

МИКРОСХЕМА ИНТЕГРАЛЬНАЯ

К1921ВК01Т

Руководство пользователя

от 10.09.2019

Содержание

| | |
|--|-----|
| Введение | 5 |
| 1 Область применения и особенности | 6 |
| 2 Краткое техническое описание микроконтроллера | 7 |
| 2.1 Функциональные параметры | 7 |
| 2.2 Электрические параметры | 23 |
| 3 Архитектура изделия..... | 26 |
| 4 Система тактирования микроконтроллера..... | 27 |
| 4.1 Описание блока синхронизации..... | 27 |
| 4.2 Алгоритм перехода на системную частоту | 28 |
| 4.3 Управление тактовыми сигналами периферии | 29 |
| 5 Сброс микроконтроллера..... | 32 |
| 6 Организация памяти | 34 |
| 7 Флеш-память и внешняя память. Загрузка микроконтроллера..... | 39 |
| 7.1 Загрузочная флеш-память | 39 |
| 7.2 Пользовательская флеш-память | 40 |
| 7.3 Алгоритм стирания всей флеш-памяти | 42 |
| 7.4 Временные параметры..... | 42 |
| 7.5 Контроллер внешней памяти | 44 |
| 7.6 Режимы загрузки микроконтроллера..... | 46 |
| 8 Контроллер прямого доступа к памяти | 49 |
| 8.1 Программное управление модулем DMA | 50 |
| 8.2 Правила обмена данными | 55 |
| 8.3 Правила арбитража | 56 |
| 8.4 Типы циклов | 58 |
| 8.5 Индикация ошибок | 69 |
| 9 Прерывания | 70 |
| 10 Порты..... | 75 |
| 11 Таймеры..... | 80 |
| 12 Блоки захвата | 81 |
| 12.1 Режим захвата времени | 82 |
| 12.2 Режим работы «генератор ШИМ» | 83 |
| 13 Модули квадратурных декодеров | 86 |
| 13.1 Обработчик сигналов входов..... | 86 |
| 13.2 Квадратурный преобразователь | 87 |
| 13.3 Счетчик позиции..... | 89 |
| 13.4 Таймер временных отсчетов | 93 |
| 13.5 Модуль захвата времени | 93 |
| 13.6 Сторожевой таймер | 95 |
| 13.7 Система прерываний | 96 |
| 14 Блоки ШИМ | 97 |
| 14.1 Таймер..... | 98 |
| 14.2 Компаратор..... | 101 |
| 14.3 Обработчик событий | 103 |
| 14.4 Пороговый выключатель..... | 106 |
| 14.5 Генератор задержки ШИМ | 107 |
| 14.6 Фильтр коротких импульсов | 108 |
| 14.7 Модулятор | 108 |
| 14.8 Детектор сигнала аварии..... | 110 |
| 14.9 Триггер событий | 111 |
| 14.10 Блок ШИМ высокого разрешения..... | 113 |

| | | |
|-------|--|-----|
| 14.11 | Сопряжение с блоком АЦП | 115 |
| 15 | Приемопередатчики UART | 116 |
| 15.1 | Функционирование блока UART | 116 |
| 15.2 | Интерфейс прямого доступа к памяти | 121 |
| 15.3 | Прерывания | 123 |
| 15.4 | Программирование | 124 |
| 16 | Контроллеры интерфейса SPI..... | 125 |
| 16.1 | Структура контроллера SPI | 125 |
| 16.2 | Функционирование | 128 |
| 16.3 | Прерывания | 132 |
| 17 | Контроллер интерфейса I2C | 133 |
| 17.1 | Протокол шины..... | 133 |
| 17.2 | Функциональное описание | 140 |
| 17.3 | Инициализация и функционирование | 143 |
| 18 | Контроллер интерфейса CAN..... | 157 |
| 18.1 | Протокол CAN | 157 |
| 18.2 | Структура и функционирование контроллера CAN..... | 163 |
| 18.3 | Узел контроллера CAN | 168 |
| 18.4 | Объекты сообщений | 173 |
| 18.5 | Прием и передача сообщений..... | 177 |
| 18.6 | Фильтрация сообщений..... | 180 |
| 18.7 | Удаленные запросы | 181 |
| 18.8 | Дополнительные режимы передачи | 182 |
| 18.9 | FIFO структура объектов сообщений | 183 |
| 18.10 | Режим шлюза..... | 186 |
| 18.11 | Прерывания объектов сообщений..... | 188 |
| 18.12 | Программирование контроллера CAN | 191 |
| 19 | Контроллер интерфейса Ethernet 10/100 | 192 |
| 19.1 | Операции передачи..... | 193 |
| 19.2 | Операции приема | 194 |
| 20 | Контроллер интерфейса USB OTG | 195 |
| 20.1 | Функции хоста | 195 |
| 20.2 | Работа хоста..... | 197 |
| 20.3 | Работа устройства | 198 |
| 20.4 | USB OTG | 201 |
| 21 | Блок АЦП | 203 |
| 21.1 | Секвенсор | 204 |
| 21.2 | Модуль АЦП | 206 |
| 21.3 | Буфер результатов измерений | 208 |
| 21.4 | Цифровой компаратор | 209 |
| 22 | Блок аналоговых компараторов | 213 |
| 22.1 | Описание функционирования..... | 213 |
| 23 | Блок часов реального времени | 217 |
| 23.1 | Рекомендации по подключению и трассировке сигналов на печатной плате | 217 |
| 23.2 | Описание работы блока..... | 217 |
| 24 | Сторожевой таймер | 219 |
| 25 | Программно-аппаратные средства отладки | 220 |
| | Заключение..... | 221 |
| | Приложение А (обязательное) Регистры микроконтроллера..... | 222 |
| А.1 | Регистры контроллера АЦП..... | 222 |
| А.2 | Регистры GPIO и общего назначения | 236 |
| А.3 | Регистры контроллера CAN..... | 251 |

| | |
|---|-----|
| A.4 Регистры контроллера Ethernet 10/100..... | 275 |
| A.5 Регистры контроллера USB OTG | 294 |
| A.6 Регистры RTC | 315 |
| A.7 Регистры сторожевого таймера | 319 |
| A.8 Регистры контроллера I2C | 322 |
| A.9 Регистры таймера | 328 |
| A.10 Регистры контроллера DMA | 330 |
| A.11 Регистры блока UART | 337 |
| A.12 Регистры блока ШИМ | 346 |
| A.13 Регистры блока захвата | 367 |
| A.14 Регистры квадратурного декодера | 372 |
| A.15 Регистры загрузочной флеш-памяти..... | 380 |
| A.16 Регистры аналогового компаратора | 385 |
| A.17 Регистры контроллера SPI | 389 |
| A.18 Регистры контроллера пользовательской флеш-памяти..... | 394 |
| Приложение Б (обязательное) Карта памяти регистров микроконтроллера | 401 |
| Приложение В (обязательное) Коды состояний функционирования блока I2C | 439 |
| Приложение Г (обязательное) Регистры прерываний | 447 |
| Приложение Д (справочное) Пример схемы включения микроконтроллера..... | 449 |
| Лист регистрации изменений | 451 |

Введение

Микроконтроллеры с вариантом ядра Cortex-M на сегодняшний день являются одними из самых популярных. Они применяются при разработке и изготовлении электронной техники. Высокая вычислительная мощность и при этом относительно низкая стоимость делают эти устройства привлекательными для самого широкого круга разработчиков.

Микросхема K1921BK01T представляет собой СБИС 32-разрядного микроконтроллера на базе ядра ARM Cortex-M4F, предназначенного для промышленных и потребительских приложений, включая системы дистанционного мониторинга, контрольно-измерительные приборы, сетевые устройства, системы автоматизации производственных процессов, автомобильную электронику, системы управления электродвигателями.

В состав микроконтроллера входит широкий набор как цифровой, так и аналоговой периферии, в связи с чем, он может применяться в различных системах цифровой обработки сигналов, в том числе требующих точных аналогово-цифровых и цифро-аналоговых преобразований, в системах управления и сбора информации.

В настоящем руководстве пользователя приведено описание архитектуры, функционального построения и периферии микроконтроллера K1921BK01T. Руководство пользователя может служить практическим руководством по применению микроконтроллера для разработчиков систем на его основе и программистов.

1 Область применения и особенности

Сфера применения микроконтроллера довольно широка – средства измерений, связи, наблюдения, безопасности, автоматизация производства, медицина, энергетика, промышленность, различные системы управления. Тем не менее, основной областью применения является электропривод.

Для эффективного управления в электромеханических системах была разработана дополнительная периферия: блоки ШИМ (в том числе ШИМ высокого разрешения), блок АЦП с интерфейсом к контроллеру прямого доступа к памяти, блок аналоговых компараторов, модуль захвата/сравнения, блок импульсного квадратурного декодера, используемого для обработки сигналов датчиков положения ротора в высокопроизводительных системах для определения положения, направления и скорости вращения. Внутри микроконтроллера блоки имеют стандартный интерфейс AMBA AHB и APB.

Разработанный микроконтроллер имеет встроенную флэш-память программ размером 1 Мбайт, которую можно использовать для хранения и загрузки пользовательского программного обеспечения. Также во флэш-памяти существует особый защищенный раздел, который может быть использован для хранения начального загрузчика. Кроме этого, реализована возможность загрузки из внешней памяти.

Помимо флэш-памяти программ, микроконтроллер содержит флэш-память данных размером 64 Кбайта, которая может использоваться для хранения пользовательских данных, и дополнительный информационный блок размером 512 байт, в котором хранятся загрузочные настройки, а также настройки доступа к страницам флэш-памяти программ и флэш-памяти данных. Так, возможно настроить постраничный запрет стирания отдельных страниц обеих флэш-памятей, а также выбрать способ загрузки микроконтроллера. Данные меры позволяют повысить стойкость микроконтроллера к возможным попыткам загрузки постороннего программного обеспечения, а также к стиранию важных областей флэш-памяти.

Система тактирования микроконтроллера позволяет использовать различные источники тактового сигнала, что позволяет расширить набор применений и решаемых задач пользователя. Микроконтроллер может тактироваться от внутреннего RC-генератора с частотой от 3,5 до 6,5 МГц, внутреннего осциллятора с внешним кварцевым резонатором, тактовым сигналом, подаваемым на вход порта ввода-вывода, а также сигналом встроенного генератора PLL. Также существует возможность гибкой настройки тактовых сигналов для блоков периферии.

Для снижения потребления микросхемы предусмотрена возможность отключения тактовых сигналов отдельных блоков периферии в случае, если они не используются пользователем. При переходе процессора в режим пониженного энергопотребления возможно отключение тактового сигнала процессора (команда WFI) или переключение тактового сигнала на низкую частоту (команда WFE).

2 Краткое техническое описание микроконтроллера

2.1 Функциональные параметры

Структурная схема микроконтроллера показана на рисунке 2.1.

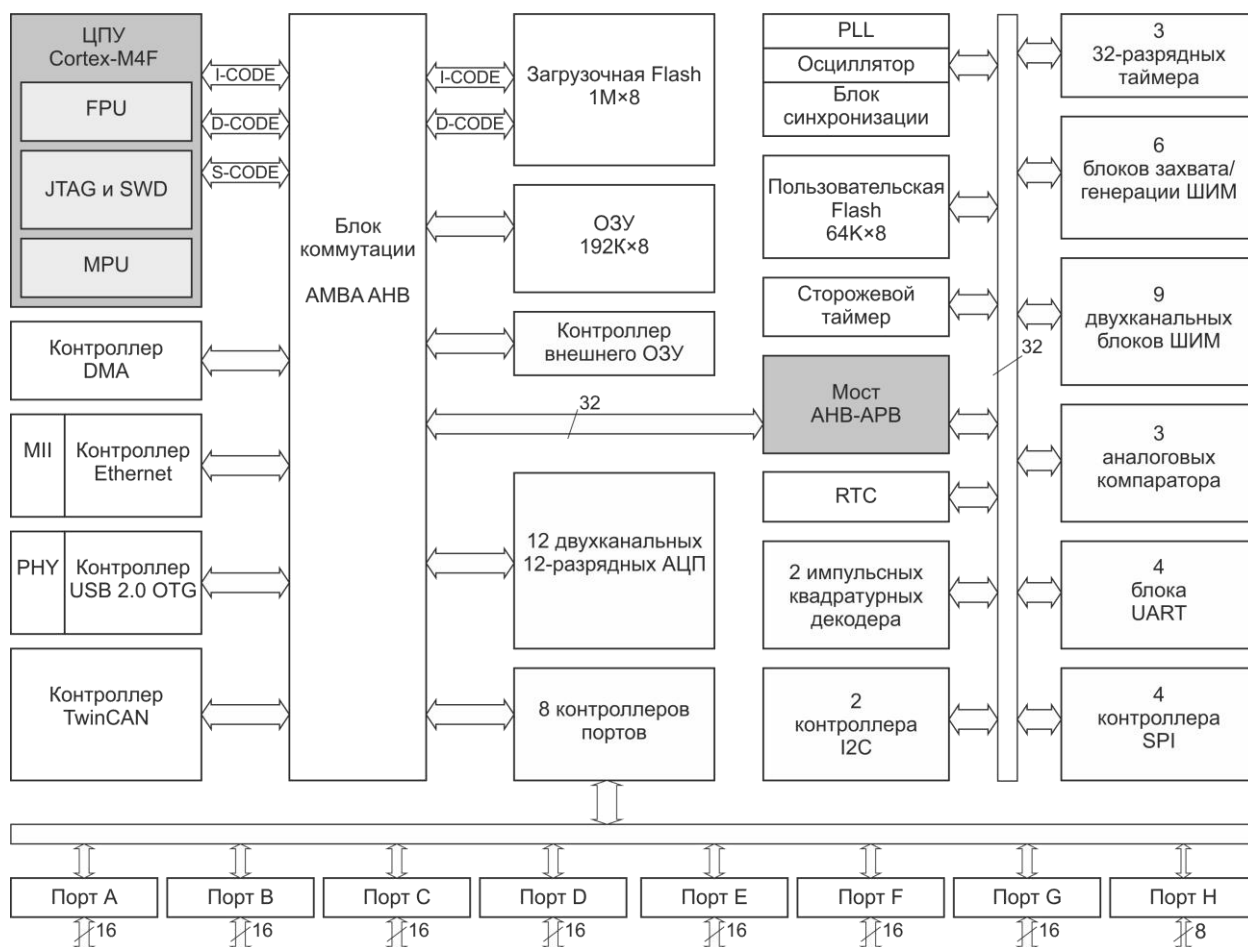


Рисунок 2.1 – Структурная схема микроконтроллера

В состав микроконтроллера входят функциональные элементы (см. рисунок 2.1):

- процессорное ядро ARM Cortex-M4F производительностью не менее 125 миллионов инструкций в секунду с поддержкой набора одноцикловых команд умножения с накоплением (блок MPU), команд централизованного управления потоком данных, арифметических и логических команд и встроенным модулем обработки команд с плавающей запятой с одинарной точностью (блок FPU);

- загрузочная флеш-память (Flash) емкостью 1 Мбайт;
- ОЗУ объемом 192 Кбайта;
- пользовательская флеш-память объемом 64 Кбайта;
- контроллер внешней статической памяти (SRAM, PROM, NOR-FLASH);
- 24-канальный контроллер прямого доступа к памяти (DMA);
- схема сброса и сторожевой таймер;
- часы реального времени с батарейным питанием (RTC);
- синтезатор частоты на основе ФАПЧ (PLL);

- двенадцать двухканальных 12-разрядных АЦП с режимами цифрового компаратора для каждого из каналов (равно или больше, равно или меньше, попадание в диапазон, выход из диапазона) и функцией автоматического запуска модулей ШИМ по событию «окончание преобразования»;

- три аналоговых компаратора с функцией автоматического запуска модулей ШИМ по событиям сравнения «равно или больше» и «равно или меньше»;
- девять модулей ШИМ, шесть из которых поддерживают режим высокого разрешения (возможность изменения длительности импульсов на величину менее периода тактового сигнала);
- два импульсных квадратурных декодера, используемых для обработки сигналов датчиков положения ротора в высокопроизводительных системах для определения положения, направления и скорости вращения;
- шесть модулей захвата/сравнения;
- три 32-разрядных таймера;
- отладочный интерфейс JTAG и ARM SWD (Serial Wire Debug);
- семь 16-разрядных и один 8-разрядный порт ввода/вывода с отдельно программируемыми мультиплексированными выводами общего назначения;
- четыре последовательных порта UART;
- контроллеры интерфейсов:
 - CAN (протокол 2.0b) с двумя портами ввода-вывода;
 - I2C с поддержкой частоты передачи данных более 1 МГц (два порта);
 - SPI (четыре порта);
 - USB 2.0 Device/Host с физическим уровнем PHY;
 - Ethernet 10/100 Мбит/с с интерфейсом МП.

Условное графическое изображение микроконтроллера приведено на рисунке 2.2.

Функциональное назначение выводов указано в таблицах 2.1 и 2.2. В таблице 2.1 в графе «Обозначение вывода» первым указывается название основной функции вывода микроконтроллера. Далее, в порядке следования, первая, вторая и третья альтернативные функции.

Примечание – Выводы A1 – A7 имеют дополнительные альтернативные функции CMP_C3+, CMP_C3-, CMP_C2+, CMP_C2-, CMP_C1+, CMP_C1- и CMP_DACSUP, которые постоянно активны.

После сброса микроконтроллера выводы портов A, B, C, D, E, F, G, H (A – H) конфигурируются как выводы общего назначения и находятся в третьем состоянии. Исключение составляют выводы отладочного модуля JTAG (B0, B1, B2, E1, E10) и выводы приемопередатчика UART0 (D11 и E0). У этих выводов микроконтроллера по умолчанию включены указанные альтернативные функции. Выбор альтернативной функции осуществляется посредством регистра GPIOCTL (за исключением функций CMP_C3+, CMP_C3-, CMP_C2+, CMP_C2-, CMP_C1+, CMP_C1- и CMP_DACSUP).

В графе «Номер вывода» указывается номер вывода микроконтроллера.

В графе «Тип вывода» используются обозначения: I – вход, O – выход, Z – третье состояние.

Микросхема выполнена в металлополимерном (пластмассовом) корпусе 4406.208-1 (QFP-208). Масса микросхемы – не более 7 г.

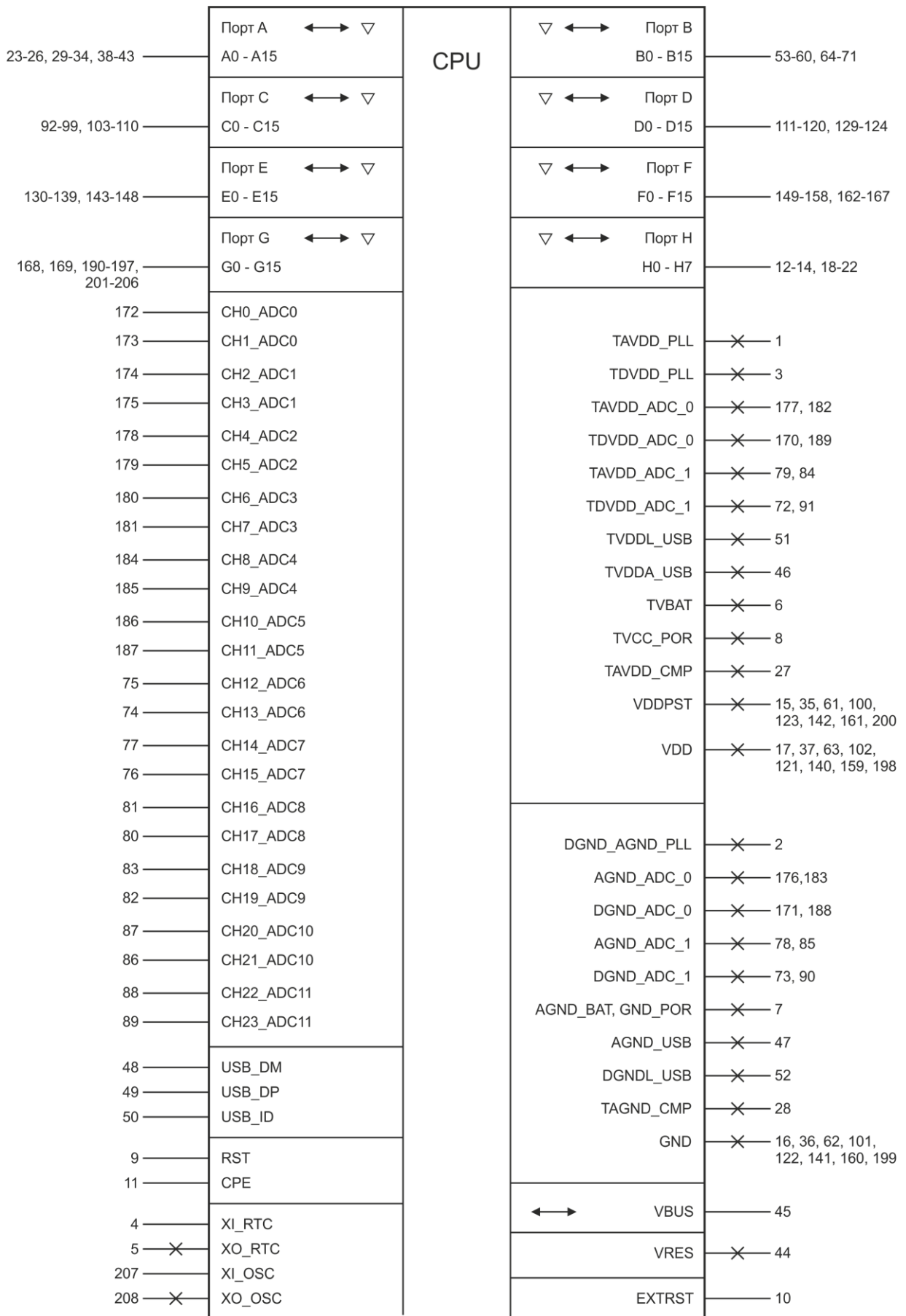


Рисунок 2.2 – Условное графическое изображение микросхемы (альтернативные функции выводов портов А, В, С, D, E, F, G, H указаны в таблице 2.1)

Таблица 2.1 – Функциональное назначение выводов, имеющих альтернативные функции

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|------------|-------------------------------------|------------|--|
| A0 | | 23 | I/O/Z | Вход/выход «порт А», разряд 0 |
| | CLK_USB | | I | Вход «тактовый сигнал USB» |
| | SPI_FSS2 | | I | Вход «выбор ведомого устройства SPI2» |
| | MEM_DATA12 | | I/O | Вход/выход шины данных внешней памяти, разряд 12 |
| A1 | | 24 | I/O/Z | Вход/выход «порт А», разряд 1 |
| | SPI_TXD0 | | O | Выход «линия передачи данных SPI0» |
| | PWM_B8 | | O | Выход «линия В блока 8 ШИМ» |
| | – | | – | – |
| CMP_C3+ | I | Аналоговый вход С+ компаратора CMP3 | | |
| A2 | | 25 | I/O/Z | Вход/выход «порт А», разряд 2 |
| | PWM_TZ2 | | I | Вход «сигнал 2 аварии» |
| | SPI_TXD1 | | O | Выход «линия передачи данных SPI1» |
| | MEM_OE0# | | O | Выход «сигнал чтения внешней памяти 0» |
| CMP_C3– | I | Аналоговый вход С– компаратора CMP3 | | |
| A3 | | 26 | I/O/Z | Вход/выход «порт А», разряд 3 |
| | UART_TXD1 | | O | Выход «линия передачи данных UART1» |
| | PWM_A6 | | O | Выход «линия А блока 6 ШИМ» |
| | MEM_OE1# | | O | Выход «сигнал чтения внешней памяти 1» |
| CMP_C2+ | I | Аналоговый вход С+ компаратора CMP2 | | |
| A4 | | 29 | I/O/Z | Вход/выход «порт А», разряд 4 |
| | UART_RXD1 | | I | Вход «линия приема данных UART1» |
| | PWM_B6 | | O | Выход «линия В блока 6 ШИМ» |
| | – | | – | – |
| CMP_C2– | I | Аналоговый вход С– компаратора CMP2 | | |
| A5 | | 30 | I/O/Z | Вход/выход «порт А», разряд 5 |
| | SPI_FSS0 | | I | Вход «выбор ведомого устройства SPI0» |
| | PWM_A7 | | O | Выход «линия А блока 7 ШИМ» |
| | – | | – | – |
| CMP_C1+ | I | Аналоговый вход С+ компаратора CMP1 | | |
| A6 | | 31 | I/O/Z | Вход/выход «порт А», разряд 6 |
| | SPI_CLK0 | | I/O | Вход/выход «синхросигнал передачи SPI0» |
| | PWM_B7 | | O | Выход «линия В блока 7 ШИМ» |
| | – | | – | – |
| CMP_C1– | I | Аналоговый вход С– компаратора CMP1 | | |
| A7 | | 32 | I/O/Z | Вход/выход «порт А», разряд 7 |
| | SPI_RXD0 | | I | Вход «линия приема данных SPI0» |
| | PWM_A8 | | O | Выход «линия А блока 8 ШИМ» |
| | – | | – | – |
| CMP_DAC SUP | I | Питание ЦАП аналоговых компараторов | | |
| A8 | | 33 | I/O/Z | Вход/выход «порт А», разряд 8 |
| | MEM_ADDR0 | | O | Выход шины адреса внешней памяти, разряд 0 |
| | MII_TXCLK | | I | Вход «синхросигнал передачи» |
| | PWM_TZ0 | | I | Вход «сигнал 0 аварии» |
| A9 | | 34 | I/O/Z | Вход/выход «порт А», разряд 9 |
| | MEM_ADDR1 | | O | Выход шины адреса внешней памяти, разряд 1 |
| | MII_TXD0 | | O | Выход «передаваемый нулевой бит данных» |
| | PWM_TZ1 | | I | Вход «сигнал 1 аварии» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|--------------------|--------------|------------|--|
| A10 | | 38 | I/O/Z | Вход/выход «порт А», разряд 10 |
| | MEM_ADDR2 | | O | Выход шины адреса внешней памяти, разряд 2 |
| | MII_TXD1 | | O | Выход «передаваемый первый бит данных» |
| | PWM_B0 | | O | Выход «линия В блока 0 ШИМ» |
| A11 | | 39 | I/O/Z | Вход/выход «порт А», разряд 11 |
| | MEM_ADDR3 | | O | Выход шины адреса внешней памяти, разряд 3 |
| | MII_TXD2 | | O | Выход «передаваемый второй бит данных» |
| | PWM_B1 | | O | Выход «линия В блока 1 ШИМ» |
| A12 | | 40 | I/O/Z | Вход/выход «порт А», разряд 12 |
| | MEM_ADDR4 | | O | Выход шины адреса внешней памяти, разряд 4 |
| | MII_TXD3 | | O | Выход «передаваемый третий бит данных» |
| | PWM_B2 | | O | Выход «линия В блока 2 ШИМ» |
| A13 | | 41 | I/O/Z | Вход/выход «порт А», разряд 13 |
| | MEM_ADDR5 | | O | Выход шины адреса внешней памяти, разряд 5 |
| | MII_TX_EN | | O | Выход «разрешение передачи» |
| | PWM_B3 | | O | Выход «линия В блока 3 ШИМ» |
| A14 | | 42 | I/O/Z | Вход/выход «порт А», разряд 14 |
| | MEM_ADDR6 | | O | Выход шины адреса внешней памяти, разряд 6 |
| | MII_TX_ER | | O | Выход «ошибка передачи» |
| | PWM_B4 | | O | Выход «линия В блока 4 ШИМ» |
| A15 | | 43 | I/O/Z | Вход/выход «порт А», разряд 15 |
| | MEM_ADDR7 | | O | Выход шины адреса внешней памяти, разряд 7 |
| | MII_CRS | | I | Вход «контроль носителя» |
| | PWM_B5 | | O | Выход «линия В блока 5 ШИМ» |
| B0 | | 53 | I/O/Z | Вход/выход «порт В», разряд 0 |
| | JTAG_TDI | | I | Вход «данные JTAG» (включен по умолчанию) |
| | SPI_FSS1 | | I | Вход «выбор ведомого устройства SPI1» |
| | MEM_WE# | | O | Выход «сигнал записи во внешнюю память» |
| B1 | | 54 | I/O/Z | Вход/выход «порт В», разряд 1 |
| | JTAG_TMS/ SWDIO | | I | Вход «переключение режима JTAG» (включен по умолчанию) |
| | SPI_CLK1 | | I/O | Вход/выход «синхросигнал передачи SPI1» |
| | MEM_CE0# | | O | Выход «сигнал 0 выбора внешнего устройства» |
| B2 | | 55 | I/O/Z | Вход/выход «порт В», разряд 2 |
| | JTAG_TCK/ SWCLK | | I | Вход «тактовый сигнал JTAG» (включен по умолчанию) |
| | SPI_RXD1 | | I | Вход «линия приема данных SPI1» |
| | MEM_CE1# | | O | Выход «сигнал 1 выбора внешнего устройства» |
| B3 | | 56 | I/O/Z | Вход/выход «порт В», разряд 3 |
| | MII_TXCLK | | I | Вход «синхросигнал передачи» |
| | CAN_TX0 | | O | Выход «линия передачи данных CAN0» |
| | MEM_ADDR0 | | O | Выход шины адреса внешней памяти, разряд 0 |
| B4 | | 57 | I/O/Z | Вход/выход «порт В», разряд 4 |
| | MEM_ADDR8 | | O | Выход шины адреса внешней памяти, разряд 8 |
| | MII_COL | | I | Вход «обнаружение конфликта» |
| | PWM_B6 | | O | Выход «линия В блока 6 ШИМ» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|------------|--------------|------------|---|
| B5 | | 58 | I/O/Z | Вход/выход «порт В», разряд 5 |
| | MEM_ADDR9 | | O | Выход шины адреса внешней памяти, разряд 9 |
| | MII_MDC | | O | Выход «линия тактового сигнала управляющих данных» |
| | PWM_A7 | | O | Выход «линия А блока 7 ШИМ» |
| B6 | | 59 | I/O/Z | Вход/выход «порт В», разряд 6 |
| | MEM_ADDR10 | | O | Выход шины адреса внешней памяти, разряд 10 |
| | MII_MDIO | | I/O | Вход/выход «линия управляющих данных» |
| | PWM_B7 | | O | Выход «линия В блока 7 ШИМ» |
| B7 | | 60 | I/O/Z | Вход/выход «порт В», разряд 7 |
| | MEM_ADDR11 | | O | Выход шины адреса внешней памяти, разряд 11 |
| | MII_RXCLK | | I | Вход «синхросигнал приема» |
| | PWM_A8 | | O | Выход «линия А блока 8 ШИМ» |
| B8 | | 64 | I/O/Z | Вход/выход «порт В», разряд 8 |
| | MEM_ADDR12 | | O | Выход шины адреса внешней памяти, разряд 12 |
| | MII_RXD0 | | I | Вход «принимаемый нулевой бит данных» |
| | PWM_B8 | | O | Выход «линия В блока 8 ШИМ» |
| B9 | | 65 | I/O/Z | Вход/выход «порт В», разряд 9 |
| | MEM_ADDR13 | | O | Выход шины адреса внешней памяти, разряд 13 |
| | MII_RXD1 | | I | Вход «принимаемый первый бит данных» |
| | PWM_SYNCI | | I | Вход «внешний синхросигнал для блоков ШИМ» |
| B10 | | 66 | I/O/Z | Вход/выход «порт В», разряд 10 |
| | MEM_ADDR14 | | O | Выход шины адреса внешней памяти, разряд 14 |
| | MII_RXD2 | | I | Вход «принимаемый второй бит данных» |
| | CMP_OUT0 | | O | Выход «результат сравнения аналогового компаратора 0» |
| B11 | | 67 | I/O/Z | Вход/выход «порт В», разряд 11 |
| | MEM_ADDR15 | | O | Выход шины адреса внешней памяти, разряд 15 |
| | MII_RXD3 | | I | Вход «принимаемый третий бит данных» |
| | CMP_OUT1 | | O | Выход «результат сравнения аналогового компаратора 1» |
| B12 | | 68 | I/O/Z | Вход/выход «порт В», разряд 12 |
| | MEM_ADDR16 | | O | Выход шины адреса внешней памяти, разряд 16 |
| | MII_RX_DV | | I | Вход «валидность принимаемых данных» |
| | CMP_OUT2 | | O | Выход «результат сравнения аналогового компаратора 2» |
| B13 | | 69 | I/O/Z | Вход/выход «порт В», разряд 13 |
| | MEM_ADDR17 | | O | Выход шины адреса внешней памяти, разряд 17 |
| | MII_RX_ER | | I | Вход «ошибка приема» |
| | SPI_FSS2 | | I | Вход «выбор ведомого устройства SPI2» |
| B14 | | 70 | I/O/Z | Вход/выход «порт В», разряд 14 |
| | MEM_ADDR18 | | O | Выход шины адреса внешней памяти, разряд 18 |
| | CAN_TX0 | | O | Выход «линия передачи данных CAN0» |
| | SPI_CLK2 | | I/O | Вход/выход «синхросигнал передачи SPI2» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|-----------|--------------|--|---|
| B15 | | 71 | I/O/Z | Вход/выход «порт В», разряд 15 |
| | MEM_DATA0 | | I/O | Вход/выход шины данных внешней памяти, разряд 0 |
| | CAN_RX0 | | I | Вход «линия приема данных CAN0» |
| | SPI_RXD2 | | I | Вход «линия приема данных SPI2» |
| C0 | | 92 | I/O/Z | Вход/выход «порт С», разряд 0 |
| | MII_TXD0 | | O | Выход «передаваемый нулевой бит данных» |
| | – | | – | – |
| | MEM_ADDR1 | O | Выход шины адреса внешней памяти, разряд 1 | |
| C1 | | 93 | I/O/Z | Вход/выход «порт С», разряд 1 |
| | MII_TXD1 | | O | Выход «передаваемый первый бит данных» |
| | CAN_TX1 | | O | Выход «линия передачи данных CAN1» |
| | MEM_ADDR2 | | O | Выход шины адреса внешней памяти, разряд 2 |
| C2 | | 94 | I/O/Z | Вход/выход «порт С», разряд 2 |
| | MII_TXD2 | | O | Выход «передаваемый второй бит данных» |
| | CAN_RX1 | | I | Вход «линия приема данных CAN1» |
| | MEM_ADDR3 | | O | Выход шины адреса внешней памяти, разряд 3 |
| C3 | | 95 | I/O/Z | Вход/выход «порт С», разряд 3 |
| | MII_TXD3 | | O | Выход «передаваемый третий бит данных» |
| | UART_TXD0 | | O | Выход «линия передачи данных UART0» |
| | MEM_ADDR4 | | O | Выход шины адреса внешней памяти, разряд 4 |
| C4 | | 96 | I/O/Z | Вход/выход «порт С», разряд 4 |
| | MII_TX_EN | | O | Выход «разрешение передачи» |
| | UART_RXD0 | | I | Вход «линия приема данных UART0» |
| | MEM_ADDR5 | | O | Выход шины адреса внешней памяти, разряд 5 |
| C5 | | 97 | I/O/Z | Вход/выход «порт С», разряд 5 |
| | MII_TX_ER | | O | Выход «ошибка передачи» |
| | UART_TXD2 | | O | Выход «линия передачи данных UART2» |
| | MEM_ADDR6 | | O | Выход шины адреса внешней памяти, разряд 6 |
| C6 | | 98 | I/O/Z | Вход/выход «порт С», разряд 6 |
| | MEM_DATA1 | | I/O | Вход/выход шины данных внешней памяти, разряд 1 |
| | UART_RXD1 | | I | Вход «линия приема данных UART1» |
| | SPI_TXD2 | | O | Выход «линия передачи данных SPI2» |
| C7 | | 99 | I/O/Z | Вход/выход «порт С», разряд 7 |
| | MEM_DATA2 | | I/O | Вход/выход шины данных внешней памяти, разряд 2 |
| | UART_TXD1 | | O | Выход «линия передачи данных UART1» |
| | PWM_TZ2 | | I | Вход «сигнал 2 аварии» |
| C8 | | 103 | I/O/Z | Вход/выход «порт С», разряд 8 |
| | MEM_DATA3 | | I/O | Вход/выход шины данных внешней памяти, разряд 3 |
| | SPI_FSS0 | | I | Вход «выбор ведомого устройства SPI0» |
| | PWM_TZ4 | | I | Вход «сигнал 4 аварии» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|------------|--------------|------------|--|
| C9 | | 104 | I/O/Z | Вход/выход «порт C», разряд 9 |
| | MEM_DATA4 | | I/O | Вход/выход шины данных внешней памяти, разряд 4 |
| | SPI_CLK0 | | I/O | Вход/выход «синхросигнал передачи SPI0» |
| | PWM_TZ5 | | I | Вход «сигнал 5 аварии» |
| C10 | | 105 | I/O/Z | Вход/выход «порт C», разряд 10 |
| | MEM_DATA5 | | I/O | Вход/выход шины данных внешней памяти, разряд 5 |
| | SPI_RXD0 | | I | Вход «линия приема данных SPI0» |
| | MII_TXCLK | | I | Вход «синхросигнал передачи» |
| C11 | | 106 | I/O/Z | Вход/выход «порт C», разряд 11 |
| | MEM_DATA6 | | I/O | Вход/выход шины данных внешней памяти, разряд 6 |
| | SPI_TXD0 | | O | Выход «линия передачи данных SPI0» |
| | MII_TXD0 | | O | Выход «передаваемый нулевой бит данных» |
| C12 | | 107 | I/O/Z | Вход/выход «порт C», разряд 12 |
| | MEM_DATA7 | | I/O | Вход/выход шины данных внешней памяти, разряд 7 |
| | TIMER_IN0 | | I | Вход «синхронизация таймера 0» |
| | MII_TXD1 | | O | Выход «передаваемый первый бит данных» |
| C13 | | 108 | I/O/Z | Вход/выход «порт C», разряд 13 |
| | MEM_DATA8 | | I/O | Вход/выход шины данных внешней памяти, разряд 8 |
| | TIMER_IN1 | | I | Вход «синхронизация таймера 1» |
| | MII_TXD2 | | O | Выход «передаваемый второй бит данных» |
| C14 | | 109 | I/O/Z | Вход/выход «порт C», разряд 14 |
| | MEM_DATA9 | | I/O | Вход/выход шины данных внешней памяти, разряд 9 |
| | TIMER_IN2 | | I | Вход «синхронизация таймера 2» |
| | MII_TXD3 | | O | Выход «передаваемый третий бит данных» |
| C15 | | 110 | I/O/Z | Вход/выход «порт C», разряд 15 |
| | MEM_DATA10 | | I/O | Вход/выход шины данных внешней памяти, разряд 10 |
| | NMI | | I | Вход «внешнее прерывание NMI» |
| | MII_TX_EN | | O | Выход «разрешение передачи» |
| D0 | | 111 | I/O/Z | Вход/выход «порт D», разряд 0 |
| | MII_CRS | | I | Вход «контроль носителя» |
| | UART_RXD2 | | I | Вход «линия приема данных UART2» |
| | MEM_ADDR7 | | O | Выход шины адреса внешней памяти, разряд 7 |
| D1 | | 112 | I/O/Z | Вход/выход «порт D», разряд 1 |
| | MII_COL | | I | Вход «обнаружение конфликта» |
| | UART_RXD3 | | I | Вход «линия приема данных UART3» |
| | MEM_ADDR8 | | O | Выход шины адреса внешней памяти, разряд 8 |
| D2 | | 113 | I/O/Z | Вход/выход «порт D», разряд 2 |
| | MII_MDC | | O | Выход «линия тактового сигнала управляющих данных» |
| | UART_TXD3 | | O | Выход «линия передачи данных UART3» |
| | MEM_ADDR9 | | O | Выход шины адреса внешней памяти, разряд 9 |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|------------|--------------|------------|--|
| D3 | | 114 | I/O/Z | Вход/выход «порт D», разряд 3 |
| | MII_MDIO | | I/O | Вход/выход «линия управляющих данных» |
| | – | | – | – |
| | MEM_ADDR10 | | O | Выход шины адреса внешней памяти, разряд 10 |
| D4 | | 115 | I/O/Z | Вход/выход «порт D», разряд 4 |
| | MII_RXCLK | | I | Вход «синхросигнал приема» |
| | – | | – | – |
| | MEM_ADDR11 | | O | Выход шины адреса внешней памяти, разряд 11 |
| D5 | | 116 | I/O/Z | Вход/выход «порт D», разряд 5 |
| | MII_RXD0 | | I | Вход «принимаемый нулевой бит данных» |
| | – | | – | – |
| | MEM_ADDR12 | | O | Выход шины адреса внешней памяти, разряд 12 |
| D6 | | 117 | I/O/Z | Вход/выход «порт D», разряд 6 |
| | MII_RXD1 | | I | Вход «принимаемый первый бит данных» |
| | – | | – | – |
| | MEM_DATA0 | | I/O | Вход/выход шины данных внешней памяти, разряд 0 |
| D7 | | 118 | I/O/Z | Вход/выход «порт D», разряд 7 |
| | MII_RXD2 | | I | Вход «принимаемый второй бит данных» |
| | – | | – | – |
| | MEM_DATA1 | | I/O | Вход/выход шины данных внешней памяти, разряд 1 |
| D8 | | 119 | I/O/Z | Вход/выход «порт D», разряд 8 |
| | MII_RXD3 | | I | Вход «принимаемый третий бит данных» |
| | – | | – | – |
| | MEM_DATA2 | | I/O | Вход/выход шины данных внешней памяти, разряд 2 |
| D9 | | 120 | I/O/Z | Вход/выход «порт D», разряд 9 |
| | MII_RX_DV | | I | Вход «валидность принимаемых данных» |
| | SCL1 | | I/O | Вход/выход «линия синхронизации I2C1» |
| | MEM_DATA3 | | I/O | Вход/выход шины данных внешней памяти, разряд 3 |
| D10 | | 129 | I/O/Z | Вход/выход «порт D», разряд 10 |
| | MII_RX_ER | | I | Вход «ошибка приема» |
| | SDA1 | | I/O | Вход/выход «линия данных I2C1» |
| | MEM_DATA4 | | I/O | Вход/выход шины данных внешней памяти, разряд 4 |
| D11 | | 128 | I/O/Z | Вход/выход «порт D», разряд 11 |
| | UART_TXD0 | | O | Выход «линия передачи данных UART0» (включен по умолчанию) |
| | CAN_TX0 | | O | Выход «линия передачи данных CAN0» |
| | CMP_OUT0 | | O | Выход «результат сравнения аналогового компаратора 0» |
| D12 | | 127 | I/O/Z | Вход/выход «порт D», разряд 12 |
| | MEM_DATA11 | | I/O | Вход/выход шины данных внешней памяти, разряд 11 |
| | UART_TXD2 | | O | Выход «линия передачи данных UART2» |
| | MII_TX_ER | | O | Выход «ошибка передачи» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|-------------|--------------|------------|---|
| D13 | | 126 | I/O/Z | Вход/выход «порт D», разряд 13 |
| | MEM_DATA12 | | I/O | Вход/выход шины данных внешней памяти, разряд 12 |
| | – | | – | – |
| | MII_CRS | | I | Вход «контроль носителя» |
| D14 | | 125 | I/O/Z | Вход/выход «порт D», разряд 14 |
| | MEM_DATA13 | | I/O | Вход/выход шины данных внешней памяти, разряд 13 |
| | – | | – | – |
| | MII_COL | | I | Вход «обнаружение конфликта» |
| D15 | | 124 | I/O/Z | Вход/выход «порт D», разряд 15 |
| | MEM_DATA14 | | I/O | Вход/выход шины данных внешней памяти, разряд 14 |
| | – | | – | – |
| | MII_MDC | | O | Выход «линия тактового сигнала управляющих данных» |
| E0 | | 130 | I/O/Z | Вход/выход «порт E», разряд 0 |
| | UART_RXD0 | | I | Вход «линия приема данных UART0» (включен по умолчанию) |
| | SCL0 | | I/O | Вход/выход «линия синхронизации I2C0» |
| | CMP_OUT1 | | O | Выход аналогового компаратора 1 |
| E1 | | 131 | I/O/Z | Вход/выход «порт E», разряд 1 |
| | JTAG_TRST | | I | Вход «сброс JTAG» (включен по умолчанию) |
| | SDA0 | | I/O | Вход/выход «линия данных I2C0» |
| | UART_TXD2 | | O | Выход «линия передачи данных UART2» |
| E2 | | 132 | I/O/Z | Вход/выход «порт E», разряд 2 |
| | CAN_RX0 | | I | Вход «линия приема данных CAN0» |
| | PWM_B5 | | O | Выход «линия В блока 5 ШИМ» |
| | MEM_DATA6 | | I/O | Вход/выход шины данных внешней памяти, разряд 6 |
| E3 | | 133 | I/O/Z | Вход/выход «порт E», разряд 3 |
| | NMI | | I | Вход «внешнее прерывание NMI» |
| | UART_RTS0 | | I | Вход «запрос на передачу UART0» |
| | MEM_DATA7 | | I/O | Вход/выход шины данных внешней памяти, разряд 7 |
| E4 | | 134 | I/O/Z | Вход/выход «порт E», разряд 4 |
| | QEP_A/XCLK0 | | I | Вход «тактовый сигнал КД0» |
| | CAP_PWM0 | | I/O | Вход/выход «внешнее событие/сигнал ШИМ блока захвата 0» |
| | TIMER_IN0 | | I | Вход «синхронизация таймера 0» |
| E5 | | 135 | I/O/Z | Вход/выход «порт E», разряд 5 |
| | QEP_B/XDIR0 | | I | Вход «сигнал направления вращения ротора для КД0» |
| | CAP_PWM1 | | I/O | Вход/выход «внешнее событие/сигнал ШИМ блока захвата 1» |
| | MEM_LB# | | O | Выход «разрешение для младшего байта» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|--------------|--------------|------------|---|
| E6 | | 136 | I/O/Z | Вход/выход «порт E», разряд 6 |
| | QEP_I0 | | I | Вход «индексный для КД0» |
| | CAP_PWM2 | | I/O | Вход/выход «внешнее событие/сигнал ШИМ блока захвата 2» |
| | MEM_DATA8 | | I/O | Вход/выход шины данных внешней памяти, разряд 8 |
| E7 | | 137 | I/O/Z | Вход/выход «порт E», разряд 7 |
| | USB_DRVVBUS | | O | Выход «управление шиной VBUS» |
| | QEP_S0 | | I | Вход «стробирование для КД0» |
| | PWM_TZ3 | | I | Вход «сигнал 3 аварии» |
| E8 | | 138 | I/O/Z | Вход/выход «порт E», разряд 8 |
| | PWM_TZ0 | | I | Вход «сигнал 0 аварии» |
| | UART_RI0 | | I | Вход «индикатор вызова UART0» |
| | SPI_FSS1 | | I | Вход «выбор ведомого устройства SPI1» |
| E9 | | 139 | I/O/Z | Вход/выход «порт E», разряд 9 |
| | PWM_TZ1 | | I | Вход «сигнал 1 аварии блока ШИМ» |
| | UART_DTR0 | | I | Вход «приемник данных готов UART0» |
| | SPI_CLK1 | | I/O | Вход/выход «синхросигнал передачи SPI1» |
| E10 | | 143 | I/O/Z | Вход/выход «порт E», разряд 10 |
| | JTAG_TDO/SWO | | O | Выход «данные JTAG» (включен по умолчанию) |
| | PWM_SYNCI | | I | Вход «внешний синхросигнал для блоков ШИМ» |
| | MEM_UB# | | O | Выход «разрешение для старшего байта» |
| E11 | | 144 | I/O/Z | Вход/выход «порт E», разряд 11 |
| | CMP_OUT2 | | O | Выход «результат сравнения аналогового компаратора 2» |
| | PWM_B8 | | O | Выход «линия В блока 8 ШИМ» |
| | MEM_DATA10 | | I/O | Вход/выход шины данных внешней памяти, разряд 10 |
| E12 | | 145 | I/O/Z | Вход/выход «порт E», разряд 12 |
| | MEM_DATA15 | | I/O | Вход/выход шины данных внешней памяти, разряд 15 |
| | PWM_TZ3 | | I | Вход «сигнал 3 аварии» |
| | MII_MDIO | | I/O | Вход/выход «линия управляющих данных» |
| E13 | | 146 | I/O/Z | Вход/выход «порт E», разряд 13 |
| | MEM_WE# | | O | Выход «сигнал записи во внешнюю память» |
| | PWM_TZ4 | | I | Вход «сигнал 4 аварии» |
| | MII_RXCLK | | I | Вход «синхросигнал приема» |
| E14 | | 147 | I/O/Z | Вход/выход «порт E», разряд 14 |
| | MEM_OE0# | | O | Выход «чтение внешнего устройства 0» |
| | PWM_TZ5 | | I | Вход «сигнал 5 аварии» |
| | MII_RXD0 | | I | Вход «принимаемый нулевой бит данных» |
| E15 | | 148 | I/O/Z | Вход/выход «порт E», разряд 15 |
| | MEM_OE1# | | O | Выход «чтение внешнего устройства 1» |
| | – | | – | – |
| | MII_RXD1 | | I | Вход «принимаемый первый бит данных» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|------------|--------------|------------|---|
| F0 | | 149 | I/O/Z | Вход/выход «порт F», разряд 0 |
| | PWM_B0 | | O | Выход «линия В блока 0 ШИМ» |
| | UART_DSR0 | | I | Вход «источник данных готов UART0» |
| | SPI_TXD1 | | O | Выход «линия передачи данных SPI1» |
| F1 | | 150 | I/O/Z | Вход/выход «порт F», разряд 1 |
| | CMP_OUT1 | | O | Выход «результат сравнения аналогового компаратора 1» |
| | UART_CTS0 | | I | Вход «готовность к приему UART0» |
| | MEM_DATA11 | | I/O | Вход/выход шины данных внешней памяти, разряд 11 |
| F2 | | 151 | I/O/Z | Вход/выход «порт F», разряд 2 |
| | PWM_B1 | | O | Выход «линия В блока 1 ШИМ» |
| | SPI_CLK3 | | I/O | Вход/выход «синхросигнал передачи SPI3» |
| | CAN_RX0 | | I | Вход «линия приема данных CAN0» |
| F3 | | 152 | I/O/Z | Вход/выход «порт F», разряд 3 |
| | PWM_B3 | | O | Выход «линия В блока 3 ШИМ» |
| | SPI_RXD2 | | I | Вход «линия приема данных SPI2» |
| | MEM_DATA14 | | I/O | Вход/выход шины данных внешней памяти, разряд 14 |
| F4 | | 153 | I/O/Z | Вход/выход «порт F», разряд 4 |
| | PWM_B2 | | O | Выход «линия В блока 2 ШИМ» |
| | SPI_TXD3 | | O | Выход «линия передачи данных SPI3» |
| | UART_RXD2 | | I | Вход «линия приема данных UART2» |
| F5 | | 154 | I/O/Z | Вход/выход «порт F», разряд 5 |
| | SCL0 | | I/O | Вход/выход «линия синхронизации I2C0» |
| | PWM_A7 | | O | Выход «линия А блока 7 ШИМ» |
| | MEM_DATA9 | | I/O | Вход/выход шины данных внешней памяти, разряд 9 |
| F6 | | 155 | I/O/Z | Вход/выход «порт F», разряд 6 |
| | MEM_CE0# | | O | Выход «сигнал 0 выбора внешнего устройства» |
| | CMP_OUT0 | | O | Выход «результат сравнения аналогового компаратора 0» |
| | MII_RXD2 | | I | Вход «принимаемый второй бит данных» |
| F7 | | 156 | I/O/Z | Вход/выход «порт F», разряд 7 |
| | MEM_CE1# | | O | Выход «сигнал 1 выбора внешнего устройства» |
| | CMP_OUT1 | | O | Выход «результат сравнения аналогового компаратора 1» |
| | MII_RXD3 | | I | Вход «принимаемый третий бит данных» |
| F8 | | 157 | I/O/Z | Вход/выход «порт F», разряд 8 |
| | MEM_LB# | | O | Выход «разрешение для младшего байта» |
| | CMP_OUT2 | | O | Выход «результат сравнения аналогового компаратора 2» |
| | MII_RX_DV | | I | Вход «валидность принимаемых данных» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|-----------|--------------|------------|---|
| F9 | | 158 | I/O/Z | Вход/выход «порт F», разряд 9 |
| | MEM_UB# | | O | Выход «разрешение для старшего байта» |
| | UART_CTS2 | | I | Вход «готовность к приему UART2» |
| | MII_RX_ER | | I | Вход «ошибка приема» |
| F10 | | 162 | I/O/Z | Вход/выход «порт F», разряд 10 |
| | UART_TXD2 | | O | Выход «линия передачи данных UART2» |
| | – | | – | – |
| | UART_CTS1 | | I | Вход «готовность к приему UART1» |
| F11 | | 163 | I/O/Z | Вход/выход «порт F», разряд 11 |
| | UART_RXD2 | | I | Вход «линия приема данных UART2» |
| | – | | – | – |
| | UART_DCD1 | | I | Вход «обнаружен информационный сигнал UART1» |
| F12 | | 164 | I/O/Z | Вход/выход «порт F», разряд 12 |
| | UART_TXD3 | | O | Выход «линия передачи данных UART3» |
| | UART_RI2 | | I | Вход «индикатор вызова UART2» |
| | UART_DSR1 | | I | Вход «источник данных готов UART1» |
| F13 | | 165 | I/O/Z | Вход/выход «порт F», разряд 13 |
| | UART_RXD3 | | I | Вход «линия приема данных UART3» |
| | UART_DTR2 | | I | Вход «приемник данных готов UART2» |
| | UART_RI1 | | I | Вход «индикатор вызова UART1» |
| F14 | | 166 | I/O/Z | Вход/выход «порт F», разряд 14 |
| | CAN_TX1 | | O | Выход «линия передачи данных CAN1» |
| | UART_RTS2 | | I | Вход «запрос на передачу UART2» |
| | UART_DTR1 | | I | Вход «приемник данных готов UART1» |
| F15 | | 167 | I/O/Z | Вход/выход «порт F», разряд 15 |
| | CAN_RX1 | | I | Вход «линия приема данных CAN1» |
| | UART_DCD2 | | I | Вход «обнаружен информационный сигнал UART2» |
| | UART_RTS1 | | I | Вход «запрос на передачу UART1» |
| G0 | | 168 | I/O/Z | Вход/выход «порт G», разряд 0 |
| | SDA0 | | I/O | Вход/выход «линия данных I2C0» |
| | PWM_B7 | | O | Выход «линия В блока 7 ШИМ» |
| | CAN_TX0 | | O | Выход «линия передачи данных CAN0» |
| G1 | | 169 | I/O/Z | Вход/выход «порт G», разряд 1 |
| | CMP_OUT0 | | O | Выход «результат сравнения аналогового компаратора 0» |
| | PWM_A8 | | O | Выход «линия А блока 8 ШИМ» |
| | QEP_S0 | | I | Вход «стробирование для КД0» |
| G2 | | 190 | I/O/Z | Вход/выход «порт G», разряд 2 |
| | PWM_A0 | | O | Выход «линия А блока 0 ШИМ» |
| | UART_DCD0 | | I | Вход «обнаружен информационный сигнал UART0» |
| | SPI_RXD1 | | I | Вход «линия приема данных SPI1» |
| G3 | | 191 | I/O/Z | Вход/выход «порт G», разряд 3 |
| | PWM_A1 | | O | Выход «линия А блока 1 ШИМ» |
| | SPI_FSS3 | | I | Вход «выбор ведомого устройства SPI3» |
| | CAN_TX1 | | O | Выход «линия передачи данных CAN1» |

Продолжение таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|-------------|--------------|------------|---|
| G4 | | 192 | I/O/Z | Вход/выход «порт G», разряд 4 |
| | PWM_A2 | | O | Выход «линия А блока 2 ШИМ» |
| | SPI_RXD3 | | I | Вход «линия приема данных SPI3» |
| | UART_TXD2 | | O | Выход «линия передачи данных UART2» |
| G5 | | 193 | I/O/Z | Вход/выход «порт G», разряд 5 |
| | PWM_A3 | | O | Выход «линия А блока 3 ШИМ» |
| | SPI_CLK2 | | I/O | Вход/выход «синхросигнал передачи SPI2» |
| | MEM_DATA13 | | I/O | Вход/выход шины данных внешней памяти, разряд 13 |
| G6 | | 194 | I/O/Z | Вход/выход «порт G», разряд 6 |
| | PWM_A4 | | O | Выход «линия А блока 4 ШИМ» |
| | SPI_TXD2 | | O | Выход «линия передачи данных SPI2» |
| | MEM_DATA15 | | I/O | Вход/выход шины данных внешней памяти, разряд 15 |
| G7 | | 195 | I/O/Z | Вход/выход «порт G», разряд 7 |
| | CAN_TX0 | | O | Выход «линия передачи данных CAN0» |
| | PWM_A5 | | O | Выход «линия А блока 5 ШИМ» |
| | MEM_DATA5 | | I/O | Вход/выход шины данных внешней памяти, разряд 5 |
| G8 | | 196 | I/O/Z | Вход/выход «порт G», разряд 8 |
| | TIMER_IN1 | | I | Вход «синхронизация таймера 1» |
| | UART_DSR2 | | I | Вход «источник данных готов UART2» |
| | CAN_RX1 | | O | Выход «линия приема данных CAN1» |
| G9 | | 197 | I/O/Z | Вход/выход «порт G», разряд 9 |
| | PWM_SYNCI | | I | Вход «внешний синхросигнал для блоков ШИМ » |
| | QEP_S1 | | I | Вход «стробирование для КД1» |
| | UART_RTS3 | | I | Вход «запрос на передачу UART3» |
| G10 | | 201 | I/O/Z | Вход/выход «порт G», разряд 10 |
| | CAP_PWM3 | | I/O | Вход/выход «внешнее событие/сигнал ШИМ блока захвата 3» |
| | QEP_A/XCLK1 | | I | Вход «тактовый сигнал КД1» |
| | — | | — | — |
| G11 | | 202 | I/O/Z | Вход/выход «порт G», разряд 11 |
| | CAP_PWM4 | | I/O | Вход/выход «внешнее событие/сигнал ШИМ блока захвата 4» |
| | QEP_B/XDIR1 | | I | Вход «сигнал направления вращения ротора для КД1» |
| | TIMER_IN1 | | I | Вход «синхронизация таймера 1» |
| G12 | | 203 | I/O/Z | Вход/выход «порт G», разряд 12 |
| | CAP_PWM5 | | I/O | Вход/выход «внешнее событие/сигнал ШИМ блока захвата 5» |
| | QEP_I1 | | I | Вход «индексный для КД1» |
| | TIMER_IN2 | | I | Вход «синхронизация таймера 2» |
| G13 | | 204 | I/O/Z | Вход/выход «порт G», разряд 13 |
| | — | | — | — |
| | PWM_TZ0 | | I | Вход «сигнал 0 аварии» |
| | PWM_B4 | | O | Выход «линия В блока 4 ШИМ» |

Окончание таблицы 2.1

| Обозначение вывода | | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--|-----------|--------------|------------|--|
| G14 | | 205 | I/O/Z | Вход/выход «порт G», разряд 14 |
| | – | | – | – |
| | PWM_TZ1 | | I | Вход «сигнал 1 аварии» |
| | PWM_A6 | | O | Выход «линия А блока 6 ШИМ» |
| G15 | | 206 | I/O/Z | Вход/выход «порт G», разряд 15 |
| | – | | – | – |
| | PWM_TZ2 | | I | Вход «сигнал 2 аварии» |
| | PWM_B6 | | O | Выход «линия В блока 6 ШИМ» |
| H0 | | 12 | I/O/Z | Вход/выход «порт H», разряд 0 |
| | – | | – | – |
| | PLL_DIV2 | | O | Выход «синхросигнал от PLL» |
| | PWM_A7 | | O | Выход «линия А блока 7 ШИМ» |
| H1 | | 13 | I/O/Z | Вход/выход «порт H», разряд 1 |
| | – | | – | – |
| | – | | – | – |
| | PWM_B7 | | O | Выход «линия В блока 7 ШИМ» |
| H2 | | 14 | I/O/Z | Вход/выход «порт H», разряд 2 |
| | PWM_B5 | | O | Выход «линия В блока 5 ШИМ» |
| | – | | – | – |
| | – | | – | – |
| H3 | | 18 | I/O/Z | Вход/выход «порт H», разряд 3 |
| | PWM_B4 | | O | Выход «линия В блока 4 ШИМ» |
| | – | | – | – |
| | UART_CTS3 | | I | Вход «готовность к приему UART3» |
| H4 | | 19 | I/O/Z | Вход/выход «порт H», разряд 4 |
| | PWM_A6 | | O | Выход «линия А блока 6 ШИМ» |
| | – | | – | – |
| | UART_DCD3 | | I | Вход «обнаружен информационный сигнал UART3» |
| H5 | | 20 | I/O/Z | Вход/выход «порт H», разряд 5 |
| | PWM_B6 | | O | Выход «линия В блока 6 ШИМ» |
| | – | | – | – |
| | UART_DSR3 | | I | Вход «источник данных готов UART3» |
| H6 | | 21 | I/O/Z | Вход/выход «порт H», разряд 6 |
| | PWM_A7 | | O | Выход «линия А блока 7 ШИМ» |
| | – | | – | – |
| | UART_RI3 | | I | Вход «индикатор вызова UART3» |
| H7 | | 22 | I/O/Z | Вход/выход «порт H», разряд 7 |
| | PWM_B7 | | O | Выход «линия В блока 7 ШИМ» |
| | – | | – | – |
| | UART_DTR3 | | I | Вход «приемник данных готов UART3» |
| Примечание – Выводы портов А – Н имеют схемы «Pull-up», подключаемые программно. | | | | |

Таблица 2.2 – Функциональное назначение выводов без альтернативных функций

| Обозначение вывода | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|--------------|------------|---|
| CH0_ADC0 | 172 | I | Вход «канал 0 АЦПО вход А» |
| CH1_ADC0 | 173 | I | Вход «канал 1 АЦПО вход В» |
| CH2_ADC1 | 174 | I | Вход «канал 2 АЦП1 вход А» |
| CH3_ADC1 | 175 | I | Вход «канал 3 АЦП1 вход В» |
| CH4_ADC2 | 178 | I | Вход «канал 4 АЦП2 вход А» |
| CH5_ADC2 | 179 | I | Вход «канал 5 АЦП2 вход В» |
| CH6_ADC3 | 180 | I | Вход «канал 6 АЦП3 вход А» |
| CH7_ADC3 | 181 | I | Вход «канал 7 АЦП3 вход В» |
| CH8_ADC4 | 184 | I | Вход «канал 8 АЦП4 вход А» |
| CH9_ADC4 | 185 | I | Вход «канал 9 АЦП4 вход В» |
| CH10_ADC5 | 186 | I | Вход «канал 10 АЦП5 вход А» |
| CH11_ADC5 | 187 | I | Вход «канал 11 АЦП5 вход В» |
| CH12_ADC6 | 75 | I | Вход «канал 12 АЦП6 вход А» |
| CH13_ADC6 | 74 | I | Вход «канал 13 АЦП6 вход В» |
| CH14_ADC7 | 77 | I | Вход «канал 14 АЦП7 вход А» |
| CH15_ADC7 | 76 | I | Вход «канал 15 АЦП7 вход В» |
| CH16_ADC8 | 81 | I | Вход «канал 16 АЦП8 вход А» |
| CH17_ADC8 | 80 | I | Вход «канал 17 АЦП8 вход В» |
| CH18_ADC9 | 83 | I | Вход «канал 18 АЦП9 вход А» |
| CH19_ADC9 | 82 | I | Вход «канал 19 АЦП9 вход В» |
| CH20_ADC10 | 87 | I | Вход «канал 20 АЦП10 вход А» |
| CH21_ADC10 | 86 | I | Вход «канал 21 АЦП10 вход В» |
| CH22_ADC11 | 88 | I | Вход «канал 22 АЦП11 вход А» |
| CH23_ADC11 | 89 | I | Вход «канал 23 АЦП11 вход В» |
| USB_DM | 48 | I | Вход «USB Dm» |
| USB_DP | 49 | I | Вход «USB Dp» |
| USB_ID | 50 | I | Вход «USB ID» |
| VBUS | 45 | I/O | Вход/выход монитора питания USB PHY |
| RST | 9 | I | Вход «внешний сброс» (к блоку POR) |
| EXTRST | 10 | O | Выход «сброс POR» (от блока POR) |
| CPE | 11 | I | Вход «выбор источника постоянного тактового сигнала» |
| VRES | 44 | – | Подключение внешнего резистора (8,2 кОм) |
| XI_RTC | 4 | I | Вход дополнительного тактового сигнала/ – Вывод для подключения кварцевого резонатора (32 кГц) |
| XO_RTC | 5 | – | Вывод для подключения кварцевого резонатора |
| XI_OSC | 207 | I | Вход основного тактового сигнала/ – Вывод для подключения кварцевого резонатора |
| XO_OSC | 208 | – | Вывод для подключения кварцевого резонатора |
| TAVDD_PLL | 1 | – | Аналоговое питание PLL (1,8 В) |
| TDVDD_PLL | 3 | – | Цифровое питание PLL (1,8 В) |
| TAVDD_ADC_0 | 177, 182 | – | Аналоговое питание модулей АЦПО – АЦП5 (1,8 В) |
| TDVDD_ADC_0 | 170, 189 | – | Цифровое питание модулей АЦПО – АЦП5 (1,8 В) |
| TAVDD_ADC_1 | 79, 84 | – | Аналоговое питание модулей АЦП6 – АЦП11 (1,8 В) |
| TDVDD_ADC_1 | 72, 91 | – | Цифровое питание модулей АЦП6 – АЦП11 (1,8 В) |
| TVDDL_USB | 51 | – | Питание USB PHY (1,8 В) |

Окончание таблицы 2.2

| Обозначение вывода | Номер вывода | Тип вывода | Функциональное назначение вывода |
|--------------------|-------------------------------------|------------|---|
| TVDDA_USB | 46 | – | Питание USB PHY (3,3 В) |
| TVBAT | 6 | – | Батарейное питание RTC (1,6 – 3,6) В |
| TVCC_POR | 8 | – | Питание POR (3,3 В) |
| VDD | 17, 37, 63, 102, 121, 140, 159, 198 | – | Питание ядра микросхемы (1,8 В) |
| VDDPST | 15, 35, 61, 100, 123, 142, 161, 200 | – | Питание буферов ввода-вывода микросхемы (3,3 В) |
| TAVDD_CMP | 27 | – | Питание аналогового компаратора (3,3 В) |
| DGND_AGND_PLL | 2 | – | Цифровая и аналоговая земля PLL |
| AGND_ADC_0 | 176, 183 | – | Аналоговая земля модулей АЦП0 – АЦП5 |
| DGND_ADC_0 | 171, 188 | – | Цифровая земля модулей АЦП0 – АЦП5 |
| AGND_ADC_1 | 78, 85 | – | Аналоговая земля модулей АЦП6 – АЦП11 |
| DGND_ADC_1 | 73, 90 | – | Цифровая земля модулей АЦП6 – АЦП11 |
| AGND_BAT, GND_POR | 7 | – | Земля RTC и блока POR |
| AGND_USB | 47 | – | Земля USB |
| DGNDL_USB | 52 | – | Земля USB PHY |
| TAGND_CMP | 28 | – | Земля аналогового компаратора |
| GND | 16, 36, 62, 101, 122, 141, 160, 199 | – | Земля |

Особенности системы питания

На плате:

- выводы питания TAVDD_PLL, TDVDD_PLL, TVDDL_USB могут быть объединены с выводами питания VDD;
- выводы питания TVCC_POR, TAVDD_CMP, TVDDA_USB могут быть объединены с выводами питания VDDPST;
- выводы питания TDVDD_ADC_0, TAVDD_ADC_0, TDVDD_ADC_1, TAVDD_ADC_1 могут быть объединены с выводами питания VDD (при этом должны быть приняты меры для снижения помех по питанию).

Питание блока RTC (вывод TVBAT) используется автоматически при отсутствии питания на выводе VDDPST.

2.2 Электрические параметры

Номинальное значение напряжения питания:

- по выводам VDDPST должно быть $(3,3 \pm 5\%)$ В;
- по выводам VDD должно быть $(1,8 \pm 5\%)$ В;
- по выводам TVDDA_USB, TVCC_POR, TAVDD_CMP должно быть $(3,3 \pm 5\%)$ В;
- по выводам TAVDD_PLL, TDVDD_PLL, TAVDD_ADC_0, TDVDD_ADC_0, TAVDD_ADC_1, TDVDD_ADC_1, TVDDL_USB должно быть $(1,8 \pm 5\%)$ В.

Значение суммарного максимального тока по выводам портов А – G не должно превышать 50 мА на каждый порт, по выводам порта H не должно превышать 25 мА.

Напряжение питания по выводу TVBAT должно быть (1,6 – 3,6) В.

Допустимая максимальная разность значений напряжений питания по выводам VDD, TAVDD_PLL, TDVDD_PLL, TAVDD_ADC_0, TDVDD_ADC_0, TAVDD_ADC_1, TDVDD_ADC_1, TVDDL_USB составляет 50 мВ.

Допустимая максимальная разность значений напряжений по выводам земли составляет 50 мВ.

Амплитудное значение пульсации напряжения питания должно быть не более 30 мВ.

Значения предельно допустимых электрических режимов эксплуатации в диапазоне рабочих температур среды соответствуют нормам, приведенным в таблице 2.3.

Таблица 2.3 – Значения предельно допустимых электрических режимов эксплуатации

| Наименование параметра режима, единица измерения | Буквенное обозначение параметра | Предельно допустимый режим | |
|---|---------------------------------------|-------------------------------|-----------------|
| | | не менее | не более |
| 1 Напряжение питания по выводам VDDPST, В | U _{CC1} | 3,14 | 3,46 |
| 2 Напряжение питания по выводам VDD, В | U _{CC2} | 1,71 | 1,89 |
| 3 Напряжение питания по выводу TAVDD_PLL, В | U _{CC3} | 1,71 | 1,89 |
| 4 Напряжение питания по выводу TDVDD_PLL, В | U _{CC4} | 1,71 | 1,89 |
| 5 Напряжение питания по выводам TAVDD_ADC_0, В | U _{CC5} | 1,71 | 1,89 |
| 6 Напряжение питания по выводам TDVDD_ADC_0, В | U _{CC6} | 1,71 | 1,89 |
| 7 Напряжение питания по выводам TAVDD_ADC_1, В | U _{CC7} | 1,71 | 1,89 |
| 8 Напряжение питания по выводам TDVDD_ADC_1, В | U _{CC8} | 1,71 | 1,89 |
| 9 Напряжение питания по выводу TVDDL_USB, В | U _{CC9} | 1,71 | 1,89 |
| 10 Напряжение питания по выводу TVDDA_USB, В | U _{CC10} | 3,14 | 3,46 |
| 11 Напряжение питания по выводу TVBAT, В | U _{CC11} | 1,6 | 3,6 |
| 12 Напряжение питания по выводу TVCC_POR, В | U _{CC12} | 3,14 | 3,46 |
| 13 Напряжение питания по выводу TAVDD_CMP, В | U _{CC13} | 3,14 | 3,46 |
| 14 Входное напряжение низкого уровня, В | U _{IL1} | -0,3 | 0,8 |
| 15 Входное напряжение высокого уровня, В | U _{IH1} | $0,7 \times U_{CC1}$ | $U_{CC1} + 0,3$ |
| 16 Входное напряжение низкого уровня по выводам тактового сигнала, В | U _{IL2} | 0 | 0,6 |
| 17 Входное напряжение высокого уровня по выводам тактового сигнала, В | U _{IH2} | $0,7 \times U_{CC1}$ | U_{CC1} |
| 18 Выходной ток низкого уровня, мА | I _{OL} | - | 6 |
| 19 Выходной ток высокого уровня, мА | I _{OH} | -6 | - |
| 20 Частота следования импульсов тактовых сигналов, МГц | f _{Cl} | 1 | 100 |
| 21 Длительность фронта и спада тактового сигнала, нс | t _{LH1} , t _{HL1} | - | 5 |
| 22 Длительность фронта и спада входных сигналов, кроме тактового, нс | t _{LH2} , t _{HL2} | - | 10 |
| 23 Емкость нагрузки, пФ | C _L | - | 40 |

Электрические параметры микросхем при приемке и поставке соответствуют нормам, приведенным в таблице 2.4.

Таблица 2.4 – Электрические параметры микросхем при приемке и поставке

| Наименование параметра, единица измерения, режим измерения | Буквенное обозначение параметра | Норма параметра | | Темпе- ратура среды, °С |
|---|---------------------------------------|--------------------|-------------|----------------------------------|
| | | не менее | не более | |
| 1 | 2 | 3 | 4 | 5 |
| 1 Выходное напряжение низкого уровня по выводам портов А – Н, В, U _{CC1} = 3,14 В, U _{CC2} = 1,8 В, I _{OL} = 6 мА | U _{OL} | - | 0,4 | -45 ± 3 25 ± 10 85 ± 3 |

Окончание таблицы 2.4

| 1 | 2 | 3 | 4 | 5 |
|---|------------|------|-----|------------------------------|
| 2 Выходное напряжение высокого уровня по выводам портов А – Н, В $U_{CC1} = 3,14 \text{ В}, U_{CC2} = 1,8 \text{ В}, I_{OH} = -6 \text{ мА}$ | U_{OH} | 2,4 | – | –45 ± 3 25 ± 10 85 ± 3 |
| 3 Ток утечки низкого уровня по выводам портов А – Н с отключенной схемой «Pull-up», мкА, $U_{CC1} = 3,46 \text{ В}, U_{CC2} = 1,8 \text{ В}, U_{IL} = 0 \text{ В}$ | I_{LL} | –5 | – | |
| 4 Ток утечки высокого уровня по выводам портов А – Н с отключенной схемой «Pull-up», мкА, $U_{CC1} = 3,46 \text{ В}, U_{CC2} = 1,8 \text{ В}, U_{IH} = 3,46 \text{ В}$ | I_{LH} | – | 5 | |
| 5 Входной ток низкого уровня по выводам портов А – Н с подключенной схемой «Pull-up», мкА, $U_{CC1} = 3,46 \text{ В}, U_{CC2} = 1,8 \text{ В}, U_{IL} = 0 \text{ В}$ | I_{IL} | –200 | –40 | |
| 6 Выходной ток низкого уровня по выводам портов А – Н, находящимся в третьем состоянии, мкА, $U_{CC1} = 3,46 \text{ В}, U_{CC2} = 1,8 \text{ В}, U_{OL} = 0 \text{ В}$ | I_{OZL} | –10 | – | |
| 7 Выходной ток высокого уровня по выводам портов А – Н, находящимся в третьем состоянии, мкА, $U_{CC1} = 3,46 \text{ В}, U_{CC2} = 1,8 \text{ В}, U_{OH} = 3,46 \text{ В}$ | I_{OZH} | – | 10 | |
| 8 Динамический ток потребления ядра, мА, $U_{CC1} = 3,46 \text{ В}, U_{CC2} = 1,89 \text{ В}, U_{CC3} = 1,89 \text{ В},$ $U_{CC4} = 1,89 \text{ В}, U_{CC5} = 1,89 \text{ В}, U_{CC6} = 1,89 \text{ В},$ $U_{CC7} = 1,89 \text{ В}, U_{CC8} = 1,89 \text{ В}, U_{CC9} = 1,89 \text{ В},$ $U_{CC10} = 3,46 \text{ В}, U_{CC11} = 3,6 \text{ В}, U_{CC12} = 3,46 \text{ В},$ $U_{CC13} = 3,46 \text{ В}, f_{Cl} = 100 \text{ МГц}$ | I_{OCC2} | – | 300 | |
| 9 Общие гармонические искажения канала АЦП в 12-битном режиме, дБ, $U_{CC1} = 3,3 \text{ В}, U_{CC2} = U_{CC5} = U_{CC6} = U_{CC7} = U_{CC8} = 1,8 \text{ В},$ $f_s = 1,3 \text{ МГц}, f_{IN} = 120 \text{ кГц}$ | THD1 | – | –65 | |
| 10 Отношение сигнал/(шум + искажения) в канале АЦП в 12-битном режиме, дБ, $U_{CC1} = 3,3 \text{ В}, U_{CC2} = U_{CC5} = U_{CC6} = U_{CC7} = U_{CC8} = 1,8 \text{ В},$ $f_s = 1,3 \text{ МГц}, f_{IN} = 120 \text{ кГц}$ | SINAD1 | 55 | – | |
| 11 Общие гармонические искажения канала АЦП в 10-битном режиме, дБ, $U_{CC1} = 3,3 \text{ В}, U_{CC2} = U_{CC5} = U_{CC6} = U_{CC7} = U_{CC8} = 1,8 \text{ В},$ $f_s = 2,3 \text{ МГц}, f_{IN} = 220 \text{ кГц}$ | THD2 | – | –60 | |
| 12 Отношение сигнал/(шум + искажения) в канале АЦП в 10-битном режиме, дБ, $U_{CC1} = 3,3 \text{ В}, U_{CC2} = U_{CC5} = U_{CC6} = U_{CC7} = U_{CC8} = 1,8 \text{ В},$ $f_s = 2,3 \text{ МГц}, f_{IN} = 220 \text{ кГц}$ | SINAD2 | 53 | – | |
| 13 Функциональный контроль, $U_{CC1} = (3,14; 3,46) \text{ В}, U_{CC2} = (1,71; 1,89) \text{ В},$ $U_{CC3} = (1,71; 1,89) \text{ В}, U_{CC4} = (1,71; 1,89) \text{ В},$ $U_{CC5} = (1,71; 1,89) \text{ В}, U_{CC6} = (1,71; 1,89) \text{ В},$ $U_{CC7} = (1,71; 1,89) \text{ В}, U_{CC8} = (1,71; 1,89) \text{ В},$ $U_{CC9} = (1,71; 1,89) \text{ В}, U_{CC10} = (3,14; 3,46) \text{ В},$ $U_{CC11} = (1,6; 3,6) \text{ В}, U_{CC12} = (3,14; 3,46) \text{ В},$ $U_{CC13} = (3,14; 3,46) \text{ В}, f_{Cl} = 100 \text{ МГц}$ | ФК | – | – | |
| Примечание – Параметры $I_{LL}, I_{LH}, I_{OZL}, I_{OZH}$ при температуре минус 45 °С не измеряются, а гарантируются нормами при температуре (25 ± 10) °С. | | | | |

3 Архитектура изделия

Микроконтроллер K1921BK01T структурно представляет собой мультистадийный RISC процессор. Ядро Cortex-M4F основано на архитектуре ARMv7-M и полностью реализует наборы команд Thumb и Thumb2.

Поддержка DSP-инструкций и наличие модуля операций с плавающей запятой существенно ускоряет обработку потоковых данных, что в свою очередь делает микроконтроллер весьма привлекательным для использования в системах управления и обработки информации.

Микроконтроллер способен параллельно выполнять четыре операции сложения/вычитания с 8-разрядными операндами или две операции сложения/вычитания с 16-разрядными операндами. Также реализовано умножение за один цикл, при этом для 16-разрядных чисел возможно параллельное исполнение двух операций. Из особенностей следует упомянуть аппаратное умножение 32-разрядных чисел за 1 цикл, а также деление 32-разрядных чисел, занимающее от 2 до 12 циклов.

В ядре предусмотрен механизм защиты памяти.

Блок коммутации микроконтроллера

Все устройства микроконтроллера соединены между собой через блок коммутации. На рисунке 3.1 приведена схема соединения основных и периферийных блоков микроконтроллера внутри блока коммутации.

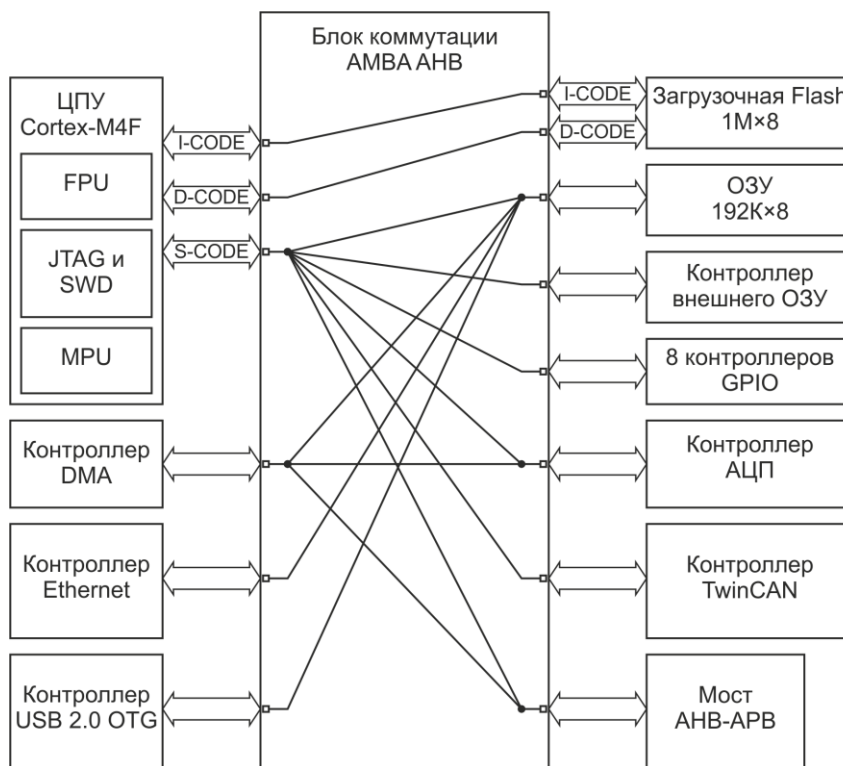


Рисунок 3.1 – Схема соединения блоков внутри блока коммутации

Помимо основной схемы блок коммутации может адресовать дополнительный адрес. Для коммутации по Remap-адресам необходимо установить управляющий бит блока коммутации.

В микроконтроллере для загрузки из внешней памяти могут изменяться адреса загрузочной флеш-памяти (шины D-CODE и I-CODE) и контроллера внешней памяти.

4 Система тактирования микроконтроллера

При включении питания микроконтроллер тактируется сигналом постоянной частоты. Источниками данного сигнала могут быть внешний кварцевый осциллятор и выходной сигнал RC-генератора внутреннего блока POR. Для работы кварцевого осциллятора снаружи должен быть подключен кварцевый резонатор (выводы XI_OSC и XO_OSC). Блок POR выдает тактовый сигнал частотой (3,5 – 6,5) МГц. Выбор источника сигнала постоянной частоты осуществляется входом CPE.

Сигнал постоянной тактовой частоты должен всегда присутствовать в микросхеме. Он используется для переключения системной частоты микроконтроллера, а также тактирует блок сторожевого таймера.

Управление тактовыми сигналами микроконтроллера осуществляет блок синхронизации. Данный блок осуществляет выбор источников тактового сигнала микроконтроллера и блоков периферии, управляет работой PLL, а также осуществляет переключение системной частоты.

4.1 Описание блока синхронизации

Блок синхронизации состоит из блока управления синхросигналом и периферии, включающей восемь источников тактового сигнала (см. рисунок 4.1). На выходе блока управления – сигнал системной тактовой частоты SysCLK.

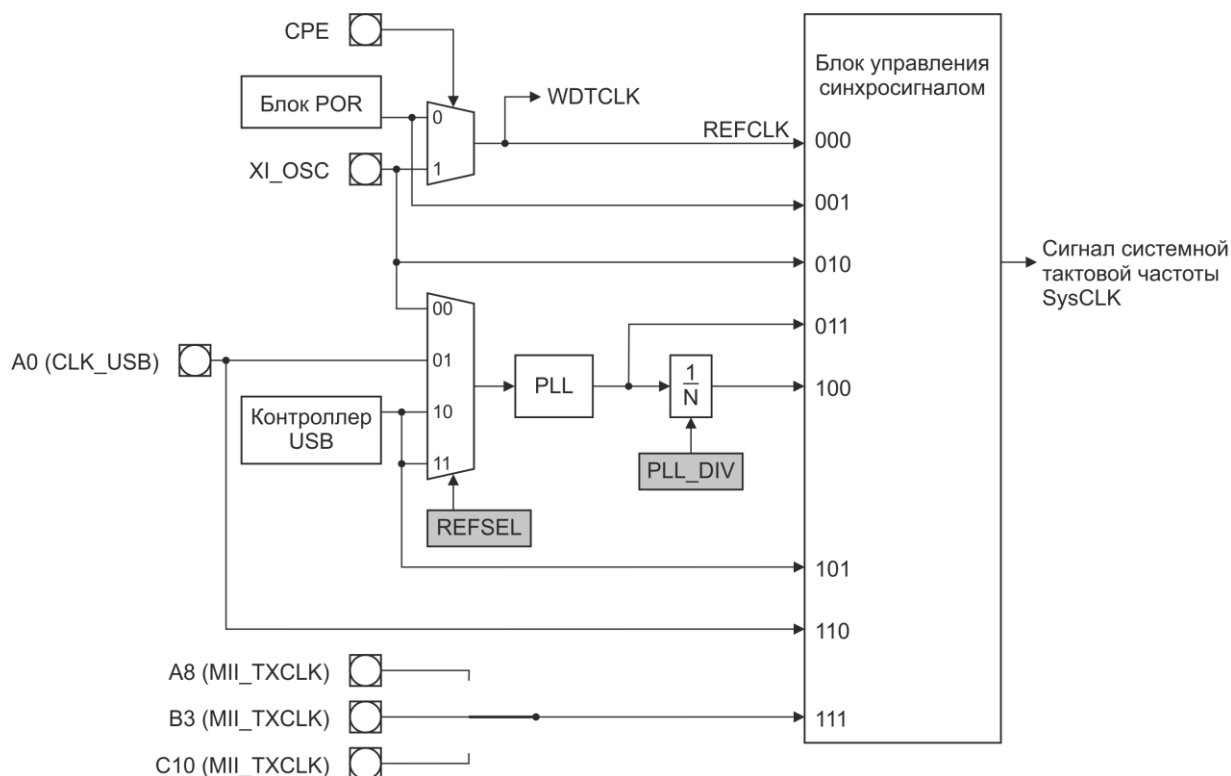


Рисунок 4.1 – Блок выдачи системной частоты

- В качестве источников синхросигнала SysCLK могут выступать:
- блок внутреннего RC-генератора POR (Power on Reset), на выходе которого сигнал CLKPOR;
 - сигнал внешнего осциллятора OSC_CLK с вывода XI_OSC микроконтроллера;
 - блок генератора PLL, на выходе которого сигнал CLK_PLL;
 - сигнал CLK_PLL, прошедший через программируемый делитель;

- сигнал CLK_USB тактирования контроллера USB с вывода A0 микроконтроллера (12 МГц);
- блок контроллера USB, на выходе которого сигнал UTMI_CLK60 (60 МГц);
- сигнал MI_TCLK тактирования контроллера Ethernet с вывода A8 или B3 микроконтроллера (25 МГц).

По умолчанию, источником сигнала тактовой частоты является сигнал REFCLK, который является сигналом на выходе блока POR, при условии, что на входе CPE логический ноль. В случае если на входе CPE логическая единица, сигнал REFCLK – это сигнал внешнего осциллятора.

Сигнал REFCLK также является сигналом тактирования сторожевого таймера.

Выбор источника для сигнала SysCLK осуществляется программированием регистра SYS_CLK.

Выбор источника тактового сигнала для PLL осуществляется полем REFSEL регистра PLL_CTRL. Частота выходного сигнала PLL может быть уменьшена с помощью делителя, значение которого задается полем PLL_DIV регистра PLL_CTRL. Коэффициент делителя N (см. рисунок 4.1) рассчитывается по формуле

$$N = 2 \times (\text{PLL_DIV} + 1). \quad (4.1)$$

Настройка PLL

Вычисление выходной частоты PLL производится по формуле

$$F_{\text{OUT}} = (F_{\text{IN}} \times N_F) / (N_R \times N_O), \quad (4.2)$$

где F_{IN} частота на входе PLL;

N_F – значение делителя обратной связи, которое зависит от состояния поля F_PLL регистра PLL_NF и равно $F_PLL + 2$;

N_R – значение опорного делителя, которое зависит от состояния поля R_PLL регистра PLL_NR и равно $R_PLL + 2$;

N_O – значение выходного делителя, которое зависит от состояния поля PL_OD регистра PLL_OD.

При расчете коэффициентов деления PLL должны выполняться следующие ограничения:

- $3,2 \text{ МГц} < F_{\text{IN}} < 150 \text{ МГц}$;
- $800 \text{ КГц} < F_{\text{ref}} < 8 \text{ МГц}$;
- $200 \text{ МГц} < F_{\text{vco}} < 500 \text{ МГц}$;
- частота фазового детектора

$$F_{\text{ref}} = F_{\text{IN}} / (2 \times N_R); \quad (4.3)$$

- частота

$$VCOF_{\text{vco}} = F_{\text{IN}} \times (N_F / N_R). \quad (4.4)$$

4.2 Алгоритм перехода на системную частоту

При включении питания в качестве системной частоты используется постоянная тактовая частота сигнала REFCLK.

Настройка PLL

Для перехода на системную частоту необходимо сначала настроить коэффициенты деления PLL, а также, если необходимо, коэффициент деления делителя $1/N$ выходного сигнала PLL.

Далее необходимо дождаться стабилизации значения частоты PLL (500 – 900) мкс, после чего выбрать источник системной частоты.

Переключение с одной частоты на другую

Если требуется переключение на другую системную частоту, т. е. смена источника системной частоты, то следует записать номер источника в поле SEL_SRC регистра SYS_CLK. Проверка заданного источника и переключение осуществляется блоком управления синхросигналом. После успешного переключения на новую частоту в поле CURR_SRC аппаратно запишется номер выбранного источника. Таким образом, совпадение значений полей SEL_SRC и CURR_SRC говорит о том, что выбранный источник системной частоты подключен.

В случае если заданный источник не функционирует или функционирует некорректно, переход на новую частоту не произойдет и значение поля CURR_SRC не изменится. Отличие значений полей SEL_SRC и CURR_SRC сигнализирует о том, что смены источника тактового сигнала не произошло.

4.3 Управление тактовыми сигналами периферии

Модули АЦП

Каждый модуль АЦП имеет независимое управление собственной тактовой частотой и собственный делитель $1/N$. Контроль синхронизации всех блоков АЦП осуществляют регистры ADC_CONTROL0, ADC_CONTROL1 и ADC_CONTROL2 (см. рисунок 4.2).

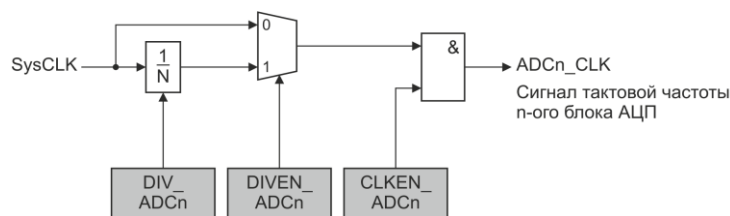


Рисунок 4.2 – Схема задания сигнала тактирования блока АЦП

Блоки UART

Каждый блок UART имеет независимое управление собственной тактовой частотой и собственный делитель $1/N$. Контроль синхронизации всех четырех блоков UART осуществляет регистр UART_CLK_CTRL (см. рисунок 4.3).

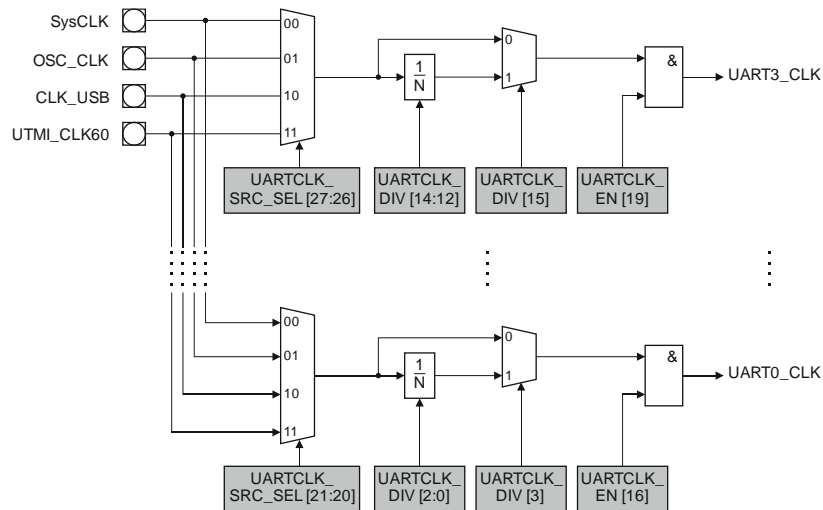


Рисунок 4.3 – Схема выбора сигналов тактирования блоков UART

Посредством поля `UARTCLK_SRC_SEL` выбирается источник тактового сигнала для каждого блока. В поле `UARTCLK_DIV` каждому блоку UART выделено по четыре бита. Старший из четырех бит определяет, как тактовый сигнал будет подан к блоку – напрямую или через делитель. Остальные три бита задают коэффициент деления.

Синхронизирование каждого блока UART может быть независимо включено/отключено установкой/сбросом соответствующего бита в поле `UARTCLK_EN`.

Блоки контроллеров SPI

Каждый блок SPI имеет независимое управление собственной тактовой частотой и собственный делитель $1/N$. Контроль синхронизации всех четырех блоков SPI осуществляет регистр `SPI_CLK_CTRL`. Схема выбора сигналов тактирования блоков SPI аналогична показанной на рисунке 4.3.

Посредством поля `SPICLK_SRC_SEL` выбирается источник тактового сигнала для каждого блока. В поле `SPICLK_DIV` каждому блоку SPI выделено по четыре бита. Старший из четырех бит определяет, как тактовый сигнал будет подан к блоку – напрямую или через делитель. Остальные три бита задают коэффициент деления.

Синхронизирование каждого блока SPI может быть независимо включено/отключено установкой/сбросом соответствующего бита в поле `SPICLK_EN`.

Блоки контроллеров I2C, сторожевого таймера, ШИМ, аналогового компаратора и квадратурных декодеров

Для каждого блока можно независимо включить/отключить системную частоту. Для этого следует установить/сбросить соответствующий блоку бит в регистре `APB_CLK_CTRL`.

Блок USB PHY

Схема выбора сигнала тактирования блока USB PHY показана на рисунке 4.4. Контроль синхронизации осуществляет регистр `USB_CTRL`.

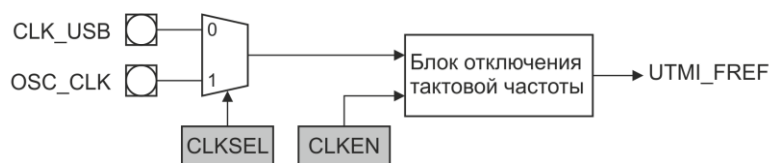


Рисунок 4.4 – Схема выбора сигнала тактирования блока USB PHY

В качестве источников опорной частоты для USB PHY могут использоваться два сигнала – OSC_CLK (выходной сигнал встроенного осциллятора) и CLK_USB (внешний входной синхросигнал). Выбор осуществляется битом CLK.

Частота сигнала на входе CLK_USB может иметь только два фиксированных значения – 12 и 24 МГц. Выбор между этими значениями осуществляется битом FSEL. Синхронизирование блока USB PHY может быть включено/отключено установкой/сбросом бита CLKEN.

5 Сброс микроконтроллера

Сброс может осуществляться как внешним сигналом с вывода RST, так и встроенным монитором питания POR. Активный уровень сигнала сброса – ноль.

Помимо внешнего сброса существуют два внутренних источника сброса – по запросу процессора и по переполнению сторожевого таймера.

На рисунке 5.1 представлена общая схема сброса микроконтроллера от внешнего источника. Входной сигнал низкого уровня с входа RST поступает на блок управления сбросом микроконтроллера. Генерируется выходной сигнал POR, который выводится на выход EXTRST микроконтроллера, а также поступает на блок часов реального времени (RTC).

Выходной сигнал POR осуществляет управление питанием блока RTC (отключает питание), поэтому при возникновении сигнала сброса блок RTC переходит на батарейное питание.

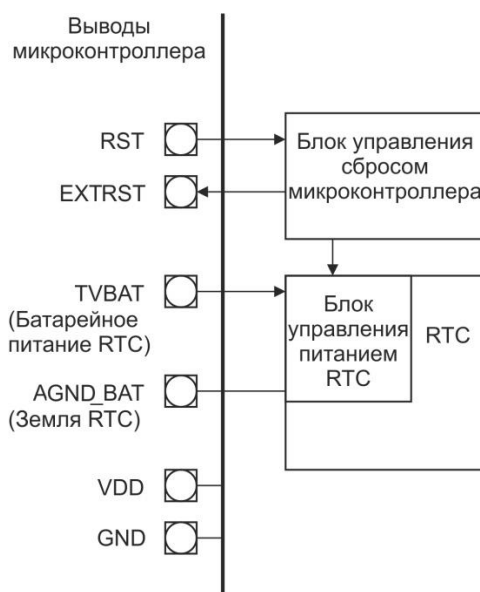


Рисунок 5.1 – Схема сброса микроконтроллера

Параметры монитора питания POR

На рисунках 5.2 и 5.3 показаны диаграммы сброса POR по внешнему сигналу с входа RST и при изменении питания. В таблицах 5.1 и 5.2 указаны динамические и электрические параметры монитора питания во время сброса.

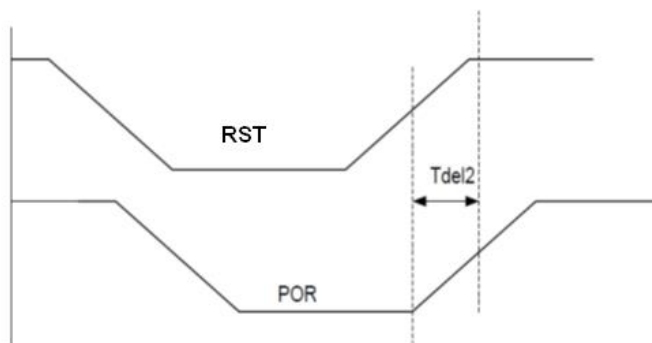


Рисунок 5.2 – Диаграмма сброса POR по внешнему сигналу с входа RST

Таблица 5.1 – Динамические параметры монитора питания

| Наименование параметра, единица измерения | Обозначение | Миним. | Номин. | Макс. |
|--|-------------|--------|--------|-------|
| Задержка срабатывания, мкс | Tdel1 | 270 | 410 | 590 |
| Задержка срабатывания при отключении внешнего сброса, нс | Tdel2 | – | – | 2 |
| Частота RC-генератора, МГц | – | 3,5 | 5 | 6,5 |

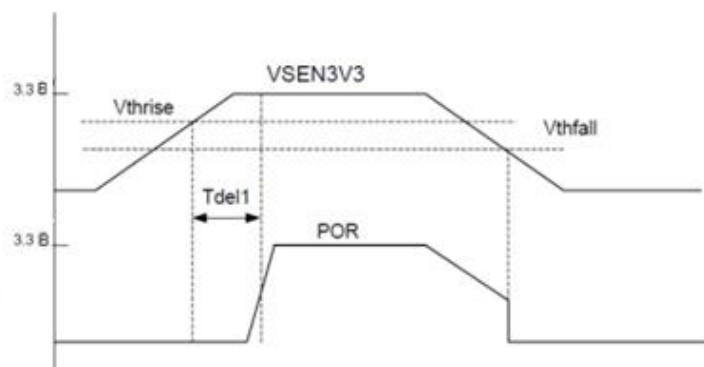


Рисунок 5.3 – Диаграмма сигнала сброса POR при изменении питания 3,3 В

Таблица 5.2 – Электрические параметры монитора питания

| Наименование параметра, единица измерения | Обозначение | Миним. | Номин. | Макс. | |
|---|------------------------------|---------|--------|-------|------|
| Уровень питания 3,3 В | Отключение схемы сброса, В | Vthrise | 2,50 | 2,60 | 2,70 |
| | Срабатывание схемы сброса, В | Vthfall | 2,33 | 2,42 | 2,52 |
| Уровень питания 1,8 В | Отключение схемы сброса, В | – | 1,62 | 1,70 | 1,80 |
| | Срабатывание схемы сброса, В | – | 1,54 | 1,60 | 1,68 |

6 Организация памяти

Память микроконтроллера имеет predetermined 32-разрядное адресное пространство с областями: программы (с защищенным блоком), данных (с информационным блоком), периферии, внешних (по отношению к контроллеру) ресурсов и внутренних ресурсов, жестко соединенных с процессором (см. рисунок 6.1).



Рисунок 6.1 – Организация памяти

Адресное пространство разбито на шесть областей (см. таблицу 6.1). При использовании внешней памяти как основной памяти программ и данных, адреса с 40000000h по 400FFFFFFh мапируются в адресное пространство внутренней памяти микроконтроллера в область 00000000h – 000FFFFFFh.

Таблица 6.1 – Организация памяти микроконтроллера

| Адресное пространство | Ремарк-адрес | Название блока | Описание |
|-------------------------|------------------------|------------------------------|---|
| 00000000h – 000FFFFFFh | 40000000h – 400FFFFFFh | Загрузочная флеш (I-Code) | Шина выборки инструкций |
| | | Загрузочная флеш (D-Code) | Шина данных, расположенных в коде программы |
| 20000000h – 2002FFFFFFh | – | Внутреннее ОЗУ 192 Кбайта | Шина данных, расположенных в области ОЗУ |
| 40000000h – 400FFFFFFh | 00000000h – 000FFFFFFh | Внешняя память | |
| 80000000h – BFFFFFFFh | – | SFR | Регистры управления периферийными блоками |
| E0000000h – E00FFFFFFh | – | Private Peripheral Bus | – |
| E0100000h – FFFFFFFFh | – | Vendor_SYS | Область системных регистров |

Системный сегмент System делится на две части.

Первая, объемом 1 Мбайт, занимает диапазон адресов E0000000h – E007FFFFFFh и зарезервирована для «личной» периферийной шины PPB (Private Peripheral Bus). PPB шина используется для выборки/записи данных и отладочного доступа – для периферии. Эта область рассматривается как строго упорядоченная память. Часть ее адресов (E000E000h – E000EFFFh) занимает пространство управления системой SCS, в котором находятся регистры блока управления системой SCB, контроллера прерываний NVIC, системного таймера SysTick, устройства защиты памяти MPU, средств отладки и другие. Байты регистров SCS всегда располагаются в порядке «младший – старший» независимо

от того, какой формат представления данных принят для остальной памяти. Доступ к большинству регистров возможен только полными двойными словами (32 бита), все исключения из этого правила оговариваются специально. Попытка кода обратиться к области PPB обычно вызывает прерывание по ошибке шины; исключениями являются возможность доступа к регистру STIR, если это разрешено соответствующим битом регистра CCR и доступность отладочных регистров.

Вторая часть является памятью устройств, занимает область E0100000h – FFFFFFFFh и является системным регионом ядра ARM (Vendor_SYS).

Регистры периферийных блоков микроконтроллера доступны в адресном пространстве 80000000h – BFFFFFFFh. Таблицы 6.2 и 6.3 представляют собой карту памяти с указанием областей памяти и блоков микроконтроллера, чьи регистры занимают эти области.

Таблица 6.2 – Регистры периферийных блоков

| Адресное пространство | Название блока | Описание | |
|-----------------------|----------------|----------------------------|-----------------|
| AHB периферийная шина | | | |
| 80000000h – 8000FFFFh | ADC | Блок АЦП | |
| 80010000h – 80010FFFh | GPIOA | Порт А | Контроллер GPIO |
| 80011000h – 80011FFFh | GPIOB | Порт В | |
| 80012000h – 80012FFFh | GPIOC | Порт С | |
| 80013000h – 80013FFFh | GPIOD | Порт D | |
| 80014000h – 80014FFFh | GPIOE | Порт E | |
| 80015000h – 80015FFFh | GPIOF | Порт F | |
| 80016000h – 80016FFFh | GPIOG | Порт G | |
| 80017000h – 80017FFFh | GPIOH | Порт H | |
| 80030000h – 8003FFFFh | COMMON_REG | Регистры общего назначения | |
| 80070000h – 8007FFFFh | CAN | Контроллер CAN | |
| 80080000h – 8008FFFFh | ETHERNET | Контроллер Ethernet | |
| 80090000h – 8009FFFFh | USB | Контроллер USB 2.0 OTG | |

Продолжение таблицы 6.2

| Адресное пространство | Название блока | Описание | |
|-----------------------|----------------|-------------------|----------------------------|
| Мост АHB-to-APB | | | |
| A0000000h – A0000FFFh | WDT | Сторожевой таймер | |
| A0001000h – A0001FFFh | I2C0 | Блок I2C_0 | Контроллеры I2C |
| A0002000h – A0002FFFh | I2C1 | Блок I2C_1 | |
| A0003000h – A0003FFFh | Timer0 | Timer_0 | Таймеры |
| A0004000h – A0004FFFh | Timer1 | Timer_1 | |
| A0005000h – A0005FFFh | Timer2 | Timer_2 | |
| A0006000h – A0006FFFh | DMA | Контроллер DMA | |
| A0007000h – A0007FFFh | UART0 | Блок UART_0 | Контроллеры UART |
| A0008000h – A0008FFFh | UART1 | Блок UART_1 | |
| A0009000h – A0009FFFh | UART2 | Блок UART_2 | |
| A000A000h – A000AFFh | UART3 | Блок UART_3 | |
| A000B000h – A000BFFFh | PWM0 | Блок PWM_0 | Контроллеры ШИМ |
| A000C000h – A000CFFFh | PWM1 | Блок PWM_1 | |
| A000D000h – A000DFFFh | PWM2 | Блок PWM_2 | |
| A000E000h – A000EFFFh | PWM3 | Блок PWM_3 | |
| A000F000h – A000FFFFh | PWM4 | Блок PWM_4 | |
| A0010000h – A0010FFFh | PWM5 | Блок PWM_5 | |
| A0011000h – A0011FFFh | PWM6 | Блок PWM_6 | |
| A0012000h – A0012FFFh | PWM7 | Блок PWM_7 | |
| A0013000h – A0013FFFh | PWM8 | Блок PWM_8 | |
| A0014000h – A0014FFFh | CAP0 | Блок CAP_0 | Контроллеры блоков захвата |
| A0015000h – A0015FFFh | CAP1 | Блок CAP_1 | |
| A0016000h – A0016FFFh | CAP2 | Блок CAP_2 | |

Окончание таблицы 6.2

| Адресное пространство | Название блока | Описание | |
|-----------------------|----------------|------------------------------|----------------------------|
| A0017000h – A0017FFFh | CAP3 | Блок CAP_3 | Контроллеры блоков захвата |
| A0018000h – A0018FFFh | CAP4 | Блок CAP_4 | |
| A0019000h – A0019FFFh | CAP5 | Блок CAP_5 | |
| A001A000h – A001AFFFh | QEP0 | Блок QEP_0 | Квадратурные декодеры |
| A001B000h – A001BFFFh | QEP1 | Блок QEP_1 | |
| A001C000h – A001CFFFh | BootFlash | Загрузочная флеш-память | |
| A001D000h – A001DFFFh | CMP | Аналоговый компаратор | |
| A001E000h – A001EFFFh | SPI0 | Блок SPI_0 | Контроллеры SPI |
| A001F000h – A001FFFFh | SPI1 | Блок SPI_1 | |
| A0020000h – A0020FFFh | SPI2 | Блок SPI_2 | |
| A0021000h – A0021FFFh | SPI3 | Блок SPI_3 | |
| A0022000h – A0022FFFh | UserFlash | Пользовательская флеш-память | |
| A0023000h – A0023FFFh | RTC | Часы реального времени | |
| A0024000h – BFFFFFFh | – | Зарезервировано | |

Таблица 6.3 – Регистры системного сегмента

| Адресное пространство | Название блока | Описание | |
|-----------------------|----------------|--|------------------------------|
| E0000000h – E0000FFFh | ITM | Блок контроля трассировки | Блок отладки DEBUG |
| E0001000h – E0001FFFh | DWT | Блок точек наблюдения и трассировки данных | |
| E0002000h – E0002FFFh | FPB | Блок флеш-патчей и точек останова | |
| E0003000h – E000DFFFh | – | Зарезервировано | |
| E000E000h – E000E00Fh | CPUID | Контроллер прерываний | Блок системного контроля SCS |
| E000E010h – E000E0FFh | SysTick | Системный таймер | |
| E000E100h – E000ECFFh | NVIC | Контроллер внешних прерываний | |

Окончание таблицы 6.3

| Адресное пространство | Название блока | Описание | |
|-----------------------|----------------|---|------------------------------|
| E000ED00h – E000ED8Ch | CPUID | Блок системного контроля SCB | Блок системного контроля SCS |
| E000ED90h – E000EDEFh | MPU | Блок защиты памяти | |
| E000EDF0h – E000EEFFh | CPUID | Блок отладки | |
| E000EF00h – E000EF30h | | | |
| E000EF34h – E000EF44h | SCB | Дополнительный блок FPU блока SCB | |
| E000EF48h – E000EF8Ch | CPUID | Блок программных прерываний | |
| E000EF90h – E000EFCFh | CPUID | Определяется реализацией (Implementation-defined) | |
| E000EFD0h – E000EFFFh | CPUID | Блок ID микроконтроллера | |
| E000F000h – E003FFFFh | – | Зарезервировано | |
| E0040000h – E0040FFFh | TPIU | Блок интерфейса точек трассировки TPIU | Блок отладки DEBUG |
| E0041000h – E00FFFFFh | PPB | «Личная » периферийная шина | |
| E0100000h – FFFFFFFFh | Vendor_SYS | Системная область | |

7 Флеш-память и внешняя память. Загрузка микроконтроллера

7.1 Загрузочная флеш-память

Загрузочная флеш-память (см. рисунок 7.1) может использоваться для хранения программ пользователя, хранения начального загрузчика, а также данных пользователя.

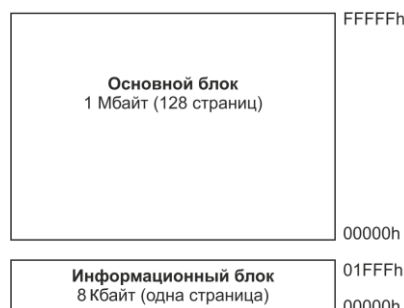


Рисунок 7.1 – Загрузочная флеш-память

Размер основной области загрузочной флеш-памяти составляет 1 Мбайт. Помимо основной области загрузочная флеш-память содержит особый защищенный блок, который может использоваться для хранения загрузчика, либо важных данных пользователя. Доступ в защищенный блок осуществляется дополнительными командами контроллера. Размер защищенного блока составляет 8 Кбайт.

Чтение флеш-памяти осуществляется через две шины АНВ – I-code (команды) и D-code (данные). На шине I-code работает функция предвыборки, на шине D-code предвыборка не осуществляется. Чтение D-code шины имеет приоритет.

На обеих шинах при попытке записи в любую область, чтении из несуществующей области, чтении во время, когда флеш занята (стирание, запись) транзакция проходит успешно с неопределенными данными на выходе.

Функции записи, страничного стирания, полного стирания реализуются через регистры, доступ к которым осуществляется посредством шины APB.

Операция предвыборки

При запросе данных на шине по адресу, по которому не осуществлялась предвыборка, выполняются следующие действия.

1 Сигнал готовности на шине устанавливается в ноль и задерживает транзакцию.

2 По запрашиваемому адресу считываются четыре двойных слова (128 бит) данных из флеш-памяти с обнуленными четырьмя младшими битами. Далее эти данные записываются во внутренний первый буфер.

3 Требуемое слово передается на шину АНВ, и сигнал готовности устанавливается в единицу.

4 Сразу после установки сигнала готовности, из флеш-памяти считываются четыре двойных слова данных по следующему адресу. Данные сохраняются во втором буфере. Если во время считывания этих данных появляются запросы по адресам, сохраненным в первом буфере, ответ возникает мгновенно, если по другим адресам, то готовность на шине устанавливается в ноль, и происходит ожидание завершения считывания во второй буфер и далее возврат к действию 2.

5 Если приходят запросы по адресам, сохраненным в первом буфере, ответ возникает мгновенно, если по адресам находящимся во втором буфере, ответ также возникает мгновенно. Далее переписывается первый буфер значением второго и считывается следующий адрес из флеш-памяти. Если приходят запросы по адресам не из первого и второго буферов, то возврат к действию 1.

Настройка загрузки из защищенного блока загрузочной флеш-памяти

Выбор области загрузки осуществляется битом BOOTFROM_IFB регистра INFO_WORD, находящегося в области пользовательской флеш-памяти.

Если бит сброшен, то чтение (по шинам I-Code и D-Code) в адресном пространстве от 0000h до 1FFFh происходит из защищенного блока загрузочной флеш, а чтение в адресном пространстве от 2000h до FFFFh происходит из основного блока (соответственно, начиная с адреса 2000h).

Если бит установлен (по умолчанию), то чтение во всем адресном пространстве от 00000h до FFFFFh происходит из основного блока загрузочной флеш (соответственно, начиная с адреса 00000h).

7.2 Пользовательская флеш-память

Пользовательская флеш-память (см. рисунок 7.2) может использоваться для хранения пользовательских настроек во время работы.



Рисунок 7.2 – Пользовательская флеш-память

Размер основной области пользовательской флеш-памяти составляет 64 Кбайта (256 страниц по 256 байт каждая). Основной блок расположен в пространстве с 0000h по FFFFh. Память доступна для чтения и записи через регистры: данных FMD, адреса FMA, команд FMC, статуса FCIS, маски прерываний FCIM и сброса статуса FCIS.

Примечание – Запись необходимо производить в предварительно очищенную ячейку памяти. Стирание памяти осуществляется полностью или постранично.

Помимо основной области пользовательская флеш-память содержит информационный блок объемом 512 байт (две страницы), в котором хранятся настройки доступа к страницам загрузочной и пользовательской флеш-памятей, а также параметры загрузки и отладки микроконтроллера. Информационный блок расположен в пространстве с 0000h по 01FFh. Доступ к информационному блоку управляется битами READ_IFB, WRITE_IFB и PAGEERASE_IFB регистра FMC.

Карта информационного блока показана на рисунке 7.3.

По адресу 000h располагается информационное слово INFO_WORD, состоящее из четырех конфигурационных байт, см. приложение А.

В диапазоне адресов с 040h по 050h расположены биты защиты страниц загрузочной флеш-памяти. Установка или сброс битов позволяет запрещать или разрешать доступ к страницам памяти для записи и стирания.

В диапазоне адресов с 080h по 0A0h расположены биты защиты страниц пользовательской флеш-памяти. Установка или сброс битов позволяет запрещать или разрешать доступ к страницам памяти для записи и стирания.

Остальной объем информационного блока – регистры пользователя.

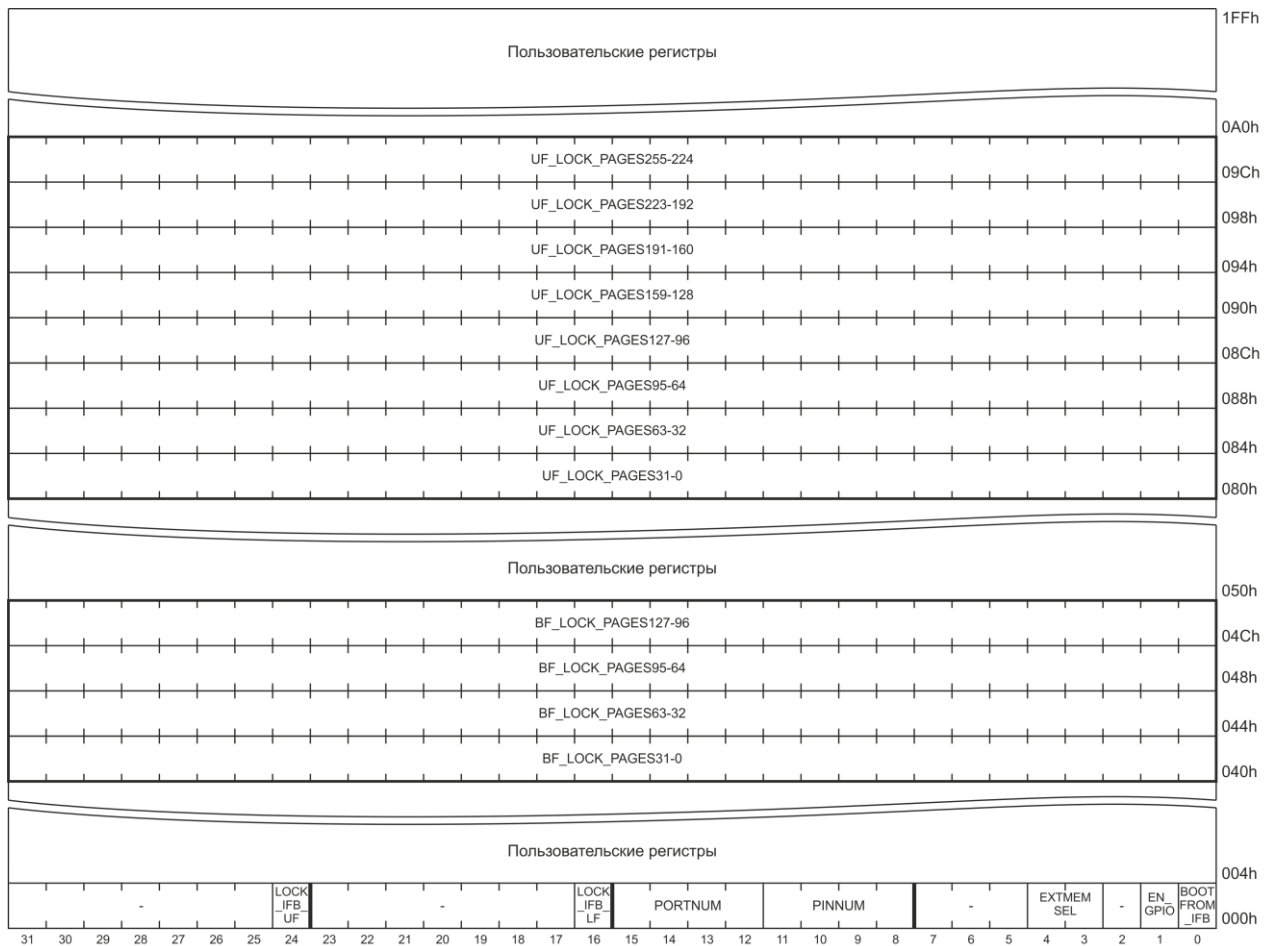


Рисунок 7.3 – Информационный блок пользовательской флеш-памяти

Логика работы контроллера пользовательской флеш

При старте автомат считывает информационный блок пользовательской флеш (52 байта) и сохраняет все в регистрах внутренней регистровой памяти (см. рисунок 7.4). В процессе работы контроллеры пользовательской и загрузочной флеш будут напрямую обращаться к этим регистрам, а не читать значения из флеш-памяти.

После выполнения этой операции схема выдает сигнал готовности, по которому микроконтроллер выводится из состояния сброса.

По окончании операции записи или стирания информационного блока флеш-памяти изменения будут записываться и в соответствующие регистры.

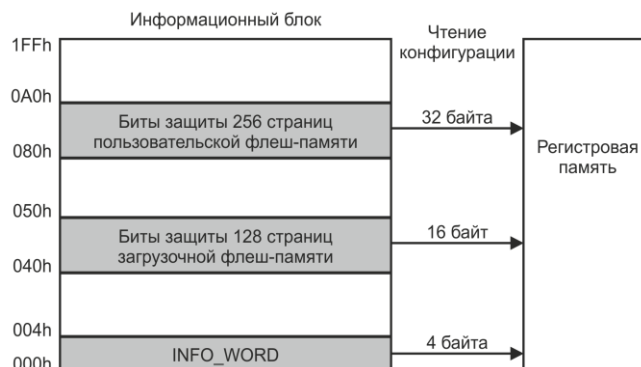


Рисунок 7.4 – Считывание конфигурации микроконтроллера

7.3 Алгоритм стирания всей флеш-памяти

1 Во время сброса микроконтроллера анализируется состояние вывода H2. Если вывод находится в состоянии логической единицы (подтянут к 3,3 В), то загрузочная и пользовательская флеш-памяти (включая защищенный и информационный блоки) переводятся в режим, в котором чтение запрещено (при чтении возвращаются нули).

2 Далее по отладочному интерфейсу (SWD или JTAG) должна быть подана команда записи значения 00000001h в регистр FLASH_FULL_ERASE, после чего будет активировано полное стирание.

Примечание – Если полное стирание не требуется во время сброса, на выводе H2 должен удерживаться логический ноль.

7.4 Временные параметры

На рисунках 7.5 – 7.8 показаны временные диаграммы циклов чтения, записи и стирания флеш-памяти. Эти диаграммы справедливы, как для загрузочной флеш-памяти, так и для пользовательской. На диаграммах присутствуют обозначения сигналов и временных параметров, описание которых приведено в таблицах 7.1 – 7.3, соответственно.

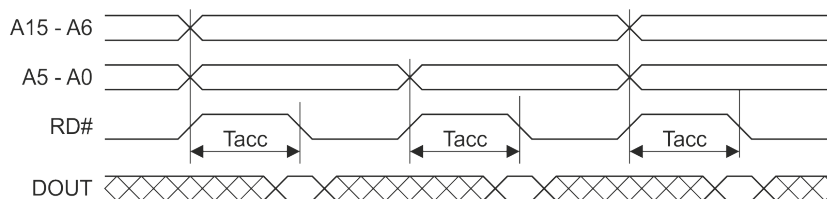


Рисунок 7.5 – Цикл чтения флеш-памяти

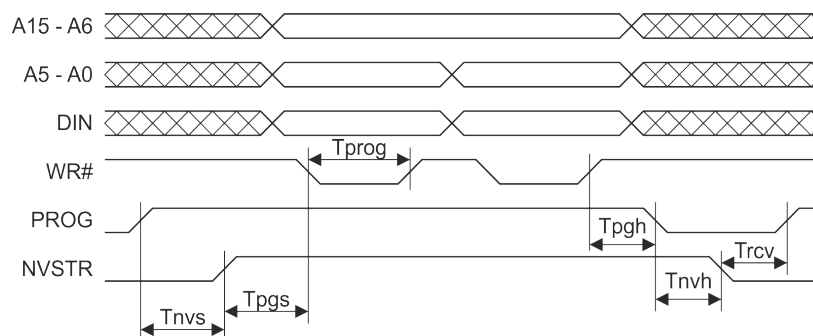


Рисунок 7.6 – Цикл записи флеш-памяти

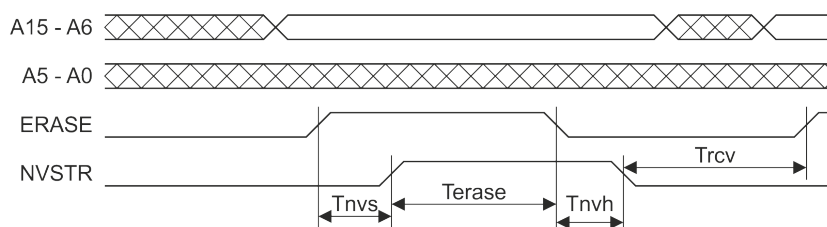


Рисунок 7.7 – Цикл стирания флеш-памяти

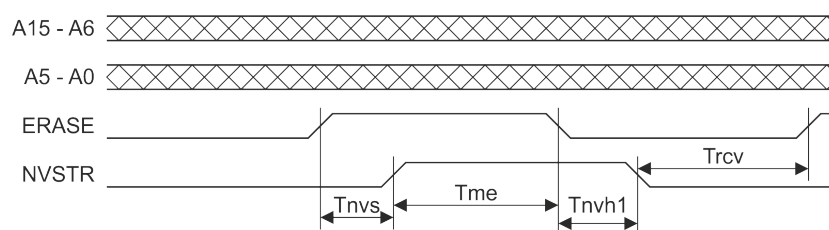


Рисунок 7.8 – Цикл массового стирания флеш-памяти

Таблица 7.1 – Сигналы, используемые при работе с флеш-памятью

| Название сигнала | Назначение |
|------------------|-------------------------------------|
| A15-A6, A5-A0 | Адреса |
| DOUT | Данные, читаемые из флеш-памяти |
| DIN | Данные, записываемые во флеш-память |
| RD# | Чтение |
| WR# | Запись |
| ERASE | Стирание |
| PROG | Разрешение программирования |
| NVSTR | Разрешение операции записи/стирания |

Таблица 7.2 – Параметры циклов обращения к флеш-памяти

| Параметр | Назначение параметра | Значение по умолчанию, мт |
|----------|---|---------------------------|
| Tacc | Задержка от установки адреса, до считывания данных из флеш-памяти | 4 |
| Tnvs | Задержка установки сигнала NVSTR после установки сигнала PROG | 500 |
| Tnvh | Задержка сброса сигнала NVSTR после сброса сигнала PROG | |
| Trcv | Задержка после сброса сигнала NVSTR до его установки в следующем цикле | 100 |
| Trpgs | Задержка сигнала записи после установки сигнала NVSTR | 1000 |
| Tprog | Длительность активного уровня сигнала записи | 2000 |
| Trpgh | Задержка сброса сигнала PROG после перехода сигнала записи в неактивное состояние | 2 |
| Terase | Длительность активного уровня сигнала стирания | 2000000 |
| Tme | Длительность активного уровня сигнала стирания в цикле массового стирания | |
| Tnvh1 | Задержка сброса сигнала NVSTR после сброса сигнала стирания | 10000 |

Таблица 7.3 – Значения параметра Tacc в зависимости от частоты

| Тактовая частота, не более, МГц | Задержка от установки адреса, до считывания данных из флеш-памяти, Tacc, мт |
|--|---|
| 100 | 6 |
| 80 | 5 |
| 60 | 4 |
| 48 | 3 |
| 30 | 2 |
| 15 | 1 |
| Примечание – Значение параметра после сброса равно 4 мт. | |

Регистры задания временных параметров для пользовательской и загрузочной флеш-памяти идентичны и отличаются только адресами. Название регистра и параметра совпадают. Для более полной информации следует обратиться к приложению А.

7.5 Контроллер внешней памяти

Контроллер внешней памяти позволяет осуществлять коммутацию процессора микроконтроллера с внешней памятью типа SRAM. Контроллер может работать в 8/16-разрядном режиме и согласовывает сигналы АНВ-шины с сигналами внешней памяти. На рисунке 7.9 показаны временные диаграммы циклов чтения и записи при работе с внешней памятью.

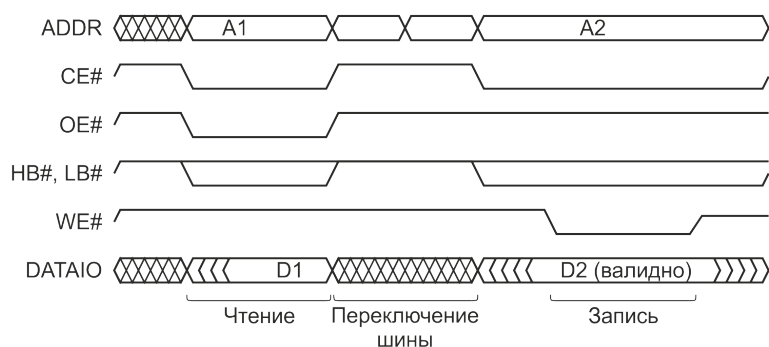


Рисунок 7.9 – Временная диаграмма работы контроллера внешней памяти (описание обозначений приведено в таблице 7.4)

Таблица 7.4 – Сигналы, используемые при работе с внешней памятью

| Название сигнала | Выходы микроконтроллера | Назначение |
|--|-------------------------|--|
| ADDR | MEM_ADDR | Адрес для чтения/записи передаваемый во внешнюю память |
| CE# | MEM_CE# | Сигнал выбора внешнего устройства |
| OE# | MEM_OE# | Сигнал (строб) разрешения чтения для внешнего устройства |
| HB#, LB# | MEM_UB#, MEM_LB# | Сигналы разрешения для старшего и младшего байт |
| WE# | MEM_WE# | Сигнал (строб) записи для внешнего устройства |
| DATAIO | MEM_DATA | Данные внешней памяти |
| Примечание – Сигналы с символом «#» имеют низкий активный уровень. | | |

Управление контроллером внешней памяти осуществляется посредством регистра EMCFG. Разрядность шины задается битом 8/16BIT.

В контроллере внешней памяти предусмотрен цикл переключения шины между циклами чтения и записи. Это реализовано в целях предотвращения кратковременных скачков тока в то время, когда процессор и внешнее устройство пытаются одновременно управлять шиной данных. Настройка длительности циклов чтения, записи и переключения осуществляется посредством полей READ, WRITEWS и RWWS. На рисунке 7.9 показан пример для случая, когда READ = WRITEWS = RWWS = 001h, т. е. длительность каждого цикла составляет два такта сигнала системной частоты.

Сигналы MEM_CE# и MEM_OE# задаются по маске в зависимости от адреса, задаваемого пользователем. Маска задается регистром EMCFG.

Для подключения внешней памяти можно использовать две группы выводов микроконтроллера. В одной группе используется первая альтернативная функция выводов, во второй – третья. Номер альтернативной функции задает поле EXTMEMSEL

информационного слова INFO_WORD. Два варианта подключения внешней памяти к микроконтроллеру отображены в таблицах 7.5 и 7.6.

Таблица 7.5 – Подключение внешней памяти к выводам микроконтроллера, использующих первую альтернативную функцию (EXTMEMSEL = 00b)

| Обозначение вывода | Номер вывода | Интерфейс | |
|--------------------|--------------|------------|---|
| E13 | 146 | MEM_WE# | Сигнал записи во внешнюю память |
| E14 | 147 | MEM_OE0# | Сигнал чтения для внешнего устройства 0 |
| F6 | 155 | MEM_CE0# | Сигнал выбора внешнего устройства 0 |
| E15 | 148 | MEM_OE1# | Сигнал чтения для внешнего устройства 1 |
| F7 | 156 | MEM_CE1# | Сигнал выбора внешнего устройства 1 |
| F9 | 158 | MEM_UB# | Сигналы разрешения для старшего байта |
| F8 | 157 | MEM_LB# | Сигналы разрешения для младшего байта |
| B14 | 70 | MEM_ADDR18 | Шина адреса |
| B13 | 69 | MEM_ADDR17 | |
| B12 | 68 | MEM_ADDR16 | |
| B11 | 67 | MEM_ADDR15 | |
| B10 | 66 | MEM_ADDR14 | |
| B9 | 65 | MEM_ADDR13 | |
| B8 | 64 | MEM_ADDR12 | |
| B7 | 60 | MEM_ADDR11 | |
| B6 | 59 | MEM_ADDR10 | |
| B5 | 58 | MEM_ADDR9 | |
| B4 | 57 | MEM_ADDR8 | |
| A15 | 43 | MEM_ADDR7 | |
| A14 | 42 | MEM_ADDR6 | |
| A13 | 41 | MEM_ADDR5 | |
| A12 | 40 | MEM_ADDR4 | |
| A11 | 39 | MEM_ADDR3 | |
| A10 | 38 | MEM_ADDR2 | |
| A9 | 34 | MEM_ADDR1 | |
| A8 | 33 | MEM_ADDR0 | |
| E12 | 145 | MEM_DATA15 | Шина данных |
| D15 | 124 | MEM_DATA14 | |
| D14 | 125 | MEM_DATA13 | |
| D13 | 126 | MEM_DATA12 | |
| D12 | 127 | MEM_DATA11 | |
| C15 | 110 | MEM_DATA10 | |
| C14 | 109 | MEM_DATA9 | |
| C13 | 108 | MEM_DATA8 | |
| C12 | 107 | MEM_DATA7 | |
| C11 | 106 | MEM_DATA6 | |
| C10 | 105 | MEM_DATA5 | |
| C9 | 104 | MEM_DATA4 | |
| C8 | 103 | MEM_DATA3 | |
| C7 | 99 | MEM_DATA2 | |
| C6 | 98 | MEM_DATA1 | |
| B15 | 71 | MEM_DATA0 | |

Таблица 7.6 – Подключение внешней памяти к выводам микроконтроллера, использующих третью альтернативную функцию (EXTMEMSEL = 10b)

| Обозначение вывода | Номер вывода | Интерфейс | | |
|--------------------|--------------|------------|---|-------------|
| B0 | 53 | MEM_WE# | Сигнал записи во внешнюю память | |
| A2 | 25 | MEM_OE0# | Сигнал чтения для внешнего устройства 0 | |
| B1 | 54 | MEM_CE0# | Сигнал выбора внешнего устройства 0 | |
| A3 | 26 | MEM_OE1# | Сигнал чтения для внешнего устройства 1 | |
| B2 | 55 | MEM_CE1# | Сигнал выбора внешнего устройства 1 | |
| E10 | 143 | MEM_UB# | Сигналы разрешения для старшего байта | |
| E5 | 135 | MEM_LB# | Сигналы разрешения для младшего байта | |
| D5 | 116 | MEM_ADDR12 | Шина адреса | |
| D4 | 115 | MEM_ADDR11 | | |
| D3 | 114 | MEM_ADDR10 | | |
| D2 | 113 | MEM_ADDR9 | | |
| D1 | 112 | MEM_ADDR8 | | |
| D0 | 111 | MEM_ADDR7 | | |
| C5 | 97 | MEM_ADDR6 | | |
| C4 | 96 | MEM_ADDR5 | | |
| C3 | 95 | MEM_ADDR4 | | |
| C2 | 94 | MEM_ADDR3 | | |
| C1 | 93 | MEM_ADDR2 | | |
| C0 | 92 | MEM_ADDR1 | | |
| B3 | 56 | MEM_ADDR0 | | |
| G6 | 194 | MEM_DATA15 | | Шина данных |
| F3 | 152 | MEM_DATA14 | | |
| G5 | 193 | MEM_DATA13 | | |
| A0 | 23 | MEM_DATA12 | | |
| F1 | 150 | MEM_DATA11 | | |
| E11 | 144 | MEM_DATA10 | | |
| F5 | 154 | MEM_DATA9 | Шина данных | |
| E6 | 136 | MEM_DATA8 | | |
| E3 | 133 | MEM_DATA7 | | |
| E2 | 132 | MEM_DATA6 | | |
| G7 | 195 | MEM_DATA5 | | |
| D10 | 129 | MEM_DATA4 | | |
| D9 | 120 | MEM_DATA3 | | |
| D8 | 119 | MEM_DATA2 | | |
| D7 | 118 | MEM_DATA1 | | |
| D6 | 117 | MEM_DATA0 | | |

7.6 Режимы загрузки микроконтроллера

После сброса микроконтроллера автоматически производится считывание загрузочных настроек из информационного блока пользовательской флеш-памяти. Далее, в зависимости от настроек производится загрузка из выбранного источника.

Микроконтроллер может загружаться из трех источников:

- основной области загрузочной флеш-памяти;
- защищенного блока загрузочной флеш-памяти;
- внешней памяти.

На рисунке 7.10 показаны три варианта заполнения адресного пространства при различных режимах загрузки микроконтроллера.



Рисунок 7.10 – Процесс мапирования при загрузке микроконтроллера (указанные диапазоны адресов соответствуют реальным объемам памяти микроконтроллера).

Карта адресного пространства:

- а) при загрузке из основной области загрузочной флеш-памяти;
- б) при загрузке из защищенной области загрузочной флеш-памяти;
- в) при загрузке из внешней памяти

Загрузка из основной области загрузочной флеш-памяти

В этом режиме мапирование не используется. Микроконтроллер загружается с адреса 00000000h основной области загрузочной флеш-памяти. Карта адресного пространства представлена на рисунке 7.10а) и в таблице 7.7.

Таблица 7.7 – Адресное пространство при загрузке из основной области загрузочной флеш-памяти

| Адресное пространство | Название блока | Описание |
|------------------------|---------------------------|---|
| 00000000h – 000FFFFFFh | Загрузочная флеш (I-Code) | Шина выборки инструкций |
| | Загрузочная флеш (D-Code) | Шина данных, расположенных в коде программы |
| 20000000h – 2002FFFFh | Внутреннее ОЗУ 192 Кбайта | Шина данных, расположенных в области ОЗУ |
| 40000000h – 400FFFFFFh | Внешняя память | |

Загрузка из защищенного блока

В этом режиме производится мапирование содержимого защищенного блока в адресное пространство загрузочной флеш-памяти. При обращении к диапазону 00000000h – 00001FFFh происходит считывание защищенного блока. Содержимое основного блока загрузочной флеш-памяти доступно в диапазоне 00002000h – 1FFFFFFFh.

Для загрузки из защищенного блока должен быть установлен бит BOOTFROM_IFB в информационном слове INFO_WORD пользовательской флеш-памяти.

Карта адресного пространства представлена на рисунке 7.10б) и в таблице 7.8.

Таблица 7.8 – Адресное пространство при загрузке из защищенной области загрузочной флеш-памяти

| Адресное пространство | Название блока | Описание |
|-----------------------|---|---|
| 00000000h – 00001FFFh | Защищенный блок загрузочной флеш (I-Code) | Шина выборки инструкций |
| | Защищенный блок загрузочной флеш (D-Code) | Шина данных, расположенных в коде программы |
| 00002000h – 000FFFFFh | Загрузочная флеш (I-Code) | Шина выборки инструкций |
| | Загрузочная флеш (D-Code) | Шина данных, расположенных в коде программы |
| 20000000h – 2002FFFFh | Внутреннее ОЗУ 192 Кбайта | Шина данных, расположенных в области ОЗУ |
| 40000000h – 400FFFFFh | Внешняя память | |

Загрузка из внешней памяти

В этом режиме производится мапирование содержимого внешней памяти в адресное пространство загрузочной флеш-памяти. При загрузке с адреса 00000000h по адрес 000FFFFFh происходит считывание внешней памяти. Содержимое основного блока загрузочной флеш-памяти доступно с адреса 40000000h.

Для загрузки из внешней памяти необходимо записать загрузочную информацию (конфигурационное слово INFO_WORD) со смещением 00h пользовательской флеш памяти со следующими параметрами:

- сбросить бит EN_GPIO;
- указать номер порта и вывода в PORTNUM и PINNUM, соответственно;
- задать номер альтернативной функции порта в поле EXTMEMSEL.

Карта адресного пространства выглядит, как показано на рисунке 7.10в) и в таблице 7.9.

Таблица 7.9 – Адресное пространство при загрузке из внешней памяти

| Адресное пространство | Название блока | Описание |
|-----------------------|---------------------------|---|
| 00000000h – 000FFFFFh | Внешняя память (I-Code) | Шина выборки инструкций |
| | Внешняя память (D-Code) | Шина данных, расположенных в коде программы |
| 20000000h – 2002FFFFh | Внутреннее ОЗУ 192 Кбайта | Шина данных, расположенных в области ОЗУ |
| 40000000h – 400FFFFFh | Загрузочная флеш | Шина данных |

8 Контроллер прямого доступа к памяти

Основные свойства и отличительные особенности:

- 24 канала DMA;
- каждый канал DMA имеет свои сигналы управления передачей данных и программируемый уровень приоритета;
- каждый уровень приоритета обрабатывается, исходя из уровня приоритета, определяемого номером канала DMA;
- поддержка различного типа передачи данных в пределах внутреннего ОЗУ: память – память, память – периферия, периферия – память;
- поддержка различных типов циклов DMA;
- поддержка передачи данных различной разрядности;
- каждому каналу DMA доступна первичная и альтернативная структура управляющих данных канала;
- все данные канала хранятся во внутреннем ОЗУ в структуре управляющих данных канала;
- разрядность данных приемника равна разрядности данных передатчика;
- количество передач в одном цикле DMA может программироваться от 1 до 1024;
- инкремент адреса передачи может быть больше, чем разрядность данных.

Аппаратные источники запросов каналов DMA указаны в таблице 8.1.

Таблица 8.1 – Аппаратные источники запросов каналов DMA

| Номер канала | Аппаратный источник запросов | Описание |
|--------------|------------------------------|--------------------------------|
| 0 | UART0_TX | Канал DMA от UART0 по передаче |
| 1 | UART1_TX | Канал DMA от UART1 по передаче |
| 2 | UART2_TX | Канал DMA от UART2 по передаче |
| 3 | UART3_TX | Канал DMA от UART3 по передаче |
| 4 | UART0_RX | Канал DMA от UART0 по приему |
| 5 | UART1_RX | Канал DMA от UART1 по приему |
| 6 | UART2_RX | Канал DMA от UART2 по приему |
| 7 | UART3_RX | Канал DMA от UART3 по приему |
| 8 | ADC_DMA0 | Канал 0 DMA контроллера АЦП |
| 9 | ADC_DMA1 | Канал 1 DMA контроллера АЦП |
| 10 | ADC_DMA2 | Канал 2 DMA контроллера АЦП |
| 11 | ADC_DMA3 | Канал 3 DMA контроллера АЦП |
| 12 | ADC_DMA4 | Канал 4 DMA контроллера АЦП |
| 13 | ADC_DMA5 | Канал 5 DMA контроллера АЦП |
| 14 | ADC_DMA6 | Канал 6 DMA контроллера АЦП |
| 15 | ADC_DMA7 | Канал 7 DMA контроллера АЦП |
| 16 | SPI0_TX | Канал DMA от SPI0 по передаче |
| 17 | SPI1_TX | Канал DMA от SPI1 по передаче |
| 18 | SPI2_TX | Канал DMA от SPI2 по передаче |
| 19 | SPI3_TX | Канал DMA от SPI3 по передаче |
| 20 | SPI0_RX | Канал DMA от SPI0 по приему |
| 21 | SPI1_RX | Канал DMA от SPI1 по приему |
| 22 | SPI2_RX | Канал DMA от SPI2 по приему |
| 23 | SPI3_RX | Канал DMA от SPI3 по приему |

8.1 Программное управление модулем DMA

Контроллер DMA выполняет передачи 8-, 16- и 32-разрядных данных. Разрядность данных источника и приемника должны быть одинаковыми.

Контроллер DMA позволяет управлять инкрементом адреса при чтении данных из источника и при записи данных в приемник. Инкремент адреса зависит от разрядности передаваемых данных: минимальная величина инкремента адреса всегда соответствует разрядности передаваемых данных; максимальная величина – одно слово. Контроллер DMA может быть настроен на работу с фиксированным адресом (например, для работы с FIFO).

Контроллер DMA имеет возможность обслуживать сигналы запроса на одиночный обмен SREQ и запроса на пакетный обмен BREQ блоков UART, SPI. Блок ADC генерирует только запросы на пакетный обмен BREQ.

Каждому каналу DMA соответствуют две структуры управляющих данных: первичная и альтернативная. В ОЗУ должна быть отведена область для хранения этих структур.

На рисунке 8.1 показана область памяти, необходимая контроллеру для структур управляющих данных каналов, при использовании всех 24 каналов и опциональной альтернативной структуры данных.

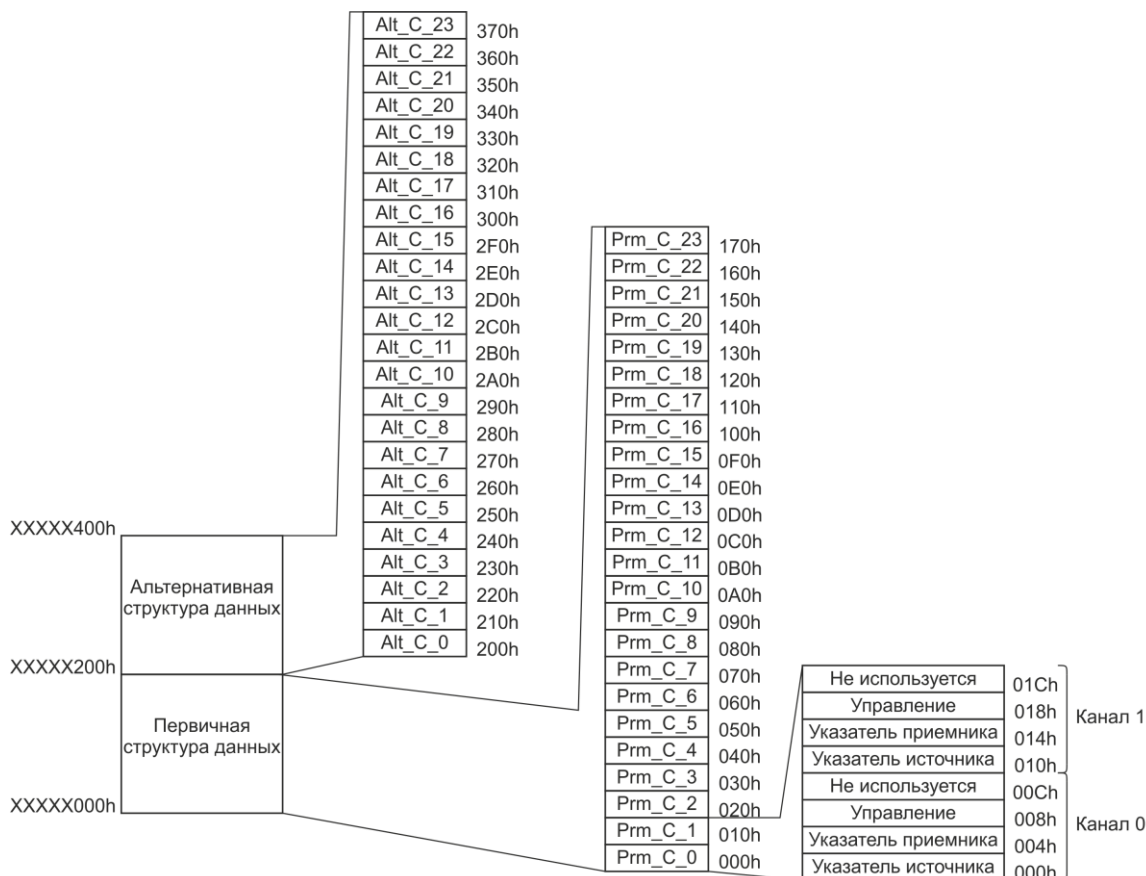


Рисунок 8.1 – Карта памяти для 24 каналов DMA, включая альтернативную структуру

Объем структуры, показанной на рисунке 8.1, составляет 768 байт. Контроллер использует младшие разряды адреса для доступа ко всем элементам структуры управляющих данных, и поэтому разрешенные значения базового адреса для первичной структуры управляющих данных XXXXX000h, XXXXX400h, XXXXX800h, XXXXXC00h.

Базовый адрес для первичной структуры управляющих данных возможно установить путем записи соответствующего значения в регистр CTRL_BASE_PTR.

В таблице 8.2 перечислены разряды адреса, обеспечивающие контроллеру доступ к различным элементам структуры управляющих данных.

Таблица 8.2 – Разряды адреса, используемые для доступа к управляющим данным 24 каналов

| Разряды адреса | | |
|----------------|---|--|
| | 9 | 8 7 6 5 4 3 2 1 0 |
| | S | CHNL EL |
| Обозначение | Биты | Действие |
| S | 9 | Выбор структуры управляющих данных |
| | | 0 Первичная |
| | | 1 Альтернативная |
| CHNL | 8-4 | Выбор канала. Допустимые значения 0h-18h |
| EL | 3-0 | Выбор управляющего элемента |
| | | 0h Указатель конца данных источника |
| | | 4h Указатель конца данных приемника |
| | | 8h Конфигурация структуры управляющих данных |
| Ch | Не используется. Контроллер не имеет доступа к этому адресу | |

Не обязательно вычислять базовый адрес альтернативной структуры управляющих данных, он вычисляется автоматически и помещается в регистр ALT_CTRL_BASE_PTR.

Любая из структур управляющих данных каждого канала состоит из двух указателей адреса (приемника и источника данных) и ячейки управления канала.

Управление канала (CHANNEL_CFG)

32-разрядная ячейка памяти, содержащая конфигурационную информацию для осуществления передач DMA (на рисунке 8.1 отмечена как «Управление»). В начале цикла DMA или начале 2^R передачи контроллер DMA считывает значение этой ячейки. После выполнения 2^R или N передач он сохраняет обновленное ее значение обратно в память. Структура регистра CHANNEL_CFG приведена в таблице 8.3.

Таблица 8.3 – Структура управляющих данных канала

| CHANNEL_CFG | | | | | | | | | | | | | | | |
|-------------|----|----------|----|---------|----|----------|----|---------------|----|----|----------------|------------|----|---------|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| dst_inc | | dst_size | | src_inc | | src_size | | dst_prot_ctrl | | | src_prot_ctrl | | | R_power | |
| R_power | | N-1 | | | | | | | | | next_use_burst | cycle_ctrl | | | |

Продолжение таблицы 8.3

| Поле | Биты | Описание | | | |
|---------------|--|--|-----------------------------------|------------------------|----------------------------|
| dst_inc | 31-30 | Шаг инкремента адреса приемника. Код, записанный в поле dst_inc, задает шаг, который в свою очередь зависит от разрядности данных источника | | | |
| | | Код | Разрядность данных источника | | |
| | | | Байт | Слово (16 бит) | Двойное слово (32 бита) |
| | | 00 | Шаг – байт | Зарезервировано | Зарезервировано |
| | | 01 | Шаг – слово (16 бит) | | Зарезервировано |
| | | 10 | Шаг – двойное слово (32 бита) | | |
| 11 | Нет инкремента. Адрес остается равным значению ячейки DST_DATA_END_PTR | | | | |
| dst_size | 29-28 | Разрядность данных приемника. Значение этого поля должно быть равно значению поля src_size Примечание – Если контроллер обнаруживает неравные значения этих полей, он при ближайшем обновлении поля N–1 устанавливает значение поля dst_size, равное src_size. | | | |
| src_inc | 27-26 | Шаг инкремента адреса источника. Код, записанный в поле, src_inc задает шаг, который в свою очередь зависит от разрядности данных источника | | | |
| | | Код | Разрядность данных источника | | |
| | | | Байт | Слово (16 бит) | Двойное слово (32 бита) |
| | | 00 | Шаг – байт | Зарезервировано | Зарезервировано |
| | | 01 | Шаг – слово (16 бит) | | Зарезервировано |
| | | 10 | Шаг – двойное слово (32 бита) | | |
| 11 | Нет инкремента. Адрес остается равным значению ячейки SRC_DATA_END_PTR | | | | |
| src_size | 25-24 | Разрядность данных источника | | | |
| | | 00 | Байт | | |
| | | 01 | Слово (16 бит) | | |
| | | 10 | Двойное слово (32 бита) | | |
| | | 11 | Зарезервировано. Не использовать! | | |
| dst_prot_ctrl | 23-21 | Задаёт параметры защиты шины АНВ-Lite при записи данных в приемник | | | |
| | | | Биты поля dst_prot_ctrl | | |
| | | | 23 | 22 | 21 |
| | | 0 | Доступ не кэшируется | Доступ не буферизуется | Доступ непривилегированный |
| 1 | Доступ кэшируется | Доступ буферизуется | Доступ привилегированный | | |
| src_prot_ctrl | 20-18 | Задаёт параметры защиты шины АНВ-Lite при чтении данных из источника | | | |
| | | | Биты поля src_prot_ctrl | | |
| | | | 20 | 19 | 18 |
| | | 0 | Доступ не кэшируется | Доступ не буферизуется | Доступ непривилегированный |
| 1 | Доступ кэшируется | Доступ буферизуется | Доступ привилегированный | | |

Продолжение таблицы 8.3

| Поле | Биты | Описание | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------|---|---|--------------------|------|---|------|---|-----|-----|------|------|----|----|----|----|----|----|----|-----|----|-----|----|-----|---------|---|
| R_power | 17-14 | <p>Параметр R. Задаёт количество передач канала DMA до выполнения контроллером процедуры арбитража (переарбитражи). Количество передач равно 2^R</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>Код</th> <th>Количество передач</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>1 (арбитраж производится после каждой передачи DMA)</td> </tr> <tr> <td>1h</td> <td>2</td> </tr> <tr> <td>2h</td> <td>4</td> </tr> <tr> <td>3h</td> <td>8</td> </tr> <tr> <td>4h</td> <td>16</td> </tr> <tr> <td>5h</td> <td>32</td> </tr> <tr> <td>6h</td> <td>64</td> </tr> <tr> <td>7h</td> <td>128</td> </tr> <tr> <td>8h</td> <td>256</td> </tr> <tr> <td>9h</td> <td>512</td> </tr> <tr> <td>Ah – Fh</td> <td>1024 (арбитраж не производится, так как максимальное количество передач DMA равно 1024)</td> </tr> </tbody> </table> | Код | Количество передач | 0h | 1 (арбитраж производится после каждой передачи DMA) | 1h | 2 | 2h | 4 | 3h | 8 | 4h | 16 | 5h | 32 | 6h | 64 | 7h | 128 | 8h | 256 | 9h | 512 | Ah – Fh | 1024 (арбитраж не производится, так как максимальное количество передач DMA равно 1024) |
| | | Код | Количество передач | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0h | 1 (арбитраж производится после каждой передачи DMA) | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1h | 2 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 2h | 4 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 3h | 8 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 4h | 16 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 5h | 32 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 6h | 64 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 7h | 128 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 8h | 256 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 9h | 512 | | | | | | | | | | | | | | | | | | | | | | | |
| | | Ah – Fh | 1024 (арбитраж не производится, так как максимальное количество передач DMA равно 1024) | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Примечание – Необходимо с осторожностью устанавливать большие значения R для низкоприоритетных каналов, так как это может привести к невозможности обслуживать запросы по высокоприоритетным каналам.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | |
| N-1 | 13-4 | <p>Перед выполнением цикла DMA эти разряды указывают общее количество передач DMA, из которых состоит цикл</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>Код</th> <th>Количество передач</th> </tr> </thead> <tbody> <tr> <td>000h</td> <td>1</td> </tr> <tr> <td>001h</td> <td>2</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>3FFh</td> <td>1024</td> </tr> </tbody> </table> | Код | Количество передач | 000h | 1 | 001h | 2 | ... | ... | 3FFh | 1024 | | | | | | | | | | | | | | |
| | | Код | Количество передач | | | | | | | | | | | | | | | | | | | | | | | |
| | | 000h | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| | | 001h | 2 | | | | | | | | | | | | | | | | | | | | | | | |
| | | ... | ... | | | | | | | | | | | | | | | | | | | | | | | |
| 3FFh | 1024 | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Примечание – Контроллер обновит это поле перед тем, как произвести процесс арбитража. Это позволяет контроллеру хранить количество оставшихся передач DMA до завершения всего цикла DMA.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | |
| next_use_burst | 3 | <p>Контролирует установку соответствующий каналу бита в регистре CHNL_USEBURST_SET, если контроллер работает в периферийном режиме «разборка-сборка» и завершает цикл DMA, используя альтернативные управляющие данные</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <p>Примечание – Перед завершением цикла DMA, использующего альтернативные управляющие данные, контроллер сбрасывает соответствующий каналу бит в регистре CHNL_USEBURST_SET, если количество оставшихся передач DMA меньше, чем 2^R. Программирование бита next_use_burst определяет, будет ли контроллер дополнительно переопределять состояние бита в регистре CHNL_USEBURST_SET.</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <p>Если контроллер выполняет цикл DMA в периферийном режиме «разборка-сборка», то после окончания цикла, использующего альтернативные управляющие данные, дальнейшие действия будут зависеть от состояния бита next_use_burst.</p> | | | | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы 8.3

| Поле | Биты | Описание | |
|----------------|------|---|---|
| next_use_burst | 3 | 0 | Контроллер не изменяет значение соответствующего каналу бита в регистре CHNL_USEBURST_SET. Если бит C_i в CHNL_USEBURST_SET сброшен, то при выполнении циклов DMA с использованием альтернативных управляющих данных контроллер отвечает, как на запросы BREQ, так и запросы SREQ от периферии |
| | | 1 | Контроллер изменяет значение соответствующего каналу бита в регистре CHNL_USEBURST_SET, а именно – устанавливает бит. Поэтому для оставшихся циклов DMA с использованием альтернативных управляющих данных контроллер реагирует только на запросы BREQ от периферии |
| cycle_ctrl | 2-0 | Поле задания типа цикла DMA | |
| | | 000b | Недействительный. Структура управляющих данных канала в запрещенном состоянии. |
| | | 001b | Основной |
| | | 010b | Авто-запрос |
| | | 011b | «Пинг-понг» |
| | | 100b | Работа с памятью в режиме «разборка-сборка» с использованием первичных управляющих данных канала |
| | | 101b | Работа с памятью в режиме «разборка-сборка» с использованием альтернативных управляющих данных канала |
| | | 110b | Работа с периферией в режиме «разборка-сборка» с использованием первичных управляющих данных канала |
| | | 111b | Работа с периферией в режиме «разборка-сборка» с использованием альтернативных управляющих данных канала |
| | | Примечание – После завершения всего цикла передач DMA, контроллер устанавливает значение поля cycle_ctrl в 000b, переводя тем самым тип цикла в «недействительный». Это позволяет избежать повторения выполненной передачи DMA. | |

Указатель конца данных источника (SRC_DATA_END_PTR) и указатель конца данных приемника (DST_DATA_END_PTR)

32-разрядные ячейки памяти, которые содержат адрес месторасположения конца данных источника и приемника, соответственно. Перед тем, как контроллер выполнит передачу DMA, необходимо определить их значения. Контроллер считывает значения этих областей перед началом 2^R передач DMA.

Для вычисления адреса источника передачи DMA, контроллер выполняет сдвиг влево значения $N-1$ на количество разрядов, соответствующее полю src_inc, и затем вычитает получившееся значение от значения SRC_DATA_END_PTR.

Подобным образом вычисляется начальный адрес приемника передачи и контроллер DMA выполняет сдвиг влево значения $N-1$ на количество разрядов, соответствующее полю dst_inc, и затем вычитает получившееся значение от значения DST_DATA_END_PTR.

8.2 Правила обмена данными

Следует избегать адресации к зарезервированным или неиспользованным адресам, так как это может привести к непредсказуемым результатам.

Необходимо заполнять неиспользуемые или зарезервированные разряды регистров нулями при записи и игнорировать значения таких разрядов при считывании.

Системный сброс или сброс по установке питания сбрасывает все регистры в состояние 00000000h, если не указано иное.

Контроллер DMA использует правила обмена данными (см. таблицу 8.4), при соблюдении следующих условий:

- канал DMA включен (установлены, соответствующие каналу, биты в регистре CHNL_ENABLE_SET и MASTER_ENBL в регистре CFG);
- запросы канала не замаскированы (сброшен, соответствующий каналу, бит в регистре CHNL_REQ_MASK_SET).

Таблица 8.4 – Перечень правил, при которых передача по каналам разрешена и запросы не маскируются (i – номер канала)

| Номер правила | Описание |
|---------------|---|
| 1 | Если канал не активен (передача не идет в данный момент), то установка бита C_i в регистре CHNL_SW_REQUEST или запрос от соответствующей периферии, инициирует передачу по каналу i |
| 2 | Одновременно активен может быть только один канал |
| 3 | Если запрос от периферии происходит в момент, когда канал активен, то контроллер обслужит этот запрос после завершения текущей передачи |
| 4 | Если приходит сразу несколько запросов от периферии для одного канала в момент, когда канал активен, то контроллер обслужит только первый запрос после завершения текущей передачи |
| 5 | Для циклов DMA, отличных по типу от периферийного режима «разборка-сборка», по окончании 2^R передач контроллер сбрасывает бит C_i в регистре CHNL_USEBURST_SET, если количество оставшихся передач меньше, чем 2^R . Позволяя периферии завершить передачи, используя как SREQ запросы, так и BREQ. В периферийном режиме «разборка/сборка» контроллер сбрасывает бит C_i в регистре CHNL_USEBURST_SET только, если количество оставшихся передач с использованием альтернативной структуры управляющих данных меньше, чем 2^R |
| 6 | Контроллер игнорирует запрос SREQ, если бит C_i регистра WAITONREQ_STATUS сброшен или установлен бит C_i регистра CHNL_USEBURST_SET |
| 7 | Необходимо с осторожностью устанавливать разряды регистра CHNL_USEBURST_SET. Если значение, указанное в регистре N-1 меньше, чем значение 2^R , то контроллер не очистит разряды CHNL_USEBURST_SET и поэтому одиночные запросы SREQ будут запрещены. Если программные запросы через регистр CHNL_SW_REQUEST не генерируются, и периферия не осуществляет запросов на пакетную обработку BREQ, то контроллер никогда не выполнит необходимых передач |

Окончание таблицы 8.4

| Номер правила | Описание |
|---------------|--|
| 8 | Для типов циклов DMA, отличных от периферийного режима «разборка-сборка», если придет запрос SREQ, то контроллер выполнит одну передачу. В периферийном режиме «Исполнение с изменением конфигурации», если придет запрос SREQ, контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных. Затем без осуществления арбитража выполняет одну передачу, используя альтернативную структуру управляющих данных |
| 9 | Для типов циклов DMA, отличных от периферийного режима «разборка-сборка», если одновременно пришли запросы SREQ и BREQ, то приоритет предоставляется BREQ, и контроллер выполняет 2^R DMA передач (или число передач, указанное в поле N-1). В периферийном режиме «разборка-сборка», если одновременно пришли запросы SREQ и BREQ, то приоритет также предоставляется BREQ, и контроллер выполняет 2^R передач с использованием первичной структуры управляющих данных. Затем без осуществления арбитража выполняет 2^R передач (или число передач, указанное в поле N-1), используя альтернативную структуру управляющих данных |
| 10 | В периферийном режиме «разборка-сборка», если бит next_use_burst в CHANNEL_CFG установлен, то контроллер устанавливает соответствующий каналу бит в регистре CHNL_USEBURST_SET после окончания цикла DMA, использующего альтернативные управляющие данные |
| 11 | Когда установлен бит C_i регистра CHNL_REQ_MASK_SET, контроллер игнорирует запросы SREQ и BREQ |

При отключении канала (бит C_i регистра CHNL_ENABLE_SET сброшен) контроллер DMA осуществляет передачи согласно правилам, представленным в таблице 8.5.

