

PIC16F62X

Однокристальные 8-разрядные
FLASH CMOS микроконтроллеры
компании Microchip Technology Incorporated

Перевод основывается на технической документации DS40300B
компании Microchip Technology Incorporated, USA.

© ООО «Микро-Чип»
Москва - 2001

Распространяется бесплатно.
Полное или частичное воспроизведение материала допускается только с письменного разрешения
ООО «Микро-Чип»
тел. (095) 737-7545
www.microchip.ru

PIC16F62X

FLASH-Based 8-Bit CMOS Microcontrollers

Trademarks: The Microchip name, logo, PIC, PICmicro, PICSTART, PICMASTER and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. FlexROM, MPLAB and fuzzy-LAB are trademarks and SQTP is a service mark of Micro-chip in the U.S.A.

All other trademarks mentioned herein are the property of their respective companies.

Микроконтроллеры, описываемые в этом документе:

- PIC16F627
- PIC16F628

Характеристика высокопроизводительного RISC ядра:

- 35 инструкций
- Все команды выполняются за один цикл (200нс), кроме инструкций переходов, выполняющихся за два цикла
- Тактовая частота:
 - DC – 20МГц, тактовый сигнал
 - DC – 200нс, один машинный цикл

| Устройство | Память | | |
|------------|------------------|--------------|---------------|
| | Программ (FLASH) | Данных (ОЗУ) | EEPROM данных |
| PIC16F627 | 1024x14 | 224x8 | 128x8 |
| PIC16F628 | 2048x14 | 224x8 | 128x8 |

- Система прерываний
- 16 специальных аппаратных регистров
- 8-уровневый аппаратный стек
- Прямой, косвенный и относительный режим адресации

Характеристика периферийных модулей:

- 15 независимых портов ввода/вывода
- Повышенная нагрузочная способность портов ввода/вывода
- Модуль аналоговых компараторов:
 - два аналоговых компаратора
 - внутренний программируемый источник опорного напряжения (V_{REF})
 - Внешний выход компараторов
- Таймер 0: 8-разрядный таймер/счетчик с 8-разрядным программируемым предделителем
- Таймер 1: 16-разрядный таймер/счетчик с возможностью подключения внешнего пьезоэлектрического резонатора
- Таймер 2: 8-разрядный таймер/счетчик с 8-разрядным программируемым предделителем и выходным делителем
- Модуль сравнения/захвата/ШИМ (CCP):
 - 16-разрядный захват (максимальная разрешающая способность 12.5нс)
 - 16-разрядное сравнение (максимальная разрешающая способность 200нс)
 - 10-разрядный ШИМ
- Последовательный синхронно-асинхронный приемопередатчик USART
- 16 байт памяти данных, доступных из всех банков.

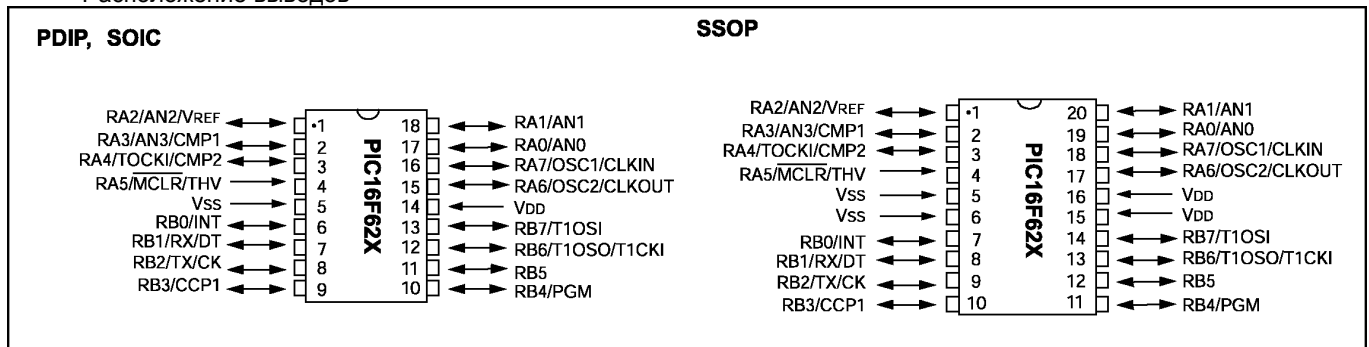
Специальные функции микроконтроллеров:

- Сброс по включению питания (POR)
- Таймер сброса (PWRT) и таймер ожидания запуска генератора (OST) после включения питания
- Детектор пониженного напряжения (BOD)
- Сторожевой таймер WDT с собственным RC генератором
- Мультиплицируемый вывод -MCLR
- Программируемые подтягивающие резисторы на входах PORTB
- Программируемая защита памяти программ
- Низковольтный режим программирования
- Режим энергосбережения SLEEP
- Выбор параметров тактового генератора:
 - FLASH биты выбора параметров тактового генератора
 - ER генератор (внешний резистор)
 - Уменьшение числа внешних компонентов
 - Двухскоростной внутренний RC генератор
 - Работа в режиме малого потребления
 - ЕС внешний тактовый сигнал
 - XT режим генератора
 - HS режим генератора
 - LP режим генератора
- Программирование в готовом устройстве (используется два вывода микроконтроллера)
- Программируемые пользователем биты ID

CMOS технология:

- Высокоскоростная, энергосберегающая CMOS FLASH технология (0.7 микрон)
- Полностью статическая архитектура
- Широкий диапазон напряжений питания:
 - PIC16F627 от 3.0 до 5.5В
 - PIC16F628 от 3.0 до 5.5В
 - PIC16LF627 от 2.0 до 5.5В
 - PIC16LF628 от 2.0 до 5.5В
- Коммерческий, промышленный и расширенный температурные диапазоны
- Малое энергопотребление:
 - < 2мА @ 5.0В, 4.0МГц
 - 15мкА @ 3.0В, 32кГц
 - < 1мкА в режиме энергосбережения @ 3.0В

Расположение выводов



Содержание

| | |
|---|-----------|
| 1.0 Введение | 8 |
| 1.1 Поддержка разработчиков | 8 |
| 2.0 Обзор семейства микроконтроллеров PIC16F62X..... | 9 |
| 2.1 FLASH микроконтроллеры | 9 |
| 2.2 Микроконтроллеры, программируемые производителем (QTP)..... | 9 |
| 2.3 Серийный выпуск продукции (SQTP SM) | 9 |
| 3.0 Обзор архитектуры..... | 10 |
| 3.1 Синхронизация выполнения команд..... | 13 |
| 3.2 Конвейерная выборка и выполнение команд..... | 13 |
| 4.0 Организация памяти | 14 |
| 4.1 Организация памяти программ | 14 |
| 4.2 Организация памяти данных | 14 |
| 4.2.1 Регистры общего назначения..... | 14 |
| 4.2.2 Регистры специального назначения | 16 |
| 4.3 Регистры PCLATH и PCL..... | 25 |
| 4.3.1 Вычисляемый переход | 25 |
| 4.3.2 Стек..... | 25 |
| 4.4 Косвенная адресация, регистры INDF и FSR..... | 26 |
| 5.0 Порты ввода/вывода | 27 |
| 5.1 Регистры PORTA и TRISA..... | 27 |
| 5.2 Регистры PORTB и TRISB..... | 33 |
| 5.3 Программирование портов ввода/вывода..... | 43 |
| 5.3.1 Двунаправленные порты ввода/вывода | 43 |
| 5.3.2 Последовательность операций с портами ввода/вывода | 43 |
| 6.0 Модуль таймера TMR0 | 44 |
| 6.1 Прерывания от TMR0 | 45 |
| 6.2 Использование внешнего источника тактового сигнала для TMR0 | 46 |
| 6.2.1 Синхронизация внешнего сигнала..... | 46 |
| 6.2.2 Задержка приращеня TMR0..... | 46 |
| 6.3 Предделитель | 46 |
| 6.3.1 Переключение предделителя..... | 47 |
| 7.0 Модуль таймера TMR1 | 48 |
| 7.1 Работа TMR1 с внутренним тактовым сигналом | 49 |
| 7.2 Работа TMR1 с внешним синхронизированным тактовым сигналом | 49 |
| 7.2.1 Синхронизация внешнего тактового сигнала | 49 |
| 7.3 Работа TMR1 с внешним не синхронизированным тактовым сигналом | 50 |
| 7.3.1 Параметры внешнего не синхронизированного тактового сигнала | 50 |
| 7.3.2 Чтение/запись TMR0 в асинхронном режиме | 50 |
| 7.4 Генератор TMR1 | 50 |
| 7.5 Сброс TMR1 триггером модуля CCP | 51 |
| 7.6 Сброс регистров TMR1 (TMR1H, TMR1L)..... | 51 |
| 7.7 Предделитель TMR1 | 51 |
| 8.0 Модуль таймера TMR2 | 52 |
| 8.1 Предделитель и выходной делитель TMR2..... | 52 |
| 8.2 Сигнал TMR2..... | 52 |

| | |
|--|-----------|
| 9.0 Модуль компараторов | 54 |
| 9.1 Настройка модуля компараторов | 56 |
| 9.2 Работа модуля компараторов | 56 |
| 9.3 Опорное напряжение для компараторов | 57 |
| 9.3.1 Внешний источник опорного напряжения | 57 |
| 9.3.2 Внутренний источник опорного напряжения | 57 |
| 9.4 Время реакции компараторов | 57 |
| 9.5 Прерывания от компараторов | 57 |
| 9.6 Выходы компараторов | 58 |
| 9.7 Работа модуля компараторов в режиме SLEEP | 59 |
| 9.8 Эффект сброса | 59 |
| 9.9 Подключение к аналоговым входам | 59 |
| 10.0 ССР модуль (захват/сравнение/ШИМ) | 60 |
| 10.1 Режим захвата | 61 |
| 10.1.1 Настройка вывода модуля ССР | 61 |
| 10.1.2 Настройка таймера TMR1 | 61 |
| 10.1.3 Обработка прерываний | 61 |
| 10.1.4 Предварительный счетчик событий модуля ССР | 61 |
| 10.2 Режим сравнения | 62 |
| 10.2.1 Настройка вывода модуля ССР | 62 |
| 10.2.2 Настройка таймера TMR1 | 62 |
| 10.2.3 Обработка прерываний | 62 |
| 10.2.4 Триггер специального события | 62 |
| 10.3 ШИМ режим | 63 |
| 10.3.1 Период ШИМ | 63 |
| 10.3.2 Сквозность ШИМ | 64 |
| 10.3.3 Последовательность настройки модуля ССР в ШИМ режиме | 64 |
| 11.0 Источник опорного напряжения | 65 |
| 11.1 Настройка источника опорного напряжения | 65 |
| 11.2 Точность источника опорного напряжения | 66 |
| 11.3 Функционирование в SLEEP режиме | 66 |
| 11.4 Эффект сброса | 66 |
| 11.5 Подключение к источнику опорного напряжения | 66 |
| 12.0 Универсальный синхронно-асинхронный приемопередатчик (USART) | 67 |
| 12.1 Генератор частоты обмена USART BRG | 69 |
| 12.1.1 Выборка | 73 |
| 12.2 Асинхронный режим USART | 74 |
| 12.2.1 Асинхронный передатчик USART | 74 |
| 12.2.2 Асинхронный приемник USART | 76 |
| 12.3 Функции USART | 78 |
| 12.3.1 Приемник с детектором 9-разрядного адреса | 78 |
| 12.4 Синхронный ведущий режим USART | 79 |
| 12.4.1 Передача синхронного ведущего | 79 |
| 12.4.2 Прием синхронного ведущего | 81 |
| 12.5 Синхронный ведомый режим USART | 82 |
| 12.5.1 Передача синхронного ведомого | 82 |
| 12.5.2 Прием синхронного ведомого | 83 |
| 13.0 EEPROM память данных | 84 |
| 13.1 Регистр EEADR | 84 |
| 13.2 Регистры EECON1, EECON2 | 84 |
| 13.3 Чтение из EEPROM памяти данных | 85 |
| 13.4 Запись в EEPROM память данных | 86 |
| 13.5 Проверка записи | 86 |
| 13.6 Защита от случайной записи в EEPROM память данных | 86 |
| 13.7 Операции с EEPROM памятью при установленном бите защиты | 87 |

| | |
|---|------------|
| 14.0 Особенности микроконтроллеров PIC16F62X | 88 |
| 14.1 Биты конфигурации | 88 |
| 14.2 Настройка тактового генератора..... | 90 |
| 14.2.1 Режимы тактового генератора..... | 90 |
| 14.2.2 Кварцевый/керамический резонатор..... | 90 |
| 14.2.3 Внешний тактовый генератор | 91 |
| 14.2.4 Внешний тактовый сигнал..... | 91 |
| 14.2.5 ER генератор | 92 |
| 14.2.6 Внутренний RC генератор 4МГц..... | 92 |
| 14.2.7 CLKOUT | 92 |
| 14.3 Программный выбор частоты тактового генератора..... | 92 |
| 14.4 Сброс | 93 |
| 14.5 POR, PWRT, OST, BOD | 94 |
| 14.5.1 Сброс по включению питания POR..... | 94 |
| 14.5.2 Таймер включения питания PWRT | 94 |
| 14.5.3 Таймер запуска генератора OST | 94 |
| 14.5.4 Детектор пониженного напряжения питания BOD | 94 |
| 14.5.5 Последовательность удержания микроконтроллера в состоянии сброса | 95 |
| 14.5.6 Регистр PCON..... | 95 |
| 14.6 Прерывания..... | 99 |
| 14.6.1 Внешнее прерывание с входа RB0/INT..... | 99 |
| 14.6.2 Прерывание по переполнению TMR0 | 100 |
| 14.6.3 Прерывание по изменению уровня сигнала на входах RB7:RB4..... | 100 |
| 14.7 Сохранение контекста при обработке прерываний..... | 101 |
| 14.8 сторожевой таймер WDT | 101 |
| 14.8.1 Период WDT | 101 |
| 14.8.2 Рекомендации по работе с WDT | 101 |
| 14.9 Режим энергосбережения SLEEP | 103 |
| 14.9.1 Выход из режима SLEEP | 103 |
| 14.10 Защита кода программы | 104 |
| 14.11 Размещение идентификатора ID | 104 |
| 14.12 Внутрисхемное программирование ICSP | 104 |
| 14.13 Режим низковольтного программирования..... | 104 |
| 15.0 Система команд | 105 |
| 15.1 Подробное описание команд..... | 107 |
| 16.0 Поддержка разработчиков | 124 |
| 16.1 Интегрированная среда проектирования MPLAB-IDE | 124 |
| 16.2 Ассемблер MPASM..... | 125 |
| 16.3 С компиляторы MPLAB-C17 и MPLAB-C18 | 125 |
| 16.4 Линкер MPLINK/ Организатор библиотек MPLIB | 125 |
| 16.5 Программный симулятор MPLAB-SIM | 125 |
| 16.6 Универсальный эмулятор MPLAB-ICE..... | 125 |
| 16.7 PICMASTER/PICMASTER-CE | 126 |
| 16.8 ICEPIC..... | 126 |
| 16.9 Внутрисхемный отладчик MPLAB-ICD | 126 |
| 16.10 Универсальный программатор PRO MATE II | 126 |
| 16.11 Программатор PICSTART | 126 |
| 16.12 Аппаратный модуль SIMICE | 126 |
| 16.13 Демонстрационная плата PICDEM-1 | 126 |
| 16.14 Демонстрационная плата PICDEM-2 для PIC16CXXX | 127 |
| 16.15 Демонстрационная плата PICDEM-3 для PIC16CXXX..... | 127 |
| 16.16 Демонстрационная плата PICDEM-17 | 127 |
| 16.17 SEEVAL (с функциями программатора) | 127 |
| 16.18 KeeLoq (с функциями программатора)..... | 127 |

| | |
|---|------------|
| 17.0 Электрические характеристики | 129 |
| 17.1 Электрические характеристики: PIC16F62X-04, PIC16F62X-20 | 131 |
| 17.2 Электрические характеристики: PIC16LF62X-04, PIC16LF62X-20 | 132 |
| 17.3 Электрические характеристики: PIC16F62X, PIC16LF62X..... | 133 |
| 17.4 Символьное обозначение временных параметров..... | 135 |
| 17.5 Временные диаграммы и спецификации | 136 |
| 18.0 Характеристики микроконтроллеров | 142 |
| 19.0 Корпуса микроконтроллеров | 143 |
| 19.1 Описание обозначений на корпусах микроконтроллеров..... | 143 |
| 19.2 Правила идентификации типа микроконтроллеров PIC16F62X | 147 |

1.0 Введение

18-выводные FLASH микроконтроллеры PIC16F62X входят в состав распространенного семейства PICmicro PIC16CXX. Микроконтроллеры этого семейства имеют 8-разрядную, высокопроизводительную и полностью статическую RISC архитектуру.

PIC16F62X имеют 8-уровневый аппаратный стек и большое количество внутренних и внешних прерываний. В гарвардской архитектуре RISC ядра микроконтроллера разделены 14-разрядная память программ и 8-разрядная память данных. Такой подход позволяет выполнять все инструкции за один машинный цикл, кроме команд ветвления, которые выполняются за два машинных цикла. Ядро микроконтроллеров поддерживает 35 простых в изучении, но очень эффективных инструкций. Дополнительные регистры управления и архитектурные новшества позволяют создавать высокоэффективные устройства.

По сравнению с 8-разрядными микроконтроллерами этого класса, при использовании PIC16F62X выигрыш в эффективности использования памяти программ достигает 2:1, а в производительности 4:1.

Специальные особенности микроконтроллеров PIC16F62X позволяют сократить число внешних компонентов, что в свою очередь снижает стоимость конечного устройства, повышает надежность системы и уменьшает энергопотребление. Дополнительную гибкость в разработках дает широкий выбор режимов работы тактового генератора: ER генератор, наиболее дешевое решение; LP генератор, минимизирует потребляемый ток; XT генератор, для подключения стандартного резонатора; INTRC внутренний RC генератор; HS генератор, для высокоскоростных режимов работы.

Энергосберегающий режим SLEEP, позволяет эффективно использовать микроконтроллеры в устройствах с питанием от батареек или аккумуляторов. Выход из режима SLEEP происходит при возникновении внешних, некоторых внутренних прерываниях и сбросе микроконтроллера. Высоконадежный сторожевой таймер WDT с собственным внутренним RC генератором предотвращает «зависание» программы.

В таблице 1-1 сведены основные характеристики микроконтроллеров PIC16F62X.

На рисунке 3-1 представлена структурная схема микроконтроллеров PIC16F62X.

Микроконтроллеры PIC16F62X удовлетворяют ряду параметров для их использования от зарядных устройств до удаленных датчиков с малым потреблением электроэнергии. FLASH технология и большое количество периферийных модулей, совместимых с предыдущими микроконтроллерами, позволяют быстро и удобно разрабатывать программное обеспечение. Высокая производительность, малая стоимость, легкость в использовании и гибкость портов ввода/вывода – делают PIC16F62X универсальными микроконтроллерами.

1.1 Поддержка разработчиков

Микроконтроллеры PIC16F62X обеспечены полнофункциональным макроассемблером, программным симулятором, эмуляторами, недорогими программаторами. Сторонними организациями разработаны C компиляторы для микроконтроллеров этого семейства.

Таблица 1-1 Основные характеристики микроконтроллеров PIC16F62X

| | | PIC16F627 | PIC16F628 | PIC16LF627 | PIC16LF628 |
|--------------------------------------|--|---|---|---|---|
| Быстродействие | Максимальная тактовая частота (МГц) | 20 | 20 | 20 | 20 |
| | Память | | | | |
| | Flash память программ (слов) | 1024 | 2048 | 1024 | 2048 |
| | Память данных (байт) | 224 | 224 | 224 | 224 |
| | EEPROM память данных (байт) | 128 | 128 | 128 | 128 |
| Периферия | Таймеры | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 |
| | Компараторов | 2 | 2 | 2 | 2 |
| | Модулей CCP | 1 | 1 | 1 | 1 |
| | Последовательный интерфейс | USART | USART | USART | USART |
| | Программируемый источник опорного напряжения | Есть | Есть | Есть | Есть |
| Дополнительные характеристики | Число источников прерываний | 10 | 10 | 10 | 10 |
| | Число портов ввода/вывода | 16 | 16 | 16 | 16 |
| | Напряжение питания (В) | 3.0 – 5.5 | 3.0 – 5.5 | 2.0 – 5.5 | 2.0 – 5.5 |
| | Детектор пониженного напряжения питания | Есть | Есть | Есть | Есть |
| | Корпус | 18-выводный DIP, SOIC 20-выводный SSOP | 18-выводный DIP, SOIC 20-выводный SSOP | 18-выводный DIP, SOIC 20-выводный SSOP | 18-выводный DIP, SOIC 20-выводный SSOP |

В состав всех микроконтроллеров PICmicro входит: сброс по включению питания PWRT, сторожевой таймер WDT, программируемая защита кода программы и высокая нагрузочная способность портов ввода/вывода. Микроконтроллеры PIC16F62X программируются в последовательном режиме с использованием двух выводов RB6 и RB7.

2.0 Обзор семейства микроконтроллеров PIC16F62X

Тип корпуса, температурный диапазон и максимальная тактовая частота – параметры, которые входят в состав полного наименования микроконтроллеров. В зависимости от того, в каком режиме будет работать микроконтроллер, необходимо правильно выбрать его тип. Описание правил обозначения микроконтроллеров PIC16F62X смотрите в разделе «Правила идентификации типа микроконтроллеров PIC16F62X».

2.1 FLASH микроконтроллеры

Данный тип микроконтроллеров чаще всего используется при разработке и отладке программного обеспечения, поскольку их память программ может быть стерта и повторно запрограммирована.

Стереть и перепрограммировать микроконтроллеры можно непосредственно в устройстве пользователя, в программаторах PICSTART, PRO MATE II и других.

2.2 Микроконтроллеры, программируемые производителем (QTP)

Компания Microchip предоставляет возможность заказать запрограммированные микроконтроллеры заранее предоставленным кодом. Данный сервис следует использовать при средних и больших объемах закупок микроконтроллеров и отработанном программном обеспечении. Поставляемые микроконтроллеры полностью соответствуют параметрам стандартных FLASH микроконтроллеров, за исключением того, что код программы и биты конфигурации были записаны на заводе изготовителе. Прежде чем микроконтроллеры будут поставлены заказчику, они пройдут серию испытаний на заводе изготовителе. Для получения дополнительной информации обратитесь к региональному представителю Microchip.

2.3 Серийный выпуск продукции (SQTP SM)

Компания Microchip предоставляет оригинальное средство программирования, в котором пользователь может определить место размещения уникального серийного номера генерируемого случайным, псевдослучайным и последовательным методом. Запрограммированный уникальный серийный номер может служить: кодом доступа, паролем или идентификационным номером устройства.

3.0 Обзор архитектуры

Высокая эффективность микроконтроллеров PIC16F62X достигается за счет архитектуры ядра, подобная архитектура обычно используемой в RISC микропроцессорах. В PIC16F62X используется Гарвардская архитектура с отдельными шинами доступа к памяти программ и памяти данных, в отличие от традиционных систем, в которых обращение к памяти программ и данных выполняется по одной шине.

Разделение памяти программ и памяти данных позволяет использовать не 8-разрядные команды или кратные разрядности шины данных. Все команды микроконтроллера 14-разрядные однословные. По 14-разрядной шине доступа к памяти программ выполняется выборка кода за один машинный цикл. Непрерывная работа ядра микроконтроллера по выборке и выполнению кодов программы дает возможность выполнять все команды за один машинный цикл (200нс @ 20МГц), кроме команд ветвления. Ядро микроконтроллеров поддерживает 35 высокоэффективных команд.

В таблице представлен объем FLASH памяти программ, памяти данных (ОЗУ) и EEPROM памяти данных.

| Микроконтроллер | Память | | |
|-----------------|------------------|--------------|---------------|
| | Программ (FLASH) | Данных (ОЗУ) | EEPROM данных |
| PIC16F627 | 1024x14 | 224x8 | 128x8 |
| PIC16F628 | 2048x14 | 224x8 | 128x8 |
| PIC16LF627 | 1024x14 | 224x8 | 128x8 |
| PIC16LF628 | 2048x14 | 224x8 | 128x8 |

В PIC16F62X адресовать память данных можно непосредственно или косвенно. Все регистры специального назначения отображаются в памяти данных, включая счетчик программ. PIC16F62X имеет ортогональную систему команд, что дает возможность выполнить любую операцию с любым регистром памяти данных, используя любой метод адресации. Это облегчает написание программ для микроконтроллеров PIC16F62X и снижает общее время разработки устройства.

Микроконтроллеры PIC16F62X содержат 8-разрядное АЛУ (арифметико-логическое устройство) с одним рабочим регистром W. АЛУ выполняет арифметические и булевы операции между рабочим регистром и любым регистром памяти данных. Основными операциями АЛУ являются: сложение, вычитание, сдвиг и логические операции. В командах с двумя операндами один операнд всегда рабочий регистр W, а второй операнд регистр памяти данных или константа. В командах с одним операндом используется регистр W или регистр памяти данных.

Используемый в операциях 8-разрядный рабочий регистр W не отображается на память данных.

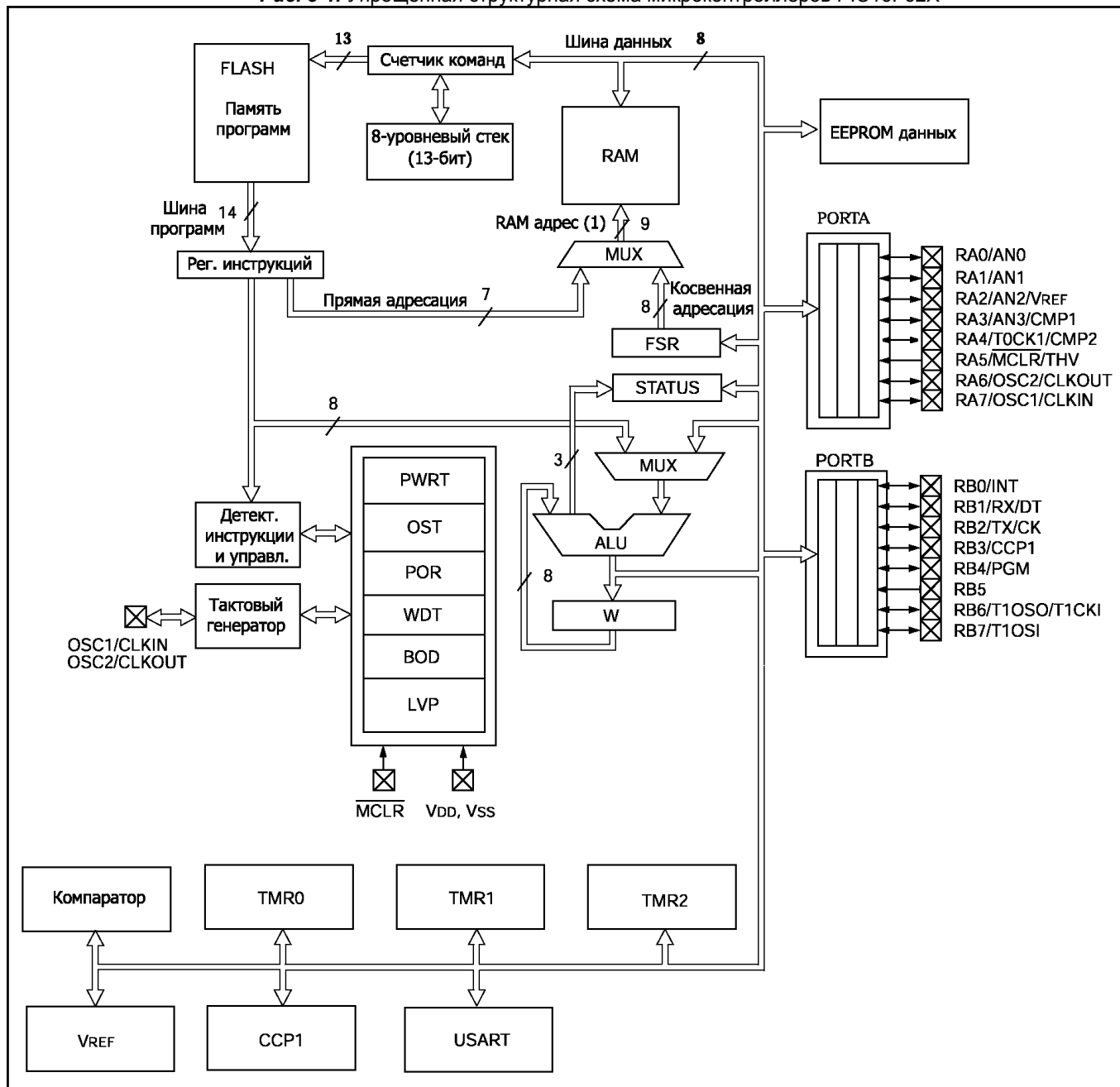
В зависимости от выполняемой команды АЛУ может влиять на следующие флаги в регистре STATUS: флаг переноса C, флаг полупереноса DC, флаг нуля Z. Флаги C и DC выполняют роль соответствующих битов заема при выполнении команд вычитания SUBLW и SUBWF.

Упрощенная блок схема микроконтроллеров PIC16F62X показана на рисунке 3-1. Назначение выводов микроконтроллеров сведено в таблицу 3-1.

В микроконтроллерах PIC16F62X существует два типа памяти данных:

- энергонезависимая EEPROM память данных, предусмотрена для хранения калибровочной информации, таблиц или любой другой информации, требующей периодического изменения. Данные, записанные в EEPROM память, не будут потеряны при отключении питания микроконтроллера;
- регулярная память данных (ОЗУ), используется для хранения временной информации во время выполнения программы. Информация в регулярной памяти данных будет потеряна при выключении питания микроконтроллера.

Рис. 3-1. Упрощенная структурная схема микроконтроллеров PIC16F62X



Примечание 1. Старшие биты регистра STATUS.

Таблица 3-1 Назначение выводов микроконтроллеров PIC16F62X

| Обозначение вывода | № вывода DIP, SOIC | № вывода SSOP | Тип I/O/P | Тип буфера | Описание |
|--------------------------|--------------------|---------------|-----------|-----------------------|--|
| RA0/AN0 | 17 | 19 | I/O | ST | Двунаправленный порт ввода/вывода, аналоговый вход компаратора |
| RA1/AN1 | 18 | 20 | I/O | ST | Двунаправленный порт ввода/вывода, аналоговый вход компаратора |
| RA2/AN2/V _{REF} | 1 | 1 | I/O | ST | Двунаправленный порт ввода/вывода, аналоговый вход компаратора, выход источника опорного напряжения V _{REF} |
| RA3/AN3/CPM1 | 2 | 2 | I/O | ST | Двунаправленный порт ввода/вывода, аналоговый вход компаратора, выход компаратора |
| RA4/T0CKI/CPM2 | 3 | 3 | I/O | ST | Двунаправленный порт ввода/вывода, может использоваться как T0CKI, выход компаратора |
| RA5/-MCLR/THV | 4 | 4 | I | ST | Вход сброса микроконтроллера, вход напряжения программирования. Когда вывод настроен как -MCLR, то по низкому уровню сигнала производится сброс микроконтроллера. При нормальной работе напряжение на -MCLR/THV не должно превышать VDD. |
| RA6/OSC2/CLKOUT | 15 | 17 | I/O | ST | Двунаправленный порт ввода/вывода, выход генератора для подключения резонатора. В режиме ER генератора на выходе CLKOUT формируется сигнал с частотой 1/4 OSC1, обозначая циклы команд |
| RA7/OSC1/CLKIN | 16 | 18 | I/O | ST | Двунаправленный порт ввода/вывода, вход генератора, вход внешнего тактового сигнала, вывод ER смещения |
| RB0/INT | 6 | 7 | I/O | TTL/ST ⁽¹⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вход внешнего прерывания |
| RB1/RX/DT | 7 | 8 | I/O | TTL/ST ⁽³⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вход приемника USART, линия данных в синхронном режиме USART |
| RB2/TX/CK | 8 | 9 | I/O | TTL/ST ⁽³⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, выход передатчика USART, линия тактового сигнала в синхронном режиме |
| RB3/CCP1 | 9 | 10 | I/O | TTL/ST ⁽⁴⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вывод модуля CCP |
| RB4/PGM | 10 | 11 | I/O | TTL/ST ⁽⁵⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP. Когда разрешено низковольтное программирование, запрещены прерывания по изменению сигнала на входе, а подтягивающий резистор отключен |
| RB5 | 11 | 12 | I/O | TTL | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP |
| RB6/T1OSO/T1CKI | 12 | 13 | I/O | TTL/ST ⁽²⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP. Выход генератора таймера 1 |
| RB7/T1OSI | 13 | 14 | I/O | TTL/ST ⁽²⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP. Вход генератора таймера 1 |
| VSS | 5 | 5,6 | P | - | Общий вывод |
| VDD | 14 | 15,16 | P | - | Положительное напряжение питания |

Обозначения: I – вход, O – выход, I/O – вход/выход, P – питание, - - не используется,
TTL – вход/выход TTL, ST – вход с триггером Шмидта.

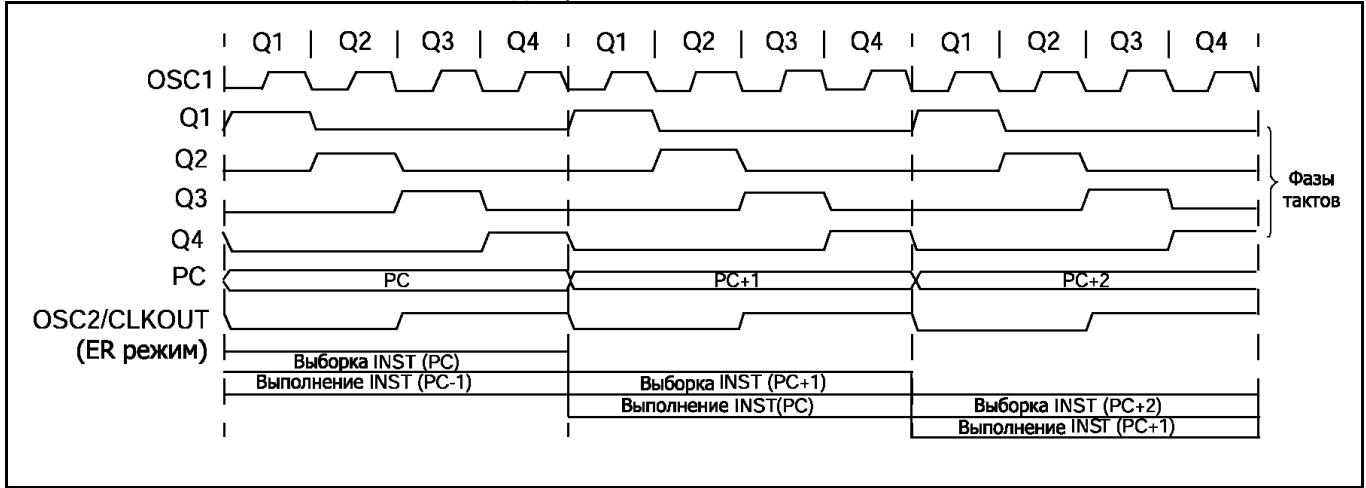
Примечания:

1. Входной буфер с триггером Шмидта при использовании внешних прерываний.
2. Входной буфер с триггером Шмидта при работе в режиме последовательного программирования.
3. Входной буфер с триггером Шмидта при использовании выводов модулем USART.
4. Входной буфер с триггером Шмидта при использовании выводов модулем CCP.
5. Входной буфер с триггером Шмидта при низковольтном программировании.

3.1 Синхронизация выполнения команд

Входной тактовый сигнал (вывод OSC1/CLKIN/RA7) внутренней схемой микроконтроллера разделяется на четыре последовательных неперекрывающихся такта Q1, Q2, Q3 и Q4. Внутренний счетчик команд (PC) увеличивается на единицу в каждом такте Q1, а выборка команды из памяти программ происходит на каждом такте Q4. Декодирование и выполнение команды происходит с такта Q1 по Q4. На рисунке 3-2 показаны циклы выполнения команд.

Рис. 3-2 Диаграмма циклов выполнения команд



3.2 Конвейерная выборка и выполнение команд

Цикл выполнения команды состоит из четырех тактов Q1, Q2, Q3 и Q4. Выборка следующей команды и выполнение текущей совмещены по времени, таким образом, выполнение команды происходит за один цикл. Если команда изменяет счетчик команд PC (команды ветвления, например GOTO), то необходимо два машинных цикла для выполнения команды (рисунок 3-3).

Цикл выборки команды начинается с приращения счетчика команд PC в такте Q1.

В цикле выполнения команды, код загруженной команды, помещается в регистр команд IR на такте Q1. Декодирование и выполнение команды происходит в тактах Q2, Q3 и Q4. Операнд из памяти данных читается в такте Q2, а результат выполнения команды записывается в такте Q4.

Рис. 3-3 Выборка и выполнения команд



4.0 Организация памяти

4.1 Организация памяти программ

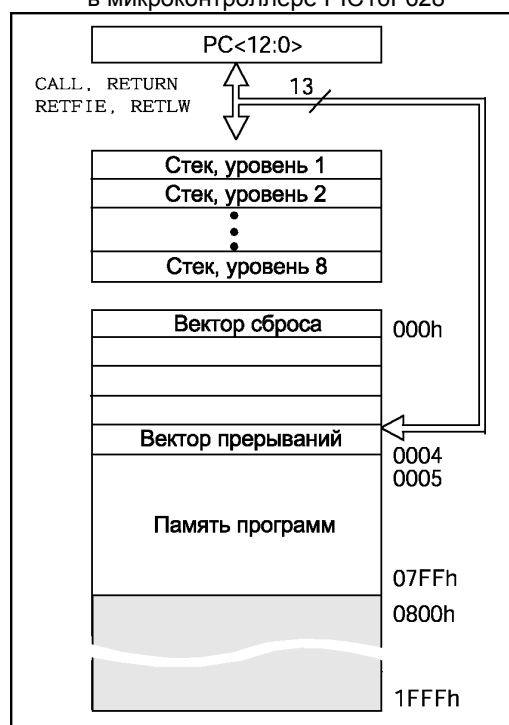
Микроконтроллеры PIC16F62X имеют 13-разрядный счетчик команд PC, способный адресовать 8К x 14 слов памяти программ. Физически реализовано 1К x 14 (0000h – 03FFh) для PIC16F627 и 2К x 14 (0000h – 7FFFh) для PIC16F628. Обращение к физически не реализованной памяти программ приведет к адресации реализованной памяти в пределах 1К x 14 для PIC16F627 и 2К x 14 для PIC16F628.

Адрес вектора сброса – 0000h. Адрес вектора прерываний – 0004h.

Рис. 4-1 Организация памяти в микроконтроллере PIC16F627



Рис. 4-2 Организация памяти в микроконтроллере PIC16F628



4.2 Организация памяти данных

Память данных (см. рисунок 4-3) разделена на четыре банка, которые содержат регистры общего и специального (SFR) назначения. Регистры специальных функций расположены в первых 32 байтах каждого банка памяти данных. Регистры общего назначения имеют адреса 20h-7Fh, A0h-FFh, 120h-14Fh, 170h-17Fh и 1F0h-1FFh реализованные как статическое ОЗУ.

В таблице показано состояние управляющих битов при обращении к банкам памяти данных.

| | RP1 | RP0 |
|--------|-----|-----|
| Банк 0 | 0 | 0 |
| Банк 1 | 0 | 1 |
| Банк 2 | 1 | 0 |
| Банк 3 | 1 | 1 |

Регистры с адресами F0h-FFh, 170h-17Fh и 1F0h-1FFh реализованы как статическое ОЗУ с отображением на регистры 70h-7Fh.

4.2.1 Регистры общего назначения

В микроконтроллерах PIC16F62X организация регистров общего назначения 224x8, обратиться к ним можно непосредственно или косвенно, используя регистр FSR (см. раздел 4.4).

4.2.2 Регистры специального назначения

С помощью регистров специального назначения выполняется управление функциями ядра и периферийными модулями микроконтроллера. Регистры специального назначения реализованы как статическое ОЗУ.

В этом разделе будут описаны регистры, управляющие функциями ядра микроконтроллера. Описание регистров периферийных модулей смотрите в соответствующем разделе документации.

Таблица 4-1 Регистры специального назначения, банк 0

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы ⁽¹⁾ |
|--------|---------|--|---------|---------|---------------------------------|---------|--------|---------|---------|-----------|------------------------------|
| Банк 0 | | | | | | | | | | | |
| 00h | INDF | Обращение к регистру, адрес которого записан в FSR (не физический регистр) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 01h | TMR0 | Регистр таймера 0 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h | PCL | Младше биты счетчика команд PC | | | | | | | | 0000 0000 | 0000 0000 |
| 03h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx | 000q q000 |
| 04h | FSR | Регистр адреса при косвенной адресации | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xxxx 0000 | xxxx 0000 |
| 06h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu |
| 07h | НЕТ | | | | | | | | | - | - |
| 08h | НЕТ | | | | | | | | | - | - |
| 09h | НЕТ | | | | | | | | | - | - |
| 0Ah | PCLATH | - | - | - | Старшие биты счетчика команд PC | | | | | ---0 0000 | ---0 0000 |
| 0Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 0Dh | НЕТ | | | | | | | | | - | - |
| 0Eh | TMR1L | Младший байт 16-разрядного таймера 1 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Старший байт 16-разрядного таймера 1 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | - | - | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 11h | TMR2 | Регистр таймера 2 | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | - | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -uuu uuuu |
| 13h | НЕТ | | | | | | | | | - | - |
| 14h | НЕТ | | | | | | | | | - | - |
| 15h | CCPR1L | Младший байт захвата/сравнения/ШИМ | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Старший байт захвата/сравнения/ШИМ | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | - | - | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG | Регистр данных передатчика USART | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | Регистр данных приемника USART | | | | | | | | 0000 0000 | 0000 0000 |
| 1Bh | НЕТ | | | | | | | | | - | - |
| 1Ch | НЕТ | | | | | | | | | - | - |
| 1Dh | НЕТ | | | | | | | | | - | - |
| 1Eh | НЕТ | | | | | | | | | - | - |
| 1Fh | CMCON | C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 0000 0000 | 0000 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание 1. Другие типы сбросов микроконтроллера – по низкому уровню сигнала на выводе -MCLR, по срабатыванию детектора пониженного напряжения (BOD), по срабатыванию сторожевого таймера WDT.

Таблица 4-2 Регистры специального назначения, банк 1

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы ⁽¹⁾ | |
|--------|--------|--|-----------------------|--------|---------------------------------|--------|--------|--------|--------|-----------|------------------------------|--------|
| Банк 1 | | | | | | | | | | | | |
| 80h | INDF | Обращение к регистру, адрес которого записан в FSR (не физический регистр) | | | | | | | | xxxx xxxx | xxxx xxxx | |
| 81h | OPTION | -RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 | |
| 82h | PCL | Младше биты счетчика команд PC | | | | | | | | 0000 0000 | 0000 0000 | |
| 83h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx | 000q q000 | |
| 84h | FSR | Регистр адреса при косвенной адресации | | | | | | | | xxxx xxxx | uuuu uuuu | |
| 85h | TRISA | TRISA7 | TRISA6 | - | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 11-1 1111 | 11-1 1111 | |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 1111 1111 | |
| 87h | HET | | | | | | | | | - | - | |
| 88h | HET | | | | | | | | | - | - | |
| 89h | HET | | | | | | | | | - | - | |
| 8Ah | PCLATH | - | - | - | Старшие биты счетчика команд PC | | | | --- | 0 0000 | --- | 0 0000 |
| 8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u | |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 | |
| 8Dh | HET | | | | | | | | | - | - | |
| 8Eh | PCON | - | - | - | - | OSCF | - | -POR | -BOD | ---- 1-0X | ---- 1-uq | |
| 8Fh | HET | | | | | | | | | - | - | |
| 90h | HET | | | | | | | | | - | - | |
| 91h | HET | | | | | | | | | - | - | |
| 92h | PR2 | Регистр периода таймера 2 | | | | | | | | 1111 1111 | 1111 1111 | |
| 93h | HET | | | | | | | | | - | - | |
| 94h | HET | | | | | | | | | - | - | |
| 95h | HET | | | | | | | | | - | - | |
| 96h | HET | | | | | | | | | - | - | |
| 97h | HET | | | | | | | | | - | - | |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 | |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 | 0000 0000 | |
| 9Ah | EEDATA | Регистр данных EEPROM | | | | | | | | xxxx xxxx | uuuu uuuu | |
| 9Bh | EEADR | - | Регистр адреса EEPROM | | | | | | | -xxx xxxx | -uuu uuuu | |
| 9Ch | EECON1 | - | - | - | - | WRERR | WREN | WR | RD | ---- x000 | ---- q000 | |
| 9Dh | EECON2 | Управляющий регистр 2 EEPROM (физически не реализован) | | | | | | | | ---- ---- | ---- ---- | |
| 9Eh | HET | | | | | | | | | - | - | |
| 9Fh | VRCON | VREN | VROE | VRR | - | VR3 | VR2 | VR1 | VR0 | 000- 0000 | 000- 0000 | |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание 1. Другие типы сбросов микроконтроллера – по низкому уровню сигнала на выводе -MCLR, по срабатыванию детектора пониженного напряжения (BOD), по срабатыванию сторожевого таймера WDT.

Таблица 4-3 Регистры специального назначения, банк 2

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы ⁽¹⁾ |
|--------|--------|--|-------|-------|---------------------------------|-------|-------|-------|-------|-----------|------------------------------|
| Банк 2 | | | | | | | | | | | |
| 100h | INDF | Обращение к регистру, адрес которого записан в FSR (не физический регистр) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 101h | TMR0 | Регистр таймера 0 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 102h | PCL | Младше биты счетчика команд PC | | | | | | | | 0000 0000 | 0000 0000 |
| 103h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx | 000q q000 |
| 104h | FSR | Регистр адреса при косвенной адресации | | | | | | | | xxxx xxxx | uuuu uuuu |
| 105h | НЕТ | | | | | | | | | - | - |
| 106h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu |
| 107h | НЕТ | | | | | | | | | - | - |
| 108h | НЕТ | | | | | | | | | - | - |
| 109h | НЕТ | | | | | | | | | - | - |
| 10Ah | PCLATH | - | - | - | Старшие биты счетчика команд PC | | | | | ---0 0000 | ---0 0000 |
| 10Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 10Ch | НЕТ | | | | | | | | | - | - |
| | НЕТ | | | | | | | | | - | - |
| 11Fh | НЕТ | | | | | | | | | - | - |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание 1. Другие типы сбросов микроконтроллера – по низкому уровню сигнала на выводе -MCLR, по срабатыванию детектора пониженного напряжения (BOD), по срабатыванию сторожевого таймера WDT.

Таблица 4-4 Регистры специального назначения, банк 3

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы ⁽¹⁾ |
|--------|--------|--|--------|--------|---------------------------------|--------|--------|--------|--------|-----------|------------------------------|
| Банк 3 | | | | | | | | | | | |
| 180h | INDF | Обращение к регистру, адрес которого записан в FSR (не физический регистр) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 181h | OPTION | -RBPU | INTEDG | TOCS | TOSE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 182h | PCL | Младше биты счетчика команд PC | | | | | | | | 0000 0000 | 0000 0000 |
| 183h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx | 000q q000 |
| 184h | FSR | Регистр адреса при косвенной адресации | | | | | | | | xxxx xxxx | uuuu uuuu |
| 185h | НЕТ | | | | | | | | | - | - |
| 186h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 1111 1111 |
| 187h | НЕТ | | | | | | | | | - | - |
| 188h | НЕТ | | | | | | | | | - | - |
| 189h | НЕТ | | | | | | | | | - | - |
| 18Ah | PCLATH | - | - | - | Старшие биты счетчика команд PC | | | | | ---0 0000 | ---0 0000 |
| 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 18Ch | НЕТ | | | | | | | | | - | - |
| | НЕТ | | | | | | | | | - | - |
| 19Fh | НЕТ | | | | | | | | | - | - |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание 1. Другие типы сбросов микроконтроллера – по низкому уровню сигнала на выводе -MCLR, по срабатыванию детектора пониженного напряжения (BOD), по срабатыванию сторожевого таймера WDT.

4.2.2.1 Регистр STATUS

В регистре STATUS содержатся флаги состояния АЛУ, флаги причины сброса микроконтроллера и биты управления банками памяти данных.

Регистр STATUS может быть адресован любой командой, как и любой другой регистр памяти данных. Если обращение к регистру STATUS выполняется командой, которая воздействует на флаги Z, DC и C, то изменение этих трех битов командой заблокирована. Эти биты сбрасываются или устанавливаются согласно логике ядра микроконтроллера. Команды изменения регистра STATUS так же не воздействуют на биты -TO и -PD. Поэтому результат выполнения команды с регистром STATUS может отличаться от ожидаемого. Например, команда CLRWF STATUS сбросит три старших бита и установит бит Z (состояние регистра STATUS после выполнения команды 000uu1uu, где u-не изменяемый бит).

При изменении битов регистра STATUS рекомендуется использовать команды, не влияющие на флаги АЛУ (SWAPF, MOVWF, BCF и BSF). Описание команд смотрите в разделе 15.0.

Примечание. Флаги C и DC используются как биты заема и десятичного заема соответственно, например, при выполнении команд вычитания SUBLW и SUBWF.

Регистр STATUS (адрес 03h, 83h, 103h или 183h)

| R/W-0 | R/W-0 | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x | |
|--|-------|-------|-----|-----|-------|-------|-------|-------|
| IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | |
| Бит 7 | | | | | | | | Бит 0 |
| <p>бит 7: IRP: Бит выбора банка при косвенной адресации 1 = банк 2, 3 (100h – 1FFh) 0 = банк 0, 1 (000h - 0FFh)</p> <p>биты 6-5: RP1:RP0: Биты выбора банка при непосредственной адресации 11 = банк 3 (180h – 1FFh) 10 = банк 2 (100h – 17Fh) 01 = банк 1 (080h – 0FFh) 00 = банк 0 (000h – 07Fh)</p> <p>бит 4: -TO: Флаг переполнения сторожевого таймера 1 = после POR или выполнения команд CLRWDT, SLEEP 0 = после переполнения WDT</p> <p>бит 3: -PD: Флаг включения питания 1 = после POR или выполнения команды CLRWDT 0 = после выполнения команды SLEEP</p> <p>бит 2: Z: Флаг нулевого результата 1 = нулевой результат выполнения арифметической или логической операции 0 = не нулевой результат выполнения арифметической или логической операции</p> <p>бит 1: DC: Флаг десятичного переноса/заема (для команд ADDWF, ADDWL, SUBWF, SUBWL), заем имеет инверсное значение 1 = был перенос из младшего полубайта 0 = не было переноса из младшего полубайта</p> <p>бит 0: C: Флаг переноса/заема (для команд ADDWF, ADDWL, SUBWF, SUBWL), заем имеет инверсное значение 1 = был перенос из старшего бита 0 = не было переноса из старшего бита</p> <p>Примечание. Флаг заема имеет инверсное значение. Вычитание выполняется путем прибавления дополнительного кода второго операнда. При выполнении команд сдвига (RRF, RLF) бит C загружается старшим или младшим битом сдвигаемого регистра.</p> | | | | | | | | |

R – чтение бита
 W – запись бита
 U – не реализовано, читается как 0
 -n – значение после POR
 -x – неизвестное значение после POR

4.2.2.2 Регистр OPTION

Регистр OPTION доступен для чтения и записи, содержит биты управления:

- предварительным делителем TMR0/WDT;
- активным фронтом внешнего прерывания RBO/INT;
- подтягивающими резисторами на входах PORTB.

Примечание. Если предварительный делитель включен перед WDT, то коэффициент деления тактового сигнала для TMR0 равен 1:1.

Регистр OPTION (адрес 81h или 181h)

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|--------------|---------------|-------------|-------------|------------|------------|------------|------------|
| -RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| Бит 7 | | | | | | | Бит 0 |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

бит 7: **-RBPU**: Включение подтягивающих резисторов на входах PORTB
1 = подтягивающие резисторы отключены
0 = подтягивающие резисторы включены

бит 6: **INTEDG**: Выбор активного фронта сигнала на входе внешнего прерывания INT
1 = прерывания по переднему фронту сигнала
0 = прерывания по заднему фронту сигнала

бит 5: **T0CS**: Выбор тактового сигнала для TMR0
1 = внешний тактовый сигнал с вывода RA4/T0CKI
0 = внутренний тактовый сигнал CLKOUT

бит 4: **T0SE**: Выбор фронта приращения TMR0 при внешнем тактовом сигнале
1 = приращение по заднему фронту сигнала (с высокого к низкому уровню) на выводе RA4/T0CKI
0 = приращение по переднему фронту сигнала (с низкого к высокому уровню) на выводе RA4/T0CKI

бит 3: **PSA**: Выбор включения предделителя
1 = предделитель включен перед WDT
0 = предделитель включен перед TMR0

биты 2-0: **PS2: PS0**: Установка коэффициента деления предделителя

| Значение | Для TMR0 | Для WDT |
|----------|----------|---------|
| 000 | 1:2 | 1:1 |
| 001 | 1:4 | 1:2 |
| 010 | 1:8 | 1:4 |
| 011 | 1:16 | 1:8 |
| 100 | 1:32 | 1:16 |
| 101 | 1:64 | 1:32 |
| 110 | 1:128 | 1:64 |
| 111 | 1:256 | 1:128 |

4.2.2.3 Регистр INTCON

Регистр INTCON доступен для чтения и записи, содержит биты разрешений и флагов некоторых источников прерываний. Дополнительно смотрите разделы 4.2.2.4 и 4.2.2.5.

Примечание. Флаги прерываний устанавливаются при возникновении условий прерываний вне зависимости от соответствующих битов разрешения и бита общего разрешения прерываний GIE (INTCON<7>).

Регистр INTCON (адрес 0Bh, 8Bh, 10Bh или 18Bh)

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-x |
|--|---|-------|-------|-------|-------|-------|-------|
| GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF |
| Бит 7 | | | | | | | Бит 0 |
| <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"> <p>R – чтение бита W – запись бита U – не реализовано, читается как 0 -n – значение после POR -x – неизвестное значение после POR</p> </div> | | | | | | | |
| бит 7: | GIE: Глобальное разрешение прерываний 1 = разрешены все немаскированные прерывания 0 = все прерывания запрещены | | | | | | |
| бит 6: | PEIE: Разрешение прерываний от периферийных модулей 1 = разрешены все немаскированные прерывания периферийных модулей 0 = прерывания от периферийных модулей запрещены | | | | | | |
| бит 5: | TOIE: Разрешение прерывания по переполнению TMR0 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | |
| бит 4: | INTE: Разрешение внешнего прерывания INT 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | |
| бит 3: | RBIE: Разрешение прерывания по изменению сигнала на входах RB7:RB4 PORTB 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | |
| бит 2: | TOIF: Флаг прерывания по переполнению TMR0 1 = произошло переполнение TMR0 (сбрасывается программно) 0 = переполнения TMR0 не было | | | | | | |
| бит 1: | INTF: Флаг внешнего прерывания INT 1 = выполнено условие внешнего прерывания на выводе RB0/INT (сбрасывается программно) 0 = внешнего прерывания не было | | | | | | |
| бит 0: | RBIF: Флаг прерывания по изменению уровня сигнала на входах RB4:RB7 PORTB 1 = зафиксировано изменение уровня сигнала на одном из входов RB7:RB4 (сбрасывается программно) 0 = не было изменения уровня сигнала ни на одном из входов RB7:RB4 | | | | | | |

2.2.2.4 Регистр PIE1

Регистр PIE1 доступен для чтения и записи, содержит биты разрешения периферийных прерываний.

Примечание. Для разрешения периферийных прерываний необходимо установить в '1' бит PEIE (INTCON<6>).

Регистр PIE1 (адрес 8Ch)

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | |
|---|--|-------|-------|-----|--------|--------|--------|-------|
| EEIE | CMIE | RCIE | TXIE | - | CCP1IE | TMR2IE | TMR1IE | |
| Бит 7 | | | | | | | | Бит 0 |
| <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"> R – чтение бита W – запись бита U – не реализовано, читается как 0 -n – значение после POR -x – неизвестное значение после POR </div> | | | | | | | | |
| бит 7: | EEIE: Разрешение прерывания по окончании записи в EEPROM данных 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | | |
| бит 6: | CMIE: Разрешение прерывания от компараторов 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | | |
| бит 5: | RCIE: Разрешение прерывания от приемника USART 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | | |
| бит 4: | TXIE: Разрешение прерывания от передатчика USART 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | | |
| бит 3: | Не реализован: читается как '0' | | | | | | | |
| бит 2: | CCP1IE: Разрешение прерывания от модуля CCP1 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | | |
| бит 1: | TMR2IE: Разрешение прерывания по переполнению TMR2 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | | |
| бит 0: | TMR1IE: Разрешение прерывания по переполнению TMR1 1 = прерывание разрешено 0 = прерывание запрещено | | | | | | | |

4.2.2.5 Регистр PIR1

Регистр PIR1 доступен для чтения и записи, содержит флаги прерываний периферийных модулей.

