

Техническое описание K1948BK018, K1948BK015
RISC-V микроконтроллер MIK32 АМУР

Содержание

1 Назначение	3
2 Технические характеристики	4
3 Состав МК32.....	5
3.1 Коммутационная матрица АНВ	11
3.2 Карта памяти	11
3.3 Порядок загрузки	13
3.4 Контроллер ПДП (DMA).....	14
3.5 Блок управления тестированием (TCB).....	29
3.6 Модуль USART	34
3.7 Модуль I2C	53
3.8 Модуль 32-разрядного таймера (TIMER32).....	106
3.9 Модуль 16-разрядного таймера (TIMER16).....	117
3.10 Часы реального времени (RTC).....	143
3.11 Программируемый контроллер прерываний (EPIC)	152
3.12 Блок вычисления контрольных сумм (CRC32).....	156
3.13 Модуль SPI	160
3.14 Модуль порта ввода/вывода общего назначения (GPIO)	181
3.15 Контроллер внешней памяти (SPIFI)	188
3.16 Ускоритель симметричной криптографии (CRYPTO).....	219
3.17 Контроллер блока однократно программируемой памяти (OTP)	223
3.18 Контроллер EEPROM	235
3.19 Модуль сторожевого таймера (WDT)	247
3.20 Сторожевой таймер шины (WDT BUS)	249
3.21 Температурный сенсор (TSENS)	252
3.22 Цифро-аналоговый преобразователь (DAC)	257
3.23 Аналого-цифровой преобразователь (ADC).....	262
3.24 Блок управления режимом загрузки (BOOT MANAGER)	269
4 Подсистема тактирования.....	270
4.1 Встроенные осцилляторы с частотой 32 МГц и 32 кГц.....	278
4.2 Вывод для подключения внешнего кварцевого резонатора до 32 МГц.....	279
4.3 Вывод для подключения внешнего кварцевого резонатора 32 кГц	280
4.4 Монитор частоты системного домена	280
5 Подсистема питания	285
5.1 Блок управления питанием и тактированием (POWER MANAGER)	290
5.2 Блок WakeUp	297
5.3 Монитор питания	300
5.4 Схема формирования сигналов POR и BOR	304
5.5 Схема слежения и переключения источника питания	307
5.6 Линейный регулятор напряжения питания батарейного домена.....	309
5.7 Основной линейный регулятор напряжения.....	311
5.8 Калируемый источник опорного напряжения	312
5.9 Калируемый источник опорного тока	313
6 Назначение выводов	314
Список использованных сокращений.....	319
Список использованных сокращений для регистров и битовых полей	320

1 Назначение

Микросхема 32х-битного микроконтроллера (далее микросхема МИК32, микросхема, микроконтроллер) предназначена для управления высокопроизводительными и многофункциональными электронными устройствами для интернета вещей, имеет развитию периферию и аппаратный криптографический блок, поддерживающий отечественные ГОСТ в области защиты информации.

Особенности (的独特性) продукции микросхемы МИК32:

- использование аппаратного криптографического блока для защиты информации в соответствии с отечественными ГОСТ;
- большинство устройств интернета вещей используют только программные криптографические средства на основе общепринятых алгоритмов защиты информации, которые имеют ряд недостатков;
- применение аппаратного криптографического блока для защиты информации в соответствии с Российскими ГОСТ – снижает вероятность успешности информационной атаки, увеличивает сложность взлома устройств и системы в целом. Таким образом, стоимость организации взлома устройств интернета вещей значительно возрастает.

Преимущества:

- применение открытой архитектуры RISC-V;
- высокая производительность;
- широчайший набор функций и возможностей;
- высокая степень интеграции компонент и блоков;
- высокая рабочая частота;
- большой запас встроенной памяти;
- отечественные разработка и производство.

2 Технические характеристики

Микросхема МИК32 имеет следующие характеристики:

- максимальная рабочая частота 32 МГц;
- объем встроенной памяти ОЗУ 16 КиБ;
- объем встроенной памяти EEPROM 8 КиБ;
- объем встроенной памяти OTP 256 бит;
- четыре источника тактирования;
- два домена питания:
 - 1) напряжение питания основного домена от 2,97 В до 3,63 В;
 - 2) напряжение питания батарейного домена от 2,5 В до 3,63 В;
 - 3) напряжение питания аналоговых блоков домена от 3,15 В до 3,45 В;
 - 4) динамический ток потребления основного домена не более 50 мА;
 - 5) динамический ток потребления батарейного домена не более 2 мА;
- диапазон рабочих температур от минус 45 °С до плюс 85 °С;
- повышенная предельная температура среды плюс 150 °С;
- пониженная предельная температура среды минус 60 °С;
- количество контактных площадок кристалла - 72;
- количество выводов корпуса – 64.

3 Состав МІК32

В состав микросхемы МІК32 (К1948ВК018, К1948ВК015) входят следующие блоки:

- ядро SCR1 компании Syntacore на базе открытой архитектуры RISC-V.
Реализована следующая конфигурация:
 - 1) RV32I - базовый целочисленный набор команд (минимальный необходимый набор команд, 32х-битное адресное пространство, 32 регистр общего назначения по 32 бита, все инструкции ширины 32 бита);
 - 2) расширение «M» - стандартное опциональное расширение для операций целочисленного умножения и деления;
 - 3) расширение «C» - стандартное опциональное расширение для сокращенных инструкций, дополнительный набор инструкций ширины 16 бит;
 - 4) встроенный интегрированный программируемый контроллер прерываний отключен;
 - 5) поддержка отладки через интерфейс JTAG IEEE Std 1149.1-2013 (совместима с RISC-V External Debug Support Spec Ver. 0.13.2), поддержка стандартных функций отладки: Reset, Halt / Resume / Step, доступ к GPR, CSR, MEM, HW Breakpoint / Watchpoint;
- трехстадийный конвейер;
- менеджер загрузки обеспечивает управление выбранным источником загрузки (SPI FLASH, ОЗУ, EEPROM);
- OSC32K – генератор с выводами для подключения внешнего часового кварцевого резонатора;
- OSC32M – генератор с выводами для подключения внешнего кварцевого резонатора для тактирования системы с частотой от 1 до 32 МГц;
- часы реального времени с функцией календаря с корректировкой для високосного года и программируемым будильником с возможностью генерации прерываний, и регистровым файлом 16x32 бит, для хранения пользовательских данных в батарейном домене;

- LDO_BU – регулятор напряжения батарейного домена с низким падением напряжения (входное напряжение 2,5 – 3,6 В, номинальное выходное напряжение 1,8 В, допустимая нагрузка до 5 мА);
- LDO_SYS – основной регулятор напряжения с низким падением напряжения (входное напряжение 3,0 – 3,6 В, номинальное выходное напряжение 1,8 В, допустимая нагрузка до 60 мА);
- POR и Bor (Power On Reset and BrownOut) – формирователь сигнала сброса батарейного домена при включении питания и контроль снижения напряжения питания ниже заданного;
- WakeUp - обеспечивает управление питанием и сбросом домена 0;
- HSI32M – встроенный генератор с частотой 32 МГц с возможностью корректировки частоты;
- LSI32K встроенных генератора с частотой 32 кГц с возможностью корректировки частоты;
- контроллер прерываний обеспечивает поддержку 32 маскируемых прерываний от периферийных устройств;
- блок управления питанием и тактированием обеспечивает управление питанием и управление тактовыми сигналами (установка коэффициентов делителей для шин, отключение тактирования отдельных устройств);
- сторожевой таймер шины, контролирует время отклика ведомых устройств нашине АHB (контроллер EEPROM, CRC32, ускоритель симметричной криптографии);
- три модуля «Таймер32», каждый из которых представляет собой настраиваемый 32х-разрядный счетчик с программируемым делителем и тремя режимами счета (прямой, обратный и двунаправленный). Timer32_1, Timer32_2 дополнительно имеют четыре независимых канала, способных работать в режиме захвата, сравнения или ШИМ;
- два монитора питания напряжением с программируемым порогом срабатывания, контролируемые напряжения: AVCC и VBAT;
- контроллер выводов, обеспечивает управление функциями мультиплексированных выводов;

- системный ТАР контроллер, обеспечивает доступ к отладочному тестовому интерфейсу, обеспечивает граничное сканирование выводов микросхемы;
- блок управления тестиированием, предназначен для управления тестиированием памяти и аналоговых блоков, а также обеспечивает доступ к регистрам периферийных модулей и ячейкам памяти без участия ядра;
- контроллер DMA, реализует передачу данных между ведомыми устройствами на коммутационной матрице без участия ядра, имеет четыре независимых канала, обеспечивающих работу в режимах память-память, периферия-периферия, память-периферия с программируемым уровнем приоритетов обработки запросов;
- блок однократно программируемой памяти размером 256 бит с возможностью программного чтения данных, предназначен для хранения поправочных коэффициентов и заводского номера изделия;
- блок электрически стираемого перепрограммируемого ПЗУ (EEPROM) размером 2048x38 с контроллером интерфейса АНВ со схемой коррекции ошибок (организация памяти 32 плюс 6) и возможностью исполнения программы;
- SPIFI контроллер, обеспечивает интерфейс к внешней флэш-памяти и поддерживает два режима работы: режим работы с периферией и режим работы с памятью. Контроллер обеспечивает выполнение инструкций из внешней памяти и имеет КЭШ-память размером 256x32. Контроллер поддерживает работу в обычном последовательном и в: 2-х и 4-х битном параллельном режимах и обеспечивает возможность исполнения программы;
- ОЗУ размером 4096x32 с поддержкой байтового доступа;
- CRC32 – модуль, предназначен для вычисления контрольной суммы (CRC) массива данных без использования вычислительных ресурсов центрального процессора на основе произвольного 32х-разрядного полинома;
- ускоритель симметричной криптографии, реализует три специализированных преобразования:
 - 1) AES128, FIPS 197 128-битный блочный шифр, со 128-битным ключом;
 - 2) «Магма», ГОСТ 34.12–2015 64х-битный блочный шифр;
 - 3) «Кузнецик», ГОСТ 34.12–2015 128-битный блочный шифр;

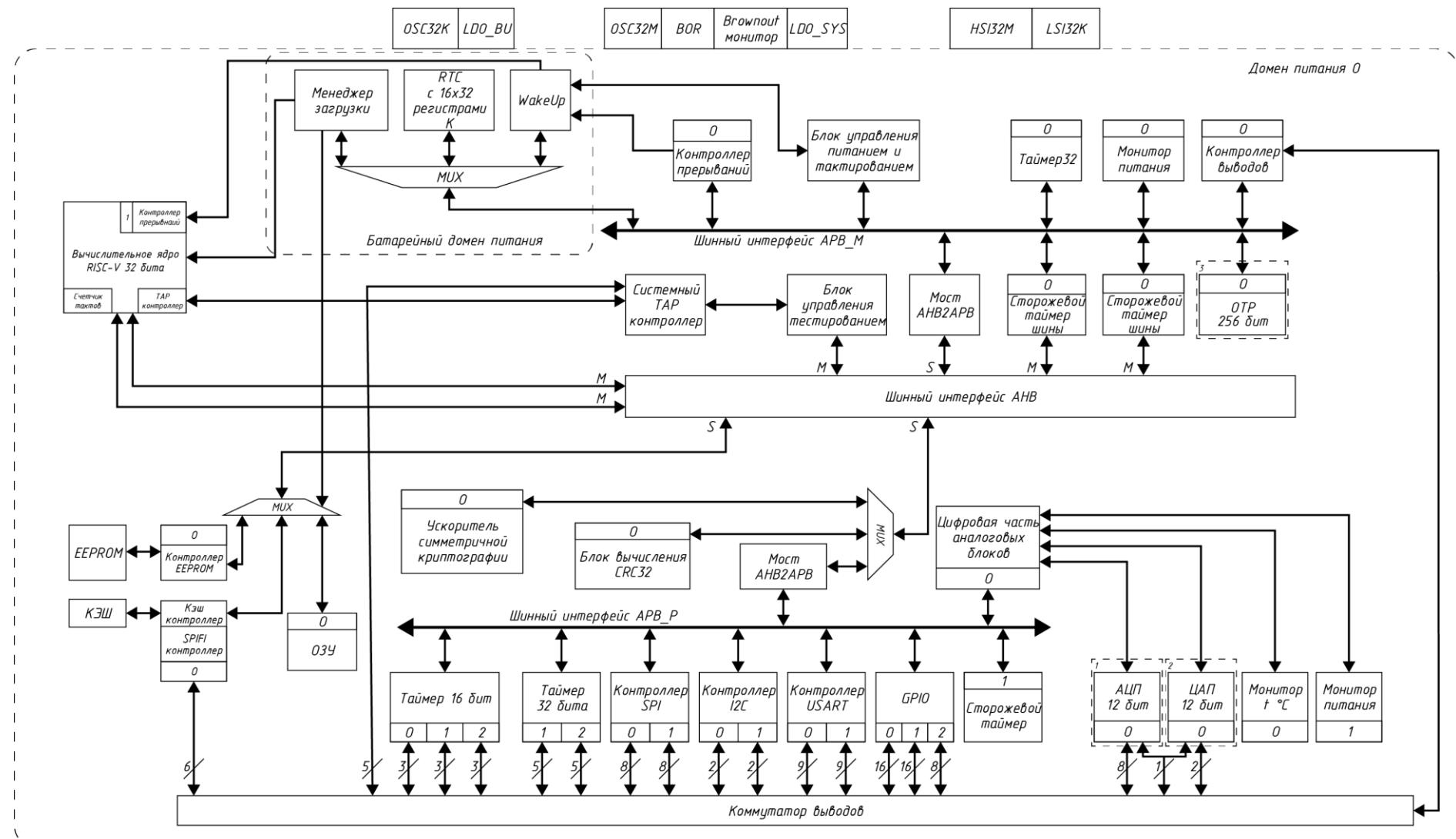
- 4) поддерживает режимы сцепления блоков (ECB, CBC, CTR);
- цифровая часть аналоговых блоков, обеспечивает доступ со стороны ядра к АЦП, ЦАП, мониторам температуры и напряжения;
 - три «Таймер16», каждый из которых представляет собой настраиваемый 16-ти разрядный счетчик с программируемым делителем (1, 2, 4, 8, 16, 32, 64, 128) с поддержкой режима ШИМ и настраиваемыми входными цифровыми фильтрами;
 - два модуля SPI для взаимодействия с различными периферийными устройствами, такими, как микросхемы памяти, датчики температуры, датчики давления, аналоговые преобразователи, часы реального времени, дисплеи и любые SD карты, поддерживающие последовательный режим обмена. Контроллер поддерживает работу в режиме ведущего и ведомого;
 - два модуля I2C, обеспечивающих последовательный интерфейс в соответствии со спецификацией на шину I2C от компании Philips/NXP. Контроллер I2C представляет собой двунаправленный, восьмибитный приемо–передатчик для взаимодействия с различными периферийными устройствами;
 - два модуля USART - синхронные асинхронные приемник–передатчики, который поддерживают широкий диапазон скоростей передач и форматов данных, задаваемых программно, с возможностью автоматически формировать бит четности и с возможностью обнаружения ошибок;
 - два 16-ти и один 8-ми разрядных порта ввода-вывода общего назначения (GPIO);
 - 12-ти разрядный АЦП, предназначен для преобразования входного аналогового сигнала в 12-разрядный цифровой код. Количество каналов 8, частота преобразования 0,8 МГц;
 - 12-ти разрядный ЦАП, предназначен для преобразования входного 12-разрядного цифрового кода в аналоговый сигнал. Количество каналов 2, частота преобразования 1 МГц;
 - монитор температуры с программируемыми порогами срабатывания;
 - монитор частоты, предназначен для контроля состояния источников тактирования во время работы микроконтроллера;

- сторожевой таймер - схема контроля над зависанием системы. Обеспечивает формирование прерывания и сброса, если не был выполнен программный сброс счетчика;
- матрица АНВ, обеспечивает коммутацию всех устройств с интерфейсами АНВ и АРВ.

Отличия исполнений от К1948ВК018 и К1948ВК015:

- К1948ВК01А8, К1948ВК01А5 – без использования блока АЦП;
- К1948ВК01Б8, К1948ВК01Б5 – без использования блока ЦАП;
- К1948ВК01В8, К1948ВК01В5 – без использования блока ОТР.

Структурная схема представлена на рисунке 1.



¹ Исполнение К1948ВК01А8, К1948ВК01А5 без использования блока АЦП

² Исполнение K1948VK01B8, K1948VK01B5 без использования блока ЦАП

³ Исполнение K1948BK01B8, K1948BK01B5 без использования блока OTP

Рисунок 1 – Структурная схема микросхемы МИК32

3.1 Коммутационная матрица AHB

Устройство предназначено для коммутации четырех ведущих и трех ведомых устройств с интерфейсом AHB Lite и обеспечивает доступ ведущих к устройствам на шинах APB_M и APB_P через соответствующие мосты. Коммутационная матрица обеспечивает одновременный доступ нескольких ведущих устройств к различным ведомым. Арбитраж нескольких ведущих к одному ведомому реализован на основе приоритетов с использованием алгоритма Round Robin. Обеспечивает организацию независимого адресного пространства для каждого из ведомых устройств. При обращении к недопустимому адресу (не входящему в адресное пространство мастера) формируется ответ с признаком ERROR. Возможные коммутации ведущих и ведомых устройств представлены в таблице.

Таблица 1 – Коммутация ведущих и ведомых устройств

Ведущий	Домен 0 и батарейный домен	Память (EEPROM/ ОЗУ/ SPIFI)	Периферийные устройства
Ядро (инструкции)	–	+	–
Ядро (данные)	+	+	+
DMA	–	+	+
Интерфейс тестирования	+	+	+

Примечание – коммутация отсутствует «–», наличие коммутации «+».

3.2 Карта памяти

Распределение адресного пространства между периферийными устройствами и модулями памяти представлено в таблице 2.

Таблица 2 – Распределение адресного пространства

Смещение	Устройство
0x00000000	Загрузочная область (16 КиБ)
0x00040000	Контроллер ПДП (DMA_CONFIG)
0x00050000	Устройства на шине APB_M
+0x0000	Блок управления питанием и тактированием (PM)

Смещение	Устройство
+0x0400	Контроллер прерываний (EPIC)
+0x0800	32-разрядный таймер 0 (TIMER32_0)
+0x0C00	Контроллер выводов (PAD_CONFIG)
+0x1000	Сторожевой таймер шины (WDT_BUS)
+0x1400	Контроллер однократно программируемой памяти (OTP)
+0x1800	Монитор напряжения VCC (PVD_CONTROL)
0x00060000	Устройства батарейного домена
+0x000	WakeUp (WU)
+0x400	Часы реального времени (RTC)
+0x800	Блок управления режимом загрузки (BOOT_MANAGER)
0x00070000	Устройства подсистемы памяти
+0x000	Контроллер внешней памяти (SPIFI_CONFIG)
+0x400	Контроллер ЭСППЗУ (EEPROM_REGS)
0x00080000	Устройства нашине AHB_P
+0x000	Ускоритель симметричной криптографии (CRYPTO)
+0x400	Блок вычисления контрольных сумм CRC32 (CRC)
0x00081000	Устройства нашине APB_P
+0x0000	Сторожевой таймер (WDT)
+0x0400	USART_0
+0x0800	USART_1
+0x0C00	16-разрядный таймер 0 (TIMER16_0)
+0x1000	16-разрядный таймер 1 (TIMER16_1)
+0x1400	16-разрядный таймер 2 (TIMER16_2)
+0x1800	32-разрядный таймер 1 (TIMER32_1)
+0x1C00	32-разрядный таймер 2 (TIMER32_2)
+0x2000	SPI_0
+0x2400	SPI_1
+0x2800	I2C_0
+0x2C00	I2C_1
+0x3000	GPIO_0
+0x3400	GPIO_1

Смещение	Устройство
+0x3800	GPIO_2
+0x3C00	Модуль формирования прерываний GPIO (GPIO_IRQ)
+0x4000	Блок управления аналоговой подсистемой (ANALOG_REG)
0x00490000	64-разрядный таймер ядра (SCR1_TIMER)
0x01000000	Внутренняя память EEPROM (8 КиБ)
0x02000000	Внутренняя память SRAM (16 КиБ)
0x80000000	Внешняя память (прямой доступ до 2 ГиБ)

3.3 Порядок загрузки

Микросхема MIK32 имеет три различных варианта старта, определяемых состоянием выводов Boot0 и Boot1 в соответствии с таблицей 3.

Таблица 3 – Варианты старта

Boot0	Boot1	Режим
0	0	Старт из встроенной памяти EEPROM
0	1	Старт из внешней памяти с использованием контроллера SPIFI
1	0	Старт из системного ОЗУ
1	1	Зарезервировано

Значение выводов Boot0 и Boot1 фиксируются при включении питания и при внешнем сбросе (вывод RST#). Программно возможно изменить способ загрузки путем записи необходимой конфигурации в регистр BOOT модуля управления режимом загрузки (BOOT_MANAGER). Обновленное значение режима загрузки будет использовано при следующих событиях (при условии разрешения соответствующего события в регистре SYS_MASK блока WAKEUP):

- выход из режима ожидания при срабатывании будильника;
- выход из режима ожидания при активном уровне внешнего вывода EXT_WU;
- сброс системного домена при снижении питания (индикация от LDO_SYS);
- сброс системного домена при снижении питания (индикация от схемы слежения);
- формирование сброса системного домена при появлении сигнала BOR.

В зависимости от выбранного режима загрузки происходит следующее изменение в адресном пространстве контроллера:

- загрузка из EEPROM: память EEPROM (8 КиБ) отображается в загрузочную область (0x00000000), но по-прежнему остается доступной также по адресу 0x01000000;
- загрузка из внешней памяти: контроллер SPIFI отображается в загрузочную область (0x00000000), но по-прежнему остается доступной также по адресу 0x80000000;
- загрузка из системной ОЗУ: блок ОЗУ отображается в загрузочную область (0x00000000), но по-прежнему остается доступной также по адресу 0x0200_0000.

3.4 Контроллер ПДП (DMA)

Специализированный контроллер прямого доступа к памяти (ПДП) реализует передачу данных между ведомыми устройствами на коммутационной матрице без участия микроконтроллера, поддерживая аппаратное адресное обращение к доменам МК32, участвующими в обмене данными.

Контроллер ПДП имеет 4 независимых каналов, обеспечивающих работу в режимах память-память, периферия-периферия, память-периферия.

Каждый из каналов ПДП имеет программируемый уровень приоритетов обработки запросов, задаваемый пользователем.

При работе с периферией контроллер ПДП, контролирует состояние внутренних буферов периферийных устройств с автоматической обработкой забросов и сигналов подтверждения. Структурная схема ПДП контроллера изображена на рисунке 2.

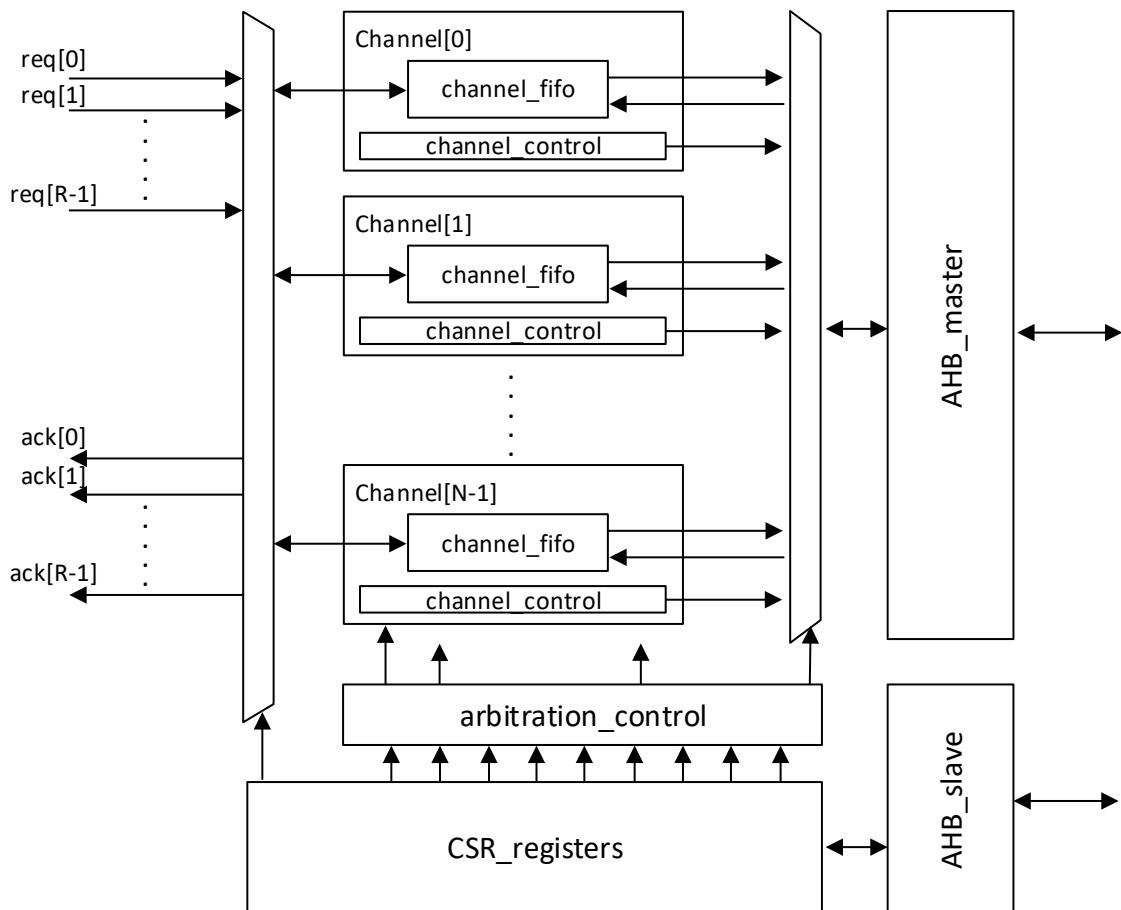


Рисунок 2 – Структурная схема ПДП контроллера

3.4.1 Доступ к шине

Контроллер управления шиной (AHB_master), при наличии данных в каналах ПДП, реализует непрерывный доступ к шине. При поддержке программных настроек управления сигнала подтверждения (нужен не для всех периферийных блоков) время выставлений транзакций на шину различается.

Сеанс передачи данных, начинается с чтения требуемых данных источника, заполнения буфера канала до указанного уровня, записи данных по адресу назначения, и далее в цикле пока не будет осуществлена требуемая передача данных.

Начало передачи данных происходит по сигналам запросов для каждого из канала, включая запросы на чтение и запись. При транзакциях памяти, внутренний сигнал запроса всегда активен.

Сигнал запроса на запись обрабатывается только тогда, когда в буфере канала есть данные. В случае конфликтов при формировании внутренних сигналов запросов со стороны конечных автоматов каналов, запросы обрабатываются в соответствии с установленным уровнем приоритета каждого из каналов. В случае совпадения уровней

программного приоритета, обрабатывается запрос в соответствии с уровнем статического приоритета, равному номеру канала (младший канал имеет наивысший приоритет).

Примеры доступа к шине показаны на временных диаграммах, на рисунках 3, 4 – Временная диаграмма длительности доступа к шине для двух каналов (длительность доступа к шине показана условно в 3 такта).

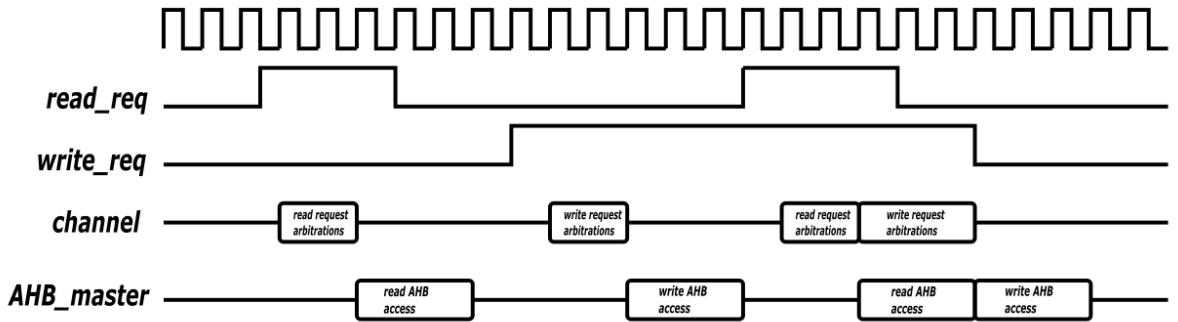


Рисунок 3 – Временная диаграмма длительности доступа к шине для одного канала

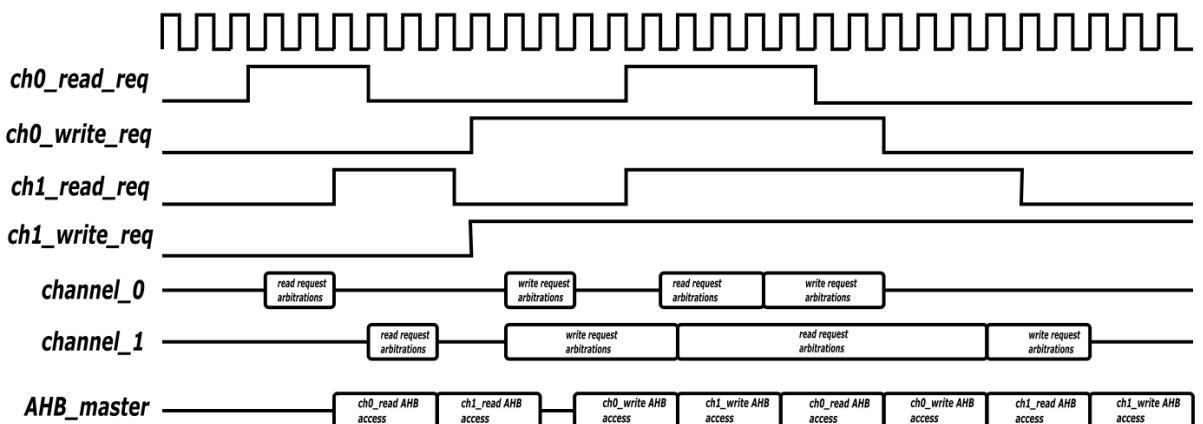


Рисунок 4 – Временная диаграмма длительности доступа к шине для двух каналов

3.4.2 Арбитраж

Арбитраж выбора канала происходит за два такта удержания линии запроса. Если одновременно поступают несколько запросов ПДП, то схема арбитража выбирает наиболее приоритетную линию запроса в соответствии с программной установкой каждого из каналов, которые поддерживают четыре уровня приоритетов:

- приоритетный (очень высокой);
- высокий;
- средний;
- низкий.

В случае активности одного канала, запрос не может быть обработан немедленно: после завершения обработки запроса добавляется один такт на выключение канала (или обработки сигнала подтверждения) и два такта для обработки нового запроса.

При работе нескольких каналов возможна параллельная обработка запросов, при этом, запросы с текущего рабочего канала при осуществлении арбитража маскируются для корректного выключения канала (или передачи сигнала подтверждения). Временные диаграммы работы изображены на рисунке 5.

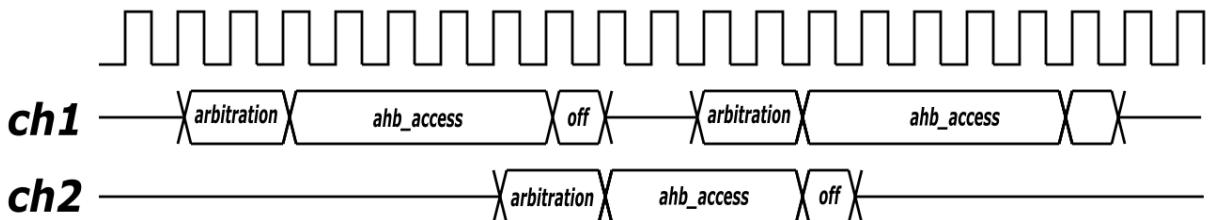


Рисунок 5 – Временные диаграммы работы нескольких каналов

3.4.3 Тактирование каналов

Для каждого из каналов предусмотрено автоматическое перекрытие тактирования на основе сигнала разрешения работы канала.

После завершения выполнения задания или обнаружения ошибки происходит автоматический сброс канала, и переход конечного автомата контроллера канала в состояние IDLE.

При этом каждый из каналов сохраняет состояния регистров, на котором была завершена его работа.

3.4.4 Выключение каналов

При передаче команды выключения каналу происходит перекрытие тактирования канала, после которого канал полностью сбрасывается.

В случае передачи команды выключения во время транзакции, мастер-интерфейс контроллера завершает начатую передачу блока данных в соответствии с установкой канала, при этом данные, передаваемые после выключения, не являются валидными. Временная диаграмма представлена на рисунке 6.

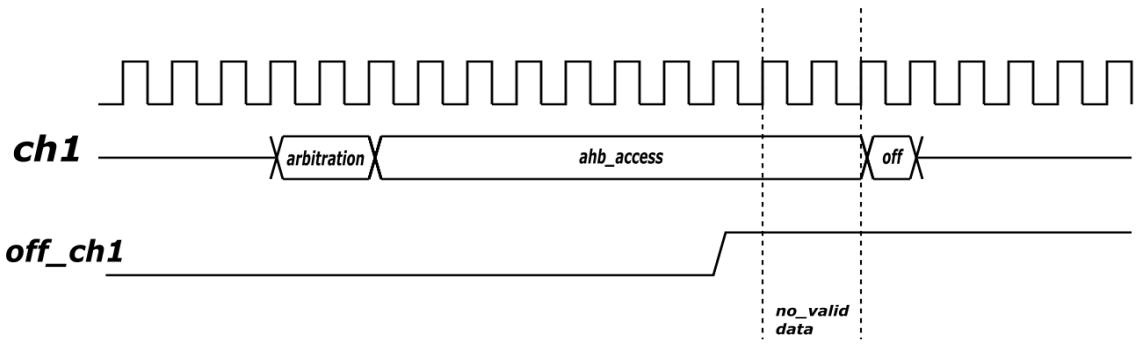


Рисунок 6 – Временная диаграмма передачи команды выключения во время транзакции, мастер-интерфейс контроллера

3.4.5 Блокные транзакции

Контроллер ПДП предусматривает возможность трансфера блочных транзакций, поддерживая смешанную передачу данных. К примеру, чтение байтового буфера и запись слов.

При этом, вводимые значения для трансфера данных должны находиться в строгом соотношении с вводимым значением длины пересылки и делением на блоки при чтении и записи. Примеры организации блочных транзакций показаны на временных диаграммах ниже (рисунки 7–9):

- чтение байтов, запись полуслова. Длина пересылки: 4 байта, размер блока чтения: 1 байт, размер блока записи 2 байта;
- чтение байтов, запись полуслова. Длина пересылки: 4 байта, размер блока чтения: 1 байт, размер блока записи 4 байта;
- чтение байтов, запись полуслова. Длина пересылки: 4 байта, размер блока чтения: 2 байта, размер блока записи 4 байта.

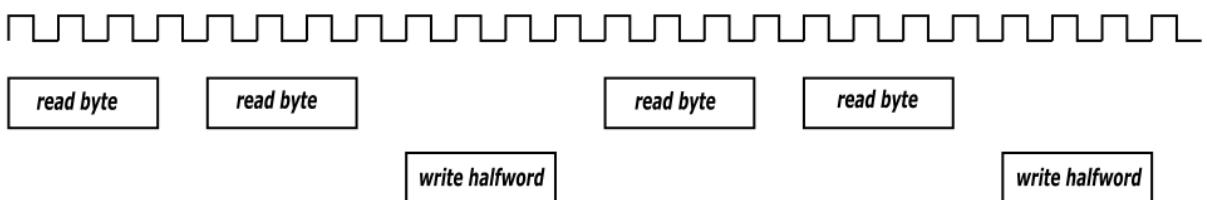


Рисунок 7 – Временная диаграмма организации блочных транзакций 1

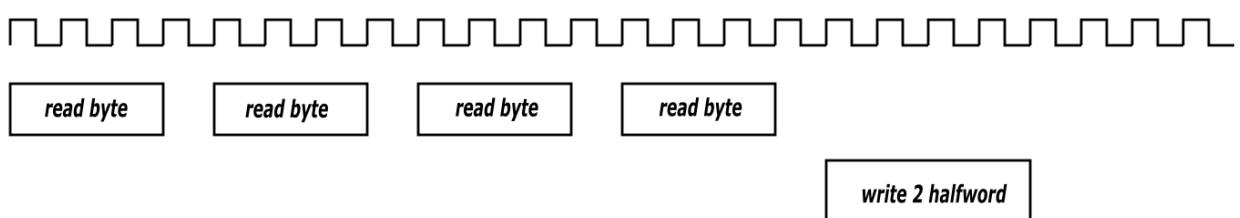


Рисунок 8 – Временная диаграмма организации блочных транзакций 2

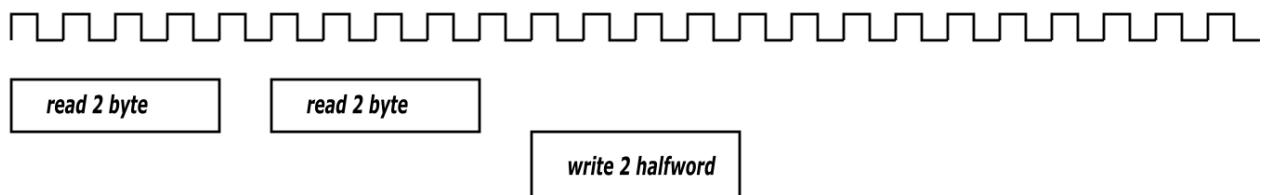


Рисунок 9 – Временная диаграмма организаций блочных транзакций 3

В случае ввода значений, не удовлетворяющих делению на блоки или ввода значений превышающих размер буфера канала возможно непредвиденное поведение контроллера.

3.4.6 Формирование прерываний

Контроллер ПДП поддерживает три типа маскируемых прерываний:

- прерывания каналов о завершении задания;
- глобальное прерывание о завершении всех заданий;
- прерывание об ошибке.

Прерывания каналов формируются в случае установки разрешения на прерывания, по умолчанию выключены.

Глобальное прерывание формируется при выполнении контроллером всех поставленных задач, по умолчанию включено.

Контроллер ПДП формирует ошибку на основе отклика об ошибке со стороны шины. Канал, обнаруживший ошибку, завершает свою задачу. Остальные каналы могут продолжать свою работу.

3.4.7 Передача в режиме периферии.

В режиме передачи с периферийными устройствами от устройства формируется запрос на транзакцию чтения или записи. Контроллер позволяет соотнести любой канал с периферийным устройством. Для настройки соответствия канала и линии от периферийного устройства используются поля READ_REQUEST, WRITE_REQUEST регистра CHx_CFG. Соответствие между значениями этих полей и линиями периферии представлено в таблице 4.

Таблица 4 – Линии запросов от периферии ПДП контроллера

Устройство	Значение READ_REQUEST / WRITE_REQUEST
USART_0	0
USART_1	1
CRYPTO	2
SPI_0	3
SPI_1	4
I2C_0	5
I2C_1	6
SPIFI	7
Timer32_1	8
Timer32_2	9
DAC0	10
DAC1	11
Timer32_0	12

При использовании запросов от 32х разрядного таймера необходимо устанавливать режим работы с подтверждением ($CHx_CFG.READ_ACK_EN = 1$ или $CHx_CFG.WRITE_ACK_EN = 1$). В таком режиме таймер формирует короткие импульсы при достижении условия переполнения, которые сбрасывают сигналы запроса от таймера. Запрос на входе DMA удерживается до окончания обработки текущего пакета. То есть, один импульс от таймера – соответствует обработке одного пакета.

Пример передачи с использованием подтверждений представлен ниже. Периферийное устройство должно снимать запрос на передачу после получения подтверждения от DMA.

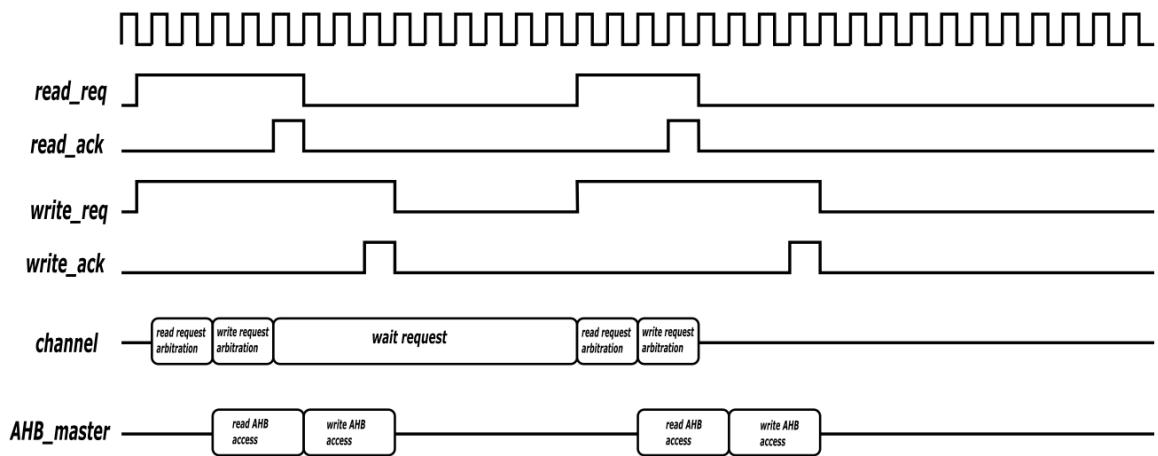


Рисунок 10 – Временная диаграмма длительности доступа к шине для одного канала с формированием сигнала снятия запроса

3.4.8 Регистры контроллера ПДП

Регистры контроллера включают регистры настройки параметров и управления каждым каналом и глобальный регистр статуса и прерываний. Регистры контроллера представлены в таблице 5. Доступ к регистрам осуществляется только словами по 32 бита.

Таблица 5 – Регистры ПДП контроллера

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CH1_DST	0x00	0x0	Регистр адреса назначения канала 1	RW
CH1_SRC	0x04	0x0	Регистр адреса источника канала 1	RW
CH1_LEN	0x08	0x0	Регистр размера передаваемых данных канала 1	RW
CH1_CFG	0x0C	0x0	Регистр управления и конфигурации канала 1	RW
CH2_DST	0x10	0x0	Регистр адреса назначения канала 2	RW
CH2_SRC	0x14	0x0	Регистр адреса источника канала 2	RW
CH2_LEN	0x18	0x0	Регистр размера передаваемых данных канала 2	RW
CH2_CFG	0x1C	0x0	Регистр управления и конфигурации канала 2	RW
CH3_DST	0x20	0x0	Регистр адреса назначения канала 3	RW
CH3_SRC	0x24	0x0	Регистр адреса источника канала 3	RW

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CH3_LEN	0x28	0x0	Регистр размера передаваемых данных канала 3	RW
CH3_CFG	0x2C	0x0	Регистр управления и конфигурации канала 3	RW
CH4_DST	0x30	0x0	Регистр адреса назначения канала 4	RW
CH4_SRC	0x34	0x0	Регистр адреса источника канала 4	RW
CH4_LEN	0x38	0x0	Регистр размера передаваемых данных канала 4	RW
CH4_CFG	0x3C	0x0	Регистр управления и конфигурации канала 4	RW
CONFIG_STATUS	0x40	0x0	Регистр прерываний и настройки контроллера	RW

Примечание – Функции отдельных полей регистра CONFIG_STATUS отличаются при выполнении операции чтения и записи.

3.4.1 Чтение статуса каналов

Для каждого из каналов, поддерживается возможность чтения текущих значений счёта (текущий адрес чтения/записи, количество переданных слов) и значений, которые были установлены при настройке канала (инициализирующий адрес чтения/записи, количество слов для передачи).

Статус каналов доступен по адресам конфигурационных регистров каналов CHx_DST при доступе на чтение. При этом каждый из адресов возвращает значение в соответствии с установленным значением INITIAL_VALUE регистра CONFIG_STATUS.

В случае установки бита INITIAL_VALUE будут возвращаться значения, установленные при настройке канала, в противном случае будут возвращаться значения текущего счёта.

3.4.1.1 Регистры CHx_DST

Регистр CHx_DST (CH1...CH4) – 32-х битный регистр адреса назначения данных каналов (таблица 6).

Таблица 6 – Обозначение, поля и описание регистра назначения данных

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DST	31:0	0x0	<p>Адрес назначения</p> <p>В режиме чтения исходных настроек (INITIAL_VALUE = 1) возвращает адрес назначения при настройке задачи.</p> <p>В режиме чтения текущего статуса (INITIAL_VALUE = 0) возвращает последнюю переданную подзадачу контроллера канала.</p> <p>В случае ошибки записи содержит указатель на текущий адрес мастер-интерфейса</p>	RW

3.4.1.2 Регистры CHx_SRC

Регистр CHx_SRC (CH1...CH4) – 32-х битный регистр адреса источника данных каналов (таблица 7).

Таблица 7 – Обозначение, поля и описание регистра источника данных

Обозначение	Разряды	Начальное значение	Назначение	Доступ
SRC	31:0	0x0	<p>Адрес источника</p> <p>В режиме чтения исходных настроек (INITIAL_VALUE = 1) возвращает адрес источника при настройке задачи.</p> <p>В режиме чтения текущего статуса (INITIAL_VALUE = 0) возвращает текущую подзадачу контроллера канала. Указатель на адрес блока, который обрабатывается вычисляется как:</p> <p>Current_ch_read_task - $2^{\text{READ_BURST_SIZE}}$</p> <p>В случае ошибки записи содержит указатель на текущий адрес мастер-интерфейса, вернувшего HRESP=1</p>	RW

3.4.1.3 Регистры CHx_LEN

Регистр CHx_LEN (CH1...CH4) – 32-х битный регистр количества байт пересылки (таблица 8). Количество пересылаемых данных, определяется как LEN + 1.

Таблица 8 – Обозначение, поля и описание регистра количества байт пересылки

Обозначение	Разряды	Начальное значение	Назначение	Доступ
LEN	31:0	0x0	<p>Количество байт пересылки рассчитывается как LEN +1.</p> <p>При принудительной остановке или в случае ошибки передачи данных поле LEN может быть использовано для оценки количества переданных данных. Счётчик обновляется при начале исполнения задачи на запись, до ее фактического исполнения.</p> <p>В режиме чтения текущего статуса (INITIAL_VALUE = 0) Если остановка канала произошла при операции чтении (STATE = 0b01), то:</p> <ul style="list-style-type: none"> – Количество записанных байт равно LEN; – Количество прочитанных байт больше или равно LEN и меньше или равно LEN+2^{WRITE_BURST_SIZE}. <p>Если остановка канала произошла при операции записи (STATE = 0b10), то:</p> <ul style="list-style-type: none"> – Количество прочитанных байт равно LEN; – Количество записанных байт больше или равно LEN и меньше или равно LEN+2^{WRITE_BURST_SIZE}. <p>Состояние статуса контроллера канала определяется чтением поля STATE регистра CHx_CFG при INITIAL_VALUE = 0.</p>	RW

3.4.1.4 Регистры CHx_CFG

Регистр CHx_CFG (CH1...CH4) предназначен для управления и получения статуса канала. Обозначение, поля и описание регистра отображены в таблице 9.

Примечание: Запись в регистр CHx_CFG останавливает работу канала.

Таблица 9 – Обозначение, поля и описание регистра управления каналами при записи и при чтении с INITIAL_VALUE = 1

Обозначение	Разряды	Начальное значение	Назначение	Доступ
IRQ_EN	27	0x0	Разрешение формирования прерывания по завершении работы канала: 0 – прерывание не формируется; 1 – прерывание формируется	RW
WRITE_ACK_EN	26	0x0	Разрешение работы логики с откликом для адресата назначения: 0 – запрещено; 1 – разрешено	RW
READ_ACK_EN	25	0x0	Разрешение работы логики с откликом для адресата источника: 0 – запрещено; 1 – разрешено	RW
WRITE_REQUEST	24:21	0x0	Выбор периферийной линии назначения	RW
READ_REQUEST	20:17	0x0	Выбор периферийной линии источника	RW
WRITE_BURST_SIZE	16:14	0x0	Количество байт в пакете Определяется как 2WRITE_BURST_SIZE * должно быть кратно WRITE_SIZE	RW
READ_BURST_SIZE	13:11	0x0	Количество байт в пакете Определяется как 2READ_BURST_SIZE * должно быть кратно READ_SIZE	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
WRITE_SIZE	10:9	0x0	Разрядность адреса назначения: 0b00 – байт; 0b01 – полуслово; 0b10 – слово; 0b11 - зарезервировано * должно быть кратно LEN	RW
READ_SIZE	8:7	0x0	Разрядность адреса источника: «0b00» – байт; «0b01» – полуслово; «0b10» – слово; «0b11» - зарезервировано * должно быть кратно LEN	RW
WRITE_INCREMENT	6	0x0	Инкремент адреса назначения: 0 – нет инкремента; 1 – есть инкремент	RW
READ_INCREMENT	5	0x0	Инкремент адреса источника: 0 – нет инкремента; 1 – есть инкремент	RW
WRITE_MODE	4	0x0	Режим адреса назначения: 0 – периферия; 1 – память	RW
READ_MODE	3	0x0	Режим адреса источника: 0 – периферия; 1 – память	RW
PRIOR	2:1	0x0	Приоритет канала: 0b00 – низкий; 0b01 – средний; 0b10 – высокий; 0b11 – приоритетный (очень высокий)	RW
ENABLE	0	0x0	Разрешение работы канала: 1 – инициализация работы канала; 0 – принудительная остановка	RW

Примечание – Запись в регистр CHx_CFG останавливает работу канала.

При чтении CHx_CFG при CURRENT_VALUE = 0 будет получен текущий статус канала.

Таблица 10 – Обозначение, поля и описание регистра управления каналами при чтении с INITIAL_VALUE = 0

Обозначение	Разряды	Начальное значение	Назначение	Доступ
STATE	3:2	0x0	Состояние статуса контроллера канала 0b01 – канал в состоянии чтения; 0b10 – канала в состоянии записи.	RO
BUS_ERROR_WRITE	1	0x0	Состояние ошибки при чтении 0 – ошибки не было; 1 – была зафиксирована ошибка шины при записи	RO
BUS_ERROR_READ	0	0x0	Состояние ошибки при записи 0 – ошибки не было; 1 – была зафиксирована ошибка шины при чтении	RO
Примечание – Запись в регистр при CURRENT_VALUE = 0 осуществляется в соответствии с битовыми полями таблицы 9.				

3.4.1.5 Регистр CONFIG_STATUS

Регистр CONFIG_STATUS обеспечивает управление прерываниями каналов и контроллера в целом. Функции отдельных полей отличаются при выполнении операции чтения и записи регистра. Обозначение, поля и описание регистра при записи представлены в таблице 11, при чтении представлены в таблице 12.

Таблица 11 – Обозначение, поля и описание регистра CONFIG_STATUS при записи

Обозначение	Разряды	Начальное значение	Назначение	Доступ
INITIAL_VALUE	8	0x0	Разрешение чтения значений при настройке каналов: 0 – запрещено (текущий статус каналов); 1 – разрешено (значения при настройке)	WO
ERROR_IRQ_ENA	7	0x0	Разрешение формирования прерывания при ошибке: 1 – разрешено; 0 – запрещено.	WO

Обозначение	Разряды	Начальное значение	Назначение	Доступ
GLOBAL_IRQ_ENA	6	0x0	Разрешение формирования глобального прерывания: 1 – разрешено; 0 – запрещено.	WO
CLEAR_ERROR_IRQ	5	0x0	Очистка прерывания ошибки: 1 – снятие запроса на прерывание.	WO
CLEAR_GLOBAL_IRQ	4	0x0	Очистка глобального прерывания: 1 – снятие запроса на прерывание.	WO
CLEAR_LOCAL_IRQ	3:0	0x0	Очистка локального прерывания: 1 – снятие запроса на прерывание. Номер бита соответствует номеру канала.	WO

Примечание – Доступ на установку регистров разрешения прерываний возможен только при отсутствии установки на снятие запроса линии прерывания.

Таблица 12 – Обозначение, поля и описание регистра CONFIG_STATUS при чтении

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CHANNEL_BUS_ERROR	11:8	0x0	Статус состояния каналов при ошибках на шине: 1 - есть ошибка; 0 - нет ошибки. Бит 8 – канал 0; Бит 9 – канал 1; Бит 10 – канал 2; Бит 11 – канал 3.	RO
CHANNEL_IRQ	7:4	0x0	Статус прерываний: 1 - есть прерывания; 0 - нет прерываний.	RO
CHANNEL_READY	3:0	0x0	Статус состояния каналов: 1 - готов к работе; 0 – занят. Номер бита соответствует номеру канала.	RO

Примечание – При передаче каналу нового задания, значение CHANNEL_BUS_ERROR сбрасывается в 0.

3.5 Блок управления тестированием (TCB)

Блок управления тестированием (Test Control Block, далее TCB) обеспечивает прием команд по интерфейсу JTAG и преобразованием их в транзакции на системные шину и в команды управления тестовыми компонентами микроконтроллера.

3.5.1 Общая структура

Блок TCB состоит из двух модулей:

- интерфейс к TAP контроллеру (TAP2FSM);
- отладочный интерфейс системной шины (AHB_FSM).

Структурная схема блока TCB представлена на рисунке 11.

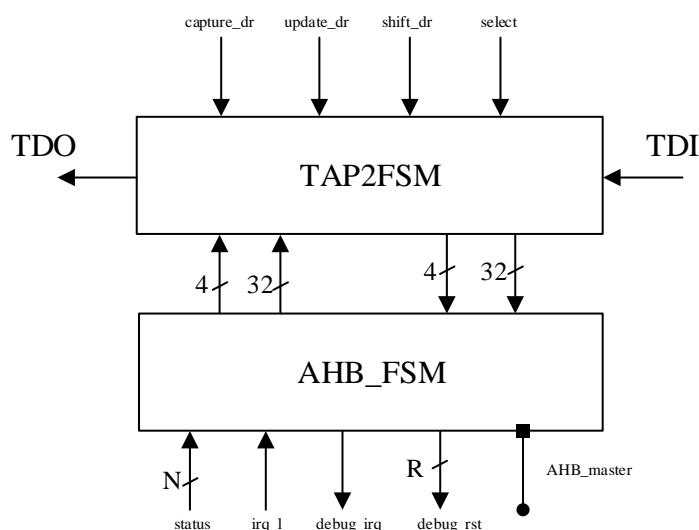


Рисунок 11 – Структурная схема блока TCB

3.5.2 Интерфейс к ТАР контроллеру

Интерфейс осуществляет преобразование последовательных команд в формат инструкций отладочного интерфейса системной шины.

Интерфейс реализуется на основе сдвигового регистра. После обновления данных в сдвиговом регистре интерфейс передаёт в отладочный интерфейс системной шины 32-х разрядное слово данных с заголовком (исполняемой инструкцией). На основе данных заголовка происходит требуемая обработка данных.

Обновление значений регистра и инициализация команды для отладочного интерфейса системной шины происходит по сигналу update_dr.

По сигналу capture_dr происходит обновление регистра данными и статусом предыдущей выполненной операции. При этом по сигналу shift_dr регистр возвращает ТАР контроллеру полученные данные.

3.5.3 Отладочный интерфейс системной шины

Отладочный интерфейс осуществляет исполнение инструкций, полученных от TAP2FSM и передачу команд на шину АНВ.

Поддерживается следующий набор инструкций:

- установка адреса с последующим инкрементом или без;
- запись данных по установленному адресу;
- чтение по передаваемому адресу;
- инвертирование линии управления;
- чтение линии статуса тестируемой системы.

3.5.4 Формат инструкций

Формат инструкций, состоящей из 36 бит представлен на рисунке 12.

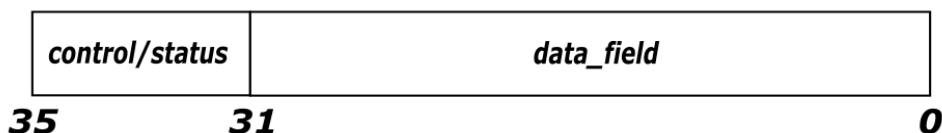


Рисунок 12 – Структура инструкции

Формат поля control/status инструкции при записи регистра представлен в таблице 13.

Таблица 13 – Формат поля control/status при записи регистра

Формат команды	Команда	Описание	Примечание
0bxx00	0b0000	Запись байта	Команды записи, требуют предварительно установленного адреса
	0b0100	Запись полуслова	
	0b1000	Запись слова	
	0b1100	Зарезервировано	
0bxx01	0b0001	Чтение байта	Команды чтения, возвращают данные по адресу поля data_field
	0b0101	Чтение полуслова	
	0b1001	Чтение слова	
	0b1101	Зарезервировано	
0bxx10	0b0010	Установка адреса с последующим инкрементированием	Команды установки адреса
	0b0110	Установка адреса без инкрементирования	

Формат команды	Команда	Описание	Примечание
	0b1010	Зарезервировано	
	0b1110	Зарезервировано	
0bxx11	0b0011	Зарезервировано	Управляющие команды
	0b0111	Инвертирование линии управления	
	0b1011	Системный сброс контроллера	
	0b1111	Пустая команда (NOP)	

Формат поля control/status инструкции при чтении регистра представлен в таблице 14.

Таблица 14 – Формат поля control/status при чтении регистра

Формат статуса	Команда	Описание	Значение
0bxxx0	0bxx10	Отклик об ошибке со стороны шины (HRESP)	Предыдущая операция не выполнена
	0bx1x0	Ожидание отклика о готовности	
	0b1xx0	Зарезервировано	
0bxxx1	0bxx11	Входная линия статуса	Предыдущая операция выполнена
	0b0101	Зарезервировано	
	0b1101	Зарезервировано	

3.5.4.1 Команды установки адреса

При выполнении команды установки адреса отладочный интерфейс устанавливает значение адреса, соответствующие полю data_field. Последующие данные команд записи будут записываться на основе установленного адреса, до его смены.

В случае установки адреса на инкремент, после каждой записи адрес будет увеличиваться на значение, соответствующее разрядности записываемого слова (байт – 1, полуслово – 2, слово – 4).

3.5.4.2 Команды записи данных

При выполнении команды записи данных отладочный интерфейс передаёт на шину команду записи данных по установленному ранее адресу. При этом передаваемые

данные в сдвиговый регистр должны быть выравнены по левому краю. Выравнивание данных для шины происходит автоматически по установленному адресу.

3.5.4.3 Команды чтения данных

При выполнении команды чтения данных отладочный интерфейс передаёт на шину команду чтения по адресу `data_field`, отладочный интерфейс всегда возвращает 32-х битное значение, полученное со стороны шины, при этом на шину выставляется значение смещения, переданное инструкцией.

3.5.4.4 Управляющие команды

Инвертирование линии управления происходит по значениям поля `data_field`. Изначально линии управления находятся в высоком состоянии.

В микроконтроллере используются 12 линий управления со стороны модуля TCB (Таблица 15).

Таблица 15 – Назначение линий управления модуля TCB

Индекс линии управления	Назначение
2:0	Управления модулем самотестирования памяти (выбор алгоритма)
3	Управления модулем самотестирования памяти (запуск теста)
4	Управления модулем самотестирования памяти (разрешение тестирование КЭШ памяти SPIFI контроллера)
5	Управления модулем самотестирования памяти (разрешение тестирование ОЗУ)
6	Сброс ОЗУ, контроллера EEPROM, контроллера SPIFI (активный уровень 0)
7	Сброс домена периферийных устройств (активный уровень 0)
8	Зарезервировано
9	Сброс контроллера DMA (активный уровень 0)
10	Сброс процессора <code>cpu_RST_n</code> (активный уровень 0)
11	Сброс процессора <code>rst_n</code> (активный уровень 0)

Команда системного сброса переводит отладочный интерфейс в IDLE состояние, применяется в случае зависания ответного сигнала `ready` со стороны шины.

3.5.4.5 Отклик об ошибке

Отклик об ошибке формируется при получении со стороны шины сигнала HRESP, при этом при передаче отладочному интерфейсу новой команды флаг состояния отклика сбрасывается.

3.5.4.6 Отклик о зависании

В случае отсутствия получения сигнала ready со стороны шины отладочный интерфейс формирует отклик об отсутствии ready при этом, пока сигнал ready не придёт со стороны шины контроллер не изменит своего состояния. Для сброса состояния используется команда системного сброса.

3.5.5 Последовательность выполнения команд

При выполнении последовательности команд читается и анализируется статус предыдущей команды. На основе анализа статуса происходит формирование следующей команды. Если статус не верен происходит возврат к выполнению предыдущей команды. При этом первой командой передаётся пустая инструкция (NOP).

Условный пример записи четырёх слов с возвращением неверного статуса показан на рисунке 13.

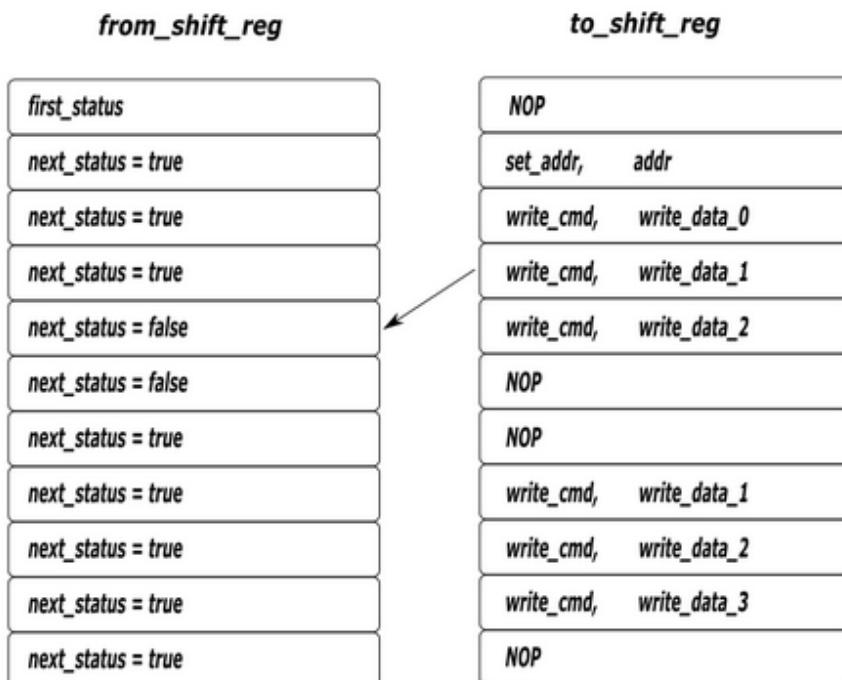


Рисунок 13 – Условный пример записи четырёх слов с возвращением неверного статуса

Условный пример чтения четырёх слов с возвращением неверного статуса показан на рисунке 14.

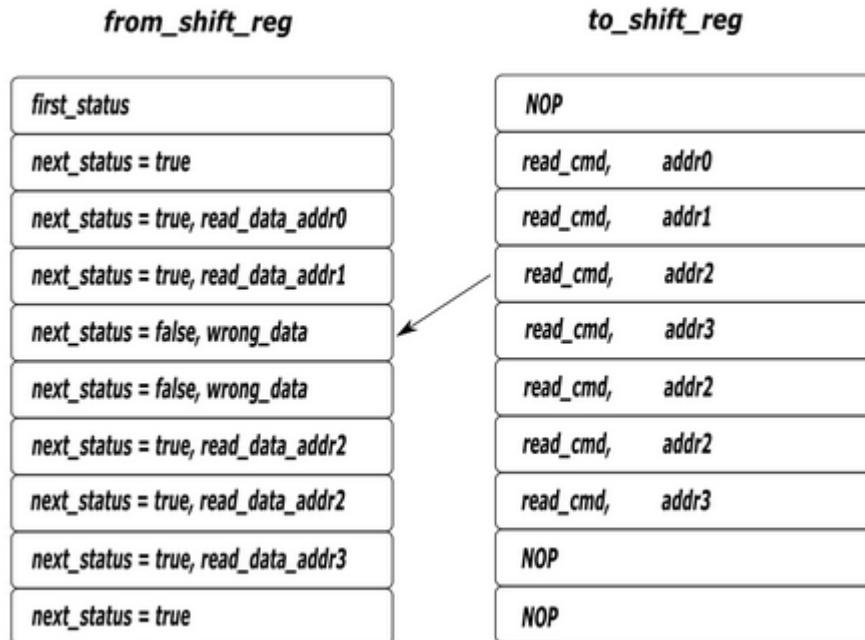


Рисунок 14 – Условный пример чтения четырёх слов с возвращением неверного статуса

3.6 Модуль USART

Универсальный синхронный асинхронный приемник-передатчик (USART) поддерживает полнодуплексный обмен данными с внешним оборудованием, а также работает в режимах синхронной односторонней связи и полуудуплексной однопроводной связи. Он также поддерживает операции с модемом.

Основные характеристики:

- программируемая длина слова данных (7, 8 или 9 бит);
- программируемый порядок данных с первым сдвигом MSB или LSB;
- настраиваемые стоп биты (1 или 2 стоп бита);
- синхронный режим и тактовый выход для синхронной связи;
- возможность работы в режиме однопроводной полуудуплексной связи;
- непрерывная связь с использованием прямого доступа к памяти;
- отдельные биты разрешения для передатчика и приемника;
- отдельный контроль полярности сигнала для передачи и приема;
- возможность смены контактов Tx / Rx;
- аппаратное управление потоком при помощи сигналов CTS и RTS;
- наличие модемных сигналов;

- контроль четности передает бит четности и проверяет четность полученного байта данных.

Обобщенная структурная схема приемо-передатчика USART представлена на рисунке 15.

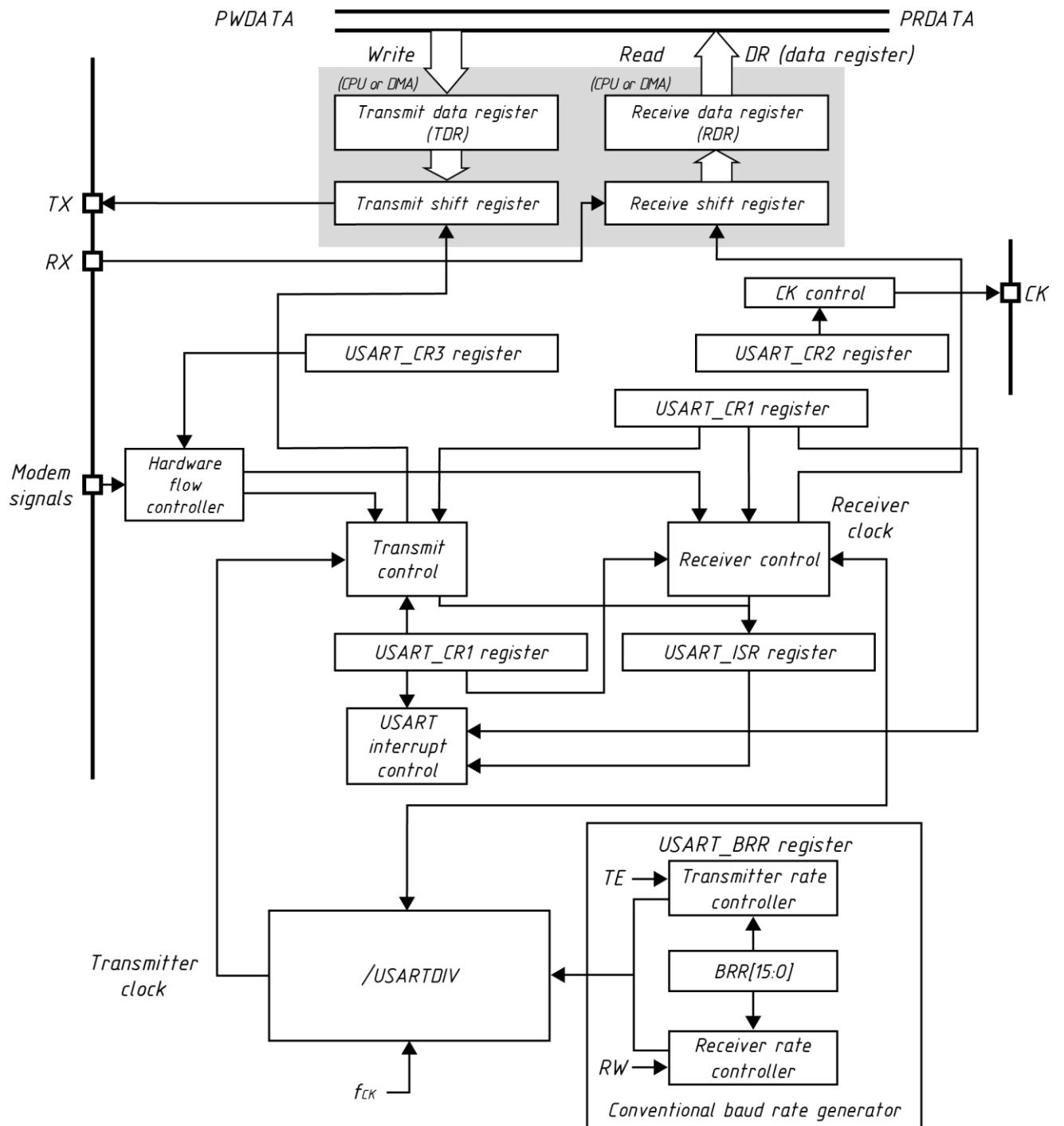


Рисунок 15 – Обобщенная структурная схема USART

3.6.1 Функциональное описание

Последовательные данные передаются и принимаются через контакты TX и RX в обычном режиме USART.

Кадры включают в себя:

- стартовый бит;
- биты данных (7, 8 или 9);
- бит четности, если выставлен соответствующий управляющий бит;
- 1 или 2 стоп бита, указывающих, что кадр завершен.

На рисунке 16 представлен пример кадра.

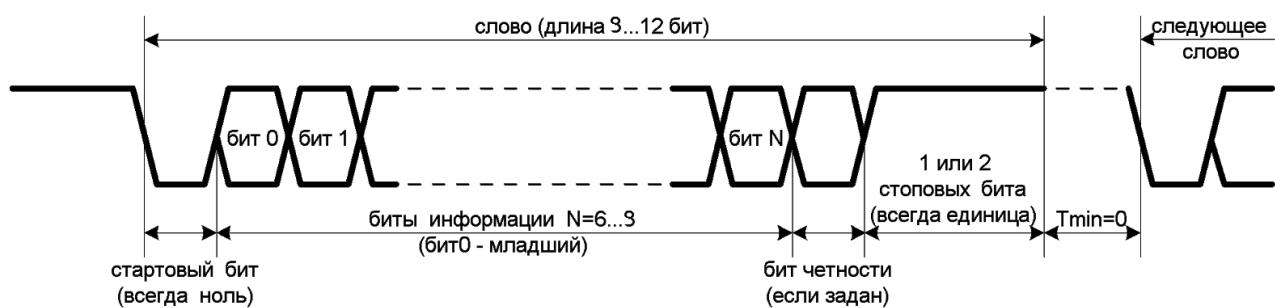


Рисунок 16 – Формат кадра

Длина кадра зависит от количества бит данных и наличия бита четности. При этом на порядок передачи бита четности не влияет поле MSBFIRST, он всегда передается перед стоп-битом. В таблице 16 приведены все возможные варианты кадров.

Таблица 16 – Возможные варианты кадров

M биты	PCE биты	Кадры USART
00	0	SB данные 8 бит STB
00	1	SB данные 7 бит PB STB
01	0	SB данные 9 бит STB
01	1	SB данные 8 бит PB STB
10	0	SB данные 7 бит STB
10	1	SB данные 6 бит PB STB

3.6.2 Синхронный режим

Выход CK используется для передачи данных в синхронном режиме. CK – выходной сигнал, он выводит тактовые импульсы данных передатчика для синхронной передачи, соответствующей режиму ведущего SPI (нет тактовых импульсов на старт и стоп-битах, а также программная опция для отправки тактового импульса на последний бит данных). Параллельно данные могут быть получены синхронно на RX. Это может

использоваться для управления периферийными устройствами, которые имеют сдвиговые регистры. Фаза и полярность тактового сигнала задаются программно.

Диаграммы работы блока с разными параметрами тактового сигнала и с разной очередностью бит внутри байта представлены на рисунке 17.

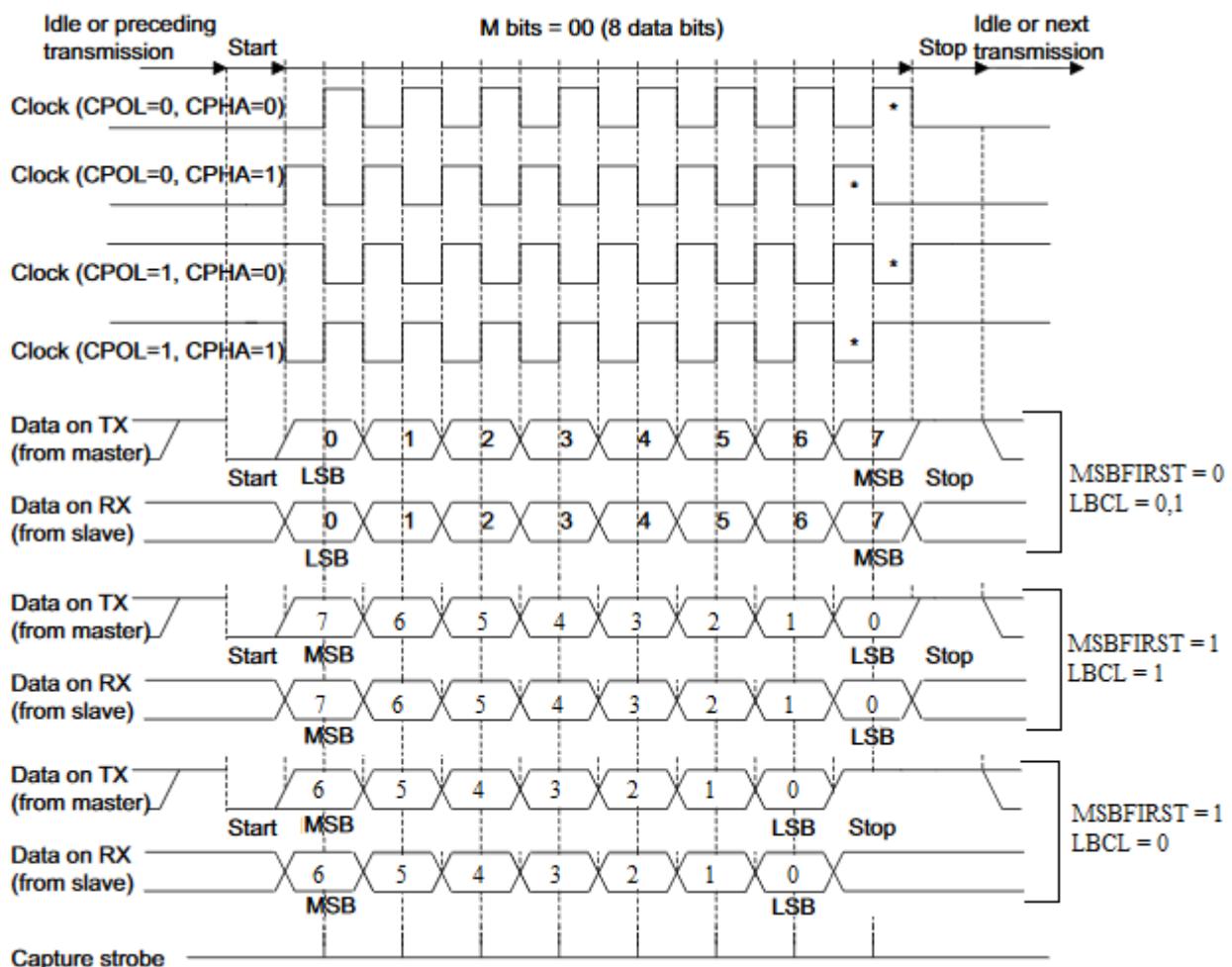


Рисунок 17 – Диаграммы работы в синхронном режиме при различных конфигурациях

3.6.3 Управление потоком данных

Следующие выводы необходимы в режиме управления потоком:

- CTS: Clear To Send блокирует передачу данных в конце текущей передачи (когда высокий уровень);
- RTS: Request To Send указывает, что USART готов к приему данных (когда низкий уровень).

Сигнал RTS – выходной, его формирование связано с состоянием приемного регистра. Состояние после сброса – 0. Сигнал переходит в 1 при наличии непрочитанных данных в приемном регистре.

Сигнал CTS – входной, он управляет передатчиком. Значение 1 запрещает передачу данных. В случае перехода сигнала в 1 во время передачи посылки, посылка будет передана, и только потом линия TX будет установлена в режим ожидания. Как только сигнал устанавливается в 0, восстанавливается передача данных. Состояние сигнала CTS доступно на чтение процессором. Примеры работы сигнала CTS представлены на рисунках 18 и 19.

В режиме loop-back mode RTS замыкается на CTS внутри блока, в обход соответствующих выводов микроконтроллера.

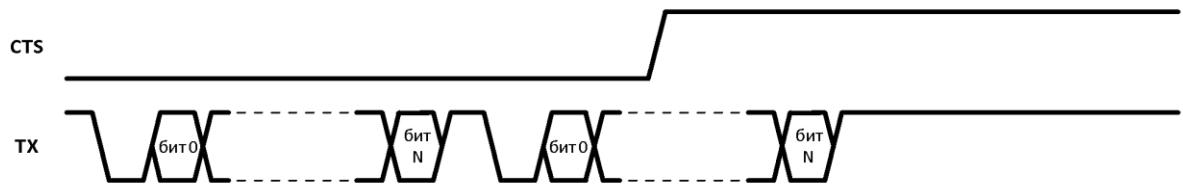


Рисунок 18 – Пример срабатывания сигнала, запрещающего передачу (CTS=1)

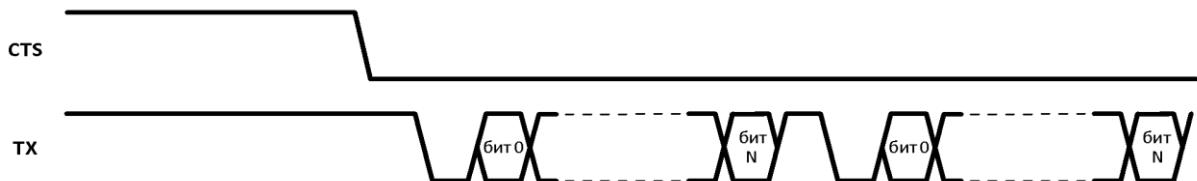


Рисунок 19 – Пример перехода сигнала, запрещающего передачу, в неактивное состояние (CTS=0)

3.6.4 Режим Break Line

Функция работает как на передачу, так и на прием. Управление функцией при передаче производится через бит SBKRQ. Активное состояние бита – 1. Если в момент записи 1 в разряд SBKRQ происходила передача данных, то посылка будет передана и только потом линия TX будет установлена в режим break (TX = 0). Пример работы режима break в передатчике представлен на рисунке 20.



Рисунок 20 – Пример работы режима break

Во время приема производится обнаружение состояния Break следующим образом. Если на линии RX была обнаружена посылка (стартовый бит плюс биты данных плюс бит четности (если установлен) плюс стоп биты) заполненная нулями, то в приемный регистр записывается этот нулевой символ, и взводится флаг обнаружения Break. Этот флаг доступен процессору на чтение. Запись в приемный регистр восстанавливается только после того, как на линии RX были обнаружены два стоп бита. Пример обнаружения режима break в приемнике представлен на рисунке 21.

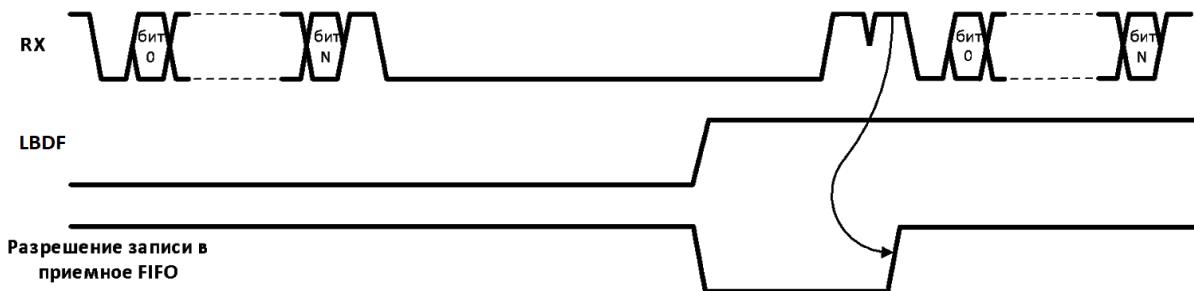


Рисунок 21 – Пример обнаружения режима break

3.6.5 Карта адресов

Регистры приемо-передатчика USART приведены в таблице 17.

Таблица 17 – Регистры приемо-передатчика USART

Обозначение	Смещение	Начальное значение	Назначение	доступ
CONTROL1	0x00	0x0	Регистр управления 1	RW
CONTROL2	0x04	0x0	Регистр управления 2	RW
CONTROL3	0x08	0x0	Регистр управления 3	RW
DIVIDER	0x0C	0x0	Регистр настройки делителя	RW
FLAGS	0x1C	0x0	Регистр прерываний	RW
RXDATA	0x24	0x0	Регистр принятых данных	RW
TXDATA	0x28	0x0	Регистр передаваемых данных	RW
MODEM	0x2C	0x0	Регистр управления модемом	RW

Обозначение, разряды, начальное значение, назначение и доступ для регистра управления 1 указаны в таблице 18.