Таблица 8.5 – Перечень правил осуществления передач для запрещенных каналов

| Номер правила | Описание |
|---------------|---|
| 1 | Если приходит запрос на пакетную обработку BREQ от периферии, то происходит вызов прерывания канала DMA (если было включено). Это позволяет сигнализировать о запросе, даже если канал выключен |
| 2 | Если приходит запрос на одиночную передачу SREQ от периферии, то происходит вызов прерывания канала DMA (если было включено) при условии, что бит C_i регистра WAITONREQ_STATUS установлен, а бит C_i регистра CHNL_USEBURST_SET сброшен. Это позволяет сигнализировать о запросе, даже если канал выключен |

8.3 Правила арбитража

Контроллер имеет возможность настройки момента арбитража при передачах DMA. Эта возможность позволяет уменьшить время отклика при обслуживании каналов с высоким приоритетом.

Контроллер имеет настройки, которые определяют количество передач по шине АНВ-Lite до повторения арбитража (перearбитрации). Это значение задается параметром R (поле R_power в регистре CHANNEL_CFG структуры управляющих данных канала).

Количество транзакций одного канала до переарбитрации при этом равно 2^R . Например, если $R = 4$, то арбитраж будет проводиться через каждые 16 передач DMA.

Необходимо с осторожностью устанавливать большие значения R для низкоприоритетных каналов, так как это может привести к невозможности обслуживать запросы по высокоприоритетным каналам.

При $N > 2^R$ (N – номер передачи) и если результат деления 2^R на N не целое число, контроллер всегда выполняет последовательность из 2^R передач до тех пор, пока не станет верным $N < 2^R$. Контроллер выполняет оставшиеся N передач в конце цикла DMA.

Приоритет

При проведении арбитража определяется канал для обслуживания в следующем цикле DMA. На выбор следующего канала влияют:

- номер канала;
- уровень приоритета, присвоенного каналу.

Каждому каналу может быть присвоен уровень приоритета по умолчанию или высокий. Изменение уровня приоритета осуществляется установкой соответствующего бита C_i (i – номер канала) в регистрах $CHNL_PRIORITY_SET$ и $CHNL_PRIORITY_CLR$. Канал 0 имеет наивысший уровень приоритета.

Порядок каналов по уменьшению уровня приоритета представлен в таблице 8.6.

После окончания цикла DMA контроллер выбирает следующий для обслуживания канал из всех включенных каналов DMA. Рисунок 8.2 иллюстрирует процесс выбора следующего канала для обслуживания.

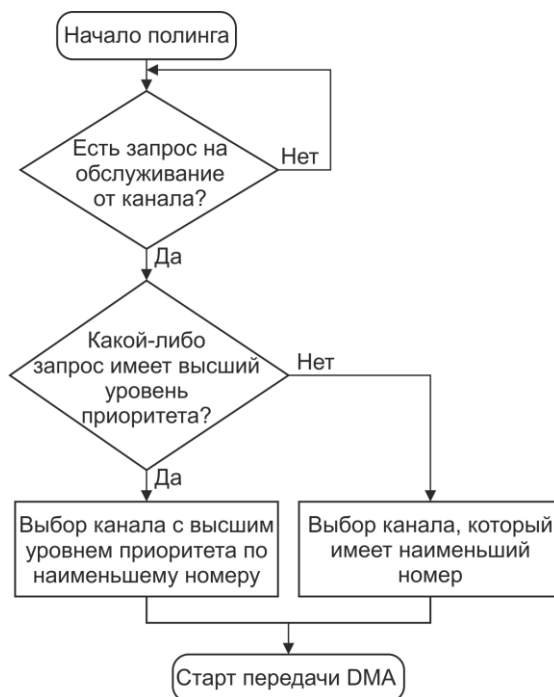



Рисунок 8.2 – Алгоритм выбора (полинга) следующего канала для обслуживания

Таблица 8.6 – Распределение приоритетов

| Номер канала Ci | Состояние бита Ci в регистре CHNL_PRIORITY_SET | Уровень приоритета | Порядок изменения уровня приоритета | |
|-----------------|--|--------------------|---|--------|
| 0 | 1 | Высокий | Снижение уровня приоритета  | |
| 1 | 1 | Высокий | | |
| 2 | 1 | Высокий | | |
| 3 | 1 | Высокий | | |
| ... | ... | ... | | |
| 22 | 1 | Высокий | | |
| 23 | 1 | Высокий | | |
| 0 | 0 | По умолчанию | | |
| 1 | 0 | По умолчанию | | |
| 2 | 0 | По умолчанию | | |
| 3 | 0 | По умолчанию | | |
| ... | ... | ... | | |
| 22 | 0 | По умолчанию | | |
| 23 | 0 | По умолчанию | | |
| | | | | Низший |

8.4 Типы циклов

Для всех типов циклов DMA повторный арбитраж происходит после 2^R передач DMA. Если установить длинный период арбитража на низкоприоритетном канале, то это заблокирует все запросы на обработку от других каналов до тех пор, пока не будут выполнены 2^R передач DMA по данному каналу. Поэтому, устанавливая значение R, необходимо учитывать, что это может привести к повышенному времени отклика на запрос на обработку от высокоприоритетных каналов.

Поддерживаются следующие типы циклов DMA:

- недействительный (структура управляющих данных канала в запрещенном состоянии);
- основной;
- авто-запрос;
- «пинг-понг»;
- работа с памятью в режиме «разборка-сборка» (scatter-gather);
- работа с периферией в режиме «разборка-сборка».

Задание типа цикла DMA осуществляется программированием поля cycle_ctl регистра CHANNEL_CFG структуры управляющих данных канала.

Недействительный цикл

После окончания цикла DMA контроллер устанавливает тип цикла в значение «недействительный» для предотвращения повтора выполненного цикла DMA.

Основной цикл

В данном режиме контроллер работает либо с первичными, либо с альтернативными управляющими данными канала, совершая по 2^R передач по каждому запросу.

Перед началом работы необходимо включить контроллер DMA и разрешить работу канала: установить соответствующий каналу бит в регистре CHNL_ENABLE_SET и бит MASTER_ENBL в регистре CFG, а также проверить, что запросы канала не замаскированы (сброшен соответствующий каналу бит в регистре CHNL_REQ_MASK_SET).

После того, как разрешена работа канала, цикл DMA выглядит следующим образом:

1 Контроллер ожидает получения запроса (программного либо от периферии) на обработку. Если запрос получен, то контроллер переходит к шагу 2.

2 Контроллер выполняет 2^R передач. Если число оставшихся передач 0, контроллер переходит к шагу 4, иначе выполняется шаг 3.

3 Происходит осуществление арбитража: если высокоприоритетный канал выдает запрос на обработку, то контроллер начинает обслуживание этого канала, иначе происходит ожидание очередного запроса на обработку по каналу, и если периферийный блок или программа его выдает, то контроллер переходит к шагу 2.

4 Контроллер указывает центральному процессору на завершение цикла DMA. Вызывается соответствующее каналу прерывание (если было включено).

Авто-запрос

Контроллеру необходим лишь одиночный запрос для разрешения работы и выполнения цикла DMA. Такая работа позволяет выполнять передачу больших пакетов данных без существенного увеличения времени отклика на обслуживание высокоприоритетных запросов и не требует множественных запросов на обработку от программы или периферийных блоков. Контроллер позволяет выбрать для использования либо первичную, либо альтернативную структуру управляющих данных канала.

Перед началом работы необходимо включить контроллер DMA и разрешить работу канала: установить соответствующий каналу бит в регистре CHNL_ENABLE_SET и бит MASTER_ENBL в регистре CFG, а также проверить, что запросы канала не замаскированы (сброшен, соответствующий каналу, бит в регистре CHNL_REQ_MASK_SET).

После того, как разрешена работа канала, цикл DMA выглядит следующим образом:

1 Контроллер ожидает получения запроса (программного либо от периферии) на обработку. Если запрос получен, то контроллер переходит к шагу 2.

2 Контроллер выполняет 2^R передач. Если число оставшихся передач 0, контроллер переходит к шагу 4, иначе выполняется шаг 3.

3 Осуществление арбитража: если высокоприоритетный канал выдает запрос на обработку, то контроллер начинает обслуживание этого канала, иначе контроллер переходит к шагу 2.

4 Контроллер указывает центральному процессору на завершение цикла DMA. Вызывается соответствующее каналу прерывание (если было включено).

Отличие от режима «основной» состоит в том, что в режиме «авто-запрос» контроллер позволит осуществить все N транзакций по одному запросу, в то время как в основном режиме по каждому запросу будет выполняться лишь 2^R передач.

«Пинг-понг»

Контроллер выполняет цикл DMA, используя одну из первичных структур управляющих данных, а затем выполняет еще один цикл DMA, используя альтернативную структуру управляющих данных. Контроллер выполняет циклы DMA с переключением структур до тех пор, пока не считает режим «недействительный» или «основной», или пока процессор не запретит работу канала.

На рисунке 8.3 показан пример функционирования контроллера DMA в режиме «пинг-понг». Пояснения к рисунку 8.3 представлены в виде таблицы 8.7.



Рисунок 8.3 – Пример функционирования контроллера в режиме «пинг-понг»

Таблица 8.7 – Пояснения к схеме на рисунке 8.3

| Шаг | Действия процессора и контроллера |
|-----|--|
| А | <p>Процессор включает контроллер DMA и разрешает работу канала.</p> <p>В программе устанавливаются первичная структура управляющих данных для шага А и альтернативная структура управляющих данных для шага В. Это позволит контроллеру переключиться к шагу В незамедлительно после выполнения шага А, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала.</p> <p>Контроллер получает запрос и выполняет 4 передачи DMA.</p> <p>Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала, контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов.</p> <p>Контроллер выполняет оставшиеся 2 передачи DMA.</p> <p>Контроллер входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов.</p> <p>Примечание – После выполнения шага А процессор может установить первичные управляющие данные канала для шага С. Это позволит контроллеру переключиться к шагу С незамедлительно после выполнения шага В, при условии, что контроллер не получит запрос на обработку от высокоприоритетного канала. После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг В.</p> |

Окончание таблицы 8.7

| Шаг | Действия процессора и контроллера |
|-----|---|
| В | <p>Контроллер выполняет 4 передачи DMA. Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов. Контроллер выполняет 4 передачи DMA. Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов. Контроллер выполняет оставшиеся 4 передачи DMA. Контроллер входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов.</p> <p>Примечание – После выполнения шага В процессор может установить альтернативные управляющие данные канала для шага D. После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг С.</p> |
| С | <p>Контроллер выполняет 2 передачи DMA. Контроллер входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов.</p> <p>Примечание – После выполнения шага С процессор может установить первичные управляющие данные канала для шага E. После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг D.</p> |
| D | <p>Контроллер выполняет 4 передачи DMA. Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов. Контроллер выполняет оставшуюся передачу DMA. Контроллер входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов.</p> <p>Примечание – После получения нового запроса на обработку от канала при условии его наивысшего приоритета исполняется шаг E.</p> |
| E | <p>Контроллер выполняет 4 передачи DMA. Контроллер выполняет арбитраж. После получения запроса на обработку от этого же канала контроллер продолжает цикл в ситуации отсутствия высокоприоритетных запросов. Контроллер выполняет оставшиеся 3 передачи DMA. Контроллер входит в процедуру арбитража. Если прерывание включено, то произойдет его вызов</p> |

Если контроллер получит новый запрос на обработку от данного канала и этот запрос будет самым приоритетным, контроллер предпримет попытку выполнения следующего шага. Однако из-за того, что процессор не установил альтернативные управляющие данные, и по окончании шага D контроллер установил поле `cycle_ctrl` альтернативной управляющей структуры в состояние 000b, передачи DMA прекращаются.

Работа с памятью в режиме «разборка-сборка»

Алгоритм работы данного режима является оптимальным именно для работы с памятью, но, несмотря на это, его использование возможно для любого типа передачи данных: память – память, периферия – память, память – периферия, с помощью, как программных запросов, так и запросов от периферии.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных.

Контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, заполняя альтернативную структуру канала данными, доступными для первичной управляющей структуры. По окончании этих передач контроллер входит в процедуру арбитража, и если более высокоприоритетных запросов не обнаружено, начинает цикл DMA, используя обновленные альтернативные управляющие данные. После – арбитраж, затем контроллер выполняет еще 4 передачи DMA, вновь заполняя альтернативную структуру данными с помощью первичной структуры.

Контроллер продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- передача с использованием альтернативной управляющей структуры будет выполнена в режиме цикла «основной»;
- контроллер считает «неправильную» структуру управляющих данных. После исполнения контроллером N передач с использованием первичных управляющих данных он делает эти управляющие данные «неправильными» путем записи в поле `cycle_ctrl` значения 000b.

Контроллер устанавливает прерывание канала DMA в этом режиме работы только тогда, когда последний цикл передач DMA выполняется с использованием режима «основной». Также необходимо помнить, что для режима «основной» авто-запросы не действуют.

В таблице 8.8 указаны константы, которые должны быть записаны пользователем в регистр CHANNEL_CFG первичной структуры управляющих данных канала для работы с памятью в режиме «разборка-сборка».

Таблица 8.8 – Конфигурация первичной структуры управляющих данных канала для работы с памятью в режиме «разборка-сборка»

| CHANNEL_CFG | | | |
|---|-------|-----------|---|
| <div style="display: flex; justify-content: space-between; font-size: 8px; margin-bottom: 5px;"> 313029282726252423222120191817161514131211109876543210 </div> <div style="display: flex; justify-content: space-between; font-size: 12px; border: 1px solid black; padding: 2px;"> 1010101010--0010---------------0100 </div> | | | |
| Поле | Биты | Константа | Пояснение |
| dst_inc | 31-30 | 10b | Контроллер производит инкремент адреса с шагом в двойное слово. |
| dst_size | 29-28 | 10b | Контроллер осуществляет передачу двойным словом. |
| src_inc | 27-26 | 10b | Контроллер производит инкремент адреса с шагом в двойное слово. |
| src_size | 25-24 | 10b | Контроллер осуществляет передачу двойными словами. |
| dst_prot_ctrl | 23-21 | – | Управление защитой шины при записи данных в приемник. Задается пользователем. |
| src_prot_ctrl | 20-18 | – | Управление защитой шины при чтении данных из источника. Задается пользователем. |

Окончание таблицы 8.8

| Поле | Биты | Константа | Пояснение |
|----------------|-------|-----------|---|
| R_power | 17-14 | 0010b | Контроллер выполняет 4 передачи DMA ($2^R = 2^2 = 4$) |
| N-1 | 13-4 | – | Настраивает контроллер на выполнение N передач DMA. Так как поле R_power задает значение 2, то необходимо задавать значение N, кратное 4. Число, равное N/4, это количество раз, которое нужно настраивать альтернативные управляющие данные. Задается пользователем. |
| next_use_burst | 3 | 0 | Для данного режима бит должен быть сброшен. |
| cycle_ctrl | 2-0 | 100b | Контроллер работает с памятью в режиме «разборка/сборка» с использованием первичных управляющих данных канала. |

В указатель конца данных источника SRC_DATA_END_PTR первичной структуры необходимо записать адрес конца области памяти, в которой последовательно расположены нужное количество наборов управляющих данных для программирования альтернативной структуры канала.

В указатель конца данных приемника DST_DATA_END_PTR первичной структуры необходимо записать адрес конца альтернативной управляющей структуры используемого канала.

На рисунке 8.4 показан пример функционирования контроллера в режиме «разборка-сборка». Пояснения к рисунку 8.4 приведены ниже.



Рисунок 8.4 – Пример функционирования контроллера в режиме «разборка-сборка»

Инициализация

1 Первичная структура управляющих данных настраивается для работы с памятью в режиме «разборка-сборка» путем записи в `cycle_ctrl` значения 100b. Так как управляющие данные канала состоят из четырех слов, $R_power = 0010b$. Поскольку количество задач равно четырем, то $N = 16$, т. е. в значение поля $N-1 = 00Fh$.

2 Управляющие данные для шагов A, B, C, D располагаются в области ОЗУ. Адрес конца этой области заносится в регистр `SRC_DATA_END_PTR` первичных управляющих данных. Пример размещения и заполнения управляющих данных для альтернативной структуры показан в таблице 8.9. Исходя из примера, в регистр `SRC_DATA_END_PTR` первичной управляющей структуры необходимо занести значение 2000015Ch.

Таблица 8.9 – Пример размещения управляющих данных для альтернативной структуры

| Тип данных | Адрес ОЗУ | Регистр | Значение |
|---------------------------------|-----------|------------------|--|
| Управляющие данные для задачи D | 2000015Ch | Не используется | XXXXXXXXXh |
| | 20000158h | CHANNEL_CFG | cycle_ctrl = 001b, $2^R = 4$, $N = 4$ |
| | 20000154h | DST_DATA_END_PTR | 2000DE00h |
| | 20000150h | SRC_DATA_END_PTR | 2000D000h |
| Управляющие данные для задачи C | 2000016Ch | Не используется | XXXXXXXXXh |
| | 20000168h | CHANNEL_CFG | cycle_ctrl = 101b, $2^R = 8$, $N = 5$ |
| | 20000164h | DST_DATA_END_PTR | 2000CE00h |
| | 20000160h | SRC_DATA_END_PTR | 2000C000h |
| Управляющие данные для задачи B | 2000017Ch | Не используется | XXXXXXXXXh |
| | 20000178h | CHANNEL_CFG | cycle_ctrl = 101b, $2^R = 2$, $N = 8$ |
| | 20000174h | DST_DATA_END_PTR | 2000BE00h |
| | 20000170h | SRC_DATA_END_PTR | 2000B000h |
| Управляющие данные для задачи A | 2000018Ch | Не используется | XXXXXXXXXh |
| | 20000188h | CHANNEL_CFG | cycle_ctrl = 101b, $2^R = 4$, $N = 3$ |
| | 20000184h | DST_DATA_END_PTR | 2000AE00h |
| | 20000180h | SRC_DATA_END_PTR | 2000A000h |

В регистр `DST_DATA_END_PTR` первичной структуры необходимо занести адрес конца альтернативной структуры управляющих данных используемого канала. Например, при использовании канала 9 в регистр `DST_DATA_END_PTR` необходимо занести значение XXXXX29Ch.

3 Включается контроллер DMA и разрешается работа канала путем установки соответствующего каналу бита в регистре `CHNL_ENABLE_SET` и бита `MASTER_ENBL` в регистре `CFG`. Также необходимо удостовериться, что запросы канала не замаскированы (сброшен соответствующий каналу бит в регистре `CHNL_REQ_MASK_SET`).

Функционирование

1 Первичная структура, копирование данных задачи А. По получении первого запроса на обслуживание контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи А. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Далее контроллер выполняет задачу А с использованием альтернативных данных, по окончании генерирует автозапрос для канала и проводит процедуру арбитража.

2 Первичная структура, копирование данных задачи В. По получении автозапроса контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи В. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Далее контроллер выполняет задачу В с использованием альтернативных данных, по окончании генерирует автозапрос для канала и проводит процедуру арбитража.

3 Первичная структура, копирование данных задачи С. По получении автозапроса на обслуживание контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи С. Контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Далее контроллер выполняет задачу С с использованием альтернативных данных, по окончании генерирует автозапрос для канала и проводит процедуру арбитража.

4 Первичная структура, копирование данных задачи D. По получении автозапроса на обслуживание контроллер выполняет четыре передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи D. Контроллер записывает в `cycle_ctrl` первичных данных значение `000b` для индикации о том, что эта структура управляющих данных является «неправильной». Далее контроллер генерирует автозапрос для канала, после чего проводит процедуру арбитража.

Далее контроллер выполняет задачу D, используя тип цикла «основной». По завершении задачи генерирует прерывание канала DMA (если было включено) и входит в процедуру арбитража. Цикл работы с памятью в режиме «разборка-сборка» завершен.

Работа с периферией в режиме «разборка-сборка»

Алгоритм работы данного режима является оптимальным именно для работы с периферией, но, несмотря на это, его использование возможно для любого типа передачи данных: память – память, периферия – память, память – периферия, с помощью, как программных запросов, так и запросов от периферии.

В данном режиме контроллер использует первичные управляющие данные для программирования альтернативных управляющих данных.

Контроллер, получая начальный запрос на обработку, выполняет 4 передачи DMA, заполняя альтернативную структуру канала данными, доступными для первичной управляющей структуры. По окончании этих передач контроллер без осуществления арбитража начинает цикл DMA, используя обновленные альтернативные управляющие данные. После – арбитраж, затем контроллер выполняет еще 4 передачи DMA, вновь заполняя альтернативную структуру данными с помощью первичной структуры. Это единственный случай, при котором контроллер не осуществляет процедуру арбитража после выполнения передачи DMA, используя первичные управляющие данные.

Контроллер продолжает выполнять циклы DMA, меняя структуры управляющих данных, пока не произойдет одно из следующих условий:

- передача с использованием альтернативной управляющей структуры будет выполнена в режиме цикла «основной»;
- контроллер считает «неправильную» структуру управляющих данных. После исполнения контроллером N передач с использованием первичных управляющих данных он делает эти управляющие данные «неправильными» путем записи в поле `cycle_ctrl` значения `000b`.

Контроллер устанавливает прерывание канала DMA в этом режиме работы только тогда, когда последний цикл передач DMA выполняется с использованием режима «основной». Также необходимо помнить, что для режима «основной» авто-запросы не действуют.

В таблице 8.10 указаны константы, которые должны быть записаны пользователем в регистр CHANNEL_CFG первичной структуры управляющих данных канала для работы с периферией в режиме «разборка-сборка».

Таблица 8.10 – Конфигурация первичной структуры управляющих данных канала для работы с периферией в режиме «разборка-сборка»

| CHANNEL_CFG | | | |
|----------------|-------|----------------|--|
| | | | |
| Поле | Биты | Конс- танта | Пояснение |
| dst_inc | 31-30 | 10b | Контроллер производит инкремент адреса с шагом в двойное слово |
| dst_size | 29-28 | 10b | Контроллер осуществляет передачу двойным словом |
| src_inc | 27-26 | 10b | Контроллер производит инкремент адреса с шагом в двойное слово |
| src_size | 25-24 | 10b | Контроллер осуществляет передачу двойными словами |
| dst_prot_ctrl | 23-21 | – | Управление защитой шины при записи данных в приемник. Задается пользователем |
| src_prot_ctrl | 20-18 | – | Управление защитой шины при чтении данных из источника. Задается пользователем |
| R_power | 17-14 | 0010b | Контроллер выполняет 4 передачи DMA ($2^R = 2^2 = 4$) |
| N-1 | 13-4 | – | Настраивает контроллер на выполнение N передач DMA. Так как поле R_power задает значение 2, то необходимо задавать значение N, кратное 4. Число, равное N/4, это количество раз, которое нужно настраивать альтернативные управляющие данные. Задается пользователем |
| next_use_burst | 3 | 0 | Для данного режима бит должен быть сброшен |
| cycle_ctrl | 2-0 | 110b | Контроллер работает с периферией в режиме «разборка-сборка» с использованием первичных управляющих данных канала |

В указатель конца данных источника SRC_DATA_END_PTR первичной структуры необходимо записать адрес конца области памяти, в которой последовательно расположены нужное количество наборов управляющих данных для программирования альтернативной структуры канала.

В указатель конца данных приемника DST_DATA_END_PTR первичной структуры необходимо записать адрес конца альтернативной управляющей структуры используемого канала.

На рисунке 8.5 показан пример функционирования контроллера в режиме «разборка-сборка».

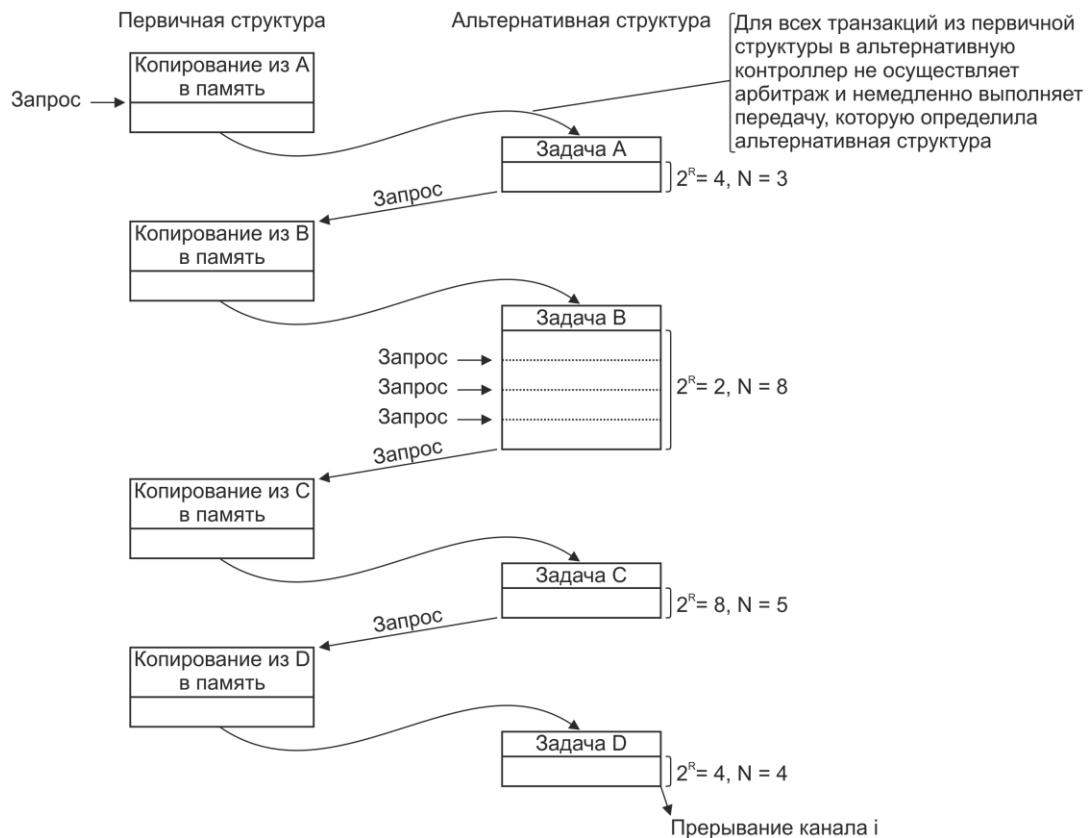


Рисунок 8.5 – Пример функционирования контроллера в режиме работы с периферией «разборка-сборка»

Инициализация

1 Первичная структура управляющих данных настраивается для работы с периферией в режиме «разборка-сборка» путем записи в `cycle_ctrl` значения 110b. Так как управляющие данные канала состоят из 4 слов, $R_power = 0010b$. Поскольку количество задач равно 4, то $N = 16$, т. е. в значение поля $N-1 = 00Fh$.

2 Управляющие данные для шагов А, В, С, D располагаются в области ОЗУ. Адрес конца этой области заносится в регистр `SRC_DATA_END_PTR` первичных управляющих данных. Пример размещения и заполнения управляющих данных для альтернативной структуры, показан в таблице 8.11. Исходя из примера, в регистр `SRC_DATA_END_PTR` необходимо занести значение 2000015Ch.

Таблица 8.11 – Пример размещения управляющих данных для альтернативной структуры

| Тип данных | Адрес ОЗУ | Регистр | Значение |
|---------------------------------|-----------|------------------|--|
| Управляющие данные для задачи D | 2000015Ch | Не используется | XXXXXXXXXh |
| | 20000158h | CHANNEL_CFG | <code>cycle_ctrl = 001b</code> , $2^R = 4$, $N = 4$ |
| | 20000154h | DST_DATA_END_PTR | 2000DE00h |
| | 20000150h | SRC_DATA_END_PTR | 2000D000h |
| Управляющие данные для задачи С | 2000016Ch | Не используется | XXXXXXXXXh |
| | 20000168h | CHANNEL_CFG | <code>cycle_ctrl = 111b</code> , $2^R = 8$, $N = 5$ |
| | 20000164h | DST_DATA_END_PTR | 2000CE00h |
| | 20000160h | SRC_DATA_END_PTR | 2000C000h |

Окончание таблицы 8.11

| Тип данных | Адрес ОЗУ | Регистр | Значение |
|---------------------------------|-----------|------------------|--------------------------------------|
| Управляющие данные для задачи В | 2000017Ch | Не используется | XXXXXXXXXh |
| | 20000178h | CHANNEL_CFG | cycle_ctrl = 111b, $2^R = 2$, N = 8 |
| | 20000174h | DST_DATA_END_PTR | 2000BE00h |
| | 20000170h | SRC_DATA_END_PTR | 2000B000h |
| Управляющие данные для задачи А | 2000018Ch | Не используется | XXXXXXXXXh |
| | 20000188h | CHANNEL_CFG | cycle_ctrl = 111b, $2^R = 4$, N = 3 |
| | 20000184h | DST_DATA_END_PTR | 2000AE00h |
| | 20000180h | SRC_DATA_END_PTR | 2000A000h |

В регистр DST_DATA_END_PTR первичной структуры необходимо занести адрес конца альтернативной структуры управляющих данных используемого канала. Например, при использовании канала 9 в регистр DST_DATA_END_PTR необходимо занести значение XXXXX29Ch.

3 Включается контроллер DMA и разрешается работа канала путем установки соответствующего каналу бита в регистре CHNL_ENABLE_SET и бита MASTER_ENBL в регистре CFG. Также необходимо удостовериться, что запросы канала не замаскированы (сброшен соответствующий каналу бит в регистре CHNL_REQ_MASK_SET).

Функционирование

1 Первичная структура, копирование данных для задачи А. По получении запроса на обслуживание контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи А.

Далее контроллер сразу же выполняет задачу А и по окончании проводит процедуру арбитража. После выставления нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается.

2 Первичная структура, копирование данных для задачи В. Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи В.

Далее контроллер выполняет задачу В. Для завершения задачи необходимо три запроса (программных или от периферии). По окончании контроллер проводит процедуру арбитража. После выставления нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается.

3 Первичная структура, копирование данных для задачи С. Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для задачи С.

Далее контроллер выполняет задачу С и по окончании проводит процедуру арбитража.

После выставления периферией нового запроса на обслуживание, при условии, что этот запрос является наиболее приоритетным, процесс продолжается.

4 Первичная структура, копирование данных для задачи D. Контроллер выполняет 4 передачи DMA. Эти передачи записывают альтернативную структуру управляющих данных для шага D. Контроллер записывает в cycle_ctrl первичных данных значение 000b для индикации о том, что эта структура управляющих данных является «неправильной».

Далее контроллер выполняет задачу D, используя основной цикл DMA, входит в прерывание канала DMA (если включено) и запускает процедуру арбитража. Цикл работы с периферией в режиме «разборка-сборка» завершен.

8.5 Индикация ошибок

Контроллер может отключить канал DMA Ci в следующих случаях:

- при завершении цикла DMA;
- при чтении режима канала «недействительный»;
- при появлении ошибки на шине АНВ-Lite.

Как только контроллер получает сообщение об ошибке по шине АНВ-Lite, он отключает канал, в котором обнаружена ошибка и устанавливает флаг ERR_CLR в одноименном регистре.

Для того чтобы определить канал, в котором произошла ошибка, программа, выполняемая процессором, должна всегда хранить данные о каналах, которые недавно вызывали прерывания, т. е. завершали работу и отключались.

Алгоритм определения канала с ошибкой:

- необходимо прочитать регистр CHNL_ENABLE_SET с целью создания текущего списка отключенных каналов;
- процессор должен сравнить список выключенных каналов, полученный в результате чтения регистра CHNL_ENABLE_SET, с данными о каналах, которые недавно вызывали прерывания. Канал, который отключился и по которому отсутствуют данные о вызове прерывания, это и есть канал, с которым связана ошибка.

В контроллере DMA присутствует возможность использования режимов защиты шины АНВ-Lite: при записи в приемник, при чтении из источника и при обращении к структурам управляющих данных каналов. Защита шины в каждой из ситуаций настраивается индивидуально. Доступными режимами защиты являются: кэширование, буферизация, привилегированный доступ.

Защита шины при записи в приемник настраивается полем dst_prot_ctrl в ячейке CHANNEL_CFG структуры управляющих данных канала.

Защита шины при чтении из источника настраивается полем src_prot_ctrl в ячейке CHANNEL_CFG структуры управляющих данных канала.

Защита шины при обращении контроллера к структурам управляющих данных каналов настраивается полем CHNL_PROT_CTRL в регистре CFG.

9 Прерывания

Таблица прерываний представляет собой перечень адресов, соответствующих определенным обработчикам прерываний (см. таблицу 9.1).

Таблица 9.1 – Таблица прерываний

| Номер вектора | Смещение | Обозначение | Описание |
|---------------|-------------|--------------|--|
| - | 0000h | SP | Вершина стека |
| - | 0004h | Reset | Сброс |
| - | 0008h | NMI | Немаскируемое прерывание NMI |
| - | 000Ch | HardFault | Любой отказ, если соответствующий обработчик не может быть запущен |
| - | 0010h | MemManage | Прерывание по отказу системы управления памятью |
| - | 0014h | BusFault | Прерывание по отказу шины АНВ |
| - | 0018h | UsageFault | Прерывание по ошибке программы |
| - | 001Ch-0028h | - | Зарезервировано |
| - | 002Ch | SVCall | Обработка прерываний, вызванных инструкцией SVC |
| - | 0030h | DebugMonitor | Прерывание монитора отладки |
| - | 0034h | - | Зарезервировано |
| - | 0038h | PendSV | Прерывание системного уровня. В приложении используется вызов «Супервизор», если этот запрос обслуживается базовой операционной системой |
| - | 003Ch | SysTick | Прерывание системного уровня. Прерывание вызывается таймером SysTick |
| 0 | 0040h | WWDG | Прерывание блока сторожевого таймера |
| 1 | 0044h | I2C0 | Прерывание блока I2C_0 |
| 2 | 0048h | I2C1 | Прерывание блока I2C_1 |
| 3 | 004Ch | TIM0 | Прерывание таймера 0 |
| 4 | 0050h | TIM1 | Прерывание таймера 1 |
| 5 | 0054h | TIM2 | Прерывание таймера 2 |
| 6 | 0058h | DMA_Stream0 | Прерывания контроллера DMA |
| 7 | 005Ch | DMA_Stream1 | |
| 8 | 0060h | DMA_Stream2 | |
| 9 | 0064h | DMA_Stream3 | |
| 10 | 0068h | DMA_Stream4 | |
| 11 | 006Ch | DMA_Stream5 | |
| 12 | 0070h | DMA_Stream6 | |
| 13 | 0074h | DMA_Stream7 | |
| 14 | 0078h | DMA_Stream8 | |
| 15 | 007Ch | DMA_Stream9 | |
| 16 | 0080h | DMA_Stream10 | |
| 17 | 0084h | DMA_Stream11 | |

Продолжение таблицы 9.1

| Номер вектора | Смещение | Обозначение | Описание |
|---------------|----------|--------------|--|
| 18 | 0088h | DMA_Stream12 | Прерывания контроллера DMA |
| 19 | 008Ch | DMA_Stream13 | |
| 20 | 0090h | DMA_Stream14 | |
| 21 | 0094h | DMA_Stream15 | |
| 22 | 0098h | DMA_Stream16 | |
| 23 | 009Ch | DMA_Stream17 | |
| 24 | 00A0h | DMA_Stream18 | |
| 25 | 00A4h | DMA_Stream19 | |
| 26 | 00A8h | DMA_Stream20 | |
| 27 | 00ACh | DMA_Stream21 | |
| 28 | 00B0h | DMA_Stream22 | |
| 29 | 00B4h | DMA_Stream23 | |
| 30 | 00B8h | UART0_MS | Прерывания блока UART 0 |
| 31 | 00BCh | UART0_RX | |
| 32 | 00C0h | UART0_TX | |
| 33 | 00C4h | UART0_RT | |
| 34 | 00C8h | UART0_E | |
| 35 | 00CCh | UART0 | |
| 36 | 00D0h | UART1_MS | Прерывания блока UART 1 |
| 37 | 00D4h | UART1_RX | |
| 38 | 00D8h | UART1_TX | |
| 39 | 00DCh | UART1_RT | |
| 40 | 00E0h | UART1_E | |
| 41 | 00E4h | UART1 | |
| 42 | 00E8h | UART2_MS | Прерывания блока UART 2 |
| 43 | 00ECh | UART2_RX | |
| 44 | 00F0h | UART2_TX | |
| 45 | 00F4h | UART2_RT | |
| 46 | 00F8h | UART2_E | |
| 47 | 00FCh | UART2 | |
| 48 | 0100h | UART3_MS | Прерывания блока UART 3 |
| 49 | 0104h | UART3_RX | |
| 50 | 0108h | UART3_TX | |
| 51 | 010Ch | UART3_RT | |
| 52 | 0110h | UART3_E | |
| 53 | 0114h | UART3 | |
| 54 | 0118h | PWM0 | Общее прерывание блока 0 ШИМ |
| 55 | 011Ch | PWM0_HD | Общее прерывание блока 0 ШИМ высокого разрешения |
| 56 | 0120h | PWM0_TZ | Прерывание детектора аварий блока 0 ШИМ |
| 57 | 0124h | PWM1 | Общее прерывание блока 1 ШИМ |
| 58 | 0128h | PWM1_HD | Общее прерывание блока 1 ШИМ высокого разрешения |
| 59 | 012Ch | PWM1_TZ | Прерывание детектора аварий блока 1 ШИМ |
| 60 | 0130h | PWM2 | Общее прерывание блока 2 ШИМ |
| 61 | 0134h | PWM2_HD | Общее прерывание блока 2 ШИМ высокого разрешения |
| 62 | 0138h | PWM2_TZ | Прерывание детектора аварий блока 2 ШИМ |

Продолжение таблицы 9.1

| Номер вектора | Смещение | Обозначение | Описание |
|---------------|----------|-------------|--|
| 63 | 013Ch | PWM3 | Общее прерывание блока 3 ШИМ |
| 64 | 0140h | PWM3_HD | Общее прерывание блока 3 ШИМ высокого разрешения |
| 65 | 0144h | PWM3_TZ | Прерывание детектора аварий блока 3 ШИМ |
| 66 | 0148h | PWM4 | Общее прерывание блока 4 ШИМ |
| 67 | 014Ch | PWM4_HD | Общее прерывание блока 4 ШИМ высокого разрешения |
| 68 | 0150h | PWM4_TZ | Прерывание детектора аварий блока 4 ШИМ |
| 69 | 0154h | PWM5 | Общее прерывание блока 5 ШИМ |
| 70 | 0158h | PWM5_HD | Общее прерывание блока 5 ШИМ высокого разрешения |
| 71 | 015Ch | PWM5_TZ | Прерывание детектора аварий блока 5 ШИМ |
| 72 | 0160h | PWM6 | Общее прерывание блока 6 ШИМ |
| 73 | 0164h | PWM6_HD | Общее прерывание блока 6 ШИМ высокого разрешения |
| 74 | 0168h | PWM6_TZ | Прерывание детектора аварий блока 6 ШИМ |
| 75 | 016Ch | PWM7 | Общее прерывание блока 7 ШИМ |
| 76 | 0170h | PWM7_HD | Общее прерывание блока 7 ШИМ высокого разрешения |
| 77 | 0174h | PWM7_TZ | Прерывание детектора аварий блока 7 ШИМ |
| 78 | 0178h | PWM8 | Общее прерывание блока 8 ШИМ |
| 79 | 017Ch | PWM8_HD | Общее прерывание блока 8 ШИМ высокого разрешения |
| 80 | 0180h | PWM8_TZ | Прерывание детектора аварий блока 8 ШИМ |
| 81 | 0184h | ADC_SEQ0 | Прерывание блока АЦП |
| 82 | 0188h | ADC_SEQ1 | |
| 83 | 018Ch | ADC_SEQ2 | |
| 84 | 0190h | ADC_SEQ3 | |
| 85 | 0194h | ADC_SEQ4 | |
| 86 | 0198h | ADC_SEQ5 | |
| 87 | 019Ch | ADC_SEQ6 | |
| 88 | 01A0h | ADC_SEQ7 | |
| 89 | 01A4h | ADC_CompInt | Прерывание компараторов блока АЦП |
| 90 | 01A8h | CAP0 | Прерывание блока захвата CAP_0 |
| 91 | 01ACh | CAP1 | Прерывание блока захвата CAP_1 |
| 92 | 01B0h | CAP2 | Прерывание блока захвата CAP_2 |
| 93 | 01B4h | CAP3 | Прерывание блока захвата CAP_3 |
| 94 | 01B8h | CAP4 | Прерывание блока захвата CAP_4 |
| 95 | 01BCh | CAP5 | Прерывание блока захвата CAP_5 |
| 96 | 01C0h | QEP0 | Прерывание квадратурного декодера QEP_0 |
| 97 | 01C4h | QEP1 | Прерывание квадратурного декодера QEP_1 |
| 98 | 01C8h | BootFlash | Прерывание контроллера загрузочной флеш-памяти |
| 99 | 01CCh | CMP0 | Прерывание блока аналогового компаратора 0 |
| 100 | 01D0h | CMP1 | Прерывание блока аналогового компаратора 1 |
| 101 | 01D4h | CMP2 | Прерывание блока аналогового компаратора 2 |
| 102 | 01D8h | SPI0 | Прерывание контроллера SPI_0 |
| 103 | 01DCh | SPI1 | Прерывание контроллера SPI_1 |
| 104 | 01E0h | SPI2 | Прерывание контроллера SPI_2 |

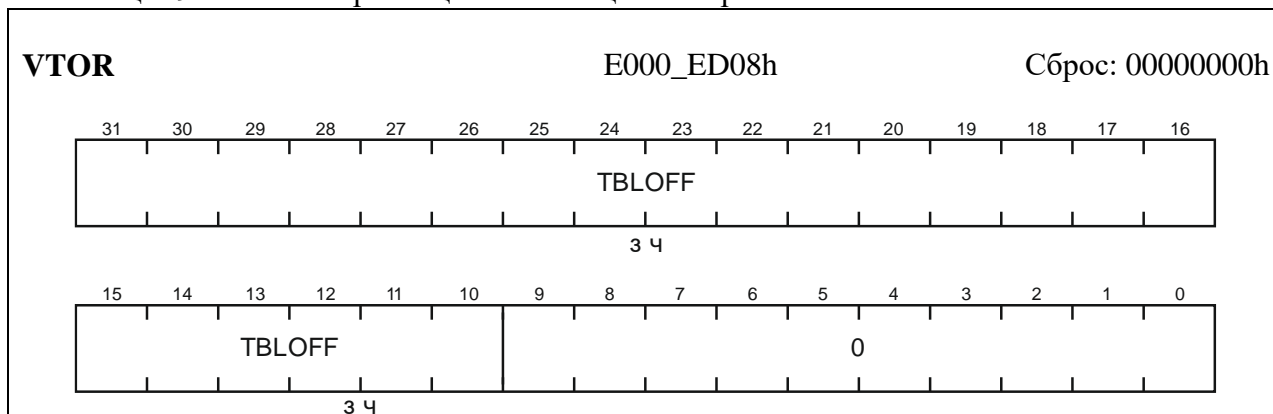
Окончание таблицы 9.1

| Номер вектора | Смещение | Обозначение | Описание |
|---------------|----------|-------------|---|
| 105 | 01E4h | SPI3 | Прерывание контроллера SPI_3 |
| 106 | 01E8h | UserFlash | Прерывание контроллера пользовательской флеш-памяти |
| 107 | 01ECh | GPIOA | Прерывание контроллера GPIO порта A |
| 108 | 01F0h | GPIOB | Прерывание контроллера GPIO порта B |
| 109 | 01F4h | GPIOC | Прерывание контроллера GPIO порта C |
| 110 | 01F8h | GIOD | Прерывание контроллера GPIO порта D |
| 111 | 01FCh | GPIOE | Прерывание контроллера GPIO порта E |
| 112 | 0200h | GPIOF | Прерывание контроллера GPIO порта F |
| 113 | 0204h | GPIOG | Прерывание контроллера GPIO порта G |
| 114 | 0208h | GPIOH | Прерывание контроллера GPIO порта H |
| 115 | 020Ch | Ethernet | Прерывание контроллера Ethernet |
| 116 | 0210h | CAN0 | Прерывания контроллера CAN |
| 117 | 0214h | CAN1 | |
| 118 | 0218h | CAN2 | |
| 119 | 021Ch | CAN3 | |
| 120 | 0220h | CAN4 | |
| 121 | 0224h | CAN5 | |
| 122 | 0228h | CAN6 | |
| 123 | 022Ch | CAN7 | |
| 124 | 0230h | CAN8 | |
| 125 | 0234h | CAN9 | |
| 126 | 0238h | CAN10 | |
| 127 | 023Ch | CAN11 | |
| 128 | 0240h | CAN12 | |
| 129 | 0244h | CAN13 | |
| 130 | 0248h | CAN14 | |
| 131 | 024Ch | CAN15 | |
| 132 | 0250h | RTC | Прерывание блока RTC |
| 133 | 0254h | USB | Прерывание контроллера USB OTG |

Первоначально адрес начала таблицы прерываний 00000000h.

Таблица векторов может быть размещена по другому адресу в памяти программ или в ОЗУ. В случае размещения таблицы прерываний в области ОЗУ появляется возможность изменять обработчики прерываний в процессе выполнения программы. Положение таблицы векторов в памяти определяется регистром VTOR, см. таблицу 9.2.

Таблица 9.2 – Регистр смещения таблицы векторов



Окончание таблицы 9.2

| Поле | Биты | Описание |
|--------|-------|---|
| TBLOFF | 31-10 | Биты 31-10 адреса таблицы векторов |
| 0 | 9-0 | Данные биты всегда должны быть равны 0, т.к. адрес таблицы векторов должен быть выровнен по границе 1024 байт |

Обработчики прерываний можно динамически менять, но при этом обязательно следует располагать следующие элементы:

- начальное значение основного указателя стека;
- вектор сброса Reset;
- вектор NMI;
- вектор исключения HardFault.

Остальные прерывания не могут генерироваться, пока не будут разрешены.

Контроллер прерываний NVIC

Контроллер обеспечивает:

- программное задание уровня приоритета независимо для каждого прерывания в диапазоне от 0 до 7 (прерывание с уровнем 0 имеет наивысший приоритет);
- генерирование сигнала прерывания по фронту и по уровню сигнала;
- динамическое изменение приоритета прерываний;
- разделение по группам с одинаковым приоритетом и по подгруппам внутри одной группы;
- передача управления из одного обработчика в другой без восстановления контекста.

Процессор автоматически сохраняет в стеке свое состояние при входе в обработчик прерывания и восстанавливает свое состояние после завершения обработки прерывания, т. е. без необходимости программирования этих операций.

Обработка прерываний по уровню и по фронту

Контроллер прерываний поддерживает прерывания, как по фронту, так и по уровню. Прерывание по фронту – импульсное прерывание, которое может иметь длительность большую или равную длительности такта системной частоты.

Прерывание по уровню возникает до тех пор, пока устройством удерживается заданный уровень сигнала. Если прерывание по уровню не было снято до завершения работы обработчика прерываний, то контроллер NVIC вновь начинает его обработку.

В случае прихода импульсного прерывания от любого источника в момент обработки предыдущего, в контроллере устанавливается флаг, сигнализирующий о приходе нового прерывания, которое будет обработано после завершения обработки текущего прерывания. В случае, если контроллер NVIC находится в состоянии ожидания и приходит импульсное прерывание от того же источника, обработка выполнится только один раз.

Для управления прерываниями используются пять групп регистров ISER, ICER, ISPR, ICPR и IABR. Подробное описание в приложении В.

10 Порты

В состав микроконтроллера входят порты ввода-вывода: 8-разрядный порт H и семь 16-разрядных портов – порт A, порт B, порт C, порт D, порт E, порт F, порт G. Структуры всех портов и функционирование идентичны.

Полученные данные сохраняются в регистре DATA порта. Данные для передачи записываются в регистр DATAOUT порта.

Каждый цифровой вывод порта микроконтроллера может использоваться как двунаправленный вывод общего назначения (режим GPIO). Помимо этого все выходы имеют альтернативные функции (от одной до трех, а выходы A0 – A7 по четыре).

Управление выводами осуществляется посредством регистров общего назначения и регистров GPIO.

На рисунке 10.1 приведена структурная схема нулевого вывода цифрового порта A микроконтроллера. Серым цветом отмечены нулевые биты регистров управления. Схемы всех выводов идентичны.

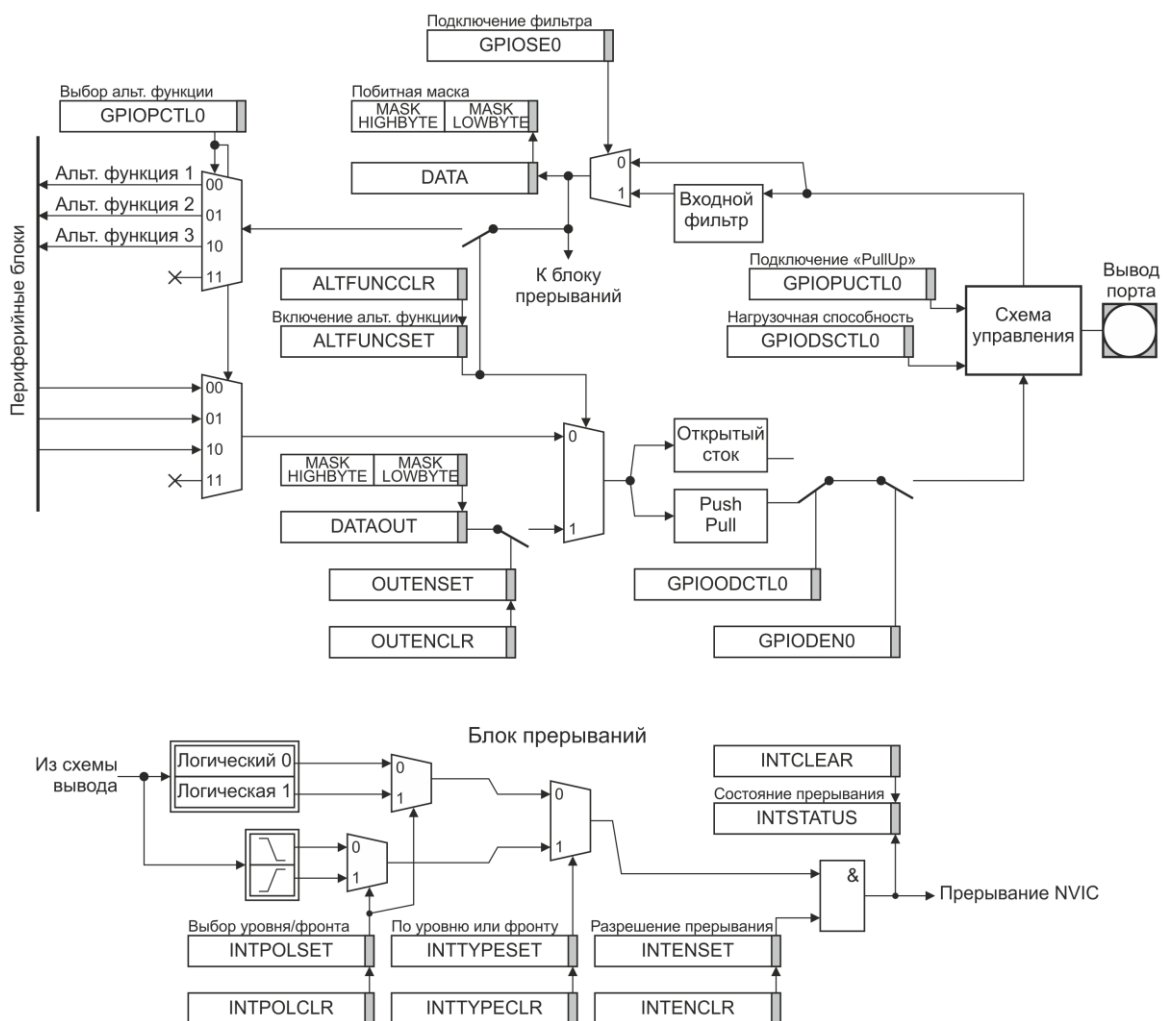


Рисунок 10.1 – Вывод цифрового порта A микроконтроллера и управляющие регистры

Схема состоит из двунаправленной площадки вывода, фильтра входных сигналов, мультиплексора выбора номера альтернативной функции, мультиплексора выбора режима работы (режим GPIO либо режим альтернативной функции).

Для каждого вывода задается режим работы (регистры GPIODEN, OUTENSET), номер альтернативной функции, режим подтяжки Pull-up (регистр GPIOUCTL), режим

нагрузочной способности (регистр GPIODSCTL), а также производится настройка на работу в режиме с открытым коллектором (регистр GPIOODCTL). Входной сигнал может подаваться для дальнейшей обработки как напрямую (асинхронный вход), так и проходить обработку через фильтр.

После сброса все выводы, кроме выводов JTAG, выводов D11 и E0, конфигурируются как выводы общего назначения (режим GPIO) и находятся в третьем состоянии. Направление работы выводов определяется состоянием бит регистра OUTENSET (для сброса установленных бит следует записать единицы в регистр OUTENCLR).

Для перевода желаемого вывода порта в режим альтернативной функции необходимо установить соответствующий бит в регистре ALTFUNCSET порта. Для отключения альтернативной функции нужно записать единицу в соответствующий бит регистра ALTFUNCCLR. Выбор номера альтернативной функции осуществляется посредством регистра GPIOCTLx. Каждому выводу соответствуют два бита регистра.

На рисунке 10.2 показана функциональная схема блока управления альтернативной функцией вывода.

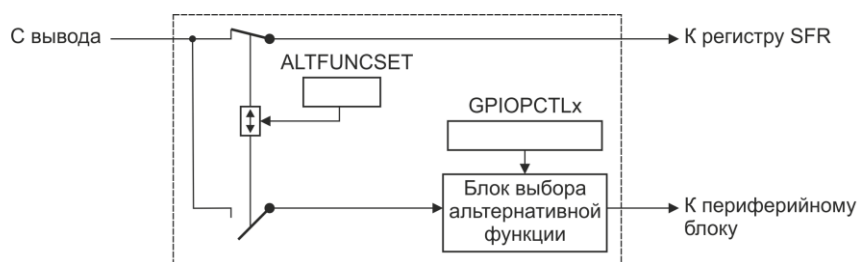


Рисунок 10.2 – Схема блока управления альтернативной функцией вывода

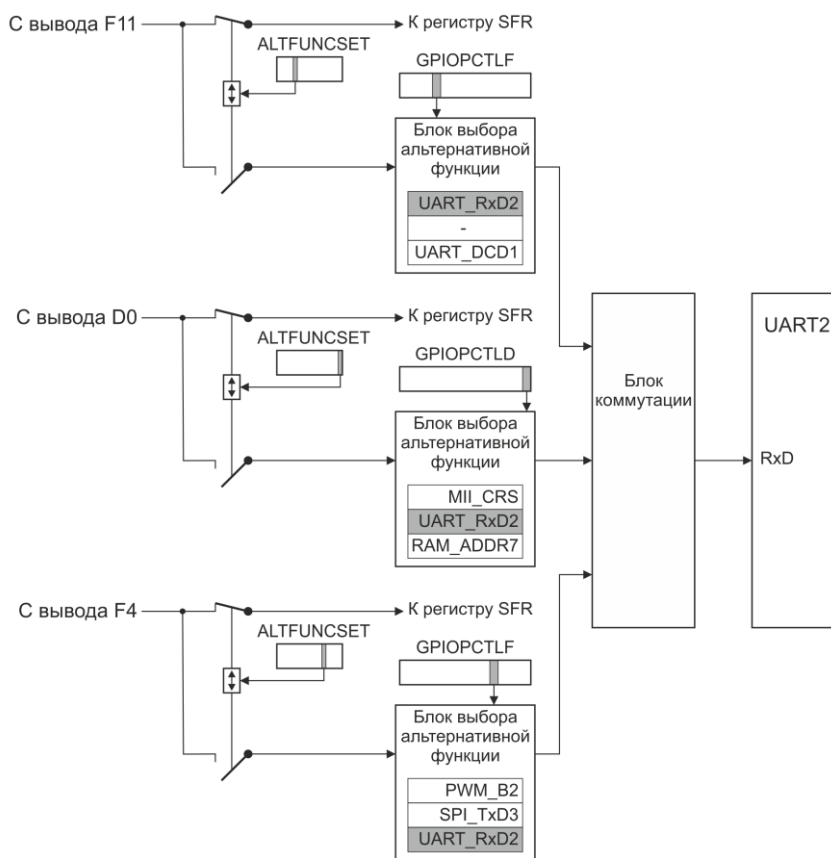


Рисунок 10.3 – Схема соединения блоков управления альтернативными функциями выводов микроконтроллера и блока UART2

Входы и выходы периферийных блоков в процессе работы коммутируются с выводами микроконтроллера при условии, что для этих выводов включен режим альтернативной функций. В связи с этим периферийный блок может передавать информацию на несколько выводов одновременно. В то же время прием информации может осуществляться только с одного вывода, во избежание конфликтов уровней сигналов (для этого дополнительно предусмотрена система приоритета альтернативных функций). Количество выводов, сигналы с которых могут быть переданы на периферийный блок, для каждого блока различно (от одного до трех).

Для примера, рассмотрим схему взаимодействия входа RxD блока UART2 и выводов микроконтроллера, запрограммированных на прием внешних сигналов (см. рисунок 10.3).

Блок UART2 может принимать информацию с одного из выводов F11, D0 или F4. Режим альтернативной функции может быть включен как для одного вывода (на который поступают внешние данные), так и для всех трех. Номер альтернативной функции UART_RxD2 для каждого из выводов разный.

Варианты программирования выводов для приема данных:

1 Прием данных предполагается осуществлять через вывод F11. Альтернативная функция UART_RxD2 является первой и имеет наивысший приоритет. Для выбора этой функции в регистре GPIOCTLF в поле Pin11 следует записать значение 00h (значение после сброса). Состояния полей Pin0 (регистр GPIOCTLD) и Pin4 (регистр GPIOCTLF) не важно. Т. е. даже если одновременно будут включены функции UART_RxD2 для вывода F11 и любого из двух других, на вход блока UART2 будет скоммутирован вывод F11.

2 Прием данных предполагается осуществлять через вывод D0. Альтернативная функция UART_RxD2 является второй и имеет средний приоритет. Для выбора этой функции в регистре GPIOCTLD в поле Pin0 следует записать значение 01h.

Состояние поля Pin4 (регистр GPIOCTLF) не важно, а вот состояние поля Pin11 (регистр GPIOCTLF) не должно быть 00b (т. е. для вывода F11 должна быть выбрана любая альтернативная функция кроме UART_RxD2). Только в этом случае вывод D0 будет скоммутирован на вход блока UART2.

Если для выводов F11 и D0 одновременно выбрана функция UART_RxD2, то согласно приоритету, на вход блока UART2 будет скоммутирован вывод F11 (даже в случае, если режим альтернативной функции этого вывода выключен) и соответственно данные с вывода D0 не будут приняты.

3 Прием данных предполагается осуществлять через вывод F4. Альтернативная функция UART_RxD2 является третьей и имеет низший приоритет. Для выбора этой функции в регистре GPIOCTLF в поле Pin4 следует записать значение 10h.

При этом состояние поля Pin11 (регистр GPIOCTLF) не должно быть 00b и состояние поля Pin0 (регистр GPIOCTLD) не должно быть 01b (т. е. для выводов F11 и D0 должны быть выбраны любые альтернативные функции кроме UART_RxD2). В этом случае вывод F4 будет скоммутирован на вход блока UART2.

Если для выводов F4 и F11 и/или D0 одновременно выбрана функция UART_RxD2, то на вход блока UART2 будет скоммутирован вывод, альтернативная функция которого имеет более высокий приоритет и соответственно данные с вывода F4 не будут приняты.

Указанные правила программирования выводов для приема данных распространяются на все периферийные блоки.

Для разрешения работы вывода необходимо установить соответствующий бит в регистре GPIODENi. Если бит сброшен, то вывод находится в третьем состоянии.

Выводы A1 – A7 могут также использоваться как входы аналогового компаратора.

Ко всем площадкам выводов подключены входные фильтры. На рисунке 10.4 показана структурная схема фильтра нулевого вывода порта A (в названии регистров присутствует «0», указывая на управляющие регистры порта A).

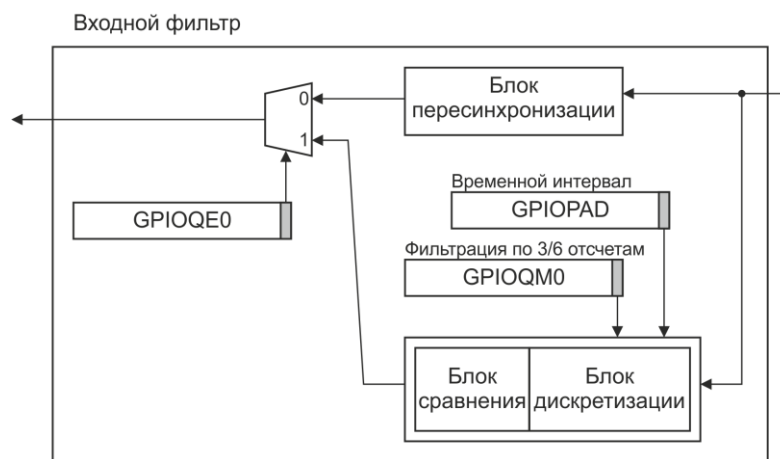


Рисунок 10.4 – Структурная схема фильтра нулевого вывода порта А

Входной сигнал с вывода порта может приниматься как напрямую (асинхронный режим), так и пересинхронизироваться (синхронизироваться с тактовой частотой работы микроконтроллера). Управление осуществляется регистром GPIOSEi. Дополнительно есть возможность включения накопления трех или шести отсчетов входного сигнала для помехоустойчивости вывода. Если результаты всех отсчетов совпадают, сигнал передается дальше по схеме, в противном случае состояние сигнала не меняется. Временные интервалы между отсчетами задаются в количестве тактов системной частоты посредством регистров GPIOQPAD и GPIOQPEN. Временной интервал задается один для всех выводов порта.

Включение фильтра и задание режима его работы осуществляется посредством регистров GPIOQEi, GPIOQMi.

Схема вывода позволяет также осуществлять гибкое управление прерываниями и задавать, по какому аппаратному событию генерировать прерывание (по какому фронту или уровню). При возникновении прерывания в регистре INTSTATUS устанавливается соответствующий флаг, и выставляется прерывание в контроллере прерываний NVIC. Прерывание может быть сгенерировано программно записью единицы в соответствующий бит регистра INTSTATUS.

Прерывание может быть сброшено программно записью единицы в соответствующий бит регистра INTCLEAR. Для разрешения прерывания вывода порта следует записать единицу в соответствующий выводу бит регистра INTENSET, а для запрета прерывания – единицу в бит регистра INTENCLR.

Для задания события, по которому генерируется прерывание, используются регистры INTTYPESET и INTPOLSET, а для сброса настроек – INTTYPECLR и INTPOLCLR, соответственно.

Для управления состоянием выводов порта дополнительно используется механизм маскирования. Он позволяет устанавливать желаемый уровень сигнала на нужном выводе, не затрагивая состояние других выводов. 16-разрядный порт условно разбивается на старший и младший байты; для доступа по маске к младшему байту используется массив регистров MASKLOWBYTE, а к старшему – MASKHIGHBYTE.

Каждый массив состоит из 256 регистров, каждый регистр имеет порядковый номер (от 00h до FFh), который является маской. Так, например, для порта А выделены две области памяти с адресами 80010400h – 800107FCh для младшего байта и 8001_0800h – 80010BFCh для старшего. Биты адреса с 9 по 2 являются маской. Таким образом, адресу 80010400h соответствует маска 00h (MASKLOWBYTE[0x00]), адресу 80010404h – 01h (MASKLOWBYTE[0x01]) и т. д.

Для того, чтобы изменить состояние выводов порта с использованием маски, нужно записать новое значение в соответствующий элемент массива (MASKLOWBYTE или MASKHIGHBYTE) с порядковым номером, совпадающим со значением маски.

Разряды порта, закрытые «нулями» маски, останутся неизменными, а остальные примут новые значения. На рисунке 10.5 показан механизм маскирования младшего байта порта А.

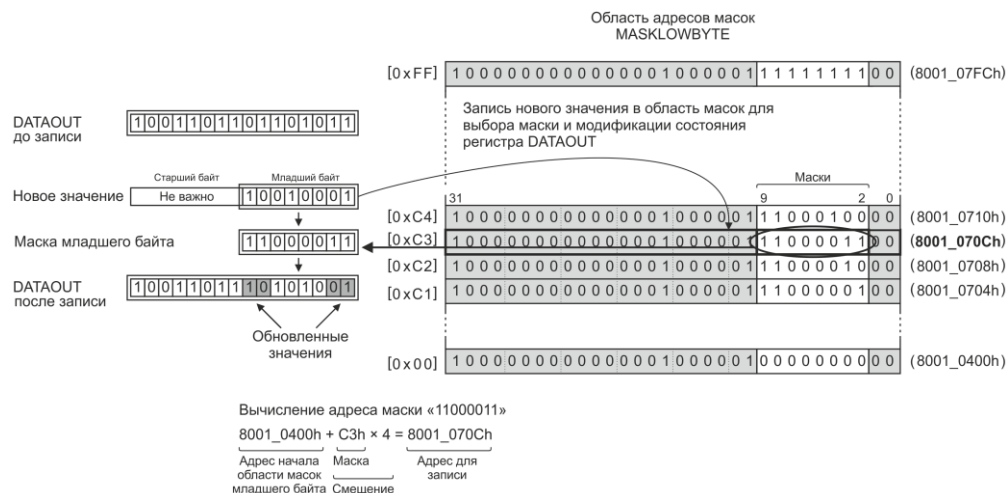


Рисунок 10.5 – Механизм изменения состояния младшего байта порта А с маскированием

Для изменения 0, 1, 6 и 7 битов регистра порта нужно использовать маску 11000011b. Эта маска является частью (биты с 9 по 2) адреса 8001070Ch (дополнительно механизм вычисления маски указан на рисунке 10.5). Новое значение XX90h данных, которое требуется передать в порт (при этом старший байт числа не важен), нужно записать в ячейку с адресом 8001070Ch. Далее это значение будет аппаратно маскировано и размещено в регистре порта DATAOUT.

Аналогично выполняется маскирование старшего байта (см. рисунок 10.6). Разница лишь в том, что в данном случае берется старший байт нового значения, а младший не важен.

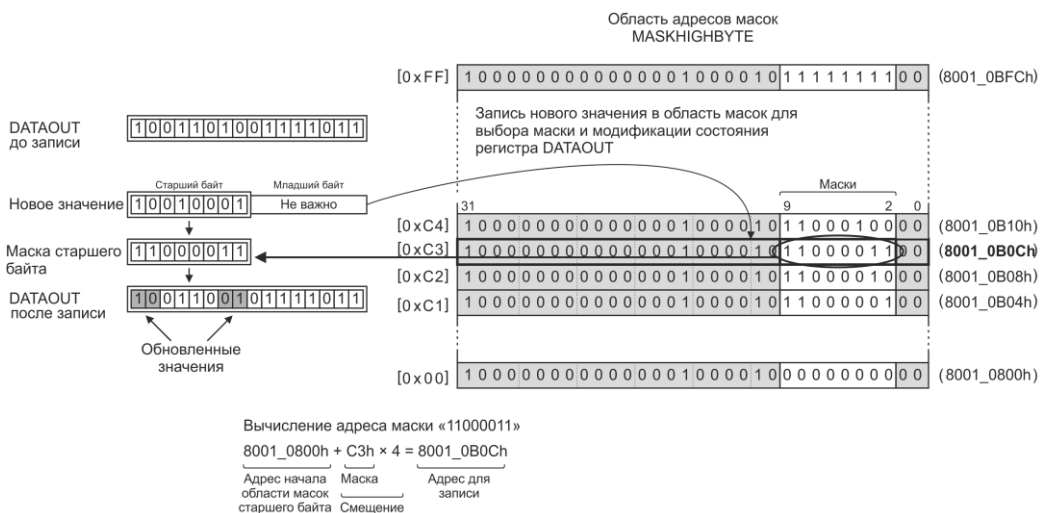


Рисунок 10.6 – Механизм изменения состояния старшего байта порта А с маскированием

11 Таймеры

Микроконтроллер содержит три блока 32-разрядных таймеров. Все блоки идентичны.

Счетчик таймера работает по системному тактовому сигналу. Кроме этого таймер может управляться внешним сигналом, а также синхронизироваться по внешнему сигналу. На рисунке 11.1 представлена функциональная схема таймера.

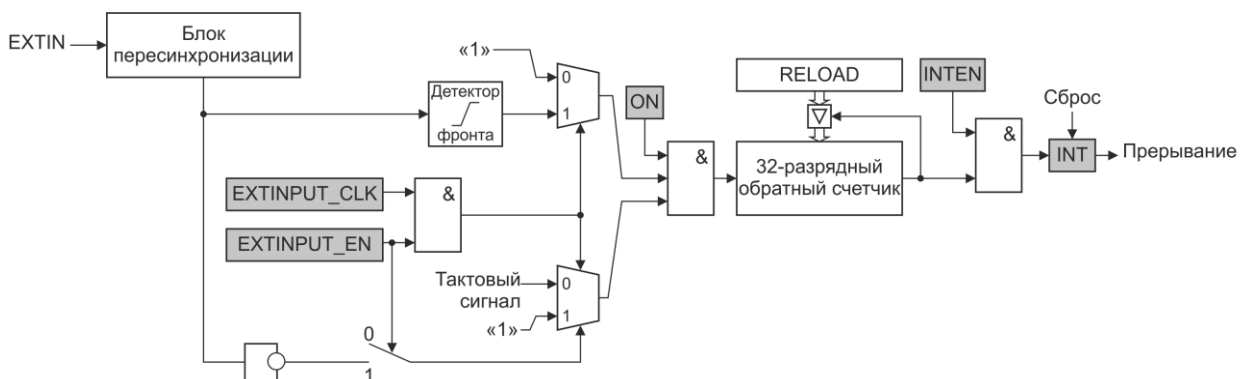


Рисунок 11.1 – Функциональная схема таймера

Управление таймером осуществляется посредством регистра CTRL. Начальное значение задается регистром VALUE. Для включения таймера нужно установить бит ON. Счетчик таймера декрементируется от значения заданного регистром VALUE до нуля на частоте тактового сигнала микроконтроллера. По достижении нуля счетчик таймера загружается значением, заданным регистром перезагрузки RELOAD, и если разрешено битом INTEN, генерируется прерывание. При возникновении прерывания устанавливается флаг INT в регистре INTSTATUS_INTCLEAR.

Если установлен бит EXTINPUT_EN, то счетчик таймера работает на частоте тактового сигнала микроконтроллера, только если сигнал на входе EXTIN имеет уровень логической единицы.

Если одновременно установлены биты EXTINPUT_EN и EXTINPUT_CLK, то тактирование счетчика таймера происходит по положительному фронту внешнего сигнала, приходящего на вход EXTIN. При этом частота внешнего сигнала должна быть как минимум в два раза меньше частоты системного тактового сигнала.

Для каждого таймера имеется свой внешний вход синхронизации EXTIN. Для таймера 0 это Timer_IN0, для таймеров 1 и 2 – Timer_IN1 и Timer_IN2, соответственно, которые являются альтернативными функциями различных выводов микроконтроллера. Дополнительно на внешний вход синхронизации таймера 2 подключены сигналы блока RTC с частотой 1 Гц и 1 кГц. Выбор сигнала внешней синхронизации для таймера 2 осуществляется посредством регистра TEST_PWM_MUX.

12 Блоки захвата

В микроконтроллере реализованы шесть блоков захвата. Все блоки идентичны.

Блоки захвата используются для:

- вычисления скорости вращения вала ротора (с использованием датчиков Холла);
- вычисления промежутков времени между срабатыванием позиционных датчиков;
- вычисления периода и скважности импульсов.

Возможности блока захвата:

- 32-разрядный таймер, с разрешающей способностью 10 нс (на 100 МГц);
- четыре 32-разрядных регистра захвата времени;
- выбор полярности фронта для обработки каждого из четырех последовательных событий;
- источники прерываний по каждому из четырех событий;
- однократный захват значений времени до четырех событий;
- режим циклической работы по событиям, с переписыванием значений (кольцевой буфер);
- режимы захвата абсолютного и относительного значений времени;
- альтернативный режим работы, если не задействована функция захвата времени – одноканальный выход ШИМ.

Функциональная схема блока захвата представлена на рисунке 12.1.

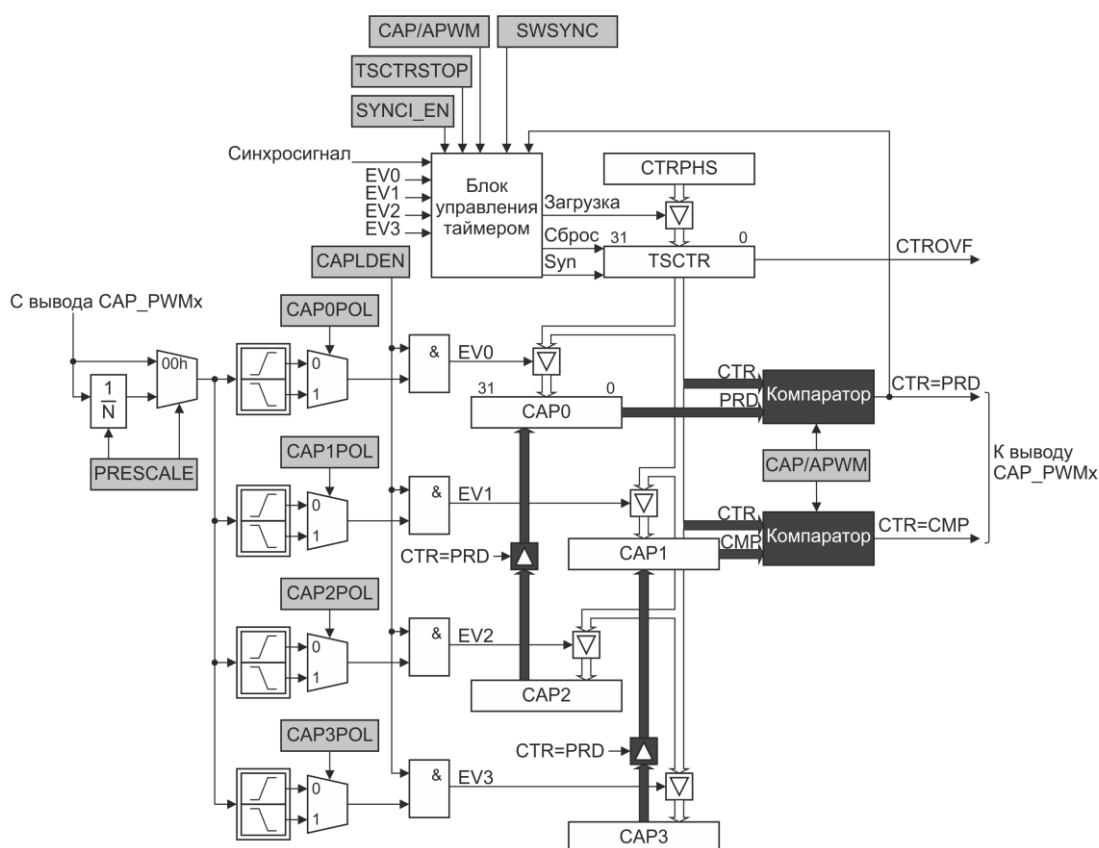


Рисунок 12.1 – Блок захвата

Реализована возможность организации двух вариантов схемы синхронизации блоков захвата с помощью переключателя, управляемого полем CAPSYNCSSEL регистра PWM_CTRL, см. рисунок 14.3. Управление синхронизацией осуществляется полями SWSYNC, SYNCO_SEL и SYNCL_EN регистра ECCTL1.

Для начала работы с блоками захвата необходимо установить соответствующие биты в поле CAPRST регистра PER_RST1.

Каждый блок захвата имеет один вывод CAP_PWMx (x – номер блока захвата от 0 до 5), соединенный с выводом микроконтроллера (альтернативная функция). В зависимости от режима работы блока захвата вывод является входом внешнего события или выходом генерируемого сигнала ШИМ.

Далее по тексту названия сигналов и регистров соответствуют указанным на рисунке 12.1.

12.1 Режим захвата времени

Режим захвата времени выбран по умолчанию. Вывод CAP_PWMx функционирует как вход.

Предварительный делитель

В случае, если события на входе CAP_PWMx приходят слишком часто и требуется уменьшить их частоту, используется предварительный делитель событий (импульсов), состоящий из собственно делителя и мультиплексора. В предварительном делителе используется счетчик, который производит выборку одного события из каждых 2 – 63 входных. Значение делителя задается полем PRESCALE регистра ECCTL0. В случае, если задано значение 00h, то делитель выключен и входной сигнал поступает на детекторы фронта напрямую.

Для примера, на рисунке 12.2 показаны несколько вариантов сигналов на выходе делителя, в зависимости от заданного значения N.

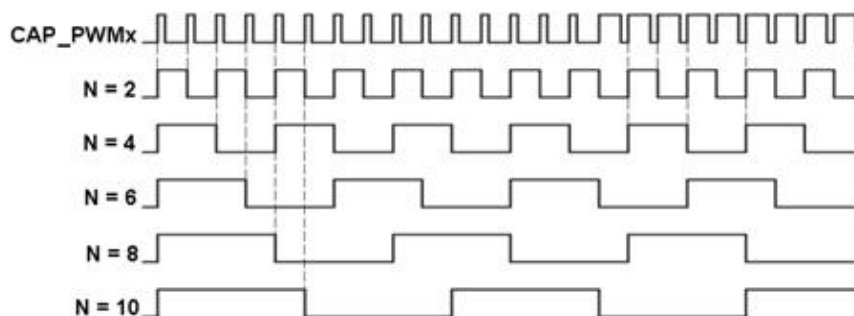


Рисунок 12.2 – Формы сигналов на выходе делителя в зависимости от значения N

Захват значения таймера

С выхода делителя сигнал поступает на четыре детектора фронта, каждый из которых управляется соответствующим битом CAPnPOL. Далее, если установлен бит CAPLDEN и обнаружен заданный фронт сигнала, формируется соответствующее событие. Возможно одновременное формирование до четырех событий (EV0 – EV3), и по переднему фронту каждого события происходит захват значения таймера TSCTR в соответствующий регистр захвата (CAP0 – CAP3).

Регистр захвата перезаписывается новым захваченным значением каждый раз при возникновении соответствующего события.

Однократный захват

Однократный захват выбирается битом CONT/ONESHT регистра ECCTL1 и включается записью единицы в бит REARM с последующей установкой бита CAPLDEN (установится аппаратно). В этом режиме происходит запуск двухразрядного счетчика событий EV0 – EV3. Количество подсчитываемых событий от одного до четырех задается полем STOP_WRAP. Подсчитывается каждое из сформированных событий и одновременно происходит захват значения таймера в соответствующие регистры захвата. Как только количество событий совпадет со значением STOP_WRAP события EV0 – EV3 больше не формируются, значение таймера захватывается регистрами CAP0 – CAP3 и

далее регистры не перезаписываются.

Для повторного запуска следует записать единицу в бит REARM (это обнулит счетчик и включит режим), после чего – разрешить формирование событий EV0 – EV3 установкой бита CAPLDEN.

Циклический захват

Циклический захват выбран по умолчанию. После установки бита CAPLDEN начинается обработка событий EV0 – EV3 и захват значения таймера. Количество подсчитываемых событий от одного до четырех задается полем STOP_WRAP.

Регистр захвата перезаписывается новым захваченным значением каждый раз при возникновении соответствующего события.

Примечание – В обоих режимах захвата значение поля STOP_WRAP не оказывает никакого влияния на счетчик и состояние бита CAPLDEN .

Таймер

Таймер представляет собой 32-разрядный счетчик, работающий на системной частоте. Контроль работы таймера осуществляет блок управления таймером. Счетчик таймера включается битом TSCTRSTOP и инкрементируется, начиная со значения 00000000h до значения FFFFFFFFh, после чего сбрасывается.

Чтобы синхронизировать работу таймера с другими блоками, счетчик таймера может быть в любой момент загружен новым значением, которое предварительно записывается в теневой регистр CTRPHS. Загрузка может быть активирована как программно – запись единицы в бит SWSYNC регистра ECCTL1, так и аппаратно – приход синхроимпульса по входу SYNCI. Разрешение синхронизации осуществляется установкой бита SYNCIEN.

Блок захвата также может генерировать сигнал синхронизации SYNCO. Источник выбирается полем SYNCOSSEL, либо используется SYNCI или SWSYNC, либо событие CTR = PRD генерации ШИМ.

Входы и выходы синхронизации блоков захвата подключены по цепочке, см. рисунок 14.3.

Таймер может быть сброшен (с предварительным захватом его значения) при формировании событий EV0 – EV3. Указать событие можно установкой соответствующего бита CTRRSTn (n от 0 до 3) в регистре EECTL0. Так, например, если установлен бит CTRRST2, то при формировании события EV2 произойдет захват значения таймера в регистр CAP2 и сброс таймера.

Регистры CAP0 – CAP3

32-разрядные регистры, сохраняющие (захватывающие) значение счетчика таймера в момент появления положительного фронта сигнала события EVn (n от 0 до 3), доступны только для чтения.

12.2 Режим работы «генератор ШИМ»

Выбирается установкой бита CAP/APWM в регистре ECCTL1. Вывод CAP_PWMx функционирует как выход. Блок захвата в этом случае используется как одноканальный 32-разрядный генератор ШИМ-сигнала.

Таймер и регистры захвата

Функционирует как 32-разрядный инкрементный счетчик, работающий на системной частоте. После включения счетчик таймера считает от значения 00000000h до значения, которое задается регистром CAP0. Как только значения счетчика и регистра совпадают, счетчик сбрасывается.

Регистр CAP0 является регистром периода таймера, а регистр CAP1 – регистром

сравнения. Регистры CAP2 и CAP3 являются регистрами отложенной загрузки для регистров CAP0 и CAP1, соответственно. Все регистры доступны как для записи, так и для чтения.

Запись в регистр CAP0 является мгновенной загрузкой, которая аппаратно дублируется записью в регистр CAP2. Аналогично для пары CAP1, CAP3.

Запись в регистры CAP2 и CAP3 является отложенной загрузкой. Как только значение счетчика таймера достигает значения периода CAP0, возникает событие $CTR = PRD$, по которому происходит сброс таймера и перегрузка значений из CAP2 и CAP3 в регистры CAP0 и CAP1 (на рисунке 12.1 отмечено стрелками черного цвета).

Регистры CAP0 и CAP1 должны быть обязательно инициализированы до начала запуска таймера. При дальнейшей работе можно изменять значения только регистра отложенной загрузки.

Генерация ШИМ

Для разрешения вывода ШИМ-сигнала следует установить значение 01h в поле SYNCO_SEL регистра EECTL1.

После инициализации регистров CAP0 и CAP1 запускается счетчик таймера. Текущее значение счетчика CTR посредством двух компараторов сравнивается одновременно со значением PRD регистра периода CAP0 и значением CMP регистра сравнения CAP1 (на рисунке 12.1 отмечено черным цветом).

Как только возникает событие $CTR = CMP$, сигнал на выходе CAP_PWMx переводится в ноль. Далее сигнал удерживается в нуле до тех пор, пока счетчик таймера не достигнет значения периода. При возникновении события $CTR = PRD$ сигнал переводится в единицу. Одновременно с этим происходит сброс таймера и перезагрузка регистров CAP0 и CAP1. Управлять полярностью сигнала можно битом APWMPOL. На рисунке 12.3 представлен пример формирования ШИМ-сигнала с активным высоким уровнем сигнала (по умолчанию $APWMPOL = 0$).

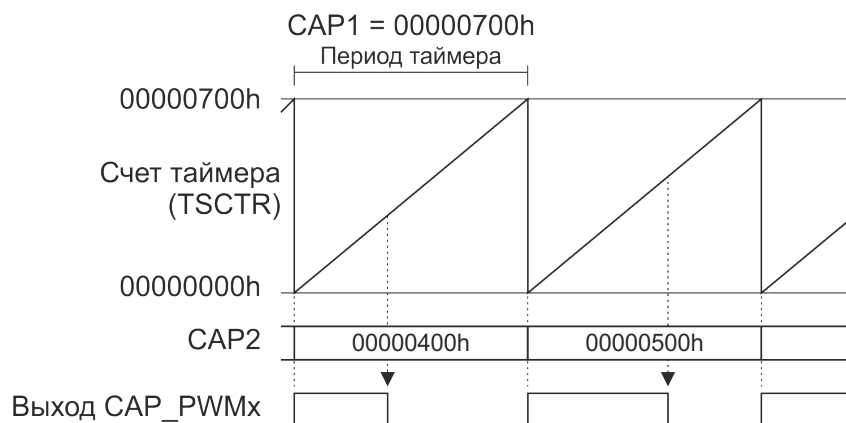


Рисунок 12.3 – Пример формирования ШИМ-сигнала

На рисунке 12.3 период таймера задан как $CAP0 = 000007000h$. Начальное значение сравнения $CAP1 = 00000400h$. Пока таймер считает, в регистр CAP2 загружается новое значение 00000500h для отложенной загрузки. По достижении значения сравнения сигнал на выходе CAP_PWMx переводится в низкий уровень. По окончании периода происходит сброс таймера и загрузка значения 00000500h (из регистра CAP2) в регистр CAP1 и перевод сигнала на выходе CAP_PWMx в высокий уровень.

Таким образом, можно достаточно гибко управлять как длительностью импульсов, изменяя период работы таймера, так и скважностью при постоянном периоде.

Прерывания

Источники прерываний блока захвата:

- события EV0 – EV3;
- переполнение счетчика таймера (CTROVF);
- события CTR = PRD;
- события CTR = CMP.

Каждое из семи прерываний имеет бит маски в регистре ECEINT, флаг прерывания в регистре ECFLG, бит сброса флага в регистре ECCLR и бит программного прерывания в регистре ECFRC. На рисунке 12.4 показан пример для прерывания по событию EV1.

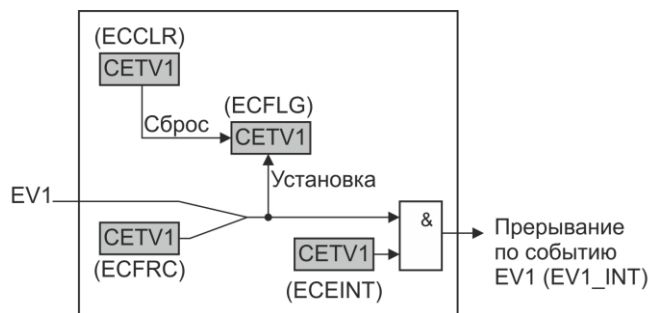


Рисунок 12.4 – Пример управления прерыванием EV1

Все прерывания по событиям поступают на блок управления прерываниями и обрабатываются, как сказано выше. При возникновении любого из этих прерываний в регистре PEINT устанавливается флаг PEINT и генерируется прерывание блока захвата CAP_INTx, см. рисунок 12.5.

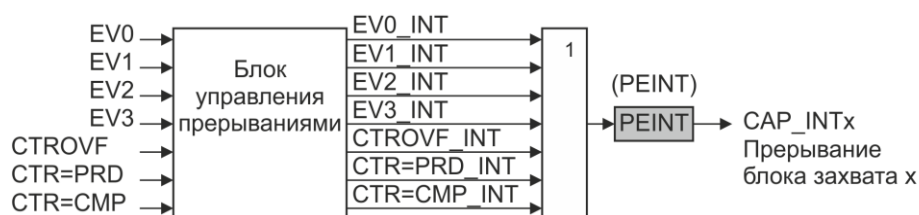


Рисунок 12.5 – Общая схема управления прерываниями

Примечание – Программа обслуживания прерывания должна сбрасывать флаг PEINT во избежание повторного обслуживания прерывания от блока захвата. Для сброса флага следует записать единицу в нулевой бит регистра PEINT.

13 Модули квадратурных декодеров

В микроконтроллере реализовано два модуля квадратурных декодеров.

Квадратурный декодер преобразует цифровой сигнал с датчика положения вала, позволяя вычислять скорость, направление вращения, а также текущее положение вала.

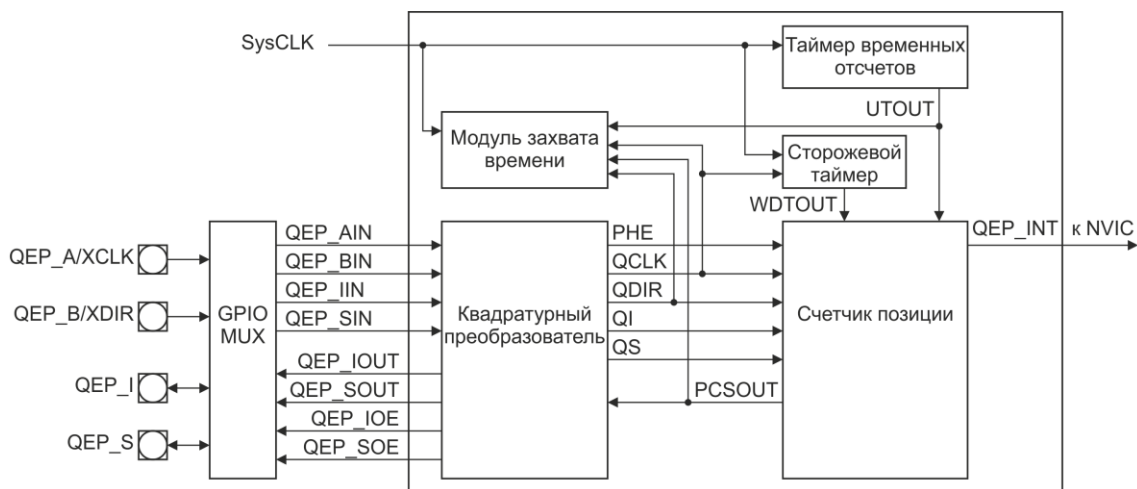


Рисунок 13.1 – Схема квадратурного декодера с мультиплексором входных/выходных сигналов

В состав квадратурного декодера входят (см. рисунок 13.1):

- настраиваемый обработчик сигналов входов;
- квадратурный преобразователь;
- счетчик позиции/блок управления;
- модуль захвата времени;
- таймер временных отсчетов;
- сторожевой таймер.

Перед началом работы с модулем для разрешения тактового сигнала необходимо установить соответствующие биты в регистре APB_CLK_CTRL и регистре PER_RST1.

13.1 Обработчик сигналов входов

Квадратурный декодер использует два квадратурных вывода контроллера, работающих на вход. Также, имеются специальный индексный вывод и вывод стробирования, которые могут работать на вход и выход.

QEP_A/XCLK и QEP_B/XDIR – в квадратурном режиме это два входа. Сигналы на входах сдвинуты по фазе на 90 градусов и по ним можно определить скорость и направление вращения ротора (см. рисунок 13.2). В режиме счета/направления сигналы на входах используются как тактовый и сигнал направления вращения ротора, по которым также можно вычислить скорость вращения.

QEP_I – индексный вход. Сигнал на входе сигнализирует о полном обороте ротора. Позволяет сбрасывать счетчик позиции поворота ротора.

QEP_S – пользовательский вход стробирования. Сигнал на входе может сбросить или защелкнуть счетчик позиции. Применяется при использовании концевых выключателей.

Сигналы на входах могут быть проинвертированы. Инверсия включается установкой соответствующего бита в регистре QDECCTL.

Установкой бита *SWAP* можно включить обратный счет, т. е. программно подать сигнал с вывода *QEP_A* на вход *QB* квадратурного преобразователя, а сигнал с вывода *QEP_B* подать на вход *QA* (входы *A* и *B* меняются местами).

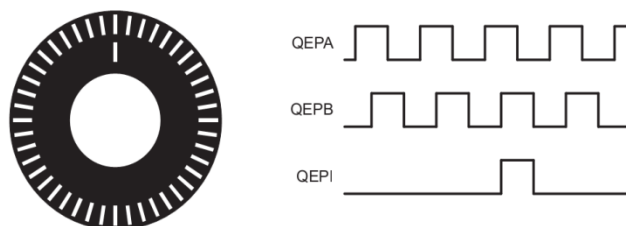


Рисунок 13.2 – Диаграмма входных сигналов

13.2 Квадратурный преобразователь

На рисунке 13.3 показана схема квадратурного преобразователя.

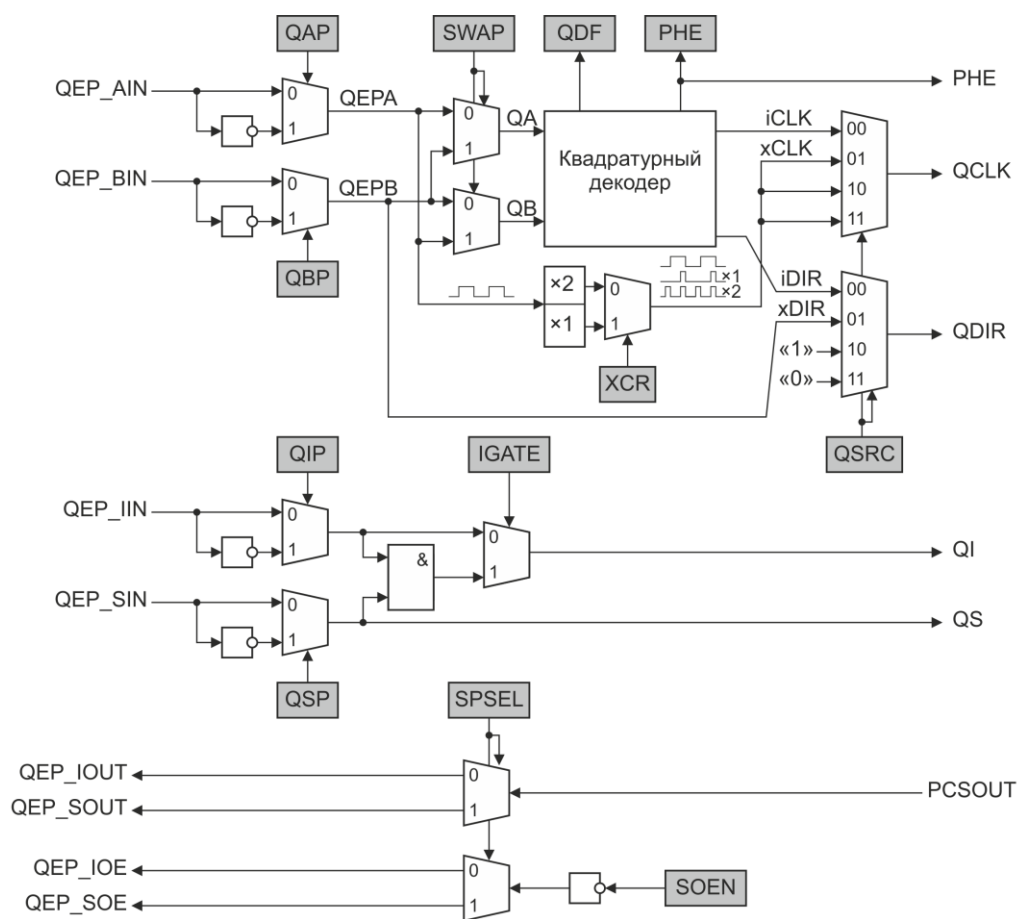


Рисунок 13.3 – Схема квадратурного преобразователя

Режимы работы

Квадратурный преобразователь поддерживает четыре режима работы:

- режим квадратурного счета;
- режим счета/направления;
- режим счета вверх;
- режим счета вниз.

Выбор режима зависит от значения поля QSRC регистра QDECCTL.

Режим квадратурного счета

Квадратурный преобразователь формирует сигнал направления вращения, тактовый сигнал и сигнал направления счета (вверх/вниз) для счетчика позиции.

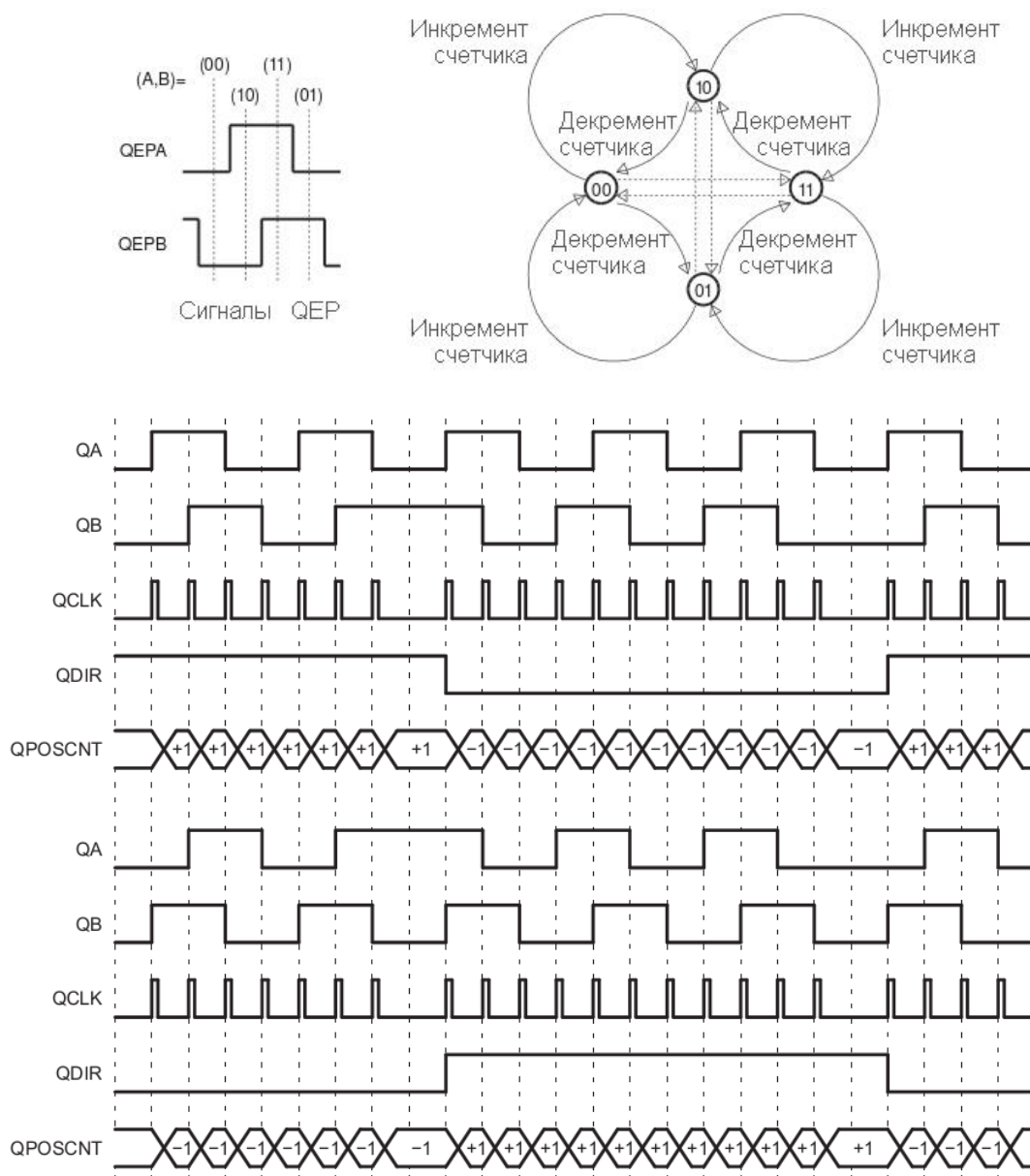


Рисунок 13.4 – Временная диаграмма и автомат состояний работы в квадратурном режиме счета

Направление вращения ротора определяется по порядку смены передних и задних фронтов на входах QEP_A и QEP_B. К примеру, если за передним фронтом сигнала на входе QEP_A следует передний фронт сигнала на входе QEP_B (см. рисунок 13.4), то направление вращения следует считать прямым, а счетчик позиции работает на увеличение. Если же за передним фронтом сигнала на входе QEP_B следует передний фронт сигнала на входе QEP_A, то направление вращения следует считать инверсным, а счетчик позиции работает на уменьшение. Если на обоих выводах зафиксировано одновременно два фронта, то такое состояние считается ошибочным.

Квадратурный преобразователь выдает четыре счетных импульса на один период входного сигнала, поскольку использует для счета передний и задний фронт сигналов.

Режим счета/направления

В этом режиме вывод QEP_A работает как вход тактовых импульсов, а вывод QEP_B – как вход задания направления счета. Счетчик позиции в этом режиме работает по каждому переднему фронту сигнала на входе QEP_A.

Режим вверх

Режим используется для вычисления частоты следования импульсов на вывод QEP_A. Фронт задается битом XCR регистра QDECCTL. Счетчик всегда работает на увеличение.

Режим вниз

Режим используется для вычисления частоты следования импульсов на выводе QEP_A. Фронт задается битом XCR. Счетчик всегда работает на уменьшение.

13.3 Счетчик позиции

Работа счетчика позиции контролируется посредством регистров QEPCTL и QPOSCTL, которыми задается режим счета, сброса и хранения, а также логика для формирования внешнего сигнала синхронизации.

Режимы сброса счетчика позиции

Счетчик позиции может накапливать результат в течение многих оборотов вала, а может подсчитывать позицию только за один оборот, сбрасываясь каждый раз по событию прихода индексной метки. В зависимости от назначения могут использоваться следующие способы сброса счетчика позиции:

- по сигналу индексации;
- по переполнению;
- только по первому сигналу индексации;
- по таймеру временных отсчетов.

Режим задается полем PCRM регистра QEPCTL.

Счетчик сбрасывается в ноль при его переполнении или при превышении значения регистра максимального значения QPOSMAX. Флаг прерывания, возникающего при переполнении счетчика, устанавливается в регистре QFLG.

Режим сброса по сигналу индексации

Режим сброса по сигналу индексации включен по умолчанию.

При получении сигнала с вывода индексации QEP_I при счете вверх, счетчик обнулится по следующему фронту сигнала тактирования QCLK. Если же сигнал индексации был получен при счете вниз, то в счетчик будет загружено значение QPOSMAX (см. рисунок 13.5).

При получении первого сигнала индексации, схема дожидается любого изменения на квадратурных входах и запоминает значение этого события – фронт, активный вывод (QEP_A или QEP_B), а также направление вращения. Этот момент времени называется маркером индексации. При появлении этого события устанавливается бит FIMF регистра QEPSTS, а направление вращения сохраняется в бите FIDF. В дальнейшем, маркер индексации можно использовать для сохранения значения счетчика.

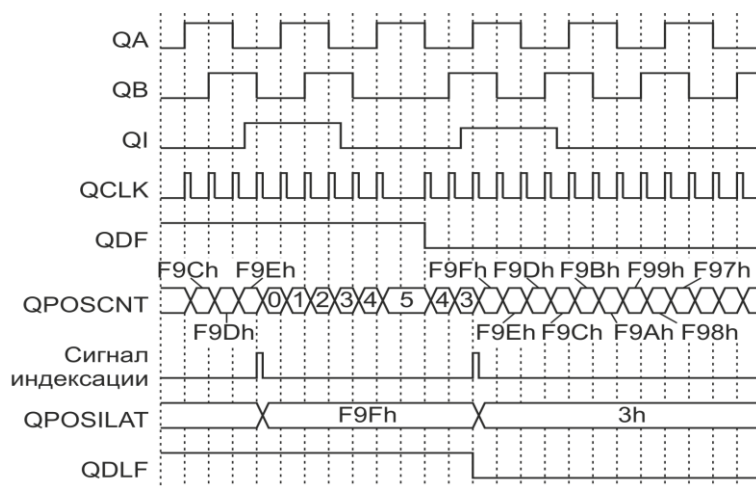


Рисунок 13.5 – Временная диаграмма сброса по сигналу индексации

По каждому сигналу индексации, включая маркер индексации, содержимое счетчика сохраняется в регистре QPOSILAT, а направление вращения в бите QDLF регистра QEPSTS. Если при сохранении значение счетчика QPOSCNT не равно ни нулю, ни значению QPOS MAX, то выставляется флаг ошибки счетчика позиции (бит PCEF в регистре QEPSTS) и флаг прерывания (бит PCE в регистре QFLG). Флаг ошибки счетчика позиции обновляется с каждым индексом, а флаг прерывания может быть сброшен только программно.

Поле настройки события индексации для сохранения счетчика позиции IEL (регистр QEPCTL) игнорируется. Также только в этом режиме могут устанавливаться флаг ошибки счетчика позиции PCEF и флаг соответствующего прерывания.

Режим сброса по переполнению

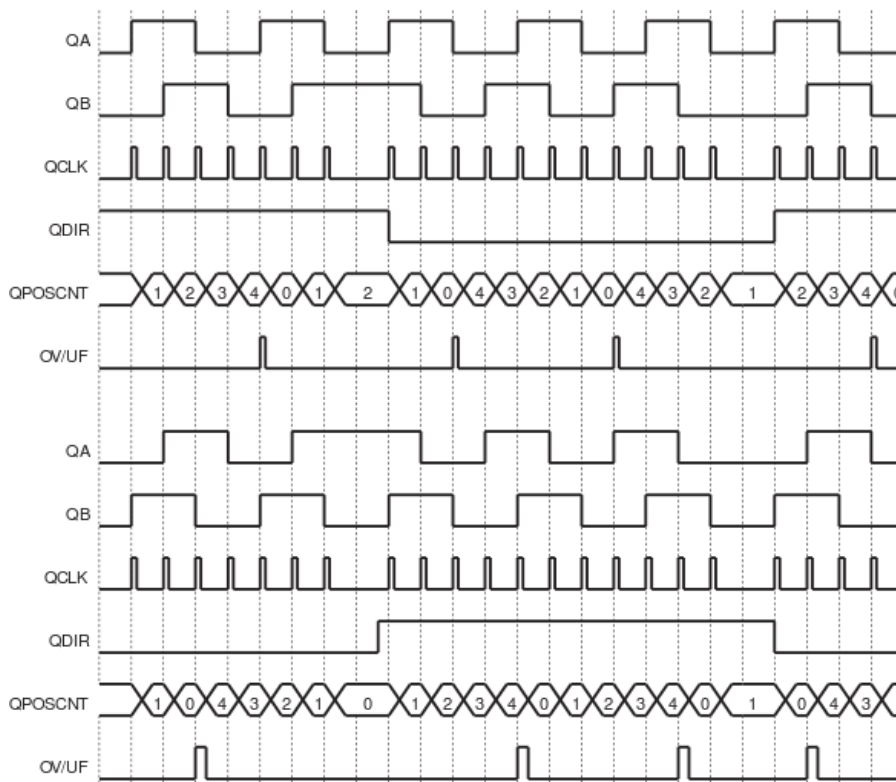


Рисунок 13.6 – Временная диаграмма сброса по сигналу переполнения

Максимальное значение счетчика позиции задается регистром QPOS MAX. Если счетчик считает вверх, и достигнуто максимальное значение, то со следующим тактом синхросигнала счетчик обнулится. Если счетчик считает вниз, и достигнуто значение нуля, то со следующим тактом синхросигнала в счетчик будет загружено значение QPOS MAX. Сброс по событию индексации не производится.

Получение значений маркера индексации происходит аналогично тому, как это происходит в режиме сброса по сигналу индексации. Полученные значения могут использоваться при инициализации по маркеру индексации, если в поле IEL записано значение 11b в регистре QEPCTL. Временная диаграмма сброса показана на рисунке 13.6.

Режим сброса по первому сигналу индексации

Если было получено событие индексации при счете вверх, то счетчик обнулится со следующим тактом синхросигнала. Если же событие индексации было зафиксировано при счете вниз, то со следующим тактом синхросигнала в счетчик будет загружено значение QPOS MAX. При последующем счете сброс может произойти только при достижении нуля или значения QPOS MAX (т. е. аналогично режиму сброса по переполнению), а дальнейшие возможные события получения сигнала на выводе индексации влиять на сброс не будут.

Получение значений маркера индексации происходит аналогично тому, как это происходит в режиме сброса по сигналу индексации. Полученные значения могут использоваться при инициализации по маркеру индексации, если в поле IEL записано значение 11b.

Режим сброса по таймеру временных отсчетов

В этом режиме счетчик сбрасывается в ноль или загружается значением QPOS MAX, в зависимости от текущего режима счета (задается полем QSRC регистра QDECCTL), по событию срабатывания таймера временных отсчетов. В остальном режим аналогичен режиму сброса по переполнению.

Также возможно настроить сохранение значения счетчика QPOSCNT в регистр QPOSLAT перед сбросом, для этого необходимо включить модуль захвата, установив бит CEN в регистре QCAPCTL. Этот режим удобен для измерения частоты.

Сохранение счетчика позиции

Внешние входы индексации и стробирования можно запрограммировать на формирование событий для сохранения значения счетчика позиции в регистры QPOSILAT и QPOSSLAT.

Сохранение по событию индексации

В некоторых задачах не требуется сбрасывать счетчик позиции по каждому сигналу индексации, и вместо этого может потребоваться увеличить разрядность счетчика до 32 бит (режимы, задаваемые значениями PCRM равными 01b и 10b). В этом случае бит QDLF (направление вращения) в регистре QEPSTS будет перезаписываться по каждому сигналу индексации, а счетчик будет сохранять значение по следующим событиям индексации:

- по переднему фронту сигнала индексации (IEL = 01b);
- по заднему фронту сигнала индексации (при IEL = 10b);
- по маркеру индексации (при IEL = 11b).

Сохранение значения счетчика по маркеру индексации будет производиться только в присутствии сигнала индекса и по событию, эквивалентному сохраненному при первой индексации по маркеру. Если направление вращения изменится, то сохраненное в маркере значение типа фронта меняется на обратное. Это сделано с целью привязки индекса к

квадратурному сигналу QA/QB, а также для более точной обработки индексации, чтобы исключить влияние ширины импульса на выводе индексации.

При сохранении значения счетчика в регистр QPOSILAT формируется флаг IEL прерывания индексации в регистре QFLG. В режиме сброса по сигналу индексации (PCRM = 00h) значение поля IEL в регистре QEPCTL игнорируется.

Сохранение по событию стробирования

Значение счетчика сохраняется в регистр QPOSSLAT по каждому переднему фронту сигнала на входе QEP_S, если сброшен бит SEL в регистре QEPCTL. Если же бит SEL установлен, то сохранение в QPOSSLAT происходит по переднему фронту сигнала строба на входе QEP_S при прямом направлении вращения и по заднему фронту для обратного вращения. При каждом сохранении счетчика в QPOSSLAT устанавливается флаг прерывания SEL.

Инициализация счетчика позиции

Счетчик событий может быть проинициализирован программно или по событиям:

- событие индексации;
- событие стробирования.

Входной сигнал индексации (QEPi) может использоваться для инициализации счетчика по переднему и заднему фронту. Если поле IEl = 10b, то счетчик загружается значением QPOSINIT по переднему фронту сигнала индексации. Аналогично, если IEl = 11b, то счетчик загружается значением QPOSINIT по заднему фронту сигнала индексации.

Если поле SEI = 10b, то счетчик загружается значением QPOSINIT по переднему фронту сигнала стробирования на входе QEP_S. Если SEI = 11b, то счетчик загружается значением QPOSINIT по переднему фронту сигнала стробирования, если идет счет вверх, и по заднему – если вниз.

Программно счетчик инициализируется при записи единицы в бит SWI регистра QEPCTL. Бит не сбрасывается автоматически, но повторная запись единицы также приведет к инициализации счетчика.

Компаратор счетчика позиции

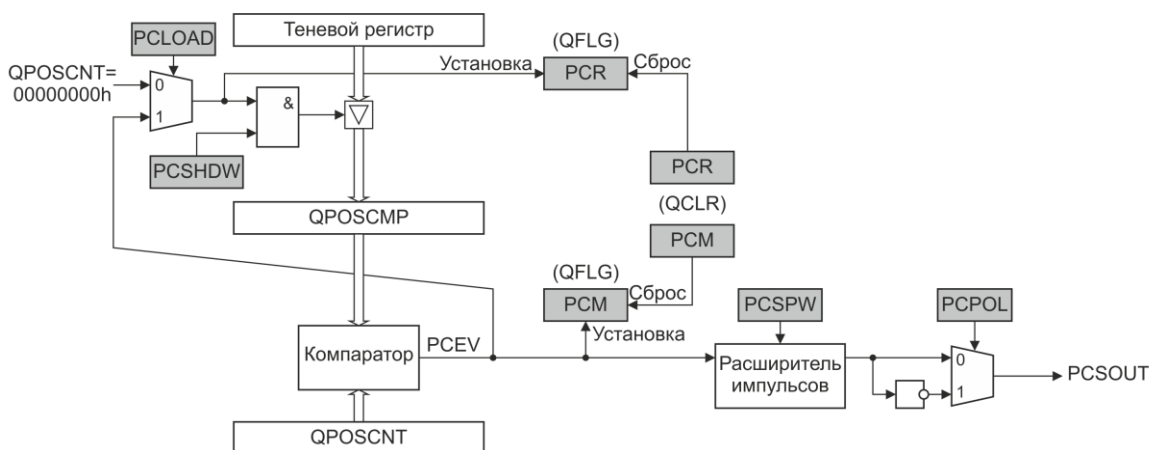


Рисунок 13.7 – Функциональная схема компаратора счетчика позиции

Компаратор (см. рисунок 13.7) сравнивает значение счетчика позиции с регистром QPOSCMP и при совпадении значений формирует прерывание, а также внешний синхросигнал, который может быть направлен на один из выводов: индексный вывод

QEP_I или вывод стробирования QEP_S. Бит SPSEL в регистре QDECCTL определяет, на какой именно вывод будет направлен сигнал синхронизации, а бит SOEN в регистре QDECCTL разрешает этому выводу работать как выход.

Регистр QPOSCMP может использовать режим отложенной загрузки, когда отложенное значение берется из теневого регистра, а если режим отложенной загрузки выключен, то запись в QPOSCMP производится сразу в активный регистр.

Отложенная загрузка происходит по следующим событиям:

- по совпадению результатов сравнения;
- по обнулению счетчика QPOSCNT.

Флаг успешного сравнения РСМ устанавливается, когда выполняется условие $QPOSCNT = QPOSCMP$, при этом также формируется синхроимпульс требуемой длительности для извещения внешнего устройства (сигнал PCSOUT). Настраиваемая длительность синхроимпульса контролируется специальной схемой задержки.

Флаг PCR готовности компаратора к отложенной загрузке значения сравнения выставляется, когда выполняется условие для отложенной записи, заданное битом PCLOAD в регистре QPOSCTL. При этом состояние флага включения режима отложенной загрузки PCSHDW (регистр QPOSCTL) не оказывает влияния на установку флага PCR и генерацию соответствующего прерывания.

13.4 Таймер временных отсчетов

Таймер, используемый для оповещения программного обеспечения о необходимости начать измерение скорости, представляет собой 32-разрядный таймер, работающий на частоте системного тактового сигнала. Включается установкой бита UTE в регистре QEPCTL. Когда значение таймера достигает порога ($QUTMR = QUPRD$), формируется прерывание и выставляется флаг UTO. Данный блок таймера может быть использован для вычисления скорости на высоких скоростях (см. рисунок 13.8), а также для сохранения счетчика позиции, регистра таймера и регистра периода в регистрах QPOSLAT, QCTMRLAT и QCPRDLAT, соответственно. Режим сохранения определяется состоянием бита QCLM в регистре QEPCTL.

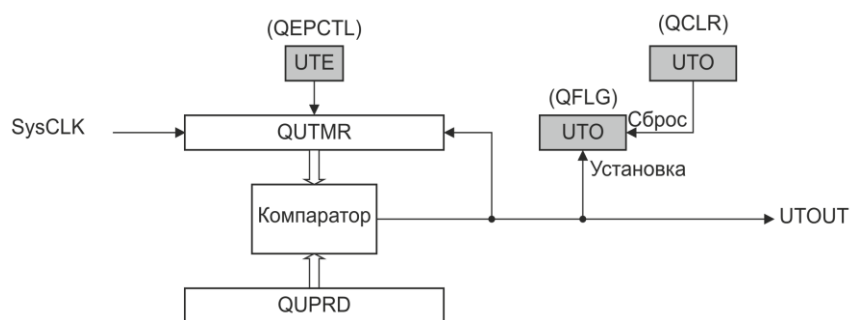


Рисунок 13.8 – Функциональная схема таймера временных отсчетов

13.5 Модуль захвата времени

Функциональная схема модуля захвата времени представлена на рисунке 13.9. Таймер использует тактовый сигнал и сигнал квадратурных событий с коэффициентами деления, программируемыми полями CCP5 и UPP5 в регистре QCAPCTL. Коэффициенты деления можно менять в процессе работы, но в этом случае может произойти событие захвата, содержащее неверные данные, которые следует игнорировать. Во избежание подобной ситуации перед изменением значений коэффициентов деления следует выключить модуль захвата (сбросить бит SEN) и снять все маски прерываний. После

изменения коэффициентов проинициализировать таймер QCTMR (записать нулевое значение), сбросить все статусы, вновь разрешить прерывания и включить модуль захвата.

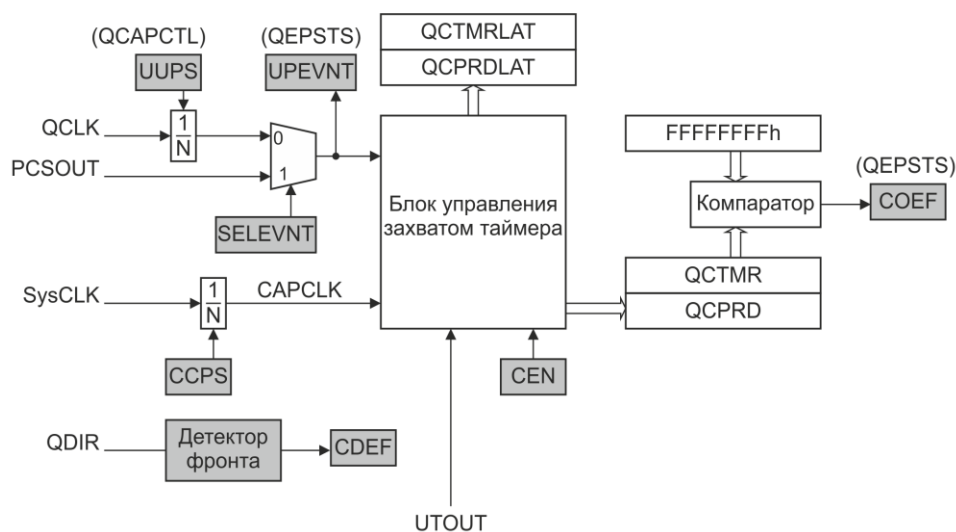


Рисунок 13.9 – Функциональная схема модуля захвата времени

Если бит SELEVNT в регистре QCAPCTL сброшен, то по деленному квадратурному событию значение таймера QCTMR загружается в регистр периода QCPRD, после чего таймер сбрасывается и устанавливается флаг UPEVNT в регистре QEPSTS, означающий обновление регистра QCPRD. Флаг сбрасывается программно записью единицы.

При установленном бите SELEVNT обновление регистра периода происходит по сигналу от выхода компаратора PCSOUT.

Значение таймера можно использовать при измерениях скорости, если:

- его значение не превысило FFFFFFFFh;
- направление вращения за время измерения не изменилось.

Если между двумя событиями UPEVNT (т. е. во время измерения) таймер QCTMR переполнился, устанавливается флаг ошибки COEF в регистре QEPSTS. Если между двумя событиями положения вала изменилось направление вращения, устанавливается флаг ошибки CDEF.

Значения таймера (QCTMR) и регистра периода (QCPRD) могут быть сконфигурированы для захвата в регистры QCTMRLAT и QCPRDLAT по событиям:

- прочитан регистр QPOSCNT;
- сработал сторожевой таймер.

Если бит QCLM сброшен, то при каждом чтении регистра счетчика позиции QPOSCNT регистры QCTMR и QCPRD загружаются в QCTMRLAT и QCPRDLAT, соответственно.

Если бит QCLM установлен, то при каждом срабатывании таймера временных отсчетов счетчик позиции, регистр таймера и регистр периода захватываются в регистры QPOSLAT, QCTMRLAT и QCPRDLAT, соответственно.

Измерения на малых скоростях вращения (низкая частота квадратурного сигнала) производятся следующим образом – таймер QCTMR, тактирующийся от системного тактового сигнала с делителем CCPS, по событию UPEVNT, сохраняет свое значение времени в регистре QCPRD, одновременно сбрасывается, и выставляет флаг UPEVNT в регистре QEPSTS, чтобы сообщить программе об окончании измерения. Событие UPEVNT возникает каждые несколько тактов QCLK, в соответствии с запрограммированным коэффициентом деления UPPS в регистре QCAPCTL. Таким

образом, зная количество квадратурных событий за измеренный отрезок времени, а также такой параметр, как количество квадратурных событий за полный оборот вала, можно вычислить скорость вращения.

Измерения на высоких скоростях (см. рисунок 13.10) могут производиться иначе. Таймер временных отсчетов формирует общую длительность измерения, счетчик позиции подсчитывает количество импульсов QCLK. Зная количество импульсов QCLK за один полный оборот, можно вычислить скорость вращения вала.

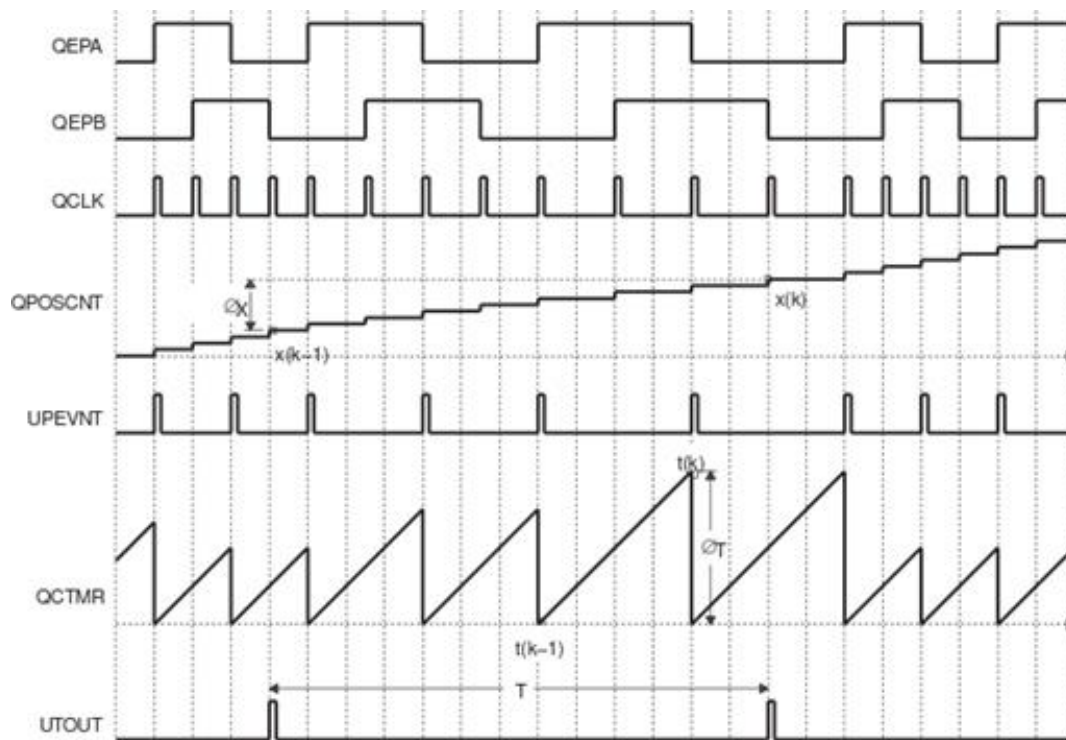


Рисунок 13.10 – Работа на высоких скоростях

Также, существует и смешанный способ измерения скорости – по заданному значению счетчика позиции, с помощью компаратора счетчика позиции, можно сформировать событие UPEVNT (необходимо установить бит SELEVNT в регистре QCAPCTL), которое, так же как и при измерениях на малых скоростях, позволит получить значение таймера QCTMR. Для использования этого способа измерения скорости необходимо разрешить прерывание по событию PCSOUT компаратора и устанавливать в этом прерывании каждый раз порог сравнения компаратора QPOSCMP на заданное количество меток вперед по сравнению с текущей позицией счетчика QPOSCNT (в зависимости от направления вращения). Тогда, устанавливая QPOSCMP дальше от QPOSCNT с увеличением скорости вращения, можно поддерживать оптимальное захватываемое время, обеспечивающее максимальную точность измерения времени для всех диапазонов вращения. Этот способ измерения наиболее сложен, но и наиболее универсален.

13.6 Сторожевой таймер

Блок квадратурного декодера содержит 32-битный сторожевой таймер, который тактируется системным тактовым сигналом, деленным на 64, и сбрасывается любым квадратурным событием (перепад на выводе QEP_A/QEP_B). Если ни одного квадратурного события не было зафиксировано до события QWDTMR = QWDPRD, сторожевой таймер формирует флаг прерывания WTO в регистре QFLG. Регистр

QWDPRD содержит значение срабатывания сторожевого таймера. Функциональная схема сторожевого таймера представлена на рисунке 13.11.

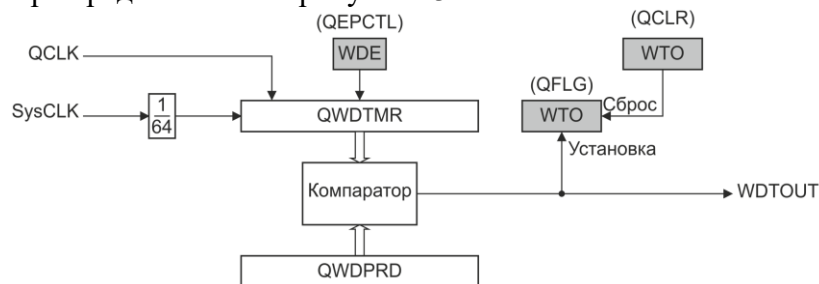


Рисунок 13.11 – Функциональная схема сторожевого таймера

13.7 Система прерываний

Блок квадратурного декодера содержит 11 источников прерываний (см. рисунок 13.12). Система прерываний состоит из регистра маски прерываний QEINT, регистра флагов прерываний QFLG, а также формирования внешнего прерывания INT по наличию активных флагов. Прерывание INT также может быть маскировано в контроллере прерывания NVIC. Сброс флагов прерываний осуществляется через регистр QCLR. Сброс флага активности прерывания INT осуществляется записью в регистр QINTCLR. Также, прерывание можно сформировать программной записью в регистр QFRC, но для этого необходимо предварительно включить счетчик позиции, установив бит QPEN в регистре QEPCTL.

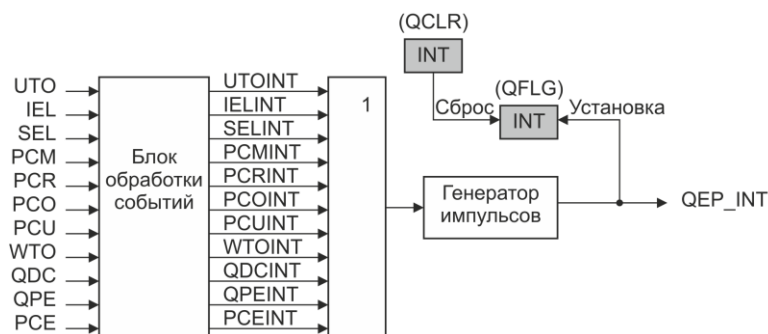


Рисунок 13.12 – Схема системы прерываний

На рисунке 13.13 показана схема формирования прерывания внутри блока обработки событий для события UTO (срабатывание таймера временных отсчетов). Схемы формирования прерываний для остальных событий идентичны.

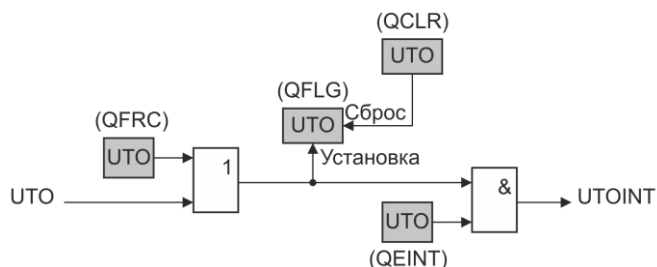


Рисунок 13.13 – Схема формирования прерывания UTOINT

14 Блоки ШИМ

Архитектура блока ШИМ разработана по принципу минимальной нагрузки на процессор, что достигается автоматизацией формирования выходных импульсов с настраиваемыми пользователем параметрами. Так, после минимальных настроек этот блок способен работать самостоятельно как выделенный канал ШИМ с собственным таймером, формируя выходные сигналы PWM_A и PWM_B.

Микроконтроллер содержит девять блоков ШИМ (первые шесть из которых PWM0 – PWM5 являются блоками ШИМ высокого разрешения) и шесть блоков захвата, объединенных общей схемой синхронизации.

Каждый блок ШИМ поддерживает следующую функциональность:

- 16-разрядный таймер;
- два вывода PWM_A и PWM_B, которые могут работать в режиме фронтальной и центрированной модуляции как полностью независимо, так и комплементарно с разделением генератором «мертвого времени»;
- выходы PWM_A и PWM_B могут управляться в зависимости от событий цифровых компараторов блока АЦП, а также от событий блока аналоговых компараторов, обеспечивая автоматический релейный режим поддержания заданной величины;
- программное управление выходами ШИМ;
- программное задание фазы счетчиков таймера для координации работы нескольких блоков ШИМ;
- аппаратный контроль фазы при координации работы нескольких блоков ШИМ;
- предотвращение наложения фронтов за счет генератора «мертвого времени» с независимой схемой задержки переднего и заднего фронтов выходного сигнала;
- сигнал аварии может переводить выходы PWM_A и PWM_B в высокое, низкое или Z-состояние;
- однократная и циклическая обработка сигналов аварии;
- все события могут инициировать прерывания, а также запускать работу секвенсоров блока АЦП;
- программируемый предделитель событий позволяет снизить нагрузку на процессор при обработке прерываний;
- ШИМ-сигнал может модулироваться высокочастотным сигналом при использовании драйверов ключей с импульсным трансформатором.

Для начала работы с блоками ШИМ для разрешения тактирования необходимо установить бит PWMEN регистра APB_CLK, а также установить биты поля PWRMST регистра PER_RST1.

Описание сигналов и выводов блока ШИМ:

- PWM_A и PWM_B – выходы ШИМ;
- TZ0 – TZ5 – входы, соединенные с выводами микроконтроллера PWM_TZ0 – PWM_TZ5, с которых принимаются сигналы аварии (общие для всех блоков ШИМ, и каждый блок может использовать или не использовать эти сигналы);
- PWM_SYNCI – вход, соединенный с одноименным выводом микроконтроллера, служащий для приема внешнего синхросигнала;
- PWM_SYNCO – внутренний сигнал синхронизации на выходе блока 0 ШИМ, являющийся синхросигналом для остальных блоков ШИМ и блоков захвата;
- PWM_SOCA и PWM_SOCB – выходные сигналы для запуска секвенсоров блока АЦП (формируются триггером событий блока ШИМ). Кроме того, эти сигналы передаются на выходы микроконтроллера PWM_An и PWM_Bn (n – номер блока ШИМ);
- TZINT, INT – внутренние сигналы прерываний.

Функциональная схема блока ШИМ показана на рисунке 14.1.

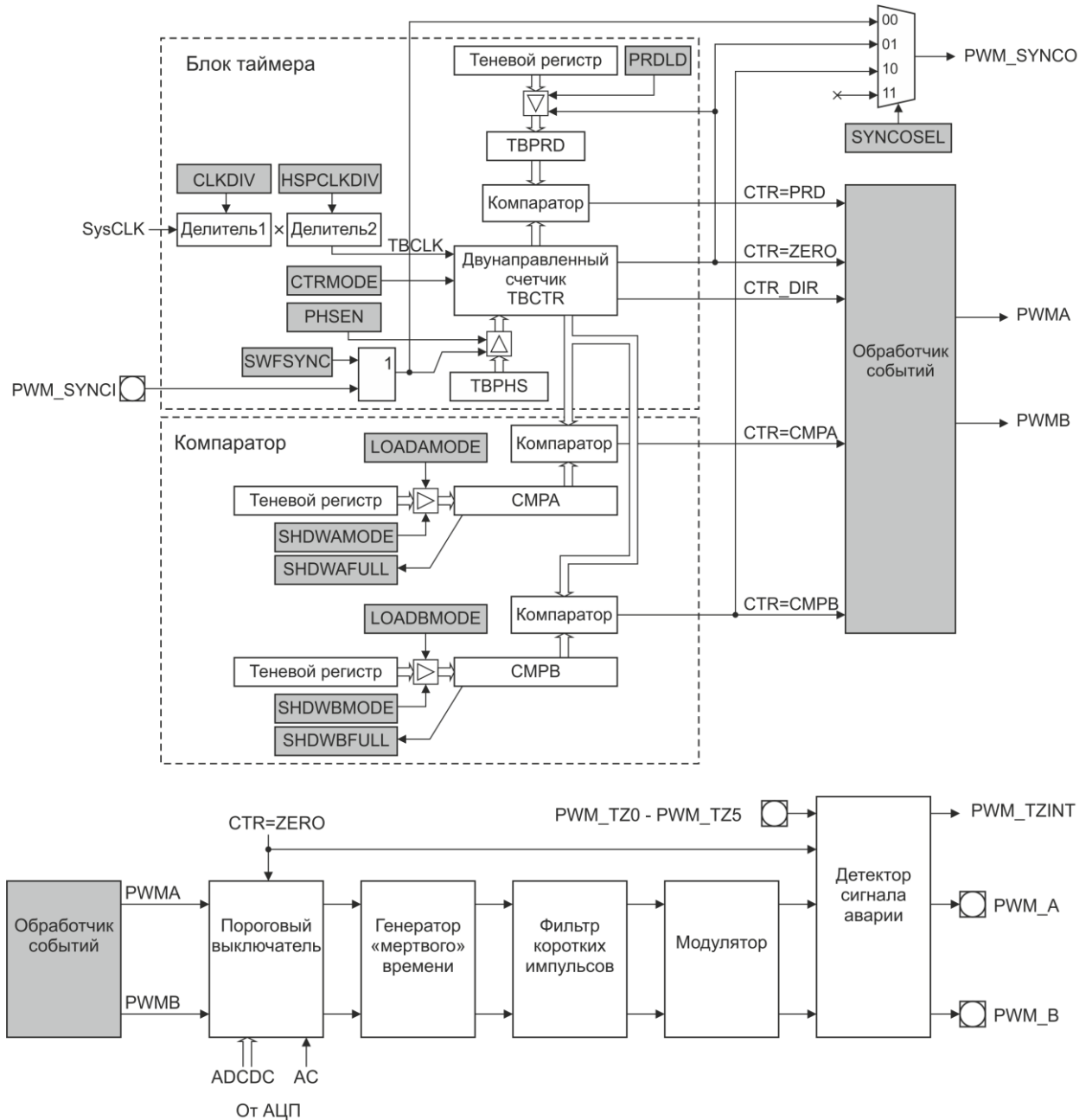


Рисунок 14.1 – Блок ШИМ

14.1 Таймер

Таймер представляет собой двухнаправленный счетчик (TBCTR), тактируемый сигналом TBCLK, который формируется на основе синхросигнала SysCLK. Частота сигнала TBCLK задается произведением коэффициентов двух делителей. Коэффициенты задаются полями CLKDIV и HSPCLKDIV регистра TBCTL. Для работы других блоков ШИМ счетчик позволяет формировать события, такие как совпадение по периоду $CTR = PRD$ ($TBCTR = TBPRD$), совпадение с нулем $CTR = Zero$ ($TBCTR = 0000h$), совпадение с регистрами $CTR = CMPA$ и $CTR = CMPB$ ($TBCTR = CMPA$ и $TBCTR = CMPB$, соответственно). Событие $TBCTR = FFFFh$ влияет только на флаг CTRMAX регистра TBSTS.

Всеми настройками работы счетчика таймера управляет регистр TBCTL.

Состояние счетчика отражают флаги регистра TBSTS.

Входной сигнал тактирования TBCLK для первого блока ШИМ формируется на основе сигнала, приходящего с вывода PWM_SYNC1 микроконтроллера. Кроме того можно программно формировать синхриимпульсы PWM_SYNC1 посредством записи единицы в бит SWFSYNC.

На выходе первого блока ШИМ формируется сигнал SYNC0, который является, синхросигналом для остальных блоков ШИМ (см. рисунок 14.2). Сигнал SYNC0 имеет три источника – сигнал на выводе PWM_SYNC1 микроконтроллера (в том числе и программно сгенерированный), событие CTR = Zero и CTR = CMPB. Выбор источника осуществляется посредством поля SYNC0SEL.

В блоке таймера находятся регистры начальной фазы счета TBPHS и периода (максимального значения счетчика) TBPRD. Регистр периода имеет теневой регистр для синхронной загрузки значения, до которого счетчик осуществляет счет. Управление загрузкой осуществляется битом PRDL.

Счетчик может работать в трех режимах счета (см. рисунок 14.2):

- вверх (от 0000h до значения TBPRD, затем сброс в 0000h и т. д.);
- вниз (от значения TBPRD до 0000h, затем загрузка значения TBPRD и т. д.);
- вверх-вниз (от 0000h до значения TBPRD, затем от значения TBPRD до 0000h и т. д.).

Параметры счета задаются полем CTRMODE.

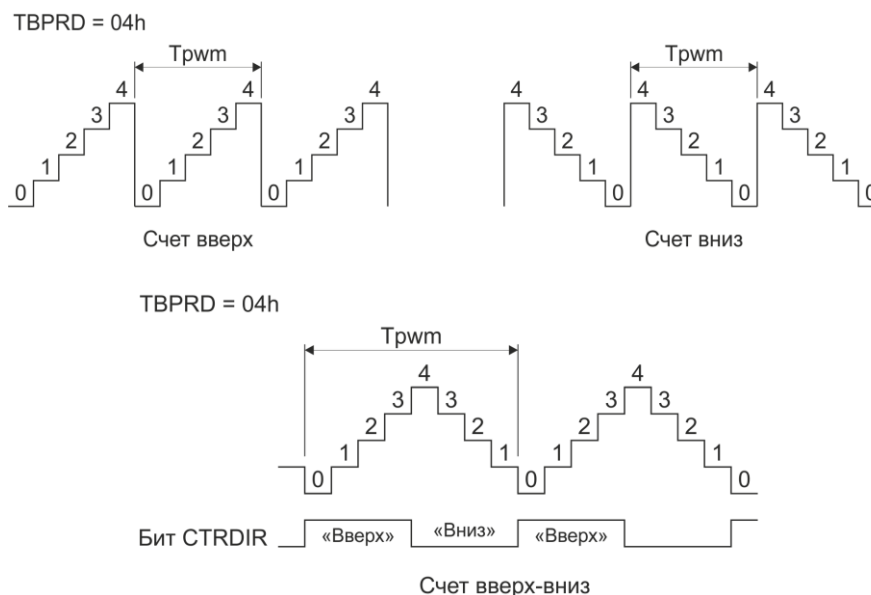
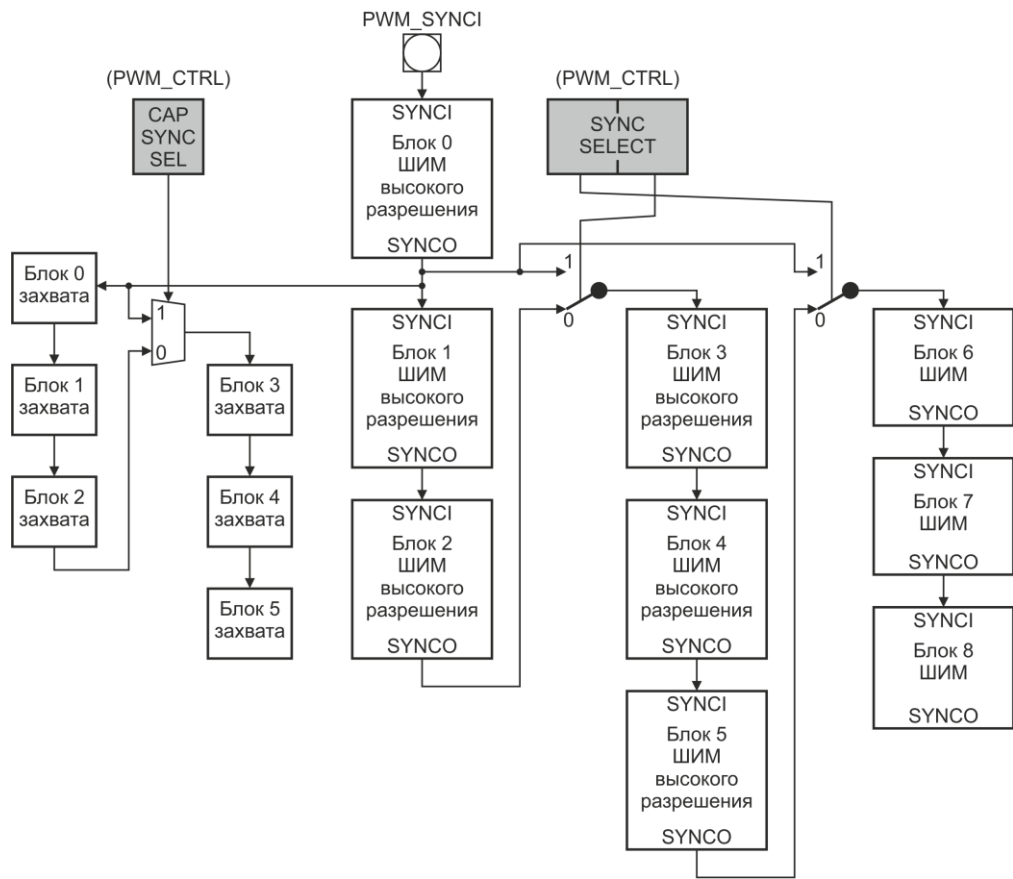


Рисунок 14.2 – Режимы работы счетчика (при значении периода 0004h, указано как T_{pwm})

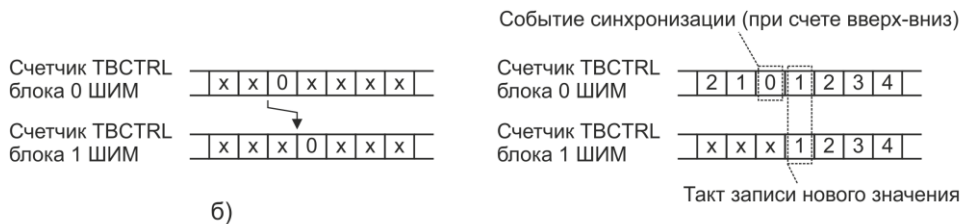
На рисунке для режима счета «вверх-вниз» дополнительно указано поведение флага CTRDIR.

Синхронизация таймеров блоков ШИМ

Реализована возможность организации четырех вариантов схемы синхронизации блоков ШИМ с помощью переключателей, управляемых полем SYNCSELECT регистра PWM_CTRL, смотри рисунок 14.3а).



а)



б)



в)

Рисунок 14.3 – Схема и особенности синхронизации модулей ШИМ

Система синхронизации таймеров включает в себя таймеры всех блоков ШИМ. Каждый блок ШИМ имеет вход синхронизации PWM_SYNCI и выход синхронизации PWM_SYNCO.

Если бит PHSEN установлен, то в счетчик таймера будет автоматически загружаться значение регистра TBPHS при выполнении каждого из условий:

- изменение входного сигнала PWM_SYNCI (в этом случае загрузка значения TBPHS в регистр TBCTR происходит на следующий такт TBCLK после поступления импульса на вход PWM_SYNCI с задержкой в два системных такта, если $TBCLK = SysCLK$, или один такт, если $TBCLK \neq SysCLK$);

- запись единицы в бит SWFSYNC (программная синхронизация), которая генерирует импульс синхронизации, аналогичный импульсу с входа PWM_SYNCI.

В режиме счета вверх-вниз необходимо запрограммировать бит PHSDIR, чтобы задать направление счета таймера после синхронизации.

Если бит PHSEN сброшен, блок ШИМ не будет реагировать на входной сигнал синхронизации, а только передавать напрямую этот сигнал на выход PWM_SYSCO, чтобы тактировать другие блоки ШИМ. Следующая особенность схемы – генерация и распространение сигнала синхронизации от блока ШИМ занимает один такт TBCLK.

К примеру, если по событию синхронизации блока 0 в счетчик блока 1 должен быть записан ноль, то этот ноль запишется только на следующий такт после события, см. рисунок 14.3б). Таким образом, при синхронизации от другого блока ШИМ нужно всегда учитывать этот такт и записывать значение фазы, следующее по порядку, в соответствии с режимом счета, см. рисунок 14.3в).

14.2 Компаратор

Компаратор – это блок, сравнивающий значение счетчика таймера с заданными значениями порогов срабатывания. Значения хранятся в регистрах CMPA и CMPB. Значения, записываемые по адресам регистров CMPA и CMPB, предварительно размещаются в теневых регистрах. Это нужно для синхронной загрузки новых значений. Управление загрузкой регистров CMPA и CMPB осуществляется битами SHDWAMODE и SHDWBMODE, а также полями LOADAMODE и LOADBMODE регистра CMPCTL.

Блок компаратора формирует на выходах два события $CTR = CMPA$ и $CTR = CMPB$, возникающие в случае совпадения значения счетчика с регистром CMPA и/или регистром CMPB, соответственно.

Для каждого компаратора событие может возникать:

- один раз за период, если счетчик считает вверх или вниз;
- один раз за период, если счетчик считает вверх-вниз, но при этом значение в регистре CMPA/CMPB равно 0000h или значению TBPRD;
- два раза за период, если счетчик считает вверх-вниз и при этом значение в регистре CMPA/CMPB лежит в диапазоне 0001h – (TBPRD – 1).

На рисунках 14.4 – 14.7 приведены примеры формирования сигналов событий.

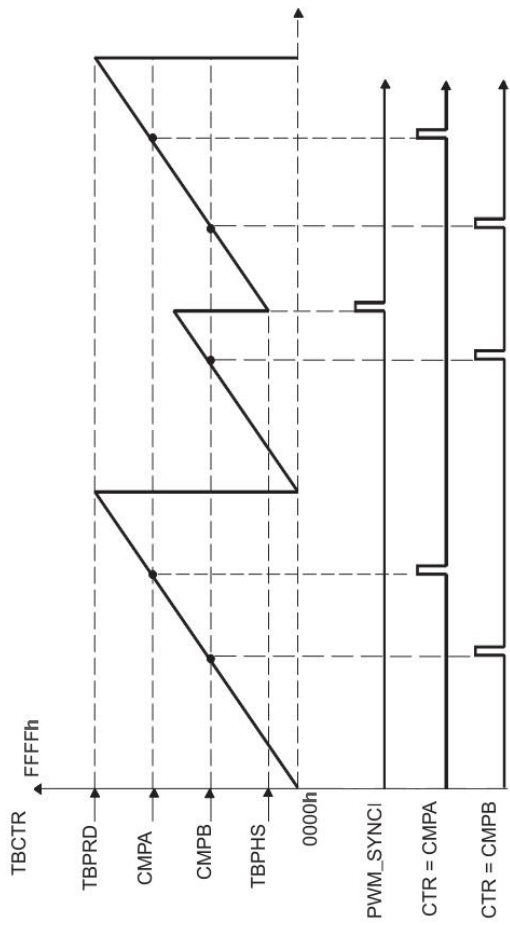


Рисунок 14.4 – Диаграмма работы при счете вверх

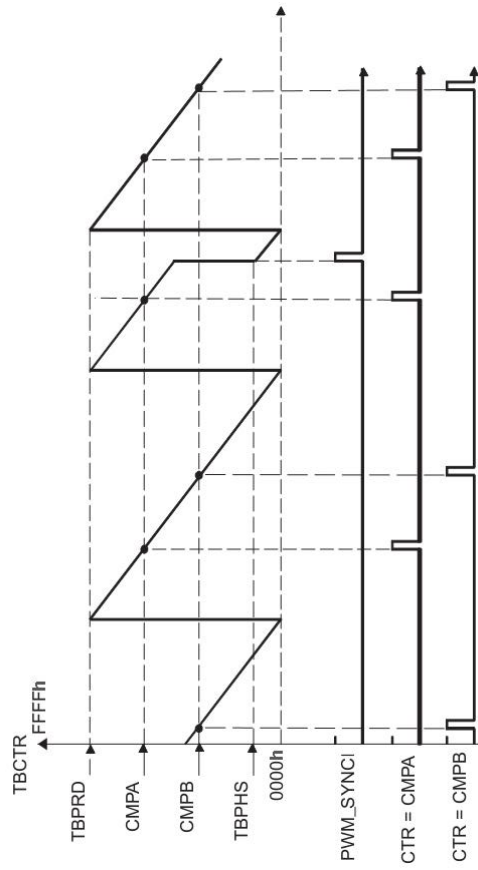


Рисунок 14.5 – Диаграмма работы при счете вниз

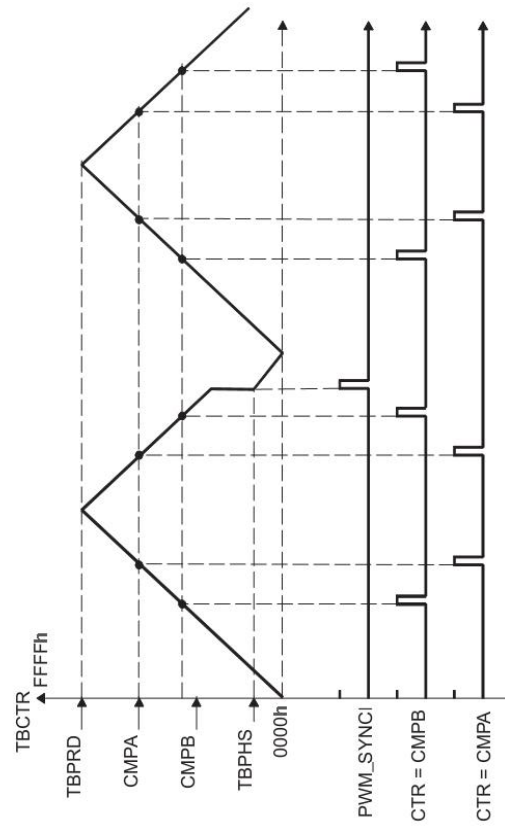


Рисунок 14.6 – Диаграмма работы при счете вверх-вниз. Синхронизация при счете вниз

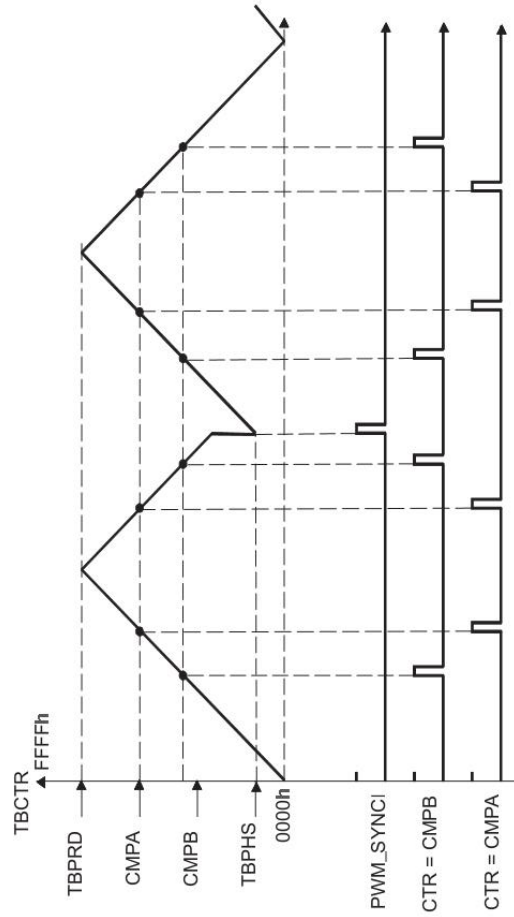


Рисунок 14.7 – Диаграмма работы при счете вверх-вниз. Синхронизация при счете вверх

14.3 Обработчик событий

Обработчик событий - блок, управляющий поведением сигналов на линиях PWMA и PWMB (см. рисунок 14.1) в зависимости от возникающих событий на входе блока и направления счета счетчика таймера. На поведение выходных сигналов влияют импульсы входных сигналов при возникновении событий: CTR = PRD, CTR = Zero, CTR = CMPA, CTR = CMPB.

Основные действия с сигналами PWMA и PWMB:

- переключение в единицу или ноль;
- инверсия (переключение в противоположное состояние);
- сохранение без изменений.

Поведение сигналов задается независимо друг от друга. Кроме этого, обработчик событий позволяет программно задавать состояние сигналов PWMA и PWMB и величину «мертвого времени» ШИМ. Управление работой блока производится посредством регистров AQCTLA, AQCTLB, AQSFRC, AQCSFRC. Существует вероятность того, что несколько событий могут произойти одновременно. Для таких ситуаций обработчик событий использует систему приоритетов событий (таблицы 14.1 – 14.3).

Таблица 14.1 – Распределение приоритетов событий при счете вверх

| Событие | Приоритет |
|---|---------------|
| Программное | 1 (наивысший) |
| CTR = TBPRD | 2 |
| CTR = CMPB (счет вверх) при счете вверх | 3 |
| CTR = CMPA (счет вверх) при счете вверх | 4 (низший) |

Таблица 14.2 – Распределение приоритетов событий при счете вниз

| Событие | Приоритет |
|---------------------------------------|---------------|
| Программное | 1 (наивысший) |
| CTR = Zero | 2 |
| CTR = CMPB (счет вниз) при счете вниз | 3 |
| CTR = CMPA (счет вниз) при счете вниз | 4 (низший) |

Таблица 14.3 – Распределение приоритетов событий при счете вверх-вниз

| Событие | Приоритет |
|--|---------------|
| Программное | 1 (наивысший) |
| CTR = CMPB (счет вверх) при счете вверх или CTR = CMPB (счет вниз) при счете вниз | 2 |
| CTR = CMPA (счет вверх) при счете вверх или CTR = CMPA (счет вниз) при счете вниз | 3 |
| CTR = Zero или CTL = PRD | 4 |
| CTR = CMPB (счет вверх) при счете вниз или CTR = CMPB (счет вниз) при счете вверх | 5 |
| CTR = CMPA (счет вверх) при счете вниз или CTR = CMPA (счет вниз) при счете вверх | 6 (низший) |

В режиме счета вверх:

- если компаратор запрограммирован так, что $CMPA/CMPB \leq TBPRD$ (счет вверх), то событие произойдет при CTR = CMPA/CMPB;
- если компаратор запрограммирован так, что $CMPA/CMPB > TBPRD$ (счет вверх), то событие не произойдет;
- если компаратор запрограммирован на срабатывание при счете вниз, то событие не произойдет.

В режиме счета вниз:

- если компаратор запрограммирован так, что $СМРА/СМРВ \leq ТВРД$ (счет вниз), то событие произойдет при $СТR = СМРА/СМРВ$;

- если компаратор запрограммирован так, что $СМРА/СМРВ \geq ТВРД$ (счет вниз), то событие произойдет при $СТR = ТВРД$;

- если компаратор запрограммирован на срабатывание при счете вверх, то событие не произойдет.

В режиме счета вверх-вниз:

- если счетчик считает вверх, а компаратор запрограммирован так, что $СМРА/СМРВ < ТВРД$ (счет вверх), то событие произойдет при $СТR = СМРА/СМРВ$;

- если $СМРА/СМРВ \geq ТВРД$ (счет вверх), то событие произойдет при $СТR = ТВРД$;

- если счетчик считает вниз, а компаратор запрограммирован так, что $СМРА/СМРВ < ТВРД$ (счет вниз), то событие произойдет при $СТR = СМРА/СМРВ$;

- если $СМРА/СМРВ \geq ТВРД$ (счет вверх), то событие произойдет при $СТR = ТВРД$.

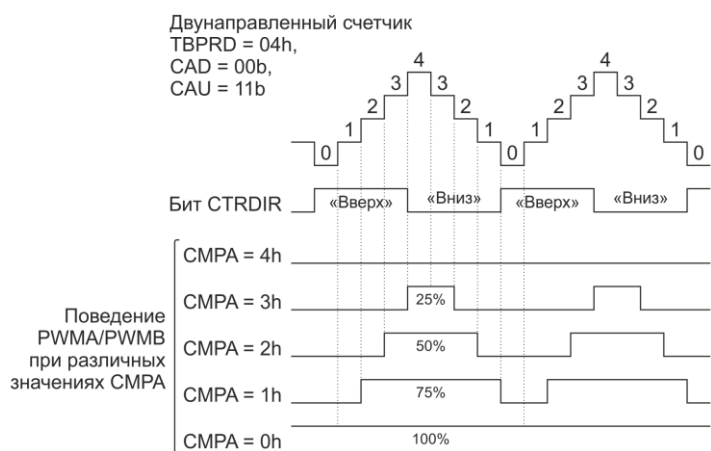


Рисунок 14.8 – Работа таймера при счете вверх-вниз с симметричным выходом (центрированная модуляция)

На рисунках 14.8 – 14.11 показано поведение линий PWMA и PWMB при различных видах модуляции. На рисунках приняты обозначения, пояснения к которым приведены в таблице 14.4.