Примечание. Флаги прерываний устанавливаются при возникновении условий прерываний вне зависимости от соответствующих битов разрешения и бита общего разрешения прерываний GIE (INTCON<7>). Программное обеспечение пользователя должно сбрасывать соответствующие флаги при обработке прерываний от периферийных модулей.

Регистр PIR1 (адрес 0Ch)

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | |
|---|-------------|-------------|-------------|-----|---------------|---------------|---------------|-------|
| EEIF | CMIF | RCIF | TXIF | - | CCP1IF | TMR2IF | TMR1IF | |
| Бит 7 | | | | | | | | Бит 0 |
| <p>бит 7: EEIF: Флаг прерывания по окончании записи в EEPROM данных 1 = запись в EEPROM данных завершена (сбрасывается программно) 0 = запись в EEPROM данных не завершена или не была начата</p> <p>бит 6: CMIF: Флаг прерывания от компараторов 1 = изменилось состояние вывода компаратора 0 = состояние вывода компаратора не изменялось</p> <p>бит 5: RCIF: Флаг прерывания от приемника USART 1 = буфер приемника USART полон 0 = буфер приемника USART пуст</p> <p>бит 4: TXIF: Флаг прерывания от передатчика USART 1 = буфер передатчика USART пуст 0 = буфер передатчика USART полон</p> <p>бит 3: Не реализован: читается как '0'</p> <p>бит 2: CCP1IF: Флаг прерывания от модуля CCP1 <u>Режим захвата</u> 1 = выполнен захват значения TMR1 (сбрасывается программно) 0 = захвата значения TMR1 не происходило</p> <p><u>Режим сравнения</u> 1 = значение TMR1 достигло указанного в регистрах CCP1H:CCPR1L(сбрасывается программно) 0 = значение TMR1 не достигло указанного в регистрах CCP1H:CCPR1L</p> <p><u>ШИМ режим</u> Не используется</p> <p>бит 1: TMR2IF: Флаг прерывания по переполнению TMR2 1 = произошло переполнение TMR2 (сбрасывается программно) 0 = переполнения TMR2 не было</p> <p>бит 0: TMR1IF: Флаг прерывания по переполнению TMR1 1 = произошло переполнение TMR1 (сбрасывается программно) 0 = переполнения TMR1 не было</p> | | | | | | | | |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

4.2.2.6 Регистр PCON

Регистр PCON содержит флаги, с помощью которых можно определить источник сброса микроконтроллера:

- сброс по включению питания;
- сброс по сигналу на выводе -MCLR;
- сброс по переполнению сторожевого таймера WDT;
- сброс по обнаружению снижения напряжения питания BOR.

Примечание. При включении питания бит -BOD имеет непредсказуемое значение и не должен учитываться. Бит -BOD предназначен для обнаружения последующих сбросов микроконтроллера при снижении напряжения питания. Состояние бита -BOD также непредсказуемое, если работа детектора пониженного напряжения заблокирована в битах конфигурации при программировании микроконтроллера (BODEN).

Регистр PCON (адрес 8Eh)

| U-0 | U-0 | U-0 | U-0 | R/W-1 | U-0 | R/W-q | R/W-q |
|-------|-----|-----|-----|-------------|-----|-------------|-------------|
| - | - | - | - | OSCF | - | -POR | -BOD |
| Бит 7 | | | | Бит 0 | | | |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

биты 7-4,2:**Не реализованы:** читаются как '0'

бит 3: **OSCF:** Выбор частоты тактового генератора в режимах INTRC/ER
1 = типовое значение 4МГц⁽¹⁾
0 = типовое значение 32кГц

бит 1: **-POR:** Флаг сброса по включению питания
1 = сброса по включению питания не было
0 = произошел сброс микроконтроллера по включению питания

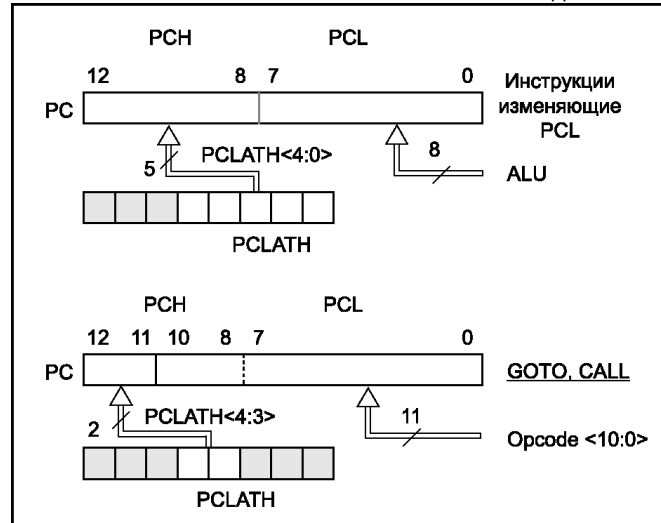
бит 0: **-BOD:** Флаг сброса по снижению напряжения питания
1 = сброса по снижению напряжения питания не было
0 = произошел сброс микроконтроллера по снижению напряжения питания

Примечание ⁽¹⁾. В режиме ER генератора и OSCF = 1 частота тактового сигнала зависит от внешнего резистора подключенного к выводу RA7/OSC1/CLKIN.

4.3 Регистры PCLATH и PCL

13-разрядный регистр счетчика команд PC указывает адрес выполняемой инструкции. Младший байт счетчика команд PCL доступен для чтения и записи. Старший байт PCH, содержащий <12:8> биты счетчика команд PC, не доступен для чтения и записи. Все операции с регистром PC происходят через дополнительный регистр PCLATH. При любом виде сброса микроконтроллера счетчик команд PC очищается. На рисунке 4-4 показано две ситуации загрузки значения в счетчик команд PC. Пример сверху, запись в счетчик команд PC происходит при записи значения в регистр PCL (PCLATH <4:0> → PCH). Пример снизу, запись значения в счетчик команд PC происходит при выполнении команды CALL или GOTO (PCLATH <4:3> → PCH).

Рис. 4-4 Запись значения в счетчик команд PC



4.3.1 Вычисляемый переход

Вычисляемый переход может быть выполнен командой приращения к регистру PCL (например, ADDWF PCL). При выполнении табличного чтения вычисляемым переходом следует заботиться о том, чтобы значение PCL не пересекло границу блока памяти (каждый блок 256 байт). Дополнительную информацию по выполнению вычисляемого перехода смотрите в документации AN556 «Выполнение табличного чтения».

4.3.2 Стек

PIC16F62X имеют 8-уровневый 13-разрядный аппаратный стек (см. рисунки 4-1, 4-2). Стек не имеет отображения на память программ и память данных, нельзя записать или прочитать данные из стека. Значение счетчика команд заносится в вершину стека при выполнении инструкций перехода на подпрограмму (CALL) или обработке прерываний. Чтение из стека и запись в счетчик команд PC происходит при выполнении инструкций возвращения из подпрограммы или обработке прерываний (RETURN, RETLW, RETFIE), при этом значение регистра PCLATH не изменяется.

Стек работает как циклический буфер. После 8 записей в стек, девятая запись запишется на место первой, а десятая запись заменит вторую и так далее.

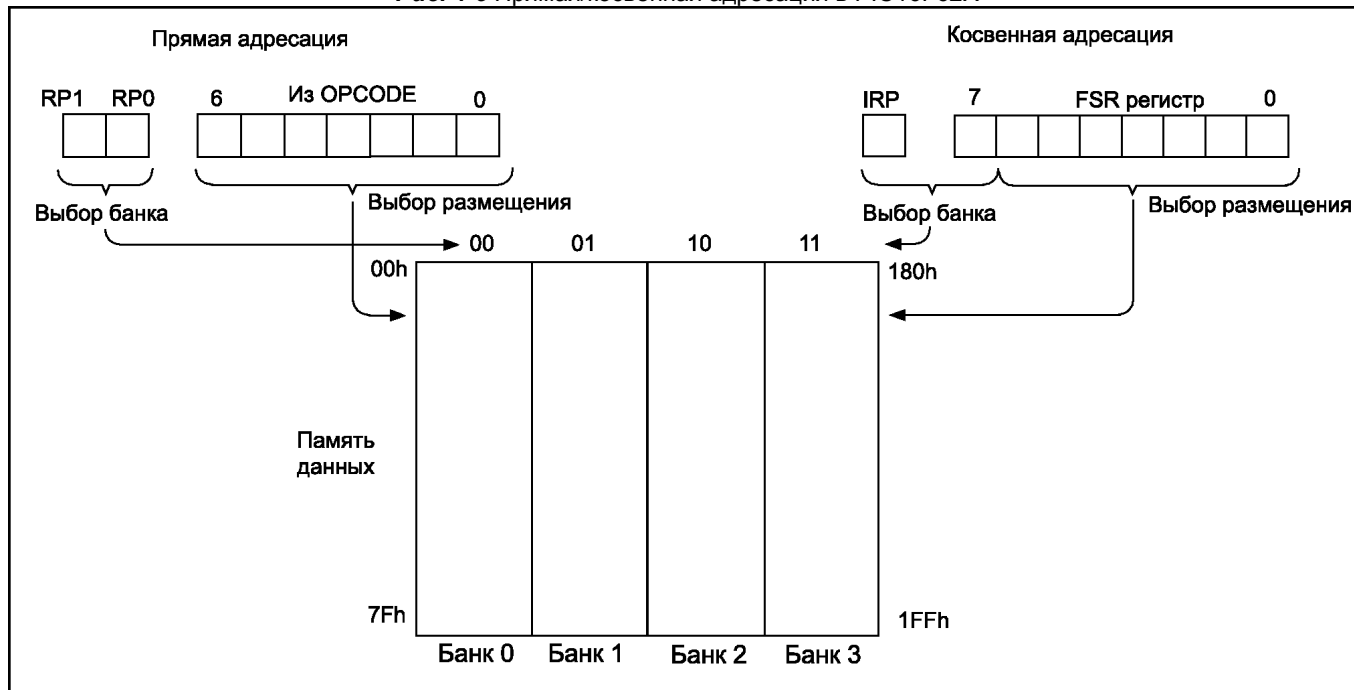
Примечания:

1. В микроконтроллерах не имеется никаких указателей о переполнении стека.
2. В микроконтроллерах не предусмотрено команд записи/чтения из стека, кроме команд вызова/возвращения из подпрограмм (CALL, RETURN, RETLW и RETFIE) или условий перехода по вектору прерываний.

4.4 Косвенная адресация, регистры INDF и FSR

Для выполнения косвенной адресации необходимо обратиться к физически нереализованному регистру INDF. Обращение к регистру INDF фактически вызовет действие с регистром, адрес которого указан в FSR. Косвенное чтение регистра INDF (FSR=0) даст результат 00h. Косвенная запись в регистр INDF не вызовет никаких действий (вызывает воздействия на флаги АЛУ в регистре STATUS). 9-бит косвенного адреса IRP сохраняется в регистре STATUS<7>. Пример 9-разрядной косвенной адресации показан на рисунке 4-5.

Рис. 4-5 Прямая/косвенная адресация в PIC16F62X



Карту памяти данных смотрите на рисунке 4-3.

В примере 4-1 показано использование косвенной адресации для очистки памяти данных (диапазон адресов 20h–2Fh).

Пример 4-1 Косвенная адресация

```

BCF    STATUS, IRP    ; Установить банк 0,1
MOVLW 0x20           ; Указать первый регистр в ОЗУ
MOVWF FSR
NEXT:
CLRF   INDF          ; Очистить регистр
INCF   FSR, F        ; Увеличить адрес
BTFSS FSR, 4        ; Завершить?
GOTO   NEXT          ; Нет, продолжить очистку

CONTINUE:
; Да

```

5.0 Порты ввода/вывода

PIC16F62X имеют два порта ввода/вывода, PORTA и PORTB. Некоторые каналы портов мультиплицированы с периферийными модулями микроконтроллера. Когда периферийный модуль включен, вывод не может использоваться как универсальный канал ввода/вывода.

5.1 Регистры PORTA и TRISA

PORTA – 8-разрядный порт ввода вывода. RA4 имеет триггер Шмидта на входе и открытый сток на выходе, мультиплицирован с тактовым входом T0CKI. RA5 имеет триггер Шмидта на входе, без выходного буфера. Все остальные каналы PORTA имеют триггер Шмидта на входе и полнофункциональные выходные КМОП буферы.

Все каналы PORTA имеют соответствующие биты направления в регистре TRISA, позволяющие настраивать канал как вход или выход.

Запись '1' в TRISA переводит соответствующий выходной буфер в 3-е состояние. Запись '0' в регистр TRISA определяет соответствующий канал как выход, содержимое защелки PORTA передается на вывод микроконтроллера.

Чтение регистра PORTA возвращает состояние на выводах порта, а запись производится в защелку PORTA. Все операции записи в порт выполняются по принципу «чтение – модификация - запись», т.е. сначала производится чтение состояния выводов порта, затем изменение и запись в защелку.

Каналы PORTA мультиплицированы с модулем компараторов и источником опорного напряжения. Настройка работы этих периферийных модулей определяется битами в регистрах CMCON (управляющий регистр компаратора) и VRCO (управляющий регистр источника опорного напряжения). Когда компаратор включен, чтение состояние выводов PORTA будет давать результат '00'.

Примечания:

1. При сбросе микроконтроллера все биты регистра TRISA устанавливаются в '1', настраивая PORTA на вход. Функция цифрового входа заблокирована, т.к. каналы PORTA подключены к модулю компараторов для снижения тока потребления.
2. Если вывод RA6/OSC2/CLKOUT настроен как CLKOUT, то соответствующие биты регистров TRISA, PORTA игнорируются и читаются как '0'.

Биты регистра TRISA управляют направлением каналов PORTA при включенном модуле компараторов. Пользователь должен удостовериться, что соответствующие каналы PORTA настроены на вход при использовании их в качестве входов компаратора.

RA2 также функционировать как выход при включенном модуле источника опорного напряжения (V_{REF} находится в режиме высокого импеданса). Пользователь должен установить в '1' бит TRISA<2> для настройки канала на вход.

В одном из режимов (определяется регистром CMCON) RA3 и RA4 становятся выходами компаратора. Биты TRISA<4:3> должны быть сброшены в '0' для работы в этом режиме.

Пример 5-1 Инициализация PORTA

```

CLRf    PORTA    ; Инициализация защелок PORTA

MOVLW  0X07    ; Выключить компаратор,
MOVWF  CMCON    ; каналы PORTA – цифровые входы/выходы

BCF    STATUS, RP1
BSF    STATUS, RP0 ; Выбрать банк 1
MOVLW  0x1F    ; Значение для инициализации
                ; направления каналов PORTA
MOVWF  TRISA    ; Настроить RA<4:0> как входы
                ; Биты TRISA<7:5> всегда
                ; читаются как '0'.

```

Рис. 5-1 Структурная схема выводов RA0/AN0 и RA1/AN1

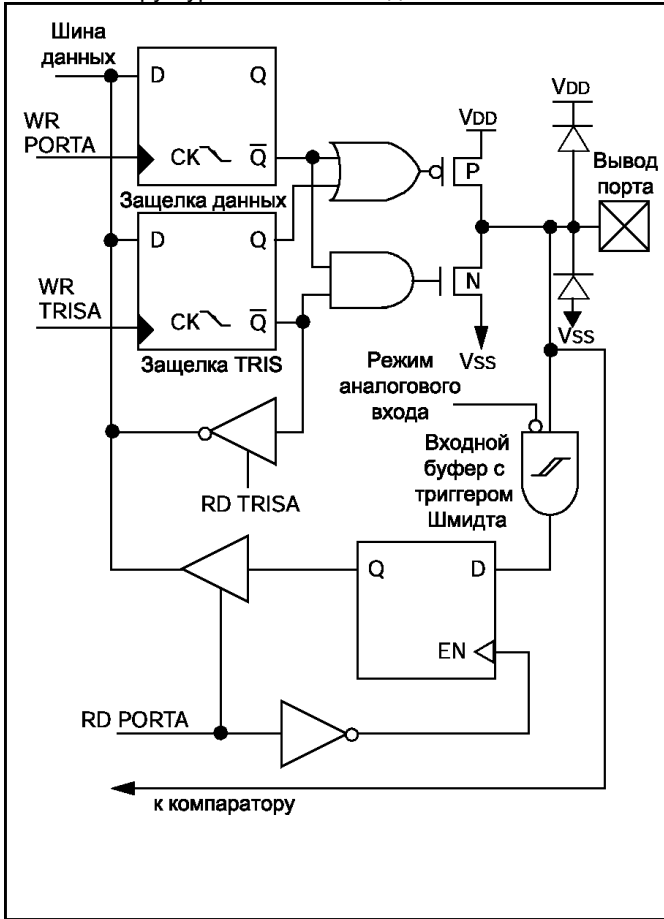


Рис. 5-2 Структурная схема вывода RA2/VREF

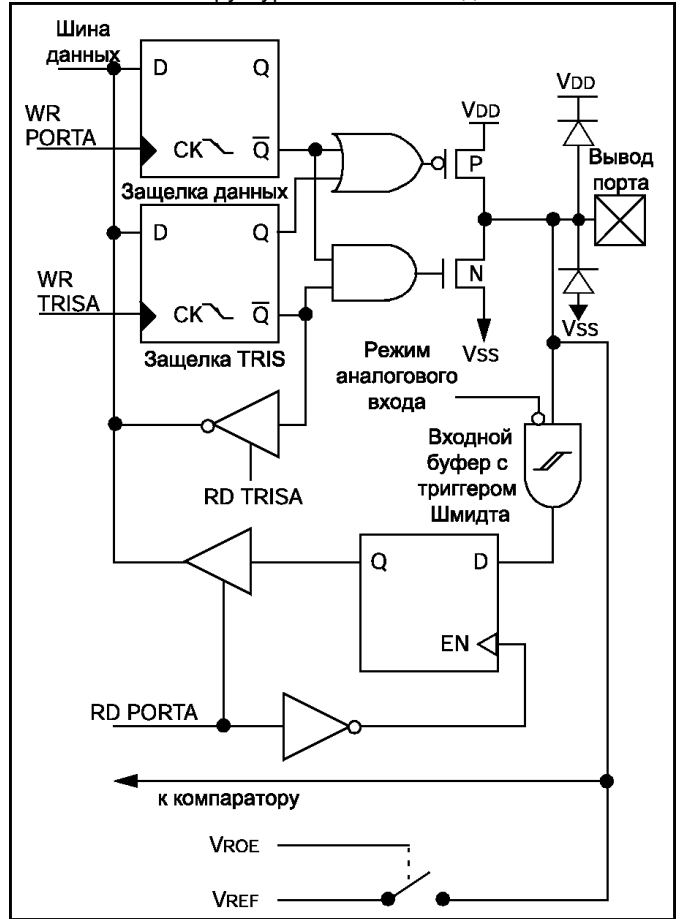


Рис. 5-3 Структурная схема вывода RA3/AN3

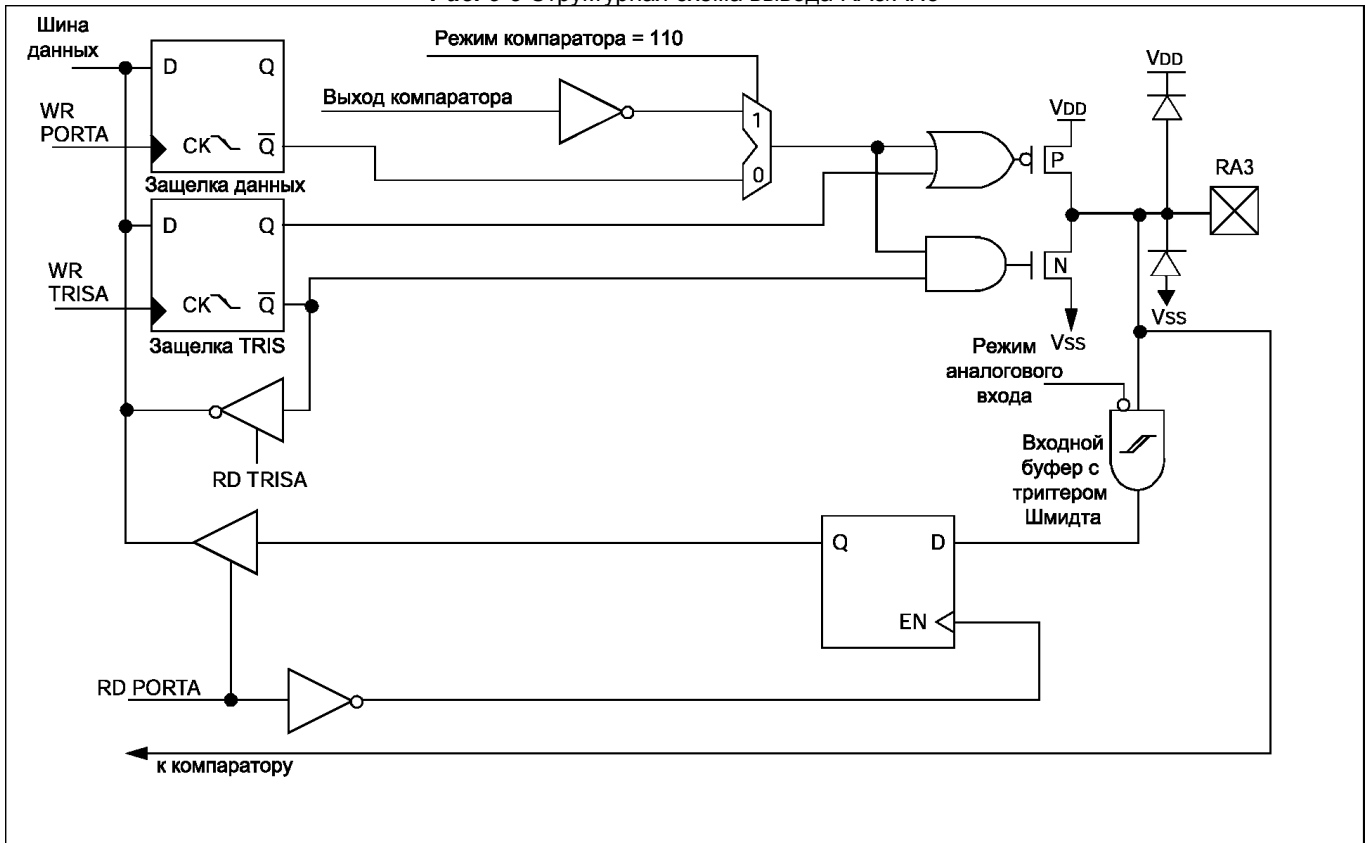


Рис. 5-4 Структурная схема вывода RA4/T0CKI

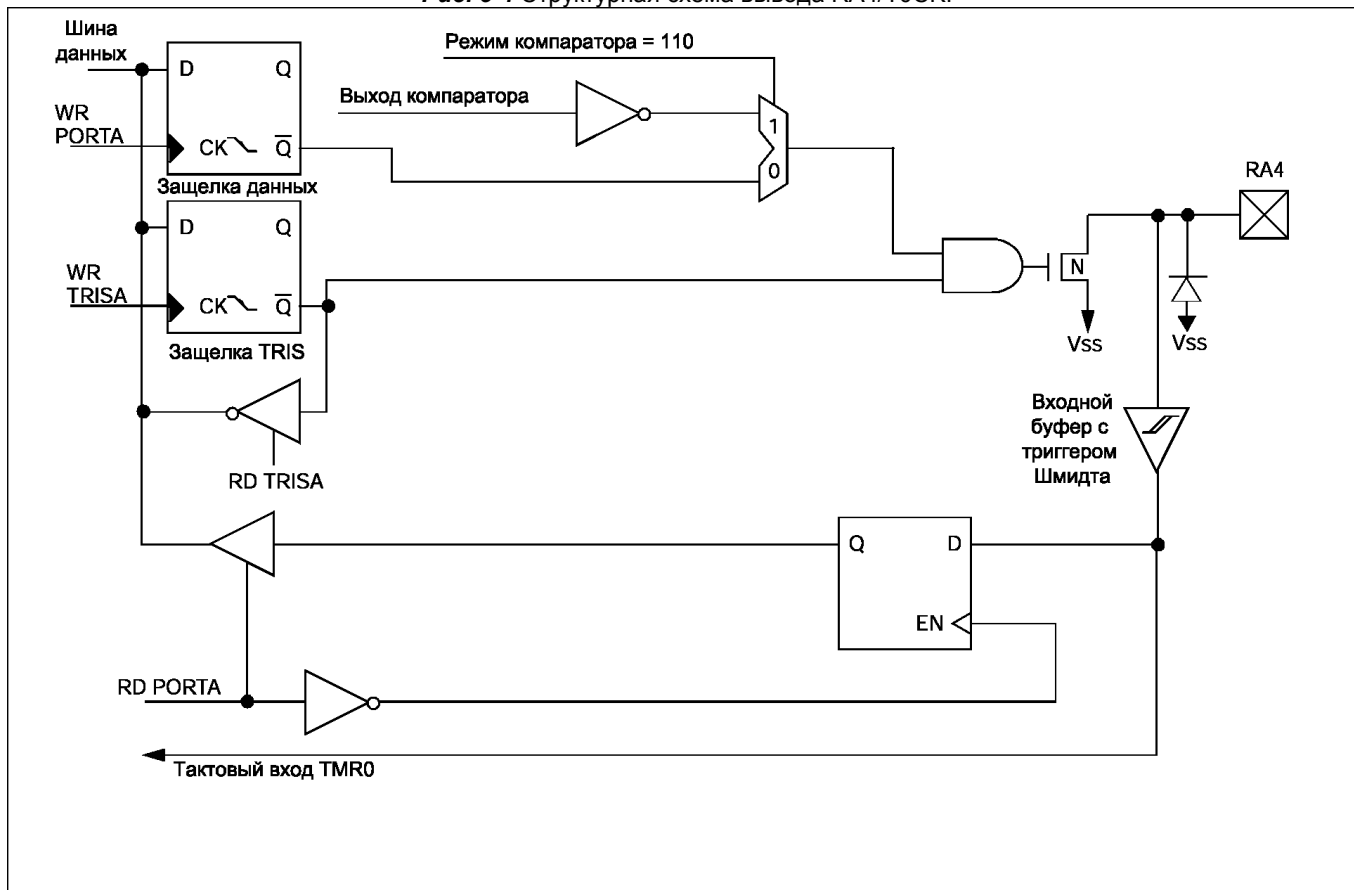


Рис. 5-5 Структурная схема вывода RA4/MCLR/THV

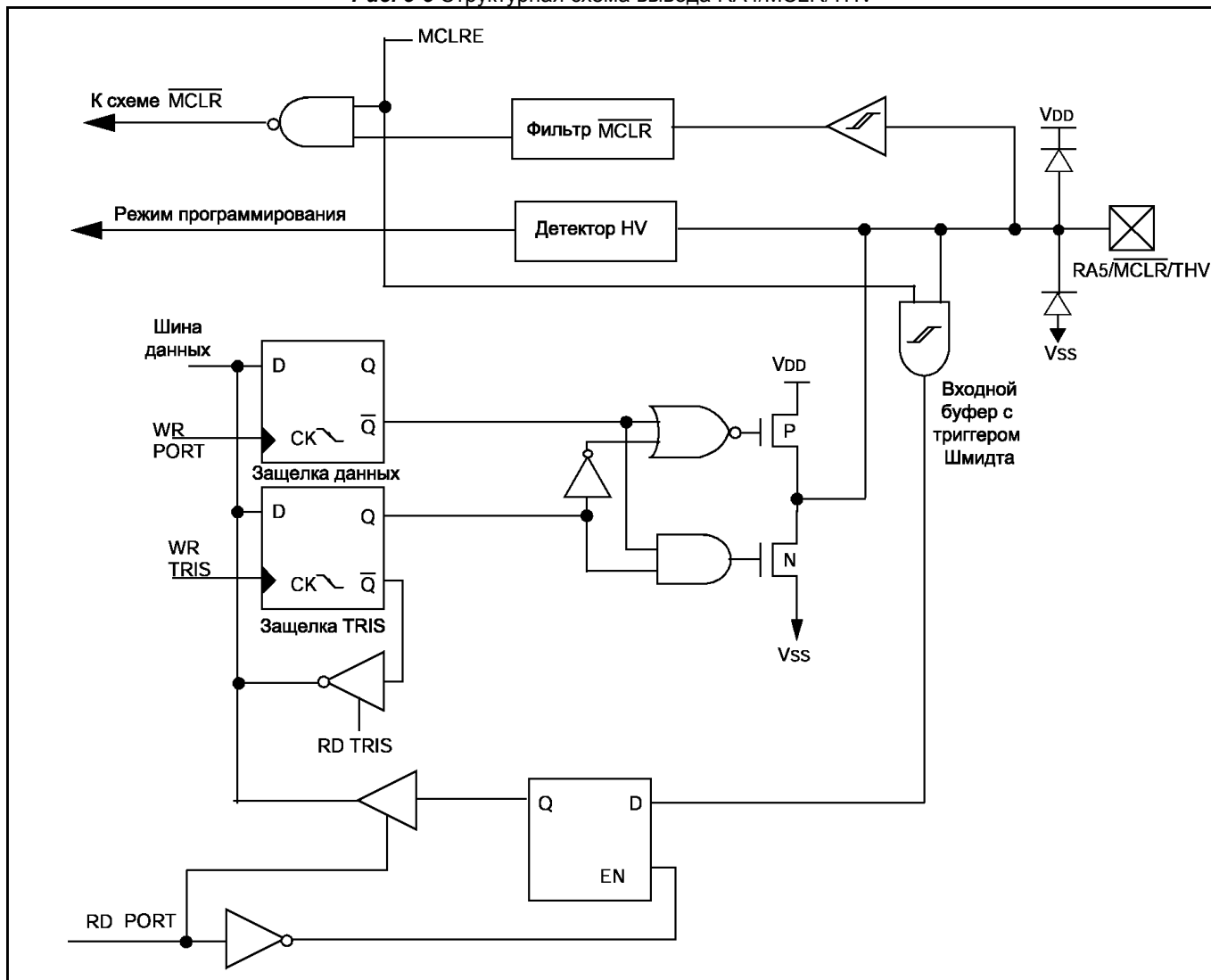


Рис. 5-6 Структурная схема вывода RA6/OSC2/CLKOUT

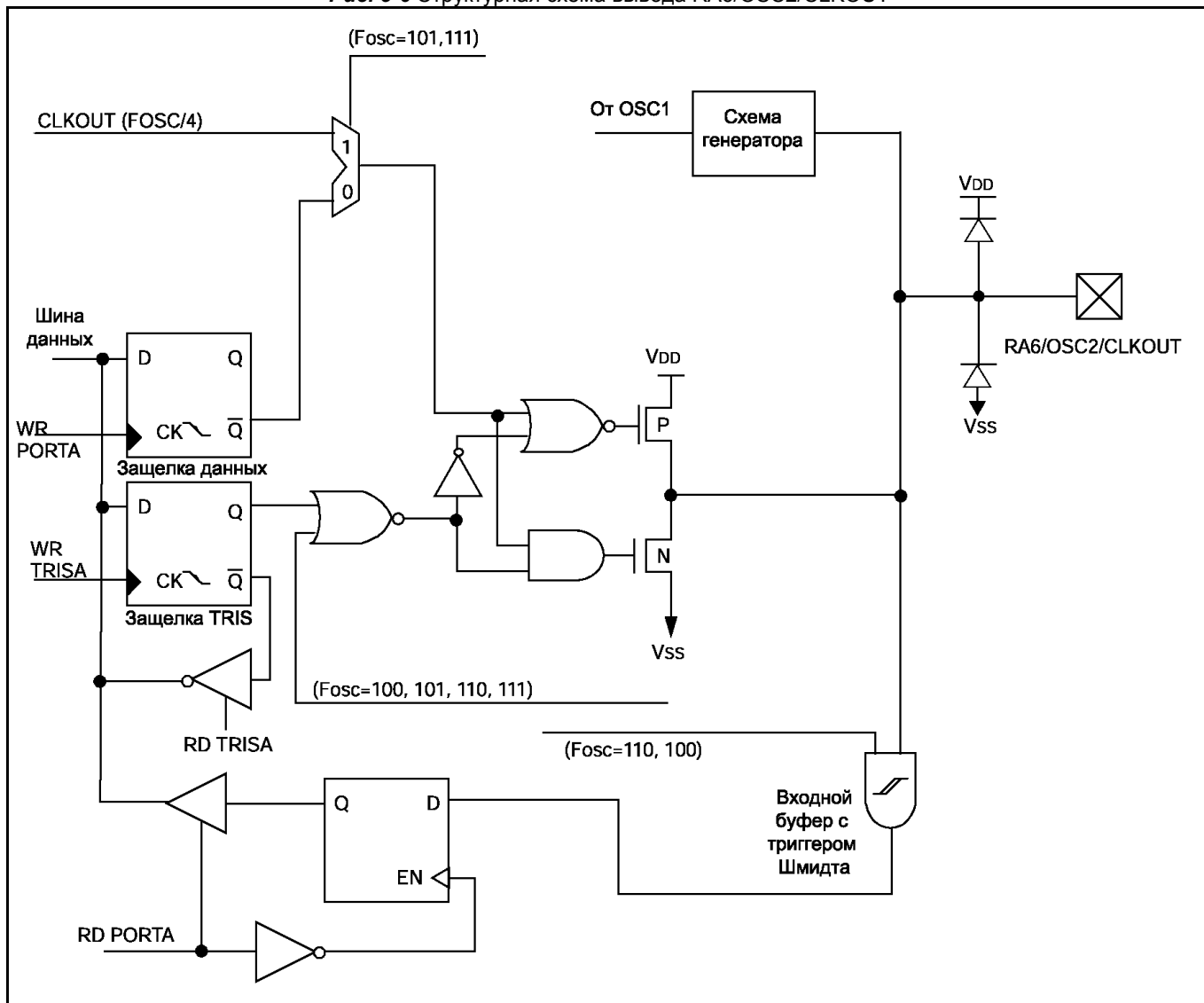


Рис. 5-7 Структурная схема вывода RA7/OSC1/CLKIN

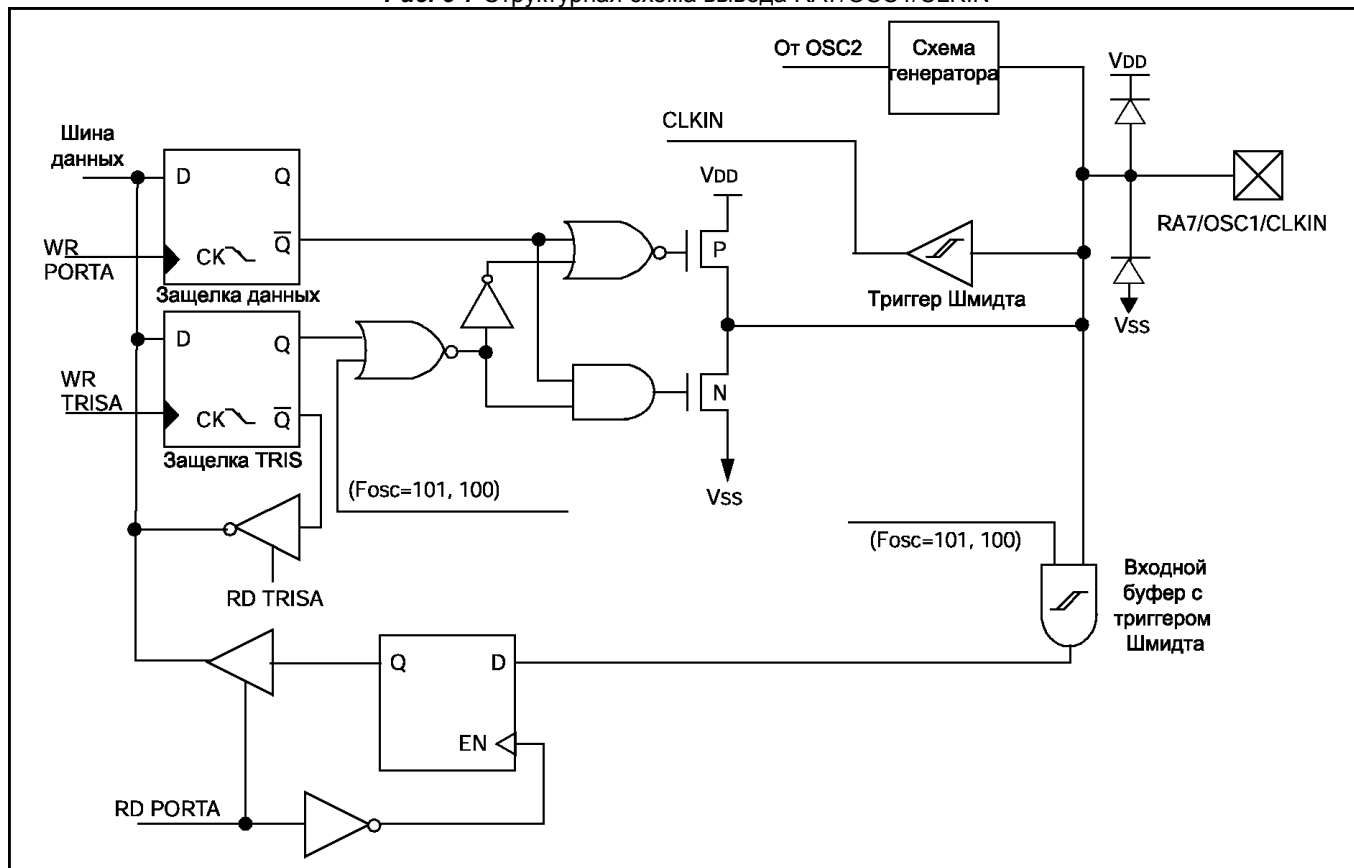


Таблица 5-1 Функциональное назначение выводов PORTA

| Обозначение вывода | № бита | Тип буфера | Описание |
|--------------------------|--------|------------|--|
| RA0/AN0 | Бит 0 | ST | Двунаправленный порт ввода/вывода, аналоговый вход компаратора |
| RA1/AN1 | Бит 1 | ST | Двунаправленный порт ввода/вывода, аналоговый вход компаратора |
| RA2/AN2/V _{REF} | Бит 2 | ST | Двунаправленный порт ввода/вывода, аналоговый вход компаратора, выход источника опорного напряжения V _{REF} |
| RA3/AN3/CPM1 | Бит 3 | ST | Двунаправленный порт ввода/вывода, аналоговый вход компаратора, выход компаратора |
| RA4/T0CKI/CPM2 | Бит 4 | ST | Двунаправленный порт ввода/вывода, может использоваться как T0CKI, выход компаратора |
| RA5/MCLR/THV | Бит 5 | ST | Вход сброса микроконтроллера, вход напряжения программирования. Когда вывод настроен как -MCLR, то по низкому уровню сигнала производится сброс микроконтроллера. При нормальной работе напряжение на -MCLR/THV не должно превышать VDD. |
| RA6/OSC2/CLKOUT | Бит 6 | ST | Двунаправленный порт ввода/вывода, выход генератора для подключения резонатора. В режиме ER генератора на выходе CLKOUT формируется сигнал с частотой 1/4 OSC1, обозначая циклы команд |
| RA7/OSC1/CLKIN | Бит 7 | ST | Двунаправленный порт ввода/вывода, вход генератора, вход внешнего тактового сигнала, вывод ER смещения |

Обозначение: ST – вход с триггером Шмидта.

Таблица 5-2 Регистры и биты связанные с работой PORTA

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|-------|-------|--------|--------|-------|--------|--------|--------|--------|--------|-----------|---------------|
| 05h | PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xxxx 0000 | xxxx 0000 |
| 85h | TRISA | TRISA7 | TRISA6 | - | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 11-1 1111 | 11-1 1111 |
| 1Fh | CON | C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 0000 0000 | 0000 0000 |
| 9Fh | VRCON | VREN | VROE | VRR | - | VR3 | VR2 | VR1 | VR0 | 000- 0000 | 000- 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.
Примечание. Затененные биты не влияют на работу PORTA.

5.2 Регистры PORTB и TRISB

PORTB – 8-разрядный двунаправленный порт ввода/вывода. Биты регистра TRISB определяют направление каналов порта. Установка бита в '1' регистра TRISB переводит выходной буфер в 3-е состояние. Запись '0' в регистр TRISB настраивает соответствующий канал как выход, содержимое защелки PORTB передается на вывод микроконтроллера.

Выводы PORTB мультиплицированы с внешним входом прерываний, USART, CCP модулем и входом тактового сигнала для TMR1. Функциональное назначение выводов PORTB сведено в таблице 5-3.

Чтение регистра PORTB возвращает состояние на выводах порта, а запись производится в защелку PORTB. Все операции записи в порт выполняются по принципу «чтение – модификация - запись», т.е. сначала производится чтение состояния выводов порта, затем изменение и запись в защелку.

К каждому выводу PORTB подключен внутренний подтягивающий резистор (типовое значение тока $\approx 200\mu\text{A}$). Бит -RBPU в регистре OPTION <7> определяет, подключены (-RBPU=0) или нет (-RBPU=1) подтягивающие резисторы. Подтягивающие резисторы автоматически отключаются, когда каналы порта настраиваются на выход и после сброса по включению питания POR.

Четыре канала PORTB RB7:RB4 настроенные на вход могут генерировать прерывания по изменению логического уровня сигнала на входе. Если один из каналов RB7:RB4 настроен на выход, то он не может быть источником прерываний. Сигнал на выводах RB7:RB4 сравнивается со значением, сохраненным при последнем чтении PORTB. В случае несовпадения одного из значений устанавливается флаг RBIF в регистре INTCON<0>, если разрешено, генерируется прерывание.

Это прерывание может ввести микроконтроллер из режима SLEEP. В подпрограмме обработки прерываний необходимо сделать следующие действия:

- выполнить чтение или запись в PORTB исключив несоответствие;
- сбросить флаг RBIF и '0'.

Несоответствие сохраненного значения с сигналом на входе PORTB всегда устанавливает бит RBIF в '1'. Чтение из PORTB прервет условие несоответствия и позволит сбросить флаг RBIF в '0'.

Прерывания по изменению сигнала на входах PORTB, программа переключения конфигурации этих каналов позволяет реализовать простой интерфейс обслуживания клавиатуры, с выходом из режима SLEEP по нажатию клавиш (см. пример AN552 в книге Microchip *Embedded Control Handbook* или на WEB узлах технической поддержки www.microchip.com и www.microchip.ru).

Примечание. Если изменение сигнала на входе RB7:RB4 происходит на начале цикла Q2 чтения PORTB, флаг RBIF в '1' может не устанавливаться.

Прерывания по изменению сигнала на входах рекомендуется использовать для выхода из режима SLEEP по нажатию клавиш, когда PORTB полностью задействован для реализации клавиатуры. Не рекомендуется опрашивать PORTB при использовании прерываний по изменению входного сигнала.

Рис. 5-8 Структурная схема вывода RB0/INT

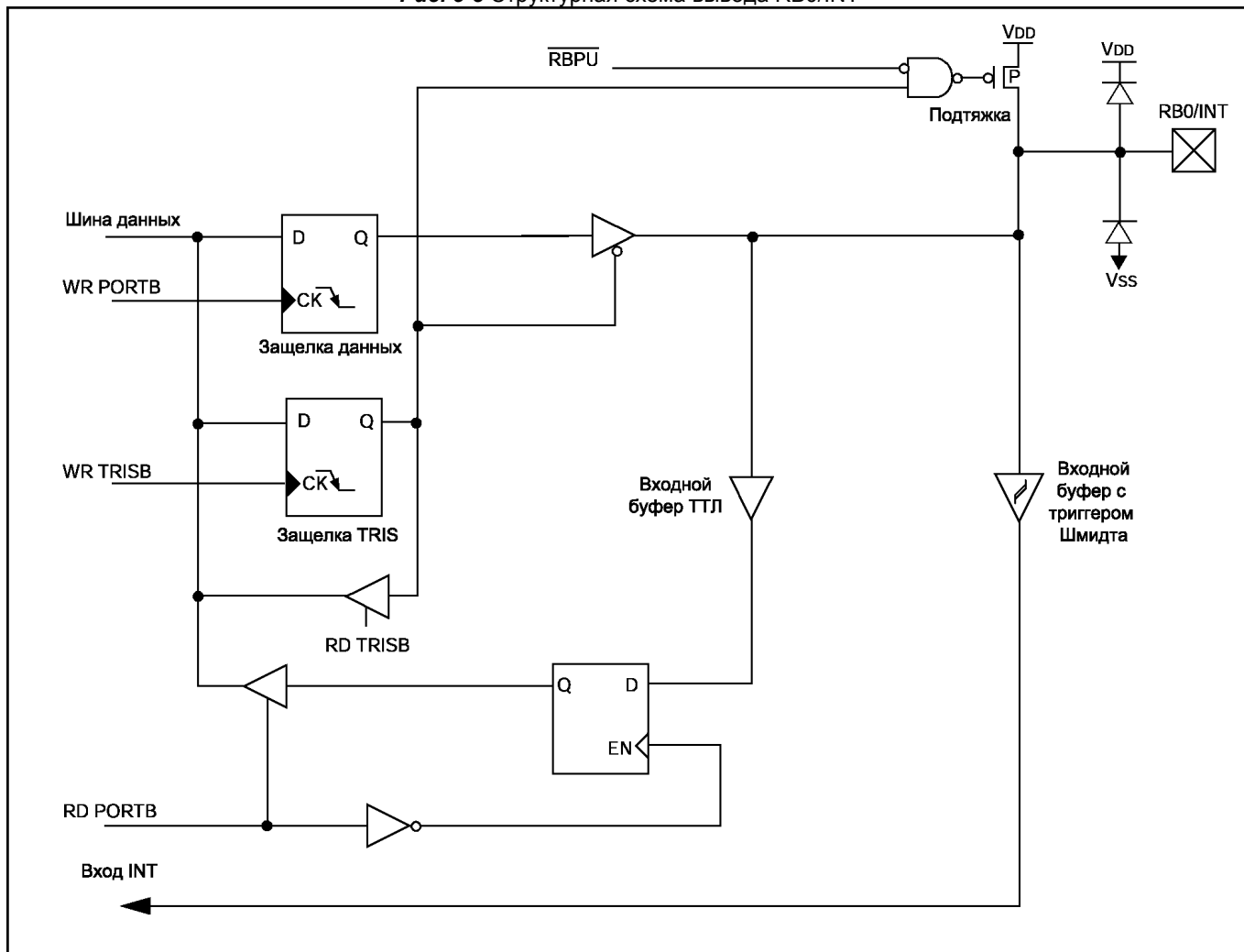
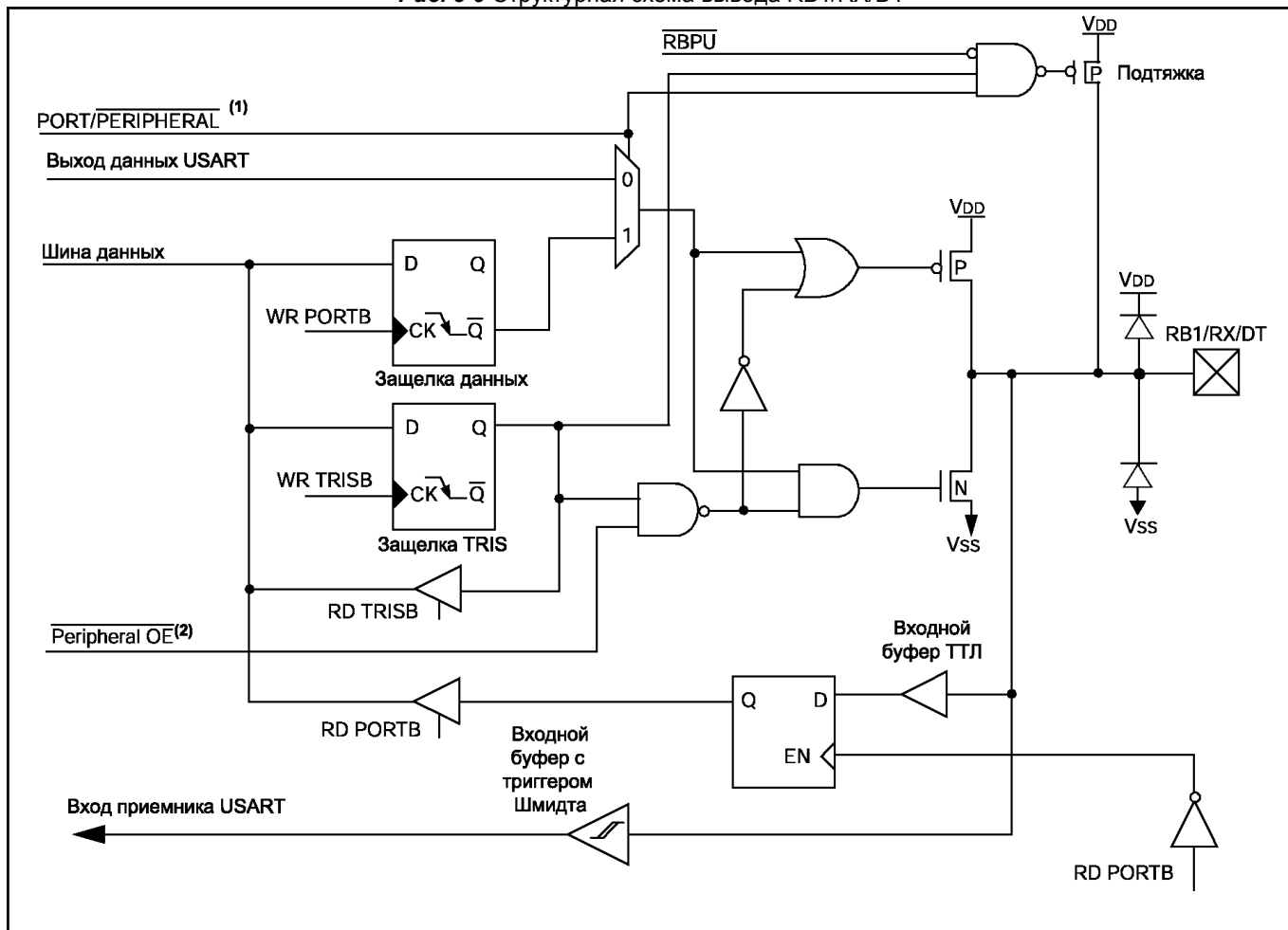
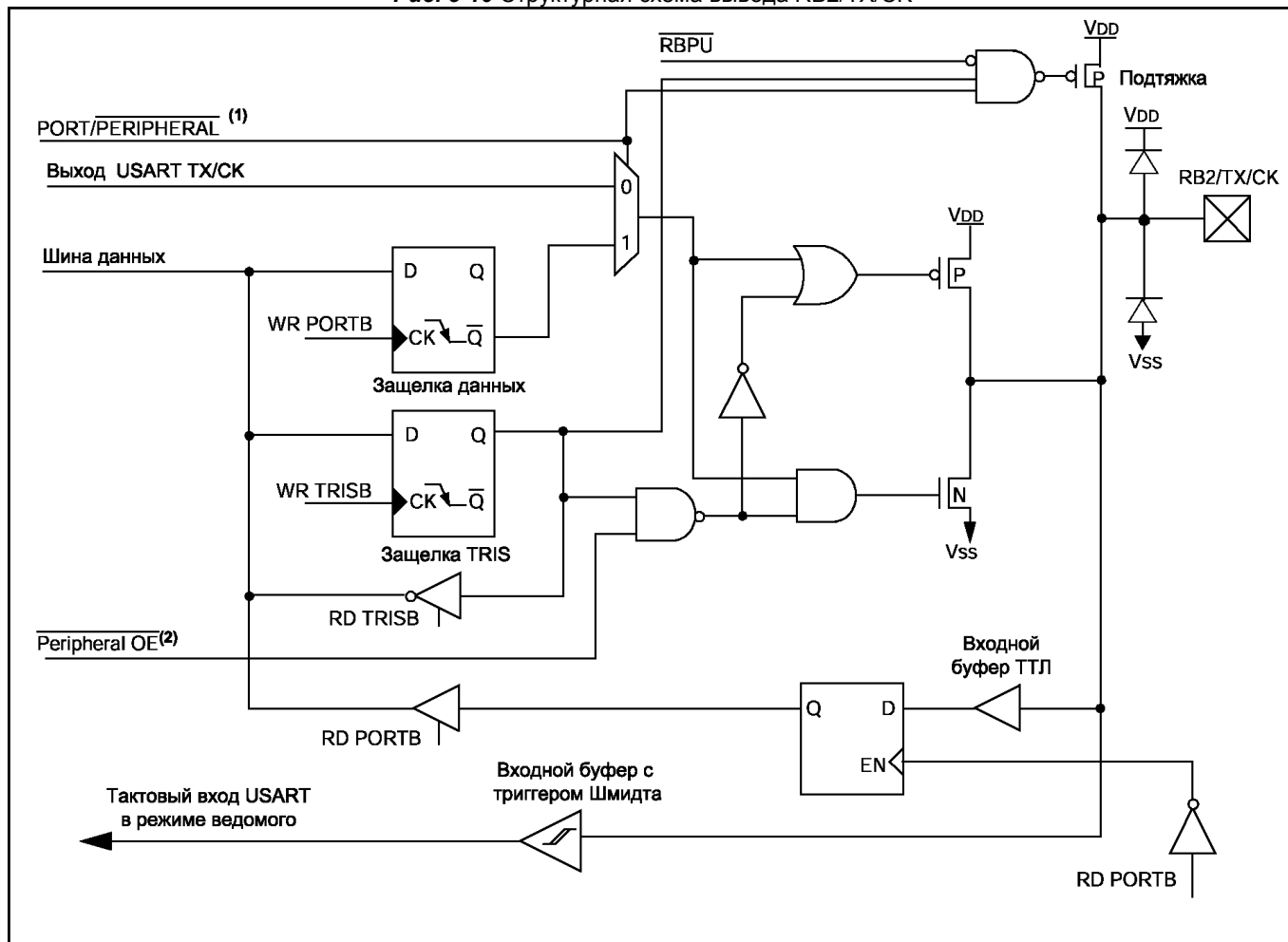


Рис. 5-9 Структурная схема вывода RB1/RX/DT

**Примечания:**

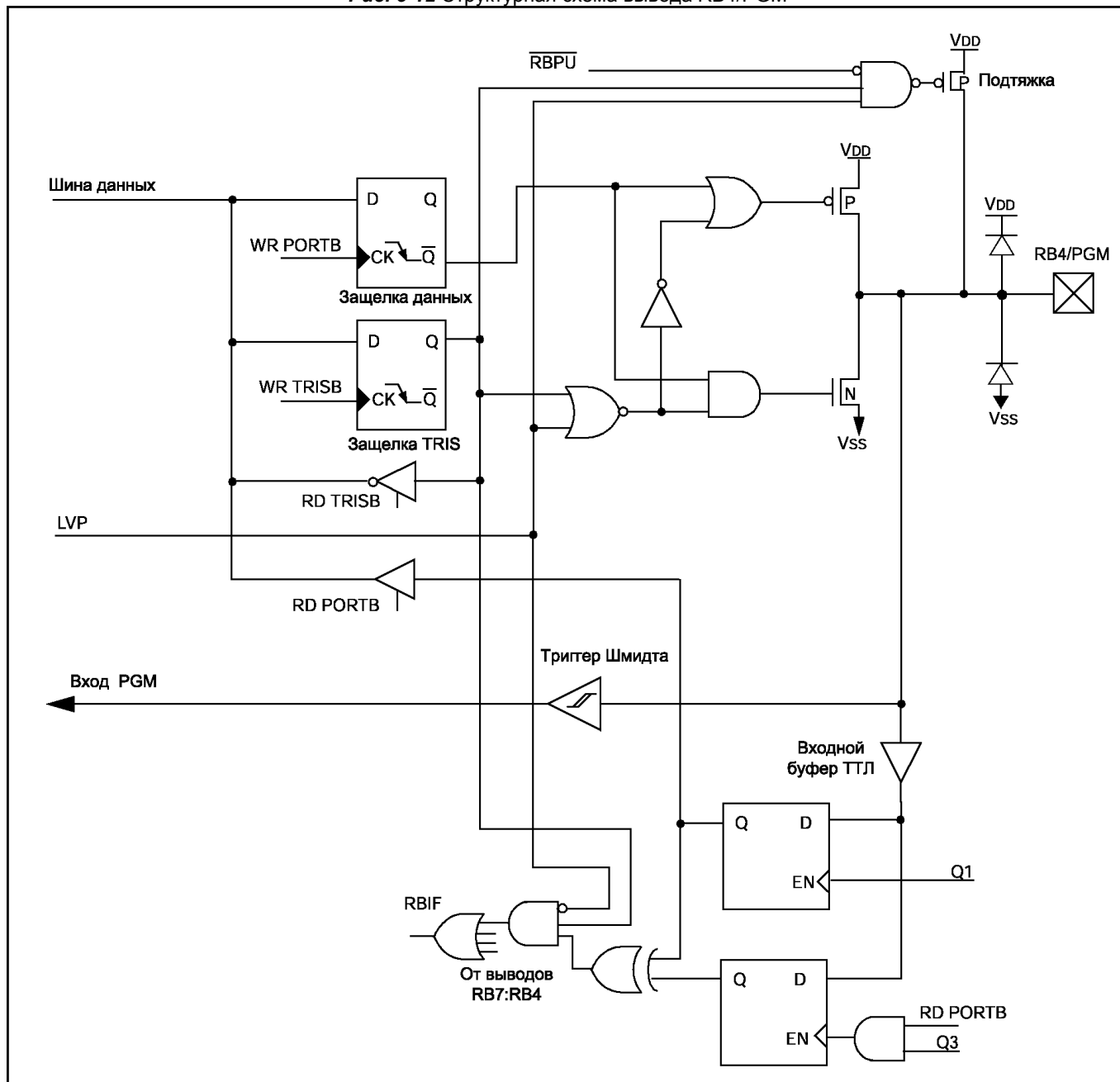
1. Сигнал режима канала – вывод используется периферийным модулем или цифровой порт ввода/вывода.
2. Сигнал разрешения (OE) от периферийного модуля, настраивать канал как выход.