Таблица 18 – Регистр управления 1 CONTROL1

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:29		Зарезервировано	
M1	28	0x0	Совместно с битом 12 (M0) управляет длиной кадра: M[1:0] = 0b00: 8 бит данных; M[1:0] = 0b01: 9 бит данных; M[1:0] = 0b10: 7 бит данных. Этот бит может быть изменен только при остановке работы (UE=0)	RW
Зарезервировано	27:13		Зарезервировано	
M0	12	0x0	Совместно с битом 28 (M1) управляет длиной кадра: M[1:0] = 0b00: 8 бит данных; M[1:0] = 0b01: 9 бит данных; M[1:0] = 0b10: 7 бит данных. Этот бит может быть изменен только при остановке работы (UE=0)	RW
Зарезервировано	11		Зарезервировано	
PCE	10	0x0	Контроль четности: 0 – контроль четности выключен; 1 – контроль четности включен. Этот бит может быть изменен только при остановке работы (UE=0)	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
PS	9	0x0	Выбор способа формирования бита четности: 0 – бит четности; 1 – бит нечетности. Этот бит может быть изменен только при остановке работы (UE=0)	RW
PEIE	8	0x0	Управление прерыванием при ошибке в принятом бите четности: 0 – прерывание по ошибке в бите четности выключено; 1 – прерывание по ошибке в бите четности включено	RW
TXEIE	7	0x0	Управление прерыванием при передаче данных в момент записи данных в сдвиговый регистр: 0 – прерывание выключено; 1 – прерывание включено	RW
TCIE	6	0x0	Управление прерыванием при успешной передаче данных: 0 – прерывание при передаче данных выключено; 1 – прерывание при передаче данных включено	RW
RXNEIE	5	0x0	Управление прерыванием при успешном приеме данных или перезаписи полученных данных: 0 – прерывание выключено; 1 – прерывание включено	RW
IDLEIE	4	0x0	Управление прерыванием при отсутствии входных транзакций: 0 – прерывание выключено; 1 – прерывание включено	RW
TE	3	0x0	Управление передатчиком: 0 – передатчик выключен; 1 – передатчик включен	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
RE	2	0x0	Управление приемником: 0 – приемник выключен; 1 – приемник включен	RW
Зарезервировано	1	Зарезервировано		
UE	0	0x0	Ресет USART. Отменяет все текущие операции. Не очищает регистры конфигурации, но сбрасывает флаги 0 – USART выключен 1 – USART включен	RW

Обозначение, разряды, начальное значение, назначение и доступ для регистра управления 2 указаны таблице 19.

Таблица 19 – Регистр управления 2 CONTROL2

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:20	Зарезервировано		
MSBFIRST	19	0x0	Управление очередностью приема и передачи данных: 0 – начинать с 0 бита; 1 – начинать с 9, 8 или 7 в зависимости от настроек длины посылки. Этот бит не влияет на передачу и прием бита четности. Этот бит может быть изменен только при остановке работы (UE=0)	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DATAINV	18	0x0	<p>Управление полярностью принимаемых и передаваемых данных: 0 – обычная полярность; 1 – обратная полярность (1=L, 0=H).</p> <p>Изменение полярности также влияет и на бит четности. Этот бит может быть изменен только при остановке работы (UE=0)</p>	RW
TXINV	17	0x0	<p>Управление полярностью TX выхода. Регистр влияет не только на данные, но и на стоп биты, старт биты и тд.</p> <p>0 – обычная полярность; 1 – обратная полярность (1=L, 0=H).</p> <p>Этот бит может быть изменен только при остановке работы (UE=0)</p>	RW
RXINV	16	0x0	<p>Управление полярностью RX входа. Регистр влияет не только на данные, но и на стоп биты, старт биты и тд</p> <p>0 – обычная полярность; 1 – обратная полярность (1=L, 0=H).</p> <p>Этот бит может быть изменен только при остановке работы (UE=0)</p>	RW
SWAP	15	0x0	<p>Регистр отвечает за обмен функциями TX выхода и RX входа:</p> <p>0 – обычное функционирование; 1 – обмен функциями (TX = RX и RX = TX).</p> <p>Этот бит может быть изменен только при остановке работы (UE=0)</p>	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
LBM	14	0x0	Регистр режима обратной внутренней петли: 0 – обычное функционирование; 1 – сигналы TX и RTS подаются внутрь блока в обход входов RX и CTS. При этом TX = 1 и RTS = 1, входы RX и CTS не активны. Этот бит может быть изменен только при остановке работы (UE=0)	RW
STOP_1	13	0x0	Количество стоп битов для приемника и передатчика: 0 – 1 стоп бит; 1 – 2 стоп бита. Этот бит может быть изменен только при остановке работы (UE=0)	RW
Зарезервировано	12	Зарезервировано		
CLKEN	11	0x0	Управление синхронным режимом: 0 – синхронный режим выключен; 1 – синхронный режим включен. Этот бит может быть изменен только при остановке работы (UE=0)	RW
CPOL	10	0x0	Управление полярностью выходного тактового сигнала CK: 0 – логический «0» вне транзакции; 1 – логическая «1» вне транзакции. Этот бит может быть изменен только при остановке работы (UE=0)	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CPHA	9	0x0	Управление фазой выходного тактового сигнала CK: 0 – первое изменение тактового сигнала – фронт; 1 – первое изменение тактового сигнала – срез (обратный фронт). Этот бит может быть изменен только при остановке работы (UE=0)	RW
LBCL	8	0x0	Управление последним тактовым импульсом: 0 – последний тактовый импульс отсутствует; 1 – последний тактовый импульс присутствует. Этот бит может быть изменен только при остановке работы (UE=0)	RW
Зарезервировано	7		Зарезервировано	
LBDIE	6	0x0	Управление прерыванием при обнаружении break состояния на RX линии: 0 – прерывание выключено; 1 – прерывание включено	RW
Зарезервировано	5:0		Зарезервировано	

Обозначение, разряды, начальное значение, назначение и доступ для регистра управления 3 представлены в таблице 20.

Таблица 20 – Регистр управления 3 CONTROL3

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:13		Зарезервировано	
OVRDIS	12	0x0	Выбор реакции на переполнение: 0 – при переполнении взводится флаг ошибки (ORE), новые данные не записываются; 1 – флаг переполнения не взводится, данные перезаписываются. Этот бит может быть изменен только при остановке работы (UE=0)	RW
Зарезервировано	11		Зарезервировано	
CTSIE	10	0x0	Управление прерыванием при обнаружении изменения CTS: 0 – прерывание выключено; 1 – прерывание включено	RW
CTSE	9	0x0	Выбор реакции на входной сигнал CTS: 0 – сигнал игнорируется; 1 – сигнал управляет передачей данных трансмиттером. Передача разрешена при (CTS = 0). Этот бит может быть изменен только при остановке работы (UE=0)	RW
RTSE	8	0x0	Управление выходным сигналом RTS: 0 – сигнал всегда в разрешающем состоянии (RTS = 0); 1 – сигнал находится в разрешающем состоянии (RTS = 0), только когда приемник готов принять данные. Этот бит может быть изменен только при остановке работы (UE=0)	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DMAT	7	0x0	Управление работой сигнала dma_trq: 0 – сигнал выключен; 1 – сигнал включен	RW
DMAR	6	0x0	Управление работой сигнала dma_rrq: 0 – сигнал выключен; 1 – сигнал включен	RW
Зарезервировано	5:4	Зарезервировано		
HDSEL	3	0x0	Выбор между полудуплексным и дуплексным режимами работы. В полудуплексном режиме RX не используется, все данные передаются и принимаются через TX. При наличии данных на передачу происходит отправка данных в остальное время прием: 0 – дуплексный режим работы; 1 – полудуплексный режим работы. Этот бит может быть изменен только при остановке работы (UE=0)	RW
SBKRQ	2	0x0	Переход передатчика в break состояние: 0 – обычный режим работы; 1 – состояние break на линии TX	RW
Зарезервировано	1	Зарезервировано		
EIE	0	0x0	Управление прерыванием при обнаружении ошибок приема данных (FE, ORE, NF): 0 – прерывание выключено; 1 – прерывание включено	RW

Обозначение, разряды, начальное значение, назначение и доступ для регистра делителя представлены в таблице 21.

Таблица 21 – Регистр делителя DIVIDER

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:16		Зарезервировано	
BRR	15:0	0x0	Значение делителя входного тактового сигнала (clk_in) При BRR < 16 USART не активен Бодрейт рассчитывается по следующей формуле: BR = F/BRR BR- бодрейт; F - значение частоты входного тактового сигнала Этот бит может быть изменен только при остановке работы (UE=0)	RW

Обозначение, разряды, начальное значение, назначение и доступ для регистра флагов и прерываний представлены в таблице 22.

Таблица 22 – Регистр флагов и прерываний FLAGS

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:23		Зарезервировано	
REACK	22	0x0	Флаг готовности приемника к работе после снятия сброса (UE и RE): 0 – сброс активен; 1 – сброс снят	RO
TEACK	21	0x0	Флаг готовности передатчика к работе после снятия сброса (UE и TE): 0 – сброс активен; 1 – сброс снят	RO
Зарезервировано	20:17		Зарезервировано	
BUSY	16	0x0	Флаг активности на линии RX: 0 – нет активности на линии RX; 1 – идет передача данных или break состояние	RO

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	15:11		Зарезервировано	
CTS	10	0x0	Флаг отображает текущее значение сигнала CTS: 0 – CTS = 0; 1 – CTS = 1	RO
CTSIF	9	0x0	Флаг взводится при изменении значения сигнала CTS. Флаг сбрасывается записью 1: 0 – состояние CTS не менялось с момента сброса флага; 1 – состояние CTS менялось с момента сброса флага	RW1C
LBDF	8	0x0	Флаг обнаружения break состояния на линии RX. Флаг сбрасывается записью 1: 0 – состояние break не обнаружено с момента сброса флага; 1 – состояние break обнаружено с момента сброса флага	RW1C
TXE	7	0x0	Флаг взводится при передаче данных в момент записи данных в сдвиговый регистр. Флаг сбрасывается записью данных в регистр TDR или записью 1: 0 – данные не записывались в сдвиговый регистр с момента сброса флага; 1 – данные записывались в сдвиговый регистр	RW1C
TC	6	0x0	Флаг взводится при передаче данных в момент отправки последнего стоп бита. Флаг сбрасывается записью 1: 0 – данные не отправлялись с момента сброса флага; 1 – данные отправлялись с момента сброса флага	RW1C

Обозначение	Разряды	Начальное значение	Назначение	Доступ
RXNE	5	0x0	Флаг взводится при записи принятых данных в регистр RDR из сдвигового регистра. Флаг сбрасывается записью 1 или чтением регистра RDR: 0 – данные не принимались после сброса флага; 1 – данные принимались после сброса флага	RW1C
IDLE	4	0x0	Флаг взводится при отсутствии активности на линии RX в течении 8 битовых тактов при взвешенном флаге RXNE. Флаг сбрасывается записью 1: 0 – состояние idle не обнаружено после сброса флага; 1 – состояние idle обнаружено после сброса флага	RW1C
ORE	3	0x0	Флаг взводится при попытке перезаписи RDR. Флаг сбрасывается записью 1: 0 – попыток перезаписи не обнаружено с момента сброса флага; 1 – попытка перезаписи обнаружена с момента сброса флага	RW1C
NF	2	0x0	Флаг обнаружения ложных переключений на линии RX. Флаг сбрасывается записью 1: 0 – помех не обнаружено с момента сброса флага; 1 – помехи обнаружены с момента сброса флага	RW1C
FE	1	0x0	Флаг взводится при обнаружении ошибок в стоп бите или битах. Флаг сбрасывается записью 1: 0 – ошибок приема не обнаружено с момента сброса флага; 1 – ошибка приема обнаружена с момента сброса флага	RW1C

Обозначение	Разряды	Начальное значение	Назначение	Доступ
PE	0	0x0	Флаг обнаружения ошибочного бита четности. Флаг сбрасывается записью 1: 0 – ошибок в битах четности не обнаружено с момента сброса флага; 1 – ошибка в бите четности обнаружена с момента сброса флага	RW1C

Обозначение, разряды, начальное значение, назначение и доступ для регистра принятых данных представлены в таблице 23.

Таблица 23 – Регистр принятых данных RXDATA

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:9		Зарезервировано	
RDR	8:0	0x0	–	RO

Обозначение, разряды, начальное значение, назначение и доступ для регистра передаваемых данных представлены в таблице 24.

Таблица 24 – Регистр передаваемых данных TXDATA

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:9		Зарезервировано	
TDR	8:0	0x0	–	RW

Обозначение, разряды, начальное значение, назначение и доступ для регистра модемных сигналов представлены в таблице 25.

Таблица 25 – Регистр модемных сигналов MODEM

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:9		Зарезервировано	
DTR	8	0x0	Регистр управления сигналом DTR: 0 – отсутствие сигнала готовности; 1 – готовность к приему данных	RW
DCD	7	0x0	Регистр содержит текущее значение сигнала DCD: 0 – отсутствие несущей; 1 – обнаружение несущей	RO
RI	6	0x0	Регистр содержит текущее значение сигнала RI: 0 – отсутствие звонка; 1 – звонок (вызов) на телефонной линии	RO
DSR	5	0x0	Регистр содержит текущее значение сигнала DSR: 0 – источник данных не готов; 1 – источник данных готов	RO
Зарезервировано	4		Зарезервировано	
DCDIF	3	0x0	Флаг взводится при изменении входного сигнала DCD. Флаг сбрасывается записью 1: 0 – изменений сигнала не обнаружено с момента сброса флага; 1 – сигнал изменен с момента сброса флага	RW1C
RIIF	2	0x0	Флаг взводится при изменении входного сигнала RI с 0 на 1. Флаг сбрасывается записью 1: 0 – изменений сигнала с 0 на 1 не обнаружено с момента сброса флага; 1 – сигнал изменен с момента сброса флага	RW1C

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DSRIF	1	0x0	Флаг взводится при изменении входного сигнала DSR. Флаг сбрасывается записью 1: 0 – изменений сигнала не обнаружено с момента сброса флага; 1 – сигнал изменен с момента сброса флага	RW1C
Зарезервировано	0	Зарезервировано		

3.7 Модуль I2C

Модуль обеспечивает интерфейс для связи контроллера и последовательной шины I2C. Позволяет работать в конфигурациях multi-master, multi-slave; управляет всеми состояниями шины I2C, протоколом передачи, соблюдением временных ограничений. Поддерживает режимы Standard (Sm), Fast (Fm), Fast Plus (Fm+). Возможно использование DMA для снижения загрузки процессора.

3.7.1 Характеристики

Характеристики модуля I²C:

- соответствие спецификации шины I2C rev03:
 - 1) режимы ведущий и ведомый;
 - 2) несколько ведущих нашине;
 - 3) режим Standard (до 100 кГц);
 - 4) режим Fast (до 400 кГц);
 - 5) режим Fast Plus (до 1 МГц);
 - 6) адресация 7 и 10 бит;
 - 7) два собственных адреса ведомого:
 - а) 7 или 10-битный;
 - б) 7-битный с настраиваемой маской;
 - 8) квитирование всех 7-битных адресов;
 - 9) общий вызов;
 - 10) программируемые времена предустановки (setup) и выдержки (hold);
 - 11) простое управление событиями;

- 12) возможно растягивание тактового сигнала шины;
 - 13) программный сброс;
- буфер данных 1 байт с возможностью работы через DMA;
 - программируемые аналоговый и цифровой фильтры шумов;
 - независимый от системного тактового сигнала (PCLK);
 - программное маскирование прерываний.

3.7.2 Состав модуля I2C

Структура модуля I2C представлена на рисунке 22.

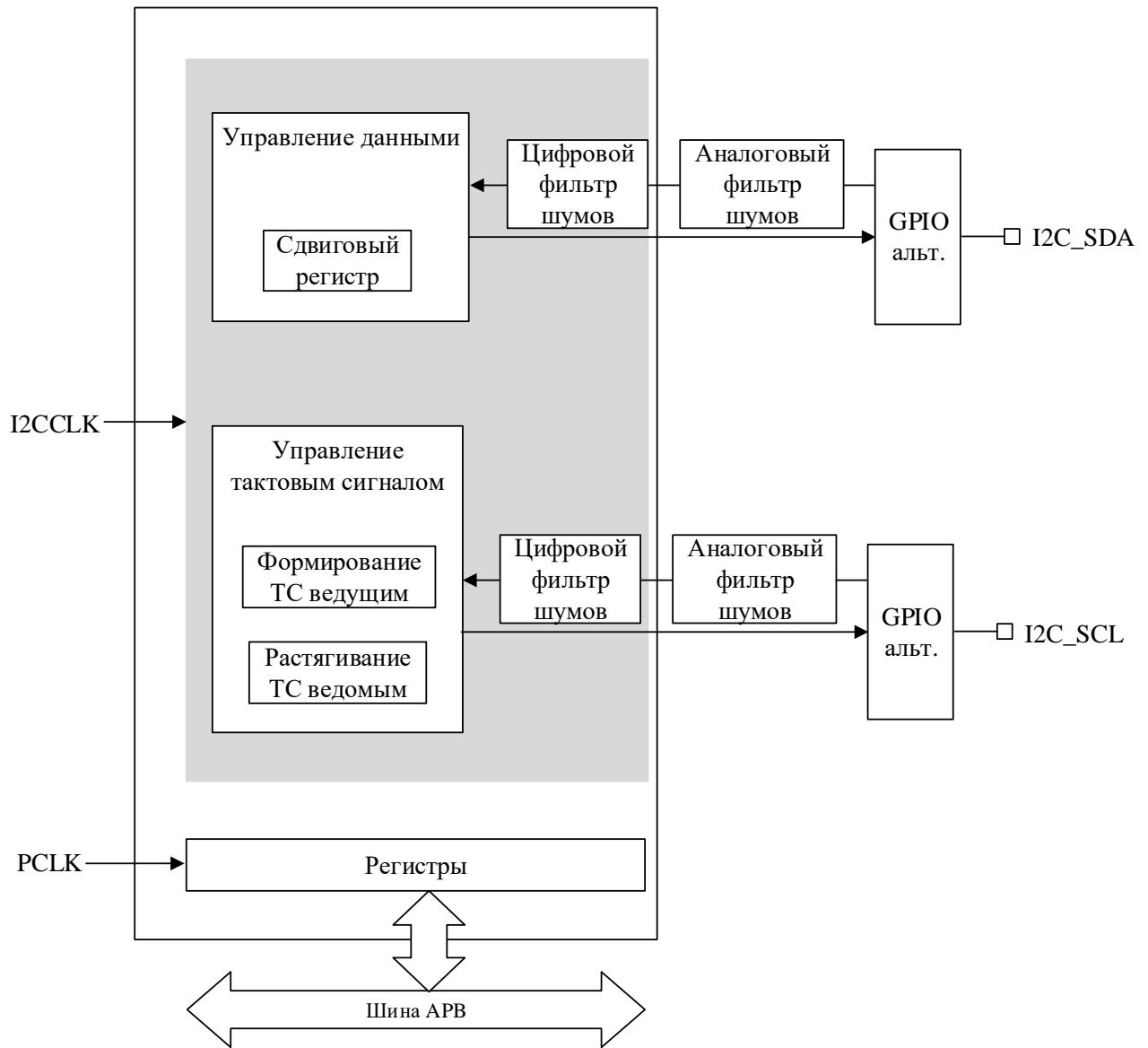


Рисунок 22 – Схема модуля I2C

3.7.3 Требования к тактовым сигналам

Ядро блока тактируется от частоты периферийной шины ($f_{I2CCLK} = f_{PCLK}$). t_{PCLK} – период тактового сигнала периферийной шины PCLK и t_{I2CCLK} – период тактового сигнала модуля I2C должны удовлетворять формулам (1):

$$\begin{aligned} t_{PCLK} &< (t_{LOW} - t_{filters})/4, \\ t_{PCLK} &< 4/3t_{SCL}, \\ t_{I2CCLK} &< t_{HIGH}. \end{aligned} \quad (1)$$

где t_{LOW} – длительность низкого уровня SCL и t_{HIGH} – длительность высокого уровня SCL;

$t_{filters}$ – суммарная задержка включенных аналогового и цифрового фильтров. Задержка цифрового фильтра равна произведению значения бит DNF на t_{I2CCLK} . t_{SCL} – период SCL.

3.7.4 Выбор режима

Интерфейс может работать в четырех режимах:

- ведомый передатчик;
- ведомый приёмник;
- ведущий передатчик;
- ведущий приёмник.

По умолчанию выбран режим «ведомый». Интерфейс автоматически переключается из режима «ведомый» в режим «ведущий», когда он вырабатывает комбинацию START на шине и из режима «ведущий» в режим «ведомый», при проигрыше арбитража или выработке комбинации STOP на шине.

В режиме «ведущий» интерфейс инициирует передачу данных и генерирует тактовый сигнал. Передача начинается с комбинации START и заканчивается комбинацией STOP. Обе комбинации в режиме «ведущий» вырабатываются программой.

В режиме «ведомый» интерфейс распознаёт собственные адреса (7 или 10 бит) и адрес общего вызова. Распознавание адреса общего вызова может отключаться программой.

Данные и адреса передаются по 8 бит старшим битом вперёд (MSB). Первый(е) байт(ы), следующие за комбинацией START, содержат адрес (один байт в 7-битном режиме, два – в 10-битном). Адрес передаётся интерфейсом только в режиме «ведущий».

В течение девятого такта после восьми тактов данных приёмник должен отправить подтверждение передатчику (ACK), что продемонстрировано на рисунке 23.

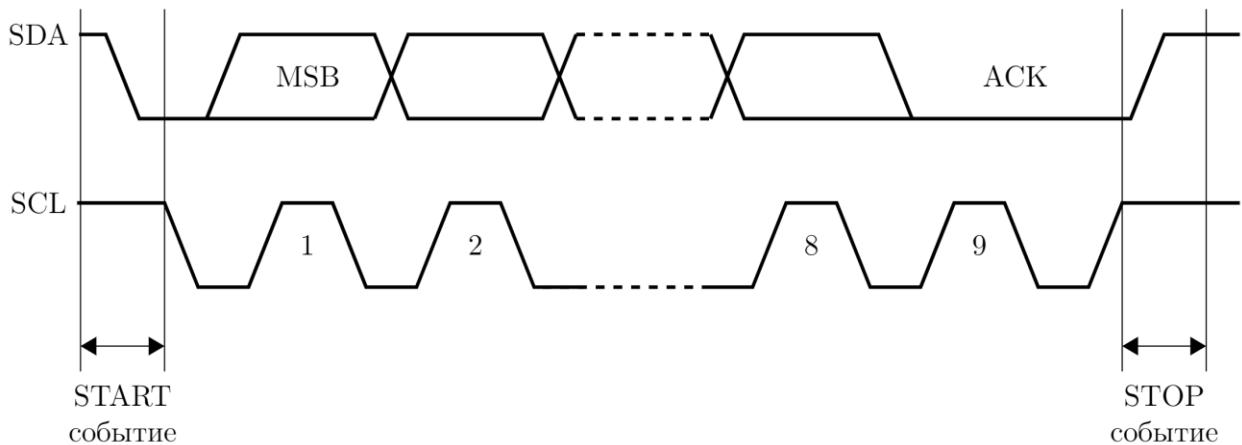


Рисунок 23 – Формат кадра

Подтверждение может быть заблокировано программой. Адреса интерфейса выбираются программой.

3.7.5 Инициализация и программный сброс

На рисунке 24 приведен алгоритм инициализации I2C.

Интерфейс включается битом PE в регистре CR1.

Программный сброс производится очисткой бита PE в регистре CR1. При этом линии SCL и SDA освобождаются, внутренние состояния и биты управления соединением, статуса сбрасываются в начальное состояние.

Список сбрасываемых битов:

- регистр CR2: START, STOP, NACK;
- регистр ISR: BUSY, TXE, TXIS, RXNE, ADDR, NACKF, TCR, TC, STOPF, BERR, ARLO, OVR.

Бит PE должен находиться в состоянии «0» как минимум три такта APB для осуществления сброса.

3.7.6 Фильтры шумов

Если необходимо задействовать фильтры шумов, они должны быть настроены до включения интерфейса.

Если на входах SDA и SCL применены аналоговые фильтры, то их состоянием можно управлять битом ANFOFF в регистре CR1.

При включенном цифровом фильтре внутреннее состояние линий SDA и SCL меняется, только если они остаются стабильными в течении более чем DNF периодов I2CCLK.

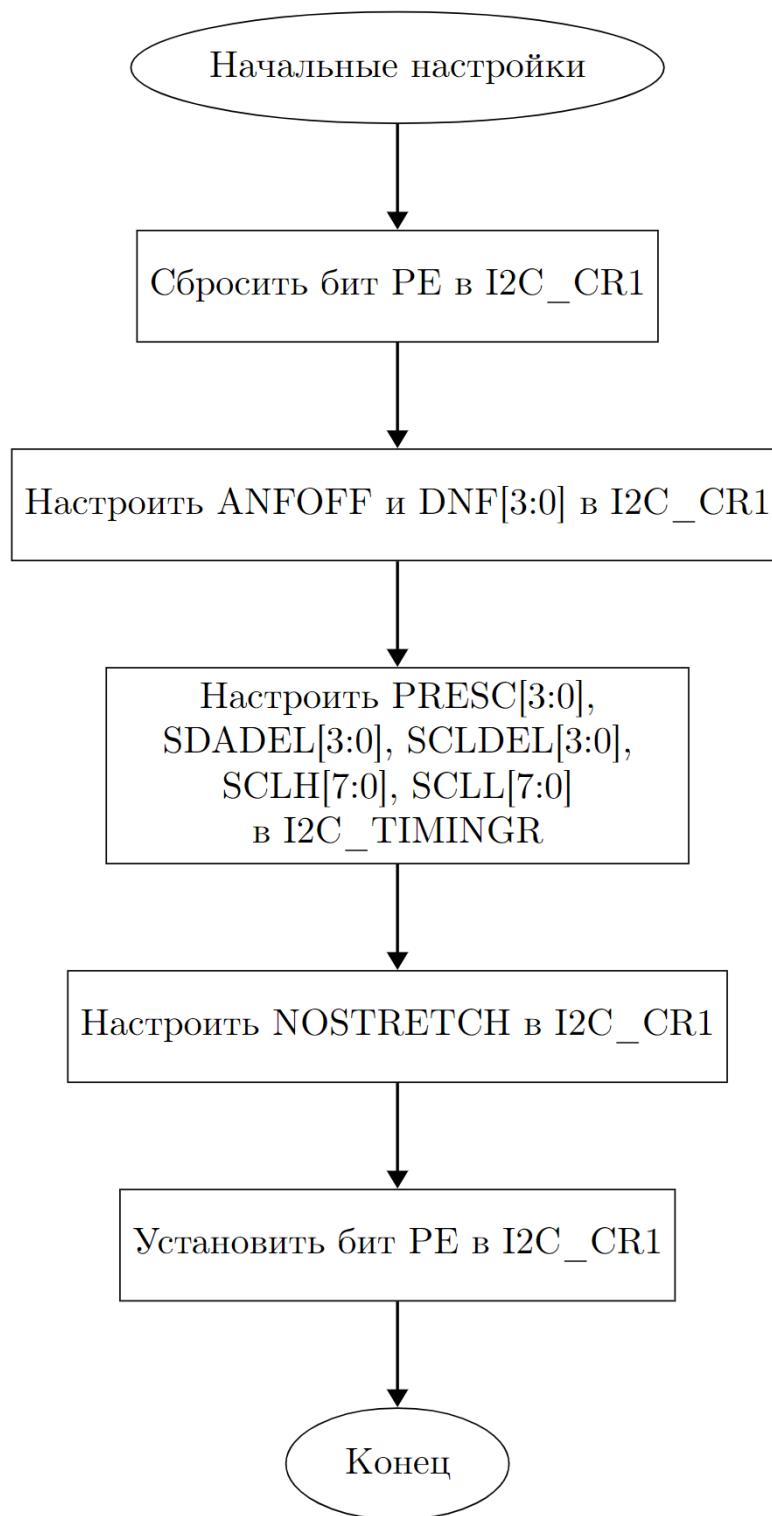


Рисунок 24 – Инициализация I2C

3.7.7 Временные ограничения

Временные ограничения, значения которых приведены в таблице 26, должны быть настроены для гарантированных предустановок и удержания данных в режимах ведущего и ведомого. Они настраиваются битами PRESC[3:0], SCLDEL[3:0] и SDADEL[3:0] в регистре TIMINGR. Временные ограничения изображены на рисунке 25.

Примечание. На каждом такте после детектирования спада SCL и ведущий и ведомый удерживают SCL на низком уровне в течение, по крайней мере, $[(SDADEL + SCLDEL + 1) \times (PRESC + 1) + 1] \times tI2CCLK$ в режимах отправки и приёма. В режиме отправки, в случае если данные не записаны в регистр TXDR по окончании счёта SDADEL, интерфейс удерживает SCL до записи данных в регистр. После выдачи старшего бита на линию SDA, SCL удерживается в низком уровне до окончания счёта SCLDEL.

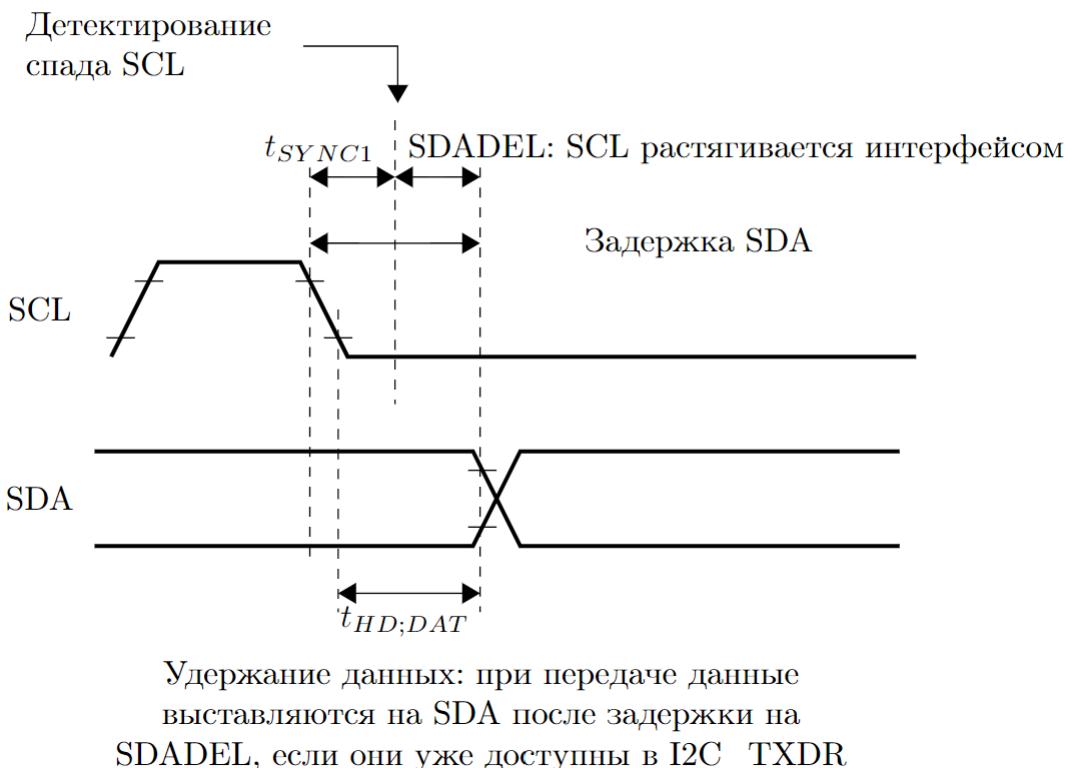
Если в режиме «ведомый» NOSTRETCH=1, SCL не удерживается. Соответственно, SDADEL должно быть запрограммировано так, чтобы гарантировать время предустановки данных.

Изменение настроек задержек не допускается при включенном интерфейсе. Режим NOSTRETCH также должен быть настроен до включения интерфейса. Изменение NOSTRETCH не допускается при включенном интерфейсе.

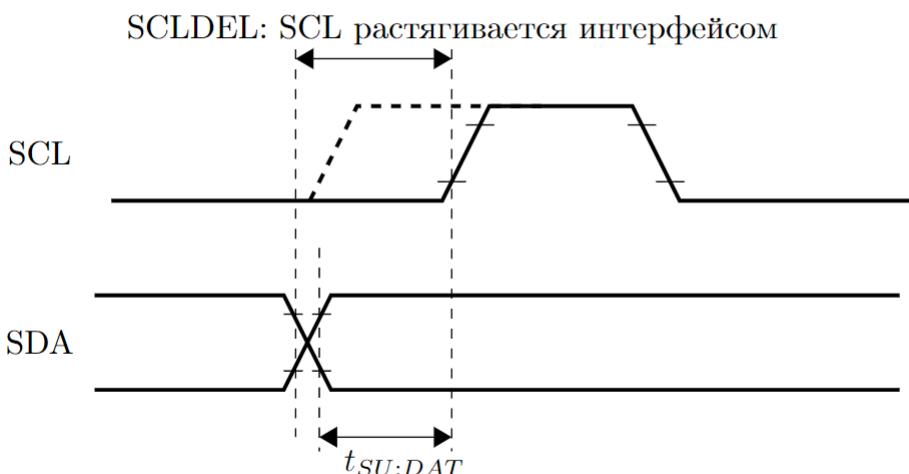
Таблица 26 - Спецификация временных ограничений шины I2C

Обозначение	Параметр	Standard-mode (Sm)		Fast-mode (Fm)		Fast-mode Plus (Fm+)		Ед. изм.
		Мин.	Макс.	Мин.	Макс.	Мин.	Макс.	
t _{HD;DAT}	Длительность удержания данных	0	—	0	—	0	—	мкс
t _{VU;DAT}	Длительность действительных данных	—	3,45	—	0,9	—	0,45	мкс
t _{SU;DAT}	Длительность предустановки данных	250	—	100	—	50	—	нс
t _r	Длительность фронта сигналов SDA, SCL	—	1000	—	300	—	120	нс
t _f	Длительность спада сигналов SDA, SCL	—	300	—	300	—	120	нс

Время удержания данных



Время предустановки данных



Предустановка данных: при передаче счетчик SCLDEL стартует, когда данные уже выставлены на линию SDA

Рисунок 25 – Временные ограничения

3.7.7.1 Длительность удержания данных

Когда регистрируется спад на линии SCL, данные по SDA отправляются с задержкой по формуле (2):

$$t_{SDADEL} = \text{SDADEL} \times t_{\text{PRESC}} + t_{\text{I2CCLK}}, \quad (2)$$

где $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$.

t_{SDADEL} влияет на время удержания $t_{HD;DAT}$.

Значение общей задержки на линии SDA определяется по формуле (3):

$$t_{SYNC1} + \{ [SDADEL \times (PRESC+1) + 1] \times t_{I2CCLK} \} \quad (3)$$

Длительность t_{SYNC1} зависит от параметров:

- наклон спада SCL;
- задержка на аналоговом фильтре шумов, если он включен ($t_{AF(min)} < t_{AF} < t_{AF(max)}$ нс);
- задержка на цифровом фильтре шумов, если он включен ($t_{DNF} = DNF \times t_{I2CCLK}$);
- задержка на синхронизацию SCL в домене I2CCLK (2 – 3 периода I2CCLK).

Значение $SDADEL$ должно быть определено, так чтобы выполнялись условия в формулах (4) и (5).

$$\{t_{f(max)} + t_{HD;DAT(min)} - t_{AF(min)} - [(DNF + 3) \times t_{I2CCLK}] \} / \{(PRESC + 1) \times t_{I2CCLK}\} \leq SDADEL \quad (4)$$

$$SDADEL \leq \{t_{HD;DAT(max)} - t_{AF(max)} - [(DNF + 4) \times t_{I2CCLK}] \} / \{(PRESC + 1) \times t_{I2CCLK}\} \quad (5)$$

$t_{AF(max)}$ и $t_{AF(min)}$ применяются если используется аналоговый фильтр.

Это применимо в случае, если устройство не растягивает тактовый сигнал SCL.

Иначе данные должны быть предустановлены, прежде чем устройство отпустит SCL.

Примечание – Условие может быть нарушено при $NOSTRETCH = 0$, т.к. устройство растягивает низкий уровень SCL, гарантируя соответствие времени предустановки значению $SCLDEL$.

3.7.7.2 Длительность предустановки данных

После задержки t_{SDADEL} или отправки данных в случае, когда ведомому пришлось растянуть тактовый сигнал из-за отсутствия данных в регистре TXDR, линия SCL удерживается в низком уровне в течение времени предустановки ($t_{SU;DAT}$), согласно формуле (6):

$$t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC} \quad (6)$$

где $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$;

t_{SCLDEL} влияет на время предустановки $t_{SU;DAT}$.

$SCLDEL$ должно быть запрограммировано согласно формуле (7):

$$\{[t_{r(max)} + t_{SU;DAT(min)}] / [(PRESC + 1) \times t_{I2CCLK}] \} - 1 \leq SCLDEL \quad (7)$$

3.7.7.3 Формирование SCL в режиме «ведущий»

Дополнительно в режиме «ведущий» длительности фаз SCL должны быть настроены через биты PRESC[3:0], SCLH[7:0] и SCLL[7:0] в регистре TIMINGR:

- после детектирования спада SCL, перед отпусканием выхода SCL, вставляется задержка согласно формуле (8):

$$t_{SCLL} = (SCLL+1) \times t_{PRESC} \quad (8)$$

где $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$;

t_{SCLL} влияет на t_{LOW} .

- после детектирования фронта SCL, перед притягиванием выхода SCL к низкому уровню, вставляется задержка согласно формуле (9):

$$t_{SCLH} = (SCLH+1) \times t_{PRESC} \quad (9)$$

где $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$;

t_{SCLH} влияет на t_{HIGH} .

3.7.8 Передача данных

Передача осуществляется через буферные регистры отправки и приёма и сдвиговый регистр.

3.7.8.1 Приём

Сдвиговый регистр заполняется данными с входа SDA. После восьмого импульса на SCL, содержимое сдвигового регистра переписывается в регистр RXDR, если он свободен (Бит RXNE=0). Если RXNE=1, то есть предыдущие полученные данные не вычтены, линия SCL удерживается на низком уровне, пока RXDR не будет прочитан. Удержание осуществляется между восьмым и девятым импульсами, это изображено на рисунке 26.

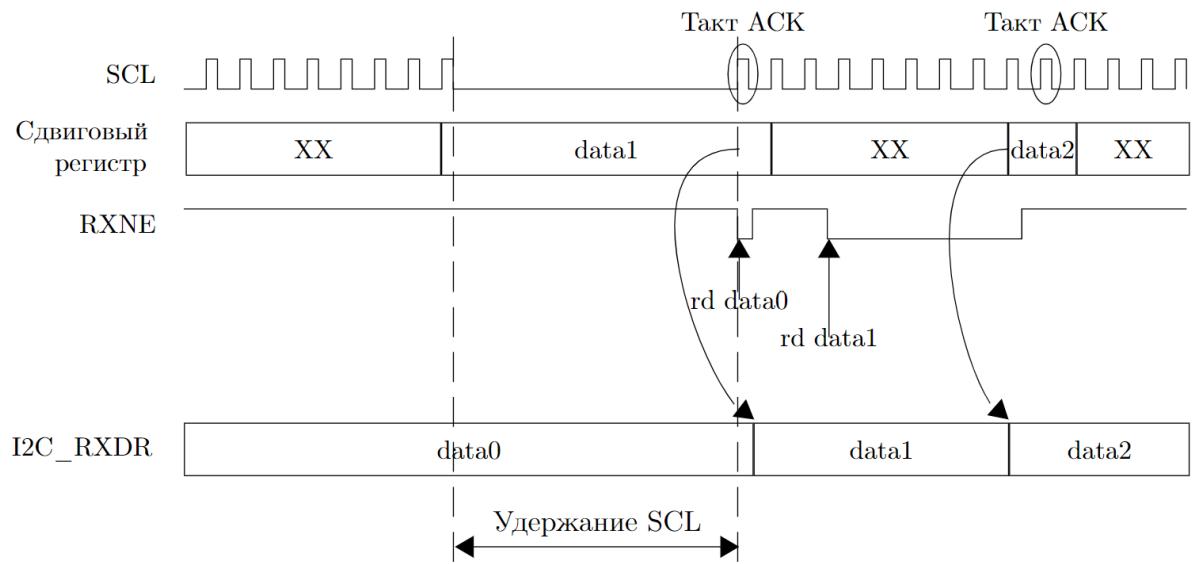


Рисунок 26 – Приём данных

3.7.8.2 Отправка

Если регистр TXDR заполнен (бит TXE=0), его содержимое переписывается в сдвиговый регистр после девятого импульса на линии SCL (такт подтверждения ACK). Затем содержимое сдвигового регистра выдвигается на линию SDA. Если бит TXE=1, то есть данные в регистре TXDR не записаны, то линия SCL удерживается на низком уровне, пока в регистре TXDR не будут записаны данные. Удержание осуществляется после девятого импульса на SCL, что представлено на рисунке 27.

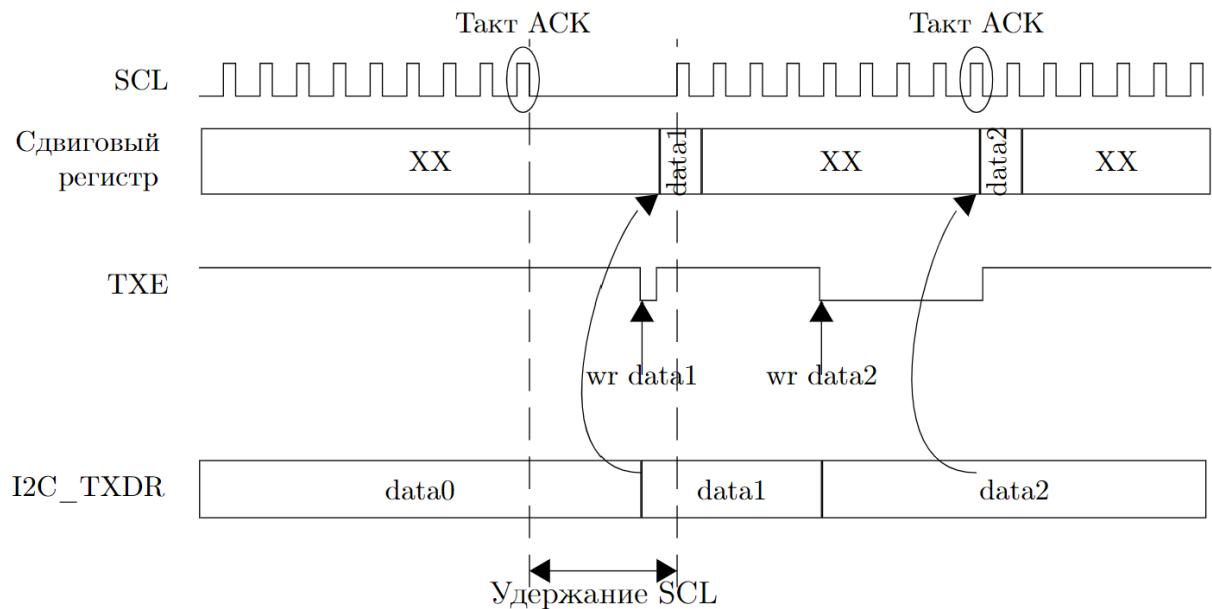


Рисунок 27 – Отправка данных

3.7.9 Аппаратное управление передачей

В интерфейсе есть счетчик байтов для организации передачи данных и закрытия соединения в следующих режимах:

- выдача комбинаций NACK, STOP и ReSTART в режиме «ведущий»;
- контроль ACK в режиме «ведомый».

Счетчик байтов всегда используется в режиме «ведущий». По умолчанию он заблокирован в режиме «ведомый», но может быть разблокирован установкой в «1» бита SBC в регистре CR2.

Количество байтов для передачи программируется в битах NBYTES[7:0] в регистре CR2. Если количество байтов больше, чем 255 или если приемник хочет контролировать значение бита подтверждения (ACK) полученного байта данных, то должен быть активирован режим перезагрузки битом RELOAD в регистре CR2. В этом режиме вводится флаг TCR и вырабатывается прерывание, если установлен бит TCIE, когда запрограммированное в NBYTES количество байтов передано. Флаг TCR сбрасывается программой, когда в NBYTES записывается ненулевое значение.

Когда в NBYTES записывается последнее количество байтов, флаг RELOAD должен быть сброшен.

Если RELOAD = 0, счетчик может использоваться в двух режимах:

- автоматическое окончание (AUTOEND = 1 в регистре CR2) – ведущий автоматически отправляет состояние STOP по окончании передачи всех данных, заданных в NBYTES;
- программное окончание (AUTOEND = 0 в регистре CR2) – по окончании передачи всех данных, заданных в NBYTES вводится флаг TC и вырабатывается прерывание, если бит TCIE установлен и ожидается реакция программы. Флаг TC сбрасывается программой, когда биты START или STOP в регистре CR2 устанавливаются в «1». Этот режим следует использовать, когда ведущий хочет отправить комбинацию RESTART.

Бит AUTOEND не имеет действия, если установлен бит RELOAD. В таблице 27 представлено аппаратное управление передачей битами SBC, RELOAD и AUTOEND.

Таблица 27 – Аппаратное управление передачей

Режим	Бит SBC	Бит RELOAD	Бит AUTOEND
Ведущий Tx/Rx NBYTES + STOP	x	0	1
Ведущий Tx/Rx + NBYTES + RESTART	x	0	0
Ведомый Tx/Rx Все принятые байты подтверждены	0	x	x
Ведомый Rx с контролем подтверждения	1	1	x

3.7.10 Режим «ведомый» I2C

3.7.10.1 Инициализация ведомого

Чтобы интерфейс работал в режиме «ведомый», нужно задействовать хотя бы один адрес ведомого. Доступны два регистра OAR1 и OAR2 для программирования собственных адресов ведомого OA1 и OA2.

- OA1 может быть настроен либо в 7-битном режиме (по умолчанию), либо в 10-битном, для чего должен быть установлен бит OA1MODE в регистре OAR1. Использование OA1 разрешается, если установлен бит OA1EN в регистре OAR1.
- При необходимости можно настроить дополнительный адрес ведомого OA2. Маскирование младших семи бит OA2 можно настроить битами OA2MSK[2:0] в регистре OAR2. Таким образом при значениях OA2MSK от 1 до 6 только OA2[7:2], OA2[7:3], OA2[7:4], OA2[7:5], OA2[7:6] или OA2[7] сравниваются с полученным адресом. Если OA2MSK не равно «0» OA2 не сравнивается с зарезервированными адресами (0b0000XXX и 0b1111XXX), они не подтверждаются (ACK). Если OA2MSK=7, все 7-битные адреса подтверждаются (за исключением зарезервированных). Адрес OA2 всегда 7-битный.
- Зарезервированные адреса подтверждаются, если они разрешены специальным битом и запрограммированы в регистре OAR1 или регистре OAR2 совместно с битами OA2MSK=0.
- Использование OA2 разрешается, если установлен бит OA2EN в регистре OAR2.
- Адрес общего вызова разрешается установкой бита GCEN в регистре CR1.

Если интерфейс выбран одним из этих разрешенных адресов, взводится флаг статуса ADDR и вырабатывается прерывание, если бит ADDRIE установлен.

По умолчанию ведомый использует удержание SCL на нижнем уровне, когда это необходимо, чтобы программа смогла завершить необходимые действия. Если ведущий не поддерживает удержание SCL, интерфейс должен быть настроен с битом NOSTRETCH=1 в регистре CR1.

После получения прерывания ADDR, если разрешено несколько адресов, пользователь должен вычитать биты ADDCODE[6:0] в регистре ISR, чтобы выяснить, какой из адресов вызван. Так же должен быть проанализирован флаг DIR, чтобы узнать направление передачи.

3.7.10.2 Удержание SCL ведомым (NOSTRETCH=0)

По умолчанию ведомый удерживает SCL в следующих случаях:

- когда взведён флаг ADDR: принятый адрес соответствует одному из разрешенных адресов ведомого. Это удержание прекращается программой после сброса флага ADDR – устанавливается бит ADDRCF;
- во время отправки, если предыдущая отправка данных завершена и новые данные не записаны в регистр TXDR или первый байт данных не записан, когда флаг ADDR сброшен (TXE=1). Это удержание прекращается после записи данных в регистр TXDR;
- во время приёма, если регистр RXDR всё ещё не вычитан и приём новых данных завершён. Это удержание прекращается после чтения из регистра RXDR;
- когда бит TCR=1 в режиме SBC или в режиме перезагрузки (SBC=1 и RELOAD=1), то есть, передан последний байт данных. Это удержание прекращается после сброса бита TCR записью ненулевого значения в биты NBUTES[7:0];
- после детектирования спада SCL интерфейс удерживает линию SCL в течении $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{I2CCLK}$.

3.7.10.3 Ведомый без удержания SCL (NOSTRETCH=1)

При NOSTRETCH=1 в регистре CR1, интерфейс не удерживает SCL в низком уровне:

- SCL не удерживается при взведённом флаге ADDR;
- во время отправки данные должны быть записаны в TXDR до первого импульса SCL, соответствующего их отправке. Иначе случается недостача данных и взводится флаг OVR в регистре ISR и вырабатывается прерывание, если установлен бит ERRIE в регистре CR1. Флаг OVR также взводится, когда начинается передача первого бита данных и флаг STOPF всё ещё взведён (до сих пор не сброшен);
- во время приёма данные должны быть вычитаны из регистра RXDR до девятого импульса SCL (ACK бит) следующего байта данных. Иначе происходит переполнение данных и взводится флаг OVR в регистре ISR и вырабатывается прерывание, если установлен бит ERRIE в регистре CR1.

3.7.10.4 Контроль передачи данных в режиме «ведомый» (SBC)

Для разрешения контроля ведомым бита ACK во время приёма должен быть включен режим контроля передачи данных установкой бита SBC в регистре CR1.

Чтобы разрешить контроль ACK в режиме приёма ведомым, должен быть выбран режим перезагрузки (RELOAD = 1). Для контроля над каждым байтом биты NBYTES следует инициализировать значением 0x1 в обработчике прерывания по ADDR и перезаписывать в них 0x1 после каждого полученного байта. Когда байт получен, взводится флаг TCR, SCL удерживается между восьмым и девятым импульсами. Пользователь может вычитать данные из регистра RXDR и затем решить, подтверждать их или нет, конфигурируя бит ACK в регистре CR2. Удержание SCL прекращается записью в NBYTES ненулевого значения, отправляется бит подтверждения приёма и следующий байт может быть получен.

В NBYTES может быть загружено значение больше, чем 0x1 и в этом случае процесс приёма продолжается до получения NBYTES байт данных.

Бит SBC должен быть установлен, пока интерфейс заблокирован или пока ведомый не адресован, или когда ADDR = 1.

Бит RELOAD может быть изменён, когда ADDR = 1 или TCR = 1.

Режим SBC не совместим с режимом NOSTRETCH.

Алгоритм инициализации I2C в режиме «ведомый» приведен на рисунке 28.

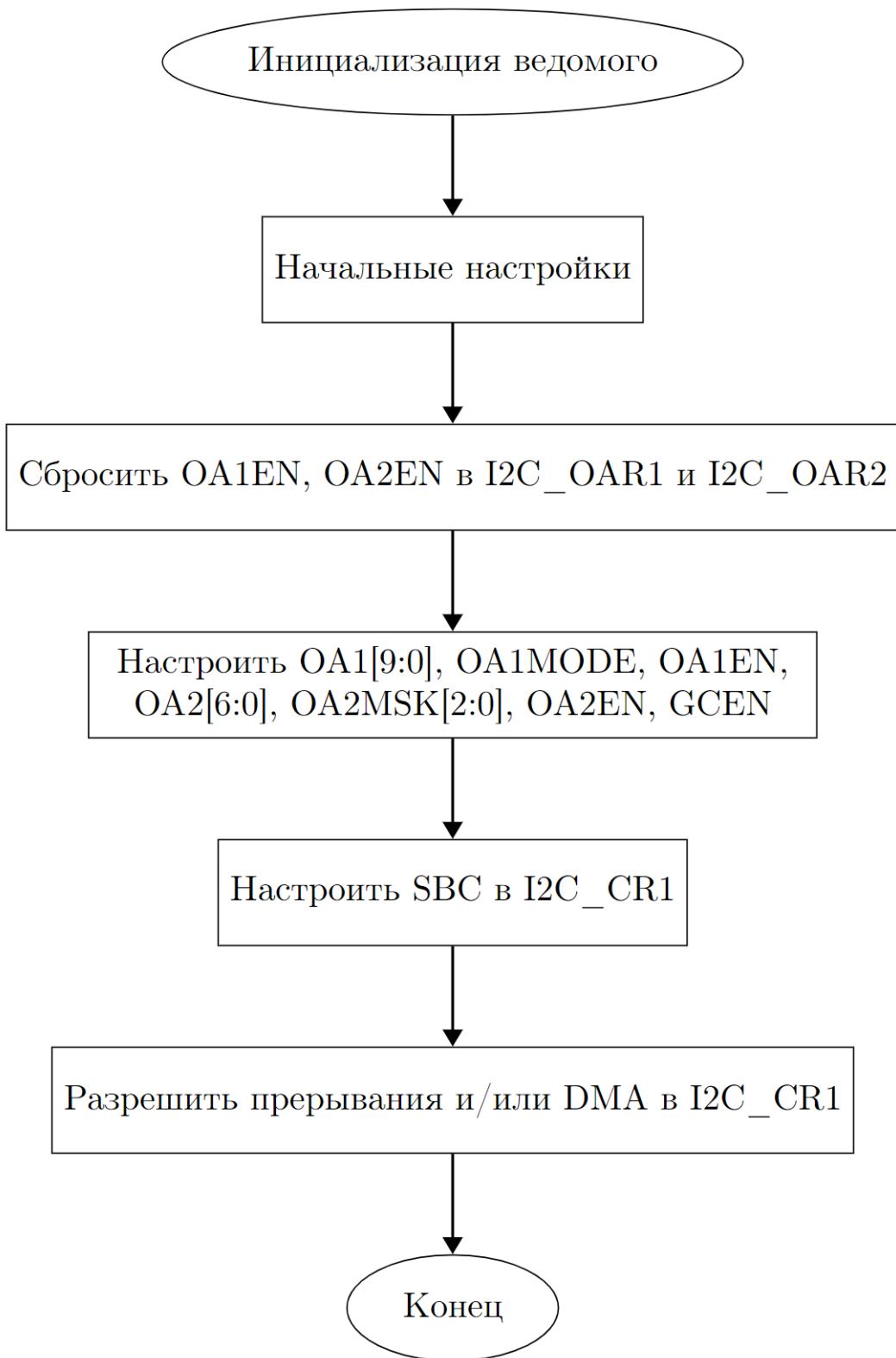


Рисунок 28 - Инициализация режима «ведомый»

3.7.10.5 Передача в режиме «ведомый»

После освобождения регистра TXDR, взводится флаг статуса прерывания TXIS. Прерывание вырабатывается, если оно разрешено битом TXIE в регистре CR1. Флаг TXIS сбрасывается после записи данных для отправки в регистр TXDR.

Если не получено подтверждение данных (NACK), взводится флаг NACKF в регистре статуса ISR и вырабатывается прерывание, если оно разрешено битом NACKIE в регистре CR1. Ведомый автоматически освобождает линии SDA и SCL, чтобы дать возможность ведущему отправить комбинацию STOP или RESTART. Флаг TXIS не взводится при получении NACK.

Если получена комбинация STOP и бит STOPIE в регистре CR1 установлен, то взводится флаг STOPF в регистре ISR и вырабатывается прерывание (рисунок 31).

В большинстве применений бит SBC устанавливается равным «0». В этом случае, если флаг TXE=0 при получении адреса ведомого (ADDR=1), пользователь может решить отправлять в качестве первого байта данных содержимое регистра TXDR или сбросить его содержимое записью «1» в бит TXE, чтобы инициировать процедуру записи байта данных.

В режиме контроля передачи данных ведомым (SBC=1) количество байт для передачи программируется в обработчике прерывания по совпадению адреса ведомого (ADDR=1). В этом случае количество срабатываний прерывания TXIS соответствует значению,енному в NBYTES.

При NOSTRETCH=1 SCL не удерживается, пока выставлен флаг ADDR и пользователь не может сбросить содержимое регистра TXDR в обработчике прерывания по совпадению адреса ведомого, чтобы записать первый байт данных. Первый байт данных для пересылки должен быть записан заранее:

- этот байт может быть записан при обработке последнего прерывания TXIS в предыдущем цикле передачи;
- если эти данные не предназначены для передачи, регистр TXDR может быть сброшен записью бита TXE, чтобы инициировать запись нового байта данных. Флаг STOPF должен быть сброшен только после указанных действий, чтобы гарантировать их выполнение до начала передачи первых данных, следующих за подтверждением (ACK) адреса;
- если флаг STOPF всё ещё установлен в момент начала передачи данных, возникает ошибка из-за неполного завершения программы (взводится флаг OVR);
- если нужно событие TXIS (для возникновения прерывания передачи или запроса данных по DMA), пользователь может установить бит TXIS в дополнение к биту TXE, чтобы сгенерировать событие TXIS.

На рисунках 29, 30 приведены алгоритмы передачи в режиме «ведомый», при различных значениях бита NOSTRETCH.

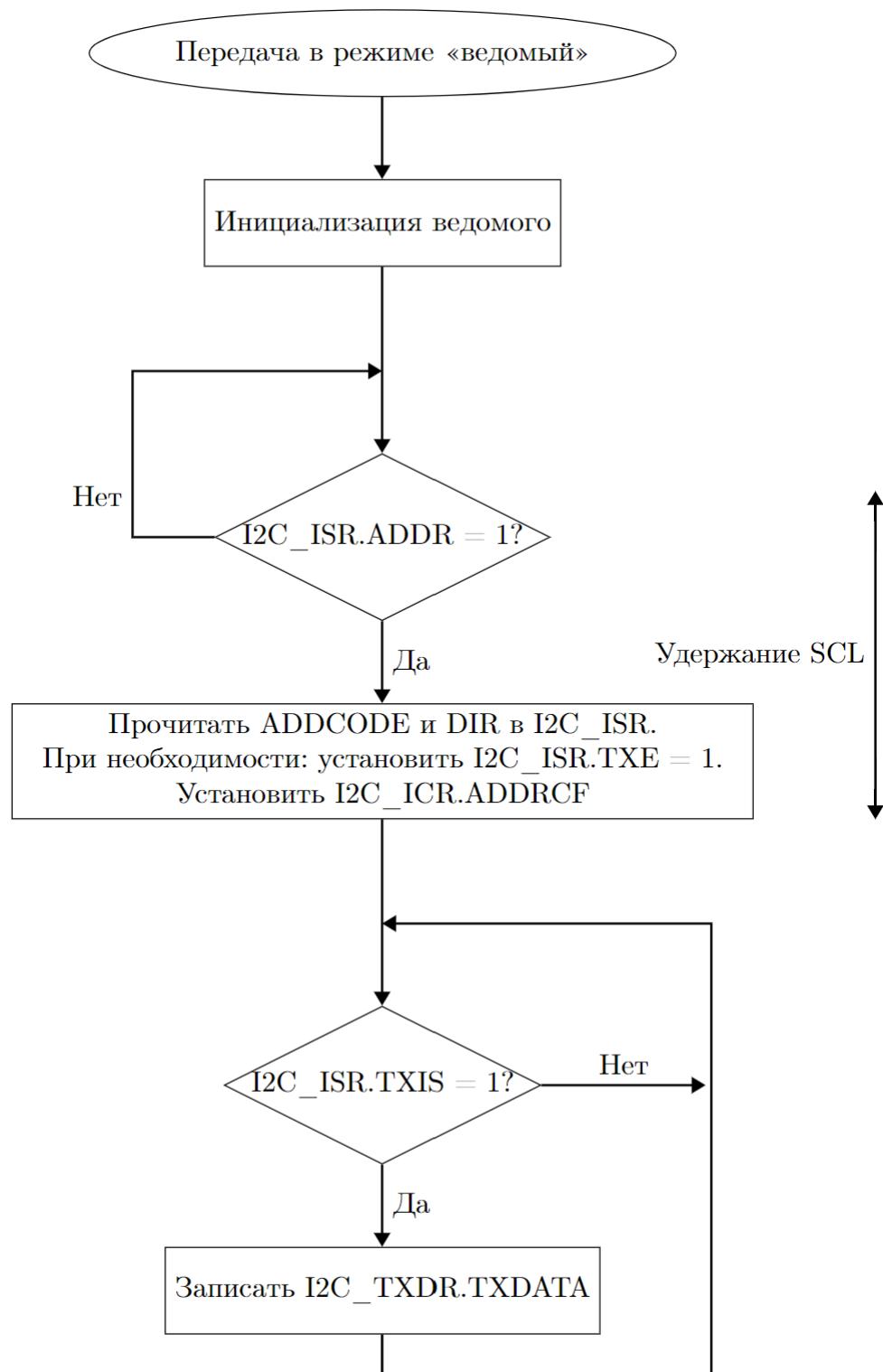


Рисунок 29 - Алгоритм передачи в режиме «ведомый», при NOSTRETCH = 0

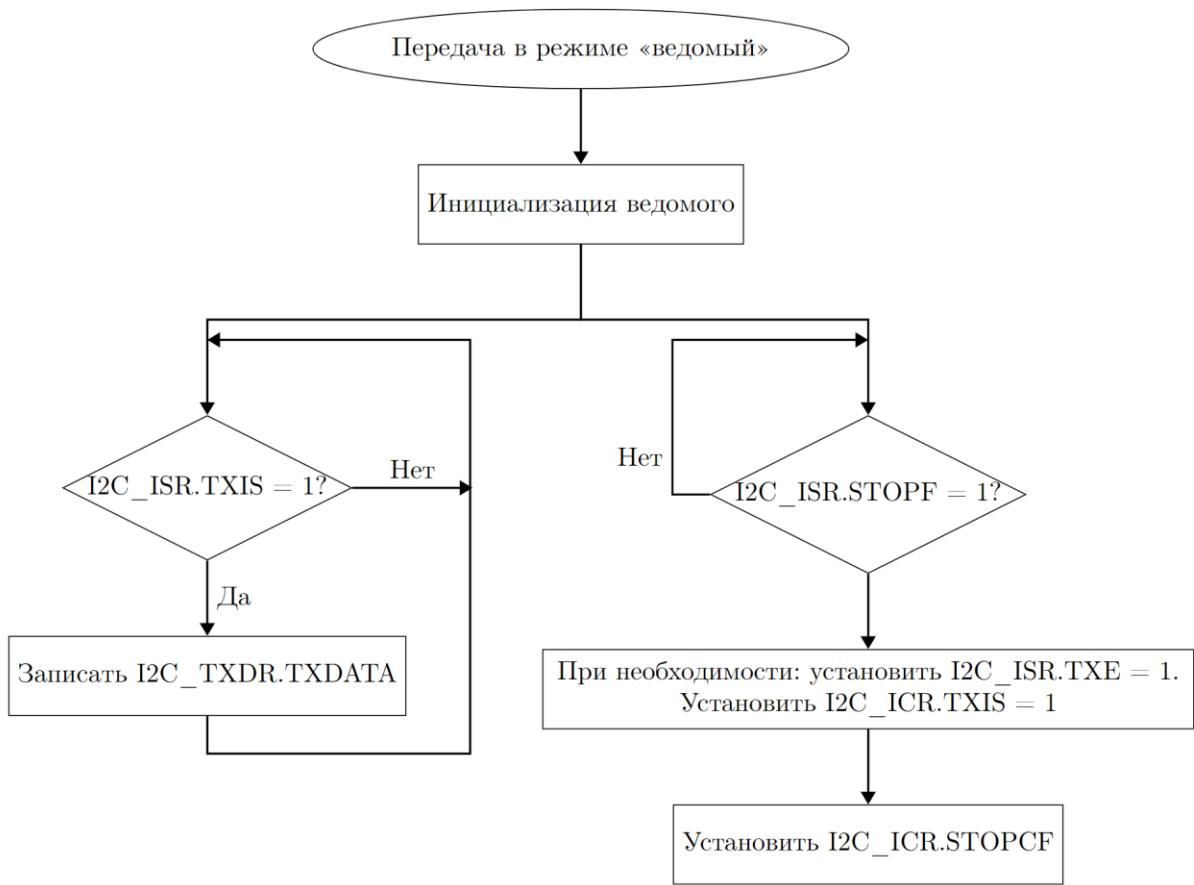
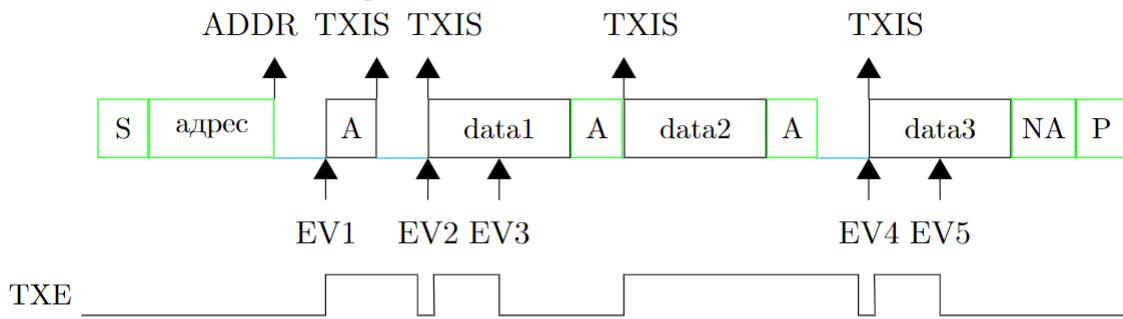


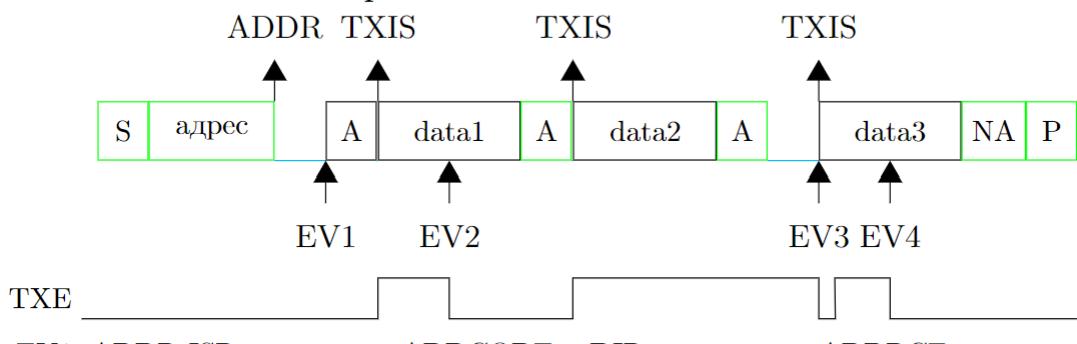
Рисунок 30 - Алгоритм передачи в режиме «ведомый», при
NOSTRETCH = 1

**Пример передачи ведомым 3 байт со сбросом первого байта
при NOSTRETCH=0:**



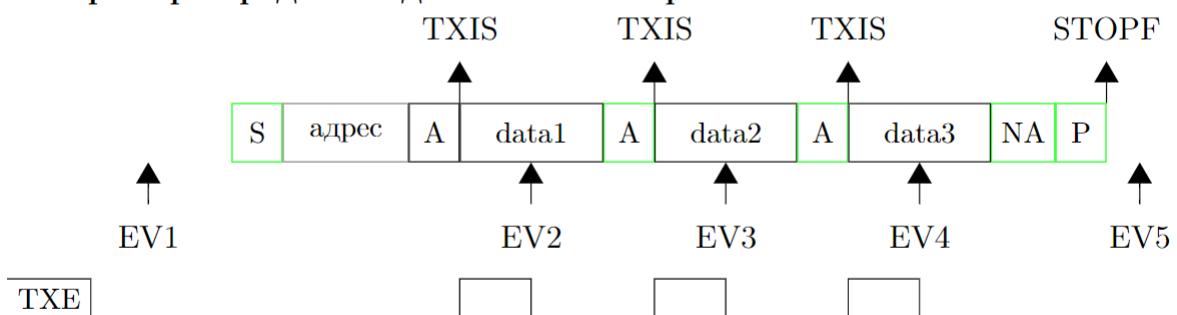
- EV1: ADDR ISR: проверить ADDCODE и DIR, установить TXE и ADDRRCF
 EV2: TXIS ISR: запись data1
 EV3: TXIS ISR: запись data2
 EV4: TXIS ISR: запись data3
 EV5: TXIS ISR: запись data4 (не отправлено)

**Пример передачи ведомым 3 байт без сброса первого байта
при NOSTRETCH=0:**



- EV1: ADDR ISR: проверить ADDCODE и DIR, установить ADDRRCF
 EV2: TXIS ISR: запись data2
 EV3: TXIS ISR: запись data3
 EV4: TXIS ISR: запись data4 (не отправлено)

Пример передачи ведомым 3 байт при NOSTRETCH=1:



- EV1: запись data1
 EV2: TXIS ISR: запись data2
 EV3: TXIS ISR: запись data3
 EV4: TXIS ISR: запись data4 (не отправлено)
 EV5: STOPF ISR: (при необходимости: установить TXE и TXIS), установить STOPCF

Рисунок 31 – Диаграммы состояния шины I²C и события при передаче данных в режиме «ведомый»

3.7.10.6 Приём в режиме «ведомый»

На рисунках 32, 33 приведены алгоритмы приёма в режиме «ведомый», при различных значениях бита NOSTRETCH.

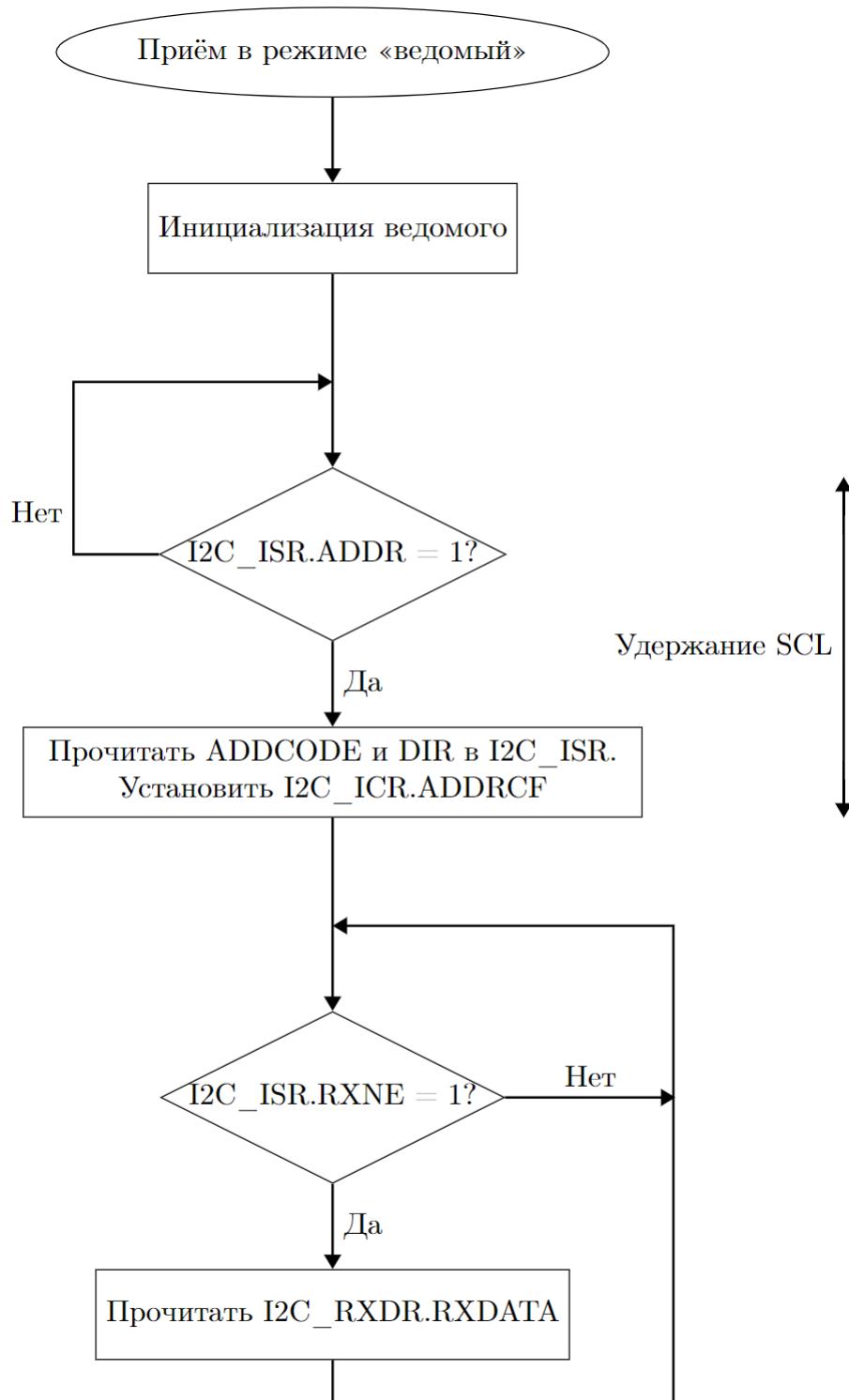


Рисунок 32 - Алгоритм приёма в режиме «ведомый», при
NOSTRETCH = 0

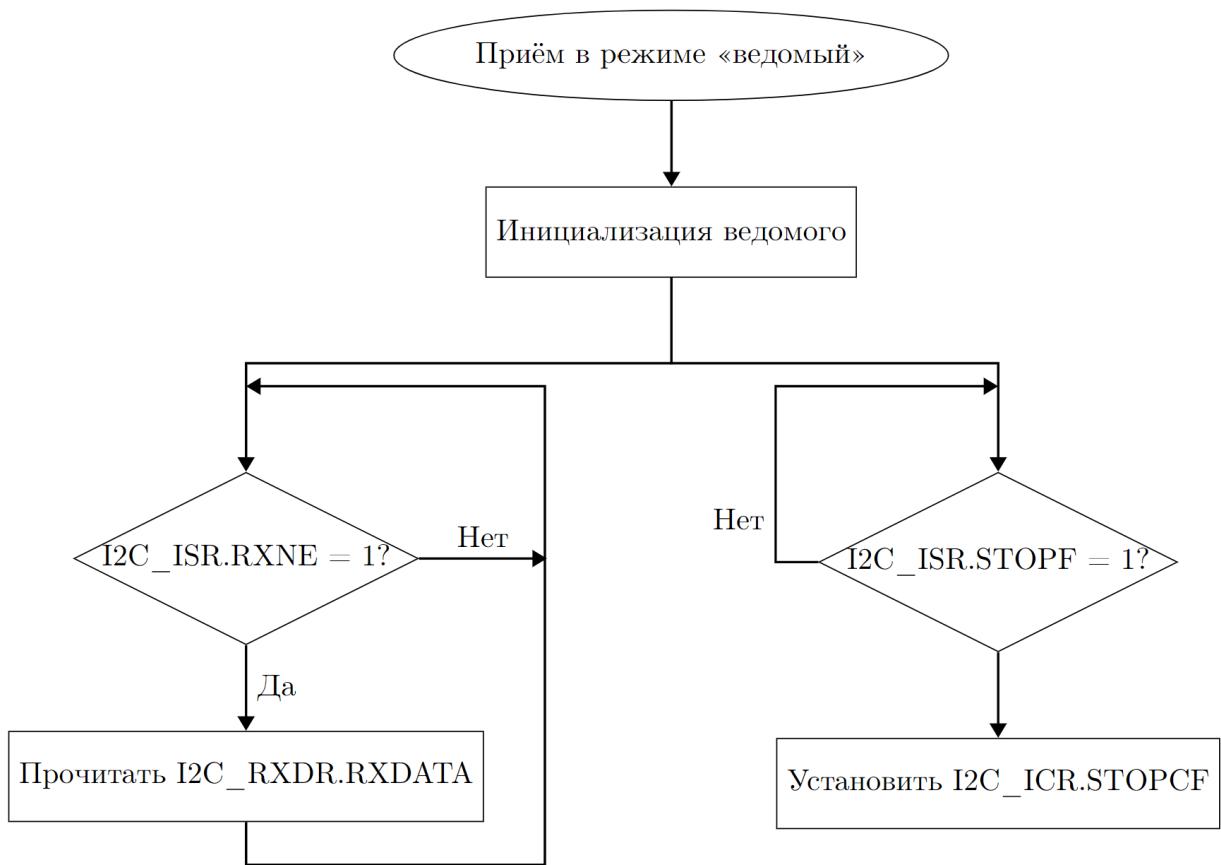
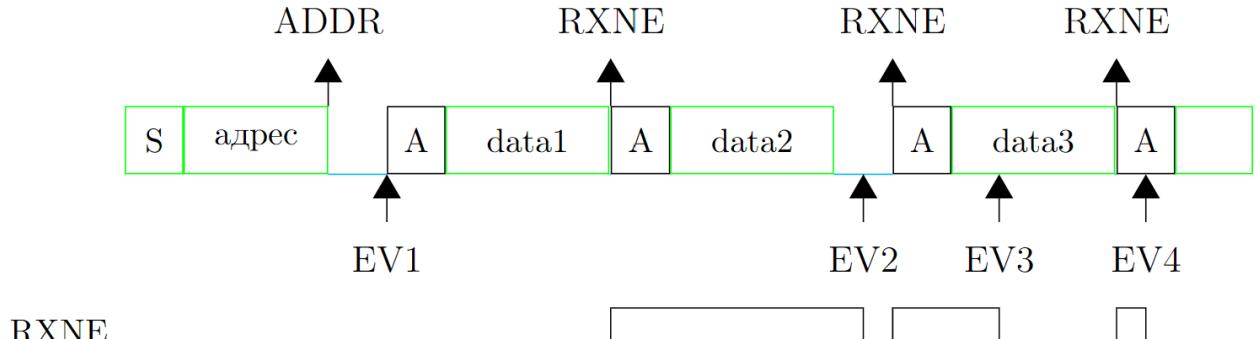


Рисунок 33 - Алгоритм приёма в режиме «ведомый», при
NOSTRETCH = 1

Когда буфер RXDR заполнен, вводится флаг RXNE в регистре ISR и вырабатывается прерывание, если оно разрешено битом RXIE в регистре CR1. Сброс флага происходит после чтения из регистра RXDR.

Если получена комбинация STOP и бит STOPIE в регистре CR1 установлен, то вводится флаг STOPF в регистре ISR и вырабатывается прерывание (рисунок 34).

Пример приёма ведомым 3 байт при NOSTRETCH=0:



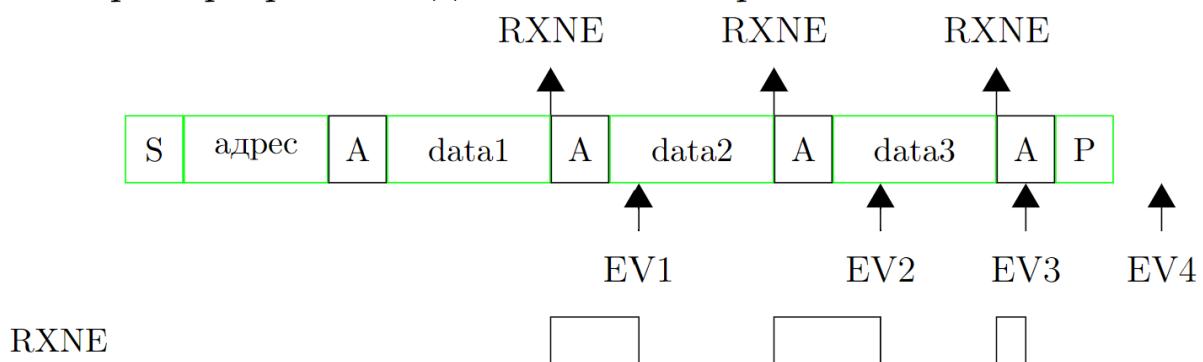
EV1: ADDR ISR: проверить ADDCODE и DIR, установить ADDRCF

EV2: RXNE ISR: прочитать data1

EV3: RXNE ISR: прочитать data2

EV4: RXNE ISR: прочитать data3

Пример приёма ведомым 3 байт при NOSTRETCH=1:



EV1: RXNE ISR: запись data1

EV2: RXNE ISR: запись data2

EV3: RXNE ISR: запись data3

EV4: STOPF ISR: установить STOPCF

Рисунок 34 – Диаграммы состояния шины I²C и события при приёме данных в режиме «ведомый»

3.7.11 Режим «ведущий» I²C

3.7.11.1 Инициализация ведущего

До включения интерфейса должен быть настроен тактовый сигнал SCL битами SCLH и SCLL в регистре TIMINGR.

Для поддержки топологии шины с несколькими ведущими и удержания тактового сигнала (далее ТС) ведомым реализован механизм синхронизации ТС:

- длительность низкого уровня ТС отсчитывается счетчиком SCLL, начиная с момента внутреннего обнаружения низкого уровня SCL;
- длительность высокого уровня ТС отсчитывается счетчиком SCLH, начиная с момента внутреннего обнаружения высокого уровня SCL.

