

RISC-V микроконтроллер MIK32 АМУР

Техническое описание

Содержание

| | |
|---|------------|
| Введение..... | 4 |
| 1 Назначение | 3 |
| 2 Технические характеристики | 4 |
| 3 Состав MCU32..... | 5 |
| 3.2 Коммутационная матрица АНВ..... | 11 |
| 3.3 Карта памяти..... | 11 |
| 3.4 Порядок загрузки..... | 13 |
| 3.5 Контроллер ПДП..... | 14 |
| 3.6 Блок ТСВ..... | 26 |
| 3.7 Модуль USART | 32 |
| 3.8 Модуль I2C..... | 52 |
| 3.9 Модуль 32-разрядного таймера | 52 |
| 3.10 Модуль 16-разрядного таймера | 118 |
| 3.11 Часы реального времени..... | 143 |
| 3.12 Программируемый контроллер прерываний..... | 150 |
| 3.13 Блок вычисления контрольных сумм..... | 153 |
| 3.14 Модуль SPI..... | 157 |
| 3.15 Модуль порта ввода/вывода общего назначения..... | 173 |
| 3.16 Контроллер SPIFI | 181 |
| 3.17 Крипто-блок..... | 211 |
| 3.18 Контроллер блока однократнопрограммируемой памяти OTP..... | 214 |
| 3.19 Контроллер EEPROM | 226 |
| 3.20 Модуль сторожевого таймера | 239 |
| 3.21 Сторожевой таймер шины..... | 241 |
| 3.22 Температурный сенсор | 242 |
| 3.23 Цифро-аналоговый преобразователь | 247 |
| 3.24 Аналого-цифровой преобразователь | 252 |
| 4 Подсистема тактирования..... | 261 |
| 4.2 Встроенные осцилляторы с частотой 32 МГц и 32 кГц | 266 |
| 4.3 Вывод для подключения внешнего кварцевого резонатора до 32 МГц..... | 268 |
| 4.4 Вывод для подключения внешнего кварцевого резонатора 32 кГц..... | 268 |
| 4.5 Монитор частоты системного домена..... | 268 |
| 5 Подсистема питания | 274 |
| 5.2 Блок управления питанием и тактированием..... | 277 |
| 5.3 Блок WakeUp | 285 |
| 5.4 Монитор питания | 287 |
| 5.5 Схема формирования сигналов POR и BOR | 291 |
| 5.6 Схема слежения и переключения источника питания | 293 |
| 5.7 Линейный регулятор напряжения питания батарейного домена..... | 296 |
| 5.8 Основной линейный регулятор напряжения | 297 |
| 5.9 Источники опорного напряжения..... | 299 |
| 5.10 Калибруемый источник опорного напряжения..... | 299 |
| 6 Назначение выводов..... | 301 |
| Список использованных сокращений..... | 266 |

1 Назначение

Микросхема 32х-битного микроконтроллера (далее микросхема MCU32, микросхема, микроконтроллер) предназначена для управления высокопроизводительными и многофункциональными электронными устройствами для интернета вещей, имеет развитую периферию и аппаратный криптографический блок, поддерживающий отечественные ГОСТ в области защиты информации.

Особенности (уникальность) продукции микросхемы MCU32:

- 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

2.1 Технические характеристики микросхемы 32х-битного микроконтроллера

Микросхема MCU32 имеет следующие характеристики:

- максимальная рабочая частота 32 МГц;
- объем встроенной памяти ОЗУ 16 Кбайт;
- объем встроенной памяти EEPROM 8 Кбайт;
- объем встроенной памяти OTP 256 бит;
- четыре источника тактирования;
- два домена питания:
 - напряжение питания основного домена от 2,97В до 3,63В;
 - напряжение питания батарейного домена от 2,5В до 3,63В;
 - напряжение питания аналоговых блоков домена от 3,15 В до 3,45В;
 - динамический ток потребления основного домена не более 50 мА;
 - динамический ток потребления батарейного домена не более 2 мА;
- диапазон рабочих температур от -40°C до +85°C;
- повышенная предельная температура среды +150°C;
- пониженная предельная температура среды -60°C;
- количество контактных площадок кристалла - 72;
- количество выводов корпуса - 64;

3 Состав MCU32

В состав микросхемы MCU32 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 «М» - стандартное опциональное расширение для операций целочисленного умножения и деления;
 - 3) расширение «С» - 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 16х32 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);
- PORandBor (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 на шине АНВ (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 VCC_BAT;
- контроллер выводов, обеспечивает управление функциями мультиплексированных выводов;
- системный ТАР контроллер, обеспечивает доступ к отладочному тестовому интерфейсу, обеспечивает граничное сканирование выводов микросхемы;

– блок управления тестированием, предназначен для управления тестированием 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) «Кузнечик», 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 1 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.

– Структурная схема представлена на рисунке 1.

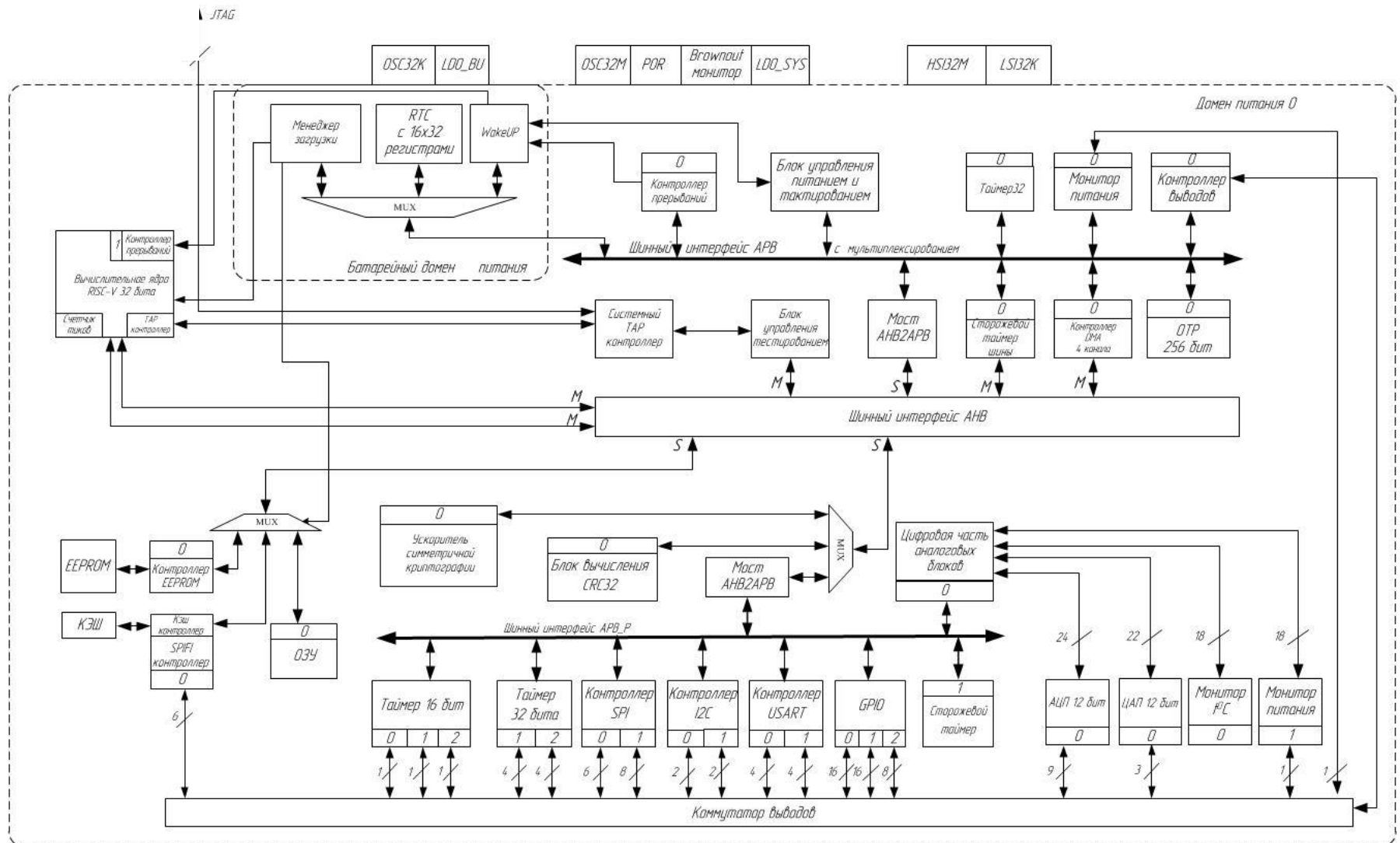


Рисунок 1 – Структурная схема микросхемы MCU32

3.2 Коммутационная матрица АНВ

Устройство предназначено для коммутации четырех ведущих и трех ведомых устройств с интерфейсом АНВ 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 1.

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.

3.3 Карта памяти

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

Таблица 2 – Распределение адресного пространства

| Смещение | Устройство |
|-------------|--|
| 0x0000_0000 | Загрузчик (16Кб) |
| 0x0004_0000 | DMA_Config |
| 0x0005_0000 | <i>Устройства на шине APB_M</i> |
| +0x0000 | Блок управления питанием и тактированием |

| Смещение | Устройство |
|-----------------|---|
| +0x0400 | Контроллер прерываний |
| +0x0800 | Таймер32_0 |
| +0x0C00 | Контроллер выводов |
| +0x1000 | Сторожевой таймер шины |
| +0x1400 | ОТР |
| +0x1800 | Монитор напряжения |
| 0x00006_0000 | <i>Устройства батарейного домена</i> |
| +0x000 | WakeUp |
| +0x400 | RTC |
| 0x00007_0000 | <i>Устройства подсистемы памяти</i> |
| +0x000 | Регистры контроллера SPIFI |
| +0x400 | Регистры контроллера EEPROM |
| 0x0008_0000 | <i>Устройства на шине АНВ_Р</i> |
| +0x000 | Ускоритель симметричной криптографии |
| +0x400 | CRC32 |
| 0x0008_1000 | <i>Устройства на шине APB_Р</i> |
| +0x0000 | Сторожевой таймер |
| +0x0400 | USART_0 |
| +0x0800 | USART_1 |
| +0x0C00 | TIMER16_0 |
| +0x1000 | TIMER16_1 |
| +0x1400 | TIMER16_2 |
| +0x1800 | TIMER32_1 |
| +0x1C00 | TIMER32_2 |
| +0x2000 | SPI_0 |
| +0x2400 | SPI_1 |
| +0x2800 | I ² C_0 |
| +0x2C00 | I ² C_1 |
| +0x3000 | GPIO16_0 |
| +0x3400 | GPIO16_1 |

| Смещение | Устройство |
|------------|--|
| +0x3800 | GPIO8_2 |
| +0x3C00 | Модуль формирования прерываний GPIO |
| +0x4000 | Блок управления аналоговой подсистемой |
| 0x01000000 | EEPROM (8Кб) |
| 0x02000000 | SRAM (16Кб) |
| 0x80000000 | Контроллер SPIFI (прямой доступ до 2 Гб) |

3.4 Порядок загрузки

Микросхема MCU32 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. Программно возможно изменить способ загрузки путем записи необходимой конфигурации в регистр BOOT модуля Wakeup.

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

- загрузка из EEPROM: память EEPROM (8 КБ) отображается в загрузочную область (0x0000_0000), но по прежнему остается доступной также по адресу 0x0100_0000;

- загрузка из внешней памяти: контроллер SPIFI отображается в загрузочную область (0x0000_0000), но по прежнему остается доступной также по адресу 0x8000_0000;

- загрузка из системной ОЗУ: блок ОЗУ отображается в загрузочную область (0x0000_0000), но по прежнему остается доступной также по адресу 0x0200_0000.

ек управления сигнала подтверждения (нужен не для всех периферийный 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.

Инициация трансфера 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, 5 (длительность доступа к шине показана условно в 3 такта).

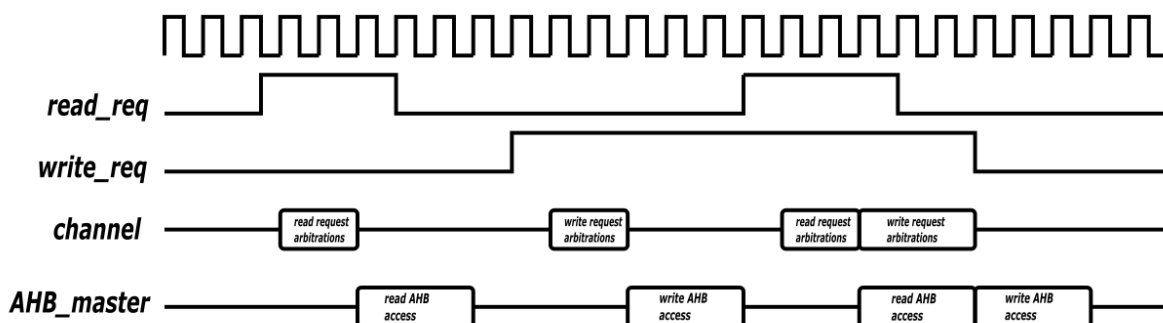


Рисунок 3 – Временная диаграмма длительности доступа к шине для одного канала

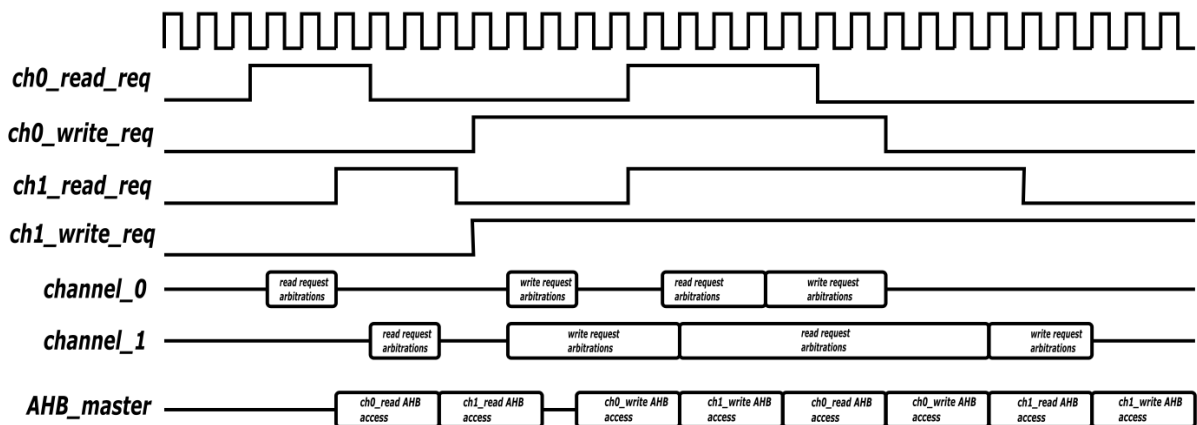


Рисунок 4 – Временная диаграмма длительности доступа к шине для двух каналов

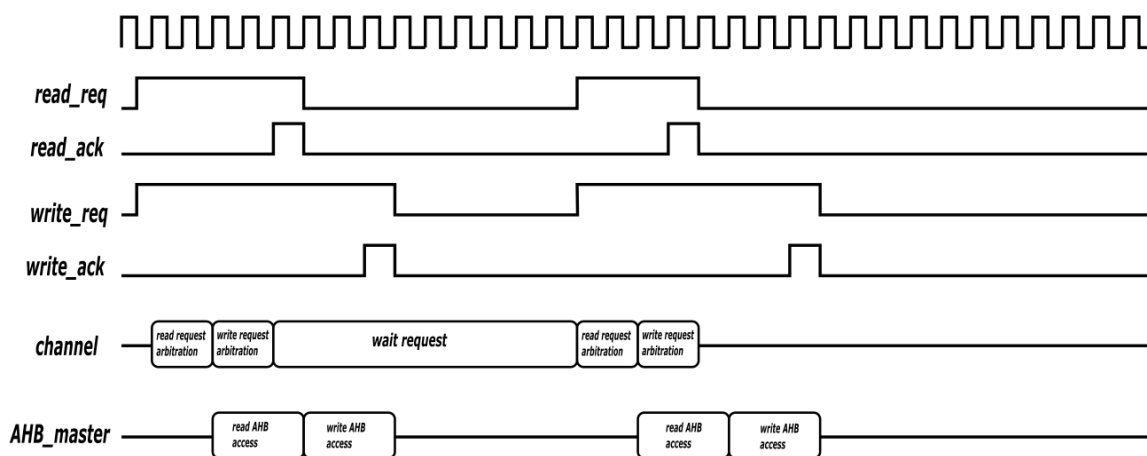


Рисунок 5 – Временная диаграмма длительности доступа к шине для одного канала с формированием сигнала снятия запроса

3.5.2 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 6.

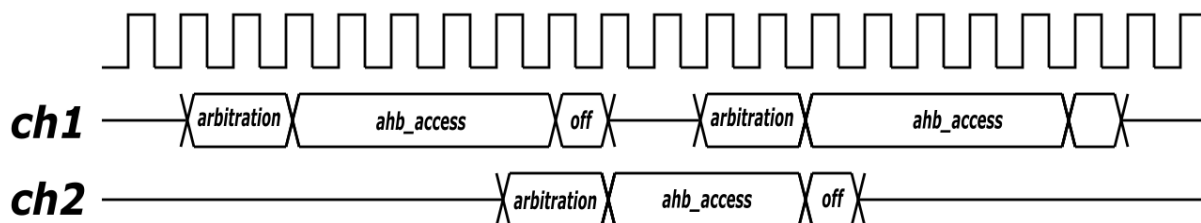


Рисунок 6 – Временные диаграммы работы нескольких каналов

3.5.3 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.

3.5.4 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 7.

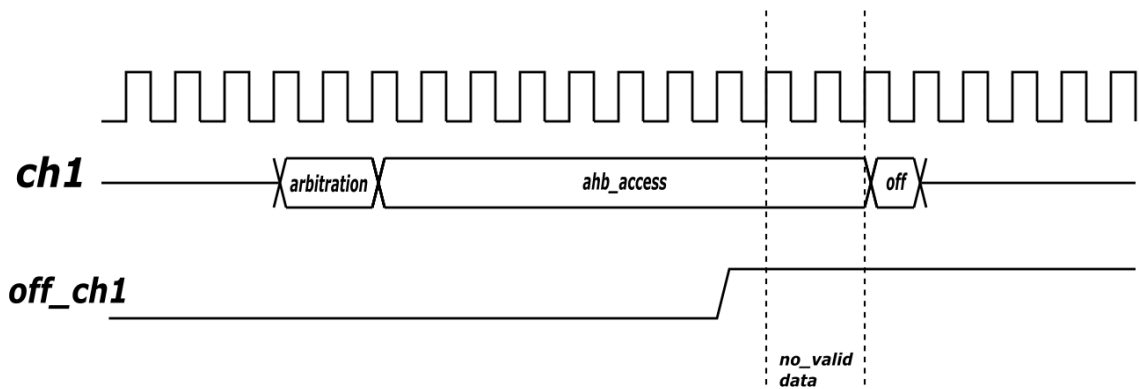


Рисунок 7 – Временная диаграмма передачи команды выключения во время транзакции, мастер-интерфейс контроллера

3.5.5 Блочные 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 находится 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 8–10):

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

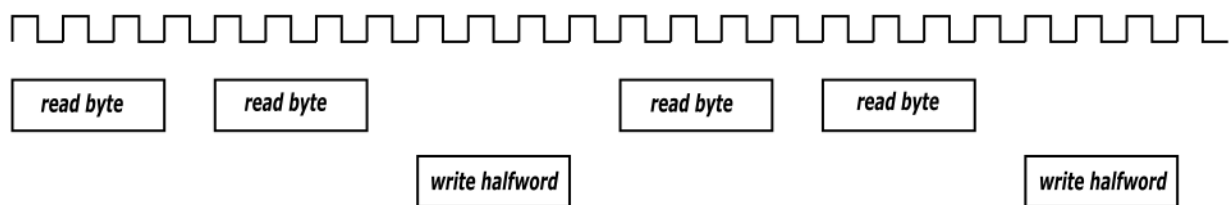


Рисунок 8 – Временная диаграмма организации блочных транзакций 1

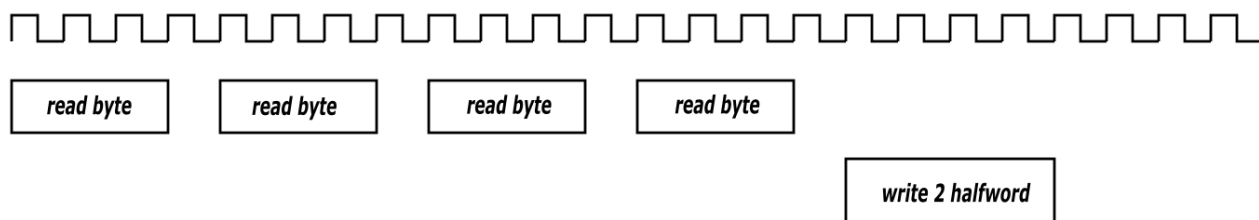


Рисунок 9 – Временная диаграмма организации блочных транзакций 2

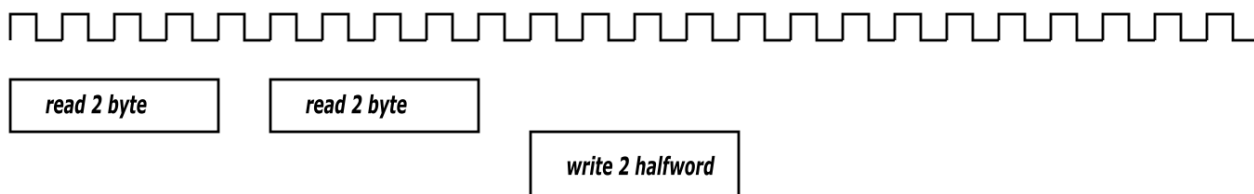


Рисунок 10 – Временная диаграмма организации блочных транзакций 3

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

3.5.6 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.

3.5.7 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 requests, Write requests регистра Config. Соответствие между значениями этих полей и линиями периферии представлено в таблице 4.

Таблица 4 – Линии запросов от периферии ПДП контроллера

| Устройство | Значение Read requests/ Write requests |
|------------|---|
| 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 |

3.5.8 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

| Адрес | Описание регистра | Название | Доступ |
|-------|--|----------|--------|
| 0x00 | Регистр адреса назначения канала 1 | CH1_DST | R/W |
| 0x04 | Регистр адреса источника канала 1 | CH1_SRC | R/W |
| 0x08 | Регистр размера передаваемых данных канала 1 | CH1_LEN | R/W |
| 0x0C | Регистр управления и конфигурации канала 1 | CH1_CFG | R/W |

| Адрес | Описание регистра | Название | Доступ |
|-------|--|----------|--------|
| 0x10 | Регистр адреса назначения канала 2 | CH2_DST | R/W |
| 0x14 | Регистр адреса источника канала 2 | CH2_SRC | R/W |
| 0x18 | Регистр размера передаваемых данных канала 2 | CH2_LEN | R/W |
| 0x1C | Регистр управления и конфигурации канала 2 | CH2_CFG | R/W |
| 0x20 | Регистр адреса назначения канала 3 | CH3_DST | R/W |
| 0x24 | Регистр адреса источника канала 3 | CH3_SRC | R/W |
| 0x28 | Регистр размера передаваемых данных канала 3 | CH3_LEN | R/W |
| 0x2C | Регистр управления и конфигурации канала 3 | CH3_CFG | R/W |
| 0x30 | Регистр адреса назначения канала 4 | CH4_DST | R/W |
| 0x34 | Регистр адреса источника канала 4 | CH4_SRC | R/W |
| 0x38 | Регистр размера передаваемых данных канала 4 | CH4_LEN | R/W |
| 0x3C | Регистр управления и конфигурации канала 4 | CH4_CFG | R/W |
| 0x40 | Регистр прерываний и настройки контроллера | DMA_CNTR | R/W |

3.5.8.1 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 | Поля регистра | Описание |
|-------------|---------------|------------------|
| Dst | 31:0 | Адрес назначения |

3.5.8.2 Регистры CHx_SRC

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

Таблица 7 – Обозначение, поля и описание регистра источника данных

| Обозначение | Поля регистра | Описание |
|-------------|---------------|-----------------|
| Src | 31:0 | Адрес источника |

3.5.8.3 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 | Поля регистра | Описание |
|-------------|---------------|--|
| Data_Len | 31:0 | Количество байт пересылки рассчитывается как Data_Len +1 |

3.5.8.4 Регистры CHx_CFG

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

Таблица 9 – Обозначение, поля и описание регистра управления каналами

| Обозначение | Поля регистра | Описание |
|-------------|---------------|--|
| Enable | 0 | Разрешение работы канала: «1» – инициализация работы канала; «0» – принудительная остановка |
| Prior | 2:1 | Приоритет канала: «00» – низкий; «01» – средний; «10» – высокий; «11» – приоритетный (очень высокий) |
| Read mode | 3 | Режим адреса источника: «0» – периферия; «1» – память |
| Write mode | 4 | Режим адреса назначения: «0» – периферия; «1» – память |
| Read incr | 5 | Инкремент адреса источника: «0» – нет инкремента; «1» – есть инкремент |
| Write incr | 6 | Инкремент адреса назначения: «0» – нет инкремента; «1» – есть инкремент |

| Обозначение | Поля регистра | Описание |
|------------------|---------------|--|
| Read size | 8:7 | Разрядность адреса источника: «00» – байт; «01» – полуслово; «10» – слово; «11» - резерв * должно быть кратно data_len |
| Write size | 10:9 | Разрядность адреса назначения: «00» – байт; «01» – полуслово; «10» – слово; «11» - резерв * должно быть кратно data_len |
| Read block size | 13:11 | Количество байт в пакете Определяется как $2^{\text{read_block_size}}$ * должно быть кратно read_size |
| Write block size | 16:14 | Количество байт в пакете Определяется как $2^{\text{write_block_size}}$ * должно быть кратно read_size |
| Read requets | 20:17 | Выбор периферийной линии источника |
| Write requets | 24:21 | Выбор периферийной линии назначения |
| Read ack ena | 25 | Разрешение работы логики с откликом для адресата источника: 0 – запрещено; 1 – разрешено |
| Write ack ena | 26 | Разрешение работы логики с откликом для адресата назначения: 0 – запрещено; 1 – разрешено |
| Irq_ena | 27 | Разрешение формирования прерывания по завершении работы канала: 0 – прерывание не формируется; 1 – прерывание формируется |

3.5.8.5 Регистр DMA_CNTR

Регистр DMA_CNTR обеспечивает управление прерываниями каналов и контроллера в целом. Функции отдельных полей регистра отличаются при выполне-

нии операции чтения и записи. Обозначение, поля и описание регистра при записи представлены в таблице 10, при чтении представлены в таблице 11.

Таблица 10 – Обозначение, поля и описание регистра DMA_CNTR при записи

| Обозначение | Поля регистра | Описание |
|---|---------------|--|
| Clear_Local_Irq | 3:0 | Очистка локального прерывания: «1» – снятие запроса на прерывание |
| Clear_Global_Irq | 4 | Очистка глобального прерывания: «1» – снятие запроса на прерывание |
| Clear_Error_Irq | 5 | Очистка прерывания ошибки: «1» – снятие запроса на прерывание |
| Global_Irq_Ena | 6 | Разрешение формирования глобального прерывания: «1» – разрешено; «0» – запрещено |
| Error_Irq_Ena | 7 | Разрешение формирования прерывания при ошибке: «1» - разрешено; «0» -запрещено |
| Current_Value | 8 | Разрешение чтения текущего статуса канала: «1» – разрешено; «0» – запрещено |
| Примечание – Доступ на установку регистров разрешения прерываний возможен только при отсутствии установки на снятие запроса линии прерывания. | | |

Таблица 11 –Обозначение, поля и описание регистра DMA_CNTR при чтении

| Обозначение | Поля регистра | Описание |
|-----------------------|---------------|---|
| Channel_Ready | 3:0 | Статус состояния каналов: «1» - готов к работе; «0» - занят |
| Channel_Irq | 7: 4 | Статус прерываний: «1» - есть прерывания; «0» - нет прерываний |
| Channel_Bus_Error | 11: 8 | Статус состояния каналов при ошибках на шине: «1» - есть ошибка; «0» - нет ошибки |
| Channel_Current_Value | 12 | Статус чтения состояния каналов: «0» - текущие значения; |

Примечание – При передаче каналу нового задания, значения Channel_Bus_Error сбрасываются в 0.

3.5.9 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 при доступе на чтение. При этом каждый из адресов возвращает значение в соответствии с установленным значением Current_Value регистра DMA_CNTR.

В случае установки бита Current_Value будут возвращаться значения текущего счёта, в противном случае будут возвращаться значения, установленные при настройке канала. В данном случае будут возвращены данные в соответствии с таблицей 12.

Таблица 12 – Значения регистров при чтении при установке бита Current_Value

| Регистр | Поля регистра | Описание |
|-------------|---------------|---|
| Destination | 31:0 | Возвращает последнюю переданную подзадачу контроллера канала. В случае ошибки записи содержит указатель на текущий адрес мастер-интерфейса |
| Source | 31:0 | Возвращает текущую подзадачу контроллера канала. Указатель на адрес блока, который обрабатывается вычисляется как: Current_ch_read_task - $2^{\text{Read block size}}$ В случае ошибки записи содержит указатель на текущий адрес мастер-интерфейса, вернувшего HRESP |
| Len | 31:0 | Возвращает счётчик байт подзадач контроллера. счётчик байт подзадач обновляется только по подзадачам записи. Кол-во байт в блоках, которые удалось записать вычисляется как: If (state==write): Current byte len - $2^{\text{Write block size}}$ If (state==read): Current byte len |
| Config | 0 | Регистр состояния ошибки при записи 0 – ошибки не было; 1 – была зафиксирована ошибка шины при чтении |
| Config | 1 | Регистр состояния ошибки при чтении 0 – ошибки не было; 1 – была зафиксирована ошибка шины при записи |
| Config | 3:2 | Регистр состояния статуса контроллера канала 01 – канал в состоянии чтения; 10 – канала в состоянии записи |

3.6 Блок TCB

Блок 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.

3.6.1 Общая структура

Блок TCB состоит из двух модулей:

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

Структурная схема блока TCB представлена на рисунке 11.

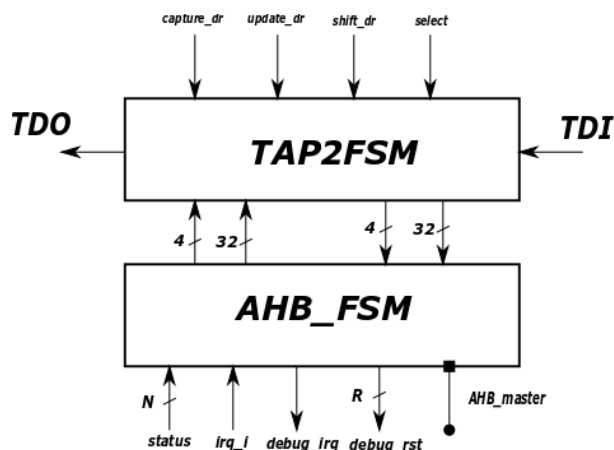


Рисунок 11 – Структурная схема блока TCB

3.6.2 Интерфейс к TAP контролеру

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 TAP контроллеру полученные данные.

3.6.3 Отладочный интерфейс системной шины

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

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.

3.6.4 Формат инструкций

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

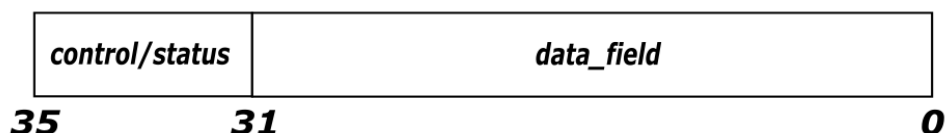


Рисунок 12 – Структура инструкции

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 |
|----------------|---------|---|--|
| xx00 | 0000 | Запись байта | Команды записи, требуют предварительно установленного адреса |
| | 0100 | Запись полуслова | |
| | 1000 | Запись слова | |
| | 1100 | Резерв | |
| xx01 | 0001 | Чтение байта | Команды чтения, возвращают данные по адресу поля data_field |
| | 0101 | Чтение полуслова | |
| | 1001 | Чтение слова | |
| | 1101 | Резерв | |
| xx10 | 0010 | Установка адреса с последующим инкрементированием | Команды установки адреса |
| | 0110 | Установка адреса без инкрементирования | |
| | 1010 | Резерв | |
| | 1110 | Резерв | |
| xx11 | 0011 | Резерв | Управляющие команды |
| | 0111 | Инвертирование линии управ- | |

| Формат команды | Команда | Описание | Примечание |
|----------------|---------|-----------------------------|------------|
| | | ления | |
| | 1011 | Системный сброс контроллера | |
| | 1111 | 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 |
|----------------|---------|--|----------------------------------|
| xxx0 | xx10 | Отклик об ошибке со стороны шины (HRESP) | Предыдущая операция не выполнена |
| | x1x0 | Ожидание отклика о готовности | |
| | 1xx0 | Резерв | |
| xxx1 | xx11 | Входная линия статуса | Предыдущая операция выполнена |
| | 0101 | Резерв | |
| | 1101 | Резерв | |

3.6.4.1 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).

3.6.4.2 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.

3.6.4.3 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.

3.6.4.4 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 ТСВ (Таблица 15).

Таблица 15 – Назначение линий управления модуля ТСВ

| 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 тестирования КЭШ памяти SPIFI контроллера) |
| [5] | Управления модулем самотестирования памяти (разрешение тестирования ОЗУ) |
| [6] | Сброс ОЗУ, контроллера EEPROM, контроллера SPIFI (активный уровень 0) |
| [7] | Сброс домена переферийных устройств (активный уровень 0) |
| [8] | Зараезервировано |
| [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.

3.6.4.5 Отклик об ошибке

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

3.6.4.6 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.

3.6.5 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.

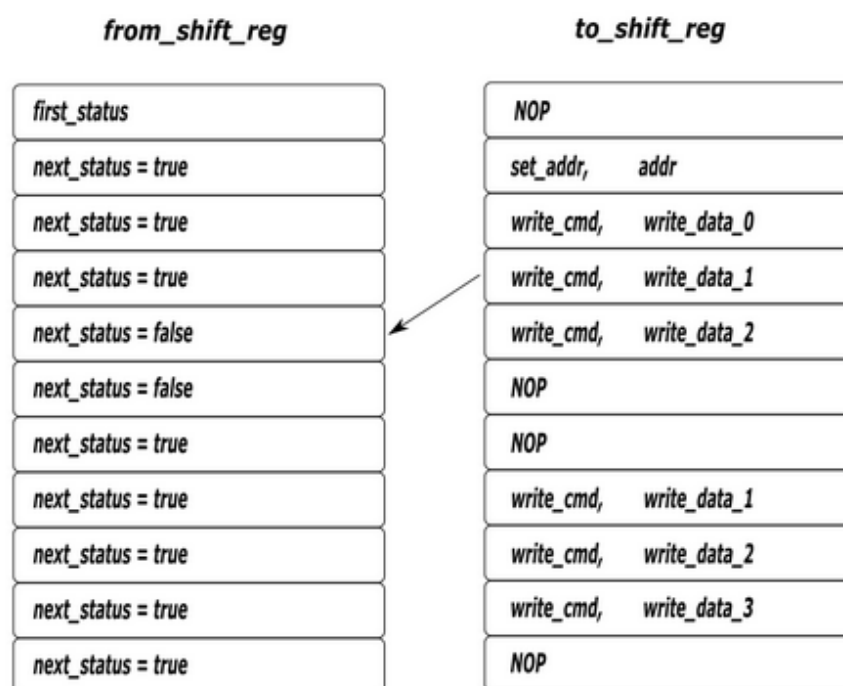


Рисунок 13 – Условный пример записи четырёх слов с возвращением неверного статуса

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

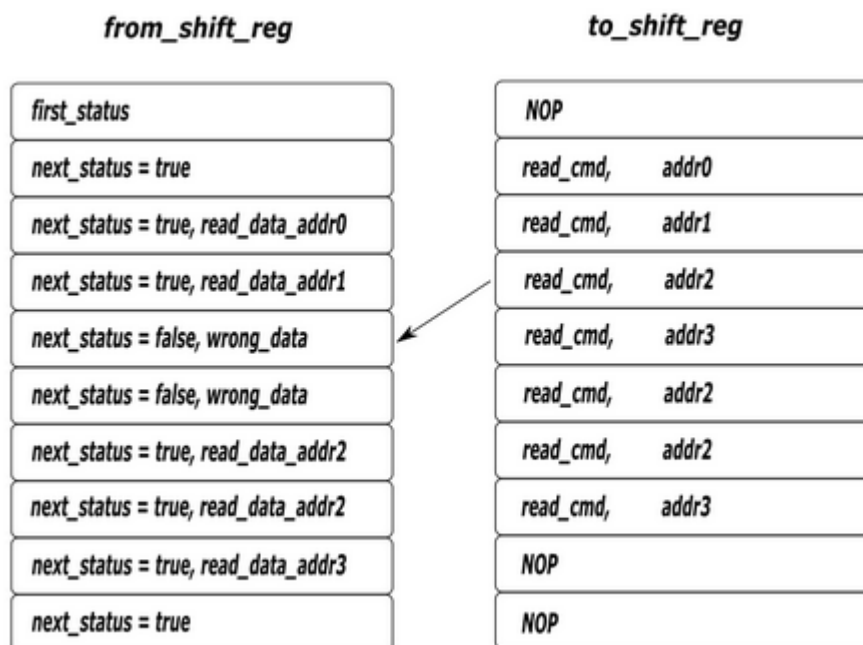


Рисунок 14 – Условный пример чтения четырёх слов с возвращением неверного статуса

3.7 Модуль USART

Универсальный синхронный асинхронный приемник-передатчик (USART) поддерживает полнодуплексный обмен данными с внешним оборудованием, а также работает в режимах синхронной односторонней связи и полудуплексной 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 полудуплексной 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 приемопередатчика USART представлена на рисунке 15.

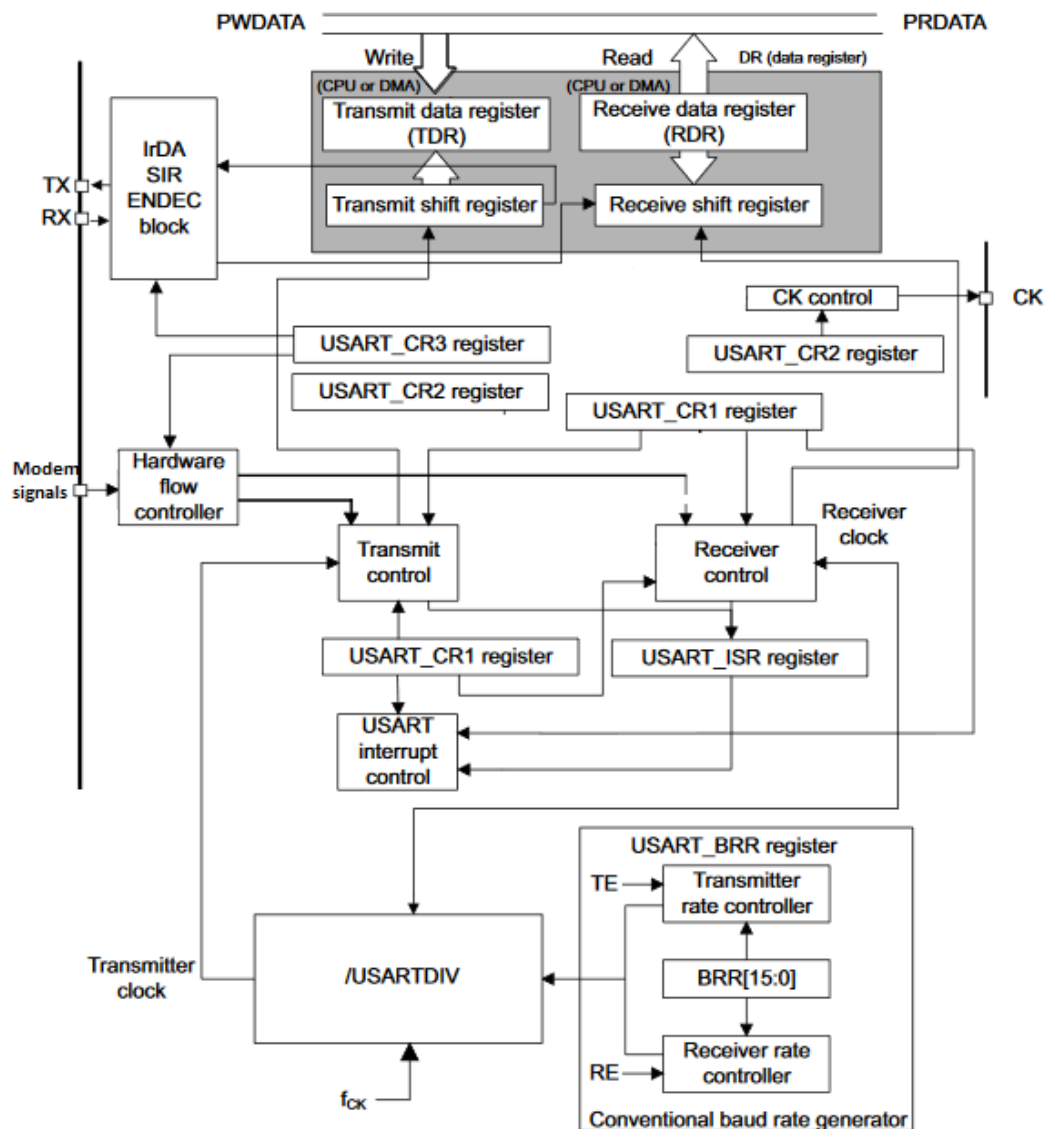


Рисунок 15 – Обобщенная структурная схема

3.7.1 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.

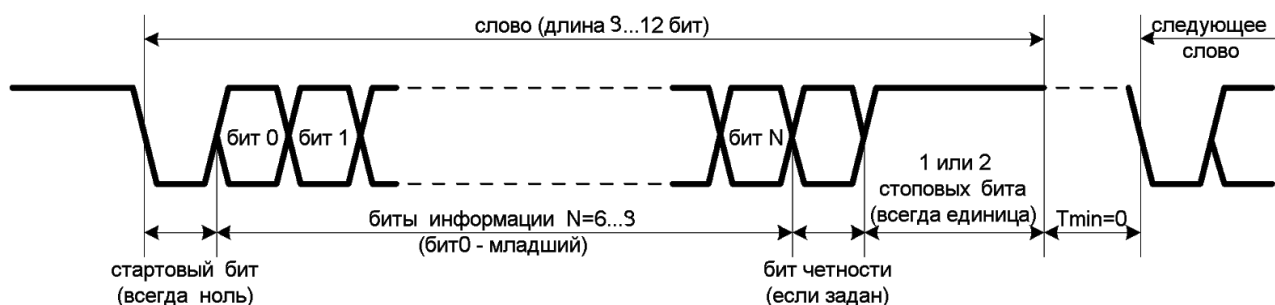


Рисунок 16 – Формат кадра

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

| М 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 |

Синхронный режим

Выход СК используется для передачи данных в синхронном режиме. СК – выходной сигнал, он выводит тактовые импульсы данных передатчика для синхронной передачи, соответствующей режиму ведущего SPI (нет тактовых импульсов на старте 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.

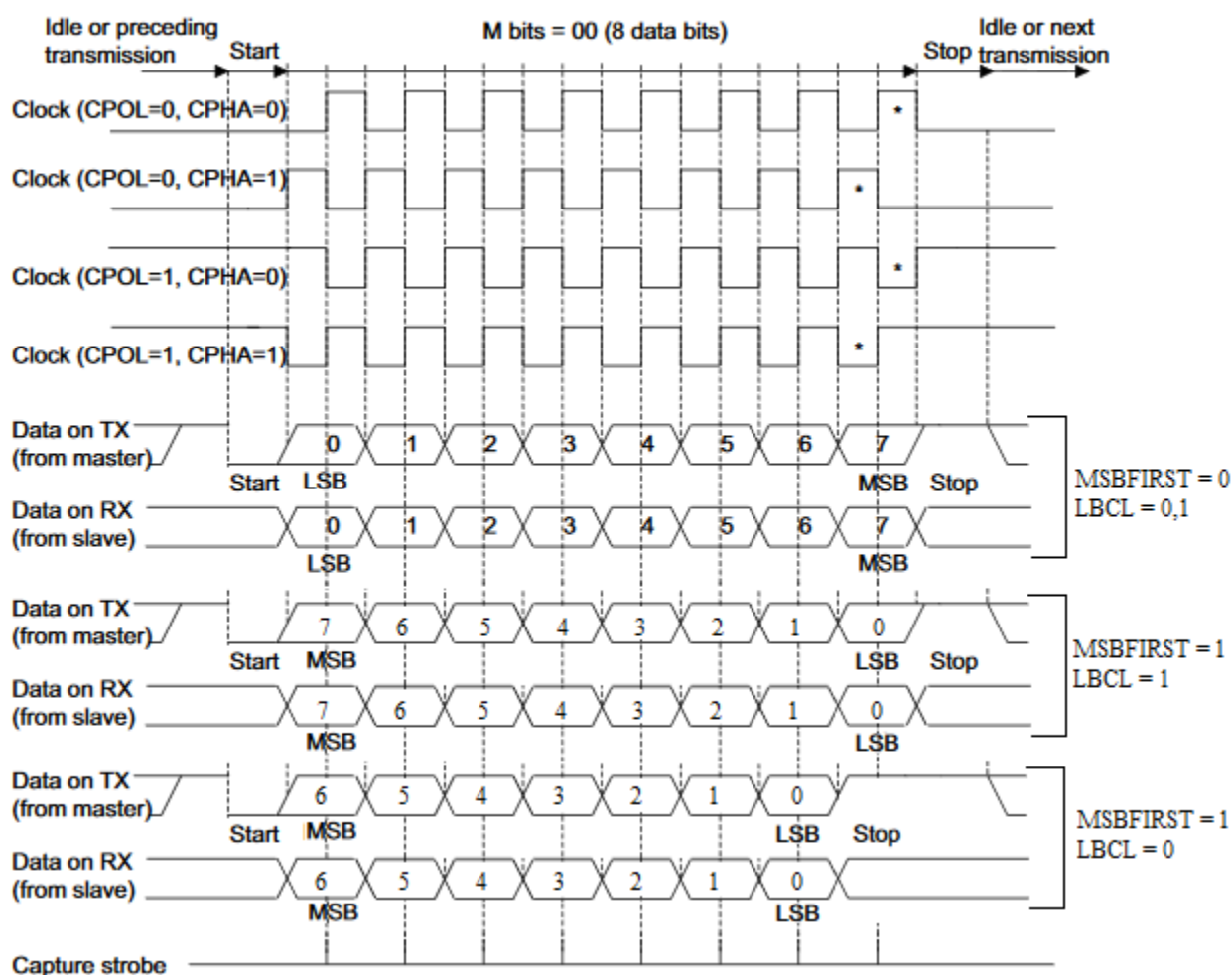


Рисунок 17 – Диаграммы работы в синхронном режиме при различных конфигурациях

3.7.2 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 СБИС.

