале 0101 – событие на втором канале 0110 – событие на третьем канале 0111 – событие на четвертом канале 1000 – событие переднего фронта ETR 1001 – событие заднего фронта ETR 1010 – событие на TMR_EVENT3
7..6	CNT_MODE[1:0]		Режим счета основного счетчика 00 – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при PSG = 0 01 – счетчик вверх/вниз с автоматическим изменением DIR при PSG = 0 10 – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при EVENT = 1 11 – счетчик вверх/вниз с автоматическим изменением DIR при EVENT = 1
5..4	FDTS[1:0]		Частота семплирования данных FDTS 00 – каждый TIM_CLK 01 – каждый второй TIM_CLK 10 – каждый третий TIM_CLK 11 – каждый четвертый TIM_CLK

Бит	Имя	Значение	Описание
3	DIR		Направление счета основного счетчика 0 – вверх, от 0 до ARR 1 – вниз, от ARR до 0
2	WR_CMPL		Окончание записи, при задании нового значения регистров CNT, PSG и ARR 1 – данные не записаны и идет запись 0 – новые данные можно записывать
1	ARRB_EN		Разрешение мгновенного обновления ARR 0 – ARR будет перезаписан в момент записи в ARR 1 – ARR будет перезаписан при завершении счета CNT
0	CNT_EN		Разрешение работы таймера 0 – таймер отключен 1 – таймер включен

6.13.9.5 Регистр CCRx

Base ADDR=	0x4008A000	Offset=	0x0000_0010												
	0x4008B000		0x0000_0014												
	0x4008C000		0x0000_0018												
	0x4008D000		0x0000_001C												
	0x4008E000														
	0x4008F000														

REG Name:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

CCR[31:0]

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

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

CCR[31:0]

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Бит	Имя	Значение	Описание
31...0	CCR[31:0]		Значение CCR, с которым сравнивается CNT при работе в ШИМ режиме. Значение CNT, при котором произошел факт захвата события, в режиме захвата

6.13.9.6 Регистр CCRx1

Base ADDR=		0x4008A000			Offset=		0x0000_0070								
		0x4008B000					0x0000_0074								
		0x4008C000					0x0000_0078								
		0x4008D000					0x0000_007C								
		0x4008E000													
		0x4008F000													
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:0]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[31:0]															

Бит	Имя	Значение	Описание
31...0	CCR1[31:0]		Значение CCR1, с которым сравнивается CNT при работе в ШИМ режиме. Значение CNT, при котором произошел факт захвата события, в режиме захвата

6.13.9.7 Регистр CHx_CNTRL

Base ADDR=		0x4008A000				Offset=		0x0000_0020									
		0x4008B000						0x0000_0024									
		0x4008C000						0x0000_0028									
		0x4008D000						0x0000_002C									
		0x4008E000															
		0x4008F000															
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
CAPnPWM	WRCPML	ETREN	BRKEN	OCCM[2:0]			OCCF	CHPSC[1:0]		CHSEL[1:0]		CHFLTR[3:0]			

Бит	Имя	Значение	Описание
31..16	-		Зарезервировано
15	CAPnPWM		Режим работы канала Захват или ШИМ 1 – канал работает в режиме Захват 0 – канал работает в режиме ШИМ
14	WRCPML		Флаг окончания записи, при задании нового значения регистра CCR 1 – данные не записаны и идет запись 0 – новые данные можно записывать
13	ETREN		Разрешения сброса по выводу ETR 0 – запрещен сброс 1 – разрешен
12	BRKEN		Разрешение сброса по выводу BRK 0 – запрещен сброс 1 – разрешен

Бит	Имя	Значение	Описание
11...9	OCCM[2:0]		Формат выработки сигнала REF в режиме ШИМ Если CCR1_EN = 0: 000 – всегда 0 001 – 1, если CNT = CCR; 010 – 0, если CNT = CCR; 011 – переключение REF, если CNT = CCR; 100 – всегда 0; 101 – всегда 1; 110 – 1, если DIR= 0 (счет вверх), CNT < CCR, иначе 0; 0, если DIR= 1 (счет вниз), CNT > CCR, иначе 1; 111 – 0, если DIR= 0 (счет вверх), CNT < CCR, иначе 1; 1, если DIR= 1 (счет вниз), 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 – 1, если DIR = 0 (счет вверх), CCR < CNT < CCR1, иначе 0; 0, если DIR = 1 (счет вниз), CCR < CNT < CCR1, иначе 1; 111 – 0, если DIR = 0 (счет вверх), CCR < CNT < CCR1, иначе 1; 1, если DIR = 1 (счет вниз), CCR < CNT < CCR1, иначе 0. При условии что CCR < CCR1
8	OCCE		Разрешение работы ETR 0 – запрет ETR 1 – разрешение ETR
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 CHx_i для фиксации значения основного счетчика (регистр MDR_TIMERx->CNT) в регистр CCR: 00 – положительный фронт на входном канале CHx_i 01 – отрицательный фронт на входном канале CHx_i 10 – положительный фронт от других каналов Для первого канала от 2 канала Для второго канала от 3 канала Для третьего канала от 4 канала Для четвертого канала от 1 канала 11 – положительный фронт от других каналов Для первого канала от 3 канала Для второго канала от 4 канала Для третьего канала от 1 канала Для четвертого канала от 2 канала
3...0	CHFLTR[3:0]		Сигнал зафиксирован: 0000 – в 1 триггере на частоте TIM_CLK 0001 – в 2 триггерах на частоте TIM_CLK 0010 – в 4 триггерах на частоте TIM_CLK 0011 – в 8 триггерах на частоте TIM_CLK 0100 – в 6 триггерах на частоте FDTs/2 0101 – в 8 триггерах на частоте FDTs/2 0110 – в 6 триггерах на частоте FDTs/4 0111 – в 8 триггерах на частоте FDTs/4 1000 – в 6 триггерах на частоте FDTs/8 1001 – в 8 триггерах на частоте FDTs/8 1010 – в 5 триггерах на частоте FDTs/16 1011 – в 6 триггерах на частоте FDTs/16

Бит	Имя	Значение	Описание
			1100 – в 8 триггерах на частоте FDTs/16 1101 – в 5 триггерах на частоте FDTs/32 1110 – в 6 триггерах на частоте FDTs/32 1111 – в 8 триггерах на частоте FDTs/32

6.13.9.8 Регистр CHx_CNTRL1

Base ADDR=	0x4008A000 0x4008B000 0x4008C000 0x4008D000 0x4008E000 0x4008F000	Offset=	0x0000_0030 0x0000_0034 0x0000_0038 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
			NINV	NSELO[1:0]	NSELOE[1:0]						NSEL	SELO [1:0]	SELOE [1:0]		

Бит	Имя	Значение	Описание
31..13	-		Зарезервировано
12	NINV		Режим выходной инверсии 0 – выход не инвертируется 1 – выход инвертируется
11..10	NSELO[1:0]		Режим работы выхода канала 00 – всегда на выход выдается 0, канал на выход не работает 01 – всегда на выход выдается 1, канал всегда работает на выход 10 – на выход выдается сигнал REF. 11 – на выход выдается сигнал с DTG
9...8	NSELOE[1:0]		Режим работы канала на выход 00 – всегда на OE выдается 0, канал на выход не работает 01 – всегда на OE выдается 1, канал всегда работает на выход 10 – на OE выдается сигнал REF, при REF = 0 вход, при REF = 1 выход. 11 – на OE выдается сигнал с DTG, при CHn = 0 вход, при CHn = 1 выход
7...5	-		Зарезервировано
4	INV		Режим выходной инверсии 0 – выход не инвертируется 1 – выход инвертируется
3...2	SELO[1:0]		Режим работы выхода канала 00 – всегда на выход выдается 0, канал на выход не работает 01 – всегда на выход выдается 1, канал всегда работает на выход 10 – на выход выдается сигнал REF. 11 – на выход выдается сигнал с DTG
1...0	SELOE[1:0]		Режим работы канала на выход 00 – всегда на OE выдается 0, канал на выход не работает 01 – всегда на OE выдается 1, канал всегда работает на выход 10 – на OE выдается сигнал REF, при REF = 0 вход, при REF = 1 выход. 11 – на OE выдается сигнал с DTG, при CH = 0 вход, при CH = 1 выход

6.13.9.10 Регистр CHx_DTG

Base ADDR=		0x4008A000				Offset=		0x0000_0040								
		0x4008B000						0x0000_0044								
		0x4008C000						0x0000_0048								
		0x4008D000						0x0000_004C								
		0x4008E000														
		0x4008F000														
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
DTG[7:0]								-			EDTS	DTGx[3:0]			

Бит	Имя	Значение	Описание
31..16	-		Зарезервировано
15...8	DTGx[7:0]		Основной делитель частоты Задержка DTGdel = DTGx*(DTG+1)
7...5	-		Зарезервировано
4	EDTS		Частота работы DTG 0 – TIM_CLK 1 – FDTs
3...0	DTG[3:0]		Предварительный делитель частоты DTG

6.13.9.11 Регистр BRKETR_CNTRL

Base ADDR=		0x4008A000				Offset=		0x0000_0050									
		0x4008B000															
		0x4008C000															
		0x4008D000															
		0x4008E000															
		0x4008F000															
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
-								ETR_FILTER[3:0]				ETR_PSC[1:0]		ETRINV	BRKINV

Бит	Имя	Значение	Описание
31..8	-		Зарезервировано
7...4	ETR_FILTER[3:0]		Цифровой фильтр на входе ETR. Сигнал зафиксирован: 0000 – в 1 триггере на частоте TIM_CLK 0001 – в 2 триггерах на частоте TIM_CLK 0010 – в 4 триггерах на частоте TIM_CLK 0011 – в 8 триггерах на частоте TIM_CLK 0100 – в 6 триггерах на частоте FDTs/2 0101 – в 8 триггерах на частоте FDTs/2 0110 – в 6 триггерах на частоте FDTs/4 0111 – в 8 триггерах на частоте FDTs/4 1000 – в 6 триггерах на частоте FDTs/8 1001 – в 8 триггерах на частоте FDTs/8 1010 – в 5 триггерах на частоте FDTs/16 1011 – в 6 триггерах на частоте FDTs/16 1100 – в 8 триггерах на частоте FDTs/16 1101 – в 5 триггерах на частоте FDTs/32 1110 – в 6 триггерах на частоте FDTs/32 1111 – в 8 триггерах на частоте FDTs/32
3...2	ETR_PSC[1:0]		Асинхронный предделитель внешней частоты 00 – без деления 01 – /2 10 – /4 11 – /8
1	ETRINV		Инверсия входа ETR 0 – без инверсии 1 – инверсия
0	BRKINV		Инверсия входа BRK 0 – без инверсии 1 – инверсия

6.13.9.12 Регистр STATUS

Base ADDR=		0x4008A000 0x4008B000 0x4008C000 0x4008D000 0x4008E000 0x4008F000				Offset=		0x0000_0054									
REG Name:																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	CCRCAP1 EVENT[3]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRCAP1EVENT[2:0]			CCRREFEVENT[3:0]				CCRCAPEVENT[3:0]				BRKEVENT	ETRFEEVENT	ETRREEVENT	CNTARR EVENT	CNTZERO EVENT

Бит	Имя	Значение	Описание
31..17	-		Зарезервировано
16..13	CCRCAPEVENT[3:0]		Событие настроенного фронта на входе CNHi канала таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
12...9	CCRREFEVENT[3:0]		Событие переднего фронта на выходе REF каналов таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
8...5	CCRCAPPEVENT[3:0]		Событие настроенного фронта на входе CNHi канала таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
4	BRKEVENT		Триггерированное по PCLK состояние входа BRK, 0 – BRK == 0 1 – BRK == 1 Сбрасывается записью 0, при условии наличия 0 на входе BRK
3	ETRFEEVENT		Событие заднего фронта на входе ETR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события
2	ETRREEVENT		Событие переднего фронта на входе ETR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события
1	CNTARREVENT		Событие совпадения CNT с ARR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса CNT и ARR не изменили состояния, то флаг повторно не взводится
0	CNTZEROEVENT		Событие совпадения CNT с нулем 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса CNT не изменил состояния, то флаг повторно не взводится

6.13.9.13 Регистр IE

Base ADDR=		0x4008A000 0x4008B000 0x4008C000 0x4008D000 0x4008E000 0x4008F000				Offset=		0x0000_0058							
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															CCRCAP1 EVENTIE [3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRCAP1EVENTIE[2:0]			CCRREFEVENTIE[3:0]				CCRCAP1EVENTIE[3:0]				BRKEVENTIE	ETRFEEVENTIE	ETRREVENTIE	CNTARR EVENTIE	CNTZERO EVENTIE

Бит	Имя	Значение	Описание
31..17	-		Зарезервировано
16..13	CCRCAP1 EVENTIE [3:0]		Флаг разрешения прерывания по событию настроенного фронта на входе CHx канала таймера (фиксация значения основного счетчика таймера в регистре CCR1) 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
12...9	CCRREF EVENTIE[3:0]		Флаг разрешения прерывания по событию переднего фронта на выходе REF каналов таймера 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
8...5	CCRCAP EVENTIE [3:0]		Флаг разрешения прерывания по событию настроенного фронта на входе CHx канала таймера (фиксация значения основного счетчика таймера в регистре CCR) 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
4	BRK EVENTIE		Флаг разрешения по триггерированному по PCLK состоянию входа BRK, 0 – нет прерывания 1 – прерывание разрешено
3	ETRFEEVENTIE		Флаг разрешения прерывания по заднему фронту на входе ETR 0 – нет прерывания 1 – прерывание разрешено
2	ETRREVENTIE		Флаг разрешения прерывания по переднему фронту на входе ETR 0 – нет прерывания 1 – прерывание разрешено

Бит	Имя	Значение	Описание
1	CNTARR EVENTIE		Флаг разрешения прерывания по событию совпадения CNT и ARR 0 – нет прерывания 1 – прерывание разрешено
0	CNTZERO EVENTIE		Флаг разрешения прерывания по событию совпадения CNT и нуля 0 – нет прерывания 1 – прерывание разрешено

6.13.9.14 Регистр DMA_REx

Base ADDR=	0x4008A000 0x4008B000 0x4008C000 0x4008D000 0x4008E000 0x4008F000	Offset=	0x0000_005C 0x0000_0080 0x0000_0084 0x0000_0088 0x0000_008C												
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															CCRCAP1 EVENTRE [3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRCAP1 EVENTRE[2:0]			CCRREF EVENTRE[3:0]				CCRCAP EVENTRE[3:0]				BRK EVENTRE	ETRFE EVENTRE	ETRRE EVENTRE	CNTARR EVENTRE	CNTZERO EVENTRE

Бит	Имя	Значение	Описание
31...17	-		Зарезервировано
16...13	CCRCAP1 EVENTRE [3:0]		Флаг разрешения запроса DMA по событию переднего фронта на выходе CAP1 каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
12...9	CCRREF EVENTRE[3:0]		Флаг разрешения запроса DMA по событию переднего фронта на выходе REF каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
8...5	CCRCAP EVENTRE [3:0]		Флаг разрешения запроса DMA по событию переднего фронта на выходе CAP каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
4	BRKEVENTRE		Флаг разрешения по пересинхронизированному по PCLK состоянию входа BRK, 0 – нет запроса DMA

Бит	Имя	Значение	Описание
			1 – запрос DMA разрешен
3	ETRFEEVENTRE		Флаг разрешения запроса DMA по заднему фронту на входе ETR 0 – нет запроса DMA 1 – запрос DMA разрешен
2	ETREEVENTRE		Флаг разрешения запроса DMA по переднему фронту на входе ETR 0 – нет запроса DMA 1 – запрос DMA разрешен
1	CNTARREVENTRE		Флаг разрешения запроса DMA по событию совпадения CNT и ARR 0 – нет запроса DMA 1 – запрос DMA разрешен
0	CNTZEROEVENTRE		Флаг разрешения запроса DMA по событию совпадения CNT и нуля 0 – нет запроса DMA 1 – запрос DMA разрешен

6.14 Контроллер CAN (CANx)

В микроконтроллере реализовано два независимых контроллера интерфейса CAN. Они являются полнофункциональными CAN-узлами, отвечающими требованиям к активным и пассивным устройствам CAN 2.0A и 2.0B и поддерживающими передачу данных на скорости не более 1 Мбит/сек.

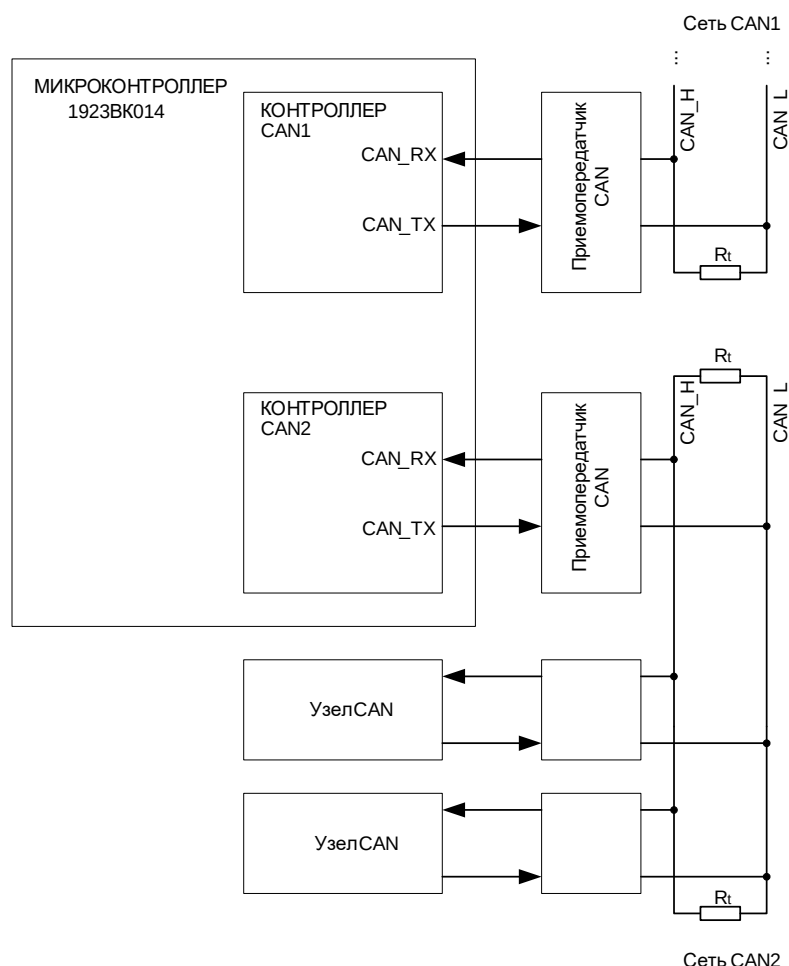


Рисунок 80 – Структурная блок – схема организации сети CAN

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 CAN 2.0 A и B;
- скорость передачи до 1 Мбит/с;
- 32 буфера приема/передачи;
- поддержка приоритетов сообщений;
- 32 фильтра приема;
- маскирование прерываний.

6.14.1 Режимы работы

CAN-контроллер 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.

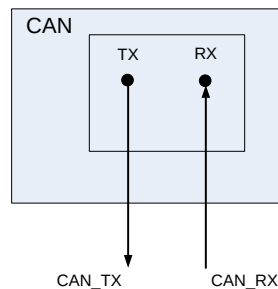


Рисунок 81 – Режим нормальной передачи

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, т.е. линия TX всегда в «1», но внутри контроллера все управляющие сигналы проходят.

Режим самотестирования – Self Test Mode (регистр CAN_STATUS: STM = 1, ROM = 0)

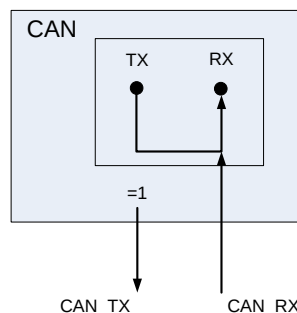


Рисунок 82 – Режим работы только на прием – ReceiveOnlyMode

Выводы CAN_TX и CAN_RX отключены, вся передаваемая информация видна только внутри контроллера.

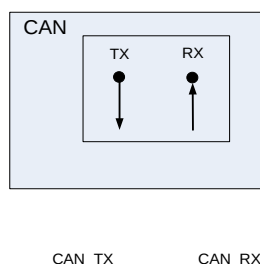


Рисунок 83 – Режим самотестирования – SelfTestMode

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.

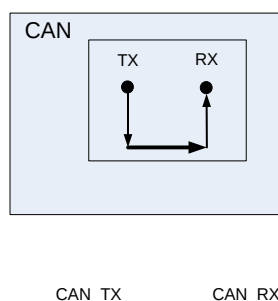


Рисунок 84 – Режим инициализации для задания параметров связи

Еще одна важная функция CAN-контроллера – фильтрация получаемых сообщений. Поскольку CAN является широковещательной шиной, каждое переданное сообщение принимается всеми узлами шины. В CAN-шине любой разумной степени сложности передается достаточно большое число сообщений. Задачей каждого подключенного к CAN-узлу ЦПУ является реагирование на CAN-сообщения. Таким образом, чтобы избавить CAN-контроллер от проблемы приема в буфер нежелательных сообщений, необходима их фильтрация. У CAN-контроллера микроконтроллеров 1986BK014 имеется 32 регистра фильтров и 32 регистра масок, которые можно использовать для блокировки всех CAN-сообщений, кроме избранных сообщений или групп сообщений.

6.14.2 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 CAN интерфейса отправляет пакет перегрузки в соответствии со стандартом;
- основными пакетами на шине CAN являются **пакеты данных**. Пакет данных передает данные от передатчика приемнику. Пакеты могут быть стандартными и расширенными. Отличие пакетов заключается в размере полей идентификатора. Пакеты с 11-разрядным идентификатором – называются стандартными пакетами, пакеты, содержащие 29-разрядные идентификаторы, называются расширенными пакетами. При передаче идентификационной информации происходит автоматический арбитраж на шине CAN таким образом, чтобы пакет с меньшим значением поля ID остался на шине. На шине не допускается наличие двух или более узлов с одним и тем же идентификатором. Размер передаваемых данных кодируется в поле DLC и может составлять от 0 до 8 байт. После передачи поля данных контроллер автоматически передает рассчитанное значение CRC. Если хотя бы один из узлов принял пакет, то он выставляет ACK подтверждение на шине, если хотя бы один из узлов обнаружит ошибку, то на шину будет выставлен пакет ошибки. Таким образом, обеспечивается гарантированность доставки сообщений.

Пакеты данных и пакеты удаленного запроса данных отделяются от предшествующих пакетов межкадровым пространством.

6.14.3 Структура пакета данных (DataFrame)

Пакет данных состоит из 7 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.

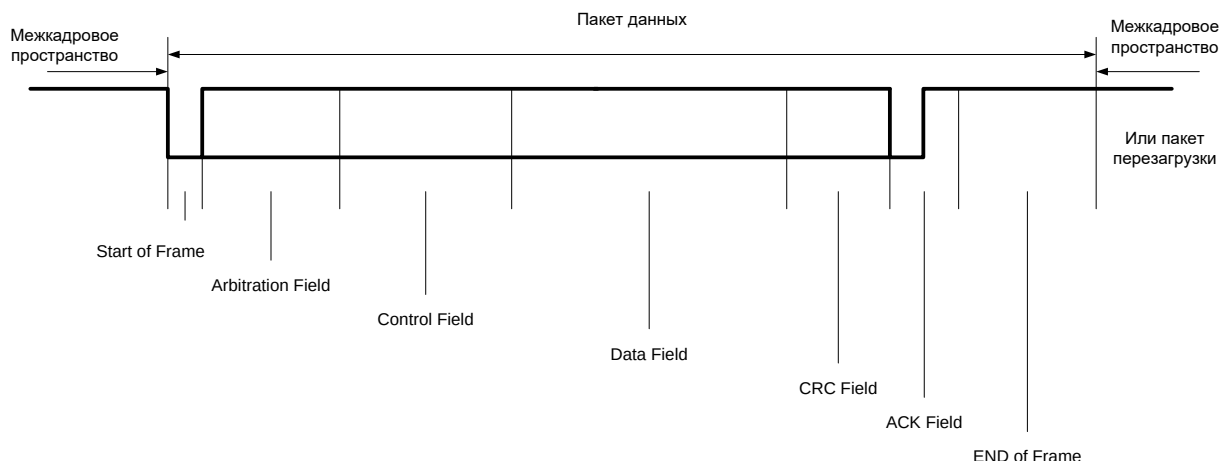


Рисунок 85 – Пакет сообщения CAN

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.

6.14.3.1 Начало пакета (Startofframe)

Начало пакета отмечает начало пакета данных или пакета удаленного запроса данных. Это поле состоит из одиночного доминантного бита. Узлу разрешено начать передачу, когда шина свободна. Все узлы должны синхронизироваться по фронту, вызванному передачей поля «начало пакета» узла, начавшего передачу первым.

6.14.3.2 Поле арбитража (Arbitrationfield)

Формат поля арбитража отличается для стандартного и расширенного форматов:

- в стандартном формате поле арбитража состоит из 11 разрядного идентификатора и RTR-бита;

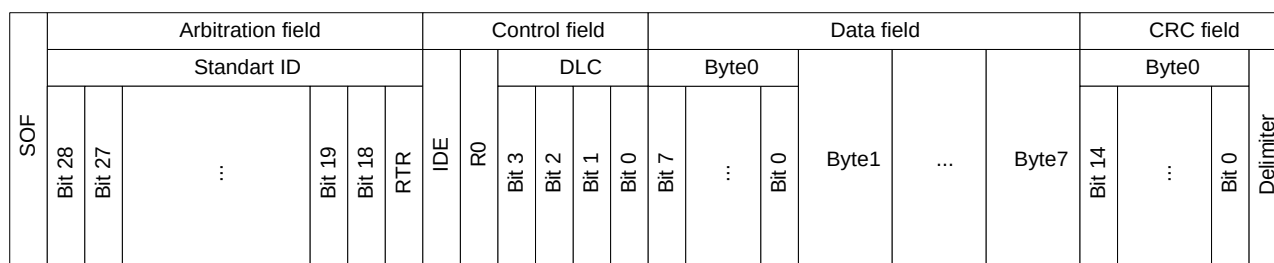


Рисунок 86 – Структура стандартного пакета данных

- в расширенном формате поле арбитража состоит из 29 разрядного идентификатора, SRR-бита, IDE-бита и RTR-бита.

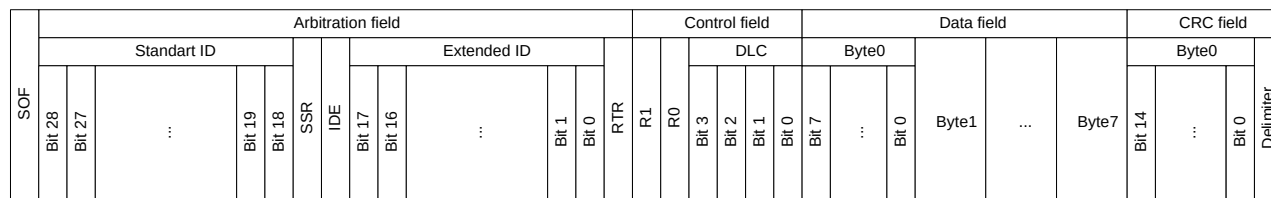


Рисунок 87 – Структура расширенного пакета данных

6.14.3.3 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 – 11 бит;**
- **ExtendedID – 18 бит.**

Standart ID состоит из 11 бит. Эта секция передается в порядке от Bit28 до Bit18. Это эквивалентно формату стандартного идентификатора. **Standart ID** определяет базовый приоритет расширенного пакета.

Extended ID состоит из 18 бит. Эта секция передается в порядке от Bit17 до Bit0. В стандартном пакете идентификатор сопровождается RTR битом.

6.14.3.4 Бит RTR

Бит запроса удаленной передачи. В пакетах данных RTR бит должен быть передан нулевым уровнем. Внутри пакета удаленного запроса данных RTR бит должен быть единичным. В расширенном пакете сначала передается **StandartID**, с последующими битами IDE и SRR. **Extended ID** передается после SRR бита.

6.14.3.5 Бит SRR (расширенный формат)

Заменитель бита удаленного запроса. SRR – единичный бит. Он передается в расширенных пакетах в позиции RTR бита. Таким образом, он заменяет RTR – бит стандартного пакета.

Следовательно, при одновременной передаче стандартного пакета и расширенного пакета, **StandartID** которого совпадает с идентификатором стандартного пакета, стандартный пакет преобладает над расширенным пакетом.

6.14.3.6 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.

6.14.3.7 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.

6.14.3.8 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.

6.14.3.9 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.

6.14.3.10 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.

6.14.3.11 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".

6.14.3.12 Конец пакета (End of frame)

Каждый пакет данных и пакет удаленного запроса данных ограничен последовательностью флагов, состоящей из семи единичных бит.

6.14.4 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, RTR бит пакета удаленного запроса данных - единичный. В этом пакете отсутствует поле данных. При этом значение кода длины данных может принимать любое значение в пределах допустимого диапазона [0,8]. Значение кода длины данных соответствует коду длины данных кадра данных. RTR бит указывает, является ли переданный кадр кадром данных.

6.14.5 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 88 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.

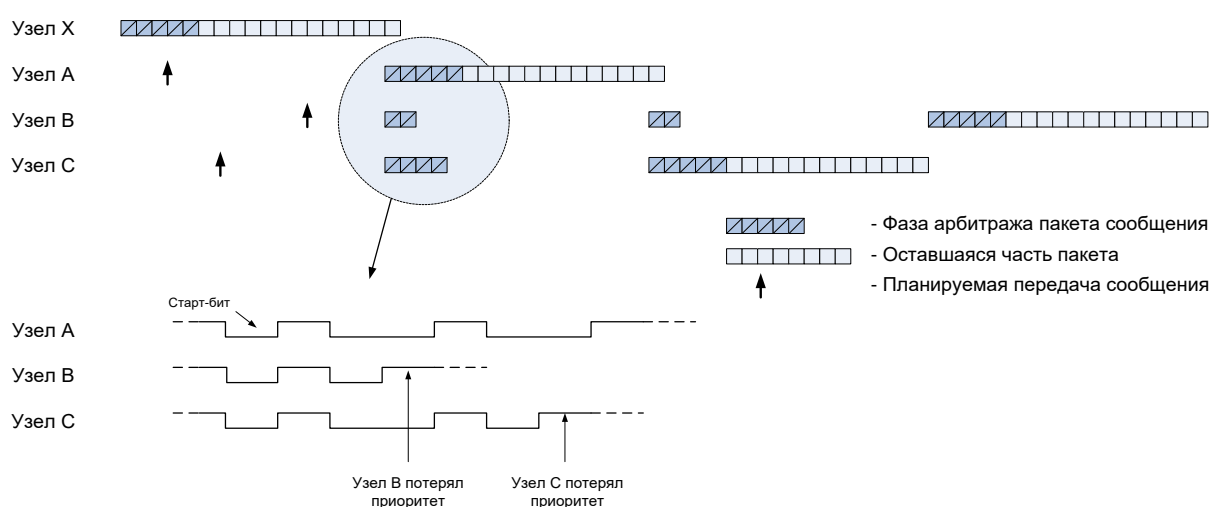


Рисунок 88 – Арбитраж на шине CAN

В случае «проигрыша» арбитража в регистре статуса контроллера CAN будет установлен флаг ID_LOWER.

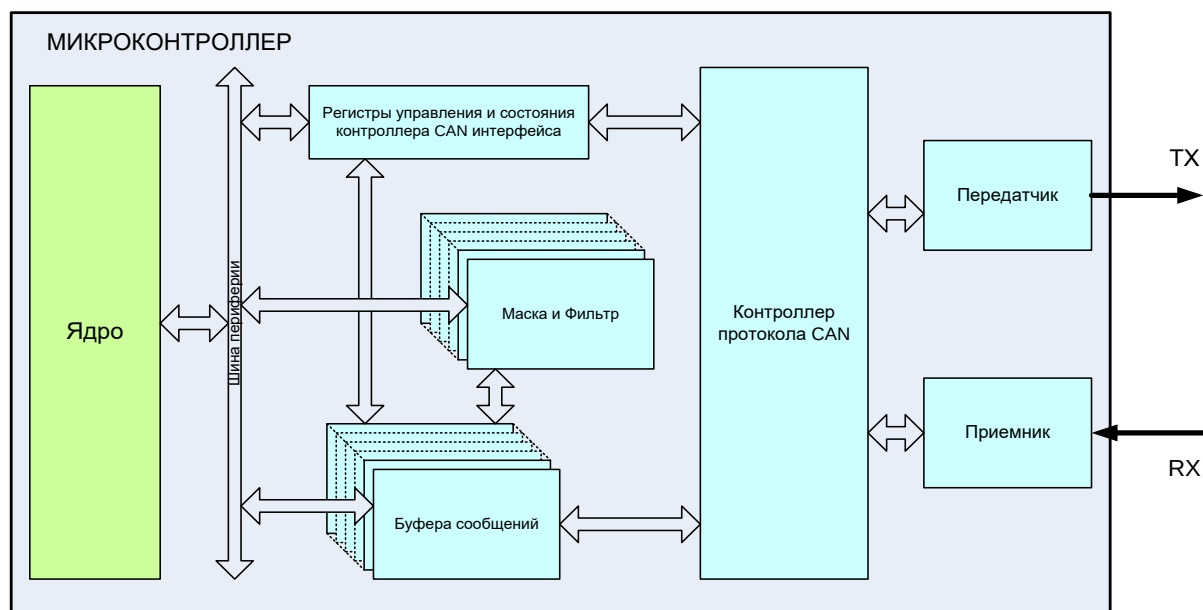


Рисунок 89 – Структурная блок-схема контроллера CAN

6.14.6 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 CAN контроллера.

После подачи тактового сигнала на блок таймера можно приступить 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.

6.14.7 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.

6.14.8 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 RTR запрос. В регистре INT_TX при необходимости настроить генерацию прерывания передачи для соответствующего буфера. В регистре управления этим буфером (BUFF_CON[x]) проверить, что флаг TX_REQ = 0, задать приоритет отправляемого сообщения PRIOR_0, установить разрешение ответа при приеме RTR в буфер (RTR_EN=1), задать RX_TX = 0 для разрешения отправки сообщения и задать EN = 1 для разрешения работы буфера. В регистре идентификации задать необходимые SID и EID, в регистре BUF_xx_DLC указать формат пакета (расширенный или стандартный) и указать длину передаваемых данных в поле DLC. В регистрах данных CAN_BUF[x].DATAL и CAN_BUF[x].DATAH задать необходимые для отправки данные. Далее можно переходить к выполнению остальной части программы с отправкой CAN сообщений. Отправка сообщения буфером будет произведена по RTR запросу, удовлетворяющему механизму фильтрации для принимаемых сообщений, который выбран для данного буфера.

6.14.9 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.

6.14.10 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.

6.14.11 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.

6.14.12 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:

$$T_{BIT} = \frac{1}{\text{Скорость передачи}}. \quad (8)$$

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:

$$\text{Normal Bit Time} = TQ \cdot (\text{Sync_Seg} + \text{PSEG} + \text{SEG1} + \text{SEG2}). \quad (9)$$

Время TQ фиксировано и определяется периодом генератора и программируемым прескалером BRP со значением от 1 до 65536:

$$TQ (\mu s) = \frac{BRP + 1}{CANCLKn (MHz)} \quad (10)$$

или

$$TQ (\mu s) = (BRP + 1) \cdot Tclk (\mu s). \quad (11)$$

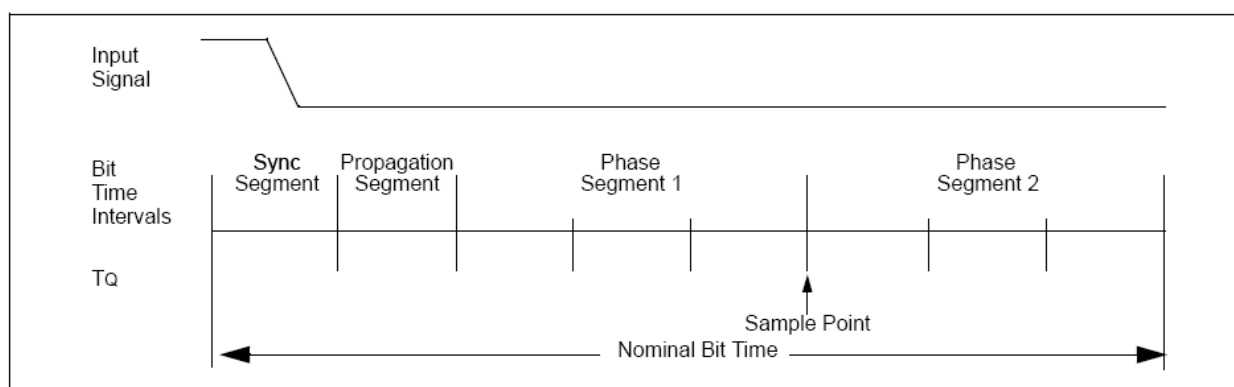


Рисунок 90 – Структура битового интервала

6.14.12.1 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.

6.14.12.2 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

6.14.12.3 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.

6.14.13 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.

6.14.13.1 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.

6.14.13.2 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).

6.14.14 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.

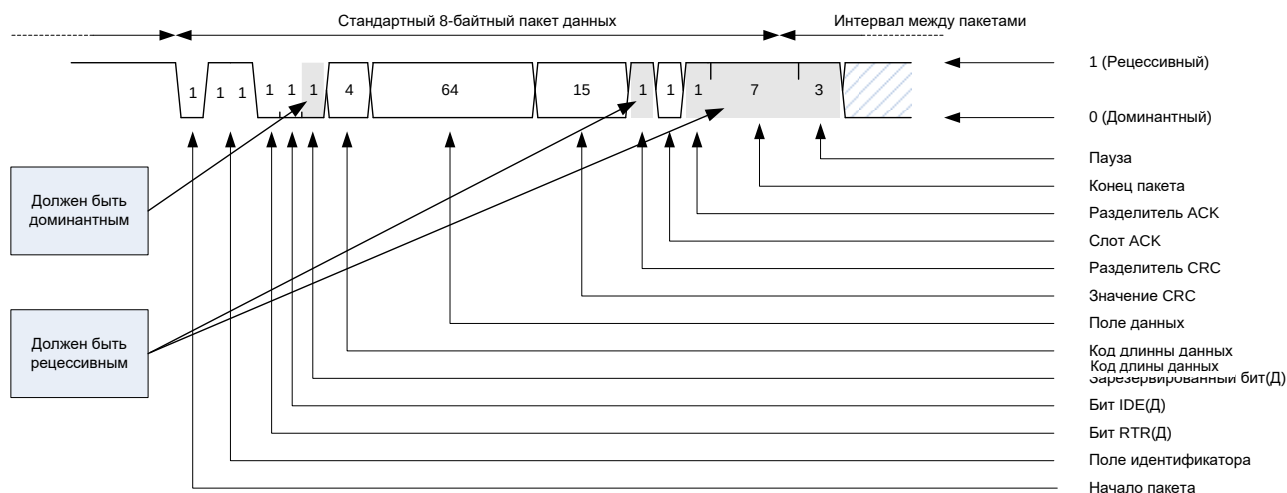


Рисунок 91 – Контроль формата пакета

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.

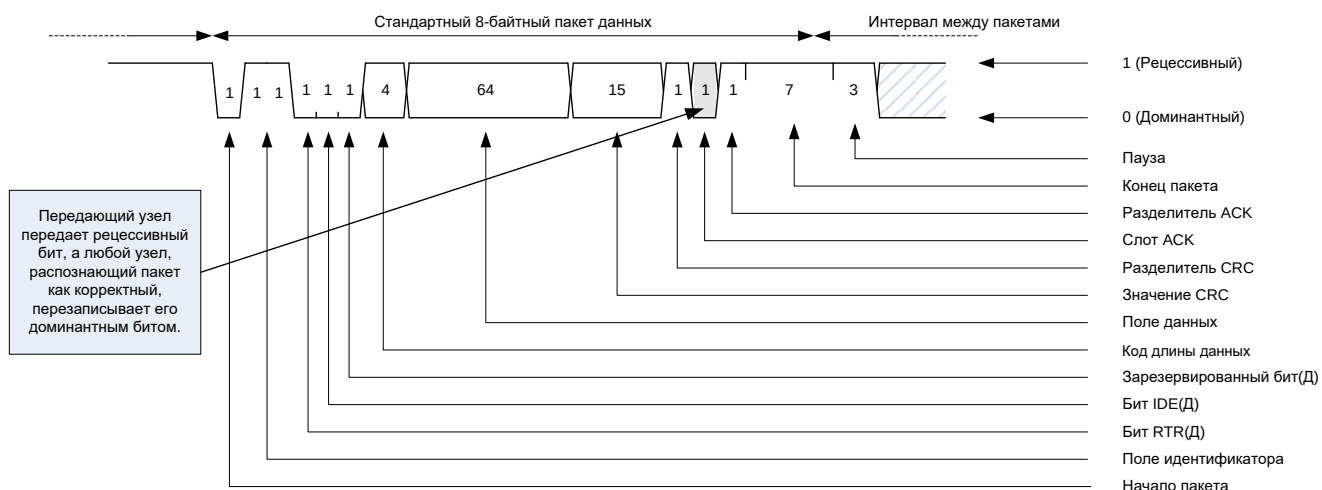


Рисунок 92 – Контроль подтверждения

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 4-х 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.

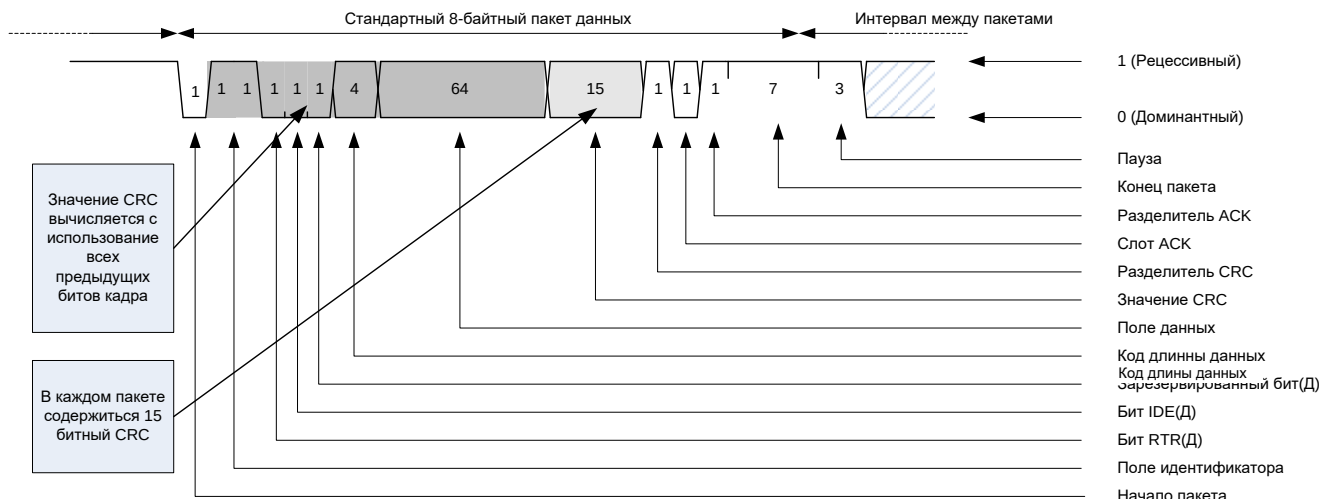


Рисунок 93 – Контроль CRC

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 шину, 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.

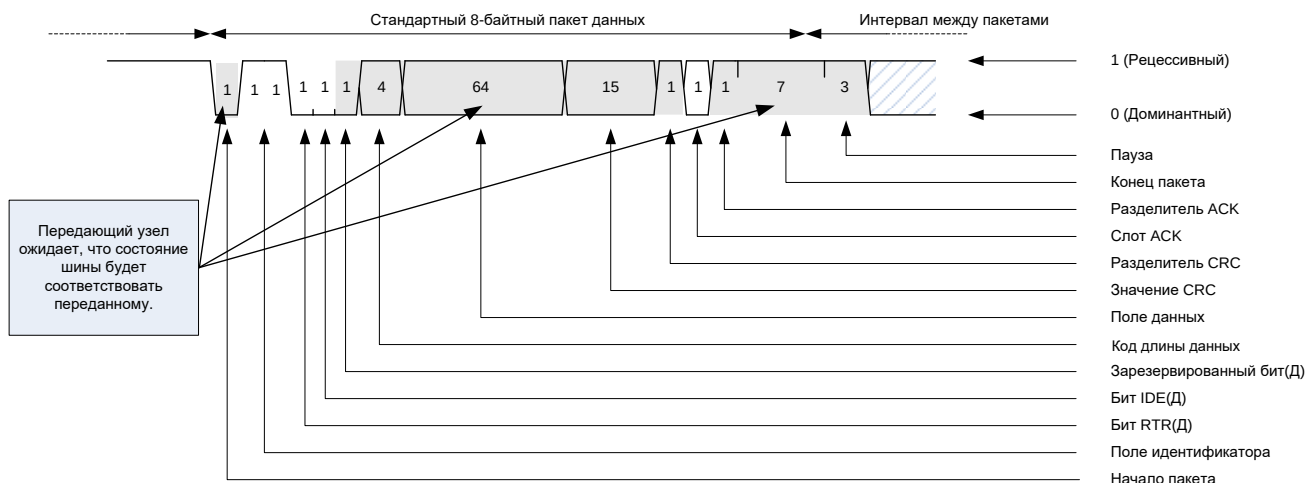


Рисунок 94 – Контроль передаваемых бит

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.

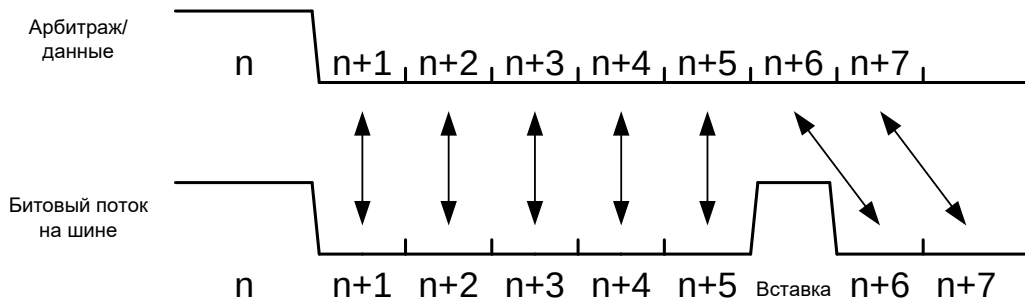


Рисунок 95 – Битстаффинг

В каждом CAN контроллере 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.