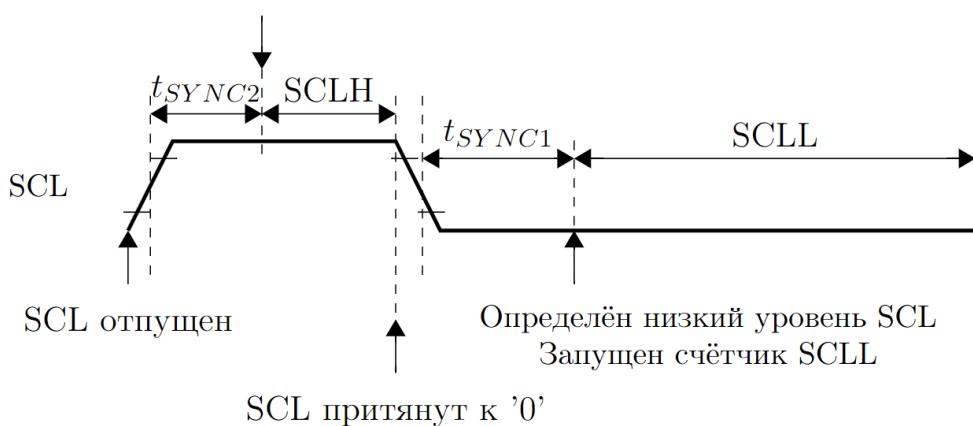
Интерфейс обнаруживает низкий уровень SCL с задержкой t_{SYNC1} , зависящей от длительности спада SCL, характеристик и настроек фильтров шумов (аналогового до 260 нс и цифрового – DNF $\times t_{I2CCLK}$), и синхронизации SCL в домене I²CCLK (2-3 периода I²CCLK). Интерфейс отпускает SCL в высокий уровень, когда счетчик SCLL достигает значения заданного в битах SCLL[7:0] в регистре TIMINGR.

Интерфейс обнаруживает высокий уровень SCL с задержкой t_{SYNC2} , зависящей от длительности фронта SCL, характеристик и настроек фильтра шумов (цифрового – DNF $\times t_{I2CCLK}$), и синхронизации SCL в домене I²CCLK (2-3 периода I²CCLK). Интерфейс притягивает SCL к низкому уровню, когда счетчик SCLH достигает значения заданного в битах SCLH[7:0] в регистре TIMINGR (рисунок 35).

Соответственно, период SCL определяется по формуле (10):

$$t_{SCL} = t_{SYNC1} + t_{SYNC2} + \{[(SCLH+1) + (SCLL+1)] \times (PRESC+1) \times t_{I2CCLK}\} \quad (10)$$

Определён высокий уровень SCL
Запущен счётчик SCLH



Синхронизация SCL в режиме "ведущий"

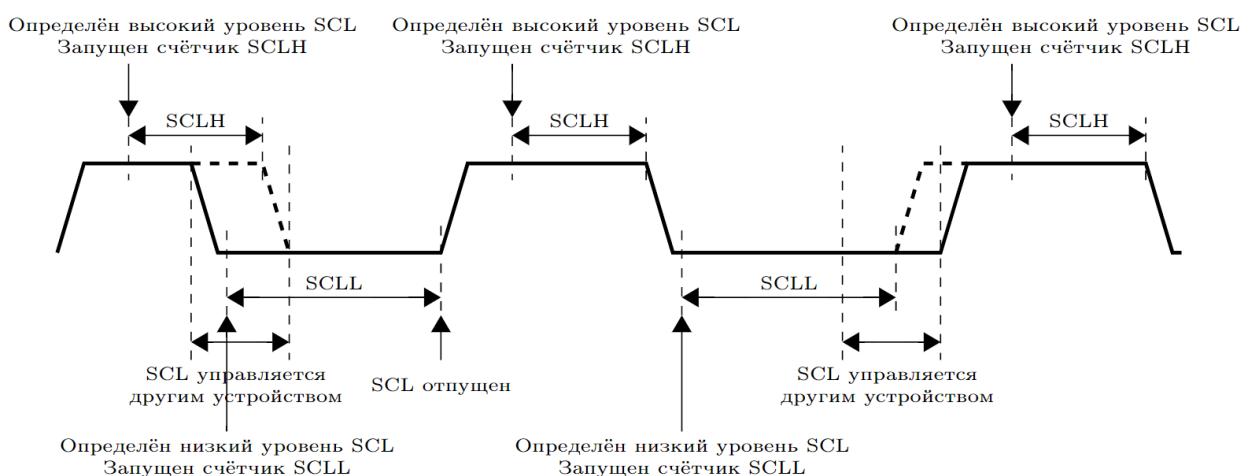


Рисунок 35 – Формирование тактового сигнала SCL в режиме «ведущий»

Таблица 28 - Спецификация временных ограничений шины I2C

Обозначение	Параметр	Standard-mode (Sm)		Fast-mode (Fm)		Fast-mode Plus (Fm+)		Ед. изм.
		Мин.	Макс.	Мин.	Макс.	Мин.	Макс.	
f_{SCL}	Частота SCL	—	100	—	400	—	1000	кГц
$t_{HD;STA}$	Длительность удержания (повторного) состояния START	4,0	—	0,6	—	0,26	—	мкс.
$t_{SU;STA}$	Длительность предустановки состояния START	4,7	—	0,6	—	0,26	—	мкс.

Обозначение	Параметр	Standard-mode (Sm)		Fast-mode (Fm)		Fast-mode Plus (Fm+)		Ед. изм.
		Мин.	Макс.	Мин.	Макс.	Мин.	Макс.	
tsu;STO	Длительность предустановки состояния STOP	4,0	—	0,6	—	0,26	—	мкс.
tBUF	Время между состояниями STOP и START	4,7	—	1,3	—	0,5	—	мкс.
tLOW	Длительность паузы SCL	4,7	—	1,3	—	0,5	—	мкс.
tHIGH	Длительность импульса SCL	4,0	—	0,6	—	0,26	—	мкс.
t _r	Длительность фронта сигналов SDA, SCL	—	1000	—	300	—	120	нс.
t _f	Длительность спада сигналов SDA, SCL	—	300	—	300	—	120	нс.

Значение SCLL также используется для генерации задержек tBUF и tSU:STA.
tBUF – время между STOP и START.

Значение SCLH также используется для генерации задержек tHD:STA и tSU:STO.

3.7.11.2 Инициализация соединения ведущим (фаза адреса)

Перед соединением в регистре CR2 должны быть настроены:

- режим адресации (7 или 10 бит) ADD10;
- адрес ведомого SADD[9:0];
- направление передачи RD_WRN;
- в случае чтения из ведомого с 10-битным адресом, бит HEAD10R. Бит HEAD10R обуславливает передачу всего адреса или только заголовка в случае смены направления;
- количество байтов на передачу NBYTES[7:0]. Если количество байтов равно или более 255, в NBYTES[7:0] сначала должно быть записано значение 0xFF.

После этого пользователь устанавливает бит START в регистре CR2. После этого изменение вышеописанных битов не допускается.

Затем после обнаружения свободной шины (BUSY=0) и задержки tBUF, ведущий автоматически отправляет комбинацию START и следом адрес ведомого.

В случае проигрыша арбитража, ведущий автоматически переключается в режим ведомого и может подтвердить свой адрес, если он адресован, как ведомый.

Примечание – Бит START сбрасывается аппаратно, в случае проигрыша арбитража или после того как отправлен адрес ведомого, независимо от значения бита подтверждения (ACK/NACK).

В режиме 10-битного адреса, если первые семь бит адреса ведомого не подтверждаются ведомым (NACK), ведущий автоматически отправляет адрес заново, пока не будет получен ACK. В этом случае, если получен NACK, должен быть установлен флаг ADDRCF для прекращения отправки адреса ведомого.

Если интерфейс адресован, как ведомый (ADDR = 1), пока бит START = 1, то он переключается в режим «ведомый», бит START сбрасывается после установки флага ADDRCF.

Такая же процедура применяется для случая комбинации RESTART. В этом случае BUSY=1.

Алгоритм инициализации I2C в режиме «ведущий» приведен на рисунке 36.

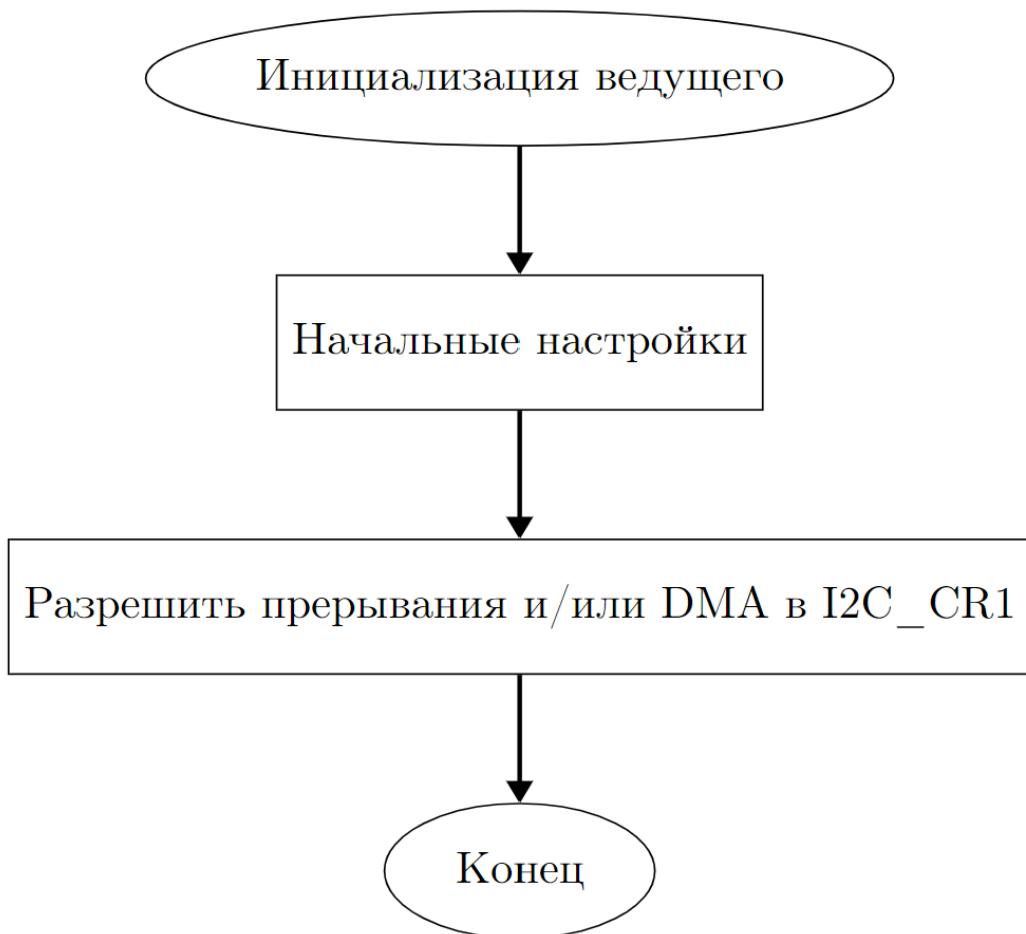


Рисунок 36 – Инициализация в режиме «ведущий»

Инициализация приёма ведущим, адресация ведомого, 10 бит:

- если адрес ведомого 10 бит, то пользователь может отправить полную последовательность чтения, очистив бит HEAD10R в регистре CR2. После установки бита START, ведущий автоматически отправит последовательность: (Re)Start + Заголовок 10-битного адреса ведомого с записью + Второй байт адреса ведомого + ReStart + Заголовок 10-битного адреса ведомого с чтением (рисунок 37);
- если ведущий адресует ведомого с 10-битным адресом, отправляет ему данные и затем читает из этого же ведомого данные, то после завершения процедуры отправки запускается процедура повторного старта с битом HEAD10R=1. В этом случае отправляемая последовательность будет следующей – ReStart + Заголовок 10-битного адреса ведомого с чтением (рисунок 38).

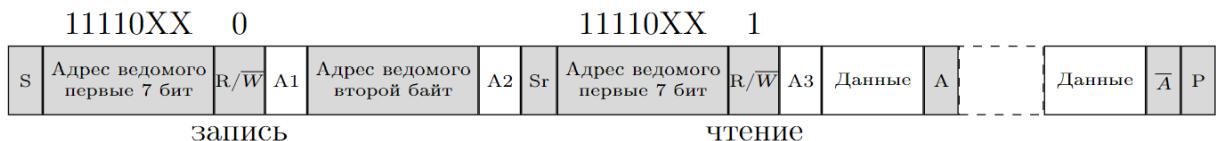


Рисунок 37 – Адресация ведомого 10 бит адреса при HEAD10R = 0



Рисунок 38 – Адресация ведомого 10 бит адреса при HEAD10R = 1

3.7.11.3 Передача в режиме «ведущий»

В случае отправки данных, флаг TXIS взводится после отправки каждого байта после девяти импульса SCL, если получено подтверждение (ACK). Прерывание вырабатывается, если установлен бит TXIE в регистре CR1. Флаг сбрасывается после записи следующего байта данных в регистр TXDR.

Количество событий TXIS соответствует количеству, записанному в NBYTES[7:0]. Если необходимо отправить более 255 байт, должен быть установлен бит RELOAD в регистре CR2. В этом случае, после отправки NBYTES данных, взводится

флаг TCR и линия SCL удерживается в низком уровне пока в NBYTES[7:0] не будет записано ненулевое значение.

На рисунках 39, 40 приведены алгоритмы передачи в режиме «ведущий», при разном количестве передаваемых байт.

Флаг TXIS не взводится, если не получено подтверждение (NACK).

Если RELOAD = 0, данные передаются:

- с автозавершением (AUTOEND = 1), комбинация STOP отправляется автоматически;
- с завершением программой (AUTOEND = 0), флаг TC взводится, и линия SCL удерживается для совершения программой следующих действий:
 - 1) может быть запрошена комбинация RESTART установкой бита START в регистре CR2 с корректной настройкой адреса ведомого и количеством байт для передачи. Установка бита START сбрасывает флаг TC, и комбинация START передаётся по шине;
 - 2) может быть запрошена комбинация STOP установкой бита STOP в регистре CR2. Установка бита STOP сбрасывает флаг TC, и комбинация STOP отправляется по шине.

Если не получено подтверждение (NACK): флаг TXIE не взводится, и комбинация STOP автоматически отправляется после получения NACK. Взводится флаг NACKF в регистре ISR и вырабатывается прерывание, если установлен бит NACKIE (рисунок 41).

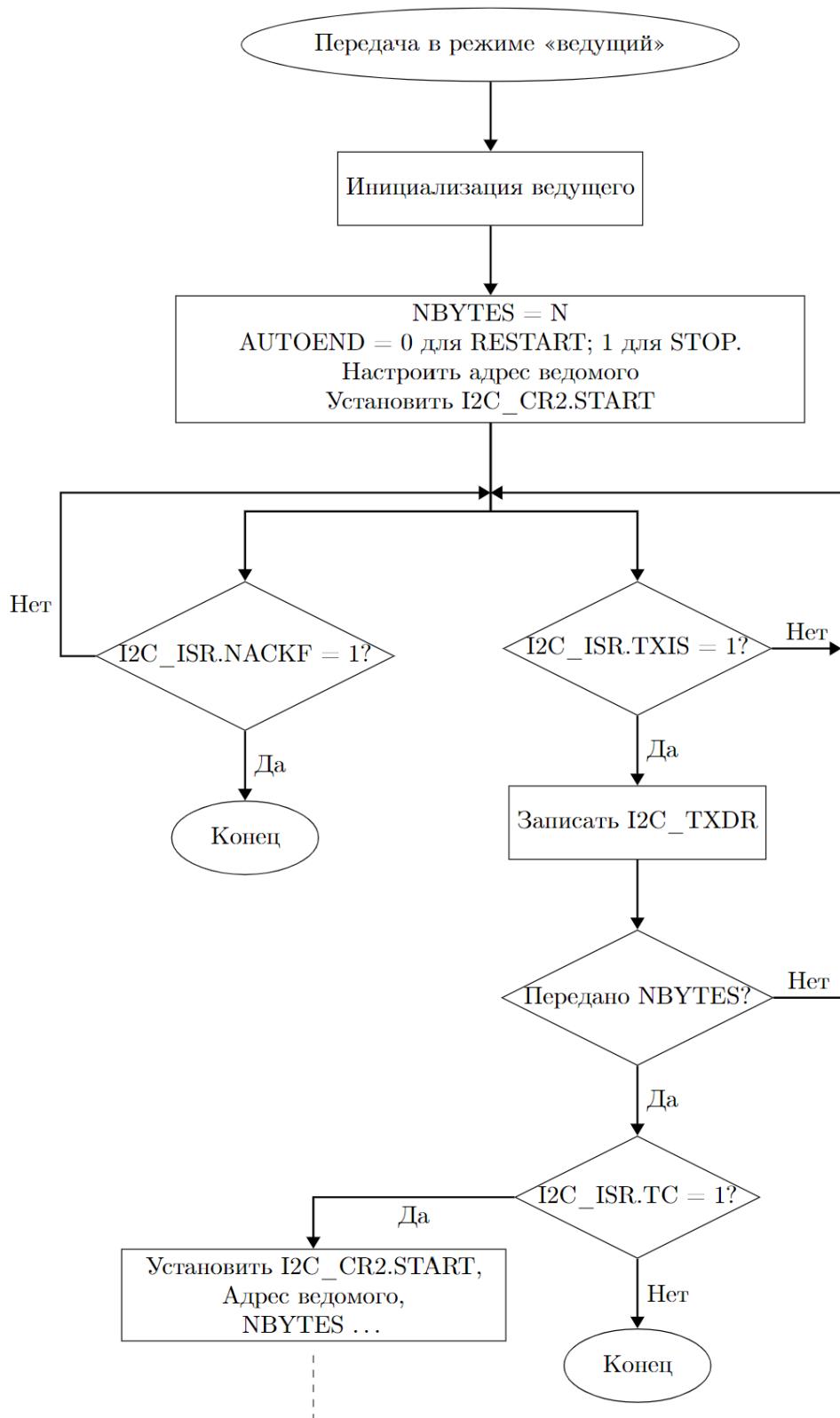


Рисунок 39 - Передача в режиме «ведущий» при $N \leq 255$ байт

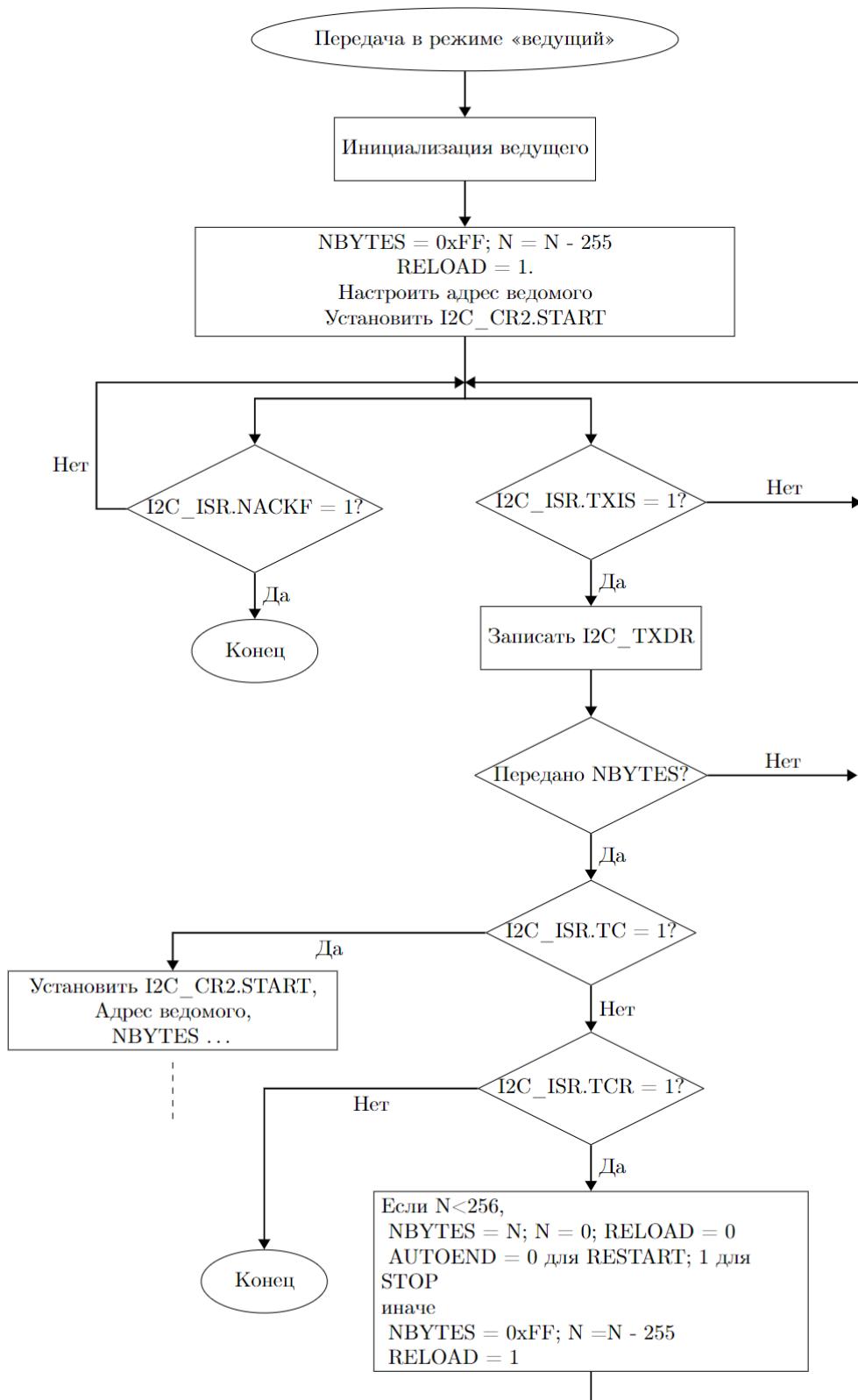
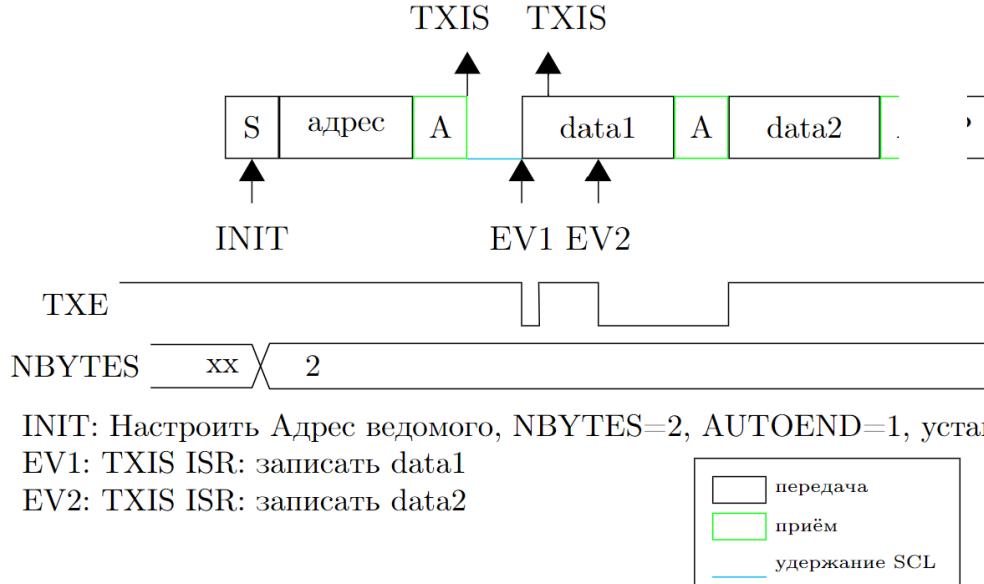


Рисунок 40 - Передача в режиме «ведущий» при $N > 255$ байт

Пример передачи ведущим 2 байт с автозавершением (STOP):



Пример передачи ведущим 2 байт с программным завершением (RESTART):

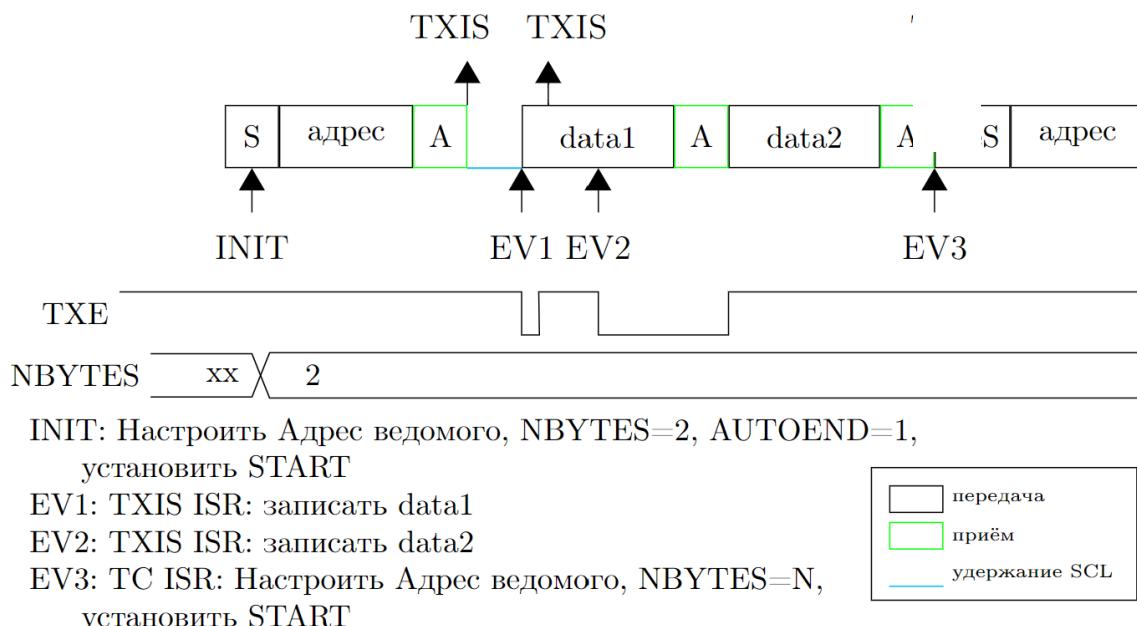


Рисунок 41 – Передача данных ведущим

3.7.11.4 Приём в режиме «ведущий»

В случае приёма данных, флаг RXNE взводится после приёма каждого байта после восьмого импульса SCL. Событие RXNE вырабатывает прерывание, если установлен бит RXIE в регистре CR1. Флаг сбрасывается после чтения из регистра RXDR.

Если общее количество байт для приёма больше 255, устанавливается бит RELOAD в регистре CR2. В этом случае после передачи всех NBYTES данных,

взводится флаг TCR и линия SCL удерживается пока в NBYTE[7:0] не будет записано ненулевое значение.

Если RELOAD = 0, и передаются NBYTE[7:0] байт данных:

- с автозавершением (AUTOEND = 1), NACK и комбинация STOP отправляются автоматически после последнего полученного байта;
- с завершением программой (AUTOEND = 0), NACK отправляется автоматически после последнего полученного байта, взводится флаг TC и линия SCL удерживается для совершения программой следующий действий (рисунок 44):

1) может быть запрошена комбинация RESTART установкой бита START в регистре CR2 с корректной настройкой адреса ведомого и количеством байт для передачи. Установка бита START сбрасывает флаг TC, и комбинация START и адрес ведомого передаются по шине;

2) может быть запрошена комбинация STOP установкой бита STOP в регистре CR2. Установка бита STOP сбрасывает флаг TC, и комбинация STOP передаётся по шине.

На рисунках 42, 43 приведены алгоритмы приема в режиме «ведущий», при разном количестве передаваемых байт.

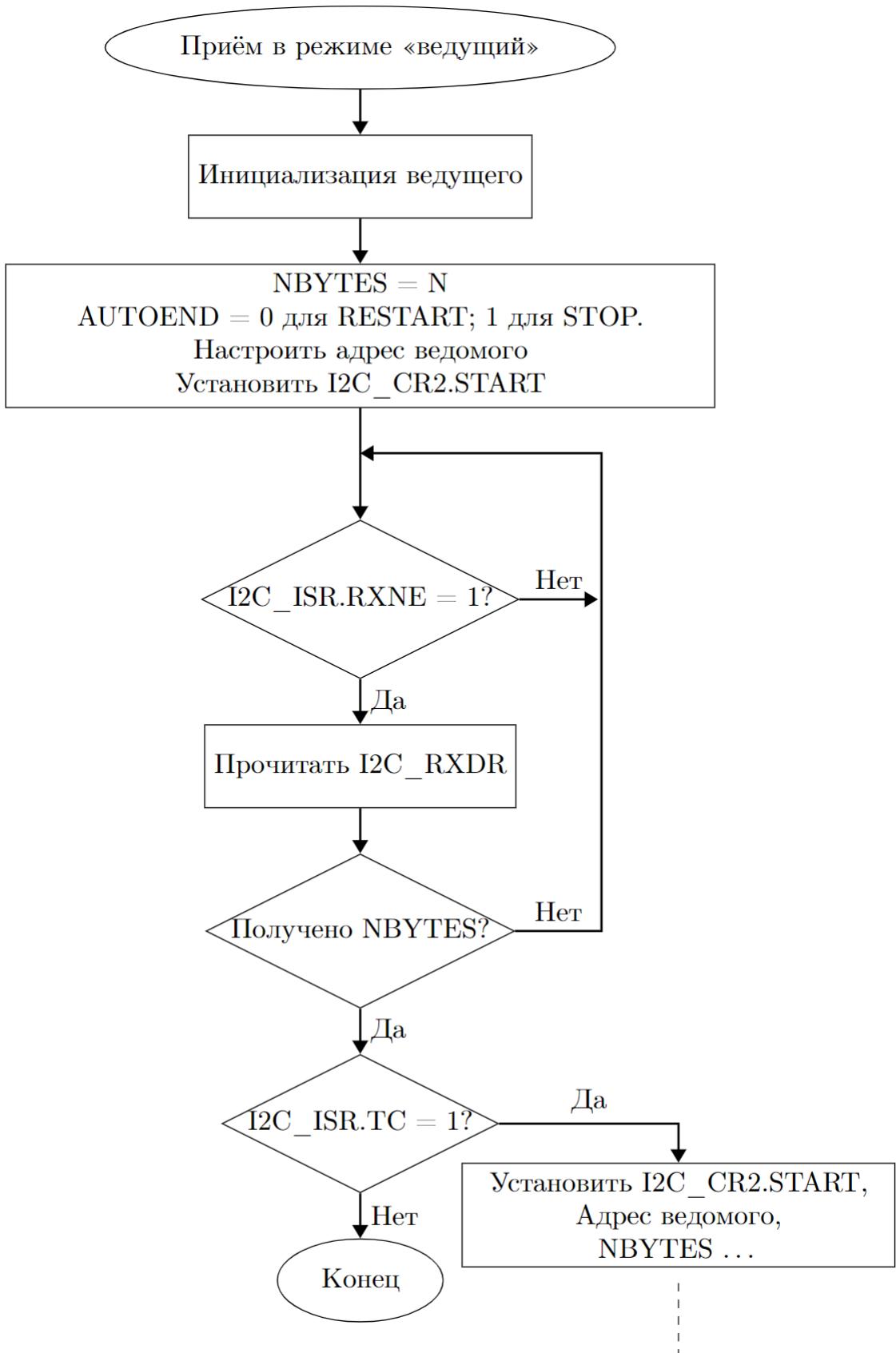


Рисунок 42 - Приём в режиме «ведущий» при $N \leq 255$ байт

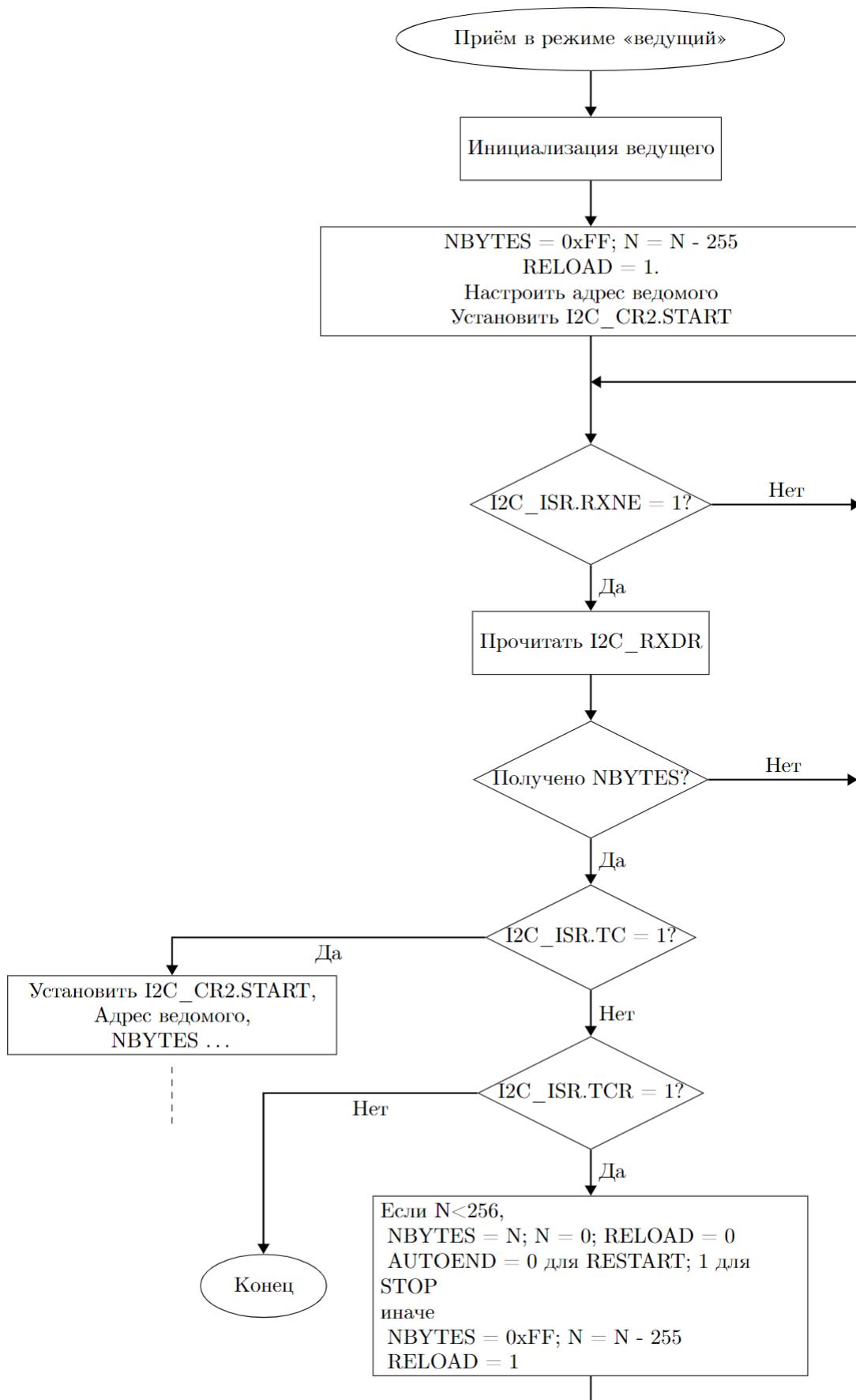
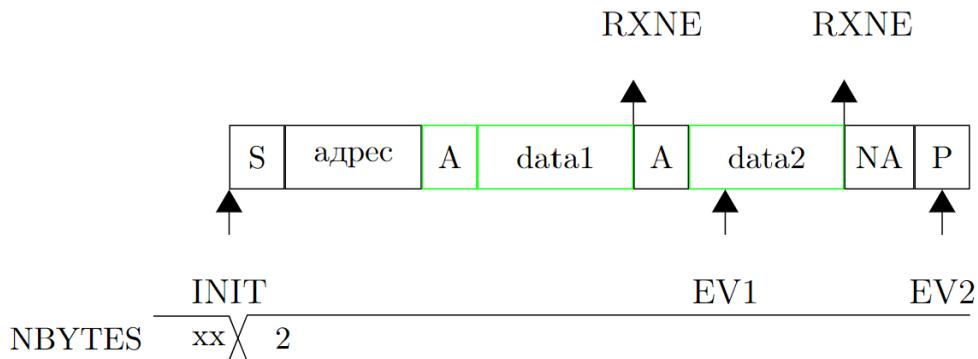


Рисунок 43 - Приём в режиме «ведущий» при N>255 байт

Пример приёма ведущим 2 байт с автозавершением (STOP):



INIT: Настроить Адрес ведомого, NBYTES=2, AUTOEND=1,
установить START

EV1: RXNE ISR: прочитать data1

EV2: RXNE ISR: прочитать data2

Пример приёма ведущим 2 байт с программным завершением
(RESTART):

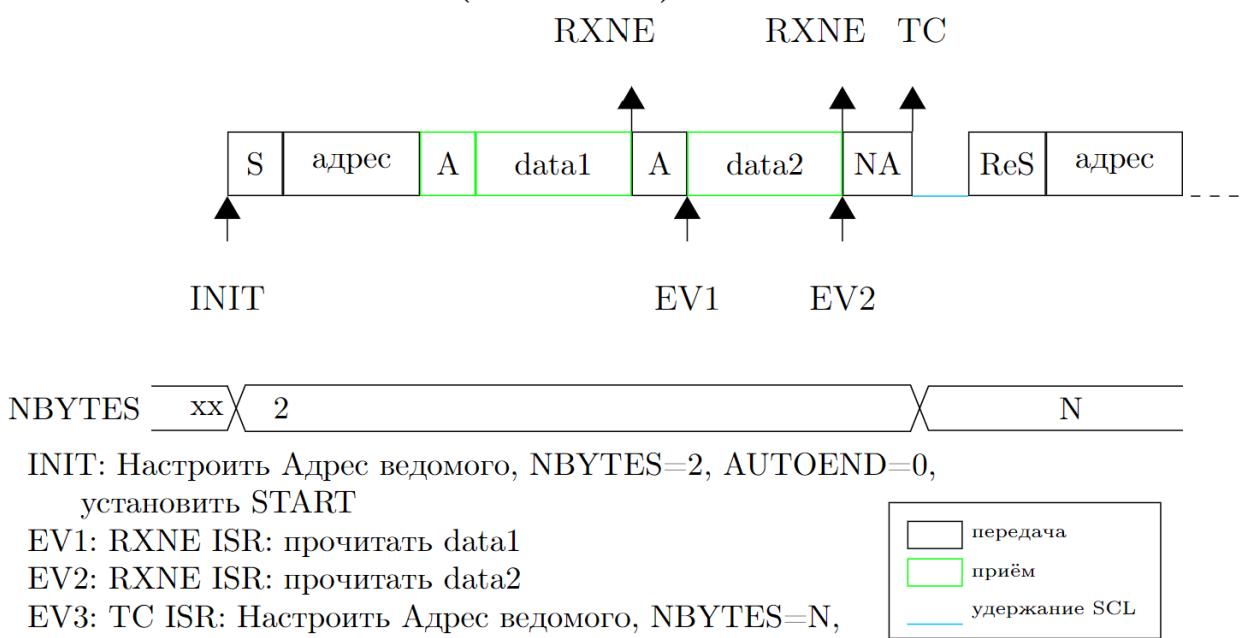


Рисунок 44 – Приём данных ведущим

3.7.12 Ошибки

3.7.12.1 Ошибка шины (BERR)

Ситуация, в которой после возникновения на шине комбинации START или STOP через девять тактов SCL не произошло изменений на шине, определяется как Ошибка

шины. Комбинациями START или STOP является изменение уровня SDA, пока SCL находится в состоянии высокого уровня.

Флаг ошибки шины взводится, только если интерфейс задействован для передачи в качестве ведущего или адресованного ведомого (например, не во время фазы адреса в режиме ведомого).

В случае ошибочного распознавания на шине START или RESTART в режиме ведомого, интерфейс переходит в стадию определения адреса, как в случае корректного события START.

При возникновении ошибки шины, взводится флаг BERR в регистре ISR и вырабатывается прерывание, если в регистре CR1 установлен бит ERRIE.

3.7.12.2 Проигрыш арбитража (ARLO)

Арбитраж шины считается проигранным, когда на линию SDA выставляется высокий уровень, а по фронту SCL на ней регистрируется низкий уровень:

- в режиме «ведущий» проигрыш арбитража определяется в фазе адреса, в фазе данных и фазе подтверждения данных. В этом случае линии SDA и SCL освобождаются, бит START сбрасывается аппаратно и ведущий переключается в режим «ведомый»;
- в режиме «ведомый» проигрыш арбитража определяется в фазе данных и фазе подтверждения данных. В этом случае передача останавливается, и линии SDA и SCL освобождаются.

Если зарегистрирован проигрыш арбитража, в регистре ISR взводится флаг ARLO и вырабатывается прерывание, если в регистре CR1 установлен бит ERRIE.

3.7.12.3 Ошибка переполнения/недозагрузка (OVR)

Ошибка OVR обнаруживается в режиме «ведомый» при NOSTRETCH=1 и если:

- во время приёма получен новый байт, а RXDR всё ещё не прочитан. Новый полученный байт игнорируется и автоматически отправляется NACK в качестве ответа на него;
- во время отправки:
 - 1) при STOPF = 1 и первый байт данных должен быть отправлен.

Отправляется содержимое регистра TXDR, если TXE = 0, иначе 0xFF;

- 2) новый байт должен быть отправлен и регистр TXDR всё ещё не записан, отправляется 0xFF.

В случае ошибки OVR в регистре ISR взводится флаг OVR и вырабатывается прерывание, если бит ERRIE установлен в регистре CR1.

3.7.13 DMA запросы

3.7.13.1 Передача с использованием DMA

Для передачи данных с использованием DMA (Direct Memory Access) необходимо установить бит TXDMAEN в регистре CR1. Данные из области памяти, настроенной для использования контроллером DMA, загружаются в регистр TXDR каждый раз, когда взводится флаг TXIS.

С использованием DMA только передаются данные:

- в режиме «ведущий» инициализация, настройка адреса ведомого, направление передачи, количество байт и бит START записываются программой. В случае передачи всех данных с использованием DMA, он должен быть настроен до установки бита START. Окончание передачи определяется счетчиком NBYTES;
- в режиме «ведомый»:
 - 1) если NOSTRETCH=0 и все данные передаются с использованием DMA, он должен быть настроен до момента совпадения адреса на шине с внутренним адресом или в обработчике прерывания ADDR до сброса флага ADDR;
 - 2) если NOSTRETCH=1, то DMA должен быть настроен до момента совпадения адреса на шине с внутренним адресом.

Примечание. При использовании DMA для передачи данных необязательно разрешать прерывание битом TXIE.

3.7.13.2 Приём с использованием DMA

Для приёма данных с использованием DMA (Direct Memory Access) необходимо установить бит RXDMAEN в регистре CR1. Данные из регистра RXDR, загружаются в область памяти, настроенной для использования DMA перipherией каждый раз, когда взводится флаг RXNE.

С использованием DMA только передаются данные:

- в режиме «ведущий» инициализация, настройка адреса ведомого, направление передачи, количество байт и бит START записываются программой. В случае передачи всех данных с использованием DMA, он должен быть настроен до установки бита START. Окончание передачи определяется счетчиком NBYTES;

– в режиме ведомого, если NOSTRETCH=0 и все данные передаются с использованием DMA, он должен быть настроен до момента совпадения адреса нашине с внутренним адресом или в обработчике прерывания ADDR до сброса флага ADDR.

Примечание – При использовании DMA для приёма данных необязательно разрешать прерывание битом RXIE.

3.7.14 Режимы пониженного энергопотребления I²C

Режим пониженного энергопотребления I²C представлен в таблице 29.

Таблица 29 – Режим пониженного энергопотребления I²C

Режим	Описание
Сон	Без эффекта Прерывание I ² C выводит устройство из сна
Стоп	Данные в регистрах I ² C сохраняются
Ожидание	Периферия I ² C отключается и должна быть инициализирована заново после выхода из режима

3.7.15 Прерывания I²C

Прерывания I²C представлены в таблице 30.

Таблица 30 – Прерывания I²C

Событие прерывания	Флаг события	Метод очистки	Контрольный бит
Буфер приёма заполнен	RXNE	Чтение регистра RXDR	RXIE
Статус буфера передачи	TXIS	Запись регистра TXDR	TXIE
Обнаружение STOP на линии	STOPF	Запись STOPCF=1	STOPIE
Конец передачи RELOAD=1	TCR	Запись CR2 с NBYTES[7:0]≠0	TCIE
Конец передачи	TC	Запись START=1 или STOP=1	
Принятый адрес совпал с собственным адресом OA1, OA2	ADDR	Запись ADDRCF=1	ADDRIE
Не получено подтверждение (NACK)	NACKF	Запись NACKCF=1	NACKIE

Событие прерывания	Флаг события	Метод очистки	Контрольный бит
Ошибка шины	BERR	Запись BERRCF=1	ERRIE
Проигрыш арбитража	ARLO	Запись ARLOCF=1	
Перебор/недобор	OVR	Запись OVRCF=1	

Чтобы включить прерывания I²C необходимо:

- настроить и активировать канал I²C в NVIC;
- настроить I²C для вырабатывания прерываний.

На рисунке 45 изображена схема формирования прерываний I²C.

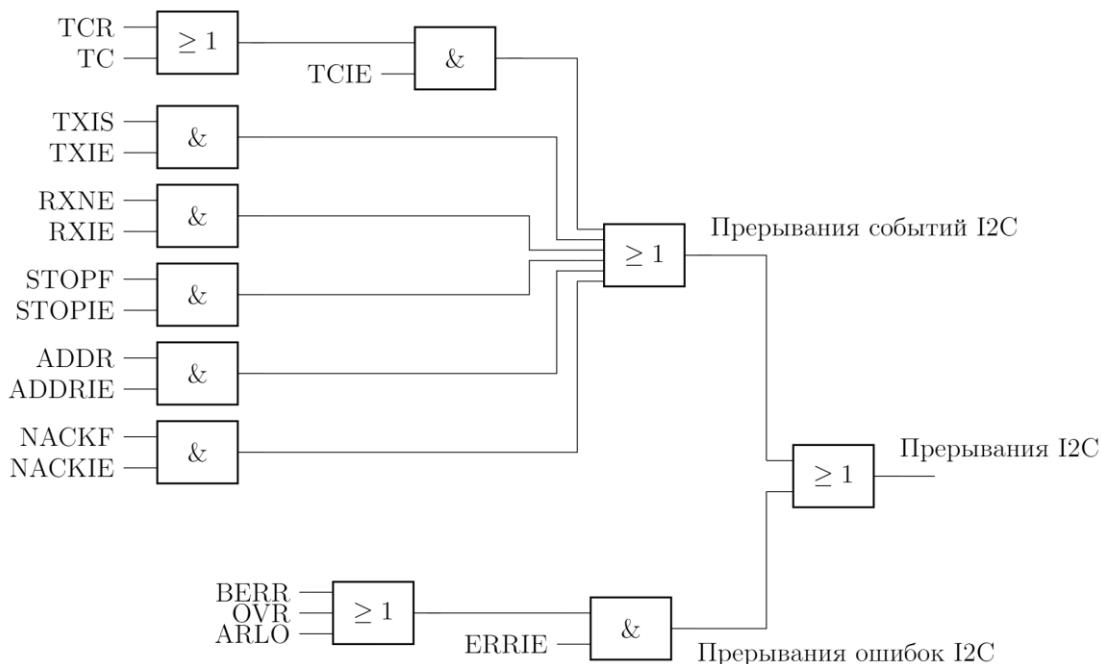


Рисунок 45 – Схема формирования прерываний

3.7.16 Регистры I²C

Регистры модуля I²C представлены в таблице 31.

Таблица 31 – Регистры модуля I²C

Обозначение	Смещение	Начальное значение	Регистр	Доступ
CR1	0x00	0x0	Регистр управления 1	RW
CR2	0x04	0x0	Регистр управления 2	RW
OAR1	0x08	0x0	Регистр адреса 1	RW
OAR2	0x0C	0x0	Регистр адреса 2	RW
TIMINGR	0x10	0x0	Регистр настройки временных ограничений	RW
Зарезервировано	0x14	0x0	Зарезервировано	RO
ISR	0x18	0x0	Регистр флагов прерываний	RW
ICR	0x1C	0x0	Регистр сброса флагов прерываний	RW
RXDR	0x20	0x0	Регистр принятых данных	RW
TXDR	0x24	0x0	Регистр передаваемых данных	RW

Доступ к регистрам осуществляется словами (32 бита). Поля регистров представлены в таблица 32 – 40.

3.7.16.1 Регистр управления 1 (CR1)

В таблице 32 приведены поля регистра управления CR1.

Таблица 32 – Регистр управления 1 (CR1)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:20		Зарезервировано	
GCEN	19	0x0	Разрешение адреса общего вызова: 0 – Адрес 0b00000000 запрещен, формируется NACK; 1 – Адрес 0b00000000 разрешен, формируется ACK	RW
Зарезервировано	18		Зарезервировано	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
NOSTRETCH	17	0x0	Отключение растягивания тактового сигнала в режиме «ведомый»: 0 – растягивание активно; 1 – растягивание выключено Изменение значения допускается только при выключенном блоке (PE = 0). В режиме «ведущий» бит должен быть установлен в «0»	RW
SBC	16	0x0	Режим аппаратного контроля передачи данных в режиме «ведомый»: 0 – контроль выключен; 1 – контроль включен	RW
RXDMAEN	15	0x0	Режим поддержки DMA при приеме данных: 0 – поддержка DMA выключена 1 – поддержка DMA включена	RW
TXDMAEN	14	0x0	Режим поддержки DMA при передаче данных: 0 – поддержка DMA выключена 1 – поддержка DMA включена	RW
Зарезервировано	13		Зарезервировано	
ANFOFF	12	0x0	Управление аналоговым фильтром шумов: 0 – фильтр включен 1 – фильтр выключен	RW
DNF	11:8	0x0	Управление цифровым фильтром шумов: 0b0000 – цифровой фильтр выключен; 0b0001 – цифровой фильтр установлен на 1 такт I ² CCLK; 0b0010 – цифровой фильтр установлен на 2 такта I ² CCLK; ... 0b1111 – цифровой фильтр установлен на 15 тактов I ² CCLK Изменение значения допускается только при выключенном блоке (PE=0)	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
ERRIE	7	0x0	Разрешение прерывания при ошибке События, вызывающие прерывание: потеря арбитража (ARLO); ошибка шины (BERR); переполнение/недозагрузка (OVR). 0 – прерывание запрещено; 1 – прерывание разрешено	RW
TCIE	6	0x0	Разрешение прерывания окончания передачи События, вызывающие прерывание: окончание передачи (TC); окончание передачи при RELOAD=1 (TCR). 0 – прерывание запрещено; 1 – прерывание разрешено	RW
STOPIE	5	0x0	Разрешение прерывания обнаружения STOP на линии: 0 – прерывание запрещено; 1 – прерывание разрешено	RW
NACKIE	4	0x0	Разрешение прерывания прием NACK: 0 – прерывание запрещено; 1 – прерывание разрешено	RW
ADDRIE	3	0x0	Разрешение прерывания соответствия адреса в режиме «ведомый»: 0 – прерывание запрещено; 1 – прерывание разрешено	RW
RXIE	2	0x0	Разрешение прерывания при приеме: 0 – прерывание запрещено; 1 – прерывание разрешено	RW
TXIE	1	0x0	Разрешение прерывания при передаче: 0 – прерывание запрещено; 1 – прерывание разрешено	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
PE	0	0x0	Управление интерфейсом: 0 – интерфейс выключен; 1 – интерфейс включен После очистки, бит должен оставаться в «0» минимум три периода тактового сигнала APB	RW

3.7.16.2 Регистр управления 2 (CR2)

В таблице 33 приведены поля регистра управления CR2.

Таблица 33 – Регистр управления 2 (CR2)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:26		Зарезервировано	
AUTOEND	25	0x0	Управление режимом автоматического окончания в режиме «ведущий»: 0 – режим автоматического окончания отключен; 1 – режим автоматического окончания включен Бит устанавливается и очищается программой. Не имеет значения при установленном бите RELOAD в режиме «ведомый»	RW
RELOAD	24	0x0	Управление режимом перезаписи NBYTES: 0 – транзакция завершена после пересылки NBYTES байт данных (на шине ожидаются STOP или RESTART); 1 – транзакция не завершена после пересылки NBYTES байт данных (значение NBYTES будет перезаписано) Бит устанавливается и очищается программой.	RW
NBYTES	23:16	0x0	Количество байт для приема / передачи Не имеет значения в режиме «ведомый» при SBC=0 Не допускается изменение при установленном бите START	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
NACK	15	0x0	<p>Формирование NACK в режиме «ведомый». Устанавливается программно, сбрасывается аппаратно:</p> <p>после отправки NACK;</p> <p>по событию STOP на шине;</p> <p>при получении своего адреса ведомого;</p> <p>PE = 0.</p> <p>0 – отправка ACK после приёма текущего байта;</p> <p>1 – отправка NACK после приёма текущего байта</p>	RW1S
STOP	14	0x0	<p>Формирование STOP в режиме «ведущий». Устанавливается программно, сбрасывается аппаратно по событию STOP на шине или при PE = 0:</p> <p>0 – STOP не формируется;</p> <p>1 – формирование STOP после передачи текущего байта.</p> <p>Запись «0» в этот бит не имеет эффекта.</p>	RW1S
START	13	0x0	<p>Формирование START.</p> <p>Устанавливается программно, сбрасывается аппаратно после отправки адреса, потеряв арбитража или PE = 0:</p> <p>0 – START не формируется;</p> <p>1 – формирование START разрешено</p> <p>Запись «0» в этот бит не имеет эффекта.</p> <p>В режиме «ведущий» отправка первой части 10-битного адреса повторяется при получении NACK до получения ACK или до очистки бита START записью в бит ADDRCF</p>	RW1S

Обозначение	Разряды	Начальное значение	Назначение	Доступ
HEAD10R	12	0x0	<p>Поддержка 10-битного адреса в режиме «ведущий», чтение:</p> <p>0 – ведущий отправляет полную последовательность для чтения для 10 битного адреса: Start + 2 байта адреса (запись) + ReStart + заголовок 10-битного адреса (чтение);</p> <p>1 – ведущий отправляет только заголовок 10-битного адреса (чтение).</p> <p>Изменение этого бита при START=1 не допускается.</p>	RW
ADD10	11	0x0	<p>Режим 10-битного адреса (режим «ведущий»):</p> <p>0 – ведущий работает в режиме 7-битного адреса;</p> <p>1 – ведущий работает в режиме 10-битного адреса.</p> <p>Изменение этого бита при START = 1 не допускается.</p>	RW
RD_WRN	10	0x0	<p>Направление передачи (режим «ведущий»):</p> <p>0 – ведущий запрашивает транзакцию записи;</p> <p>1 – ведущий запрашивает транзакцию чтения</p> <p>Изменение этого бита при START = 1 не допускается.</p>	RW
SADD	9:8	0x0	<p>Адрес ведомого (режим «ведущий»):</p> <p>В режиме 7-битного адреса (ADD10 = 0) содержимое этих битов не имеет значения.</p> <p>В режиме 10-битного адреса (ADD10 = 1) в эти биты должны быть записаны биты 9:8 адреса ведомого.</p> <p>Изменение этих битов при START = 1 не допускается.</p>	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
SADD	7:1	0x0	Адрес ведомого (режим «ведущий»): В режиме 7-битного адреса (ADD10 = 0) в эти биты должны быть записаны 7 бит адреса ведомого. В режиме 10-битного адреса (ADD10 = 1) в эти биты должны быть записаны биты 7:1 адреса ведомого. Изменение этих битов при START = 1 не допускается.	RW
SADD	0	0x0	Адрес ведомого (режим «ведущий»): В режиме 7-битного адреса (ADD10=0) содержимое бита не имеет значения. В режиме 10-битного адреса (ADD10=1) в этот бит должен быть записан бит 0 адреса ведомого. Изменение этого бита при START=1 не допускается.	RW

3.7.16.3 Регистр собственного адреса 1 (OAR1)

В таблице 34 приведены поля регистра OAR1.

Таблица 34 – Регистр собственного адреса 1 (OAR1)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:16		Зарезервировано	
OA1EN	15	0x0	Использование собственного адреса OA1: 0 – при получении адреса OA1 формируется NACK; 1 – при получении адреса OA1 формируется ACK	RW
Зарезервировано	14:11		Зарезервировано	
OA1MODE	10	0x0	Режим 10-битного адреса OA1: 0 – OA1 – 7-битный адрес; 1 – OA1 – 10-битный адрес Изменение бита допускается при OA1EN=0	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
OA1	9:8	0x0	Собственный адрес 1 Режим 7-битного адреса: не имеют значения. Режим 10-битного адреса: биты 9:8 адреса. Изменение бита допускается при OA1EN=0	RW
OA1	7:1	0x0	Собственный адрес 1 Режим 7-битного адреса: 7-битный адрес. Режим 10-битного адреса: биты 7:1 адреса. Изменение бита допускается при OA1EN=0	RW
OA1	0	0x0	Собственный адрес 1 Режим 7-битного адреса: не имеет значения. Режим 10-битного адреса: бит 0 адреса. Изменение бита допускается при OA1EN=0	RW

3.7.16.4 Регистр собственного адреса 2 (OAR2)

В таблице 35 приведены поля регистра OAR2.

Таблица 35 – Регистр собственного адреса 2 (OAR2)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:16		Зарезервировано	
OA2EN	15	0x0	Использование собственного адреса OA2: 0 – при получении адреса OA2 формируется NACK; 1 – при получении адреса OA2 формируется ACK	RW
Зарезервировано	14:11		Зарезервировано	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
OA2MSK	10:8	0x0	<p>Маска адреса OA2:</p> <p>0b000 – Нет маски</p> <p>0b001 – сравниваются только OA2[7:2];</p> <p>0b010 – сравниваются только OA2[7:3];</p> <p>0b011 – сравниваются только OA2[7:4];</p> <p>0b100 – сравниваются только OA2[7:5];</p> <p>0b101 – сравниваются только OA2[7:6];</p> <p>0b110 – сравниваются только OA2[7];</p> <p>0b111 – OA2[7:1] маскируются, подтверждаются (ACK) все 7-битные адреса (кроме зарезервированных)</p> <p>Если OA2MSK ≠ 0, зарезервированные адреса I²C (0b0000xxx, 0b1111xxx) не подтверждаются, даже если адреса совпадают.</p> <p>Изменение битов допускается при OA2EN=0</p>	RW
OA2	7:1	0x0	<p>Собственный 7-битный адрес 2</p> <p>Изменение битов допускается при OA2EN=0</p>	RW
Зарезервировано	0		Зарезервировано	

3.7.16.5 Регистр временные ограничения (TIMINGR)

В таблице 36 приведены поля регистра TIMINGR.

Регистр должен конфигурироваться, пока интерфейс заблокирован (PE = 0).

Таблица 36 – Регистр временные ограничения (TIMINGR)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
PRESC	31:28	0x0	Предварительный делитель частоты I ² CCLK. Используется для вычисления значения t _{PRES} используемого счетчиками предустановки, удержания, низкого и высокого уровней. t _{PRES} = (PRESC+1) x t _{I2CCLK}	RW
Зарезервировано	27:24	Зарезервировано		
SCLDEL	23:20	0x0	Длительность предустановки данных t _{SU:DAT} . Задержка между изменением SDA и фронтом SCL. t _{SCLDEL} = (SCLDEL+1) x t _{PRES} Используется для генерации задержек t _{SU:DAT}	RW
SDADEL	19:16	0x0	Длительность предустановки данных t _{HD:DAT} . Задержка между спадом SCL и изменением SDA в режиме ведущего и ведомого при NOSTRETCH = 0 t _{SCADEL} = (SCADEL+1) x t _{PRES} Используется для генерации задержек t _{HD:DAT}	RW
SCLH	15:8	0x0	Длительность удержания SCL в состоянии логической «1» в режиме «ведущий» t _{SCLH} = (SCLH+1) x t _{PRES} Также используется для генерации задержек t _{HD:STA} и t _{SU:STO}	RW
SCLL	7:0	0x0	Длительность удержания SCL в состоянии логического «0» в режиме «ведущий» t _{SCLL} = (SCLL+1) x t _{PRES} Также используется для генерации задержек t _{BUF} и t _{SU:STA}	RW

3.7.16.6 Регистр прерываний и статуса (ISR)

В таблице 37 приведены поля регистра ISR.

Таблица 37 – Регистр прерываний и статуса (ISR)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:24		Зарезервировано	
ADDCODE	23:17	0x0	Код совпадшего адреса. Обновляется в режиме «ведомый» при совпадении адреса (ADDR = 1). В режиме 10-битного адреса содержит заголовок (0b11110) и два старших бита адреса.	RO
DIR	16	0x0	Направление передачи (режим «ведомый»). Обновляется при совпадении адреса (ADDR = 1): 0 – тип передачи «запись», ведомый переходит в режим приемника; 1 – тип передачи «чтение», ведомый переходит в режим передатчика.	RO
BUSY	15	0x0	Флаг индикации занятой шины. Устанавливается после события START нашине и сбрасывается после события STOP нашине	RO
Зарезервировано	14:11		Зарезервировано	
OVR	10	0x0	Флаг переполнения/недозагрузки (режим «ведомый» при NOSTRETCH=1) Устанавливается аппаратно, сбрасывается программной установкой бита OVRCF или аппаратно при PE=0.	RO
ARLO	9	0x0	Флаг проигрыша арбитража. Устанавливается аппаратно, сбрасывается программной установкой бита ARLOCF или аппаратно при PE=0.	RO

Обозначение	Разряды	Начальное значение	Назначение	Доступ
BERR	8	0x0	Флаг ошибки шины Устанавливается аппаратно при детектировании неуместного события START или STOP на шине, если интерфейс участвует в передаче. Не устанавливается в фазе адреса в режиме «ведомый». Сбрасывается программной установкой бита BERRCF или аппаратно при PE=0.	RO
TCR	7	0x0	Флаг окончания передачи (режим «ведущий» или «ведомый» с установленным битом SBC). Устанавливается аппаратно при RELOAD=1, после передачи NBYTES байт. Сбрасывается записью в NBYTES ненулевого значения или аппаратно при PE=0.	RO
TC	6	0x0	Флаг окончания передачи (режим «ведущий»). Устанавливается аппаратно при RELOAD=0, AUTOEND=0, после передачи NBYTES байт. Сбрасывается программной установкой бита START или STOP или аппаратно при PE=0.	RO
STOPF	5	0x0	Флаг детектирования STOP на шине. Устанавливается аппаратно, если интерфейс участвует в передаче. Сбрасывается программной установкой бита STOPCF или аппаратно при PE=0.	RO
NACKF	4	0x0	Флаг «не получено подтверждение» (NACK). Устанавливается аппаратно, после передачи байта. Сбрасывается программной установкой бита NACKCF или аппаратно при PE = 0.	RO

Обозначение	Разряды	Начальное значение	Назначение	Доступ
ADDR	3	0x0	Флаг соответствия адреса (режим «ведомый»). Устанавливается аппаратно, если полученный адрес совпадает с одним из разрешенных в OAR1, OAR2. Сбрасывается программной установкой бита ADDRCF или аппаратно при PE = 0.	RO
RXNE	2	0x0	Флаг «буфер принятых данных заполнен» (режимы приёма). Устанавливается аппаратно, после записи принятых данных в регистр RXDR. Сбрасывается при чтении RXDR или аппаратно при PE = 0.	RO
TXIS	1	0x0	Состояние прерывания передатчика. Устанавливается аппаратно, когда регистр TXDR пуст и следующий байт данных должен быть в него записан. Сбрасывается записью следующего байта данных в регистр TXDR или аппаратно при PE = 0. Этот бит может быть установлен программой только при NOSTRETCH = 1 для выработки события TXIS (в результате: прерывание, при TXIE = 1 или DMA запрос, при TXDMAEN = 1)	RW1S
TXE	0	0x0	Флаг «буфер передаваемых данных» TXDR пуст (режимы отправки). Устанавливается аппаратно, если буфер пуст; при PE = 0 или программно, чтобы сбросить содержимое регистра TXDR. Сбрасывается записью следующего байта данных в регистр TXDR.	RW1S

3.7.16.7 Регистр сброса прерываний (ICR)

В таблице 38 приведены поля регистра ISR. Сброс каждого флага производится записью «1» в соответствующий бит регистра.

Таблица 38 – Регистр сброса прерываний (ICR)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:11	Зарезервировано		
OVRCF	10	0x0	Сброс флага прерывания переполнения/недозагрузки	WO
ARLOCF	9	0x0	Сброс флага проигрыша арбитража	WO
BERRCF	8	0x0	Сброс флага ошибки шины	WO
Зарезервировано	7:6	Зарезервировано		
STOPCF	5	0x0	Сброс флага детектирования STOP нашине	WO
NACKCF	4	0x0	Сброс флага «не получено подтверждение» (NACK)	WO
ADDRCF	3	0x0	Сброс флага соответствия адреса	WO
Зарезервировано	2:0	Зарезервировано		

3.7.16.8 Регистр принятых данных (RXDR)

В таблице 39 приведены поля регистра RXDR.

Таблица 39 – Регистр принятых данных (RXDR)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:8	Зарезервировано		
TXDATA	7:0	0x0	Буфер принятых данных	RO

3.7.16.9 Регистр передаваемых данных (TXDR)

В таблице 40 приведены поля регистра TXDR.

Таблица 40 – Регистр передаваемых данных (TXDR)

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:8	Зарезервировано		
TXDATA	7:0	0x0	Буфер передаваемых данных	RW

3.8 Модуль 32-разрядного таймера (TIMER32)

Модуль таймера выполнен на основе 32-х битного счётчика, поддерживает три режима счёта – прямой, обратный и двунаправленный и имеет четыре канала ШИМ и поддержку режимов захвата и сравнения (модуль таймера32_0 реализован без ШИМ и режимов захвата/сравнения).

Особенности таймера/счетчика:

- 32-х битный счётчик трехрежимный;
- 32-х битный программируемый делитель тактовой частоты;
- четыре вида маскируемых прерываний (переполнение счётчика, опустошение счётчика, захват значения, совпадение по сравнению);
- возможность тактирования от системной тактовой частоты или внешнего вывода;
- четыре независимых канала, способных работать в режиме захвата, сравнения или ШИМ (для TIMER32_1, TIMER32_2).

Структурная схема представлена на рисунке 46.

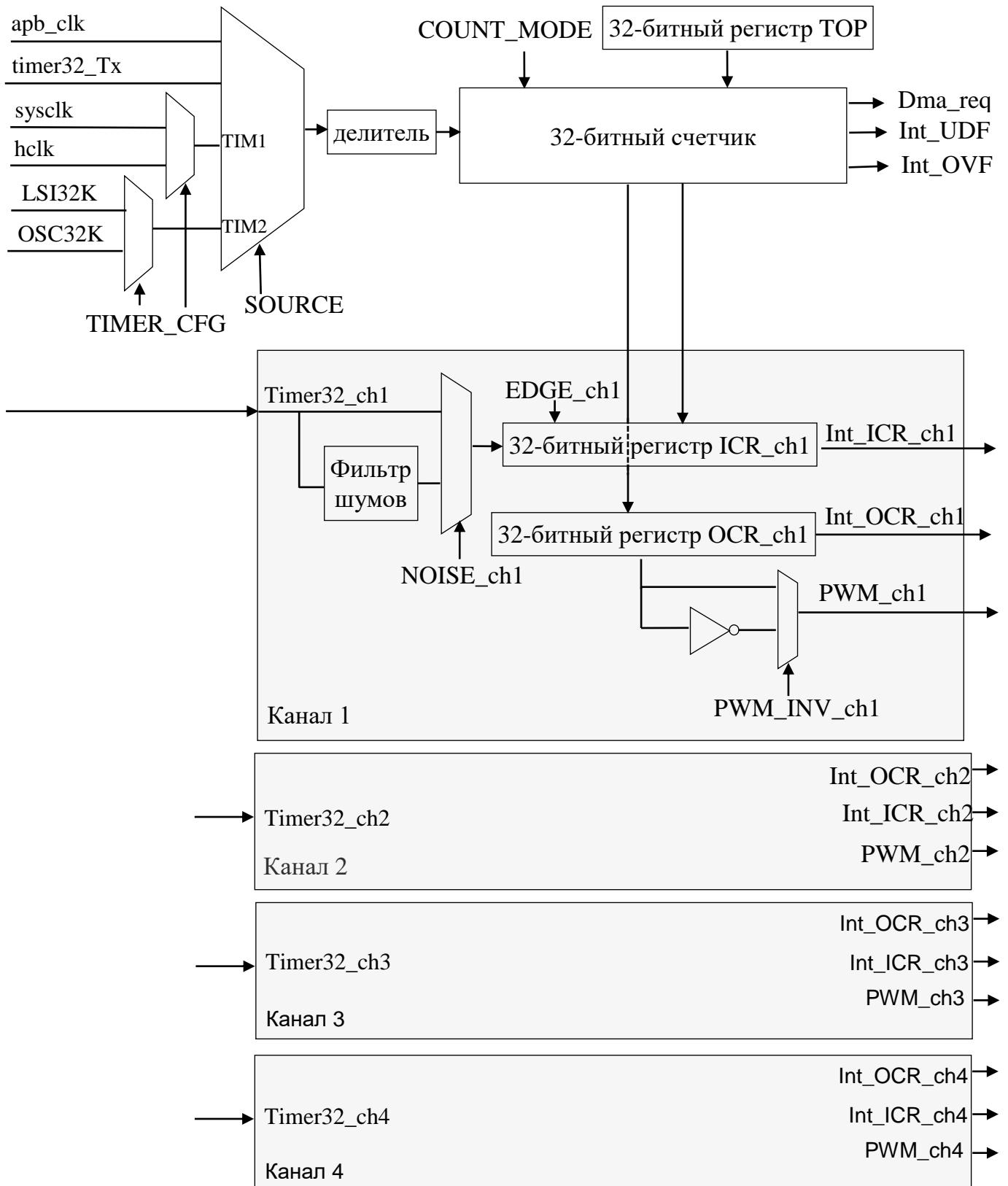


Рисунок 46 – Внутренняя структура 32-разрядного таймера

3.8.1 Блок сравнения, захвата, ШИМ

Блок ШИМ обеспечивает дополнительные режимы работы таймеров:

- таймер 0 – реализован без режима сравнения, захвата, ШИМ;
- таймер (1 – 2) – режимы сравнения, захвата, ШИМ.

Каждый канал имеет свой конфигурационный регистр и полностью независим от других каналов. Конфигурирование канала осуществляется при помощи регистра CH_x_CNTRL (где x – номер канала). Значение сравнения хранится в регистре CH_x_OCR. Значение захвата хранится в регистре CH_x_ICR.

Для запуска канала в режиме захвата необходимо в регистре соответствующего канала CH_x_CNTRL установить следующие поля:

- MODE = 2 (режим захвата);
- EDGE = 0 для захвата фронтов или EDGE = 1 для захвата срезов;
- NOISE = 1 для включении фильтрации помех или NOISE = 0 если фильтрация не нужна;
- EN=1 (включение канала).

При включении канал будет контролировать состояние входа Timer32_1_chx для TIMER32_1 или Timer32_2_chx для TIMER32_2. При изменении на данном входе уровня логического сигнала (с 0 на 1 при EDGE = 0 или с 1 на 0 при EDGE = 1) текущее значение счетчика будет переписано в регистр захвата CH_x_ICR. В регистре INT_FLAG будет установлен в «1» флаг соответствующего канала IC_INT_CHx. При разрешенном прерывании по данному событию (установлено в «1» поле IC_INT_CHx в регистре INT_MASK) будет сформировано прерывание IC_INT .

Для настройки канала в режим сравнения необходимо в регистре соответствующего канала CH_x_CNTRL установить следующие поля:

- MODE = 1 (режим сравнения);
- EN = 1 (включение канала).

При достижении счетчиком значения равного регистру сравнения CH_x_OCR в регистре INT_FLAG будет установлен в «1» флаг соответствующего канала OC_INT_CHx. При разрешенном прерывании по данному событию (установлено в «1» поле OC_INT_CHx в регистре INT_MASK) будет сформировано прерывание OC_INT .

В режиме ШИМ каналы формируют на выходе импульсы заданной формы.

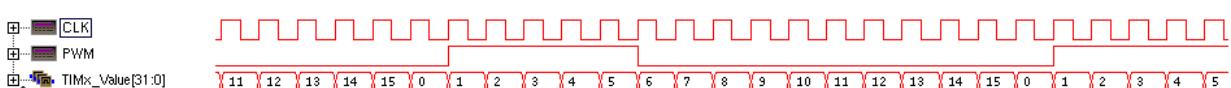


Рисунок 47 – диаграмма работы канала в режиме ШИМ (TOP = 15, OCR = 5)

3.8.2 Регистры модуля таймера 32

В таблице 41 перечислены регистры модулей TIMER32_x.

Таблица 41 – Регистры модуля таймера 32

Обозначение	Смещение	Начальное значение	Назначение	Доступ
VALUE	0x00	0x00000000	Текущее значение таймера	RO
TOP	0x04	0xFFFFFFFF	Максимальное значение счётной последовательности	RW
PRESCALE	0x08	0x00000000	Значение делителя	RW
CONTROL	0x0C	0x00000000	Конфигурационный регистр таймера	RW
ENABLE	0x10	0x00000000	Регистр включения таймера	RW
INT_MASK	0x14	0x00000000	Регистр маски прерываний	RW
INT_CLEAR	0x18	0xFFFFFFFF	Регистр сброса флагов прерываний	RW
INT_FLAG	0x1C	0x00000000	Регистр флагов прерываний	RO
CH1_CNTRL	0x80	0x00000000	Конфигурационный регистр 1 канала	RW
CH1_OCR	0x84	0x00000000	Значение сравнения 1 канала	RW
CH1_ICR	0x88	0x00000000	Значение захвата 1 канала	RW
CH2_CNTRL	0x90	0x00000000	Конфигурационный регистр 2 канала	RW
CH2_OCR	0x94	0x00000000	Значение сравнения 2 канала	RW
CH2_ICR	0x98	0x00000000	Значение захвата 2 канала	RW
CH3_CNTRL	0xA0	0x00000000	Конфигурационный регистр 3 канала	RW
CH3_OCR	0xA4	0x00000000	Значение сравнения 3 канала	RW
CH3_ICR	0xA8	0x00000000	Значение захвата 3 канала	RW
CH4_CNTRL	0xB0	0x00000000	Конфигурационный регистр 4 канала	RW
CH4_OCR	0xB4	0x00000000	Значение сравнения 4 канала	RW
CH4_ICR	0xB8	0x00000000	Значение захвата 4 канала	RW

Примечание – Регистры с смещениями 0x80 – 0xB8 не доступны для TIMER32_0.

Назначение полей регистров представлены в таблицах 42 – 49.

Таблица 42 – Регистр текущего значения таймера VALUE

Обозначение	Разряды	Начальное значение	Назначение	Доступ
VALUE	31:0	0x0	Текущее значение счетчика	RO

Таблица 43 – Регистр максимального значения счётчика TOP

Обозначение	Разряды	Начальное значение	Назначение	Доступ
TOP	31:0	0xFFFFFFFF	Максимальное значение счетчика (ограничивает счетную последовательность сверху)	RW

Таблица 44 – Регистр значения делителя PRESCALE

Обозначение	Разряды	Начальное значение	Назначение	Доступ
PRESCALE	31:0	0x0	Значение предварительного делителя. Предделитель вырабатывает тактовый сигнал для счета, частота которого в (TIM_PRESCALE + 1) раз меньше входной. Диапазон значений от 0 до ($2^{32} - 1$).	RW

Таблица 45 – Конфигурационный регистр таймера CONTROL

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:4		Зарезервировано	
SOURCE	3:2	0x0	Выбор источника тактового сигнала для счета: 0b00 – Частота APB (для timer32_0 – APB_M, для timer32_1, timer32_2 – APB_P); 0b01 – синхронный вход тактирования TIM1 (настраивается в регистрах TIMER_CFG блока управления питанием и тактированием (таблица 172)); 0b10 – вход микросхемы timerX_Tx_pin_pad (X – номер таймера); 0b11 – асинхронный вход тактирования TIM2 настраивается в регистрах TIMER_CFG блока управления питанием и тактированием (таблица 172);	RW
COUNT_MODE	1:0	0x0	Режим счёта таймера: 0b00 – прямой; 0b01 – обратный; 0b10 – двунаправленный	RW

Таблица 46 – Регистр включения ENABLE

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:2		Зарезервировано	
TIM_CLR	1	0x0	Сброс (обнуление) текущего значения счетчика при записи «1»	W1R
TIM_EN	0	0x0	Запуск /останов работы счетчика. 0-Счетчик выключен, 1-счетчик работает	RW

Таблица 47 – Конфигурационный регистр канала x CHx_CNTRL

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:10		Зарезервировано	
DIR	9	0x0	Направление передачи данных. Устанавливается автоматически в зависимости от режима работы (1 — выход, 0 – вход)	RO
PWM_INV	8	0x0	Тип вывода в режиме ШИМ инвертирование: 0 – прямой (не инвертированный выход); 1 – инвертированный выход	RW
EN	7	0x0	Включение/выключение канала 0- канал выключен, 1- канала работает	RW
MODE	6:5	0x0	Режим работы канала: 0b00 – зарезервировано; 0b01 – режим сравнения; 0b10 – режим захвата; 0b11 – ШИМ	RW
EDGE	4	0x0	Режим сигнала захвата: 0 – фронт; 1 – спрез	RW
Зарезервировано	3:1		Зарезервировано	
NOISE	0	0x0	Управление фильтрацией входных помех на входах микросхеме Timer32_1_ch- Timer32_1_ch4, Timer32_2_ch1- Timer32_2_ch4 в режиме захвата. 0 – фильтрация выключена; 1 – фильтрация включена (фильтруются помехи продолжительностью менее 3 тактов частоты APB_P)	RW

Таблица 48 – Регистр значения сравнения канала x CHx_OCR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
OCR	31:0	0x0	Значение таймера в режиме сравнения	RW

Таблица 49 – Регистр значения захвата канала x CHx_ICR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
ICR	31:0	0x0	Значение таймера в режиме захвата	RW

3.8.3 Прерывания

Таймер может генерировать четыре типа прерываний:

- переполнение счётчика – OVF_INT;
- опустошение счётчика – UDF_INT;
- захват значения – IC_INT;
- совпадение по сравнению – OC_INT.

Прерывания по совпадению и захвату значения для каждого канала. Прерывания вырабатываются, всякий раз, когда происходят указанные события и выставляют соответствующие флаги в регистре INT_FLAG. Для разрешения прерываний необходимо записать «1» в бит, соответствующий данному прерыванию в регистре масок INT_MASK. Назначение полей регистров прерываний указано в таблицах 50-52.

Таблица 50 – Регистр маски прерываний INT_MASK

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:10		Зарезервировано	
OC_INT_CH4	9	0x0	Маска прерывания: совпадение 4 канала таймера	RW
OC_INT_CH3	8	0x0	Маска прерывания: совпадение 3 канала таймера	RW
OC_INT_CH2	7	0x0	Маска прерывания: совпадение 2 канала таймера	RW
OC_INT_CH1	6	0x0	Маска прерывания: совпадение 1 канала таймера	RW
IC_INT_CH4	5	0x0	Маска прерывания: захват 4 канала таймера	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
IC_INT_CH3	4	0x0	Маска прерывания: захват 3 канала таймера	RW
IC_INT_CH2	3	0x0	Маска прерывания: захват 2 канала таймера	RW
IC_INT_CH1	2	0x0	Маска прерывания: захват 1 канала таймера	RW
UDF_INT	1	0x0	Маска прерывания: опустошение счетчика	RW
OVF_INT	0	0x0	Маска прерывания: переполнение счетчика	RW

Таблица 51 – Регистр сброса флагов прерываний INT_CLEAR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:10	Зарезервировано		
OC_INT_CH4	9	0x1	Сброс прерывания: совпадение 4 канала таймера	RW
OC_INT_CH3	8	0x1	Сброс прерывания: совпадение 3 канала таймера	RW
OC_INT_CH2	7	0x1	Сброс прерывания: совпадение 2 канала таймера	RW
OC_INT_CH1	6	0x1	Сброс прерывания: совпадение 1 канала таймера	RW
IC_INT_CH4	5	0x1	Сброс прерывания: захват 4 канала таймера	RW
IC_INT_CH3	4	0x1	Сброс прерывания: захват 3 канала таймера	RW
IC_INT_CH2	3	0x1	Сброс прерывания: захват 2 канала таймера	RW
IC_INT_CH1	2	0x1	Сброс прерывания: захват 1 канала таймера	RW
UDF_INT	1	0x1	Сброс прерывания: опустошение счетчика	RW
OVF_INT	0	0x1	Сброс прерывания: переполнение счетчика	RW

Примечание – При чтении считывается последнее записанное значение в регистр.