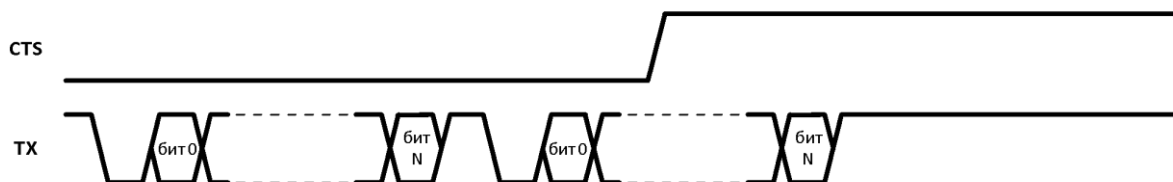


Рисунок 18 – Пример срабатывания сигнала, запрещающего передачу (CTS=1)



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

3.7.3 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.

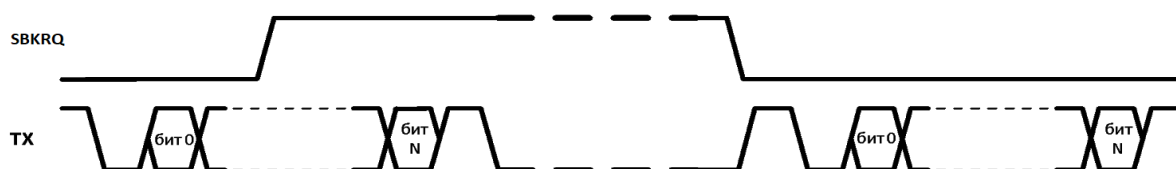


Рисунок 20 – Пример работы режима break

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.

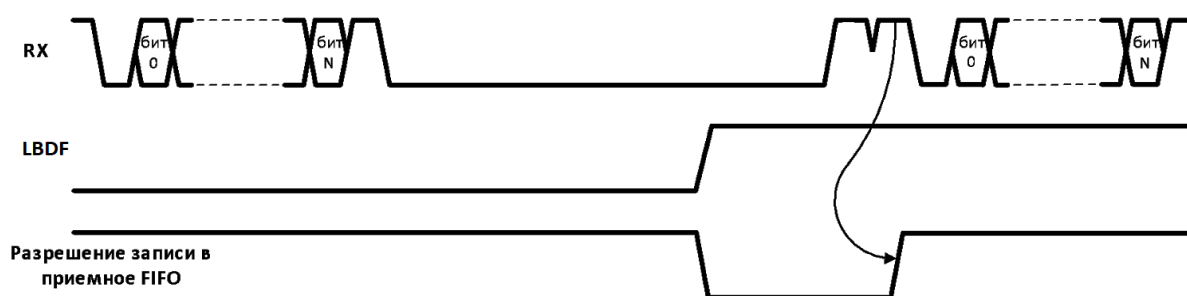


Рисунок 21 – Пример обнаружения режима break

3.7.4 Карта адресов

Регистры приемо-передатчика USART представлены в таблице 17.

Таблица 17 – Регистры приемо-передатчика USART

| Адрес | Назначение | Обозначение |
|-------|-----------------------------|-------------|
| 0x00 | Регистр управления 1 | USART_CR1 |
| 0x04 | Регистр управления 2 | USART_CR2 |
| 0x08 | Регистр управления 3 | USART_CR3 |
| 0x0C | Регистр настройки делителя | USART_DIV |
| 0x1C | Регистр прерываний | USART_INTR |
| 0x24 | Регистр принятых данных | USART_TX |
| 0x28 | Регистр передаваемых данных | USART_RX |
| 0x2C | Регистр управления модемом | USART_MODEM |

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

Таблица 18 – Регистр управления 1

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 31:29 | 0x0 | Зарезервировано | RW |
| M1 | 28 | 0x0 | Совместно с битом 12 (M0) управляет длиной кадра: M[1:0] = 00: 8 бит данных; M[1:0] = 01: 9 бит данных; M[1:0] = 10: 7 бит данных. Этот бит может быть изменен только при остановке работы (UE=0) | RW |
| Зарезервировано | 27:13 | 0x0 | Зарезервировано | RW |
| M0 | 12 | 0x0 | Совместно с битом 28 (M1) управляет длиной кадра: M[1:0] = 00: 8 бит данных; M[1:0] = 01: 9 бит данных; M[1:0] = 10: 7 бит данных. Этот бит может быть изменен только при остановке работы (UE=0) | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 11 | 0x0 | Зарезервировано | RW |
| 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 |
| 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 | – | 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|---|---------|--------------------|--|--------|
| 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 |
| RE | 2 | 0x0 | Управление приемником: 0 – приемник выключен; 1 – приемник включен | RW |
| Зарезервировано | 1 | 0x0 | Зарезервировано | RW |
| UE | 0 | 0x0 | Ресет USART. Отменяет все текущие операции. Не очищает регистры конфигурации, но сбрасывает флаги 0 – USART выключен 1 – USART включен | RW |
| Примечания 1 Смещение: 0x00. 2 Значение после сброса: 0x00000000. | | | | |

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

Т а б л и ц а 19 – Регистр управления 2

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|-----------------|--------|
| Зарезервировано | 31:20 | 0x0 | Зарезервировано | RW |

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

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|--|--------|
| SWAP | 15 | 0x0 | Регистр отвечает за обмен функциями TX выхода и RX входа: 0 – обычное функционирование; 1 – обмен функциями (TX = RX и RX = TX). Этот бит может быть изменен только при остановке работы (UE=0) | RW |
| 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 | 0x0 | Зарезервировано | RW |
| 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|---|---------|--------------------|--|--------|
| OCPL | 10 | 0x0 | Управление полярностью выходного тактового сигнала СК: 0 – логический «0» вне транзакции; 1 – логическая «1» вне транзакции. Этот бит может быть изменен только при остановке работы (UE=0) | RW |
| CPHA | 9 | 0x0 | Управление фазой выходного тактового сигнала СК: 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 | 0x0 | Зарезервировано | RW |
| LBDIE | 6 | 0x0 | Управление прерыванием при обнаружении break состояния на RX линии: 0 – прерывание выключено; 1 – прерывание включено | RW |
| Зарезервировано | 5:0 | 0x0 | Зарезервировано | RW |
| Примечания 1 Смещение: 0x00. 2 Значение после сброса: 0x00000000. | | | | |

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

Таблица 20 – Регистр управления 3

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 31:13 | 0x0 | 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 | 0x0 | 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| 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 |
| DMAT | 7 | 0x0 | Управление работой сигнала dma_trq: 0 – сигнал выключен; 1 – сигнал включен | RW |
| DMAR | 6 | 0x0 | Управление работой сигнала dma_grq: 0 – сигнал выключен; 1 – сигнал включен | RW |
| Зарезервировано | 5:4 | 0x0 | 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|--|---------|--------------------|--|--------|
| Зарезервировано | 1 | 0x0 | Зарезервировано | – |
| EIE | 0 | 0x0 | Управление прерыванием при обнаружении ошибок приема данных (FE, ORE, NF): 0 – прерывание выключено; 1 – прерывание включено | RW |
| <p>Примечания</p> <p>1 Смещение: 0x08.</p> <p>2 Значение после сброса: 0x00000000.</p> | | | | |

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

Таблица 21 – Регистр делителя

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|--|---------|--------------------|---|--------|
| Зарезервировано | 31:16 | 0x0 | 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 |
| <p>Примечания</p> <p>1 Смещение: 0x0C.</p> <p>2 Значение после сброса: 0x00000000.</p> | | | | |

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

Таблица 22 – Регистр флагов и прерываний

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 31:23 | 0x0 | Зарезервировано | – |
| REACK | 22 | 0x0 | Флаг готовности приемника к работе после снятия сброса (UE и RE): 0 – ресет активен; 1 – ресет снят | R |
| TEACK | 21 | 0x0 | Флаг готовности передатчика к работе после снятия сброса (UE и TE): 0 – ресет активен; 1 – ресет снят | R |
| Зарезервировано | 20:17 | 0x0 | Зарезервировано | – |
| BUSY | 16 | 0x0 | Флаг активности на линии RX: 0 – нет активности на линии RX; 1 – идет передача данных или break состояние | R |
| Зарезервировано | 15:11 | 0x0 | Зарезервировано | – |
| CTS | 10 | 0x0 | Флаг отображает текущее значение сигнала CTS: 0 – CTS = 0; 1 – CTS = 1 | R |
| 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| 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 |
| 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| IDLE | 4 | 0x0 | Флаг взводится при отсутствии активности на линии RX в течении 8 битовых тактов при взведенном 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 |
| 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 | Доступ |
|--------------------------------------|---------|--------------------|------------|--------|
| Примечания | | | | |
| 1 Смещение: 0x1C. | | | | |
| 2 Значение после сброса: 0x00000000. | | | | |

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

Таблица 23 – Регистр принятых данных

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|------------------------------------|---------|--------------------|-----------------|--------|
| Зарезервировано | 31:9 | 0x0 | Зарезервировано | – |
| RDR | 8:0 | 0x0 | – | R |
| Примечания | | | | |
| 1 Смещение: 0x24. | | | | |
| 2 Значение после сброса 0x00000000 | | | | |

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

Таблица 24 – Регистр передаваемых данных

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|--------------------------------------|---------|--------------------|-----------------|--------|
| Зарезервировано | 31:9 | 0x0 | Зарезервировано | – |
| TDR | 8:0 | 0x0 | – | RW |
| Примечания | | | | |
| 1 Смещение: 0x28. | | | | |
| 2 Значение после сброса: 0x00000000. | | | | |

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

Таблица 25 – Регистр модемных сигналов

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|--|--------|
| Зарезервировано | 31:9 | 0x0 | 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 | R |
| RI | 6 | 0x0 | Регистр содержит текущее значение сигнала RI: 0 – отсутствие звонка; 1 – звонок (вызов) на телефонной линии | R |
| DSR | 5 | 0x0 | Регистр содержит текущее значение сигнала DSR: 0 – источник данных не готов; 1 – источник данных готов | R |
| Зарезервировано | 4 | 0x0 | 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 | 0x0 | Зарезервировано | – |
| <p>Примечания</p> <p>1 Смещение: 0x2C.</p> <p>2 Значение после сброса: 0x00000000.</p> | | | | |

3.8 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.

3.8.1 Характеристики

Характеристики модуля I²C:

- соответствие спецификации шины I²C rev03:
 - 1) режимы ведущий и ведомый;
 - 2) несколько ведущих на шине;
 - 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:
 - 7 или 10-битный;
 - 7-битный с настраиваемой маской;
 - 8) квитиование 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.

3.8.2 Состав модуля I²C

Структура модуля I²C представлена на рисунке 22.

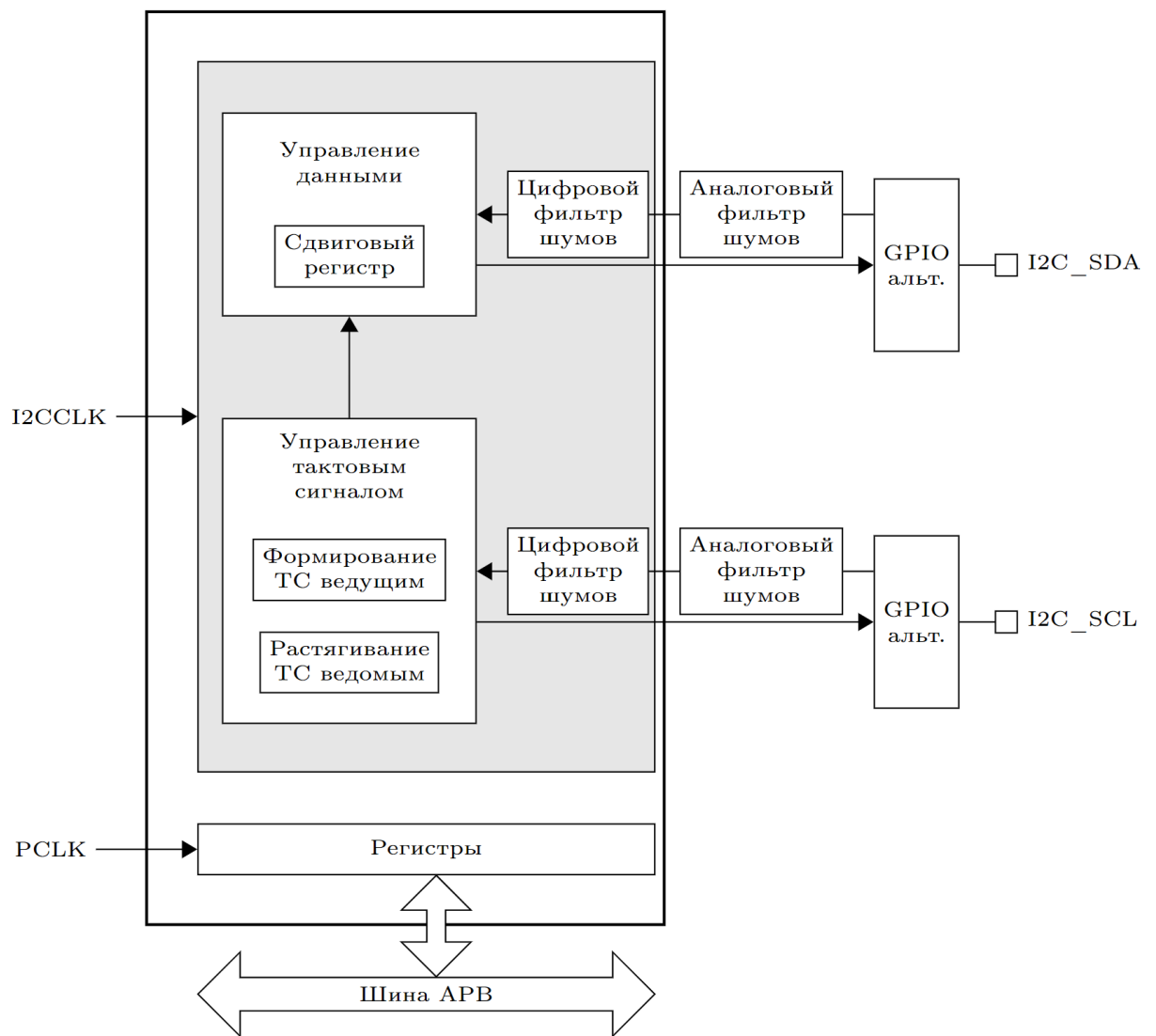


Рисунок 22 – Схема модуля I²C

3.8.3 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.

$$t_{PCLK} < (t_{LOW} - t_{filters})/4$$

$$t_{PCLK} < 4/3t_{SCL}$$

$$t_{I2CCLK} < t_{HIGH} (1)$$

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.

3.8.4 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.

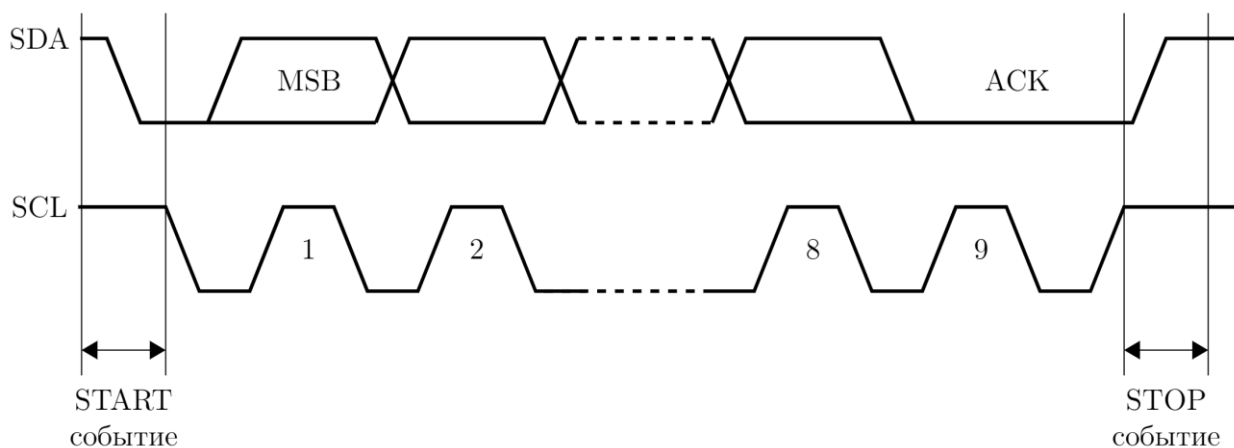


Рисунок 23 – Формат кадра

Подтверждение может быть заблокировано программой. Адреса интерфейса выбираются программой.

3.8.5 Инициализация и программный сброс

На рисунке 24 приведен алгоритм инициализации I2C.

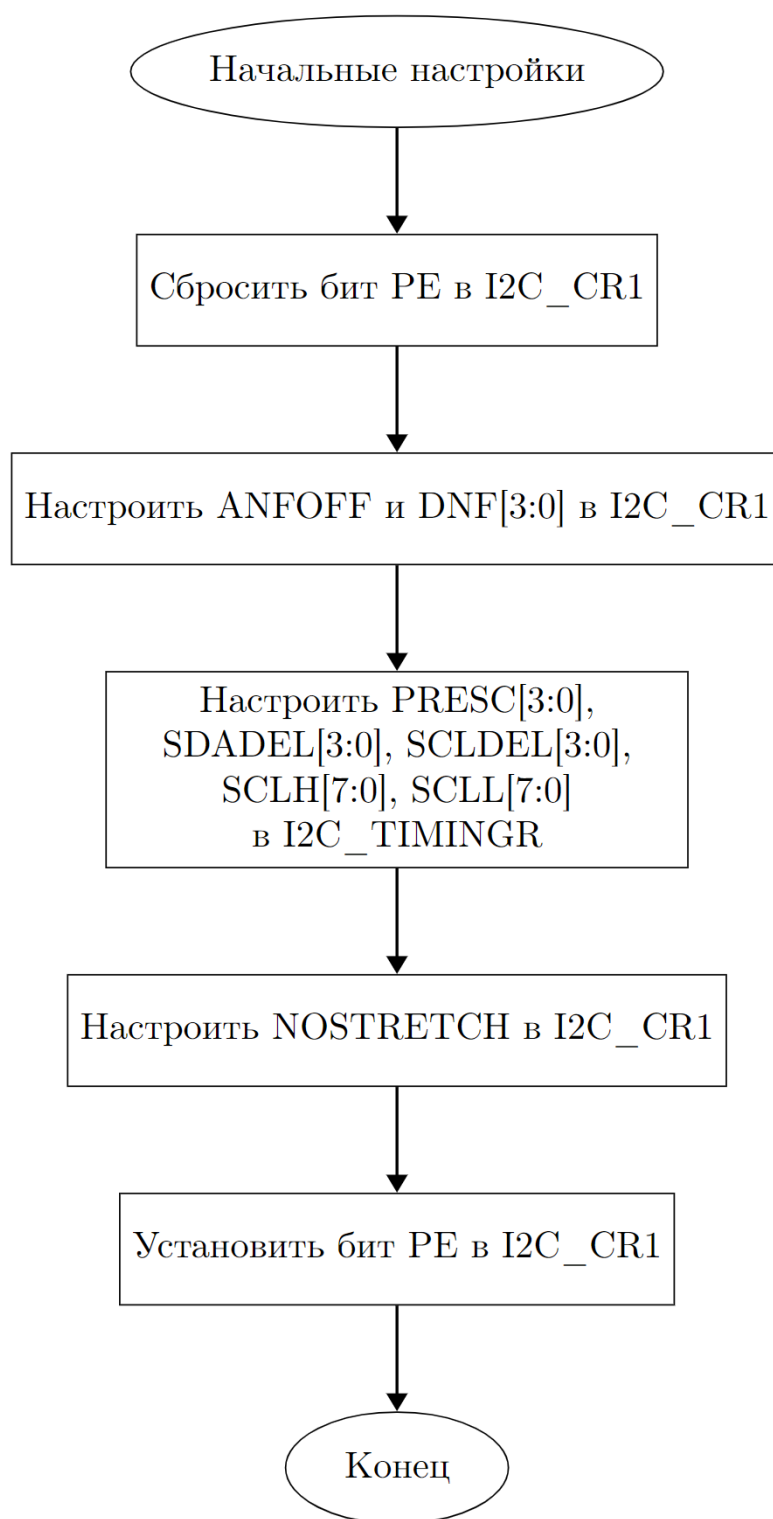


Рисунок 24 – Инициализация I2C

Интерфейс включается битом PE в регистре I2C_CR1.

Программный сброс производится очисткой бита PE в регистре I2C_CR1. При этом линии SCL и SDA освобождаются, внутренние состояния и биты управления соединением, статуса сбрасываются в начальное состояние.

Список сбрасываемых битов:

- регистр I2C_CR2: START, STOP, NACK;
- регистр I2C_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.

3.8.6 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 I2C_CR1.

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

3.8.7 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 I2C_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, $[(SDADEL + SCLDEL + 1) \times (PRESC + 1) + 1] \times t_{I2CCLK}$ в режимах отправки и приёма. В режиме отправки, в случае если данные не записаны в регистр I2C_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.

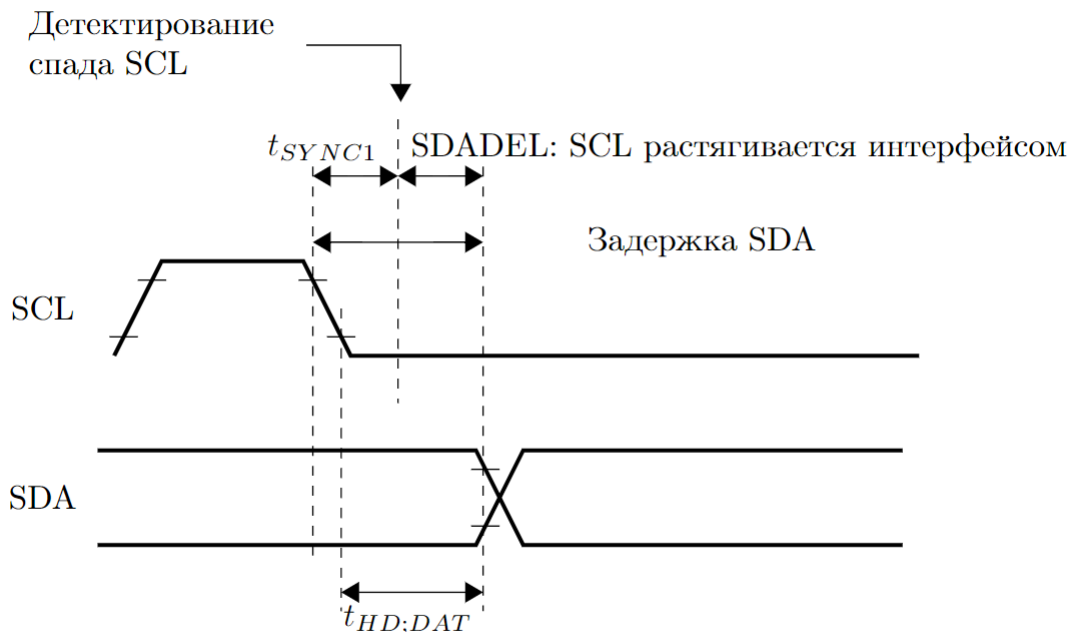
Если в режиме «ведомый» NOSTRETCH=1, SCL не удерживается. Соответственно, SDADEL должно быть запрограммировано так, чтобы гарантировать время предустановки данных.

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

Таблица 26 - Спецификация временных ограничений шины I2C

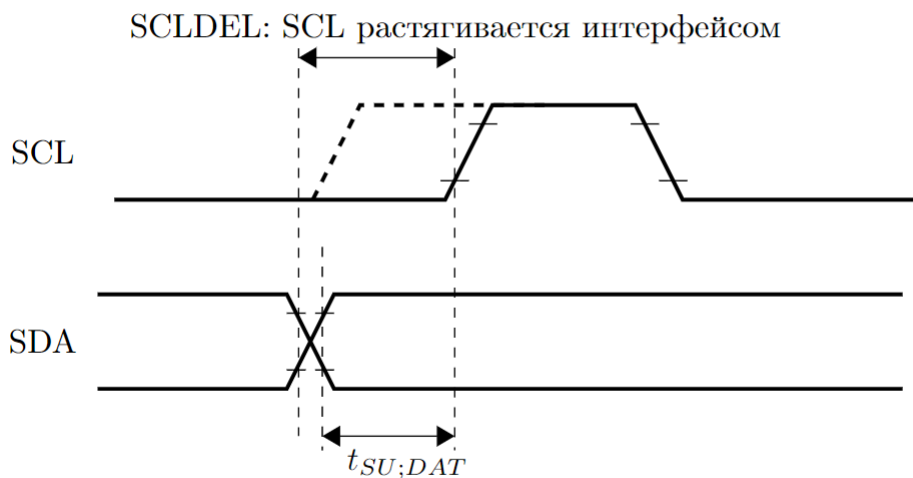
| Обозначение | Параметр | Standard-mode (Sm) | | Fast-mode (Fm) | | Fast-mode Plus (Fm+) | | Ед. изм. |
|--------------|---------------------------------------|--------------------|-------|----------------|-------|----------------------|-------|----------|
| | | Мин. | Макс. | Мин. | Макс. | Мин. | 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_f | Длительность фронта сигналов SDA, SCL | – | 1000 | – | 300 | – | 120 | нс. |
| t_f | Длительность спада сигналов SDA, SCL | – | 300 | – | 300 | – | 120 | нс. |

Время удержания данных



Удержание данных: при передаче данные выставляются на SDA после задержки на SDADEL, если они уже доступны в I2C_TXDR

Время предустановки данных



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

Рисунок 25 – Временные ограничения

3.8.7.1 Длительность удержания данных

Когда регистрируется спад на линии SCL, данные по SDA отправляются с задержкой по формуле 3

$$t_{SDADEL} = SDADEL \times t_{PRESC} + t_{I2CCLK}, \quad (3)$$

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 4

$$t_{SYNC1} + \{[SDADEL \times (PRESC+1) + 1] \times t_{I2CCLK}\} \quad (4)$$

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 5 и 6.

$$\{t_{f(max)} + t_{HD;DAT(min)} - t_{AF(min)} - [(DNF + 3) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\} \leq SDADEL \quad (5)$$

$$SDADEL \leq \{t_{HD;DAT(max)} - t_{AF(max)} - [(DNF + 4) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\} \quad (6)$$

$t_{AF(max)}$ 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.

3.8.7.2 Длительность предустановки данных

После задержки t_{SDADEL} или отправки данных в случае, когда ведомому пришлось растянуть тактовый сигнал из-за отсутствия данных в регистре I2C_TXDR, линия SCL удерживается в низком уровне в течение времени предустановки ($t_{SU;DAT}$), согласно формуле 7.

$$t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}, (7)$$

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

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

$SCLDEL$ должно быть запрограммировано согласно формуле 8

$$\{[t_{tr(max)} + t_{SU;DAT(min)}] / [(PRESC + 1) \times t_{I2CCLK}]\} - 1 \leq SCLDEL \quad (8)$$

3.8.7.3 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 $I2C_TIMINGR$:

1) после детектирования спада SCL, перед отпусканием выхода SCL, вставляется задержка согласно формуле 9

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

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

t_{SCLL} влияет на t_{LOW} ;

2) после детектирования фронта SCL, перед притягиванием выхода SCL к низкому уровню, вставляется задержка согласно формуле 10

$$t_{SCLH} = (SCLH+1) \times t_{PRESC}, \quad (10)$$

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

t_{SCLH} влияет на t_{HIGH} .

3.8.8 Передача данных

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

3.8.8.1 Приём

Сдвиговый регистр заполняется данными с входа SDA. После восьмого импульса на SCL, содержимое сдвигового регистра переписывается в регистр $I2C_RXDR$, если он свободен (Бит $RXNE=0$). Если $RXNE=1$, то есть предыдущие полученные данные не вычитаны, линия SCL удерживается на низком уровне, пока

I2C_RXDR не будет прочитан. Удержание осуществляется между восьмым и девятым импульсами, это изображено на рисунке 26.

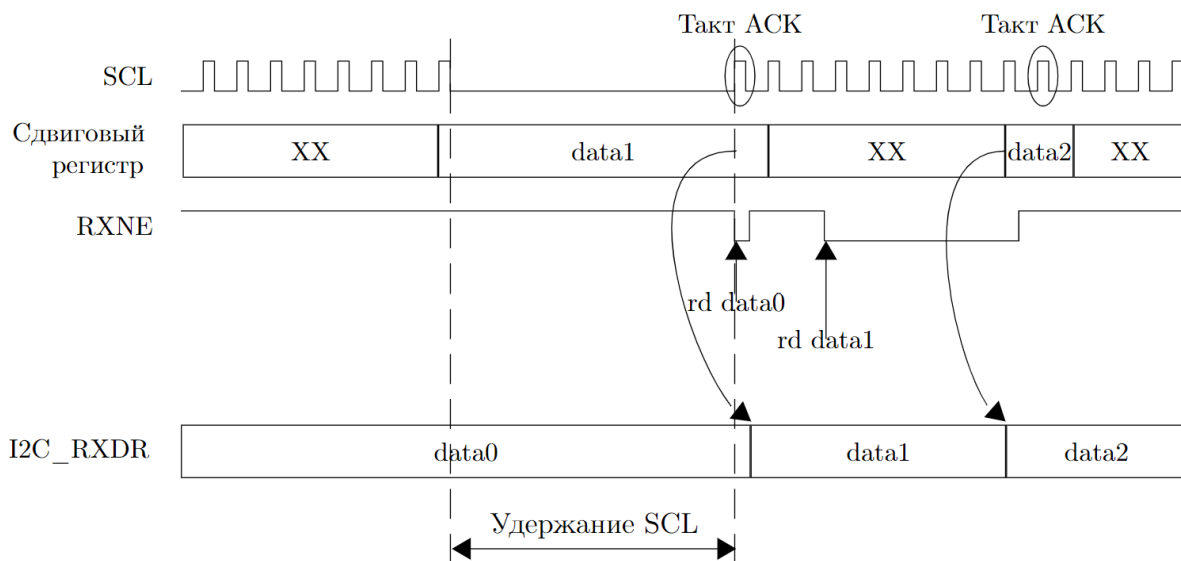


Рисунок 26 – Приём данных

3.8.8.2 Отправка

Если регистр I2C_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 регистр I2C_TXDR не записаны, то линия SCL удерживается на низком уровне, пока в регистр I2C_TXDR не будут записаны данные. Удержание осуществляется после девятого импульса на SCL, что представлено рисунке 27.

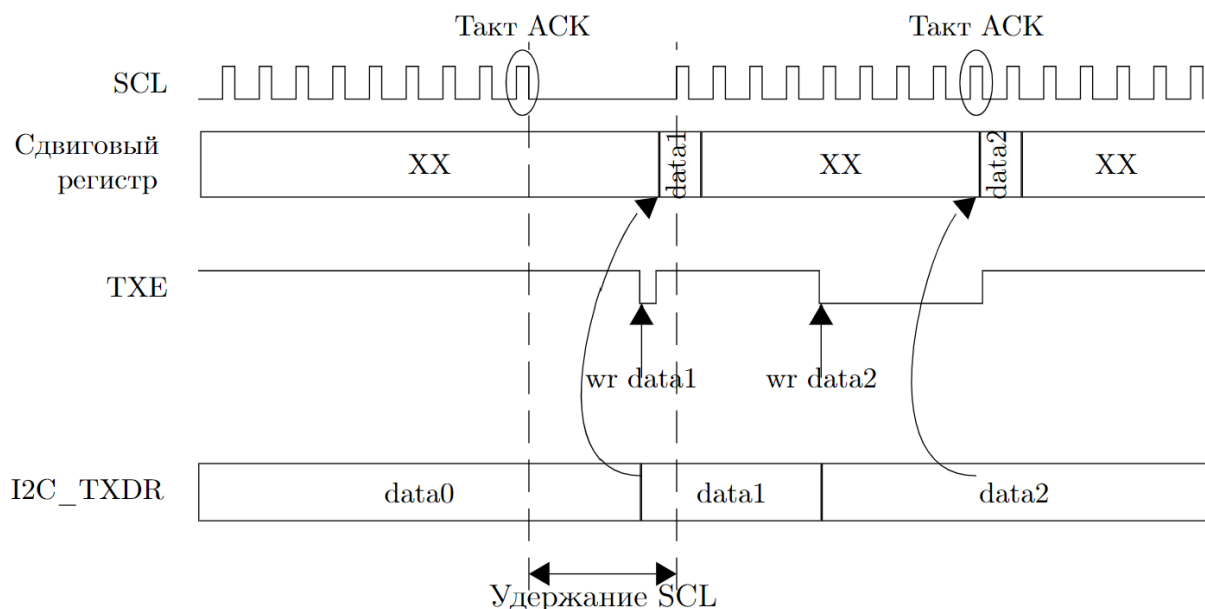


Рисунок 27 – Отправка данных

3.8.9 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 I2C_CR2.

Количество байтов для передачи программируется в битах NBYTES[7:0] в регистре I2C_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 I2C_CR2. В этом режиме взводится 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 I2C_CR2) – ведущий автоматически отправляет состояние STOP по окончании передачи всех данных, заданных в NBYTES;

– программное окончание (AUTOEND=0 в регистре I2C_CR2) – по окончании передачи всех данных, заданных в NBYTES взводится флаг TC и вырабатывается прерывание, если бит TCIE установлен и ожидается реакция программы. Флаг TC сбрасывается программой, когда биты START или STOP в регистре I2C_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.

Т а б л и ц а 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 |

3.8.10 Режим «ведомый» I²C

3.8.10.1 Инициализация ведомого

Чтобы интерфейс работал в режиме «ведомый», нужно задействовать хотя бы один адрес ведомого. Доступны два регистра I2C_OAR1 и I2C_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 I2C_OAR1. Использование OA1 разрешается, если установлен бит OA1EN в регистре I2C_OAR1.
- При необходимости можно настроить дополнительный адрес ведомого OA2. Маскирование младших семи бит OA2 можно настроить битами

OA2MSK[2:0] в регистре I2C_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 I2C_OAR1 или регистре I2C_OAR2 совместно с битами OA2MSK=0.

Использование OA2 разрешается, если установлен бит OA2EN в регистре I2C_OAR2.

- Адрес общего вызова разрешается установкой бита GCEN в регистре I2C_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 I2C_CR1.

После получения прерывания ADDR, если разрешено несколько адресов, пользователь должен вычитать биты ADDCODE[6:0] в регистре I2C_ISR, чтобы выяснить, какой из адресов вызван. Так же должен быть проанализирован флаг DIR, чтобы узнать направление передачи.

3.8.10.2 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 I2C_TXDR или первый байт данных не записан, когда

флаг ADDR сброшен (TXE=1). Это удержание прекращается после записи данных в регистр I2C_TXDR;

- во время приёма, если регистр I2C_RXDR всё ещё не вычитан и приём новых данных завершён. Это удержание прекращается после чтения из регистра I2C_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 NBYTES[7:0];

- после детектирования спада SCL интерфейс удерживает линию SCL в течение $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{2CCLK}$.

3.8.10.3 Ведомый без удержания SCL (NOSTRETCH=1)

При NOSTRETCH=1 в регистре I2C_CR1, интерфейс не удерживает SCL в низком уровне:

- SCL не удерживается при взведённом флаге ADDR;

- во время отправки данные должны быть записаны в I2C_TXDR до первого импульса SCL, соответствующего их отправке. Иначе случается нехватка данных и взводится флаг OVR в регистре I2C_ISR и вырабатывается прерывание, если установлен бит ERRIE в регистре I2C_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 I2C_RXDR до девятого импульса SCL (ACK бит) следующего байта данных. Иначе происходит переполнение данных и взводится флаг OVR в регистре I2C_ISR и вырабатывается прерывание, если установлен бит ERRIE в регистре I2C_CR1.

3.8.10.4 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 I2C_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 I2C_RXDR и затем решить, подтверждать их или нет, конфигурируя бит ACK в регистре I2C_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.

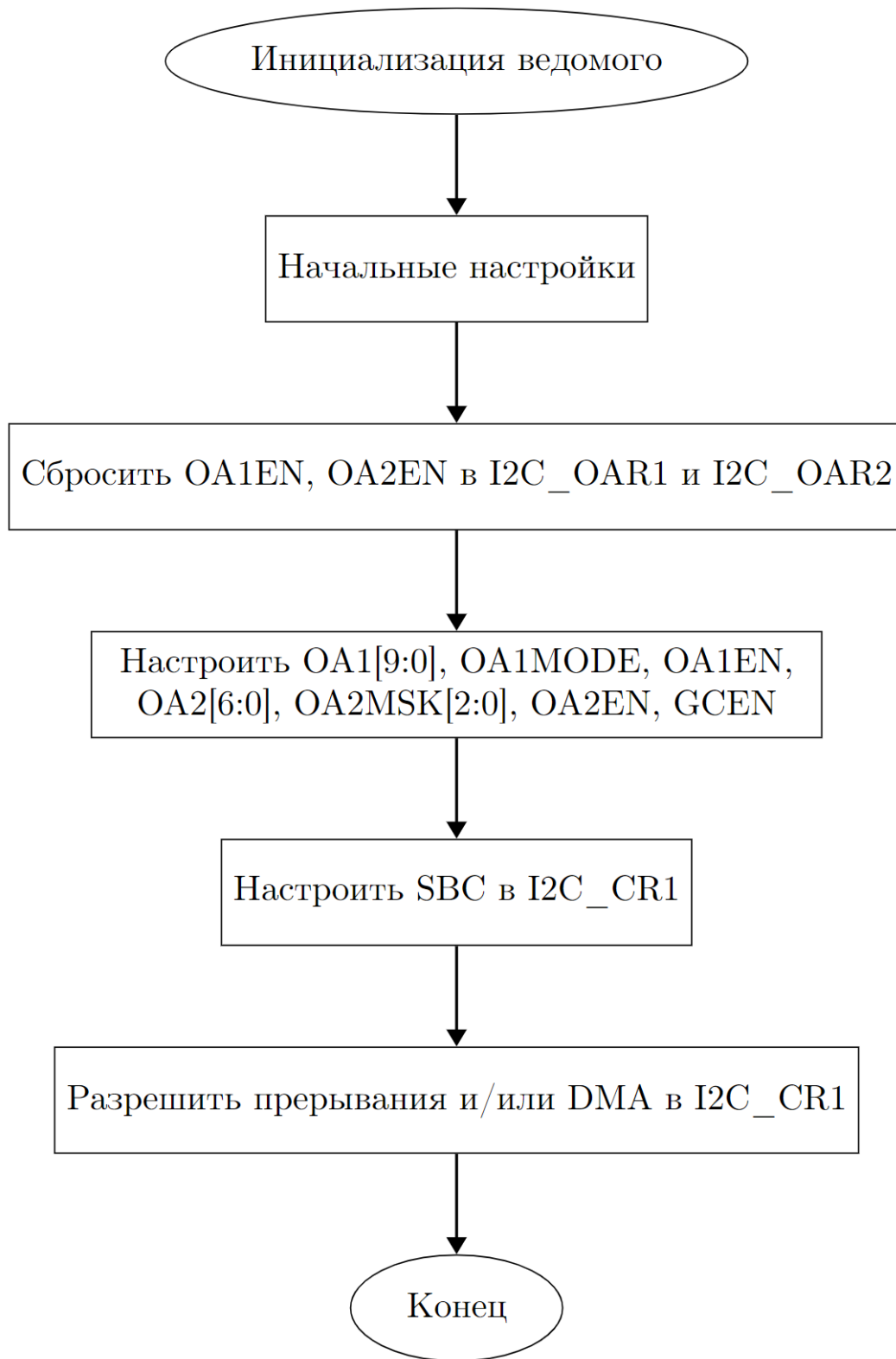


Рисунок 28 - Инициализация режима «ведомый»

3.8.10.5 Передача в режиме «ведомый»

После освобождения регистра I2C_TXDR, взводится флаг статуса прерывания TXIS. Прерывание вырабатывается, если оно разрешено битом TXIE в регистре

I2C_CR1. Флаг TXIS сбрасывается после записи данных для отправки в регистр I2C_TXDR.

Если не получено подтверждение данных (NACK), взводится флаг NACKF в регистре статуса I2C_ISR и вырабатывается прерывание, если оно разрешено битом NACKIE в регистре I2C_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 I2C_CR1 установлен, то взводится флаг STOPF в регистре I2C_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 I2C_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, записанному в NBYTES.

При NOSTRETCH=1 SCL не удерживается, пока выставлен флаг ADDR и пользователь не может сбросить содержимое регистра I2C_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 I2C_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.

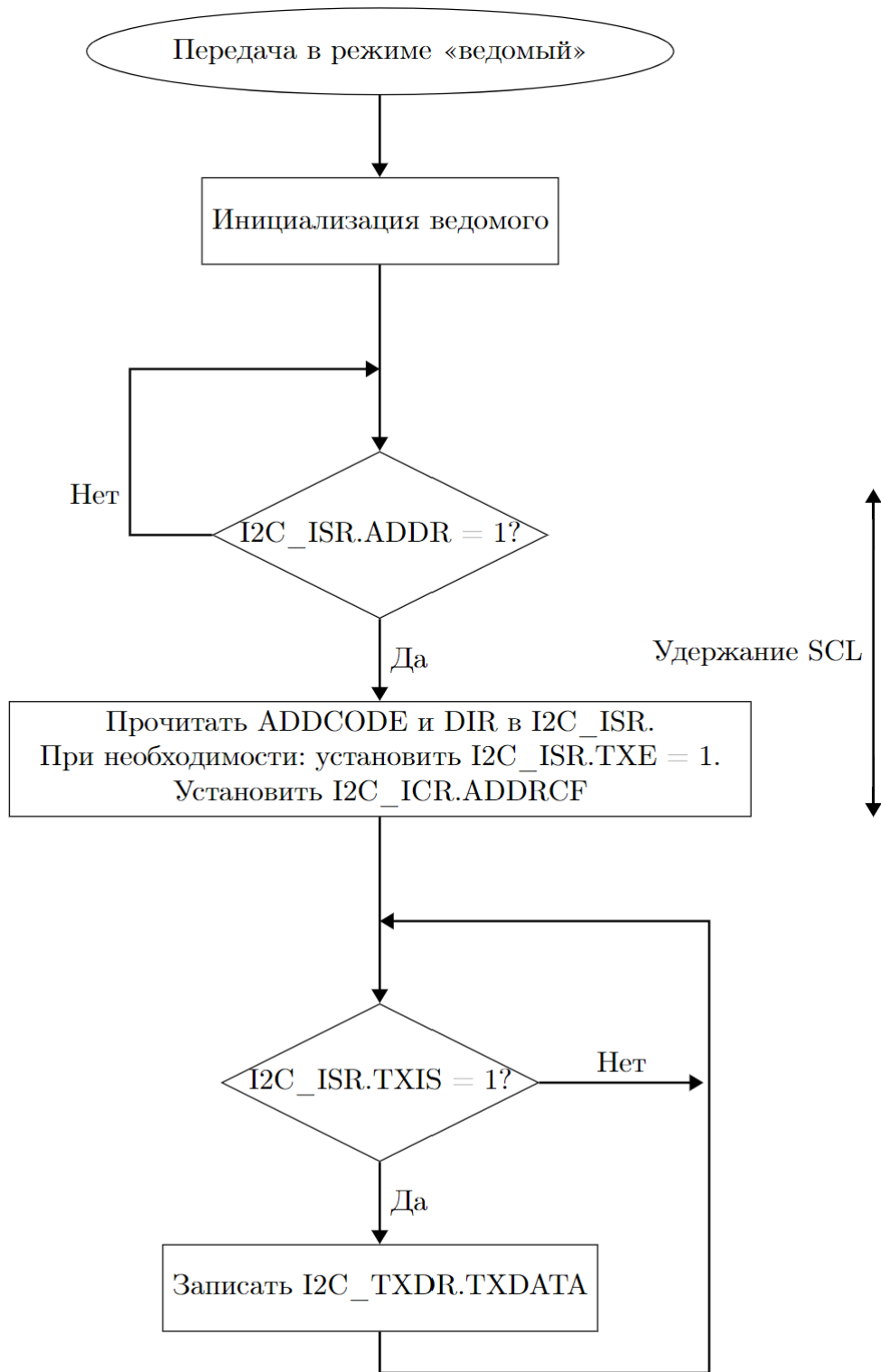


Рисунок 29 - Алгоритм передачи в режиме «ведомый», при NOSTRETCH=0

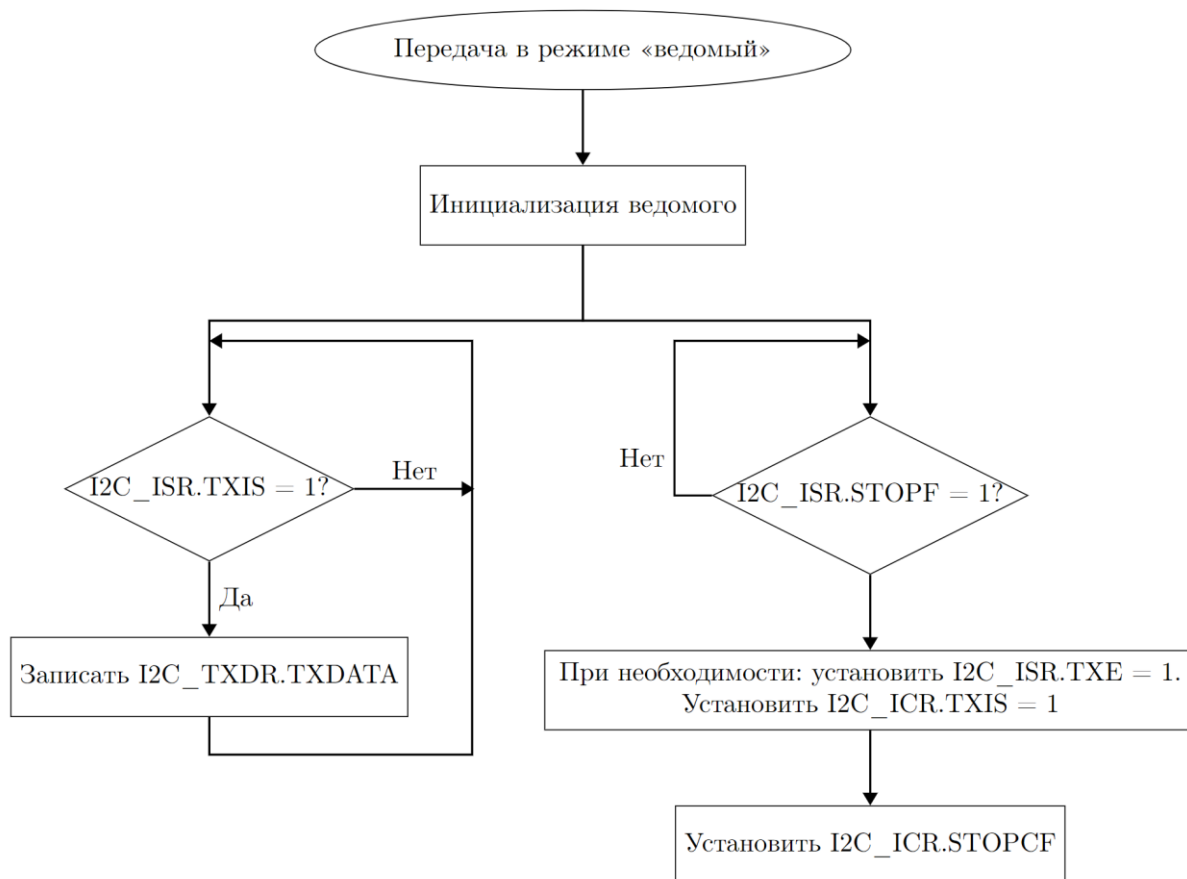
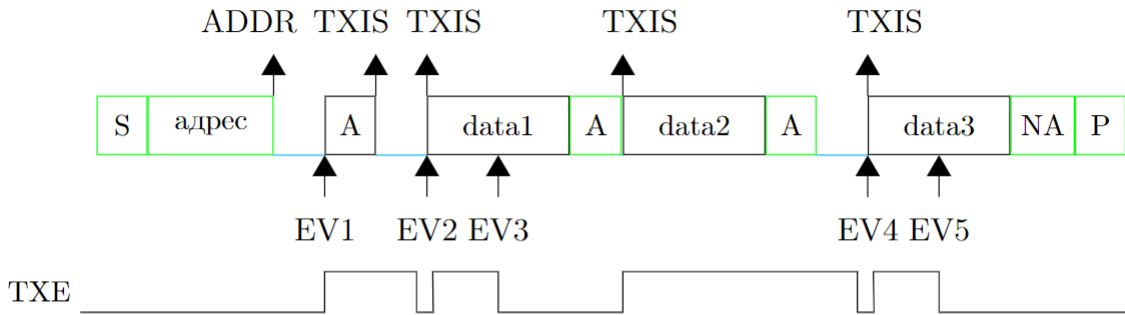
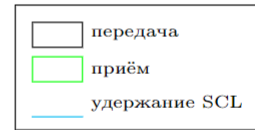