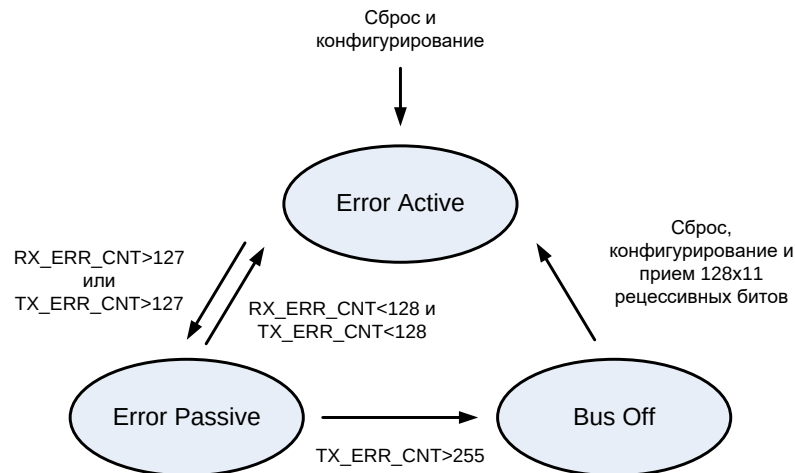


Рисунок 96 – Счетчики ошибок

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.

6.14.15 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.

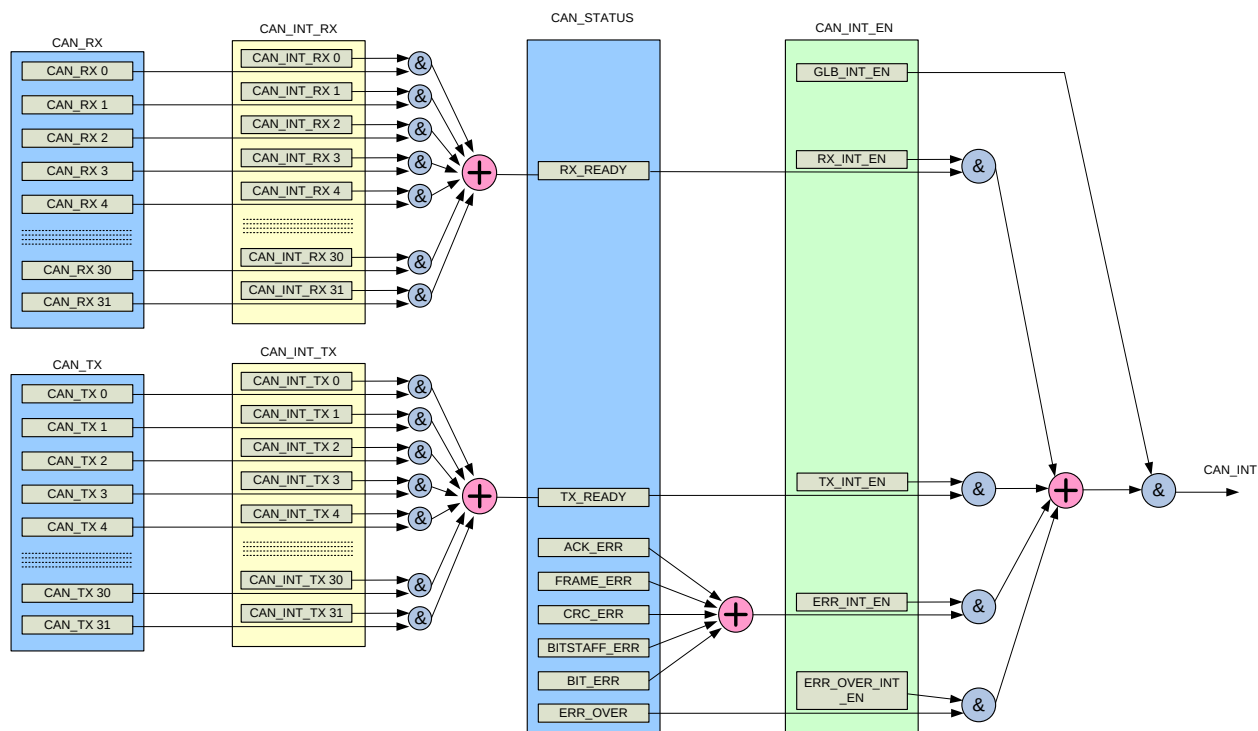


Рисунок 97 – Схема формирования прерывания блока CAN

6.14.16 Описание регистров

Таблица 110 – Описание регистров

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x4009_0000		MDR_CAN0		
0x4009_1000		MDR_CAN1		

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
	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	RXDATAH		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
	0x0000_0800	FILTER[0].MASK		Маска для приема сообщения в буфер 00
	0x0000_0804	FILTER[0].FILTER		Фильтр для приема сообщения в буфер 00
	0x0000_0808	FILTER[1].MASK		Маска для приема сообщения в буфер 01

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
		...		
	0x0000_08FC	FILTER[31].FILTER		Фильтр для приема сообщения в буфер 31

6.14.16.1 Регистр CONTROL

Base ADDR=		0x4009_0000		Offset=		0x0000_0000									
REG Name:		0x4009_1000													
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

Бит	Имя	Значение	Описание
31...5	-		Зарезервировано
4	ROP		Прием собственных пакетов (Receiveownpackets): 1 – контроллер принимает собственные пакеты; 0 – контроллер принимает только чужие пакеты
3	SAP		Подтверждение собственных пакетов (SendACKownpackets): 1 – контроллер подтверждает прием собственных пакетов; 0 – контроллер подтверждает прием только чужих пакетов
2	STM		Режим самотестирования (SelfTestMode): 1 – контроллер работает в режиме самотестирования; 0 – контроллер работает в нормальном режиме
1	ROM		Режим «Только прием» (ReadOnlyMode): 1 – контроллер работает только на прием; 0 – контроллер работает в нормальном режиме
0	CAN_EN		Режим работы контроллера CAN: 1 – разрешение работы; 0 – сброс

6.14.16.2 Регистр STATUS

Base ADDR=	0x4009_0000	Offset=	0x0000_0004												
REG Name:	0x4009_1000														
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

Бит	Имя	Значение	Описание
31...24	TXERRCNT [7:0]		Счетчик ошибок передатчика TEC, биты [7:0]: TEC > 127, ERROR PASSIVE
23...16	RXERRCNT [7:0]		Счетчик ошибок приемника REC, биты [7:0]: REC > 127, ERROR PASSIVE
15...13	-		
12	TXERRCNT8		Счетчик ошибок передатчика TEC, бит 8: 0 – TEC менее 255; 1 – TEC более 255
11	RXERRCNT8		Счетчик ошибок приемника REC, бит 8: 0 – REC менее 255; 1 – REC более 255
10...9	ERRSTATUS[1:0]		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	IDLOWER		Флаг «проигрыша» арбитража: 0 – при передаче не было проигрыша арбитража; 1 – при передаче был проигран арбитраж
7	ACKERR		Флаг ошибки подтверждения приема: 0 – нет ошибки; 1 – есть ошибка
6	FRAMEERR		Флаг ошибки формата пакета: 0 – нет ошибки; 1 – есть ошибка
5	CRCERR		Флаг ошибки контрольной суммы пакета: 0 – нет ошибки; 1 – есть ошибка
4	BITSTUFF ERR		Флаг ошибки вставленных бит пакета: 0 – нет ошибки; 1 – есть ошибка
3	BITERR		Флаг ошибки передаваемых бит пакета: 0 – нет ошибки; 1 – есть ошибка
2	ERROROVER		Флаг превышения TEC и REC уровня, заданного ERROR_MAX: 0 – ERROR_MAX < TEC и REC; 1 – ERROR_MAX ≥ TEC или REC

Бит	Имя	Значение	Описание
1	TXREADY		Флаг наличия буферов для отправки: 0 – нет буферов готовых для отправки сообщений; 1 – есть буфер готовый для отправки сообщений
0	RXREADY		Флаг наличия принятых сообщений: 0 – нет буферов с принятыми сообщениями; 1 – есть буфер с принятым сообщением

6.14.16.3 Регистр BITTMNG

Base ADDR=	0x4009_0000	Offset=	0x0000_0008												
	0x4009_1000														
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]															

Бит	Имя	Значение	Описание
31...28	-		Зарезервировано
27	SB		Семплирование: 0 – однократное; 1 – трехкратное с мажоритарным контролем
26...25	SJW [1:0]		Значение размера фазы SJW: 11 = Synchronization jump width time = 4 x TQ 10 = Synchronization jump width time = 3 x TQ 01 = Synchronization jump width time = 2 x TQ 00 = Synchronization jump width time = 1 x 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]		Значение размера фазы SEG2: 111 = Phase Segment 2 time = 8 x TQ 110 = Phase Segment 2 time = 7 x TQ 101 = Phase Segment 2 time = 6 x TQ 100 = Phase Segment 2 time = 5 x TQ 011 = Phase Segment 2 time = 4 x TQ 010 = Phase Segment 2 time = 3 x TQ 001 = Phase Segment 2 time = 2 x TQ 000 = Phase Segment 2 time = 1 x TQ SEG2 – это время, используемое для сокращения битового интервала при подстройке
21...19	SEG1 [2:0]		Значение размера фазы SEG1: 111 = Phase Segment 1 time = 8 x TQ 110 = Phase Segment 1 time = 7 x TQ 101 = Phase Segment 1 time = 6 x TQ 100 = Phase Segment 1 time = 5 x TQ 011 = Phase Segment 1 time = 4 x TQ 010 = Phase Segment 1 time = 3 x TQ 001 = Phase Segment 1 time = 2 x TQ 000 = Phase Segment 1 time = 1 x TQ SEG1 – это время, используемое для увеличения битового интервала при подстройке

Бит	Имя	Значение	Описание
18...16	PSEG[2:0]		Значение размера фазы PSEG 111 = Propagation time = 8 x TQ 110 = Propagation time = 7 x TQ 101 = Propagation time = 6 x TQ 100 = Propagation time = 5 x TQ 011 = Propagation time = 4 x TQ 010 = Propagation time = 3 x TQ 001 = Propagation time = 2 x TQ 000 = Propagation time = 1 x TQ PSEG - это время компенсирующее задержку распространения сигналов в шине CAN
15...0	BRP [15:0]		Предделитель системной частоты: $CLK = CANCLKn / (BRP + 1)$ $TQ(us) = 1 / CLK (MHz) = (BRP + 1) / CANCLKn (MHz)$

6.14.16.4 Регистр INT_EN

Base ADDR=	0x4009_0000	Offset=	0x0000_0010												
REG Name:	0x4009_1000														
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

Бит	Имя	Значение	Описание
31...5	-		Зарезервировано
4	ERROVER INTEN		Флаг разрешения прерывания по превышению TEC или REC допустимого значения в ERROR_MAX: 0 – запрещено прерывание; 1 – разрешено прерывание
3	ERRINT EN		Флаг разрешения прерывания по возникновению ошибки: 0 – запрещено прерывание; 1 – разрешено прерывание
2	TXINTEN		Флаг разрешения прерывания по возможности передачи: 0 – запрещено прерывание; 1 – разрешено прерывание
1	RXINTEN		Флаг разрешения прерывания по приему сообщений: 0 – запрещено прерывание; 1 – разрешено прерывание
0	GLBINTEN		Общий флаг разрешения прерывания блока CAN: 0 – запрещено прерывание; 1 – разрешено прерывание

6.14.16.5 Регистр OVER

Base ADDR=	0x4009_0000 0x4009_1000	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

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7...0	ERRORMAX [7:0]		Регистр границы счетчика ошибок Допустимое значение счетчиков ошибок TEC и REC, при превышении которого вырабатывается флаг ERROR_OVER

6.14.16.6 Регистр BUF_CON[x]

Base ADDR=	0x4009_0000 0x4009_1000	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

Бит	Имя	Значение	Описание
31...8	-		Зарезервировано
7	OVER_WR		Флаг перезаписи принятого сообщения: 0 – не было перезаписи; 1 – была перезапись принятого сообщения
6	RX_FULL		Флаг готовности приема: 0 – нет принятого сообщения; 1 – принятое сообщение в буфере
5	TX_REQ		Запрос на отправку сообщения: 0 – нет запроса или отправлено; 1 – запрос на отправку
4	PRIOR_0		Приоритет при отправке: 0 – приоритет; 1 – нет приоритета
3	RTR_EN		Режим ответа на RTR: 0 – не отвечать при приеме RTR; 1 – ответить при приеме RTR в буфер
2	OVER_EN		Разрешение перезаписи принятого сообщения: 0 – не разрешена перезапись; 1 – разрешена перезапись сообщения

Бит	Имя	Значение	Описание
1	RX_TXn		Режим работы буфера: 0 – на передачу; 1 – на прием
0	EN		Разрешение работы буфера: 0 – отключен; 1 – работает

6.14.16.7 Регистр INT_RX

Base ADDR=	0x4009_0000 0x4009_1000	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]															

Бит	Имя	Значение	Описание
31...0	CAN_INT_RX[31:0]		Флаги разрешения прерываний от буферов по приему сообщений: CAN_INT_RX[0] – для первого буфера CAN_INT_RX[1] – для второго буфера и так далее

6.14.16.8 Регистр RX

Base ADDR=	0x4009_0000 0x4009_1000	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]															

Бит	Имя	Значение	Описание
31...0	CAN_RX[31:0]		Флаги RX_FULL разрешенных на прием буферов: CAN_RX[0] – флаг RX_FULL от первого буфера

6.14.16.9 Регистр INT_TX

Base ADDR=		0x4009_0000 0x4009_1000				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]															

Бит	Имя	Значение	Описание
31...0	CAN_INT_TX[31:0]		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

6.14.16.10 Регистр TX

Base ADDR=		0x4009_0000 0x4009_1000				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]															

Бит	Имя	Значение	Описание
31...0	CAN_TX[31:0]		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

6.14.16.11 Регистр RXID

6.14.16.12 Регистр TXID

6.14.16.13 Регистр CAN_BUF[x].ID

6.14.16.14 Регистр CAN_BUF_FILTER[x].MASK

6.14.16.15 Регистр CAN_BUF_FILTER[x].FILTER

Base ADDR=		0x4009_0000 0x4009_1000				Offset=		0x0000_0020 0x0000_0030 0x0000_0200 0x0000_0210 .. 0x0000_03F0 0x0000_0800 0x0000_0804 0x0000_0808 .. 0x0000_08FC									
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]																	

Бит	Имя	Значение	Описание
31...29	-		Зарезервировано
28...18	SID [10:0]		Поле SID. Для стандартного и расширенного пакетов CAN. Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже
17...0	EID [17:0]		Поле EID. Для расширенных пакетов CAN. Чем меньше значение поля, тем больший приоритет имеет пакет при арбитраже

6.14.16.16 Регистр RXDLC

6.14.16.17 Регистр TXDLC

6.14.16.18 Регистр CAN_BUF[x].DLC

Base ADDR=	0x4009_0000 0x4009_1000	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	SSR	R0	R1	RTR								

Бит	Имя	Значение	Описание
31...13	-		Зарезервировано
12	IDE		Поле IDE. Поле, обозначающее формат пакета: 0 – стандартный пакет; 1 – расширенный пакет
11	SSR		Поле SSR, расширенного формата. Всегда должно быть равно “1”
10	R0		Поле R0. Всегда должно быть равно “0”
9	R1		Поле R1, расширенного формата. Всегда должно быть равно “1”
8	RTR		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	-		Зарезервировано
3...0	DLC[3:0]		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 0110 – 6 байт 0111 – 7 байт 1000 – 8 байт 1xxx – 8 байт и недопустимо

6.14.16.19 Регистр RXDATAL

6.14.16.20 Регистр TXDATAL

6.14.16.21 Регистр CAN_BUF[x].DATAL

Base ADDR=		0x4009_0000 0x4009_1000					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]							

Бит	Имя		Значение		Описание										
31...24	DB3[7:0]				Поле DB3. Четвертый байт, передаваемый в пакете										
23...16	DB2[7:0]				Поле DB2. Третий байт, передаваемый в пакете										
15...8	DB1[7:0]				Поле DB1. Второй байт, передаваемый в пакете										
7...0	DB0[7:0]				Поле DB0. Первый байт, передаваемый в пакете										

6.14.16.22 Регистр RXDATAH

6.14.16.23 Регистр TXDATAH

6.14.16.24 Регистр CAN_BUF[x].DATAH

Base ADDR=		0x4009_0000 0x4009_1000					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]							

Бит	Имя		Значение		Описание										
31...24	DB7[7:0]				Поле DB7. Четвертый байт, передаваемый в пакете										
23...16	DB6[7:0]				Поле DB5. Третий байт, передаваемый в пакете										
15...8	DB5[7:0]				Поле DB4. Второй байт, передаваемый в пакете										
7...0	DB4[7:0]				Поле DB3. Первый байт, передаваемый в пакете										

6.14.16.25 Регистр BUF[x]

Base ADDR=	0x4009_0000 0x4009_1000	Offset=	0x0000_0200 0x0000_0210 .. 0x0000_03F0												
REG Name:															
127															112
DATAH[31:0]															
111															96
95															80
DATAH[31:0]															
79															64
63															48
DLC[31:0]															
47															32
31															16
ID[31:0]															
15															0

Бит	Имя	Значение	Описание
127...96	DATAH[31:0]		Данные сообщения
95...64	DATAH[31:0]		Данные сообщения
63...32	DLC[31:0]		DLC сообщения
31...0	ID[31:0]		ID сообщения

6.14.16.26 Регистр FILTER[x]

Base ADDR=	0x4009_0000 0x4009_1000	Offset=	0x0000_0500 0x0000_0508 .. 0x0000_05F8												
REG Name:															
63															48
FILTER[31:0]															
47															32
31															16
MASK[31:0]															
15															0

Бит	Имя	Значение	Описание
63...32	FILTER[31:0]		Фильтр для приёма сообщения
31...0	MASK[31:0]		Маска для приёма сообщения

6.15 Контроллер SSP (SSPx)

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

- интерфейс SPI фирмы Motorola;
- интерфейс SSI фирмы Texas Instruments;
- интерфейс Microwire фирмы National Semiconductor.

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

- преобразование данных, размещенных во внутреннем буфере FIFO передатчика (восемь 32-разрядных ячеек данных) из параллельного в последовательный формат;
- преобразование данных из последовательного в параллельный формат и их запись в аналогичный буфер FIFO приемника (восемь 32-разрядных ячеек данных).

Модуль формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов FIFO приемника и передатчика;
- переполнение буфера FIFO приемника;
- наличие данных в буфере FIFO приемника по истечении времени таймаута;
- наличие данных в FIFO приемника;
- отсутствие данных в FIFO передатчика;
- отсутствие данных в сдвиговом регистре передатчика.

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 32 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 32 бит;
- независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по переполнению буфера приемника, по наличию данных в FIFO приемника, по отсутствию данных в FIFO передатчика, а также по отсутствию данных в сдвиговом регистре передатчика;
- доступна возможность тестирования по шлейфу, соединяющему вход с выходом;
- поддержка прямого доступа к памяти (DMA).

Структурная схема модуля представлена далее – см. рисунок 98.

6.15.1 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

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

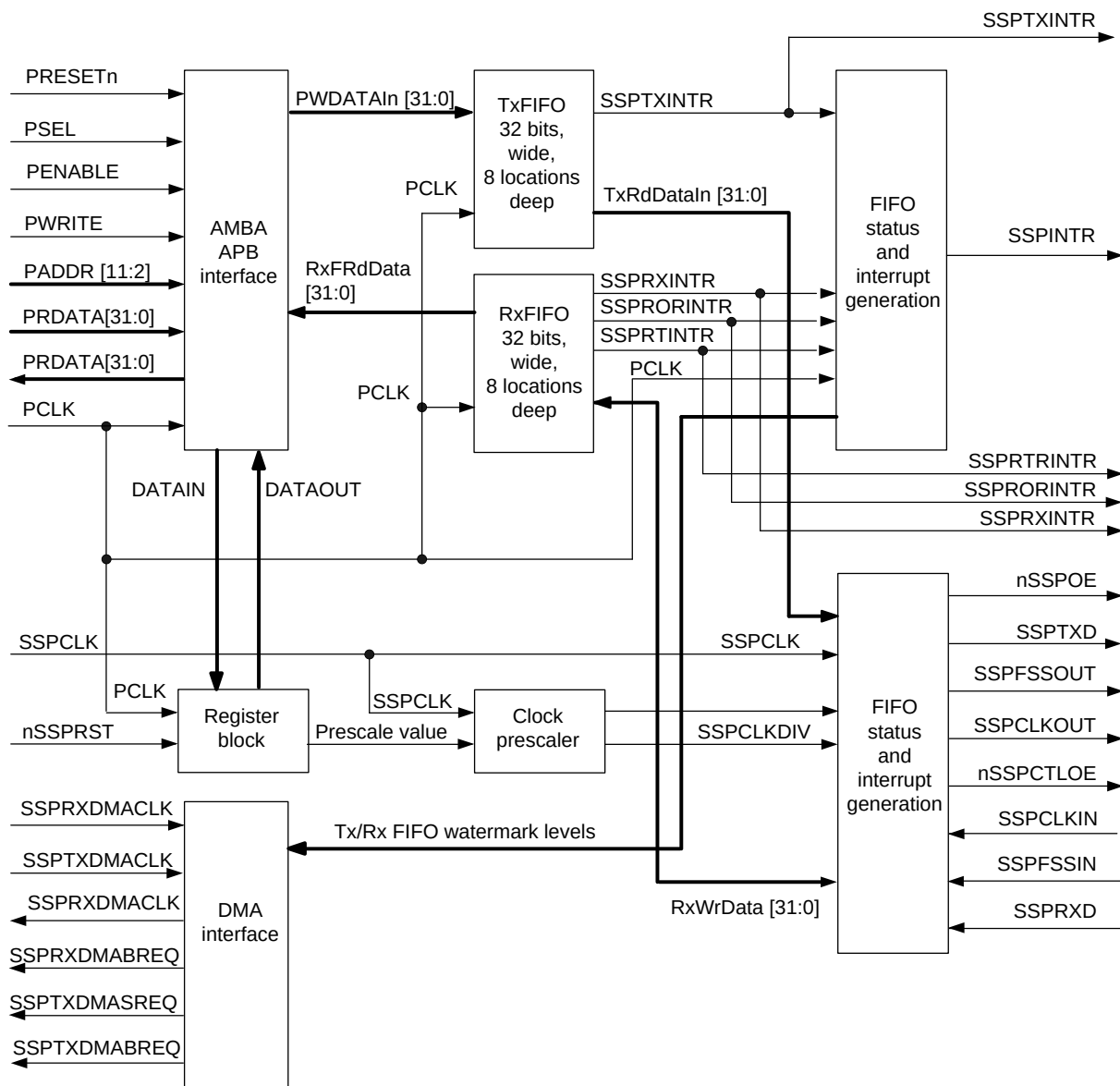


Рисунок 98 – Структурная схема модуля SSP

6.15.1.1 Характеристики интерфейса SPI

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:

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

6.15.1.2 Характеристики интерфейса Microwire

Интерфейс Microwire фирмы National Semiconductor обеспечивает полудуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

6.15.1.3 Характеристики интерфейса SSI

Интерфейс SSI фирмы Texas Instruments обеспечивает:

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

6.15.2 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 формы и последовательную;
- 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 32 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).

6.15.2.1 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.

6.15.2.2 Буфер FIFO передатчика

Буфер передатчика имеет ширину 32 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. Существует возможность программного сброса с помощью сигнала RESTxFIFO из управляющего регистра CR1.

6.15.2.3 Буфер FIFO приемника

Буфер приемника имеет ширину 32 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 в до тех пор, пока не будут считаны центральным процессором.

6.15.2.4 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, передаётся 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.

6.15.2.5 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.

6.15.2.6 Интерфейс прямого доступа к памяти

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

6.15.2.7 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.

6.15.2.8 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 32-разрядных слов заблаговременно перед установкой этого бита, либо может заполняться передаваемыми данными в процедуре обслуживания прерывания.

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

6.15.2.9 Соотношения между тактовыми сигналами

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

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

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

установки и удержания сигнала SSP_RXD по отношению к сигналу SSP_CLK должно выбираться с запасом, гарантирующим правильное считывание данных. Для обеспечения корректной работы устройства необходимо, чтобы частота SSPCLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSP_CLK.

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

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

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

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

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

$$\begin{cases} F_{SSPCLK_{min}} \geq 2 \times F_{SSPCLKOUT_{max}} (master\ mode) \\ F_{SSPCLK_{min}} \geq 12 \times F_{SSPCLKIN_{max}} (slave\ mode) \end{cases} \quad (12)$$

При активации сигнала SSPFRX в регистре управления CR0, это соотношение приобретает вид:

$$\begin{cases} F_{SSPCLK_{min}} \geq 2 \times F_{SSPCLKOUT_{max}} (master\ mode) \\ F_{SSPCLK_{min}} \geq 4 \times F_{SSPCLKIN_{max}} (slave\ mode) \end{cases} \quad (13)$$

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

$$\begin{cases} F_{SSPCLK_{max}} \leq 254 \times 256 \times F_{SSPCLKOUT_{min}} (master\ mode) \\ F_{SSPCLK_{max}} \leq 254 \times 256 \times F_{SSPCLKIN_{min}} (slave\ mode). \end{cases} \quad (14)$$

6.15.2.10 Программирование регистра управления CR0

Регистр CR0 предназначен для:

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

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.

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

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

6.15.2.11 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;
- программного сброса FIFO передатчика.

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).

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

6.15.2.12 Формирование тактового сигнала обмена данными

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

Частота выходного тактового сигнала обмена данными SSP_CLK определяется следующим соотношением:

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \times (1 + SCR)}. \quad (15)$$

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.

6.15.2.13 Формат информационного кадра

Каждый информационный кадр содержит в зависимости от запрограммированного значения от 4 до 32 бит данных. Передача данных начинается со старшего значащего разряда. 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 32 бит, таким образом общая длительность информационного кадра составляет от 13 до 41 бит.

6.15.2.14 Формат синхронного обмена SSI фирмы Texas Instruments

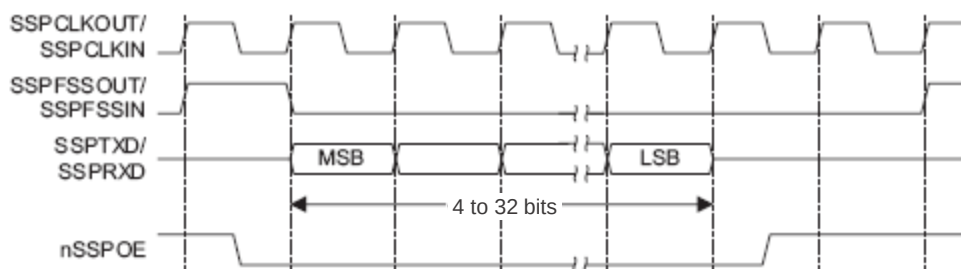


Рисунок 99 – Формат синхронного обмена протокола SSI (единичный обмен)

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 – 32 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 99 – передача единичного информационного кадра, Рисунок 100 – передача последовательности кадров.

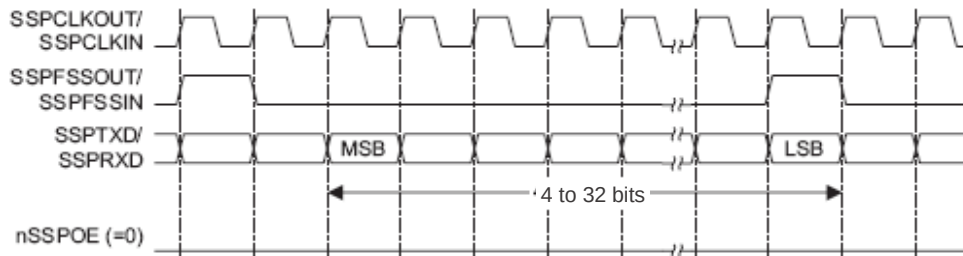


Рисунок 100 – Формат синхронного обмена протокола SSI (непрерывный обмен)

6.15.2.15 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.

6.15.2.16 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 101, 102.

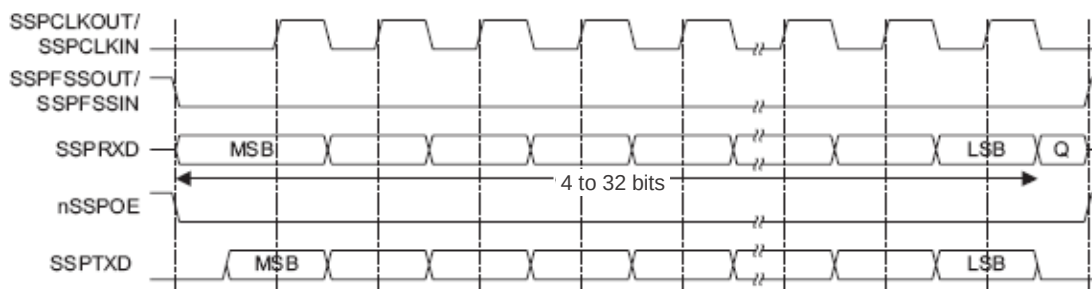


Рисунок 101 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (одиначный обмен)

Примечание – На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

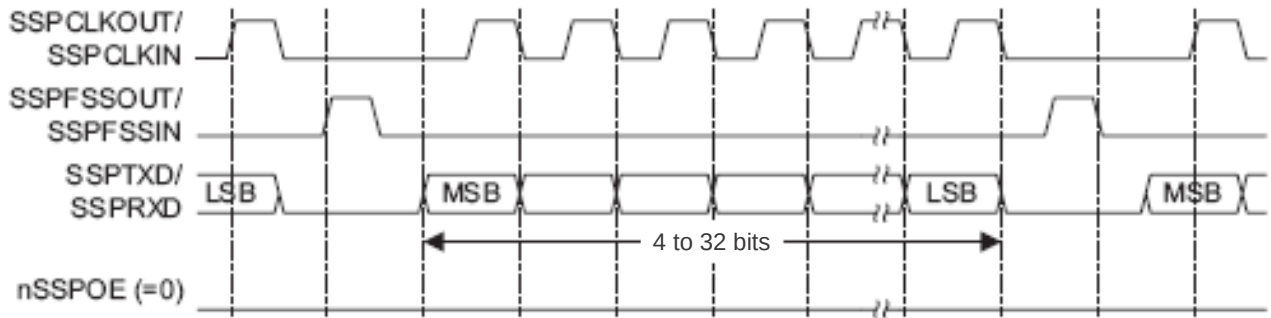


Рисунок 102 – Формат синхронного обмена протокола SPI, SPO=0, SPH=0 (непрерывный обмен)

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.

6.15.2.17 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 103 – одиночный и непрерывный обмен.

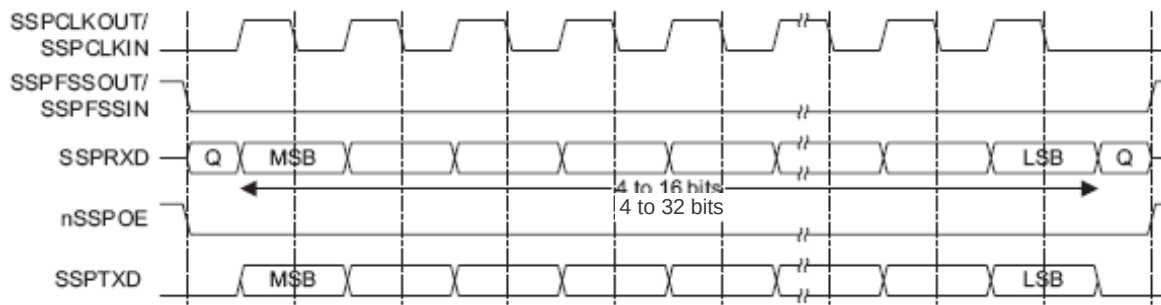


Рисунок 103 – Формат синхронного обмена протокола SPI, SPO=0, SPH=1

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.

6.15.2.18 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 104, 105.

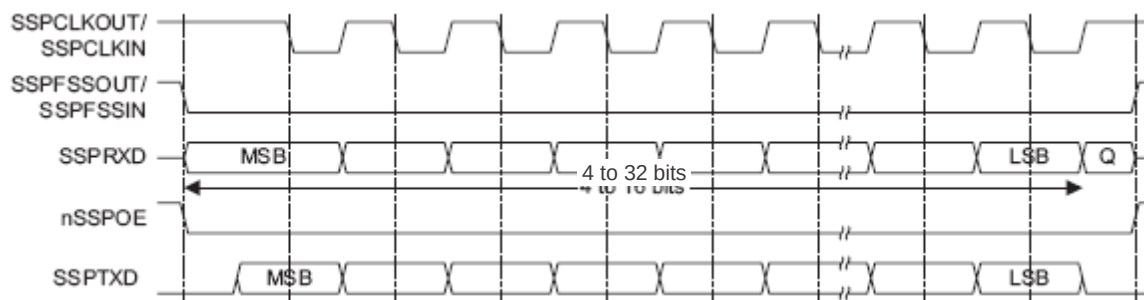


Рисунок 104 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0
(одиночный обмен)

Примечание – На рисунке буквой Q обозначен сигнал с неопределенным уровнем.

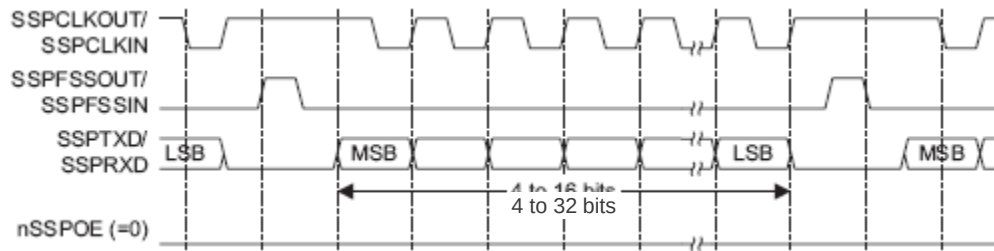


Рисунок 105 – Формат синхронного обмена протокола SPI, SPO=1, SPH=0 (непрерывный обмен)

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.

6.15.2.19 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 106 – одиночный и непрерывный обмен.

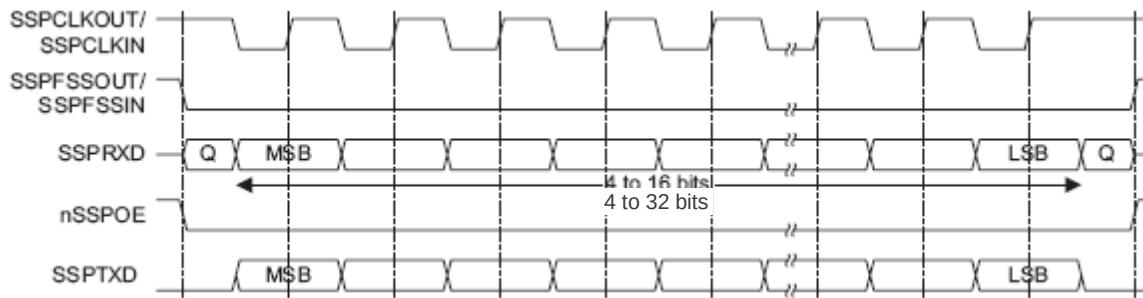


Рисунок 106 – Формат синхронного обмена протокола SPI, SPO=1, SPH=1

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 передатчика SSP_TXD переходит из высокоимпедансного в активное состояние.

По истечении полутакта сигнала SSP_CLK, на линиях обмена как ведущего, так и ведомого устройств сформированы значения первых бит передаваемых данных. В это же время включается линия SSP_CLK и на ней формируется передний фронт сигнала. Далее, данные регистрируются по переднему фронту и выдаются в линию по заднему фронту сигнала SSP_CLK.

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

В режиме непрерывной передачи данных, линия SSP_FSS постоянно находится в низком логическом уровне и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

6.15.2.20 Формат синхронного обмена Microwire фирмы NationalSemiconductor

Временные диаграммы последовательного синхронного обмена в режиме Microwire показаны на рисунках 107, 108.

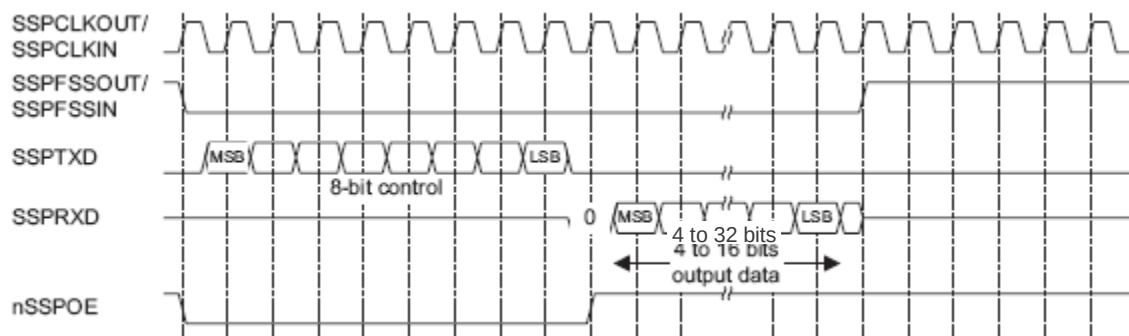


Рисунок 107 – Формат синхронного обмена протокола Microwire (одиначный обмен)

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 32 бит, таким образом, общая длительность информационного кадра составляет от 13 до 41 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.

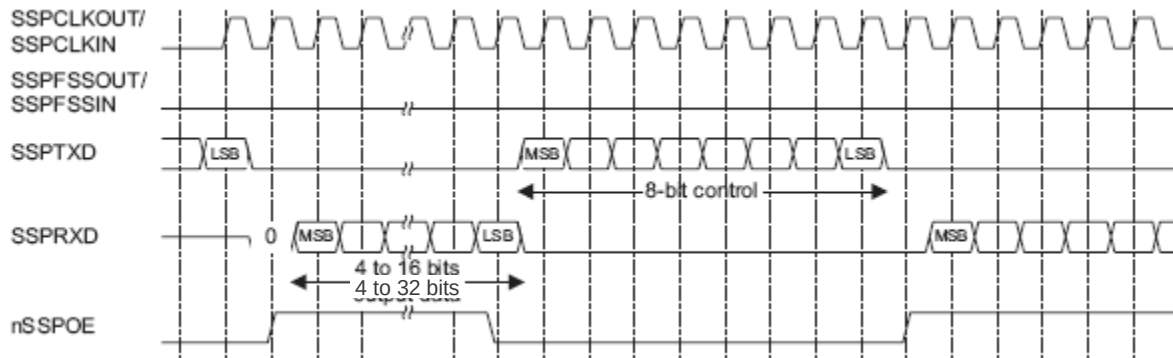


Рисунок 108 – Формат синхронного обмена протокола Microwire (непрерывный обмен)

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_CLK, должны гарантировать достаточное время установки и удержания сигнала SSP_FSS, по отношению к переднему фронту сигнала SSP_CLK.

Данные требования иллюстрирует рисунок 109. 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.

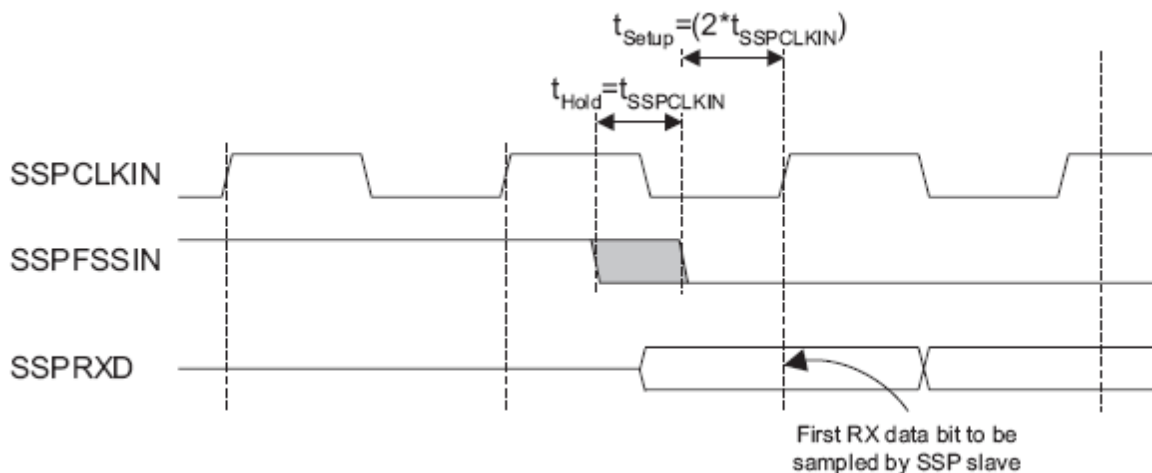


Рисунок 109 – Формат Microwire, требования к времени установки и удержания сигнала

6.15.2.21 Примеры конфигурации модуля в ведущем и ведомом режимах

На рисунках 110 – 112 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 приёмопередатчик должен быть изначально сконфигурирован в одном из этих режимов.

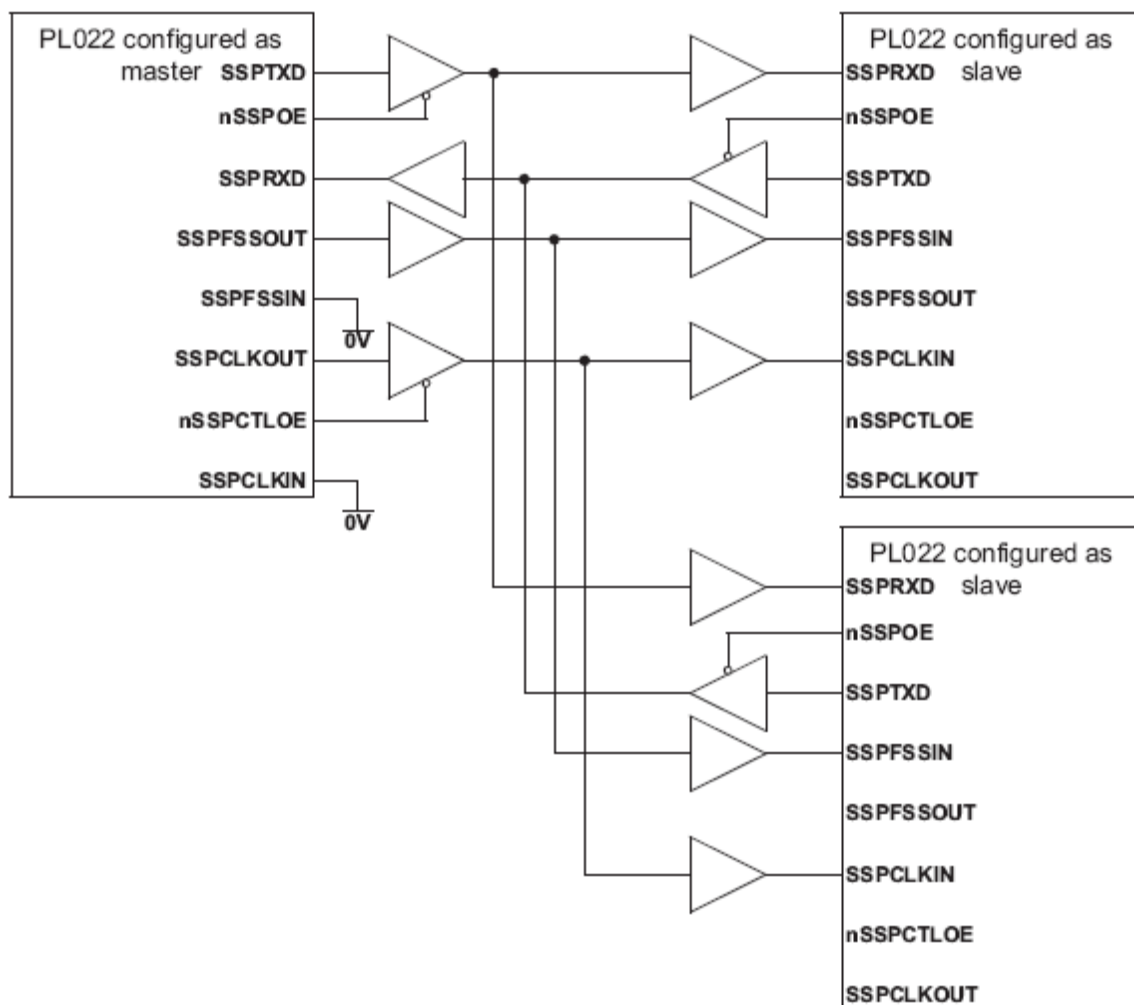


Рисунок 110 – Ведущее устройство SSP подключено к двум ведомым

Рисунок 110 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.

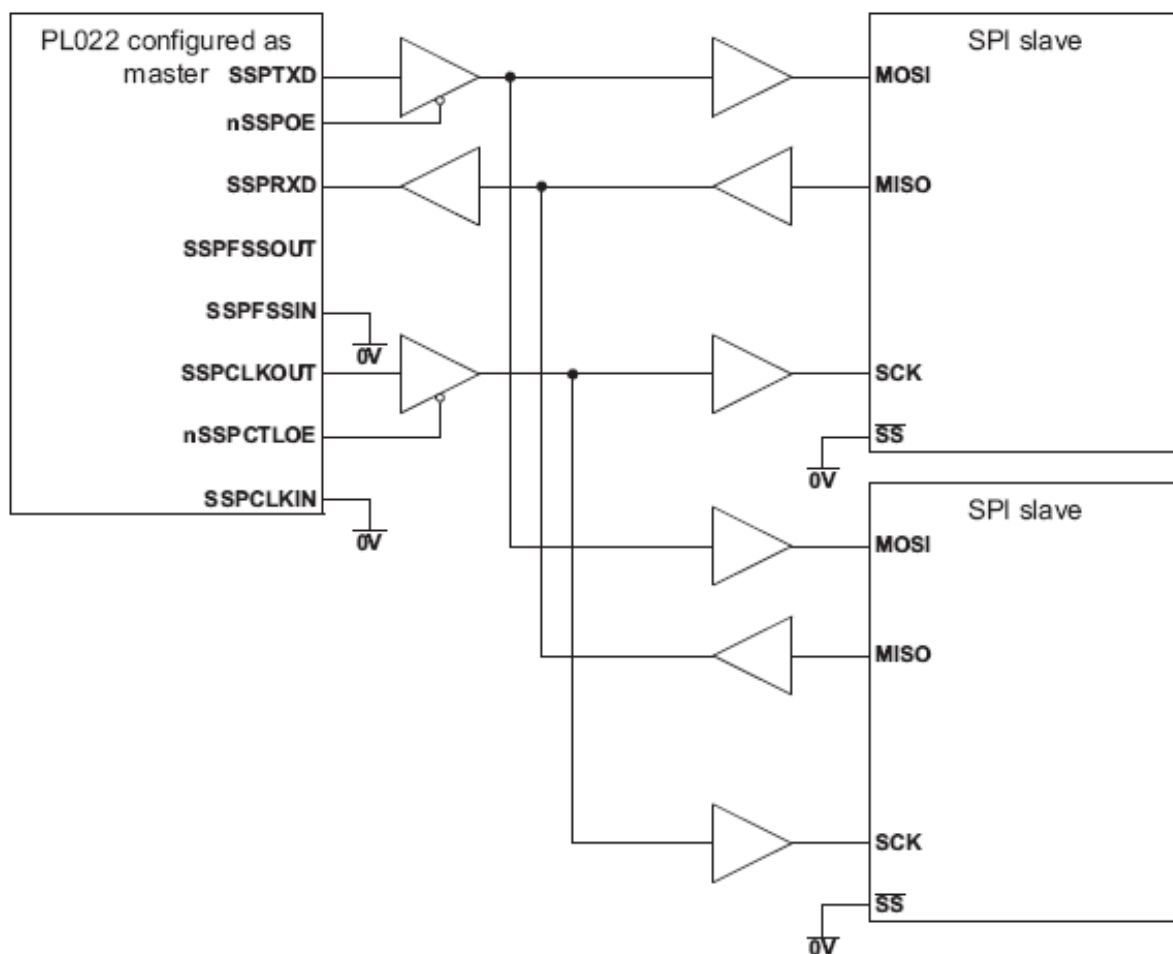


Рисунок 111 – Ведущее устройство SSP подключено к двум ведомым, поддерживающим SPI

Рисунок 111 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.