Таблица 14.4 – Пояснения к обозначениям на рисунках 14.9 – 14.11

| Обозначение | | | Пояснение | |
|-------------|---------|----------------------|---|--|
| P × | CA × | CB × | События $СТR = PRD$, $СТR = СМРА$, $СТR = СТРВ$, соответственно. Символ «×» указывает на то, что при возникновении этого события сигнал на линии PWMA/PWMB остается без изменений. Пунктирными линиями отмечены моменты возникновения события. Так, например (см. рисунок 14.9), при возникновении события $СТR = СТРВ$ сигнал на линии PWMA остается без изменения, а сигнал на линии PWMB переключается в ноль | |
| Z ↑ | Z ↓ | Событие $СТR = Zero$ | | |
| CA ↑ | CA ↓ | Событие $СТR = СМРА$ | | |
| CB ↑ | CB ↓ | Событие $СТR = СМРВ$ | | |
| | | | Символ «↑»/«↓» указывает на то, что при возникновении этого события сигнал на линии PWMA/PWMB переключается в единицу/ноль | |

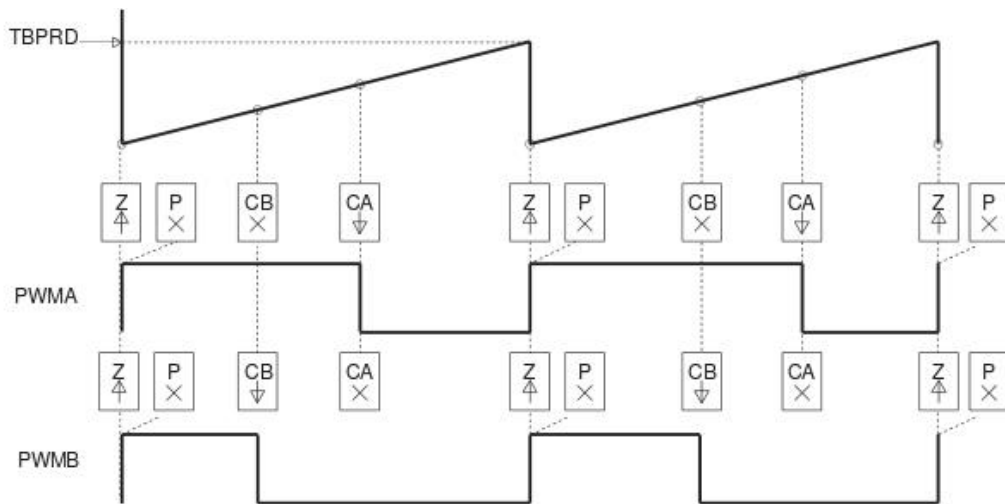


Рисунок 14.9 – Независимый режим работы выходов (фронтная модуляция)

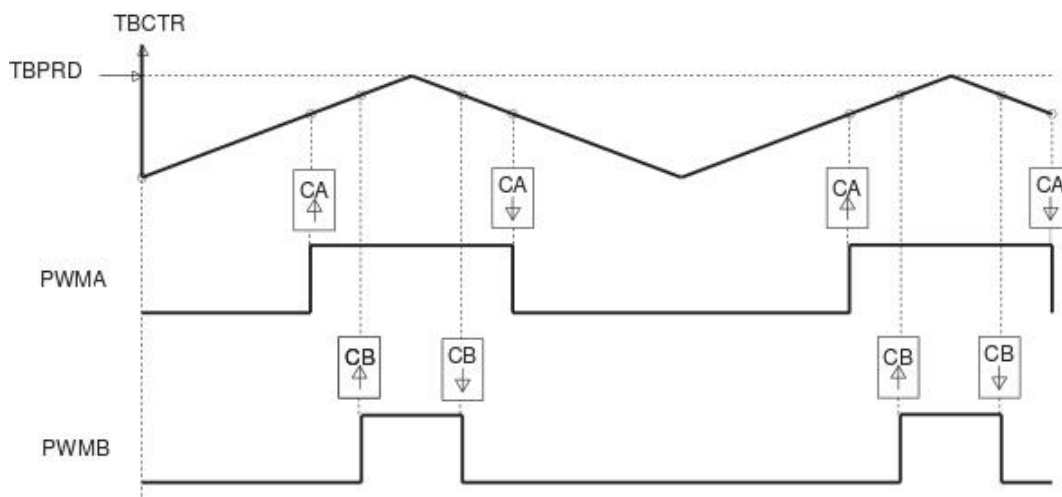


Рисунок 14.10 – Симметричный режим работы при счете вверх-вниз (центрированная модуляция)

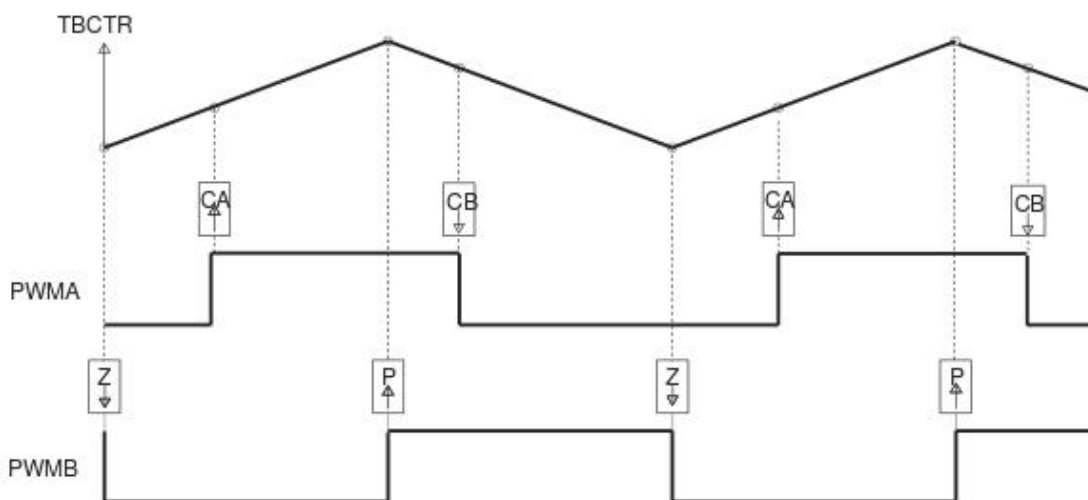


Рисунок 14.11 – Ассиметричный режим работы при счете вверх-вниз

14.4 Пороговый выключатель

Пороговый выключатель контролирует выходные сигналы PWMA и PWMB обработчика событий и позволяет удерживать их в определенном заданном пользователем состоянии в случае прихода сигнала триггера от цифровых компараторов блока АЦП. Этот блок удобен для организации релейного поддержания заданного уровня какой-либо физической величины, например для организации контура тока. В этом случае цифровой компаратор, к каналу АЦП которого подключен сигнал датчика тока контура, формирует сигнал о превышении током задания, а соответствующий пороговый выключатель реагирует на это превышение и включает/отключает соответствующий силовой транзистор посредством влияния на выход ШИМ.

Функциональные возможности:

- входные события от компараторов блока АЦП могут использоваться всеми блоками ШИМ;
- при регистрации события от компаратора блока АЦП, выходные сигналы обработчика событий могут быть переведены в состояние логической единицы, нуля или оставлены без изменений;
- поддерживаются однократное и циклическое срабатывания для удержания выхода;
- входное событие от компаратора блока АЦП может анализироваться в однократном и циклическом режимах;
- событие срабатывания компаратора блока АЦП может быть сгенерировано программно;
- пороговый выключатель может быть отключен, если он не требуется.

Управление пороговым выключателем осуществляется посредством регистров HDSEL, HDCTL и HDFRC.

Функционирование

Когда выходные сигналы компараторов блока АЦП переходят в состояние высокого уровня, формируется событие. Каждый пороговый выключатель блока ШИМ может использовать, а может не использовать эти события в своей работе; выбор, по сигналу какого компаратора блока АЦП формировать событие удержания, задается с помощью регистра HDSEL. Длительность импульса на входном сигнале от компаратора блока АЦП не должна быть меньше периода системного синхросигнала. Каждый входной сигнал компаратора блока АЦП должен быть настроен на однократное или циклическое формирование события, выбор режима задается битами CBC и OSHT, а источник события полем DCMR.

При получении события от компаратора блока АЦП в режиме циклической обработки немедленно формируется реакция на основе содержимого регистра HDCTL, в результате чего меняется состояние сигналов на выходе порогового выключателя взамен полученных от обработчика событий PWMA и/или PWMB на заданное пользователем в регистре HDCTL. Дополнительно устанавливается флаг CBC в регистре HDFLG, и генерируется прерывание PWM_HDINT. Удержание выходных сигналов PWMA и PWMB заканчивается по событию TVCTR = 0000h, при условии, что событие компаратора блока АЦП уже не активно. Таким образом, в режиме циклической обработки состояние удержания сбрасывается в каждом периоде ШИМ. При этом флаг CBC остается активным до его программного сброса. Если после сброса флага CBC вновь будет получено событие компаратора блока АЦП, то флаг установится вновь.

При получении события компаратора блока АЦП в режиме однократной обработки, также немедленно формируется реакция на основе содержимого регистра HDCTL, которая меняет состояние выходных сигналов PWMA и/или PWMB. В дополнение, устанавливается флаг OST, и генерируется прерывание PWM_HDINT. Удержание

выходных сигналов будет производиться до программного сброса записью единицы в бит OST регистра HDCLR.

Способ удержания выходных сигналов при получении события компаратора блока АЦП программируется индивидуально для выходных сигналов PWMA и PWMB в регистр HDCTL.

14.5 Генератор задержки ШИМ

Блок имеет на входе сигналы ШИМ А и В с выходов обработчика событий, а на выходах повторяет эти сигналы, но со вставкой задержки («мертвое время») в момент переключения сигналов (если это необходимо).

Задержку можно учесть при программировании обработчика событий, но чтобы с высокой вероятностью избежать ошибок, желательно использовать генератор задержки ШИМ.

Основные функции генератора:

- генерация пары выходных сигналов (PWMA и PWMB), с выдержкой интервалов (задержек) времени относительно входных сигналов PWMA, PWMB;
- программирование задержки для активного высокого и активного низкого уровня сигналов каналов PWMA и PWMB;
- добавление программируемой задержки передних фронтов сигналов;
- добавление программируемой задержки для задних фронтов сигналов;
- возможность передачи сигналов с входов на выходы без изменений.

Генератор задержки ШИМ программируется посредством регистров DBCTL, DBRED и DBFED. Структурная схема генератора «мертвого времени» ШИМ представлена на рисунке 14.12.

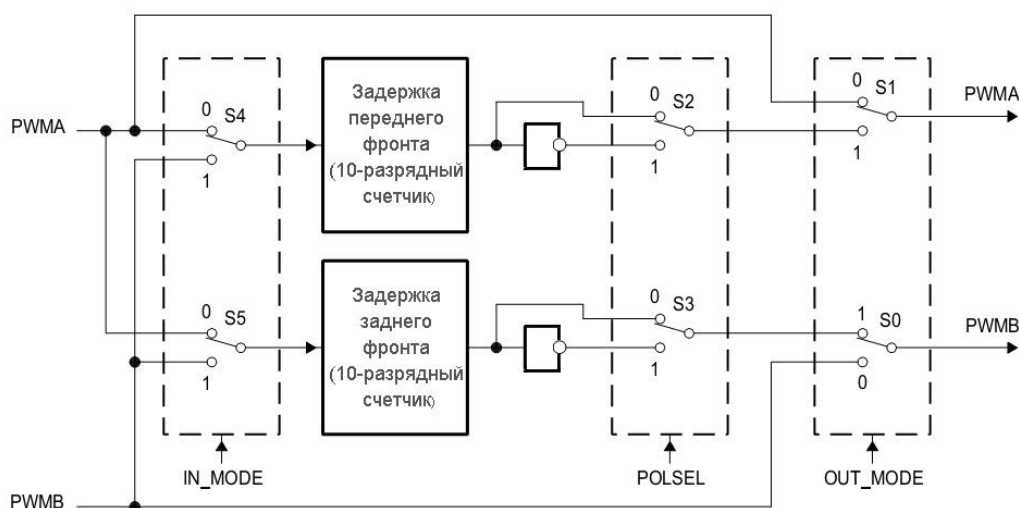


Рисунок 14.12 – Структурная схема генератора «мертвого времени» ШИМ

Функционирование

Генератор задержки ШИМ может работать с четырьмя источниками (фронты сигналов PWMA и PWMB). Выбор источника задается полем MODE регистра DBCTL.

Поле POLSEL позволяет задать инверсию (переключение значения на противоположное) сигнала после внесения задержки (см. рисунок 14.13).

Величины задержек по переднему и заднему фронту программируются отдельно посредством регистров DBRED и DBFED, соответственно.

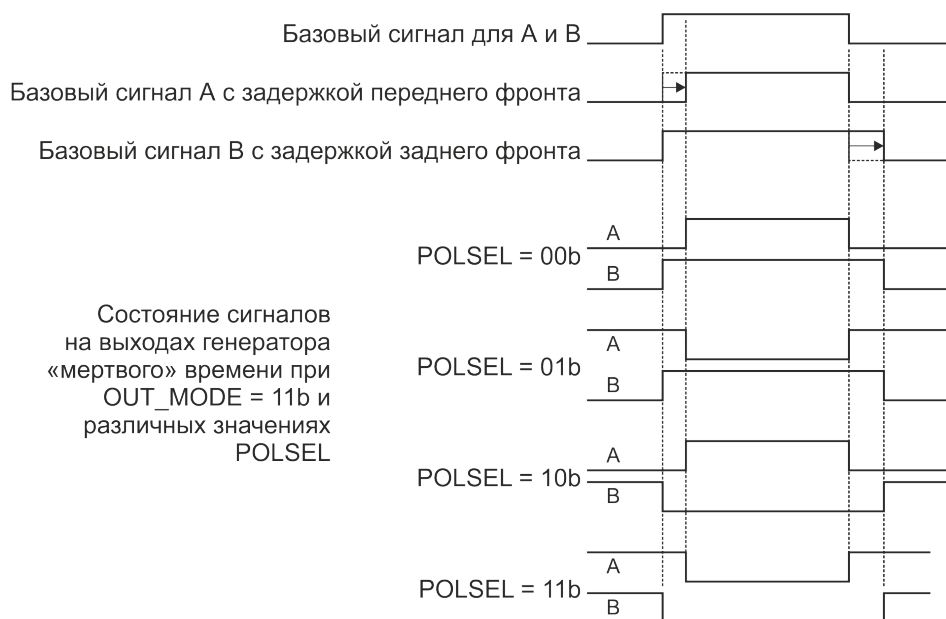


Рисунок 14.13 – Временные диаграммы работы генератора «мертвого времени» в типовой конфигурации

14.6 Фильтр коротких импульсов

Фильтр коротких импульсов предназначен для блокирования прохождения на выход импульсов с длительностью меньше заданной. Этот блок может применяться, если драйвер силового ключа инвертора не имеет такой функции, а для обеспечения правильного режима работы транзистора необходимо запретить открытие/закрытие транзистора на очень короткие времена.

Основные функции фильтра:

- программируемая ширина минимального пропускаемого импульса;
- фильтр может быть отключен, если он не требуется.

Ширина минимального импульса, допускаемого к прохождению на выход, задается в регистре FWDTH в тактах PCLK и может принимать значение от 00h (фильтр выключен) до FFh. Импульсы длительностью меньше заданной пропускаться не будут.

14.7 Модулятор

Блок позволяет модулировать выходной ШИМ сигнал с помощью высокочастотных импульсов программируемой скважности. Модулирование требуется для управления силовыми ключами через импульсный трансформатор.

Основные функции модулятора:

- программируемая частота;
- программируемая ширина первого импульса;
- программируемая скважность второго и последующего импульсов;
- модулятор может быть отключен (бит CHPEN регистра PCCTL).

Модулятор программируется посредством регистра PCCTL. Структурная схема модулятора приведена на рисунке 14.14.

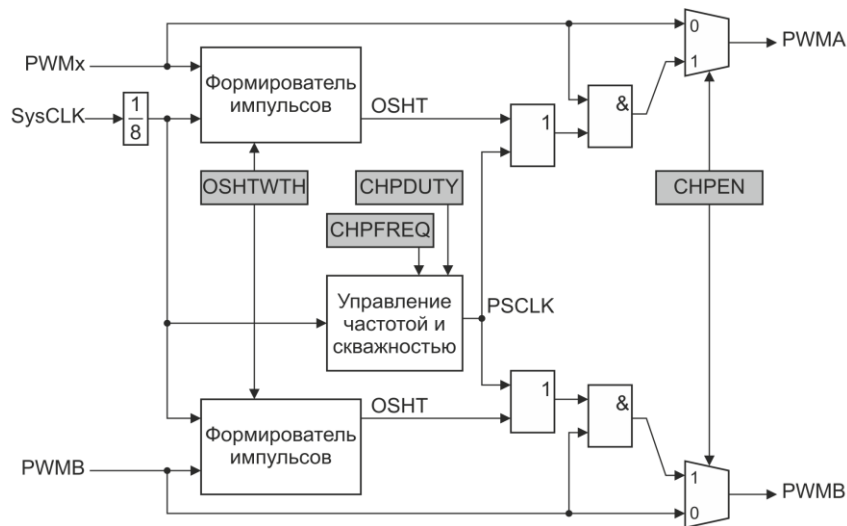


Рисунок 14.14 – Структурная схема модулятора

Функционирование

Ширина первого импульса программируется независимо полем OSHTWTH, это требуется для открывания ключа. Для остальных импульсов частота модуляции формируется на основе системной частоты при помощи делителя, программируемого полем SHPFREQ. Скважность импульсов программируется полем CHPDUTY.

Значения поля OSHTWTH лежат в диапазоне 0h – Fh.

Ширина L первого импульса определяется по формуле

$$L = T \times 8 \times (\text{OSHTWTH} + 1), \quad (14.1)$$

где T – период синхросигнала SysCLK.

Значения поля CHPDUTY лежат в диапазоне 0h – 7h.

Скважность D (с шагом 12,5 %) последующих импульсов определяется по формуле

$$D = 12,5 \times (\text{CHPDUTY} + 1). \quad (14.2)$$

На рисунке 14.15 приведен пример временных диаграмм работы модулятора.

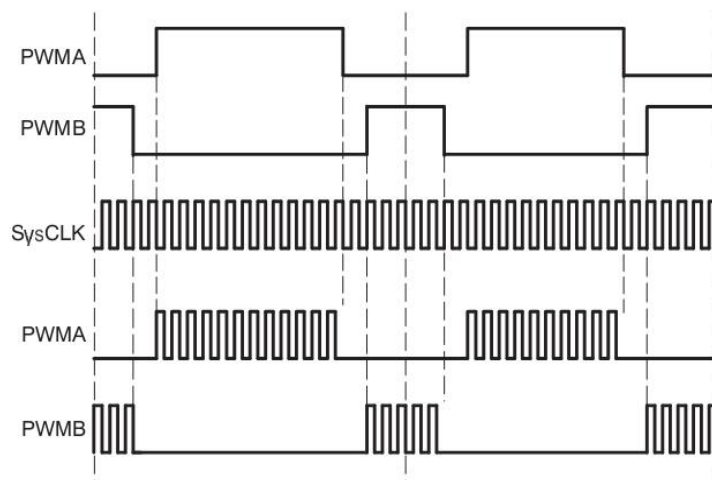


Рисунок 14.15 – Временные диаграммы работы модулятора

14.8 Детектор сигнала аварии

Блок контролирует выходы PWMA и PWMB и может переводить их в определенное (запрограммированное) состояние в случае, если поступит сигнал аварии.

Основные функции:

- входные сигналы аварии с выводов микроконтроллера PWM_TZ0 – PWM_TZ5 могут использоваться любым блоком ШИМ;
- в случае если поступит сигнал аварии, выходы ШИМ могут быть переведены в одно из состояний: логического нуля, логической единицы, высокоимпедансное или оставлены без изменения;
- поддерживается однократная блокировка выводов для ситуации короткого замыкания или перегрузки по току;
- поддерживается циклическая блокировка для режима ограничения тока;
- каждый входной источник сигнала аварии может быть обработан в однократном и циклическом режимах;
- поддерживается программная генерация сигнала аварии;
- детектор сигнала аварии может быть отключен, если он не требуется.

Детектор сигнала аварии программируется посредством регистров TZSEL, TZCTL, TZEINT, TZFLG, TZCLR и TZFRC.

Функционирование

Структурная схема детектора сигналов аварии показана на рисунке 14.16.

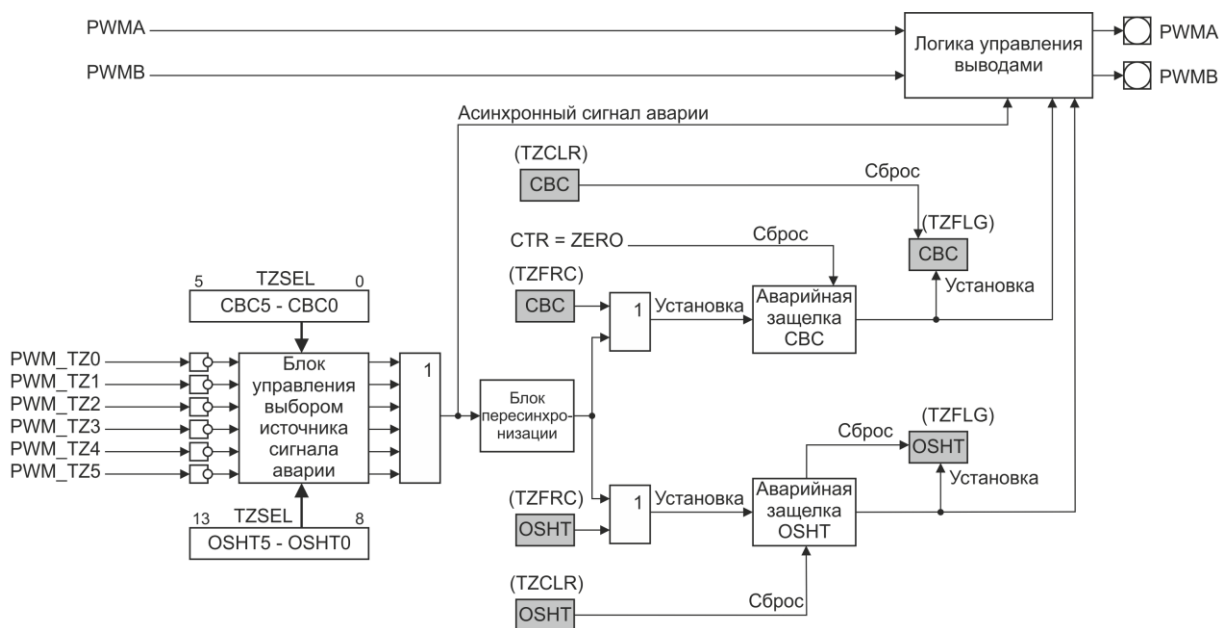


Рисунок 14.16 – Структурная схема детектора сигналов аварии

Переход входных сигналов аварии PWM_TZn из состояния логической единицы в состояние логического нуля формирует событие аварии. Каждый блок ШИМ может использовать или не использовать эти события в своей работе (программируется посредством регистра TZSEL). События могут формироваться синхронно (с цифровым фильтром помех) или асинхронно (программируется через регистры GPIO микроконтроллера). При синхронной обработке, длительность импульса на входном сигнале сбоя должна быть не меньше периода синхросигнала TVCLK. Если же обработка производится в асинхронном режиме, то событие формируется и обрабатывается даже в

том случае, если по какой-либо причине отключилось тактирование микроконтроллера. Каждый входной сигнал аварии должен быть настроен на однократное или циклическое формирование события аварии (программируется посредством регистра TZSEL).

При получении события аварии в режиме циклической обработки немедленно выполняется действие, заданное регистром TZCTL, и устанавливается флаг CBC в регистре TZFLG, а также генерируется прерывание PWM_TZINT (если разрешено в регистре TZEINT и контроллером прерываний). Аварийное удержание выводов заканчивается по событию ТВCTR = 0000h, при условии, что событие аварии уже неактивно. Таким образом, в режиме циклической обработки событие аварии сбрасывается в каждом периоде ШИМ, хотя флаг аварии CBC остается установленным до принудительного программного сброса. Если после сброса регистра флага CBC вновь будет получено событие аварии, то флаг установится вновь. На рисунке 14.17 показана схема формирования прерывания.

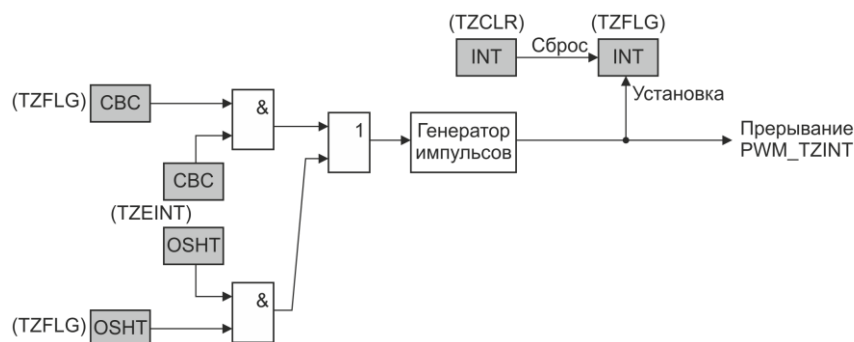


Рисунок 14.17 – Схема формирования прерывания

При получении события аварии в режиме однократной обработки, немедленно выполняется действие, заданное регистром TZCTL, и устанавливается флаг OST в регистре TZFLG, а также генерируется прерывание PWM_TZINT (если разрешено в регистре TZEINT и контроллером прерываний). Аварийное удержание выводов заканчивается после принудительного программного сброса записью в бит OST регистра TZCLR.

Аварийное состояние выводов при получении события сбоя программируется индивидуально для выхода PWMA и PWMB полями TZA и TZB регистра TZCTL.

14.9 Триггер событий

Основные функции:

- получение событий, сформированных таймером и компаратором;
- использование информации о направлении счета (вверх/вниз);
- использование делителя событий, для формирования сигнала прерывания и команды на запуск секвенсоров блока АЦП, со следующими состояниями:
 - делитель выключен (срабатывание на каждое событие);
 - срабатывание на каждое второе событие;
 - срабатывание на каждое третье событие;
- предоставление доступа процессора к содержимому регистра флагов событий и счетчикам событий;
- возможность программного выставления прерывания и сигнала запуска секвенсоров блока АЦП.

Функционирование

Триггер событий (не показан на рисунке 14.1) программируется посредством регистров ETSEL, ETPS, ETFLG, ETCLR и ETFRC. Функциональная схема триггера событий показана на рисунке 14.18.

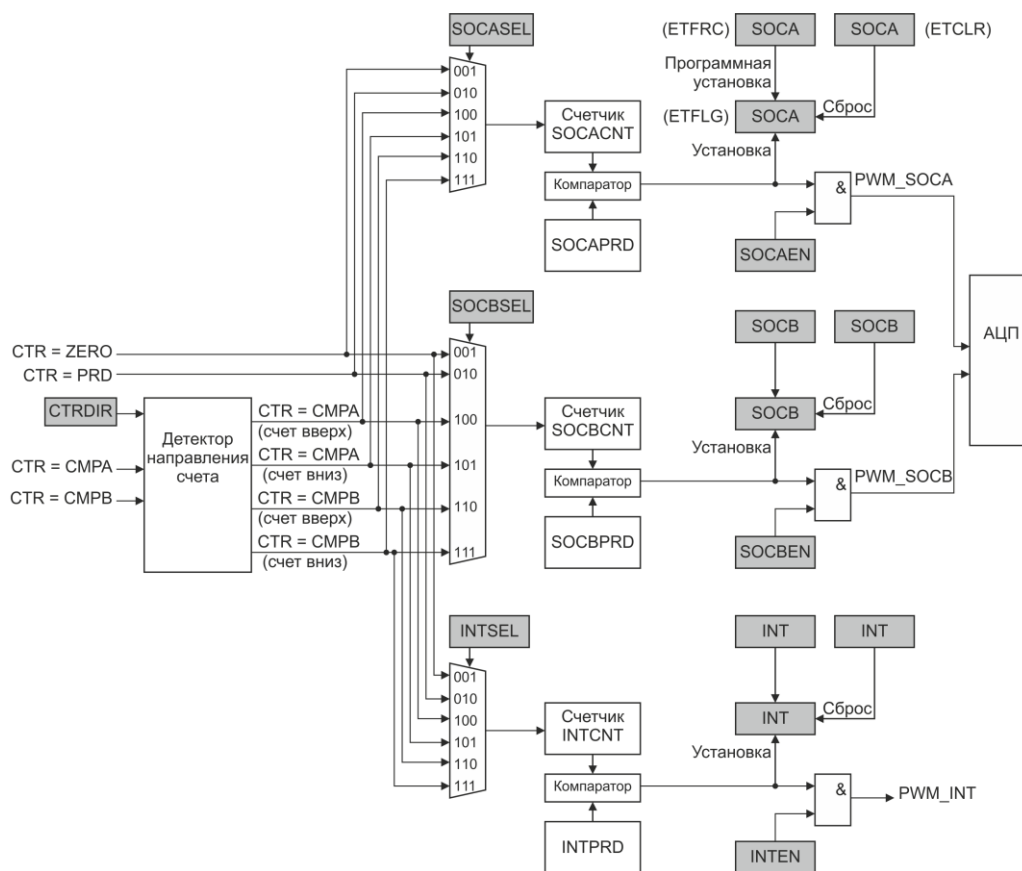


Рисунок 14.18 – Структурная схема триггера событий

Триггер может генерировать прерывания (если разрешено битом INTEN регистра ETSEL) по каждому первому, второму или третьему событию (поле INTPRD), которое задается полем INTSEL. Количество возникших событий отражается в поле INTCNT. Счетчик INTCNT считает от 00b до INTPRD и сбрасывается только вместе с отправкой активного прерывания.

Когда возникает совпадение INTCNT и INTPRD, то возможны варианты:

- если прерывание разрешено и сброшен флаг INT (регистр ETFLG), то генерируется прерывание и устанавливается флаг INT, а счетчик INTCNT сбрасывается в 00b и начинает считать заново;
- если прерывание запрещено или флаг INT установлен, то счетчик перестает считать события;
- если прерывание разрешено, но флаг от предыдущего прерывания еще не сброшен, то счетчик хранит свое максимально достигнутое значение ($INTCNT = INTPRD$) до сброса флага INT. Это позволяет обработать еще прерывание, пришедшее за то время, пока обрабатывалось предыдущее.

Каждая запись в INTPRD сбрасывает счетчик INTCNT. Запись единицы в бит INT регистра ETFRC увеличит значение счетчика на единицу.

14.10 Блок ШИМ высокого разрешения

Микроконтроллер содержит шесть блоков ШИМ высокого разрешения PWM0 – PWM5. Принцип работы аналогичен описанному выше блоку ШИМ.

Функция высокого разрешения блока ШИМ является дополнительной и имеет особенности:

- улучшенный контроль скважности выходного сигнала ШИМ канала А (сигнал PWMA);
- улучшенная точность переключения фронтов, с использованием увеличенной разрядности регистров CMPA и TVPHS (подключаются поля CMPAHR и TVPHSHR);
- логика калибровки для программной коррекции линии задержки.

Дополнительный регистр управления расширенными возможностями – HRCNFG.

Функциональная схема блока ШИМ с функцией высокого разрешения показана на рисунке 14.19.

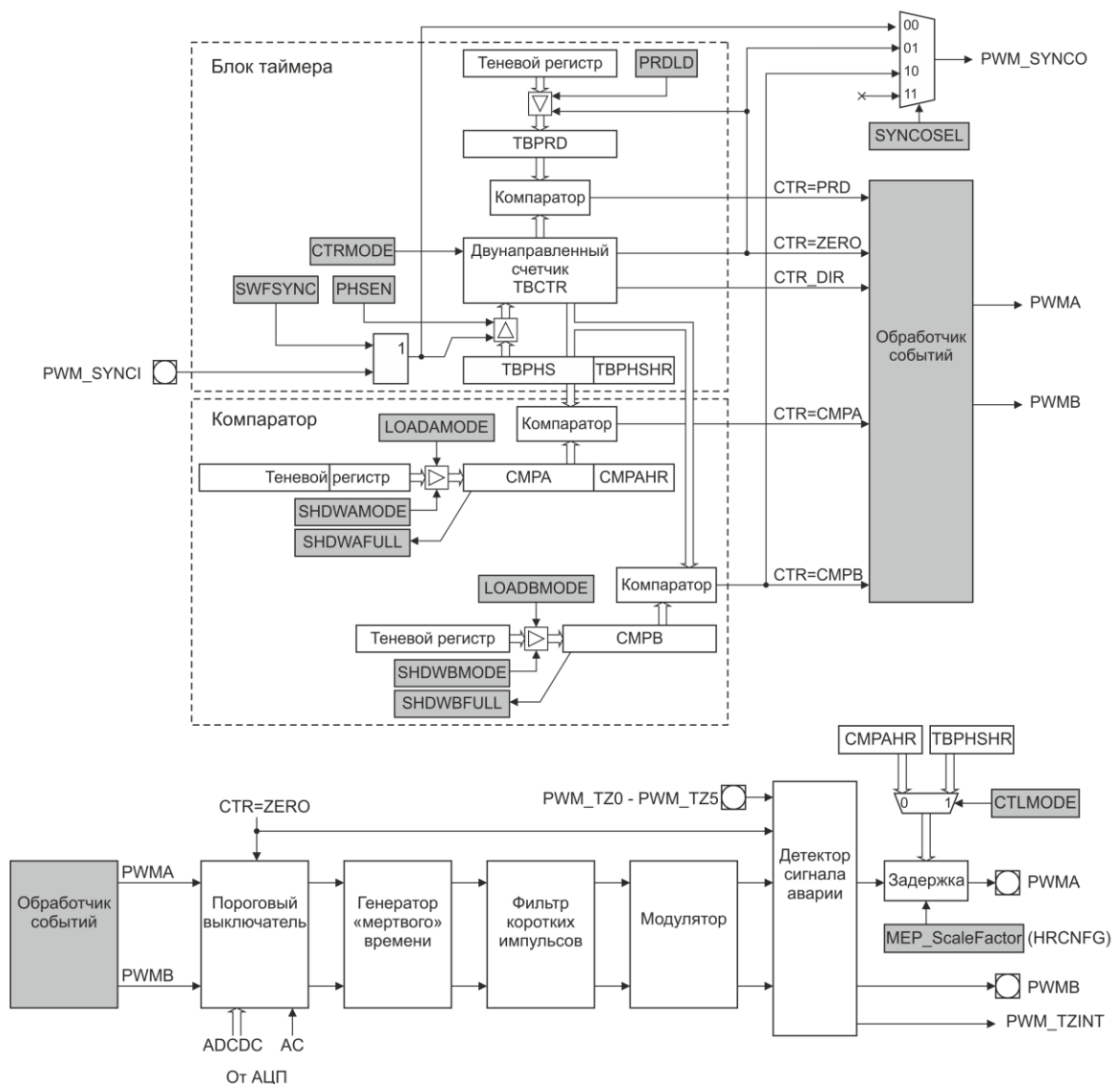


Рисунок 14.19 – Схема блока ШИМ высокого разрешения

Улучшение разрешающей способности

В блоке ШИМ высокого разрешения используется специальная линия задержки с шириной отсчета 150 пс, что увеличивает разрешающую способность. Формула

вычисления разрешающей способности по ширине единичного отсчета и периоду следования импульсов ШИМ представлена на рисунке 14.20.

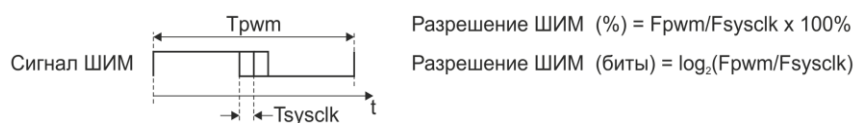


Рисунок 14.20 – Вычисление разрешающей способности блока ШИМ

Конфигурация

Кроме обычных функций блок ШИМ высокого разрешения реализует дополнительные два режима работы.

Режим фронтов – для высокоточного контроля выставления переднего и заднего фронтов сигнала на выходе PWMA подключается линия задержки. Управление фронтами по отдельности используется в схемах, где требуется прецизионная точность скважности формируемых импульсов. Управление обоими фронтами одновременно используется в схемах, где требуется высокая точность в выставлении фазы формируемого сигнала (к примеру, в мостах).

Режим управления – линия задержки управляется либо полем SMPAHR (управление скважностью), либо полем TVPNSHR (управление фазой). Отдельно передний или задний фронты выставляются полем SMPA, а вместе – полем TVPNSHR.

Примечание – Поле SMPAHR может использовать отложенную загрузку (теневой регистр), полностью аналогично тому, как это реализовано для регистра SMPA.

Функционирование

На рисунке 14.21 приведен пример работы блока ШИМ высокого разрешения.

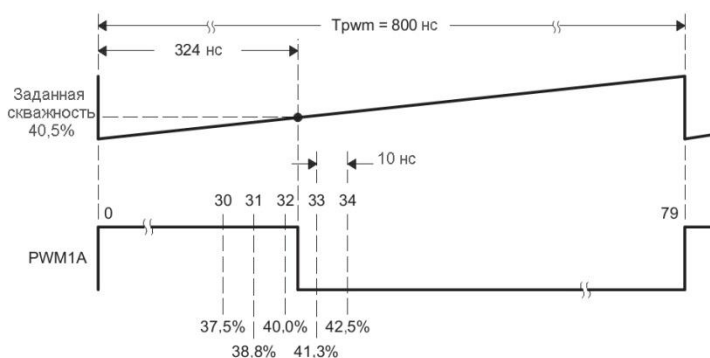


Рисунок 14.21 – Пример работы блока ШИМ высокого разрешения

За основу принято, что длительность отсчета (задержка одного элемента) равна 180 пс, а рабочая частота 100 МГц (период 10 нс). Формирование скважности 40,5 % от периода $T_{pwm} = 800$ нс складывается из 32 отсчетов (регистр SMPA) счетчика, работающего на системной частоте, формируя задержку 320 нс, и линии задержки, запрограммированной на 22 отсчета (регистр SMPAHR), формируя дополнительную задержку 3,96 нс. Общая задержка получается 323,96 нс, что составляет почти нулевую погрешность от значения 324 нс (40,5 %).

Ограничения диапазона скважности

1 Линия задержки не может работать в течение всего периода формируемого сигнала ШИМ, поэтому вводится ограничение на минимальную длительность импульса:

- три периода системного такта, если схема калибровки неактивна;
- шесть периодов системного такта, если схема калибровки активна.

В случае если в соответствии с требуемой скважностью длительность импульса выходного сигнала ШИМ оказалась короче минимальной, то фронты будут выставлены с точностью, как и в обычном блоке ШИМ.

При работе на низких частотах ограничение по управлению скважностью сигнала ШИМ практически незаметно.

2 Сумарная задержка, генерируемая блоком ШИМ высокого разрешения, не может превышать двух периодов системной частоты (сигнал SysCLK) плюс 1 нс.

Калибровка

Линия задержки состоит из 255 одинаковых, последовательно включенных элементов. Длительность отсчета (задержки одного элемента) составляет 150 пс и определяется технологическими нормами, в соответствии с которыми выполнен микроконтроллер. Длительность отсчета также может незначительно меняться, в зависимости от текущего значения питания микроконтроллера и текущей температуры.

14.11 Сопряжение с блоком АЦП

Выходы девяти блоков ШИМ (PWM_SOCA и PWM_SOCB каждого блока) объединяются по ИЛИ, как показано на рисунке 14.22. Сигналы с выходов элементов ИЛИ защелкиваются в шести триггерах и формируют импульсы запуска секвенсоров блока АЦП.

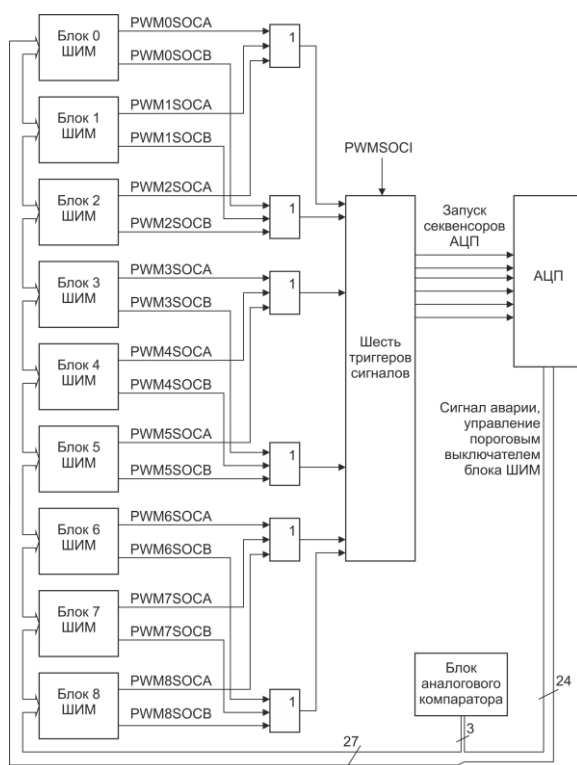


Рисунок 14.22 – Схема формирования сигналов запуска секвенсоров блока АЦП

15 Приемопередатчики UART

В состав микроконтроллера входят четыре идентичных универсальных асинхронных приемопередатчика UART0, UART1, UART2, UART3.

В состав приемопередатчика входят два буфера типа FIFO. Буфер приемника имеет разрядность 12, буфер передатчика – разрядность восемь. Каждый буфер может хранить до 32 байт данных, и каждый буфер может быть сконфигурирован (программно) как 32-байтный или как однобайтный.

Приемопередатчик обеспечивает:

- независимое маскирование прерываний от буфера передатчика, буфера приемника, по таймауту приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки;

- возможность деления тактовой частоты в диапазоне от 1 до 65 535 (допускается использование нецелых коэффициентов деления, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц);

- возможность организации аппаратного управления потоком данных;

- поддержку прямого доступа к памяти;

- поддержку функции управления модемом (линии CTS, DCD, DSR, RTS, DTR и RI).

Приемопередатчик реализует:

- передачу данных длиной от 5 до 8 бит со скоростью до 921 600 бит/с;

- контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение либо не передается);

- формирование одного или двух стоповых бит;

- обнаружение ложных стартовых битов;

- формирование и обнаружение сигнала разрыва линии.

Функциональные возможности

Режим работы приемопередатчика и скорость обмена данными контролируются регистром LCR_N и регистрами делителя IBRD и FBRD.

Устройство может формировать следующие сигналы:

- независимые маскируемые прерывания от приемника (в том числе по таймауту), передатчика, а также по изменению состояния модема и в случае обнаружения ошибки;

- общее прерывание, возникающее в случае, если возникло одно из независимых немаскированных прерываний;

- сигналы запроса на прямой доступ к памяти для совместной работы с контроллером DMA.

В случае возникновения ошибки в структуре сигнала, четности данных, а также разрыва линии соответствующий бит ошибки устанавливается и сохраняется в буфере приемника. В случае переполнения буфера приемника также устанавливается соответствующий бит, а буфер становится недоступным для записи.

Приемопередатчик поддерживает режим модема.

15.1 Функционирование блока UART

На рисунке 15.1 показана упрощенная функциональная схема приемопередатчика.

Генератор тактового сигнала приемопередатчика формирует синхросигнал последовательного обмена данными, который представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UART_CLK, и частотой в 16 раз превышающей частоту передачи данных.

Буфер передатчика предназначен для хранения данных (полученных от ЦП) до тех пор, пока они не будут переданы внешнему устройству.

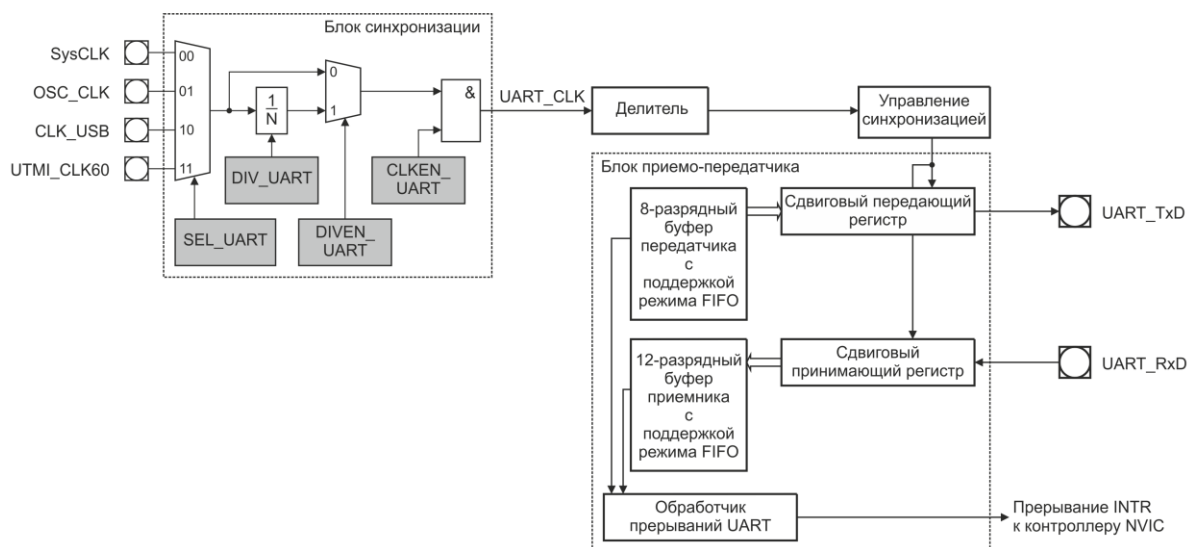


Рисунок 15.1 – Функциональная схема приемопередатчика

Буфер приемника предназначен для хранения данных и кодов ошибки (принятых от внешнего устройства) до тех пор, пока они не будут прочитаны ЦП.

Обработчик прерываний генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения независимых прерываний по схеме ИЛИ. Сигнал прерывания передается на контроллер NVIC.

Сброс модуля

Модуль UART может быть сброшен только сигналом сброса микроконтроллера.

Приемопередающая логика модуля может быть также сброшена соответствующим битом поля UARTRST регистра PER_RST0, при этом управляющие регистры и FIFO не сбрасываются.

Синхронизация

Существует ограничение на соотношение между тактовыми частотами SysCLK и UART_CLK:

$$\text{UART_CLK} \leq 5/3 \times \text{SysCLK}$$

Например, для достижения максимальной скорости передачи данных 921 600 бод (при $F_{\text{uart_clk}} = 921\,600 \times 16 = 14,7456$ МГц) частота SysCLK должна быть не менее 8,84736 МГц.

Для точной настройки частоты передачи данных используются два делителя (см. рисунок 15.1). Коэффициент деления первого задается полем DIV_UART регистра UART_CLK. Коэффициент деления второго делителя имеет целую и дробную части, которые задаются регистрами IBRD и FBRD. Возможность задания нецелых коэффициентов деления позволяет осуществлять обмен данными на стандартных скоростях, используя в качестве источника тактовый сигнал с произвольной частотой более 3,6864 МГц.

Коэффициент деления K частоты сигнала UART_CLK рассчитывается по формуле

$$K = F_{\text{uart_clk}} / (16 \times \text{baud_rate}), \quad (15.1)$$

где $F_{\text{uart_clk}}$ – частота сигнала синхронизации блока UART, Гц;
 baud_rate – скорость передачи, бод.

Получившееся дробное десятичное число следует разделить на две части – целую и дробную.

Целая часть после преобразования в двоичный формат записывается в регистр IBRD.

Дробная часть умножается на 64 и округляется до ближайшего целого числа. Полученное число преобразовывается в двоичный формат и записывается в регистр FBRD.

Для примера, пусть требуемая скорость передачи данных 230 400 бит/с и частота тактового сигнала UART_CLK равна 4 МГц. Тогда:

$$K = (4 \times 10^6) / (16 \times 230\,400) = 1,085.$$

Получившееся число разбивается на две части – 1 и 0,085.

В регистр IBRD записывается значение 0001h.

Значение $(0,085 \times 64)$ округляется и преобразовывается в 05h для записи в регистр FBRD.

Таким образом, реальные значения коэффициента деления частоты и скорости передачи будут следующими:

$$K = 1 + 5/64 = 1,078,$$

$$\text{baud_rate} = (4 \times 10^6) / (16 \times 1,078) = 231\,911 \text{ бит/с.}$$

Ошибка установки скорости:

$$\Delta = ((231\,911 - 230\,400) / 230\,400) \times 100 \% = 0,656 \%$$

Максимальная ошибка установки скорости передачи данных:

$$\Delta = (1/64) \times 100 \% = 1,560 \%$$

Такая ошибка возникает в случае $K = 1$, при этом разница накапливается в течение 64 тактовых интервалов.

Содержимое регистров LCR_H, IBRD и FBRD обновляется при записи в регистр LCR_H. Таким образом, для того, чтобы новые параметры коэффициента деления вступили в силу, после их записи в регистры IBRD и FBRD, необходимо осуществить запись в регистр LCR_H и только в такой последовательности.

Примечание – Изменение содержимого регистров IBRD, FBRD и LCR_H допускается только во время, когда приемопередатчик запрещен и не осуществляется передача/прием байта.

Передача и прием данных

Данные для передачи заносятся в буфер передатчика посредством записи в регистр DR. После записи хотя бы одного байта в буфер передатчика устанавливается флаг BUSY в регистре FR. Это состояние флага сохраняется, пока буфер передатчика не пуст (даже если работа приемопередатчика запрещена). Далее если работа приемопередатчика разрешена (установлены биты UARTEN и TXE регистра CR), начинается передача информационного кадра с параметрами, указанными в регистре управления линией LCR_H. Передача данных продолжается до опустошения буфера передатчика (до окончания передачи всех байт). По окончании передачи сбрасывается флаг BUSY.

При приеме байта данных (установлены биты UARTEN и RXE регистра CR) для каждого бита производится три выборки уровня, и решение о значении бита принимается по мажоритарному принципу.

В случае если приемник находился в неактивном состоянии (постоянный высокий уровень сигнала на линии UART_RxD), и произошла смена уровня входного сигнала с высокого на низкий (стартовый бит), включается счетчик, тактируемый внутренним сигналом, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов.

Стартовый бит считается достоверным в случае, если сигнал на линии UART_RxD сохраняет низкий логический уровень в течение восьми периодов внутреннего синхросигнала с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

После обнаружения достоверного стартового бита очередной бит данных фиксируется каждые 16 отсчетов тактового сигнала. Производится регистрация всех бит данных (согласно запрограммированным параметрам) и бита четности (если включен режим контроля четности).

По окончании приема байта производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UART_RxD). После чего байт заносится в буфер приемника вместе с тремя битами признаков ошибки (см. рисунок 15.2) и битом переполнения буфера.

В 12-разрядной ячейке буфера байт данных располагается в области младших восьми бит, три бита признаков ошибки – в битах с 8 по 10.

Флаг переполнения буфера приемника выставляется в том случае, если к моменту, когда очередной кадр данных полностью принят, буфер уже заполнен. В этом случае принятый кадр остается в сдвиговом принимающем регистре и, в случае приема следующего кадра данных, будет потерян.

Как только в буфере приемника освобождается место для записи, кадр данных, находящийся в сдвиговом регистре, переписывается в буфер, а флаг переполнения сбрасывается.

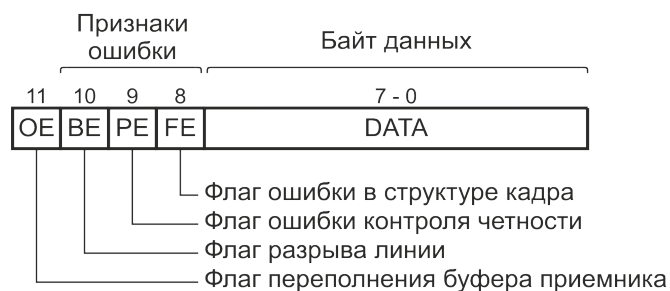


Рисунок 15.2 – 12-разрядная ячейка принимающего буфера

Данные из буфера приемника можно прочитать посредством регистра DR. Состояние признаков ошибки и флага переполнения определяется чтением регистра RSR_ECR и относится к последнему байту, считанному из регистра DR, в связи с этим регистр DR всегда должен считываться первым.

Все флаги сбрасываются одновременно записью любого значения в регистр RSR_ECR или после сброса устройства.

Примечания

1 Необходимо запрещать работу приемопередатчика перед перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние во время передачи или приема, то перед остановкой он завершает выполняемую операцию.

2 Целостность данных в буферах передатчика и приемника не гарантируется, если установлен флаг BRK (разрыв линии), или если программное обеспечение произвело остановку приемопередатчика после его повторного перевода в разрешенное состояние.

Режим модема

Приемопередатчик может использоваться как оконечное устройство или как оборудование передачи данных. Сигналы модема в режиме оконечного устройства и их назначение представлено в таблице 15.1.

Таблица 15.1 – Назначение сигналов в режиме модема

| Сигнал | Назначение в зависимости от режима работы | | Режим работы вывода |
|----------|---|------------------------------|---------------------|
| | Оконечное устройство | Оборудование передачи данных | |
| UART_RTS | Готов к передаче данных | Запрос передачи данных | Выход |
| UART_CTS | Запрос передачи данных | Готов к передаче данных | Вход |
| UART_DTR | Приемник данных готов | Источник данных готов | Выход |
| UART_DSR | Источник данных готов | Приемник данных готов | Вход |
| UART_DCD | Обнаружен информационный сигнал | – | Вход |
| UART_RI | Индикатор вызова | – | Вход |

Аппаратное управление потоком данных

Программно активируемый режим аппаратного управления потоком данных позволяет контролировать (приостанавливать и возобновлять) информационный обмен с помощью сигналов UART_RTS и UART_CTS. Иллюстрация взаимодействия двух устройств последовательной связи с аппаратным управлением потоком данных представлена рисунке 15.3.

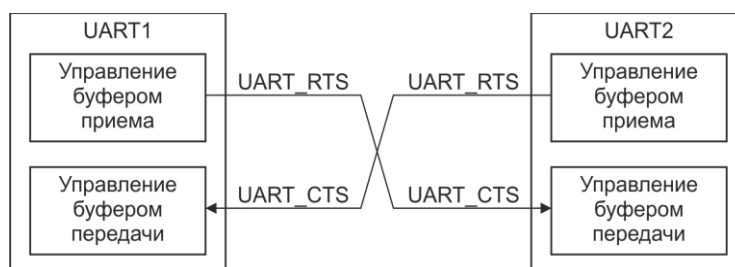


Рисунок 15.3 – Взаимодействие двух устройств последовательной связи с аппаратным управлением потоком данных

Если разрешено управление потоком данных по сигналу RTS, линия UART_RTS переводится в активное состояние только после того, как в буфере приемника появляется заданное количество свободных ячеек.

Если разрешено управление потоком данных по сигналу CTS, передача данных осуществляется только после перевода линии UART_CTS в активное состояние.

Режим аппаратного управления потоком данных задается путем задания битов RTSEN и CTSEN в регистре управления CR.

Примечание – В случае если выбран режим управления потоком данных по RTS, программное обеспечение не может использовать бит RTSEN для проверки состояния линии RTS.

Логика управления потоком данных по RTS использует данные о превышении уровня заполнения буфера приемника. Сигнал на линии UART_RTS переводится в активное состояние только после того, как в буфере приемника появляется заданное количество свободных ячеек. После достижения порогового уровня заполнения буфера приемника сигнал UART_RTS снимается (переводится в пассивное состояние), указывая, таким образом, на отсутствие свободного места для сохранения принятых данных. При этом дальнейшая передача данных должна быть прекращена по завершении передачи текущего кадра.

Обратно в активное состояние сигнал UART_RTS переводится после считывания данных из буфера приемника в количестве, достаточном для того, чтобы заполнение буфера оказалось ниже порогового уровня.

В случае если управление потоком данных по RTS запрещено, но при этом работа приемопередатчика разрешена, прием будет осуществляться до полного заполнения буфера приемника либо до завершения передачи данных.

В случае выбора одного из режимов с управлением потоком данных по CTS передатчик осуществляет проверку состояния линии UART_CTS перед началом передачи очередного байта данных. Передача осуществляется только в случае, если данная линия активна, и продолжается до тех пор, пока активное состояние линии сохраняется и буфер передатчика не пуст.

При переходе линии UART_CTS в неактивное состояние модуль завершает выдачу текущего передаваемого кадра, после чего передача данных прекращается.

Если управление потоком данных по CTS запрещено, и при этом работа приемопередатчика UART разрешена, данные будут выдаваться до опустошения буфера передатчика.

15.2 Интерфейс прямого доступа к памяти

Приемопередатчик оснащен интерфейсом подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMACR. Интерфейс DMA включает в себя шесть сигналов.

UARTRXDMASREQ (для приема) – запрос передачи отдельного символа, инициируемый приемопередатчиком. Размер символа в режиме приема данных – до 12 бит. Сигнал переводится в активное состояние в случае, если буфер приемника содержит, по меньшей мере, один символ.

UARTRXDMABREQ (для приема) – запрос блочного обмена данными, инициируемый приемопередатчиком. Сигнал переходит в активное состояние в случае, если заполнение буфера приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера посредством полей регистра IFLS.

UARTRXDMACLR (для приема) – сброс запроса на DMA, инициируемый приемопередатчиком. В случае если был запрошен блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

UARTTXDMASREQ (для передачи) – запрос передачи отдельного символа, инициируемый приемопередатчиком. Размер символа в режиме передачи данных – до восьми бит. Сигнал переводится в активное состояние в случае, если буфер передатчика содержит, по меньшей мере, одну свободную ячейку.

UARTTXDMABREQ (для передачи) – запрос блочного обмена данными, инициируемый приемопередатчиком. Сигнал переводится в активное состояние в случае, если заполнение буфера передатчика ниже заданного порога. Порог программируется индивидуально для каждого буфера посредством полей регистра IFLS.

UARTTXDMACLR (для передачи) – сброс запроса на DMA, инициируемый контроллером DMA с целью сброса принятого запроса. В случае если был запрошен

блочный обмен данными, сигнал сброса формируется в ходе передачи последнего символа данных в блоке.

Сигналы блочного и одноэлементного обмена данными не являются взаимно исключаящими, они могут быть инициированы одновременно. Например, в случае, если заполнение данными буфера приемника превышает пороговое значение, формируется как сигнал запроса одноэлементного обмена, так и сигнал запроса блочного обмена данными. В случае если количество данных в буфере приема меньше порогового значения, формируется только запрос одноэлементного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера блока.

Пусть, например, нужно принять 19 символов, а порог заполнения буфера установлен равным четырем. Тогда контроллер DMA осуществит четыре передачи блоков по четыре символа, а оставшиеся три символа передаст в ходе трех одноэлементных обменов, поскольку для них блок UART не может инициировать процедуру блочного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае установки в ноль бита управления DMATXDMAE или RXDMAE в регистре управления DMACR.

В случае запрета буферов устройство способно передавать и принимать только одиночные символы и, как следствие, контроллер может инициировать DMA только в одноэлементном режиме. При этом модуль в состоянии формировать только сигналы управления DMAUARTRXDMAREQ и UARTTXDREQ.

Когда буферы включены, обмен данными может производиться в ходе как одноэлементных, так и блочных передач данных, в зависимости от установленной величины порога заполнения буферов и их фактического заполнения.

В таблице 15.2 указаны значения параметров срабатывания запросов блочного обмена UARTRXDMAREQ и UARTTXDREQ в зависимости от порога заполнения буфера.

Таблица 15.2 – Параметры срабатывания запросов блочного обмена данными в режиме DMA

| Пороговый уровень | Количество незаполненных ячеек буфера передатчика | Количество заполненных ячеек буфера приемника |
|-------------------|---|---|
| 1/8 | 28 | 4 |
| 1/4 | 24 | 8 |
| 1/2 | 16 | 16 |
| 3/4 | 8 | 24 |
| 7/8 | 4 | 28 |

В регистре управления DMACR предусмотрен бит DMAONERR, который позволяет запретить DMA от приемника в случае активного состояния линии прерывания по обнаружению ошибки UARTEINTR. При этом соответствующие линии запроса DMA – UARTRXDMAREQ и UARTRXDMAREQ переводятся в неактивное состояние (маскируются) до сброса UARTEINTR. На линии запроса DMA обслуживающие передатчик состояние UARTEINTR не влияет.

15.3 Прерывания

В модуле предусмотрено 11 маскируемых источников прерываний. В результате формируется один общий сигнал, представляющий собой комбинацию независимых сигналов, объединенных по схеме ИЛИ.

Сигналы запросов на прерывания:

UART_RX – от приемника;

UART_TX – от передатчика;

UART_RT – по таймауту приемника;

UART_MS – по состоянию модема;

UART_E – по ошибке;

UART_INT – логическое ИЛИ сигналов запросов на прерывания.

Каждый из сигналов может быть маскирован путем установки соответствующего бита в регистре маски IMSC.

Источник прерывания также можно определить, считав состояние регистра RIS или регистра MIS (маскированные прерывания).

UART_RX

Прерывание возникает в случае обнаружения одного из событий:

- буфер приемника в режиме FIFO и его заполнение достигло заданного порогового значения;

- буфер приемника имеет одну ячейку (режим FIFO запрещен) и принят один кадр данных.

Линия прерывания переходит в высокое состояние и удерживается в нем до тех пор, пока из буфера не будет прочитан как минимум один байт или выполнен программный сброс прерывания (регистр ICR).

UART_TX

Прерывание возникает в случае обнаружения одного из событий:

- буфер передатчика в режиме FIFO и его опустошение достигло заданного порогового значения;

- буфер передатчика имеет одну ячейку (режим FIFO запрещен) и пуст.

Линия прерывания переходит в высокое состояние и удерживается в нем до тех пор, пока в буфер не будет записан как минимум один байт или выполнен программный сброс прерывания.

Запись данных в буфер передатчика допускается как перед разрешением работы приемопередатчика и прерываний, так и после разрешения.

Примечание – Прерывание передатчика работает по фронту, а не по уровню сигнала. В случае если работа приемопередатчика и прерывания от него разрешена до осуществления записи данных в буфер передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера.

UART_RT

Прерывание возникает в случае, если буфер приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание сбрасывается после считывания данных из буфера приемника до его опустошения или программно.

UART_MS

Прерывание возникает в случае изменения любой из линий состояний модема (UART_CTS, UART_DCD, UART_DSR, UART_RI). Прерывание сбрасывается программно.

UART_E

Прерывание возникает в случае ошибки при приеме данных. Оно может быть вызвано:

- ошибкой в структуре кадра;
- ошибкой контроля четности;
- разрывом линии;
- переполнением буфера приемника.

Причину возникновения прерывания можно определить, прочитав содержимое регистра RIS либо регистра MIS.

Сброс прерывания осуществляется программно.

15.4 Программирование

Для программирования рекомендуется следующая последовательность действий:

- запретить работу приемопередатчика;
- дождаться окончания приема и/или передачи текущего байта данных;
- сбросить буфер передатчика посредством сброса бита FEN регистра LCR_H;
- изменить настройки регистра CR;
- разрешить работу приемопередатчика.

16 Контроллеры интерфейса SPI

Контроллер интерфейса SPI реализует интерфейс последовательной синхронной связи в режиме ведущего (мастера) и ведомого устройства и обеспечивает обмен данными с подключенным ведомым или ведущим периферийным устройством в соответствии с одним из трех протоколов фирм Motorola, National Semiconductor, Texas Instruments.

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает полнодуплексный обмен данными по четырехпроводной линии и программное задание фазы и полярности тактового сигнала.

Интерфейс Microwire фирмы National Semiconductor обеспечивает полудуплексный обмен данными с использованием 8-битных управляющих последовательностей.

Интерфейс SSI фирмы Texas Instruments обеспечивает полнодуплексный обмен данными по четырехпроводной линии и возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

Выбор интерфейса осуществляется посредством поля FRF регистра SPI_CR0.

В состав микроконтроллера входят четыре идентичных контроллера SPI0, SPI1, SPI2, SPI3.

В режиме мастера и в режиме ведомого устройства контроллер SPI обеспечивает:

- передачу данных, размещенных в буфере передатчика (восемь 16-разрядных ячеек);

- прием данных и размещение их в буфере приемника (восемь 16-разрядных ячеек).

Контроллер формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов приемника и/или передатчика;
- переполнение буфера приемника;
- наличие данных в буфере приемника по истечении времени таймаута.

Основные характеристики:

- программное управление скоростью обмена;
- программируемая длительность информационного кадра от 4 до 16 бит;
- независимое маскирование прерываний от буферов передатчика и приемника;
- поддержка прямого доступа к памяти (DMA).

16.1 Структура контроллера SPI

Упрощенная функциональная схема контроллера SPI с блоком синхронизации показана на рисунке 16.1.

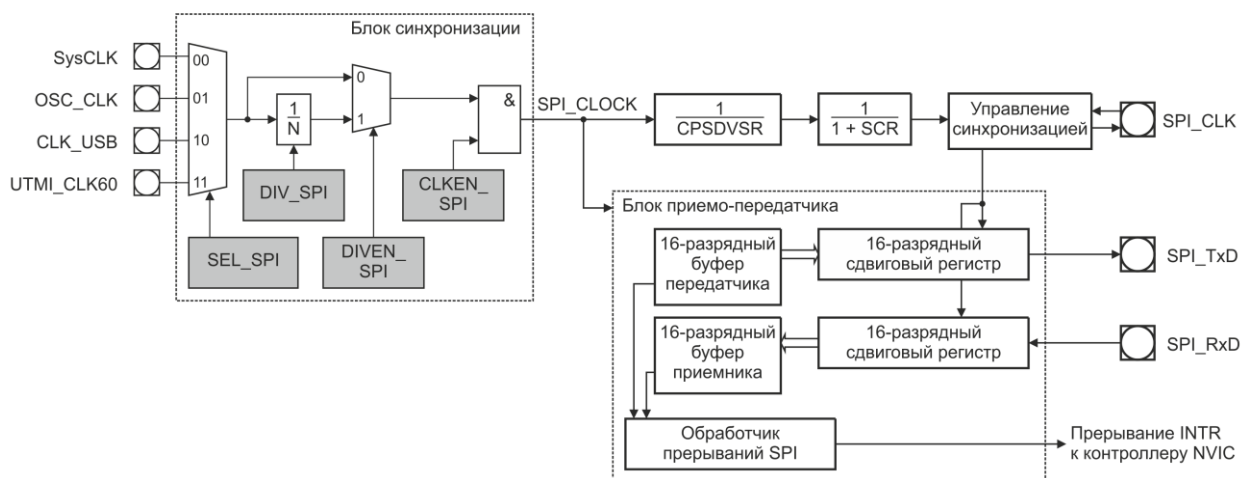


Рисунок 16.1 – Упрощенная функциональная схема контроллера SPI

Синхронизация

Тактирование контроллера SPI осуществляется тактовым сигналом SPI_CLOCK, который формируется на основе одного из четырех базовых синхросигналов (см. раздел 4 «Система тактирования микроконтроллера»).

Существует ограничение на соотношение между тактовыми частотами SysCLK и SPI_CLOCK:

$$\text{SPI_CLOCK} \leq \text{SysCLK}.$$

В режиме мастера на основе сигнала SPI_CLOCK посредством двух последовательно стоящих делителей формируется сигнал тактирования передачи и приема данных с частотой Fspi_clk, которую можно вычислить по формуле

$$F_{\text{spi_clk}} = F_{\text{spi_clock}} / (\text{CPSDVSR} \times (1 + \text{SCR})), \quad (16.1)$$

где – Fspi_clock – частота входного синхросигнала SPI_CLOCK;

CPSDVSR – коэффициент первого делителя частоты (задается в регистре SPI_CPSR);

SCR – коэффициент второго делителя частоты (задается в регистре SPI_CR0).

Сформированный синхросигнал подается на вывод SPI_CLK (skonфигурированный как выход) микроконтроллера и далее к подключенным внешним ведомым устройствам.

В режиме ведомого значения коэффициентов делителей не важны. Внешний синхросигнал подается на вывод SPI_CLK (skonфигурированный как вход) и тактирует прием и передачу данных.

Для корректной работы всегда должны соблюдаться условия:

- в режиме мастера для формируемого синхросигнала

$$F_{\text{spi_clk}} \leq \text{SysCLK}/2;$$

- в режиме ведомого для входящего внешнего синхросигнала

$$F_{\text{spi_clk}} \leq \text{SysCLK}/12.$$

Буферы приема и передачи

Для хранения передаваемых и принятых данных в контроллере SPI имеются два 16-разрядных буфера, организованных по типу FIFO. Каждый буфер может хранить до восьми слов данных. Буфер для передаваемых данных доступен только для записи, а буфер принятых данных – только для чтения.

Данные для передачи записываются в буфер через регистр SPI_DR. Допускается заранее заполнить буфер или записывать в него данные в течение работы контроллера. Состояние буфера можно контролировать с помощью битов TNF и TFE регистра SPI_SR. Если контроллер выключен (сброшен бит SSE регистра SPI_CR1), то запись в регистр SPI_DR приведет к тому, что данные будут размещены в буфере и будут переданы после включения контроллера. Если контроллер включен и выбран режим мастера, то в случае отсутствия данных в буфере запись в регистр SPI_DR приведет к немедленному началу передачи. Если запись данных в регистр SPI_DR происходит во время текущей передачи, то данные размещаются в буфере.

Полученные данные автоматически сохраняются в буфере принятых данных. Извлечь данные из буфера возможно чтением регистра SPI_DR. Состояние буфера можно контролировать с помощью битов RFFF и RNE регистра SPI_SR.

Размер передаваемого кадра данных может быть от 4 до 16 бит, что задается полем DSS регистра SPI_CR0. Если выбран размер кадра менее 16 бит, данные выравниваются по правой границе; неиспользуемые биты игнорируются.

Оба буфера могут генерировать четыре независимых маскируемых прерывания TXINTR (запрос на обслуживание буфера передатчика), RXINTR (запрос на обслуживание буфера приемника), RTINTR (таймаут ожидания чтения данных из буфера приемника), RORINTR (переполнение буфера приемника). Четыре линии прерываний объединены по ИЛИ. Появление любого из прерываний генерирует прерывание контроллера SPI, которое передается в контроллер прерываний NVIC.

Управление прерываниями и контроль их состояния осуществляется посредством регистров SPI_IMSC, SPI_RIS, SPI_MIS, SPI_ICR.

Интерфейс прямого доступа к памяти

Контроллер SPI имеет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром SPI_DMACR.

Сигналы для приема:

- SSPRXDMASREQ – запрос передачи отдельного слова, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер приемника содержит, по меньшей мере, одно слово;

- SSPRXDMABREQ – запрос блочного обмена данными, инициируется приемопередатчиком. Сигнал переходит в активное состояние в случае, если буфер приемника содержит четыре или более слов;

- SSPRXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен пакетный обмен данными, сигнал сброса формируется в ходе передачи последнего слова данных в пакете.

Сигналы для передачи:

- SSPTXDMASREQ – запрос передачи отдельного слова, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер передатчика содержит, по меньшей мере, одну свободную ячейку;

- SSPTXDMABREQ – запрос блочного обмена данными, инициируется приемопередатчиком. Сигнал переводится в активное состояние в случае, если буфер передатчика содержит четыре или менее слов;

- SSPTXDMACLR – сброс запроса на DMA, инициируется контроллером DMA с целью сброса принятого запроса. В случае если был запрошен пакетный обмен данными, сигнал сброса формируется в ходе передачи последнего слова данных в пакете.

Сигналы пакетного и одиночного обмена данными не являются взаимоисключающими, они могут быть инициированы одновременно. Например, в случае, если в буфере приемника более четырех слов данных, формируются как сигнал запроса одиночного обмена, так и сигнал запроса пакетного обмена данными. В случае если количество слов данных в буфере приема меньше четырех, формируется только запрос одиночного обмена. Это бывает полезно в ситуациях, при которых объем данных меньше размера пакета.

Например, нужно принять 19 слов. Тогда контроллер DMA осуществит четыре пакетные передачи блоков по четыре слова, а оставшиеся три слова – в ходе трех одиночных обменов, поскольку для них контроллер SPI не инициирует процедуру пакетного обмена.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом от контроллера DMA.

После снятия сигнала сброса приемопередатчик вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае снятия сигнала разрешения DMA.

Для срабатывания запросов пакетного обмена DMABREQ необходимо, чтобы буферы приемника и передатчика были заполнены наполовину.

16.2 Функционирование

После сброса микроконтроллера работа приемопередатчика запрещена.

Примечание – Модуль SPI может быть сброшен только сигналом сброса микроконтроллера. Приемопередающая логика модуля может быть также сброшена соответствующим битом поля SPIRST регистра PER_RST0, при этом управляющие регистры и FIFO не сбрасываются.

Прежде чем разрешить работу битом SSE регистра SPI_CR1 следует сконфигурировать контроллер посредством регистров SPI_CR0 и SPI_CR1, а также, если это необходимо, запрограммировать маски прерываний.

Динамическое изменение конфигурации устройства не допускается.

Для протокола SPI дополнительно задаются полярность и фаза сигнала (биты SPH и SPO регистра SPI_CR0).

После разрешения работы приемопередатчик готов к обмену данными с внешними устройствами по линиям SPI_TxD (передача данных к внешнему устройству) и SPI_RxD (прием данных от внешнего устройства).

В зависимости от режима работы сигнал на линии SPI_FSS используется либо для кадровой синхронизации (интерфейс SSI, активное состояние – высокий уровень), либо для выбора устройства в режиме ведомого (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

Во всех трех режимах SPI, Microwire и SSI синхросигнал SPI_CLK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SPI_CLK в неактивное состояние используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

Установка бита MS регистра SPI_CR1 включает режим ведомого устройства. В этом режиме разрешение или запрещение передачи данных через выход SPI_TxD контролируется битом SOD. На прием синхросигнала и данных состояние этого бита влияния не оказывает.

Интерфейс SPI

Реализует полнодуплексный режим передачи данных.

Включает одну линию синхронизации SPI_CLK, две линии приема и передачи данных SPI_RxD и SPI_TxD, а также линию выбора устройства (для режима ведомого) SPI_FSS.

Если устройство функционирует в режиме ведомого, то на его вход SPI_FSS должен подаваться низкий уровень сигнала в течение всей передачи кадра (последовательность передаваемых бит данных длиной от 4 до 16 бит).

Передача данных может быть одиночной (один кадр) или непрерывной (более одного кадра подряд). Данные передаются старшим битом вперед.

Особенностью интерфейса SPI является то, что в нем реализована возможность задания полярности и фазы тактового сигнала. Бит SPO регистра SPI_CR0 задает полярность тактового сигнала, т. е. определяет, какой уровень сигнала будет удерживаться на линии SPI_CLK в то время, когда линия не активна.

Бит SPH задает фазу тактового сигнала. Фактически, он задает порядок считывания и выставления данных. По умолчанию, бит SPH сброшен, и выставление данных на

линиях SPI_TxD и SPI_RxD происходит по переднему фронту сигнала синхронизации, а выборка – по заднему.

Передним всегда считается тот фронт сигнала, который является началом передачи первого бита (см. рисунок 16.2).

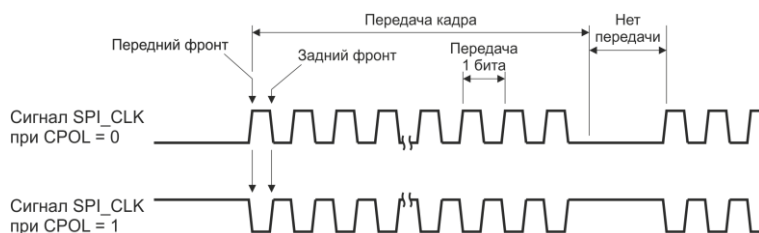


Рисунок 16.2 – Сигнал синхронизации SCK при разных состояниях бита CPOL

Комбинации битов SPO и SPH задают четыре режима обмена данными (см. рисунок 16.3).

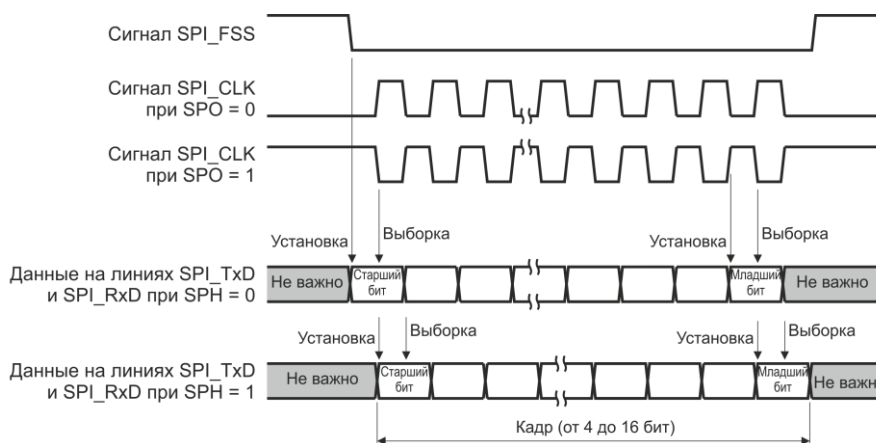


Рисунок 16.3 – Передача кадров данных в интерфейсе SPI

На рисунке 16.4 показано поведение сигналов при непрерывной передаче кадров данных.

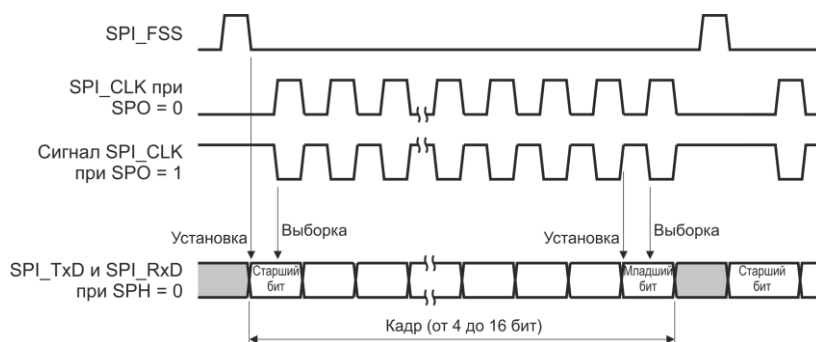


Рисунок 16.4 – Непрерывная передача кадров данных

В режиме непрерывной передачи данных при условии $SPH = 0$ на линии SPI_FSS должны формироваться импульсы между передачами кадров данных. Это связано с тем, что в этом режиме низкий уровень сигнала на линии SPI_FSS ведомого устройства блокирует запись в сдвиговый регистр. Поэтому мастер должен переводить линию SPI_FSS в высокий уровень по окончании передачи каждого кадра, разрешая, таким

образом, запись новых данных. По окончании приема последнего бита кадра линия SSP_FSS переводится в состояние логической единицы по истечении одного такта сигнала SPI_CLK.

В режиме непрерывной передачи данных при условии $SPH = 1$ низкий уровень сигнала на линии SPI_FSS не блокирует запись в сдвиговый регистр. Поэтому линия SPI_FSS может оставаться в состоянии нуля в течение передачи всех кадров и переведена в состояние логической единицы только по окончании передачи.

Интерфейс Microwire

Реализует полудуплексный режим передачи данных.

Включает линию синхронизации SPI_CLK, две линии приема и передачи данных SPI_RxD и SPI_TxD, а также линию выбора устройства (для режима ведомого) SPI_FSS.

Если устройство функционирует в режиме ведомого, то на его вход SPI_FSS должен подаваться низкий уровень сигнала в течение всей передачи кадра (последовательность передаваемых бит данных длиной от 4 до 16 бит).

Передача данных может быть одиночной (один кадр) или непрерывной (более одного кадра подряд). Данные передаются старшим битом вперед.

Перед началом передачи линия SPI_FSS переводится в низкое состояние.

Каждая передача начинается с передачи от мастера к ведомой 8-битной управляющей последовательности. В течение передачи этой последовательности приемник мастера не обрабатывает входящие данные. После того как управляющая последовательность передана и декодирована одним из ведомых устройств, этот ведомый выдерживает паузу в один такт синхросигнала и начинает передавать мастеру кадр данных (см. рисунок 16.5).

Выставление данных происходит по заднему фронту сигнала SPI_CLK, а считывание – по переднему.

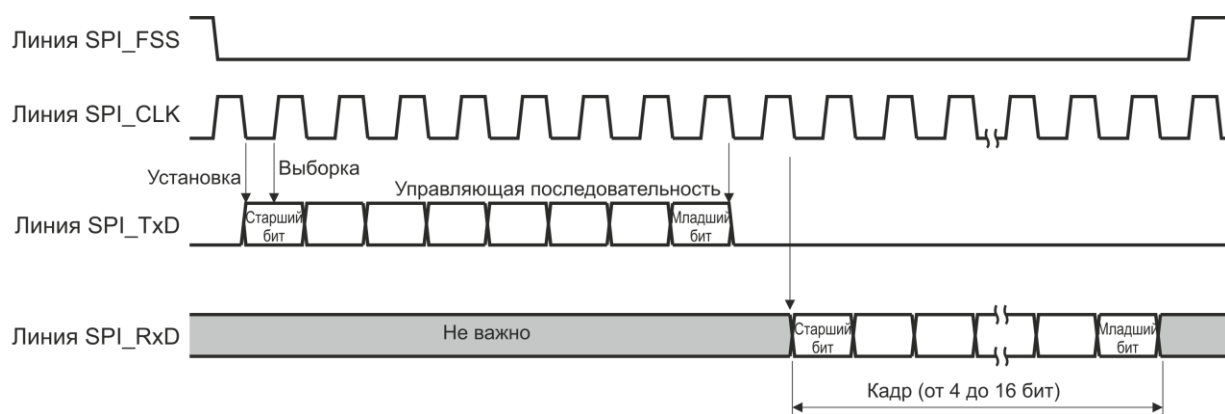


Рисунок 16.5 – Передача кадра данных в интерфейсе Microwire

По окончании приема данных линия SPI_FSS переводится в высокое состояние.

Примечание – В течение времени, когда передается управляющая последовательность и между передачами линия SPI_RxD может находиться в третьем состоянии.

В режиме непрерывной передачи начало и завершение передачи нескольких кадров данных аналогично передаче одного кадра. Линия SPI_FSS удерживается в нуле в течение всего сеанса передачи. По окончании передачи одного кадра данных начинается передача управляющей последовательности без паузы (см. рисунок 16.6).

Примечание – Буферы FIFO приема и передачи данных не очищаются автоматически, даже в случае запрещения работы сбросом бита SSE.

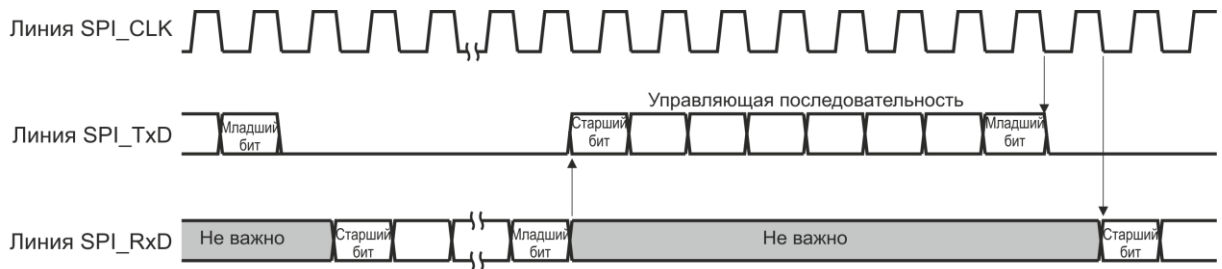


Рисунок 16.6 – Передача кадров данных в интерфейсе Microwire

Интерфейс SSI

Реализует полнодуплексный режим передачи данных.

Включает одну линию синхронизации SPI_CLK, две линии приема и передачи данных SPI_RxD и SPI_TxD, а также линию выбора устройства SPI_FSS.

Перед началом передачи каждого кадра на линии SPI_FSS формируется импульс длительностью в один период сигнала SPI_CLK. Далее мастер и ведомый передают данные. Установка данных производится по переднему фронту синхросигнала, а выборка – по заднему (см. рисунок 16.7). Весь цикл передачи начинается сразу же после появления хотя бы одного элемента в буфере FIFO передатчика.

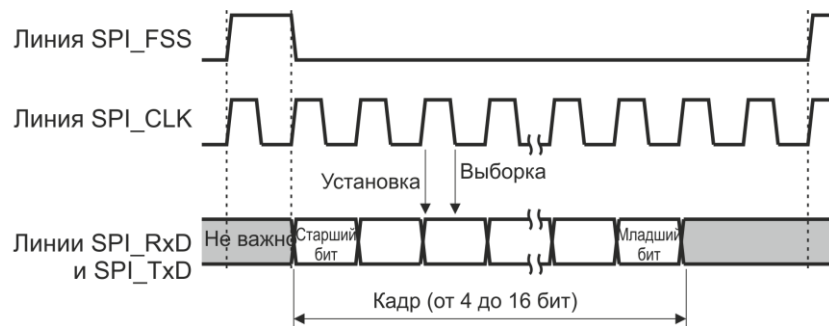


Рисунок 16.7 – Передача кадра данных в интерфейсе SSI

Режим непрерывной передачи кадров данных показан на рисунке 16.8.

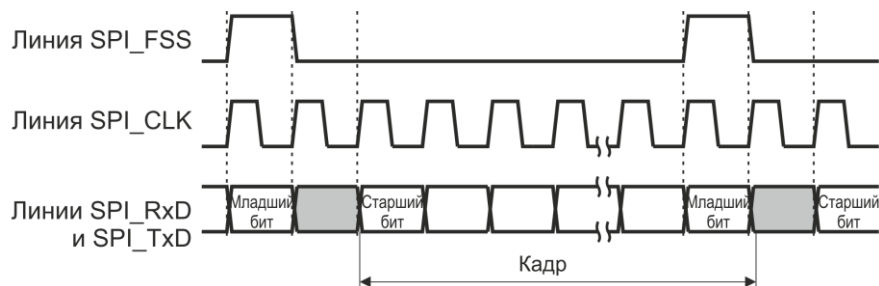


Рисунок 16.8 – Передача кадров данных в интерфейсе SSI

16.3 Прерывания

Четыре маскируемых источника прерываний объединены по ИЛИ, и в результате формируется один общий сигнал прерывания SPI, поступающий на контроллер NVIC.

Сигналы источников запросов на прерывания:

RX – от приемника при заполнении его наполовину;

TX – от передатчика при опустошении его наполовину;

RT – по таймауту приемника (буфер приемника не пуст и не было попыток обращения к нему в течение времени равного передаче 32 бит);

ROR – по переполнению буфера приемника.

Каждый из сигналов может быть маскирован путем сброса соответствующего бита в регистре маски SPI_IMSC.

Для определения источника прерывания следует прочитать регистр SPI_RIS или регистр SPI_MIS (маскированные прерывания).

17 Контроллер интерфейса I2C

Модуль контроллера I2C обеспечивает полную поддержку двухпроводного последовательного синхронного интерфейса I2C/SMBus. Результат такой совместимости – легкое соединение со многими запоминающими устройствами и устройствами ввода-вывода, включая EEPROM, SRAM, счетчики, АЦП, ЦАП, периферийные устройства.

Функциональные возможности модуля:

- совместимость с протоколами SMBus 1.1 и SMBus 2.0, ACCESS.Bus, I2C 2.1;
- поддержка скоростного/стандартного (FS) и высокоскоростного (HS) режимов;
- программирование действий мастера/ведомого;
- возможность подключения к шине нескольких ведущих устройств, т. е. поддержка режима мультимастер (ММ);
- один программно задаваемый адрес;
- 7- или 10-битная адресация ведомого;
- поддержка адреса общего вызова.

Особые возможности SMBus:

- отслеживание времени простоя линии SCL;
- наличие функции отслеживания ошибок в пакетах данных (PEC) с использованием метода расчета контрольной суммы (CRC);
- поддержка адреса отклика мастера;
- поддержка полинга и контроля прерываний.

17.1 Протокол шины

Протокол I2C использует двухпроводной интерфейс для двусторонней связи между устройствами, подключенными к шине. Двухнаправленная шина состоит из двух линий: данных SDA и тактового сигнала SCL. Эти линии подключены к источнику питания через подтягивающие резисторы. Шинные формирователи любых устройств, подключаемых к шине, выполняются по схеме с открытым коллектором или открытым стоком. Устройства могут выставить только низкий уровень на соответствующей линии. Следовательно, обе линии SDA и SCL реализуют функцию «монтажное И».

Протокол поддерживает режим мультимастер, в котором шина может контролироваться одним или несколькими устройствами из подключенных к шине. Каждое устройство, подключенное к шине, имеет свой адрес и может быть как приемником, так и передатчиком (некоторые только приемниками).

Операции с данными

Устройство, которое начинает передачу данных, становится мастером. Мастер генерирует тактовый сигнал SCL, а также инициирует и завершает передачу данных по шине. За один такт сигнала SCL передается один бит данных по линии SDA (рисунок 17.1).

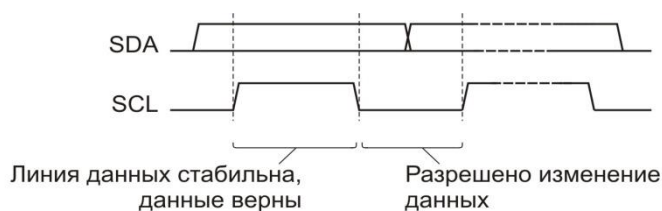


Рисунок 17.1 – Передача бита данных

Данные валидны (верны), пока уровень сигнала на линии SCL высокий. Когда на линии SCL низкий уровень сигнала, данные могут меняться.

Старт и стоп

Состояние старта формируется тогда, когда на линии SCL держится высокий уровень сигнала, а на линии SDA возникает перепад уровня сигнала из высокого в низкий (см. рисунок 17.2).

Состояние стопа (останова) формируется тогда, когда на линии SCL держится высокий уровень сигнала, а на линии SDA возникает перепад уровня сигнала из низкого в высокий (см. рисунок 17.2).

Состояния старта и стопа формирует только мастер.

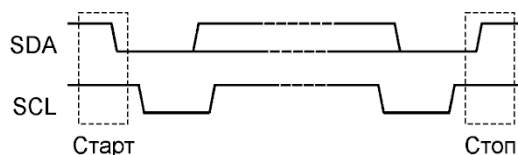


Рисунок 17.2 – Состояния старта и стопа

После того, как сформировано состояние старта, шина считается занятой и другие устройства не должны пытаться управлять ею. Шина считается занятой до тех пор, пока не будет сформировано состояние стопа. В середине передачи может быть сформировано состояние повторного старта, если мастеру нужно обратиться к другому ведомому или если требуется изменение направления передачи данных без потери контроля над шиной (см. рисунок 17.3).

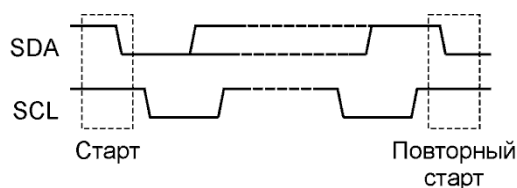


Рисунок 17.3 – Состояние повторного старта

Арбитраж

Арбитраж выполняется в момент времени, когда на линии SCL находится «1». Два устройства могут сгенерировать стартовое состояние в одно и то же время. Далее арбитраж будет продолжаться до тех пор, пока одно из устройств сформирует «0», а другое – «1» на линии SDA. Устройство, которое установило «1» на линии SDA, проигрывает арбитраж. На рисунке 17.4 приведен пример арбитража.

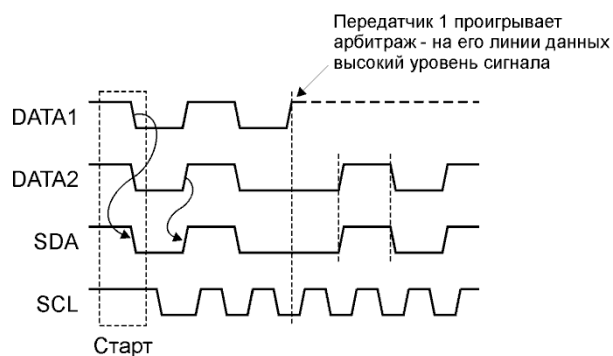


Рисунок 17.4 – Арбитраж на линии SDA

Два устройства передают свои данные DATA1 и DATA2 на линию SDA. В момент времени, когда очередной бит данных DATA1 равен «1», а бит данных DATA2 равен «0», второе устройство выигрывает арбитраж и продолжает передачу своих данных, а первое устройство прекращает передачу.

Если устройство проигрывает арбитраж во время передачи первого байта после старта (во время передачи адреса ведомого), оно становится ведомым приемником и мониторит передаваемый адрес на случай совпадения. Арбитраж также может быть проигран в режиме мастера приемника во время квитирования или в режиме ведомого передатчика во время ответа на адрес отклика на сигнал предупреждения.

В случае проигрывания арбитража в битовом поле MODE регистра ST устанавливается соответствующий код и генерируется прерывание.

Синхронизация

Синхронизация тактовых сигналов разных устройств, подключенных к шине I2C, реализуется в случаях, когда несколько устройств являются мастерами, и выполняется с использованием той особенности, что линия SCL реализована как монтажное «И» линий тактовых сигналов этих устройств. Для примера рассмотрим синхронизацию двух мастеров с линиями тактовых сигналов CLK1 и CLK2 (см. рисунок 17.5).

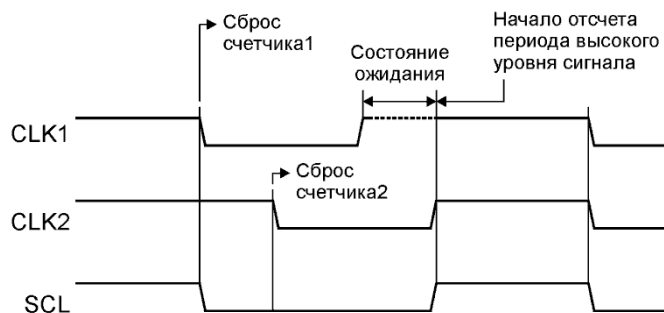


Рисунок 17.5 – Синхронизация

Линия SCL переводится в состояние «0» сразу, как только один из мастеров выставляет на своей линии тактового сигнала низкий уровень сигнала (CLK1 на рисунке 17.5). При этом его внутренний счетчик длительности низкого уровня сигнала сбрасывается и начинает отсчет. Второй мастер выставляет низкий уровень позже, и его счетчик также сбрасывается (CLK2).

Как только внутренний счетчик первого мастера переполнится, мастер выставит на линии CLK1 высокий уровень сигнала. Тем не менее, линия SCL будет по-прежнему оставаться в состоянии «0», удерживаемая вторым мастером. В связи с этим, первый мастер перейдет в состояние ожидания (см. рисунок 17.5). Когда переполнится счетчик второго мастера, он выставит на линии CLK2 высокий уровень сигнала, и в этот момент линия SCL перейдет в состояние «1». С этого момента внутренние счетчики длительности высокого уровня сигнала обоих мастеров начнут синхронный отсчет.

Каждая передача данных состоит из начального состояния «старт», состояний передач битов и состояния «стоп». Данные передаются старшим битом (MSB) вперед. Передача каждого байта завершается квитированием, т.е. приемник подтверждает окончание приема сигналом подтверждения (ACK). Ведомое устройство может увеличивать паузу между тактовыми импульсами, удерживая на линии SCL сигнал низкого уровня, пока происходит обработка принятых данных или подготовка данных для следующей передачи. Этот процесс может происходить после передачи любого бита/байта (см. рисунок 17.6).

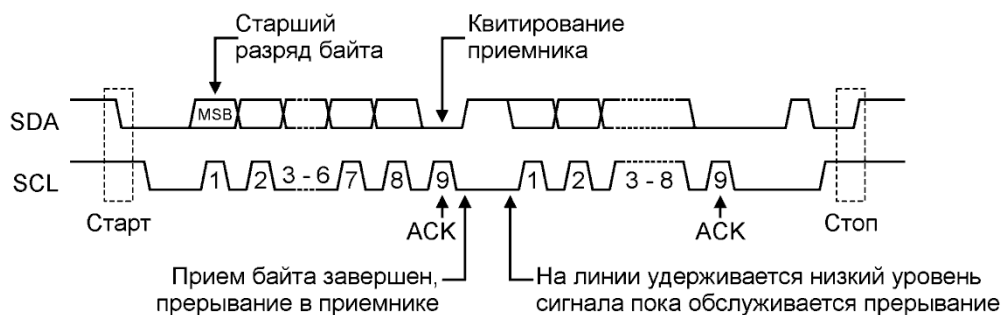


Рисунок 17.6 – Передача данных

Квитирование

Каждый байт посылки должен быть завершен квитированием, т. е. ответом на прием сигнала запроса подтверждения приема (ACK). На рисунках 17.6 и 17.7 показано положение момента квитирования в пределах посылки.

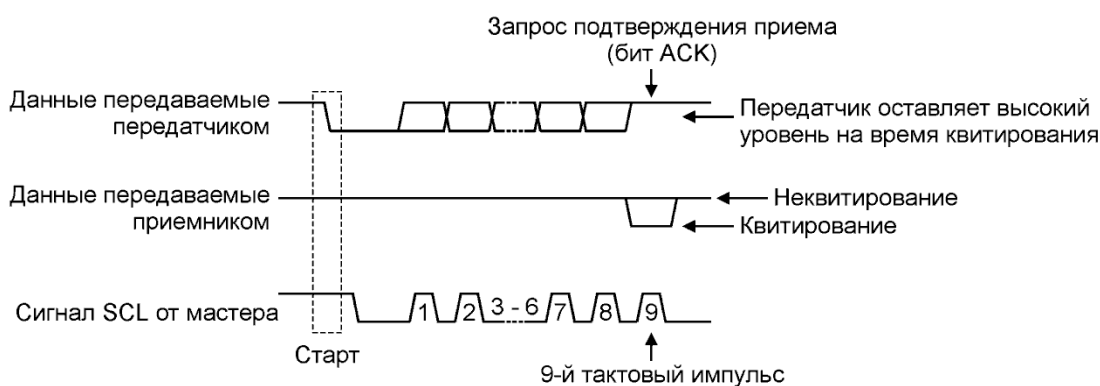


Рисунок 17.7 – Квитирование и неквитирование бита подтверждения ACK

Бит запроса подтверждения приема генерируется мастером. Передатчик (мастер или ведомый) в момент девятого такта синхросигнала оставляет линию SDA в состоянии «1» (бит ACK). В свою очередь, приемник должен сбросить линию SDA в «0» в течение времени, пока на линии SCL удерживается высокий уровень девятого импульса тактового сигнала, т. е. квитировать прием (см. рисунок 17.7). Если приемник не отвечает на запрос подтверждения и не подтверждает прием байта, то он оставляет линию SDA без изменений в состоянии «1», т. е. не квитирует прием.

Примечание – Все устройства, подсоединенные к шине I2C, в обязательном порядке должны квитировать бит ACK при получении байта с их собственным адресом. Этот механизм используется для отслеживания наличия отключившихся (самостоятельно или по каким-то причинам) от шины устройств.

Ведомое устройство имеет право не квитировать бит ACK в следующих случаях:

- если ведомый не может принять данные или он занят. Мастер, обнаружив неквитирование байта, должен сгенерировать состояние стопа и прервать передачу. Как альтернатива, ведомый может затянуть период низкого уровня сигнала тактирования на линии SCL для завершения своих операций и продолжить передачу;
- если ведомый обнаружил некорректную команду или некорректные данные. В этом случае, ведомый должен неквитировать принятый байт. Мастер, обнаружив неквитирование байта, должен сгенерировать состояние стопа и повторить передачу;
- если мастер функционирует как приемник, то, приняв байт, он должен сообщить ведомому об окончании данных неквитированием бита ACK, посланного ведомым. После

этого ведомый передатчик должен освободить линию SDA для того, чтобы мастер смог сгенерировать состояние завершения передачи (состояние стопа).

Формат передачи данных с 7-битной адресацией

На рисунке 17.8 показана передача адреса и двух байт данных. Каждому устройству, подключенному к шине, присваивается уникальный 7-битный адрес. Первые семь бит, передаваемые после старта, представляют собой адрес ведомого, восьмой бит (R/W#) определяет направление передачи – от ведомого (чтение, если R/W# = «1») или к ведомому (запись, если R/W# = «0»).

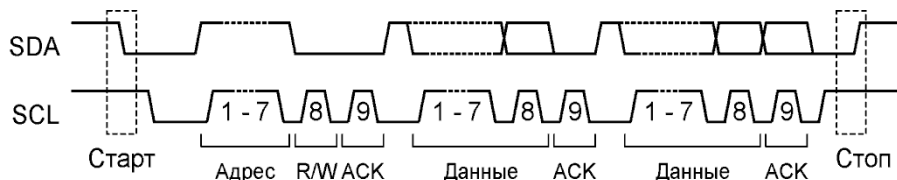


Рисунок 17.8 – Передача данных с 7-битной адресацией

Каждый ведомый, получивший байт адреса, сравнивает его со своим собственным адресом. Если адрес распознается как «свой», ведомый квитирует прием и далее, в зависимости от состояния бита R/W#, становится передатчиком или приемником.

Протокол SMBus/I2C позволяет генерировать адрес общего вызова для одновременного обращения ко всем устройствам, подключенным к шине. Первым передается адрес общего вызова (00h), затем следует байт назначения общего вызова. Ведомые, которые ожидают данные, квитируют этот байт и становятся приемниками, остальные игнорируют общий вызов.

Протокол SMBus/I2C поддерживает уникальную функцию – распознавание адреса отклика на сигнал предупреждения (Alert Response Address – ARA). В системах с несколькими ведомыми каждое устройство может послать мастеру сигнал предупреждения. Для этого используется дополнительная третья линия ALERT#, физически идентичная линиям SDA и SCL, реализованная по принципу монтажное «И». К этой линии также подключаются все устройства. Когда какому-то ведомому (или нескольким ведомым) необходимо обратиться к мастеру, он (или они) выставляет на линии ALERT# низкий уровень сигнала – это сигнал предупреждения (см. рисунок 17.9).



Рисунок 17.9 – Передача адреса отклика на сигнал предупреждения

Мастер, обнаружив «0» на линии ALERT#, обращается ко всем ведомым, посылая адрес отклика на сигнал предупреждения (ARA). Адрес состоит из семи битов (0001_100b) и бита R/W# = «1» (чтение). Ведомый, который отправил сигнал предупреждения, получив ARA, квитирует его и затем отправляет свой 7-битный адрес (восьмой бит может быть как «0», так и «1»), сообщая, таким образом, ведомому, какое именно устройство послало сигнал предупреждения. Кроме этого, ведомый, который выставлял «0» на линии ALERT#, должен перестать удерживать линию, чтобы на ней установился высокий уровень сигнала. В том случае, если несколько устройств посылали

сигнал предупреждения, то после получения ARA, свой адрес передает то устройство, которое захватывает шину по стандартным правилам арбитража. Если после обслуживания ведомого мастер все еще обнаруживает на линии ALERT# низкий уровень сигнала, он понимает это как то, что сигнал предупреждения посылался несколькими ведомыми. Мастер снова отправляет ARA и затем общается со следующим ведомым. Появление на линии ALERT# высокого уровня сигнала означает, что все ведомые, которые требовали обращения, обслужены.

Примечание – Описываемый в настоящем КФДЛ.431295.057ТО модуль I2C не имеет выделенной линии ALERT#. При необходимости, пользователь может задействовать свободный вывод микроконтроллера и программно реализовать возможность передачи сигнала предупреждения от ведомого к мастеру. В свою очередь, функция распознавания адреса отклика (ARA) и последующей отправки собственного адреса реализована полностью. Включить функцию можно установкой бита SMBARE в регистре CTL0.

Формат передачи данных с 10-битной адресацией

10-битная адресация позволяет адресовать до 1024 ведомых устройств, с использованием резервной комбинации 1111_0xxb, которая передается по линии SDA сразу после старта. 10-битный формат полностью совместим с 7-битным форматом и может использоваться одновременно с ним, что позволяет соединять по шине I2C устройства с разной адресацией.

Основной идеей формата является передача 10-битного адреса в двух первых байтах, следующих сразу после старта. В первом байте передается значение 1111_0xxb, где «xx» – это два старших бита адреса и бит R/W# (на рисунке 17.10 обозначен символом «W» – запись), который должен быть равен «0», чтобы ведомый понял, что в следующем байте будут переданы остальные 8 бит адреса. Во втором байте передаются 8 бит адреса (см. рисунок 17.10).

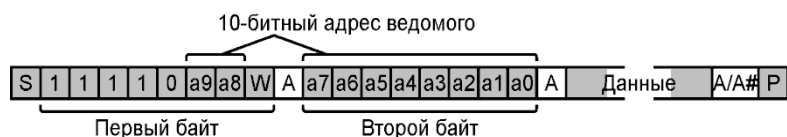


Рисунок 17.10 – Передача данных ведомому с 10-битным адресом (для расшифровки обозначений, применяемых на рисунке, следует обратиться к таблице 17.1)

Чтобы осуществить чтение ведомого, которого адресует мастер после второго байта адреса, следует отправить бит повторного старта и затем комбинацию 1111_0xxb и бит R/W# (обозначен символом «R» – чтение), который на этот раз равен «1» (см. рисунок 17.11).

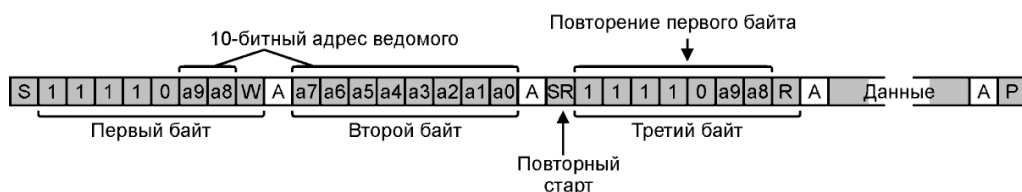



Рисунок 17.11 – Получение данных от ведомого с 10-битным адресом

На рисунках 17.10 и 17.11 биты посылки условно обозначены буквами S, W и др. или состояния битов указаны непосредственно «0» или «1». В дальнейшем на подобных рисунках, поясняющих содержимое посылки при передаче или приеме данных, будут

применяться такие же и другие обозначения. Все обозначения, которые будут использоваться, указаны в таблице 17.1 с подробными пояснениями.

Таблица 17.1 – Условные обозначения, принятые на рисунках, показывающих содержимое посылок данных или адресов на линии SDA

| Обозначение | Расшифровка обозначения |
|---|---|
| S | Состояние старта. Символом «S» обозначается стартовый бит посылки |
| SR | Состояние повторного старта |
| R/W | Бит указания направления передачи. В тексте настоящего описания он упоминается как R/W#. Наличие этого обозначения в бите посылки указывает на то, что этот бит может быть равен как «0», так и «1» |
| R | Частный случай обозначения бита направления передачи R/W#. Если в обозначении бита стоит символ «R», то это указывает на то, что в данной посылке бит R/W# должен быть равен «1», т. е. направление передачи данных происходит от ведомого к мастеру (чтение) |
| W | Частный случай обозначения бита направления передачи R/W#. Если в обозначении бита стоит символ «W», то это указывает на то, что в данной посылке бит R/W# должен быть равен «0», т. е. направление передачи данных происходит от мастера к ведомому (запись) |
| A/A# | Бит квитированного/неквитированного приема, посылаемый приемником в ответ на запрос передатчика подтвердить прием. Наличие этого обозначения в бите посылки указывает на то, что этот бит может быть равен как «0», так и «1» |
| A | Частный случай обозначения бита A/A#. Если в обозначении бита стоит символ «A», то это указывает на то, что в данной посылке в ответ на запрос подтверждения приема байта произошло квитирование, т. е. приемник установил линию SDA в «0». В тексте настоящего описания квитированный бит запроса подтверждения приема обозначается как ACK |
| A# | Частный случай обозначения бита A/A#. Если в обозначении бита стоит символ «A#», то это указывает на то, что в данной посылке в ответ на запрос подтверждения приема байта произошло неквитирование, т. е. приемник не изменил линию SDA и оставил ее в состоянии «1». В тексте настоящего описания, неквитированный бит запроса подтверждения приема обозначается как NACK |
| P | Состояние окончания передачи. Символом «P» обозначается стоповый бит посылки |
| Код мастера | 8-битный код мастера. Значение 0000_1xxx _b , где «xxx» – уникальный код каждого мастера в системе нескольких устройств |
| Адрес | 7-битный адрес ведомого, передаваемый мастером |
| Адрес ведомого | Адрес ведомого, передаваемый во втором байте посылки. В режиме HS это 7-битный адрес, на что указывает идущий следом бит R/W#, в остальных случаях это восемь младших бит 10-битного адреса |
| Данные | Байт или несколько байт данных |
| GC | Байт адреса общего вызова (0000_0000 _b) |
| AR | Адрес отклика (0001_100 _b) |
|  | Изображение числа в овале с линией, прикрепляющей его к изображению передачи битов, обозначает код операции и указывает момент, в который этот код записывается в поле MODE регистра ST |
| Цветное поле | Серым цветом обозначены биты, передаваемые от мастера к ведомому |
| Белое поле | Белым цветом обозначены биты, передаваемые от ведомого к мастеру |

17.2 Функциональное описание

Структурная схема модуля I2C представлена на рисунке 17.12. Далее приводится краткое описание назначения блоков модуля.

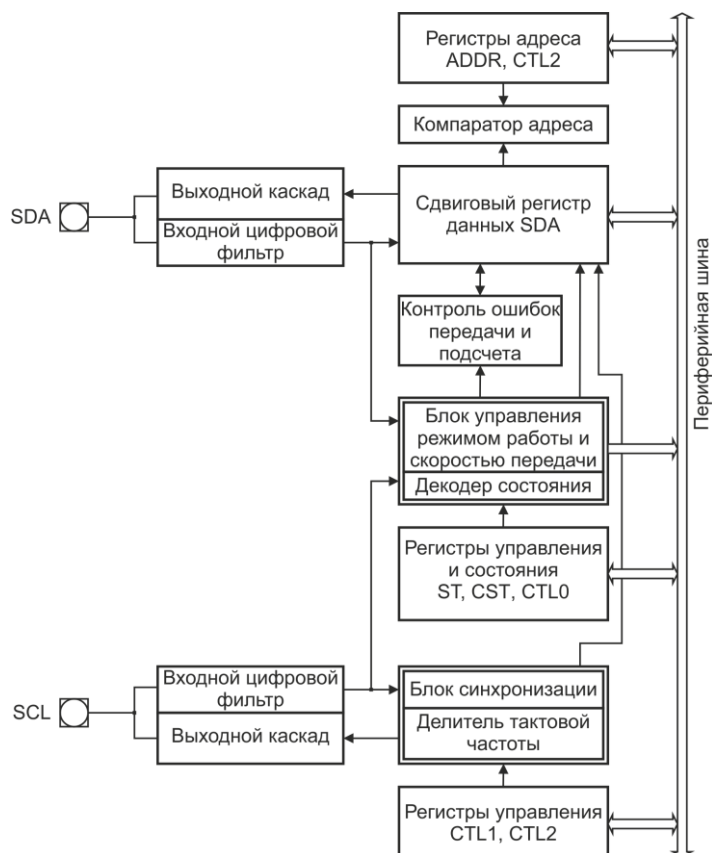


Рисунок 17.12 – Структурная схема модуля I2C

Входные и выходные каскады линий SDA и SCL

Для обеих линий используются входные шумовые фильтры. В режиме FS эти фильтры подавляют любые импульсы входного сигнала, длительность которых не превышает один такт системного синхросигнала. Выходные каскады включают в себя понижающие (до уровня «0») устройства с открытым стоком. Функционирование входных и выходных каскадов зависит от состояния модуля I2C, т. е. включен или выключен.

Управление режимом работы и опрос состояния

Управление модулем осуществляют блоки управления режимом работы и скоростью передачи, регистров управления и состояния. В состав этих блоков входят следующие регистры:

- ST – содержит биты, отражающие текущую конфигурацию модуля I2C (мастер или ведомый, передатчик или приемник) и бит флага прерывания;
- CST – является одновременно регистром управления шиной и регистром состояния шины;
- CTL0 – управляет генерированием состояний старта, повторного старта и останова, а также квитированием;
- CTL1 и CTL2 – устанавливают параметры тактового сигнала в режиме мастера и контролируют режим 10-битной адресации.

Регистры адреса и компаратор адреса

В регистр адреса ADDR может быть записан 7-битный адрес, который является адресом устройства при работе его в режиме ведомого. Распознавание адреса включается установкой бита SAEN.

Компаратор адреса сравнивает принятый 7-битный адрес со значением, хранящимся в поле ADDR. Если разрешено распознавание адреса общего вызова (установлен бит GCMEN регистра CTL0), то компаратор сравнивает принятый адрес со значением 0000_000b. Если разрешено распознавание адреса отклика на сигнал предупреждения (установлен бит SMBARE регистра CTL0), то компаратор сравнивает принятый адрес со значением 0001_100b.

Если включен режим 10-битной адресации (одновременно установлены биты SAEN и S10EN регистров ADDR и CTL2, соответственно), компаратор сравнивает старшие пять битов первого полученного байта со значением 1111_0b, а следующие два бита со значением второго и первого битов поля S10ADR регистра CTL2. Старший бит второго полученного байта сравнивается со значением нулевого бита поля S10ADR, а оставшиеся семь битов – со значением битового поля ADDR регистра ADDR (см. рисунок 17.13).

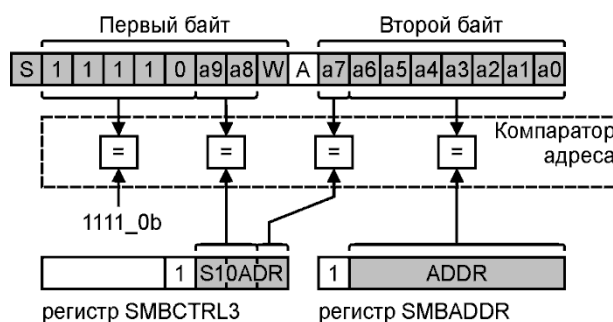


Рисунок 17.13 – Компаратор адреса в режиме 10-битной адресации

Сдвиговый регистр данных

Регистр SDA представляет собой сдвиговый регистр, используемый для приема и передачи данных. Старший бит регистра передается/принимается первым, младший бит – последним. Запись в регистр SDA возможна только, если установлен бит INT регистра ST. Регистр может быть прочитан в любой момент времени, но прочитанные данные будут гарантированно достоверными только при установленном бите INT. Регистр SDA не очищается при сбросе и хранит случайные данные до тех пор, пока не будет перезаписан программно или аппаратно после приема байта.

Генерация тактового сигнала и синхронизация

Последовательный тактовый сигнал (выходной сигнал модуля I2C в режиме мастера) формируется генератором на базе системного тактового сигнала (с частотой fosc).

Модуль I2C может функционировать в двух глобальных режимах – стандартном/скоростном (FS) и высокоскоростном (HS).

В режиме FS используется 7-битный делитель. Значение старших 6 бит определяется значением битового поля SCLFRQ регистра CTL1, а младший бит всегда равен нулю (деление производится только на четное число). Минимальный и максимальный коэффициенты деления – 8 и 128 соответственно. Блок синхронизации тактового сигнала производит синхронизацию генератора тактового сигнала и выходного сигнала SCL с тактовым сигналом других устройств, подключенных к шине.

В режиме HS используется 4-битный делитель. Значение старших бит определяется значением битового поля HSDIV регистра CTL2, младший бит всегда равен нулю.

Определяемое спецификацией протокола SMBus наименьшее время ожидания на линии SCL составляет 25 мс. Если пауза между двумя тактовыми импульсами превысила 25 мс, то устройство должно прервать текущую передачу. Мастер должен сформировать состояние старта в процессе передачи или после ее окончания. Водомый должен освободить шину. Устройства, обнаружившие данное состояние, должны восстановить свои соединения и ожидать формирования состояния старта в пределах 10 мс.

Для отслеживания периодов ожиданий на шине в модуле I2C имеется счетчик времени ожидания. Функциональная схема счетчика показана на рисунке 17.14.

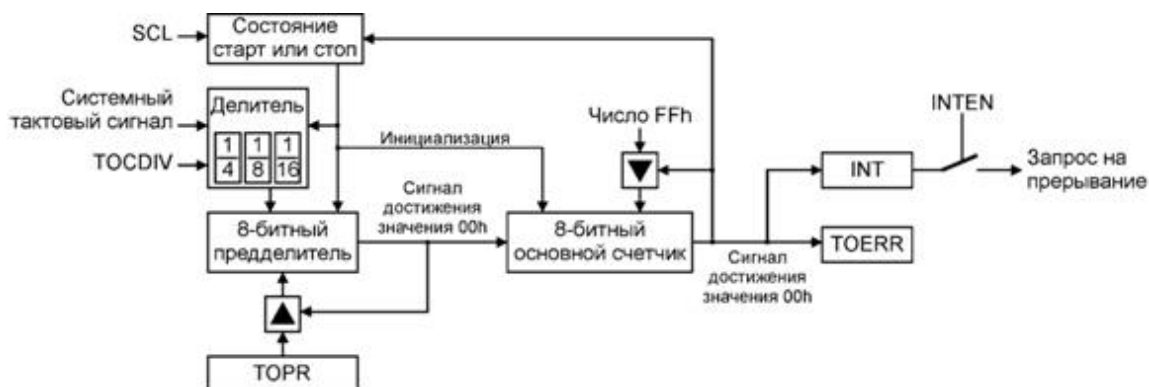


Рисунок 17.14 – Функциональная схема счетчика времени ожидания

Счетчик времени ожидания состоит из делителя, 8-битного программируемого предделителя и 8-битного основного счетчика. Все элементы счетчика времени ожидания начинают работу по отрицательному фронту сигнала на линии SCL (если работа счетчика разрешена). Положительный фронт сигнала на линии SCL сбрасывает значения делителя, предделителя и основного счетчика. Предделитель считает вниз, начиная со значения, записанного в регистр TOPR. После достижения нуля счетчиком предделителя, он загружается значением из регистра TOPR. Основной счетчик считает вниз от значения FFh. Каждое достижение нуля предделителем декрементирует значение основного счетчика. Обнуление основного счетчика и загрузка его значением FFh вызывает остановку основного счетчика, предделителя и делителя и установку флага TOERR в регистре CST. Дополнительно устанавливается флаг INT и если разрешено, генерируется прерывание.

Период времени ожидания определяется следующим выражением

$$T_{\text{ожид}} = T_{\text{osc}} \times \text{TOCDIV} \times (\text{SMBTOPR} + 1) \times 256, \quad (17.1)$$

где T_{osc} – период системного тактового сигнала с частотой f_{osc} .

Арбитраж и обнаружение ошибок на шине

Арбитраж в режиме мастера передатчика может быть потерян в случае, когда два мастера одновременно формируют состояние старта и начинают передачу данных. Потеря арбитража может происходить как во время передачи адреса, так и во время передачи данных.

В случае потери приоритета при передаче байта адреса, мастер переходит в режим ведомого приемника и начинает принимать адрес. Если принятый адрес оказался «своим», модуль I2C далее функционирует в режиме ведомого. Если принятый адрес не оказался «своим», то модуль I2C переходит в режим безадресного ведомого.

В случае потери приоритета при передаче байта данных модуль I2C сразу переходит в режим безадресного ведомого.

Обнаружение и исправление ошибок на шине

Состояние ошибки на шине возникает в том случае, если во время передачи адреса/данных или во время квитирования на шине обнаруживаются состояния старта или стопа. При обнаружении ошибки на шине выполняются действия:

- в поле MODE регистра ST записывается код ошибки 1Fh;
- генерируется прерывание (если разрешено);
- модуль I2C переходит в режим безадресного ведомого;
- линии SDA и SCL освобождаются.

Обнаружение ошибки на шине может вызвать у простой шины некорректное формирование состояния старта и отключение модуля I2C. Поэтому для возврата к нормальной работе следует выполнить действия:

- выключить и снова включить модуль I2C (бит ENABLE в регистре CTL1);
- в течение времени простоя проверить, не подключен ли другой активный мастер к шине (бит BB регистра CST должен быть обнулен);
- в режиме мастера шины сформировать состояние старта, передать адрес и затем сформировать состояние останова, таким образом, проведя синхронизацию всех ведомых устройств (в том числе и тех, которые не обнаружили ошибку на шине).

Режим IDLE

Переход в режим IDLE происходит при отключении внешнего сигнала тактирования модуля I2C записью нуля в бит I2CEN регистра APB_CLK. Переход в режим IDLE подобен программному выключению модуля I2C (очистка бита ENABLE в регистре CTL1). Регистры CTL0, ST и CST очищаются, чтобы гарантировать нормальный старт после возобновления функционирования модуля.

Выход из режима IDLE осуществляется записью единицы в бит I2CEN и включением модуля битом ENABLE.

17.3 Инициализация и функционирование

В целом модуль I2C поддерживает два базовых режима – режим FS и режим HS.

Стандартный/скоростной режим или режим FS – стандартный режим работы, в котором модуль функционирует по умолчанию. Диапазон частот сигнала на линии SCL – от 23,6 до 750,3 кГц (при XTAL1 = 24 МГц).

Высокоскоростной режим или режим HS – режим работы, который включается программно. Режим HS значительно превосходит режим FS по скорости – диапазон частот сигнала на линии SCL от 0,25 до 2 МГц (при XTAL1 = 24 МГц).

Все операции режима HS начинаются в режиме FS в следующем порядке:

- стартовое состояние;
- 8-битный код мастера (значение 0000_1xxxh, где «xxx» – уникальный код каждого мастера в системе нескольких устройств);
- неквитирование.

Арбитраж на шине происходит в момент передачи несколькими мастерами своих уникальных кодов. Выигравший арбитраж мастер захватывает шину. В связи с такой организацией режима HS дальнейший арбитраж и синхронизация на шине не реализуются.

После выполнения вышеуказанных шагов устройства, поддерживающие режим HS, переключаются в этот режим. Мастер генерирует состояние повторного старта (SR), а затем передает адрес ведомого и бит направления передачи R/W# (см. рисунок 17.15. Расшифровка обозначений, принятых на рисунке, приведена в таблице 17.1).

Все передачи в режиме HS по формату идентичны передачам режима FS, что делает эти два режима полностью совместимыми.

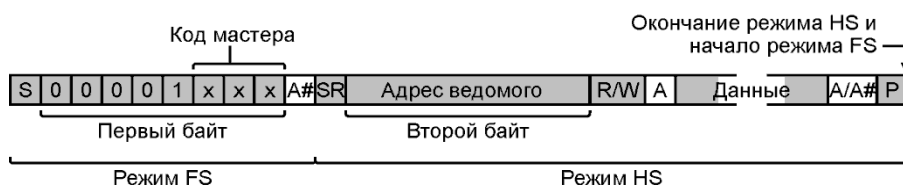


Рисунок 17.15 – Переход в режим HS и обратно в режим FS

Выход из режима HS происходит генерированием состояния окончания передачи (P), после которого все устройства переключаются обратно в режим FS. В каждом из двух базовых режимов – FS и HS – модуль I2C может функционировать как мастер или ведомый, получать или передавать информацию.

Далее все режимы работы модуля I2C будут рассмотрены подробно.

Инициализация

Для начала работы следует произвести инициализацию:

- 1 Включить модуль I2C установкой бита ENABLE в регистре CTL1.
- 2 Если активен режим мастера, записать нужный коэффициент деления в битовое поле SCLFRQ в регистре CTL1 для выбора периода тактового сигнала SCL (для режима HS записать коэффициент деления в поле HSDIV регистра CTL2).
- 3 Если активен режим ведомого, необходимо:
 - записать «собственный» адрес ведомого в битовое поле ADDR и установить бит SAEN регистра ADDR;
 - для реализации 10-битной адресации записать старшие биты адреса в битовое поле S10AD и установить бит S10EN регистра CTL2;
 - для включения функции распознавания адреса общего вызова установить бит GCMEN в регистре CTL0;
 - для включения функции распознавания адреса отклика установить бит SMBARE в регистре CTL0.
- 4 При необходимости отслеживания периодов ожидания на шине записать желаемые значения в регистр TOPR и в битовое поле TOCDIV (регистр CST) для отсчета времени ожидания на линии SCL. Для автоматического отслеживания времени ожидания записать ненулевое значение в битовое поле TOCDIV регистра CST.
- 5 Для разрешения формирования запроса на прерывание установить бит INTEN в регистре CTL0.

Функционирование

Модуль I2C может работать в режиме мастера или ведомого. Также он может функционировать как передатчик или приемник. Итого, модуль I2C поддерживает девять режимов:

- безадресный ведомый;
- мастер передатчик в режиме FS;
- мастер передатчик в режиме HS;
- мастер приемник в режиме FS;
- мастер приемник в режиме HS;
- ведомый передатчик в режиме FS;
- ведомый передатчик в режиме HS;
- ведомый приемник в режиме FS;
- ведомый приемник в режиме HS.

Передача информации по шине состоит из последовательности различных действий (начало передачи, прием данных и др.). Каждое действие называется состоянием (состояние старта, состояние останова и др.). После того, как то или иное состояние сформировано, его код аппаратно записывается в регистр ST в битовое поле MODE и может быть прочитано программно. В таблицах 17.2 и 17.3 приводятся все возможные состояния, их мнемонические обозначения и коды. На квитирование или неквитирование приема указывает запись «ACK» или «NACK», соответственно. Так, например, если мастер отправил байт адреса ведомому, который после получения квитировал прием, то на это будет указывать «ACK», а в поле MODE регистра ST будет записан код 04h, соответствующий состоянию с мнемоническим обозначением «MTADPA». Более подробно каждый режим работы модуля I2C будет рассмотрен далее. На рисунках 17.16 – 17.23, поясняющих работу модуля в том или ином режиме, приняты обозначения, расшифровка которых приводится в таблице 17.1. Для получения дополнительной информации и понимания работы модуля I2C можно воспользоваться приложением В.

Таблица 17.2 – Коды функционирования модуля I2C в режиме FS

| Режим | Код | Мнемоника | Описание состояния на момент записи кода в поле MODE регистра ST | ACK/NACK | |
|---------------------|-------------------------|-----------|--|---|------|
| Общий | 00h | IDLE | IDLE, нет доступной валидной информации о статусе | – | |
| Мастер в режиме FS | – | 01h | STDONE | Сформировано состояние старта | – |
| | | 02h | RSDONE | Сформировано состояние повторного старта | – |
| | | 03h | IDLARL | Потеря арбитража, переход в режим безадресного ведомого | – |
| | Передача | 04h | MTADPA | Отправлен адрес ведомого | ACK |
| | | 05h | MTADNA | Отправлен адрес ведомого | NACK |
| | | 06h | MTDAPA | Отправлен байт данных | ACK |
| | | 07h | MTDANA | Отправлен байт данных | NACK |
| | Прием | 08h | MRADPA | Отправлен адрес ведомого | ACK |
| | | 09h | MRADNA | Отправлен адрес ведомого | NACK |
| | | 0Ah | MRDAPA | Принят байт данных | ACK |
| | | 0Bh | MRDANA | Принят байт данных | NACK |
| – | 0Ch | MTMCER | Отправлен код мастера, обнаружена ошибка | ACK | |
| – | 0Dh – 0Fh | | Зарезервировано. Не использовать! | – | |
| Ведомый в режиме FS | Прием | 10h | SRADPA | Принят адрес | ACK |
| | | 11h | SRAAPA | Принят адрес после потери арбитража | ACK |
| | | 12h | SRDAPA | Принят байт данных | ACK |
| | | 13h | SRDANA | Принят байт данных | NACK |
| | Передача | 14h | STADPA | Принят адрес | ACK |
| | | 15h | STAAPA | Принят адрес после потери арбитража | ACK |
| | | 16h | STDAPA | Отправлен байт данных | ACK |
| | | 17h | STDANA | Отправлен байт данных | NACK |
| | Передача адреса отклика | 18h | SATADP | Принят адрес отклика на предупреждение | ACK |
| | | 19h | SATAAP | Принят адрес отклика на предупреждение после потери арбитража | ACK |
| | | 1Ah | SATDAP | Отправлены данные в ответ на получение адреса отклика | ACK |
| | | 1Bh | SATDAN | Отправлены данные в ответ на получение адреса отклика | NACK |

Окончание таблицы 17.2

| Режим | | Код | Мнемоника | Описание состояния на момент записи кода в поле MODE регистра ST | ACK/ NACK |
|--|---|-----|-----------|--|--------------|
| Ведомый в режиме FS | – | 1Ch | SSTOP | Обнаружено состояние останова ведомого | – |
| | | 1Dh | SGADPA | Принят адрес общего вызова | ACK |
| | | 1Eh | SDAAPA | Принят адрес общего вызова после потери арбитража | ACK |
| Общий | | 1Fh | BERROR | Обнаружена ошибка на шине (некорректное состояние старта или останова) | – |
| Примечание – Диапазон значений кодов 0Dh–0Fh зарезервирован и не доступен для использования. Дополнительная информация находится в приложении В. | | | | | |

Таблица 17.3 – Коды функционирования модуля I2C в режиме HS

| Режим | | Код | Мнемоника | Описание состояния на момент записи кода в поле MODE регистра ST | ACK/ NACK |
|--|----------|---------|--------------------|--|--------------|
| Мастер в режиме HS | – | 21h | HMTMCOK | Код мастера передан успешно, переход в режим HS | – |
| | | 22h | HRSDONE | Сформировано состояние повторного старта | – |
| | | 23h | HIDLARL | Потеря арбитража, переход в режим HS безадресного ведомого | – |
| | Передача | 24h | HMTADPA | Отправлен адрес ведомого | ACK |
| | | 25h | HMTADNA | Отправлен адрес ведомого | NACK |
| | | 26h | HMTDAPA | Отправлен байт данных | ACK |
| | | 27h | HMTDANA | Отправлен байт данных | NACK |
| | Прием | 28h | HMRADPA | Отправлен адрес ведомого | ACK |
| | | 29h | HMRADNA | Отправлен адрес ведомого | NACK |
| | | 2Ah | HMRDAPA | Принят байт данных | ACK |
| 2Bh | | HMRDANA | Принят байт данных | NACK | |
| Ведомый в режиме HS | Прием | 30h | HSRADPA | Принят адрес | ACK |
| | | 32h | HSRDAPA | Принят байт данных | ACK |
| | | 33h | HSRDANA | Принят байт данных | NACK |
| | Передача | 34h | HSTADPA | Принят адрес | ACK |
| | | 36h | HSTDAPA | Отправлен байт данных | ACK |
| | | 37h | HSTDANA | Отправлен байт данных | NACK |
| Примечание – Диапазоны значений кодов 2Ch–2Fh и 38h–3Fh, а также коды 20h, 31h, 35h зарезервированы и не доступны для использования. Дополнительная информация находится в приложении В. | | | | | |

Режим безадресного ведомого

Режим работы по умолчанию (MODE = 00h). После включения модуль I2C начинает функционировать в режиме безадресного ведомого и непрерывно мониторит шину. При обнаружении состояния старта или повторного старта переходит в режим ведомого приемника. Для перехода в режим мастера передатчика нужно сформировать корректное состояние старта.

Переключение в режим безадресного ведомого происходит в случаях:

- стартовое состояние не было успешно сформировано, так как другое устройство удерживало на линии SCL низкий уровень сигнала;

- произошла потеря арбитража во время передачи байта данных в режиме мастера передатчика или во время передачи бита R/W# в режиме мастера приемника;
- произошла потеря арбитража во время ответа на полученный адрес отклика;
- неквитирование принятого адреса в режиме ведомого приемника (адрес не совпал со «своим» или запрещен);
- неквитирование в конце переданного байта в режиме ведомого передатчика;
- обнаружено состояние останова;
- обнаружена ошибка на шине;
- модуль I2C был сброшен;
- модуль I2C был выключен.

Режим FS мастера передатчика

Включение режима:

1 Переход в режим мастера передатчика происходит после успешного формирования состояния старта. Первый байт, передаваемый мастером сразу после старта, состоит из адреса ведомого и бита направления.

2 В зависимости от состояния бита направления (R/W#), модуль I2C далее функционирует как мастер передатчик (если R/W# = «0») или как мастер приемник (если R/W# = «1»). Для перехода в режим HS мастер может передать код мастера (0000_1xxxh) вместо первого байта адреса.

3 Переход в режим мастера произойдет после установки бита START в регистре CTL0. Если бит BB в регистре CST сброшен, т. е. шина свободна, будет сгенерировано состояние старта. Если бит BB = 1b, то бит START останется установленным, а состояние старта будет сгенерировано по истечении времени, равного одному такту сигнала тактирования на линии SCL, после освобождения шины.

4 Как только стартовое состояние будет сгенерировано успешно, бит START сбросится, модуль I2C перейдет в состояние STDONE (в поле MODE запишется значение 01h), установится флаг INT, и линия SCL будет удерживаться в «0» до тех пор, пока флаг INT не будет сброшен. Если разрешено битом INTEN (регистр CTL0), сгенерируется прерывание.

Передача адреса и данных:

1 Пока удерживается флаг INT, программа записывает адрес ведомого и бит направления передачи в регистр данных SDA (адрес записывается в биты с седьмого по первый).

2 После записи в регистр SDA флаг INT сбрасывается программно установкой бита CLRST в регистре CTL0.

3 После сброса флага INT и по истечении времени, требуемого для установки данных, на линии SCL появляется тактовый сигнал и данные, хранящиеся в регистре SDA, начинают передаваться по линии SDA.

4 После завершения передачи байта и получения ответа на запрос подтверждения передачи (ACK), т. е. после девятого такта сигнала тактирования на линии SCL, аппаратная часть анализирует квитирование/неквитирование передачи и устанавливает соответствующий код в поле MODE.

5 Во время передачи линии SCL и SDA постоянно мониторятся с целью выявления возможных конфликтов с другими устройствами, подключенными к шине. В случае обнаружения конфликта передача прерывается, и в поле MODE записывается код 11b (состояние SRAAPA – переход в режим ведомого приемника после потери арбитража) или код 03h (состояние IDLARL – переход в режим безадресного ведомого после потери арбитража).

6 Если бит направления равен единице и не обнаружено ошибок на шине, модуль I2C переходит в режим мастера приемника.

7 Если бит направления равен нулю и передача адреса ведомого завершена успешно (значение кода в поле MODE не равно 05h/1Fh), устанавливается флаг INT, указывая на то, что ожидается запись первого байта данных в регистр SDA для дальнейшей передачи, и, если разрешено, генерируется прерывание. Пока флаг INT будет оставаться установленным, линия SCL будет удерживаться в «0».

8 Байт данных записывается программно в регистр SDA и передача продолжается.

9 Если ведомый приемник не квитирует отправленный ему байт данных, в поле MODE записывается код 0Bh (состояние MRDANA). На линии SCL будет установлен низкий уровень сигнала и, если разрешено, сгенерировано прерывание.

Для отслеживания ошибок в пакетах данных применяется механизм вычисления контрольной суммы (CRC) для нескольких байт данных. В режиме мастера передатчика установка бита PECNEXT в регистре CST вызовет перенос содержимого регистра ошибок (не доступен программно) в регистр SDA и инициирует передачу байта CRC (байт контрольной суммы) ведомому. Передача байта CRC должна выполняться после передачи последнего байта данных и перед формированием состояния останова или повторного старта.

Мастер передатчика контролирует шину и может адресовать любое ведомое устройство и изменять направление передачи без потери контроля над шиной, используя возможность формирования состояния повторного старта. Для формирования состояния следует:

1 Установить бит START.

2 В режиме мастера приемника прочитать последний полученный байт из регистра SDA.

3 Сбросить флаг прерывания INT.

После этих действий будет освобождена линия SCL, сгенерировано состояние повторного старта и сгенерировано прерывание. В поле MODE будет записан код 02h (состояние RSDONE).

Модуль I2C может быть выведен из режима мастера передатчика генерированием состояния останова. Для этого необходимо:

1 Установить бит STOP в регистре CTL0.

2 В режиме мастера приемника прочитать последний полученный байт из регистра SDA.

3 Сбросить флаг INT.

Вышеуказанные действия приведут к незамедлительному формированию состояния останова и очистке бита STOP.

Состояние останова может быть сформировано только, если модуль I2C функционирует как мастер и контролирует шину (в поле MODE находится любое значение кода из диапазона 01h – 0Bh).

Дополнительно можно обратиться к приложению В.

На рисунке 17.16 представлено графическое пояснение к описанию режима.

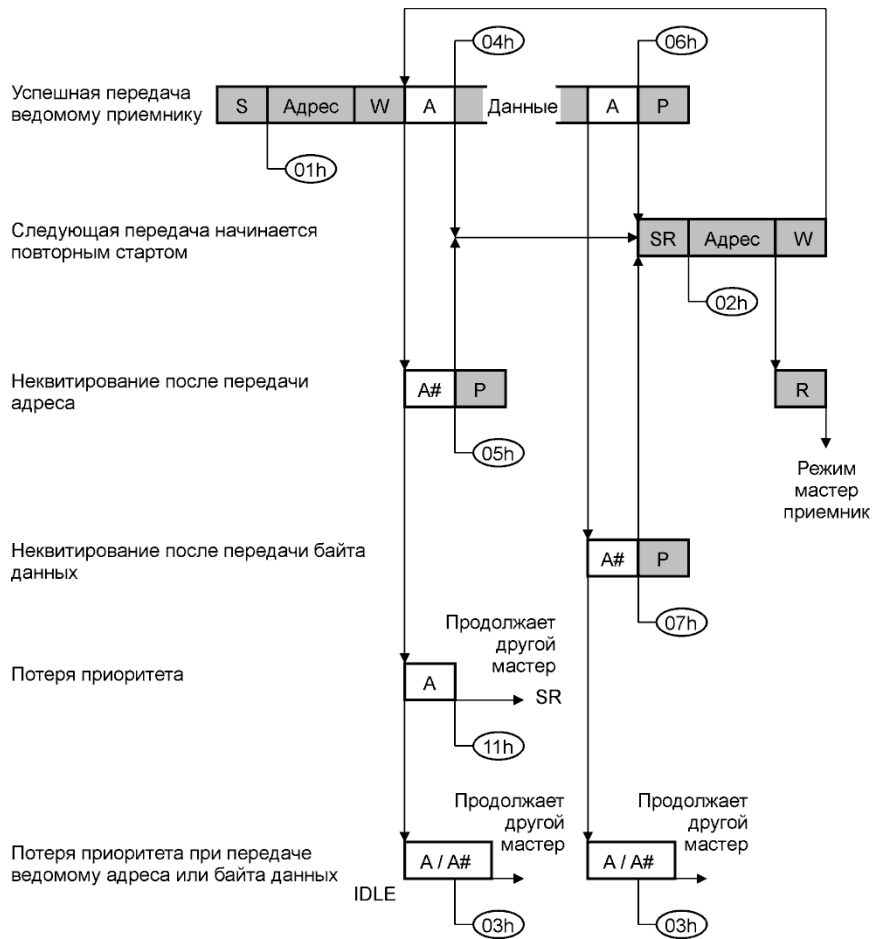


Рисунок 17.16 – Режим FS мастера передатчика

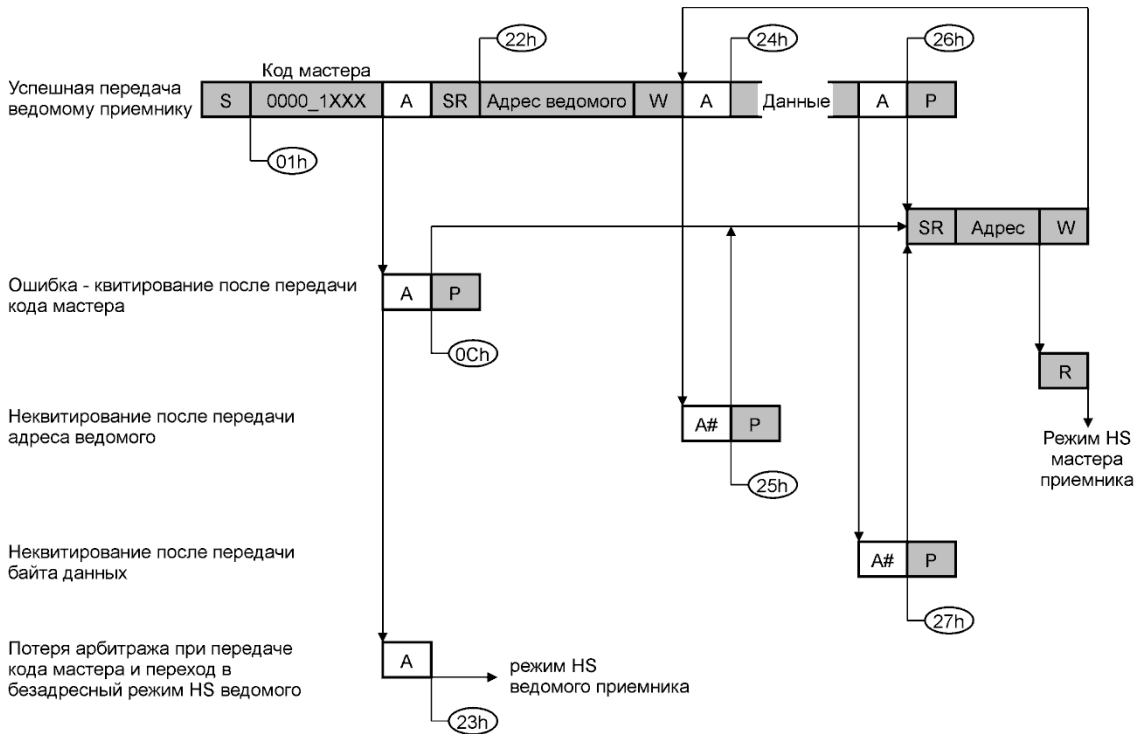


Рисунок 17.17 – Режим HS мастера передатчика

Режим HS мастера передатчика

Переход в режим HS мастера передатчика происходит в том случае, если после состояния старта мастер передает код мастера (0000_1xxx_b) вместо адреса ведомого. По окончании передачи кода мастера устанавливается флаг INT и, если разрешено, генерируется прерывание. Вслед за успешной передачей кода мастера в поле MODE записывается код 21h (состояние HMTMCOК), и мастер переходит в режим HS.

Далее необходимо сформировать состояние повторного старта, записав единицу в бит START и сбросить флаг INT, записью единицей в бит CLRST.

После сгенерированного состояния повторного старта устанавливается флаг INT и в поле MODE записывается код 22h (состояние HRSDONE). Дальнейший порядок действий по передаче адреса и данных аналогичен описанному режиму FS мастера передатчика.

Дополнительно можно обратиться к приложению В.

На рисунке 17.17 представлено графическое пояснение к описанию режима.

Режим FS мастера приемника

Переход в режим мастера приемника происходит после успешной передачи адреса ведомого с единичным битом направления (R/W# = «1»). В режиме мастера приемника модуль I2C получает данные от ведомого устройства, поэтому теряет контроль над шиной SDA. В тоже время мастер продолжает тактировать передачу и должен отвечать на бит АСК каждого принятого байта.

После каждого принятого байта устанавливается флаг INT, и пользовательская программа читает полученные данные из регистра SDA. Линия SCL удерживается в «0», пока установлен флаг INT. После сброса флага INT может стартовать прием следующего байта. После этого (согласно протоколу SMBus) состояния повторного старта или стопа не должны генерироваться мастером, поскольку мастер теперь не является единственным контролером линии SDA. В конце приема каждого байта мастер не квитирует прием, сообщая, таким образом, ведомому об успешном приеме.

После приема предпоследнего байта перед сбросом флага INT следует записать ноль в бит АСК регистра CTL0. В тоже время, если требуется отправка байта CRC, следует установить бит PECNEXT в регистре CST. После сброса флага INT будет принят последний байт данных и не квитирован. По окончании приема мастер возвращается в режим передатчика и теперь может сгенерировать состояние повторного старта или останова.

Если механизм отслеживания ошибок включен, то последний переданный от ведомого байт будет байтом CRC. В случае если результат вычисления контрольной суммы не нулевой, то установится флаг ошибки PECFAULT в регистре CST.

Дополнительно можно обратиться к приложению В.

На рисунке 17.18 представлено графическое пояснение к описанию режима.

Режим HS мастера приемника

Переход в режим HS мастера приемника происходит, если после переданного кода мастера и последовавшего за ним состоянием повторного старта, производится передача адреса ведомого с битом направления R/W# = «1». Модуль I2C переходит в режим HS мастера приемника, устанавливается флаг INT, а в поле MODE записывается соответствующий код из диапазона 28h – 2Bh.

Дополнительно можно обратиться к приложению В.

На рисунке 17.19 представлено графическое пояснение к описанию режима.

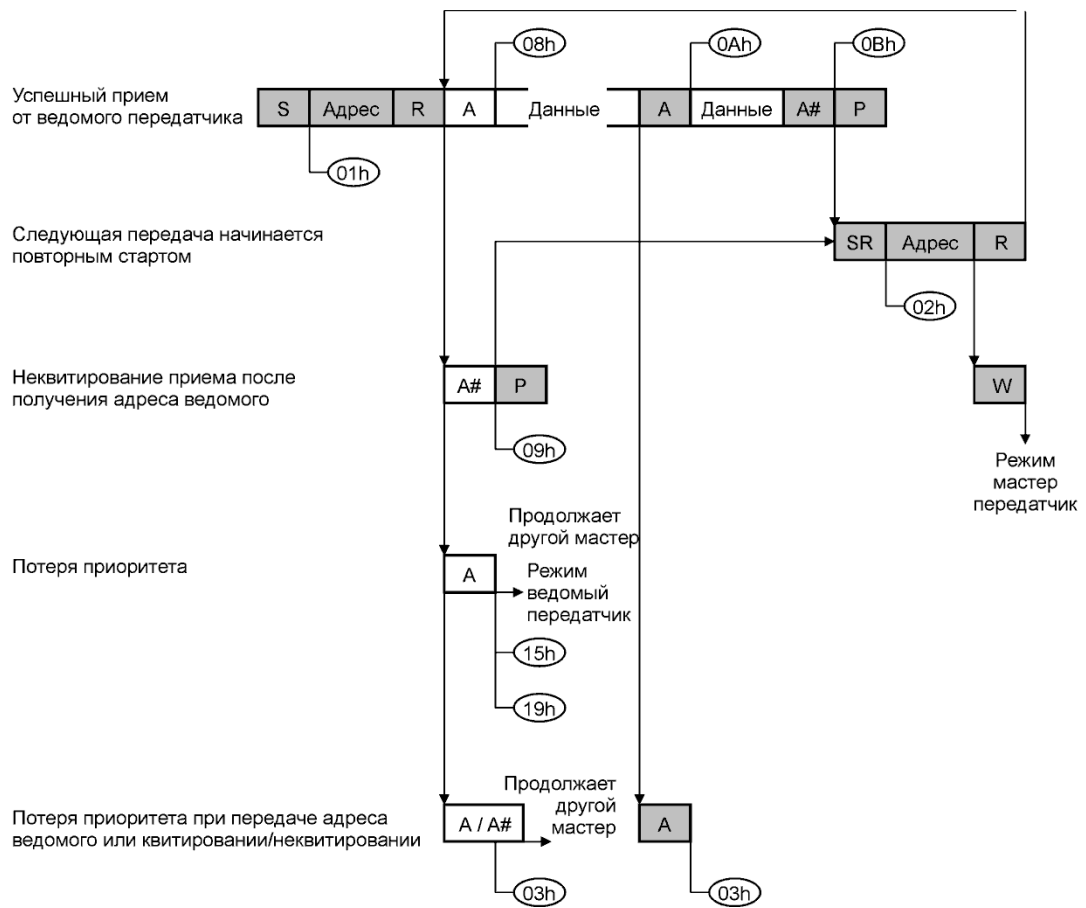


Рисунок 17.18 – Режим FS мастера приемника

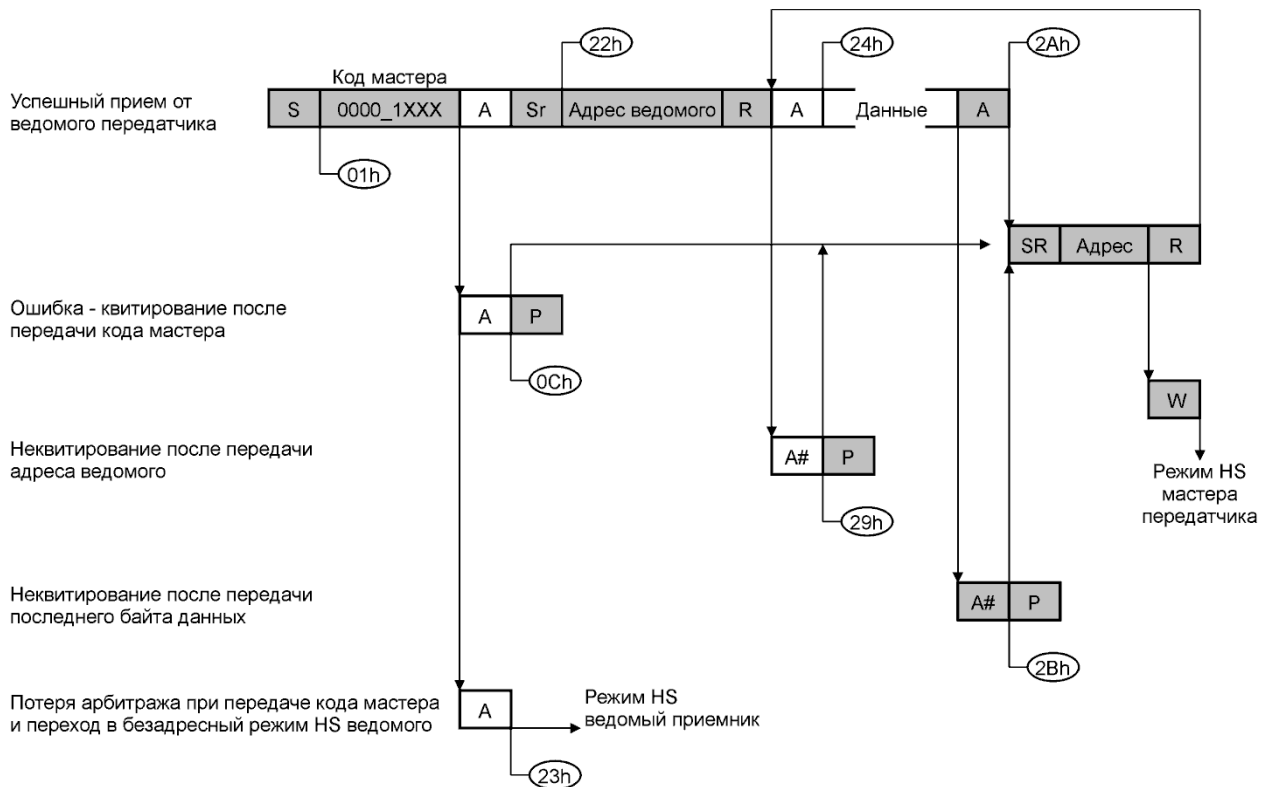


Рисунок 17.19 – Режим HS мастера приемника

Режим FS ведомого приемника

В этом режиме данные принимаются от мастера передатчика. Ведомый квитирует или не квитирует прием каждого байта.

После включения модуль I2C мониторит шину. При обнаружении состояния старта, модуль I2C переключается в режим ведомого приемника и начинает принимать семь бит адреса и бит направления передачи от мастера. Мастер передатчика может переключиться в режим ведомого приемника вследствие потери арбитража при передаче адреса.

После получения байта адреса ведомый сравнивает полученный адрес:

- по полю ADDR регистра ADDR, если установлен бит SAEN;
- со значением 0000_000b (адрес общего вызова), если установлен бит GCMEN;
- со значением 0001_100b (адрес отклика), если установлен бит SMBARE.

Квитирование приема производится, если принятый адрес совпал с «собственным» (запрограммированным пользователем), адресом общего вызова или адресом отклика. После обнаружения совпадения адреса и квитирования в поле MODE записывается соответствующий код и устанавливается флаг INT. Также, если разрешено битом INTEN, генерируется прерывание. Принятый байт (адрес и бит направления) переписывается в регистр SDA.

В зависимости от состояния бита направления, модуль I2C переходит в режим ведомого передатчика (если R/W# = «1») или остается в режиме ведомого приемника (R/W# = «0»).

После каждого принятого байта устанавливается флаг INT, указывающий на то, что необходимо прочитать данные из регистра SDA, а линия SCL удерживается в «0». После программного чтения регистра SDA флаг INT сбрасывается (записью единицы в бит CLRST), и линия SCL освобождается.

Установка битов SAEN и S10EN включает режим 10-битной адресации ведомого приемника. После обнаружения состояния старта ведомый последовательно принимает два байта, в которых содержится адрес.

Последовательность передачи бит в посылке при 10-битной адресации была рассмотрена ранее в подразделе 17.1 настоящего ТО.

Механизм распознавания адреса изложен в подразделе 17.2 настоящего ТО и показан на рисунке 17.13.

После корректного приема ведомым двух байтов и совпадении принятого адреса с собственным байты сохраняются в регистре SDA и сдвиговом регистре, прием квитируется, устанавливается флаг INT, а в поле MODE записывается соответствующий код состояния – 10h или 17h.

Если включен механизм обнаружения ошибок, последний байт, принятый от мастера передатчика, будет байтом CRC. Если результат вычисления контрольной суммы не нулевой, устанавливается флаг ошибки PECFAULT и передача не квитируется. Программа пользователя должна «знать» о количестве передаваемых мастером байт и устанавливать бит PECNEXT перед чтением предпоследнего байта из регистра SDA и потом сбрасывать флаг INT. В результате будет аппаратно рассчитана контрольная сумма, и результат отправлен мастеру в момент передачи бита ACK. Если ошибок нет, будет выполнено квитирование (отправлен «0» в ответ на запрос ACK), если ошибки есть – неквитирование (отправлена «1» в ответ на запрос ACK).

Если ведомому приемнику нужно сообщить мастеру, что он не может более принимать данные, следует сначала установить бит ACK, а затем – бит CLRST (для сброса флага INT). Далее будет принят последний байт данных, который не будет квитируван (бит ACK = 1b) и установится флаг INT. После этого программа может прочитать последний полученный байт из регистра SDA и сбросить флаг INT, после чего модуль I2C освободит шину.

Дополнительно можно обратиться к приложению В.

На рисунке 17.20 представлено графическое пояснение к описанию режима.

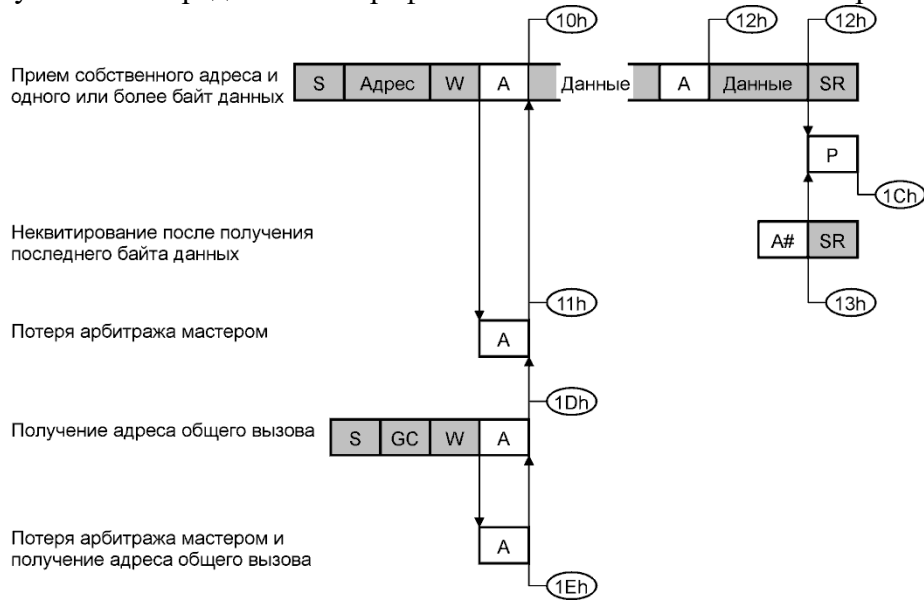


Рисунок 17.20 – Режим FS ведомого приемника



Рисунок 17.21 – Режим HS ведомого приемника

Режим HS ведомого приемника

Включение режима происходит после получения валидного кода мастера (0000_1xxxh). После передачи кода мастера формируется состояние повторного старта, а затем передается адрес ведомого с нулевым битом направления ($R/W\# = \langle 0 \rangle$). После получения байта адреса ведомый проверяет его на совпадение (см. ранее «Режим FS ведомого приемника»).

Дополнительно можно обратиться к приложению В.

На рисунке 17.21 представлено графическое пояснение к описанию режима.

Режим FS ведомого передатчика

В этом режиме данные передаются от ведомого передатчика к мастеру приемнику. Ведомый проверяет ответ мастера на бит ACK.

Переход в режим передатчика происходит из режима ведомого приемника. После получения собственного адреса и бита направления, равного единице ($R/W\# = \langle 1 \rangle$), ведомый становится передатчиком. Флаг INT устанавливается, указывая на то, что в регистр SDA следует записать данные.

Пока установлен флаг INT, линия SCL удерживается в «0». После записи данных в регистр SDA следует сбросить флаг INT. После этого, по истечении времени, необходимого для установки данных на линии SDA, линия SCL освобождается, и данные начинают передаваться.

Передача данных аналогична передаче в режиме мастера передатчика. После каждого успешного приема байта устанавливается флаг INT, а в поле MODE записывается соответствующий код. Линия SCL удерживается в состоянии «0» до тех пор, пока флаг INT остается установленным. Флаг INT должен сбрасываться только после записи данных в регистр SDA. Каждый последующий байт должен записываться в регистр SDA до тех пор, пока в поле MODE не появится код 17h (состояние STDANA), указывающий на то, что мастер «не желает» далее принимать данные.

Вывод ведомого из режима передатчика осуществляется только мастером приемника. Мастер приемника должен не квитировать последний (согласно запланированному количеству) полученный байт данных. При обнаружении неквитирования переданных данных, модуль I2C переходит в режим безадресного ведомого и в поле MODE записывается код 00h (состояние IDLE). Далее ведомый мониторит шину в ожидании состояния старта или повторного старта.

Для работы в режиме с 10-битной адресацией следует осуществить действия, аналогичные описанным для режима FS ведомого приемника.

Сначала модуль I2C переходит в режим ведомого приемника и получает 10-битный адрес. Если программно не требуется никаких действий, то флаг INT не устанавливается, линия SCL не удерживается в «0» и поле MODE содержит соответствующую информацию о состоянии. Далее (см. ранее «Формат передачи данных с 10-битной адресацией»), вслед за вторым байтом адреса может последовать состояние повторного старта и затем повторная передача первого байта адреса с той лишь разницей, что бит направления содержит единицу (R/W# = «1»). Таким образом, после приема трех байт, если принятый 10-битный адрес окажется «своим», установится флаг INT и ведомый переключится в режим передатчика. В поле MODE запишется один из двух кодов – 14h или 15h.

Если включен механизм распознавания ошибок, то последний отправленный ведомым передатчиком байт будет байтом CRC. Программа должна «знать» количество байт, посылаемых в пакете данных, и после отправки всех байт устанавливать бит PECNEXT (вместо записи очередных данных в регистр SDA) для того, чтобы в регистр SDA записался байт контрольной суммы.

В модуле I2C поддерживается функция распознавания адреса отклика, который передается мастером шины ко всем ведомым. Ведомое устройство, получившее адрес отклика (0001_100b), переключается в режим передатчика и начинает передавать свой собственный адрес (подробнее – см. подраздел 17.1 «Формат передачи данных с 7-битной адресацией»).

Для включения функции распознавания адреса отклика следует установить бит SMBARE в регистре CTL0.

Модуль I2C реагирует на адрес отклика только при работе в режиме ведомого. В ответ на получение адреса отклика начать передачу адресов могут несколько ведомых. Ведомый, выигравший арбитраж, продолжает передачу, остальные – освобождают шину.

Дополнительно можно обратиться к приложению В.

На рисунке 17.22 представлено графическое пояснение к описанию режима.