Рисунок 30 - Алгоритм передачи в режиме «ведомый», при NOSTRETCH=1

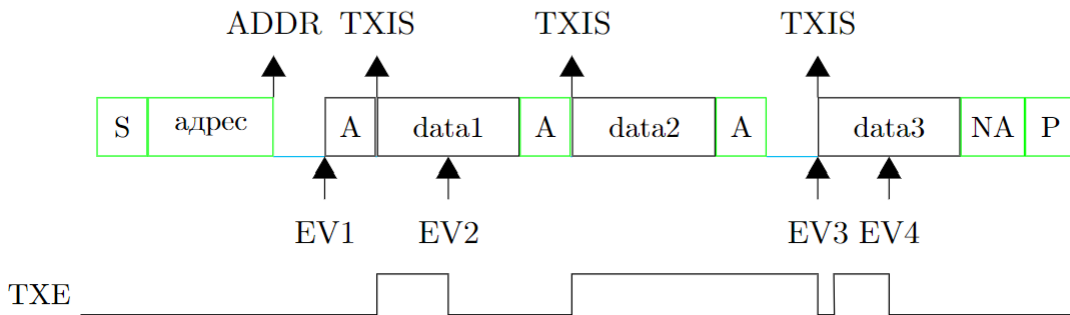
Пример передачи ведомым 3 байт со сбросом первого байта при NOSTRETCH=0:



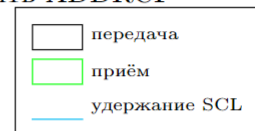
- EV1: ADDR ISR: проверить ADDCODE и DIR, установить TXE и ADDRCF
- EV2: TXIS ISR: запись data1
- EV3: TXIS ISR: запись data2
- EV4: TXIS ISR: запись data3
- EV5: TXIS ISR: запись data4 (не отправлено)



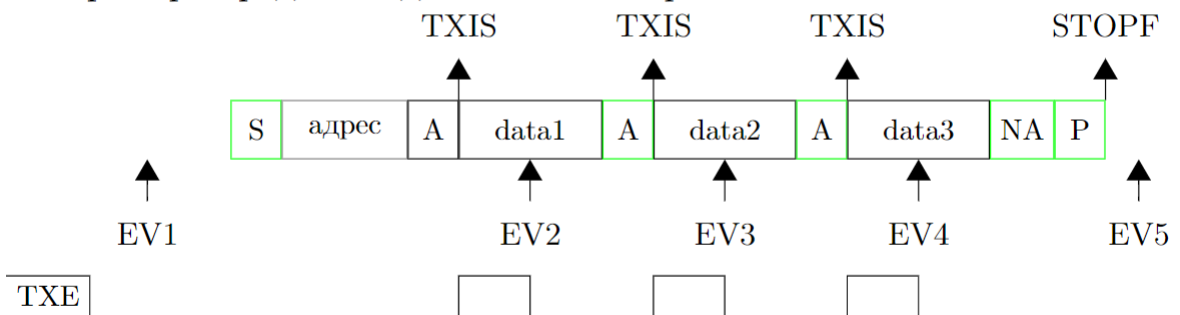
Пример передачи ведомым 3 байт без сброса первого байта при NOSTRETCH=0:



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

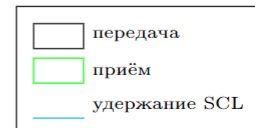


Рисунок 31 – Диаграммы состояния шины I²C и события при передаче данных в режиме «ведомый»

3.8.10.6 Приём в режиме «ведомый»

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

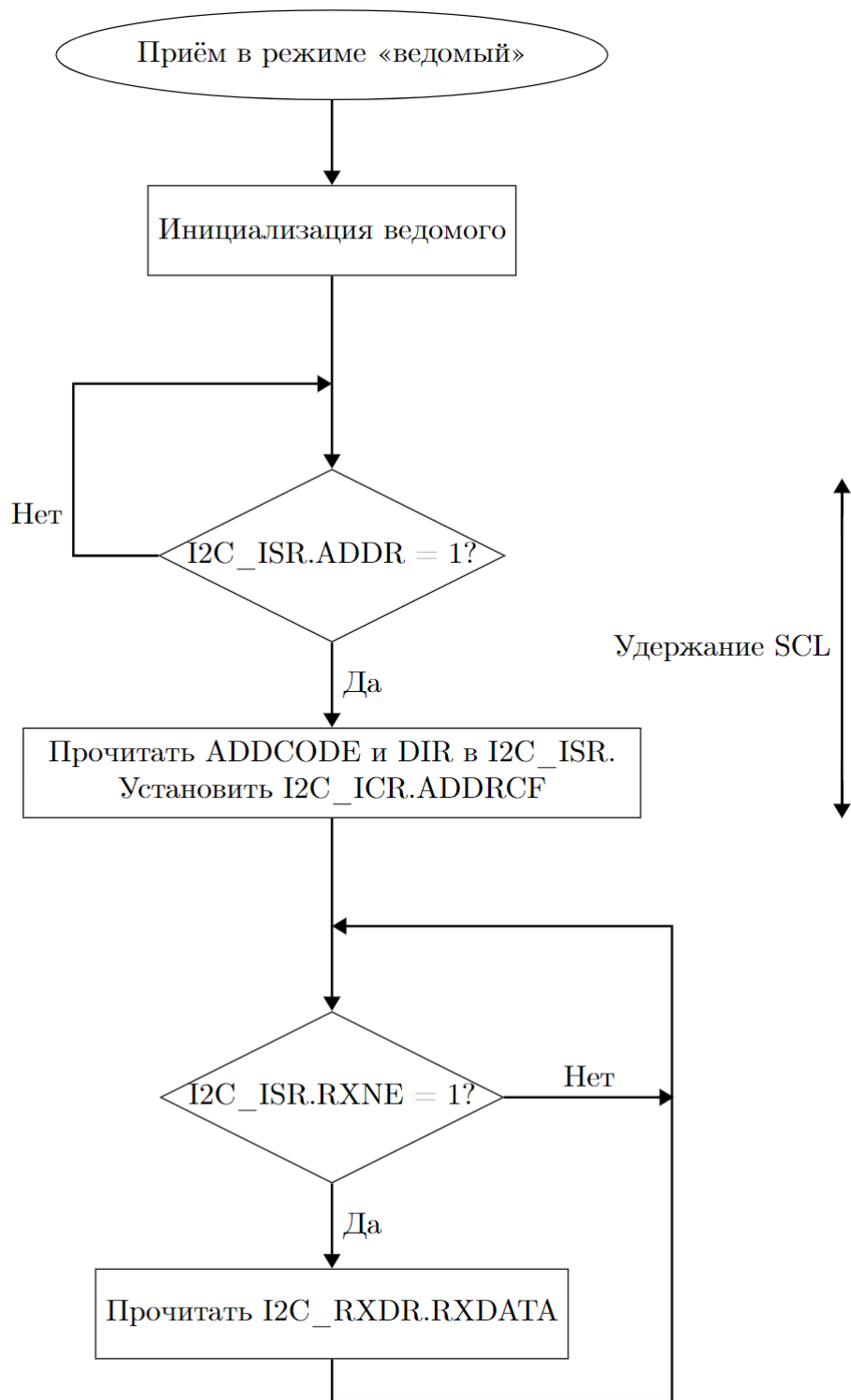


Рисунок 32 - Алгоритм приёма в режиме «ведомый», при NOSTRETCH=0

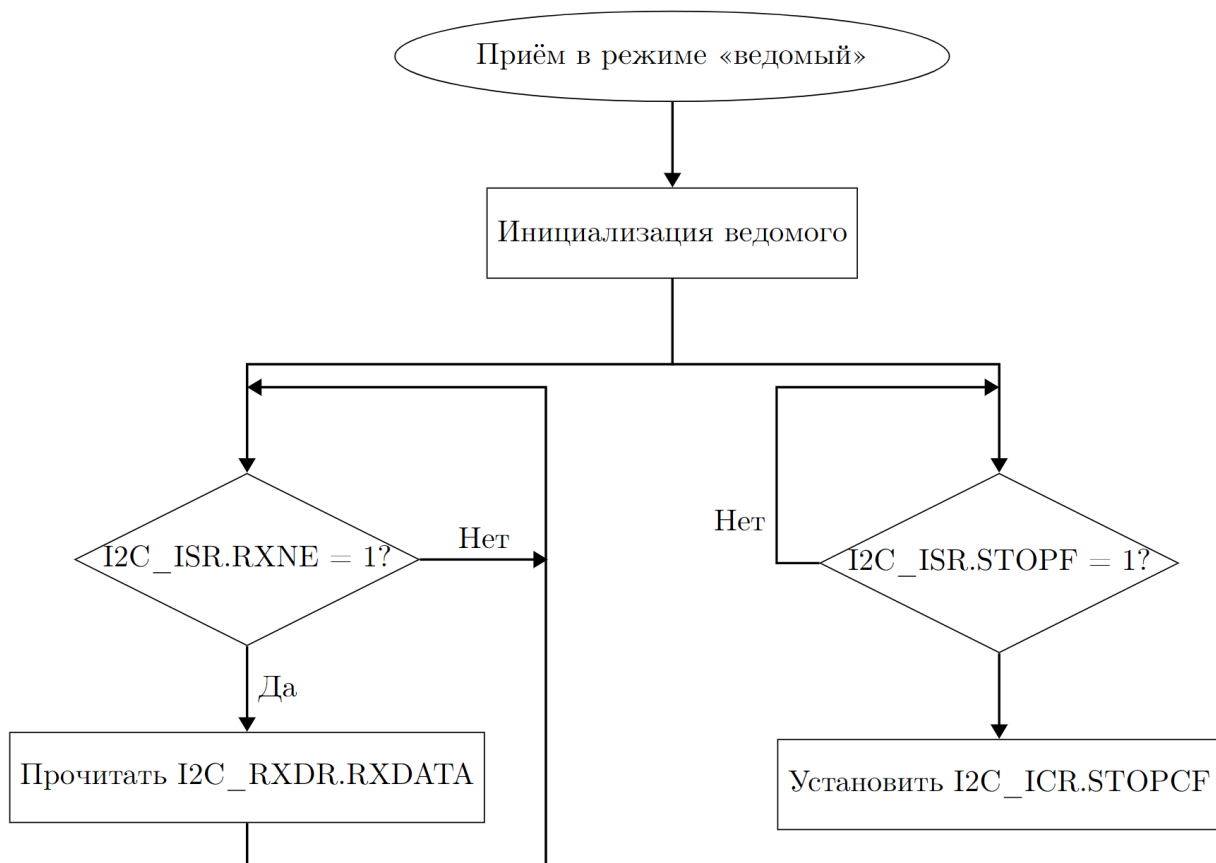
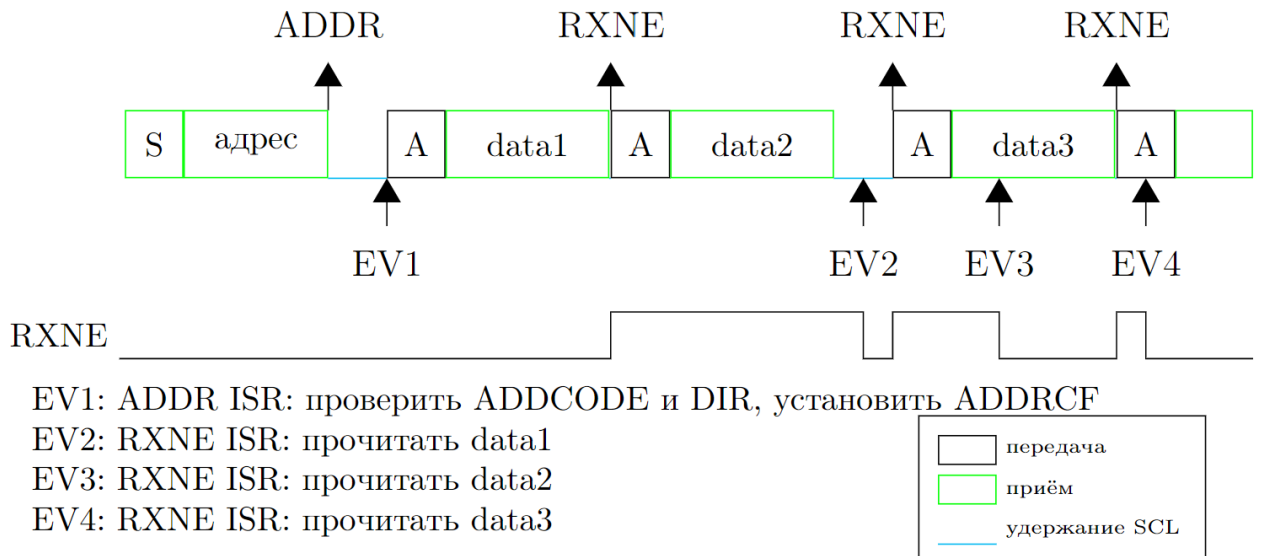


Рисунок 33 - Алгоритм приёма в режиме «ведомый», при NOSTRETCH=1

Когда буфер I2C_RXDR заполнен, взводится флаг RXNE в регистре I2C_ISR и вырабатывается прерывание, если оно разрешено битом RXIE в регистре I2C_CR1. Сброс флага происходит после чтения из регистра I2C_RXDR.

Если получена комбинация STOP и бит STOPIE в регистре I2C_CR1 установлен, то взводится флаг STOPF в регистре I2C_ISR и вырабатывается прерывание (рисунок 34).

Пример приёма ведомым 3 байт при NOSTRETCH=0:



Пример приёма ведомым 3 байт при NOSTRETCH=1:

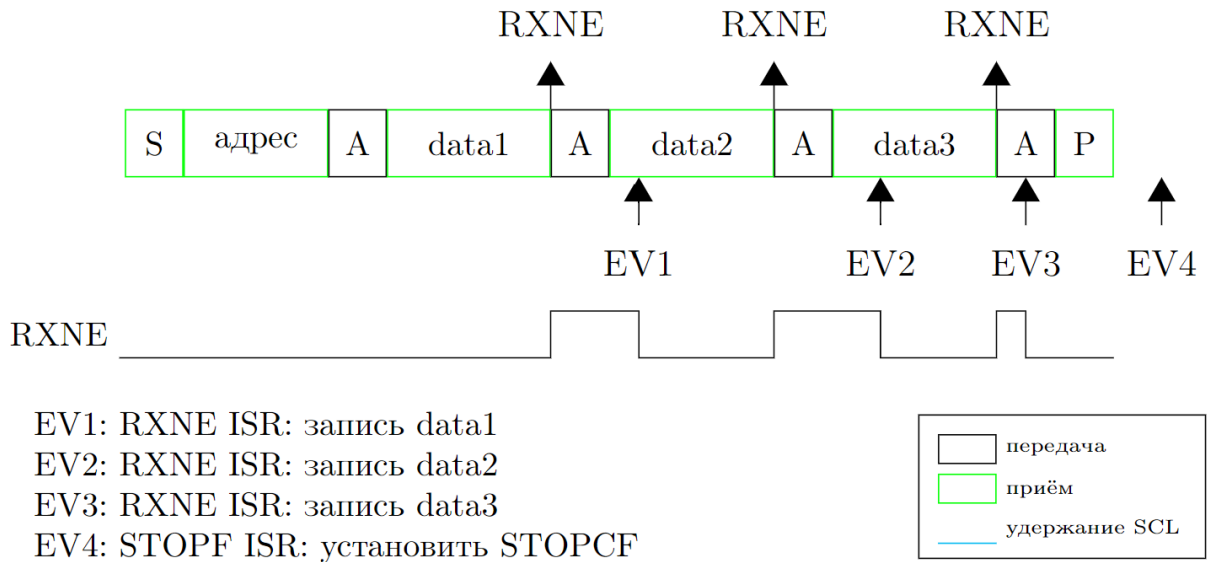


Рисунок 34 – Диаграммы состояния шины I²C и события при приёме данных в режиме «ведомый»

3.8.11 Режим «ведущий» I²C

3.8.11.1 Инициализация ведущего

До включения интерфейса должен быть настроен тактовый сигнал SCL битами SCLH и SCLL в регистре I2C_TIMINGR.

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

– длительность низкого уровня TC отсчитывается счетчиком SCLL, начиная с момента внутреннего обнаружения низкого уровня SCL;

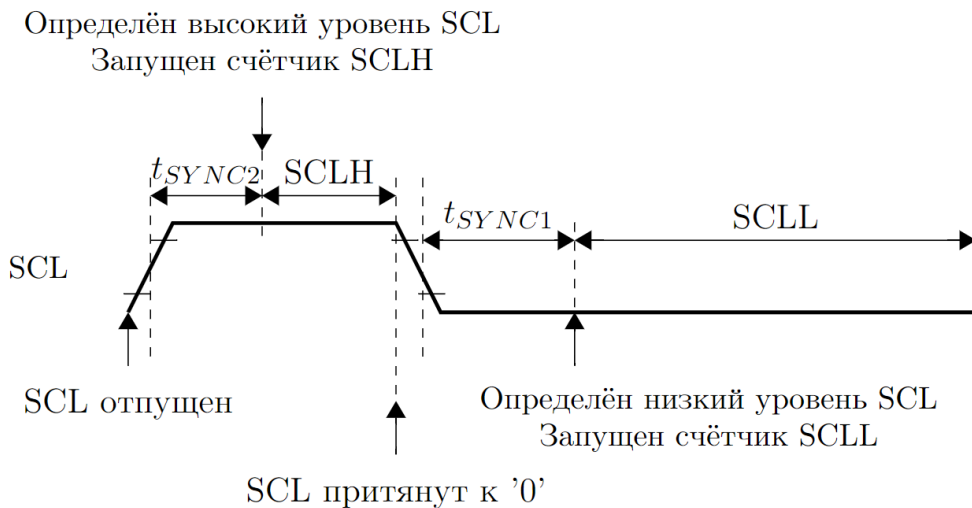
– длительность высокого уровня TC отсчитывается счетчиком SCLH, начиная с момента внутреннего обнаружения высокого уровня SCL.

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 – $\text{DNF} \times t_{\text{I2CCLK}}$), 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 I2C_TIMINGR.

Интерфейс обнаруживает высокий уровень SCL с задержкой t_{SYNC2} , зависящей от длительности фронта SCL, характеристик и настроек фильтра шумов (цифрового – $\text{DNF} \times t_{\text{I2CCLK}}$), 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 I2C_TIMINGR (рисунок 35).

Соответственно, период SCL определяется по формуле 11.

$$t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH}+1) + (\text{SCLL}+1)] \times (\text{PRESC}+1) \times t_{\text{I2CCLK}}\} \quad (11)$$



Синхронизация SCL в режиме "ведущий"

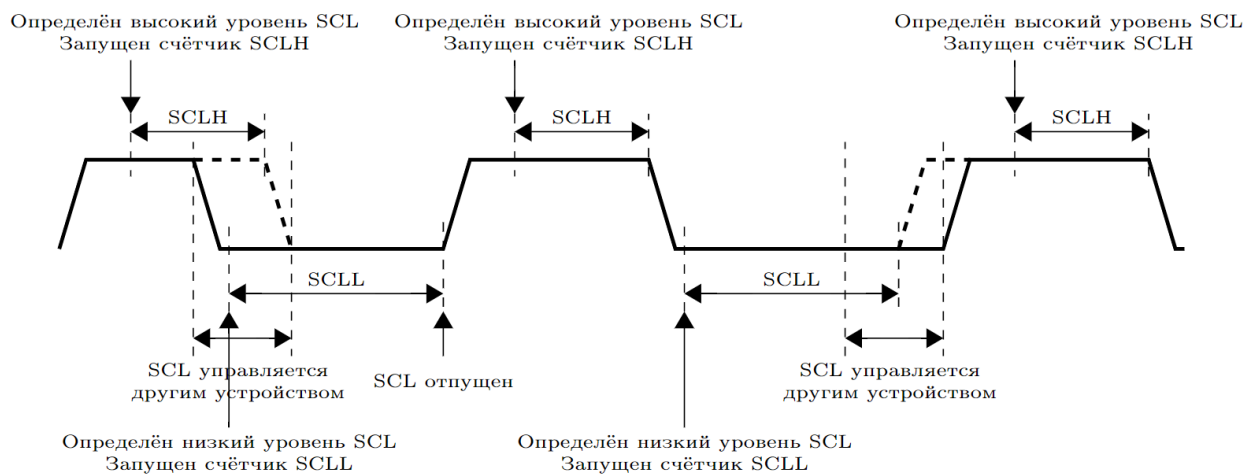


Рисунок 35 – Формирование тактового сигнала SCL в режиме «ведущий»

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. |
| $t_{SU:STO}$ | Длительность предустановки состояния STOP | 4,0 | – | 0,6 | – | 0,26 | – | мкс. |
| t_{BUF} | Время между состояниями STOP и START | 4,7 | – | 1,3 | – | 0,5 | – | мкс. |
| t_{LOW} | Длительность паузы SCL | 4,7 | – | 1,3 | – | 0,5 | – | мкс. |
| t_{HIGH} | Длительность импульса SCL | 4,0 | – | 0,6 | – | 0,26 | – | мкс. |
| t_f | Длительность фронта сигналов SDA, SCL | – | 1000 | – | 300 | – | 120 | нс. |
| t_r | Длительность спада сигналов SDA, SCL | – | 300 | – | 300 | – | 120 | нс. |

Значение SCLL также используется для генерации задержек t_{BUF} и $t_{SU:STA}$. t_{BUF} – время между STOP и START.

Значение SCLH также используется для генерации задержек $t_{HD:STA}$ и $t_{SU:STO}$.

3.8.11.2 Инициализация соединения ведущим (фаза адреса)

Перед соединением в регистре I2C_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 I2C_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 ADDRCONF 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 ADDRCONF.

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

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

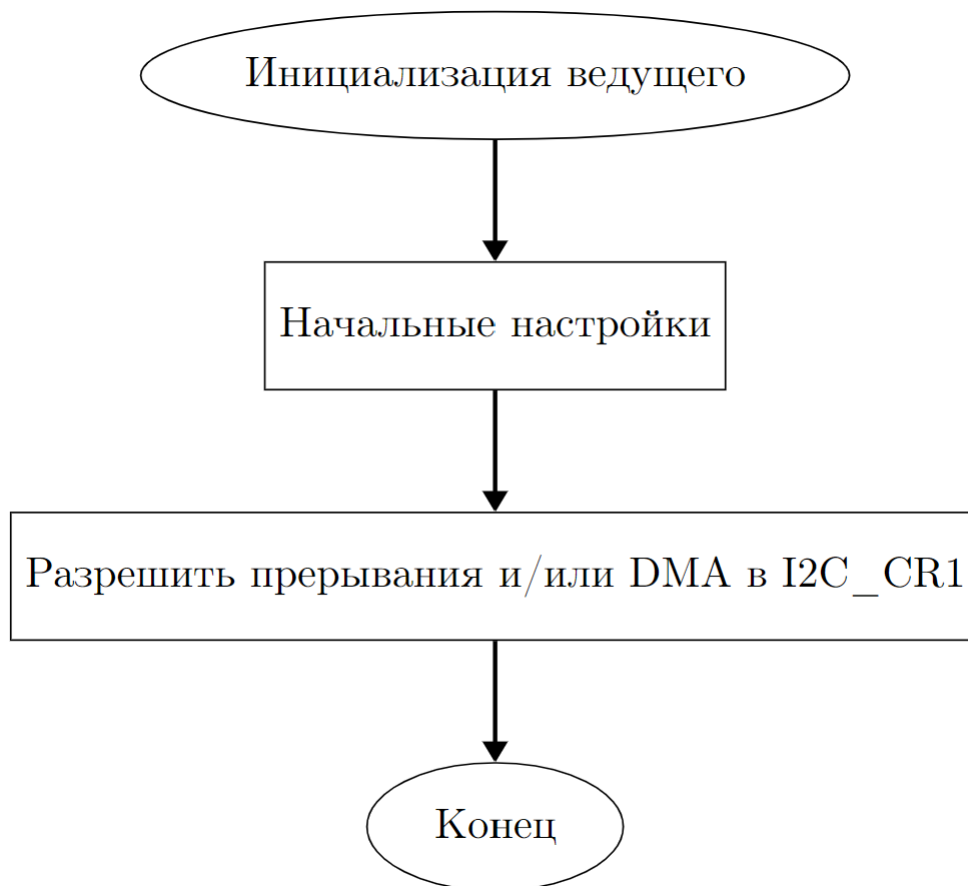


Рисунок 36 - Инициализация в режиме «ведущий»

Инициализация приёма ведущим, адресация ведомого, 10 бит:

– если адрес ведомого 10 бит, то пользователь может отправить полную последовательность чтения, очистив бит HEAD10R в регистре I2C_CR2. После установки бита START, ведущий автоматически отправит последовательность: (Re)Start + Заголовок 10-битного адрес ведомого с записью + Второй байт адреса ведомого + ReStart + Заголовок 10-битного адрес 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 адрес ведомого с чтением (рисунок 38).

1) может быть запрошена комбинация RESTART установкой бита START в регистре I2C_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 I2C_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 I2C_ISR и вырабатывается прерывание, если установлен бит NACKIE (рисунок 41).

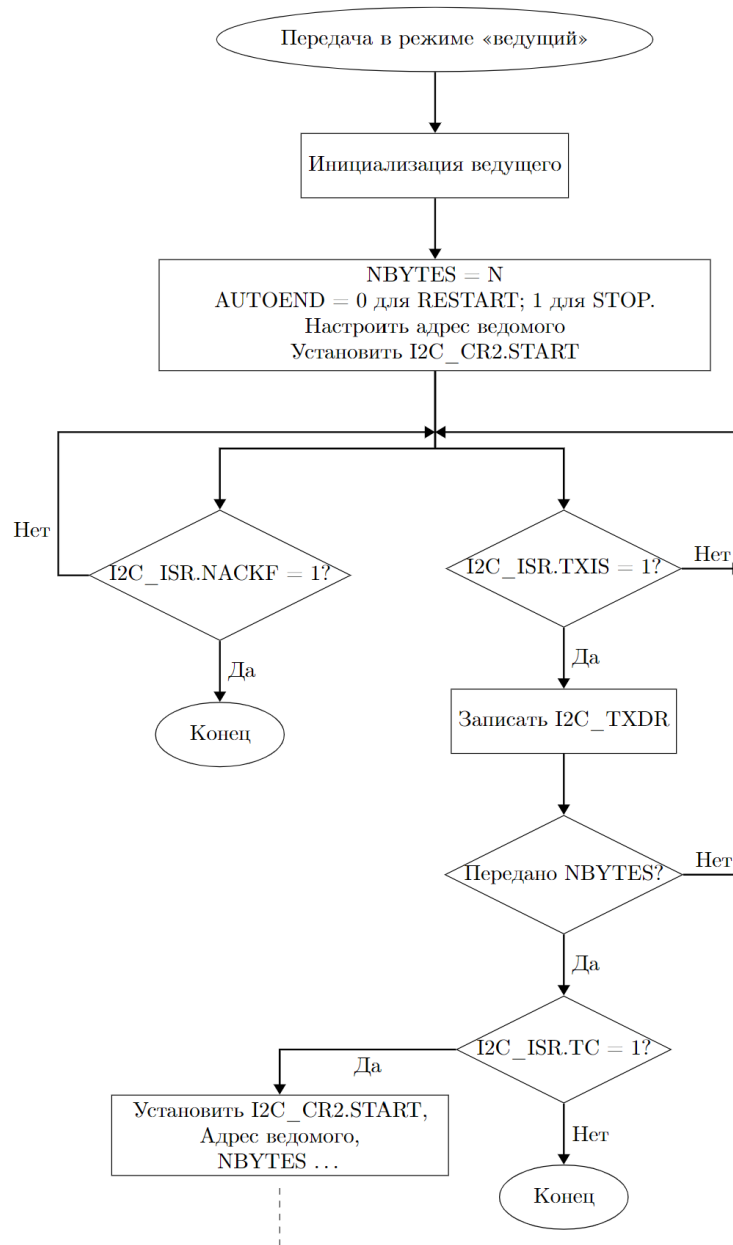


Рисунок 39 - Передача в режиме «ведущий» при $N \leq 255$ байт

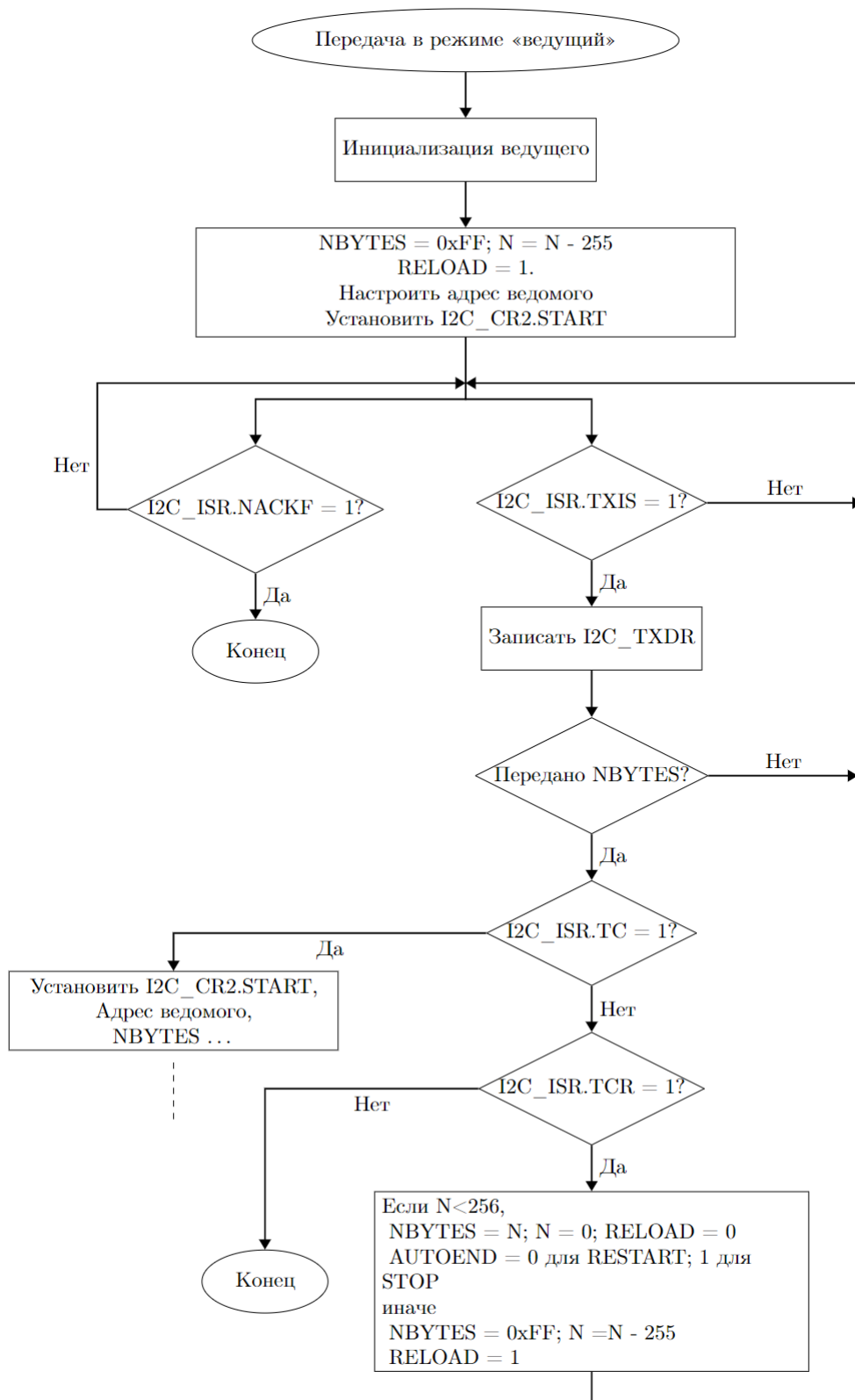
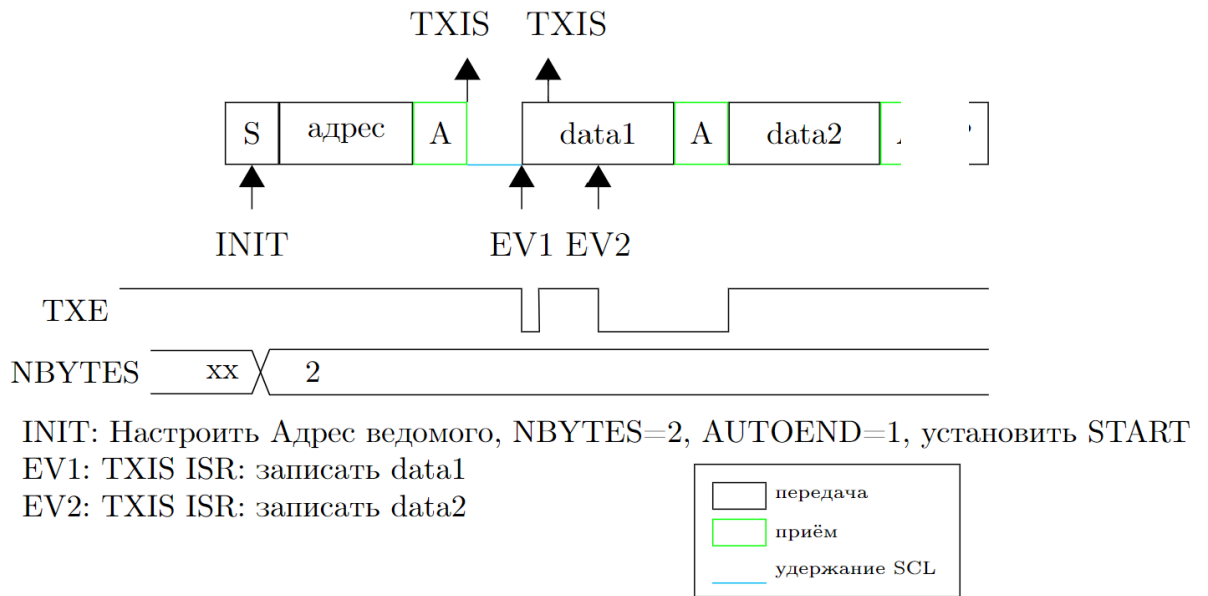


Рисунок 40 - Передача в режиме «ведущий» при $N > 255$ байт

Пример передачи ведущим 2 байт с автозавершением (STOP):



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

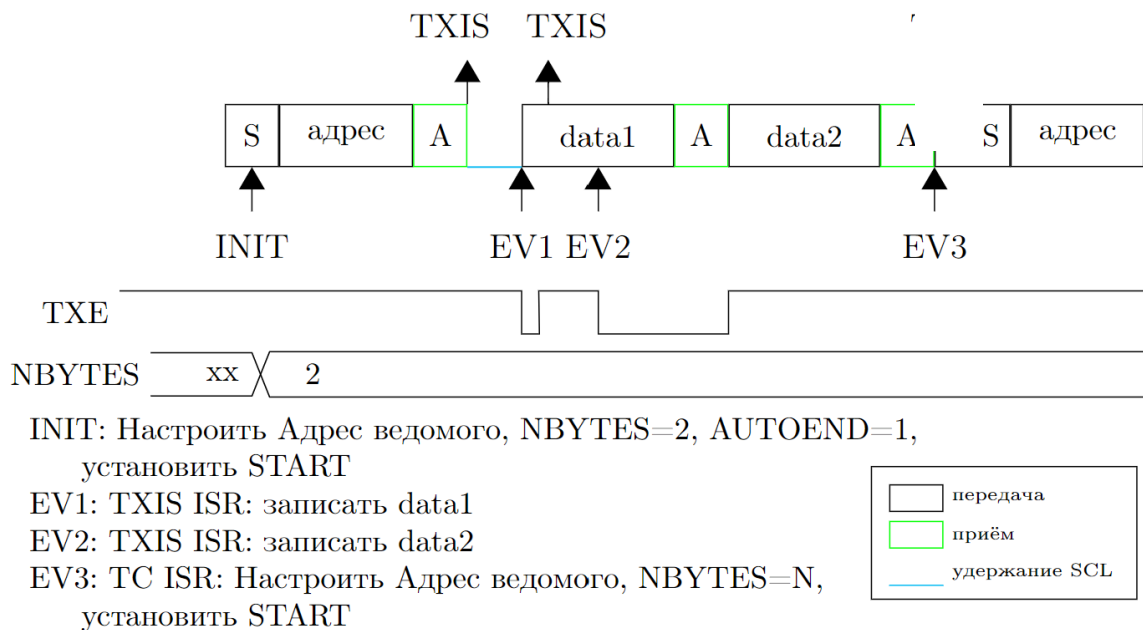


Рисунок 41 – Передача данных ведущим

3.8.11.4 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 I2C_CR1. Флаг сбрасывается после чтения из регистра I2C_RXDR.

Если общее количество байт для приёма больше 255, устанавливается бит RELOAD в регистре I2C_CR2. В этом случае после передачи всех NBYTES данных, взводится флаг TCR и линия SCL удерживается пока в NBYTES[7:0] не будет записано ненулевое значение.

Если RELOAD = 0, и передаются NBYTES[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 следующих действий (рисунок 44):

1) может быть запрошена комбинация RESTART установкой бита START в регистре I2C_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 I2C_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.

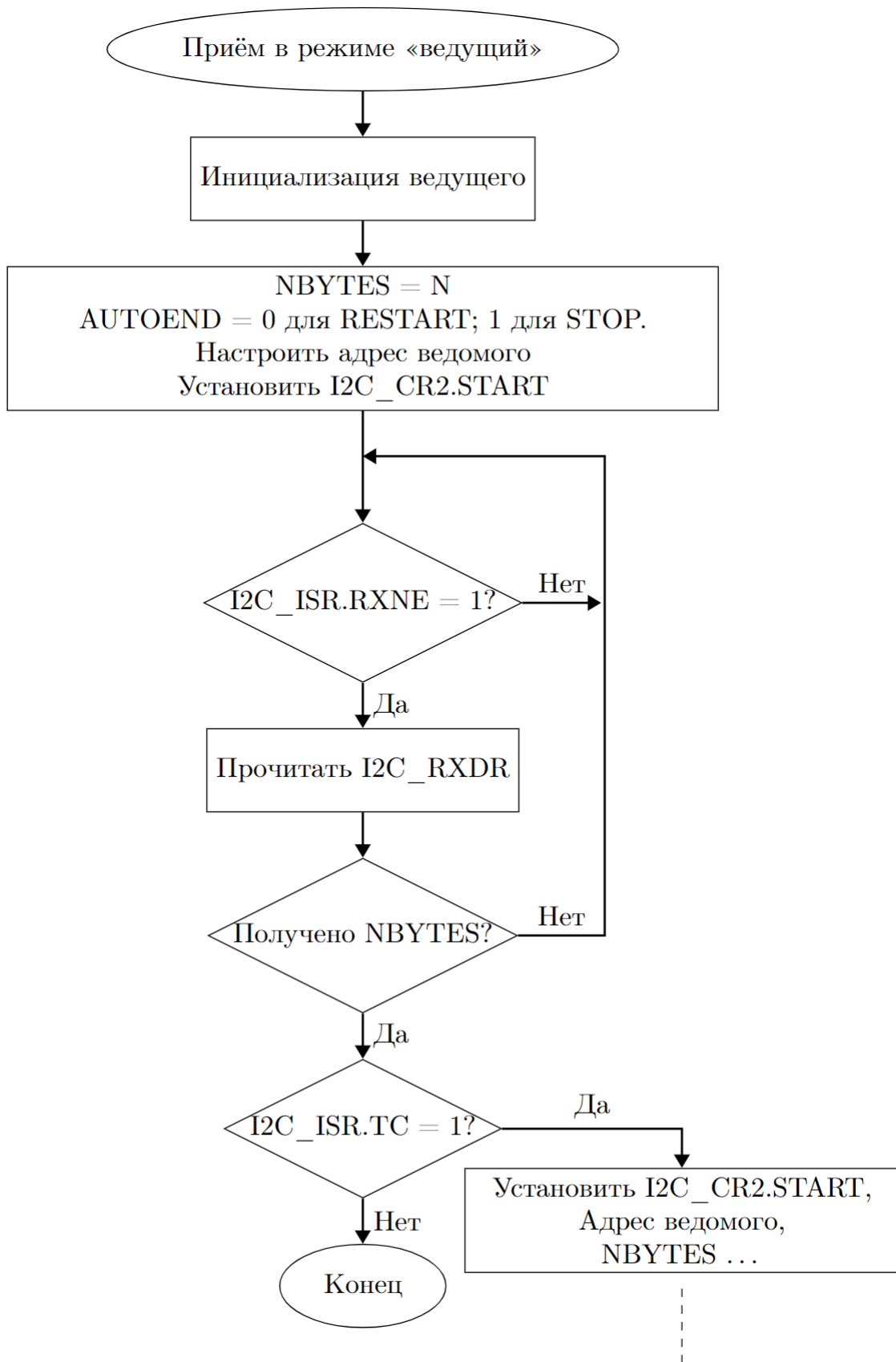


Рисунок 42 - Приём в режиме «ведущий» при $N \leq 255$ байт

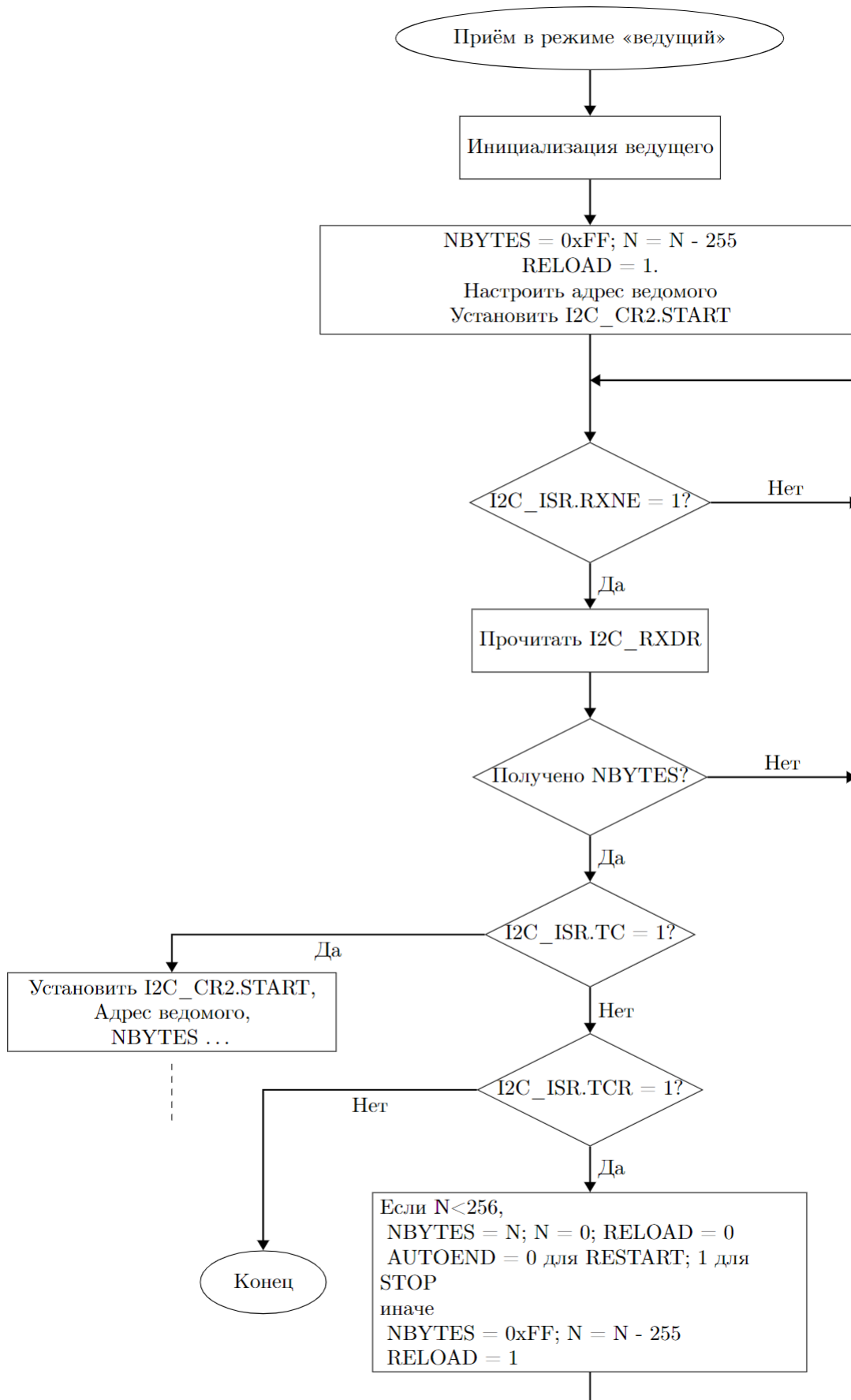
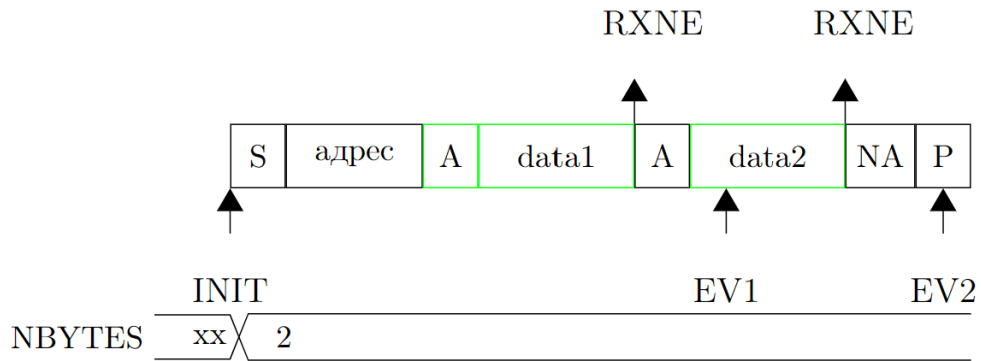


Рисунок 43 - Приём в режиме «ведущий» при N > 255 байт

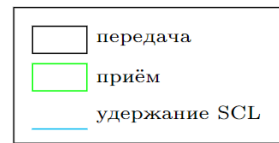
Пример приёма ведущим 2 байт с автозавершением (STOP):



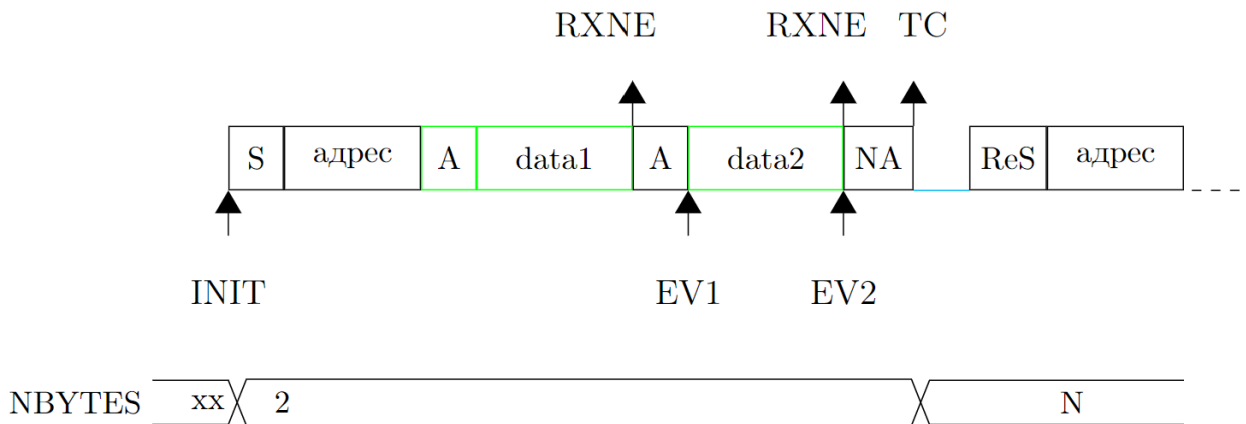
INIT: Настроить Адрес ведомого, NBYTES=2, AUTOEND=1,
установить START

EV1: RXNE ISR: прочитать data1

EV2: RXNE ISR: прочитать data2



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



INIT: Настроить Адрес ведомого, NBYTES=2, AUTOEND=0,
установить START

EV1: RXNE ISR: прочитать data1

EV2: RXNE ISR: прочитать data2

EV3: TC ISR: Настроить Адрес ведомого, NBYTES=N,
установить START

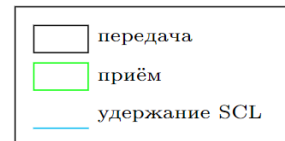


Рисунок 44 – Приём данных ведущим

3.8.12 Ошибки

3.8.12.1 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 I2C_ISR и вырабатывается прерывание, если в регистре I2C_CR1 установлен бит ERRIE.