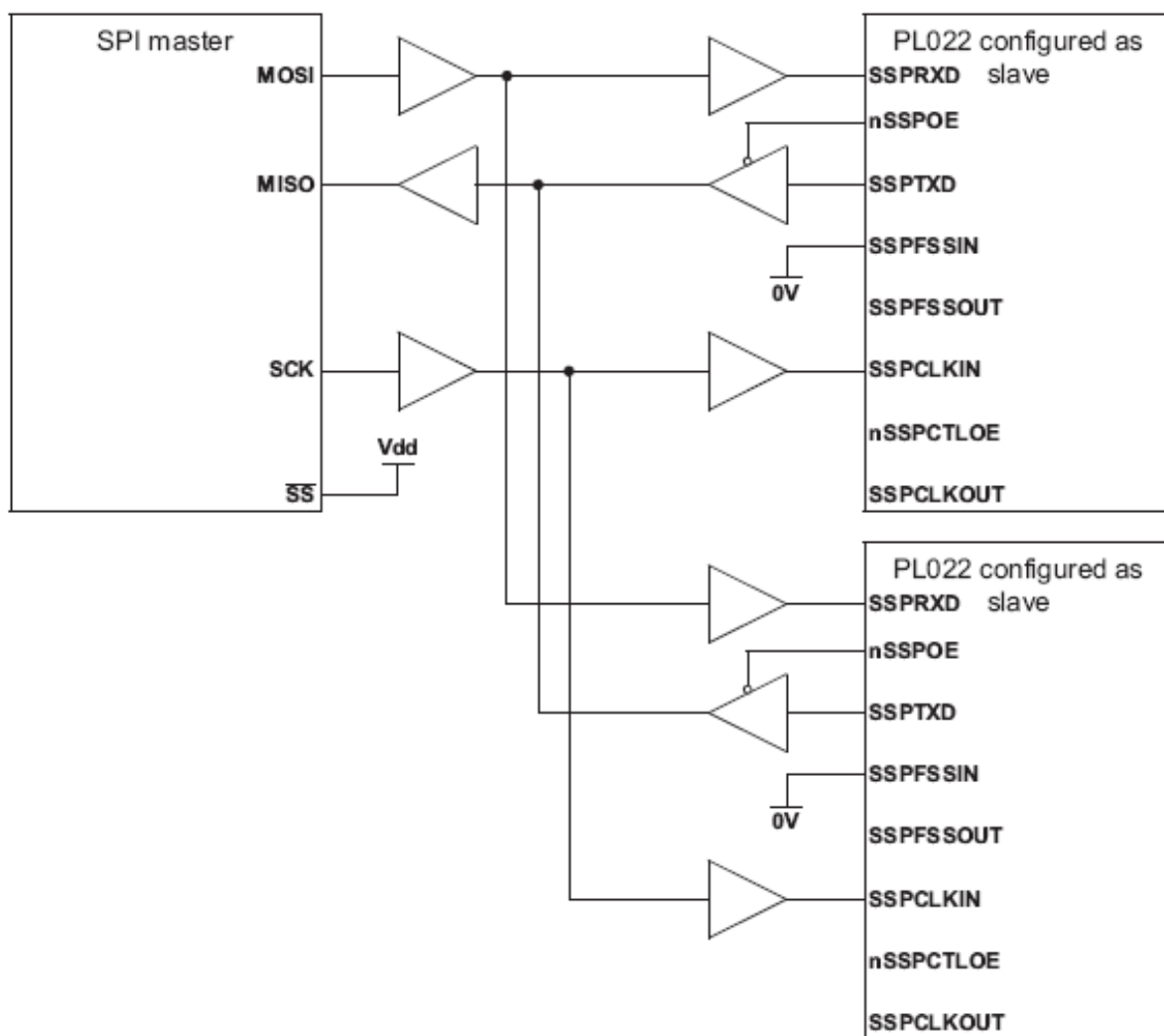


Рисунок 112 – Ведущее устройство, протокол SPI, подключено к двум ведомым модулям SSP

Рисунок 112 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.

6.15.3 Интерфейс прямого доступа к памяти

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

Интерфейс DMA включает в себя следующие сигналы:

Для приема:

- SSPRXDMASREQ – запрос передачи отдельного символа, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит по меньшей мере один символ;
- SSPRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если буфер FIFO приемника содержит четыре или более символов;
- SSPRXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Для передачи:

- SSPTXDMASREQ – запрос передачи отдельного символа, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит по меньшей мере одну свободную ячейку;
- SSPTXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переводится в активное состояние в случае, если буфер FIFO передатчика содержит четыре или менее символов;
- SSPTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае, если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

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

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

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

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

В таблице 111 приведены значения порогов заполнения буферов приемника и передатчика, необходимых для срабатывания запросов блочного обмена DMABREQ.

Таблица 111 – 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)
1/2	4	4

Рисунок 113 показывает временные диаграммы одноэлементного и блочного запросов DMA, в том числе действие сигнала DMACLR. Все сигналы должны быть синхронизированы с PCLK.

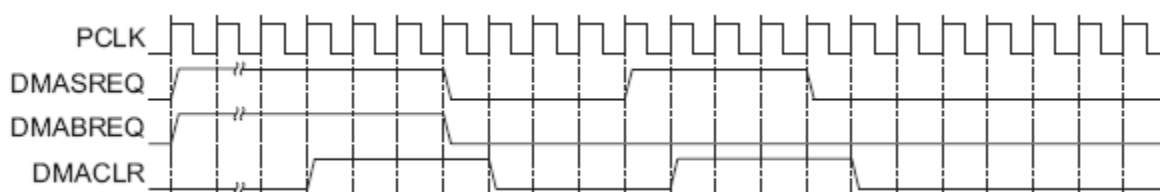


Рисунок 113 – Временные диаграммы обмена в режиме DMA

6.15.4 Прерывания

В модуле предусмотрено семь маскируемых линий запроса на прерывание с выводом на один общий сигнал, представляющий собой комбинацию независимых по схеме ИЛИ.

Сигналы запроса на прерывание:

- SSPRXINTR – запрос на обслуживание буфера FIFO приемника;
- SSPTXINTR – запрос на обслуживание буфера FIFO передатчика;
- SSPRORINTR – переполнение буфера FIFO приемника;
- SSPRTINTR – таймаут приемника;
- SSPRNEINTR – наличие данных в FIFO приемника;
- SSPTFEINTR – отсутствие данных в FIFO передатчика;
- SSPTNBSYINTR – отсутствие данных в сдвиговом регистре передатчика;
- SSPINTR – логическое ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRTINTR, SSPRORINTR, SSPRNEINTR, SSPTFEINTR и SSPTNBSYINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSPIMSC. Установка бита в 1 разрешает соответствующее прерывание, а в 0 – запрещает.

Доступность индивидуальных линий, и общей линии запроса, позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйвера устройства, построенного по модульному принципу.

Прерывания от приемника и передатчика SSPRXINTR и SSPTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать данные сигналы запроса для обеспечения чтения и записи данных согласованно с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний SSPRIS, либо из маскированного регистра прерываний SSPMIS.

6.15.4.1 SSPRXINTR

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

6.15.4.2 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.

6.15.4.3 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.

6.15.4.4 SSPRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. 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.

6.15.4.5 SSPRNEINTR

Прерывание по наличию данных в FIFO приемника. Сигнал формируется при записи хотя бы одного слова в FIFO приемника и сохраняется до тех пор, пока все данные не будут считаны.

6.15.4.6 SSPTFEINTR

Прерывание по отсутствию данных в FIFO передатчика. Находится в активном уровне до тех пор, пока в FIFO не будет записано хотя бы одно слово. После передачи всех данных возвращается в активный уровень.

6.15.4.7 SSPTNBSYINTR

Прерывание по отсутствию данных в сдвиговом регистре передатчика.

6.15.4.8 SSPINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов SSPRXINTR, SSPTXINTR, SSPRNEINTR, SSPTNBSYINTR, SSPTBSYINTR, SSPRTINTR и SSPRORINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерываний, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

6.15.5 Программное управление модулем

6.15.5.1 Общая информация

В микроконтроллере реализовано два модуля SSP, базовые адреса каждого модуля указаны в таблице 112. 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.

6.15.6 Описание регистров

Данные о регистрах модуля SSP приведены в таблице 112.

Таблица 112 – Обобщенные данные о регистрах модуля SSP

Базовый адрес	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x4009_5000	MDR_SSP0				Контроллер SSP0
0x4009_6000	MDR_SSP1				Контроллер SSP1
0x4009_7000	MDR_SSP2				Контроллер SSP2
0x4009_8000	MDR_SSP3				Контроллер SSP3
0x400B_1000	MDR_SSP4				Контроллер SSP4
0x400B_2000	MDR_SSP5				Контроллер SSP5
Смещение					
0x000	CR0	RW	0x000000	18	Регистр управления 0
0x004	CR1	RW	0x00	5	Регистр управления 1
0x008	DR	RW	0x----	32	Буфера FIFO приемника (чтение) Буфер FIFO передатчика (запись)
0x00C	SR	RO	0x03	3	Регистр состояния
0x010	CPSR	RW	0x00	8	Регистр делителя тактовой частоты
0x014	IMSC	RW	0x00	7	Регистр маски прерывания
0x018	RIS	RO	0x68	7	Регистр состояния прерываний без учета маскирования
0x01C	MIS	RO	0x00	7	Регистр состояния прерываний с учетом маскирования
0x020	ICR	WO	0x0	4	Регистр сброса прерывания
0x024	DMACR	RW	0x0	2	Регистр управления прямым доступом к памяти
0x80	TCR	RW	0x0	2	Зарезервировано
0x84	ITIP	RW	0x0	2	Зарезервировано
0x88	ITOP	RW	0x0	14	Зарезервировано
0x8C	TDR	RO	0x00000000	32	Зарезервировано
Примечание – В столбце «тип» указан вид доступа к регистру: RW – чтение и запись; RO – только чтение; WO – только запись.					

6.15.6.1 Регистр CR0

Регистр управления 0

Регистр CR0 содержит семь битовых полей, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 113.

Таблица 113 – Формат регистра CR0

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...18	-	Зарезервировано
17	SSPFRX	Для ведомого модуля включает быстрый режим протокола SPI и активирует синхронизацию сигнала RXD. 1 – активация быстрого режима работы ведомого 0 – нормальный режим работы
16	DSS[4]	Старший разряд размера слова данных: 0 – DSS[3:0]; 1 – DSS[3:0] + 16 бит
15...8	SCR	Скорость последовательного обмена. Значение поля SCR используется при формировании тактового сигнала обмена данными. Информационная скорость удовлетворяет соотношению $F_{SSPCLK} / (CPSDVR * (1 + SCR))$, где CPSDVR – 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. раздел «Формат SPI фирмы Motorola»
6	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. подраздел «Формат синхронного обмена SPI фирмы Motorola»
5...4	FRF	Формат информационного кадра. 00 – протокол SPI фирмы Motorola; 01 – протокол SSI фирмы Texas Instruments; 10 – протокол Microwire фирмы National Semiconductor; 11 – резерв
3...0	DSS[3:0]	Младшие 4 разряда размера слова данных. При DSS[4] = 0: 0000 – резерв 0001 – резерв 0010 – резерв 0011 – 4 бита 0100 – 5 бит 0101 – 6 бит ... 1110 – 15 бит 1111 – 16 бит При DSS[4] = 1: 0000 – 17 бит 0001 – 18 бит 0010 – 19 бит 0011 – 20 бит 0100 – 21 бит 0101 – 22 бита ... 1110 – 31 бит 1111 – 32 бита

6.15.6.2 Регистр CR1

Регистр управления 1

Регистр CR1 содержит четыре битовых поля, предназначенных для управления блоками модуля SSP. Назначение разрядов регистра представлено в таблице 114.

Таблица 114 – Регистр CR1

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...5		Резерв, при чтении результат не определен. При записи следует устанавливать в 0
4	RESTxFIFO	Программный сброс FIFO передатчика: 0 – нормальный режим работы FIFO передатчика; 1 – активация сброса FIFO передатчика. Сбрасывается в ноль автоматически по завершении сброса
3	SOD	Запрет выходных линий в режиме ведомого устройства. Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. Бит SOD следует установить в случае, если данный ведомый модуль SSP не должен в настоящее время осуществлять передачу данных в линию SSP_TXD. При этом линии обмена данных ведомых устройств можно соединить параллельно. 0 – управление линией SSP_TXD в ведомом режиме разрешено; 1 – управление линией SSP_TXD в ведомом режиме запрещено
2	MS	Выбор ведущего или ведомого режима работы: 0 – ведущий модуль (устанавливается по умолчанию); 1 – ведомый модуль
1	SSE	Разрешение работы приемопередатчика: 0 – работа запрещена; 1 – работа разрешена
0	LBM	Тестирование по шлейфу: 0 – нормальный режим работы приемопередатчика; 1 – выход регистра сдвига передатчика соединен со входом регистра сдвига приемника

6.15.6.3 Регистр DR

Регистр данных

Регистр SSPDR имеет разрядность 32 бита и предназначен для чтения принятых, и записи передаваемых данных.

Операция чтения обеспечивает доступ к последней несчитанной ячейке буфера FIFO приемника. Запись данных в этот буфер FIFO осуществляет блок приемника.

Операция записи позволяет занести очередное слово в буфер FIFO передатчика. Извлечение данных из этого буфера осуществляет блок передатчика. При этом извлеченные данные помещаются в регистр сдвига передатчика, откуда последовательно выдаются на линию SSP_TXD с заданной скоростью информационного обмена.

В случае если выбран размер информационного слова менее 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 115.

Таблица 115– Формат регистра DR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...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

6.15.6.4 Регистр SR

Регистр состояния

Регистр состояния доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика, и занятости модуля SSP.

Назначение бит регистра SSPCPSR представлено в таблице 116.

Таблица 116– Регистр SR

№ 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

6.15.6.5 Регистр 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 117 отображает назначение бит регистра SSPSR.

Таблица 117 – Регистр CPSR

№ 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

6.15.6.6 Регистр IMSC

Регистр установки и сброса маски прерывания

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание. При этом запись 1 в разряд разрешает соответствующее прерывание, запись 0 – запрещает.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Назначение бит регистра IMSC показано в таблице 118.

Таблица 118– Регистр IMSC

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...7		Резерв. Не модифицируйте. При чтении выдаются нули
6	TNBSYIM	Маска прерывания по пустоте сдвигового регистра передатчика. 1 – не маскирована. 0 – маскирована
5	TFEIM	Маска прерывания по пустоте FIFO передатчика. 1 – не маскирована. 0 – маскирована
4	RNEIM	Маска прерывания по наличию данных в FIFO приемника. 1 – не маскирована. 0 – маскирована
3	TXIM	Маска прерывания по заполнению на 50% и менее буфера FIFO передатчика. 1 – не маскирована. 0 – маскирована
2	RXIM	Маска прерывания по заполнению на 50% и менее буфера FIFO приемника. 1 – не маскирована. 0 – маскирована

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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

6.15.6.7 Регистр RIS

Регистр состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Таблица 119 отображает назначение бит в регистре RIS.

Таблица 119 – Регистр RIS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31... 7		Резерв. Не модифицируйте. При чтении выдаются нули
6	TNBSYRIS	Состояние до маскирования прерывания SSPTNBSYINTR
5	TFERIS	Состояние до маскирования прерывания SSPTFEINTR
4	RNERIS	Состояние до маскирования прерывания SSPRNEINTR
3	TXRIS	Состояние до маскирования прерывания SSPTXINTR
2	RXRIS	Состояние до маскирования прерывания SSPRXINTR
1	RTRIS	Состояние до маскирования прерывания SSPRTINTR
0	RORRIS	Состояние до маскирования прерывания SSPRORINTR

6.15.6.8 Регистр MIS

Регистр маскированного состояния прерываний

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

Назначение бит в регистре SSPMIS представлено в таблице 120.

Таблица 120 – Регистр MIS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...7		Резерв. Не модифицируйте. При чтении выдаются нули
6	TNBSYMIS	Состояние маскированного прерывания SSPTNBSYINTR
5	TFEMIS	Состояние маскированного прерывания SSPTFEINTR
4	RNEMIS	Состояние маскированного прерывания SSPRNEINTR
3	TXMIS	Состояние маскированного прерывания SSPTXINTR
2	RXMIS	Состояние маскированного прерывания SSPRXINTR
1	RTMIS	Состояние маскированного прерывания SSPRTINTR
0	RORMIS	Состояние маскированного прерывания SSPRORINTR

6.15.6.9 Регистр ICR

Регистр сброса прерываний

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись в любой из разрядов регистра 0 игнорируется.

Назначение бит в регистре SSPICR представлено в таблице 121.

Таблица 121 – Регистр ICR

№ 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

6.15.6.10 Регистр 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 122.

Таблица 122 – Регистр DMACR

№ 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

6.16 Контроллер UART (UARTx)

Модуль универсального асинхронного приемопередатчика (UART – Universal Synchronous Asynchronous Receiver Transmitter) является периферийным устройством микроконтроллера.

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

В следующих подразделах представлены основные сведения о модуле:

- основные характеристики;
- программируемые параметры;
- отличия от приемопередатчика 16C650.

6.16.1 Основные характеристики модуля UART

Модуль может быть запрограммирован для использования, как в качестве универсального асинхронного приемопередатчика, так и для инфракрасного обмена данными (SIR).

Содержит независимые буферы приема (16x13) и передачи (16x9) 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.

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

Возможность организации аппаратного управления потоком данных.

Полностью программируемый асинхронный последовательный интерфейс с характеристиками:

- данные длиной 5, 6, 7, 8 или 9 бит;
- формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение, либо не передается);
- формирование 1 или 2 стоповых бит;
- скорость передачи данных – от 0 до UARTCLK/16 Бод.

Кодек ИУ обмена данными IrDA SIR обеспечивает:

- программный выбор обмена данными по линиям асинхронного приемопередатчика либо кодека ИК связи IrDA SIR;
- поддержку длительности бит для нормального режима (3/16) и для режима пониженного энергопотребления (1.41 – 2.23 мкс);
- программируемое деление опорной частоты UARTCLK для получения заданной длительности бит в режиме пониженного энергопотребления.

Наличие идентификационного регистра, однозначно идентифицирующего модуль, что позволяет операционной системе выполнять автоматическую конфигурацию.

6.16.2 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

- скорость передачи данных – целая и дробная часть числа;
- количество бит данных;
- количество стоповых бит;
- режим контроля четности;
- разрешение или запрет использования буферов FIFO (глубина очереди данных – 32 элемента или один элемент, соответственно);
- порог срабатывания прерывания по заполнению буферов FIFO (1/8, 1/4, 1/2, 3/4 и 7/8);

- частота внутреннего тактового генератора (номинальное значение – 1,8432 МГц) может быть задана в диапазоне от 1,42 до 2,12 МГц для обеспечения возможности формирования бит данных с укороченной длительностью в режиме пониженного энергопотребления;
- режим аппаратного управления потоком данных.

6.16.2.1 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.

6.16.2.2 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 формы и последовательную.

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 промышленного 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.

Поддерживаются входные линии состояния модема: «готовность к приему» (Clear To Send, CTS), «обнаружен информационный сигнал» (Data Carrier Detected, DCD), «источник данных готов» (Data Set Ready, DSR) и «индикатор вызова» (Ring Indicator, RI), а также выходные линии: «запрос на передачу» (Request to Send, RTS) и «приемник данных готов» (Data Terminal Ready, DTR).

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

Блок последовательного интерфейса инфракрасной передачи данных в соответствии с протоколом IrDA SIR реализует протокол обмена данными ENDEC. В случае его активизации обмен информацией осуществляется не с помощью сигналов UARTTXD и UARTRXD, а посредством сигналов nSIROUT и SIRIN.

В этом случае устройство переводит линию UARTTXD в пассивное состояние (высокий уровень), и перестает реагировать на изменение состояния модема, а также сигнала на линии UARTRXD. Протокол SIR ENDEC обеспечивает возможность обмена данными исключительно в режиме полудуплекса, то есть он не может передавать во время приема данных и принимать во время передачи данных.

В соответствии со спецификацией физического уровня протокола IrDA SIR, задержка между передачей и приемом должна составлять не менее 10 мс.

6.16.3 Описание функционирования блока UART

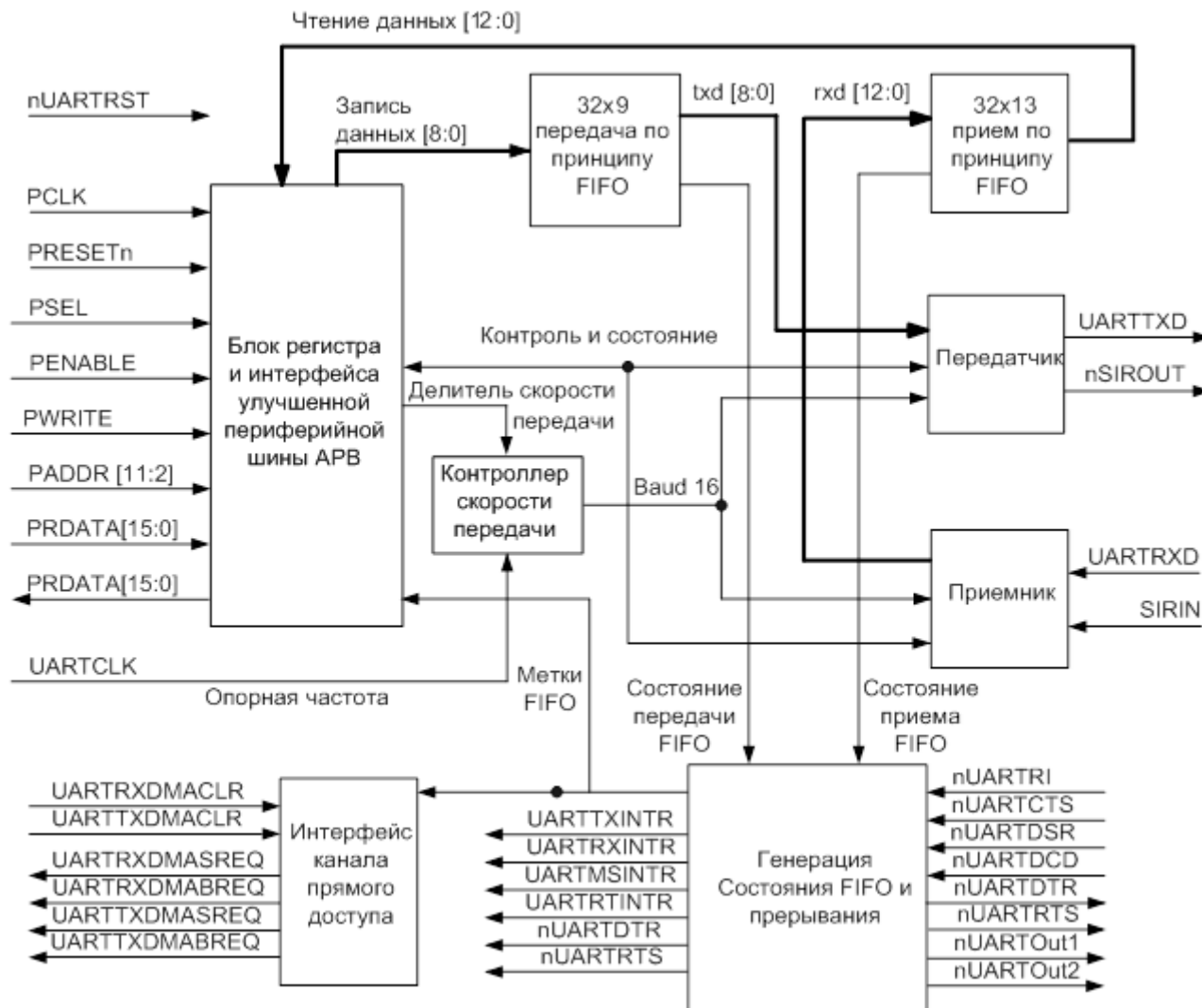


Рисунок 114 – Блок-схема универсального асинхронного приёмопередатчика (UART)

6.16.3.1 Генератор тактового сигнала приемопередатчика

Генератор содержит счетчики без цепи сброса, формирующие внутренние тактовые сигналы Baud16 и IrLPBaud16.

Сигнал Baud16 используется для синхронизации схем управления приемником и передатчиком последовательного обмена данными. Он представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UARTCLK и частотой, в 16 раз выше скорости передачи данных.

Сигнал IrLPBaud16 предназначен для синхронизации схемы формирования импульсов с длительностью, требуемой для ИК обмена данными в режиме с пониженным энергопотреблением.

6.16.3.2 Буфер FIFO передатчика

Буфер передатчика имеет ширину 9 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.

6.16.3.3 Буфер FIFO приемника

Буфер приемника имеет ширину 13 бит, глубину 16 слов, схему организации доступа типа «первый вошел, первый вышел». Принятые от периферийного устройства данные и соответствующие коды ошибки сохраняются логикой приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину APB. Буфер FIFO приемника может быть запрещен, в этом случае он будет действовать как однобайтовый буферный регистр.

6.16.3.4 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.

6.16.3.5 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.

6.16.3.6 Блок формирования прерываний

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

Комбинированный сигнал прерывания может быть подан на внешний контроллер прерываний системы, при этом появится дополнительная возможность маскирования устройства в целом, что облегчает построение модульных драйверов устройств.

Другой подход состоит в подаче на системный контроллер прерываний независимых линий запроса на прерывание от приемопередатчика. В этом случае процедура обработки сможет одновременно считать информацию обо всех источниках прерывания. Данный подход привлекателен в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

Для более подробной информации см. подраздел «Прерывания».

6.16.4 Интерфейс прямого доступа к памяти

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

6.16.5 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.

6.16.6 Описание функционирования ИК кодека IrDASIR

Структурная схема кодека представлена на рисунке 115.

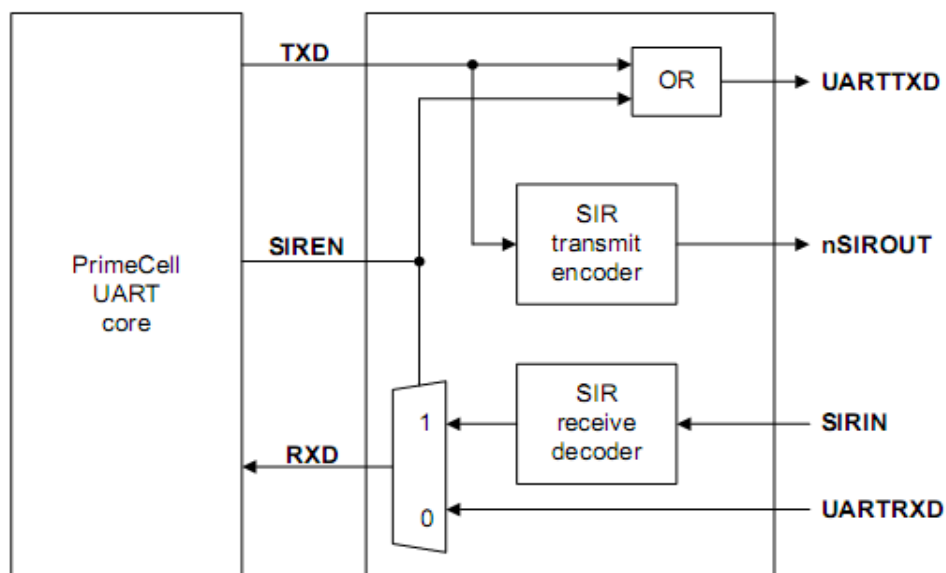


Рисунок 115 – Структурная схема кодека IrDA

6.16.6.1 Кодер ИК передатчика

Кодер преобразует поток данных с выхода асинхронного передатчика, сформированный по закону модуляции без возврата к нулю (NRZ). Спецификация физического уровня протокола IrDA SIR подразумевает использование модуляции с возвратом к нулю и инверсией (RZI), в соответствии с которой передача логического нуля соответствует излучению одного светового ИК импульса. Сформированный выходной поток импульсов подается на усилитель и, далее, на ИК светодиод.

Длительность импульса в режиме IrDA составляет, согласно спецификации, 3 периода внутреннего тактового генератора с частотой Baud16, то есть 3/16 периода времени, выделенного на передачу одного бита.

В режиме IrDA с пониженным энергопотреблением ширина импульса задана как 3/16 периода, выделенного на передачу бита, при скорости передачи данных 115200 бит/с. Данное требование реализуется за счет формирования трех периодов тактового сигнала IrLPBaud16 с номинальной частотой 1,8432 МГц, в свою очередь, формируемого путем деления частоты UARTCLK. Значение частоты IrLPBaud16 задается путем записи соответствующего коэффициента деления частоты в регистр UARTILPR.

Выход кодера имеет активное низкое состояние. При передаче логической единицы выход кодера остается в низком состоянии, при передаче логического нуля – формируется импульс, при этом выход кратковременно переводится в высокое состояние.

Как в нормальном режиме, так и в режиме пониженного энергопотребления использование нецелых значений коэффициента деления скорости передачи данных увеличивает джиттер («дребезжание») фронтов импульсов данных. Наличие джиттера в случае использования дробных коэффициентов деления связано с тем, что интервалы между тактовыми импульсами Baud16 будут нерегулярными – период сигнала Baud16 в разное время будет содержать различное количество периодов сигнала UARTCLK.

Можно показать, что в наихудшем случае величина джиттера в потоке ИК импульсов может достигать трех периодов UARTCLK. В соответствии со спецификацией стандарта IrDA SIR, джиттер не должен превышать величины 13 %. В случае, если частота сигнала UARTCLK составляет более 3.6834 МГц, а скорость передачи данных меньше или равна 115200 бит/с, величина джиттера не превышает 9 %. Таким образом, требования стандарта выполняются.

6.16.6.2 Декодер ИК приемника

Декодер преобразует поток данных, сформированных по закону возврата к нулю, полученного от приемника ИК сигнала, и выдает поток данных без возврата к нулю на вход приемника UART. В неактивном состоянии вход декодера находится нормально в высоком состоянии. Выходной сигнал кодера имеет полярность, противоположную полярности входа декодера.

Обнаружение стартового бита осуществляется при низком уровне сигнала на входе декодера.

Примечание – Для того чтобы исключить ложные срабатывания UART от импульсных помех, на входе SIRIN игнорируются импульсы с длительностью менее, чем:

- 3/16 длительности Baud16 в режиме IrDA;
- 3/16 длительности IrLPBaud16 в режиме IrDA с пониженным энергопотреблением.

6.16.7 Описание работы UART

6.16.7.1 Сброс модуля

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

6.16.7.2 Тактовые сигналы

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

$$F_{UARTCLK}(min) \geq 16 \times baud_rate_{max} \quad (16)$$

$$F_{UARTCLK}(max) \leq 16 \times 65535 \times baud_rate_{min} \quad (17)$$

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота UARTCLK должна находиться в интервале от 7,3728 до 115,34 МГц.

Частота UARTCLK, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами CPU_CLK и UARTCLK. Частота UARTCLK должна быть не более, чем в 5/3 раз выше частоты CPU_CLK

$$F_{UARTCLK} \leq \frac{5}{3} \cdot F_{CPU_CLK} \quad (18)$$

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.

6.16.7.3 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:
 - 7 параметры передачи данных;
 - 8 длину слова;
 - 9 режим буферизации;
 - 10 количество передаваемых стоповых бит;
 - 11 режим контроля четности;
 - 12 формирование сигнала разрыва линии;
- UARTIBRD – определяет целую часть коэффициента деления для скорости передачи данных;
- UARTFBRD – определяет дробную часть коэффициента деления для скорости передачи данных.

6.16.7.4 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:

$$\text{Коэффициент деления} = \frac{UARTCLK}{(16 \cdot \text{скорость передачи данных})} = BRD_I + BRD_F, \quad (19)$$

где BRD_I – целая часть;

BRD_F – дробная часть коэффициента деления.

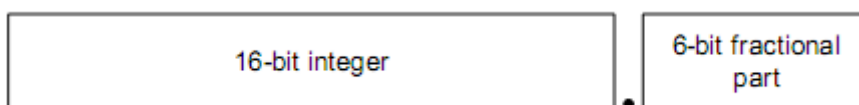


Рисунок 116 – Коэффициент деления

Шестибитное значение, записываемое в регистр UARTFBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на 2^6 , где n – ширина регистра UARTFBRD) и округления до ближайшего целого числа:

$$M = \lfloor BRD_F \cdot 2^n + 0,5 \rfloor, \quad (20)$$

где $\lfloor \rfloor$ – 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.

6.16.7.5 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 123).

6.16.7.6 Биты ошибки

Три бита признаков ошибки, ассоциированные с принятым символом данных, заносятся в разряды [11...9] слова данных в буфере FIFO приемника. Также предусмотрен признак ошибки переполнения буфера FIFO в разряде 12 слова данных.

Таблица 123 описывает назначение всех бит слова данных в FIFO-буфере приемника.

Т а б л и ц а 123 – Назначение бит слова данных в FIFO-буфере приемника

Бит буфера FIFO	Назначение
12	Признак переполнения буфера
11	Ошибка – «разрыв линии»
10	Ошибка проверки на четность
9	Ошибка формирования кадра
8...0	Принятые данные

6.16.7.7 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.

6.16.7.8 Запрет буфера FIFO

Предусмотрена возможность отключения FIFO буферов приемника и передатчика. В этом случае приемная и передающая сторона контроллера UART располагают лишь однокбайтными 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.

6.16.7.9 Проверка по шлейфу

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

6.16.8 Работа кодека ИК обмена данными IrDASIR

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

Предусмотрено два режима работы:

- **В режиме IrDA** уровень логического нуля передается на линию nSIROUT в виде импульса с высоким логическим уровнем и длительностью 3/16 от выбранного периода следования бит данных. Логическая единица при этом передается в виде постоянного низкого уровня сигнала. Сформированный выходной сигнал далее подается на передатчик ИК-сигнала, обеспечивая излучение светового импульса всякий раз при передаче нулевого бита. На приемной стороне световые импульсы воздействуют на базу фототранзистора

ИК приемника, который в результате формирует низкий логический уровень. Это, в свою очередь, обуславливает низкий уровень на входе SIRIN.

- В режиме IrDA с пониженным энергопотреблением длительность передаваемых импульсов ИК излучения устанавливается в три раза выше длительности импульсов внутреннего опорного сигнала IrLPBaud16 (равной 1,63 мкс при номинальной частоте 1,8432 МГц). Данный режим активизируется путем установки бита SIRLP в регистре управления UARTCR.

Как в нормальном режиме, так и в режиме пониженного энергопотребления:

- кодирование осуществляется на основе бит данных, сформированных асинхронным передатчиком модуля;
- в ходе приема данных декодированные биты далее обрабатываются блоком асинхронного приема.

В соответствии со спецификацией физического уровня протокола IrDA SIR, обмен данными должен осуществляться в режиме полудуплекса, при этом задержка между передачей и приемом данных должна составлять не менее 10 мс. Эта задержка должна формироваться программно. Необходимость ее введения обусловлена тем, что воздействие передающего ИК светодиода на находящийся рядом ИК приемник может привести к искажению принимаемого сигнала или даже ввести приемный тракт в состояние насыщения. Задержка между окончанием передачи и началом приема данных именуется латентность, или время установки (готовности) приемника.

Сигнал IrLPBaud16 формируется путем деления частоты сигнала UARTCLK в соответствии с коэффициентом деления, записанным в регистре UARTILPR.

Коэффициент деления вычисляется по формуле

$$\frac{F_{UARTCLK}}{F_{IrLPBaud16}}, \quad (21)$$

где номинальное значение IrLPBaud16 составляет 1,8432 МГц.

Коэффициент деления должен быть выбран так, чтобы выполнялось соотношение

$$1,42 \text{ МГц} < F_{IrLPBaud16} < 2,12 \text{ МГц} . \quad (22)$$

6.16.8.1 Проверка по шлейфу

Проверка по шлейфу выполняется после установки в 1 бита LBE регистра управления контроллером UARTCR с одновременной установкой в 1 бита SIRTEST регистра управления тестированием UARTTCR.

В этом режиме данные, передаваемые на выход nSIROUT, должны подаваться на вход SIRIN.

Примечание – Проверка по шлейфу – это единственный случай использования тестового регистра в нормальном режиме функционирования модуля.

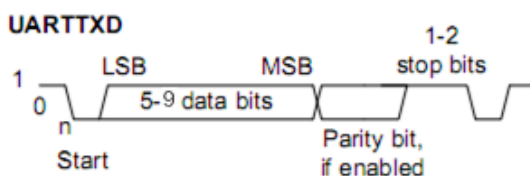


Рисунок 117 – Кадр передачи данных

6.16.9 Модуляция данных IrDA

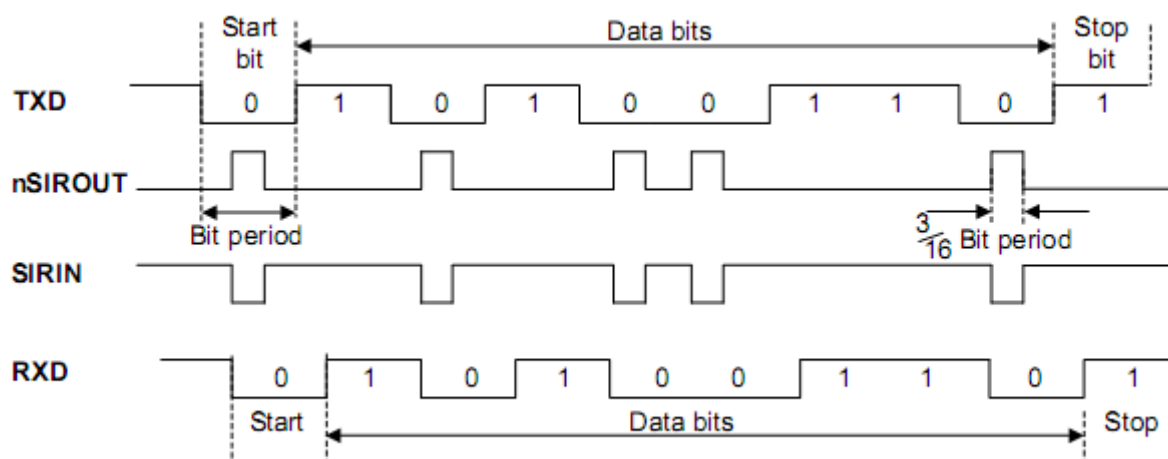


Рисунок 118 – Модуляция данных IrDA

6.16.10 Линии управления модемом

Модуль универсального асинхронного приемопередатчика может использоваться как в режиме оконечного оборудования (DTE), так и в режиме оборудования передачи данных (DCE). Сигналы модема в режиме DTE показаны ранее (см. рисунок 114).

Назначение сигналов в режимах DTE и DCE представлено в таблице 124.

Таблица 124 – Назначение управления модемом в режимах DTE и DCE

Сигнал	Назначение	
	Режим оконечного оборудования	Режим оборудования передачи данных
nUARTCTS	Готов к передаче данных	Запрос передачи данных
nUARTDSR	Источник данных готов	Приемник данных готов
nUARTDCD	Обнаружен информационный сигнал	-
nUARTRI	Индикатор вызова	-
nUARTRTS	Запрос передачи данных	Готов к передаче данных
nUARTDTR	Приемник данных готов	Источник данных готов
nUARTOUT1	-	Обнаружен информационный сигнал
nUARTOUT2	-	Индикатор вызова

6.16.10.1 Аппаратное управление потоком данных

Программно активизируемый режим аппаратного управления потоком данных позволяет контролировать (приостанавливать и возобновлять) информационный обмен с помощью сигналов nUARTRTS и nUARTCTS. Иллюстрация взаимодействия двух устройств последовательной связи с аппаратным управлением потоком данных представлена на рисунке 119.

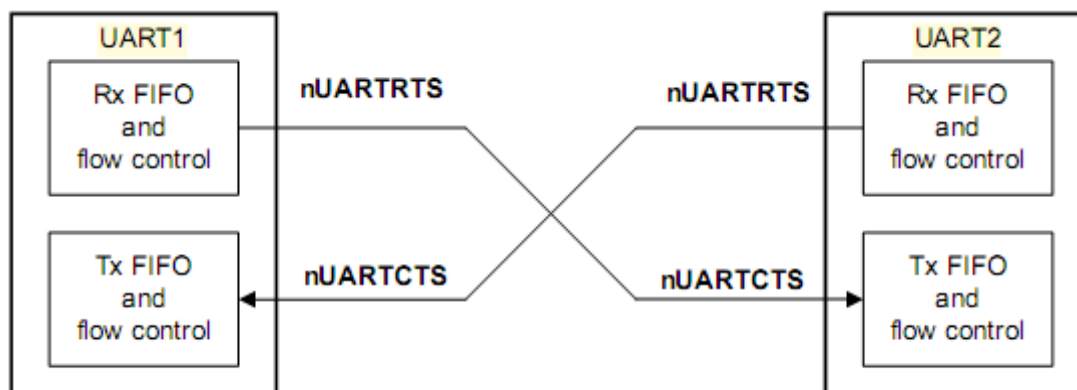


Рисунок 119 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

Если разрешено управление потоком данных по сигналу RTS, линия nUARTRTS переводится в активное состояние только после того, как в FIFO буфере приема появляется заданное количество свободных элементов.

Если разрешено управление потоком данных по сигналу CTS, передача данных осуществляется только после перевода линии nUARTCTS в активное состояние.

Режим аппаратного управления потоком данных задается путем установки значений бит RTSEn и CTSEn в регистре управления UARTCR. Таблица 125 показывает необходимые установки для различных режимов управления потоком данных.

Таблица 125 – Режимы управления потоком данных

CTSEn	RTSEn	Описание
1	1	Разрешено управление потоком данных по CTS и RTS
1	0	Управления потоком данных осуществляется по линии CTS
0	1	Управления потоком данных осуществляется по линии RTS
0	0	Управления потоком данных запрещено

Примечание – В случае если выбран режим управления потоком данных по RTS, программное обеспечение не может использовать бит RTSEn регистра UARTCR для проверки состояния линии RTS.

6.16.10.2 Управление потоком данных по линии RTS

Логика управления потоком данных по RTS использует данные о превышении пороговых уровней заполнения буфера FIFO приемника. В случае выбора режимов с управлением по RTS, сигнал на линии nUARTRTS переводится в активное состояние только после того, как в FIFO буфере приема появляется заданное количество свободных элементов. После достижения порогового уровня заполнения буфера приемника, сигнал nUARTRTS снимается (переводится в пассивное состояние), указывая на отсутствие свободного места для сохранения принятых данных. При этом дальнейшая передача данных должна быть прекращена по завершении передачи текущего символа.

Обратно, в активное состояние, сигнал nUARTRTS переводится после считывания данных из приемного буфера FIFO в количестве, достаточном для того, чтобы заполнение буфера оказалось ниже порогового уровня.

В случае если управление потоком данных по RTS запрещено, при этом работа приемопередатчика UART разрешена, прием будет осуществляться до полного заполнения буфера FIFO, либо до завершения передачи данных.

6.16.10.3 Управление потоком данных по линии CTS

В случае выбора одного из режимов с управлением потоком данных по CTS, передатчик осуществляет проверку состояния линии nUARTCTS перед началом передачи очередного байта данных. Передача осуществляется только в случае, если данная линия активна, и продолжается до тех пор, пока активное состояние линии сохраняется и буфер передатчика не пуст.

При переходе линии nUARTCTS в неактивное состояние модуль завершает выдачу текущего передаваемого символа, после чего передача данных прекращается.

Если управление потоком данных по CTS запрещено и при этом работа приемопередатчика UART разрешена – данные будут выдаваться до опустошения буфера FIFO передатчика.

6.16.11 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 13 бит. Сигнал переводится в активное состояние в случае, если буфер FIFO приемника содержит, по меньшей мере, один символ.
- UARTRXDMABREQ – запрос блочного обмена данными, инициируется модулем приемопередатчика. Сигнал переходит в активное состояние в случае, если заполнение буфера FIFO приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера FIFO, путем записи значения в регистр 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 126 показывает значения параметров срабатывания запросов блочного обмена UARTRXDMABREQ и UARTTXDMABREQ в зависимости от порога заполнения буфера.

Т а б л и ц а 126 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

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

В регистре управления DMA UARTDMACR предусмотрен бит DMAONERR, который позволяет запретить DMA от приемника в случае активного состояния линии прерывания по обнаружению 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 120 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.

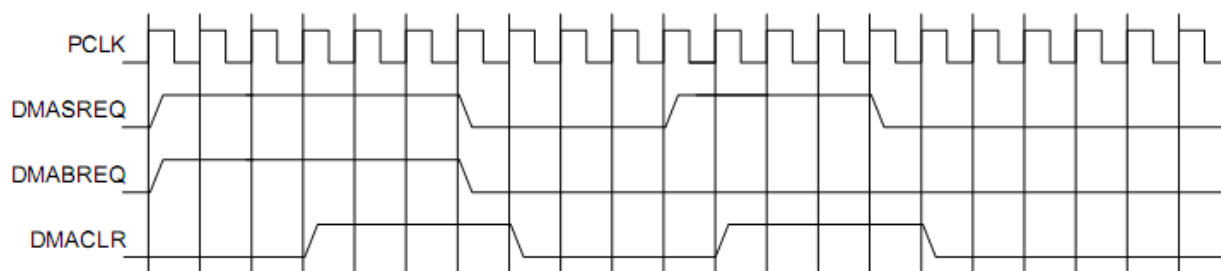


Рисунок 120 – Временные диаграммы одноэлементного и блочного запросов DMA

6.16.12 Прерывания

В модуле предусмотрено 14 маскируемых источников прерывания. В результате формируется один общий сигнал, представляющий собой комбинацию независимых сигналов, объединенных по схеме ИЛИ.

Сигналы запроса на прерывание:

- UARTRXINTR – прерывание от приемника;
- UARTTXINTR – прерывание от передатчика;
- UARTRNEINTR – FIFO приемника не пусто;
- UARTTFEINTR – FIFO передатчика пусто;
- UARTTNBSYINTR – сдвиговый регистр передатчика пуст;
- UARTRTINTR – прерывание по таймауту приемника;
- UARTMSINTR – прерывание по состоянию модема:
 - UARTRIINTR, изменение состояния линии nUARTRI;
 - UARTCTSINTR, изменение состояния линии nUARTCTS;
 - UARTDCDINTR, изменение состояния линии nUARTDCD;
 - UARTDSRINTR, изменение состояния линии nUARTDSR.
- UARTEINTR – ошибка:
 - UARTOEINTR, переполнение буфера;
 - UARTBEINTR, прерывание приема – разрыв линии;
 - UARTPEINTR, ошибка контроля четности;
 - UARTEINTR, ошибка в структуре кадра.
- UARTINTR – логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR, UARTEINTR, UARTRNEINTR, UARTTFEINTR и UARTTTBSYINTR.