Рис. 5-10 Структурная схема вывода RB2/TX/CK

**Примечания:**

1. Сигнал режима канала – вывод используется периферийным модулем или цифровой порт ввода/вывода.
2. Сигнал разрешения (OE) от периферийного модуля, настраивать канал как выход.

Рис. 5-12 Структурная схема вывода RB4/PGM



Примечание. Режим низковольтного программирования отключает прерывания по изменению сигнала на входе и подтягивающий резистор для вывода RB4.

Рис. 5-13 Структурная схема вывода RB5

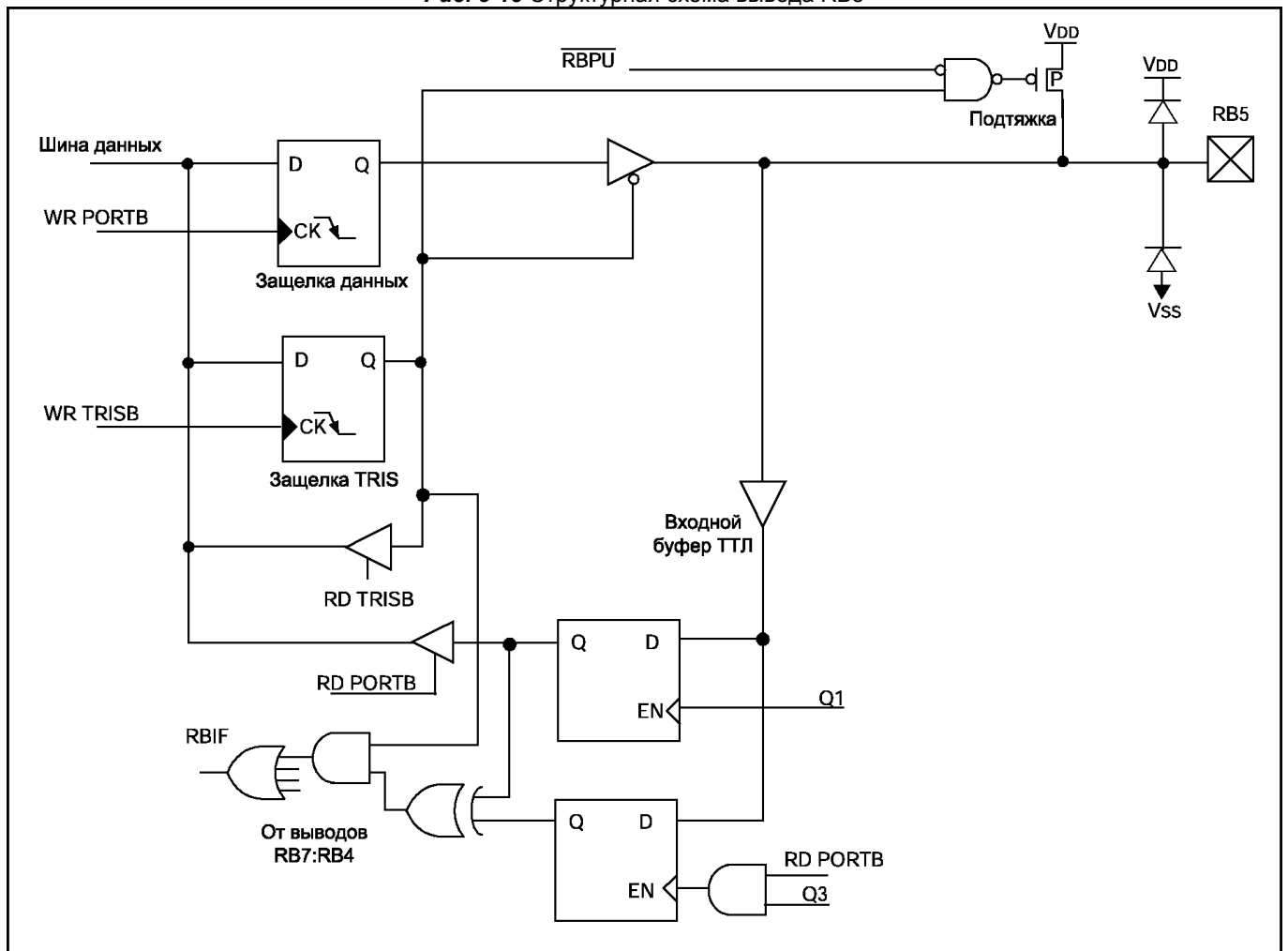


Рис. 5-14 Структурная схема вывода RB6/T1OSO/T1CKI

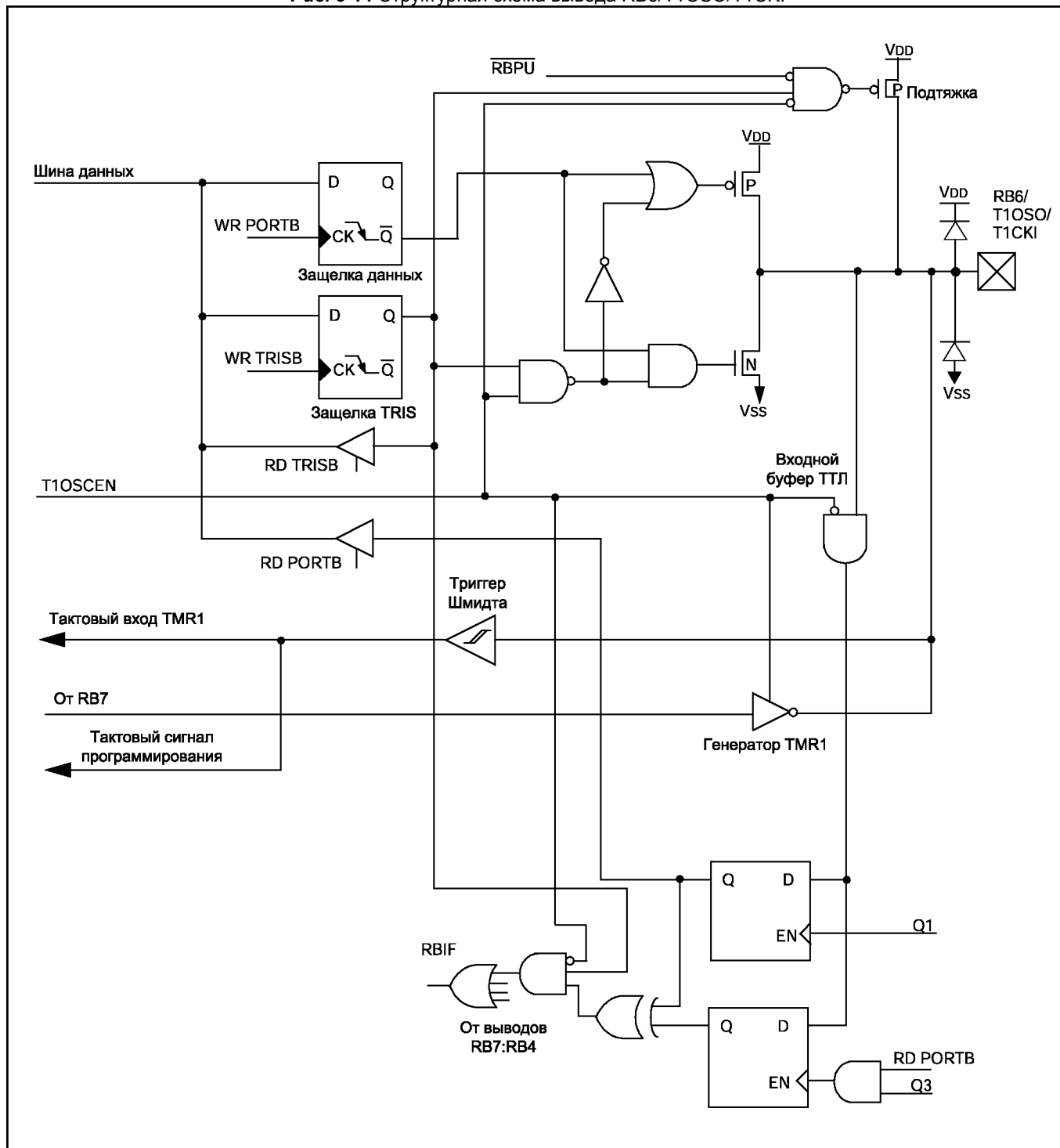


Рис. 5-15 Структурная схема вывода RB7/T10SI

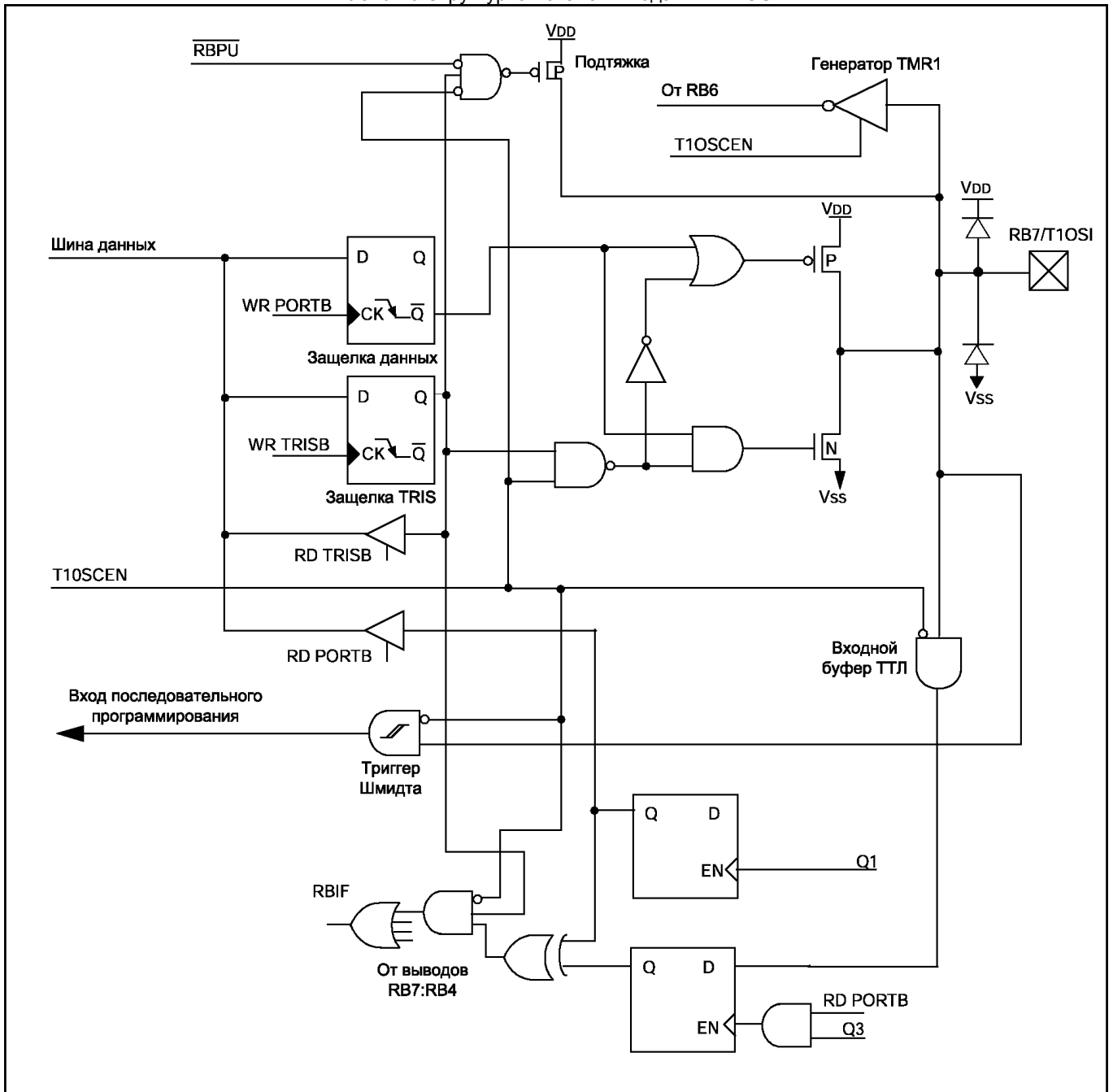


Таблица 5-3 Функциональное назначение выводов PORTB

| Обозначение вывода | № бита | Тип буфера | Описание |
|--------------------|--------|-----------------------|--|
| RB0/INT | Бит 0 | TTL/ST ⁽¹⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вход внешнего прерывания |
| RB1/RX/DT | Бит 1 | TTL/ST ⁽³⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вход приемника USART, линия данных в синхронном режиме USART |
| RB2/TX/CK | Бит 2 | TTL/ST ⁽³⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, выход передатчика USART, линия тактового сигнала в синхронном режиме |
| RB3/CCP1 | Бит 3 | TTL/ST ⁽⁴⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вывод модуля CCP |
| RB4/PGM | Бит 4 | TTL/ST ⁽⁵⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP. Когда разрешено низковольтное программирование, запрещены прерывания по изменению сигнала на входе, а подтягивающий резистор отключен |
| RB5 | Бит 5 | TTL | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP |
| RB6/T1OSO/T1CKI | Бит 6 | TTL/ST ⁽²⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP. Выход генератора таймера 1 |
| RB7/T1OSI | Бит 7 | TTL/ST ⁽²⁾ | Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора. Изменение сигнала на входе может вывести микроконтроллер из режима SLEEP. Вход генератора таймера 1 |

Обозначения: TTL – вход/выход ТТЛ, ST – вход с триггером Шмидта.

Примечания:

1. Входной буфер с триггером Шмидта при использовании внешних прерываний.
2. Входной буфер с триггером Шмидта при работе в режиме последовательного программирования.
3. Входной буфер с триггером Шмидта при использовании выводов модулем USART.
4. Входной буфер с триггером Шмидта при использовании выводов модулем CCP.
5. Входной буфер с триггером Шмидта при низковольтном программировании.

Таблица 5-4 Регистры и биты связанные с работой PORTB

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----------|---------------|
| 06h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 1111 1111 |
| 81h | OPTION | -RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу PORTB.

5.3 Программирование портов ввода/вывода

5.3.1 Двухнаправленные порты ввода/вывода

Все операции записи в порт выполняются по принципу «чтение – модификация - запись». Например, команды BCF и BSF считывают значение в регистр CPU, выполняют битовую операцию и записывают результат обратно в регистр. Требуется некоторая осторожность при применении подобных команд к регистрам портов ввода/вывода. Например, команда BSF PORTB,5 считывает все восемь битов PORTB в CPU, изменяет состояние бита 5 и записывает результат в выходные защелки PORTB. Если другой канал PORTB (например, RB0) настроен на вход, то сигнал на выводе будет считан в CPU и записан в защелку данных, поверх предыдущего значения. Пока RB0 настроен как вход, никаких проблем не возникает. Однако, если RB0 будет позже настроен как выход, значение в защелке данных может отличаться от требуемого.

При чтении регистра порта, читается текущее состояние порта ввода/вывода. Запись в регистр порта, сохраняет значение в защелке порта ввода/вывода. Когда используются команды «чтение - модификация – запись» (например, BSF, BCF и т.д.) считывается текущее состояние порта ввода/вывода, выполняется требуемая операция и полученное значение записывается в защелку порта.

В примере 5-2 показан эффект последовательного выполнения команд «чтение - модификация – запись» (например, BSF, BCF и т.д.) с регистром порта ввода/вывода.

На активный вывод порта не должны подключаться нагрузки включенные по схемам «монтажное И» или «монтажное ИЛИ». Возможные большие токи могут повредить микроконтроллер.

Пример 5-2 Эффект выполнения команд «чтение - модификация – запись».

Начальные установки порта: PORTB<7:4> входы, PORTB <3:0> выходы.

Выводы RB7:RB6 имеют внешние подтягивающие резисторы и не подключены к другим цепям схеме.

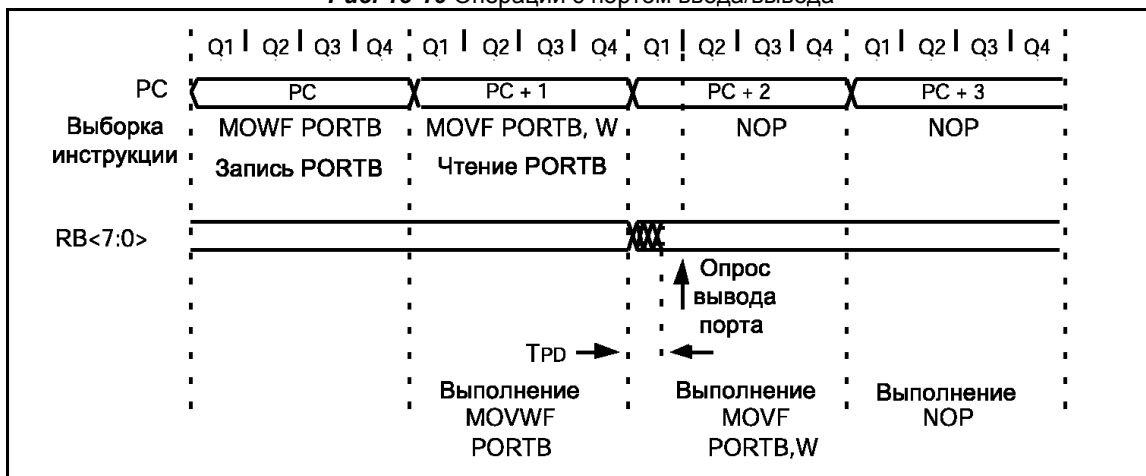
| | | Защелка PORTB | Выводы PORTB |
|-----|-------------|---------------|--------------|
| BCF | STATUS,RPO | ; | |
| BCF | PORTB, 7 | ; 01pp rppp | 11pp rppp |
| BCF | PORTB, 6 | ; 10pp rppp | 11pp rppp |
| BSF | STATUS,RP0; | | |
| BCF | TRISB, 7 | ; 10pp rppp | 11pp rppp |
| BCF | TRISB, 6 | ; 10pp rppp | 10pp rppp |

Обратите внимание. Возможно, пользователь ожидал, что после выполнения программы на выходах PORTB будет значение 00pp rppp. Однако, 2-я команда BCF установила в '1' RB7.

5.3.2 Последовательность операций с портами ввода/вывода

Запись в порт ввода/вывода фактически происходит в конце машинного цикла, а чтение данных выполняется в начале цикла (см. рисунок 5-16). Поэтому требуется некоторая осторожность при записи в порт ввода/вывода, если перед записью выполняется чтение состояния порта. Последовательность команд должна быть такой, чтобы установилось напряжение на выводе порта прежде, чем будет выполнена команда записи в порт, сопровождаемая чтением состояния выводов (иначе вместо нового значения может быть считано предыдущее). Если возможна описанная ситуация, разделите команды записи инструкциями NOP или любыми другими командами, которые не обращаются к порту ввода/вывода.

Рис. 15-16 Операции с портом ввода/вывода



Примечание к рисунку. На рисунке показан пример чтения из PORTB сразу после записи в него. Время установления данных на PORTB равно $T = 0.25 T_{CY} - T_{PD}$. Где: T_{CY} – длительность машинного цикла микроконтроллера, T_{PD} – задержка распространения. Следовательно, при высокой тактовой частоте микроконтроллера, чтение с порта ввода/вывода непосредственно после записи может возвращать неверные значения.

6.0 Модуль таймера TMR0

TMR0 – таймер/счетчик, имеет следующие особенности:

- 8-разрядный таймер/счетчик;
- возможность чтения и записи текущего значения счетчика;
- 8-разрядный программируемый предделитель;
- внутренний или внешний источник тактового сигнала;
- выбор активного фронта внешнего тактового сигнала
- прерывания при переполнении (переход от FFh к 00h).

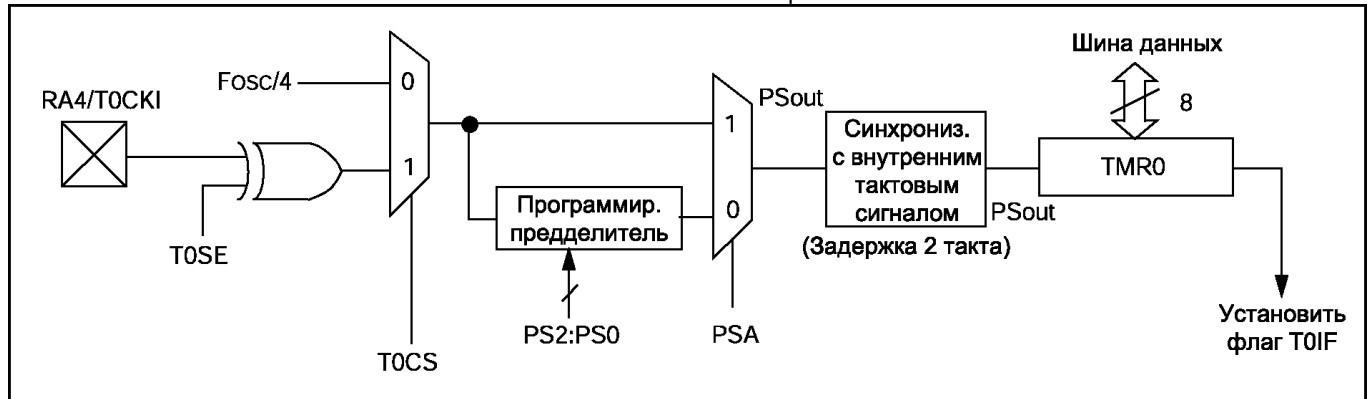
Блок схема модуля TMR0 показана на рисунке 6-1.

Когда бит TOCS сброшен в '0' (OPTION<5>), TMR0 работает от внутреннего тактового сигнала. Приращение счетчика TMR0 происходит в каждом машинном цикле (если предделитель отключен). После записи в TMR0 приращение счетчика запрещено два следующих цикла (см. рисунки 6-2 и 6-3). Пользователь должен скорректировать эту задержку перед записью нового значения в TMR0.

Если бит TOCS установлен в '1' (OPTION<5>), TMR0 работает от внешнего источника тактового сигнала с входа RA4/T0CKI. Активный фронт внешнего тактового сигнала выбирается битом T0SE в регистре OPTION<4> (T0SE=0 – активным является передний фронт сигнала). Работа модуля TMR0 с внешним источником тактового сигнала будет рассмотрена в разделе 6.2.

Предделитель может быть включен перед WDT или TMR0, в зависимости от состояния бита PSA в регистре OPTION<3>. Если бит PSA сброшен в '0', то предделитель включен перед TMR0. Нельзя прочитать или записать новое значение в предделитель. Когда предделитель включен перед TMR0, можно выбрать его коэффициент деления 1:2, 1:4, ..., 1:256. Подробное описание работы с предделителем смотрите в разделе 6.3.

Рис. 6-1 Блок схема таймера TMR0



Примечания.

1. Биты управления T0CS, T0SE, PS2, PS1, PS0, PSA расположены в регистре OPTION.
2. Схему включения предделителя перед WDT смотрите на рисунке 6-6.

Рис. 6-2 Временная диаграмма работы TMR0 от внутреннего источника тактового сигнала (предделителя выключен)

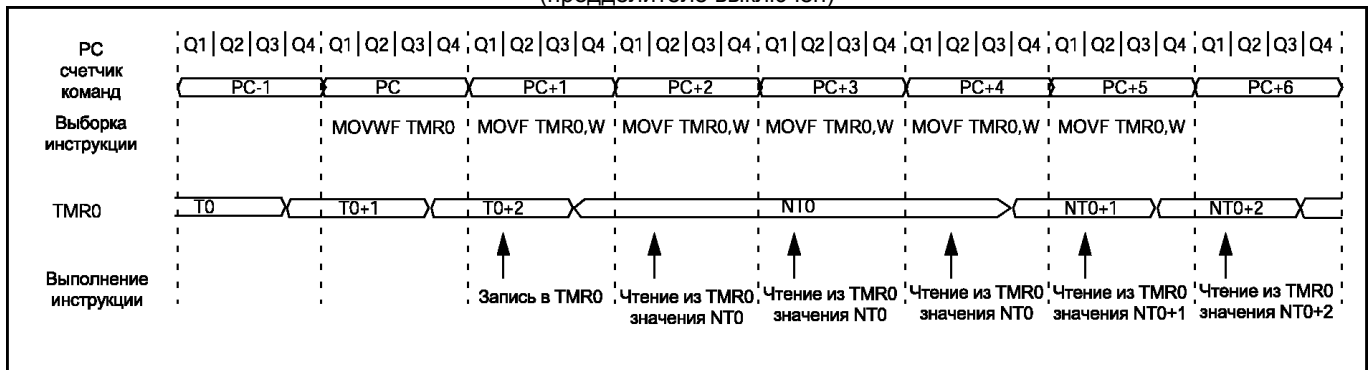
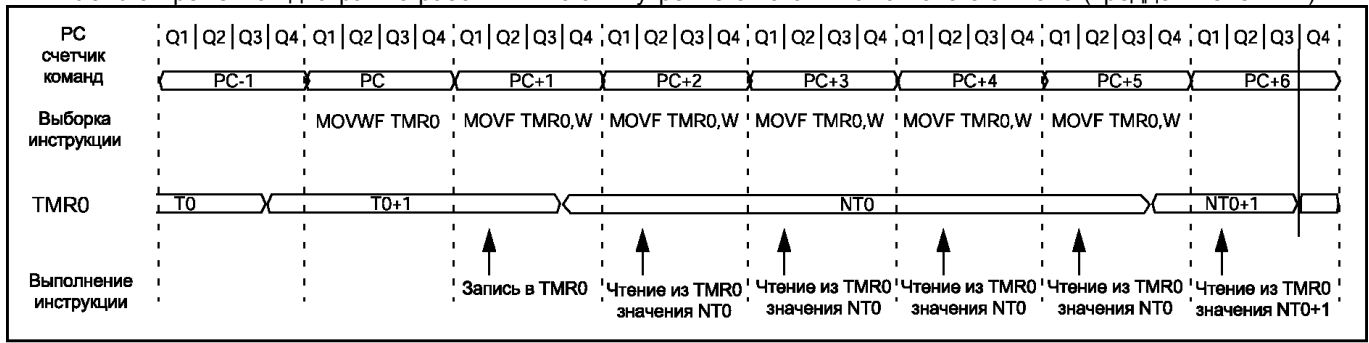


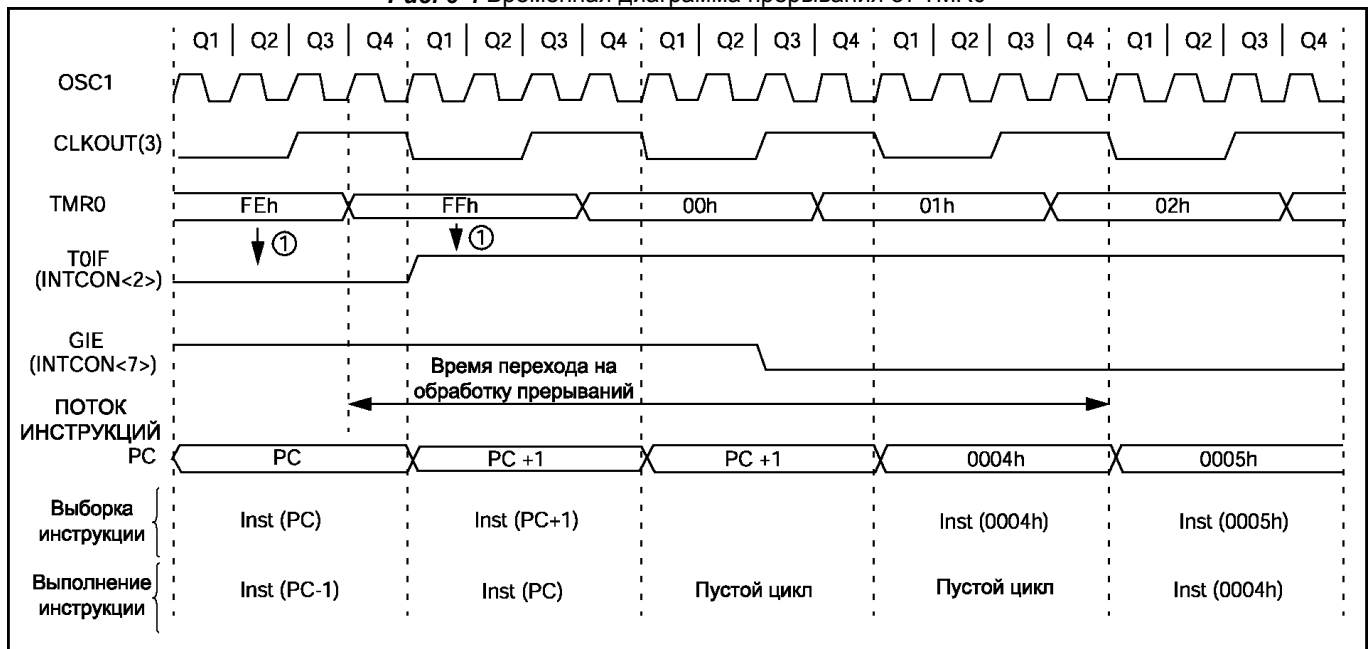
Рис. 6-3 Временная диаграмма работы TMR0 от внутреннего источника тактового сигнала (предделителем 1:2)



6.1 Прерывания от TMR0

Прерывания от TMR0 возникают при переполнении счетчика, т.е. при переходе его значения от FFh к 00h. При возникновении прерывания устанавливается в '1' бит T0IF. Само прерывание может быть разрешено/запрещено установкой/сбросом бита T0IE в регистре INTCON<5>. Флаг прерывания от TMR0 T0IF (INTCON<2>) должен быть сброшен в подпрограмме обработки прерываний. В SLEEP режиме микроконтроллера модуль TMR0 выключен и не может генерировать прерывания. На рисунке 6-4 показана временная диаграмма возникновения прерывания от TMR0.

Рис. 6-4 Временная диаграмма прерывания от TMR0



Примечания.

1. Проверка установки флага T0IF выполняется в каждом цикле Q1.
2. Время перехода на обработку прерывания 3T_{CY}, где T_{CY} – машинный цикл.
3. CLKOUT доступен только ER и INTRC режиме тактового генератора.

6.2 Использование внешнего источника тактового сигнала для TMR0

При использовании внешнего тактового сигнала для TMR0 необходимо учитывать некоторые детали работы таймера. Активный фронт внешнего тактового сигнала синхронизируется с внутренней тактовой частотой микроконтроллера, из-за чего возникает задержка от получения активного фронта сигнала до приращения TMR0.

6.2.1 Синхронизация внешнего сигнала

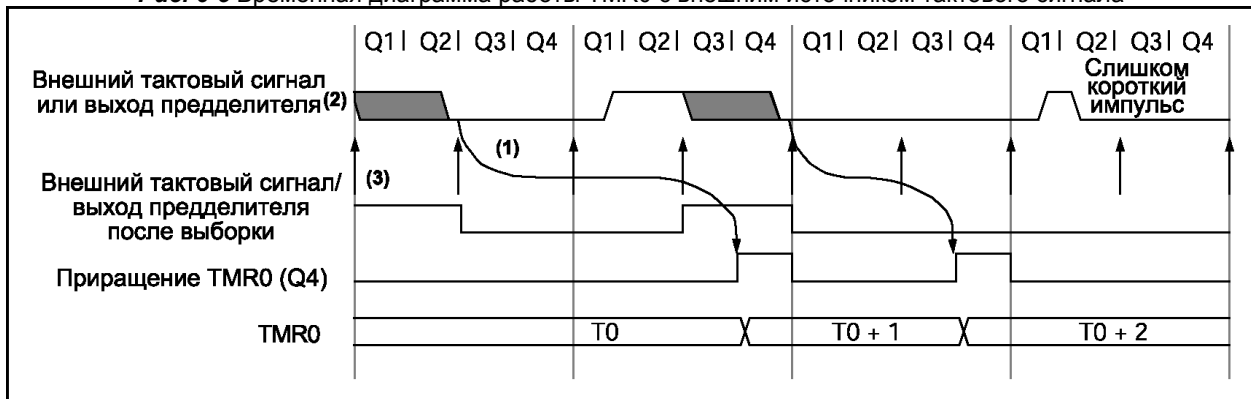
Если предделитель не используется, внешний тактовый сигнал поступает непосредственно на синхронизатор. Синхронизация T0CKI с таковым сигналом микроконтроллера усложняется из-за опроса выхода синхронизатора в машинные циклы Q2 и Q4 (см. рисунок 6-5). Поэтому длительность высокого или низкого логического уровня внешнего сигнала должна быть не меньше $2T_{OSC}$ (плюс небольшая задержка внутренней RC цепи 20нс). Дополнительную информацию смотрите в разделе электрических характеристик.

Если предделитель включен перед TMR0, то на вход синхронизатора поступает сигнал с асинхронного предделителя. Период сигнала T0CKI должен быть не менее $4T_{OSC}$ (плюс небольшая задержка внутренней RC цепи 40нс) деленное на коэффициент предделителя. Дополнительное требование, высокий и низкий логический уровень внешнего сигнала должен быть не менее 10нс . Смотрите параметры 40, 41 и 42 в разделе электрических характеристик.

6.2.2 Задержка приращения TMR0

Поскольку сигнал с выхода предделителя синхронизируется с внутренним тактовым сигналом микроконтроллера, возникает задержка от получения активного фронта сигнала до приращения TMR0 (см. рисунок 6-5).

Рис. 6-5 Временная диаграмма работы TMR0 с внешним источником тактового сигнала



Примечания:

1. Задержка от активного фронта тактового сигнала до приращения TMR0 от $3T_{OSC}$ до $7T_{OSC}$. Следовательно, максимальная ошибка измерения интервала между двумя активными фронтами тактового сигнала $\pm 4T_{OSC}$.
2. Если предделитель выключен, на вход синхронизатора поступает внешний тактовый сигнал.
3. Стрелками указаны точки выборки уровня сигнала.

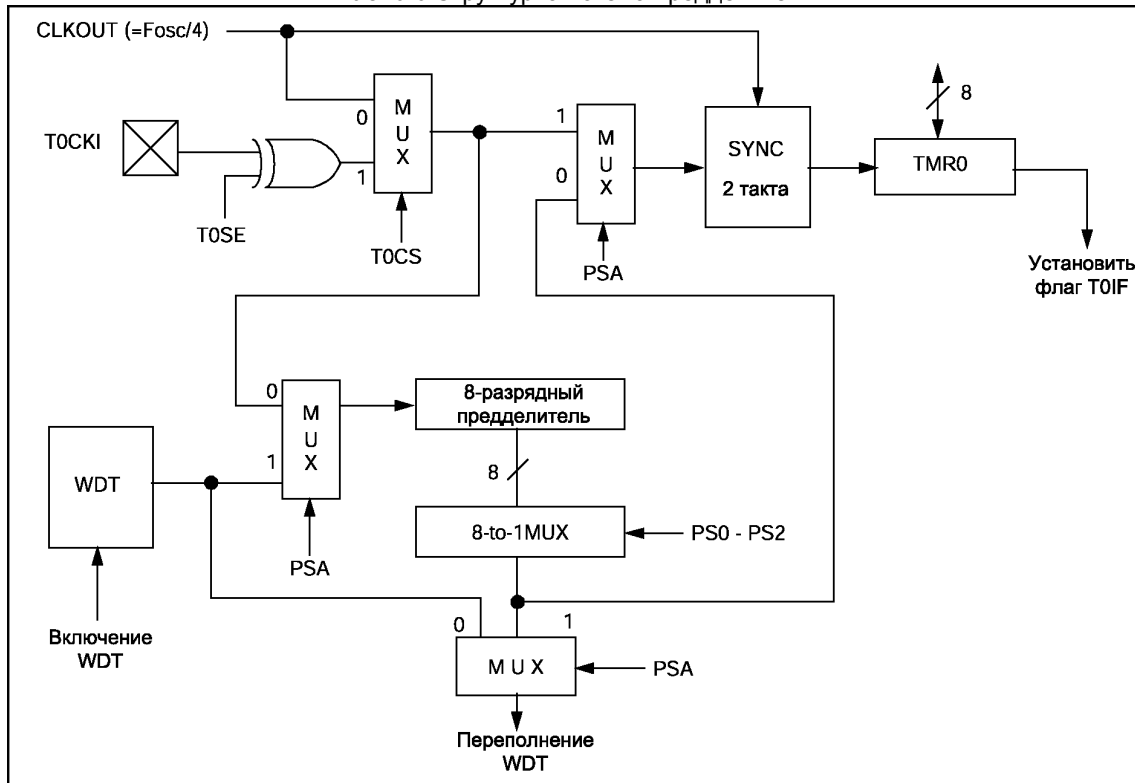
6.3 Предделитель

8-разрядный счетчик может работать как предделитель TMR0 или выходной делитель WDT (см. рисунок 6-6). Для простоты описания этот счетчик всегда будем называть «предделитель». Обратите внимание, что существует только один предделитель, который может быть включен перед TMR0 или WDT. Использование предделителя перед TMR0 означает, что WDT работает без предделителя, и наоборот.

Коэффициент деления предделителя определяется битами PSA и PS2:PS0 в регистре OPTION<3:0>.

Если предделитель включен перед TMR0, любые команды записи в TMR0 (например, CLRF 1, MOVWF 1, BSF 1,x и т.д.) сбрасывают предделитель. Когда предделитель подключен к WDT, команда CLRWDT сбросит предделитель вместе с WDT. Предделитель также очищается при сбросе микроконтроллера.

Рис. 6-6 Структурная схема предделителя



Примечание. Биты управления T0CS, T0SE, PS2, PS1, PS0, PSA расположены в регистре OPTION.

6.3.1 Переключение предделителя

Переключение предделителя выполняется программным способом, т.е. переключение можно сделать во время выполнения программы. В примере 6-1 показана рекомендуемая последовательность инструкций переключения предделителя от TMR0 на WDT для предотвращения неожиданного сброса микроконтроллера.

Переключение предделителя от WDT на TMR0 показано в примере 6-2. Меры осторожности должны применяться, даже если сторожевой таймер WDT выключен.

Пример 6-1 Переключения предделителя от TMR0 к WDT

- | | | |
|-----------|---------------|---|
| 1. BCF | STSTATUS, PRO | ; Выбрать банк 0 |
| 2. CLRWDT | | ; Сбросить WDT |
| 3. CLRF | TMR0 | ; Сбросить TMR0 и предделитель |
| 4. BSF | STATUS, PRO | ; Выбрать банк 1 |
| 5. MOVLW | b'00101111' | ; Три строки (5, 6, 7) должны быть включены в |
| 6. MOVWF | OPTION_REG | ; текст программы только, если биты |
| 7. CLRWDT | | ; PS<2:0> равны значению 000 или 001 |
| 8. MOVLW | b'00101xxx' | ; Переключить предделитель на WDT, |
| 9. MOVWF | OPTION_REG | ; выбирать коэффициент деления |
| 10. BCF | STSTATUS, PRO | ; Выбрать банк 0 |

Пример 6-2 Переключения предделителя от WDT к TMR0

- | | | |
|--------|---------------|---|
| CLRWDT | | ; Сбросить WDT и предделитель |
| BSF | STATUS, PRO | ; Выбрать банк 1 |
| MOVLW | b'xxxx0xxx' | ; Включить предделитель перед TMR0 и |
| MOVWF | OPTION_REG | ; выбрать новое значение коэффициента деления |
| BCF | STSTATUS, PRO | ; Выбрать банк 0 |

Таблица 6-1 Регистры и биты связанные с работой TMR0

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|-------------------|--------|-------|--------|--------|--------|--------|--------|-----------|---------------|
| 01h | TMR0 | Регистр таймера 0 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 81h | OPTION | -RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 85h | TRISA | TRISA7 | TRISA6 | - | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 11-1 1111 | 11-1 1111 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу TMR0.

7.0 Модуль таймера TMR1

TMR1 – 16-разрядный таймер/счетчик, состоящий из двух 8-разрядных регистров (TMR1H и TMR1L) доступных для чтения и записи. Счет выполняется в спаренных регистрах (TMR1H : TMR1L), инкрементируя их значение от 0000h до FFFFh, далее считает с 0000h. При переполнении счетчика устанавливается в '1' флаг прерывания TMR1IF в регистре PIR1<0>. Само прерывание можно разрешить/запретить установкой/сбросом бита TMR1IE в регистре PIE1<0>.

TMR1 может работать в двух режимах:

- таймер;
- счетчик.

Включение модуля TMR1 осуществляется установкой бита TMR1ON в '1' (T1CON<0>).

Битом TMR1CS (T1CON<1>) выбирается источник тактовых импульсов. В режиме таймера TMR1 инкрементируется на каждом машинном цикле. Если TMR1 работает с внешним источником тактового сигнала, то приращение происходит по каждому переднему фронту сигнала.

TMR1 имеет внутренний вход сброса от CPP модуля (см. раздел 10.0).

Для микроконтроллеров PIC16F627/628, когда включен генератор тактовых импульсов (T1OSCEN=1) выходы RB7/T1OSI и RB6/T1OSO/T1CKI настроены как входы. Значение битов TRISB<7:6> игнорируется.

Управляющие биты TMR1 находятся в регистре T1CON.

Регистр T1CON (адрес 10h)

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |
|-------|-----|---------|---------|---------|--------|--------|--------|-------|
| - | - | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | |
| Бит 7 | | | | | | | | Бит 0 |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

биты 7-6: **Не реализованы:** читаются как '0'

бит 5-4: **T1CKPS1:T1CKPS0:** Выбор коэффициента деления делителя TMR1
11 = 1:8
10 = 1:4
01 = 1:2
00 = 1:1

бит 3: **T1OSCEN:** Включение внутреннего тактового генератора TMR1
1 = генератор включен
0 = генератор выключен
Примечание. Инвертирующий элемент и резистивная обратная связь выключены для уменьшения тока потребления.

бит 2: **T1SYNC:** Синхронизация внешнего тактового сигнала

TMR1CS = 1
1 = не синхронизировать внешний тактовый
0 = синхронизировать внешний тактовый

TMR1CS = 0
Значение бита игнорируется

бит 1: **TMR1CS:** Выбор источника тактового сигнала
1 = внешний источник с вывода RB6/T1OSO/T1CKI
0 = внутренний источник Fosc/4

бит 0: **TMR1ON:** Включение модуля TMR1
1 = включен
0 = выключен

7.1 Работа TMR1 с внутренним тактовым сигналом

Приращение таймера происходит от внутреннего сигнала $F_{osc}/4$, когда бит TMR1CS (T1CON<1>) сброшен в '0'. В этом режиме бит синхронизации T1SYNC (T1CON<2>) игнорируется, потому что внутренний тактовый сигнал всегда синхронизирован.

7.2 Работа TMR1 с внешним синхронизированным тактовым сигналом

Работа TMR1 от внешнего источника тактового сигнала выбирается установкой бита TMR1CS в '1'. В этом режиме приращение таймера происходит по каждому переднему фронту сигнала на выводе RB7/T1OSI (если T1OSCEN=1) или RB6/T1OSO/T1CKI (если T1OSCEN=0).

Если T1SYNC=0, то активный фронт внешнего тактового сигнала синхронизируется с внутренним тактовым сигналом на выходе асинхронного делителя.

В SLEEP режиме микроконтроллера счетчик не будет инкрементироваться (при наличии тактового сигнала), т.к. синхронизатор выключен (делитель продолжает счет тактовых импульсов).

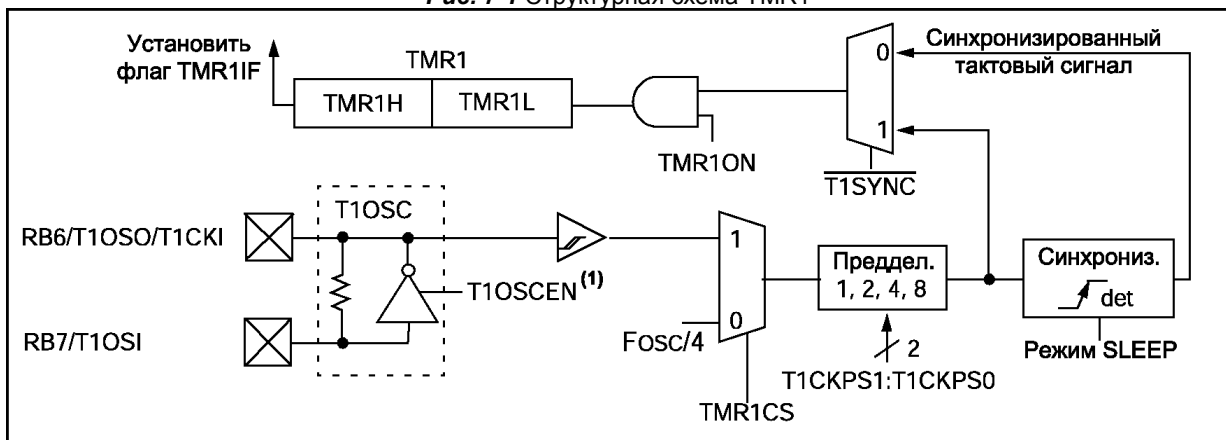
7.2.1 Синхронизация внешнего тактового сигнала

Когда используется синхронизация внешнего тактового сигнала, необходимо учитывать ряд требований. Фаза внешнего сигнала синхронизируется с внутренним тактовым сигналом микроконтроллера (период T_{osc}), из-за чего возникает задержка от получения активного фронта сигнала до приращения TMR1.

Если коэффициент делителя 1:1, то внешний тактовый сигнал поступает непосредственно на вход синхронизатора. Синхронизация T1CKI с тактовым сигналом микроконтроллера усложняется из-за опроса выхода синхронизатора в машинные циклы Q2 и Q4. Поэтому длительность высокого или низкого логического уровня внешнего сигнала должна быть не меньше $2T_{osc}$ (плюс небольшая задержка внутренней RC цепи 20нс). Дополнительную информацию смотрите в разделе электрических характеристик (параметры 45, 46 и 47).

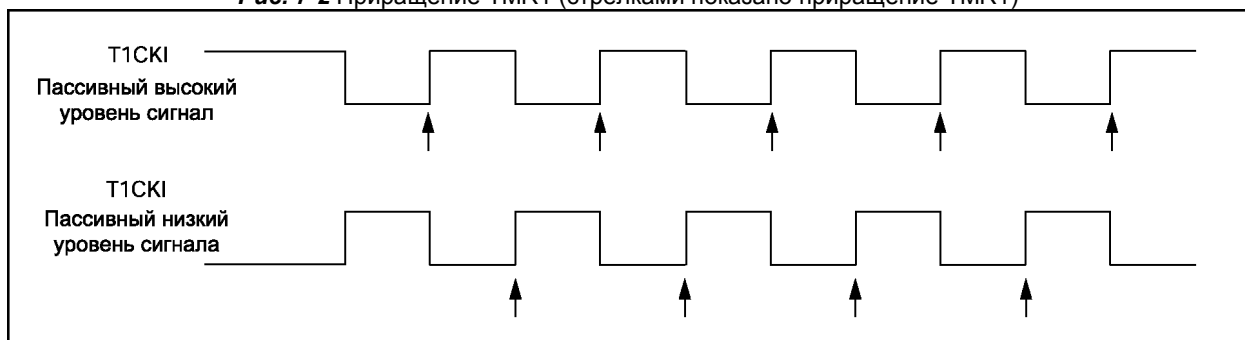
Если делитель имеет коэффициент деления отличный от 1:1, то на вход синхронизатора поступает сигнал с асинхронного делителя. Период сигнала T1CKI должен быть не менее $4T_{osc}$ (плюс небольшая задержка внутренней RC цепи 40нс) деленное на коэффициент делителя. Дополнительное требование, высокий и низкий логический уровень внешнего сигнала должен быть не менее 10нс. Смотрите параметры 40, 42, 45, 46 и 47 в разделе электрических характеристик.

Рис. 7-1 Структурная схема TMR1



Примечание. Если T1OSCEN=0, то инвертирующий элемент и резистивная обратная связь выключены для уменьшения тока потребления.

Рис. 7-2 Приращение TMR1 (стрелками показано приращение TMR1)



7.3 Работа TMR1 с внешним не синхронизированным тактовым сигналом

Если бит T1SYNC (T1CON<2>) установлен в '1', внешний тактовый сигнал TMR1 не будет синхронизироваться с внутренним тактовым сигналом микроконтроллера, таймер продолжает работать в режиме SLEEP микроконтроллера. Переполнение таймера вызовет «пробуждение» микроконтроллера, если разрешено прерывание от TMR1. Однако требуется осторожность при записи/чтении TMR1 (см. раздел 7.3.2).

В этом режиме TMR1 не может использоваться для отсчета времени сбора данных и операций сравнения.

7.3.1 Параметры внешнего не синхронизированного тактового сигнала

Если бит T1SYNC=1, то приращение таймера выполняется асинхронно. Минимальная длительность логических уровней внешнего тактового сигнала смотрите в разделе электрических характеристик (параметры 45, 46 и 47).

7.3.2 Чтение/запись TMR1 в асинхронном режиме

Чтение TMR1H или TMR1L, во время счета в асинхронном режиме, гарантирует получение текущего значения счетчика (реализовано аппаратно). Однако пользователь должен иметь в виду, что чтение 16-разрядного значения выполняется по байтно. Это накладывает некоторые ограничения, т.к. таймер может переполниться между чтениями байт.

Запись в TMR1 рекомендуется выполнять после остановки таймера. Запись в регистры TMR1 во время приращения таймера может привести к непредсказуемому значению регистра.

Пример 7-1 Подпрограмма чтения 16-разрядного значения таймера для случая, когда TMR1 не может быть остановлен.

```

; Выключить все прерывания
MOVF    TMR1H, W    ; Чтение старшего байта
MOVWF   TMPH        ;
MOVF    TMR1L, W    ; Чтение младшего байта
MOVWF   TMPL        ;
MOVF    TMR1H, W    ; Чтение старшего байта
SUBWF   TMPH, W     ; Сравнение с предыдущим чтением
BTFSC   STATUS, Z   ;
GOTO    CONTINUE    ; 16-разрядное значение прочитано правильно

; Возможно между чтениями байтов произошло
; переполнение таймера
; Прочитать значения заново
MOVF    TMR1H, W    ; Чтение старшего байта
MOVWF   TMPH        ;
MOVF    TMR1L, W    ; Чтение младшего байта
MOVWF   TMPL        ;
CONTINUE:           ;
; Включить прерывания

```

7.4 Генератор TMR1

Резонатор подключается к выводам T1OSI (вход) и T1OSO (выход усилителя). Максимальная частота резонатора 200кГц. Тактовый генератор TMR1 (идентичный LP генератору) в основном предназначен для кварцевого резонатора 32кГц. Включение генератора производится установкой бита T1OSEN в регистре T1CON<3>, что позволяет работать TMR1 в SLEEP режиме микроконтроллера.

Пользователь должен обеспечить программную задержку, чтобы гарантировать надлежащий запуск генератора. В таблицы 7-1 указаны рекомендуемые значения конденсаторов для генератора TMR1.

Таблица 7-1 Выбор конденсаторов для генератора TMR1

| Тип генератора | Частота | C1 | C2 |
|---------------------------------|---------|-------|-------|
| LP | 32 кГц | 33 пФ | 33 пФ |
| | 100 кГц | 15 пФ | 15 пФ |
| | 200 кГц | 15 пФ | 15 пФ |
| Ориентировочные значения | | | |

7.5 Сброс TMR1 триггером модуля CCP

Если модуль CCP работает в режиме сравнения с триггером специальных функций (CCP1M3 : CCP1M0=1011), то сигнал триггера сбросит TMR1.

Примечание. Сигнал с триггера специальных функций модуля CCP1 не будет устанавливать флаг TMRIF (PIR<0>) в '1'.

TMR1 должен работать в режиме синхронизированного внешнего тактового сигнала или внутреннего тактового сигнала. В асинхронном режиме эта функция не работает.

Когда запись в TMR1 совпадает с сигналом сброса от триггера специальных событий, приоритет отдается записи в TMR1.

В этом режиме модуля CCP период сброса TMR1 сохраняется в регистрах CCPR1H:CCPR1L.

7.6 Сброс регистров TMR1 (TMR1H, TMR1L)

Регистры TMR1H и TMR1L не сбрасываются в 00h при сбросе по включению питания POR и других видах сброса, кроме сброса по сигналу триггера специальных событий модуля CCP.

Регистр T1CON сбрасывается в 00h при сбросе POR и BOR (TMR1 выключается, коэффициент делителя равен 1:1). При всех остальных видах сброса значение регистра T1CON не изменяется.

7.7 Делитель TMR1

Делитель TMR1 очищается при записи в регистр TMR1L или TMR1H.

Таблица 7-2 Регистры и биты связанные с работой TMR1

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|--------------------------------------|-------|---------|---------|---------|--------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 0Eh | TMR1L | Младший байт 16-разрядного таймера 1 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Старший байт 16-разрядного таймера 1 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | - | - | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу TMR1.

8.0 Модуль таймера TMR2

TMR2 – 8-разрядный таймер с программируемыми предделителем и выходным делителем, 8-разрядным регистром периода PR2. TMR2 может быть опорным таймером для CCP модуля в ШИМ режиме. Регистры TMR2 доступны для записи/чтения и очищаются при любом виде сброса.

Входной тактовый сигнал ($F_{osc}/4$) поступает через предделитель с программируемым коэффициентом деления (1:1, 1:4 или 1:16), определяемый битами T2CKPS1:T2CKPS0 (T2CON<1:0>).

TMR2 считает инкрементируя от 00h до значения в регистре PR2, затем сбрасывается в 00h на следующем машинном цикле. Регистр PR2 доступен для записи и чтения. После сброса значение регистра PR2 равно FFh.

Сигнал переполнения TMR2 проходит через выходной 4-разрядный делитель с программируемым коэффициентом деления (от 1:1 до 1:16 включительно) для установки флага TMR2IF в регистре PIR1<1>.

Для уменьшения энергопотребления таймер TMR2 может быть выключен сбросом бита TMR2ON (T2CON<2>) в '0'.

8.1 Предделитель и выходной делитель TMR2

Счетчик предделителя и выходного делителя сбрасываются в случае:

- записи в регистр TMR2;
- записи в регистр T2CON;
- любого вида сброса микроконтроллера (POR, BOR, сброс WDT или активный сигнал -MCLR).

Регистр TMR2 не очищается при записи в T2CON.

8.2 Сигнал TMR2

Сигнал переполнения TMR2 (до выходного предделителя) поступает в модуль SSP для управления скоростью передачи данных (в микроконтроллерах PIC16F62X модуля SSP нет).

Регистр T2CON (адрес 12h)

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-------|---------|---------|---------|---------|--------|---------|---------|
| - | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| Бит 7 | | | | | | | Бит 0 |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

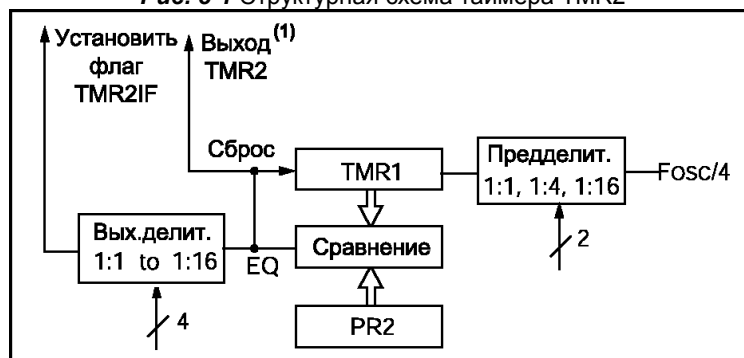
бит 7: **Не реализован:** читается как '0'

биты 6-3: **TOUTPS3:TOUTPS0:** Выбор коэффициента выходного делителя TMR2
0000 = 1:1
0001 = 1:2
:
:
1111 = 1:16

бит 2: **TMR2ON:** Включение модуля TMR2
1 = включен
0 = выключен

биты 1-0: **T2CKPS1:T2CKPS0:** Выбор коэффициента деления предделителя TMR2
00 = 1:1
01 = 1:4
1x = 1:16

Рис. 8-1 Структурная схема таймера TMR2



Примечание 1. TMR2 может использоваться для программного выбора скорости обмена данными модуля SSP.