3.8.12.2 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 I2C_ISR взводится флаг ARLO и вырабатывается прерывание, если в регистре I2C_CR1 установлен бит ERRIE.

3.8.12.3 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 I2C_TXDR, если TXE=0, иначе 0xFF;

2) новый байт должен быть отправлен и регистр I2C_TXDR всё ещё не записан, отправляется 0xFF.

В случае ошибки OVR в регистре I2C_ISR взводится флаг OVR и вырабатывается прерывание, если бит ERRIE установлен в регистре I2C_CR1.

3.8.13 DMA запросы

3.8.13.1 Передача с использованием DMA

Для передачи данных с использованием DMA (Direct Memory Access) необходимо установить бит TXDMAEN в регистре I2C_CR1. Данные из области памяти, настроенной для использования контроллером DMA, загружаются в регистр I2C_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.

3.8.13.2 Приём с использованием DMA

Для приёма данных с использованием DMA (Direct Memory Access) необходимо установить бит RXDMAEN в регистре I2C_CR1. Данные из регистра I2C_RXDR, загружаются в область памяти, настроенной для использования DMA периферией 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 на шине с внутренним адресом или в обработчике прерывания ADDR до сброса флага ADDR.

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

3.8.14 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 |

3.8.15 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 I2C_RXDR | RXIE |
| Статус буфера передачи | TXIS | Запись регистра I2C_TXDR | TXIE |
| Обнаружение STOP на линии | STOPF | Запись STOPCF=1 | STOPIE |
| Конец передачи RELOAD=1 | TCR | Запись I2C_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 ADDRDCF=1 | ADDRIE |
| Не получено подтверждение (NACK) | NACKF | Запись NACKCF=1 | NACKIE |
| 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 выработки прерываний.

На рисунке 45 изображена схема формирования прерываний I²C.

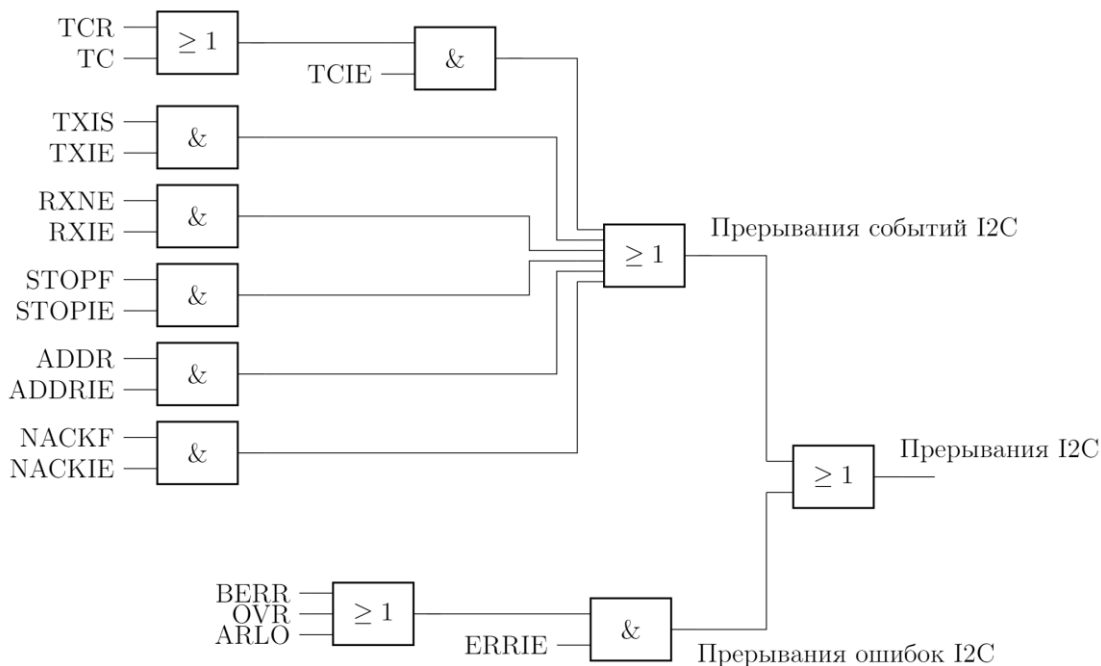


Рисунок 45 – Схема формирования прерываний

3.8.16 Регистры I²C

Регистры модуля I²C представлены в таблице 31.

Таблица 31 – Регистры модуля I²C

| Адрес | Регистр | Тип |
|-------|---|-----|
| 0x00 | Регистр управления 1 (I2C_CR1) | R/W |
| 0x04 | Регистр управления 2 (I2C_CR2) | R/W |
| 0x08 | Регистр адреса 1 (I2C_OAR1) | R/W |
| 0x0C | Регистр адреса 2 (I2C_OAR2) | R/W |
| 0x10 | Регистр настройки временных ограничений (I2C_TIMINGR) | R/W |
| 0x14 | Резерв | R/O |
| 0x18 | Регистр флагов прерываний (I2C_ISR) | R/W |
| 0x1C | Регистр сброса флагов прерываний (I2C_ICR) | R/W |
| 0x20 | Регистр принятых данных (I2C_RXDR) | R/W |
| 0x24 | Регистр передаваемых данных (I2C_TXDR) | R/W |

Доступ к регистрам осуществляется словами (32 бита). Поля регистров представлены в таблицах 64 – 80.

3.8.16.1 Регистр управления 1 (I2C_CR1)

В таблице 32 приведены поля регистра управления I2C_CR1.

Таблица 32 – Регистр управления 1 (I2C_CR1)

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| Резерв | 31:20 | 0x0 | Резерв | – |
| GCEN | 19 | 0x0 | Разрешение адреса общего вызова: 0 – Адрес 0b00000000 запрещен, формируется NACK; 1 – Адрес 0b00000000 разрешен, формируется ACK | RW |
| Резерв | 18 | 0x0 | – | – |
| 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 |
| Резерв | 13 | 0x0 | – | – |
| ANFOFF | 12 | 0x0 | Управление аналоговым фильтром шумов: 0 – фильтр включен 1 – фильтр выключен | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| DNF | 11:8 | 0x0 | Управление цифровым фильтром шумов: 4'b0000 – цифровой фильтр выключен; 4'b0001 – цифровой фильтр установлен на 1 такт I ² CCLK; 4'b0010 – цифровой фильтр установлен на 2 такта I ² CCLK; ... 4'b1111 – цифровой фильтр установлен на 15 тактов I ² CCLK Изменение значения допускается только при выключенном блоке (PE=0) | RW |
| 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| 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 |
| 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 |

3.8.16.2 Регистр управления 2 (I2C_CR2)

В таблице 33 приведены поля регистра управления I2C_CR2.

Таблица 33 – Регистр управления 2 (I2C_CR2)

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| Резерв | 31:26 | 0x0 | Резерв | – |
| 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 и в режиме «ведомый» | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| 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 |
| NACK | 15 | 0x0 | Формирование NACK в режиме «ведомый». Устанавливается программно, сбрасывается аппаратно: <ul style="list-style-type: none"> • после отправки NACK; • по событию STOP на шине; • при получении своего адреса ведомого; • PE=0. 0 – отправка ACK после приёма текущего байта; 1 – отправка NACK после приёма текущего байта | RS |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| 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> | RS |
| START | 13 | 0x0 | <p>Формирование START. Устанавливается программно, сбрасывается аппаратно после отправки адреса, потере 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> | RS |
| 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 + заголовков 10-битного адреса (чтение);</p> <p>1 – ведущий отправляет только заголовков 10-битного адреса (чтение).</p> <p>Изменение этого бита при START=1 не допускается.</p> | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| ADD10 | 11 | 0x0 | Режим 10-битного адреса (режим «ведущий»): 0 – ведущий работает в режиме 7-битного адреса; 1 – ведущий работает в режиме 10-битного адреса. Изменение этого бита при START=1 не допускается. | RW |
| RD_WRN | 10 | 0x0 | Направление передачи (режим «ведущий»): 0 – ведущий запрашивает транзакцию записи; 1 – ведущий запрашивает транзакцию чтения Изменение этого бита при START=1 не допускается. | RW |
| SADD[9:8] | 9:8 | 0x0 | Адрес ведомого (режим «ведущий»): В режиме 7-битного адреса (ADD10=0) содержимое этих битов не имеет значения. В режиме 10-битного адреса (ADD10=1) в эти биты должны быть записаны биты 9:8 адреса ведомого. Изменение этих битов при START=1 не допускается. | RW |
| SADD[7:1] | 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| SADD[0] | 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 |

3.8.16.3 Регистр собственного адреса 1 (I2C_OAR1)

В таблице 34 приведены поля регистра I2C_OAR1.

Таблица 34 – Регистр собственного адреса 1 (I2C_OAR1)

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| Резерв | 31:16 | 0x0 | Резерв | – |
| OAIEN | 15 | 0x0 | Использование собственного адреса OA1: 0 – при получении адреса OA1 формируется NACK; 1 – при получении адреса OA1 формируется ACK | RW |
| Резерв | 14:11 | 0x0 | Резерв | – |
| OAI1MODE | 10 | 0x0 | Режим 10-битного адреса OA1: 0 – OA1 – 7-битный адрес; 1 – OA1 – 10-битный адрес Изменение бита допускается при OAIEN=0 | RW |
| OAI1[9:8] | 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 OAIEN=0 | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| OA1[7:1] | 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] | 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 |

3.8.16.4 Регистр собственного адреса 2 (I2C_OAR2)

В таблице 35 приведены поля регистра I2C_OAR2.

Таблица 35 – Регистр собственного адреса 2 (I2C_OAR2)

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

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| OA2MSK | 10:8 | 0x0 | <p>Маска адреса OA2:</p> <p>000 – Нет маски</p> <p>001 – сравниваются только OA2[7:2];</p> <p>010 – сравниваются только OA2[7:3];</p> <p>011 – сравниваются только OA2[7:4];</p> <p>100 – сравниваются только OA2[7:5];</p> <p>101 – сравниваются только OA2[7:6];</p> <p>110 – сравниваются только OA2[7];</p> <p>111 – OA2[7:1] маскируются, подтверждаются (ACK) все 7-битные адреса (кроме зарезервированных)</p> <p>Если OA2MSK \neq 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 |
| Резерв | 0 | 0x0 | – | – |

3.8.16.5 Регистр временные ограничения (I2C_TIMINGR)

В таблице 36 приведены поля регистра I2C_TIMINGR.

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

Таблица 36 – Регистр временные ограничения (I2C_TIMINGR)

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| PRESC | 31:28 | 0x0 | <p>Предварительный делитель частоты I²CCLK. Используется для вычисления значения t_{PRESC} используемого счетчиками предустановки, удержания, низкого и высокого уровней.</p> <p>$t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$</p> | RW |
| Резерв | 27:24 | 0x0 | Резерв | – |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| SCLDEL | 23:20 | 0x0 | Длительность предустановки данных $t_{SU:DAT}$. Задержка между изменением SDA и фронтом SCL. $t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}$ 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 $NOSTRETCH = 0$ $t_{SCADEL} = (SCADEL+1) \times t_{PRESC}$ Используется для генерации задержек $t_{HD:DAT}$ | RW |
| SCLH | 15:8 | 0x0 | Длительность удержания SCL в состоянии логической «1» в режиме «ведущий» $t_{SCLH} = (SCLH+1) \times t_{PRESC}$ 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» $t_{SCLL} = (SCLL+1) \times t_{PRESC}$ Также используется для генерации задержек t_{BUF} и $t_{SU:STA}$ | RW |

3.8.16.6 Регистр прерываний и статуса (I2C_ISR)

В таблице 37 приведены поля регистра I2C_ISR.

Таблица 37 – Регистр прерываний и статуса (I2C_ISR)

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|------------|--------|
| Резерв | 31:24 | 0x0 | Резерв | – |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| ADDCODE | 23:17 | 0x0 | Код совпавшего адреса. Обновляется в режиме «ведомый» при совпадении адреса (ADDR=1). В режиме 10-битного адреса содержит заголовок (5b11110) и два старших бита адреса. | R |
| 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. | R |
| BUSY | 15 | 0x0 | Флаг индикации занятой шины. Устанавливается после события START на шине и сбрасывается после события STOP на шине | R |
| Резерв | 14:11 | 0x0 | Резерв | – |
| OVR | 10 | 0x0 | Флаг переполнения/недозагрузки (режим «ведомый» при NOSTRETCH=1) Устанавливается аппаратно, сбрасывается программной установкой бита OVRCF или аппаратно при PE=0. | R |
| ARLO | 9 | 0x0 | Флаг проигрыша арбитража. Устанавливается аппаратно, сбрасывается программной установкой бита ARLOCF или аппаратно при PE=0. | R |

| 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. | R |
| 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. | R |
| 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. | R |
| 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. | R |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| NACKF | 4 | 0x0 | Флаг «не получено подтверждение» (NACK). Устанавливается аппаратно, после передачи байта. Сбрасывается программной установкой бита NACKCF или аппаратно при PE=0. | R |
| ADDR | 3 | 0x0 | Флаг соответствия адреса (режим «ведомый») Устанавливается аппаратно, если полученный адрес совпадает с одним из разрешенных в I2C_OAR1, I2C_OAR2. Сбрасывается программной установкой бита ADDRCF или аппаратно при PE=0. | R |
| RXNE | 2 | 0x0 | Флаг «буфер принятых данных заполнен» (режимы приёма). Устанавливается аппаратно, после записи принятых данных в регистр I2C_RXDR. Сбрасывается при чтении I2C_RXDR или аппаратно при PE=0. | RS |
| TXIS | 1 | 0x0 | Состояние прерывания передатчика. Устанавливается аппаратно, когда регистр I2C_TXDR пуст и следующий байт данных должен быть в него записан. Сбрасывается записью следующего байта данных в регистр I2C_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) | RS |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| TXE | 0 | 0x0 | Флаг «буфер передаваемых данных» I2C_TXDR пуст (режимы отправки). Устанавливается аппаратно, если буфер пуст; при PE=0 или программно, чтобы сбросить содержимое регистра I2C_TXDR. Сбрасывается записью следующего байта данных в регистр I2C_TXDR. | |

3.8.16.7 Регистр сброса прерываний (I2C_ICR)

В таблице 38 приведены поля регистра I2C_ISR. Сброс каждого флага производится записью '1' в соответствующий бит регистра.

Т а б л и ц а 38 – Регистр сброса прерываний (I2C_ICR)

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| Резерв | 31:11 | 0x0 | Резерв | – |
| OVRDCF | 10 | 0x0 | Сброс флага прерывания переполнения/недозагрузки | W |
| ARLOCF | 9 | 0x0 | Сброс флага проигрыша арбитража | W |
| BERRDCF | 8 | 0x0 | Сброс флага ошибки шины | W |
| Резерв | 7:6 | 0x0 | – | – |
| STOPCF | 5 | 0x0 | Сброс флага детектирования STOP на шине | W |
| NACKCF | 4 | 0x0 | Сброс флага «не получено подтверждение» (NACK) | W |
| ADDRDCF | 3 | 0x0 | Сброс флага соответствия адреса | W |
| Резерв | 2:0 | 0x0 | – | – |

3.8.16.8 Регистр принятых данных (I2C_RXDR)

В таблице 39 приведены поля регистра I2C_RXDR.

Т а б л и ц а 39 – Регистр принятых данных (I2C_RXDR)

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|-----------------------|--------|
| Резерв | 31:8 | 0x0 | Резерв | – |
| RXDATA | 7:0 | 0x0 | Буфер принятых данных | R |

3.8.16.9 Регистр передаваемых данных (I2C_TXDR)

В таблице 40 приведены поля регистра I2C_TXDR.

Т а б л и ц а 40 – Регистр передаваемых данных (I2C_TXDR)

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---------------------------|--------|
| Резерв | 31:8 | 0x0 | Резерв | – |
| TXDATA | 7:0 | 0x0 | Буфер передаваемых данных | RW |

3.9 Модуль 32-разрядного таймера

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.

3.9.1 Регистры модуля таймера

В таблице 41 перечислены регистры модуля таймера.

Таблица 41 – Регистры модуля таймера

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|---|----------|--------------------|--|--------|
| VALUE | 0x00 | 0x0 | Текущее значение основного таймера | RO |
| TOP | 0x04 | 0xFFFFFFFF | Максимальное значение счётной последовательности | RW |
| PRESCALE | 0x08 | 0x0 | Значение делителя | RW |
| CONTROL | 0x0C | 0x0 | Конфигурационный регистр основного таймера | RW |
| ENABLE | 0x10 | 0x0 | Регистр включения таймера | RW |
| INT_MASK | 0x14 | 0x0 | Регистр маски прерываний | RW |
| INT_CLEAR | 0x18 | 0xFFFFFFFF | Регистр сброса флагов прерываний | W1C |
| INT_FLAG | 0x1C | 0x0 | Регистр флагов прерываний | RO |
| CNTR_CH1 | 0x80 | 0x0 | Конфигурационный регистр 1 канала | RW |
| OCR_CH1 | 0x84 | 0x0 | Значение сравнения 1 канала | RW |
| ICR_CH1 | 0x88 | 0x0 | Значение захвата 1 канала | RW |
| CNTR_CH2 | 0x90 | 0x0 | Конфигурационный регистр 2 канала | RW |
| OCR_CH2 | 0x94 | 0x0 | Значение сравнения 2 канала | RW |
| ICR_CH2 | 0x98 | 0x0 | Значение захвата 2 канала | RW |
| CNTR_CH3 | 0xA0 | 0x0 | Конфигурационный регистр 3 канала | RW |
| OCR_CH3 | 0xA4 | 0x0 | Значение сравнения 3 канала | RW |
| ICR_CH3 | 0xA8 | 0x0 | Значение захвата 3 канала | RW |
| CNTR_CH4 | 0xB0 | 0x0 | Конфигурационный регистр 4 канала | RW |
| OCR_CH4 | 0xB4 | 0x0 | Значение сравнения 4 канала | RW |
| ICR_CH4 | 0xB8 | 0x0 | Значение захвата 4 канала | RW |
| Примечание – Регистры с адресами 0x80 – 0xB8 не доступны для Таймер 32_0. | | | | |

Назначение полей регистров представлены в таблицах 42 – 49.

Таблица 42 – Регистр VALUE модуля таймера

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---------------------------|--------|
| TIM_VAL | 31:0 | 0x0 | Текущее значение счетчика | RO |

Таблица 43 – Регистр TOP модуля таймера

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| TIM_TOP | 31:0 | 32'hFFFFFFFF | Максимальное значение счетчика (ограничивает счетную последовательность сверху) | RW |

Таблица 44 – Регистр PRESCALE модуля таймера

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|--------------|---------|--------------------|--|--------|
| TIM_PRESCALE | 31:0 | 0x0 | Значение предварительного делителя. Предделитель вырабатывает тактовый сигнал для счета, частота которого в целое раз меньше входной | RW |

Таблица 45 – Регистр CONTROL модуля таймера

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| – | 31:4 | – | Зарезервировано | – |
| SOURCE | 3:2 | 0x0 | Выбор источника тактового сигнала для счета: 00 – выход предделителя; 01 – вход модуля TIM1; 10 – вход микросхемы timerX_Tx_pin_pad (X – номер таймера); 11 – вход модуля TIM2 | RW |
| COUNT_MODE | 1:0 | 0x0 | Режим счёта таймера: 00 – прямой; 01 – обратный; 10 – двунаправленный | RW |

Таблица 46 – Регистр ENABLE модуля таймера

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

Таблица 47 – Регистр CNx_CNTRL модуля таймера

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| – | 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 | RW |
| MODE | 6:5 | 0x0 | Режим работы канала: 00 – зарезервировано; 01 – режим сравнения (для timer0); 10 – режим захвата (для timer0); 11 – ШИМ | RW |
| EDGE | 4 | 0x0 | Режим сигнала захвата: 0 – фронт; 1 – срез | RW |
| – | 3:1 | – | Зарезервировано | – |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| NOISE | 0 | 0x0 | Управление фильтрацией входных помех на входе ic_port: 0 – фильтрация выключена; 1 – фильтрация выключена | RW |

Таблица 48 – Регистр CHy_OCR модуля таймера

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

Таблица 49 – Регистр CHy_ICR модуля таймера

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

3.9.2 Блок сравнения, захвата, ШИМ

Блок ШИМ обеспечивает дополнительные режимы работы таймеров:

- таймер 0 – режимы сравнения, захвата, ШИМ;
- таймер (1 – 3) – режим ШИМ.

Блок таймера 0 может работать в одном из заявленных режимов и тактируется от выхода 8-битного предделителя. Каждый канал имеет свой конфигурационный регистр и полностью независим от других каналов. Конфигурирование канала осуществляется при помощи регистра CHx_CNTRL (где x – номер канала). Значение сравнения хранится в регистре TIMx_CHy_OCR. Значение захвата хранится в регистре TIMx_CHy_ICR.

3.9.3 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 TIMx_INTF. Для маскирования прерываний необходимо записать 1 в бит, соответствующий данному прерыванию в регистре масок TIMx_INT_MSK. Назначение полей регистров прерываний указано в таблицах 50 и 51.

Таблица 50 – Регистр INT_FLAG модуля таймера

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| – | 31:10 | – | Зарезервировано | – |
| OC_Int_CH4 | 9 | 0x0 | Статус прерывания совпадения 4 канала таймера | R |
| OC_Int_CH3 | 8 | 0x0 | Статус прерывания совпадения 3 канала таймера | R |
| OC_Int_CH2 | 7 | 0x0 | Статус прерывания совпадения 2 канала таймера | R |
| OC_Int_CH1 | 6 | 0x0 | Статус прерывания совпадения 1 канала таймера | R |
| IC_Int_CH4 | 5 | 0x0 | Статус прерывания захвата 4 канала таймера | R |
| IC_Int_CH3 | 4 | 0x0 | Статус прерывания захвата 3 канала таймера | R |
| IC_Int_CH2 | 3 | 0x0 | Статус прерывания захвата 2 канала таймера | R |
| IC_Int_CH1 | 2 | 0x0 | Статус прерывания захвата 1 канала таймера | R |
| UDF_Int | 1 | 0x0 | Статус прерывания опустошения счетчика | R |
| OVF_Int | 0 | 0x0 | Статус прерывания по реполнению счетчика | R |

Таблица 51 – Регистр INT_MSK модуля таймера

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| – | 31:10 | – | Зарезервировано | – |
| OC_Int_CH4 | 9 | 0x0 | Маска прерывания совпадения 4 канала таймера | RW |
| OC_Int_CH3 | 8 | 0x0 | Маска прерывания совпадения 3 канала таймера | RW |
| OC_Int_CH2 | 7 | 0x0 | Маска прерывания совпадения 2 канала таймера | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| OC_Int_CH1 | 6 | 0x0 | Маска прерывания совпадения 1 канала таймера | RW |
| IC_Int_CH4 | 5 | 0x0 | Маска прерывания захвата 4 канала таймера | RW |
| IC_Int_CH3 | 4 | 0x0 | Маска прерывания захвата 3 канала таймера | RW |
| IC_Int_CH2 | 3 | 0x0 | Маска прерывания захвата 2 канала таймера | RW |
| IC_Int_CH1 | 2 | 0x0 | Маска прерывания захвата 1 канала таймера | RW |
| UDF_Int | 1 | 0x0 | Маска прерывания опустошения счетчика | RW |
| OVF_Int | 0 | 0x0 | Маска прерывания по реполнению счетчика | RW |

3.9.4 Особенности работы с таймером.

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

- [2:0] – Таймер32_0;
- [5:3] – Таймер32_1;
- [8:6] – Таймер32_2;

Значения полей дешифрируются следующим образом:

- 0x0 – системная частота (sys_clk);
- 0x1 – частота шины АНВ (hclk);
- 0x2 – частота внешнего осцилятора OSC32M;
- 0x3 – частота встроенного осцилятора HSI32M;
- 0x4 – частота внешнего осцилятора OSC32K;
- 0x5 – частота встроенного осцилятора 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 TIM_PRESCALE 0;
- Записать в регистр INT_CLEAR 0xFFFFFFFF;
- Установить TIM_EN в 0.

3.10 Модуль 16-разрядного таймера

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 вперед;
- битный делитель 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;

- Первая конфигурация - когда TIMER16 тактируется внешним сигналом, но в то же время внутренний тактовый сигнал подается на TIMER16 либо от АНВ, 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 АНВ (hclk);
- 0x2 – частота внешнего осциллятора OSC32M;
- 0x3 – частота встроенного осциллятора HSI32M;
- 0x4 – частота внешнего осциллятора OSC32K;
- 0x5 – частота встроенного осциллятора LSI32K.

3.10.2 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 SKFLT
- 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.

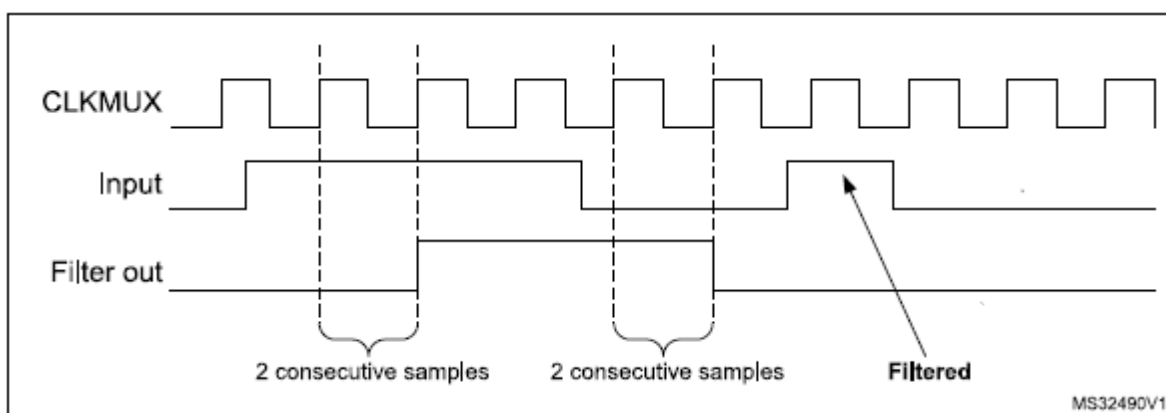


Рисунок 47 – Временная диаграмма glitch-фильтра

На рисунке 47 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.

3.10.3 Предделитель

16-битному счетчику TIMER16 предшествует конфигурируемый предделитель степени 2. Коэффициент деления предделителя управляется 3-битным полем PRESC[2:0]. В таблице 50 перечислены все возможные коэффициенты деления:

Таблица 52 – Коэффициенты деления прескалера

| Значие поля PRESC | Коэффициент деления |
|-------------------|---------------------|
| 000 | 1 |
| 001 | 2 |
| 010 | 4 |
| 011 | 8 |
| 100 | 16 |
| 101 | 32 |
| 110 | 64 |
| 111 | 128 |

3.10.4 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 '00', 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 '00', 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 проигнорированна.

3.10.5 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.

3.10.5.1 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 48.

В случае программного запуска (TRIGEN[1:0] = '00'), установка SNGSTRT запустит счетчик для однократного счета.

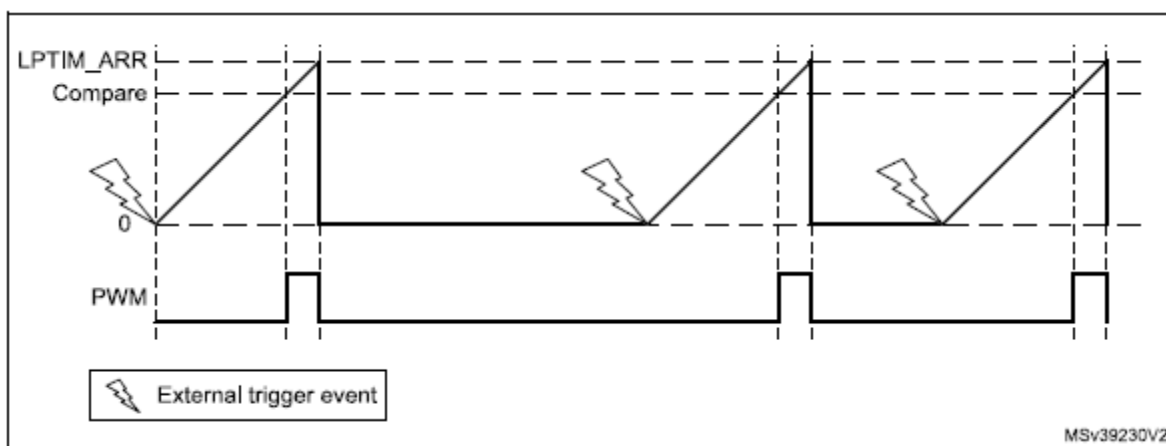


Рисунок 48 – Форма выходного сигнала TIMER16, конфигурация режима одиночного счета и активированный режим однократный (бит WAVE установлен)

3.10.5.2 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 38.

В случае программного запуска ($TRIGEN[1:0] = '00'$), установка CNTSTRT запустит счетчик для непрерывного счета.

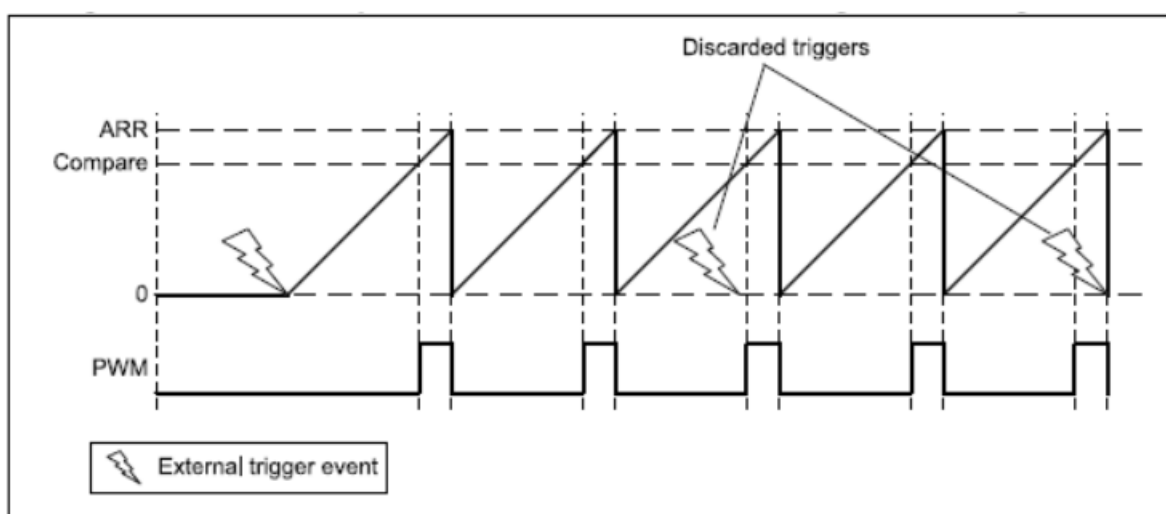


Рисунок 49 – Форма выходного сигнала TIMER16, конфигурация режима непрерывного счета

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.

3.10.6 Функция тайм-аута

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

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.

3.10.7 Генерация волновой формы

Два 16-битных регистра, TIMER16_ARR (регистр автозагрузки) и TIMER16_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 TIMER16_CMP и TIMER16_CNT. Выход TIMER16 сбрасывается, как только возникает соответствие между регистрами TIMER16_ARR и TIMER16_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).

Описанные выше режимы требуют, чтобы значение регистра `TIMER16_ARR` было строго больше значения регистра `TIMER16_CMP`.

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 39 показаны три возможные формы сигналов, которые могут быть сгенерированы на выходе `TIMER16`. Также здесь показан эффект изменения полярности с помощью бита `WAVPOL`.

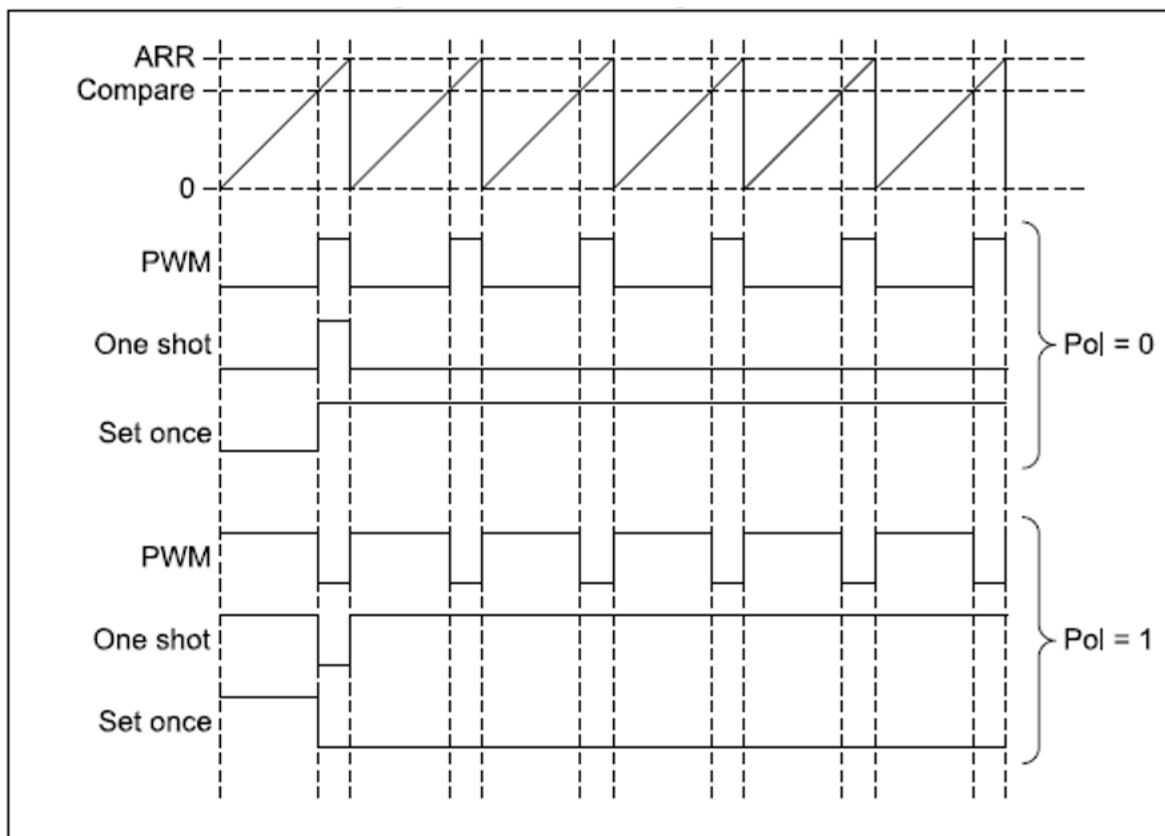


Рисунок 50 – Генерация формы волны

3.10.8 Обновление регистра

Регистр `TIMER16_ARR` и регистр `TIMER16_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 `TIMER16_ARR` и `TIMER16_CMP`:

- когда бит `PRELOAD` сброшен в '0', регистры `TIMER16_ARR` и `TIMER16_CMP` немедленно обновляются после любого доступа на запись.
- когда бит `PRELOAD` установлен в '1', регистры `TIMER16_ARR` и `TIMER16_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 TIMER16_ISR указывают, когда операция записи завершена соответственно в регистр TIMER16_ARR и регистр TIMER16_CMP.

После записи в регистр TIMER16_ARR или TIMER16_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.

3.10.9 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.

3.10.10 Включение таймера

Бит ENABLE, расположенный в регистре TIMER16_CR, используется для включения/выключения логики ядра TIMER16. После установки бита ENABLE необходима задержка в два такта, прежде чем TIMER16 будет фактически включен.

Регистры TIMER16_CFGR и TIMER16_IER должны быть изменены только тогда, когда TIMER16 отключен.

3.10.11 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 TIMER16_ARR (0 до ARR или ARR до 0 в зависимости от направления). Поэтому перед началом работы необходимо сконфигурировать TIMER16_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 TIMER16_ISR. Кроме того, для обоих событий изменения направления может быть сгенерировано прерывание, если оно разрешено через регистр TIMER16_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 53 приведены возможные комбинации в предположении, что Input1 и Input2 не переключаются одновременно.

Т а б л и ц а 53 – 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.

Внимание: В этом режиме 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 '000').

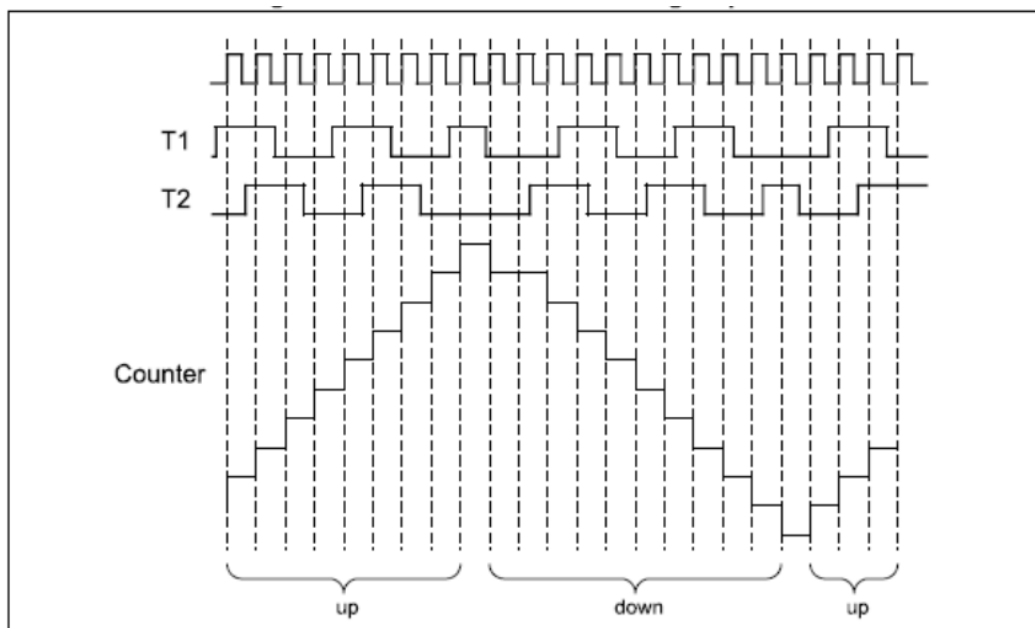


Рисунок 51 – Последовательность счета в режиме энкодера

3.10.12 Прерывания TIMER16

Следующие события генерируют событие прерывание/пробуждение, если они разрешены с помощью функции регистра TIMER16_IER:

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

Примечание – Если какой-либо бит в регистре TIMER16_IER (регистр разрешения прерывания) установлен после того, как установлен соответствующий флаг в регистре TIMER16_ISR (регистр состояния), прерывание не утверждается.

Т а б л и ц а 54 – События прерывания

| Событие прерывания | Описание |
|----------------------------------|---|
| Совпадение значения | Флаг прерывания поднимается, когда содержимое регистра счетчика (TIMER16_CNT) совпадает с содержимым регистра сравнения (TIMER16_CMP). |
| Автоперезагрузка значения | Флаг прерывания поднимается, когда содержимое регистра счетчика (TIMER16_CNT) совпадает с содержимым регистра автоперезагрузки (TIMER16_ARR). |
| Внешнее пусковое событие | Флаг прерывания поднимается при обнаружении внешнего триггерного события |
| Обновление регистра сравнения ОК | Флаг прерывания поднимается, когда операция записи в регистр TIMER16_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. |

3.10.13 Регистры TIMER16

Регистры 16-разрядного таймера представлены в таблицах 55-59.

Т а б л и ц а 55 – Регистры 16-разрядного таймера

| Адрес | Назначение | Обозначение |
|-------|-------------------------------------|--------------|
| 0x00 | Регистр флагов прерываний | TIMER16_ISR |
| 0x04 | Регистр сброса флагов прерываний | TIMER16_ICR |
| 0x08 | Регистр разрешения прерываний | TIMER16_IER |
| 0x0C | Регистр конфигурации | TIMER16_CFGR |
| 0x10 | Регистр управления | TIMER16_CR |
| 0x14 | Регистр сравнения | TIMER16_CMP |
| 0x18 | Регистр автоматической перезагрузки | TIMER16_ARR |
| 0x1C | Регистр счетчика | TIMER16_CNT |

Таблица 56 – Регистр TIMER16_ISR

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| – | 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 ОК. ARROK устанавливается аппаратными средствами, чтобы сообщить приложению, что операция записи в регистр TIMER16_ARR шины APB успешно завершена | RO |
| CMPOK | 3 | 0x0 | Обновление регистра сравнения ОК. CMPOK устанавливается аппаратными средствами, чтобы сообщить приложению, что операция записи в регистр TIMER16_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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| ARRM | 1 | 0x0 | Соответствие автозагрузки. ARRM устанавливается аппаратурой, чтобы сообщить приложению, что значение регистра TIMER16_CNT достигло значения регистра TIMER16_ARR. | RO |
| CMPM | 0 | 0x0 | Совпадение сравнения. Бит СМРМ устанавливается аппаратными средствами, чтобы сообщить приложению, что значение регистра TIMER16_CNT достигло значения регистра TIMER16_CMP | RO |

Таблица 57 – Регистр TIMER16_ICR

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| – | 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 |
| ARRMCF | 1 | 0x0 | Запись «1» в этот бит снимает флаг ARRM в регистре LPT_ISR | W1C |
| CMPMCF | 0 | 0x0 | Запись «1» в этот бит снимает флаг СМРМ в регистре LPT_ISR | W1C |

Таблица 58 – Регистр TIMER16_IER

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| – | 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| CMPOKIE | 3 | 0x0 | «1» - прерывание СМПОК разрешены «0» - прерывание СМПОК 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 СМРМ разрешены «0» - прерывание СМРМ запрещены | RW |

Таблица 59 – Регистр TIMER16_CFGR

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| – | 31:25 | | Зарезервировано | |
| ENC | 24 | 0x0 | Разрешение режима энкодера. Бит ENC управляет режимом работы энкодера «0» - Режим энкодера отключен «1» - Режим энкодера включен | RW |
| 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 TIMER16_ARR и TIMER16_CMP «0» - Регистры обновляются после каждого доступа к записи на шине APB «1» - Регистры обновляются в конце текущего периода TIMER16 | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| WAVEPOL | 21 | 0x0 | Полярность формы волны. Бит WAVEPOL управляет полярностью выходного сигнала «0» - Выход TIMER16 отражает результаты сравнения между регистрами TIMER16_ARR и TIMER16_CMP «1» - Выход TIMER16 отражает инверсные результаты сравнения между регистрами TIMER16_ARR и TIMER16_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 бита OPMODE) «1» - Активировать режим Set-once | RW |
| TIMEOUT | 19 | 0x0 | Разрешение тайм-аута. Бит TIMOUT управляет функцией тайм-аута «0» - триггерное событие, поступающее, когда таймер уже запущен, будет проигнорировано «1» - Триггерное событие, поступающее, когда таймер уже запущен, сбросит и перезапустит счетчик. | RW |
| TRIGEN | 18:17 | 0x0 | 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: «00» - программный триггер (начало отсчета иницируется программно) «01» - нарастающий фронт является активным фронтом «10» - падающий фронт является активным фронтом «11» - оба фронта являются активными фронтами энергопотреблением (TIMER16) | RW |
| – | 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> <p>«000» - timer16_0 - GPIO0[7] timer16_1 - GPIO1[9] timer16_2 - GPIO2[3]</p> <p>«001» - timer16_0 - GPIO0[4] timer16_1 - GPIO1[8] timer16_2 - GPIO2[2]</p> <p>«010» - timer16_0 - GPIO0[15] timer16_1 - GPIO1[7] timer16_2 - GPIO2[1]</p> <p>«011» - timer16_0 - GPIO0[14] timer16_1 - GPIO1[6] timer16_2 - GPIO2[0]</p> <p>«100» - окончание преобразования термосенсора «101» - окончание преобразования АЦП «110» - Прерывание RTC «111» - Будильник</p> | RW |
| – | 12 | Зарезервировано | | |
| PRESC | 11:9 | 0x0 | <p>Тактовый прескалер. Биты PRESC задают коэффициент деления прескалера. Он может быть одним из следующих коэффициентов деления:</p> <p>000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128</p> | RW |
| – | 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>«00» - любое изменение активного уровня триггера считается действительным триггером</p> <p>«01» - изменение активного уровня триггера должно быть стабильным в течение не менее 2 тактовых периодов, прежде чем он будет считаться действительным триггером.</p> <p>«10» - изменение активного уровня триггера должно быть стабильным в течение как минимум 4 тактовых периодов, прежде чем он будет считаться действительным триггером.</p> <p>«11» - изменение активного уровня триггера должно быть стабильным в течение не менее 8 тактовых периодов, прежде чем он будет считаться действительным триггером.</p> | RW |
| — | 5 | Зарезервировано | | |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| СКFLT | 4:3 | 0x0 | <p>Конфигурируемый цифровой фильтр для внешнего тактового генератора. Значение СКFLT 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>«00» - любое изменение уровня внешнего тактового сигнала рассматривается как допустимый переход</p> <p>«01» - изменение уровня внешнего тактового сигнала должно быть стабильным в течение как минимум 2 тактовых периодов, прежде чем оно будет считаться допустимым переходом.</p> <p>«10» - изменение уровня внешнего тактового сигнала должно быть стабильным в течение как минимум 4 тактовых периодов, прежде чем оно будет считаться действительным переходом.</p> <p>«11» - 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 |
|---|---------|--------------------|---|--------|
| СКPOL | 2:1 | 0x0 | <p>Полярность синхронизации. Когда TIMER16 тактируется внешним источником тактового сигнала, биты СКPOL используются для настройки активного фронта или фронтов, используемых счетчиком:</p> <p>«00» - нарастающий фронт является активным фронтом, используемым для подсчета</p> <p>«01» - спадающий фронт является активным фронтом, используемым для подсчета</p> <p>«10» - 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>«11» - не разрешается</p> <p>Если TIMER16 сконфигурирован в режиме энкодера (установлен бит ENC):</p> <p>«00» - активен подрежим энкодера 1</p> <p>«01» - активен подрежим энкодера 2</p> <p>«10» - активен подрежим энкодера 3</p> | RW |
| СКSEL | 0 | 0x0 | <p>Селектор тактовых импульсов. Бит СКSEL 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 |
| <p>Примечание: Регистр TIMER16_CFGR может быть изменен только тогда, когда TIMER16 отключен (бит ENABLE сброшен в '0').</p> | | | | |

Таблица 60 – Регистр TIMER16_CR

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| – | 31:3 | Зарезервировано | | |
| CNTSTRT | 2 | 0x0 | <p>Запуск таймера в непрерывном режиме Этот бит устанавливается программно и очищается аппаратно.</p> <p>В случае программного запуска (TRIGEN[1:0] = '00'), установка этого бита запускает TIMER16 в непрерывном режиме. Если программный запуск отключен (TRIGEN[1:0] отличен от '00'), 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 TIMER16_ARR и TIMER16_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 |
| SNGSTRT | 1 | 0x0 | <p>Запуск TIMER16 в одиночном режиме Этот бит устанавливается программно и очищается аппаратно.</p> <p>В случае программного запуска (TRIGEN[1:0] = '00'), установка этого бита запускает TIMER16 в режиме одиночного импульса. Если программный запуск отключен (TRIGEN[1:0] отличен от '00'), установка этого бита запускает TIMER16 в режиме одиночного импульса, как только обнаруживается внешний триггер. Если этот бит установлен, когда TIMER16 находится в режиме непрерывного счета, то TIMER16 остановится при следующем совпадении регистров TIMER16_ARR и TIMER16_CNT.</p> <p>Этот бит может быть установлен только при включенном TIMER16. Он будет автоматически сброшен аппаратными средствами.</p> | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| ENABLE | 0 | 0x0 | Разрешение TIMER16. Бит ENABLE устанавливается и очищается программно. | RW |

Таблица 61 – Регистр TIMER16_CMP

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

Примечание: Регистр TIMER16_CMP может быть изменен только тогда, когда TIMER16 включен (бит ENABLE установлен в '1').