Таблица 52 – Регистр флагов прерываний INT_FLAG

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:10	Зарезервировано		
OC_INT_CH4	9	0x0	Статус события: совпадение 4 канала таймера	RO
OC_INT_CH3	8	0x0	Статус события: совпадение 3 канала таймера	RO
OC_INT_CH2	7	0x0	Статус события: совпадение 2 канала таймера	RO
OC_INT_CH1	6	0x0	Статус события: совпадение 1 канала таймера	RO
IC_INT_CH4	5	0x0	Статус события: захват 4 канала таймера	RO
IC_INT_CH3	4	0x0	Статус события: захват 3 канала таймера	RO
IC_INT_CH2	3	0x0	Статус события: захват 2 канала таймера	RO
IC_INT_CH1	2	0x0	Статус события: захват 1 канала таймера	RO
UDF_INT	1	0x0	Статус события: опустошение счетчика	RO
OVF_INT	0	0x0	Статус события: переполнение счетчика	RO

3.8.4 Работа с контроллером ПДП

32х разрядные таймеры могут быть настроены для организации передач DMA через заданные интервалы времени. Для этого необходимо настроить мультиплексор запросов ПДП для работы с выбранным таймером. Номера линий запросов от таймеров, следующие:

- TIMER32_0 - линия 12;
- TIMER32_1 - линия 8;
- TIMER32_2 - линия 9;

Таймеры формируют импульс продолжительностью один такт счета при достижении максимального значения счета. На основе этого импульса (после пересинхронизации на системную частоту) формируются запросы на ПДП на запись и

чтение. Запросы автоматически снимаются контроллером ПДП после окончания обработки запроса.

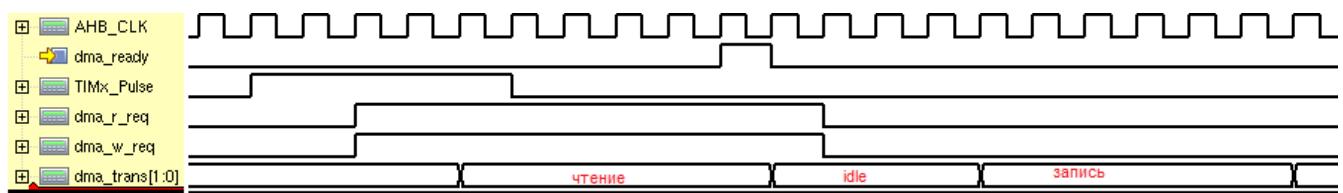


Рисунок 48 – диаграмма работы ПДП по запросу от 32-разрядного таймера

Для непрерывного преобразования с частотой меньше частоты преобразования без участия ядра необходимо настроить 32х разрядный таймер и канал ПДП контроллера для передачи. Пример настройки канала ПДП для пересылки данных из памяти в DAC0:

```
CHx_CFG.ENABLE = 1;  
CHx_CFG.READ_MODE = 1; // Чтение выполняется из памяти  
CHx_CFG.READ_INCREMENT = 1; // при чтении адрес инкрементируется  
CHx_CFG.READ_SIZE = 2; // чтение по словам  
CHx_CFG.READ_REQUEST = 8; // линия Timer32_1  
CHx_CFG.READ_ACK_EN = 1; // формирование сигналов подтверждения  
  
CHx_CFG.WRITE_MODE = 0; // запись выполняется по запросу от таймера  
CHx_CFG.WRITE_INCREMENT = 0; // запись выполняется в один адрес  
CHx_CFG.WRITE_SIZE = 2; //запись по словам  
CHx_CFG.WRITE_REQUEST = 8; // линия Timer32_1  
CHx_CFG.WRITE_ACK_EN = 1; // формирование сигналов подтверждения  
LEN = 15; // 16 байт  
SRC = <RAM_ADDR>;  
DST = 0x00085000 + 0x04; // (ANALOG_REG_BASE_ADDRESS + Offset DAC0  
VALUE)
```

По запросу таймера из ОЗУ будет считано слово, после чего будет ожидаться флаг разрешения записи с использованием ПДП. Флаг формируется после окончания текущего преобразования, то есть если частота запросов таймера меньше частоты преобразования данный, то дополнительного ожидания для записи не будет.

3.8.5 Особенности работы с таймером.

При использовании качестве источника тактового сигнала для счета настраиваемого входа таймера (поле SOURCE регистра CONTROL установлено в «1») выбор тактового сигнала для счета задается в блоке управления питанием тактирования в регистре TIMER_CFG блока Power Manager (смещение 0x30). Следующие поля регистра отвечают за таймеры:

- [1:0] – Таймер32_0;
- [4:3] – Таймер32_1;
- [7:6] – Таймер32_2;

Значения полей дешифрируются следующим образом:

- Младший разряд отвечает за вход TIM1:
 - 1) значение 0 – системная частота (sys_clk);
 - 2) значение 1 – частота шины АHB (hclk);
- Старший разряд отвечает за вход TIM2:
 - 1) значение 0 – частота внешнего осциллятора OSC32K;
 - 2) значение 1 – частота встроенного осциллятора LSI32K.

Внутренний сигнал разрешения работы таймера устанавливается на следующем такте выбранного источника тактирования после установки TIM_EN в регистре ENABLE.

Включение таймера рекомендуется выполнять в следующем порядке:

- Задать необходимый режим работы. В регистре INT_MASK «0»;
- Записать в регистр INT_CLEAR 0xFFFFFFFF;
- Установить TIM_EN в «1»;
- Если нужны прерывания от таймера, установить нужную маску прерываний в регистре INT_MASK.

Выключение таймера рекомендуется выполнять в следующем порядке:

- Записать в регистр INT_MASK «0»;
- Записать в регистр PRESCALE «0»;
- Записать в регистр INT_CLEAR 0xFFFFFFFF;
- Установить TIM_EN в «0».

3.9 Модуль 16-разрядного таймера (TIMER16)

Модуль представляет собой 16-разрядный таймер с низким энергопотреблением (далее Timer16). Так как Timer16 может работать от разных источников тактового сигнала, в том числе и от внешнего тактового сигнала, что делает его применимым в качестве счетчика импульсов. Timer16 может вызывать просыпание системы из режимов с низким потреблением.

Timer16 имеет гибкую схему тактирования, которая дает возможность реализовать достаточное функционирование и производительность при минимизации потребляемой мощности.

Основные функциональные возможности:

- 16 битный счетчик с прямым счетом;
- предделитель частоты с 8 коэффициентами (1, 2, 4, 8, 16, 32, 64, 128);
- выбор источника тактового сигнала (внутренние тактовые сигналы или с внешнего вывода);
- 16 битный регистр автоматической загрузки (ARR);
- 16 битный регистр сравнения;
- постоянный/однократный режим;
- выбор программного/аппаратного входного триггера;
- программируемый цифровой фильтр помех;
- конфигурируемый выход: Импульсный или PWM;
- конфигурируемая полярность входа/выхода;
- режим декодирования.

Структурная схема Timer16 представлена на рисунке 49.

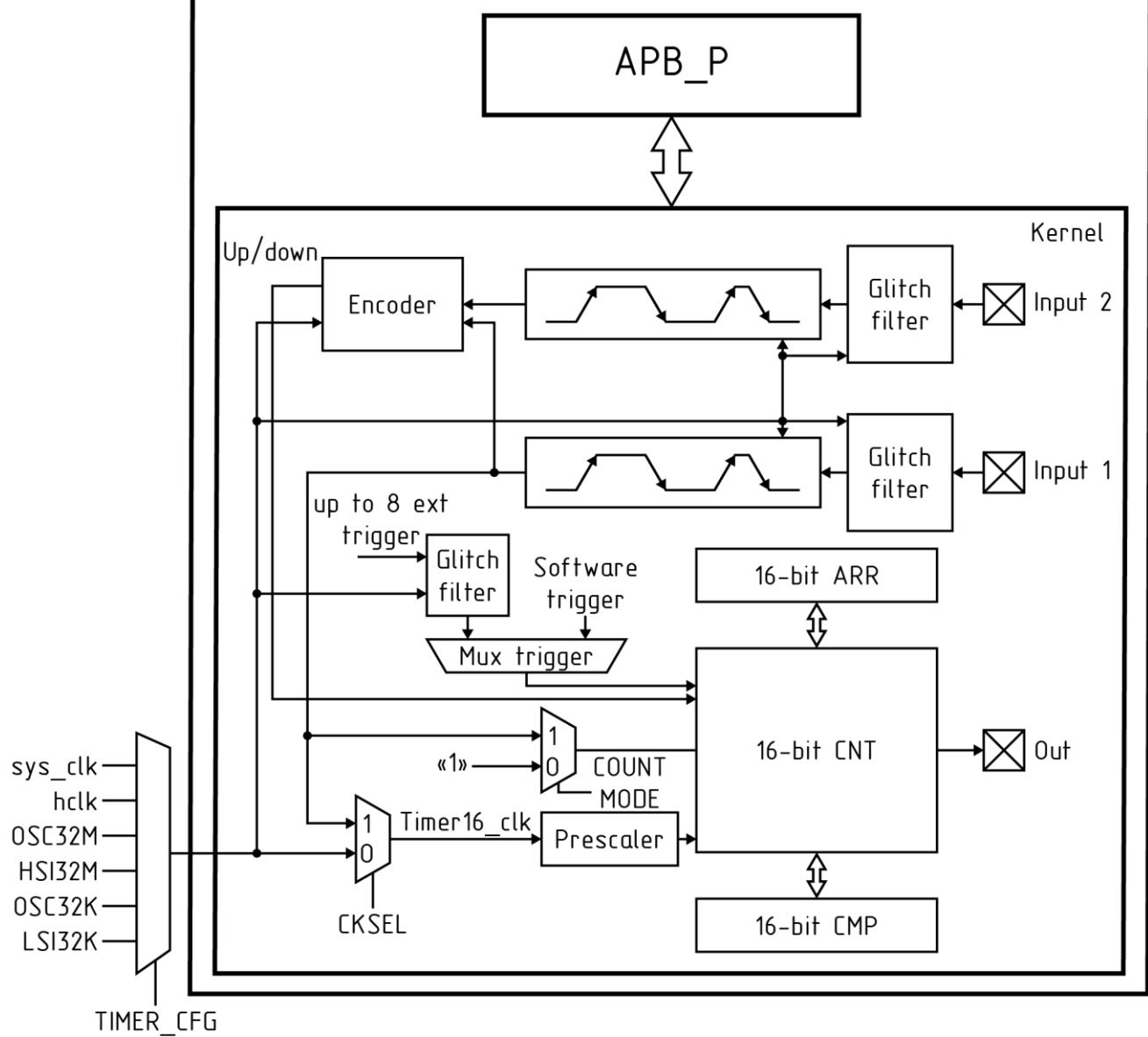


Рисунок 49 – Внутренняя структура 16-разрядного таймера

3.9.1 Тактирование Timer16

TIMER16 может тактироваться с использованием нескольких источников тактового сигнала. Он может тактироваться с помощью внутреннего тактового сигнала, который может быть выбран из источников АHB, SYSCLK, OSC32M, OSC32K, HSI32M или LSI32K через блок управления питанием и тактированием доменов (PowerManager). Также TIMER16 может тактироваться с помощью внешнего тактового сигнала, подаваемого на его внешний вывод Input1. При использовании внешнего источника тактового сигнала TIMER16 может работать в одной из двух возможных конфигураций:

- первая конфигурация - когда TIMER16 тактируется внешним сигналом, но в то же время внутренний тактовый сигнал подается на TIMER16 либо от АHB,

либо от любого другого встроенного генератора, включая OSC32M, OSC32K, HSI32M и LSI32K;

- вторая конфигурация - когда TIMER16 получает тактовую частоту только от внешнего источника через внешний вывод Input1. Эта конфигурация используется для реализации функции Timeout или функции счетчика импульсов, когда все встроенные генераторы выключаются после перехода в режим пониженного энергопотребления.

Программирование битов CKSEL и COUNTMODE позволяет контролировать, будет ли TIMER16 использовать внешний источник синхронизации или внутренний.

При конфигурации для использования внешнего источника тактового сигнала биты CKPOL используются для выбора активного фронта внешнего тактового сигнала. Если оба фронта сконфигурированы как активные, необходимо также обеспечить внутренний тактовый сигнал (первая конфигурация). В этом случае частота внутреннего тактового сигнала должна быть как минимум в четыре раза выше частоты внешнего тактового сигнала.

Выбор тактового сигнала для счета задается в блоке управления питанием тактирования в регистре TIMER_CFG (смещение 0x30). Следующие поля отвечают за таймеры:

- биты [11:9] – Таймер16_0;
- биты [14:12] – Таймер16_1;
- биты [17:15] – Таймер16_2.

Значения полей дешифрируются следующим образом:

- 0x0 – системная частота (sys_clk);
- 0x1 – частота шины АHB (hclk);
- 0x2 – частота внешнего осциллятора OSC32M;
- 0x3 – частота встроенного осциллятора HSI32M;
- 0x4 – частота внешнего осциллятора OSC32K;
- 0x5 – частота встроенного осциллятора LSI32K.

3.9.2 Glitch-фильтр

Входы TIMER16, как внешние (подключенные к GPIO микроконтроллера), так и внутренние (подключенные на уровне микросхемы к другим встроенным периферийным устройствам), защищены цифровыми фильтрами, которые

предотвращают распространение любых помех и шумовых возмущений внутри TIMER16. Это необходимо для предотвращения ложных отсчетов или срабатываний.

Прежде чем активировать цифровые фильтры, на TIMER16 сначала должен быть подан внутренний источник синхронизации. Это необходимо для обеспечения правильной работы фильтров.

Цифровые фильтры делятся на две группы:

– первая группа цифровых фильтров защищает внешние входы TIMER16.

Чувствительность цифровых фильтров управляется битами CKFLT

– вторая группа цифровых фильтров защищает входы внутреннего триггера TIMER16. Чувствительность цифровых фильтров управляется битами TRGFLT.

Примечание - Чувствительность цифровых фильтров управляется группами.

Невозможно настроить чувствительность каждого цифрового фильтра отдельно внутри одной группы.

Чувствительность фильтра зависит от количества последовательных одинаковых выборок, которые должны быть обнаружены на одном из входов TIMER16, чтобы считать изменение уровня сигнала допустимым переходом.

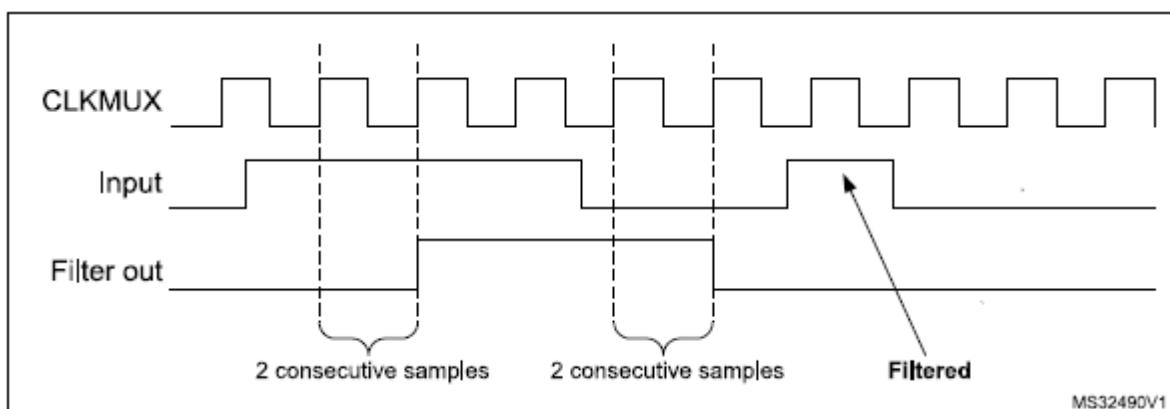


Рисунок 50 – Временная диаграмма glitch-фильтра

На рисунке 50 показан пример поведения glitch-фильтра в случае программирования двойных последовательных выборок.

Примечание – В случае отсутствия внутреннего тактового сигнала цифровой фильтр должен быть деактивирован путем установки битов CKFLT и TRGFLT в «0». В этом случае для защиты внешних входов TIMER16 от помех можно использовать внешний аналоговый фильтр.

3.9.3 Предделитель

16-битному счетчику TIMER16 предшествует конфигурируемый предделитель степени 2. Коэффициент деления предделителя управляется 3-битным полем PRESC[2:0]. В таблице Таблица 53 перечислены все возможные коэффициенты деления:

Таблица 53 – Коэффициенты деления делителя

Значение поля PRESC	Коэффициент деления
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

3.9.4 Мультиплексор триггеров

Счетчик TIMER16 может быть запущен либо программно, либо после обнаружения активного фронта импульса на одном из 8 триггерных входов.

TRIGEN[1:0] используется для выбора источника запуска TIMER16:

- когда TRIGEN[1:0] равен «0b00», счетчик TIMER16 запускается, как только один из битов CNTSTRT или SNGSTRT устанавливается программно. Три оставшихся возможных значения для TRIGEN[1:0] используются для настройки активного фронта, используемого входами триггера. Счетчик TIMER16 запускается, как только активный фронт обнаружен.
- когда TRIGEN[1:0] отличен от «0b00», TRIGSEL[2:0] используется для выбора того, какой из 8 входов используется для запуска счетчика.

Внешние события считаются асинхронными сигналами для TIMER16. Поэтому после обнаружения события требуется задержка в два тактовых периода, прежде чем таймер начнет работать из-за синхронизации.

Если новое событие произойдет, когда таймер уже запущен, оно будет проигнорировано (если не включена функция тайм-аута).

Примечание – Перед установкой битов SNGSTRT/CNTSTRT таймер должен быть включен. Любая запись в эти биты при отключенном таймере будет проигнорирована.

3.9.5 Режим работы

TIMER16 имеет два режима работы:

- непрерывный режим: таймер работает в свободном режиме, таймер запускается по событию запуска и никогда не останавливается, пока таймер не будет отключен;
- однократный режим: таймер запускается от триггерного события и останавливается при достижении значения ARR.

3.9.5.1 Режим однократного события

Для включения однократного подсчета необходимо установить бит SNGSTRT.

Новое событие запуска повторно запустит таймер. Любое триггерное событие, возникающее после запуска счетчика и до того, как счетчик достигнет ARR, будет отброшено.

Если выбран внешний триггер, то каждое событие внешнего триггера, поступающее после установки бита SNGSTRT и после того, как регистр счетчика остановился (содержит нулевое значение), запускает счетчик на новый цикл однократного счета, как показано на рисунке 51.

В случае программного запуска (TRIGEN[1:0] = «0b00»), установка SNGSTRT запустит счетчик для однократного счета.

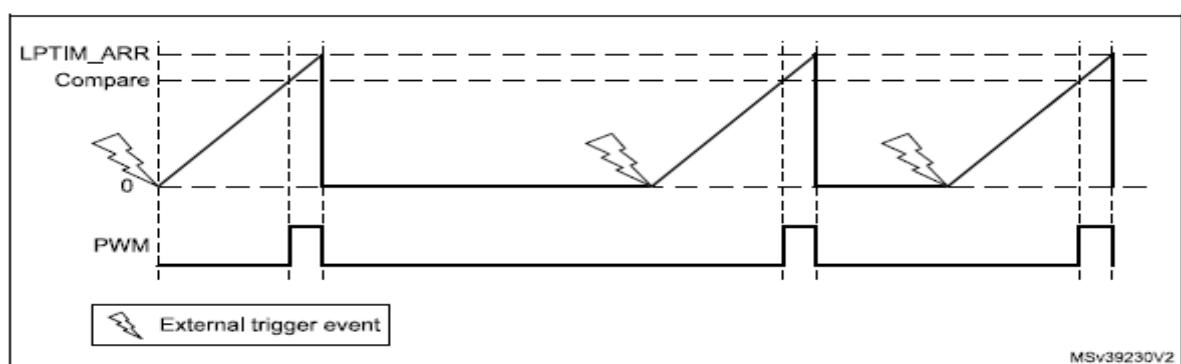


Рисунок 51 – Форма выходного сигнала TIMER16, конфигурация режима однократного счета и активированный режим однократный (бит WAVE установлен)

3.9.5.2 Режим непрерывный

Чтобы включить непрерывный подсчет, необходимо установить бит CNTSTRT.

Если выбран внешний триггер, то событие внешнего триггера, поступающее после установки значения CNTSTRT, запустит счетчик для непрерывного счета. Любое последующее событие внешнего триггера будет отброшено, как показано на рисунке 52.

В случае программного запуска (`TRIGEN[1:0] = «0b0»`), установка CNTSTRT запустит счетчик для непрерывного счета.

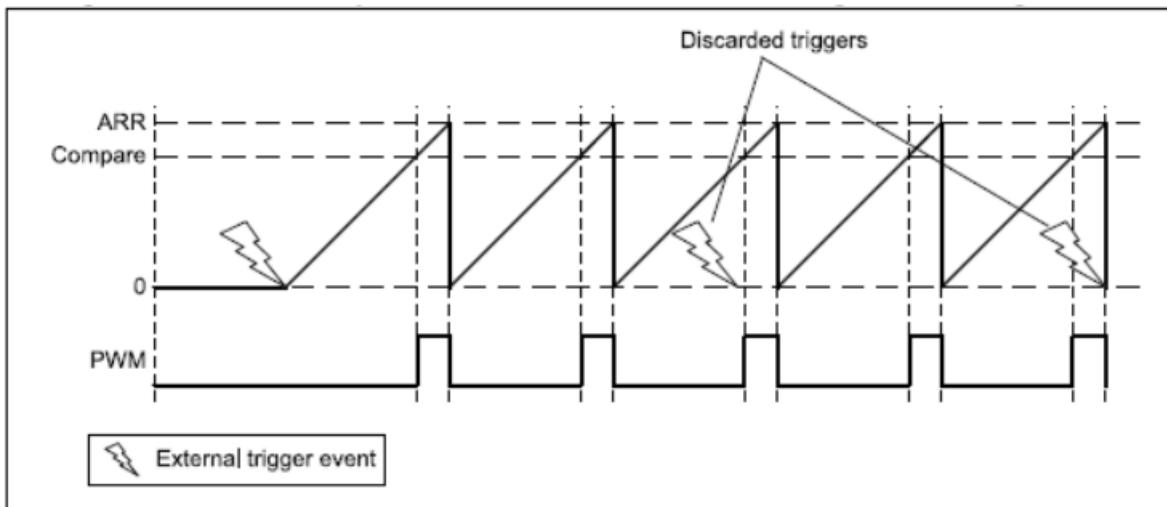


Рисунок 52 – Форма выходного сигнала TIMER16, конфигурация режима непрерывного счета

Биты SNGSTRT и CNTSTRT могут быть установлены только тогда, когда таймер включен (бит ENABLE установлен в «1»). Можно «на лету» перейти из режима однократный в непрерывный режим.

Если ранее был выбран непрерывный режим, установка SNGSTRT переключит TIMER16 в режим однократный. Счетчик (если он активен) остановится, как только достигнет ARR.

Если ранее был выбран режим однократный, установка CNTSTRT переключит TIMER16 в режим непрерывный. Счетчик (если он активен) перезапустится, как только достигнет ARR.

3.9.6 Функция тайм-аута

Обнаружение активного фронта на одном выбранном входе триггера может быть использовано для сброса счетчика TIMER16. Эта функция управляется через бит TIMEOUT.

Первое событие запускает таймер, любое последующее событие сбрасывает счетчик и таймер перезапускается.

Может быть реализована функция тайм-аута с низким энергопотреблением. Значение тайм-аута соответствует значению сравнения; если в течение ожидаемого периода времени триггер не срабатывает, MCU пробуждается по событию совпадения сравнения.

3.9.7 Генерация волновой формы

Два 16-битных регистра, ARR (регистр автозагрузки) и CMP (регистр сравнения), используются для генерации нескольких различных форм сигнала на выходе TIMER16.

Таймер может генерировать следующие формы сигнала:

- режим ШИМ: выход TIMER16 устанавливается, как только возникает соответствие между регистрами CMP и CNT. Выход TIMER16 сбрасывается, как только возникает соответствие между регистрами ARR и CNT;
- одноимпульсный режим: форма выходного сигнала аналогична форме сигнала в режиме ШИМ для первого импульса, затем выход постоянно сбрасывается;
- режим однократный: форма выходного сигнала аналогична режиму одноимпульсный, за исключением того, что на выходе поддерживается последний уровень сигнала (зависит от полярности настроенного выхода).

Форма выходного сигнала TIMER16 может быть настроена через бит WAVE следующим образом:

- сброс бита WAVE в «0» заставляет TIMER16 генерировать либо форму ШИМ, либо форму одного импульса в зависимости от того, какой бит установлен: CNTSTRT или SNGSTRT;
- установка бита WAVE в «1» заставляет TIMER16 генерировать осциллограмму в режиме однократный.

Бит WAVPOL управляет полярностью выхода TIMER16. Изменение вступает в силу немедленно, поэтому выходное значение по умолчанию изменится сразу после перенастройки полярности, еще до включения таймера.

Можно генерировать сигналы с частотой до тактовой частоты TIMER16, деленной на 2. На рисунке 53 показаны три возможные формы сигналов, которые могут быть сгенерированы на выходе TIMER16. Также здесь показан эффект изменения полярности с помощью бита WAVPOL. На рисунке 54 изображена работа канала в режиме ШИМ.

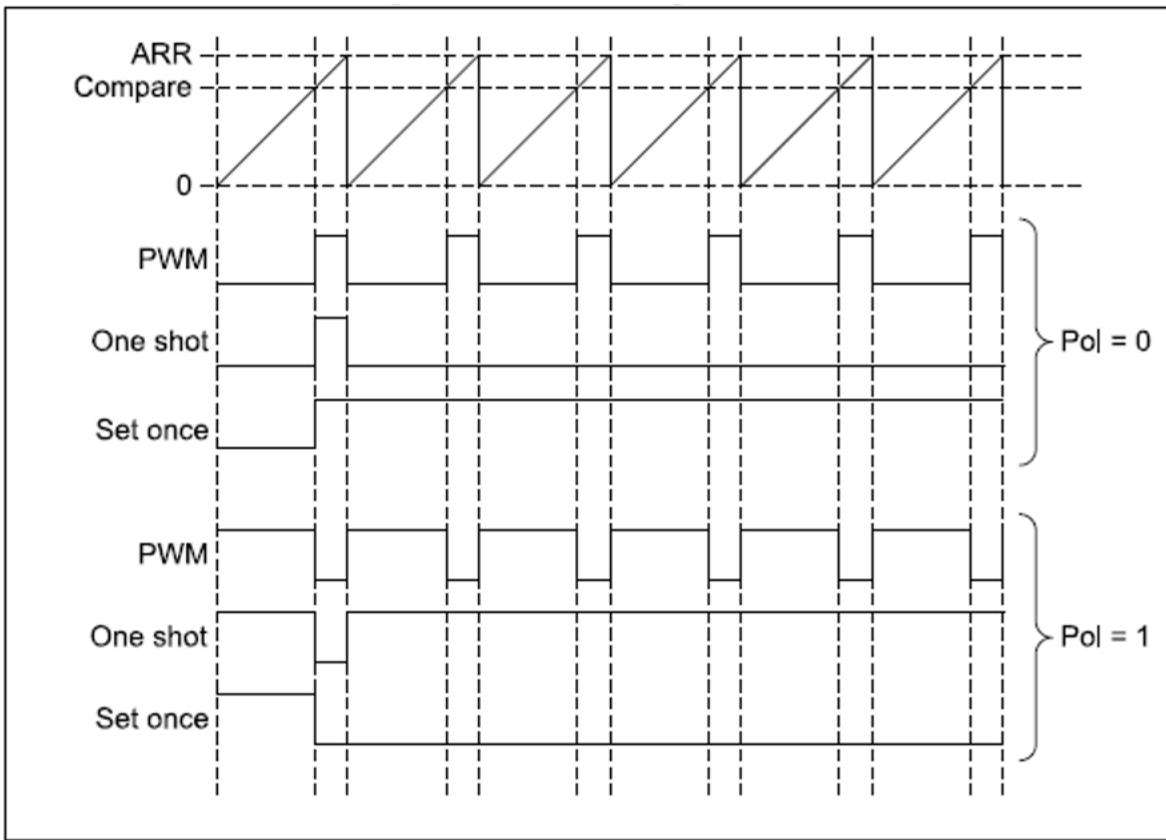


Рисунок 53 – Генерация формы волны

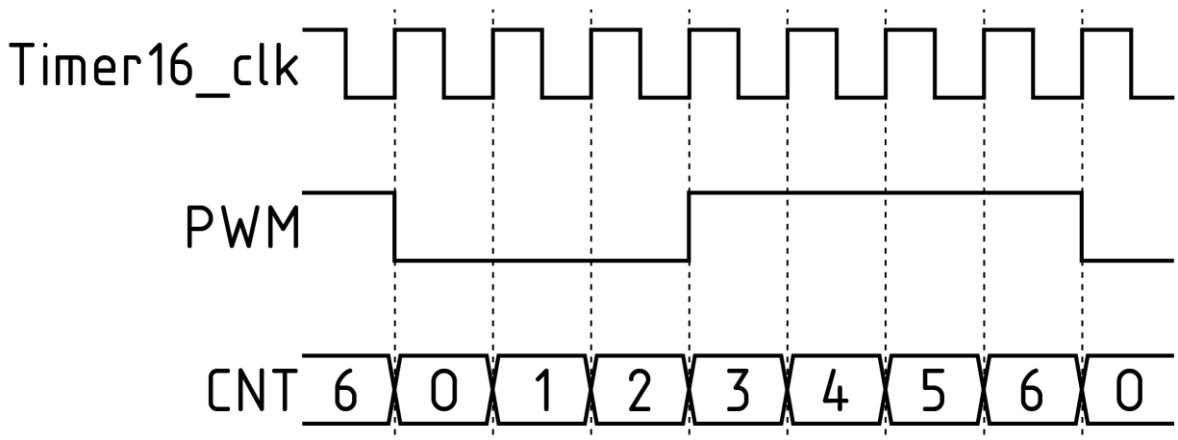


Рисунок 54 - Диаграмма работы канала в режиме ШИМ (PRESC = 0, ARR = 6, CMP = 3, pol = 0)

Время низкого и высокого уровня ШИМ сигнала при $\text{pol} = 0$ зависит от значения делителя (PRESC) и входной частоты таймера до делителя (Timer16_clk):

- при $\text{PRESC} = 0$ низкий уровень ШИМ занимает CMP тактов Timer16_clk, а высокий уровень $\text{ARR} - \text{CMP} + 1$ тактов;
- при $\text{PRESC} > 0$ низкий уровень ШИМ занимает $\text{PRESC} * \text{CMP} - 1$ тактов Timer16_clk, а высокий уровень $\text{PRESC} * (\text{ARR} - \text{CMP} + 1) - (\text{PRESC} - 1)$ тактов.

3.9.8 Обновление регистра

Регистр ARR и регистр CMP обновляются сразу после операции записи в шину APB или в конце текущего периода, если таймер уже запущен.

Бит PRELOAD управляет тем, как обновляются регистры ARR и CMP:

- когда бит PRELOAD сброшен в «0», регистры ARR и CMP немедленно обновляются после любого доступа на запись.
- когда бит PRELOAD установлен в «1», регистры ARR и CMP обновляются в конце текущего периода, если таймер уже был запущен.

Интерфейс APB TIMER16 и логика ядра TIMER16 используют разные тактовые сигналы, поэтому существует некоторая задержка между записью APB и моментом, когда эти значения становятся доступны компаратору счетчика. В течение этого периода задержки следует избегать любых дополнительных записей в эти регистры.

Флаг ARROK и флаг CMPOK в регистре ISR указывают, когда операция записи завершена соответственно в регистр ARR и регистр CMP.

После записи в регистр ARR или CMP новая операция записи в тот же регистр может быть выполнена только после завершения предыдущей операции записи. Любая последующая запись до установки флага ARROK или флага CMPOK приведет к непредсказуемым результатам.

3.9.9 Режим счетчика

Счетчик TIMER16 может использоваться для подсчета внешних событий на входе Input1 TIMER16 или для подсчета внутренних тактовых циклов. Биты CKSEL и COUNTMODE управляют тем, какой источник будет использоваться для обновления счетчика.

Если TIMER16 настроен на подсчет внешних событий на входе Input1, счетчик может быть обновлен по нарастающему фронту, по спадающему фронту или по обоим фронтам в зависимости от значения, записанного в биты CKPOL[1:0].

В зависимости от значений CKSEL и COUNTMODE могут быть выбраны следующие режимы счета:

- CKSEL = 0, COUNTMODE = 0. TIMER16 настроен на тактирование от внутреннего источника тактовых импульсов, а счетчик TIMER16 настроен на обновление после каждого внутреннего тактового импульса.

- CKSEL = 0, COUNTMODE = 1. Вход Input1 TIMER16 дискретизируется внутренним тактовым генератором, подаваемым на TIMER16. Следовательно, чтобы не пропустить ни одного события, частота изменений внешнего сигнала Input1 никогда не должна превышать частоту внутреннего тактового генератора, подаваемого на TIMER16. Кроме того, внутренний тактовый генератор, подаваемый на TIMER16, не должен быть предварительно масштабирован (PRESC[2:0] = 000).
- CKSEL = 1: TIMER16 тактируется внешним источником тактового сигнала. Значение COUNTMODE – не важно. В такой конфигурации TIMER16 не нуждается во внутреннем источнике тактового сигнала (за исключением случаев, когда включены фильтры glitch). Сигнал, подаваемый на Input1 TIMER16, используется в качестве системного тактового генератора для TIMER16. Эта конфигурация подходит для режимов работы, в которых не включен встроенный генератор. При такой конфигурации счетчик TIMER16 может обновляться либо по нарастающему, либо по спадающему фронту тактового сигнала Input1, но не по двум (нарастающему и спадающему фронту) одновременно. Поскольку сигнал, подаваемый на внешний Input1 TIMER16, также используется для тактирования логики ядра TIMER16, существует некоторая начальная задержка (после включения TIMER16) перед началом счета. Точнее, первые пять активных фронтов на внешнем Input1 TIMER16 (после включения TIMER16) теряются.

3.9.10 Включение и выключение таймера

Бит ENABLE, расположенный в регистре CR, используется для включения/выключения логики ядра TIMER16. После установки бита ENABLE необходима задержка в два такта, прежде чем TIMER16 будет фактически включен.

Регистры CFGR и IER должны быть изменены только тогда, когда TIMER16 отключен.

При отключении таймера и повторном включении необходимо заново проинициализировать значение в регистре ARR.

3.9.11 Режим энкодера

Этот режим позволяет обрабатывать сигналы от квадратурных кодеров, используемых для определения углового положения поворотных элементов. Режим

интерфейса энкодера действует просто как внешний тактовый генератор с выбором направления. Это означает, что счетчик просто ведет непрерывный счет между 0 и значением автоперезагрузки, запрограммированным в регистре ARR (0 до ARR или ARR до 0 в зависимости от направления). Поэтому перед началом работы необходимо сконфигурировать ARR. Из двух внешних входных сигналов, Input1 и Input2, генерируется тактовый сигнал для синхронизации счетчика TIMER16. Фаза между этими двумя сигналами определяет направление счета.

Режим энкодера доступен только в том случае, если TIMER16 работает от внутреннего источника синхронизации. Частота сигналов на обоих входах Input1 и Input2 не должна превышать частоту внутреннего тактового генератора TIMER16, деленную на 4. Это обязательно для обеспечения нормальной работы TIMER16.

Смена направления сигнализируется двумя флагами Down и Up в регистре ISR. Кроме того, для обоих событий изменения направления может быть сгенерировано прерывание, если оно разрешено через регистр IER.

Для активации режима энкодера бит ENC должен быть установлен в «1». Сначала TIMER16 должен быть сконфигурирован в непрерывном режиме.

Когда активен режим энкодера, счетчик TIMER16 изменяется автоматически в соответствии со скоростью и направлением инкрементного энкодера. Поэтому его содержимое всегда представляет положение энкодера. Направление счета, сигнализируемое флагами Up и Down, соответствует направлению вращения ротора энкодера.

В соответствии с чувствительностью фронта импульса, настроенной с помощью битов CKPOL[1:0], возможны различные сценарии подсчета. В таблице 54 приведены возможные комбинации в предположении, что Input1 и Input2 не переключаются одновременно.

Таблица 54 – Сценарии подсчета энкодера

Активный фронт	Уровень противоположного сигнала (Input1 для Input2, Input2 для Input1)	сигнал Input1		сигнал Input2	
		Подъем	Падение	Подъем	Падение
Восходящий фронт	Высокий	Вниз	Не считать	Вверх	Не считать
	Низкий	Вверх	Не считать	Вниз	Не считать
Падающий фронт	Высокий	Не считать	Вверх	Не считать	Вниз
	Низкий	Не считать	Вниз	Не считать	Вверх
Оба фронта	Высокий	Вниз	Вверх	Вверх	Вниз
	Низкий	Вверх	Вниз	Вниз	Вверх

На рисунке 40 показана последовательность отсчета для режима кодера, в котором настроена чувствительность по обоим краям.

Примечание – В режиме энкодера TIMER16 должен тактироваться внутренним источником тактового сигнала, поэтому бит CKSEL должен иметь значение сброса, равное «0». Кроме того, коэффициент деления предделителя должен быть равен его начальному значению, которое равно 1 (биты PRESC[2:0] должны быть равны «0b000»).

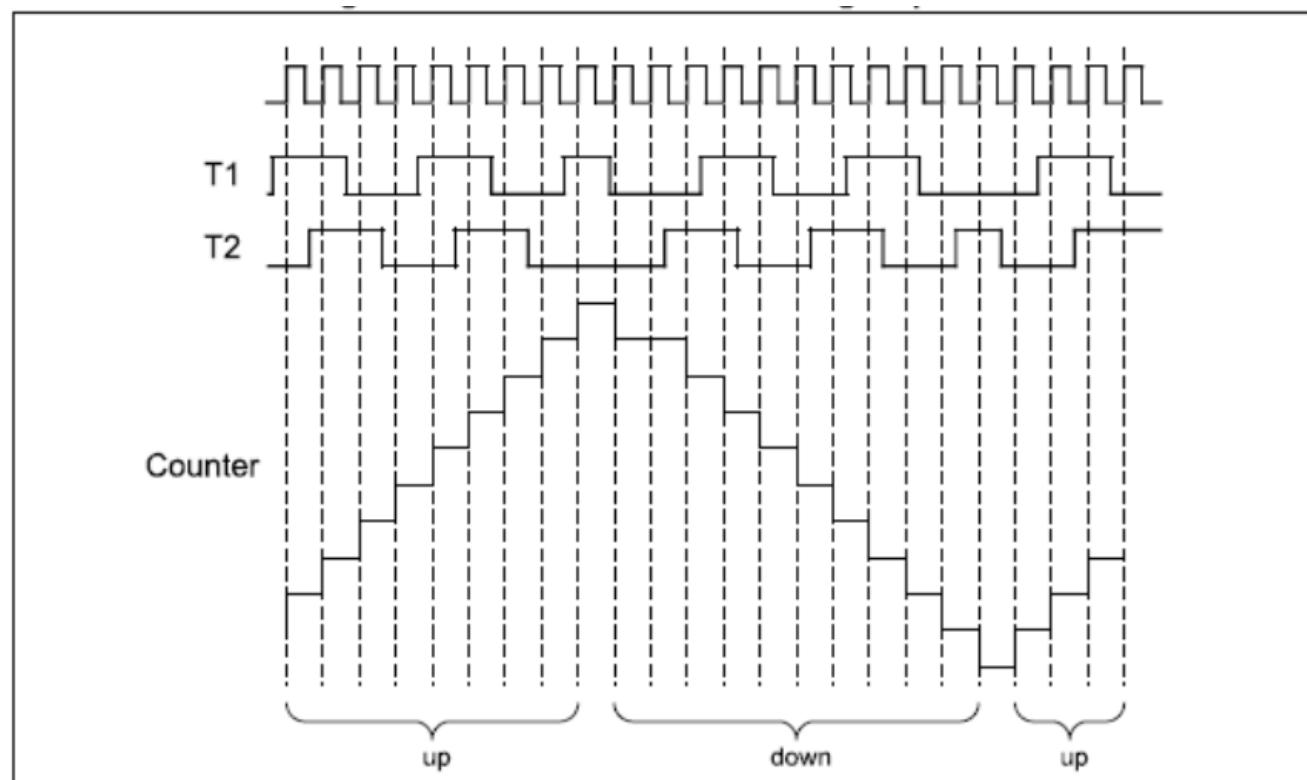


Рисунок 55 – Последовательность счета в режиме энкодера

3.9.12 Прерывания TIMER16

Следующие события генерируют событие прерывание/пробуждение, если они разрешены с помощью функции регистра IER:

- совпадение значения;
- автоперезагрузка значения (независимо от направления в режиме энкодера);
- внешнее пусковое событие;
- запись в регистр автоперезагрузки завершена;
- запись в регистр сравнения завершена;
- изменение направления (режим энкодера), программируемое (вверх/вниз/в обе стороны).

Примечание – Если какой-либо бит в регистре IER (регистр разрешения прерывания) установлен после того, как установлен соответствующий флаг в регистре ISR (регистр состояния), прерывание не утверждается.

Таблица 55 – События прерывания

Событие прерывания	Описание
Совпадение значения	Флаг прерывания поднимается, когда содержимое регистра счетчика (CNT) совпадает с содержимым регистра сравнения (CMP).
Автоперезагрузка значения	Флаг прерывания поднимается, когда содержимое регистра счетчика (CNT) совпадает с содержимым регистра автоперезагрузки (ARR).
Внешнее пусковое событие	Флаг прерывания поднимается при обнаружении внешнего триггерного события
Регистра автоперезагрузки обновился	Флаг прерывания поднимается, когда операция записи в регистр ARR завершена.
Регистр сравнения обновился	Флаг прерывания поднимается, когда операция записи в регистр CMP завершена.
Изменение направления	Используется в режиме энкодера. Два флага прерывания встроены для сигнализации изменения направления: Флаг UP сигнализирует о смене направления счета вверх; Флаг DOWN сигнализирует о смене направления счета вниз.

3.9.13 Регистры TIMER16

Регистры 16-разрядного таймера представлены в таблицах 56-59.

Таблица 56 – Регистры 16-разрядного таймера

Обозначение	Смещение	Начальное значение	Назначение	Доступ
ISR	0x00	0x0	Регистр флагов прерываний	RO
ICR	0x04	0x0	Регистр сброса флагов прерываний	W1C
IER	0x08	0x0	Регистр разрешения прерываний	RW
CFGReg	0x0C	0x0	Регистр конфигурации	RW
CR	0x10	0x0	Регистр управления	RW
CMP	0x14	0x0	Регистр сравнения	RW
ARR	0x18	0x1	Регистр автоматической перезагрузки	RW
CNT	0x1C	0x0	Регистр счетчика	RO

Таблица 57 – Регистр флагов прерываний ISR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:7		Зарезервировано	
DOWN	6	0x0	Изменение направления счетчика вверх на вниз. В режиме энкодера бит DOWN устанавливается аппаратно, чтобы сообщить приложению, что направление счетчика изменилось с «вверх» на «вниз».	RO
UP	5	0x0	Изменение направления счетчика с вниз на вверх. В режиме энкодера бит UP устанавливается аппаратно, чтобы сообщить приложению, что направление счетчика изменилось с «вниз» на «вверх».	RO
ARROK	4	0x0	Обновление регистра автозагрузки OK. ARROK устанавливается аппаратными средствами, чтобы сообщить приложению, что операция записи в регистр ARR шины APB успешно завершена	RO

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CMPOK	3	0x0	Обновление регистра сравнения ОК. CMPOK устанавливается аппаратными средствами, чтобы сообщить приложению, что операция записи в регистр CMP шины APB успешно завершена.	RO
EXTTRIG	2	0x0	Событие фронта внешнего триггера. EXTTRIG устанавливается аппаратно, чтобы сообщить приложению, что на выбранном входе внешнего триггера возник достоверный фронт импульса. Если триггер игнорируется, так как таймер уже запущен, то этот флаг не устанавливается	RO
ARRM	1	0x0	Соответствие автозагрузки. ARRM устанавливается аппаратурой, чтобы сообщить приложению, что значение регистра CNT достигло значения регистра ARR.	RO
CMPM	0	0x0	Совпадение сравнения. Бит CMPM устанавливается аппаратными средствами, чтобы сообщить приложению, что значение регистра CNT достигло значения регистра CMP	RO

Таблица 58 – Регистр сброса флагов прерываний ICR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:7	Зарезервировано		
DOWNCF	6	0x0	Запись «1» в этот бит снимает флаг DOWN в регистре LPT_ISR	W1C
UPCF	5	0x0	Запись «1» в этот бит снимает флаг UP в регистре LPT_ISR	W1C
ARROKCF	4	0x0	Запись «1» в этот бит снимает флаг ARROK в регистре LPT_ISR	W1C
CMPOKCF	3	0x0	Запись «1» в этот бит снимает флаг CMPOK в регистре LPT_ISR	W1C
EXTTRIGCF	2	0x0	Запись «1» в этот бит снимает флаг EXTTRIG в регистре LPT_ISR	W1C

Обозначение	Разряды	Начальное значение	Назначение	Доступ
ARRMCF	1	0x0	Запись «1» в этот бит снимает флаг ARRMв регистре LPT_ISR	W1C
CMPMCF	0	0x0	Запись «1» в этот бит снимает флаг CMPMв регистре LPT_ISR	W1C

Таблица 59 – Регистр разрешения прерываний IER

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:7		Зарезервировано	
DOWNIE	6	0x0	1 - прерывание DOWN разрешены 0 - прерывание DOWN запрещены	RW
UPIE	5	0x0	1 - прерывание UP разрешены 0 - прерывание UP запрещены	RW
ARROKIE	4	0x0	1 - прерывание ARROK разрешены 0 - прерывание ARROK запрещены	RW
CMPOKIE	3	0x0	1 - прерывание CMPOK разрешены 0 - прерывание CMPOK запрещены	RW
EXTTRIGIE	2	0x0	1 - прерывание EXTTRIG разрешены 0 - прерывание EXTTRIG запрещены	RW
ARRMIE	1	0x0	1 - прерывание ARRM разрешены 0 - прерывание ARRM запрещены	RW
CMPMIE	0	0x0	1 - прерывание CMPM разрешены 0 - прерывание CMPM запрещены	RW

Таблица 60 – Регистр конфигурации CFGR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:25		Зарезервировано	
ENC	24	0x0	Разрешение режима энкодера. Бит ENC управляет режимом работы энкодера 0 - Режим энкодера отключен 1 - Режим энкодера включен	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
COUNT MODE	23	0x0	Бит COUNTMODE выбирает, какой источник тактового сигнала используется TIMER16 для синхронизации счетчика: 0 - счетчик инкрементируется после каждого внутреннего тактового импульса 1 - счетчик увеличивается после каждого действительного тактового импульса на внешнем Input1 TIMER16.	RW
PRELOAD	22	0x0	Режим обновления регистров. Бит PRELOAD управляет модальностью обновления регистров ARR и CMP 0 - Регистры обновляются после каждого доступа к записи на шине APB 1 - Регистры обновляются в конце текущего периода TIMER16	RW
WAVEPOL	21	0x0	Полярность формы волны. Бит WAVEPOL управляет полярностью выходного сигнала 0 - Выход TIMER16 отражает результаты сравнения между регистрами ARR и CMP 1 - Выход TIMER16 отражает инверсные результаты сравнения между регистрами ARR и CMP	RW
WAVE	20	0x0	Форма волны. Бит WAVE управляет формой выходного сигнала 0 - Деактивировать режим Set-once. Форма волны ШИМ / один импульс (в зависимости от того, какой бит установлен: CNTSTART или SNGSTART) 1 - Активировать режим Set-once	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
TIMEOUT	19	0x0	<p>Разрешение тайм-аута. Бит TIMEOUT управляет функцией тайм-аута</p> <p>0 - триггерное событие, поступающее, когда таймер уже запущен, будет проигнорировано</p> <p>1 - Триггерное событие, поступающее, когда таймер уже запущен, сбросит и перезапустит счетчик.</p>	RW
TRIGEN	18:17	0x0	<p>Разрешение и полярность триггера. Бит TRIGEN управляет тем, запускается ли счетчик TIMER16 внешним триггером или нет. Если выбрана опция внешнего запуска, возможны три конфигурации активного фронта триггера:</p> <p>0b00 - программный триггер (начало отсчета инициируется программно)</p> <p>0b01 - нарастающий фронт является активным фронтом</p> <p>0b10 - падающий фронт является активным фронтом</p> <p>0b11 - оба фронта являются активными фронтами энергопотреблением (TIMER16)</p>	RW
Зарезервировано	16		Зарезервировано	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
TRIGSEL	15:13	0x0	<p>Селектор триггера. Биты TRIGSEL выбирают источник триггера, который будет служить событием запуска для TIMER16, из 8 доступных ниже источников:</p> <ul style="list-style-type: none"> 0b000 - timer16_0 - GPIO0_7 timer16_1 - GPIO1_9 timer16_2 - GPIO2_3 0b001 - timer16_0 - GPIO0_4 timer16_1 - GPIO1_8 timer16_2 - GPIO2_2 0b010 - timer16_0 - GPIO0_15 timer16_1 - GPIO1_7 timer16_2 - GPIO2_1 0b011 - timer16_0 - GPIO0_14 timer16_1 - GPIO1_6 timer16_2 - GPIO2_0 0b100 - окончание преобразования термосенсора 0b101 - окончание преобразования АЦП 0b110 - прерывание RTC 0b111 - будильник 	RW
Зарезервировано	12	Зарезервировано		
PRESC	11:9	0x0	<p>Делитель частоты. Биты PRESC задают коэффициент деления делителя. Он может быть одним из следующих коэффициентов деления:</p> <ul style="list-style-type: none"> 0b000: /1 0b001: /2 0b010: /4 0b011: /8 0b100: /16 0b101: /32 0b110: /64 0b111: /128 	RW
Зарезервировано	8	Зарезервировано		

Обозначение	Разряды	Начальное значение	Назначение	Доступ
TRGFLT	7:6	0x0	<p>Конфигурируемый цифровой фильтр для триггера. Значение TRGFLT устанавливает количество последовательных одинаковых выборок, которые должны быть обнаружены при изменении уровня на внутреннем триггере, прежде чем это будет считаться действительным переходом уровня. Для использования этой функции необходимо наличие источника внутреннего тактового сигнала.</p> <p>0b00 - любое изменение активного уровня триггера считается действительным триггером</p> <p>0b01 - изменение активного уровня триггера должно быть стабильным в течение не менее 2 тактовых периодов, прежде чем он будет считаться действительным триггером.</p> <p>0b10 - изменение активного уровня триггера должно быть стабильным в течение как минимум 4 тактовых периодов, прежде чем он будет считаться действительным триггером.</p> <p>0b11 - изменение активного уровня триггера должно быть стабильным в течение не менее 8 тактовых периодов, прежде чем он будет считаться действительным триггером.</p>	RW
Зарезервировано	5		Зарезервировано	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CKFLT	4:3	0x0	<p>Конфигурируемый цифровой фильтр для внешнего тактового генератора. Значение CKFLT устанавливает количество последовательных одинаковых выборок, которые должны быть обнаружены при изменении уровня внешнего тактового сигнала, прежде чем это будет считаться действительным переходом уровня. Для использования этой функции необходимо наличие внутреннего источника тактового сигнала.</p> <p>0b00 - любое изменение уровня внешнего тактового сигнала рассматривается как допустимый переход</p> <p>0b01 - изменение уровня внешнего тактового сигнала должно быть стабильным в течение как минимум 2 тактовых периодов, прежде чем оно будет считаться допустимым переходом.</p> <p>0b10 - изменение уровня внешнего тактового сигнала должно быть стабильным в течение как минимум 4 тактовых периодов, прежде чем оно будет считаться действительным переходом.</p> <p>0b11 - изменение уровня внешнего тактового сигнала должно быть стабильным в течение как минимум 8 тактовых периодов, прежде чем оно будет считаться допустимым переходом.</p>	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CKPOL	2:1	0x0	<p>Полярность синхронизации. Когда TIMER16 тактируется внешним источником тактового сигнала, биты CKPOL используются для настройки активного фронта или фронтов, используемых счетчиком:</p> <p>0b00 - нарастающий фронт является активным фронтом, используемым для подсчета</p> <p>0b01 - спадающий фронт является активным фронтом, используемым для подсчета</p> <p>0b10 - оба фронта являются активными фронтами. Когда оба фронта внешнего тактового сигнала считаются активными, TIMER16 должен также тактироваться внутренним источником тактового сигнала с частотой, по крайней мере в четыре раза превышающей частоту внешнего тактового сигнала.</p> <p>0b11 - не разрешается</p> <p>Если TIMER16 сконфигурирован в режиме энкодера (установлен бит ENC):</p> <p>0b00 - активен подрежим энкодера 1</p> <p>0b01 - активен подрежим энкодера 2</p> <p>0b10 - активен подрежим энкодера 3</p>	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CKSEL	0	0x0	<p>Селектор тактовых импульсов. Бит CKSEL выбирает, какой источник тактовых импульсов будет использовать TIMER16:</p> <p>0 - TIMER16 тактируется внутренним источником тактового сигнала (APB тактовый генератор или любой из встроенных генераторов)</p> <p>1 - TIMER16 тактируется внешним источником тактового сигнала через внешний Input1 TIMER16.</p>	RW

Примечание: Регистр CFGR может быть изменен только тогда, когда TIMER16 отключен (бит ENABLE сброшен в «0»).

Таблица 61 – Регистр управления CR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:3		Зарезервировано	
CNTSTRT	2	0x0	<p>Запуск таймера в непрерывном режиме</p> <p>Этот бит устанавливается программно и очищается аппаратно.</p> <p>В случае программного запуска (TRIGEN[1:0] = 0b00), установка этого бита запускает TIMER16 в непрерывном режиме. Если программный запуск отключен (TRIGEN[1:0] отличен от 0b00), установка этого бита запускает таймер в непрерывном режиме, как только будет обнаружен внешний триггер.</p> <p>Если этот бит установлен, когда идет счет в режиме одиночного импульса, то таймер не остановится при очередном совпадении регистров ARR и CNT, и счетчик TIMER16 продолжит счет в непрерывном режиме. Этот бит может быть установлен только при включенном TIMER16. Он будет автоматически сброшен аппаратными средствами.</p>	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
SNGSTRT	1	0x0	<p>Запуск TIMER16 в одиночном режиме</p> <p>Этот бит устанавливается программно и очищается аппаратно.</p> <p>В случае программного запуска (TRIGEN[1:0] = 0b00), установка этого бита запускает TIMER16 в режиме одиночного импульса.</p> <p>Если программный запуск отключен (TRIGEN[1:0] отличен от 0b00), установка этого бита запускает TIMER16 в режиме одиночного импульса, как только обнаруживается внешний триггер.</p> <p>Если этот бит установлен, когда TIMER16 находится в режиме непрерывного счета, то TIMER16 остановится при следующем совпадении регистров ARR и CNT.</p> <p>Этот бит может быть установлен только при включенном TIMER16. Он будет автоматически сброшен аппаратными средствами.</p>	RW
ENABLE	0	0x0	Разрешение TIMER16. Бит ENABLE устанавливается и очищается программно.	RW

Таблица 62 – Регистр сравнения CMP

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:16	Зарезервировано		
CMP	15:0	0x0	Сравниваемое значение. CMP — это значение сравнения, используемое TIMER16.	RW

Примечание – Регистр CMP может быть изменен только тогда, когда TIMER16 включен (бит ENABLE установлен в «1»).

Таблица 63 – Регистр автоматической перезагрузки ARR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:16		Зарезервировано	
ARR	15:0	0x01	Значение автоматической перезагрузки ARR - значение автозагрузки для TIMER16. Это значение должно быть строго больше, чем значение CMP[15:0].	RW
Примечание – Регистр ARR может быть изменен только тогда, когда TIMER16 включен (бит ENABLE установлен в «1»).				

Таблица 64 – Регистр счетчика CNT

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:16		Зарезервировано	
CNT	15:0	0x0	Значение счетчика. Когда TIMER16 работает с асинхронными тактовыми сигналами, чтение регистра CNT может вернуть недостоверные значения. Поэтому в этом случае необходимо выполнить два последовательных доступа на чтение и убедиться, что два возвращенных значения идентичны.	RO

3.10 Часы реального времени (RTC)

Модуль реализует функцию часов реального времени. Основные возможности:

- 24-часовой режим работы;
- функция календаря с корректировкой для високосного года;
- программируемый будильник с возможностью генерации прерываний;
- двоично-десятичное кодирование данных.

3.10.1 Описание модуля

Модуль предназначен для учёта времени и даты. Внутренняя структура модуля представлена на рисунке 55.

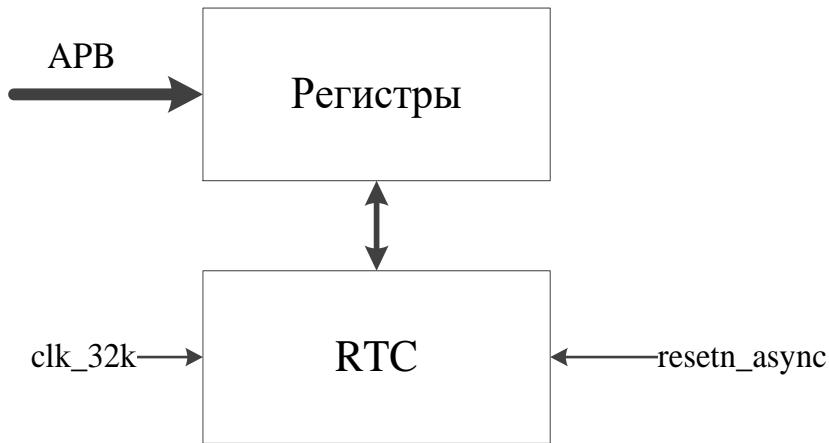


Рисунок 56 – Внутренняя структура модуля

Модуль состоит из двух функциональных элементов. Первый содержит набор конфигурационных регистров. Второй – функциональный модуль подсчёта времени/даты.

3.10.2 Выбор источника тактирования

Модуль часов реального времени может тактироваться от одного из двух источников: внутренний LSI32K генератор и внешний осциллятор OSC32K. В микроконтроллере реализована схема слежения за тактовыми сигналами часов реального времени. При отключении или потери приоритетного источника тактирования происходит автоматическое переключение на второй источник. Выбор приоритетного источника тактирования осуществляется в регистре управления тактированием батарейного домена CLOCKS_BU в полях RTC_CLK_MUX. Допустимые значения RTC_CLK_MUX:

- 0x0 – автоматический выбор. При наличии обоих источников 32 кГц выбирается внутренний LSI32K;
- 0x1 – внутренний LSI32K;
- 0x2 – внешний осциллятор OSC32K.

3.10.3 Конфигурация

Для конфигурации времени/даты необходимо сбросить бит EN регистра RRTC_CTRL (если он установлен) и записать желаемое время/дату. В противном случае, результат работы модуля может быть непредсказуем, т.к. не производится никакого аппаратного контроля корректности установленных значений. Для кодирования данных используется BCD код. Это означает, что, например, для установки текущего времени необходимо записать шесть чисел.

3.10.4 Будильник

Модуль поддерживает функцию будильника по совпадению времени/даты. В качестве источника сравнения может быть выбрано одно или несколько полей. В случае выбора нескольких полей, совпадением считается равенство всех выбранных полей.

Список полей, доступных для сравнения, представлен ниже:

- секунды;
- минуты;
- часы;
- дни;
- день недели;
- месяцы;
- года;
- века.

При совпадении заданного и текущего времени формируется флаг будильника (текущий статус RRTC_CTRL.ALRM), прерывание (если оно разрешено) и формируется активный уровень «1» на выводе RTC_ALARM. Продолжительность активного уровня - один период источника тактирования часов реального времени. Для сброса флага будильника необходимо записать «0» в поле RRTC_CTRL.ALRM. Для сброса признака установки активного уровня на выводе RTC_ALARM необходимо записать «1» в поле RRTC_CTRL.ALRM_PAD. Если при сбросе признака активного уровня вывода RTC_ALARM (запись «1» в RRTC_CNTL.ALRM_PAD) флаг будильника активный, то будет сформирован еще один строб на выводе RTC_ALARM.

Сбросить счетчики, флаги и признак будильника RTC можно с помощью записи в регистр RTC_CONTROL модуля WakeUp.

3.10.5 Регистры модуля RTC

В таблице

Таблица 65 указаны смещения регистров, доступных по интерфейсу конфигурации.

Таблица 65 – Регистры модуля RTC

Обозначение	Смещение	Начальное значение	Назначение	Доступ
RRTC_TIME	0x00	0x0	Регистр времени	RW
RRTC_DATE	0x04	0x0	Регистр даты	RW
RRTC_TALRM	0x08	0x0	Регистр времени будильника	RW
RRTC_DALRM	0x0C	0x0	Регистр даты будильника	RW
RRTC_CTRL	0x10	0x0	Регистр управления	RW
RRTC_REG0	0x20	0x0	Регистры общего назначения REG0	RW
RRTC_REG1	0x24	0x0	Регистры общего назначения REG1	RW
RRTC_REG2	0x28	0x0	Регистры общего назначения REG2	RW
RRTC_REG3	0x2C	0x0	Регистры общего назначения REG3	RW
RRTC_REG4	0x30	0x0	Регистры общего назначения REG4	RW
RRTC_REG5	0x34	0x0	Регистры общего назначения REG5	RW
RRTC_REG6	0x38	0x0	Регистры общего назначения REG6	RW
RRTC_REG7	0x3C	0x0	Регистры общего назначения REG7	RW
RRTC_REG8	0x40	0x0	Регистры общего назначения REG8	RW
RRTC_REG9	0x44	0x0	Регистры общего назначения REG9	RW
RRTC_REG10	0x48	0x0	Регистры общего назначения REG10	RW
RRTC_REG11	0x4C	0x0	Регистры общего назначения REG11	RW
RRTC_REG12	0x50	0x0	Регистры общего назначения REG12	RW
RRTC_REG13	0x54	0x0	Регистры общего назначения REG13	RW
RRTC_REG14	0x58	0x0	Регистры общего назначения REG14	RW
RRTC_REG15	0x5C	0x0	Регистры общего назначения REG15	RW

3.10.5.1 Регистр времени

RRTC_TIME – регистр установки времени. Используется BCD-кодировка.

Установка недопустимых значений может привести к неопределённому результату. Для установки новых значений необходимо сбросить бит EN регистра RRTC_CTRL. В противном случае результат может быть непредсказуемым. Используемые поля указаны в таблице 66.

Таблица 66 – Поля регистра RRTC_TIME

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:27		Зарезервировано	
DOW	26:24	0x0	День недели. Допустимые значения в двоичном виде: 0b001 – понедельник; 0b010 – вторник; 0b011 – среда; 0b100 – четверг; 0b101 – пятница; 0b110 – суббота; 0b111 – воскресенье	RW
TH	23:22	0x0	Поле десятков часов. Доступные значения от 0 до 2	RW
H	21:18	0x0	Поле единиц часов. Допустимые значения: TH = 0 – От 0 до 9; TH = 2 – От 0 до 3	RW
TM	17:15	0x0	Поле десятков минут. Допустимые значения от 0 до 5	RW
M	14:11	0x0	Поле единиц минут. Допустимые значения от 0 до 9	RW
TS	10:8	0x0	Поле десятков секунд. Допустимые значения от 0 до 5	RW
S	7:4	0x0	Поле единиц секунд. Допустимые значения от 0 до 9	RW
Зарезервировано	3:0		Зарезервировано	

3.10.5.2 Регистр даты

RRTC_DATE – регистр установки даты. Используется BCD-кодировка. Установка недопустимых значений может привести к неопределённому результату. Для установки новых значений необходимо сбросить бит EN регистра RRTC_CTRL. В противном случае результат может быть непредсказуемым. Используемые поля указаны в таблице 66.

Таблица 67 – Поля регистра RRTC_DATE

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:27		Зарезервировано	
TC	26:23	0x0	Поле десятков веков. Допустимые значения от 0 до 9	RW
C	22:19	0x0	Поле единиц веков. Допустимые значения от 0 до 9	RW
TY	18:15	0x0	Поле десятков годов. Допустимые значения от 0 до 9	RW
Y	14:11	0x0	Поле единиц годов. Допустимые значения от 0 до 9	RW
TM	10	0x0	Поле десятков месяцев. Допустимые значения от 0 до 1	RW
M	9:6	0x0	Поле единиц месяцев. Допустимые значения: TM = 0 – от 0 до 9; TM = 1 – от 0 до 2	RW
TD	5:4	0x0	Поле десятков дней. Допустимые значения: {TM,M} != 03 – от 0 до 3; {TM,M} == 03 – от 0 до 2	RW
D	3:0	0x0	Поле единиц дней. Допустимые значения: TD = 2 – от 0 до 3; TD = 3 – от 0 до 1	RW

3.10.5.3 Регистр времени будильника

RRTC_TALRM – регистр хранит время, при совпадении которого со значением регистра RRTC_TIME, будет сгенерировано соответствующее прерывание. Структура регистра полностью идентична регистру RRTC_TIME за исключением полей управления. Используемые поля указаны в таблице 68.

Таблица 68 – Поля регистра RRTC_TALRM

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CDOW	31	0x0	Разрешает сравнения дней недели, когда установлен	RW
CH	30	0x0	Разрешает сравнения часов, когда установлен	RW
CM	29	0x0	Разрешает сравнения минут, когда установлен	RW
CS	28	0x0	Разрешает сравнения секунд, когда установлен	RW
Зарезервировано	27		Зарезервировано	
DOW	26:24	0x0	День недели. Допустимые значения в двоичном виде: 0b001 – понедельник; 0b010 – вторник; 0b011 – среда; 0b100 – четверг; 0b101 – пятница; 0b110 – суббота; 0b111 – воскресенье	RW
TH	23:22	0x0	Поле десятков часов. Доступные значения от 0 до 2	RW
H	21:18	0x0	Поле единиц часов. Допустимые значения: TH = 0 – от 0 до 9; TH = 2 – от 0 до 3	RW
TM	17:15	0x0	Поле десятков минут. Допустимые значения от 0 до 5	RW
M	14:11	0x0	Поле единиц минут. Допустимые значения от 0 до 9	RW
TS	10:8	0x0	Поле десятков секунд. Допустимые значения от 0 до 5	RW
S	7:4	0x0	Поле единиц секунд. Допустимые значения от 0 до 9	RW
Зарезервировано	3:0		Зарезервировано	

3.10.5.4 Регистр даты будильника

RRTC_DALRM – регистр хранит дату, при совпадении которой со значением регистра RRTC_DATE, будет сгенерировано соответствующее прерывание. Структура регистра полностью идентична регистру RRTC_DATE за исключением полей управления. Используемые поля указаны в таблице 68.

Таблица 69 – Поля регистра RRTC_DALRM

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31		Зарезервировано	
CC	30	0x0	Разрешает сравнения веков, когда установлен	RW
CY	29	0x0	Разрешает сравнения годов, когда установлен	RW
CM	28	0x0	Разрешает сравнения месяцев, когда установлен	RW
CD	27	0x0	Разрешает сравнения дней, когда установлен	RW
TC	26:23	0x0	Поле десятков веков. Допустимые значения от 0 до 9	RW
C	22:19	0x0	Поле единиц веков. Допустимые значения от 0 до 9	RW
TY	18:15	0x0	Поле десятков годов. Допустимые значения от 0 до 9	RW
Y	14:11	0x0	Поле единиц годов. Допустимые значения от 0 до 9	RW
TM	10	0x0	Поле десятков месяцев. Допустимые значения от 0 до 1	RW
M	9:6	0x0	Поле единиц месяцев. Допустимые значения: TM = 0 – от 0 до 9; TM = 1 – от 0 до 2	RW
TD	5:4	0x0	Поле десятков дней. Допустимые значения: $\{TM, M\} = 03$ – от 0 до 3; $\{TM, M\} = 03$ – от 0 до 2	RW
D	3:0	0x0	Поле единиц дней. Допустимые значения: TD = 2 – от 0 до 3; TD = 3 – от 0 до 1	RW

3.10.5.5 Регистр управления

RRTC_CTRL – регистр управления модулем. Используемые поля указаны в таблице 69.

Таблица 70 – Поля регистра RRTC_CTRL

Обозначение	Разряды	Начальное значение	Назначение	Доступ
EN	31	0x0	Модуль включён и производит отсчёт времени, когда установлен	RW
ALRM	30	0x0	Установленный бит свидетельствует о совпадении одного разрешенного или всех разрешенных полей будильника. Бит необходимо сбрасывать при помощи управляющего ПО	RW
INTE	29	0x0	Бит разрешения прерывания от будильника. Прерывание формируется при наличии установленного бита ALRM	RW
FLAG	28	–	Сигнал проведения синхронизации между тактовыми доменами. После записи в любой регистр и пока данный бит читается равным «1», запрещено выполнять любую новую запись	RO
ALRM_PAD	27	0x0	Запись «1» – сброс признака установки активного уровня ALARM_PAD	WO
Зарезервировано	26:0		Зарезервировано	

3.10.5.6 Регистры общего назначения RRTC_REG0 – RRTC_REG15.

В адресном пространстве модуля доступно 16 32х-разрядных регистра. Они могут использоваться в произвольных целях. Начальное значения регистров при включении питания 0x00000000. Регистры располагаются в батарейном домене и сохраняют свое значение при отключении системного питания и при внешнем сбросе.

3.11 Программируемый контроллер прерываний (EPIC)

Контроллер прерываний реализует поддержку 32 линий прерываний от внешних устройств. Модуль обеспечивает формирование прерываний как по уровню, так и при детектировании фронта(спада) на линии прерывания. В модулях микроконтроллера реализован механизм «зашелкивания» прерываний, поэтому рекомендуется использовать режим установки прерываний по уровню. Исключением является модуль прерываний портов ввода-вывода, сконфигурированный для работы по уровню. Прерывания по фронту следует использовать для модулей, у которых нет собственного механизма сброса прерывания, например, АЦП. Перечень линий прерывания для устройств, подключённых к контроллеру прерываний, указан в таблице 70.

Таблица 71 – Перечень линий прерывания для устройств, подключенных к контроллеру прерываний

Номер линии прерывания	Имя источника прерывания
0	Timer32_0
1	USART_0
2	USART_1
3	SPI_0
4	SPI_1
5	GPIO
6	I ² C_0
7	I ² C_1
8	Сторожевой таймер
9	Timer16_0
10	Timer16_1
11	Timer16_2
12	Timer32_1
13	Timer32_2
14	SPIFI
15	RTC
16	EEPROM
17	Сторожевой таймер шины (периферийные устройства)
18	Сторожевой таймер шины (SPIFI)

Номер линии прерывания	Имя источника прерывания
19	Сторожевой таймер шины (EEPROM)
20	ПДП
21	Монитор частоты
22	Монитор напряжения AVCC (ниже порога)
23	Монитор напряжения AVCC (выше порога)
24	Монитор напряжения VCC (ниже порога)
25	Монитор напряжения VCC (выше порога)
26	Недостаточное напряжение батареи
27	BrownOut детектор
28	Монитор температуры
29	АЦП
30	DAC0
31	DAC1

В таблице 72 представлены адреса, обозначения и наименования регистров контроллера прерывания, доступных по интерфейсу APB. Поля регистров контроллера прерывания показаны в таблицах 73 – 79. При чтении полей, доступных только для записи, возвращается в «0». Все зарезервированные поля регистров физически не представлены в модуле.

Таблица 72 – Регистры контроллера прерываний

Обозначение	Смещение	Начальное значение	Назначение	Доступ
MASK_EDGE_SET	0x00	0x0	Установка маски прерываний по фронту	RW1S
MASK_EDGE_CLEAR	0x04	0x0	Сброс маски прерываний по фронту	RW1C
MASK_LEVEL_SET	0x08	0x0	Установка маски прерываний по уровню	W1S
MASK_LEVEL_CLEAR	0x0C	0x0	Сброс маски прерываний по уровню	W1C
CLEAR	0x18	0x0	Сброс флагов в статусе прерываний	W1C
STATUS	0x1C	0x0	Статус прерываний	RO
RAW_STATUS	0x20	0x0	Текущее состояние линий прерываний	RO

Таблица 73 – Поля регистра MASK_EDGE_SET

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MASK_EDGE_SET	31:0	0x0	При чтении – текущее состояние масок прерываний по фронту. При записи, «1» разрешает прерывание по фронту соответствующего источника прерываний.	RW1S

Таблица 74 – Поля регистра MASK_EDGE_CLEAR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MASK_EDGE_CLEAR	31:0	0x0	При чтении – текущее состояние масок прерываний по фронту. При записи, «1» запрещает прерывание по фронту соответствующего источника прерываний.	RW1C

Таблица 75 – Поля регистра MASK_LEVEL_SET

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MASK_LEVEL_SET	31:0	0x0	Запись «1» разрешает прерывание по уровню соответствующего источника прерываний.	W1S

Таблица 76 – Поля регистра MASK_LEVEL_CLEAR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MASK_LEVEL_CLEAR	31:0	0x0	Запись «1» запрещает прерывание по уровню соответствующего источника прерываний.	W1C

Таблица 77 - Поля регистра CLEAR

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CLEAR	31:0	0x0	Запись «1» сбрасывает флаг соответствующего источника прерывания.	W1C

Таблица 78 – Поля регистра STATUS

Обозначение	Разряды	Начальное значение	Назначение	Доступ
STATUS	31:0	0x0	Текущее состояние регистра прерываний с учетом масок для прерываний по фронту	RO

Таблица 79 – Поля регистра RAW_STATUS

Обозначение	Разряды	Начальное значение	Назначение	Доступ
RAW_STATUS	31:0	0x0	Текущее состояние линий – источников прерываний	RO

3.12 Блок вычисления контрольных сумм (CRC32)

Блок предназначен для вычисления циклической контрольной суммы (CRC) массива данных без использования вычислительных ресурсов центрального процессора.

Структурная схема блока представлена на рисунке 57.

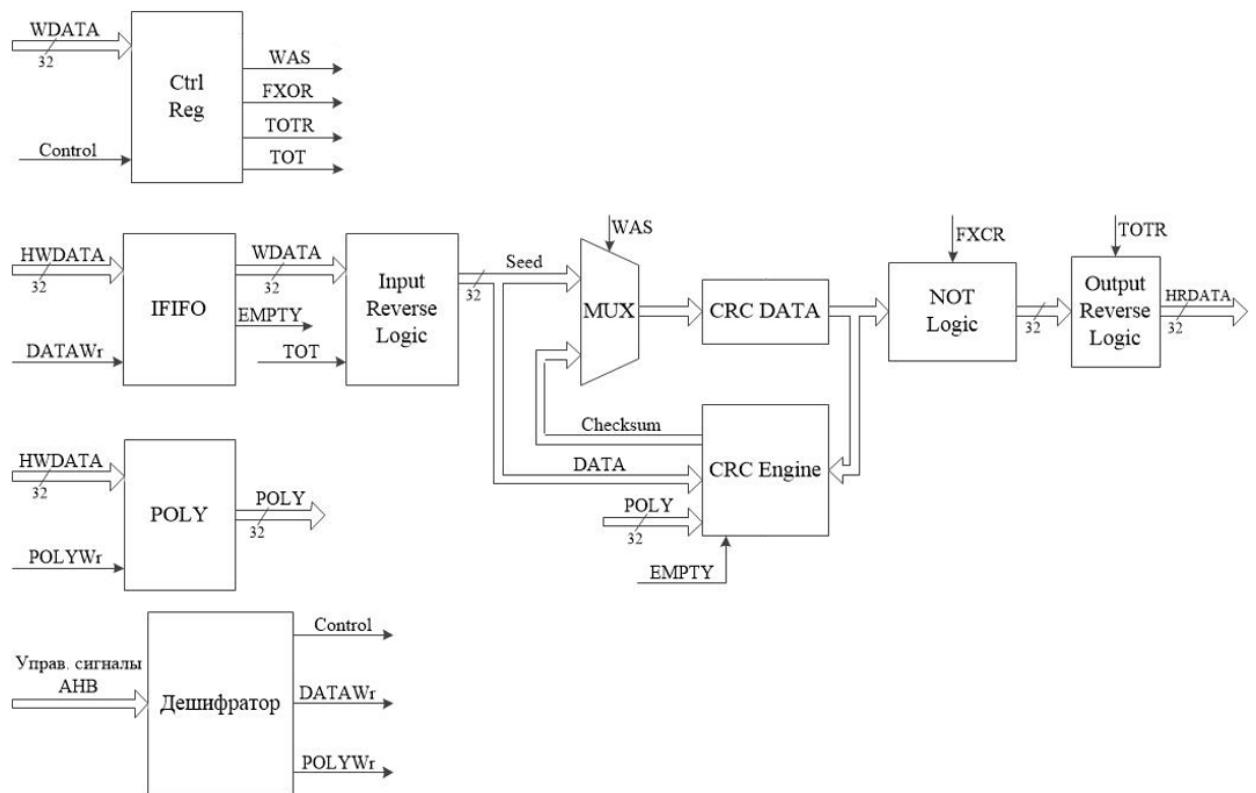


Рисунок 57 – Схема блока CRC

Контроллер состоит из следующих компонентов:

- дешифратора системных сигналов шины АHB,
- регистра управления CTRL,
- регистра полинома POLY,
- входного буфера IFIFO объемом 16 слов по 32 бита,
- логики перестановки битов входных данных Input Reverse Logic,
- устройства кодирования CRC_Engine с мультиплексором MUX и сдвиговым регистром CRC DATA,
- логики инверсии NOT Logic и логики перестановки битов выходных данных Output Reverse Logic.

Дешифратор системных сигналов формирует сигналы чтения и записи всех регистров контроллера в соответствии с адресным пространством. Регистр управления CTRL управляет всеми настройками процесса вычисления суммы. Регистр полинома

POLY предназначен для хранения значения полинома. Входной буфер IFIFO осуществляет буферизацию входных данных, по которым будет считаться контрольная сумма. Этот буфер, а также регистры управления и полинома подключены напрямую к шине HWDATA. При полном заполнении буфера активизируется сигнал HREADY_o, останавливающий транзакцию по шине АНВ. Логика перестановки битов входных данных предназначена для перестановки битов и байтов слова входных данных и управляет битами поля TOT регистра управления. Устройство кодирования CRC_Engine с мультиплексором MUX и сдвиговым регистром CRC DATA производят вычисление контрольной суммы последовательности входных данных. Вычисленная контрольная сумма записывается в выходной регистр логики перестановки битов выходных данных Output Reverse Logic через логику инверсии NOT Logic, которая инвертирует или не инвертирует выходные данные в зависимости от значения бита FXOR регистра управления. Логика перестановки битов выходных данных предназначена для перестановки битов и байтов слова вычисленной контрольной суммы и управляет битами поля TOTR регистра управления. Выходные биты контрольной суммы подключены напрямую к шине HRDATA.

Карта регистров блока приведена в таблице 80. Запись данных в регистр данных может осуществляться в 8, 16 или 32-разрядном режиме. Доступ к остальным регистрам осуществляется только в 32-разрядном режиме.

Таблица 80 – Регистры блока вычисления контрольных сумм

Обозначение	Смещение	Назначение	Доступ
DATA	0x00	Регистр данных	RW
POLY	0x04	Регистр полинома	RW
CTRL	0x08	Регистр управления	RW

Регистр управления определяет режимы работы блока. Описание битов регистра управления приведено в таблице 81.

Таблица 81 – Описание полей регистра управления блока CTRL

Обозначение	Разряды	Начальное значение	Назначение	Доступ
TOT	31:30	0x0	Перестановка битов/байтов входных данных: 0b00 – перестановка выключена; 0b01 – биты в байтах перестанавливаются (бит 7 с битом 0, бит 15 с битом 8, с остальными байтами так же), байты НЕ перестанавливаются; 0b10 – перестанавливаются и биты, и байты 0b11 – биты в байтах НЕ перестанавливаются, байты перестанавливаются (меняются местами байты 3 и 0, 2 и 1);	RW
TOTR	29:28	0x0	Перестановки битов/байтов выходных данных: 0b00 – перестановка выключена; 0b01 – биты в байтах перестанавливаются (бит 7 с битом 0, бит 6 с битом 1 и т.д.; бит 15 с битом 8, бит 14 с битом 9 и т.д., с остальными байтами так же), байты НЕ перестанавливаются; 0b10 – перестанавливаются и биты, и байты 0b11 – биты в байтах НЕ перестанавливаются, байты перестанавливаются (меняются местами байты 3 и 0, 2 и 1);	RW
Зарезервировано	27	Зарезервировано		
FXOR	26	0x0	Инверсия контрольной суммы. Некоторые протоколы подсчета контрольной суммы требуют инверсии вычисленного значения контрольной суммы (выполняется операция XOR со значением 0xFFFFFFFF): 0 – инверсия выключена; 1 – инверсия включена (операция XOR выполняется)	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
WAS	25	0x0	Назначение регистра данных: 0 – записываем данные; 1 – записываем начальное значение	RW
Зарезервировано	24:1	Зарезервировано		
Busy	0	0x0	Бит занятости автомата: 0 – автомат закончил вычисления; 1 – автомат занят. После записи в регистр данных слова для вычисления контрольной суммы бит занятости перейдет в состояние единицы через один такт после такта записи. То есть чтение регистра управления, идущее на шине АНВ сразу на следующем такте после записи данных, вернет результат с нулевым (еще не обновленным) значением бита Busy	RO

Регистр полинома служит для хранения значения полинома, по которому будет считаться контрольная сумма.