Таблица 8-1 Регистры и биты связанные с работой TMR2

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|---------------------------|---------|---------|---------|---------|--------|---------|---------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 11h | TMR2 | Регистр таймера 2 | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | - | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -uuu uuuu |
| 92h | PR2 | Регистр периода таймера 2 | | | | | | | | 1111 1111 | 1111 1111 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу TMR2.

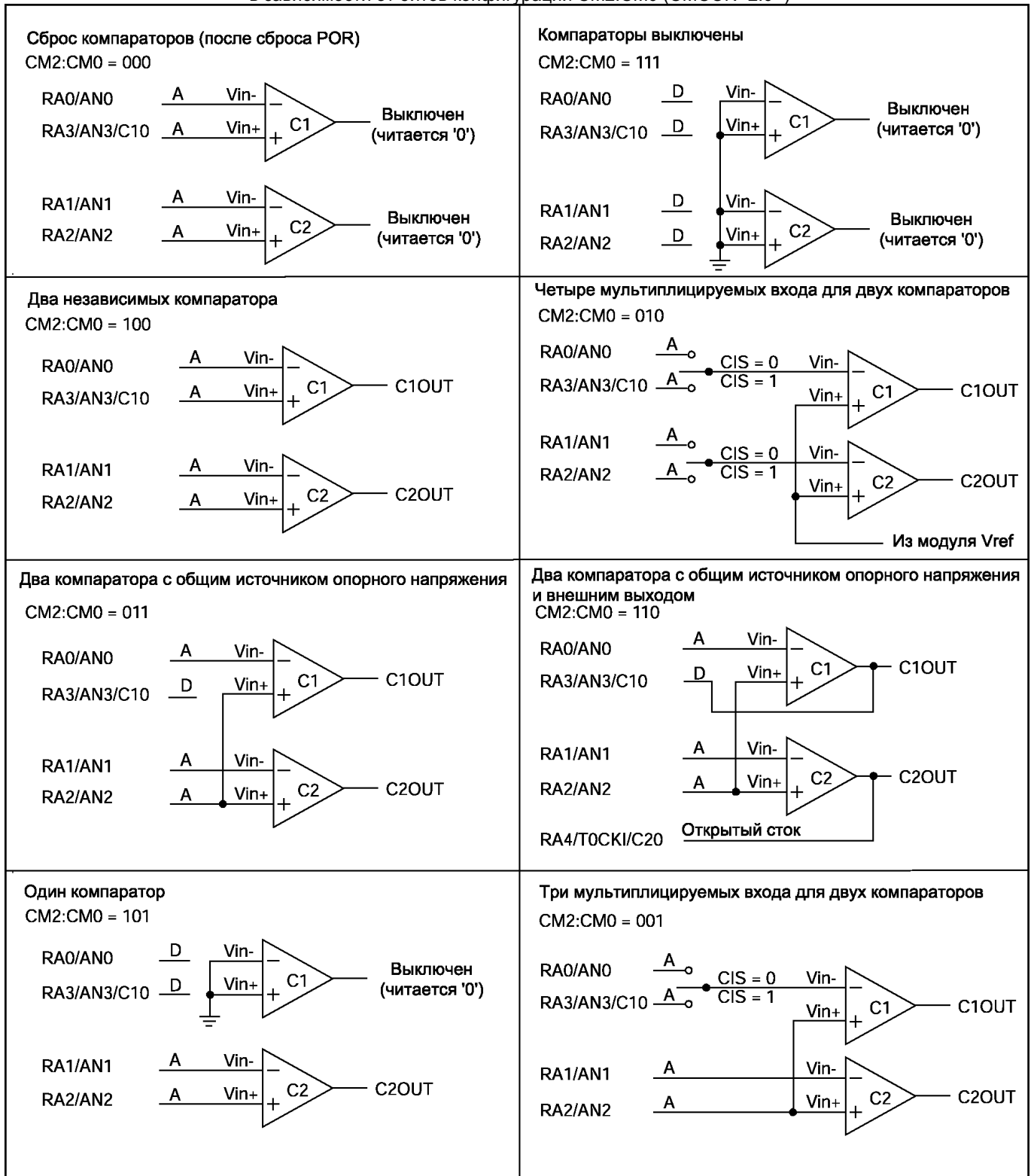
9.0 Модуль компараторов

Модуль компараторов содержит два аналоговых компаратора, выходы которых мультиплицированы с каналами ввода/вывода RA0 и RA3. Выход интегрированного источника опорного напряжения может быть подключен на вход компараторов. В регистре CMCON находятся биты управления модулем компараторов. Структурная схема модуля компараторов показана на рисунке 9-1.

Регистр CMCON (адрес 1Fh)

| R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|--|---|-------|-------|-------|-------|-------|-------|
| C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 |
| Бит 7 | | | | | | | Бит 0 |
| <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> R – чтение бита W – запись бита U – не реализовано, читается как 0 -n – значение после POR -x – неизвестное значение после POR </div> | | | | | | | |
| бит 7: | C2OUT: Выход компаратора 2 <u>Если C2INV=0</u> 1 = C2 $V_{IN+} > C2 V_{IN-}$ 0 = C2 $V_{IN+} < C2 V_{IN-}$ <u>Если C2INV=1</u> 0 = C2 $V_{IN+} > C2 V_{IN-}$ 1 = C2 $V_{IN+} < C2 V_{IN-}$ | | | | | | |
| бит 6: | C1OUT: Выход компаратора 1 <u>Если C1INV=0</u> 1 = C1 $V_{IN+} > C1 V_{IN-}$ 0 = C1 $V_{IN+} < C1 V_{IN-}$ <u>Если C1INV=1</u> 0 = C1 $V_{IN+} > C1 V_{IN-}$ 1 = C1 $V_{IN+} < C1 V_{IN-}$ | | | | | | |
| бит 5: | C2INV: Инверсный выход компаратора 2 1 = C2 инверсный выход 0 = C2 не инверсный выход | | | | | | |
| бит 4: | C1INV: Инверсный выход компаратора 1 1 = C1 инверсный выход 0 = C1 не инверсный выход | | | | | | |
| бит 3: | CIS: Подключение входов компараторов <u>Если CM2:CM3 = 001</u> 1 = C1 V_{IN-} подключен к RA3 0 = C1 V_{IN-} подключен к RA0 <u>Если CM2:CM3 = 010</u> 1 = C1 V_{IN-} подключен к RA3 C2 V_{IN-} подключен к RA2 0 = C1 V_{IN-} подключен к RA0 C2 V_{IN-} подключен к RA1 | | | | | | |
| биты 2-0: | CM2:CM0: Режим работы компараторов Смотрите рисунок 9-1. | | | | | | |

Рис. 9-1 Структурная схема модуля компараторов в зависимости от битов конфигурации CM2:CM0 (CMCON<2:0>)



Обозначения:

A = аналоговых вход, канал ввода/вывода читается как '0'; D = цифровой вход;
CIS = управляющий бит регистра CMCON<3>

9.1 Настройка модуля компараторов

Существует восемь режимов работы модуля компараторов, устанавливаемые битами CM2:CM0 (смотрите рисунок 9-1). Биты регистра TRISA управляют направлением каналов ввода/вывода для каждого режима модуля компараторов. При изменении режима работы модуля компараторов, параметры указанные в таблице 17-1 могут не соблюдаться.

Примечание. Для предотвращения ложных прерываний рекомендуется запретить прерывания от модуля компараторов, а затем изменить режим его работы.

В примере 9-1 показана настройка модуля компараторов (RA3, RA4 – цифровые выходы; RA0, RA1 – входы компараторов V-; RA2 – вход V+обоих компараторов).

Пример 9-1 Инициализация модуля компараторов

```
BCF          INTCON,GIE      ;Глобальное запрещение прерываний
BCF          INTCON,PEIE     ;Запретить прерывания от периферийных модулей
CLRF        PORTA           ;Инициализация PORTA
MOVLW      0x03             ;Установить режим модуля компараторов
MOVWF      CMCON            ;CM<2:0> = 011
BSF         STATUS,RP0      ;Выбрать банк 1
MOVLW      0x07             ;Направление выводов PORTA
MOVWF      TRISA            ;RA<2:0> входы
                                   ;RA<4:3> выходы
                                   ;TRISA<7:5> читаются как '0'

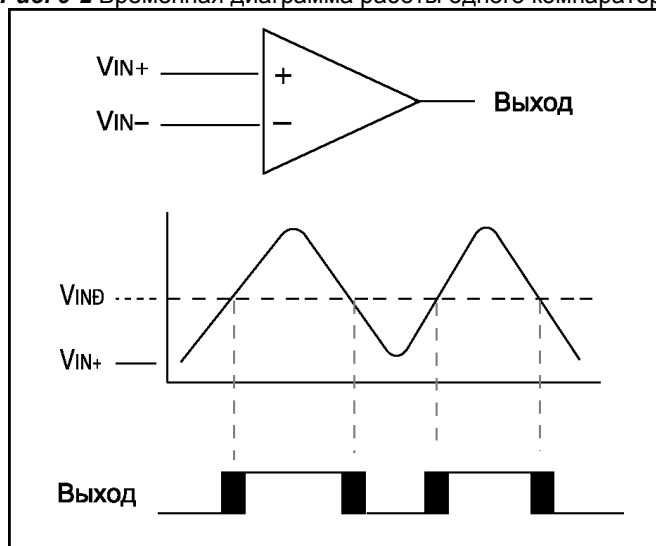
BCF         STATUS,RP0      ;Выбрать банк 0
CALL       DELAY_10         ;Задержка 10мкс
MOVF      CMCON,F           ;Чтение CMCON для устранения несоответствия
BCF         PIR1,CMIF       ;Сбросить флаг прерываний
BSF         STATUS,RP0      ;Выбрать банк 1
BSF         PIE1,CMIE       ;Разрешить прерывания от модуля компараторов
BCF         STATUS,RP0      ;Выбрать банк 0
BSF         INTCON,PEIE     ;Разрешить прерывания от периферийных модулей
BSF         INTCON,GIE      ;Глобальное разрешение прерываний
```

9.2 Работа модуля компараторов

Временная диаграмма работы одного компаратора показана на рисунке 9-2 (соотношение входных аналоговых сигналов и выходного цифрового сигнала).

Когда аналоговый сигнал на входе V_{IN+} меньше V_{IN-} , на цифровом выходе установлен логический нуль. Если сигнал на входе V_{IN+} больше V_{IN-} , то на цифровом выходе будет установлена логическая единица. Затененные области на рисунке 9-2 показывают неуверенный уровень цифрового сигнала.

Рис. 9-2 Временная диаграмма работы одного компаратора



9.3 Опорное напряжение для компараторов

Допускается использование внешнего или внутреннего источника опорного напряжения для компараторов (определяется режимом работы модуля компараторов). Аналоговый сигнал, присутствующий на выводе V_{IN-} сравнивается с сигналом V_{IN+} , по результатам сравнения формируется выходной цифровой сигнал (см. рисунок 9-2).

9.3.1 Внешний источник опорного напряжения

Модуль компараторов может быть настроен таким образом, что внешний источник опорного напряжения подключается на вход обоих компараторов или для каждого компаратора отдельный источник. Напряжение опорного источника должно быть в пределах от V_{SS} до V_{DD} .

9.3.2 Внутренний источник опорного напряжения

Модуль компараторов позволяет использовать внутренний источник опорного напряжения, описанный в разделе 11.0. Сигнал внутреннего источника опорного напряжения подключается к выводам V_{IN+} обоих компараторов, когда биты конфигурации $CM<2:0>=010$ (см. рисунок 9-1).

9.4 Время реакции компараторов

Время реакции – это гарантированная максимальная задержка изменения выходного цифрового сигнала после изменения входных сигналов. Если изменяется напряжение внутреннего опорного источника подключенного к компараторам, то должна рассматриваться задержка установки напряжения внутреннего опорного источника. Во всех остальных случаях используется максимальная задержка, указанная в таблице 17-1.

9.5 Прерывания от компараторов

Модуль компараторов устанавливает флаг прерывания $CMIF$ ($PIR1<6>$) в '1' при изменении уровня сигнала на выходе любого компаратора. Пользователь должен проверить, какой компаратор вызвал установку флага $CMIF$ чтением битов $CMCON<7:6>$. Флаг прерывания от компараторов $CMIF$ должен быть сброшен программно в '0'. Программой установкой бита $CMIF$ в '1' моделируется возникновение прерывания от модуля компараторов.

Примечание. Если изменения в регистре $CMCON$ (бит $C1OUT$ или $C2OUT$) произошло, когда выполнялась операция чтения (начало такта Q2), флаг прерывания $CMIF$ ($PIR1<6>$) может не установиться в '1'.

Биты $CMIE$ ($PIE1<6>$), $PEIE$ ($INTCON<6>$) и GIE ($INTCON<7>$) должны быть установлены в '1', чтобы разрешить генерацию прерывания от модуля компараторов. Если любой из битов сброшен в '0' прерывания не генерируются, но флаг $CMIF$ устанавливается в '1' при возникновении условия прерывания.

В подпрограмме обработки прерываний необходимо выполнить следующие действия:

1. Произвести запись или чтение регистра $CMCON$ для устранения условия несоответствия.
2. Сбросить флаг $CMIF$ в '0'.

Флаг $CMIF$ будет аппаратно устанавливаться в '1' до тех пор, пока не будет устранено условие несоответствия. Чтение регистра $CMCON$ устраним условие несоответствия и позволит сбросить флаг $CMIF$ в '0'.

9.6 Выходы компараторов

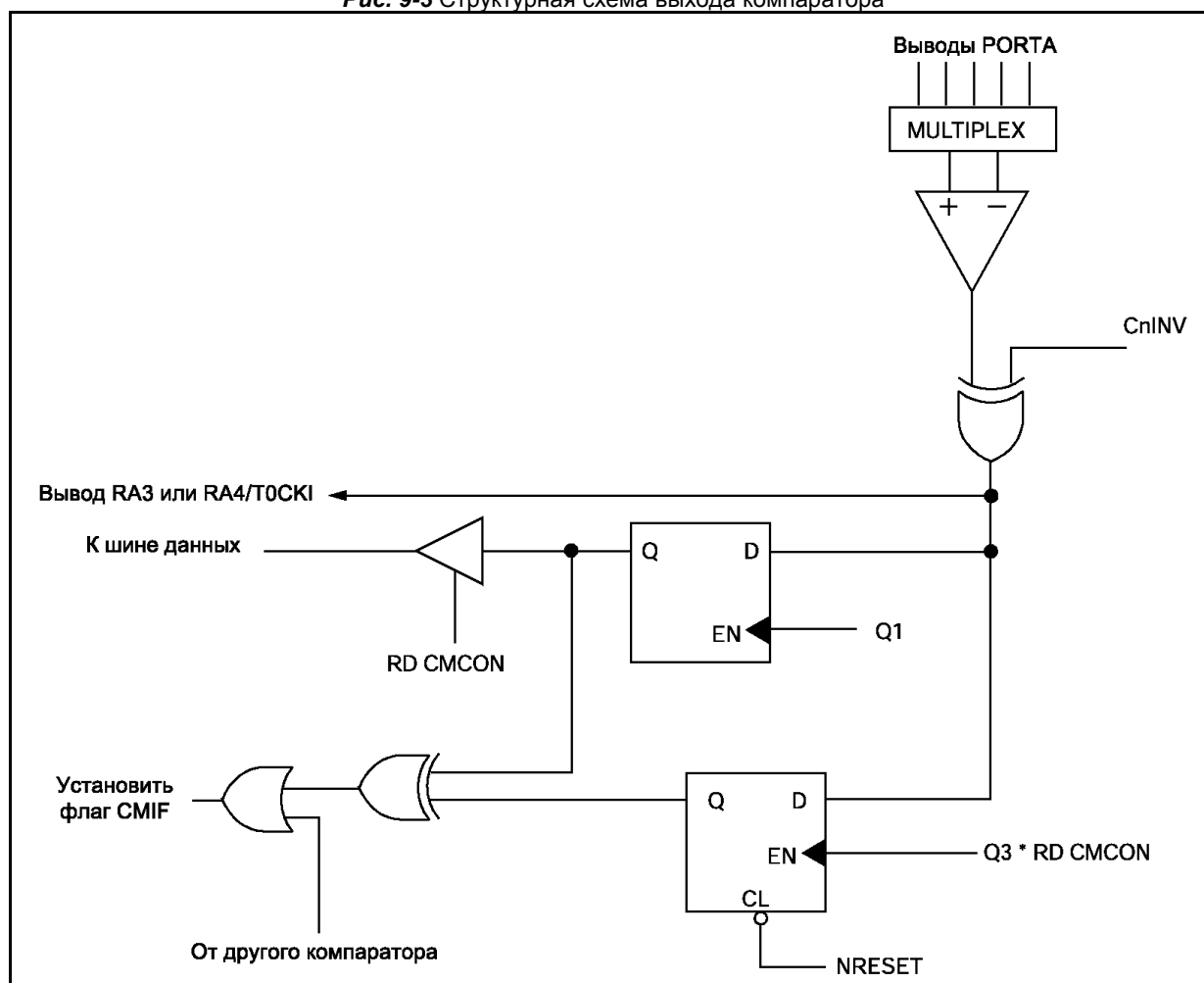
Состояние выходов компараторов можно прочесть в регистре CMCON (биты выходов компараторов доступны только на чтение). Выводы компараторов также могут быть подключены к каналам портов ввода/вывода RA3 и RA4 (CM<2:0> = 110 или 001). В этом режиме модуля компараторов выводы RA3 и RA4/T0CKI становятся несинхронизированными выходами компараторов. Зона неуверенного сигнала на выходе зависит от источника опорного напряжения и времени реакции компаратора (см. электрические характеристики). На рисунке 9-3 показана структурная схема выхода компаратора.

Соответствующие биты в регистре TRISA определяют подключать или нет вывод компараторов к выводам RA3 и RA4/T0CKI (если CM<2:0> = 110 или 001).

Примечания:

1. При чтении регистра PORTA, все выводы, настроенные как аналоговые, будут давать результат '0'. Выводы, настроенные как цифровые входы, конвертируют аналоговый сигнал согласно характеристикам входного триггера Шмидта.
2. Аналоговые уровни сигналов, подаваемые на цифровые входы, могут быть причиной повышенного энергопотребления.

Рис. 9-3 Структурная схема выхода компаратора



9.7 Работа модуля компараторов в режиме SLEEP

Если модуль компараторов включен, то при переходе микроконтроллера в режим SLEEP компараторы продолжают работать. Если прерывания от компараторов разрешены, то по возникновению прерывания микроконтроллер выйдет из режима SLEEP.

При включенных компараторах ток потребления микроконтроллера в режиме SLEEP несколько выше, чем указано в спецификации (каждый включенный компаратор потребляет дополнительный ток). Если в режиме SLEEP компараторы не используются, то рекомендуется их выключать (CM<2:0> = 111) перед переходом в режим SLEEP для уменьшения суммарного тока потребления.

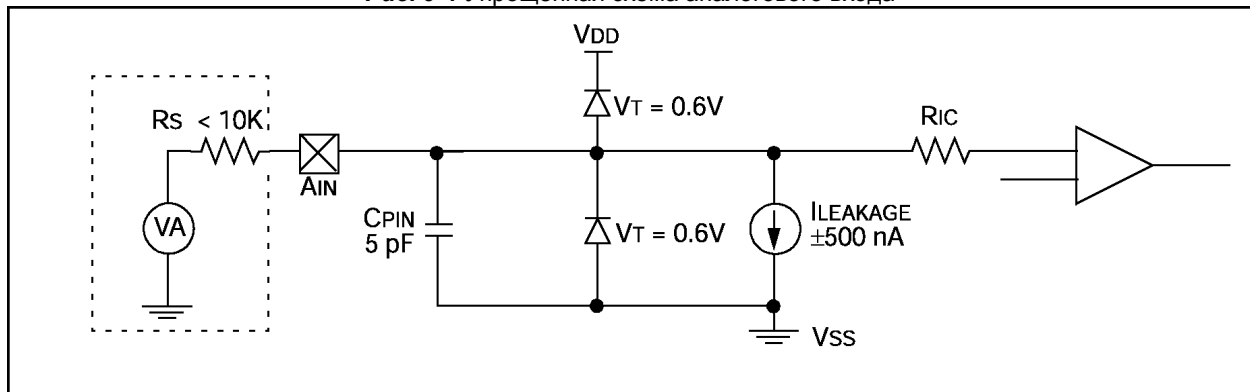
9.8 Эффект сброса

При любом виде сброса микроконтроллера все биты регистра CMCON сбрасываются в '0'. Сброс включает оба компаратора (CM2:CM0=000), делая все входы аналоговыми. Настройка каналов ввода/вывода как аналоговые входы при сбросе микроконтроллера позволяет минимизировать потребляемый ток.

9.9 Подключение к аналоговым входам

Упрощенная схема аналогового входа показана на рисунке 9-4. Т.к. аналоговые входы мультиплицированы с цифровыми входами, они имеют пару защитных диодов подключенных к V_{DD} и V_{SS}. Амплитуда аналогового сигнала должна быть в пределах от V_{SS} до V_{DD}. Амплитуда входного сигнала ограничивается в пределах от V_{SS}-0.6В до V_{DD}+0.6В. Внутреннее сопротивление источника аналогового сигнала должно быть меньше 10кОм. Компоненты, подключаемые к аналоговому входу (конденсатор, стабилитрон и т.д.), должны иметь минимальный ток утечки.

Рис. 9-4 Упрощенная схема аналогового входа



Обозначения:

- C_{PIн} – входная емкость;
- V_T – напряжение ограничения;
- I_{LEAKAGE} – ток утечки вывода;
- R_{IC} – сопротивление соединения;
- R_S – сопротивление источника;
- VA – аналоговый сигнал.

Таблица 9-1 Регистры и биты связанные с работой модуля компараторов

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|--------|--------|-------|--------|--------|--------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 1Fh | CMCON | C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 0000 0000 | 0000 0000 |
| 9Fh | VRCON | VREN | VROE | VRR | - | VR3 | VR2 | VR1 | VR0 | 000- 0000 | 000- 0000 |
| 85h | TRISA | TRISA7 | TRISA6 | - | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 11-1 1111 | 11-1 1111 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу модуля компараторов.

10.0 CCP модуль (захват/сравнение/ШИМ)

Модуль CCP содержит 16-разрядный регистр, который может использоваться в качестве:

- 16-разрядного регистра захвата данных;
- 16-разрядного регистра сравнения;
- 8-разрядные (ведущий и ведомый) регистры ШИМ.

В таблице 10-1 показано использование таймеров в различных режимах модуля CCP.

Регистр CCPR1 модуля CCP состоит из двух 8-разрядных регистров: CCPR1L (младший байт), CCPR1H (старший байт). В регистре CCP1CON находятся управляющие биты модуля CCP1 доступные для записи и чтения. Дополнительную информацию по модулю CCP смотрите в документации DS33023 «PICmicro™ Mid-Range Reference Manual».

Таблица 10-1 Использование таймеров модулем CCP

| Режим модуля CCP | Таймер |
|------------------|--------|
| Захват | TMR1 |
| Сравнение | TMR1 |
| ШИМ | TMR2 |

Регистр CCP1CON (адрес 17h)

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |
|-------|-----|--------------|--------------|---------------|---------------|---------------|---------------|-------|
| - | - | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | |
| Бит 7 | | | | | | | | Бит 0 |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

бит 7-6: **Не используются:** читаются как '0'

биты 5-4: **CCP1X:CCP1Y:** Младшие биты скважности ШИМ
Режим захвата: не используются
Режим сравнения: не используются
Режим ШИМ: два младших бита скважности. Восемь старших находятся в CCPR1L.

биты 3-0: **CCP1M3:CCP1M0:** Режим работы модуля CCP1
0000 = модуль CCP1 выключен (сброс модуля CCP1)
0100 = захвата по каждому заднему фронту сигнала
0101 = захват по каждому переднему фронту сигнала
0110 = захват по каждому 4-му переднему фронту сигнала
0111 = захват по каждому 16-му переднему фронту сигнала
1000 = сравнение, устанавливает выходной сигнал (устанавливается флаг CCP1IF в '1')
1001 = сравнение, сбрасывает выходной сигнал (устанавливается флаг CCP1IF в '1')
1010 = сравнение, на выходной сигнал не влияет (устанавливается флаг CCP1IF в '1')
1011 = сравнение, триггер специальных функций (устанавливается флаг CCP1IF в '1'; CCP1 сбрасывает TMR1)
11xx = ШИМ режим

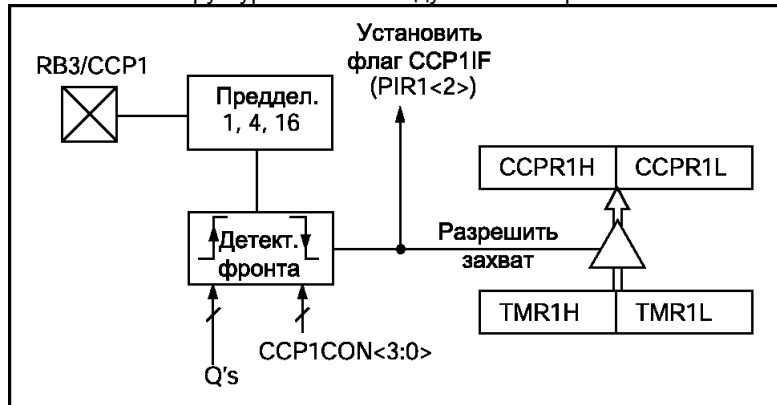
10.1 Режим захвата

При возникновении события захвата 16-разрядное значение счетчика TMR1 переписывается в регистры CCP1L:CCP1H модуля CCP1. Событием захвата может быть:

- каждый задний фронт сигнала на входе RB3/CCP1;
- каждый передний фронт сигнала на входе RB3/CCP1;
- каждый 4-й передний фронт сигнала на входе RB3/CCP1;
- каждый 16-й передний фронт сигнала на входе RB3/CCP1.

Тип события захвата устанавливается битами CCP1M3:CCP1M0 в регистре CCP1CON. После выполнения захвата устанавливается флаг прерывания CCP1IF (PIR1<2>) в '1', который должен быть сброшен программно. Если происходит события захвата до того как предыдущие данные были прочитаны, старое значение будет потеряно.

Рис. 10-1 Структурная схема модуля CCP1 в режиме захвата



10.1.1 Настройка вывода модуля CCP

Порт ввода/вывода RB3/CCP1 должен быть настроен на вход установкой бита TRISB<3> в '1'.

Примечание. Если канал порта ввода/вывода настроен на выход, то захват может происходить командой из программы.

10.1.2 Настройка таймера TMR1

TMR1 должен работать в синхронизированном режиме, в случае использования внешнего тактового сигнала с вывода RB7/T1OSI. В асинхронном режиме TMR1 модуль CCP1 работать не будет.

10.1.3 Обработка прерываний

Когда изменяется режим работы модуля CCP необходимо запрещать прерывания сбросом бита CCP1IE (PIE<2>) в '0' для предотвращения ложных прерываний. После изменение режима работы модуля CCP1, перед разрешением прерываний, необходимо сбросить флаг CCP1IF (PIR1<2>) в '0'.

10.1.4 Предварительный счетчик событий модуля CCP

Существует четыре режима работы предварительного счетчика событий (определяется битами CCP1M3:CCP1M0). Включение режима захвата очищает предварительный счетчик событий. Переключение между типами событий не очищает счетчик событий, поэтому результат первого захвата после переключения может быть недостоверным.

В примере 10-1 показано как нужно производить переключение типа события, чтобы не вызвать ложное срабатывание.

Пример 10-1 Переключение типа события

```

CLRf    CCP1CON    ; Выключить CCP модуль
MOVLW   NEW_CAPT_PS ; Записать W новый тип захвата и режим работы CCP
MOVWF   CCP1CON    ; Загрузить настройку в регистр CCP1CON

```

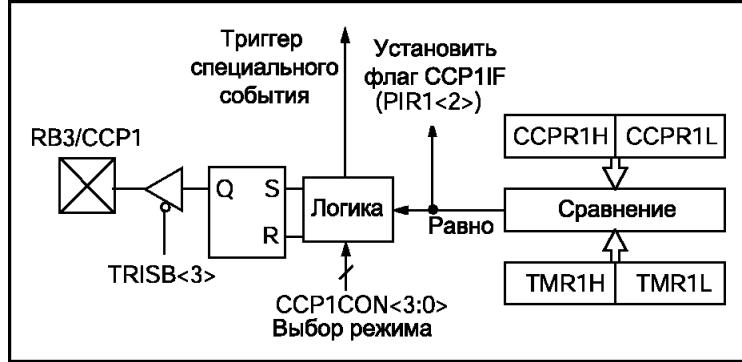
10.2 Режим сравнения

В этом режиме, 16-разрядный регистр CCP1 сравнивается со значением TMR1. Как только значения в регистрах становятся одинаковыми, модуль CCP1 изменяет состояние вывода RB3/CCP1:

- устанавливает высокий уровень сигнала;
- устанавливает низкий уровень сигнала;
- на вывод не воздействует.

Действие при совпадении может быть выбрано битами CCP1M3:CCP1M0 в регистре CCP1CON. В момент изменение состояния вывода устанавливается флаг прерывания CCP1IF в '1'.

Рис. 10-2 Структурная схема модуля CCP1 в режиме сравнения



10.2.1 Настройка вывода модуля CCP

Для изменения состояния вывода RB3/CCP1, он должен быть настроен на выход сбросом бита TRISB<3> в '0'.

Примечание. При очистке регистра CCP1CON на выводе RB3/CCP1 появится сигнал низкого уровня, что не является результатом сравнения.

10.2.2 Настройка таймера TMR1

TMR1 должен работать в синхронизированном режиме, в случае использования внешнего тактового сигнала с вывода RB7/T1OSI. В асинхронном режиме TMR1 модуль CCP1 работать не будет.

10.2.3 Обработка прерываний

Программное изменение уровня сигнала на выходе CCP1 не вызовет генерацию прерывания. Прерывание генерируются только модулем CCP1.

10.2.4 Триггер специального события

В режиме сравнения модуля CCP1 может быть включен триггер специального события.

Триггер специального события сбрасывает значения таймера TMR1 при каждом положительно выполненном сравнении. Регистр CCP1R является 16-разрядным программируемым регистром периода для TMR1.

Таблица 10-2 Регистры и биты связанные с работой модуля CCP1 в режиме захват/сравнение

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|---------|--------------------------------------|-------|---------|---------|---------|--------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 86h | TRISB | Регистр направления PORTB | | | | | | | | 1111 1111 | 1111 1111 |
| 0Eh | TMR1L | Младший байт 16-разрядного таймера 1 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Старший байт 16-разрядного таймера 1 | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | - | - | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 15h | CCPR1L | Младший байт захвата/сравнения/ШИМ | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Старший байт захвата/сравнения/ШИМ | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | - | - | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу модуля CCP1 в режиме захват/сравнение.

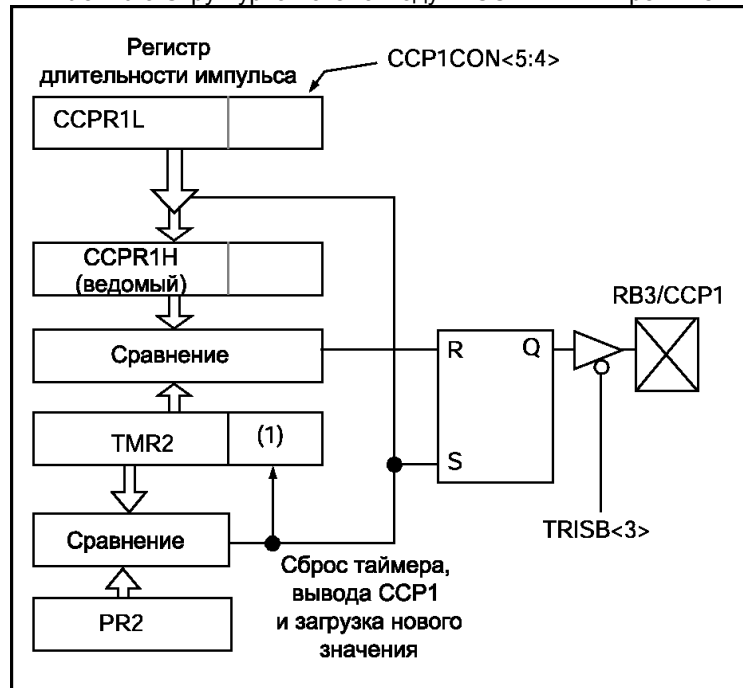
10.3 ШИМ режим

В ШИМ режиме модуля CCP1 вывод RB3/CCP1 используется в качестве выхода 10-разрядного ШИМ. Т.к. вход CCP1 мультиплицирован с цифровым каналом порта ввода/вывода, бит направления TRISB<3> должен быть сброшен в '0'.

Примечание. Очистка регистра CCP1CON вынудит перевести вывод CCP1 в низкий логический уровень. Низкий логический уровень не является данными из защелки PORTB.

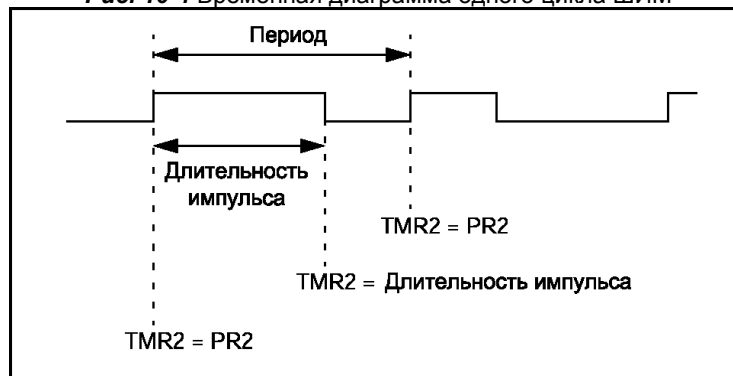
На рисунке 10-3 показана структурная схема модуля CCP1 в ШИМ режиме. Пошаговое описание настройки модуля CCP1 в ШИМ режиме смотрите в разделе 10.3.3.

Рис. 10-3 Структурная схема модуля CCP1 в ШИМ режиме



На рисунке 10-4 показана временная диаграмма одного цикла ШИМ (период ШИМ и длительность высокого уровня сигнала). Частота ШИМ есть обратная величина периоду (1/период).

Рис. 10-4 Временная диаграмма одного цикла ШИМ



10.3.1 Период ШИМ

Период ШИМ определяется значением в регистре PR2 и может быть вычислен по формуле:

$$\text{Период ШИМ} = [(PR2) + 1] \times 4 \times T_{osc} \times (\text{коэффициент делителя TMR2})$$

$$\text{Частота ШИМ} = 1 / \text{Период ШИМ}$$

Когда значение TMR2 сравнивается с PR2, выполняются следующие действия:

- TMR2 сбрасывается в 00h;
- устанавливается высокий уровень сигнал на выводе CCP1 (Если скважность равна 0%, то сигнал в высокий уровень устанавливаться не будет);
- модуль ШИМ начинает новый цикл, загружая значение из регистра CCPR1L в CCPR1H.

Примечание. Выходной делитель TMR2 (см. раздел 8.0) не влияет на частоту ШИМ. Он может использоваться для отсчета времени, когда необходимо изменить скважность ШИМ.

10.3.2 Сквозность ШИМ

Сквозность ШИМ определяется битами в регистрах CCP1L и CCP1CON<5:4>. Для 10-разрядного ШИМ старшие восемь бит сохраняются в регистре CCP1L, а младшие два бита в регистре CCPCON<5:4> (CCP1L:CCPCON<5:4>). Для вычисления длительности сигнала высокого уровня, воспользуйтесь следующей формулой:

$$\text{Длительность импульса ШИМ} = (\text{CCP1L:CCPCON}<5:4>) \times T_{osc} \times (\text{коэффициент делителя TMR2})$$

Биты в регистре CCP1L и CCP1CON<5:4> могут быть изменены в любое время, но значение в регистре CCP1H не изменяется, пока не произойдет соответствие PR2 и TMR2. В ШИМ режиме регистр CCP1H доступен только для чтения.

Регистр CCP1H и внутренняя двух разрядная защелка образуют буфер ШИМ. Эффект буферизации необходим при записи нового значения длительности импульса ШИМ.

Когда значение CCP1H и 2-разрядной внутренней защелки соответствует значению TMR2 и внутреннему 2-разрядному счетчику, в такте Q2 на выводе CCP1 будет установлен низкий уровень сигнала.

Расчет максимального разрешения ШИМ для данной частоты можно вычислить по формуле (бит):

$$\log\left(\frac{F_{osc}}{F_{pwm}}\right) = \frac{\log(2)}{\log(2)}$$

Примечание. Если длительность импульса ШИМ больше периода ШИМ, вывод CCP1 не будет иметь низкий уровень сигнала.

Примеры вычисления периода ШИМ, длительности импульса ШИМ и разрядность ШИМ смотрите в документации DS33023 «PICmicro™ Mid-Range Reference Manual».

10.3.3 Последовательность настройки модуля CCP в ШИМ режиме

Рекомендованная последовательность включения модуля CCP в ШИМ режиме:

1. Установить период ШИМ в регистре PR2;
2. Установить длительность импульса в регистрах CCP1L и CCP1CON <5:4>;
3. Настроить вывод CCP1 как выход, сбросив бит TRISC<2>;
4. Настроить делитель и включить TMR2 в регистре T2CON;
5. Включить CCP1 в режиме ШИМ.

Таблица 10-3 Соответствие частоты ШИМ и разрешения ШИМ при тактовой частоте микроконтроллера 20МГц

| Частота ШИМ | 1.22кГц | 4.88кГц | 19.53кГц | 78.12кГц | 156.3кГц | 208.3кГц |
|---------------------------|---------|---------|----------|----------|----------|----------|
| Коэффициент делителя TMR2 | 16 | 4 | 1 | 1 | 1 | 1 |
| Значение PR2 | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Разрешение ШИМ (бит) | 10 | 10 | 10 | 8 | 7 | 5.5 |

Таблица 10-4 Регистры и биты связанные с работой модуля CCP1 в ШИМ режиме

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|---------|------------------------------------|---------|---------|---------|---------|--------|---------|---------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 86h | TRISB | Регистр направления PORTB | | | | | | | | 1111 1111 | 1111 1111 |
| 11h | TMR2 | Регистр таймера 2 | | | | | | | | 0000 0000 | 0000 0000 |
| 92h | PR2 | Регистр периода таймера 2 | | | | | | | | 1111 1111 | 1111 1111 |
| 12h | T2CON | - | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -uuu uuuu |
| 15h | CCP1L | Младший байт захвата/сравнения/ШИМ | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCP1H | Старший байт захвата/сравнения/ШИМ | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | - | - | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу модуля CCP1 в ШИМ режиме.

11.0 Источник опорного напряжения

Модуль источника опорного напряжения содержит 16 последовательно включенных резисторов, обеспечивающие выбор нужного напряжения. Резисторы разделены на сегменты для организации двух диапазонов напряжений VREF и имеют возможность выключения для уменьшения тока потребления, когда источник опорного напряжения не используется.

В регистре VCON находятся биты управления источником опорного напряжения. На рисунке 11-1 показана структурная схема источника опорного напряжения.

11.1 Настройка источника опорного напряжения

Источник опорного напряжения может иметь 16 различных уровней напряжения для каждого диапазона. Уравнение вычисления напряжения:

$$\begin{aligned} \text{Если } VRR = 1: & \quad V_{REF} = (VR<3:0>/24) \times V_{DD} \\ \text{Если } VRR = 0: & \quad V_{REF} = (V_{DD} \times 1/4) + (VR<3:0>/32) \times V_{DD} \end{aligned}$$

Время установки напряжения должно определяться по напряжению на выводе VREF. В примере 11-1 показана настройка опорного источника на напряжения 1.25В при напряжении питания 5В.

Регистр VRCON (адрес 9Fh)

| | | | | | | | |
|-------------|-------------|------------|-----|------------|------------|------------|------------|
| R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| VREN | VROE | VRR | - | VR3 | VR2 | VR1 | VR0 |
| Бит 7 | | | | | | | Бит 0 |

R – чтение бита
 W – запись бита
 U – не реализовано, читается как 0
 -n – значение после POR
 -x – неизвестное значение после POR

бит 7: **VREN:** Включение источника опорного напряжения
 1 = источник опорного напряжения включен
 0 = источник опорного напряжения выключен и не потребляет тока

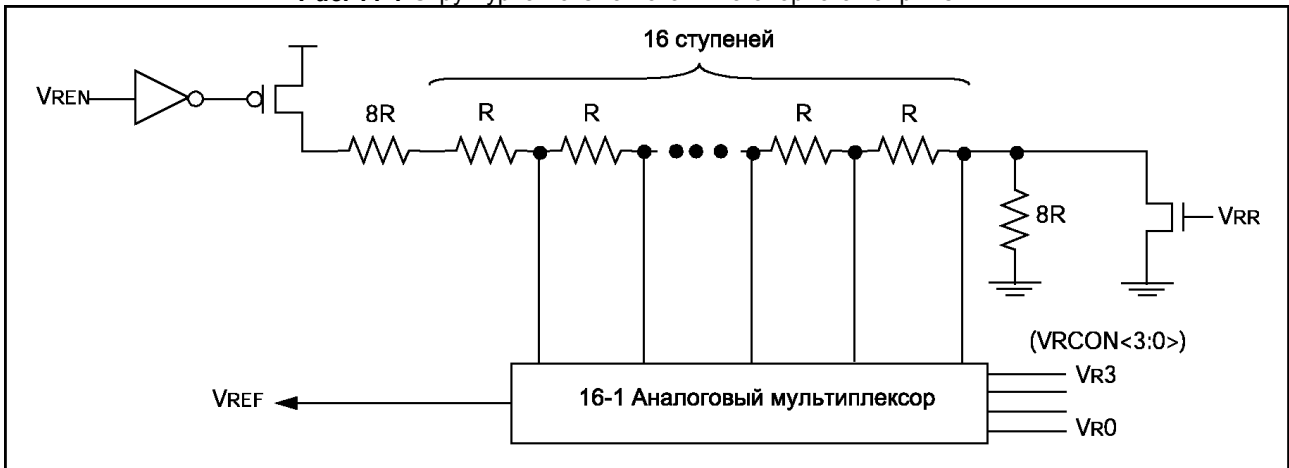
бит 6: **VROE:** Включение выхода VREF
 1 = выход VREF подключен к RA2
 0 = выход VREF не подключен к RA2

бит 5: **VRR:** Диапазон выходного напряжения VREF
 1 = нижний диапазон
 0 = верхний диапазон

бит 4: **Не используется:** читается как '0'

биты 3-0: **VR3:VR0:** Выбор выходного напряжения VREF $0 \leq VR[3:0] \leq 15$
 Если VRR = 1: $V_{REF} = (VR<3:0>/24) \times V_{DD}$
 Если VRR = 0: $V_{REF} = (V_{DD} \times 1/4) + (VR<3:0>/32) \times V_{DD}$

Рис. 11-1 Структурная схема источника опорного напряжения



Пример 11-1 Настройка опорного источника на напряжения 1.25В при напряжении питания 5В

```

MOVLW      0x02      ; 4 аналоговых входа
MOVWF      CMCON     ; 2-х компараторов.
BSF        STATUS,RP0 ; Выбрать Банк 1
MOVLW      0x07      ; RA3-RA0 настроить
MOVWF      TRISA     ; как выходы
MOVLW      0xA6      ; включить VREF
MOVWF      VRCON     ; нижний диапазон напряжений
                          ; установить VR<3:0>=6
BCF        STATUS,RP0 ; Выбрать Банк 0
CALL       DELAY_10  ; задержка 10 мкс

```

11.2 Точность источника опорного напряжения

Полный диапазон выходных напряжений (от V_{SS} до V_{DD}) не может быть реализован из-за особенностей схемы источника опорного напряжения. Транзисторы, включенные в начале и конце резистивной цепочки, создают некоторое смещение (см. рисунок 11-1). Выходное напряжение формируется относительно V_{DD} , поэтому может изменяться с колебаниями V_{DD} . Абсолютную точность источника опорного напряжения смотрите в таблице 17-2.

11.3 Функционирование в SLEEP режиме

Когда микроконтроллер выходит из режима SLEEP или происходит переполнение сторожевого таймера WDT, значение регистра VCON не изменяется. Для уменьшения суммарного тока потребления микроконтроллером в режиме SLEEP, модуль источника опорного напряжения следует выключать перед переходом в режим SLEEP.

11.4 Эффект сброса

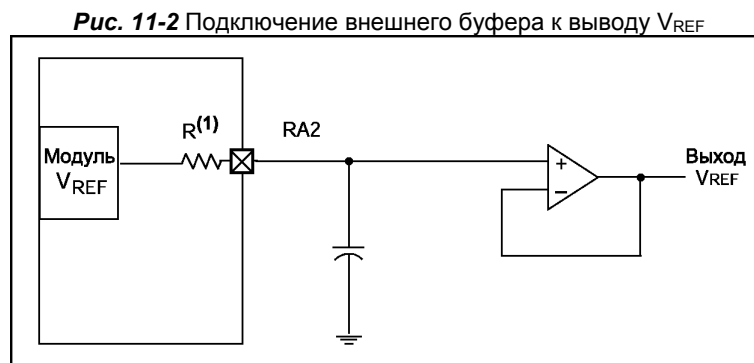
При сбросе микроконтроллера очищаются биты регистра VCON:

- VREN (VRCON<7>) – модуль источника опорного напряжения выключен;
- VROE (VRCON<6>) – VREF отсоединен от RA2;
- VRR (VRCON<5>) – верхний диапазон напряжений;
- VRCON<3:0> - напряжение опорного источника.

11.5 Подключение к источнику опорного напряжения

Модуль источника опорного напряжения работает независимо от модуля компараторов. Вывод источника опорного напряжения может быть подключен к RA2, если биты TRISA<2> и VROE (VRCON<6>) установлены в '1'. Подключение источника опорного напряжения к выводу RA2, с присутствующим на выводе внешним сигналом, может увеличить ток потребления микроконтроллером. Настройка RA2 как цифрового выхода также увеличит ток потребления.

Вывод RA2 может использоваться как простой ЦАП с малой разрешающей способностью. Вывод источника опорного напряжения имеет малую нагрузочную способность, поэтому на выходе VREF необходимо устанавливать дополнительный буфер (см. рисунок 11-2).



Примечание 1. Выходное сопротивление R зависит от битов VRCON<5> и VRCON<3:0>.

Таблица 11-1 Регистры и биты связанные с работой источника опорного напряжения

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|-------|-------|--------|--------|-------|--------|--------|--------|--------|--------|-----------|---------------|
| 9Fh | VRCON | VREN | VROE | VRR | - | VR3 | VR2 | VR1 | VR0 | 000- 0000 | 000- 0000 |
| 1Fh | CMCON | C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 0000 0000 | 0000 0000 |
| 85h | TRISA | TRISA7 | TRISA6 | - | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 11-1 1111 | 11-1 1111 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу источника опорного напряжения.

12.0 Универсальный синхронно-асинхронный приемопередатчик (USART)

USART – это модуль последовательного ввода/вывода, который может работать в полнодуплексном асинхронном режиме для связи с терминалами, персональными компьютерами или синхронном полудуплексном режиме для связи с микросхемами ЦАП, АЦП, последовательными EEPROM и т.д.

USART может работать в трех режимах:

- асинхронный, полный дуплекс;
- ведущий синхронный, полудуплекс;
- ведомый синхронный, полудуплекс.

Биты SPEN (RCSTA<7>) и TRISB<2:1> должны быть установлены в '1' для использования выводов RB2/TX/CK и RB1/RX/DT в качестве портов универсального синхронно-асинхронного приемопередатчика.

TXSTA (адрес 98h) Регистр управления и статуса передатчика

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R-1 | R/W-0 |
|-------------|------------|-------------|-------------|-----|-------------|-------------|-------------|
| CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D |
| Бит 7 | | | | | | | Бит 0 |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
–n – значение после POR
–x – неизвестное значение после POR

бит 7: **CSRC:** Выбор источника тактового сигнала
Синхронный режим
1 = ведущий, внутренний тактовый сигнал от BGR
0 = ведомый, внешний тактовый сигнал с входа CK

Асинхронный режим
Не имеет значения

бит 6: **TX9:** Разрешение 9-битной передачи
1 = 9-битная передача
0 = 8-битная передача

бит 5: **TXEN:** Разрешение передачи
1 = разрешена
0 = запрещена
Примечание. В синхронном режиме биты SREN/CREN отменяют действие бита TXEN.

бит 4: **SYNC:** Режим работы USART
1 = синхронный
0 = асинхронный

бит 3: **Не используется:** читается как '0'

бит 2: **BRGH:** Выбор высокоскоростного режима
Синхронный режим
Не имеет значения

Асинхронный режим
1 = высокоскоростной режим
0 = низкоскоростной режим

бит 1: **TRMT:** Флаг очистки сдвигового регистра передатчика TSR
1 = TSR пуст
0 = TSR полон

бит 0: **TX9D:** 9-й бит передаваемых данных (может использоваться для программной проверки четности)

RCSTA (адрес 18h) Регистр управления и статуса приемника

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
|-------------|------------|-------------|-------------|--------------|-------------|-------------|-------------|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| Бит 7 | | | | | | | Бит 0 |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

бит 7: **SPEN:** Разрешение работы последовательного порта (вывод RB1/RX/DT и RB2/TX/CK используются USART когда биты TRISB<2:1> установлены в '1')
1 = модуль последовательного порта включен
0 = модуль последовательного порта выключен

бит 6: **RX9:** Разрешение 9-битного приема
1 = 9-битный прием
0 = 8-битный прием

бит 5: **SREN:** Разрешение одиночного приема
Синхронный режим
1 = разрешен одиночный прием
0 = запрещен одиночный прием
Сбрасывается в '0' по завершению приема.
Примечание. В режиме ведомого не имеет значения

Асинхронный режим
Не имеет значения

бит 4: **CREN:** Разрешение приема
Синхронный режим
1 = прием разрешен (при установке бита CREN автоматически сбрасывается бит SREN)
0 = прием запрещен

Асинхронный режим
1 = прием разрешен
0 = прием запрещен

бит 3: **ADDEN:** Разрешение детектирования адреса
Асинхронный 9-битный прием (RX9=1)
1 = детектирование адреса разрешено. Если бит RSR<8>=1, то генерируется прерывание и загружается приемный буфер.
0 = детектирование адреса запрещено. Принимаются все байты, девятый бит может использоваться для проверки четности.

Асинхронный 8-битный прием (RX9=0)
Не имеет значения

Синхронный режим
Не имеет значения

бит 2: **FERR:** Ошибка кадра, сбрасывается при чтении регистра RCREG
1 = произошла ошибка кадра
0 = ошибки кадра не было

бит 1: **OERR:** Ошибка переполнения внутреннего буфера, устанавливается в '0' при сбросе бита CREN
1 = произошла ошибка переполнения
0 = ошибки переполнения не было

бит 0: **RX9D:** 9-й бит принятых данных (может использоваться для программной проверки четности)

12.1 Генератор частоты обмена USART BRG

BRG используется для работы USART в синхронном ведущем и асинхронном режимах. BRG представляет собой отдельный 8-разрядный генератор скорости обмена в бодах, период которого определяется значением в регистре SPBRG. В асинхронном режиме бит BRGH (TXSTA<2>) тоже влияет на скорость обмена (в синхронном режиме бит BRGH игнорируется). В таблице 12-1 указаны формулы для вычисления скорости обмена в бодах при различных режимах работы модуля USART (относительно внутреннего тактового сигнала микроконтроллера).

Учитывая требуемую скорость и F_{OSC} , выбирается самое близкое целое значение для записи в регистр SPBRG, рассчитанное по формулам приведенным в таблице 12-1. Затем рассчитывается ошибка скорости обмена.

В примере 12-1 показан расчет значения для регистра SPBRG и погрешность скорости обмена для следующих условий:

$F_{OSC} = 16$ МГц;
Скорость приема/передачи данных 9600 бит/с;
BRGH = 0;
SYNC = 0.

Пример 12-1 Расчет значения для регистра SPBRG и погрешность скорости обмена

Желаемое значение скорости = $F_{OSC} / (64 (X + 1))$

$9600 = 16\,000\,000 / (64 (X + 1))$

$X = [25.042] = 25$

Вычисленное значение скорости = $16\,000\,000 / (64 (25 + 1)) = 9615$

Ошибка = $100 \times (\text{Вычисленное} - \text{Желаемое}) / \text{Желаемое значение скорости}$

Ошибка = $100 \times (9615 - 9600) / 9600 = 0.16\%$

В некоторых случаях может быть выгодно использовать высокоскоростной режим работы USART (BRGH=1), поскольку уравнение $F_{OSC} / (16 (X + 1))$ позволяет уменьшить погрешность скорости. Запись нового значения в регистр SPBRG сбрасывает таймер BRG, гарантируя немедленный переход на новую скорость.

Таблица 12-1 Формулы расчета скорости обмена данными

| SYNC | BRGH = 0 | BRGH = 1 |
|------|--|--|
| 0 | (Асинхронный) Скорость обмена = $F_{OSC} / (64 (X + 1))$ | (Асинхронный) Скорость обмена = $F_{OSC} / (16 (X + 1))$ |
| 1 | (Синхронный) Скорость обмена = $F_{OSC} / (4 (X + 1))$ | (Синхронный) Скорость обмена = $F_{OSC} / (4 (X + 1))$ |

X = значение регистра SPBRG (от 0 до 255)

Таблица 12-2 Регистры и биты связанные с работой генератора BRG

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|-------|-------|-----------------------------------|-------|-------|-------|-------|-------|-------|-------|-----------|---------------|
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 | 0000 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу генератора BRG.