Таблица 62 – Регистр TIMER16_ARR

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| – | 31:16 | Зарезервировано | | |
| ARR | 15:0 | 0x0 | Значение автоматической перезагрузки ARR - значение автозагрузки для TIMER16. Это значение должно быть строго больше, чем значение CMP[15:0]. | RW |

Примечание: Регистр TIMER16_ARR может быть изменен только тогда, когда TIMER16 включен (бит ENABLE установлен в '1').

Таблица 63 – Регистр TIMER16_CNT

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| – | 31:16 | Зарезервировано | | |
| CNT | 15:0 | 0x0 | Значение счетчика. Когда TIMER16 работает с асинхронными тактовыми сигналами, чтение регистра TIMER16_CNT может вернуть недостоверные значения. Поэтому в этом случае необходимо выполнить два последовательных доступа на чтение и убедиться, что два возвращенных значения идентичны. | RO |

3.11 Часы реального времени

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.

3.11.1 Описание модуля

Модуль предназначен для учёта времени и даты. Внутренняя структура модуля представлена на рисунке 52.

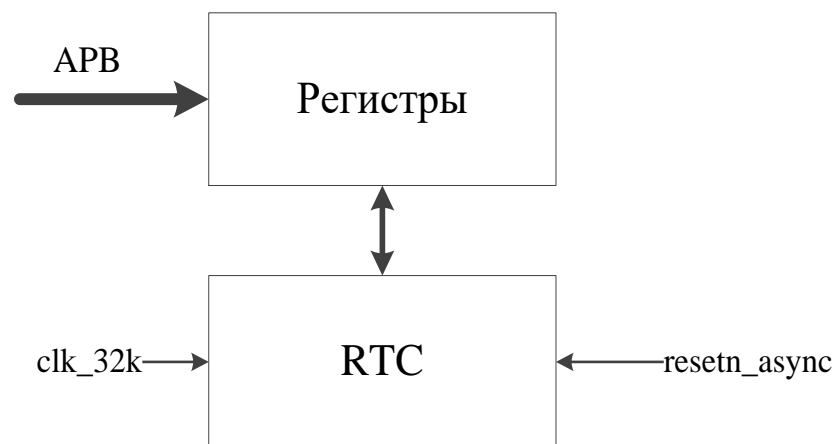


Рисунок 52 – Внутренняя структура модуля

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

3.11.2 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.

3.11.3 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.

3.11.4 Регистры модуля RTC

В таблице 64 указаны смещения регистров, доступных по интерфейсу конфигурации.

Таблица 64 – Регистры модуля RTC

| Обозначение | Смещение | Назначение | Доступ |
|-------------|----------|---------------------------------|--------|
| RRTC_TIME | 0x00 | Регистр времени | RW |
| RRTC_DATE | 0x04 | Регистр даты | RW |
| RRTC_TALRM | 0x08 | Регистр времени будильника | RW |
| RRTC_DALRM | 0x0C | Регистр даты будильника | RW |
| RRTC_CTRL | 0x10 | Регистр управления | RW |
| RRTC_REG0 | 0x20 | Регистры общего назначения REG0 | RW |
| RRTC_REG1 | 0x24 | Регистры общего назначения REG1 | RW |
| RRTC_REG2 | 0x28 | Регистры общего назначения REG2 | RW |
| RRTC_REG3 | 0x2C | Регистры общего назначения REG3 | RW |
| RRTC_REG4 | 0x30 | Регистры общего назначения REG4 | RW |

| | | | |
|------------|------|----------------------------------|----|
| RRTC_REG5 | 0x34 | Регистры общего назначения REG5 | RW |
| RRTC_REG6 | 0x38 | Регистры общего назначения REG6 | RW |
| RRTC_REG7 | 0x3C | Регистры общего назначения REG7 | RW |
| RRTC_REG8 | 0x40 | Регистры общего назначения REG8 | RW |
| RRTC_REG9 | 0x44 | Регистры общего назначения REG9 | RW |
| RRTC_REG10 | 0x48 | Регистры общего назначения REG10 | RW |
| RRTC_REG11 | 0x4C | Регистры общего назначения REG11 | RW |
| RRTC_REG12 | 0x50 | Регистры общего назначения REG12 | RW |
| RRTC_REG13 | 0x54 | Регистры общего назначения REG13 | RW |
| RRTC_REG14 | 0x58 | Регистры общего назначения REG14 | RW |
| RRTC_REG15 | 0x5C | Регистры общего назначения REG15 | RW |

3.11.4.1 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 65.

Т а б л и ц а 65 – Поля регистра RRTC TIME

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| S | 4 – 7 | 0 | Поле единиц секунд. Допустимые значения от 0 до 9 | RW |
| TS | 8 – 10 | 0 | Поле десятков секунд. Допустимые значения от 0 до 5 | RW |
| M | 11 – 14 | 0 | Поле единиц минут. Допустимые значения от 0 до 9 | RW |
| TM | 15 – 17 | 0 | Поле десятков минут. Допустимые значения от 0 до 5 | RW |
| H | 18 – 21 | 0 | Поле единиц часов. Допустимые значения: – TH = 0 – От 0 до 9; – TH = 2 – От 0 до 3 | RW |
| TH | 22, 23 | 0 | Поле десятков часов. Доступные значения от 0 до 2 | RW |
| DOW | 24 – 26 | 0 | День недели. Допустимые значения в двоичном виде: | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|-------------------|--------------------|---|--------|
| | | | <ul style="list-style-type: none"> – «001» – понедельник; – «010» – вторник; – «011» – среда; – «100» – четверг; – «101» – пятница; – «110» – суббота; – «111» – воскресенье | |
| Резерв | 0 – 3, 27 – 31 | – | – | – |

3.11.4.2 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.

Т а б л и ц а 66 – Поля регистра RRTC DATE

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

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|------------|--------|
| Резерв | 27 – 31 | – | – | – |

3.11.4.3 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 67.

Таблица 67 – Поля регистра RRTC TALRM

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| – | 0 – 3 | – | Резерв | – |
| S | 4 – 7 | 0 | Поле единиц секунд. Допустимые значения от 0 до 9 | RW |
| TS | 8 – 10 | | Поле десятков секунд. Допустимые значения от 0 до 5 | |
| M | 11 – 14 | | Поле единиц минут. Допустимые значения от 0 до 9 | |
| TM | 15 – 17 | | Поле десятков минут. Допустимые значения от 0 до 5 | |
| H | 18 – 21 | | Поле единиц часов. Допустимые значения: – TH = 0 – от 0 до 9; – TH = 2 – от 0 до 3 | |
| TH | 22, 23 | | Поле десятков часов. Доступные значения от 0 до 2 | |
| DOW | 24 – 26 | | 0 | |
| – | 27 | – | Резерв | – |
| CS | 28 | 0 | Разрешает сравнения секунд, когда установлен | RW |
| CM | 29 | | Разрешает сравнения минут, когда установлен | |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| CH | 30 | | Разрешает сравнения часов, когда установлен | |
| CDOW | 31 | | Разрешает сравнения дней недели, когда установлен | |

3.11.4.4 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 68 – Поля регистра RRTC DALRM

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

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| СУ | 29 | | Разрешает сравнения годов, когда установлен | |
| СС | 30 | | Разрешает сравнения веков, когда установлен | |
| Резерв | 31 | – | – | – |

3.11.4.5 Регистр управления

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

Таблица 69 – Поля регистра RRTC_CTRL

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| 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 |
| INTE | 29 | 0 | Бит разрешения прерывания на выходах irq и irq_async. Прерывания появляются при наличии установленного бита ALRM | RW |
| ALRM | 30 | | Установленный бит свидетельствует о совпадении одного разрешенного или всех разрешённых полей будильника. Бит необходимо сбрасывать при помощи управляющего ПО | |
| EN | 31 | | Модуль включён и производит отсчёт времени, когда установлен | |
| Резерв | 0 – 28 | – | – | – |

3.11.4.6 Регистры общего назначения RRTC_REG0 – RRTC_REG15.

В адресном пространстве модуля доступно 16 32х-разрядных регистра. Они могут использоваться в произвольных целях. Начальные значения регистров при включении питания 0x00000000. Регистры располагаются в батарейном домене и

сохраняют свое значение при отключении системного питания и при внешнем сбросе.

3.12 Программируемый контроллер прерываний

Модуль реализует поддержку 32 линий прерываний от внешних устройств. Модуль обеспечивает формирование прерываний как по уровню, так и при детектировании фронта(спада) на линии прерывания.

Перечень линий прерывания для устройств, подключённых к контроллеру прерываний, указан в таблице 70.

Таблица 70 – 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 (переферирийные 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 71 представлены адреса, обозначения и наименования регистров контроллера прерывания, доступных по интерфейсу APB. Поля регистров контроллера прерывания показаны в таблицах 72 – 77. При чтении полей, доступных только для записи, возвращается в «0». Все зарезервированные поля регистров физически не представлены в модуле.

Таблица 71 – Регистры контроллера прерываний

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|------------------|----------|--------------------|--------------------------------------|--------|
| MASK_LEVEL_SET | 00 | 0 | Установка маски прерываний по уровню | RW |
| MASK_LEVEL_CLEAR | 04 | 0 | Сброс маски прерываний по уровню | RW |
| MASK_EDGE_SET | 08 | 0 | Установка маски прерываний по фронту | RW |
| MASK_EDGE_CLEAR | 0C | 0 | Сброс маски прерываний по фронту | RW |
| STATUS | 18 | 0 | Статус прерываний | RO |
| RAW_STATUS | 1C | 0 | Текущее состояние линий прерываний | W1C |

Таблица 72 – Поля регистра MASK_LEVEL_SET

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|----------------|---------|--------------------|--|--------|
| MASK_LEVEL_SET | 31:0 | 0 | При чтении – текущее состоянии масок прерываний по уровню. При записи, «1» разрешает прерывание по уровню соответствующего источника прерываний. | RW |

Таблица 73 – Поля регистра MASK_LEVEL_CLEAR

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|------------------|---------|--------------------|--|--------|
| MASK_LEVEL_CLEAR | 31:0 | 0 | При чтении – текущее состоянии масок прерываний по уровню. При записи, «1» запрещает прерывание по уровню соответствующего источника прерываний. | RW |

Таблица 74 – Поля регистра MASK_EDGE_SET

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

Таблица 75 – Поля регистра MASK_EDGE_CLEAR

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

Таблица 76 – Поля регистра STATUS

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| STATUS | 31:0 | 0 | Текущее состояние регистра прерываний с учетом масок | RO |

| | | | | |
|--|--|--|-----------------|--|
| | | | уровня и фронта | |
|--|--|--|-----------------|--|

Таблица 77 – Поля регистра RAW_STATUS

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

3.13 Блок вычисления контрольных сумм

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

Структурная схема блока представлена на рисунке 53.

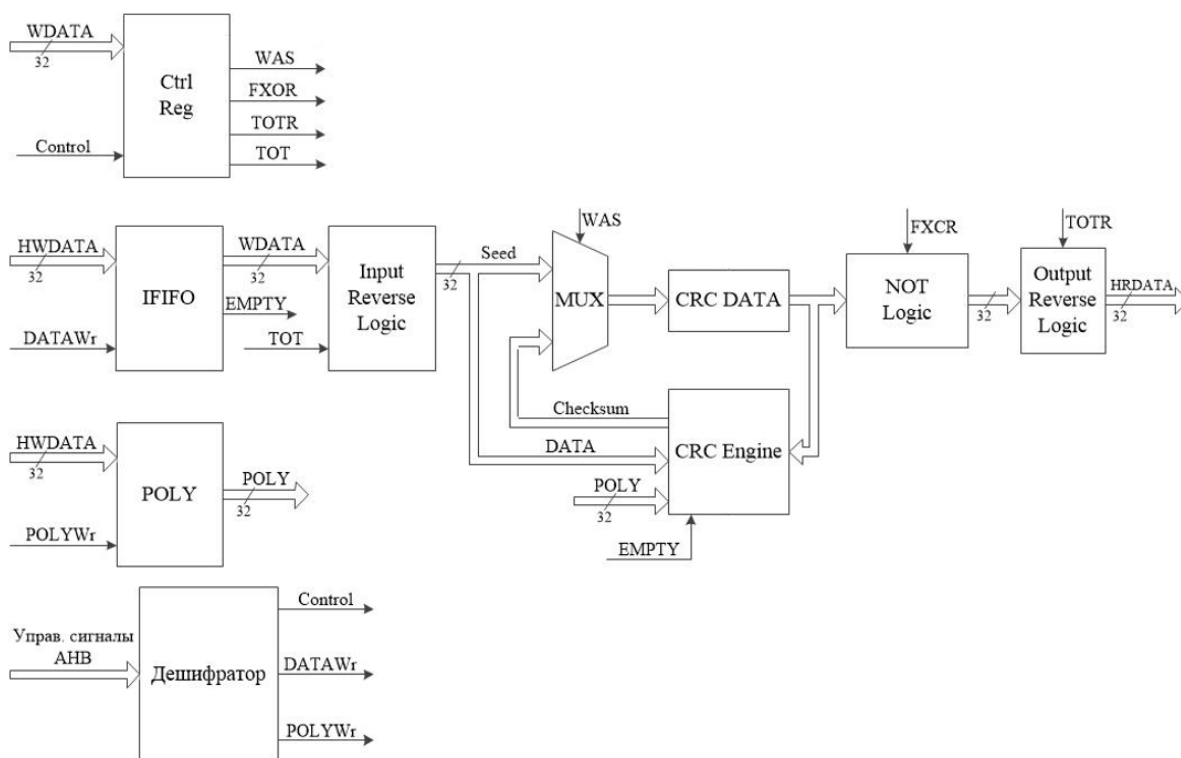


Рисунок 53 – Схема блока CRC

Контроллер состоит из следующих компонентов:

- дешифратора системных сигналов шины АНВ,
- регистра управления CtrlReg,
- регистра полинома POLY Reg,
- 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 CtrlReg управляет всеми настройками процесса вычисления суммы. Регистр полинома POLY Reg 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 управляется 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 управляется битами поля TOTR регистра управления. Выходные биты контрольной суммы подключены напрямую к шине HRDATA.

Карта регистров блока приведена в таблице 78. Запись данных в регистр данных может осуществляться в 8, 16 или 32-разрядном режиме. Доступ к остальным регистрам осуществляется только в 32-разрядном режиме.

Т а б л и ц а 78 – 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 79.

Таблица 79 – Описание полей регистра управления блока CRC

| Обозначение | Значение после сброса | Назначение | Доступ |
|-----------------|-----------------------|---|--------|
| TOT [31:30] | 00 | Перестановка битов/байтов входных данных: 00 – перестановка выключена; 01 – биты в байтах перестанавливаются (бит 7 с битом 0, бит 15 с битом 8, с остальными байтами так же), байты НЕ перестанавливаются; 10 – перестанавливаются и биты, и байты 11 – биты в байтах НЕ перестанавливаются, байты перестанавливаются (меняются местами байты 3 и 0, 2 и 1); | RW |
| TOTR [29:28] | 00 | Перестановки битов/байтов выходных данных: 00 – перестановка выключена; 01 – 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; 10 – перестанавливаются и биты, и байты 11 – биты в байтах НЕ перестанавливаются, байты перестанавливаются (меняются местами байты 3 и 0, 2 и 1); | RW |
| [27] | | Зарезервировано | – |
| FXOR [26] | 0 | 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 | Значение после сброса | Назначение | Доступ |
|-------------|-----------------------|---|--------|
| WAS [25] | 0 | Назначение регистра данных: 0 – записываем данные; 1 – записываем начальное значение | RW |
| [24:1] | | Зарезервировано | – |
| Busy | 0 | 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 | R |

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 TOT, 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.

3.14 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 ведущих 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 сигнал выбора ведомого SS.

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.

3.14.1 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 (TxFIFO);
- система тактирования;
- схема управления;
- блок ведущего;
- блок ведомого.

Структурная схема SPI контроллера показана на рисунке 54.

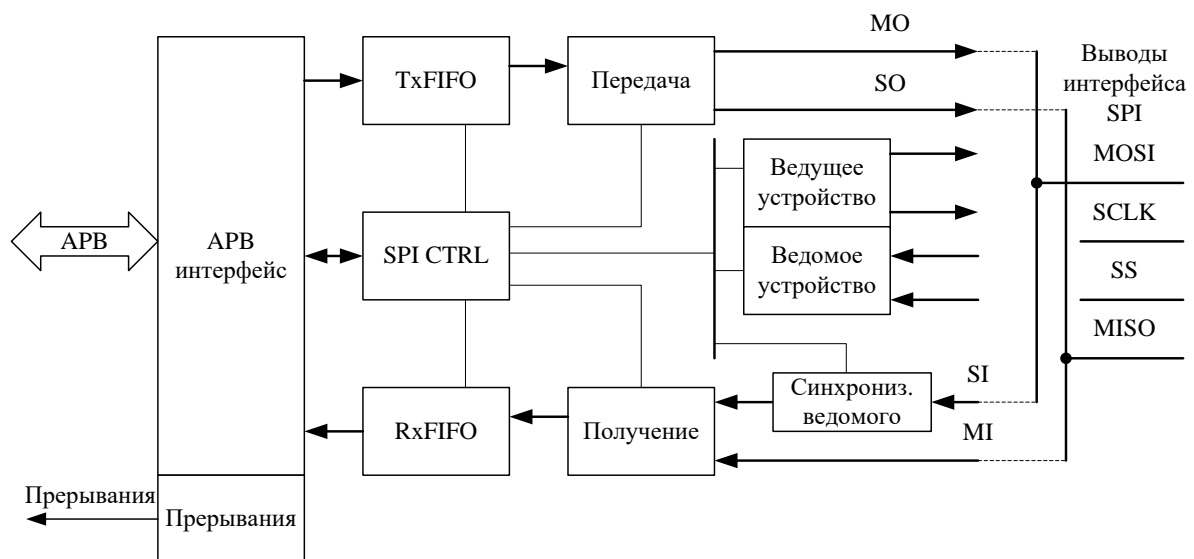


Рисунок 54 – Функциональная схема SPI контроллера

3.14.1.1 APB интерфейс

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

3.14.1.2 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 TxFIFO 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.

3.14.1.3 Регистр принимаемых данных приема (RxFIFO)

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

3.14.1.4 Регистр передаваемых данных (TxFIFO)

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 буфер приема не будет считан до конца. Если регистр передаваемых данных переполнен, бит [RX_OVERFLOW] устанавливается в 1.

3.14.1.5 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.

3.14.2 Функциональное описание

3.14.2.1 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 устройств, 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.

Поддержка режима нескольких ведущих устройств

В режиме поддержки нескольких ведущих устройств контроллер конфигурируется в режим ведущего устройства, установкой бита [MODE_SEL], и может инициировать обмен данными с любым выбранным ведомым устройством. Когда про-

грамма готова начать обмен данными, разрешается работа контроллера установкой бита [SPI_EN]. Когда обмен завершен, программа отключает контроллер. Контроллер не может быть выбран внешним ведомым устройством, когда он находится в режиме ведущего.

Контроллер обнаруживает другое ведущее устройство на шине путем контроля состояния сигнала выбор ведомого, который имеет тип открытый исток и имеет низкий активный уровень. Когда контроллер обнаруживает другое ведущее устройство, он устанавливает бит статуса прерывания `Intr_status_reg0 [MODE_FAIL]` и очищает управляющий бит `En_reg0 [SPI_EN]`. Программа может обрабатывать прерывание [MODE_FAIL] следующим образом: прервать передачу, сбросить контроллер и начать передачу заново.

3.14.2.2 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_reg0 [MODE_SEL] = 0` и `En_reg0 [SPI_EN] = 1`.

По сигналу SCLK происходит защелкивание данных на входе MOSI. Контроллер игнорирует данные на входе MOSI, если на вывод выбор ведомого устройства подан высокий логический уровень (неактивное состояние). Сигнал выбор ведомого устройства должен удерживаться в активном состоянии на время передачи. Контроллер устанавливает бит прерывания `Intr_status_reg0 [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.

3.14.3 Фаза и полярность

Существует 4 комбинации для задания фазы и полярности управляющих бит [CLK_PH] и [CLK_POL] в регистре Config_reg0. 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 80 приведены значения управляющих сигналов бит [CLK_PH] и [CLK_POL].

Таблица 80 – 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 | |

В режиме ведущего устройства, значение бита фазы тактового сигнала [CLK_PH] в регистре Config_reg0 [CLK_PH] влияет на протокол обмена, посредством регистра Delay_reg0 следующим образом:

- CLK_PH = 0

SS активный: Ведущее устройство автоматически переводит выходы SS в неактивное состояние (высокое состояние) на время, заданное битами Delay_reg0 [d_nss], $Time = (1 + [d_nss]) * SPI_Ref_Clk$. Минимальное время два периода сигнала SPI_Ref_Clk.

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 последова-

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

– CLK_PN = 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_reg0). Это позволяет выгрузить TxFIFO и подготовить к следующему преобразованию из параллельного кода в последовательный.

3.14.4 Прерывания FIFO

Прерывания буферов RxFIFO и TxFIFO показаны на рисунке 55.

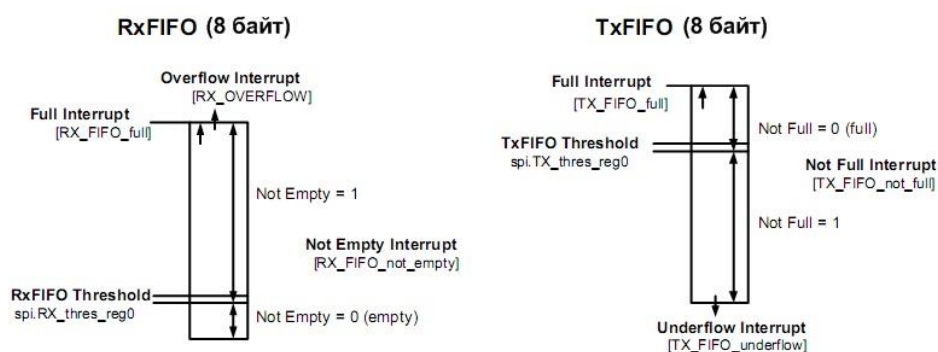


Рисунок 55 – Прерывания регистром RxFIFO и TxFIFO

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 56.

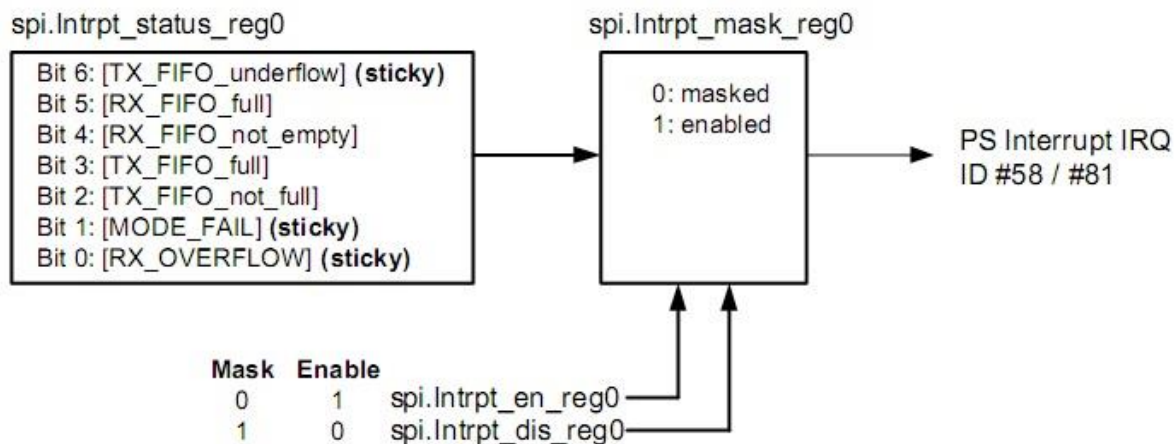


Рисунок 56 – Регистры прерывания

3.14.5 Регистры модуля контроллера SPI

Описание регистров представлено в таблицах 81 – 93.

Таблица 81 – Регистры модуля контроллера SPI

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-----------------------|----------|--------------------|---|--------|
| Config_reg0 | 0x00 | 0x00000000 | Регистр конфигурации SPI | * |
| Intr_status_reg0 | 0x04 | 0x00000004 | Регистр статуса прерываний SPI | * |
| Intrpt_en_reg0 | 0x08 | 0x00000000 | Регистр разрешения прерываний | WO |
| Intrpt_dis_reg0 | 0x0C | 0x00000000 | Регистр запрета прерываний | WO |
| Intrpt_mask_reg0 | 0x10 | 0x00000000 | Регистр текущих масок прерываний | RO |
| En_reg0 | 0x14 | 0x00000000 | Регистр включения/выключения SPI | * |
| Delay_reg0 | 0x18 | 0x00000000 | Регистр задержек | RW |
| Tx_data_reg0 | 0x1C | 0x00000000 | Регистр передаваемых данных | WO |
| Rx_data_reg0 | 0x20 | 0x00000000 | Регистр принимаемых данных | RO |
| Slave_Idle_count_reg0 | 0x24 | 0x000000FF | Регистр счетчика останова ведомого устройства | * |
| TX thres reg0 | 0x28 | 0x00000001 | Регистр пороговых значений TX_FIFO | RW |
| module_ID | 0x2C | 0x01090100 | Идентификационный номер модуля | RO |

* Тип доступа зависит от поля/разряда регистра.

Таблица 82 – Регистр Config_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| – | 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: «xxx0» – выбрано устройство 4; «xx01» – выбрано устройство 1; «x011» – выбрано устройство 2; «0111» – выбрано устройство 3; «1111» – ведомые устройства не выбраны | RW |
| PERI_SEL | 9 | 0x0 | Выбор периферийного деэнкодера: «1» – использование внешнего деэнкодера; «0» – выбор только 1 из 4 устройств | RW |
| REF_CLK | 8 | 0x0 | Выбор опорной тактовой частоты «1» – не поддерживается; «0» – используется опорная частота SPI | RW |
| DATA_SZ | 7:6 | 0x0 | Выбор длины передаваемой посылки: «00» – 8 бит; «01» – 16 бит; «10» – 24 бита; «11» – 32 бита | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------------|---------|--------------------|---|--------|
| BAUD_RATE_DIV | 5:3 | 0x0 | Управляет скоростью передачи данных, задает коэффициент деления частоты spi_ref_clk: «000» – не поддерживается; «001» – деление на 4; «010» – деление на 8; «011» – деление на 16; «100» – деление на 32; «101» – деление на 64; «110» – деление на 128; «111» – 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 83 – Регистр Intr status_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| – | 31:16 | Зарезервировано | | |
| SPI_active | 15 | 0x0 | Статус сеанса передачи: «1» – контроллер SPI в процессе обмена; «0» – контроллер SPI в состоянии ожидания обмена | RO |
| – | 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 | 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 значение THRESHOLD или больше | RO |
| MODE_FAIL (IXR_MODF) | 1 | 0x0 | Напряжение на выводе n _{ss_in} не соответствующему режиму работы SPI: «1» – если n _{ss_in} имеет низкий уровень в режиме ведомого устройства; «0» – корректная работа | RC |
| 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 устанавливаются в «1», если произошло событие вызывающее прерывание и прерывание разрешено с учётом маскирования. | | | | |

Таблица 84 – Регистр Intrpt_en_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|------------------------------|---------|--------------------|--|--------|
| – | 31:7 | Зарезервировано | | |
| TX_FIFO_underflow (IXR_TXUF) | 6 | 0x0 | Запись «1» устанавливает маску прерывания (разрешает прерывание) при опустошении TX_FIFO | WO |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------------------------------|---------|--------------------|--|--------|
| 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 85 – Регистр Intrpt_dis_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------------------------------|---------|--------------------|--|--------|
| – | 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|--------------------------------|---------|--------------------|---|--------|
| 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 86 – Регистр Intrpt_mask_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------------------------------|---------|--------------------|---|--------|
| – | 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 |
| 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|--------------------------------|---------|--------------------|--|--------|
| 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 87 – Регистр En_reg0 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 88 – Регистр Delay_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|------------------|---------|--------------------|---|--------|
| – | 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| 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 89 – Регистр Tx_data_reg0 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 90 – Регистр Rx_data_reg0 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 91 – Регистр Slave Idle count_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|--|--------|
| – | 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 92 – Регистр TX_thres_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|------------|--------|
|-------------|---------|--------------------|------------|--------|

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

Таблица 93 – Регистр Mod_id_reg0 модуля контроллера SPI

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--------------------------------|--------|
| module_ID | 31:0 | 0x01090100 | Идентификационный номер модуля | RO |

3.15 Модуль порта ввода/вывода общего назначения

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

В таблице 94 перечислены регистры модуля GPIO. Все регистры имеют разрядность, соответствующую количеству выводов блока.

Таблица 94 – 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 бит выводу устанавливается в «1» | RW |
| CLEAR | 0x04 | 0x0 | Текущее состояние прерываний / Установка «0» При чтении – текущее состояние запросов прерываний. При записи бита «1» соответствующий бит выводу устанавливается в «0» | WO |

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|---------------|----------|--------------------|---|--------|
| 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 |
| 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 |

3.15.1 Многофункциональные выходы

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 95 – 98.

Т а б л и ц а 95 – 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 |
| 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 |
| Смещение указано относительно базового адреса модуля 0x00051000 | | | | |

Т а б л и ц а 96 – Поля регистров PADx_CFG управления функцией выводов

| Номер вывода | Биты регистра PADx_CFG | Назначение |
|--------------|------------------------|---|
| Port_x_0 | [1:0] | Значения двух бит кодируют выбранный функционал для вывода 00 – первая функция (порт общего назначения) 01 – вторая функция (последовательный ин- |
| Port_x_1 | [3:2] | |
| Port_x_2 | [5:4] | |
| Port_x_3 | [7:6] | |
| Port_x_4 | [9:8] | |

| | | |
|-----------|---------|---|
| Port_x_5 | [11:10] | терфейс) 10 – третья функция (последовательный интерфейс или таймер) 11 – четвертая функция (аналоговый сигнал) |
| Port_x_6 | [13:12] | |
| Port_x_7 | [15:14] | |
| Port_x_8 | [17:16] | |
| Port_x_9 | [19:18] | |
| Port_x_10 | [21:20] | |
| Port_x_11 | [23:22] | |
| Port_x_12 | [25:24] | |
| Port_x_13 | [27:26] | |
| Port_x_14 | [29:28] | |
| Port_x_15 | [31:30] | |

Таблица 97 – Поля регистров PADx_DS управления нагрузочной способностью выводов

| Номер вывода | Биты регистра PADx_DS | Назначение |
|--------------|-----------------------|---|
| Port_x_0 | [1:0] | Значения двух бит кодируют выбранный режим для вывода 00 – 2мА, 01 – 4мА, 1x – 8мА |
| Port_x_1 | [3:2] | |
| Port_x_2 | [5:4] | |
| Port_x_3 | [7:6] | |
| Port_x_4 | [9:8] | |
| Port_x_5 | [11:10] | |
| Port_x_6 | [13:12] | |
| Port_x_7 | [15:14] | |
| Port_x_8 | [17:16] | |
| Port_x_9 | [19:18] | |
| Port_x_10 | [21:20] | |
| Port_x_11 | [23:22] | |
| Port_x_12 | [25:24] | |
| Port_x_13 | [27:26] | |
| Port_x_14 | [29:28] | |
| Port_x_15 | [31:30] | |

Т а б л и ц а 98 – Поля регистров PADx_PUPD управления подтягивающими резисторами

| Номер вывода | Биты регистра PADx_PUPD | Назначение |
|--------------|-------------------------|--|
| Port_x_0 | [1:0] | Значения двух бит кодируют выбранный режим для вывода 00 – регистры подтяжки не подключаются, 01 – подключен регистр подтяжки 50 кОм к питанию, 10 – подключен регистр подтяжки 50 кОм к земле, 11 – зарезервировано |
| Port_x_1 | [3:2] | |
| Port_x_2 | [5:4] | |
| Port_x_3 | [7:6] | |
| Port_x_4 | [9:8] | |
| Port_x_5 | [11:10] | |
| Port_x_6 | [13:12] | |
| Port_x_7 | [15:14] | |
| Port_x_8 | [17:16] | |
| Port_x_9 | [19:18] | |
| Port_x_10 | [21:20] | |
| Port_x_11 | [23:22] | |
| Port_x_12 | [25:24] | |
| Port_x_13 | [27:26] | |
| Port_x_14 | [29:28] | |
| Port_x_15 | [31:30] | |

3.15.2 Прерывания

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 IRQ_LINE_MUX. Схема коммутации представлена в таблице 99.

Т а б л и ц а 99 – 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 |

| 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]=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 |

Описание регистров блока прерываний представлено в таблице 100.

Т а б л и ц а 100 – Регистры модуля управления прерываниями GPIO

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|--------------|----------|--------------------|--|--------|
| STATE | 0x00 | 0x0 | Текущее состояние выводов линий после мультиплексоров | RO |
| IRQ_LINE_MUX | 0x04 | 0x0 | Управление мультиплексорами. Каждый мультиплексор управляется 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 | RW |

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------------|----------|--------------------|--|--------|
| INTR_ENABLE_SET | 0x0C | 0x0 | Регистр разрешения прерываний При чтении – текущее состояние разрешений прерываний. При записи «1» разрешает прерывание от соответствующего вывода | RW |
| INTR_ENABLE_CLEAR | 0x10 | 0x0 | Регистр запрета прерываний При чтении – текущее состояние разрешений прерываний. При записи «1» запрещает прерывание от соответствующего вывода | RW |
| INTR_EDGE | 0x14 | 0x0 | Регистр типа прерываний. При чтении – текущий тип прерывания: «1» – по событию (фронт или спад), «0» – по уровню. Запись «1» – прерывание формируется по событию для соответствующего вывода | RW |
| INTR_LEVEL | 0x18 | 0xFF | Регистр типа прерываний. При чтении – текущий тип прерывания (инвертированный): «0» – по событию; «1» – по уровню. Запись «1» – прерывание формируется по уровню для соответствующего вывода | RW |

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------------|----------|--------------------|--|--------|
| INTR_LEVEL_SET | 0x1C | 0x0 | Регистр выбора 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 нарастающему фронту или уровню логической «1» для соответствующего вывода | RW |
| INTR_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 вывода | RW |
| INTR_ANY_EDGE_SET | 0x24 | 0x0 | Регистр прерываний по любому событию. Запись «1» – прерывание формируется по любому изменению соответствующего вывода | RW |

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|---------------------|----------|--------------------|--|--------|
| INTR_ANY_EDGE_CLEAR | 0x28 | 0x0 | Регистр прерываний по любому событию. Запись «1» – прерывание не формируется по любому изменению соответствующего вывода | RW |
| INTR_CLEAR | 0x2C | – | Регистр сброса флагов прерываний по событию. Запись «1» – очищает флаг прерывания соответствующего вывода | W1C |

3.16 Контроллер 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 флэш-памяти, а также при необходимости чтение и запись произвольных данных во внешнюю флэш-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; четырехканальный. По умолчанию используется одноканальный режим, двухканальный и четырехканальный 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.

3.16.1 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 однонаправленным линиям (MISO/MOSI), так и в параллельных режимах (двух и четырехбитном), когда данные передаются/принимаются по двум или четырем двунаправленным линиям.

Структурная схема контроллера представлена на рисунке 44.

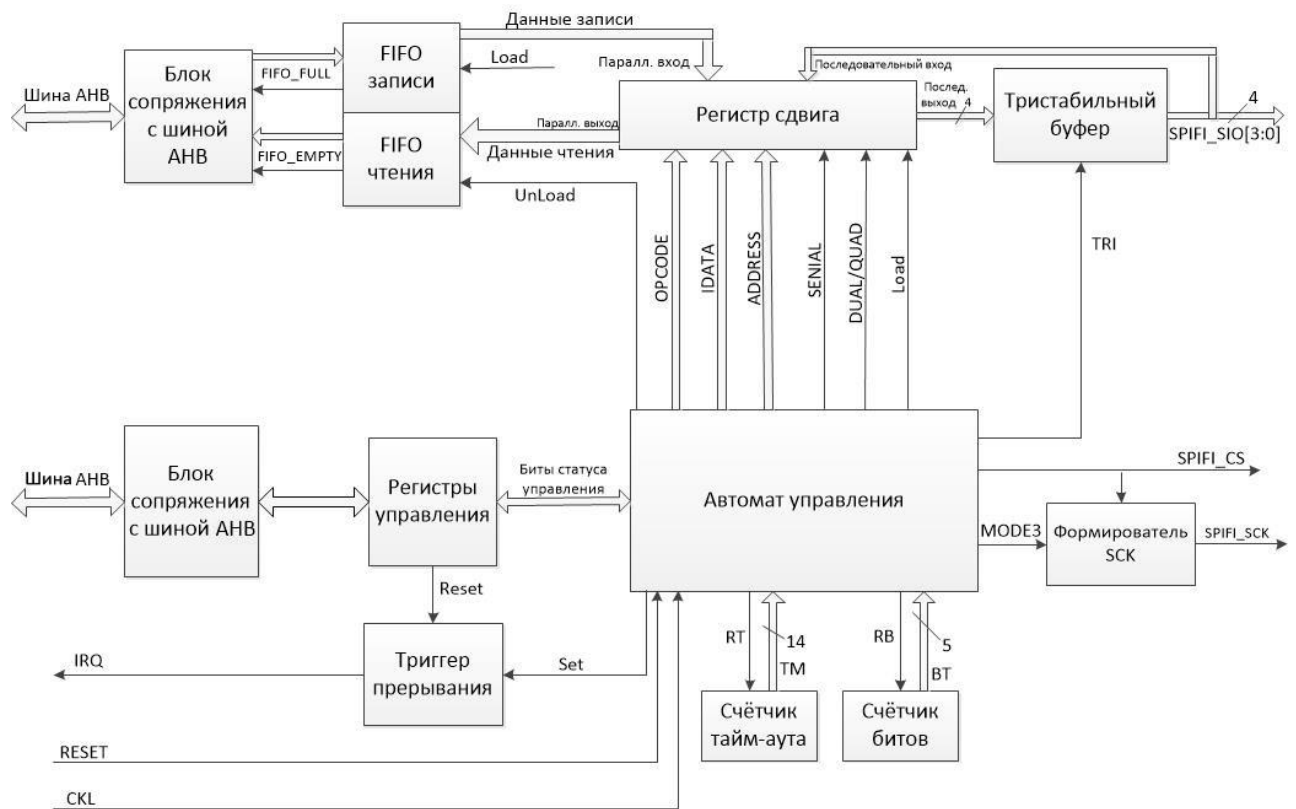


Рисунок 57 – Структурная схема контроллера флэш-памяти SPIFI

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;
- тристабильного 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 получают информацию от блока сопряжения с АНВ шиной. В адресном пространстве FIFO буфера данные находятся по адресу регистра данных DATA.

3.16.2 Регистры контроллера SPIFI

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

Таблица 101 – Перечень регистров контроллера SPIFI

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|--|--------|
| CTRL | 0x000 | 0x400F FFFF | SPIFI регистр управления | RW |
| CMD | 0x004 | 0x0000 0000 | SPIFI регистр команд | RW |
| ADDR | 0x008 | 0x0000 0000 | SPIFI регистр адреса | RW |
| IDATA | 0x00C | 0x0000 0000 | SPIFI регистр промежуточных данных | RW |
| CLIMIT | 0x010 | 0x0000 0000 | SPIFI регистр верхней границы адреса кеширования | RW |
| DATA | 0x014 | 0x0000 0000 | SPIFI регистр данных | RW |
| MCMD | 0x018 | 0x0380 0000 | SPIFI регистр команд памяти | RW |
| STAT | 0x01C | 0x0200 0001 | SPIFI регистр статуса | RW |

3.16.2.1 Регистр управления CTRL

Данный регистр управляет выполнением всех операций контроллера и должен быть настроен перед инициализацией любой команды. Поля данного регистра приведены в таблице 102.

Таблица 102 – CTRL регистр управления

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|-------|--------------------|--|--------|
| TIMEOUT | 15:0 | 0x FFFF | Количество периодов сигнала SPIFI_SCK без чтения данных в режиме работы с памятью, которое вызывает завершение выполнения команды установкой сигнала SPIFI_CS в состояние «1» и сбросом бита CMD | RW |
| CSHIGH | 19:16 | 1111 | Количество периодов сигнала SPIFI_SCK минус один, в течение которых сигнал SPIFI_CS остается в неактивном состоянии перед началом выполнения команды | RW |
| CACHE_EN | 20 | 0 | Бит разрешения кэширования | |
| D_CACHE_DIS | 21 | 0 | Бит запрещения кеширования данных (транзакций АНВ, для которых сигнал HPROT[0] = 1): «0» – при разрешенном кэшировании кэширование данных выполняется; «1» – данные не кэшируются | RW |
| INTEN | 22 | 0 | Бит разрешения прерывания при завершении выполнения команды (если этот бит равен «1», то прерывание разрешено) | RW |
| MODE3 | 23 | 0 | Бит режима 3 | RW |
| SCK_DIV | 26:24 | 0 | 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 |

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|------|--------------------|---|--------|
| PRFTCH_DIS | 27 | 0 | Бит разрешения упреждающих выборок кэш памяти «0» – выборки разрешены | RW |
| DUAL | 28 | 0 | Бит выбора протокола: сигналы IO[3:0] – (0–4) битовый протокол; сигналы IO[1:0] – (1–2) битовый протокол | RW |
| RFCLK | 29 | 0 | Бит выбора активного перепада сигнала для стробирования входных данных: «0» – положительный; «1» – отрицательный | RW |
| FBLK | 30 | 1 | Бит выбора сигнала стробирования входных данных с выходного буфера тактового сигнала: «0» – внутренний сигнал, «1» – внешний сигнал | RW |
| DMAEN | 31 | 0 | Бит разрешения запросов DMA от контроллера SPIFI «1» – запросы разрешены. Устанавливать только в периферийном режиме | RW |

3.16.2.2 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 103.

Т а б л и ц а 103 – CMD регистр команд

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|------|--------------------|------------|--------|
|-------------|------|--------------------|------------|--------|

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|-------|--------------------|--|--------|
| DATALEN | 13:0 | 0 | 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 |
| POLL | 14 | 0 | 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. После завершения процесса поллинга необходимо считать один байт из регистра DATA, так как в нем сохраняется последний принятый статус FLASH. | RW |
| DOUT | 15 | 0 | Бит направления передачи данных «0» – чтение из флэш | RW |
| INTLEN | 18:16 | 0 | Количество байт промежуточных данных, которые хранятся в регистре IDATA. Если количество байт больше, чем может хранить регистр IDATA, то остаток добывается нулевыми значениями | RW |

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

3.16.2.3 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 104.

Таблица 104 – ADDRESS регистр адреса

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|------|--------------------|------------------------------|--------|
| ADDRESS | 31:0 | 0 | Исполнительный адрес команды | RW |

3.16.2.4 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 105.

Таблица 105 – Регистр промежуточных данных

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|------|--------------------|----------------------|--------|
| IDATA | 31:0 | 0 | Промежуточные данные | RW |

3.16.2.5 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 106.

Таблица 106 – Регистр верхней границы адреса кэширования

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|------|--------------------|----------------------------------|--------|
| CLIMIT | 31:0 | 0x0000 0000 | Верхний предел кэшируемой памяти | RW |

3.16.2.6 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 107.

Таблица 107 – Регистр данных DATA

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|-----------------------------|--------|
| DATA | 31:0 | 0 | Входные или выходные данные | RW |

3.16.2.7 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 108.

Таблица 108 – Регистр команд памяти MCMD

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|------|--------------------|----------------------------------|--------|
| Резерв | 13:0 | 0 | – | – |
| POLL | 14 | 0 | Бит должен быть установлен в «0» | RW |
| DOUT | 15 | 0 | Бит должен быть установлен в «0» | RW |

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|-------|--------------------|---|--------|
| INTLEN | 18:16 | 0 | 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 |
| FIELDFORM | 20:19 | 0 | 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 |
| FRAMEFORM | 23:21 | 4 | 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 |

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|-------|--------------------|--|--------|
| OPCODE | 31:24 | 3 | Код операции (не используется для некоторых значений поля FRAMEFORM) | RW |

3.16.2.8 Регистр STAT

Данный регистр содержит текущее состояние контроллера. Поля данного регистра приведены в таблице 109.