Регистр данных при чтении выдает содержимое регистра вычисленной контрольной суммы. При записи в этот регистр назначение принимаемых данных различное и определяется значением бита WAS регистра управления модуля: если этот бит равен 1, то записанное в регистр данных 32-разрядное значение является начальным значением контрольной суммы, если бит WAS равен нулю, то пишутся обычные данные.

Для вычисления контрольной суммы необходимо выполнить следующую последовательность действий:

- установить необходимые значения битов перестановки и инверсии (поля ТОТ, TOTR и FXOR);
- записать в регистр полинома требуемое 32-разрядное значение полинома;
- установить в единицу бит WAS регистра управления;
- записать в регистр данных 32-разрядное начальное значение;
- сбросить в ноль бит WAS регистра управления;
- записать необходимое количество данных в регистр данных;
- прочитать из регистра данных вычисленное значение контрольной суммы.

3.13 Модуль SPI

Модуль контроллера SPI обеспечивает взаимодействие с различными периферийными устройствами такими, как блоки памяти, датчики температуры, датчики давления, аналоговые преобразователи, часы реального времени, дисплеи и любые SD карты, поддерживающие последовательный режим обмена.

В режиме ведущего устройства, модуль контроллера SPI управляет последовательным сигналом синхронизации и сигналом выбора ведомого устройства с поддержкой режима несколько ведущий устройств. Последовательный сигнал синхронизации формируется из сигнала синхронизации подсистемы. Контроллер инициирует передачу сообщений, используя для этого до 4-х выходных сигналов выбора ведомого, которые могут быть внешне расширены. Контроллер считывает и записывает данные из ведомых устройств, используя 32 битный регистр данных чтения/записи.

В режиме ведомого устройства, модуль контроллера SPI принимает сигнал синхронизации от внешнего устройства и использует его в качестве сигнала SPI_Ref_Clk для синхронизации захвата данных. В режиме ведомого устройства в контроллере реализован программируемый механизм распознавания начала передачи, при котором контроллер активируется удержанием сигнала выбора ведомого SS.

Регистры FIFO для чтения и записи обеспечивают буферизацию между интерфейсом ввода/вывода SPI и ПО модуля, взаимодействующего с ним через интерфейс APB. Буферы FIFO используются как в режиме ведомого, так и в режиме ведущего устройства.

Основные свойства модуля контроллера SPI:

- четырехпроводная шина – MOSI, MISO, SCLK и SS;
- для режима ведущего устройства могут быть выбраны до четырех ведомых устройств;
- работа в дуплексном режиме (передача и прием данных одновременно);
- порты чтения/записи Rx/Tx FIFO отображены в общее адресное пространство;
- программируемый пороговый уровень FIFO со своим статусом и прерыванием;
- автоматический и ручной режимы выбора ведомого устройства;
- программируемая задержка сигналов SS и MOSI;
- программируемый режим детектирования старта;
- программируемая фаза и полярность (CPHA, CPOL);
- поддержка прерываний.

3.13.1 Структурная схема модуля контроллера SPI

В состав модуля входят следующие компоненты:

- APB интерфейс;
- регистры FIFO передатчика и приемника;
- регистр принимаемых данных приема (RxFIFO);
- регистр передаваемых данных (Tx FIFO);
- система тактирования;
- схема управления;
- блок ведущего;
- блок ведомого.

Структурная схема SPI контроллера показана на рисунке 58.

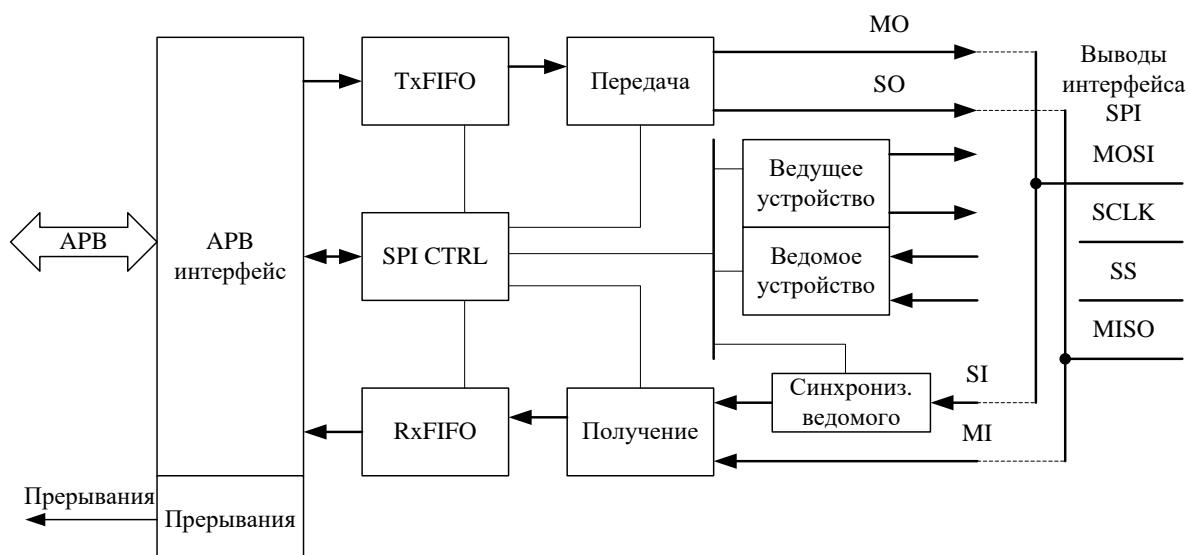


Рисунок 58 – Функциональная схема SPI контроллера

3.13.1.1 APB интерфейс

Для доступа к регистрам и FIFO используется 32-битный APB интерфейс. Все регистровые передачи имеют разрядность 32 бита.

3.13.1.2 Регистры FIFO передатчика и приемника

Каждый регистр FIFO имеет 8 байт. Программа осуществляет чтение и запись регистров FIFO через порты данных регистров, отображенных в общее адресное пространство. Регистры FIFO выполняют функцию моста между двумя доменами тактовой частоты: APB интерфейс и опорная частота контроллера SPI_Ref_Clk. Программа записывает данные в регистр Tx FIFO в домене тактовой частоты APB и

контроллер считывает данные из регистра TxFIFO в домене тактовой частоты SPI_Ref_Clk. Контроллер заполняет регистр RxFIFO в домене тактовой частоты SPI_Ref_Clk, а программа считывает регистр RxFIFO в домен тактовой частоты APB.

3.13.1.3 Регистр принимаемых данных приема (RxFIFO)

Если контроллер пытается сдвинуть данные в полностью заполненный регистр принимаемых данных, происходит потеря информации и устанавливается флаг переполнения [RX_OVERFLOW]. Данные не могут быть добавлены в заполненный регистр принимаемых данных приема. Программа записывает 1 для очистки бита [RX_OVERFLOW].

3.13.1.4 Регистр передаваемых данных (Tx FIFO)

Если программа пытается записывать данные в полностью заполненный регистр передаваемых данных передачи, запись будет проигнорирована. Данные не могут быть добавлены в заполненный регистр передаваемых данных. Бит [TX_FIFO_FULL] удерживается, пока из FIFO не будет прочитан очередной байт и помещен в сдвиговый регистр.

3.13.1.5 Система тактирования

Подсистема тактирования формирует опорный тактовый сигнал для SPI контроллера. Тактовый сигнал SPI_Ref_Clk используется для тактирования логических схем контроллера и генератором скорости передачи данных для формирования тактового сигнала SCLK в режиме ведущего устройства.

3.13.2 Функциональное описание

3.13.2.1 Режим ведущего устройства

В режиме ведущего устройства SPI контроллер может передавать данные ведомому устройству или инициировать прием данных от ведомого устройства. Контроллер выбирает одно ведомое устройство в данный момент, используя одну из четырех линий выбора ведомого устройства. Если необходимо подключить к ведущему устройству более четырех ведомых устройства, это можно сделать, используя внешний декодер.

Передача данных

Ведущее устройство управляет тактовым сигналом SCLK и сигналом MOSI. Данные, которые необходимо передать, программа записывает в регистр TxFIFO, используя запись в регистр, и затем выгружает для передачи аппаратными средствами контроллера. Данные поступают на вывод выходных данных ведущего устройства (MOSI). Передача осуществляется до тех пор, пока есть данные в регистре TxFIFO.

Данные принимаются последовательно на выводе MISO и загружаются по 8 бит в регистр RxFIFO. Программа считывает регистр RxFIFO, используя регистры чтения. На каждые N байтов, записанных в регистр TxFIFO, приходится N байтов, сохраненных в регистре RxFIFO, которые должен быть считан программой до начала следующей передачи.

Выбор ведомого устройства может быть реализован аппаратно или программно. При установке бита CONFIG.Manual_CS = 1 контроллер не управляет сигналами выбора ведомого и этот функционал должен быть реализован программно.

При установке CONFIG.Manual_CS=0 контроллер формирует автоматически сигналы выбора ведомого. В этом случае сигнал выбора ведомого определяется регистрами CONFIG.CS и CONFIG.PERI_SEL:

- если CONFIG.PERI_SEL = 1, то при передаче на выводах SPIx_n_ss_out формируются значения, соответствующие полям CONFIG.CS:
 - 1) На вывод SPIx_n_ss_out_0 значение CONFIG.CS[0]
 - 2) На вывод SPIx_n_ss_out_1 значение CONFIG.CS[1]
 - 3) На вывод SPIx_n_ss_out_2 значение CONFIG.CS[2]
 - 4) На вывод SPIx_n_ss_out_3 значение CONFIG.CS[3]
- если CONFIG.PERI_SEL = 0, то при передаче будет выбран только один активный вывод:
 - 1) При CONFIG.CS = 0b1110 используется вывод SPIx_n_ss_out_0;
 - 2) при CONFIG.CS = 0b1101 используется вывод SPIx_n_ss_out_1;
 - 3) при CONFIG.CS = 0b1011, то используется вывод SPIx_n_ss_out_2;
 - 4) при CONFIG.CS = 0b0111, то используется вывод SPIx_n_ss_out_3;
 - 5) при CONFIG.CS = 0b1111, то ведомые не выбираются.

Примечание - При работе в режиме ведущего необходимо учитывать настройки выводов микросхемы SPI0_n_ss_in и SPI1_n_ss_in и конфигурировать их таким образом, чтобы в момент передачи сигнал n_ss_in контроллера был установлен в 1. Для

функционирования схемы с несколькими ведущими необходимо учитывать, что устройство, находящее в режиме ведущего, нельзя выбирать в качестве ведомого, так как это приводит к ошибке MODE_FAIL.

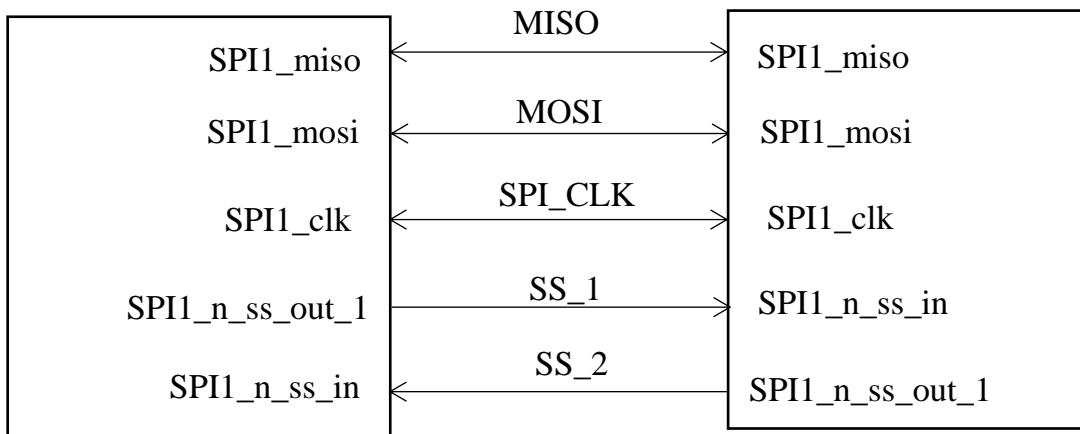


Рисунок 59 – Схема подключения двух устройств в режиме нескольких ведущих

3.13.2.2 Режим ведомого устройства

В режиме ведомого устройства, контроллер получает сообщения от внешнего ведущего устройства и выдает ответ. Контроллер переходит в режим ведомого устройства, когда конфигурационные биты установлены следующим образом: CONFIG [MODE_SEL] = 0 и ENABLE [SPI_EN] = 1.

По сигналу SCLK происходит защелкивание данных на входе MOSI. Контроллер игнорирует данные на входе MOSI, если на вывод выбора ведомого устройства подан высокий логический уровень (неактивное состояние). Сигнал выбора ведомого устройства должен удерживаться в активном состоянии на время передачи. Контроллер устанавливает бит статуса STATUS [MODE_FAIL], если сигнал выбора ведомого устройства прерывается во время передачи. Программа обрабатывает прерывание [MODE_FAIL] следующим образом, прерывает передачу, сбрасывает контроллер и начинает передачу заново.

Данные, которые должны быть переданы ведущему устройству, записываются в регистр TxFIFO программным обеспечением, и затем контроллер преобразует данные в последовательный поток и выдает на вывод MISO. Передача продолжается до тех пор, пока не опустошен регистр TxFIFO и сигнал выбора ведомого устройства удерживается в активном состоянии (низкий уровень).

Синхронизация

Входной сигнал выбор ведомого устройства должен быть синхронизирован с входным сигналом SCLK. Контроллер работает с частотой SPI_Ref_Clk. Входные сигналы синхронизируются и анализируются с частотой SPI_Ref_Clk.

Обнаружение слова

Процедура определения начала слова выполняется с частотой SPI_Ref_Clk. Обнаружение слова, когда контроллер включен: если контроллер переходит в активное состояние (из неактивного состояния) и в это время сигнал выбор ведомого устройства имеет низкий уровень (активное состояние), контроллер будет пропускать информацию и ждать, когда сигнал SCLK станет неактивным (граница слова) прежде, чем захватить данные. Контроллер отсчитывает количество неактивных состояний сигнала SCLK с частотой SPI_Ref_Clk. Новое слово считается принятым, когда значение счетчика достигнет значения, запрограммированного в битовом поле [Slave_Idle_coun].

Обнаружение слова, когда удерживается сигнал выбор ведомого устройства: когда контроллер находится в активном состоянии и сигнал выбор ведомого устройства находится в высоком состоянии (неактивный режим), контроллер считает, что слово начинается по следующему активному фронту сигнала SCLK после перехода сигнала SS в низкий уровень (активное состояние).

Примечание – Состояние, соответствующее началу передачи, необходимо удерживать по крайней мере в течение 4 периодов тактового сигнала SPI_Ref_Clk.

С небольшой вероятностью может произойти ложная синхронизация и повреждение передаваемого пакета, если режим ведомого устройства активизируется ближе к моменту начала передачи данных.

Избежать этой проблемы можно одним из следующих способов:

- обеспечить, чтобы внешнее ведущее устройство не начинало передачу данных, в течение 10 периодов сигнала SPI_Ref_Clk, после перехода ведомого устройства в активный режим;
- обеспечить, чтобы ведомое устройство было включено до активирования внешнего ведущего устройства;
- обеспечить, чтобы сигнал выбора ведомого устройства был не активен при включенном ведомом устройстве.

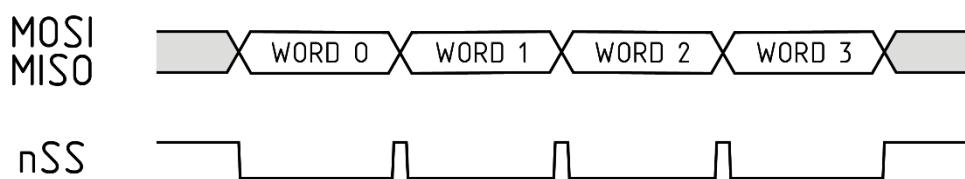
3.13.3 Фаза и полярность

Существует 4 комбинации для задания фазы и полярности управляющих бит [CLK_PH] и [CLK_POL] в регистре CONFIG. Эти параметры определяют активный фронт, ограничения на сигнал выбора ведомого устройства и тактовый сигнал SCLK. Параметр фазы тактового сигнала определяет состояние сигнала SS между словами и состояние SCLK, когда контроллер не передает биты. В таблице 82 приведены значения управляющих сигналов бит [CLK_PH] и [CLK_POL].

Таблица 82 – Управляющие сигналы [CLK_PH] и [CLK_POL]

	CLK_PH = 0		CLK_PH = 1	
	CLK_POL = 0	CLK_POL = 1	CLK_POL = 0	CLK_POL = 1
Управляющий фронт	отрицательный	положительный	положительный	отрицательный
Фронт выборки	положительный	отрицательный	отрицательный	положительный
Состояние сигнала SS между словами	активный		неактивный	
Состояние SCLK вне слова	активный		неактивный	

CLK_PH = 0



CLK_PH = 1

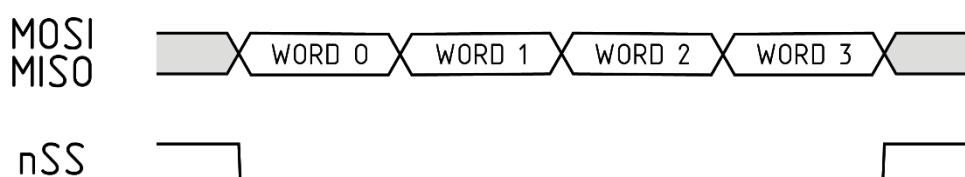


Рисунок 60 – Диаграммы сигналов SPI при передаче пакета в различных режимах работы

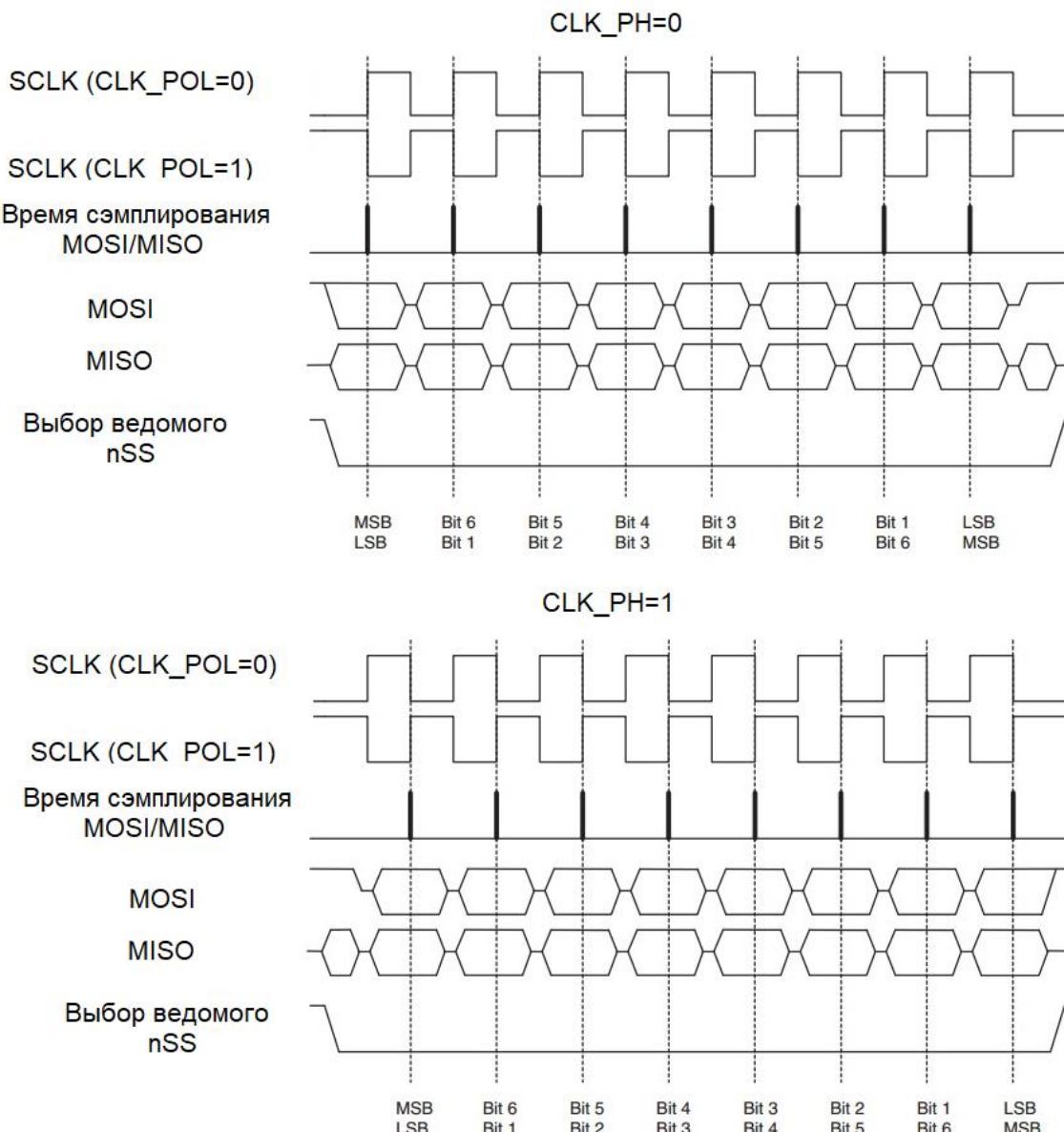


Рисунок 61 – Диаграммы сигналов SPI в различных режимах работы

В режиме ведущего устройства, значение бита фазы тактового сигнала [CLK_PH] в регистре CONFIG [CLK_PH] влияет на протокол обмена, посредством регистра DELAY следующим образом:

- CLK_PH = 0

SS активный: Ведущее устройство автоматически переводит выходы SS в неактивное состояние (высокое состояние) на время равное 1 периоду тактового сигнала шины APB_P.

Задержка между словами: Задержка между последним битом текущего слова и первым битом следующего слова Time = $(2 + [D_BTWN]) * SPI_Ref_Clk$. Минимальная задержка три периода SPI_Ref_Clk. Эта задержка позволяет выгрузить TxFIFO и подготовить к следующему преобразованию из параллельного кода в последовательный,

а также переключить сигнал выбора ведомого в неактивное состояние (высокий уровень).

- CLK_PH = 1

SS активный: Сигналы SS не переводятся в неактивное состояние между словами.

Задержка между словами: Минимальная задержка между последним битом текущего слова и первым битом следующего слова по умолчанию, составляет один период сигнала SPI_Ref_Clk (конфигурируется регистром DELAY). Это позволяет выгрузить TxFIFO и подготовить к следующему преобразованию из параллельного кода в последовательный.

3.13.4 Прерывания FIFO

Прерывания буферов RxFIFO и TxFIFO показаны на рисунке 62.

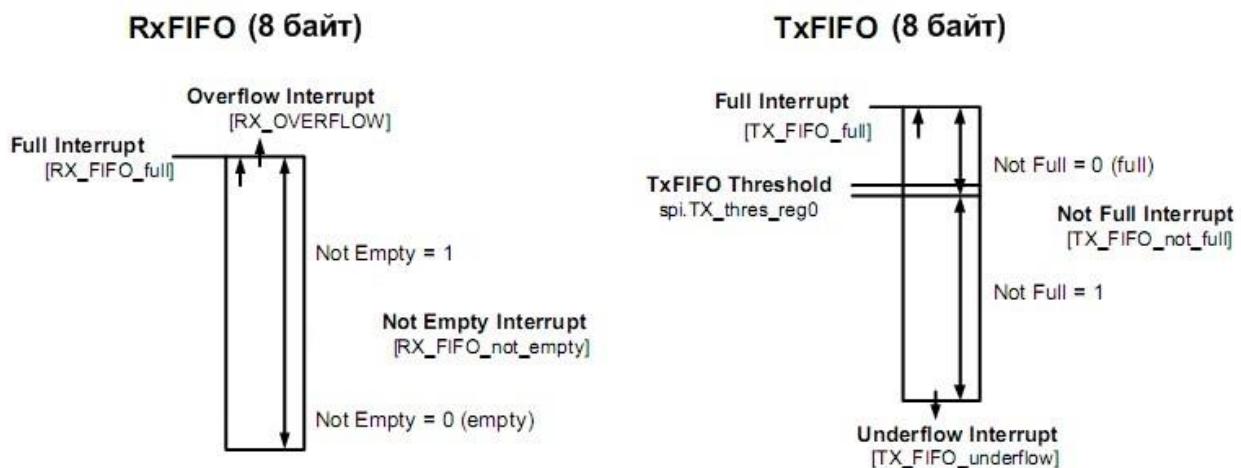


Рисунок 62 – Прерывания регистром RxFIFO и TxFIFO

Биты статуса прерываний (статические и динамические) отфильтровываются по регистру маски и затем поступают в системный контроллер прерываний. Регистром маски управляют регистры разрешения и запрета прерываний прерывания, показанные на рисунке 63.

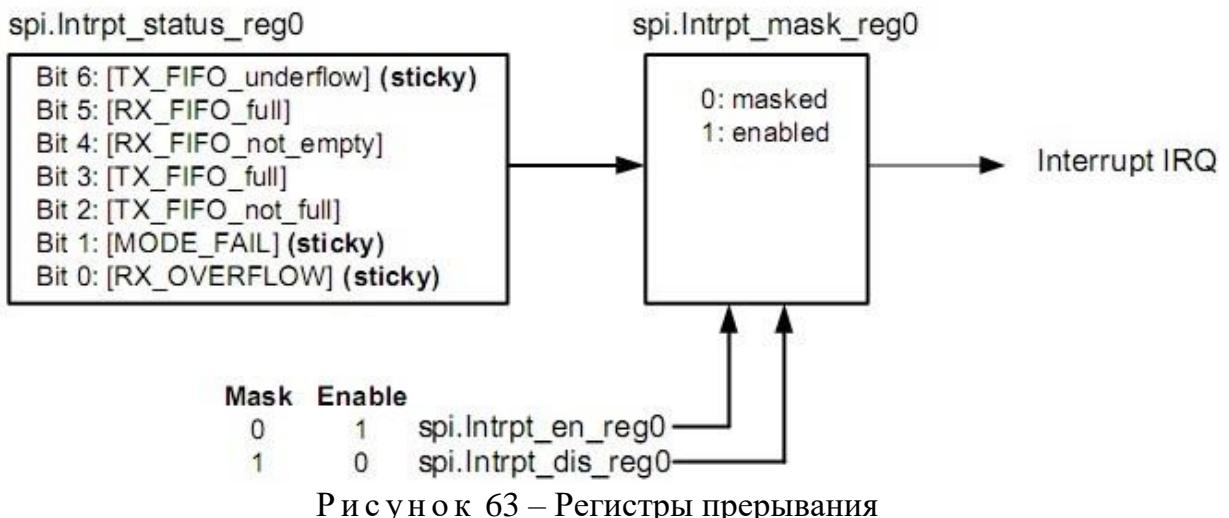


Рисунок 63 – Регистры прерывания

3.13.5 Порядок работы с контроллером SPI

При работе с контроллером необходимо выполнить следующие операции:

- Инициализация

- 1) Выключить контроллер (записать ENABLE.SPI_EN=0);
- 2) Очистить FIFO приемника (записать ENABLE.CLEAR_RX=1);
- 3) Сбросить маски прерываний (записать INT_DISABLE= 0x3F);
- 4) Сбросить флаги событий (записать STATUS= 0x3F);
- 5) Установить необходимые маски прерываний (записать в регистр INT_ENABLE);
- 6) Разрешить прерывания от используемого SPI контроллера (IRQ[3] для SPI_0 или IRQ[4] для SPI[1]) в программируемом контроллере прерываний;
- 7) Установить режим работы для ведущего:
 - а) установить необходимое значение CONFIG.Manual_CS,
 - б) установить необходимое значение CONFIG.CS,
 - в) установить необходимое значение CONFIG.BAUD_RATE_DIV
 - г) установить CONFIG.MODE_SEL=1,
 - д) установить необходимое значение CONFIG.CLK_PH,
 - е) установить необходимое значение CONFIG.CLK_CPOL
- 8) Установить режим работы для ведомого:
 - а) установить CONFIG.MODE_SEL=0,
 - б) установить необходимое значение CONFIG.CLK_PH,
 - в) установить необходимое значение CONFIG.CLK_CPOL

- Передача данных в режиме ведущего (на примере передачи по одному байту)
 - 1) записать в регистр TX_THR=1;
 - 2) записать данные в FIFO передатчика (запись в регистр TXDATA);
 - 3) ожидать окончание передачи байта (опрос бита STATUS. TX_NOT_FULL или появление соответствующего прерывания);
 - 4) прочитать принятые значения из FIFO приемника (регистр RXDATA)
- Передача данных в режиме ведомый
 - 1) записать данные в FIFO передатчика (запись в регистр TXDATA)
 - 2) ожидать передачу данных (опрос бита STATUS. RX_NOT_EMPTY или появление соответствующего прерывания)
 - 3) прочитать принятые значения из FIFO приемника (регистр RXDATA)
- Окончание работы
 - 1) Выключить контроллер (записать ENABLE.SPI_EN=0);
 - 2) Очистить FIFO приемника (записать ENABLE. CLEAR_RX=1);
 - 3) Сбросить флаги событий (записать STATUS=STATUS & 0x3F);
 - 4) Сбросить маски прерываний (записать INT_DISABLE= 0x3F);
 - 5) Сбросить флаги событий (записать STATUS= 0x3F);

3.13.6 Работа контроллера SPI совместно с контроллером ПДП

Контроллер SPI независимо от масок прерываний формирует запросы на чтение и запись контроллеру ПДП. Запрос на запись соответствует не полному TX_FIFO, запрос на чтение соответствует непустому RX_FIFO.

Для чтения из RX_FIFO с использованием ПДП необходимо установить:

```
CHx_CFG.READ_MODE = 0; // чтение в режиме периферийного уст-ва
CHx_CFG.READ_INCREMENT = 0; // чтение выполняется из одного адреса
CHx_CFG.READ_SIZE = 0; // чтение 1 байта
для SPI_0 - CHx_CFG.READ_REQUESTS = 3; // линия SPI_0
для SPI_1 - CHx_CFG.READ_REQUESTS = 4; // линия SPI_1
LEN = 0; // 1 байт
для SPI_0: SRC = 0x00083020;
для SPI_1: SRC = 0x00083420;
```

Пример чтения с использованием ПДП представлен ниже. На рисунке показаны следующие события:

- окончание передачи байта по SPI. Установка флага появления данных в RX_FIFO (RX_FIFO_NOT_EMPTY). Флаг является запросом на запись для ПДП.
- Формирование контроллером ПДП запроса на чтение по шине АНВ по адресу 0x0083020
- Чтение из контроллера SPI данных из RX_FIFO
- Запись полученных данных контроллером ПДП в ОЗУ

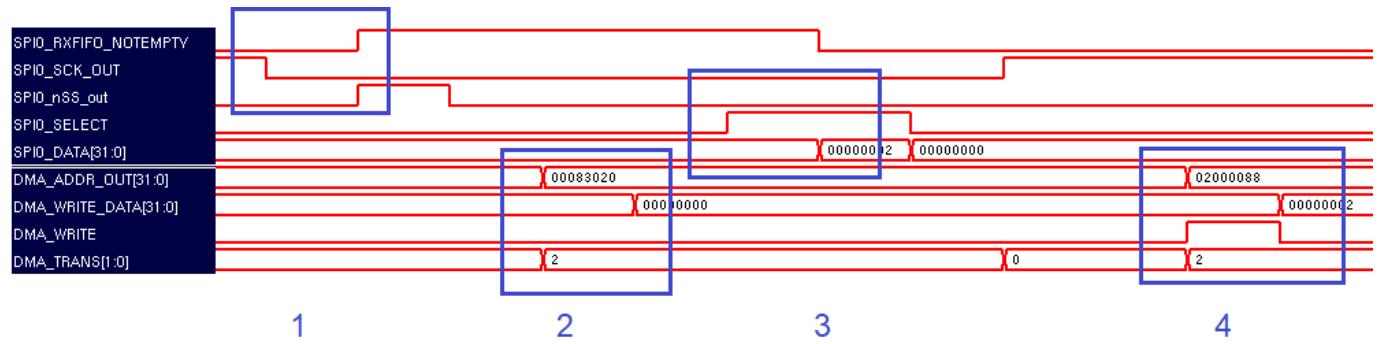


Рисунок 64 – Чтение данных с использованием ПДП

Для записи в TX_FIFO с использованием ПДП необходимо установить:

```
CHx_CFG.WRITE_MODE = 0; // запись в режиме периферийного уст-ва
CHx_CFG.WRITE_INCREMENT = 0; // запись выполняется в один адрес
CHx_CFG.WRITE_SIZE = 0; // запись 1 байта
для SPI_0 - CHx_CFG.WRITE_REQUESTS = 3; // линия SPI_0
для SPI_1 - CHx_CFG.WRITE_REQUESTS = 4; // линия SPI_1
LEN = 0; // 1 байт
для SPI_0: DST = 0x0008301C;
для SPI_1: DST = 0x0008341C;
```

Пример записи с использованием ПДП представлен ниже. На рисунке показаны следующие события:

- Чтение данных из ОЗУ. Ожидание запроса от SPI на запись данных
- окончание передачи байта по SPI. Установка флага наличие «места» для данных в TX_FIFO (TX_FIFO_NOT_FULL). Флаг является запросом на запись для ПДП.
- Формирование контроллером ПДП запроса на запись по шине АНВ по адресу 0x008301C
- Запись в контроллер SPI данных в TX_FIFO, сброс запроса к ПДП (TX_FIFO_NOT_FULL)

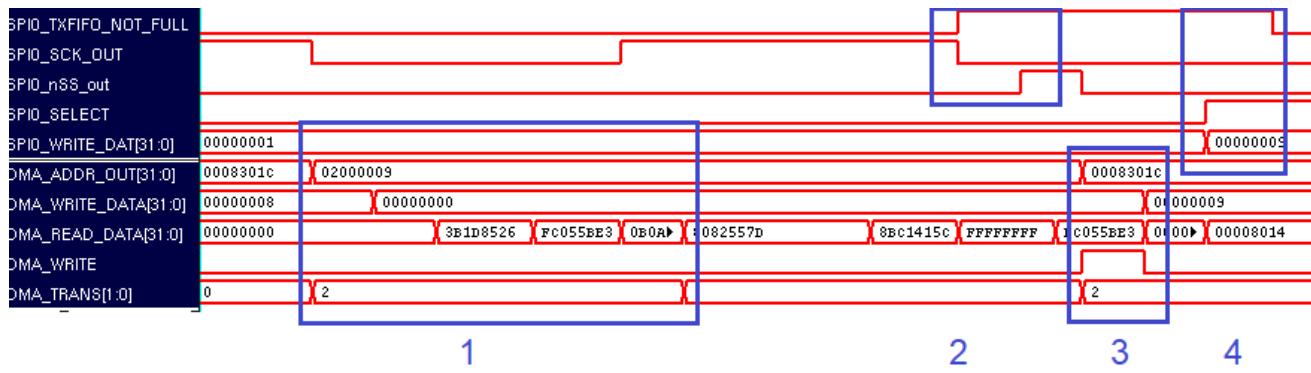


Рисунок 65 – Запись данных с использованием ПДП

3.13.7 Регистры модуля контроллера SPI

Описание регистров представлено в таблицах 83 – 95.

Таблица 83 – Регистры модуля контроллера SPI

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CONFIG	0x00	0x00000000	Регистр конфигурации SPI	*
STATUS	0x04	0x00000004	Регистр статуса	*
INT_ENABLE	0x08	0x00000000	Регистр разрешения прерываний	WO
INT_DISABLE	0x0C	0x00000000	Регистр запрета прерываний	WO
INT_MASK	0x10	0x00000000	Регистр текущих масок прерываний	RO
ENABLE	0x14	0x00000000	Регистр включения/выключения SPI	*
DELAY	0x18	0x00000000	Регистр задержек	RW
TXDATA	0x1C	0x00000000	Регистр передаваемых данных	WO
RXDATA	0x20	0x00000000	Регистр принимаемых данных	RO
SIC	0x24	0x000000FF	Регистр счетчика останова ведомого устройства	*
TX_THR	0x28	0x00000001	Регистр пороговых значений TX_FIFO	RW
ID	0x2C	0x01090100	Идентификационный номер модуля	RO

* Тип доступа зависит от поля/разряда регистра.

Таблица 84 – Регистр CONFIG модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:15		Зарезервировано	
MANUAL_CS	14	0x0	Выбор режима управления сигналом выбора ведомого: 1 – ручной режим; 0 – автоматический режим	RW
CS	13:10	0xF	Выбор ведомых устройств: 0bxxx0 – выбрано устройство 4; 0bx01 – выбрано устройство 1; 0bx011 – выбрано устройство 2; 0b0111 – выбрано устройство 3; 0b1111 – ведомые устройства не выбраны	RW
PERI_SEL	9	0x0	Выбор периферийного декодера: «1» – использование внешнего декодера; «0» – выбор только 1 из 4 устройств	
REF_CLK	8	0x0	Выбор опорной тактовой частоты 1 – не поддерживается; 0 – используется опорная частота SPI	RW
Зарезервировано	7:6		Зарезервировано	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
BAUD_RATE _DIV	5:3	0x0	Управляет скоростью передачи данных, задает коэффициент деления частоты spi_ref_clk: 0b000 – деление на 2; 0b001 – деление на 4; 0b010 – деление на 8; 0b011 – деление на 16; 0b100 – деление на 32; 0b101 – деление на 64; 0b110 – деление на 128; 0b111 – деление на 256	RW
CLK_PH (CPHA)	2	0x0	Выбор фазы тактового сигнала: 1 – тактовая частота SPI неактивна вне слова; 0 – тактовая частота SPI активна вне слова	RW
CLK_POL (CPOL)	1	0x0	Выбор полярности тактового сигнала вне слова: 1 – тактовый сигнал удерживается на высоком уровне; 0 – тактовый сигнал удерживается на низком уровне	RW
MODE_SEL (MSTREN)	0	0x0	Выбор режима: 1 – SPI в режиме ведущего устройства; 0 – SPI в режиме ведомого устройства	RW

Таблица 85 – Регистр STATUS модуля контроллера SPI

Обозначение	Разряды	Начальное Значение	Назначение	Доступ
Зарезервировано	31:16	Зарезервировано		
SPI_ACTIVE	15	0x0	Статус сеанса передачи: «1» – контроллер SPI в процессе обмена; «0» – контроллер SPI в состоянии ожидания обмена	RO
Зарезервировано	14:7	Зарезервировано		

Обозначение	Разряды	Начальное Значение	Назначение	Доступ
TX_FIFO_UNDERFLOW (IXR_TXUF)	6	0x0	Регистр TX FIFO опустошен: 1 – определение опустошения; 0 – опустошение не детектируется Устанавливается в режиме ведомого в случае, если к началу обмена в TX_FIFO нет данных	RC
RX_FIFO_FULL (IXR_RXFULL)	5	0x0	Регистр RX_FIFO заполнен: 1 – FIFO заполнен; 0 – FIFO не заполнен	RO
RX_FIFO_NOT_EMPTY (IXR_RXNEMPTY)	4	0x0	Регистр RX_FIFO не пустой: 1 – в FIFO есть хотя бы один байт; 0 – FIFO пустой	RO
TX_FIFO_FULL (IXR_TXFULL)	3	0x0	Регистр TX_FIFO заполнен: 1 – FIFO заполнен; 0 – FIFO не заполнен	RO
TX_FIFO_NOT_full (IXR_TXOW)	2	0x1	Регистр TX_FIFO не заполнен: 1 – регистр заполнен меньше чем THRESHOLD; 0 – регистр заполнен до значения THRESHOLD или больше	RO
MODE_FAIL (IXR_MODEF)	1	0x0	Напряжение на выводе n_ss_in не соответствует режиму работы SPI: 1 – если n_ss_in имеет низкий уровень в режиме ведомого устройства; 0 – корректная работа	RC

Обозначение	Разряды	Начальное Значение	Назначение	Доступ
RX _OVERFLOW (IXR_RXOVR)	0	0x0	Прерывание при переполнении RX_FIFO, значение сбрасывается при чтении: 1 – переполнение RX_FIFO; 0 – нет переполнения	RC
Примечание – Биты регистра [6:0] устанавливаются в «1», если произошло событие вызывающее прерывание.				

Таблица 86 – Регистр INT_ENABLE модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:7	Зарезервировано		
TX_FIFO_UNDERFLOW (IXR_TXUF)	6	0x0	Запись «1» устанавливает маску прерывания (разрешает прерывание) при опустошении TX_FIFO	WO
RX_FIFO_FULL (IXR_RXFULL)	5	0x0	Запись «1» устанавливает маску прерывания (разрешает прерывание) при заполнении RX_FIFO	WO
RX_FIFO_NOT_EMPTY (IXR_RXNEMPTY)	4	0x0	Запись «1» устанавливает маску прерывания (разрешает прерывание), когда RX_FIFO не пустой	WO
TX_FIFO_FULL (IXR_TXFULL)	3	0x0	Запись «1» устанавливает маску прерывания (разрешает прерывание), когда TX_FIFO заполнен	WO
TX_FIFO_NOT _full (IXR_TXOW)	2	0x0	Запись «1» устанавливает маску прерывания (разрешает прерывание) когда TX_FIFO не заполнен	WO
MODE_FAIL (IXR_MODF)	1	0x0	Запись «1» устанавливает маску прерывания (разрешает прерывание) при нарушении режима	WO
RX _OVERFLOW (IXR_RXOVR)	0	0x0	Запись «1» устанавливает маску прерывания (разрешает прерывание) при переполнении при приеме	WO

Таблица 87 – Регистр INT_DISABLE модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:7	Зарезервировано		
TX_FIFO_UNDERFLOW (IXR_TXUF)	6	0x0	Запись «1» сбрасывает маску прерывания (запрещает прерывание) при опустошении TX_FIFO	WO
RX_FIFO_FULL (IXR_RXFULL)	5	0x0	Запись «1» сбрасывает маску прерывания (запрещает прерывание) при заполнении RX_FIFO	WO
RX_FIFO_NOT_EMPTY (IXR_RXNEMPTY)	4	0x0	Запись «1» сбрасывает маску прерывания (запрещает прерывание), когда RX_FIFO не пустой	WO
TX_FIFO_FULL (IXR_TXFULL)	3	0x0	Запись «1» сбрасывает маску прерывания (запрещает прерывание), когда TX_FIFO заполнен	WO
TX_FIFO_NOT_FULL (IXR_TXOW)	2	0x0	Запись «1» сбрасывает маску прерывания (запрещает прерывание), когда TX_FIFO не заполнен	WO
MODE_FAIL (IXR_MODF)	1	0x0	Запись «1» сбрасывает маску прерывания (запрещает прерывание) при нарушении режима	WO
RX_OVERFLOW (IXR_RXOVR)	0	0x0	Запись «1» сбрасывает маску прерывания (запрещает прерывание) при переполнении при приеме	WO

Таблица 88 – Регистр INT_MASK модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:7	Зарезервировано		
TX_FIFO_UNDERFLOW (IXR_TXUF)	6	0x0	Текущее состояние маски прерывания при опустошении TX_FIFO: 0 – прерывание запрещено; 1 – прерывание разрешено	RO

Обозначение	Разряды	Начальное значение	Назначение	Доступ
RX_FIFO_FULL (IXR_RXFULL)	5	0x0	Текущее состояние маски прерывания при заполнении RX_FIFO: 0 – прерывание запрещено; 1 – прерывание разрешено	RO
RX_FIFO_NOT_EMPTY (IXR_RXNEMPTY)	4	0x0	Текущее состояние маски прерывания, когда RX_FIFO не пустой: 0 – прерывание запрещено; 1 – прерывание разрешено	RO
TX_FIFO_FULL (IXR_TXFULL)	3	0x0	Текущее состояние маски прерывания, когда TX_FIFO заполнен: 0 – прерывание запрещено; 1 – прерывание разрешено	RO
TX_FIFO_NOT_FULL (IXR_TXOW)	2	0x0	Текущее состояние маски прерывания, когда TX_FIFO не заполнен: 0 – прерывание запрещено; 1 – прерывание разрешено	RO
MODE_FAIL (IXR_MODF)	1	0x0	Текущее состояние маски прерывания при нарушении режима: 0 – прерывание запрещено; 1 – прерывание разрешено	RO
RX_OVERFLOW (IXR_RXOVR)	0	0x0	Текущее состояние маски прерывания при переполнении при приеме: 0 – прерывание запрещено; 1 – прерывание разрешено	RO

Таблица 89 – Регистр ENABLE модуля контроллера SPI

Название	Разряды	Начальное значение	Описание	Доступ
Зарезервировано	31:4	Зарезервировано		
CLEAR_RX_FIFO	3	0x0	Запись 1 при SPI_EN = 0 очищает буфер RX_FIFO.	RW
CLEAR_TX_FIFO	2	0x0	Запись 1 при SPI_EN = 0 очищает буфер TX_FIFO.	RW
SPI_EN (ENABLE)	0	0x0	SPI_Enable: 1 – включение модуля SPI; 0 – выключение модуля SPI	RW

Таблица 90 – Регистр DELAY модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:24	Зарезервировано		
D_BTWN (BTWN)	23:16	0x0	Задержка в периодах опорного тактового сигнала или внешнего тактового сигнала ext_clk между снятием сигнала выбора одного ведомого устройства и установкой сигнала выбора другого ведомого устройства	RW
D_AFTER (AFTER)	15:8	0x0	Задержка в периодах опорного тактового сигнала или внешнего тактового сигнала ext_clk между последним битом текущего слова и первым битом следующего слова	RW
D_INT (INIT)	7:0	0x0	Дополнительная задержка в периодах опорного тактового сигнала или внешнего тактового сигнала ext_clk между установкой сигнала n_ss_out в «0» и передачей первого бита	RW

Таблица 91 – Регистр TXDATA модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
TX_FIFO_data	31:0	0x0	Данные для TX_FIFO. Доступные биты данных: [7:0]	WO

Таблица 92 – Регистр RXDATA модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
RX_FIFO_data	31:0	0x0	Данные из RX_FIFO. Доступные биты данных: [7:0]	RO

Таблица 93 – Регистр SIC модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:8		Зарезервировано	
Slave_Idle_coun	7:0	0xFF	Модуль SPI в режиме ведомого устройства начинает передачу только когда тактовый сигнал sclk_in (внешнего ведущего устройства) не изменяется в течение количества периодов опорного тактового сигнала SPI заданного в этом поле или когда модуль SPI не активен	RW

Таблица 94 – Регистр TX_THR модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Threshold_of_TX_FIFO	31:0	0x1	Задает уровень, при котором TX_FIFO считается не заполненным и формируется прерывание	RW

Таблица 95 – Регистр ID модуля контроллера SPI

Обозначение	Разряды	Начальное значение	Назначение	Доступ
ID	31:0	0x01090100	Идентификационный номер модуля	RO

3.14 Модуль порта ввода/вывода общего назначения (GPIO)

Каждый модуль ввода/вывода (GPIO) имеет интерфейс APB и управляет 16 или 8 внешними выводами. Каждый из внешних выводов управляется индивидуально.

В таблице 96 перечислены регистры модуля GPIO. Все регистры имеют разрядность, соответствующую количествам выводов блока.

Таблица 96 – Регистры модуля GPIO

Обозначение	Смещение	Начальное значение	Назначение	Доступ
SET / STATE	0x00	0x0	Текущее состояние выводов / Установка «1» При чтении – текущее состояние выводов. При записи бита «1» соответствующий биту вывод устанавливается в «1»	RW
CLEAR	0x04	0x0	Текущее состояние прерываний / Установка «0» При чтении – текущее состояние запросов прерываний. При записи бита «1» соответствующий биту вывод устанавливается в «0»	WO
DIRECTION_OUT	0x08	0xFFFF (0xFF)	Установка направления выводов как выход При чтении – текущее направление выводов: 0 – выход; 1 – вход. Запись «1» устанавливает соответствующий вывод как «выход»	RW
DIRECTION_IN	0x0C	0xFFFF (0xFF)	Установка направления выводов как вход При чтении – текущее направление выводов: 0 – выход; 1 – вход. Запись «1» устанавливает соответствующий вывод как «вход»	RW

Обозначение	Смещение	Начальное значение	Назначение	Доступ
OUTPUT	0x10	0x0	Выходной регистр Чтение возвращает содержимое выходного регистра независимо от текущего направления выводов Запись устанавливает значения всех битов выходного регистра	RW
CONTROL	0x14	0x0	Тестовый режим Запись «1» в [0] разряд – включение тестового режима (выходы замкнуты на входы внутри модуля) Запись «0» в [0] разряд – нормальный режим	RW

3.14.1 Многофункциональные выводы (PAD CONFIG)

Входы-выходы GPIO замультиплексированы с функциональными выводами. Назначение и схема мультиплексирования представлена в таблице выводов. Для управления функциями используются регистры PAD0_CFG, PAD1_CFG, PAD2_CFG. Ячейка ввода-вывода позволяет регулировать нагрузочную способность вывода (регистры PAD0_DS, PAD1_DS, PAD2_DS) и подключать резисторы подтяжки к питанию/земле (регистры PAD0_PUPD, PAD1_PUPD, PAD2_PUPD). Назначение этих регистров и отдельных полей представлено в таблицах 97 – 100.

Таблица 97 – Регистры управления функцией выводов.

Обозначение	Смещение	Начальное Значение	Назначение	Доступ
PAD0_CFG	0x00	0x55400000	Управление функциями выводов PORT0	RW
PAD0_DS	0x04	0x00000000	Управление нагрузочной способностью выводов PORT0	RW
PAD0_PUPD	0x08	0x00000000	Управление резисторами подтяжки выводов PORT0	RW
PAD1_CFG	0x0C	0x00000000	Управление функциями выводов PORT1	RW

Обозначение	Смещение	Начальное Значение	Назначение	Доступ
PAD1_DS	0x10	0x00000000	Управление нагрузочной способностью выводов PORT1	RW
PAD1_PUPD	0x14	0x00000000	Управление резисторами подтяжки выводов PORT1	RW
PAD2_CFG	0x18	0x00000000	Управление функциями выводов PORT2	RW
PAD2_DS	0x1C	0x00000000	Управление нагрузочной способностью выводов PORT2	RW
PAD2_PUPD	0x20	0x00000000	Управление резисторами подтяжки выводов PORT2	RW
Примечание – Смещение указано относительно базового адреса модуля 0x00050C00				

Таблица 98 – Поля регистров PADx_CFG управления функцией выводов

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Port_x_15	31:30	0x0	Значения двух бит кодируют выбранный функционал для вывода: 0b00 – первая функция (порт общего назначения) 0b01 – вторая функция (последовательный интерфейс) 0b10 – третья функция (последовательный интерфейс или таймер) 0b11 – четвертая функция (аналоговый сигнал)	RW
Port_x_14	29:28	0x0		RW
Port_x_13	27:26	0x0		RW
Port_x_12	25:24	0x0		RW
Port_x_11	23:22	0x0		RW
Port_x_10	21:20	0x0		RW
Port_x_9	19:18	0x0		RW
Port_x_8	17:16	0x0		RW
Port_x_7	15:14	0x0		RW
Port_x_6	13:12	0x0		RW
Port_x_5	11:10	0x0		RW
Port_x_4	9:8	0x0		RW
Port_x_3	7:6	0x0		RW
Port_x_2	5:4	0x0		RW
Port_x_1	3:2	0x0		RW
Port_x_0	1:0	0x0		RW

Таблица 99 – Поля регистров PADx_DS управления нагрузочной способностью выводов

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Port_x_15	31:30	0x0	Значения двух бит кодируют выбранный режим для вывода: 0b00 – 2mA, 0b01 – 4mA, 0b10, 0b11 – 8mA	RW
Port_x_14	29:28	0x0		RW
Port_x_13	27:26	0x0		RW
Port_x_12	25:24	0x0		RW
Port_x_11	23:22	0x0		RW
Port_x_10	21:20	0x0		RW
Port_x_9	19:18	0x0		RW
Port_x_8	17:16	0x0		RW
Port_x_7	15:14	0x0		RW
Port_x_6	13:12	0x0		RW
Port_x_5	11:10	0x0		RW
Port_x_4	9:8	0x0		RW
Port_x_3	7:6	0x0		RW
Port_x_2	5:4	0x0		RW
Port_x_1	3:2	0x0		RW
Port_x_0	1:0	0x0		RW

Таблица 100 – Поля регистров PADx_PUPD управления подтягивающими резисторами

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Port_x_15	31:30	0x0	Значения двух бит кодируют выбранный режим для вывода 0b00 – регистры подтяжки не подключаются, 0b01 – подключен регистр подтяжки 50 кОм к питанию, 0b10 – подключен регистр подтяжки 50 кОм к земле, 0b11 – зарезервировано	RW
Port_x_14	29:28	0x0		RW
Port_x_13	27:26	0x0		RW
Port_x_12	25:24	0x0		RW
Port_x_11	23:22	0x0		RW
Port_x_10	21:20	0x0		RW
Port_x_9	19:18	0x0		RW
Port_x_8	17:16	0x0		RW
Port_x_7	15:14	0x0		RW
Port_x_6	13:12	0x0		RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Port_x_5	11:10	0x0		RW
Port_x_4	9:8	0x0		RW
Port_x_3	7:6	0x0		RW
Port_x_2	5:4	0x0		RW
Port_x_1	3:2	0x0		RW
Port_x_0	1:0	0x0		RW

3.14.2 Модуль прерываний портов ввода-вывода (GPIO IRQ)

Обработка прерываний выводов реализована в отдельном модуле. Входные линии трех блоков ввода-вывода подключаются к 8 мультиплексорам формируя 8 линий, для которых формируются прерывания. Каждый мультиплексор управляется независимо, на основе соответствующих бит поля LINE_MUX.

Схема коммутации представлена в таблице 101.

Таблица 101 – Коммутация линий прерываний.

Прерывание	Линия_0 Mux[3:0]	Линия_1 Mux[7:4]	Линия_2 Mux[11:8]	Линия_3 Mux[15:12]	Линия_4 Mux[19:16]	Линия_5 Mux[23:20]	Линия_6 Mux[27:24]	Линия_7 Mux [31:28]
MUX [i]=0	GPIO0_0	GPIO0_1	GPIO0_2	GPIO0_3	GPIO0_4	GPIO0_5	GPIO0_6	GPIO0_7
MUX [i]=1	GPIO0_8	GPIO0_9	GPIO0_10	GPIO0_11	GPIO0_12	GPIO0_13	GPIO0_14	GPIO0_15
MUX [i]=2	GPIO1_0	GPIO1_1	GPIO1_2	GPIO1_3	GPIO1_4	GPIO1_5	GPIO1_6	GPIO1_7
MUX [i]=3	GPIO1_8	GPIO1_9	GPIO1_10	GPIO1_11	GPIO1_12	GPIO1_13	GPIO1_14	GPIO1_15
MUX [i]=4	GPIO2_0	GPIO2_1	GPIO2_2	GPIO2_3	GPIO2_4	GPIO2_5	GPIO2_6	GPIO2_7
MUX [i]=5	GPIO0_4	GPIO0_5	GPIO0_6	GPIO0_7	GPIO0_0	GPIO0_1	GPIO0_2	GPIO0_3
MUX [i]=6	GPIO0_12	GPIO0_13	GPIO0_14	GPIO0_15	GPIO0_8	GPIO0_9	GPIO0_10	GPIO0_11
MUX [i]=7	GPIO1_4	GPIO1_5	GPIO1_6	GPIO1_7	GPIO1_0	GPIO1_1	GPIO1_2	GPIO1_3
MUX [i]=8	GPIO1_12	GPIO1_13	GPIO1_14	GPIO1_15	GPIO1_8	GPIO1_9	GPIO1_10	GPIO1_11
MUX [i]=9	GPIO2_4	GPIO2_5	GPIO2_6	GPIO2_7	GPIO2_0	GPIO2_1	GPIO2_2	GPIO2_3
MUX [i]=10	0	0	0	0	0	0	0	0

Примечание - когда модуль прерываний портов ввода-вывода сконфигурирован для работы по уровню, выход прерывания не «зашелкивается». В этом случае следует конфигурировать внешний контроллер прерываний (EPIC) на работу по

фронту. Когда модуль прерываний портов ввода-вывода (GPIO IRQ) сконфигурирован для работы по событию (фронт/спад), выход прерывания «зашелкивается».

Описание регистров блока прерываний представлено в таблице 102.

Таблица 102 – Регистры модуля управления прерываниями GPIO

Обозначение	Смещение	Начальное значение	Назначение	Доступ
STATE	0x00	0x0	Текущее состояние выводов линий после мультиплексоров. Номер бита соответствует номеру канала.	RO
LINE_MUX	0x04	0x0	Управление мультиплексорами. Каждый мультиплексор управляетя своими 4 битами: Mux_0 – [3:0]; Mux_1 – [7:4]; Mux_2 – [11:8]; Mux_3 – [15:12]; Mux_4 – [19:16]; Mux_5 – [23:20]; Mux_6 – [27:24]; Mux_7 – [31:28]	RW
INTERRUPT	0x08	0x0	Текущее состояние прерываний. Номер бита соответствует номеру канала.	RW
ENABLE_SET	0x0C	0x0	Регистр разрешения прерываний При чтении – текущее состояние разрешений прерываний. При записи «1» разрешает прерывание от соответствующего канала.	RW
ENABLE_CLEAR	0x10	0x0	Регистр запрета прерываний При чтении – текущее состояние разрешений прерываний. При записи «1» запрещает прерывание от соответствующего канала	RW

Обозначение	Смещение	Начальное значение	Назначение	Доступ
EDGE	0x14	0x0	<p>Регистр типа прерываний.</p> <p>При чтении – текущий тип прерывания:</p> <p>«1» – по событию (фронт или спад), «0» – по уровню.</p> <p>Запись «1» – прерывание формируется по событию для соответствующего канала</p> <p>Запись «0» - не влияет</p>	RW
LEVEL	0x18	0xFF	<p>Регистр типа прерываний.</p> <p>При чтении – текущий тип прерывания (инвертированный):</p> <p>«0» – по событию;</p> <p>«1» – по уровню.</p> <p>Запись «1» – прерывание формируется по уровню для соответствующего канала</p> <p>Запись «0» - не влияет</p>	RW
LEVEL_SET	0x1C	0x0	<p>Регистр выбора уровня / события прерываний.</p> <p>При чтении:</p> <p>«0» – прерывания формируются по спаду или уровню логического «0»;</p> <p>«1» – прерывания формируются по нарастающему фронту или уровню логической «1»</p> <p>Запись «1» – прерывание формируется по нарастающему фронту или уровню логической «1» для соответствующего канала</p>	RW

Обозначение	Смещение	Начальное значение	Назначение	Доступ
LEVEL_CLEAR	0x20	0x0	Регистр уровня / события прерываний. При чтении: «0» – прерывания формируются по спаду или уровню логического «0»; «1» – прерывания формируются по нарастающему фронту или уровню логической «1» Запись «1» – прерывание формируется по спаду или уровню логического «0» для соответствующего канала	RW
ANY_EDGE_SET	0x24	0x0	Регистр прерываний по любому событию. Запись «1» – прерывание формируется по любому изменению соответствующего канала	RW
ANY_EDGE_CLEAR	0x28	0x0	Регистр прерываний по любому событию. Запись «1» – прерывание не формируется по любому изменению соответствующего канала	RW
CLEAR	0x2C	–	Регистр сброса флагов прерываний по событию. Запись «1» – очищает флаг прерывания соответствующего канала	W1C

3.15 Контроллер внешней памяти (SPIFI)

Контроллер SPIFI с КЭШ предназначен для организации взаимодействия микропроцессорного ядра с микросхемой внешней FLASH-памяти. Это позволяет обеспечить исполнение кода программы, записанного в микросхеме флеш-памяти, а также при необходимости чтение и запись произвольных данных во внешнюю флеш-память в процессе выполнения программы.

Контроллер SPIFI с КЭШ обеспечивает работу с микросхемами FLASH – памяти через SPI интерфейс в одном из трех режимов: одноканальный; двухканальный;

четырёхканальный. По умолчанию используется одноканальный режим, двухканальный и четырёхканальный режим включается через программное обеспечение.

Микросхема внешней FLASH – памяти, подключаемая к блоку SPI Flash с КЭШ, должна иметь встроенный интерфейс SPI (Single, Dual, Quatro) и поддерживать набор команд управления, определенных в стандарте JEDEC.

Настройка временных характеристик диаграмм команд чтения/записи данных из внешней флэш памяти осуществляется программным обеспечением через регистры управления контроллера SPIFI с КЭШ.

Встроенный аппаратный драйвер поддерживает команды определения типа Flash, запускаемые через регистровый файл, а также поддерживает опции выбора используемых команд для коммуникации (одиночные чтения или удержание в режиме чтения последовательности).

По умолчанию используются наиболее распространённые временные диаграммы без оптимизации быстродействия для обеспечения возможности исполнения кода программы непосредственно из микросхемы флэш памяти.

3.15.1 Контроллер интерфейса флэш-памяти SPIFI

Контроллер интерфейса флэш-памяти SPIFI – это цифровой автомат, предназначенный для информационного сопряжения процессора системы на кристалле (CPU SOC) с микросхемами флэш-памяти (МФП) ныне существующих и перспективных типов. Данный контроллер избавляет процессор от выполнения рутинных операций при работе с флэш-памятью (при выполнении операций чтения, записи и стирания памяти).

В начале работы процессор настраивает контроллер SPIFI, а в процессе работы процессор обменивается информацией с флэш-памятью по параллельной скоростной синхронной шине АНВ.

Контроллер может работать в двух режимах:

- режим работы с периферией;
- режим работы с памятью.

Контроллер SPIFI также имеет режим загрузки «boot mode» (разновидность режима работы с памятью), в котором начальная загрузка SOC может быть осуществлена из SPI флэш-памяти. В этом случае отпадает необходимость грузиться с параллельной загрузочной памяти с большим количеством выводов (что плохо для габаритов и надежности).

Контроллер SPIFI может работать с флэш-памятью как в обычном последовательном режиме, когда данные передаются/принимаются по двум односторонним линиям (MISO/MOSI), так и в параллельных режимах (двух и четырехбитном), когда данные передаются/принимаются по двум или четырем двухнаправленным линиям.

Структурная схема контроллера представлена на рисунке 44.

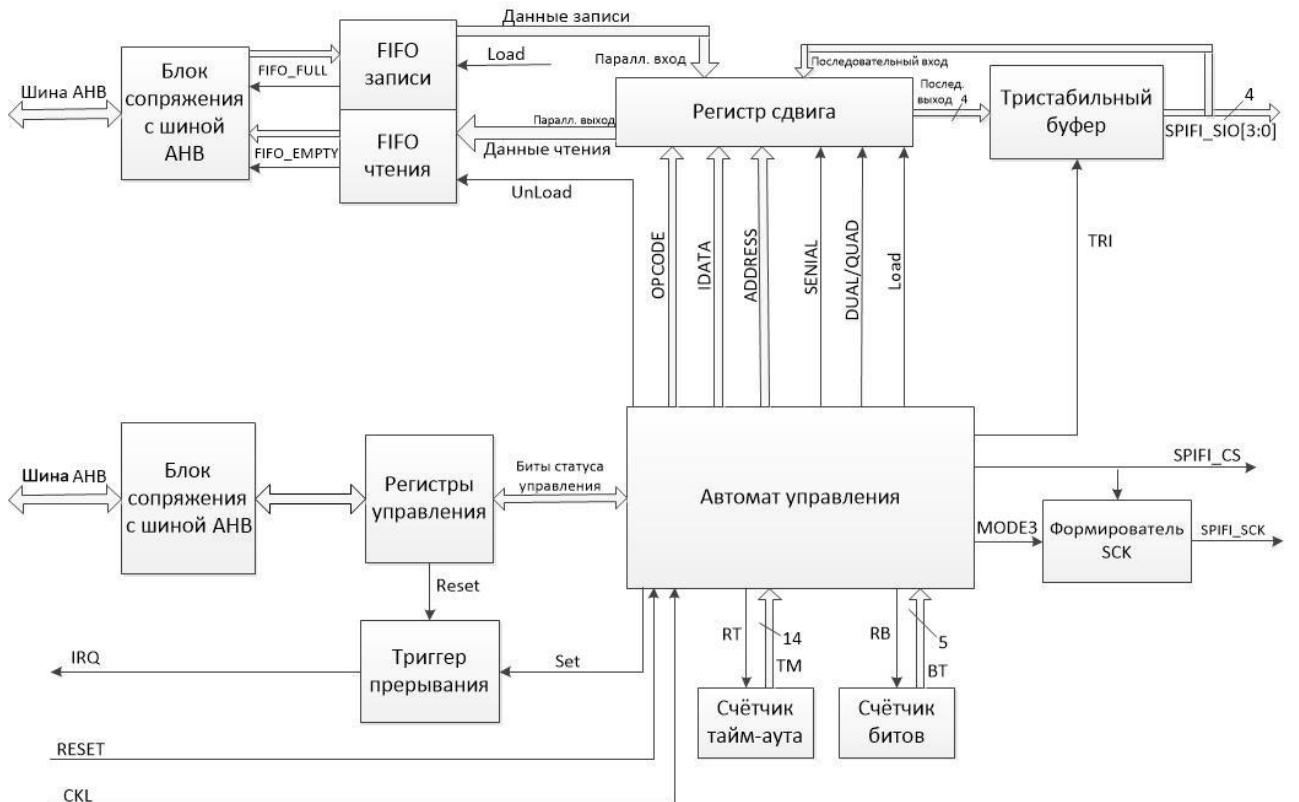


Рисунок 66 – Структурная схема контроллера флэш-памяти SPIFI

Контроллер интерфейса флэш-памяти SPIFI состоит из следующих компонентов:

- блоков сопряжения с шиной АНВ;
- FIFO буферов;
- комплекта регистров управления;
- триггера запроса прерывания;
- регистра сдвига;
- автомата управления;
- триистабильного буфера;
- формирователя тактового сигнала SCK;
- счетчиков (тайм-аута и сдвига).

Блок сопряжения с шиной АНВ преобразует сигналы этой шины в сигналы чтения и записи регистров управления, кроме регистра данных. Блок регистров управления

формирует все сигналы управления и статуса контроллера. Сигналы управления приходят на автомат управления, который управляет процессом обмена информацией с флэш-памятью. Автомат также управляет формированием тактового сигнала SPIFI_SCK через соответствующий формирователь. Процесс сдвига данных управляется автоматом с помощью счетчика сдвигов. Регистр сдвига сдвигает записанную автоматом информацию в флэш-память и обратно. Данные поступают во флэш-память через тристабильный буфер, который необходим для выключения выходов сдвигового регистра во время приема информации из флэш-памяти при работе контроллера в режиме dual/quad. Параллельное чтение и запись данных в сдвиговый регистр и из него осуществляется через FIFO буфер, который в свою очередь получает информацию от блока сопряжения с АНВ шиной. В адресном пространстве FIFO буфера данные находятся по адресу регистра данных DATA.

3.15.2 Регистры контроллера SPIFI

Карта регистров контроллера SPIFI приведено в таблице 88. Обмен информацией между процессором и регистрами контроля осуществляется через шину АНВ.

Таблица 103 – Перечень регистров контроллера SPIFI

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CTRL	0x000	0x400FFFFF	SPIFI регистр управления	RW
CMD	0x004	0x00000000	SPIFI регистр команд	RW
ADDR	0x008	0x00000000	SPIFI регистр адреса	RW
IDATA	0x00C	0x00000000	SPIFI регистр промежуточных данных	RW
CLIMIT	0x010	0x00000000	SPIFI регистр верхней границы адреса кеширования	RW
DATA	0x014	0x00000000	SPIFI регистр данных	RW
MCMD	0x018	0x03800000	SPIFI регистр команд памяти	RW
STAT	0x01C	0x02000001	SPIFI регистр статуса	RW

3.15.2.1 Регистр управления CTRL

Данный регистр управляет выполнением всех операций контроллера и должен быть настроен перед инициализацией любой команды. Поля данного регистра приведены в таблице 104.

Таблица 104 – CTRL регистр управления

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DMAEN	31	0x0	Бит разрешения запросов DMA от контроллера SPIFI 1 – запросы разрешены. Устанавливать только в периферийном режиме	RW
FBCLK	30	0x1	Бит выбора сигнала стробирования входных данных с выходного буфера тактового сигнала: 0 – внутренний сигнал, 1 – внешний сигнал	RW
RFCLK	29	0x0	Бит выбора активного перепада сигнала для стробирования входных данных: 0 – положительный; 1 – отрицательный	RW
DUAL	28	0x0	Бит выбора протокола: сигналы IO[3:0] – (0–4) битовый протокол; сигналы IO[1:0] – (1–2) битовый протокол	RW
PRFTCH_DIS	27	0x0	Бит разрешения упреждающих выборок кэш памяти 0 – выборки разрешены	RW
SCK_DIV	26:24	0x0	Делитель для получения тактового сигнала SPIFI_SCK из системного тактового сигнала HCLK. Частота рассчитывается по формуле: $F_{SPIFI_SCK} = F_{HCLK} / 2^{(SCK_DIV+1)}$	RW
MODE3	23	0x0	Бит режима 3	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
INTEN	22	0x0	Бит разрешения прерывания при завершении выполнения команды (если этот бит равен «1», то прерывание разрешено)	RW
D_CACHE_DIS	21	0x0	Бит запрещения кэширования данных (транзакций АНВ, для которых сигнал HPROT[0] = 1): 0 – при разрешенном кэшировании кэширование данных выполняется; 1 – данные не кэшируются	RW
CACHE_EN	20	0x0	Бит разрешения кэширования	
CSHIGH	19:16	0xF	Количество периодов сигнала SPIFI_SCK минус один, в течение которых сигнал SPIFI_CS остается в неактивном состоянии перед началом выполнения команды	RW
Зарезервировано	15:0		Зарезервировано	

3.15.2.2 Регистр команд CMD

Регистр команд CMD определяет последовательность действий контроллера при выполнении команд периферийного режима. Этот регистр может быть записан только тогда, когда биты CMD и MCINIT равны нулю. Поля данного регистра приведены в таблице 105.

Таблица 105 – CMD регистр команд

Обозначение	Разряды	Начальное значение	Назначение	Доступ
OPCODE	31:24	0x0	Код операции (не используется для некоторых значений поля FRAMEFORM)	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
FRAMEFORM	23:21	0x0	Бит управления полями кода операции и адреса команды: 0 – резерв; 1 – выдается только код операции, адреса нет; 2 – код операции и младший байт адреса; 3 – код операции и два младших байта адреса; 4 – код операции и три младших байта адреса; 5 – код операции и 4 байта адреса; 6 – нет кода операции, три младших байта адреса; 7 – нет кода операции, 4 байта адреса	RW
FIELDFORM	20:19	0x0	Формат вывода полей команды: 0 – все поля выводятся в последовательном режиме; 1 – данные выводятся в четырех или двух битовом режиме, а остальные поля в последовательном режиме; 2 – код операции выводится в последовательном режиме, а остальные в четырех или двух битовом; 3 – все поля в четырех или двух битовом режиме	RW
INTLEN	18:16	0x0	Количество байт промежуточных данных, которые хранятся в регистре IDATA. Если количество байт больше, чем может хранить регистр IDATA, то остаток добавляется нулевыми значениями	RW
DOUT	15	0x0	Бит направления передачи данных 0 – чтение из флэш	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
POLL	14	0x0	<p>Бит должен быть установлен при выполнении команды, которая содержит входное поле данных и циклически запрашивает состояние бита входного потока битов из регистра статуса флэш-памяти. Номер проверяемого бита в байте статуса должен быть указан в битах DATALEN[2:0], а требуемое значение бита – в бите DATALEN[3]. Как только значения этих битов станут равны, контроллер завершает выполнение текущей команды, деактивируя сигнал SPIFI_CS, и формирует прерывание, если оно разрешено.</p> <p>После завершения процесса поллинга необходимо считать один байт из регистра DATA, так как в нем сохраняется последний принятый статус FLASH.</p>	RW
DATALEN	13:0	0x0	В тех случаях, когда бит POLL равен «0», это поле определяет количество байт данных при выполнении команды. Если это поле равно «0», то выполняемая команда не содержит данных	RW

3.15.2.3 Регистр адреса ADDRESS

Данный регистр должен быть настроен перед инициализацией любой команды. При выполнении команды первым выводится старший бит адреса. Поле данного регистра приведено в таблице 106.

Таблица 106 – ADDRESS регистр адреса

Обозначение	Разряды	Начальное значение	Назначение	Доступ
ADDRESS	31:0	0x0	Исполнительный адрес команды	RW

3.15.2.4 Регистр промежуточных данных IDATA

Данный регистр должен быть настроен перед инициализацией команды, в которой используются промежуточные данные. При выполнении команды первым выводится младший бит слова. Поле данного регистра приведено в таблице 107.

Таблица 107 – Регистр промежуточных данных

Обозначение	Разряды	Начальное значение	Назначение	Доступ
IDATA	31:0	0x0	Промежуточные данные	RW

3.15.2.5 Регистр верхней границы адреса кэширования CLIMIT

Контроллер SPIFI имеет возможность кэширования полученных из флэш-памяти данных в целях увеличения производительности. В регистре CLIMIT хранится значение верхней границы кэшируемой памяти. Ячейки памяти с адресами большими, чем значение CLIMIT, не являются кэшируемыми. Поле данного регистра приведено в таблице 108.

Таблица 108 – Регистр верхней границы адреса кэширования

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CLIMIT	31:0	0x0	Верхний предел кэшируемой памяти	RW

Примечание – Этот регистр может быть изменен, когда биты CMD и MCINIT в регистре статуса STAT равны нулю.

3.15.2.6 Регистр данных DATA

По адресу данного регистра подключены FIFO буфера данных, которые, с другой стороны, подключены к шине АНВ. Поэтому при обращении по шине АНВ к данному регистру данные читаются из или пишутся в соответствующий FIFO буфер. Если выходной буфер чтения пуст или входной буфер записи полон, то при отправке запроса по шине АНВ будет вызвано исключение (код 5 «Load access fault»). Всего в регистр DATA записывается или читается из него количество байт, указанное в поле DATALEN регистра CMD. Если включен режим поллинга (бит POLL регистра CMD активен), то данные в выходной буфер не записываются, а первый байт входного буфера используется для хранения статуса FLASH. После завершения поллинга необходимо считать один байт из регистра DATA. Поле данного регистра приведено в таблице 109.

Таблица 109 – Регистр данных DATA

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DATA	31:0	0x0	Входные или выходные данные	RW

3.15.2.7 Регистр команд памяти MCMD

Регистр определяет последовательность действий контроллера при выполнении команд режима работы с памятью. Этот регистр может быть записан только тогда, когда биты CMD и MCINIT равны нулю. Поля данного регистра приведены в таблице 110.

Таблица 110 – Регистр команд памяти MCMD

Обозначение	Разряды	Начальное значение	Назначение	Доступ
OPCODE	31:24	0x3	Код операции (не используется для некоторых значений поля FRAMEFORM)	RW
FRAMEFORM	23:21	0x4	Бит управления полями кода операции и адреса команды: 0 – резерв; 1 – резерв; 2 – код операции + младший байт адреса; 3 – код операции + два младших байта адреса; 4 – код операции + три младших байта адреса; 5 – код операции + 4 байта адреса; 6 – нет кода операции, три младших байта адреса; 7 – нет кода операции, 4 байта адреса	RW
FIELDFORM	20:19	0x0	Формат вывода полей команды: 0 – все поля выводятся в последовательном режиме; 1 – данные выводятся в четырех или двух битовом режиме, остальные поля – в последовательном; 2 – код операции выводится в последовательном режиме, остальные – в четырех или двух битовом; 3 – все поля в четырех или двух битовом режиме	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
INTLEN	18:16	0x0	Кол-во байт промежуточных данных (они хранятся в регистре IDATA). Если кол-во байт больше, чем может хранить регистр IDATA, остаток заполняется нулевыми значениями	RW
DOUT	15	0x0	Бит должен быть установлен в «0»	RW
POLL	14	0x0	Бит должен быть установлен в «0»	RW
Зарезервировано	13:0	Зарезервировано		

3.15.2.8 Регистр STAT

Данный регистр содержит текущее состояние контроллера. Поля данного регистра приведены в таблице.

Таблица 111 – Регистр статуса STAT

Обозначение	Разряды	Начальное значение	Назначение	Доступ
VERSION	31:24	0x02	Версия контроллера SPIFI (поле доступно только для чтения)	RW
Зарезервировано	23:6	Зарезервировано		
INTRQ	5	0x0	Запись «1» в бит сбрасывает запрос на прерывание от контроллера SPIFI	RW
RESET	4	0x0	Бит предназначен для того, чтобы прервать текущую команду периферийного режима или режима памяти. Бит сбрасывается, когда контроллер готов к выполнению новой команды	RW
Зарезервировано	3:2	Зарезервировано		
CMD	1	0x0	Бит устанавливается в «1», если программное обеспечение успешно записало регистр команд. Сброс этого бита осуществляется теми же сигналами, что и бит MCINIT. Также сброс бита CMD происходит при завершении выполнения команды, когда деактивируется сигнал SPIFI_CS	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MCINIT	0	0x1	Бит устанавливается в «1», если программное обеспечение успешно записало регистр команд памяти. Сброс бита осуществляется аппаратным сбросом или установкой в «1» бита RESET	RW

3.15.3 Структура и принцип работы контроллера SPIFI с КЭШ

SPI Flash контроллера с КЭШ представляет собой систему взаимосвязанных блоков: контроллер шины AHB; контроллер шины AHB_M; SPIFI кэш контроллер; КЭШ инструкций страничного типа, ассоциативная память, контроллер последовательного интерфейса SPIFI (включающий JEDEC Flash Driver и SPIM).

Структура SPI Flash контроллера с КЭШ представлена на рисунке 42.



Рисунок 67 – Общая структура контроллера

В зависимости от состояния бита CACHE_EN блок контроллера SPIFI с КЭШ работает в одном из режимов:

- CACHE_EN = 0, обход КЭШ (режим работы по умолчанию);
- CACHE_EN = 1, режим кеширования включен.

По шине AHB_M возможны только транзакции чтения. Шинный контроллер AHB_M осуществляет анализ транзакций нашине AHB_M. При обращении на чтение в адресное пространство флэш-памяти шинный контроллер проверяет наличие данных в КЭШ. При «промахе» КЭШ контроллер КЭШ формирует соответствующий внутренний запрос на чтение данных через контроллер SPIFI.

Кэшированию подлежат инструкции и данные, значение которых не меняется.

3.15.3.1 Принцип работы блока SPIFI КЭШ контроллер

Перед началом работы с блоком КЭШ производится следующие настройки:

- настраивается верхняя граница кэшируемой области установкой соответствующих значений CLIMIT (CLIMIT 32 разрядная шина);
- настраивается контроллер последовательного интерфейса на операции одиночного чтения;
- устанавливается режим кеширования CACHE_EN;
- осуществляется переход на адрес кода программы для чтения первой инструкции.

Далее работа происходит следующим образом, при обращении мастера АНВ шины КЭШ контроллер производит анализ транзакции (анализируются только транзакции чтения). КЭШ контроллер по значению HADDR[31:7] (HADDR - адрес транзакции шины АНВ) и содержимому полей ТЭГ ассоциативной памяти, а также регистров запыленности, определяет «попадание» или «промах» КЭШ.

При «попадании» из блока страничной КЭШ памяти на шину АНВ выдаётся требуемая инструкция. Требуемая инструкция (страница и слово) вычисляется на основании HADDR[6:2].

При «промахе» КЭШ контроллер опускает сигнал HREADY_O и формирует запрос в контроллер SPIFI, на вычитывание запрошенных данных. При получении данных КЭШ контроллер выдает их на шину, поднимает сигнал HREADY_O и помечает соответствующий бит запыленности. При отсутствии обращений к блоку возможна теневая подгрузка всей страницы данных.

Теневая подгрузка основывается на возможности prefetch, реализованной в SPIFI контроллере. При установленном сигнале PREFETCH_DIS=0, SPIFI контроллер автоматически вычитывает следующее слово из флэш-памяти и готов выдать его. Если обращений по шине AHB_M, вызывающих изменение адреса, к КЭШ не будет и PREFETCH_DIS=0, КЭШ контроллер в свободное время будет заполнять страницу до конца.

Запись во внешнюю флэш – память доступна по шине АНВ, минуя блок КЭШ памяти. Для записи данных SPIFI контроллер необходимо перевести в режим записи. Нужно учитывать, что операция записи во внешнюю флэш требует больших промежутков времени, в течении которого вычитывание инструкций из внешней флэш – памяти будет недоступно.

В КЭШ контроллере реализована возможность включения/выключения кэширования константных значений данных. Для включения кэширования данных необходимо установить сигнал D_CACHE_DIS=0.

3.15.3.2 Блок КЭШ и ассоциативная память

Блок КЭШ предназначен выдачи ранее загруженных инструкций и данных на АНВ шину в процессе исполнения микропроцессорным ядром программы из внешней флэш памяти. Фактически блок КЭШ имитирует работу ОЗУ инструкций со временем доступа к инструкциям за 1 такт HCLK шины АМВА АНВ.

Блок КЭШ представляет собой блок памяти из 8 страниц КЭШ инструкций по 128 байт каждая. Структура КЭШ представлена на рисунке 68. В данной структуре любая страница флэш памяти может отображаться в любой блок КЭШ памяти. При полном заполнении КЭШ вытесняться будет блок, к которому дольше всех не обращались.