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

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

Прерывание UARTRNEINTR возникает в случае, если FIFO приемника получает хотя бы одно слово данных, то есть становится не пусто.

Прерывание UARTTFEINTR выставляется, если FIFO передатчика не имеет никаких данных, то есть пусто в данный момент.

UARTTNBSYINTR выставляется в том случае, если сдвиговый регистр передатчика пуст.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра прерываний UARTRIS, либо из маскированного регистра прерываний UARTMIS.

6.16.12.1 UARTMSINTR

Прерывание по состоянию модема возникает в случае изменения любой из линий состояний модема (nUARTCTS, nUARTDCD, nUARTDSR, nUARTRI). Сброс прерывания осуществляется путем записи 1 в соответствующий (в зависимости от линии состояния модема, вызвавшей прерывание) разряд регистра сброса прерывания UARTICR.

6.16.12.2 UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен и его заполнение достигло заданного порогового значения. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения данных из буфера приемника до тех пор, пока его заполнение не станет меньше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), принят один символ данных. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения одного байта данных, либо после сброса прерывания.

6.16.12.3 UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

- буфер FIFO разрешен и его заполнение меньше или равно заданному пороговому значению. В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи данных в буфера передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания;
- буфер FIFO запрещен (имеет размер один символ), данные в буферном регистре передатчика отсутствуют. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи одного байта данных, либо после сброса прерывания.

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

Примечание – Прерывание передатчика 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.

6.16.12.4 UARTRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения (или считывания одного байта в случае, если буфер FIFO запрещен), либо путем записи 1 в соответствующий бит регистра сброса прерывания UARTICR.

6.16.12.5 UARTEINTR

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

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

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

6.16.12.6 UARTINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов UARTRXINTR, UARTRTXINTR, UARTRTINTR, UARTMSINTR, UARTEINTR, UARTRNEINTR, UARTRFEINTR и UARTRTNBSYINTR с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерываний, что позволит ввести дополнительное маскирование запросов на уровне периферийных устройств.

6.16.13 Программное управление модулем

6.16.13.1 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. в таблицу 127) определяет режим доступа к регистру в соответствии с обозначениями:
 - RW – чтение и запись;
 - RO – только чтение;
 - WO – только запись.

6.16.14 Описание регистров

Данные о регистрах модуля универсального асинхронного приемопередатчика приведены в таблице 127.

Т а б л и ц а 127 – Обобщенные данные о регистрах устройства

Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
0x40099000	MDR_UART1				Контроллер UART1
0x4009A000	MDR_UART2				Контроллер UART2
0x000	DR	RW	0x---	13/9	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	ILPR	RW	0x00	8	Регистр управления ИК обменом в режиме пониженного энергопотребления
0x024	IBRD	RW	0x0000	16	Целая часть делителя скорости обмена данными
0x028	FBRD	RW	0x00	6	Дробная часть делителя скорости обмена данными
0x02C	LCR_H	RW	0x00	9	Регистр управления линией
0x030	CR	RW	0x0300	16	Регистр управления
0x034	IFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO
0x038	IMSC	RW	0x0000	14	Регистр маски прерывания
0x03C	RIS	RO	0x100-	14	Регистр состояния прерываний
0x040	MIS	RO	0x000-	14	Регистр состояния прерываний с маскированием
0x044	ICR	WO	-	11	Регистр сброса прерывания
0x048	DMACR	RW	0x00	3	Регистр управления DMA

6.16.14.1 Регистр DR

В ходе передачи данных:

Если буфер FIFO передатчика разрешен, то слово данных, записанное в рассматриваемый регистр, направляется в буфер FIFO передатчика.

В противном случае, записанное слово фиксируется в буферный регистр передатчика (последний элемент буфера FIFO).

Операция записи в регистр инициирует передачу данных. Слово данных предваряется стартовым битом, дополняется битом контроля четности (если режим контроля четности включен) и стоповым битом. Сформированное слово отправляется в линию передачи данных.

В ходе приема данных:

Если буфер FIFO приемника разрешен, байт данных и четыре бита состояния (разрыв, ошибка формирования кадра, четность, переполнение) сохраняются в 13-битном буфере.

В противном случае байт данных и биты состояния записываются в буферный регистр (последний элемент буфера FIFO).

Полученные из линии связи байты данных считываются путем чтения из регистра UARTDR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра UATRSR/UARTECR (см. таблицу 128).

Таблица 128 – Формат регистра UARTDR

№ бита	Сигнал	Назначение
15...13		Резерв
12	OE	Переполнение буфера приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время, как буфер заполнен. Сбрасывается в 0 после того, как в буфере появится свободное место
11	BE	Разрыв линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита
10	PE	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам бит EPS и SPS в регистре управления линией UARTECR_H. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
9	FE	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер
8...0	DATA	Принимаемые данные (чтение). Передаваемые данные (запись)

Примечание – Необходимо запрещать работу приемопередатчика перед любым перепрограммированием его регистров управления. Если приёмопередатчик переводится в

отключенное состояние во время передачи или приема символа, то перед остановкой он завершает выполняемую операцию.

6.16.14.2 Регистр RSR_ECR

Состояние приемника также может быть считано из регистра UARTSR. 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.

Таблица 129 показывает назначение бит регистра UARTSR/UARTECR.

Таблица 129 – Регистр UARTSR/UARTECR

№ 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, 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 UARTSR необходимо считать данные, принятые из линии, путем обращения к регистру данных UARTDR. Противоположная последовательность действий не допускается, так как регистр UARTSR 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 UARTSR/UARTECR любого кода сбрасывает признаки ошибок формирования кадра, проверки на четность, разрыва линии и переполнения буфера.

6.16.14.3 Регистр FR

После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE – в 1. В таблице 130 представлена информация о назначении бит регистра UARTFR.

Таблица 130 – Регистр UARTFR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...9		Резерв. Не модифицируйте. При чтении заполняются нулями
8	RI	Инверсия линии nUARTRI
7	TXFE	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр передатчика пуст. В противном случае он равен 1, если пуст буфер FIFO передатчика. Данный бит не дает никакой информации о наличии данных в регистре сдвига передатчика
6	RXFF	Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника занят. В противном случае он равен 1, если заполнен буфер FIFO приемника
5	TXFF	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит равен 1, когда буферный регистр передатчика занят. В противном случае он равен 1, если заполнен буфер FIFO передатчика
4	RXFE	Буфер FIFO приемника пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, бит устанавливается в 1, когда буферный регистр приемника пуст. В противном случае он равен 1, если пуст буфер FIFO приемника
3	BUSY	UART занят. Бит равен 1 в случае, если контроллер передает в линию данные. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Кроме того, бит занятости устанавливается в 1 при наличии данных в буфере FIFO передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен)
2	DCD	Инверсия линии nUARTDCD
1	DSR	Инверсия линии nUARTDSR
0	CTS	Инверсия линии nUARTCT

6.16.14.4 Регистр ILPR

Этот восьмиразрядный регистр, доступный для чтения и записи, содержит значение коэффициента деления частоты UARTCLK, для формирования тактового сигнала IrLPBaud16. Назначение разрядов регистра показано в таблице 131.

Требуемое значение коэффициента деления для формирования сигнала IrLPBaud16 вычисляется по формуле

$$ILPDVSR = \frac{F_{UARTCLK}}{F_{IrLPBaud16}}, \quad (23)$$

где номинальное значение частоты $F_{IrLPBaud16}$ составляет 1,8432 МГц.

Коэффициент деления должен быть установлен таким образом, чтобы выполнялось соотношение

$$1,42 \text{ МГц} < F_{IrLPBaud16} < 2,12 \text{ МГц}, \quad (24)$$

что, в свою очередь, гарантирует формирование кодеком импульсов данных с длительностью $1,41 \div 2,11$ мкс (в три раза длиннее периода сигнала IrLPBaud16).

Таблица 131 – Регистр UARTILPR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7...0	ILPDVSR	Коэффициент деления частоты UARTCLK, для формирования тактового сигнала IrLPBaud16. После сброса устанавливается в 0. Примечание – Коэффициент 0 – запрещенное значение. В случае его установки импульсы IrLPBaud16 формироваться не будут

Примечание – В интересах подавления помех при работе в режиме IrDA с пониженным энергопотреблением кодек игнорирует поступающие на вход SIRIN импульсы с длительностью, меньшей трех периодов сигнала IrLPBaud16.

6.16.14.5 Регистр IBRD

Назначение бит регистра UARTBIRD показано ниже.

Таблица 132 – Регистр UARTBIRD

№ 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

6.16.14.6 Регистр FBRD

Таблица 133 показывает назначение бит регистра.

Таблица 133 – Регистр UARTBFRD

№ 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

$$BAUDDIV = \frac{F_{UARTCLK}}{16 \cdot Baud_rate}, \quad (25)$$

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.

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, BRD = 1, FBRD = 0,085.

Следовательно, значение, записываемое в регистр UARTBFRD, равно

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

Реальное значение коэффициента деления = $1 + 5/64 = 1,078$.

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

$$\text{Ошибка установки скорости} = \frac{231911 - 230400}{230400} \times 100\% = 0,656\%.$$

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

В следующей таблице 134 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 134 – 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

Коэффициент деления	Скорость передачи данных
0x0030	9600
0x00C0	2400
0x0180	1200
0x105D	110

В таблице 135 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц.

Таблица 135 – 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

6.16.14.7 Регистр LCR_H

Данный регистр обеспечивает доступ к разрядам с 30 по 22 регистра UARTLCR. При сбросе все биты регистра UARTLCR_H обнуляются.

Таблица 136 показывает назначение разрядов регистра UARTLCR_H.

Таблица 136 – Регистр UARTLCR_H

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15...9		Резерв. Не модифицируйте. При чтении выдаются нули
8	WLEN[2]	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0 – длина слова определяется WLEN[1:0] 1 – 9 бит
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 запрещен
6...5	WLEN[1:0]	Длина слова – количество передаваемых или принимаемых информационных бит в кадре: 0b11 – 8 бит 0b10 – 7 бит 0b01 – 6 бит 0b00 – 5 бит
4	FEN	Разрешение работы буфера FIFO приемника и передатчика. 0 – запрещено; 1 – разрешено

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3	STP2	Режим передачи двух стоповых бит. 0 – один стоповый бит; 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре
2	EPS	Четность/нечетность. 0 – бит четности дополняет количество единиц в информационной части кадра до нечетного; 1 – до четного числа. Значение бита EPS не играет роли в случае, если битом PEN формирование и проверка бита четности запрещена
1	PEN	Разрешение проверки четности. 0 – кадр не содержит бита четности; 1 – бит четности передается в кадре и проверяется при приеме данных
0	BRK	Разрыв линии. Если этот бит установлен в 1, 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 31-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

- Изменение значений трех регистров можно осуществить корректно двумя способами:
 - запись UARTIBRD, запись UARTFBRD, запись UARTLCR_H;
 - запись UARTFBRD, запись UARTIBRD, запись UARTLCR_H.
- Для того чтобы изменить значение лишь одного из регистров (UARTIBRD или UARTFBRD) необходимо выполнить следующий шаг:
 - запись UARTIBRD (или UARTFBRD), запись UARTLCR_H.

Таблица 137 показывает таблицу истинности для бит управления контролем четности PEN, EPS и SPS регистра управления линией UARTLCR_H.

Таблица 137 – 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

- 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.

6.16.14.8 Регистр CR

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 138.

Таблица 138 – Регистр управления UARTCR

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
15	CTSEn	Разрешение управления потоком данных по CTS. 1 – разрешено, данные передаются в линию только при активном значении сигнала nUARTCTS.
14	RTSEn	Разрешение управления потоком данных по RTS. 1 – разрешено. Запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере FIFO приемника
13	Out2	Инверсия сигнала на линии состояния модема nUARTOut2. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «сигнал вызова» (RI)
12	Out1	Инверсия сигнала на линии состояния модема nUARTOut1. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «обнаружен информационный сигнал» (DCD)
11	RTS	Инверсия сигнала на линии состояния модема nUARTRTS
10	DTR	Инверсия сигнала на линии состояния модема nUARTDTR
9	RXE	Прием разрешен. Установка бита в 1 разрешает работу приемника. Прием данных осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчика в запрещенное состояние в ходе приема данных, он завершает прием текущего символа перед остановкой
8	TXE	Передача разрешена. Установка бита в 1 разрешает работу передатчика. Передача осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. В случае перевода приемопередатчик в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа: перед остановкой

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
7	LBE	0 – запрещено; 1 – шлейф разрешен. В режиме разрешенного шлейфа: Если установлены бит SIREN=1 и бит регистра управления тестированием UARTRCSIRTEST=1, то сигнал с выхода кодека nSIROUT инвертируется и подается на вход кодека SIRIN. Бит SIRTEST устанавливается в 1 для того, чтобы вывести устройство из полудуплексного режима, характерного для интерфейса SIR. После окончания тестирования по шлейфу бит SIRTEST должен быть установлен в 0. Если бит SIRTEST=0, то выходная линия передатчика UARTRTXD коммутируется на вход приемника UARTRXD. Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии. После сброса бит устанавливается в 0
6...3	–	Резерв. Не модифицируйте. При чтении выдаются нули.
2	SIRLP	Выбор режима ИК обмена с пониженным энергопотреблением: 0 – длительность импульсов данных равна 3/16 длительности передачи бита; 1 – длительность импульсов данных равна трем тактам сигнала IrLPBaud16 вне зависимости от выбранной скорости передачи данных. Выбор этого режима снижает энергопотребление, однако может привести к уменьшению дальности связи
1	SIREN	Разрешение работы кодека ИК передачи данных IrDASIR: 0 – запрещено. Сигнал nSIROUT находится в низком состоянии, данные на входе SIRIN не обрабатываются. 1 – разрешено. Данные передаются на выход nSIROUT и принимаются с входа SIRIN. Линия UARTRTXD находится в высоком состоянии. Данные на входе UARTRXD и линиях состояния модема не обрабатываются. В случае, если UARTEN=0 значение бита не играет роли
0	UARTEN	Разрешение работы приемопередатчика: 0 – работа запрещена. Перед остановкой завершается прием и/или передача обрабатываемого в текущий момент символа. 1 – работа разрешена. Производится обмен данными либо по линиям асинхронного обмена, либо по линиям ИК обмена SIR, в зависимости от состояния бита SIREN

Примечания

- 1 Для того, чтобы разрешить передачу данных, биты TXE и UARTEN необходимо установить в логическую 1. Аналогично, для разрешения приема данных необходимо установить в 1 биты RXE и UARTEN.
- 2 Для программирования регистров управления рекомендуется следующая последовательность действий:
 1. остановите работу приемопередатчика;
 2. дождитесь окончания приема и/или передачи текущего символа данных;
 3. сбросьте буфер передатчика установкой в 0 бита FEN регистра UARTRLCR_H;
 4. измените настройки регистра UARTRCR;
 5. возобновите работу приемопередатчика.

6.16.14.9 Регистр IFLS

Данный регистр используется для установки порогового значения заполнения буферов передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода величины заполнения буфера через заданное значение.

После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра UARTIFLS и значения его бит представлены в таблице 139.

Таблица 139 – Регистр UARTIFLS

№ 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

6.16.14.10 Регистр IMSC

При чтении выдается текущее значение маски. При записи производится установка или сброс маски на соответствующее прерывание.

После сброса все биты регистра маски устанавливаются в нулевое состояние. Назначение бит регистра UARTIMSC показано в таблице 140.

Таблица 140 – Регистр UARTIMSC

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14		Зарезервировано. Не модифицируйте. При чтении выдаются нули
13	TNBSYIM	Маска прерывания по отсутствию данных в сдвиговом регистре передатчика UARTTNBSYINTR: 1 – установлена; 0 – сброшена
12	TFEIM	Маска прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR: 1 – установлена; 0 – сброшена
11	RNEIM	Маска прерывания по наличию данных в FIFO приемника UARTRNEINTR: 1 – установлена; 0 – сброшена

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
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 UARCTXINTR. 1 – установлена; 0 – сброшена
4	RXIM	Маска прерывания от приемника UARTRXINTR. 1 – установлена; 0 – сброшена
3	DSRMIM	Маска прерывания UARTDSRINTR по изменению состояния линии nUARTDSR: 1 – установлена; 0 – сброшена
2	DCDMIM	Маска прерывания UARTDCDINTR по изменению состояния линии nUARTDCD: 1 – установлена; 0 – сброшена
1	CTSMIM	Маска прерывания UARTCTSINTR по изменению состояния линии nUARTCTS: 1 – установлена; 0 – сброшена
0	RIMIM	Маска прерывания UARTRIINTR по изменению состояния линии nUARTRI: 1 – установлена; 0 – сброшена

6.16.14.11 Регистр RIS

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Предупреждение. После сброса все биты регистра, за исключением бит прерывания по состоянию модема (биты с 3 по 0) и бита прерывания по отсутствию данных в FIFO передатчика (бит 12), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Назначение бит регистра UARTRIS представлено в следующей таблице 141.

Таблица 141 – Регистр UARTRIS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14		Зарезервировано. Не модифицируйте. При чтении выдаются нули
13	TBNSYRIS	Состояние прерывания по отсутствию данных в сдвиговом регистре передатчика UARTTNBSYINTR
12	TFERIS	Состояние прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR
11	RNERIS	Состояние прерывания по наличию данных в FIFO приемника UARTRNEINTR
10	OERIS	Состояние прерывания по переполнению буфера UARTOEINTR
9	BERIS	Состояние прерывания по разрыву линии UARTBEINTR
8	PERIS	Состояние прерывания по ошибке контроля четности UARTPEINTR
7	FERIS	Состояние прерывания по ошибке в структуре кадра UARTFEINTR
6	RTRIS	Состояние прерывания по таймауту приема данных UARTRTINTR Примечание - Бит RTRIS может быть установлен только при установленной маске прерывания по таймауту приема данных UARTRTINTR в регистре UARTIMSC. Это вызвано тем, что сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Чтение состояния прерывания по таймауту из регистров 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

6.16.14.12 Регистр MIS

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением 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 142.

Таблица 142 – Регистр UARTMIS

№ бита	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...14	–	Зарезервировано. Не модифицируйте. При чтении выдаются нули
13	TNBSYMIS	Маскированное состояние прерывания по отсутствию данных в сдвиговом регистре передатчика UARTTNBSYINTR
12	TFEMIS	Маскированное состояние прерывания по отсутствию данных в FIFO передатчика UARTTFEINTR
11	RNEMIS	Маскированное состояние прерывания по наличию данных в FIFO приемника UARTRNEINTR
10	OEMIS	Маскированное состояние прерывания по переполнению буфера UARTOEINTR
9	BEMIS	Маскированное состояние прерывания по разрыву линии UARTBEINTR
8	PEMIS	Маскированное состояние прерывания по ошибке контроля четности UARTPEINTR
7	FEMIS	Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR
6	RTMIS	Маскированное состояние прерывания по таймауту приема данных UARTRTINTR
5	TXMIS	Маскированное состояние прерывания от передатчика UARTTXINTR
4	RXMIS	Маскированное состояние прерывания от приемника UARTRXINTR
3	DSRMMIS	Маскированное состояние прерывания UARTDSRINTR по изменению линии nUARTDSR
2	DCDMMIS	Маскированное состояние прерывания UARTDCDINTR по изменению линии nUARTDCD
1	CTSMMIS	Маскированное состояние прерывания UARTCTSINTR по изменению линии nUARTCTS
0	RIMMIS	Маскированное состояние прерывания UARTRIINTR по изменению линии nUARTRI

6.16.14.13 Регистр ICR

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 143.

Таблица 143 – Регистр UARTICR

№ 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	DSRMIC	Сброс прерывания UARTDSRINTR по изменению линии nUARTDSR
2	DCDMIC	Сброс прерывания UARTDCDINTR по изменению линии nUARTDCD
1	CTSMIC	Сброс прерывания UARTCTSINTR по изменению линии nUARTCTS
0	RIMIC	Сброс прерывания UARTRIINTR по изменению линии nUARTRI

6.16.14.14 Регистр DMACR

Регистр доступен по чтению и записи. После сброса, все биты регистра обнуляются. Назначение бит регистра UARTDMACR представлено в таблице 144.

Таблица 144 – Регистр UARTDMACR

№ 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 UARTRXDMAREQ и UARTRXDMAREQ
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

6.17 Контроллер интерфейса МКПД (MILx)

В микроконтроллере имеется четыре 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 приёма/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 контроллера 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 приёме сообщения, данные можно считывать из памяти, как на «лету», так и после установки флага VALMESS.

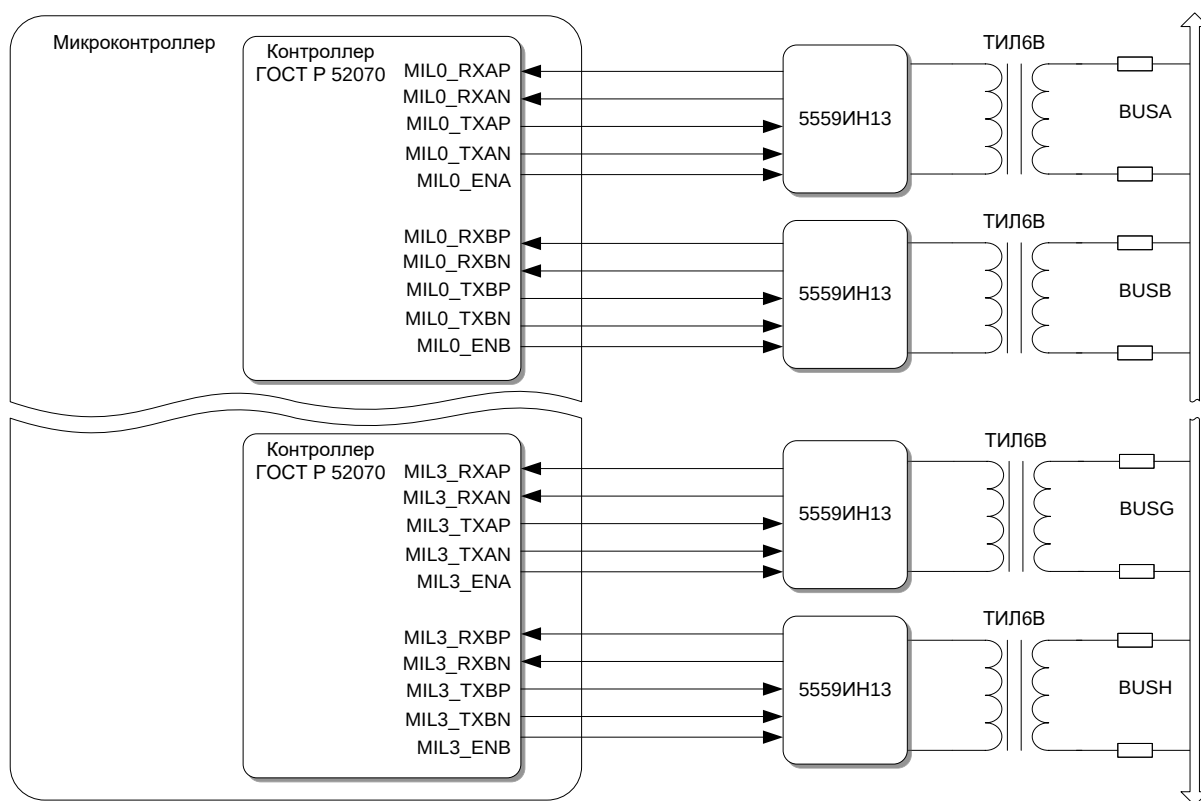


Рисунок 121 – Структурная схема контроллера интерфейса МКПД

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 1Kx16;
- двухпортовая память передаваемых данных 1Kx16;

- возможность формирования прерываний при успешном приеме и при возникновении ошибок на шине;
- маскирование прерываний.

12.1.1 Режимы работы

Контроллер поддерживает три режима работы:

- контроллер шины (КШ);
- оконечное устройство (ОУ);
- неадресуемый монитор (М).

6.17.1.1 Режим 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 приёма из магистрали хранится в регистре StatusWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр StatusWord2. Для передачи и приёма 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.

6.17.1.2 Режим 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 приёме, или если поле «Адрес ОУ» соответствует коду собственного адреса ОУ или коду 11111 (групповая команда). Если командное слово определено как достоверное, ОУ посылает в линию ответное слово (ОС) и в зависимости от поля «Приём/Передача» принимает или передаёт число данных, соответствующее полю «Число СД/Код КУ». Если же происходит приём 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 приёма 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.

6.17.1.3 Режим 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 передаёт 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 приёма 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 определённый код в этом регистре.

6.17.2 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 ниже (см. рисунок 122), 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, фиксируется отсутствие ОС от ОУ и формируется соответствующий признак ошибки.

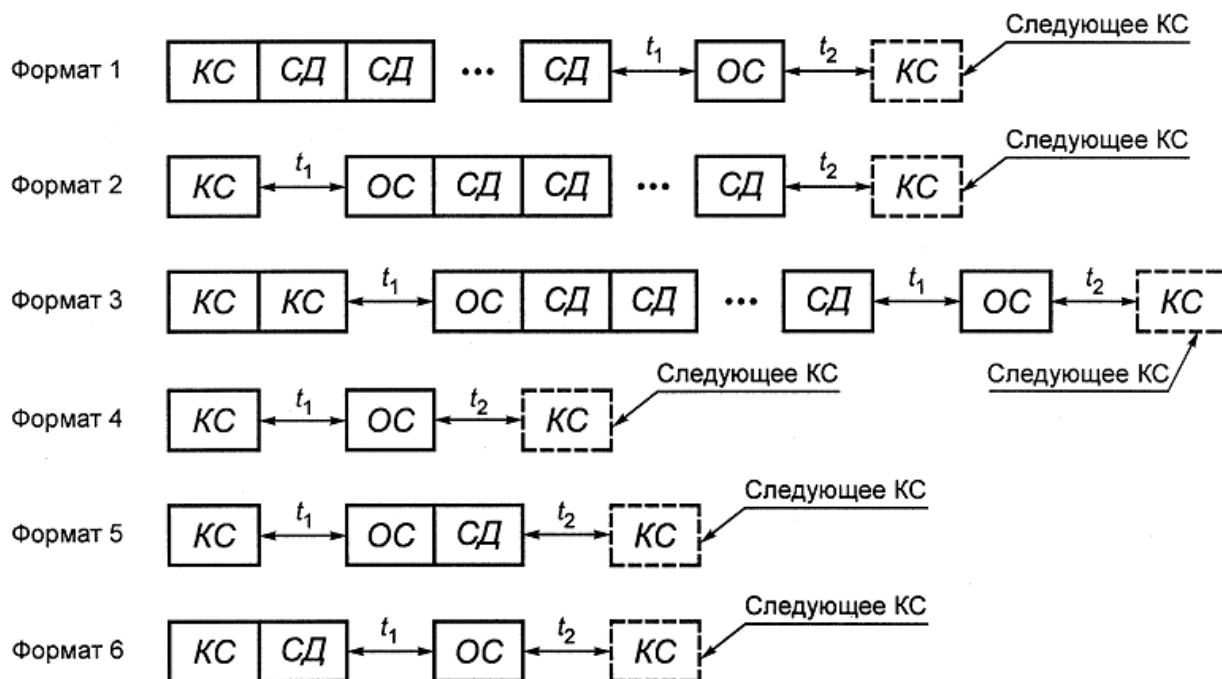


Рисунок 122 – Форматы сообщений (часть 1)

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 приём СД от ОУ.

Формат 6 – передача КУ и СД окончному устройству.

Групповые сообщения, приведенные ниже (см. Рисунок 123), 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.

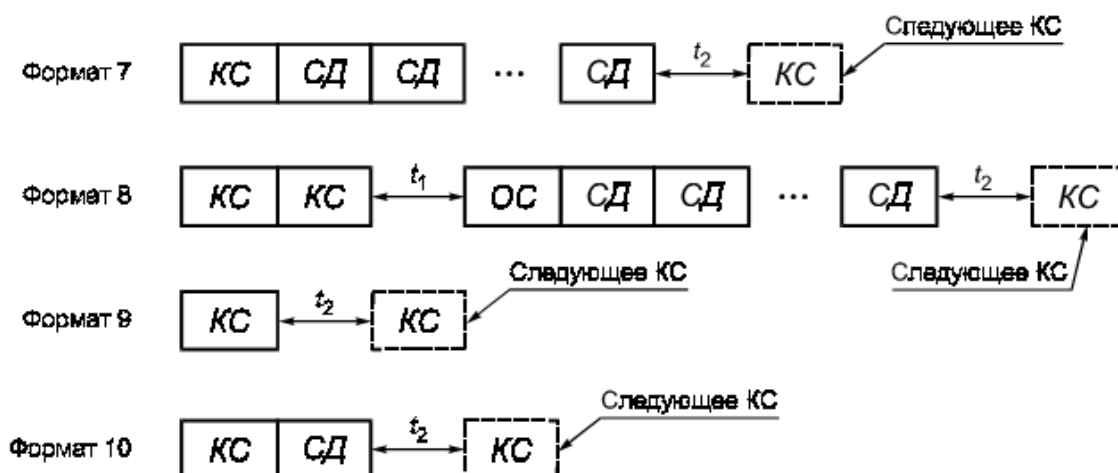


Рисунок 123 – Форматы сообщений (часть 2)

Если ОУ в формате сообщения ОУ-ОУ получило достоверное командное слово на приём информации, то первое СД должно быть им принято через паузу не более (57 ± 3) мкс, в противном случае формируется соответствующий признак ошибки.

6.17.3 Формат слов

Каждое слово начинается с сигнала пословной синхронизации (с синхросигнала) и имеет 17 информационных разрядов, включая разряд контроля по чётности. Форматы слов приведены на рисунке 124.

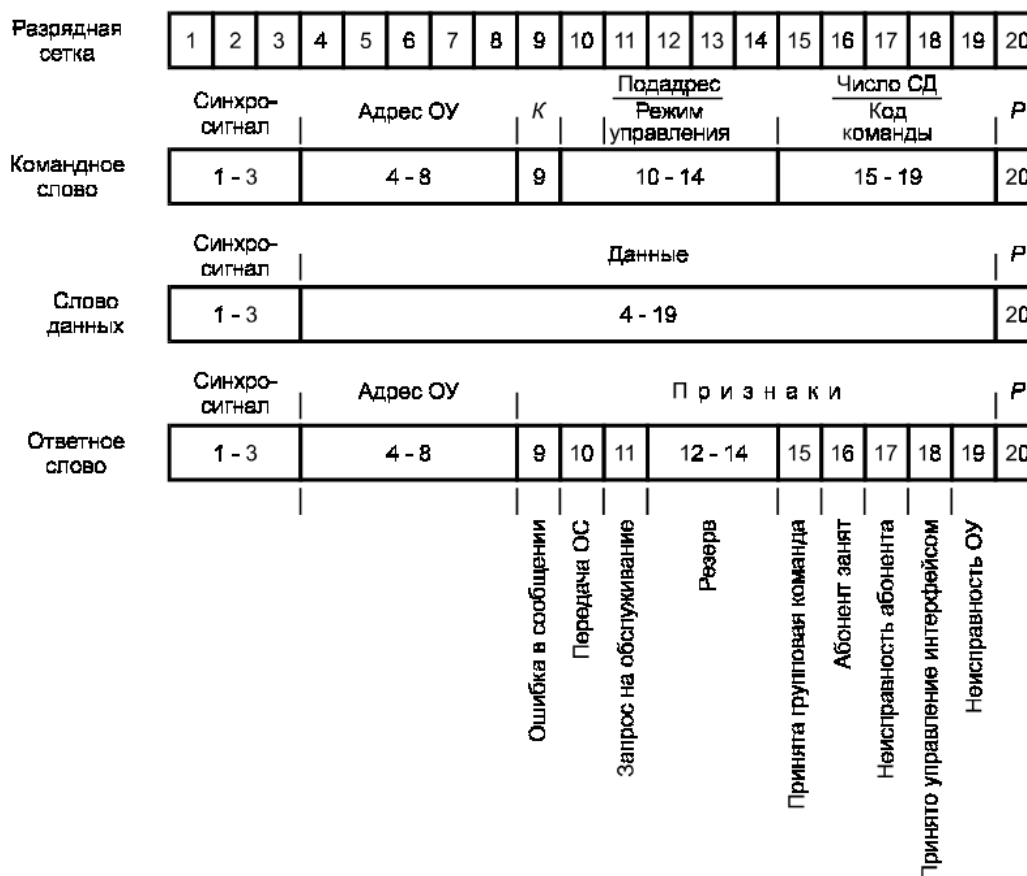


Рисунок 124 – Форматы слов

Командное слово содержит:

- синхросигнал;
- поле «Адрес ОУ»;
- разряд «Приём/Передача»;
- поле «Подадрес/Режим управления»;
- поле «Число СД/Код КУ»;
- разряд контроля по чётности.

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 «Приём/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 приёмом 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 чётности используется для контроля по чётности предшествующих ему 16 разрядов КС. Разряд принимает такое значение, чтобы сумма значений всех 17 информационных разрядов слова (включая контрольный разряд) была нечётной.

Слово данных содержит:

- синхросигнал;
- данные;
- разряд контроля по чётности.

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 чётности 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 чётности формируется так же, как в командном слове.

6.17.4 Структурная схема в режиме КШ

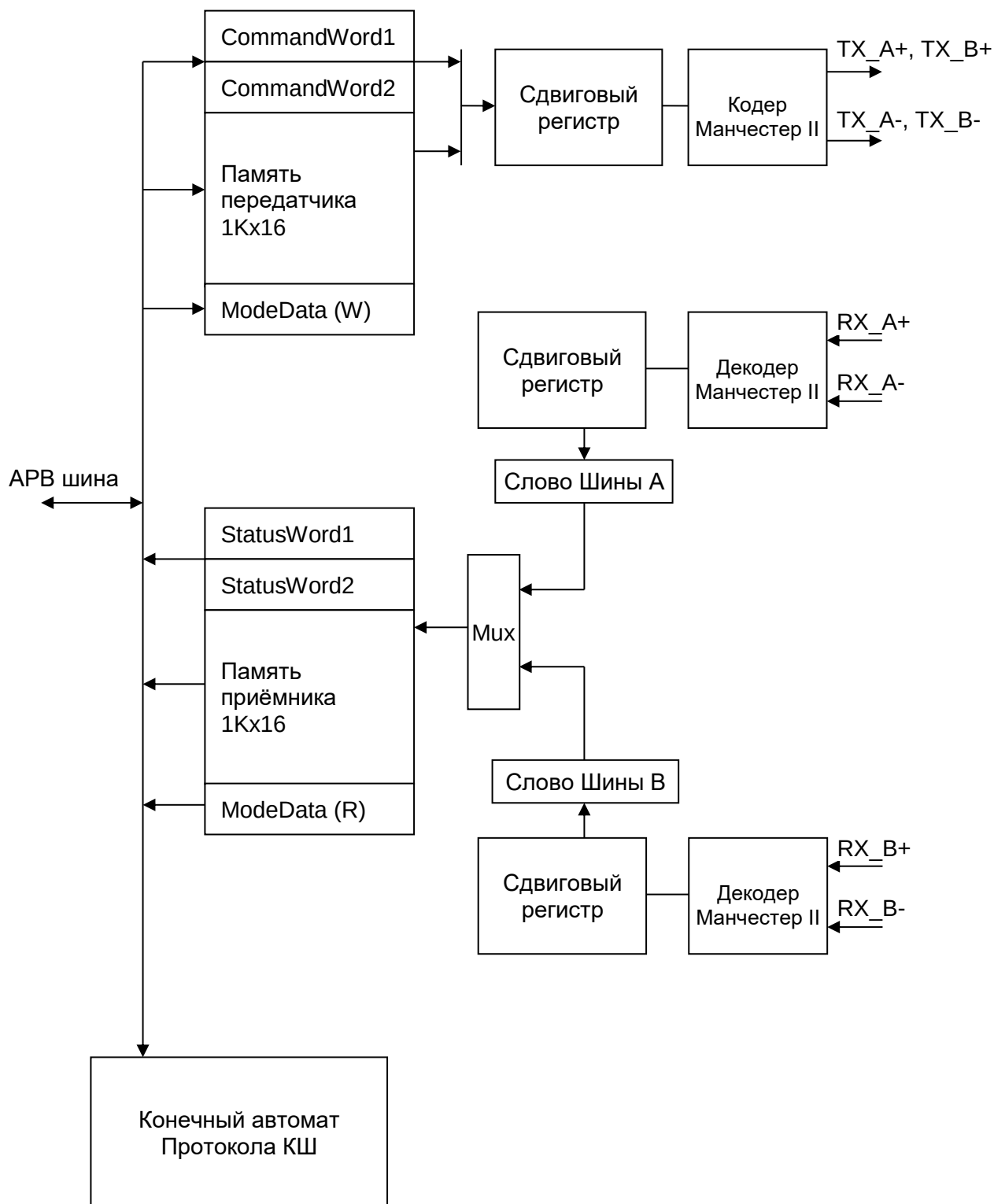


Рисунок 125 – Структурная схема в режиме КШ

6.17.5 Структурная схема в режиме ОУ

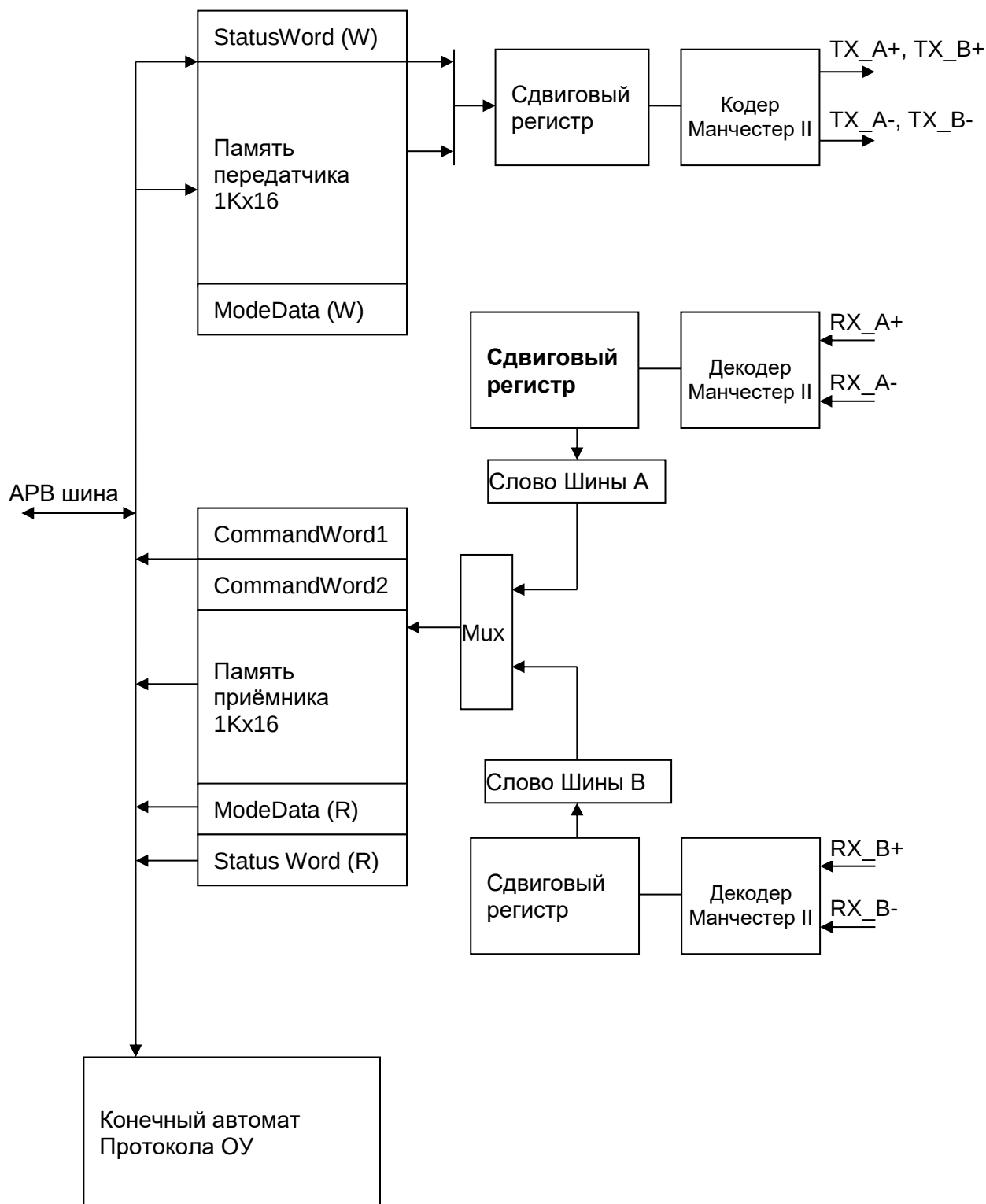


Рисунок 126– Структурная схема в режиме ОУ

6.17.6 Структурная схема в режиме M

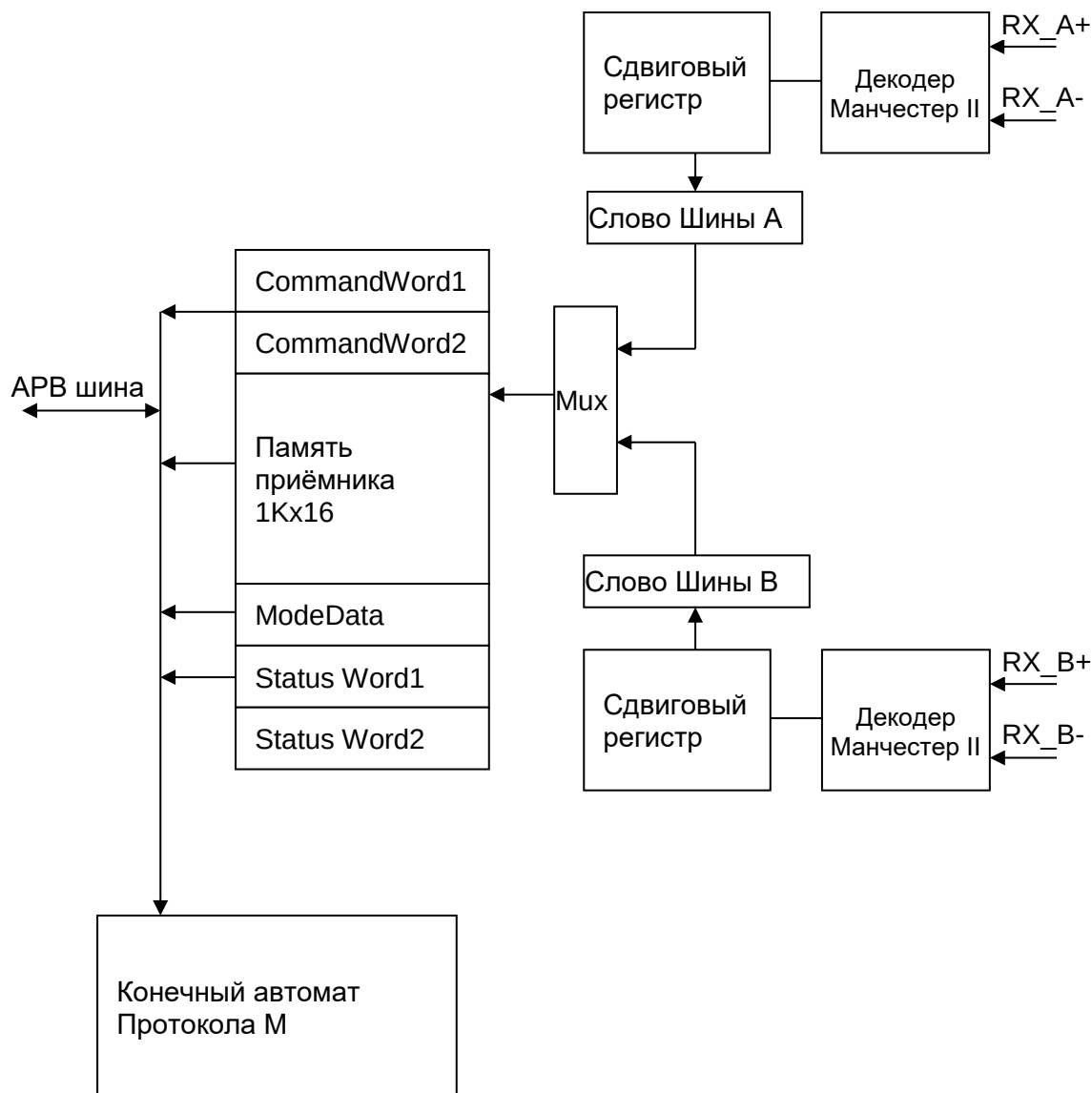


Рисунок 127 – Структурная схема в режиме M

6.17.7 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 задаётся адрес ОУ, который должен соответствовать адресу в поле «Адрес ОУ» командного слова, если идёт обращение к этому ОУ.

Пример инициализации ОУ

```
*((volatile unsigned int *) (0x40051000))=0x00000001; //установка бита MR=1
*((volatile unsigned int *) (0x40051000))=0x00014078;
//RTMODE=1, TRB=TRA=1, RTA=1, DIV=40
```

Пример инициализации КШ

```
*((volatile unsigned int *) (0x40051000))=0x00000001; //установка бита MR=1
*((volatile unsigned int *) (0x40051000))=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.

6.17.8 Приём и передача в режиме ОУ

Для того, чтобы настроить контроллер в режиме ОУ, необходимо выполнить все пункты, описанные в подразделе 6.17.7 «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 *) (0x40051018))=0x00000800;
```

В данном случае в регистр заносятся только старшие 5 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 *) (0x40051014))=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 определённых 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-разрядные, 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 *) (0x40050000+addon))=i;
addon = addon +4;
}
```

Из примера видно, что стартовый адрес памяти СД для подадреса 1 – 0x80, для последующих подадресов: $n \cdot 0x80$, где n – номер подадреса ($n=1-31$).

При приёме 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 *) (0x40051014));
```

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 *) (0x40050000+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 определённый 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.