Таблица 109 – Регистр статуса STAT

| Обозначение | Биты | Начальное значение | Назначение | Доступ |
|-------------|------|--------------------|---|--------|
| MCINIT | 0 | 1 | Бит устанавливается в «1», если программное обеспечение успешно записало регистр команд памяти. Сброс бита осуществляется аппаратным сбросом или установкой в «1» бита RESET | RW |
| CMD | 1 | 0 | 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 |
| – | 3:2 | – | Зарезервировано | – |
| RESET | 4 | 0 | Бит предназначен для того, чтобы прервать текущую команду периферийного режима или режима памяти. Бит сбрасывается, когда контроллер готов к выполнению новой команды | RW |

| | | | | |
|---------|-------|------|---|----|
| INTRQ | 5 | 0 | Запись «1» в бит сбрасывает запрос на прерывание от контроллера SPIFI | RW |
| – | 23:6 | – | Зарезервировано | – |
| VERSION | 31:24 | 0x02 | Версия контроллера SPIFI (поле доступно только для чтения) | RW |

3.16.3 Структура и принцип работы контроллера SPIFI с КЭШ

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

Структура SPI Flash контроллера с КЭШ представлен на рисунке 42.

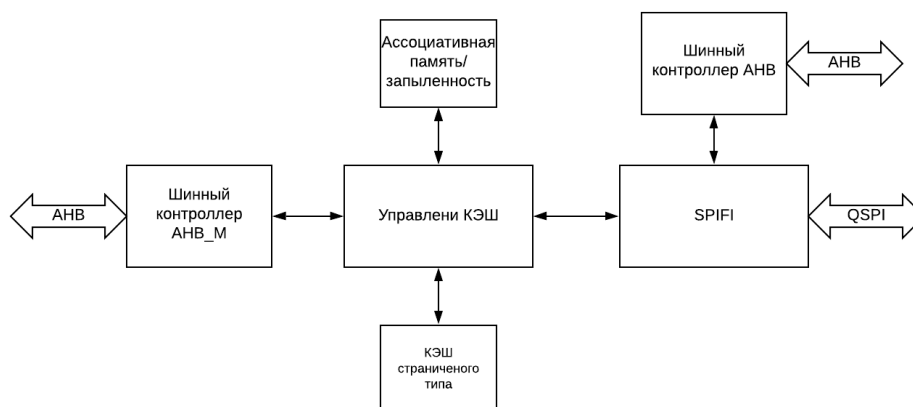


Рисунок 58 – Общая структура контроллера

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 АНВ_М возможны только транзакции чтения. Шинный контроллер АНВ_М осуществляет анализ транзакций на шине АНВ_М. 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.

3.16.3.1 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 кэширования CACHE_EN;
- осуществляется переход на адрес кода программы для чтения первой инструкции.

Далее работа происходит следующим образом, при обращении мастера АНВ_М шины КЭШ контроллер производит анализ транзакции (в данной версии анализируются только транзакции чтения). КЭШ контроллер по значению HADDR[31:10] и содержимому полей ТЭГ ассоциативной памяти, а так же 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[9:2].

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 АНВ_М, 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 запи-

си. 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 $D_CACHE_DIS=0$.

3.16.3.2 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 AMBA АНВ.

Блок КЭШ представляет собой блок памяти из 8 страниц КЭШ инструкций по 128 байт каждая. Структура КЭШ представлена на рисунке 59. 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.

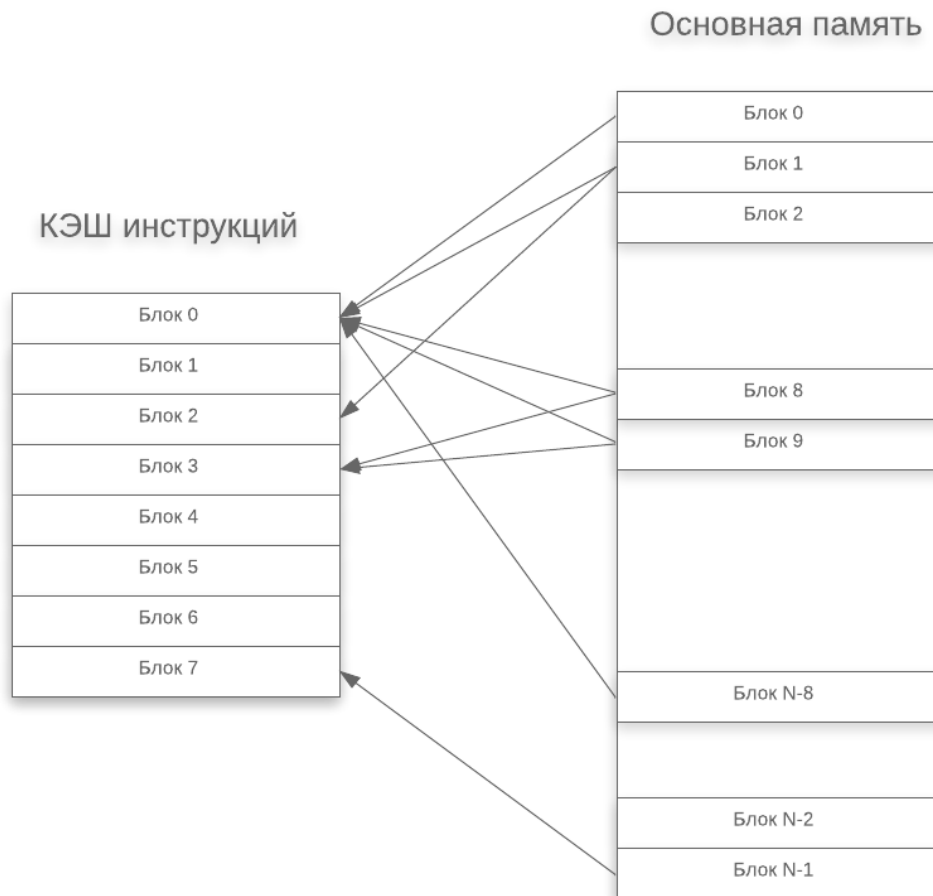


Рисунок 59 – Структура отображения блока КЭШ

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

Ассоциативная память построена на триггерах (защелках) и имеет размер 8x25 бит (может быть скорректировано при ограничении адресного пространства).

В ассоциативной памяти хранится ТЭГ отображенного 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 попада-

ют 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 110.

Таблица 110 – Структура ассоциативной памяти КЭШ

| ТЭГ | Проверки адресов |
|-------------------|---|
| ТЭГ1: HADDR[31:7] | СЛОВО0: HADDR[6:2] ... СЛОВО127: HADDR[6:2] |
| ТЭГ2: HADDR[31:7] | |
| ТЭГ3: HADDR[31:7] | |
| ТЭГ4: HADDR[31:7] | |
| ТЭГ5: HADDR[31:7] | |
| ТЭГ6: HADDR[31:7] | |
| ТЭГ7: HADDR[31:7] | |
| ТЭГ8: HADDR[31:7] | |

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 установятся 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.

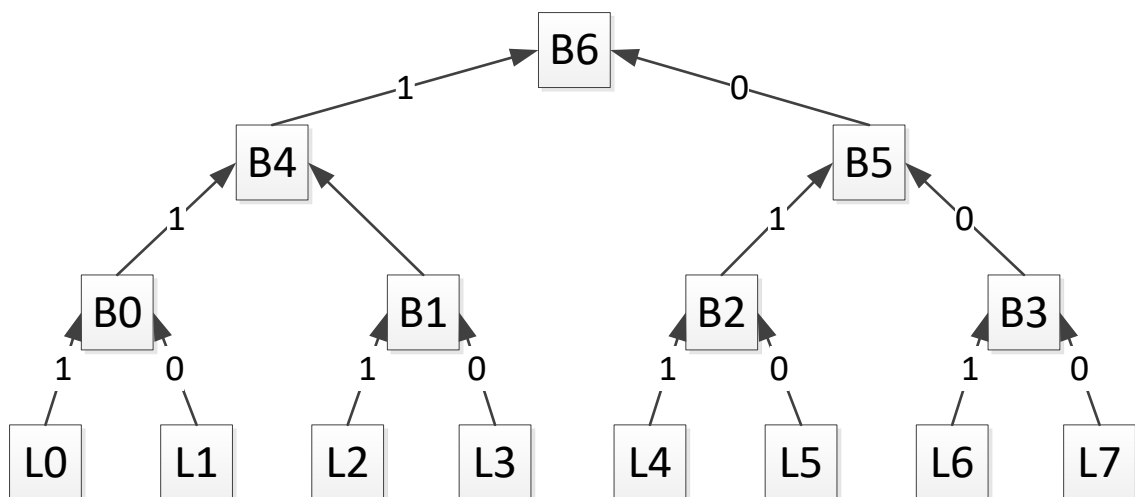


Рисунок 60 – Диаграмма работы алгоритма PLRU

Тогда таблица замещения будет выглядеть, как представлено в таблице 86.

Таблица 111 – 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 |

3.16.4 Работа контроллера SPIFI интерфейса флэш-памяти

3.16.4.1 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 АНВ 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 тристабильный 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).

3.16.4.2 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 коман-

ды. 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 (12):

$$N = N_b / R, \quad (12)$$

где N_b – 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.

3.16.4.3 Алгоритм работы автомата контроллера в периферийном режиме

Алгоритм работы автомата контроллера в периферийном режиме приведен на рисунке 61.

Автомат ждет сигнала начала работы CMD (бит 1 регистра статуса). Как только этот бит устанавливается в «1», что происходит в случае удачной 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 IFIFO, либо считает из МФП в буфер OFIFO байты данных в количестве, равном значению DATALEN[13:0], после чего автомат закончит свою работу.

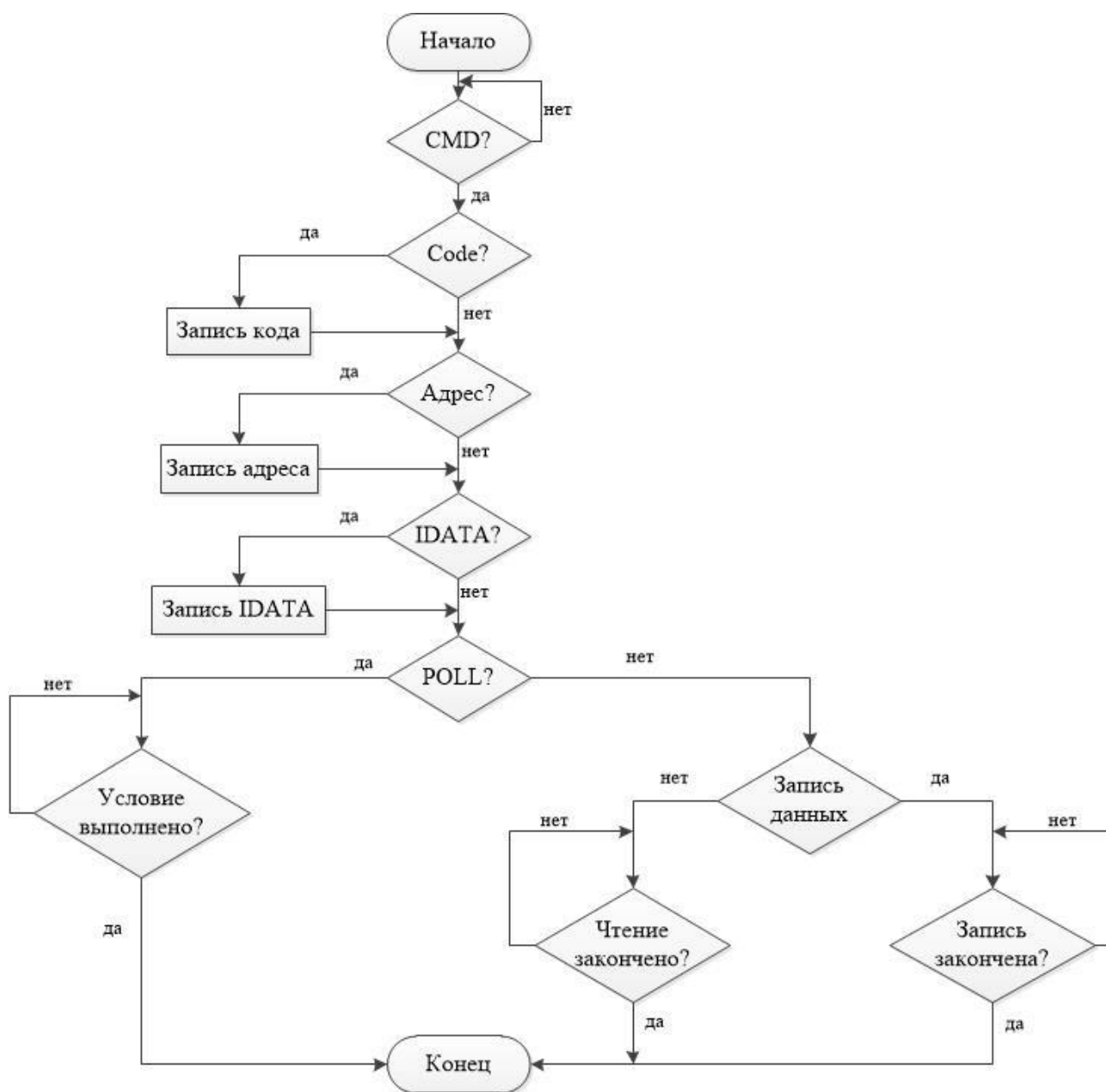


Рисунок 61 – Алгоритм работы автомата контроллера в периферийном режиме

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. Передача осуществляется постоянно (repetitive) до момента завершения процесса программирования. В байте статуса имеется бит окончания про-

цесса программирования, который устанавливается в определенное состояние в 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 бита 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.

3.16.4.4 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 62.

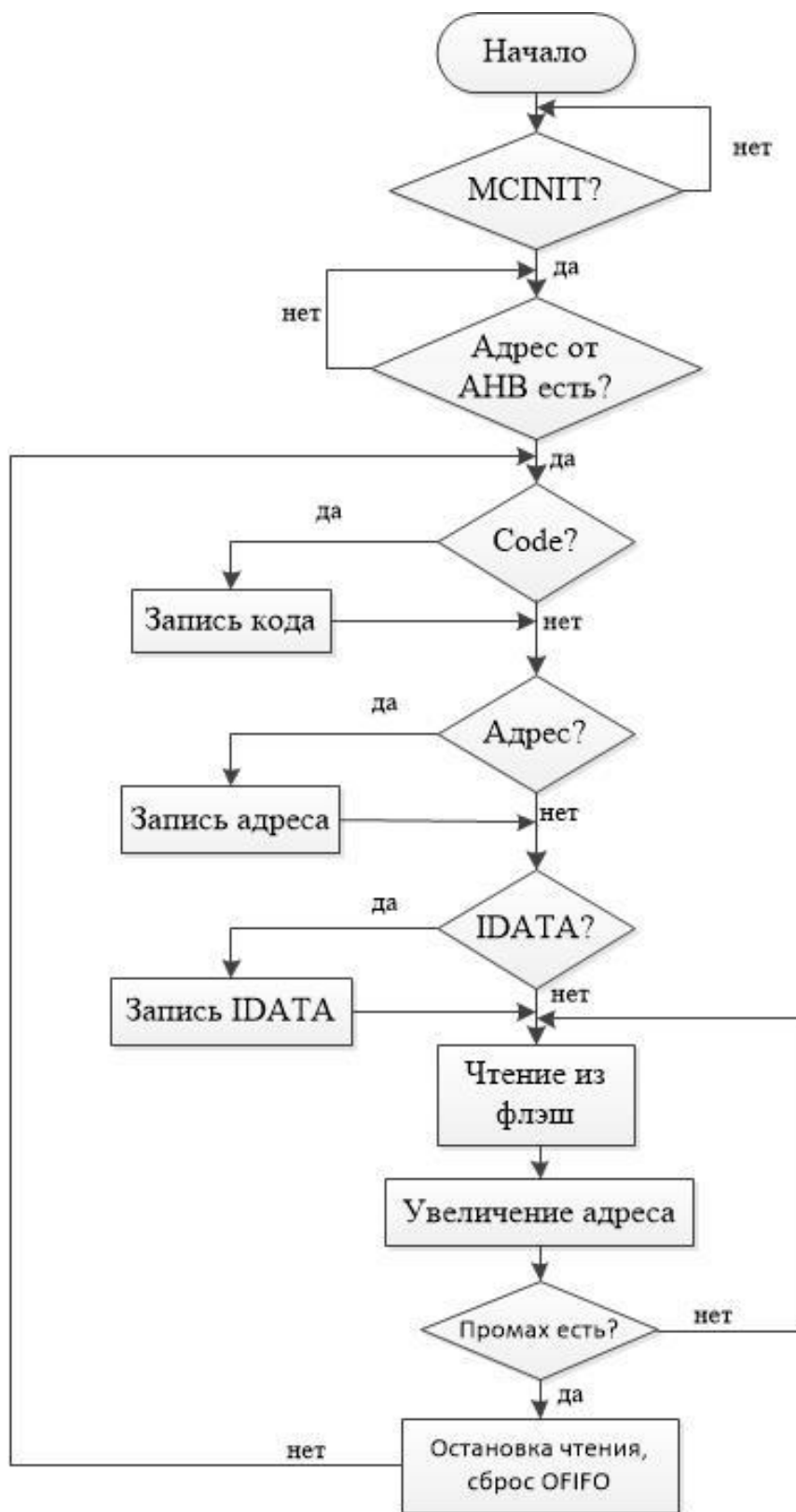


Рисунок 62 – Алгоритм работы автомата контроллера в режиме работы с памятью

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.

3.16.5 Базовые спецификации SPI интерфейса и Flash драйвера

Базовые спецификации SPI интерфейса и Flash драйвера:

- полнодуплексный синхронный режим работы для стандартного режима SPI;
- полудуплексный 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 63 показан простейший случай, когда команда состоит только из одной инструкции. Такая команда, например, может быть использована для разблокирования режима программирования памяти.

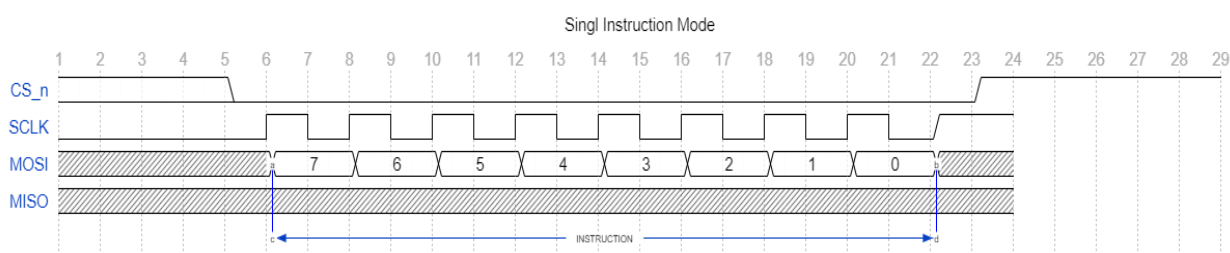


Рисунок 63 – Временная диаграмма команды из одной инструкции

На рисунке 64 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.

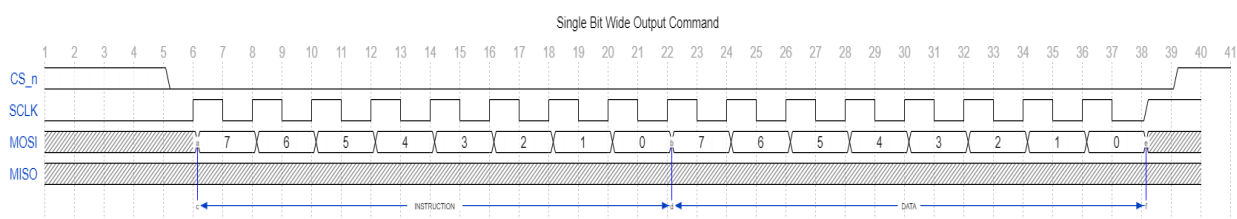


Рисунок 64 – Временная диаграмма инструкции без данных без добавления адреса и задержки

На рисунке 65 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.

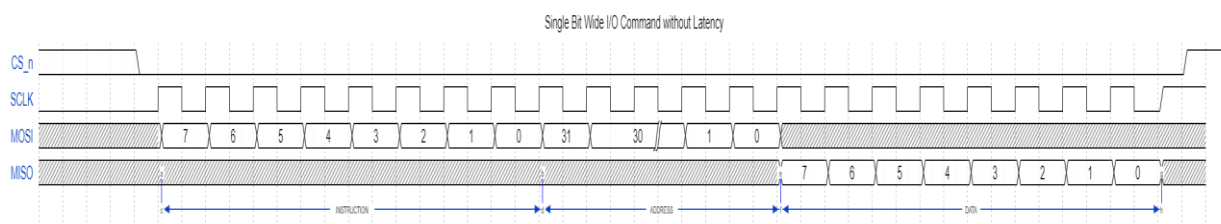


Рисунок 65 – Временная диаграмма инструкции с данными без адреса

На рисунке 66 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.

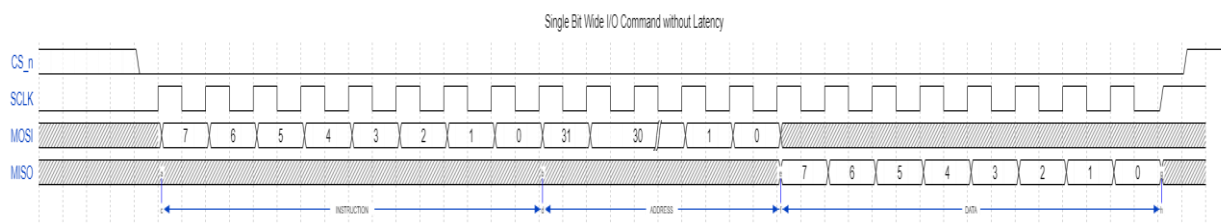


Рисунок 66 – Временная диаграмма инструкции с адресом и принимаемыми данными

На рисунке 67 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 уточнять в технической документации на соответствующую микросхему.

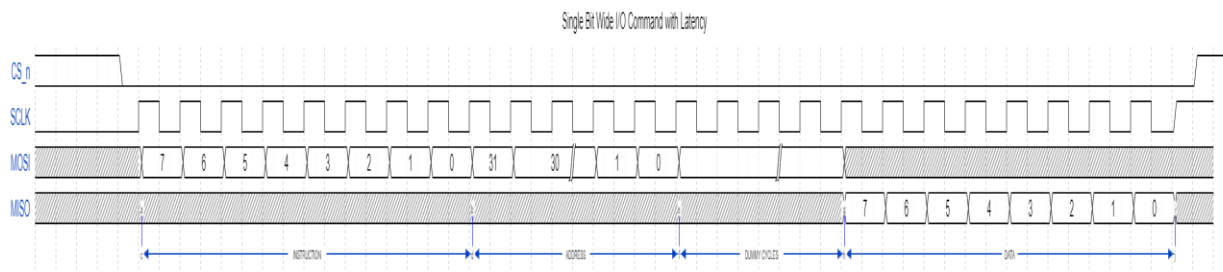


Рисунок 67 – Временная диаграмма инструкции с адресом и принимаемыми данными и задержкой

3.17 Кристо-блок

Кристо-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;
- «Кузнечик», ГОСТ 34.12–2018 128-битный блочный шифр.

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

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

Условная структурная схема кристо-блока представлена на рисунке 68.

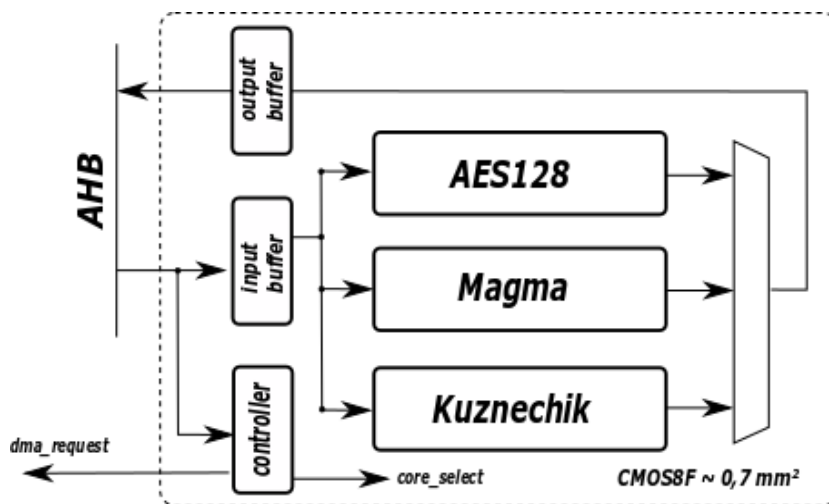


Рисунок 68 – Структурная схема крипто-блока

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 112.

Таблица 112 – Регистры крипто-блока

| Регистр | Смещение | Описание | Доступ |
|---------|----------|------------------------------------|--------|
| BLOCK | 0x0 | Буфер данных для шифрования | RW |
| KEY | 0x4 | Буфер данных ключа | W |
| INIT | 0x8 | Буфер данных вектора инициализации | W |
| CONFIG | 0xC | Регистр конфигурации | RW |

Описание полей регистра конфигурации приведено в таблице 113.

Таблица 113 – Описание полей регистра конфигурации

| Поле | Разряд | Описание | Доступ |
|----------|--------|---|--------|
| decode | 0 | Задаёт режим хода вычислительного ядра: «0» – прямой ход (шифрование); «1» – обратный ход (расшифровка) | RW |
| core_sel | 2:1 | Регистр выбора алгоритма шифрования: «0» – «Кузнечик»; «1» – «Магма»; «2» – «AES» | RW |

| Поле | Разряд | Описание | Доступ |
|--------------|--------|--|--------|
| mode_sel | 4:3 | Регистр выбора режима шифрования: «0» – ECB; «1» – CBC; «2» – CTR | RW |
| swap_mode | 6:5 | Регистр выбора режима перестановки слова: «0» – нет перестановки; «1» – перестановка по полуслову; «2» – перестановки по байтам; «3» – перестановка по битам | RW |
| order_mode | 7 | Выбор порядка загрузки/выгрузки: «0» – от младшего слова к старшему; «1» – от старшего слова к младшему | RW |
| c_reset | 8 | Сброс счётчиков загружаемых/выгружаемых данных | S |
| Ready | 9 | Состояние доступности модуля: «0» – занято; «1» – доступно | R |
| Write status | 10 | Буфер данных доступен для записи: «0» – недоступно; «1» – доступно | R |
| Read status | 11 | Буфер данных доступен для чтения: «0» – недоступно; «1» – доступно | R |

3.17.1 Порядок работы с крипто-блоком

Алгоритм «Кузнечик»:

- 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.

– Алгоритмы «Магма» и AES128

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.

3.18 Контроллер блока однократнопрограммируемой памяти OTP

Данный блок предназначен для осуществления доступа к массиву блока antifuse OTP на запись и чтение со стороны интерфейса APB. Данный 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 69 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.

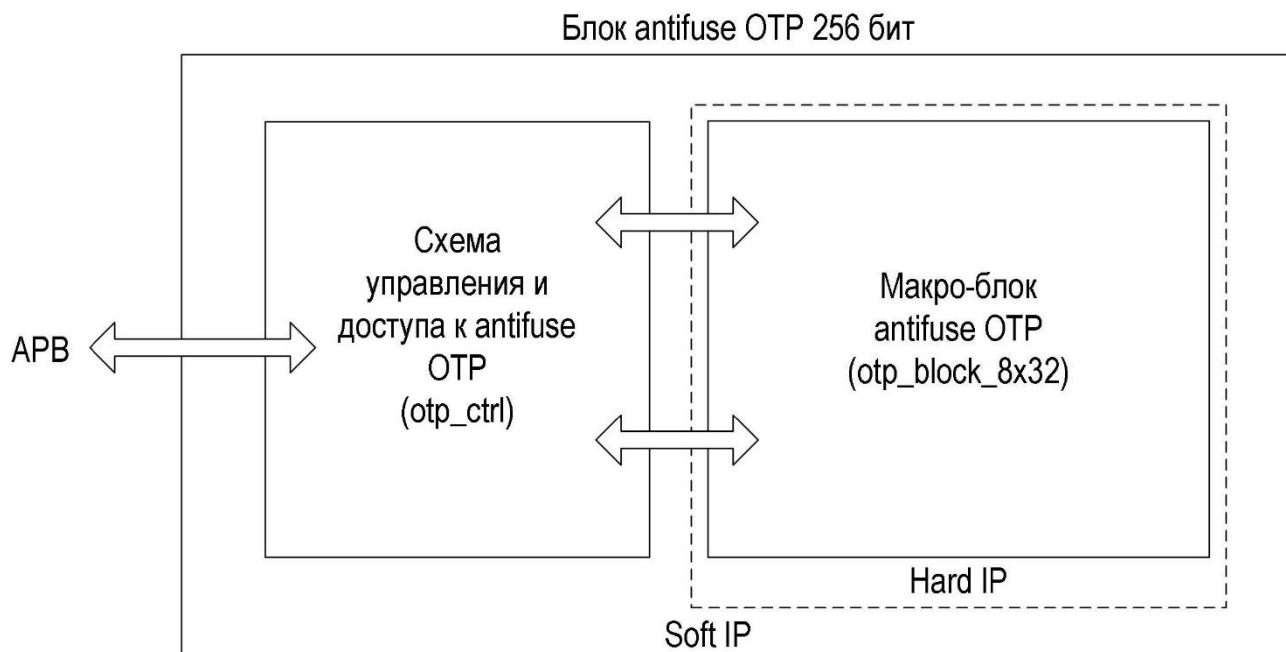


Рисунок 69 – Схема блока OTP

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;
- возможность ручного формирования входных сигналов Hard IP.

3.18.1 Режимы чтения по интерфейсу APB

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

- запись адреса (OTPA) – чтение регистра данных (OTPDAT);

– запись адреса (ОТРА) – опрос флага готовности (ОТПСТА.BSY) – чтение регистра данных (ОТПДАТ).

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 (ОТПСТА.BSY). Для переключения между режимами используется регистр ОТПСОН.АРБНВС. При ОТПСОН.АРБНВС = 1 используется режим чтения в три этапа (без тактов ожидания APB).

Если ОТПСОН.АРБНВС = 0 при чтении регистра ОТПДАТ происходит автоматическое инкрементирование адреса ОТРА. Таким образом, запись адреса не требуется, если необходимо считать слово, имеющее адрес на один больше предыдущего. При автоматическом инкрементировании адреса происходит только изменение разрядов ОТРА[2:0]. Если выполняется инкрементирование адреса, содержащего ОТРА[2:0] = 111b, то новое значение будет содержать ОТРА[2:0] = 000b, а другие разряды не изменятся. Если ОТПСОН.АРБНВС = 1 при чтении регистра ОТПДАТ автоматическое инкрементирование адреса ОТРА не происходит.

Если ОТПСОН.АРБНВС = 0, контроллер осуществляет автоматическое считывание слова из Hard IP при каждом чтении регистра ОТПДАТ.

Если ОТПСОН.АРБНВС = 1, контроллер осуществляет автоматическое считывание слова из Hard IP при каждой записи в регистр ОТРА.

3.18.2 Последовательность записи слова

Для записи слова рекомендуется следовать описанной ниже последовательности действий:

– выполнить запись адреса в регистр ОТРА. Если выполняется запись тестового слова, то ОТРА[3] = 1, иначе – ОТРА[3] = 0. Если выполняется запись тестового столбца, то ОТРА[4] = 1, иначе – ОТРА[4] = 0;

– выполнить запись данных в регистр ОТПДАТ;

– опрашивать флаг ОТПСТА.BSY до тех пор, пока он не будет очищен.

По окончании процедуры записи происходит автоматическое инкрементирование адреса ОТПА. 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 ОТПА[2:0]. Если выполняется инкрементирование адреса, содержащего ОТПА[2:0] = 111b, то новое значение будет содержать ОТПА[2:0] = 000b а другие разряды не изменятся.

3.18.3 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 70 представлены формы сигналов в процедуре записи. На рисунке 71 представлены формы сигналов в процедуре чтения.

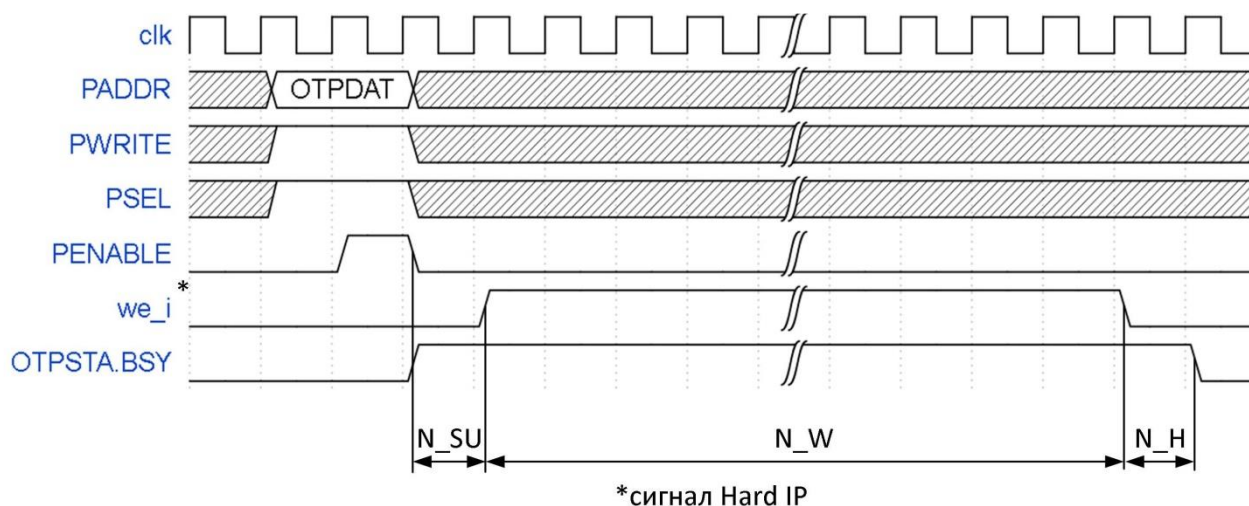


Рисунок 70 – Формы сигналов в процедуре записи

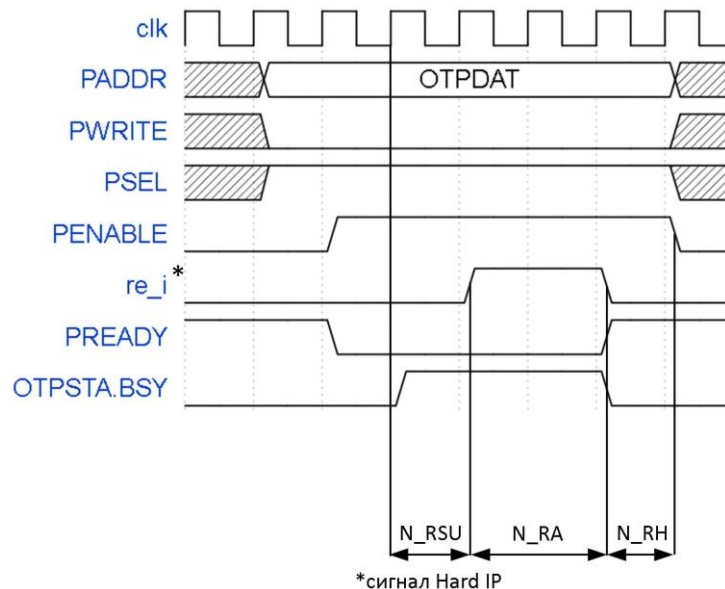


Рисунок 71 – Формы сигналов в процедуре чтения

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 $N_{SU} = \text{ceil}(40/P_{clk})$, где P_{clk} - период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа. Пример: 2 такта для частоты $F_{clk} = 33,3$ МГц;
- N_W – длительность высокого уровня сигнала we_i (вход Hard IP) в тактах. Рекомендуемое значение $N_W = 50\,000\,000 \text{ нс} / P_{clk}$, где P_{clk} – период тактового сигнала в нс. Пример: 1 666 667 тактов для частоты $F_{clk} = 33,3$ 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 $N_H = \text{ceil}(40/P_{clk})$, где P_{clk} – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа. Пример: 2 такта для частоты $F_{clk} = 33,3$ МГц. На рисунках 70 и 71 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 ОТПА, 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 $N_{RA} = \text{ceil}(40/Pclk)$, 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 $N_{RH} = \text{ceil}(40/Pclk)$, где $Pclk$ – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа. Пример: 2 такта для частоты $Fclk = 33,3$ МГц;
- Согласно спецификации Hard IP, после окончания операции записи до начала следующей за ней операции чтения должно пройти не менее 1 мкс. Согласно спецификации Hard IP, после окончания операции чтения до начала следующей за ней операции записи должно пройти не менее 1 мкс.

3.18.4 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 ОТП необходимо установить $ADDR[4:3] = 00b$, разряды $ADDR[2:0]$ используются для выбора строки основного массива. Для выбора основных 32-х ячеек тестовой строки необходимо установить $ADDR[4:3] = 01b$, разряды $ADDR[2:0]$ игнорируются. Для выбора тестового столбца (1 ячейка) тестовой строки необходимо установить $ADDR[4:3] = 11b$, разряды $ADDR[2:0]$ игнорируются. Для выбора тестового столбца, прилегающего к основному массиву, необходимо установить $ADDR[4:3] = 10b$, 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 0b, кроме младшего.

3.18.5 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 вывода 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.

3.18.6 Регистры

Список регистров и их назначение представлено в таблице 114. В таблицах 113–120 предоставлены регистры контроллера OTP.

Таблица 114 – Регистры контроллера OTP

| Смещение | Обозначение | Описание |
|----------|-------------|---|
| 0x00 | OTPDAT | Регистр чтения/записи данных |
| 0x04 | OTPA | Регистр адреса страницы |
| 0x08 | OTPCON | Регистр управления |
| 0x0C | OTPSTA | Регистр статуса |
| 0x10 | OTPDEC | Регистр бит дешифратора |
| 0x14 | OTPADJ | Регистр управления временными параметрами процедуры чтения и доп. настройками |
| 0x18 | OTPWT1 | Регистр подстройки длительности процедуры записи 1 |
| 0x1C | OTPWT2 | Регистр подстройки длительности процедуры записи 2 |

Таблица 115 – Регистр чтения/записи данных OTPDAT

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-------------|------|---------------------|---|--------|
| DATA | 31:0 | неопределено | 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 регистра считываются данные, полученные из OTP | RW |

Таблица 116 – Регистр адреса страницы ОТПА

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-----------------|------|---------------------|---|--------|
| ADDR | 4:0 | 0x0 | 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 |
| Зарезервировано | 31:5 | Зарезервировано | | |

Таблица 117 – Регистр управления OTPCON

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-------------|------|---------------------|---|--------|
| 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 ОТПА до тех пор, пока EESTA.BSY не станет равным 0 | WO |
| man we i | 1 | 0x0 | Ручное управление сигналом we i блока Hard IP. Служит для тестовых целей, использовать не рекомендуется | WO |

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-----------------|------|---------------------|--|--------|
| man re i | 2 | 0x0 | Ручное управление сигналом re i блока Hard IP. Служит для тестовых целей, использовать не рекомендуется | WO |
| MME | 3 | 0x0 | Режим ручного управления включен. В этом режиме автоматизированные операции чтения и записи не выполняются. Служит для тестовых целей, использовать не рекомендуется | WO |
| Зарезервировано | 31:4 | Зарезервировано | | |

Таблица 118 – Регистр статуса OTPSTA

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-----------------|------|---------------------|---|--------|
| BSY | 0 | 0x0 | Блок занят (выполняется запрошенная операция) | RO |
| Зарезервировано | 31:1 | Зарезервировано | | |

Таблица 119 – Регистр бит дешифратора OTPDEC

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-----------------|------|---------------------|---|--------|
| DECO | 8:0 | 0x0 | Служит для отображения выхода дешифратора строк блока antifuse OTP для целей тестирования | RO |
| Зарезервировано | 31:9 | Зарезервировано | | |

Таблица 120 – Регистр управления временными параметрами ОТПАДЖ

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-----------------|-------|---------------------|---|--------|
| N_RSU | 2:0 | 0x0 | Время между моментом начала транзакции на APB и положительным фронтом ge_i в тактах. Должно использоваться для обеспечения требования к временам предустановки Hard IP. Учитывая то, что на входе i удерживаются стабильные уровни в результате предыдущей операции записи в регистр ОТПА, при частотах ниже 200 МГц данное значение рекомендуется устанавливать равным 0 | RW |
| Зарезервировано | 7:3 | Зарезервировано | | |
| N_RA | 10:8 | 0x1 | Длительность высокого уровня сигнала ge_i (вход Hard IP) в тактах. Рекомендуемое значение $N_{RA} = \text{ceil}(40/P_{clk})$, где P_{clk} – период тактового сигнала в ceil – функция округления до ближайшего большего целого числа. Пример: два такта для частоты $F_{clk} = 33,3$ МГц. Для корректной работы схемы значение должно быть больше 0 | RW |
| Зарезервировано | 15:11 | Зарезервировано | | |

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|---------------------|-------|---------------------|---|--------|
| N_RH | 18:16 | 0x1 | <p>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 $N_RH = \lceil 40/Pclk \rceil$, 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 | 23:19 | Зарезервировано | | |
| sel_upp_read_i[2:0] | 26:24 | 0x1 | <p>Выбор напряжения на UPP матрицы: X00b – 2,0 В; X01b – 2,5 В; X11b – 3,0 В; 010b – VDD18; 110b – VDD5</p> | RW |
| sel_read_cur_i | 27 | 0x0 | <p>Выбор тока считывания: «0» – 2 мкА; «1» – 0,2 мкА</p> | RW |
| power_off_i | 28 | 0x0 | <p>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.</p> <p>После вывода Hard IP из режима пониженного энергопотребления требуется некоторое время перед тем, как могут быть начаты новые операции</p> | RW |

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-----------------|-------|---------------------|----------|--------|
| Зарезервировано | 31:29 | Зарезервировано | | |

Таблица 121 – Регистр подстройки длительности процедуры записи 1 OTPWT1

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-----------------|-------|---------------------|---|--------|
| 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 пред-установки Hard IP. Рекомендуемое значение $N_SU = \text{ceil}(40/Pclk)$, 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 |
| Зарезервировано | 7:3 | 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 $N_H = \text{ceil}(40/Pclk)$, где $Pclk$ – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа.</p> <p>Пример – 1df такта для частоты $Fclk = 33,3$ МГц. Для корректной работы схемы значение должно быть больше 0</p> | RW |
| Зарезервировано | 31:11 | Зарезервировано | | |

Таблица 122 – Регистр подстройки длительности процедуры записи 2 OTPWT2

| Обозначение | Биты | Значение при сбросе | Описание | Доступ |
|-----------------|-------|---------------------|---|--------|
| N_W | 23:0 | 1666667 | Длительность высокого уровня сигнала we i (вход Hard IP) в тактах. Рекомендуемое значение $N_W = 50000000 \text{ нс}/Pclk$, где Pclk – период тактового сигнала в нс. Пример – 1666667 тактов для частоты Fclk = 33,3 МГц | RW |
| Зарезервировано | 31:24 | Зарезервировано | | |

3.19 Контроллер EEPROM

Данный контроллер предназначен для осуществления доступа к массиву блока EEPROM на запись и чтение со стороны интерфейсов ANB-Lite (только чтение) и APB (чтение и запись в несколько этапов). Данный контроллер является совместимым с Hard IP EEPROM объемом 8 Кбайт с организацией (32+6)x2k, разработанным для серийной технологии CMOSF8_4M_5V ПАО Микрон (F8 EEPROM High Density 8 Kbytes (32+6)x2k Macro (core)) и требуемым для него блоком формирования напряжений и токов (Common Supply Block for F8 EEPROM High-Density macros(supply)). Иерархически блоки Hard IP core и supply помещаются внутрь данного Soft IP. Имя блока верхнего уровня, содержащего ee_ctrl и подключенный к нему Hard IP, - EEPROM_8K_BLK.

Структурная схема контроллера EEPROM представлена на рисунке 72.

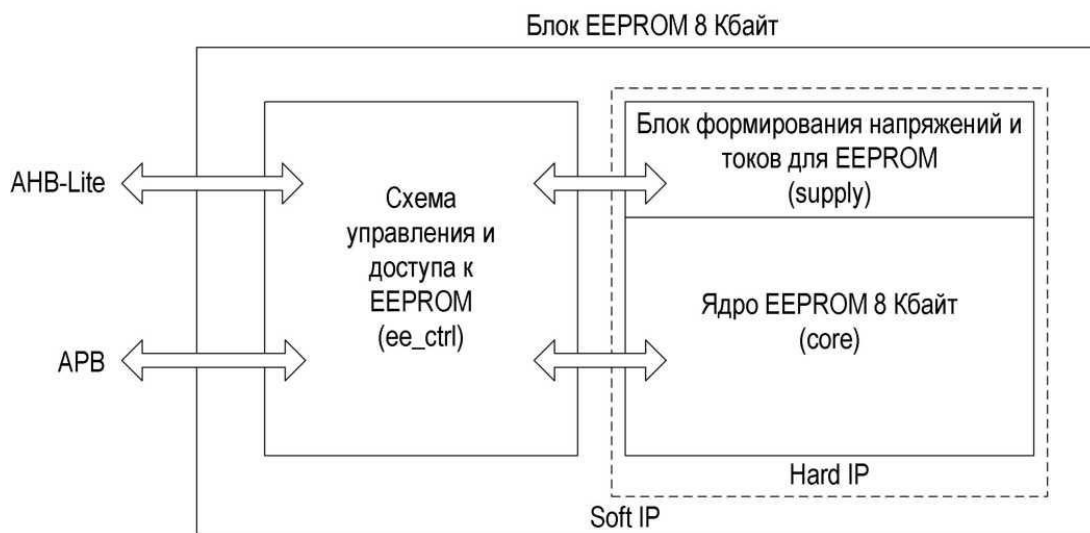


Рисунок 72 – Структурная схема контроллера EEPROM

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 АРВ-интерфейс в несколько этапов;
- чтение через АРВ-интерфейс в 2 или 3 этапа;
- чтение через АНВ-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.

3.19.1 Режимы чтения по интерфейсу АРВ

Контроллер позволяет выполнять чтение по интерфейсу АРВ 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 слова считываются 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.

Чтоб считывать фактическое значение, хранимое в EEPROM, а не скорректированное, перед чтением необходимо отключить схему коррекции ошибок, записав «1» в бит EECON.DISECC.

3.19.2 Чтение данных через АНВ-Lite-интерфейс

В данном блоке реализована операция чтения данных через АНВ-Lite-интерфейс.

Одновременно возможно работать только с одним интерфейсом, либо с АНВ-Lite, либо с APB. Следовательно, на момент начала передачи по АНВ-Lite блок должен находиться в режиме ожидания поступления новых команд (EESTA.BSY=0).

3.19.3 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.WRБЕН = 00b). 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 WRБЕН[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.

3.19.4 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 = 01b (стирание);
 - 3) EECON.WRVEN = 00b (обычное) или 01b (только четные страницы) или 10b (только нечетные страницы) или 11b (все страницы);
- опрашивать флаг EESTA.BSY до тех пор, пока он не будет очищен.