Таблица 12-3 Скорость обмена в синхронном режиме

| Скорость обмена (К) | Fosc = 20 МГц | | | Fosc = 16 МГц | | | Fosc = 10 МГц | | |
|---------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 0,3 | Нет | - | - | Нет | - | - | Нет | - | - |
| 1,2 | Нет | - | - | Нет | - | - | Нет | - | - |
| 2,4 | Нет | - | - | Нет | - | - | Нет | - | - |
| 9,6 | Нет | - | - | Нет | - | - | 9,166 | +1,73 | 255 |
| 19,2 | 19,53 | +1,73 | 255 | 19,23 | +0,16 | 207 | 19,23 | +0,16 | 129 |
| 76,8 | 76,92 | +0,16 | 64 | 76,92 | +0,16 | 51 | 7576 | -1,36 | 32 |
| 96 | 96,15 | +0,16 | 51 | 95,24 | -0,79 | 41 | 96,15 | +0,16 | 25 |
| 300 | 294,1 | -1,96 | 16 | 307,69 | +2,56 | 12 | 312,5 | +4,17 | 7 |
| 500 | 500 | 0 | 9 | 500 | 0 | 7 | 500 | 0 | 4 |
| Максим. | 5000 | - | 0 | 4000 | - | 0 | 2500 | - | 0 |
| Миним. | 19,53 | - | 255 | 15,625 | - | 255 | 9,766 | - | 255 |

| Скорость обмена (К) | Fosc = 7,15909 МГц | | | Fosc = 5,0688 МГц | | | Fosc = 4 МГц | | |
|---------------------|--------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 0,3 | Нет | - | - | Нет | - | - | Нет | - | - |
| 1,2 | Нет | - | - | Нет | - | - | Нет | - | - |
| 2,4 | Нет | - | - | Нет | - | - | Нет | - | - |
| 9,6 | 9,622 | +0,23 | 185 | 9,6 | 0 | 131 | 9,615 | +0,16 | 103 |
| 19,2 | 19,24 | +0,23 | 92 | 19,2 | 0 | 65 | 19,231 | +0,16 | 51 |
| 76,8 | 77,82 | +1,32 | 22 | 79,2 | +3,13 | 15 | 76,923 | +0,16 | 12 |
| 96 | 94,20 | -1,88 | 18 | 97,48 | +1,54 | 12 | 100 | +4,17 | 9 |
| 300 | 298,3 | -0,57 | 5 | 316,8 | +5,60 | 3 | Нет | - | - |
| 500 | Нет | - | - | Нет | - | - | Нет | - | - |
| Максим. | 1789,8 | - | 0 | 1267 | - | 0 | 1000 | - | 0 |
| Миним. | 6,991 | - | 255 | 4,950 | - | 255 | 3,906 | - | 255 |

| Скорость обмена (К) | Fosc = 3,579545 МГц | | | Fosc = 1 МГц | | | Fosc = 32,768 кГц | | |
|---------------------|---------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 0,3 | Нет | - | - | Нет | - | - | 0,303 | +1,14 | 26 |
| 1,2 | Нет | - | - | 1,202 | +0,16 | 207 | 1,170 | -2,48 | 6 |
| 2,4 | Нет | - | - | 2,404 | +0,16 | 103 | Нет | - | - |
| 9,6 | 9,622 | +0,23 | 92 | 9,615 | +0,16 | 25 | Нет | - | - |
| 19,2 | 19,04 | -0,83 | 46 | 19,24 | +0,16 | 12 | Нет | - | - |
| 76,8 | 74,57 | -2,90 | 11 | 83,34 | +8,51 | 2 | Нет | - | - |
| 96 | 99,43 | +3,57 | 8 | Нет | - | - | Нет | - | - |
| 300 | 298,3 | -0,57 | 2 | Нет | - | - | Нет | - | - |
| 500 | Нет | - | - | Нет | - | - | Нет | - | - |
| Максим. | 894,9 | - | 0 | 250 | - | 0 | 8,192 | - | 0 |
| Миним. | 3,496 | - | 255 | 0,9766 | - | 255 | 0,032 | - | 255 |

Таблица 12-4 Скорость обмена в асинхронном режиме (BRGH=0)

| Скорость обмена (К) | Fosc = 20 МГц | | | Fosc = 16 МГц | | | Fosc = 10 МГц | | |
|---------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 0,3 | Нет | - | - | Нет | - | - | Нет | - | - |
| 1,2 | 1,221 | +1,73 | 255 | 1,202 | +0,16 | 207 | 1,202 | +0,16 | 129 |
| 2,4 | 2,404 | +0,16 | 129 | 2,404 | +0,16 | 103 | 2,404 | +0,16 | 64 |
| 9,6 | 9,469 | -1,36 | 32 | 9,615 | +0,16 | 25 | 9,615 | +1,73 | 15 |
| 19,2 | 19,53 | +1,73 | 15 | 19,23 | +0,16 | 12 | 19,53 | +1,73 | 7 |
| 76,8 | 78,13 | +1,73 | 3 | 83,33 | +8,51 | 2 | 78,13 | +1,73 | 1 |
| 96 | 104,2 | +8,51 | 2 | Нет | - | - | Нет | - | - |
| 300 | 312,5 | +4,17 | 0 | Нет | - | - | Нет | - | - |
| 500 | Нет | - | - | Нет | - | - | Нет | - | - |
| Максим. | 312,5 | - | 0 | 250 | - | 0 | 156,3 | - | 0 |
| Миним. | 1,221 | - | 255 | 0,977 | - | 255 | 0,6104 | - | 255 |

| Скорость обмена (К) | Fosc = 7,15909 МГц | | | Fosc = 5,0688 МГц | | | Fosc = 4 МГц | | |
|---------------------|--------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 0,3 | Нет | - | - | 0,31 | +3,13 | 255 | 0,3005 | -0,17 | 207 |
| 1,2 | 1,203 | +0,23 | 92 | 1,2 | 0 | 65 | 1,202 | +1,67 | 51 |
| 2,4 | 2,380 | -0,83 | 46 | 2,4 | 0 | 32 | 2,404 | +1,67 | 25 |
| 9,6 | 9,322 | -2,90 | 11 | 9,9 | +3,13 | 7 | Нет | - | - |
| 19,2 | 18,64 | -2,90 | 5 | 19,8 | +3,13 | 3 | Нет | - | - |
| 76,8 | Нет | - | - | 79,2 | +3,13 | 0 | Нет | - | - |
| 96 | Нет | - | - | Нет | - | - | Нет | - | - |
| 300 | Нет | - | - | Нет | - | - | Нет | - | - |
| 500 | Нет | - | - | Нет | - | - | Нет | - | - |
| Максим. | 111,9 | - | 0 | 79,2 | - | 0 | 62,500 | - | 0 |
| Миним. | 0,437 | - | 255 | 0,3094 | - | 255 | 3,906 | - | 255 |

| Скорость обмена (К) | Fosc = 3,579545 МГц | | | Fosc = 1 МГц | | | Fosc = 32,768 кГц | | |
|---------------------|---------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 0,3 | 0,301 | +0,23 | 185 | 0,300 | +0,16 | 51 | 0,256 | -14,67 | 1 |
| 1,2 | 1,190 | -0,83 | 46 | 1,202 | +0,16 | 12 | Нет | - | - |
| 2,4 | 2,432 | +1,32 | 22 | 2,232 | -6,99 | 6 | Нет | - | - |
| 9,6 | 9,322 | -2,90 | 5 | Нет | - | - | Нет | - | - |
| 19,2 | 18,64 | -2,90 | 2 | Нет | - | - | Нет | - | - |
| 76,8 | Нет | - | - | Нет | - | - | Нет | - | - |
| 96 | Нет | - | - | Нет | - | - | Нет | - | - |
| 300 | Нет | - | - | Нет | - | - | Нет | - | - |
| 500 | Нет | - | - | Нет | - | - | Нет | - | - |
| Максим. | 55,93 | - | 0 | 15,63 | - | 0 | 0,512 | - | 0 |
| Миним. | 0,2185 | - | 255 | 0,0610 | - | 255 | 0,0020 | - | 255 |

Таблица 12-5 Скорость обмена в асинхронном режиме (BRGH=1)

| Скорость обмена (К) | Fosc = 20 МГц | | | Fosc = 16 МГц | | | Fosc = 10 МГц | | |
|---------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 9,6 | 9,615 | +0,16 | 129 | 9,615 | +0,16 | 103 | 9,615 | +0,16 | 64 |
| 19,2 | 19,230 | +0,16 | 64 | 19,230 | +0,16 | 51 | 18,939 | -1,36 | 32 |
| 38,4 | 37,878 | -1,36 | 32 | 38,461 | +0,16 | 25 | 39,062 | +1,7 | 15 |
| 57,6 | 56,818 | -1,36 | 21 | 58,823 | +2,12 | 16 | 56,818 | -1,36 | 10 |
| 115,2 | 113,636 | -1,36 | 10 | 111,111 | -3,55 | 8 | 125 | +8,51 | 4 |
| 250 | 250 | 0 | 4 | 250 | 0 | 3 | Нет | - | - |
| 625 | 625 | 0 | 1 | Нет | - | - | 625 | 0 | 0 |
| 1250 | 1250 | 0 | 0 | Нет | - | - | Нет | - | - |

| Скорость обмена (К) | Fosc = 7,16 МГц | | | Fosc = 5,068 МГц | | | Fosc = 4 МГц | | |
|---------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 9,6 | 9,520 | -0,83 | 46 | 9,6 | 0 | 32 | Нет | - | - |
| 19,2 | 19,454 | +1,32 | 22 | 18,645 | -2,94 | 16 | 1,202 | +0,17 | 207 |
| 38,4 | 37,286 | -2,90 | 11 | 39,6 | +3,12 | 7 | 2,403 | +0,13 | 103 |
| 57,6 | 55,930 | -2,90 | 7 | 52,8 | -8,33 | 5 | 9,615 | +0,16 | 25 |
| 115,2 | 111,860 | -2,90 | 3 | 105,6 | -8,33 | 2 | 19,231 | +0,16 | 12 |
| 250 | Нет | - | - | Нет | - | - | Нет | - | - |
| 625 | Нет | - | - | Нет | - | - | Нет | - | - |
| 1250 | Нет | - | - | Нет | - | - | Нет | - | - |

| Скорость обмена (К) | Fosc = 3,579 МГц | | | Fosc = 1 МГц | | | Fosc = 32,768 кГц | | |
|---------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|-------------------|----------|-------------------------|
| | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) | Реальная скорость | Ошибка % | Значение SPBRG (десят.) |
| 9,6 | 9,727 | +1,32 | 22 | 8,928 | -6,99 | 6 | Нет | - | - |
| 19,2 | 18,643 | -2,90 | 11 | 20,833 | +8,51 | 2 | Нет | - | - |
| 38,4 | 37,286 | -2,90 | 5 | 31,25 | -18,61 | 1 | Нет | - | - |
| 57,6 | 55,930 | -2,90 | 3 | 62,5 | +8,51 | 0 | Нет | - | - |
| 115,2 | 111,860 | -2,90 | 1 | Нет | - | - | Нет | - | - |
| 250 | 223,721 | -10,51 | 0 | Нет | - | - | Нет | - | - |
| 625 | Нет | - | - | Нет | - | - | Нет | - | - |
| 1250 | Нет | - | - | Нет | - | - | Нет | - | - |

12.1.1 Выборка

Сигнал с входа RB1/RX/DT опрашивается цепью мажоритарного детектора три раза за такт передачи, чтобы определить высокого или низкого уровня сигнал присутствует на входе. Если выбран низкоскоростной режим (BRRH=0), то выборка производится по седьмому, восьмому и девятому заднему фронту тактового сигнала x16 (см. рисунок 12-1). Если BRGH = 1 (выбран высокоскоростной режим), выборка производится на втором такте сигнала x4 тремя запросами (см. рисунки 12-2, 12-3).

Рис. 12-1 Временная диаграмма выборки (BRGH = 0)

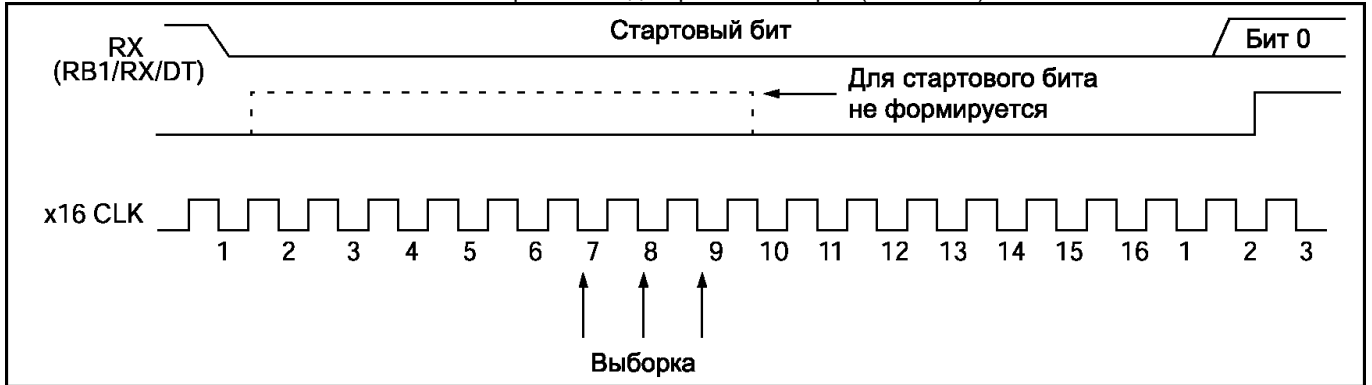


Рис. 12-2 Временная диаграмма выборки (BRGH = 1)

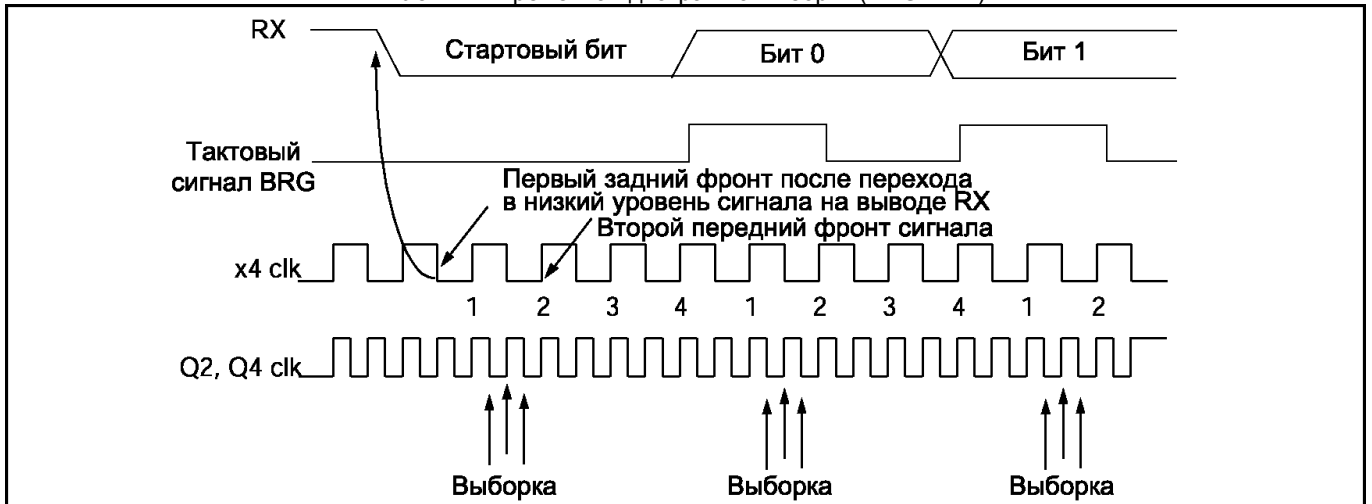
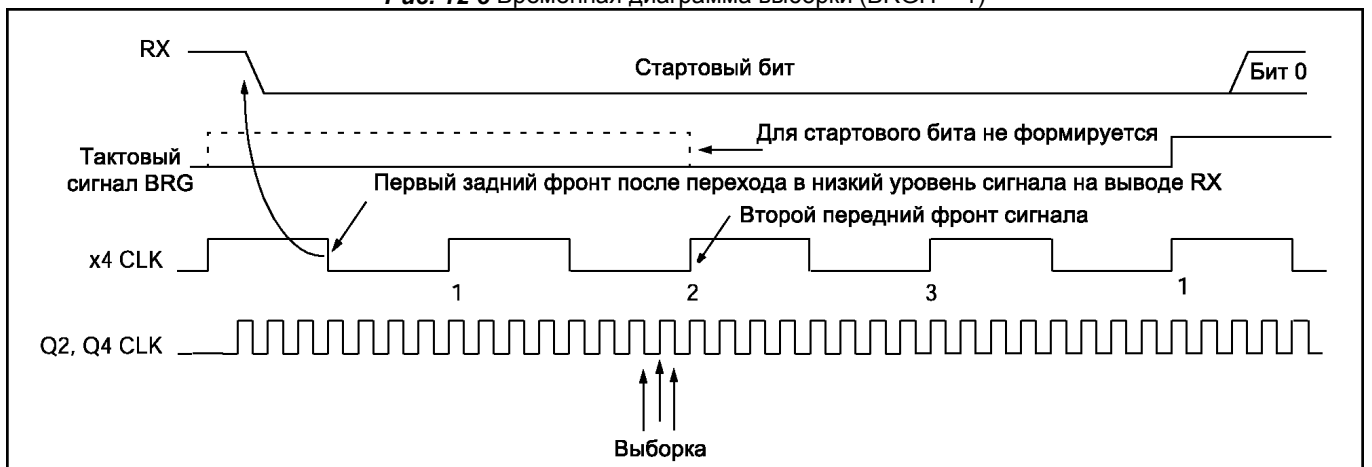


Рис. 12-3 Временная диаграмма выборки (BRGH = 1)



12.2 Асинхронный режим USART

В этом режиме USART использует стандартный формат NRZ: один стартовый бит, восемь или девять битов данных и один стоповый бит. Наиболее часто встречается 8-разрядный формат передачи данных. Интегрированный 8-разрядный генератор BRG позволяет получить стандартные скорости передачи данных. Генератор скорости обмена может работать в одном из двух режимов: высокоскоростной ($x16$ BRGH=1 TXSTA<2>), низкоскоростной ($x64$ BRGH=0 TXSTA<2>). Приемник и передатчик последовательного порта работают независимо друг от друга, но используют один и тот же формат данных и одинаковую скорость обмена. Бит четности аппаратно не поддерживается, но может быть реализован программно, применяя 9-разрядный формат данных. Все данные передаются младшим битом вперед. В SLEEP режиме микроконтроллера модуль USART (асинхронный режим) выключен. Выбор асинхронного режима USART выполняется сбросом бита SYNC в '0' (TXSTA<4>).

Модуль USART в асинхронном режиме состоит из следующих элементов:

- генератор скорости обмена;
- цепь опроса;
- асинхронный передатчик;
- асинхронный приемник.

12.2.1 Асинхронный передатчик USART

Структурная схема асинхронного передатчика USART показана на рисунке 12-4. Главным в передатчике является сдвиговый регистр TSR, который получает данные из буфера передатчика TXREG. Данные в регистр TXREG загружаются программно. После передачи стопового бита предыдущего байта, в последнем машинном такте цикла BRG, TSR загружается новым значением из TXREG (если оно присутствует), после чего выставляется флаг прерывания TXIF (PIR1<4>). Прерывание может быть разрешено или запрещено битом TXIE (PIE1<1>). Флаг TXIF устанавливается независимо от состояния бита TXIF и не может быть сброшен в '0' программно. Очистка флага TXIF происходит только после загрузки новых данных в регистр TXREG. Аналогичным образом бит TRMT (TXSTA<1>) отображает состояние регистра TSR. Бит TRMT доступен только на чтение и не может вызвать генерацию прерывания.

Примечания:

1. Регистр TSR не отображается на память и не доступен для чтения.
2. Флаг TXIF устанавливается в '1' только, когда бит TXEN=1.

Для разрешения передачи необходимо установить бит TXEN (TXSTA<5>) в '1'. Передача данных не начнется до тех пор, пока в TXREG не будут загружены новые данные, не придет очередной тактовый импульс от генератора BRG (см рисунок 12-5). Можно сначала загрузить данные в TXREG, а затем установить бит TXEN. Как правило, после разрешения передачи регистр TSR пуст, таким образом, данные записываемые в TXREG сразу передаются в TSR, а TXREG остается пустым. Это позволяет реализовать слитную передачу данных (см. рисунок 12-6). Сброс бита TXEN в '0' вызовет немедленное прекращение передачи, сброс передатчика и перевод вывода RB2/TX/CK в третье состояние.

Для разрешения 9-разрядной передачи, необходимо установить бит TX9 (TXSTA<6>) в '1'. Девятый бит данных записывается в бит TX9D (TXSTA<0>). Девятый бит данных должен быть указан до записи в регистр TXREG, потому что данные, записанные в регистр TXREG, могут быть сразу загружены в сдвиговый регистр TSR.

Рекомендованная последовательность действий для передачи данных в асинхронном режиме:

1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 12.1).
2. Выбрать асинхронный режим сбросом бита SYNC в '0' и установкой бита SPEN в '1'.
3. Если необходимо, разрешить прерывания установкой бита TXIE в '1'.
4. Если передача 9-разрядная, установить бит TX9 в '1'.
5. Разрешить передачу установкой бита TXEN в '1', автоматически устанавливается флаг TXIF.
6. Если передача 9-разрядная, записать 9-й бит данных в TX9D.
7. Записать данные в регистр TXREG.

Рис. 12-4 Структурная схема асинхронного передатчика USART

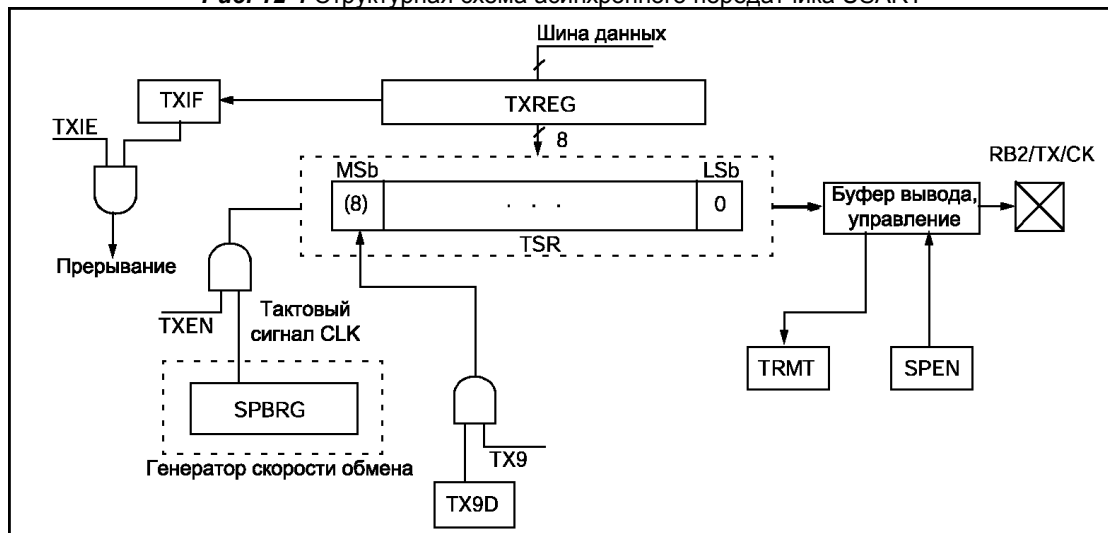


Рис. 12-5 Временная диаграмма асинхронной передачи данных

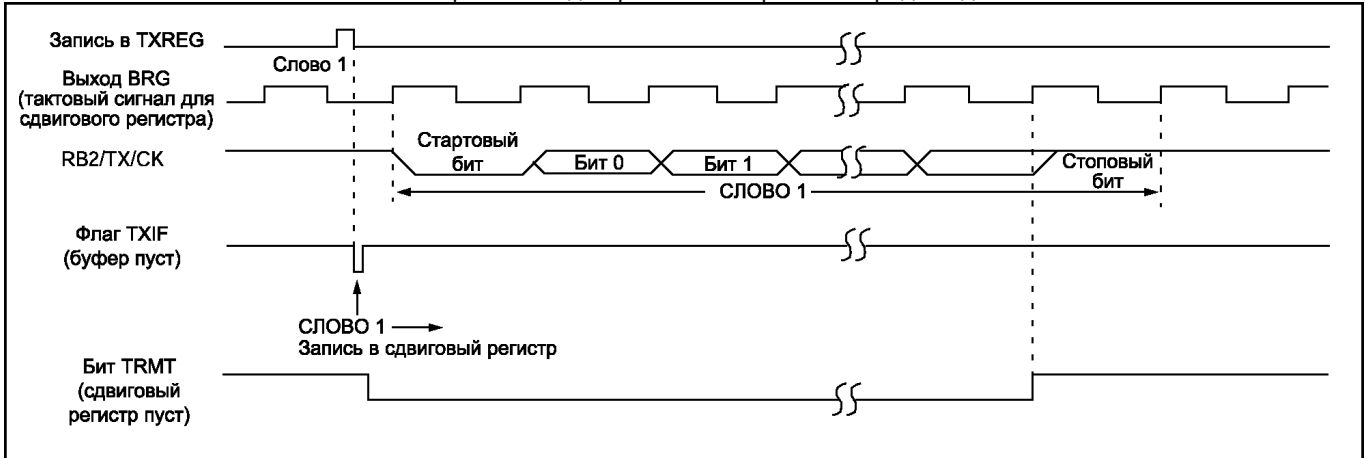


Рис. 12-6 Временная диаграмма слитная асинхронной передачи (последовательная передача двух байт)

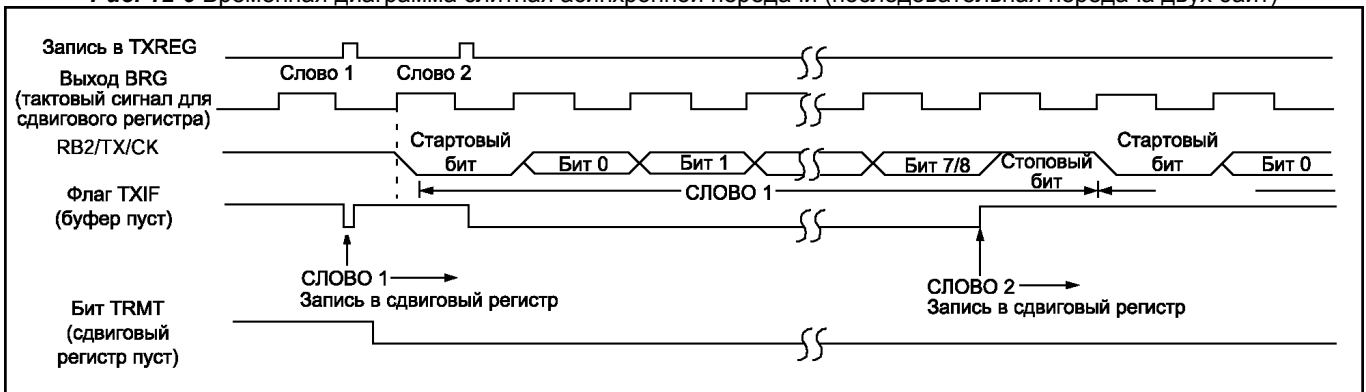


Таблица 12-6 Регистры и биты связанные с работой передатчика USART в асинхронном режиме

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|-----------------------------------|-------|-------|-------|-------|-------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG | Регистр данных передатчика USART | | | | | | | | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 | 0000 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу передатчика USART в асинхронном режиме.

12.2.2 Асинхронный приемник USART

Структурная схема асинхронного приемника USART показана на рисунке 12-7. Данные подаются на вход RB1/RX/DT в блок восстановления данных, представляющий собой скоростной сдвиговый регистр, работающий на частоте в 16 раз превышающей скорость передачи или F_{osc} .

Включение приемника производится установкой бита CREN регистра RCSTA в '1'.

Главным в приемнике является сдвиговый регистр RSR. После получения стопового бита данные переписываются в регистр RCREG, если он пуст. После записи в регистр RCREG выставляется флаг прерывания RCIF (PIR1<5>). Прерывание можно разрешить/запретить установкой/сбросом бита RCIE (PIE1<5>). Флаг RCIF доступен только на чтение, сбрасывается аппаратно при чтении из регистра RCREG. Регистр RCREG имеет двойную буферизацию, т.е. представляет собой двухуровневый буфер FIFO. Поэтому можно принять 2 байта данных в FIFO RCREG и третий в регистр RSR. Если FIFO заполнен и обнаружен стоповый бит третьего байта, устанавливается бит переполнения приемника OERR (RCSTA<1>). Байт, принятый в RSR, будет потерян. Для извлечения двух байт из FIFO, необходимо дважды прочитать регистр RCREG. Бит OERR нужно программно очистить сбросом бита CREN, т.е. запрещением приема. В любом случае, если бит OERR установлен, логика приемника выключена.

Бит ошибки кадра FERR (RCSTA<2>) устанавливается в '1', если не обнаружен стоповый бит. FERR и девятый бит принятых данных буферизируются также, как принятые данные. Рекомендуется сначала прочитать регистр RCSTA, затем RCREG, чтобы не потерять информацию RX9D и FERR.

Рекомендованные действия при приеме данных в асинхронном режиме:

1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 12.1).
2. Выбрать асинхронный режим сбросом бита SYNC в '0' и установкой бита SPEN в '1'.
3. Если необходимо, разрешить прерывания установкой бита RCIE в '1'.
4. Если прием 9-битный, установить бит RX9 в '1'.
5. Разрешить прием установкой бита CREN в '1'.
6. Ожидать установку бита RCIF, или прерывание, если оно разрешено битом RCIE.
7. Считать 9-й бит данных (если разрешен 9-разрядный прием) из регистра RCSTA и проверить возникновение ошибки.
8. Считать 8 бит данных из регистра RCREG.
9. При возникновении ошибки переполнения сбросить бит CREN в '0'.

Рис. 12-7 Структурная схема асинхронного приемника USART

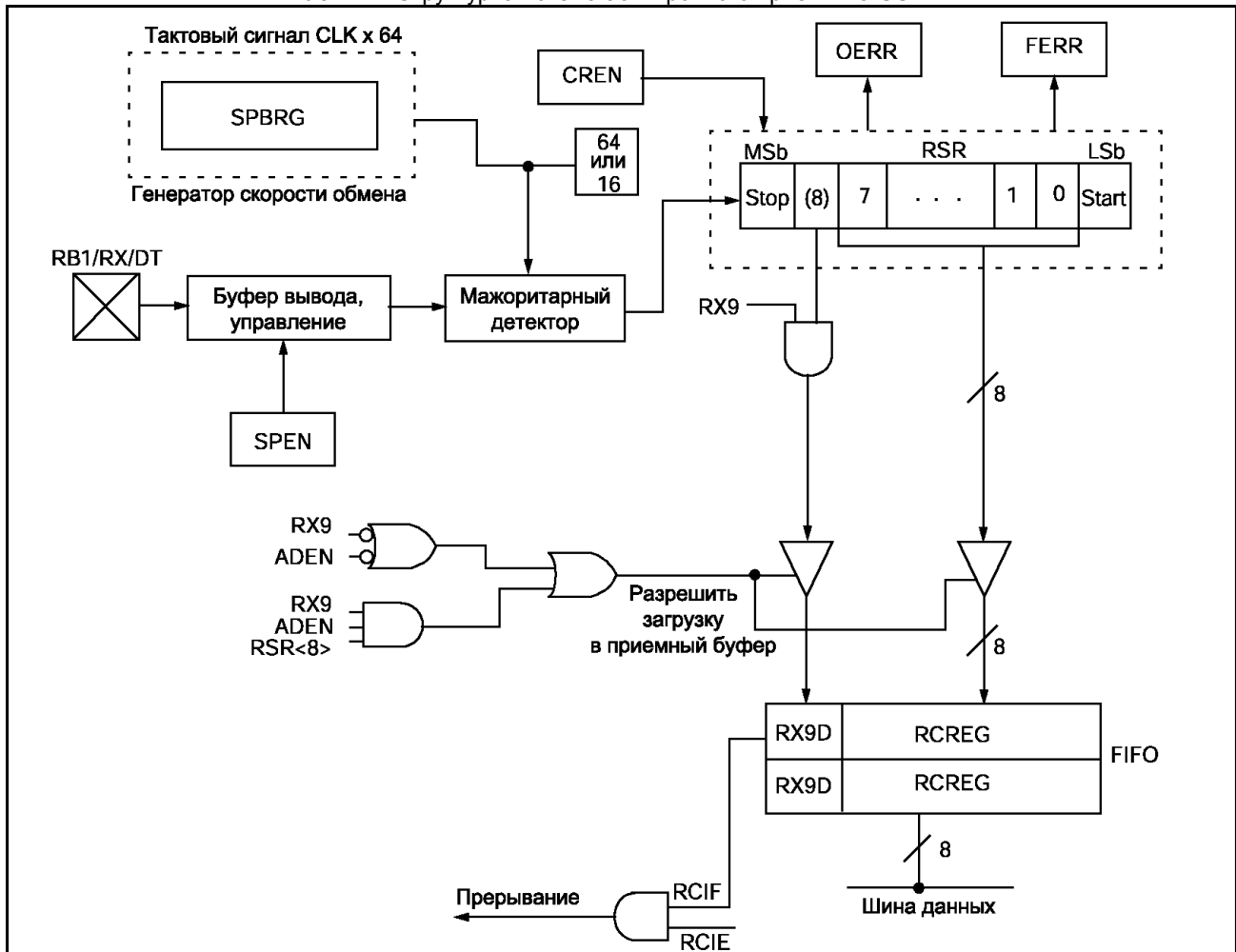
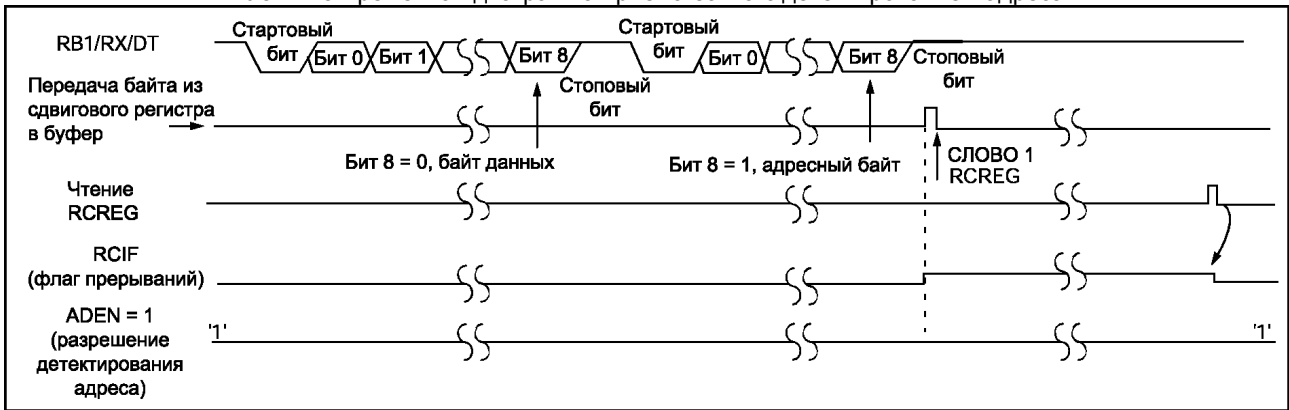
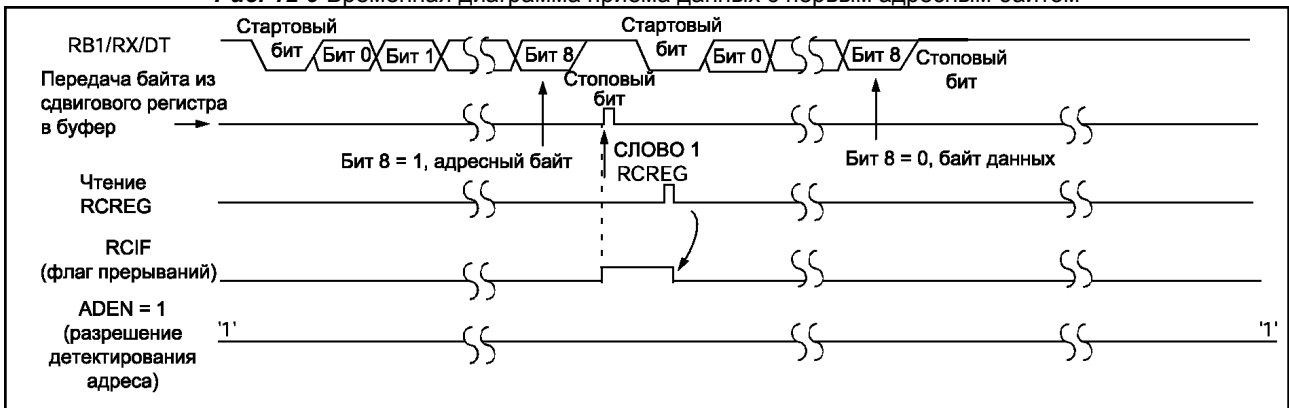


Рис. 12-8 Временная диаграмма приема байта с детектированием адреса



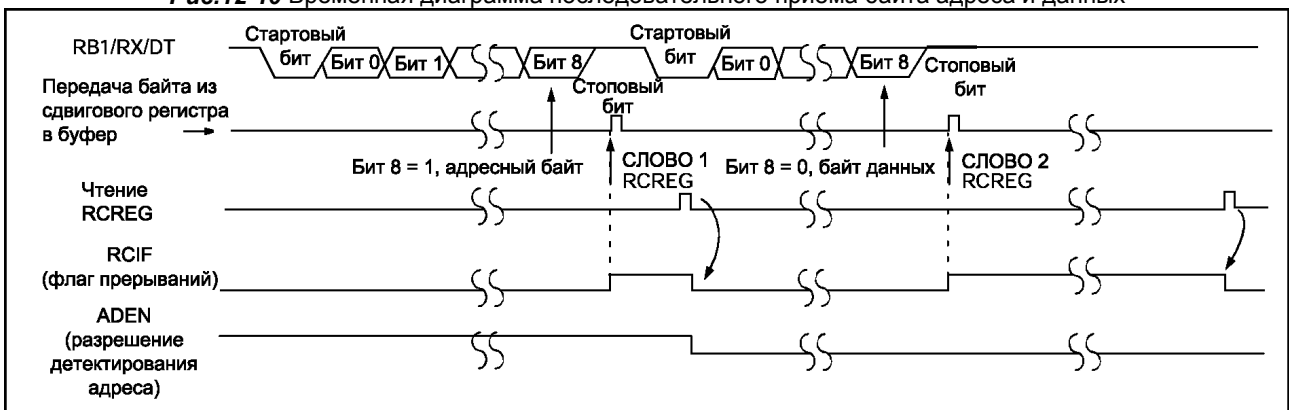
Примечание к рисунку. На временной диаграмме показан последовательный прием байта данных и байта адреса. Байт данных не записывается в RCREG, т.к. ADDEN=1, а бит8 = 0.

Рис. 12-9 Временная диаграмма приема данных с первым адресным байтом



Примечание к рисунку. На временной диаграмме показан последовательный прием байта адреса и байта данных. Байт данных не записывается в RCREG, т.к. ADDEN не был сброшен в '0' (ADDEN=1), а бит8 = 0.

Рис.12-10 Временная диаграмма последовательного приема байта адреса и данных



Примечание к рисунку. На временной диаграмме показан последовательный прием байта адреса и байта данных. Байт данных сохраняется в регистре RCREG, потому что бит ADDEN был сброшен в '0' после совпадения адреса. Содержимое регистра RSR всегда будет передаваться в регистр RCREG независимо от состояния бита 8.

Таблица 12-7 Регистры и биты связанные с работой приемника USART в асинхронном режиме

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|-----------------------------------|-------|-------|-------|-------|-------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 1Ah | RCREG | Регистр данных приемника USART | | | | | | | | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 | 0000 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу приемника USART в асинхронном режиме.

12.3 Функции USART

Функции модуля USART аналогичны модулю в микроконтроллере PIC16C74B, включая высокоскоростной режим (BRGH = 1).

12.3.1 Приемник с детектором 9-разрядного адреса

Если установлен бит RX9 в регистре RCSTA, 9-й бит принимаемых данных сохраняется в бите RX9D регистра RCSTA. Модуль USART имеет функцию детектирование адреса для организации сетевого обмена. Детектирование адреса разрешено, если установлены биты ADDEN(RCSTA<3>) и RX9 (RCSTA<6>) в '1'. В этом режиме принимаемые данные записываются в регистр RCREG (9-бит в RX9D регистра RCSTA), если девятый бит принимаемых данных равен 1.

Для передачи данных в сетевой структуре ведущее устройство должно сначала передать адрес ведомого устройства. В байте адреса 9 бит (RSR<8>) равен 1 (в байте данных RSR<8>=0). Если в регистре RCSTA биты RX9 и ADDEN установлены в '1', то разрешено детектирование адреса, все байты данных будут игнорироваться. Однако если 9 бит принятых данных будет равен 1 (принят адресный байт), содержимое регистра RSR передается в приемный буфер. Это позволяет ведомому устройству обрабатывать только адресные байты. Если принятый адресный байт соответствует адресу ведомого устройства, необходимо сбросить бит ADDEN в '0' для перехода в режим приема данных.

Когда бит ADDEN = 1, все принимаемые байты данных игнорируются. После приема стопового бита, данные не загружаются в приемный буфер, прерывание не генерируется. Если принят следующий байт, предыдущий байт в регистре RSR будет потерян.

Детектирование адреса разрешено, если только включен 9-разрядный прием данных (RX9 = 1). Если детектирование адреса запрещено (ADDEN = 0), девятый бит принимаемых данных может использоваться для контроля четности.

Структурная схема асинхронного приемника USART показана на рисунке 12-7.

Прием разрешен, когда бит CREN (RCSTA<4>) = 1.

12.3.1.1 Настройка 9-разрядного асинхронного приема с детектированием адреса

Рекомендованная последовательность действия при использовании детектора адреса:

1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 12.1).
2. Выбрать асинхронный режим сбросом бита SYNC в '0' и установкой бита SPEN в '1'.
3. Если необходимо, разрешить прерывания установкой бита RCIE в '1'.
4. Установить бит RX9 в '1' для включения 9-разрядного приема.
5. Установить бит ADDEN в '1' для разрешения детектирования адреса.
6. Разрешить прием установкой бита CREN в '1'.
7. Ожидать установку бита RCIF, или прерывание, если оно разрешено битом RCIE.
8. Считать 8 бит данных из регистра RCREG для проверки адресации устройства.
9. При возникновении ошибки переполнения сбросить бит CREN в '0'.
10. Если принятый адрес соответствует адресу устройства, сбросить биты ADDEN и RCIF в '0' для начала приема данных.

Таблица 12-8 Регистры и биты связанные с работой приемника USART в асинхронном режиме

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|-----------------------------------|-------|-------|-------|-------|-------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 1Ah | RCREG | Регистр данных приемника USART | | | | | | | | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 | 0000 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу приемника USART в асинхронном режиме.

12.4 Синхронный ведущий режим USART

В ведущем синхронном режиме данные передаются полудуплексом, т.е. прием и передача не происходит одновременно. При передаче запрещен прием и наоборот. Синхронный режим включается установкой бита SYNC (TXSTA<4>) в '1'. Также необходимо включить модуль USART, установкой бита SPEN (RCSTA<7>) в '1', для настройки портов ввода вывода RB2/TX/CK и RB1/RX/DT в качестве тактового сигнала CK и линии данных DT соответственно. В режиме ведущего модуль USART формирует тактовый сигнал CK. Выбор режима ведущего производится установкой бита CSRC (TXSTA<7>) в '1'.

12.4.1 Передача синхронного ведущего

Структурная схема передатчика USART показана на рисунке 12-4. Главным в передатчике является сдвиговый регистр TSR. Сдвиговый регистр получает данные из буфера передатчика TXREG. В регистр TSR не загружаются новые данные, пока не будет передан последний бит предыдущего байта. После передачи последнего бита предыдущего байта, TSR загружается новым значением из TXREG (если оно присутствует), и устанавливается флаг прерывания TXIF (PIR1<4>). Это прерывание может быть разрешено/запрещено битом TXIE (PIE<4>). Флаг TXIF устанавливается вне зависимости от состояния бита TXIE и может быть сброшен только загрузкой новых данных в регистр TXREG. Также как TXIF отображает состояние TXREG, бит TRMT (TSTA<1>) показывает состояние регистра TSR. Этот бит не вызывает генерацию прерывания, доступен только на чтение и устанавливается в '1', когда регистр TSR пуст. Регистр TSR не отображается на память и не доступен пользователю.

Передача разрешается установкой бита TXEN (TXSTA<5>), но не начнется до тех пор, пока не будут загружены регистр TXREG. Данные появятся на выходе по первому переднему фронту тактового сигнала CK. Выходные данные стабилизируются к заднему фронту тактового сигнала (см. рисунок 12-11). Можно сначала загрузить данные в TXREG и потом установить бит TXEN в '1' (см. рисунок 12-13). Это полезно при низких скоростях передачи данных, когда генератор BRG остановлен, а биты TXEN, CREN, SREN сброшены в нуль. Установка бита TXEN в '1' запустит генератор BRG, который немедленно начнет формировать тактовый сигнал. Обычно после разрешения передачи регистр TSR пуст, и в результате записи в TXREG данные переписываются в TSR, что позволяет реализовать слитную передачу данных.

Сброс бита TXEN в '0' вызовет немедленное прекращение передачи, остановку логики передатчика и переведет выходы CK, DT в третье состояние. Установка бита CREN или SREN во время передачи вызовет ее прекращение и переведет вывод в третье состояние (для приема данных), а вывод CK останется выходом тактового сигнала, если бит CSRC установлен. Логика передатчика не сбрасывается, хотя отключена от вывода, для сброса логики передатчика необходимо очистить бит TXEN. Если бит SREN был установлен в '1', чтобы прервать текущую передачу и принять одиночное слово. Получив слово, бит SREN сбросится в '0', последовательный порт продолжит передачу, если установлен бит TXEN. Линия данных DT переключится из третьего состояния для начала передачи данных. Чтобы это предотвратить, необходимо сбросить бит TXEN в '0'.

Для разрешения 9-разрядной передачи, необходимо установить бит TX9 (TXSTA<6>) в '1'. Девятый бит данных записывается в бит TX9D (TXSTA<0>). Девятый бит данных должен быть указан до записи в регистр TXREG, потому что данные, записанные в регистр TXREG, могут быть сразу загружены в сдвиговый регистр TSR.

Рекомендованная последовательность действий для передачи данных в синхронном ведущем режиме:

1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 12.1).
2. Выбрать синхронный ведущий режим установкой битов SYNC, SPEN и CSRC в '1'.
3. Если необходимо, разрешить прерывания установкой бита TXIE в '1'.
4. Если передача 9-разрядная, установить бит TX9 в '1'.
5. Разрешить передачу установкой бита TXEN в '1'.
6. Если передача 9-разрядная, записать 9-й бит данных в TX9D.
7. Записать данные в регистр TXREG.

Рис. 12-11 Временная диаграмма синхронной передачи двух 8-разрядных слов (SPBRG = 0)

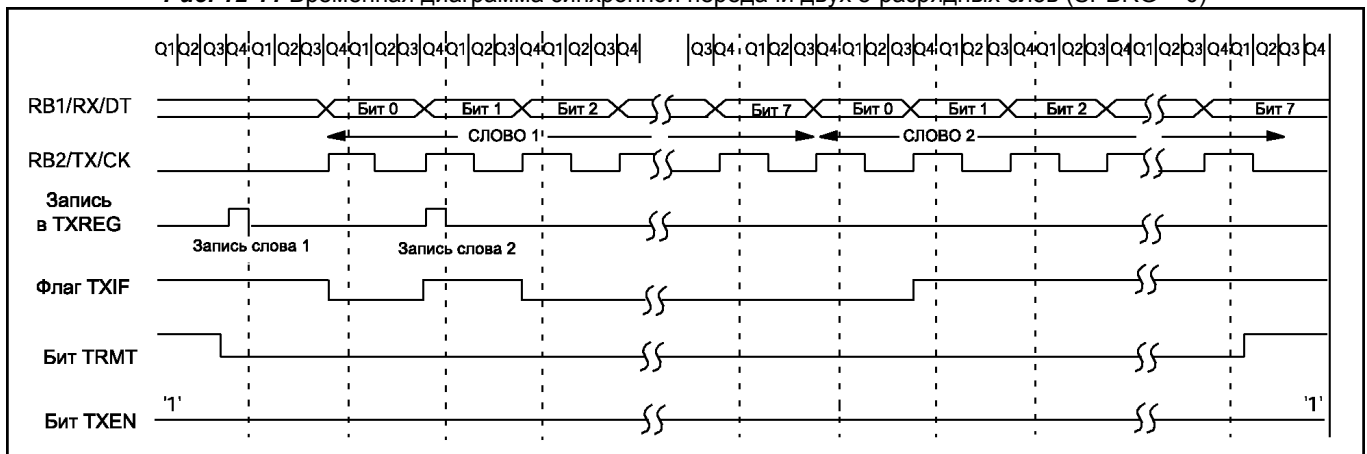
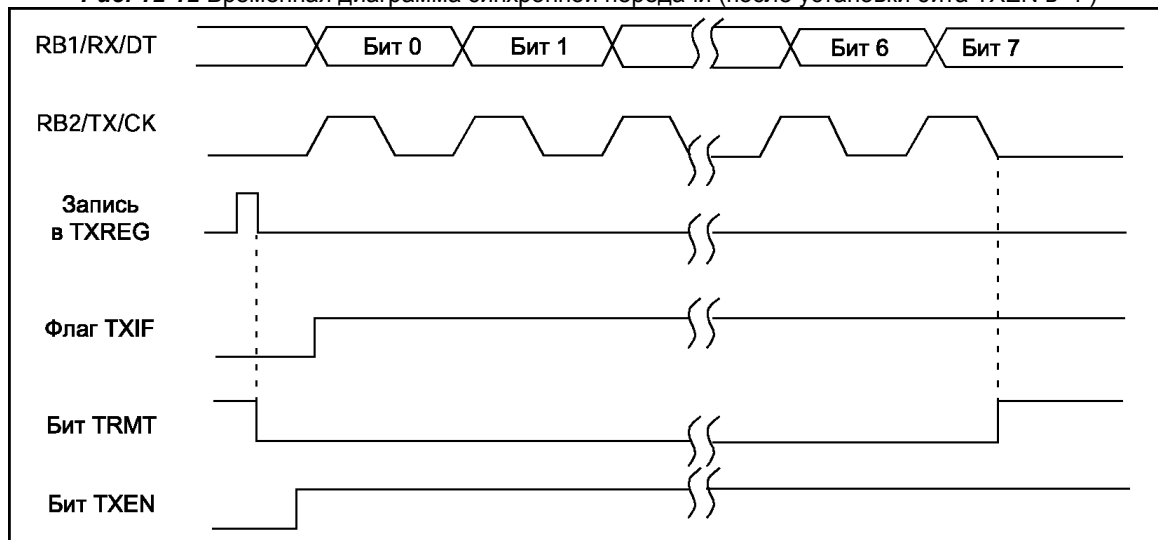


Рис. 12-12 Временная диаграмма синхронной передачи (после установки бита TXEN в '1')**Таблица 12-9** Регистры и биты связанные с работой передатчика USART в синхронном ведущем режиме

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|-----------------------------------|-------|-------|-------|-------|-------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG | Регистр данных передатчика USART | | | | | | | | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 | 0000 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу передатчика USART в синхронном ведущем режиме.

12.4.2 Прием синхронного ведущего

В синхронном режиме прием разрешается установкой битов CREN (RCSTA<4>) или SREN (RCSTA<5>) в '1'. Линия данных RB1/RX/DT опрашивается по заднему фронту тактового сигнала. Если бит SREN установлен в '1', то принимается одиночное слово. Если бит CREN установлен в '1', то в не зависимости от состояния бита SREN будет производиться поточный прием данных. Получив последний бит очередного слова, данные переписываются из RSR в регистр RCREG (если он пуст). После записи в регистр RCREG выставляется флаг прерывания RCIF (PIR1<5>). Прерывание можно разрешить/запретить установкой/сбросом бита RCIE (PIE1<5>). Флаг RCIF доступен только на чтение, сбрасывается аппаратно при чтении из регистра RCREG. Регистр RCREG имеет двойную буферизацию, т.е. представляет собой двухуровневый FIFO. Поэтому можно принять 2 байта данных в FIFO RCREG и третий в регистр RSR. Если FIFO заполнен и обнаружен последний бит третьего байта, устанавливается бит переполнения приемника OERR (RCSTA<1>) в '1'. Байт принятый в RSR будет потерян. Для извлечения двух байт из FIFO необходимо дважды прочитать из регистра RCREG. Бит OERR нужно программно очистить сбросом бита CREN, т.е. запрещением приема. В любом случае, если бит OERR установлен, логика приема отключена.

Девятый бит принятых данных буферизируются так же, как принятые данные. Рекомендуется сначала прочитать регистр RCSTA, затем RCREG, чтобы не потерять бит записанный в RX9D.

Рекомендованные действия при приеме данных в синхронном ведущем режиме:

1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 12.1).
2. Выбрать синхронный ведущий режим установкой битов SYNC, SPEN и CSRC в '1'.
3. Сбросить биты SREN и CREN в '0'.
4. Если необходимо, разрешить прерывания установкой бита RCIE в '1'.
5. Если прием 9-битный, установить бит RX9 в '1'.
6. Если необходимо выполнить одиночный прием установите бит SREN в '1'. Для поточного приема установите бит CREN в '1'.
7. Ожидать установку бита RCIF, или прерывание, если оно разрешено битом RCIE.
8. Считать 9-й бит данных (если разрешен 9-разрядный прием) из регистра RCSTA и проверить возникновение ошибки.
9. Считать 8 бит данных из регистра RCREG.
10. При возникновении ошибки переполнения сбросить бит CREN в '0'.

Рис. 12-13 временная диаграмма синхронного приема в режиме ведущего (SREN = 1, SPBRG = 0)

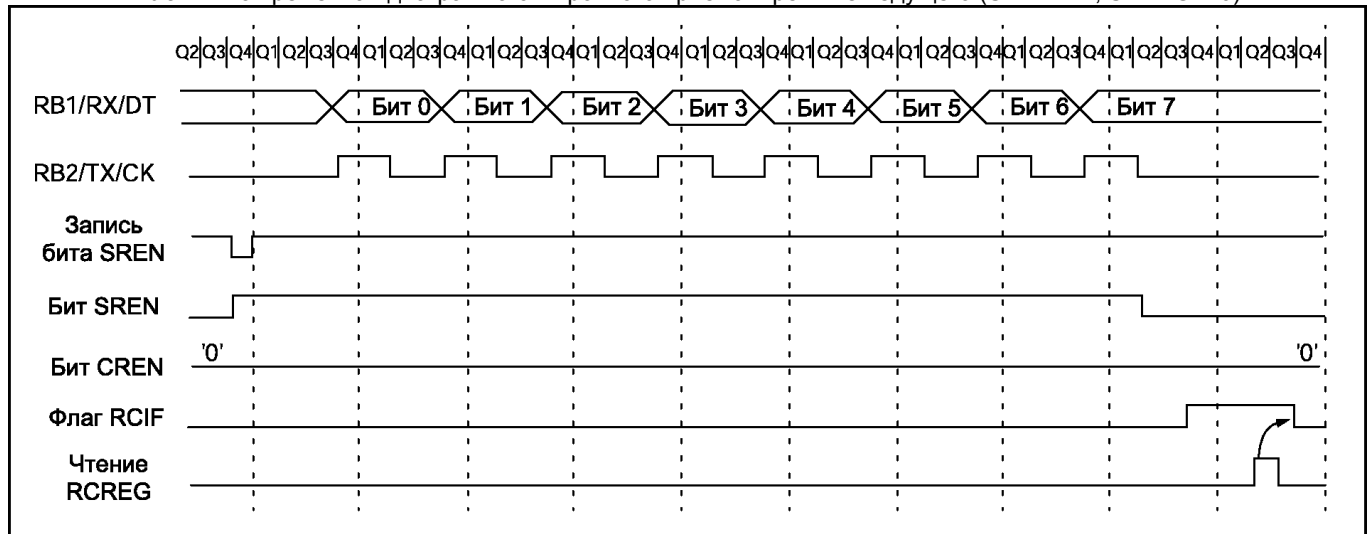


Таблица 12-10 Регистры и биты связанные с работой приемника USART в синхронном ведущем режиме

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|-----------------------------------|-------|-------|-------|-------|-------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 1Ah | RCREG | Регистр данных приемника USART | | | | | | | | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Регистр генератора скорости USART | | | | | | | | 0000 0000 | 0000 0000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу приемника USART в синхронном ведущем режиме.

12.5 Синхронный ведомый режим USART

Режим ведомого отличается от ведущего тем, что микроконтроллер использует тактовый сигнал с входа RB2/TX/CK, а не формирует его самостоятельно. Это позволяет устройству принимать и передавать данные в SLEEP режиме. Выбрать режим ведомого можно сбросом бита CSRC (TXSTA<7>) в '0'.

12.5.1 Передача синхронного ведомого

Работа передатчика в обоих синхронных режимах одинакова, за исключением работы ведомого в SLEEP режиме микроконтроллера.

Если в TXREG были записаны два слова подряд и исполнена команда SLEPP, выполняются следующее действия:

- Первое слово сразу записывается в TSR и передается по мере прихода тактового сигнала.
- Второе слово остается в TXREG
- Флаг TXIF не устанавливается в '1'.
- После передачи первого слова, второе слово передается из TXREG в TSR, и устанавливается флаг TXIF в '1'.
- Если установлен бит TXIE в '1', микроконтроллер выходит из режима SLEEP, происходит переход по вектору 0004h, если GIE=1.

Рекомендованная последовательность действий для передачи данных в синхронном ведомом режиме:

- Выбрать синхронный ведомый режим установкой битов SYNC, SPEN в '1' и сбросом CSRC в '0'.
- Сбросить биты SREN и CREN в '0'.
- Если необходимо, разрешить прерывания установкой бита TXIE в '1'.
- Если передача 9-разрядная, установить бит TX9 в '1'.
- Разрешить передачу установкой бита TXEN в '1'.
- Если передача 9-разрядная, записать 9-й бит данных в TX9D.
- Для начала передачи записать данные в регистр TXREG.

Таблица 12-11 Регистры и биты связанные с работой передатчика USART в синхронном ведомом режиме

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|----------------------------------|-------|-------|-------|-------|-------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG | Регистр данных передатчика USART | | | | | | | | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу передатчика USART в синхронном ведомом режиме.

12.5.2 Прием синхронного ведомого

Работа приемника в обоих синхронных режимах одинакова, кроме работы в режиме SLEEP. В синхронном ведомом режиме не учитывается состояние бита SREN.

Если перед выполнением команды SLEEP был разрешен прием (бит CREN = 1), то модуль USART может принять слово в SLEEP режиме микроконтроллера. По окончании приема данные передаются из регистра RSR в RCREG, и если бит RCIE = 1, микроконтроллер выйдет из режима SLEEP. Если GIE=1, произойдет переход по адресу вектора прерываний 0004h.

Рекомендованные действия при приеме данных в синхронном ведомом режиме:

1. Выбрать синхронный ведомый режим установкой битов SYNC, SPEN в '1' и сбросом CSRC в '0'.
2. Если необходимо, разрешить прерывания установкой бита RCIE в '1'.
3. Если прием 9-битный, установить бит RX9 в '1'.
4. Установите бит CREN в '1' для разрешения приема.
5. Ожидать установку бита RCIF, или прерывание, если оно разрешено битом RCIE.
6. Считать 9-й бит данных (если разрешен 9-разрядный прием) из регистра RCSTA и проверить возникновение ошибки.
7. Считать 8 бит данных из регистра RCREG.
8. При возникновении ошибки переполнения сбросить бит CREN в '0'.

Таблица 12-12 Регистры и биты связанные с работой приемника USART в синхронном ведомом режиме

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|---------|--------|--------------------------------|-------|-------|-------|-------|-------|--------|--------|-----------|---------------|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 1Ah | RCREG | Регистр данных приемника USART | | | | | | | | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | - | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу приемника USART в синхронном ведомом режиме.

13.0 EEPROM память данных

EEPROM память данных доступна для записи/чтения в нормальном режиме работы микроконтроллера во всем диапазоне рабочего напряжения питания (V_{DD}). EEPROM память не отображается на адресное пространство памяти данных, а доступна через регистры специального назначения. Для доступа к EEPROM памяти данных используются 4 регистра специального назначения:

- EECON1
- EECON2 (не физический регистр)
- EEDATA
- EEADR

В регистре EEDATA сохраняются 8-разрядные данные записи/чтения, регистр EEADR содержит адрес регистра EEPROM памяти данных. В микроконтроллерах PIC16F62X реализовано 128 байт EEPROM памяти данных (адреса 00h–7Fh).