6.17.9 Приём 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 инициализировать процедуру приёма 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 *) (0x4005100C))=0x00000820;    //Командное слово 1
*((volatile unsigned int *) (0x40051010))=0x00000000;    //Командное слово 2
*((volatile unsigned int *) (START_ADDR_APB+0x1000))=0x00014016; //Регистр управления
```

Как видно из примера, в командном слове 1 задаётся код слов данных 00000, что соответствует 32 СД. Данные будут передаваться от контроллера шины окончному устройству с адресом 1 из подадреса 1. Второе командное слово задаётся 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 иницирует 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 приём 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.

6.17.10 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 четырёх флагов:

- флаг ошибки;
- флаг успешного завершения транзакции в канале;
- флаг приёма достоверного КС, ОС или слова данных команды управления;
- флаг неактивности контроллера.

Каждый из флагов может быть маскирован битами разрешения прерывания по какому-либо флагу.

6.17.11 Описание регистров

Таблица 145 – Описание регистров

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x400A_6000		MIL-STD-1553B0		
0x400A_8000		MIL-STD-1553B1		
0x400A_A000		MIL-STD-1553B2		
0x400A_C000		MIL-STD-1553B3		
	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

6.17.11.1 Регистр CONTROL

Base ADDR=		0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000				Offset=		0x1000								
REG Name:																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
								RT_HW	INPINV	AUTOTUNE	ENFILTER	INVTR	RERR	DIV[6:5]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[4:0]					RTA4	RTA3	RTA2	RTA1	RTA0	TRB	TRA	RTMODE	BCMODE	BCSTART	MR

Бит	Имя	Значение	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 приёмника 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 положительный вход декодера манчестерского кода. Применение этого бита актуально для приёмопередатчиков с принудительной установкой выхода приемника в состояние логическая «1», например, 5559IH74T (H1574)
21	AUTOTUNE		Бит автоматической подстройки середины битовых интервалов 0 – автоподстройка разрешена 1 – автоподстройка запрещена
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 5559IH67T, необходимо устанавливать бит в единицу для корректного приёма. В этом случае контроль длительностей импульсов NRZ не осуществляется
19	INVTR		Разрешение инверсии сигналов MILx_ENA, MILx_ENB 1 – инверсия 0 – прямой выход
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 МКМО до 1 МГц Содержит значение, на которое необходимо поделить частоту PCLK, чтобы получить 1 МГц. Частота PCLK обязательно должна быть не более 120 МГц и кратна 8. Если PCLK не кратна 8, $DIV[6:3]=(PCLK/8)+1$, $DIV[2:0]=0$, но стабильность приёма 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

Бит	Имя	Значение	Описание
1	BCSTART		Иницирует передачу сообщения в канал в режиме КШ. 1 – старт сообщения 0 – стоп сообщения Сбрасывается в ноль автоматически по завершению сообщения
0	MR		Сброс контроллера. 1 – контроллер сбрасывается в исходное состояние 0 – разрешение работы контроллера

6.17.11.2 Регистр STATUS

Base ADDR=	0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000	Offset=	0x1004												
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
					RCVB_ stick	RCVA_ stick	ERR_ stick	VALMESS_ stick	RFLAGN_ stick	RCVB	RCVA	ERR	VALMESS	RFLAGN	IDLE

Бит	Имя	Значение	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 относиться 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 относиться 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 приём и передачу пакетов. Необходимо следить за своевременным сбросом, иначе бит может относиться не к своему пакету

Бит	Имя	Значение	Описание
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 относиться не к своему пакету
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 относиться 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

6.17.11.3 Регистр ERROR

Base ADDR=	0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000	Offset=	0x1008												
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
									PROERR	CONERR	GAPERR	CSYCERR/S EQERR	DSYCERR/S YNCERR	MANERR	NORCV

Бит	Имя	Значение	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 приёма команды в режиме ОУ и М (SEQERR). 1 – обнаружена пауза после приёма командного слова с битом 10 равным нулю или обнаружены слова данных после приёма командного слова с битом 10 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 NRZ кода. 1 – ошибка в количестве принятых бит или ошибка в бите контроля чётности 0 – ошибок нет
0	NORCV		Ошибка приёма. 1 – не получено ответное слово в интервале 14 мкс или не получены ожидаемые данные 0 – ошибок нет

6.17.11.4 Регистр CommandWord1

Base ADDR=		0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000				Offset=		0x100C							
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
Адрес ОУ					Приём/ Передача	Подадрес/ Режим управления					Число СД/ Код команды				

Бит	Имя	Значение	Описание
31..16	-		Зарезервировано
15..11	Адрес ОУ		Адрес оконечного устройства, которому предназначено командное слово
10	Приём/передача		Бит приёма/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.

6.17.11.5 Регистр CommandWord2

Base ADDR=		0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000				Offset=		0x1010							
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
Адрес ОУ					1	Подадрес/ Режим управления					Число СД/ Код команды				

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..11	Адрес ОУ		Адрес оконечного устройства, которому предназначено командное слово
10	Приём/передача		Бит приёма/передачи 1 – режим работы ОУ-ОУ 0 – командное слово не используется
9..5	Подадрес		Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД
4..0	Число СД		Содержит количество принимаемых или передаваемых слов данных

Примечание – В режиме ОУ и М регистр доступен только на чтение и содержит второе командное слово транзакции ОУ-ОУ. В режиме КШ регистр доступен только на запись и используется для транзакции ОУ-ОУ, если установлен в единицу бит 10.

6.17.11.6 Регистр ModeData

Base ADDR=		0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000				Offset=		0x1014							
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
Слово данных команды управления															

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..0	ModeData		Содержит принятое или передаваемое слово данных в команде управления

6.17.11.7 Регистр StatusWord1

Base ADDR=		0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000				Offset=		0x1018							
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
				Адрес ОУ	Ошибка в сообщ.	Пер.ОС	Запр. на обл.	.	.	.	Прин. ГК	Абон. зан.	Неисп. абон.	Прин. упр. инт.	Неисп. ОУ

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..11	Адрес ОУ		Адрес ОУ, от которого принято ответное слово в режиме КШ. Адрес ОУ, которое передаёт ответное слово в режиме ОУ
10	Ошибка в сообщ.		Ошибка в сообщении
9	Пер.ОС		Передача ответного слова
8	Запр. на обл.		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.

6.17.11.8 Регистр StatusWord2

Base ADDR=		0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000				Offset=		0x101C							
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
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Адрес ОУ					Ошибка в сообщ.	Пер. ОС	Запр. на обл.				Прин. ГК	Абонан.	Неисп. абон.	Прин. упр. инт.	Неисп. ОУ

Бит	Имя	Значение	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 обл.		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.

6.17.11.9 Регистр INTEN

Base ADDR=		0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000		Offset=		0x1020									
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
												ERRIE	VALMESSIE	RFLAGIE	IDLEIE

Бит	Имя	Значение	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 приёме достоверного слова. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при приёме 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

6.17.11.10 Регистр MSG

Base ADDR=	0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000	Offset=	0x1024												
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
		Код сообщения													

Бит	Имя	Значение	Описание
31...14	-		Зарезервировано
13...0	Код сообщения		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 146 – 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 приёма КШ-ОУ, не групповая	RTA 0 00001-11110 XXXXX	
0080 Команда приёма КШ-ОУ, групповая	11111 0 00001-11110 XXXXX	
0004 Команда приёма ОУ-ОУ, не групповая	RTA 0 00001-11110 XXXXX	XXXXX 1 00001-11110 XXXXX
0100 Команда приёма 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

Код сообщения	CommandWord1	CommandWord2
<i>Команда управления</i>		
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	

6.17.11.11 Регистр DATA

Base ADDR=		0x400A_6000 0x400A_8000 0x400A_A000 0x400A_C000				Offset=		0x0000- 0x0FFC							
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[15:0]															

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0			Данные читаются из памяти или записываются в память в соответствии с поадресом (биты с 9 по 5) достоверного командного слова. Каждому поадресу соответствует 32x16 ячеек памяти на приём и 32x16 ячеек памяти на передачу. Общий объём памяти данных 2Kx16

6.18 Контроллер приемника телекомандной информации (CCSDS_RX)

Структурная схема приемника телекомандной информации приведена на рисунке 128.

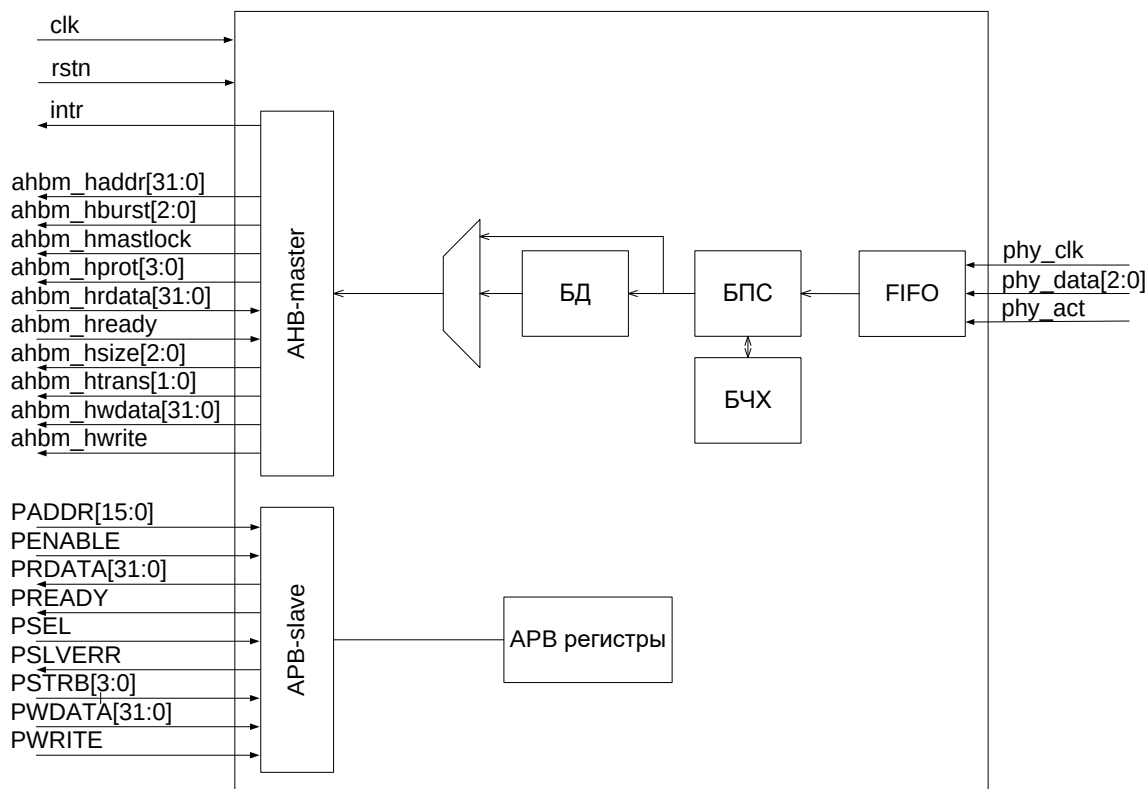


Рисунок 128 – Структурная схема приемника телекомандной информации

Приемник телекомандной информации состоит из следующих функциональных блоков:

- блок поиска синхромаркера (БПС);
- декодер БЧХ;
- блок дерандомизатора (БД).

6.18.1 Блок приемника телекомандной информации

Функциональный блок приемника телекомандной информации предназначен для приема битового потока данных, закодированных в соответствии со стандартом CCSDS 231.0-B-2, его декодирования и записи полученных данных в пользовательскую память.

Интерфейс физического уровня содержит входной тактовый сигнала **phy_clk**, сигнал активности данных **phy_act** и шину данных **phy_data[2:0]**. Логическая "1" на шине данных передается в виде высокого уровня сигнала, логический "0" – в виде низкого уровня. Используется только один из входов шины данных **phy_data[0]**.

Данные с шины считываются в блок по возрастающему фронту сигнала **phy_clk**. Высокий уровень сигнала **phy_act** свидетельствует о наличии данных на шине в момент прихода возрастающего фронта сигнала **phy_clk**. Для корректной работы декодеров блока частота тактового входного сигнала F_{phy_clk} должна быть меньше системной тактовой частоты F_{SYS} в соответствии с условием

$$F_{phy_clk} \leq 0,4 \cdot F_{SYS} . \quad (26)$$

Информация о входном интерфейсе физического уровня приведена в таблице 147.

Таблица 147 – Входной интерфейс приемника телекомандной информации

Наименование сигнала	Направление	Функциональное назначение
phy_clk	Вход	Входной тактовый сигнал
phy_act	Вход	Признак наличия данных на шине
phy_data[0]	Вход	Входные данные, считываются по возрастающему фронту тактового сигнала
phy_data[2:1]	Вход	Не используется

Временная диаграмма сигналов входного интерфейса изображена на рисунке 129.

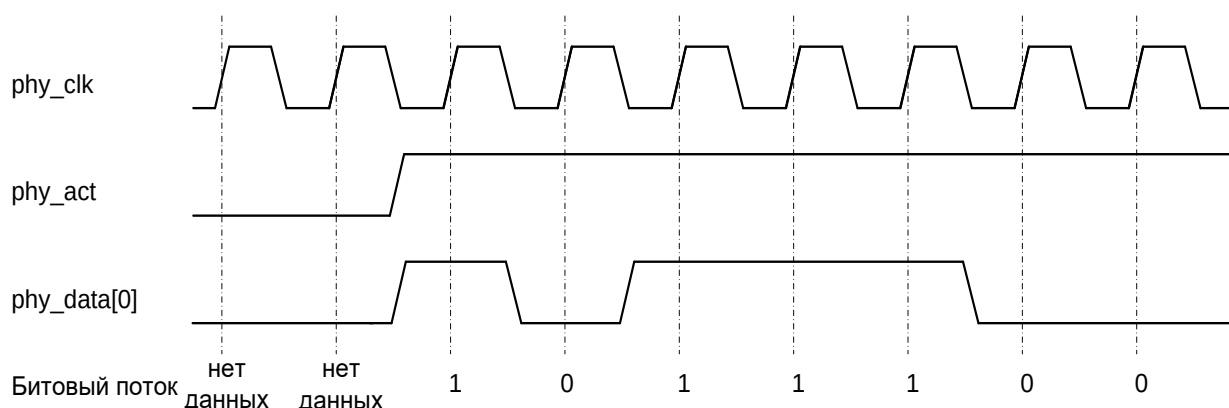


Рисунок 129 – Входной интерфейс приемника телекомандной информации

Размер входного FIFO равен 256 бит.

Входной битовый поток поступает из входного FIFO на блок поиска синхромаркера. В соответствии со стандартом синхромаркер представляет из себя фиксированную последовательность из 16 бит, значение которой устанавливается программно. Если во входном потоке встречается указанная последовательность, это трактуется блоком БПС как начало кадра, который передается для дальнейшей обработки.

Согласно стандарту, кадр состоит из 64-битных слов, закодированных кодом БЧХ (63,56) (64-й бит всегда равен нулю). Декодер БЧХ позволяет найти и исправить одну битовую ошибку в каждом кодовом слове или обнаружить двойную ошибку.

Если БЧХ декодер обнаруживает неисправимую ошибку, то есть если в кодовом слове искажено два или более бита данных, прием пакета завершается, при этом в принимающем дескрипторе выставляется флаг ошибки.

Если в передающей системе данные были рандомизированы, декодированный битовый поток должен быть подан на дерандомизатор путем настройки соответствующих регистров приемника телекомандной информации.

В конце каждого принимаемого пакета должна присутствовать завершающая последовательность, имеющая размер 64 бита. Прием такой последовательности свидетельствует об окончании приема пакета. Блок приема телекомандной информации никак не ограничивает максимальный размер принимаемого пакета.

Передача декодированных данных в область пользовательской памяти осуществляется при помощи системы дескрипторов. Поддерживается шестнадцать дескрипторов (0..15), каждый из которых при открытии позволяет принять от 1 до 1024 байт декодированных данных. При приеме битового потока дескрипторы

используются последовательно, то есть при завершении работы дескриптора с номером N активен становится дескриптор с номером N+1, если он был до этого настроен на работу. Если же этот дескриптор не был открыт к моменту переключения на него, прием данных остановится и при наличии данных на входном интерфейсе произойдет переполнение входного FIFO, что приведет к потере данных. Если для дальнейшей работы требуется опустошить FIFO, необходимо перезапустить контроллер.

После закрытия дескриптора с номером 15 активным становится дескриптор с номером 0. Если пользователь решает использовать не весь набор дескрипторов, переключение между ними можно производить программно. Номер активного дескриптора можно изменить, но только в момент, когда текущий активный дескриптор не осуществляет прием данных. По умолчанию активным является дескриптор с наименьшим номером. В то время пока активный дескриптор не готов к работе, поступающие на вход данные накапливаются во входном FIFO, что может привести к его переполнению и потере данных.

Если размер принимаемого пакета превышает размер активного дескриптора, по заполнению дескриптор будет закрыт, а данные продолжат передаваться с помощью следующего дескриптора. В этом случае для анализа принятых данных в дескрипторах используются флаги, с помощью которых можно определить, что дескриптор принял начало или конец пакета.

По закрытию каждого дескриптора происходит вызов прерывания (сигнал intr).

Прерывание является импульсным. Длительность импульса равна одному такту системной частоты.

Тактирование блока осуществляется сигналом clk. Сброс блока осуществляется сигналом rstn, активный уровень – логический «0». Интерфейсы контроллера ANB и APB соответствуют стандарту AMBA-Lite.

6.18.2 Программная модель управления приемником телекомандной информации

Для приема телекомандной информации необходимо осуществить предварительную настройку блоков декодирования, затем настроить и включить дескрипторы в работу. Управление блоком осуществляется через запись и чтение регистров на шине APB. Последовательность действий в общем случае описана ниже.

- 1 Отключить блоки приемника телекоманд записью в регистр GLOBAL_ENABLE значения 0. Это необходимо для того, чтобы вносимые изменения вступили в силу после включения блоков, т.к. обновление параметров блоков происходит по переходу GLOBAL_ENABLE из 0 в 1.
- 2 Произвести настройку блока поиска синхромаркера. Содержимое стартовой последовательности задается в регистре START_SEQ, содержимое хвостовой последовательности задается в регистрах TAIL1_SEQ и TAIL2_SEQ. Путем задания специального бита в регистре START_SEQ можно включить разрешение единичной битовой ошибки в стартовой последовательности. В этом случае последовательность бит, отличающаяся одним разрядом от установленной в регистре, также будет считаться корректной стартовой последовательностью.
- 3 Произвести включение/отключение отдельных функциональных блоков: БЧХ-декодера, рандомизатора. Блоки включаются в соответствующих полях регистра CODER_SEL. Регистр GLOBAL_ENABLE используется для включения декодеров в работу, при этом обновленные значения из регистров CODER_SEL окажут влияние только по переходу содержимого GLOBAL_ENABLE из 0 в 1.
- 4 Включить необходимые дескрипторы в работу. Для включения каждого дескриптора используется по два регистра. Регистр DSCR_PTRx используется

для указания начала области пользовательской памяти, в которую будет производиться запись декодированных данных. Для открытия дескриптора регистр DSCR_FLAG_SIZE_x используется следующим образом:

- задается размер дескриптора в поле DATA_SIZE (от 1 до 1024 байт);
- дескриптор включается в работу записью 1 в поле DSCR_READY. Для работы с большими объемами входных данных, поступающих непрерывно, рекомендуется использовать все 16 дескрипторов, каждый из которых предварительно открывается на прием максимального количества данных (1024 байта).

5 Задать номер активного дескриптора в регистре DSCR_CURRENT. Прием данных начнется с данного дескриптора.

6 Запустить блоки приемника телекоманд в работу записью в GLOBAL_ENABLE значения 1.

7 По мере поступления данных на вход блока приемника телекоманд активный в настоящий момент дескриптор будет производить запись данных в пользовательскую память после прохождения их через декодирующие модули. Дескриптор будет закрыт в одном из случаев:

- количество принятых данных превысило выделенный в дескрипторе размер;
- пакет завершился до того, как количество принятых данных превысило выделенный в дескрипторе размер;
- декодером БЧХ была обнаружена неисправимая ошибка при приеме пакета;
- переход сигнала phy_act из 1 в 0.

В каждом из перечисленных случаев управление будет передано следующему по номеру дескриптору.

Если следующий дескриптор в этот момент открыт, поступающие далее на вход данные продолжают записываться в пользовательскую память в соответствии с указанным в дескрипторе адресом.

Если же следующий дескриптор закрыт, данные будут накапливаться во входном FIFO. Если в буфере FIFO закончится место для приема последующих данных, будет выставлен флаг переполнения FIFO в регистре PHY_FIFO_OVERFLOW. В случае переполнения запись последующих пакетов в FIFO будет приостановлена. Для дальнейшей работы необходимо открыть текущий дескриптор либо сменить номер текущего дескриптора на открытый. После этого следует сбросить флаг переполнения PHY_FIFO_OVERFLOW.

8 После закрытия дескриптора происходят следующие события:

- выставляется прерывание;
- в поле DATA_SIZE регистра DSCR_FLAG_SIZE_x записывается количество принятых дескриптором данных;
- поле DSCR_READY регистра DSCR_FLAG_SIZE_x сбрасывается в 0;
- в поле DSCR_FLAGS регистра DSCR_FLAG_SIZE_x записывается значение флагов о получении начала пакета, конца пакета или ошибки декодирования;
- поле DSCR_NUM регистра DSCR_CURRENT инкрементируется. Если до этого значение DSCR_NUM было равно 15, в него запишется значение 0.

9 После обработки полученных данных с помощью завершившего работу дескриптора необходимо заново перезапустить дескриптор (см. п. 5) для приема последующих поступающих данных.

6.18.3 Описание регистров приёмника телекомандной информации

Состав регистров приемника телекомандной информации приведен в таблице 148.

Таблица 148 – Состав регистров приемника телекомандной информации

Смещение	Регистр	Описание
0x00	DSCR_PTR0	Указатель на начало области памяти для дескриптора 0
0x04	DSCR_FLAG_SIZE0	Управление дескриптором 0
0x08 ... 0x74	DSCR_PTR1 DSCR_FLAG_SIZE1 ... DSCR_PTR14 DSCR_FLAG_SIZE14	
0x78	DSCR_PTR15	Указатель на начало области памяти для дескриптора 15
0x7C	DSCR_FLAG_SIZE15	Управление дескриптором 15
0x80	DSCR_CURRENT	Текущий дескриптор
0x84	GLOBAL_ENABLE	Включение блока в режим приема данных
0x88	CODER_SEL	Настройка декодеров
0x0C	PHY_FIFO_OVERFLOW	Статус переполнения входного FIFO
0x90	START_SEQ	Содержимое стартовой последовательности
0x94	TAIL_SEQ1	Содержимое хвостовой последовательности
0x98	TAIL_SEQ2	Содержимое хвостовой последовательности
0x9C-0xFFFF	-	Не используется, читается 0

6.18.3.1 Регистры DSCR_PTRx

Номер	31:2	1	0
Доступ	RW		
Сброс	0		
	DSCR_PTR[31:2]		

Бит	Имя	Значение	Описание
31:2	DSCR_PTR		Указатель на начало области во внутренней памяти, предназначенной для записи принятых данных. Начало области должно быть выровнено по 4 байтной границе, младшие 2 бита указателя – нулевые
1, 0	-		-

6.18.3.2 Регистры DSCR_FLAG_SIZEx

Номер	31	30:19	18	17	16	15:10	9:0
Доступ	RW		R	R	R		RW
Сброс	0						0
	DSCR_READY	-	ERR	EOF	SOF	-	DATA_SIZE

Бит	Имя	Значение	Описание
31	DSCR_READY		Указатель на начало области во внутренней памяти, предназначенной для записи принятых данных. Начало области должно быть выровнено по 4 байтной границе, младшие 2 бита указателя – нулевые
1, 0	-		-

Бит	Имя	Значение	Описание
31	DSCR_READY		Необходимо программно установить в «1», когда область памяти, на которую указывает этот дескриптор, освобождена и готова для записи принятых данных. Когда все принятые данные записаны в область, на которую указывает этот дескриптор, то в этот бит аппаратно сбрасывается в 0
30:19	-		-
18	ERR		Флаг ошибки устанавливается в «1», если принятый пакет завершается не маркером хвоста, а неисправимой ошибкой БЧХ-декодера
17	EOF		Устанавливается в «1», если записанные принятые данные содержат последний байт пакета. Используется для случаев, когда принятый пакет записывается в несколько дескрипторов
16	SOF		Устанавливается в «1», если записанные принятые данные содержат первый байт пакета. Используется для случаев, когда принятый пакет записывается в несколько дескрипторов
15:10	-		-
9:0	DATA_SIZE		Размер области, в которую будут записываться принятые данные. 0 означает, что размер области памяти 1024 байта. Если размер пакета принятых данных меньше размера области, по окончании записи данных в это поле будет аппаратно записан размер записанных данных. Если размер пакета больше, то поле останется без изменения, а остаток данных запишется в следующий дескриптор. Флаги записываются аппаратно, когда все принятые данные записаны в область, на которую указывает этот дескриптор

6.18.3.3 Регистр DSCR_CURRENT

Номер	31	30:4	3:0
Доступ	RO		RW
Сброс			
	WORK_FLAG	-	DSCR_NUM

Бит	Имя	Значение	Описание
31	WORK_FLAG		Устанавливается в «1», если текущий дескриптор находится в работе и устанавливается в «0», когда текущий дескриптор не готов, то есть когда флаг текущего дескриптора DSCR_READY находится в «0»
30:4	-		-
3:0	DSCR_NUM		Номер текущего дескриптора. RO когда WORK_FLAG равен «1», и автоматически инкрементируется после отработки дескриптора. Когда WORK_FLAG равен «0», номер дескриптора можно перезаписать программно

6.18.3.4 Регистр GLOBAL_ENABLE

Номер	31:1	0
Доступ		RW
Сброс		

	-	GLOBAL_ENABLE
--	---	---------------

Бит	Имя	Значение	Описание
31:1	-		-
0	GLOBAL_ENABLE		0 – все декодеры, FIFO, AHB-контроллер находятся в асинхронном сбросе. 1 – включение всех модулей, ответственных за прием данных. Значение по сбросу: 0. Примечание – конфигурация декодеров, которая задается через регистры CODER_SEL, START_SEQ, TAIL_SEQ1, TAIL_SEQ2 загружается только в момент перехода GLOBAL_ENABLE из «0» в «1». Изменение значений этих регистров во время, когда GLOBAL_ENABLE = 1, не влияет на работу блока. Чтобы новые значения вступили в силу, требуется пересбросить блок через GLOBAL_ENABLE

6.18.3.5 Регистр CODER_SEL

Номер	31:2	1	0
Доступ		RW	RW
Сброс			
	-	BCH_SEL	RAND_SEL

Бит	Имя	Значение	Описание
31:2	-		-
1	BCH_SEL		1 – выбор рандомизатора
0	RAND_SEL		1 – выбор декодера БЧХ, исправляет одну ошибку в 7-ми байтах данных

6.18.3.6 Регистр PHY_FIFO_OVERFLOW

Номер	31:1	0
Доступ		RW1C
Сброс		
	-	PHY_FIFO_OVERFLOW

Бит	Имя	Значение	Описание
31:1	-		-
0	PHY_FIFO_OVERFLOW		Флаг переполнения входного FIFO. Флаг сбрасывается записью 1

6.18.3.7 Регистр START_SEQ

Номер	31	30:16	15:0
Доступ			
Сброс	0		
	ALLOW_1_ERR	-	START_SEQ

Бит	Имя	Значение	Описание
31	ALLOW_1_ERR		1 – позволить единичную ошибку в маркере начала пакета
30:16	-		-

Бит	Имя	Значение	Описание
15:0	START_SEQ		Маркер начала пакета. Первый принятый бит сравнивается с 15-ым битом START_SEQ, последний принятый бит сравнивается с 0-ым битом. В соответствии со стандартом битам START_SEQ должно быть присвоено значение 0xEB20

6.18.3.8 Регистры TAIL_SEQ1, TAIL_SEQ2

Номер	31:0
Доступ	RW
Сброс	0
	TAIL_SEQ

Бит	Имя	Значение	Описание
31:0	TAIL_SEQ		Маркер конца пакета. Первый принятый бит сравнивается с 31-ым битом регистра TAIL_SEQ1, последний принятый бит сравнивается с 0-ым битом TAIL_SEQ2. В соответствии со стандартом битам регистра должно быть присвоено значение: TAIL_SEQ1 = 0xC5C5C5C5; TAIL_SEQ2 = 0xC5C5C579

6.19 Контроллер передатчика телеметрической информации (CCSDS_TX)

Структурная схема передатчика телеметрической информации (ПдТМИ) приведена на рисунке 130.

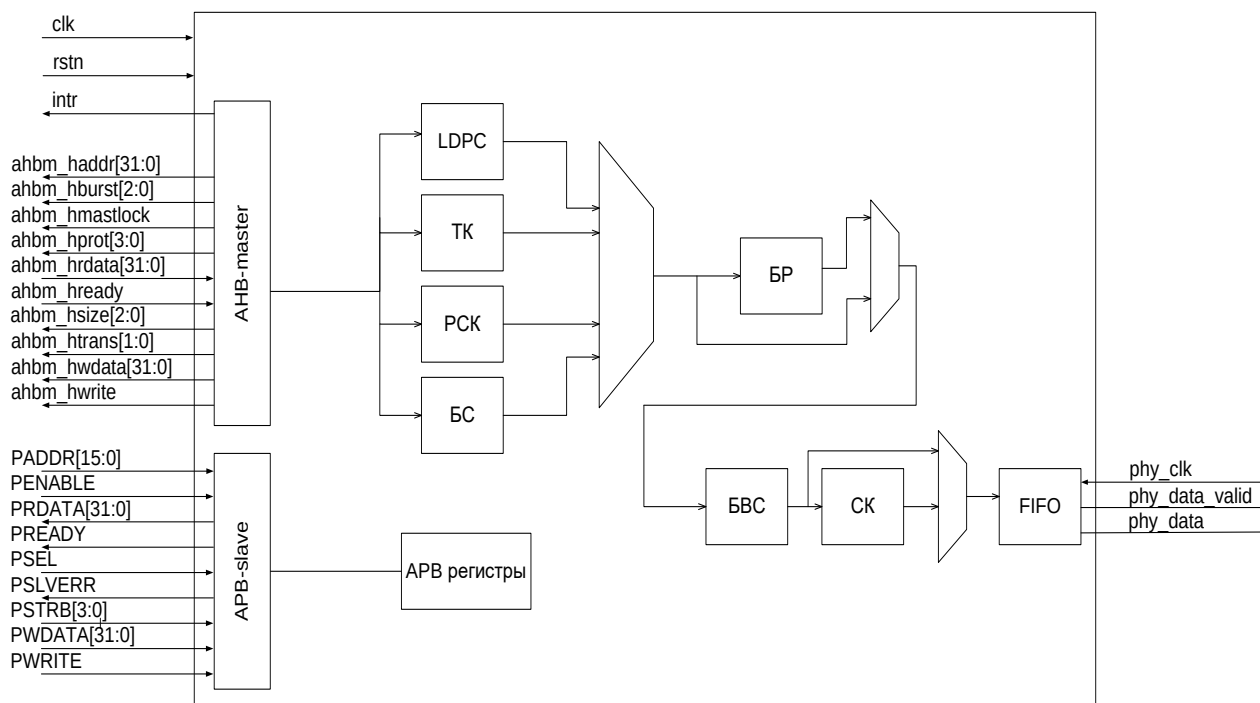


Рисунок 130 – Структурная схема ПдТМИ

ПдТМИ состоит из следующих составляющих блоков:

- кодер LDPC;
- турбо-кодер (ТК);

- кодер Рида-Соломона (РСК);
- блок сериалайзера (БС);
- блок рандомизатора (БР);
- блок вставки синхромаркера (БВС);
- сверточный кодер (СК).

6.19.1 Описание работы блока передатчика телеметрической информации

Сложный функциональный блок передатчика телеметрической информации функционирует согласно стандарту CCSDS 131.0-B-2. ПдТМИ предназначен для помехоустойчивого кодирования данных из заданной области пользовательской памяти, их компоновки в кадры со вставкой синхромаркеров и выдачи данных в виде битового потока на выходную шину.

Мастер на шине AMBA АНВ, управляемый посредством настройки двух дескрипторов, осуществляет передачу данных из пользовательской памяти на вход выбранного блока помехоустойчивого кодирования. В каждом дескрипторе задается адрес начала области данных для передачи и количество передаваемых данных, и затем подается команда на включение дескриптора. Задаваемое количество данных является целым числом байт. В то время, пока один дескриптор осуществляет передачу данных на кодеры передатчика, второй дескриптор доступен для настройки. Данные передаются на кодер младшим битом вперед.

Таким образом достигается непрерывность передаваемого битового потока. По завершению работы дескриптора вызывается прерывание, после чего возможна повторная настройка дескриптора для последующей передачи.

Кодирование битового потока осуществляется одним из алгоритмов помехозащищенного кодирования:

- LDPC (8160, 7136);
- Turbo-кодирование со скоростями 1/2, 1/3, 1/4 или 1/6 с информационным блоком размером 1784 бита;
- кодирование Рида-Соломона (255,223) или (255, 239) без перемежения;
- сверточное кодирование со скоростями 1/2, 2/3, 3/4, 5/6 или 7/8;
- каскадное кодирование кодом Рида-Соломона и сверточным кодом.

Для обеспечения минимальной частоты перемены нулей и единиц в передаваемом битовом потоке возможна рандомизация передаваемых данных.

Программная настройка позволяет пропускать данные насквозь без использования помехоустойчивого кодирования и рандомизации, при этом вместо блоков помехоустойчивого кодирования используется блок сериалайзера.

В соответствии со стандартом CCSDS данные телеметрии передаются в виде фреймов определенного размера. Перед началом каждого фрейма в битовом потоке выполняется вставка синхромаркера в виде заданной битовой последовательности. Блок вставки синхромаркера (БВС) осуществляет добавление этой последовательности с настраиваемым периодом, длиной и содержимым.

После прохождения через блоки кодирования данные поступают на выходной буфер типа FIFO, откуда передаются на внешнюю шину при наличии входного тактового сигнала `phy_clk`. Отслеживать состояние наполненности буфера можно по специальному флагу, выставление которого свидетельствует о том, что буфер пуст. Размер выходного буфера FIFO равен 256 бит.

Передача данных осуществляется по одноканальной шине `phy_data` и тактируется внешним входным тактовым сигналом `phy_clk`. По возрастающему фронту сигнала

phy_clk на шину phy_data выдается следующий бит кодированного потока данных и сигнал phy_data_valid принимает значение "высокий уровень".

Логическая "1" кодируется на шине высоким уровнем, а логический "0" – низким уровнем сигнала. Максимальное значение частоты входного тактового сигнала F_{phy_clk} , определяющей скорость выходного битового потока, зависит от включенных кодеков, но при любых настройках не должно быть более половины системной частоты F_{sys} :

$$F_{phy_clk} \leq \frac{F_{sys}}{2}. \quad (27)$$

Информация о выходном интерфейсе физического уровня обобщена в таблице 149.

Таблица 149 – Выходной интерфейс передатчика телеметрической информации

Наименование сигнала	Направление	Функциональное назначение
phy_clk	Вход	Входной тактовый сигнал
phy_data_valid	Выход	Признак наличия данных на шине
phy_data	Выход	Выходные данные, обновляются по возрастающему фронту тактового сигнала

Временная диаграмма сигналов выходного интерфейса изображена на рисунке 131.

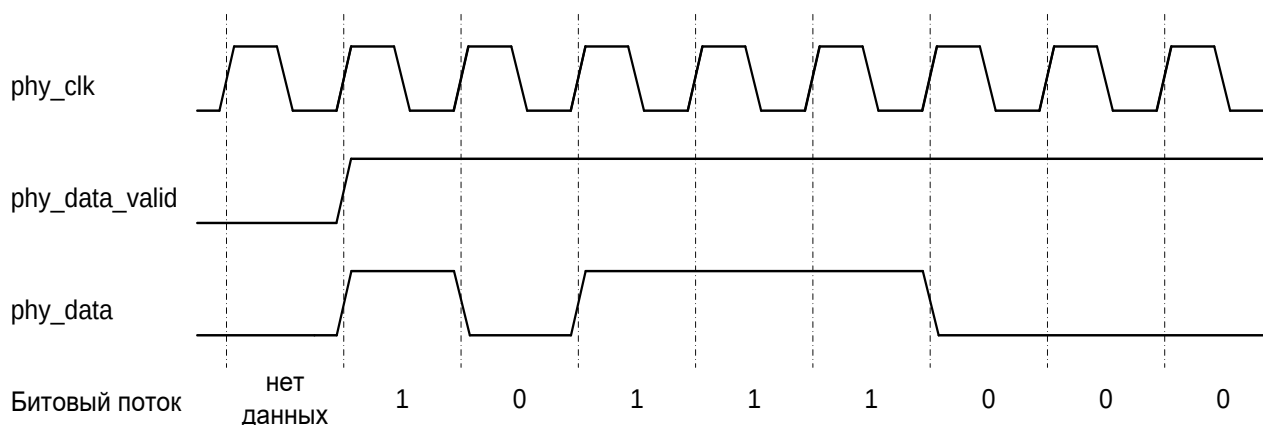


Рисунок 131 – Выходной интерфейс передатчика телеметрической информации

По закрытию каждого дескриптора происходит вызов прерывания (сигнал intr). Прерывание является импульсным. Длительность импульса равна одному такту системной частоты.

Тактирование блока осуществляется сигналом clk. Сброс блока осуществляется сигналом rstn, активный уровень – логический «0». Интерфейсы контроллера АНВ и АРВ соответствуют стандарту AMBA-Lite.

6.19.2 Программная модель управления передатчиком телеметрической информации

Для передачи телеметрической информации необходимо осуществить предварительную настройку блоков кодирования, затем настроить и включить дескрипторы в работу. Управление блоком осуществляется через запись и чтение регистров на шине АРВ. Последовательность действий в общем случае следующая:

- 1 Отключить блоки кодирования записью в регистр GLOBAL_ENABLE значения 0. Это необходимо для того, чтобы вносимые изменения вступили в силу после включения блоков, т.к. обновление параметров блоков происходит по переходу GLOBAL_ENABLE из 0 в 1.
- 2 С помощью регистра CODER_SEL включить в работу необходимый блок помехозащищенного кодирования, рандомизатор, сверточный кодер.
- 3 Произвести настройку кодирующих блоков в регистре CODER_CONFIG. Каждый блок может работать в одном из нескольких режимов, поддерживаемых стандартом.
- 4 Произвести настройку синхромаркера. В регистре MARKER_AND_FRAME_SIZE необходимо настроить период вставки синхромаркера и его размер в битах (не более 192 бит). Для работы в соответствии со стандартом CCSDS необходимо настроить период вставки синхромаркера (размер фрейма) в зависимости от выбранных кодеков. То есть задаваемый период должен быть равен размеру фрейма после проведения кодирования (см. таблицу). Регистры MARKER1 – MARKER6 определяют содержимое синхромаркера.

Т а б л и ц а 150 – Данные для определения размеров фрейма в зависимости от выбранного кодера и его режима работы

Тип кодера	Размер фрейма информационных данных на входе кодера (в байтах)	Размер фрейма на выходе кодера (в битах)
Reed-Solomon (255,239)	239	$255 \times 8 = 2040$
Reed-Solomon (255,223)	223	$255 \times 8 = 2040$
Turbo coder, rate 1/2	223	$223 \times 8 \times 2 + 4 \times 2 = 3576$
Turbo coder, rate 1/3	223	$223 \times 8 \times 3 + 4 \times 3 = 5364$
Turbo coder, rate 1/4	223	$223 \times 8 \times 4 + 4 \times 4 = 7152$
Turbo coder, rate 1/6	223	$223 \times 8 \times 6 + 4 \times 6 = 10728$
LDPC coder (8160, 7136)	892	8160

- 5 Настроить дескрипторы на передачу. Для управления каждым дескриптором используется два регистра DSCRx_PTR и DSCRx_FLAG_SIZE ($x = 1, 2$). Регистр DSCRx_PTR задает адрес начала области данных в пользовательской памяти, с которого начнется вычитывание данных для передачи на блоки кодирования. Регистр DSCRx_FLAG_SIZE определяет количество вычитываемых байт (от 1 до 2048) и включает дескриптор в работу.
- 6 Запустить блоки кодирования в работу записью в GLOBAL_ENABLE значения 1. После этого открытые дескрипторы начнут вычитывание данных. Первым начинает работу дескриптор с номером 1. Также после начала передачи данных следует сбросить флаг опустошения FIFO путем записи в регистр PHY_FIFO_UNDERRUN значения 1. Это позволит отслеживать состояние передающего буфера во время работы.
- 7 По завершении работы дескриптора вызывается прерывание. Для обоих дескрипторов используется один общий вектор прерывания. Определить, какой из дескрипторов завершил работу и готов для дальнейшей настройки, можно по состоянию бита DATA_PRES в регистре DSCRx_FLAG_SIZE. Если необходимо передавать поток данных большего размера, чем может быть

помещен в два дескриптора, потребуется повторное открытие дескриптора после его закрытия. При этом для обеспечения непрерывности битового потока на выходе блока дескриптор должен быть повторно настроен до того, как второй дескриптор завершит работу.

- 8 После того, как все необходимые данные были переданы, флаг PHY_FIFO_UNDERRUN будет выставлен в 1. Это свидетельствует о том, что блоки кодирования полностью завершили обработку и передачу данных. Если флаг выставился до того, как пользователь передал все необходимые данные, это значит, что нарушилась непрерывность передачи данных вследствие того, что дескрипторы передавали данные в FIFO с меньшей скоростью, чем данные выдавались на выходную шину.

6.19.3 Регистры передатчика телеметрической информации

Перечень регистров передатчика телеметрической информации приведен в таблице 151.

Т а б л и ц а 151 – Перечень регистров передатчика телеметрической информации

Смещение	Регистр	Описание
0x00	DSCR1_PTR	Дескриптор 1: указатель на область памяти
0x04	DSCR1_FLAG_SIZE	Дескриптор 1: флаги и размер области памяти
0x08	DSCR2_PTR	Дескриптор 2: указатель на область памяти
0x0C	DSCR2_FLAG_SIZE	Дескриптор 2: флаги и размер области памяти
0x10	GLOBAL_ENABLE	Включение блока в режим передачи
0x14	CODER_SEL	Выбор кодеров
0x18	PHY_FIFO_UNDERRUN	Флаг опустошения FIFO
0x1C	MARKER_AND_FRAME_SIZE	Размер фрейма и размер синхромаркера
0x20	MARKER1	Содержимое синхромаркера
0x24	MARKER2	Содержимое синхромаркера
0x28	MARKER3	Содержимое синхромаркера
0x2C	MARKER4	Содержимое синхромаркера
0x30	MARKER5	Содержимое синхромаркера
0x34	MARKER6	Содержимое синхромаркера
0x38	CODERS_CONFIG	Настройки кодеров
0x3C-0xFFFF	-	Не используется, читается 0

6.19.3.1 Регистры DSCR1_PTR, DSCR2_PTR

Номер	31:2	1:0
Доступ	RW	
Сброс	0	
	DSCR_PTR	-

Бит	Имя	Значение	Описание
31:2	DSCR_PTR		Указатель начала передаваемой области данных во внутренней памяти. Начало области должно быть выровнено по 4-байтной границе, младшие 2 бита указателя нули
1:0	-		-

6.19.3.2 Регистры DSCR1_FLAG_SIZE, DSCR2_FLAG_SIZE

Номер	31	30:11	10:0
Доступ	RW		RW
Сброс	0		0
	DATA_PRES	-	DATA_SIZE

Бит	Имя	Значение	Описание
31	DATA_PRES		Следует записать «1», когда данные дескриптора готовы для передачи. Когда GLOBAL_ENABLE = 1, данный дескриптор будет задействован, и данные, на которые ссылается этот дескриптор, будут считываться из внутренней памяти и записываться во внутренние буферы кодеров. В этот момент изменение полей DSCR_PTR и DATA_SIZE дескриптора не оказывает влияния. Когда все данные будут считаны и записаны во внутренние буферы, в поле DATA_PRES будет аппаратно записан 0. Примечание – если установлены флаги DATA_PRES в двух дескрипторах, и аппаратный блок не работает ни с одним из них (это может быть только в первый момент после включения блока через GLOBAL_ENABLE), первым для передачи выбирается дескриптор номер 1(DSCR1)
30:11	-		-
10:0	DATA_SIZE		Размер передаваемой области данных в байтах минус 1. Например: если 0 – передается 1 байт; если 1 – передается 2 байта; если 2047 – передается 2048 байт

6.19.3.3 Регистр GLOBAL_ENABLE

Номер	31:1	0
Доступ		RW
Сброс		0
	-	GLOBAL_ENABLE

Бит	Имя	Значение	Описание
31:1	-		-

Бит	Имя	Значение	Описание
0	GLOBAL_ENABLE		0 – все кодеры, FIFO, АНВ-контроллер находятся в асинхронном сбросе. 1 – включение всех модулей, ответственных за передачу данных. Примечание – конфигурация кодеров, которая задается через регистры CODER_SEL, MARKER_AND_FRAME_SIZE, MARKER1-6, CODERS_CONFIG, загружается только в момент перехода GLOBAL_ENABLE из «0» в «1». При изменении значений этих регистров во время, когда GLOBAL_ENABLE = 1, не влияет на работу блока. Чтобы новые значения вступили в силу, требуется выполнить сброс блока через регистр GLOBAL_ENABLE

6.19.3.4 Регистр PHY_FIFO_UNDERRUN

Номер	31:1	0
Доступ		RW1C
Сброс		1
	-	PHY_FIFO_UNDERRUN

Бит	Имя	Значение	Описание
31:1	-		-
0	PHY_FIFO_UNDERRUN		Флаг опустошения асинхронного FIFO, нет данных для передачи на физический уровень. В этом случае сигнал phy_data_valid устанавливается в 0. Это может происходить вследствие длительного формирования данных для передачи во внутренней памяти, когда флаги DATA_PRESENCE обоих дескрипторов находятся в нуле длительное время. Чтобы этого избежать, необходимо оптимизировать работу программы или уменьшить частоту следования выходных данных за счет снижения тактовой частоты на входе phy_clk. Примечание – в самом начале работы при переходе GLOBAL_ENABLE из нуля в единицу, флаг устанавливается в «1», необходимо подождать, пока данные будут подаваться в выходной блок FIFO и далее сбросить флаг

6.19.3.5 Регистр CODER_SEL

Номер	31:4	3	2	1:0
Доступ		RW	RW	RW
Сброс		0		
	-	RAND_SEL	CONV_SEL	BIG_CODER_SEL

Бит	Имя	Значение	Описание
31:4	-		-
3	RAND_SEL		1 – включение рандомизатора

Бит	Имя	Значение	Описание
2	CONV_SEL		1 – включение сверточного кодера. Работа сверточного кодера предусмотрена стандартом только одиночно или совместно с кодером Рида-Соломона. Поэтому установка бита CONV_SEL в «1» допускается только совместно с установкой BIG_CODER_SEL в значения 0 или 3
1:0	BIG_CODER_SEL		Выбор одного из кодеров: 1 – LDPC-кодер; 2 – Turbo-кодер; 3 – RS-кодер; 0 – без LDPC, Turbo и RS-кодеров

6.19.3.6 Регистр MARKER_AND_FRAME_SIZE

Номер	31:24	23:16	15:0
Доступ		RW	RW
Сброс		0	0
	-	MARKER_SIZE	FRAME_SIZE

Бит	Имя	Значение	Описание
31:24	-		-
23:16	MARKER_SIZE		Размер тела синхромаркера в битах
15:0	FRAME_SIZE		Период вставки синхромаркера в битах

Примечания

- Размер фрейма определяется типом используемого помехозащищенного кодера. Пользователь должен высчитать размер фрейма с учетом дополнительных бит создаваемых кодером, и записать нужное значение в поле FRAME_SIZE.
- Согласно стандарту размер синхромаркера должен быть равен:
 - 32 – при работе без помехоустойчивого кодирования, применении сверточного, каскадного кодирования, кодирования Рида-Соломона или LDPC;
 - 64 – при применении турбо-кодирования со скоростью 1/2;
 - 96 – при применении турбо-кодирования со скоростью 1/3;
 - 28 – при применении турбо-кодирования со скоростью 1/4;
 - 192 – при применении турбо-кодирования со скоростью 1/6.
 Если размер маркера равен 0, то синхромаркер не вставляется.

6.19.3.7 Регистры MARKER1 – MARKER6

Номер	31:0
Доступ	RW
Сброс	0
	MARKER

Бит	Имя	Значение	Описание
31:0	MARKER		Содержимое синхромаркера. Первым передается бит с номером 0 регистра MARKER1, последним, если размер синхромаркера выставлен 192 бита, – 31-ый бит регистра MARKER6. Для задания синхромаркера согласно стандарту регистры MARKER1 – MARKER6 должны иметь следующие значения:

Бит	Имя	Значение	Описание
			– 32 – при работе без помехоустойчивого кодирования, применении сверточного, каскадного кодирования, кодирования Рида-Соломона или LDPC MARKER1 = 0xB83FF358; – 64 – при применении турбо-кодирования со скоростью 1/2; MARKER1 = 0xE36EE2C0; MARKER2 = 0x0DA914E4; – 96 – при применении турбо-кодирования со скоростью 1/3; MARKER1 = 0x7303ABA4; MARKER2 = 0x936F0991; MARKER3 = 0x39EFD862; – 128 – при применении турбо-кодирования со скоростью 1/4; MARKER1 = 0xE36EE2C0; MARKER2 = 0x0DA914E4; MARKER3 = 0x1C911D3F; MARKER4 = 0xF256EB1B; – при применении турбо-кодирования со скоростью 1/6; MARKER1 = 0x7303ABA4; MARKER2 = 0x936F0991; MARKER3 = 0x39EFD862; MARKER4 = 0x8CFC545B; MARKER5 = 0x6C90F66E; MARKER6 = 0xC610279D

6.19.3.8 Регистр CODERS_CONFIG

Номер	31:18	17:16	15:10	9:8	7:3	2:0
Доступ						
Сброс						
	-	RS_Type	-	Turbo_rate	-	Conv_rate

Бит	Имя	Значение	Описание
31:18	-		-
17:16	RS_Type		Тип кодера RS: [16] 0 – (255,239); 1 – (255,223); [17] 0 – обычный; 1 – dual basis converted
15:10	-		-
9:8	Turbo_rate		Тип Turbo кодера: 0 – rate 1/2; 1 – rate 1/3; 2 – rate 1/4; 3 – rate 1/6
7:3	-		-
2:0	Conv_rate		Тип сверточного кодера: 2 – rate 2/3; 3 – rate 3/4; 5 – rate 5/6; 7 – rate 7/8; остальные значения – rate 1/2

6.20 Контроллер блока расчета проверочных кодов (CRC)

Микроконтроллер имеет в своем составе блок вычисления CRC с произвольной степенью полинома от 4 до 64. Контроллер принимает согласно интерфейсу *AMBA® APBProtocolVersion: 2.0 Specification* (APB): 8, 16 и 32 битные слова и может их обрабатывать как в прямом порядке (начиная со старшего бита), так и в обратном (начиная с младшего бита).

6.20.1 Общие сведения о циклическом избыточном коде

Циклический избыточный код (англ. *Cyclic redundancy code, CRC*) – алгоритм вычисления контрольной суммы, предназначенный для проверки целостности передаваемых данных. Алгоритм CRC обнаруживает все одиночные ошибки, двойные ошибки и ошибки в нечетном числе битов.

6.20.2 Алгоритм CRC

Алгоритм CRC базируется на свойствах деления с остатком двоичных многочленов, то есть многочленов над конечным полем $GF(2)$. Значение CRC является по сути остатком от деления многочлена, соответствующего входным данным, на некий фиксированный порождающий многочлен.

Каждой конечной последовательности битов $a_0, a_1 \dots a_{N-1}$ взаимнооднозначно сопоставляется двоичный многочлен $\sum_{n=0}^{N-1} a_n x^n$, последовательность коэффициентов которого представляет собой исходную последовательность. Например, последовательность битов 1011010 соответствует многочлену

$$P(x) = 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 0 \cdot x^0 = x^6 + x^4 + x^3 + x^1. \quad (28)$$

Количество различных многочленов степени меньше N равно 2^N , что совпадает с числом всех двоичных последовательностей длины N .

Значение CRC с порождающим многочленом $G(x)$ степени N определяется как битовая последовательность длины N , представляющая многочлен $R(x)$, получившийся в остатке при делении многочлена $P(x)$, представляющего входной поток бит, на многочлен $G(x)$:

$$R(x) = P(x) \cdot x^N \bmod G(x), \quad (29)$$

где $R(x)$ — многочлен, представляющий значение CRC.

$P(x)$ — многочлен, коэффициенты которого представляют входные данные.

$G(x)$ — порождающий многочлен.

$N = \deg G(x)$ — степень порождающего многочлена.

Умножение x^N осуществляется приписыванием N нулевых битов к входной последовательности, что улучшает качество хеширования для коротких входных последовательностей.

При делении с остатком степень многочлена-остатка строго меньше степени многочлена-делителя, то есть при делении на многочлен $G(x)$ степени N можно получить 2^N различных остатков от деления. При «правильном» выборе порождающего многочлена $G(x)$, остатки от деления на него будут обладать нужными свойствами хеширования: хорошей перемешиваемостью и быстрым алгоритмом вычисления. Второе обеспечивается тем, что степень порождающего многочлена обычно пропорциональна длине байта или машинного слова (например, 8, 16 или 32).

Операция деления на примитивный полином также эквивалентна следующей схеме:

Пусть выбран примитивный полином, задающий цикл де Брейна 0010111001011100... и блок данных 0111110, построена таблица, верхняя строка заполнена блоком данных, а нижние строки – смещения на 0,1,2 бит цикла де Брейна



Тогда контрольная сумма будет равна операции XOR тех столбцов, над которыми в верхней строке расположена 1. В этом случае

$$010 \text{ xor } 101 \text{ xor } 011 \text{ xor } 111 \text{ xor } 110 = 101 \text{ (CRC).}$$

6.20.3 Пример вычисления CRC

Для вычисления CRC необходимо выбрать делитель (полином). Следует обратить внимание, что старший бит всегда должен быть равен 1, следовательно, после того, как выбрана степень полинома, необходимо подобрать лишь значения младших его битов.

Если необходимо вычислять CRC для последовательности бит, следует убедиться, что обработаны все биты. Поэтому в конце последовательности необходимо добавить W (степень полинома) нулевых бит.

Рассмотрим пример:

Полином = 10011, степень $W=4$

Последовательность бит + W нулей = 110101101 + 0000

10011/1101011010000\110000101 (частное от деления не используется)