3.19.5 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 = 10b (программирование);
 - 3) EECON.WRVEN = 00b (обычное) или 01b (только нечетные страницы) или 10b (только четные страницы) или 11b (все страницы);
- опрашивать флаг EESTA.BSY до тех пор, пока он не будет очищен.

3.19.6 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.

3.19.7 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 73 – 75.

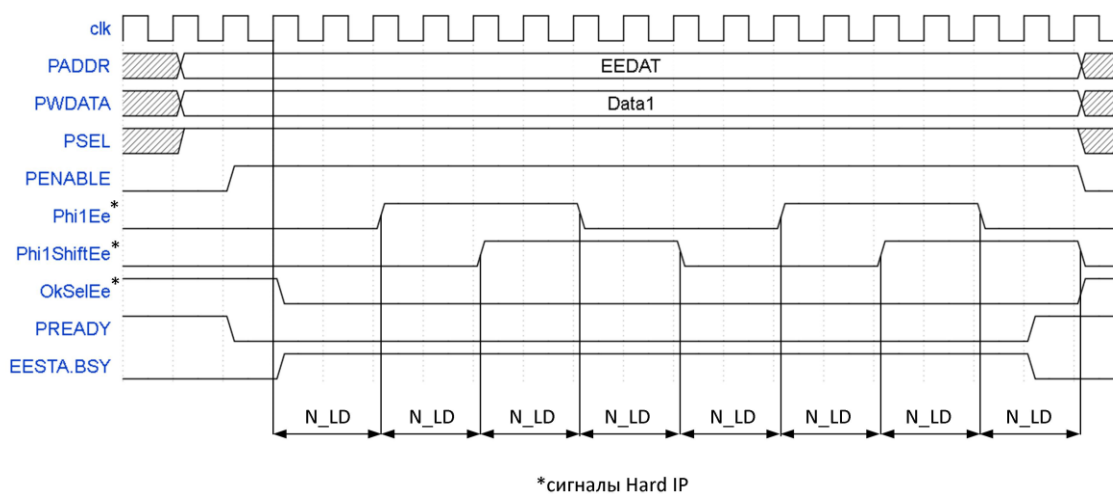


Рисунок 73 – Формы сигналов в процедуре заполнения буфера записи

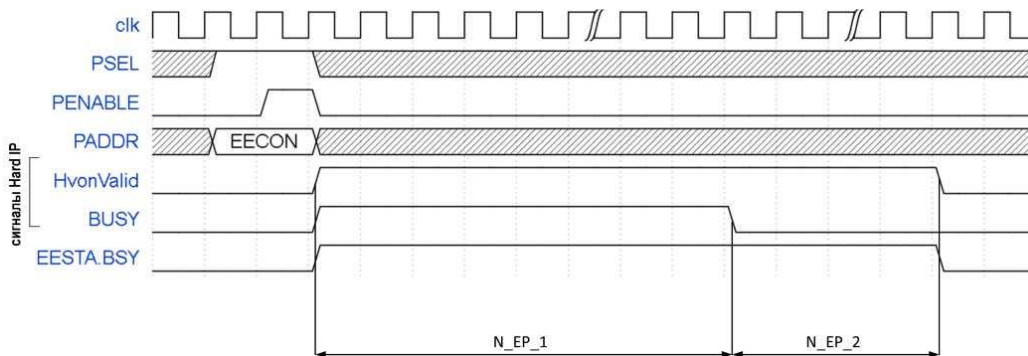
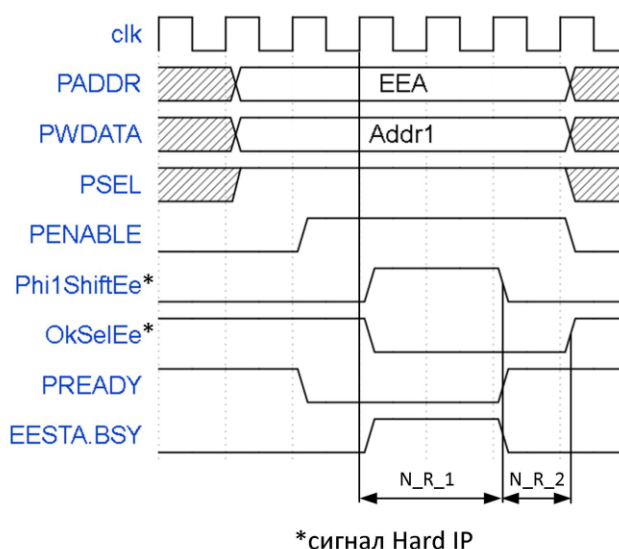


Рисунок 74 – Формы сигналов в процедурах стирания и программирования



*сигнал Hard IP

Рисунок 75 – 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 $N_{EP_1} = 2000000 \text{ нс} / Pclk$, где $Pclk$ – период тактового сигнала в нс. Пример – 66667 тактов для частоты $Fclk = 33,3 \text{ МГц}$;
- N_{EP_2} – задержка между задними фронтами BUSY и HvonValid (входы Hard IP) в тактах. Рекомендуемое значение $N_{EP_2} = 15000 \text{ нс} / Pclk$, где $Pclk$ – период тактового сигнала в нс. Пример – 500 тактов для частоты $Fclk = 33,3 \text{ МГц}$;
- 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.

3.19.8 Регистры

Список регистров и их назначение представлено в таблице 123.

Т а б л и ц а 123 – Регистры контроллера EEPROM

| Смещение | Обозначение | Описание |
|----------|-------------|--|
| 0x00 | EEDAT | Регистр чтения/записи данных |
| 0x04 | EEA | Регистр адреса страницы |
| 0x08 | EECON | Регистр управления |
| 0x0C | EESTA | Регистр статуса |
| 0x10 | EERB | Регистр бит коррекции прочитанного слова |
| 0x14 | EEADJ | Регистр настроек |
| 0x18 | NCYCRL | Регистр подстройки длительности процедур чтения и заполнения буфера записи |
| 0x1C | NCYCEP1 | Регистр 1 подстройки длительности процедур стирания и программирования |
| 0x20 | NCYCEP2 | Регистр 2 подстройки длительности процедур стирания и программирования |

В таблицах 123 – 132 представлены описания регистров модуля.

Таблица 124 – Регистр чтения/записи данных EEDAT

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-------------|------|---------------------|---|--------|
| DATA | 31:0 | x | В процессе чтения/записи данного регистра производится автоматическое инкрементирование адреса (ЕЕА). | RW |

Таблица 125 – Регистр адреса страницы ЕЕА

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-----------------|-------|---------------------|--|--------|
| Зарезервировано | 1:0 | 0x0 | Резерв | RO |
| ADDR | 12:2 | x | Адрес слова для выполнения стирания/программирования /чтения. Если необходимо выполнить стирание или программирование, данный адрес должен использоваться в процедуре заполнения буфера записи | WO |
| Зарезервировано | 31:13 | | Зарезервировано | |

Таблица 126 – Регистр управления EЕCON

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-------------|------|---------------------|--|--------|
| EX | 0 | 0x0 | Запуск процедуры. Запись 1 в данный бит инициирует выполнение процедуры, процедура определяется битами ОР | WO |
| ОР | 2:1 | 0x0 | Выбор процедуры: 00b – чтение страницы; 01b – стирание; 10b – программирование; 11b – значение не используется | WO |

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-------------|------|---------------------|--|--------|
| WRVEN | 4:3 | 0x0 | Поведение операции стирания/программирования: 00b – обычное стирание/программирование; 01b – стирание/программирование всех четных страниц; 10b – стирание/программирование всех нечетных страниц; 11b – стирание/программирование всех страниц. Данные биты должны быть установлены перед операцией заполнения буфера записи перед операциями стирания и программирования | WO |
| 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 | WO |
| DISECC | 6 | 0x0 | Отключение схемы коррекции ошибок во время выполнения запрошенной процедуры. 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 | WO |

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-----------------|------|---------------------|--|--------|
| 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) | WO |
| IESERaR | 8 | 0x0 | Разрешение запроса прерывания при поднятии флага SERR: 0 – запрос не поступает; 1 – запрос поступает | WO |
| Зарезервировано | 31:9 | Зарезервировано | | |

Таблица 127 – Регистр статуса EESTA

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-----------------|------|---------------------|---|--------|
| BSY | 0 | 0x0 | Блок занят (выполняется запрошенная операция) | RO |
| 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 | WO |
| Зарезервировано | 31:2 | Зарезервировано | | |

Таблица 128 – Регистр бит коррекции прочитанного слова EERB

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-----------------|------|---------------------|---|--------|
| CORRECT | 5:0 | 0x0 | Одержит 6 бит коррекции последнего считанного слова | RO |
| Зарезервировано | 31:6 | Зарезервировано | | |

Таблица 129 – Регистр настроек блоков Hard IP EEADJ

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-----------------|-------|---------------------|---|--------|
| OSCX2 | 0 | 0x0 | Увеличение частоты тактового сигнала, поступающего на устройство с зарядовой накачкой, в два раза | RW |
| HivCtl2:0 | 3:1 | 0x0 | Подстройка уровня внутреннего сигнала HIV | RW |
| PlusImix | 4 | 0x0 | Сигнал управления блоком мониторинга аналоговых сигналов | RW |
| VboostCtl1:0 | 6:5 | 0x0 | Старший разряд – подстройка внутреннего сигнала Vboost, младший разряд – сигнал управления блоком мониторинга аналоговых сигналов | RW |
| MinusImix | 7 | 0x0 | Сигнал управления блоком мониторинга аналоговых сигналов | RW |
| DummyCtl1:0 | 9:8 | 0x0 | Подстройка момента захвата выдаваемых данных | RW |
| VbgCtl2:0 | 12:10 | 0x4 | Подстройка опорного напряжения bandgap | RW |
| RefCtl2:0 | 15:13 | 0x4 | Подстройка тока, используемого при чтении | RW |
| VcgCtl2:0 | 18:16 | 0x2 | Подстройка напряжения, подаваемого на управляющие затворы при чтении | RW |
| CGSTRICT | 19 | 0x0 | Включение режима точной подстройки напряжения, используемого при чтении | RW |
| VppCtl1:0 | 21:20 | 0x3 | Подстройка времени нарастания высокого напряжения | RW |
| StrictCtl2:0 | 24:22 | 0x0 | Точная подстройка тока, используемого при чтении | RW |
| StopEe | 25 | 0x0 | Перевод Hard IP в режим пониженного энергопотребления | RW |
| IddqEn | 26 | 0x0 | Перевод Hard IP в состояние бездействия (quiescent) для Iddq тестирования | RW |
| Зарезервировано | 31:27 | | Зарезервировано | |

Таблица 130 – Регистр подстройки длительности процедур чтения и заполнения буфера записи NCYCRL

| Обозначение | Биты | Значение при сбросе | 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 |
| N_R_1 | 15:8 | 0x2 | Количество тактов, в течение которого Phi1ShiftEe имеет высокий уровень в процедуре чтения. Рекомендуемое значение $N_{RA} = \text{ceil}(51/P_{clk})$, где P_{clk} – период тактового сигнала в нс, ceil – функция округления до ближайшего большего целого числа. Запрещено устанавливать равным 0 | RW |
| 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 |
| Зарезервировано | 31:24 | Зарезервировано | | |

Таблица 131 – Регистр 1 подстройки длительности процедур стирания и программирования NCYCEP1

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-------------|------|---------------------|------------|--------|
|-------------|------|---------------------|------------|--------|

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-----------------|-------|---------------------|---|--------|
| 1_PE_N | 17:0 | 66667 | Длительность высокого уровня сигнала BUSY (вход Hard IP) в тактах. Рекомендуемое значение $N_{RA} = 2000000 \text{ нс} / P_{clk}$, где P_{clk} – период тактового сигнала в нс. Пример – 66667 тактов для частоты $F_{clk} = 33,3 \text{ МГц}$. Запрещено устанавливать равным 0 | RW |
| Зарезервировано | 31:18 | Зарезервировано | | |

Таблица 132 – Регистр 2 подстройки длительности процедур стирания и программирования NCUSER2

| Обозначение | Биты | Значение при сбросе | Назначение | Доступ |
|-----------------|-------|---------------------|---|--------|
| N_EP_2 | 10:0 | 500 | Задержка между задними фронтами BUSY и HvonValid (входы Hard IP) в тактах. Рекомендуемое значение $N_{RA} = 15000 \text{ нс} / P_{clk}$, где P_{clk} – период тактового сигнала в нс. Пример – 500 тактов для частоты $F_{clk} = 33,3 \text{ МГц}$. Запрещено устанавливать равным 0 | RW |
| Зарезервировано | 31:11 | Зарезервировано | | |

3.20 Модуль сторожевого таймера

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.

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.

3.20.1 Регистры сторожевого таймера

В таблице 133 перечислены регистры, доступные по интерфейсу APB.

Таблица 133 – Регистры модуля сторожевого таймера

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|-------------------|--------|
| Key | 0x9C | – | Регистр ключа | W |
| Con | 0x84 | 0x00000000 | Конфигурация | RW |
| Sta | 0x9C | – | Регистр состояния | R |

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 134. Обновление регистра возможно только в остановленном состоянии сторожевого таймера.

Таблица 134 – Поля регистра конфигурации сторожевого таймера

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| Preload | 11:00 | 0x000 | Начальное значение таймера при запуске или перезапуске (таймер считает в сторону увеличения значений) | RW |
| Prescale | 14:12 | 0b000 | Делитель входной частоты (Fclk) для таймера: | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| | | | 000 – Fclk / 1; 001 – Fclk / 2; 010 – Fclk / 4; 011 – Fclk / 16; 100 – Fclk / 64; 101 – Fclk / 256; 110 – Fclk / 1024; 111 – Fclk / 4096 | |

Описание полей регистра состояния сторожевого таймера приведено в таблице 135.

Бит активности таймера изменяет свое значение через некоторое время после обращения к регистру Key, необходимое для проведения синхронизации сигналов.

Т а б л и ц а 135 – Поля регистра состояния сторожевого таймера

| Обозначение | Разряды | Назначение | Доступ |
|--------------|---------|--|--------|
| timerEnabled | 0 | Бит активности таймера: 1 – таймер запущен; 0 – таймер остановлен | RW |
| timerLoading | 1 | Бит перезагрузки значения: 1 – выполняется перезагрузка значения в таймере; 0 – перезагрузка значения таймера в данный момент не выполняется (завершена) | RW |
| – | 7:2 | Зарезервировано | |
| wdt_rst_flag | 8 | Флаг генерации сброса сторожевым таймером. Сбрасывается в 0 только при снятии и последующей подаче питания | R |

3.21 Сторожевой таймер шины

Сторожевой таймер шины предотвращает блокировку системы ведомым устройством АНВ. Модуль состоит из регистров управления и трех мониторов шины, подключенных к шинам EEPROM, SPIFI, шине периферийных устройств. Мониторы размещаются между матрицей шины и ведомым устройством. Если есть активная передача на ведомое устройство, и ведомое устройство удерживает

HREADY в значении «0» более определенного количества тактов, 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 возвращает ответ об ошибке и блокирует доступ к ведомому устройству. Монитор прекращает формировать сообщения об ошибках и маскировать доступ к ведомому, когда ведомое завершает передачу, устанавливая HREADY в «1». Если идет пакетная передача, маскируются оставшиеся транзакции в пакетной передаче.

Регистры сторожевого таймера шины представлены в таблице 136.

Т а б л и ц а 136 – Поля регистра состояния сторожевого таймера шины

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|---|--------|
| TIMEOUT | 0x00 | 0x0 | Определяет количество циклов ожидания до формирования ошибки. | RW |
| IRQ_CLEAR | 0x04 | – | Сброс прерываний. | W1C |
| ENABLE | 0x08 | 0x0 | Запуск/отключение мониторов шины. | RW |

3.22 Температурный сенсор

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 32КГц-100КГц

3.22.1 Состав и принцип работы сенсора

Структурная схема представлена на рисунке 76. Сенсор состоит из блока формирования опорного тока, блока формирования двух напряжений база-эмиттер, сигма-дельта АЦП и регистра накопления результатов АЦП.

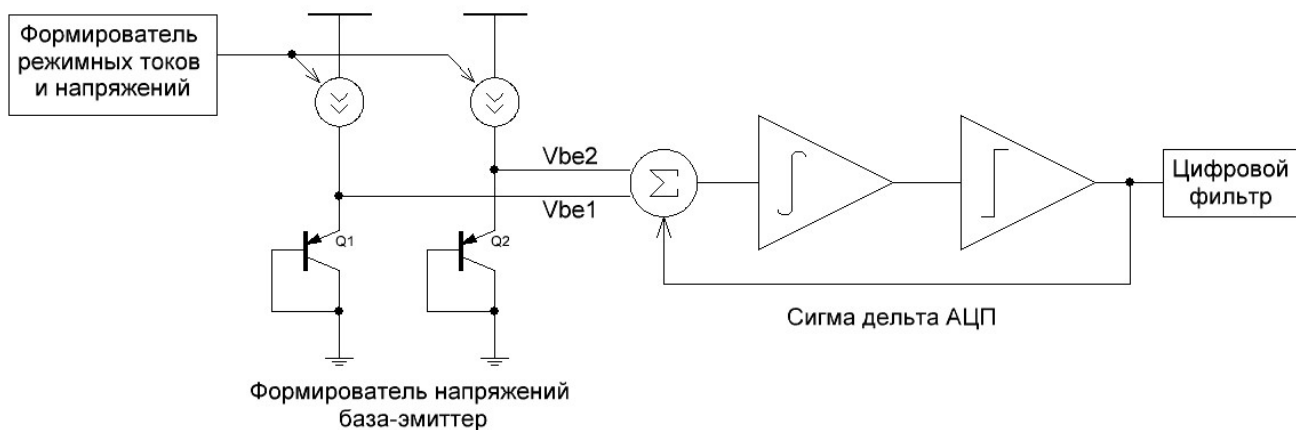


Рисунок 76 – Структурная схема сенсора

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 V_{be} обратно пропорционально температуре (СТАТ), а разница напряжений ΔV_{be} прямо пропорциональна (РТАТ). Эти напряжения линейны, а линейная комбинация создает температурно-независимое постоянное напряжение V_{ref} (рисунок 77, формула 13, 14), соответствующее ширине запрещенной зоны.

$$V_{ref} = V_{be1} + \alpha(V_{be2} - V_{be1}) \quad (13)$$

$$\Delta V_{be} = V_{be2} - V_{be1} \quad (14)$$

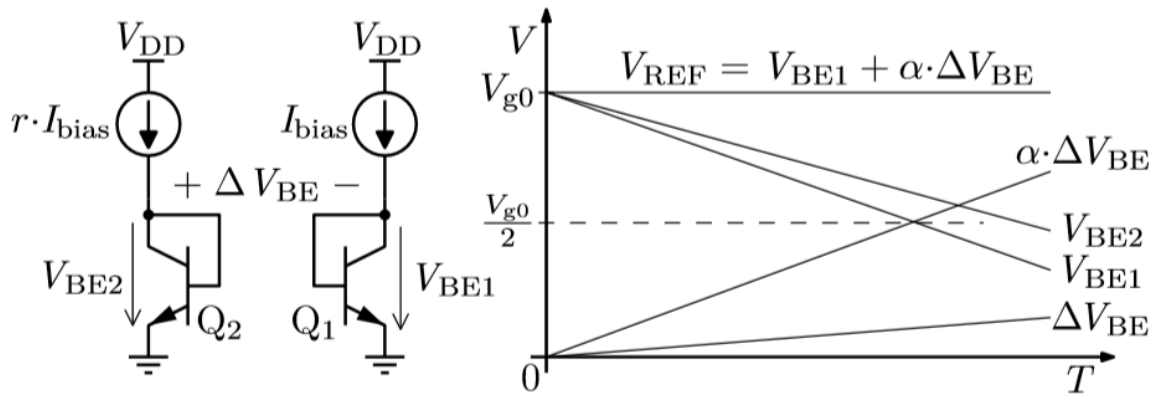


Рисунок 77 – Принцип работы сенсора

Основное преимущество такой схемы, что формируемое РТАТ напряжение не зависимо от технологического процесса и вычисляется по формуле 15

$$\Delta V_{be} = \frac{kT}{q} \ln(r) \quad (15)$$

Эти напряжения переводятся в базис заряда и используются сигма-дельта АЦП на переключающихся конденсаторах. Плотность потока на выходе АЦП меняется от 0 до 1 в зависимости от температуры и равна согласно равенству зарядов по формуле 16. Сама температура вычисляется в градусах Цельсия по формуле 17, при $A=619,2$ и $B=-273,15$ (константы, следующие из формул 13 и 16).

$$\mu = \frac{\alpha \Delta V_{be}}{V_{be} + \alpha \Delta V_{be}} = \frac{V_{ptat}}{V_{ref}} \quad (16)$$

$$T = A\mu + B \quad (17)$$

3.22.2 Регистры управления

Для управления температурным сенсором используются регистры блока управления аналоговой подсистемой. Состав регистров управления сенсором представлен в таблице 137. В таблицах 138–144 представлены описания полей регистров сенсора.

Таблица 137 – Регистры управления температурным сенсором

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|------------|--------|
| | | | | |

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-----------------|----------|--------------------|--|--------|
| TSENS_CFG | 0x1C | 0x0 | Регистр настроек | RW |
| TSENS_TRESHOLD | 0x20 | 0x0 | 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_CONTINIUS | 0x34 | 0x0 | Регистр запуска непрерывного измерения | WO |

Примечание – Смещение относительно стартового адреса блока управления аналоговой подсистемой (0x00085000 в адресном пространстве микроконтроллера).

Таблица 138 – 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 $T_{SENS} = F_{IN} / (Div + 1)$ | RW |
| Clk_mux | 5:3 | 0x0 | Выбор источника тактирования сенсора (F_{IN}): 0x0 – системная частота (sys_clk); 0x1 – частота шины АНВ (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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| Npd | 0 | 0x0 | Управление питанием сенсора: «0» – сенсор выключен; «1» – сенсор включен | RW |

Таблица 139 – Регистр TSENS_TRESHOLD температурного сенсора

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|--|--------|
| Зарезервировано | 31:20 | Зарезервировано | | |
| Low_Threshold | 19:10 | 387 | При значении выхода сенсора менее Low_Threshold будет формироваться прерывание. Значение по умолчанию соответствует температуре 125 °C | RW |
| Hi_Threshold | 9:0 | 658 | При значении выхода сенсора больше Hi_Threshold будет формироваться прерывание. Значение по умолчанию соответствует температуре 125 °C | RW |

Таблица 140 – 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 окончании 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 окончании преобразования | RW |

Таблица 141 – 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 |
| Hi_Clear | 1 | – | Сброс прерывания события «выход сенсора больше порогового значения Hi_Threshold» | W1C |
| Eoc_Clear | 0 | – | Сброс прерывания по окончании преобразования | W1C |

Таблица 142 – 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 143 – Регистр TSENS_SINGLE температурного сенсора

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|-------------------------|--------|
| Зарезервировано | 31:1 | Зарезервировано | | |
| Single | 0 | – | Запуск одного измерения | W1 |

Таблица 144 – Регистр TSENS_CONTINIUS температурного сенсора

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 31:1 | Зарезервировано | | |
| Continius | 0 | 0 | Запуск / остановка непрерывного измерения температуры | RW |

3.23 Цифро-аналоговый преобразователь

Цифро-аналоговый преобразователь (ЦАП) предназначен для формирования напряжения пропорционального 12-битному управляющему коду.

Параметры преобразователя представлены в таблице 145.

Таблица 145 – 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 | U_{REF} | – | 1,2 | – |
| Ток по входу опорного напряжения, мкА | I_{UREF} | – | 0 | – |
| Диапазон изменения напряжения на выводе V_{OUT} (выводы V_{OUT} и V_{FB_N} замкнуты), В | dU_{DAC} | – | $0 - U_{REF}$ | – |
| Сопrotивление нагрузки на выводе V_{OUT} (номинал), кОм | R_{load} | – | 50 | – |
| Емкость нагрузки на выводе V_{OUT} (номинал), пФ | C_{load} | – | 2 | – |
| Время установления выходного напряжения, мкс | $T_{U_{вых}}$ | – | – | 0,7 |
| Внешний опорный ток, мкА | I_{OUT} | 8 | 10 | 12 |
| Ток потребления, мА | I_{CC} | – | 0,9 | 1,07 |

3.23.1 Состав и принцип работы резистивного ЦАП

Модуль состоит из резистивного ЦАП, входного буфера/делителя внешнего опорного напряжения на 2, внутреннего ратиометрического источника опорного напряжения, выходного буфера с коэффициентом усиления два, а также дополнительных резисторов, ключей и цифровой управляющей логики. Схема с делением опорного напряжения пополам и последующего умножения напряжения на 2 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 набросы при

отсутствии буферизации между каскадами резистивной матрицы. Структурная схема представлена на рисунке 78.

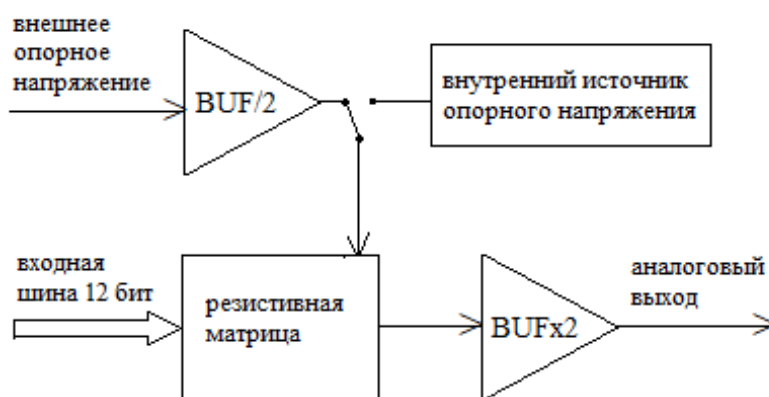


Рисунок 78 – Схема ЦАП

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.

3.23.1 Регистры управления

Для управления ЦАП используются регистры блока управления аналоговой подсистемой. Состав регистров управления ЦАП представлен в таблице 146 .

В таблицах 147, 148 представлены описания полей регистров ЦАП.

Т а б л и ц а 146 – Регистры управления ЦАП

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|------------|--------|
|-------------|----------|--------------------|------------|--------|

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|---|----------|--------------------|---|--------|
| DAC0_CFG | 0x00 | 0x0 | Регистр настройки ЦАП0 | RW |
| DAC1_CFG | 0x04 | 0x0 | Регистр настройки ЦАП1 | RW |
| DAC0_VALUE | 0x08 | 0x0 | Входные данные для ЦАП0 | WO |
| DAC1_VALUE | 0x0C | 0x0 | Входные данные для ЦАП1 | WO |
| REF_CLB | 0x38 | 0x0 | Управление калибруемыми источниками тока и напряжения | RW |
| Примечание – Смещение относительно стартового адреса блока управления аналоговой подсистемой (0x00085000 в адресном пространстве микроконтроллера). | | | | |

Таблица 147 – Регистры DAC0_CFG, DAC1_CFG

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|--|--------|
| Зарезервировано | 31:14 | Зарезервировано | | |
| Empty_Read | 13 | 0x0 | «1» - значение, хранившееся в DAC_VALUE было сдвинуто в ЦАП, возможна запись следующего значения; «0» - в регистре DAC_Value находится необработанное значение; *Автоматически сбрасывается при записи в DAC_Value | RO |
| Reserved | 12 | Зарезервировано | | |
| Ext_Clb | 11 | 0x0 | Выбор источника внешнего опорного напряжения: «1» - внешний вывод; «0» - настраиваемый ОИИ | RW |
| ExtRef | 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 |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| En | 0 | 0x0 | Управление питанием ЦАП: «0» – выключен; «1» – включен | RW |

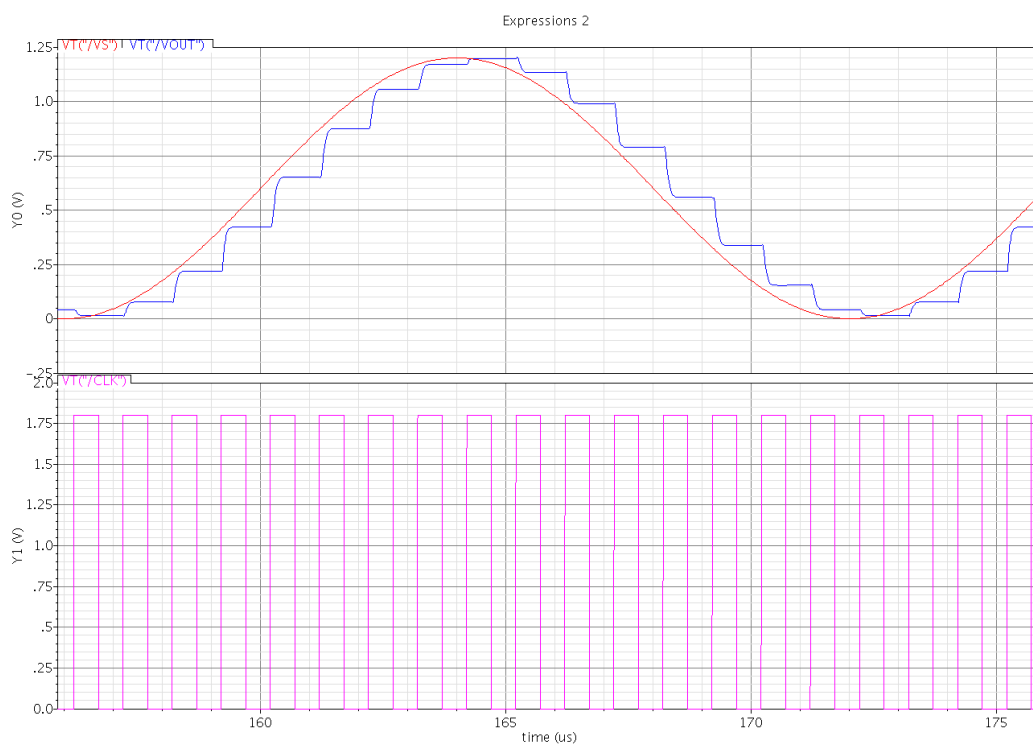
Таблица 148 – Регистры DAC_VALUE0, DAC_VALUE1

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|-----------------------------------|--------|
| Зарезервировано | 31: 12 | Зарезервировано | | |
| Value | 11:0 | 0x0 | Входные данные для преобразования | RW |

Таблица 149 – Регистр управления калибруемыми источниками напряжения и тока

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|--------------|---------|--------------------|---|--------|
| CLB_EN | 8:8 | 0x01 | Разрешение работы калибруемых источников: «1» - включить; «0» - выключить | RW |
| Coef_RefIclb | 7:4 | 0x08 | Коэффициент настройки опорного источника тока | RW |
| Coef_RefVclb | 3:0 | 0x08 | Коэффициент настройки опорного источника напряжения | RW |

3.23.2 Диаграммы выходного сигнала



Выходной сигнал представлен на рисунке 79.

Рисунок 79 – Выходной сигнал ЦАП

3.24 Аналого–цифровой преобразователь

Аналого–цифровой преобразователь (АЦП) предназначен для формирования 12-битного цифрового кода, пропорционального аналоговому входному напряжению.

Параметры АЦП представлены в таблице 150 .

Т а б л и ц а 150 – Параметры АЦП

| Наименование параметра, единица измерения | Буквенное обозначение | Норма параметра | | |
|---|--------------------------|-----------------|---------|----------|
| | | не менее | номинал | не более |
| Число разрядов АЦП | – | – | 12 | – |
| Число кналов мультиплексо- ра | – | – | 8 | – |
| Тактовый сигнал АЦП, МГц | F_{CLK} | 0.16 | 32 | 32 |
| Частота преобразования, КГц | F_S | 4 | | 800 |
| Динамический диапазон, свободный от паразитных со- ставляющих, дБ | S_{FDR} | 64.9 | 71.9 | |
| Эффективное число разрядов АЦП | ENOB | 10.5 | 11.94 | – |

| Наименование параметра, единица измерения | Буквенное обозначение | Норма параметра | | |
|---|--------------------------|-----------------|-------------------|----------|
| | | не менее | номинал | не более |
| 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 |

3.24.1 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 80.

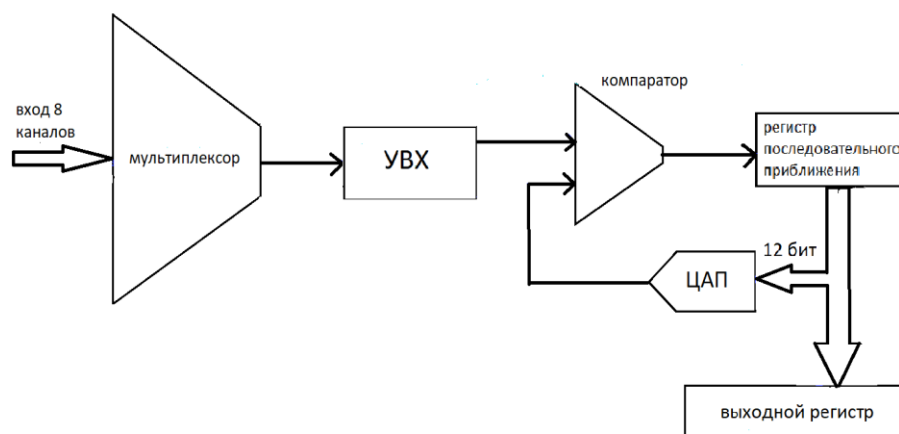


Рисунок 80 – Схема АЦП

АЦП последовательного приближения состоит из резистивного ЦАП, модуля SAR14, SAH (УВХ), компаратора, модуля тестирования выхода ЦАП, а также модулей дерева clock 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. 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 большее сопротивление (300 Ом по сравнению с 75 Ом).

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 $U_{ист} = 1,8$ 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 S/H/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 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 $DIN<0>$.

3.24.2 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, $N'b100...00$, где MSB - лог. «1». При этом выход ЦАП устанавливается в $\frac{1}{2}$ опорного напряжения (VRF). После этого компаратор сравнивает V_{in} (выход УВХ) и VDAC. Если V_{in} больше, чем VDAC, выход компаратора устанавливается в лог. «1», при этом MSB N-разрядного регистра остается в «1»; Если V_{in} меньше, чем VDAC, выход компаратора устанавливается в лог. «0», при этом MSB сбрасывается в «0». Следующий по старшинству разряд устанавливается в «1» и процесс сравне-

ния 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 81. Вместе со сбросом/установкой и циклом записи данных в УВХ преобразование занимает 40 тактов сигнала CLK (12 тактов записи данных в УВХ и 28 тактов собственно преобразование). Поэтому для обеспечения частоты сэмплирования 0.8МГц требуется входной тактовый сигнал частотой не менее 32МГц. 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.

В общем случае сигнал EOC формируется по переднему фронту CLK с задержкой порядка 8ns и имеет длительность порядка $2/F_{clk}$. Активный уровень сигнала EOC на выходе блока всегда заканчивается позже переднего фронта сигнала CLK, при этом данное соотношение не гарантируется вне блока.

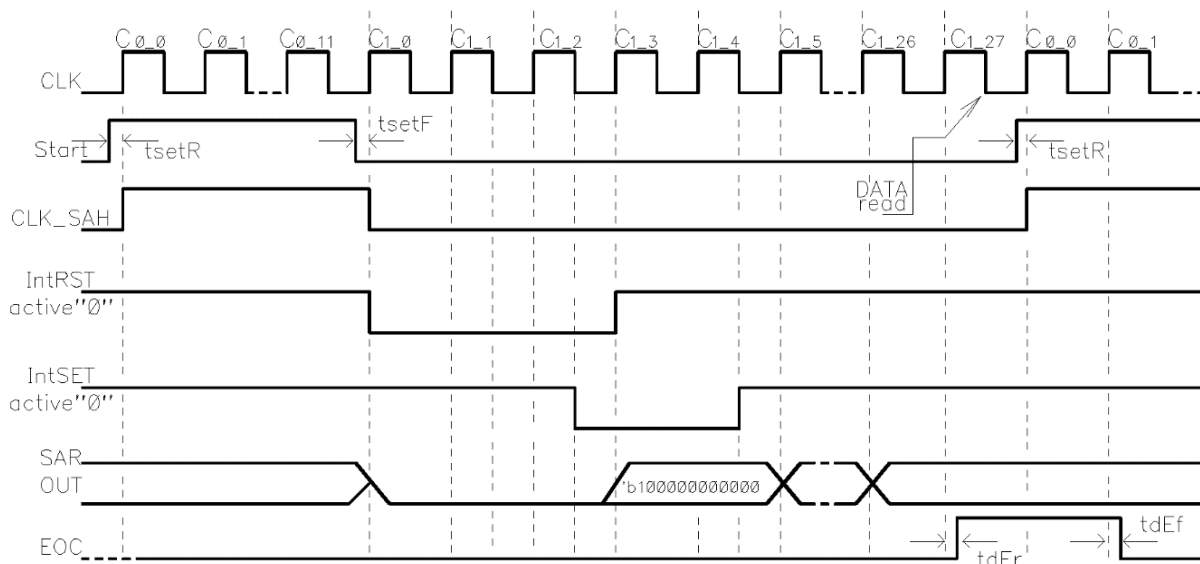


Рисунок 81 – Временная диаграмма цикла преобразования

Для управления АЦП используются регистры блока управления аналоговой подсистемой. Состав регистров управления АЦП представлен в таблице 149.

В таблицах 152 – 156 представлены описания полей регистров АЦП.

Таблица 151 – 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 | 0x01 | Управление калибруемыми источниками тока и напряжения | RW |

Примечание – Смещение относительно стартового адреса блока управления аналоговой подсистемой (0x00085000 в адресном пространстве микроконтроллера).

Таблица 152 – Регистр ADC_CONFIG АЦП

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|-----------------|--------|
| Зарезервировано | 31:9 | | Зарезервировано | |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|--|--------|
| SAH_Time | 13:8 | 0xС | Время выборки очередного отсчета в тактах АЦП | RW |
| Зарезервировано | 7 | Зарезервировано | | |
| Sel | 6:4 | 0x0 | Выбор канала АЦП | RW |
| Ext_Clb | 3 | 0x0 | Выбор источника внешнего опорного напряжения: «1» – внешний вывод; «0» – настраиваемый ОИН | RW |
| ExtRef | 2 | 0x0 | Выбор источника опорного напряжения: «1» – внешний; «0» – встроенный | RW |
| Rn | 1 | 0x0 | Управление сбросом АЦП, активный уровень «0» | RW |
| En | 0 | 0x0 | Управление питанием АЦП: «0» – выключен; «1» – включен | RW |

Таблица 153 – Регистр ADC_CONTINUOUS АЦП

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 31: 1 | Зарезервировано | | |
| Continuous | 0 | 0x0 | Запуск / остановка непрерывного измерения | RW |

Таблица 154 – Регистр ADC_SINGLE АЦП

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|-------------------------------|--------|
| Зарезервировано | 31: 1 | Зарезервировано | | |
| Single | 0 | 0x0 | Запуск однократного измерения | WO |

Таблица 155 – Регистр ADC_VALID АЦП

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|------------|--------|
| Зарезервировано | 31: 1 | Зарезервировано | | |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|-----------------------------------|--------|
| Valid | 0 | 0x0 | Признак наличия актуальных данных | RO |

Таблица 156 – Регистр ADC_VALUE АЦП

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|--------------------------|--------|
| Зарезервировано | 31: 12 | Зарезервировано | | |
| Value | 11:0 | 0x0 | Результат преобразования | RO |

Таблица 157 – Регистр управления калибруемыми источниками напряжения и тока

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|--------------|---------|--------------------|---|--------|
| CLB_EN | 8 | 0x01 | Разрешение работы калибруемых источников: «1» - включить; «0» - выключить | RW |
| Coef_RefIclb | 7:4 | 0x08 | Коэффициент настройки опорного источника тока | RW |
| Coef_RefVclb | 3:0 | 0x08 | Коэффициент настройки опорного источника напряжения | RW |

3.24.3 Диаграммы выходного сигнала

На рисунке 82 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.

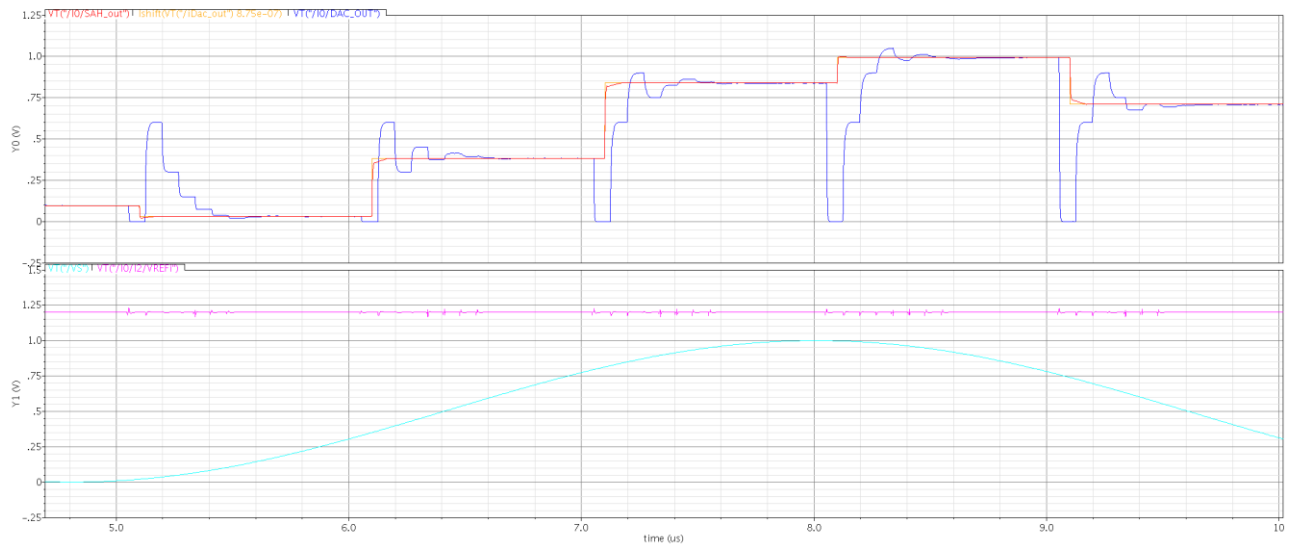


Рисунок 82 – Временные диаграммы работы АЦП

4 Подсистема тактирования

Микросхема MCU32 имеет гибкую систему управления тактовыми сигналами. Общая схема тактирования представлена на рисунке 83.

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 АНВ;
- возможность задания делителей периферийных шин APB_M/APB_P,
- наличие монитора источников тактового сигнала.

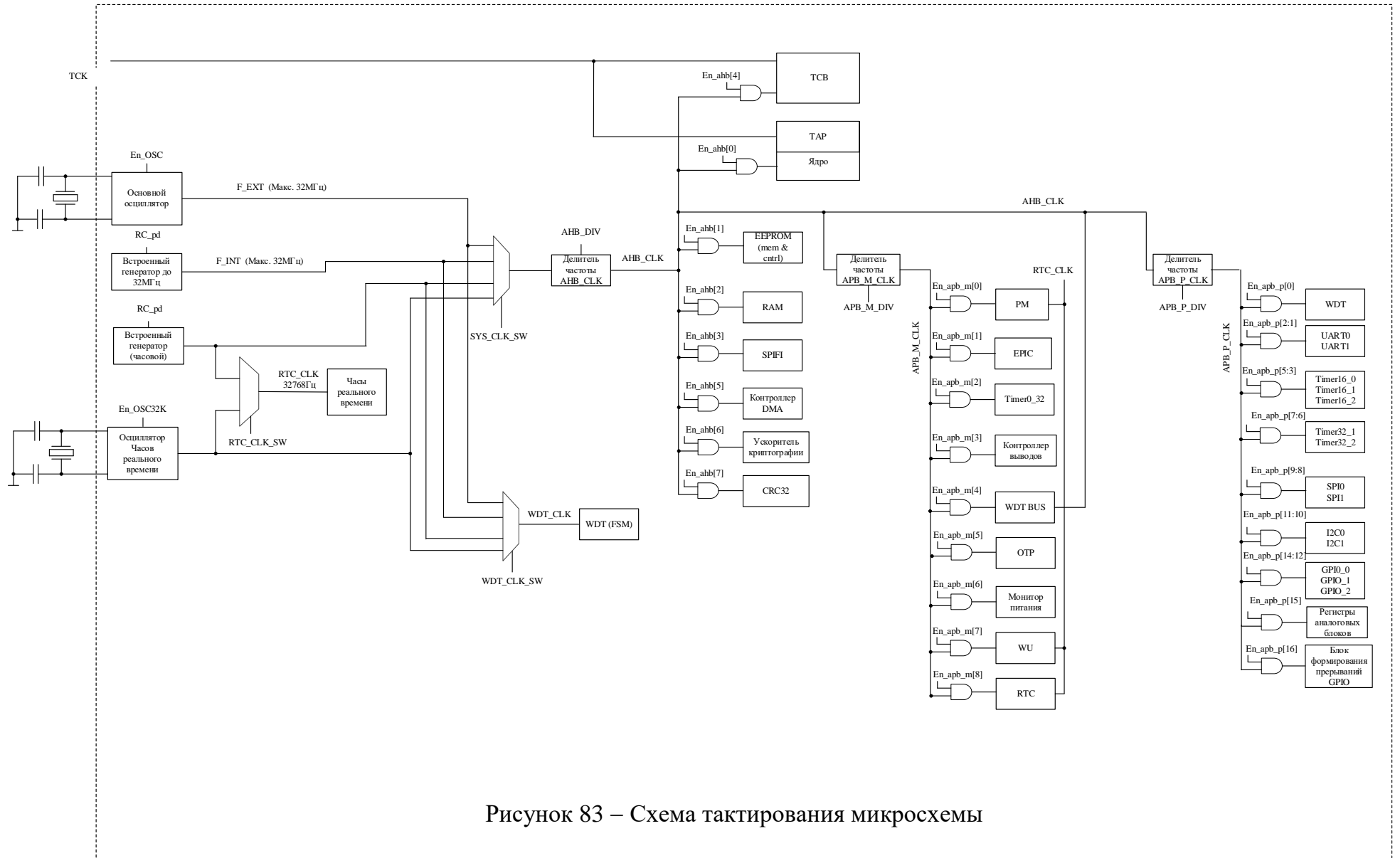


Рисунок 83 – Схема тактирования микросхемы

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 регистра модуля WakeUp.

При включении питания и по внешнему сбросу:

- тактирование осуществляется от встроенного осциллятора 32 МГц (HSI32M);
- значения делителей шин AHB, APB_M, APB_P равны 0, то есть указанные шины работают на частоте осциллятора;
- тактирование RAM, EEPROM, SPIFI, PowerManager, WakeUp включено, тактирование остальных устройств выключено;
- тактирование сторожевого таймера осуществляется от встроенного осциллятора 32 МГц;
- внешние осцилляторы OSC32K и OSC32M включены.

Перед использованием внешнего осциллятора его необходимо включить записью «0» в поле OCS32M_en регистра Generator блока WakeUp. При необходимости, для снижения потребления встроенный осциллятор 32 МГц выключен путем записи «1» в поле RC32M_en регистра Generator блока PM.

Для подстройки встроенного осциллятора HSI32M используются поправочные коэффициент с допустимым значением от 0 до 63.

Для подстройки встроенного осциллятора LSI32K используются поправочные коэффициент с допустимым значением от 0 до 15.