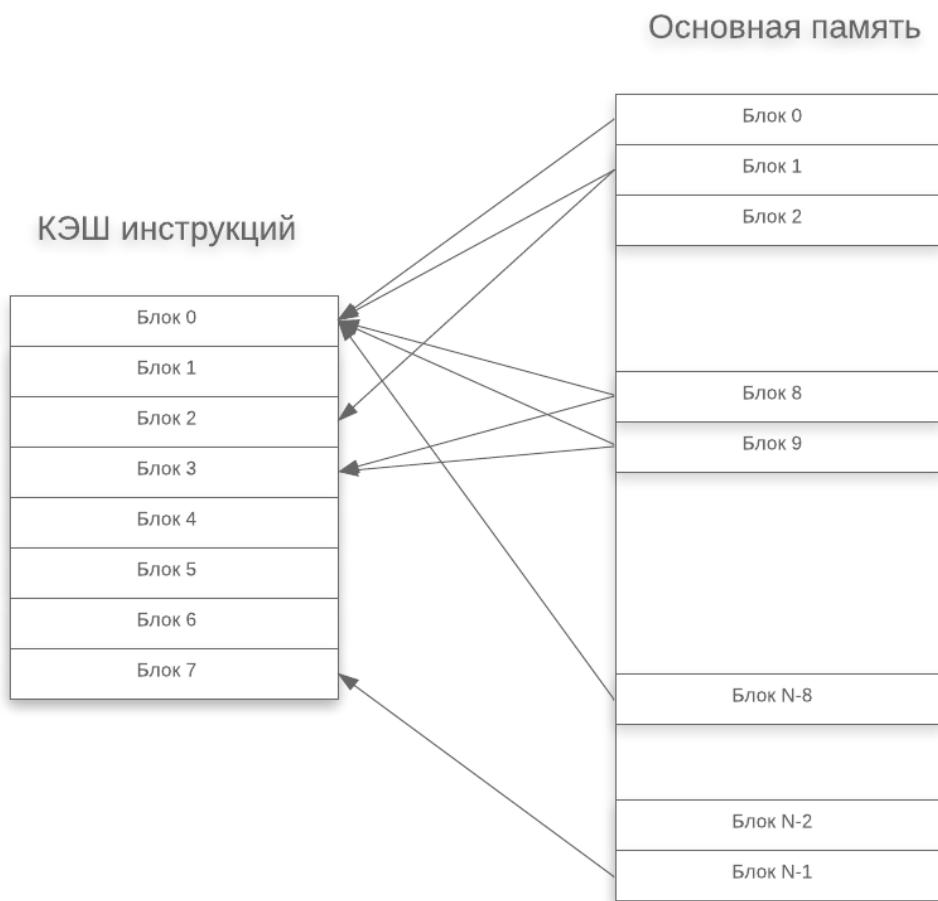


Рисунок 68 – Структура отображения блока КЭШ

КЭШ использует блок памяти 256x32 бита, загружаемый словами (выравнивание по границе слова всегда).

Ассоциативная память построена на триггерах (зашелках) и имеет размер 8x25 бит.

В ассоциативной памяти хранится ТЭГ отображеного блока, а также регистры запыленности. При анализе запроса на чтение, адрес чтения сравнивается с ТЭГами блоков, записанных в КЭШ, и определяет «попадание» или «промах». При совпадении запрошенного адреса с ТЭГ определяется «попадание» в страницу КЭШ и через бит запыленности проверяется актуальность данных и происходит чтение.

Алгоритм учета запыленности строится следующим образом. Каждому блоку кэш памяти соответствует регистр запыленности с размером 8 бит. Каждым 4 словам в блоке КЭШ соответствует один бит в регистрах запыленности (128 байт = 32 слова в блоке). При сбросе системы все биты устанавливаются в «0», что означает отсутствие данных в КЭШ. При загрузке данных из внешней флэш, данные попадают в КЭШ и, соответствующий бит устанавливается в «1», что соответствует наличию актуальных данных (данные записываются по 4 слова последовательно, что приводит к возникновению задержки). При замещении страницы, одновременно с записью нового ТЭГ, регистр запыленности данного блока сбрасывается в «0».

Структура проверки адресов и вычитывания информации представлена в таблице 111.

Таблица 112 – Структура ассоциативной памяти КЭШ

БЛОК 0 ТЭГ 0 = HADDR[31:7]	Бит запыленности 0 (HADDR[6:4] = 0)	СЛОВО 0 (HADDR[3:2] = 0)
		СЛОВО 1 (HADDR[3:2] = 1)
		СЛОВО 2 (HADDR[3:2] = 2)
		СЛОВО 3 (HADDR[3:2] = 3)
	Бит запыленности 1 (HADDR[6:4] = 1)	СЛОВО 4 ... СЛОВО 7
	...	
	Бит запыленности 7 (HADDR[6:4] = 7)	СЛОВО 28 ... СЛОВО 31
	БЛОК 1	
...		
БЛОК 7		

КЭШ имеет одноканальную полностью ассоциативную структуру отображения на 8 страниц по 128 байт каждая с замещением блоков по принципу PLRU (pseudo least

recently used). Алгоритм работает следующим образом: рассмотрим бинарное дерево поиска. Каждый узел дерева имеет однобитный флаг, обозначающий «перейти влево, чтобы найти псевдо-LRU элемент» или «перейти вправо, чтобы найти псевдо-LRU элемент». Элемент псевдо-LRU будет найден, если пройти по дереву в соответствии со значениями флагов. При доступе к элементу, произойдет обновление дерева к этому элементу, и во время обхода устанавливаются флаги узлов для обозначения направления, противоположного выбранному направлению.

Этот алгоритм может быть неоптимальным, так как он является аппроксимацией. Для алгоритма с 8 модулями КЭШ памяти принцип изменения бит LRU бит будет следующим (рисунок 44):

- если последнее обращение к блоку L0, L1, L2 или L3, то бит B6 устанавливается в состояние 1, а при обращении к блоку L4, L5, L6 или L7 бит B6 сбрасывается в 0;
- если последнее обращение к блоку L0 или L1, то бит B4 устанавливается в состояние 1, а при обращении к блоку L2 или L3 бит B4 сбрасывается в 0;
- если последнее обращение к блоку L4 или L5, то бит B5 устанавливается в состояние 1, а при обращении к блоку L4 или L5 бит B5 сбрасывается в 0;
- если последнее обращение к блоку L0, то бит B0 устанавливается в состояние 1, а при обращении к блоку L1 бит B0 сбрасывается в 0;
- если последнее обращение к блоку L2, то бит B1 устанавливается в состояние 1, а при обращении к блоку L3 бит B1 сбрасывается в 0;
- если последнее обращение к блоку L4, то бит B2 устанавливается в состояние 1, а при обращении к блоку L5 бит B2 сбрасывается в 0;
- если последнее обращение к блоку L6, то бит B3 устанавливается в состояние 1, а при обращении к блоку L7 бит B3 сбрасывается в 0.

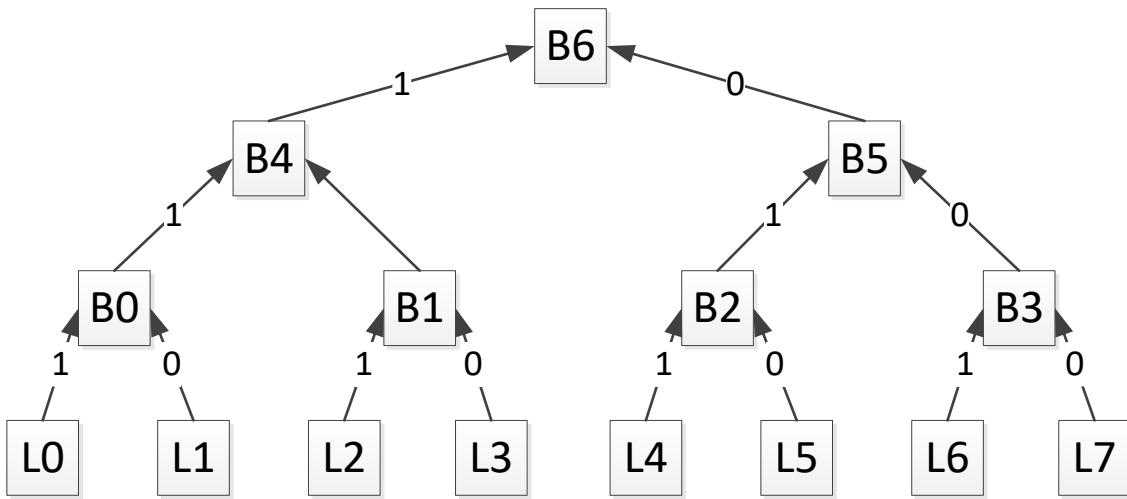


Рисунок 69 – Диаграмма работы алгоритма PLRU

Тогда таблица замещения будет выглядеть, как представлено в таблице 112.

Таблица 113 – Замещения для варианта 8 блоков

B6	B5	B4	B3	B2	B1	B0	Замена блока
0	X	0	X	X	X	0	L0
0	X	0	X	X	X	1	L1
0	X	1	X	X	0	X	L2
0	X	1	X	X	1	X	L3
1	0	X	X	0	X	X	L4
1	0	X	X	1	X	X	L5
1	1	X	0	X	X	X	L6
1	1	X	1	X	X	X	L7

3.15.4 Работа контроллера SPIFI интерфейса флэш-памяти

3.15.4.1 Начальная инициализация

Начальная инициализация производится следующим образом:

- необходимо установить в активное состояние сигнала RESET не менее чем на один такт сигнала CLK;
- в ответ на это блок сопряжения с шиной АHB установится в исходное состояние, в котором все сигналы запросов будут деактивированы;
- указатели адресов внутри FIFO примут нулевые значения, сигнал FIFO_FULL – «0», сигнал FIFO_EMPTY – «1»;

- блок сопряжения с шиной АНВ на сброс не реагирует, однако все регистры управления примут значения по умолчанию таблица 2 «Начальное значение»;
- триггер запроса прерывания сбрасывается;
- автомат управления устанавливается в начальное состояние IDLE, в котором триистабильный буфер находится в третьем состоянии. Сигналы Set, Load, UnLoad и SPIFI_CS неактивны (SPIFI_CS – «1», остальные – «0»), сигналы RT и RB активны – «1», удерживается в нуле счетчик таймаута и счетчик битов;
- формирователь тактового сигнала SCK удерживает сигнал SPIFI_SCK в нуле, поскольку сигнал SPIFI_CS – «1», а бит MODE3 равен «0»;
- регистр сдвига сбрасывается в нулевое состояние (для определенности).
- Вывод контроллера из режима работы с памятью осуществляется установкой бита RESET регистра статуса STAT

3.15.4.2 Подготовка контроллера SPIFI к выполнению команды

Общение контроллера SPIFI с МФП сводится к тому, что контроллер записывает в управляющий автомат МФП команды, а автомат МФП выполняет эти команды. Команда в общем случае состоит из кода операции, адреса внутри адресного пространства МФП, промежуточных и основных данных. Поэтому перед отправкой команды в флэш-память необходимо записать необходимое количество байт данных в регистр (буфер FIFO) (если посылаем команду записи), необходимое количество байт данных в регистр промежуточных данных IDATA, необходимое количество байт адреса в регистр ADDRESS. Далее необходимо настроить регистры управления CTRL и регистр команд CMD (если выполняется команда периферийного режима) или MCMD (если выполняется команда режима с памятью).

При этом следует помнить, что:

- не все команды используют код операции (поля FRAMEFORM регистра CMD или MCMD);
- не все команды используют адрес (поля FRAMEFORM регистра CMD или MCMD);
- не все команды используют промежуточные данные (поля INTLEN регистра CMD или MCMD). Если посылаемая команда не имеет в своем составе промежуточных данных, это поле должно быть нулевым;

- только команды записи (периферийный режим) используют данные DATA. Если команда не имеет в своем составе данных, то поле DATALEN регистра CMD должно быть нулевым.
- При заполнении полей регистра управления CTRL нужно детально представлять динамические характеристики и протокольные особенности той МФП, с которой в данный момент взаимодействует центральный процессор.

Значения полей:

- CSHIGH – определяется значением времени неактивного состояния сигнала CS между соседними обращениями к МФП;
- RFCLK и FBCLK – определяются частотными свойствами МФП и временными задержками сигналов, путешествующих от процессора к МФП и обратно (определяется моделированием PCB – файла с использованием IBIS модели МФП);
- MODE3 – определяется способностью МФП работать в режиме MODE1 или MODE3;
- DUAL, DMAEN и INTEN – определяются протокольными особенностями МФП (двух или четырех битный протокол, используется DMA или нет, используется прерывание по завершении команды или нет).

Помимо этого, значение поля FIELDFORM также определяется протокольными особенностями, регистр сдвига может работать в режимах сдвига по одному, по двум или по четырем битам. При сдвиге данных в сдвиговом регистре каждое слово будет сдвигаться N раз, согласно формуле (11):

$$N = Nb / R \quad (11)$$

где Nb – количество сдвигаемых бит (8, 16, 24 или 32);

R – разрядность сдвига.

Очевидно, что число N будет находиться в пределах от 2 (сдвиг восьми бит с разрядностью сдвига четыре) до 32 (сдвиг 32 бит с разрядностью сдвига один). В процессе сдвига данных счетчик сдвигов считает количество сдвигов и, как только это количество станет равно N-1, то автомат управления со следующего такта переходит к выводу следующего слова.

Значения полей остальных регистров управления зависят от режима работы (периферийный или режим работы с памятью), типов команды (чтение, запись или стирание) и будет приведено ниже отдельно для каждого случая.

3.15.4.3 Алгоритм работы автомата контроллера в периферийном режиме

Алгоритм работы автомата контроллера в периферийном режиме приведен на рисунке 70.

Автомат ждет сигнала начала работы CMD (бит 1 регистра статуса). Как только этот бит устанавливается в «1», что происходит в случае успешной записи в регистр команд CMD, автомат записывает в регистр сдвига код операции (если это предусмотрено полем FRAMFORM), затем адрес (если FRAMFORM это предусматривает), затем промежуточные данные IDATA (если поле INTLEN не равно 0), а затем выполняется один из трех алгоритмов: алгоритм поллинга, чтения или записи. В случае если бит POLL равен «1», то автомат переходит в состояние непрерывного чтения байта статуса из регистра МФП. Как только бит, номер которого записан в биты DATALEN[2:0], станет равен значению бита DATALEN[3], то автомат поймет, что условие выполнено и закончит выполнение процедуры поллинга. Если бит POLL равен «0», то в зависимости от значения бита DOUT автомат либо запишет в МФП из буфера FIFO, либо считает из МФП в буфер OFIFO байты данных в количестве, равном значению DATALEN[13:0], после чего автомат закончит свою работу.

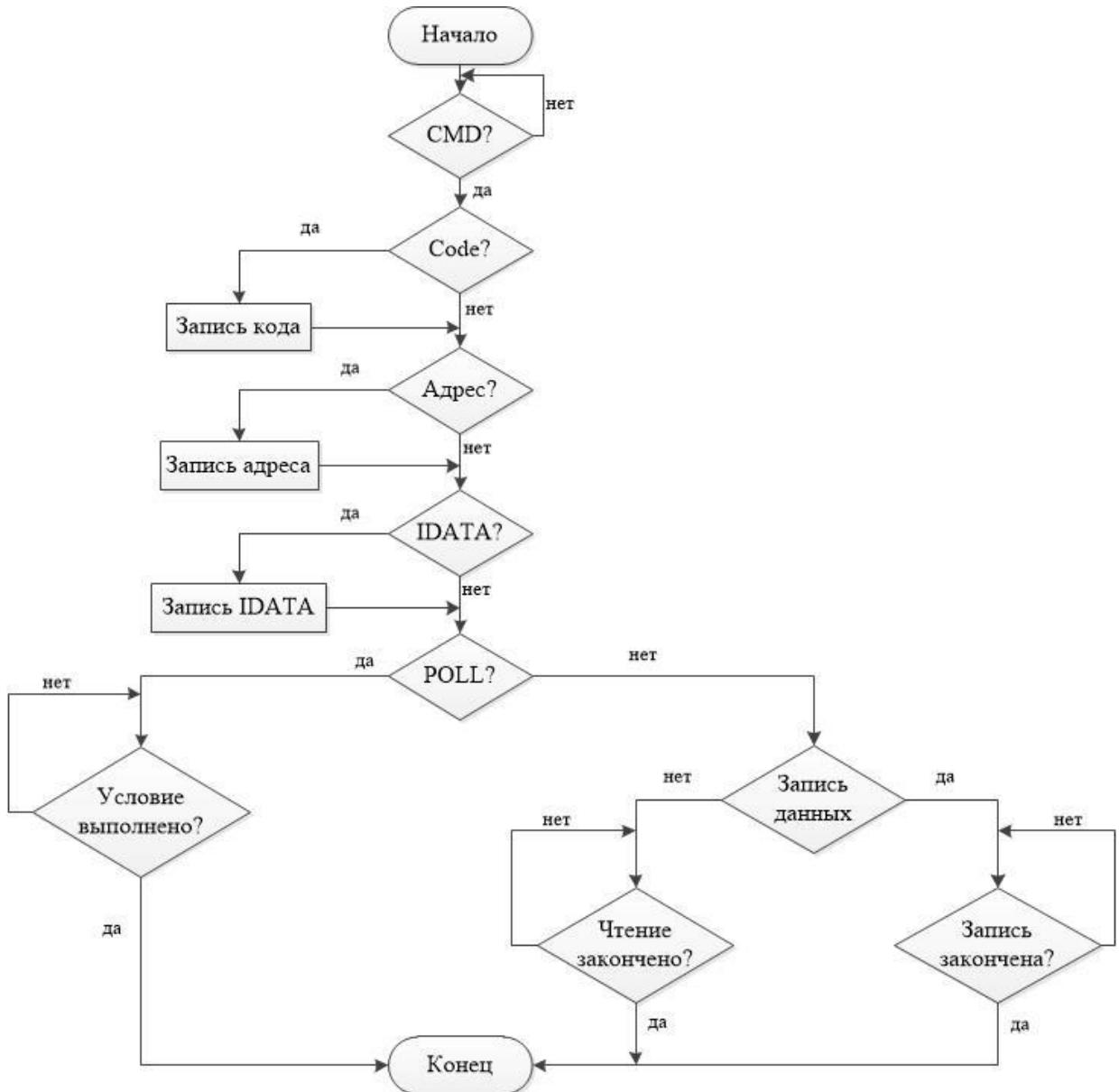


Рисунок 70 – Алгоритм работы автомата контроллера в периферийном режиме

Чтение из МФП

Для того чтобы выполнить команду чтения из МФП (периферийный режим), необходимо выполнить следующие настройки:

- установить требуемые значения полей регистра CTRL:
 - 1) CSHIGH, RFCLK, FBCLK, MODE3, DUAL, DMAEN и INTEN;
 - 2) остальные поля регистра CTRL должны быть нулевыми;
- установить требуемые значения регистров IDATA и ADDRESS;
- установить требуемые значения полей регистра CMD:
 - 1) DATALEN – записать количество считываемых из МФП байт данных;
 - 2) POLL и DOUT – записать нули (поллинга нет, данные читаем);

- 3) INTLEN – записать требуемое значение количества промежуточных данных (это значение варьируется от 0 до 7 байт; в случае если это значение больше четырех, т.е. количество байт больше, чем можно сохранить в регистре IDATA – в этом случае 5, 6 и 7 байты будут равны нулю);
- 4) FIELDFORM, FRAMEFORM и OPCODE – записать исходя из данных протокола информационного взаимодействия с данной МФП. Следует помнить, что запись информации в регистр CMD возможна только в том случае, если биты MCINIT и CMD регистра статуса STAT оба равны «0».

Если регистр команд был записан успешно, о чем свидетельствует установка бита CMD регистра статуса STAT, то автомат управления SPIFI выходит из состояния IDLE и переходит в состояние CSWAIT, в котором сигнал RB деактивируется и позволяет счетчику сдвигов считать такты сдвига CSHIGH + 1 раз. Сигнал SPIFI_CS в течение всего этого времени остается неактивным. По завершению состояния CSWAIT автомат активизирует сигналы «SPIFI_CS» и «TRI» (этот сигнал открывает тристабильный буфер-данные переходят в МФП), а затем переходит в состояние «TRCODE», если поле FRAMEFORM предусматривает передачу кода операции «OPCODE». В этом состоянии автомат записывает «OPCODE» в сдвиговый регистр, а затем сдвигает «OPCODE» в МФП. Счетчик сдвигов при этом считает такты сдвига и после N тактов, автомат перейдет в состояние выдачи адреса «ADDR», если поле FRAMEFORM предусматривает передачу адреса. А далее будут выведены промежуточные данные, если поле INTLEN больше нуля. Длительность каждого состояния также равна N. Считает эту длительность также счетчик тактов сдвига. На этом передача данных в МФП заканчивается, и автомат переходит в состояние «RDDATA».

В этом состоянии сигнал «TRI» деактивируется, что позволяет сдвигать информацию из МФП в сдвиговый регистр. Процесс сдвига будет продолжаться до тех пор, пока из МФП не будет считано «DATALEN» байтов. По мере записи считанных из МФП данных в FIFO буфер блок сопряжения с шиной АНВ периодически устанавливает сигнал запроса DMA, а контроллер DMA осуществляет перекачку данных в память SOC (если DMA запросы разрешены установкой бита DMAEN в регистре CTRL). По завершению процесса приема данных из МФП автомат управления переходит в состояние «EOCMD», в котором автомат сбрасывает бит CMD регистра статуса STAT и устанавливает запрос прерывания INT активизацией сигнала «Set», если запросы

прерываний не запрещены битом INTEN регистра CTRL. Сброс сигнала INT осуществляется записью единицы в бит INTRQ регистра статуса STAT.

Запись в МФП

Запись в МФП не осуществляется напрямую, в отличие от чтения. Запись выполняется в два этапа:

- сначала производится запись данных во внутренний буфер МФП – это обычная статическая память размером со страницу флэш-памяти;
- затем выполняется команда программирования страницы, когда данные из буфера переписываются в флэш-память.

Запись данных в буфер МФП осуществляется точно так же, как и чтение из МФП, за одним исключением – команда и данные поступают в одном направлении (в МФП), поэтому бит DOUT регистра команд CMD должен быть установлен в единицу и сигнал «TRI» будет активен все время выполнения команды.

Выдача команды на программирование страницы памяти осуществляется точно так же, как и выдача команды чтения, за исключением того, что бит POLL регистра команд CMD нужно установить в единицу. После того, как выдача команды будет завершена, управляющий автомат МФП начнет передавать в регистр сдвига байт статуса МФП.

Данные для программирования записываются в регистр DATA.

Затем по биту регистра статуса МФП, который устанавливается в определенное состояние, можно определить момент завершения процесса программирования страницы. Перед выдачей команды на программирование страницы в биты поля DATALEN[2:0] нужно записать номер бита статуса, в поле DATALEN[3] – значение, при котором процесс программирования будет завершен. Запись значений байта статуса в выходной буфер FIFO контроллера не осуществляется. После завершения процесса программирования страницы автомат управления выполняет те же действия, что и при выполнении команды чтения.

Стирание страницы/блока МФП

Команда стирания страницы/блока МФП выполняется абсолютно так же, как и команда программирования страницы, разница только в кодах операций.

Выключение контроллера

Остановка выполнения любой команды может быть инициирована установкой бита RESET регистра статуса STAT. Программное обеспечение должно убедиться, что контроллер вышел из режима выполнения команды, читая регистр статуса до тех пор, пока биты MCINIT или CMD не станут равны нулю.

3.15.4.4 Алгоритм работы с контроллером ПДП

Для чтения из микросхемы Flash памяти с использованием ПДП в периферийном режиме необходимо настроить канал ПДП следующим образом:

```
CHx_CFG.READ_MODE = 0; // чтение в режиме периферийного уст-ва  
CHx_CFG.READ_INCREMENT = 0; // чтение выполняется из одного адреса  
CHx_CFG.READ_SIZE = 2; // чтение 4х байтными словами  
CHx_CFG.READ_BURST_SIZE = 2; // BURST словами  
CHx_CFG.READ_REQUESTS = 7; // линия SPIFI  
LEN = 16; //передача 4х слов  
SRC = 0x00070014;
```

Контроллер SPIFI должен быть настроен в соответствии с описанием раздела «Чтение из МФП», за следующим исключением:

- в поле CMD.DATALEN указывается число принимаем байт из МДП, при этом нужно иметь ввиду, что запрос ПДП формируется при наличии не менее 4 байт в FIFO. Если в FIFO осталось 1-3 байта их необходимо вычитывать вручную.
- необходимо разрешить формированием запроса к ПДП CNTR. DMAEN = 1

Пример чтения с использованием ПДП представлен ниже. На рисунке показаны следующие события (передача начинается после отправки CMD):

1. чтение 3 байт с МФП (количество байт в FIFO показывает сигнал SPIFI_FIFO_CNT);
2. чтение 4 байта. Установка запроса DMA (SPIFI_DMA_R_REQ);
3. формирование контроллером ПДП запроса на чтение по шине АНВ по адресу 0x00070014 (регистр DATA). Чтение из контроллера SPIFI данных;
4. запись полученных данных контроллером ПДП в ОЗУ.

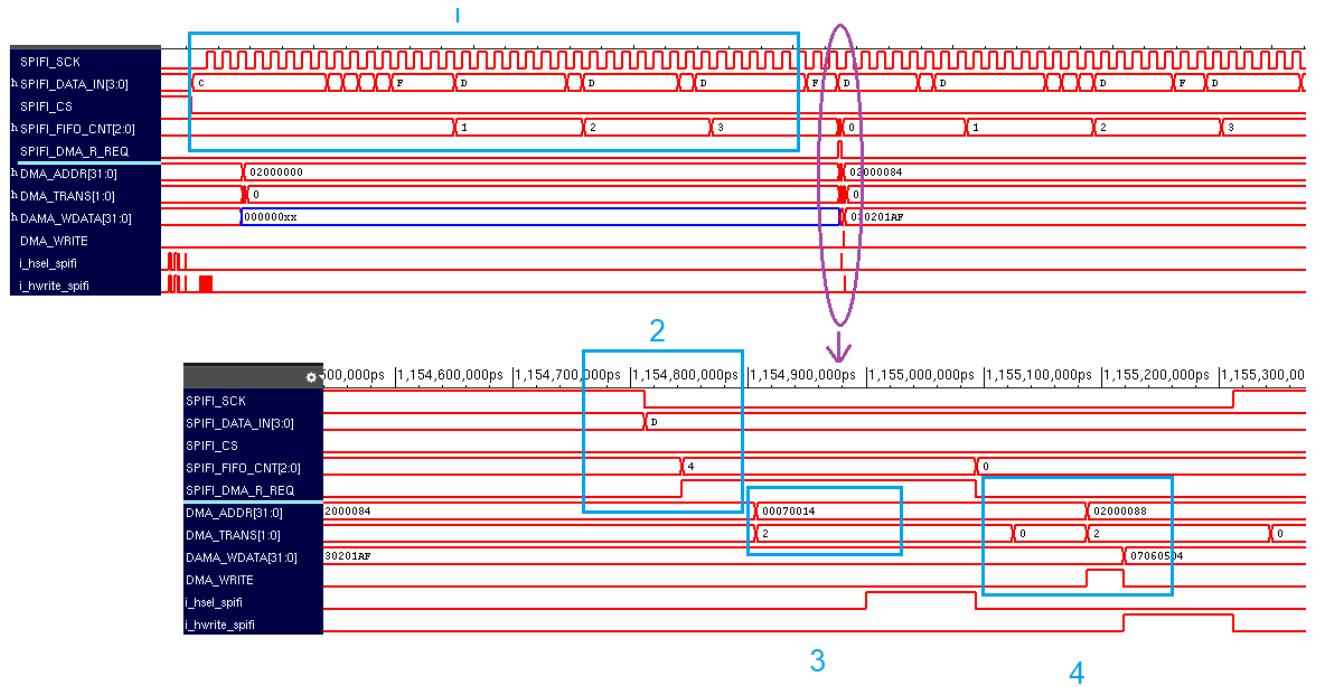


Рисунок 71 – Алгоритм чтения с использованием ПДП

Для записи в микросхему Flash памяти с использованием ПДП необходимо настроить канал ПДП следующим образом:

```
CHx_CFG.WRITE_MODE = 0; // запись в режиме периферийного уст-ва
CHx_CFG.WRITE_INCREMENT = 0; // запись выполняется в один адрес
CHx_CFG.WRITE_SIZE = 2; // запись 4x байтными словами
CHx_CFG.WRITE_BURST_SIZE = 2; //BURST словами
CHx_CFG.WRITE_REQUESTS = 7; // линия SPIFI
LEN = 16; // передача 4x слов
DST = 0x00070014;
```

Контроллер SPIFI должен быть настроен в соответствии с описанием раздела «Запись в МФП», за следующим исключением:

- в поле CMD.DATALEN указывается число принимаем байт, соответствующее значению LEN ПДП;
- необходимо разрешить формированием запроса к ПДП CNTR.DMAEN = 1.

Пример записи с использованием ПДП представлен ниже. На рисунке показаны следующие события:

1. инициализация SPIFI. Отправка CMD. Запрос на запись устанавливается автоматически, если FIFO пустое;
2. чтение контроллером ПДП слова из данных из ОЗУ;

3. формирование контроллером ПДП запроса на запись по шине АНВ по адресу 0x0070014 в регистр SPIFI.DATA. После этого начинается передача в МФП четырех байт;
4. окончание передачи, опустошение FIFO установка запроса ПДП на запись.

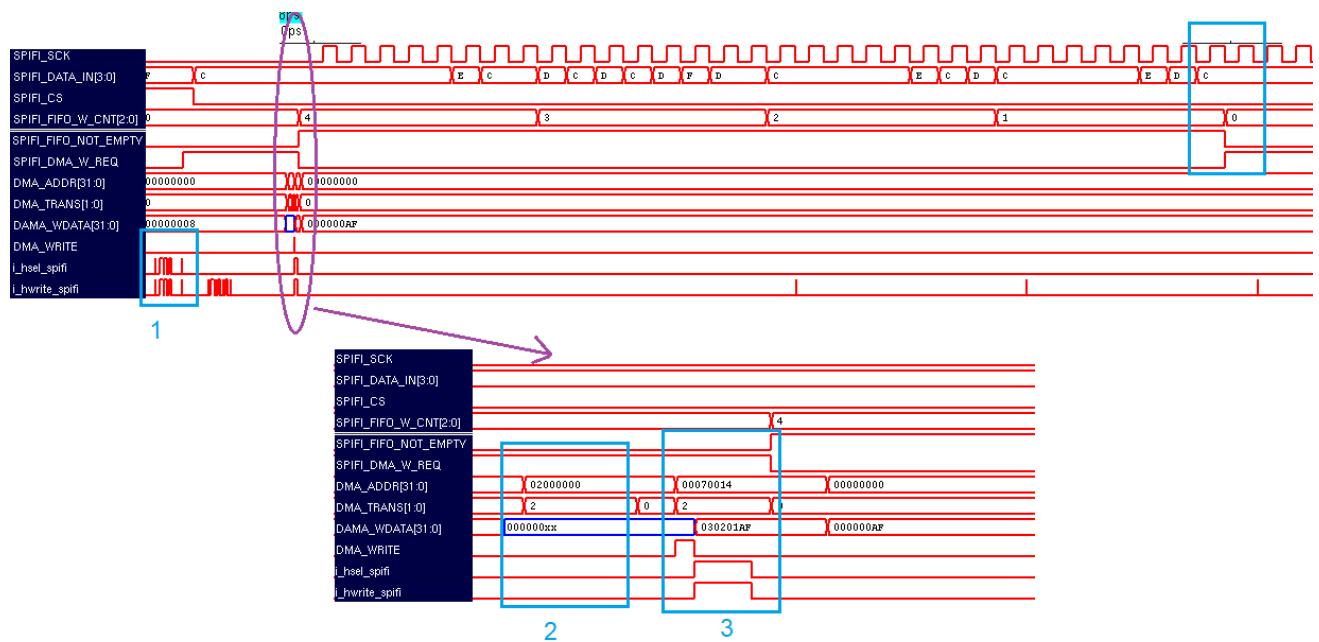


Рисунок 72 – Алгоритм записи с использованием ПДП

3.15.4.5 Работа контроллера в режиме работы с памятью

В данном режиме контроллер программируется практически так же, как и в периферийном режиме чтения за исключением того, что вместо регистра CMD используется регистр MCMD. Поля DATALEN, POLL и DOUT не используются (должны быть нулевыми), а назначение остальных полей аналогично назначению полей регистра CMD.

Алгоритм работы автомата контроллера в режиме работы с памятью приведен на рисунке 73.

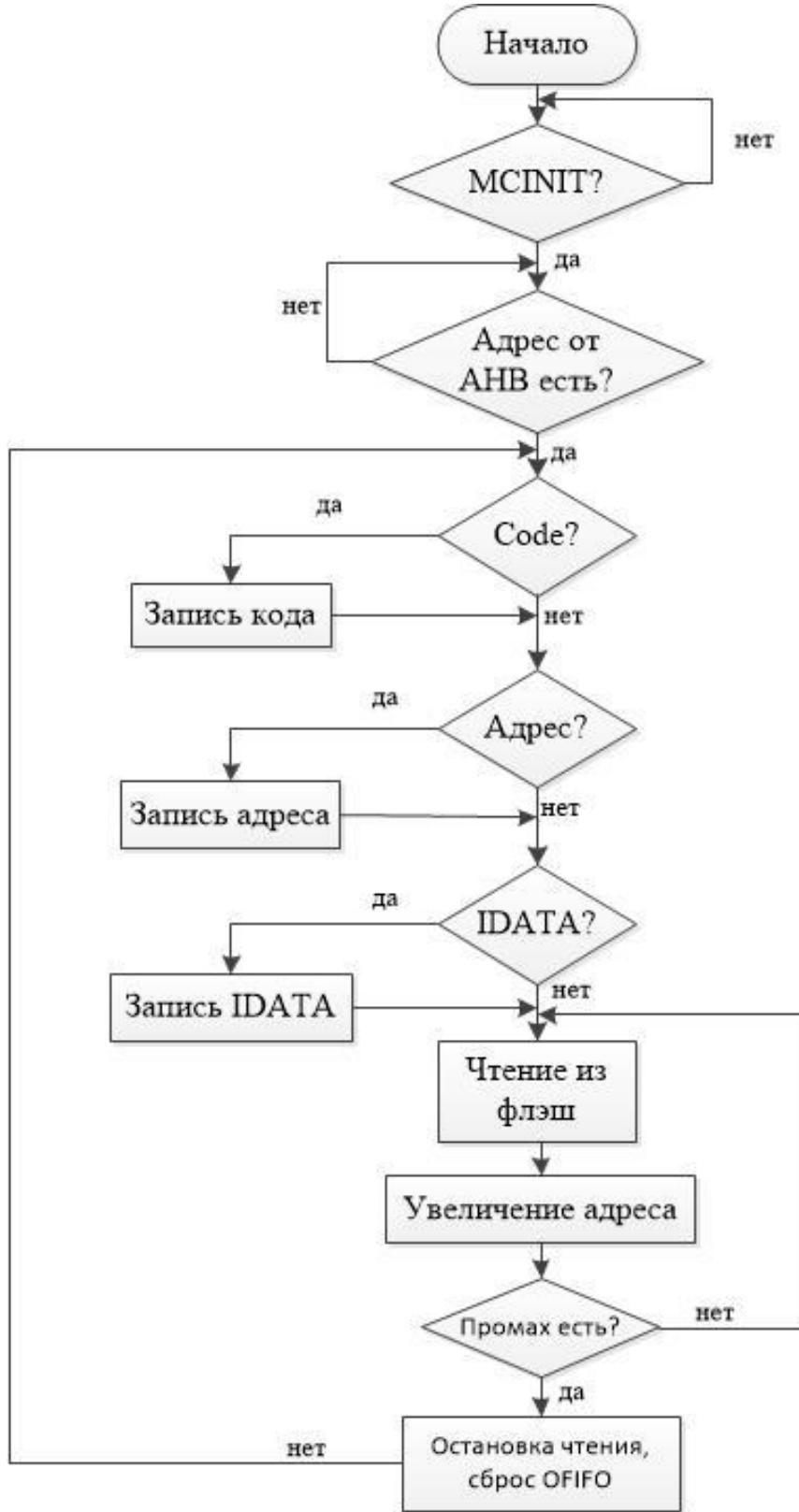


Рисунок 73 – Алгоритм работы автомата контроллера в режиме работы с памятью

Перед тем, как начать обращения к памяти по адресам загрузки, процессор должен настроить регистры контроллера CTRL, IDATA и MCMD (остальные регистры используются для периферийного режима). Правила настройки этих трех регистров описаны в предыдущих пунктах (регистры MCMD и CMD имеют одинаковые поля).

Обращения к области памяти загрузки должны осуществляться только после успешной инициализации контроллера в режиме работы с памятью, когда бит MCINIT будет установлен в «1». Попытки обращения к памяти загрузки до инициализации контроллера, а также попытки записи в загрузочную память после инициализации закончатся выдачей ошибки по АНВ (HRESP = 1). Как только бит MCINIT будет установлен в «1», контроллер перейдет в режим ожидания первого обращения процессора в адресном пространстве загрузки. Как только первое обращение произойдет, контроллер запишет в МФП код операции (это будет Continuous READ), адрес операции (адрес первой транзакции в адресном пространстве загрузки), необходимое количество байтов промежуточных данных IDATA (если поле INTLEN не равно «0»), а затем перейдет к выполнению алгоритма чтения из МФП. Считываемые из МФП данные помещаются в буфер OFIFO так же, как это делается в периферийном режиме. Таким образом, в буфер OFIFO будет укладываться сегмент памяти из области загрузки и, если данные будут считываться по последовательным адресам из адресного пространства загрузки, то усредненная скорость загрузки кодового сегмента будет достигать четырех тактов сигнала HCLK за один байт считываемой информации. Для частоты HCLK 200 МГц эта скорость будет достигать значения 50 Мбайт/с. При этом производительность шины АНВ на такой частоте составит 800 Мбайт/с. Таким образом, в среднем транзакция по шине АНВ будет занимать 16 тактов HCLK (из них 15 тактов ожидания).

Процесс чтения типа Continuous READ будет продолжаться до тех пор, пока считывание кодового сегмента будет осуществляться по последовательным адресам. В случае если транзакция чтения будет произведена по непоследовательному адресу, то будет сформирован сигнал промаха. Автомат контроллера постоянно опрашивает сигнал промаха, и в случае его активизации процесс чтения данных из МФП завершается, сигнал CS деактивируется, буфер OFIFO сбрасывается (данные при этом пропадают), затем контроллер инициирует новую последовательность инициализации команды Continuous READ, но уже с новым адресом.

Вывод контроллера из режима работы с памятью осуществляется установкой бита RESET регистра статуса STAT. Автомат контроллера, увидев сигнал RESET, корректно закончит процедуру Continuous READ, выключит МФП, деактивировав сигнал SPIFI_CS, сбросит буфер OFIFO и перейдет в режим ожидания сигнала MCINIT.

3.15.5 Базовые спецификации SPI интерфейса и Flash драйвера

Базовые спецификации SPI интерфейса и Flash драйвера:

- полнодуплексный синхронный режим работы для стандартного режима SPI;
- полуудуплексный режим работы для многоканального режима работы SPI;
- 8 – битный формат посылок;
- 8 – битный делитель частоты системной шины;
- изменяемые фаза и полярность тактового сигнала;
- изменяемое направление сдвига данных MSb – first или LSb – first;
- наличие флагов: опустошения буфера передачи; наличия данных в приемном буфере; занятости; окончания транзакции;
- каждая команда состоит из пяти сегментов (инструкция + адрес + байты режима + цикл ожидания + данные), а каждый сегмент в свою очередь может быть сконфигурирован произвольным образом;
- 8 – битная инструкция может быть добавлена в команду путем установки соответствующего бита в единицу;
- 24 / 32 – битный адрес может быть добавлен в команду путем установки соответствующего бита в единицу, а также для обеспечения обратной совместимости с устаревшими моделями микросхем памяти есть возможность выбора длины адреса 24 – бита или 32 – бита;
- байты режима необходимы для высокоскоростного обмена данными по двум/четырем линиям Quad SPI (позволяют не повторять команду при каждой транзакции, если команда повторяется);
- циклы ожидания необходимы для выполнения некоторых инструкций внутри микросхемы FLASH-памяти и могут быть добавлены в команду путем установки соответствующего бита в единицу, а также для тонкой подстройки под данную микросхему памяти есть возможность выбора длины данного сегмента;
- сегмент данных может быть добавлен в команду путем установки соответствующего бита в единицу, а также может быть выбрана длина сегмента (1 байт).

Ниже представлены временные диаграммы, демонстрирующие типовые сценарии обмена данными по последовательному интерфейсу с микросхемой FLASH-памяти.

На рисунке 74 показан простейший случай, когда команда состоит только из одной инструкции. Такая команда, например, может быть использована для разблокирования режима программирования памяти.

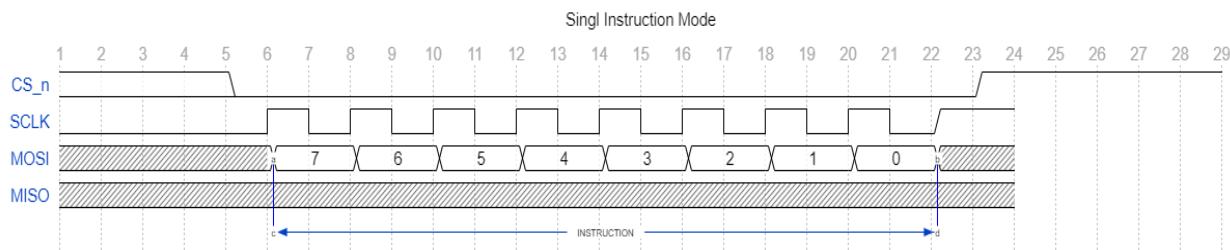


Рисунок 74 – Временная диаграмма команды из одной инструкции

На рисунке 75 показана команда, содержащая инструкцию, за которой следуют передаваемые данные без добавления адреса и задержки. Такая команда может быть использована для конфигурирования микросхемы памяти, например, переключение в 32-битный режим адресации.

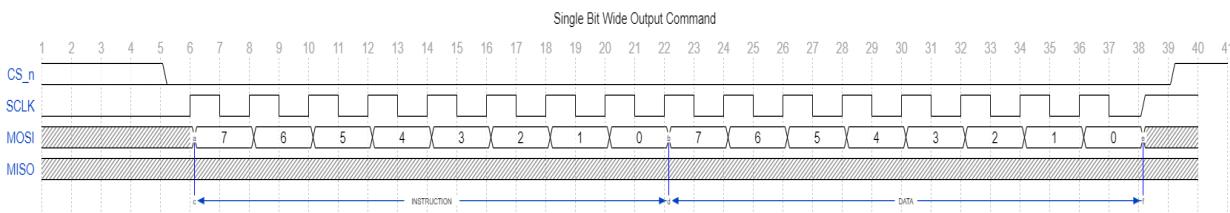


Рисунок 75 – Временная диаграмма инструкции без данных без добавления адреса и задержки

На рисунке 76 показана команда, содержащая инструкцию, за которой следуют принимаемые данные без добавления адреса и задержки. Такая команда может быть использована для чтения статус-регистра памяти, если требуется уточнить доступность микросхемы для выполнения последующих операций.

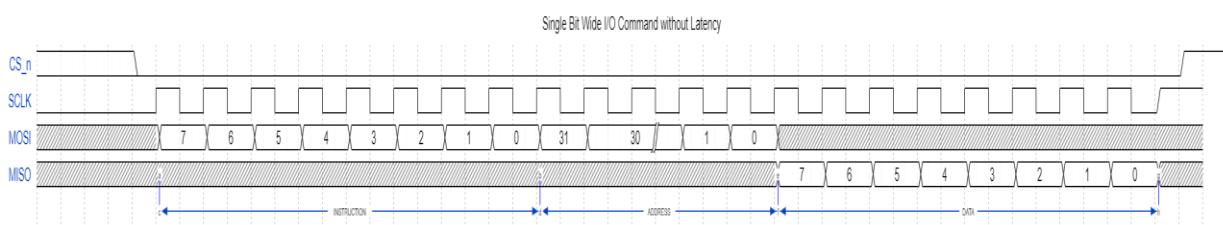


Рисунок 76 – Временная диаграмма инструкции с данными без адреса

На рисунке 77 показана команда, содержащая инструкцию, за которой следуют адрес и принимаемые данные без добавления задержки. Такая комбинация используется для чтения идентификационных номеров микросхемы и производителя, а также чтения всех информационных полей, прописанных стандартом JEDEC. Таким образом,

достаточно отправить команду 0x90 и 24-битный адрес 0x000000 для получения идентификационных номеров или 0x9F для доступа к универсальному интерфейсу взаимодействие с микросхемами FLASH-памяти CFI JEDEC. Данный случай может быть использован и для чтения данных из памяти путем отправки соответствующей команды READ для данной микросхемы и желаемый адрес, с которого начнется процесс, удерживая CS в низком состоянии можно продолжать непрерывное чтение данных (адрес инкрементируется автоматически), а по достижении конца адресного пространства, чтение продолжится с нулевого адреса. В данном формате также возможно и программирование памяти, но при условии, что предварительно была отправлена команда разблокировки режима записи.

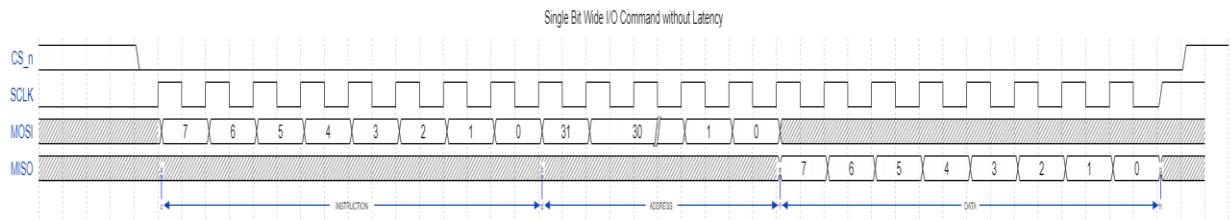


Рисунок 77 – Временная диаграмма инструкции с адресом и принимаемыми данными

На рисунке 78 показана команда, содержащая инструкцию, за которой следуют адрес, задержка и принимаемые данные. Данное сочетание требуется для команд с приставкой FAST, например, FAST-READ. Такие команды позволяют запросить данные на чтение и после определённой задержки, получить отклик от памяти на значительно большей частоте, чем при использовании стандартных команд. Количество циклов ожидания зависит от микросхемы и ее конфигурации и должно уточняться в технической документации на соответствующую микросхему.

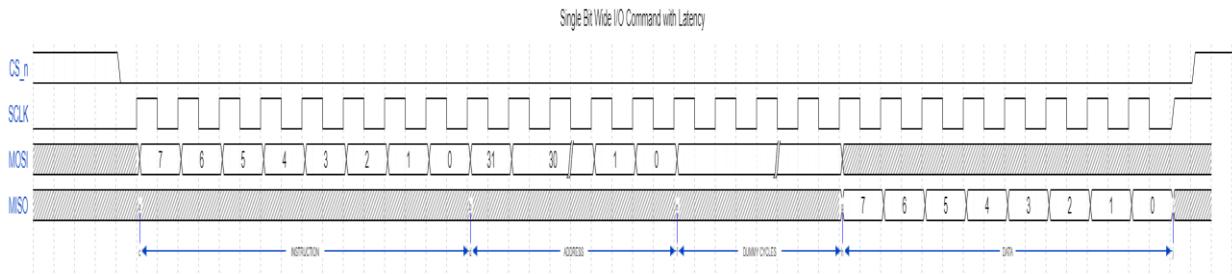


Рисунок 78 – Временная диаграмма инструкции с адресом и принимаемыми данными и задержкой

3.16 Ускоритель симметричной криптографии (CRYPTO)

Ускоритель симметричной криптографии (крипто-блок) реализует три специализированных преобразования:

- AES128, FIPS 197 128-битный блочный шифр, со 128-битным ключом;
- «Магма», ГОСТ 34.12–2018 64-битный блочный шифр;
- «Кузнецчик», ГОСТ 34.12–2018 128-битный блочный шифр.

Крипто-блок состоит из трёх вычислительных ядер с общим управляемым конечным автоматом и общим буфером ввода вывода.

В процессе вычислений крипто-блок настраивается на выполнение одного из трёх преобразований, при этом работает только одно из трёх ядер. В процессе работы крипто-блока формирует флаги разрешения записи и чтения для ПДП контроллера, сообщающие о состоянии готовности входных и выходных буферов к передаче данных.

Условная структурная схема крипто-блока представлена на рисунке 79.

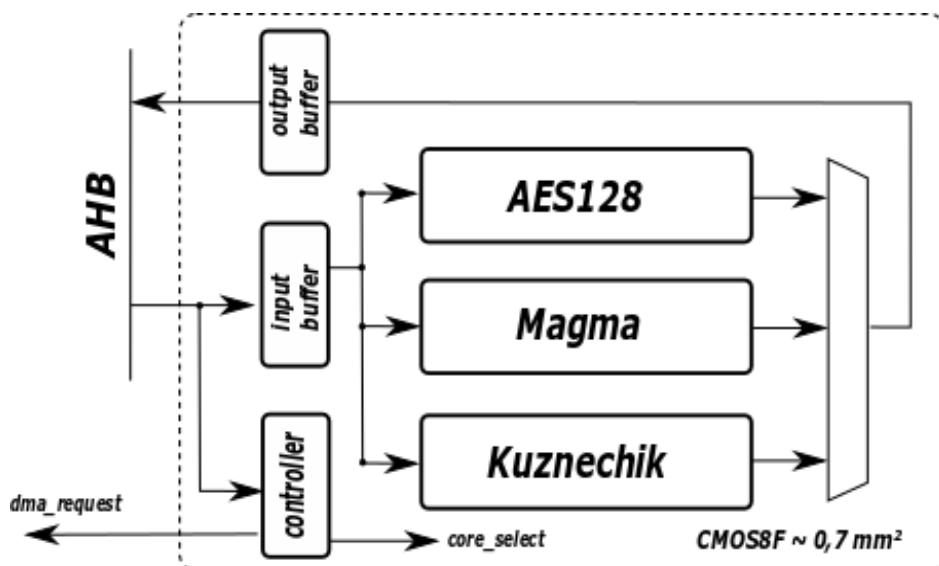


Рисунок 79 – Структурная схема крипто-блока

Крипто-блок работает полностью в автоматическом режиме, т.е. для инициализации вычислений необходимо передать в блок нужную порцию данных, по завершении передачи необходимого количества данных инициализируется вычисление, соответствующее регистру режима и типу данных (развёртка ключа, шифрование).

Доступ к крипто-блоку осуществляется словами по 32-бита. Регистры крипто-блока приведены в таблице 113.

Таблица 114 – Регистры крипто-блока

Обозначение	Смещение	Начальное значение	Описание	Доступ
BLOCK	0x0	-	Буфер данных для шифрования	RW
KEY	0x4	-	Буфер данных ключа	WO
INIT	0x8	-	Буфер данных вектора инициализации	WO
CONFIG	0xC	0x600	Регистр конфигурации	RW

Описание полей регистра конфигурации приведено в таблице 115.

Таблица 115 – Описание полей регистра конфигурации

Обозначение	Разряд	Начальное значение	Описание	Доступ
READ STATUS	11	0x1	Буфер данных доступен для чтения: 0 – недоступно; 1 – доступно	RO
WRITE STATUS	10	0x1	Буфер данных доступен для записи: 0 – недоступно; 1 – доступно	RO
READY	9	0x0	Состояние доступности модуля: 0 – занято; 1 – доступно	RO
C_RESET	8	0x0	Сброс счётчиков, загружаемых/выгружаемых данных	RW

Обозначение	Разряд	Начальное значение	Описание	Доступ
ORDER_MODE	7	0x0	Выбор порядка загрузки/выгрузки: 0 – от младшего слова к старшему; 1 – от старшего слова к младшему	RW
SWAP_MODE	6:5	0x0	Регистр выбора режима перестановки слова: 0 – нет перестановки; 1 – перестановка по полуслову; 2 – перестановки по байтам; 3 – перестановка по битам	RW
MODE_SEL	4:3	0x0	Регистр выбора режима шифрования: 0 – ECB; 1 – CBC; 2 – CTR	RW
CORE_SEL	2:1	0x0	Регистр выбора алгоритма шифрования: 0 – «Кузнецик»; 1 – «Магма»; 2 – «AES»	RW
DECODE	0	0x0	Задаёт режим хода вычислительного ядра: 0 – прямой ход (шифрование); 1 – обратный ход (расшифровка)	RW

3.16.1 Порядок работы с крипто-блоком

Алгоритм «Кузнецик»:

- установка регистра конфигурации;
- установка 256-ти битного ключа – загрузка восьми 32-х разрядных слов по адресу 0x4. При этом ключ всегда должен загружаться в режиме шифрования (CONFIG.decode = 0). По получении восьми слов крипто-блок начнёт процедуру развёртки ключа, при этом до завершения процедуры развёртки ключа блок будет находиться в неактивном состоянии с запретом на доступ к блоку со стороны шины (сигнал готовности ready в неактивном состоянии);
- установка режима хода вычислительного ядра в регистре конфигурации;
- установка 128-ти битного блока данных – загрузка четырех 32-х разрядных слов по адресу 0x0. По получении четырех слов крипто-блок начнёт процедуру

шифрования, при этом будет перекрыт доступ со стороны шины (ready в неактивном состоянии) и опущен флаг разрешения записи данных для ПДП контроллера;

- чтение 128-ти битного блока данных.

По завершении шифрования к крипто-блоку будет разрешён доступ со стороны шины, при этом флаг на чтение данных для ПДП контроллера будет установлен в активное состояние. По завершении чтения четырех 32-х разрядных слов из крипто-блока флаг на чтение для ПДП контроллера будет установлен в неактивное состояние, а флаг на запись в активное. Крипто-блок готов к обработке следующего блока.

Порядок шифрования для алгоритмов «Магма» и AES128 полностью аналогичны, за исключением разрядности обрабатываемых данных. Для алгоритма «Магма» - 256 битный ключ, 64-х битный блок. Для AES128 128-ми битный ключ, 128-ми битный блок.

При шифровании в режимах сцепления блоков (CBC), инициализация первого блока для сцепления реализуется автоматически после загрузки четырех 32-х разрядных слов в регистр вектора инициализации. Последующее сцепление блоков будут реализовано относительно установленного в регистре значения, до момента его перезагрузки.

При шифровании в режиме гаммирования (CTR) в регистр INIT следует записать вектор инициализации, длина которого равна половине блока, и после, такое же количество нулей. В итоге размер данных загруженных в INIT будет равно размеру блока.

При шифровании в режиме простой замены (ECB) вектор инициализации загружать не нужно.

3.16.2 Порядок работы совместно с ПДП контроллером

Для обработки данных без участия ядра возможно использовать ПДП контроллер. Для этого необходимо настроить ускоритель криптографии на работу в необходимом режиме, загрузить таблицы (при необходимости), проверить доступность статус ускорителя (ready = 1) и настроить два канала ПДП контроллера, один на загрузку данных для преобразования, другой для чтения обработанных данных.

Для записи данных с использованием ПДП необходимо настроить канал ПДП следующим образом:

```
CHx_CFG.WRITE_MODE = 0; //запись в режиме периферийного уст-ва
```

```
CHx_CFG.WRITE_INCREMENT = 0; // запись выполняется в один адрес  
CHx_CFG.WRITE_SIZE = 2; // запись 4х байтными словами  
CHx_CFG.WRITE_BURST_SIZE = 2; // BURST словами  
CHx_CFG.WRITE_REQUESTS = 2; // линия CRYPTO  
LEN = <размер сообщения>; // кратно размеру блока для шифрования  
DST = 0x00080000; // регистр BLOCK.
```

Для чтения данных после преобразования из блока с использованием ПДП необходимо настроить канал ПДП следующим образом:

```
CHx_CFG.READ_MODE = 0; // чтение в режиме периферийного уст-ва  
CHx_CFG.READ_INCREMENT = 0; // чтение выполняется из одного адреса  
CHx_CFG.READ_SIZE = 2; // чтение 4х байтными словами  
CHx_CFG.READ_BURST_SIZE = 2; // BURST словами  
CHx_CFG.READ_REQUESTS = 2; // линия CRYPTO  
LEN = <размер сообщения>; // кратно размеру блока для шифрования  
SRC = 0x00080000; // регистр BLOCK
```

Признаком окончания обработки всего сообщения будет выполнении заданий двумя каналами ПДП

3.17 Контроллер блока однократно программируемой памяти (OTP)

Данный блок предназначен для осуществления доступа к массиву блока antifuse OTP на запись и чтение со стороны интерфейса APB.

Примечание - В исполнении K1948BK01B8 и K1948BK01B5 блок OTP не используется.

Блок является совместимым с Hard IP antifuse OTP объемом 256 бит с организацией 8x32 otp_block_8x32_pwr_off, разработанным для серийной технологии CMOSF8_4M_5V ПАО Микрон. На рисунке 80 приведена схема блока.

Конфигурация блока antifuse OTP Hard IP:

- информационная емкость – 256 бит;
- количество бит в строке массива – 32 бита;
- количество строк в массиве – 8+1 (присутствует одна тестовая строка);
- разрядность входа – 32 бита;
- разрядность выхода – 32 бита;
- тестовые столбцы – 1.

Блок antifuse OTP 256 бит

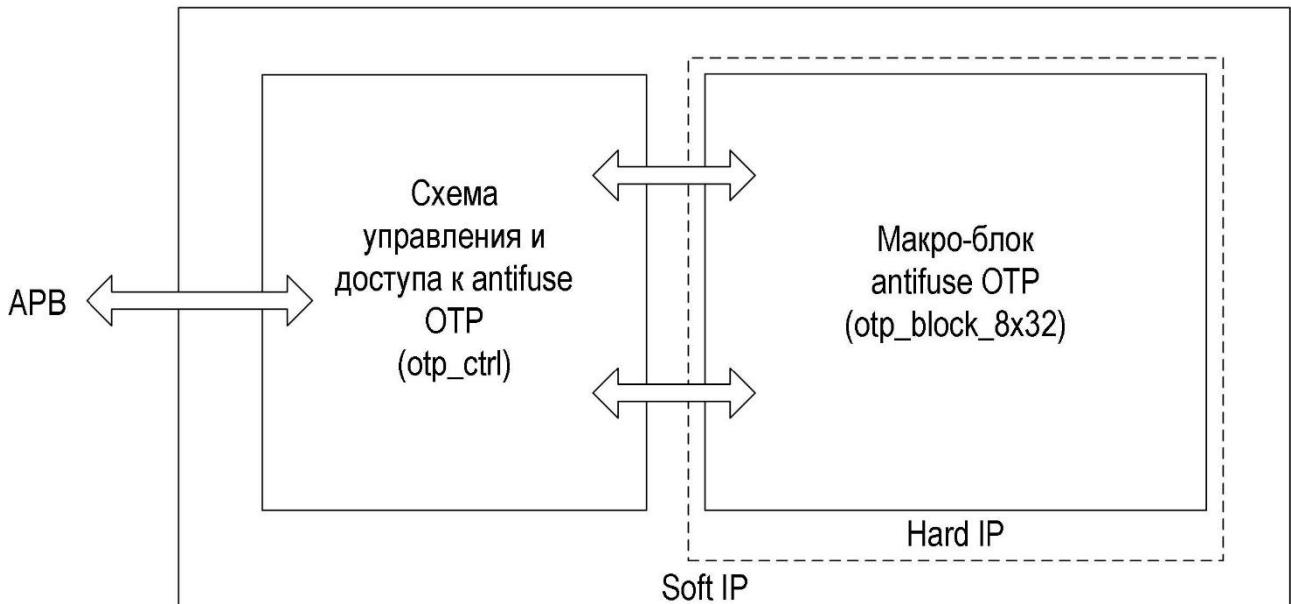


Рисунок 80 – Схема блока OTP

Особенности контроллера OTP:

- запись через APB-интерфейс в 3 этапа;
- чтение через APB-интерфейс в 2 или 3 этапа;
- возможность доступа к тестовой строке и тестовому столбцу для производственного тестирования Hard IP;
- возможность задания количества тактов для операций записи и чтения через регистры управления.

3.17.1 Чтение

Схема позволяет выполнять чтение по интерфейсу APB в 2 или 3 этапа:

- запись адреса (OTPA) – чтение регистра данных (OTPDAT);
- запись адреса (OTPA) – опрос флага готовности (OTPSTA.BSY) – чтение регистра данных (OTPDAT).

Если используется двухэтапное чтение, то вместо процедуры проверки флага готовности данных вводятся такты ожидания интерфейса APB, в течение которых сигнал READY удерживается в «0». Если используется трехэтапное чтение, READY удерживается в «1», но требуется дополнительная проверка флага готовности (OTPSTA.BSY). Для переключения между режимами используется регистр OTPCON.APBNWS. При OTPCON.APBNWS = 1 используется режим чтения в три этапа (без тактов ожидания APB).

Если OTPCON.APBNWS = 0 при чтении регистра OTPDAT происходит автоматическое инкрементирование адреса OTPA. Таким образом, запись адреса не требуется, если необходимо считать слово, имеющее адрес на один больше предыдущего. При автоматическом инкрементировании адреса происходит только изменение разрядов OTPA[2:0]. Если выполняется инкрементирование адреса, содержащего OTPA[2:0] = 0b111, то новое значение будет содержать OTPA[2:0] = 0b000, а другие разряды не изменятся. Если OTPCON.APBNWS = 1 при чтении регистра OTPDAT автоматическое инкрементирование адреса OTPA не происходит.

Если OTPCON.APBNWS = 0, контроллер осуществляет автоматическое считывание слова из Hard IP при каждом чтении регистра OTPDAT.

Если OTPCON.APBNWS = 1, контроллер осуществляет автоматическое считывание слова из Hard IP при каждой записи в регистр OTPA.

3.17.2 Программирование

Питание программирования (вывод VPRG) составляет 8.0–9.0 В, длительность импульса не менее 50 мс. Ток на VPRG до 4.0 мА на одну пробиваемую ячейку. Необходимо обязательно закладывать повторные (до 10 раз) операции программирования в случае, если нужная информация не прошилась с первого раза, а также по возможности изменение в большую сторону времени (до 100 мс) и напряжения программирования (до 10 В).

После окончания операции записи необходимо опустить VPRG до уровня VDD5 (напряжение считывание на выводе VPRG, диапазон 3,0 В – 5,5 В), либо 0 В.

С первого раза гарантируется запись 16 бит, поэтому одно слово 32 бит рекомендуется программировать за две итерации с поочередным маскированием одной половины слова, затем – другой. Допускается программирование любого числа бит (1–32) за одну итерацию, при этом непрограммируемые биты должны быть маскированы нулями. Также возможен вариант записи сразу всего слова 32 бит с последующими дополнительными прошивками этим же паттерном.

Внутри ОТР блока реализован аналоговый ключ, который коммутирует внутреннюю шину питания с внешним VPRG при программировании или с выходом внутреннего стабилизатора напряжения, если программирование не выполняется. Поэтому возможно проводить считывание сразу после операции записи при высоком

программирующем напряжении. В таком случае для считывания рекомендуется задавать настройку OTPADJ.sel_upp_read_i = 0b000.

При программировании «нули» переходят в «единицы». Обратный переход невозможен. Имеется возможность изменять содержимое прошитого слова путем программирования бит, содержащих «нули». Например, слово «0b0101» может быть изменено на «0b0111», либо на «0b1101», либо на «0b1111».

Для записи слова рекомендуется следовать описанной ниже последовательности действий:

- выполнить запись адреса в регистр OTPA. Если выполняется запись тестового слова, то OTPA[3] = 1, иначе – OTPA[3] = 0. Если выполняется запись тестового столбца, то OTPA[4] = 1, иначе – OTPA[4] = 0;
- выполнить запись данных в регистр OTPDAT;
- опрашивать флаг OTPSTA.BSY до тех пор, пока он не будет очищен.

По окончании процедуры записи происходит автоматическое инкрементирование адреса OTPA. Таким образом, запись адреса не требуется, если необходимо записать новое слово, имеющее адрес на один больше предыдущего.

При автоматическом инкрементировании адреса происходит только изменение разрядов OTPA[2:0]. Если выполняется инкрементирование адреса, содержащего OTPA[2:0] = 0b111, то новое значение будет содержать OTPA[2:0] = 0b000, а другие разряды не изменятся.

3.17.3 Временные ограничения

Чтобы формы сигналов управления, поступающих из Soft IP в Hard IP, соответствовали требованиям к временным параметрам данных сигналов при любых значениях частоты тактового сигнала clk, возможно изменение задержек между некоторыми событиями на временных диаграммах. На рисунке 81 представлены формы сигналов в процедуре записи. На рисунке 82 представлены формы сигналов в процедуре чтения.

На диаграммах использованы следующие обозначение:

- сигналы протокола APB:
 - 1) clk - тактовый вход контроллера OTP
 - 2) PADDR – шина данных;
 - 3) PWRITE – тип операции нашине APB (0-чтение, 1-запись);

- 4) PSEL – сигнал выбора контроллера OTP для обмена данными ;
 5) PENABLE – сигнал разрешения, показывает второй и последующий такты транзакции;
 6) PREADY – сигнал готовности данных;
- сигналы блока OTP:
- 1) we_i – вход блока, Сигнал разрешения записи/программирования «0» – нет записи. «1» – запись
 - 2) re_i – вход блока, Сигнал разрешения чтения «0» – нет чтения. «1» – чтение

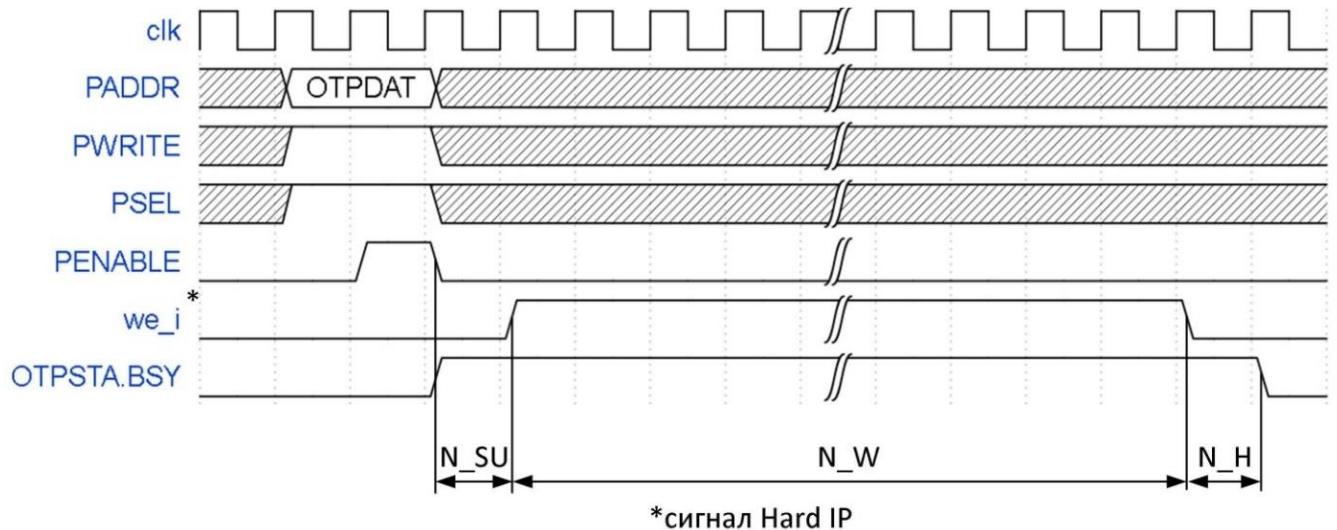


Рисунок 81 – Формы сигналов в процедуре записи

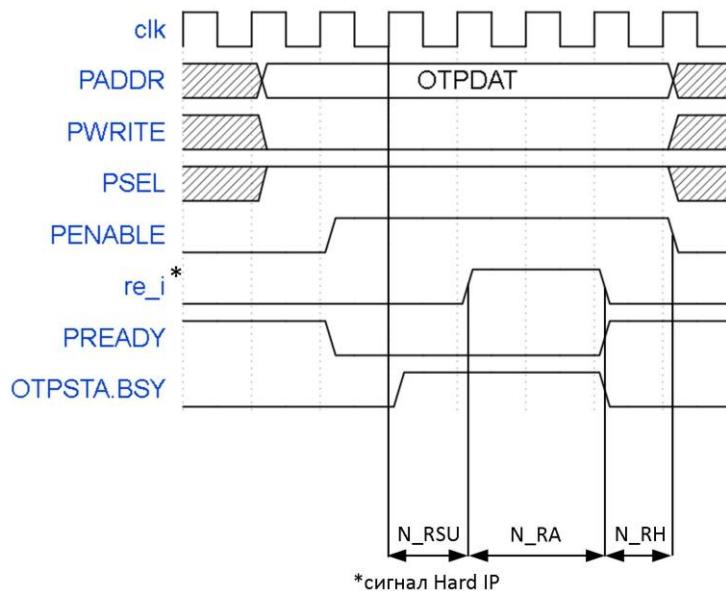


Рисунок 82 – Формы сигналов в процедуре чтения

На рисунках представлены следующие временные интервалы:

- N_SU – время между моментом окончания транзакции на APB и положительным фронтом we_i в тактах. Должно использоваться для обеспечения требования к временам предустановки Hard IP. Рекомендуемое значение $N_{SU} = \text{ceil}(40/\text{Pclk})$, где Pclk - период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа. Пример: 2 такта для частоты Fclk = 33,3 МГц;
- N_W – длительность высокого уровня сигнала we_i (вход Hard IP) в тактах. Рекомендуемое значение $N_W = 50\ 000\ 000$ нс / Pclk, где Pclk – период тактового сигнала в нс. Пример: 1 666 667 тактов для частоты Fclk = 33,3 МГц;
- N_H – время между задним фронтом we_i и задним фронтом OTPSTA.BSY в тактах. Должно использоваться для обеспечения требования к временам удержания Hard IP. Рекомендуемое значение $N_H = \text{ceil}(40/\text{Pclk})$, где Pclk – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа. Пример: 2 такта для частоты Fclk = 33,3 МГц. На рисунках 81 и 82 представлены формы сигналов в процедурах записи и чтения соответственно;
- N_RSU – время между моментом начала транзакции на APB и положительным фронтом re_i в тактах. Должно использоваться для обеспечения требования к временам предустановки Hard IP. Учитывая то, что на входе a_i удерживаются стабильные уровни в результате предыдущей операции записи в регистр OTPA, при частотах ниже 200 МГц данное значение рекомендуется устанавливать равным «0»;
- N_RA – длительность высокого уровня сигнала re_i (вход Hard IP) в тактах. Рекомендуемое значение $N_{RA} = \text{ceil}(40/\text{Pclk})$, где Pclk – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа. Пример: 2 такта для частоты Fclk = 33,3 МГц;
- N_RH – время между задним фронтом re_i и моментом, в который может быть начата новая транзакция на APB. Должно использоваться для обеспечения требования к временам удержания Hard IP. Рекомендуемое значение $N_{RH} = \text{ceil}(40/\text{Pclk})$, где Pclk – период тактового сигнала в нс, ceil

- функция округления до ближайшего большего целого числа. Пример: 2 такта для частоты $F_{clk} = 33,3$ МГц;
- Согласно спецификации Hard IP, после окончания операции чтения до начала следующей за ней операции записи должно пройти не менее 1 мкс.

3.17.4 Доступ к тестовым строке и столбцу

Доступ на запись и чтение тестовых строки и столбца производится аналогично доступу к столбцам и строкам основного массива. Для выбора основного массива OTP необходимо установить $ADDR[4:3] = 0b00$, разряды $ADDR[2:0]$ используются для выбора строки основного массива. Для выбора основных 32-х ячеек тестовой строки необходимо установить $ADDR[4:3] = 0b01$, разряды $ADDR[2:0]$ игнорируются. Для выбора тестового столбца (1 ячейка) тестовой строки необходимо установить $ADDR[4:3] = 0b11$, разряды $ADDR[2:0]$ игнорируются. Для выбора тестового столбца, прилегающего к основному массиву, необходимо установить $ADDR[4:3] = 0b10$, разряды $ADDR[2:0]$ используются для выбора ячейки тестового столбца. При записи в тестовый столбец все разряды данных (OTPDAT) кроме младшего игнорируются. При чтении из тестового столбца все разряды данных (OTPDAT) читаются как 0, кроме младшего.

3.17.5 Режим пониженного энергопотребления

В блоке Hard IP antifuse OTP реализована функция пониженного энергопотребления по сигналу `power_off_i`. Для ввода Hard IP в этот режим необходимо выполнить запись 1 в бит `OTPRTADJ.power_off_i`. Для выведения Hard IP из этого режима необходимо выполнить запись 0 в бит `OTPRTADJ.power_off_i`. В режиме пониженного энергопотребления блок неактивен. После выхода из режима пониженного энергопотребления необходимо выдержать паузу не менее 100 мкс после перехода сигнала `power_off_i` из 1 в 0 перед считыванием информации, переключением `re_i` из 0 в 1. После подачи питания на Hard IP операции чтения и записи могут производиться не ранее чем через 500 мкс.

3.17.6 Регистры

Список регистров и их назначение представлено в таблице 116. В таблицах 117 – 124 представлены регистры контроллера OTP.

Таблица 116 – Регистры контроллера OTP

Обозначение	Смещение	Начальное значение	Описание	Доступ
OTPDAT	0x00	-	Регистр чтения/записи данных	RW
OTPA	0x04	0x00000001	Регистр адреса страницы	RW
OTPCON	0x08	-	Регистр управления	WO
OTPSTA	0x0C	0x00000000	Регистр статуса	RO
OTPDEC	0x10	0x00000002	Регистр дешифратора строк	RO
OTPADJ	0x14	0x1020200	Регистр управления временными параметрами процедуры чтения и доп. настройками	RW
OTPWT1	0x18	0x0000202	Регистр подстройки длительности процедуры записи 1	RW
OTPWT2	0x1C	0x0196E6B	Регистр подстройки длительности процедуры записи 2	RW

Таблица 117 – Регистр чтения/записи данных OTPDAT

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DATA	31:0	-	Сразу после выполнения записи в этот регистр будет запущена внутренняя процедура записи загруженных данных в массив antifuse OTP. Временные параметры данной операции могут быть настроены с помощью других регистров. При чтении из этого регистрачитываются данные, полученные из OTP	RW

Таблица 118 – Регистр адреса страницы OTPA

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:5		Зарезервировано	
ADDR	4:0	0x1	Адрес слова для выполнения записи/чтения. При ADDR[3] = 0 выбран основной массив OTP. При ADDR[3]=1 выбрана тестовая строка. При ADDR[4] = 1 выбран тестовый столбец	RW

Таблица 119 – Регистр управления OTPCON

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:4		Зарезервировано	
MME	3	0x0	Режим ручного управления включен. В этом режиме автоматизированные операции чтения и записи не выполняются. Используется для тестирования. Не рекомендуется изменять в штатном режиме работы. Допускается изменять	WO
man re i	2	0x0	Ручное управление сигналом re i блока Hard IP. Используется для тестирования. Не рекомендуется изменять в штатном режиме работы. Допускается изменять	WO
man we i	1	0x0	Ручное управление сигналом we i блока Hard IP. Используется для тестирования. Не рекомендуется изменять в штатном режиме работы. Допускается изменять	WO
APBNWS	0	0x0	Отключение вставки тактов ожидания в процессе обмена по APB при чтении данных (обращении к регистру OTPDAT на чтение). Если такты ожидания отключены (APBNWS=1), то требуется производить опрос флага OTPSTA.BSY после записи адреса в OTPA до тех пор, пока EESTA.BSY не станет равным 0	WO

Таблица 120 – Регистр статуса OTPSTA

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:1		Зарезервировано	
BSY	0	0x0	Блок занят (выполняется запрошенная операция)	RO

Таблица 121 – Регистр дешифратора строк OTPDEC

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:9		Зарезервировано	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DECO	8:0	0x0	Выход дешифратора строк блока ОТР. Используется для тестирования. Проверка функционирования дешифратора строк осуществляется путем смены адреса тестового столбца, страницы и контроля выходов дешифратора	RO

Таблица 122 – Регистр управления временными параметрами ОТРАДЖ

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:29		Зарезервировано	
power_off_i	28	0x0	Режим пониженного энергопотребления: «0» – Hard IP выведен из режима пониженного энергопотребления и может выполнять операции чтения и записи; «1» – Hard IP введен в режим пониженного энергопотребления, операции записи и чтения запрещены. После вывода Hard IP из режима пониженного энергопотребления требуется некоторое время перед тем, как могут быть начаты новые операции	RW
sel_read_cur_i	27	0x0	Выбор тока считывания: «0» – 2 мА; «1» – 0,2 мА Используется для тестирования. Не рекомендуется изменять в штатном режиме работы.	RW
sel_upp_read_i[2:0]	26:24	0x1	Выбор напряжения чтения 0b000, 0b100 – 2,0 В; 0b001, 0b101 – 2,5 В; 0b111, 0b011 – 3,0 В; 0b010 – VDD18 напряжение питания 1,8 В, подключено к основному LDO; 0b110 – VDD5 диапазон 3.0 - 5.5В	RW
Зарезервировано	23:19		Зарезервировано	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
N_RH	18:16	0x1	<p>Время между задним фронтом <code>re_i</code> и моментом, в который может быть начата новая транзакция на APB. Должно использоваться для обеспечения требования к временем удержания Hard IP. Рекомендуемое значение $N_RH = \lceil 40/Pclk \rceil$, где $Pclk$ – период тактового сигнала в нс, $\lceil \cdot \rceil$ – функция округления до ближайшего большего целого числа.</p> <p>Пример: два такта для частоты $Fclk = 33,3$ МГц. Для корректной работы схемы значение должно быть больше 0</p>	RW
Зарезервировано	15:11	Зарезервировано		
N_RA	10:8	0x1	<p>Длительность высокого уровня сигнала <code>re_i</code> (вход Hard IP) в тактах. Рекомендуемое значение $N_RA = \lceil 40/Pclk \rceil$, где $Pclk$ – период тактового сигнала в нс, $\lceil \cdot \rceil$ – функция округления до ближайшего большего целого числа.</p> <p>Пример: два такта для частоты $Fclk = 33,3$ МГц. Для корректной работы схемы значение должно быть больше 0</p>	RW
Зарезервировано	7:3	Зарезервировано		
N_RSU	2:0	0x0	Время между моментом начала транзакции на APB и положительным фронтом <code>re_i</code> в тактах. Должно использоваться для обеспечения требования к временем предустановки Hard IP. Учитывая то, что на входе <code>i</code> удерживаются стабильные уровни в результате предыдущей операции записи в регистр ОТРА, при частотах ниже 200 МГц данное значение рекомендуется устанавливать равным 0	RW

Таблица 123 – Регистр подстройки длительности процедуры записи 1 OTPWT1

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31:11		Зарезервировано		
N_H	10:8	2	<p>Время между задним фронтом we_i и задним фронтом OTPSTA.BSY в тактах. Должно использоваться для обеспечения требования к временам удержания Hard IP. Рекомендуемое значение N_H=ceil(40/Pclk), где Pclk – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа.</p> <p>Пример – ldf такта для частоты Fclk = 33,3 МГц. Для корректной работы схемы значение должно быть больше 0</p>	RW	
Зарезервировано	7:3		Зарезервировано		
N_SU	2:0	2	<p>Время между моментом окончания транзакции на APB и положительным фронтом we i в тактах. Должно использоваться для обеспечения требования к временам предустановки Hard IP. Рекомендуемое значение N_SU ==ceil(40/Pclk), где Pclk – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа.</p> <p>Пример – два такта для частоты Fclk = 33,3 МГц. Для корректной работы схемы значение должно быть больше 0</p>	RW	

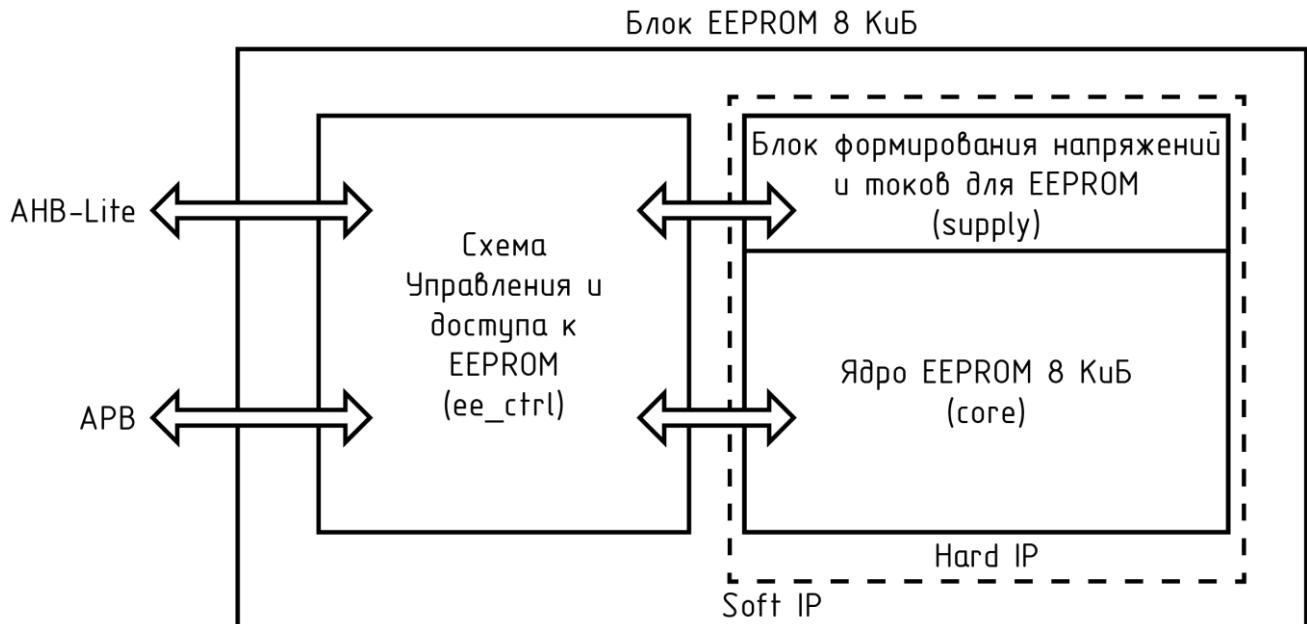
Таблица 124 – Регистр подстройки длительности процедуры записи 2 OTPWT2

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31:24		Зарезервировано		
N_W	23:0	1666667	<p>Длительность высокого уровня сигнала we i (вход Hard IP) в тактах. Рекомендуемое значение</p> <p>N_W = 50000000 нс/Pclk, где Pclk – период тактового сигнала в нс.</p> <p>Пример – 1666667 тактов для частоты Fclk = 33,3 МГц</p>	RW	

3.18 Контроллер EEPROM

Данный контроллер предназначен для осуществления доступа к массиву блока EEPROM на запись и чтение со стороны интерфейсов AHB-Lite (только чтение) и APB (чтение и запись в несколько этапов). Данный контроллер используется совместно с Hard IP EEPROM объемом 8 КиБ и требуемым для него блоком формирования напряжений и токов (supply).