```

10011 ||||| -
-----|||
10011 |||||
10011 ||||| -
-----|||
00001 |||
00000 ||| -
-----|||
00010 |||
00000 ||| -
-----|||
00101 |||
00000 ||| -
-----|||
01010 |||
00000 ||| -
-----|||
10100 |||
10011 ||| -
-----|||
01110 |||
00000 ||| -
-----|||
11100
10011 -
-----

```

1111 -> остаток от деления – CRC

Здесь необходимо отметить два важных момента:

- 1 Операция XOR выполняется только в том случае, когда старший бит последовательности равен 1, в противном случае мы просто сдвигаем ее на один бит влево.
- 2 Операция XOR выполняется только с младшими W битами, так как старший бит всегда в результате дает 0.

6.20.4 Реализация последовательного алгоритма CRC

Для последовательного алгоритма CRC поток данных, начиная со старшего разряда, должен быть постепенно разделен на заданный полином. Реализация представлена на рисунке 132. Для этого значение CRC задается последовательным сдвиговым регистром, а полином CRC запоминается в регистр хранения и управляет мультиплексорами, отвечающие за XOR значение прошлого CRC бита с обратной связью степени полинома с новым входным значением. Степень полинома задается мультиплексором обратной связи.

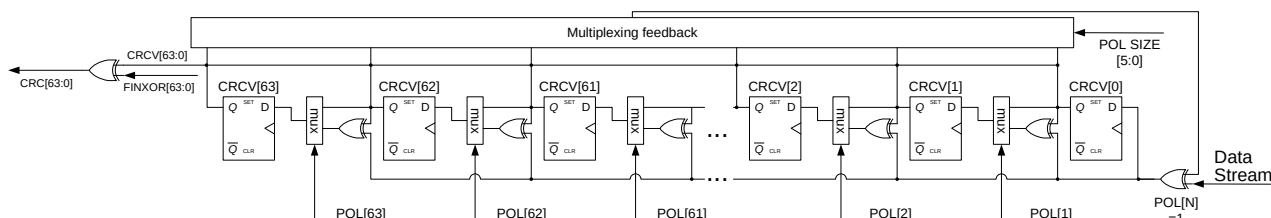


Рисунок 132 – Реализация последовательного алгоритма CRC

6.20.5 Реализация алгоритма CRC в микроконтроллере

Скорость подсчета CRC в микроконтроллере составляет 4 бит/PCLK (частота APB шины) в параллельном режиме работы и 1 бит/PCLK – в последовательном режиме работы. Для подсчета потока данных не кратных 8-ми используется последовательный режим подсчета CRC. Контроллер имеет FIFO глубиной 4 отчета, а также DMA канал для загрузки новых слов. Запрос для DMA формируется, если FIFO пусто, либо если FIFO не полное. Контроллер начинает обрабатывать новые слова, как только они появляются в FIFO и обрабатывает до последнего бита. После обработки последнего бита выставляется флаг завершения расчета CRC. Регистр CRC имеет доступ как на чтение (считать рассчитанное значение), так и на запись (установить начальное значение).

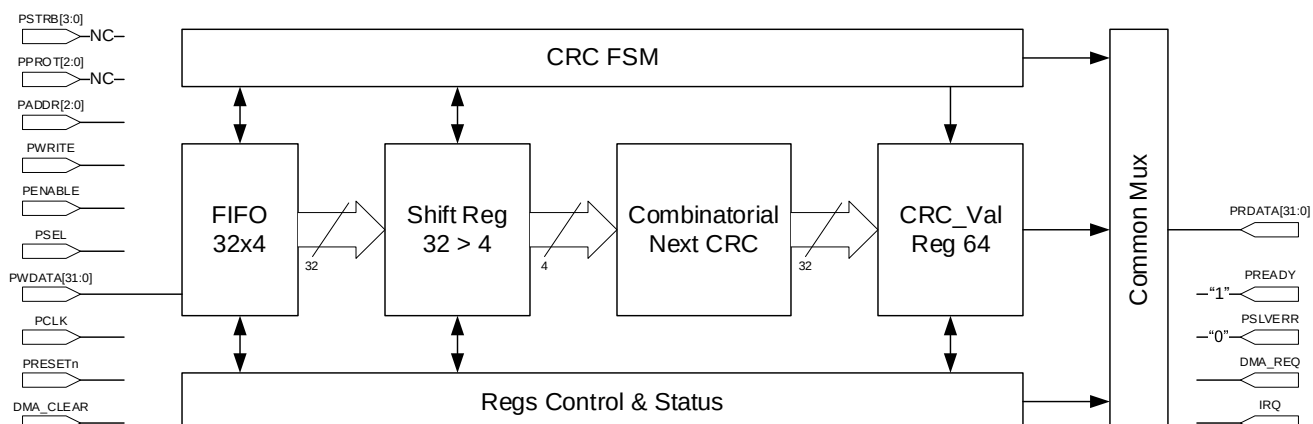


Рисунок 133 – Структурная схема модуля CRC

Сигналы шины APB: PSTRB, PPROT – не используются, PREADY – установлен всегда в «1», PSLVERR – «0».

Сигнал DMA_CLEAR – делает принудительный сброс DMA_REQ.

При записи данных в FIFO выравнивать записываемые данные необходимо к младшему биту (см. рисунок 134).

Слово 0	31:24	23:16	15: 8	7: 0
Слово 1	31:24	23:16	15: 8	7: 0
Слово 2	31:24	23:16	15: 8	7: 0
Слово 3	31:24	23:16	15: 8	7: 0
Слово 4	31:24	23:16	15: 8	7: 0
...	...			
Слово N-1	31:24	23:16	15: 8	7: 0
Слово N	L: 0			

Рисунок 134 – Порядок записи данных в FIFO (CRC_DIN)

При подсчете CRC *в последовательном режиме* (FlowEn=0, SerALG=1, регистр CRC_CTRL) за одну транзакцию можно загрузить и подсчитать от 1 до 32 бит (указать данное количество бит в поле SerDinN, регистр CRC_CTRL). Ширина входных данных (DLSize, регистр CRC_CTRL) игнорируется и устанавливается в загрузку 32 бит (DLSize=10). При прямой последовательности входных данных (DinRev=0, регистр CRC_CTRL) перед подсчетом CRC данные будут сдвигаться к старшему биту на (32 – SerDinN) бит, и на подсчет CRC данные будут подаваться, начиная со старшего бита. При реверсивной последовательности входных данных (DinRev=1, регистр CRC_CTRL) перед подсчетом CRC данные уже выровнены к младшему биту и на подсчет CRC данные будут подаваться, начиная с младшего бита.

При подсчете CRC *параллельным способом* (FlowEn=0, SerALG=0, регистр CRC_CTRL, по 4 бита за один PCLK) за одну транзакцию можно загрузить 8, 16 и 32 бита (указать данную ширину входных данных в поле DLSize, регистр CRC_CTRL), далее при записи в FIFO произвести выравнивание данных относительно младшего бита. В поле Compl (регистр CRC_STAT) отслеживается завершение подсчета CRC.

При подсчете CRC *в потоковом режиме* (FlowEn=1, регистр CRC_CTRL) вначале выставляется общее количество бит, которое необходимо подсчитать в потоке (регистр CRC_CNTFLOW), далее при записи в FIFO произвести выравнивание данных относительно младшего бита. По завершении подсчета заданного потока данных вырабатывается прерывание и отображается статус бит завершения подсчета (FLEnd=1, регистр CRC_STAT, записью единицы в данное поле производится сброс прерывания), маскирование прерывания производится в поле Msklrq (регистр CRC_CTRL).

6.20.6 Примеры установки регистров для наиболее популярных протоколов

Таблица 152 – Ниже описаны полиномы CRC наиболее популярных протоколов

Название	Использование	Значение полинома (норм./реверс)	Значение регистров при нормальном полиноме, (hex)		
			CRC_POLH	CRC_POLL	CRC_CTRL PolSize
CRC-4-ITU	$x^4 + x + 1$ (ITU G.704)	0x03 / 0xC	0000_0000	0000_0003	0x03
CRC-5-EPC	$x^5 + x^3 + 1$ (Gen 2 RFID)	0x09 / 0x12	0000_0000	0000_0009	0x04
CRC-5-ITU	$x^5 + x^4 + x^2 + 1$ (ITU G.704)	0x15 / 0x15	0000_0000	0000_0015	0x04
CRC-5-USB	$x^5 + x^2 + 1$ (USB token packets)	0x05 / 0x14	0000_0000	0000_0005	0x04
CRC-6-ITU	$x^6 + x + 1$ (ITU G.704)	0x03 / 0x30	0000_0000	0000_0003	0x05
CRC-7	$x^7 + x^3 + 1$ (системы телекоммуникации, ITU-TG.707, ITU-TG.832, MMC, SD)	0x09 / 0x48	0000_0000	0000_0009	0x06
CRC-8-CCITT	$x^8 + x^2 + x + 1$ (ATMHEC, ISDNHeader Error Control and Cell Delineation ITU-T I.432.1 (02/99))	0x07 / 0xE0	0000_0000	0000_0007	0x07
CRC-8-Dallas/Maxim	$x^8 + x^5 + x^4 + 1$ (1-Wirebus)	0x31 / 0x8C	0000_0000	0000_0031	0x07
CRC-8	$x^8 + x^7 + x^4 + x^2 + 1$ (ETSI EN 302 307, 5.1.4)	0xD5 / 0xAB	0000_0000	0000_00D5	0x07
CRC-8-SAE J1850	$x^8 + x^4 + x^3 + x^2 + 1$ (AES3)	0x1D / 0xB8	0000_0000	0000_001D	0x07
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x + 1$ (ATM; I.610)	0x233 / 0x331	0000_0000	0000_0233	0x09
CRC-11	$x^{11} + x^9 + x^8 + x^7 + x^2 + 1$ (FlexRay)	0x385 / 0x50E	0000_0000	0000_0385	0x0A
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$ (системы телекоммуникации)	0x80F / 0xF01	0000_0000	0000_080F	0x0B
CRC-15-CAN	$x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$	0x4599 / 0x4CD1	0000_0000	0000_4599	0x0E
CRC-16-IBM	$x^{16} + x^{15} + x^2 + 1$ (Bisync, Modbus, USB, ANSI X3.28 и др.; также известен как CRC-16 и CRC-16-ANSI)	0x8005 / 0xA001	0000_0000	0000_8005	0x0F
CRC-16-USB3	$x^{16} + x^{12} + x^3 + x + 1$ (USB3)	0x100B	0000_0000	0000_100B	
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$ (X.25, HDLC, XMODEM, Bluetooth, SD и др.; также известен как CRC-CCITT)	0x1021 / 0x8408	0000_0000	0000_1021	0x0F
CRC-16-T10-DIF	$x^{16} + x^{15} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (SCSI DIF)	0x8BB7 / 0xEDD1	0000_0000	0000_8BB7	0x0F
CRC-16-DNP	$x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$ (DNP, IEC 870, M-Bus)	0x3D65 / 0xA6BC	0000_0000	0000_3D65	0x0F

Название	Использование	Значение полинома (норм./реверс)	Значение регистров при нормальном полиноме, (hex)		
			CRC_POLH	CRC_POLL	CRC_CTRL PolSize
CRC-24	$x^{24} + x^{22} + x^{20} + x^{19} + x^{18} + x^{16} + x^{14} + x^{13} + x^{11} + x^{10} + x^8 + x^7 + x^6 + x^3 + x + 1$ (FlexRay)	0x5D6DCB / 0xD3B6BA	0000_0000	005D_6DCB	0x17
CRC-24-Radix-64	$x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$ (OpenPGP)	0x864CFB / 0xDF3261	0000_0000	0086_4CFB	0x17
CRC-30	$x^{30} + x^{29} + x^{21} + x^{20} + x^{15} + x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^6 + x^2 + x + 1$ (CDMA)	0x2030B9C7 / 0x38E74301	0000_0000	2030_B9C7	0x1D
CRC-32-IEEE 802.3	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (HDLC, ANSI X3.66, ITU-T V.42, Ethernet, Serial ATA, MPEG-2, PKZIP, Gzip, Bzip2, PNG, и др.)	0x04C11DB7 / 0xEDB88320	0000_0000	04C1_1DB7	0x1F
CRC-32C (Castagnoli)	$x^{32} + x^{28} + x^{27} + x^{26} + x^{25} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{14} + x^{13} + x^{11} + x^{10} + x^9 + x^8 + x^6 + 1$ (iSCSI, SCTP, G.hn payload, SSE4.2, Btrfs, ext4)	0x1EDC6F41 / 0x82F63B78	0000_0000	1EDC_6F41	0x1F
CRC-32K (Koopman)	$x^{32} + x^{30} + x^{29} + x^{28} + x^{26} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{11} + x^{10} + x^7 + x^6 + x^4 + x^2 + x + 1$	0x741B8CD7 / 0xEB31D82E	0000_0000	741B_8CD7	0x1F
CRC-32Q	$x^{32} + x^{31} + x^{24} + x^{22} + x^{16} + x^{14} + x^8 + x^7 + x^5 + x^3 + x + 1$ (aviation; AIXM)	0x814141AB / 0xD5828281	0000_0000	8141_41AB	0x1F
CRC-40-GSM	(GSM control channel)	0x0004820009 / 0x9000412000	0000_0000	0482_0009	0x27
CRC-64-ISO	$x^{64} + x^4 + x^3 + x + 1$ (HDLC-ISO3309, Swiss-Prot/TrEMBL; considered weak for hashing)	0x0000000000000001B / 0xD800000000000000	0000_0000	0000_0001	0x3F
CRC-64-ECMA-182	$x^{64} + x^{62} + x^{57} + x^{55} + x^{53} + x^{52} + x^{47} + x^{46} + x^{45} + x^{40} + x^{39} + x^{38} + x^{37} + x^{35} + x^{33} + x^{32} + x^{31} + x^{29} + x^{27} + x^{24} + x^{23} + x^{22} + x^{21} + x^{19} + x^{17} + x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^4 + x + 1$ (ECMA-182, XZ Utils)	0x42F0E1EBA9EA3693 / 0xC96C5795D7870F42	42F0_E1EB	A9EA_3693	0x3F

6.20.6.1 CRC-3/ROHC

Width=3, Poly=0x3, Init=0x7, RefIn=true, RefOut=true, XorOut=0x0, Check=0x6.

Любые полиномы степенью менее 4 в модуле можно подсчитать последовательным способом.

Проверочное слово ASCII=123456789 или HEX=31 32 33 34 35 36 37 38 39.

	CRC-3/ROHC
CRC_CTRL	0x0009_C10D
MskIrq	0
FlowEn	0
SerALG	1
SerDinN	0x07
SelReq	0
PolSize	0x02
DLSize	0
DmaEn	0
CrcRev	1
DinRev	1
DinRevB	0
CrcEn	1
CRC_VALL	0x7
CRC_VALH (нач. значение)	0x0
CRC_POLL	0x00000003
CRC_POLH	0x0
CRC_FINXORL	0x0
CRC_FINXORH	0x00000000
CRC_CNTFLOW	0
CRC_DIN	0x3433_3231 0x3837_3635 0x0000_0039
CRC_VALL	0x3C55A1BC
CRC_VALH	0xD420EFBC
Результат CRC	b110

6.20.6.2 Примеры расчета CRC-5/USB3.0, CRC-16/USB3.0, CRC-32/USB3.0

USB3.0 CRC описано в UniversalSerialBus 3.0 Revision 1.0, стр. с 182 (7-4; 7.2.1.1.2) по 187 (7-9; 7.2.1.2.2).

Возьмем данные из примера (стр. 530, D-1):

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				
Device Address 000_0100b										Routing String 0000_0000_0000_0010_0001b																Type 01000b									
Data Length 0000_0000_0000_1001b														S 0	Rsvd 000b			Ept Num 0001b			D 0	EOB 1		R 0	Seq Num 00010b										
Rsvd 0000b			PP 1		Reserved 000_0000_0000_0000_0000_0000b																														
Link Control Word 01011b				0		0		010b			000b			001b			CRC-16 00011000b								11101101b										
Byte 3 0111_0110b						Byte 2 0101_0100b						Byte 1 0011_0010b						Byte 0 0001_0000b																	
Byte 7 1111_1110b						Byte 6 1101_1100b						Byte 5 1011_1010b						Byte 4 1001_1000b																	
0110_0010b						CRC32 1111_0011b						1000_1011b						Byte 8 0001_0000b																	
																												CRC32 0000_0111b							

U-165

	USB3.0_5	USB3.0_16	USB3.0_32
CRC_CTRL	0x000A_824D	0x0000_07DD	0x0030_0FDD
MskIrq	0	0	1
FlowEn	0	0	1
SerALG	1	0	0
SerDinN	0x0a	0	0
SelReq	0	0	0
PolSize	0x04	0x0F	0x1F
DLSIZE	10	10	10
DmaEn	0	1	1
CrcRev	1	1	1
DinRev	1	1	1
DinRevB	0	0	0
CrcEn	1	1	1
CRC_VALL	0x0000_001F	0x0000_FFFF	0xFFFF_FFFF
CRC_VALH (нач. значение)	0x0000_0000	0x0000_0000	0x0000_0000
CRC_POLL	0x0000_0005	0x0000_100B	0x04C1_1DB7
CRC_POLH	0x0000_0000	0x0000_0000	0x0000_0000
CRC_FINXORL	0x0000_0000	0x0000_0000	0x0000_0000
CRC_FINXORH	0xF800_0000	0xFFFF_0000	0xFFFF_FFFF
CRC_CNTFLOW	11	96	72
CRC_DIN	0x0000_0081	0x0800_0428 0x0009_0142 0x0800_0000	0x7654_3210 0xFEDC_BA98 0x0000_0010
CRC_VALL	0x0000_0000	0xAE88_7002	0x8D67_36A0
CRC_VALH	0x5CAF_0000	0x18ED_AA67	0x0762_F38B
Результат CRC	01011	0x18ED	0x0762_F38B

В примерах разобраны все три режима работы модуля:

- USB3.0_5 – последовательный подсчет;
- USB3.0_16 – параллельный подсчет;
- USB3.0_32 – потоковый подсчет.

6.20.7 Описание регистров

Базовый адрес	Название	Описание
0x400A_F000	CRC	Контроллер CRC
Смещение		
0x000	CRC_KEY	Ключ разрешения записи в конфигурационные регистры
0x004	CRC_CTRL	Общее управление для контроллера CRC
0x008	CRC_STAT	Статус CRC блока, регистр недоступен до записи ключа
0x00C	CRC_DIN	Регистр FIFO входных данных, регистр недоступен до записи ключа
0x010	CRC_VALL	Регистр подсчитанного CRC младшая часть
0x014	CRC_VALH	Регистр подсчитанного CRC старшая часть
0x018	CRC_POLL	Полином для расчета CRC младшая часть
0x01C	CRC_POLH	Полином для расчета CRC старшая часть
0x020	CRC_FINXORL	Регистр дополнительного XOR младшая часть
0x024	CRC_FINXORH	Регистр дополнительного XOR старшая часть
0x028	CRC_CNTFLOW	Декрементирующий счетчик потока

6.20.7.1 Регистр CRC_KEY

Base ADDR=	0x400A_F000	Offset=	0x000												
REG Name:	CRC_KEY														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
RW-0															

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

Бит	Имя	Значение	Описание
31:0	Key	0	При значении CRC_KEY = 0x8555AAA1, открыт доступ по записи в конфигурационные регистры

6.20.7.2 Регистр CRC_CTRL

Base ADDR=	0x400A_F000	Offset=	0x004												
REG Name:	CRC_CTRL														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserve										CrcRevB	MskIrq	FlowEn	SerALG	SerDinN	
		RW-0								RW-0	RW-0	RW-0	RW-0	RW-0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SerDinN		SelReq	PolSize				DLSize		DmaEn	CrcRev	DinRev	DinRevB	CrcEn		
RW-0		RW-0	RW-001111				RW-10		RW-0	RW-0	RW-0	RW-0	RW-0		

Бит	Имя	Значение по сбросу	Описание
22	CrcRevB	0	Побайтовая реверсия вычисленного CRC перед выходом финального XOR Слово CRC_VALH[31:0] преобразуется в CRC_VALH{[24:31],[16:23],[8:15],[0:7]} и Слово CRC_VALL[31:0] преобразуется в CRC_VALL{[24:31],[16:23],[8:15],[0:7]}
21	MskIrq	0	Маска включения прерывания: 0 – запрет (по умолчанию); 1 – разрешение прерывания

Бит	Имя	Значение по сбросу	Описание
20	FlowEn	0	Разрешение работы с потоком данных: 0 – отключено (по умолчанию); 1 – включено отслеживание потока (в конце потока вырабатывается прерывание)
19	SerALG	0	Включение подсчета CRC последовательным способом: 0 – параллельный подсчет (по умолчанию); 1 – последовательный подсчет
18..14	SerDinN	0	Число обрабатываемых входных данных последовательным способом 0x00 – 1 (по умолчанию) ... 0x1F – 32
13	SelReq	0	Выбор разрешения установки DMA_REQ в «1»: 0 – при пустом FIFO (по умолчанию); 1 – при не полном FIFO;
12..7	PolSize[5:0]	001111	Размер полинома CRC (=N-1, где N – длина полинома, от 4 до 64): 0x03 – длина полинома = 4; ... 0x3F – длина полинома = 64
6..5	DLSIZE[1:0]	10	Размер загружаемых данных: 00 – байт (8 бит, при этом загружаемый байт записывается в CRC_DIN[7:0]) 01 – полуслово (16 бит, при этом загружаемое полуслово записывается в CRC_DIN[15:0]) 10 – слово (32 бита, при этом загружаемое слово записывается в CRC_DIN[31:0]) 11 – зарезервировано
4	DmaEn	0	Разрешение формирования запроса для DMA 0 – запрос не формируется 1 – запрос формируется
3	CrcRev	0	Реверс вычисленного CRC перед выходом финального XOR 0 – преобразования содежимого не происходит 1 – Слово CRC_VALH[31:0] преобразуется в CRC_VALL[0:31] и слово CRC_VALL[31:0] преобразуется в CRC_VALH[0:31]
2	DinRev	0	Порядок следования байтов в данных 0 – начиная со старшего разряда (прямой) 1 – начиная с младшего разряда (реверсивный)
1	DinRevB	0	Реверсия входных данных относительно байта Байт CRC_DIN [31:24 23:16 15:8 7:0] преобразуется в CRC_DIN [24:31 16:23 8:15 0:7]
0	CrcEn	0	Разрешение работы блока 0 – блок выключен 1 – блок включен

6.20.7.3 Регистр CRC_STAT

Base ADDR=	0x400A_F000					Offset=	0x008								
REG Name:	CRC_STAT														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserve															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserve											FLEnd	FOver	FEmpty	FFull	Compl
											R-0	R-0	R-1	R-0	R-1

Бит	Имя	Значение	Описание
4	FLEnd	0	Поток данных: 0 – неактивен; 1 – обработан весь поток данных. По окончании обработки потока данных вырабатывается прерывание, для сброса флага в данное поле необходимо записать единицу
3	FOver	0	Переполнение FIFO: 0 – корректная работа; 1 – была запись в полное FIFO, что привело к потере данных. Для сброса флага в данное поле необходимо записать единицу
2	FEmpty	1	FIFO пусто: 0 – FIFO имеет по крайней мере одну заполненную ячейку 1 – FIFO пусто
1	FFull	0	FIFO заполнено 0 – FIFO имеет по крайней мере одну свободную ячейку; 1 – FIFO не имеет свободных ячеек
0	Compl	1	Завершение расчета CRC: 0 – расчет идет; 1 – расчет завершен и FIFO пусто или блок отключен

6.20.7.4 Регистр CRC_DIN

Base ADDR=	0x400A_F000					Offset=	0x00C								
REG Name:	CRC_DIN														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIN[31:16]															
W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN[15:0]															
W-0															

Бит	Имя	Значение	Описание
31:0	DIN	0	Регистр для записи нового отчета в FIFO

6.20.7.5 Регистр CRC_VALL

Base ADDR=	0x400A_F000					Offset=	0x010									
REG Name:	CRC_VALL															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CRCOUTL[31:16]																
RW-0																

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

Бит	Имя	Значение	Описание
31:0	CRCOUTL	0	Младшая часть рассчитанного значения /начального значения. Начальное значение записывать только, когда блок отключен или когда закончен расчет CRC (COMPL=1)

6.20.7.6 Регистр CRC_VALH

Base ADDR=	0x400A_F000					Offset=	0x014									
REG Name:	CRC_VALH															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CRCOUTH[31:16]																
RW-0																

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

Бит	Имя	Значение	Описание
31:0	CRCOUTH	0	Старшая часть рассчитанного значения /начального значения. Начальное значение записывать только, когда блок отключен или когда закончен расчет CRC (COMPL=1)

6.20.7.7 Регистр CRC_POLL

Base ADDR=	0x400A_F000					Offset=	0x018									
REG Name:	CRC_POLL															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CRC_POLL[31:16]																
RW-0x0000_0001																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CRC_POLL[15:0]																
RW-0x0000_0001																

Бит	Имя	Значение	Описание
31:0	CRC_POLL	0	Младшая часть полинома для расчета CRC

6.20.7.8 Регистр CRC_POLH

Base ADDR=	0x400A_F000					Offset=	0x01C									
REG Name:	CRC_POLH															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CRC_POLL[31:16]																
RW-0x0000_0000																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CRC_POLL[15:0]																
RW-0x0000_0000																

Бит	Имя	Значение	Описание
31:0	CRC_POLH	0	Старшая часть полинома для расчета CRC

6.20.7.9 Регистр CRC_FINXORL

Base ADDR=	0x400A_F000					Offset=	0x020									
REG Name:	CRC_FINXORL															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CRC_FINXORL[31:16]																
RW-0																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CRC_FINXORL[15:0]																
RW-0																

Бит	Имя	Значение	Описание
31:0	CRC_FINXORL	0	Регистр дополнительного XOR, младшая часть.

6.20.7.10 Регистр CRC_FINXORH

Base ADDR=	0x400A_F000					Offset=	0x024									
REG Name:	CRC_FINXORH															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CRC_FINXORH[31:16]																
RW-0																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CRC_FINXORH[15:0]																
RW-0																

Бит	Имя	Значение	Описание
31:0	CRC_FINXORH	0	Регистр дополнительного XOR, старшая часть

Base	0x1000_5000	Offset	0x020							
------	-------------	--------	-------	--	--	--	--	--	--	--

105-16

14 500

– последовательный.

Слово 0	31:24	23:16	15: 8	7: 0
Слово 1	31:24	23:16	15: 8	7: 0
...	...			
Слово N-1	31:24	23:16	15: 8	7: 0
Слово N	L: 0			

Рисунок 136 – Порядок записи данных в FIFO (ECC_DIN)

При *последовательном* алгоритме подсчета ($ECC_CTRL.FlowEn = 0$ и $ECC_CTRL.SerALG = 1$) скорость подсчета составляет 1 бит/PCLK. При подсчете за одну транзакцию можно загрузить и подсчитать от 1 до 32 бит (указать данное количество бит в поле $ECC_CTRL.SerDinN$), ширина входных данных ($ECC_CTRL.DLSize$) игнорируется и устанавливается в загрузку 32 бит. При прямой последовательности входных данных ($ECC_CTRL.DinRev = 0$) перед подсчетом ECC данные будут сдвигаться к старшему биту на значение равное $(32 - SerDinN)$ бит, и на подсчет ECC данные будут подаваться, начиная со старшего бита. При реверсивной последовательности входных данных ($ECC_CTRL.DinRev = 1$) перед подсчетом ECC данные уже выровнены к младшему биту, и на подсчет ECC данные будут подаваться, начиная с младшего бита.

При *параллельном* алгоритме ($ECC_CTRL.FlowEn = 0$, $ECC_CTRL.SerALG = 0$) скорость подсчета составляет 4 бит/PCLK. При подсчете за одну транзакцию можно загрузить и подсчитать 8, 16, 32 бит ($ECC_CTRL.DLSize$), далее при записи в FIFO произвести выравнивание данных относительно младшего бита. В поле $ECC_STAT.Cmpl$ отслеживается завершение подсчета ECC.

Алгоритм подсчета *потока* данных ($ECC_CTRL.FlowEn = 1$) состоит из параллельного и последовательного подсчета ECC. Если общее количество бит потока не кратно четырем, последние биты (1, 2, 3 бита) считаются последовательным способом, а все предыдущие – параллельным. В начале работы необходимо выставить общее количество бит в потоке (регистр $ECC_CNTFLOW$), далее при записи в FIFO произвести выравнивание данных относительно младшего бита. По завершению подсчета заданного потока данных вырабатывается прерывание и отображается статус бит завершения подсчета ($ECC_STAT.FLEnd = 1$, записью единицы в данное поле производится сброс прерывания), маскирование прерывания производится в поле $ECC_CTRL.Msklrq$.

6.21.2 Описание регистров управления блока ECC

Базовый адрес	Название	Описание
0x400B_0000	ECC	Контроллер ECC
Смещение		
0x000	KEY	Ключ разрешения записи в конфигурационные регистры
0x004	CTRL	Общее управление для контроллера ECC
0x008	STAT	Статус ECC блока
0x00C	DATIL	Регистр записываемых данных младшего слова
0x010	DATIH	Регистр записываемых данных старшего слова
0x014	ECCI	Регистр записываемого ECC
0x018	DLER	Подсветка ошибки данных младшего слова
0x01C	DHER	Подсветка ошибки данных старшего слова
0x020	ECER	Подсветка ошибки вычисленного и записываемого ECC
0x024	DATOL	Регистр исправленных данных младшего слова
0x028	DATOH	Регистр исправленных данных старшего слова
0x02C	ECCO	Регистр вычисленного или исправленного ECC
0x040	H000	Значение H-матрицы с $h_{0,0}$ по $h_{0,31}$
0x044	H032	Значение H-матрицы с $h_{0,32}$ по $h_{0,63}$
0x048	H100	Значение H-матрицы с $h_{1,0}$ по $h_{1,31}$
0x04C	H132	Значение H-матрицы с $h_{1,32}$ по $h_{1,63}$
0x050	H200	Значение H-матрицы с $h_{2,0}$ по $h_{2,31}$
0x054	H232	Значение H-матрицы с $h_{2,32}$ по $h_{2,63}$
0x058	H300	Значение H-матрицы с $h_{3,0}$ по $h_{3,31}$
0x05C	H332	Значение H-матрицы с $h_{3,32}$ по $h_{3,63}$
0x060	H400	Значение H-матрицы с $h_{4,0}$ по $h_{4,31}$
0x064	H432	Значение H-матрицы с $h_{4,32}$ по $h_{4,63}$
0x068	H500	Значение H-матрицы с $h_{5,0}$ по $h_{5,31}$
0x06C	H532	Значение H-матрицы с $h_{5,32}$ по $h_{5,63}$
0x070	H600	Значение H-матрицы с $h_{6,0}$ по $h_{6,31}$
0x074	H632	Значение H-матрицы с $h_{6,32}$ по $h_{6,63}$
0x078	H700	Значение H-матрицы с $h_{7,0}$ по $h_{7,31}$
0x07C	H732	Значение H-матрицы с $h_{7,32}$ по $h_{7,63}$

6.21.2.1 Регистр KEY

Base ADDR=		0x400B_0000					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]															
Бит		Имя			Значение		Описание								
31:0		Key			0		При значении KEY = 0x8AAA_5551, открыт доступ по записи в конфигурационные регистры								

6.21.2.2 Регистр CTRL

Base ADDR=	0x400B_0000					Offset=	0x004									
REG Name:	ECC_CTRL															
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	
-	-	MskIrq	-	-	DatSize[5:0]					EccSize[3:0]			EncDec			
	RW-0	RW-0			RW-0x3F					RW-0x7			RW-0			

Бит	Имя	Значение	Описание
31...15	-		Зарезервировано
14	HamEPB	0	Включение дополнительного бита четности в подсчете ECC методом Хэмминга: 0 – запрет (по умолчанию); 1 – разрешение. Заполнение H-матрицы дополнительного бита ECC должно быть заполнено «1» самостоятельно
13	MskIrq	0	Маска включения прерывания: 0 – запрет (по умолчанию); 1 – разрешение прерывания
12...11	-		Зарезервировано
10...5	DatSize[5:0]	0x3F	Размер входных данных: 0x00 – 1; ... 0x3F – 64
4...1	EccSize[3:0]	0x7	Размер входных проверочных бит: 0x0 – 1; ... 0x7 – 8; 0x8 – 9; – резерв ... 0xF – 16; – резерв
0	EncDec	0	Режим работы модуля: 0 – включен Encoder; 1 – включен Decoder

6.21.2.3 Регистр STAT

Base ADDR=	0x400B_0000					Offset=	0x008									
REG Name:	ECC_STAT															
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	
-													ERR1	ERR2	Compl	
													R-0	R-0	R-1	

Бит	Имя	Значение	Описание
31...3	-		Зарезервировано
2	ERR1	0	Одиночная ошибка ECC данных при декодировании
1	ERR2	0	Двойная ошибка ECC данных при декодировании
0	Compl	1	Завершение расчета ECC, чтение регистра ECC_STAT сбрасывает в ноль. 0 – расчет идет или блок отключен (по умолчанию); 1 – расчет завершен (условие прерывания)

6.21.2.4 Регистр DATIL

Base ADDR=	0x400B_0000				Offset=	0x00C									
REG Name:															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATIL[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATIL[15:0]															
Бит	Имя				Значение				Описание						
31:0	DATIL				0				Регистр записываемых данных младшего слова						

6.21.2.5 Регистр DATIH

Base ADDR=	0x400B_0000							Offset=			0x010																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
------------	-------------	--	--	--	--	--	--	---------	--	--	-------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

6.21.2.6 Регистр ECC_ECCI

Base ADDR=	0x400B_0000								Offset=	0x014																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
------------	-------------	--	--	--	--	--	--	--	---------	-------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

6.21.2.7 Регистр ECC_DLER

Base ADDR=				0x400B_0000				Offset=				0x018																			
REG Name:				ECC_DLER																											
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
DLER[31:0]																															
R-0																															

Бит	Имя	Значение	Описание
31:0	DLER	0	Подсветка ошибки данных младшего слова

[illegible][illegible]

Base ADDR=	0x400B_0000	Offset=	0x024																												
REG Name:	ECC_DATOL																														
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
DATOL[31:0]																															
R-0																															

[illegible]

[illegible]

6.21.2.13 Регистры ECC_H000 – ECC_H732

Бит	Имя	Значение	Описание
31:0	Hmx	0	Значения H-матрицы

Diagram illustrating the layout of ECC_H000, ECC_H032, ECC_H700, and ECC_H732. The layout shows memory addresses and corresponding ECC values.

ECC_H000: 0 1 ... 30 31 32 33 ... 62 63
 $h_{0.0}$ $h_{0.1}$... $h_{0.30}$ $h_{0.31}$ $h_{0.32}$ $h_{0.33}$... $h_{0.62}$ $h_{0.63}$

ECC_H032: $h_{0.32}$ $h_{0.33}$... $h_{0.62}$ $h_{0.63}$

ECC_H700: $h_{7.0}$ $h_{7.1}$... $h_{7.30}$ $h_{7.31}$ $h_{7.32}$ $h_{7.33}$... $h_{7.62}$ $h_{7.63}$

ECC_H732: $h_{7.32}$ $h_{7.33}$... $h_{7.62}$ $h_{7.63}$

© АО «ПКК Миландр»

7 Контроллер прерываний NVIC

В разделе описан векторный контроллер прерываний с возможностью вложения (NVIC – Nested Vectored Interrupt Controller) и используемые им регистры.

Контроллер обеспечивает поддержку:

- программное задание уровня приоритета в диапазоне от 0 до 192 с шагом 64 независимо каждому прерыванию. Более высокое значение соответствует меньшему приоритету, таким образом, уровень 0 отвечает наивысшему приоритету прерывания;
- срабатывание сигнала прерывания по импульсу и по уровню;
- передача управления из одного обработчика исключения на другой без восстановления контекста.

Процессор автоматически сохраняет в стеке свое состояние (контекст) по входу в обработчик прерывания и восстанавливает его по завершению обработчика, без необходимости непосредственного программирования этих операций. Это обеспечивает обработку исключительных ситуаций с малой задержкой.

Назначение регистров контроллера прерываний представлено в таблице 153.

Таблица 153 – Обобщенная информация о регистрах контроллера NVIC

Адрес	Имя	Тип	Значение после сброса	Описание
0xE000E100	ISER	RW	0x00000000	Регистр разрешения прерываний
0xE000E180	ICER	RW	0x00000000	Регистр запрета прерывания
0xE000E200	ISPR	RW	0x00000000	Регистр перевода прерывания в состояние ожидания обслуживания
0xE000E280	ICPR	RW	0x00000000	Регистр сброса состояния ожидания обслуживания
0xE000E400 – 0xE000E41C	IPR0-7	RW	0x00000000	Регистр приоритета прерываний

7.1 Логика работы прерываний контроллера NVIC.

В данном разделе описывается функционирование контроллера NVIC при поступлении на его вход запросов прерываний IRQ от различных модулей периферии микроконтроллера.

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.

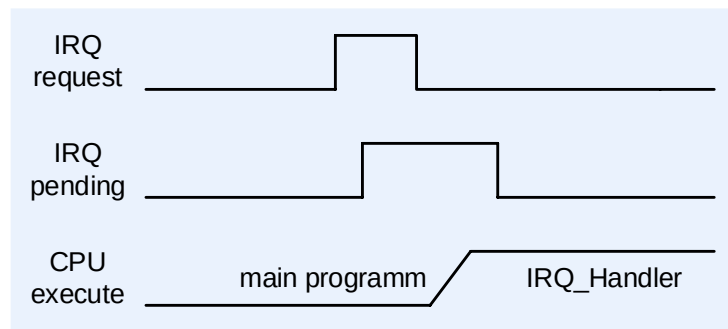


Рисунок 138 – Выставление отложенного запроса на прерывание и последующая его обработка

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 139.

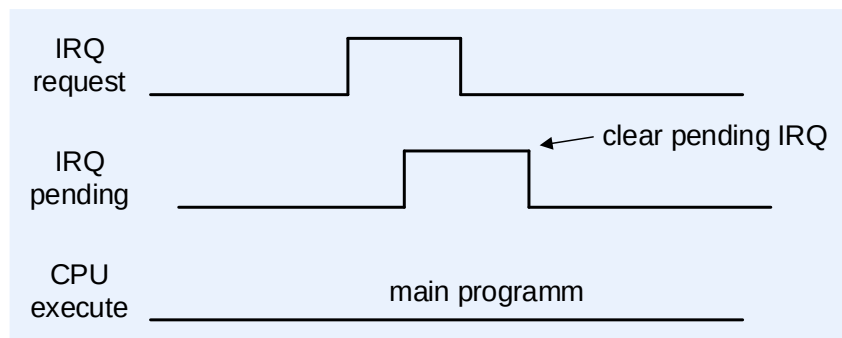


Рисунок 139 – Сброс признака отложенного прерывания, до обработки ядром

Если произойдет снятие запроса IRQ request от источника, «защелкивание» признака отложенного прерывания гарантирует отработку его ядром в соответствии с приоритетом. – рисунок 140. Сам IRQ pending признак снимается автоматически, когда прерывание становится активным, о чём сигнализирует признак IRQ active.

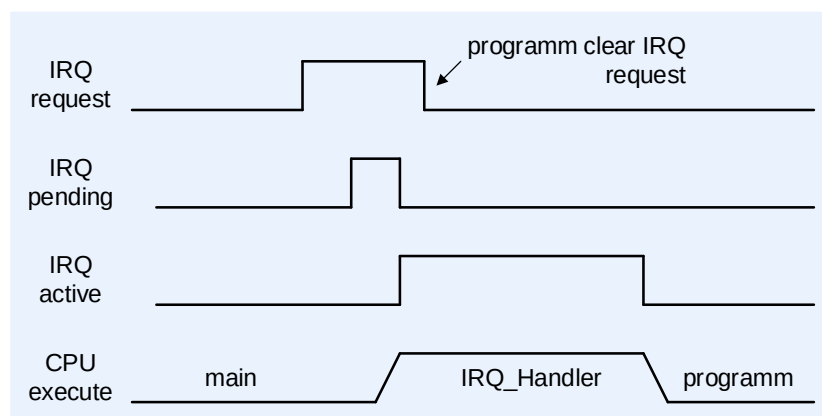


Рисунок 140 – Сброс признака отложенного прерывания, до обработки ядром

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 141.

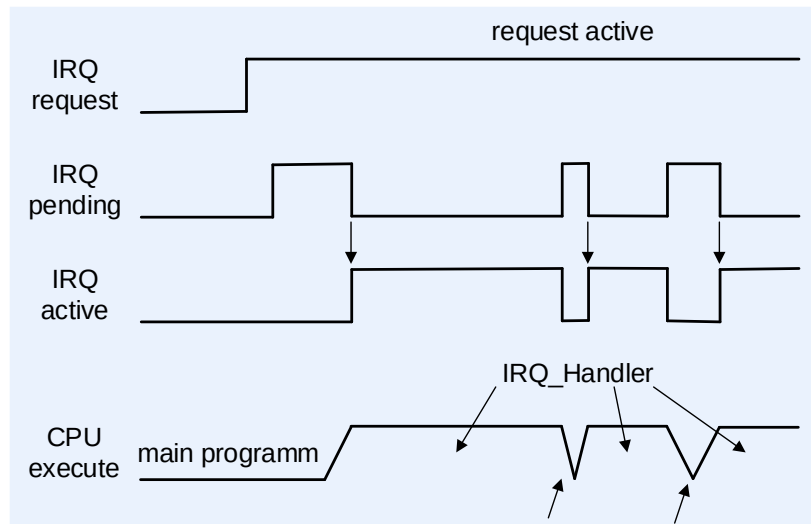


Рисунок 141 – Повторная обработка прерываний при удержании запроса от источника

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 142.

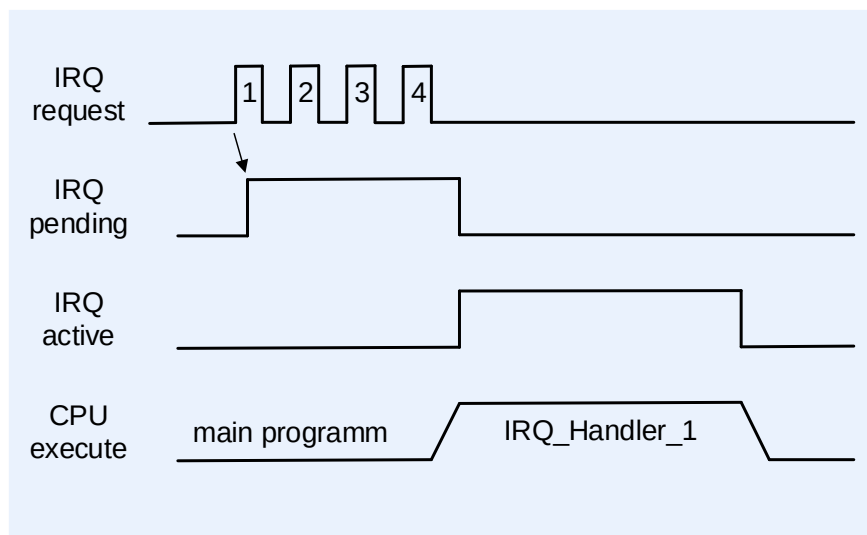


Рисунок 142 – Многократная установка снятие запроса IRQ request

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 143.

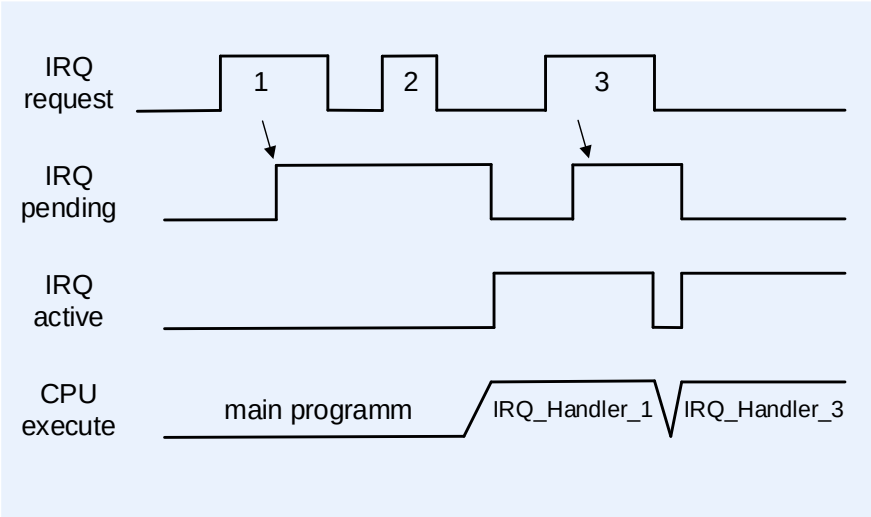


Рисунок 143 – Повторная установка запроса на прерывание в момент выполнения обработчика исключения

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.

7.1 Регистр разрешения прерываний

Регистр ISER предназначен для разрешения прерываний (запись) и определения, какие из прерываний разрешены (чтение). Более подробная информация представлена в таблице 153.

Распределение битов регистра представлено на рисунке 144.

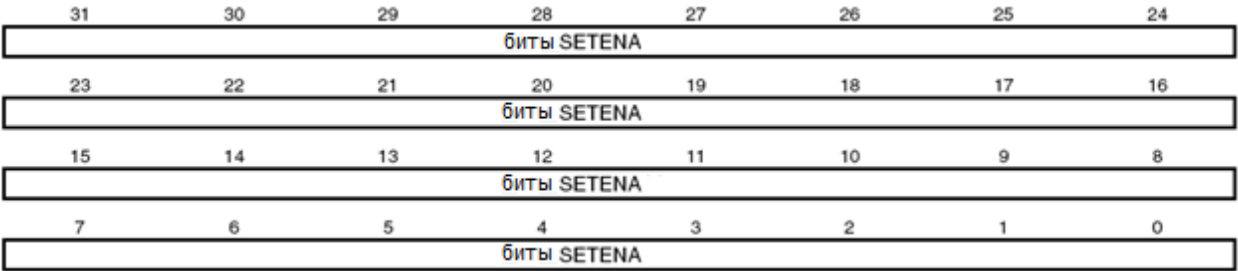


Рисунок 144 – Распределение битов регистра ISER

Таблица 154 – Регистр разрешения прерываний

Биты	Поле	Функция
31...0	SETENA	Биты разрешения прерывания. При записи: 1 – разрешение прерывания; 0 – не оказывает влияния. При чтении: 1 – прерывание разрешено; 0 – прерывание запрещено.

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.

7.2 Регистр запрета прерываний

Регистр ICER предназначен для запрета прерываний (запись) и определения, какие из прерываний разрешены (чтение). Более подробная информация представлена в таблице 153.

Распределение битов регистра представлено на рисунке 145.

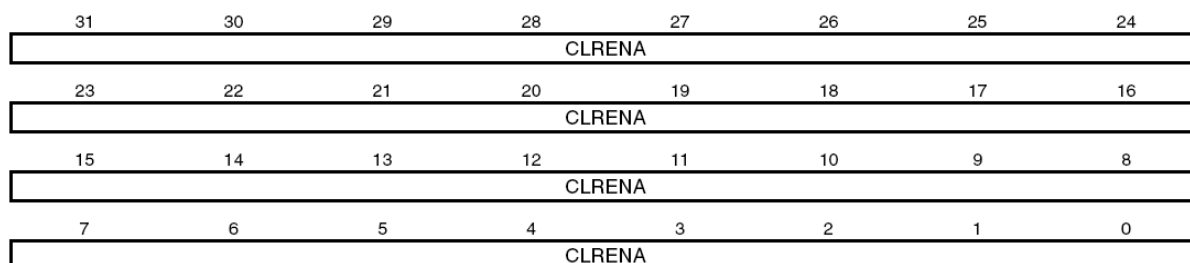


Рисунок 145 – Распределение битов регистра ICER

Таблица 155 – Регистр запрета прерываний

Биты	Поле	Функция
31...0	CLRENA	Биты запрещения прерывания. При записи: 1 – запрещает прерывание; 0 – не оказывает влияния. При чтении: 1 – прерывание разрешено; 0 – прерывание запрещено.

7.3 Регистр установки состояния ожидания для прерывания

Регистр ISPR предназначен для принудительного перевода прерываний в состояние ожидания обслуживания (запись) и определения, какие из прерываний находятся в этом состоянии (чтение).

Более подробная информация представлена в таблице 153.

Распределение битов регистра представлено на рисунке 146.

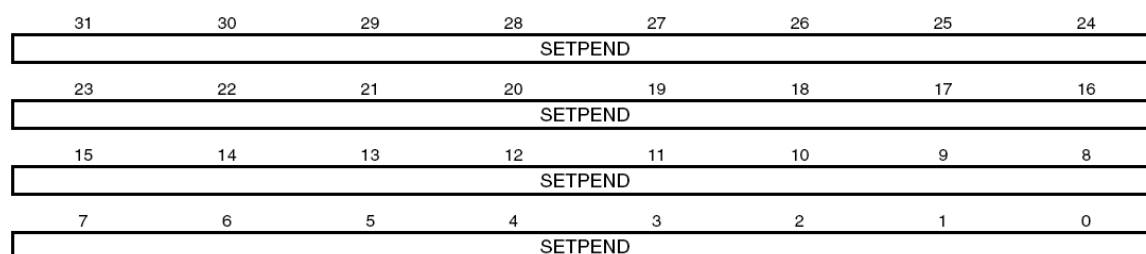


Рисунок 146 – Распределение битов регистра ISPR

Таблица 156 – Регистр установки состояния ожидания для прерывания

Биты	Поле	Функция
31...0	SETPEND	При записи: 1 – перевод прерывания в состояние ожидания; 0 – не оказывает влияния. При чтении: 1 – прерывание в состоянии ожидания обслуживания; 0 – прерывание не в состоянии ожидания обслуживания.

Запись 1 в бит регистра ISPR, соответствующий:

- прерыванию, уже ожидающему обслуживания – не влияет на работу системы;
- запрещенному прерыванию – переводит его в состояние ожидания.

7.4 Регистр сброса состояния ожидания для прерывания

Регистр ICPR предназначен для принудительного сброса состояния ожидания обслуживания прерывания (запись) и определения, какие из прерываний находятся в состоянии ожидания (чтение).

Более подробная информация представлена в таблице 153.

Распределение битов регистра представлено на рисунке 147.

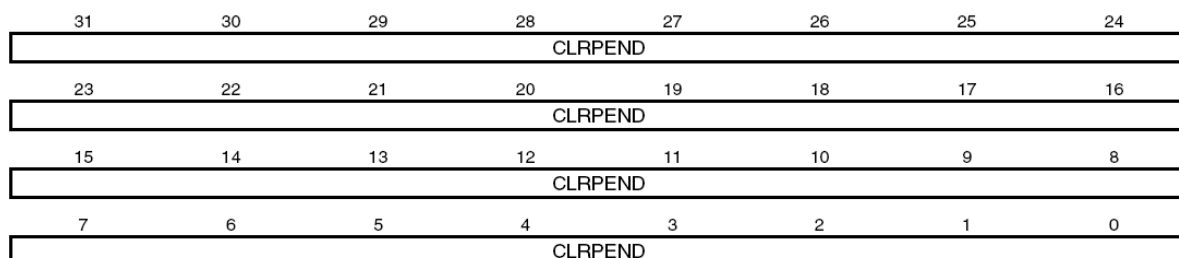


Рисунок 147 – Распределение битов регистра ICPR

Таблица 157 – Регистр сброса состояния ожидания обслуживания

Биты	Поле	Функция
31...0	CLRPEND	При записи: 1 – сбрасывает состояние ожидания обслуживания; 0 – не оказывает влияния. При чтении: 1 – прерывание в состоянии ожидания обслуживания; 0 – прерывание не в состоянии ожидания обслуживания.

Запись 1 в разряд регистра ICPR, соответствующего прерыванию в активном состоянии, не влияет на работу системы.

7.5 Регистры приоритета прерываний

Регистры IPR0-IPR7 представляют собой набор 8-битовых полей, каждое из которых соответствует одному прерыванию. Регистры доступны пословно. Обобщенная информация об их характеристиках представлена в таблице 153.

Каждый из регистров содержит четыре поля приоритета, которые отображаются на четыре элемента массива PRI[0] ... PRI[31] CMSIS, как показано 148.

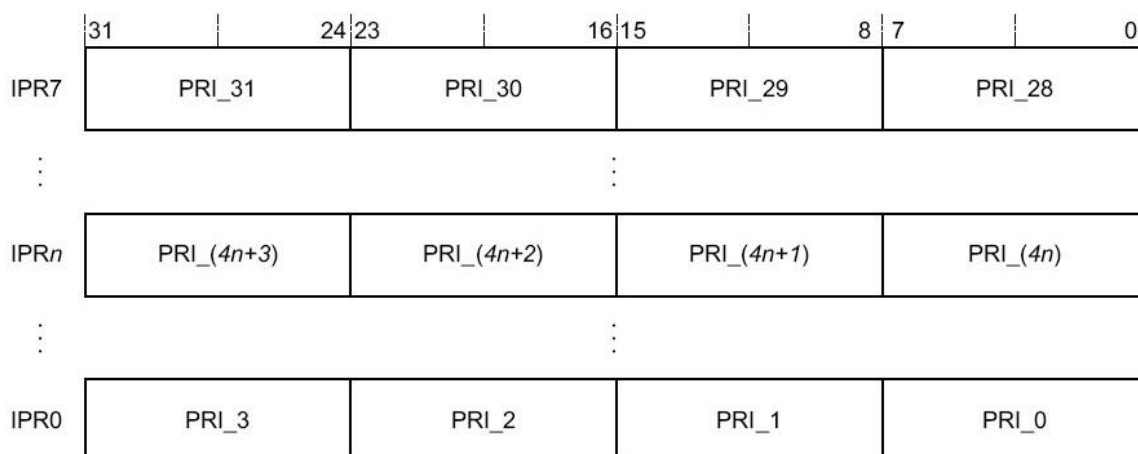


Рисунок 148 – Назначение бит

Каждое поле содержит значение приоритета в диапазоне от 0 до 192, причем меньшие значения соответствуют более высокому приоритету соответствующего прерывания. Процессор обеспечивает доступ только к битам [7:6] приоритета, биты [5:0] при чтении всегда равны нулю, а при записи игнорируются. Поэтому, например, запись 255 в регистр запишется как 192.

Для того, чтобы определить номер регистра IPR и смещение данных в регистре необходимо выполнить следующие операции:

- для заданного номера прерывания N номер M соответствующего регистра приоритета равен $M = N \text{ DIV } 4$;
- смещение данных в регистре в зависимости от значения $N \text{ MOD } 4$ равно:
 - 0 – биты регистра [7:0];
 - 1 – биты регистра [15:8];
 - 2 – биты регистра [23:16];
 - 3 – биты регистра [31:24].

7.6 Прерывания, срабатывающие по уровню сигнала

Процессор способен обрабатывать прерывания, сформированные по уровню сигнала.

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 «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.

7.7 Аппаратное и программное управление прерываниями

Процессор Cortex-M0 регистрирует все поступающие прерывания. Перевод прерывания, сформированного периферийным устройством, в состояние ожидания обслуживания осуществляется в одном из следующих случаев:

- контроллер прерываний NVIC обнаруживает, что сигнал запроса имеет высокий логический уровень, а прерывание не активно;
- контроллер прерываний NVIC обнаруживает передний фронт сигнала запроса прерывания;
- программное обеспечение осуществляет запись в соответствующий разряд регистра ISPR0 или соответствующего значения в регистр STIR.

Прерывание находится в состоянии ожидания до тех пор, пока не произойдет одно из следующих событий:

- процессор передаст управление процедуре обработки прерывания. В этом случае прерывание переходит в активное состояние, после чего по завершении обработки прерывания, срабатывающего по уровню, контроллер NVIC проверяет состояние сигнала запроса на прерывание. Если этот сигнал активен, прерывание вновь переводится в состояние ожидания обслуживания, что приводит к немедленной повторной передаче управления на обработчик. В противном случае прерывание переводится в неактивное состояние.
- если в период выполнения процедуры обработки прерывания, настроенного на срабатывание по фронту, не было зафиксировано импульсов на линии запроса, прерывание переводится в неактивное состояние.
- программное обеспечение осуществляет запись в соответствующий разряд регистра сброса состояния ожидания прерывания.

7.8 Рекомендации по работе с контроллером прерываний

Доступ к регистрам контроллера из программного обеспечения должен осуществляться по корректно выровненным адресам. Процессор не поддерживает возможность доступа к контроллеру по невыровненным адресам. Требования по выравниванию приведены в описании регистров.

Прерывание может быть переведено в состояние ожидания обслуживания даже в случае, если оно запрещено.

Программное разрешение или запрещение прерываний может осуществляться с помощью инструкций CPSIE I и CPSID I. В CMSIS предусмотрены следующие встроенные функции, генерирующие эти инструкции:

```
void __disable_irq(void) // Disable Interrupts
void __enable_irq(void) // Enable Interrupts
```

Кроме того, в CMSIS имеется ряд дополнительных функций, обеспечивающих управление контроллером прерываний NVIC (Таблица 158).

Т а б л и ц а 158 – Функции CMSIS для управления контроллером прерываний

Функция	Описание
void NVIC_EnableIRQ(IRQn_t IRQn)	Разрешить IRQn
void NVIC_DisableIRQ(IRQn_t IRQn)	Запретить IRQn
uint32_t NVIC_GetPendingIRQ (IRQn_t IRQn)	Вернуть истину, если прерывание IRQn ожидает обслуживания, ложь – в противном случае
void NVIC_SetPendingIRQ (IRQn_t IRQn)	Перевести IRQn в состояние ожидания обслуживания

Функция	Описание
void NVIC_ClearPendingIRQ (IRQn_t IRQn)	Сбросить состояние ожидания обслуживания для IRQn
void NVIC_SetPriority (IRQn_t IRQn, uint32_t priority)	Установить приоритет для IRQn
uint32_t NVIC_GetPriority (IRQn_t IRQn)	Считать приоритет IRQn
void NVIC_SystemReset (void)	Сбросить систему

Более подробная информация отражена в документации по CMSIS.

8 Прерывания и исключения

Состояние исключений:

Inactive – исключение не находится в стадии Active или Pending.

Pending – исключение находится в состоянии ожидания обработки процессором. Запрос прерывания от периферийных блоков или программы может изменить состояние соответствующего прерывания на состояние pending.

Active – исключение начало обрабатываться процессором, но еще не закончено. Обработчик исключения может быть прерван другим обработчиком исключения. В этом случае оба исключения находятся в состоянии Active.

Active и **Pending** – исключения начало обрабатываться процессором, но появилось новое исключение в состоянии pending от того же источника.

8.1 Типы исключений

Исключения бывают следующих типов:

- RESET;
- NON MASKABLE INTERRUPT (NMI);
- HARD FAULT;
- SVCALL;
- PendSV;
- SysTick;
- Прерывания (IRQ);
- Прерывания (IRQ).

8.1.1 RESET

RESET вызывается при включении питания и горячем сбросе. Модель исключений трактует RESET как специальная форма исключения. Когда выставляется RESET, работа процессора останавливается потенциально в любой точке инструкций. Когда RESET убирается, выполнение перезапускается с адреса, заданного в таблице векторов для сброса. Выполнение перезапускается в thread режиме.

8.1.2 NON MASKABLE INTERRUPT (NMI)

Не маскируемое прерывание (NMI) может быть вызвано периферией или установлено программой. Это самое высокоприоритетное исключение после сброса. Всегда разрешено и имеет фиксированный приоритет -2.

NMI не может быть:

- замаскировано или предотвращено от активации из другого исключения;
- прерывает любые исключения кроме RESET.

8.1.3 HARD FAULT

Hard Fault исключение происходит при ошибке во время обработки исключений или потому, что исключение не может быть обработано каким-либо другим механизмом. Hard fault имеет фиксированный приоритет -1, означающий, что он имеет больший приоритет чем любое из исключений с конфигурируемым приоритетом.

8.1.4 SVCALL

Исключение Supervisor Call (SVCALL) возникает при выполнении инструкции SVC. В приложениях с использованием Операционных Сред инструкция SVC может использоваться для доступа к функциям ОС и драйверам устройств.

8.1.5 PendSV

PendSV является прерыванием запросом сервисов системного уровня. В приложениях с использованием ОС PendSV используется для переключения контекстов, когда нет других активных исключений.

8.1.6 SysTick

Исключение SysTick является исключением, генерируемым системным таймером, когда он обнуляется. Программное обеспечение также может генерировать исключение SysTick. В приложениях с использованием ОС процессор может использовать это исключение для подсчета системных циклов

8.1.7 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 159 – Таблица различных типов исключений

Номер исключения	IRQ номер	Тип	Приоритет	Адрес вектора обработчика (смещение)	Активация
1	-	RESET	-3, наивысший	0x0000_0004	Асинхронный
2	-14	NMI	-2	0x0000_0008	Асинхронный
3	-13	Hard Fault	-1	0x0000_000C	-
4-10	-	Reserved	-	Зарезервировано	-
11	-5	SVCALL	Конфигурируемый	0x0000_002C	Синхронный
12-13	-	-	-	Зарезервировано	-
14	-2	PendSV	Конфигурируемый	0x0000_0038	Асинхронный
15	-1	SysTick	Конфигурируемый	0x0000_003C	Асинхронный
16 и выше	0 и выше	IRQ	Конфигурируемый	0x0000_0040 и выше	Асинхронный

Для асинхронных исключений, кроме RESET, процессор может выполнить другие инструкции между возникновением сигнала исключения и входом в обработчик.

Программа в Privileged режиме может запретить прерывания, имеющие конфигурируемый приоритет.

8.2 Обработчики исключений

Для обработки исключений используются:

Процедуры обработки прерываний (Interrupt Service Routines – ISRs)

Прерывания с IRQ0 по IRQ31 обрабатываются ISRs.

Обработчики ошибок (Fault Handlers)

Обрабатываются только исключения Hard Fault.

Системные обработчики (System handlers)

Обрабатываются исключения NMI, PendSV, SVCcall, SysTick и Hard Fault.

8.3 Таблица векторов

Таблица векторов содержит указатель стека, вектор входа по RESET и стартовые адреса обработчиков, также называемых векторами. На рисунке 149 представлена последовательность векторов в таблице. Младший бит всех векторов должен быть равен 1, указывая на то, что обработчик выполняется в Thumb-режиме.

Номер исключения	номер IRQ	Вектор	Смещение
16+n	n	IRQn	0x40+4n
.		.	.
.		.	.
.		.	.
18	2	IRQ2	0x48
17	1	IRQ1	0x44
16	0	IRQ0	0x40
15	-1	если применяется SysTick	0x3C
14	-2	PendSV	0x38
13		Зарезервировано	
12			
11	-5	SVCcall	0x2C
10			
9			
8			
7			
6			
5			
4			
3	-13	HardFault	0x10
2	-14	NMI	0x0C
			0x08
1		Reset	0x04
		Начальное значение указателя стека	0x00

Рисунок 149 – Таблица векторов

При системном сбросе, таблица векторов располагается по фиксированному адресу 0x00000000.

8.4 Распределение векторов прерываний

Таблица 160 – Распределение векторов прерываний

Номер исключения	Номер прерывания	Смещение в таблице векторов	Название	Описание
		0x0000	Initial SP value	
1		0x0004	Reset	
2	-14	0x0008	NMI	
3	-13	0x000C	Hard fault	
4		-	-	
5		-	-	
6		-	-	
7		-	-	
8		-	-	
9		-	-	
10		-	-	
11	-5	0x002C	SVCall	
12		-	-	
13	-	-	-	
14	-2	0x0038	PendSV	
15	-1	0x003C	Systick	
16	0	0x0040	FT_ERR_IF	
17	1	0x0044	CLK_IF	Прерывание от CLOCK_CNTR по событиям ошибок в регистрах. Маскируется на уровне NVIC
18	2	0x0048	PVD_IF	
19	3	0x004C	RTC_IF	
20	4	0x0050	BKP_IF	
21	5	0x0054	EXTBUS_ERR_IF	
22	6	0x0058	DMA_DONE_IF	
23	7	0x005C	CCSDS_RX_IF	
24	8	0x0060	CCSDS_TX_IF	
25	9	0x0064	PORT_IF	
26	10	0x0068	SPW0_INT_IF	
27	11	0x006C	SPW1_INT_IF	
28	12	0x0070	TMR0_INT_IF	
29	13	0x0074	TMR1_INT_IF	
30	14	0x0078	TMR2_INT_IF	
31	15	0x007C	TMR3_INT_IF	
32	16	0x0080	CAN0_INT_IF	
33	17	0x0084	CAN1_INT_IF	
34	18	0x0088	UART0_INT_IF	
35	19	0x008C	UART1_INT_IF	
36	20	0x0090	MIL0_INT_IF	
37	21	0x0094	MIL1_INT_IF	
38	22	0x0098	MIL2_INT_IF	
39	23	0x009C	MIL3_INT_IF	
40	24	0x00A0	SSP0_INT_IF	
41	25	0x00A4	SSP1_INT_IF	
42	26	0x00A8	SSP2_INT_IF	
43	27	0x00AC	SSP3_INT_IF	
44	28	0x00B0	SSP4_INT_IF	
45	29	0x00B4	SSP5_INT_IF	

Номер исключения	Номер прерывания	Смещение в таблице векторов	Название	Описание
46	30	0x00B8	CRC_INT_IF	
47	31	0x00BC	ECC_INT_IF	

8.5 Приоритеты исключений

Меньшее значение приоритета означает больший приоритет. Конфигурируемы все приоритеты, кроме RESET и Hard Fault.

Если программное обеспечение не задает приоритетов, то все они имеют приоритет 0.

Конфигурируемый приоритет может быть в диапазоне от 0 до 192 с шагом 64. Это означает, что RESET, Hard Fault и NMI, имеющие отрицательное значение приоритета, всегда имеют больший приоритет.

Если имеется несколько исключений с одинаковым приоритетом, то больший приоритет имеет исключение с меньшим порядковым номером.

Если процессор выполняет обработчик исключения и происходит исключение с большим приоритетом, то происходит переход на обработчик исключения с большим приоритетом. Если при выполнении обработчика произошло исключение с таким же приоритетом, то это исключение будет выполнено по завершению текущего обработчика, несмотря на порядковый номер исключения.

8.6 Вход в обработчик и выход из обработчика

При описании используются следующие термины:

8.6.1 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. В случае, если внутри обработчика исключения возникает прерывание более высокого приоритета возникает ситуация, называемая вложенным исключением. Подробнее данный вопрос рассмотрен в подразделе «Вход в процедуру обработки исключения».

8.6.2 Возврат

Возврат из обработчика осуществляется по завершении обработки исключительной ситуации, с одновременным выполнением следующих условий:

- в системе отсутствуют необработанные исключения с достаточным приоритетом;
- завершённый обработчик не обрабатывал запоздавшее исключение (late-arriving exception).

Процессор обращается к стеку и восстанавливает состояние, имевшее место до вызова обработчика. Более подробная информация дана в подразделе «Возврат из обработчика исключения».

8.6.3 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 присутствуют исключения, требующие вызова обработчика, восстановление состояния процессора из стека не производится, а управление передается непосредственно на новый обработчик.

8.6.4 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.

8.6.5 Вход в процедуру обработки исключения

Вызов процедуры обработки исключения возникает в случае наличия необработанных исключительных ситуаций с достаточным приоритетом и выполнения одного из следующих условий:

- процессор находится в режиме приложения (thread mode);
- новая исключительная ситуация имеет приоритет выше, чем обрабатываемая в текущий момент времени, что приводит к приоритетному прерыванию выполнения текущего обработчика. В этом случае возникает вложение одного исключения в другое.

Для того, чтобы исключительная ситуация имела достаточный приоритет, необходимо, чтобы уровень ее приоритета был выше значений, заданных в регистрах маскирования. В противном случае исключение находится в состоянии ожидания, процедура его обработки не вызывается.

При необходимости вызова обработчика, за исключением случаев обработки запоздавшего исключения и передачи управления на обработчик без восстановления контекста, процессор заносит в текущий стек восемь слов данных, называемые далее стековым фреймом. Этот фрейм включает в себя следующие значения:

- Регистры R0-R3, R12;
- Адрес возврата;
- Регистр PSR;
- Регистр LR.

Указанная операция далее будет называться сохранением контекста. Непосредственно после ее выполнения указатель стека равен младшему адресу стекового фрейма.

Во время сохранения контекста производится выравнивание адреса стека по границе двойного слова.

Стековый фрейм содержит адрес возврата, указывающий на ближайшую невыполненную инструкцию прерванной программы. По завершении процедуры

обработки исключения значений адреса возврата заносится в счетчик команд, после чего выполнение программы возобновляется с прерванной точки.

Одновременно с сохранением контекста процессор осуществляет выборку адреса точки входа в процедуру обработки исключения из таблицы векторов исключений. По завершении операции сохранения контекста процессор передает управление на полученный из таблицы адрес.

Одновременно в регистр LR записывается значение EXC_RETURN, позволяющее определить, какой из двух указателей стека соответствует данному стековому фрейму и в каком режиме находился процессор перед входом в обработчик.

Если во время передачи управления не возникло исключения с более высоким приоритетом, процессор начинает выполнение вызванной процедуры обработки и автоматически изменяет состояние текущего прерывания с ожидающего обработки на активное.

В противном случае процессор передает управление обработчика высокоприоритетной исключительной ситуации без изменения состояния отложенного прерывания в соответствии с правилами, изложенными в разделе «Запоздавшее исключение».

8.6.6 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.

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 161.

Процессор устанавливает биты EXC_RETURN [31:4] в 0xFFFFFFFF. Загрузка данного значения в PC указывает на завершение процедуры обработки исключения и заставляет процессор выполнить необходимые действия для возврата из обработчика.

Таблица 161 – Возврат из обработчика исключения

EXC_RETURN [3:0]	Описание
bXXX0	Резерв.
B0001	Возврат в режим обработчика. Восстановление контекста осуществляется из стека MSP. Дальнейшая работа осуществляется со стеком MSP.
B0011	Резерв.
B01X1	Резерв.
B1001	Возврат в режим приложения. Восстановление контекста осуществляется из стека MSP. Дальнейшая работа осуществляется со стеком MSP.
B1101	Возврат в режим приложения. Восстановление контекста осуществляется из стека PSP. Дальнейшая работа осуществляется со стеком PSP.
B1X11	Резерв.

9 Типовая схема включения

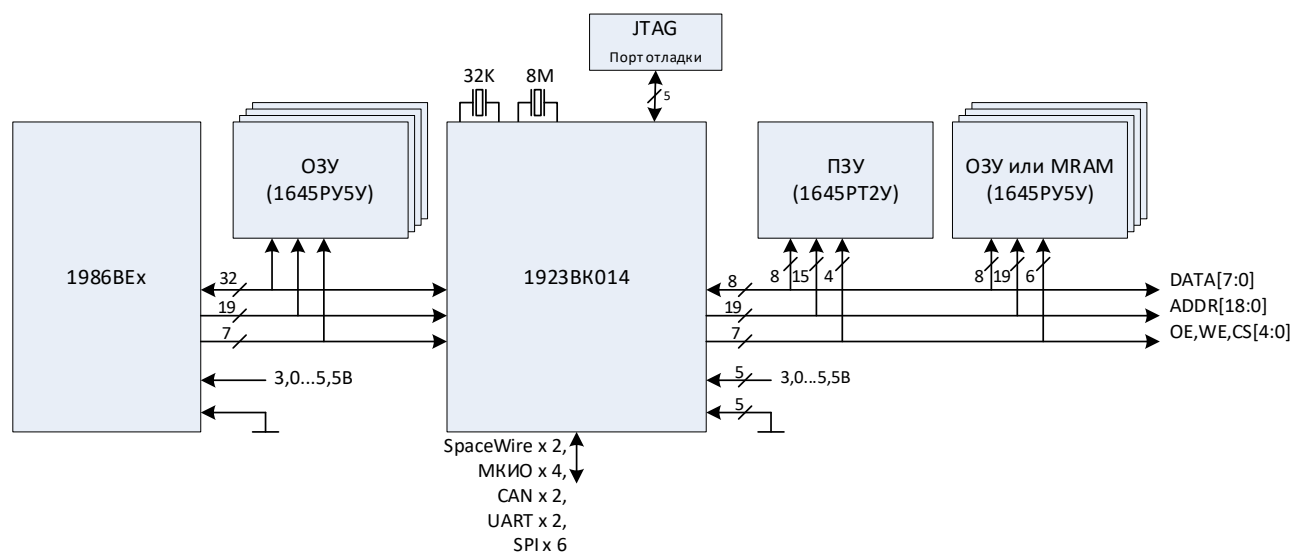


Рисунок 150 – Типовая схема включения микросхем

10 Типовые зависимости

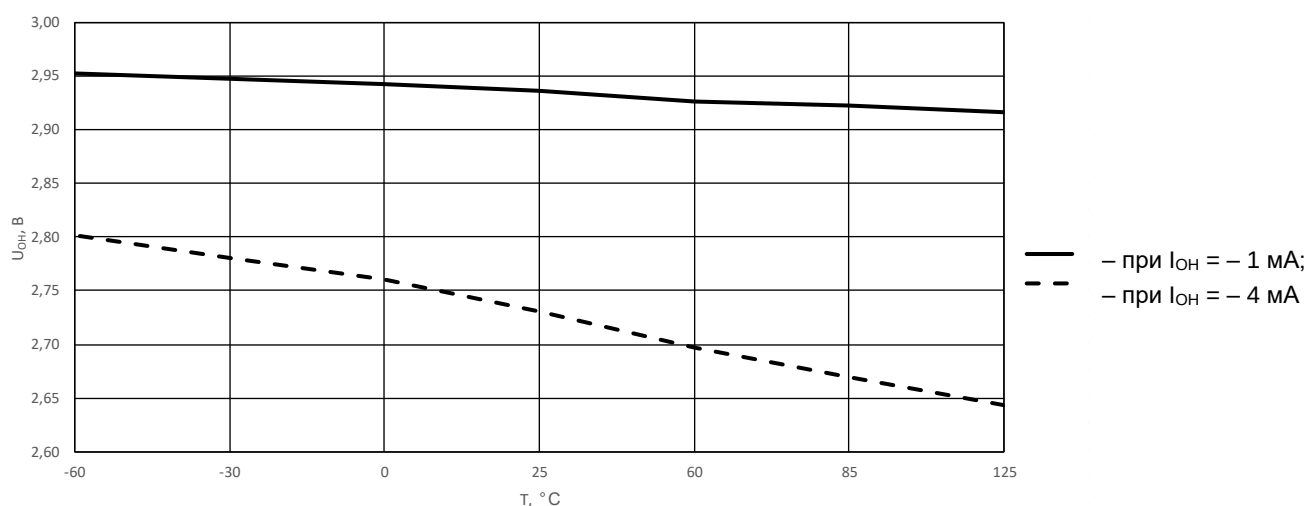


Рисунок 151 – Зависимость выходного напряжения высокого уровня от температуры среды при $I_{OH} = -1$ мА и $I_{OH} = -4$ мА, $U_{CC} = U_{CC_SPW} = 3,0$ В

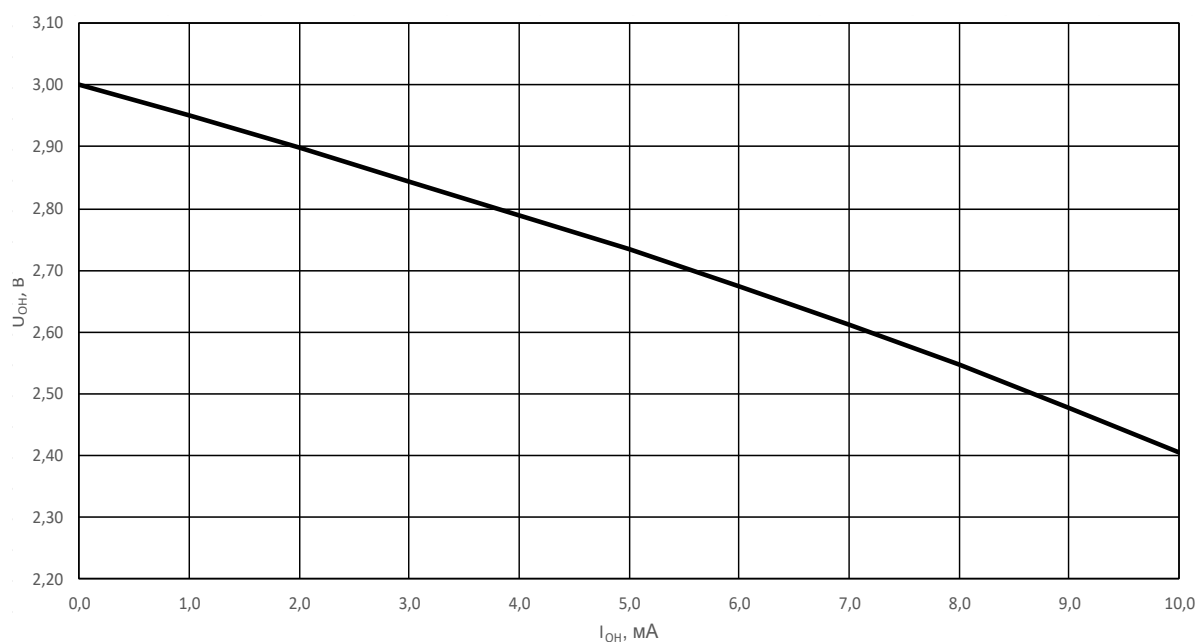


Рисунок 152 – Зависимость выходного напряжения высокого уровня от выходного тока высокого уровня в нормальных условиях при $U_{CC} = 3,0$ В

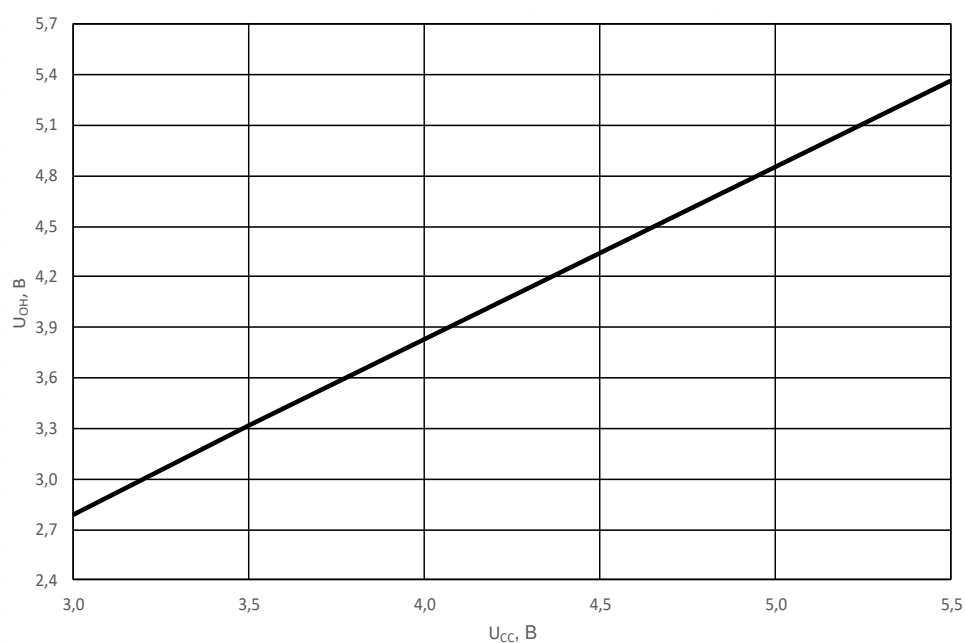


Рисунок 153 – Зависимость выходного напряжения высокого уровня от напряжения питания в нормальных условиях при $I_{OH} = 4,0 \text{ мА}$

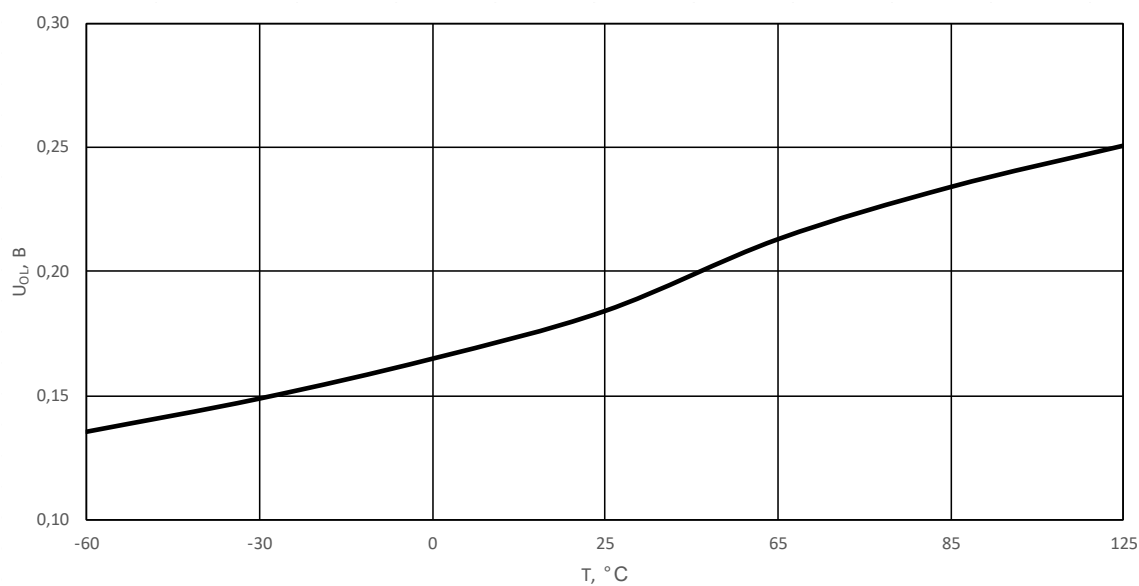


Рисунок 154 – Зависимость выходного напряжения низкого уровня от температуры среды при $U_{CC} = U_{CC_SPW} = 3,0 \text{ В}$, $U_{I_LDO} = 2,5 \text{ В}$, $I_{OL} = 4 \text{ мА}$

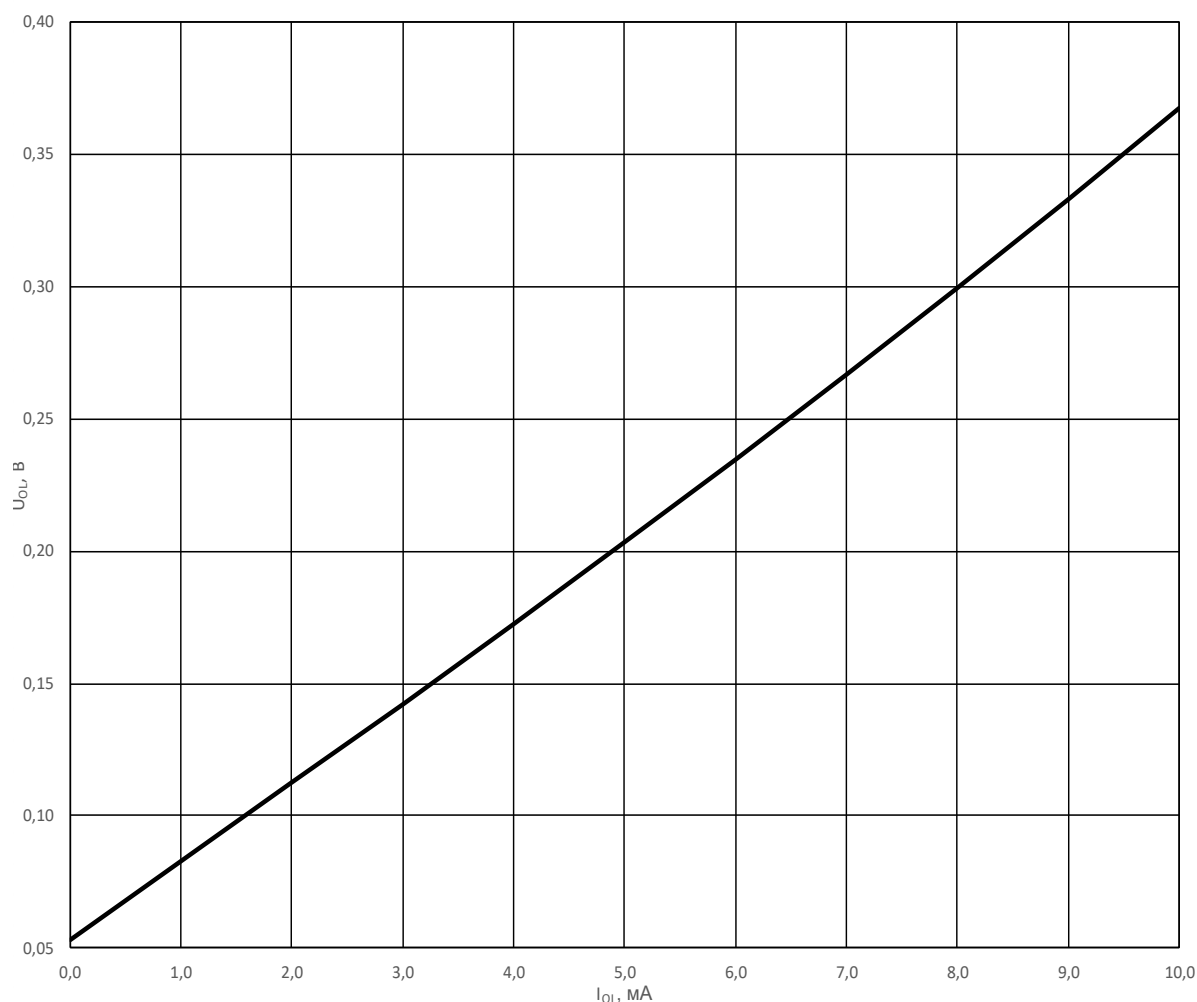


Рисунок 155 – Зависимость выходного напряжения низкого уровня от выходного тока низкого уровня в нормальных условиях при $U_{CC} = 3,0$ В

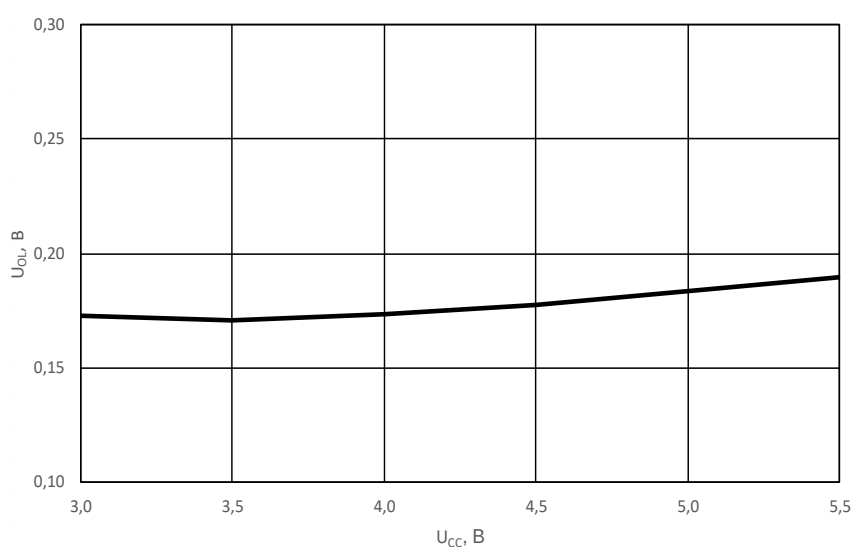


Рисунок 156 – Зависимость выходного напряжения низкого уровня от напряжения питания в нормальных условиях при $I_{OL} = 4,0$ мА

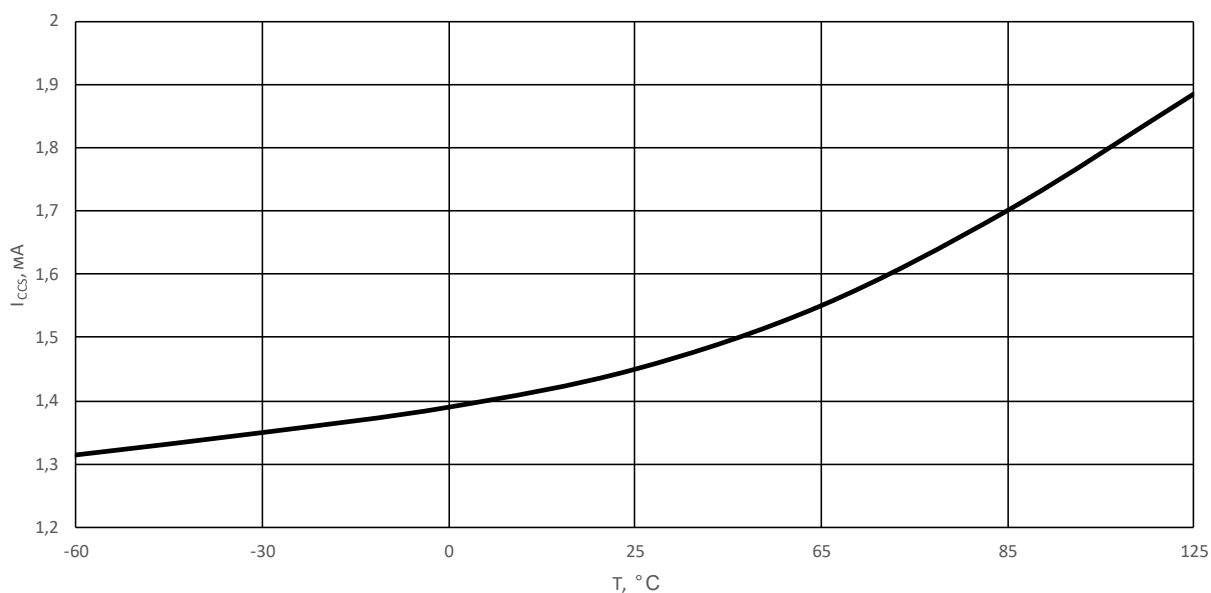


Рисунок 157 – Зависимость статического тока потребления от температуры среды при $U_{CC} = 5,5$ В

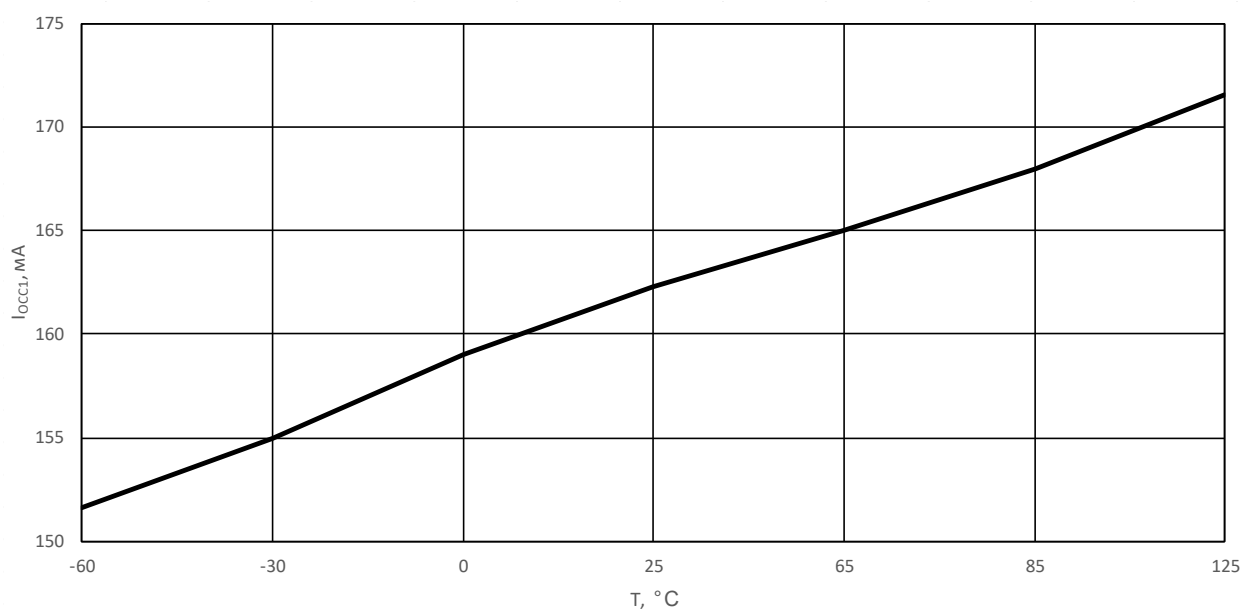


Рисунок 158 – Зависимость динамического тока потребления от температуры среды при $U_{CC} = U_{I_LDO} = U_{CC_SPW} = 3,6$ В, $f_C = 50$ МГц, $f_{C_PLL} = 10$ МГц, $f_{C_CPU} = 50$ МГц

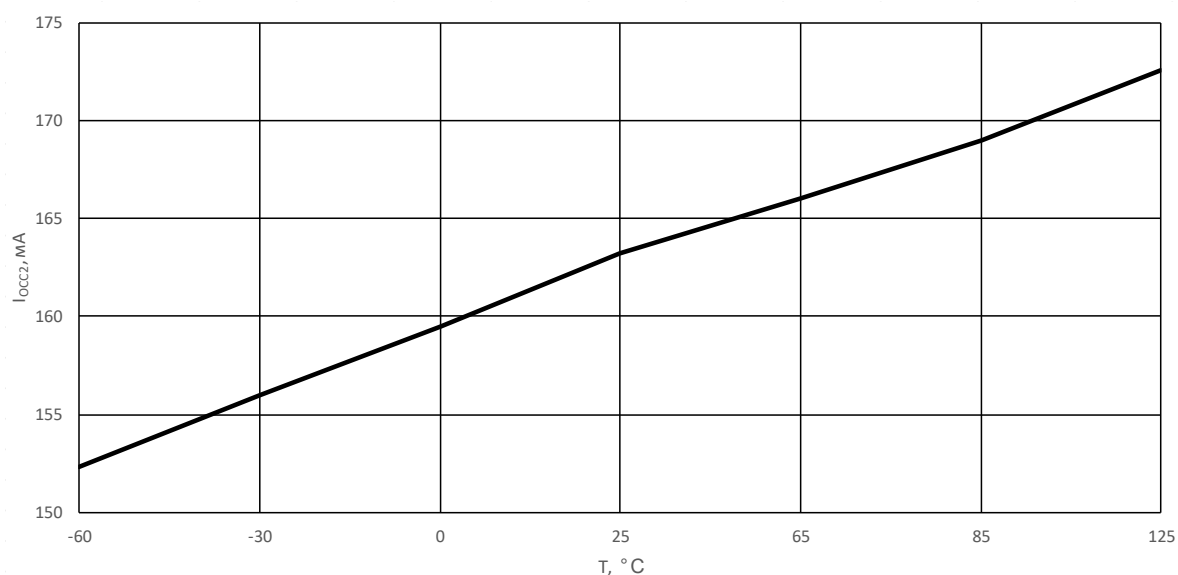


Рисунок 159 – Зависимость динамического тока потребления от температуры среды при $U_{CC} = U_{LDO} = U_{CC_SPW} = 5,5 \text{ В}$, $f_C = 50 \text{ МГц}$, $f_{C_PLL} = 10 \text{ МГц}$, $f_{C_CPU} = 50 \text{ МГц}$

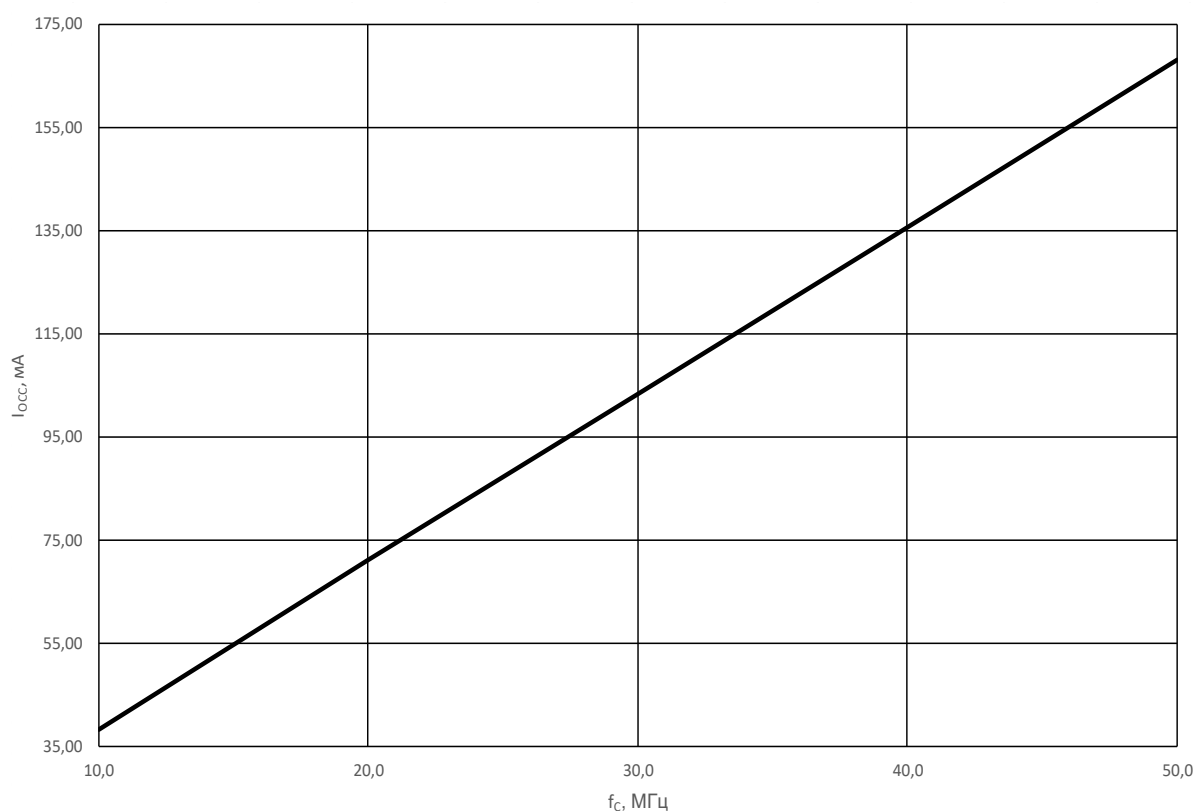


Рисунок 160 – Зависимость динамического тока потребления от частоты следования импульсов тактовых сигналов в нормальных условиях при $U_{CC} = 3,0 \text{ В}$

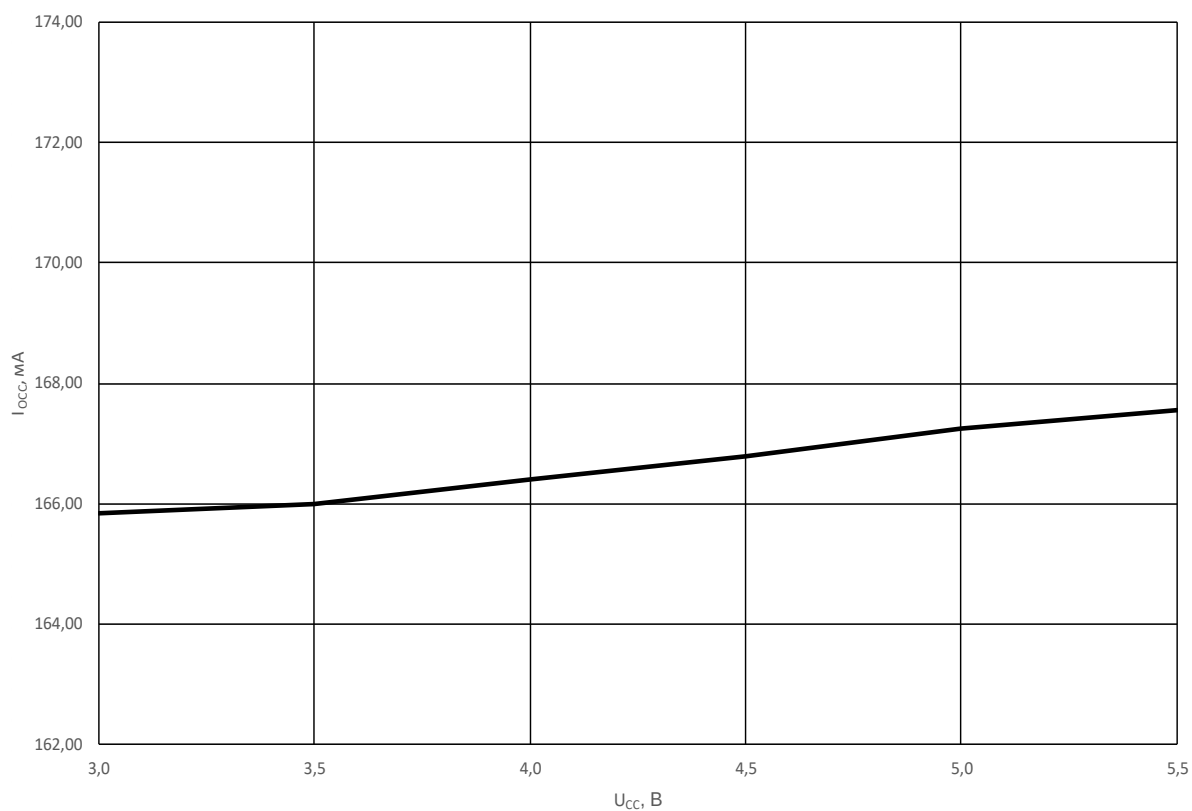


Рисунок 161 – Зависимость динамического тока потребления от напряжения питания в нормальных условиях при $f_c = 50$ МГц

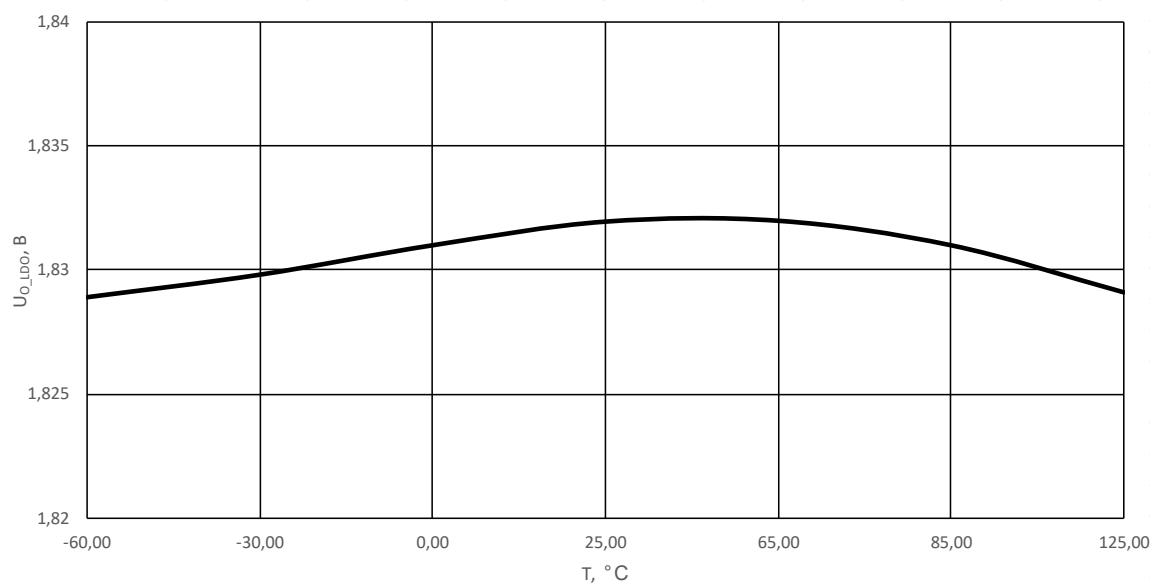


Рисунок 162 – Зависимость выходного напряжения встроенного регулятора от температуры среды при $U_{CC} = U_{CC_SPW} = 3,0$ В, $U_{I_LDO} = 2,5$ В

11 Электрические параметры микросхемы

Таблица 162 – Электрические параметры микросхем при приёмке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °C
		не менее	не более	
Выходное напряжение высокого уровня, В, на входах / выходах портов А – Е, при: – $I_{OH} = -1 \text{ мА}$; – $I_{OH} = -4 \text{ мА}$	U_{OH}	2,8	–	25, 125, – 60
		2,4	–	
Выходное напряжение низкого уровня, В, на входах / выходах портов А – Е	U_{OL}	–	0,4	
Напряжение питания формирования сигнала «Сброс» встроенного регулятора, В – при повышении U_{CC} ; – при понижении U_{CC}	U_{POR}	–	3,0	
		1,2	–	
Выходное напряжение встроенного регулятора, В	U_{O_LDO}	1,62	1,98	
Входной ток высокого уровня, мкА, на выводах SHDN, JTAG_EN	I_{IH}	10	500	
Входной ток низкого уровня, мкА, на выводах nRESET0, nRESET1	I_{IL}	– 500	– 10	
Входной ток утечки высокого уровня, мкА, – на входах / выходах портов А – Е в цифровом режиме, на выводах генератора тактового сигнала – на выводах nRESET0, nRESET1, при $U_{IH2} = U_{CC}$	I_{ILH}	– 10	10	