EEPROM память данных позволяет выполнить чтение и запись байта. При записи байта происходит автоматическое стирание ячейки и запись новых данных (стирание перед записью). EEPROM память данных рассчитана на большое количество циклов стирание/запись. Время записи управляется интегрированным таймером и зависит от напряжения питания, температуры (см. раздел «электрические характеристики»).

При установке защиты на доступ к EEPROM памяти данных, программа микроконтроллера имеет возможность выполнить запись/чтение EEPROM памяти данных. Доступ закрыт для записи/чтения программатором.

Дополнительную информацию о работе с EEPROM памятью данных смотрите в документации DS33023 «PICmicro™ Mid-Range Reference Manual».

13.1 Регистр EEADR

С помощью регистра EEADR можно адресовать 256 байт EEPROM памяти данных. В микроконтроллерах PIC16F62X реализовано только 128 байт EEPROM памяти данных, поэтому в адресации участвуют только 7 бит регистра EEADR<6:0>.

Старший бит регистра EEADR тоже участвует в декодировании адреса, поэтому всегда должен равняться нулю (для гарантированной адресации памяти в 128 байт).

Регистр EEADR (адрес 9Bh)

| U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|-------|--------|--------|--------|--------|--------|--------|--------|
| - | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 |
| Бит 7 | | | | | | | Бит 0 |

R – чтение бита
 W – запись бита
 U – не реализовано, читается как 0
 -n – значение после POR
 -x – неизвестное значение после POR

бит 7: **Не используется:** читается как '0'

биты 6-0: **EEADR:** Адрес в EEPROM памяти данных для выполнения операций записи/чтения

13.2 Регистры EECON1, EECON2

Регистр EECON1 содержит 4 (младших) физически реализованных управляющих битов. Четыре старших бита не реализованы и читаются как '0'.

Управляющие биты RD и WR инициализируют соответственно чтение и запись данных. Программно эти биты могут быть только установленные в '1', сброс в '0' происходит аппаратно по завершению операции чтения/записи. Защита от программного сброса этих битов позволяет предотвратить преждевременное завершение операции записи.

Если бит WREN=1, то разрешена запись в EEPROM память данных. После сброса по включению питания (POR) бит WREN сбрасывается в '0'. Бит WRERR устанавливается в '1', если во время выполнения записи в EEPROM память данных произошел сброс по сигналу -MCLR или по переполнению сторожевого таймера WDT. Проверив состояние бита WREER, пользователь может повторить запись (регистры EEDATA и EEADR не изменяют своего значения).

После завершения записи в EEPROM память данных устанавливается флаг EEIF (PIR1<>) в '1' (сбрасывается программно).

Регистр EECON2 не реализован физически, читается как 00h. Он используется в операциях записи в EEPROM память данных.

Регистр EECON1 (адрес 9Ch)

| U-0 | U-0 | U-0 | U-0 | R/W-x | R/W-0 | R/S-0 | R/S-x |
|-------|-----|-----|-----|--------------|-------------|-----------|-----------|
| - | - | - | - | WRERR | WREN | WR | RD |
| Бит 7 | | | | Бит 0 | | | |

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

биты 7-4: **Не используются:** читаются как '0'

бит 3: **WRERR:** Флаг ошибки записи в EEPROM память данных
1 = запись прервана (произошел один из сбросов: по сигналу -MCLR, по переполнению WDT, по снижению напряжения питания BOR)
0 = запись завершена

бит 2: **WREN:** Разрешение записи в EEPROM память данных
1 = запись разрешена
0 = запись запрещена

бит 1: **WR:** Инициализировать запись в EEPROM память данных (программно может быть только установлен в '1')
1 = инициализировать запись (сбрасывается в '0' аппаратно)
0 = запись завершена

бит 0: **RD:** Инициализировать чтение из EEPROM памяти данных (программно может быть только установлен в '1')
1 = инициализировать чтение (сбрасывается в '0' аппаратно)
0 = чтение завершено

13.3 Чтение из EEPROM памяти данных

Для чтения EEPROM памяти данных необходимо записать адрес в регистр EEADR и установить бит RD (EECON1<0>) в '1'. В следующем машинном цикле данные доступны для чтения из регистра EEDATA. Прочитанное значение из EEPROM памяти данных будет храниться в регистре EEDATA до следующего чтения или записи в этот регистр по команде микроконтроллера.

Пример 13-1 Чтение из EEPROM памяти данных

```
BSF    STATUS, RP0    ; Выбрать банк 1
BCF    STATUS, RP1
MOVLW  CONFIG_ADDR  ;
MOVWF  EEADR         ; Адрес считываемого регистра
BSF    EECON1, RD    ; Чтение
MOVF   EEDATA, W     ; W = EEDATA
```

13.4 Запись в EEPROM память данных

Для записи в EEPROM память данных необходимо записать адрес в регистр EEADR, данные в регистр EEDATA и выполнить последовательность команд, показанных в примере 13-2.

Пример 13-2 Запись в EEPROM память данных

| | | | |
|------------------------------------|-------|--------------|--|
| | BSF | STATUS, RP0 | ; Выбрать банк 1 |
| | BSF | EECON1, WREN | ; Разрешить запись |
| Обязательная последовательность | BCF | INTCON, GIE | ; Запретить прерывания |
| | MOVLW | 55h | ; |
| | MOVWF | EECON2 | ; Записать 55h |
| | MOVLW | AAh | ; |
| | MOVWF | EECON2 | ; Записать AAh |
| | BSF | EECON1, WR | ; Установить бит WR ; для начала записи |
| | BSF | INTCON, GIE | ; Разрешить прерывания |

Запись байта не будет произведена, если не выполнена указанная последовательность (запись 55h в EECON2, запись AAh в EECON2, установка бита WR в '1'). Рекомендуется запрещать прерывания при выполнении обязательной последовательности команд. Если во время выполнения указанной последовательности произойдет переход по вектору прерывания, запись байта выполнена не будет.

Чтобы разрешить запись в EEPROM память данных, необходимо установить бит WREN (EECON1<2>) в '1', защищающий от случайной записи. Пользователь должен установить бит WREN в '1' перед началом записи, а после окончания записи сбросить его в '0' (аппаратно бит WREN в '0' не сбрасывается).

После инициализации записи сброс бита WREN в '0' не повлияет на цикл записи, но установка бита WR в '1' будет запрещена, пока WREN = 0.

По окончании записи бит WR аппаратно сбрасывается в '0', а флаг прерывания EEIF устанавливается в '1'. Пользователь может использовать прерывания для проверки окончания записи в EEPROM память данных. Флаг EEIF сбрасывается в '0' программно.

13.5 Проверка записи

Рекомендуется после выполнения операции записи в EEPROM память данных произвести контрольное чтение (см. пример 13-3). Основные ошибки возникают при записи отдельных битов равных 1, чтение будет давать результат 0.

Пример 13-3 Проверка записи

| | | | |
|--|-------|-------------|--|
| | BSF | STATUS, RP0 | ; Выбрать банк 1 |
| | : | : | ; Текст программы |
| | : | : | ; |
| | MOVF | EEDATA, W | ; Чтение записываемых данных |
| | BSF | EECON1, RD | ; Инициализация чтения из EEPROM |
| | : | : | ; записанных данных |
| | : | : | ; |
| | : | : | ; Проверить, равно значение в регистре W |
| | : | : | ; и прочитанные данные из EEPROM (EEDATA)? |
| | : | : | ; |
| | SUBWF | EEDATA, W | ; |
| | BTFSS | STATUS, Z | ; Результат 0? |
| | GOTO | WRITE_ERR | ; НЕТ, данные записаны неправильно |
| | : | : | ; ДА, данные записаны правильно |
| | : | : | ; Продолжение программы |

13.6 Защита от случайной записи в EEPROM память данных

Существует несколько условий, когда запись байта в EEPROM память данных не выполняется:

1. После сброса по включению питания POR бит WREN = 0.
2. Таймер включения питания (в течение 72мс) запрещает запись в EEPROM память данных.
3. Обязательная последовательность инициализации записи и бит WREN предотвращают случайную запись.

13.7 Операции с EEPROM памятью при установленном бите защиты

При установке защиты на доступ к EEPROM памяти данных, программа микроконтроллера имеет возможность выполнить запись/чтение EEPROM памяти данных. Доступ закрыт для записи/чтения программатором.

Таблица 13-1 Регистры и биты связанные с работой EEPROM памяти данных

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы | |
|---------|--------|--|-----------------------|-------|-------|-------|-------|--------|--------|-----------|---------------|-----------|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u | |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 | |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 | |
| 9Ah | EEDATA | Регистр данных EEPROM | | | | | | | | xxxx xxxx | uuuu uuuu | |
| 9Bh | EEADR | - | Регистр адреса EEPROM | | | | | | | | -xxx xxxx | -uuu uuuu |
| 9Ch | EECON1 | - | - | - | - | WRERR | WREN | WR | RD | ---- x000 | ---- q000 | |
| 9Dh | EECON2 | Управляющий регистр 2 EEPROM (физически не реализован) | | | | | | | | ---- ---- | ---- ---- | |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу EEPROM память данных.

14.0 Особенности микроконтроллеров PIC16F62X

В настоящее время, устройства работающие в режиме реального времени часто содержат микроконтроллер как основной элемент схемы. PIC16F62X имеют много усовершенствований повышающие надежность системы, снижающие стоимость устройства и число внешних компонентов. Микроконтроллеры PIC16F62X имеют режимы энергосбережения и возможность защиты кода программы.

Основные достоинства:

1. Выбор тактового генератора.
2. Сброс:
 - сброс по включению питания (POR);
 - таймер включения питания (PWRT);
 - таймер запуска генератора (OSC);
 - сброс по снижению напряжения питания (BOR).
3. Прерывания.
4. сторожевой таймер (WDT).
5. Режим энергосбережения (SLEEP).
6. Защита кода программы.
7. Область памяти для идентификатора.
8. Внутрисхемное программирование по последовательному порту (ICSP).

В микроконтроллере PIC16F62X встроен сторожевой таймер WDT, который может быть выключен только в битах конфигурации микроконтроллера. Для повышения надежности сторожевой таймер WDT имеет собственный RC генератор. Дополнительно два таймера выполняют задержку старта работы микроконтроллера. Первый таймер запуска генератора (OST), удерживает микроконтроллер в состоянии сброса, пока не стабилизируется частота тактового генератора. Второй таймер включения питания (PWRT), срабатывает после включения питания и удерживает микроконтроллер в состоянии сброса в течение 72мс (типичное значение), пока не стабилизируется напряжение питания. Также встроена схема сброса микроконтроллера при снижении напряжения питания, удерживая состояние сброса в течение 72 мс (типичное значение). В большинстве приложений эти функции микроконтроллера позволяют исключить внешние схемы сброса.

Режим SLEEP предназначен для обеспечения сверхнизкого энергопотребления. Микроконтроллер может выйти из режима SLEEP по сигналу внешнего сброса, по переполнению сторожевого таймера или при возникновении прерываний. Выбор режима работы тактового генератора позволяет использовать микроконтроллеры в различных приложениях. Режим тактового генератора ER позволяет уменьшить стоимость устройства, а режим LP снизить энергопотребление. Биты конфигурации микроконтроллера используются для указания режима его работы.

14.1 Биты конфигурации

Биты конфигурации расположены в памяти программ по адресу 2007h, они могут быть запрограммированы в '0' или оставленными в '1'. Заметьте, что адрес 2007h расположен за пределами пользовательской памяти программ. Фактически, к конфигурационному регистру (область памяти 2000h – 3FFFh) можно обратиться только в режиме программирования микроконтроллера.

Слово конфигурации (адрес 2007h)

| CP1 | CP0 | CP1 | CP0 | - | CPD | LVP | BODEN | MCLRE | FOSC2 | -PWRT | WDTE | FOSC1 | FOSC0 |
|-----|-----|-----|-----|---|-----|-----|-------|-------|-------|-------|------|-------|-------|
|-----|-----|-----|-----|---|-----|-----|-------|-------|-------|-------|------|-------|-------|

Бит 13

Бит 0

биты 13-10:**CP1:CP0:** Биты защита памяти программ ⁽²⁾

Микроконтроллеры с памятью программ 2к

11 = защита памяти программ выключена

10 = защищена память программ с адресами 0400h-07FFh

01 = защищена память программ с адресами 0200h-07FFh

00 = защищена память программ с адресами 0000h-07FFh

Микроконтроллеры с памятью программ 1к

11 = защита памяти программ выключена

10 = защита памяти программ выключена

01 = защищена память программ с адресами 0200h-03FFh

00 = защищена память программ с адресами 0000h-03FFh

бит 9: **Не реализован:** читается как '1'

бит 8: **CPD:** Бит защиты EEPROM памяти данных ⁽³⁾

1 = защита памяти данных выключена

0 = защита памяти данных включена

бит 7: **LVP:** Бит разрешения низковольтного программирования

1 = вывод RB4/PGM работает как PGM, режим низковольтного программирования включен

0 = вывод RB4/PGM работает как цифровой порт ввода/вывода, вывод HV используется для программирования микроконтроллера

бит 6: **BODEN:** Бит разрешения сброса по снижению напряжения питания ⁽¹⁾

1 = разрешен сброс BOR

0 = запрещен сброс BOR

бит 5: **MCLRE:** Бит выбора режима работы вывода RA5/-MCLR

1 = RA5/-MCLR работает как -MCLR

0 = RA5/-MCLR работает как цифровой порт ввода/вывода, используется внутренний сброс -MCLR

бит 3: **-PWRT:** Бит разрешения работы таймера включения питания ⁽¹⁾

1 = PWRT выключен

0 = PWRT включен

бит 2: **WDTE:** Бит разрешения работы сторожевого таймера

1 = WDT включен

0 = WDT выключен

биты 4,1,0:**FOSC2:FOSC0:** Биты выбора режима тактового генератора ⁽⁴⁾

111 = ER генератор: вывод RA6/OSC2/CLKOUT работает как CLKOUT, резистор подключается к выводу RA7/OSC1/CLKIN

110 = ER генератор: вывод RA6/OSC2/CLKOUT работает как цифровой порт ввода/вывода, резистор подключается к выводу RA7/OSC1/CLKIN

101 = INTRC генератор: вывод RA6/OSC2/CLKOUT работает как CLKOUT, вывод RA7/OSC1/CLKIN работает как цифровой порт ввода/вывода

100 = INTRC генератор: вывод RA6/OSC2/CLKOUT работает как цифровой порт ввода/вывода, вывод RA7/OSC1/CLKIN работает как цифровой порт ввода/вывода

011 = EC генератор: вывод RA6/OSC2/CLKOUT работает как цифровой порт ввода/вывода, вывод RA7/OSC1/CLKIN работает как CLKIN

010 = HS генератор: высокочастотный резонатор подключается к выводам RA6/OSC2/CLKOUT, RA7/OSC1/CLKIN

001 = XT генератор: резонатор подключается к выводам RA6/OSC2/CLKOUT, RA7/OSC1/CLKIN

000 = LP генератор: резонатор подключается к выводам RA6/OSC2/CLKOUT, RA7/OSC1/CLKIN

Примечания:

1. При возникновении сброса по снижению напряжения питания (BOR) автоматически запускается таймер PWRT, независимо от состояния бита -PWRT.
2. Чтобы установить защиту памяти программ, все пары CP1:CP0 должны иметь одинаковое значение.
3. После выключения защиты EEPROM памяти данных вся информация будет стерта.
4. Когда сигнал MCLR удерживается в активном уровне, внутренний тактовый генератор

14.2 Настройка тактового генератора

14.2.1 Режимы тактового генератора

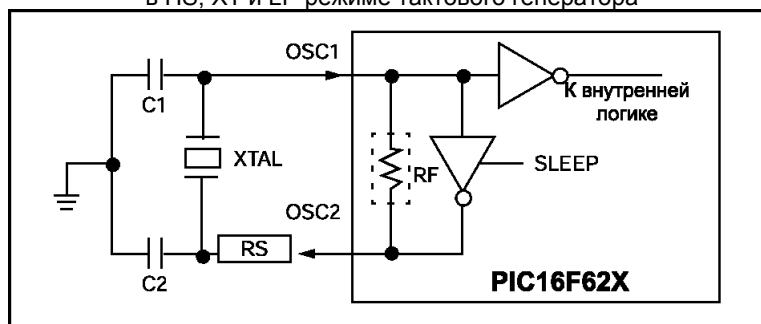
Микроконтроллеры PIC16F62X могут работать в одном из восьми режимов тактового генератора. Указать режим тактового генератора можно при программировании микроконтроллера в битах конфигурации (FOSC2:FOSC0):

- LP – низкочастотный резонатор;
- XT – обычный резонатор;
- HS – высокочастотный резонатор;
- ER – внешний резистор (2 режима);
- INTRC – внутренняя RC цепочка (2 режима);
- EC – внешний тактовый сигнал.

14.2.2 Кварцевый/керамический резонатор

В режимах тактового генератора XT, LP и HS кварцевый или керамический резонатор подключается к выводам OSC1, OSC2 (см. рисунок 14-1). Для микроконтроллеров PIC16F62X нужно использовать резонаторы с параллельным резонансом. Использование резонаторов с последовательным резонансом может привести к получению тактовой частоты не соответствующей параметрам резонатора. В режимах XT, LP и HS микроконтроллер может работать от внешнего источника тактового сигнала OSC1 (см. рисунок 14-2).

Рис. 14-1 Подключение кварцевого/керамического резонатора в HS, XT и LP режиме тактового генератора



Примечание. Для некоторых типов резонаторов может потребоваться последовательно включенный резистор.

Рис. 14-2 Подключение внешнего тактового сигнала в HS, XT и LP режиме тактового генератора

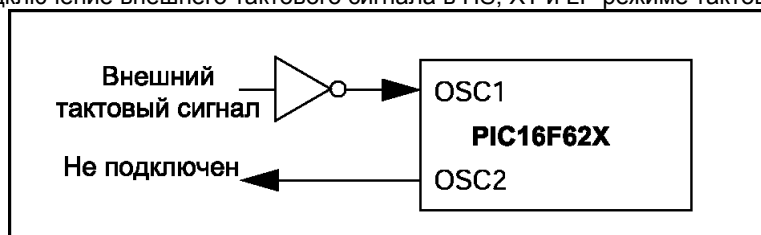


Таблица 14-1 Параметры конденсаторов для керамического резонатора

| Режим | Частота | OSC1(C1) | OSC2(C2) | Большая емкость увеличивает стабильность генератора, но также увеличивается и время запуска. Значения емкости конденсаторов указанные в таблице являются оценочными, т.к. каждый резонатор имеет собственные характеристики. Проконсультируйтесь у производителя резонаторов для правильного подбора внешних компонентов. |
|-------|----------|----------|----------|---|
| XT | 455 кГц | 22-100пФ | 22-100пФ | |
| | 2.0 МГц | 15-68пФ | 15-68пФ | |
| | 4.0 МГц | 15-68пФ | 15-68пФ | |
| HS | 8.0 МГц | 10-68пФ | 10-68пФ | |
| | 16.0 МГц | 10-22пФ | 10-22пФ | |

Таблица 14-2 Параметры конденсаторов для кварцевого резонатора

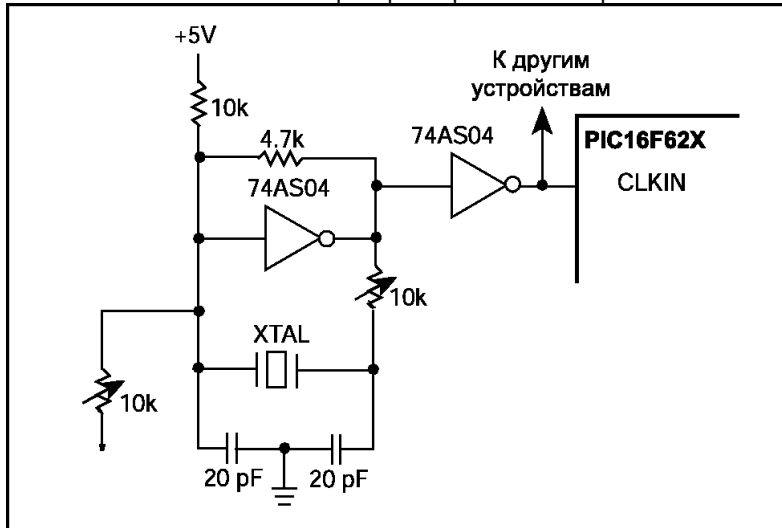
| Режим | Частота | OSC1(C1) | OSC2(C2) | Большая емкость увеличивает стабильность генератора, но также увеличивается и время запуска. Значения емкости конденсаторов указанные в таблице являются оценочными. Последовательный резистор Rs может потребоваться в HS и XT режиме для предотвращения возбуждения резонатора на низкой частоте, т.к. каждый резонатор имеет собственные характеристики. Проконсультируйтесь у производителя резонаторов для правильного подбора внешних компонентов. |
|-------|---------|----------|----------|--|
| LP | 32 кГц | 68-150пФ | 68-150пФ | |
| | 200 кГц | 15-30пФ | 15-30пФ | |
| XT | 100 кГц | 68-150пФ | 68-150пФ | |
| | 2 МГц | 15-30пФ | 15-30пФ | |
| | 4 МГц | 15-30пФ | 15-30пФ | |
| HS | 8 МГц | 15-30пФ | 15-30пФ | |
| | 10 МГц | 15-30пФ | 15-30пФ | |
| | 20 МГц | 15-30пФ | 15-30пФ | |

14.2.3 Внешний тактовый генератор

В качестве внешнего тактового генератора можно использовать готовый генератор, либо собрать простую схемы с ТТЛ выходом. Качественный кварцевый резонатор обеспечивает высокую эффективность ТТЛ схемы. Существует две основных схемы включения кварцевых резонаторов: с параллельным резонансом, с последовательным резонансом.

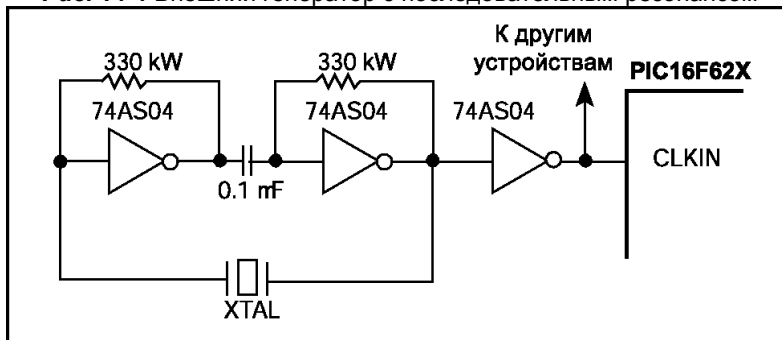
На рисунке 14-3 показана типовая схема генератора с параллельным резонансом, предназначенная для работы на основной частоте кварцевого резонатора. Инвертор 74AS04 производит необходимый для параллельного резонанса сдвиг фазы на 180°. Для обеспечения стабильности схемы в отрицательной обратной связи включен резистор 47кОм. Потенциометр 10кОм предназначен для смещения рабочей точки инвертора в линейную область.

Рис. 14-3 Внешний генератор с параллельным резонансом



На рисунке 14-4 показана типовая схема генератора с последовательным резонансом, тоже предназначенная для работы на основной частоте кварцевого резонатора. Инверторы выполняют сдвиг фазы на 180°. Резисторы 330кОм создают отрицательную обратную связь для смещения рабочих точек инверторов в линейную область.

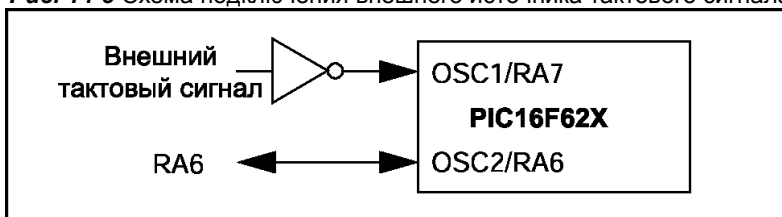
Рис. 14-4 Внешний генератор с последовательным резонансом



14.2.4 Внешний тактовый сигнал

Если в устройстве генерируется тактовый сигнал, он может использоваться для управления PIC16F62X при условии, что внешний тактовый сигнал удовлетворяет требованиям раздела 17.4. На рисунке 14-5 показана схема подключения внешнего источника тактового сигнала.

Рис. 14-5 Схема подключения внешнего источника тактового сигнала

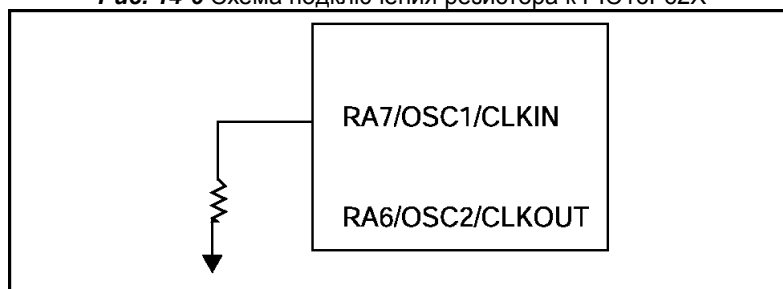


14.2.5 ER генератор

В приложениях, не требующих высокостабильной тактовой частоты, возможно использовать ER режим генератора уменьшающего стоимость устройства. Для выбора тактовой частоты генератора к выводу CLKIN подключается только один резистор, создающий постоянный ток смещения. На частоту тактового генератора также влияет напряжение питания и температура. Т.к. частота тактового генератора управляется постоянным током, то тип резистора, паразитная емкость и топология проводника не влияют на частоту генератора.

На рисунке 14-6 показана схема подключения резистора к PIC16F62X. Для сопротивления резистора меньше 38кОм частота тактового генератора может быть нестабильна или генерация может прекратиться. Для очень большого сопротивления (больше 1МОм) генератор тактового сигнала становится чувствителен к внешним помехам, токам утечки и влажности. Рекомендуется выбирать сопротивления резисторов от 38кОм до 1МОм.

Рис. 14-6 Схема подключения резистора к PIC16F62X



В разделе «электрические характеристики» приводятся соотношения частоты тактового генератора и рабочей температуры для различных значений R и напряжения питания V_{DD} .

В ER режиме тактового генератора вывод OSC2 может работать как цифровой порт ввода/вывода или выход тактового сигнала $F_{osc}/4$ для испытаний (синхронизации).

Ориентировочное значение частоты тактового генератора ($V_{DD}=5В @ 25^{\circ}C$)

| R | F_{osc} |
|--------|-----------|
| 40кОм | 3708кГц |
| 60кОм | 2620кГц |
| 100кОм | 1696кГц |
| 220кОм | 816кГц |

| R | F_{osc} |
|--------|-----------|
| 360кОм | 512кГц |
| 470кОм | 400кГц |
| 680кОм | 284кГц |
| 1МОм | 192кГц |

14.2.6 Внутренний RC генератор 4МГц

Внутренний тактовый генератор формирует тактовый сигнал с частотой 4МГц (номинальное значение) при напряжении питания $V_{DD}=5В$ и температуре $25^{\circ}C$. Графики зависимости частоты внутреннего RC генератора от температуры и напряжения питания смотрите в разделе «электрические характеристики».

14.2.7 CLKOUT

В микроконтроллерах PIC16F62X на вывод CLKOUT может выдаваться частота тактового сигнала $F_{osc}/4$ (устанавливается в битах конфигурации) для испытательных целей или синхронизации внешней логики.

14.3 Программный выбор частоты тактового генератора

Программный выбор частоты тактового генератора доступен в режимах ER или INTRC микроконтроллеров PIC16F62X. Эта особенность позволит динамически переключать частоту тактового генератора между 4МГц и 37кГц. В ER режиме генератора частота будет зависеть от значения внешнего резистора при выборе 4МГц, а при выборе 37кГц значение внешнего резистора не влияет на частоту. Программный выбор частоты тактового генератора подходит для приложений, использующие режим энергосбережения без перехода в режим SLEEP. Бит OSCF в регистре PCON управляет работой тактового генератора (см. раздел 4.2.2.6).

14.4 Сброс

PIC16F62X различает следующие виды сбросов:

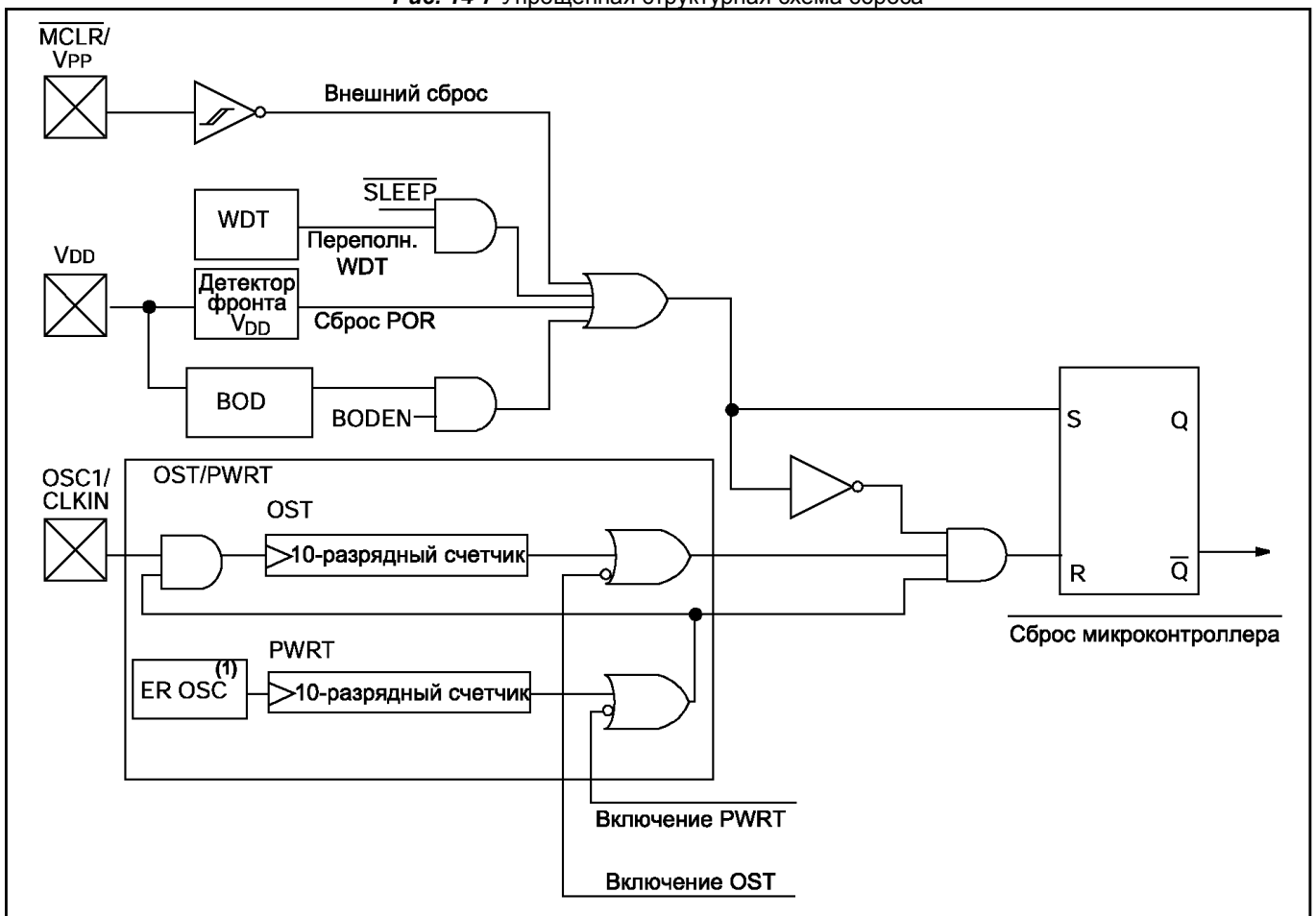
- a) сброс по включению питания POR;
- b) сброс по сигналу -MCLR в нормальном режиме работы;
- c) сброс по сигналу -MCLR в SLEEP режиме;
- d) сброс от WDT в нормальном режиме работы;
- e) сброс от WDT в режиме SLEEP;
- f) сброс по снижению напряжения питания BOR.

Некоторые регистры не изменяются после любого вида сброса, но после сброса по включению питания POR они содержат неизвестное значение. Большинство регистров сбрасываются в начальное состояние при сбросах POR, -MCLR и WDT в нормальном режиме, -MCLR в режиме SLEEP. Сброс WDT в SLEEP режиме рассматривается как возобновление нормальной работы и на значение регистров не влияет. Биты -TO и -PD принимают определенные значения при различных видах сброса (см. таблицу 14-4). Программное обеспечение может использовать эти биты для детектирования вида сброса микроконтроллера. Состояние регистров специального назначения после сброса смотрите в таблице 14-7.

Упрощенная структурная схема сброса показана на рисунке 14-7.

На входе -MCLR есть внутренний фильтр, не пропускающий короткие импульсы.

Рис. 14-7 Упрощенная структурная схема сброса



Примечание 1. Отдельный генератор для INTRC/ER режима.

14.5 POR, PWRT, OST, BOD

14.5.1 Сброс по включению питания POR

Интегрированная схема POR удерживает микроконтроллер в состоянии сброса, пока напряжение V_{DD} не достигнет требуемого уровня. Для включения схемы POR необходимо соединить вывод -MCLR с V_{DD} через резистор, не требуя внешней RC цепочки, обычно используемой для сброса. Максимальное время нарастания V_{DD} смотрите в разделе «электрические характеристики». Схема POR не выполняет сброс микроконтроллера при снижении напряжения V_{DD} .

Когда микроконтроллер переходит в режим нормальной работы из состояния сброса, рабочие параметры (напряжение питания, частота, температура и т.д.) должны соответствовать указанным в разделе «электрические характеристики». Если рабочие параметры не удовлетворяют требованиям, микроконтроллер должен находиться в состоянии сброса.

Дополнительную информацию смотрите в документации AN607 «Power-up Trouble Shooting».

14.5.2 Таймер включения питания PWRT

Таймер включения питания обеспечивает задержку в 72мс по сигналу схемы сброса POR или BOD. Таймер включения питания работает от внутреннего RC генератора и удерживает микроконтроллер в состоянии сброса по активному сигналу от PWRT. Задержка PWRT позволяет достигнуть напряжению V_{DD} номинального значения.

Битом -PWRTЕ в слове конфигурации можно выключить (-PWRTЕ=1) или включить (-PWRTЕ=0) таймер включения питания. Время задержки PWRT варьируется в каждом микроконтроллере и зависит от напряжения питания и температуры (см. раздел «электрические характеристики»).

14.5.3 Таймер запуска генератора OST

Таймер запуска генератора обеспечивает задержку в 1024 такта генератора (вход OSC1) после окончания задержки от PWRT. Это гарантирует, что частота кварцевого/керамического резонатора стабилизировалась. Задержка OST включается только в режимах HS, XT и LP тактового генератора после сброса POR или выхода микроконтроллера из режима SLEEP.

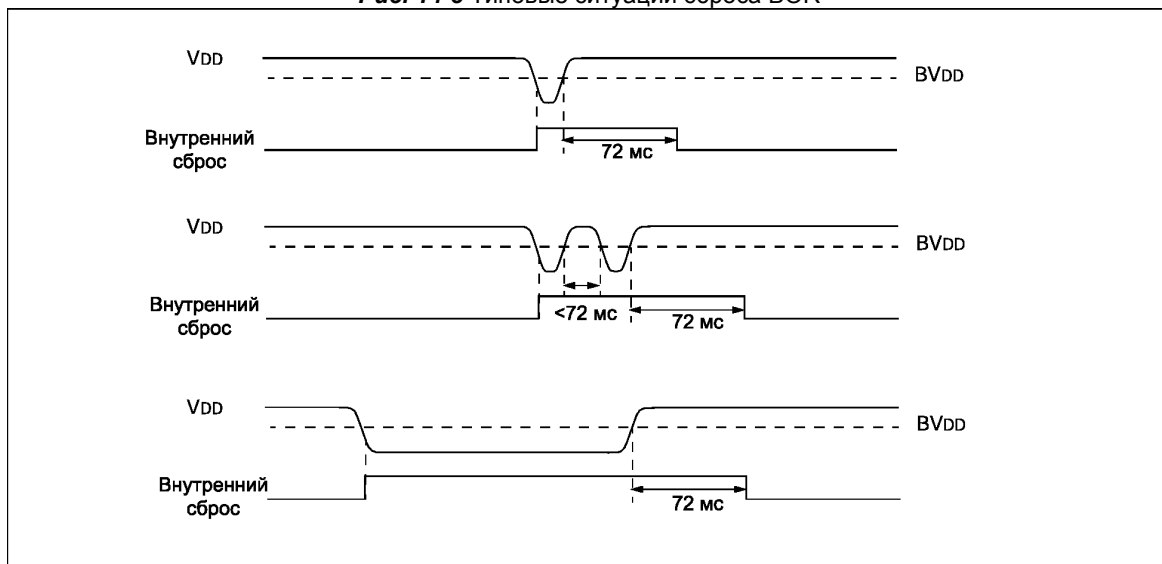
14.5.4 Детектор пониженного напряжения питания BOD

В микроконтроллерах PIC16F62X встроена схема детектора снижения напряжения питания. Битом BODEN в слове конфигурации можно выключить (BODEN = 0) или включить (BODEN = 1) детектор снижения напряжения питания. Если напряжение V_{DD} опускается ниже 4.0В (см. параметр D005 V_{BOD} в таблице 17-1) на время больше(или равное) T_{BOD} , произойдет сброс по снижению напряжения питания. Если длительность снижения напряжения питания меньше T_{BOD} , сброс микроконтроллера не произойдет.

При любом виде сброса (POR, -MCLR, WDT и т.д.) микроконтроллер находится в состоянии сброса, пока напряжение V_{DD} не будет выше V_{BOD} . После нормализации напряжения питания микроконтроллер находится в состоянии сброса еще 72мс.

Если напряжение питания V_{DD} стало ниже V_{BOD} во время работы таймера по включению питания, микроконтроллер возвращается в состояние сброса BOR, а таймер инициализируется заново. Каждый переход напряжения питания V_{DD} через границу V_{BOD} инициализирует PWRT, создавая задержку в 72мс. При включении схемы BOD всегда нужно включать таймер PWRT. На рисунки 14-8 показаны типовые ситуации сброса BOR.

Рис. 14-8 Типовые ситуации сброса BOR



14.5.5 Последовательность удержания микроконтроллера в состоянии сброса

При включении питания выполняется следующая последовательность удержания микроконтроллера в состоянии сброса: сброс POR, задержка PWRT, задержка OST. Полное время задержки изменяется в зависимости от режима работы тактового генератора и состояния бита -PWRT. Например, в режиме ER генератора и при -PWRT=0 задержка будет отсутствовать. На рисунках 14-9, 14-10 и 14-11 показаны последовательности удержания микроконтроллера в состоянии сброса. Удержание микроконтроллера в состоянии сброса происходит после сброса POR. Если сигнал -MCLR удерживается в низком уровне достаточно долго (дольше времени всех задержек), после перехода -MCLR в высокий уровень программа начнет выполняться немедленно (см. рисунок 14-10). Это может быть полезно при одновременном запуске нескольких микроконтроллеров, работающих параллельно.

В таблице 14-6 показано состояние некоторых регистров специального назначения, а в таблице 14-7 состояние всех регистров после сбросов.

14.5.6 Регистр PCON

Регистр PCON (адрес 8Eh) содержит два бита статуса питания.

Бит 0 – -BOD (детектор пониженного напряжения питания)

Бит -BOD имеет неопределенное значение после сброса POR. Пользователь должен программно установить бит -BOD в '1' и проверять его состояние при возникающих сбросах микроконтроллера. Если -BOD =0, то произошел сброс по снижению напряжения питания (BOR). Бит -BOD не устанавливается в '1' аппаратно и имеет непредсказуемое значение, если детектор пониженного напряжения питания выключен (BODEN=0).

Бит 1 – -POR (сброс по включению питания)

Бит сбрасывается в '0' при возникновении сброса POR. Пользователь должен программно установить этот бит в '1' после сброса по включению питания. При последующих сбросах, если -POR=0, то произошел сброс по включению питания (или напряжение V_{DD} стало слишком низким).

Таблица 14-3 Время задержки при различных видах сброса

| Режим генератора | Сброс POR | | Сброс BOR | Выход из режима SLEEP |
|------------------|-----------------------|----------------|-----------------------|-----------------------|
| | PWRT=0 | PWRT=1 | | |
| XT, HS, LP | 72мс + 1024 T_{OSC} | 1024 T_{OSC} | 72мс + 1024 T_{OSC} | 1024 T_{OSC} |
| ER | 72мс | - | 72мс | - |

Таблица 14-4 Состояние некоторых битов регистров STATUS/PCON

| -POR | -BOD | -TO | -PD | |
|------|------|-----|-----|--|
| 0 | x | 1 | 1 | Сброс по включению питания |
| 0 | x | 0 | x | Недействительный -TO, если установлен -POR |
| 0 | x | x | 0 | Недействительный -PD, если установлен -POR |
| 1 | 0 | x | x | Сброс по снижению напряжения питания |
| 1 | 1 | 0 | u | Сброс от WDT |
| 1 | 1 | 0 | 0 | Выход из режима SLEEP от WDT |
| 1 | 1 | u | u | Сброс -MCLR при нормальном режиме работы |
| 1 | 1 | 1 | 0 | Сброс -MCLR в SLEEP режиме |

Обозначения: u = не изменяется; x = неопределенное значение

Таблица 14-5 Регистры и биты связанные с детектированием сброса микроконтроллера

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|-------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-----------|---------------|
| 03h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx | 000q q000 |
| 8Eh | PCON | - | - | - | - | OSCF | - | -POR | -BOD | ---- 1-0X | ---- 1-uq |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечания

1. Затененные биты не используются при детектировании сброса микроконтроллера.
2. Другие сбросы: -MCLR, BOR, переполнение WDT в нормальном режиме работы.

Таблица 14-6 Состояние особых регистров после сброса

| Вид сброса | Счетчик команд PC | Регистр STATUS | Регистр PCON |
|--|-----------------------|----------------|--------------|
| Сброс по включению питания | 000h | 0001 1xxx | ---- 1-0x |
| Сброс по сигналу -MCLR в нормальном режиме | 000h | 000u uuuu | ---- 1-uu |
| Сброс по сигналу -MCLR в SLEEP режиме | 000h | 0001 0uuu | ---- 1-uu |
| Сброс от WDT | 000h | 0000 uuuu | ---- 1-uu |
| Выход из режима SLEEP от WDT | PC + 1 | uuu0 0uuu | ---- u-uu |
| Сброс по снижению напряжения питания | 000h | 000x xuuu | ---- 1-u0 |
| Выход из режима SLEEP от прерываний | PC + 1 ⁽¹⁾ | uuu1 0uuu | ---- u-uu |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно.

Примечание 1. При выходе из режима SLEEP по возникновению прерывания, если GIE=1, в счетчик команд PC загружается вектор прерываний (0004h) после выполнения PC+1.

Таблица 14-7 Состояние регистров специального назначения после сброса

| Регистр | Адрес | Сброс по включению питания | - Сброс по сигналу -MCLR в нормальном режиме - Сброс по сигналу -MCLR в SLEEP режиме - Сброс от WDT - Сброс по снижению напряжения питания | - Выход из режима SLEEP от прерываний - Выход из режима SLEEP от WDT |
|---------|-------|----------------------------|---|---|
| W | - | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | 00h | - | - | - |
| TMR0 | 01h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 02h | 0000 0000 | 0000 0000 | PC+1 ⁽³⁾ |
| STATUS | 03h | 0001 1xxx | 000q quuu ⁽⁴⁾ | uuuq quuu ⁽⁴⁾ |
| FSR | 04h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | 05h | xxxx 0000 | xxxx 0000 | uuuu uuuu |
| PORTB | 06h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | 10h | --00 0000 | --uu uuuu | --uu uuuu |
| T2CON | 12h | -000 0000 | -000 0000 | -uuu uuuu |
| CCP1CON | 17h | --00 0000 | --00 0000 | --uu uuuu |
| RCSTA | 18h | 0000 000x | 0000 000x | uuuu uuuu |
| CMCON | 1Fh | 0000 0000 | 0000 0000 | uuuu uuuu |
| PCLATH | 0Ah | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 0Bh | 0000 000x | 0000 000u | uuuu uqqq ⁽²⁾ |
| PIR1 | 0Ch | 0000 -000 | 0000 -000 | uuuu -uuu ^(2,5) |
| OPTION | 81h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 85h | 11-1 1111 | 11-1 1111 | uu-u uu |
| TRISB | 86h | 1111 1111 | 1111 1111 | uuuu uuuu |
| PIE1 | 8Ch | 0000 -000 | 0000 -000 | uuuu -uuu |
| PCON | 8Eh | ---- 1-0x | ---- 1-uq ^(1,6) | ---- u-uu |
| TXSTA | 98h | 0000 -010 | 0000 -010 | uuuu -uuu |
| EECON1 | 9Ch | ---- x000 | ---- q000 | ---- uuuu |
| VRCON | 9Fh | 000- 0000 | 000- 0000 | uuu- uuuu |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечания:

1. Если напряжение V_{DD} будет слишком низкое, то возникнет сброс по включению питания POR и регистры будут инициализированы иначе.
2. Один или несколько битов INTCON, PIR1 будут изменены при выходе из режима SLEEP.
3. Если бит GIE=1 при выходе из режима SLEEP, в счетчик команд будет загружен вектор прерываний (0004h).
4. Смотрите в таблице 14-6 состояние битов регистра STATUS.
5. Если выход из режима SLEEP возник по изменению уровня на выходе компаратора, бит 6 = 1, иначе бит 6 = u.
6. Если произошел сброс по снижению напряжения питания, то бит 0 = 0, иначе бит 0 = u.

Рис. 14-9 Временная диаграмма задержки PWRT (-MCLR не подключен к V_{DD}), случай 1

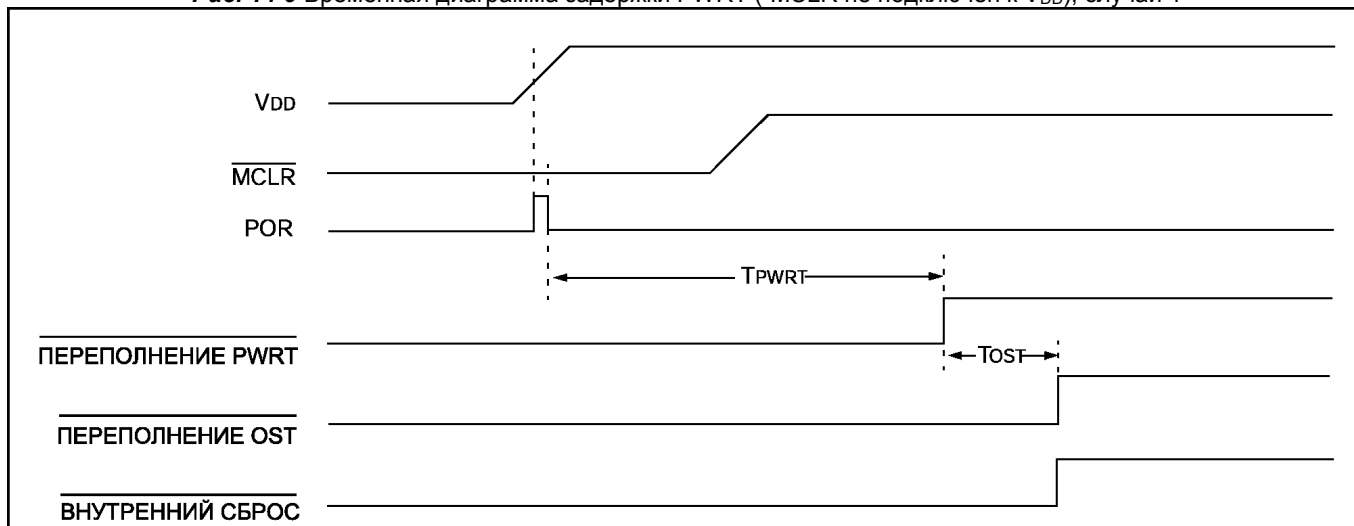


Рис. 14-10 Временная диаграмма задержки PWRT (-MCLR не подключен к V_{DD}), случай 2

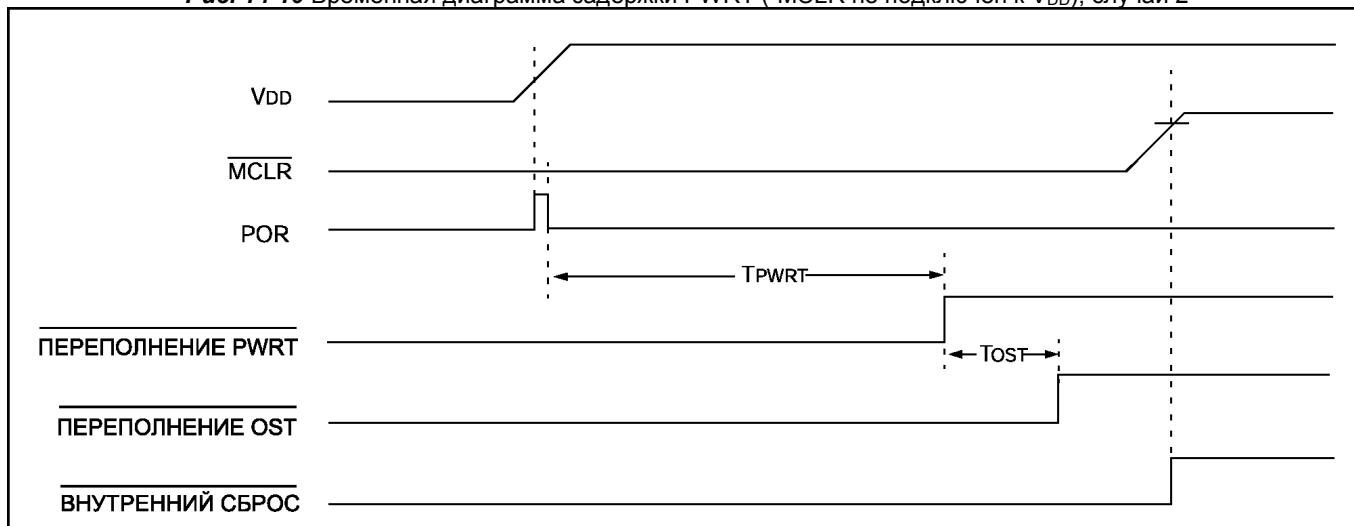


Рис. 14-11 Временная диаграмма задержки PWRT (-MCLR подключен к V_{DD})

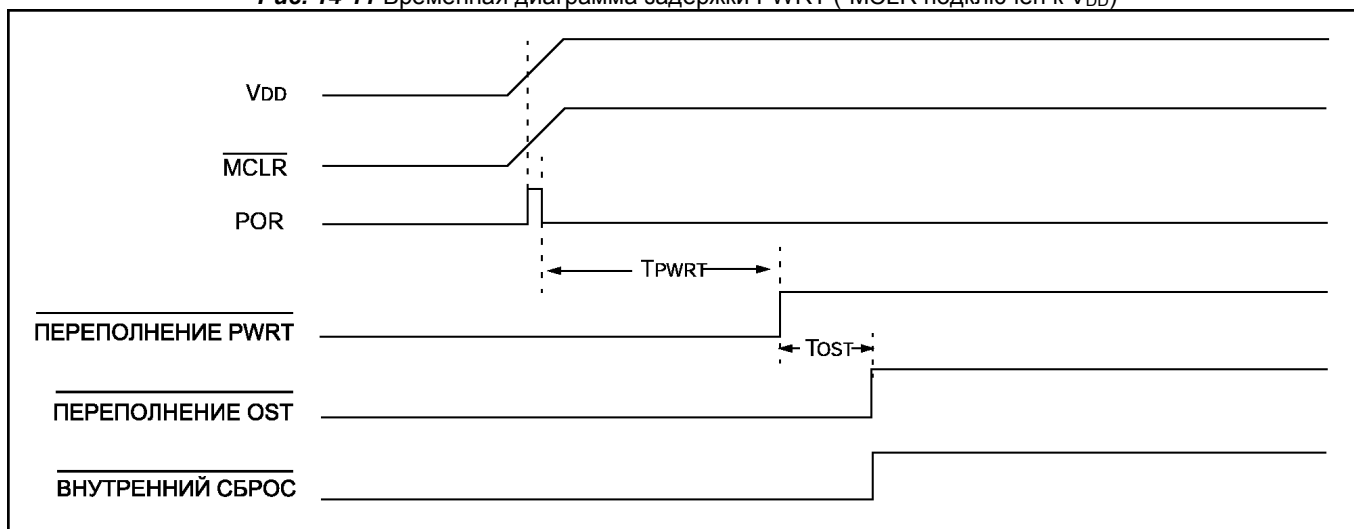
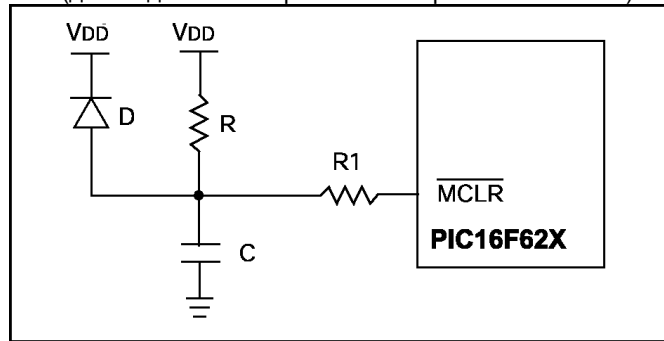


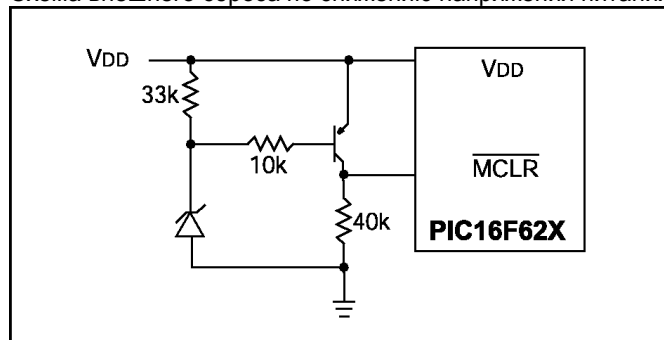
Рис. 14-12 Схема внешнего сброса по включению питания (для медленного нарастания напряжения питания)



Примечания:

1. Схема внешнего сброса требуется только, если очень большое время нарастания напряжения питания. Диод D предназначен для быстрой разрядки конденсатора при снижении напряжения питания.
2. Сопротивление резистора R рекомендуется выбирать меньше 40кОм.
3. Резистор R1 от 100Ом до 1кОм ограничивает ток вывода MCLR в случае его разрушения от электростатического разряда (ESD) или перенапряжения (EOS).

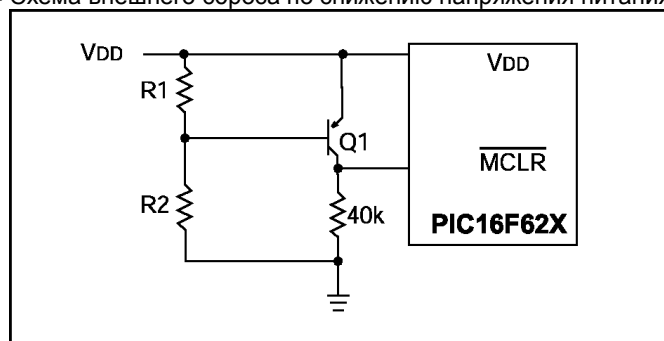
Рис. 14-13 Схема внешнего сброса по снижению напряжения питания (1 вариант)



Примечания:

1. Эта схема будет сбрасывать микроконтроллер, когда V_{DD} будет ниже $V_Z + 0.7V$, где V_Z —напряжение стабилизации стабилитрона.
2. Внутренняя схема сброса по снижению напряжения питания должна быть выключена.

Рис. 14-14 Схема внешнего сброса по снижению напряжения питания (2 вариант)



Примечания:

1. Недорогая схема сброса, но менее точная по сравнению с 1 вариантом. Транзистор Q1 закрывается, когда напряжение питания ниже определенного порога.

$$V_{dd} \cdot \frac{R1}{R1 + R2} = 0.7$$

2. Внутренняя схема сброса по снижению напряжения питания должна быть выключена.
3. Номиналы резисторов должны быть выбраны с учетом типа транзистора.

14.6 Прерывания

PIC16F62X имеют 10 источников прерываний:

- внешнее прерывание с входа RB0/INT;
- прерывание по переполнению TMR0;
- прерывание по изменению уровня сигнала на входах RB7:RB4;
- прерывание от модуля компараторов;
- прерывания от USART;
- прерывания от CCP;
- прерывание по переполнению TMR1;
- прерывание от TMR2.