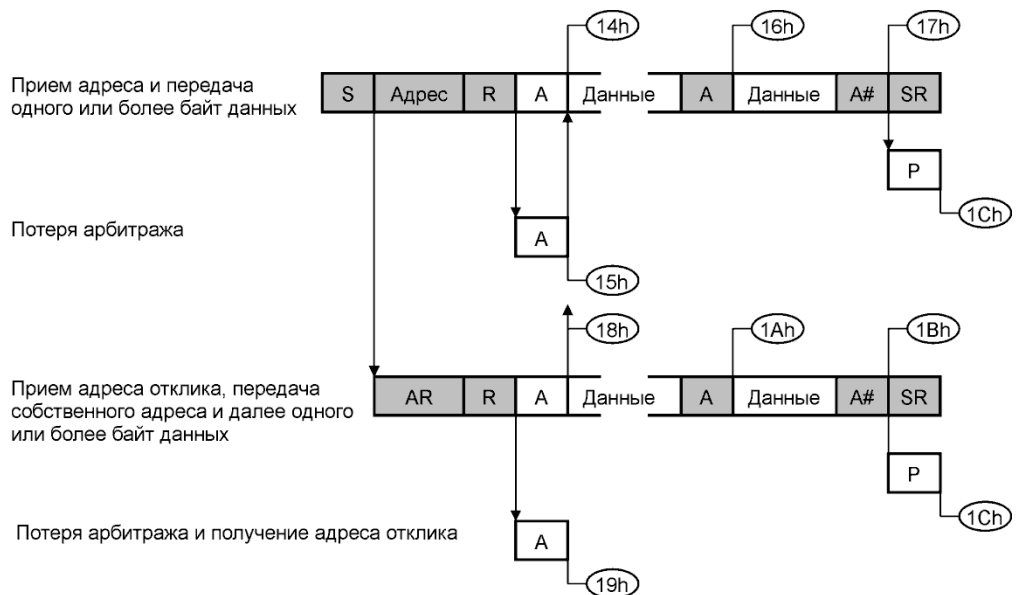


Рисунок 17.22 – Режим FS ведомого передатчика

Режим HS ведомого передатчика

Модуль I2C переходит в режим HS ведомого после получения валидного кода мастера (0000_1xxx). Далее следует состояние повторного старта и передача адреса ведомого с единичным битом направления ($R/W\# = \langle 1 \rangle$). После этого ведомый переключается в режим HS ведомого передатчика. Функционирование в этом режиме в целом идентично режиму FS ведомого передатчика, с теми отличиями, что поддерживается более высокая скорость передачи, а значения кодов состояний (поле MODE) находятся в диапазоне 34h – 37h.

Дополнительно можно обратиться к приложению В.

На рисунке 17.23 представлено графическое пояснение к описанию режима.



Рисунок 17.23 – Режим HS ведомого передатчика

Дополнительная информация о работе модуля

1 Когда модуль I2C выключен, бит ВВ регистра CST очищен. Включения модуля в системе с более чем одним мастером, может произойти в момент времени, когда по шине идет передача. Бит ВВ не сможет это показать. Во избежание создания ошибок на шине модуль I2C должен синхронизироваться с сигналами на шине прежде, чем сделать попытку стать мастером. Для этого следует дождаться момента, когда на шине не будет

выявлена активность, т. е. периодически проверять бит ВВ через периоды времени, равные периоду ожидания на шине.

2 Бит ВВ позволяет мониторить шину и не допускать формирования ошибочных состояний старта в процессе передачи между другими устройствами на шине.

3 В некоторых случаях шина может «зависать» при активных (с нулевым уровнем) сигналах на линиях SDA и/или SCL. Источниками таких состояний могут быть необнаруженные ошибочные стартовые или стоповые состояния, сформировавшиеся в течение приема ведомых данных. Если считать, что причиной зависания явился модуль I2C, то возможны следующие два варианта развития событий:

а) если зависла линия SCL, ничего не будет происходить, а мастер, захвативший шину, должен освободить ее;

б) если зависла линия SDA, мастер должен освободить шину. Следует помнить, что в нормальном состоянии удерживать линию SCL может только текущий мастер шины. Последовательность действий для выхода из зависания следующая (при условии, что на шине только один мастер):

- выключить и включить модуль I2C для перевода его в режим безадресного ведомого;

- установить бит START для создания состояния старта;

- проверить, удерживается ли линия SDA в «0» (активное состояние) чтением бита TSDA регистра CST. Если линия активна, отправить одиночный импульс по линии SCL, установив бит TGSCS в регистре CST;

- проверить, что в поле MODE записан код 01b (состояние STDONE), который укажет на то, что состояние старта сформировано. Если нет, то повторять предыдущий и этот шаги до тех пор, пока линия SDA не освободится.

18 Контроллер интерфейса CAN

18.1 Протокол CAN

Последовательный интерфейс CAN (Controller Area Network) – интерфейс связи, эффективно поддерживающий распределенное управление в масштабе реального времени с высокой помехозащищенностью. Протокол связи определен в спецификации CAN 2.0B.

Протокол CAN оптимизирован для систем, в которых должно передаваться относительно небольшое количество информации (по сравнению с Ethernet или USB) к любому или всем узлам сети. Множественный доступ с опросом состояния шины позволяет каждому узлу получить доступ к шине с учетом приоритетов. Неадресная структура сообщений позволяет организовать многоабонентскую доставку данных с сокращением трафика шины. Быстрая устойчивая передача информации с системой контроля ошибок позволяет отключать неисправные узлы от шины, что гарантирует доставку критических по времени сообщений.

Область применения протокола CAN: от высокоскоростных сетей связи до электропроводов в автомобиле. Высокая скорость передачи данных (до 1 Мбит/с), хорошая помехозащищенность протокола, защита от неисправности узлов – делают шину CAN подходящей для промышленных приложений управления типа Device Net.

CAN имеет асинхронную последовательную структуру шины с одним логическим сегментом сети. CAN сеть может состоять из двух или более узлов с возможностью подключения/отключения узлов от шины без перенастройки других устройств (см. рисунок 18.1).

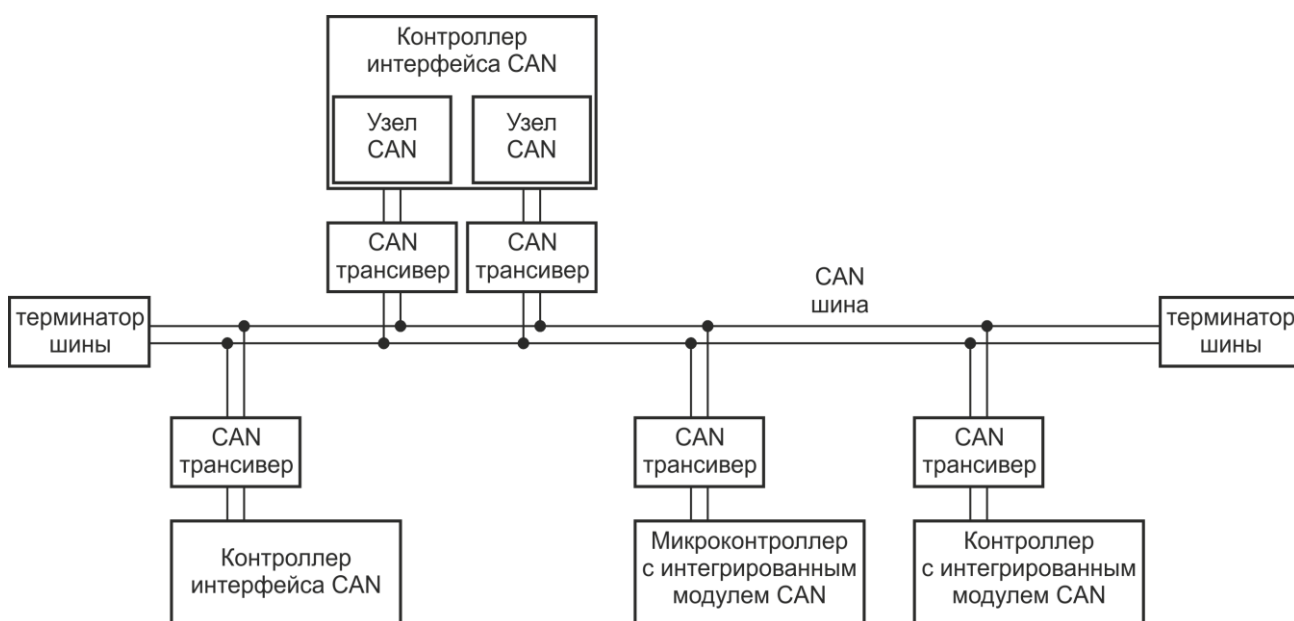


Рисунок 18.1 – Общая структура CAN сети

Логика шины работает по механизму монтажного И, в котором рецессивный бит соответствует логической единице, а доминантный – логическому нулю. Пока ни один узел не формирует доминантный бит, шина находится в рецессивном состоянии. Появление на шине доминантного бита (выставленного одним или несколькими узлами) создает доминантное состояние шины. Отсюда следует, что при выборе среды передачи данных необходимо точно определить, какое состояние будет доминантным, а какое – рецессивным. Одним из наиболее распространенных и дешевых вариантов линии связи является пара скрученных проводов. Линии шины тогда называются CANH и CANL и

могут быть подключены непосредственно к устройствам. Не существует никакого дополнительного стандарта на среду передачи данных.

При использовании в качестве линии связи пары скрученных проводов с нагрузочными резисторами на концах можно получить максимальную скорость передачи данных 1 Мбит/с при длине линии до 40 м. Для линий связи протяженностью более 40 м необходимо снизить скорость передачи данных (для линии 1 000 м скорость шины должна быть не более 40 Кбит/с). Из-за дифференциального характера линии связи шина CAN малочувствительна к электромагнитным помехам. Экранирование шины значительно снизит воздействие внешнего электромагнитного поля, что особенно важно для высокоскоростных режимов работы.

Двоичная информация кодируется. Доминантным является низкий уровень, рецессивным – высокий. Для гарантированной синхронизации данных всеми узлами шины используется принцип «бит-стаффинга». Это означает, что при последовательной передаче пяти бит одинаковой полярности передатчик вставляет один дополнительный бит противоположной полярности перед передачей остальных битов. Приемник также проверяет полярность и удаляет дополнительные биты.

В CAN протоколе при передаче данных приемные узлы не адресуются, а указывается идентификатор передатчика. С помощью идентификатора указывается содержание сообщения (например, применительно автомобиля – обороты, температура двигателя и т. д.) и степень приоритета сообщения. Более высокий приоритет у идентификатора, имеющего меньшее бинарное значение.

При коллективном доступе к шине используется неразрушающий арбитраж с опросом состояния шины. Перед началом передачи данных узел проверяет состояние шины (отсутствие активности на шине). При начале передачи сообщения узел становится управляющим шины, все остальные узлы переходят в режим приема. После приема сообщения (подтвержденного каждым узлом) каждый узел проверяет идентификатор в сообщении и сохраняет сообщение, если это требуется. В противном случае, сообщение сбрасывается. Если два или более узлов начинают передачу данных одновременно, поразрядный арбитраж позволяет избежать конфликта на шине. Каждый узел выдает на шину свой идентификатор (старший бит формируется первым) и контролирует ее состояние. Если узел посылает «1», а читает «0», значит, арбитраж потерян, и узел переключается в режим приема. Это происходит тогда, когда идентификатор конкурирующего узла имеет меньшее бинарное значение. Таким образом, узел с высоким приоритетом выигрывает арбитраж без необходимости повторять сообщение. Все остальные узлы будут пытаться передать сообщение после освобождения шины. Данный механизм не позволяет передавать сообщения одновременно разными узлами. Для этого программно должно быть обеспечено, чтобы узлы, передающие данные, не имели одинаковых идентификаторов. Оригинальная спецификация в версии CAN 2.0b (так называемая расширенная версия CAN) определяет возможность идентификатора иметь длину 11 или 29 бит.

Протокол CAN предусматривает следующие типы сообщений:

- сообщение данных (стандартное и расширенное);
- удаленный запрос данных;
- сообщение об ошибке;
- сообщение о перезагрузке.

Стандартное сообщение данных

Формируется, когда узел желает передать данные. Формат сообщения показан на рисунке 18.2.

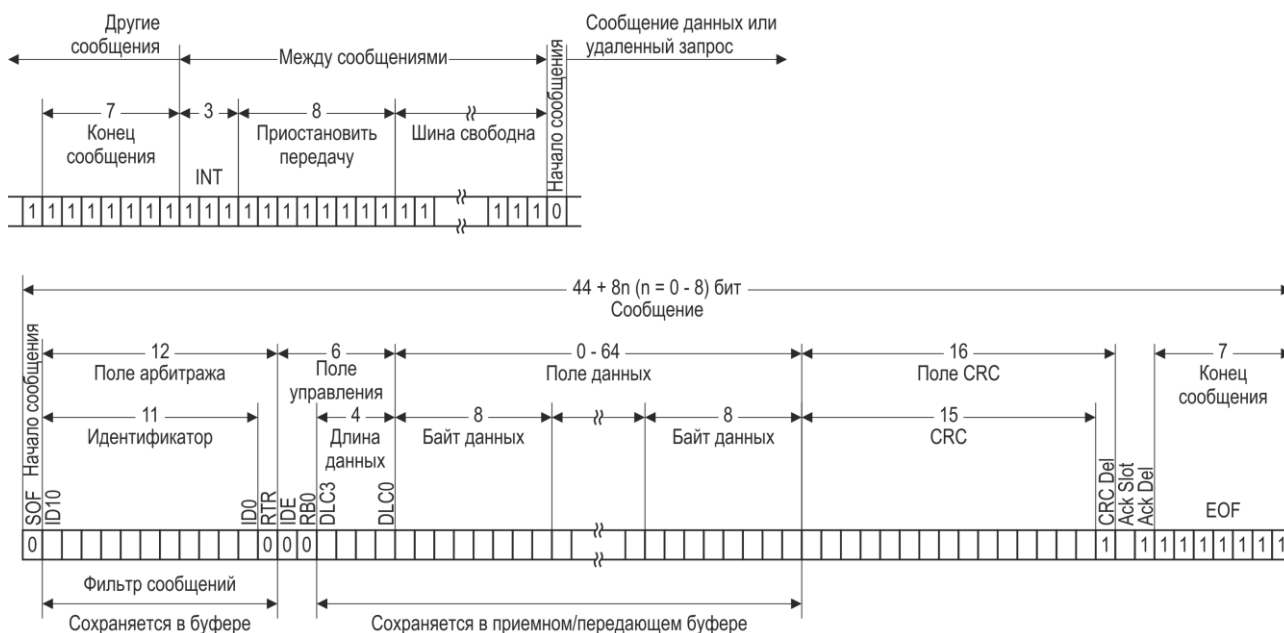


Рисунок 18.2 – Стандартное сообщение данных

Стандартное сообщение имеет в своем составе:

- бит SOF – доминантный («0») бит начала сообщения для жесткой синхронизации всех узлов;

- поле арбитража (12 бит), включающее поле ID идентификатора (11 бит) и бит RTR передачи по удаленному запросу (RTR = «0» соответствует сообщению данных, RTR = «1» соответствует удаленному запросу);

- поле управления (6 бит), включающее бит IDE – указатель расширенного идентификатора (IDE = «0» соответствует стандартному идентификатору, IDE = «1» соответствует расширенному идентификатору), бит RBU – резервный доминантный бит и поле DLC – числа байт данных (4 бита), которое указывает, сколько байт данных содержится в сообщении (допустимые значения – от 0 до 8, другие значения использоваться не могут);

- поле данных (от 0 до 64 бит), содержащее целое число байт данных;

- поле контрольной суммы CRC (16 бит), включающее поле CRC (15 бит), используемое для обнаружения возможных ошибок передачи данных и бит CRC Del рецессивный разделитель CRC;

- поле подтверждения (2 бита), включающее бит ACK Slot подтверждения передачи (передающий узел выдает рецессивный бит, а любой узел, который принял сообщение без ошибок, заменяет его сформированным доминантным битом) и бит ACK Del рецессивный разделитель подтверждения;

- поле EOF конца сообщения (7 бит).

Между передачами двух любых сообщений шина должна оставаться в рецессивном состоянии как минимум в течение времени появления 3 бит (поле INT простоя). Если после появления трех рецессивных битов (поле INT) ни один узел не начал передачу, шина переходит в состояние бездействия IDLE и находится в рецессивном состоянии до появления доминантного бита сообщения.

Расширенное сообщение данных

Формируется, когда узел желает передать данные. Формат сообщения показан на рисунке 18.3.

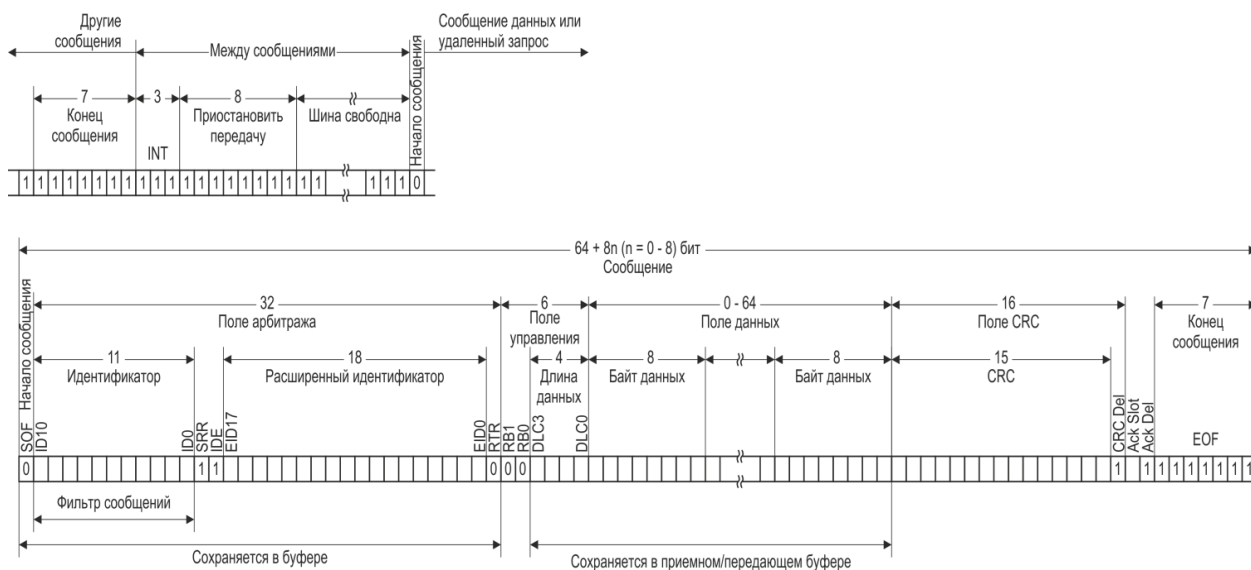


Рисунок 18.3 – Расширенное сообщение данных

Расширенное сообщение имеет в своем составе:

- бит SOF – доминантный («0») бит начала сообщения для жесткой синхронизации всех узлов;
- поле арбитража (38 бит), включающее поле стандартного идентификатора (11 бит), бит SRR – заменитель удаленного запроса, бит IDE – указатель расширенного идентификатора (рецессивный, что соответствует расширенному идентификатору) и поле расширенного идентификатора (18 бит);
- бит RTR – передачи по удаленному запросу (RTR = «0» соответствует сообщению данных, RTR = «1» соответствует удаленному запросу);
- поле управления (6 бит), включающее бит RB0 – резервный доминантный бит, бит RB1 – резервный доминантный бит и поле DLC – числа байт данных (4 бита), которое указывает, сколько байт данных содержится в сообщении (допустимые значения – от 0 до 8, другие значения использоваться не могут);
- поле данных (от 0 до 64 бит), содержащее целое число байт данных;
- поле контрольной суммы CRC (16 бит), включающее поле CRC (15 бит) – используемое для обнаружения возможных ошибок передачи данных и бит CRCDel – рецессивный разделитель CRC;
- поле подтверждения (2 бита), включающее бит ACK Slot – подтверждения передачи (передающий узел выдает рецессивный бит, а любой узел, который принял сообщение без ошибок, заменяет его сформированным доминантным битом) и бит ACKDel – рецессивный разделитель подтверждения;
- поле EOF – конца сообщения (7 бит).

Удаленный запрос данных

Формируется, когда узлу требуются данные другого узла. Узел назначения посылает удаленный запрос с идентификатором источника. Соответствующий узел источника (распознавший свой идентификатор) посылает стандартное или расширенное сообщение в ответ на запрос.

Удаленный запрос данных существует в стандартном и расширенном вариантах (на рисунке 18.4 представлен вариант удаленного запроса со стандартным идентификатором).

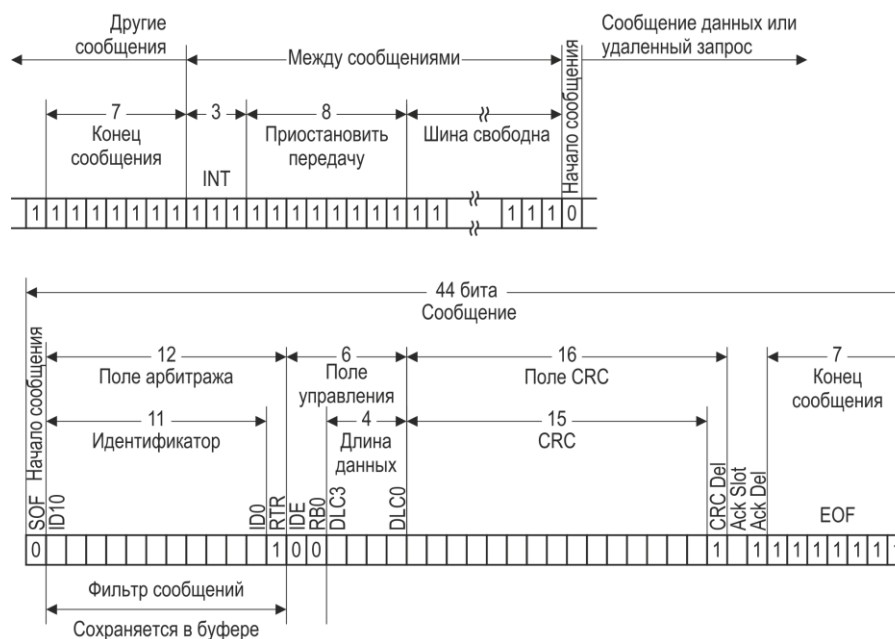


Рисунок 18.4 – Удаленный запрос данных (стандартный формат)

Имеются только два отличия содержимого удаленного запроса от сообщения данных:

- бит RTR в удаленном запросе передается в рецессивном состоянии;
- поле данных отсутствует (в сообщении не передается никаких данных, значение в поле DLC любое в пределах от 0 до 8).

В самом маловероятном случае, когда одновременно формируется удаленный запрос, и устройство пытается передать данные с одинаковыми идентификаторами, арбитраж будет выигран устройством, передающим данные, из-за доминантного состояния бита RTR.

Узел, который посылал запрос, получает данные немедленно.

Сообщение об ошибке

Формируется любым узлом, который обнаруживает ошибку на шине. Формат сообщения показан на рисунке 18.5.

Сообщение об ошибке состоит из двух полей: поле разделителя ошибки и поле флага ошибки. Возможны два типа поля флага ошибки, в зависимости от вида ошибки узла, обнаружившего ее.

Если ошибку обнаружил активный узел (как в примере на рисунке 18.5), тогда он прерывает передачу текущего сообщения, формируя флаг активной ошибки. Флаг активной ошибки состоит из шести последовательных доминантных битов, которые нарушают правила бит-стаффинга (правила заполнения и передачи битов на шине). Остальные узлы также обнаруживают ошибку и начинают формировать сообщение об ошибке. Таким образом, поле флага ошибки может содержать от 6 до 12 доминантных битов (сформированных одним узлом или более). Поле флага ошибки дополняется разделителем ошибки, состоящим из восьми рецессивных битов и позволяющим перезапустить связь с шиной после обнаружения ошибки. После перехода шины в нормальное состояние узлы возобновляют передачу данных, остановленный узел повторяет передачу сообщения, переданного до этого с ошибкой.

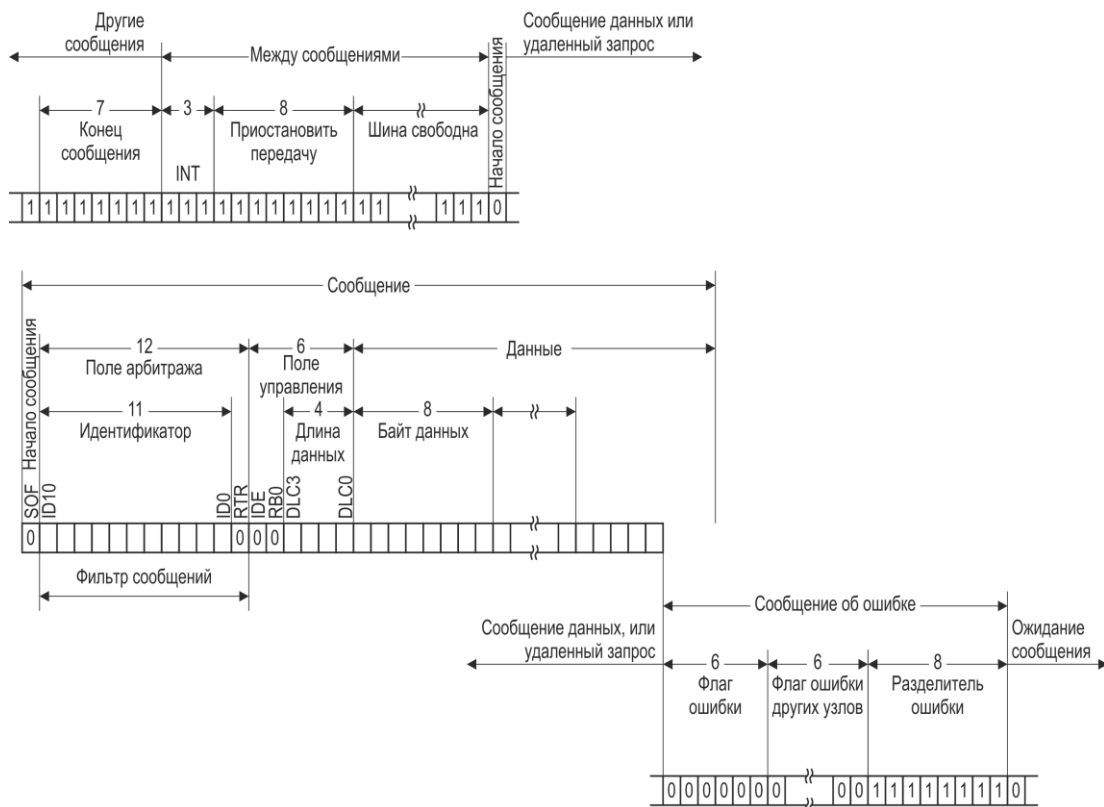


Рисунок 18.5 – Сообщение об ошибке

Если ошибку обнаружил пассивный узел, тогда он формирует флаг пассивной ошибки, состоящий из шести последовательных рецессивных битов и затем разделитель ошибки. Таким образом, сообщение о пассивной ошибке состоит из 14 рецессивных битов. Это не нарушает правила бит-стаффинга на шине и не оказывает влияния на передачи других узлов. Исключение составляет узел, который передает данные узлу, обнаружившему ошибку. В этом случае правила бит-стаффинга нарушаются и передача данных прекращается. После передачи пассивной ошибки узел должен ожидать шесть последовательных рецессивных битов для восстановления связи с шиной.

Сообщение о перезагрузке

Формат сообщения о перезагрузке аналогичен формату сообщения об ошибке, но может быть сформирован только, когда шина простаивает.

Сообщение о перезагрузке показано на рисунке 18.6.



Рисунок 18.6 – Сообщение о перезагрузке

Разделитель перезагрузки состоит из восьми последовательных рецессивных битов.

Узел может сформировать сообщение о перезагрузке в двух случаях:

- между сообщениями обнаружен доминантный бит, что является ненормальным во время простоя шины;

- для задержки передачи нового сообщения.

Узел может последовательно сформировать не более двух сообщений перезагрузки.

Флаг перезагрузки состоит из шести последовательных доминантных битов. Другие узлы обнаруживают перезагрузку и начинают формировать ее самостоятельно. Поэтому на шине во время выполнения перезагрузки может быть до 12 доминантных битов.

18.2 Структура и функционирование контроллера CAN

В состав контроллера CAN входят два идентичных независимых узла CAN0 и CAN1, ОЗУ для хранения сообщений, которое является общим для узлов, и система управления. Контроллер CAN имеет следующие функциональные особенности:

- соответствие ISO 11898;
- функционирование согласно спецификации CAN 2.0b (активная версия);
- отдельные управляющие регистры для каждого из двух узлов;
- программируемая скорость передачи информации до 1 Мбит/с;
- гибкий и полный контроль передачи сообщений и обработки ошибок.

Контроллер CAN реализует 16 линий прерываний и 256 объектов сообщений для хранения сообщений и их параметров в ОЗУ. Каждый объект сообщения может быть привязан к любому из узлов, сконфигурирован для передачи или приема как стандартных, так и расширенных сообщений и удаленных запросов. Каждый объект имеет индивидуальную маску для фильтрации принимаемых сообщений. Объекты сообщений могут объединяться в классы, с разными уровнями приоритета, могут объединяться для построения структур FIFO произвольных размеров (до 256 объектов в одной структуре). Кроме того, реализована возможность попарного соединения объектов для формирования шлюзов для автоматической передачи сообщений между узлами. Параллельно с вышеуказанными свойствами объекты сообщений могут организовываться в списки с постоянно доступной реорганизацией (совместимость с TwinCan-устройствами, которые не имеют списков).

Структура контроллера CAN приведена на рисунке 18.7.

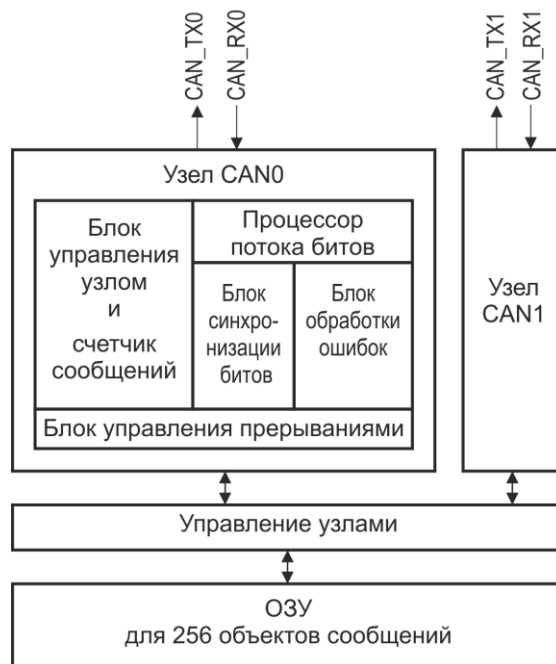


Рисунок 18.7 – Общая структура контроллера CAN

Синхронизация

Тактирующим сигналом контроллера CAN является сигнал Fclc (Fin), приходящий с генератора тактовых сигналов. На основе этого сигнала посредством программируемого дробного делителя частоты формируется внутренний сигнал Fcan (Fout), синхронизирующий работу контроллера и являющийся базовым синхросигналом для передачи/приема сообщений по внешней шине CAN.

Включение контроллера CAN

По умолчанию, после сброса микроконтроллера контроллер CAN выключен. На это также указывает состояние флага DISS регистра CLC. Когда контроллер выключен, этот флаг установлен.

Для включения контроллера CAN следует записать ноль в бит DISR регистра CLC. После этого флаг DISS сбросится. Рекомендуется проверять состояние флага DISS, перед началом программирования регистров контроллера, которые не доступны в выключенном состоянии.

Выключение контроллера CAN

Программно можно перевести контроллер CAN в режим выключения установкой бита DISR. Контроллер завершает все текущие операции, после чего устанавливает флаг DISS и отключает внутреннее тактирование, в связи с чем, все регистры становятся недоступными для обращения.

Простой шины

Между передачами сообщений шина CAN находится в рецессивном состоянии. Для выполнения условий простой шины необходимо, чтобы было получено, как минимум, три рецессивных бита после завершения передачи/приема очередного сообщения.

Анализ работы контроллера CAN

Для анализа работы контроллера доступны два режима – общего анализа и внутренней петли.

Режим общего анализа включается установкой бита CALM регистра NCR узла и позволяет осуществлять независимый мониторинг работы узла, не затрагивая шину CAN. В этом режиме сообщения данных и удаленные запросы отслеживаются без участия узла в операциях на шине. Выходы узла находятся в рецессивном состоянии. Узел может получать сообщения данных, сообщения удаленных запросов и сообщения об ошибках, но работа узла на передачу запрещена. Полученные сообщения данных/удаленных запросов остаются без подтверждения (бит подтверждения остается в рецессивном состоянии), но принимаются и сохраняются (при совпадении идентификаторов) в соответствующих объектах сообщений. В ответ на входящие сообщения не выдается подтверждение, и не генерируются сообщения об ошибках. На удаленные запросы не выдаются сообщения данных, а сами сообщения данных не могут быть переданы установкой бита запроса передачи TXRQ регистра состояния объекта сообщения MOSTAT. Прерывания после приема генерируются (если это разрешено) для всех принятых сообщений, не содержащих ошибок.

Режим внутренней петли включается установкой бита LBM регистра NPCR и позволяет проводить внутреннее тестирование контроллера CAN, а также отладку управляющей программы без доступа к внешней шине CAN. Внутренняя петля состоит из внутренней шины CAN (внутри контроллера CAN) и переключателя выбора шины для каждого узла (см. рисунок 18.8). С помощью переключателя каждый узел CAN может быть подключен либо к внутренней шине (режим внутренней петли), либо к внешней шине (нормальный режим работы). Если выбран режим внутренней петли, то на внешнем

передающем выводе узла CAN поддерживается рецессивный уровень сигнала, а состояние принимающего вывода игнорируется.

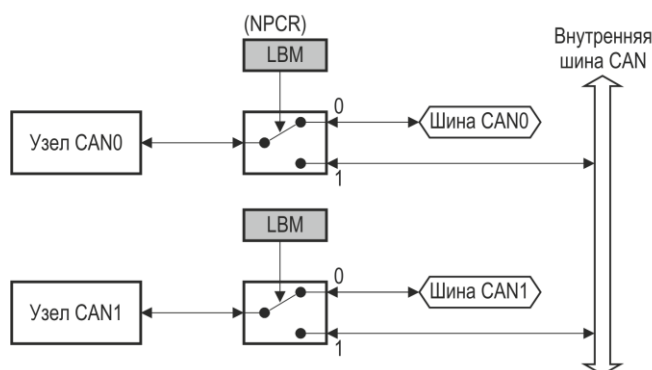


Рисунок 18.8 – Режим внутренней петли

Если оба узла CAN функционируют в режиме внутренней петли, они взаимодействуют друг с другом посредством внутренней шины CAN, не оказывая влияние на работу других модулей, функционирующих в нормальном режиме.

Дробный делитель

Дробный делитель позволяет генерировать частоту f_{out} из входной тактовой частоты f_{in} (SysCLK) путем программирования делителя посредством регистра FDR.

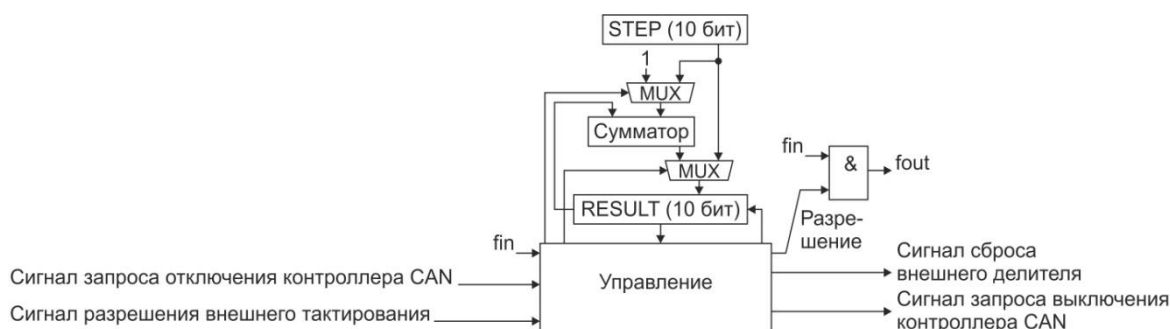


Рисунок 18.9 – Схема дробного делителя

Задаваемое значение входной частоты f_{in} зависит от длительности передачи одного бита информации и должно быть n -кратно ей. Поскольку длительность передачи бита определяется количеством квантов времени (Nt_q , см. далее), то для расчета частоты f_{in} в МГц следует пользоваться формулой:

$$f_{in} = n \times Nt_q, \quad (18.1)$$

где Nt_q – количество квантов времени t_q ;
 n – целое число, начиная с 1 (для задания кратности).

Дробный делитель делит частоту f_{in} путем умножения на величину $1/val$ или величину $1024/val$ для любого val от 0 до 1023, выдавая на выходе тактовый сигнал f_{out} (f_{can}).

На рисунке 18.9 показана блок-схема дробного делителя. Логика дробного делителя работает по-разному, в зависимости от режима, задаваемого полем DM.

В режиме нормального деления ($DM = 01b$) делитель работает как перегружаемый счетчик с шагом инкрементирования, равным единице. Состояние счетчика доступно посредством поля RESULT. Каждый раз, при переполнении (т. е. когда $RESULT = 3FFh$), формируется импульс сигнала Fout, после чего в счетчик загружается значение из поля STEP.

Выходная частота f_{out} определяется по формуле

$$f_{out} = f_{in} \times 1 / (1024 - STEPd), \quad (18.2)$$

где STEPd – значение поля STEP в десятичном формате.

Отсюда следует, что для получения частоты $f_{out} = f_{in}$, значение STEP должно быть равно $3FFh$. На рисунке 18.10 показано формирование сигнала Fout при значении $STEP = 3FDh$ ($1021d$).

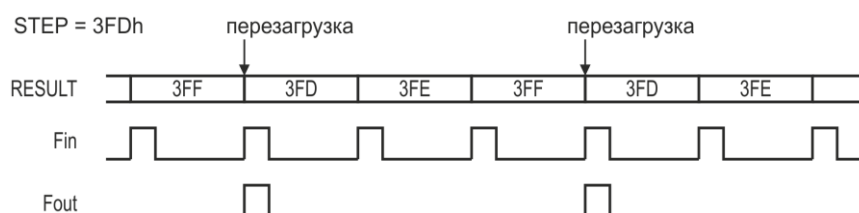


Рисунок 18.10 – Формирование сигнала с частотой f_{out} в нормальном режиме

В режиме дробного деления ($DM = 10b$) делитель работает как перезагружаемый счетчик, но шаг инкрементирования в этом случае равен значению поля STEP. Если результат инкрементирования значения RESULT на величину STEP превышает $3FFh$, возникает переполнение счетчика, формируется импульс сигнала Fout, после чего в счетчик загружается значение, на которое результат инкрементирования превысил $3FFh$.

Выходная частота f_{out} определяется по формуле

$$f_{out} = f_{in} \times STEPd / 1024d . \quad (18.3)$$

В целом, режим дробного деления позволяет программировать частоту f_{out} с более высокой точностью, чем нормальный режим, но сигнал может иметь джиттер периода, не превышающий одного периода f_{in} , в связи с чем, не рекомендуется использовать режим дробного деления при высоких скоростях передач.

На рисунке 18.11 показано формирование сигнала Fout при значении $STEP = 234h$ ($564d$). $f_{out} = f_{in} \times 564 / 1024 = 0,55 \times f_{in}$

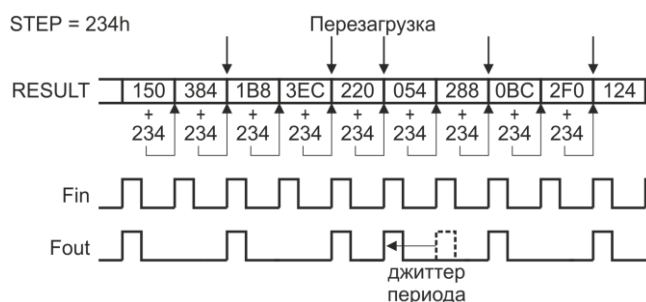


Рисунок 18.11 – Формирование сигнала с частотой f_{out} в режиме дробного деления

Процесс выключения делителя начинается одновременно с возникновением запроса выключения контроллера CAN.

Контроллер сообщений

Управляет обменом сообщениями между CAN узлами и памятью сообщений и выполняет следующие функции:

- фильтрация входящих сообщений для определения корректного объекта сообщения для сохранения полученных данных;
- определение объекта сообщения, содержимое которого будет передано в первую очередь (для каждого узла индивидуально);
- передача содержимого объекта сообщения к CAN узлу с параллельной вставкой в сообщение битов управления и состояния;
- осуществление буферизации FIFO и функционирования шлюза;
- объединение битов уведомления ждущих обработки сообщений.

Управление прерываниями блока CAN

На рисунке 18.12 показана структура формирования запроса на прерывание.

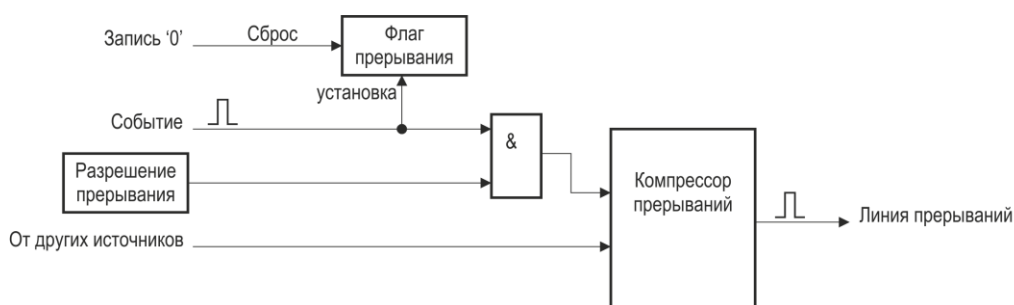


Рисунок 18.12 – Структура формирования запроса на прерывание

Событие, по которому должен быть сгенерирован запрос на прерывание, устанавливает флаг прерывания и (если разрешено) формирует запрос на прерывание на одной из 16 линий прерываний. Импульс запроса на прерывание генерируется независимо от состояния флага прерывания. Флаг прерывания может быть сброшен программно, записью нуля. Если к одной линии прерываний подключены несколько источников прерываний, то появление импульса от любого источника сформирует запрос на прерывание. Логика управления прерываниями использует схему компрессии прерываний.

Источниками прерываний являются:

- CAN узлы (восемь источников – по четыре для каждого узла);
- объекты сообщений (512 источников – по два для каждого объекта);
- программное прерывание (источник – регистр MITR).

Каждый аппаратный источник прерывания управляется 4 битами указателя прерываний, который определяет для него одну из 16 линий прерываний, что позволяет коммутировать на одну линию несколько источников прерываний. На рисунке 18.13 представлена схема коммутации линий прерываний.

Когда объект сообщения *n* генерирует запрос на прерывание по окончании приема или передачи сообщения, запрос передается на линию прерываний, выбранную в битовом поле RXINP или TXINP регистра MOIPR объекта сообщения *n*. Если количество объектов сообщений больше, чем количество линий прерываний, то на одну линию могут приходиться несколько запросов прерываний. Для разрешения конфликтов на линиях прерываний в контроллере CAN предусмотрен механизм распределения приоритетов для объектов сообщений.

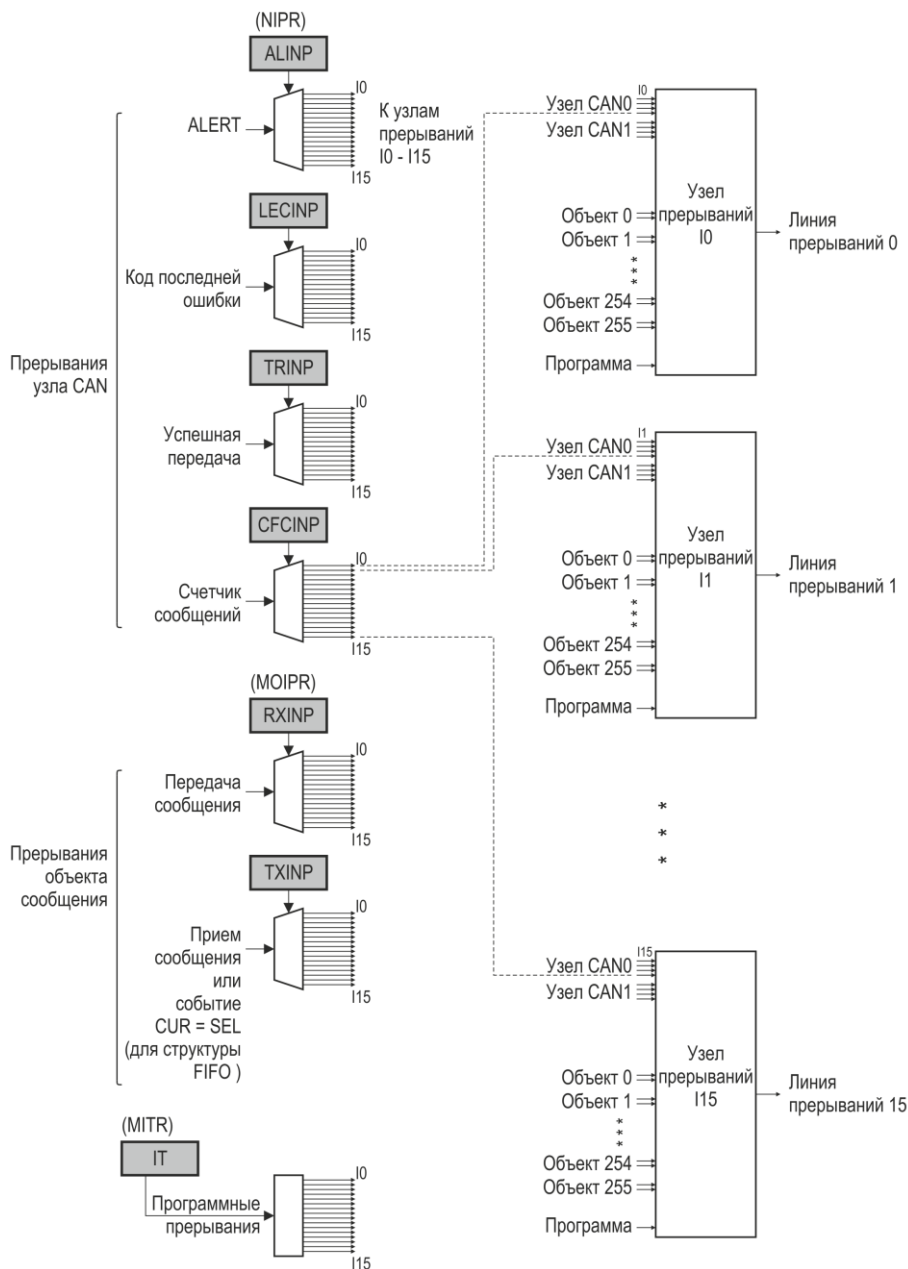


Рисунок 18.13 – Схема коммутации линий прерываний

18.3 Узел контроллера CAN

Каждый узел CAN имеет свою собственную логику управления и выдачи информации о состоянии и может быть сконфигурирован и работать независимо от другого узла.

Режим конфигурации включается установкой бита SSE регистра NCR. Режим конфигурации позволяет изменять параметры синхронизации битов и состояния счетчиков ошибок.

Конфигурация прерываний задается битами TRIE, ALIE и LECIE:

- бит TRIE управляет разрешением прерывания после передачи сообщения;
- бит ALIE управляет разрешением прерываний по ошибке;
- бит LECIE управляет разрешением прерывания по коду последней ошибки.

Регистр NSR отражает текущее состояние, содержит информацию о передачах и ошибках узла.

Блок управления узлом

Координирует работу:

- разрешает/запрещает действия узла на шине;
- разрешает/запрещает и генерирует различные события, касающиеся работы узла (ошибка на шине, успешное завершение передачи сообщения), которые приводят к формированию запросов на прерывания;
- управляет счетчиком сообщений.

Блок синхронизации битов

Согласно стандарту ISO 11898 время передачи одного бита разделено на сегменты, которые, в свою очередь, составлены из целочисленных отрезков времени, называемых квантами времени t_q (см. рисунок 18.14). Квант времени – фиксированная единица времени, получаемая из частоты синхронизации и делителя контроллера CAN.

Сегмент синхронизации T_{sync} позволяет синхронизировать начало обмена данными между передатчиком и приемником. Длительность сегмента всегда равна одному кванту времени.

Сегмент распространения – T_{prop} . Используется для компенсации физического времени запаздывания сигнала в пределах сети. Длительность сегмента рассчитывается с учетом времени прохождения сигнала от передатчика к приемнику и обратно, входной задержки компаратора и задержки выхода драйвера и может составлять от 1 до 8 квантов времени.

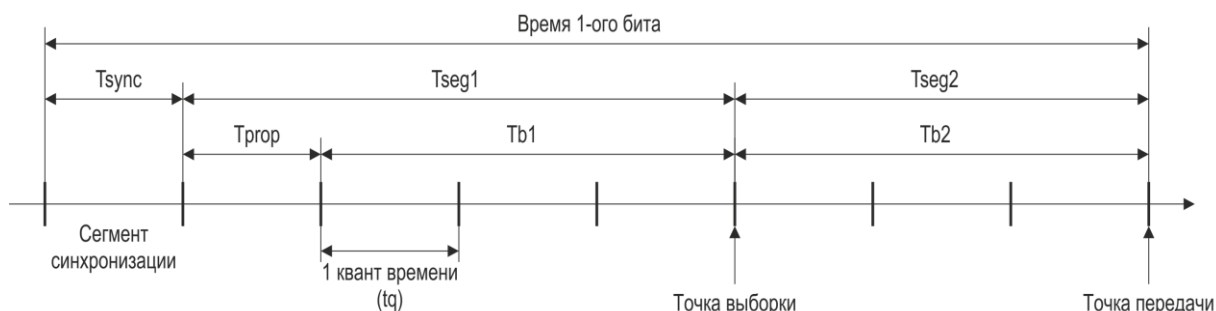


Рисунок 18.14 – Структура одного бита

Сегменты буфера фазы 1 и буфера фазы 2 – T_{b1} и T_{b2} , расположенные до и после точки выборки, используются для компенсации смещения фазы тактовых частот источника и приемника, обнаруживаемой после появления сегмента синхронизации, а также для оптимального расположения точки выборки полученного бита.

Точка выборки – момент, когда читается состояние шины для определения принятого бита. Как правило, длительность временного интервала от начала бита до точки выборки составляет (60 – 70) % времени бита, в зависимости от системных параметров.

Сегмент распространения и сегмент буфера фазы 1 вместе составляют сегмент параметра 1 (T_{seg1}), который определяется битовым полем $TSEG1$ регистра синхронизации битов $NBTR$ (может быть записан, только если установлен бит CCE регистра NCR). Согласно стандарту ISO, минимальная длительность сегмента параметра 1 должна составлять три кванта времени.

Сегмент параметра 2 (T_{seg2}) определяется битовым полем $TSEG2$ регистра $NBTR$ и охватывает сегмент буфера фазы 2. Минимальная длительность сегмента параметра 2 составляет два кванта времени.

Согласно стандарту ISO, минимальная длительность одного бита, получающаяся сложением сегментов T_{sync} , T_{seg1} и T_{seg2} не должна быть менее 8 квантов времени.

Максимальная длительность бита – 25 квантов времени.

Примечание – Минимальное номинальное время передачи одного бита составляет 1 мкс, что соответствует скорости передачи 1 Мбит/с.

Формулы вычисления значений сегментов и времени одного бита Tbit:

- при DIV8 = 0 значение кванта времени

$$tq = (BRP + 1) / f_{out}; \quad (18.4)$$

- при DIV8 = 1 значение кванта времени

$$tq = 8 \times (BRP + 1) / f_{out}; \quad (18.5)$$

- $T_{sync} = 1 \times tq$;

- $T_{seg1} = (TSEG1 + 1) \times tq \geq 3tq$;

- $T_{seg2} = (TSEG2 + 1) \times tq \geq 2tq$;

- $T_{bit} = T_{sync} + T_{seg1} + T_{seg2} \geq 8tq$.

Чтобы компенсировать смещение фазы между частотами генераторов различных узлов шины, каждое устройство должно синхронизироваться по фронту смены уровня сигнала на шине от рецессивного к доминантному. Как только фронт обнаруживается, логика синхронизации сравнивает его текущее положение с ожидаемым и выполняет настройку значений параметров Tseg1 и Tseg2.

Контроллер CAN использует два механизма синхронизации – аппаратный и ресинхронизацию (синхронизация с восстановлением тактовых интервалов).

Аппаратная синхронизация выполняется по каждому фронту смены уровня сигнала на шине от рецессивного к доминантному. При аппаратной синхронизации временные интервалы сегментов, из которых складываются времена битов, не изменяются в течение всего сообщения.

Ресинхронизация выполняется автоматическим удлинением сегмента Tseg1 или укорачиванием сегмента Tseg2. Максимальное значение изменения сегментов колеблется в пределах от 1 до 4 квантов времени. Синхронизация выполняется только при появлении фронта смены уровня сигнала на шине от рецессивного к доминантному. Фиксированное значение максимального числа последовательных бит одинаковой полярности гарантирует своевременное восстановление синхронизации. Смещение фазы фронта смены уровня сигнала на шине отслеживается относительно сегмента синхронизации и измеряется в квантах времени.

Если величина фазового смещения меньше или равна запрограммированному значению ширины перехода ресинхронизации Ts_{sjw}, выполняется аппаратная синхронизация.

Если величина смещения фазы больше, чем Ts_{sjw}, а фазовое смещение положительно, то удлиняется сегмент Tseg1, в случае отрицательного фазового смещения укорачивается сегмент Tseg2.

Значение Ts_{sjw} определяется полем SJW регистра NBTRx по формуле

$$T_{sjw} = (SJW + 1) \times tq. \quad (18.6)$$

Помимо прочего, должны соблюдаться следующие правила:

$T_{seg1} \geq T_{sjw} + T_{prop}$ и $T_{seg2} \geq T_{sjw}$.

Соотношения между максимальным отклонением частоты f_{out} и сегментами буферов фаз и шириной перехода ресинхронизации следующие:

- $\Delta f_{out} \leq T/2 \times (13 \times T_{bit} - T_{b2})$;

- $\Delta f_{out} \leq T_{sjw} / 20 \times T_{bit}$,

где T – меньшее из T_{b1} и T_{b2} .

В итоге:

- T_{sync} составляет 1 квант времени;
- T_{rtr} – от 1 до 8 квантов времени;
- T_{b1} – от 1 до 8 квантов времени;
- T_{b2} – выбирается равным двум квантам времени или равным сегменту T_{b1} , если его значение более двух квантов времени;
- T_{sjw} может составлять максимально 4 кванта времени, однако, в типовых приложениях достаточно 1.

Корректные значения параметров синхронизации битов должны быть записаны в регистр NBTR (доступен, если установлен бит CCE) до окончания инициализации (до сброса бита INIT регистра NCR), т. е. до начала работы CAN узла.

Процессор потока битов

Процессор потока битов формирует (на основе содержимого объектов сообщений) сообщения данных и удаленные запросы непосредственно перед отправкой на шину CAN. Процессор потока управляет генератором CRC (генератор контрольной суммы) и добавляет контрольную сумму к сообщению. После вставки битов начала (SOF) и конца (EOF) сообщения, процессор потока начинает передачу сообщения по правилам арбитража шины CAN. В течение всего времени передачи сообщения процессор потока битов ведет мониторинг шины. Если обнаруживается несовпадение текущего (определяемого мониторингом) и ожидаемого (выдаваемого CAN узлом) уровня напряжения на шине, генерируется ошибка и соответствующий ей запрос на прерывание. Код возникшей ошибки отражается в битовом поле LEC регистра NSR.

Корректность получаемых данных проверяется и подтверждается или не подтверждается кодом CRC. В случае отсутствия подтверждения возникает ошибка, генерируется запрос на прерывание и код ошибки выставляется в регистре NSR. Кроме этого, на шину выдается сообщение об ошибке.

После получения сообщения, не содержащего ошибок, и разбиения его на идентификатор и пакет данных полученная информация записывается в буфер блока обработки сообщений, формируется соответствующее прерывание, и обновляются регистры состояния.

Блок обработки ошибок

Блок обработки ошибок предназначен для выявления ошибок в работе устройств узла. В составе блока есть два счетчика: счетчик ошибок приема (поле REC в регистре NECNT) и счетчик ошибок передачи (поле TEC). Инкрементированием и декрементированием счетчиков управляет процессор потока битов.

Если процессор потока битов сам выявляет ошибку в процессе передачи, то счетчик TEC инкрементируется на 8. Инкрементирование на 1 происходит, если об ошибке сообщено внешним CAN-устройством путем генерирования сообщения об ошибке. Направление передачи с ошибочным сообщением и узел, сообщивший об ошибке передачи, указывают на соответствующие узлы CAN в регистрах NECNT, что используется для анализа ошибки.

В зависимости от значений счетчиков ошибок узел CAN может находиться в одном из трех состояний:

- активной ошибки;
- пассивной ошибки;
- отключен от шины.

Узел находится в состоянии активной ошибки, если значение каждого из счетчиков ошибок меньше 128. Узел в состоянии активной ошибки присоединен к шине и посылает флаг активной ошибки при обнаружении ошибок.

Узел находится в состоянии пассивной ошибки, если значение хотя бы одного из счетчиков ошибок больше или равно 128. Узел подключен к шине, но при обнаружении ошибок посылает флаг пассивной ошибки. После передачи узел в состоянии пассивной ошибки будет ждать инициализации дальнейшей передачи.

Узел находится в состоянии отключения от шины, если значение счетчика ошибок ТЕС больше или равно 256. О том, что CAN узел находится в состоянии отключения от шины, сигнализирует флаг BOFF регистра NSR. Узел в состоянии отключения от шины не может работать с шиной (выходные передатчики отключены).

Флаг EWRN регистра NSR устанавливается, когда хотя бы один из счетчиков достиг или превысил лимит ошибок, определенный в битовом поле EWRNLVL регистра NECNT. Как только значения обоих счетчиков перестанут превышать лимит ошибок, флаг EWRN сбросится.

Счетчик сообщений

Счетчик сообщений может использоваться для получения информации о завершении передачи/приема сообщения соответствующего узла CAN. Подсчет сообщений осуществляется 16 разрядным счетчиком, который управляется регистром NFCR. Битовые поля CFMOD и CFSEL определяют режим работы и событие для инкрементирования счетчика.

Каждый узел CAN имеет в своем составе 16-разрядный счетчик сообщений/синхросчетчик, который подсчитывает количество принятых и переданных сообщений. Битовое поле CFSEL определяет один из трех режимов работы счетчика.

В режиме подсчета сообщений после успешной передачи и/или приема сообщения, содержимое счетчика копируется в битовое поле CFCVAL регистра MOIPR объекта сообщения n, участвующего в пересылке данных. После чего счетчик сообщений инкрементируется.

Прерывания узла CAN

Коммутация линий запросов прерываний показана на рисунке 18.15.

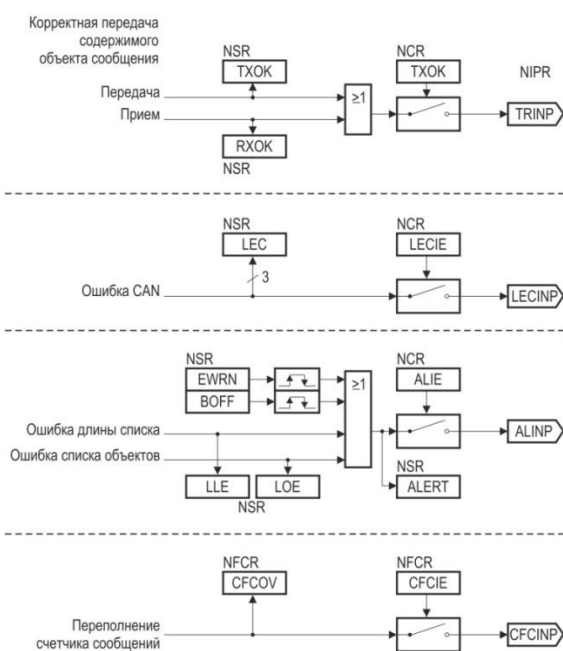


Рисунок 18.15 – Прерывания CAN узла

Узел может генерировать запросы на прерывания в случае:

- успешной передачи/приема сообщения;
- обнаружения кода последней ошибки;
- переполнения счетчика сообщений;
- состояния ALERT (состояние, возникающее, когда хотя бы один из счетчиков ошибок узла достиг значения своего лимита, изменяется состояние «отключен от шины», возникает ошибка длины списка или ошибка списка объектов).

После каждой успешной передачи или успешного приема сообщения генерируется (если разрешено соответствующими битами TXOK и RXOK) прерывание. Битовое поле TRINP регистра NIPR задает одну (из 16) линию прерывания.

Прерывание узла при возникновении кода последней ошибки формируется (если разрешено битом LECIE), если после модификации поля LEC его значение больше нуля. Битовое поле LECINP задает линию прерывания.

Прерывание узла при переполнении счетчика сообщений генерируется, если оно разрешено битом CFCIE регистра NFCR. Битовое поле CFCINP задает линию прерывания.

Прерывание ALERT может быть сформировано (если разрешено битом ALERT) любым из следующих событий:

- изменение состояния бита BOFF;
- изменение состояния бита EWRN;
- ошибка длины списка, которая также выставляет бит LLE;
- ошибка элемента списка, которая также выставляет бит LOE;
- бит INIT выставлен аппаратно.

Битовое поле ALINP задает линию прерывания.

В дополнение к аппаратным прерываниям есть возможность программного генерирования прерываний с использованием регистра прерываний MITR. Запись единицы в n-й разряд битового поля IT генерирует сигнал запроса прерывания на соответствующей ему n-ой линии прерываний (одной из 16). Установка нескольких битов приводит к параллельному генерированию запросов прерываний на соответствующих установленным битам линиях прерываний.

18.4 Объекты сообщений

Регистры управления и состояния объектов сообщений

В состав каждого объекта сообщения входят девять 32-разрядных регистров:

- управления и состояния – MOCTR (только запись) и MOSTAT (только чтение), доступные по одному адресу;
- арбитража – MOAR;
- данных – MODATAN и MODATAL;
- маски – MOAMR;
- указателя прерываний – MOIPR;
- указателя FIFO/шлюза – MOFGPR;
- управления функционированием – MOFCR.

Расположение регистров представлено на рисунке 18.16, где для примера взят пятый объект сообщения.

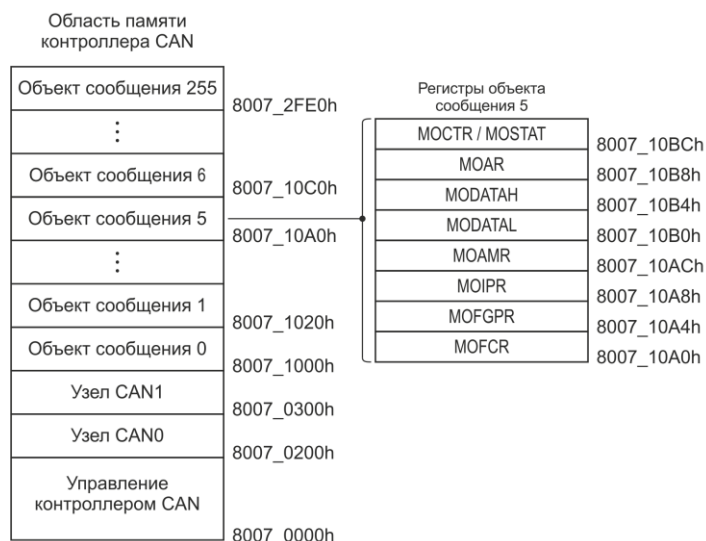


Рисунок 18.16 – Структура памяти регистров

Объекты сообщений контроллера CAN могут быть организованы в восемь списков (см. рисунок 18.17).

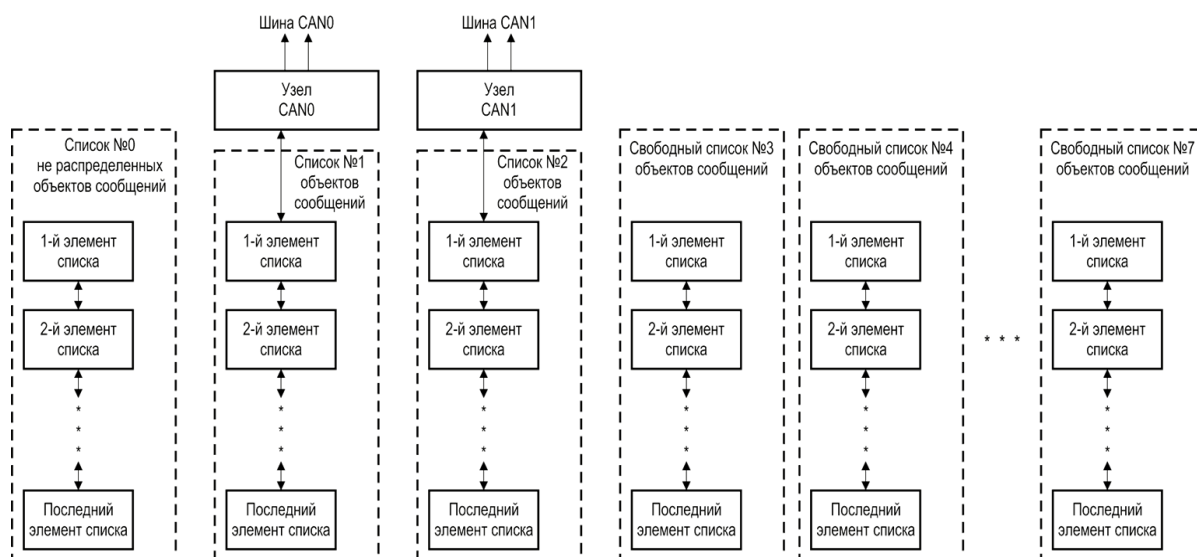


Рисунок 18.17 – Списки контроллера CAN

Каждый объект сообщения может быть добавлен в один из списков. Каждый узел CAN имеет свой список и соответствующий регистр списка. Регистр LIST1 отражает состояние списка №1 узла CAN0, регистр LIST2 – списка №2 узла CAN1.

Примечание – Узел может оперировать только с теми объектами сообщений, которые занесены в принадлежащий ему список.

Положение объекта сообщения n в списке определяется посредством регистра MOSTAT, который содержит указатели на предшествующий ему и следующий за ним элементы списка (объекты). Нераспределенные между узлами CAN объекты сообщений по умолчанию организуются в отдельный список №0, состояние которого отражается в регистре LIST0. Остальные пять списков с номерами от 3 до 7 являются свободными (не принадлежат ни одному узлу) и имеют соответствующие регистры LIST3 – LIST7.

Примечание – Объекты сообщений, распределенные в списки с 3 по 7, не могут быть использованы узлами CAN.

Механизмы FIFO и шлюза (см. далее) оперируют с объектами сообщений независимо от их распределения по спискам, что дает возможность работы со всеми восемью списками. Следовательно, при использовании механизмов FIFO и шлюза следует внимательно следить за содержимым списков.

На рисунке 18.18 представлен вариант, когда объекты сообщений с номерами 3, 5 и 16 занесены в список № 2, принадлежащий узлу CAN1. Состояние списка отражено в регистре LIST2.

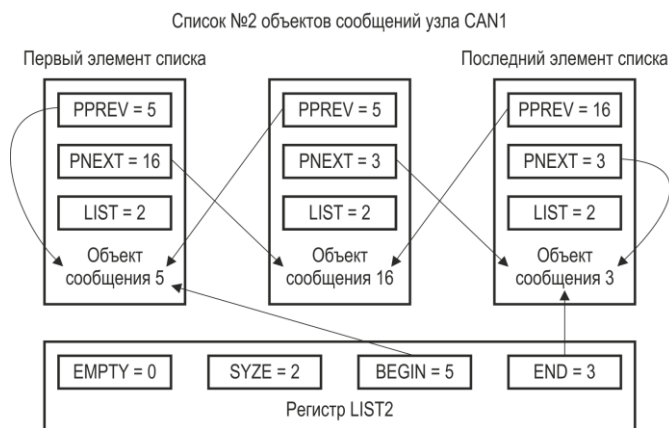


Рисунок 18.18 – Пример списка объектов сообщений

Значение поля BEGIN регистра LIST2 указывает на первый элемент списка (объект сообщения 5). Значение поля END указывает на последний элемент списка (объект сообщения 3). Количество элементов списка (количество объектов сообщений в списке) отражается в поле SIZE (значение SIZE всегда на единицу меньше количества элементов списка). Бит EMPTY является индикатором заполнения списка. Если список пуст, бит EMPTY установлен, в противном случае бит сброшен.

Каждый объект сообщения содержит номер списка (поле LIST), к которому он относится, а также указатели PNEXT и PPREV на следующий по списку объект сообщения и предшествующий, соответственно. Поле PPREV первого по списку объекта сообщения должно указывать на этот же объект. Поле PNEXT последнего по списку объекта сообщения должно указывать на этот же объект.

На рисунке 18.18 указатель PPREV пятого объекта сообщения (первого в списке) имеет значение 5h, а указатель PNEXT третьего объекта сообщения (последнего в списке) имеет значение 3h. Значение поля LIST всех трех объектов сообщений равно 2h.

Объект сообщения, у которого LIST = 0h относится к нулевому списку нераспределенных объектов. После сброса все объекты сообщений считаются нераспределенными. По умолчанию, порядок элементов списка № 0 следующий: объект сообщения (n – 1) является предыдущим объекта сообщения n, а объект сообщения (n + 1) – следующим.

Для просмотра структуры списка объектов сообщений узла достаточно обратиться к соответствующим регистрам LIST1/LIST2 и MOSTAT.

Структура списка управляется и изменяется посредством контроллера списка, который, в свою очередь, управляется панелью команд, основное назначение которой – упрощение внесения изменений в структуру списка, отслеживание этих изменений и проверка их корректности с помощью регистра PANCTR.

Панель команд запускается записью соответствующей команды в битовое поле PANCMD. До записи кода команды должны быть записаны соответствующие аргументы команды в битовые поля PANAR1 и PANAR2.

Примечание – Запись новых значений в поля PANAR1 и PANAR2 не изменяет сразу их содержимого. Новые значения сначала попадают в специальный теневой регистр. Далее, одновременно с записью кода команды в поле PANCMD, новые значения из теневого регистра переносятся в поля PANAR1 и PANAR2.

С записью корректного кода команды выставляется флаг BUSY, и в дальнейшем все попытки записи в регистр PANCTR игнорируются. Флаг BUSY остается активным, а панель команд заблокированной до тех пор, пока не завершится выполнение записанной команды.

После сброса микроконтроллера контроллер списка формирует список № 0 нераспределенных объектов сообщений. Во время этой операции флаг BUSY установлен, и все обращения к объектам сообщений запрещены. По окончании этой операции флаг BUSY сбрасывается, и объекты становятся доступными.

В случае появления команды динамического распределения, по которой какой-либо элемент забирается из списка № 0 и переносится в другой указанный список, наряду с битом BUSY, устанавливается бит RBUSY. Это указывает на то, что значения битовых полей PANAR1 и PANAR2 будут обновлены контроллером списка следующим образом:

- номер объекта сообщения, переносимого из списка № 0 нераспределенных объектов сообщений, записывается в PANAR1;

- если установлен бит ERR (седьмой бит поля PANAR2), значит, список № 0 пуст и выполнение команды завершается; если бит ERR сброшен – список № 0 не пуст и команда выполняется.

Результаты выполнения команды динамического распределения записываются до того, как контроллер списка начнет процесс распределения. Как только результаты станут доступны, бит RBUSY сбрасывается. Это позволяет пользователю запрограммировать настройки желаемого объекта сообщения, в то время как контроллер списка распределяет объекты. Во время операций со списками доступ к объектам сообщений не запрещен, но следует помнить, что любой доступ к регистрам объектов сообщений в течение процесса распределения объектов вносит задержку (в процесс), равную длительности доступа.

Код команды «нет операции» автоматически записывается в битовое поле PANCMD.

Новая команда может быть записана в любое время, когда бит BUSY сброшен.

Все битовые поля регистра PANCTR, исключая биты BUSY и RBUSY, могут быть записаны программно, что делает возможным сохранять и восстанавливать значения регистра PANCTR, если панель команд используется независимой подпрограммой обработки прерываний. Если возникает такая ситуация, то любые задачи, которые используют панель команд и которые могут прерывать выполнение других задач, тоже использующих панель команд, будут опрашивать состояние флага BUSY. До тех пор, пока флаг BUSY будет оставаться установленным, содержимое регистра PANCTR будет сохранено в соответствующей области памяти до операции восстановления. Как только подпрограмма обработки прерываний закончится, содержимое регистра PANCTR будет восстановлено.

До того, как объект сообщения, занесенный в список активного узла CAN, будет перенесен на другую позицию этого же списка или перенесен в другой список, бит MSGVAL регистра MOSTATn объекта сообщения n должен быть очищен.

Примечание – Если требуется перераспределить объекты сообщений в списки повторно, необходимо приостановить работу узлов CAN (установить бит INIT регистра NCR), а после занесения объектов в списки возобновить ее (сбросить бит INIT).

18.5 Прием и передача сообщений

Прием сообщения

После завершения приема сообщение сохраняется в объекте сообщения в соответствии с установленным алгоритмом (см. рисунок 18.19).

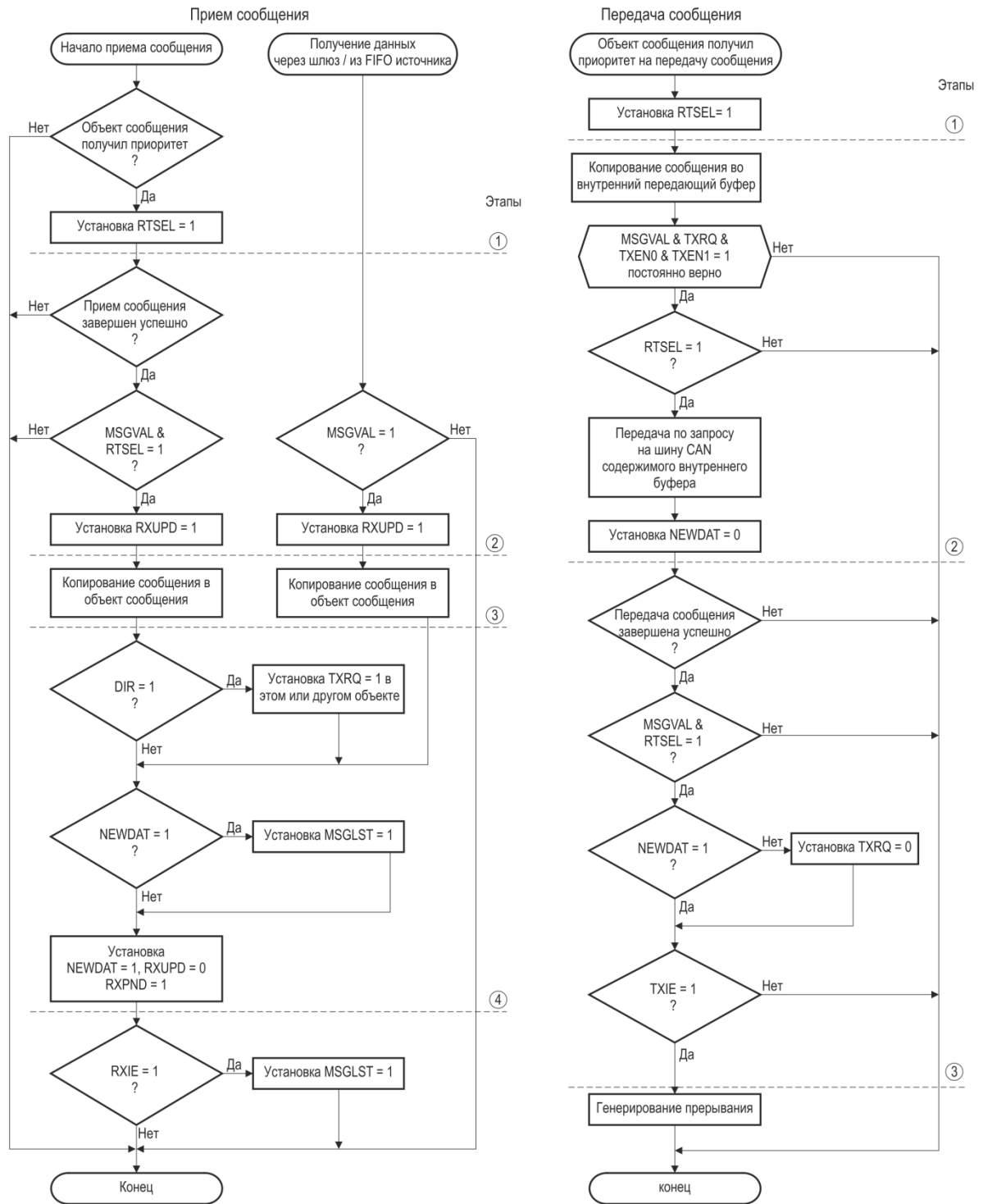


Рисунок 18.19 – Алгоритмы приема и передачи сообщения

Помимо сохранения данных в объекте сообщения, контроллер CAN осуществляет обмен данными с ЦП.

При приеме сообщения информация сохраняется в объекте сообщения только в том случае, если установлен бит MSGVAL регистра MOSTAT. Если ЦП очищает бит MSGVAL, контроллер CAN останавливает запись в объект сообщения, и далее объект может быть реконфигурирован центральным процессором с последующей записью в него информации без участия контроллера CAN.

Полученное с шины сообщение может быть сохранено в объекте сообщения только в случае, если установлен бит RXEN. Контроллер CAN проверяет состояние бита RXEN только во время фильтрации принимаемого сообщения. После того, как сообщение принято, состояние бита не имеет значения и не оказывает влияния на дальнейшее сохранение данных в объекте сообщения.

Бит RXEN позволяет управлять блокированием объекта сообщения – после сброса бита RXEN полученное сообщение сохраняется в объекте сообщения, который получил приоритет, но в сохранении последующих сообщений этот объект не принимает участия.

Реконфигурация объекта сообщения центральным процессором во время работы контроллера CAN (например, сброс бита MSGVAL, изменение объекта сообщения и повторная установка бита MSGVAL) происходят следующим образом:

- объект сообщения получает приоритет;
- ЦП очищает бит MSGVAL для реконфигурации объекта сообщения;
- после реконфигурации ЦП снова устанавливает бит MSGVAL;
- завершается получение сообщения;
- если установлен бит MSGVAL, полученные данные сохраняются в объекте сообщения, генерируется запрос на прерывание, устанавливается соответствующий флаг;
- если сконфигурировано, производятся шлюзовые и FIFO операции.

Примечание – После реконфигурации объекта сохранение данных по завершении получения сообщения может быть нежелательным. Запретить запись данных в объект сообщения можно посредством бита RTSEL.

После получения объектом сообщения приоритета его бит RTSEL устанавливается контроллером CAN, открывая, таким образом, объект сообщения для записи. После приема сообщения контроллер CAN дополнительно проверяет возможность записи в объект сообщения, а именно – установлен ли все еще бит RTSEL. И только в том случае, если бит RTSEL установлен, полученные данные сохраняются в объекте сообщения (вместе со всеми последующими действиями, которые указаны выше).

Если во время операций контроллера CAN объект сообщения становится некорректным (сброс бита MSGVAL), бит RTSEL должен быть сброшен до того, как бит MSGVAL будет установлен снова, или, по крайней мере, одновременно с ним. Это необходимо для предотвращения сохранения старой информации в объекте сообщения.

Реконфигурация объекта сообщения должна происходить следующим образом:

- сброс бита MSGVAL;
- реконфигурация объекта сообщения, пока бит MSGVAL сброшен;
- сброс бита RTSEL и далее установка бита MSGVAL.

Индикатором процесса сохранения (изменения) данных в объекте сообщения является флаг RXUPD, который выставляется с началом процесса сохранения (изменения) и сбрасывается с его окончанием.

После сохранения полученного сообщения (идентификатора, бита IDE, кода длины данных, поля данных, в случае сообщения данных) выставляется флаг NEWDAT. Если к моменту выставления (завершение сохранения/изменения данных) флаг NEWDAT был

уже установлен, выставляется флаг MSGLST, который говорит о том, что произошла потеря данных.

Флаги RXUPD и NEWDAT позволяют произвести чтение корректных данных из объекта сообщения во время текущих операций контроллера CAN. Рекомендуемая последовательность действий следующая:

- сброс флага NEWDAT;
- чтение данных (идентификатор, данные и т. д.) из объекта сообщения;
- проверка флагов NEWDAT и RXUPD – оба флага должны быть сброшены. В случае невыполнения этого условия возвращение к первому действию;
- если флаги NEWDAT и RXUPD сброшены, то содержимое объекта сообщения корректно и не используется контроллером CAN в течение операции чтения.

Поведение флагов RXUPD, NEWDAT и MSGLST идентично как для сообщений данных, так и для сообщений удаленных запросов.

Передача сообщения

Алгоритм передачи сообщений показан на рисунке 18.19. Одновременно с копированием данных (идентификатора, бита IDE, бита RTR, равного биту DIR, кода длины данных и собственно данных) из объекта сообщения, содержимое которого должно быть передано во внутренний передающий буфер соответствующего узла CAN, для контроля соблюдения четкой последовательности выполнения всех операций устанавливаются биты состояния.

Сообщение может быть передано только в случае, когда все четыре бита MSGVAL, TXEN0, TXEN1 и TXRQ установлены.

Бит RTSEL выставляется после того, как объект сообщения получает приоритет для передачи своего содержимого. Когда данные объекта сообщения копируются в передающий буфер, бит RTSEL проверяется, и если он установлен, сообщение передается. После успешной передачи сообщения бит RTSEL проверяется снова, и если он установлен, осуществляются дальнейшие операции.

Для полной и завершенной реконфигурации корректного объекта сообщения должны быть выполнены следующие шаги:

- очистка бита MSGVAL;
- реконфигурация объекта сообщения, пока бит MSGVAL сброшен;
- сброс бита RTSEL и установка бита MSGVAL.

Сброс бита RTSEL гарантирует как полное отключение объекта сообщения от текущей передачи, так и то, что никакие операции (копирование данных в передающий буфер, включая сброс бита NEWDAT, очистка бита TXRQ, прерывание сообщения и т. д.), относящиеся к старой конфигурации этого объекта сообщения, не повлияют на новую конфигурацию после установки бита MSGVAL.

После завершения передачи содержимого объекта сообщения в передающий буфер узла CAN, флаг NEWDAT аппаратно сбрасывается, тем самым обозначая, что объект сообщения открыт для записи новых данных.

Если после успешной передачи сообщения (на CAN-шину) флаг NEWDAT все еще остается сброшенным (в объект сообщения не были записаны новые данные), флаг TXRQ аппаратно сбрасывается. Если же флаг NEWDAT был установлен программно (в связи с необходимостью передачи новых данных), флаг TXRQ не сбрасывается, тем самым разрешая передачу новых данных.

18.6 Фильтрация сообщений

Фильтрация при получении сообщений

При получении узлом CAN сообщения определяется объект сообщения, в котором будут сохранены получаемые данные в случае успешного приема.

Объект сообщения считается корректным для приема, если одновременно соблюдаются условия:

- объект сообщения распределен в список объектов сообщений узла, который принимает сообщение;

- бит MSGVAL установлен;

- бит RXEN установлен;

- бит DIR равен биту RTR принимаемого сообщения. Если бит DIR установлен, объект сообщения (объект передачи) может принять только сообщение удаленного запроса. Если бит DIR сброшен (объект приема), объект сообщения может принять только сообщение данных;

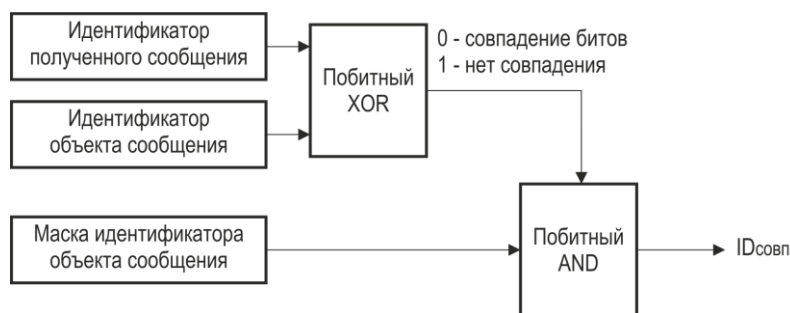
- если бит MIDE установлен, то бит IDE получаемого сообщения оказывает следующее влияние:

- если бит IDE (регистр MOAR) установлен, то бит IDE принимаемого сообщения должен быть равен единице (расширенный идентификатор);

- если бит IDE сброшен, бит IDE принимаемого сообщения должен быть равен нулю (стандартный идентификатор);

- если бит MIDE сброшен, значение бита IDE принимаемого сообщения не важно, т.е. допускаются сообщения, как со стандартным, так и с расширенным идентификатором;

- идентификатор полученного сообщения полностью (побитно) совпадает с идентификатором, хранящимся в регистре MOARn объекта сообщения, за исключением битов, закрытых маской регистра MOAMRn, значение которых не важно. На рисунке 18.20 показан пример проверки идентификатора.



IDсовп = 0: идентификатор ID полученного сообщения совпал с ID объекта сообщения
IDсовп > 0: идентификатор ID полученного сообщения не совпал с ID объекта сообщения

Рисунок 18.20 – Проверка идентификатора полученного сообщения

Среди всех объектов сообщений, которые отвечают указанным выше критериям, для сохранения полученного сообщения выбирается объект с наивысшим приоритетом. Для задания приоритета используется поле PRI в регистре MOAR. Объект сообщения, у которого значение поля PRI меньше, имеет больший приоритет. При равенстве значений поля PRI приоритетным считается объект сообщения, который предшествует следующему в списке.

Фильтрация при передаче сообщений

Когда требуется передача содержимого какого-либо объекта сообщения, в соответствующих управляющих регистрах выставляются флаги, указывающие на необходимость передачи. Объект сообщения считается корректным для передачи, если одновременно соблюдаются условия:

- объект сообщения распределен в список объектов сообщений узла CAN;
- флаг MSGVAL установлен;
- флаг TXRQ установлен;
- флаги TXEN0 и TXEN1 установлены.

Может возникнуть ситуация, когда передачи требуют одновременно несколько объектов сообщений. Среди всех объектов, которые отвечают указанным выше критериям, для передачи выбирается объект с наивысшим приоритетом.

Объект сообщения, у которого значение поля PRI меньше, имеет больший приоритет. При равенстве значений поля PRI разных объектов приоритет определяется следующим образом:

- при PRI = 10b – согласно правилам арбитража передачи сообщения;
- при PRI = 01b/11b приоритет имеет объект сообщения, который предшествует следующему в списке.

Объект сообщения, являющийся корректным для передачи и имеющий приоритет, будет осуществлять передачу первым. Остальные объекты сообщений будут переданы по очереди, согласно их приоритетам.

Объект сообщения определяется как стандартный объект сообщения, если в регистре MOFCR значение битового поля MMC равно нулю. Стандартный объект сообщения может принимать и передавать сообщения, согласно правилам, описанным выше.

На рисунке 18.21 показано формирование запроса на передачу объекта сообщения.

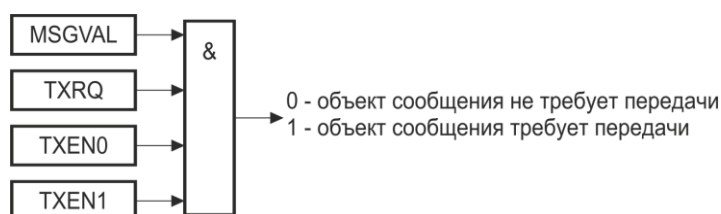


Рисунок 18.21 – Формирование запроса на передачу объекта сообщения

18.7 Удаленные запросы

После получения узлом CAN сообщения удаленного запроса и сохранения его в объекте сообщения, выставляется бит запроса передачи для ответа на удаленный запрос (отправка сообщения данных) или для автоматического повторения запроса.

В зависимости от состояния бита FRREN объекта сообщения, который принял сообщение удаленного запроса, возможны два варианта действий:

- если бит FRREN сброшен, то устанавливается флаг TXRQ этого объекта;
- если бит FRREN установлен, то устанавливается флаг TXRQ того объекта, на который указывает поле CUR объекта, принявшего удаленный запрос. При этом поле CUR не меняет своего значения.

Состояние регистров объекта сообщения, передающего сообщение удаленного запроса

У объекта сообщения, передающего сообщение удаленного запроса, в регистре MOSTAT должен быть сброшен бит DIR (объект передает сообщение данных) и установлены биты TXEN0, TXEN1, MSGVAL и TXRQ. Значение идентификатора в регистре MOAR передающего объекта сообщения должно быть равно значению идентификатора принимающего объекта сообщения (или совместно с регистром MOAMR обеспечивать успешное прохождение фильтрации), чтобы сообщение удаленного запроса было принято принимающим объектом другого узла. Само сообщение удаленного запроса должно содержать идентификатор принимающего объекта сообщения, поэтому значение регистра MODATAL передающего объекта сообщения должно быть равно значению регистра MOAR принимающего объекта.

Состояние регистров объекта сообщения, принимающего сообщение удаленного запроса при FRREN = 0

У объекта сообщения, принимающего сообщение удаленного запроса, должны быть установлены биты DIR (объект принимает сообщение удаленного запроса), TXEN0 и TXEN1 (если отвечать на запрос будет сам), RXEN и MSGVAL. Регистры MODATAL и MODATAH должны содержать данные, которые будут переданы в ответ на запрос.

Состояние регистров объекта сообщения, принимающего сообщение удаленного запроса (при FRREN = 1) и содержащего данные для ответа на запрос

У объекта сообщения, принимающего сообщение удаленного запроса, должны быть установлены биты DIR, RXEN и MSGVAL. Битовое поле CUR должно указывать на номер объекта сообщения (должен находиться в том же узле, что и объект принявший сообщение удаленного запроса), содержащего данные, предназначенные для передачи в ответ на поступивший удаленный запрос.

В свою очередь у объекта сообщения, хранящего данные для отправки в ответ на запрос, должны быть установлены биты DIR, (объект передает сообщение данных), TXEN0, TXEN1 и MSGVAL. Бит TXRQ устанавливается автоматически при приеме сообщения удаленного запроса принимающим объектом сообщения.

Прием ответа на запрос (переданного сообщения данных) осуществляется стандартным объектом сообщения запрашивающего узла CAN (обмен данными происходит между объектом сообщения, хранящим данные для отправки в ответ на запрос, и объектом сообщения запрашивающего узла).

18.8 Дополнительные режимы передачи

Дополнительно имеются два режима, каждый из которых может быть выбран индивидуально:

- режим передачи данных с защитой от повторений;
- режим однократной пересылки данных.

Режим передачи данных с защитой от повторения

Выбирается установкой бита SDT регистра MOFCR.

После приема сообщения данных и сохранения его в объекте с установленным битом SDT, бит MSGVAL этого объекта аппаратно сбрасывается, чтобы исключить возможность повторного приема и записи в этот объект. Этот режим нельзя использовать для базового объекта FIFO структуры.

В ответ на сообщение удаленного запроса, принятое объектом с установленным битом SDT, будут отправлены данные из объекта сообщения, на который указывает поле

CUR объекта, принявшего удаленный запрос. После этого бит MSGVAL объекта принявшего сообщение удаленного запроса сбросится.

Примечание – Объект, принявший сообщение удаленного запроса, не может быть источником данных, передаваемых в ответ на запрос. Это означает, что в данном режиме бит FRREN объекта, принявшего удаленный запрос, обязательно должен быть установлен.

Режим однократной пересылки данных

Выбирается установкой бита STT.

Бит TXRQ сбрасывается, когда содержимое объекта сообщения копируется в передающий буфер узла CAN. Таким образом, в дальнейшем, при неудачной (вследствие ошибок) пересылке сообщения по CAN-шине, повторной передачи не будет.

18.9 FIFO структура объектов сообщений

Регистр MOFGPRn объекта сообщения n содержит установки указателей на объекты сообщений, которые используются при операциях FIFO и шлюзовых операциях.

В случае сильной загрузки ЦП обработка серии сообщений может быть затруднена – например, вследствие получения и/или передачи большого числа сообщений за малые промежутки времени. Для таких случаев предусмотрена система буферов быстрого ввода-вывода, так называемая FIFO структура, которая может функционировать автоматически и позволяет избежать потери принимаемых сообщений, минимизировать время подготовки сообщений к отправке, а также генерировать прерывания по окончании операций.

Допускается организация нескольких параллельных FIFO структур. Число структур и их составляющих зависит только от количества доступных объектов сообщений. FIFO структура может быть создана, изменена и удалена в любой момент времени, даже во время операций контроллера CAN.

На рисунке 18.22 представлена основная FIFO структура. Она состоит из одного базового объекта и n-ого числа вспомогательных объектов.

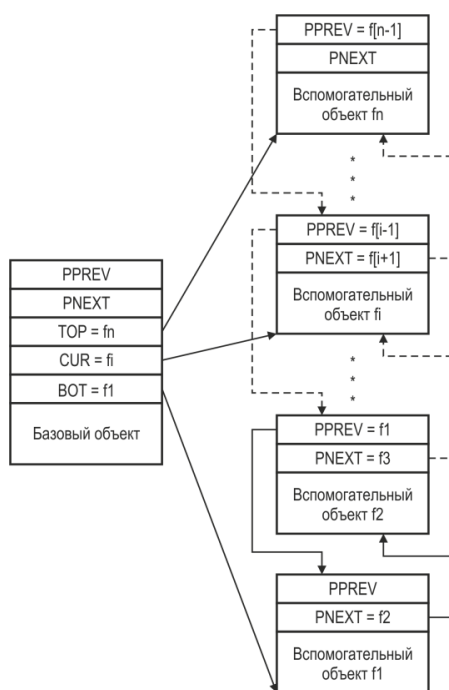


Рисунок 18.22 – FIFO структура с базовым объектом и n вспомогательными объектами

Вспомогательные объекты объединяются последовательно в списки (подобно спискам объектов сообщений). Базовый объект передающей FIFO-структуры может быть занесен в любой список. Хотя на рисунке базовый объект не относится ни к одному из списков, он может быть вставлен в любую последовательность вспомогательных объектов. Это означает, что базовый объект одновременно является и вспомогательным объектом (шлюзовые операции не возможны). Порядковые номера объектов сообщений (0, 1, 2 и т. д.) не имеют никакого значения при FIFO операциях с объектами.

Вспомогательные объекты должны быть определены в общий список (так как они последовательно связаны). С помощью указателей (битовые поля BOT, CUR и TOP) можно присоединять базовый объект к вспомогательному объекту, независимо от того, принадлежат базовый и вспомогательный объекты одному списку или разным спискам, но базовый должен быть первым в списке в таком случае.

Минимальная FIFO структура может состоять из одного объекта сообщения, который будет одновременно являться и базовым, и вспомогательным (фактически не используется). Максимальная FIFO структура может включать в себя все 256 объектов сообщений.

В базовом объекте FIFO границы установлены: поле BOT указывает на самый младший элемент FIFO структуры, поле TOP – на самый старший элемент, поле CUR – на вспомогательный объект, который в настоящий момент выбран контроллером CAN для передачи сообщения. Как только начинается передача, в CUR записывается номер следующего по списку вспомогательного объекта сообщения ($CUR = PNEXT$ используемого объекта). Если значение битового поля CUR достигло номера старшего элемента списка ($CUR = TOP$), то следующим значением будет BOT (реализация автоматического перехода в начало списка). Таким образом, реализуется замкнутая FIFO структура, в которой битовые поля TOP и BOT устанавливают связь между началом и концом списка.

Битовое поле SEL позволяет определить вспомогательный объект в пределах списка, для которого генерируется прерывание всякий раз, когда указатель CUR достигает значения указателя SEL. Также битовое поле SEL позволяет отследить окончание запланированной передачи серии сообщений или выдать прерывание, предупреждающее о том, что FIFO структура становится заполненной.

Вспомогательные объекты приемной FIFO-структуры могут принадлежать списку любого узла.

FIFO структура для приема

Используется для буферизации входящих сообщений данных и удаленных запросов.

FIFO структура для приема активируется записью значения 0001b в битовое поле MMC регистра MOFCR базового объекта. Эта запись автоматически определяет объект как базовый объект приема FIFO. Типы вспомогательных объектов FIFO не имеют значения при операциях.

Когда базовый объект FIFO получает сообщение от узла CAN, которому он принадлежит, сообщение сохраняется не в этом базовом объекте, а во вспомогательном объекте сообщения, на который указывает битовое поле CUR. При этом по умолчанию предполагается, что для вспомогательного объекта $MMC = 0000b$ (действительное значение MMC игнорируется), и никаких операций фильтрации принимаемого сообщения не производится.

Одновременно с приемом сообщения текущее значение указателя CUR базового объекта меняется на номер следующего по списку вспомогательного объекта FIFO структуры. Этот вспомогательный объект будет использован для приема следующего сообщения.

Если установлен флаг OVIE регистра MOFCR базового объекта и значение указателя CUR становится равным значению указателя SEL, генерируется прерывание переполнения. Это прерывание генерируется на узле прерываний с указателем TXINP

базового объекта сразу после сохранения полученного сообщения во вспомогательном объекте. Прерывания генерируются, если это разрешено битом TXIE.

Следует помнить, что сообщение сохраняется в базовом и вспомогательном объектах FIFO, только если установлен бит MSGVAL.

Во избежание непосредственного приема сообщения вспомогательным объектом, как если бы он был независимым объектом и не принадлежал FIFO структуре, флаги RXEN всех вспомогательных объектов должны быть сброшены. Состояние флага RXEN неважно в случае, когда вспомогательный объект занесен в список, не связанный с узлом CAN.

FIFO структура для передачи

Используется для буферизации серий сообщений данных или удаленных запросов, которые должны быть отправлены. FIFO структура для передачи состоит из базового объекта и одного или более вспомогательных объектов.

FIFO структура для передачи активируется записью значения 0010b в поле MMC регистра MOFCR базового объекта. В отличие от FIFO структуры для приема, в битовые поля MMC вспомогательных объектов (FIFO структуры для передачи) должно быть записано значение 0011b. Указатели CUR всех вспомогательных объектов должны указывать на базовый объект FIFO передачи (чтобы инициализироваться программно).

Флаги TXEN1 всех вспомогательных объектов сообщений, за исключением одного, на который указывает указатель CUR базового объекта, должны быть программно сброшены. Флаг TXEN1 указанного объекта должен быть установлен. Указатель CUR базового объекта может быть инициализирован для любого вспомогательного объекта.

При определении корректности объектов сообщений FIFO структуры для начала FIFO-операций базовый объект должен быть определен первым как корректный, т. е. MSGVAL должен быть установлен.

В случае необходимости удаления FIFO структуры, прежде чем начнется операция удаления, все вспомогательные объекты, принадлежащие этой FIFO структуре, должны быть определены как некорректные (биты MSGVAL должны быть сброшены).

FIFO структура для передачи использует флаги TXEN1 всех своих объектов для выбора сообщения для передачи. В результате фильтрации право передавать сообщение получает тот объект, у которого выставлен флаг TXEN1. После передачи сообщения флаг TXEN1 аппаратно сбрасывается, а в указатель CUR записывается номер следующего объекта, требующего отправки сообщения, для которого уже выставлен (аппаратно) свой флаг TXEN1, и так далее для всей FIFO структуры.

Если установлен флаг OVIE регистра MOFCRn базового объекта и значение указателя CUR становится равным значению указателя SEL, генерируется прерывание переполнения. Это прерывание генерируется на узле прерываний с указателем RXINP базового объекта после завершения операций получения сообщения. Прерывания приема базового объекта генерируются, если это разрешено битом RXIE.

Программирование регистров для FIFO структуры

1 Для передающего базового объекта:

- сбросить бит MSGVAL;

- задать поля CUR, BOT, TOP, SEL;

- записать значение 0010b в поле MMC, задать DLC, установить биты OVIE и RXIE (если необходимо).

Примечание – Состояние регистров MOAR и MOAMR передающего базового объекта не важно, поскольку в передаче участвуют передающие вспомогательные объекты и принимающий базовый объект. Поле RXINP указывает линию, на которую будет выдаваться прерывание переполнения (CUR = SEL).

2 Для передающих вспомогательных объектов:

- сбросить бит MSGVAL;
- установить биты DIR, TXEN1 (только для того вспомогательного объекта, на который указывает поле CUR передающего базового объекта, у остальных вспомогательных объектов бит TXEN1 должен быть сброшен), TXEN0;
- записать в поле CUR номер передающего базового объекта;
- записать значение 0011b в поле MMC, задать DLC.

Примечание – Значения регистров MOAR передающих вспомогательных объектов должно совпадать (или совместно с регистрами MOAMR обеспечивать успешное прохождение фильтрации) со значением регистра MOAR принимающего базового объекта, так как процесс передачи фактически происходит между ними (или иного принимающего объекта, если на приеме используется не FIFO структура).

3 Для принимающего базового объекта:

- установить бит RXEN;
- задать поля CUR, BOT, TOP, SEL;
- записать значение 0001b в поле MMC, задать DLC, установить биты OVIE и TXIE (если необходимо).

Примечание – Значение регистра MOAR принимающего базового объекта должно быть равно значению регистров MOAR передающих вспомогательных объектов передачи (или совместно с регистром MOAMR обеспечивать успешное прохождение фильтрации). Поле TXINP указывает, на какую линию будет выдаваться прерывание переполнения (прерывание после операции сохранения полученного сообщения во вспомогательных объектах при CUR = SEL).

4 Для принимающих вспомогательных объектов:

- сбросить бит RXEN (не требуется, если вспомогательные объекты занесены в список, не связанный с узлом CAN);
- задать поле DLC (состояние поля MMC не важно).

Примечание – Состояние регистров MOAR, принимающих вспомогательные объекты, не важно.

5 Установить бит MSGVAL в первую очередь у передающего базового объекта, а затем у всех остальных объектов.

6 Установить бит TXRQ для всех передающих вспомогательных объектов, начиная с того, на который указывает поле CUR передающего базового объекта.

18.10 Режим шлюза

Режим позволяет реализовывать автоматическую передачу информации через шлюз между двумя независимыми шинами CAN без участия ЦП.

Шлюз можно сформировать на уровне объектов сообщений и осуществлять передачу информации между узлами CAN. Шлюз может быть сформирован между двумя любыми объектами сообщений, принадлежащими разным узлам CAN. Количество шлюзов зависит только от количества объектов сообщений, допускающих формирование шлюзов.

Режим шлюза активируется записью значения 0100b в битовое поле MMC регистра MOFCR объекта сообщения n, инициализирует его как шлюзовый объект-источник. Объект сообщения, который будет являться шлюзовым объектом-приемником, выбирается указателем CUR объекта-источника. Для формирования шлюза достаточно,

чтобы объект-приемник был корректным (установлен бит MSGVAL). Остальные параметры не влияют на возможность осуществления передачи между объектами от источника к приемнику.

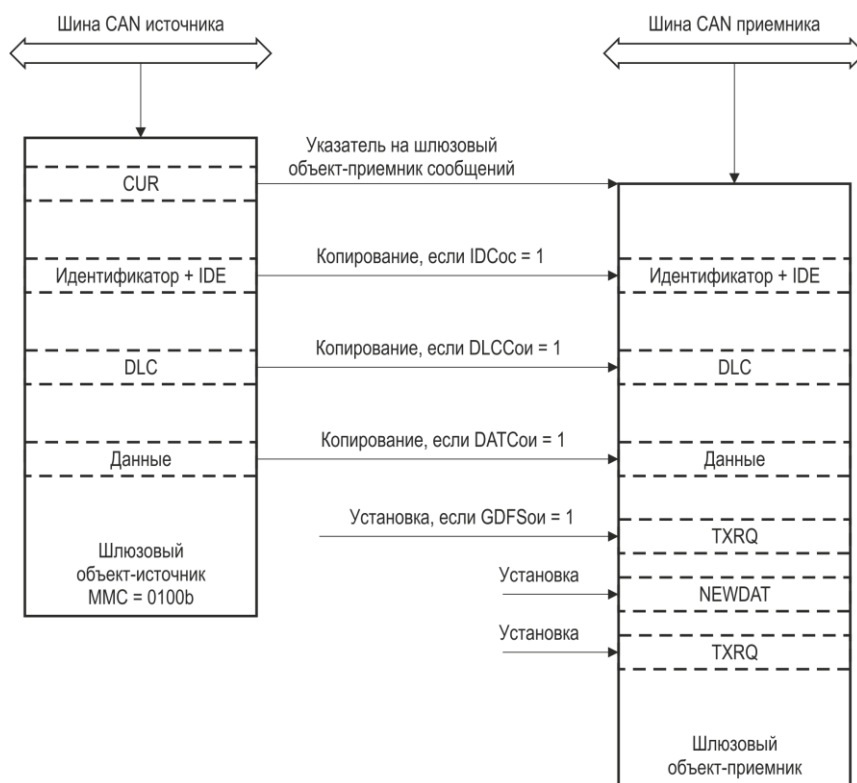


Рисунок 18.23 – Передача через шлюз от источника к приемнику

Шлюзовый объект-источник (см. рисунок 18.23) функционирует как обычный объект сообщения с тем отличием, что возможны дополнительные действия контроллера CAN при приеме и сохранении сообщения в объекте-приемнике:

1 Если установлен флаг DLCC регистра MOFCR_n объекта-источника, код длины данных DLC копируется из шлюзового объекта-источника в шлюзовый объект-приемник.

2 Если установлен флаг IDC объекта-источника, идентификатор ID и расширение IDE копируются из шлюзового объекта-источника в шлюзовый объект-приемник.

3 Если установлен флаг DATC объекта-источника, байты данных, хранящиеся в двух регистрах MODATAL и MODATAN объекта-источника, копируются из шлюзового объекта-источника в шлюзовый объект-приемник. Копируются все 8 байт данных, вне зависимости от значения поля DLC.

4 Если установлен флаг GDFS объекта-источника, то устанавливается бит запроса передачи TXRQ объекта-приемника.

5 Устанавливаются флаги RXPND и NEWDAT регистра MOSTAT объекта-приемника.

6 Если установлен флаг RXIE регистра MOSTAT объекта-приемника, то генерируется запрос на прерывание.

7 Указатель CUR объекта-источника переводится на следующий объект-приемник по правилам FIFO структуры. Сформировать шлюз между объектом-источником и одним объектом-приемником (значение указателя CUR будет оставаться неизменным) возможно программированием:

TOP = BOT = CUR = номер объекта-приемника.

Организация шлюза «объект-источник – объект-приемник» аналогична организации FIFO структуры «базовый объект – вспомогательный объект», что указывает на

возможность формирования шлюза с интегрированным FIFO-приемником. При получении сообщения данных (объект-источник является объектом приема, т. е. его бит DIR сброшен) и при получении удаленного запроса (объект-источник является объектом передачи) через шлюз используется один и тот же механизм.

Несмотря на то, что механизм удаленных запросов работает независимо от типа объекта сообщения, он наиболее полезен при использовании шлюзов, для формирования удаленных запросов на шине шлюзового объекта-источника после получения удаленного запроса на шине шлюзового объекта-приемника. В зависимости от значения бита FRREN шлюзового объекта-приемника, есть два варианта обработки удаленного запроса, возникшего с той стороны шлюза, где расположен объект-приемник (при условии, что происходит передача из объекта-источника в объект-приемник, т. е. DIR (источника) = 0 и DIR (приемника) = 1).

1 Обработка запроса шлюзового объекта-приемника с FRREN = 0b:

- сообщение удаленного запроса принимается шлюзовым объектом-приемником;
- бит TXRQ шлюзового объекта-приемника устанавливается автоматически;
- сообщение данных с текущей информацией, хранящейся в объекте-приемнике, передается на шину приемника.

2 Обработка запроса шлюзового объекта-приемника с FRREN = 1b:

- сообщение удаленного запроса принимается шлюзовым объектом-приемником;
- бит TXRQ шлюзового объекта-источника (объект должен быть указан в поле CUR объекта-приемника), устанавливается автоматически;
- сообщение данных передается объектом-источником на шину CAN источника;
- получатель удаленного запроса в ответ выдает сообщение данных на шину источника;
- сообщение данных сохраняется в объекте-источнике;
- сообщение данных копируется в объект-приемник (через шлюз);
- выставляется бит TXRQ объекта-приемника (при условии, что GDFS источника = 1);
- новые данные, сохраненные в объекте-приемнике, передаются на шину приемника, в ответ на удаленный запрос на шине приемника.

Рекомендации по записи в регистры в режиме шлюза при передаче удаленного запроса с FRREN = 1.

Обмен запрос – данные происходят в данном случае между стандартным объектом сообщения одного узла и объектом-приемником шлюза другого узла. Но при этом данные для ответа на запрос в шлюзовый объект-приемник поступают по шлюзу от объекта-источника. При получении удаленного запроса от объекта сообщения объектом-приемником флаг TXRQ устанавливается не у самого объекта-приемника, а у объекта-источника, благодаря установленному биту FRREN и битовому полю CUR (указывает на объект-источник) объекта-приемника. Данные из MODATAL и MODATAN объекта-источника копируются в MODATAL и MODATAN объекта-приемника (установлен бит DATC регистра MOFCR объекта-источника), вследствие чего автоматически устанавливается бит TXRQ регистра MOCTR объекта-приемника (установлен бит GDFS объекта-источника шлюза), и осуществляется передача сообщения данных (ответ на запрос) запрашивающему объекту сообщения.

После успешного приема/передачи сообщения ЦП получает уведомление о завершении операции для задания дальнейших действий, связанных с объектом сообщения.

18.11 Прерывания объектов сообщений

После сохранения принятого сообщения в объект сообщения или успешной передачи формируется соответствующее прерывание. Каждый объект сообщения может

формировать прерывания. Каждое прерывание направляется на одну из 16 выходных линий прерываний. Прерывания приема (после сохранения сообщения) также формируются после операций FIFO и шлюзовых операций. Флаги TXPND и RXPND всегда устанавливаются после успешной операции передачи/приема, независимо от состояния соответствующих флагов разрешения прерываний.

Объект сообщения может формировать FIFO прерывания. Если флаг OVIE регистра MOFCRn установлен, то формирование FIFO прерывания будет зависеть от типа объекта сообщений (см. рисунок 18.24):

- если объект сообщения является принимающим базовым объектом, то выходная линия прерываний для этого объекта определяется битовым полем TXINP регистра MOIPR;

- если объект сообщения является передающим базовым объектом, то выходная линия прерываний определяется битовым полем RXINP.

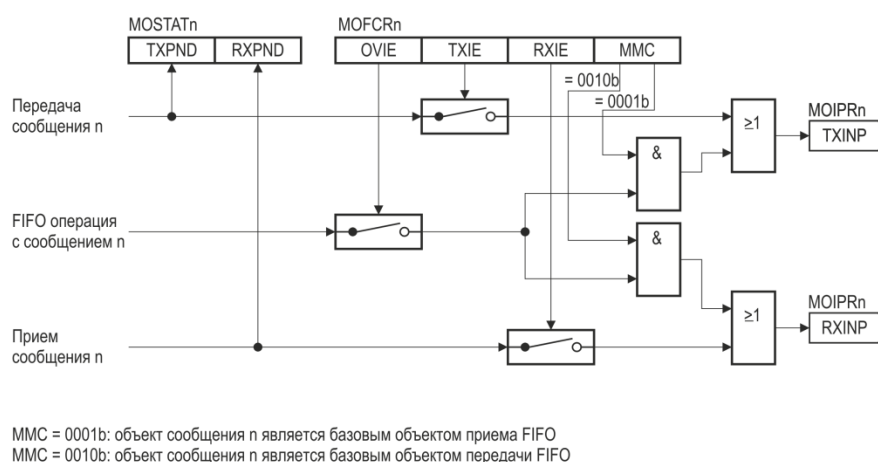


Рисунок 18.24 – Распределение прерываний

Ждущие сообщения

Когда генерируется запрос на прерывание (после приема/передачи сообщения), в одном из восьми регистров ждущих прерываний MSPNDx (x от 0 до 7) выставляется флаг ждущего сообщения. Восемь регистров образуют область из 32×8 битов – по два бита (один бит для операций приема и один бит для операций передачи) для каждого из объектов сообщений. Позиция флага ждущего сообщения определяется демультиплексорами DMUX, см. рисунки 18.25 и 18.26.

В зависимости от значения поля MPSEL регистра MCR, реализуется один из двух режимов выбора и установки флагов, ждущих сообщения:

- режим 1 в случае MPSEL = 0h;
- режим 2 в случае MPSEL = Fh.

Если нет необходимости в определении источника прерывания (прием или передача сообщения), то можно использовать любой из двух режимов, в противном случае, следует использовать второй режим.

В первом режиме установка флага ждущего сообщения происходит следующим образом:

- 7, 6 и 5 биты поля MPN выбирают регистр MSPNDx, в котором будет установлен флаг ждущего сообщения;
- пять младших бит поля MPN (на рисунке 18.25 выделены серым цветом) выбирают позицию флага (от 0 до 31), который будет установлен в выбранном регистре MSPNDx.

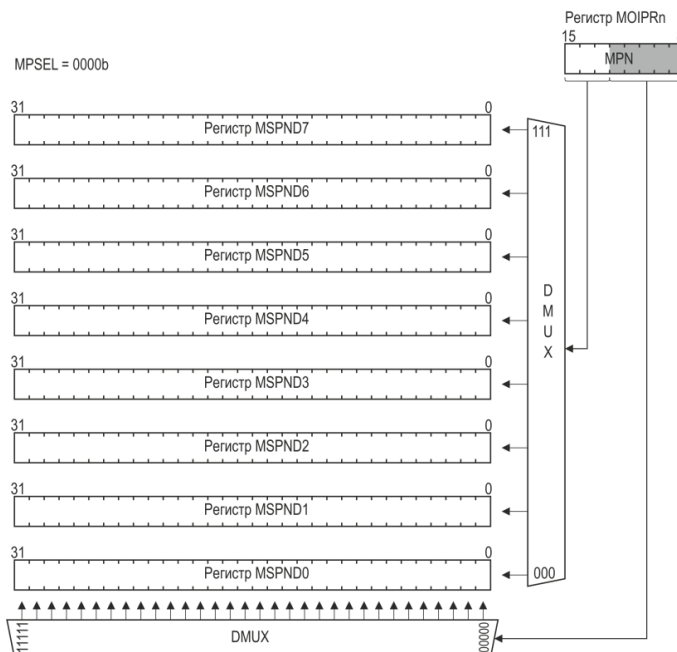


Рисунок 18.25 – Режим выбора и установки флагов при MPSEL = 0h

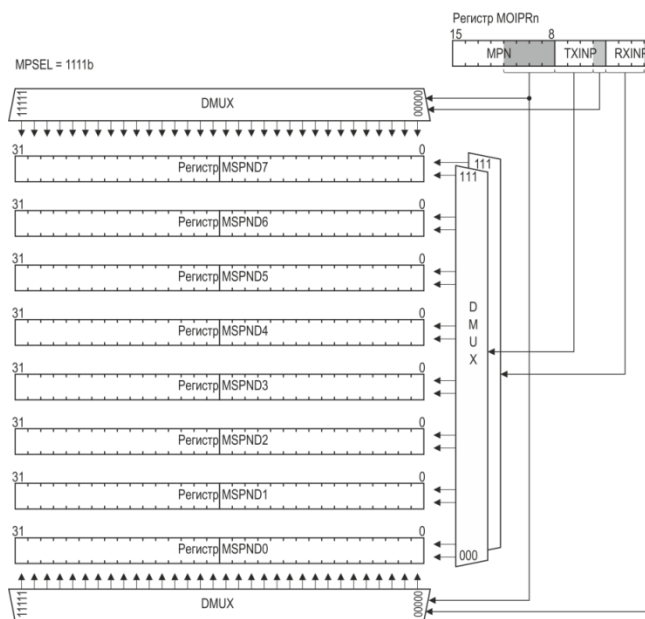


Рисунок 18.26 – Режим выбора и установки флагов при MPSEL = Fh

Во втором режиме при определении позиции флага ждущего сообщения принимаются в расчет значения поля MPN, полей RXINP (для приема) и TXINP (для передачи). При этом для флагов могут использоваться любые биты выбранного регистра MSPNDx. Установка флага ждущего сообщения происходит следующим образом:

- 3, 2 и 1 биты поля TXINP/RXINP выбирают регистр MSPNDx, в котором будет установлен флаг по окончании передачи/приема сообщения;

- четыре младших бита поля MPN (на рисунке 18.26 выделены серым цветом) совместно с нулевыми битами полей TXINP и RXINP выбирают позицию флага (от 0 до 31). Фактически нулевой бит поля TXINP/RXINP выбирает старшее или младшее слово выбранного регистра MSPNDx, а четыре бита поля MPN задают позицию в выбранном слове.

Регистры MSPNDx могут быть записаны программно. Биты, в которые записываются единицы, остаются без изменений, а биты, в которые записываются нули, очищаются. Такой механизм записи позволяет избежать конфликта между одновременной аппаратной установкой и программной очисткой битов регистра.

Каждый регистр MSPNDx связан с соответствующим регистром индекса сообщения MSIDx, который отражает позицию самого младшего бита из всех установленных в регистре MSPNDx. Регистры MSIDx доступны только для чтения и обновляются незамедлительно после изменения (как аппаратного, так и программного) содержимого соответствующих регистров MSPNDx.

Регистр маски индекса сообщения MSIMASK содержит маску для регистров MSPNDx. Только незакрытые маской биты могут обслуживаться. Регистр MSIMASK используется одновременно для всех регистров MSPNDx и соответствующих им регистров MSIDx.

18.12 Программирование контроллера CAN

Для корректной работы контроллера CAN следует соблюдать порядок программирования регистров.

Для запуска контроллера:

- записать регистр CLC;
- проверить, что сброшен бит DISR, регистр PANCTR = 00000000h и после этого записать регистр FDR.

Далее для конфигурирования узла CAN с номером x (от 0 до 3) выполнить:

- в регистре узла NCRx установить биты INIT и CCE, после чего регистры NBTRx и NPCRx станут доступны для записи и чтения, а регистр NECNTx – только для чтения;
- записать регистр NPCRx;
- записать регистр NIPRx;
- записать регистр NBTRx;
- записать регистр NFCRx (если необходимо);
- в регистре NCRx сбросить биты INIT и CCE, после чего регистры NBTRx и NPCRx будут не доступны для записи;
- распределить объекты сообщений в списки посредством регистра PANCTR.

Для корректной работы объектов сообщений регистры каждого из них должны быть проинициализированы. Для объектов, использование которых не предусматривается, достаточно записать ноль в бит MSGVAL регистра МОСТР.

Рекомендуемый порядок инициализации регистров объекта сообщения:

- установить бит DIR в регистре MOSTAT для передачи сообщения данных/приема удаленного запроса или сбросить бит DIR для приема сообщения данных/передачи удаленного запроса; установить биты TXEN0 и TXEN1 (для передачи) или RXEN (для приема) в регистре МОСТР;
- записать регистр MOFCR;
- записать регистр MOAR;
- записать регистр MOAMR (если необходимо);
- записать регистр MOFGPR (если будут использоваться FIFO структуры);
- записать регистр MOIPR;
- записать регистры MODATAL и MODATAH;
- установить бит MSGVAL корректности объекта сообщения в регистре МОСТР (для неиспользуемых объектов этот бит должен быть сброшен);
- для активирования передачи установить бит TXRQ регистра МОСТР.

19 Контроллер интерфейса Ethernet 10/100

Контроллер Ethernet 10/100 реализует стандарт IEEE 802.3. Он осуществляет прием/передачу данных по интерфейсу МП на скорости 10/100 Мбит/с. Прием/передача данных по интерфейсу МП осуществляется в/из буфера блока 32-разрядной памяти объемом 16 Кбайт.

Контроллер Ethernet 10/100 использует 32-разрядный интерфейс для связи с процессором или памятью и осуществляет обмен транслируемыми данными с процессором или памятью через 32-разрядную оперативную память объемом 16 Кбайт. Для накопления и формирования принимаемых и передаваемых пакетов имеются 2 FIFO: для приема – 36-разрядное объемом 4 Кбайта, для передачи – 40-разрядное объемом 2 Кбайта. Обмен данными с процессором или памятью осуществляется на частоте до 50 МГц. Обмен данными с устройством, работающим на физическую линию РНУ: для 100-Мбитного режима на частоте 25 МГц, для 10-Мбитного режима на частоте 2,5 МГц.

Интерфейсный блок контроллера Ethernet 10/100 содержит один контроллер прямого доступа к памяти (ПДП), имеющий два канала, которые используются для операций передачи и приема. Оба канала конкурируют за использование контроллера ПДП, реализуя циклический алгоритм обслуживания конкурирующих запросов.

Типовая передача данных в любом направлении использует кольцевой буфер в пределах назначенной для контроллера Ethernet 10/100 памяти. Кольцевой буфер для передачи определен закрытым связанным списком Tx-дескрипторов. Кольцевой буфер для приема определен закрытым связанным списком Rx-дескрипторов. Два кольцевых буфера формируются из равных 32-разрядных сегментов памяти, способных сохранять пакет максимальной длины. Эти кольцевые буферы должны быть кратными 1-Кбайтной области памяти (максимальный размер пакета данных), и должны располагаться последовательно, не затрагивая области других компонент (область дескрипторов и область кольцевых буферов) контроллера Ethernet 10/100. Предварительно проинициализированный контроллер ПДП может автономно и непрерывно заполнять/освобождать кольцевые буферы. Программное обеспечение может использовать систему прерываний или опрос флагов дескрипторов для поддержания синхронизации потоков данных между контроллером Ethernet 10/100 и процессором или памятью.

На рисунке 19.1 показан пример построения кольцевого буфера из трех дескрипторов. Адрес первого дескриптора задается регистром DMATXCTRL.

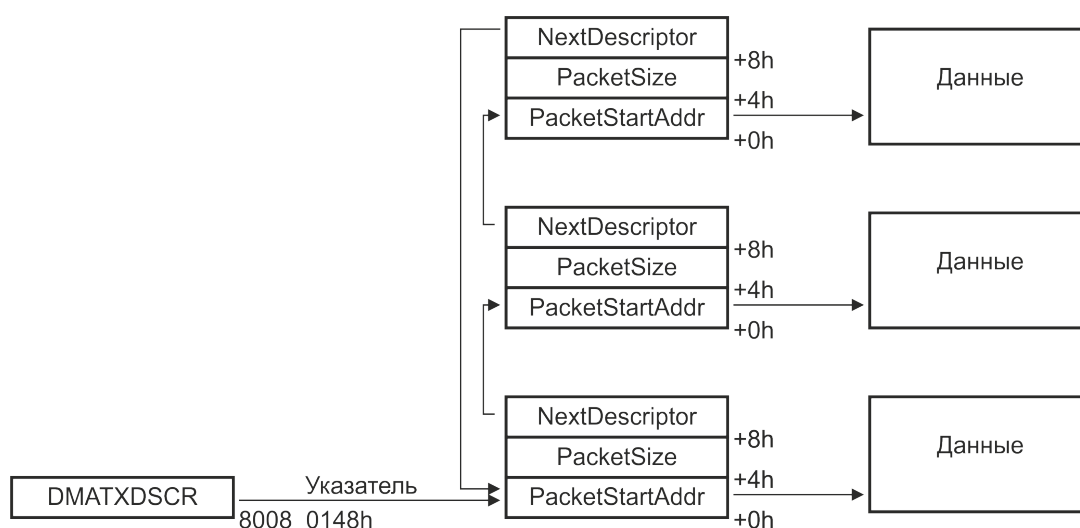


Рисунок 19.1 – Кольцевой буфер с Tx-дескрипторами

Дескриптор формируется тремя последовательно расположенными 32-разрядными ячейками ОЗУ. Назначение ячеек:

- PacketStartAddr – содержит адрес ячейки ОЗУ, где расположены данные для передачи (пакет данных);

- PacketSize – содержит размер пакета данных и флаги управления;

- NextDescriptor – содержит адрес следующего дескриптора.

Для подробного ознакомления с содержимым ячеек следует обратиться к приложению А.

Примечания

1 Порядок назначения ячеек менять нельзя.

2 Адресация оперативной памяти контроллера Ethernet 10/100 на LocalBus и контроллере ПДП различается и связана в соотношении:

$$\text{Адрес(LocalBus)} = 00710000\text{h} + ((\text{Адрес(ПДП)}) / 4).$$

Таким образом, адресу 00710000h на LocalBus соответствует адрес 00000000h ПДП, а адресу 00710001h на LocalBus соответствует адрес 00000004h ПДП.

19.1 Операции передачи

Перед передачей пакета, должна быть записана группа Tx дескрипторов, определяющих кольцевой буфер для операций передачи. Стартовые адреса начала всех сегментов должны быть 32-разрядные, сегменты равные по размерам должны быть достаточны для обработки пакета максимальной длины. Кроме того, в поле PacketSize дескриптора передачи должна быть записана длина пакета, а 31 бит EMPTYFLAG должен быть установлен, чтобы указать, что кольцевой буфер пока не содержит достоверных данных.

Четыре младших бита регистра DMAINT также должны быть установлены, чтобы специфицировать типы генерируемых прерываний.

После этого процессор должен записать в кольцевые буферы и в дескрипторы, которые связаны с этими сегментами памяти, данные для передачи одного или более пакетов. Затем в поле PacketSize записывается длина пакета, и бит EMPTYFLAG очищается, что сигнализирует контроллеру ПДП о наличии достоверных данных для передачи. Далее данные в кольцевые буферы разрешено записывать, если установлен бит EMPTYFLAG соответствующего дескриптора.

Местоположение точки входа в кольцевой буфер указывается адресом соответствующего дескриптора в регистре DMATXDSCR (см. рисунок 19.1).

Для начала передачи необходимо установить бит TXENABLE в регистре DMATXCTRL. После этого контроллер ПДП прочитает регистр DMATXDSCR и определит адрес стартового дескриптора. Далее:

- если EMPTYFLAG установлен, то дескриптор не связан с достоверными данными. В этом случае контроллер ПДП прекратит последовательную передачу пакетов, установит бит TXUNDERRUN в регистре DMAXSTAT и сбросит бит TXENABLE. Если разрешено прерывание, то оно будет сгенерировано. Для возобновления передачи следует установить бит TXENABLE;

- если EMPTYFLAG сброшен, контроллер ПДП прочитает адрес начала буферного сегмента пакета данных и определит его размер.

Далее передача пакета будет стартовать, если FIFO подтвердит контроллеру ПДП о наличии в FIFO достаточного места для приема передаваемого пакета максимальной длины.

Если передача завершена успешно, то контроллер ПДП установит EMPTYFLAG, установит флаг TXPKTSENT в регистре DMATXSTAT и, если разрешено, сгенерирует прерывание, и увеличит на единицу число, записанное в поле TX_PKT_COUNT. После этого контроллер ПДП перейдет к обработке следующего пакета.

Если произойдет ошибка в канале связи при передаче, то контроллер ПДП прекратит последовательную передачу пакетов, установит бит BUSERROR, сбросит бит TXENABLE и, если разрешено, сгенерирует.

Для последующей передачи потребуется обновление регистра DMATXSTAT для того, чтобы установить новую стартовую позицию в кольцевом буфере, и установить бит TXENABLE.

19.2 Операции приема

Перед приемом пакета должна быть записана группа Rx дескрипторов, определяющих кольцевой буфер для операций приема. Стартовые адреса начала всех сегментов должны быть 32-битные, сегменты должны быть одинаковыми по размеру и достаточными для обработки пакета максимальной длины. Поле PacketSize дескриптора приема должно быть заполнено, а бит EMPTYFLAG установлен, чтобы указать, что кольцевой буфер приема не содержит принятых пакетов.

Биты с седьмого по четвертый регистра DMAINT должны быть установлены, чтобы специфицировать типы генерируемых прерываний.

В регистр DMARXDSCR должен быть записан адрес стартового Rx дескриптора.

Для начала работы следует установить бит RXENABLE в регистре MAC1 для разрешения контроллеру ПДП обрабатывать принимаемый пакет. После этого встроенный контроллер ПДП читает регистр DMARXDSCR для определения адреса первого дескриптора, затем читает этот дескриптор. Далее:

- если EMPTYFLAG сброшен, то это означает, что предыдущий принятый пакет еще не был считан программой. В этом случае контроллер ПДП прекращает последовательный прием пакетов, устанавливает бит RXOVERFLOW в регистре DMARXSTAT, сбрасывает бит RXENABLE, и если разрешено прерывание, то оно будет сгенерировано. Любой последующий прием будет возможен только после обновления регистра DMARXDSCR и установки бита RXENABLE;

- если EMPTYFLAG установлен, то это означает, что область памяти доступна для сохранения пакета.

Контроллер ПДП начнет прием пакета, если FIFO сообщит контроллеру о наличии принятого пакета. Если прием пакета закончен успешно, то контроллер ПДП запишет значение числа принятых байт в поле PACKET_SIZE дескриптора приема, сбросит бит EMPTYFLAG и установит бит RXPKTREC в регистре DMARXSTAT. Если разрешено, будет сгенерировано прерывание и увеличено на единицу значение в счетчике RX_PKT_COUNT.

Если FIFO сообщит, что имеется принятый пакет, то контроллер ПДП начнет передачу этого пакета в следующий сегмент кольцевого буфера.

Программное обеспечение должно обработать прерывание считыванием пакета из кольцевого буфера приема с последующей установкой бита EMPTYFLAG дескриптора, помечая этот сегмент кольцевого буфера как доступный для сохранения следующего пакета.

Если произошла ошибка при приеме, то контроллер ПДП прекращает последовательную обработку принимаемых пакетов, устанавливает бит BUSERROR в регистре DMARXSTAT, сбрасывает бит RXENABLE и, если разрешено, генерирует прерывание.

Для следующего приема следует обновить регистр DMARXDSCR, записью правильной стартовой позиции кольцевого буфера, и установить бит RXENABLE.

20 Контроллер интерфейса USB OTG

USB (Universal Serial Bus) – последовательная шина, которая позволяет передавать данные между хост-компьютером и различными периферийными устройствами. Наиболее важным свойством USB является возможность подключения и отключения периферийных устройств без выключения всей системы (один из классов механизма «Plug and Play»). USB-хост имеет соединения типа точка-точка с периферийными устройствами через поярусную топологию типа «звезда», в узлах которой содержится USB-устройство, называемое хабом. USB позволяет соединить до 127 устройств.

20.1 Функции хоста

Главные функции хоста:

- определение подключения/отключения USB-устройств;
- установление нисходящего соединения (от хаба до периферийного устройства);
- посылка маркеров начала фрейма (SOF);
- проведение операций приостановки/возобновления работы устройств;
- обработка данных во фрейме или пакете, если их больше, чем было запрошено (packet или frame babbling);
- определение потери соединения или активности на шине;
- упаковка пакета во фреймы и передача данных из/в системную память.

Хост планирует и управляет передачей данных. Как правило, передачи должны всегда инициироваться хостом, и все передачи должны производиться с использованием до трех типов структур данных – Setup-/OUT-/IN-токенов, Data-пакетов и подтверждений Handshake.

Токены обеспечивают USB-устройства информацией о направлении и типе передачи, адресе USB-устройства и адресе конечной точки (Endpoint), к которой (или от которой) обеспечивается передача.

Data-пакеты – это данные, которые принимает или передает хост. Данные разбиваются на пакеты и дополняются соответствующей контрольной суммой.

Подтверждения Handshake, отправленные к источнику передачи, показывают, была ли передача успешной или нет.

USB протокол поддерживает четыре типа передач:

- управляющие передачи (control transfer);
- передачи массивов данных (bulk transfer);
- передачи по прерываниям (interrupt transfer);
- изохронные передачи (isochronous transfer).

Управляющие передачи и передачи массивов данных являются неперiodическими, а передачи по прерываниям и изохронные передачи – периодическими. Для каждого из этого типа передач, поток данных может быть как от хоста к устройству (OUT), так и от устройства к хосту (IN). Для каждого типа передачи, в устройстве имеется буфер, называемый Endpoint, для хранения байт данных для транзакции. Каждое устройство должно иметь контрольный буфер Endpoint, который отвечает на контрольные запросы, передавая хосту конфигурационную информацию об устройстве.

Управляющие передачи

Эти неперiodические посылки данных следует использовать для пересылки информации о конфигурации, пересылки команд и статуса между клиентской программой и USB-устройством. Контрольные посылки делятся на три этапа:

- передача команды (Setup stage);
- передача с подтверждением одного или нескольких пакетов данных (Data stage);

- прием с подтверждением пустого пакета данных, подтверждающего успешную транзакцию (Status stage).

Управляющие передачи показаны на рисунке 20.1.

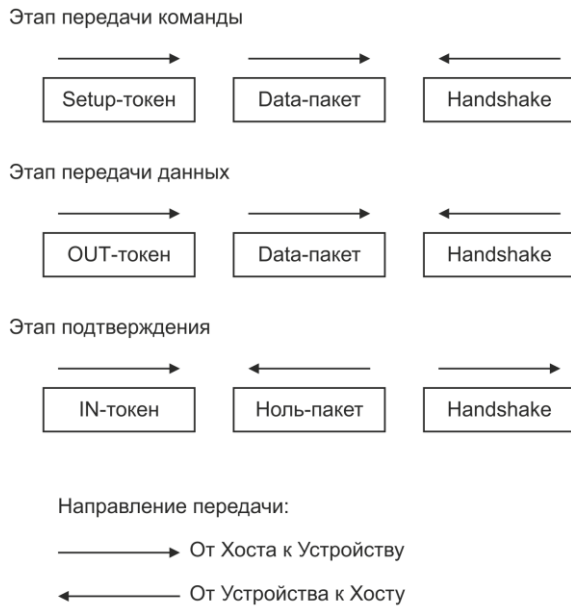


Рисунок 20.1 – Управляющие передачи

Передачи массивов данных

Этот тип передачи используется устройствами, к которым нет особого требования по скорости передачи. Классический пример использования этого типа передачи – принтер. Никаких проблем не возникнет, если передачу данных осуществлять на малой скорости. Каждая посылка дополняется контрольной суммой для гарантии правильности передачи данных.

Передачи по прерываниям

Этот тип используется для пересылки небольших данных между устройством и клиентской программой, для которых задается период опроса в конфигурации устройства. Пример использования – устройства ввода-вывода (клавиатура и т. п.), для которых устанавливается определенное время опроса для приема данных о нажатии клавиш. Передачи по прерываниям показаны на рисунке 20.2.

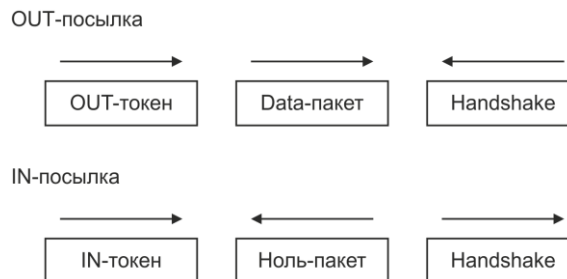


Рисунок 20.2 – Передачи по прерываниям

Изохронные передачи

Этот тип передачи применяется в приложениях реального времени, для которых важны минимальное время доставки данных, но не важна целостность данных. Посылки при этом не требуют подтверждения ответной стороной о приеме данных. Такой тип

посылок применяется при передаче аудио/видео потоковых данных и т.п. Изохронные передачи показаны на рисунке 20.3.

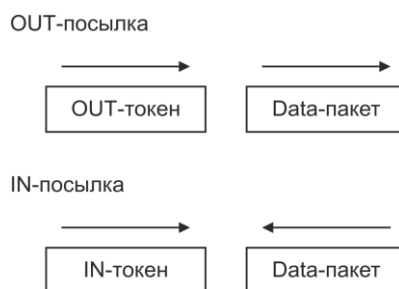


Рисунок 20.3 – Изохронные передачи

20.2 Работа хоста

Определение подключения

Когда устройство подключается к хосту, состояние линии отражает скорость работы устройства. Если состояние линии 1, то подключенное устройство работает в режиме High speed или Full speed. Хост-контроллер первым детектирует это состояние линии. Затем следующие события происходят в соответствии с программным обеспечением (ПО):

1 Программное обеспечение разрешает прерывания USB-хоста в регистре USB_CMD_STS_INTR.

2 Биты PORTCONNECT и PORTCONCHNG регистра PORT_STS устанавливаются аппаратно.

3 Хост-контроллер посылает прерывание.

4 ПО определяет источник прерывания и очищает бит PORTCONCHNG.

5 ПО устанавливает бит PORTRESET регистра PORT_STS для сброса USB линии. Это соответствует протоколу сброса USB-устройств.

6 Если устройство включено после сброса линии, бит PORTENABLE регистра PORT_STS устанавливается аппаратно.

7 Определение отключения. Когда в течение определенного времени возникает тайм-аут, контроллер хоста определяет это как отключение устройства.

Управляющие передачи

Для выполнения управляющей передачи следует (программно):

- в регистр BUFF_START_ADDR записать стартовый адрес буфера для текущей контрольной передачи;

- в регистр TBT записать общее количество байт для передачи;

- в регистре USB_PKT_FLD_B задать поле MPS (максимальный размер пакета);

- в регистре USB_PKT_FLD_A установить бит ENTRANSFER (разрешение передачи), задать поле DEV_ADDR (адрес устройства), задать поле ENDPOINTTYPE = 00b (тип контрольной точки – Control), задать поле PID = 10b (метка Setup) и поле EP_ADDR = 0h (адрес конечной точки).

IN-посылки (от устройства к хосту)

Для выполнения IN-передачи следует (программно):

- в регистр BUFF_START_ADDR записать стартовый адрес буфера для текущей IN-передачи;

- в регистр TBT записать общее количество байт для IN-передачи;

- в регистре USB_PKT_FLD_B задать поле MPS;

- в регистре USB_PKT_FLD_A установить бит ENTRANSFER, задать поле DEV_ADDR, задать поле ENDPOINTTYPE (Iso, Bulk, Interrupt), задать поле PID = 01b (метка In) и поле EP_ADDR.

OUT-посылки (от хоста к устройству)

Для выполнения OUT-передачи следует (программно):

- в регистр BUFF_START_ADDR записать стартовый адрес буфера для текущей контрольной передачи;
- в регистр TBT записать общее количество байт для OUT-передачи;
- в регистре USB_PKT_FLD_B задать поле MPS;
- в регистре USB_PKT_FLD_A установить бит ENTRANSFER, задать поле DEV_ADDR, задать поле ENDPOINTTYPE (Iso, Bulk, Interrupt), задать поле PID = 00b (метка Out) и поле EP_ADDR.

20.3 Работа устройства

Работа при управляющих передачах

Возможны четыре типа управляющих передач:

- управляющая запись с передачей данных;
- управляющая запись без передачи данных;
- управляющее чтение с передачей данных;
- управляющее чтение без передачи данных.

Этап передачи команды (Setup stage)

Передача команды – это Setup-токен с восемью байтами управляющих данных, которые содержат запрашиваемую информацию от хоста к устройству (см. рисунок 20.4).

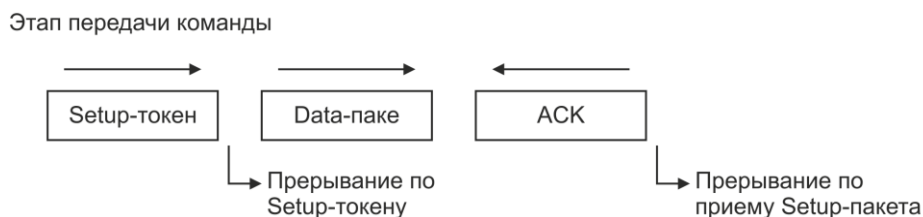


Рисунок 20.4 – Передача команды

Для этого этапа доступны два прерывания: по приему Setup-токена и по приему Setup-пакета (после приема 8 байт данных без каких-либо ошибок).

ЦП должен сбросить биты соответствующих прерываний, если они возникли, обработать принятые на этом этапе данные и подготовиться к опциональному этапу передачи данных (Data stage). Четыре регистра SETUP1_0, SETUP3_2, SETUP5_4 и SETUP7_6 хранят данные, принятые контрольной точкой на этапе передачи команды.

Этап передачи данных (Data stage)

В случае управляющей записи с передачей данных, когда происходит OUT-передача, принятые данные записываются в буфер контрольной точки, генерируется прерывание и устанавливается флаг DATAPKTRECINT в регистре USB_EP_x_IRQ_STAT (x – номер контрольной точки). ЦП должен обработать принятые данные, количество байт которых указывается в регистре OUT_TRNSFR_CNT. Передача данных показана на рисунке 20.5.

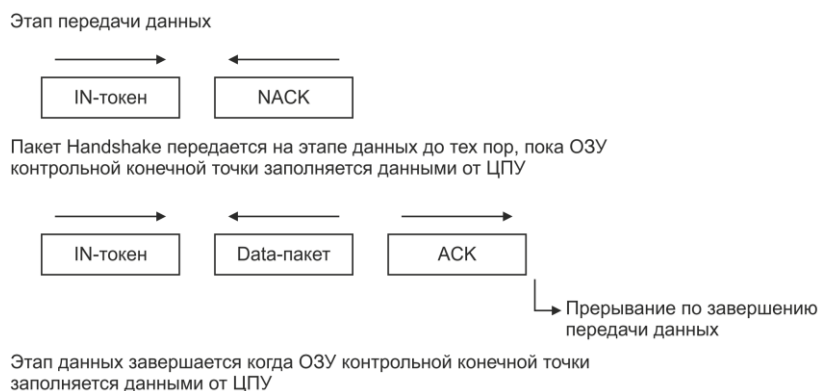


Рисунок 20.5 – Передача данных

В случае управляющего чтения с передачей данных, когда происходит IN-передача, ЦП должен загрузить буфер контрольной точки данными для передачи хосту, затем загрузить регистр `IN_TRNSFR_CNT` значением количества передаваемых данных. Запись значения количества байт в регистр разрешает USB-устройству передать данные хосту на приходящий IN-токен. После, ЦП может ожидать прерывание, которое будет сигнализировать о завершении этапа передачи данных и установки флага `DATAPKTTRINT` в регистре `USB_EP_x_IRQ_STAT`.

В случае если требуется послать `STALL-Handshake` хосту, ЦП должен установить бит `STALL` в регистре `CEP_CTRL_STAT`.

Этап подтверждения (Status stage)

После того, как микроконтроллер завершил этап передачи данных, бит `NAK_CLEAR` должен быть сброшен (см. рисунок 20.6). Сбрасывая этот бит, ЦП позволяет хосту перейти к следующей управляющей передаче. Этот бит устанавливается USB-устройством после приема `Setup`-токена.

В случае если требуется послать `STALL-Handshake` хосту, ЦП должен установить бит `STALL` и сбросить бит `NAK_CLEAR`.

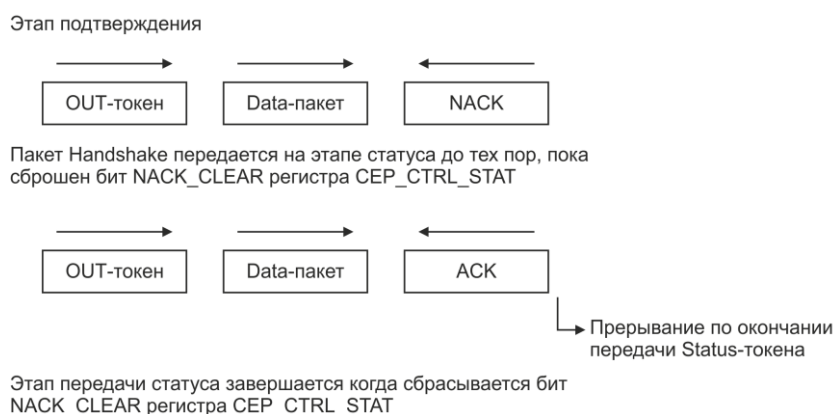


Рисунок 20.6 – Подтверждение

Работа при приеме IN-токенов

Данные для любой IN-посылки записываются в буфер микроконтроллером в момент приема устройством IN-токена от хоста. Есть три различных режима подтверждения данных, при которых они будут посланы хосту:

- автоподтверждения;
- подтверждения вручную;
- подтверждения «на лету».

Режим автоподтверждения

Если конечная точка настроена на работу в режиме автоподтверждения, то она будет отсылать количество байт данных, равное значению EP_MPS регистра USB_EP_x_MPS. Контроллер конечной точки будет ждать, пока количество байт данных в буфере станет равным EP_MPS, после чего подтвердит отправку. Если микроконтроллеру требуется отправить короткий пакет в конце передачи, следует установить бит PKTEND регистра USB_EP_x_RSP_SC. Когда бит будет установлен, оставшееся количество байт данных будет отправлено хосту на следующий IN-токен.

Этот режим требует минимум вмешательства центрального процессора, т. к. большинство работы будет сделано на аппаратном уровне. Режим может быть выбран, когда размер пересылаемых данных в хост всегда равен значению EP_MPS. Более подробно действия контроллера точки отображены в таблице 20.1.

Таблица 20.1 – Действия Endpoint в режиме автоподтверждения

| Бит PacketEnd | Количество доступных данных в буфере | Действие |
|---------------|--------------------------------------|--|
| 0 | < EP_MPS | Ответ NAK |
| 0 | ≥ EP_MPS | Ответ количеством данных, равных EP_MPS |
| 1 | < EP_MPS | Ответ количеством данных, доступных в буфере |
| 1 | ≥ EP_MPS | Ответ количеством данных, равных EP_MPS |

Режим подтверждения вручную

Если конечная точка настроена на работу в режиме подтверждения вручную, то она будет отсылать данные тогда, когда это будет разрешено микроконтроллером. ЦП сначала записывает данные в буфер, а затем записывает количество байт данных (значение EP_CNT) в регистр USB_EP_x_CNT. Как только будет записано верное количество передаваемых байт, данные будут отправлены хосту на следующий IN-токен.

Этот режим требует вмешательства ЦП каждую посылку и может быть использован, если количество передаваемых байт каждый раз непостоянное и решается ЦП. Более подробно действия контроллера точки отображены в таблице 20.2.

Таблица 20.2 – Действия Endpoint в режиме подтверждения вручную

| EP_CNT записан | Количество доступных данных в буфере | Действие |
|----------------|--------------------------------------|---|
| Нет | - | Ответ NAK |
| Да | EP_CNT | Ответ количеством данных, равных EP_CNT |

Режим подтверждения «на лету»

Простейший режим работы, где не требуется процедуры подтверждения. Буфер заполняется центральным процессором. Если хост отправил IN-токен, данные в буфере автоматически подтверждаются и отправляются хосту. Если количество данных в буфере превышает размер одного пакета EP_MPS, контроллер точки автоматически упаковывает данные в цепочку пакетов, размером не более EP_MPS, и отправляет их хосту.

Этот режим требует минимум вмешательства ЦП и лучше всего подходит для изохронных передач, где скорость доставки данных более важна, чем размер пакета. Более подробно действия контроллера точки отображены в таблице 20.3.

Таблица 20.3 – Действия Endpoint в режиме подтверждения «на лету»

| Количество доступных данных в буфере | Количество отправленных данных |
|--------------------------------------|--|
| $\geq EP_MPS$ | Ответ количеством данных, равных EP_MPS |
| $< EP_MPS$ | Ответ количеством данных, доступных в буфере |

20.4 USB OTG

USB OTG – это портативные устройства, которые используют единственное гнездо типа Micro-AB для работы, как в режиме хоста, так и в режиме периферийного устройства. OTG-устройства всегда могут работать, как стандартное USB-устройство при подключении к стандартному USB-хосту.

OTG-устройства так же могут соединяться друг с другом. Режим работы (хост или устройство) определяется OTG-кабелем либо протоколом HNP (Host Negotiation Protocol).

Режимы работы соединяемых устройств могут быть изменены программно без физического «переворачивания» OTG-кабеля.

Описание работы

После подключения OTG-кабеля через вывод ID определяется тип OTG-устройства. При неподключенном кабеле вывод ID подтянут внутренним резистором до логической единицы, что определяет тип устройства B-device (периферийное устройство). Если в OTG-гнездо устройства Micro-AB подключена вилка типа Micro-A, то вывод ID переводится в логический ноль, что указывает на хост. Если подключена вилка типа Micro-B, то вывод ID не меняет своего логического состояния (остается в единице), что указывает на периферийное устройство.

Изменение вывода ID вызывает прерывание и установку флага HANDOFF_INTR в регистре OTG_IRQ_STAT.

Примечание – Для разрешения прерывания, следует установить соответствующий бит в регистре OTG_IRQ_EN.

Тип подключенной вилки Micro-A или Micro-B отображается в битах A_DEVICE и B_DEVICE.

В зависимости от типа (хост или устройство), следует установить бит B_BUS_REQ или бит A_BUS_REQ.

Если устройство является B-device, то после установки бита B_BUS_REQ оно начинает генерировать SRP-запросы (включается на некоторое время подтяжка по линии D+, после чего анализируется шина питания VBUS). Хост, видя запрос SRP, должен подать питание на устройство. После подачи питания хостом на устройство по SRP запросу, возникнет прерывание и установится флаг DEV_SYNCCHNG, а также бит DEVICE, что говорит о том, что устройство перешло в состояние «b_peripheral» (согласно диаграмме состояний B-device). По этому прерыванию можно настраивать устройство на работу.

Если устройство является A-device, то при SRP-запросе от B-device возникнет прерывание и установится флаг SRPDETINTR. В этом прерывании следует установить бит A_BUS_REQ. После установки этого бита контроллер с помощью управляющего сигнала USB_DRVVBUS начнет подавать питание на USB-разъем. После этого подключенное периферийное устройство будет готово к дальнейшей работе, а хост перейдет в состояние «a_host» (согласно диаграмме состояний A-device), установив бит HOST. После этого будет сгенерировано прерывание и установлен флаг HOSTSYNCCHNG.

USB_DRVVBUS (альтернативная функция вывода E.7 микроконтроллера) активируется аппаратно при работе в режиме хоста и управляет подачей питания на USB-разъем. В режиме устройства вывод всегда удерживается в состоянии логического нуля. В приложении Д приводится схема подключения USB разъема micro_AB с питанием, управляемым сигналом USB_DRVVBUS.

При подключении двух OTG-устройств друг к другу, если контроллер, работая в режиме B-device, принимает от хоста Set_Feature (протокол HNP), то следует установить бит V_HNP_EN. Аналогично, если контроллер работает в режиме A-device, после отправления запроса Set_Feature (протокол HNP) к B-device, должен быть установлен бит A_HNP_EN.

Если в режиме A-device больше не требуется использование линии USB, то следует установить бит A_BUS_DROP.

Чтобы послать запрос SUSPEND на линию USB, следует установить бит SUSPEND_EN.

21 Блок АЦП

Блок АЦП объединяет 12 модулей АЦП последовательного приближения (архитектура SAR), ОЗУ результатов измерений и схему управления прерываниями. Структурная схема блока АЦП показана на рисунке 21.1.

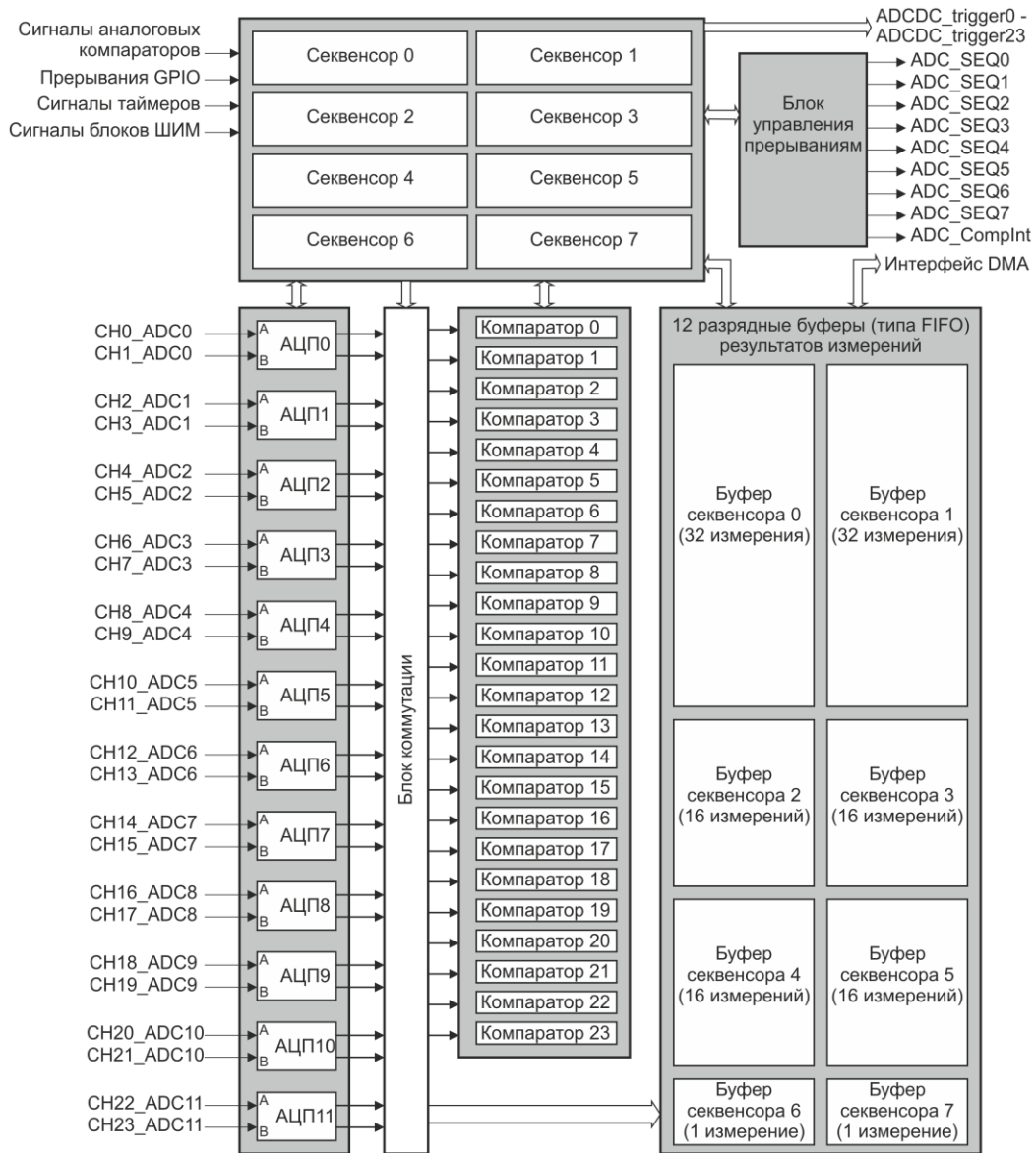


Рисунок 21.1 – Структурная схема блока АЦП

В блок АЦП входят:

- 12 двухканальных (каждый может работать в дифференциальном режиме) модулей АЦП разрядностью 12/10 бит и рабочей частотой 12/24 МГц;
- восемь секвенсоров, каждый из которых позволяет независимо произвести запуск измерений по всем 24 каналам АЦП и по окончании выставить прерывание;
- 24 независимых цифровых компаратора, отслеживающих и сравнивающих измерения с пороговыми значениями для формирования прерываний и сигналов управления другими блоками микроконтроллера;
- восемь буферов результатов измерений (каждый организован по типу FIFO);
- блок управления прерываниями.

Блок АЦП имеет 24 канала – по два (А и В) на каждый модуль АЦП. Соответствие номеров каналов и входов модулей АЦП указано в таблице 21.1.

Примечание – 23 канал не способен работать в одиночном режиме. Он может работать только в дифференциальном режиме А - В (Канал 22 – Канал 23).

Таблица 21.1 – Соответствие номеров каналов контроллера АЦП и входов его модулей

| Номер канала | Вход модуля АЦП | Модуль | Номер канала | Вход модуля АЦП | Модуль |
|--------------|-----------------|--------|--------------|-----------------|--------|
| 0 | А | АЦП0 | 12 | А | АЦП6 |
| 1 | В | | 13 | В | |
| 2 | А | АЦП1 | 14 | А | АЦП7 |
| 3 | В | | 15 | В | |
| 4 | А | АЦП2 | 16 | А | АЦП8 |
| 5 | В | | 17 | В | |
| 6 | А | АЦП3 | 18 | А | АЦП9 |
| 7 | В | | 19 | В | |
| 8 | А | АЦП4 | 20 | А | АЦП10 |
| 9 | В | | 21 | В | |
| 10 | А | АЦП5 | 22 | А | АЦП11 |
| 11 | В | | 23 | В | |

Настройка тактирования контроллера АЦП и его модулей осуществляется посредством регистров ADC_CTRL0, ADC_CTRL1 и ADC_CTRL2.

Для правильной работы необходимо обеспечить тактирование модулей АЦП:

- в 10-разрядном режиме частотой 24 МГц (диапазон от 6 до 28,8 МГц);
- в 12-разрядном режиме частотой 12 МГц (допустимый диапазон от 5 до 14,4 МГц).

21.1 Секвенсор

Секвенсор представляет собой управляющий блок, позволяющий разгрузить процессор от управления модулями АЦП. Секвенсор управляет запуском модулей АЦП, обработкой полученных результатов измерений и генерацией прерываний. В состав блока АЦП входят восемь секвенсоров. Структурная схема секвенсора представлена на рисунке 21.2.

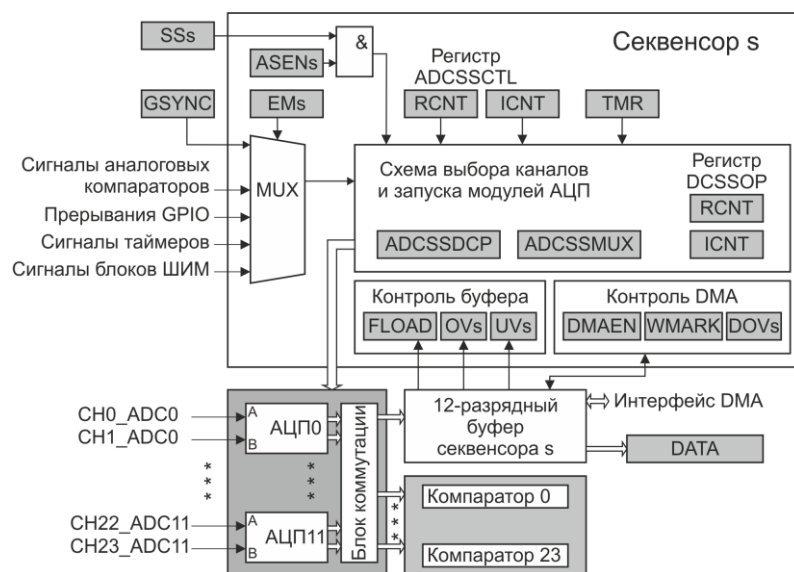


Рисунок 21.2 – Структурная схема секвенсора

Каждый секвенсор может независимо запускаться по одному из событий, которое выбирается полем EMs регистра EMUX:

- установка бита GSYNC регистра PSSI;
- сигнал от одного из блоков аналоговых компараторов;
- сигнал от блока таймеров;
- сигнал от одного из блоков ШИМ;
- сигнал прерывания GPIO.

После запуска по событию секвенсор формирует сигнал запуска модуля АЦП. Задержка запуска модуля задается полем TMR регистра TMR секвенсора. По окончании измерения секвенсор может автоматически перезапустить модуль АЦП. Перезапуск может выполняться до 256 раз (поле RCNT регистра CTL) с заданной периодичностью (поле TMR). Также секвенсор может быть запрограммирован на циклический запуск модулей АЦП без остановки (в этом случае значение поля RCNT не важно).

Секвенсор может одновременно запустить измерения на всех 24 каналах блока АЦП. Если выбранный канал соответствует входу А модуля АЦП, то измерение будет запущено только для этого входа. В случае если выбранный канал соответствует входу В, то сначала будет запущено измерение по входу А, а затем по входу В (реализовано на аппаратном уровне и влияет на время измерений).

Выбор каналов для измерений осуществляется посредством регистра MUX. Результаты измерений сохраняются в буфере результатов секвенсора. Контроль состояния буфера осуществляется посредством регистров FSTAT, OSTAT и USTAT.

Секвенсор может генерировать прерывания с заданной периодичностью (по умолчанию – в конце каждого измерения). По завершении каждого измерения инкрементируется счетчик запусков, и как только он достигает значения, заданного полем ICNT регистра CTL, генерируется прерывание. Сброс счетчика запусков происходит при запуске секвенсора по событию (если ICNT \neq 01h) или программно – при каждой записи в поле ICNT.

Прерывания, генерируемые секвенсором и передаваемые в контроллер NVIC, могут использоваться блоком DMA для аппаратного копирования содержимого буфера результатов АЦП в ОЗУ.

Для каждого секвенсора выделена одна из восьми линий прерываний IS0 – IS7.

Флаги INRs и INs прерываний секвенсоров устанавливаются в регистрах RIS и ISC (маскированные прерывания). Маска прерываний задается регистром IM. Флаги сбрасываются записью единиц в соответствующие биты регистра ISC.

Диаграмма работы секвенсора показана на рисунке 21.3.