Структурная схема контроллера EEPROM представлена на рисунке 83.



Конфигурация совместимого блока EEPROM core:

- информационная емкость – 2 тыс. слов;
- количество бит в слове – 38 (32 бита данных пользователя и 6 бит для коррекции одиночных ошибок);
- разрядность входа – 32 + 6 бит;
- разрядность выхода – 32 + 6 бит;
- количество строк в массиве – 64;
- количество столбцов в массиве – 1216.

Особенности контроллера:

- запись через APB-интерфейс в несколько этапов;
- чтение через APB-интерфейс в 2 или 3 этапа;
- чтение через AHB-Lite-интерфейс напрямую из массива (с задержкой, требуемой на процедуру чтения);

- схема коррекции одиночных ошибок с функцией поднятия флага при одиночной ошибке;
- возможность прямого доступа к битам коррекции на запись и чтение;
- возможность задания количества тактов для операций записи и чтения через регистры управления.

3.18.1 Режимы чтения по интерфейсу APB

Контроллер позволяет выполнять чтение по интерфейсу APB в два или три этапа:

- запись адреса (EEA) – чтение регистра данных (EEDAT);
- запись адреса (EEA) – опрос флага готовности (EESTA.BSY) – чтение регистра данных (EEDAT).

Если используется двухэтапное чтение, то вместо процедуры проверки флага готовности данных вводятся такты ожидания интерфейса APB, в течение которых сигнал PREADY удерживается в «0». Если используется трехэтапное чтение, PREADY удерживается в «1», но требуется дополнительная проверка флага готовности (EESTA.BSY). Для переключения между режимами используется регистр EECON.APBNWS. При EECON.APBNWS = 1 используется режим чтения в три этапа (без тактов ожидания APB).

При EECON.APBNWS = 0, при чтении регистра EEDAT происходит автоматическое инкрементирование адреса EEA. Таким образом, в этом режиме запись адреса не требуется, если необходимо считать слово, имеющее адрес на один больше предыдущего.

При EECON.APBNWS = 0 контроллер осуществляет автоматическое считывание слова из EEPROM при чтении регистра EEDAT.

При EECON.APBNWS = 1 контроллер осуществляет автоматическое считывание слова из EEPROM при записи в регистр адреса EEA.

Для выполнения чтения бит коррекции через APB обращается к регистру EERB. Когда необходимо выполнять чтение и бит коррекции, и данных, регистр бит коррекции каждого словачитываются в первую очередь (т.е. сначала производится чтение регистра бит коррекции EERB, далее вычитываются четыре байта данных через регистр EEDAT). Такая последовательность должна быть соблюдена, т.к. при чтении регистра EEDAT происходит автоматическое инкрементирование адреса EEA.

Чтобы считывать фактическое значение, хранимое в EEPROM, а не скорректированное, перед чтением необходимо отключить схему коррекции ошибок, записав «1» в бит EECON.DISECC.

3.18.2 Чтение данных через AHB-Lite-интерфейс

В данном блоке реализована операция чтения данных через AHB-Lite-интерфейс.

Одновременно возможно работать только с одним интерфейсом, либо с AHB-Lite, либо с APB. Следовательно, на момент начала передачи по AHB-Lite блок должен находиться в режиме ожидания поступления новых команд (EESTA.BSY = 0).

3.18.3 Процедура заполнения буфера

Особенностью Hard IP EEPROM является необходимость заполнения буфера записи перед операцией стирания и программирования. В ходе операции программирования содержимое буфера записи переносится в строку массива (или несколько строк при EECON.WRBEH = 0b00). В ходе операции стирания производится стирание тех слов, которые заполнялись произвольными значениями в ходе заполнения буфера. Размер буфера записи – 32 слова данных. При заполнении буфера записи адрес слова внутри буфера определяется разрядами EEA[6:2]. При заполнении буфера записи разряды EEA[12:7] должны оставаться неизменными (используются как разряды, управляющие дешифратором строк).

Перед заполнением буфера записи требуется установить бит разрешения записи в буфер EECON.BWE = 1. После процедуры стирания/программирования данный бит будет автоматически очищен.

Перед заполнением буфера записи требуется установить значение бит WRBEH[1:0], определяющих тип последующей операции (операция с четными строками, нечетными строками, всеми строками).

Процедура заполнения буфера записи может выполняться путем многократного повторения 3-х процедур до тех пор, пока буфер не будет заполнен:

- запись адреса слова в EEA;
- запись произвольных данных (стирание) или записываемых данных (программирование) в EEDAT;
- опрос флага готовности EESTA.BSY, пока он не станет равным 0 (если EECON.APBNWS = 1).

Опрос флага готовности можно не производить, если установить EECON.APBNWS = 0 (по умолчанию). В этом случае, пока процедура записи в буфер не завершится, на шине APB будут сформированы такты ожидания, во время которых PREADY = 0. Если EECON.APBNWS = 1, то PREADY будет всё время удерживаться в 1, но между операциями записи в буфер необходимо выполнять опрос флага EESTA.BSY.

Процедура заполнения буфера может быть ускорена за счет автоматической инкрементации адреса после записи в EEDAT. После каждой записи в EEDAT внутренний адрес слова для записи автоматически инкрементируется. Если необходимо запрограммировать всю строку массива, рекомендуется выполнить только запись адреса первого слова в EEA, а затем последовательно записывать 32 слова данных в EEDAT, не записывая адреса слов между операциями записи в EEDAT.

3.18.4 Стирание слов

Для стирания слов в странице/части страниц/всех страницах рекомендуется следовать описанной ниже последовательности действий:

- заполнить буфер записи;
- выполнить запись в регистр EECON:
 - 1) EECON.EX = 1 (запуск процедуры);
 - 2) EECON.OP = 0b01 (стирание);
 - 3) EECON.WRBEH = 0b00 (обычное) или 0b01 (только четные страницы) или 0b10 (только нечетные страницы) или 0b11 (все страницы);
- опрашивать флаг EESTA.BSY до тех пор, пока он не будет очищен.

3.18.5 Программирование слов

Для программирования слов в странице/части страниц/всех страницах рекомендуется следовать описанной ниже последовательности действий:

- заполнить буфер записи;
- выполнить запись в регистр EECON:
 - 1) EECON.EX = 1 (запуск процедуры);
 - 2) EECON.OP = 0b10 (программирование);
 - 3) EECON.WRBEH = 0b00 (обычное) или 0b01 (только нечетные страницы) или 0b10 (только четные страницы) или 0b11 (все страницы);

- опрашивать флаг EESTA.BSY до тех пор, пока он не будет очищен.

3.18.6 Прерывание при одиночной ошибке

Если включена схема коррекции ошибок (EECON.DISECC = 0) и производится чтение по APB или AHB-Lite-интерфейсу, то при обнаружении одиночной ошибки схемой коррекции ошибок будет поднят флаг EESTA.SERR.

При обнаружении одиночной ошибки в считанном слове может быть сформирован запрос прерывания. Для включения данного запроса прерывания необходимо установить бит IESERR = 1.

Флаг EESTA.SERR и, одновременно, запрос прерывания, могут быть очищены путем записи «0» в бит EESTA.SERR.

3.18.7 Временные ограничения

Чтобы формы сигналов управления, поступающих из Soft IP в Hard IP, соответствовали требованиям к временным параметрам данных сигналов при любых значениях частоты тактового сигнала clk, возможно изменение задержек между некоторыми событиями на временных диаграммах. Формы сигналов при различных операциях показаны на рисунках 84 – 86.

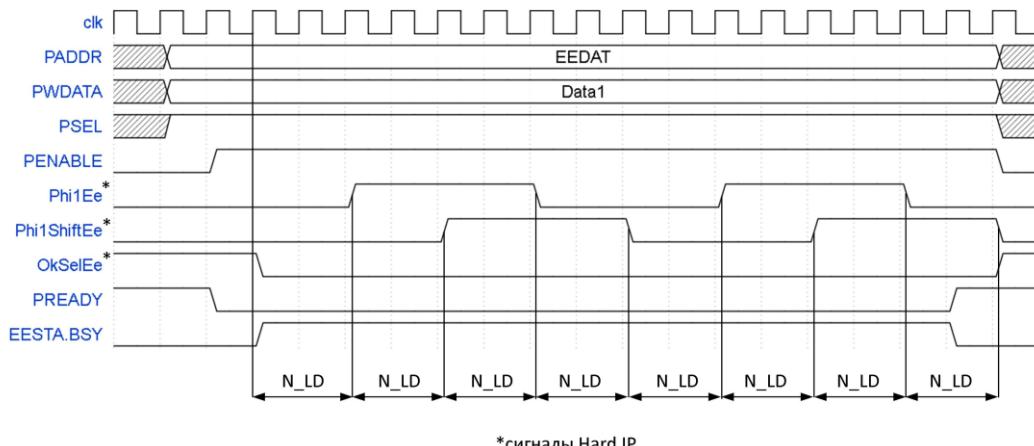


Рисунок 84 – Формы сигналов в процедуре заполнения буфера записи

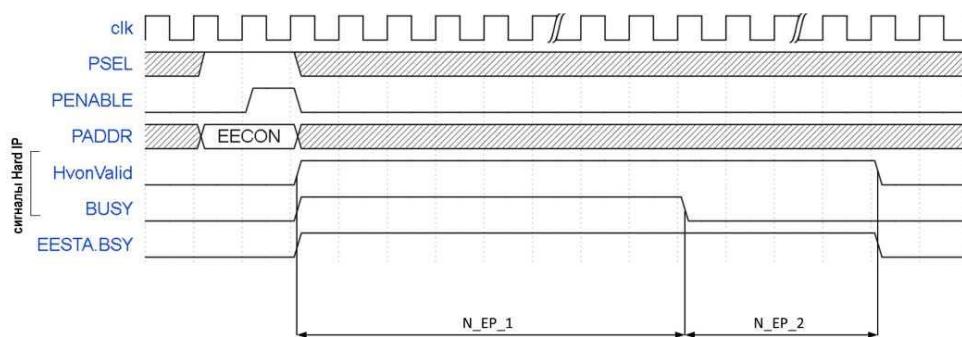


Рисунок 85 – Формы сигналов в процедурах стирания и программирования

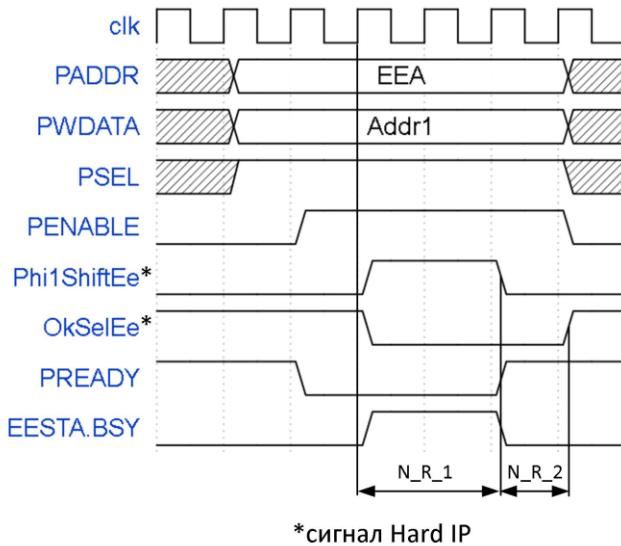


Рисунок 86 – Формы сигналов в процедуре чтения

На рисунках представлены следующие временные интервалы:

- N_{LD} – количество тактов между передними и задними фронтами сигналов Hard IP $\Phi 1Ee$ и $\Phi 1ShiftEe$ и задержка от начала обращения по шине до первого фронта $\Phi 1Ee$. Рекомендуемое значение при любых частотах clk 1 такт;
- N_{EP_1} – длительность высокого уровня сигнала BUSY (вход Hard IP) в тактах. Рекомендуемое значение $N_{EP_1} = 2000000$ нс / $Pclk$, где $Pclk$ – период тактового сигнала в нс. Пример – 66667 тактов для частоты $Fclk = 33,3$ МГц;
- N_{EP_2} – задержка между задними фронтами BUSY и $HvonValid$ (входы Hard IP) в тактах. Рекомендуемое значение $N_{EP_2} = 15000$ нс / $Pclk$, где $Pclk$ – период тактового сигнала в нс. Пример – 500 тактов для частоты $Fclk = 33,3$ МГц;
- N_{R_1} – количество тактов, в течение которого $\Phi 1ShiftEe$ имеет высокий уровень в процедуре чтения. Рекомендуемое значение $N_{R_1} = \text{ceil}(51/Pclk)$, где $Pclk$ – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа;
- N_{R_2} – количество тактов, требуемое для завершения процедуры чтения (перевода $OkSelEe$ на высокий уровень после процедуры чтения). Рекомендуемое значение при любых частотах clk – 1 такт;
- при доступе по интерфейсу АНВ-Lite формы сигналов имеют аналогичный вид и также зависят от показанных параметров. При записи значения адреса в EEA,

если EECON.APBNWS = 1, формы сигналов также имеют аналогичный вид и также зависят от показанных параметров.

При работе с блоком обязательно должны быть учтены следующие временные ограничения:

- время от момента подачи питания или перевода из «1» в «0» сигнала BadPowerEe до момента любого обращения к Hard IP должно составлять не менее 100 мкс;
- время от момента перевода сигнала StopEe из «1» в «0» до момента любого обращения к Hard IP должно составлять не менее 500 нс.

3.18.8 Регистры

Список регистров и их назначение представлено в таблице 125.

Таблица 125 – Регистры контроллера EEPROM

Обозначение	Смещение	Начальное значение	Описание	Доступ
EEDAT	0x00	-	Регистр чтения/записи данных	RW
EEA	0x04	-	Регистр адреса страницы	RW
EECON	0x08	0x000000	Регистр управления	RW
EESTA	0x0C	0x000000	Регистр статуса	RW
EERB	0x10	0x000007	Регистр бит коррекции прочитанного слова	RO
EEADJ	0x14	0x329000	Регистр настроек	RW
NCYCRL	0x18	0x010201	Регистр подстройки длительности процедур чтения и заполнения буфера записи	RW
NCYCEP1	0x1C	0x01046B	Регистр 1 подстройки длительности процедур стирания и программирования	RW
NCYCEP2	0x20	0x0001F4	Регистр 2 подстройки длительности процедур стирания и программирования	RW

В таблицах 126 – 134 представлены описания регистров модуля.

Таблица 126 – Регистр чтения/записи данных EEDAT

Обозначение	Разряды	Начальное значение	Назначение	Доступ
DATA	31:0	-	В процессе чтения/записи данного регистра производится автоматическое инкрементирование адреса (EEA).	RW

Таблица 127 – Регистр адреса страницы EEA

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31:13		Зарезервировано		
ADDR	12:2	-	Адрес слова для выполнения стирания/программирования/чтения. Если необходимо выполнить стирание или программирование, данный адрес должен использоваться в процедуре заполнения буфера записи	RW	
Зарезервировано	1:0		Зарезервировано		

Таблица 128 – Регистр управления EECON

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31:9		Зарезервировано		
IESERaR	8	0x0	Разрешение запроса прерывания при поднятии флага SERR: 0 – запрос не поступает; 1 – запрос поступает	RW	
BWE	7	0x0	Разрешение записи в буфер. Данный бит следует устанавливать в «1» перед операцией заполнения буфера записи перед операциями стирания и программирования. После операций стирания и программирования данный бит автоматически очищается (то есть для последующей записи в буфер следует повторно выполнить запись «1» в этот бит)	RW	
DISECC	6	0x0	Отключение схемы коррекции ошибок во время выполнения запрошенной процедуры. Отключение производится записью «1» в этот бит. Если схема коррекции отключена, то при записи значение бит коррекции определяется шестью младшими битами слова. Если схема коррекции включена, то при записи значение бит коррекции вычисляется схемой SEC32 - ENC. Если схема коррекции отключена, то при чтении не производится коррекция возможных ошибок в слове. Если схема коррекции включена, то при чтении данные корректируются схемой DEC32_ENC	RW	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
APBNWS	5	0x0	Отключение вставки тактов ожидания в процессе обмена по APB при заполнении буфера записи (во время записи в EEDAT) и при записи адреса для процедуры чтения данных (EEA). Если такты ожидания отключены (APBNWS=1), то требуется производить опрос флага EESTA.BSY после этих операций до тех пор, пока EESTA.BSY не станет равным 0	RW
WRBEH	4:3	0x0	Поведение операции стирания/программирования: 0b00 – обычное стирание / программирование одной страницы; 0b01 – стирание / программирование всех четных страниц; 0b10 – стирание / программирование всех нечетных страниц; 0b11 – стирание / программирование всех страниц. Данные биты должны быть установлены перед операцией заполнения буфера записи перед операциями стирания и программирования	RW
OP	2:1	0x0	Выбор процедуры: 0b00 – чтение страницы; 0b01 – стирание; 0b10 – программирование; 0b11 – значение не используется	RW
EX	0	0x0	Запуск процедуры. Запись 1 в данный бит инициирует выполнение процедуры, процедура определяется битами OP	RW

Таблица 129 – Регистр статуса EESTA

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:2	Зарезервировано		
SERR	1	0x0	Индикатор одиночной (исправимой) ошибки. Аппаратно может быть только поднят, т.е. если после слова с ошибкой считано слово без ошибки, флаг не очистится. Может быть очищен программно путем записи 0	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
BSY	0	0x0	Блок занят (выполняется запрошенная операция)	RO

Таблица 130 – Регистр бит коррекции прочитанного слова EERB

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31:6	Зарезервировано			
CORRECT	5:0	0x7	Содержит 6 бит коррекции последнего считанного слова	RO	

Таблица 131 – Регистр настроек блоков Hard IP EEADJ

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31:27	Зарезервировано			
*IddqEn	26	0x0	Перевод Hard IP в состояние бездействия (quiescent) для Iddq тестирования	RW	
StopEe	25	0x0	Перевод Hard IP в режим пониженного энергопотребления	RW	
*StrictCtl	24:22	0x0	Точная подстройка тока, используемого при чтении	RW	
*VppCtl	21:20	0x3	Подстройка времени нарастания высокого напряжения	RW	
*CGSTRICT	19	0x0	Включение режима точной подстройки напряжения, используемого при чтении	RW	
*VcgCtl	18:16	0x2	Подстройка напряжения, подаваемого на управляющие затворы при чтении	RW	
*RefCtl	15:13	0x4	Подстройка тока, используемого при чтении	RW	
*VbgCtl	12:10	0x4	Подстройка опорного напряжения bandgap	RW	
*DummyCtl	9:8	0x0	Подстройка момента захвата выдаваемых данных	RW	
*MinusImix	7	0x0	Сигнал управления блоком мониторинга аналоговых сигналов	RW	

Обозначение	Разряды	Начальное значение	Назначение	Доступ
*VboostCtl	6:5	0x0	Старший разряд – подстройка внутреннего сигнала Vboost, младший разряд – сигнал управления блоком мониторинга аналоговых сигналов	RW
*PlusImix	4	0x0	Сигнал управления блоком мониторинга аналоговых сигналов	RW
*HivCtl	3:1	0x0	Подстройка уровня внутреннего сигнала HIV	RW
*OSCX2	0	0x0	Увеличение частоты тактового сигнала, поступающего на устройство с зарядовой накачкой, в два раза	RW
Примечание – указанные «*» поля предназначены для тонкой настройки блока и выполнения процедуры тестирования. Не гарантируется работа блока при использовании не рекомендованных производителем значений.				

Таблица 132 – Регистр подстройки длительности процедур чтения и заполнения буфера записи NCYCRL

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:24		Зарезервировано	
N_R_2	23:16	0x1	Количество тактов, требуемое для завершения процедуры чтения (перевода OkSelEe на высокий уровень после процедуры чтения). Рекомендуемое значение при любых частотах clk 1 такт. Запрещено устанавливать равным 0	RW
N_R_1	15:8	0x2	Количество тактов, в течение которого Phi1ShiftEe имеет высокий уровень в процедуре чтения. Рекомендуемое значение N RA = ceil(51/Pclk), где Pclk – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа. Запрещено устанавливать равным 0	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
N_LD	7:0	0x1	Количество тактов между передними и задними фронтами сигналов Hard IP Phi1Ee и Phi1ShiftEe и задержка от начала обращения по шине до первого фронта Phi1Ee. Рекомендуемое значение при любых частотах clk 1 такт. Запрещено устанавливать равным 0	RW

Таблица 133 – Регистр 1 подстройки длительности процедур стирания и программирования NCYCEP1

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:18		Зарезервировано	
N_EP_1	17:0	0x1046B	Длительность высокого уровня сигнала BUSY (вход Hard IP) в тактах. Рекомендуемое значение N RA = 2000000 нс / Pclk, где Pclk – период тактового сигнала в нс. Пример – 66667 тактов для частоты Fclk = 33,3 МГц. Запрещено устанавливать равным 0	RW

Таблица 134 – Регистр 2 подстройки длительности процедур стирания и программирования N CYCER2

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:11		Зарезервировано	
N_EP_2	10:0	0x1F4	Задержка между задними фронтами BUSY и H von Valid (входы Hard IP) в тактах. Рекомендуемое значение N RA = 15000 нс / Pclk, где Pclk – период тактового сигнала в нс. Пример – 500 тактов для частоты Fclk = 33,3 МГц. Запрещено устанавливать равным 0	RW

3.19 Модуль сторожевого таймера (WDT)

Модуль сторожевого таймера (WDT) реализует функцию контроля «зависания» исполняемой программы. Сторожевой таймер должен быть проинициализирован и запущен в начале работы программы. После чего он самостоятельно отсчитывает заданное время, и если в течение этого времени программа не перезагрузит его, то таймер вырабатывает сигнал сброса системы. За 256 тактов до окончания заданного времени таймер дополнительно выдает сигнал прерывания.

Примечание – Сигнал сброса системы, вызванный сторожевым таймером, не подается на вывод RST#.

Сторожевой таймер тактируется независимым внутренним сигналом, благодаря чему он продолжает работу, даже если пропадает системный тактовый сигнал. Таймер состоит из 12-разрядного счетчика тактов с 12-ти разрядным предварительным делителем. Таким образом, максимально возможный интервал времени составляет 2^{24} тактов.

В модуле реализована защита от случайного изменения содержимого регистров со стороны программы.

3.19.1 Регистры сторожевого таймера

В таблице 135 перечислены регистры, доступные по интерфейсу APB.

Таблица 135 – Регистры модуля сторожевого таймера

Обозначение	Смещение	Начальное значение	Назначение	Доступ
KEY	0x9C	–	Регистр ключа	WO
CON	0x84	0x0	Конфигурация	RW
STA	0x9C	–	Регистр состояния	RO

Для записи конфигурации модуля необходимо выполнить процедуру разблокировки. Для этого в регистр KEY нужно записать значение 0x1E. Блокировка активируется через 15 тактов или при выполнении записи в какой-либо регистр.

Для запуска или перезагрузки значения сторожевого таймера в регистр KEY необходимо записать значение 0x71. Для остановки сторожевого таймера в регистр KEY необходимо записать значение 0xE8.

Описание полей регистра конфигурации сторожевого таймера приведено в таблице 136. Обновление регистра возможно только в остановленном состоянии сторожевого таймера.

Таблица 136 – Поля регистра конфигурации сторожевого таймера CON

Обозначение	Разряды	Начальное значение	Назначение	Доступ
PRESCALE	14:12	0x0	Делитель входной частоты (Fclk) для таймера: 0b000 – Fclk / 1; 0b001 – Fclk / 2; 0b010 – Fclk / 4; 0b011 – Fclk / 16; 0b100 – Fclk / 64; 0b101 – Fclk / 256; 0b110 – Fclk / 1024; 0b111 – Fclk / 4096	RW
PRELOAD	11:00	0x0	Начальное значение таймера при запуске или перезапуске (таймер считает в сторону увеличения значений)	RW

Описание полей регистра состояния сторожевого таймера приведено в таблице 137.

Бит активности таймера изменяет свое значение через некоторое время после обращения к регистру KEY, необходимое для проведения синхронизации сигналов.

Таблица 137 – Поля регистра состояния сторожевого таймера STA

Обозначение	Разряды	Начальное значение	Назначение	Доступ
WDT_RST_FLAG	8	0x0	Флаг генерации сброса сторожевым таймером. Сбрасывается в 0 только при снятии и последующей подаче питания	RO
–	7:2	Зарезервировано		
TIMERLOADING	1	0x0	Бит перезагрузки значения: 1 – выполняется перезагрузка значения в таймере; 0 – перезагрузка значения таймера в данный момент не выполняется (завершена)	RO
TIMERENABLED	0	0x0	Бит активности таймера: 1 – таймер запущен; 0 – таймер остановлен	RO

3.20 Сторожевой таймер шины (WDT BUS)

Сторожевой таймер шины предотвращает блокировку системы ведомым устройством АHB. Модуль состоит из регистров управления и трех мониторов шины, подключенных к шинам EEPROM, SPIFI, шине периферийных устройств: ускоритель криптографии, модуль расчета контрольных сумм, устройства на шине APB_P. Мониторы размещаются между матрицей шины и ведомым устройством АHB.

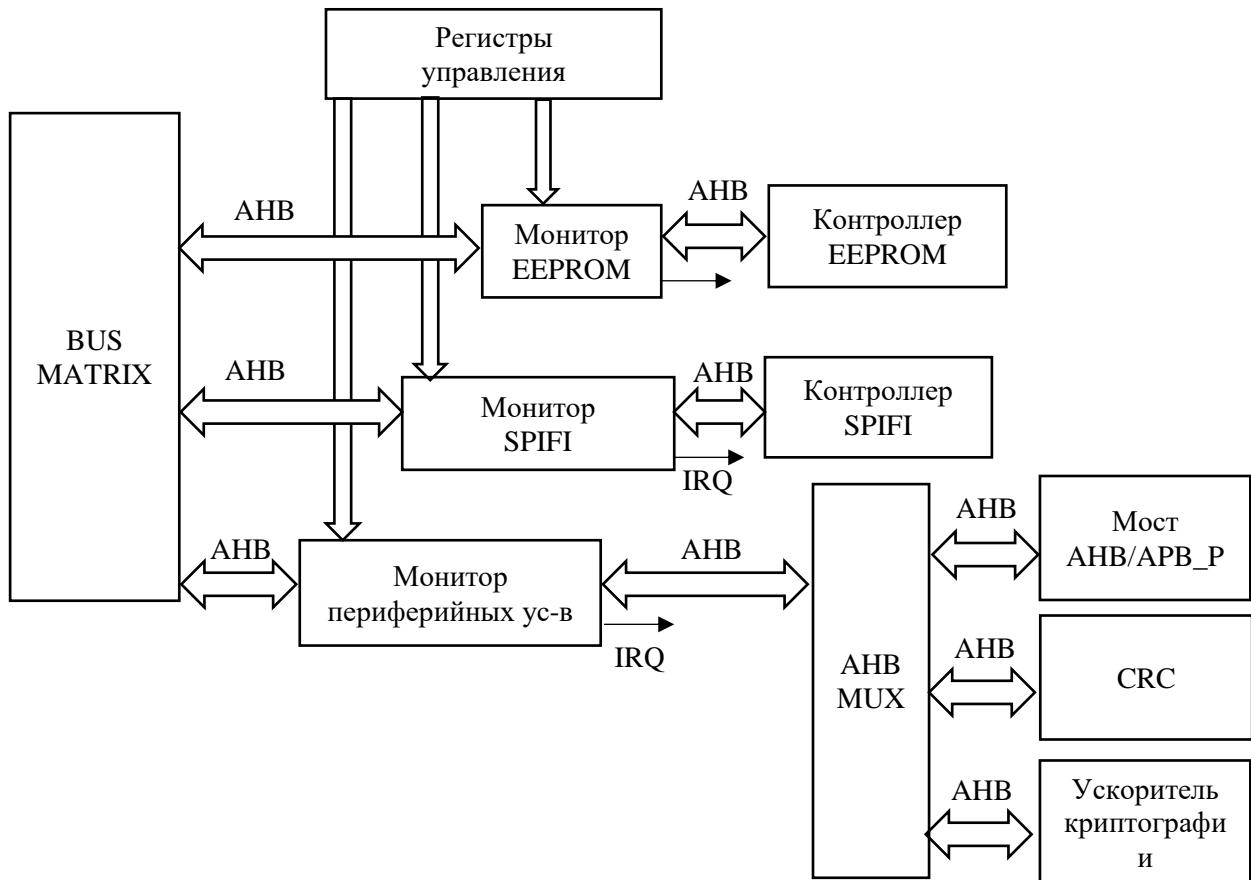


Рисунок 87 – Функциональная схема сторожевого таймера шины

Если есть активная передача на ведомое устройство, и ведомое устройство удерживает сигнал готовности данных HREADY в «0» более определенного количества тактов (указанных в поле TIMEOUT), сторожевой таймер выполняет следующие действия:

- формирует для ведущего признак ошибки на шине (HRESP = «1»);
- формирует прерывание (если для указанного ведомого оно разрешено);
- формирует сигнал сброса ведомого.

Если мастер шины формирует последующий запрос к этому ведомому устройству, монитор формирует на шине ошибку и не передает данный запрос ведомому, таким образом фактически блокирует доступ к ведомому. Монитор прекращает формировать сообщения об ошибках и блокировать доступ к ведомому, когда ведомое завершает передачу, устанавливая HREADY в «1». Если идет пакетная передача, блокируются оставшиеся транзакции в пакетной передаче.

При удержании ведомым шины более 2^{TIMEOUT} тактов формирует прерывания, подключённые к следующим линиям контроллера прерываний:

- IRQ[17] - Сторожевой таймер шины (периферийные устройства);

- IRQ[18] - Сторожевой таймер шины (SPIFI);
- IRQ[19] - Сторожевой таймер шины (EEPROM).
- сброс прерывания осуществляется записью 1 в соответствующий бит регистра IRQ_CLEAR.

Регистры сторожевого таймера шины представлены в таблице 138.

Таблица 138 – Регистры управления сторожевого таймера шины

Обозначение	Смещение	Начальное значение	Назначение	Доступ
TIMEOUT	0x00	0x0	Определяет количество циклов ожидания до формирования ошибки. Значения от 0 до 15. Количество циклов ожидания вычисляется как 2TIMEOUT	RW
IRQ_CLEAR	0x04	–	Сброс прерываний: IRQ_CLEAR[0] = 1 сброс прерываний для монитора шины периферийных устройств IRQ_CLEAR[1] = 1 сброс прерываний для монитора шины SPIFI IRQ_CLEAR[2] = 1 сброс прерываний для монитора шины EEPROM	W1C
ENABLE	0x08	0x0	Запуск/отключение мониторов шины. 1 – включен, 0 выключен ENABLE [0] – управление монитором шины периферийных устройств ENABLE [1] – управление монитором шины SPIFI ENABLE [0] – управление монитором шины EEPROM	RW

3.21 Температурный сенсор (TSENS)

Сенсор преобразует электрические величины, прямо пропорционально зависящие от температуры кристалла, в двоичное представление этой температуры.

Технические характеристики:

- Точность измерения температуры $\pm 0,1$ °C;
- Погрешность измерения температуры в диапазоне [−40 °C; 125 °C] не превышает 1 %;
- Наличие спящего режима с пониженным потреблением тока;
- Функция подстройки для уменьшения влияния разброса техпроцесса;
- Тактовая частота 32 кГц-100 кГц.

3.21.1 Состав и принцип работы сенсора

Структурная схема представлена на рисунке 88. Сенсор состоит из блока формирования опорного тока, блока формирования двух напряжений база-эмиттер, сигма-дельта АЦП и регистра накопления результатов АЦП.

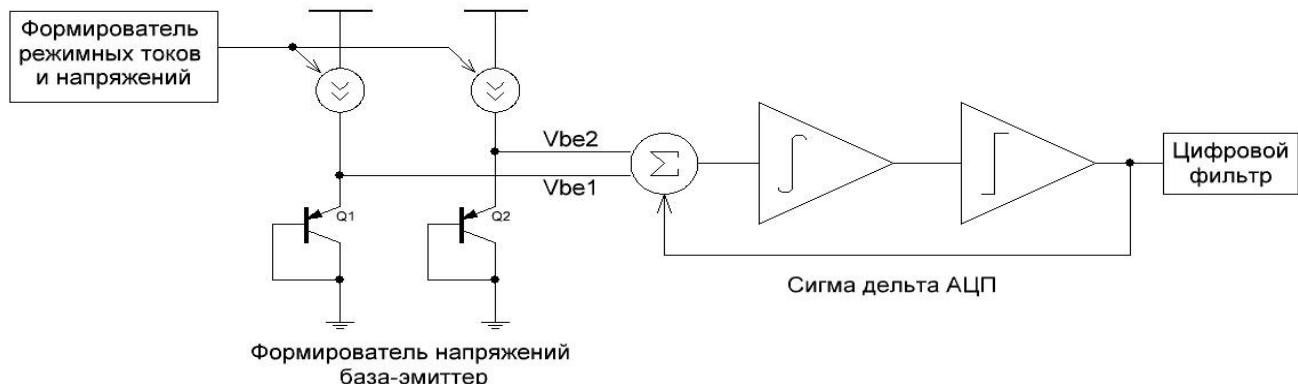


Рисунок 88 – Структурная схема сенсора

Блок сенсора основан на биполярных транзисторах, так как их технологический разброс параметров мал по сравнению с альтернативами. Для генерации данных о температуре достаточно всего двух транзисторов в диодном включении, управляемых разными токами. Напряжение база-эмиттер Vbe обратно пропорционально температуре (СТАТ), а разница напряжений ΔVbe прямо пропорциональна (РТАТ).

Формулы пересчета кода термосенсора q в градусы Цельсия:

$$x = (4096 - q) / q \quad (12)$$

$$\mu x = \alpha / (\alpha + x) \quad (13)$$

$$Temp = A * \mu x + B \quad (14)$$

Типовые значения:

- $A = 622$;
- $B = -273,15$;
- $\alpha = 10,3$.

3.21.2 Регистры управления

Для управления температурным сенсором используются регистры блока управления аналоговой подсистемой. Состав регистров управления сенсором представлен в таблице 139. В таблицах 140–146 представлены описания полей регистров сенсора.

Таблица 139 – Регистры управления температурным сенсором

Обозначение	Смещение	Начальное значение	Назначение	Доступ
TSENS_CFG	0x1C	0x0	Регистр настроек	RW
TSENS_THRESHOLD	0x20	0x3865B	Регистр пороговых значений	RW
TSENS_IRQ	0x24	0x0	Регистр прерываний	RW
TSENS_CLEAR_IRQ	0x28	0x0	Регистр сброса прерываний	W1C
TSENS_VALUE	0x2C	0x0	Регистр данных	RO
TSENS_SINGLE	0x30	0x0	Регистр запуска однократного измерения	WO
TSENS_CONTINUOUS	0x34	0x0	Регистр запуска непрерывного измерения	WO

Примечание – Смещение относительно стартового адреса блока управления аналоговой подсистемой (0x00085000 в адресном пространстве микроконтроллера).

Таблица 140 – Регистр TSENS_CFG температурного сенсора

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:16		Зарезервировано	
DIV	15:6	0x0	Значение делителя тактового сигнала. Частота сенсора определяется как TSENS = FIN/(2*(DIV + 1))	RW
CLK_MUX	5:3	0x0	Выбор источника тактирования сенсора (FIN): 0x0 – системная частота (sys_clk); 0x1 – частота шины АHB (hclk); 0x2 – частота внешнего осциллятора 32 МГц; 0x3 – частота HSI32M; 0x4 – частота внешнего осциллятора 32 кГц; 0x5 – частота LSI32K	RW
NRESET	2	0x0	Управление сбросом сенсора, активный уровень «0»	RW
NPD_CLK	1	0x0	Управление тактированием сенсора: 0 – тактирование сенсора выключено; 1 – тактирование сенсора включено	RW
NPD	0	0x0	Управление питанием сенсора: 0 – сенсор выключен; 1 – сенсор включен	RW

Таблица 141 – Регистр TSENS_THRESHOLD температурного сенсора

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:20		Зарезервировано	
THRESHOLD_LOW	19:10	225	При значении выхода сенсора менее LOW_THRESHOLD будет формироваться прерывание. Значение по умолчанию соответствует температуре -40 °C	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
THRESHOLD_HI	9:0	603	При значении выхода сенсора больше HI_THRESHOLD будет формироваться прерывание. Значение по умолчанию соответствует температуре 125 °C	RW

Таблица 142 – Регистр TSENS_IRQ температурного сенсора

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:6		Зарезервировано	
LOW_IRQ	5	0x0	Статус прерывания события «выход сенсора меньше порогового значения LOW_THRESHOLD»	RO
HI_IRQ	4	0x0	Статус прерывания события «выход сенсора больше порогового значения HI_THRESHOLD»	RO
EOC_IRQ	3	0x0	Статус прерывания по окончанию преобразования	RO
LOW_MASK	2	0x0	Маска прерывания при событии «выход сенсора меньше порогового значения LOW_THRESHOLD»	RW
HI_MASK	1	0x0	Маска прерывания при событии «выход сенсора больше порогового значения HI_THRESHOLD»	RW
EOC_MASK	0	0x0	Маска прерывания по окончанию преобразования	RW

Таблица 143 – Регистр TSENS_CLEAR_IRQ температурного сенсора

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:3		Зарезервировано	
LOW_CLEAR	2	–	Сброс прерывания события «выход сенсора меньше порогового значения LOW_THRESHOLD»	W1C

Обозначение	Разряды	Начальное значение	Назначение	Доступ
HI_CLEAR	1	—	Сброс прерывания события «выход сенсора больше порогового значения HI_THRESHOLD»	W1C
EOC_CLEAR	0	—	Сброс прерывания по окончанию преобразования	W1C

Таблица 144 – Регистр TSENS_VALUE температурного сенсора

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:11		Зарезервировано	
EOC	10	0x0	Текущее значение выхода окончания преобразования	RO
VALUE	9:0	0x0	Последнее измеренное значение сенсором	RO

Таблица 145 – Регистр TSENS_SINGLE температурного сенсора

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:1		Зарезервировано	
SINGLE	0	—	Запуск одного измерения	W1S

Таблица 146 – Регистр TSENS_CONTINUOUS температурного сенсора

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:1		Зарезервировано	
CONTINUUS	0	0	Запуск / остановка непрерывного измерения температуры	RW

3.22 Цифро-аналоговый преобразователь (DAC)

Цифро-аналоговый преобразователь (ЦАП) предназначен для формирования напряжения пропорционального 12-битному управляющему коду.

Примечание - В исполнении К1948ВК01Б8 и К1948ВК01Б5 блок ЦАП не используется.

Параметры преобразователя представлены в таблице 147.

Таблица 147 – Параметры ЦАП

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра		
		не менее	номинал	не более
Число разрядов ЦАП	–	–	12	–
Дифференциальная нелинейность, Lsb	–	–	0,36	–
Интегральная нелинейность, Lsb	–	–	0,74	0,83
Входное напряжение источника опорного напряжения, В	UREF	–	1,2	–
Ток по входу опорного напряжения, мкА	IUREF	–	0	–
Диапазон изменения напряжения выходного сигнала ЦАП, В	dUDAC	–	0 – UREF	–
Сопротивление нагрузки на выводе выходного сигнала ЦАП, кОм	Rload	–	50	–
Емкость нагрузки на выводе выходного сигнала ЦАП, пФ	Cload	–	2	–
Время установления выходного напряжения, мкс	TUвых	–	–	0,7
Внешний опорный ток, мкА	IOUT	8	10	12
Ток потребления, мА	ICC	–	0,9	1,07

3.22.1 Состав и принцип работы резистивного ЦАП

Модуль состоит из резистивного ЦАП, входного буфера/делителя внешнего опорного напряжения на 2, внутреннего рациометрического источника опорного напряжения, выходного буфера с коэффициентом усиления 2, а также дополнительных резисторов, ключей и цифровой управляющей логики. Схема с делением опорного напряжения пополам и последующего увеличения напряжения в 2 раза выбрана для

того, чтобы использовать только «тонкие» N-транзисторы в ключах, которые обеспечивают требуемое сопротивление в диапазоне напряжений от 0 до 0.6В при небольших размерах ключей, что позволяет минимизировать скачки напряжения при отсутствии буферизации между каскадами резистивной матрицы. Структурная схема представлена на рисунке 89.

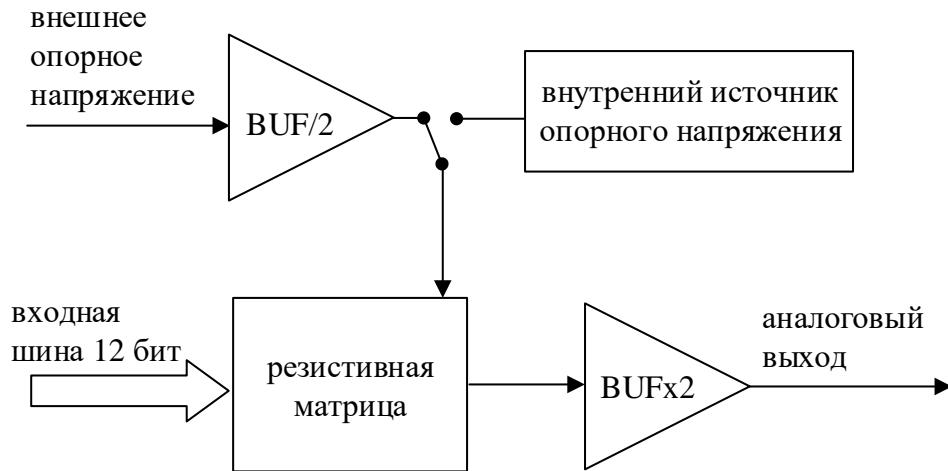


Рисунок 89 – Схема ЦАП

Резистивный ЦАП состоит из двух каскадов, каждый из которых обеспечивает разрешение шесть бит. Каждый единичный элемент каскада состоит одного резистора. Топологически эти элементы расположены симметрично относительно центра (common centroid) для уменьшения разброса параметров. Выходы первого каскада не буферизируются, поэтому для уменьшения систематической ошибки единичные элементы второго каскада имеют значительно большее сопротивление (1625 Ом по сравнению с 200 Ом).

Выходной буфер представляет собой неинвертирующий операционный усилитель (ОУ) с коэффициентом усиления два. В остальных случаях – неинвертирующий ОУ с коэффициентом усиления один. Типичный ток потребления ОУ составляет 200мкА (не учитывая выходной ток). В выключенном состоянии типичный ток потребления ОУ составляет 30нА.

3.22.1 Регистры управления

Для управления ЦАП используются регистры блока управления аналоговой подсистемой. Состав регистров управления ЦАП представлен в таблице 147.

В таблицах 149, 150 представлены описания полей регистров ЦАП.

Таблица 148 – Регистры управления ЦАП

Обозначение	Смещение	Начальное значение	Назначение	Доступ
DAC0_CFG	0x00	0x0	Регистр настройки ЦАП0	RW
DAC0_VALUE	0x04	0x0	Входные данные для ЦАП0	WO
DAC1_CFG	0x08	0x0	Регистр настройки ЦАП1	RW
DAC1_VALUE	0x0C	0x0	Входные данные для ЦАП1	WO
REF_CLB	0x38	0x188	Управление калибруемыми источниками тока и напряжения	RW

Примечание – Смещение относительно стартового адреса блока управления аналоговой подсистемой (0x00085000 в адресном пространстве микроконтроллера).

Таблица 149 – Регистры DAC0_CFG, DAC1_CFG

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:14		Зарезервировано	
EMPTY_READ	13	0x0	1 - значение, хранящееся в DAC_VALUE, было сдвинуто в ЦАП, возможна запись следующего значения; 0 - в регистре DACX_VALUE находится необработанное значение; *Автоматически сбрасывается при записи в DACx_VALUE	RO
Зарезервировано	12		Зарезервировано	
EXTPAD	11	0x0	Выбор источника внешнего опорного напряжения: 1 - внешний вывод; 0 - настраиваемый ИОН	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
EXTEN	10	0x0	Выбор источника опорного напряжения: 1 - внешний; 0 - встроенный	RW
DIV	9:2	0x0	Значение делителя тактового сигнала. Частота определяется как $F_{ЦАП}=F_{IN}/(DIV+1)$	RW
RN	1	0x0	Управление сбросом ЦАП, активный уровень «0»	RW
EN	0	0x0	Управление питанием ЦАП: 0 – выключен; 1 – включен	RW

Таблица 150 – Регистры DAC_VALUE0, DAC_VALUE1

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31: 12		Зарезервировано	
VALUE	11:0	0x0	Входные данные для преобразования	RW

Таблица 151 – Регистр управления калибруемыми источниками напряжения и тока

Обозначение	Разряды	Начальное значение	Назначение	Доступ
EN	8	0x01	Разрешение работы калибруемых источников: 1 - включить; 0 - выключить	RW
ICOEF	7:4	0x08	Коэффициент настройки опорного источника тока	RW
VCOEF	3:0	0x08	Коэффициент настройки опорного источника напряжения	RW

3.22.2 Работа с контроллером ПДП

В процессе работы ЦАП после окончания текущего преобразования формируется флаг (EMPTY_READ регистра DAC0_CFG или DAC1_CFG) разрешения записи для ПДП контроллера.

Пример настройки канала ПДП для пересылки данных из памяти в DAC_0:

CHx_CFG.ENABLE = 1;

```

CHx_CFG.READ_MODE = 1; // Чтение выполняется из памяти
CHx_CFG.READ_INCREMENT = 1; // чтение с инкрементом адреса
CHx_CFG.READ_SIZE = 2; //чтение слова
CHx_CFG.READ_ACK_EN = 0; // формирование сигналов подтверждения для
чтения отключено

```

```

CHx_CFG.WRITE_MODE = 0; //запись выполняется по запросу от DAC
CHx_CFG.WRITE_REQUEST = 10; // линия DAC_0
CHx_CFG.WRITE_INCREMENT = 0; //запись выполняется в один адрес
CHx_CFG.WRITE_SIZE = 2; //запись слова
CHx_CFG.READ_ACK_EN = 0; // формирование сигналов подтверждения для
записи отключено

```

```

LEN = 3; // 4 байта
SRC = <RAM_ADDR>;
DST=0x00085000 + 0x04; // (ANALOG_REG_BASE_ADDRESS + Offset
DAC0_VALUE)

```

С использованием ПДП из ОЗУ будет считано слово. после чего будет ожидаться флаг разрешения записи. Затем слово запишется в регистр DAC0_VALUE.

3.22.3 Диаграммы выходного сигнала

Выходной сигнал представлен на рисунке 90.

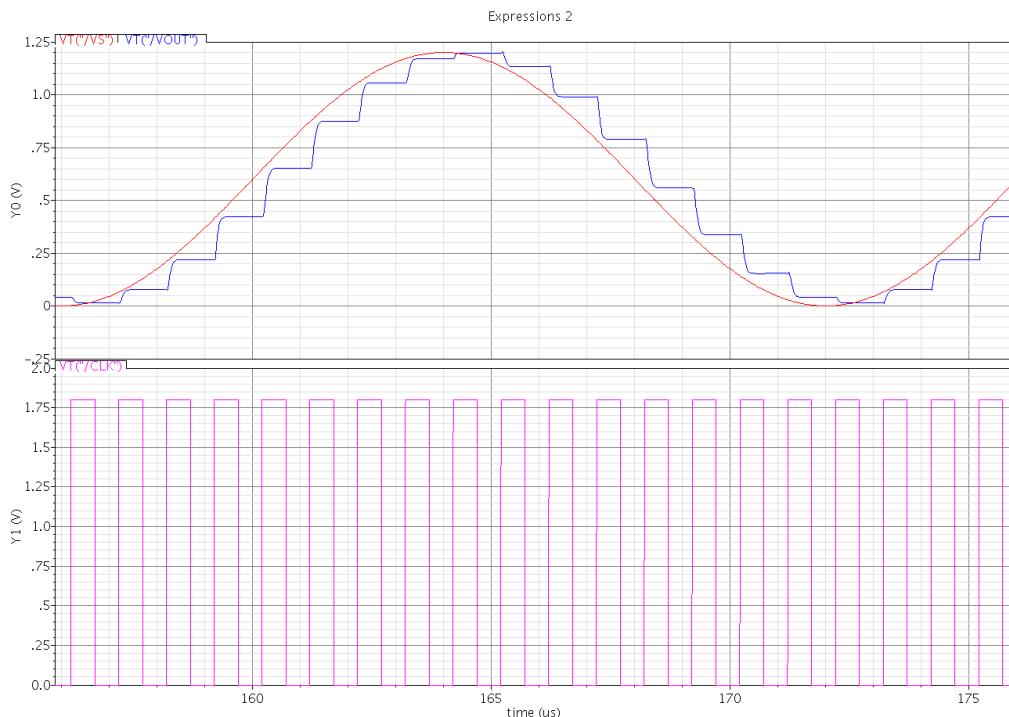


Рисунок 90 – Выходной сигнал ЦАП

3.23 Аналого–цифровой преобразователь (ADC)

Аналого–цифровой преобразователь (АЦП) предназначен для формирования 12-битного цифрового кода, пропорционального аналоговому входному напряжению.

Примечание - В исполнении K1948BK01A8 и K1948BK01A5 блок АЦП не используется.

Параметры АЦП представлены в таблице 152.

Таблица 152 – Параметры АЦП

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра		
		не менее	номинал	не более
Число разрядов АЦП	—	—	12	—
Число каналов мультиплексора	—	—	8	—
Тактовый сигнал АЦП, МГц	F _{CLK}	0.16	32	32
Частота преобразования, КГц	F _s	4		800
Динамический диапазон, свободный от паразитных составляющих, дБ	S _{FDR}	64.9	71.9	
Эффективное число разрядов АЦП	ENOВ	10.5	11.94	—
Дифференциальная нелинейность, Lsb	—	—	1	—
Интегральная нелинейность, Lsb	—	—	4	—
Входное напряжение источника опорного напряжения, В	U _{REF}	1,1	1,2	1,3
Диапазон входного сигнала АЦП, В	—	—	0,015 - U _{REF}	—
Ток по входу опорного напряжения, мкА	I _{UREF}	—	0	—
Внешний опорный ток тестового выходного буфера, мкА	I _{TEST}	0,45	0,5	1
Ток потребления, мкА	I _{CC}	—	—	1022

3.23.1 Состав и принцип работы

Модуль представляет собой АЦП последовательного приближения и входного мультиплексора 8 в 1 с дешифратором 3 в 8. Структурная схема представлена на рисунке 91.

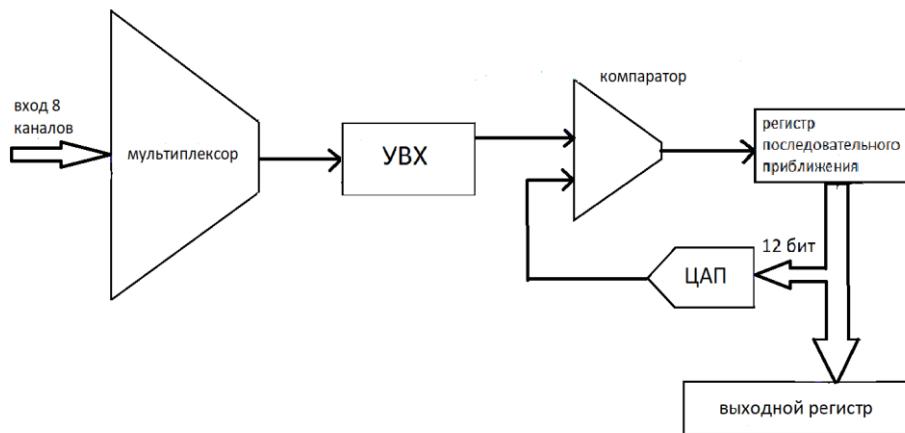


Рисунок 91 – Схема АЦП

АЦП последовательного приближения состоит из резистивного ЦАП, модуля SAR14, SAH (УВХ), компаратора и цифровой управляющей логики.

Аналоговое входное напряжение хранится в блоке УВХ (SAH). Исполнение алгоритма последовательного приближения обеспечивает блок SAR. Выходы ЦАП выставляются согласно значению из регистра SAR. Компаратор выполняет сравнение значений напряжений из УВХ и ЦАП и результат сравнения подается на вход SAR.

Резистивный ЦАП состоит из двух каскадов, каждый из которых обеспечивает разрешение 6 бит. Каждый единичный элемент каскада состоит из одного резистора.

Выходной буфер представляет собой неинвертирующий ОУ с коэффициентом усиления один. Формирователь внутреннего опорного напряжения от источника аналогового напряжения 1,8 В обеспечивает напряжение 1,2 В при $U_{ist} = 1,8$ В. Типичный ток потребления ОУ составляет 250 мкА (не учитывая выходной ток). В выключенном состоянии типичный ток потребления ОУ составляет 30 нА.

Аналоговое входное напряжение, V_{in} , хранится в устройстве выборки/хранения SAH/УВХ. Данные записываются по сигналу Start передним фронтом clock половинной частоты и хранятся с течение цикла преобразования на конденсаторе.

Входная частота делится на 28 и на 2, с формированием дополнительного синхросигнала с фазой, сдвинутой на 90 градусов (блок CLK_div). По сигналу Start в соответствии с прямым и смещенным синхросигналами формируются сигналы сброса установки SAR.

Входной мультиплексор данных включает в себя регистр выбора адреса входных данных, дешифратор 3 в 8 и проходные ключи. Адрес записывается в регистр по сигналу СК, формируемому в блоке SAR в такте C_{1_2}, и сохраняется до следующего СК, что позволяет закончить цикл преобразования данных с текущего входа, даже если адрес сменился. До появления первого сигнала СК по сигналу сброса по умолчанию устанавливается вход данных DIN<0>.

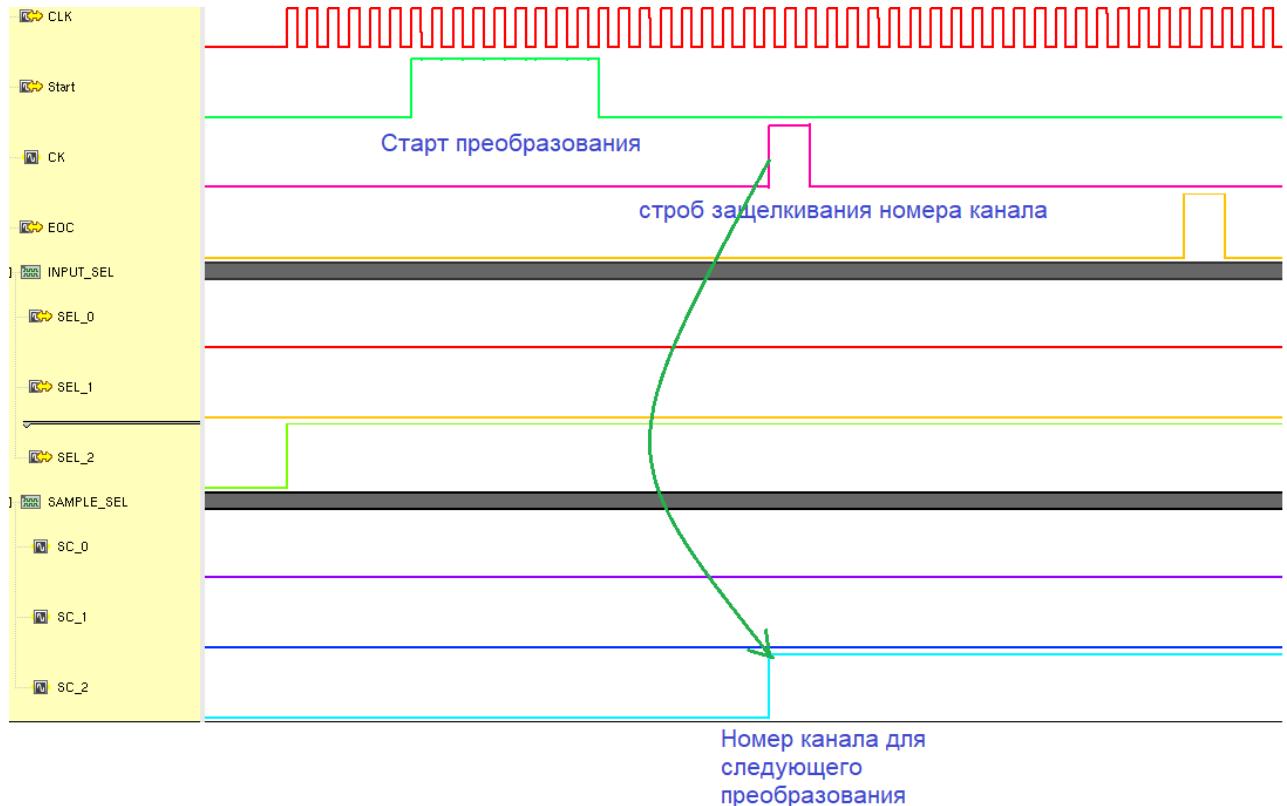


Рисунок 92 – Временные диаграммы переключения каналов АЦП

3.23.2 Управление АЦП

В начале каждого цикла преобразования SAR по сигналу Start и переднему фронту сигнала CLK N-разрядный регистр в составе SAR сначала сбрасывается в 0, а затем устанавливается в значение, равное половине шкалы преобразования, 0b100...00, где MSB - лог. «1». При этом выход ЦАП устанавливается в $\frac{1}{2}$ опорного напряжения (VRF). После этого компаратор сравнивает Vin (выход УВХ) и VDAC. Если Vin больше, чем VDAC, выход компаратора устанавливается в лог. «1», при этом MSB N-разрядного регистра остается в «1»; Если Vin меньше, чем VDAC, выход компаратора устанавливается в лог. «0», при этом MSB сбрасывается в «0». Следующий по старшинству разряд устанавливается в «1» и процесс сравнения повторяется.

Последовательность повторяется для всех разрядов, вплоть до LSB. В конце цикла преобразования вырабатывается сигнал EOC, преобразование завершается, данные из регистра SAR могут быть считаны внешней логикой. Временная диаграмма цикла преобразования приведена на рисунке 93. Вместе со сбросом/установкой и циклом записи данных в УВХ преобразование занимает SAH_TIME тактов сигнала CLK записи данных в УВХ и 28 тактов преобразования. Поэтому для обеспечения частоты сэмплирования 0.8 МГц требуется входной тактовый сигнал частотой не менее 32 МГц. Для корректной работы входного мультиплексора данные дешифратора выбора соответствующего входа данных записываются во внутренний регистр адреса по завершении двух из 28 внутренних тактов преобразования. Таким образом, в SAH данные по новому входу записываются в начале следующего цикла преобразования, позволяя завершить текущий. Сигнал Start должен подаваться на АЦП и защёлкиваться по переднему фронту CLK. При этом его длительность должна быть не менее 12 тактов CLK при заданных параметрах входа данных.

Считывание цифровых данных осуществляется по заднему фронту CLK в такте C1_27 или по переднему фронту CLK в такте C0_0 следующего цикла конвертации при активной «1» на выходе EOC.

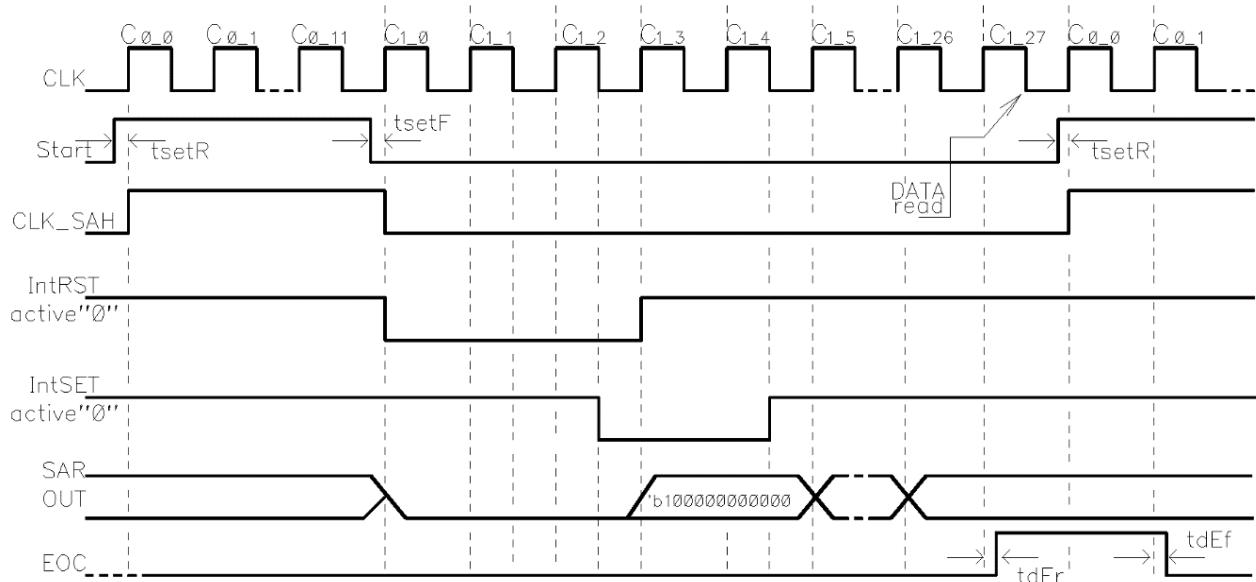


Рисунок 93 – Временная диаграмма цикла преобразования

Для управления АЦП используются регистры блока управления аналоговой подсистемой. Состав регистров управления АЦП представлен в таблице 153.

В таблицах 154 – 158 представлены описания полей регистров АЦП.

Таблица 153 – Регистры управления АЦП

Обозначение	Смещение	Начальное значение	Назначение	Доступ
ADC_CONFIG	0x3C	0x0	Регистр настроек	RW
ADC_CONTINUOUS	0x40	0x0	Управление непрерывным измерением	RW
ADC_SINGLE	0x44	0x0	Управление однократным измерением	WO
ADC_VALID	0x48	0x0	Регистр статуса	RO
ADC_VALUE	0x4C	0x0	Регистр данных	RO
REF_CLB	0x38	0x188	Управление калибруемыми источниками тока и напряжения	RW

Примечание – Смещение относительно стартового адреса блока управления аналоговой подсистемой (0x00085000 в адресном пространстве микроконтроллера).

Таблица 154 – Регистр ADC_CONFIG АЦП

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:9		Зарезервировано	
SAH_TIME	13:8	0xC	Время выборки очередного отсчета в тактах АЦП	RW
Зарезервировано	7		Зарезервировано	
SEL	6:4	0x0	Выбор канала АЦП	RW
EXTPAD	3	0x0	Выбор источника внешнего опорного напряжения: 1 – внешний вывод; 0 – настраиваемый ИОН	RW
EXTEN	2	0x0	Выбор источника опорного напряжения: 1 – внешний; 0 – встроенный	RW
RN	1	0x0	Управление сбросом АЦП, активный уровень «0»	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
EN	0	0x0	Управление питанием АЦП: 0 – выключен; 1 – включен	RW

Таблица 155 – Регистр ADC_CONTINUOUS АЦП

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31: 1	Зарезервировано			
CONTINUOUS	0	0x0	Запуск / остановка непрерывного измерения	RW	

Таблица 156 – Регистр ADC_SINGLE АЦП

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31: 1	Зарезервировано			
SINGLE	0	0x0	Запуск однократного измерения	WO	

Таблица 157 – Регистр ADC_VALID АЦП

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31: 1	Зарезервировано			
VALID	0	0x0	Признак наличия актуальных данных	RO	

Таблица 158 – Регистр ADC_VALUE АЦП

Обозначение	Разряды	Начальное значение	Назначение	Доступ	
Зарезервировано	31: 12	Зарезервировано			
VALUE	11:0	0x0	Результат преобразования	RO	

Таблица 159 – Регистр управления калируемыми источниками напряжения и тока

Обозначение	Разряды	Начальное значение	Назначение	Доступ
EN	8	0x01	Разрешение работы калируемых источников: 1 - включить; 0 - выключить	RW
ICOEF	7:4	0x08	Коэффициент настройки опорного источника тока	RW
VCOEF	3:0	0x08	Коэффициент настройки опорного источника напряжения	RW

3.23.3 Диаграммы выходного сигнала

На рисунке 94 показаны временные диаграммы работы АЦП в режиме с внутренним референсом.

Сигналы АЦП:

- iDac_out – результат преобразования выходного сигнала АЦП с помощью идеального ЦАП;
- SAH_out – выход УВХ;
- DAC_OUT – выход встроенного ЦАП;
- VS – входной сигнал с амплитудой 1 В;
- VREFi – внутренний буферизированный опорный сигнал. Частота дискретизации 1000 кГц, частота сигнала 156250 Гц.

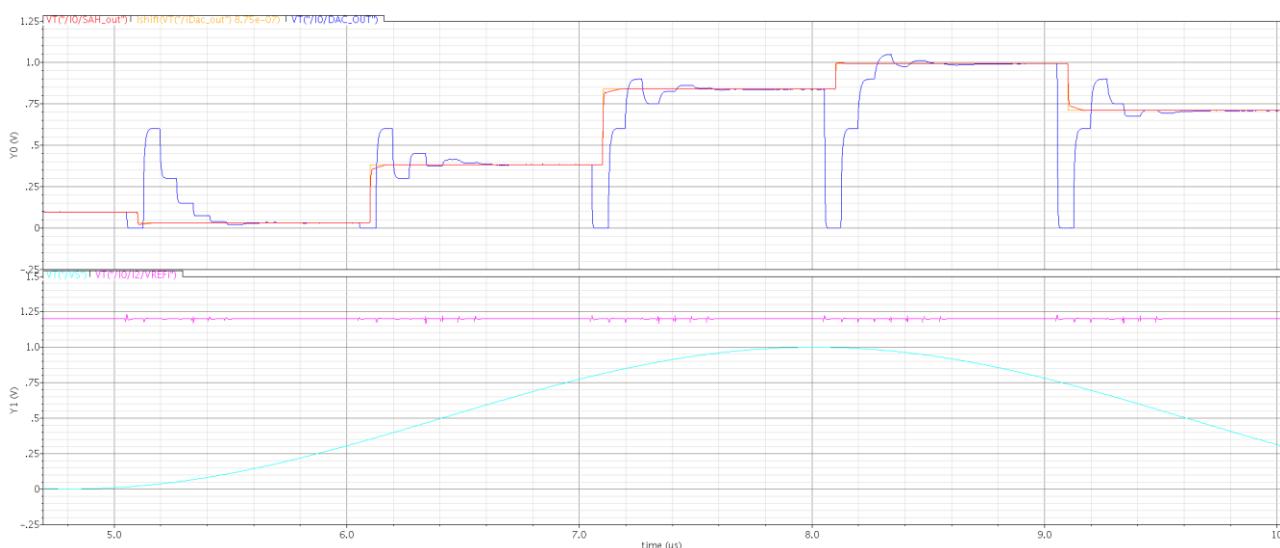


Рисунок 94 – Временные диаграммы работы АЦП

3.24 Блок управления режимом загрузки (BOOT MANAGER)

Блок обеспечивает изменение режима загрузки. Значение выводов Boot0 и Boot1 фиксируются при включении питания и при внешнем сбросе (вывод RST#). Программно возможно изменить способ загрузки путем записи необходимой конфигурации в регистр BOOT. Обновленное значение режима загрузки будет использовано при следующих событиях (при условии разрешения соответствующего события в регистре SYS_MASK блока WAKEUP):

- выход из режима ожидания при срабатывании будильника;
- выход из режима ожидания при активном уровне внешнего вывода EXT_WU;
- сброса системного домена при снижении питания (индикация от LDO_SYS);
- сброса системного домена при снижении питания (индикация от схемы слежения);
- формирования сброса системного домена при появлении сигнала BOR.
- Регистра блока управления режимом загрузки представлены в таблице 160, поля регистров представлены в таблице 162.

Таблица 160 – Регистры блока управления загрузкой

Обозначение	Смещение	Начальное значение	Назначение	Доступ
BOOT	0x00	-	Регистр режима загрузки.	RW

Таблица 161 – Регистр управления загрузкой

Обозначение	Разряды	Начальное значение	Назначение	Доступ
BOOT_MODE	1:0	-	Регистр режима загрузки. При чтении, возвращает режим, который будет использоваться для следующего сброса системного домена. При записи – установка нового режима загрузки. 0b00 - Старт из встроенной памяти EEPROM; 0b01 - Старт из системного ОЗУ; 0b10 - Старт из внешней памяти с использованием контроллера SPIFI; 0b11 - Зарезервировано	RW

4 Подсистема тактирования

Микросхема МК32 имеет гибкую систему управления тактовыми сигналами.

Общая схема тактирования представлена на рисунке 95.

Основные возможности схемы управления тактовыми сигналами:

- возможность выбора одного из четырех источников для тактового сигнала ядра (встроенный осциллятор 32 МГц (HSI32M), внешний осциллятор (OSC32M), встроенный осциллятора 32 кГц (LSI32K), внешний часовой осциллятор (OSC32K);
- возможность выбора одного из четырех источников для тактового сигнала сторожевого таймера (встроенный осциллятор 32 МГц (HSI32M), внешний осциллятор 32 МГц (OSC32M), встроенный осциллятор 32 кГц (LSI32K), внешний часовой осциллятор (OSC32K);
- возможность отключения тактовых сигналов периферийных устройств (для каждого независимо от других);
- возможность отключения тактовых сигналов памяти (ОЗУ/EEPROM/контроллер внешней памяти SPIFI);
- возможность задания делителя частоты шины АHB;
- возможность задания делителей периферийных шин APB_M/APB_P,
- наличие монитора источников тактового сигнала.

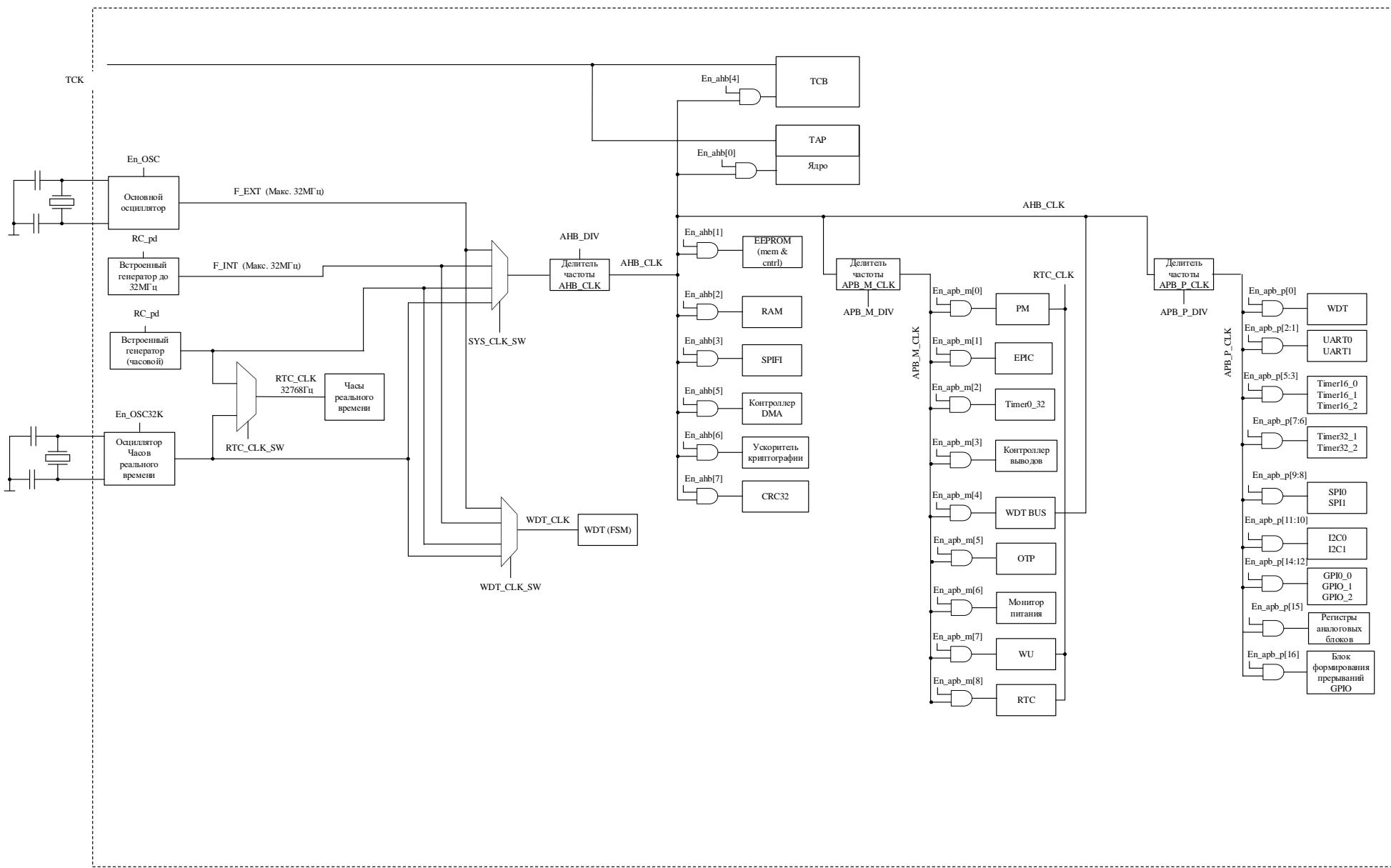


Рисунок 95 – Схема тактирования микросхемы

Для выбора источников тактирования, задания делителей частоты, управлением HSI32M и OSC32M используются регистры модулей PowerManager и WakeUp. Для управления тактированием часов реального времени и управлением LSI32K и OSC32K используются регистры модуля WakeUp.

При включении питания и по внешнему сбросу:

- приоритетным источником тактирования системы является внешний осциллятор 32 МГц (OSC32M);
- разрешено автоматическое переключение на другой источник тактирования системы;
- значения делителей шин АНВ, APB_M, APB_P равны 0, то есть указанные шины работают на частоте осциллятора;
- тактирование ядра, блока TCB, блоков RAM, EEPROM, SPIFI, PowerManager, WakeUp, модуля порта ввода-вывода включено, тактирование остальных устройств выключено;
- тактирование сторожевого таймера осуществляется от встроенного осциллятора 32 МГц;
- внешние осцилляторы OSC32K и OSC32M включены,
- встроенные генераторы HSI32M и LSI32K включены.

Для подстройки встроенного осциллятора HSI32M используются поправочные коэффициент с допустимым значением от 0 до 255. Управление питанием OSC32M, HSI32M и калибровкой HSI32M осуществляется регистром CLOCKS_SYS блока WakeUp.

Для подстройки встроенного осциллятора LSI32K используются поправочные коэффициент с допустимым значением от 0 до 15. Управление питанием OSC32K, LSI32M и калибровкой LSI32M осуществляется регистром CLOCKS_BU блока WakeUp.

Описание полей регистров модуля WU для управления тактированием представлено в таблице 162. Описание полей регистров модуля PM для управления тактированием представлено в таблице 165.

Таблица 162 – Регистры тактированием модуля WU

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CLOCKS_BU	0x10	0x200	Регистр управления тактированием батарейного домена (таблица 179)	RW
CLOCKS_SYS	0x14	0x200	Регистр управления тактированием системного домена (таблица 180)	RW

Таблица 163 - Регистр управления тактированием батарейного домена CLOCKS_BU

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:15		Зарезервировано	
OSC32K_SM	14	0x0	Режим повышенного потребления, активный уровень «0» для OSC32K. Используется для более быстрого и стабильного запуска генератора, повышает устойчивость генерации на предельных режимах эксплуатации. После запуска допускается установить значение «1».	RW
RTC_CLK_MUX	11:10	0x0	Выбор приоритетного источника тактирования часов реального времени: 0x0 – автоматический выбор. При наличии обоих источников 32кГц выбирается внутренний LSI32K; 0x1 – внутренний LSI32K; 0x2 – внешний осциллятор OSC32K	RW
ADJ_LSI32K	9:6	0x80	Поправочные коэффициенты LSI32K	RW
Зарезервировано	5:2		Зарезервировано	
LSI32K_EN	1	0x0	Включение/отключение LSI32K 0 – включение; 1 – отключение	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
OSC32K_EN	0	0x0	Включение/отключение внешнего осциллятора на 32 КГц 0 – включение; 1 – отключение	RW

Таблица 164 - Регистр управления тактированием системного домена **CLOCKS_SYS**

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:12		Зарезервировано	
FORCE_32K_CLK	11:10	0x0	Принудительное переключение на опорный источник для монитора частоты 0b00 и 0b11 – автоматический выбор; 0b01 – принудительно выбран LSI32K; 0b10 – принудительно выбран OSC32K.	RW
ADJ_HSI32M	9:2	0x80	Поправочные коэффициенты HSI32M	RW
HSI32M_EN	1	0x0	Включение/отключение HSI32M 0 – включение; 1 – отключение	RW
OSC32M_EN	0	0x0	Включение/отключение внешнего осциллятора на 32 МГц 0 – включение; 1 – отключение	RW

Таблица 165 – Регистры управления тактированием модуля РМ

Обозначение	Смещение	Начальное значение	Назначение	Доступ
DIV_AHB	0x00	0x00	Задает значение делителя шины АНВ. Частота шины АНВ (F_{AHB}) рассчитывается, как $F_{SYS}/(DIV_AHB+1)$	RW
DIV_APB_M	0x04	0x00	Задает значение делителя шины APB_M. Частота шины APB_M (F_{APB_M}) рассчитывается, как $F_{APB}/(Div_APM_M+1)$	RW
DIV_APB_P	0x08	0x00	Задает значение делителя шины APB_P. Частота шины APB_P (F_{APB_P}) рассчитывается, как $F_{APB}/(Div_APM_P+1)$	RW
CLK_AHB_SET	0x0C	0x1F	Регистр включения тактированием устройств на шине АНВ Каждому биту соответствует устройство: 0 – Ядро; 1 – EEPROM; 2 – RAM; 3 – SPIFI; 4 – TCB; 5 – DMA; 6 – ускоритель криптографии; 7 – CRC32	RW1S
CLK_AHB_CLEAR	0x10	0x00	Регистр выключения тактированием устройств на шине АНВ. Каждому биту соответствует устройство, аналогично CLK_AHB_SET	RW1C

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CLK_APB_M_SET	0x14	0x89	Регистр включения та��ированием устройств на шине APB_M. Каждому биту соответствует устройство: 0 – Блок управления питанием (PM - Power Monitor); 1 – контроллер прерываний; 2 – таймер32_0; 3 – контроллер выводов; 4 – Сторожевой таймер шины 5 – OTP; 6 – монитор питания системного домена 7 – блок WU батарейного домена 8 – RTC батарейного домена	RW1S
CLK_APB_M_CLEAR	0x18	0x00	Регистр выключения та��ированием устройств на шине APB_M. Каждому биту соответствует устройство, аналогично CLK_APB_M_SET	RW1C

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CLK_APB_P_SET	0x1C	0x00	Регистр включения тактированием устройств на шине APB_P. Каждому биту соответствует одно устройство: 0 – сторожевой таймер; 1 – UART_0; 2 – UART_1; 3 – таймер16_0; 4 – таймер16_1; 5 – таймер16_2; 6 – таймер32_1; 7 – таймер32_2; 8 – SPI_0; 9 – SPI_1; 10 – I ² C_0; 11 – I ² C_1; 12 – GPIO_0; 13 – GPIO_1; 14 – GPIO_2; 15 – регистры аналоговых блоков; 16 – схема формирования прерываний GPIO	RW1S
CLK_APB_P_CLEAR	0x20	0x00	Регистр выключения тактированием устройств на шине APB_P. Каждому биту соответствует устройство, аналогично CLK_APB_P_SET	RW1C
AHB_MUX	0x24	0x00	Настройка источника тактирования системы (таблица 171)	RW
WDT_CLK_MUX	0x28	0x00	Выбор источника тактирования сторожевого таймера: 0 – внешний OSC32M; 1 – внутренний HSI32M; 2 – внешний OSC32K; 3 – внутренний LSI32K;	RW

Частоты сигналов рассчитываются по формулам (15– 17):

$$F_{AHB} = F_{источника}/(DIV_AHB + 1), \quad (15)$$

$$F_{APB_M} = F_{AHB}/(DIV_APB_M + 1), \quad (16)$$

$$F_{APB_P} = F_{AHB}/(DIV_APB_P + 1). \quad (17)$$

4.1 Встроенные осцилляторы с частотой 32 МГц и 32 кГц

Для создания точных временных импульсов используется генератор линейно нарастающего напряжения и компаратор, сравнивающий сигнал выхода ГЛИН с опорным потенциалом. Пара таких компараторов поочередно сбрасывает и устанавливает rs-триггер, формирующий выходной импульсный сигнал (рисунок 96). Опорный потенциал на компараторы подается с ЦАП, что позволяет отрегулировать появившиеся вследствие разброса техпроцесса смещения.