Регистр INTCON содержит флаги отдельных прерываний, биты разрешения этих прерываний и бит глобального разрешения прерываний.

Если бит GIE (INTCON<7>) установлен в '1', разрешены все немаскированные прерывания. Если GIE=0, то все прерывания запрещены. Каждое прерывание в отдельности может быть разрешено/запрещено установкой/сбросом соответствующего бита в регистрах INTCON и PIE1. При сбросе микроконтроллера бит GIE сбрасывается в '0'.

При возвращении из подпрограммы обработки прерывания, по команде RETFIE, бит GIE аппаратно устанавливается в '1' разрешая все немаскированные прерывания.

В регистре INTCON находятся флаги следующих прерываний: внешнего сигнала INT, изменения уровня сигнала на входах RB7:RB4, переполнения TMR0.

В регистре PIR1 содержатся флаги прерываний периферийных модулей микроконтроллера, а в регистре PIE1 соответствующие биты разрешения прерываний.

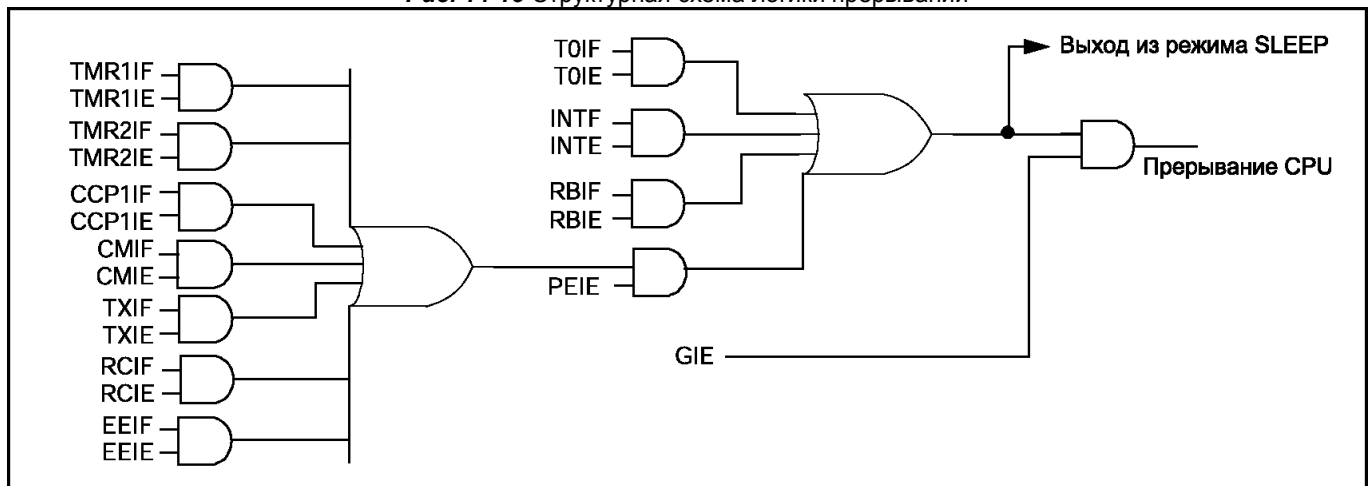
При переходе на подпрограмму обработки прерываний бит GIE аппаратно сбрасывается в '0' запрещая прерывания, адрес возврата из подпрограммы обработки прерываний помещается в стек, а в счетчик команд PC загружается вектор прерывания 0004h. Источник прерываний может быть определен проверкой флагов прерываний, которые должны быть сброшены программно перед разрешением прерываний, чтобы избежать повторного вызова.

Для внешних источников прерываний (сигнал INT, изменения уровня сигнала на входах RB7:RB4) время перехода на подпрограмму обработки прерываний будет составлять 3-4 машинных цикла. Точное время перехода зависит от конкретного случая (см. рисунок 14-16), оно одинаково для 1 и 2-х цикловых команд. Флаги прерываний устанавливаются независимо от состояния соответствующих битов маски и бита GIE.

Примечания:

1. Индивидуальные флаги прерываний устанавливаются независимо от состояния соответствующих битов маски и бита GIE.
2. Если выполнена команда, сбрасывающая бит GIE в '0', то любое прерывание, ожидаемое выполнения в следующем машинном цикле, игнорируется. Микроконтроллер выполнит пустой цикл NOP следом за командой сброса бита GIE. Прерывание будет ожидать установку бита GIE в '1'.

Рис. 14-15 Структурная схема логики прерываний



14.6.1 Внешнее прерывание с входа RB0/INT

Внешнее прерывание с входа RB0/INT происходит: по переднему фронту сигнала, если бит INTEDG (OPTION<6>) установлен в '1'; по заднему фронту сигнала, если бит INTEDG сброшен в '0'. Когда активный фронт сигнала появляется на входе RB0/INT, бит INTF (INTCON<1>) устанавливается в '1'. Прерывание может быть запрещено сбросом бита INTE (INTCON<4>) в '0'. Флаг прерывания INTF должен быть сброшен программно в подпрограмме обработки прерываний. Прерывание INT может вывести микроконтроллер из режима SLEEP, если бит INTE=1 до перехода в режим SLEEP. Состояние бита GIE определяет: переходить ли на подпрограмму обработки прерываний после выхода из режима SLEEP. На рисунке 14-18 показана временная диаграмма выхода микроконтроллера из режима SLEEP по прерыванию с входа RB0/INT.

14.6.2 Прерывание по переполнению TMR0

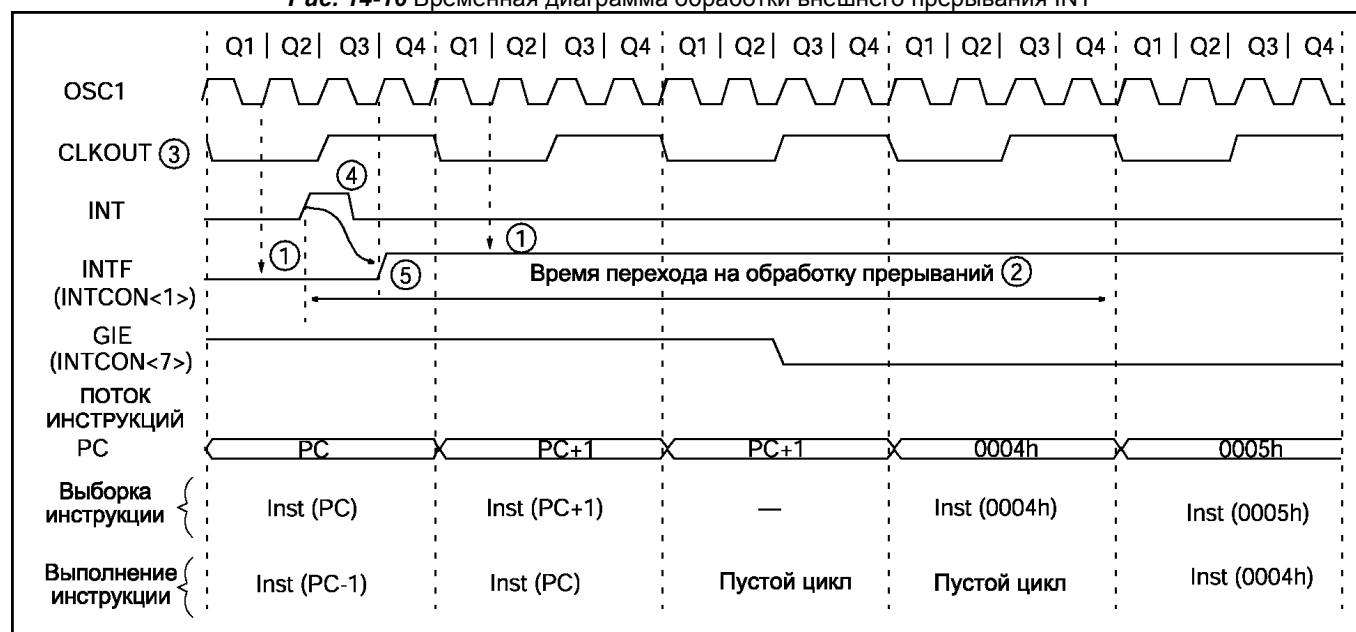
Переполнение таймера TMR0 (FFh → 00h) устанавливает флаг T0IF (INTCON<2>) в '1'. Прерывание от TMR0 можно разрешить/запретить установкой/сбросом бита T0IE (INTCON<5>). Описание работы модуля TMR0 смотрите в разделе 6.0.

14.6.3 Прерывание по изменению уровня сигнала на входах RB7:RB4

Изменение уровня сигнала на входах RB7:RB4 вызывает установку флага RBIF (INTCON<0>). Прерывание можно разрешить/запретить установкой/сбросом бита RBIE (INTCON<4>). Описание работы PORTB смотрите в разделе 5.2.

Примечание. Если изменение сигнала на входе RB7:RB4 происходит на начале цикла Q2 чтения PORTB, флаг RBIF в '1' может не устанавливается.

Рис. 14-16 Временная диаграмма обработки внешнего прерывания INT



Примечания:

1. Флаг INTF проверяется в такте Q1.
2. Время переход на обработку прерываний: не синхронизированный сигнал $3-4T_{CY}$; синхронизированный сигнал $3T_{CY}$. Время перехода не зависит от выполняемой инструкции (одно или двух цикловая команда).
3. CLKOUT доступен только в ER и INTRC режиме генератора.
4. Минимальную длительность импульса INT смотрите в разделе электрические характеристики.
5. Флаг INTF может быть установлен в любой момент.

Таблица 14-8 Регистры и биты связанные с обработкой прерываний

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|-------|--------|-------|-------|-------|-------|-------|-------|--------|--------|-----------|---------------|
| 0Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | - | CCP1F | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | - | CCP1E | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечания

1. Затененные биты не используются при обработке прерываний.
2. Другие сбросы: -MCLR, BOR, переполнение WDT в нормальном режиме работы.

14.7 Сохранение контекста при обработке прерываний

При переходе на подпрограмму обработки прерываний в стеке сохраняется только адрес возврата. Как правило, необходимо сохранять значения ключевых регистров при обработке прерываний (например, регистр W и STATUS), что выполняется программным способом. В примере 14-1 показано сохранение и восстановление регистров STATUS и W. Регистр W_TEMP должен располагаться в области памяти, доступной во всех банках (70h-7Fh). Регистр STATUS_TEMP размещается в банке 0. Последовательность операций в примере 14-1:

- сохранить регистр W;
- сохранить регистр STATUS в банке памяти 0;
- выполнить код обработки прерываний;
- восстановить регистр STATUS (с восстановлением текущего банка);
- восстановить регистр W.

Пример 14-1 Сохранение и восстановление регистров STATUS и W

```

MOVWF    W_TEMP                ;Сохранить W в регистре
                                ;текущего банка
SWAPF    STATUS,W              ;Обменять местами полубайты и сохранить в W
CLRF     STATUS                 ;Выбрать банк 0
MOVWF    STATUS_TEMP           ;Сохранить регистр STATUS
:
:                               ; Код программы обработки прерываний
:
SWAPF    STATUS_TEMP,W          ;Прочитать регистр STATUS_TEMP
                                ;в W, восстанавливая банк памяти программ
MOVWF    STATUS                 ;Переписать W в регистр STATUS
SWAPF    W_TEMP,F              ; Обменять местами полубайты в W_TEMP
SWAPF    W_TEMP,W              ; Обменять местами полубайты в W_TEMP и записать в W

```

В конкретных приложениях может потребоваться сохранять и другие специальные регистры (например, FSR, PCLATH).

14.8 Сторожевой таймер WDT

Встроенный сторожевой таймер WDT работает от отдельного RC генератора, не требующего внешних компонентов. Это позволяет работать сторожевому таймеру WDT при выключенном тактовом генераторе (выводы OSC1, OSC2) в SLEEP режиме микроконтроллера. В нормальном режиме работы при переполнении WDT происходит сброс микроконтроллера. Если микроконтроллер находится в SLEEP режиме, переполнение WDT выводит его из режима SLEEP с продолжением нормальной работы. WDT выключен, если WDTE = 0 в слове конфигурации.

14.8.1 Период WDT

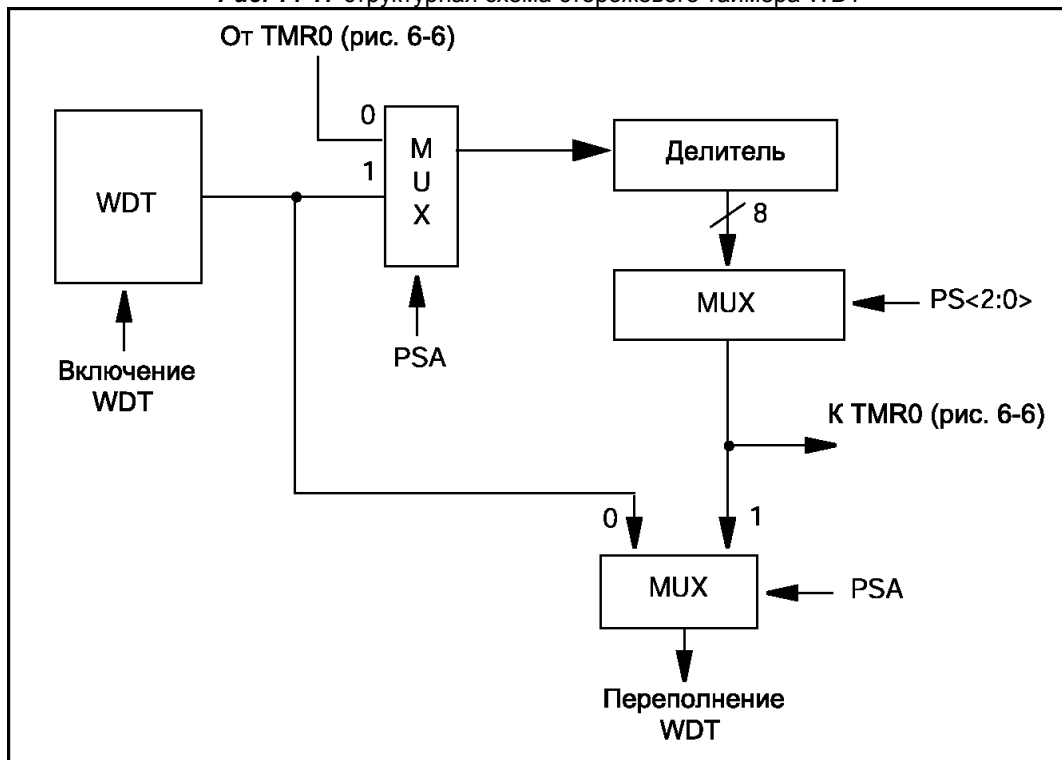
WDT имеет номинальное время переполнения 18мс (без предделителя). Время переполнения зависит от температуры, напряжения питания V_{DD} и разброса технологических параметров микроконтроллера (см. раздел "электрические характеристики"). Если требуется большее время переполнения WDT, необходимо программно подключить предделитель в регистре OPTION с максимальным коэффициентом деления 1:128. С включенным предделителем время переполнения может достигать 2.3с.

Команды CLRWDТ и SLEEP сбрасывают сторожевой таймер и предделитель, если он подключен к WDT, откладывая сброс устройства. В регистре STATUS бит -TO=0, если произошел сброс по переполнению WDT.

14.8.2 Рекомендации по работе с WDT

Даже в самых плохих условиях работы требуется несколько секунд для переполнения WDT (минимальное напряжение питания V_{DD} , максимальная температура, максимальный коэффициент предделителя подключенного к WDT).

Рис. 14-17 структурная схема сторожевого таймера WDT



Примечание. Биты PSA, PS2:PS0 находятся в регистре OPTION.

Таблица 14-9 Регистры и биты связанные с работой WDT

| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Сброс POR | Другие сбросы |
|-------|-------------|-------|--------|-------|-------|--------|-------|-------|-------|-----------|---------------|
| 2007h | Слово конф. | LVP | BOREN | MCLR | FOSC2 | -PWRTE | WDTE | FOSC1 | FOSC0 | uuuu uuuu | uuuu uuuu |
| 03h | STATUS | IRP | RP1 | RP0 | -TO | -PD | Z | DC | C | 0001 1xxx | 000q q000 |
| 81h | OPTION | -RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

Обозначения: - - не используется, читается как 0; u – не изменяется; x – не известно; q – зависит от условий.

Примечание. Затененные биты не влияют на работу WDT.

14.9 Режим энергосбережения SLEEP

Переход в режим энергосбережения происходит по команде SLEEP. При переходе в режим SLEEP сторожевой таймер WDT сбрасывается, но продолжает работать. В регистре STATUS бит -TO сбрасывается в '0', бит -PD устанавливается в '1', тактовый генератор микроконтроллера выключен. Порты ввода/вывода остаются в том же состоянии, что и до выполнения команды SLEEP (высокий уровень, низкий уровень, третье состояние). Для снижения энергопотребления в SLEEP режиме все каналы ввода/вывода должны быть подключены к V_{DD} или V_{SS} при отсутствии токов из внешней схемы через выходы портов, выходы модуля компараторов и источника опорного напряжения выключены. Выводы находящиеся в третьем состоянии должны иметь высокий или низкий уровень сигнала, чтобы избежать токов переключения входных буферов. Вход T0CKI должен быть подключен к V_{DD} или V_{SS} для снижения энергопотребления. Должны учитываться внутренние подтягивающие резисторы, включенные на входах PORTB. На входе -MCLR должен быть высокий уровень сигнала.

Примечание. Сброс сгенерированный при переполнении WDT не переводит сигнал на входе -MCLR в низкий логический уровень.

14.9.1 Выход из режима SLEEP

Микроконтроллер выйдет из режима SLEEP по одному из следующих событий:

1. внешний сброс по сигналу на входе -MCLR;
2. переполнение сторожевого таймера WDT (если он разрешен);
3. периферийное прерывание (по сигналу INT, изменение уровня сигнала на входах RB7:RB4 и др.).

Внешний сброс по сигналу -MCLR вызывает сброс микроконтроллера. Два других события вызывают продолжение выполнения программы.

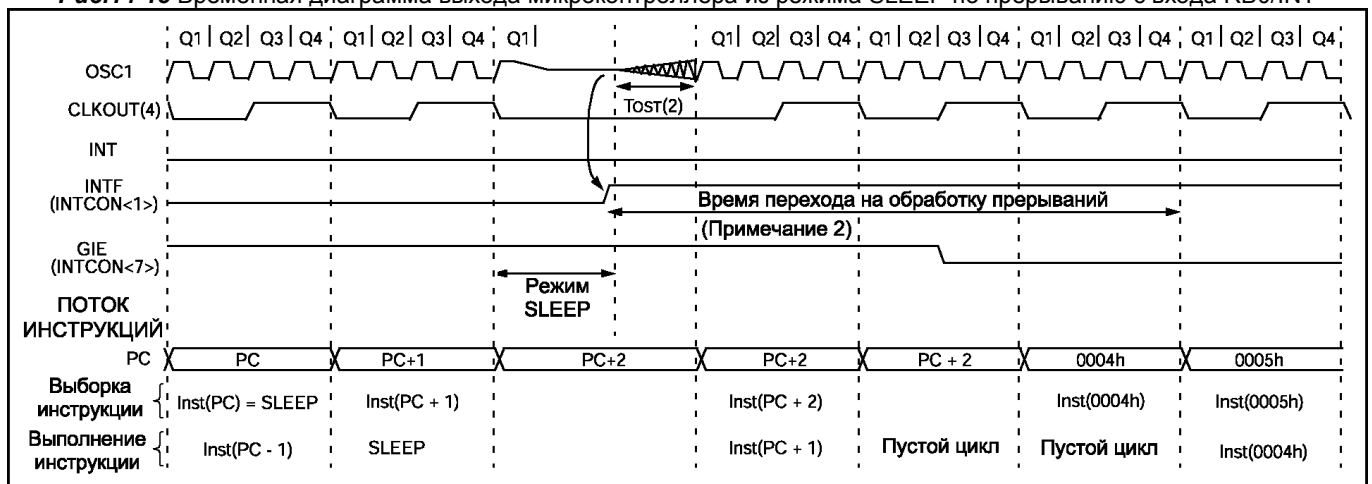
Биты -TO и -PD в регистре STATUS могут использоваться для определения причины сброса микроконтроллера. Бит -PD сбрасывается в '0' при переходе в режим SLEEP. Бит -TO сбрасывается в '0', если произошло переполнение WDT.

При выполнении команды SLEEP происходит предвыборка следующей (PC+1). Если прерывание должно вывести микроконтроллер из режима SLEEP, соответствующий бит разрешения прерывания устанавливается в '1'. Микроконтроллер выходит из режима SLEEP независимо от состояния бита GIE. Если GIE=0, выполняется следующая инструкция после SLEEP без перехода по вектору прерываний. Если GIE=1, исполняется следующая инструкция после SLEEP и происходит переход на подпрограмму обработки прерываний (адрес 0004h). Когда выполнение какой-либо команды при выходе из режима SLEEP нежелательно, необходимо поле команды SLEEP использовать инструкцию NOP.

Примечание. Если выполнена команда, сбрасывающая бит GIE в '0', то любое прерывание ожидаемое выполнения в следующем машинном цикле игнорируется. Микроконтроллер выполнит пустой цикл NOP следом за командой сброса бита GIE. Прерывание будет ожидать установку бита GIE в '1'.

При выходе из режима SLEEP сторожевой таймер WDT сбрасывается, независимо от источника "пробуждения".

Рис. 14-18 Временная диаграмма выхода микроконтроллера из режима SLEEP по прерыванию с входа RB0/INT



Примечания:

1. Режим генератора XT, HS или LP.
2. T_{ost} = 1024 T_{osc} (не масштабный рисунок). Приблизительное время задержки для ER режима генератора 1 мс.
3. Предполагается, что GIE=1. После выхода из режима SLEEP произойдет переход по вектору прерывания.
4. CLKOUT не доступен для этих режимов генератора, но показан для пояснения диаграммы.

14.10 Защита кода программы

Если защита кода программы (EEPROM памяти данных) не была включена, то память программ (EEPROM память данных) может быть прочитана для проверки программирования.

Примечание. При выключении защиты память программ и EEPROM память данных будет полностью стерта, кроме калибровочной информации внутреннего генератора INTRC.

14.11 Размещение идентификатора ID

Четыре ячейки памяти программ (2000h-2003h) предназначены для размещения идентификатора, которые могут использоваться для сохранения контрольной суммы или другой информации. Эти ячейки недоступны программе микроконтроллера, но могут быть прочитаны и изменены при программировании. Используются только 4 младших бита каждой ячейки.

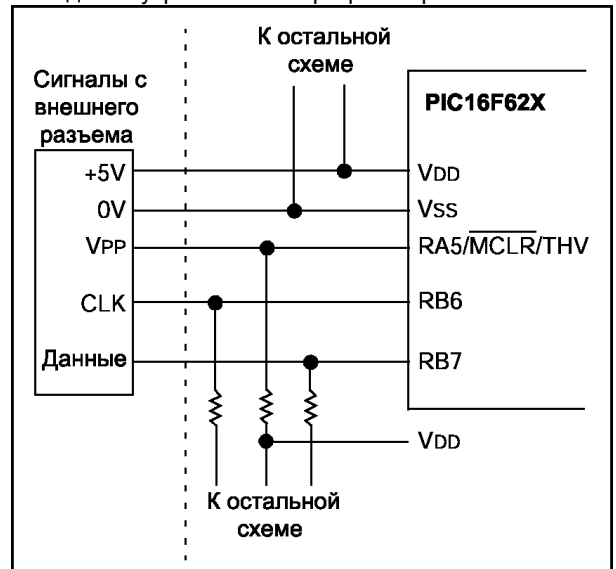
14.12 Внутрисхемное программирование ICSP

Микроконтроллеры PIC16F62X могут быть запрограммированы по последовательному интерфейсу в готовом изделии. Программирование выполняется по двум линиям последовательно интерфейса (данные, синхронизация) и трем дополнительным линиям: напряжение питания, общий провод, напряжение программирования. Это позволяет изготавливать платы с не запрограммированными микроконтроллерами, а затем загружать в них программу перед поставкой изделия. Данная функция также позволяет обновлять программное обеспечение микроконтроллеров.

Микроконтроллер переходит в режим программирования/проверки при удержании на выводах RB6, RB7 низкого уровня во время перехода сигнала на входе -MCLR от V_{IL} к V_{IH} (см. спецификацию программирования микроконтроллера). После этого RB6 становится тактовым входом, RB7 входом данных. Оба вывода имеют входные триггеры Шмидта.

После перехода в режим программирования/проверки (счетчик команд PC сбрасывается в 000h) можно передать 6-разрядную команду. В зависимости от типа команды можно записать/прочитать 14-разрядные данные из микроконтроллера.

Рис. 14-9 Типовая схема включения микроконтроллера для внутрисхемного программирования



Дополнительную информацию смотрите в спецификации программирования. На рисунке 14-19 показана типовая схема включения микроконтроллера для внутрисхемного программирования.

14.13 Режим низковольтного программирования

Бит LVP в слове конфигурации используется для разрешения режима низковольтного программирования. Этот режим позволяет запрограммировать микроконтроллер по интерфейсу ICSP при одном источнике питания +5В (не требуется подавать напряжение V_{IH} на вывод -MCLR). По умолчанию LVP=1, разрешая низковольтное программирование. При этом вывод RB4/PGM используется для низковольтного программирования и перестает быть цифровым портом ввода/вывода. Микроконтроллер переходит в режим программирования, когда на выводе RB4/PGM высокий уровень сигнала. Режим стандартного программирования по-прежнему доступен (когда на выводе -MCLR напряжение V_{IH}).

Примечания:

1. В режиме низковольтного программирования вывод RB4/PGM не может использоваться как цифровой порт ввода/вывода.
2. Напряжение питания V_{DD} должно быть $+5B \pm 10\%$ в течение всего процесса программирования/проверки.

Если режим низковольтного программирования не используется, бит LVP должен быть сброшен в '0', вывод RB4/PGM становится цифровым портом ввода/вывода. Бит LVP может быть изменен только в стандартном режиме программирования (когда на выводе -MCLR напряжение V_{IH}). Когда бит LVP=0, возможен только стандартный режим программирования/проверки микроконтроллера.

15.0 Система команд

Каждая команда микроконтроллеров PIC16F62X состоит из одного 14-разрядного слова, разделенного на код операции (OPCODE), определяющий тип команды и один или несколько операндов, определяющие операцию команды. Полный список команд смотрите в таблице 15-2. Команды разделены на следующие группы: байт ориентированные команды, бит ориентированные команды, команды управления и операций с константами. Описание полей кода операции смотрите в таблице 15-1.

Для байт ориентированных команд 'f' является указателем регистра, а 'd' указателем адресата результата. Указатель регистра определяет, какой регистр должен использоваться в команде. Указатель адресата определяет, где будет сохранен результат. Если 'd'=0, результат сохраняется в регистре W. Если 'd'=1, результат сохраняется в регистре, который используется в команде.

В бит ориентированных командах 'b' определяет номер бита участвующего в операции, а 'f' - указатель регистра, который содержит этот бит.

В командах управления или операциях с константами 'k' представляет восемь или одиннадцать бит константы или значения литералов.

Система команд аккумуляторного типа, ортогональна и разделена на три основных группы:

- байт ориентированные команды;
- бит ориентированные команды;
- команды управления и операций с константами.

Все команды выполняются за один машинный цикл, кроме команд условия, в которых получен ложный результат и инструкций изменяющих значение счетчика команд PC. В случае выполнения команды за два машинных цикла, во втором цикле выполняется инструкция NOP. Один машинный цикл состоит из четырех тактов генератора. Для тактового генератора с частотой 4 МГц все команды выполняются за 1мкс, если условие истинно или изменяется счетчик команд PC, команда выполняется за 2мкс.

Мнемоника команд, поддерживаемая ассемблером MPASM, показана в таблице 15-2. На рисунке 15-1 показан форма команд трех основных групп.

Примечание. Для совместимости программного обеспечения со следующими версиями микроконтроллеров PICmicro не используйте команды TRIS и OPTION.

Во всех примерах используется следующий формат шестнадцатеричных чисел:

0xhh, где h - шестнадцатеричная цифра.

Таблица 15-1 Описание полей кода операции

| Поле | Описание |
|---------------|---|
| f | Адрес регистра (от 0x00 до 0x7F) |
| w | Рабочий регистр (аккумулятор) |
| b | Номер бита в 8-разрядном регистре |
| k | Константа (данные или метка) |
| x | Не имеет значения (0 или 1). Ассемблер генерирует x=0 для совместимости программы микроконтроллера с инструментальными средствами |
| d | Указатель адресата результата операции: d = 0 - результат сохраняется в регистре w d = 1 - результат сохраняется в регистре f По умолчанию d = 1 |
| label | Имя метки |
| TOS | Вершина стека |
| PC | Счетчик команд |
| PCLATH | Буфер старшего байта счетчика команд |
| GIE | Бит глобального разрешения прерываний |
| WDT | Сторожевой таймер |
| -TO | Флаг переполнения WDT |
| -PD | Флаг сброса по включению питания |
| dest | Приемник, регистр w или регистр памяти |
| [] | Дополнительные параметры |
| () | Содержимое |
| → | Присвоение |
| < > | Битовое поле |
| € | Из набора |
| <i>Курсив</i> | Термин, определяемый пользователем |

Рис 15-1 Форма команд трех основных групп



Таблица 15-2 Список команд микроконтроллеров PIC16F62X

| Мнемоника команды | Описание | Циклов | 14-разрядный код | | Изм. флаги | Прим. |
|--|---|--------|------------------|-----------|---------------|-------|
| | | | Бит 13 | Бит 0 | | |
| Байт ориентированные команды | | | | | | |
| ADDWF f,d | Сложение W и f | 1 | 00 0111 | dfff ffff | C,DC,Z | 1,2 |
| ANDWF f,d | Побитное 'И' W и f | 1 | 00 0101 | dfff ffff | Z | 1,2 |
| CLRF f | Очистить f | 1 | 00 0001 | 1fff ffff | Z | 2 |
| CLRWF - | Очистить W | 1 | 00 0001 | 0000 0011 | Z | |
| COMF f,d | Инвертировать f | 1 | 00 1001 | dfff ffff | Z | 1,2 |
| DECf f,d | Вычесть 1 из f | 1 | 00 0011 | dfff ffff | Z | 1,2 |
| DECFSZ f,d | Вычесть 1 из f и пропустить если 0 | 1(2) | 00 1011 | dfff ffff | | 1,2,3 |
| INCf f,d | Прибавить 1 к f | 1 | 00 1010 | dfff ffff | Z | 1,2 |
| INCFSZ f,d | Прибавить 1 к f и пропустить если 0 | 1(2) | 00 1111 | dfff ffff | | 1,2,3 |
| IORWF f,d | Побитное 'ИЛИ' W и f | 1 | 00 0100 | dfff ffff | Z | 1,2 |
| MOVF f,d | Переслать f | 1 | 00 1000 | dfff ffff | Z | 1,2 |
| MOVWF f | Переслать W в f | 1 | 00 0000 | 1fff ffff | | |
| NOP - | Нет операции | 1 | 00 0000 | 0xx0 0000 | | |
| RLF f,d | Циклический сдвиг f влево через перенос | 1 | 00 1101 | dfff ffff | C | 1,2 |
| RRF f,d | Циклический сдвиг f вправо через перенос | 1 | 00 1100 | dfff ffff | C | 1,2 |
| SUBWF f,d | Вычесть W из f | 1 | 00 0010 | dfff ffff | C,DC,Z | 1,2 |
| SWAPF f,d | Поменять местами полубайты в регистре f | 1 | 00 1110 | dfff ffff | | 1,2 |
| XORWF f,d | Побитное 'исключающее ИЛИ' W и f | 1 | 00 0110 | dfff ffff | Z | 1,2 |
| Бит ориентированные команды | | | | | | |
| BCF f,b | Очистить бит b в регистре f | 1 | 01 00bb | bfff ffff | | 1,2 |
| BSF f,b | Установить бит b в регистре f | 1 | 01 01bb | bfff ffff | | 1,2 |
| BTFSC f,b | Проверить бит b в регистре f, пропустить если 0 | 1(2) | 01 10bb | bfff ffff | | 3 |
| BTFSS f,b | Проверить бит b в регистре f, пропустить если 1 | 1(2) | 01 11bb | bfff ffff | | 3 |
| Команды управления и операций с константами | | | | | | |
| ADDLW k | Сложить константу с W | 1 | 11 111x | kkkk kkkk | C,DC,Z | |
| ANDLW k | Побитное 'И' константы и W | 1 | 11 1001 | kkkk kkkk | Z | |
| CALL k | Вызов подпрограммы | 2 | 10 0kkk | kkkk kkkk | | |
| CLRWDT - | Очистить WDT | 1 | 00 0000 | 0110 0100 | -TO,-PD | |
| GOTO k | Безусловный переход | 2 | 10 1kkk | kkkk kkkk | | |
| IORLW k | Побитное 'ИЛИ' константы и W | 1 | 11 1000 | kkkk kkkk | Z | |
| MOVLW k | Переслать константу в W | 1 | 11 00xx | kkkk kkkk | | |
| RETFIE - | Возврат из подпрограммы с разрешением прерываний | 2 | 00 0000 | 0000 1001 | | |
| RETLW k | Возврат из подпрограммы с загрузкой константы в W | 2 | 11 01xx | kkkk kkkk | | |
| RETURN - | Возврат из подпрограммы | 2 | 00 0000 | 0000 1000 | | |
| SLEEP - | Перейти в режим SLEEP | 1 | 00 0000 | 0110 0011 | -TO,-PD | |
| SUBLW k | Вычесть W из константы | 1 | 11 110x | kkkk kkkk | C,DC,Z | |
| XORLW k | Побитное 'исключающее ИЛИ' константы и W | 1 | 11 1010 | kkkk kkkk | Z | |

Примечания:

1. При выполнении операции "чтение - модификация - запись" с портом ввода/вывода исходные значения считываются с выводов порта, а не из выходных защелок. Например, если в выходной защелке было записано '1', а на соответствующем выходе низкий уровень сигнала, то обратно будет записано значение '0'.
2. При выполнении записи в TMR0 (и d=1) предделитель TMR0 сбрасывается, если он подключен к модулю TMR0.
3. Если условие истинно или изменяется значение счетчика команд PC, то инструкция выполняется за два цикла. Во втором цикле выполняется команда NOP.

15.1 Подробное описание команд

| ADDLW | Сложить константу с W | | |
|---------------|---|-------|------|
| Синтаксис: | <i>[label]</i> | ADDLW | k |
| Операнды: | $0 \leq k \leq 255$ | | |
| Операция: | $(W) + k \rightarrow (W)$ | | |
| Измен. флаги: | C, DC, Z | | |
| Код: | 11 | 111x | kkkk |
| Описание: | Содержимое регистра W складывается с 8-разрядной константой 'k'. Результат сохраняется в регистре W. | | |
| Слов: | 1 | | |
| Циклов: | 1 | | |
| Пример: | ADDLW | 0x15 | |
| | До выполнения команды | | |
| | W = 0x10 | | |
| | После выполнения команды | | |
| | W = 0x25 | | |
| ADDWF | Сложение W и f | | |
| Синтаксис: | <i>[label]</i> | ADDWF | f,d |
| Операнды: | $0 \leq f \leq 127$ $d \in [0,1]$ | | |
| Операция: | $(W) + (f) \rightarrow (dest)$ | | |
| Измен. флаги: | C, DC, Z | | |
| Код: | 00 | 0111 | dfff |
| Описание: | Сложить содержимое регистров W и 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'. | | |
| Слов: | 1 | | |
| Циклов: | 1 | | |
| Пример: | ADDWF | FSR,0 | |
| | До выполнения команды | | |
| | W = 0x17 | | |
| | FSR = 0xC2 | | |
| | После выполнения команды | | |
| | W = 0xD9 | | |
| | FSR = 0xC2 | | |
| ANDLW | Побитное 'И' константы и W | | |
| Синтаксис: | <i>[label]</i> | ANDLW | k |
| Операнды: | $0 \leq k \leq 255$ | | |
| Операция: | $(W) .AND. k \rightarrow (W)$ | | |
| Измен. флаги: | Z | | |
| Код: | 11 | 1001 | kkkk |
| Описание: | Выполняется побитное 'И' содержимого регистра W и 8-разрядной константы 'k'. Результат сохраняется в регистре W. | | |
| Слов: | 1 | | |
| Циклов: | 1 | | |
| Пример: | ANDLW | 0x5F | |
| | До выполнения команды | | |
| | W = 0xA3 | | |
| | После выполнения команды | | |
| | W = 0x03 | | |

| ANDWF | Побитное 'И' W и f | | | |
|---------------|--|------|------|------|
| Синтаксис: | [<i>label</i>] ANDWF f,d | | | |
| Операнды: | 0 ≤ f ≤ 127 d ∈ [0,1] | | | |
| Операция: | (W) .AND. (f) → (dest) | | | |
| Измен. флаги: | Z | | | |
| Код: | 00 | 0101 | dfff | ffff |
| Описание: | Выполняется побитное 'И' содержимого регистров W и 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'. | | | |
| Слов: | 1 | | | |
| Циклов: | 1 | | | |
| Пример: | ANDWF FSR,1 До выполнения команды W = 0x17 FSR = 0xC2 После выполнения команды W = 0x17 FSR = 0x02 | | | |
| BCF | Очистить бит b в регистре f | | | |
| Синтаксис: | [<i>label</i>] BCF f,b | | | |
| Операнды: | 0 ≤ f ≤ 127 0 ≤ b ≤ 7 | | | |
| Операция: | 0 → (f) | | | |
| Измен. флаги: | Нет | | | |
| Код: | 01 | 00bb | bfff | ffff |
| Описание: | Очистить бит 'b' в регистре 'f'. | | | |
| Слов: | 1 | | | |
| Циклов: | 1 | | | |
| Пример: | BCF FLAG_REG,7 До выполнения команды FLAG_REG = 0xC7 После выполнения команды FLAG_REG = 0x47 | | | |
| BSF | Установить бит b в регистре f | | | |
| Синтаксис: | [<i>label</i>] BSF f,b | | | |
| Операнды: | 0 ≤ f ≤ 127 0 ≤ b ≤ 7 | | | |
| Операция: | 1 → (f) | | | |
| Измен. флаги: | Нет | | | |
| Код: | 01 | 01bb | bfff | ffff |
| Описание: | Установить бит 'b' в регистре 'f'. | | | |
| Слов: | 1 | | | |
| Циклов: | 1 | | | |
| Пример: | BSF FLAG_REG,7 До выполнения команды FLAG_REG = 0x0A После выполнения команды FLAG_REG = 0x8A | | | |

BTFS Проверить бит **b** в регистре **f**, пропустить если **0**Синтаксис: `[label] BTFS f,b`Операнды: $0 \leq f \leq 127$
 $0 \leq b \leq 7$ Операция: пропустить если $(f < b) = 0$

Измен. флаги: Нет

Код:

| | | | |
|----|------|------|------|
| 01 | 10bb | bfff | ffff |
|----|------|------|------|

Описание: Если бит 'b' в регистре 'f' равен '1', то выполняется следующая инструкция.
Если бит 'b' в регистре 'f' равен '0', то следующая инструкция не выполняется, команда выполняется за два цикла. Во втором цикле выполняется NOP.

Слов: 1

Циклов: 1(2)

Пример: `HERE BTFS FLAG,1`
`FALSE GOTO PROCESS_CODE`
`TRUE` •
•

До выполнения команды

PC = адрес HERE

После выполнения команды

Если $FLAG < 1 > = 0$,

PC = адрес TRUE

Если $FLAG < 1 > = 1$,

PC = адрес FALSE

BTFS Проверить бит **b** в регистре **f**, пропустить если **1**Синтаксис: `[label] BTFS f,b`Операнды: $0 \leq f \leq 127$
 $0 \leq b \leq 7$ Операция: пропустить если $(f < b) = 1$

Измен. флаги: Нет

Код:

| | | | |
|----|------|------|------|
| 01 | 11bb | bfff | ffff |
|----|------|------|------|

Описание: Если бит 'b' в регистре 'f' равен '0', то выполняется следующая инструкция.
Если бит 'b' в регистре 'f' равен '1', то следующая инструкция не выполняется, команда выполняется за два цикла. Во втором цикле выполняется NOP.

Слов: 1

Циклов: 1(2)

Пример: `HERE BTFS FLAG,1`
`FALSE GOTO PROCESS_CODE`
`TRUE` •
•

До выполнения команды

PC = адрес HERE

После выполнения команды

Если $FLAG < 1 > = 0$,

PC = адрес FALSE

Если $FLAG < 1 > = 1$,

PC = адрес TRUE

| CALL | Вызов подпрограммы | | | | |
|---------------|--|------|------|------|------|
| Синтаксис: | <i>[label]</i> CALL k | | | | |
| Операнды: | $0 \leq k \leq 2047$ | | | | |
| Операция: | (PC) + 1 → TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11> | | | | |
| Измен. флаги: | Нет | | | | |
| Код: | <table border="1"> <tr> <td>10</td> <td>0kkk</td> <td>kkkk</td> <td>Kkkk</td> </tr> </table> | 10 | 0kkk | kkkk | Kkkk |
| 10 | 0kkk | kkkk | Kkkk | | |
| Описание: | Вызов подпрограммы. Адрес следующей инструкции (PC+1) помещается в вершину стека. Одиннадцать бит адреса загружаются из кода команды в счетчик команд PC<10:0>. Два старших бита загружаются в счетчик команд PC<12:11> из регистра PCLATH. Команда CALL выполняется за два цикла. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 2 | | | | |
| Пример: | HERE CALL THERE До выполнения команды PC = адрес HERE После выполнения команды PC = адрес THERE TOS = адрес HERE + 1 | | | | |

| CLRF | Очистить f | | | | |
|---------------|--|------|------|------|------|
| Синтаксис: | <i>[label]</i> CLRF f | | | | |
| Операнды: | $0 \leq f \leq 127$ | | | | |
| Операция: | 00h → (f) 1 → Z | | | | |
| Измен. флаги: | Z | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>0001</td> <td>1fff</td> <td>ffff</td> </tr> </table> | 00 | 0001 | 1fff | ffff |
| 00 | 0001 | 1fff | ffff | | |
| Описание: | Очистить содержимое регистра 'f' и установить флаг Z | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | CLRF FLAG_REG До выполнения команды FLAG_REG = 0x5A После выполнения команды FLAG_REG = 0x00 Z = 1 | | | | |

| CLRW | Очистить W | | | | |
|---------------|--|------|------|------|------|
| Синтаксис: | <i>[label]</i> CLRW | | | | |
| Операнды: | Нет | | | | |
| Операция: | 00h → (W) 1 → Z | | | | |
| Измен. флаги: | Z | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>0001</td> <td>0xxx</td> <td>xxxx</td> </tr> </table> | 00 | 0001 | 0xxx | xxxx |
| 00 | 0001 | 0xxx | xxxx | | |
| Описание: | Очистить содержимое регистра W и установить флаг Z | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | CLRW До выполнения команды W = 0x5A После выполнения команды W = 0x00 Z = 1 | | | | |
| CLRWDT | Очистить WDT | | | | |
| Синтаксис: | <i>[label]</i> CLRWDT | | | | |
| Операнды: | Нет | | | | |
| Операция: | 00h → WDT, 00h → предделитель WDT, 1 → -TO 1 → -PD | | | | |
| Измен. флаги: | -TO, -PD | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0100</td> </tr> </table> | 00 | 0000 | 0110 | 0100 |
| 00 | 0000 | 0110 | 0100 | | |
| Описание: | Инструкция CLRWDT сбрасывает WDT и предделитель, если он подключен к WDT. В регистре STATUS устанавливает биты -TO и -PD. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | CLRWDT До выполнения команды Счетчик WDT = ? После выполнения команды Счетчик WDT = 0 Предделитель WDT = 0 -TO = 1 - PD = 1 | | | | |
| COMF | Инвертировать f | | | | |
| Синтаксис: | <i>[label]</i> COMF f,d | | | | |
| Операнды: | 0 ≤ f ≤ 127 d ∈ [0,1] | | | | |
| Операция: | (-f) → (dest) | | | | |
| Измен. флаги: | Z | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>1101</td> <td>dfff</td> <td>ffff</td> </tr> </table> | 00 | 1101 | dfff | ffff |
| 00 | 1101 | dfff | ffff | | |
| Описание: | Инвертировать все биты в регистре 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | COMF REG1,0 До выполнения команды REG1 = 0x13 После выполнения команды REG1 = 0x13 W = 0xEC | | | | |

| DECF | Вычитать 1 из f | | | |
|---------------|---|------|------|------|
| Синтаксис: | [<i>label</i>] | DECF | f,d | |
| Операнды: | 0 ≤ f ≤ 127 d ∈ [0,1] | | | |
| Операция: | (f) - 1 → (dest) | | | |
| Измен. флаги: | Z | | | |
| Код: | 00 | 0011 | dfff | ffff |
| Описание: | Декрементировать содержимое регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'. | | | |
| Слов: | 1 | | | |
| Циклов: | 1 | | | |
| Пример: | DECF CNT,1 | | | |
| | До выполнения команды CNT = 0x01 Z = 0 | | | |
| | После выполнения команды CNT = 0x00 Z = 1 | | | |

| DECFSZ | Вычитать 1 из f и пропустить если 0 | | | |
|---------------|---|--------|------|------|
| Синтаксис: | [<i>label</i>] | DECFSZ | f,d | |
| Операнды: | 0 ≤ f ≤ 127 d ∈ [0,1] | | | |
| Операция: | (f) - 1 → (dest); пропустить если результат равен 0 | | | |
| Измен. флаги: | Нет | | | |
| Код: | 00 | 1011 | dfff | ffff |
| Описание: | Декрементировать содержимое регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'. Если результат не равен '0', то выполняется следующая инструкция. Если результат равен '0', то следующая инструкция не выполняется, команда выполняется за два цикла. Во втором цикле выполняется NOP. | | | |
| Слов: | 1 | | | |
| Циклов: | 1(2) | | | |
| Пример: | HERE DECFSZ CNT,1 | | | |
| | GOTO LOOP | | | |
| | CONTINUE • | | | |
| | • | | | |
| | До выполнения команды PC = адрес HERE | | | |
| | После выполнения команды CNT = CNT - 1 Если CNT = 0, PC = адрес CONTINUE Если CNT ≠ 0, PC = адрес HERE + 1 | | | |

GOTO **Безусловный переход**Синтаксис: *[label]* GOTO kОперанды: $0 \leq k \leq 2047$ Операция: $k \rightarrow PC<10:0>$,
(PCLATH<4:3>) $\rightarrow PC<12:11>$

Измен. флаги: Нет

Код:

| | | | |
|----|------|------|------|
| 10 | 1kkk | kkkk | kkkk |
|----|------|------|------|

Описание: Выполнить безусловный переход. Одиннадцать бит адреса загружаются из кода команды в счетчик команд PC<10:0>. Два старших бита загружаются в счетчик команд PC<12:11> из регистра PCLATH. Команда GOTO выполняется за два цикла.

Слов: 1

Циклов: 2

Пример: GOTO THERE
После выполнения команды
PC = адрес THERE**INCF** **Прибавить 1 к f**Синтаксис: *[label]* INCF f,dОперанды: $0 \leq f \leq 127$ $d \in [0,1]$ Операция: $(f) + 1 \rightarrow (dest)$

Измен. флаги: Z

Код:

| | | | |
|----|------|------|------|
| 00 | 1010 | dfff | ffff |
|----|------|------|------|

Описание: Инкрементировать содержимое регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.

Слов: 1

Циклов: 1

Пример: INCF CNT,1
До выполнения команды
CNT = 0xFF
Z = 0
После выполнения команды
CNT = 0x00
Z = 1

INCFSZ Прибавить 1 к f и пропустить если 0

| | | | | | |
|---------------|--|------|------|------|------|
| Синтаксис: | <i>[label]</i> INCFSZ f,d | | | | |
| Операнды: | $0 \leq f \leq 127$ $d \in [0,1]$ | | | | |
| Операция: | $(f) + 1 \rightarrow (dest)$; пропустить если результат равен 0 | | | | |
| Измен. флаги: | Нет | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>1111</td> <td>dfff</td> <td>ffff</td> </tr> </table> | 00 | 1111 | dfff | ffff |
| 00 | 1111 | dfff | ffff | | |
| Описание: | <p>Инкрементировать содержимое регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.</p> <p>Если результат не равен '0', то выполняется следующая инструкция. Если результат равен '0', то следующая инструкция не выполняется, команда выполняется за два цикла. Во втором цикле выполняется NOP.</p> | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1(2) | | | | |
| Пример: | <pre>HERE INCFSZ CNT,1 GOTO LOOP CONTINUE • • До выполнения команды PC = адрес HERE После выполнения команды CNT = CNT - 1 Если CNT = 0, PC = адрес CONTINUE Если CNT ≠ 0, PC = адрес HERE + 1</pre> | | | | |

IORLW Побитное 'ИЛИ' константы и W

| | | | | | |
|---------------|---|------|------|------|------|
| Синтаксис: | <i>[label]</i> IORLW k | | | | |
| Операнды: | $0 \leq k \leq 255$ | | | | |
| Операция: | $(W) .OR. k \rightarrow (W)$ | | | | |
| Измен. флаги: | Z | | | | |
| Код: | <table border="1"> <tr> <td>11</td> <td>1000</td> <td>kkkk</td> <td>kkkk</td> </tr> </table> | 11 | 1000 | kkkk | kkkk |
| 11 | 1000 | kkkk | kkkk | | |
| Описание: | <p>Выполняется побитное 'ИЛИ' содержимого регистра W и 8-разрядной константы 'k'. Результат сохраняется в регистре W.</p> | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | <pre>IORLW 0x35 До выполнения команды W = 0x9A После выполнения команды W = 0xBF</pre> | | | | |

| IORWF | Побитное 'ИЛИ' W и f | | | | |
|---------------|---|------|------|------|------|
| Синтаксис: | <i>[label]</i> IORWF f,d | | | | |
| Операнды: | $0 \leq f \leq 127$ $d \in [0,1]$ | | | | |
| Операция: | (W) .OR. (f) → (dest) | | | | |
| Измен. флаги: | Z | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>0100</td> <td>dfff</td> <td>ffff</td> </tr> </table> | 00 | 0100 | dfff | ffff |
| 00 | 0100 | dfff | ffff | | |
| Описание: | Выполняется побитное 'ИЛИ' содержимого регистров W и 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | IORWF RESULT,0 До выполнения команды RESULT = 0x13 W = 0x91 После выполнения команды RESULT = 0x13 W = 0x93 | | | | |

| MOVF | Переслать f | | | | |
|---------------|--|------|------|------|------|
| Синтаксис: | <i>[label]</i> MOVF f,d | | | | |
| Операнды: | $0 \leq f \leq 127$ $d \in [0,1]$ | | | | |
| Операция: | (f) → (dest) | | | | |
| Измен. флаги: | Z | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>1000</td> <td>dfff</td> <td>ffff</td> </tr> </table> | 00 | 1000 | dfff | ffff |
| 00 | 1000 | dfff | ffff | | |
| Описание: | Содержимое регистра 'f' пересылается в регистр адресата. Если d=0, значение сохраняется в регистре W. Если d=1, значение сохраняется в регистре 'f'. d=1 используется для проверки содержимого регистра 'f' на ноль. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | MOVF FSR,0 После выполнения команды W = значение регистра FSR | | | | |

| MOVLW | Переслать константу в W | | | | |
|---------------|--|------|------|------|------|
| Синтаксис: | <i>[label]</i> MOVLW k | | | | |
| Операнды: | $0 \leq k \leq 255$ | | | | |
| Операция: | $k \rightarrow (W)$ | | | | |
| Измен. флаги: | Нет | | | | |
| Код: | <table border="1"> <tr> <td>11</td> <td>00xx</td> <td>kkkk</td> <td>kkkk</td> </tr> </table> | 11 | 00xx | kkkk | kkkk |
| 11 | 00xx | kkkk | kkkk | | |
| Описание: | Переслать константу 'k' в регистр W. В неиспользуемых битах ассемблер устанавливает '0'. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | MOVLW 0x5A После выполнения команды W = 0x5A | | | | |

| MOVWF | Переслать W в f | | | | |
|---------------|---|------|------|------|------|
| Синтаксис: | <i>[label]</i> MOVWF f | | | | |
| Операнды: | $0 \leq f \leq 127$ | | | | |
| Операция: | $(W) \rightarrow (f)$ | | | | |
| Измен. флаги: | Нет | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>0000</td> <td>1fff</td> <td>ffff</td> </tr> </table> | 00 | 0000 | 1fff | ffff |
| 00 | 0000 | 1fff | ffff | | |
| Описание: | Переслать содержимое регистра W в регистр 'f'. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | MOVWF OPTION_REG До выполнения команды OPTION = 0xFF W = 0x4F После выполнения команды OPTION = 0x4F W = 0x4F | | | | |

| NOP | Нет операции | | | | |
|---------------|--|------|------|------|------|
| Синтаксис: | <i>[label]</i> NOP | | | | |
| Операнды: | Нет | | | | |
| Операция: | Нет операции | | | | |
| Измен. флаги: | Нет | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>0000</td> <td>0xx0</td> <td>0000</td> </tr> </table> | 00 | 0000 | 0xx0 | 0000 |
| 00 | 0000 | 0xx0 | 0000 | | |
| Описание: | Нет операции | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | NOP | | | | |

| OPTION | Загрузить регистр OPTION | | | | |
|--|---|--|------|------|------|
| Синтаксис: | <i>[label]</i> OPTION | | | | |
| Операнды: | Нет | | | | |
| Операция: | $(W) \rightarrow \text{OPTION}$ | | | | |
| Измен. флаги: | Нет | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0010</td> </tr> </table> | 00 | 0000 | 0110 | 0010 |
| 00 | 0000 | 0110 | 0010 | | |
| Описание: | Переслать содержимое регистра W в регистр OPTION. Инструкция поддерживается для совместимости программы с семейством микроконтроллеров PIC16C5X. Запись/чтение регистра OPTION можно выполнить прямой или косвенной адресацией. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример: | <table border="1"> <tr> <td>Для совместимости программного обеспечения с последующими выпускаемыми микроконтроллерами семейства PIC16CXX не рекомендуется использовать эту инструкцию.</td> </tr> </table> | Для совместимости программного обеспечения с последующими выпускаемыми микроконтроллерами семейства PIC16CXX не рекомендуется использовать эту инструкцию. | | | |
| Для совместимости программного обеспечения с последующими выпускаемыми микроконтроллерами семейства PIC16CXX не рекомендуется использовать эту инструкцию. | | | | | |

RETFIE **Возврат из подпрограммы с разрешением прерываний**Синтаксис: *[label]* RETFIE

Операнды: Нет

Операция: TOS → PC
1 → GIE

Измен. флаги: Нет

Код:

| | | | |
|----|------|------|------|
| 00 | 0000 | 0000 | 1001 |
|----|------|------|------|

Описание: Возврат из подпрограммы обработки прерываний. Вершина стека TOS загружается в счетчик команд PC. Устанавливается в '1' флаг глобального разрешения прерываний GIE(INTCON<7>). Инструкция выполняется за 2 цикла.

Слов: 1

Циклов: 2

Пример: RETFIE
До выполнения командыПосле выполнения команды
PC = TOS
GIE = 1**RETLW** **Возврат из подпрограммы с загрузкой константы в W**Синтаксис: *[label]* RETLW kОперанды: $0 \leq k \leq 255$ Операция: k → (W)
TOS → PC

Измен. флаги: Нет

Код:

| | | | |
|----|------|------|------|
| 11 | 01xx | kkkk | kkkk |
|----|------|------|------|

Описание: В регистр W загружается 8-разрядная константа. Вершина стека TOS загружается в счетчик команд PC. Инструкция выполняется за 2 цикла.

Слов: 1

Циклов: 2

Пример: CALL TABLE
•
•
TABLE ADDWF PCL,f
RETLW k1
RETLW k2
•
•
RETLW knДо выполнения команды
W = 0x07После выполнения команды
W = значение k8

RETURN Возврат из подпрограммыСинтаксис: `[label] RETURN`

Операнды: Нет

Операция: TOS → PC

Измен. флаги: Нет

Код:

| | | | |
|----|------|------|------|
| 00 | 0000 | 0000 | 1000 |
|----|------|------|------|

Описание: Возврат из подпрограммы. Вершина стека TOS загружается в счетчик команд PC. Инструкция выполняется за 2 цикла.

Слов: 1

Циклов: 2

Пример: RETURN

После выполнения команды
PC = TOS**RLF Циклический сдвиг f влево через перенос**Синтаксис: `[label] RLF f,d`Операнды: $0 \leq f \leq 127$
 $d \in [0,1]$

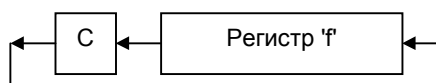
Операция: См. описание

Измен. флаги: C

Код:

| | | | |
|----|------|------|------|
| 00 | 1101 | dfff | ffff |
|----|------|------|------|

Описание: Выполняется циклический сдвиг влево содержимого регистра 'f' через бит C регистра STATUS. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.