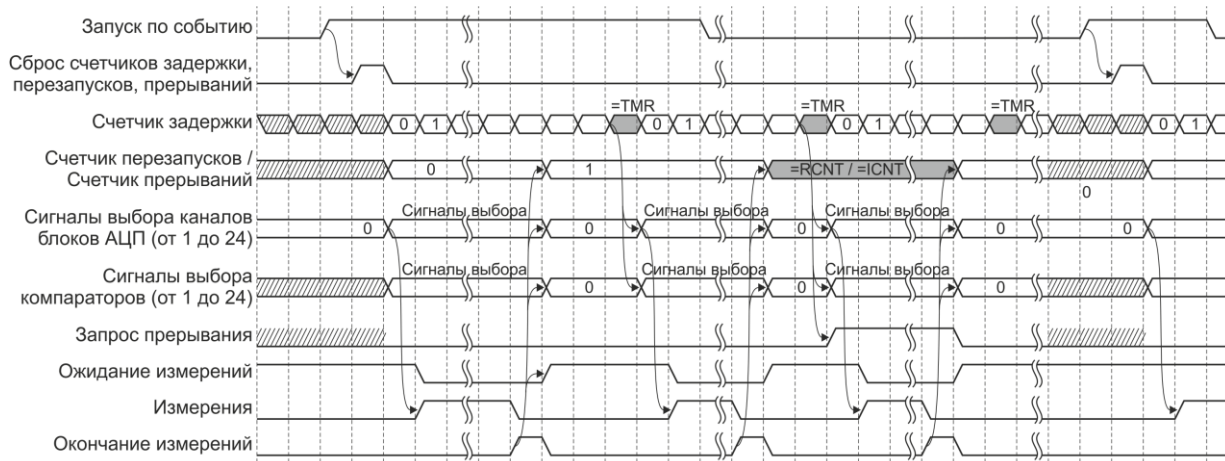


Рисунок 21.3 – Диаграмма работы секвенсора

Проведение измерений

1 Если сигнал запуска получает только один секвенсор, то он запускается. Если модуль настроен на проведение нескольких измерений, то по окончании измерения секвенсор запускается снова и запуск происходит по истечении времени, определяемого задержкой TMR (незамедлительно, если TMR = 00000h).

Если в течение преобразования возникают сигналы запуска других секвенсоров, то эти секвенсоры переходят в состояние ожидания.

2 Если несколько секвенсоров получают сигнал запуска одновременно, то сначала запускается секвенсор с наименьшим порядковым номером. Остальные секвенсоры переходят в состояние ожидания (см. рисунок 21.4). Если в течение преобразования какой-либо из незапущенных ранее секвенсоров получает сигнал запуска, он также переходит в состояние ожидания.



Рисунок 21.4 – Запуск нескольких секвенсоров

3 На момент окончания измерения происходит проверка состояния всех секвенсоров и незамедлительный синхронный запуск секвенсоров, которые находятся в состоянии ожидания.

При этом секвенсор, закончивший преобразование (измерение), в этот момент не запускается (например, если работает в непрерывном цикле), а переходит в состояние ожидания.

Примечание – Рекомендуется использовать только один секвенсор либо несколько, но работающих синхронно (одинаковые события запуска, одинаковые задержки перезапуска измерений).

21.2 Модуль АЦП

Структурная схема двухканального модуля АЦП показана на рисунке 21.5.

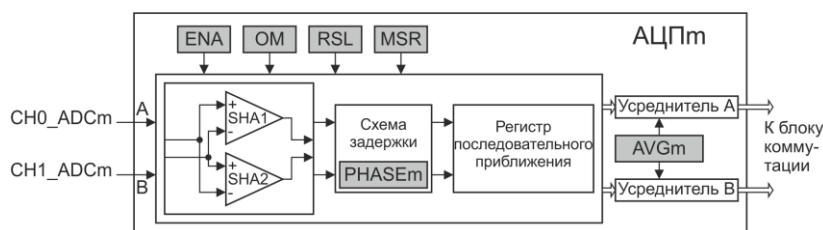


Рисунок 21.5 – Модуль АЦП

В зависимости от выбранных каналов для преобразования могут быть задействованы от одного до 12 модулей АЦП. Модуль АЦП выбирается автоматически, если секвенсором выбран хотя бы один из его каналов. Поскольку один канал может быть выбран несколькими секвенсорами, то на один модуль АЦП могут приходиться одновременно несколько сигналов запусков. Все эти сигналы объединены по ИЛИ. Не выбранные модули АЦП не функционируют. Для разрешения работы модуля АЦП следует установить бит ENA в регистре PPM, а также записать единицы в первые два бита поля OM.

Режимы работы

Модуль АЦП может функционировать как в однополярном, так и в дифференциальном режиме работы. Выбор режима задается полем ОМ регистра PPM.

Диапазон входных напряжений в однополярном режиме от 0 до 1,5 В, в дифференциальном от 0,15 до 1,65 В.

В однополярном (см. рисунок 21.6) режиме сигналы с входов А и В через схему задержки передаются к регистру последовательного приближения (РПП) напрямую.

Далее:

- если для измерения был выбран канал А, то на выходе РПП будет результат измерения только канала А;

- если для измерения был выбран канал В или оба канала, то сначала будет выполнено измерение канала А, а затем канала В, и на выходе РПП будут результаты измерения обоих каналов (общее время измерения, соответственно, увеличивается вдвое).

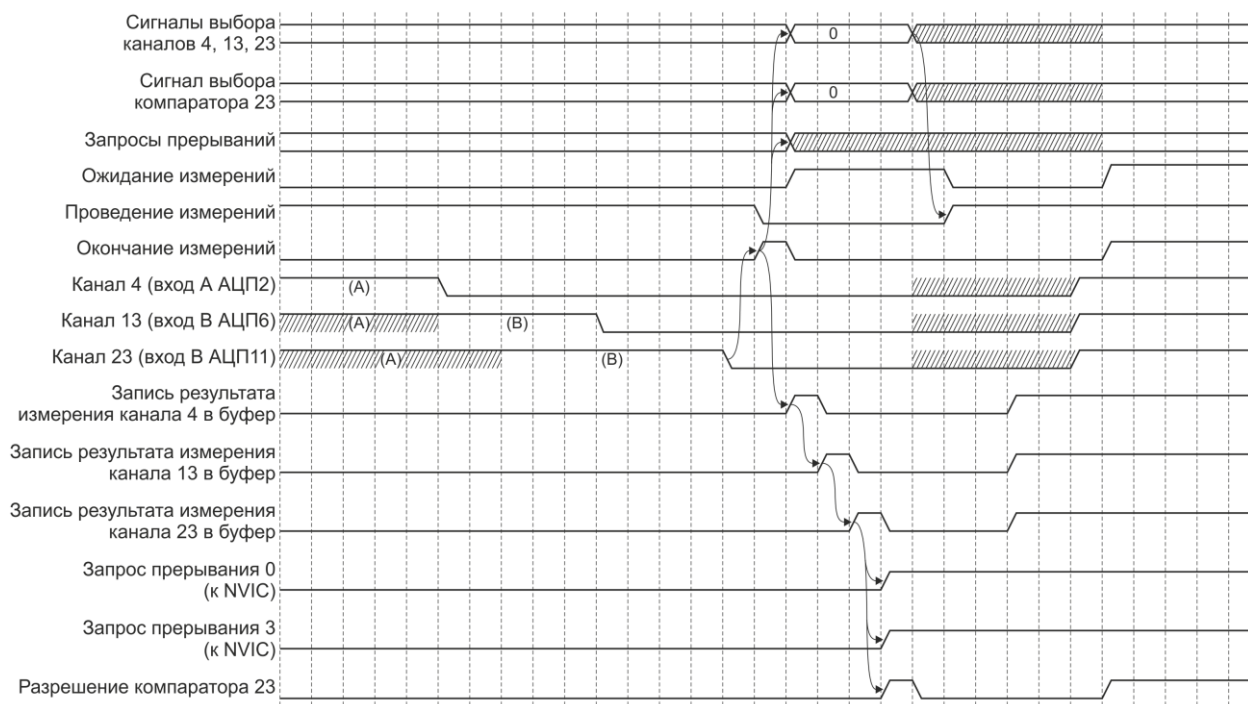


Рисунок 21.6 – Функционирование модулей АЦП в однополярном режиме

В дифференциальном режиме дифференциальный усилитель (SHA) усиливает разность напряжений между входами А и В. Усиленное таким образом значение через схему задержки передается к РПП. В составе модуля АЦП функционируют два усилителя SHA1 и SHA1 (см. рисунок 21.5). На выходе первого формируется значение «А – В», а на выходе второго – «В – А». Дифференциальный режим не может быть включен для двух блоков усилителей одновременно.

Модуль АЦП имеет схему фазовой задержки, которая позволяет задержать начало преобразования входных сигналов после того как модуль АЦП получил сигнал запуска от секвенсора (см. рисунок 21.7). Задержка включается одновременно для обоих каналов и программируется посредством соответствующего регистра SPCn.



Рисунок 21.7 – Фазовая задержка

Результат преобразования каждого канала передается на схему усреднения. В составе модуля АЦП два усреднителя, для которых задается общий параметр AVGM. Если AVGM = 0, то усреднители не используются; в противном случае модуль АЦП выполняет количество измерений, задаваемое полем AVGM, усреднители суммируют результаты и вычисляют усредненное значение для каждого из каналов.

Усреднители сбрасываются при поступлении сигнала запуска от секвенсора.

Результаты измерений с выходов модуля АЦП записываются в буферы секвенсоров и/или передаются на компараторы.

Расчет времени одного измерения

Временем одного измерения считается время от момента запуска модулей АЦП до момента появления результатов измерения на выходах модуля, имеющего наибольшую фазовую задержку и/или наибольшее количество перезапусков для усреднения результатов. Время T одного измерения модуля АЦП рассчитывается по формуле

$$T = R \times A \times (T_{adc} + 1t + tsysclk \times (P_{adc} + k)), \quad (21.1)$$

где R – количество перезапусков (R = 1 при RCNT = 00h);

A – количество измерений при усреднении (A = 1 при AVGM = 0h);

T_{adc} – время преобразования для:

- одного канала (A) равняется семи тактам рабочей частоты (например, для 24 МГц параметр T_{adc} = 292 нс);

- двух каналов (сначала A, затем B) равняется 14 тактам рабочей частоты (для 24 МГц параметр T_{adc} = 584 нс);

1t – один такт рабочей частоты;

tsysclk – длительность такта системной частоты SysCLK, нс;

P_{adc} – фазовая задержка (P_{adc} = 0 при PHASEm = 0h);

k – количество тактов системной частоты (4 для однократного режима запуска и 12 для циклического).

21.3 Буфер результатов измерений

Каждый секвенсор имеет собственный 12-разрядный буфер результатов измерений, организованный по типу FIFO. В каждой ячейке буфера может храниться результат одного измерения.

Количество ячеек в буфере:

- 32 для секвенсоров 0 и 1;

- 16 для секвенсоров 2 – 5;

- одна для секвенсоров 6 и 7.

Результаты измерений модулей АЦП записываются в буферы секвенсоров в порядке возрастания номеров каналов, т. е. первым будет записан результат канала А модуля АЦП0, а последним – результат канала В модуля АЦП11. Каналы, на которые не поступили сигналы запуска, пропускаются. При этом порядок записи не изменяется. После записи последнего значения в буфер и/или компаратор, формируется прерывание.

Примечание – Запущенные модули АЦП могут иметь разное время измерения и, как следствие, выдавать результаты измерений не одновременно. Тем не менее, сохранение результатов в буферах секвенсоров выполняется только после того как все запущенные модули закончат измерение.

По мере заполнения буфера секвенсора инкрементируется счетчик количества сохраненных результатов. Состояние счетчика доступно для чтения посредством регистра

FSTAT. Как только буфер будет заполнен полностью, установится флаг OV_s в регистре OSTAT.

Сохраненные в буфере данные доступны для чтения посредством регистра FIFO. При считывании данных буфер секвенсора будет опустошаться. Как только буфер будет полностью пуст, установится флаг UV_s в регистре USTAT.

Использование прямого доступа к памяти

Для разрешения использования DMA секвенсором, необходимо установить бит DMAEN в регистре CTL.

Поле WMARK задает уровень заполнения буфера секвенсора, по достижении которого будет запущен DMA. Перенос данных будет выполняться до полного опустошения буфера.

Если очередной запрос на запуск DMA пришел раньше, чем закончился предыдущий цикл DMA от того же секвенсора, то запрос кешируется и будет обслужен по окончании работы DMA. Для одного секвенсора может кешироваться только один запрос DMA. При возникновении еще одного запроса будет выставлен флаг ошибки DOV_s в регистре OSTAT.

21.4 Цифровой компаратор

Структурная схема компаратора показана на рисунке 21.8.

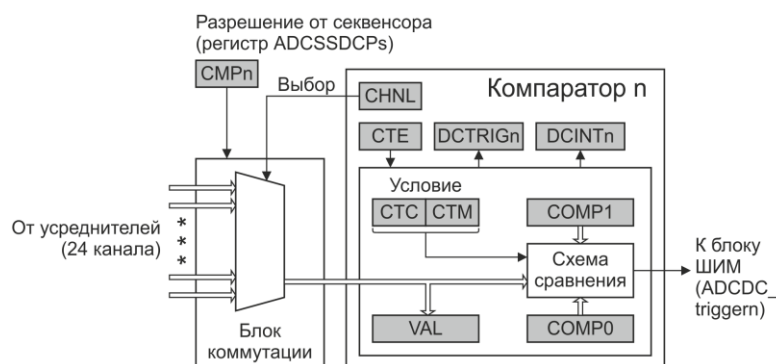


Рисунок 21.8 – Компаратор

Все компараторы блока АЦП независимы. Каждый компаратор может обрабатывать результат измерения любого канала, поскольку может быть скоммутирован с любым из 24 выходов модулей АЦП (на рисунке 21.8 сигналы «От усреднителей»).

Правила настройки:

1 Посредством регистра MUX выбираются каналы для измерений (установкой битов CH_n).

2 Посредством регистра DCP выбираются (разрешаются) компараторы для обработки полученных результатов измерений (установкой битов CMP_n). Запрещенные компараторы не обрабатывают полученные результаты.

3 Для каждого компаратора n в его регистре DCCTL_n в поле CHNL указывается номер канала, результат измерения которого будет передан на компаратор. По умолчанию, значение CHNL = 00h, т. е. все компараторы настроены на работу с нулевым каналом.

Примечание – Для разрешенных компараторов в поле CHNL может быть указан номер только того канала, который выбран в регистре MUX. Для запрещенных компараторов в поле CHNL может быть указан номер канала, который выбран в регистре MUX или записано любое значение от 18h до 1Fh.

Результат измерения, полученный компаратором, передается в схему сравнения и одновременно с этим сохраняется в регистре DCCTLn. Схема сравнения выполняет проверку соответствия результата измерения заданному условию (поля CTC, STM регистра DCCTLn и поля COMP0, COMP1 регистра DCCMPn) и в зависимости от результата проверки переключает выходной триггер. Работа триггера разрешается установкой бита STE регистра DCCTLn.

Функционирование компаратора

1 Сравнение по условию «Измерение \leq COMP0» (CTC = 00b):

- В однократном режиме (STM = 01b) выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме (STM = 00b) выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- В однократном режиме с гистерезисом (STM = 11b) выходной триггер переключится в единицу только в случае, если после прекращения выполнения условия «COMP1 \leq Измерение», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме с гистерезисом (STM = 10b) выходной триггер переключится в единицу в случае, если после прекращения выполнения условия «COMP1 \leq Измерение», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным, и далее триггер будет оставаться в состоянии единицы до тех пор, пока снова не выполнится условие «COMP1 \leq Измерение».

Пример функционирования триггера показан на рисунке 21.9.

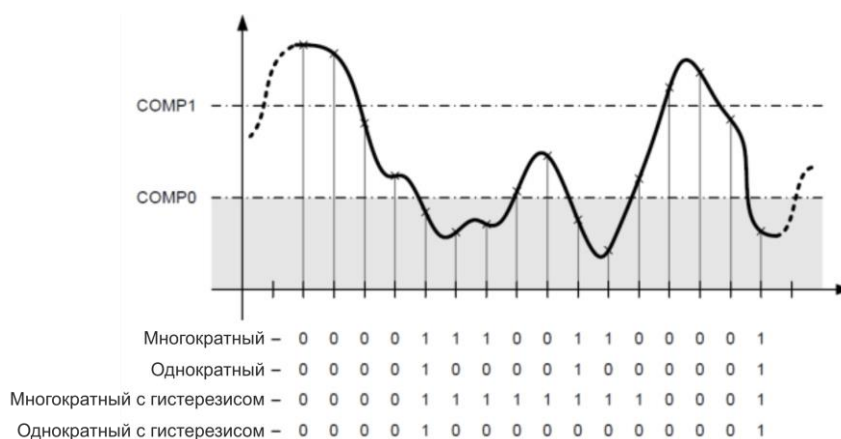


Рисунок 21.9 – Функционирование триггера при CTC = 00h

2 Сравнение по условию «COMP0 \leq Измерение \leq COMP1» (CTC = 01b):

- В однократном режиме выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- Однократный и многократный режимы с гистерезисом не поддерживаются.

Пример функционирования триггера показан на рисунке 21.10.

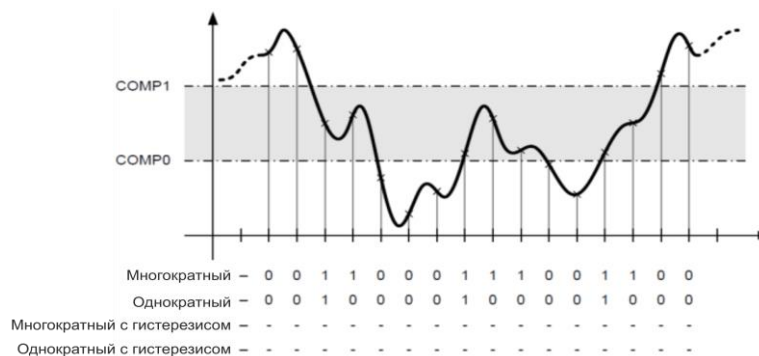


Рисунок 21.10 – Функционирование триггера при CTC = 01h

3 Сравнение по условию «COMP1 ≤ Измерение» (CTC = 11b):

- В однократном режиме выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- В однократном режиме с гистерезисом выходной триггер переключится в единицу только в случае, если после прекращения выполнения условия «Измерение ≤ COMP0», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме с гистерезисом (CTM = 11b) выходной триггер переключится в единицу в случае, если после прекращения выполнения условия «Измерение ≤ COMP0», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным, и далее триггер будет оставаться в состоянии единицы до тех пор, пока снова не выполнится условие «Измерение ≤ COMP0».

Пример функционирования триггера показан на рисунке 21.11.

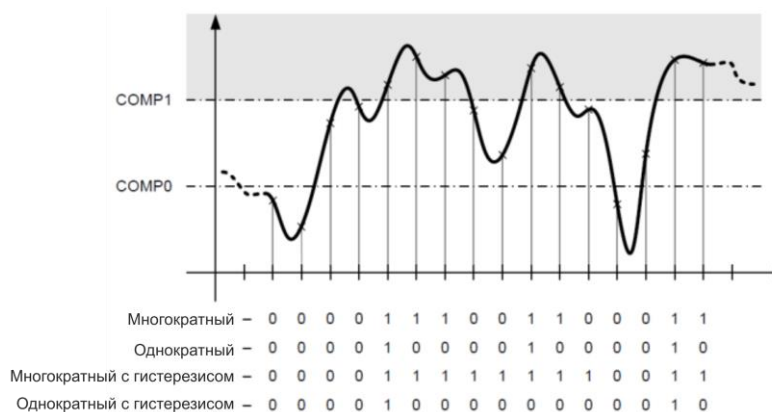


Рисунок 21.11 – Функционирование триггера при CTC = 11h

Переключение выходного триггера в единицу устанавливает соответствующий флаг DCTRIGn в регистре DCRTC и генерирует управляющий сигнал для пороговых выключателей блоков ШИМ. Сброс триггера и флага выполняется записью единицы в соответствующий бит регистра DCRTC.

Независимо от состояния триггера (разрешен или запрещен) в случае положительного результата сравнения компаратор может генерировать прерывание. Для этого следует установить бит CIE регистра DCCTLn и задать условия C1C и C1M.

Примечание – Условия срабатывания выходного триггера компаратора и условия генерирования прерываний могут не совпадать.

При генерировании прерываний устанавливаются флаги INRDCn и DCINn в регистрах RIS и ISC (маскированные прерывания). Маска прерывания задается регистром IM. Флаги сбрасываются записью единицы в соответствующий бит регистра ISC.

Флаги DCINn компараторов объединены по ИЛИ, и установка любого из них вызывает формирование прерывания ADC_CompInt блока АЦП.

22 Блок аналоговых компараторов

Блок компараторов содержит три идентичных аналоговых компаратора.

Каждый аналоговый компаратор сравнивает два аналоговых сигнала и формирует логический выходной сигнал с результатом сравнения. Выходной сигнал может быть использован как внутри микроконтроллера, так и подан на внешний вывод. Также компаратор может формировать прерывания и/или сигналы запуска секвенсоров блока АЦП, а также сигналы управления пороговым выключателем блока ШИМ.

22.1 Описание функционирования

Для начала работы блока компараторов и разрешения тактового сигнала необходимо установить бит ЕСМРЕН регистра APB_CLK и снять сигнал сброса в регистре PER_RST1. После этого включить блок компараторов установкой бита PWU регистра POWER.

Входные аналоговые сигналы на выводах A1 – A6 микроконтроллера являются сигналами C3+, C3–, C2+, C2–, C1+ и C1–, которые подаются на входы +ve и –ve аналоговых компараторов, а напряжение питания с вывода A7 – на входы CMP_DACSUP цифро-аналоговых преобразователей.

Примечание – Альтернативные функции CMP_C3+, CMP_C3–, CMP_C2+, CMP_C2–, CMP_C1+, CMP_C1– и CMP_DACSUP выводов A1 – A7 микроконтроллера постоянно активны и не требуют настройки и включения посредством регистров GPIO.

При срабатывании компараторов может генерироваться прерывание, а также сигналы на выводах микросхемы D11, E0 и E11, альтернативные функции CMP_OUT0, CMP_OUT1 и CMP_OUT2 соответственно.

Управление работой компараторов производится посредством регистров CTLn, управление питанием на встроенные ЦАП – регистров REFCTLn.

Функциональная схема блока компараторов показана на рисунке 22.1.

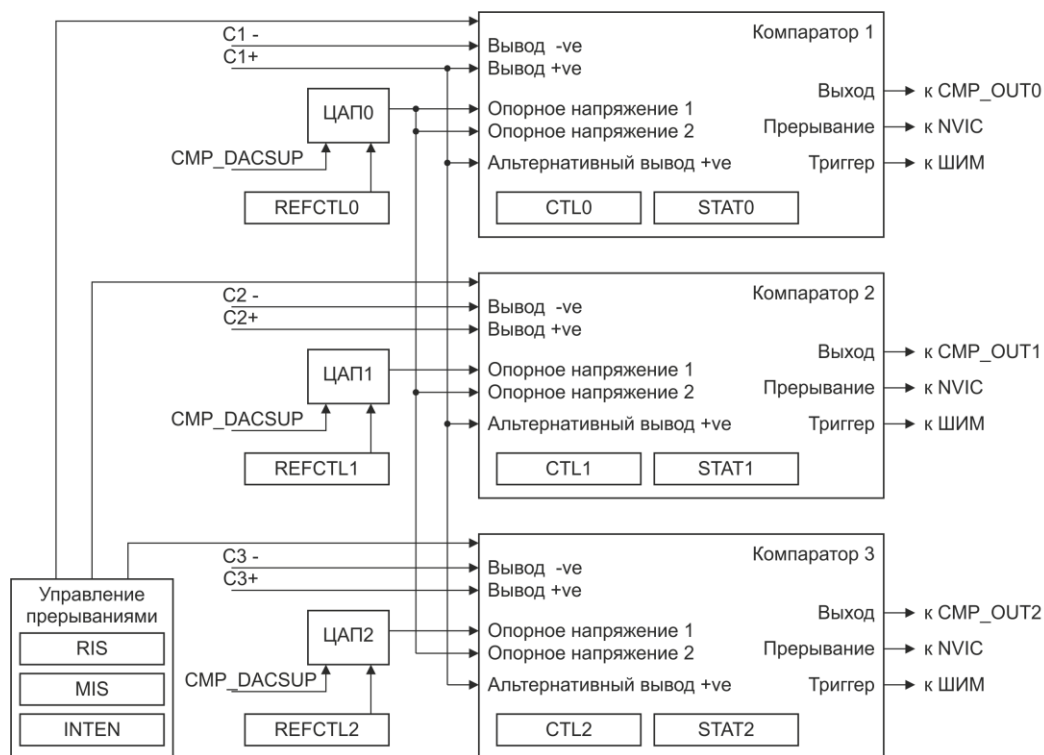


Рисунок 22.1 – Функциональная схема блока аналоговых компараторов

Компаратор может сравнивать напряжение на внешнем выводе микросхемы с напряжением на другом выводе, а также с напряжением на встроенном ЦАП.

Выходы блоков компараторов подключены к выводам микроконтроллера CMP_OUT0, CMP_OUT1 и CMP_OUT2.

Программирование параметров опорного напряжения ЦАП осуществляется посредством регистра REFCTLn.

Компаратор

Функциональная схема компаратора показана на рисунке 22.2

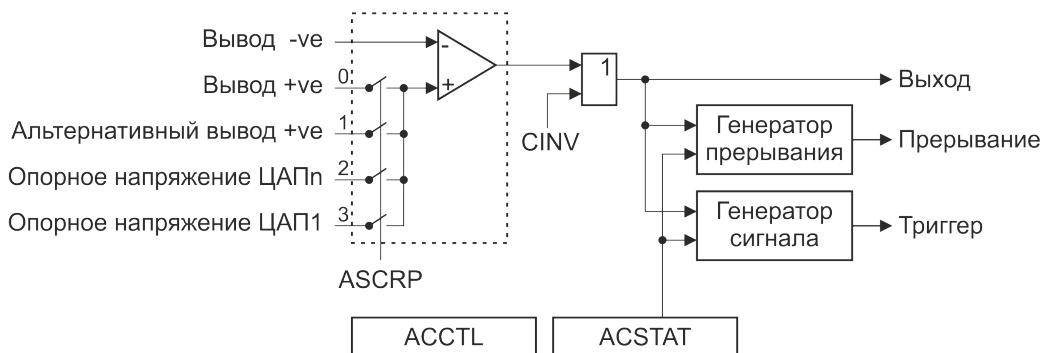


Рисунок 22.2 – Структура блока аналогового компаратора

Компаратор сравнивает два сигнала на входах «-» и «+». На вход «-» всегда подается напряжение -ve, снимаемое с соответствующего этому компаратору входа Cn-.

На вход «+» подается одно из четырех напряжений:

- напряжение +ve, снимаемое с соответствующего этому компаратору входа Cn+;
- напряжение с ЦАП, соответствующего этому компаратору;
- напряжение с ЦАП1 или с входа C1+ микроконтроллера.

Источник напряжения для входа «+» задается полем ASCRP регистра CTLn компаратора.

Логика сравнения:

- если «-» < «+», то на выходе CO логическая единица;
- если «-» > «+», то на выходе CO логический ноль.

Результат сравнения отражается в регистре STATn. Изменить полярность выходного сигнала можно битом CINV регистра CTLn.

Прерывания программируются посредством регистров MIS, RIS, INTEN и CTLn. Также регистр CTLn позволяет управлять событиями включения порогового включателя блока ШИМ и секвенсоров блока АЦП.

ЦАП

Структурная схема ЦАП представлена на рисунке 22.3.

ЦАП реализован на резистивных делителях напряжения, управление которыми осуществляется регистром REFCTLn. Бит RNG отключает дополнительный резистивный делитель N*R.

Характеристики ЦАП:

- напряжение питания – 3,3 В;
- компаратор работает в режиме Rail-to-Rail;
- выход компаратора совместим с ТТЛ;
- CMRR компаратора – не менее 50 дБ (на частоте $f = 1$ кГц на всем диапазоне уровней от 0 до 3,3 В);

- потребляемый ток I_o – не более 700 мкА (три компаратора плюс программируемый источник VIREF плюс источник опорных потенциалов);
- время установления TRT – не более 400 нс (п/у сигнал от минус 25 до плюс 25 мВ относительно уровня, уровень – от 0 до 3,3 В);
- напряжение смещения нуля V_{offset} – не более ± 30 мВ;
- гистерезис V_{HYS} – типовое значение 9 мВ, минимальное – не менее 6 мВ, максимальное – не более 16 мВ;
- потребляемый ток в режиме Hibernate – 3 мкА, максимальный – не более 5 мкА;
- время установления схемы в режим (включение питания, выход из режима Hibernate) – 5 мкс, максимальное – не более 10 мкс.

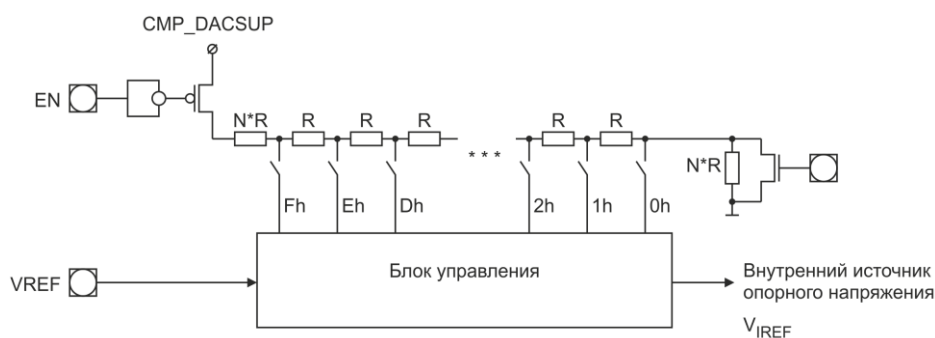


Рисунок 22.3 – Структурная схема модуля ЦАП

Если бит EN сброшен, опорное напряжение составит 0 В (питание ЦАП не подано).

В таблице 22.1 представлены значения выходного напряжения ЦАП при установленном бите EN и сброшенном бите RNG. Шаг опорного напряжения 113,2 мВ.

В таблице 22.2 представлены значения выходного напряжения ЦАП при установленных битах EN и RNG. Шаг опорного напряжения 148,7 мВ.

Значения выходного напряжения ЦАП VIREF приведены для трех значений напряжения питания VDDA.

Таблица 22.1 – Значения выходного напряжения ЦАП при EN = 1 и RNG = 0

| Поле VREF | VIREF_min (VDDA = 3 В) | VIREF_typical (VDDA = 3,3 В) | VIREF_max (VDDA = 3,6 В) |
|-----------|---------------------------|---------------------------------|-----------------------------|
| 0h | 0,718 | 0,792 | 0,866 |
| 1h | 0,821 | 0,905 | 0,990 |
| 2h | 0,924 | 1,019 | 1,114 |
| 3h | 1,027 | 1,132 | 1,237 |
| 4h | 1,129 | 1,245 | 1,361 |
| 5h | 1,232 | 1,358 | 1,485 |
| 6h | 1,335 | 1,472 | 1,608 |
| 7h | 1,437 | 1,585 | 1,732 |
| 8h | 1,540 | 1,698 | 1,856 |
| 9h | 1,643 | 1,811 | 1,979 |
| Ah | 1,745 | 1,924 | 2,103 |
| Bh | 1,848 | 2,038 | 2,227 |
| Ch | 1,951 | 2,151 | 2,350 |
| Dh | 2,054 | 2,264 | 2,474 |
| Eh | 2,156 | 2,377 | 2,597 |
| Fh | 2,259 | 2,490 | 2,720 |

Таблица 22.2 – Значения выходного напряжения ЦАП при EN = 1 и RNG = 1

| Поле VREF | VIREF_min (VDDA = 3 В) | VIREF_typical (VDDA = 3,3 В) | VIREF_max (VDDA = 3,6 В) |
|-----------|---------------------------|---------------------------------|-----------------------------|
| 0h | 0,004 | 0,006 | 0,008 |
| 1h | 0,141 | 0,155 | 0,168 |
| 2h | 0,276 | 0,303 | 0,331 |
| 3h | 0,411 | 0,452 | 0,493 |
| 4h | 0,546 | 0,601 | 0,655 |
| 5h | 0,681 | 0,750 | 0,819 |
| 6h | 0,815 | 0,898 | 0,980 |
| 7h | 0,950 | 1,047 | 1,143 |
| 8h | 1,085 | 1,196 | 1,307 |
| 9h | 1,220 | 1,344 | 1,468 |
| Ah | 1,354 | 1,493 | 1,630 |
| Bh | 1,490 | 1,642 | 1,795 |
| Ch | 1,625 | 1,790 | 1,956 |
| Dh | 1,760 | 1,939 | 2,119 |
| Eh | 1,895 | 2,088 | 2,280 |
| Fh | 2,028 | 2,236 | 2,443 |

23 Блок часов реального времени

Часы реального времени (RTC) предназначены для отсчета времени в микроконтроллере. Данный блок продолжает работу после отключения основного питания VDDPST и VDD. При этом происходит переключение на батарейное питание VBAT.

В качестве источника тактовой частоты используется кварцевый генератор с подключенным кварцевым резонатором (выводы XI_RTC, XO_RTC) с частотой 32,768 кГц.

На рисунке 23.1 показана схема подключения внешнего генератора синхросигнала.

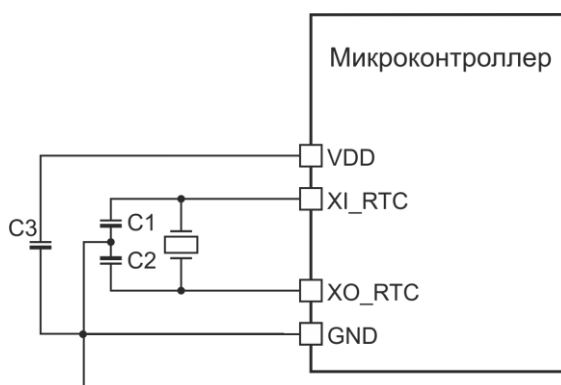


Рисунок 23.1 – Подключение внешнего генератора синхросигнала

23.1 Рекомендации по подключению и трассировке сигналов на печатной плате

Основные требования к подключению:

- минимизировать емкостные связи между сигналами RTC и другими сигналами;
- избегать параллельной трассировки сигналов RTC с другими высокоскоростными сигналами на печатной плате;
- поместить кварцевый осциллятор как можно ближе к выводам микросхемы;
- использовать симметричную трассировку для сигналов XI_RTC и XO_RTC;
- номиналы емкостей $C1 = C2 = (10 - 50) \text{ пФ}$, $C3 > 4,7 \text{ мкФ}$.

23.2 Описание работы блока

Блок RTC работает от батарейного питания напряжением (1,6 – 3,6) В. Функционирование RTC основано на подсчете импульсов тактового сигнала частотой 1,024 кГц, который в свою очередь получается делением внешнего входного сигнала частотой 32,768 кГц на 32 с помощью 5-разрядного счетчика.

Счетчик долей секунд инкрементируется с частотой 1,024 кГц. Когда значение счетчика достигает 1024, инкрементируется счетчик секунд. По достижении счетчиком секунд значения 59 инкрементируется счетчик минут и т. д. до счетчика лет.

Все счетчики имеют двоично-десятичный формат, за исключением счетчика долей секунд, который имеет двоичный формат. Каждый счетчик имеет теневой регистр, из которого загружается по команде в программе пользователя.

Блок RTC может корректно обрабатывать високосные года. Максимально значение лет, которые может отсчитать блок RTC, составляет 99.

Инициализация счетчиков осуществляется записью корректных значений.

Примечание – Сигнал сброса микроконтроллера не сбрасывает счетчики блока RTC, а только их теневые регистры. После снятия сигнала сброса рекомендуется подождать 1 мс, прежде чем осуществлять операции записи/чтения блока RTC.

Корректные значения для записи в регистры блока RTC

В течение работы значения счетчиков постоянно переписываются в теневые регистры по заднему фронту тактового сигнала. В это время запись и чтение регистров блока RTC запрещена. Для считывания информации о текущем времени и дате следует запретить перезаписывание теневых регистров записью значения 00000000h в регистр SHDW (при этом счетчики продолжают инкрементироваться).

Только после этого можно выполнить операцию чтения над одним или несколькими регистрами. Также можно считать состояние общего регистра TIME. После того как все желаемые регистры прочитаны нужно снова включить обновление (перезаписывание) теневых регистров записью значения 00000080h в регистр SHDW.

Для того чтобы изменить значение одного или нескольких счетчиков, следует выключить обновление теневых регистров (записать 00000000h в регистр SHDW), последовательно записать новые значения, после чего снова включить обновление. Новые значения будут загружены в соответствующие счетчики по ближайшему положительному фронту тактового сигнала (1,024 кГц).

Примечание – Задержка времени от включения обновления до перезаписи счетчиков составляет два такта синхросигнала и равняется 2 мс. Эта задержка может накапливаться, поэтому если перезагрузка счетчиков происходит без остановки обновления теневых регистров, запись в регистр долей секунд следует производить в последнюю очередь.

24 Сторожевой таймер

Сторожевой таймер позволяет сбросить систему в случае отказа программного обеспечения. Пользователь может включать или выключать таймер по собственному усмотрению.

Сторожевой таймер представляет собой 32-битный обратный счетчик, который загружается значением из регистра LOAD. Счетчик уменьшается на единицу по каждому нарастающему фронту тактового сигнала WDTCLK, если тактирование разрешено битом WDEN регистра APB_CLK и установлен бит WDRST регистра PER_RST0.

Включение счета таймера и его прерывания осуществляется установкой бита INTEN в регистре CTRL. Когда счетчик таймера достигает нуля, устанавливается флаг WDTINT в регистре MIS, а в счетчик загружается значение из регистра LOAD.

Далее, если установлен бит RESEN, счетчик продолжает декрементироваться. Если на момент повторного достижения нуля флаг WDTINT установлен, производится сброс микроконтроллера. Алгоритм работы таймера показан на рисунке 24.1.

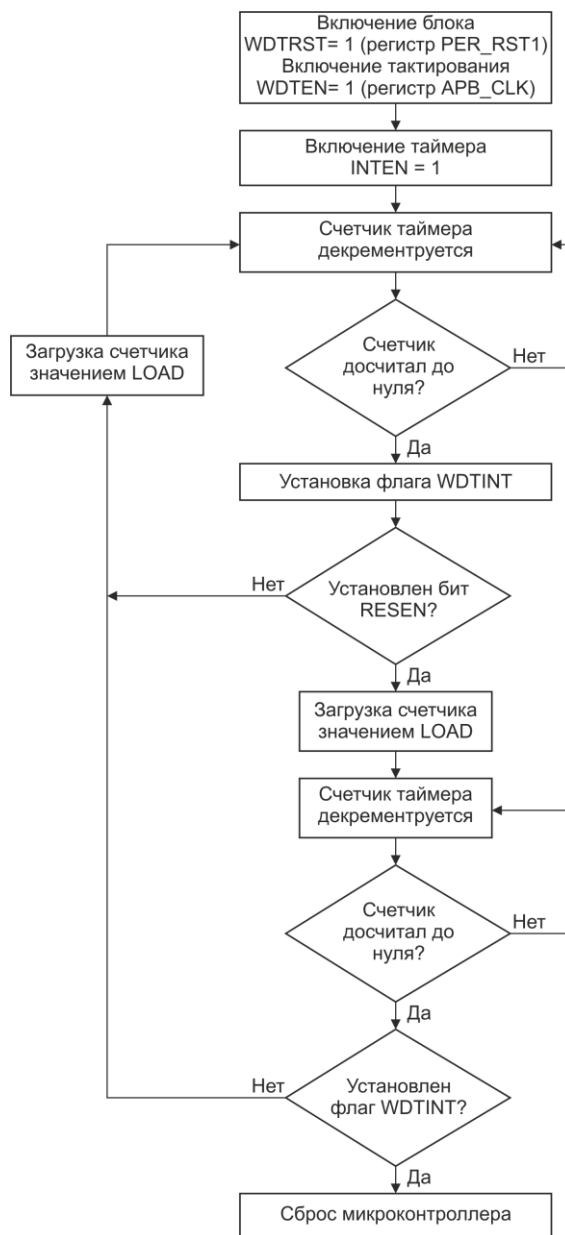


Рисунок 24.1 – Алгоритм работы таймера

25 Программно-аппаратные средства отладки

Для освоения и изучения 32-разрядных микроконтроллеров K1921BK01T, а также для макетирования и отладки систем пользователя на их основе следует использовать макетно-отладочную плату, которая позволяет подключать внешние элементы к портам микроконтроллера, работать с внешними интерфейсами, а также программировать встроенную флеш-память и выполнять отладку и оценку работы прикладных программ.

Описание макетно-отладочной платы приведено в КФДЛ.441461.010РЭ.

Для создания программного обеспечения рекомендуется использовать программный продукт CodeMaster++[ARM], который представляет собой набор программно-аппаратных средств для разработки и отладки систем на базе микроконтроллеров K1921BK01T.

Среда разработки CodeMaster++[ARM] включает в себя менеджер проектов, редактор исходных кодов, компилятор (C, C++), средства отладки и симуляции микроконтроллера K1921BK01T. Среда позволяет осуществлять отладку программ, а также программирование микроконтроллера посредством JTAG эмулятора JEM-NT-СМ4.

Адаптер JEM-NT-СМ4 обеспечивает взаимодействие между интегрированной средой разработки CodeMaster++[ARM], установленной на персональном компьютере, и отладочными ресурсами, встроенными в микроконтроллер K1921BK01T, а также выполнение отладочных функций. Информационный обмен с микроконтроллером осуществляется по одному из отладочных портов: JTAG или SWD.

Отладка пользовательской программы предполагает два основных режима работы: выполнение программы в режиме реального времени (RUN) и останов программы на определенном адресе или в определенный момент выполнения (HALT). Большинство отладочных функций доступно исключительно в режиме останова. В этом режиме отладчик JEM-NT-СМ4 во взаимодействии с CodeMaster++[ARM] позволяет анализировать и изменять ход исполнения пользовательской программы между отдельными участками программы, исполняемыми в режиме RUN.

При работе с пользовательской программой JEM-NT-СМ4 обеспечивает выполнение следующих отладочных действий:

- сброс микроконтроллера с остановом пользовательской программы на начальном адресе;
- запуск на выполнение программы в режиме реального времени (RUN);
- останов программы в произвольный момент времени (STOP);
- определение и изменение адреса выполнения программы (чтение и запись счетчика команд);
- чтение и запись доступных ресурсов микроконтроллера (ОЗУ, Flash, SFR и т. д.);
- установку и снятие точек останова по адресу выполнения программы;
- запуск на выполнение программы до определенного места в исходном коде (до курсора, до адреса);
- пошаговое исполнение программы: шаги низкого и высокого уровней, с заходом и без захода в подпрограммы.

Заключение

В настоящем Руководстве пользователя было представлено описание архитектуры, функционального построения и периферии микроконтроллера K1921BK01T. Техническое описание может служить практическим руководством по применению микроконтроллера для разработчиков систем на его основе и программистов.

Приложение А (обязательное) Регистры микроконтроллера

В таблицах А.1.1 – А.18.9 представлены регистры с указанием их назначений, мнемонических названий и адресов. Для большинства регистров показаны форматы и дано подробное описание их битов; для остальных регистров указаны разрядность и особенности программирования. При работе с регистрами, адреса которых не указаны, следует обратиться к приложению Б настоящего технического описания.

А.1 Регистры контроллера АЦП

Таблица А.1.1 – Регистр включения секвенсоров

| ACTSS | | 8000_0000h | | Сброс: 00000000h | | | | | | | | | | | |
|-------|------|--|-----------|------------------|----|----|----|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | ASEN | ASEN | ASEN | ASEN | ASEN | ASEN | ASEN | ASEN |
| - | | | | | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| ASENs | 7-0 | Бит разрешения работы секвенсора s (s от 0 до 7) | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | |
| – | 31-8 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.1.2 – Регистр флагов немаскированных прерываний

| RIS | | 8000_0004h | | Сброс: 00000000h | | | | | | | | | | | |
|--------|------|---|--|------------------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD |
| C23 | C22 | C21 | C20 | C19 | C18 | C17 | C16 | C15 | C14 | C13 | C12 | C11 | C10 | C9 | C8 |
| ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INRD | INRD | INRD | INRD | INRD | INRD | INRD | INRD | INR7 | NR6 | INR5 | INR4 | INR3 | INR2 | INR1 | INR0 |
| C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 | | | | | | | | |
| ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| INRDCn | 31-8 | Флаг прерываний компаратора n (n от 0 до 23) | | | | | | | | | | | | | |
| | | 0 | Нет прерывания или флаг сброшен | | | | | | | | | | | | |
| | | 1 | Поступил запрос на прерывание | | | | | | | | | | | | |
| INRs | 7-0 | Флаг прерываний секвенсора s (завершения измерения АЦП) | | | | | | | | | | | | | |
| | | 0 | Нет действий или АЦП в процессе измерения | | | | | | | | | | | | |
| | | 1 | Модули АЦП завершили измерения, и счетчик прерываний секвенсора s досчитал до значения ICNT регистра ADCSSCTLs | | | | | | | | | | | | |

Таблица А.1.3 – Регистр маски прерываний

| IM | | 8000_0008h | | | | | | | | | | | | | | Сброс: 00000000h | | | | | | | | | | | | | | | |
|-------|------|---|-------------|------|------|------|------|------|------|------|------|------|------|------|------|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MIDC | MASK | MASK | MASK | MASK | MASK | MASK | MASK | MASK |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч |
| Поле | Биты | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MIDCn | 31-8 | Маска прерывания компаратора n (n от 0 до 23) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Маскировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MASKs | 7-0 | Маска прерывания секвенсора s (s от 0 до 7) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Маскировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.1.4 – Регистр маскированных прерываний

| ISC | | 8000_000Ch | | | | | | | | | | | | | | Сброс: 00000000h | | | | | | | | | | | | | | | |
|---|------|---|------------------------------------|------|------|------|------|------|------|------|------|------|------|------|------|------------------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | DCIN | IN7 | IN6 | IN5 | IN4 | IN3 | IN2 | IN1 | IN0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч |
| Поле | Биты | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DCINn | 31-8 | Флаг прерывания компаратора n с учетом маски (n от 0 до 23) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Нет прерывания или оно маскировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Запрос на прерывание | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| INs | 7-0 | Флаг прерывания секвенсора s с учетом маски (s от 0 до 7) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Нет прерывания или оно маскировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Запрос на прерывание | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Запись единиц сбрасывает соответствующие флаги (также и в регистре RIS) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.1.5 – Регистр флагов буферов результатов и блока DMA

| OSTAT | | 8000_0010h | Сброс: 00000000h | | | | | | | | | | | | |
|-------|-------------|---|---|----|----|----|----|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | | DOV7 | DOV6 | DOV5 | DOV4 | DOV3 | DOV2 | DOV1 | DOV0 |
| | | | | | | | | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | OV7 | OV6 | OV5 | OV4 | OV3 | OV2 | OV1 | OV0 |
| | | | | | | | | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| DOVs | 23-16 | Флаг ошибки DMA | | | | | | | | | | | | | |
| | | 0 | Нет ошибки | | | | | | | | | | | | |
| | | 1 | При наличии двух обрабатываемых запросов DMA от секвенсора s, пришел третий запрос, который не может быть обработан (s от 0 до 7) | | | | | | | | | | | | |
| | | Флаг сбрасывается записью единицы | | | | | | | | | | | | | |
| OVs | 7-0 | Флаг заполнения буфера секвенсора s (s от 0 до 7) | | | | | | | | | | | | | |
| | | 0 | В буфере есть как минимум одна свободная ячейка | | | | | | | | | | | | |
| | | 1 | Буфер заполнен. Все последующие записи в буфер блокируются до появления как минимум одной свободной ячейки | | | | | | | | | | | | |
| | | Флаг сбрасывается записью единицы | | | | | | | | | | | | | |
| – | 31-24, 15-8 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.1.6 – Регистр выбора событий запуска секвенсоров

| EMUX | | 8000_0014h | Сброс: 00000000h | | | | | | | | | | | | |
|------|--|--|---|-----|----|----|----|-----|----|----|----|-----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EM7 | | | | EM6 | | | | EM5 | | | | EM4 | | | |
| 3ч | | | | 3ч | | | | 3ч | | | | 3ч | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EM3 | | | | EM2 | | | | EM1 | | | | EM0 | | | |
| 3ч | | | | 3ч | | | | 3ч | | | | 3ч | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| EMs | 31-0 | Поле выбора события для запуска секвенсора s (s от 0 до 7) | | | | | | | | | | | | | |
| | | 0h | Установка бита GSYNC в регистре PSSI | | | | | | | | | | | | |
| | | 1h | Сигнал от блока аналогового компаратора 0 | | | | | | | | | | | | |
| | | 2h | Сигнал от блока аналогового компаратора 1 | | | | | | | | | | | | |
| | | 3h | Сигнал от блока аналогового компаратора 2 | | | | | | | | | | | | |
| | | 4h | Любое прерывание GPIO | | | | | | | | | | | | |
| | | 5h | Сигнал от блока таймеров | | | | | | | | | | | | |
| 6h | Сигналы от блоков ШИМ0, ШИМ1, ШИМ2 – канал А | | | | | | | | | | | | | | |

Окончание таблицы А.1.6

| Поле | Биты | Описание | |
|------|------|----------|---|
| EMs | 31-0 | 7h | Сигналы от блоков ШИМ0, ШИМ1, ШИМ2 – канал В |
| | | 8h | Сигналы от блоков ШИМ3, ШИМ4, ШИМ5 – канал А |
| | | 9h | Сигналы от блоков ШИМ3, ШИМ4, ШИМ5 – канал В |
| | | Ah | Сигналы от блоков ШИМ6, ШИМ7, ШИМ8 – канал А |
| | | Bh | Сигналы от блоков ШИМ6, ШИМ7, ШИМ8 – канал В |
| | | Ch-Eh | Зарезервировано |
| | | Fh | Циклическая работа. Активируется сразу после включения секвенсора |

Таблица А.1.7 – Регистр пустоты буферов результатов

| USTAT | | 8000_0018h | Сброс: 00000000h |
|---|------|---|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border: 1px solid black; padding: 5px;"> - UV7UV6UV5UV4UV3UV2UV1UV0 </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> 3 ч3 ч3 ч3 ч3 ч3 ч3 ч3 ч </div> | | | |
| Поле | Биты | Описание | |
| UVs | 7-0 | Флаг пустоты буфера секвенсора s (s от 0 до 7) | |
| | | 0 | Буфер не пуст |
| | | 1 | Буфер пуст |
| | | Состояние бита обновляется после чтения FIFO. Флаг может быть сброшен программно записью единицы | |
| – | 31-8 | Зарезервировано | |

Таблица А.1.8 – Регистры фаз запуска модулей АЦП

| SPC0 | | 8000_0020h | Сброс: 00000000h |
|---|--|------------|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> - PHASE1 </div> <div style="text-align: center; margin-top: 5px;">3 ч</div> </div> | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> - PHASE0 </div> <div style="text-align: center; margin-top: 5px;">3 ч</div> </div> | | | |

Продолжение таблицы А.1.8

| | | | | | | | | | | | | | | | |
|-------------|----|------------|----|--------|----|----|----|----|----|----|----|------------------|----|----|----|
| SPC1 | | 8000_0024h | | | | | | | | | | Сброс: 00000000h | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | PHASE3 | | | | | | | | | | | |
| | | | | 3 4 | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | PHASE2 | | | | | | | | | | | |
| | | | | 3 4 | | | | | | | | | | | |
| SPC2 | | 8000_0028h | | | | | | | | | | Сброс: 00000000h | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | PHASE5 | | | | | | | | | | | |
| | | | | 3 4 | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | PHASE4 | | | | | | | | | | | |
| | | | | 3 4 | | | | | | | | | | | |
| SPC3 | | 8000_002Ch | | | | | | | | | | Сброс: 00000000h | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | PHASE7 | | | | | | | | | | | |
| | | | | 3 4 | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | PHASE6 | | | | | | | | | | | |
| | | | | 3 4 | | | | | | | | | | | |
| SPC4 | | 8000_0030h | | | | | | | | | | Сброс: 00000000h | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | PHASE9 | | | | | | | | | | | |
| | | | | 3 4 | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | PHASE8 | | | | | | | | | | | |
| | | | | 3 4 | | | | | | | | | | | |

Окончание таблицы А.1.8

| SPC5 | | 8000_0034h | Сброс: 00000000h | | | | | | | | | | | | |
|--------|--------------|--|------------------|---------|----|----|----|----|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | PHASE11 | | | | | | | | | | | |
| | | | | 3 ч | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | PHASE10 | | | | | | | | | | | |
| | | | | 3 ч | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| PHASEm | 27-16, 11-0 | Поле задания фазовой задержки (количество тактов сигнала системной частоты SysCLK) начала преобразования модулем АЦПm (m от 0 до 11) после запуска модуля секвенсором. Значение 0h – соответствует отсутствию задержки | | | | | | | | | | | | | |
| – | 31-28, 15-12 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.1.9 – Регистр параметров усреднителей

| SAC | | 8000_0038h | Сброс: 00000000h | | | | | | | | | | | | | | | | |
|------|---|--|-----------------------------------|-----|------|----|----|-----|------|----|----|-----|------|----|----|-----|--|--|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | |
| - | AVG7 | | | - | AVG6 | | | - | AVG5 | | | - | AVG4 | | | | | | |
| | | | | 3 ч | | | | 3 ч | | | | 3 ч | | | | 3 ч | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | |
| - | AVG3 | | | - | AVG2 | | | - | AVG1 | | | - | AVG0 | | | | | | |
| | | | | 3 ч | | | | 3 ч | | | | 3 ч | | | | 3 ч | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | | | | |
| AVGm | 30-28, 26-24, 22-20, 18-16, 14-12, 10-8, 6-4, 2-0 | Поле параметра усреднителей модуля АЦПm (m от 0 до 7). Задаёт количество измерений, используемых для получения усредненного результата преобразования | | | | | | | | | | | | | | | | | |
| | | 0h | Ноль. Усреднители не используется | | | | | | | | | | | | | | | | |
| | | 1h | 2 | | | | | | | | | | | | | | | | |
| | | 2h | 4 | | | | | | | | | | | | | | | | |
| | | 3h | 8 | | | | | | | | | | | | | | | | |
| | | 4h | 16 | | | | | | | | | | | | | | | | |
| | | 5h | 32 | | | | | | | | | | | | | | | | |
| | | 6h | 64 | | | | | | | | | | | | | | | | |
| 7h | Зарезервировано | | | | | | | | | | | | | | | | | | |
| – | 31, 27, 23, 19, 15, 11, 7, 3 | Зарезервировано | | | | | | | | | | | | | | | | | |

Таблица А.1.10 – Регистр настройки режима сброса счетчика прерываний

| RICNT | | 8000_003Ch | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---------------------------|---|--|----|----|----|-------|------------|------------|------------|------------|------------|------------|------------|------------|----|------|----|----|---|--|--|--|--|--|--|--|------------|------------|------------|------------|------------|------------|------------|------------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|--|-------|--|--|---|--|-------|--|--|---|--|------|--|--|---|--|------|--|--|--|----|--|--|--|--|----|--|--|--|--|----|--|--|--|--|----|--|
| <table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:3.125%; text-align:center;">31</td><td style="width:3.125%; text-align:center;">30</td><td style="width:3.125%; text-align:center;">29</td><td style="width:3.125%; text-align:center;">28</td><td style="width:3.125%; text-align:center;">27</td><td style="width:3.125%; text-align:center;">26</td><td style="width:3.125%; text-align:center;">25</td><td style="width:3.125%; text-align:center;">24</td><td style="width:3.125%; text-align:center;">23</td><td style="width:3.125%; text-align:center;">22</td><td style="width:3.125%; text-align:center;">21</td><td style="width:3.125%; text-align:center;">20</td><td style="width:3.125%; text-align:center;">19</td><td style="width:3.125%; text-align:center;">18</td><td style="width:3.125%; text-align:center;">17</td><td style="width:3.125%; text-align:center;">16</td> </tr> <tr> <td colspan="8" style="text-align:center;">-</td> <td style="text-align:center;">RICN T7</td><td style="text-align:center;">RICN T6</td><td style="text-align:center;">RICN T5</td><td style="text-align:center;">RICN T4</td><td style="text-align:center;">RICN T3</td><td style="text-align:center;">RICN T2</td><td style="text-align:center;">RICN T1</td><td style="text-align:center;">RICN T0</td> </tr> <tr> <td colspan="8"></td> <td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td> </tr> <tr> <td style="width:3.125%; text-align:center;">15</td><td style="width:3.125%; text-align:center;">14</td><td style="width:3.125%; text-align:center;">13</td><td style="width:3.125%; text-align:center;">12</td><td style="width:3.125%; text-align:center;">11</td><td style="width:3.125%; text-align:center;">10</td><td style="width:3.125%; text-align:center;">9</td><td style="width:3.125%; text-align:center;">8</td><td style="width:3.125%; text-align:center;">7</td><td style="width:3.125%; text-align:center;">6</td><td style="width:3.125%; text-align:center;">5</td><td style="width:3.125%; text-align:center;">4</td><td style="width:3.125%; text-align:center;">3</td><td style="width:3.125%; text-align:center;">2</td><td style="width:3.125%; text-align:center;">1</td><td style="width:3.125%; text-align:center;">0</td> </tr> <tr> <td colspan="2" style="text-align:center;">-</td> <td colspan="3" style="text-align:center;">AVG11</td> <td colspan="2" style="text-align:center;">-</td> <td colspan="3" style="text-align:center;">AVG10</td> <td colspan="2" style="text-align:center;">-</td> <td colspan="3" style="text-align:center;">AVG9</td> <td colspan="2" style="text-align:center;">-</td> <td colspan="2" style="text-align:center;">AVG8</td> </tr> <tr> <td colspan="2"></td> <td colspan="3" style="text-align:center;">3ч</td> <td colspan="2"></td> <td colspan="3" style="text-align:center;">3ч</td> <td colspan="2"></td> <td colspan="3" style="text-align:center;">3ч</td> <td colspan="2"></td> <td colspan="2" style="text-align:center;">3ч</td> </tr> </table> | | | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | - | | | | | | | | RICN T7 | RICN T6 | RICN T5 | RICN T4 | RICN T3 | RICN T2 | RICN T1 | RICN T0 | | | | | | | | | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | - | | AVG11 | | | - | | AVG10 | | | - | | AVG9 | | | - | | AVG8 | | | | 3ч | | | | | 3ч | | | | | 3ч | | | | | 3ч | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | RICN T7 | RICN T6 | RICN T5 | RICN T4 | RICN T3 | RICN T2 | RICN T1 | RICN T0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | AVG11 | | | - | | AVG10 | | | - | | AVG9 | | | - | | AVG8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 3ч | | | | | 3ч | | | | | 3ч | | | | | 3ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RICNTs | 23-16 | Бит выбора режима сброса счетчика прерываний секвенсора s (s от 0 до 7) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Нет действий | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Запрет сброса счетчика прерываний по запуску секвенсора. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | Счетчик прерываний может быть сброшен программно (записью в регистр ADCSSOP) или установкой бита ASENs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AVGm | 14-12, 10-8, 6-4, 2-0 | Поле параметра усреднителей модуля АЦПm (m от 8 до 11). Задаёт количество измерений, используемых для получения усредненного результата преобразования | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0h | Ноль. Усреднители не используется | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1h | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 2h | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 3h | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 4h | 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 5h | 32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 6h | 64 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7h | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| – | 31 – 24, 19, 15, 11, 7, 3 | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.1.11 – Регистр выбора каналов для измерений

| MUX | | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|------|------------------|------|------|------|-----|-----|------|------|------|------|------|------|------|------|----|----|---|--|--|--|--|--|--|--|------|------|------|------|------|------|------|------|--|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| <table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:3.125%; text-align:center;">31</td><td style="width:3.125%; text-align:center;">30</td><td style="width:3.125%; text-align:center;">29</td><td style="width:3.125%; text-align:center;">28</td><td style="width:3.125%; text-align:center;">27</td><td style="width:3.125%; text-align:center;">26</td><td style="width:3.125%; text-align:center;">25</td><td style="width:3.125%; text-align:center;">24</td><td style="width:3.125%; text-align:center;">23</td><td style="width:3.125%; text-align:center;">22</td><td style="width:3.125%; text-align:center;">21</td><td style="width:3.125%; text-align:center;">20</td><td style="width:3.125%; text-align:center;">19</td><td style="width:3.125%; text-align:center;">18</td><td style="width:3.125%; text-align:center;">17</td><td style="width:3.125%; text-align:center;">16</td> </tr> <tr> <td colspan="8" style="text-align:center;">-</td> <td style="text-align:center;">CH23</td><td style="text-align:center;">CH22</td><td style="text-align:center;">CH21</td><td style="text-align:center;">CH20</td><td style="text-align:center;">CH19</td><td style="text-align:center;">CH18</td><td style="text-align:center;">CH17</td><td style="text-align:center;">CH16</td> </tr> <tr> <td colspan="8"></td> <td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td> </tr> <tr> <td style="width:3.125%; text-align:center;">15</td><td style="width:3.125%; text-align:center;">14</td><td style="width:3.125%; text-align:center;">13</td><td style="width:3.125%; text-align:center;">12</td><td style="width:3.125%; text-align:center;">11</td><td style="width:3.125%; text-align:center;">10</td><td style="width:3.125%; text-align:center;">9</td><td style="width:3.125%; text-align:center;">8</td><td style="width:3.125%; text-align:center;">7</td><td style="width:3.125%; text-align:center;">6</td><td style="width:3.125%; text-align:center;">5</td><td style="width:3.125%; text-align:center;">4</td><td style="width:3.125%; text-align:center;">3</td><td style="width:3.125%; text-align:center;">2</td><td style="width:3.125%; text-align:center;">1</td><td style="width:3.125%; text-align:center;">0</td> </tr> <tr> <td style="text-align:center;">CH15</td><td style="text-align:center;">CH14</td><td style="text-align:center;">CH13</td><td style="text-align:center;">CH12</td><td style="text-align:center;">CH11</td><td style="text-align:center;">CH10</td><td style="text-align:center;">CH9</td><td style="text-align:center;">CH8</td><td style="text-align:center;">CH7</td><td style="text-align:center;">CH6</td><td style="text-align:center;">CH5</td><td style="text-align:center;">CH4</td><td style="text-align:center;">CH3</td><td style="text-align:center;">CH2</td><td style="text-align:center;">CH1</td><td style="text-align:center;">CH0</td> </tr> <tr> <td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td><td style="text-align:center;">3ч</td> </tr> </table> | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | - | | | | | | | | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 | | | | | | | | | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.1.11

| Поле | Биты | Описание |
|------|-------|--|
| СНп | 23-0 | Бит выбора канала n (n от 0 до 23) для измерения. Каждый секвенсор может запускать измерение по любым каналам |
| | | 0 Канал не выбран |
| | | 1 Канал выбран |
| – | 31-24 | Зарезервировано |

Таблица А.1.12 – Регистр управления

| Поле | Биты | Описание |
|--|-----------------|--|
| CTL | | Сброс: 00000000h |
|  | | |
| DMAEN | 28 | Бит разрешения использования блока DMA 0 Запрещено 1 Разрешено |
| WMARK | 26-24 | Поле задания количества результатов измерений записанных в буфер секвенсора, по достижению которого вызывается DMA |
| | | 000 Зарезервировано |
| | | 001 Одна запись в буфер |
| | | 010 2 |
| | | 011 4 |
| | | 100 8 |
| | | 101 16 |
| | | 110 32 |
| 111 Зарезервировано | | |
| ICNT | 23-16 | Цикличность прерываний. Поле задания количества перезапусков модулей АЦП секвенсором, по достижении которого генерируется прерывание. Значение 00h означает выставление прерывания по каждому перезапуску модуля АЦП, значение FFh – каждые 256 перезапусков |
| RCNT | 7-0 | Количество перезапусков АЦП. Поле задания количества перезапусков модулей АЦП секвенсором после его запуска по событию. Значение 00h задает один пуск модулей АЦП при запуске секвенсора, значение FFh – 256 перезапусков |
| – | 31-29, 27, 15-8 | Зарезервировано |

Таблица А.1.13 – Регистр измерения

| FIFO | | Сброс: 00000000h |
|--|-------|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> - DATA </div> <div style="text-align: center; margin-top: 5px;">Ч</div> </div> | | |
| Поле | Биты | Описание |
| DATA | 11-0 | Результат измерения. Чтение поля DATA возвращает результат измерения из буфера секвенсора |
| – | 31-12 | Зарезервировано |

Таблица А.1.14 – Регистр количества измерений

| FSTAT | | Сброс: 00000000h |
|---|------|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> - FLOAD </div> <div style="text-align: center; margin-top: 5px;">Ч</div> </div> | | |
| Поле | Биты | Описание |
| FLOAD | 0-5 | Значение количества результатов измерений, сохраненных в буфере секвенсора |
| – | 31-6 | Зарезервировано |

Таблица А.1.15 – Регистр флагов измерений

| OP | | Сброс: 00000000h |
|--|--|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> - ICNT </div> <div style="text-align: center; margin-top: 5px;">3 Ч</div> </div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> - RCNT </div> <div style="text-align: center; margin-top: 5px;">Ч</div> </div> | | |

Окончание таблицы А.1.15

| Поле | Биты | Описание |
|------|------------|--|
| ICNT | 23-16 | Счетчик измерений, используемый для вызова прерываний. При равенстве значения счетчика и значения ICNT регистра ADCSSCTL генерируется прерывание. Любая запись в регистр ADCSSOP сбрасывает счетчик ICNT |
| RCNT | 7-0 | Счетчик количества измерений, проведенных модулями АЦП с момента запуска секвенсора |
| – | 31-8,24-31 | Зарезервировано |

Таблица А.1.16 – Регистр выбора компаратора

| Поле | Биты | Описание |
|---|-------|---|
| <p>DCP Сброс: 00000000h</p> | | |
| CMPn | 23-0 | Бит разрешения работы компаратора n секвенсором |
| | 0 | Запрещен |
| | 1 | Разрешен |
| – | 31-24 | Зарезервировано |

Таблица А.1.17 – Регистр задержки перезапусков

| Поле | Биты | Описание |
|---|-------|---|
| <p>TMR Сброс: 00000000h</p> | | |
| TMR | 23-0 | Поле задания задержки запуска модуля АЦП. Значение TMR = 000000h задает немедленный перезапуск по окончании измерения |
| – | 31-24 | Зарезервировано |

Таблица А.1.18 – Регистр управления компаратора n (n от 0 до 23)

| DCCTLn | | | | | | | | | | | | | | | Сброс: 00000000h | | | | |
|--|-------------------|---|--|--|-----|--|-----|--|---|------|--|-----|-----|-----|------------------|--|--|--|--|
| | | | | | | | | | | CHNL | | | | | | | | | |
| | | | | | | | | | | 3 ч | | | | | | | | | |
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | | | | | | | | | | | | | | | | | | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | | | | | | | | | | | | | | | | | | |
| - | | | СТЕ | | СТС | | СТМ | | - | | | СІЕ | СІС | СІМ | | | | | |
| | | | 3 ч | | 3 ч | | 3 ч | | | | | 3 ч | 3 ч | 3 ч | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | | | | |
| CHNL | 20-16 | Номер канала. Поле выбирает канал, результат измерения которого будет передан на компаратор | | | | | | | | | | | | | | | | | |
| СТЕ | 12 | Бит разрешения срабатывания выходного триггера компаратора | | | | | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | | | | | |
| СТС | 11-10 | Поле задания условия срабатывания выходного триггера. Если для значения, полученного в результате измерения, выполняется условие, то состояние триггера единица, в противном случае – ноль | | | | | | | | | | | | | | | | | |
| | | 00 | Измерение \leq COMP0 | | | | | | | | | | | | | | | | |
| | | 01 | COMP0 \leq Измерение \leq COMP1 | | | | | | | | | | | | | | | | |
| | | 10 | Зарезервировано | | | | | | | | | | | | | | | | |
| | | 11 | COMP1 \leq Измерение | | | | | | | | | | | | | | | | |
| Параметры COMP0 и COMP1 задаются в регистре DCCMPn | | | | | | | | | | | | | | | | | | | |
| СТМ | 9-8 | Поле задания режима срабатывания выходного триггера | | | | | | | | | | | | | | | | | |
| | | 00 | Многократный | | | | | | | | | | | | | | | | |
| | | 01 | Однократный | | | | | | | | | | | | | | | | |
| | | 10 | Многократный с гистерезисом | | | | | | | | | | | | | | | | |
| | | 11 | Однократный с гистерезисом | | | | | | | | | | | | | | | | |
| СІЕ | 4 | Бит разрешения прерывания компаратора | | | | | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | | | | | |
| | | 1 | Разрешено. Прерывание генерируется каждый раз при одновременном выполнении условий СІС и СІМ | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |
| СІС | 3-2 | Поле задания условия генерирования прерывания | | | | | | | | | | | | | | | | | |
| | | 00 | Измерение \leq COMP0 | | | | | | | | | | | | | | | | |
| | | 01 | COMP0 \leq Измерение \leq COMP1 | | | | | | | | | | | | | | | | |
| | | 10 | Зарезервировано | | | | | | | | | | | | | | | | |
| | | 11 | COMP1 \leq Измерение | | | | | | | | | | | | | | | | |
| СІМ | 1-0 | Поле задания режима генерирования прерывания | | | | | | | | | | | | | | | | | |
| | | 00 | Многократный | | | | | | | | | | | | | | | | |
| | | 01 | Однократный | | | | | | | | | | | | | | | | |
| | | 10 | Многократный с гистерезисом | | | | | | | | | | | | | | | | |
| | | 11 | Однократный с гистерезисом | | | | | | | | | | | | | | | | |
| - | 31-21, 15-13, 7-5 | Зарезервировано | | | | | | | | | | | | | | | | | |

Таблица А.1.19 – Регистр диапазона компаратора n (n – номер от 0 до 23)

| DCCMPn | | Сброс: 00000000h | | | | | | | | | | | | | |
|-----------------|-----------------|---|----|-------|----|----|----|----|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | COMP1 | | | | | | | | | | | |
| 3 ч | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | COMP0 | | | | | | | | | | | |
| 3 ч | | | | | | | | | | | | | | | |
| COMP1, COMP0 | 27-16, 11-0 | Поля значений верхнего и нижнего, соответственно, порогов диапазона измерений. Всегда должно выполняться условие $COMP0 \leq COMP1$ | | | | | | | | | | | | | |
| - | 31-28, 15-12 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.1.20 – Регистр результата измерения компаратора n (n – номер от 0 до 23)

| DCVALn | | Сброс: 00000000h | | | | | | | | | | | | | |
|---------------|-------|---|----|-----|----|----|----|----|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | VAL | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| VAL | 11-0 | Значение результата измерения, которое последним использовалось компаратором при проверке на соответствие условиям СТС и СТМ (см. регистр DCCTLn) | | | | | | | | | | | | | |
| - | 31-12 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.1.21 – Регистр настроек модуля АЦПm (m от 0 до 11)

| PPm | | Сброс: 0220300Bh | | | | | | | | | | | | | |
|------------|----|------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ENA | - | 0 | 0 | 0 | 0 | 1 | 0 | 1 | | | 0M | | | | |
| 3 ч | | | 3 ч | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | | | | | | | |

Окончание таблицы А.1.21

| Поле | Биты | Описание |
|--------------------------------------|----------|--|
| ENA | 31 | Бит разрешения работы модуля |
| | | 0 Запрещено |
| | | 1 Разрешено |
| – | 29-24 | Предустановлено. При чтении – 02h |
| OM | 23 | Бит управления опорным током АЦП |
| | | 0 Нормальный режим |
| | | 1 Смещение +25 % |
| | 22 | Режим функционирования блока SHA2 |
| | | 0 Однополярный |
| | | 1 Дифференциальный. Не может быть включен одновременно с дифференциальным режимом блока SHA1 |
| | 21 | Предустановлен. При чтении – 1b |
| | 20 | Режим функционирования блока SHA1 |
| | | 0 Однополярный |
| | | 1 Дифференциальный. Не может быть включен одновременно с дифференциальным режимом блока SHA2 |
| | 19 | Разрядность модуля |
| | | 0 12 |
| | | 1 10 |
| | 18-16 | Режим модуля |
| | | 000 Powerdown. Модуль выключен |
| | | 001 StandBy |
| 011 Модуль включен | | |
| Остальные комбинации зарезервированы | | |
| – | 30, 15-0 | Зарезервировано |

Таблица А.1.22 – Регистр синхронизации секвенсоров

| Поле | Биты | Описание |
|---|------|--|
| PSSI | | 8000_03F8h |
| | | Сброс: 00000000h |
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | |
| G SYNC | | - |
| 3 ч | | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | |
| | | - |
| | | SS7 SS6 SS5 SS4 SS3 SS2 SS1 SS0 |
| | | 3 3 3 3 3 3 3 3 |
| GSYNC | 31 | Бит запуска секвенсоров. Запись единицы запускает секвенсоры, работа которых разрешена и для которых установлены биты SSs (s от 0 до 7) |
| SSs | 7-0 | Бит разрешения запуска секвенсора s |
| | | 0 Запрещено |
| | | 1 Разрешено (если установлен бит ASENs в регистре ACTSS) |
| – | 30-8 | Зарезервировано |

Таблица А.1.23 – Регистр сброса флагов компараторов

| DCRTC (запись) | | | | | | | | | | | | | | | | 8000_02ECh | Сброс: 00000000h |
|--------------------------------|------------------|---|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|---------------------|---------------------|------------------|------------------|------------|------------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | |
| DC TRIG 23 | DC TRIG 22 | DC TRIG 21 | DC TRIG 20 | DC TRIG 19 | DC TRIG 18 | DC TRIG 17 | DC TRIG 16 | DC TRIG 15 | DC TRIG 14 | DC TRIG 13 | DC TRIG 12 | DC TRIG 11, 1 | DC TRIG 10, 0 | DC TRIG 9 | DC TRIG 8 | | |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| DC TRIG 7 | DC TRIG 6 | DC TRIG 5 | DC TRIG 4 | DC TRIG 3 | DC TRIG 2 | | | | | | | - | | | | | |
| 3 | 3 | 3 | 3 | 3 | 3 | | | | | | | | | | | | |
| DCRTC (чтение) | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | |
| | | | | - | | | | DC TRIG 23 | DC TRIG 22 | DC TRIG 21 | DC TRIG 20 | DC TRIG 19 | DC TRIG 18 | DC TRIG 17 | DC TRIG 16 | | |
| | | | | | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| DC TRIG 15 | DC TRIG 14 | DC TRIG 13 | DC TRIG 12 | DC TRIG 11 | DC TRIG 10 | DC TRIG 9 | DC TRIG 8 | DC TRIG 7 | DC TRIG 6 | DC TRIG 5 | DC TRIG 4 | DC TRIG 3 | DC TRIG 2 | DC TRIG 1 | DC TRIG 0 | | |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | | |
| DCTRIGN (режим записи) | 31-10 | Запись единицы в бит сбрасывает выходной триггер соответствующего компаратора. Примечание – Запись единицы в бит 19 активирует сброс триггеров компараторов 11 и 1, запись единицы в бит 18 активирует сброс триггеров компараторов 10 и 0 | | | | | | | | | | | | | | | |
| DCTRIGN (режим чтения) | 23-0 | Чтение бита возвращает текущее состояние выходного триггера соответствующего компаратора | | | | | | | | | | | | | | | |
| Остальные биты зарезервированы | | | | | | | | | | | | | | | | | |

А.2 Регистры GPIO и общего назначения

Таблица А.2.1 – 16-разрядные регистры порта

| Мнемоника и назначение | Обращение к регистру | Описание | Сброс |
|---|----------------------|--|-------|
| DATA Регистр входных данных порта | Чтение | Данные на входе порта | ----h |
| | Запись | Значение выхода порта. Чтение записанного значения возможно после задержки в два такта системной частоты из-за пересинхронизации входного значения | |
| DATAOUT Выходной регистр данных порта | Чтение | Текущее значение регистра | 0000h |
| | Запись | Запись значения регистра данных | |

16-разрядные регистры, представленные в таблице А.2.2, управляют выводами порта. Каждому биту регистра соответствует один вывод порта – нулевому биту соответствует нулевой вывод, 15 биту – 15 вывод.

Таблица А.2.2 – 16-разрядные регистры порта

| Мнемоника и назначение | Обращение к биту регистра | Описание | Сброс | |
|--|---------------------------|----------|---|--|
| OUTENSET Регистр разрешения выходов порта | Чтение | 0 | Вывод в состоянии вход | 0000h |
| | | 1 | Вывод в состоянии выход | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Перевод вывода в состояние выход | |
| OUTENCLR Регистр сброса разрешения выходов порта | Чтение | 0 | Вывод в состоянии вход | 0000h |
| | | 1 | Вывод в состоянии выход | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Перевод вывода в третье состояние | |
| ALTFUNCSET Регистр включения альтернативной функции порта | Чтение | 0 | Вывод в режиме GPIO | Порт А: 0000h Порт В: 0007h Порт С: 0000h Порт D: 0800h Порт E: 0403h Порт F: 0000h Порт G: 0000h Порт H: 0000h |
| | | 1 | Вывод в режиме альтернативной функции | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Перевод вывода в режим альтернативной функции | |
| ALTFUNCCLR Регистр выключения альтернативной функции порта | Чтение | 0 | Вывод в режиме GPIO | 0000h |
| | | 1 | Вывод в режиме альтернативной функции | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Перевод вывода в режим GPIO | |
| INTENSET Регистр разрешения прерываний порта | Чтение | 0 | Прерывания запрещены | 0000h |
| | | 1 | Прерывания разрешены | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Разрешение прерываний | |

Окончание таблицы А.2.2

| Мнемоника и назначение | Обращение к биту регистра | | Описание | Сброс |
|---|---------------------------|---|---|-------|
| INTENCLR Регистр сброса разрешения прерываний порта | Чтение | 0 | Прерывания запрещены | 0000h |
| | | 1 | Прерывания разрешены | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Запрещение прерываний | |
| INTTYPESET Регистр типа прерываний порта | Чтение | 0 | Прерывания по уровню | 0000h |
| | | 1 | Прерывания по фронту | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Установка прерывания по фронту | |
| INTTYPECLR Регистр сброса типа прерываний порта | Чтение | 0 | Прерывания по уровню | 0000h |
| | | 1 | Прерывания по фронту | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Установка прерывания по уровню | |
| INTPOLSET Регистр полярности события прерывания порта | Чтение | 0 | Прерывания по низкому уровню или отрицательному фронту | 0000h |
| | | 1 | Прерывания по высокому уровню или положительному фронту | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Установка прерывания по высокому уровню или положительному фронту | |
| INTPOLCLR Регистр сброса полярности события прерывания порта | Чтение | 0 | Прерывания по низкому уровню или отрицательному фронту | 0000h |
| | | 1 | Прерывания по высокому уровню или положительному фронту | |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Установка прерывания по низкому уровню или отрицательному фронту | |
| INTSTATUS, INTCLEAR Регистр состояния и сброса прерываний порта | Чтение | Статус прерывания | | 0000h |
| | Запись | 0 | Состояние вывода не изменяется | |
| | | 1 | Сброс прерывания | |
| MASKLOWBYTE Массив из 256 регистров масок младшего байта порта | Чтение Запись | Доступ по маске для младших восьми бит порта. Биты адреса с 9 по 2 используются как битовая маска. Механизм маскирования показан на рисунке 10.5. | | ----h |
| MASKHIGHBYTE Массив из 256 регистров масок старшего байта порта | Чтение Запись | Доступ по маске для старших восьми бит порта. Биты адреса с 9 по 2 используются как битовая маска. Механизм маскирования показан на рисунке 10.6. | | ----h |

Часть регистров портов (см. таблицу А.2.3) условно собраны в четыре группы. Каждая группа позволяет управлять парой портов. На рисунке А.2.1 показано, какие порты образуют пары и соответствие битов регистров и выводов портов. Биты 31–16 регистра соответствуют битам 15–0 одного порта. Аналогично биты 15–0 регистра соответствуют битам 15–0 другого порта.

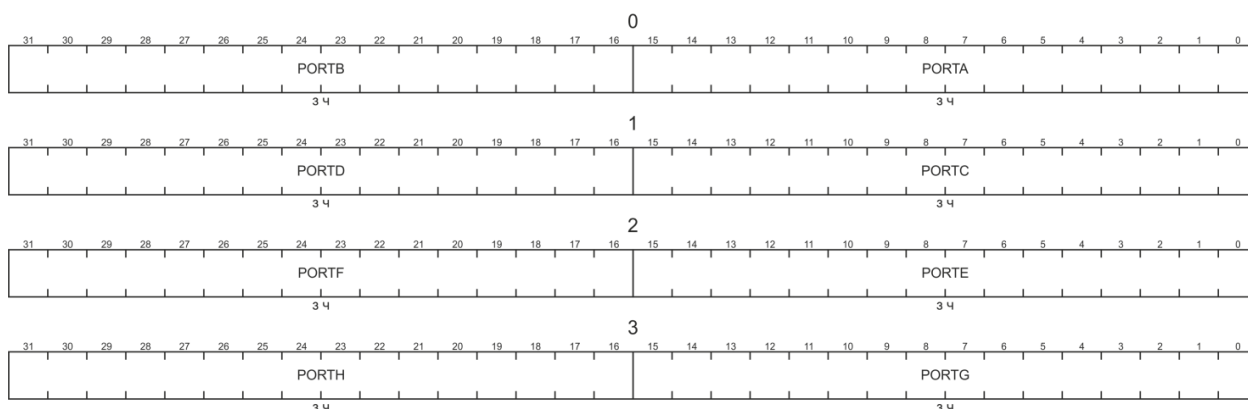


Рисунок А.2.1 – Форматы регистров управления портами

Индекс i в конце мнемонического названия регистра указывает на группу и может иметь четыре значения:

- 0 – группа портов А и В;
- 1 – группа портов С и D;
- 2 – группа портов Е и F;
- 3 – группа портов G и H.

Таблица А.2.3 – 32-разрядные регистры портов ($i = 0, 1, 2, 3$)

| Мнемоника и назначение | Состояние бита | Описание состояния вывода, которым управляет бит |
|--|----------------|--|
| GPIOENi Регистр разрешения выходов портов | 0 | Вывод находится в третьем состоянии |
| | 1 | Вывод активен |
| GPIOODCTLi Регистр включения режима открытого коллектора портов | 0 | Нормальная работа вывода |
| | 1 | Вывод находится в режиме открытого коллектора |
| GPIOДСCTLi Регистр управления нагрузочной способностью выводов портов | 0 | Режим нормальной нагрузочной способности вывода (макс. ток 4,5 мА) |
| | 1 | Режим повышенной нагрузочной способности вывода (макс. ток 6 мА) |
| GPIOPUCTLi Регистр управления схемой «PullUp» портов | 0 | Схема «Pull-up» вывода отключена |
| | 1 | Схема «Pull-up» вывода включена |
| GPIOSEi Регистр управления пересинхронизацией входов портов | 0 | Сигнал с входа передается без изменений |
| | 1 | Сигнал с входа пересинхронизируется через два триггера |
| GPIOQEi Регистр включения фильтров GPIO портов | 0 | Фильтр вывода отключен |
| | 1 | Фильтр вывода подключен |

Окончание таблицы А.2.3

| Мнемоника и назначение | Состояние бита | Описание состояния вывода, которым управляет бит |
|---|----------------|--|
| GPIOQMi Регистр режимов работы фильтров портов | 0 | Выбрана фильтрация по трем отсчетам входного сигнала. Если все три отсчета имеют одинаковое значение, то данное значение подается на выход, иначе остается предыдущее |
| | 1 | Выбрана фильтрация по шести отсчетам входного сигнала. Если все шесть отсчета имеют одинаковое значение, то данное значение подается на выход, иначе остается предыдущее |
| Примечание – Состояние всех регистров после сброса микроконтроллера 00000000h. Исключение составляют три регистра: GPIODEN0 – 00020062h; GPIODEN1 – 08000000h; GPIODEN2 – 00000400h. | | |

Таблица А.2.4 – Регистр выбора функции порта x (x – название порта от А до Н)

| GPIOCTLx | | Сброс: 00000000h | | | | | | | | | | | | | |
|--|------|---|-----------------|-------|----|-------|----|-------|----|-------|----|------|----|------|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Pin15 | | Pin14 | | Pin13 | | Pin12 | | Pin11 | | Pin10 | | Pin9 | | Pin8 | |
| 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Pin7 | | Pin6 | | Pin5 | | Pin4 | | Pin3 | | Pin2 | | Pin1 | | Pin0 | |
| 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| Pinn | 31-0 | Поле выбора альтернативной функции вывода порта | | | | | | | | | | | | | |
| | | 00 | Функция 1 | | | | | | | | | | | | |
| | | 01 | Функция 2 | | | | | | | | | | | | |
| | | 10 | Функция 3 | | | | | | | | | | | | |
| | | 11 | Зарезервировано | | | | | | | | | | | | |
| Примечание – n – порядковый номер вывода порта от 0 до 15. | | | | | | | | | | | | | | | |

Таблица А.2.5 – Регистры настройки фильтров портов

| GPIOQPAD | | 8003_0110h | | Сброс: 00000000h | | | | | | | | | | | |
|-----------------|----|------------|----|------------------|----|-------|----|----|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SPRDD | | | | | | SPRDC | | | | | | | | | |
| 3 ч | | | | | | 3 ч | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPRDB | | | | | | SPRDA | | | | | | | | | |
| 3 ч | | | | | | 3 ч | | | | | | | | | |

Окончание таблицы А.2.5

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|------------|------------------|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|-------|--|--|--|--|--|--|--|-------|--|--|--|--|--|--|--|-----|--|--|--|--|--|--|--|-----|--|--|--|--|--|--|--|--|--|
| GPIOQREN | 8003_0114h | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td> </tr> <tr> <td colspan="8" style="text-align: center;">SPRDH</td> <td colspan="8" style="text-align: center;">SPRDG</td> </tr> <tr> <td colspan="8" style="text-align: center;">3 ч</td> <td colspan="8" style="text-align: center;">3 ч</td> </tr> </table> | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | SPRDH | | | | | | | | SPRDG | | | | | | | | 3 ч | | | | | | | | 3 ч | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SPRDH | | | | | | | | SPRDG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 ч | | | | | | | | 3 ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td colspan="8" style="text-align: center;">SPRDF</td> <td colspan="8" style="text-align: center;">SPRDE</td> </tr> <tr> <td colspan="8" style="text-align: center;">3 ч</td> <td colspan="8" style="text-align: center;">3 ч</td> </tr> </table> | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | SPRDF | | | | | | | | SPRDE | | | | | | | | 3 ч | | | | | | | | 3 ч | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SPRDF | | | | | | | | SPRDE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 ч | | | | | | | | 3 ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Примечание – Буква в конце названия битового поля указывает на порт, которым оно управляет. Битовое поле задает временной интервал (в тактах системной частоты) между отсчетами фильтров соответствующего порта. При 00h отсчеты производятся каждый такт. Временной интервал задается одновременно для всех выводов порта.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.2.6 – Регистр управления настройками PLL

| PLL_CTRL | 8003_0094h | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|------------|---|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|---------|---|--|--|--|--|--|--|--|--|--|---|--------|--|--|-----|-----|--|--|--|--|--|--|--|--|--|--|--|-----|--|--|--|--|--|
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td> </tr> <tr> <td style="text-align: center;">PD</td><td colspan="14" style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">3 ч</td><td colspan="14"></td> </tr> </table> | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | PD | - | | | | | | | | | | | | | | 3 ч | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PD | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td colspan="11" style="text-align: center;">PLL_DIV</td> <td style="text-align: center;">-</td> <td colspan="4" style="text-align: center;">REFSEL</td> </tr> <tr> <td colspan="11" style="text-align: center;">3 ч</td> <td></td> <td colspan="4" style="text-align: center;">3 ч</td> </tr> </table> | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | PLL_DIV | | | | | | | | | | | - | REFSEL | | | | 3 ч | | | | | | | | | | | | 3 ч | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PLL_DIV | | | | | | | | | | | - | REFSEL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 ч | | | | | | | | | | | | 3 ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PD | 31 | Бит включения режима PowerDown | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 Нет действий | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 Запись единицы включает режим | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PLL_DIV | 15-8 | Поле задания значения для расчета коэффициента N делителя сигнала на выходе блока PLL. Коэффициент N рассчитывается по формуле $N = 2 \times (PLL_DIV + 1)$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| REFSEL | 1-0 | Поле выбора источника опорного сигнала PLL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 00 Сигнал на входе микроконтроллера XI_OSC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 01 Сигнал на выводе микроконтроллера A0 (альтернативная функция CLK_USB) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 10 Сигнал на выходе блока USB (60 МГц). При переходе USB в режим Suspend сигнал может быть нестабильным | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| – | 30-16, 6-2 | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.2.7 – Регистр задания выходного делителя PLL

| PLL_OD | | 8003_0098h | Сброс: 00000000h |
|---|------|----------------------|------------------|
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | - | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | - | |
| | | PLL_OD | |
| | | 3 ч | |
| Поле | Биты | Описание | |
| PLL_OD | 1-0 | Выходной делитель NO | |
| | | 00 | Выключен |
| | | 01 | 1/2 |
| | | 10 | 1/4 |
| 11 | 1/4 | | |
| - | 31-2 | Зарезервировано | |

Таблица А.2.8 – Регистр задания опорного делителя PLL

| PLL_NR | | 8003_009Ch | Сброс: 00000000h |
|---|------|---|------------------|
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | - | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | - | |
| | | R_PLL | |
| | | 3 ч | |
| Поле | Биты | Описание | |
| R_PLL | 4-0 | Значение опорного делителя. Вычисляется по формуле $NR = R_PLL + 2$ | |
| - | 31-5 | Зарезервировано | |

Таблица А.2.9 – Регистр задания делителя обратной связи PLL

| | | | |
|---|--|------------|------------------|
| PLL_NF | | 8003_00A0h | Сброс: 00000000h |
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | - | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | - | |
| | | F_PLL | |
| | | 3 ч | |

Окончание таблицы А.2.9

| Поле | Биты | Описание |
|-------|------|---|
| F_PLL | 8-0 | Значение делителя обратной связи. Вычисляется по формуле $NF = F_PLL + 2$ Значение выходной частоты PLL вычисляется с использованием значений опорного и выходного делителей, а также делителя обратной связи по формуле $FOUT = (FIN \times NF) / (NO \times NR)$, где FIN – входная частота PLL При расчете коэффициентов деления PLL должны выполняться следующие условия: - $3,2 \text{ МГц} < fin < 150 \text{ МГц}$; - $800 \text{ КГц} < fref < 8\text{МГц}$; - $200 \text{ МГц} < fvco < 500\text{МГц}$, где частота фазового детектора вычисляется по формуле $fref = fin / (2 \times NR)$, а частота VCO вычисляется по формуле $fvco = fin \times (NF / NR)$ |
| – | 31-9 | Зарезервировано |

Таблица А.2.10 – Регистр настройки контроллера внешней памяти

| Поле | Биты | Описание |
|-------------|------------|--|
| EXT_MEM_CFG | 8003_00A4h | Сброс: 80000007h |
| 8/16 BIT | 31 | CE_MASK |
| CE_MASK | 17-9 | Маска адреса для сигналов RamCen [1] и RamOen [1]. Сигналы RamCen вычисляются по следующей схеме: $RamCen[1] = \text{not} \{ ExtMemCE_MASK \& ExtMem_ADDR[19 : 11] \}$ $RamCen[0] = \text{not} RamCen[1]$ ExtMem_ADDR[17 : 10] – выходной адрес контроллера внешней памяти |
| RWWS | 8-6 | Длительность цикла переключения шины. Поле задает паузу между циклами чтения и записи на внешней шине в количестве тактов сигнала системной частоты. Значение 000b соответствует одному такту, значение 111b – восьми |
| WRITEWS | 5-3 | Длительность цикла записи слова данных. Поле задает длительность цикла записи на внешней шине в количестве тактов сигнала системной частоты. Значение 000b соответствует одному такту, значение 111b – восьми |

Окончание таблицы А.2.10

| Поле | Биты | Описание |
|--------|-------|---|
| READWS | 2-0 | Длительность цикла чтения слова данных. Поле задает длительность цикла чтения на внешней шине в количестве тактов сигнала системной частоты. Значение 000b соответствует одному такту, значение 111b – восьми |
| – | 30-18 | Зарезервировано |

Таблица А.2.11 – Регистры 0, 1 и 2 настройки блоков АЦП

| ADC_CTRL0 | | 8003_00A8h | Сброс: 00000000h | | | | | | | | | | | | |
|------------------|-----------------------------------|--|------------------|----|----|-------------|--------------|-----------|----|----|----|----|----|-------------|--------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIV_ADC3 | | | | | | DIVEN_ADC3 | CLK_EN_ADC3 | DIV_ADC2 | | | | | | DIVEN_ADC2 | CLK_EN_ADC2 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV_ADC1 | | | | | | DIVEN_ADC1 | CLK_EN_ADC1 | DIV_ADC0 | | | | | | DIVEN_ADC0 | CLK_EN_ADC0 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| ADC_CTRL1 | | 8003_00ACh | Сброс: 00000000h | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIV_ADC7 | | | | | | DIVEN_ADC7 | CLK_EN_ADC7 | DIV_ADC6 | | | | | | DIVEN_ADC6 | CLK_EN_ADC6 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV_ADC5 | | | | | | DIVEN_ADC5 | CLK_EN_ADC5 | DIV_ADC4 | | | | | | DIVEN_ADC4 | CLK_EN_ADC4 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| ADC_CTRL2 | | 8003_013Ch | Сброс: 00000000h | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIV_ADC11 | | | | | | DIVEN_ADC11 | CLK_EN_ADC11 | DIV_ADC10 | | | | | | DIVEN_ADC10 | CLK_EN_ADC10 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV_ADC9 | | | | | | DIVEN_ADC9 | CLK_EN_ADC9 | DIV_ADC8 | | | | | | DIVEN_ADC8 | CLK_EN_ADC8 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| DIV_ADCn | 31-26, 23-18, 15-10, 7-2 | Поле задания коэффициента делителя частоты тактового сигнала, подаваемого на АЦП | | | | | | | | | | | | | |

Окончание таблицы А.2.11

| Поле | Биты | Описание | |
|----------------|---------------|--|--|
| DIVEN_ ADCn | 25,17, 9,1 | Бит включения делителя | |
| | | 0 | Частота тактового сигнала равна системной частоте |
| | | 1 | Частота тактового сигнала определяется по формуле $F_{sysclk}/(2 \times (X+1))$, X – значение поля DIV_ADCn |
| CLKEN_ ADCn | 24,16, 8,0 | Бит включения тактового сигнала (n от 0 до 11) | |
| | | 0 | Выключен |
| | | 1 | Включен |

Таблица А.2.12 – Регистр синхронизации блоков ШИМ

| Поле | Биты | Описание |
|--|------|---|
| <p>PWM_SYNC 8003_00B0h Сброс: 00000000h</p> | | |
| TBCLKSYNC | 8-0 | <p>Биты сброса счетчиков предварительных делителей блоков ШИМ.</p> <p>Нулевой бит поля TBCLKSYNC сбрасывает счетчик блока ШИМ0, восьмой бит – сбрасывает счетчик блока ШИМ8</p> <p>0 Сброс бита сбрасывает соответствующий счетчик</p> <p>1 Установка бита задает режим счета</p> |
| – | 31-9 | Зарезервировано |

Таблица А.2.13 – Регистр настройки синхронизации блоков ШИМ

| Поле | Биты | Описание |
|--|------|--|
| <p>PWM_CTRL 8003_00B4h Сброс: 00000000h</p> | | |
| SYNCR EG | 3 | <p>Генератор импульса синхронизации</p> <p>0 Нет действий</p> <p>1 Запись единицы генерирует импульс синхронизации SYNCI на входе блока ШИМ1</p> |

Окончание таблицы А.2.13

| Поле | Биты | Описание |
|----------------|------|---|
| CAPSY NCSEL | 2 | Бит выбора источника синхронизации для блоков захвата 3, 4 и 5 (см. рисунок 14.3) |
| | | 0 Синхронизация от блока захвата 2 |
| | | 1 Синхронизация от блока ШИМ0 |
| SYNCS ELECT | 1-0 | Выбор схемы синхронизации модулей ШИМ |
| | | 00 Последовательное включение: (ШИМ0–ШИМ1–ШИМ2–ШИМ3–ШИМ4–ШИМ5–ШИМ6–ШИМ7–ШИМ8) |
| | | 01 Группы: (ШИМ0–ШИМ1–ШИМ2) и (ШИМ0–ШИМ3–ШИМ4–ШИМ5–ШИМ6–ШИМ7–ШИМ8) |
| | | 10 Группы: (ШИМ0–ШИМ1–ШИМ2–ШИМ3–ШИМ4–ШИМ5) и (ШИМ0–ШИМ6–ШИМ7–ШИМ8) |
| | | 11 Группы: (ШИМ0–ШИМ1–ШИМ2), (ШИМ0–ШИМ3–ШИМ4–ШИМ5) и (ШИМ0–ШИМ6–ШИМ7–ШИМ8) |
| – | 31-4 | Зарезервировано |

Таблица А.2.14 – Регистр настройки системной частоты микроконтроллера

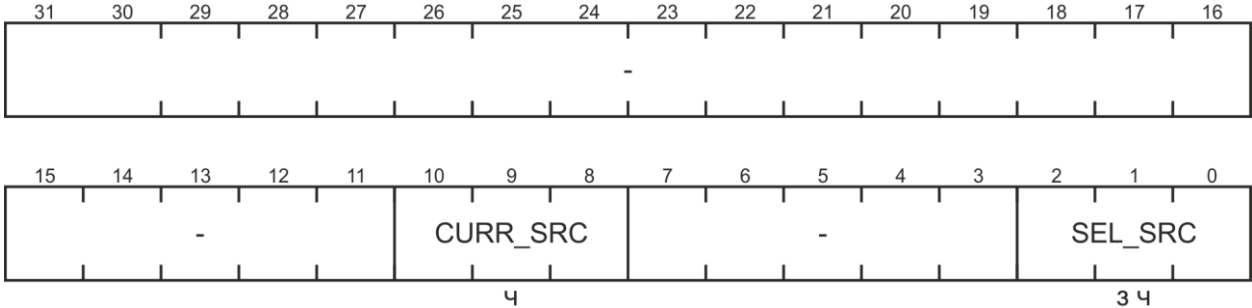
| Поле | Биты | Описание |
|---|-------------------------|---|
| <p>SYS_CLK 8003_00B8h Сброс: 00000000h</p>  | | |
| CURR_SRC | 10-8 | Поле-индикатор источника синхросигнала. Значение поля указывает на блок, который в текущий момент времени является источником синхросигнала (см. SEL_SRC) |
| SEL_SRC | 2-0 | Поле выбора источника синхросигнала |
| | | 000 Выбор источника зависит от состояния сигнала на входе микроконтроллера CPE_pad |
| | | 0 Блок POR |
| | | 1 Вход микроконтроллера XI_OSC |
| | | 001 Блок POR |
| | | 010 Вход микроконтроллера XI_OSC |
| | | 011 Блок PLL |
| | | 100 Блок PLL с делителем PLLDIV (см. регистр PLL_CTRL) |
| | | 101 Блок USB (60 МГц) |
| 110 Вывод микроконтроллера A0 (для вывода должна быть включена альтернативная функция CLK_USB) | | |
| 111 Вывод микроконтроллера A8 или B3 (для вывода должна быть включена альтернативная функция MII_TXCLK) | | |
| – | 31-16, 15-11, 7-3 | Зарезервировано |

Таблица А.2.15 – Регистр настройки разрешения тактовых сигналов периферии

| APB_CLK | | 8003_00BCh | Сброс: 00000000h | | | | | | | | | | | | |
|---------|-----------------|---|--|----|----|----------------|-----------|----|----|----|-------|----------|-------|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | ADC EN | - | | | I2CEN | WD EN | PWMEN | | |
| | | | | | | | 3 ч | | | | 3 ч | 3 ч | 3 ч | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PWMEN | | | | | | E CMP EN | - | | | | | QEPEN | | - | |
| 3 ч | | | | | | 3 ч | | | | | | 3 ч | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| ADCEN | 24 | Бит разрешения тактирования контроллера АЦП | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | |
| I2CEN | 21-20 | Бит разрешения тактирования блоков I2C | | | | | | | | | | | | | |
| | | Младший бит поля управляет тактированием блока I2C0, старший бит – блоком I2C1. | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | |
| WDEN | 19 | Бит разрешения тактирования сторожевого таймера | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | |
| PWMEN | 18-10 | Биты разрешения тактирования блоков ШИМ. | | | | | | | | | | | | | |
| | | Младший бит поля PWMEN управляет тактированием блока ШИМ0, старший бит – блоком ШИМ8. | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | |
| ECMPEN | 8 | Бит разрешения тактирования аналогового компаратора | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | |
| QEPEN | 2-1 | Биты разрешения тактирования квадратурных декодеров | | | | | | | | | | | | | |
| | | 00 | Тактирование обоих декодеров запрещено | | | | | | | | | | | | |
| | | 01 | Разрешено тактирование декодера 1 | | | | | | | | | | | | |
| | | 10 | Разрешено тактирование декодера 2 | | | | | | | | | | | | |
| | | 11 | Тактирование обоих декодеров разрешено | | | | | | | | | | | | |
| - | 31-25,23-22,8-3 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.2.16 – Регистр настройки тактовых сигналов блоков UART

| UART_CLK | | 8003_00C0h | Сброс: 00000000h | | | | | | | | | | | | |
|-------------|-----------------------------------|--|--|----|----|-------------|-------------|-----------|----|----|----|----|----|-------------|-------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIV_UART3 | | | | | | DIVEN_UART3 | CLKEN_UART3 | DIV_UART2 | | | | | | DIVEN_UART2 | CLKEN_UART2 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV_UART1 | | | | | | DIVEN_UART1 | CLKEN_UART1 | DIV_UART0 | | | | | | DIVEN_UART0 | CLKEN_UART0 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| DIV_UARTn | 31-26, 23-18, 15-10, 7-2 | Поле задания коэффициента делителя частоты тактового сигнала, подаваемого на UARTn | | | | | | | | | | | | | |
| DIVEN_UARTn | 25,17,9, 1 | Бит включения делителя | | | | | | | | | | | | | |
| | | 0 | Частота тактового сигнала равна системной частоте | | | | | | | | | | | | |
| | | 1 | Частота тактового сигнала определяется по формуле $F_{osc}/(2 \times (X+1))$, X – значение поля DIV_UARTn | | | | | | | | | | | | |
| CLKEN_UARTn | 24,16,8, 0 | Бит включения тактового сигнала (n от 0 до 11) | | | | | | | | | | | | | |
| | | 0 | Выключен | | | | | | | | | | | | |
| | | 1 | Включен | | | | | | | | | | | | |

Таблица А.2.17 – Регистр настройки тактовых сигналов блоков SPI

| SPI_CLK | | 8003_00C4h | Сброс: 00000000h | | | | | | | | | | | | |
|-------------|-----------------------------------|--|--|----|----|------------|------------|----------|----|----|----|----|----|------------|------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIV_SPI3 | | | | | | DIVEN_SPI3 | CLKEN_SPI3 | DIV_SPI2 | | | | | | DIVEN_SPI2 | CLKEN_SPI2 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV_SPI1 | | | | | | DIVEN_SPI1 | CLKEN_SPI1 | DIV_SPI0 | | | | | | DIVEN_SPI0 | CLKEN_SPI0 |
| 3 4 | | | | | | 3 4 | 3 4 | 3 4 | | | | | | 3 4 | 3 4 |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| DIV_SPIIn | 31-26, 23-18, 15-10, 7-2 | Поле задания коэффициента делителя частоты тактового сигнала, подаваемого на SPIIn | | | | | | | | | | | | | |
| DIVEN_SPIIn | 25,17,9,1 | Бит включения делителя | | | | | | | | | | | | | |
| | | 0 | Частота тактового сигнала равна системной частоте | | | | | | | | | | | | |
| | | 1 | Частота тактового сигнала определяется по формуле $F_{osc}/(2 \times (X+1))$, X – значение поля DIV_SPIIn | | | | | | | | | | | | |
| CLKEN_SPIIn | 24,16,8,0 | Бит включения тактового сигнала (n от 0 до 11) | | | | | | | | | | | | | |
| | | 0 | Выключен | | | | | | | | | | | | |
| | | 1 | Включен | | | | | | | | | | | | |

Таблица А.2.18 – Регистр сброса блоков периферии

| Поле | | Бит | Описание |
|--------------|--|-------|---------------------------------|
| ETHERNET RST | | 15 | Бит сброса контроллера Ethernet |
| SPIRST | | 14 | Бит сброса контроллера SPI3 |
| | | 13 | Бит сброса контроллера SPI2 |
| | | 12 | Бит сброса контроллера SPI1 |
| | | 11 | Бит сброса контроллера SPI0 |
| UARTRST | | 10 | Бит сброса блока UART3 |
| | | 9 | Бит сброса блока UART2 |
| | | 8 | Бит сброса блока UART1 |
| | | 7 | Бит сброса блока UART0 |
| TIMERRST | | 6 | Бит сброса таймера 2 |
| | | 5 | Бит сброса таймера 1 |
| | | 4 | Бит сброса таймера 0 |
| USBPHYRST | | 3 | Бит сброса контроллера USB |
| I2CRST | | 2 | Бит сброса контроллера I2C1 |
| | | 1 | Бит сброса контроллера I2C0 |
| WDRST | | 0 | Бит сброса сторожевого таймера |
| – | | 31-16 | Зарезервировано |

Примечание – Для всех битов справедливо: запись нуля в бит вызывает сброс соответствующего модуля, установка бита активирует соответствующий модуль.

Таблица А.2.19 – Регистр два сброса блоков периферии

| Поле | | Бит | Описание |
|--------|--|-----|----------------------------|
| CAPRST | | 17 | Бит сброса контроллера CAP |
| | | 16 | Бит сброса контроллера CAP |
| PWMRST | | 15 | Бит сброса таймера PWM1 |
| | | 14 | Бит сброса таймера PWM2 |
| | | 13 | Бит сброса таймера PWM3 |
| QEPRST | | 12 | Бит сброса таймера QE1 |
| | | 11 | Бит сброса таймера QE2 |
| | | 10 | Бит сброса таймера QE3 |

Окончание таблицы А.2.19

| Поле | Бит | Описание |
|---------|-------|-------------------------------------|
| ECMPRST | 17 | Бит сброса аналогового компаратора |
| CAPRST | 16 | Бит сброса модуля захвата CAP5 |
| | 15 | Бит сброса модуля захвата CAP4 |
| | 14 | Бит сброса модуля захвата CAP3 |
| | 13 | Бит сброса модуля захвата CAP2 |
| | 12 | Бит сброса модуля захвата CAP1 |
| | 11 | Бит сброса модуля захвата CAP0 |
| PWMRST | 10 | Бит сброса модуля ШИМ8 |
| | 9 | Бит сброса модуля ШИМ7 |
| | 8 | Бит сброса модуля ШИМ6 |
| | 7 | Бит сброса модуля ШИМ5 |
| | 6 | Бит сброса модуля ШИМ4 |
| | 5 | Бит сброса модуля ШИМ3 |
| | 4 | Бит сброса модуля ШИМ2 |
| | 3 | Бит сброса модуля ШИМ1 |
| | 2 | Бит сброса модуля ШИМ0 |
| QEPRST | 1 | Бит сброса квадратурного декодера 2 |
| | 0 | Бит сброса квадратурного декодера 1 |
| – | 31-18 | Зарезервировано |

Примечание – Для всех битов справедливо: запись нуля в бит вызывает сброс соответствующего модуля, установка бита активирует соответствующий модуль.

Таблица А.2.20 – Регистр настройки USB PHY

| Поле | Бит | Описание |
|--------|------|---|
| CLKEN | 2 | Бит разрешения синхронизации USBPHY 0 Запрещено 1 Разрешено |
| CLKSEL | 1 | Бит выбора источника сигнала тактирования 0 Вход CLK_USB 1 Сигнал OSC_CLK |
| FSEL | 0 | Бит выбора частоты сигнала тактирования 0 12 МГц 1 24 МГц |
| – | 31-3 | Зарезервировано |

Таблица А.2.21 – Регистр настройки тактовых сигналов блоков UART и SPI

| UART_SPI_CLK_SEL | | 8003_0138h | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-----------------------------------|--|------------------|----------|----------|----------|----------|----------|----------|----------|----------|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 12.5%;">SEL_SPI</td> <td style="width: 12.5%;">SEL_SPI</td> <td style="width: 12.5%;">SEL_SPI</td> <td style="width: 12.5%;">SEL_SPI</td> <td style="width: 12.5%;">SEL_UART</td> <td style="width: 12.5%;">SEL_UART</td> <td style="width: 12.5%;">SEL_UART</td> <td style="width: 12.5%;">SEL_UART</td> </tr> <tr> <td>3</td> <td>2</td> <td>1</td> <td>0</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>3 ч</td> <td>3 ч</td> <td>3 ч</td> <td>3 ч</td> <td>3 ч</td> <td>3 ч</td> <td>3 ч</td> <td>3 ч</td> </tr> </table> | | | | SEL_SPI | SEL_SPI | SEL_SPI | SEL_SPI | SEL_UART | SEL_UART | SEL_UART | SEL_UART | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 ч | 3 ч | 3 ч | 3 ч | 3 ч | 3 ч | 3 ч | 3 ч |
| SEL_SPI | SEL_SPI | SEL_SPI | SEL_SPI | SEL_UART | SEL_UART | SEL_UART | SEL_UART | | | | | | | | | | | | | | | | | | | | |
| 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | |
| 3 ч | 3 ч | 3 ч | 3 ч | 3 ч | 3 ч | 3 ч | 3 ч | | | | | | | | | | | | | | | | | | | | |
| SEL_SPI _n | 15-14, 13-12, 11-10, 9-8 | Поле выбора источника тактового сигнала для блока SPI _n 00 Системная частота 01 Осциллятор 10 Вход CLK_USB (Вывод А.0) 11 Частота контроллера USB (60 МГц) | | | | | | | | | | | | | | | | | | | | | | | | | |
| SEL_UART _n | 7-6, 5-4, 3-2, 1-0 | Поле выбора источника тактового сигнала для блока UART _n 00 Системная частота 01 Осциллятор 10 Вход CLK_USB (Вывод А.0) 11 Частота контроллера USB (60 МГц) | | | | | | | | | | | | | | | | | | | | | | | | | |
| – | 31-16 | – | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.2.22 – Регистр стирания флеш-памяти

| FLASH_FULL_ERASE | | 8003_0164h | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|------|--|------------------|--|--|--|--|--|--|--|--|--|--|--|----------------|-----|--|--|----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-----|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;">ERASE FLASH</td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3 ч</td> </tr> </table> | | | | | | | | | | | | | | | | | | | ERASE FLASH | | | | | | | | | | | | | | | | | 3 ч |
| | | | | | | | | | | | | | | | ERASE FLASH | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 3 ч | | | | | | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ERASE_FLASH | 0 | Бит полного стирания флеш-памяти. Запись единицы активизирует стирание памяти. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| – | 31-1 | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

А.3 Регистры контроллера CAN

Таблица А.3.1 – Регистр управления частотой

| CLC | | 8007_0000h | Сброс: 00000003h |
|---|------|--------------------------------|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0; display: flex; justify-content: space-between;"> <div style="width: 120px;"></div> <div style="width: 30px; text-align: center;">DIS S</div> <div style="width: 30px; text-align: center;">DIS R</div> </div> | | | |
| 4 3 4 | | | |
| Поле | Бит | Описание | |
| DISS | 1 | Бит состояния контроллера CAN | |
| | | 0 | Включен |
| | | 1 | Выключен |
| DISR | 0 | Бит выключения контроллера CAN | |
| | | 0 | Нет действий |
| | | 1 | Запись единицы запускает механизм выключения |
| – | 31-2 | Зарезервировано | |
| <p>Примечание – Когда контроллер CAN находится в выключенном состоянии, только регистр CLC доступен для записи и чтения, доступ к остальным регистрам не возможен.</p> | | | |

Таблица А.3.2 – Регистр идентификации

| ID | | 8007_0008h | Сброс: 002BC051h |
|--|-------|---|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0; display: flex; justify-content: center; align-items: center;"> MOD_NUMBER </div> | | | |
| 4 | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0; display: flex; justify-content: space-between;"> <div style="width: 60%; text-align: center;">MOD_TYPE</div> <div style="width: 30%; text-align: center;">MOD_REV</div> </div> | | | |
| 4 3 4 | | | |
| Поле | Биты | Описание | |
| MOD_NUMBER | 31-16 | Идентификационный номер контроллера CAN | |
| MOD_TYPE | 15-8 | Разрядность контроллера CAN | |
| MOD_REV | 7-0 | Число модификаций контроллера CAN | |

Таблица А.3.3 – Регистр делителя

| FDR | | 8007_000Ch | | | | | | | | | | | | Сброс: 00000000h | | |
|---------|-----------------|--|---|----|------|----|--------|----|----|----|----|----|----|------------------|----|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| DIS CLK | | EN HW | | - | | | RESULT | | | | | | | | | |
| 3 ч | | 3 ч | | | | | 4 | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| DM | | - | | | STEP | | | | | | | | | | | |
| 3 ч | | | | | 3 ч | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | |
| DISCLK | 31 | Бит запрета внутреннего тактирования | | | | | | | | | | | | | | |
| | | 0 | Генерирование сигнала F _{scan} разрешено | | | | | | | | | | | | | |
| | | 1 | Генерирование сигнала F _{scan} запрещено | | | | | | | | | | | | | |
| ENHW | 30 | Бит контроля синхронизации. Это бит аппаратно удерживается в сброшенном состоянии и не может быть установлен | | | | | | | | | | | | | | |
| RESULT | 25-16 | Счетчик делителя частоты | | | | | | | | | | | | | | |
| DM | 15-14 | Поле задания режима делителя частоты | | | | | | | | | | | | | | |
| | | 00 | Счетчик выключен. Синхросигнал F _{out} не генерируется. Сигнал сброса внешнего делителя в состоянии логической единицы. Поле RESULT не меняется | | | | | | | | | | | | | |
| | | 01 | Нормальный режим работы. Сигнал сброса внешнего делителя в состоянии логического нуля. При активации режима поле RESULT загружается значением 3FFh. Далее периодически загружается значением из STEP. Формируется сигнал F _{out} | | | | | | | | | | | | | |
| | | 10 | Режим дробного деления. Сигнал сброса внешнего делителя в состоянии логического нуля. При активации режима поле RESULT загружается значением 3FFh. Далее периодически загружается значением из STEP. Формируется сигнал F _{out} | | | | | | | | | | | | | |
| STEP | 9-0 | Шаг делителя. Поле хранит значение, которое загружается в RESULT при переполнении счетчика делителя | | | | | | | | | | | | | | |
| - | 29-26, 13-10 | Зарезервировано | | | | | | | | | | | | | | |

Таблица А.3.4 – Регистр списка №0 и регистр свободного списка x (x от 1 до 7)

| LIST0 | | 8007_0100h | | | | | | | | | | | | Сброс: 007F7F00h | |
|-------|----|------------|----|----|----|----|-------|-------|------|----|----|----|----|------------------|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | EMPTY | | SIZE | | | | | | |
| | | | | | | | 4 | | 4 | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| END | | | | | | | | BEGIN | | | | | | | |
| 4 | | | | | | | | 4 | | | | | | | |

Окончание таблицы А.3.4

| LISTx | | Сброс: 01000000h |
|-------|-------|---|
| | | |
| Поле | Биты | Описание |
| EMPTY | 24 | Индикатор пустого списка |
| | | 0 В списке есть как минимум один элемент |
| | | 1 Список пуст |
| SIZE | 23-16 | Размер списка. Количество элементов (объектов сообщений) в списке. Значение поля SIZE всегда на единицу меньше числа элементов. Если список пуст, SIZE = 00h |
| END | 15-8 | Номер объекта сообщения, находящегося последним в списке. Поле может принимать значения от 00h до FFh, согласно количеству объектов сообщений (256) |
| BEGIN | 7-0 | Номер объекта сообщения, находящегося первым в списке. Поле может принимать значения от 00h до FFh, согласно количеству объектов сообщений |
| — | 31-25 | Зарезервировано |

Таблица А.3.5 – Регистр ждущих прерываний (x от 0 до 7)

| MSPNDx | | Сброс: 00000000h |
|--------|------|---|
| | | |
| Поле | Биты | Описание |
| PND | 31-0 | Поле ждущих битов сообщений. Каждому объекту сообщения выделяется один бит. Биты устанавливаются только аппаратно. Установленные биты сбрасываются аппаратно по окончании обслуживания запроса прерывания или могут быть сброшены в любой момент программно |

Таблица А.3.6– Регистр индекса сообщения (x – от 0 до 7)

| MSIDx | | | | | | | | | | | | Сброс: 00000020h | | | |
|--------------|------|--|----|----|----|----|----|----|----|-------|----|------------------|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | | INDEX | | | | | |
| ч | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| INDEX | 5-0 | Поле номера ждущего бита. Если в регистре MSPND есть установленные биты, которые не маскируются соответствующими битами регистра MSIMASK, то поле INDEX будет указывать на самый младший из них. Если в регистре MSPND нет установленных битов или они замаскированы, то в поле INDEX будет находиться значение 20h, указывающее на бит 31 регистра MSPND | | | | | | | | | | | | | |
| – | 31-6 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.3.7 – Регистр маски индекса сообщения

| MSIMASK | | | | | | | | | | | | 8007_01C0h | | | | Сброс: 00000000h | | | |
|----------------|------|--|----|----|----|----|----|----|----|----|----|------------|----|----|----|------------------|--|--|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | |
| IM | | | | | | | | | | | | | | | | | | | |
| 3 ч | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | |
| IM | | | | | | | | | | | | | | | | | | | |
| 3 ч | | | | | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | | | | |
| IM | 31-0 | Маска для ждущих битов сообщений. Учитывается состояние только тех бит регистра MSPND, для которых в поле IM установлены соответствующие биты | | | | | | | | | | | | | | | | | |

Таблица А.3.8 – Регистр панели команд

| PANCTR | | 8007_01C4h | Сброс: 00000301h |
|--------|-------|---|--|
| | | | |
| Поле | Биты | Описание | |
| PANAR2 | 31-24 | Панель аргумента 2 (см. таблицу А.3.9) | |
| PANAR1 | 23-16 | Панель аргумента 1 (см. таблицу А.3.9) | |
| RBUSY | 9 | Флаг занятости панелей аргументов | |
| | | 0 | Нет действий |
| | | 1 | Выполняется команда списка, результат выполнения которой будет записан в PANAR1 и PANAR2 |
| BUSY | 8 | Флаг занятости панелей аргументов | |
| | | 0 | Панели готовы для записи |
| | | 1 | Панели заняты – ожидают записи по окончании выполнения команды |
| PANCMD | 7-0 | Поле команды (см. таблицу А.3.9). После выполнения команды в это поле записывается 00h | |
| – | 15-10 | Зарезервировано | |

Таблица А.3.9 – Коды команд работы со списками

| PANCMD | Поле PANAR 2 | Поле PANAR 1 | Описание команды |
|--------|---|--------------|--|
| 00h | – | – | Нет операции. Никаких действий не выполняется |
| 01h | Результат: бит 7 – ошибка, бит 6 – не определен | – | Инициализация списков. Запуск инициализации для очистки битовых полей CTRL и LIST всех объектов сообщений. Регистры LIST0 – LIST8 устанавливаются в свои значения после сброса. Это приводит к переносу всех объектов сообщений в список №0 (список нераспределенных объектов сообщений). Инициализация списков требует, чтобы биты INIT и SSE регистра NCR были установлены для обоих узлов. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – Инициализация завершена успешно; - 1 – Инициализация не завершена, поскольку не все биты INIT и SSE были установлены. Команда инициализации списков автоматически запускается при каждом сбросе контроллера CAN, за исключением случая, когда все регистры объектов сообщений уже сброшены |

Окончание таблицы А.3.9

| PAN CMD | Поле PANAR2 | Поле PANAR1 | Описание команды |
|-----------------|---|--|--|
| 02h | Аргумент: номер списка | Аргумент: номер объекта сообщения | Статическое занесение объекта сообщения в список. Объект сообщения переносится из текущего списка в список, указанный полем PANAR2 и добавляется в его конец. Эта команда также используется для деразпределения объекта сообщения, т. е. переноса его в список № 0 (если PANAR2 равно 00h) |
| 03h | Аргумент: номер списка Результат: бит 7 – ошибка, бит 6– не определен | Результат: номер объекта сообщения | Динамическое занесение объекта сообщения в список. Первый объект сообщения списка №0 переносится в список, указанный полем PANAR2, и добавляется в его конец. Номер объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – Операция выполнена; - 1 – Операция не выполнена – список №0 пуст |
| 04h | Аргумент: номер объекта сообщения | Аргумент: текущий номер объекта сообщения | Перемещение по списку вверх. Перенос объекта сообщения с номером PANAR1 на одну позицию выше, чем расположен объект сообщения с номером PANAR2 |
| 05h | Аргумент: номер объекта сообщения Результат: бит 7 – ошибка, бит 6 – не определен | Результат: номер добавлен- ного объекта сообщения | Динамическая вставка в список. Первый объект сообщений списка №0 вставляется на одну позицию выше, чем расположен объект сообщения с номером PANAR2. Номер добавленного объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – Операция выполнена; - 1 – Операция не выполнена – список №0 пуст |
| 06h | Аргумент: номер объекта сообщения | Аргумент: текущий номер объекта сообщения | Перемещение по списку вниз. Перенос объекта сообщения с номером PANAR1 на одну позицию ниже, чем расположен объект сообщения с номером PANAR2 |
| 07h | Аргумент: номер объекта сообщения Результат: бит 7– ошибка, бит 6 – не определен | Результат: номер добавлен- ного объекта сообщения | Динамическая вставка в список. Первый объект сообщения списка №0 вставляется на одну позицию ниже, чем расположен объект сообщения с номером PANAR2. Номер добавленного объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – Операция выполнена; - 1 – Операция не выполнена – список №0 пуст |
| 08h – FFh | – | – | Зарезервировано |

Таблица А.3.10 – Регистр управления

| | | | |
|------------|-------------|---|------------------|
| MCR | | 8007_01C8h | Сброс: 00000000h |
| | | | |
| Поле | Биты | Описание | |
| MPSEL | 15-12 | Поле задания позиции ждущего бита сообщения после приема/передачи сообщения | |
| – | 31-16, 11-0 | Зарезервировано | |

Таблица А.3.11 – Регистр прерываний

| | | | |
|-------------|-------|---|------------------|
| MITR | | 8007_01CCh | Сброс: 00000000h |
| | | | |
| Поле | Биты | Описание | |
| IT | 15-0 | Поле генератора прерываний. Каждый бит поля связан с одной из линий прерываний. Номера битов от 0 до 15 соответствуют номерам линий прерываний. Для того, чтобы сгенерировать одно или несколько прерываний, следует установить соответствующие биты. Установленные биты сбрасываются аппаратно | |
| – | 31-16 | Зарезервировано | |

Таблица А.3.12 – Регистр управления узла

| NCR | | Сброс: 00000001h |
|---|-----|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> CAL MCCE-CAN DISALI ELEC IETRIEINIT </div> <div style="display: flex; justify-content: space-between; padding-top: 5px;"> 3 43 4-3 43 43 43 43 4 </div> | | |
| Поле | Бит | Описание |
| CALM | 7 | Бит включения режима анализа узла |
| | | 0 Режим выключен |
| | | 1 Установка бита включает режим анализа узла. В этом режиме сообщения могут только приниматься, бит подтверждения не посылается после успешного приема сообщения, флаг активной ошибки посылается рецессивным вместо доминантного. На линии отправки сообщений поддерживается высокий уровень сигнала |
| | | Бит может быть установлен только, если установлен бит INIT |
| CCE | 6 | Бит разрешения изменения конфигурации узла. Управляет доступом к регистрам NBTRx, NPCRx и NECNTx |
| | | 0 Только чтение |
| | | 1 Полный доступ |
| CANDIS | 4 | Бит выключения узла |
| | | 0 Сброс бита включает узел |
| | | 1 Установка бита выключает узел. Сначала узел переходит в состояние «простоя» или «отключен от шины», далее аппаратно устанавливается бит INIT и, если разрешено, генерируется прерывание ALERT |
| ALIE | 3 | Бит разрешения прерывания ALERT от узла |
| | | 0 Запрещено |
| | | 1 Разрешено |
| LECIE | 2 | Бит разрешения прерывания от узла при обнаружении кода последней ошибки |
| | | 0 Запрещено |
| | | 1 Разрешено |
| TRIE | 1 | Бит разрешения прерывания от узла по окончании передачи/приема |
| | | 0 Запрещено |
| | | 1 Разрешено |
| INIT | 0 | Инициализация узла |
| | | 0 Сброс бита разрешает участие узла в трафике CAN шины. Узел ожидает последовательность из 11 рецессивных бит на шине и включается в трафик. Если на момент сброса бита INIT узел находился в состоянии «отключен от шины», начинается процесс выхода из этого состояния в следующем порядке: получение 128 последовательностей бит (каждая из 11 рецессивных бит), выход из состояния «отключен от шины», включение в трафик |

Окончание таблицы А.3.12

| Поле | Бит | Описание | |
|------|---------|-----------------|---|
| INIT | 0 | 1 | Установка бита INIT прекращает участие узла в трафике. Все текущие передачи останавливаются, линии передач переходят в рецессивное состояние. Если на момент установки бита INIT узел находился в состоянии «отключен от шины», процесс выхода из этого состояния продолжается до его завершения. Далее узел остается неактивным до тех пор, пока установлен бит INIT |
| – | 31-8, 5 | Зарезервировано | |

Таблица А.3.13 – Регистр состояния узла

| NSR | | Сброс: 00000000h | |
|------|------|---|--|
| | | | |
| Поле | Биты | Описание | |
| LOE | 9 | Флаг ошибки номера списка | |
| | | 0 | Ошибок не обнаружено |
| | | 1 | Обнаружена ошибка при фильтрации принимаемого сообщения. В регистре MOSTAT объекта сообщения обнаружен неверный номер списка |
| | | Бит должен сбрасываться программно записью нуля | |
| LLE | 8 | Флаг ошибки списка | |
| | | 0 | Ошибок не обнаружено |
| | | 1 | Обнаружена ошибка при фильтрации принимаемого сообщения. Количество элементов списка, принадлежащего узлу, отличается от указанного в поле SIZE соответствующего регистра списка |
| | | Бит должен сбрасываться программно записью нуля | |
| BOFF | 7 | Флаг состояния «отключен от шины» | |
| | | 0 | Узел не находится в состоянии «отключен от шины» |
| | | 1 | Узел находится в состоянии «отключен от шины» |
| EWRN | 6 | Флаг критического количества ошибок | |
| | | 0 | Лимит ошибок еще не достигнут |
| | | 1 | По крайней мере, один из счетчиков ошибок (REC, TEC) достиг лимита ошибок, заданного полем EWRNLVL регистра NECNT узла |

Продолжение таблицы А.3.13

| Поле | Биты | Описание | |
|-------|------|--|--|
| ALERT | 5 | Флаг предупреждения ALERT | |
| | | 0 | Нет событий |
| | | 1 | Произошло одно или несколько не взаимоисключающих событий: - модификация бита BOFF; - модификация/установка бита LOE; - установка бита LLE; - аппаратная установка бита INIT |
| | | Бит должен сбрасываться программно записью нуля | |
| RXOK | 4 | Флаг успешного приема сообщения | |
| | | 0 | Полученных сообщений нет |
| | | 1 | Сообщение получено |
| | | Бит должен сбрасываться программно записью нуля | |
| TXOK | 3 | Флаг успешной передачи сообщения | |
| | | 0 | Переданных сообщений нет |
| | | 1 | Сообщение передано без ошибок с получением подтверждения |
| | | Бит должен сбрасываться программно записью нуля | |
| LEC | 2-0 | Код последней ошибки. Поле хранит код последней из обнаруженных ошибок работы узла | |
| | | 000 | Ошибок нет |
| | | 001 | Ошибка стаффинга (заполнения, STUFF ERROR). Может быть обнаружена во время передачи шестого бита из последовательности шести одинаковых бит в поле сообщения, которое должно быть кодировано методом разрядного заполнения (заключается в том, что после передачи пяти битов одинаковой полярности, шестой бит должен иметь противоположную полярность и вставляться передатчиком в поток данных автоматически, приемник пропускает этот бит) |
| | | 010 | Ошибка формы (FORM ERROR). Обнаруживается, если: - в битовом поле фиксированного формата содержится количество битов, отличающееся от установленного; - на месте рецессивного бита находятся доминантный или наоборот. Исключение – для приемника доминантный бит в течение последнего бита поля «конец кадра» не интерпретируется как ошибка формы |
| | | 011 | Ошибка подтверждения (ACKNOWLEDGMENT ERROR). Обнаруживается передатчиком всякий раз, когда он не обнаруживает доминантный бит ACK в «области подтверждения» |
| | 100 | Разрядная ошибка или ошибка бита 1 (BIT 1 ERROR). Узел, который передает данные на шину, осуществляет мониторинг шины. Ошибка бита 1 имеет место, если при передаче рецессивного «1» бита (за исключением битов полей арбитража и подтверждения) на шине обнаруживается доминантный «0» бит | |

Окончание таблицы А.3.13

| Поле | Биты | Описание | |
|-------|------|---|---|
| LEC | 2-0 | 101 | Разрядная ошибка или ошибка бита 0 (BIT 0 ERROR). Ошибка возникает в случаях: - во время передачи сообщения (или бита подтверждения, флага активной ошибки, флага перезагрузки), узел передает доминантный бит «0», но на шине обнаруживается рецессивный «1»; - во время выхода из состояния «отключен от шины» при каждом обнаружении последовательности из 11 рецессивных битов. В этом случае, ЦП может использовать код 101 для отслеживания длительного простоя шины |
| | | 110 | Ошибка циклического избыточного кода (CRC ERROR). Передатчик по установленному алгоритму вычисляет значение контрольной суммы (CRC) для передаваемых данных и вставляет ее в сообщение. Приемник, после получения данных, вычисляет CRC по тому же алгоритму, что и передатчик, и сравнивает вычисленное значение с принятым значением. В случае несовпадения фиксируется ошибка |
| | | 111 | Код разрешения аппаратной записи в поле LEC |
| | | После аппаратной записи в поле LEC значения кода, отличного от 111b, поле становится закрытым для записи и далее центральный процессор не может изменить его состояние до тех пор, пока в это поле не будет программно записано значение 111b | |
| 31-10 | – | Зарезервировано | |

Таблица А.3.14 – Регистр указателя прерываний узла

| Поле | Биты | Описание |
|-------------|-------|--|
| NIPR | | Сброс: 00000000h |
| | | |
| CFCINP | 15-12 | Указатель линии прерывания для прерывания при переполнении счетчика фреймов узла |
| TRINP | 11-8 | Указатель линии прерывания для прерывания по окончании передачи/приема сообщения |
| LECINP | 7-4 | Указатель линии прерывания для прерывания при записи кода последней ошибки |
| ALINP | 3-0 | Указатель линии прерывания для прерывания ALERT |
| – | 31-16 | Зарезервировано |

Окончание таблицы А.3.14

Примечание – Каждый из указателей позволяет задать номер одной из 16 линий прерываний для каждого из четырех источников. Значение 00h соответствует нулевой линии прерываний, значение 01h – первой и так далее до значения FFh, которое соответствует линии 15 прерываний/

Таблица А.3.15 – Регистр управления портом узла

| NPCR | | Сброс: 00000000h | | |
|---|--|--|---|----------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px; display: flex; justify-content: space-between;"> LBM </div> | | | | |
| 3 ч | | | | |
| Поле | Бит | Описание | | |
| LBM | 8 | Бит включения режима обратной петли (Loop-Back) | | |
| | | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>Режим выключен</td> </tr> <tr> <td style="width: 20px; text-align: center;">1</td> <td>Включен режим обратной петли. В этом режиме узел подсоединяется к внутренней виртуальной CAN шине. Если для обоих узлов включен режим обратной петли, то они объединяются виртуальной CAN шиной и могут взаимодействовать друг с другом. При этом на внешних выводах узлов, соединенных с внешней физической CAN шиной, поддерживается рецессивный уровень сигнала, т. е. узлы не активны</td> </tr> </table> | 0 | Режим выключен |
| 0 | Режим выключен | | | |
| 1 | Включен режим обратной петли. В этом режиме узел подсоединяется к внутренней виртуальной CAN шине. Если для обоих узлов включен режим обратной петли, то они объединяются виртуальной CAN шиной и могут взаимодействовать друг с другом. При этом на внешних выводах узлов, соединенных с внешней физической CAN шиной, поддерживается рецессивный уровень сигнала, т. е. узлы не активны | | | |
| – | 31-9, 7-0 | Зарезервировано | | |

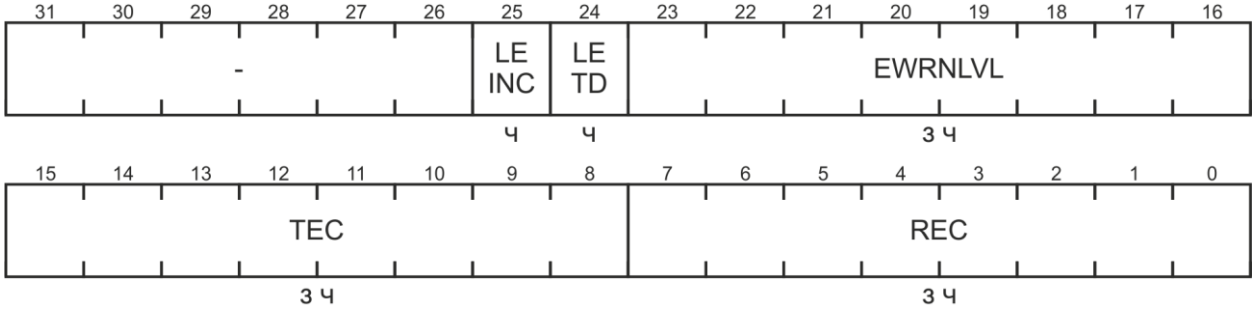
Таблица А.3.16 – Регистр синхронизации битов

| NBTR | | Сброс: 00000000h | |
|---|------|----------------------------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px; display: flex; justify-content: space-between;"> <div style="width: 100px; text-align: center;">DIV 8</div> <div style="width: 100px; text-align: center;">TSEG2</div> <div style="width: 100px; text-align: center;">TSEG1</div> <div style="width: 100px; text-align: center;">SJW</div> <div style="width: 100px; text-align: center;">BRP</div> </div> | | | |
| 3 ч 3 ч 3 ч 3 ч 3 ч | | | |
| Поле | Биты | Описание | |
| DIV8 | 15 | Делитель частоты на восемь | |
| | | 0 | Длительность кванта времени (BRP + 1), тактов частоты |
| | | 1 | Длительность кванта времени 8 × (BRP + 1), тактов частоты |

Окончание таблицы А.3.16

| Поле | Биты | Описание |
|-------|-------|---|
| TSEG2 | 14-12 | Параметр 2. Временной промежуток от точки выборки до точки передачи, определяемый пользователем. Длительность сегмента равна $tq \times (TSEG2 + 1)$ и может быть уменьшена за счет ресинхронизации. Допустимые значения для TSEG1: от 01h до 07h |
| TSEG1 | 11-8 | Параметр 1. Временной промежуток от сегмента синхронизации до точки выборки, определяемый пользователем и включающий в себя сегмент распространения. Длительность равна $tq \times (TSEG1 + 1)$ и может быть увеличена за счет ресинхронизации. Допустимые значения для TSEG1: от 02h до 0Fh |
| SJW | 7-6 | Ширина перехода ресинхронизации. Длительность равна $tq \times (SJW + 1)$ |
| BRP | 5-0 | Предделитель скорости передачи. Если DIV8 = 0b, тогда длительность одного кванта времени равна (BRP + 1) тактам частоты. Если DIV8 = 1b, тогда длительность одного кванта времени равна $8 \times (BRP + 1)$ тактам частоты |
| – | 31-16 | Зарезервировано |

Таблица А.3.17 – Регистр счетчика ошибок узла

| Поле | Биты | Описание |
|--|-------|---|
| <p>NECNT Сброс: 00600000h</p>  | | |
| LEINC | 25 | Индикатор инкрементирования при последней ошибке 0 Обнаруженная ошибка приводит к инкрементированию счетчика ошибок на единицу 1 Обнаруженная ошибка приводит к инкрементированию счетчика ошибок на восемь |
| LETD | 24 | Флаг последней ошибки передачи 0 При приеме сообщения обнаружена ошибка, и произошло инкрементирование поля REC 1 При передаче сообщения обнаружена ошибка, и произошло инкрементирование поля TEC |
| EWRNLVL | 23-16 | Поле задания лимита ошибок, по достижении которого выставляется флаг EWRN в регистре NSR (по умолчанию, количество ошибок – 96) |
| TEC | 15-8 | Поле счетчика ошибок передачи сообщений |

Окончание таблицы А.3.17

| Поле | Биты | Описание |
|------|-------|---------------------------------------|
| REC | 7-0 | Поле счетчика ошибок приема сообщений |
| – | 31-26 | Зарезервировано |

Таблица А.3.18 – Регистр счетчика сообщений узла

| Поле | Биты | Описание |
|-------------|-----------|--|
| NFCR | | Сброс: 00000000h |
| | | |
| CFCOV | 23 | Флаг переполнения счетчика сообщений 0 Счетчик не переполнен 1 Счетчик переполнился. В режиме синхросчетчика этот флаг устанавливается при изменении поля CFC и, если установлен бит CFCIE, формируется прерывание Бит сбрасывается программно |
| CFCIE | 22 | Бит разрешения прерывания от счетчика сообщений 0 Запрещено 1 Разрешено |
| CFMOD | 20-19 | Поле задания режима работы счетчика сообщений 00 Счетчик сообщений. Инкрементируется после каждого успешного приема/передачи сообщения 01-11 Зарезервировано. Не использовать! |
| CFSEL | 18-16 | Поле задания параметров выбранного режима счетчика сообщений (см. таблицу А.3.19) |
| CFC | 15-0 | Поле счетчика сообщений Хранит значение счетчика сообщений при CFMOD = 00b |
| – | 31-24, 21 | Зарезервировано |

Таблица А.3.19 – Коды задания параметров режима счетчика сообщений

| CFSEL | Действия |
|--------|--|
| * * 1b | Счетчик инкрементируется каждый раз при получении сообщения, не имеющего объекта сообщения |
| * 1 *b | Счетчик инкрементируется каждый раз при получении сообщения, имеющего соответствующий объект сообщения |
| 1 * *b | Счетчик инкрементируется каждый раз при успешной отправке сообщения |
| 0 0 0b | Зарезервировано. Не использовать! |

Примечание – «*» указывает на то, что состояние этого бита поля CFSEL не важно для включения параметра режима. Все три параметра могут комбинироваться между собой (например, 110b или 101b).