Входной ток утечки низкого уровня, мкА, – на входах/выходах портов А – Е в цифровом режиме, на выводах генератора тактового сигнала – на выводах SHDN, JTAG_EN, при $U_{IL} = 0 \text{ В}$	I_{ILL}	– 10	10	
Входной ток, мкА, на входах/выходах портов А – Е в аналоговом режиме, при $U_i = 2,5 \text{ В}$	I_{IN1}	– 500	500	
Входной ток, мкА, на выводах приемника SpaceWire: – на выводах SPW0_RXDP, SPW0_RXSP, SPW1_RXDP, SPW1_RXSP, при $U_{IN_SPW} = U_{CC_SPW}$ на выводах SPW0_RXDP, SPW0_RXSP, SPW1_RXDP, SPW1_RXSP; – на выводах SPW0_RXDP, SPW0_RXSP, SPW1_RXDP, SPW1_RXSP, при $U_{IN_SPW} = 0 \text{ В}$ на выводах SPW0_RXDP, SPW0_RXSP, SPW1_RXDP, SPW1_RXSP; – на выводах SPW0_RXDN, SPW0_RXSN, SPW1_RXDN, SPW1_RXSN, при $U_{IN_SPW} = U_{CC_SPW}$ на выводах SPW0_RXDN, SPW0_RXSN, SPW1_RXDN, SPW1_RXSN; – на выводах SPW0_RXDN, SPW0_RXSN, SPW1_RXDN, SPW1_RXSN, при $U_{IN_SPW} = 0 \text{ В}$ на выводах SPW0_RXDN, SPW0_RXSN, SPW1_RXDN, SPW1_RXSN	I_{IN2}	6	21	
		– 6	6	
		– 6	6	
		– 21	– 6	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °C
		не менее	не более	
Статический ток потребления, mA	I _{CCS}	–	5	25, 125, – 60
Динамический ток потребления, mA, при U _{CC} = U _{I_LDO} = U _{CC_SPW} = 3,6 B	I _{OCC1}	–	300	
Динамический ток потребления, mA, при U _{CC} = U _{I_LDO} = U _{CC_SPW} = 5,5 B	I _{OCC2}	–	500	
Параметры генератора тактовой частоты				
Выходная частота генератора HSE, МГц, в режиме резонатора максимальная	f _{O_HSE}	30	–	25, 125, – 60
минимальная		–	1	
Максимальная выходная частота генератора HSE, МГц, в режиме генератора	f _{O_HSEmax}	50	–	
Выходная частота генератора HSI, МГц	f _{O_HSI}	1	16	
Выходная частота генератора LSI, кГц	f _{O_LSI}	20	60	
Параметры блока умножения тактовой частоты PLL				
Выходная частота блока умножения тактовой частоты PLL, МГц максимальная	f _{O_PLL}	100	–	25, 125, – 60
минимальная		–	40	
Коэффициент деления входной частоты, 1	K _{QPLL}	1	16	
Коэффициент деления синтезируемой частоты, 1	K _{NPLL}	4	75	
Параметры SpaceWire				
Выходное напряжение дифференциальное, мВ	U _{OD}	250	450	25, 125, – 60
Выходное напряжение синфазное, В	U _{OC}	1,125	2,0	
Примечание – Режимы измерения параметров приведены в АЕНВ.431290.327ТУ.				

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В.

12 Предельно допустимые характеристики микросхемы

Таблица 163 – Предельно-допустимые режимы эксплуатации и предельные режимы микросхем

Наименование параметра, единица измерения	Буквенное обозначение параметра	Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение источника питания, В, на выводах VDD	U_{CC}	3,0	5,5	– 0,3	6,0
Напряжение питания SpaceWire, В, на выводе VDD_SPW	$U_{CC_SPW}^{1)}$	3,0	5,5	– 0,3	6,0
Напряжение питания цифровой части, В, на выводах VDDO, VDO_PLL	U_{CCD}	1,62	1,98	–	–
Входное напряжение встроенного регулятора LDO, В, на выводах VI_LDO	U_{I_LDO}	2,5	U_{CC}	– 0,3	6,0
Входное напряжение высокого уровня, В, на 5 В толерантных выводах: входах / выходах портов A – E	U_{IH1}	2,0	5,5	–	6,0
Входное напряжение высокого уровня, В, на выводах JTAG_EN, SHDN, nRESET0, nRESET1	U_{IH2}	2,0	U_{CC}	–	$U_{CC} + 0,3$
Входное напряжение низкого уровня, В, на 5 В толерантных выводах: входах / выходах портов A – E; на выводах JTAG_EN, SHDN, nRESET0, nRESET1	U_{IL}	0	0,8	– 0,3 ²⁾	–
Выходной ток высокого уровня, мА	I_{OH}	– 4	–	– 10	–
Выходной ток низкого уровня, мА	I_{OL}	–	4	–	10
Частота следования импульсов тактовых сигналов, МГц	f_C	–	50	–	–
Частота следования импульсов CLK, МГц	f_{C_CLK}	–	50	–	–
Частота следования импульсов CPU CLK, МГц	f_{C_CPU}	–	50	–	–
Частота следования импульсов тактовых сигналов PLL, МГц	f_{C_PLL}	2	16	–	–
Емкость нагрузки на выводах, пФ	C_L	–	50	–	–
Параметры генератора тактовых сигналов					
Входное напряжение высокого уровня, В, на выводах генератора тактового сигнала в режиме BYPASS	U_{IH_HSE}	$0,8 \cdot U_{CC}$	U_{CC}	–	$U_{CC} + 0,3$
Входное напряжение низкого уровня, В, на выводах генератора тактового сигнала в режиме BYPASS	U_{IL_HSE}	0	$0,2 \cdot U_{CC}$	– 0,3	–

Наименование параметра, единица измерения	Буквенное обозначение параметра	Предельно- допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Амплитуда входного импульса генератора HSE, В, в режиме OSC	U_{HSE}	0,05	U_{CC}	—	—
Частота следования импульсов тактовых сигналов HSE, МГц, в режиме генератора	f_{C_HSEmax}	—	50	—	—
Частота следования импульсов тактовых сигналов HSE, МГц, в режиме резонатора	f_{C_HSE}	1	30	—	—
Параметры SpaceWire					
Входное напряжение на выводах приемника SpaceWire, В	U_{IN_SPW}	0	U_{CC_SPW}	— 0,3	$U_{CC} + 0,3$
Входное напряжение дифференциальное, В	U_{ID_SPW}	0,2	0,6	—	—
Входное напряжение синфазное, В	U_{IS_SPW}	0,05	2,35	—	—
1) Напряжение питания SpaceWire U_{CC_SPW} должно отличаться от напряжения источника питания U_{CC} не более, чем на $\pm 0,2$ В. 2) Режим без ограничения входного тока. Допускается вместо ограничения входного напряжения ограничивать входной вытекающий ток не более 1 мА. Примечание – Не допускается одновременное задание более одного предельного режима.					

13 Справочные параметры

Зависимости основных электрических параметров микросхем от режимов и условий эксплуатации приведены на рисунках 151 – 162

Значение собственной резонансной частоты не менее 4,4 кГц.

Тепловое сопротивление кристалл-окружающая среда в нормальных условиях не более 6,7 °С/Вт.

Справочные параметры микросхемы приведены в таблице 164.

Предельная температура р-п перехода кристалла 150 °С.

Таблица 164 – Справочные параметры микросхемы

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °C
		не менее	не более	
Питание				
Напряжение питания U _{сс} разрешения работы блока сброса POVR, В	U _{POVR_max}	5,5	–	25, 125, – 60
Напряжение питания U _{сс} запрещения работы блока сброса POVR, В	U _{POVR_min}	–	5,7	
Ток потребления по выводам SPW_VDD, мА	I _{CC4}	–	40	
Ток потребления по выводам SPW_VDD в режиме замыкания, мА	I _{CC5}	–	40	
Время задержки сигнала сброса от момента появления питания, мс	t _{PORRSTn}	2	8	
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+1024 тактов	25, 125, – 60
Длительность фильтруемых иголок в HSE, нс	t _{HSEFILTER}	–	3	
Время запуска генератора LSE, нс, в режиме резонатора	t _{LSEON}	–	100+1024 тактов	
Длительность фильтруемых иголок в LSE, нс	t _{LSEFILTER}	–	3	
Время запуска PLL, мкс	t _{PLLON}	–	60	

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °C
		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 передатчика SpaceWire, Ом	R _{TX_OUT_SPW}	98	224	
Время нарастания/спада выходного сигнала, нс, (20 – 80) %	t _{RISE_TX} t _{FALL_TX}	–	0,8	
Скважность выходного сигнала, %	CK _{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	–	
Источник опорного напряжения				
Сопротивление внешнего токозадающего резистора на выводе EXTR, кОм	R _{L_EXTR}	114	126	25, 125, – 60

14 Габаритный чертеж микросхемы

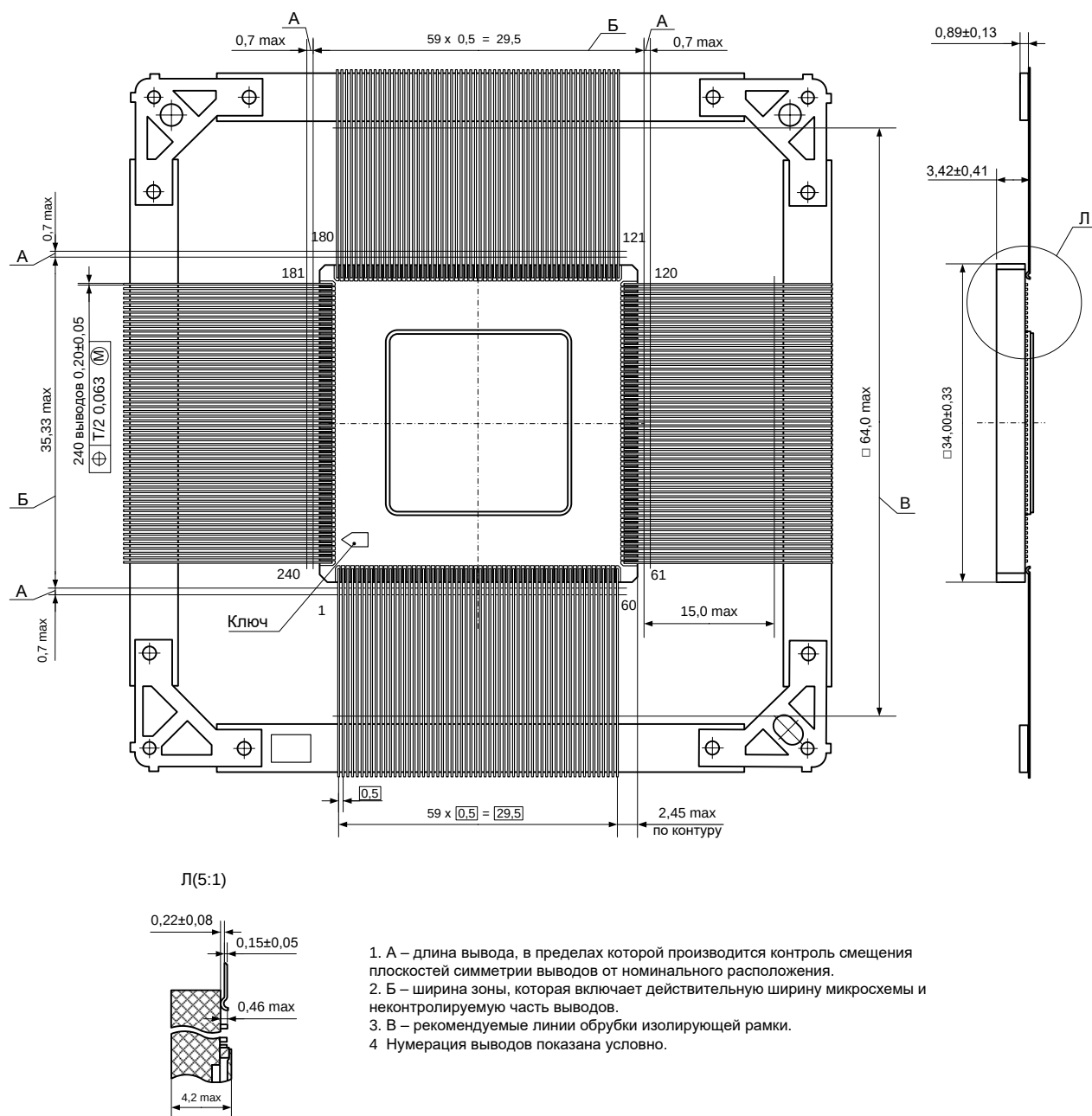
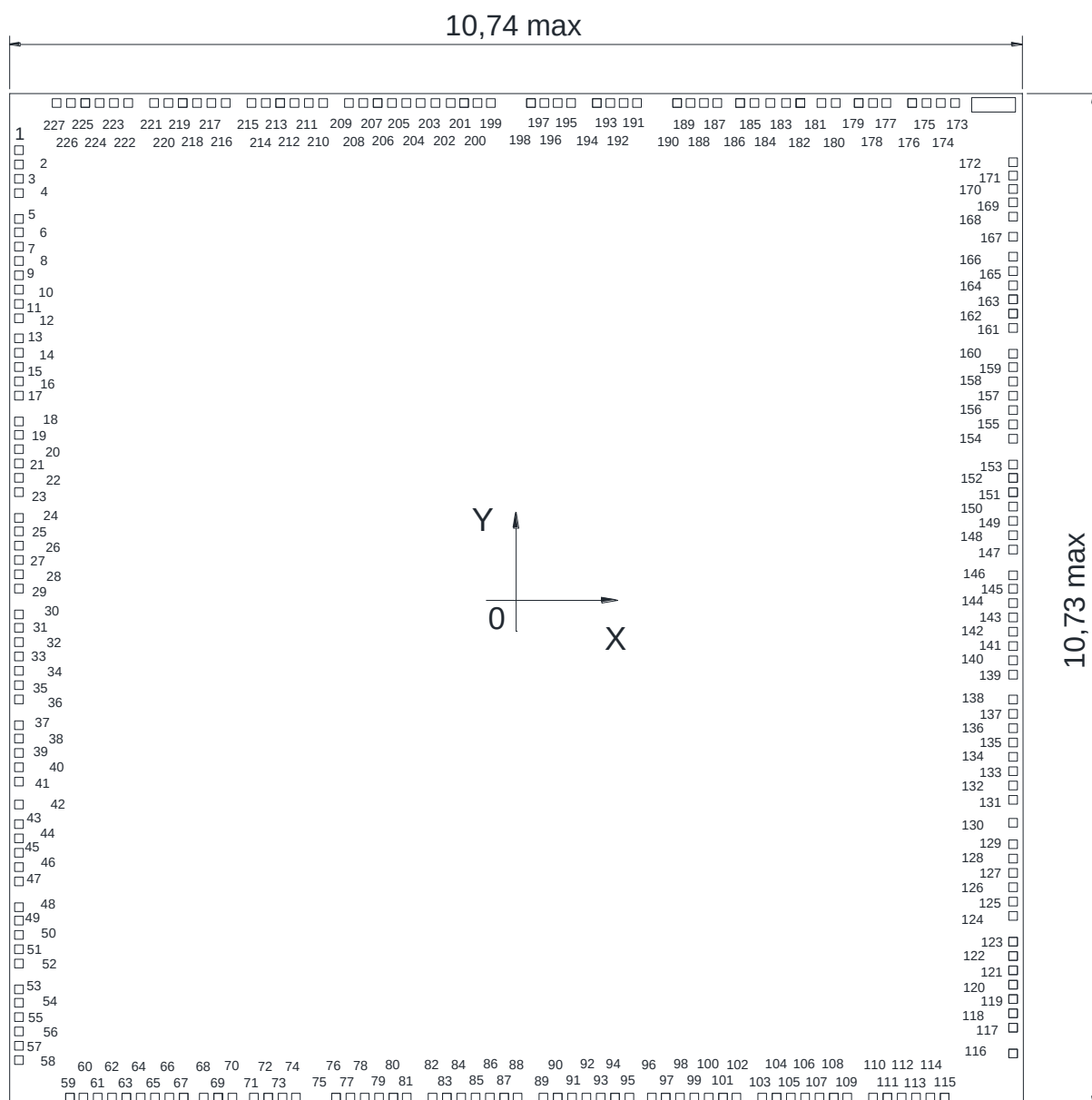


Рисунок 163 – Микросхема в корпусе 4245.240-6.01



- 1 Размеры контактных площадок (КП) кристалла (90 x 90) мкм.
Материал КП - AlCu (Cu 0,5%).
- 2 Толщина кристалла (0,48 ± 0,025)мм.
- 3 A - маркировка MLDR132, показана условно.
- 4 Координаты КП - см. таблицу ниже.
- 5 Номера КП, кроме первой, присвоены условно. Расположение КП соответствует топологическому чертежу.

Рисунок 164 – Кристалл (бескорпусное исполнение)

Таблица 165 – Координаты КП кристалла

№ КП	Обозначение КП	Координаты КП		№ КП	Обозначение КП	Координаты КП	
		Х	У			Х	У
1	OSCO2	-5202,900	4712,175	47	PD28	-5202,900	-2942,825
2	OSCI2	-5202,900	4562,175	48	PD27	-5202,900	-3212,825
3	PE31	-5202,900	4412,175	49	PD26	-5202,900	-3352,825
4	PE30	-5202,900	4262,175	50	PD25	-5202,900	-3502,825
5	PE29	-5202,900	3992,175	51	PD24	-5202,900	-3652,825
6	PE28	-5202,900	3852,175	52	JTAG_EN	-5202,900	-3802,825
7	PE27	-5202,900	3702,175	53	PD23	-5202,900	-4072,825
8	PE26	-5202,900	3552,175	54	PD22	-5202,900	-4212,825
9	PE25	-5202,900	3402,175	55	PD21	-5202,900	-4362,825
10	PE24	-5202,900	3252,175	56	PD20	-5202,900	-4512,825
11	VDD	-5202,900	3102,175	57	PD19	-5202,900	-4662,825
12	VI_LDO	-5202,900	2952,175	58	PD18	-5202,900	-4812,825
13	VI_LDO	-5202,900	2742,175	59	PD17	-4668,900	-5205,825
14	GND	-5202,900	2592,175	60	PD16	-4528,900	-5205,825
15	GND	-5202,900	2442,175	61	PD15	-4378,900	-5205,825
16	PE23	-5202,900	2292,175	62	PD14	-4228,900	-5205,825
17	PE22	-5202,900	2142,175	63	SHDN	-4078,900	-5205,825
18	PE21	-5202,900	1872,175	64	nRESET1	-3928,900	-5205,825
19	PE20	-5202,900	1732,175	65	nRESET0	-3778,900	-5205,825
20	PE19	-5202,900	1582,175	66	VDD	-3628,900	-5205,825
21	PE18	-5202,900	1432,175	67	VI_LDO	-3478,900	-5205,825
22	PE17	-5202,900	1282,175	68	VI_LDO	-3268,900	-5205,825
23	PE16	-5202,900	1132,175	69	GND	-3117,970	-5205,825
24	PE15	-5202,900	862,175	70	GND	-2968,900	-5205,825
25	PE14	-5202,900	722,175	71	VDDO	-2743,900	-5205,825
26	PE13	-5202,900	572,175	72	PD13	-2593,900	-5205,825
27	PE12	-5202,900	422,175	73	PD12	-2443,900	-5205,825
28	PE11	-5202,900	272,175	74	PD11	-2303,900	-5205,825
29	PE10	-5202,900	122,175	75	PD10	-2033,900	-5205,825
30	PE9	-5202,900	-147,825	76	PD9	-1883,900	-5205,825
31	PE8	-5202,900	-287,825	77	PD8	-1733,900	-5205,825
32	PE7	-5202,900	-437,825	78	PD7	-1583,900	-5205,825
33	PE6	-5202,900	-587,825	79	PD6	-1433,900	-5205,825
34	PE5	-5202,900	-737,825	80	PD5	-1283,900	-5205,825
35	PE4	-5202,900	-887,825	81	PD4	-1143,900	-5205,825
36	PE3	-5202,900	-1037,825	82	PD3	-873,900	-5205,825
37	PE2	-5202,900	-1307,825	83	PD2	-723,900	-5205,825
38	PE1	-5202,900	-1447,825	84	PD1	-573,900	-5205,825
39	PE0	-5202,900	-1597,825	85	PD0	-423,900	-5205,825
40	PD31	-5202,900	-1747,825	86	PC31	-273,900	-5205,825
41	PD30	-5202,900	-1897,825	87	PC30	-123,900	-5205,825
42	VDDO	-5202,900	-2137,825	88	PC29	16,100	-5205,825
43	VDD	-5202,900	-2342,825	89	PC28	286,100	-5205,825
44	VI_LDO	-5202,900	-2492,825	90	PC27	436,100	-5205,825
45	GND	-5202,900	-2642,825	91	PC26	586,100	-5205,825
46	PD29	-5202,900	-2792,825	92	PC25	736,100	-5205,825
93	PC24	886,100	-5205,825	140	PB18	5202,900	-628,825
94	PC23	1036,100	-5205,825	141	PB17	5202,900	-478,825
95	PC22	1186,100	-5205,825	142	PB16	5202,900	-328,825
96	VDD	1421,100	-5205,825	143	PB15	5202,900	-178,825

П №	Обозначение КП	Координаты КП		П №	Обозначение КП	Координаты КП	
		Х	У			Х	У
97	VI_LDO	1566,100	-5205,825	144	PB14	5202,900	-28,825
98	GND	1716,100	-5205,825	145	PB13	5202,900	121,175
99	PC21	1866,100	-5205,825	146	PB12	5202,900	261,175
100	PC20	2016,100	-5205,825	147	PB11	5202,900	531,175
101	PC19	2166,100	-5205,825	148	PB10	5202,900	681,175
102	PC18	2306,100	-5205,825	149	PB9	5202,900	831,175
103	PC17	2576,100	-5205,825	150	PB8	5202,900	981,175
104	PC16	2726,100	-5205,825	151	PB7	5202,900	1131,175
105	PC15	2876,100	-5205,825	152	PB6	5202,900	1281,175
106	PC14	3026,100	-5205,825	153	PB5	5202,900	1421,175
107	PC13	3176,100	-5205,825	154	PB4	5202,900	1691,175
108	PC12	3326,100	-5205,825	155	PB3	5202,900	1841,175
109	PC11	3466,100	-5205,825	156	PB2	5202,900	1991,175
110	PC10	3736,100	-5205,825	157	PB1	5202,900	2141,175
111	PC9	3886,100	-5205,825	158	PB0	5202,900	2291,175
112	PC8	4036,100	-5205,825	159	PA31	5202,900	2441,175
113	PC7	4186,100	-5205,825	160	PA30	5202,900	2581,175
114	PC6	4336,100	-5205,825	161	PA29	5202,900	2851,175
115	PC5	4486,100	-5205,825	162	PA28	5202,900	3001,175
116	PC4	5202,900	-4743,825	163	VI_LDO	5202,900	3151,175
117	PC3	5202,900	-4473,825	164	VDD	5202,900	3301,175
118	PC2	5202,900	-4323,825	165	TESTI	5202,900	3451,175
119	PC1	5202,900	-4173,825	166	GND	5202,900	3601,175
120	PC0	5202,900	-4023,825	167	VDD_PLL	5202,900	3806,175
121	PB31	5202,900	-3873,825	168	VDD_PLL	5202,900	4015,175
122	PB30	5202,900	-3723,825	169	OSCI0	5202,900	4166,175
123	PB29	5202,900	-3573,825	170	OSCO0	5202,900	4306,175
124	PB28	5202,900	-3303,825	171	OSCI1	5202,900	4446,175
125	VDD	5202,900	-3153,825	172	OSCO1	5202,900	4586,175
126	VI_LDO	5202,900	-3003,825	173	PA27	4596,100	5205,825
127	VI_LDO	5202,900	-2853,825	174	PA26	4446,100	5205,825
128	GND	5202,900	-2703,825	175	PA25	4296,100	5205,825
129	GND	5202,900	-2553,825	176	PA24	4146,100	5205,825
130	VDDO	5202,900	-2328,825	177	PA23	3876,100	5205,825
131	PB27	5202,900	-2088,825	178	PA22	3736,100	5205,825
132	PB26	5202,900	-1938,825	179	PA21	3586,100	5205,825
133	PB25	5202,900	-1788,825	180	VDDO	3346,100	5205,825
134	PB24	5202,900	-1638,825	181	VI_LDO	3196,100	5205,825
135	PB23	5202,900	-1488,825	182	VI_LDO	2976,100	5205,825
136	PB22	5202,900	-1338,825	183	VDD	2826,100	5205,825
137	PB21	5202,900	-1188,825	184	GND	2661,100	5205,825
138	PB20	5202,900	-1048,825	185	GND	2496,100	5205,825
139	PB19	5202,900	-778,825	186	SPW_GND	2345,100	5205,825
187	SPW1_RXDP	2106,100	5205,825	208	PA19	-1598,900	5205,825
188	SPW1_RXDN	1966,100	5205,825	209	PA18	-1748,900	5205,825
189	SPW1_RXSP	1826,100	5205,825	210	PA17	-2018,900	5205,825
190	SPW1_RXSN	1686,100	5205,825	211	PA16	-2168,900	5205,825
191	SPW1_TXSP	1266,100	5205,825	212	PA15	-2318,900	5205,825
192	SPW1_TXSN	1126,100	5205,825	213	PA14	-2468,900	5205,825
193	SPW1_TXDP	986,100	5205,825	214	PA13	-2618,900	5205,825
194	SPW1_TXDN	846,100	5205,825	215	PA12	-2768,900	5205,825

П №	Обозначение КП	Координаты КП		П №	Обозначение КП	Координаты КП	
		Х	У			Х	У
195	SPW0_RXDP	576,100	5205,825	216	PA11	-3038,900	5205,825
196	SPW0_RXDN	436,100	5205,825	217	PA10	-3188,900	5205,825
197	SPW0_RXSP	296,100	5205,825	218	PA9	-3338,900	5205,825
198	SPW0_RXSN	156,100	5205,825	219	PA8	-3488,900	5205,825
199	SPW0_TXSP	-263,900	5205,825	220	PA7	-3638,900	5205,825
200	SPW0_TXSN	-403,900	5205,825	221	PA6	-3788,900	5205,825
201	SPW0_TXDP	-543,900	5205,825	222	PA5	-4058,900	5205,825
202	SPW0_TXDN	-683,900	5205,825	223	PA4	-4208,900	5205,825
203	VDD_SPW	-843,900	5205,825	224	PA3	-4358,900	5205,825
204	VDD	-998,900	5205,825	225	PA2	-4508,900	5205,825
205	VI_LDO	-1148,900	5205,825	226	PA1	-4658,900	5205,825
206	GND	-1298,900	5205,825	227	PA0	-4808,900	5205,825
207	PA20	-1448,900	5205,825				

15 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон, °С
1923BK014	1923BK014	4245.240-6.01	от – 60 до 125
K1923BK014H4	K1923BK014H4 (на таре)	бескорпусная	от 0 до 70

Примечание – Микросхемы в бескорпусном исполнении поставляются в виде отдельных кристаллов, получаемых разделением пластины. Микросхемы поставляются в таре (кейсах) без потери ориентации.

Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	23.04.2018	2.0.0	Введена впервые	Все
2	29.08.2018	2.1.0	Внесены исправления в таблицы 37, 112; в описания регистров PER0_CLK, PER1_CLK, FILTER[x]. Внесено дополнение в таблицу 34	102, 351, 352 106, 107, 108, 109, 362 90
3	15.11.2018	2.2.0	Исправлено описание регистров контроллера тактовых частот (раздел 6.1.1)	99 – 130
4	28.11.2018	2.3.0	В таблице 147 добавлены базовые адреса для контроллера МКПД	433
5	22.04.2019	2.4.0	Плановая корректировка спецификации по замечаниям. Внесение исправлений: на рисунках 8, 91-94, 98-108; в таблицах 9, 24, 73, 110, 114, 115, 150; в подразделах 5.5.3, 6.1.1.5, 6.10.1, 6.12.1, 6.14.16, 6.20.7.2, 6.21.2	42, 340, 341, 357, 363-370; 52, 71, 238, 277, 376, 378, 443; 54, 107, 257-275, 283, 345-355, 466, 473
6	15.03.2021	2.5.0	Внесение исправлений: в таблицах 2, 9, 116, 143, 154; в рисунках 1, 2, 80, 138; в подразделах 5.5.3, 5.5.5, 5.5.7, 5.8, 5.10.1, 6.1.1.4, 6.1.1.21, 6.2.3.6, 6.11, 6.13.5, 6.13.9.9, 6.14.8, 6.14.12, 6.14.16.3, 6.14.16.7, 6.15.5.1, 6.15.2.4, 6.15.2.9, 6.15.6.2, 6.20.4, 6.20.6.1, 6.20.7.2	21, 51, 376, 411, 459 3, 475 54-56, 80, 85, 105, 118, 142, 299, 300, 321, 336, 337, 347, 349, 359, 463, 464
7	01.04.2021	2.6.0	Исключены микросхемы ОТК. Исправление на рисунках 1, 2 и 138 (удален сигнал PE). Добавлено примечание к описанию регистра REGION[n].CNTRL	По тексту 3, 475 276
8	25.10.2021	2.7.0	Исправление опечаток Корректировка перечня ИМС Рисунки 16, 17 скорректированы Таблица 32 – добавлено примечание для регистра AMASK Подраздел 5.8.1 – добавлено описание поведения контроллера для первой ревизии Подраздел 6.10 – добавлено описание условия возникновения прерывания на портах Подраздел 5.3.5 – изменен в связи с отсутствием вывода FT_RESET Подраздел 6.17.9 – дополнено описание режима работы ОУ-ОУ Добавлены разделы «Контроллер прерываний NVIC» и «Прерывания и исключения» Подраздел «Распределение векторов прерываний» перенесен в раздел «Прерывания и исключения»	По тексту По тексту 73, 74 83 82 251 45 420 469, 478 481