Описание полей регистров модуля WU для управления тактированием представлено в таблице 158. Описание полей регистров модуля PM для управления тактированием представлено в таблице 159.

Таблица 158 – Регистры и поля управления тактированием модуля WU

| Регистр | Смещение | Обозначение | Номер разряда | Описание |
|------------|----------|-------------|---------------|---|
| Clocks_SYS | 0x14 | OCS32M_en | [0] | Включение/отключение внешнего осциллятора |

| Регистр | Смещение | Обозначение | Номер разряда | Описание |
|-----------|----------|---------------|---------------|--|
| | | RC32M_en | [1] | Включение/отключение HSI32M |
| | | Adj_RC32M | [9:2] | Поправочные коэффициенты HSI32M |
| | | Force_32K_Clk | [11:10] | Принудительное переключение на опорный источник для монитора частоты 2'b01 – принудительно выбран OSC32K 2'b10 – принудительно выбран LSI32K |
| Clocks_BU | 0x10 | OCS32K_en | [0] | Включение/отключение внешнего осциллятора на 32 КГц |
| | | RC32K_en | [1] | Включение/отключение LSI32K |
| | | Adj_RC32K | [9:6] | Поправочные коэффициенты LSI32K |
| | | RTC_Clk_Mux | [10] | Выбор источника тактирования часов реального времени: 0 – внутренний LSI32K; 1 – внешний осциллятор OSC32K |
| | | OSC32K_sm | [14] | Режим повышенного потребления, активный уровень “0” для OSC32K |

Таблица 159 – Регистры управления тактированием модуля PM

| Регистр | Смещение | Обозначение | Номер разряда | Описание |
|-----------|----------|-------------|---------------|--|
| Div_AHB | 0x00 | Div_AHB | [7:0] | Делитель частоты AHB |
| Div_APB_M | 0x04 | Div_APB_M | [7:0] | Делитель частоты APB_M |
| Div_APB_P | 0x08 | Div_APB_P | [7:0] | Делитель частоты APB_P |
| AHB_Mux | 0x24 | AHB_Clk_Mux | [1:0] | Выбор источника тактирования системы: 0 – внешний OSC32M; 1 – внутренний OSC32M; 2 – внешний OSC32K; 3 – внутренний OSC32K |
| | | Force_Mux | [3] | Запрет на принудительное переключение с выбраного источника тактирования при пропадании тактирования |

| Регистр | Смещение | Обозначение | Номер разряда | Описание |
|--------------------------------|--------------|--------------------------------|---------------|---|
| WDT_Clk_Mux | 0x28 | WDT_Clk_Mux | [1:0] | Выбор источника тактирования сторожевого таймера: 0 – внешний OSC32M; 1 – внутренний HSI32M; 2 – внешний OSC32K; 3 – внутренний LSI32K; |
| Clk_AHB_Set Clk_AHB_Clear | 0x0C 0x10 | Clk_AHB_Set Clk_AHB_Clear | [7:0] | Управление тактированием устройств на шине АHB. Запись «1» в Clk_AHB_Set включает тактирования, запись «1» в Clk_AHB_Clear отключает тактирование. Каждому биту соответствует одно устройство: 0 – Ядро; 1 – RAM; 2 – EEPROM; 3 – SPIFI; 4 – DMA; 5 – TCB; 6 – ускоритель криптографии; 7 – CRC32 |
| Clk_APBm_Set Clk_APBm_Clear | 0x14 0x18 | Clk_APBm_Set Clk_APBm_Clear | [8:0] | Управление тактированием устройств на шине APB_M. Запись «1» в Clk_APBm_Set включает тактирование, запись «1» в Clk_APBm_Clear отключает тактирование. Каждому биту соответствует одно устройство: 0 – Блок управления питанием (PM - Power Monitor); 1 – контроллер прерываний; 2 – таймер32_0; 3 – контроллер выводов; 4 – Сторожевой таймер шины 5 – ОTR; 6 – монитор питания системного домена 7 – блок WU батарейного домена 8 – RTC батарейного домена |

| Регистр | Смещение | Обозначение | Номер разряда | Описание |
|----------------|----------|----------------|---------------|--|
| Clk_APBp_Set | 0x1C | Clk_APBp_Set | [16:0] | Управление тактированием устройств на шине APB_P. Запись «1» в Clk_APBp_Set включает тактирование, запись «1» в Clk_APBp_Clear отключает тактирование. 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 |
| Clk_APBp_Clear | 0x20 | Clk_APBp_Clear | | |

Частоты сигналов рассчитываются по формулам (20 – 22):

$$F_{\text{АНВ}} = F_{\text{ИСТОЧНИКА}} / (\text{Div_АНВ} + 1), \quad (20)$$

$$F_{\text{APB_M}} = F_{\text{АНВ}} / (\text{Div_APB_M} + 1), \quad (21)$$

$$F_{\text{APB_P}} = F_{\text{АНВ}} / (\text{Div_APB_P} + 1). \quad (22)$$

4.2 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 84). Опорный потенциал на компараторы подается с ЦАП, что позволяет отрегулировать появившиеся вследствие разброса техпроцесса смещения.

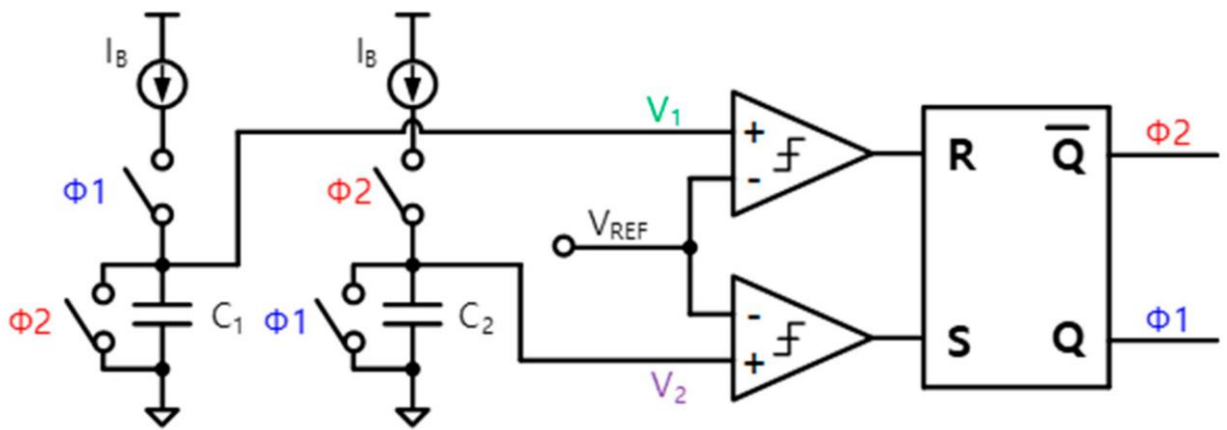


Рисунок 84 – Схема осциллятора

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 $\pm 25\%$.

4.3 Вывод для подключения внешнего кварцевого резонатора до 32 МГц

Осциллятор с частотой 32 МГц построен на схеме Пирса (рисунок 85).

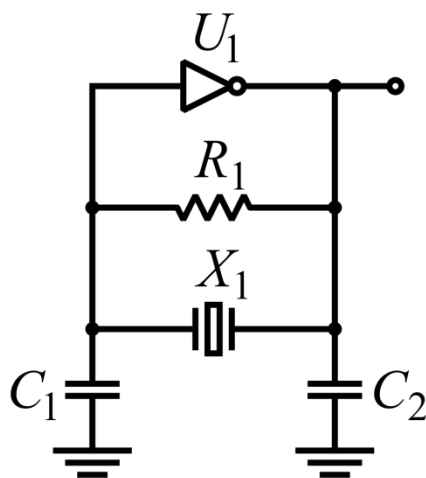


Рисунок 85 – Осциллятор Пирса

КМОП схема генератора размещается в IО ячейке. В нее входят инвертирующий усилитель, источник опорных потенциалов для питания усилителя, резистор обратной связи 500 кОм R_{O0} , ESD защита контактной площадки, драйвер для преобразования гармонического сигнала генератора в выходной импульсный.

4.4 Вывод для подключения внешнего кварцевого резонатора 32 кГц

Осциллятор построен на схеме Пирса. КМОП схема генератора размещается в IО ячейке. В нее входят инвертирующий усилитель, источник опорных потенциалов IО 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.

4.5 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.

4.5.1 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 взаимноисключающим 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 `s_32k_ext_det` и `s_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.

4.5.2 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 160. Резуль-

таты автоматической детекции могут быть использованы для выбора синхросигналов и формирования прерываний.

Таблица 160 – 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 86 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.

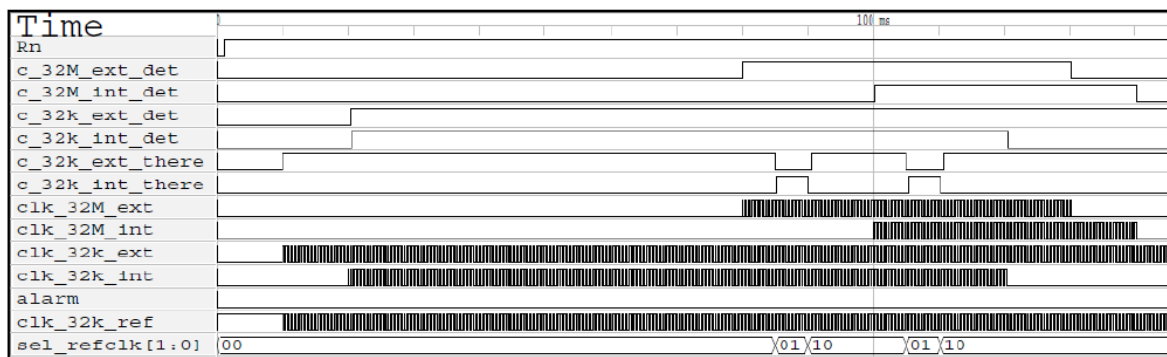


Рисунок 86 – Стартовый и динамический режимы

4.5.3 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 87 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 88 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.

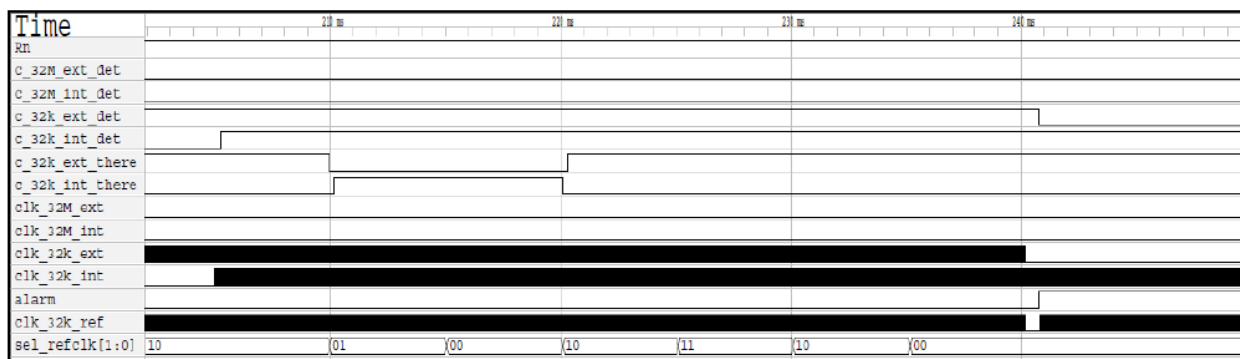


Рисунок 87 – Динамический и аварийный режимы

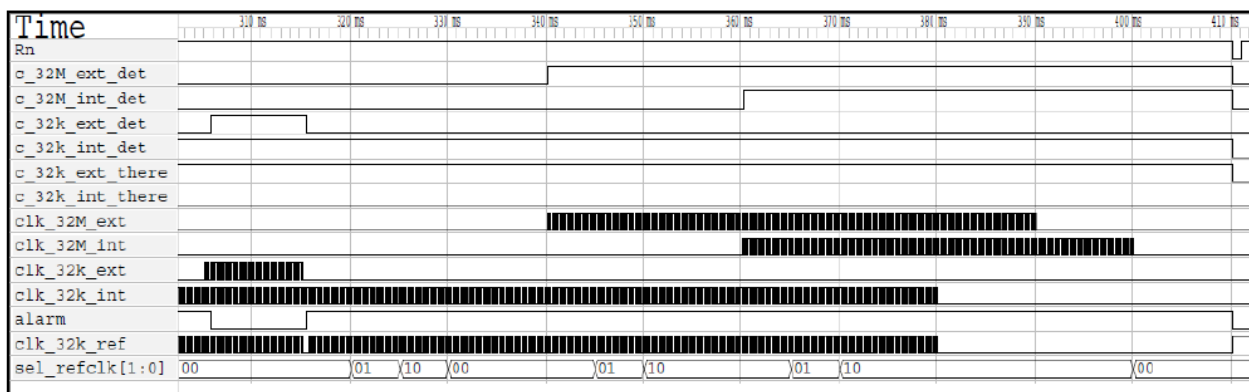


Рисунок 88 – Аварийный режим

Управление монитором частоты системного реализуется через следующие регистры блока управления питанием и тактирования (таблица 161) и регистры блока WakeUp (таблица 162)

Таблица 161 – Регистры блока управления питанием и тактированием для управления монитором частоты

| Обозначение | Смещение | Поля | Назначение |
|-------------|----------|-------|--|
| AHB_Clk_Mux | 0x24 | [1:0] | Выбор источника тактирования системы: 0 – внешний OSC32M; 1 – внутренний HSI32M; 2 – внешний OSC32K; 3 – внутренний LSI32K |
| Freq_Mask | 0x34 | [4:0] | Настройки монитора частоты: Freq_Mask [0] – маска прерывания при пропадании частоты внешнего осцилятора 32 МГц; Freq_Mask [1] – маска прерывания при пропадании частоты HSI32M; Freq_Mask [2] – маска прерывания при пропадании частоты внешнего осцилятора 32 кГц; Freq_Mask [3] – маска прерывания при пропадании частоты LSI32K; Freq_Mask [4] – разрешение установки дилителя шины AHB в 0, при пропадании внешней и внутренней частоты 32МГц |
| Freq_Status | 0x38 | [3:0] | Статус монитора частоты (0 – источник тактирования не детектируется, 1 – тактовый источник детектируется) Freq_Status [0] – внешний OSC32M; Freq_Status [1] – внутренний HSI32M; Freq_Status [2] – внешний OSC32K; Freq_Status [3] – внутренний LSI32K |
| Force_Mux | 0x24 | [2] | Запрет на принудительное переключение с выбраного источника тактирования при пропадании тактирования |

Таблица 162 – Регистры блока Wakeup

| Обозначение | Биты | Назначение |
|-------------|------|------------|
|-------------|------|------------|

| Обозначение | Биты | Назначение |
|-------------|---------|---|
| Clocks_SYS | [11:10] | Принудительное переключение на опорный источник 2'b01 – принудительно выбран OSC32K 2'b10 – принудительно выбран LSI32K |

5 Подсистема питания

В микросхеме MCU32 реализовано независимое управление питанием и тактированием системы в целом и отдельными устройствами.

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 шифрования, блок вычисления контрольных сумм, регистр управления аналоговыми блоками, HSI32M, OSC32M

Распределение блоков по доменам питания микросхемы представлены на рисунке 89.

Основные режимы работы микроконтроллера представлены в таблице 163. В режимах «Активный», «Спящий», «Пониженного потребления» возможно программное снижение частоты шин APB_M, APB_P, а также отключения тактирования отдельных устройств.

Управление питанием осуществляется модулями WakeUP, PowerManager, управление тактированием модулем PowerManager.

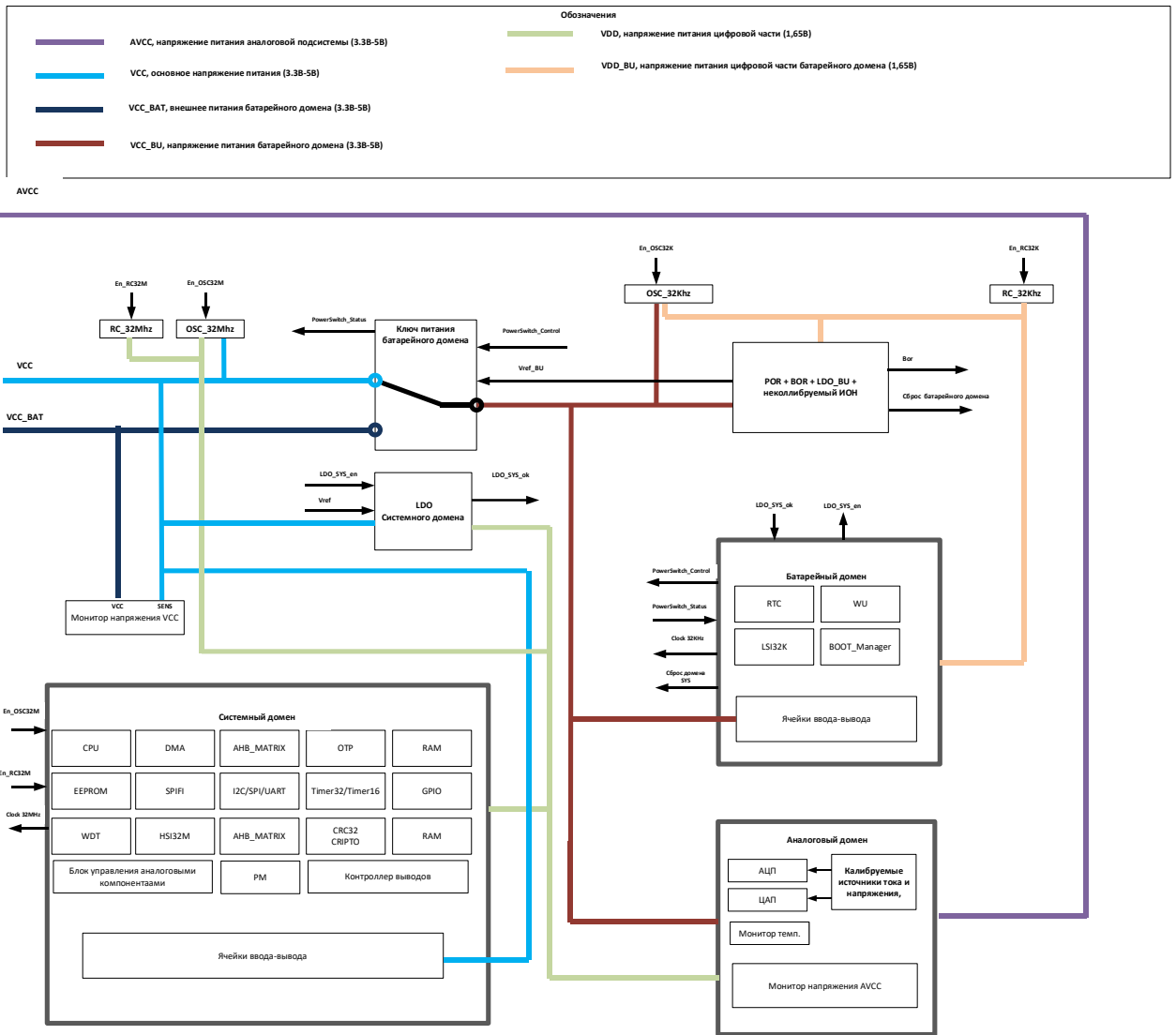


Рисунок 89 – Домены питания

Таблица 163 – Основные режимы работы микроконтролера

| 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 осцилятора с частотой до 32 МГц От встроенного осцилятора с частотой до 32 МГц | Режим после сброса/подачи питания | – |
| Пониженного энергопотребления | Процессор работает в нормальном режиме. | ОЗУ, EEPROM, SPIFI Работают в нормальном режиме. Тактируются только выбранные блоки памяти | Периферия работает в нормальном режиме. Выключено тактирование не используемых блоков | Системный домен включен, батарейный домен включен | От внешнего часового осцилятора от встроенного часового осцилятора От внешнего осцилятора с частотой до 32 МГц с делителем системной шины 1024 | Программно, переключившись на нужный источник тактирования | Программно, переключившись на нужный источник тактирования |
| Спящий | Ядро в состоянии sleep | Тактирование ОЗУ, EEPROM, SPIFI выключено | Тактированием периферия выключено, за исключением блоков, которые могут формировать прерывание | Системный домен включен, батарейный домен включен | От внешнего осцилятора с частотой до 32 МГц От встроенного осцилятора с частотой до 32 МГц | Запись в регистр PowerManager | По прерыванию/сбросу/RTC/внешнему сигналу |
| 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 | По сбросу/RTC /внешнему выводу |
| Ожидание | Питание выключено | Питание выключено | Питание выключено | Системный домен выключен, батарейный домен включен | Тактирование системы выключено | Запись в регистр WakeUp | По сбросу/RTC /внешнему выводу |

5.2 Блок управления питанием и тактированием

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 164.

Т а б л и ц а 164 – Регистры блока управления питанием и тактированием

| Обозначение | Смещение | Значение по умолчанию | Назначение |
|-------------|----------|-----------------------|---|
| Div_AHB | 0x00 | 0x00 | Задаёт значение делителя шины АHB. Частота шины АHB (F_{AHB}) рассчитывается, как $F_{\text{SYS}} / (\text{Div_AHB} + 1)$ |

| Обозначение | Смещение | Значение по умолчанию | Назначение |
|-------------|----------|-----------------------|---|
| Div_APB_M | 0x04 | 0x00 | <p>Задаёт значение делителя шины APB_M. Частота шины APB_M (F_{APB_M}) рассчитывается, как $F_{APB}/(Div_APB_M+1)$</p> |

| Обозначение | Смещение | Значение по умолчанию | Назначение |
|-------------|----------|-----------------------|---|
| Div_APB_P | 0x08 | 0x00 | <p>Задаёт значение делителя шины APB_P. Частота шины APB_P (F_{APB_P}) рассчитывается, как $F_{APB}/(Div_APB_P+1)$</p> |

| Обозначение | Смещение | Значение по умолчанию | Назначение |
|---------------|----------|-----------------------|--|
| Clk_AHB_Set | 0x0C | 0x1F | <p>Регистр включения тактированием устройств на шине АHB</p> <p>Каждому биту соответствует устройство:</p> <ul style="list-style-type: none"> 0 – Ядро; 1 – EEPROM; 2 – RAM; 3 – SPIFI; 4 – TCB; 5 – DMA; 6 – ускоритель криптографии; 7 – CRC32 |
| Clk_AHB_Clear | 0x10 | 0x00 | <p>Регистр выключения тактированием устройств на шине АHB. Каждому биту соответствует устройство, аналогично Clk_AHB_Set</p> |

| Обозначение | Смещение | Значение по умолчанию | Назначение |
|-----------------|----------|-----------------------|--|
| Clk_APB_M_Set | 0x14 | 0x89 | Регистр включения тактированием 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 – ОТП; 6 – монитор питания системного домена 7 – блок WU батарейного домена 8 – RTC батарейного домена |
| Clk_APB_M_Clear | 0x18 | 0x00 | Регистр выключения тактированием устройств на шине APB_M. Каждому биту соответствует устройство, аналогично Clk_APB_M_Set |
| 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 |
| Clk_APB_P_Clear | 0x20 | 0x00 | Регистр выключения тактированием устройств на шине APB_P. Каждому биту соответствует устройство, аналогично Clk_APB_P_Set |

| Обозначение | Смещение | Значение по умолчанию | Назначение |
|-----------------|----------|-----------------------|---|
| AHB_Mux | 0x24 | 0x00 | Настройка источника тактирования системы (таблица 165) |
| Wdt_Clk_Mux | 0x28 | 0x00 | Выбор источника тактирования сторожевого таймера: 0 – внешний OSC32M; 1 – внутренний HSI32M; 2 – внешний OSC32K; 3 – внутренний LSI32K; |
| Cpu_RTC_Clk_Mux | 0x2C | 0x00 | Выбор источника тактирования RTC в состоянии ядра 0 – внешний OSC32K; 1 – внутренний LSI32K; |
| Timer_CFG | 0x30 | 0x00 | Выбор источника тактирования для таймеров (таблица 166) |
| Freq_Mask | 0x34 | 0x10 | Настройки прерываний монитора частоты (таблица 167) |
| Freq_Status | 0x38 | 0x00 | Статус монитора частоты (таблица 168) |
| Sleep_Mode | 0x3C | 0x00 | Переход в спящий режим осуществляется записью “1” |

Таблица 165 – Регистр настройки источника тактирования системы AHB_Mux

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| AHB_Clk Mux | [1:0] | 0x00 | Источника тактирования системы: 0 – внешний OSC32M; 1 – внутренний OSC32M; 2 – внешний OSC32K; 3 – внутренний OSC32K | RW |
| Force_Mux | [2:2] | 0x00 | Запрет на принудительное переключение с выбранного источника тактирования при пропадании тактирования 0 – разрешение автоматической смены частоты; 1 – запрет автоматической смены частоты; | RW |

Таблица 166 – Регистр настройки тактирования для таймеров

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|--|--------|
| Mux_Tim32_0 | [2:0] | 0x00 | Выбор тактового сигнала для Timer32_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 |
| Mux_Tim32_1 | [5:3] | 0x00 | Выбор тактового сигнала для Timer32_1. Значение, аналогичны Mux_Tim32_0 | RW |
| Mux_Tim32_2 | [8:6] | 0x00 | Выбор тактового сигнала для Timer32_2. Значение, аналогичны Mux_Tim32_0 | RW |
| Mux_Tim16_0 | [11:9] | 0x00 | Выбор тактового сигнала для Timer16_1. Значение, аналогичны Mux_Tim32_0 | RW |
| Mux_Tim16_1 | [14:12] | 0x00 | Выбор тактового сигнала для Timer16_2. Значение, аналогичны Mux_Tim32_0 | RW |
| Mux_Tim16_2 | [117:15] | 0x00 | Выбор тактового сигнала для Timer16_3. Значение, аналогичны Mux_Tim32_0 | RW |

Таблица 167 – Регистр настройки масок прерываний монитора частоты Freq_Mask

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| Mask_osc32k | [0] | 0x00 | Разрешение прерывания при пропадании частоты внешнего осциллятора 32 МГц | RW |
| Mask_hsi32m | [1] | 0x00 | Разрешение прерывания при пропадании частоты HSI32M | RW |
| Mask_osc32m | [2] | 0x00 | Разрешение прерывания при пропадании частоты внешнего осциллятора 32 кГц | RW |
| Mask_lsi32k | [3] | 0x00 | Разрешение прерывания при пропадании частоты LSI32K | RW |

Таблица 168 – Регистр статуса монитора частоты Freq_Status

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| Mask_osc32k | [0] | 0x00 | Статус частоты внешнего осциллятора 32 МГц | RO |
| Mask_hsi32m | [1] | 0x00 | Статус частоты HSI32M | RO |
| Mask_osc32m | [2] | 0x00 | Статус частоты внешнего осциллятора 32 кГц | RO |
| Mask_lsi32k | [3] | 0x00 | Статус частоты LSI32K | RO |

5.3 Блок WakeUp

Блок WakeUp обеспечивает управление питанием системным доменом, управление источниками тактовых сигналов. Регистры блока WakeUp представлены в таблице 169.

Таблица 169 – Регистры блока WakeUp

| Обозначение | Смещение | Значение по умолчанию | Назначение |
|--------------|----------|-----------------------|--|
| Sys_Mask | 0x00 | 0x10 | Маски событий для включения и сброса системного домена (таблица 170) |
| Sys_Level | 0x04 | 0x03 | Активные уровни событий для включения и сброса системного домена (таблица 171) |
| Sys_PowerOff | 0x08 | 0x00 | Запись в регистр «1» отключает питание системного домена |
| PowerSwitch | 0x0C | 0x00 | Регистр управления и статуса схемы слежения за питанием (таблица 172) |
| Clocks_BU | 0x10 | 0x200 | Регистр управления тактированием батарейного домена |
| Clocks_SYS | 0x14 | 0x200 | Регистр управления тактированием системного домена |
| RTC_Control | 0x18 | 0x00 | Сброс RTC происходит при записи «1» |
| Stop | 0x1C | 0x00 | Переход в режим СТОП осуществляется записью «1» |
| CPU_Dly | 0x20 | 0x02 | Дополнительная задержка до запуска ядра (в тактах 32КГц) |

Таблица 170 – Регистр масок событий для включения и сброса системного домена Sys_Mask

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| Sys_Up_RTC | [0] | 0x00 | Разрешение включение системного домена (из режима СТОП) при срабатывании будильника | RW |
| Sys_Up_WU | [1] | 0x00 | Разрешение включение системного домена (из режима СТОП) при активном уровне внешнего вывода ext_wu | RW |
| Sys_Rst_Ldo | [2] | 0x01 | Разрешение формирования сброса системного домена при снижении питания (индикация от LDO_SYS) | RW |
| Sys_Rst_PS | [3] | 0x00 | Разрешение формирования сброса системного домена при снижении питания (индикация от схемы слежения) | RW |
| Sys_Rst_Bor | [4] | 0x00 | Разрешение формирования сброса системного домена при появлении сигнала BOR | RW |
| Bu_Rst_Bor | [5] | 0x00 | Разрешение формирования сброса батарейного домена при появлении сигнала BOR | RW |

Таблица 171 – Регистр уровней событий для включения и сброса системного домена Sys_Level

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|--|--------|
| Lvl_RTC | [0] | 0x01 | Активный уровень срабатывания будильника | RW |
| Lvl_WU | [1] | 0x01 | Активный уровень внешнего вывода ext_wu | RW |

Таблица 172 – Регистр управления и статуса схемы слежения за питанием PowerSwitch

| 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 принудительного переключения (при En = "1") 0 – VCC 1 – VCC_BU | RW |
| En | 0 | 0x0 | Разрешение принудительного переключения на один из источников питания | RW |

5.4 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 до 3,696 В в диапазоне температур от -40 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 тестируемых 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 overTh и underTh позволяет не только проверить работу компаратора, но и оценить значение смещения на его входах.

Вычисление пороговых значений монитора выполняются по формулам

$$\text{overTh} = v_{\text{sup}} \cdot 1,04 + v_{\text{step}} * \text{overTh} ,$$

$$\text{underTh} = v_{\text{sup}} \cdot 0,96 - v_{\text{step}} * \text{underThD} ,$$

где vstep – Шаг изменения порогового значения;

v_{sup} – Напряжение питание VCC (3,3 В).

Расчитанные значения для напряжения питания 3,3 В представлены в таблице 173.

Т а б л и ц а 173 – Выбор значений срабатывания монитора питания

| Код overTh/underTh | Уровень Out_under | Уровень Out_over |
|-----------------------|-------------------|---------------------|
| 0000 | 3,168 | 3,432 |
| 0001 | 3,1504 | 3,4496 |
| 0010 | 3,1328 | 3,4672 |
| 0011 | 3,1152 | 3,4848 |
| 0100 | 3,0976 | 3,5024 |
| 0101 | 3,08 | 3,52 |
| 0110 | 3,0624 | 3,5376 |
| 0111 | 3,0448 | 3,5552 |

5.4.1 Регистры управления

Для управления монитором напряжения AVCC используются регистры блока управления аналоговой подсистемой. Состав регистров управления монитором напряжения AVCC представлен в таблице 174.

Для управления монитором напряжения VCC используются регистры PVD_CONTROL. Состав регистров управления монитором напряжения VCC представлен в таблице 175.

В таблицах 176 – 178 представлены описания полей регистров мониторов.

Таблица 174 – Регистры управления монитора напряжения AVCC

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|-----------------------------|--------|
| DPF | 0x10 | 1500 | Настройка цифрового фильтра | RW |
| PVD_CFG | 0x14 | 0x0 | Регистр настроек | RW |
| PVD_STATUS | 0x1C | 0x0 | Регистр статуса | RO |

Примечание – Смещение относительно стартового адреса блока управления аналоговой подсистемой (0x00085000 в адресном пространстве микроконтроллера).

Таблица 175 – Регистры управления монитора напряжения VCC

| Обозначение | Смещение | Начальное значение | Назначение | Доступ |
|-------------|----------|--------------------|-----------------------------|--------|
| PVD_DPF | 0x0 | 1500 | Настройка цифрового фильтра | RW |
| PVD_CFG | 0x04 | 0x0 | Регистр настроек | RW |
| PVD_STATUS | 0x08 | 0x0 | Регистр статуса | RO |

Примечание – Смещение относительно стартового адреса PVD_CONTROL (0x000051800 в адресном пространстве микроконтроллера).

Таблица 176 – Регистр PVD_DPF монитора напряжения

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 31:16 | Зарезервировано | | |
| Dpf | 15:0 | 1500 | Коэффициент, определяющий длительность отфильтровываемых импульсов (на системной частоте) | RW |

Таблица 177 – Регистр PVD_CFG монитора напряжения

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 31:13 | Зарезервировано | | |
| OverTh | 12 | 0x0 | Использование в качестве опорного источника напряжения калибруемый ОИН | RW |
| OverTh | 11:8 | 0x0 | Верхний порог срабатывания монитора | RW |
| UnderTh | 7:4 | 0x0 | Нижний порог срабатывания монитора | RW |
| Test | 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 178 – Регистр PVD_STATUS монитора напряжения

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|---|--------|
| Зарезервировано | 31:4 | Зарезервировано | | |
| Out_over_Flag | 3 | 0x0 | Текущее состояние линии монитора верхнего порога с учетом цифровой фильтрации | RO |
| Out_under_Flag | 2 | 0x0 | Текущее состояние линии монитора нижнего порога с учетом цифровой фильтрации | RO |
| Out_over | 1 | 0x0 | Текущее состояние линии монитора верхнего порога без цифровой фильтрации | RO |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| Out_under | 0 | 0x0 | Текущее состояние линии монитора нижнего порога без цифровой фильтрации | RO |

5.5 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 por1 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 por1 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 por1 отслеживает как уровень напряжения на шине vcc, так и уровень напряжения на шине vdd

Если значение напряжения питания vcc опускается ниже значения vldr, блок выдает на выход сигнал сброса, пока напряжение питания снова не станет выше vpor.

Поведение схемы формирования сигнала BOR (активным является уровень логического “0”) аналогична поведению схемы формирования сигнала POR, но уровни срабатывания vbpr и vbdr находятся выше, чем в схеме формирования POR (см. таблицу 1).

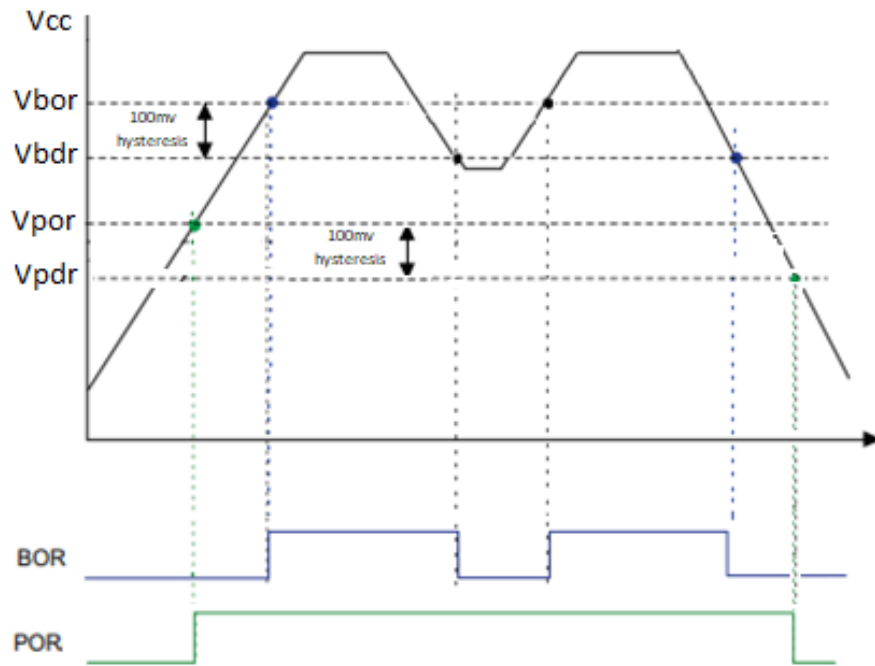


Рисунок 90 – Упрощенная диаграмма работы схемы формирования сигналов POR и BOR (влияние напряжения на шине vdd не показано).

Таблица 179 – 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 | vpdr | 2,37 | 2,4 | 2,43 | В | с учетом разброса источника опорного напряжения $\pm 1\%$ |
| Уровень напряжения перехода сигнала POR в пассивное состояние | vpor | 2,47 | 2,5 | 2,53 | В | |
| Уровень напряжения перехода сигнала BOR в активное состояние | vbdr | 2,62 | 2,65 | 2,68 | В | |
| Уровень напряжения перехода сигнала BOR в пассивное состояние | vbor | 2,72 | 2,75 | 2,78 | В | |
| Уровень перехода сигнала слежения за питанием vdd в активное состояние | vpdr_vdd | 1,18 | 1,2 | 1,22 | В | |
| Уровень перехода сигнала слежения за | vpor_vdd | 1,23 | 1,25 | 1,27 | В | |

| | | | | | | |
|---|----------------------|-----|-----|-----|-----|--|
| питанием vdd в пассивное состояние | | | | | | |
| Задержка срабатывания схемы по выходу porh из "0" в "1" | Tporh01 | 35 | 68 | 115 | нс | с учетом разброса источника опорного напряжения $\pm 1\%$; превышение над порогом 100мВ и емкостью нагрузки 1пФ |
| 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 | |
| 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 | I _{sup_vcc} | 15 | 23 | 40 | мкА | с учетом разброса источника опорного напряжения $\pm 1\%$ |
| Ток потребления по цепи vdd | I _{sup_vdd} | - | - | < 1 | нА | |

5.6 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, автоматического 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 PowerSwitch модуля Wakeup (смещение 0x0C относительно базового адреса Wakeup). Назначение полей регистра представлено в таблице 180.

Таблица 180 – Регистр PowerSwitch модуля Wakeup

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-----------------|---------|--------------------|--|--------|
| Зарезервировано | 31:3 | Зарезервировано | | |
| Batt_Good | 2 | 0x0 | Флаг состояние резервного (батарейного) источника питания | RO |
| Control | 1 | 0x0 | Выбор источника при принудительного переключения (при En = "1") 0 – VCC 1 – VCC_BU | RW |

| Обозначение | Разряды | Начальное значение | Назначение | Доступ |
|-------------|---------|--------------------|---|--------|
| En | 0 | 0x0 | Разрешение принудительного переключения на один из источников питания | RW |

Основные электрические характеристики представлены в таблице 181.

Таблица 181 – 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 | T _{vcc2vbat} | 370 | 500 | 720 |
| Задержка между подъемом vcc выше уровня vbat2vcc и переключением на основной источник питания, нс | T _{vbat2vcc} | 50 | 80 | 120 |
| Задержка между сигналом frwr и переключением на резервный источник питания, нс | T _{vcc2vbat_frwr} | 6 | 7 | 9 |
| Задержка между сигналом frwr и переключением на основной источник питания, нс | T _{vbat2vcc_frwr} | 10 | 12 | 14 |

Структурная схемы слежения и переключения источника внешнего питания представлена на рисунке 91.

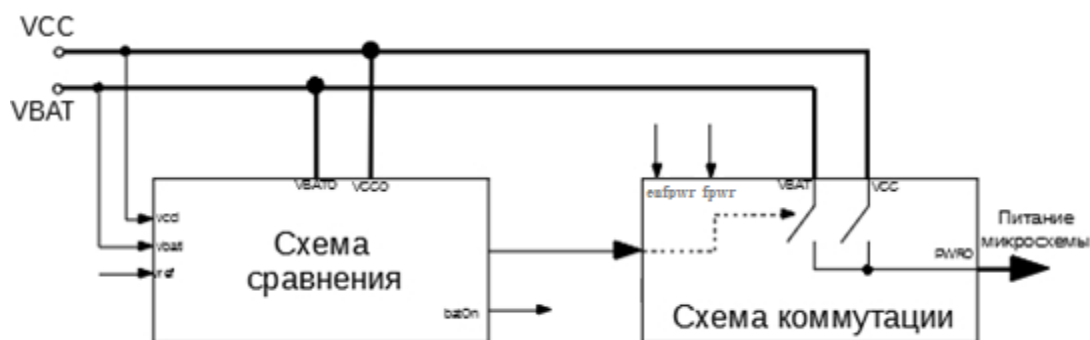


Рисунок 91 – Структурная схема слежения и переключения источника внешнего питания

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.

5.7 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 представлены на рисунке 92. Регулятор напряжения выполнен по схеме класса АВ на основе усилителя ошибки регулирования, проходного транзистора Р-типа, разрядного транзистора N-типа, резистивного делителя и схемы формирования сигнала о состоянии питания на основе компаратора. Для обес-

печения 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.

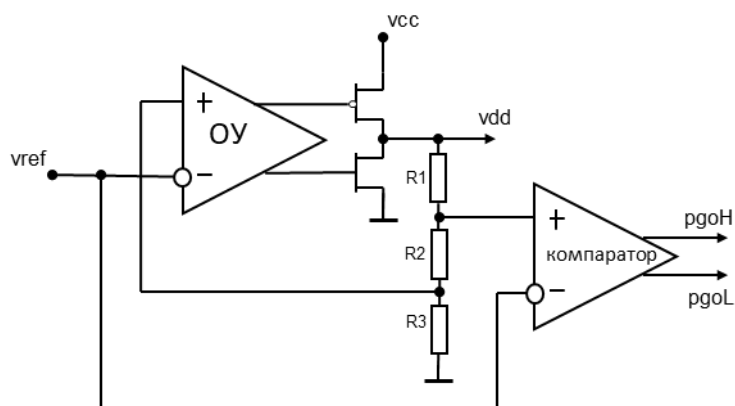


Рисунок 92 – Структурная схема регулятора напряжения

Основные электрические характеристики приведены в таблице 182.

Т а б л и ц а 182 – 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, мкА | I _{сс} | 21 | 29 | 44 |
| Ток потребления в выключенном режиме, нА | I _{qc} | – | – | 25 |
| Ток нагрузки, мА | I _{dd} | 0 | 3 | 4,5 |
| Уровень переключения из «1» в «0» схемы формирования сигнала о состоянии питания vdd, В | U _{pg10} | 1,32 | 1,35 | 1,38 |
| Уровень переключения из «0» в «1» схемы формирования сигнала о состоянии питания vdd, В | U _{pg01} | 1,42 | 1,45 | 1,48 |
| Коэффициент подавления помех по питанию, дБ | PSSR | 40 | 45 | 52 |

5.8 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 183.

Т а б л и ц а 183 – 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 93. 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». Предусмотрено выключение регулятора внешним сигналом для обеспечения возможности подачи внешнего питания 1,8 В. 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.

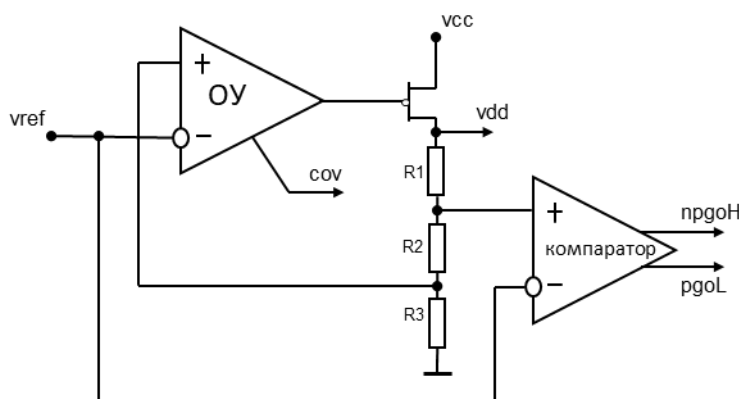


Рисунок 93 – Структурная схема регулятора напряжения

5.9 Источники опорного напряжения

Схема источника опорного тока формирует выходной ток, слабо зависящий от температуры и возможностью перевода схемы в выключенное состояние в целях снижение энергопотребления.

5.10 Калибруемый источник опорного напряжения

Схема источника опорного тока формирует выходной ток, слабо зависящий от температуры и возможностью перевода схемы в выключенное состояние в целях снижение энергопотребления.

Особенностью данной схемы является наличие блока калибровки, с помощью которого можно подстраивать значение выходного напряжения опорного источника в диапазоне ± 75 мВ с шагом ~ 11 мВ по отношению к расчетному и оптимальному с точки зрения температурной зависимости значению. Код «1000» соответствует расчетному и оптимальному с точки зрения температурной зависимости значению выходного напряжения опорного источника, код «0000» соответствует минимальному значению выходного напряжения, код «1111» соответствует максимальному значению выходного напряжения.

6 Назначение выводов

Назначение выводов, представлено в таблице 182.

| | | | | | | | | | | | | |
|----|-----------|--|---------|------------|------------------------------------|-----------------|-------|---|-------------------|-------|--|-------|
| 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 OTP 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_pard | Выход | Выход цифровой подсистемы тестирования аналоговых блоков | АЦП_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 | UART0_ddis | Выход | Сигнал считывания принятых данных | |
| 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 | АЦП_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 | |

| | | | | | | | | | | | | | |
|----|-----------|--|----------|------------|-------------------------------------|-----------------|------------|--|---------------|-------|--------------------------------|-------|--|
| | | (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 управление | Timer16_2_out | Выход | 16-разрядный таймер 2.Выход | АЦП_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_in2 | Вход | 16-разрядный таймер 2.Вход 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_in1 | Вход | 16-разрядный таймер 2.Вход 1 | АЦП_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_in2 | Вход | 16-разрядный таймер 1.Вход 2 | АЦП_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_in1 | Вход | 16-разрядный таймер 1.Вход 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 | АЦП_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_in2 | Вход | 16-разрядный таймер 0.Вход 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_in1 | Вход | 16-разрядный таймер 0.Вход 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 | АЦП_3 | |

| | | | | | | | | | | | | | |
|----|-----------|--|----------|------------|-------------------------------------|-----------------|------------|--|----------------|------------|---|--|--|
| | | | | | | | | | | | | вход тактирования | |
| 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 режиме ведомого | Timer32_1_ch 4 | Вход-выход | 32х-разрядный таймер1. Каналы 1-4. Выход-ШИМ, Вход - захват/сравнение | АЦП_2 | |
| 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 | | | |
| 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 (ANA_3V) | |

Список использованных сокращений

| | |
|------------------|--|
| АЦП | – аналогово-цифровой преобразователь |
| ОКР | – опытно-конструкторская работа |
| ОЗУ | – оперативное запоминающее устройство (random access memory (RAM)) |
| ПЗУ | – постоянное запоминающее устройство (read-only memory (ROM)) |
| ЦАП | – цифро-аналоговый преобразователь |
| ШИМ (PWM) | – широтно-импульсная модуляция (pulse width modulation) |
| АНВ | – высокопроизводительная шина AMBA (AMBA High-performance Bus) |
| АРВ | – шина AMBA для периферийных устройств (AMBA Peripheral Bus) |
| CMOS | – комплементарная логика на транзисторах металл-оксид-полупроводник (Complementary-symmetry/metal-oxide semi- |
| EEPROM | – электрически стираемое перепрограммируемое ПЗУ (Electrically Erasable Programmable Read-Only Memory(ЭСППЗУ)) |
| GPIO | – порт ввода/вывода общего назначения (general purpose port input/output) |
| I ² C | – 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) |
| PID | – идентификатор устройства (Product Identifier) |
| UART | – универсальный синхронно/асинхронный приёмопередатчик (universal asynchronous receiver-transmitter) |
| OTP | – One Time Programmable –однократно программируемая память |