Таблица А.3.20 – Регистр управления функционированием объекта сообщения

| MOFCR | | | | | | | | | | | | | | | |
|-------|-----------|--|--|----------|----------|-----|----------|-----|-----|-----|-----------|-----|----------|----------|----------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | DLC | | | | STT | SDT | RMM | FRR EN | - | OV IE | TX IE | RX IE |
| | | | | 3 4 | | | | 3 4 | 3 4 | 3 4 | 3 4 | - | 3 4 | 3 4 | 3 4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | DAT C | DLC C | IDC | GDF S | - | | | | MMC | | | |
| | | | | 3 4 | 3 4 | 3 4 | 3 4 | | | | | 3 4 | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| DLC | 27- 24 | Код длины данных. Показывает количество байт данных, находящихся в объекте сообщения. Диапазон – значение от 0 до 8. Если значение DLC больше 8, это автоматически указывает на 8 байт. Значение DLC полученного сообщения сохраняется таким, каким было получено | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| STT | 23 | Бит задания однократной пересылки данных | | | | | | | | | | | | | |
| | | 0 | Нет действий | | | | | | | | | | | | |
| | | 1 | Если бит установлен, тогда бит TXRQ сбрасывается после начала передачи объекта сообщения n. В связи с этим, в случае неудачной передачи, повторной передачи сообщения не будет | | | | | | | | | | | | |
| SDT | 22 | Бит задания однократного участия объекта сообщения n в пересылке | | | | | | | | | | | | | |
| | | 0 | Нет действий | | | | | | | | | | | | |
| | | 1 | Если бит установлен, и объект сообщения n не является объектом FIFO, тогда бит MSGVAL сбрасывается после успешного приема данных | | | | | | | | | | | | |
| RMM | 21 | Бит включения удаленного мониторинга объекта передачи | | | | | | | | | | | | | |
| | | 0 | Выключен. Идентификатор, бит IDE и поле DLC объекта сообщения n остаются без изменений до получения корректного фрейма удаленного запроса | | | | | | | | | | | | |
| | | 1 | Включен. Идентификатор, бит IDE и поле DLC корректного фрейма удаленного запроса копируются в объект передачи n в порядке получения битов фрейма удаленного запроса монитора | | | | | | | | | | | | |
| | | Состояние бита оказывает влияние только на объекты передач | | | | | | | | | | | | | |
| FRREN | 20 | Бит разрешения удаленного запроса. Определяет, будет ли устанавливаться бит TXRQ в объекте сообщения n или в другом объекте сообщения, на который указывает CUR | | | | | | | | | | | | | |
| | | 0 | Бит TXRQ объекта сообщения n устанавливается после получения корректного фрейма удаленного запроса | | | | | | | | | | | | |
| | | 1 | Бит TXRQ другого объекта сообщения (на который указывает CUR) устанавливается после получения им корректного фрейма удаленного запроса | | | | | | | | | | | | |

Продолжение таблицы А.3.20

| Поле | Бит | Описание | |
|------|-----|--|---|
| OVIE | 18 | Бит разрешения прерывания по заполнению FIFO объекта сообщения n. Прерывание генерируется, когда указатель CUR (указатель на текущий объект) достигает значения SEL регистра MOFGPRn | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| | | Если объект сообщения n является объектом приема FIFO, то поле TXINP (регистр MOIPRn) указывает на одну из 16 линий прерываний. Если объект сообщения n является объектом передачи FIFO, то поле RXINP (регистр MOIPRn) указывает на одну из 16 линий прерываний. Для всех других режимов объекта сообщения состояние бита OVIE не важно | |
| TXIE | 17 | Бит разрешения прерывания по окончании передачи сообщения | |
| | | 0 | Запрещено |
| | | 1 | Разрешено. Прерывание генерируется, если сообщение из объекта сообщения n было успешно передано. Поле TXINP (регистр MOIPRn) указывает на одну из 16 линий прерываний |
| RXIE | 16 | Бит разрешения прерывания по окончании приема сообщения | |
| | | 0 | Запрещено |
| | | 1 | Разрешено. Прерывание генерируется, если сообщение было успешно принято объектом сообщения n (напрямую или через шлюз). Поле RXINP (регистр MOIPRn) указывает на одну из 16 линий прерываний |
| DATC | 11 | Индикатор копирования данных | |
| | | 0 | Данные не копируются |
| | | 1 | Данные в регистрах MODATANn и MODATALn объекта-источника шлюза (после сохранения принятого фрейма в источнике) копируются через шлюз в объект-приемник |
| | | Бит DATC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует | |
| DLCC | 10 | Индикатор копирования кода длины данных DLC | |
| | | 0 | Код не копируется |
| | | 1 | Код длины данных объекта-источника шлюза (после сохранения принятого фрейма в источнике) копируется через шлюз в объект-приемник |
| | | Бит DLCC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует | |
| IDC | 9 | Индикатор копирования идентификатора | |
| | | 0 | Идентификатор не копируется |
| | | 1 | Идентификатор объекта-источника шлюза (после сохранения принятого фрейма в источнике) копируется через шлюз в объект-приемник |
| | | Бит IDC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует | |

Окончание таблицы А.3.20

| Поле | Бит | Описание |
|--|-----------------------|---|
| GDFS | 8 | Индикатор отправки фрейма через шлюз |
| | | 0 Состояние бита TXRQ объекта-приемника без изменений |
| | | 1 Установлен бит TXRQ объекта-приемника после внутренней передачи из объекта-источника |
| | | Бит GDFS используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует |
| MMC | 3-0 | Задание режима объекта сообщения n |
| | | 0000 Стандартный объект сообщения |
| | | 0001 Базовый объект приемной структуры FIFO |
| | | 0010 Базовый объект передающей структуры FIFO |
| | | 0011 Вспомогательный объект передающей структуры FIFO |
| | | 0100 Объект-источник шлюза |
| | | Остальные комбинации зарезервированы |
| – | 31–28, 19, 15–12, 7–4 | Зарезервировано |
| <p>Примечание – Под корректным фреймом удаленного запроса подразумевается фрейм, идентификатор которого совпадает с идентификатором объекта сообщения.</p> | | |

Таблица А.3.21 – Регистр указателя FIFO/шлюза объекта сообщения

| Поле | Биты | Описание |
|----------------------|-------|--|
| <p>MOFGPR</p> | | |
| SEL | 31-24 | Указатель объекта сообщения. Второй (программный) указатель в дополнение к аппаратному указателю CUR при работе с FIFO. Поле SEL используется для общего мониторинга (генерирование прерываний FIFO) |
| CUR | 23-16 | Указатель на текущий объект в пределах FIFO или шлюза. После каждой операции FIFO или передачи через шлюз указатель CUR обновляется – в него заносится номер следующего объекта сообщения в списке (поле PNEXT регистра MOSTATn) – до тех пор, пока не будет достигнут верхний элемент FIFO (поле TOP), после чего CUR сбрасывается, и в него загружается номер нижнего элемента списка (из поля BOT) |
| TOP | 15-8 | Указатель верхнего элемента FIFO. В поле находится номер последнего элемента |
| BOT | 7-0 | Указатель нижнего элемента FIFO. В поле находится номер первого элемента |

Таблица А.3.22 – Регистр указателя прерываний объекта сообщения

| Поле | Биты | Описание |
|--------|-------|--|
| CFCVAL | 31-16 | Количество фреймов. Каждый раз после записи принятого сообщение в объект сообщения n или успешной передачи объекта сообщения n, значение счетчика фреймов CFC (регистр NFCRn) копируется в CFCVAL |
| MPN | 15-8 | Номер ждущего бита сообщения. Указывает позицию бита, соответствующего объекту сообщения n в регистре MSPNDx |
| TXINP | 7-4 | Указатель линии прерываний для прерывания после передачи. Всего доступно 16 линий прерываний с номерами от 0 до 15. Значение 0000b, записанное в TXINP, выбирает нулевую линию прерываний, 0001b – первую, 0010b – вторую и т. д. Дополнительно бит TXINP используется для выбора позиции ждущего бита объекта сообщения n |
| RXINP | 3-0 | Указатель линии прерываний для прерывания после приема. Всего доступно 16 линий прерываний с номерами от 0 до 15. Значение 0000b, записанное в TXINP, выбирает нулевую линию прерываний, 0001b – первую, 0010b – вторую и т. д. Дополнительно бит RXINP используется для выбора позиции ждущего бита объекта сообщения n |

Таблица А.3.23 – Регистр маски объекта сообщения

| Поле | Биты | Описание |
|------|-------|-------------------------|
| MIDE | 29-28 | Маска прерываний (MIDE) |
| AM | 23-16 | Маска прерываний (AM) |
| AM | 15-8 | Маска прерываний (AM) |
| AM | 7-0 | Маска прерываний (AM) |

Окончание таблицы А.3.23

| Поле | Биты | Описание |
|------|--------|--|
| MIDE | 29 | Маска бита IDE сообщения |
| | | 0 Объект сообщения n может принимать как стандартные, так и расширенные фреймы |
| | | 1 Объект сообщения n может принимать только те фреймы, у которых состояние бита IDE совпадает с его битом IDE |
| AM | 28-0 | Маска идентификатора. При приеме расширенного сообщения используется вся маска. При приеме стандартного сообщения используются биты 28–18, при этом состояние битов 17–0 не важно |
| – | 31, 30 | Зарезервировано |

Таблица А.3.24 – Регистры данных объекта сообщения

| MODATAH | | |
|----------------|-------|-----------------------|
| | | |
| MODATAL | | |
| | | |
| Поле | Биты | Описание |
| DB7 | 31-24 | Седьмой байт данных |
| DB6 | 23-16 | Шестой байт данных |
| DB5 | 15-8 | Пятый байт данных |
| DB4 | 7-0 | Четвертый байт данных |
| DB3 | 31-24 | Третий байт данных |
| DB2 | 23-16 | Второй байт данных |
| DB1 | 15-8 | Первый байт данных |
| DB0 | 7-0 | Нулевой байт данных |

Таблица А.3.25 – Регистр арбитража объекта сообщения

| MOAR | | | |
|------|-------|---|--|
| | | | |
| Поле | Биты | Описание | |
| PRI | 31-30 | Класс приоритета. Поле определяет один из четырех классов (0, 1, 2 и 3) приоритета объекта сообщения n. Нулевой класс устанавливает наивысший приоритет. Объекты сообщений с нулевым классом всегда выигрывают арбитраж при передаче и приеме сообщений. Фильтрация сообщений на основе идентификатора (маскируемого) и позиции в списке организуется только для объектов сообщений с равным приоритетом. Кроме этого, поле PRI определяет метод фильтрации | |
| | | 00 | Зарезервировано |
| | | 01 | Фильтрация в зависимости от положения объекта сообщения в списке. Объект сообщения n получает приоритет на передачу сообщения только в случае, если нет других объектов сообщений с установленными битами MSGVAL, TXEN0 и TXEN1, стоящих выше по списку |
| | | 10 | Фильтрация в зависимости от значения идентификатора. Объект сообщения n получает приоритет на передачу сообщения только в случае, если в списке нет других объектов сообщений с «Идентификатор + IDE + DIR» более высокого приоритета (согласно правилам арбитража в таблице А.3.26) |
| | | 11 | Фильтрация в зависимости от положения объекта сообщения в списке (как при PRI = 01b) |
| IDE | 29 | Бит расширения идентификатора объекта сообщения n | |
| | | 0 | Объект сообщения n оперирует с фреймами со стандартным 11-битным идентификатором |
| | | 1 | Объект сообщения n оперирует с фреймами с расширенным 29-битным идентификатором |
| ID | 28-0 | Идентификатор объекта сообщения n. При оперировании с расширенными фреймами используются биты 28–0. При оперировании со стандартными фреймами используются биты 28–18, при этом состояние битов 17–0 не важно | |

Таблица А.3.26 – Распределение приоритета между объектами сообщений согласно правилам арбитража

| Установки для объектов сообщений 0 и 1, которые участвуют в арбитраже (приоритет объекта 0 выше приоритета объекта 1) | Пояснение |
|--|--|
| MOAR0[28:18] < MOAR1[28:18] (11-битный стандартный идентификатор объекта 0 меньше по числовому значению, чем 11-битный идентификатор объекта 1) | Стандартный фрейм с идентификатором, имеющим меньшее значение, обладает более высоким приоритетом |
| MOAR0[28:18] = MOAR1[28:18]. В регистре MOAR0 бит IDE = 0. В регистре MOAR1 бит IDE = 1. | При равенстве значений стандартных идентификаторов, стандартный фрейм имеет приоритет перед расширенным |
| MOAR0[28:18] = MOAR1[28:18]. Биты IDE обоих объектов сброшены. В регистре MOSTAT0 бит DIR = 1. В регистре MOSTAT1 бит DIR = 0. | При равенстве значений идентификаторов стандартный фрейм данных имеет приоритет перед стандартным фреймом удаленного запроса |
| MOAR0[28:0] = MOAR1[28:0] Биты IDE обоих объектов установлены. В регистре MOSTAT0 бит DIR = 1. В регистре MOSTAT1 бит DIR = 0. | При равенстве значений идентификаторов расширенный фрейм данных имеет приоритет перед расширенным фреймом удаленного запроса |
| MOAR0[28:0] < MOAR1[28:0] Биты IDE обоих объектов установлены. (29-битный идентификатор объекта 0 меньше по числовому значению, чем 29-битный идентификатор объекта 1) | Расширенный фрейм с идентификатором, имеющим меньшее значение, обладает более высоким приоритетом |

Таблица А.3.27 – Регистр управления объектом сообщения

| МОСТР | | | | | | | | | | | | | | | |
|--|----|------------|----|------------------------|----------|----------|----------|----------|------------|-------------|-------------|-------------|------------|------------|-----------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | SET DIR | SET TXEN | SET TXEN | SET TXRQ | SET RXEN | SET RT SEL | SET MSG VAL | SET MSG LST | SET NEW DAT | SET RXUP D | SET TXPN D | RES RXPND |
| | | | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | RES DIR | RES TXEN | RES TXEN | RES TXRQ | RES RXEN | RES RT SEL | RES MSG VAL | RES MSG LST | RES NEW DAT | RES RXUP D | RES TXPN D | RES RXPND |
| | | | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| <p>Биты регистра работают попарно. Комбинация состояний бит каждой пары оказывает влияние на один (соответствующий этой паре) бит регистра MOSTATn того же объекта сообщения. Так, пара SETDIR-RESDIR устанавливает и сбрасывает бит DIR, пара SETTXN1-RESTXN1 устанавливает и сбрасывает бит TXEN1 и т. д.</p> <p>После записи старшего или младшего слова регистра МОСТРn аппаратная часть проверяет состояние бит каждой пары и, в зависимости от обнаруженной комбинации, выполняет соответствующее действие</p> | | | | | | | | | | | | | | | |
| Бит SET*** | | Бит RES*** | | Действие над битом *** | | | | | | | | | | | |
| 0 | | 0 | | Нет | | | | | | | | | | | |
| 1 | | 1 | | Нет | | | | | | | | | | | |
| 1 | | 0 | | Установка | | | | | | | | | | | |
| 0 | | 1 | | Сброс | | | | | | | | | | | |
| Биты 31-28 и 15-12 являются зарезервированными | | | | | | | | | | | | | | | |

Таблица А.3.28 – Регистр состояния объекта сообщения

| MOSTAT | | |
|--------|-------|---|
| | | |
| Поле | Бит | Описание |
| PNEXT | 31-24 | Указатель на следующий элемент списка. В поле находится номер объекта сообщения, расположенного выше по списку относительно текущего |
| PPREV | 23-16 | Указатель на предыдущий элемент списка. В поле находится номер объекта сообщения, расположенного ниже по списку относительно текущего |
| LIST | 15-12 | Номер списка, которому принадлежит объект сообщения n. Поле обновляется аппаратно при распределении/перераспределении объекта сообщения |
| DIR | 11 | Бит распределения |
| | | <p>0 Объект приема сообщения данных. Объект принимает сообщение данных. При установленном бите TXRQ объект формирует сообщение удаленного запроса с идентификатором объекта n, а затем передает его. Полученное в ответ сообщение данных с соответствующим идентификатором сохраняется в объекте сообщения n</p> <p>1 Объект передачи сообщения данных. При установленном бите TXRQ объект формирует, а затем передает сообщение данных. Если объект n получает сообщение удаленного запроса с соответствующим идентификатором, то устанавливается флаг TXRQ его регистра MOSTATn, после чего в ответ передается сообщение данных, содержащихся в объекте n</p> |
| TXEN1 | 10 | Бит разрешения передачи фрейма |
| | | <p>0 Запрещено</p> <p>1 Передача фрейма разрешена. Объект сообщения n может участвовать в передаче только, если установлены оба бита – TXEN1 и TXEN0. Контроллер CAN использует бит TXEN1 для выбора активного объекта передачи сообщения из FIFO</p> |

Продолжение таблицы А.3.28

| Поле | Бит | Описание |
|--------|-----|---|
| TXEN0 | 9 | Бит разрешения передачи фрейма |
| | | 0 Запрещено |
| | | 1 Передача фрейма разрешена. Объект сообщения n может участвовать в передаче, только если установлены оба бита – TXEN0 и TXEN1. Контроллер CAN использует бит TXEN1 для выбора активного объекта передачи сообщения из FIFO. Можно программно очищать бит TXEN0 для запрета передачи сообщения, которое в настоящий момент формируется, или для запрета автоматической передачи в ответ на удаленный запрос |
| TXRQ | 8 | Бит инициации передачи |
| | | 0 Нет действий |
| | | 1 Установка бита иницирует передачу фрейма из объекта сообщения n. Инициация передачи фрейма возможна только в случае, если установлены биты TXRQ, TXEN0, TXEN1 и MSGVAL. Также бит TXRQ устанавливается аппаратно при получении фрейма удаленного запроса. Бит сбрасывается аппаратно при успешном завершении передачи и если при этом не был повторно программно установлен бит NEWDAT |
| RXEN | 7 | Бит разрешения приема |
| | | 0 Запрещено |
| | | 1 Объект сообщения может принимать сообщения Состояние бита учитывается только при фильтрации принимаемых сообщений |
| RTSEL | 6 | Индикатор возможности приема/передачи |
| | | 0 Объект сообщения не может принимать/передавать сообщения |
| | | 1 Объект сообщения может принимать/передавать сообщения Прием фрейма. Бит RTSEL устанавливается аппаратно после того, как выбран объект сообщения n для сохранения только что принятого фрейма. Прежде, чем записать принятые данные в объект сообщения n, аппаратная часть проверяет состояние бита RTSEL. ЦПУ может сбрасывать этот бит, чтобы запретить запись принятого фрейма в объект сообщения n. Передача фрейма. Бит RTSEL устанавливается аппаратно после того, как выбран следующий объект сообщения n для передачи фрейма. Аппаратная часть перед началом передачи проверяет: установлен ли бит RTSEL и сброшен ли бит NEWDAT. Бит RTSEL должен оставаться установленным до окончания передачи. Проверка состояния бита RTSEL производится только при попытке изменения содержимого объекта сообщения n во избежание одновременного выполнения операций передачи фрейма и его изменения. Бит не участвует в фильтрации сообщений, и не сбрасывается аппаратно |
| MSGVAL | 5 | Бит активности объекта сообщения n |
| | | 0 Не активен |
| | | 1 Активен Только те объекты сообщений, для которых установлен этот бит, могут использоваться для операций приема и передачи |

Окончание таблицы А.3.28

| Поле | Бит | Описание | |
|--------|-----|------------------------------|---|
| MSGLST | 4 | Бит потери сообщения | |
| | | 0 | Ни одно сообщение не потеряно |
| | | 1 | Принятое сообщение потеряно вследствие того, что контроллер CAN попытался установить бит NEWDAT по окончании приема сообщения при том, что флаг NEWDAT уже был установлен ранее после записи другого сообщения |
| NEWDAT | 3 | Индикатор новых данных | |
| | | 0 | С момента сброса бита NEWDAT никаких изменений объекта сообщения не обнаружено |
| | | 1 | Объект сообщения был изменен. Бит устанавливается аппаратно после того, как принятое сообщение было сохранено в объекте сообщения n. Бит сбрасывается аппаратно после начала передачи объекта сообщения n. Бит NEWDAT следует устанавливать программно после того, как новые данные для передачи будут сохранены в объекте сообщения n для предотвращения автоматического сброса бита TRXQ в конце текущей передачи |
| RXUPD | 2 | Индикатор изменений | |
| | | 0 | Нет текущих изменений |
| | | 1 | Идентификатор сообщения, поле длины данных DLC и данные в объекте сообщения изменяются |
| TXPND | 1 | Индикатор окончания передачи | |
| | | 0 | Переданных сообщений нет |
| | | 1 | Сообщение объекта n было успешно передано |
| RXPND | 0 | Индикатор окончания приема | |
| | | 0 | Принятых сообщений нет |
| | | 1 | Сообщение было успешно принято объектом сообщения n (напрямую или через шлюз). Бит должен сбрасываться программно |

А.4 Регистры контроллера Ethernet 10/100

Компоненты дескриптора

Таблица А.4.1 – Стартовый адрес для пакета данных

| PacketStartAddr | | |
|----------------------|------|-------------------------------|
| | | |
| Поле | Биты | Описание |
| PACKET_START_ADDRESS | 31-2 | Стартовый адрес пакета данных |
| – | 1-0 | Зарезервировано |

Таблица А.4.2 – Размер пакета данных

| PacketSize | | |
|------------|------|--|
| | | |
| Поле | Биты | Описание |
| EMPTYFLAG | 31 | <p>Для операций передачи этот бит указывает на доступность данных передачи, связанных с пакетом. Для операций приема, этот бит показывает наличие места для сохранения принимаемого пакета. Установка этого флага используется для проверки правильности дескриптора.</p> <p>При завершении операции передачи, контроллер DMA устанавливает этот бит, что говорит, что эти данные использованы для передачи. При успешном завершении операции приема, контроллер DMA сбрасывает этот бит, что указывает на то, что отведенное место использовано для сохранения пакета. Первое действие гарантирует, что данные не будут переданы дважды, второе – что сохраненные данные не будут заперчены записью следующего пакета</p> |

Окончание таблицы А.4.2

| Поле | Бит | Описание | | |
|--------------------|-----------------|---|------------------------------------|--------------------------------------|
| FTPP_ OVERRIDES | 20-16 | Флаги управления FIFO во время обмена пакетами | | |
| | | Номер бита | Название | Описание |
| | | 20 | FTCFRM | FIFO передает Control Frame |
| | | 19-18 | FTPPPADMODE | FIFO передает пакет в режиме PADMode |
| | | 17 | FTPPGENFCS | FIFO передает пакет с генерацией FCS |
| | 16 | FTPPEN | Разрешение FIFO на передачу пакета | |
| PACKET_ SIZE | 11-0 | Поле для операций передачи. Содержит размер пакета передачи в байтах. При приеме это поле заполняет контроллер DMA. Значение этого поля до приема будет контроллером проигнорировано | | |
| – | 30-21, 15-12 | Зарезервировано | | |

Таблица А.4.3 – Адрес следующего дескриптора

| Поле | Биты | Описание |
|------------------------|------|------------------------------|
| – | 1-0 | Зарезервировано |
| DESCRIPTOR_ ADDRESS | 31-2 | Адрес следующего дескриптора |

NextDescriptor

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

DESCRIPTOR_ADDRESS

3 ч

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

DESCRIPTOR_ADDRESS

3 ч

Регистры контроллера

Таблица А.4.4 – Регистр 1 конфигурации MAC

| MAC1 | | 8008_0000h | | | | Сброс: 00008000h | | | | | | | | | |
|---|------------------|------------|----|-------------------|-------------------|-------------------|-------------------|---|---|---|--------------|-----------------|-----------------|-------------|------------------|
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOFT RE SET | SIM RE SET | - | - | RE SET RMCS | RE SET RFUN | RE SET TMCS | RE SET TFUN | - | - | - | LOOP BACK | TX PAU SE | RX PAU SE | PASS ALL | RX ENA BLE |
| 3 ч | 3 ч | | | 3 ч | 3 ч | 3 ч | 3 ч | | | | 3 ч | 3 ч | 3 ч | 3 ч | 3 ч |

Окончание таблицы А.4.4

| Поле | Бит | Описание |
|-----------|-------------------------|--|
| SOFTRESET | 15 | Бит сброса блока MAC контроллера Ethernet 10/100 кроме блока связи с процессором |
| | | 0 Нет действий |
| | | 1 Установка бита сбрасывает блок MAC |
| SIMRESET | 14 | Бит сброса генератора случайных чисел устройства передачи |
| | | 0 Нет действий |
| | | 1 Установка бита сбрасывает генератор |
| RESETRMCS | 11 | Бит сброса устройств MAC-уровня, отвечающих за управление фильтрацией адресов при приеме пакетов (контроль домена) |
| | | 0 Нет действий |
| | | 1 Установка бита сбрасывает устройства MAC-уровня |
| RESETRFUN | 10 | Бит сброса логики устройств приема пакетов |
| | | 0 Нет действий |
| | | 1 Установка бита сбрасывает логику устройств приема пакетов |
| RESETMCS | 9 | Бит сброса устройств MAC-уровня, отвечающих за управление адресами при передаче информации (контроль домена) |
| | | 0 Нет действий |
| | | 1 Установка бита сбрасывает устройства MAC-уровня |
| RESETTFUN | 8 | Бит сброса логики устройств передачи пакетов |
| | | 0 Нет действий |
| | | 1 Установка бита сбрасывает логику устройств передачи пакетов |
| LOOPBACK | 4 | Бит активации приема передаваемых пакетов обратно через MACReceive-интерфейс. Очистка бита не вызывает прием передаваемых пакетов обратно |
| | | 0 Нет действий |
| | | 1 Установка бита вызывает прием передаваемых пакетов |
| TXPAUSE | 3 | Бит разрешения передачи пауз в Frame. |
| | | 0 Передача пауз в Frame заблокирована |
| | | 1 Разрешено |
| RXPAUSE | 2 | Бит разрешения приема паузы как части Frame |
| | | 0 Пауза игнорируется |
| | | 1 Разрешено |
| PASSALL | 1 | Бит управления PASS |
| | | 0 PASS подтверждается для текущего принимаемого Frame при успешной передаче Frame |
| | | 1 Блок MAC выдает PASS для текущего принимаемого Frame независимо от его типа (т. е. для всех Frame) |
| RXENABLE | 0 | Бит разрешения приема Frame |
| | | 0 Frame не принимаются |
| | | 1 Frame принимаются. Внутренняя MAC-синхронизация использует этот бит для приема потока и выхода SYNCHRONIZED RECEIVE ENABLE, используемого MAC для уточнения принимаемого фрейма |
| – | 31-16, 12- 13,7-5 | Зарезервировано |

Таблица А.4.5 – Регистр 2 конфигурации MAC

| MAC2 | | 8008_0004h | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------|--|---|---|--------|------|------|------|------|------|------|------|------|------|------|-----|------|------|------|--|-----|------|------|--|--|-----|-----|-----|-----|-----|-----|----|-----|------|-----|--|------|------|-----|--|--|--|--|--|--|-----|-----|-----|----|-----|-----|--|-----|-----|-----|--|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="text-align: center; margin: 5px 0;">-</div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center; font-size: small;"> <tr> <td style="width: 2.5%;">-</td> <td style="width: 2.5%;">EXCESS</td> <td style="width: 2.5%;">BP/N</td> <td style="width: 2.5%;">NO</td> <td style="width: 2.5%;">-</td> <td style="width: 2.5%;">-</td> <td style="width: 2.5%;">LONG</td> <td style="width: 2.5%;">PURE</td> <td style="width: 2.5%;">AUTO</td> <td style="width: 2.5%;">VLAN</td> <td style="width: 2.5%;">PAD</td> <td style="width: 2.5%;">CRC</td> <td style="width: 2.5%;">DAL</td> <td style="width: 2.5%;">HUGE</td> <td style="width: 2.5%;">LENG</td> <td style="width: 2.5%;">FULL</td> </tr> <tr> <td></td> <td>DEF</td> <td>OBAC</td> <td>BACK</td> <td></td> <td></td> <td>PRE</td> <td>PRE</td> <td>PAD</td> <td>PAD</td> <td>ENA</td> <td>ENA</td> <td>AY</td> <td>FRA</td> <td>THCH</td> <td>DUP</td> </tr> <tr> <td></td> <td>KOFF</td> <td>KOFF</td> <td>OFF</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>BLE</td> <td>BLE</td> <td>CRC</td> <td>ME</td> <td>ECK</td> <td>LEX</td> </tr> <tr> <td></td> <td>3 4</td> <td>3 4</td> <td>3 4</td> <td></td> <td></td> <td>3 4</td> <td>3 4</td> <td>3 4</td> <td>3 4</td> <td>3 4</td> <td>3 4</td> <td>3 4</td> <td>3 4</td> <td>3 4</td> <td>3 4</td> </tr> </table> | | | | - | EXCESS | BP/N | NO | - | - | LONG | PURE | AUTO | VLAN | PAD | CRC | DAL | HUGE | LENG | FULL | | DEF | OBAC | BACK | | | PRE | PRE | PAD | PAD | ENA | ENA | AY | FRA | THCH | DUP | | KOFF | KOFF | OFF | | | | | | | BLE | BLE | CRC | ME | ECK | LEX | | 3 4 | 3 4 | 3 4 | | | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 |
| - | EXCESS | BP/N | NO | - | - | LONG | PURE | AUTO | VLAN | PAD | CRC | DAL | HUGE | LENG | FULL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | DEF | OBAC | BACK | | | PRE | PRE | PAD | PAD | ENA | ENA | AY | FRA | THCH | DUP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | KOFF | KOFF | OFF | | | | | | | BLE | BLE | CRC | ME | ECK | LEX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 3 4 | 3 4 | 3 4 | | | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EXCESSDEF | 14 | Бит выбора режима обработки пакета | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Обработка пакета прерывается в случае избыточной задержки, которая превышает установленный лимит | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Блок MAC будет неопределенно долго осуществлять обработку пакета (в соответствии со стандартом) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BP/OBACKOFF | 13 | Бит разрешения повторной передачи при коллизии | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | В случае коллизии повторная передача будет повторена немедленно, без периода ожидания, с целью повышения вероятности успешной передачи пакета | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NOBACKOFF | 12 | Бит задания параметра повторной передачи при коллизии | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | В случае коллизии период ожидания до повторной передачи определяется бинарно-экспоненциальным алгоритмом в соответствии со стандартом IEEE 802-3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LONGPRE | 9 | Бит выбора длины преамбулы принимаемых пакетов | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Блок MAC допускает прием пакетов с преамбулами любой длины, как в стандарте | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Блок MAC позволяет принимать только пакеты, преамбула в которых менее 12 байт | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PUREPRE | 8 | Бит включения проверки преамбулы | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Проверка преамбулы не проводится | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Блок MAC верифицирует содержимое преамбулы на содержание ошибок. При обнаружении ошибки в преамбуле обработка пакета прекращается | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AUTOPAD | 7 | Бит включения автоматического определения типа Frame. Бит игнорируется, если бит PADENABLE сброшен (см. таблицу А.4.6) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Блок MAC автоматически определяет тип Frame (теговый или не теговый), сравнивая 2 октета адреса последующего источника с 8100h (VLAN протокол ID) или PAD, соответственно | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.4.5

| Поле | Бит | Описание |
|-------------|-----------------|---|
| VLANPAD | 6 | Бит включения дополнения коротких Frame. Бит игнорируется, если бит PADENABLE сброшен (см. таблицу А.4.6) |
| | | 0 |
| | | 1 Блок MAC добавляет PAD ко всем коротким Frame, доводя их длину до 64 байт, и присоединяет значение CRC |
| PADENABLE | 5 | Бит разрешения функционирования битов AUTOPAD и VLANPAD |
| | | 0 Бит следует сбросить, если все Frame имеют правильную длину |
| | | 1 Блок MAC добавляет PAD во все короткие Frame. Этот бит используется совместно с битами AUTO PAD ENABLE или VLAN PAD ENABLE (см. таблицу XX) |
| CRCENABLE | 4 | Бит разрешения вставки CRC |
| | | 0 Бит следует сбросить, если CRC уже включен во Frame |
| | | 1 CRC автоматически добавляется к Frame (в PAD), независимо от того, требовалось это или нет |
| DALAYCRC | 3 | Бит разрешения добавления 4 байт CRC |
| | | 0 Бит следует сбросить, если добавление CRC отсутствует |
| | | 1 4 байта CRC поля служебной информации, которые расположены в начале Frame, добавляются при передаче и вырезаются при приеме |
| HUGEFRAME | 2 | Бит разрешения приема и передачи Frame произвольной длины |
| | | 0 Запрещено |
| | | 1 Разрешено |
| LENGTHCHECK | 1 | Бит разрешения компарации длины Frame |
| | | 0 |
| | | 1 Длина передаваемого и принимаемого Frame компарируется с полем Length/TypeFrame. При успешной компарации – проверка выполнена. О несовпадении выдается сообщение в Transmit/Receive Statistics Vector |
| FULLDUPLEX | 0 | Бит выбора режима выполнения MAC-операций |
| | | 0 Half-Duplex |
| | | 1 Full-Duplex |
| – | 31-15, 11-10 | Зарезервировано |

Таблица А.4.6 – PAD-операции

| Тип | Биты регистра MAC2 | | | Вид Frame |
|-----|--------------------|----------|------------|--|
| | AUTO PAD | VLAN PAD | PAD ENABLE | |
| Все | X | X | 0 | Без PAD, проверка CRC |
| Все | 0 | 0 | 1 | PAD 60 бит, присоединено CRC |
| Все | X | 1 | 1 | PAD 64 бит, присоединено CRC |
| Все | 1 | 0 | 1 | Если безтеговый: PAD 60 бит, присоединено CRC. Если VLAN – теги: PAD 64 бит, присоединено CRC |

Таблица А.4.7 – Регистр Back-to-Back Inter-Packet-Gap

| | | | |
|--------------|-------------|--|------------------|
| IPGT | | 8008_0008h | Сброс: 00000000h |
| | | | |
| Поле | Биты | Описание | |
| BtBIPG_TRANS | 6-0 | Поле задания минимального времени от окончания передачи последнего полубайта пакета до начала передачи следующего пакета. В режиме Full-Duplex значение этого поля следует задавать равным периоду передачи полубайта минус три. Рекомендуемое значение – 15h, которое обеспечивает минимальный IPG равный 0,96 мкс для 100 Мб/с и 9.6 мкс для 10 Мб/с. В Half-Duplex режиме значение поля следует задавать равным периоду передачи полубайта минус шесть. Рекомендуемое значение – 12h, которое обеспечивает минимальный IPG равный 0,96 мкс для 100 Мб/с и 9.6 мкс для 10 Мб/с | |
| – | 31-7 | Зарезервировано | |

Таблица А.4.8 – Регистр Non-Back-to-Back Inter-Packet-Gap

| | | | |
|---------------------|-------------|--|------------------|
| IPGR | | 8008_000Ch | Сброс: 00000000h |
| | | | |
| Поле | Биты | Описание | |
| NBtBIPG_TRANS_PART1 | 14-8 | Поле представляет собой опцию carrierSense (определение несущей частоты), описанную в IEEE 802.3/4.2.3.2.1 «Carrier Deference». Если несущая частота идентифицирована во время синхронизации IPGR1, то блок MAC подстраивается под несущую частоту. Но, если несущая частота определена после IPGR1, то блок MAC в течении IPGR2 продолжает передачу, принудительно создавая коллизию, и таким образом, обеспечивая явный доступ к передающей среде. Поле может содержать значения 0h до NBtBIPG_TRANS_PART2. Рекомендуемое значение – 0Ch | |

Окончание таблицы А.4.8

| Поле | Бит | Описание |
|-----------------------------|--------------|---|
| NBtBIPG_ TRANS_ PART2 | 6-0 | Поле представляет собой non-back-to-back Inter-Packet-Gap. Рекомендуемое значение –12h (обеспечивает минимальное IPG равное 0,96 мкс для 100 Мб/с и 9.6 мкс для 10 Мб/с). |
| – | 31- 15, 7 | Зарезервировано |

Таблица А.4.9 – Регистр окна коллизий/повторов

| Поле | Биты | Описание |
|----------------------|---------------|---|
| COLLISION_ WINDOW | 13-8 | Поле содержит время слота или окна коллизий, во время которого возможны коллизии в конфигурированных сетях. Окно коллизий начинается с начала преамбулы, включая SFD. Значение окна коллизий, по умолчанию 37h, что соответствует числу байт в Frame в конце окна |
| RETRANS_MAX | 3-0 | Поле задания числа попыток повторной передачи после коллизии, прежде чем передача пакета будет прервана из-за превышения числа коллизий. Рекомендуемое значение Fh |
| – | 31-14, 7-4 | Зарезервировано |

Таблица А.4.10 – Регистр верхней границы размера Frame

| Поле | Биты | Описание |
|-----------------|-------|--|
| MAXFRAME_LENGTH | 31-28 | Поле задает верхнюю границу размера Frame в байтах. Рекомендуемое значение 00000600h |
| – | 30-0 | Зарезервировано |

Окончание таблицы А.4.10

| Поле | Биты | Описание |
|-----------------|-------|--|
| MAXFRAME_LENGTH | 15-0 | Поле задания максимальной длины Frame. По умолчанию, поле содержит значение 0600h, что является максимально возможной длиной Frame равной 1536 октет. Безтеговый максимальный размер Frame равен 1518 октет. Теговый – на 4 октета больше и составляет 1522 октета. Примечание – Для VLAN тегового Frame к его размеру следует прибавлять 4 байта. |
| – | 31-16 | Зарезервировано |

Таблица А.4.11 – Регистр поддержки PHY-интерфейса

| SUPP | | 8008_0018h | | | | | | | | | | | | | | Сброс: 00001000h | |
|------------------|-----|---|-------------------|----|-----------|-------------------|--------------------|------------------|--------------|------------------|----|--------------------|-----------------|----|----|------------------|----|
| | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | - | | | | | | | | | | | | | | | |
| | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RE SET INT | - | PHY MO DE | RE SET RMII | - | SPE ED | RE SET 100X | FORC EQUI ET | NO CIPH ER | LINK FAIL | RE SET 10T | - | ENBL JAB BER | BIT MO DE | | | | |
| 3 4 | | 3 4 | 3 4 | | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 4 | | 3 4 | 3 4 | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | |
| RESETRMII | 15 | Бит сброса физического интерфейса. Если подключен один модуль физического интерфейса, то бит можно использовать вместо битов RESETRMII, RESET100X и RESET10T | | | | | | | | | | | | | | | |
| | 0 | Нормальные операции с модулем физического интерфейса | | | | | | | | | | | | | | | |
| | 1 | Установка сбрасывает выбранный модуль физического интерфейса (устройство связи с физической линией) | | | | | | | | | | | | | | | |
| PHYMODE | 12 | Бит конфигурации последовательного MI с подключенными SMI-устройствами (используется при подключении SMIPHY) | | | | | | | | | | | | | | | |
| | 0 | Активны функции SMIMAC. Если выбран SMIMAC, то операции приема/передачи пакетов выполняются в режиме Full Duplex на частоте 100 Мб/с | | | | | | | | | | | | | | | |
| | 1 | | | | | | | | | | | | | | | | |
| RESETRMII | 11 | Бит сброса логики упрощенного MI | | | | | | | | | | | | | | | |
| | 0 | Нет действий | | | | | | | | | | | | | | | |
| | 1 | Установка бита активирует сброс | | | | | | | | | | | | | | | |
| SPEED | 8 | Бит задания рабочей скорости упрощенного MI | | | | | | | | | | | | | | | |
| | 0 | 10 Мб/с | | | | | | | | | | | | | | | |
| | 1 | 100 Мб/с | | | | | | | | | | | | | | | |
| RESET100X | 7 | Бит сброса модуля, который содержит логику шифратора/дешифратора символов 4 бит/5 бит (4B/5B) | | | | | | | | | | | | | | | |
| | 0 | Нет действий | | | | | | | | | | | | | | | |
| | 1 | Установка бита активирует сброс | | | | | | | | | | | | | | | |

Окончание таблицы А.4.11

| Поле | Бит | Описание |
|------------|---------------------------------|--|
| FORCEQUIET | 6 | Бит включения шифрации |
| | | 0 Выполняется нормальная операция (без шифрации) |
| | | 1 На выход передаются зашифрованные (4В/5В) данные |
| NOIPHER | 5 | Бит выбора режима шифрования |
| | | 0 Передача с нормальным шифрованием |
| | | 1 Передача 5В символов без шифрования |
| LINKFAIL | 4 | Бит разрешения моделирования |
| | | 0 Выполняется нормальная операция |
| | | 1 330 мс LinkFail таймер отключается для разрешения короткого моделирования |
| RESET10T | 3 | Бит сброса модуля, который преобразует потоки полубайт МП в последовательный поток двоичных сигналов приемопередатчика режима 10Т |
| | | 0 Нет действий |
| | | 1 Установка бита активирует сброс |
| ENBLJABER | 1 | Бит включения защиты от неправильных данных при передаче в режиме 10Т ENDEC. Условием защиты является передача одного значения в линию более 50 мс и служит для устранения этого условия для передач другими станциями |
| | | 0 Выключено |
| | | 1 Включено |
| BITMODE | 0 | Блок активации режима 10BASE-T ENDEC |
| | | 0 |
| | | 1 Блок MAC работает в режиме 10BASE-T ENDEC, при котором синхронизация происходит по фактическим данным битов, а не на основе тактового генератора полубайт |
| – | 31-16, 14,13, 10, 9, 2 | Зарезервировано |

Таблица А.4.12 – Регистр управления конфигурацией МП

| MCFG | | 8008_0020h | Сброс: 00000000h |
|--|------|-------------------------------------|------------------|
| <div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; width: 100%; text-align: center; line-height: 30px;">-</div> | | | |
| <div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">RE SET MGMT</div> <div style="border: 1px solid black; flex-grow: 1; text-align: center; line-height: 20px;">-</div> <div style="border: 1px solid black; padding: 2px;">CLOCK_SELECT</div> <div style="border: 1px solid black; padding: 2px;">NO PRE</div> <div style="border: 1px solid black; padding: 2px;">SCAN INC</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> 3 4 3 4 3 4 3 4 </div> | | | |
| Поле | Биты | Описание | |
| RESETMGMT | 15 | Бит сброса модуля управления МП | |
| | | 0 Нет действий | |
| | | 1 Установка бита активирует сброс | |

Окончание таблицы А.4.12

| Поле | Бит | Описание | |
|--------------|-------------|--|---|
| CLOCK_SELECT | 4-2 | Поле выбора делителя частоты. Делитель используется при делении частоты для формирования управляющей частоты (MDC) модуля МП, значение которой определено в стандарте IEEE 802.3u как не превышающее 2,5 МГц Примечание – Некоторые РНУ поддерживают частоту до 12,5 МГц. | |
| | | 000, 001 | 1/4 |
| | | 010 | 1/6 |
| | | 011 | 1/8 |
| | | 100 | 1/10 |
| | | 101 | 1/14 |
| | | 110 | 1/20 |
| | | 111 | 1/28 |
| NOPRE | 1 | Бит управления преамбулой | |
| | | 0 | Поддерживаются нормальные циклы чтения/записи. Некоторые РНУ поддерживают работу без преамбулы |
| | | 1 | Модуль управления МП выполняет циклы чтения/записи без 32-битного поля преамбулы |
| SCANINC | 0 | | |
| | | 0 | Непрерывное чтение одного РНУ |
| | | 1 | Модуль МП выполняет чтение всех адресов РНУ, начиная с адреса 1 (поле возможных адресов, биты с 4 по 0) |
| – | 31-16, 14-5 | Зарезервировано | |

Таблица А.4.13 – Регистр команд МП

| Поле | Бит | Описание | |
|---|------|-----------------|---|
| <p>MCMD 8008_0024h Сброс: 00000000h</p> <div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; width: 100%; position: relative;"> <div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); font-size: 2em;">-</div> </div> <div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; width: 100%; position: relative;"> <div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); font-size: 2em;">-</div> <div style="position: absolute; right: 0; bottom: 0; text-align: right; padding-right: 5px;"> SCANREAD </div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 ч3 ч </div> | | | |
| SCAN | 1 | | |
| | | 0 | |
| | | 1 | Модуль управления МП выполняет циклы чтения непрерывно. Это полезно, к примеру, для мониторинга сбоя LinkFail |
| READ | 0 | | |
| | | 0 | |
| | | 1 | Модуль управления МП выполняет одиночные циклы чтения. Прочитанные данные возвращаются в регистр MRDD |
| – | 31-2 | Зарезервировано | |

Таблица А.4.14 – Регистр адреса МП

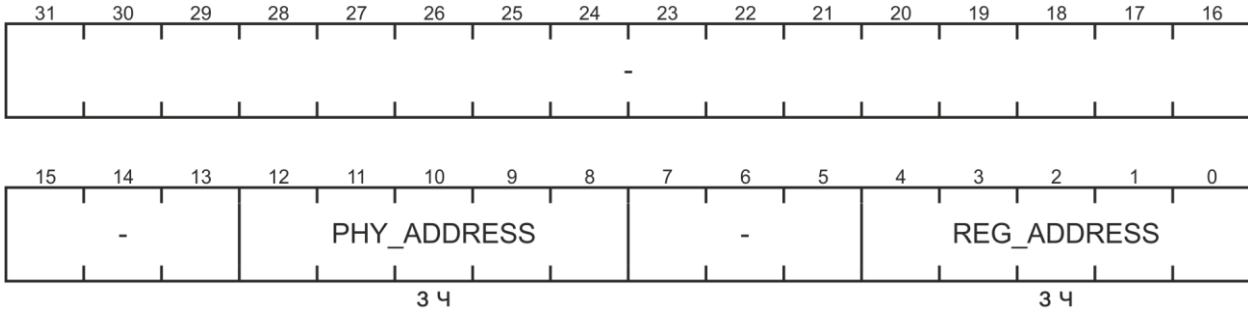
| | | |
|--|---------------|--|
| MADR | 8008_0028h | Сброс: 00000000h |
|  | | |
| Поле | Биты | Описание |
| PHY_ADDRESS | 12-8 | Поле содержит 5-битный адрес PHY-устройства для циклов управления от МП. Можно адресовать до 31 PHY. Значение 00h является зарезервированным |
| REG_ADDRESS | 4-0 | Поле содержит 5-битный адрес регистра PHY-устройства для циклов управления от МП. Можно адресовать до 32 регистров |
| – | 31-13, 7-5 | Зарезервировано |

Таблица А.4.15 – Регистр записываемых данных в МП

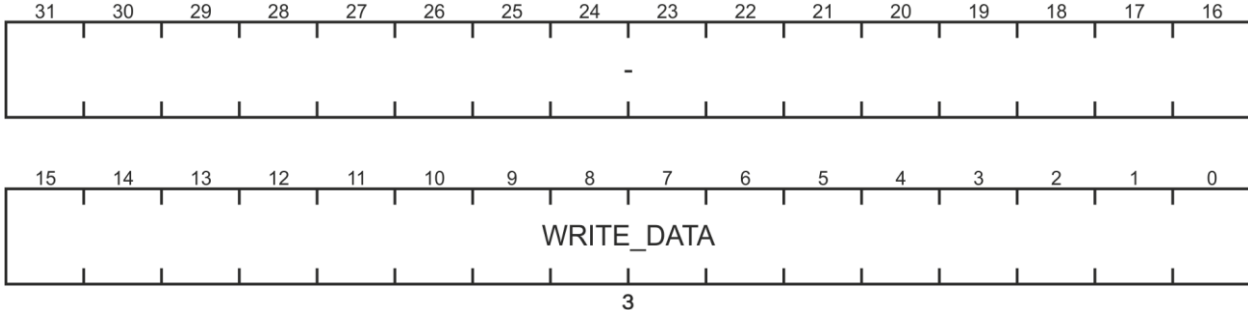
| | | |
|--|------------|---|
| MWTD | 8008_002Ch | Сброс: 00000000h |
|  | | |
| Поле | Биты | Описание |
| WRITE_DATA | 15-0 | Поле данных. При записи модуль управления МП использует данные этого регистра и предварительно сконфигурированные адрес PHY и МП для записи в PHY устройство |
| – | 31-16 | Зарезервировано |

Таблица А.4.16 – Регистр считываемых данных из МП

| | | | |
|-------------|-------|---|------------------|
| MRDD | | 8008_0030h | Сброс: 00000000h |
| | | | |
| Поле | Биты | Описание | |
| READ_DATA | 15-0 | После чтения модулем управления МП из PHY-устройства прочитанные данные можно считать из этого регистра (по адресу 0070000Ch) | |
| – | 31-16 | Зарезервировано | |

Таблица А.4.17 – Регистр флагов состояния МП

| | | | |
|-------------|------|---|-------------------|
| MIND | | 8008_0034h | Сброс: 00000000h |
| | | | |
| Поле | Бит | Описание | |
| LINKFAIL | 3 | Флаг сбоя управления МП (управление PHY-устройством) | |
| | | 0 | Нормальная работа |
| | 1 | Сбой | |
| NOTVALID | 2 | Флаг незавершенного цикла чтения МП (недоступности данных для чтения) | |
| | | 0 | |
| | 1 | Цикл чтения не завершен | |
| SCAN | 1 | Флаг продолжающейся скан-операции (продолжаются циклы чтения управления МП) | |
| | | 0 | |
| | 1 | Скан-операция в процессе | |
| BUSY | 0 | Флаг продолжающегося цикла чтения/записи управления МП | |
| | | 0 | |
| | 1 | Чтение/запись в процессе | |
| – | 31-4 | Зарезервировано | |

Таблица А.4.18 – Регистр состояния контроллера МП

| SMII | | 8008_0038h | Сброс: 00000000h |
|---|------|--|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: flex-end; gap: 10px;"> <div style="border: 1px solid black; padding: 2px;">CLA SH</div> <div style="border: 1px solid black; padding: 2px;">JAB BER</div> <div style="border: 1px solid black; padding: 2px;">LINK</div> <div style="border: 1px solid black; padding: 2px;">DUP LEX</div> <div style="border: 1px solid black; padding: 2px;">SPE ED</div> </div> | | | |
| <div style="display: flex; justify-content: flex-end; gap: 10px;"> 44444 </div> | | | |
| Поле | Бит | Описание | |
| CLASH | 4 | Флаг выбранного режима | |
| | | 0 | |
| | 1 | MAC-to-MAC, за исключением, если обнаружен PHY | |
| JABBER | 3 | Флаг возникновения условия Jabber | |
| | | 0 | Нет |
| | 1 | Условие обнаружено | |
| LINK | 2 | Индикатор состояния LINK | |
| | | 0 | Выключено |
| | 1 | Включено | |
| DUPLEX | 1 | Индикатор режима | |
| | | 0 | Half-Duplex |
| | 1 | Full-Duplex | |
| SPEED | 0 | Индикатор скорости | |
| | | 0 | 10Мбит/с |
| | 1 | 100Мбит/с | |
| – | 31-5 | Зарезервировано | |

Таблица А.4.19 – Регистр конфигурации буфера МПFIFO

| FIFOCFG | | 8008_003Ch | Сброс: 0000001Fh |
|---|-------|---------------------------------|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: flex-end; gap: 10px;"> ENRPLY </div> | | | |
| <div style="display: flex; justify-content: flex-end; gap: 10px;"> 4 </div> | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: flex-end; gap: 10px;"> ENREQRST </div> | | | |
| <div style="display: flex; justify-content: flex-end; gap: 10px;"> 3 43 4 </div> | | | |
| Поле | Бит | Описание | |
| ENRPLY | 20-16 | Индикация работы модулей МПFIFO | |
| | | 00h | Отключены |
| | | 1Fh | Включены |

Окончание таблицы А.4.19

| Поле | Бит | Описание |
|-------|-------------------|---|
| ENREQ | 12-8 | Разрешение запросов модуля МПFIFO |
| | | 00h Запрещены |
| | | 1Fh Разрешены |
| RST | 4-0 | Сброс модулей МПFIFO |
| | | 00h Модули функционируют |
| | | 1Fh Переключение в режим сброса модулей |
| – | 31-21, 15-13, 7-5 | Зарезервировано |

Таблица А.4.20 – Регистры адреса станции

| | | | | |
|--|--|------|--|------------------|
| SA0 | | | 8008_0040h | Сброс: 00000000h |
| | | | | |
| SA1 | | | 8008_0044h | Сброс: 00000000h |
| | | | | |
| SA2 | | | 8008_0048h | Сброс: 00000000h |
| | | | | |
| Поле | | Биты | Описание | |
| STATION_ADDRESS 1 Byte, STATION_ADDRESS 3 Byte, STATION_ADDRESS 5 Byte | | 15-8 | 1-й октет адреса станции, 3-й октет адреса станции, 5-й октет адреса станции | |

Окончание таблицы А.4.20

| Поле | Биты | Описание |
|--|------|--|
| STATION_ADDRESS 2 Byte, STATION_ADDRESS 4 Byte, STATION_ADDRESS 6 Byte | 7-0 | 2-й октет адреса станции, 2-й октет адреса станции, 2-й октет адреса станции |
| Биты 31-16 каждого регистра зарезервированы | | |

Таблица А.4.21 – Регистр управления передачей

| Поле | Бит | Описание |
|--|------|--|
| DMATXCTRL 8008_0180h Сброс: 00000000h | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 2px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; width: 100%; display: flex; align-items: center; justify-content: center;">-</div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 2px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; width: 100%; display: flex; align-items: center; justify-content: center;">- TX ENA BLE</div> | | |
| 3 ч | | |
| TXENABLE | 0 | Бит разрешения устройству ПДП доступа к пакету, предназначенному для передачи. Бит очищается встроенным ПДП-контроллером, если происходит неполная передача или возникает состояние ошибки в передающей линии |
| | 0 | Запрещено |
| | 1 | Разрешено |
| – | 31-1 | Зарезервировано |

Таблица А.4.22 – Регистр указателя дескриптора передачи

| Поле | Биты | Описание |
|--|------|--|
| DMATXDSCR 8008_0184h Сброс: 00000000h | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 2px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; width: 100%; display: flex; align-items: center; justify-content: center;">DESCRIPTOR_ADDRESS</div> | | |
| 3 ч | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 2px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; width: 100%; display: flex; align-items: center; justify-content: center;">DESCRIPTOR_ADDRESS -</div> | | |
| 3 ч | | |
| DESCRIPTOR_ADDRESS | 31-2 | Поле адреса регистра памяти, в котором расположены данные о первом передаваемом пакете. Встроенный ПДП контроллер читает этот регистр с целью получения адреса только в том случае, если процессором установлен бит TXENABLE в регистре DMATXCTRL |
| – | 1,0 | Зарезервировано |

Таблица А.4.23 – Регистр статуса передачи

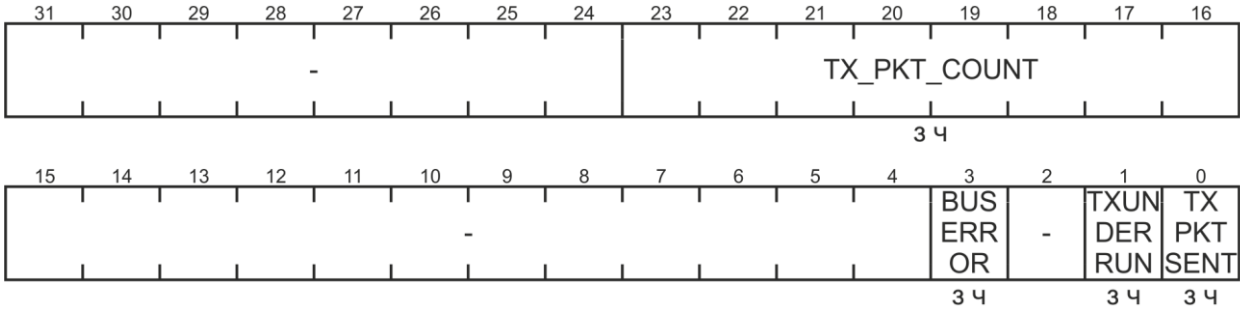
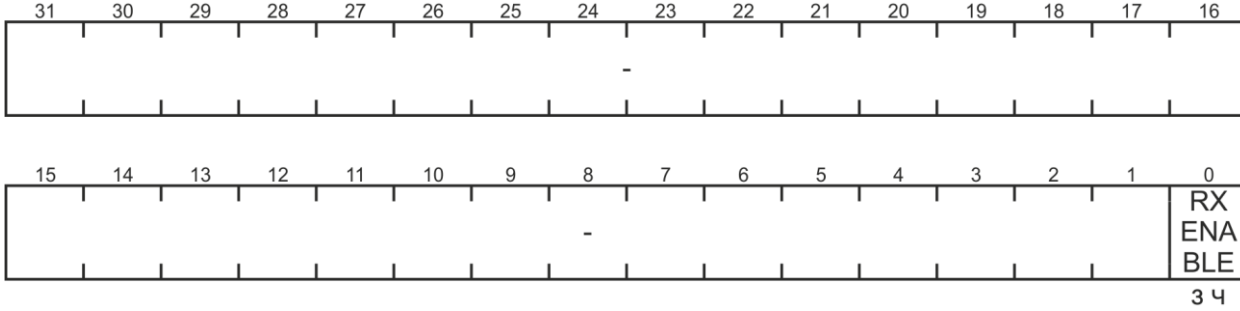
| DMATXSTAT | | 8008_0188h | Сброс: 00000000h |
|--|----------------|---|---|
|  | | | |
| Поле | Биты | Описание | |
| TX_PKT_COUNT | 23-16 | 8-битный счетчик пакетов передачи. Увеличивается на единицу каждый раз, когда встроенный ПДП контроллер успешно завершает передачу пакета, и уменьшается на единицу, если головной процессор записывает единицу в бит TXPKTSENT | |
| BUS ERROR | 3 | 0 | Нормальная работа |
| | | 1 | Ошибка |
| TXUNDER RUN | 1 | 0 | Индикатор обработки данных |
| | | 1 | ПДП контроллер читает и устанавливает флаг Empty в дескрипторе |
| TXPKT SENT | 0 | 0 | Флаг успешной передачи одного или более пакетов |
| | | 1 | Бит сбрасывается всякий раз, когда счетчик TX_PKT_COUNT обнуляется |
| - | 31-24, 15-4, 2 | 0 | Установка бита приводит к уменьшению значения TX_PKT_COUNT на единицу |
| | | 1 | Зарезервировано |

Таблица А.4.24 – Регистр управления приемом

| DMARXCTRL | | 8008_018Ch | Сброс: 00000000h |
|--|--|------------------|------------------|
|  | | | |
| | | RX ENABLE 3 4 | |

Окончание таблицы А.4.24

| Поле | Бит | Описание | |
|----------|------|--|---|
| RXENABLE | 0 | Бит разрешения устройству ПДП принимать пакеты. При установленном бите Бит очищается в случае обнаружения состояния RxOverflow или BusError | |
| | | 0 | Запрещено |
| | | 1 | Разрешено. Встроенный ПДП-контроллер готов начать принимать новый пакет, как только FIFO покажет, что новый пакет доступен (FRSOF подтвержден) |
| – | 31-1 | Зарезервировано | |

Таблица А.4.25 – Регистр указателя дескриптора приема

| Поле | Биты | Описание |
|--|------|---|
| DMARXDSCR 8008_0190h Сброс: 00000000h | | |
| <div style="text-align: center;"> </div> | | |
| DESCRIPTOR_ADDRESS | 31-2 | Поле адреса регистра памяти, в который будут записаны данные о первом принятом пакете. Встроенный ПДП контроллер читает этот регистр с целью получения адреса только в том случае, если процессором установлен бит RXENABLE в регистре DMATRCtrl |
| – | 1,0 | Зарезервировано |

Таблица А.4.26 – Регистр статуса приема

| Поле | Биты | Описание |
|--|------|----------|
| DMARXSTAT 8008_0194h Сброс: 00000000h | | |
| <div style="text-align: center;"> </div> | | |
| – | – | – |

Окончание таблицы А.4.26

| Поле | Биты | Описание |
|--------------|-------------------|--|
| RX_PKT_COUNT | 23-16 | 8-битный счетчик принятых пакетов. Увеличивается на единицу каждый раз, когда встроенный ПДП контроллер успешно завершает транзакцию пакета, и уменьшается на единицу, если головной процессор записывает единицу в бит RXPKTSENT |
| BUSERROR | 3 | Флаг ошибки на шине либо в режиме обмена данными процессором с регистрами устройства, либо в режиме прямого доступа к памяти. |
| | | 0 Нормальная работа 1 Ошибка |
| RXOVERFLOW | 2 | Индикатор обработки данных |
| | | 0 1 ПДП контроллер прочитал «0» из флага Empty в дескрипторе приема во время обработки данных |
| RXPKTREC | 0 | Флаг успешного приема одного или более пакетов |
| | | 0 Бит сбрасывается всякий раз, когда счетчик RX_PKT_COUNT обнуляется 1 Установка бита приводит к уменьшению значения RX_PKT_COUNT на единицу |
| – | 31-24, 15-4, 1 | Зарезервировано |

Таблица А.4.27 – Регистр маски прерываний

| Поле | Бит | Описание |
|--------------|-----|--|
| BUSERROR(RX) | 7 | Бит разрешения прерывания при установке бита BUSERROR регистра DMARxStatus |
| RXOVERFLOW | 6 | Бит разрешения прерывания при установке бита RXOVERFLOW регистра DMARxStatus |
| RXPKTREC | 4 | Бит разрешения прерывания при установке бита RXPKTREC регистра DMARxStatus |
| BUSERROR(TX) | 3 | Бит разрешения прерывания при установке бита BUSERROR регистра DMATxStatus |
| TXUNDERRUN | 1 | Бит разрешения прерывания при установке бита TXUNDERRUN регистра DMATxStatus |
| TXPKTSENT | 0 | Бит разрешения прерывания при установке бита TXPKTSENT регистра DMATxStatus |

| | | | | | | | | | | | | | | | |
|-------------------|----|------------|----|----|----|----|----|--------------------|--------------------|----|------------------|--------------------|----|--------------------|-------------------|
| DMAINTMASK | | 8008_0198h | | | | | | | | | | Сброс: 00000000h | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | BUS ERR (RX) | RX OVER FLOW | - | RX PKT REC | BUS ERR (TX) | - | TXUN DER RUN | TX PKT SENT |
| | | | | | | | | 3 4 | 3 4 | | 3 4 | 3 4 | | 3 4 | 3 4 |

Окончание таблицы А.4.27

| Поле | Биты | Описание |
|---|------------|-----------------|
| – | 31-8, 5, 2 | Зарезервировано |
| <p>Примечание – Для разрешения прерывания следует установить соответствующий бит. Сброшенный бит запрещает установку соответствующего флага в регистре DMAInterrupt (т. е. запрещает генерирование прерывания).</p> | | |

Таблица А.4.28 – Регистр прерываний

| Поле | Бит | Описание |
|---|------------|--|
| BUSERROR(RX) | 7 | Флаг прерывания при установке бита BUSERROR регистра DMARxStatus |
| RXOVERFLOW | 6 | Флаг прерывания при установке бита RXOVERFLOW регистра DMARxStatus |
| RXPKTREC | 4 | Флаг прерывания при установке бита RXPKTREC регистра DMARxStatus |
| BUSERROR(TX) | 3 | Флаг прерывания при установке бита BUSERROR регистра DMATxStatus |
| TXUNDERRUN | 1 | Флаг прерывания при установке бита TXUNDERRUN регистра DMATxStatus |
| TXPKTSENT | 0 | Флаг прерывания при установке бита TXPKTSENT регистра DMATxStatus |
| – | 31-8, 5, 2 | Зарезервировано |
| <p>Примечание – Флаг прерывания устанавливается только в том случае, если оба бита – бит-источник прерывания и соответствующий бит регистра маски прерываний – установлены.</p> | | |

А.5 Регистры контроллера USB OTG

Таблица А.5.1 – Регистр версии

| VERSION | | 8009_0000h | Сброс: 0000001h | | | | | | | | | | | | |
|---------|------|----------------------------------|-----------------|----|----|----|----|---------|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | VERSION | | | | | | | |
| ч | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| VERSION | 7-0 | Версия используемого контроллера | | | | | | | | | | | | | |
| - | 31-8 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.5.2 – Регистр управления 1

| USBCMD_STS_INTR | | 8009_0004h | Сброс: 01008000h | | | | | | | | | | | | | | | | | |
|------------------|-------|--|---|------------------------|---------------------|----|----|----|-----------------------|----|-------------------------|-----------------------|--------------------------|-----------------------|--------------------|-------------------|--------------|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | |
| INT_SPACE | | | | | | | | | | | | - | | | CON FIG FLAG | HOST RES ET | RUN/ STOP | | | |
| ч | | | | | | | | | | | | ч | | | 3 | 4 | 3 | 4 | 3 | 4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | |
| HC HALT ED | - | PORT CHNG FLAG | - | MICRO FRAME FLAG | FRAME SOF INT | - | - | - | PORT CHNG INTEN | - | MICRO FRAME INTEN | FRAME SOF INTEN | INDATA READY INTEN | OUT SPACE INTEN | - | | | | | |
| ч | | 3 | 4 | 3 | 4 | 3 | 4 | | 3 | | 3 | 3 | 3 | 4 | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | |
| INT_SPACE | 31-24 | Управление интервалами прерываний. В текущей версии в этом поле записано значение 01h, означающее, что запросы прерываний будут передаваться хосту с интервалом 1 мс | | | | | | | | | | | | | | | | | | |
| CONFIGFLAG | 18 | Бит конфигурации | | | | | | | | | | | | | | | | | | |
| | | 0 | Нет действий | | | | | | | | | | | | | | | | | |
| | 1 | По умолчанию портами управляет контроллер хоста Бит устанавливается программой пользователя и показывает, что контроллер хоста сконфигурирован | | | | | | | | | | | | | | | | | | |
| HOSTRESET | 17 | Установка бита переводит контроллер и все регистры в исходное состояние. По окончании сброса контроллер сбрасывает этот бит | | | | | | | | | | | | | | | | | | |
| RUN/STOP | 16 | Бит запуска работы Хост контроллера | | | | | | | | | | | | | | | | | | |
| | | 0 | Сброс бита приводит к тому, что контроллер завершает текущую передачу данных и останавливается, устанавливая бит HCHALTED в единицу | | | | | | | | | | | | | | | | | |
| | | 1 | Контроллер выполняет передачу данных, счетчик кадров запускается с начального значения, указанного в регистре frame_index | | | | | | | | | | | | | | | | | |

Окончание таблицы А.5.2

| Поле | Бит | Описание |
|--------------------------------|-----|---|
| HCHALTED | 15 | Индикатор остановки контроллера |
| | | 0 Запуск контроллера |
| | | 1 Остановка контроллера |
| PORTCHNG FLAG | 13 | Контроллер устанавливает этот бит когда любой из битов CHANGE переходит из нуля в единицу, или бит FORCEPORTRESUME регистра PORT_STS устанавливается в единицу после детектирования перехода J-K в приостановленном порту. Бит сбрасывается записью единицы |
| MICROFRAME FLAG | 11 | Флаг прерывания Microframesof. Контроллер устанавливает этот бит каждые 125 мкс для индикации начала микрокадра. Бит сбрасывается записью единицы |
| FRAMESOFINT | 10 | Флаг прерывания Framesoa. Контроллер устанавливает этот бит в единицу каждые 1 мс для индикации начала кадра |
| PORTCHNG INTEN | 6 | Бит разрешения прерывания Port change detect |
| | | 0 Запрещено |
| | | 1 Разрешено |
| MICROFRAME INTEN | 4 | Бит разрешения прерывания Microframe soa |
| | | 0 Запрещено |
| | | 1 Разрешено |
| FRAMESOF INTEN | 3 | Бит разрешения прерывания Frame sof |
| | | 0 Запрещено |
| | | 1 Разрешено |
| INDATA READY INTEN | 2 | Бит разрешения прерывания InDataReady |
| | | 0 Запрещено |
| | | 1 Разрешено, если установлен бит slave mode access |
| OUTSPACE INTEN | 1 | Бит разрешения аппаратного прерывания Out_space_available |
| | | 0 Запрещено |
| | | 1 Разрешено, если установлен бит slave mode access |
| Остальные биты зарезервированы | | |

Таблица А.5.3 – Регистр версии

| FRAME_REG | | 8009_0008h | Сброс: 00200000h | | | | | | | | | | | | |
|------------------|-------|--|------------------|----|----|----|----|----|----|----|-------------|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | FRAMELENGTH | | | | |
| | | | | | | | | | | | 4 | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | FRAMEINDEX | | | | | | | | | | | | | |
| | | 3 4 | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| FRAMELENGTH | 21-16 | В текущей версии поле содержит значение 20h, указывающее на длину микрокадра 60000 бит | | | | | | | | | | | | | |

Окончание таблицы А.5.3

| Поле | Бит | Описание |
|------------|-----------------|---|
| FRAMEINDEX | 13-0 | Поле обратного счетчика. В 11 старших битах поля хранится номер кадра. В трех младших битах – номер микрокадра. Запись в поле разрешена контроллеру только в том случае, если установлен бит HCHALTED регистра USBCMD/STS/INT |
| – | 31-22, 15-14 | Зарезервировано |

Таблица А.5.4 – Регистр управления 2

| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-----|---|---------------------|----------------------|-------------------|-------------------------|-------------|-------------------|---------------|-------------------|-----------------------|--------------------|---------------------|---------------------|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------------|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|---------------------|----------------------|-------------------|-------------------------|-------------|-------------------|---------------|-------------------|-----------------------|--------------------|---------------------|---------------------|--|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <p>PORT_STS 8009_0010h Сброс: 00000000h</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> </tr> <tr> <td colspan="13"></td> <td>PORT OWN ER</td> <td>-</td> </tr> <tr> <td colspan="15"></td> <td>4</td> </tr> <tr> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="3"></td> <td>XCVR SUSP END</td> <td>XCVR RESL EVEL</td> <td>XCVR RES ET</td> <td>FORCE PORT RESUME</td> <td>SUSP END</td> <td>PORT OWN ER</td> <td>LINE STATE</td> <td>PORT RES ET</td> <td>PORT N/DIS CHNG</td> <td>PORT ENA BLE</td> <td>PORT CON CHNG</td> <td>PORT CON NECT</td> </tr> <tr> <td colspan="3"></td> <td>3</td><td>4</td> <td>3</td><td>4</td> <td>3</td><td>4</td> <td>4</td> <td>4</td> <td>3</td><td>4</td> <td>3</td><td>4</td> <td>4</td> </tr> </table> | | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | | | | | | PORT OWN ER | - | | | | | | | | | | | | | | | | 4 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | XCVR SUSP END | XCVR RESL EVEL | XCVR RES ET | FORCE PORT RESUME | SUSP END | PORT OWN ER | LINE STATE | PORT RES ET | PORT N/DIS CHNG | PORT ENA BLE | PORT CON CHNG | PORT CON NECT | | | | 3 | 4 | 3 | 4 | 3 | 4 | 4 | 4 | 3 | 4 | 3 | 4 | 4 |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | PORT OWN ER | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | XCVR SUSP END | XCVR RESL EVEL | XCVR RES ET | FORCE PORT RESUME | SUSP END | PORT OWN ER | LINE STATE | PORT RES ET | PORT N/DIS CHNG | PORT ENA BLE | PORT CON CHNG | PORT CON NECT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | 3 | 4 | 3 | 4 | 3 | 4 | 4 | 4 | 3 | 4 | 3 | 4 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PORTOWNER | 18 | Индикатор низкоскоростного устройства. Бит функционирует совместно с 7 битом регистра 0 Подключено высокоскоростное устройство 1 Подключено низкоскоростное устройство | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XCVR SUSPEND | 12 | Приостановка приемопередатчика 0 – 1 Установка бита указывает на то, что приемопередатчик следует приостановить на время, определяемое драйвером. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XCVRRE LEVEL | 11 | Бит задания активного уровня сигнала сброса приемопередатчика 0 Низкий 1 Высокий | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| XCVRRESET | 10 | Бит сброса приемопередатчика 0 Сброс бита выводит приемопередатчик из состояния сброса 1 Установка переводит приемопередатчик в состояние сброса | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FORCEPORT RESUME | 9 | Флаг возобновления 0 Нет возобновления 1 Обнаружена команда возобновления Функционирование этого бита зависит от состояния бита SUSPEND. Если порт не приостановлен результат перевода этого бита в единицу не определен. Если порт приостановлен, то контроллер устанавливает этот бит при детектировании перехода J-K. В этом случае также устанавливается бит Port Change Detect. Программно бит FORCEPORTRESUME можно установить для сигнализации возобновления работы. Установки бита Port Change Detect при этом не происходит | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.5.4

| Поле | Бит | Описание | | |
|--------------------|-----------|--|--|------------|
| SUSPEND | 8 | Бит приостановки работы | | |
| | | 0 | Запись нуля в бит игнорируется контроллером. Контроллер самостоятельно сбрасывает бит в случаях, если бит FORCEPORTRESUME изменяется программно | |
| | | 1 | Порт находится в приостановленном состоянии. В этом состоянии блокируется любая передача данных, кроме сброса порта. Если в момент установки бита проводится передача данных, блокирование происходит по ее окончании. В приостановленном состоянии порт может воспринимать команду возобновления работы | |
| PORTOWNER | 7 | Бит задания скорости работы порта. Функционирует совместно с 18 битом регистра | | |
| | | PORTOWNER | | |
| | | 18-й бит | 7-й бит | Скорость |
| | | 0 | 0 | High Speed |
| | | 0 | 1 | Full Speed |
| | | 1 | 0 | Low Speed |
| LINE STATE | 6-5 | Поле отражения текущего логического уровня сигналов D+ и D- | | |
| | | 00 | SE0 – Не LS; сброс | |
| | | 01 | J – Не LS; сброс | |
| | | 10 | K – LS; игнорировать устройство | |
| | | 11 | Зарезервировано | |
| PORTRESET | 4 | Установка бита запускает процедуру сброса согласно стандарту USB2.0. Сброс бита завершает процедуру. По завершении сброса состояние поля LINE STATE определяет установленную скорость шины | | |
| | | LINE STATE | Скорость шины | |
| | | 00 | HS | |
| | | 01 | Неверно | |
| | | 10 | FS | |
| | | 11 | Неверно | |
| PORTEN/ DISCHNG | 3 | Индикатор изменения состояния бита PORTENABLE | | |
| | | 0 | Нет изменения | |
| | | 1 | Бит изменился | |
| PORTENABLE | 2 | Бит включения порта | | |
| | | 0 | Порт выключен | |
| | | 1 | Порт включен | |
| | | Порт может быть включен только контроллером во время процедуры сброса и не может быть включен программно, но может быть выключен путем сброса бита | | |
| PORT CONCHNG | 1 | Индикатор изменения состояния бита PORTCONNECT | | |
| | | 0 | Нет изменения | |
| | | 1 | Бит изменился | |
| PORT CONNECT | 0 | Индикатор подключения устройства | | |
| | | 0 | Устройство не подключено | |
| | | 1 | Устройство подключено | |
| – | 31- 19 | Зарезервировано | | |

Таблица А.5.5 – Регистр А управления передачей данных

| USB_PKT_FLD_A | | 8009_0014h | | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------------|---|---|------------------|----|----|--------------------|-------------|---------|----|----|----|----|----|----|----|----|----|----|----|----|--------------------|-------------|--|--|--|--|--|--------------------|-------------|---------|--|--|--|--|--|--|-----|-----|--|--|--|--|--|-----|-----|-----|--|--|--|--|--|--|
| <table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:3.33%; text-align:center;">31</td> <td style="width:3.33%; text-align:center;">30</td> <td style="width:3.33%; text-align:center;">29</td> <td style="width:3.33%; text-align:center;">28</td> <td style="width:3.33%; text-align:center;">27</td> <td style="width:3.33%; text-align:center;">26</td> <td style="width:3.33%; text-align:center;">25</td> <td style="width:3.33%; text-align:center;">24</td> <td style="width:3.33%; text-align:center;">23</td> <td style="width:3.33%; text-align:center;">22</td> <td style="width:3.33%; text-align:center;">21</td> <td style="width:3.33%; text-align:center;">20</td> <td style="width:3.33%; text-align:center;">19</td> <td style="width:3.33%; text-align:center;">18</td> <td style="width:3.33%; text-align:center;">17</td> <td style="width:3.33%; text-align:center;">16</td> </tr> <tr> <td style="text-align:center;">EN TRAN SFER</td> <td colspan="6" style="text-align:center;">HUB_ADDRESS</td> <td style="text-align:center;">DATA TOG GLE</td> <td colspan="8" style="text-align:center;">PORT_NUMBER</td> </tr> <tr> <td style="text-align:center;">3 ч</td> <td colspan="6" style="text-align:center;">3 ч</td> <td style="text-align:center;">3 ч</td> <td colspan="8" style="text-align:center;">3 ч</td> </tr> </table> | | | | | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | EN TRAN SFER | HUB_ADDRESS | | | | | | DATA TOG GLE | PORT_NUMBER | | | | | | | | 3 ч | 3 ч | | | | | | 3 ч | 3 ч | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EN TRAN SFER | HUB_ADDRESS | | | | | | DATA TOG GLE | PORT_NUMBER | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 ч | 3 ч | | | | | | 3 ч | 3 ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:3.33%; text-align:center;">15</td> <td style="width:3.33%; text-align:center;">14</td> <td style="width:3.33%; text-align:center;">13</td> <td style="width:3.33%; text-align:center;">12</td> <td style="width:3.33%; text-align:center;">11</td> <td style="width:3.33%; text-align:center;">10</td> <td style="width:3.33%; text-align:center;">9</td> <td style="width:3.33%; text-align:center;">8</td> <td style="width:3.33%; text-align:center;">7</td> <td style="width:3.33%; text-align:center;">6</td> <td style="width:3.33%; text-align:center;">5</td> <td style="width:3.33%; text-align:center;">4</td> <td style="width:3.33%; text-align:center;">3</td> <td style="width:3.33%; text-align:center;">2</td> <td style="width:3.33%; text-align:center;">1</td> <td style="width:3.33%; text-align:center;">0</td> </tr> <tr> <td style="text-align:center;">IOC</td> <td colspan="6" style="text-align:center;">DEV_ADDR</td> <td style="text-align:center;">ENDPOINT TYPE</td> <td style="text-align:center;">PID</td> <td colspan="7" style="text-align:center;">EP_ADDR</td> </tr> <tr> <td style="text-align:center;">3 ч</td> <td colspan="6" style="text-align:center;">3 ч</td> <td style="text-align:center;">3 ч</td> <td style="text-align:center;">3 ч</td> <td colspan="7" style="text-align:center;">3 ч</td> </tr> </table> | | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | IOC | DEV_ADDR | | | | | | ENDPOINT TYPE | PID | EP_ADDR | | | | | | | 3 ч | 3 ч | | | | | | 3 ч | 3 ч | 3 ч | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IOC | DEV_ADDR | | | | | | ENDPOINT TYPE | PID | EP_ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 ч | 3 ч | | | | | | 3 ч | 3 ч | 3 ч | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ENTRANSFER | 31 | Бит разрешения начала текущей передачи данных. По завершении передачи бит сбрасывается аппаратно | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Нет действий | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Разрешено. Программно можно только устанавливать бит при необходимости начать передачу с перемещением количества данных, указанного в регистре TBT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HUB_ADDRESS | 30-24 | Поле адреса маршрутизатора, к которому обращена передача типа "Split" | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DATATOGGLE | 23 | Поле toggle bit данных, который будет использован в текущей передаче | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_NUMBER | 22-16 | Поле номера порта маршрутизатора, к которому подключено устройство FS/LS. Используется только при передачах типа "Split" | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IOC | 15 | Бит задания поведения контроллера хоста по окончании передачи | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | По окончании передачи контроллер хоста не запросит прерывание ЦПУ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | По окончании передачи контроллер хоста запросит прерывание ЦПУ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DEV_ADDR | 14-8 | Поле адреса, к которому обращается текущая операция | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ENDPOINTTYPE | 7-6 | Поле задания типа буфера | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 00 | Control | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 01 | Iso | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 10 | Bulk | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 11 | Interrupt | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PID | 5-4 | Поле метки, которая должна быть послана в нисходящем потоке | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 00 | Out | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 01 | In | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 10 | Setup | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 11 | Ping | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EP_ADDR | 3-0 | Поле адреса буфера, к которому обращена передача | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.5.6 – Регистр В управления передачей данных

| USB_PKT_FLD_B | | 8009_0018h | | Сброс: 00000000h | | | | | | | | | | | |
|--------------------|------------|--|--|-----------------------|-----|-----|----|-------------|----|----|----|-----|--------------------|-----------------------|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ONCI ENA BLE | | - | SLA VE MODE | MAX_NAK_CNT | | | | MAX_ERR_CNT | | | | - | STA RT SPLIT | COMP LETE SPLIT | |
| 3 4 | | 3 4 | | 3 4 | | | | 3 4 | | | | 3 4 | | 3 4 | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPE ED | EFI ELD | IGNOR SHORT PKT | XFR LEV EL | ABORT TRANS FER | MPS | | | | | | | | | | |
| 3 4 | | 3 4 | | 3 4 | | 3 4 | | 3 4 | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| ONCIENABLE | 31 | Индикатор запроса | | | | | | | | | | | | | |
| | | 0 | В регистре USB_PKT_FIELD_B содержится запрос на начало high-speed операции контроллером | | | | | | | | | | | | |
| | | 1 | В регистре USB_PKT_FIELD_B содержится запрос на начало full-speed операции контроллером | | | | | | | | | | | | |
| SLAVEMODE | 28 | Бит задания режима, в котором будет происходить обращение к буферу данных | | | | | | | | | | | | | |
| | | 0 | Мастер | | | | | | | | | | | | |
| | | 1 | Ведомый | | | | | | | | | | | | |
| MAX_NAK_CNT | 27-24 | Поле задания максимального числа меток NAK, которые разрешено принять хосту. При установке этого поля в ноль хост будет посылать запросы бесконечно | | | | | | | | | | | | | |
| MAX_ERR_CNT | 23-20 | Поле максимально допустимого числа последовательных ошибок передачи. При установке этого поля в ноль хост будет посылать запросы бесконечно | | | | | | | | | | | | | |
| STARTSPLIT | 17 | Бит запуска операции типа Split. Для запуска контроллером операции этот бит следует установить одновременно с битом ENTRANSFER регистра USB_PKT_FLD_A | | | | | | | | | | | | | |
| COMPLETESPLIT | 16 | Бит завершения операции типа Split. Для завершения операции этот бит следует установить одновременно с битом ENTRANSFER регистра USB_PKT_FLD_A | | | | | | | | | | | | | |
| SPEED | 15 | Индикатор типа устройства, подключенного к порту, указанным в полях HUB_ADDRESS и PORT_NUMBER, соответственно (регистр USB_PKT_FLD_A) | | | | | | | | | | | | | |
| | | 0 | Устройство Low Speed | | | | | | | | | | | | |
| | | 1 | Устройство Full Speed | | | | | | | | | | | | |
| EFIELD | 14 | Бит управления передачей. Используется совместно с битом SPEED для передач типа Isosplit для идентификации загрузки начала, середины или конца загрузки | | | | | | | | | | | | | |
| IGNORSHORTPKT | 13 | Индикация игнорирования коротких пакетов | | | | | | | | | | | | | |
| | | 0 | Передача в обычном режиме | | | | | | | | | | | | |
| | | 1 | Контроллер продолжает посылать токены, счетчик пакетов TBT не равен 0, независимо от приема безошибочных коротких, или нулевых пакетов | | | | | | | | | | | | |

Окончание таблицы А.5.6

| Поле | Бит | Описание | |
|----------------|--------------|---|---|
| XFRLEVEL | 12 | Уровень передач | |
| | | 0 | Статус каждой транзакции обновляется сразу после ее завершения |
| | | 1 | При установке бита контроллер должен переслать количество данных, указанное в регистре TBT за одну, или более операций. При этом должны быть указаны условия повторных запросов при приеме меток NAK, NYET и ошибках в состоянии шины |
| ABORT TRANSFER | 11 | Бит завершения текущей операции | |
| | | 0 | Нет действий |
| | | 1 | Установка бита указывает контроллеру завершить текущую операцию и вернуться в исходное состояние |
| MPS | 10-0 | Поле максимального размера пакета в нисходящем потоке | |
| – | 30-29, 19-18 | Зарезервировано | |

Таблица А.5.7 – Регистр передаваемых данных

| Поле | Бит | Описание |
|---|-------|---|
| <p>TBT 8009_0020h Сброс: 00000000h</p> | | |
| | | |
| TOTAL_BYTES_TRANS | 19-0 | Поле числа байт, которые требуется переслать устройству одним IRP. Одиночный IRP может быть расширен как на IN, так и на OUT транзакции |
| – | 31-20 | Зарезервировано |

Таблица А.5.8 – Регистр состояния системы

| Поле | Бит | Описание |
|--|-----|----------|
| <p>USB_STATUS 8009_0024h Сброс: 00000000h</p> | | |
| | | |
| USB_ERR_INT | 25 | 4 |
| USB_INT | 24 | 4 |
| HALTED | 23 | 4 |
| BABBLE | 22 | 4 |
| XACTNER_ROR | 21 | 4 |
| PING | 20 | 4 |
| PAGE_CROSS | 19 | 4 |
| NAK | 18 | 4 |
| NYET | 17 | 4 |
| TIME_OUT | 16 | 4 |
| SHORT_PKT | 9 | 4 |
| UNEXPECTED_PID | 8 | 4 |
| TOGGLE_ERROR | 15 | 4 |
| CRC_ERROR | 14 | 4 |
| PID_ERROR | 13 | 4 |
| INDATA_READY | 12 | 4 |
| OUTSPACE_AVAILABLE | 11 | 4 |
| ERR_PID | 10 | 4 |

Окончание таблицы А.5.8

| Поле | Бит | Описание |
|---------------|---------------|--|
| USBERRINT | 25 | Флаг ошибки, указывающий на то, что последняя операция была завершена из-за ошибочного состояния шины. Запись 1 сбрасывает бит |
| USBINT | 24 | Флаг завершения последней операции Запись 1 сбрасывает бит |
| HALTED | 23 | Флаг нахождения системы в состоянии Stall или Packet Babble |
| BABBLE | 22 | Флаг нахождения системы в состоянии Packet Babble (если количество пришедших байт больше, чем количество запрошенных) |
| XACTNERROR | 21 | Флаг ошибки, устанавливающийся при любой ошибке, возникающей при выполнении операции |
| PING | 20 | Флаг отклика, устанавливающийся в случае, если выполняемая операция является выходящей, высокоскоростной и асинхронной, и устройство отвечает меткой NAK, или NYET |
| PAGECROSS | 19 | Бит устанавливается, если при передачи данных требуется pagecross |
| NAK | 18 | Флаг приема метки NAK |
| NYET | 17 | Флаг приема метки NYET |
| TIMEOUT | 16 | Флаг истечения времени ожидания ответа |
| TOGGLEERROR | 15 | Этот бит установится в случае приема неверного бит data toggle в процессе транзакции от устройства |
| CRCERROR | 14 | Флаг наличия ошибки CRC в принятых данных |
| PIDERROR | 13 | Флаг приема неверного PID |
| INDATAREADY | 12 | Флаг успешного завершения входящей операции. Устанавливается в единицу только при установленном бите SLAVEMODE регистра USB_PKT_FLD_B |
| OUTSPACEAVLBL | 11 | Флаг запроса исходящей операции при пустом буфере данных. Устанавливается в единицу только при установленном бите SLAVEMODE регистра USB_PKT_FLD_B |
| ERR_PID | 10 | Флаг приема ответа ERROR PID на метку Complete Split |
| SHORT_PKT | 9 | Флаг приема меньшего числа бит, чем указано в поле TOTAL_BYTES_TRANS регистра TBT |
| UNEXPECTEDPID | 8 | Флаг приема PID, отличного от ожидаемого |
| – | 31-26, 7-0 | Зарезервировано |

Таблица А.5.9 – Регистр принятых данных

| | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------|----|----|----|-------|----|----|----|---|----|----|----|----|----|------------|----|-----------------|--|------------------|--|--|--|--|--|
| SLAVE_IN_COUNT | | | | | | | | | | | | | | 8009_0030h | | | | Сброс: 00000000h | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | - | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | TOTAL_BYTES_REC | | | | | | | |
| 3 ч | | | | | | | | | | | | | | | | | | | | | | | |
| Поле | | | | Бит | | | | Описание | | | | | | | | | | | | | | | |
| TOTAL_BYTES_REC | | | | 10-0 | | | | Указатель количества байт, принятых от устройства при запросе прерывания «Data ready» | | | | | | | | | | | | | | | |
| - | | | | 31-11 | | | | Зарезервировано | | | | | | | | | | | | | | | |

Таблица А.5.10 – Регистры Хоста

| Мнемоника | Адрес | Доступ к регистру | Назначение и описание | Сброс |
|------------------------|-----------|-------------------|--|-----------|
| BUFF_START_ADDR | 8009001Ch | Запись Чтение | 32-разрядный регистр адреса начала массива передаваемых данных | 00000000h |
| DataBuffer | 8009002Ch | Запись Чтение | 32-разрядный регистр данных. Запись или чтение из этого регистра приводит к обращению к буферу данных внутри хоста. ВПО должно выполнять запись/чтение из этого регистра для заполнения/ опустошения буфера в режиме ведомого устройства | 00000000h |

Таблица А.5.11 – Регистр состояния прерываний

| | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|-----|----|----|----|--|----|----|----|----|----|------------|----|---|--|------------------|--|---------|---------|---------|-------------|
| IRQ_STAT_L | | | | | | | | | | | | | | 8009_0000h | | | | Сброс: 00000000h | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | - | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | - | | | | EP1_INT | EP0_INT | CEP_INT | USB BUS INT |
| | | | | | | | | | | | | ч | | ч | | ч | | ч | | | | | |
| Поле | | | | Бит | | | | Описание | | | | | | | | | | | | | | | |
| EP1_INT | | | | 3 | | | | Флаг прерывания конечной точки EP2. Причина прерывания отображается в регистре USB_IRQ_STAT | | | | | | | | | | | | | | | |

Окончание таблицы А.5.11

| Поле | Бит | Описание |
|-----------|------|---|
| EP0_INT | 2 | Флаг прерывания конечной точки EP1. Причина прерывания отображается в регистре USB_IRQ_STAT Запись 1 сбрасывает бит |
| CEP_INT | 1 | Флаг прерывания управляющего буфера Причина прерывания отображается в регистре USB_IRQ_STAT Запись 1 сбрасывает бит |
| USBBUSINT | 0 | Флаг прерывания от событий на шине USB. Причина прерывания отображается в регистре USB_IRQ_STAT Запись 1 сбрасывает бит |
| – | 31-4 | Зарезервировано |

Таблица А.5.12 – Регистр разрешения прерываний

| Поле | Бит | Описание |
|-------------|------|--|
| EP1_INTEN | 3 | Биты разрешения прерываний для конечной точки EP2 |
| EP0_INTEN | 2 | Биты разрешения прерываний для конечной точки EP1 |
| CEP_INTEN | 1 | Бит разрешения прерываний для управляющей конечной точки |
| USBBUSINTEN | 0 | Бит разрешения прерываний от событий на шине USB |
| – | 31-4 | Зарезервировано |

Примечание – Для разрешения прерывания следует установить соответствующий бит.

Таблица А.5.13 – Регистр состояния прерываний

| Поле | Бит | Описание |
|-----------------------------|-----|---|
| CLKUNSTBLINT (CLKUSABLEINT) | 6 | Устанавливается при наличии стабильной тактовой частоты блока USB.(при низкой тактовой частоте работы контроллера есть вероятность пропуска этого прерывания) |
| DMACMPLINT | 5 | Флаг завершения передачи данных по DMA |
| HIGHSPEEDSETTLE | 4 | Флаг завершения сброса и переключения устройства в режим high-speed |

Окончание таблицы А.5.13

| Поле | Бит | Описание |
|--|------|--|
| SUSPENDREQ | 3 | Бит запроса режима SUSPEND. По умолчанию, бит установлен и должен быть сброшен перед сбросом USB. Также бит устанавливается, при приходе запроса остановки от хоста |
| RESUME | 2 | Флаг возобновления работы устройства |
| RESSTATUS | 1 | Флаг завершения сброса корневого порта |
| SOF | 0 | Флаг приема пакета SOF |
| – | 15-7 | Зарезервировано |
| Примечание – Все флаги (за исключением флага CLKUNSTBLINT) сбрасываются программно, записью единицы в соответствующий бит. | | |

Таблица А.5.14 – Регистр разрешения прерываний

| USB_IRQ_ENB | | | | | | | | | | | | | | | | 8009_0014h | | | | Сброс: 0040h | | | | | | | | | | | |
|---|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--------------------------------------|--|---------------|--|--------------|--|--------------|--|-------------|--|----------------|--|----------|--|--|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | CLKUNSTBLINTEN | | DMA CMPLINTEN | | HISPEEDINTEN | | SUSPENDINTEN | | RESUMEINTEN | | RESSTATUSINTEN | | SOFINTEN | | | |
| | | | | | | | | | | - | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Биты с шестого по нулевой являются битами разрешения установки флагов прерываний в регистре USB_IRQ_STAT. Для разрешения прерывания следует записать единицу в соответствующий бит. | | | | | | | | | | | | | | | | Биты с 15 по седьмой зарезервированы | | | | | | | | | | | | | | | |

Таблица А.5.15 – Регистр операций

| USB_OPER | | | | | | | | | | | | | | | | 8009_0018h | | | | Сброс: 0000h | | | | | | | | | | | | | |
|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--------------|--|---|--|--------------|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | CURRENTSPEED | | HIGHSPEED | | GENRESUME | | | | | | | | | | | | | |
| | | | | | | | | | | - | | | | | | | | | | | | | | | | | | | | | | | |
| Поле | | | | | | | | | | | | | | | | Бит | | Описание | | | | | | | | | | | | | | | |
| CURRENTSPEED | | | | | | | | | | | | | | | | 2 | | Индикатор скорости работы контроллера устройства | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 0 | | Full Speed | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 1 | | High Speed | | | | | | | | | | | | | | | |
| HIGHSPEED | | | | | | | | | | | | | | | | 1 | | Бит запуска протокола «Chirp» | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 0 | | Протокол «Chirp» не запускается, позволяя устройству начать работу в режиме Full Speed даже если оно подключено к хосту USB 2.0 | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 1 | | Установка бита указывает контроллеру запустить протокол «Chirp» при выполнении сброса | | | | | | | | | | | | | | | |
| GEN_RESUME | | | | | | | | | | | | | | | | 0 | | Бит запуска восстановления работы | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 0 | | Нет действий | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 1 | | Установка бита запускает последовательность восстановления работы, если удаленное восстановление разрешено | | | | | | | | | | | | | | | |
| – | | | | | | | | | | | | | | | | 15-3 | | Зарезервировано | | | | | | | | | | | | | | | |

Таблица А.5.16 – Регистр счетчика кадров

| | | |
|----------------------|------------|--|
| USB_FRAME_CNT | 8009_001Ch | Сброс: 0000h |
| | | |
| Поле | Бит | Описание |
| FRAME_COUNTER | 13-3 | Поле счетчика кадров с момента последнего пакета SOF |
| MICROFRAME_COUNTER | 2-0 | Поле номера текущего микрокадра |
| – | 15-14 | Зарезервировано |

Таблица А.5.17 – Регистр адреса

| | | |
|-----------------|------------|--|
| USB_ADDR | 8009_0020h | Сброс: 0000h |
| | | |
| Поле | Бит | Описание |
| CURRENT_ADDR | 6-0 | Поле текущего адреса устройства. Поле сбрасывается при сбросе корневого порта |
| – | 15-7 | Зарезервировано |

Таблица А.5.18 – Регистр управления и состояния буфера

| | | |
|----------------------|------------|--|
| CEP_CTRL_STAT | 8009_002Ch | Сброс: 0000h |
| | | |
| Поле | Бит | Описание |
| CEP-FLUSH | 3 | Бит сброса буфера Установка бита сбрасывает содержимое буфера |
| ZEROLEN | 2 | Бит разрешения передачи пакета нулевой длины |
| | | 0 Запрещено 1 При работе в режиме "Auto Validation" установка этого бита разрешает посылку пакета нулевой длины в ответ на метку IN. Сразу после отправки пакета бит сбрасывается |
| STALL | 1 | Бит разрешения отправки «Stall» После записи единицы в бит, управляющий буфер отправляет подтверждение "Stall" в ответ на любую метку IN, или OUT. При этом бит NAK_CLEAR должен быть сброшен, т. к. он имеет более высокий приоритет |

Окончание таблицы А.5.18

| Поле | Бит | Описание |
|-----------|------|---|
| NAK_CLEAR | 0 | Флаг приема метки SETUP Устанавливается контроллером каждый раз при приеме метки SETUP. Пока бит установлен, контроллер будет отвечать меткой NAK на все запросы |
| – | 15-4 | Зарезервировано |

Таблица А.5.19 – Регистр разрешения прерываний буфера

| CEP_IRQ_ENB | | 8009_0030h | | | | | | | | | | | | | | Сброс: 0000h | | | | | | | | | | | | |
|--|----|------------|------------------------|-----------------------|------------------------|---------------------|------------------------|----------------------|-------------------------|------------------------|------------------------|----------------------|-----------------------|-----------------------|-------------------------|--------------|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | |
| | | | BUFF EMPTY INTEN | BUFF FULL INTEN | STAT CMLPN INTEN | USB ERR INTEN | STALL SENT INTEN | NAK SENT INTEN | DATAP KTREC INTEN | DATA PKTTR INTEN | PING TOKEN INTEN | IN TOKEN INTEN | OUT TOKEN INTEN | SETUP PKT INTEN | SETUP TOKEN INTEN | | | | | | | | | | | | | |
| | | | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 |
| <p>Биты с 12 по нулевой являются битами разрешения установки флагов прерываний в регистре CEP_IRQ_STAT. Для разрешения прерывания следует записать единицу в соответствующий бит.</p> <p>Биты с 15 по 13 зарезервированы</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.5.20 – Регистр флагов прерываний буфера

| CEP_IRQ_STAT | | 8009_0034h | | | | | | | | | | | | | | Сброс: 0000h | | | | | | | | | | | | | |
|---------------|-----|---|----------------------|--------------------|----------------------|-------------------|----------------------|--------------------|-----------------------|----------------------|----------------------|--------------------|---------------------|---------------------|-----------------------|--------------|---|---|---|---|---|---|---|---|---|---|---|---|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | |
| | | | BUFF EMPTY INT | BUFF FULL IN | STAT CMLPN INT | USB ERR INT | STALL SENT INT | NAK SENT INT | DATAP KTREC INT | DATA PKTTR INT | PING TOKEN INT | IN TOKEN INT | OUT TOKEN INT | SETUP PKT INT | SETUP TOKEN INT | | | | | | | | | | | | | | |
| | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BUFFEMPTYINT | 12 | Флаг опустошения управляющего буфера | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BUFFFULLINT | 11 | Флаг заполнения управляющего буфера | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| STATCMLPNINT | 10 | Флаг успешного завершения стадии «Status» операции на шине USB | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| USBERRINT | 9 | Флаг ошибки при проведении операции | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| STALLSENTINT | 8 | Флаг окончания посылки метки STALL в ответ на метку IN или OUT | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NAKSENTINT | 7 | Флаг окончания посылки метки NAK в ответ на метку IN или OUT | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DATAPKTRECINT | 6 | Флаг успешного приема пакета данных, следующего за меткой OUT, в ответ на который была отправлена метка ACK | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DATAPKTTRINT | 5 | Флаг успешной отправки пакета данных в ответ на метку IN с получением подтверждающей метки ACK | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PINGTOKENINT | 4 | Флаг окончания прием метки PING от хоста | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| INTOKENINT | 3 | Флаг окончания прием метки IN от хоста | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OUTTOKENINT | 2 | Флаг окончания прием метки OUT от хоста | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.5.20

| Поле | Бит | Описание |
|---------------|-------|--|
| SETUPPKTINT | 1 | Флаг окончания приема пакета Setup от хоста. Флаг должен быть сброшен до приема следующего пакета Setup. В противном случае последующие пакеты будут записываться в буфер поверх предыдущих |
| SETUPTOKENINT | 0 | Флаг окончания приема метки SETUP от хоста |
| — | 15-13 | Зарезервировано |

Таблица А.5.21 – Регистр нулевого и первого байтов пакета Setup

| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------|-------------------------------|----|----|----|----|----|-------------|---|---|---|---|---|---|---|---|---|---|-------------|--|--|--|--|--|--|--|-------------|--|--|--|--|--|--|--|---|--|--|--|--|--|--|--|---|--|--|--|--|--|--|--|
| <p>SETUP1_0 8009_0044h Сброс: 0000h</p> <div style="text-align: center;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 2.5%; text-align: center;">15</td><td style="width: 2.5%; text-align: center;">14</td><td style="width: 2.5%; text-align: center;">13</td><td style="width: 2.5%; text-align: center;">12</td><td style="width: 2.5%; text-align: center;">11</td><td style="width: 2.5%; text-align: center;">10</td><td style="width: 2.5%; text-align: center;">9</td><td style="width: 2.5%; text-align: center;">8</td><td style="width: 2.5%; text-align: center;">7</td><td style="width: 2.5%; text-align: center;">6</td><td style="width: 2.5%; text-align: center;">5</td><td style="width: 2.5%; text-align: center;">4</td><td style="width: 2.5%; text-align: center;">3</td><td style="width: 2.5%; text-align: center;">2</td><td style="width: 2.5%; text-align: center;">1</td><td style="width: 2.5%; text-align: center;">0</td> </tr> <tr> <td colspan="8" style="text-align: center;">SETUPPKT_1B</td> <td colspan="8" style="text-align: center;">SETUPPKT_0B</td> </tr> <tr> <td colspan="8" style="text-align: center;">4</td> <td colspan="8" style="text-align: center;">4</td> </tr> </table> </div> | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | SETUPPKT_1B | | | | | | | | SETUPPKT_0B | | | | | | | | 4 | | | | | | | | 4 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SETUPPKT_1B | | | | | | | | SETUPPKT_0B | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SETUPPKT_1B | 15-8 | Старший байт пакета Setup | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 00h Получить статус | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 01h Сбросить параметр | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 02h Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 03h Установить параметр | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 04h Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 05h Установить адрес | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 06h Установить дескриптор | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 07h Получить дескриптор | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 08h Установить конфигурацию | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 09h Получить конфигурацию | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0Ah Установить интерфейс | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0Bh Получить интерфейс | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0Ch Синхрокадр | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SETUPPKT_0B | 7-0 | Младший байт пакета Setup | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 7 | Бит направления | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | От хоста к устройству | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | От устройства к хосту | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 6-5 | Тип | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 00 | Standart | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 01 | Class | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 10 | Vendor | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 4-0 | Получатель | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0h | Устройство | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1h | Интерфейс | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 2h | Буфер | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 3h | Другое | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 4h-1Fh | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.5.22 – Регистры со второго по седьмой байтов пакета Setup

| | | |
|---|------------|--------------------------|
| SETUP3_2 | 8009_0048h | Сброс: 0000h |
| | | |
| SETUP5_4 | 8009_004Ch | Сброс: 0000h |
| | | |
| SETUP7_6 | 8009_0050h | Сброс: 0000h |
| | | |
| Поле | Бит | Описание |
| SETUPPKT_3B, SETUPPKT_5B, SETUPPKT_7B | 15-8 | Старший байт поля wValue |
| SETUPPKT_2B, SETUPPKT_4B, SETUPPKT_6B | 7-0 | Младший байт поля wValue |

Таблица А.5.23 – Регистр управления DMA

| | | |
|---------------------|------------|---|
| DMA_CTRL_STS | 8009_005Ch | Сброс: 0000h |
| | | |
| Поле | Бит | Описание |
| DMAENABLE | 5 | Бит разрешения работы DMA |
| | | 0 Запрещено |
| | | 1 Разрешено |
| DMAR/W | 4 | Бит типа операций DMA |
| | | 0 Чтение |
| | | 1 Запись |
| DMA_EP_ADDR | 3-0 | Поле адреса буфера, к которому обращается DMA |
| — | 15-6 | Зарезервировано |

Таблица А.5.24 – Регистры устройства

| Мнемоника | Адрес | Доступ к регистру | Назначение и описание | Сброс |
|-----------------------|-----------|-------------------|---|-----------|
| CEP_DATA_BUF | 80090028h | Запись Чтение | 16-разрядный буфер данных для передачи | 0000h |
| IN_TRNSFR_CNT | 80090038h | Запись Чтение | 16-разрядный регистр числа байт, которое должно быть отправлено в ответ на метку IN. Если оно больше, чем значение в поле MPS (регистр USB_PKT_FLD_B), то будет отправлено MPS байт | 0000h |
| OUT_TRNSFR_CNT | 8009003Ch | Чтение | 16-разрядный регистр число принятых байт данных | 0000h |
| CEP_START_ADDR | 80090054h | Запись Чтение | 16-разрядный регистр начала адресного пространства, выделенного под управляющий буфер | 0000h |
| CEP_END_ADDR | 80090058h | Чтение | 16-разрядный регистр конца адресного пространства, выделенного под управляющий буфер | 0000h |
| DMA_CNT | 80090060h | Запись Чтение | 16-разрядный Регистр количества байт для операции DMA | 0000h |
| AHB_DMA_ADDR | 80090700h | Запись Чтение | 32-разрядный регистр адреса записи и чтения в режиме DMA | 00000000h |

Таблица А.5.25 – Регистр флагов прерываний не управляющего буфера

| USB_EP_x_IRQ_STAT | | | | | | | | | | | | | | | | 8009_0000h | | Сброс: 0000h | | | | | | | | | |
|--------------------------|-----|--|----|----|----|---|---|---|---|---|---|---|---|---|---|--------------------|---------------------|----------------------|---------------------|----------------------|--------------------|---------------------|-----------------------|----------------------|-----------------------|----------------------|---------------------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | ERR SENT INT | NYET SENT INT | STALL SENT INT | NACK SENT INT | PING TOKE NINT | IN TOKE NINT | OUT TOKE NINT | DATA PKTR ECINT | DATA PKTT RINT | SHOR TPKT TRINT | BUFF EMPT YINT | BUFF FULL INT |
| | | | | | | | | | | | | | | | | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | |
| ERRSENTINT | 11 | Флаг возникновения любой ошибки при проведении операции | | | | | | | | | | | | | | | | | | | | | | | | | |
| NYETSENTINT | 10 | Флаг недостатка свободной памяти для приема следующего пакета данных | | | | | | | | | | | | | | | | | | | | | | | | | |
| STALLSENTINT | 9 | Флаг непринятого последнего пакета. Устанавливается, если последний пакет не мог быть принят, т. к. буфер находился в состоянии "STALL", и была отправлена метка STALL | | | | | | | | | | | | | | | | | | | | | | | | | |
| NACKSENTINT | 8 | Флаг неотправленного последнего пакета. Устанавливается, если последний пакет в ответ на метку IN не мог быть отправлен, и была отправлена метка NAK | | | | | | | | | | | | | | | | | | | | | | | | | |
| PINGTOKENINT | 7 | Флаг приема метки PING от хоста | | | | | | | | | | | | | | | | | | | | | | | | | |
| INTOKENINT | 6 | Флаг приема метки IN от хоста | | | | | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.5.25

| Поле | Бит | Описание |
|---------------|-------|---|
| OUTTOKENINT | 5 | Флаг приема метки OUT от хоста |
| DATAPKTRECINT | 4 | Флаг успешного приема пакета данных от хоста |
| DATAPKTTRINT | 3 | Флаг отправки пакета данных от буфера к хосту |
| SHORTPKTTRINT | 2 | Флаг размера последнего пакета. Устанавливается, если размер последнего пакета был меньше максимального, указанного в регистре EP_MPS |
| BUFFEMPTYINT | 1 | Флаг пустоты буфера. Устанавливается если: - буфер пуст в случае буфера типа IN; - текущий выбранный буфер пуст, или недоступен в случае буфера типа OUT |
| BUFFFULLINT | 0 | Флаг заполнения буфера. Устанавливается если: - текущий выбранный буфер полон, или недоступен в случае буфера типа IN; - буфер полон, в случае буфера типа OUT |
| – | 15-12 | Зарезервировано |

Примечание – Все флаги, за исключением флагов BUFFEMPTYINT и BUFFFULLINT сбрасываются записью единицы в соответствующий бит.

Таблица А.5.26 – Регистр разрешения прерываний не управляющего буфера

| USB_EP_x_IRQ_ENB | | 8009_0000h | Сброс: 0000h | | | | | | | | | | | | |
|------------------|----|------------|--------------|----------------------|-----------------------|------------------------|-----------------------|----------------------|--------------------|---------------------|-----------------------|-----------------------|------------------------|------------------------|-----------------------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | ERR SENT INTEN | NYET SENT INTEN | STALL SENT INTEN | NACK SENT INTEN | PING TKN INTEN | IN TKN INTEN | OUT TKN INTEN | DATA PKTR INTEN | DATA PKTT INTEN | SHRT PKTTR INTEN | BUFF EMPTY INTEN | BUFF FULL INTEN |
| | | | | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 |

Биты с 11 по нулевой являются битами разрешения установки флагов прерываний в регистре EP_IRQ_STAT. Для разрешения прерывания следует записать единицу в соответствующий бит.
Биты с 15 по 12 зарезервированы

Таблица А.5.27 – Регистр числа байт в буфере

| USB_EP_x_AVAIL_CNT | | 8009_0000h | Сброс: 0000h | | | | | | | | | | | | |
|--------------------|----|------------|--------------|---------------|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | BUFF_BYTE_NUM | | | | | | | | | | | |
| | | | | 4 | | | | | | | | | | | |

| Поле | Бит | Описание |
|---------------|-------|---------------------|
| BUFF_BYTE_NUM | 11-0 | Число байт в буфере |
| – | 15-12 | Зарезервировано |

Таблица А.5.28 – Регистр установки и сброса отклика

| USB_EP_x_RSP_SC | | 8009_0000h | Сброс: 0000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|------|---|-------------------------|----|----|----|-------------|------------|-------------------|----------------------|-----------------------|------|---|-------------------|---|---|---|---|---|---|--|--|--|--|--|--|-------------|------------|-------------------|----------------------|-----------------------|------|--|-------------------|--|--|--|--|--|--|--|-----|-----|-----|-----|-----|-----|--|-----|
| <table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:12.5%; text-align:center;">15</td> <td style="width:12.5%; text-align:center;">14</td> <td style="width:12.5%; text-align:center;">13</td> <td style="width:12.5%; text-align:center;">12</td> <td style="width:12.5%; text-align:center;">11</td> <td style="width:12.5%; text-align:center;">10</td> <td style="width:12.5%; text-align:center;">9</td> <td style="width:12.5%; text-align:center;">8</td> <td style="width:12.5%; text-align:center;">7</td> <td style="width:12.5%; text-align:center;">6</td> <td style="width:12.5%; text-align:center;">5</td> <td style="width:12.5%; text-align:center;">4</td> <td style="width:12.5%; text-align:center;">3</td> <td style="width:12.5%; text-align:center;">2</td> <td style="width:12.5%; text-align:center;">1</td> <td style="width:12.5%; text-align:center;">0</td> </tr> <tr> <td colspan="7" style="text-align:center;">-</td> <td style="text-align:center;">NAK SEND</td> <td style="text-align:center;">PKT END</td> <td style="text-align:center;">ZERO LEN IN</td> <td style="text-align:center;">ENDP OINT HALT</td> <td style="text-align:center;">ENDP OINTT OGGL</td> <td colspan="2" style="text-align:center;">MODE</td> <td style="text-align:center;">BUFF FLU SH</td> </tr> <tr> <td colspan="7"></td> <td style="text-align:center;">3 4</td> <td style="text-align:center;">3 4</td> <td style="text-align:center;">3 4</td> <td style="text-align:center;">3 4</td> <td style="text-align:center;">3 4</td> <td colspan="2" style="text-align:center;">3 4</td> <td style="text-align:center;">3 4</td> </tr> </table> | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | - | | | | | | | NAK SEND | PKT END | ZERO LEN IN | ENDP OINT HALT | ENDP OINTT OGGL | MODE | | BUFF FLU SH | | | | | | | | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | | 3 4 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | NAK SEND | PKT END | ZERO LEN IN | ENDP OINT HALT | ENDP OINTT OGGL | MODE | | BUFF FLU SH | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | | 3 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NAKSEND | 7 | Бит ответа в режиме работы OUT конечной точки | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Ответ NAK не посылается | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | Ответ NAK посылается | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PKTEND | 6 | Индикатор числа байт В режиме Auto-Validate устанавливается, если по окончании передачи в буфере осталось число байт, не равное величине MPS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ZEROLENIN | 5 | Бит выбора ответа на метку IN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Не отправляет пакет | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | В ответ на метку IN отправляется пакет нулевой длины | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ENDPOINT HALT | 4 | Бит отправки Stall HandShake на любой токен от хоста | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Не отправляет ответ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | В ответ на любую метку от хоста отправляется метка STALL_Handshake. Бит должен быть установлен при приеме команды «Set Feature (ep_halt)» от хоста | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ENDPOINT TOGGLE | 3 | Бит управления битом DATATOGGLE. Установка бита сбрасывает бит DATATOGGLE регистра USB_PKT_FLD_A. Чтение бита возвращает текущее состояние бита DATATOGGLE. Флаг сбрасывается записью единицы | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MODE | 2-1 | Поле режима функционирования буфера типа IN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 00 | Auto-Validate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 01 | Manual-Validate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 10 | Fly | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BUFFFUSH | 0 | Бит сброса содержимого буфера и регистра EP_AVAIL_CNT. Установка бита активирует сброс | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| – | 15-7 | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.5.29 – Регистр числа байт буфера типа IN

| USB_EP_x_CNT | | 8009_0000h | Сброс: 0000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|------|---|--------------|----|----|----|-----------------|----|----|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|-----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-----|--|--|--|--|--|--|--|
| <table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:12.5%; text-align:center;">15</td> <td style="width:12.5%; text-align:center;">14</td> <td style="width:12.5%; text-align:center;">13</td> <td style="width:12.5%; text-align:center;">12</td> <td style="width:12.5%; text-align:center;">11</td> <td style="width:12.5%; text-align:center;">10</td> <td style="width:12.5%; text-align:center;">9</td> <td style="width:12.5%; text-align:center;">8</td> <td style="width:12.5%; text-align:center;">7</td> <td style="width:12.5%; text-align:center;">6</td> <td style="width:12.5%; text-align:center;">5</td> <td style="width:12.5%; text-align:center;">4</td> <td style="width:12.5%; text-align:center;">3</td> <td style="width:12.5%; text-align:center;">2</td> <td style="width:12.5%; text-align:center;">1</td> <td style="width:12.5%; text-align:center;">0</td> </tr> <tr> <td colspan="7" style="text-align:center;">-</td> <td colspan="8" style="text-align:center;">BYTEVAL_TO_HOST</td> </tr> <tr> <td colspan="7"></td> <td colspan="8" style="text-align:center;">3 4</td> </tr> </table> | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | - | | | | | | | BYTEVAL_TO_HOST | | | | | | | | | | | | | | | 3 4 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | BYTEVAL_TO_HOST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | 3 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BYTEVAL_ TO_HOST | 7-0 | Поле числа байт, которое необходимо послать хосту в режиме Manual-Validate. Действует только для буфера типа IN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| – | 15-8 | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.5.30 – Регистр конфигурации не управляющего буфера

| USB_EP_x_CFG | | 8009_0000h | Сброс: 0000h |
|---------------|-------|--|-------------------|
| | | | |
| Поле | Бит | Описание | |
| MULT | 9-8 | Поле числа операций, выполняемых в одном микрокадре | |
| | | 00 | Одна |
| | | 01 | Две |
| | | 10 | Три |
| | | 11 | Неверное значение |
| BUFFNUM | 7-4 | Поле номера буфера. Допустимые значения от 1h до Fh | |
| BUFFDIR | 3 | Поле типа (направления) буфера | |
| | | 0 | OUT |
| | | 1 | IN |
| BUFFTYPE | 2-1 | Тип буфера | |
| | | 00 | Зарезервировано |
| | | 01 | Bulk |
| | | 10 | Interrupt |
| | | 11 | Isochronous |
| ENDPOINTVALID | 0 | Разрешение работы буфера. | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| – | 15-10 | Зарезервировано | |

Таблица А5.31 – Регистры EP

| Мнемоника | Доступ к регистру | Назначение | Сброс |
|----------------------------|-------------------|---|-------|
| USB_EP_x_DATA_BUF | Запись Чтение | 16-разрядный регистр обмена данными с буфером | 0000h |
| USB_EP_x_MPS | Запись Чтение | 16-разрядный регистр максимального размера пакета | 0000h |
| USB_EP_x_START_ADDR | Запись Чтение | 16-разрядный регистр начала адресного пространства, выделенного под буфер | – |
| USB_EP_x_END_ADDR | Запись Чтение | 16-разрядный регистр конца адресного пространства, выделенного под буфер | – |

Таблица А.5.32 – Регистр состояния

| OTG_IRQ_STAT | | 8009_0704h | Сброс: 00000000h | | | | | | | | | | | | |
|---|-----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| <div style="display: flex; justify-content: space-between; font-size: small;"> 31 30 29 28 27 26 25 24 23 22 21 DIS CON NECT HOST SYNC CHNG DEV SYNC CHNG BDEV SYNC CHNG ADEV SYNC CHNG </div> <div style="display: flex; justify-content: center; margin-top: 5px;"> - </div> <div style="display: flex; justify-content: center; font-size: x-small;"> 3 4 3 4 3 4 3 4 3 4 </div> | | | | | | | | | | | | | | | |
| <div style="display: flex; justify-content: space-between; font-size: small;"> 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SUSP EN SESS ION FAIL B_DEV ICE A_DEV ICE ASUS PEND REQ B_BUS REQ B_HNP EN A_HNP EN A_BUS DROP A_BUS REQ HOST DEV ICE SRP FAIL HAND OFF INTR SRP DET INTR VBUS ERR INTR </div> <div style="display: flex; justify-content: center; font-size: x-small;"> 3 4 3 4 4 4 3 4 3 4 3 4 3 4 3 4 3 4 4 4 3 4 3 4 3 4 3 4 </div> | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| DISCONNECT | 20 | Устанавливается при разрыве связи не в приостановленном режиме | | | | | | | | | | | | | |
| HOSTSYNCCHNG | 19 | Устанавливается при ресинхронизации хоста | | | | | | | | | | | | | |
| DEV_SYNCCHNG | 18 | Устанавливается при ресинхронизации устройства | | | | | | | | | | | | | |
| BDEVSYNCCHNG | 17 | Устанавливается когда происходит смена устройства В | | | | | | | | | | | | | |
| ADEVSYNCCHNG | 16 | Устанавливается когда происходит смена устройства А | | | | | | | | | | | | | |
| SUSPEND_EN | 15 | Запрос приостановки работы UTMI PHY по линии USB | | | | | | | | | | | | | |
| | | 0 | переход UTMI PHY в нормальный режим | | | | | | | | | | | | |
| | | 1 | переход UTMI PHY в режим приостановки | | | | | | | | | | | | |
| SESSION_FAIL | 14 | Бит устанавливается устройством В, когда устройство А обрывает сигнал Vbus | | | | | | | | | | | | | |
| B_DEVICE | 13 | Индикатор подключений устройства В к разъему | | | | | | | | | | | | | |
| | | 0 | Не подключено | | | | | | | | | | | | |
| | | 1 | Подключено | | | | | | | | | | | | |
| A_DEVICE | 12 | Индикатор подключений устройства А к разъему | | | | | | | | | | | | | |
| | | 0 | Не подключено | | | | | | | | | | | | |
| | | 1 | Подключено | | | | | | | | | | | | |
| ASUSPEND_REQ | 11 | Аппаратный сброс UTMI PHY и связанного с ним входного FIFO буфера. Запись логической 1 приводит к инициализации сброса. Бит сбрасывается сам после установки | | | | | | | | | | | | | |
| B_BUS_REQ | 10 | Запрос управления шины устройством В | | | | | | | | | | | | | |
| | | 0 | Шина не используется приложением устройства В | | | | | | | | | | | | |
| | | 1 | Запрос использования шины от приложения устройства В | | | | | | | | | | | | |
| B_HNP_EN | 9 | Бит разрешения выполнения HNP устройством А | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | |
| | | 1 | Разрешение выполнения HNP устройством А. | | | | | | | | | | | | |
| A_HNP_EN | 8 | Бит разрешения выполнения HNP устройством А. Бит устанавливается только если устройство А подключено напрямую к устройству В, поддерживающему HNP | | | | | | | | | | | | | |
| A_BUS_DROP | 7 | Бит запроса от устройства А на отключение питания шины | | | | | | | | | | | | | |
| | | 0 | Нет запроса от устройства А на отключение питания шины | | | | | | | | | | | | |
| | | 1 | Запрос от устройства А на отключение питания шины | | | | | | | | | | | | |
| A_BUS_REQ | 6 | Бит запроса на управление шиной устройством А | | | | | | | | | | | | | |
| | | 0 | Приложение устройства А не использует шину | | | | | | | | | | | | |
| | | 1 | Приложение устройства А использует шину | | | | | | | | | | | | |

Окончание таблицы А.5.32

| Поле | Бит | Описание |
|--|-------|---|
| HOST | 5 | Индикатор работы в режиме хоста (A-device) |
| | | 0 Не работает как устройство А 1 Контроллер работает как устройство А |
| DEVICE | 4 | Индикатор работы в режиме устройства (B-device) |
| | | 0 Не работает как устройство В 1 Контроллер работает как устройство В |
| SRP_FAIL | 3 | Индикатор отсутствия ответа от хоста на SRP запрос от устройства В. Устанавливается при работе в качестве устройства В |
| HANDOFF_INTR | 2 | Флаг прерывания при изменении состояния вывода ID |
| SRPDETINTR | 1 | Разрешение прерывания srp_detected |
| VBUSERRINTR | 0 | Разрешение прерывания VBUS_err |
| – | 31-21 | Зарезервировано |
| <p>Примечания</p> <p>1 При установке битов 20 – 16, 14 и 3 – 0 генерируются прерывания, если это разрешено соответствующими битами регистра OTG_IRQ_EN.</p> <p>2 Биты с 20 по 14, а также биты 3 и 2 сбрасываются записью единицы.</p> | | |

Таблица А.5.33 – Регистр разрешения прерываний

| Поле | | Бит | Описание |
|---|--|-------|--|
| DISCONINTEN | | 9 | Разрешение прерывания disconnect |
| HOSTSYNINTEN | | 8 | Разрешение прерывания host_sync_change |
| DEVSYNINTEN | | 7 | Разрешение прерывания device_sync_change |
| BDEVINTEN | | 6 | Разрешение прерывания b_device_change |
| ADEVINTEN | | 5 | Разрешение прерывания a_device_change |
| SESSFAILINTEN | | 4 | Разрешение прерывания session_fail |
| SRP_FAILINTEN | | 3 | Разрешение прерывания srf_fail |
| TRANSITINTEN | | 2 | Разрешение прерывания handoff_intr |
| SRPDETINTEN | | 1 | Разрешение прерывания srp_detected |
| VBUSERRINTEN | | 0 | Разрешение прерывания VBUS_err |
| – | | 31-10 | Зарезервировано |
| <p>Примечание – Для разрешения прерывания следует установить соответствующий бит.</p> | | | |

А.6 Регистры RTC

Таблица А.6.1 – Регистр долей секунд

| PSECONDS | | A002_3000h | Сброс: 0000000h | | | | | | | | | | | | |
|-----------------|-------|---|-----------------|----|----|---|---|---|----------|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | PART_SEC | | | | | | |
| 3 Ч | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| PART_SEC | 9-0 | Доля секунды (Двоичный формат). Корректные значения: 000h – 3FFh | | | | | | | | | | | | | |
| – | 15-10 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.6.2 – Регистр секунд

| SECOND | | A002_3004h | Сброс: 0000000h | | | | | | | | | | | | |
|---------------|------|---|-----------------|----|----|---|---|---|--------|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | SECOND | | | | | | |
| 3 Ч | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| SECOND | 6-0 | Секунда (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 29h, 30h – 39h, 40h – 49h, 50h – 59h | | | | | | | | | | | | | |
| – | 15-7 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.6.3 – Регистр минут

| MINUTE | | A002_3008h | Сброс: 0000000h | | | | | | | | | | | | |
|---------------|------|--|-----------------|----|----|---|---|---|--------|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | MINUTE | | | | | | |
| 3 Ч | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| MINUTE | 6-0 | Минута (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 29h, 30h – 39h, 40h – 49h, 50h – 59h | | | | | | | | | | | | | |
| – | 15-7 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.6.4 – Регистр часов

| HOUR | | A002_3010h | Сброс: 0000000h |
|-------------|------|---|-----------------|
| | | 3 ч | |
| Поле | Биты | Описание | |
| HOUR | 5-0 | Час (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 23h | |
| – | 15-6 | Зарезервировано | |

Таблица А.6.5 – Регистр дней недели

| WDAY | | A002_3018h | Сброс: 0000001h |
|-------------|------|---|-----------------|
| | | 3 ч | |
| Поле | Биты | Описание | |
| DAYWEEK | 2-0 | День недели (BCD формат). Корректные значения: 00h – 07h | |
| – | 15-3 | Зарезервировано | |

Таблица А.6.6 – Регистр чисел

| DAY | | A002_3020h | Сброс: 0000022h |
|------------|------|--|-----------------|
| | | 3 ч | |
| Поле | Биты | Описание | |
| DAY | 5-0 | Число (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 29h, 30h – 31h | |
| – | 15-6 | Зарезервировано | |

Таблица А.6.7 – Регистр месяцев

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|----|------|----|--|----|---|---|---|---|---|---|---|---|---|------------|---|--|--|--|-----------------|-------|--|--|--|--|
| MONTH | | | | | | | | | | | | | | | A002_3024h | | | | | Сброс: 0000011h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | - | | | | | MONTH | | | | |
| | | | | | | | | | | | | | | | | | | | | 3 ч | | | | | |
| Поле | | Биты | | Описание | | | | | | | | | | | | | | | | | | | | | |
| MONTH | | 4-0 | | Месяц (BCD формат). Корректные значения: 00h – 09h, 10h – 12h | | | | | | | | | | | | | | | | | | | | | |
| – | | 15-5 | | Зарезервировано | | | | | | | | | | | | | | | | | | | | | |

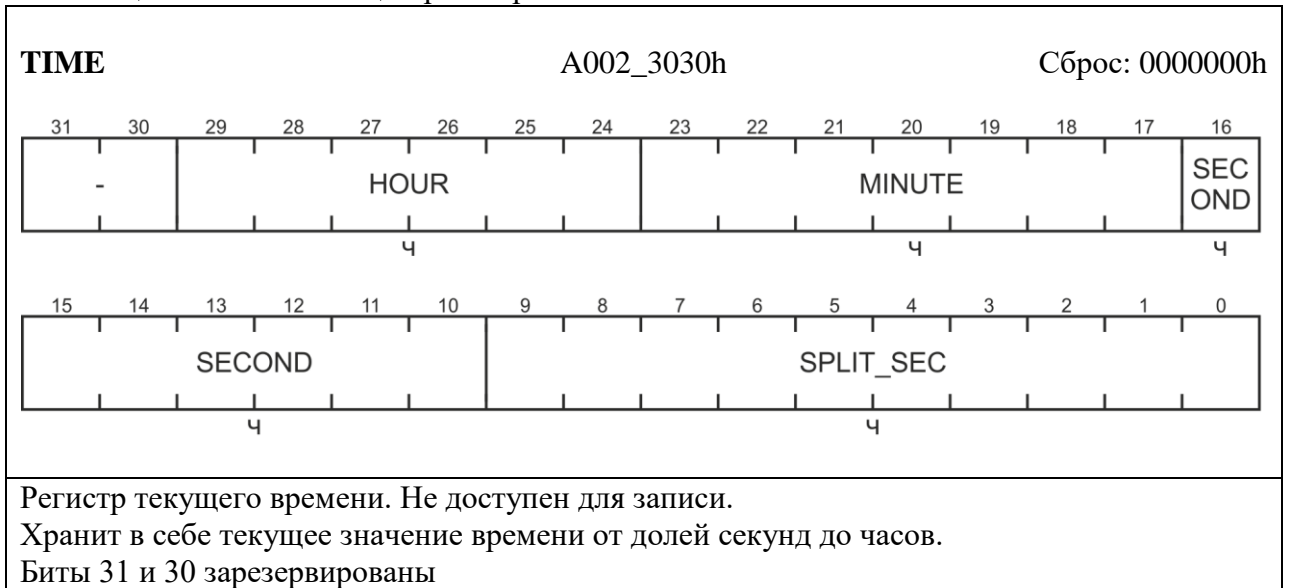
Таблица А.6.8 – Регистр лет

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------|----|------|----|---|----|---|---|---|---|---|---|---|---|---|------------|---|--|--|--|-----------------|------|--|--|--|--|
| YEAR | | | | | | | | | | | | | | | A002_3028h | | | | | Сброс: 0000077h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | - | | | | | YEAR | | | | |
| | | | | | | | | | | | | | | | | | | | | 3 ч | | | | | |
| Поле | | Биты | | Описание | | | | | | | | | | | | | | | | | | | | | |
| YEAR | | 7-0 | | Год (BCD формат). Корректные значения: 00h – 09h, 10h – 19h, 20h – 29h, 30h – 39h, 40h – 49h, 50h – 59h, 60h – 69h, 70h – 79h, 80h – 89h, 90h – 99h | | | | | | | | | | | | | | | | | | | | | |
| – | | 15-8 | | Зарезервировано | | | | | | | | | | | | | | | | | | | | | |

Таблица А.6.9 – Общий регистр

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------|----|--------------|----|--|----|----|----|----|----|----|----|----|----|----|------------|---|--|--|--|-----------------|--|--|--|--|--|
| SHDW | | | | | | | | | | | | | | | A002_302Ch | | | | | Сброс: 0000080h | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | - | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | 3 | | | | | |
| Поле | | Бит | | Описание | | | | | | | | | | | | | | | | | | | | | |
| UPDTEN | | 7 | | Бит разрешения обновлений теневого регистра. По умолчанию равен единице | | | | | | | | | | | | | | | | | | | | | |
| – | | 31-8, 6-0 | | Зарезервировано | | | | | | | | | | | | | | | | | | | | | |

Таблица А.6.10 – Общий регистр



А.7 Регистры сторожевого таймера

Таблица А7.1 – Регистры

| Мнемоника | Адрес регистра | Доступ к регистру | Назначение и описание | Сброс |
|---------------|----------------|-------------------|--|-----------|
| LOAD | A0000000h | Запись Чтение | 32-разрядный регистр, хранящий начальное значение счетчика. Когда происходит запись в этот регистр, счетчик сразу иницируется этим новым значением. Минимальное допустимое значение 0000_0001h | FFFFFFFFh |
| VALUE | A0000004h | Чтение | 32-разрядный регистр текущего значения счетчика | FFFFFFFFh |
| INTCLR | A000000Ch | Запись | 32-разрядный регистр сброса сторожевого таймера. Запись любого значения в этот регистр приводит к сбросу прерывания сторожевого таймера и загрузке счетчика значением из регистра LOAD | 00000000h |

Таблица А.7.2 – Регистр управления

| CTRL | | A000_0008h | Сброс: 00000000h |
|--|------|--|------------------|
| <div style="display: flex; justify-content: space-between;"> 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 </div> <div style="border: 1px solid black; height: 40px; width: 100%;"></div> | | | |
| <div style="display: flex; justify-content: space-between;"> 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 </div> <div style="border: 1px solid black; height: 40px; width: 100%;"></div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">RES EN</div> <div style="border: 1px solid black; padding: 2px;">INT EN</div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 4 3 4 </div> | | | |
| RESEN | 1 | Бит разрешения сброса микроконтроллера по сторожевому таймеру. Работает по функции «Логическое И» с битом INTEN регистра WDTCTRL 0 Сброс бита выключает сброс 1 Установка включает сброс | |
| INTEN | 0 | Бит включения счета и разрешения прерывания сторожевого таймера 0 Сброс бита выключает счетчик и снимает прерывание 1 Установка бита включает счетчик и генерирует прерывание. Если счетчик был включен на момент установки бита, то он иницируется значением из регистра LOAD | |
| – | 31-2 | Зарезервировано | |

Таблица А.7.3 – Регистр прерывания

| RIS | | A000_0010h | | | | | | | | | | | | | | | | Сброс: 00000000h | |
|------------|------|--|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|---|
| | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | |
| | | - | | | | | | | | | | | | | | | | | |
| | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | RAW WDT INT | Ч |
| | | - | | | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | |
| RAWWDTINT | 0 | Индикатор состояния немаскированного бита прерывания | | | | | | | | | | | | | | | | | |
| | | 0 | Сброшен | | | | | | | | | | | | | | | | |
| | | 1 | Установлен | | | | | | | | | | | | | | | | |
| – | 31-1 | Зарезервировано | | | | | | | | | | | | | | | | | |

Таблица А.7.4 – Регистр маскированного прерывания

| MIS | | A000_0014h | | | | | | | | | | | | | | | | Сброс: 00000000h | |
|------------|------|---|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|---|
| | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | |
| | | - | | | | | | | | | | | | | | | | | |
| | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | WDT INT | Ч |
| | | - | | | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | |
| WDTINT | 0 | Индикатор состояния маскированного бита прерывания. Сигнализирует о появлении маскированного прерывания от счетчика. Состояние бита WDTINT это «логическое И» битов RAWWDTINT и INTEN | | | | | | | | | | | | | | | | | |
| | | 0 | Сброшен | | | | | | | | | | | | | | | | |
| | | 1 | Установлен | | | | | | | | | | | | | | | | |
| – | 31-1 | Зарезервировано | | | | | | | | | | | | | | | | | |

Таблица А.7.5 – Регистр блокировки

| Поле | Бит | Описание |
|------------|------|--|
| REG_WR_DIS | 0 | Бит запрета записи во все регистры сторожевого таймера (кроме LOCK). Функция необходима для предотвращения отключения сторожевого таймера сбойными программами |
| | 0 | Разрешена (по умолчанию). Для сброса бита следует записать в регистр WDTLOCK значение 1ACCE551h |
| | 1 | Запрещена. Для установки бита следует записать в регистр WDTLOCK любое значение, кроме 1ACCE551h |
| – | 31-1 | Зарезервировано |

А.8 Регистры контроллера I2C

Таблица А.8.1 – Сдвиговый регистр данных

| SDA | | | | | | | | | | | | | | | Сброс: 00XXh | |
|------------|------|-----------------|----|----|----|---|---|------|---|---|---|---|---|---|--------------|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| - | | | | | | | | DATA | | | | | | | | |
| | | | | | | | | 3 ч | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | |
| DATA | 7-0 | Поле данных | | | | | | | | | | | | | | |
| – | 15-8 | Зарезервировано | | | | | | | | | | | | | | |

Таблица А.8.2 – Регистр состояния

| ST | | | | | | | | | | | | | | | Сброс: 0000h | | | |
|-----------|-----|--|----|----|----|---|---|-----|---|------|---|---|---|---|--------------|--|--|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| - | | | | | | | | INT | - | MODE | | | | | | | | |
| | | | | | | | | 4 | | | | 4 | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | |
| INT | 7 | <p>Флаг прерывания.</p> <p>Устанавливается после девятого такта сигнала SCL (когда SCL = 0) в любое запрограммированное время. Условия выставления флага INT:</p> <ul style="list-style-type: none"> - во время приема/передачи, как в режиме мастера, так и в режиме ведомого; - при совпадении адреса (адреса ведомого, адреса отклика или адреса общего вызова) содержимое регистра SDA должно контролироваться программно для определения типа полученного адреса; - после успешного формирования стартового состояния или состояния повторного старта; - в случае не квитирования переданной информации; - при потере арбитража во время передачи последнего бита; - при обнаружении валидного состояния останова или состояния повторного старта; - при обнаружении ошибки на шине. <p>Пока установлен флаг INT, на линии SCL удерживается низкий уровень сигнала.</p> <p>Флаг INT может быть сброшен установкой бита CLRST в регистре CTL0 или выключением модуля I2C (обнуление бита ENABLE в регистре CTL1).</p> <p>Условия выставления флага INT (не влияющие на уровень сигнала на линии SCL):</p> <ul style="list-style-type: none"> - простой на линии SCL; - состояние останова в режиме ведомого (MODE = 1Ch); - потеря арбитража, вследствие чего ведомый переключился в безадресный режим (MODE = 03h или MODE = 23h); - не квитированная передача байта данных (MODE = 17h) | | | | | | | | | | | | | | | | |

Окончание таблицы А.8.2

| Поле | Бит | Описание |
|------|------------|---|
| MODE | 5-0 | Код состояния. Возникновение того или иного состояния в течение функционирования модуля I2C сопровождается записью соответствующего кода в поле MODE |
| – | 15-8, 6 | Зарезервировано |

Таблица А.8.3 – Регистр управления и статуса

| CST | | Сброс: 0000h |
|----------|-----|---|
| | | |
| Поле | Бит | Описание |
| PECFAULT | 7 | Флаг ошибки. Устанавливается в случае, если после расчета контрольной суммы для пакета данных и сравнения ее с полученной суммой, значение во внутреннем регистре ошибок не нулевое |
| PECNEXT | 6 | Бит управления отправкой байта контрольной суммы. Установка бита указывает на то, что следующий передаваемый байт будет байтом CRC (байт контрольной суммы). Реакция на установку бита PECNEXT зависит от режима работы. В режиме мастера передатчика установка бита PECNEXT вызовет загрузку результата вычисления CRC в регистр SDA. После сброса флага INT начнется передача байта CRC. В режиме приемника установка этого бита будет указывать логике управления на то, что следующий байт, который будет принят, будет байтом CRC. В режиме ведомого приемника модуль I2C автоматически будет квитировать или не квитировать прием байта CRC, в зависимости от того, будет ли выявлена ошибка пакета данных или нет. В режиме мастера приемника по окончании приема байта CRC, будет отправлено значение бита ACK регистра CTL0 |
| TGSCL | 5 | Бит переключения SCL. Бит позволяет переключать вывод SCL во время восстановления после ошибки. Когда на выводе SDA – низкий уровень сигнала, запись «1» в бит TGSCL переключит вывод SCL на один такт. Когда на SDA высокий уровень сигнала, запись «1» в бит TGSCL игнорируется. Бит очищается аппаратно по окончании такта |
| TSDA | 4 | Бит тестирования SDA. Содержит текущее значение SDA. Этот бит можно использовать для отслеживания окончания процесса восстановления после ошибки, в течение которого ведомый постоянно поддерживает низкий уровень сигнала на выводе SDA |

Окончание таблицы А.8.3

| Поле | Бит | Описание |
|--------|------------------|---|
| TOERR | 3 | Флаг ошибки простоя на шине. Если TOERR = 1b, это указывает на то, что на линии SCL был обнаружен простой. Флаг TOERR выставляется по обнулению основного счетчика времени простоя и может быть сброшен записью «1» в бит CLRST регистра CTLO |
| TOCDIV | 2-1 | Поле коэффициента делителя. Устанавливает коэффициент деления системного тактового сигнала, подаваемого на предделитель времени простоя линии SCL |
| | | 00 Тактовый сигнал отсутствует |
| | | 01 Деление на 4 |
| | | 10 Деление на 8 |
| | 11 Деление на 16 | |
| VB | 0 | Флаг занятости шины. Если VB = 1b, это указывает на то, что шина занята. Устанавливается, как только шина переходит в активное состояние (одновременное появление низкого уровня сигнала на выводах SDA и SCL или хотя бы на одном из них) или в стартовое состояние. Сбрасывается при выключении интерфейса I2C, либо при обнаружении состояния останова |
| – | 15-8 | Зарезервировано |

Таблица А.8.4 – Регистр управления 0

| Поле | Бит | Описание |
|---|-----|---|
| CTLO | | Сброс: 0000h |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> <div style="width: 10%;"></div> <div style="width: 10%;"></div> <div style="width: 10%;"></div> <div style="width: 10%; text-align: center;">-</div> <div style="width: 10%;"></div> <div style="width: 10%;"></div> <div style="width: 10%; text-align: center;">CLR ST</div> <div style="width: 10%; text-align: center;">SMB ARE</div> <div style="width: 10%; text-align: center;">GCM EN</div> <div style="width: 10%; text-align: center;">ACK</div> <div style="width: 10%; text-align: center;">-</div> <div style="width: 10%; text-align: center;">INT EN</div> <div style="width: 10%; text-align: center;">STOP</div> <div style="width: 10%; text-align: center;">STA RT</div> </div> <div style="display: flex; justify-content: space-between; padding-top: 5px;"> <div style="width: 10%;"></div> <div style="width: 10%;"></div> <div style="width: 10%;"></div> <div style="width: 10%;"></div> <div style="width: 10%;"></div> <div style="width: 10%;"></div> <div style="width: 10%; text-align: center;">3</div> <div style="width: 10%; text-align: center;">3 4</div> <div style="width: 10%; text-align: center;">3 4</div> <div style="width: 10%; text-align: center;">3 4</div> <div style="width: 10%;"></div> <div style="width: 10%; text-align: center;">3 4</div> <div style="width: 10%; text-align: center;">3 4</div> <div style="width: 10%; text-align: center;">3 4</div> </div> | | |
| CLRST | 7 | Бит сброса флага прерывания INT. Запись «0» в бит CLR игнорируется. Запись «1» в бит CLR сбросит флаг INT в регистре ST. Чтение этого бита всегда возвращает «0» |
| SMBARE | 6 | Бит управления реакцией на получение адреса отклика |
| | | 0 Полученный адрес не проверяется на совпадение с адресом отклика |
| | | 1 Адрес, полученный сразу после старта, проверяется на совпадение с адресом отклика (0001_100b) |
| | | Бит очищается при выходе ведомого из режима IDLE |
| GCMEN | 5 | Бит управления реакцией на получение адреса общего вызова |
| | | 0 Полученный адрес не проверяется на совпадение с адресом общего вызова |
| | | 1 Адрес, полученный сразу после старта, проверяется на совпадение с адресом общего вызова (0000_000b) |
| | | Бит очищается при выходе ведомого из режима IDLE |

Окончание таблицы А.8.4

| Поле | Бит | Описание |
|-------|------------|--|
| АСК | 4 | Бит квитирования приема. В режиме передатчика не используется. В режиме приемника (мастера/ведомого) содержит значение, которое передается в течение цикла отклика на запрос передатчика подтвердить прием. Передача нуля по окончании передачи байта (квитирование) означает, что данные успешно получены. Передача единицы (неквитирование) означает, что приемник не может продолжать работу по каким-либо причинам. Бит АСК очищается аппаратно по окончании цикла отклика |
| INTEN | 2 | Бит разрешения прерывания |
| | | 0 Запрещено |
| | | 1 Разрешено |
| STOP | 1 | Бит останова. В режиме мастера установка бита STOP генерирует состояние останова, которое завершает или прерывает текущую передачу. После прекращения передачи бит STOP очищается аппаратно |
| START | 0 | Бит старта. Этот бит устанавливается, когда требуется сформировать стартовое состояние на шине. Бит START очищается аппаратно по окончании цикла стартового состояния, а также при обнаружении ошибки на шине (состояние с кодом 1Fh) |
| – | 15-8, 3 | Зарезервировано |

Таблица А.8.5 – Регистр собственного адреса

| Поле | Бит | Описание |
|-------------|------|---|
| ADDR | | Сброс: 0000h |
| | | |
| SAEN | 7 | Бит разрешения распознавания адреса |
| | | 0 Безадресный режим |
| | | 1 Включена функция распознавания принятого адреса |
| ADDR | 6–0 | Поле собственного 7-битного адреса. При работе в режиме ведомого первые 7 бит, принятые после стартового состояния, сравниваются со значением ADDR. Если обнаружено совпадение и установлен бит SAEN, ведомый переходит в режим приемника или передатчика (в зависимости от состояния бита направления R/W#) |
| – | 15-8 | Зарезервировано |

Таблица А.8.6 – Регистр управления 1

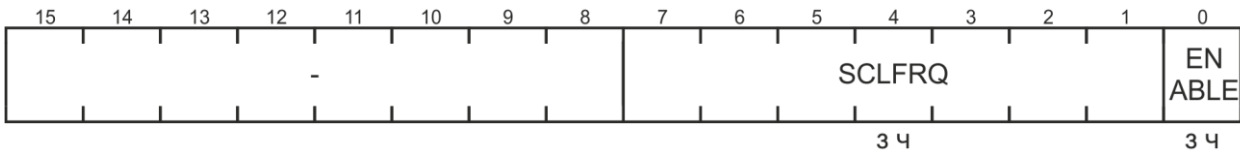
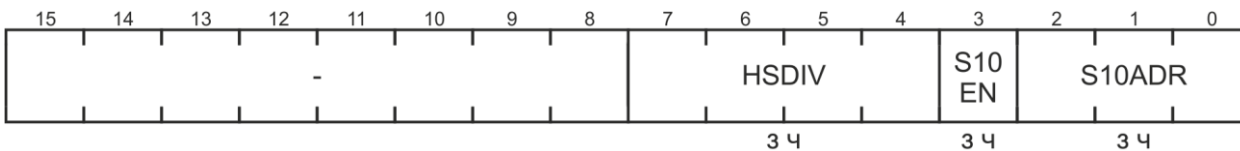
| CTL1 | | Сброс: 0000h | |
|--|------|--|--|
|  | | | |
| Поле | Биты | Описание | |
| SCLFRQ | 7-1 | <p>Поле выбора частоты f_{SCL} сигнала на выводе SCL в режиме мастера. Длительности высокого (T_{SCLH}) и низкого (T_{SCLL}) уровней сигнала SCL зависят от тактовой частоты f_{osc} модуля I2C и рассчитываются по формуле</p> $T_{SCLH} = T_{SCLL} = 2 \times SCLFRQ \times (1/f_{osc}).$ <p>Таким образом, частота сигнала на выводе SCL равна</p> $f_{SCL} = 1/(T_{SCLH} + T_{SCLL}).$ <p>В поле SCLFRQ можно записать любое значение в диапазоне от 04h до 7Fh. При попытке записи любого значения меньше «04h», оно будет записано со смещением 04h. Например, при записи числа 02h, к нему будет аппаратно добавлено смещение 04h и, в итоге, в поле SCLFRQ окажется значение «06h»</p> | |
| ENABLE | 0 | 0 | Модуль выключен. Тактирование не осуществляется. Регистры CTL0, ST, CST сброшены |
| | | 1 | Модуль включен |
| – | 15-8 | Зарезервировано | |

Таблица А.8.7 – Регистр загрузки предделителя

| TOPR | | Сброс: 0000h |
|--|------|---|
|  | | |
| Поле | Биты | Описание |
| SMBTOPR | 7-0 | Поле значения перезагрузки предделителя |
| – | 15-8 | Зарезервировано |

Таблица А.8.8 – Регистр управления 2

| CTL2 | | Сброс: 0000h |
|--|------|--------------|
|  | | |
| Поле | Биты | Описание |
| HSDIV | 6-4 | 3 бита |
| S10 EN | 3 | 1 бит |
| S10ADR | 2-0 | 3 бита |

Окончание таблицы А.8.8

| Поле | Биты | Описание | |
|--------|------|---|--|
| HSDIV | 7-4 | <p>Поле выбора частоты f_{SCL} сигнала на выводе SCL в режиме HS мастера. Длительности высокого (T_{HSCLH}) и низкого (T_{HSCLL}) уровней сигнала на выводе SCL зависят от тактовой частоты f_{osc} модуля I2C и рассчитываются по формулам</p> $T_{HSCLH} = HSDIV \times (1/f_{osc}),$ $T_{HSCLL} = 2 \times HSDIV \times (1/f_{osc}).$ <p>Таким образом, частота сигнала на выводе SCL равна</p> $f_{SCL} = 1/(T_{HSCLH} + T_{HSCLL}).$ <p>В поле HSDIV можно записать любое значение в диапазоне от 2h до Fh. При попытке записи любого значения меньше «2h» в поле HSDIV, оно будет записано со смещением 2h. Например, при записи числа 1h к нему будет аппаратно добавлено смещение 2h и, в итоге, в поле SCLFRQ окажется значение «3h»</p> | |
| S10EN | 3 | Бит разрешения 10-битной адресации ведомого | |
| | | 0 | Запрещена |
| | | 1 | Разрешена при условии, что установлен бит SAEN в регистре ADDR |
| S10ADR | 2-0 | <p>Поле старших битов 10-битного адреса ведомого. Поле содержит старшие три разряда адреса ведомого при 10-битной адресации. Первый принятый байт адреса сравнивается со значением [11110b, S10ADR[2:1]], второй байт адреса – со значением [S10ADR[0], ADDR]</p> | |
| – | 15-8 | Зарезервировано | |

А.9 Регистры таймера

Таблица А.9.1 – Регистр управления блока таймера

| CTRL | | Сброс: 00000000h |
|--|------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 25%; height: 20px;"></div> <div style="border-right: 1px solid black; width: 10%; text-align: center;">INT EN</div> <div style="border-right: 1px solid black; width: 10%; text-align: center;">EXTI NPUT CLK</div> <div style="border-right: 1px solid black; width: 10%; text-align: center;">EXIN TPUT EN</div> <div style="width: 10%; text-align: center;">ON</div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 4 3 4 3 4 3 4 </div> | | |
| Поле | Бит | Описание |
| INTEN | 3 | Бит разрешения прерывания таймера |
| | | 0 Запрещено |
| | | 1 Разрешено |
| EXTINPUT_ CLK | 2 | Бит включения внешнего входа синхронизации как тактового |
| | | 0 Нет действий |
| | | 1 Сигнал на входе EX TIN является тактовым |
| EXTINPUT_ EN | 1 | Бит разрешения работы таймера, если сигнал на соответствующем входе равен единице |
| | | 0 Запрещено |
| | | 1 Таймер декрементируется с частотой синхросигнала микроконтроллера |
| ON | 0 | Бит включения таймера |
| | | 0 Выключен |
| | | 1 Таймер декрементируется с частотой синхросигнала микроконтроллера |
| – | 31-4 | Зарезервировано |

Таблица А.9.2 – Регистр прерывания таймера

| INTSTATUS_INTCLEAR | | Сброс: 0000h |
|---|------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 95%; height: 20px;"></div> <div style="width: 5%; text-align: center;">INT</div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 4 </div> | | |
| Поле | Бит | Описание |
| INT | 0 | Флаг прерывания таймера |
| | | 0 Нет прерывания |
| | | 1 Запрос на прерывание |
| | | Для сброса бита следует записать в него единицу |
| – | 31-1 | Зарезервировано |

Таблица А.9.3 – Регистры

| Мнемоника | Назначение | Сброс |
|---------------|---|-----------|
| VALUE | 32-разрядный регистр текущего значения таймера | 00000000h |
| RELOAD | 32-разрядный регистр начального значения счетчика таймера | 00000000h |

А.10 Регистры контроллера DMA

Таблица А.10.1 – Регистр статуса DMA

| STATUS | | A000_6000h | | Сброс: 0nnn0000h | |
|---|-----------------------|---|--|------------------|--|
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | - | | CHNLS-1 | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | - | | STATE | |
| | | - | | MASTER_ENBL | |
| | | 4 | | 4 | |
| Поле | Биты | Описание | | | |
| CHNLS-1 | 20-16 | Количество доступных каналов DMA минус один | | | |
| | | 00h | 1 канал | | |
| | | 01h | 2 канала | | |
| | | 02h | 3 канала | | |
| | | 1Fh | 32 канала | | |
| STATE | 7-4 | Текущее состояние конечного автомата управления контроллера | | | |
| | | 0h | В покое | | |
| | | 1h | Чтение управляющих данных канала | | |
| | | 2h | Чтение указателя конца данных источника | | |
| | | 3h | Чтение указателя конца данных приемника | | |
| | | 4h | Чтение данных источника | | |
| | | 5h | Запись данных в приемник | | |
| | | 6h | Ожидание запроса на выполнение прямого доступа | | |
| | | 7h | Запись управляющих данных канала | | |
| | | 8h | Приостановлен | | |
| | | 9h | Выполнен | | |
| | | Ah | Режим работы с периферией «Разборка-сборка» (scatter-gather) | | |
| Bh-Fh | Зарезервировано | | | | |
| MASTER_ENBL | 0 | Состояние контроллера DMA | | | |
| | | 0 | Работа контроллера запрещена | | |
| | | 1 | Работа контроллера разрешена | | |
| - | 31-21 15-8, 3-1 | Зарезервировано | | | |
| Примечание – Регистр доступен только для чтения. Возвращает состояние контроллера DMA. Во время сброса чтение регистра запрещено. | | | | | |

Таблица А.10.2 – Регистр конфигурации DMA

| CFG | | A000_6004h | | | | | | | | | | | | | | |
|----------------|-------------------|---|--------------------------|------------------------|----------------------------|--|--|---|--|--|--|--|--|--|--|--|
| | | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | |
| CHNL_PROT_CTRL | 7-5 | Задаёт параметры защиты шины АНВ-Lite при обращении контроллера к структурам управляющих данных каналов | | | | | | | | | | | | | | |
| | | Биты поля CHNL_PROT_CTRL | | | | | | | | | | | | | | |
| | | 7 | | | 6 | | | 5 | | | | | | | | |
| | | 0 | Доступ не кэшируется | Доступ не буферизуется | Доступ непривилегированный | | | | | | | | | | | |
| 1 | Доступ кэшируется | Доступ буферизуется | Доступ привилегированный | | | | | | | | | | | | | |
| MASTER_ENBL | 0 | Бит разрешения работы контроллера DMA | | | | | | | | | | | | | | |
| | | 0 | Запрещена | | | | | | | | | | | | | |
| | | 1 | Разрешена | | | | | | | | | | | | | |
| – | 31-8, 4-1 | Зарезервировано | | | | | | | | | | | | | | |

Таблица А.10.3 – Регистр базового адреса управляющих данных каналов

| CTRL_BASE_PTR | | A000_6008h | | | | | | | | | | | | | | Сброс: 00000000h | |
|--|-------|---|--|--|--|--|--|--|--|--|--|--|--|--|--|------------------|--|
| | | | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | | |
| CTRL_BASE_PTR | 31-10 | Указатель на базовый адрес первичной структуры управляющих данных | | | | | | | | | | | | | | | |
| – | 9-0 | Зарезервировано | | | | | | | | | | | | | | | |
| <p>Примечание – Регистр доступен для чтения и записи и определяет базовый адрес системной памяти размещения управляющих данных каналов. Во время сброса чтение регистра запрещено.</p> | | | | | | | | | | | | | | | | | |

Таблица А.10.4 – Регистр базового адреса альтернативных управляющих данных каналов

| | | | | | | | | | | | | | | | | | |
|--|----|------|----|---|----|----|----|----|----|----|----|----|----|------------|----|------------------|--|
| ALT_CTRL_BASE_PTR | | | | | | | | | | | | | | A000_6008h | | Сброс: 00000000h | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | |
| ALT_CTRL_BASE_PTR | | | | | | | | | | | | | | | | | |
| Ч | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| ALT_CTRL_BASE_PTR - | | | | | | | | | | | | | | | | | |
| Ч | | | | | | | | | | | | | | | | | |
| Поле | | Биты | | Описание | | | | | | | | | | | | | |
| ALT_CTRL_BASE_PTR | | 31-0 | | Указатель базового адреса альтернативной структуры управляющих данных каналов | | | | | | | | | | | | | |
| Примечания | | | | | | | | | | | | | | | | | |
| 1 Регистр доступен только для чтения. Возвращает указатель базового адреса альтернативных управляющих данных каналов. Во время сброса чтение регистра запрещено. | | | | | | | | | | | | | | | | | |
| 2 Регистр позволяет не производить вычисления базового адреса альтернативных управляющих данных каналов. | | | | | | | | | | | | | | | | | |

Таблица А.10.5 – Регистры управления каналами

| | | | | | | | | | | | | | | | |
|---|-----|-----|-----|--|--|----|----|-----|-----|-----|-----|-----|-----|-----|-----|
| Формат регистров | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | C23 | C22 | C21 | C20 | C19 | C18 | C17 | C16 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| C15 | C14 | C13 | C12 | C11 | C10 | C9 | C8 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
| Каждый бит связан с одним из 24 каналов. Нулевой бит связан с каналом 0, 23-й бит – с каналом 23. i – порядковый номер канала от 0 до 23. Биты с 31 по 24 зарезервированы | | | | | | | | | | | | | | | |
| Мнемоника | | | | Назначение и описание | | | | | | | | | | | |
| WAITONREQ_STATUS | | | | Регистр статуса ожидания запросов для передачи. Информация, получаемая при чтении бита Ci | | | | | | | | | | | |
| Только чтение по адресу A0006010h | | | | 0 | Доступны только BREQ запросы от периферии | | | | | | | | | | |
| | | | | 1 | Доступны BREQ и SREQ запросы от периферии | | | | | | | | | | |
| CHNL_SW_REQUEST | | | | Регистр программного запроса на обработку каналов DMA Запись в бит Ci | | | | | | | | | | | |
| Только запись по адресу A0006014h | | | | 0 | Нет запроса | | | | | | | | | | |
| | | | | 1 | Устанавливает запрос на выполнение цикла DMA по каналу i | | | | | | | | | | |
| | | | | Запись бита, который соответствует нереализованному каналу, означает, что запрос на выполнение цикла DMA не будет установлен | | | | | | | | | | | |

Продолжение таблицы A10.5

| Мнемоника | Назначение и описание | |
|--|---|---|
| CHNL_USEBURST_SET Запись и чтение по адресу A0006018h | Регистр установки пакетного обмена каналов DMA. Отключает выполнение одиночных запросов по запросу SREQ и поэтому, будут обрабатываться и исполняться только запросы по BREQ. Информация, получаемая при чтении бита Ci | |
| | 0 | Канал DMA i выполняет циклы DMA в ответ на запросы на одиночную передачу SREQ и пакетный обмен BREQ. |
| | 1 | Канал DMA i выполняет циклы DMA только в ответ на запросы пакетного обмена (BREQ) |
| | Запись в бит Ci | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_USEBURST_CLR и сбросить в нем соответствующий бит Ci |
| | 1 | Отключает возможность обрабатывать запросы одиночного обмена SREQ на выполнение циклов DMA |
| | Запись в бит, который соответствует нереализованному каналу, не дает эффекта | |
| CHNL_USEBURST_CLR Только запись по адресу A000601Ch | Регистр сброса пакетного обмена каналов DMA Запись в бит Ci | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_USEBURST_SET для отключения обработки запросов одиночного обмена SREQ |
| | 1 | Разрешает обрабатывать одиночные запросы SREQ на выполнение циклов DMA |
| | Запись в бит, который соответствует нереализованному каналу, не дает эффекта | |
| CHNL_REQ_MASK_SET Запись и чтение по адресу A0006020h | Регистр маскирования запросов от периферии на обслуживание каналов DMA. При чтении возвращает состояние установок маскирования запросов SREQ и BREQ на обслуживание каналов. Информация, получаемая при чтении бита Ci | |
| | 0 | Канал i выполняет циклы DMA в ответ на поступающие запросы |
| | 1 | Канал i не выполняет циклы DMA в ответ на поступающие запросы |
| | Запись в бит Ci: | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_REQ_MASK_CLR для разрешения установки запросов |
| | 1 | Отключает выполнение циклов DMA по запросам SREQ и BREQ |
| | Запись в бит, который соответствует нереализованному каналу, не дает эффекта | |

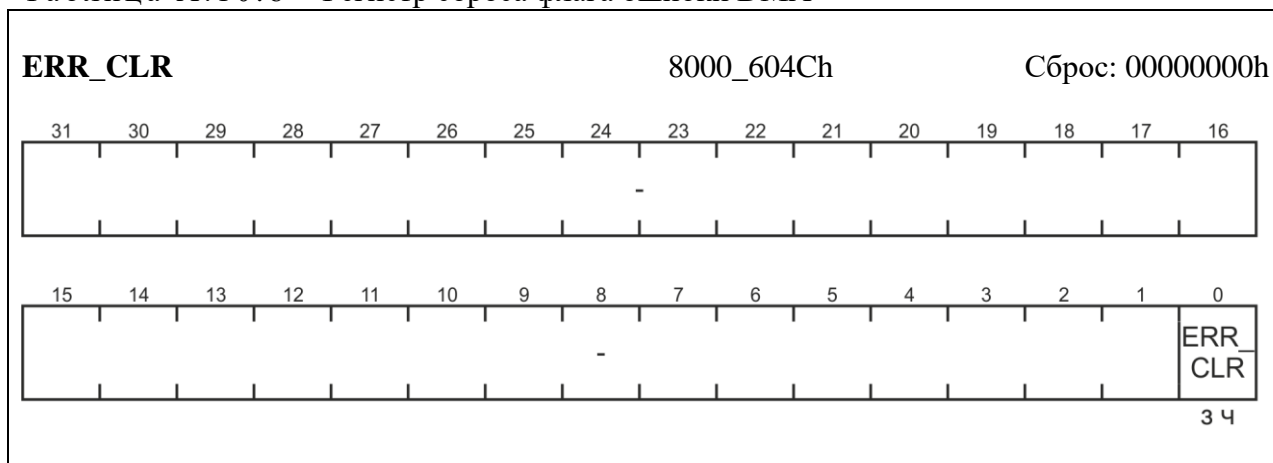
Продолжение таблицы А.10.5

| Мнемоника | Назначение и описание | |
|--|--|--|
| CHNL_REQ_MASK_CLR Только запись по адресу A0006024h | Регистр очистки маскирования запросов на обслуживание каналов DMA. Запись в бит Ci | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_REQ_MASK_SET для отключения установки запросов |
| | 1 | Разрешает выполнение циклов DMA по запросам SREQ и BREQ |
| | Запись в бит, который соответствует нереализованному каналу, не дает эффекта | |
| CHNL_ENABLE_SET Запись и чтение по адресу A0006028h | Регистр установки разрешения работы каналов DMA. Информация, получаемая при чтении бита Ci | |
| | 0 | Канал DMA i отключен |
| | 1 | Работа канала DMA i разрешена |
| | Запись в бит Ci: | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_ENABLE_CLR для отключения канала i |
| | 1 | Разрешает работу канала i |
| Запись в бит, который соответствует нереализованному каналу, не дает эффекта | | |
| CHNL_ENABLE_CLR Только запись по адресу A000602Ch | Регистр сброса разрешения работы каналов DMA. Запись в бит Ci | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_ENABLE_SET для разрешения работы канала i |
| | 1 | Запрещает работу канала i |
| | Запись в бит, который соответствует нереализованному каналу, не дает эффекта Примечание – Контроллер может отключить канал DMA i, в следующих случаях: - при завершении цикла DMA; - при чтении из channel_cfg с полем cycle_ctrl установленным в 000b; - при появлении ошибки на шине АНВ-Lite. | |
| CHNL_PRI_ALT_SET Запись и чтение по адресу A0006030h | Регистр установки первичной/альтернативной структуры управляющих данных каналов DMA. Информация, получаемая при чтении бита Ci | |
| | 0 | Канал DMA i использует первичную структуру управляющих данных |
| | 1 | Канал DMA i использует альтернативную структуру управляющих данных |
| | Запись в бит Ci | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_PRI_ALT_CLR для сброса бита Ci |
| | 1 | Включает использование альтернативной структуры управляющих данных каналом DMA i |

Окончание таблицы А.10.5

| Мнемоника | Назначение и описание | |
|--|---|---|
| CHNL_PRI_ALT_SET Запись и чтение по адресу A0006030h | Запись в бит, который соответствует нереализованному каналу, не дает эффекта. Примечание – Контроллер может переключить состояние бита Ci в следующих случаях: - при завершении четырех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режимах работы с памятью или периферией «разборка-сборка»; - при завершении всех передач DMA указанных в первичной структуре управляющих данных при выполнении цикла DMA в режиме «Пинг-понг»; - при завершении всех передач DMA указанных в альтернативной структуре управляющих данных при выполнении цикла DMA в режимах: - «пинг-понг»; - работа с памятью и периферией в режиме «разборка-сборка». | |
| CHNL_PRI_ALT_CLR Только запись по адресу A0006034h | Регистр сброса первичной/альтернативной структуры управляющих данных каналов DMA Запись в бит Ci | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_PRI_ALT_SET для выбора альтернативных управляющих данных |
| | 1 | Включает использование первичной структуры управляющих данных каналом DMA i |
| CHNL_PRIORITY_SET Запись и чтение по адресу A0006038h | Регистр установки приоритета каналов DMA Информация, получаемая при чтении бита Ci | |
| | 0 | Каналу DMA i присвоен уровень приоритета по умолчанию |
| | 1 | Каналу DMA i присвоен высокий уровень приоритета |
| | Запись в бит Ci: | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_PRIORITY_CLR для присвоения каналу i уровня приоритета по умолчанию |
| | 1 | Присваивает каналу DMA i высокий уровень приоритета |
| | Запись в бит, который соответствует нереализованному каналу, не дает эффекта | |
| CHNL_PRIORITY_CLR Только запись по адресу A000603Ch | Регистр сброса установок приоритета каналов DMA Запись в бит Ci | |
| | 0 | Нет эффекта. Необходимо использовать регистр CHNL_PRIORITY_SET для присвоения каналу i высокого уровня приоритета |
| | 1 | Присваивает каналу DMA i уровень приоритета по умолчанию |
| | Запись в бит, который соответствует нереализованному каналу, не дает эффекта | |

Таблица А.10.6 – Регистр сброса флага ошибки DMA



| Поле | Биты | Описание |
|--|------|--|
| ERR_CLR | 0 | Флаг ошибки по шине АHB-Lite Информация, получаемая при чтении: |
| | | 0 Ошибок не обнаружено |
| | | 1 Произошла ошибка |
| | | Запись в регистр: |
| | | 0 Нет эффекта |
| | | 1 Сброс флага ошибки ERR_CLR |
| Примечание - При сбросе флага ERR_CLR одновременно с появлением ошибки на шине АHB-Lite, приоритет отдается ошибке и, следовательно, бит ERR_CLR остается установленным. | | |
| – | 31-1 | Зарезервировано |

А.11 Регистры блока UART

Таблица А.11.1 – Регистр данных

| DR | | Сброс: 00000000h |
|----------------|--------------|--|
| | | |
| Поле | Биты | Описание |
| OE, BE, PE, FE | 11, 10, 9, 8 | См. описание бит в регистре RSR_ECR |
| DATA | 7-0 | Поле данных. Результатом записи в поле DATA является размещение байта в буфере передатчика, а результатом чтения – считывание байта из буфера приемника |
| – | 31-12 | Зарезервировано |

Таблица А.11.2 – Регистр состояния приемника и сброса ошибки приемника

| RSR_ECR | | Сброс: 00000000h |
|---------|------|--|
| | | |
| Поле | Биты | Описание |
| OE | 3 | Флаг переполнения буфера приемника |
| | | 0 В буфере есть свободное место или бит был сброшен после записи в регистр RSR_ECR. Содержимое буфера остается верным, так как перезаписан был только сдвиговый регистр. Центральный процессор должен считать данные для того, чтобы освободить буфер |
| | | 1 Буфер заполнен, а данные продолжают поступать |

Окончание таблицы А.11.2

| Поле | Биты | Описание | |
|--|------|-------------------------------|--|
| BE | 2 | Флаг разрыва линии | |
| | | 0 | Нормальная работа или бит был сброшен после записи в регистр RSR_ECR |
| | | 1 | Обнаружен признак разрыва линии, то есть наличие низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного кадра данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном режиме FIFO данная ошибка ассоциируется с последним байтом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой кадр. Прием данных возобновляется только после перехода линии в логическую единицу и последующего обнаружения корректного стартового бита |
| PE | 1 | Флаг ошибки контроля четности | |
| | | 0 | Нормальная работа или бит был сброшен после записи в регистр RSR_ECR |
| | | 1 | Четность принятого кадра данных не соответствует установкам битов EPS и SPS в регистре управления линией LCR_H. При включенном режиме FIFO данная ошибка ассоциируется с байтом, находящимся на вершине буфера |
| FE | 0 | Флаг ошибка в структуре кадра | |
| | | 0 | Нормальная работа или бит был сброшен после записи в регистр RSR_ECR |
| | | 1 | В принятом символе не обнаружен корректный стоповый бит (единица). При включенном режиме FIFO данная ошибка ассоциируется с байтом, находящимся на вершине буфера |
| – | 31-4 | Зарезервировано | |
| Все флаги сбрасываются записью единицы | | | |

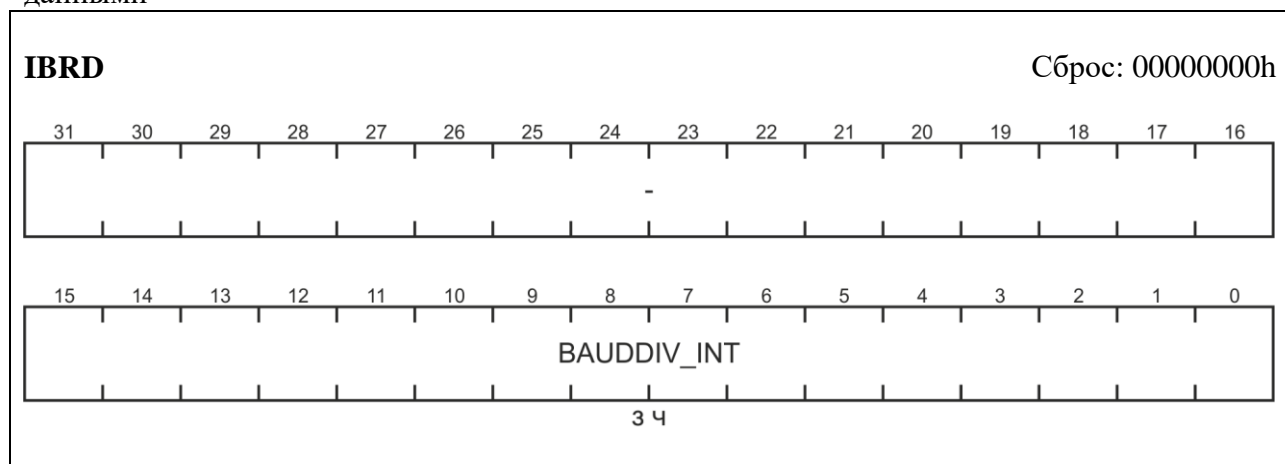
Таблица А.11.3 – Регистр флагов

| FR | | | | | | | | | | | | | Сброс: 0000090h | | | | | | | | | | | | | | | |
|-----------|-----|-----------------------|-----------|----|----|----|----|----|----|----|----|----|-----------------|----|----|----|------|------|------|------|------|-----|-----|-----|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | RI | TXFE | RXFF | TXFF | RXFE | BUSY | DCD | DSR | CST | | | | |
| - | | | | | | | | | | | | | | | | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч | ч |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RI | 8 | Инверсия линии UARTRI | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Выключено | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Включено | | | | | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.11.3

| Поле | Биты | Описание |
|---|---------|--|
| TXFE/ RXFE | 7/ 4 | Флаг пустоты буфера передатчика/приемника. Установка флага зависит от состояния бита FEN регистра LCR_H |
| | | 0 Буфер не пуст |
| | | 1 Буфер пуст |
| Примечание – Бит TXFE/RXFE не дает никакой информации о наличии данных в сдвиговом передающем регистре. | | |
| RXFF/ TXFF | 6/ 5 | Флаг заполнения буфера приемника/передатчика. Установка флага зависит от состояния бита FEN регистра LCR_H (т. е. включен режим FIFO или нет) |
| | | 0 Буфер не заполнен |
| | | 1 Если режим FIFO запрещен, бит устанавливается, когда буферный регистр приемника/передатчика занят. Если режим FIFO разрешен бит устанавливается, если заполнен буфер приемника/ передатчика |
| BUSY | 3 | Бит занятости блока UART |
| | | 0 Блок не занят |
| | | 1 Блок передает данные на линию. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Также бит устанавливается при наличии данных в буфере передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен) |
| DCD | 2 | Индикатор инверсии сигнала на линии UART_DCD |
| | | 0 Прямая передача |
| | | 1 Инверсия |
| DSR | 1 | Индикатор инверсии сигнала на линии UART_DSR |
| | | 0 Прямая передача |
| | | 1 Инверсия |
| CTS | 0 | Индикатор инверсии сигнала на линии UART_CTS |
| | | 0 Прямая передача |
| | | 1 Инверсия |
| – | 31-9 | Зарезервировано |

Таблица А.11.4 – Регистры целой и дробной частей делителя скорости обмена данными



Окончание таблицы А.11.4

| FBRD | | Сброс: 00000000h |
|---|------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> - BAUDDIV_FRAC </div> <div style="text-align: right; margin-top: 5px;">3 4</div> </div> | | |
| Поле | Биты | Описание |
| BAUDDIV_INT | 15-0 | Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. Минимальное значение 0001h |
| BAUDDIV_FRAC | 5-0 | Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. При BAUDDIV_INT = FFFFh, значение BAUDDIV_FRAC может быть только 00h. Невыполнение этого условия приведет к прерыванию приема/передачи |
| Остальные биты регистров зарезервированы | | |

Таблица А.11.5 – Регистр управления линией

| LCR_H | | Сброс: 00000000h | | |
|---|--|--|-------|-----------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> - SPS WLEN FEN STP2 EPS PEN BRK </div> <div style="text-align: right; margin-top: 5px;"> 3 4 3 4 3 4 3 4 3 4 3 4 3 4 </div> </div> | | | | |
| Поле | Биты | Описание | | |
| SPS | 7 | Бит разрешения передачи бита четности с фиксированным значением. Состояние бита не важно, если бит PEN сброшен | | |
| | | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>Запрещено</td> </tr> <tr> <td style="width: 20px; text-align: center;">1</td> <td>На месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. При EPS = 0 на месте бита четности передается единица. При EPS = 1 на месте бита четности передается ноль</td> </tr> </table> | 0 | Запрещено |
| 0 | Запрещено | | | |
| 1 | На месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. При EPS = 0 на месте бита четности передается единица. При EPS = 1 на месте бита четности передается ноль | | | |
| WLEN | 6-5 | Поле количества передаваемых/принимаемых информационных бит | | |
| | | 00 | 5 бит | |
| | | 01 | 6 бит | |
| | | 10 | 7 бит | |
| | | 11 | 8 бит | |

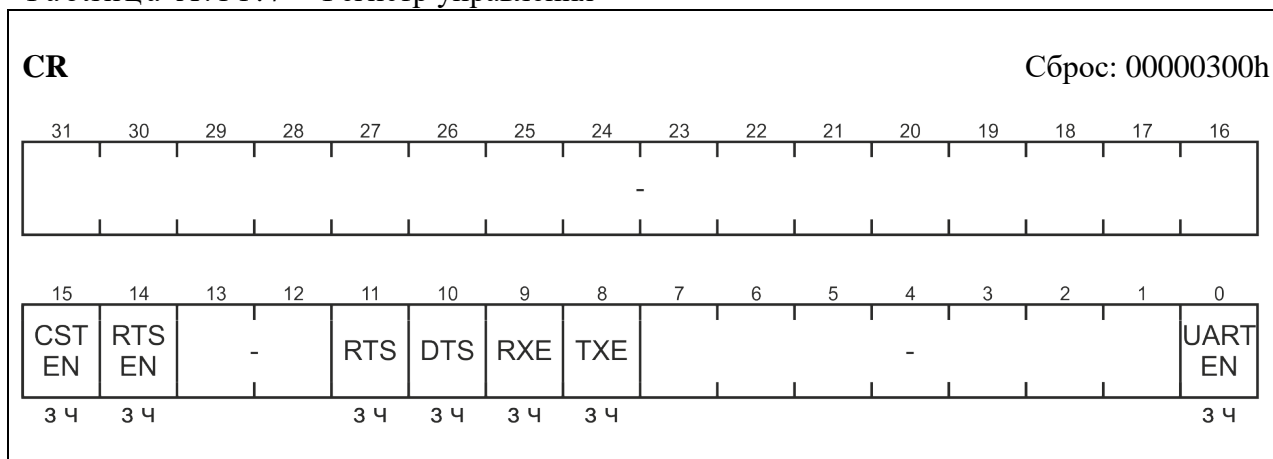
Окончание таблицы А.11.5

| Поле | Биты | Описание |
|---|------|---|
| FEN | 4 | Бит включения режима FIFO буфера приемника и передатчика |
| | | 0 Выключен |
| | | 1 Включен |
| STP2 | 3 | Бит выбора режима передачи стопового бита |
| | | 0 Один стоповый бит |
| | | 1 Два стоповых бита |
| | | Примечание – Приемник не проверяет наличие дополнительного стопового бита в кадре |
| EPS | 2 | Бит паритета Состояние бита не важно, если бит PEN сброшен |
| | | 0 Бит четности дополняет количество единиц в информационной части кадра до нечетного числа |
| | | 1 Бит четности дополняет количество единиц в информационной части кадра до четного числа |
| PEN | 1 | Бит включения проверки четности |
| | | 0 Выключена. Кадр не содержит бита четности |
| | | 1 Включена. Бит четности передается в кадре и проверяется при приеме данных |
| BRK | 0 | Флаг разрыва линии |
| | | 0 Нормальная работа |
| | | 1 Если бит установлен, то по завершении передачи текущего символа на выходе передатчика устанавливается низкий уровень сигнала. Для правильного выполнения этой операции программное обеспечение должно обеспечить передачу сигнала разрыва в течение, как минимум, времени передачи двух информационных кадров |
| – | 31-8 | Зарезервировано |
| Примечание – Дополнительная информация о комбинациях состояний битов SPS, EPS и PEN для контроля паритета представлена в таблице А11.6. | | |

Таблица А.11.6 – Зависимость бита паритета от состояния битов регистра LCR_H

| Биты регистра LCR_H | | | Наличие и состояние бита паритета |
|---------------------|----------|-----|--------------------------------------|
| SPS | EPS | PEN | |
| Не важно | Не важно | 0 | Не передается, не проверяется |
| 0 | 0 | 1 | Проверка нечетности слова данных |
| 0 | 1 | 1 | Проверка четности слова данных |
| 1 | 0 | 1 | Бит четности постоянно равен единице |
| 1 | 1 | 1 | Бит четности постоянно равен нулю |

Таблица А.11.7 – Регистр управления



| Поле | Биты | Описание |
|-------------|-------------------------|---|
| CTSEN | 15 | Бит разрешения аппаратного управления потоком данных по линии CTS |
| | | 0 Запрещено |
| | | 1 Разрешено. Данные передаются в линию только при активном значении сигнала UART_CTS |
| RTSEN | 14 | Бит разрешения аппаратного управления потоком данных по линии RTS |
| | | 0 Запрещено |
| | | 1 Разрешено. Запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере приемника |
| RTS | 11 | Бит программного управления состоянием линии модема UART_RTS |
| | | 0 Высокий уровень |
| | | 1 Низкий уровень |
| DTS | 10 | Бит программного управления состоянием линии модема UART_DTR |
| | | 0 Высокий уровень |
| | | 1 Низкий уровень |
| RXE/ TXE | 9/ 8 | Бит разрешения приема/передачи |
| | | 0 Запрещено |
| | | 1 Разрешено |
| UARTEN | 0 | Бит разрешения работы приемопередатчика |
| | | 0 Запрещено |
| | | 1 Разрешено |
| – | 31-16, 13,12, 7-1 | Зарезервировано |

Таблица А.11.8 – Регистр порога прерывания по заполнению буфера в режиме FIFO

| IFLS | | Сброс: 00000012h | |
|--|-------------|---|-------------------------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 100%; height: 40px;"></div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">RXIFLSEL</div> <div style="width: 30px; text-align: center;">TXIFLSEL</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 3 ч3 ч </div> | | | |
| Поле | Биты | Описание | |
| RXIFLSEL/ TXIFLSEL | 5-3/ 2-0 | Порог заполнения/опустошения буфера приемника/передатчика, по достижении которого будет генерироваться прерывание | |
| | | Для приемника | |
| | | 000 | Заполнение на 1/8 |
| | | 001 | Заполнение на 1/4 |
| | | 010 | Заполнение на 1/2 (по умолчанию) |
| | | 011 | Заполнение на 3/4 |
| | | 100 | Заполнение на 7/8 |
| | | Остальные комбинации зарезервированы | |
| – | 31-6 | Зарезервировано | |

Таблица А.11.9 – Регистр прерываний

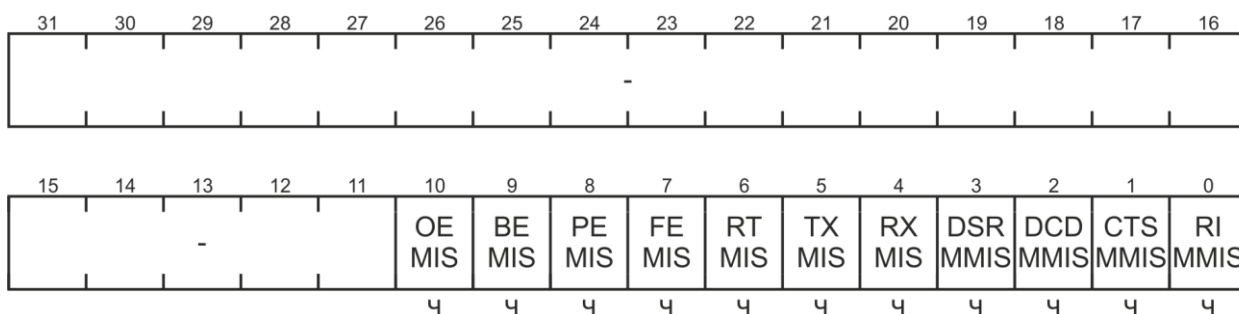
| Регистр состояния прерываний | | Сброс: 0000000xh |
|---|--|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 100%; height: 40px;"></div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">OE RIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">BE RIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">PE RIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">FE RIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">RT RIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">TX RIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">RX RIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">DSR RMIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">DCD RMIS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">CTS RMIS</div> <div style="width: 30px; text-align: center;">RI RMIS</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> ччччччччччч </div> | | |
| Регистр маски прерываний | | Сброс: 00000000h |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 100%; height: 40px;"></div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">OEIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">BEIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">PEIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">FEIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">RTIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">TXIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">RXIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">DSR MIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">DCD MIM</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">CTS MIM</div> <div style="width: 30px; text-align: center;">RI MIM</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 3 ч3 ч3 ч3 ч3 ч3 ч3 ч3 ч3 ч3 ч3 ч </div> | | |

Окончание таблицы А.11.9

Регистр состояния прерываний с маскированием

MIS

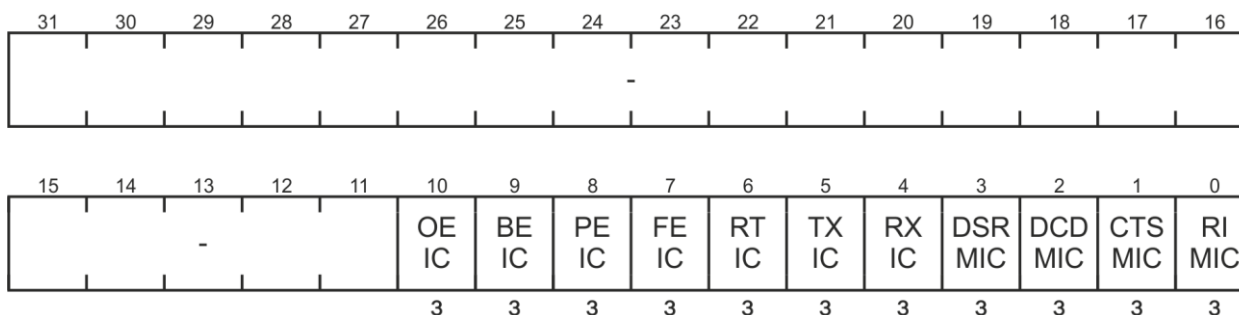
Сброс: 0000000xh



Регистр сброса прерываний

ICR

Сброс: 00000000h



| Поле | Бит | Описание |
|------|-------|-------------------------------------|
| OE | 10 | Переполнение буфера приемника |
| BE | 9 | Разрыв линии |
| PE | 8 | Ошибка контроля четности |
| FE | 7 | Ошибка в структуре кадра |
| RT | 6 | Таймаут приема данных |
| TX | 5 | Порог опустошения буфер передатчика |
| RX | 4 | Порог переполнения буфера приемника |
| DSR | 3 | Изменение состояния линии UART_DSR |
| DCD | 2 | Изменение состояния линии UART_DCD |
| CTS | 1 | Изменение состояния линии UART_CTS |
| RI | 0 | Изменение состояния линии UART_RI |
| – | 31-11 | Зарезервировано |

Функционирование регистров.