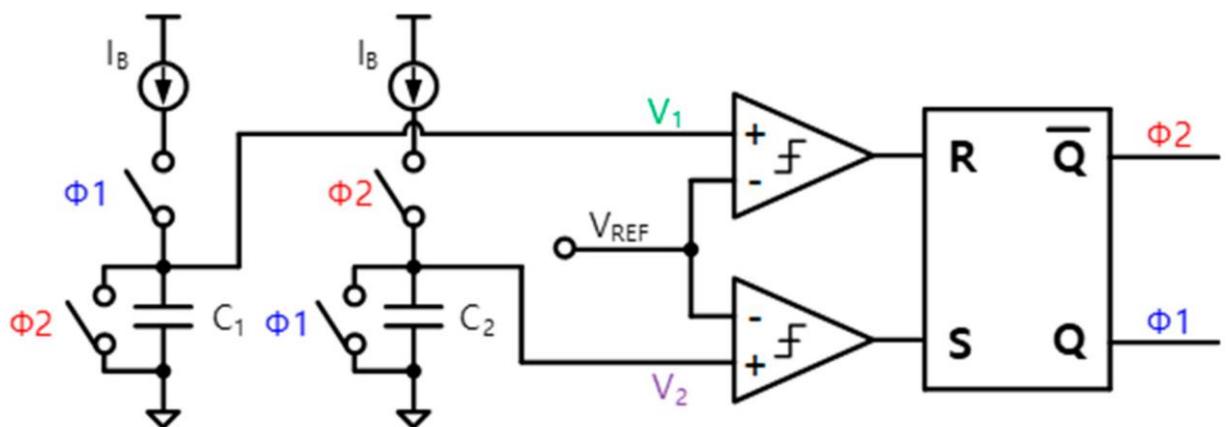


Рисунок 96 – Схема осциллятора

Схема осциллятора состоит из генератора линейно нарастающего напряжения, имеющего пару выходов, сигнал формируется на том выходе, который выбран входным сигналом phase. К каждому из выходов ГЛИН подключено по компаратору, они формируют сигналы R и S для триггера. Выход RS-триггера, служит сигналом phase, а также выходным тактовым сигналом. Опорный потенциал на компараторы подается с ЦАП, что позволяет отрегулировать появившиеся вследствие разброса техпроцесса смещения.

Схема ГЛИН имеет две согласованные емкости, которые поочередно заряжаются одним опорным источником. Используются конденсаторы поликремний 1-поликремний

2 ср1р2, емкость которых не зависит от напряжения на обкладках и температуры. Разброс конденсаторов (от минус 11 до плюс 9 %). Согласно формуле, время роста $T = VC/I$, таким образом, смещения в конденсаторе можно компенсировать опорным потенциалом.

Источник токов имеет статичное смещение от разброса порядка 10 %, которое также можно скомпенсировать опорным потенциалом.

Компаратор состоит из входного дифференциального каскада, ток в который задается от gm-согласованного источника тока, что уменьшает влияние температуры на задержку преобразования. Время преобразования и входное смещение – это статичные ошибки, которые также могут быть скомпенсированы опорным потенциалом генератора.

Опорный потенциал подается с 8-ми битного ЦАП, программируется в диапазоне от 1,24 до 0,75 В, позволяя корректировать частоту $\pm 25 \%$.

Нагрузка кварцевых резонаторов выбирается 10-30 пФ в зависимости от нагрузочной емкости резонатора:

$$C_1 = C_2 = 2(C_L - (C_k + C_m)) \quad (18)$$

где C_L – нагрузочная емкость используемого кварцевого резонатора;

C_k – суммарная емкость выводов XI и XO;

C_m – суммарная емкость проводников на печатной плате.

В большинстве случаев для предварительной оценки можно принимать сумму $C_k + C_m = (3..7) \text{ пФ}$. Рекомендуется использовать кварцевые резонаторы с нагрузочной ёмкостью $C_L = (10..15) \text{ пФ}$.

При отсутствии внешнего кварца вывод XI следует заземлить.

Возможно использование внешних генераторов вместо резонаторов. Для этого вывод генератора OUT подключается к выводу OSC32K_XI или OSC32M_XI микроконтроллера. Выводы OSC32K_XO и OSC32M_XO при использовании генератора можно оставить неподключенным.

4.2 Вывод для подключения внешнего кварцевого резонатора до 32 МГц

Осциллятор построен на схеме Пирса (рисунок 97) обеспечивает работу в диапазоне частот 1-32 МГц.

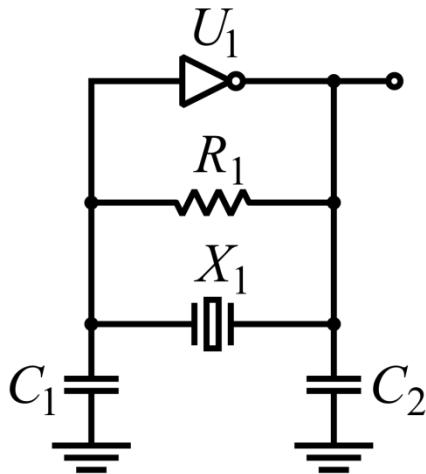


Рисунок 97 – Осциллятор Пирса

КМОП схема генератора размещается в ИО ячейке. В нее входят инвертирующий усилитель, источник опорных потенциалов для питания усилителя, резистор обратной связи 500 кОм RO0, ESD защита контактной площадки, драйвер для преобразования гармонического сигнала генератора в выходной импульсный.

4.3 Вывод для подключения внешнего кварцевого резонатора 32 кГц

Осциллятор построен на схеме Пирса. КМОП схема генератора размещается в ИО ячейке. В нее входят инвертирующий усилитель, источник опорных потенциалов И0 для питания усилителя, резистор обратной связи, ESD защита контактной площадки, компаратор для преобразования гармонического сигнала генератора в выходной импульсный, драйвер для понижения уровня сигнала до уровня ядра и коррекции скважности сигнала.

4.4 Монитор частоты системного домена

Блок монитора тактовой частоты системного домена предназначен для формирования сигналов прерывания/состояния в случаях отсутствия или пропадания входных синхросигналов.

Функциональные характеристики:

- выбор одного из входных синхросигналов clk_32k опорным в стартовом режиме после снятия сброса, с возможностью последующего переключения;
- независимое детектирование наличия/пропадания входных синхросигналов;

- автоматическое переключение опорного сигнала, в случае его пропадания, на запасной и продолжение детектирования в аварийном режиме.

Модуль может работать в трех режимах: стартовом, динамическом рабочем и аварийном.

4.4.1 Стартовый режим

В стартовом режиме определяется опорный источник частоты 32к по расчету заданного количества импульсов, причем предпочтение отдается внутреннему. Первый досчитавший счетчик позволяет своему источнику тактового сигнала стать взаимоисключающим опорным для монитора, соответствующий входной сигнал подключается к выходу модуля clk_32k_ref.

Перед выставлением значений на входе sel_refclk для смены опорного синхросигнала необходимо убедиться в наличии соответствующего синхросигнала, проверив состояние флагов c_32k_ext_det и c_32k_int_det. В противном случае схема переключится, и выход clk_32k_ref не будет выдавать синхросигнал, из этого состояния блок может выйти только с помощью сброса.

4.4.2 Динамический рабочий режим

После выбора опорного тактового сигнала запускается динамическая схема детекции. Флаги *_det будут выставлены в активный уровень «1», при насчете определенного количества тактов входного синхросигнала. При пропадании какого-либо из сигналов через то же количество тактов пропадет соответствующий флаг *_det. Требуемое количество тактов для каждого синхросигнала указано в таблице 166. Результаты автоматической детекции могут быть использованы для выбора синхросигналов и формирования прерываний.

Таблица 166 – Количество тактов для детектирования частоты

Имя сигнала	Флаг	Количество полных тактов опорной частоты до изменения флага
clk_32k_int	c_32k_ext_det	16
clk_32k_ext	c_32k_int_det	16
clk_32M_ext	c_32M_ext_det	4
clk_32M_int	c_32M_int_det	4

Для начала детекции сигналов на входах clk_32k и работы соответствующих флагов *_det требуется подача обоих синхросигналов.

На рисунке 98 показано начало работы блока в стартовом режиме, когда первым появляется сигнал clk_32k_ext, он в этом случае выбирается опорным. Далее показана работа флагов det для остальных входов. В моменты времени 85 мс и 105 мс произведены принудительные переключения опорного сигнала с внешнего clk_32k_ext на внутренний clk_32k_int и обратно.

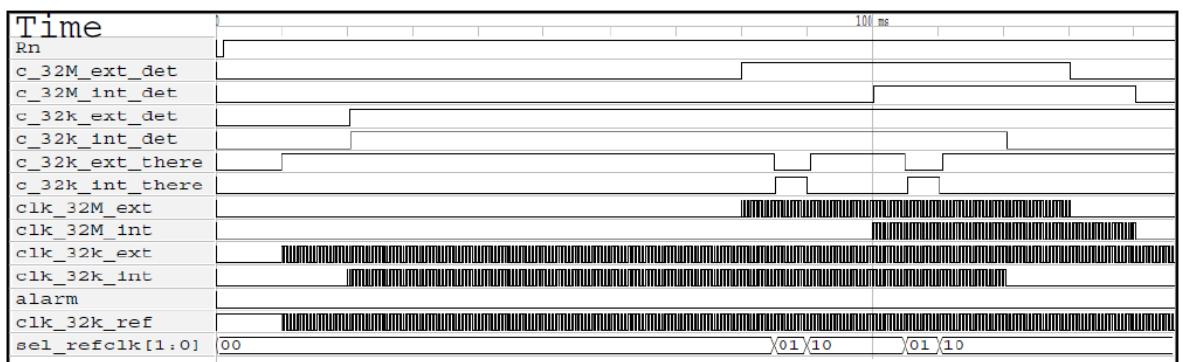


Рисунок 98 – Стартовый и динамический режимы

4.4.3 Аварийный режим

Переход в аварийный режим происходит при пропадании входного сигнала, выбранного опорным. Сам переход сопровождается установкой флага alarm. При этом продолжает работу резервная часть модуля, позволяющая отслеживать наличие сигналов clk_32M_*. Выход clk_32k_ref подключается к резервному сигналу при его наличии.

На рисунке 99 показана установка различных комбинаций на входе sel_refclk. В момент времени 240 мс происходит пропадание сигнала clk_32k_ext, выбранного опорным (флаг c_32k_ext_there указывает на это), и переход модуля в аварийный режим работы, о чем сообщает флаг alarm. Опорный сигнал clk_32k_ref переходит на подключение к резервному входу clk_32k_int.

На рисунке 100 вначале показан выход из аварийного режима при появлении clk_32k_ext. Сигнал clk_32k_ref возвращается к выдаче основного сигнала clk_32k_ext. Затем повторяется пропадание опорного сигнала clk_32k_ext и переход в аварийный режим. В аварийном режиме вход sel_refclk отключается от модуля и не влияет на его работу. Далее показана выдача флагов 32M_*_det в аварийном режиме и пропадание резервного сигнала, в результате которого работа модуля полностью останавливается.

Работоспособность может быть восстановлена только в случае появления обоих сигналов clk_32k или после полного сброса модуля.

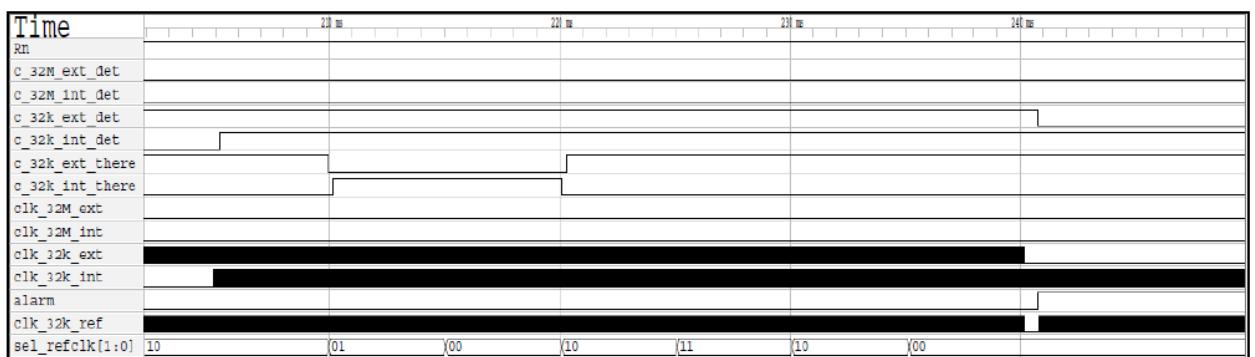


Рисунок 99 – Динамический и аварийный режимы

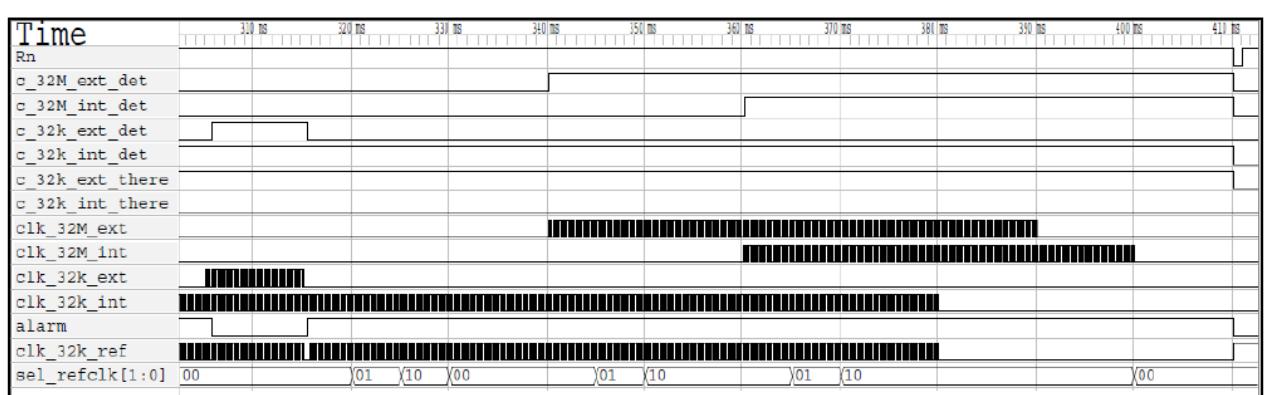


Рисунок 100 – Аварийный режим

Управление монитором частоты системного реализуется через следующие регистры блока управления питанием и тактирования (таблица 167) и регистры блока WakeUp (таблица 189)

Таблица 167 – Регистры блока управления питанием и тактированием для управления монитором частоты

Обозначение	Смещение	Разряды	Назначение	Доступ
AHB_CLK_MUX	0x24	1:0	Выбор источника тактирования системы: 0 – внешний OSC32M; 1 – внутренний HSI32M; 2 – внешний OSC32K; 3 – внутренний LSI32K	RW
FORCE_MUX	0x24	2	Запрет на принудительное переключение с выбранного источника тактирования при пропадании тактирования	RW

Обозначение	Смещение	Разряды	Назначение	Доступ
FREQ_MASK	0x34	4:0	<p>Настройки монитора частоты:</p> <p>FREQ_MASK[0] – маска прерывания при пропадании частоты LSI32K;</p> <p>FREQ_MASK[1] – маска прерывания при пропадании частоты внешнего осциллятора 32 кГц – OSC32K;</p> <p>FREQ_MASK[2] – маска прерывания при пропадании частоты HSI32M;</p> <p>FREQ_MASK[3] – маска прерывания при пропадании частоты внешнего осциллятора 32 МГц – OSC32M;</p> <p>FREQ_MASK[4] – разрешение установки делителя шины АНВ в 0, при пропадании внешней и внутренней частоты 32МГц – HSI32M и OSC32M</p>	RW
FREQ_STATUS	0x38	3:0	<p>Статус монитора частоты (0 – источник тактирования не детектируется, 1 – тактовый источник детектируется)</p> <p>FREQ_STATUS[0] – внутренний LSI32K;</p> <p>FREQ_STATUS[1] – внешний OSC32K;</p> <p>FREQ_STATUS[2] – внутренний HSI32M;</p> <p>FREQ_STATUS[3] – внешний OSC32M</p>	RO

Таблица 168 – Регистры блока Wakeup

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CLOCKS_SYS	11:10	0x200	<p>Принудительное переключение на опорный источник:</p> <p>0b00 и 0b11 – автоматический выбор;</p> <p>0b01 – принудительно выбран LSI32K;</p> <p>0b10 – принудительно выбран OSC32K.</p>	RW

5 Подсистема питания

В микросхеме MIK32 реализовано независимое управление питанием и тактированием системы в целом и периферийных устройств.

Микроконтроллер имеет два домена питания:

- батарейный домен, содержащий часы реального времени, набор регистров для хранения данных при отключении основного питания, блок управления режимами (модуль WakeUp), LSI32K, OSC32K;
- системный домен, содержащий коммутационную матрицу АНВ, контроллер прямого доступа, контроллер прерываний, ядро SCR1, внутреннюю оперативную память, EEPROM и контроллер внешней памяти SPIFI, периферийные устройства, ускоритель криптографии, блок вычисления контрольных сумм, регистр управления аналоговыми блоками, HSI32M, OSC32M.

Распределение блоков по доменам питания микросхемы представлены на рисунке 101.

Режим «Активный» подразумевает режим с максимальной производительностью: максимальная системная частота, минимальные делители, тактирование всех задействованных периферийных блоков включено.

Режим «Пониженного потребления» предполагает уменьшение производительности и потребления за счет снижения рабочей частоты, временное отключение тактирования периферийных устройств, которые не используются.

Режим «Активный» и режим «Пониженного потребления» достаточно условны, так как, отличаются выбранными источником тактирования, установленными делителями частот, отключенным тактированием тех или иных устройств, что определяется конкретной задачей. Основное отличие от других режимов - функционирование ядра.

«Спящий» режим подразумевает автоматическое отключение частоты ядра, некоторых модулей памяти. Переход осуществляется записью в регистр SLEEP_MODE модуля PowerManager. При записи отключается тактирование ядра и в зависимости от записываемого значения отключается тактирование следующих модулей:

- SLEEP_MODE[0] = 1 – отключение тактирования ядра;

- SLEEP_MODE[1] = 1 – отключение тактирования EEPROM;
- SLEEP_MODE[2] = 1 – отключение тактирования ОЗУ;
- SLEEP_MODE[3] = 1 – отключение тактирования контроллера SPIFI.

Выход из режима по прерыванию от используемых периферийных устройств.

В режимах «Активный», «Спящий», «Пониженного потребления» возможно программное снижение частоты шин APB_M, APB_P, а также отключения тактирования отдельных периферийных блоков.

В режиме «Стоп» автоматически отключается тактирование всех периферий за исключением блоков батарейного домена. Переход осуществляется записью в регистр STOP модуля WakeUp. Выход из режима по событию на выводе EXT_WU, будильника, или по сбросу.

Во всех режимах, кроме ожидания, питание всех доменов включено. В режиме ожидания остается включенным только батарейный домен. Системный LDO выключен. Переход осуществляется записью в регистр SYS_POWEROFF модуля WakeUp. Выход из режима по событию на выводе EXT_WU, будильника, или по сбросу.

Управление питанием осуществляется модулями WakeUP, PowerManager, а управление тактированием модулем PowerManager.

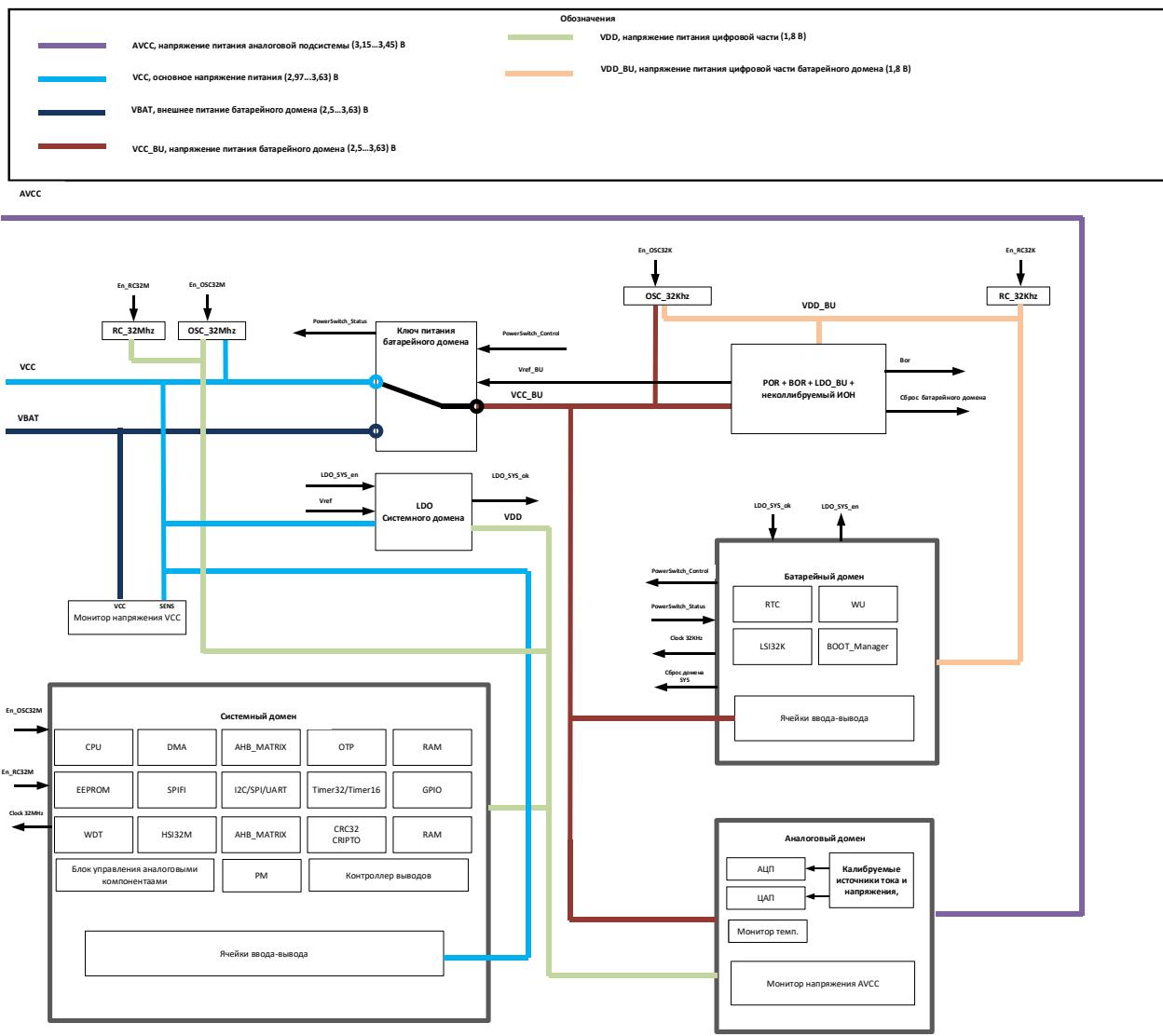


Рисунок 101 – Домены питания

Таблица 169 – Основные режимы работы микроконтроллера

Режим	Состояние процессора	Состояние памяти	Состояние периферии	Состояние системного и батарейного домена	Тактирование	Вход в режим	Выход из режима
Активный	Процессор работает в нормальном режиме.	ОЗУ, EEPROM, SPIFI Работают в нормальном режиме	Периферия работает в нормальном режиме.	Системный домен включен, батарейный домен включен	От внешнего осциллятора с частотой до 32 МГц От встроенного осциллятора с частотой до 32 МГц	Режим после сброса/подачи питания	–
Пониженного энергопотребления	Процессор работает в нормальном режиме.	ОЗУ, EEPROM, SPIFI Работают в нормальном режиме. Рекомендуется отключать тактирование неиспользуемых периферийных модулей программно (пользователем) записью в регистры: PM.CLK_AHB_CLEAR PM.CLK_APB_M_CLEAR PM.CLK_APB_P_CLEAR	Рекомендуется отключать тактирование неиспользуемых периферийных модулей программно (пользователем) записью в регистры: PM.CLK_AHB_CLEAR PM.CLK_APB_M_CLEAR PM.CLK_APB_P_CLEAR	Системный домен включен, батарейный домен включен	От внешнего часовогого осциллятора от встроенного часовогого осциллятора От внешнего осциллятора с частотой до 32 МГц с делителем системной шины	Программно, переключившись на нужный источник тактирования или выставив необходимый делитель	Программно, переключившись на нужный источник тактирования или выставив необходимый делитель
Спящий	Ядро в состоянии sleep	Тактирование ОЗУ, EEPROM, SPIFI может быть выключено при записи в PM.SLEEP_MODE	Рекомендуется отключать тактирование неиспользуемых периферийных модулей программно (пользователем) записью в регистры: PM.CLK_AHB_CLEAR PM.CLK_APB_M_CLEAR PM.CLK_APB_P_CLEAR	Системный домен включен, батарейный домен включен	От внешнего осциллятора с частотой до 32 МГц От встроенного осциллятора с частотой до 32 МГц	Записью в регистр PM.SLEEP_MODE	По прерыванию
Стоп	Ядро в состоянии sleep	Тактирование ОЗУ, EEPROM, SPIFI выключено	Тактированием периферия выключено	Системный домен включен, батарейный домен включен.	Тактирование системы выключено	Запись в регистр WAKEUP.STOP	По сбросу/RTC/выводу EXT_WU

Режим	Состояние процессора	Состояние памяти	Состояние периферии	Состояние системного и батарейного домена	Тактирование	Вход в режим	Выход из режима
Ожидание	Питание выключено	Питание выключено	Питание выключено	Системный домен выключен, батарейный домен включен	Тактирование системы выключено	Запись в регистр WAKEUP. SYS_POWEROFF	По сбросу/RTC/выводу EXT_WU

5.1 Блок управления питанием и тактированием (POWER MANAGER)

Блок управления питанием и тактированием доменов (PowerManager) обеспечивает управление тактовыми сигналами (установка коэффициентов делителей для шин, отключение тактирования отдельных устройств).

Регистры блока представлены в таблице 170.

Таблица 170 – Регистры блока управления питанием и тактированием

Обозначение	Смещение	Начальное значение	Назначение	Доступ
DIV_AHB	0x00	0x00	Задает значение делителя шины АHB. Частота шины АHB (F_{AHB}) рассчитывается, как $F_{SYS}/(DIV_AHB+1)$	RW
DIV_APB_M	0x04	0x00	Задает значение делителя шины APB_M. Частота шины APB_M (F_{APB_M}) рассчитывается, как $F_{APB}/(Div_APM_M+1)$	RW
DIV_APB_P	0x08	0x00	Задает значение делителя шины APB_P. Частота шины APB_P (F_{APB_P}) рассчитывается, как $F_{APB}/(Div_APM_P+1)$	RW
CLK_AHB_SET	0x0C	0x1F	Регистр включения тактированием устройств нашине АHB Каждому биту соответствует устройство: 0 – Ядро; 1 – EEPROM; 2 – RAM; 3 – SPIFI; 4 – TCB; 5 – DMA; 6 – ускоритель криптографии; 7 – CRC32	RW1S
CLK_AHB_CLEAR	0x10	0x00	Регистр выключения тактированием устройств нашине АHB. Каждому биту соответствует устройство, аналогично CLK_AHB_SET	RW1C

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CLK_APB_M_SET	0x14	0x89	<p>Регистр включения таскированием устройств на шине APB_M. Каждому биту соответствует устройство:</p> <ul style="list-style-type: none"> 0 – Блок управления питанием (PM - Power Monitor); 1 – контроллер прерываний; 2 – таймер32_0; 3 – контроллер выводов; 4 – Сторожевой таймер шины 5 – OTP; 6 – монитор питания системного домена 7 – блок WU батарейного домена 8 – RTC батарейного домена 	RW1S
CLK_APB_M_CLEAR	0x18	0x00	Регистр выключения таскированием устройств на шине APB_M. Каждому биту соответствует устройство, аналогично CLK_APB_M_SET	RW1C

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CLK_APB_P_SET	0x1C	0x00	Регистр включения тактированием устройств на шине APB_P. Каждому биту соответствует одно устройство: 0 – сторожевой таймер; 1 – UART_0; 2 – UART_1; 3 – таймер16_0; 4 – таймер16_1; 5 – таймер16_2; 6 – таймер32_1; 7 – таймер32_2; 8 – SPI_0; 9 – SPI_1; 10 – I ² C_0; 11 – I ² C_1; 12 – GPIO_0; 13 – GPIO_1; 14 – GPIO_2; 15 – регистры аналоговых блоков; 16 – схема формирования прерываний GPIO	RW1S
CLK_APB_P_CLEAR	0x20	0x00	Регистр выключения тактированием устройств на шине APB_P. Каждому биту соответствует устройство, аналогично CLK_APB_P_SET	RW1C
AHB_MUX	0x24	0x00	Настройка источника тактирования системы (таблица 171)	RW
WDT_CLK_MUX	0x28	0x00	Выбор источника тактирования сторожевого таймера: 0 – внешний OSC32M; 1 – внутренний HSI32M; 2 – внешний OSC32K; 3 – внутренний LSI32K;	RW

Обозначение	Смещение	Начальное значение	Назначение	Доступ
CPU_RTC_CLK_MUX	0x2C	0x00	Выбор источника тактирования RTC в составе ядра 0 – внешний OSC32K; 1 – внутренний LSI32K;	RW
TIMER_CFG	0x30	0x00	Выбор источника тактирования для таймеров (таблица 172)	RW
FREQ_MASK	0x34	0x10	Настройки прерываний монитора частоты (таблица 173)	RW
FREQ_STATUS	0x38	0x00	Статус монитора частоты (таблица 174)	RW
SLEEP_MODE	0x3C	0x00	Переход в спящий режим осуществляется записью в данный регистр. При записи отключается тактирование ядра. В зависимости от записываемого значения отключается тактирование следующих модулей: SLEEP_MODE [0] = 1 – отключение тактирования ядра; SLEEP_MODE [1] = 1 – отключение тактирования EEPROM; SLEEP_MODE [2] = 1 – отключение тактирования ОЗУ; SLEEP_MODE [3] = 1 – отключение тактирования контроллера SPIFI.	RW

Таблица 171 – Регистр настройки источника тактирования системы АНВ_MUX

Обозначение	Разряды	Начальное значение	Назначение	Доступ
FORCE_MUX	2	0x00	Запрет на принудительное переключение с выбранного источника тактирования при пропадании тактирования 0 – разрешение автоматической смены частоты; 1 – запрет автоматической смены частоты;	RW
AHB_Clk Mux	1:0	0x00	Источника тактирования системы: 0 – внешний OSC32M; 1 – внутренний HSI32M; 2 – внешний OSC32K; 3 – внутренний LSI32K	RW

Таблица 172 – Регистр настройки тактирования для таймеров TIMER_CFG

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MUX_TIM16_2	17:15	0x00	Выбор тактового сигнала для Timer16_3. Значение, аналогичны MUX_TIM16_0	RW
MUX_TIM16_1	14:12	0x00	Выбор тактового сигнала для Timer16_2. Значение, аналогичны MUX_TIM16_0	RW
MUX_TIM16_0	11:9	0x00	Выбор тактового сигнала для Timer16_0: 0x0 – системная частота (sys_clk); 0x1 – частота шины АНВ (hclk); 0x2 – частота внешнего осциллятора OSC32M; 0x3 – частота встроенного осциллятора HSI32M; 0x4 – частота внешнего осциллятора OSC32K; 0x5 – частота встроенного осциллятора LSI32K.	RW
Зарезервировано	8	Зарезервировано		

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MUX_TIM32_2_TIM2	7	0x00	Выбор асинхронного тактового сигнала для входа TIM2 Timer32_2: 0 – OSC32K; 1 – LSI32K;	RW
MUX_TIM32_2_TIM1	6	0x00	Выбор синхронного тактового сигнала для входа TIM1 Timer32_2: 0 – sys_clk; 1 – hclk;	RW
Зарезервировано	5	Зарезервировано		
MUX_TIM32_1_TIM2	4	0x00	Выбор асинхронного тактового сигнала для входа TIM2 Timer32_1: 0 – OSC32K; 1 – LSI32K;	RW
MUX_TIM32_1_TIM1	3	0x00	Выбор синхронного тактового сигнала для входа TIM1 Timer32_1: 0 – sys_clk; 1 – hclk;	RW
Зарезервировано	2	Зарезервировано		
MUX_TIM32_0_TIM2	1	0x00	Выбор асинхронного тактового сигнала для входа TIM2 Timer32_0: 0 – OSC32K; 1 – LSI32K;	RW
MUX_TIM32_0_TIM1	0	0x00	Выбор синхронного тактового сигнала для входа TIM1 Timer32_0: 0 – sys_clk; 1 – hclk;	RW

Таблица 173 – Регистр настройки масок прерываний монитора частоты FREQ_MASK

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MASK_OSC32M	3	0x00	Разрешение прерывания при пропадании частоты внешнего осциллятора 32 МГц	RW
MASK_HSI32M	2	0x00	Разрешение прерывания при пропадании частоты HSI32M	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MASK_OSC32K	1	0x00	Разрешение прерывания при пропадании частоты внешнего осциллятора 32 кГц	RW
MASK_LSI32K	0	0x00	Разрешение прерывания при пропадании частоты LSI32K	RW

Таблица 174 – Регистр статуса монитора частоты FREQ_STATUS

Обозначение	Разряды	Начальное значение	Назначение	Доступ
MASK_OSC32M	3	0x00	Статус частоты внешнего осциллятора 32 МГц	RO
MASK_HSI32M	2	0x00	Статус частоты HSI32M	RO
MASK_OSC32K	1	0x00	Статус частоты внешнего осциллятора 32 кГц	RO
MASK_LSI32K	0	0x00	Статус частоты LSI32K	RO

5.2 Блок WakeUp

Блок WakeUp обеспечивает управление питанием системным доменом, управление источниками тактовых сигналов. Регистры блока WakeUp представлены в таблице 175.

Таблица 175 – Регистры блока WakeUp

Обозначение	Смещение	Начальное значение	Назначение	Доступ
SYS_MASK	0x00	0x10	Маски событий для включения и сброса системного домена (таблица 176)	RW
SYS_LEVEL	0x04	0x03	Активные уровни событий для включения и сброса системного домена (таблица 177)	RW
SYS_POWEROFF	0x08	0x00	Запись в регистр «1» отключает питание системного домена	RW1S
POWER_SWITCH	0x0C	0x00	Регистр управления и статуса схемы слежения за питанием (таблица 178)	RW
CLOCKS_BU	0x10	0x200	Регистр управления тактированием батарейного домена (таблица Таблица 179)	RW
CLOCKS_SYS	0x14	0x200	Регистр управления тактированием системного домена (таблица Таблица 180)	RW
RTC_CONTROL	0x18	0x00	Сброс внутренних счетчиков RTC (текущие часы, минуты, секунды), внутренний флаг признака срабатывания будильника alarm_pad, флаг признака будильника ALRM. Сброс RTC происходит при записи «0»	RW
STOP	0x1C	0x00	Переход в режим «Стоп». Осуществляется записью «1» Отключает тактирования системной шины.	W1S

Таблица 176 – Регистр масок событий для включения и сброса системного домена **SYS_MASK**

Обозначение	Разряды	Начальное значение	Назначение	Доступ
BU_RST_BOR	5	0x01	Запрещение формирования сброса батарейного домена при появлении сигнала BOR	RW
SYS_RST_BOR	4	0x01	Запрещение формирования сброса системного домена при появлении сигнала BOR	RW
SYS_RST_PS	3	0x00	Запрещение формирования сброса системного домена при снижении питания (индикация от схемы слежения)	RW
SYS_RST_LDO	2	0x00	Запрещение формирования сброса системного домена при снижении питания (индикация от LDO_SYS)	RW
SYS_UP_WU	1	0x00	Разрешение включения системного домена (из режима стоп или ожидания) при активном уровне внешнего вывода EXT_WU	RW
SYS_UP_RTC	0	0x00	Разрешение включения системного домена (из режима стоп или ожидания) при срабатывании будильника	RW

Таблица 177 – Регистр уровней событий для включения и сброса системного домена **SYS_LEVEL**

Обозначение	Разряды	Начальное значение	Назначение	Доступ
LVL_WU	1	0x01	Активный уровень внешнего вывода EXT_WU	RW
LVL_RTC	0	0x01	функциональной нагрузки не несет. Для работы должен быть установлен в 1	RW

Таблица 178 – Регистр управления и статуса схемы слежения за питанием **POWER_SWITCH**

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:3		Зарезервировано	
BATT_GOOD	2	0x0	Флаг состояние резервного (батарейного) источника питания	RO

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CONTROL	1	0x0	Выбор источника при принудительном переключении (при En = «1») 0 – VCC 1 – VBAT	RW
EN	0	0x0	Разрешение принудительного переключения на один из источников питания	RW

Таблица 179 - Регистр управления тактированием батарейного домена CLOCKS_BU

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:15		Зарезервировано	
OSC32K_SM	14	0x0	Режим повышенного потребления, активный уровень «0» для OSC32K. Используется для более быстрого и стабильного запуска генератора, повышает устойчивость генерации на предельных режимах эксплуатации. После запуска допускается установить значение 1.	RW
RTC_CLK_MUX	11:10	0x0	Выбор приоритетного источника тактирования часов реального времени: 0x0 – автоматический выбор. При наличии обоих источников 32кГц выбирается внутренний LSI32K; 0x1 – внутренний LSI32K; 0x2 – внешний осциллятор OSC32K	RW
ADJ_LSI32K	9:6	0x80	Поправочные коэффициенты LSI32K	RW
Зарезервировано	5:2		Зарезервировано	
LSI32K_EN	1	0x0	Включение/отключение LSI32K 0 – включение; 1 – отключение	RW

Обозначение	Разряды	Начальное значение	Назначение	Доступ
OSC32K_EN	0	0x0	Включение/отключение внешнего осциллятора на 32 КГц 0 – включение; 1 – отключение	RW

Таблица 180 - Регистр управления тактированием системного домена CLOCKS_SYS

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:12		Зарезервировано	
FORCE_32K_CLK	11:10	0x0	Принудительное переключение на опорный источник для монитора частоты 0b00 и 0b11 – автоматический выбор; 0b01 – принудительно выбран LSI32K; 0b10 – принудительно выбран OSC32K.	RW
ADJ_HSI32M	9:2	0x80	Поправочные коэффициенты HSI32M	RW
HSI32M_EN	1	0x0	Включение/отключение HSI32M 0 – включение; 1 – отключение	RW
OSC32M_EN	0	0x0	Включение/отключение внешнего осциллятора на 32 МГц 0 – включение; 1 – отключение	RW

5.3 Монитор питания

Монитор питания формирует флаги при повышении или понижении питания относительно заданного порогового значения. Пороговые значения могут быть запрограммированы в диапазоне $\pm (4 \div 12) \%$ от номинального значения. Шаг установки пороговых значений монитора напряжения питания составляет 1/16 диапазона контроля. Точность установки пороговых значений – $1/2$ шага. Схема монитора питания работает в

широком диапазоне напряжений питания – от 1,584 В до 3,696 В в диапазоне температур от -45 °С до +125 °С.

Напряжение опорного источника (внутреннего или внешнего) буферизируется и делится с помощью резистивной матрицы формирователя пороговых напряжений. Выходы дешифратора управляют ключами, выбирая одно из 16-ти контрольных напряжений в каждом из диапазонов отдельно. Схема выбора диапазона контролируемых напряжений питания выбирает, какой из резистивных делителей будет определять выходное напряжение в верхнем и нижнем диапазоне контролируемых напряжений питания. Далее компараторы сравнивают выбранные значения делителя матрицы формирователя пороговых напряжений со значениями выхода делителя питания. На выходе компараторов стоит схема фильтрации импульсов длительностью менее 1 мкс и логика управления сбросом NRESETO/NRESETU.

При включении режима самотестирования на опорный вход компаратора подается напряжение, соответствующее восьмому из 16-ти пороговых значений монитора питания (как для верхнего, так и для нижнего контролируемого диапазона). Выбор соответствующих значений OVER_THRESH и UNDER_THRESH позволяет не только проверить работу компаратора, но и оценить значение смещения на его входах.

Вычисление пороговых значений монитора выполняются по формулам:

$$\text{OUT_OVER} = \text{vsup} \cdot 1,04 + \text{vstep} * \text{OVER_THRESH},$$

$$\text{OUT_UNDER} = \text{vsup} \cdot 0,96 - \text{vstep} * \text{UNDER_THRESH},$$

где vstep – Шаг изменения порогового значения;

vsup – Напряжение питание VCC (3,3 В).

Рассчитанные значения для напряжения питания 3,3 В представлены в таблице 181.

Таблица 181 – Выбор значений срабатывания монитора питания

Код OVER_THRESH/UNDER_THRESH	Уровень OUT_UNDER	Уровень OUT_OVER
0b0000	3,168	3,432
0b0001	3,1504	3,4496
0b0010	3,1328	3,4672
0b0011	3,1152	3,4848
0b0100	3,0976	3,5024

Код OVER_THRESH/UNDER_THRESH	Уровень OUT_UNDER	Уровень OUT_OVER
0b0101	3,08	3,52
0b0110	3,0624	3,5376
0b0111	3,0448	3,5552

5.3.1 Регистры управления

Для управления монитором напряжения AVCC используются регистры блока управления аналоговой подсистемой. Состав регистров управления монитором напряжения AVCC представлен в таблице 182.

Для управления монитором напряжения VCC используются регистры PVD_CONTROL. Состав регистров управления монитором напряжения VCC представлен в таблице 183.

В таблицах 184 – 186 представлены описания полей регистров мониторов.

Таблица 182 – Регистры управления монитора напряжения AVCC

Обозначение	Смещение	Начальное значение	Назначение	Доступ
PVD_DPF_VALUE	0x10	0x5DC	Настройка цифрового фильтра	RW
PVD_CFG	0x14	0x0	Регистр настроек	RW
PVD_STATUS	0x18	0x0	Регистр статуса	RO

Примечание – Смещение относительно стартового адреса блока управления аналоговой подсистемой (0x00085000 в адресном пространстве микроконтроллера).

Таблица 183 – Регистры управления монитора напряжения VCC

Обозначение	Смещение	Начальное значение	Назначение	Доступ
DPF_VALUE	0x0	0x5DC	Настройка цифрового фильтра	RW
CONFIG	0x04	0x0	Регистр настроек	RW
STATUS	0x08	0x0	Регистр статуса	RO

Примечание – Смещение относительно стартового адреса PVD_CONTROL (0x000051800 в адресном пространстве микроконтроллера).

Таблица 184 – Регистр DPF_VALUE монитора напряжения

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:16		Зарезервировано	
DPF	7:0	220	Коэффициент, определяющий длительность отфильтровываемых импульсов (на системной частоте)	RW

Таблица 185 – Регистр CONFIG монитора напряжения

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:13		Зарезервировано	
EN_VREFCLB	12	0x0	Использование в качестве опорного источника напряжения калибруемый ИОН: 1 – ИОН используется; 0 – ИОН не используется.	RW
OVER_THRESH	11:8	0x0	Верхний порог срабатывания монитора	RW
UNDER_THRESH	7:4	0x0	Нижний порог срабатывания монитора	RW
TESTMODE	3	0x0	Переход в тестовый режим	RW
NRESETO	2	0x0	Отключение (сброс) детектирования верхнего порога. 1 – нормальная работа; 0 – детектирование не выполняется, схема в состоянии сброса	RW
NRESETU	1	0x0	Отключение (сброс) детектирования нижнего порога. 1 – нормальная работа; 0 – детектирование не выполняется, схема в состоянии сброса	RW
PD	0	0x1	Управление питанием монитором: 1 – монитор выключен; 0 – монитор включен	RW

Таблица 186 – Регистр STATUS монитора напряжения

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:4		Зарезервировано	
OVER_FLAG	3	0x0	Текущее состояние линии монитора верхнего порога с учетом цифровой фильтрации	RO
UNDER_FLAG	2	0x0	Текущее состояние линии монитора нижнего порога с учетом цифровой фильтрации	RO
OVER_VALUE	1	0x0	Текущее состояние линии монитора верхнего порога без цифровой фильтрации	RO
UNDER_VALUE	0	0x0	Текущее состояние линии монитора нижнего порога без цифровой фильтрации	RO

5.4 Схема формирования сигналов POR и BOR

Схема формирования сигналов POR (Power on Reset) и BOR (Brownout Reset) обеспечивает удержание микросхемы в состоянии сброса при подаче питания VCC на микросхему на время нарастания напряжения питания до порогового значения включения и в случае снижения напряжения питания ниже порогового значения выключения (сигнал POR), а также формирования флага при просадке питания VCC ниже определенного уровня (сигнал BOR).

При нарастании напряжения питания VCC схема выдает на выход сигнал сброса POR (активным является уровень логического «0») до установления напряжения на VCC выше значения vpor.

Стоит отметить, что сигнал porl схемы формирования POR используется для сброса цифровой части схемы, которая питается от шины VDD. Напряжение на VDD формируется встроенным регулятором. При включении внешнего питания может сложиться ситуация, когда напряжение на VCC уже превышает уровень vpor, сигнал porl сбрасывается в пассивное состояние (уровень логической «1»), а питание vdd на цифровой части еще отсутствует. В этом случае цифровая часть не будет сброшена. Поэтому, схема формирования сигнала porl отслеживает как уровень напряжения на шине VCC, так и уровень напряжения на шине VDD.

Если значение напряжения питания VCC опускается ниже значения vpdr, блок выдает на выход сигнал сброса, пока напряжение питания снова не станет выше vpor.

Поведение схемы формирования сигнала BOR (активным является уровень логического «0») аналогично поведению схемы формирования сигнала POR, но уровни срабатывания v_{bor} и v_{bdr} находятся выше, чем в схеме формирования POR (см. таблицу 1).

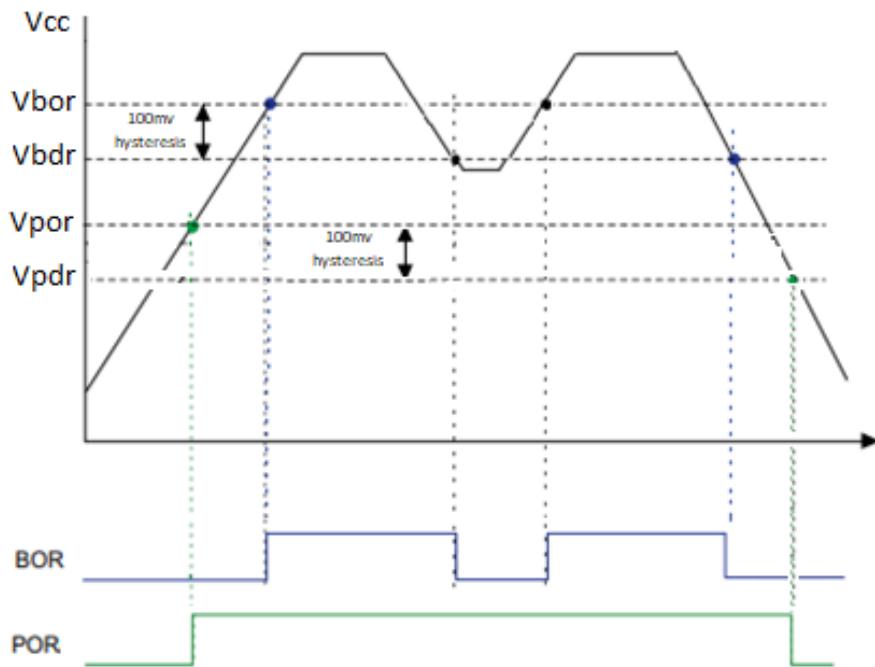


Рисунок 102 – Упрощенная диаграмма работы схемы формирования сигналов POR и BOR (влияние напряжения на шине VDD не показано).

Таблица 187 – Основные электрические характеристики схемы формирования сигналов POR и BOR

Параметр	Обозначение	Мин.	Тип.	Макс.	Единицы измерения	Примечания
Уровень напряжения перехода сигнала POR в активное состояние	v_{pdr}	2,37	2,4	2,43	В	
Уровень напряжения перехода сигнала POR в пассивное состояние	v_{por}	2,47	2,5	2,53	В	
Уровень напряжения перехода сигнала BOR в активное состояние	v_{bdr}	2,62	2,65	2,68	В	
Уровень напряжения перехода сигнала BOR в пассивное состояние	v_{bor}	2,72	2,75	2,78	В	
Уровень перехода сигнала слежения за питанием VDD в активное состояние	v_{pdr_vdd}	1,18	1,2	1,22	В	

с учетом разброса источника опорного напряжения ±1 %

Уровень перехода сигнала слежения за питанием VDD в пассивное состояние	vpor_vdd	1,23	1,25	1,27	В	
Задержка срабатывания схемы по выходу porh из «0» в «1»	Tporh01	35	68	115	нс	
Задержка срабатывания схемы по выходу porh из «1» в «0»	Tporh10	135	234	364	нс	
Задержка срабатывания схемы по выходу borh из «0» в «1»	Tborh01	35	68	115	нс	
Задержка срабатывания схемы по выходу borh из «1» в «0»	Tborh10	135	234	364	нс	
Задержка срабатывания схемы по выходу porl из «0» в «1»	Tporl01	31	62	110	нс	с учетом разброса источника опорного напряжения ±1 %; превышение над порогом 100 мВ и емкостью нагрузки 1 пФ
Задержка срабатывания схемы по выходу porl из «1» в «0»	Tporl10	161	273	427	нс	
Задержка срабатывания схемы по выходу borl из «0» в «1»	Tborl01	30	61	106	нс	
Задержка срабатывания схемы по выходу borl из «1» в «0»	Tborl10	160	270	425	нс	
Ток потребления по цепи VCC	Isup_vcc	15	23	40	мкА	с учетом разброса источника опорного напряжения ±1 %
Ток потребления по цепи VDD	Isup_vdd	-	-	< 1	нА	

5.5 Схема слежения и переключения источника питания

Схема слежения и переключения источника внешнего питания (VoltageMonitor) предназначена для отслеживания состояния напряжения основного питания и, при необходимости, переключения на резервное (батарейное) питание.

Основными функциональными характеристиками схемы являются:

- отслеживания состояния напряжения основного питания и, при необходимости, автоматическое переключения на резервное (батарейное) питание;
- возможность принудительного переключения на один из источников питания;
- отслеживание состояния резервного (батарейного) питания и формирования соответствующего флага.

Алгоритм работы схемы слежения и переключения источника внешнего питания, следующий:

- если основное питание снижается до уровня $(3,3 \times 0,9 - 0,02)$ В включается режим сравнения уровней напряжения основного и резервного источников питания;
- если в режиме сравнения напряжение основного источника более чем на 20 мВ ниже напряжения резервного источника, то схема переключается на работу от резервного источника;
- когда напряжение основного источника снова превысит напряжение резервного более чем на 20 мВ, либо будет больше, чем $(3,3 \times 0,9 + 0,02)$ В, схема переключится на работу от основного источника питания.

Управление модулем осуществляется через регистр POWER_SWITCH модуля Wakeup (смещение 0x0C относительно базового адреса Wakeup). Назначение полей регистра представлено в таблице 188.

Таблица 188 – Регистр POWER_SWITCH модуля Wakeup

Обозначение	Разряды	Начальное значение	Назначение	Доступ
Зарезервировано	31:3		Зарезервировано	
BATT_GOOD	2	0x0	Флаг состояние резервного (батарейного) источника питания	RO

Обозначение	Разряды	Начальное значение	Назначение	Доступ
CONTROL	1	0x0	Выбор источника при принудительном переключении (при En = «1») 0 – VCC 1 – VBAT	RW
EN	0	0x0	Разрешение принудительного переключения на один из источников питания	RW

Основные электрические характеристики представлены в таблице 189.

Таблица 189 – Основные электрические характеристики схемы слежения

Параметр, единица измерения	Обозначение	Значение параметра		
		Мин.	Тип.	Макс.
Уровень VCC, при котором включается схема сравнения VCC и VBAT, В	Ucc_cmp_on	2,945	2,95	2,955
Уровень VCC, при котором выключается схема сравнения VCC и VBAT, В	Ucc_cmp_off	2,985	2,99	2,995
Относительный уровень VCC, при котором происходит переключение на резервный источник	vcc2vbat	VBAT – 25мВ	VBAT – 20мВ	VBAT – 15мВ
Относительный уровень VCC, при котором происходит обратное переключение на основной источник	vbat2vcc	VBAT + 15мВ	VBAT + 20мВ	VBAT + 25мВ
Уровень переключения из «1» в «0» схемы формирования сигнала о состоянии батареи, В	Vbgo10	2,78	2,8	2,82
Уровень переключения из «0» в «1» схемы формирования сигнала о состоянии батареи, В	Vbgo01	2,84	2,86	2,88
Задержка между спадом VCC ниже уровня vcc2vbat и переключением на резервный источник питания, нс	Tvcc2vbat	370	500	720
Задержка между подъемом VCC выше уровня vbat2vcc и переключением на основной источник питания, нс	Tvbat2vcc	50	80	120
Задержка между сигналом fpwr и переключением на резервный источник питания, нс	Tvcc2vbat_fpwr	6	7	9

Параметр, единица измерения	Обозначение	Значение параметра		
		Мин.	Тип.	Макс.
Задержка между сигналом fpwr и переключением на основной источник питания, нс	Tvbat2vcc_fpwr	10	12	14

Структурная схемы слежения и переключения источника внешнего питания представлена на рисунке 103.

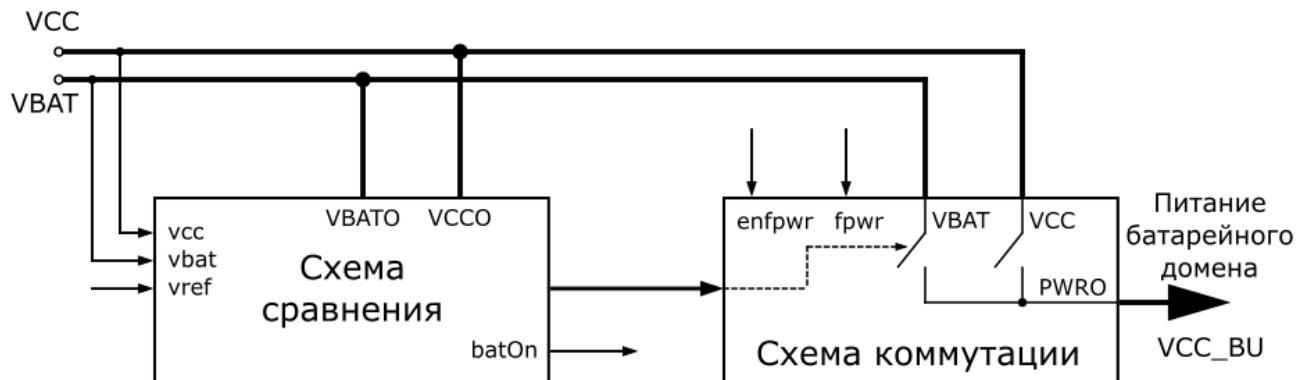


Рисунок 103 – Структурная схема слежения и переключения источника внешнего питания

Схема сравнения построена на двух компараторах. Первый компаратор отслеживает событие, когда напряжение основного источника питания (VCC) вышло за нижнюю границу рабочего диапазона, и при его наступлении включает режим сравнения напряжений основного и резервного (VBAT) источников питания. Сравнение осуществляется схемой второго компаратора. Сравнение осуществляется схемой второго компаратора. Ширина петли гистерезиса каждого из компараторов составляет 40 мВ.

5.6 Линейный регулятор напряжения питания батарейного домена

Схема линейного регулятора напряжения (LDO_LP) предназначена для формирования стабильного выходного напряжения при нестабильном входном напряжении.

Основными функциональными характеристиками регулятора напряжения являются:

- формирование стабильного выходного напряжения при нестабильном входном напряжении;
- формирования флагов, сигнализирующих о состоянии выходного напряжения.

Структурная схема регулятора напряжения выполнена по схеме класса АВ на основе усилителя ошибки регулирования, проходного транзистора Р-типа, разрядного транзистора Н-типа, резистивного делителя и схемы формирования сигнала о состоянии питания на основе компаратора. Для обеспечения устойчивости работы регулятора необходимо между выводом VDD и «землей» подключить корректирующий электролитический конденсатор номиналом $\sim 2,2$ мкФ ($ESR \sim 1 \pm 0,5$ Ом) параллельно с керамическим конденсатором ~ 100 нФ.

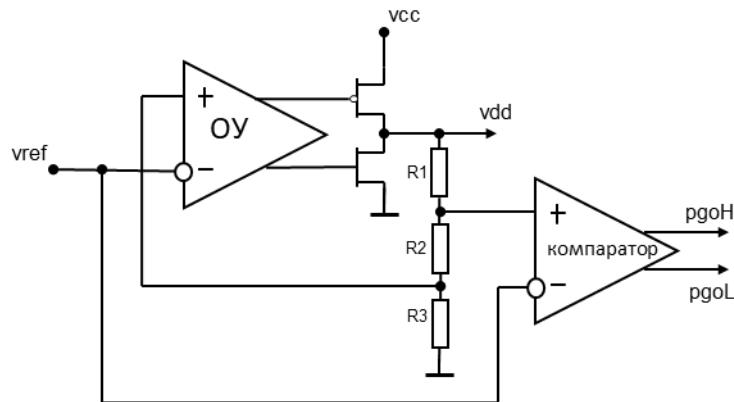


Рисунок 104 – Структурная схема регулятора напряжения

Основные электрические характеристики приведены в таблице 190.

Таблица 190 – Основные электрические характеристики регулятора

Параметр, единица измерения	Обозначение	Значение параметра		
		Мин.	Тип.	Макс.
Выходное (регулируемое) напряжение, В	vdd	1,77	1,79	1,8
Ток потребления, мА	Icc	21	29	44
Ток потребления в выключенном режиме, нА	Iqc	–	–	25
Ток нагрузки, мА	Idd	0	3	4,5
Уровень переключения из «1» в «0» схемы формирования сигнала о состоянии питания VDD, В	Upg10	1,32	1,35	1,38
Уровень переключения из «0» в «1» схемы формирования сигнала о состоянии питания VDD, В	Upg01	1,42	1,45	1,48
Коэффициент подавления помех по питанию, дБ	PSSR	40	45	52

5.7 Основной линейный регулятор напряжения

Схема линейного регулятора напряжения (LDO) предназначена для формирования стабильного выходного напряжения при нестабильном входном напряжении в широком диапазоне токов нагрузки.

Основными функциональными характеристиками регулятора напряжения являются:

- формирование стабильного выходного напряжения в широком диапазоне токов нагрузки;
- формирования флагов, сигнализирующих о состоянии выходного напряжения;
- формирования флага, сигнализирующего о переходе регулятора в режим ограничения по выходному току;
- возможность внешнего выключения регулятора в целях экономии потребления.

Основные электрические характеристики приведены в таблице 191.

Таблица 191 – Основные электрические характеристики регулятора

Параметр, единица измерения	Обозначение	Значение параметра		
		Мин.	Тип.	Макс.
Входное напряжение, В	vcc	2,97	3,3	3,63
Выходное (регулируемое) напряжение, В	vdd	1,78	1,8	1,85
Ток потребления, мкА	Icc	63	84	125
Ток потребления в выключенном режиме, нА	Iqc	—	—	500
Ток нагрузки, мА	Idd	0	60	75
Ток нагрузки срабатывания схемы защиты, мА	Idd_lim	50	60	75
Уровень переключения из «1» в «0» схемы формирования сигнала о состоянии питания VDD, В	Upg10	1,32	1,35	1,38
Уровень переключения из «0» в «1» схемы формирования сигнала о состоянии питания VDD, В	Upg01	1,42	1,45	1,48
Коэффициент подавления помех по питанию, дБ	PSSR	51	53	56

Структурная схема регулятора напряжения представлена на рисунке 105. Регулятор напряжения выполнен по традиционной схеме на основе усилителя ошибки регулирования, проходного транзистора, резистивного делителя и схемы формирования сигнала о состоянии питания на основе компаратора. Добавлена пороговая схема ограничения тока нагрузки и схема формирования сигнала о состоянии питания. При превышении порогового значения тока нагрузки схема переходит в режим ограничения по выходному току и вырабатывается сигнал на выходе «cov». Для обеспечения устойчивости работы регулятора необходимо между выводом VDD и «землей» подключить корректирующий электролитический конденсатор номиналом $\sim 2,2$ мкФ ($ESR \sim 1\pm0,5$ Ом) параллельно с керамическим конденсатором ~ 100 нФ.

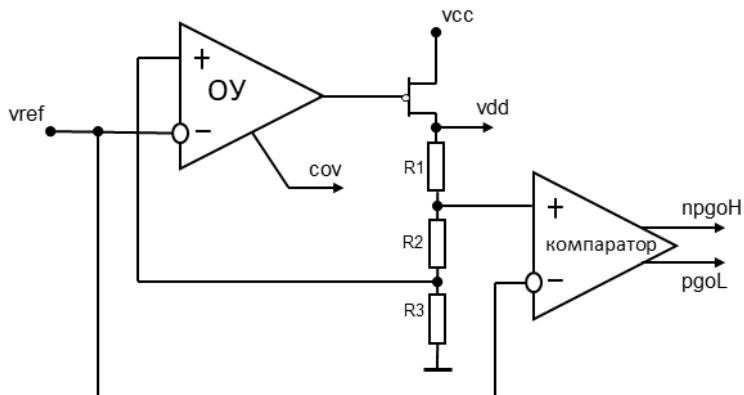


Рисунок 105 – Структурная схема регулятора напряжения

5.8 Калибруемый источник опорного напряжения

Схема калибруемого источника опорного напряжения формирует стабильное выходное напряжение в широком диапазоне температур. Схема источника опорного напряжения построена на использовании ширины запрещенной зоны полупроводника. Особенностью данной схемы является наличие блока калибровки, с помощью которого можно подстраивать значение выходного напряжения опорного источника в диапазоне ± 105 мВ с шагом ~ 15 мВ по отношению к расчетному и оптимальному с точки зрения температурной зависимости значению. Код «0b1000» соответствует расчетному и оптимальному с точки зрения температурной зависимости значению выходного напряжения опорного источника (1,218 В), код «0b0000» соответствует минимальному значению выходного напряжения, код «0b1111» соответствует максимальному значению выходного напряжения. Включение калибруемого источника опорного напряжения

осуществляется записью «1» в поле CLB_EN регистра управления аналоговой подсистемы REF_CLB (установка этого бита в «1» также включает калируемый источник тока). Настройка коэффициентов выполняется через поля COEF_REFVCLB этого же регистра.

5.9 Калируемый источник опорного тока

Схема источника опорного тока формирует выходной ток, слабо зависящий от температуры с возможностью перевода схемы в выключенное состояние в целях снижения энергопотребления.

Схема калируемого источника опорного тока формирует ток, зависящий от входного опорного напряжения по следующей формуле:

$$I_{\text{вых.}} = V_{\text{вх.}} / R,$$

где $V_{\text{вх.}}$ – независящее от температуры входное опорное напряжение,

R – сопротивление с минимальным коэффициентом зависимости от температуры (в данном случае поликремний без салицида),

$I_{\text{вых}}$ – выходной опорный ток.

Особенностью данной схемы является наличие блока калибровки с помощью которого можно подстраивать выходное значение тока опорного источника в диапазоне $\pm 0,6 \text{ мкА}$ с шагом $\sim 0,08 \text{ мкА}$ по отношению к среднему значению при значении кода подстройки «0b1000» источника опорного напряжения refVclb. Код «0b1000» соответствует среднему значению выходного тока опорного источника ($2,1 \text{ мкА}$), код «0b0000» соответствует максимальному значению выходного тока, код «0b1111» соответствует минимальному значению выходного тока. Включение калируемого источника опорного тока осуществляется записью «1» в поле CLB_EN регистра управления аналоговой подсистемы REF_CLB (установка этого бита в «1» также включает калируемый источник напряжения). Настройка коэффициентов выполняется через поля COEF_REFICLB этого же регистра.

Назначение выводов, представлено в таблице выводов 192.

6 Назначение выводов

Таблица 192 - Назначение выводов

№ вывода корпуса	Обозначение вывода	Описание	PADx_CFG = 0			PADx_CFG = 1			PADx_CFG = 2			PADx_CFG = 3 Аналоговая функция
			Назначение	Тип	Описание	Назначение	Тип	Описание	Назначение	Тип	Описание	
1	PORT_1_12	Многофункциональный вывод. Порт 1, разряд 12	GPIO1_12	Вход-выход	Порт общего назначения 1. Разряд 12	I2C1_sda	Вход-выход	Интерфейс I2C 1, линия данных	UART0_ndtr	Выход	Интерфейс UART_0. Готовность данных приемника	DAC0
2	PORT_1_11	Многофункциональный вывод. Порт 1, разряд 11	GPIO1_11	Вход-выход	Порт общего назначения 1. Разряд 11	UART1_nrts	Выход	Интерфейс UART 1, запрос на передачу				REF_ADC_DAC
3	PORT_1_10	Многофункциональный вывод. Порт 1, разряд 10	GPIO1_10	Вход-выход	Порт общего назначения 1. Разряд 10	UART1_ncts	Вход	Интерфейс UART 1, готовность передачи				
4	PORT_1_9	Многофункциональный вывод. Порт 1, разряд 9	GPIO1_9	Вход-выход	Порт общего назначения 1. Разряд 9	UART1_txd	Выход	Интерфейс UART 1, данные для передачи				
5	VCC	Основное питание 3.3В										
6	VDD	Выход системного LDO для подключения конденсаторов										
7	PORT_1_8	Многофункциональный вывод. Порт 1, разряд 8	GPIO1_8	Вход-выход	Порт общего назначения 1. Разряд 8	UART1_rxd	Вход	Интерфейс UART 1, принимаемые данные				
8	VBAT	Запасное питание батарейного домена 3.3В										
9	VSS	Общий вывод										
10	VCC_BU	Выход внутренней линии питания батарейного домена 3.3В										
11	VDD_BU	Выход батарейного LDO для подключения конденсаторов										
12	OSC32K_XO	Выход осциллятора 32кГц (3.3В)										
13	OSC32K_XI	Вход осциллятора 32кГц (3.3В)										
14	RST#	Внешний сброс (активный уровень 0)										

15	EXT_WU	Выход из режима пониженного потребления										
16	RTC_ALARM	Будильник (активный уровень 1)										
17	BOOT_0	Вход управления устройством загрузки										
18	BOOT_1	Вход управления устройством загрузки										
19	VPRG	Напряжение программирования ОТР 8.0В										
20	VCC	Основное питание 3.3В										
21	PORT_1_7	Многофункциональный вывод. Порт 1, разряд 7	GPIO1_7	Вход-выход	Порт общего назначения 1. Разряд 7	SPI1_n_ss_out_3	Выход	Интерфейс SPI 1, сигнал выбора ведомого 3	digital_test_pad	Выход	Выход цифровой подсистемы тестирования аналоговых блоков	ADC_1
22	PORT_1_6	Многофункциональный вывод. Порт 1, разряд 6	GPIO1_6	Вход-выход	Порт общего назначения 1. Разряд 6	SPI1_n_ss_out_2	Выход	Интерфейс SPI 1, сигнал выбора ведомого 2				
23	PORT_1_5	Многофункциональный вывод. Порт 1, разряд 5	GPIO1_5	Вход-выход	Порт общего назначения 1. Разряд 5	SPI1_n_ss_out_1	Выход	Интерфейс SPI 1, сигнал выбора ведомого 1	UART0_ck	Выход	Сигнал синхронизации для приема и передачи	ADC_0
24	VDD	Выход системного LDO для подключения конденсаторов										
25	PORT_1_4	Многофункциональный вывод. Порт 1, разряд 4	GPIO1_4	Вход-выход	Порт общего назначения 1. Разряд 4	SPI1_n_ss_out_0	Выход	Интерфейс SPI 0, сигнал выбора ведомого 0	Timer32_2_Tx	Вход	32х-разрядный таймер2. Внешний вход тактирования	
26	PORT_1_3	Многофункциональный вывод. Порт 1, разряд 3	GPIO1_3	Вход-выход	Порт общего назначения 1. Разряд 3	SPI1_n_ss_in	Вход	Интерфейс SPI 1, сигнал выбора в режиме ведомого	Timer32_2_ch4	Вход-выход	32х-разрядный таймер2. Каналы 1-3. Выход-ШИМ, Вход - захват/сравнение	
27	PORT_1_2	Многофункциональный вывод. Порт 1, разряд 2	GPIO1_2	Вход-выход	Порт общего назначения 1. Разряд 2	SPI1_clk	Вход-выход	Интерфейс SPI 0, тактовый сигнал	Timer32_2_ch3	Вход-выход		
28	PORT_1_1	Многофункциональный вывод. Порт 1, разряд 1	GPIO1_1	Вход-выход	Порт общего назначения 1. Разряд 1	SPI1_mosi	Вход-выход	Интерфейс SPI 1, линия данных от	Timer32_2_ch2	Вход-выход		

								ведущего к ведомому				
29	VSS	Общий вывод										
30	PORT_1_0	Многофункциональный вывод. Порт 1, разряд 0	GPIO1_0	Вход-выход	Порт общего назначения 1. Разряд 0	SPI1_miso	Вход-выход	Интерфейс SPI 1, линия данных от ведомого к ведущему	Timer32_2_ch 1	Вход-выход	32х-разрядный таймер2. Канал 4	
31	PORT_2_7	Многофункциональный вывод. Порт 2, разряд 7	GPIO2_7	Вход-выход	Порт общего назначения 2. Разряд 7							
32	PORT_2_6	Многофункциональный вывод. Порт 2, разряд 6	GPIO2_6	Вход-выход	Порт общего назначения 2. Разряд 6	SPI0_n_ss_out_3	Выход	Интерфейс SPI 0, сигнал выбора ведомого 3	UART1_ck	Выход	Сигнал синхронизации для приема и передачи	
33	PORT_2_5	Многофункциональный вывод. Порт 2, разряд 5	GPIO2_5	Вход-выход	Порт общего назначения 2. Разряд 5	SPIFI_DATA_3	Вход-выход	Интерфейс SPIFI, шина данных, разряд 3				
34	PORT_2_4	Многофункциональный вывод. Порт 2, разряд 4	GPIO2_4	Вход-выход	Порт общего назначения 2. Разряд 4	SPIFI_DATA_2	Вход-выход	Интерфейс SPIFI, шина данных, разряд 2				
35	PORT_2_3	Многофункциональный вывод. Порт 2, разряд 3	GPIO2_3	Вход-выход	Порт общего назначения 2. Разряд 3	SPIFI_DATA_1	Вход-выход	Интерфейс SPIFI, шина данных, разряд 1	UART1_nri	Вход	Звонок (вызов) на телефонной линии	
36	PORT_2_2	Многофункциональный вывод. Порт 2, разряд 2	GPIO2_2	Вход-выход	Порт общего назначения 2. Разряд 2	SPIFI_DATA_0	Вход-выход	Интерфейс SPIFI, шина данных, разряд 0	UART1_ndsr	Вход	Сигнал готовности источника данных	
37	PORT_2_1	Многофункциональный вывод. Порт 2, разряд 1	GPIO2_1	Вход-выход	Порт общего назначения 2. Разряд 1	SPIFI_CS	Выход	Интерфейс SPIFI, сигнал CS	UART1_ndcd	Вход	Сигнал обнаружения несущей	
38	PORT_2_0	Многофункциональный вывод. Порт 2, разряд 0	GPIO2_0	Вход-выход	Порт общего назначения 2. Разряд 0	SPIFI_SCLK	Вход-выход	Интерфейс SPIFI, тактовый сигнал	UART1_ndtr	Выход	Готовность приемника данных	
39	OSC32M_XO	Выход осциллятора 32МГц (3.3В)										
40	OSC32M_XI	Вход осциллятора 32МГц (3.3В)										
41	VDD	Выход системного LDO для подключения конденсаторов										
42	PORT_0_15	Многофункциональный вывод. Порт 0, разряд 15	GPIO0_15	Вход-выход	Порт общего назначения 0. Разряд 15	TDO	Выход	Интерфейс JTAG. Выход данных				

43	PORT_0_14	Многофункциональный вывод. Порт 0, разряд 14	GPIO0_14	Вход-выход	Порт общего назначения 0. Разряд 14	TRSTn	Вход	Интерфейс JTAG. Сброс (активный уровень 0)				
44	PORT_0_13	Многофункциональный вывод. Порт 0, разряд 13	GPIO0_13	Вход-выход	Порт общего назначения 0. Разряд 13	TMS	Вход	Интерфейс JTAG. Сигнала управления	Timer16_2_out	Выход	16-разрядный таймер 2.Выход	ADC_7
45	PORT_0_12	Многофункциональный вывод. Порт 0, разряд 12	GPIO0_12	Вход-выход	Порт общего назначения 0. Разряд 12	TCK	Вход	Интерфейс JTAG. Сигнал тактирования	Timer16_2_in_2	Вход	16-разрядный таймер 2.Выход 2	
46	PORT_0_11	Многофункциональный вывод. Порт 0, разряд 11	GPIO0_11	Вход-выход	Порт общего назначения 0. Разряд 11	TDI	Вход	Интерфейс JTAG. Вход данных	Timer16_2_in_1	Вход	16-разрядный таймер 2.Выход 1	ADC_6
47	PORT_0_10	Многофункциональный вывод. Порт 0, разряд 10	GPIO0_10	Вход-выход	Порт общего назначения 0. Разряд 10	I2C0_scl	Вход-выход	Интерфейс I2C 0, сигнал тактирования	Timer16_1_out	Выход	16-разрядный таймер 1.Выход	
48	PORT_0_9	Многофункциональный вывод. Порт 0, разряд 9	GPIO0_9	Вход-выход	Порт общего назначения 0. Разряд 9	I2C0_sda	Вход-выход	Интерфейс I2C 0, линия данных	Timer16_1_in_2	Вход	16-разрядный таймер 1.Выход 2	ADC_5
49	PORT_0_8	Многофункциональный вывод. Порт 0, разряд 8	GPIO0_8	Вход-выход	Порт общего назначения 0. Разряд 8	UART0_nrts	Выход	Интерфейс UART 0, запрос на передачу	Timer16_1_in_1	Вход	16-разрядный таймер 1.Выход 1	
50	PORT_0_7	Многофункциональный вывод. Порт 0, разряд 7	GPIO0_7	Вход-выход	Порт общего назначения 0. Разряд 7	UART0_ncts	Вход	Интерфейс UART 0, готовность передачи	Timer16_0_out	Выход	16-разрядный таймер 0.Выход	ADC_4
51	PORT_0_6	Многофункциональный вывод. Порт 0, разряд 6	GPIO0_6	Вход-выход	Порт общего назначения 0. Разряд 6	UART0_txd	Выход	Интерфейс UART 0, данные для передачи	Timer16_0_in_2	Вход	16-разрядный таймер 0.Выход 2	
52	VCC	Основное питание 3.3В										
53	PORT_0_5	Многофункциональный вывод. Порт 0, разряд 5	GPIO0_5	Вход-выход	Порт общего назначения 0. Разряд 5	UART0_rxd	Вход	Интерфейс UART 0, принимаемые данные	Timer16_0_in_1	Вход	16-разрядный таймер 0.Выход 1	
54	PORT_0_4	Многофункциональный вывод. Порт 0, разряд 4	GPIO0_4	Вход-выход	Порт общего назначения 0. Разряд 4	SPI0_n_ss_out_0	Выход	Интерфейс SPI 0, сигнал выбора ведомого 0	Timer32_1_Tx	Вход	32х-разрядный таймер1. Внешний вход тактирования	ADC_3
55	PORT_0_3	Многофункциональный вывод. Порт 0, разряд 3	GPIO0_3	Вход-выход	Порт общего назначения 0. Разряд 3	SPI0_n_ss_in	Вход	Интерфейс SPI 0, сигнал выбора в	Timer32_1_ch_4	Вход-выход	32х-разрядный таймер1.	

								режиме ведомого			Каналы 1-4. Выход-ШИМ, Вход - захват/сравнение	
56	PORT_0_2	Многофункциональный вывод. Порт 0, разряд 2	GPIO0_2	Вход-выход	Порт общего назначения 0. Разряд 2	SPI0_clk	Вход-выход	Интерфейс SPI 0, тактовый сигнал	Timer32_1_ch 3	Вход-выход		ADC_2
57	PORT_0_1	Многофункциональный вывод. Порт 0, разряд 1	GPIO0_1	Вход-выход	Порт общего назначения 0. Разряд 1	SPI0_mosi	Вход-выход	Интерфейс SPI 0, линия данных от ведущего к ведомому	Timer32_1_ch 2	Вход-выход		
58	PORT_0_0	Многофункциональный вывод. Порт 0, разряд 0	GPIO0_0	Вход-выход	Порт общего назначения 0. Разряд 0	SPI0_miso	Вход-выход	Интерфейс SPI 0, линия данных от ведомого к ведущему	Timer32_1_ch 1	Вход-выход		REF_TST, Внешний опорный сигнал для тестирования
59	VSS	Общий вывод										
60	PORT_1_15	Многофункциональный вывод. Порт 1, разряд 15	GPIO1_15	Вход-выход	Порт общего назначения 1. Разряд 15	SPI0_n_ss_out_2	Выход	Интерфейс SPI 0, сигнал выбора ведомого 2	UART0_nri	Вход	Интерфейс UART_0. Звонок (вызов) на телефонной линии	TEST MUX OUT, Выход системы аналогового тестирования
61	PORT_1_14	Многофункциональный вывод. Порт 1, разряд 14	GPIO1_14	Вход-выход	Порт общего назначения 1. Разряд 14	SPI0_n_ss_out_1	Выход	Интерфейс SPI 0, сигнал выбора ведомого 1	UART0_ndsr	Вход	Интерфейс UART_0. Сигнал готовности источника данных	
62	AVCC	Питание аналоговых блоков 3.3В										
63	AVSS	Общий вывод аналоговых блоков										
64	PORT_1_13	Многофункциональный вывод. Порт 1, разряд 13	GPIO1_13	Вход-выход	Порт общего назначения 1. Разряд 13	I2C1_scl	Вход-выход	Интерфейс I2C 1, сигнал тактирования	UART0_ndcd	Вход	Интерфейс UART_0. Сигнал обнаружения несущей	DAC1

Список использованных сокращений

AHB	высокопроизводительна шина AMBA (AMBA High-performance Bus)
APB	шина AMBA для периферийных устройств (AMBA Peripheral Bus)
EEPROM	электрически стираемое перепрограммируемое ПЗУ (Electrically Erasable Programmable Read-Only Memory(ЭСППЗУ))
GPIO	порт ввода/вывода общего назначения (general purpose port input/output)
I2C	интерфейс межсхемной коммутации (inter-integrated circuit interface)
JTAG	интерфейс JTAG (Joint Test Action Group – название рабочей группы по разработке стандарта IEEE1149)
LDO	преобразователи напряжения с малым падением напряжения (Low drop out)
OTP	One Time Programmable –однократно программируемая память
UART	универсальный синхронно/асинхронный приёмопередатчик (universal asynchronous receiver-transmitter)
АЦП	аналогово-цифровой преобразователь
ИОН	Источник опорного напряжения
ОЗУ	оперативное запоминающее устройство (random access memory (RAM))
ПЗУ	постоянное запоминающее устройство (read-only memory (ROM))
ЦАП	цифро-аналоговый преобразователь
ШИМ (PWM)	широкото-импульсная модуляция (pulse width modulation)

Список использованных сокращений для регистров и битовых полей

RW	Read/Write (доступно на чтение, доступно на запись)
RO	Read Only (доступно на чтение, запись игнорируется)
WO	Write Only (читается 0, доступно на запись)
R/W1S	Read/Write 1 to Set (доступно на чтение, запись 1 устанавливает соответствующий бит)
R/W1C	Read/Write 1 to Clear (доступно на чтение, запись 1 удаляет соответствующий бит)
W1S	Write 1 to Set (читается 0, запись 1 устанавливает соответствующий бит)
W1C	Write 1 to Clear (читается 0, запись 1 удаляет соответствующий бит)
RC	Clear At Read