При возникновении прерываний устанавливаются соответствующие им немаскируемые флаги в регистре RIS.

Установка бит в регистре IMSC формирует маску.

В регистре MIS устанавливаются только те флаги, которые не закрыты маской регистра IMSC.

Запись единиц в биты регистра ICR сбрасывает соответствующие им флаги в регистрах RIS и MIS, а также прерывания, вызвавшие установку этих флагов

Таблица А.11.10 – Регистр управления прямым доступом к памяти

| DMACR | | Сброс: 00000000h | | | | | | | | | | | | | |
|-------------------|---------|--|-----------|----|----|----|----|----|----|----|----|----|------------------|----------------|----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | | | | | DMA ON ERR | TX DMA E | RX DMA E |
| | | | | | | | | | | | | | 3 4 | 3 4 | 3 4 |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| DMAONERR | 2 | Бит управления блокированием запросов DMA от приемника RXDMASREQ и RXDMABREQ в случае возникновения прерывания по ошибке | | | | | | | | | | | | | |
| | | 0 | Выключено | | | | | | | | | | | | |
| | | 1 | Включено | | | | | | | | | | | | |
| TXDMAE/ RXDMAE | 1/ 0 | Бит разрешения формирования запросов DMA для обслуживания буфера передатчика/приемника | | | | | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | | | | | |
| – | 31-3 | Зарезервировано | | | | | | | | | | | | | |

А.12 Регистры блока ШИМ

Таблица А.12.1 – Регистр управления таймером

| ТВCTL | | Сброс: 00000000h |
|--|-------|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">FREE/SOFT</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">PHS DIR</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">CLKDIV</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">HSPCLKDIV</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">SWF SYNC</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">SYNCOSEL</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">PRD LD</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">PHS EN</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">CTRMODE</div> </div> <div style="display: flex; justify-content: space-between; padding-top: 5px;"> 3 ч3 ч3 ч3 ч3 ч3 ч3 ч3 ч3 ч </div> | | |
| Поле | Биты | Описание |
| FREE/ SOFT | 15-14 | Поле задания поведения счетчика ШИМ после перехода в режим останова во время отладки |
| | | 00 Счетчик будет остановлен на следующий такт ТВCLK |
| | | 01 Счетчик будет остановлен по достижении события: - ТВCTR = ТВPRD (при счете вверх); - ТВCTR = 0000h (при счете вниз или вверх-вниз). |
| | | 1x Счетчик продолжит работу |
| PHSDIR | 13 | Бит задания фазового направления (используется только при двунаправленном счете). Задаёт направление счета после синхронизации. Загружается вместе с регистром фазы ТВPHS |
| | | 0 Вниз |
| | | 1 Вверх |
| CLKDIV | 12-10 | Поле задания первого делителя тактовой частоты |
| | | 000 1 |
| | | 001 1/2 |
| | | 010 1/4 |
| | | 011 1/8 |
| | | 100 1/16 |
| | | 101 1/32 |
| | | 110 1/64 |
| | | 111 1/128 |
| HSPCLKDIV | 9-7 | Поле задания второго делителя тактовой частоты. Конечное значение делителя является произведением значений делителей, задаваемых полями CLKDIV и HSPCLKDIV |
| | | 000 1 |
| | | 001 1/2 |
| | | 010 1/4 |
| | | 011 1/6 |
| | | 100 1/8 |
| | | 101 1/10 |
| | | 110 1/12 |
| | | 111 1/14 |

Окончание таблицы А.12.1

| Поле | Биты | Описание | |
|----------|-------|---|---|
| SWFSYNC | 6 | Бит программной эмуляция появления синхроимпульса | |
| | | 0 | Нет действий |
| | | 1 | Запись единицы вызывает появление синхроимпульса в цепи PWM_SYNCI |
| SYNCOSEL | 5-4 | Поле выбора источника для выходного сигнала синхронизации PWM_SYNCO | |
| | | 00 | PWM_SYNCI |
| | | 01 | CTR = 0000h |
| | | 10 | CTR = CMPB |
| | | 11 | Выдача синхроимпульса запрещена |
| PRDLD | 3 | Бит управления загрузкой регистра TBPRD | |
| | | 0 | Режим отложенной загрузки регистра TBPRD разрешен |
| | | 1 | Запись в TBPRD будет произведена сразу в активный регистр |
| PHSEN | 2 | Бит разрешения загрузки счетчика таймера | |
| | | 0 | Запрещено |
| | | 1 | Разрешена загрузка счетчика TBCTR значением регистра фазы TBPHS при получении события синхронизации (импульс на входе PWM_SYNCI или запись в бит SWFSYNC) |
| CTRMODE | 1-0 | Поле задания направления счета | |
| | | 00 | Вверх |
| | | 01 | Вниз |
| | | 10 | Вверх-вниз |
| | | 11 | Счетчик остановлен |
| – | 31-16 | Зарезервировано | |

Таблица А.12.2 – Регистр статуса таймера

| Поле | Бит | Описание |
|---|-----|---|
| <div style="display: flex; justify-content: space-between;"> TBSTS Сброс: 00000000h </div> | | |
|  | | |
| CTRMAX | 2 | Флаг достижения счетчиком таймера своего максимального значения FFFFh |
| | 0 | Значение не достигнуто или флаг был сброшен |
| | 1 | Значение было достигнуто |
| | | Запись единицы сбрасывает флаг |

Окончание таблицы А.12.2

| Поле | Биты | Описание |
|--------|------|--|
| SYNCl | 1 | Флаг синхронизации |
| | | 0 Синхронизация не достигнута или флаг был сброшен |
| | | 1 Синхронизация произошла |
| | | Запись единицы сбрасывает флаг |
| CTRDIR | 0 | Флаг направление счета таймера |
| | | 0 Вниз |
| | | 1 Вверх |
| — | 31-3 | Зарезервировано |

Таблица А.12.3 – Регистр фазы

| Поле | Биты | Описание |
|--------------|-------|--|
| ТВРНС | | Сброс: 00000000h |
| | | |
| ТВРНС | 31-16 | Поле задания начальной фазы таймера при получении сигнала синхронизации |
| ТВРНСHR | 15-8 | Поле дополнительных разрядов начальной фазы таймер. Доступен в блоке ШИМ высокого разрешения. Если бит РНSEN сброшен, то синхронизация отключена, и таймер не будет загружаться значением ТВРНС. Если бит РНSEN установлен, то по получению события синхронизации в счетчик таймера ТВСТР будет загружено значение ТВРНС |
| — | 7-0 | Зарезервировано |

Таблица А.12.4 – Регистр текущего значения таймера

| Поле | Биты | Описание |
|--------------|------|------------------|
| ТВСТР | | Сброс: 00000000h |
| | | |
| — | 31-0 | Зарезервировано |

Окончание таблицы А.12.4

| Поле | Биты | Описание |
|-------|-------|--|
| TBCTR | 15-0 | Текущее значение счетчика таймера. Запись в регистр изменяет значение таймера. Запись происходит асинхронно с TBLK и не использует отложенный механизм загрузки |
| – | 31-16 | Зарезервировано |

Таблица А.12.5 – Регистр максимального значения таймера

| Поле | Биты | Описание |
|--------------|-------|--|
| TBPRD | | Сброс: 00000000h |
| | | |
| | | 3 ч |
| Поле | Биты | Описание |
| TBPRD | 15-0 | Период таймера (максимальное значение счета таймера). Отложенная загрузка в этот регистр программируется битом PRDLД регистра TBCTL. По умолчанию бит PRDLД сброшен и запись в регистр TBPRD приводит к записи в теневой регистр. Активный регистр будет загружен по событию TBCTR = Zero. Если бит PRDLД установлен, то запись выполняется напрямую в активный регистр |
| – | 31-16 | Зарезервировано |

Таблица А.12.6 – Регистр управления компаратором

| Поле | Биты | Описание |
|---------------|------|--|
| СМРCTL | | Сброс: 00000000h |
| | | |
| | | 4 ч 4 ч 3 ч 3 ч 3 ч 3 ч |
| Поле | Биты | Описание |
| SHDWB FULL | 9 | Флаг отложенной загрузки в регистр СМРВ |
| | | 0 Нет действий |
| | 1 | Активный регистр загружен значением из теневого регистра |
| SHDWA FULL | 8 | Флаг отложенной загрузки в регистр СМРА |
| | | 0 Нет действий |
| | 1 | Активный регистр загружен значением из теневого регистра |

Окончание таблицы А.12.6

| Поле | Биты | Описание |
|-----------|----------------|--|
| SHDWBMODE | 6 | Бит управления загрузкой регистра CMPB |
| | | 0 Значение, записываемое в регистр CMPB, размещается в теневом регистре (отложенная загрузка) |
| | | 1 Производится загрузка напрямую в активный регистр |
| SHDWAMODE | 4 | Бит управления загрузкой регистра CMPA |
| | | 0 Значение, записываемое в регистр CMPA, размещается в теневом регистре (отложенная загрузка) |
| | | 1 Производится загрузка напрямую в активный регистр |
| LOADBMODE | 3-2 | Поле задания события загрузки отложенного значения в регистр CMPB (при условии, что бит SHDWBMODE сброшен) |
| | | 00 CTR = Zero |
| | | 01 CTR = PRD |
| | | 10 CTR = Zero или CTR = PRD |
| | | 11 Загрузка запрещена |
| LOADAMODE | 1-0 | Поле задания события загрузки отложенного значения в регистр CMPA (при условии, что бит SHDWAMODE сброшен) |
| | | 00 CTR = Zero |
| | | 01 CTR = PRD |
| | | 10 CTR = Zero или CTR = PRD |
| | | 11 Загрузка запрещена |
| – | 31-10, 7, 5 | Зарезервировано |

Таблица А.12.7 – Регистр порога срабатывания А

| Поле | Биты | Описание |
|--|-------|---|
| <p>СМРА Сброс: 00000000h</p> | | |
| СМРА | 31-16 | Активное значение порога срабатывания канала А, которое сравнивается со значением счетчика таймера. При совпадении значений формируется событие CTR = СМРА, которое влияет на поведение сигналов на линиях РWМА и РWМВ |
| СМРАНР | 15-8 | Дополнительные младшие биты значения порога срабатывания канала А (используются только для блока ШИМ высокого разрешения). Отложенная загрузка включается и работает также как и для поля СМРА |
| – | 7-0 | Зарезервировано |

Таблица А.12.8 – Регистр порога срабатывания В

| СМРВ | | | | | | | | | | | | | Сброс: 00000000h | | |
|-------------|-------|---|----|----|----|----|----|----|----|----|----|----|------------------|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| СМРВ | | | | | | | | | | | | | | | |
| 3 ч | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| СМРВ | 31-16 | Активное значение порога срабатывания канала В, которое сравнивается со значением счетчика таймера. При совпадении значений формируется событие CTR = СМРВ, которое влияет на поведение сигналов на линиях РWМА и РWМВ | | | | | | | | | | | | | |
| – | 15-0 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.12.9 – Регистр обработчика для выхода А/В

| AQCTLA / AQCTLB | | | | | | | | | | | | | Сброс: 00000000h | | | |
|---|-------|---|----|-----|----|----|-----|----|-----|----|-----|----|------------------|----|-----|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| - | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| - | | | | CBD | | | CBU | | CAD | | CAU | | PRD | | ZRO | |
| | | | | 3 ч | | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | |
| Поле | Биты | Описание | | | | | | | | | | | | | | |
| CBD | 11-10 | Действие на выводе РWМА/РWМВ при CTR = СМРВ при счете вниз | | | | | | | | | | | | | | |
| CBU | 9-8 | Действие на выводе РWМА/РWМВ при CTR = СМРВ при счете вверх | | | | | | | | | | | | | | |
| CAD | 7-6 | Действие на выводе РWМА/РWМВ при CTR = СМРА при счете вниз | | | | | | | | | | | | | | |
| CAU | 5-4 | Действие на выводе РWМА/РWМВ при CTR = СМРА при счете вверх | | | | | | | | | | | | | | |
| PRD | 3-2 | Действие на выводе РWМА/РWМВ при CTR = PRD | | | | | | | | | | | | | | |
| ZRO | 1-0 | Действие на выводе РWМА/РWМВ при CTR = Zero | | | | | | | | | | | | | | |
| – | 31-12 | Зарезервировано | | | | | | | | | | | | | | |
| Для каждого события может быть задано одно из четырех действий: | | | | | | | | | | | | | | | | |
| 00 – Нет реакции; | | | | | | | | | | | | | | | | |
| 01 – Переключение РWМА/РWМВ в ноль; | | | | | | | | | | | | | | | | |
| 10 – Переключение РWМА/РWМВ в единицу; | | | | | | | | | | | | | | | | |
| 11 – Инверсия РWМА/РWМВ | | | | | | | | | | | | | | | | |

Таблица А.12.10 – Регистр программного управления однократным действием

| AQSFRC | | Сброс: 00000000h |
|---|------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 40%;"></div> <div style="border-right: 1px solid black; width: 5%; text-align: center;">-</div> <div style="border-right: 1px solid black; width: 10%; text-align: center;">RLDCSF</div> <div style="border-right: 1px solid black; width: 5%; text-align: center;">OT SFB</div> <div style="border-right: 1px solid black; width: 10%; text-align: center;">ACTSFB</div> <div style="border-right: 1px solid black; width: 5%; text-align: center;">OT SFA</div> <div style="width: 10%; text-align: center;">ACTSFA</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 3 ч3 ч3 ч3 ч3 ч </div> | | |
| Поле | Биты | Описание |
| RLDCSF | 7-6 | Не используется. Записывать только 00b! |
| OTSFB | 5 | Запись единицы инициирует ожидание события CTR = Zero для однократного переключения вывода в состояние, заданное полем ACTSFB |
| ACTSFB | 4-3 | Выбор действия с выходным сигналом на выводе |
| | | 00 Нет действий |
| | | 01 PWMB = 0 |
| | | 10 PWMB = 1 |
| 11 Инверсия PWMB | | |
| OTSFA | 2 | Запись единицы инициирует ожидание события CTR = Zero для однократного переключения вывода в состояние, заданное полем ACTSFA |
| ACTSFA | 1-0 | Выбор действия с выходным сигналом на выводе |
| | | 00 Нет действий |
| | | 01 PWMA = 0 |
| | | 10 PWMA = 1 |
| 11 Инверсия PWMA | | |
| – | 31-8 | Зарезервировано |

Таблица А.12.11 – Регистр обработчика для циклического программного управления

| AQCSFRC | | Сброс: 00000000h |
|--|--|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 40%;"></div> <div style="border-right: 1px solid black; width: 5%; text-align: center;">-</div> <div style="border-right: 1px solid black; width: 10%; text-align: center;">CSFB</div> <div style="width: 10%; text-align: center;">CSFA</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 3 ч3 ч </div> | | |

Окончание таблицы А.12.11

| Поле | Биты | Описание |
|---------------|-------------|---|
| CSFB/ CSFA | 3-2/ 1-0 | Выбор действия с выходным сигналом на выводе PWMB/PWMA каждый раз, когда возникает событие CTR = Zero |
| | | 00 Нет действий |
| | | 01 PWMB/PWMA = 0 |
| | | 10 PWMB/PWMA = 1 |
| | | 11 Нет действий |
| – | 31-4 | Зарезервировано |

Таблица А.12.12 – Регистр управления генератором «мертвого времени» ШИМ

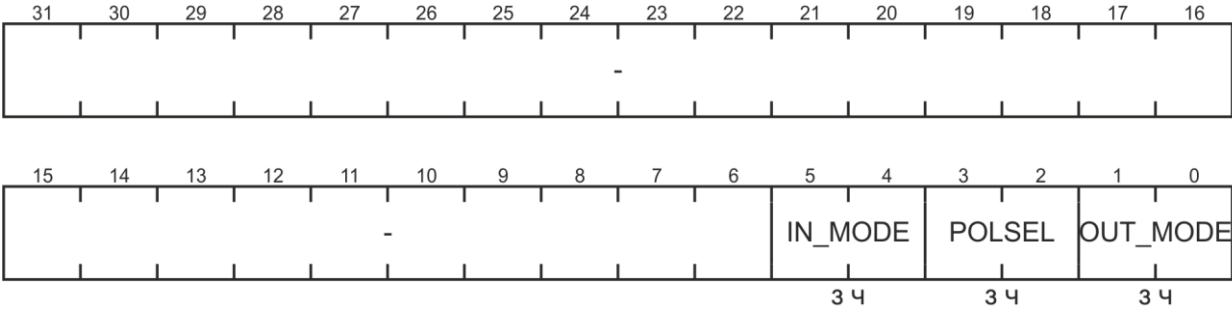
| Поле | Биты | Описание |
|---|------|---|
| <p>DBCTL Сброс: 00000000h</p>  | | |
| IN_MODE | 5-4 | Поле выбора источника для контроля по фронту и срезу. Старший бит поля управляет ключом S5, младший – ключом S4 (см. рисунок 14.12) |
| | | 00 Сигнал PWMA используется для контроля по переднему и заднему фронтам |
| | | 01 Сигнал PWMA используется для контроля по заднему фронту, а сигнал PWMB – по переднему |
| | | 10 Сигнал PWMA используется для контроля по переднему фронту, а сигнал PWMB – по заднему |
| | | 11 Сигнал PWMB используется для контроля по переднему и заднему фронтам |
| POLSEL | 3-2 | Поле задания полярности сигнала на выходе. Старший бит поля управляет ключом S3, а младший – ключом S2 (см. рисунок 14.12) |
| | | 00 Инверсия запрещена |
| | | 01 Инверсия только на выводе PWMA |
| | | 10 Инверсия только на выводе PWMB |
| | | 11 Инверсия на выводах PWMA и PWMB |
| OUT_MODE | 1-0 | Поле выбора фронта, для которого включена задержка («мертвое время»). Старший бит поля управляет ключом S1, а младший – ключом S0 (см. рисунок 14.12) |
| | | 00 Не задано |
| | | 01 Задний фронт PWMB |
| | | 10 Передний фронт PWMA |
| | | 11 Передний фронт PWMA и задний фронт PWMB |
| – | 31-6 | Зарезервировано |

Таблица А.12.13 – Регистр управления «мертвым временем»

| DBRED | | Сброс: 00000000h | | | | | | | | | | | | | |
|--------------|-------|---|--|--|--|--|--|--|--|--|--|--|--|--|--|
| DBFED | | Сброс: 00000000h | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| DEL (DBRED) | 9-0 | Величина задержки переднего фронта для генератора «мертвого времени» ШИМ (в периодах тактового сигнала TBCLK) | | | | | | | | | | | | | |
| DEL (DBFED) | 9-0 | Величина задержки заднего фронта для генератора «мертвого времени» ШИМ (в периодах тактового сигнала TBCLK) | | | | | | | | | | | | | |
| – | 31-10 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.12.14 – Регистр источника сигнала аварии

| TZSEL | | Сброс: 00000000h | | | | | | | | | | | | | |
|---|-------------|---|--|--|--|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| OSHTn | 13-8 | Бит разрешения источника сигнала аварии с вывода PWM_TZn в однократном режиме | | | | | | | | | | | | | |
| | | 0 Запрещено | | | | | | | | | | | | | |
| | | 1 Разрешено | | | | | | | | | | | | | |
| CBCn | 5-0 | Бит разрешения источника сигнала аварии с вывода PWM_TZn в циклическом режиме | | | | | | | | | | | | | |
| | | 0 Запрещено | | | | | | | | | | | | | |
| | | 1 Разрешено | | | | | | | | | | | | | |
| – | 31-14, 7, 6 | Зарезервировано | | | | | | | | | | | | | |
| Примечание – n – порядковый номер от 0 до 5 | | | | | | | | | | | | | | | |

Таблица А.12.15 – Регистр управления детектором сигнала аварии

| TZCTL | | Сброс: 00000000h |
|--|-------------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center; border: 1px solid black; padding: 5px;"> <div style="border: 1px solid black; width: 100%; height: 30px; margin-bottom: 5px;"></div> <div style="border: 1px solid black; padding: 2px;">TZB</div> <div style="border: 1px solid black; padding: 2px;">TZA</div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 ч3 ч </div> | | |
| Поле | Биты | Описание |
| TZB/ TZA | 3-2/ 1-0 | Поле задания поведения вывода PWM_B/PWM_A в случае получения сигнала аварии. Источник сигнала аварии при этом определяется регистром TZSEL |
| | | 00 Переключение в третье состояние |
| | | 01 Переключение в единицу |
| | | 10 Переключение в ноль |
| | | 11 Нет действий |
| – | 31-4 | Зарезервировано |

Таблица А.12.16 – Регистр маски прерываний детектора сигнала аварии

| TZEINT | | Сброс: 00000000h |
|--|---------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center; border: 1px solid black; padding: 5px;"> <div style="border: 1px solid black; width: 100%; height: 30px; margin-bottom: 5px;"></div> <div style="border: 1px solid black; padding: 2px;">OST</div> <div style="border: 1px solid black; padding: 2px;">CBC</div> <div style="border: 1px solid black; padding: 2px;">-</div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 ч3 ч </div> | | |
| Поле | Бит | Описание |
| OST | 2 | Бит разрешения генерации прерывания в однократном режиме обработки аварии |
| | | 0 Запрещено 1 Разрешено |
| CBC | 1 | Бит разрешения генерации прерывания в циклическом режиме обработки аварии |
| | | 0 Запрещено 1 Разрешено |
| – | 31-3, 0 | Зарезервировано |

Таблица А.12.17 – Регистр флагов прерываний детектора сигнала аварии

| TZFLG | | Сброс: 00000000h |
|---|------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | |
| OST CBC INT | | |
| 3 4 3 4 3 4 | | |
| Поле | Бит | Описание |
| OSHT | 2 | Флаг прерывания в однократном режиме обработки аварии |
| | | 0 Нет прерывания |
| | | 1 Запрос на прерывание |
| | | При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова |
| CBC | 1 | Флаг прерывания в циклическом режиме обработки аварии |
| | | 0 Нет прерывания |
| | | 1 Запрос на прерывание |
| | | При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова |
| INT | 0 | Флаг внешнего прерывания NVIC |
| | | 0 Нет прерывания |
| | | 1 Запрос на прерывание |
| | | Если флаг был сброшен, а один из флагов CBC или OST установлен, флаг установится снова |
| – | 31-3 | Зарезервировано |

Таблица А.12.18 – Регистр сброса флагов прерываний детектора сигнала аварии

| TZCLR | | Сброс: 00000000h |
|---|--|------------------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> | | |
| OST CBC INT | | |
| 3 4 3 4 3 4 | | |

Окончание таблицы А.12.18

| Поле | Бит | Описание |
|------|------|--|
| OST | 2 | Бит сброса флага прерывания в однократном режиме обработки аварии. Запись единицы сбрасывает бит OST в регистре TZFLG |
| CBC | 1 | Бит сброса флага прерывания в циклическом режиме обработки аварии. Запись единицы сбрасывает бит CBC в регистре TZFLG |
| INT | 0 | Бит сброса флага внешнего прерывания NVIC. Запись единицы сбрасывает бит INT в регистре TZFLG |
| – | 31-3 | Зарезервировано |

Таблица А.12.19 – Регистр программной эмуляции сигнала аварии

| Поле | Бит | Описание |
|------|------------|---|
| | | Сброс: 00000000h |
| | | 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 |
| | | - |
| | | 3 ч |
| | | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
| | | - OST CBC - |
| | | 3 ч 3 ч |
| OSHT | 2 | Бит программной генерации сигнала аварии в однократном режиме. Запись единицы устанавливает бит OST в регистре TZFLG |
| CBC | 1 | Бит программной генерации сигнала аварии в циклическом режиме. Запись единицы устанавливает бит CBC в регистре TZFLG |
| – | 31-3, 0 | Зарезервировано |

Примечание – Чтение битов OST и CBC всегда возвращает нули.

Таблица А.12.20 – Регистр источника триггера событий

| Поле | Биты | Описание |
|-------------------|-----------|---|
| | | Сброс: 00000000h |
| | | 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 |
| | | - |
| | | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
| | | SOCBEN SOCBSEL SOCAEN SOCASEL - INTEN INTSEL |
| | | 3 ч 3 ч 3 ч 3 ч 3 ч 3 ч |
| SOCBEN/ SOCAEN | 15/ 11 | Бит разрешения генерации внешнего сигнала PWM_SOCB/ PWM_SOCA для запуска АЦП |
| | 0 | Запрещено |
| | 1 | Разрешено |

Окончание таблицы А.12.20

| Поле | Биты | Описание | |
|--------------------------------|---------------------------|--|----------------------------|
| SOCBSEL/ SOCASEL/ INTSEL | 14-12/ 10-8/ 2-0 | Поле выбора события, по которому будет сформирован импульс PWM_SOCB/PWM_SOCА/PWM_INT | |
| | | 000 | Зарезервировано |
| | | 001 | CTR = Zero |
| | | 010 | CTR = PRD |
| | | 011 | Зарезервировано |
| | | 100 | CTR = CMPA при счете вверх |
| | | 101 | CTR = CMPA при счете вниз |
| | | 110 | CTR = CMPB при счете вверх |
| 111 | CTR = CMPB при счете вниз | | |
| INTEN | 3 | Бит разрешения генерации внешнего прерывания PWM_INT | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| – | 31-16, 7-4 | Зарезервировано | |

Таблица А.12.21 – Регистр предделителя триггера событий

| Поле | Биты | Описание | |
|---|-----------------|---|--|
| ETPS | | Сброс: 00000000h | |
| <div style="display: flex; justify-content: space-between; font-size: small;"> 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | | |
| <div style="display: flex; justify-content: space-between; font-size: small;"> 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 </div> <div style="display: flex; justify-content: space-between; font-size: small;"> <div style="border: 1px solid black; padding: 2px;">SOCBCNT</div> <div style="border: 1px solid black; padding: 2px;">SOCBPRD</div> <div style="border: 1px solid black; padding: 2px;">SOCACNT</div> <div style="border: 1px solid black; padding: 2px;">SOCAPRD</div> <div style="border: 1px solid black; padding: 2px;">-</div> <div style="border: 1px solid black; padding: 2px;">INTCNT</div> <div style="border: 1px solid black; padding: 2px;">INTPRD</div> </div> <div style="display: flex; justify-content: space-between; font-size: x-small; margin-top: 5px;"> 4 3 ч 4 3 ч 4 3 ч </div> | | | |
| SOCBCNT/ SOCACNT | 15-14/ 11-10 | Счетчик событий сигнала PWM_SOCB/ PWM_SOCА. Счетчик автоматически сбрасывается, когда сформировано прерывание, и перестает считать, когда достигает значения SOCBPRD/SOCAPRD соответственно | |
| SOCBPRD/ SOCAPRD | 13-12/ 9-8 | Поле задания количества событий, заданных полем SOCBSEL/SOCASEL регистра ETSEL, по которым будет сформирован сигнал запуска АЦП PWM_SOCB/PWM_SOCА. Для разрешения генерации сигнала нужно установить бит SOCBEN/ SOCAEN регистра ETSEL. Сигнал будет сформирован, даже если флаг SOCB/SOCA (регистр ETFLG) предыдущего сигнала не был сброшен. Как только сигнал PWM_SOCB/PWM_SOCА отправлен, счетчик SOCBCNT/SOCACNT автоматически сбрасывается | |
| | | 00 | По каждому событию (SOCBCNT/SOCACNT = 00b) |
| | | 01 | По каждым двум событиям (SOCBCNT/SOCACNT = 01b) |
| | | 10 | По каждым трём событиям (SOCBCNT/SOCACNT = 10b) |
| | | 11 | По каждым четырем событиям (SOCBCNT/SOCACNT = 11b) |

Окончание таблицы А.12.21

| Поле | Биты | Описание | |
|--------|---------------|---|---|
| INTCNT | 3-2 | Значение счетчика событий прерываний. Счетчик автоматически сбрасывается, когда сформировано прерывание, и перестает считать, когда достигает значения INTPRD | |
| INTPRD | 1-0 | Поле задания количества событий, заданных полем INTSEL регистра ETSEL, по которым будет сформировано внешнее прерывание PWM_INT. Для разрешения генерации прерывания нужно установить бит INTEN в регистре ETSEL. Если флаг прерывания INT (регистр ETFLG) установлен от предыдущего прерывания, то текущее прерывание не будет активировано до сброса этого флага (сбрасывается записью единицы в бит INT регистра ETCLR). Такой механизм позволяет обрабатывать одно прерывание, в то время как другое ждет своей очереди | |
| | | 00 | Прерывания по каждому событию (INTCNT = 00b) |
| | | 01 | Прерывания по каждому двум событиям (INTCNT = 01b) |
| | | 10 | Прерывания по каждому трём событиям (INTCNT = 10b) |
| | | 11 | Прерывания по каждому четырем событиям (INTCNT = 11b) |
| – | 31-16, 7-4 | Зарезервировано | |

Таблица А.12.22 – Регистр флагов триггера событий

| Поле | Бит | Описание |
|---|---------------|---|
| | | Сброс: 00000000h |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center; margin: 5px 0;"> <div style="border: 1px solid black; flex-grow: 1; height: 20px;"></div> <div style="border: 1px solid black; padding: 2px; text-align: center;">SOC B</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">SOC A</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">-</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">INT</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> чччч </div> | | |
| SOCB/ SOCA/ INT | 3/ 2/ 0 | Флаг внешнего сигнала АЦП PWM_SOCB/PWM_SOCA/PWM_INT |
| | | 0 Не установлен или сброшен |
| | | 1 Установлен |
| – | 31-4, 1 | Зарезервировано |

Таблица А.12.23 – Регистр сброса флагов триггера событий

| ETCLR | | Сброс: 00000000h | | | | | | | | | | | | | | | |
|---|---------------|--|--------------------------------|--|--|--|--|--|--|--|--|--|-------|--|-------|---|-----|
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | SOC B | | SOC A | - | INT |
| | | | | | | | | | | | | | 3 | | 3 | | 3 |
| Поле | Бит | Описание | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | | | | | | | | | | | | | | | |
| SOCB/ SOCA/ INT | 3/ 2/ 0 | Бит сброса флага SOCB/SOCA /INT в регистре ETFLG | | | | | | | | | | | | | | | |
| | | 0 | Нет действий | | | | | | | | | | | | | | |
| | | 1 | Запись единицы сбрасывает флаг | | | | | | | | | | | | | | |
| - | 31-4, 1 | Зарезервировано | | | | | | | | | | | | | | | |

Таблица А.12.24 – Регистр программной эмуляции флагов триггера событий

| ETFRC | | Сброс: 00000000h | | | | | | | | | | | | | | | |
|---|---------------|--|-----------------------------------|--|--|--|--|--|--|--|--|--|-------|--|-------|---|-----|
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | SOC B | | SOC A | - | INT |
| | | | | | | | | | | | | | 3 | | 3 | | 3 |
| Поле | Бит | Описание | | | | | | | | | | | | | | | |
| SOCB/ SOCA/ INT | 3/ 2/ 0 | Бит установки флага SOCB/SOCA/INT в регистре ETFLG | | | | | | | | | | | | | | | |
| | | 0 | Нет действий | | | | | | | | | | | | | | |
| | | 1 | Запись единицы устанавливает флаг | | | | | | | | | | | | | | |
| - | 31-4, 1 | Зарезервировано | | | | | | | | | | | | | | | |

Таблица А.12.25 – Регистр управления модулятором

| PCCTL | | Сброс: 00000000h | | | | | | | | | | | | | | | | | | |
|---|--|------------------|--|--|--|--|--|--|--|---------|--|--|--|---------|--|--|---------|--|--|--------|
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | CHPDUTY | | | | SHPFREQ | | | OSHTWTH | | | CHP EN |
| | | | | | | | | | | 3 4 | | | | 3 4 | | | 3 4 | | | 3 4 |

Окончание таблицы А.12.25

| Поле | Биты | Описание | |
|---------|-------|--|---------------|
| CHPDUTY | 10-8 | Поле задания скважности второго и последующих импульсов | |
| | | 000 | 1/8 (13,5 %), |
| | | 001 | 2/8 (25,0 %), |
| | | ... | ... |
| | | 110 | 7/8 (87,5 %), |
| | 111 | Зарезервировано | |
| SHPFREQ | 7-5 | Поле выбора делителя частоты синхронизации для задания частоты второго и последующих импульсов | |
| | | 000 | 1 |
| | | 001 | 1/2 |
| | | ... | ... |
| | | 110 | 1/7 |
| | 111 | 1/8 | |
| OSHTWTH | 4-1 | Поле задания ширины первого импульса | |
| | | 0h | 1 × fclk/8 |
| | | 1h | 2 × fclk/8 |
| | | ... | ... |
| | | Eh | 15 × fclk/8 |
| | Fh | 16 × fclk/8 | |
| CHPEN | 0 | Бит разрешения работы модулятора | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| – | 31-11 | Зарезервировано | |

Таблица А.12.26 – Регистр конфигурации блока ШИМ высокого разрешения

| Поле | Биты | Описание | |
|-----------------|-------|--|---|
| HRCNFG | | Сброс: 00000000h | |
| | | | |
| MEP_ScaleFactor | 23-16 | Не используется | |
| HRLOAD | 3 | Бит выбора события, по которому производится загрузка отложенного значения в активный регистр CMPANR | |
| | | 0 | Событие CTR = Zero |
| | | 1 | Событие CTR = PRD. При этом режим отложенной загрузки доступен, только если бит CTLMODE сброшен, а загрузка поля CMPA осуществляется аналогичным образом (в регистре CMPCTL поле LOADMODE = 00/01) |

Окончание таблицы А.12.26

| Поле | Биты | Описание |
|---------|----------------|--|
| CTLMODE | 2 | Бит выбора регистра для задания значение задержки |
| | | 0 Поле CMPAHR |
| | | 1 Поле ТВPHSHR |
| EDGMODE | 1-0 | Поле выбора фронта сигнала ШИМ, который сдвигается линией задержки |
| | | 00 Зарезервировано |
| | | 01 Передний |
| | | 10 Задний |
| | | 11 Передний и задний |
| – | 31-24, 15-4 | Зарезервировано |

Таблица А.12.27 – Регистр ширины фильтрации

| Поле | Биты | Описание |
|--------------|------|--|
| FWDTH | | Сброс: 00000000h |
| | | |
| FWDTH | 7-0 | Поле задания ширины фильтрации коротких импульсов в тактах PCLK. Если $f_{PCLK} = 100$ МГц, то фильтруются импульсы до 2,55 мкс с шагом 0,01 мкс |
| | 0h | Фильтр выключен |
| | 1h | 1 такт PCLK |
| | ... | ... |
| | FFh | 255 тактов PCLK |
| – | 31-8 | Зарезервировано |

Таблица А.12.28 – Регистр источника сигнала события удержания

| HDSEL | | Сброс: 00000000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|----|------------------|-----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OSH | | | CBC | - | A | A | A | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | |
| T | | - | | | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | |
| | | | | | 2 | 1 | 0 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | 3ч | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.12.28

| Поле | Биты | Описание |
|---------------------------|------------------|--|
| OSHT | 31 | Бит разрешения события по источнику DCMP/АСМР в однократном режиме обработки аварии |
| | | 0 Запрещено |
| | | 1 Разрешено |
| CVC | 28 | Бит разрешения события по источнику DCMP/АСМР в циклическом режиме обработки аварии |
| | | 0 Запрещено |
| | | 1 Разрешено |
| АСМР2, АСМР1, АСМР0 | 26, 25, 24 | Биты включения аналоговых компараторов 0, 1 и 2 |
| | | 0 Выключен |
| | | 1 Включен |
| DCMP23– DCMP0 | 23-0 | Бит выбора цифрового компаратора (0 – 23) блока АЦП, с выхода которого берется сигнал для формирования события удержания |
| | | 0 Не выбран |
| | | 1 Выбран |
| – | 30, 29, 27 | Зарезервировано |

Таблица А.12.29 – Регистр управления детектором событий удержания

| Поле | Биты | Описание |
|--|-------------|---|
| | | Сброс: 00000000h |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border: 1px solid black; height: 40px; margin: 5px 0;"> <div style="flex-grow: 1;"></div> <div style="border: 1px solid black; padding: 2px; text-align: center;">HDB</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">HDA</div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 ч3 ч </div> | | |
| HDB/ HDA | 3-2/ 1-0 | Поле задания поведения сигнала PWMB/PWMA в случае сбоя (аварии). (Источник сбоя определяется регистром HDSEL) |
| | | 00 Зарезервировано |
| | | 01 Переключается в состояние единицы |
| | | 10 Переключается в состояние нуля |
| | | 11 Остается без изменений |
| – | 31-4 | Зарезервировано |

Таблица А.12.30 – Регистр маски прерывания порогового выключателя

| HDEINT | | Сброс: 00000000h |
|---|---------|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 2px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 20px; margin: 2px 0;"></div> | | - |
| 3 ч | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 2px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 20px; margin: 2px 0;"></div> | | - |
| | | OST CBC - |
| | | 3 ч 3 ч |
| Поле | Бит | Описание |
| OST | 2 | Бит разрешения генерации прерывания в однократном режиме |
| | | 0 Запрещено |
| | | 1 Разрешено |
| CBC | 1 | Бит разрешения генерации прерывания в циклическом режиме |
| | | 0 Запрещено |
| | | 1 Разрешено |
| – | 31-3, 0 | Зарезервировано |

Таблица А.12.31 – Регистр флагов прерывания порогового выключателя

| HDFLG | | Сброс: 00000000h |
|---|---------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 2px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 20px; margin: 2px 0;"></div> | | - |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 2px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 20px; margin: 2px 0;"></div> | | - |
| | | OST CBC INT |
| | | 3 ч 3 ч 3 ч |
| Поле | Бит | Описание |
| OSHT, CBC | 2, 1 | Флаг прерывания в однократном режиме, флаг прерывания в циклическом режиме |
| | | 0 Нет прерывания |
| | | 1 Запрос на прерывание |
| | | При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова |
| INT | 0 | Флаг внешнего прерывания NVIC |
| | | 0 Нет прерывания |
| | | 1 Запрос на прерывание |
| | | Если флаг был сброшен, а один из флагов CBC или OST установлен, флаг установится снова |
| – | 31-3 | Зарезервировано |

Таблица А.12.32 – Регистр сброса флагов порогового выключателя

| HDCLR | | Сброс: 00000000h |
|---|------|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0; display: flex; align-items: center; justify-content: center;"> - <div style="border: 1px solid black; padding: 2px;">OST</div> <div style="border: 1px solid black; padding: 2px; margin-left: 5px;">CBC</div> <div style="border: 1px solid black; padding: 2px; margin-left: 5px;">INT</div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 4 3 4 3 4 </div> | | |
| Поле | Бит | Описание |
| OST | 2 | Бит сброса флага прерывания в однократном режиме |
| CBC | 1 | Бит сброса флага прерывания в циклическом режиме |
| INT | 0 | Бит сброса флага внешнего прерывания NVIC |
| – | 31-3 | Зарезервировано |
| Запись единицы в бит регистра сбрасывает соответствующий бит в регистре HDFLG | | |

Таблица А.12.33 – Регистр программной активации порогового выключателя

| HDFRC | | Сброс: 00000000h |
|--|---------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0; display: flex; align-items: center; justify-content: center;"> - <div style="border: 1px solid black; padding: 2px;">OST</div> <div style="border: 1px solid black; padding: 2px; margin-left: 5px;">CBC</div> <div style="border: 1px solid black; padding: 2px; margin-left: 5px;">-</div> </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 3 4 3 4 </div> | | |
| Поле | Бит | Описание |
| OST | 2 | Бит активации порогового выключателя в однократном режиме обработки аварии |
| | | 0 Нет действий |
| | | 1 Запись единицы активирует выключатель и устанавливает флаг OST в регистре HDFLG |
| CBC | 1 | Бит активации порогового выключателя в циклическом режиме обработки аварии |
| | | 0 Нет действий |
| | | 1 Запись единицы активирует выключатель и устанавливает флаг CBC в регистре HDFLG |
| – | 31-3, 0 | Зарезервировано |

Таблица А.12.34 – Регистры сброса прерываний порогового выключателя, детектора событий аварии и таймера блока ШИМ

| HDINTCLR TZINTCLR INTCLR | | Сброс: 00000000h |
|---|------|--|
| | | |
| Поле | Бит | Описание |
| INT | 0 | Бит сброса прерывания. Запись единицы в бит сбрасывает запрос прерывания. Запись в бит должна производиться программой обработки прерывания, во избежание повторного запуска программы обслуживания прерывания |
| – | 31-1 | Зарезервировано |

А.13 Регистры блока захвата

Таблица А.13.1 – Регистр блока захвата

| Мнемоника | Назначение и описание | Сброс |
|---------------|---|-----------|
| TSCTR | 32-разрядный регистр счетчика таймера. Запись задает начальное значение таймера. Чтение возвращает текущее значение таймера | 00000000h |
| CTRPHS | 32-разрядный регистр отложенной загрузки таймера. Значение из регистра загружается в таймер по событиям SYNC1 или под управлением процессора. Регистр используется для синхронизации с другими блоками CAP/PWM | 00000000h |
| CAP0 | 32-разрядный регистр захвата 0. Варианты загрузки: - значением таймера в режиме захвата; - значением, задаваемым микропроцессором; - из регистра отложенной загрузки CAP3 в режиме APWM | 00000000h |
| CAP1 | 32-разрядный регистр захвата 1. Варианты загрузки: - значением таймера в режиме захвата; - значением, задаваемым микропроцессором; - из регистра отложенной загрузки CAP4 в режиме APWM | 00000000h |
| CAP2 | 32-разрядный регистр захвата 2. Варианты загрузки: - значением таймера в режиме захвата; - значением, задаваемым микропроцессором (в тестовых целях). Регистр используется как регистр отложенной загрузки APRD в режиме APWM | 00000000h |
| CAP3 | 32-разрядный регистр захвата 3. Варианты загрузки: - значением таймера в режиме захвата; - значением, задаваемым микропроцессором (в тестовых целях). Регистр используется как регистр отложенной загрузки APRD в режиме APWM | 00000000h |

Таблица А.13.2 – Регистр контроля захвата 0

| ECCTL0 | | | | | | | | | | | | | | Сброс: 00000000h | |
|---------------|----|----|----------|----|----|----|-----------|----------|----------|----------|----------|----------|----------|------------------|----------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FREE/SOFT | | | PRESCALE | | | | CAP LD EN | CTR RST3 | CAP3 POL | CTR RST2 | CAP2 POL | CTR RST1 | CAP1 POL | CTR RST0 | CAP0 POL |
| 3 4 | | | 3 4 | | | | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 |

Окончание таблицы А.13.2

| Поле | Биты | Описание | |
|-----------|---------------|--|-----------------------------------|
| FREE/SOFT | 15-14 | Управление остановкой таймера в режиме отладки | |
| | | 00 | Моментальная остановка таймера |
| | | 01 | Остановка таймера по переполнению |
| | | 10, 11 | Таймер не останавливается |
| PRESCALE | 13-9 | Предварительный делитель. Если записано значение 00h – делитель выключен. | |
| CAPLDEN | 8 | Бит разрешения захвата регистрами CAP0–CAP3 | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| CTRRSTn | 7, 5, 3, 1 | Бит сброса таймера после события n (n от 0 до 3) | |
| | | 0 | Нет действий |
| | | 1 | Сброс таймера после события n |
| CAPnPOL | 6, 4, 2, 0 | Бит выбора фронта захвата (n от 0 до 3) | |
| | | 0 | Захват по переднему фронту |
| | | 1 | Захват по заднему фронту |
| – | 31-16 | Зарезервировано | |

Таблица А.13.3 – Регистр контроля захвата 1

| Поле | Бит | Описание |
|---|-----|--|
| ECCTL1 | | Сброс: 00000000h |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; width: 100%;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> APWM POLCAP/ APWMSW SYNCSYNCO_SELSYNCl _ENTSC TR STOPRE ARMSTOP_WRAPCONT/ ONE SHT </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 3333333333 </div> | | |
| APWMPOL | 10 | Бит задания активного уровня в режиме APWM 0 Высокий 1 Низкий |
| CAP/APWM | 9 | Бит выбора режима 0 Работа в режиме захвата. Блокирование сброса таймера по событию CTR=PRD. Блокирование отложенной загрузки CAP0/1. Разрешение захвата CAP0–CAP3. Внешний порт работает на вход. 1 Работа в режиме APWM. Разрешение сброса таймера по CTR = PRD. Разрешение теневой загрузки CAP0/1. Блокирование захвата CAP0–CAP3. Внешний порт работает на выход |
| SWSYNC | 8 | Межблочная синхронизация таймеров 0 Нет действий 1 Запись единицы: - загружает значение таймера из отложенного регистра при условии, что установлен бит SYNCl_EN; - генерирует выходной сигнал синхронизации при условии, что в поле SYNC_SEL записано 00b или 01b. Примечание – В режиме APWM синхронизация также происходит автоматически по событию CTR = PRD. |

Окончание таблицы А.13.3

| Поле | Биты | Описание | |
|-----------------|---------------|--|---|
| SYNCO_SEL | 7-6 | Выбор источника выходного синхросигнала | |
| | | 00 | Пропуск сигнала синхронизации с входа на выход |
| | | 01 | Передача события CTR = PRD в качестве выходного сигнала синхронизации |
| | | 10, 11 | Запрет выходного сигнала синхронизации |
| SYNCI_EN | 5 | Бит разрешения синхронизации | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| TSCTRSTOP | 4 | Бит управления работой таймера | |
| | | 0 | Остановлен |
| | | 1 | Запущен |
| REARM | 3 | Запись единицы запускает следующую последовательность действий: сброс управляющего контроллера, разрешение работы управляющего контроллера, и загрузку регистров захвата | |
| STOP_WRAP | 2-1 | Значение компаратора остановки в режимах захвата | |
| | | 00 | Останов при значении счетчика 00 |
| | | 01 | Останов при значении счетчика 01 |
| | | 10 | Останов при значении счетчика 10 |
| | | 11 | Останов при значении счетчика 11 |
| | | Примечание – Остановка управляющего контроллера приводит также к блокировке загрузки регистров захвата. | |
| CONT/ ONESHT | 0 | Режим работы захвата | |
| | | 0 | Циклический |
| | | 1 | Однократный |
| – | 31-16, 7-3 | Зарезервировано | |

Таблица А.13.4 – Регистр маски прерываний

| Поле | Бит | Описание |
|--|------------|--|
| ECEINT | | Сброс: 00000000h |
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | |
| | | CTR=CMP CTR=PRD CTR=OVF CEVT3 CEVT2 CEVT1 CETV0 |
| | | 3 4 3 4 3 4 3 4 3 4 3 4 |
| CTR=CMP | 7 | Бит разрешения генерации прерывания по событию CTR = CMP |
| CTR=PRD | 6 | Бит разрешения генерации прерывания по событию CTR = PRD |
| CTROVF | 5 | Бит разрешения генерации прерывания по событию CTROVF |
| CEVTn | 4-1 | Бит разрешения генерации прерывания по событию VTn (n от 0 до 3) |
| – | 31-8, 0 | Зарезервировано |
| Для всех бит: установленный бит разрешает прерывание, а сброшенный – запрещает | | |

Таблица А.13.5 – Регистр статуса прерываний

| ECFLG | | Сброс: 00000000h | | | | | | | | | | | | | |
|---|------|--|----|----|----|----|----|-------------|-------------|------------|-----------|-----------|-----------|-----------|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | CTR= CMP | CTR= PRD | CTR OVF | CEV T3 | CEV T2 | CEV T1 | CET V0 | INT |
| | | | | | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| CTR=CMP | 7 | Флаг прерывания по событию CTR = CMP | | | | | | | | | | | | | |
| | | 0 Событие не произошло | | | | | | | | | | | | | |
| | | 1 Событие произошло | | | | | | | | | | | | | |
| CTR=PRD | 6 | Флаг прерывания по событию CTR = PRD | | | | | | | | | | | | | |
| | | 0 Событие не произошло | | | | | | | | | | | | | |
| | | 1 Событие произошло | | | | | | | | | | | | | |
| CTROVF | 5 | Флаг прерывания по событию CTROVF | | | | | | | | | | | | | |
| | | 0 Событие не произошло | | | | | | | | | | | | | |
| | | 1 Событие произошло | | | | | | | | | | | | | |
| CEVTn | 4-1 | Флаг прерывания по событию CEVTn (n от 0 до 3) | | | | | | | | | | | | | |
| | | 0 Событие не произошло | | | | | | | | | | | | | |
| | | 1 Событие произошло | | | | | | | | | | | | | |
| INT | 0 | Флаг прерывания | | | | | | | | | | | | | |
| - | 31-8 | Зарезервировано | | | | | | | | | | | | | |
| Примечание – Все флаги сбрасываются записью единиц в биты регистра ECCLR. | | | | | | | | | | | | | | | |

Таблица А.13.6 – Регистр сброса прерываний

| ECCLR | | Сброс: 00000000h | | | | | | | | | | | | | |
|---|----|------------------|----|----|----|----|----|-------------|-------------|------------|-----------|-----------|-----------|-----------|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | CTR= CMP | CTR= PRD | CTR OVF | CEV T3 | CEV T2 | CEV T1 | CET V0 | INT |
| | | | | | | | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Биты с 31 по 8 зарезервированы. Запись единиц в биты с 7 по 0 сбрасывает соответствующие флаги в регистре ECFLG | | | | | | | | | | | | | | | |

Таблица А.13.7 – Регистр программных прерываний

| | | | | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|-------------|-------------|------------|-----------|-----------|-----------|------------------|----|--|--|
| ECFRC | | | | | | | | | | | | | | Сброс: 00000000h | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | |
| - | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| - | | | | | | | | CTR= CMP | CTR= PRD | CTR OVF | CEV T3 | CEV T2 | CEV T1 | CET V0 | - | | |
| | | | | | | | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | | | |
| Биты с 31 по 8 и 0 зарезервированы. Запись единиц в биты с 7 по 1 генерирует прерывания | | | | | | | | | | | | | | | | | |

Таблица А.13.8 – Регистр активного прерывания

| PEINT | | | | | | | | | | | | | | Сброс: 00000000h | | | | |
|--------------|------|--|----|----|----|----|----|----|----|----|----|----|----|------------------|-----------|---|--|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | |
| - | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| - | | | | | | | | | | | | | | | PE INT | | | |
| | | | | | | | | | | | | | | | 3 | 4 | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | |
| PEINT | 0 | Флаг активного прерывания. Устанавливается при возникновении прерывания блока. Сбрасывается только программно, записью единицы | | | | | | | | | | | | | | | | |
| – | 31-1 | Зарезервировано | | | | | | | | | | | | | | | | |

А.14 Регистры квадратурного декодера

Таблица А.14.1 – 32-разрядные регистры

| Мнемоника | Назначение и описание | Сброс |
|-----------------|--|-----------|
| QPOSCNT | Регистр счета счетчика позиции. Доступен только для чтения | 00000000h |
| QPOSINIT | Регистр инициализации счетчика позиции | 00000000h |
| QPOSMAX | Регистр максимального значения счетчика позиции | 00000000h |
| QPOSCMP | Регистр сравнения счетчика позиции | 00000000h |
| QPOSILAT | Регистр хранения позиции по индексации Доступен только для чтения | 00000000h |
| QPOSSLAT | Регистр хранения позиции по стробу Доступен только для чтения | 00000000h |
| QPOSLAT | Регистр хранения позиции по таймеру временных отсчетов. Доступен только для чтения | 00000000h |
| QUTMR | Регистр таймера временных отсчетов Доступен только для чтения | 00000000h |
| QUPRD | Регистр длительности счета таймера временных отсчетов | 00000000h |
| QWDTMR | Регистр счета сторожевого таймера Доступен только для чтения | 00000000h |
| QWDPRD | Регистр длительности счета сторожевого таймера | 00000000h |
| QCTMR | Регистр таймера блока захвата | 00000000h |
| QCPRD | Регистр длительности измерения блока захвата | 00000000h |
| QCTMRLAT | Регистр хранения таймера блока захвата. Доступен только для чтения | 00000000h |
| QCPRDLAT | Регистр хранения длительности измерения блока захвата | 00000000h |
| INTCLR | Регистр сброса прерываний декодера | 00000000h |

Таблица А.14.2 – Регистр управления входами

| QDECCTL | | | | | | | | | | | Сброс: 00000000h | | | | | | | | | | |
|---------|-------|--------------|---------------------------------|--------|----|-----|----|-------|----|--------|------------------|-----|----|-----|----|-----|--|-----|--|---|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
| QSRC | | SO EN | | SP SEL | | XCR | | SW AP | | I GATE | | QAP | | QBP | | QIP | | QSP | | - | |
| 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | | | | | | | |
| QSRC | 15-14 | Режим работы | | | | | | | | | | | | | | | | | | | |
| | | 00 | Квадратурный | | | | | | | | | | | | | | | | | | |
| | | 01 | Счета/направления | | | | | | | | | | | | | | | | | | |
| | | 10 | Счет вверх (QCLK=xCLK, QDIR=1), | | | | | | | | | | | | | | | | | | |
| | | 11 | Счет вниз (QCLK=xCLK, QDIR=0). | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.14.2

| Поле | Биты | Описание |
|---|------------|--|
| SOEN | 13 | Бит разрешения выдачи выходного сигнала компаратора |
| | | 0 Запрещено |
| | | 1 Разрешено |
| SPSEL | 12 | Бит выбора вывода для выдачи выходного сигнала компаратора |
| | | 0 Стrobeирующий вывод |
| | | 1 Индексный вывод |
| XCR | 11 | Бит выбора фронта квадратурного входа |
| | | 0 Передний фронт |
| | | 1 Передний и задний фронты |
| SWAP | 10 | Бит обмена входов QEPА и QEPВ |
| | | 0 Нет действий |
| | | 1 Входы QEPА и QEPВ меняются местами |
| IGATE | 9 | Бит включения стробирования входного сигнала индексации |
| QAP | 8 | Бит включения инвертирования входного сигнала с QEPА |
| QBP | 7 | Бит включения инвертирования входного сигнала с QEPВ |
| QIP | 6 | Бит включения инвертирования входного сигнала с QEPІ |
| QSP | 5 | Бит включения инвертирования входного сигнала с QEPS |
| – | 31-16, 4-0 | Зарезервировано |
| Примечание – Для битов с 9 по 5 справедливо: 0 – выключено, 1 – включено. | | |

Таблица А.14.3 – Регистр управления квадратурного декодера

| Поле | Биты | Описание |
|--|-------|---|
| QEPCTL | | Сброс: 00000000h |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> FREE/SOFTPCRMSEIIEISWISELIELQP ENQC LMUTEWDE </div> <div style="display: flex; justify-content: space-between; padding: 0 10px;"> 3 43 43 43 43 43 43 43 43 43 43 4 </div> | | |
| FREE/SOFT | 15-14 | Поле управления счетчиками QPOSCNT, QWDTMR, QUTMR, QCTMR в режиме отладки |
| | | 00 Принудительная блокировка счета |
| | | 01 Счет до переполнения |
| | | 10, 11 Разблокирование счета |
| PCRM | 13-12 | Поле задания события для сброса счетчика позиции |
| | | 00 Событие индексации |
| | | 01 Достижение максимальной позиции |
| | | 10 Первое событие индексации |
| | | 11 Окончание временного отсчета |

Окончание таблицы А.14.3

| Поле | Биты | Описание | |
|------|-------|--|--|
| SEI | 11-10 | Поле задания события стробирования для инициализации счетчика позиции (QPOSCNT = QPOSINIT) | |
| | | 00, 01 | Работа без инициализации |
| | | 10 | Передний фронт сигнала QEPS |
| | | 11 | Передний фронт QEPS при вращении по часовой стрелке или задний фронт QEPS при вращении против часовой стрелки |
| IEI | 9-8 | Поле задания события индексации для инициализации счетчика позиции (QPOSCNT = QPOSINIT) | |
| | | 00, 01 | Работа без инициализации |
| | | 10 | По переднему фронту сигнала QEPI |
| | | 11 | По заднему фронту сигнала QEPI |
| SWI | 7 | Бит программной инициализации счетчика позиции. Не сбрасывается аппаратно | |
| | | 0 | Нет действий |
| | | 1 | Запись единицы загружает счетчик позиции QPOSCNT значением QPOSINIT |
| SEL | 6 | Бит задания события стробирования для сохранения значения счетчика позиции (QPOSSLAT = POSCNT) | |
| | | 0 | По переднему фронту QEPS |
| | | 1 | По переднему фронту QEPS при вращении по часовой стрелке или по заднему фронту QEPS при вращении против часовой стрелки |
| IEL | 5-4 | Поле задания события индексации для сохранения значения счетчика позиции (QPOSILAT = POSCNT) | |
| | | 00 | Без сохранения |
| | | 01 | По переднему фронту сигнала индексации |
| | | 10 | По заднему фронту сигнала индексации |
| | | 11 | По маркеру индексации |
| QPEN | 3 | Бит разрешения работы счетчика позиции | |
| | | 0 | Запись нуля останавливает счетчик и сбрасывает его |
| | | 1 | Работа разрешена |
| QCLM | 2 | Бит задания события сохранения значения регистров модуля захвата | |
| | | 0 | По чтению QPOSCNT регистры QCTMR и QCPRD сохраняются в регистры QCTMRLAT и QCPRDLAT соответственно. |
| | | 1 | По окончанию временного отсчета регистры QPOSCNT, QCTMR и QCPRD сохраняются в регистры QPOSLAT, QCTMRLAT и QCPRDLAT соответственно |
| UTE | 1 | Бит разрешения работы таймера временных отсчетов | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| WDE | 0 | Бит разрешения работы сторожевого таймера | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| — | 31-16 | Зарезервировано | |

Таблица А.14.4 – Регистр захвата

| QCAPCTL | | Сброс: 00000000h |
|--|----------------|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="text-align: center; margin: 5px 0;">-</div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid black; padding-bottom: 5px;"> <div style="border-right: 1px solid black; padding: 0 5px;">CEN</div> <div style="border-right: 1px solid black; padding: 0 5px;">-</div> <div style="border-right: 1px solid black; padding: 0 5px;">SEL EV NT</div> <div style="border-right: 1px solid black; padding: 0 5px;">CCPS</div> <div style="padding: 0 5px;">UPPS</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> 3 ч3 ч3 ч3 ч </div> | | |
| Поле | Биты | Описание |
| CEN | 15 | Бит разрешения работы модуля захвата времени |
| | | 0 Запрещено |
| | | 1 Разрешено |
| SELEVNT | 7 | Бит сброса таймера |
| | | 0 По деленному квадратурному событию |
| | | 1 По получении сигнала PCSOUT от Компаратора |
| CCPS | 6-4 | Поле задания делителя системного такта |
| | | 000 Нет деления |
| | | 001 1/2 |
| | | 010 1/4 |
| | | 011 1/8 |
| | | 100 1/16 |
| | | 101 1/32 |
| | | 110 1/64 |
| | | 111 1/128 |
| UPPS | 3-0 | Поле задания делителя квадратурного сигнала |
| | | 0h Нет деления |
| | | 1h 1/2 |
| | | 2h 1/4 |
| | | 3h 1/8 |
| | | 4h 1/16 |
| | | 5h 1/32 |
| | | 6h 1/64 |
| | | 7h 1/128 |
| | | 8h 1/256 |
| | | 9h 1/512 |
| | | Ah 1/1024 |
| | | Bh 1/2048 |
| | | Ch-Fh Зарезервировано |
| - | 31-16, 14-8 | Зарезервировано |

Таблица А.14.5 – Регистр управления счетчиком позиции

| QPOSCTL | | Сброс: 00000000h | | | | | | | | | |
|--|-------|--|--|----|----|----|------|----|-----|-----|-------|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="text-align: center; margin: 5px 0;">-</div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid black; padding-bottom: 5px;"> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PC</td></tr> <tr><td>SH</td></tr> <tr><td>DW</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PC</td></tr> <tr><td>LOAD</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PC</td></tr> <tr><td>POL</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PCE</td></tr> </table> </div> <div style="text-align: center; flex-grow: 1;"> <table border="1" style="font-size: 8px;"> <tr><td>PCSPW</td></tr> </table> </div> </div> <div style="display: flex; justify-content: space-between; font-size: 8px; margin-top: 5px;"> 3 43 43 43 43 4 </div> | | | PC | SH | DW | PC | LOAD | PC | POL | PCE | PCSPW |
| PC | | | | | | | | | | | |
| SH | | | | | | | | | | | |
| DW | | | | | | | | | | | |
| PC | | | | | | | | | | | |
| LOAD | | | | | | | | | | | |
| PC | | | | | | | | | | | |
| POL | | | | | | | | | | | |
| PCE | | | | | | | | | | | |
| PCSPW | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | |
| PCSHDW | 15 | Бит разрешения режима отложенной загрузки | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | |
| PCLOAD | 14 | Бит выбора события загрузки в режиме отложенной записи | | | | | | | | | |
| | | 0 | Загрузка отложенного значения в активный регистр по событию QPOSCNT = 0. | | | | | | | | |
| | | 1 | Загрузка по QPOSCNT = QPOSCMP | | | | | | | | |
| PCPOL | 13 | Бит выбора полярности выхода синхронизации | | | | | | | | | |
| | | 0 | Активная единица | | | | | | | | |
| | | 1 | Активный ноль | | | | | | | | |
| PCE | 12 | Бит разрешения работы компаратора | | | | | | | | | |
| | | 0 | Запрещено | | | | | | | | |
| | | 1 | Разрешено | | | | | | | | |
| PCSPW | 11-0 | Поле задания ширины импульса выхода синхронизации | | | | | | | | | |
| | | 000h | Отсутствие импульса | | | | | | | | |
| | | 001h | 2 × P | | | | | | | | |
| | | ... | ... | | | | | | | | |
| | | 007h | 8 × P | | | | | | | | |
| | | ... | ... | | | | | | | | |
| | | FFFh | 4096 × P | | | | | | | | |
| P – период системного тактового сигнала | | | | | | | | | | | |
| – | 31-16 | Зарезервировано | | | | | | | | | |

Таблица А.14.6 – Регистр масок прерываний

| QEINT | | Сброс: 00000000h | | | | | | | | | | | |
|--|--|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="text-align: center; margin: 5px 0;">-</div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid black; padding-bottom: 5px;"> <div style="text-align: center;">-</div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>UTO</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>IEL</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>SEL</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PCM</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PCR</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PCO</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PCU</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>WTO</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>QDC</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>QPE</td></tr> </table> </div> <div style="text-align: center;"> <table border="1" style="font-size: 8px;"> <tr><td>PCE</td></tr> </table> </div> <div style="text-align: center;">-</div> </div> <div style="display: flex; justify-content: space-between; font-size: 8px; margin-top: 5px;"> 3 43 43 43 43 43 43 43 43 43 43 43 43 4 </div> | | | UTO | IEL | SEL | PCM | PCR | PCO | PCU | WTO | QDC | QPE | PCE |
| UTO | | | | | | | | | | | | | |
| IEL | | | | | | | | | | | | | |
| SEL | | | | | | | | | | | | | |
| PCM | | | | | | | | | | | | | |
| PCR | | | | | | | | | | | | | |
| PCO | | | | | | | | | | | | | |
| PCU | | | | | | | | | | | | | |
| WTO | | | | | | | | | | | | | |
| QDC | | | | | | | | | | | | | |
| QPE | | | | | | | | | | | | | |
| PCE | | | | | | | | | | | | | |

Окончание таблицы А.14.6

| Поле | Бит | Описание |
|------|----------|--|
| UTO | 11 | Бит разрешения прерывания по срабатыванию таймера временных отсчетов |
| IEL | 10 | Бит разрешения прерывания по событию индексации |
| SEL | 9 | Бит разрешения прерывания по событию стробирования |
| PCM | 8 | Бит разрешения прерывания по срабатыванию компаратора |
| PCR | 7 | Бит разрешения прерывания по готовности компаратора к загрузке значения сравнения из отложенного регистра |
| PCO | 6 | Бит разрешения прерывания при достижении счетчиком позиции максимального значения QPOS MAX при счете вверх |
| PCU | 5 | Бит разрешения прерывания при достижении счетчиком позиции минимального значения при счете вниз |
| WTO | 4 | Бит разрешения прерывания при срабатывании сторожевого таймера |
| QDC | 3 | Бит разрешения прерывания при смене направления вращения |
| QPE | 2 | Бит разрешения прерывания по ошибке фазы на квадратурном входе |
| PCE | 1 | Бит разрешения прерывания счетчика позиции |
| – | 31-12, 0 | Зарезервировано |

Установленный бит разрешает генерирование соответствующего прерывания, сброшенный – запрещает

Таблица А.14.7 – Регистр флагов прерываний

| Поле | Бит | Описание |
|------|-----|--|
| UTO | 11 | Флаг прерывания по срабатыванию таймера временных отсчетов |
| IEL | 10 | Флаг прерывания по событию индексации |
| SEL | 9 | Флаг прерывания по событию стробирования |
| PCM | 8 | Флаг прерывания по срабатыванию компаратора |
| PCR | 7 | Флаг прерывания по готовности компаратора к загрузке значения сравнения из отложенного регистра |
| PCO | 6 | Флаг прерывания при достижении счетчиком позиции максимального значения QPOS MAX при счете вверх |
| PCU | 5 | Флаг прерывания при достижении счетчиком позиции минимального значения при счете вниз |
| WTO | 4 | Флаг прерывания при срабатывании сторожевого таймера |
| QDC | 3 | Флаг прерывания при смене направления вращения |
| QPE | 2 | Флаг прерывания по ошибке фазы на квадратурном входе |
| PCE | 1 | Флаг прерывания ошибки счетчика позиции |

Окончание таблицы А.14.7

| Поле | Биты | Описание |
|------|-------|--|
| INT | 0 | Флаг выходного прерывания блока квадратурного декодера |
| – | 31-12 | Зарезервировано |

Установленный бит является индикатором запроса соответствующего прерывания.
Сброс флагов прерываний осуществляется посредством регистра QCLR

Таблица А.14.8 – Регистр сброса флагов прерываний

| QCLR | | | | | | | | | | | | | | Сброс: 00000000h | |
|------|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------------|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | UTO | IEL | SEL | PCM | PCR | PCO | PCU | WTO | QDC | QPE | PCE | INT |
| | | | | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 |

Запись единицы в бит сбрасывает соответствующий флаг прерывания в регистре QFLG.
Биты с 31 по 12 зарезервированы

Таблица А.14.9 – Регистр эмуляции прерываний

| QFRC | | | | | | | | | | | | | | Сброс: 00000000h | |
|------|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------------|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | UTO | IEL | SEL | PCM | PCR | PCO | PCU | WTO | - | QPE | PCE | - |
| | | | | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 | 3 4 |

Запись единицы в бит устанавливает соответствующий флаг прерывания в регистре QFLG. Биты с 31 по 12, 3 и 0 зарезервированы

Таблица А.14.10 – Регистр статуса

| QEPSTS | | | | | | | | | | | | | | Сброс: 00000000h | |
|--------|----|----|----|----|----|----|----|----------------|------|-----|------|------|------|------------------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | UP EV NT | FIDF | QDF | QDLF | COEF | CDEF | FIMF | PCEF |
| | | | | | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |

Окончание таблицы А.14.10

| Поле | Бит | Описание |
|--------|------|---|
| UPEVNT | 7 | Флаг сброса QCTMR и обновления QCPRD |
| | | 0 Нет событий |
| | | 1 Зафиксировано событие сброса и обновления |
| | | Сбрасывается записью 1. |
| FIDF | 6 | Индикатор направления вращения по событию первого импульса индексации |
| | | 0 Против часовой стрелки (счет вниз) |
| | | 1 По часовой стрелке (счет вверх) |
| | | Доступен только для чтения |
| QDF | 5 | Флаг направления вращения. Обновляется по каждому событию на входах квадратур |
| | | 0 Вращение вала ротора против часовой стрелки |
| | | 1 Вращение вала ротора по часовой стрелке. |
| | | Доступен только для чтения. |
| QDLF | 4 | Флаг направления вращения. Обновляется по каждому сигналу индексации |
| | | 0 Вращение вала ротора против часовой стрелки |
| | | 1 Вращение вала ротора по часовой стрелке. |
| | | Доступен только для чтения. |
| COEF | 3 | Флаг ошибки переполнения счетчика QCTMR модуля захвата |
| | | 0 Ошибка отсутствует |
| | | 1 Произошло переполнение |
| | | Сбрасывается записью 1 |
| CDEF | 2 | Флаг ошибки изменения направления вращения вала ротора между двумя событиями UPEVNT |
| | | 0 Ошибка отсутствует |
| | | 1 Произошло изменение направления вращения во время измерения |
| | | Сбрасывается записью 1 |
| FIMF | 1 | Флаг приема первого импульса сигнала индексации |
| | | 0 Импульсов нет, либо первый импульс уже был принят |
| | | 1 Принят первый импульс сигнала индексации |
| | | Сбрасывается записью 1 |
| PCEF | 0 | Флаг ошибки счетчика позиции. Обновляется по каждому сигналу индексации |
| | | 0 Во время последнего сигнала индексации ошибки не возникло |
| | | 1 Ошибка счетчика позиции |
| | | Доступен только для чтения. |
| – | 31-8 | Зарезервировано |

А.15 Регистры загрузочной флеш-памяти

Таблица А.15.1 – Регистр адреса загрузочной флеш

| FMA | | A001_C000h | Сброс: 00000000h | | | | | | | | | | | | |
|---------------|-------|---|------------------|----|----|----|----|----|----|----|----|---------------|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | BOOTFLASHADDR | | | |
| 3 ч | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BOOTFLASHADDR | | | | | | | | | | | | | | | |
| 3 ч | | | | | | | | | | | | | | | |
| Поле | Биты | Описание | | | | | | | | | | | | | |
| BOOTFLASHADDR | 19-0 | Адрес, используемый при командах записи и постраничного стирания. Должен быть записан до установки бита выполнения команды записи/стирания. Примечания 1 При команде записи младшие 4 бита не имеют значения. 2 При постраничном стирании младшие 13 бит не имеют значения. | | | | | | | | | | | | | |
| — | 31-20 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.15.2 – Регистры слов данных загрузочной флеш-памяти

| | | |
|-------------|------------|------------------|
| FMD0 | A001_C004h | Сброс: 00000000h |
| FMD1 | A001_C050h | Сброс: 00000000h |
| FMD2 | A001_C054h | Сброс: 00000000h |
| FMD3 | A001_C058h | Сброс: 00000000h |

32-разрядные регистры слов данных, используемые при выполнении команд записи через регистр FMC. Все четыре слова данных должны быть загружены в регистры до установки бита команды записи

Таблица А.15.3 – Регистр команд загрузочной флеш

| | | | | | | | | | | | | | | | |
|------------|----|------------|------------------|----|----|----|----|----|----|-----------------------|-------------------|----|---------------------|---------------------|-----------|
| FMC | | A001_C008h | Сброс: 00000000h | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MAGIC_KEY | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | | PAGE ERAS E_IFB | WRI TE_ IFB | - | FULL _ERA _SE | PAGE _ERA _SE | WRI TE |
| | | | | | | | | | | 3 | 3 | | 3 | 3 | 3 |

Окончание таблицы А.15.3

| Поле | Бит | Описание |
|---------------|------------|---|
| MAGIC_KEY | 31-16 | Код запуска команды. Все команды для вступления в силу должны сопровождаться записью в поле MAGIC_KEY значения A442h. Команды должны выполняться по одной, т. е. запись следующей команды разрешена, только после завершения предыдущей. Одновременная запись нескольких команд приведет к невыполнению ни одной из них и выдаст флаг ошибки. Чтение поля MAGIC_KEY всегда возвращает 0000h |
| PAGEERASE_IFB | 5 | Бит постраничного стирания информационного блока. Данные в регистры FMD1-FMD4 должны быть записаны до установки бита PAGEERASE_IFB. Адрес не важен, поскольку в информационном блоке всего одна страница. Установка бита PAGEERASE_IFB активирует команду постраничного стирания в информационном блоке загрузочной флеш. По окончании выполнения команды бит сбрасывается автоматически |
| WRITE_IFB | 4 | Бит записи в информационный блок. Данные в регистры FMD1-FMD4 и адрес в регистр FMA (разряды с 12 по 4) должны быть записаны до установки бита WRITE_IFB. Установка бита WRITE_IFB активирует команду записи в информационный блок загрузочной флеш. По окончании выполнения команды бит сбрасывается автоматически |
| FULL_ERASE | 2 | Бит стирания основного блока. Установка бита FULL_ERASE активирует команду полного стирания основного блока загрузочной флеш. По окончании выполнения команды бит сбрасывается автоматически |
| PAGE_ERASE | 1 | Бит постраничного стирания основного блока. Данные в регистры FMD1-FMD4 и адрес в регистр FMA (разряды с 19 по 13) должны быть записаны до установки бита PAGE_ERASE. Установка бита PAGE_ERASE активирует команду постраничного стирания в основном блоке загрузочной флеш. По окончании выполнения команды бит сбрасывается автоматически |
| WRITE | 0 | Бит записи в основной блок. Данные в регистры FMD1-FMD4 и адрес в регистр FMA (разряды с 19 по 4) должны быть записаны до установки бита WRITE. Установка бита WRITE активирует команду записи в основной блок загрузочной флеш. По окончании выполнения команды бит сбрасывается автоматически |
| – | 15-6, 3 | Зарезервировано |

Таблица А.15.4 – Регистр статуса загрузочной флеш

| FCIS | | A001_C00Ch | Сброс: 00000000h |
|----------|------|--|------------------|
| | | | |
| Поле | Бит | Описание | |
| OP_ERROR | 1 | Флаг ошибки записи. Устанавливается в случае попытки стирания блок защищенного от записи, попытки записи в не существующий блок либо чтения из несуществующего блока флеш. Флаг сбрасывается записью единицы в соответствующий бит регистра FCIS | |
| OP_CMLT | 0 | Флаг завершения операции. Устанавливается по завершении операции чтения/записи/стирания. Флаг сбрасывается записью единицы в соответствующий бит регистра FCIS | |
| – | 31-2 | Зарезервировано | |

Таблица А.15.5 – Регистр маски прерываний загрузочной флеш

| FCIM | | A001_C010h | Сброс: 00000000h |
|--------------|------|---|------------------|
| | | | |
| Поле | Бит | Описание | |
| MASK_OP_CMLT | 0 | Бит разрешения генерирования прерывания по завершении операции чтения/записи/стирания | |
| | | 0 | Запрещено |
| | | 1 | Разрешено |
| – | 31-1 | Зарезервировано | |

Таблица А.15.6 – Регистр сброса флагов статуса загрузочной флеш

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|------------|--|---|--|--|--|--|--|--|--|--|--|--|-------------------|--|---|------------|--|-------------------------|--|--|--|--|--|--|--|--|--|
| FCIS | | | | | | | | | | | | | | | A001_C014h | | | | | Сброс: 00000000h | | | | | | | | | |
| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 | | | | | | | | | | | | | | | - | | | | | | | | | | | | | | |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 | | | | | | | | | | | | | | | 1 | | 0 | CLR_OP_ERR | | CLR_OP_CMLT | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | 3 | | 3 | | | | | | | | | |
| Поле | | Бит | | Описание | | | | | | | | | | | | | | | | | | | | | | | | | |
| CLR_OP_ERR | | 1 | | Бит сброса флага ошибки записи. Запись единицы сбрасывает флаг OP_ERROR в регистре FCIS | | | | | | | | | | | | | | | | | | | | | | | | | |
| CLR_OP_CMLT | | 0 | | Бит сброса флага завершения операции. Запись единицы сбрасывает флаг OP_CMLT в регистре FCIS | | | | | | | | | | | | | | | | | | | | | | | | | |
| – | | 31-2 | | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.15.7 – Регистры временных параметров загрузочной флеш

| Мнемоника | Назначение параметра, задаваемого регистром | Значение по сбросу | Диапазон значений |
|---------------|---|--------------------|------------------------|
| T_ACC | Используется в транзакциях чтения из флеш-памяти, задает задержку от установки адреса, до считывания данных из флеш-памяти | 4 такта | 4 такта и более |
| T_NVS | Используется в транзакциях записи и стирания, задает задержку поднятия сигнала NVSTR после поднятия сигнала PROG | 500 тактов | 500 тактов и более |
| T_NVH | Используется в транзакциях записи и стирания, задает задержку отпускания сигнала NVSTR после отпускания сигнала PROG | | |
| T_RCV | Используется в транзакциях записи и стирания, задает задержку от отпускания сигнала NVSTR до поднятия его на следующей транзакции | 100 тактов | 100 тактов и более |
| T_PGS | Используется в транзакциях записи, задает задержку поднятия сигнала YE после поднятия сигнала NVSTR | 1000 тактов | 1000 тактов и более |
| T_PROG | Используется в транзакциях записи, задает длительность сигнала YE в транзакциях записи | 2000 тактов | От 2000 до 4000 тактов |
| T_PGH | Используется в транзакциях записи, задает задержку отпускания сигнала NVSTR после отпускания сигнала YE | 2 такта | 2 такта и более |

Окончание таблицы А.15.7

| Мнемоника | Назначение параметра, задаваемого регистром | Значение по сбросу | Диапазон значений |
|--|--|--------------------|------------------------------|
| T_ERASE | Используется в транзакциях стирания, задает длительность сигнала ERASE в транзакциях стирания | 2000000 тактов | От 2000000 до 4000000 тактов |
| T_ME | Используется в транзакциях массового стирания, задает длительность сигнала ERASE в транзакциях массового стирания | | |
| T_NVH1 | Используется в транзакциях массового стирания, задает задержку отпускания сигнала NVSTR после отпускания сигнала ERASE | 10000 тактов | 10000 тактов и более |
| <p>Примечания</p> <p>1 Все параметры задаются в тактах системного сигнала.</p> <p>2 Мнемоническое название регистра параметра совпадает с названием параметра.</p> | | | |

А.16 Регистры аналогового компаратора

Таблица А.16.1 – Регистр маскированных статусов прерываний

| MIS | | A001_D000h | Сброс: 0000000h | | | | | | | | | | | | |
|------|------|--|-----------------|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | | | | | IN2 | IN1 | IN0 |
| | | | | | | | | | | | | | 3 Ч | 3 Ч | 3 Ч |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| IN2 | 2 | Индикатор установленной маски и статуса прерывания 2 Запись единицы сбрасывает этот бит, а также бит IN2 в регистре ARCIS | | | | | | | | | | | | | |
| IN1 | 1 | Индикатор установленной маски и статуса прерывания 1 Запись единицы сбрасывает этот бит, а также бит IN1 в регистре ARCIS | | | | | | | | | | | | | |
| IN0 | 0 | Индикатор установленной маски и статуса прерывания 0 Запись единицы сбрасывает этот бит, а также бит IN0 в регистре ARCIS | | | | | | | | | | | | | |
| – | 31-3 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.16.2 – Регистр статусов прерываний

| RIS | | A001_D004h | Сброс: 0000000h | | | | | | | | | | | | |
|------|------|---|-----------------|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | | | | | IN2 | IN1 | IN0 |
| | | | | | | | | | | | | | 3 Ч | 3 Ч | 3 Ч |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| IN2 | 2 | Флаг прерывания 2 Флаг сбрасывается битом IN2 регистра ACMIS | | | | | | | | | | | | | |
| IN1 | 1 | Флаг прерывания 1 Флаг сбрасывается битом IN1 регистра ACMIS | | | | | | | | | | | | | |
| IN0 | 0 | Флаг прерывания 0 Флаг сбрасывается битом IN0 регистра ACMIS | | | | | | | | | | | | | |
| – | 31-3 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.16.3 – Регистр маски прерываний

| INTEN | | A001_D008h | Сброс: 0000000h | | | | | | | | | | | | |
|-------|------|------------------------|-----------------|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | | | | | | | | | | | | | IN2 | IN1 | IN0 |
| | | | | | | | | | | | | | 3 4 | 3 4 | 3 4 |
| Поле | Бит | Описание | | | | | | | | | | | | | |
| IN2 | 2 | Бит маски прерывания 2 | | | | | | | | | | | | | |
| IN1 | 1 | Бит маски прерывания 1 | | | | | | | | | | | | | |
| IN0 | 0 | Бит маски прерывания 0 | | | | | | | | | | | | | |
| – | 31-3 | Зарезервировано | | | | | | | | | | | | | |

Таблица А.16.4 – Регистр управления опорным напряжением блока n (n – 0, 1, 2)

| REFCTLn | | | | | | | | | | | | | | | | Сброс: 0000000h |
|---|---------------|--|----|----|----|-----|-----|----|----|----|----|------|----|----|----|-----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| - | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| - | | | | | | EN | RNG | - | | | | VREF | | | | |
| | | | | | | 3 4 | 3 4 | | | | | 3 4 | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | |
| EN | 9 | Поле задания опорного напряжения, в зависимости от выбранного диапазона работы ЦАП | | | | | | | | | | | | | | |
| RNG | 8 | Бит задает диапазон работы ЦАП. Отключает дополнительный делитель напряжения | | | | | | | | | | | | | | |
| VREF | 3-0 | Поле управления выходным напряжением | | | | | | | | | | | | | | |
| – | 31-10, 7-4 | Зарезервировано | | | | | | | | | | | | | | |
| Примечание – Влияние состояний битов EN, RNG и поля VREF на выходное напряжение ЦАП указано в таблице 22.1. | | | | | | | | | | | | | | | | |

Таблица А.16.5 – Регистр статуса компаратора блока n (n – 0, 1, 2)

| ACSTATn | | | | | | | | | | | | | | Сброс: 0000000h | | | |
|----------------|---------|--|-------------|----|----|----|----|----|----|----|----|----|----|-----------------|----|------|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | |
| - | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | OVAL | - |
| - | | | | | | | | | | | | | | | | | |
| ч | | | | | | | | | | | | | | | | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | |
| OVAL | 1 | Индикатор значение выходного сигнала компаратора | | | | | | | | | | | | | | | |
| | | 0 | VIN- > VIN+ | | | | | | | | | | | | | | |
| | | 1 | VIN- < VIN+ | | | | | | | | | | | | | | |
| – | 31-2, 0 | Зарезервировано | | | | | | | | | | | | | | | |

Таблица А.16.6 – Регистр управления компаратора блока n (n – 0, 1, 2)

| ACCTLn | | | | | | | | | | | | | | Сброс: 0000000h | | | | | | | | | | |
|---------------|-----|--|---------------------------------------|-----|----|-----|----|-----|----|-----|----|-----|----|-----------------|----|-------|-------|--------|------|-------|------|------|-----|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | TO EN | ASRCP | TS VAL | TSEN | ISVAL | ISEN | CINV | CEN | |
| - | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | 3 ч | | |
| Поле | Бит | Описание | | | | | | | | | | | | | | | | | | | | | | |
| TOEN | 10 | Бит запрещения генерации события запуска АЦП | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Разрешено | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Запрещено | | | | | | | | | | | | | | | | | | | | | |
| ASRCP | 9-8 | Поле задания источника сигнала входа «+» компаратора | | | | | | | | | | | | | | | | | | | | | | |
| | | 00 | Вывод Сn+ | | | | | | | | | | | | | | | | | | | | | |
| | | 01 | Вывод С1+ | | | | | | | | | | | | | | | | | | | | | |
| | | 10 | Опорное напряжение с внутреннего ЦАПn | | | | | | | | | | | | | | | | | | | | | |
| | | 11 | Опорное напряжение с внутреннего ЦАП0 | | | | | | | | | | | | | | | | | | | | | |
| TSVAL | 7 | Бит выбора уровня для генерации события запуска АЦП | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | Низкий | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | Высокий | | | | | | | | | | | | | | | | | | | | | |

Окончание таблицы А.16.6

| Поле | Бит | Описание |
|-------|-------|---|
| TSEN | 6-5 | Поле выбора события выходного сигнала компаратора, по которому будет сформировано событие запуска АЦП |
| | | 00 По уровню, заданному битом TSVAl |
| | | 01 По переднему фронту |
| | | 10 По заднему фронту |
| | | 11 По обоим фронтам |
| ISVAL | 4 | Бит выбора уровня для генерации прерывания |
| | | 0 Низкий |
| | | 1 Высокий |
| ISEN | 3-2 | Поле выбора события выходного сигнала компаратора, по которому будет сформировано прерывание |
| | | 00 По уровню, заданному битом ISVAL |
| | | 01 По переднему фронту |
| | | 10 По заднему фронту |
| | | 11 По обоим фронтам |
| CINV | 1 | Бит изменения полярности выходного сигнала компаратора |
| | | 0 Прямой |
| | | 1 Инверсный |
| CEN | 0 | Бит разрешения работы компаратора |
| | | 0 Запрещено |
| | | 1 Разрешено |
| – | 31-11 | Зарезервировано |

Таблица А.16.7 – Регистр питания аналоговой части

| Поле | Бит | Описание |
|--|------|---|
| <p>POWER A001D07Ch Сброс: 0000h</p> <p>31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16</p> <p>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</p> <p style="text-align: right;">PWU</p> <p style="text-align: right;">3 ч</p> | | |
| PWU | 0 | Бит включения подачи аналогового питания на часть блока |
| – | 31-1 | Зарезервировано |

А.17 Регистры контроллера SPI

Таблица А.17.1 – Регистр управления 0

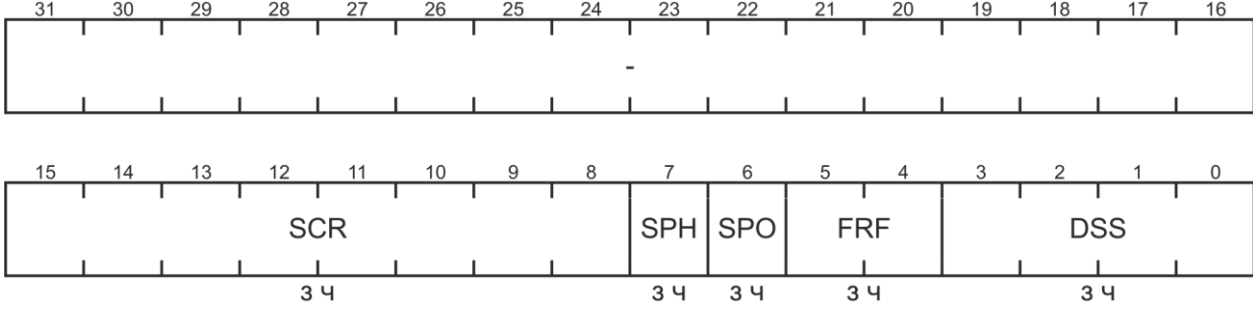
| SPI_CR0 | | Сброс: 00000000h |
|--|-------|---|
|  | | |
| Поле | Биты | Описание |
| SCR | 15-8 | Коэффициент деления второго делителя. Может принимать значения 00h до FFh |
| SPH | 7 | Фаза сигнала SSPCLKOUT (только для протокола обмена SPI) |
| | | 0 Выборка данных по переднему фронту синхросигнала, а установка по заднему |
| SPO | 6 | 1 Выборка данных по заднему фронту синхросигнала, а установка по переднему |
| | | Полярность сигнала SSPCLKOUT (только для протокола обмена SPI) |
| FRF | 5-4 | 0 В режиме ожидания линия SPI_CLK удерживается в состоянии логического нуля |
| | | 1 В режиме ожидания линия SPI_CLK удерживается в состоянии логической единицы |
| FRF | 5-4 | Поле выбора протокола обмена информацией |
| | | 00 SPI |
| | | 01 SSI |
| | | 10 Microwire |
| | | 11 Зарезервировано |
| DSS | 3-0 | Размер слова данных: |
| | | 0h-2h Зарезервировано |
| | | 3h 4 бита |
| | | 4h 5 бит |
| | | 5h 6 бит |
| | | 6h 7 бит |
| | | 7h 8 Бит |
| | | 8h 9 бит |
| | | 9h 10 бит |
| | | Ah 11 бит |
| | | Bh 12 бит |
| | | Ch 13 бит |
| | | Dh 14 бит |
| Eh 15 бит | | |
| Fh 16 бит | | |
| – | 31-16 | Зарезервировано |

Таблица А.17.2 – Регистр управления 1

| SPI_CR1 | | Сброс: 00000000h |
|--|---------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: space-between; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 100%; height: 20px;"></div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">SOD</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">MS</div> <div style="border-right: 1px solid black; width: 30px; text-align: center;">SSE</div> <div style="width: 30px; text-align: center;">-</div> </div> | | |
| <div style="display: flex; justify-content: flex-end; gap: 20px;"> 3 ч3 ч3 ч </div> | | |
| Поле | Бит | Описание |
| SOD | 3 | Бит запрета передачи данных. В режиме мастера значение бита игнорируется. В режиме ведомого бит контролирует выход данных. Пока бит сброшен передача и прием данных разрешены. Установка бита блокирует передачу данных и переводит вывод SPI_TXD в состояние слабой логической единицы, при этом прием тактового сигнала и прием данных не блокируются |
| MS | 2 | Бит выбора режима работы |
| | | 0 Мастер 1 Ведомый |
| SSE | 1 | Бит разрешения работы приемопередатчика |
| | | 0 Запрещено 1 Разрешено |
| – | 31-4, 0 | Зарезервировано |

Таблица А.17.3 – Регистр данных

| SPI_DR | | Сброс: 00000000h |
|--|-------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 40px; margin: 5px 0;"></div> | | |
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="display: flex; justify-content: center; align-items: center; border: 1px solid black; padding: 5px;"> <div style="border-right: 1px solid black; width: 100%; height: 20px;"></div> <div style="margin: 0 10px;">DATA</div> </div> | | |
| 3 ч | | |
| Поле | Бит | Описание |
| DATA | 15-0 | 16-разрядный буфер FIFO приемника и передатчика. Данные для передачи записываются в регистр. В случае если размер передаваемых данных менее 16 бит, перед записью в регистр они должны быть выравнены по правой границе. Неиспользуемые биты игнорируются. Принятые данные автоматически выравниваются по правой границе в блоке приемника. При чтении регистр возвращает принятые данные |
| – | 31-16 | Зарезервировано |

Таблица А.17.4 – Регистр состояния

| SPI_SR | | Сброс: 00000003h |
|--|------|---|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 20px; margin: 5px 0;"></div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> BSYRFFRNETNFTFE </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 44444 </div> </div> | | |
| Поле | Бит | Описание |
| BSY | 4 | Флаг активности |
| | | 0 Приемопередатчик не активен |
| | | 1 Приемопередатчик передает/принимает данные, либо буфер FIFO передатчика не пуст |
| RFF | 3 | Флаг заполнения буфера FIFO приемника |
| | | 0 Не заполнен |
| | | 1 Заполнен |
| RNE | 2 | Индикатор того, что буфер FIFO приемника не пуст |
| | | 0 Пуст |
| | | 1 Не пуст |
| TNF | 1 | Индикатор того, что буфера FIFO передатчика не заполнен |
| | | 0 Заполнен |
| | | 1 Не заполнен |
| TFE | 0 | Флаг пустоты буфера FIFO передатчика |
| | | 0 Не пуст |
| | | 1 Пуст |
| – | 31-5 | Зарезервировано |

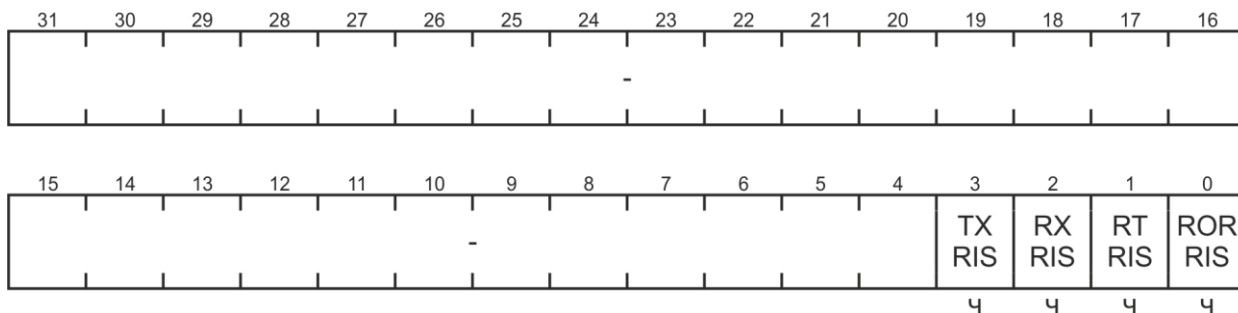
Таблица А.17.5 – Регистр делителя тактовой частоты

| SPI_CPSR | | Сброс: 00000000h |
|---|------|--|
| <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 31302928272625242322212019181716 </div> <div style="border: 1px solid black; height: 20px; margin: 5px 0;"></div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; padding-bottom: 5px;"> 1514131211109876543210 </div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> CPSDVSR 0 </div> <div style="display: flex; justify-content: flex-end; margin-top: 5px;"> 344 </div> </div> | | |
| Поле | Биты | Описание |
| CPSDVSR | 7-0 | Коэффициент деления первого делителя. Может принимать четные значения от 02h до FEh |
| – | 31-8 | Зарезервировано |

Таблица А.17.6 – Регистр прерываний

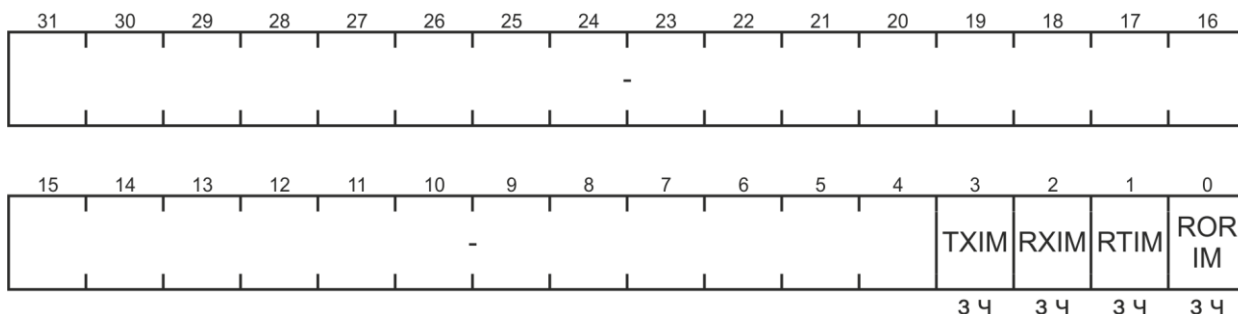
Регистр состояния прерываний
SPI_RIS

Сброс: 00000008h



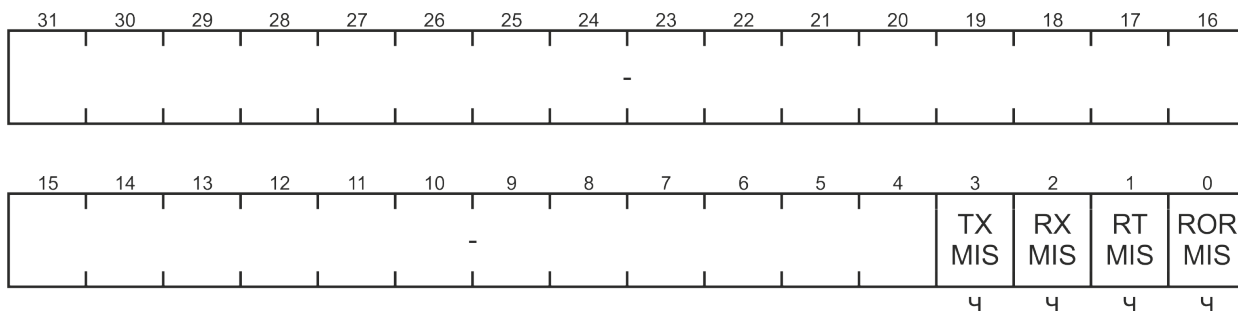
Регистр маски прерываний
SPI_IMSC

Сброс: 00000003h



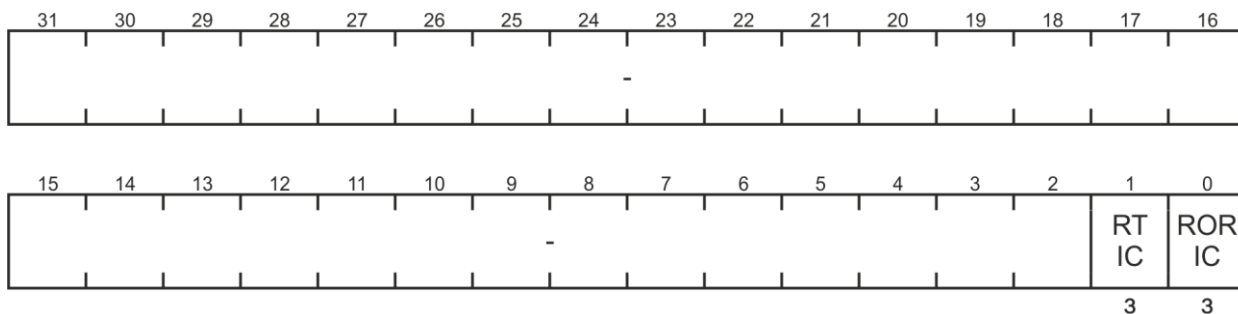
Регистр состояния прерываний с маскированием
SPI_MIS

Сброс: 00000000h



Регистр сброса прерываний
SPI_ICR

Сброс: 00000000h



Окончание таблицы А.17.6

| Поле | | Бит | Описание |
|------|---------------------------|------|--|
| TX | RIS/ IM/ MIS/ IC | 3 | Буфер передатчика опустошен наполовину |
| RX | | 2 | Буфер приемника заполнен наполовину |
| RT | | 1 | Таймаут приема данных |
| ROR | | 0 | Переполнению буфера приемника |
| – | | 31-4 | Зарезервировано |

Функционирование регистров.
 При возникновении прерываний устанавливаются соответствующие им немаскируемые флаги в регистре RIS.
 Установка/сброс бит в регистре IMSC формирует маску. По умолчанию, все биты сброшены, и установка флагов запрещена. Для того чтобы убрать маску следует установить соответствующий бит.
 В регистре MIS устанавливаются только те флаги, которые не закрыты маской регистра IMSC.
 Запись единиц в биты регистра ICR сбрасывает соответствующие им флаги в регистрах RIS и MIS, а также прерывания, вызвавшие установку этих флагов. Биты RTRIS и RTMIS также сбрасываются после чтения буфера приемника

Таблица А.17.7 – Регистр управления прямым доступом к памяти

| Поле | | Бит | Описание |
|------------------|--|------|--|
| SPI_DMACR | | | Сброс: 00000000h |
| | | | |
| TXDMAE | | 1 | Бит разрешения использования DMA при передаче |
| | | 0 | Не используется |
| | | 1 | Разрешено формирование запросов DMA для обслуживания буфера FIFO передатчика |
| RXDMAE | | 0 | Бит разрешения использования DMA при приеме |
| | | 0 | Не используется |
| | | 1 | Разрешено формирование запросов DMA для обслуживания буфера FIFO приемника |
| – | | 31-2 | Зарезервировано |

A.18 Регистры контроллера пользовательской флеш-памяти

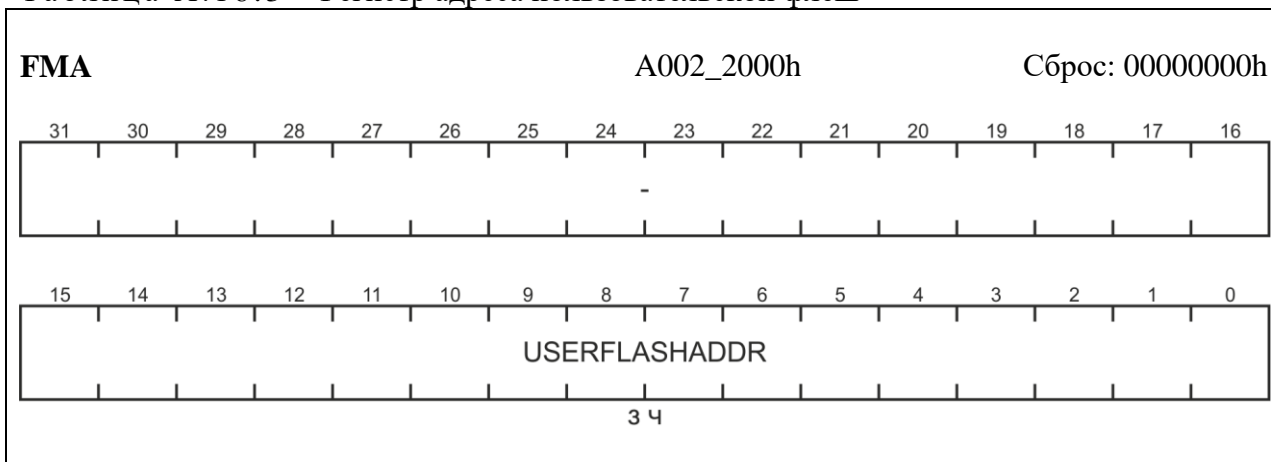
Таблица А.18.1 – Информационное слово пользовательской флеш

| Поле | Биты | Описание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|----|--|----|---|----|--------------------------------|----|-------------|----------------------|---|--|--|--|--|--|--|--|----|----|----|----|----|----|----|----|---------------------|---|--|--|--|--|--|--|--|----|----|----|----|----|----|---|---|---------|---|--|--|--|--|--|--|--|---|---|---|---|---|---|---|---|---------------|---|-------------|----------------------|---|--|--|--|--|--|--|--|
| <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>INFO_WORD + 0000h (смещение относительно начального адреса флеш-памяти)</p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td style="width: 4%;">31</td><td style="width: 4%;">30</td><td style="width: 4%;">29</td><td style="width: 4%;">28</td><td style="width: 4%;">27</td><td style="width: 4%;">26</td><td style="width: 4%;">25</td><td style="width: 4%;">24</td> <td style="width: 4%;">LOCK _IFB_ UF</td> </tr> <tr> <td colspan="8" style="text-align: center;">-</td> </tr> </table> </div> <div style="width: 48%;"> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td style="width: 4%;">23</td><td style="width: 4%;">22</td><td style="width: 4%;">21</td><td style="width: 4%;">20</td><td style="width: 4%;">19</td><td style="width: 4%;">18</td><td style="width: 4%;">17</td><td style="width: 4%;">16</td> <td style="width: 4%;">LOCK _IFB_ LF</td> </tr> <tr> <td colspan="8" style="text-align: center;">-</td> </tr> </table> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td style="width: 4%;">15</td><td style="width: 4%;">14</td><td style="width: 4%;">13</td><td style="width: 4%;">12</td><td style="width: 4%;">11</td><td style="width: 4%;">10</td><td style="width: 4%;">9</td><td style="width: 4%;">8</td> <td style="width: 4%;">PORTNUM</td> </tr> <tr> <td colspan="8" style="text-align: center;">-</td> </tr> </table> </div> <div style="width: 48%;"> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td style="width: 4%;">7</td><td style="width: 4%;">6</td><td style="width: 4%;">5</td><td style="width: 4%;">4</td><td style="width: 4%;">3</td><td style="width: 4%;">2</td><td style="width: 4%;">1</td><td style="width: 4%;">0</td> <td style="width: 4%;">EXTMEM SEL</td><td style="width: 4%;">-</td><td style="width: 4%;">EN_ GPIO</td><td style="width: 4%;">BOOT FROM _IFB</td> </tr> <tr> <td colspan="8" style="text-align: center;">-</td> </tr> </table> </div> </div> | | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | LOCK _IFB_ UF | - | | | | | | | | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | LOCK _IFB_ LF | - | | | | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | PORTNUM | - | | | | | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | EXTMEM SEL | - | EN_ GPIO | BOOT FROM _IFB | - | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | LOCK _IFB_ UF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | LOCK _IFB_ LF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | PORTNUM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | EXTMEM SEL | - | EN_ GPIO | BOOT FROM _IFB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LOCK_ IFB_UF | 24 | <p>Бит включения защиты пользовательской флеш</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>Запрет записи и стирания информационного блока пользовательской флеш</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Защита выключена (по умолчанию)</td> </tr> </table> | 0 | Запрет записи и стирания информационного блока пользовательской флеш | 1 | Защита выключена (по умолчанию) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Запрет записи и стирания информационного блока пользовательской флеш | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Защита выключена (по умолчанию) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LOCK_ IFB_LF | 16 | <p>Бит включения защиты загрузочной флеш</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>Запрет записи и стирания информационного блока загрузочной флеш</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Защита выключена (по умолчанию)</td> </tr> </table> | 0 | Запрет записи и стирания информационного блока загрузочной флеш | 1 | Защита выключена (по умолчанию) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Запрет записи и стирания информационного блока загрузочной флеш | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Защита выключена (по умолчанию) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PORTNUM, PINNUM | 15-12, 11-8 | <p>Поле указания вывода микроконтроллера, выбирающего источник загрузки. Поле PORTNUM задает номер порта (от 000b до 111b), поле PINNUM – номер вывода</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EXTMEM SEL | 4-3 | <p>Выбор номера функции выводов GPIO, на которые подключен интерфейс внешней памяти</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">00</td> <td>Активна первая функция выводов</td> </tr> <tr> <td style="text-align: center;">01</td> <td>Активна вторая функция выводов</td> </tr> <tr> <td style="text-align: center;">10</td> <td>Активна третья функция выводов</td> </tr> </table> | 00 | Активна первая функция выводов | 01 | Активна вторая функция выводов | 10 | Активна третья функция выводов | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | Активна первая функция выводов | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | Активна вторая функция выводов | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | Активна третья функция выводов | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EN_GPIO | 1 | <p>Бит включения функции выбора мапируемой памяти</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>Включена функция выбора вывода микроконтроллера, которая определяет, что мапируется в глобальный адрес памяти 0000h – флеш-память или внешняя память. Для указания вывода микроконтроллера используются поля PORTNUM и PINNUM</td> </tr> <tr> <td style="text-align: center;">1</td> <td>В адрес 0000h мапируется флеш-память (по умолчанию)</td> </tr> </table> | 0 | Включена функция выбора вывода микроконтроллера, которая определяет, что мапируется в глобальный адрес памяти 0000h – флеш-память или внешняя память. Для указания вывода микроконтроллера используются поля PORTNUM и PINNUM | 1 | В адрес 0000h мапируется флеш-память (по умолчанию) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Включена функция выбора вывода микроконтроллера, которая определяет, что мапируется в глобальный адрес памяти 0000h – флеш-память или внешняя память. Для указания вывода микроконтроллера используются поля PORTNUM и PINNUM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | В адрес 0000h мапируется флеш-память (по умолчанию) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BOOT FROM_IFB | 0 | <p>Бит выбора блока памяти для мапирования в адрес 0000h основного блока загрузочной флеш-памяти</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>Информационный блок загрузочной флеш-памяти мапируется в основной блок в область 0000h – 1FFFh. И далее с адреса 0000h стартует программа пользователя. Основной блок в этом случае начинается с 2000h</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Старт программы с адреса 0000h основного блока загрузочной флеш. По умолчанию, бит установлен</td> </tr> </table> | 0 | Информационный блок загрузочной флеш-памяти мапируется в основной блок в область 0000h – 1FFFh. И далее с адреса 0000h стартует программа пользователя. Основной блок в этом случае начинается с 2000h | 1 | Старт программы с адреса 0000h основного блока загрузочной флеш. По умолчанию, бит установлен | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Информационный блок загрузочной флеш-памяти мапируется в основной блок в область 0000h – 1FFFh. И далее с адреса 0000h стартует программа пользователя. Основной блок в этом случае начинается с 2000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Старт программы с адреса 0000h основного блока загрузочной флеш. По умолчанию, бит установлен | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| – | 31-25, 23-17, 7-5, 2 | Зарезервировано | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Таблица А.18.2 – Регистры информационного блока пользовательской флеш

| | |
|--|---|
| 32-разрядные доступные побайтно пользовательские регистры. | |
| USER_REG | + 004h – + 03Ch + 050h – + 07Ch + 0A0h – + FFCh |
| 32-разрядные доступные побайтно регистры защиты основного блока загрузочной флеш | |
| BF_LOCK_PAGES31-0 | + 040h |
| BF_LOCK_PAGES63-32 | + 044h |
| BF_LOCK_PAGES95-64 | + 048h |
| BF_LOCK_PAGES127-96 | + 04Ch |
| 32-разрядные доступные побайтно регистры защиты основного блока пользовательской флеш. | |
| UF_LOCK_PAGES31-0 | + 080h |
| UF_LOCK_PAGES63-32 | + 084h |
| UF_LOCK_PAGES95-64 | + 088h |
| UF_LOCK_PAGES127-96 | + 08Ch |
| UF_LOCK_PAGES159-128 | + 090h |
| UF_LOCK_PAGES191-160 | + 094h |
| UF_LOCK_PAGES223-192 | + 098h |
| UF_LOCK_PAGES255-224 | + 09Ch |
| <p>Каждый регистр защиты взаимодействует с 32 страницами (каждой странице соответствует один бит). Цифры в названии регистров указывают на страницы, которые могут быть защищены. Запись нуля в бит регистра запрещает запись и стирание соответствующей страницы. По умолчанию, состояние всех регистров FFFFFFFFh. Вместо адреса регистра указано смещение относительно начального адреса (000h) информационного блока</p> | |

Таблица А.18.3 – Регистр адреса пользовательской флеш



Окончание таблицы А.18.3

| Поле | Биты | Описание |
|---------------|-------|---|
| USERFLASHADDR | 15-0 | Адрес, используемый при командах записи, чтения и постраничного стирания. Должен быть записан до установки бита выполнения команды записи/чтения/стирания Примечание – При постраничном стирании младшие 8 бит не имеют значения. |
| – | 31-16 | Зарезервировано |

Таблица А.18.4 – Регистр данных пользовательской флеш

| Поле | Биты | Описание |
|---------------|-------------------|--|
| – | 31-16 | Зарезервировано |
| FMD | A002_2004h | Сброс: 00000000h |
| | | |
| USERFLASHDATA | 7-0 | Байт данных. Используется при командах записи и чтения через регистр FMC. Байт данных должен быть загружены в регистр до установки бита команды записи. При чтении данные загружаются автоматически вместе с очисткой бита команды чтения |
| – | 31-8 | Зарезервировано |

Таблица А.18.5 – Регистр команд пользовательской флеш

| Поле | Биты | Описание |
|-----------------|-------------------|-------------------------|
| – | 31-16 | Зарезервировано |
| FMC | A002_2008h | Сброс: 00000000h |
| | | |
| RE AD_IFB | 6-4 | 3 |
| PAGE ERAS E_IFB | 5-3 | 3 |
| WRI TE_IFB | 4-2 | 3 |
| READ | 3-1 | 3 |
| FULL _ERA SE | 2-0 | 3 |
| PAGE _ERA SE | 1-0 | 3 |
| WRI TE | 0 | 3 |

Продолжение таблицы А.18.5

| Поле | Бит | Описание |
|--|-------|--|
| MAGIC_KEY | 31-16 | Код запуска команды. Все команды для вступления в силу должны сопровождаться записью в поле MAGIC_KEY значения А442h. Команды должны выполняться по одной, т. е. запись следующей команды разрешена, только после завершения предыдущей. Одновременная запись нескольких команд приведет к невыполнению ни одной из них и выдаст флаг ошибки. Чтение поля MAGIC_KEY всегда возвращает 0000 |
| READ_IFB | 6 | Бит чтения информационного блока. Адрес в регистр FMA (разряды с 8 по 0) должен быть записан до установки бита READ_IFB. Установка бита READ_IFB активирует команду чтения из информационного блока пользовательской флеш. Прочитанные данные появляются в регистре FMD |
| PAGEERASE_IFB | 5 | Бит постраничного стирания информационного блока. Данные в регистр FMD и адрес в регистр FMA (разряд 8) должны быть записаны до установки бита PAGEERASE_IFB. Адрес нужно указывать поскольку в информационном блоке две страницы. Установка бита PAGEERASE_IFB активирует команду постраничного стирания в информационном блоке загрузочной флеш. |
| WRITE_IFB | 4 | Бит записи в информационный блок. Данные в регистр FMD и адрес в регистр FMA (разряды с 8 по 0) должны быть записаны до установки бита WRITE_IFB. Установка бита WRITE_IFB активирует команду записи в информационный блок загрузочной флеш |
| READ | 3 | Бит чтения основного блока. Адрес в регистр FMA должен быть записан до установки бита READ. Установка бита READ активирует команду чтения из основного блока пользовательской флеш. Прочитанные данные появляются в регистре FMD |
| FULL_ERASE | 2 | Бит стирания основного блока. Установка бита FULL_ERASE активирует команду полного стирания основного блока пользовательской флеш |
| PAGE_ERASE | 1 | Бит постраничного стирания основного блока. Данные в регистр FMD и адрес в регистр FMA (разряды с 15 по 8) должны быть записаны до установки бита PAGE_ERASE. Установка бита PAGE_ERASE активирует команду постраничного стирания в основном блоке пользовательской флеш |
| WRITE | 0 | Бит записи в основной блок. Данные в регистр FMD и адрес в регистр FMA должны быть записаны до установки бита WRITE. Установка бита WRITE активирует команду записи в основной блок пользовательской флеш |
| – | 15-7 | Зарезервировано |
| Каждый бит 6 по 0 по окончании выполнения команды сбрасывается автоматически | | |

Таблица А.18.6 – Регистр статуса пользовательской флеш

| FCIS | | A002_200Ch | | Сброс: 00000000h | |
|---|------|--|--|------------------|--|
| | | | | | |
| Поле | Бит | Описание | | | |
| OP_ERROR | 1 | Флаг ошибки записи. Устанавливается в случае попытки стирания блок защищенного от записи, попытки записи в не существующий блок либо чтения из несуществующего блока флеш | | | |
| OP_CMLT | 0 | Флаг завершения операции. Устанавливается по завершении операции чтения/записи/стирания | | | |
| – | 31-2 | Зарезервировано | | | |
| Каждый флаг сбрасывается записью единицы в соответствующий бит регистра FCI | | | | | |

Таблица А.18.7 – Регистр маски прерываний пользовательской флеш

| FCIM | | A002_2010h | | Сброс: 00000000h | |
|--------------|------|---|-----------|------------------|--|
| | | | | | |
| Поле | Бит | Описание | | | |
| MASK_OP_CMLT | 0 | Бит разрешения генерирования прерывания по завершении операции чтения/записи/стирания | | | |
| | | 0 | Запрещено | | |
| | | 1 | Разрешено | | |
| – | 31-1 | Зарезервировано | | | |

Таблица А.18.8 – Регистр сброса флагов статуса пользовательской флеш

| FCIS | | A002_2014h | Сброс: 00000000h |
|-------------|------|---|------------------|
| | | | |
| Поле | Бит | Описание | |
| CLR_OP_ERR | 1 | Бит сброса флага ошибки записи. Запись единицы сбрасывает флаг OP_ERROR в регистре FCIS | |
| CLR_OP_CMLT | 0 | Бит сброса флага завершения операции. Запись единицы сбрасывает флаг OP_CMLT в регистре FCIS | |
| – | 31-2 | Зарезервировано | |

Таблица А.18.9 – Регистры временных параметров пользовательской флеш-памяти

| Мнемоника | Назначение параметра, задаваемого регистром | Значение по сбросу | Диапазон значений |
|---------------|---|--------------------|------------------------|
| T_ACC | Используется в транзакциях чтения из флеш-памяти, задает задержку от установки адреса, до считывания данных из флеш-памяти | 4 такта | 3 такта и более |
| T_NVS | Используется в транзакциях записи и стирания, задает задержку поднятия сигнала NVSTR после поднятия сигнала PROG | 500 тактов | 500 тактов и более |
| T_NVH | Используется в транзакциях записи и стирания, задает задержку отпускания сигнала NVSTR после отпускания сигнала PROG | | |
| T_RCV | Используется в транзакциях записи и стирания, задает задержку от отпускания сигнала NVSTR до поднятия его на следующей транзакции | 100 тактов | 100 тактов и более |
| T_PGS | Используется в транзакциях записи, задает задержку поднятия сигнала YE после поднятия сигнала NVSTR | 1000 тактов | 1000 тактов и более |
| T_PROG | Используется в транзакциях записи, задает длительность сигнала PROG в транзакциях записи | 2000 тактов | От 2000 до 4000 тактов |
| T_PGH | Используется в транзакциях записи, задает задержку отпускания сигнала NVSTR после отпускания сигнала YE | 2 такта | 2 такта и более |

Окончание таблицы А.18.9

| Мнемоника | Назначение параметра, задаваемого регистром | Значение по сбросу | Диапазон значений |
|--|--|--------------------|------------------------------|
| T_ERASE | Используется в транзакциях стирания, задает длительность сигнала ERASE в транзакциях стирания | 2000000 тактов | От 2000000 до 4000000 тактов |
| T_ME | Используется в транзакциях массового стирания, задает длительность сигнала ERASE в транзакциях массового стирания | | |
| T_NVH1 | Используется в транзакциях массового стирания, задает задержку отпускания сигнала NVSTR после отпускания сигнала ERASE | 10000 тактов | 10000 тактов и более |
| <p>Примечания</p> <p>1 Все параметры задаются в тактах системного сигнала.</p> <p>2 Мнемоническое название регистра параметра совпадает с названием параметра.</p> | | | |

Приложение Б (обязательное) Карта памяти регистров микроконтроллера

Области памяти регистров, мнемоника и названия регистров приведены в таблицах Б.1 – Б.35.

Таблица Б.1 – Регистры контроллера АЦП

| Адрес | Мнемоника | Сброс | Название |
|-----------|-----------|-----------|---|
| 80000000h | ACTSS | 00000000h | Регистр включения секвенсоров |
| 80000004h | CRIS | 00000000h | Регистр флагов немаскированных прерываний |
| 80000008h | CIM | 00000000h | Регистр маски прерываний |
| 8000000Ch | ISC | 00000000h | Регистр флагов маскированных прерываний |
| 80000010h | OSTAT | 00000000h | Регистр флагов буферов результатов и блока DMA |
| 80000014h | EMUX | 00000000h | Регистр выбора событий запуска секвенсоров |
| 80000018h | USTAT | 00000000h | Регистр пустоты буферов результатов |
| 8000001Ch | – | – | Зарезервировано |
| 80000020h | SPC0 | 00000000h | Регистр фаз запуска модулей АЦП0 и АЦП1 |
| 80000024h | SPC1 | 00000000h | Регистр фаз запуска модулей АЦП2 и АЦП3 |
| 80000028h | SPC2 | 00000000h | Регистр фаз запуска модулей АЦП4 и АЦП5 |
| 8000002Ch | SPC3 | 00000000h | Регистр фаз запуска модулей АЦП6 и АЦП7 |
| 80000030h | SPC4 | 00000000h | Регистр фаз запуска модулей АЦП8 и АЦП9 |
| 80000034h | SPC5 | 00000000h | Регистр фаз запуска модулей АЦП10 и АЦП11 |
| 80000038h | SAC | 00000000h | Регистр параметров усреднителей |
| 8000003Ch | RICNT | 00000000h | Регистр настройки режима сброса счетчика прерываний |

Таблица Б.2 – Адреса регистров секвенсоров контроллера АЦП

| Мнемоника | Секвенсоры | | | | | | | |
|-----------------|-----------------|-------|-------|-------|-------|-------|-------|-------|
| | Адрес 8000_xxxx | | | | | | | |
| | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 |
| MUX | 0040h | 0060h | 0080h | 00A0h | 00C0h | 00E0h | 0100h | 0120h |
| CTL | 0044h | 0064h | 0084h | 00A4h | 00C4h | 00E4h | 0104h | 0124h |
| Зарезервировано | 0048h | 0068h | 0088h | 00A8h | 00C8h | 00E8h | 0108h | 0128h |
| FSTAT | 004Ch | 006Ch | 008Ch | 00ACh | 00CCh | 00ECh | 010Ch | 012Ch |
| OP | 0050h | 0070h | 0090h | 00B0h | 00D0h | 00F0h | 0110h | 0130h |
| DCP | 0054h | 0074h | 0094h | 00B4h | 00D4h | 00F4h | 0114h | 0134h |
| TMR | 0058h | 0078h | 0098h | 00B8h | 00D8h | 00F8h | 0118h | 0138h |
| FIFO | 005Ch | 007Ch | 009Ch | 00BCh | 00DCh | 00FCh | 011Ch | 013Ch |

Таблица Б.3 – Мнемоника и соответствующие названия регистров секвенсоров

| Мнемоника | Сброс | Название |
|-----------|-----------|--|
| MUX | 00000000h | Регистр выбора каналов для измерений |
| CTL | 00000000h | Регистр управления |
| FIFO | 00000000h | Регистр измерения |
| FSTAT | 00000000h | Регистр флагов |
| OP | 00000000h | Регистр флагов измерений |
| DCP | 00000000h | Регистр выбора компаратора |
| TMR | 00000000h | Регистр таймера перезапусков модулей АЦП |

Таблица Б.4 – Регистры контроллера АЦП

| Адрес | Мнемоника | Сброс | Название |
|-----------|-----------|-----------|------------------------------------|
| 80000140h | DCCTL0 | 00000000h | Регистр управления компаратором 0 |
| 80000144h | DCCTL1 | 00000000h | Регистр управления компаратором 1 |
| 80000148h | DCCTL2 | 00000000h | Регистр управления компаратором 2 |
| 8000014Ch | DCCTL3 | 00000000h | Регистр управления компаратором 3 |
| 80000150h | DCCTL4 | 00000000h | Регистр управления компаратором 4 |
| 80000154h | DCCTL5 | 00000000h | Регистр управления компаратором 5 |
| 80000158h | DCCTL6 | 00000000h | Регистр управления компаратором 6 |
| 8000015Ch | DCCTL7 | 00000000h | Регистр управления компаратором 7 |
| 80000160h | DCCTL8 | 00000000h | Регистр управления компаратором 8 |
| 80000164h | DCCTL9 | 00000000h | Регистр управления компаратором 9 |
| 80000168h | DCCTL10 | 00000000h | Регистр управления компаратором 10 |
| 8000016Ch | DCCTL11 | 00000000h | Регистр управления компаратором 11 |
| 80000170h | DCCTL12 | 00000000h | Регистр управления компаратором 12 |
| 80000174h | DCCTL13 | 00000000h | Регистр управления компаратором 13 |
| 80000178h | DCCTL14 | 00000000h | Регистр управления компаратором 14 |
| 8000017Ch | DCCTL15 | 00000000h | Регистр управления компаратором 15 |
| 80000180h | DCCTL16 | 00000000h | Регистр управления компаратором 16 |
| 80000184h | DCCTL17 | 00000000h | Регистр управления компаратором 17 |
| 80000188h | DCCTL18 | 00000000h | Регистр управления компаратором 18 |
| 8000018Ch | DCCTL19 | 00000000h | Регистр управления компаратором 19 |
| 80000190h | DCCTL20 | 00000000h | Регистр управления компаратором 20 |
| 80000194h | DCCTL21 | 00000000h | Регистр управления компаратором 21 |
| 80000198h | DCCTL22 | 00000000h | Регистр управления компаратором 22 |
| 8000019Ch | DCCTL23 | 00000000h | Регистр управления компаратором 23 |
| 800001A0h | DCCMP0 | 00000000h | Регистр диапазона компаратора 0 |
| 800001A4h | DCCMP1 | 00000000h | Регистр диапазона компаратора 1 |
| 800001A8h | DCCMP2 | 00000000h | Регистр диапазона компаратора 2 |
| 800001ACh | DCCMP3 | 00000000h | Регистр диапазона компаратора 3 |
| 800001B0h | DCCMP4 | 00000000h | Регистр диапазона компаратора 4 |
| 800001B4h | DCCMP5 | 00000000h | Регистр диапазона компаратора 5 |
| 800001B8h | DCCMP6 | 00000000h | Регистр диапазона компаратора 6 |
| 800001BCh | DCCMP7 | 00000000h | Регистр диапазона компаратора 7 |
| 800001C0h | DCCMP8 | 00000000h | Регистр диапазона компаратора 8 |
| 800001C4h | DCCMP9 | 00000000h | Регистр диапазона компаратора 9 |
| 800001C8h | DCCMP10 | 00000000h | Регистр диапазона компаратора 10 |
| 800001CCh | DCCMP11 | 00000000h | Регистр диапазона компаратора 11 |
| 800001D0h | DCCMP12 | 00000000h | Регистр диапазона компаратора 12 |
| 800001D4h | DCCMP13 | 00000000h | Регистр диапазона компаратора 13 |
| 800001D8h | DCCMP14 | 00000000h | Регистр диапазона компаратора 14 |
| 800001DCh | DCCMP15 | 00000000h | Регистр диапазона компаратора 15 |
| 800001E0h | DCCMP16 | 00000000h | Регистр диапазона компаратора 16 |
| 800001E4h | DCCMP17 | 00000000h | Регистр диапазона компаратора 17 |
| 800001E8h | DCCMP18 | 00000000h | Регистр диапазона компаратора 18 |
| 800001ECh | DCCMP19 | 00000000h | Регистр диапазона компаратора 19 |
| 800001F0h | DCCMP20 | 00000000h | Регистр диапазона компаратора 20 |
| 800001F4h | DCCMP21 | 00000000h | Регистр диапазона компаратора 21 |
| 800001F8h | DCCMP22 | 00000000h | Регистр диапазона компаратора 22 |
| 800001FCh | DCCMP23 | 00000000h | Регистр диапазона компаратора 23 |

Продолжение таблицы Б.4

| Адрес | Мнемоника | Сброс | Название |
|-----------|-----------|-----------|---|
| 80000200h | DCVAL0 | 00000000h | Регистр результата измерения компаратора 0 |
| 80000204h | DCVAL1 | 00000000h | Регистр результата измерения компаратора 1 |
| 80000208h | DCVAL2 | 00000000h | Регистр результата измерения компаратора 2 |
| 8000020Ch | DCVAL3 | 00000000h | Регистр результата измерения компаратора 3 |
| 80000210h | DCVAL4 | 00000000h | Регистр результата измерения компаратора 4 |
| 80000214h | DCVAL5 | 00000000h | Регистр результата измерения компаратора 5 |
| 80000218h | DCVAL6 | 00000000h | Регистр результата измерения компаратора 6 |
| 8000021Ch | DCVAL7 | 00000000h | Регистр результата измерения компаратора 7 |
| 80000220h | DCVAL8 | 00000000h | Регистр результата измерения компаратора 8 |
| 80000224h | DCVAL9 | 00000000h | Регистр результата измерения компаратора 9 |
| 80000228h | DCVAL10 | 00000000h | Регистр результата измерения компаратора 10 |
| 8000022Ch | DCVAL11 | 00000000h | Регистр результата измерения компаратора 11 |
| 80000230h | DCVAL12 | 00000000h | Регистр результата измерения компаратора 12 |
| 80000234h | DCVAL13 | 00000000h | Регистр результата измерения компаратора 13 |
| 80000238h | DCVAL14 | 00000000h | Регистр результата измерения компаратора 14 |
| 8000023Ch | DCVAL15 | 00000000h | Регистр результата измерения компаратора 15 |
| 80000240h | DCVAL16 | 00000000h | Регистр результата измерения компаратора 16 |
| 80000244h | DCVAL17 | 00000000h | Регистр результата измерения компаратора 17 |
| 80000248h | DCVAL18 | 00000000h | Регистр результата измерения компаратора 18 |
| 8000024Ch | DCVAL19 | 00000000h | Регистр результата измерения компаратора 19 |
| 80000250h | DCVAL20 | 00000000h | Регистр результата измерения компаратора 20 |
| 80000254h | DCVAL21 | 00000000h | Регистр результата измерения компаратора 21 |
| 80000258h | DCVAL22 | 00000000h | Регистр результата измерения компаратора 22 |
| 8000025Ch | DCVAL23 | 00000000h | Регистр результата измерения компаратора 23 |

Окончание таблицы Б.4

| Адрес | Мнемоника | Сброс | Название |
|---------------------|-----------|-----------|-----------------------------------|
| 80000260h-800002E8h | – | – | Зарезервировано |
| 800002ECh | DCRTC | 00000000h | Регистр сброса компараторов |
| 800002F0h-800003FCh | – | – | Зарезервировано |
| 80000300h | PP0 | 00000000h | Регистр настроек модуля АЦП0 |
| 80000304h | PP1 | 00000000h | Регистр настроек модуля АЦП1 |
| 80000308h | PP2 | 00000000h | Регистр настроек модуля АЦП2 |
| 8000030Ch | PP3 | 00000000h | Регистр настроек модуля АЦП3 |
| 80000310h | PP4 | 00000000h | Регистр настроек модуля АЦП4 |
| 80000314h | PP5 | 00000000h | Регистр настроек модуля АЦП5 |
| 80000318h | PP6 | 00000000h | Регистр настроек модуля АЦП6 |
| 8000031Ch | PP7 | 00000000h | Регистр настроек модуля АЦП7 |
| 80000320h | PP8 | 00000000h | Регистр настроек модуля АЦП8 |
| 80000324h | PP9 | 00000000h | Регистр настроек модуля АЦП9 |
| 80000328h | PP10 | 00000000h | Регистр настроек модуля АЦП10 |
| 8000032Ch | PP11 | 00000000h | Регистр настроек модуля АЦП11 |
| 80000330h-800003F4h | – | – | Зарезервировано |
| 800003F8h | PSSI | 00000000h | Регистр синхронизации секвенсоров |
| 800003FCh-8000FFCh | – | – | Зарезервировано |

Таблица Б.5 – Адреса регистров контроллера GPIO портов микроконтроллера

| Мнемоника | Порты | | | | | | | |
|------------------------|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | Адрес 8001_xxxx | | | | | | | |
| | A | B | C | D | E | F | G | H |
| DATA | 0000h | 1000h | 2000h | 3000h | 4000h | 5000h | 6000h | 7000h |
| DATAOUT | 0004h | 1004h | 2004h | 3004h | 4004h | 5004h | 6004h | 7004h |
| Зарезервировано | 0008h-000Ch | 1008h-100Ch | 2008h-200Ch | 3008h-300Ch | 4008h-400Ch | 5008h-500Ch | 6008h-600Ch | 7008h-700Ch |
| OUTENSET | 0010h | 1010h | 2010h | 3010h | 4010h | 5010h | 6010h | 7010h |
| OUTENCLR | 0014h | 1014h | 2014h | 3014h | 4014h | 5014h | 6014h | 7014h |
| ALTFUNCSET | 0018h | 1018h | 2018h | 3018h | 4018h | 5018h | 6018h | 7018h |
| ALTFUNCCLR | 001Ch | 101Ch | 201Ch | 301Ch | 401Ch | 501Ch | 601Ch | 701Ch |
| INTENSET | 0020h | 1020h | 2020h | 3020h | 4020h | 5020h | 6020h | 7020h |
| INTENCLR | 0024h | 1024h | 2024h | 3024h | 4024h | 5024h | 6024h | 7024h |
| INTTYPESET | 0028h | 1028h | 2028h | 3028h | 4028h | 5028h | 6028h | 7028h |
| INTTYPECLR | 002Ch | 102Ch | 202Ch | 302Ch | 402Ch | 502Ch | 602Ch | 702Ch |
| INTPOLSET | 0030h | 1030h | 2030h | 3030h | 4030h | 5030h | 6030h | 7030h |
| INTPOLCLR | 0034h | 1034h | 2034h | 3034h | 4034h | 5034h | 6034h | 7034h |
| INTSTATUS, INTCLEAR | 0038h | 1038h | 2038h | 3038h | 4038h | 5038h | 6038h | 7038h |

Таблица Б.6 – Адреса регистров MASKLOWBYTE масок портов

| Маска | Порты | | | | | | | |
|-------|-----------------|-------|-------|-------|-------|-------|-------|-------|
| | Адрес 8001_xxxx | | | | | | | |
| | A | B | C | D | E | F | G | H |
| 00h | 0400h | 1400h | 2400h | 3400h | 4400h | 5400h | 6400h | 7400h |
| 01h | 0404h | 1404h | 2404h | 3404h | 4404h | 5404h | 6404h | 7404h |
| 02h | 0408h | 1408h | 2408h | 3408h | 4408h | 5408h | 6408h | 7408h |
| 03h | 040Ch | 140Ch | 240Ch | 340Ch | 440Ch | 540Ch | 640Ch | 740Ch |
| 04h | 0410h | 1410h | 2410h | 3410h | 4410h | 5410h | 6410h | 7410h |
| 05h | 0414h | 1414h | 2414h | 3414h | 4414h | 5414h | 6414h | 7414h |
| 06h | 0418h | 1418h | 2418h | 3418h | 4418h | 5418h | 6418h | 7418h |
| 07h | 041Ch | 141Ch | 241Ch | 341Ch | 441Ch | 541Ch | 641Ch | 741Ch |
| 08h | 0420h | 1420h | 2420h | 3420h | 4420h | 5420h | 6420h | 7420h |
| 09h | 0424h | 1424h | 2424h | 3424h | 4424h | 5424h | 6424h | 7424h |
| 0Ah | 0428h | 1428h | 2428h | 3428h | 4428h | 5428h | 6428h | 7428h |
| 0Bh | 042Ch | 142Ch | 242Ch | 342Ch | 442Ch | 542Ch | 642Ch | 742Ch |
| 0Ch | 0430h | 1430h | 2430h | 3430h | 4430h | 5430h | 6430h | 7430h |
| 0Dh | 0434h | 1434h | 2434h | 3434h | 4434h | 5434h | 6434h | 7434h |
| 0Eh | 0438h | 1438h | 2438h | 3438h | 4438h | 5438h | 6438h | 7438h |
| 0Fh | 043Ch | 143Ch | 243Ch | 343Ch | 443Ch | 543Ch | 643Ch | 743Ch |
| 10h | 0440h | 1440h | 2440h | 3440h | 4440h | 5440h | 6440h | 7440h |
| 11h | 0444h | 1444h | 2444h | 3444h | 4444h | 5444h | 6444h | 7444h |
| 12h | 0448h | 1448h | 2448h | 3448h | 4448h | 5448h | 6448h | 7448h |
| 13h | 044Ch | 144Ch | 244Ch | 344Ch | 444Ch | 544Ch | 644Ch | 744Ch |
| 14h | 0450h | 1450h | 2450h | 3450h | 4450h | 5450h | 6450h | 7450h |
| 15h | 0454h | 1454h | 2454h | 3454h | 4454h | 5454h | 6454h | 7454h |
| 16h | 0458h | 1458h | 2458h | 3458h | 4458h | 5458h | 6458h | 7458h |
| 17h | 045Ch | 145Ch | 245Ch | 345Ch | 445Ch | 545Ch | 645Ch | 745Ch |
| 18h | 0460h | 1460h | 2460h | 3460h | 4460h | 5460h | 6460h | 7460h |
| 19h | 0464h | 1464h | 2464h | 3464h | 4464h | 5464h | 6464h | 7464h |
| 1Ah | 0468h | 1468h | 2468h | 3468h | 4468h | 5468h | 6468h | 7468h |
| 1Bh | 046Ch | 146Ch | 246Ch | 346Ch | 446Ch | 546Ch | 646Ch | 746Ch |
| 1Ch | 0470h | 1470h | 2470h | 3470h | 4470h | 5470h | 6470h | 7470h |
| 1Dh | 0474h | 1474h | 2474h | 3474h | 4474h | 5474h | 6474h | 7474h |
| 1Eh | 0478h | 1478h | 2478h | 3478h | 4478h | 5478h | 6478h | 7478h |
| 1Fh | 047Ch | 147Ch | 247Ch | 347Ch | 447Ch | 547Ch | 647Ch | 747Ch |
| 20h | 0480h | 1480h | 2480h | 3480h | 4480h | 5480h | 6480h | 7480h |
| 21h | 0484h | 1484h | 2484h | 3484h | 4484h | 5484h | 6484h | 7484h |
| 22h | 0488h | 1488h | 2488h | 3488h | 4488h | 5488h | 6488h | 7488h |
| 23h | 048Ch | 148Ch | 248Ch | 348Ch | 448Ch | 548Ch | 648Ch | 748Ch |
| 24h | 0490h | 1490h | 2490h | 3490h | 4490h | 5490h | 6490h | 7490h |
| 25h | 0494h | 1494h | 2494h | 3494h | 4494h | 5494h | 6494h | 7494h |
| 26h | 0498h | 1498h | 2498h | 3498h | 4498h | 5498h | 6498h | 7498h |
| 27h | 049Ch | 149Ch | 249Ch | 349Ch | 449Ch | 549Ch | 649Ch | 749Ch |
| 28h | 04A0h | 14A0h | 24A0h | 34A0h | 44A0h | 54A0h | 64A0h | 74A0h |
| 29h | 04A4h | 14A4h | 24A4h | 34A4h | 44A4h | 54A4h | 64A4h | 74A4h |
| 2Ah | 04A8h | 14A8h | 24A8h | 34A8h | 44A8h | 54A8h | 64A8h | 74A8h |
| 2Bh | 04ACh | 14ACh | 24ACh | 34ACh | 44ACh | 54ACh | 64ACh | 74ACh |
| 2Ch | 04B0h | 14B0h | 24B0h | 34B0h | 44B0h | 54B0h | 64B0h | 74B0h |
| 2Dh | 04B4h | 14B4h | 24B4h | 34B4h | 44B4h | 54B4h | 64B4h | 74B4h |

Продолжение таблицы Б.6

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 2Eh | 04B8h | 14B8h | 24B8h | 34B8h | 44B8h | 54B8h | 64B8h | 74B8h |
| 2Fh | 04BCh | 14BCh | 24BCh | 34BCh | 44BCh | 54BCh | 64BCh | 74BCh |
| 30h | 04C0h | 14C0h | 24C0h | 34C0h | 44C0h | 54C0h | 64C0h | 74C0h |
| 31h | 04C4h | 14C4h | 24C4h | 34C4h | 44C4h | 54C4h | 64C4h | 74C4h |
| 32h | 04C8h | 14C8h | 24C8h | 34C8h | 44C8h | 54C8h | 64C8h | 74C8h |
| 33h | 04CCh | 14CCh | 24CCh | 34CCh | 44CCh | 54CCh | 64CCh | 74CCh |
| 34h | 04D0h | 14D0h | 24D0h | 34D0h | 44D0h | 54D0h | 64D0h | 74D0h |
| 35h | 04D4h | 14D4h | 24D4h | 34D4h | 44D4h | 54D4h | 64D4h | 74D4h |
| 36h | 04D8h | 14D8h | 24D8h | 34D8h | 44D8h | 54D8h | 64D8h | 74D8h |
| 37h | 04DCh | 14DCh | 24DCh | 34DCh | 44DCh | 54DCh | 64DCh | 74DCh |
| 38h | 04E0h | 14E0h | 24E0h | 34E0h | 44E0h | 54E0h | 64E0h | 74E0h |
| 39h | 04E4h | 14E4h | 24E4h | 34E4h | 44E4h | 54E4h | 64E4h | 74E4h |
| 3Ah | 04E8h | 14E8h | 24E8h | 34E8h | 44E8h | 54E8h | 64E8h | 74E8h |
| 3Bh | 04ECh | 14ECh | 24ECh | 34ECh | 44ECh | 54ECh | 64ECh | 74ECh |
| 3Ch | 04F0h | 14F0h | 24F0h | 34F0h | 44F0h | 54F0h | 64F0h | 74F0h |
| 3Dh | 04F4h | 14F4h | 24F4h | 34F4h | 44F4h | 54F4h | 64F4h | 74F4h |
| 3Eh | 04F8h | 14F8h | 24F8h | 34F8h | 44F8h | 54F8h | 64F8h | 74F8h |
| 3Fh | 04FCh | 14FCh | 24FCh | 34FCh | 44FCh | 54FCh | 64FCh | 74FCh |
| 40h | 0500h | 1500h | 2500h | 3500h | 4500h | 5500h | 6500h | 7500h |
| 41h | 0504h | 1504h | 2504h | 3504h | 4504h | 5504h | 6504h | 7504h |
| 42h | 0508h | 1508h | 2508h | 3508h | 4508h | 5508h | 6508h | 7508h |
| 43h | 050Ch | 150Ch | 250Ch | 350Ch | 450Ch | 550Ch | 650Ch | 750Ch |
| 44h | 0510h | 1510h | 2510h | 3510h | 4510h | 5510h | 6510h | 7510h |
| 45h | 0514h | 1514h | 2514h | 3514h | 4514h | 5514h | 6514h | 7514h |
| 46h | 0518h | 1518h | 2518h | 3518h | 4518h | 5518h | 6518h | 7518h |
| 47h | 051Ch | 151Ch | 251Ch | 351Ch | 451Ch | 551Ch | 651Ch | 751Ch |
| 48h | 0520h | 1520h | 2520h | 3520h | 4520h | 5520h | 6520h | 7520h |
| 49h | 0524h | 1524h | 2524h | 3524h | 4524h | 5524h | 6524h | 7524h |
| 4Ah | 0528h | 1528h | 2528h | 3528h | 4528h | 5528h | 6528h | 7528h |
| 4Bh | 052Ch | 152Ch | 252Ch | 352Ch | 452Ch | 552Ch | 652Ch | 752Ch |
| 4Ch | 0520h | 1520h | 2520h | 3520h | 4520h | 5520h | 6520h | 7520h |
| 4Dh | 0534h | 1534h | 2534h | 3534h | 4534h | 5534h | 6534h | 7534h |
| 4Eh | 0538h | 1538h | 2538h | 3538h | 4538h | 5538h | 6538h | 7538h |
| 4Fh | 053Ch | 153Ch | 253Ch | 353Ch | 453Ch | 553Ch | 653Ch | 753Ch |
| 50h | 0540h | 1540h | 2540h | 3540h | 4540h | 5540h | 6540h | 7540h |
| 51h | 0544h | 1544h | 2544h | 3544h | 4544h | 5544h | 6544h | 7544h |
| 52h | 0548h | 1548h | 2548h | 3548h | 4548h | 5548h | 6548h | 7548h |
| 53h | 054Ch | 154Ch | 254Ch | 354Ch | 454Ch | 554Ch | 654Ch | 754Ch |
| 54h | 0550h | 1550h | 2550h | 3550h | 4550h | 5550h | 6550h | 7550h |
| 55h | 0554h | 1554h | 2554h | 3554h | 4554h | 5554h | 6554h | 7554h |
| 56h | 0558h | 1558h | 2558h | 3558h | 4558h | 5558h | 6558h | 7558h |
| 57h | 055Ch | 155Ch | 255Ch | 355Ch | 455Ch | 555Ch | 655Ch | 755Ch |
| 58h | 0560h | 1560h | 2560h | 3560h | 4560h | 5560h | 6560h | 7560h |
| 59h | 0564h | 1564h | 2564h | 3564h | 4564h | 5564h | 6564h | 7564h |
| 5Ah | 0568h | 1568h | 2568h | 3568h | 4568h | 5568h | 6568h | 7568h |
| 5Bh | 056Ch | 156Ch | 256Ch | 356Ch | 456Ch | 556Ch | 656Ch | 756Ch |
| 5Ch | 0570h | 1570h | 2570h | 3570h | 4570h | 5570h | 6570h | 7570h |
| 5Dh | 0574h | 1574h | 2574h | 3574h | 4574h | 5574h | 6574h | 7574h |

Продолжение таблицы Б.6

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 5Eh | 0578h | 1578h | 2578h | 3578h | 4578h | 5578h | 6578h | 7578h |
| 5Fh | 057Ch | 157Ch | 257Ch | 357Ch | 457Ch | 557Ch | 657Ch | 757Ch |
| 60h | 0580h | 1580h | 2580h | 3580h | 4580h | 5580h | 6580h | 7580h |
| 61h | 0584h | 1584h | 2584h | 3584h | 4584h | 5584h | 6584h | 7584h |
| 62h | 0588h | 1588h | 2588h | 3588h | 4588h | 5588h | 6588h | 7588h |
| 63h | 058Ch | 158Ch | 258Ch | 358Ch | 458Ch | 558Ch | 658Ch | 758Ch |
| 64h | 0590h | 1590h | 2590h | 3590h | 4590h | 5590h | 6590h | 7590h |
| 65h | 0594h | 1594h | 2594h | 3594h | 4594h | 5594h | 6594h | 7594h |
| 66h | 0598h | 1598h | 2598h | 3598h | 4598h | 5598h | 6598h | 7598h |
| 67h | 059Ch | 159Ch | 259Ch | 359Ch | 459Ch | 559Ch | 659Ch | 759Ch |
| 68h | 05A0h | 15A0h | 25A0h | 35A0h | 45A0h | 55A0h | 65A0h | 75A0h |
| 69h | 05A4h | 15A4h | 25A4h | 35A4h | 45A4h | 55A4h | 65A4h | 75A4h |
| 6Ah | 05A8h | 15A8h | 25A8h | 35A8h | 45A8h | 55A8h | 65A8h | 75A8h |
| 6Bh | 05ACh | 15ACh | 25ACh | 35ACh | 45ACh | 55ACh | 65ACh | 75ACh |
| 6Ch | 05B0h | 15B0h | 25B0h | 35B0h | 45B0h | 55B0h | 65B0h | 75B0h |
| 6Dh | 05B4h | 15B4h | 25B4h | 35B4h | 45B4h | 55B4h | 65B4h | 75B4h |
| 6Eh | 05B8h | 15B8h | 25B8h | 35B8h | 45B8h | 55B8h | 65B8h | 75B8h |
| 6Fh | 05BCh | 15BCh | 25BCh | 35BCh | 45BCh | 55BCh | 65BCh | 75BCh |
| 70h | 05C0h | 15C0h | 25C0h | 35C0h | 45C0h | 55C0h | 65C0h | 75C0h |
| 71h | 05C4h | 15C4h | 25C4h | 35C4h | 45C4h | 55C4h | 65C4h | 75C4h |
| 72h | 05C8h | 15C8h | 25C8h | 35C8h | 45C8h | 55C8h | 65C8h | 75C8h |
| 73h | 05CCh | 15CCh | 25CCh | 35CCh | 45CCh | 55CCh | 65CCh | 75CCh |
| 74h | 05D0h | 15D0h | 25D0h | 35D0h | 45D0h | 55D0h | 65D0h | 75D0h |
| 75h | 05D4h | 15D4h | 25D4h | 35D4h | 45D4h | 55D4h | 65D4h | 75D4h |
| 76h | 05D8h | 15D8h | 25D8h | 35D8h | 45D8h | 55D8h | 65D8h | 75D8h |
| 77h | 05DCh | 15DCh | 25DCh | 35DCh | 45DCh | 55DCh | 65DCh | 75DCh |
| 78h | 05E0h | 15E0h | 25E0h | 35E0h | 45E0h | 55E0h | 65E0h | 75E0h |
| 79h | 05E4h | 15E4h | 25E4h | 35E4h | 45E4h | 55E4h | 65E4h | 75E4h |
| 7Ah | 05E8h | 15E8h | 25E8h | 35E8h | 45E8h | 55E8h | 65E8h | 75E8h |
| 7Bh | 05ECh | 15ECh | 25ECh | 35ECh | 45ECh | 55ECh | 65ECh | 75ECh |
| 7Ch | 05F0h | 15F0h | 25F0h | 35F0h | 45F0h | 55F0h | 65F0h | 75F0h |
| 7Dh | 05F4h | 15F4h | 25F4h | 35F4h | 45F4h | 55F4h | 65F4h | 75F4h |
| 7Eh | 05F8h | 15F8h | 25F8h | 35F8h | 45F8h | 55F8h | 65F8h | 75F8h |
| 7Fh | 05FCh | 15FCh | 25FCh | 35FCh | 45FCh | 55FCh | 65FCh | 75FCh |
| 80h | 0600h | 1600h | 2600h | 3600h | 4600h | 5600h | 6600h | 7600h |
| 81h | 0604h | 1604h | 2604h | 3604h | 4604h | 5604h | 6604h | 7604h |
| 82h | 0608h | 1608h | 2608h | 3608h | 4608h | 5608h | 6608h | 7608h |
| 83h | 060Ch | 160Ch | 260Ch | 360Ch | 460Ch | 560Ch | 660Ch | 760Ch |
| 84h | 0610h | 1610h | 2610h | 3610h | 4610h | 5610h | 6610h | 7610h |
| 85h | 0614h | 1614h | 2614h | 3614h | 4614h | 5614h | 6614h | 7614h |
| 86h | 0618h | 1618h | 2618h | 3618h | 4618h | 5618h | 6618h | 7618h |
| 87h | 061Ch | 161Ch | 261Ch | 361Ch | 461Ch | 561Ch | 661Ch | 761Ch |
| 88h | 0620h | 1620h | 2620h | 3620h | 4620h | 5620h | 6620h | 7620h |
| 89h | 0624h | 1624h | 2624h | 3624h | 4624h | 5624h | 6624h | 7624h |
| 8Ah | 0628h | 1628h | 2628h | 3628h | 4628h | 5628h | 6628h | 7628h |
| 8Bh | 062Ch | 162Ch | 262Ch | 362Ch | 462Ch | 562Ch | 662Ch | 762Ch |
| 8Ch | 0620h | 1620h | 2620h | 3620h | 4620h | 5620h | 6620h | 7620h |
| 8Dh | 0634h | 1634h | 2634h | 3634h | 4634h | 5634h | 6634h | 7634h |

Продолжение таблицы Б.6

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 8Eh | 0638h | 1638h | 2638h | 3638h | 4638h | 5638h | 6638h | 7638h |
| 8Fh | 063Ch | 163Ch | 263Ch | 363Ch | 463Ch | 563Ch | 663Ch | 763Ch |
| 90h | 0640h | 1640h | 2640h | 3640h | 4640h | 5640h | 6640h | 7640h |
| 91h | 0644h | 1644h | 2644h | 3644h | 4644h | 5644h | 6644h | 7644h |
| 92h | 0648h | 1648h | 2648h | 3648h | 4648h | 5648h | 6648h | 7648h |
| 93h | 064Ch | 164Ch | 264Ch | 364Ch | 464Ch | 564Ch | 664Ch | 764Ch |
| 94h | 0650h | 1650h | 2650h | 3650h | 4650h | 5650h | 6650h | 7650h |
| 95h | 0654h | 1654h | 2654h | 3654h | 4654h | 5654h | 6654h | 7654h |
| 96h | 0658h | 1658h | 2658h | 3658h | 4658h | 5658h | 6658h | 7658h |
| 97h | 065Ch | 165Ch | 265Ch | 365Ch | 465Ch | 565Ch | 665Ch | 765Ch |
| 98h | 0660h | 1660h | 2660h | 3660h | 4660h | 5660h | 6660h | 7660h |
| 99h | 0664h | 1664h | 2664h | 3664h | 4664h | 5664h | 6664h | 7664h |
| 9Ah | 0668h | 1668h | 2668h | 3668h | 4668h | 5668h | 6668h | 7668h |
| 9Bh | 066Ch | 166Ch | 266Ch | 366Ch | 466Ch | 566Ch | 666Ch | 766Ch |
| 9Ch | 0670h | 1670h | 2670h | 3670h | 4670h | 5670h | 6670h | 7670h |
| 9Dh | 0674h | 1674h | 2674h | 3674h | 4674h | 5674h | 6674h | 7674h |
| 9Eh | 0678h | 1678h | 2678h | 3678h | 4678h | 5678h | 6678h | 7678h |
| 9Fh | 067Ch | 167Ch | 267Ch | 367Ch | 467Ch | 567Ch | 667Ch | 767Ch |
| A0h | 0680h | 1680h | 2680h | 3680h | 4680h | 5680h | 6680h | 7680h |
| A1h | 0684h | 1684h | 2684h | 3684h | 4684h | 5684h | 6684h | 7684h |
| A2h | 0688h | 1688h | 2688h | 3688h | 4688h | 5688h | 6688h | 7688h |
| A3h | 068Ch | 168Ch | 268Ch | 368Ch | 468Ch | 568Ch | 668Ch | 768Ch |
| A4h | 0690h | 1690h | 2690h | 3690h | 4690h | 5690h | 6690h | 7690h |
| A5h | 0694h | 1694h | 2694h | 3694h | 4694h | 5694h | 6694h | 7694h |
| A6h | 0698h | 1698h | 2698h | 3698h | 4698h | 5698h | 6698h | 7698h |
| A7h | 069Ch | 169Ch | 269Ch | 369Ch | 469Ch | 569Ch | 669Ch | 769Ch |
| A8h | 06A0h | 16A0h | 26A0h | 36A0h | 46A0h | 56A0h | 66A0h | 76A0h |
| A9h | 06A4h | 16A4h | 26A4h | 36A4h | 46A4h | 56A4h | 66A4h | 76A4h |
| AAh | 06A8h | 16A8h | 26A8h | 36A8h | 46A8h | 56A8h | 66A8h | 76A8h |
| Abh | 06Ach | 16Ach | 26Ach | 36Ach | 46Ach | 56Ach | 66Ach | 76Ach |
| ACh | 06B0h | 16B0h | 26B0h | 36B0h | 46B0h | 56B0h | 66B0h | 76B0h |
| Adh | 06B4h | 16B4h | 26B4h | 36B4h | 46B4h | 56B4h | 66B4h | 76B4h |
| Aeh | 06B8h | 16B8h | 26B8h | 36B8h | 46B8h | 56B8h | 66B8h | 76B8h |
| Afh | 06BCh | 16BCh | 26BCh | 36BCh | 46BCh | 56BCh | 66BCh | 76BCh |
| B0h | 06C0h | 16C0h | 26C0h | 36C0h | 46C0h | 56C0h | 66C0h | 76C0h |
| B1h | 06C4h | 16C4h | 26C4h | 36C4h | 46C4h | 56C4h | 66C4h | 76C4h |
| B2h | 06C8h | 16C8h | 26C8h | 36C8h | 46C8h | 56C8h | 66C8h | 76C8h |
| B3h | 06CCh | 16CCh | 26CCh | 36CCh | 46CCh | 56CCh | 66CCh | 76CCh |
| B4h | 06D0h | 16D0h | 26D0h | 36D0h | 46D0h | 56D0h | 66D0h | 76D0h |
| B5h | 06D4h | 16D4h | 26D4h | 36D4h | 46D4h | 56D4h | 66D4h | 76D4h |
| B6h | 06D8h | 16D8h | 26D8h | 36D8h | 46D8h | 56D8h | 66D8h | 76D8h |
| B7h | 06DCh | 16DCh | 26DCh | 36DCh | 46DCh | 56DCh | 66DCh | 76DCh |
| B8h | 06E0h | 16E0h | 26E0h | 36E0h | 46E0h | 56E0h | 66E0h | 76E0h |
| B9h | 06E4h | 16E4h | 26E4h | 36E4h | 46E4h | 56E4h | 66E4h | 76E4h |
| Bah | 06E8h | 16E8h | 26E8h | 36E8h | 46E8h | 56E8h | 66E8h | 76E8h |
| BBh | 06Ech | 16Ech | 26Ech | 36Ech | 46Ech | 56Ech | 66Ech | 76Ech |
| BCh | 06F0h | 16F0h | 26F0h | 36F0h | 46F0h | 56F0h | 66F0h | 76F0h |
| BDh | 06F4h | 16F4h | 26F4h | 36F4h | 46F4h | 56F4h | 66F4h | 76F4h |

Продолжение таблицы Б.6

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| BEh | 06F8h | 16F8h | 26F8h | 36F8h | 46F8h | 56F8h | 66F8h | 76F8h |
| BFh | 06FCh | 16FCh | 26FCh | 36FCh | 46FCh | 56FCh | 66FCh | 76FCh |
| C0h | 0700h | 1700h | 2700h | 3700h | 4700h | 5700h | 6700h | 7700h |
| C1h | 0704h | 1704h | 2704h | 3704h | 4704h | 5704h | 6704h | 7704h |
| C2h | 0708h | 1708h | 2708h | 3708h | 4708h | 5708h | 6708h | 7708h |
| C3h | 070Ch | 170Ch | 270Ch | 370Ch | 470Ch | 570Ch | 670Ch | 770Ch |
| C4h | 0710h | 1710h | 2710h | 3710h | 4710h | 5710h | 6710h | 7710h |
| C5h | 0714h | 1714h | 2714h | 3714h | 4714h | 5714h | 6714h | 7714h |
| C6h | 0718h | 1718h | 2718h | 3718h | 4718h | 5718h | 6718h | 7718h |
| C7h | 071Ch | 171Ch | 271Ch | 371Ch | 471Ch | 571Ch | 671Ch | 771Ch |
| C8h | 0720h | 1720h | 2720h | 3720h | 4720h | 5720h | 6720h | 7720h |
| C9h | 0724h | 1724h | 2724h | 3724h | 4724h | 5724h | 6724h | 7724h |
| CAh | 0728h | 1728h | 2728h | 3728h | 4728h | 5728h | 6728h | 7728h |
| CBh | 072Ch | 172Ch | 272Ch | 372Ch | 472Ch | 572Ch | 672Ch | 772Ch |
| CCh | 0720h | 1720h | 2720h | 3720h | 4720h | 5720h | 6720h | 7720h |
| CDh | 0734h | 1734h | 2734h | 3734h | 4734h | 5734h | 6734h | 7734h |
| CEh | 0738h | 1738h | 2738h | 3738h | 4738h | 5738h | 6738h | 7738h |
| CFh | 073Ch | 173Ch | 273Ch | 373Ch | 473Ch | 573Ch | 673Ch | 773Ch |
| D0h | 0740h | 1740h | 2740h | 3740h | 4740h | 5740h | 6740h | 7740h |
| D1h | 0744h | 1744h | 2744h | 3744h | 4744h | 5744h | 6744h | 7744h |
| D2h | 0748h | 1748h | 2748h | 3748h | 4748h | 5748h | 6748h | 7748h |
| D3h | 074Ch | 174Ch | 274Ch | 374Ch | 474Ch | 574Ch | 674Ch | 774Ch |
| D4h | 0750h | 1750h | 2750h | 3750h | 4750h | 5750h | 6750h | 7750h |
| D5h | 0754h | 1754h | 2754h | 3754h | 4754h | 5754h | 6754h | 7754h |
| D6h | 0758h | 1758h | 2758h | 3758h | 4758h | 5758h | 6758h | 7758h |
| D7h | 075Ch | 175Ch | 275Ch | 375Ch | 475Ch | 575Ch | 675Ch | 775Ch |
| D8h | 0760h | 1760h | 2760h | 3760h | 4760h | 5760h | 6760h | 7760h |
| D9h | 0764h | 1764h | 2764h | 3764h | 4764h | 5764h | 6764h | 7764h |
| DAh | 0768h | 1768h | 2768h | 3768h | 4768h | 5768h | 6768h | 7768h |
| DBh | 076Ch | 176Ch | 276Ch | 376Ch | 476Ch | 576Ch | 676Ch | 776Ch |
| DCh | 0770h | 1770h | 2770h | 3770h | 4770h | 5770h | 6770h | 7770h |
| DDh | 0774h | 1774h | 2774h | 3774h | 4774h | 5774h | 6774h | 7774h |
| DEh | 0778h | 1778h | 2778h | 3778h | 4778h | 5778h | 6778h | 7778h |
| DFh | 077Ch | 177Ch | 277Ch | 377Ch | 477Ch | 577Ch | 677Ch | 777Ch |
| E0h | 0780h | 1780h | 2780h | 3780h | 4780h | 5780h | 6780h | 7780h |
| E1h | 0784h | 1784h | 2784h | 3784h | 4784h | 5784h | 6784h | 7784h |
| E2h | 0788h | 1788h | 2788h | 3788h | 4788h | 5788h | 6788h | 7788h |
| E3h | 078Ch | 178Ch | 278Ch | 378Ch | 478Ch | 578Ch | 678Ch | 778Ch |
| E4h | 0790h | 1790h | 2790h | 3790h | 4790h | 5790h | 6790h | 7790h |
| E5h | 0794h | 1794h | 2794h | 3794h | 4794h | 5794h | 6794h | 7794h |
| E6h | 0798h | 1798h | 2798h | 3798h | 4798h | 5798h | 6798h | 7798h |
| E7h | 079Ch | 179Ch | 279Ch | 379Ch | 479Ch | 579Ch | 679Ch | 779Ch |
| E8h | 07A0h | 17A0h | 27A0h | 37A0h | 47A0h | 57A0h | 67A0h | 77A0h |
| E9h | 07A4h | 17A4h | 27A4h | 37A4h | 47A4h | 57A4h | 67A4h | 77A4h |
| EAh | 07A8h | 17A8h | 27A8h | 37A8h | 47A8h | 57A8h | 67A8h | 77A8h |
| EBh | 07ACh | 17ACh | 27ACh | 37ACh | 47ACh | 57ACh | 67ACh | 77ACh |
| ECh | 07B0h | 17B0h | 27B0h | 37B0h | 47B0h | 57B0h | 67B0h | 77B0h |
| EDh | 07B4h | 17B4h | 27B4h | 37B4h | 47B4h | 57B4h | 67B4h | 77B4h |

Окончание таблицы Б.6

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| EEh | 07B8h | 17B8h | 27B8h | 37B8h | 47B8h | 57B8h | 67B8h | 77B8h |
| EFh | 07BCh | 17BCh | 27BCh | 37BCh | 47BCh | 57BCh | 67BCh | 77BCh |
| F0h | 07C0h | 17C0h | 27C0h | 37C0h | 47C0h | 57C0h | 67C0h | 77C0h |
| F1h | 07C4h | 17C4h | 27C4h | 37C4h | 47C4h | 57C4h | 67C4h | 77C4h |
| F2h | 07C8h | 17C8h | 27C8h | 37C8h | 47C8h | 57C8h | 67C8h | 77C8h |
| F3h | 07CCh | 17CCh | 27CCh | 37CCh | 47CCh | 57CCh | 67CCh | 77CCh |
| F4h | 07D0h | 17D0h | 27D0h | 37D0h | 47D0h | 57D0h | 67D0h | 77D0h |
| F5h | 07D4h | 17D4h | 27D4h | 37D4h | 47D4h | 57D4h | 67D4h | 77D4h |
| F6h | 07D8h | 17D8h | 27D8h | 37D8h | 47D8h | 57D8h | 67D8h | 77D8h |
| F7h | 07DCh | 17DCh | 27DCh | 37DCh | 47DCh | 57DCh | 67DCh | 77DCh |
| F8h | 07E0h | 17E0h | 27E0h | 37E0h | 47E0h | 57E0h | 67E0h | 77E0h |
| F9h | 07E4h | 17E4h | 27E4h | 37E4h | 47E4h | 57E4h | 67E4h | 77E4h |
| FAh | 07E8h | 17E8h | 27E8h | 37E8h | 47E8h | 57E8h | 67E8h | 77E8h |
| FBh | 07ECh | 17ECh | 27ECh | 37ECh | 47ECh | 57ECh | 67ECh | 77ECh |
| FCh | 07F0h | 17F0h | 27F0h | 37F0h | 47F0h | 57F0h | 67F0h | 77F0h |
| FDh | 07F4h | 17F4h | 27F4h | 37F4h | 47F4h | 57F4h | 67F4h | 77F4h |
| FEh | 07F8h | 17F8h | 27F8h | 37F8h | 47F8h | 57F8h | 67F8h | 77F8h |
| FFh | 07FCh | 17FCh | 27FCh | 37FCh | 47FCh | 57FCh | 67FCh | 77FCh |

Таблица Б.7 – Адреса регистров MASKHIGHBYTE масок портов

| Маска | Порты | | | | | | | |
|-------|-----------------|-------|-------|-------|-------|-------|-------|-------|
| | Адрес 8001_xxxx | | | | | | | |
| | A | B | C | D | E | F | G | H |
| 00h | 0800h | 1800h | 2800h | 3800h | 4800h | 5800h | 6800h | 7800h |
| 01h | 0804h | 1804h | 2804h | 3804h | 4804h | 5804h | 6804h | 7804h |
| 02h | 0808h | 1808h | 2808h | 3808h | 4808h | 5808h | 6808h | 7808h |
| 03h | 080Ch | 180Ch | 280Ch | 380Ch | 480Ch | 580Ch | 680Ch | 780Ch |
| 04h | 0810h | 1810h | 2810h | 3810h | 4810h | 5810h | 6810h | 7810h |
| 05h | 0814h | 1814h | 2814h | 3814h | 4814h | 5814h | 6814h | 7814h |
| 06h | 0818h | 1818h | 2818h | 3818h | 4818h | 5818h | 6818h | 7818h |
| 07h | 081Ch | 181Ch | 281Ch | 381Ch | 481Ch | 581Ch | 681Ch | 781Ch |
| 08h | 0820h | 1820h | 2820h | 3820h | 4820h | 5820h | 6820h | 7820h |
| 09h | 0824h | 1824h | 2824h | 3824h | 4824h | 5824h | 6824h | 7824h |
| 0Ah | 0828h | 1828h | 2828h | 3828h | 4828h | 5828h | 6828h | 7828h |
| 0Bh | 082Ch | 182Ch | 282Ch | 382Ch | 482Ch | 582Ch | 682Ch | 782Ch |
| 0Ch | 0820h | 1820h | 2820h | 3820h | 4820h | 5820h | 6820h | 7820h |
| 0Dh | 0834h | 1834h | 2834h | 3834h | 4834h | 5834h | 6834h | 7834h |
| 0Eh | 0838h | 1838h | 2838h | 3838h | 4838h | 5838h | 6838h | 7838h |
| 0Fh | 083Ch | 183Ch | 283Ch | 383Ch | 483Ch | 583Ch | 683Ch | 783Ch |
| 10h | 0840h | 1840h | 2840h | 3840h | 4840h | 5840h | 6840h | 7840h |
| 11h | 0844h | 1844h | 2844h | 3844h | 4844h | 5844h | 6844h | 7844h |
| 12h | 0848h | 1848h | 2848h | 3848h | 4848h | 5848h | 6848h | 7848h |
| 13h | 084Ch | 184Ch | 284Ch | 384Ch | 484Ch | 584Ch | 684Ch | 784Ch |
| 14h | 0850h | 1850h | 2850h | 3850h | 4850h | 5850h | 6850h | 7850h |
| 15h | 0854h | 1854h | 2854h | 3854h | 4854h | 5854h | 6854h | 7854h |
| 16h | 0858h | 1858h | 2858h | 3858h | 4858h | 5858h | 6858h | 7858h |
| 17h | 085Ch | 185Ch | 285Ch | 385Ch | 485Ch | 585Ch | 685Ch | 785Ch |
| 18h | 0860h | 1860h | 2860h | 3860h | 4860h | 5860h | 6860h | 7860h |

Продолжение таблицы Б.7

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 19h | 0864h | 1864h | 2864h | 3864h | 4864h | 5864h | 6864h | 7864h |
| 1Ah | 0868h | 1868h | 2868h | 3868h | 4868h | 5868h | 6868h | 7868h |
| 1Bh | 086Ch | 186Ch | 286Ch | 386Ch | 486Ch | 586Ch | 686Ch | 786Ch |
| 1Ch | 0870h | 1870h | 2870h | 3870h | 4870h | 5870h | 6870h | 7870h |
| 1Dh | 0874h | 1874h | 2874h | 3874h | 4874h | 5874h | 6874h | 7874h |
| 1Eh | 0878h | 1878h | 2878h | 3878h | 4878h | 5878h | 6878h | 7878h |
| 1Fh | 087Ch | 187Ch | 287Ch | 387Ch | 487Ch | 587Ch | 687Ch | 787Ch |
| 20h | 0880h | 1880h | 2880h | 3880h | 4880h | 5880h | 6880h | 7880h |
| 21h | 0884h | 1884h | 2884h | 3884h | 4884h | 5884h | 6884h | 7884h |
| 22h | 0888h | 1888h | 2888h | 3888h | 4888h | 5888h | 6888h | 7888h |
| 23h | 088Ch | 188Ch | 288Ch | 388Ch | 488Ch | 588Ch | 688Ch | 788Ch |
| 24h | 0890h | 1890h | 2890h | 3890h | 4890h | 5890h | 6890h | 7890h |
| 25h | 0894h | 1894h | 2894h | 3894h | 4894h | 5894h | 6894h | 7894h |
| 26h | 0898h | 1898h | 2898h | 3898h | 4898h | 5898h | 6898h | 7898h |
| 27h | 089Ch | 189Ch | 289Ch | 389Ch | 489Ch | 589Ch | 689Ch | 789Ch |
| 28h | 08A0h | 18A0h | 28A0h | 38A0h | 48A0h | 58A0h | 68A0h | 78A0h |
| 29h | 08A4h | 18A4h | 28A4h | 38A4h | 48A4h | 58A4h | 68A4h | 78A4h |
| 2Ah | 08A8h | 18A8h | 28A8h | 38A8h | 48A8h | 58A8h | 68A8h | 78A8h |
| 2Bh | 08ACh | 18ACh | 28ACh | 38ACh | 48ACh | 58ACh | 68ACh | 78ACh |
| 2Ch | 08B0h | 18B0h | 28B0h | 38B0h | 48B0h | 58B0h | 68B0h | 78B0h |
| 2Dh | 08B4h | 18B4h | 28B4h | 38B4h | 48B4h | 58B4h | 68B4h | 78B4h |
| 2Eh | 08B8h | 18B8h | 28B8h | 38B8h | 48B8h | 58B8h | 68B8h | 78B8h |
| 2Fh | 08BCh | 18BCh | 28BCh | 38BCh | 48BCh | 58BCh | 68BCh | 78BCh |
| 30h | 08C0h | 18C0h | 28C0h | 38C0h | 48C0h | 58C0h | 68C0h | 78C0h |
| 31h | 08C4h | 18C4h | 28C4h | 38C4h | 48C4h | 58C4h | 68C4h | 78C4h |
| 32h | 08C8h | 18C8h | 28C8h | 38C8h | 48C8h | 58C8h | 68C8h | 78C8h |
| 33h | 08CCh | 18CCh | 28CCh | 38CCh | 48CCh | 58CCh | 68CCh | 78CCh |
| 34h | 08D0h | 18D0h | 28D0h | 38D0h | 48D0h | 58D0h | 68D0h | 78D0h |
| 35h | 08D4h | 18D4h | 28D4h | 38D4h | 48D4h | 58D4h | 68D4h | 78D4h |
| 36h | 08D8h | 18D8h | 28D8h | 38D8h | 48D8h | 58D8h | 68D8h | 78D8h |
| 37h | 08DCh | 18DCh | 28DCh | 38DCh | 48DCh | 58DCh | 68DCh | 78DCh |
| 38h | 08E0h | 18E0h | 28E0h | 38E0h | 48E0h | 58E0h | 68E0h | 78E0h |
| 39h | 08E4h | 18E4h | 28E4h | 38E4h | 48E4h | 58E4h | 68E4h | 78E4h |
| 3Ah | 08E8h | 18E8h | 28E8h | 38E8h | 48E8h | 58E8h | 68E8h | 78E8h |
| 3Bh | 08ECh | 18ECh | 28ECh | 38ECh | 48ECh | 58ECh | 68ECh | 78ECh |
| 3Ch | 08F0h | 18F0h | 28F0h | 38F0h | 48F0h | 58F0h | 68F0h | 78F0h |
| 3Dh | 08F4h | 18F4h | 28F4h | 38F4h | 48F4h | 58F4h | 68F4h | 78F4h |
| 3Eh | 08F8h | 18F8h | 28F8h | 38F8h | 48F8h | 58F8h | 68F8h | 78F8h |
| 3Fh | 08FCh | 18FCh | 28FCh | 38FCh | 48FCh | 58FCh | 68FCh | 78FCh |
| 40h | 0900h | 1900h | 2900h | 3900h | 4900h | 5900h | 6900h | 7900h |
| 41h | 0904h | 1904h | 2904h | 3904h | 4904h | 5904h | 6904h | 7904h |
| 42h | 0908h | 1908h | 2908h | 3908h | 4908h | 5908h | 6908h | 7908h |
| 43h | 090Ch | 190Ch | 290Ch | 390Ch | 490Ch | 590Ch | 690Ch | 790Ch |
| 44h | 0910h | 1910h | 2910h | 3910h | 4910h | 5910h | 6910h | 7910h |
| 45h | 0914h | 1914h | 2914h | 3914h | 4914h | 5914h | 6914h | 7914h |
| 46h | 0918h | 1918h | 2918h | 3918h | 4918h | 5918h | 6918h | 7918h |
| 47h | 091Ch | 191Ch | 291Ch | 391Ch | 491Ch | 591Ch | 691Ch | 791Ch |
| 48h | 0920h | 1920h | 2920h | 3920h | 4920h | 5920h | 6920h | 7920h |

Продолжение таблицы Б.7

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 49h | 0924h | 1924h | 2924h | 3924h | 4924h | 5924h | 6924h | 7924h |
| 4Ah | 0928h | 1928h | 2928h | 3928h | 4928h | 5928h | 6928h | 7928h |
| 4Bh | 092Ch | 192Ch | 292Ch | 392Ch | 492Ch | 592Ch | 692Ch | 792Ch |
| 4Ch | 0920h | 1920h | 2920h | 3920h | 4920h | 5920h | 6920h | 7920h |
| 4Dh | 0934h | 1934h | 2934h | 3934h | 4934h | 5934h | 6934h | 7934h |
| 4Eh | 0938h | 1938h | 2938h | 3938h | 4938h | 5938h | 6938h | 7938h |
| 4Fh | 093Ch | 193Ch | 293Ch | 393Ch | 493Ch | 593Ch | 693Ch | 793Ch |
| 50h | 0940h | 1940h | 2940h | 3940h | 4940h | 5940h | 6940h | 7940h |
| 51h | 0944h | 1944h | 2944h | 3944h | 4944h | 5944h | 6944h | 7944h |
| 52h | 0948h | 1948h | 2948h | 3948h | 4948h | 5948h | 6948h | 7948h |
| 53h | 094Ch | 194Ch | 294Ch | 394Ch | 494Ch | 594Ch | 694Ch | 794Ch |
| 54h | 0950h | 1950h | 2950h | 3950h | 4950h | 5950h | 6950h | 7950h |
| 55h | 0954h | 1954h | 2954h | 3954h | 4954h | 5954h | 6954h | 7954h |
| 56h | 0958h | 1958h | 2958h | 3958h | 4958h | 5958h | 6958h | 7958h |
| 57h | 095Ch | 195Ch | 295Ch | 395Ch | 495Ch | 595Ch | 695Ch | 795Ch |
| 58h | 0960h | 1960h | 2960h | 3960h | 4960h | 5960h | 6960h | 7960h |
| 59h | 0964h | 1964h | 2964h | 3964h | 4964h | 5964h | 6964h | 7964h |
| 5Ah | 0968h | 1968h | 2968h | 3968h | 4968h | 5968h | 6968h | 7968h |
| 5Bh | 096Ch | 196Ch | 296Ch | 396Ch | 496Ch | 596Ch | 696Ch | 796Ch |
| 5Ch | 0970h | 1970h | 2970h | 3970h | 4970h | 5970h | 6970h | 7970h |
| 5Dh | 0974h | 1974h | 2974h | 3974h | 4974h | 5974h | 6974h | 7974h |
| 5Eh | 0978h | 1978h | 2978h | 3978h | 4978h | 5978h | 6978h | 7978h |
| 5Fh | 097Ch | 197Ch | 297Ch | 397Ch | 497Ch | 597Ch | 697Ch | 797Ch |
| 60h | 0980h | 1980h | 2980h | 3980h | 4980h | 5980h | 6980h | 7980h |
| 61h | 0984h | 1984h | 2984h | 3984h | 4984h | 5984h | 6984h | 7984h |
| 62h | 0988h | 1988h | 2988h | 3988h | 4988h | 5988h | 6988h | 7988h |
| 63h | 098Ch | 198Ch | 298Ch | 398Ch | 498Ch | 598Ch | 698Ch | 798Ch |
| 64h | 0990h | 1990h | 2990h | 3990h | 4990h | 5990h | 6990h | 7990h |
| 65h | 0994h | 1994h | 2994h | 3994h | 4994h | 5994h | 6994h | 7994h |
| 66h | 0998h | 1998h | 2998h | 3998h | 4998h | 5998h | 6998h | 7998h |
| 67h | 099Ch | 199Ch | 299Ch | 399Ch | 499Ch | 599Ch | 699Ch | 799Ch |
| 68h | 09A0h | 19A0h | 29A0h | 39A0h | 49A0h | 59A0h | 69A0h | 79A0h |
| 69h | 09A4h | 19A4h | 29A4h | 39A4h | 49A4h | 59A4h | 69A4h | 79A4h |
| 6Ah | 09A8h | 19A8h | 29A8h | 39A8h | 49A8h | 59A8h | 69A8h | 79A8h |
| 6Bh | 09ACh | 19ACh | 29ACh | 39ACh | 49ACh | 59ACh | 69ACh | 79ACh |
| 6Ch | 09B0h | 19B0h | 29B0h | 39B0h | 49B0h | 59B0h | 69B0h | 79B0h |
| 6Dh | 09B4h | 19B4h | 29B4h | 39B4h | 49B4h | 59B4h | 69B4h | 79B4h |
| 6Eh | 09B8h | 19B8h | 29B8h | 39B8h | 49B8h | 59B8h | 69B8h | 79B8h |
| 6Fh | 09BCh | 19BCh | 29BCh | 39BCh | 49BCh | 59BCh | 69BCh | 79BCh |
| 70h | 09C0h | 19C0h | 29C0h | 39C0h | 49C0h | 59C0h | 69C0h | 79C0h |
| 71h | 09C4h | 19C4h | 29C4h | 39C4h | 49C4h | 59C4h | 69C4h | 79C4h |
| 72h | 09C8h | 19C8h | 29C8h | 39C8h | 49C8h | 59C8h | 69C8h | 79C8h |
| 73h | 09CCh | 19CCh | 29CCh | 39CCh | 49CCh | 59CCh | 69CCh | 79CCh |
| 74h | 09D0h | 19D0h | 29D0h | 39D0h | 49D0h | 59D0h | 69D0h | 79D0h |
| 75h | 09D4h | 19D4h | 29D4h | 39D4h | 49D4h | 59D4h | 69D4h | 79D4h |
| 76h | 09D8h | 19D8h | 29D8h | 39D8h | 49D8h | 59D8h | 69D8h | 79D8h |
| 77h | 09DCh | 19DCh | 29DCh | 39DCh | 49DCh | 59DCh | 69DCh | 79DCh |
| 78h | 09E0h | 19E0h | 29E0h | 39E0h | 49E0h | 59E0h | 69E0h | 79E0h |

Продолжение таблицы Б.7

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 79h | 09E4h | 19E4h | 29E4h | 39E4h | 49E4h | 59E4h | 69E4h | 79E4h |
| 7Ah | 09E8h | 19E8h | 29E8h | 39E8h | 49E8h | 59E8h | 69E8h | 79E8h |
| 7Bh | 09ECh | 19ECh | 29ECh | 39ECh | 49ECh | 59ECh | 69ECh | 79ECh |
| 7Ch | 09F0h | 19F0h | 29F0h | 39F0h | 49F0h | 59F0h | 69F0h | 79F0h |
| 7Dh | 09F4h | 19F4h | 29F4h | 39F4h | 49F4h | 59F4h | 69F4h | 79F4h |
| 7Eh | 09F8h | 19F8h | 29F8h | 39F8h | 49F8h | 59F8h | 69F8h | 79F8h |
| 7Fh | 09FCh | 19FCh | 29FCh | 39FCh | 49FCh | 59FCh | 69FCh | 79FCh |
| 80h | 0A00h | 1A00h | 2A00h | 3A00h | 4A00h | 5A00h | 6A00h | 7A00h |
| 81h | 0A04h | 1A04h | 2A04h | 3A04h | 4A04h | 5A04h | 6A04h | 7A04h |
| 82h | 0A08h | 1A08h | 2A08h | 3A08h | 4A08h | 5A08h | 6A08h | 7A08h |
| 83h | 0A0Ch | 1A0Ch | 2A0Ch | 3A0Ch | 4A0Ch | 5A0Ch | 6A0Ch | 7A0Ch |
| 84h | 0A10h | 1A10h | 2A10h | 3A10h | 4A10h | 5A10h | 6A10h | 7A10h |
| 85h | 0A14h | 1A14h | 2A14h | 3A14h | 4A14h | 5A14h | 6A14h | 7A14h |
| 86h | 0A18h | 1A18h | 2A18h | 3A18h | 4A18h | 5A18h | 6A18h | 7A18h |
| 87h | 0A1Ch | 1A1Ch | 2A1Ch | 3A1Ch | 4A1Ch | 5A1Ch | 6A1Ch | 7A1Ch |
| 88h | 0A20h | 1A20h | 2A20h | 3A20h | 4A20h | 5A20h | 6A20h | 7A20h |
| 89h | 0A24h | 1A24h | 2A24h | 3A24h | 4A24h | 5A24h | 6A24h | 7A24h |
| 8Ah | 0A28h | 1A28h | 2A28h | 3A28h | 4A28h | 5A28h | 6A28h | 7A28h |
| 8Bh | 0A2Ch | 1A2Ch | 2A2Ch | 3A2Ch | 4A2Ch | 5A2Ch | 6A2Ch | 7A2Ch |
| 8Ch | 0A20h | 1A20h | 2A20h | 3A20h | 4A20h | 5A20h | 6A20h | 7A20h |
| 8Dh | 0A34h | 1A34h | 2A34h | 3A34h | 4A34h | 5A34h | 6A34h | 7A34h |
| 8Eh | 0A38h | 1A38h | 2A38h | 3A38h | 4A38h | 5A38h | 6A38h | 7A38h |
| 8Fh | 0A3Ch | 1A3Ch | 2A3Ch | 3A3Ch | 4A3Ch | 5A3Ch | 6A3Ch | 7A3Ch |
| 90h | 0A40h | 1A40h | 2A40h | 3A40h | 4A40h | 5A40h | 6A40h | 7A40h |
| 91h | 0A44h | 1A44h | 2A44h | 3A44h | 4A44h | 5A44h | 6A44h | 7A44h |
| 92h | 0A48h | 1A48h | 2A48h | 3A48h | 4A48h | 5A48h | 6A48h | 7A48h |
| 93h | 0A4Ch | 1A4Ch | 2A4Ch | 3A4Ch | 4A4Ch | 5A4Ch | 6A4Ch | 7A4Ch |
| 94h | 0A50h | 1A50h | 2A50h | 3A50h | 4A50h | 5A50h | 6A50h | 7A50h |
| 95h | 0A54h | 1A54h | 2A54h | 3A54h | 4A54h | 5A54h | 6A54h | 7A54h |
| 96h | 0A58h | 1A58h | 2A58h | 3A58h | 4A58h | 5A58h | 6A58h | 7A58h |
| 97h | 0A5Ch | 1A5Ch | 2A5Ch | 3A5Ch | 4A5Ch | 5A5Ch | 6A5Ch | 7A5Ch |
| 98h | 0A60h | 1A60h | 2A60h | 3A60h | 4A60h | 5A60h | 6A60h | 7A60h |
| 99h | 0A64h | 1A64h | 2A64h | 3A64h | 4A64h | 5A64h | 6A64h | 7A64h |
| 9Ah | 0A68h | 1A68h | 2A68h | 3A68h | 4A68h | 5A68h | 6A68h | 7A68h |
| 9Bh | 0A6Ch | 1A6Ch | 2A6Ch | 3A6Ch | 4A6Ch | 5A6Ch | 6A6Ch | 7A6Ch |
| 9Ch | 0A70h | 1A70h | 2A70h | 3A70h | 4A70h | 5A70h | 6A70h | 7A70h |
| 9Dh | 0A74h | 1A74h | 2A74h | 3A74h | 4A74h | 5A74h | 6A74h | 7A74h |
| 9Eh | 0A78h | 1A78h | 2A78h | 3A78h | 4A78h | 5A78h | 6A78h | 7A78h |
| 9Fh | 0A7Ch | 1A7Ch | 2A7Ch | 3A7Ch | 4A7Ch | 5A7Ch | 6A7Ch | 7A7Ch |
| A0h | 0A80h | 1A80h | 2A80h | 3A80h | 4A80h | 5A80h | 6A80h | 7A80h |
| A1h | 0A84h | 1A84h | 2A84h | 3A84h | 4A84h | 5A84h | 6A84h | 7A84h |
| A2h | 0A88h | 1A88h | 2A88h | 3A88h | 4A88h | 5A88h | 6A88h | 7A88h |
| A3h | 0A8Ch | 1A8Ch | 2A8Ch | 3A8Ch | 4A8Ch | 5A8Ch | 6A8Ch | 7A8Ch |
| A4h | 0A90h | 1A90h | 2A90h | 3A90h | 4A90h | 5A90h | 6A90h | 7A90h |
| A5h | 0A94h | 1A94h | 2A94h | 3A94h | 4A94h | 5A94h | 6A94h | 7A94h |
| A6h | 0A98h | 1A98h | 2A98h | 3A98h | 4A98h | 5A98h | 6A98h | 7A98h |
| A7h | 0A9Ch | 1A9Ch | 2A9Ch | 3A9Ch | 4A9Ch | 5A9Ch | 6A9Ch | 7A9Ch |
| A8h | 0AA0h | 1AA0h | 2AA0h | 3AA0h | 4AA0h | 5AA0h | 6AA0h | 7AA0h |

Продолжение таблицы Б.7

| Маска | A | B | C | D | E | F | G | H |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| A9h | 0AA4h | 1AA4h | 2AA4h | 3AA4h | 4AA4h | 5AA4h | 6AA4h | 7AA4h |
| AAh | 0AA8h | 1AA8h | 2AA8h | 3AA8h | 4AA8h | 5AA8h | 6AA8h | 7AA8h |
| ABh | 0AACh | 1AACh | 2AACh | 3AACh | 4AACh | 5AACh | 6AACh | 7AACh |
| ACh | 0AB0h | 1AB0h | 2AB0h | 3AB0h | 4AB0h | 5AB0h | 6AB0h | 7AB0h |
| ADh | 0AB4h | 1AB4h | 2AB4h | 3AB4h | 4AB4h | 5AB4h | 6AB4h | 7AB4h |
| A Eh | 0AB8h | 1AB8h | 2AB8h | 3AB8h | 4AB8h | 5AB8h | 6AB8h | 7AB8h |
| AFh | 0ABCh | 1ABCh | 2ABCh | 3ABCh | 4ABCh | 5ABCh | 6ABCh | 7ABCh |
| B0h | 0AC0h | 1AC0h | 2AC0h | 3AC0h | 4AC0h | 5AC0h | 6AC0h | 7AC0h |
| B1h | 0AC4h | 1AC4h | 2AC4h | 3AC4h | 4AC4h | 5AC4h | 6AC4h | 7AC4h |
| B2h | 0AC8h | 1AC8h | 2AC8h | 3AC8h | 4AC8h | 5AC8h | 6AC8h | 7AC8h |
| B3h | 0ACCh | 1ACCh | 2ACCh | 3ACCh | 4ACCh | 5ACCh | 6ACCh | 7ACCh |
| B4h | 0AD0h | 1AD0h | 2AD0h | 3AD0h | 4AD0h | 5AD0h | 6AD0h | 7AD0h |
| B5h | 0AD4h | 1AD4h | 2AD4h | 3AD4h | 4AD4h | 5AD4h | 6AD4h | 7AD4h |
| B6h | 0AD8h | 1AD8h | 2AD8h | 3AD8h | 4AD8h | 5AD8h | 6AD8h | 7AD8h |
| B7h | 0ADCh | 1ADCh | 2ADCh | 3ADCh | 4ADCh | 5ADCh | 6ADCh | 7ADCh |
| B8h | 0AE0h | 1AE0h | 2AE0h | 3AE0h | 4AE0h | 5AE0h | 6AE0h | 7AE0h |
| B9h | 0AE4h | 1AE4h | 2AE4h | 3AE4h | 4AE4h | 5AE4h | 6AE4h | 7AE4h |
| BAh | 0AE8h | 1AE8h | 2AE8h | 3AE8h | 4AE8h | 5AE8h | 6AE8h | 7AE8h |
| BBh | 0AECh | 1AECh | 2AECh | 3AECh | 4AECh | 5AECh | 6AECh | 7AECh |
| BCh | 0AF0h | 1AF0h | 2AF0h | 3AF0h | 4AF0h | 5AF0h | 6AF0h | 7AF0h |
| BDh | 0AF4h | 1AF4h | 2AF4h | 3AF4h | 4AF4h | 5AF4h | 6AF4h | 7AF4h |
| BEh | 0AF8h | 1AF8h | 2AF8h | 3AF8h | 4AF8h | 5AF8h | 6AF8h | 7AF8h |
| BFh | 0AFCh | 1AFCh | 2AFCh | 3AFCh | 4AFCh | 5AFCh | 6AFCh | 7AFCh |
| C0h | 0B00h | 1B00h | 2B00h | 3B00h | 4B00h | 5B00h | 6B00h | 7B00h |
| C1h | 0B04h | 1B04h | 2B04h | 3B04h | 4B04h | 5B04h | 6B04h | 7B04h |
| C2h | 0B08h | 1B08h | 2B08h | 3B08h | 4B08h | 5B08h | 6B08h | 7B08h |
| C3h | 0B0Ch | 1B0Ch | 2B0Ch | 3B0Ch | 4B0Ch | 5B0Ch | 6B0Ch | 7B0Ch |
| C4h | 0B10h | 1B10h | 2B10h | 3B10h | 4B10h | 5B10h | 6B10h | 7B10h |
| C5h | 0B14h | 1B14h | 2B14h | 3B14h | 4B14h | 5B14h | 6B14h | 7B14h |
| C6h | 0B18h | 1B18h | 2B18h | 3B18h | 4B18h | 5B18h | 6B18h | 7B18h |
| C7h | 0B1Ch | 1B1Ch | 2B1Ch | 3B1Ch | 4B1Ch | 5B1Ch | 6B1Ch | 7B1Ch |
| C8h | 0B20h | 1B20h | 2B20h | 3B20h | 4B20h | 5B20h | 6B20h | 7B20h |
| C9h | 0B24h | 1B24h | 2B24h | 3B24h | 4B24h | 5B24h | 6B24h | 7B24h |
| CAh | 0B28h | 1B28h | 2B28h | 3B28h | 4B28h | 5B28h | 6B28h | 7B28h |
| CBh | 0B2Ch | 1B2Ch | 2B2Ch | 3B2Ch | 4B2Ch | 5B2Ch | 6B2Ch | 7B2Ch |
| CCh | 0B20h | 1B20h | 2B20h | 3B20h | 4B20h | 5B20h | 6B20h | 7B20h |
| CDh | 0B34h | 1B34h | 2B34h | 3B34h | 4B34h | 5B34h | 6B34h | 7B34h |
| CEh | 0B38h | 1B38h | 2B38h | 3B38h | 4B38h | 5B38h | 6B38h | 7B38h |
| CFh | 0B3Ch | 1B3Ch | 2B3Ch | 3B3Ch | 4B3Ch | 5B3Ch | 6B3Ch | 7B3Ch |
| D0h | 0B40h | 1B40h | 2B40h | 3B40h | 4B40h | 5B40h | 6B40h | 7B40h |
| D1h | 0B44h | 1B44h | 2B44h | 3B44h | 4B44h | 5B44h | 6B44h | 7B44h |
| D2h | 0B48h | 1B48h | 2B48h | 3B48h | 4B48h | 5B48h | 6B48h | 7B48h |
| D3h | 0B4Ch | 1B4Ch | 2B4Ch | 3B4Ch | 4B4Ch | 5B4Ch | 6B4Ch | 7B4Ch |
| D4h | 0B50h | 1B50h | 2B50h | 3B50h | 4B50h | 5B50h | 6B50h | 7B50h |
| D5h | 0B54h | 1B54h | 2B54h | 3B54h | 4B54h | 5B54h | 6B54h | 7B54h |
| D6h | 0B58h | 1B58h | 2B58h | 3B58h | 4B58h | 5B58h | 6B58h | 7B58h |
| D7h | 0B5Ch | 1B5Ch | 2B5Ch | 3B5Ch | 4B5Ch | 5B5Ch | 6B5Ch | 7B5Ch |
| D8h | 0B60h | 1B60h | 2B60h | 3B60h | 4B60h | 5B60h | 6B60h | 7B60h |

Окончание таблицы Б.7

| Маска | A | B | C | D | E | F | G | H |
|----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| D9h | 0B64h | 1B64h | 2B64h | 3B64h | 4B64h | 5B64h | 6B64h | 7B64h |
| DAh | 0B68h | 1B68h | 2B68h | 3B68h | 4B68h | 5B68h | 6B68h | 7B68h |
| DBh | 0B6Ch | 1B6Ch | 2B6Ch | 3B6Ch | 4B6Ch | 5B6Ch | 6B6Ch | 7B6Ch |
| DCh | 0B70h | 1B70h | 2B70h | 3B70h | 4B70h | 5B70h | 6B70h | 7B70h |
| DDh | 0B74h | 1B74h | 2B74h | 3B74h | 4B74h | 5B74h | 6B74h | 7B74h |
| DEh | 0B78h | 1B78h | 2B78h | 3B78h | 4B78h | 5B78h | 6B78h | 7B78h |
| DFh | 0B7Ch | 1B7Ch | 2B7Ch | 3B7Ch | 4B7Ch | 5B7Ch | 6B7Ch | 7B7Ch |
| E0h | 0B80h | 1B80h | 2B80h | 3B80h | 4B80h | 5B80h | 6B80h | 7B80h |
| E1h | 0B84h | 1B84h | 2B84h | 3B84h | 4B84h | 5B84h | 6B84h | 7B84h |
| E2h | 0B88h | 1B88h | 2B88h | 3B88h | 4B88h | 5B88h | 6B88h | 7B88h |
| E3h | 0B8Ch | 1B8Ch | 2B8Ch | 3B8Ch | 4B8Ch | 5B8Ch | 6B8Ch | 7B8Ch |
| E4h | 0B90h | 1B90h | 2B90h | 3B90h | 4B90h | 5B90h | 6B90h | 7B90h |
| E5h | 0B94h | 1B94h | 2B94h | 3B94h | 4B94h | 5B94h | 6B94h | 7B94h |
| E6h | 0B98h | 1B98h | 2B98h | 3B98h | 4B98h | 5B98h | 6B98h | 7B98h |
| E7h | 0B9Ch | 1B9Ch | 2B9Ch | 3B9Ch | 4B9Ch | 5B9Ch | 6B9Ch | 7B9Ch |
| E8h | 0BA0h | 1BA0h | 2BA0h | 3BA0h | 4BA0h | 5BA0h | 6BA0h | 7BA0h |
| E9h | 0BA4h | 1BA4h | 2BA4h | 3BA4h | 4BA4h | 5BA4h | 6BA4h | 7BA4h |
| EAh | 0BA8h | 1BA8h | 2BA8h | 3BA8h | 4BA8h | 5BA8h | 6BA8h | 7BA8h |
| EBh | 0BACh | 1BACh | 2BACh | 3BACh | 4BACh | 5BACh | 6BACh | 7BACh |
| ECh | 0BB0h | 1BB0h | 2BB0h | 3BB0h | 4BB0h | 5BB0h | 6BB0h | 7BB0h |
| EDh | 0BB4h | 1BB4h | 2BB4h | 3BB4h | 4BB4h | 5BB4h | 6BB4h | 7BB4h |
| EEh | 0BB8h | 1BB8h | 2BB8h | 3BB8h | 4BB8h | 5BB8h | 6BB8h | 7BB8h |
| EFh | 0BBCh | 1BBCh | 2BBCh | 3BBCh | 4BBCh | 5BBCh | 6BBCh | 7BBCh |
| F0h | 0BC0h | 1BC0h | 2BC0h | 3BC0h | 4BC0h | 5BC0h | 6BC0h | 7BC0h |
| F1h | 0BC4h | 1BC4h | 2BC4h | 3BC4h | 4BC4h | 5BC4h | 6BC4h | 7BC4h |
| F2h | 0BC8h | 1BC8h | 2BC8h | 3BC8h | 4BC8h | 5BC8h | 6BC8h | 7BC8h |
| F3h | 0BCCh | 1BCCh | 2BCCh | 3BCCh | 4BCCh | 5BCCh | 6BCCh | 7BCCh |
| F4h | 0BD0h | 1BD0h | 2BD0h | 3BD0h | 4BD0h | 5BD0h | 6BD0h | 7BD0h |
| F5h | 0BD4h | 1BD4h | 2BD4h | 3BD4h | 4BD4h | 5BD4h | 6BD4h | 7BD4h |
| F6h | 0BD8h | 1BD8h | 2BD8h | 3BD8h | 4BD8h | 5BD8h | 6BD8h | 7BD8h |
| F7h | 0BDCh | 1BDCh | 2BDCh | 3BDCh | 4BDCh | 5BDCh | 6BDCh | 7BDCh |
| F8h | 0BE0h | 1BE0h | 2BE0h | 3BE0h | 4BE0h | 5BE0h | 6BE0h | 7BE0h |
| F9h | 0BE4h | 1BE4h | 2BE4h | 3BE4h | 4BE4h | 5BE4h | 6BE4h | 7BE4h |
| FAh | 0BE8h | 1BE8h | 2BE8h | 3BE8h | 4BE8h | 5BE8h | 6BE8h | 7BE8h |
| FBh | 0BECh | 1BECh | 2BECh | 3BECh | 4BECh | 5BECh | 6BECh | 7BECh |
| FCh | 0BF0h | 1BF0h | 2BF0h | 3BF0h | 4BF0h | 5BF0h | 6BF0h | 7BF0h |
| FDh | 0BF4h | 1BF4h | 2BF4h | 3BF4h | 4BF4h | 5BF4h | 6BF4h | 7BF4h |
| FEh | 0BF8h | 1BF8h | 2BF8h | 3BF8h | 4BF8h | 5BF8h | 6BF8h | 7BF8h |
| FFh | 0BFCh | 1BFCh | 2BFCh | 3BFCh | 4BFCh | 5BFCh | 6BFCh | 7BFCh |
| Зарезер- вировано | 0C00h- 0FFCh | 1C00h- 1FFCh | 2C00h- 2FFCh | 3C00h- 3FFCh | 4C00h- 4FFCh | 5C00h- 5FFCh | 6C00h- 6FFCh | 7C00h- 7FFCh |

Таблица Б.8 – Мнемоника и соответствующие названия регистров контроллера GPIO

| Мнемоника | Сброс | Название |
|-----------|---------|---|
| 4DATA | - - - h | Регистр данных порта |
| DATAOUT | 0000h | Выходной регистр данных порта |
| OUTENSET | 0000h | Регистр разрешения выходов порта |
| OUTENCLR | 0000h | Регистр сброса разрешения выходов порта |

Окончание таблицы Б.8

| Мнемоника | Сброс | Название |
|--|-----------|--|
| ALTFUNCSET | 0000h | Регистр включения альтернативной функции порта |
| ALTFUNCCLR | 0000h | Регистр выключения альтернативной функции порта |
| INTENSET | 0000h | Регистр разрешения прерываний порта |
| INTENCLR | 0000h | Регистр сброса разрешения прерываний порта |
| INTTYPESET | 0000h | Регистр типа прерываний порта |
| INTTYPECLR | 0000h | Регистр сброса типа прерываний порта |
| INTPOLSET | 0000h | Регистр полярности события прерывания порта |
| INTPOLCLR | 0000h | Регистр сброса полярности события прерывания порта |
| INTSTATUS, INTCLEAR | 0000h | Регистр состояния и сброса прерываний порта |
| MASKLOWBYTE | - - - - h | Регистр маски n младшего байта порта |
| MASKHIGHBYTE | - - - - h | Регистры масок n старших байт порта |
| Адреса 80018000h-8002FFFCh являются зарезервированными | | |

Таблица Б.9 – Регистры общего назначения

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|------------|-----------|--|
| 80030000h | GPIODEN0 | 00020062h | Регистр разрешения выходов портов А и В |
| 80030004h | GPIODEN1 | 08000000h | Регистр разрешения выходов портов С и D |
| 80030008h | GPIODEN2 | 00000400h | Регистр разрешения выходов портов Е и F |
| 8003000Ch | GPIODEN3 | 00000000h | Регистр разрешения выходов портов G и H |
| 80030010h- 80030014h | - | - | Зарезервировано |
| 80030018h | GPIOPCTLA | 00000000h | Регистр выбора функции порта А |
| 8003001Ch | GPIOPCTLB | 00000000h | Регистр выбора функции порта В |
| 80030020h | GPIOPCTLC | 00000000h | Регистр выбора функции порта С |
| 80030024h | GPIOPCTLD | 00000000h | Регистр выбора функции порта D |
| 80030028h | GPIOPCTLE | 00000000h | Регистр выбора функции порта Е |
| 8003002Ch | GPIOPCTLF | 00000000h | Регистр выбора функции порта F |
| 80030030h | GPIOPCTLG | 00000000h | Регистр выбора функции порта G |
| 80030034h | GPIOPCTLH | 00000000h | Регистр выбора функции порта H |
| 80030038h- 80030044h | - | - | Зарезервировано |
| 80030048h | GPIOODCTL0 | 00000000h | Регистр включения режима открытого коллектора портов А и В |
| 8003004Ch | GPIOODCTL1 | 00000000h | Регистр включения режима открытого коллектора портов С и D |
| 80030050h | GPIOODCTL2 | 00000000h | Регистр включения режима открытого коллектора портов Е и F |
| 80030054h | GPIOODCTL3 | 00000000h | Регистр включения режима открытого коллектора портов G и H |
| 80030058h- 8003005Ch | - | - | Зарезервировано |
| 80030060h | GPIODSCTL0 | 00000000h | Регистр управления нагрузочной способностью выводов портов А и В |
| 80030064h | GPIODSCTL1 | 00000000h | Регистр управления нагрузочной способностью выводов портов С и D |
| 80030068h | GPIODSCTL2 | 00000000h | Регистр управления нагрузочной способностью выводов портов Е и F |

Окончание таблицы Б.9

| Адрес | Мнемоника | Сброс | Название |
|--------------------------|-------------|-----------|--|
| 8003006Ch | GPIODSCTL3 | 00000000h | Регистр управления нагрузочной способностью выводов портов G и H |
| 80030070h- 80030074h | – | – | Зарезервировано |
| 80030078h | GPIOPUCTL0 | 00000000h | Регистр управления схемой «Pull-up» портов A и B |
| 8003007Ch | GPIOPUCTL1 | 00000000h | Регистр управления схемой «Pull-up» портов C и D |
| 80030080h | GPIOPUCTL2 | 00000000h | Регистр управления схемой «Pull-up» портов E и F |
| 80030084h | GPIOPUCTL3 | 00000000h | Регистр управления схемой «Pull-up» портов G и H |
| 80030088h- 80030090h | – | – | Зарезервировано |
| 80030094h | PLL_CTRL | 00000000h | Регистр управления настройками PLL |
| 80030098h | PLL_OD | 00000000h | Регистр задания выходного делителя PLL |
| 8003009Ch | PLL_NR | 00000000h | Регистр задания опорного делителя PLL |
| 800300A0h | PLL_NF | 00000000h | Регистр задания делителя обратной связи PLL |
| 800300A4h | EXT_MEM_CFG | 80000007h | Регистр настройки контроллера внешней памяти |
| 800300A8h | ADC_CTRL0 | 00000000h | Регистр 0 настройки блоков АЦП |
| 800300ACh | ADC_CTRL1 | 00000000h | Регистр 1 настройки блоков АЦП |
| 800300B0h | PWM_SYNC | 00000000h | Регистр синхронизации блоков ШИМ |
| 800300B4h | PWM_CTRL | 00000000h | Регистр настройки синхронизации блоков ШИМ |
| 800300B8h | SYS_CLK | 00000000h | Регистр настройки системной частоты микроконтроллера |
| 800300BCh | APB_CLK | 00000000h | Регистр настройки разрешения тактовых сигналов периферии |
| 800300C0h | UART_CLK | 00000000h | Регистр настройки тактовых сигналов блоков UART |
| 800300C4h | SPI_CLK | 00000000h | Регистр настройки тактовых сигналов блоков SPI |
| 800300C8h | PER_RST0 | 00000000h | Регистр 0 сброса блоков периферии |
| 800300CCh | PER_RST1 | 00000000h | Регистр 1 сброса блоков периферии |
| 800300D0h | GPIOSE0 | 00000000h | Регистр управления пересинхронизацией входов GPIO портов A и B |
| 800300D4h | GPIOSE1 | 00000000h | Регистр управления пересинхронизацией входов GPIO портов C и D |
| 800300D8h | GPIOSE2 | 00000000h | Регистр управления пересинхронизацией входов GPIO портов E и F |
| 800300DCh | GPIOSE3 | 00000000h | Регистр управления пересинхронизацией входов GPIO портов G и H |
| 800300E0h - 800300ECh | – | – | Зарезервировано |
| 800300F0h | GPIOQE0 | 00000000h | Регистр включения фильтра GPIO для портов A и B |

Окончание таблицы Б.9

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|--------------------------|-----------|---|
| 800300F4h | GPIOQE1 | 00000000h | Регистр включения фильтра GPIO для портов С и D |
| 800300F8h | GPIOQE2 | 00000000h | Регистр включения фильтра GPIO для портов Е и F |
| 800300FCh | GPIOQE3 | 00000000h | Регистр включения фильтра GPIO для портов G и H |
| 80030100h | GPIOQM0 | 00000000h | Регистр режима работы фильтра GPIO для портов А и В |
| 80030104h | GPIOQM1 | 00000000h | Регистр режима работы фильтра GPIO для портов С и D |
| 80030108h | GPIOQM2 | 00000000h | Регистр режима работы фильтра GPIO для портов Е и F |
| 8003010Ch | GPIOQM3 | 00000000h | Регистр режима работы фильтра GPIO для портов G и H |
| 80030110h | GPIOQPAD | 00000000h | Регистр настройки фильтров портов А-D |
| 80030114h | GPIOQPEH | 00000000h | Регистр настройки фильтров портов Е-H |
| 80030118h- 8003012Ch | – | – | Зарезервировано |
| 80030130h | USB_CTRL | 00000000h | Регистр настройки USB PHY |
| 80030134h | – | – | Зарезервировано |
| 80030138h | UART_SPI_ CLK_SEL | 00000000h | Регистр выбора источников тактовых сигналов блоков UART и SPI |
| 8003013Ch | ADC_CTRL2 | 00000000h | Регистр 2 настройки блоков АЦП |
| 80030140h- 80030160h | – | – | Зарезервировано |
| 80030164h | FLASH_FULL – ERASE | 00000000h | Регистр полного стирания флеш-памяти |
| 80030168h- 8006FFFCh | – | – | Зарезервировано |

Таблица Б.10 – Регистры контроллера CAN

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|-----------|-----------|-----------------------------|
| 80070000h | CLC | 00000003h | Регистр управления частотой |
| 80070004h | – | – | Зарезервировано |
| 80070008h | ID | 002BC051h | Регистр идентификации |
| 8007000Ch | FDR | 00000000h | Регистр делителя |
| 80070010h- 800700FCh | – | – | Зарезервировано |
| 80070100h | LIST0 | 007F7F00h | Регистр списка 0 |
| 80070104h | LIST1 | 01000000h | Регистр свободного списка 1 |
| 80070108h | LIST2 | 01000000h | Регистр свободного списка 2 |
| 8007010Ch | LIST3 | 01000000h | Регистр свободного списка 3 |
| 80070110h | LIST4 | 01000000h | Регистр свободного списка 4 |
| 80070114h | LIST5 | 01000000h | Регистр свободного списка 5 |
| 80070118h | LIST6 | 01000000h | Регистр свободного списка 6 |
| 8007011Ch | LIST7 | 01000000h | Регистр свободного списка 7 |

Окончание таблицы Б.10

| Адрес | Мнемоника | Сброс | Название |
|---------------------|-----------------|-----------|---------------------------------|
| 80070120h-8007013Ch | – | – | Зарезервировано |
| 80070140h | MSPND0 | 00000000h | Регистр 0 ждущих прерываний |
| 80070144h | MSPND1 | 00000000h | Регистр 1 ждущих прерываний |
| 80070148h | MSPND2 | 00000000h | Регистр 2 ждущих прерываний |
| 8007014Ch | MSPND3 | 00000000h | Регистр 3 ждущих прерываний |
| 80070150h | MSPND4 | 00000000h | Регистр 4 ждущих прерываний |
| 80070154h | MSPND5 | 00000000h | Регистр 5 ждущих прерываний |
| 80070158h | MSPND6 | 00000000h | Регистр 6 ждущих прерываний |
| 8007015Ch | MSPND7 | 00000000h | Регистр 7 ждущих прерываний |
| 80070160h-8007017Ch | – | – | Зарезервировано |
| 80070180h | MSID0 | 00000020h | Регистр 0 индекса сообщения |
| 80070184h | MSID1 | 00000020h | Регистр 1 индекса сообщения |
| 80070188h | MSID2 | 00000020h | Регистр 2 индекса сообщения |
| 8007018Ch | MSID3 | 00000020h | Регистр 3 индекса сообщения |
| 80070190h | MSID4 | 00000020h | Регистр 4 индекса сообщения |
| 80070194h | MSID5 | 00000020h | Регистр 5 индекса сообщения |
| 80070198h | MSID6 | 00000020h | Регистр 6 индекса сообщения |
| 8007019Ch | MSID7 | 00000020h | Регистр 7 индекса сообщения |
| 800701A0h-800701BCh | – | – | Зарезервировано |
| 800701C0h | MSIMASK | 00000000h | Регистр маски индекса сообщения |
| 800701C4h | PANCTR | 00000301h | Регистр панели команд |
| 800701C8h | MCR | 00000000h | Регистр управления |
| 800701CCh | MITR | 00000000h | Регистр прерываний |
| 800701D0h-800701FCh | Зарезервировано | | |

Таблица Б.11 – Адреса регистров узлов контроллера CAN

| Мнемоника | Адреса регистров узлов | | Сброс | Название |
|--|------------------------|-----------|-----------|-----------------------------------|
| | Узел 0 | Узел 1 | | |
| NCR | 80070200h | 80070300h | 00000001h | Регистр управления узла |
| NSR | 80070204h | 80070304h | 00000000h | Регистр состояния узла |
| NIPR | 80070208h | 80070308h | 00000000h | Регистр указателя прерываний узла |
| NPCR | 8007020Ch | 8007030Ch | 00000000h | Регистр управления портом узла |
| NBTR | 80070210h | 80070310h | 00000000h | Регистр синхронизации битов узла |
| NECNT | 80070214h | 80070314h | 00600000h | Регистр счетчика ошибок узла |
| NFCR | 80070218h | 80070318h | 00000000h | Регистр счетчика сообщений узла |
| Адреса 8007021Ch-80070FFCh являются зарезервированными | | | | |

Таблица Б.12 – Адреса регистров объектов сообщений контроллера CAN

| Объект сообщения | Регистры объектов сообщений | | | | | | | |
|---------------------|-----------------------------|------------|-----------|-----------|-------------|-------------|----------|------------------|
| | Адрес 8007_xxxx | | | | | | | |
| | МО FCR | МО FGPR | МО IPR | МО AMR | МО DATAL | МО DATAH | МО AR | МОCTR/ MOSTAT |
| 0 | 1000h | 1004h | 1008h | 100Ch | 1010h | 1014h | 1018h | 101Ch |
| 1 | 1020h | 1024h | 1028h | 102Ch | 1030h | 1034h | 1038h | 103Ch |
| 2 | 1040h | 1044h | 1048h | 104Ch | 1050h | 1054h | 1058h | 105Ch |
| 3 | 1060h | 1064h | 1068h | 106Ch | 1070h | 1074h | 1078h | 107Ch |
| 4 | 1080h | 1084h | 1088h | 108Ch | 1090h | 1094h | 1098h | 109Ch |
| 5 | 10A0h | 10A4h | 10A8h | 10ACh | 10B0h | 10B4h | 10B8h | 10BCh |
| 6 | 10C0h | 10C4h | 10C8h | 10CCh | 10D0h | 10D4h | 10D8h | 10DCh |
| 7 | 10E0h | 10E4h | 10E8h | 10ECh | 10F0h | 10F4h | 10F8h | 10FCh |
| 8 | 1100h | 1104h | 1108h | 110Ch | 1110h | 1114h | 1118h | 111Ch |
| 9 | 1120h | 1124h | 1128h | 112Ch | 1130h | 1134h | 1138h | 113Ch |
| 10 | 1140h | 1144h | 1148h | 114Ch | 1150h | 1154h | 1158h | 115Ch |
| 11 | 1160h | 1164h | 1168h | 116Ch | 1170h | 1174h | 1178h | 117Ch |
| 12 | 1180h | 1184h | 1188h | 118Ch | 1190h | 1194h | 1198h | 119Ch |
| 13 | 11A0h | 11A4h | 11A8h | 11ACh | 11B0h | 11B4h | 11B8h | 11BCh |
| 14 | 11C0h | 11C4h | 11C8h | 11CCh | 11D0h | 11D4h | 11D8h | 11DCh |
| 15 | 11E0h | 11E4h | 11E8h | 11ECh | 11F0h | 11F4h | 11F8h | 11FCh |
| 16 | 1200h | 1204h | 1208h | 120Ch | 1210h | 1214h | 1218h | 121Ch |
| 17 | 1220h | 1224h | 1228h | 122Ch | 1230h | 1234h | 1238h | 123Ch |
| 18 | 1240h | 1244h | 1248h | 124Ch | 1250h | 1254h | 1258h | 125Ch |
| 19 | 1260h | 1264h | 1268h | 126Ch | 1270h | 1274h | 1278h | 127Ch |
| 20 | 1280h | 1284h | 1288h | 128Ch | 1290h | 1294h | 1298h | 129Ch |
| 21 | 12A0h | 12A4h | 12A8h | 12ACh | 12B0h | 12B4h | 12B8h | 12BCh |
| 22 | 12C0h | 12C4h | 12C8h | 12CCh | 12D0h | 12D4h | 12D8h | 12DCh |
| 23 | 12E0h | 12E4h | 12E8h | 12ECh | 12F0h | 12F4h | 12F8h | 12FCh |
| 24 | 1300h | 1304h | 1308h | 130Ch | 1310h | 1314h | 1318h | 131Ch |
| 25 | 1320h | 1324h | 1328h | 132Ch | 1330h | 1334h | 1338h | 133Ch |
| 26 | 1340h | 1344h | 1348h | 134Ch | 1350h | 1354h | 1358h | 135Ch |
| 27 | 1360h | 1364h | 1368h | 136Ch | 1370h | 1374h | 1378h | 137Ch |
| 28 | 1380h | 1384h | 1388h | 138Ch | 1390h | 1394h | 1398h | 139Ch |
| 29 | 13A0h | 13A4h | 13A8h | 13ACh | 13B0h | 13B4h | 13B8h | 13BCh |
| 30 | 13C0h | 13C4h | 13C8h | 13CCh | 13D0h | 13D4h | 13D8h | 13DCh |
| 31 | 13E0h | 13E4h | 13E8h | 13ECh | 13F0h | 13F4h | 13F8h | 13FCh |
| 32 | 1400h | 1404h | 1408h | 140Ch | 1410h | 1414h | 1418h | 141Ch |
| 33 | 1420h | 1424h | 1428h | 142Ch | 1430h | 1434h | 1438h | 143Ch |
| 34 | 1440h | 1444h | 1448h | 144Ch | 1450h | 1454h | 1458h | 145Ch |
| 35 | 1460h | 1464h | 1468h | 146Ch | 1470h | 1474h | 1478h | 147Ch |
| 36 | 1480h | 1484h | 1488h | 148Ch | 1490h | 1494h | 1498h | 149Ch |
| 37 | 14A0h | 14A4h | 14A8h | 14ACh | 14B0h | 14B4h | 14B8h | 14BCh |
| 38 | 14C0h | 14C4h | 14C8h | 14CCh | 14D0h | 14D4h | 14D8h | 14DCh |
| 39 | 14E0h | 14E4h | 14E8h | 14ECh | 14F0h | 14F4h | 14F8h | 14FCh |
| 40 | 1500h | 1504h | 1508h | 150Ch | 1510h | 1514h | 1518h | 151Ch |
| 41 | 1520h | 1524h | 1528h | 152Ch | 1530h | 1534h | 1538h | 153Ch |
| 42 | 1540h | 1544h | 1548h | 154Ch | 1550h | 1554h | 1558h | 155Ch |

Продолжение таблицы Б.12

| Объект сообщения | МО FCR | МО FGPR | МО IPR | МО AMR | МО DATAL | МО DATAH | МО AR | МОСТР/ MOSTAT |
|---------------------|-----------|------------|-----------|-----------|-------------|-------------|----------|------------------|
| 43 | 1560h | 1564h | 1568h | 156Ch | 1570h | 1574h | 1578h | 157Ch |
| 44 | 1580h | 1584h | 1588h | 158Ch | 1590h | 1594h | 1598h | 159Ch |
| 45 | 15A0h | 15A4h | 15A8h | 15ACh | 15B0h | 15B4h | 15B8h | 15BCh |
| 46 | 15C0h | 15C4h | 15C8h | 15CCh | 15D0h | 15D4h | 15D8h | 15DCh |
| 47 | 15E0h | 15E4h | 15E8h | 15ECh | 15F0h | 15F4h | 15F8h | 15FCh |
| 48 | 1600h | 1604h | 1608h | 160Ch | 1610h | 1614h | 1618h | 161Ch |
| 49 | 1620h | 1624h | 1628h | 162Ch | 1630h | 1634h | 1638h | 163Ch |
| 50 | 1640h | 1644h | 1648h | 164Ch | 1650h | 1654h | 1658h | 165Ch |
| 51 | 1660h | 1664h | 1668h | 166Ch | 1670h | 1674h | 1678h | 167Ch |
| 52 | 1680h | 1684h | 1688h | 168Ch | 1690h | 1694h | 1698h | 169Ch |
| 53 | 16A0h | 16A4h | 16A8h | 16ACh | 16B0h | 16B4h | 16B8h | 16BCh |
| 54 | 16C0h | 16C4h | 16C8h | 16CCh | 16D0h | 16D4h | 16D8h | 16DCh |
| 55 | 16E0h | 16E4h | 16E8h | 16ECh | 16F0h | 16F4h | 16F8h | 16FCh |
| 56 | 1700h | 1704h | 1708h | 170Ch | 1710h | 1714h | 1718h | 171Ch |
| 57 | 1720h | 1724h | 1728h | 172Ch | 1730h | 1734h | 1738h | 173Ch |
| 58 | 1740h | 1744h | 1748h | 174Ch | 1750h | 1754h | 1758h | 175Ch |
| 59 | 1760h | 1764h | 1768h | 176Ch | 1770h | 1774h | 1778h | 177Ch |
| 60 | 1780h | 1784h | 1788h | 178Ch | 1790h | 1794h | 1798h | 179Ch |
| 61 | 17A0h | 17A4h | 17A8h | 17ACh | 17B0h | 17B4h | 17B8h | 17BCh |
| 62 | 17C0h | 17C4h | 17C8h | 17CCh | 17D0h | 17D4h | 17D8h | 17DCh |
| 63 | 17E0h | 17E4h | 17E8h | 17ECh | 17F0h | 17F4h | 17F8h | 17FCh |
| 64 | 1800h | 1804h | 1808h | 180Ch | 1810h | 1814h | 1818h | 181Ch |
| 65 | 1820h | 1824h | 1828h | 182Ch | 1830h | 1834h | 1838h | 183Ch |
| 66 | 1840h | 1844h | 1848h | 184Ch | 1850h | 1854h | 1858h | 185Ch |
| 67 | 1860h | 1864h | 1868h | 186Ch | 1870h | 1874h | 1878h | 187Ch |
| 68 | 1880h | 1884h | 1888h | 188Ch | 1890h | 1894h | 1898h | 189Ch |
| 69 | 18A0h | 18A4h | 18A8h | 18ACh | 18B0h | 18B4h | 18B8h | 18BCh |
| 70 | 18C0h | 18C4h | 18C8h | 18CCh | 18D0h | 18D4h | 18D8h | 18DCh |
| 71 | 18E0h | 18E4h | 18E8h | 18ECh | 18F0h | 18F4h | 18F8h | 18FCh |
| 72 | 1900h | 1904h | 1908h | 190Ch | 1910h | 1914h | 1918h | 191Ch |
| 73 | 1920h | 1924h | 1928h | 192Ch | 1930h | 1934h | 1938h | 193Ch |
| 74 | 1940h | 1944h | 1948h | 194Ch | 1950h | 1954h | 1958h | 195Ch |
| 75 | 1960h | 1964h | 1968h | 196Ch | 1970h | 1974h | 1978h | 197Ch |
| 76 | 1980h | 1984h | 1988h | 198Ch | 1990h | 1994h | 1998h | 199Ch |
| 77 | 19A0h | 19A4h | 19A8h | 19ACh | 19B0h | 19B4h | 19B8h | 19BCh |
| 78 | 19C0h | 19C4h | 19C8h | 19CCh | 19D0h | 19D4h | 19D8h | 19DCh |
| 79 | 19E0h | 19E4h | 19E8h | 19ECh | 19F0h | 19F4h | 19F8h | 19FCh |
| 80 | 1A00h | 1A04h | 1A08h | 1A0Ch | 1A10h | 1A14h | 1A18h | 1A1Ch |
| 81 | 1A20h | 1A24h | 1A28h | 1A2Ch | 1A30h | 1A34h | 1A38h | 1A3Ch |
| 82 | 1A40h | 1A44h | 1A48h | 1A4Ch | 1A50h | 1A54h | 1A58h | 1A5Ch |
| 83 | 1A60h | 1A64h | 1A68h | 1A6Ch | 1A70h | 1A74h | 1A78h | 1A7Ch |
| 84 | 1A80h | 1A84h | 1A88h | 1A8Ch | 1A90h | 1A94h | 1A98h | 1A9Ch |
| 85 | 1AA0h | 1AA4h | 1AA8h | 1AACh | 1AB0h | 1AB4h | 1AB8h | 1ABCh |
| 86 | 1AC0h | 1AC4h | 1AC8h | 1ACCh | 1AD0h | 1AD4h | 1AD8h | 1ADCh |
| 87 | 1AE0h | 1AE4h | 1AE8h | 1AECh | 1AF0h | 1AF4h | 1AF8h | 1AFCh |
| 88 | 1B00h | 1B04h | 1B08h | 1B0Ch | 1B10h | 1B14h | 1B18h | 1B1Ch |

Продолжение таблицы Б.12

| Объект сообщения | МО FCR | МО FGPR | МО IPR | МО AMR | МО DATAL | МО DATAH | МО AR | МОСТР/ MOSTAT |
|---------------------|-----------|------------|-----------|-----------|-------------|-------------|----------|------------------|
| 89 | 1B20h | 1B24h | 1B28h | 1B2Ch | 1B30h | 1B34h | 1B38h | 1B3Ch |
| 90 | 1B40h | 1B44h | 1B48h | 1B4Ch | 1B50h | 1B54h | 1B58h | 1B5Ch |
| 91 | 1B60h | 1B64h | 1B68h | 1B6Ch | 1B70h | 1B74h | 1B78h | 1B7Ch |
| 92 | 1B80h | 1B84h | 1B88h | 1B8Ch | 1B90h | 1B94h | 1B98h | 1B9Ch |
| 93 | 1BA0h | 1BA4h | 1BA8h | 1BACH | 1BB0h | 1BB4h | 1BB8h | 1BBCh |
| 94 | 1BC0h | 1BC4h | 1BC8h | 1BCCh | 1BD0h | 1BD4h | 1BD8h | 1BDCh |
| 95 | 1BE0h | 1BE4h | 1BE8h | 1BECh | 1BF0h | 1BF4h | 1BF8h | 1BFCh |
| 96 | 1C00h | 1C04h | 1C08h | 1C0Ch | 1C10h | 1C14h | 1C18h | 1C1Ch |
| 97 | 1C20h | 1C24h | 1C28h | 1C2Ch | 1C30h | 1C34h | 1C38h | 1C3Ch |
| 98 | 1C40h | 1C44h | 1C48h | 1C4Ch | 1C50h | 1C54h | 1C58h | 1C5Ch |
| 99 | 1C60h | 1C64h | 1C68h | 1C6Ch | 1C70h | 1C74h | 1C78h | 1C7Ch |
| 100 | 1C80h | 1C84h | 1C88h | 1C8Ch | 1C90h | 1C94h | 1C98h | 1C9Ch |
| 101 | 1CA0h | 1CA4h | 1CA8h | 1CACh | 1CB0h | 1CB4h | 1CB8h | 1CBCh |
| 102 | 1CC0h | 1CC4h | 1CC8h | 1CCCh | 1CD0h | 1CD4h | 1CD8h | 1CDCh |
| 103 | 1CE0h | 1CE4h | 1CE8h | 1CECh | 1CF0h | 1CF4h | 1CF8h | 1CFCh |
| 104 | 1D00h | 1D04h | 1D08h | 1D0Ch | 1D10h | 1D14h | 1D18h | 1D1Ch |
| 105 | 1D20h | 1D24h | 1D28h | 1D2Ch | 1D30h | 1D34h | 1D38h | 1D3Ch |
| 106 | 1D40h | 1D44h | 1D48h | 1D4Ch | 1D50h | 1D54h | 1D58h | 1D5Ch |
| 107 | 1D60h | 1D64h | 1D68h | 1D6Ch | 1D70h | 1D74h | 1D78h | 1D7Ch |
| 108 | 1D80h | 1D84h | 1D88h | 1D8Ch | 1D90h | 1D94h | 1D98h | 1D9Ch |
| 109 | 1DA0h | 1DA4h | 1DA8h | 1DACh | 1DB0h | 1DB4h | 1DB8h | 1DBCCh |
| 110 | 1DC0h | 1DC4h | 1DC8h | 1DCCh | 1DD0h | 1DD4h | 1DD8h | 1DDCh |
| 111 | 1DE0h | 1DE4h | 1DE8h | 1DECh | 1DF0h | 1DF4h | 1DF8h | 1DFCh |
| 112 | 1E00h | 1E04h | 1E08h | 1E0Ch | 1E10h | 1E14h | 1E18h | 1E1Ch |
| 113 | 1E20h | 1E24h | 1E28h | 1E2Ch | 1E30h | 1E34h | 1E38h | 1E3Ch |
| 114 | 1E40h | 1E44h | 1E48h | 1E4Ch | 1E50h | 1E54h | 1E58h | 1E5Ch |
| 115 | 1E60h | 1E64h | 1E68h | 1E6Ch | 1E70h | 1E74h | 1E78h | 1E7Ch |
| 116 | 1E80h | 1E84h | 1E88h | 1E8Ch | 1E90h | 1E94h | 1E98h | 1E9Ch |
| 117 | 1EA0h | 1EA4h | 1EA8h | 1EACH | 1EB0h | 1EB4h | 1EB8h | 1EBCCh |
| 118 | 1EC0h | 1EC4h | 1EC8h | 1ECCh | 1ED0h | 1ED4h | 1ED8h | 1EDCh |
| 119 | 1EE0h | 1EE4h | 1EE8h | 1EECh | 1EF0h | 1EF4h | 1EF8h | 1EFCh |
| 120 | 1F00h | 1F04h | 1F08h | 1F0Ch | 1F10h | 1F14h | 1F18h | 1F1Ch |
| 121 | 1F20h | 1F24h | 1F28h | 1F2Ch | 1F30h | 1F34h | 1F38h | 1F3Ch |
| 122 | 1F40h | 1F44h | 1F48h | 1F4Ch | 1F50h | 1F54h | 1F58h | 1F5Ch |
| 123 | 1F60h | 1F64h | 1F68h | 1F6Ch | 1F70h | 1F74h | 1F78h | 1F7Ch |
| 124 | 1F80h | 1F84h | 1F88h | 1F8Ch | 1F90h | 1F94h | 1F98h | 1F9Ch |
| 125 | 1FA0h | 1FA4h | 1FA8h | 1FACH | 1FB0h | 1FB4h | 1FB8h | 1FBCCh |
| 126 | 1FC0h | 1FC4h | 1FC8h | 1FCCh | 1FD0h | 1FD4h | 1FD8h | 1FDCh |
| 127 | 1FE0h | 1FE4h | 1FE8h | 1FECh | 1FF0h | 1FF4h | 1FF8h | 1FFCh |
| 128 | 2000h | 2004h | 2008h | 200Ch | 2010h | 2014h | 2018h | 201Ch |
| 129 | 2020h | 2024h | 2028h | 202Ch | 2030h | 2034h | 2038h | 203Ch |
| 130 | 2040h | 2044h | 2048h | 204Ch | 2050h | 2054h | 2058h | 205Ch |
| 131 | 2060h | 2064h | 2068h | 206Ch | 2070h | 2074h | 2078h | 207Ch |
| 132 | 2080h | 2084h | 2088h | 208Ch | 2090h | 2094h | 2098h | 209Ch |
| 133 | 20A0h | 20A4h | 20A8h | 20ACH | 20B0h | 20B4h | 20B8h | 20BCCh |
| 134 | 20C0h | 20C4h | 20C8h | 20CCh | 20D0h | 20D4h | 20D8h | 20DCh |

Продолжение таблицы Б.12

| Объект сообщения | МО FCR | МО FGPR | МО IPR | МО AMR | МО DATAL | МО DATAH | МО AR | МОСТР/ MOSTAT |
|---------------------|-----------|------------|-----------|-----------|-------------|-------------|----------|------------------|
| 135 | 20E0h | 20E4h | 20E8h | 20ECh | 20F0h | 20F4h | 20F8h | 20FCh |
| 136 | 2100h | 2104h | 2108h | 210Ch | 2110h | 2114h | 2118h | 211Ch |
| 137 | 2120h | 2124h | 2128h | 212Ch | 2130h | 2134h | 2138h | 213Ch |
| 138 | 2140h | 2144h | 2148h | 214Ch | 2150h | 2154h | 2158h | 215Ch |
| 139 | 2160h | 2164h | 2168h | 216Ch | 2170h | 2174h | 2178h | 217Ch |
| 140 | 2180h | 2184h | 2188h | 218Ch | 2190h | 2194h | 2198h | 219Ch |
| 141 | 21A0h | 21A4h | 21A8h | 21ACh | 21B0h | 21B4h | 21B8h | 21BCh |
| 142 | 21C0h | 21C4h | 21C8h | 21CCh | 21D0h | 21D4h | 21D8h | 21DCh |
| 143 | 21E0h | 21E4h | 21E8h | 21ECh | 21F0h | 21F4h | 21F8h | 21FCh |
| 144 | 2200h | 2204h | 2208h | 220Ch | 2210h | 2214h | 2218h | 221Ch |
| 145 | 2220h | 2224h | 2228h | 222Ch | 2230h | 2234h | 2238h | 223Ch |
| 146 | 2240h | 2244h | 2248h | 224Ch | 2250h | 2254h | 2258h | 225Ch |
| 147 | 2260h | 2264h | 2268h | 226Ch | 2270h | 2274h | 2278h | 227Ch |
| 148 | 2280h | 2284h | 2288h | 228Ch | 2290h | 2294h | 2298h | 229Ch |
| 149 | 22A0h | 22A4h | 22A8h | 22ACh | 22B0h | 22B4h | 22B8h | 22BCh |
| 150 | 22C0h | 22C4h | 22C8h | 22CCh | 22D0h | 22D4h | 22D8h | 22DCh |
| 151 | 22E0h | 22E4h | 22E8h | 22ECh | 22F0h | 22F4h | 22F8h | 22FCh |
| 152 | 2300h | 2304h | 2308h | 230Ch | 2310h | 2314h | 2318h | 231Ch |
| 153 | 2320h | 2324h | 2328h | 232Ch | 2330h | 2334h | 2338h | 233Ch |
| 154 | 2340h | 2344h | 2348h | 234Ch | 2350h | 2354h | 2358h | 235Ch |
| 155 | 2360h | 2364h | 2368h | 236Ch | 2370h | 2374h | 2378h | 237Ch |
| 156 | 2380h | 2384h | 2388h | 238Ch | 2390h | 2394h | 2398h | 239Ch |
| 157 | 23A0h | 23A4h | 23A8h | 23ACh | 23B0h | 23B4h | 23B8h | 23BCh |
| 158 | 23C0h | 23C4h | 23C8h | 23CCh | 23D0h | 23D4h | 23D8h | 23DCh |
| 159 | 23E0h | 23E4h | 23E8h | 23ECh | 23F0h | 23F4h | 23F8h | 23FCh |
| 160 | 2400h | 2404h | 2408h | 240Ch | 2410h | 2414h | 2418h | 241Ch |
| 161 | 2420h | 2424h | 2428h | 242Ch | 2430h | 2434h | 2438h | 243Ch |
| 162 | 2440h | 2444h | 2448h | 244Ch | 2450h | 2454h | 2458h | 245Ch |
| 163 | 2460h | 2464h | 2468h | 246Ch | 2470h | 2474h | 2478h | 247Ch |
| 164 | 2480h | 2484h | 2488h | 248Ch | 2490h | 2494h | 2498h | 249Ch |
| 165 | 24A0h | 24A4h | 24A8h | 24ACh | 24B0h | 24B4h | 24B8h | 24BCh |
| 166 | 24C0h | 24C4h | 24C8h | 24CCh | 24D0h | 24D4h | 24D8h | 24DCh |
| 167 | 24E0h | 24E4h | 24E8h | 24ECh | 24F0h | 24F4h | 24F8h | 24FCh |
| 168 | 2500h | 2504h | 2508h | 250Ch | 2510h | 2514h | 2518h | 251Ch |
| 169 | 2520h | 2524h | 2528h | 252Ch | 2530h | 2534h | 2538h | 253Ch |
| 170 | 2540h | 2544h | 2548h | 254Ch | 2550h | 2554h | 2558h | 255Ch |
| 171 | 2560h | 2564h | 2568h | 256Ch | 2570h | 2574h | 2578h | 257Ch |
| 172 | 2580h | 2584h | 2588h | 258Ch | 2590h | 2594h | 2598h | 259Ch |
| 173 | 25A0h | 25A4h | 25A8h | 25ACh | 25B0h | 25B4h | 25B8h | 25BCh |
| 174 | 25C0h | 25C4h | 25C8h | 25CCh | 25D0h | 25D4h | 25D8h | 25DCh |
| 175 | 25E0h | 25E4h | 25E8h | 25ECh | 25F0h | 25F4h | 25F8h | 25FCh |
| 176 | 2600h | 2604h | 2608h | 260Ch | 2610h | 2614h | 2618h | 261Ch |
| 177 | 2620h | 2624h | 2628h | 262Ch | 2630h | 2634h | 2638h | 263Ch |
| 178 | 2640h | 2644h | 2648h | 264Ch | 2650h | 2654h | 2658h | 265Ch |
| 179 | 2660h | 2664h | 2668h | 266Ch | 2670h | 2674h | 2678h | 267Ch |
| 180 | 2680h | 2684h | 2688h | 268Ch | 2690h | 2694h | 2698h | 269Ch |

Продолжение таблицы Б.12

| Объект сообщения | МО FCR | МО FGPR | МО IPR | МО AMR | МО DATAL | МО DATAH | МО AR | МОСТР/ MOSTAT |
|---------------------|-----------|------------|-----------|-----------|-------------|-------------|----------|------------------|
| 181 | 26A0h | 26A4h | 26A8h | 26ACh | 26B0h | 26B4h | 26B8h | 26BCh |
| 182 | 26C0h | 26C4h | 26C8h | 26CCh | 26D0h | 26D4h | 26D8h | 26DCh |
| 183 | 26E0h | 26E4h | 26E8h | 26ECh | 26F0h | 26F4h | 26F8h | 26FCh |
| 184 | 2700h | 2704h | 2708h | 270Ch | 2710h | 2714h | 2718h | 271Ch |
| 185 | 2720h | 2724h | 2728h | 272Ch | 2730h | 2734h | 2738h | 273Ch |
| 186 | 2740h | 2744h | 2748h | 274Ch | 2750h | 2754h | 2758h | 275Ch |
| 187 | 2760h | 2764h | 2768h | 276Ch | 2770h | 2774h | 2778h | 277Ch |
| 188 | 2780h | 2784h | 2788h | 278Ch | 2790h | 2794h | 2798h | 279Ch |
| 189 | 27A0h | 27A4h | 27A8h | 27ACh | 27B0h | 27B4h | 27B8h | 27BCh |
| 190 | 27C0h | 27C4h | 27C8h | 27CCh | 27D0h | 27D4h | 27D8h | 27DCh |
| 191 | 27E0h | 27E4h | 27E8h | 27ECh | 27F0h | 27F4h | 27F8h | 27FCh |
| 192 | 2800h | 2804h | 2808h | 280Ch | 2810h | 2814h | 2818h | 281Ch |
| 193 | 2820h | 2824h | 2828h | 282Ch | 2830h | 2834h | 2838h | 283Ch |
| 194 | 2840h | 2844h | 2848h | 284Ch | 2850h | 2854h | 2858h | 285Ch |
| 195 | 2860h | 2864h | 2868h | 286Ch | 2870h | 2874h | 2878h | 287Ch |
| 196 | 2880h | 2884h | 2888h | 288Ch | 2890h | 2894h | 2898h | 289Ch |
| 197 | 28A0h | 28A4h | 28A8h | 28ACh | 28B0h | 28B4h | 28B8h | 28BCh |
| 198 | 28C0h | 28C4h | 28C8h | 28CCh | 28D0h | 28D4h | 28D8h | 28DCh |
| 199 | 28E0h | 28E4h | 28E8h | 28ECh | 28F0h | 28F4h | 28F8h | 28FCh |
| 200 | 2900h | 2904h | 2908h | 290Ch | 2910h | 2914h | 2918h | 291Ch |
| 201 | 2920h | 2924h | 2928h | 292Ch | 2930h | 2934h | 2938h | 293Ch |
| 202 | 2940h | 2944h | 2948h | 294Ch | 2950h | 2954h | 2958h | 295Ch |
| 203 | 2960h | 2964h | 2968h | 296Ch | 2970h | 2974h | 2978h | 297Ch |
| 204 | 2980h | 2984h | 2988h | 298Ch | 2990h | 2994h | 2998h | 299Ch |
| 205 | 29A0h | 29A4h | 29A8h | 29ACh | 29B0h | 29B4h | 29B8h | 29BCh |
| 206 | 29C0h | 29C4h | 29C8h | 29CCh | 29D0h | 29D4h | 29D8h | 29DCh |
| 207 | 29E0h | 29E4h | 29E8h | 29ECh | 29F0h | 29F4h | 29F8h | 29FCh |
| 208 | 2A00h | 2A04h | 2A08h | 2A0Ch | 2A10h | 2A14h | 2A18h | 2A1Ch |
| 209 | 2A20h | 2A24h | 2A28h | 2A2Ch | 2A30h | 2A34h | 2A38h | 2A3Ch |
| 210 | 2A40h | 2A44h | 2A48h | 2A4Ch | 2A50h | 2A54h | 2A58h | 2A5Ch |
| 211 | 2A60h | 2A64h | 2A68h | 2A6Ch | 2A70h | 2A74h | 2A78h | 2A7Ch |
| 212 | 2A80h | 2A84h | 2A88h | 2A8Ch | 2A90h | 2A94h | 2A98h | 2A9Ch |
| 213 | 2AA0h | 2AA4h | 2AA8h | 2AACh | 2AB0h | 2AB4h | 2AB8h | 2ABCh |
| 214 | 2AC0h | 2AC4h | 2AC8h | 2ACCh | 2AD0h | 2AD4h | 2AD8h | 2ADCh |
| 215 | 2AE0h | 2AE4h | 2AE8h | 2AECh | 2AF0h | 2AF4h | 2AF8h | 2AFCh |
| 216 | 2B00h | 2B04h | 2B08h | 2B0Ch | 2B10h | 2B14h | 2B18h | 2B1Ch |
| 217 | 2B20h | 2B24h | 2B28h | 2B2Ch | 2B30h | 2B34h | 2B38h | 2B3Ch |
| 218 | 2B40h | 2B44h | 2B48h | 2B4Ch | 2B50h | 2B54h | 2B58h | 2B5Ch |
| 219 | 2B60h | 2B64h | 2B68h | 2B6Ch | 2B70h | 2B74h | 2B78h | 2B7Ch |
| 220 | 2B80h | 2B84h | 2B88h | 2B8Ch | 2B90h | 2B94h | 2B98h | 2B9Ch |
| 221 | 2BA0h | 2BA4h | 2BA8h | 2BACH | 2BB0h | 2BB4h | 2BB8h | 2BBCh |
| 222 | 2BC0h | 2BC4h | 2BC8h | 2BCCh | 2BD0h | 2BD4h | 2BD8h | 2BDCh |
| 223 | 2BE0h | 2BE4h | 2BE8h | 2BECh | 2BF0h | 2BF4h | 2BF8h | 2BFCh |
| 224 | 2C00h | 2C04h | 2C08h | 2C0Ch | 2C10h | 2C14h | 2C18h | 2C1Ch |
| 225 | 2C20h | 2C24h | 2C28h | 2C2Ch | 2C30h | 2C34h | 2C38h | 2C3Ch |
| 226 | 2C40h | 2C44h | 2C48h | 2C4Ch | 2C50h | 2C54h | 2C58h | 2C5Ch |

Окончание таблицы Б.12

| Объект сообщения | МО FCR | МО FGPR | МО IPR | МО AMR | МО DATAL | МО DATAH | МО AR | МОСТР/MOSTAT |
|--|--------|---------|--------|--------|----------|----------|-------|--------------|
| 227 | 2C60h | 2C64h | 2C68h | 2C6Ch | 2C70h | 2C74h | 2C78h | 2C7Ch |
| 228 | 2C80h | 2C84h | 2C88h | 2C8Ch | 2C90h | 2C94h | 2C98h | 2C9Ch |
| 229 | 2CA0h | 2CA4h | 2CA8h | 2CACH | 2CB0h | 2CB4h | 2CB8h | 2CBCh |
| 230 | 2CC0h | 2CC4h | 2CC8h | 2CCCh | 2CD0h | 2CD4h | 2CD8h | 2CDCh |
| 231 | 2CE0h | 2CE4h | 2CE8h | 2CECh | 2CF0h | 2CF4h | 2CF8h | 2CFCh |
| 232 | 2D00h | 2D04h | 2D08h | 2D0Ch | 2D10h | 2D14h | 2D18h | 2D1Ch |
| 233 | 2D20h | 2D24h | 2D28h | 2D2Ch | 2D30h | 2D34h | 2D38h | 2D3Ch |
| 234 | 2D40h | 2D44h | 2D48h | 2D4Ch | 2D50h | 2D54h | 2D58h | 2D5Ch |
| 235 | 2D60h | 2D64h | 2D68h | 2D6Ch | 2D70h | 2D74h | 2D78h | 2D7Ch |
| 236 | 2D80h | 2D84h | 2D88h | 2D8Ch | 2D90h | 2D94h | 2D98h | 2D9Ch |
| 237 | 2DA0h | 2DA4h | 2DA8h | 2DACH | 2DB0h | 2DB4h | 2DB8h | 2DBCh |
| 238 | 2DC0h | 2DC4h | 2DC8h | 2DCCh | 2DD0h | 2DD4h | 2DD8h | 2DDCh |
| 239 | 2DE0h | 2DE4h | 2DE8h | 2DECh | 2DF0h | 2DF4h | 2DF8h | 2DFCh |
| 240 | 2E00h | 2E04h | 2E08h | 2E0Ch | 2E10h | 2E14h | 2E18h | 2E1Ch |
| 241 | 2E20h | 2E24h | 2E28h | 2E2Ch | 2E30h | 2E34h | 2E38h | 2E3Ch |
| 242 | 2E40h | 2E44h | 2E48h | 2E4Ch | 2E50h | 2E54h | 2E58h | 2E5Ch |
| 243 | 2E60h | 2E64h | 2E68h | 2E6Ch | 2E70h | 2E74h | 2E78h | 2E7Ch |
| 244 | 2E80h | 2E84h | 2E88h | 2E8Ch | 2E90h | 2E94h | 2E98h | 2E9Ch |
| 245 | 2EA0h | 2EA4h | 2EA8h | 2EACH | 2EB0h | 2EB4h | 2EB8h | 2EBCh |
| 246 | 2EC0h | 2EC4h | 2EC8h | 2ECCh | 2ED0h | 2ED4h | 2ED8h | 2EDCh |
| 247 | 2EE0h | 2EE4h | 2EE8h | 2EECh | 2EF0h | 2EF4h | 2EF8h | 2EFCh |
| 248 | 2F00h | 2F04h | 2F08h | 2F0Ch | 2F10h | 2F14h | 2F18h | 2F1Ch |
| 249 | 2F20h | 2F24h | 2F28h | 2F2Ch | 2F30h | 2F34h | 2F38h | 2F3Ch |
| 250 | 2F40h | 2F44h | 2F48h | 2F4Ch | 2F50h | 2F54h | 2F58h | 2F5Ch |
| 251 | 2F60h | 2F64h | 2F68h | 2F6Ch | 2F70h | 2F74h | 2F78h | 2F7Ch |
| 252 | 2F80h | 2F84h | 2F88h | 2F8Ch | 2F90h | 2F94h | 2F98h | 2F9Ch |
| 253 | 2FA0h | 2FA4h | 2FA8h | 2FACH | 2FB0h | 2FB4h | 2FB8h | 2FBCh |
| 254 | 2FC0h | 2FC4h | 2FC8h | 2FCCh | 2FD0h | 2FD4h | 2FD8h | 2FDCh |
| 255 | 2FE0h | 2FE4h | 2FE8h | 2FECh | 2FF0h | 2FF4h | 2FF8h | 2FFCh |
| Адреса 80073000h-8007FFFCh являются зарезервированными | | | | | | | | |

Таблица Б.13 – Мнемоника и соответствующие названия регистров объектов сообщений контроллера CAN

| Мнемоника | Сброс | Название |
|--------------|------------------|--|
| MOFCR | 00000000h | Регистр управления функционированием объекта сообщения |
| MOFGPR | 00000000h | Регистр указателя FIFO/шлюза объекта сообщения |
| MOIPR | 00000000h | Регистр указателя прерываний объекта сообщения |
| MOAMR | 3FFFFFFFh | Регистр маски объекта сообщения |
| MODATAL | 00000000h | Младший регистр данных объекта сообщения |
| MODATAH | 00000000h | Старший регистр данных объекта сообщения |
| MOAR | 00000000h | Регистр арбитража объекта сообщения |
| МОСТР/MOSTAT | См. таблицу Б.14 | Регистр управления/состояния объекта сообщения |

Таблица Б.14 – Состояние регистров МОСТР/MOSTAT объектов сообщений после сброса (n-порядковый номер объекта сообщения)

| n (код) | МОСТР / MOSTAT | | |
|-----------|-----------------------|-----------------------|-----------|
| | PNEXT (биты 31–24) | PPREV (биты 23–16) | Биты 15–0 |
| 0 (00h) | 01h | 00h | 0000h |
| 1 (01h) | 02h | 00h | |
| 2 (02h) | 03h | 01h | |
| 3 (03h) | 04h | 02h | |
| ... | ... | ... | |
| 252 (FCh) | FDh | FBh | |
| 253 (FDh) | FEh | FCh | |
| 254 (FEh) | FFh | FDh | |
| 255 (FFh) | FFh | FEh | |

Таблица Б.15 – Регистры контроллера Ethernet

| Адрес | Мнемоника | Сброс | Название |
|--------------------------|------------|-----------|---|
| 80080000h | MAC1 | 00008000h | Регистр1 конфигурации MAC |
| 80080004h | MAC2 | 00000000h | Регистр2 конфигурации MAC |
| 80080008h | IPGT | 00000000h | Back-to-Back Inter-Packet-Gap регистр |
| 8008000Ch | IPGR | 00000000h | Регистр Non-Back-to-Back Inter-Packet-Gap |
| 80080010h | CLRT | 0000370Fh | Регистр окна коллизий/повторов |
| 80080014h | MAXF | 00000600h | Регистр верхней границы размера Frame |
| 80080018h | SUPP | 00001000h | Регистр поддержки РНУ-интерфейса |
| 8008001Ch | – | – | Зарезервировано |
| 80080020h | MCFG | 00000000h | Регистр управления конфигурацией МП |
| 80080024h | MCMD | 00000000h | Регистр команд МП |
| 80080028h | MADR | 00000000h | Регистр адреса МП |
| 8008002Ch | MWTD | 00000000h | Регистр записываемых данных в МП |
| 80080030h | MRDD | 00000000h | Регистр считываемых данных из МП |
| 80080034h | MIND | 00000000h | Регистр флагов состояния МП |
| 80080038h | SMPI | 00000000h | Регистр состояния |
| 8008003Ch | FIFOCFG | 0000001Fh | Регистр конфигурации буфера МПFIFO |
| 80080040h | SA0 | 00000000h | Регистр адреса станции SA0 |
| 80080044h | SA1 | 00000000h | Регистр адреса станции SA1 |
| 80080048h | SA2 | 00000000h | Регистр адреса станции SA2 |
| 8008004Ch- 8008017Ch | – | – | Зарезервировано |
| 80080180h | DMATxCTRL | 00000000h | Регистр управления передачей |
| 80080184h | DMATxDSCR | 00000000h | Регистр указателя дескриптора передачи |
| 80080188h | DMATxSTAT | 00000000h | Регистр статуса передачи |
| 8008018Ch | DMARxCTRL | 00000000h | Регистр управления приемом |
| 80080190h | DMARxDSCR | 00000000h | Регистр указателя дескриптора приема |
| 80080194h | DMARxSTAT | 00000000h | Регистр статуса приема |
| 80080198h | DMAINTMASK | 00000000h | Регистр маски прерывания |
| 8008019Ch | DMAINT | 00000000h | Регистр прерываний |
| 800881A0h- 8008FFFCCh | – | – | Зарезервировано |

Таблица Б.16 – Регистры контроллера USB (хост)

| Адрес | Мнемоника | Сброс | Название |
|-----------|------------------|-----------|---|
| 80090000h | VERSION | 00000001h | Регистр версии |
| 80090004h | USBCMD_STS_INTR | 01008000h | Регистр управления 1 |
| 80090008h | FRAME_REG | 00200000h | Регистр кадра |
| 8009000Ch | – | – | Зарезервировано |
| 80090010h | PORT_STS | 00000000h | Регистр управления 1 |
| 80090014h | USB_PKT_FIELDS_A | 00000000h | Регистр А управления передачей данных |
| 80090018h | USB_PKT_FIELDS_B | 00000000h | Регистр В управления передачей данных |
| 8009001Ch | BUFF_START_ADDR | 00000000h | Регистр адреса начала массива передаваемых данных |
| 80090020h | TBT | 00000000h | Регистр передаваемых данных |
| 80090024h | USB_STATUS | 00000000h | Регистр состояния системы |
| 80090028h | – | – | Зарезервировано |
| 8009002Ch | Data_Buffer | 00000000h | Регистр данных. |
| 80090030h | SLAVE_IN_COUNT | 00000000h | Регистр принятых данных |

Таблица Б.17 – Регистры контроллера USB (устройство)

| | | | |
|-------------------------------------|----------------|-----------|---|
| 80090000h | IRQ_STAT_L | 00000000h | Регистр состояния прерываний |
| 80090008h | IRQ_ENB_L | 00000000h | Регистр разрешения прерываний |
| 80090010h | USB_IRQ_STAT | 00000004h | Регистр состояния прерываний |
| 80090014h | USB_IRQ_ENB | 00000040h | Регистр разрешения прерываний |
| 80090018h | USB_OPER | 00000002h | Регистр операций |
| 8009001Ch | USB_FRAME_CNT | 00000000h | Регистр счетчика кадров |
| 80090020h | USB_ADDR | 00000000h | Регистр адреса |
| 80090024h | – | – | Зарезервировано |
| Регистры контрольной конечной точки | | | |
| 80090028h | CEP_DATA_BUF | 00000000h | Регистр данных для передачи |
| 8009002Ch | CEP_CTRL_STAT | 00000000h | Регистр управления и состояния буфера |
| 80090030h | CEP_IRQ_ENB | 00000000h | Регистр разрешения прерываний буфера |
| 80090034h | CEP_IRQ_STAT | 00000000h | Регистр флагов прерываний буфера |
| 80090038h | CEP_IN_XFRCNT | 00000000h | Регистр числа байт для отправки |
| 8009003Ch | CEP_OUT_XFRCNT | 00000000h | Регистр числа принятых байт |
| 80090040h | – | – | Зарезервировано |
| 80090044h | CEP_SETUP1_0 | 00000000h | Регистр нулевого и первого байтов пакета Setup |
| 80090048h | CEP_SETUP3_2 | 00000000h | Регистр второго и третьего байтов пакета Setup |
| 8009004Ch | CEP_SETUP5_4 | 00000000h | Регистр четвертого и пятого байтов пакета Setup |
| 80090050h | CEP_SETUP7_6 | 00000000h | Регистр шестого и седьмого байтов пакета Setup |
| 80090054h | CEP_START_ADDR | 00000000h | Регистр начала адресного пространства |
| 80090058h | CEP_END_ADDR | 00000000h | Регистр конца адресного пространства |
| 8009005Ch | DMA_CTRL_STS | 00000000h | Регистр управления DMA |
| 80090060h | DMA_CNT | 00000000h | Регистр количества байт |

Таблица Б.18 – Регистры конечной точки (x – 0, 1)

| Мнемоника | Адреса регистров | | Сброс | Название |
|---------------------|------------------|-----------|-------|---|
| | EP0 | EP1 | | |
| USB_EP_x_DATA_BUF | 80090064h | 8009008Ch | 0000h | Регистр обмена данными с буфером |
| USB_EP_x_IRQ_STAT | 80090068h | 80090090h | 0002h | Регистр флагов прерываний не управляющего буфера |
| USB_EP_x_IRQ_ENB | 8009006Ch | 80090094h | 0002h | Регистр разрешения прерываний не управляющего буфера |
| USB_EP_x_AVAIL_CNT | 80090070h | 80090098h | 0000h | Регистр числа байт в буфере |
| USB_EP_x_RSP_SC | 80090074h | 8009009Ch | 0000h | Регистр установки и сброса отклика |
| USB_EP_x_MPS | 80090078h | 800900A0h | 0000h | Регистр максимального размера пакета |
| USB_EP_x_CNT | 8009007Ch | 800900A4h | 0000h | Регистр числа байт буфера типа IN |
| USB_EP_x_EP_CFG | 80090080h | 800900A8h | 0000h | Регистр конфигурации не управляющего буфера |
| USB_EP_x_START_ADDR | 80090084h | 800900ACh | 0000h | Регистр начала адресного пространства, выделенного под буфер. |
| USB_EP_x_END_ADDR | 80090088h | 800900B0h | 0000h | Регистр конца адресного пространства, выделенного под буфер |

Таблица Б.19 – Регистры контроллера USB

| Адрес | Мнемоника | Сброс | Название |
|---------------------|--------------|-----------|---|
| 800900B4h-800906FCh | – | – | Зарезервировано |
| 80090700h | AHB_DMA_ADDR | 00000000h | Регистр адреса записи и чтения в режиме DMA |
| 80090704h | OTG_IRQ_STAT | 00000000h | Регистр состояния |
| 80090708h | OTG_IRQ_EN | 00000000h | Регистр разрешения прерываний |
| 8009070Ch-9FFFFFFCh | – | – | Зарезервировано |

Таблица Б.20 – Регистры сторожевого таймера

| Адрес | Мнемоника | Сброс | Название |
|---------------------|-----------|-----------|---|
| A0000000h | LOAD | FFFFFFFFh | Регистр начального значения счетчика |
| A0000004h | VALUE | FFFFFFFFh | Регистр текущего значения счетчика |
| A0000008h | CTRL | 00000000h | Регистр программного управления |
| A000000Ch | INTCLR | 00000000h | Регистр сброса |
| A0000010h | RIS | 00000000h | Регистр прерывания |
| A0000014h | MIS | 00000000h | Регистр маскированного прерывания счетчика |
| A0000018h-A00000BCh | – | – | Зарезервировано |
| A0000C00h | CLOCK | 00000000h | Регистр управления доступом к остальным регистрам блока |
| A0000C04h-A0000FFCh | – | – | Зарезервировано |

Таблица Б.21 – Адреса регистров контроллеров I2C

| Мнемоника | Адреса регистров | | Сброс | Название |
|-----------|-------------------------|-------------------------|-------|-------------------------------|
| | I2C0 | I2C1 | | |
| SDA | A0001000h | A0002000h | xxh | Сдвиговой регистр данных |
| ST | A0001004h | A0002004h | 00h | Регистр состояния |
| CST | A0001008h | A0002008h | 00h | Регистр управления и статуса |
| CTL0 | A000100Ch | A000200Ch | 00h | Регистр 0 управления |
| ADDR | A0001010h | A0002010h | 00h | Регистр собственного адреса |
| CTL1 | A0001014h | A0002014h | 00h | Регистр 1 управления |
| TOPR | A0001018h | A0002018h | 00h | Регистр загрузки предделителя |
| CTL2 | A000101Ch | A000201Ch | 00h | Регистр 2 управления |
| – | A0001020h- A0001FFCh | A0002020h- A0002FFCh | – | Зарезервировано |

Таблица Б.22 – Адреса регистров таймеров

| Мнемоника | Таймеры | | | Сброс | Название |
|------------------------|-----------------|-----------------|-----------------|-----------|--------------------------------------|
| | Адрес A000_xxxx | | | | |
| | Таймер 0 | Таймер 1 | Таймер 2 | | |
| CTRL | 3000h | 4000h | 5000h | 0000h | Регистр управления |
| VALUE | 3004h | 4004h | 5004h | 00000000h | Регистр текущего значения |
| RELOAD | 3008h | 4008h | 5008h | 00000000h | Регистр начального значения счетчика |
| INTSTATUS_ INTCLEAR | 300Ch | 400Ch | 500Ch | 0000h | Регистр прерывания |
| – | 3010h- 3FFCh | 4010h- 4FFCh | 5010h- 5FFCh | – | Зарезервировано |

Таблица Б.23 – Регистры контроллера DMA

| Адрес | Мнемоника | Сброс | Название |
|-----------|-----------------------|-----------|---|
| A0006000h | STATUS | 0nn0000h | Регистр статуса |
| A0006004h | CFG | – | Регистр конфигурации |
| A0006008h | CTRL_BASE_PTR | 00000000h | Регистр базового адреса управляющих данных каналов |
| A000600Ch | ALT_CTRL_ BASE_PTR | 000000nnh | Регистр базового адреса альтернативных управляющих данных каналов |
| A0006010h | WAITONREQ_ STATUS | 00000000h | Регистр статуса ожидания запроса на обработку каналов |
| A0006014h | CHNL_SW_ REQUEST | – | Регистр программного запроса на обработку каналов |
| A0006018h | CHNL_ USEBURST_SET | 00000000h | Регистр установки пакетного обмена каналов |
| A000601Ch | CHNL_ USEBURST_CLR | – | Регистр сброса пакетного обмена каналов |
| A0006020h | CHNL_REQ_ MASK_SET | 00000000h | Регистр маскирования запросов на обслуживание каналов |
| A0006024h | CHNL_REQ_ MASK_CLR | – | Регистр очистки маскирования запросов на обслуживание каналов |
| A0006028h | CHNL_ ENABLE_SET | 00000000h | Регистр установки разрешения каналов |

Окончание таблицы Б.23

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|-------------------|-----------|---|
| A000602Ch | CHNL_ENABLE_CLR | – | Регистр сброса разрешения каналов |
| A0006030h | CHNL_PRI_ALT_SET | 00000000h | Регистр установки первичной/альтернативной структуры управляющих данных каналов |
| A0006034h | CHNL_PRI_ALT_CLR | – | Регистр сброса первичной/альтернативной структуры управляющих данных каналов |
| A0006038h | CHNL_PRIORITY_SET | 00000000h | Регистр установки приоритета каналов |
| A000603Ch | CHNL_PRIORITY_CLR | – | Регистр сброса установок приоритета каналов |
| A0006040h- A0006048h | – | – | Зарезервировано |
| A000604Ch | ERR_CLR | 00000000h | Регистр сброса флага ошибки |
| A0006050h- A0006FFCh | – | – | Зарезервировано |

Таблица Б.24 – Адреса регистров контроллеров UART

| Мнемоника | Контроллеры UART | | | | Сброс | Название |
|-----------|------------------|-----------------|-----------------|-----------------|-------|---|
| | Адрес A000_xxxx | | | | | |
| | UART0 | UART 1 | UART 2 | UART 3 | | |
| DR | 7000h | 8000h | 9000h | A000h | 0000h | Регистр данных |
| RSR_ECR | 7004h | 8004h | 9004h | A004h | 0000h | Регистр состояния приемника и сброса ошибки приемника |
| – | 7008h- 7014h | 8008h- 8014h | 9008h- 9014h | A008h- A014h | – | Зарезервировано |
| FR | 7018h | 8018h | 9018h | A018h | 0090h | Регистр флагов |
| – | 701Ch, 7020h | 801Ch, 8020h | 901Ch, 9020h | A01Ch, A020h | – | Зарезервировано |
| IBRD | 7024h | 8024h | 9024h | A024h | 0000h | Регистр целой части делителя |
| FBRD | 7028h | 8028h | 9028h | A028h | 0000h | Регистр дробной части делителя |
| LCR_H | 702Ch | 802Ch | 902Ch | A02Ch | 0000h | Регистр управления линией |
| CR | 7030h | 8030h | 9030h | A030h | 0300h | Регистр управления |
| IFLS | 7034h | 8034h | 9034h | A034h | 0012h | Регистр порога прерывания по заполнению буфера FIFO |
| IMSC | 7038h | 8038h | 9038h | A038h | 0000h | Регистр маски прерывания |
| RIS | 703Ch | 803Ch | 903Ch | A03Ch | 0000h | Регистр состояния прерываний |
| MIS | 7040h | 8040h | 9040h | A040h | 0000h | Регистр состояния прерываний с маскированием |
| ICR | 7044h | 8044h | 9044h | A044h | 0000h | Регистр сброса прерывания |
| DMACR | 7048h | 8048h | 9048h | A048h | 0000h | Регистр управления прямым доступом к памяти |
| – | 7050h- 7FFCh | 8050h- 8FFCh | 9050h- 9FFCh | A050h- AFFCh | – | Зарезервировано |

Таблица Б.25 – Адреса регистров блоков ШИМ

| Мнемоника | Блоки ШИМ | | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | Адрес A000_хххх | | | | | Адрес A001_хххх | | | |
| | PWM0 | PWM 1 | PWM 2 | PWM 3 | PWM 4 | PWM 5 | PWM 6 | PWM 7 | PWM 8 |
| TBCTL | B000h | C000h | D000h | E000h | F000h | 0000h | 1000h | 2000h | 3000h |
| TBSTS | B004h | C004h | D004h | E004h | F004h | 0004h | 1004h | 2004h | 3004h |
| TBPHS | B008h | C008h | D008h | E008h | F008h | 0008h | 1008h | 2008h | 3008h |
| TBCTR | B00Ch | C00Ch | D00Ch | E00Ch | F00Ch | 000Ch | 100Ch | 200Ch | 300Ch |
| TBPRD | B010h | C010h | D010h | E010h | F010h | 0010h | 1010h | 2010h | 3010h |
| CMPCCTL | B014h | C014h | D014h | E014h | F014h | 0014h | 1014h | 2014h | 3014h |
| CMPPA | B018h | C018h | D018h | E018h | F018h | 0018h | 1018h | 2018h | 3018h |
| CMPPB | B01Ch | C01Ch | D01Ch | E01Ch | F01Ch | 001Ch | 101Ch | 201Ch | 301Ch |
| AQCTLA | B020h | C020h | D020h | E020h | F020h | 0020h | 1020h | 2020h | 3020h |
| AQCTLB | B024h | C024h | D024h | E024h | F024h | 0024h | 1024h | 2024h | 3024h |
| AQSFRC | B028h | C028h | D028h | E028h | F028h | 0028h | 1028h | 2028h | 3028h |
| AQCSFRC | B02Ch | C02Ch | D02Ch | E02Ch | F02Ch | 002Ch | 102Ch | 202Ch | 302Ch |
| DBCTL | B030h | C030h | D030h | E030h | F030h | 0030h | 1030h | 2030h | 3030h |
| DBRED | B034h | C034h | D034h | E034h | F034h | 0034h | 1034h | 2034h | 3034h |
| DBFED | B038h | C038h | D038h | E038h | F038h | 0038h | 1038h | 2038h | 3038h |
| TZSEL | B03Ch | C03Ch | D03Ch | E03Ch | F03Ch | 003Ch | 103Ch | 203Ch | 303Ch |
| TZCTL | B040h | C040h | D040h | E040h | F040h | 0040h | 1040h | 2040h | 3040h |
| TZEINT | B044h | C044h | D044h | E044h | F044h | 0044h | 1044h | 2044h | 3044h |
| TZFLG | B048h | C048h | D048h | E048h | F048h | 0048h | 1048h | 2048h | 3048h |
| TZCLR | B04Ch | C04Ch | D04Ch | E04Ch | F04Ch | 004Ch | 104Ch | 204Ch | 304Ch |
| TZFRC | B050h | C050h | D050h | E050h | F050h | 0050h | 1050h | 2050h | 3050h |
| ETSEL | B054h | C054h | D054h | E054h | F054h | 0054h | 1054h | 2054h | 3054h |
| ETPS | B058h | C058h | D058h | E058h | F058h | 0058h | 1058h | 2058h | 3058h |
| ETFLG | B05Ch | C05Ch | D05Ch | E05Ch | F05Ch | 005Ch | 105Ch | 205Ch | 305Ch |
| ETCLR | B060h | C060h | D060h | E060h | F060h | 0060h | 1060h | 2060h | 3060h |
| ETFRC | B064h | C064h | D064h | E064h | F064h | 0064h | 1064h | 2064h | 3064h |
| PCCTL | B068h | C068h | D068h | E068h | F068h | 0068h | 1068h | 2068h | 3068h |
| HRCNFG | B06Ch | C06Ch | D06Ch | E06Ch | F06Ch | 006Ch | 106Ch | 206Ch | 306Ch |
| FWDTH | B070h | C070h | D070h | E070h | F070h | 0070h | 1070h | 2070h | 3070h |
| Зарезервировано | B074h- B084h | C074h- C084h | D074h- D084h | E074h- E084h | F074h- F084h | 0074h- 0084h | 1074h- 1084h | 2074h- 2084h | 3074h- 3084h |
| HDSEL | B088h | C088h | D088h | E088h | F088h | 0088h | 1088h | 2088h | 3088h |
| HDCTL | B08Ch | C08Ch | D08Ch | E08Ch | F08Ch | 008Ch | 108Ch | 208Ch | 308Ch |
| HDEINT | B090h | C090h | D090h | E090h | F090h | 0090h | 1090h | 2090h | 3090h |
| HDFLG | B094h | C094h | D094h | E094h | F094h | 0094h | 1094h | 2094h | 3094h |
| HDCLR | B098h | C098h | D098h | E098h | F098h | 0098h | 1098h | 2098h | 3098h |
| HDFRC | B09Ch | C09Ch | D09Ch | E09Ch | F09Ch | 009Ch | 109Ch | 209Ch | 309Ch |
| HDINTCLR | B0A0h | C0A0h | D0A0h | E0A0h | F0A0h | 00A0h | 10A0h | 20A0h | 30A0h |
| TZINTCLR | B0A4h | C0A4h | D0A4h | E0A4h | F0A4h | 00A4h | 10A4h | 20A4h | 30A4h |
| INTCLR | B0A8h | C0A8h | D0A8h | E0A8h | F0A8h | 00A8h | 10A8h | 20A8h | 30A8h |
| Зарезервировано | B0ACh- BFFCh | C0ACh- CFFCh | D0ACh- DFFCh | E0ACh- EFFCh | F0ACh- FFFCh | 00ACh- 0FFCh | 10ACh- 1FFCh | 20ACh- 2FFCh | 30ACh- 3FFCh |

Таблица Б.26 – Мнемоника и названия регистров блоков ШИМ

| Мнемоника | Сброс | Название |
|-----------|-------|--|
| TBCTL | 00h | Регистр управления таймером |
| TBSTS | 00h | Регистр статуса таймера |
| TBPHS | 00h | Регистр фазы |
| TBCTR | 00h | Регистр текущего значения таймера ШИМ |
| TBPRD | 00h | Регистр максимального значения таймера |
| CMPCCTL | 00h | Регистр управления компаратором |
| CMPA | 00h | Регистр порога срабатывания А |
| CMPB | 00h | Регистр порога срабатывания В |
| AQCTLA | 00h | Регистр обработчика для выхода А |
| AQCTLB | 00h | Регистр обработчика для выхода В |
| AQSFR | 00h | Регистр обработчика для однократного программного управления |
| AQCSFR | 00h | Регистр обработчика для циклического программного управления |
| DBCTL | 00h | Регистр управления генератором «мертвого времени» блока ШИМ |
| DBRED | 00h | Регистр задержки фронта |
| DBFED | 00h | Регистр задержки среза |
| TZSEL | 00h | Регистр источника сигнала аварии |
| TZCTL | 00h | Регистр управления детектором событий аварии |
| TZEINT | 00h | Регистр маски прерывания детектора событий аварии |
| TZFLG | 00h | Регистр флагов прерывания детектора событий аварии |
| TZCLR | 00h | Регистр сброса флагов прерывания детектора событий аварии |
| TZFRC | 00h | Регистр программной эмуляции сигнала аварии |
| ETSEL | 00h | Регистр источника триггера событий |
| ETPS | 00h | Регистр предделителя триггера событий |
| ETFLG | 00h | Регистр флагов триггера событий |
| ETCLR | 00h | Регистр сброса флагов триггера событий |
| ETFRC | 00h | Регистр программной эмуляции событий |
| PCCTL | 00h | Регистр управления модулятором |
| HRCNFG | 00h | Регистр конфигурации блока ШИМ высокого разрешения |
| FWDTH | 00h | Регистр ширины фильтрации |
| HDSEL | 00h | Регистр источника события удержания |
| HDCTL | 00h | Регистр управления детектором событий удержания |
| HDEINT | 00h | Регистр маски прерывания порогового выключателя |
| HDFLG | 00h | Регистр флагов прерывания порогового выключателя |
| HDCLR | 00h | Регистр сброса флагов порогового выключателя |
| HDFRC | 00h | Регистр программной активации порогового выключателя |
| HDINTCLR | 00h | Регистр сброса прерывания порогового выключателя |
| TZINTCLR | 00h | Регистр сброса прерывания детектора событий аварии |
| INTCLR | 00h | Регистр сброса прерывания таймера блока ШИМ |

Таблица Б.27 – Адреса регистров блоков захвата

| Мнемо-ника | Блоки захвата | | | | | | Название |
|---|-----------------|-------------|-------------|-------------|-------------|-------------|---------------------------------------|
| | Адрес A001_xxxx | | | | | | |
| | CAP0 | CAP 1 | CAP 2 | CAP 3 | CAP 4 | CAP 5 | |
| TSCTR | 4000h | 5000h | 6000h | 7000h | 8000h | 9000h | Регистр таймера |
| STRPHS | 4004h | 5004h | 6004h | 7004h | 8004h | 9004h | Регистр загрузки таймера |
| CAP0 | 4008h | 5008h | 6008h | 7008h | 8008h | 9008h | Регистр захвата 0 |
| CAP1 | 400Ch | 500Ch | 600Ch | 700Ch | 800Ch | 900Ch | Регистр захвата 1 |
| CAP2 | 4010h | 5010h | 6010h | 7010h | 8010h | 9010h | Регистр захвата 2 |
| CAP3 | 4014h | 5014h | 6014h | 7014h | 8014h | 9014h | Регистр захвата 3 |
| – | 4018h-4024h | 5018h-5024h | 6018h-6024h | 7018h-7024h | 8018h-8024h | 9018h-9024h | Зарезервировано |
| ECCTL0 | 4028h | 5028h | 6028h | 7028h | 8028h | 9028h | Регистр контроля захвата 0 |
| ECCTL1 | 402Ch | 502Ch | 602Ch | 702Ch | 802Ch | 902Ch | Регистр контроля захвата 1 |
| ECEINT | 4030h | 5030h | 6030h | 7030h | 8030h | 9030h | Регистр маски прерываний |
| ECFLG | 4034h | 5034h | 6034h | 7034h | 8034h | 9034h | Регистр статуса прерываний |
| ECCLR | 4038h | 5438h | 6038h | 7038h | 8038h | 9038h | Регистр сброса прерываний |
| ECFRC | 403Ch | 503Ch | 603Ch | 703Ch | 803Ch | 903Ch | Тестовый регистр генерации прерываний |
| PEINT | 4040h | 5040h | 6040h | 7040h | 8040h | 9040h | Регистр прерывания |
| – | 4044h-4FFCh | 5044h-5FFCh | 6044h-6FFCh | 7044h-7FFCh | 8044h-8FFCh | 9044h-9FFCh | Зарезервировано |
| Примечание – Состояние всех регистров после сброса 00000000h. | | | | | | | |

Таблица Б.28 – Адреса регистров квадратурных декодеров

| Мнемоника | Декодеры | | Название |
|-----------|-----------------|-------|---|
| | Адрес A001_xxxx | | |
| | QEP0 | QEP1 | |
| QPOSCNT | A000h | B000h | Регистр счетчика позиции |
| QPOSINIT | A004h | B004h | Регистр загрузки счетчика позиции |
| QPOSMAX | A008h | B008h | Регистр максимального значения счетчика позиции |
| QPOSCMP | A00Ch | B00Ch | Регистр сравнения счетчика позиции |
| QPOSILAT | A010h | B010h | Регистр хранения позиции по индексации |
| QPOSSLAT | A014h | B014h | Регистр хранения позиции по стробу |
| QPOSLAT | A018h | B018h | Регистр хранения позиции по сторожевому таймеру |
| QUTMR | A01Ch | B01Ch | Регистр таймера временных отсчетов |
| QUPRD | A020h | BA20h | Регистр порога таймера временных отсчетов |
| QWDTMR | A024h | B024h | Регистр сторожевого таймера |
| QWDPRD | A028h | B028h | Регистр длительности сторожевого отсчета |
| QDECCTL | A02Ch | B02Ch | Регистр управления входами |
| QEPCTL | A030h | B030h | Регистр управления квадратурного декодера |
| QCAPCTL | A034h | B034h | Регистр блока захвата |
| QPOSCCTL | A038h | B038h | Регистр управления счетчиком позиции |
| QEINT | A03Ch | B03Ch | Регистр масок прерываний |

Окончание таблицы Б.28

| Мнемоника | QEP0 | QEP1 | Название |
|---|-----------------|-----------------|---|
| QFLG | A040h | B040h | Регистр флагов прерываний |
| QCLR | A044h | B044h | Регистр сброса прерываний |
| QFRC | A048h | B048h | Регистр генерации прерываний |
| QEPSTS | A04Ch | B04Ch | Регистр статуса |
| QCTMR | A050h | B050h | Регистр таймера |
| QCPRD | A054h | B054h | Регистр длительности измерения |
| QCTMRLAT | A058h | B058h | Регистр хранения таймера |
| QCPRDLAT | A05Ch | B05Ch | Регистр хранения длительности измерения |
| – | A060h- A06Ch | B060h- B06Ch | Зарезервировано |
| INTCLR | A070h | B070h | Регистр сброса прерываний |
| – | A074h- AFFCh | B074h- BFFCh | Зарезервировано |
| Примечание – Состояние всех регистров после сброса 00000000h. | | | |

Таблица Б.29 – Регистры загрузочной флеш-памяти

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|-----------|-------|---------------------------------|
| A001C000h | FMA | | Регистр адреса |
| A001C004h | FMD0 | | Регистр 0 слова данных |
| A001C008h | FMC | | Регистр команд |
| A001C00Ch | FCIS | | Регистр статуса |
| A001C010h | FCIM | | Регистр маски прерываний |
| A001C014h | FCIC | | Регистр сброса флагов статуса |
| A001C018h- A001C04Ch | – | – | Зарезервировано |
| A001C050h | FMD1 | | Регистр 1 слова данных |
| A001C054h | FMD2 | | Регистр 2 слова данных |
| A001C058h | FMD3 | | Регистр 3 слова данных |
| A001C05Ch- A001C07Ch | – | – | Зарезервировано |
| A001C080h | T_ACC | | Регистр временных параметров 1 |
| A001C084h | T_NVS | | Регистр временных параметров 2 |
| A001C088h | T_NVH | | Регистр временных параметров 3 |
| A001C08Ch | T_RCV | | Регистр временных параметров 4 |
| A001C090h | T_PGS | | Регистр временных параметров 5 |
| A001C094h | T_PROG | | Регистр временных параметров 6 |
| A001C098h | T_PGH | | Регистр временных параметров 7 |
| A001C09Ch | T_ERASE | | Регистр временных параметров 8 |
| A001C0A0h | T_ME | | Регистр временных параметров 9 |
| A001C0A4h | T_NVH1 | | Регистр временных параметров 10 |
| A001C0A8h- A001CFFCh | – | – | Зарезервировано |

Таблица Б.30 – Регистры аналогового компаратора

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|-----------|-----------|---|
| A001D000h | MIS | 00000000h | Регистр маскированных статусов прерываний |
| A001D004h | RIS | 00000000h | Регистр статусов прерываний |
| A001D008h | INTEN | 00000000h | Регистр маски прерываний |
| A001D00Ch | REFCTL0 | 00000000h | Регистр управления опорным напряжением 0 |
| A001D010h | REFCTL1 | 00000000h | Регистр управления опорным напряжением 1 |
| A001D014h | REFCTL2 | 00000000h | Регистр управления опорным напряжением 2 |
| A001D018h- A001D01Ch | – | – | Зарезервировано |
| A001D020h | STAT0 | 00000000h | Регистр статуса компаратора 0 |
| A001D024h | CTL0 | 00000000h | Регистр управления компаратором 0 |
| A001D028h- A001D03Ch | – | – | Зарезервировано |
| A001D040h | STAT1 | 00000000h | Регистр статуса компаратора 1 |
| A001D044h | CTL1 | 00000000h | Регистр управления компаратором 1 |
| A001D048h- A001D05Ch | – | – | Зарезервировано |
| A001D060h | STAT2 | 00000000h | Регистр статуса компаратора 2 |
| A001D064h | CTL2 | 00000000h | Регистр управления компаратором 2 |
| A001D07Ch | POWER | 00000000h | Регистр питания аналоговой части |
| A001D068h- A001DFFCh | – | – | Зарезервировано |

Таблица Б.31 – Адреса регистров контроллеров SPI

| Мнемо-ника | Контроллеры SPI | | | | Сброс | Название |
|------------|--------------------|-----------------|--------------------|-----------------|-------|---|
| | Адрес A001_xxxx | | Адрес A002_xxxx | | | |
| | SPI0 | SPI1 | SPI2 | SPI3 | | |
| SPI_CR0 | E000h | F000h | 0000h | 1000h | 0000h | Регистр управления 0 |
| SPI_CR1 | E004h | F004h | 0004h | 1004h | 00h | Регистр управления 1 |
| SPI_DR | E008h | F008h | 0008h | 1008h | 0xh | Буфер FIFO приемника и передатчика |
| SPI_SR | E00Ch | F00Ch | 000Ch | 100Ch | 03h | Регистр состояния |
| SPI_CPSR | E010h | F010h | 0010h | 1010h | 00h | Регистр делителя тактовой частоты |
| SPI_IMSC | E014h | F014h | 0014h | 1014h | 00h | Регистр маски прерывания |
| SPI_RIS | E018h | F018h | 0018h | 1018h | 08h | Регистр состояния прерываний без учета маскирования |
| SPI_MIS | E01Ch | F01Ch | 001Ch | 101Ch | 00h | Регистр состояния прерываний с учетом маскирования |
| SPI_ICR | E020h | F020h | 0020h | 1020h | 00h | Регистр сброса прерывания |
| SPI_DMACR | E024h | F024h | 0024h | 1024h | 00h | Регистр управления прямым доступом к памяти |
| – | E028h- EFFCh | F028h- FFFCh | 0028h- 0FFCh | 1028h- 1FFCh | – | Зарезервировано |

Таблица Б.32 – Регистры пользовательской флеш-памяти

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|-----------|-------|---------------------------------|
| A0022000h | FMA | 0000h | Регистр адреса |
| A0022004h | FMD | 0000h | Регистр 1 слова данных |
| A0022008h | FMC | 0000h | Регистр команд |
| A002200Ch | FCIS | 0000h | Регистр статуса |
| A0022010h | FCIM | 0000h | Регистр маски прерываний |
| A0022014h | FCIC | 0000h | Регистр сброса флагов статуса |
| A0022018h- A002207Ch | – | – | Зарезервировано |
| A0022080h | T_ACC | 0000h | Регистр временных параметров 1 |
| A0022084h | T_NV5 | 0000h | Регистр временных параметров 2 |
| A0022088h | T_NVH | 0000h | Регистр временных параметров 3 |
| A002208Ch | T_RCV | 0000h | Регистр временных параметров 4 |
| A0022090h | T_PGS | 0000h | Регистр временных параметров 5 |
| A0022094h | T_PROG | 0000h | Регистр временных параметров 6 |
| A0022098h | T_PGH | 0000h | Регистр временных параметров 7 |
| A002209Ch | T_ERASE | 0000h | Регистр временных параметров 8 |
| A00220A0h | T_ME | 0000h | Регистр временных параметров 9 |
| A00220A4h | T_NVH1 | 0000h | Регистр временных параметров 10 |
| A00220A8h- A0022FFCh | – | – | Зарезервировано |

Таблица Б.33 – Регистры блока RTC

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|-----------|-----------|--|
| A0023000h | PSECOND | 0000h | Регистр долей секунд |
| A0023004h | SECOND | 0000h | Регистр секунд |
| A0023008h | MINUTE | 0000h | Регистр минут |
| A002300Ch | – | – | Зарезервировано |
| A0023010h | HOUR | 0000h | Регистр часов |
| A0023014h | – | – | Зарезервировано |
| A0023018h | DAY | 0000h | Регистр дней |
| A002301Ch | – | – | Зарезервировано |
| A0023020h | DATE | 0000h | Регистр даты |
| A0023024h | MONTH | 0000h | Регистр месяца |
| A0023028h | YEAR | 0000h | Регистр года |
| A002302Ch | SHDW | 00000080h | Регистр управления загрузкой теневых регистров |
| A0023030h | TIME | 0000h | Регистр текущего времени |
| A0023034h- DFFFFFFCh | – | – | Зарезервировано |

Таблица Б.34 – Регистры контроллера прерываний NVIC

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|-----------|-------|---|
| E000E100h | ISER0 | 0000h | Регистр разрешения прерываний по векторам 0 – 31 |
| E000E104h | ISER1 | 0000h | Регистр разрешения прерываний по векторам 32 – 63 |
| E000E108h | ISER2 | 0000h | Регистр разрешения прерываний по векторам 64 – 95 |
| E000E10Ch | ISER3 | 0000h | Регистр разрешения прерываний по векторам 96 – 127 |
| E000E110h | ISER4 | 0000h | Регистр разрешения прерываний по векторам 127 – 133 |
| E000E114h- E000E17Ch | – | – | Зарезервировано |

Продолжение таблицы Б.34

| Адрес | Мнемоника | Сброс | Название |
|-------------------------|-----------|-------|---|
| E000E180h | ICER0 | 0000h | Регистр сброса разрешений прерываний по векторам 0 – 31 |
| E000E184h | ICER1 | 0000h | Регистр сброса разрешений прерываний по векторам 32 – 63 |
| E000E188h | ICER2 | 0000h | Регистр сброса разрешений прерываний по векторам 64 – 95 |
| E000E18Ch | ICER3 | 0000h | Регистр сброса разрешений прерываний по векторам 96 – 127 |
| E000E190h | ICER4 | 0000h | Регистр сброса разрешений прерываний по векторам 127–133 |
| E000E194h- E000E1FCh | – | – | Зарезервировано |
| E000E200h | ISPR0 | 0000h | Регистр ждущих прерываний по векторам 0 – 31 |
| E000E204h | ISPR1 | 0000h | Регистр ждущих прерываний по векторам 32 – 63 |
| E000E208h | ISPR2 | 0000h | Регистр ждущих прерываний по векторам 64 – 95 |
| E000E20Ch | ISPR3 | 0000h | Регистр ждущих прерываний по векторам 96 – 127 |
| E000E210h | ISPR4 | 0000h | Регистр ждущих прерываний по векторам 127 – 133 |
| E000E214h- E000E27Ch | – | – | Зарезервировано |
| E000E280h | ICPR0 | 0000h | Регистр запрета ждущих прерываний по векторам 0 – 31 |
| E000E284h | ICPR1 | 0000h | Регистр запрета ждущих прерываний по векторам 32 – 63 |
| E000E288h | ICPR2 | 0000h | Регистр запрета ждущих прерываний по векторам 64 – 95 |
| E000E28Ch | ICPR3 | 0000h | Регистр запрета ждущих прерываний по векторам 96 – 127 |
| E000E290h | ICPR4 | 0000h | Регистр запрета ждущих прерываний по векторам 127 – 133 |
| E000E294h- E000E2FCh | – | – | Зарезервировано |
| E000E300h | IABR0 | 0000h | Регистр флагов прерываний по векторам 0 – 31 |
| E000E304h | IABR1 | 0000h | Регистр флагов прерываний по векторам 32 – 63 |
| E000E308h | IABR2 | 0000h | Регистр флагов прерываний по векторам 64 – 95 |
| E000E30Ch | IABR3 | 0000h | Регистр флагов прерываний по векторам 96 – 127 |
| E000E310h | IABR4 | 0000h | Регистр флагов прерываний по векторам 127 – 133 |
| E000E314h- E000E3FCh | – | – | Зарезервировано |
| E000E400h | IPR00 | 0000h | Регистр приоритетов векторов прерываний 0 – 3 |
| E000E404h | IPR01 | 0000h | Регистр приоритетов векторов прерываний 4 – 7 |
| E000E408h | IPR02 | 0000h | Регистр приоритетов векторов прерываний 8 – 11 |
| E000E40Ch | IPR03 | 0000h | Регистр приоритетов векторов прерываний 12 – 15 |
| E000E410h | IPR04 | 0000h | Регистр приоритетов векторов прерываний 16 – 19 |
| E000E414h | IPR05 | 0000h | Регистр приоритетов векторов прерываний 20 – 23 |
| E000E418h | IPR06 | 0000h | Регистр приоритетов векторов прерываний 24 – 27 |
| E000E41Ch | IPR07 | 0000h | Регистр приоритетов векторов прерываний 28 – 31 |
| E000E420h | IPR08 | 0000h | Регистр приоритетов векторов прерываний 32 – 35 |
| E000E424h | IPR09 | 0000h | Регистр приоритетов векторов прерываний 36 – 39 |
| E000E428h | IPR10 | 0000h | Регистр приоритетов векторов прерываний 40 – 43 |
| E000E42Ch | IPR11 | 0000h | Регистр приоритетов векторов прерываний 44 – 47 |
| E000E430h | IPR12 | 0000h | Регистр приоритетов векторов прерываний 48 – 51 |
| E000E434h | IPR13 | 0000h | Регистр приоритетов векторов прерываний 52 – 55 |
| E000E438h | IPR14 | 0000h | Регистр приоритетов векторов прерываний 56 – 59 |
| E000E43Ch | IPR15 | 0000h | Регистр приоритетов векторов прерываний 60 – 63 |
| E000E440h | IPR16 | 0000h | Регистр приоритетов векторов прерываний 64 – 67 |
| E000E444h | IPR17 | 0000h | Регистр приоритетов векторов прерываний 68 – 71 |
| E000E448h | IPR18 | 0000h | Регистр приоритетов векторов прерываний 72 – 75 |
| E000E44Ch | IPR19 | 0000h | Регистр приоритетов векторов прерываний 76 – 79 |

Окончание таблицы Б.34

| Адрес | Мнемоника | Сброс | Название |
|-----------|-----------|-------|---|
| E000E450h | IPR20 | 0000h | Регистр приоритетов векторов прерываний 80 – 83 |
| E000E454h | IPR21 | 0000h | Регистр приоритетов векторов прерываний 84 – 87 |
| E000E458h | IPR22 | 0000h | Регистр приоритетов векторов прерываний 88 – 91 |
| E000E45Ch | IPR23 | 0000h | Регистр приоритетов векторов прерываний 92 – 95 |
| E000E460h | IPR24 | 0000h | Регистр приоритетов векторов прерываний 96 – 99 |
| E000E464h | IPR25 | 0000h | Регистр приоритетов векторов прерываний 100 – 103 |
| E000E468h | IPR26 | 0000h | Регистр приоритетов векторов прерываний 104 – 107 |
| E000E46Ch | IPR27 | 0000h | Регистр приоритетов векторов прерываний 108 – 111 |
| E000E470h | IPR28 | 0000h | Регистр приоритетов векторов прерываний 112 – 115 |
| E000E474h | IPR29 | 0000h | Регистр приоритетов векторов прерываний 116 – 119 |
| E000E478h | IPR30 | 0000h | Регистр приоритетов векторов прерываний 120 – 123 |
| E000E47Ch | IPR31 | 0000h | Регистр приоритетов векторов прерываний 124 – 127 |
| E000E480h | IPR32 | 0000h | Регистр приоритетов векторов прерываний 128 – 131 |
| E000E484h | IPR33 | 0000h | Регистр приоритетов векторов прерываний 132 и 133 |

Таблица Б.35 – Регистры ядра

| Адреса | Описание |
|---|---------------------------|
| E0000000h-E000E0FCh, E000E488h-FFFFFFFFh | Зарезервированная область |

Приложение В (обязательное) Коды состояний функционирования блока I2C

В таблицах В.1 – В.11 представлена информация о соответствии кодов и операций.

Условные обозначения, принятые в таблицах:

- [ADR, 0], [ADR, 1] – 8-разрядное значение, состоящее из 7-разрядного адреса ADR и бита направления передачи R/W#, значение которого «0» или «1» указывается непосредственно;

- DAT – байт данных;

- код мастера – 8-разрядное значение 0000_1xxx_b, где «xxx» – уникальный код каждого мастера в системе нескольких устройств;

- «с ACK» – выражение, обозначающее, что после передачи адреса/байта в ответ на запрос подтверждения передачи (бит ACK) передатчик получает подтверждение передачи от ведомого (квитирование);

- «с NACK» – выражение, обозначающее, что после передачи адреса/байта в ответ на запрос подтверждения передачи (бит ACK) передатчик получает неподтверждение передачи от ведомого (неквитирование);

- X – бит может быть установленным (1b) или сброшенным (0b), в зависимости от режима работы, состояния и дальнейших действий модуля I2C.

Таблица В.1 – Исключительные состояния

| Код | Описание состояния | Регистр SDA | Биты регистра CTLO | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|--------------------|-------------|--------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 00h | IDLE | – | – | – | – | – | Ожидать завершения текущей передачи байта |
| 1Fh | Ошибка на шине | – | 1 | 0 | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |

Таблица В.2 – Режим FS мастера передатчика (дополнительно см. таблицу В.4)

| Код | Описание состояния | Регистр SDA | Биты регистра CTLO | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|--------------------|-------------|--------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 01h | Старт | Код мастера | 1 | 0 | 0 | 0 | Передать код мастера и перейти в режим HS (0Ch/ 21h) |
| | | [ADR, 0] | | | | | Передать адрес ведомого (04h/ 05h) |
| 02h | Повторный Старт | [ADR, 0] | 1 | 0 | 0 | 0 | Передать адрес ведомого (04h/ 05h) |
| | | [ADR, 1] | | | | | Передать адрес ведомого, после чего перейти в режим приемника (08h/ 09h) |

Окончание таблицы В.2

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----|--|-----|---|---|---|---|--|
| 03h | Потеря арбитража, мастер перешел в режим безадресного ведомого | – | 1 | 0 | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |
| 04h | Отправлен адрес ведомого с ACK | DAT | 1 | 0 | 0 | 0 | Передать байт данных (06h/ 07h) |
| | | – | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |
| 05h | Отправлен адрес ведомого с NACK | – | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |
| 06h | Отправлен байт данных с ACK | DAT | 1 | 0 | 0 | 0 | Передать байт данных (06h/ 07h) |
| | | – | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |
| 07h | Отправлен байт данных с NACK | – | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |

Таблица В.3 – Режим FS мастера приемника

| Код | Описание состояния | Регистр SDA | Биты регистра CTL0 | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|---------------------------------|-------------|--------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 08h | Отправлен адрес ведомого с ACK | – | 1 | 0 | 0 | 0 | Получить байт данных, квитировать прием (0Ah) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитировать прием (0Bh) |
| 09h | Отправлен адрес ведомого с NACK | – | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |
| 0Ah | Принят байт данных и квитирован | DAT | 1 | 0 | 0 | 0 | Получить байт данных, квитировать прием (0Ah) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитировать прием (0Bh) |

Окончание таблицы В.3

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----|-------------------------------------|-----|---|---|---|---|--|
| 0Bh | Принят байт данных и не квити-рован | DAT | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |

Таблица В.4 – Режим FS мастера передатчика (дополнительно см. таблицу В.2)

| Код | Описание состояния | Регистр SDA | Биты регистра CTRL0 | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|--|-------------|---------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 0Ch | Отправлен код мастера, обнаружена ошибка (ACK) | - | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |

Таблица В.5 – Режим FS ведомого приемника (дополнительно см. таблицу В.7)

| Код | Описание состояния | Регистр SDA | Биты регистра CTRL0 | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|---|-------------|---------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 10h | Принят адрес и квити-рован | - | 1 | 0 | 0 | 0 | Получить байт данных, квитиловать прием (12h) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитиловать прием (13h) |
| 11h | Принят адрес после потери арбитража и квити-рован | - | 1 | 0 | 0 | 0 | Получить байт данных, квитиловать прием (12h) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитиловать прием (13h) |
| 12h | Принят байт данных и квити-рован | DAT | 1 | 0 | 0 | 0 | Получить байт данных, квитиловать прием (12h) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитиловать прием (13h) |
| 13h | Принят байт данных и не квити-рован | DAT | 1 | 0 | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |
| | | | 1 | 0 | 0 | 1 | Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h) |

Таблица В.6 – Режим FS ведомого передатчика

| Код | Описание состояния | Регистр SDA | Биты регистра CTL0 | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|---|-------------|--------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 14h | Принят адрес и квитирован | DAT | 1 | X | 0 | 0 | Передать байт данных, квитировать/ не квитировать (16h/17h) |
| 15h | Принят адрес после потери арбитража и квитирован | DAT | 1 | X | 0 | 0 | Передать байт данных, квитировать/ не квитировать (16h/17h) |
| 16h | Отправлен байт данных с ACK | DAT | 1 | X | 0 | 0 | Передать байт данных, квитировать/ не квитировать (16h/17h) |
| 17h | Отправлен байт данных с NACK | – | 1 | X | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |
| | | | 1 | X | 0 | 1 | Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h) |
| 18h | Принят адрес отклика и квитирован | DAT | 1 | X | 0 | 0 | Передать байт данных, квитировать/ не квитировать (1Ah/1Bh) |
| 19h | Принят адрес отклика после потери арбитража и квитирован | DAT | 1 | X | 0 | 0 | Передать байт данных, квитировать/ не квитировать (1Ah/1Bh) |
| 1Ah | Отправлен байт данных в ответ на получение адреса отклика с ACK | DAT | 1 | X | 0 | 0 | Передать байт данных, квитировать/ не квитировать (1Ah/1Bh) |

Окончание таблицы В.6

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----|--|---|---|---|---|---|--|
| 1Bh | Отправлен байт данных в ответ на получение адреса отклика с NACK | — | 1 | X | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |
| | | | 1 | X | 0 | 1 | Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h) |

Таблица В.7 – Режим FS ведомого приемника (дополнительно см. таблицу В.5)

| Код | Описание состояния | Регистр SDA | Биты регистра CTL0 | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|--|-------------|--------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 1Ch | Стоп | — | 1 | 0 | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |
| | | | 1 | 0 | 0 | 1 | Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h) |
| 1Dh | Принят адрес общего вызова и квитирован | — | 1 | 0 | 0 | 0 | Получить байт данных, квитировать прием (12h) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитировать прием (13h) |
| 1Eh | Принят адрес общего вызова после потери арбитража и квитирован | — | 1 | 0 | 0 | 0 | Получить байт данных, квитировать прием (12h) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитировать прием (13h) |

Таблица В.8 – Режим HS мастера передатчика

| Код | Описание состояния | Регистр SDA | Биты регистра CTLO | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|---|-------------|--------------------|-----|------|-------|---|
| | | | CLRST | ACK | STOP | START | |
| 21h | Успешно отправлен код мастера, мастер перешел в режим HS | – | 1 | 0 | 0 | 1 | Сделать повторный старт (22h) |
| 22h | Повторный старт | [ADR, 0] | 1 | 0 | 0 | 0 | Передать адрес ведомого (28h/29h) |
| | | [ADR, 1] | | | | | Передать адрес ведомого, после квитирования/не квитирования переключиться в режим мастера приемника (28h/29h) |
| 23h | Потеря арбитража, мастер перешел в режим HS безадресного ведомого | – | 1 | 0 | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |
| 24h | Отправлен адрес ведомого с ACK | DAT | 1 | 0 | 0 | 0 | Передать байт данных (26h/27h) |
| | | – | 1 | 0 | 0 | 1 | Сделать повторный старт (22h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |
| 25h | Отправлен адрес ведомого с NACK | – | 1 | 0 | 0 | 1 | Сделать повторный старт (22h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |
| 26h | Отправлен байт данных с ACK | DAT | 1 | 0 | 0 | 0 | Передать байт данных (26h/27h) |
| | | – | 1 | 0 | 0 | 1 | Сделать повторный старт (22h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |
| 27h | Отправлен байт данных с NACK | – | 1 | 0 | 0 | 1 | Сделать повторный старт (22h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |

Таблица В.9 – Режим HS мастера приемника

| Код | Описание состояния | Регистр SDA | Биты регистра CTL0 | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|------------------------------------|-------------|--------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 28h | Отправлен адрес ведомого с ACK | – | 1 | 0 | 0 | 0 | Получить байт данных, квитировать прием (2Ah) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитировать прием (2Bh) |
| 29h | Отправлен адрес ведомого с NACK | – | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |
| 2Ah | Принят байт данных и квитирован | DAT | 1 | 0 | 0 | 0 | Получить байт данных, квитировать прием (2Ah) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитировать прием (2Bh) |
| 2Bh | Принят байт данных и не квитирован | DAT | 1 | 0 | 0 | 1 | Сделать повторный старт (02h) |
| | | | 1 | 0 | 1 | 0 | Остановить передачу (00h) |
| | | | 1 | 0 | 1 | 1 | Остановить передачу, а затем сделать повторный старт (01h) |

Таблица В.10 – Режим HS ведомого приемника

| Код | Описание состояния | Регистр SDA | Биты регистра CTL0 | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|------------------------------------|-------------|--------------------|-----|------|-------|--|
| | | | CLRST | ACK | STOP | START | |
| 30h | Принят адрес и квитирован | – | 1 | 0 | 0 | 0 | Получить байт данных, квитировать прием (32h) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитировать прием (33h) |
| 32h | Принят байт данных и квитирован | DAT | 1 | 0 | 0 | 0 | Получить байт данных, квитировать прием (32h) |
| | | | 1 | 1 | 0 | 0 | Получить байт данных, не квитировать прием (33h) |
| 33h | Принят байт данных и не квитирован | DAT | 1 | 0 | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |
| | | | 1 | 0 | 0 | 1 | Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h) |

Таблица В.11 – Режим HS ведомого передатчика

| Код | Описание состояния | Регистр SDA | Биты регистра CTRL0 | | | | Возможные дальнейшие действия и коды результатов их выполнения |
|-----|------------------------------|-------------|---------------------|-----|------|--|--|
| | | | CLRST | ACK | STOP | START | |
| 34h | Принят адрес и квити-рован | DAT | 1 | X | 0 | 0 | Передать байт данных, квитиловать/ не квитиловать (36h/37h) |
| 36h | Отправлен байт данных с ACK | DAT | 1 | X | 0 | 0 | Передать байт данных, квитиловать/ не квитиловать (36h/37h) |
| 37h | Отправлен байт данных с NACK | - | 1 | X | 0 | 0 | Функционировать в режиме безадресного ведомого (00h) |
| | 1 | | X | 0 | 1 | Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h) | |

Приложение Г (обязательное) Регистры прерываний

Для управления прерываниями используются пять групп регистров $ISER_i$, $ICER_i$, $ISPR_i$, $ICPR_i$ и $IABR_i$, где индекс i – 0, 1, 2, 3, 4 (см. таблицу Г.1). Группы имеют идентичную структуру. Набор прерываний, которыми управляет регистр группы, зависит от индекса. На рисунке Г.1 показана одна группа регистров и указано соответствие номеров векторов прерываний и бит регистров. Управление прерыванием осуществляется записью единицы или нуля в соответствующий бит. Допускается одновременное управление несколькими прерываниями.

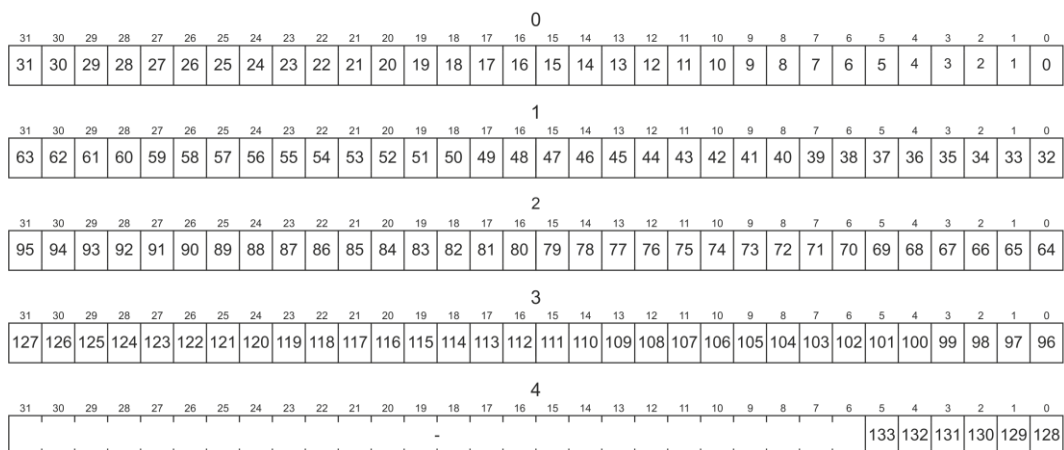


Рисунок Г.1 – Соответствие векторов прерываний и бит управляющих регистров

Таблица Г.1 – 32-разрядные регистры управления прерываниями

| Мнемоника и назначение | Операция над битом | | Влияние на соответствующее прерывание |
|---|--------------------|--------|---|
| | Чтение | Запись | |
| ISER_i Регистр разрешения прерываний от источников | Чтение | 0 | Прерывание запрещено |
| | | 1 | Прерывание разрешено |
| | Запись | 0 | Нет влияния |
| | | 1 | Разрешение прерывания |
| ICER_i Регистр сброса разрешения прерываний от источников | Чтение | 0 | Прерывание запрещено |
| | | 1 | Прерывание разрешено |
| | Запись | 0 | Нет влияния |
| | | 1 | Запрет прерывания |
| ISPR_i Регистр ждущих прерываний | Чтение | 0 | Ждущего прерывания нет |
| | | 1 | Есть ждущее прерывание |
| | Запись | 0 | Нет влияния |
| | | 1 | Установка ждущего прерывания (программное прерывание) |
| ICPR_i Регистр сброса ждущих прерываний | Чтение | 0 | Ждущего прерывания нет |
| | | 1 | Есть ждущее прерывание |
| | Запись | 0 | Нет влияния |
| | | 1 | Сброс ждущего прерывания |
| IABR_i Регистр флагов прерываний | Чтение | 0 | Флага прерывания нет |
| | | 1 | Флаг прерывания установлен. Сбрасывается аппаратно по окончании обслуживания прерывания |

Для задания приоритетов прерываний используются регистры IPR_i , где i индекс от 0 до 33. В таблице Г.2 представлен формат регистра IPR_i .

Таблица Г.2 – Регистр приоритетов

| | | | | | | | | | | | | | | | | |
|--|-------|----|--|----|----|----|----|--------|----|----|----|-----------------|----|----|----|--|
| IPRi | | | | | | | | | | | | Сброс: 0000000h | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| PRIn+3 | | | | | | | | PRIn+2 | | | | | | | | |
| 3 ч | | | | | | | | 3 ч | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| PRIn+1 | | | | | | | | PRIn | | | | | | | | |
| 3 ч | | | | | | | | 3 ч | | | | | | | | |
| Поле | Бит | | Описание | | | | | | | | | | | | | |
| PRIn+3 | 31-24 | | Поле задания приоритета (n+3)-ого вектора прерывания | | | | | | | | | | | | | |
| PRIn+2 | 23-16 | | Поле задания приоритета (n+2)-ого вектора прерывания | | | | | | | | | | | | | |
| PRIn+1 | 15-8 | | Поле задания приоритета (n+1)-ого вектора прерывания | | | | | | | | | | | | | |
| PRIn | 7-0 | | Поле задания приоритета n-ого вектора прерывания $n = 4 \times i$. | | | | | | | | | | | | | |
| <p>Примечание – Запись в PRIn[4:0] игнорируется, читаются – нули. Диапазон допустимых значений для битов PRIn[7:5] – 0-7. Подробнее о том, как приоритеты разделяются на группы и подгруппы в таблице Г.3.</p> | | | | | | | | | | | | | | | | |

Последний в группе регистр приоритетов IPR33 (для векторов прерываний 132 и 133) имеет формат, показанный на рисунке Г.2.

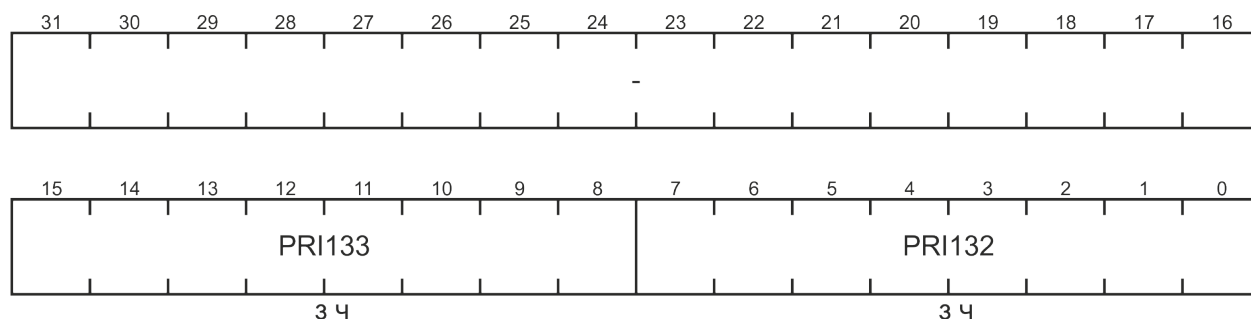


Рисунок Г.2 – Формат регистра IPR33 (адрес E000E484h)

Таблица Г.3 – Распределение приоритетов на группы и подгруппы

| Поле PRIGROUP регистра AIRCR ядра | PRIn[7:5] | | | Количество | |
|-----------------------------------|-------------------------------------|-------------|----------------|------------|-----------|
| | Распределение на группы и подгруппы | Биты группы | Биты подгруппы | Группы | Подгруппы |
| 0-4 | 0bxxx | [7:5] | Отсутствуют | 8 | 0 |
| 5 | 0bxx.y | [7:6] | [5] | 4 | 2 |
| 6 | 0bx.yy | [7] | [6:5] | 2 | 4 |
| 7 | 0byyy | Отсутствуют | [7:5] | 0 | 8 |

Приложение Д (справочное) Пример схемы включения микроконтроллера

На рисунке Д.1 показан пример типовой схемы подключения микроконтроллера. После сброса микроконтроллера выводы В0, В1, В2, Е1, Е10 конфигурируются как выводы отладочного модуля JTAG, а выводы D11 и Е0 – выводы приемопередатчика UART0.

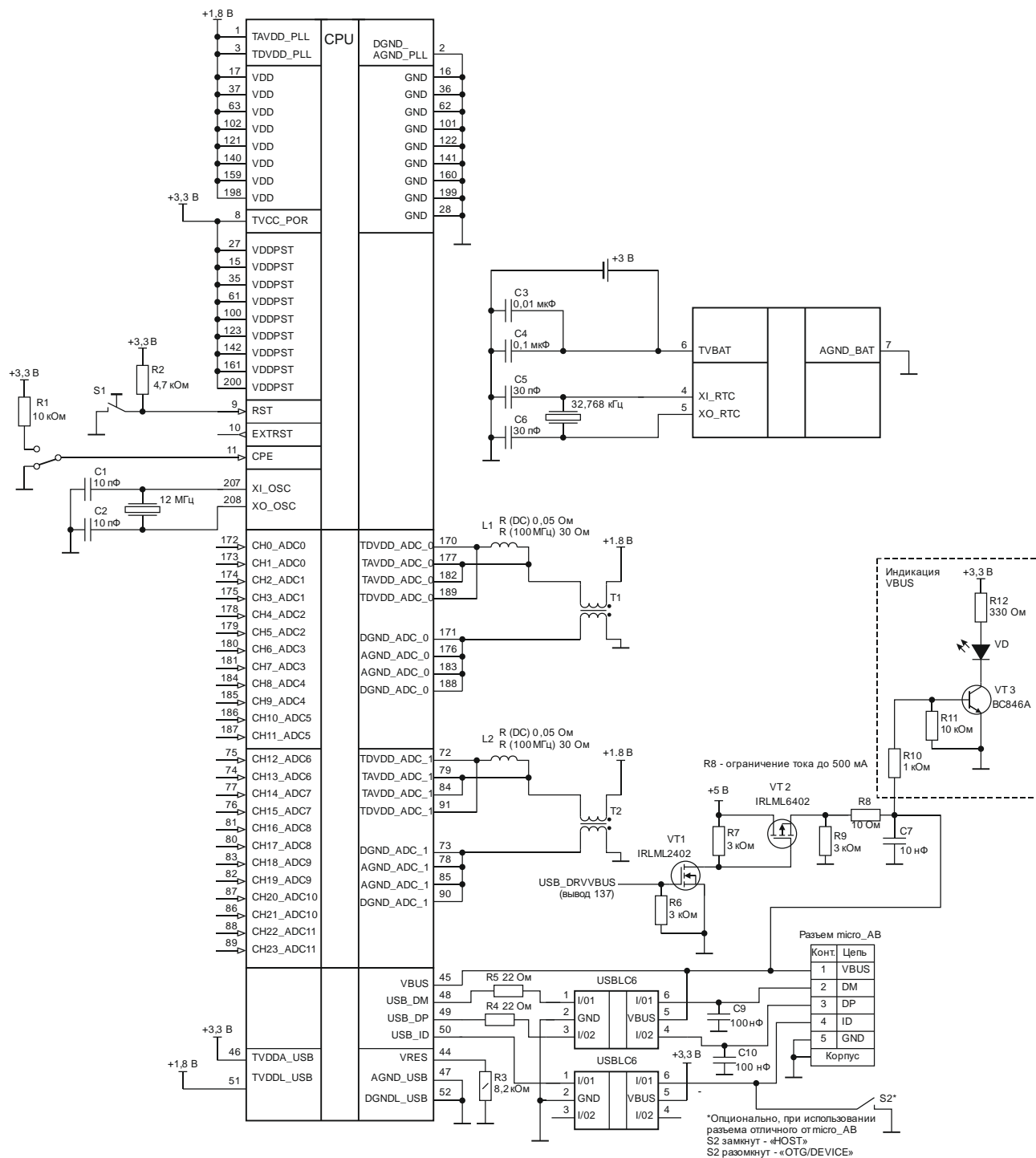


Рисунок Д.1, лист 1 – Пример типовой схемы включения МК K1921BK01T

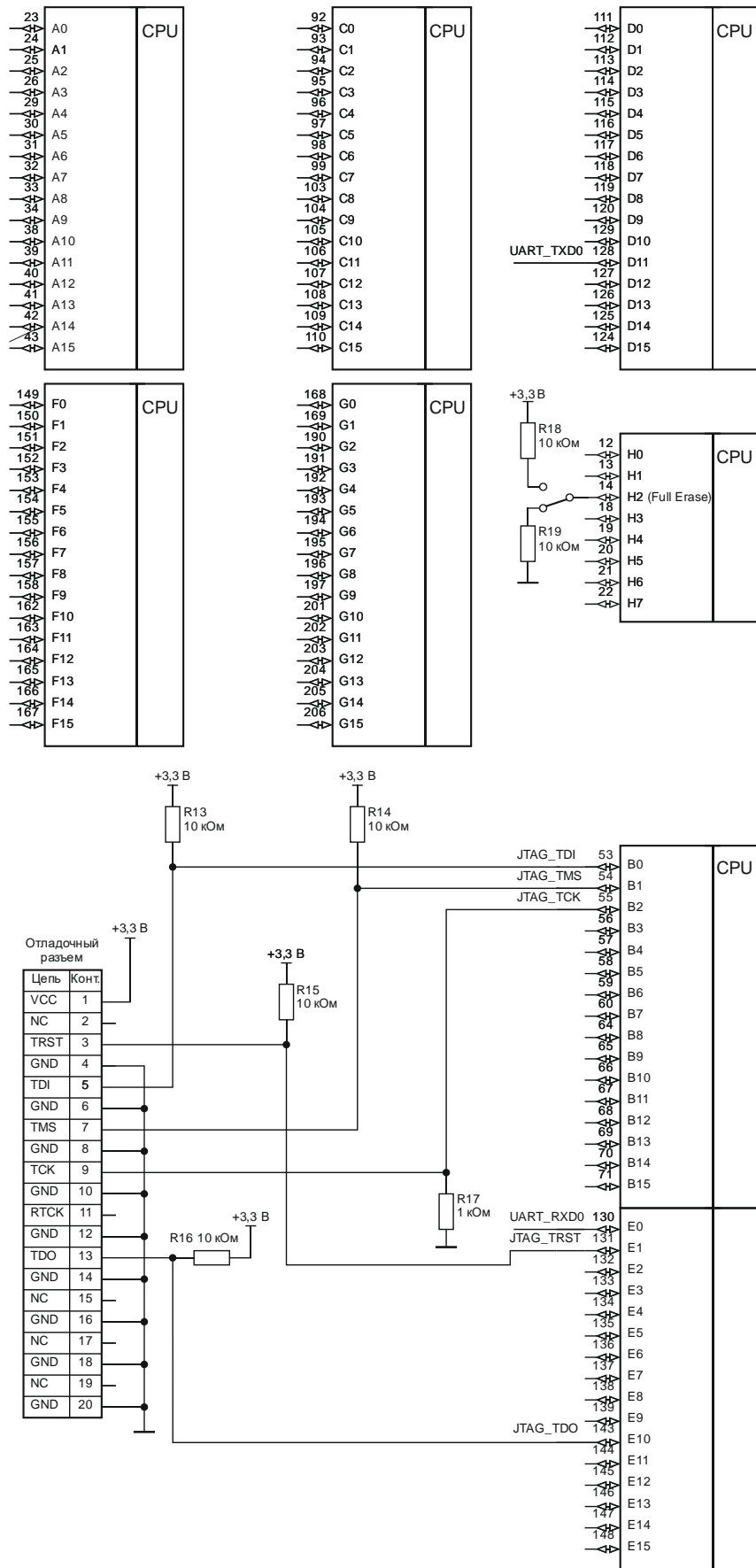


Рисунок Д.1, лист 2

Лист регистрации изменений

| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в докум. | № докум. | Подп. | Дата |
|------|-------------------------|---|-------|----------------|---------------------------------|------------|-------|----------|
| | измененных | замененных | новых | аннулированных | | | | |
| - | - | - | все | - | 451 | КФДЛ.9022 | | 21.01.16 |
| 1 | 1 | - | - | - | - | КФДЛ.9022 | | 21.01.16 |
| 2 | - | 25 | - | - | - | КФДЛ.9037 | | 20.02.16 |
| 3 | - | 25, 208, 369 | - | - | - | КФДЛ.9068 | | 25.03.16 |
| 4 | - | 93, 94, 114, 155, 449 | - | - | - | КФДЛ.9081 | | 08.04.16 |
| 5 | - | 24,25,362,363, 364,375 | - | - | - | КФДЛ.9096 | | 05.05.16 |
| 6 | - | 30,39,40,83,97,98, 110,113,117,125, 126,128,129,165, 170,192,193,194, 219,249,250,276, 311,346,347,352, 358,361,369,401 | - | - | - | КФДЛ.9231 | | 12.10.16 |
| 7 | - | 75, 76, 238, 416, 417 | - | - | - | КФДЛ.9357 | | 22.03.17 |
| 8 | - | 2, 24, 25, 75-79, 82-84, 93, 94, 99-104, 236-240, 346, 416, 417 | - | - | - | КФДЛ.9422 | | 22.06.17 |
| 9 | - | 15, 23, 73, 74, 81-83, 244, 245, 352, 353 | - | - | - | КФДЛ.9443 | | 09.08.17 |
| 10 | - | 203, 204 | - | - | - | КФДЛ.9491 | | 01.11.17 |
| 11 | - | 43, 383, 384, 399, 400 | - | - | - | КФДЛ.9511 | | 13.11.17 |
| 12 | - | 13, 75, 76, 83-85, 97, 107, 108, 112, 126, 184, 192, 204, 207, 217, 224, 225, 249, 254, 358-360, 362, 449, 450 | - | - | - | КФДЛ.9852 | | 01.03.19 |
| 13 | - | 30, 73, 74, 77, 86, 234, 238, 313, 448 | - | - | - | КФДЛ.10003 | | 10.09.19 |
| | | | | | | | | |
| | | | | | | | | |