Слов: 1

Циклов: 1

Пример: RLF REG1,0

До выполнения команды

REG1 = 1110 0110

C = 0

После выполнения команды

REG1 = 1110 0110

W = 1100 1100

C = 1

RRF **Циклический сдвиг f вправо через перенос**Синтаксис: `[label] RRF f,d`Операнды: $0 \leq f \leq 127$
 $d \in [0,1]$

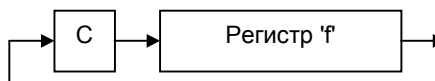
Операция: См. описание

Измен. флаги: C

Код:

| | | | |
|----|------|------|------|
| 00 | 1100 | dfff | ffff |
|----|------|------|------|

Описание: Выполняется циклический сдвиг вправо содержимого регистра 'f' через бит C регистра STATUS. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.



Слов: 1

Циклов: 1

Пример: `RRF REG1,0`

До выполнения команды

`REG1 = 1110 0110``C = 0`

После выполнения команды

`REG1 = 1110 0110``W = 0111 0011``C = 0`**SLEEP** **Перейти в режим SLEEP**Синтаксис: `[label] SLEEP`Операнды: Нет
`00h` → WDTОперация: `00h` → предделитель WDT`1` → - TO`0` → - PD

Измен. флаги: -TO, -PD

Код:

| | | | |
|----|------|------|------|
| 00 | 0000 | 0110 | 0011 |
|----|------|------|------|

Описание: Сбросить флаг включения питания -PD в '0'. Установить флаг переполнения WDT -TO в '1'. Очистить таймер WDT и его предделитель. Перевести микроконтроллер в режим SLEEP и выключить тактовый генератор. Подробное описание смотрите в разделе 14.9.

Слов: 1

Циклов: 1

Пример: `SLEEP`

SUBLW**Вычесть W из константы**Синтаксис: `[label] SUBLW k`Операнды: $0 \leq k \leq 255$ Операция: $k - (W) \rightarrow (W)$

Измен. флаги: C, DC, Z

Код:

| | | | |
|----|------|------|------|
| 11 | 110x | kkkk | kkkk |
|----|------|------|------|

Описание: Вычесть содержимое регистра W из 8-разрядной константы 'k'. Результат сохраняется в регистре W.

Слов: 1

Циклов: 1

Пример 1: `SUBLW 0x02`

До выполнения команды

W = 1

C = ?

Z = ?

После выполнения команды

W = 1

C = 1 ; результат положительный

Z = 0

Пример 2: До выполнения команды

W = 2

C = ?

Z = ?

После выполнения команды

W = 0

C = 1 ; результат нулевой

Z = 1

Пример 3: До выполнения команды

W = 3

C = ?

Z = ?

После выполнения команды

W = 0xFF

C = 0 ; результат отрицательный

Z = 0

| SUBWF | Вычесть W из f | | | | |
|---------------|---|------|------|------|------|
| Синтаксис: | <code>[label] SUBWF f,d</code> | | | | |
| Операнды: | $0 \leq f \leq 127$ $d \in [0,1]$ | | | | |
| Операция: | $(f) - (W) \rightarrow (\text{dest})$ | | | | |
| Измен. флаги: | C, DC, Z | | | | |
| Код: | <table border="1"> <tr> <td>00</td> <td>0010</td> <td>dfff</td> <td>ffff</td> </tr> </table> | 00 | 0010 | dfff | ffff |
| 00 | 0010 | dfff | ffff | | |
| Описание: | Вычесть содержимое регистра W из регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'. | | | | |
| Слов: | 1 | | | | |
| Циклов: | 1 | | | | |
| Пример 1: | <p><code>SUBWF REG1,1</code></p> <p>До выполнения команды</p> <p>REG1 = 3 W = 2 C = ? Z = ?</p> <p>После выполнения команды</p> <p>REG1 = 1 W = 2 C = 1 ; результат положительный Z = 0</p> | | | | |
| Пример 2: | <p>До выполнения команды</p> <p>REG1 = 2 W = 2 C = ? Z = ?</p> <p>После выполнения команды</p> <p>REG1 = 0 W = 2 C = 1 ; результат нулевой Z = 1</p> | | | | |
| Пример 3: | <p>До выполнения команды</p> <p>REG1 = 1 W = 2 C = ? Z = ?</p> <p>После выполнения команды</p> <p>REG1 = 0xFF W = 2 C = 0 ; результат отрицательный Z = 0</p> | | | | |

| SWAPF | Поменять местами полубайты в регистре f | | | |
|---------------|--|------|------|------|
| Синтаксис: | [<i>label</i>] SWAPF f,d | | | |
| Операнды: | 0 ≤ f ≤ 127 d ∈ [0,1] | | | |
| Операция: | (f<3:0>) → (dest<7:4>) (f<7:4>) → (dest<3:0>) | | | |
| Измен. флаги: | Нет | | | |
| Код: | 00 | 1110 | dfff | ffff |
| Описание: | Поменять местами старший и младший полубайты регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'. | | | |
| Слов: | 1 | | | |
| Циклов: | 1 | | | |
| Пример: | SWAPF REG,0 До выполнения команды REG = 0xA5 После выполнения команды REG = 0xA5 W = 0x5A | | | |
| TRIS | Загрузить регистр TRIS | | | |
| Синтаксис: | [<i>label</i>] TRIS f | | | |
| Операнды: | 5 ≤ f ≤ 7 | | | |
| Операция: | (W) → TRIS регистр f | | | |
| Измен. флаги: | Нет | | | |
| Код: | 00 | 0000 | 0110 | 0fff |
| Описание: | Переслать содержимое W в регистр TRIS. Инструкция поддерживается для совместимости программы с семейством микроконтроллеров PIC16C5X. Запись/чтение регистра OPTION можно выполнить прямой или косвенной адресацией. | | | |
| Слов: | | | | |
| Циклов: | | | | |
| Пример: | Для совместимости программного обеспечения с последующими выпускаемыми микроконтроллерами семейства PIC16CXX не рекомендуется использовать эту инструкцию. | | | |

XORLW **Побитное 'исключающее ИЛИ' константы и W**Синтаксис: `[label] XORLW k`Операнды: $0 \leq k \leq 255$ Операция: $(W) .XOR. k \rightarrow (W)$

Измен. флаги: Z

Код:

| | | | |
|----|------|------|------|
| 11 | 1010 | kkkk | kkkk |
|----|------|------|------|

Описание: Выполняется побитное 'исключающее ИЛИ' содержимого регистра W и 8-разрядной константы 'k'. Результат сохраняется в регистре W.

Слов: 1

Циклов: 1

Пример: `XORLW 0x35`

До выполнения команды

W = 0xAF

После выполнения команды

W = 0x1A

XORWF **Побитное 'исключающее ИЛИ' W и f**Синтаксис: `[label] XORWF f,d`Операнды: $0 \leq f \leq 127$ $d \in [0,1]$ Операция: $(W) .XOR. (f) \rightarrow (dest)$

Измен. флаги: Z

Код:

| | | | |
|----|------|------|------|
| 00 | 0100 | dfff | ffff |
|----|------|------|------|

Описание: Выполняется побитное 'исключающее ИЛИ' содержимого регистров W и 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.

Слов: 1

Циклов: 1

Пример: `XORWF REG,1`

До выполнения команды

REG = 0xAF

W = 0xB5

После выполнения команды

REG = 0x1A

W = 0xB5

16.0 Поддержка разработчиков

Микроконтроллеры PICmicro обеспечены большим спектром аппаратных и программных инструментальных средств проектирования:

- Интегрированная среда проектирования:
 - Программное обеспечение MPLAB IDE.
- Ассемблер/Компилятор/Линкер:
 - Ассемблер MPASM;
 - Компиляторы MLAB-C17 и MPLAB-C18;
 - Линкер MPLINK/ Организатор библиотек MPLIB.
- Симулятор:
 - Программный симулятор MLAB-SIM.
- Эмуляторы:
 - Внутрисхемный эмулятор реального времени MPLAB-ICE;
 - Внутрисхемный эмулятор PICMASTER/PICMASTER-CE;
 - ICEPIC.
- Внутрисхемный отладчик:
 - MLAB-ICD для микроконтроллеров семейства PIC16F87X.
- Программаторы:
 - Универсальный программатор PRO MATE II;
 - Недорогой программатор PICSTART для начала работы.
- Недорогие демонстрационные платы:
 - SIMICE;
 - PICDEM-1;
 - PICDEM-2;
 - PICDEM-3;
 - PICDEM-17;
 - SEEVAL;
 - KeeLoq.

16.1 Интегрированная среда проектирования MPLAB-IDE

Программное обеспечение MPLAB-IDE предназначено для разработки программного обеспечения 8-разрядных микроконтроллеров PICmicro, работающее под управлением операционной системы Windows.

Основные характеристики MPLAB-IDE:

- Многофункциональные возможности:
 - Редактор;
 - Симулятор;
 - Программатор (приобретается отдельно);
 - Эмулятор (приобретается отдельно).
- Полнофункциональный редактор.
- Организатор проекта.
- Настройка панелей инструментов и параметров отображения.
- Строка состояния.
- Интерактивная помощь.

MPLAB-IDE позволяет Вам:

- Редактировать исходные файлы написанные на языке ассемблера или C.
- Быстро выполнять трансляцию и компиляцию проекта автоматически загружая параметры используемого микроконтроллера PICmicro.
- Выполнять отладку программы с использованием:
 - Исходных файлов;
 - Листинга программы;
 - Объектного кода.

Однотипная работа инструментальных модулей интегрированной среды проектирования MPLAB-IDE позволяет легко перейти от программного симулятора MPLAB-SIM к использованию полнофункционального эмулятора.

16.2 Ассемблер MPASM

MPASM - полнофункциональный универсальный макроассемблер для всех семейств микроконтроллеров PICmicro. Ассемблер может генерировать шестнадцатиразрядный файл пригодный для записи в микроконтроллер или формировать перемещаемые объектные файлы для линкера MPLINK.

MPASM имеет интерфейс командной строки и оконный интерфейс, работает под управлением операционной системы Windows 3.X и выше, может работать как автономное приложение. MPASM генерирует объектные файлы, шестнадцатеричные HEX файлы в стандарте Intel, файл карты памяти (для детализации использования памяти микроконтроллера), файл листинга программы (текст программы совмещен с кодами микроконтроллера) и файл отладки для MPLAB-IDE.

Особенности MPASM:

- MPASM и MPLINK интегрированы в MPLAB-IDE;
- MPASM поддерживает систему макрокоманд, упрощающих написание текста программы;
- Позволяет выполнять компиляцию условных блоков текста программы;
- Директивы MPASM дают возможность управлять компиляцией исходного текста программы.

16.3 С компиляторы MPLAB-C17 и MPLAB-C18

MPLAB-C17 и MPLAB-C18 - полнофункциональные ANSI 'C' компиляторы с интегрированной средой обработки для микроконтроллеров семейств PIC17CXXX и PIC18CXXX соответственно. Для упрощения отладки текста программы компиляторы обеспечивают интеграцию в средства проектирования с передачей информации об используемых переменных в формате совместимом с MPLAB-IDE.

16.4 Линкер MPLINK/ Организатор библиотек MPLIB

MPLINK - линкер перемещаемых объектных файлов сгенерированных программами MPASM, MPLAB-C17 и MPLAB-C18. Линкер выполняет связь объектных файлов с предварительно скомпилированными файлами библиотек и файлами сценария.

MPLIB - организатор библиотек предварительно откомпилированных исходных файлов, которые нужно использовать с MPLINK. Когда подпрограмма библиотечного файла вызывается из исходного файла, в приложение будет включена только необходимый модуль. Это позволяет эффективно использовать большие библиотеки в различных приложениях. MPLIB управляет созданием и изменением библиотечных файлов.

Особенности MPLINK:

- MPLINK работает совместно с MPASM, MPLAB-C17 и MPLAB-C18;
- MPLINK позволяет разбивать память микроконтроллера на разделы.

Особенности MPLIB:

- MPLIB упрощает подключение дополнительных файлов потому, что позволяет подключить одну библиотеку вместо множества мелких файлов;
- MPLIB группирует связанные модули;
- MPLIB позволяет добавлять, изменять, удалять и заменять модули в библиотечных файлах.

16.5 Программный симулятор MPLAB-SIM

Симулятор MPLAB-SIM позволяет проследить выполнение программы микроконтроллеров PICmicro на уровне команд по шагам или в режиме анимации. На любой команде выполнение программы может быть остановлено для проверки и изменения памяти. Функции стимула позволяют моделировать сигнал с логическими уровнями на входах микроконтроллера. MPLAB-SIM полностью поддерживает символьную отладку, используя MPLAB-C17, MPLAB-C18 и MPASM. MPLAB-SIM является доступным и удобным средством отладки программ микроконтроллеров PICmicro.

16.6 Универсальный эмулятор MPLAB-ICE

Универсальный эмулятор MPLAB-CE обеспечивает разработчиков полным набором инструментальных средств для проектирования устройств с применением микроконтроллеров PICmicro. Управление работой эмулятора выполняется из интегрированной среды проектирования MPLAB-IDE с возможностью редактирования, компиляции, загрузки и выполнения программы.

Заменяемые поды позволяют быстро перенастроить эмулятор для работы с другим типом микроконтроллеров. Универсальная архитектура MPLAB-ICE дает возможность поддерживать новые типы микроконтроллеров PICmicro.

Эмулятор MPLAB-ICE был разработан как система эмуляции (анимации) в реальном масштабе времени с дополнительными возможностями, присутствующих в дорогих инструментальных средствах. Эмулятор работает под управлением распространенной операционной системы Microsoft Windows 3.x/95/98.

MPLAB-ICE 2000 - полнофункциональная система эмуляции с усовершенствованными функциями трассировки, триггеров и управляющих особенностей. Оба эмулятора используют одинаковые поды и работают во всех допустимых режимах микроконтроллеров PICmicro.

16.7 PICMASTER/PICMASTER-CE

PICMASTER эмулятор компании Microchip для профессиональных разработчиков. Этот эмулятор представляет собой высококачественную универсальную платформу для эмуляции 8-разрядных микроконтроллеров PICmicro. Эмуляторы PICMASTER продаются во всем мире, эмуляторы с префиксом CE предназначены для стран ЕС.

16.8 ICEPIC

ICEPIC - недорогой эмулятор, предназначенный для однократно программируемых (OTP) 8-разрядных микроконтроллеров семейств PIC16C5X, PIC16C6X, PIC16C7X и PIC16CXXX. Модульная структура позволяет поддерживать все типы микроконтроллеров семейства PIC16C5X и PIC16CXXX за счет сменных подов.

16.9 Внутрисхемный отладчик MPLAB-ICD

Внутрисхемный отладчик MPLAB-ICD является мощным недорогим инструментом отладки программы. Работа MPLAB-ICD основана на функции внутрисхемной отладки Flash микроконтроллеров семейства PIC16F87X. Эта особенность, совместно с функцией внутрисхемного последовательного программирования, позволяет запрограммировать микроконтроллер непосредственно из среды проектирования MPLAB IDE. MPLAB-ICD позволяет быстро выполнить отладку программы, выполняя ее по шагам, в режиме анимации или в режиме реального времени.

16.10 Универсальный программатор PRO MATE II

Универсальный программатор PRO MATE II может работать автономно и под управлением PC совместимого компьютера. Для максимальной надежности программирования в программаторе PRO MATE II можно указать напряжения V_{DD} и V_{PP} . В программатор встроен ЖКИ дисплей для вывода сообщений об ошибках и клавиатура для ввода команд. Модульная колодка позволяет программировать микросхемы в различных корпусах. В автономном режиме программатор PRO MATE II может проверять микроконтроллер и устанавливать биты защиты.

16.11 Программатор PICSTART

Недорогой программатор PICSTART (PICSTART+CE) предназначен для начала работы с микроконтроллерами PICmicro, подключается к PC совместимому компьютеру через COM (RS-232) порт и работает под управлением интегрированной среды проектирования MPLAB IDE. PICSTART поддерживает все микроконтроллеры PICmicro в корпусах до 40 выводов. Микроконтроллеры с большим числом выводов (PIC16C92X, PIC17C76X) поддерживаются при использовании адаптеров.

16.12 Аппаратный модуль SIMICE

SIMICE предназначен для работы совместно с симулятором MPLAB-SIM. SIMICE и MPLAB-SIM работают под управлением интегрированной среды проектирования MPLAB-IDE. SIMICE поддерживает работу микроконтроллеров PIC12C5XX, PIC12CE5XX и PIC16C5X с эмуляцией портов ввода/вывода не в реальном масштабе времени. SIMICE позволяет разработчику выполнить код программы непосредственно в устройстве, что освобождает от необходимости написания файлов стимула. SIMICE ценный инструмент отладки программного обеспечения при начале работы с микроконтроллерами PICmicro.

16.13 Демонстрационная плата PICDEM-1

Демонстрационная плата PICDEM-1 предназначена для микроконтроллеров PIC16C5X (PIC26C54, PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 и PIC17C44. В комплект поставки входят необходимые аппаратные модули, программное обеспечение и демонстрационные программы. Записать демонстрационные программы в микроконтроллер можно с помощью программатора PRO MATE II или PICSTART. Пользователь может подключить к демонстрационной плате эмулятор MPLAB-ICE и выполнять отладку программы. На демонстрационной плате имеется полигон для установки дополнительных элементов пользователя. В состав демонстрационной платы входит: драйвер интерфейса RS-232, потенциометр для моделирования аналогового входа, выключатели и восемь светодиодов подключенных к PORTB.

16.14 Демонстрационная плата PICDEM-2 для PIC16CXXX

Демонстрационная плата PICDEM-2 предназначена для микроконтроллеров PIC16C62, PIC16C64, PIC16C65, PIC16C73 и PIC16C74. В комплект поставки входят необходимые аппаратные модули, программное обеспечение и демонстрационные программы. Записать демонстрационные программы в микроконтроллер можно с помощью программатора PRO MATE II или PICSTART. Пользователь может подключить к демонстрационной плате эмулятор MPLAB-ICE и выполнять отладку программы. На демонстрационной плате имеется полигон для установки дополнительных элементов пользователя. В состав демонстрационной платы входит: драйвер интерфейса RS-232, потенциометр для моделирования аналогового входа, последовательная EEPROM для демонстрации работы шины I2C, выводы для подключения ЖКИ и дополнительной клавиатуры.

16.15 Демонстрационная плата PICDEM-3 для PIC16CXXX

Демонстрационная плата PICDEM-3 предназначена для микроконтроллеров PIC16C923 и PIC16C924 выполненных в 44-выводном PLCC корпусе с интегрированным ЖКИ модулем. В комплект поставки входят необходимые аппаратные модули, программное обеспечение и демонстрационные программы. Записать демонстрационные программы в микроконтроллер можно с помощью программатора PRO MATE II или PICSTART. Пользователь может подключить к демонстрационной плате эмулятор MPLAB-ICE и выполнять отладку программы. На демонстрационной плате имеется полигон для установки дополнительных элементов пользователя. В состав демонстрационной платы входит: драйвер интерфейса RS-232, выключатели; потенциометр для моделирования аналогового входа; термистор; выводы для подключения ЖКИ и дополнительной клавиатуры; 12-разрядный ЖКИ для отображения времени, даты и температуры; дополнительный интерфейс RS-232; программное обеспечение работающее под управлением операционной системы Windows 3.x для передачи данных на PC совместимый компьютер.

16.16 Демонстрационная плата PICDEM-17

Демонстрационная плата PICDEM-17 предназначена для микроконтроллеров PIC17C752, PIC17C756, PIC17C762 и PIC17C766. В комплект поставки входят необходимые аппаратные модули, программное обеспечение и демонстрационные программы. Записать демонстрационные программы в микроконтроллер можно с помощью программатора PRO MATE II или PICSTART. Пользователь может подключить к демонстрационной плате эмулятор MPLAB-ICE и выполнять отладку программы. На демонстрационной плате имеется полигон для установки дополнительных элементов пользователя.

16.17 SEEVAL (с функциями программатора)

Комплект SEEVAL EEPROM поддерживает весь спектр 2-х/3-х проводных последовательных микросхем EEPROM фирмы Microchip. Комплект позволяет выполнять чтение, стирание и запись любой микросхемы последовательного EEPROM фирмы Microchip. Система позволяет сделать анализ обмена данных, число циклов и надежность записи. Полный комплект SEEVAL позволяет уменьшить время проектирование устройства.

16.18 KeeLoq (с функциями программатора)

Оценочная система KeeLoq предназначена для микросхем HCS фирмы Microchip. В состав комплекта входит: ЖКИ дисплей для отображения изменяющихся кодов, декодер, интерфейс программирования.

17.0 Электрические характеристики

Максимально допустимые значения (*)

| | |
|--|---------------------------|
| Предельная рабочая температура | от -40°C до +125°C |
| Температура хранения | от -65°C до +150°C |
| Напряжение V_{DD} относительно V_{SS} | от -0.3В до +6.5В |
| Напряжение -MCLR и RA4 относительно V_{SS} | от -0.3В до +14В |
| Напряжение на остальных выводах относительно V_{SS} | от -0.3В до $V_{DD}+0.3В$ |
| Рассеиваемая мощность ⁽¹⁾ | 800мВт |
| Максимальный ток вывода V_{SS} | 300мА |
| Максимальный ток вывода V_{DD} | 250мА |
| Входной запирающий ток I_{IK} ($V_I < 0$ или $V_I > V_{DD}$) | ± 20 мА |
| Выходной запирающий ток I_{OK} ($V_O < 0$ или $V_O > V_{DD}$) | ± 20 мА |
| Максимальный выходной ток стока канала ввода/вывода | 25мА |
| Максимальный выходной ток истока канала ввода/вывода | 25мА |
| Максимальный выходной ток стока портов ввода/вывода PORTA и PORTB | 200мА |
| Максимальный выходной ток истока портов ввода/вывода PORTA и PORTB | 200мА |

Примечание 1. Потребляемая мощность рассчитывается по формуле:

$$P = V_{DD} \times \{I_{DD} - \Sigma I_{OH}\} + \Sigma \{(V_{DD} - V_{OH}) \times I_{OH}\} + \Sigma (V_{OL} \times I_{OL})$$

Примечание * Выход за указанные значения может привести к необратимым повреждениям микроконтроллера. Не предусмотрена работа микроконтроллера в предельном режиме в течении длительного времени. Длительная эксплуатация микроконтроллера в допустимых условиях может повлиять на его надежность.

Примечание. Броски напряжения на выводе -MCLR ниже V_{SS} приводят к появлению больших токов (около 80мА), что может привести к срабатыванию защелки. Поэтому рекомендуется последовательно включать резистор сопротивлением от 500Ом до 1000Ом для подачи низкого уровня на этот вывод вместо непосредственного подключения к V_{SS} .

Рис. 17-1 График рекомендованных комбинаций значений напряжения питания и тактовой частоты для PIC16F62X, $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ (рекомендованная комбинация отмечена затененной областью)

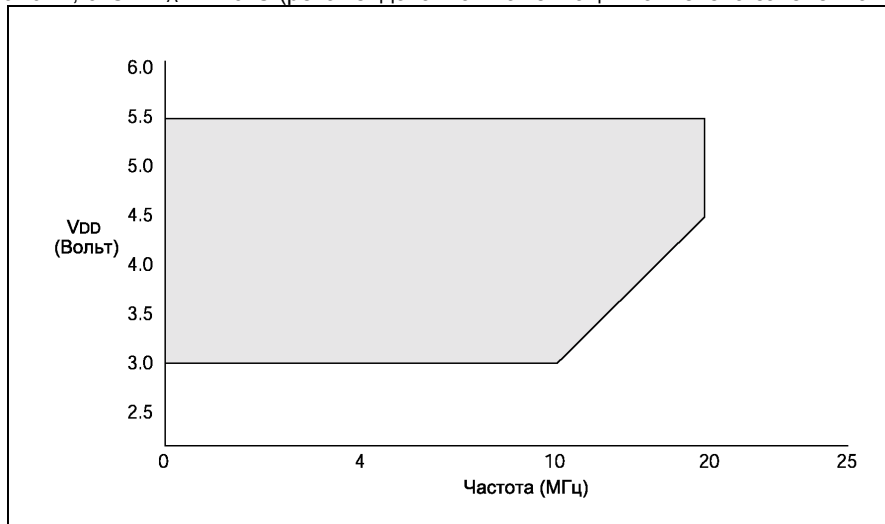


Рис. 17-2 График рекомендованных комбинаций значений напряжения питания и тактовой частоты для PIC16F62X, $-40^{\circ}\text{C} \leq T_A < 0^{\circ}\text{C}$, $+70^{\circ}\text{C} < T_A \leq +85^{\circ}\text{C}$ (рекомендованная комбинация отмечена затененной областью)

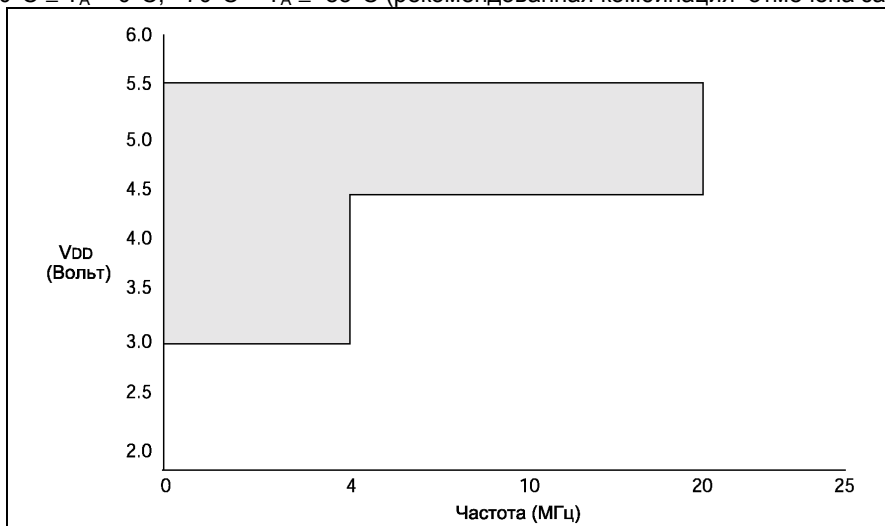


Рис. 17-3 График рекомендованных комбинаций значений напряжения питания и тактовой частоты для PIC16LF62X, $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ (рекомендованная комбинация отмечена затененной областью)

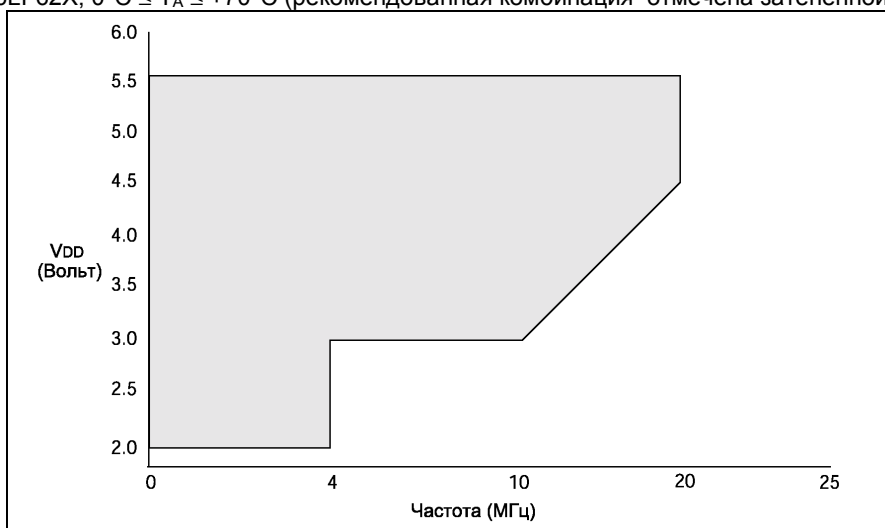
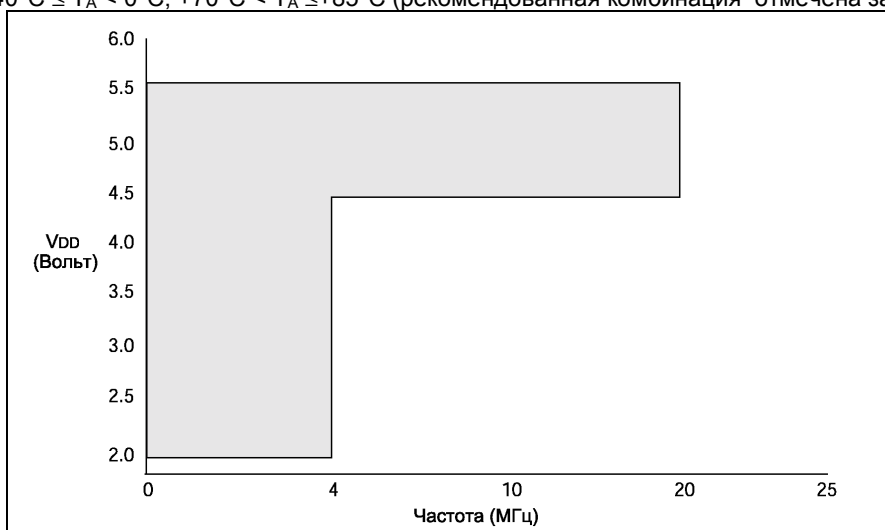


Рис. 17-4 График рекомендованных комбинаций значений напряжения питания и тактовой частоты для PIC16LF62X, $-40^{\circ}\text{C} \leq T_A < 0^{\circ}\text{C}$, $+70^{\circ}\text{C} < T_A \leq +85^{\circ}\text{C}$ (рекомендованная комбинация отмечена затененной областью)



17.1 Электрические характеристики:**PIC16F62X-04 (Коммерческий, Промышленный, Расширенный)****PIC16F62X-20 (Коммерческий, Промышленный, Расширенный)**

| Стандартные рабочие условия (если не указано иное) | | | | | | | |
|--|--------------------|---|--|-----------------|-------|------|---|
| Температурный диапазон: | | | | | | | |
| | | Коммерческий | $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ | | | | |
| | | Промышленный | $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ | | | | |
| | | Расширенный | $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ | | | | |
| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
| D001 | V _{DD} | Напряжение питания | 3.0 | - | 5.5 | В | |
| D002 | V _{DR} | Напряжение сохранения данных в ОЗУ ⁽¹⁾ | - | 1.5* | - | В | В SLEEP режиме |
| D003 | V _{POR} | Стартовое напряжение V _{DD} для формирования POR | - | V _{SS} | - | В | Смотрите раздел "сброс POR" |
| D004 | S _{VDD} | Скорость нарастания V _{DD} для формирования POR | 0.05* | - | - | В/мс | Смотрите раздел "сброс POR" |
| D005 | V _{BOD} | Напряжение детектора BOD | 3.65 | 4.0 | 4.35 | В | Бит BODEN = 0 |
| D010 | I _{DD} | Ток потребления ^(2,5) | - | - | 0.7 | мА | F _{OSC} =4.0МГц, V _{DD} =3.0В |
| D013 | | | | 4.0 | 2.0 | мА | F _{OSC} =4.0МГц, V _{DD} =5.5В* |
| | | | | | 7.0 | мА | F _{OSC} =20.0МГц, V _{DD} =5.5В* |
| D014 | | | | | 6.0 | мА | F _{OSC} =20.0МГц, V _{DD} =4.5В |
| | | | | | 2.0 | мА | F _{OSC} =10.0МГц, V _{DD} =3.0В ⁽⁶⁾ |
| D020 | I _{PD} | Ток потребления в SLEEP режиме ⁽³⁾ | | | 2.2 | мкА | V _{DD} =3.0В |
| | | | | | 5.0 | мкА | V _{DD} =4.5В |
| | | | | | 9.0 | мкА | V _{DD} =5.5В* |
| | | | | | 15.0 | мкА | V _{DD} =5.5В* (расширенный) |
| D022 | ΔI _{WDT} | Ток потребления WDT ⁽⁴⁾ | - | 6.0 | 20 | мкА | V _{DD} =4.0В |
| D022A | ΔI _{BOD} | Ток потребления BOD ⁽⁴⁾ | - | 75 | 125 | мкА | (125°C) |
| D023 | ΔI _{COMP} | Ток потребления каждого компаратора ⁽⁴⁾ | - | 30 | 50 | мкА | BOD включен, V _{DD} =5.0В |
| D023A | ΔI _{VREF} | Ток потребления V _{REF} ⁽⁴⁾ | - | - | 135 | мкА | V _{DD} =4.0В |
| 1A | F _{OSC} | LP генератор | 0 | - | 200 | кГц | Любая температура |
| | | XT генератор | 0 | - | 4 | МГц | Любая температура |
| | | HS генератор | 0 | - | 20 | МГц | Любая температура |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при V_{DD}=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.**Примечания:**

- Предел, до которого может быть понижено напряжение питания V_{DD} в SLEEP режиме без потери данных в ОЗУ.
- Ток потребления в основном зависит от напряжения питания и тактовой частоты. Другие факторы влияющие на ток потребления: выходная нагрузка и частота переключения каналов ввода/вывода; тип тактового генератора; температура и выполняемая программа. Измерения I_{DD} проводилось в следующих условиях: внешний тактовый сигнал (меандр); каналы портов ввода/вывода в третьем состоянии и подтянуты к V_{DD}; -MCLR = V_{DD}; WDT выключен.
- Потребляемый ток в SLEEP режиме не зависит от типа тактового генератора. При измерении тока все каналы портов ввода/вывода в третьем состоянии и подтянуты к V_{DD} или V_{SS}.
- Δ ток - дополнительный потребляемый ток, если периферийный модуль включен. Этот ток должен быть добавлен к I_{DD} или I_{PD}.
- В ER режиме генератора ток через внешний резистор не учитывается. Ток протекающий через внешний резистор может быть рассчитан по формуле $I_r = V_{DD}/2R_{EXT}$ (мА), где R_{EXT} в кОм.
- Только для коммерческого диапазона температур.

17.2 Электрические характеристики:**PIC16LF62X-04 (Коммерческий, Промышленный, Расширенный)****PIC16LF62X-20 (Коммерческий, Промышленный, Расширенный)**

| Стандартные рабочие условия (если не указано иное) | | | | | | | |
|--|--------------------|---|--|-----------------|-------|------|---|
| Температурный диапазон: | | | | | | | |
| | | Коммерческий | $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ | | | | |
| | | Промышленный | $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ | | | | |
| | | Расширенный | $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ | | | | |
| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
| D001 | V _{DD} | Напряжение питания | 2.0 | - | 5.5 | В | |
| D002 | V _{DR} | Напряжение сохранения данных в ОЗУ ⁽¹⁾ | - | 1.5* | - | В | В SLEEP режиме |
| D003 | V _{POR} | Стартовое напряжение V _{DD} для формирования POR | - | V _{SS} | - | В | Смотрите раздел "сброс POR" |
| D004 | S _{VDD} | Скорость нарастания V _{DD} для формирования POR | 0.05* | - | - | В/мс | Смотрите раздел "сброс POR" |
| D005 | V _{BOD} | Напряжение детектора BOD | 3.65 | 4.0 | 4.35 | В | Бит BODEN = 0 |
| D010 | I _{DD} | Ток потребления ^(2,5) | - | - | 0.6 | мА | F _{OSC} =4.0МГц, V _{DD} =2.0В ⁽⁵⁾ |
| D013 | | | | 4.0 | 0.7 | мА | F _{OSC} =4.0МГц, V _{DD} =5.5В* |
| | | | | | 7.0 | мА | F _{OSC} =20.0МГц, V _{DD} =4.5В |
| D014 | | | | | 6.0 | мА | F _{OSC} =10.0МГц, V _{DD} =3.0В ⁽⁶⁾ |
| | | | | | 2.0 | мА | F _{OSC} =32кГц, V _{DD} =3.0В |
| D020 | I _{PD} | Ток потребления в SLEEP режиме ⁽³⁾ | | | 2.0 | мкА | V _{DD} =2.5В* |
| | | | | | 2.2 | мкА | V _{DD} =3.3В* |
| | | | | | 9.0 | мкА | V _{DD} =5.5В |
| | | | | | 15.0 | мкА | V _{DD} =5.5В* (расширенный) |
| D022 | ΔI _{WDT} | Ток потребления WDT ⁽⁴⁾ | - | 6.0 | 15 | мкА | V _{DD} =3.0В |
| D022A | ΔI _{BOD} | Ток потребления BOD ⁽⁴⁾ | - | 75 | 125 | мкА | BOD включен, V _{DD} =5.0В |
| D023 | ΔI _{COMP} | Ток потребления каждого компаратора ⁽⁴⁾ | - | 30 | 50 | мкА | V _{DD} =3.0В |
| D023A | ΔI _{VREF} | Ток потребления V _{REF} ⁽⁴⁾ | - | - | 135 | мкА | V _{DD} =3.0В |
| 1A | F _{OSC} | LP генератор | 0 | - | 200 | кГц | Любая температура |
| | | XT генератор | 0 | - | 4 | МГц | Любая температура |
| | | HS генератор | 0 | - | 20 | МГц | Любая температура |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при V_{DD}=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.**Примечания:**

1. Предел, до которого может быть понижено напряжение питания V_{DD} в SLEEP режиме без потери данных в ОЗУ.
2. Ток потребления в основном зависит от напряжения питания и тактовой частоты. Другие факторы влияющие на ток потребления: выходная нагрузка и частота переключения каналов ввода/вывода; тип тактового генератора; температура и выполняемая программа. Измерения I_{DD} проводилось в следующих условиях: внешний тактовый сигнал (меандр); каналы портов ввода/вывода в третьем состоянии и подтянуты к V_{DD}; -MCLR = V_{DD}; WDT выключен.
3. Потребляемый ток в SLEEP режиме не зависит от типа тактового генератора. При измерении тока все каналы портов ввода/вывода в третьем состоянии и подтянуты к V_{DD} или V_{SS}.
4. Δ ток - дополнительный потребляемый ток, если периферийный модуль включен. Этот ток должен быть добавлен к I_{DD} или I_{PD}.
5. В ER режиме генератора ток через внешний резистор не учитывается. Ток протекающий через внешний резистор может быть рассчитан по формуле $I_r = V_{DD}/2R_{EXT}$ (мА), где R_{EXT} в кОм.
6. Только для коммерческого диапазона температур.

17.3 Электрические характеристики:**PIC16F62X (Коммерческий, Промышленный, Расширенный)
PIC16LF62X (Коммерческий, Промышленный, Расширенный)**

| Стандартные рабочие условия (если не указано иное) | | | | | | | |
|--|------------|--|--|--------|---------------|-----|--|
| Температурный диапазон: | | | | | | | |
| | | Коммерческий | $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ | | | | |
| | | Промышленный | $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ | | | | |
| | | Расширенный | $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ | | | | |
| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
| D030 | V_{IL} | Входное напряжение низкого уровня | | | | | $V_{DD} = \text{от } 4.5\text{В до } 5.5\text{В}$ иначе |
| | V_{IL} | Канал порта ввода/вывода | | | | | |
| | | ТТЛ буфер | V_{SS} | - | 0.8 | В | |
| | | | V_{SS} | - | $0.15V_{DD}$ | В | |
| | | Триггер Шмидта | V_{SS} | - | $0.2V_{DD}$ | В | |
| D031 | | -MCLR, RA4/T0CKI, OSC1 (ER) ⁽¹⁾ | V_{SS} | - | $0.2V_{DD}$ | В | |
| D032 | | OSC1 (XT,HS) | V_{SS} | - | $0.3V_{DD}$ | В | |
| D033 | | OSC1 (LP) | V_{SS} | - | $0.6V_{DD}-1$ | В | |
| D040 | V_{IH} | Входное напряжение высокого уровня | | | | | $V_{DD} = \text{от } 4.5\text{В до } 5.5\text{В}$ иначе |
| | V_{IH} | Канал порта ввода/вывода | | | | | |
| | | ТТЛ буфер | 2.0 | - | V_{DD} | В | |
| | | | $0.25V_{DD}+0.8$ | - | V_{DD} | В | |
| | | Триггер Шмидта | $0.8V_{DD}$ | - | V_{DD} | В | |
| | | -MCLR, RA4/T0CKI | $0.8V_{DD}$ | - | V_{DD} | В | |
| D042 | | OSC1 (XT,HS, LP) | $0.7V_{DD}$ | - | V_{DD} | В | |
| D043 | | OSC1 (ER) ⁽¹⁾ | $0.9V_{DD}$ | - | V_{DD} | В | |
| D043A | | OSC1 (ER) ⁽¹⁾ | $0.9V_{DD}$ | - | V_{DD} | В | |
| D070 | I_{PURB} | Ток через подтягивающие резисторы PORTB | 50 | 200 | 400 | мкА | $V_{DD} = 5.0\text{В}, V_{PIN} = V_{SS}$ |
| D060 | I_{IL} | Входной ток утечки ^(2,3) | - | - | ± 1.0 | мкА | $V_{SS} \leq V_{PIN} \leq V_{DD}$, 3-е сост. $V_{SS} \leq V_{PIN} \leq V_{DD}$, 3-е сост. $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$, XT, HS, LP |
| | | Порт ввода/вывода (искл.PORTA) | - | - | ± 0.5 | мкА | |
| | | PORTA | - | - | ± 1.0 | мкА | |
| | | RA4/T0CKI | - | - | ± 5.0 | мкА | |
| D061 | | OSC1/MCLR | - | - | ± 5.0 | мкА | |
| D080 | V_{OL} | Выходное напряжение низкого уровня | | | | | $V_{DD} = 4.5\text{В}$ $I_{OL}=8.5\text{ мА}, -40^{\circ}\text{C до } +85^{\circ}\text{C}$ $I_{OL}=7.0\text{ мА}, +125^{\circ}\text{C}$ $I_{OL}=1.6\text{ мА}, -40^{\circ}\text{C до } +85^{\circ}\text{C}$ $I_{OL}=1.2\text{ мА}, +125^{\circ}\text{C}$ |
| | | Канал ввода/вывода | - | - | 0.6 | В | |
| | | | - | - | 0.6 | В | |
| | | | - | - | 0.6 | В | |
| D083 | | OSC2/CLKOUT (ER) | - | - | 0.6 | В | |
| D090 | V_{OH} | Выходное напряжение высокого уровня ⁽³⁾ | | | | | $V_{DD} = 4.5\text{В}$ $I_{OH}=-3.0\text{ мА}, -40^{\circ}\text{C до } +85^{\circ}\text{C}$ $I_{OH}=-2.5\text{ мА}, +125^{\circ}\text{C}$ $I_{OH}=-1.3\text{ мА}, -40^{\circ}\text{C до } +85^{\circ}\text{C}$ $I_{OH}=-1.0\text{ мА}, +125^{\circ}\text{C}$ |
| | | Канал ввода/вывода (кроме RA4) | $V_{DD} - 0.7$ | - | - | В | |
| | | | $V_{DD} - 0.7$ | - | - | В | |
| | | | $V_{DD} - 0.7$ | - | - | В | |
| D092 | | OSC2/CLKOUT (ER) | $V_{DD} - 0.7$ | - | - | В | |
| D150* | V_{OD} | Напряжение на выходе с открытым стоком | - | - | 8.5* | В | RA4 для PIC16F62X, PIC16LF62X |
| D100 | C_{OSC2} | Емкостная нагрузка на выходах | | | | | XT, HS, LP |
| | | Вывод OSC2 | - | - | 15 | пФ | |
| D101 | C_{IO} | Все каналы ввода/вывода и OSC2 в ER режиме | - | - | 50 | пФ | |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при $V_{DD}=5.0\text{В}$ @ 25°C , если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.**Примечания:**

1. В ER режиме генератора на входе OSC1 включен триггер Шмидта. Не рекомендуется использовать внешний тактовый сигнал для PIC16F62X в ER режиме тактового генератора.
2. Ток утечки на выводе -MCLR зависит от приложенного напряжения. Параметры указаны для нормального режима работы. В других режимах может возникнуть больший ток утечки.
3. Отрицательный ток показывает, что он вытекает из вывода.

Таблица 17-1 Параметры компараторов

| Рабочие условия: $3.0V \leq V_{DD} \leq 5.5V$, $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ (если не указано иное) | | | | | | | |
|---|-------------|------------------------------|------|-----------|--------------|----------|-------------------------|
| № пар. | Обоз. | Описание | Мин. | Тип. | Макс. | Ед. | Примечание |
| D300 | V_{IOFF} | Входное напряжение смещения | - | ± 5.0 | ± 10 | мВ | |
| D301 | V_{ICM} | Входное напряжение* | 0 | - | $V_{DD}-1.5$ | В | |
| D302 | CMRR | Коэффициент отражения* | 55 | - | - | db | |
| 300 300A | T_{RESP} | Время реакции ⁽¹⁾ | - | 150 | 400 600 | нс нс | PIC16F62X PIC16LF62X |
| 301 | T_{MC2OV} | Время смены режима* | - | - | 10 | мкс | |

* - Эти параметры определены, но не протестированы.

Примечание 1. Время реакции измерялось при напряжении на одном из входов $(V_{DD}-1.5)/2$, а на другом был сформирован переход от V_{SS} к V_{DD} .

Таблица 17-2 Параметры источника опорного напряжения

| Рабочие условия: $3.0V \leq V_{DD} \leq 5.5V$, $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ (если не указано иное) | | | | | | | |
|---|-----------|--------------------------------|-------------|------|-------------|------------|----------------|
| № пар. | Обоз. | Описание | Мин. | Тип. | Макс. | Ед. | Примечание |
| D310 | V_{RES} | Разрешающая способность | $V_{DD}/24$ | - | $V_{DD}/32$ | Lsb | |
| D311 | VR_{AA} | Абсолютная точность | - | - | 1/4 1/2 | Lsb Lsb | VRR=1 VRR=0 |
| D312 | VR_{UR} | Сопротивление резистора R* | - | 2 | - | кОм | |
| 310 | T_{SET} | Время установки ⁽¹⁾ | - | - | 10 | мкс | |

* - Эти параметры определены, но не протестированы.

Примечание 1. Время измерено при VRR=1 и переходе VR<3:0> от 0000 к 1111.

17.4 Символьное обозначение временных параметров

Символьное обозначение временных параметров имеет один из следующих форматов:

1. TppS2ppS
2. TppS

| | | | |
|----------|---------|---|-------|
| T | | | |
| F | Частота | T | Время |

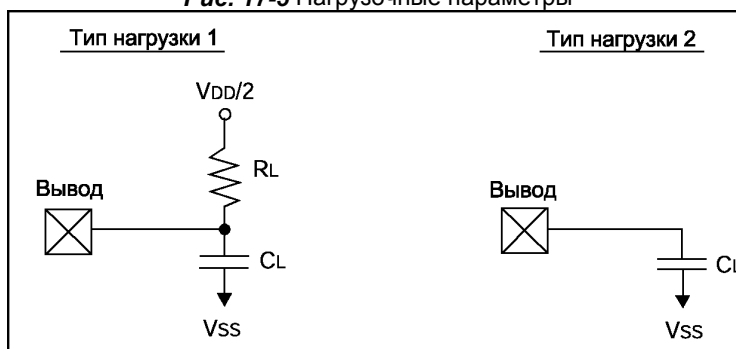
Строчные символы (pp) и их значение

| | | | |
|-----------|--------------------|-----|-------|
| pp | | | |
| ck | CLKOUT | osc | OSC1 |
| io | Канал ввода/вывода | t0 | T0CKI |
| mc | -MCLR | | |

Прописные символы и их значение

| | | | |
|----------|--------------------------|---|----------------|
| S | | | |
| F | Задний фронт | P | Период |
| H | Высокий уровень | R | Передний фронт |
| I | Неверный (3-е состояние) | V | Верный |
| L | Низкий уровень | Z | 3-е состояние |

Рис. 17-5 Нагрузочные параметры



$R_L = 464 \text{ Ом}$
 $C_L = 50 \text{ пФ}$ (для всех выводов, кроме OSC2)
 $C_L = 15 \text{ пФ}$ (для вывода OSC2)

Таблица 17-3 Электрические параметры PIC16F62X, PIC16LF62X

| Стандартные рабочие условия (если не указано иное) | | | | | | | |
|--|-------|--|-----------|--------|-------|-----|------------|
| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
| D120 | Ed | EEPROM память данных | | | | | |
| D121 | Vdrw | Число циклов стирание/запись | 1M* | 10M | - | C/3 | 5B @ 25°C |
| D122 | Tdew | Напряжение питания для записи/чтения | V_{MIN} | - | 5.5 | В | |
| | | Время цикла стирание/запись | - | 4 | 8* | мс | |
| D130 | Ep | FLASH память программ | | | | | |
| D131 | Vpr | Число циклов стирание/запись | 1000* | 10^5 | - | | |
| D132 | Vprew | Напряжение питания для чтения | V_{MIN} | - | 5.5 | | |
| D133 | Trew | Напряжение питания для стирания/записи | 4.5 | - | 5.5 | | |
| | | Время цикла стирание/запись | - | 4 | 8* | | |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при $V_{DD}=5.0\text{В}$ @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

17.5 Временные диаграммы и спецификации

Рис. 17-6 Временная диаграмма внешнего тактового сигнала

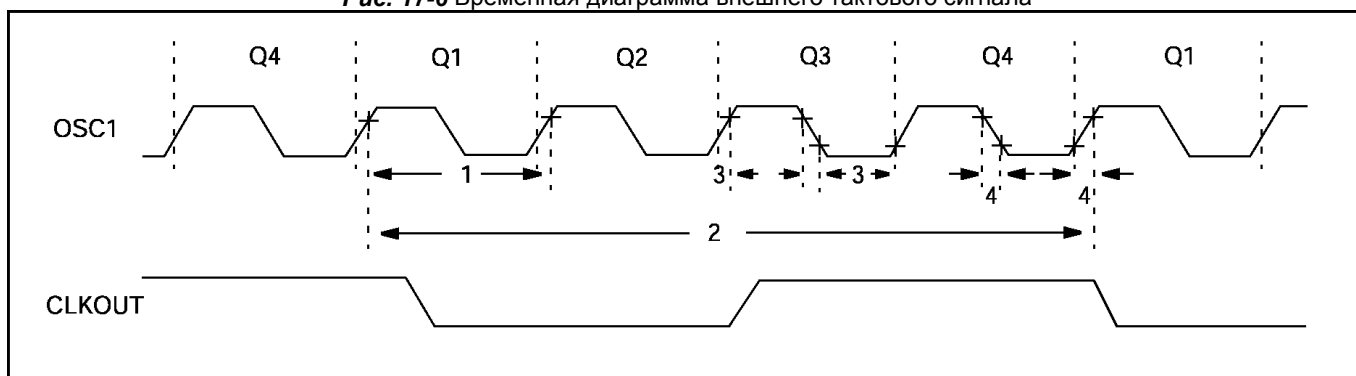


Таблица 17-4 Параметры внешнего тактового сигнала

| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание | |
|--------|--|---|-------|-----------------|-------|-----------------------|--------------------------------------|----------|
| | F _{osc} | Частота внешнего тактового сигнала ⁽¹⁾ | DC | - | 4 | МГц | XT, ER режим V _{DD} =5В | |
| | | | DC | - | 20 | МГц | | HS режим |
| | | | Dc | - | 200 | кГц | | LP режим |
| | Частота генератора ⁽¹⁾ | | 0.1 | - | 4 | МГц | ER режим V _{DD} =5В | |
| | | | 1 | - | 4 | МГц | XT режим | |
| | | | | - | 20 | МГц | HS режим | |
| | | | | - | 200 | кГц | LP режим | |
| | | | | 4 | МГц | INTRC режим (быстрый) | | |
| 1 | T _{osc} | Период внешнего тактового сигнала ⁽¹⁾ | 250 | - | - | нс | XT, ER режим | |
| | | | 50 | - | - | нс | HS режим | |
| | | | 5 | - | - | мкс | LP режим | |
| | | | 250 | - | - | нс | ER режим | |
| | | | 250 | - | 10000 | нс | XT режим | |
| | Период генератора ⁽¹⁾ | | 50 | - | 1000 | нс | HS режим | |
| | | | 5 | - | | мкс | LP режим | |
| | | | | 250 | | нс | INTRC режим (быстрый) | |
| | | | | 27 | | мкс | INTRC режим (медл.) | |
| 2 | T _{cy} | Время выполнения инструкции ⁽¹⁾ | 1.0 | T _{cy} | DC | нс | T _{cy} = 4/F _{osc} | |
| 3 | T _{OSL} , T _{OSH} | Длительность высокого/низкого уровня CLKIN (OSC1) | 100* | - | - | нс | XT режим | |
| 4 | INTRC | Внутренняя калибровка ER | 3.65 | 4.00 | 4.28 | МГц | V _{DD} = 5В | |
| 5 | ER | Внешняя установка частоты ER | 10кГц | | 8МГц | | V _{DD} = 5В | |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при V_{DD}=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

Примечание 1. Машинный цикл микроконтроллера равняется 4 периодам тактового сигнала. Все приведенные значения основываются на характеристиках конкретного типа генератора в стандартных условиях при выполнении программы. Выход за указанные пределы может привести к нестабильной работе генератора и/или к большому потребляемому току.

Рис. 17-7 Временная диаграмма CLKOUT и каналов ввода/вывода

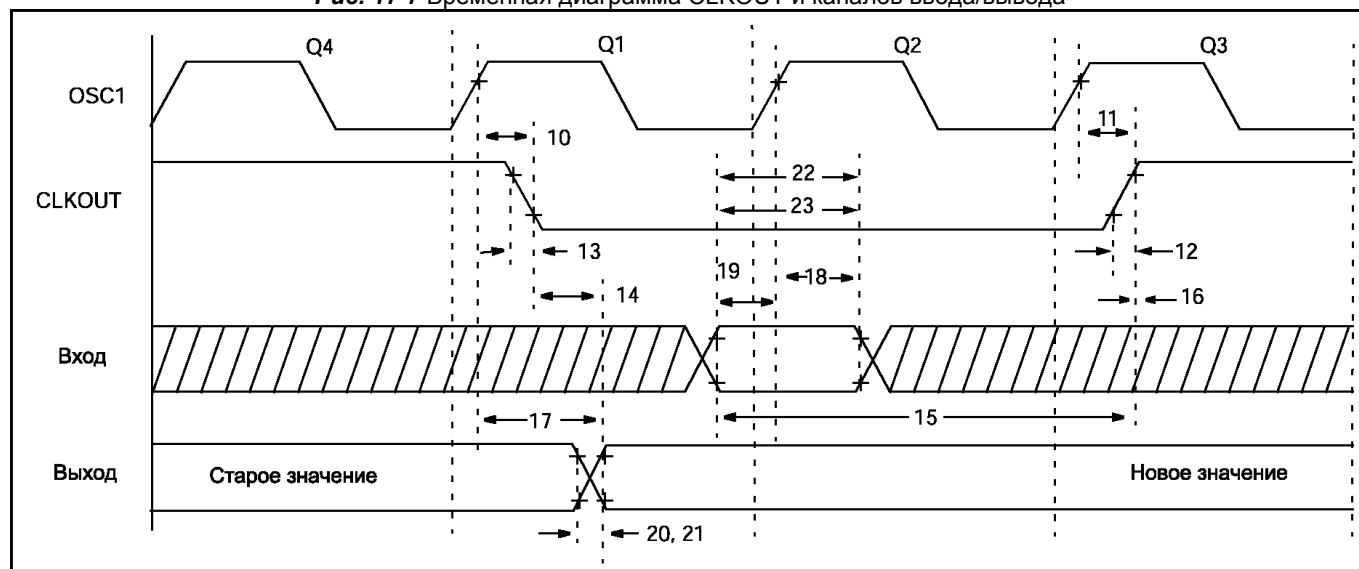


Таблица 17-5 Параметры CLKOUT и каналов ввода/вывода

| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
|-----------|----------|---|--|---------|-------------|----------|------------|
| 10 10A | TosH2ckL | От OSC1 ↑ до CLKOUT ↓ 16F62X 16LF62X | - - | 75 - | 200 400 | нс нс | |
| 11 11A | TosH2ckH | От OSC1 ↑ до CLKOUT ↑ 16F62X 16LF62X | - - | 75 - | 200 400 | нс нс | |
| 12 12A | TckR | CLKOUT длит. переднего фронта 16F62X 16LF62X | - - | 35 - | 100 200 | нс нс | |
| 13 13A | TckF | CLKOUT длит. заднего фронта 16F62X 16LF62X | - - | 35 - | 100 200 | нс нс | |
| 14 | TckL2ioV | От CLKOUT ↓ до установл. выхода | - | - | 20 | нс | |
| 15 | TioV2ckH | От установл. входа до CLKOUT ↑ 16F62X 16LF62X | $T_{OSC}+200\text{нс}$ $T_{OSC}+400\text{нс}$ | - - | - - | нс нс | |
| 16 | TckH2iol | Удержание входа после CLKOUT ↑ | 0 | - | - | нс | |
| 17 | TosH2ioV | От OSC1 ↑ до установл. выхода 16F62X 16LF62X | - - | 50 - | 150* 300 | нс нс | |
| 18 | TosH2iol | Удержание входа после OSC1 ↑ | 200 | - | - | нс | |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при $V_{DD}=5.0\text{В}$ @ 25°C , если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

Рис. 17-8 Временная диаграмма сброса, WDT, OST, PWRT

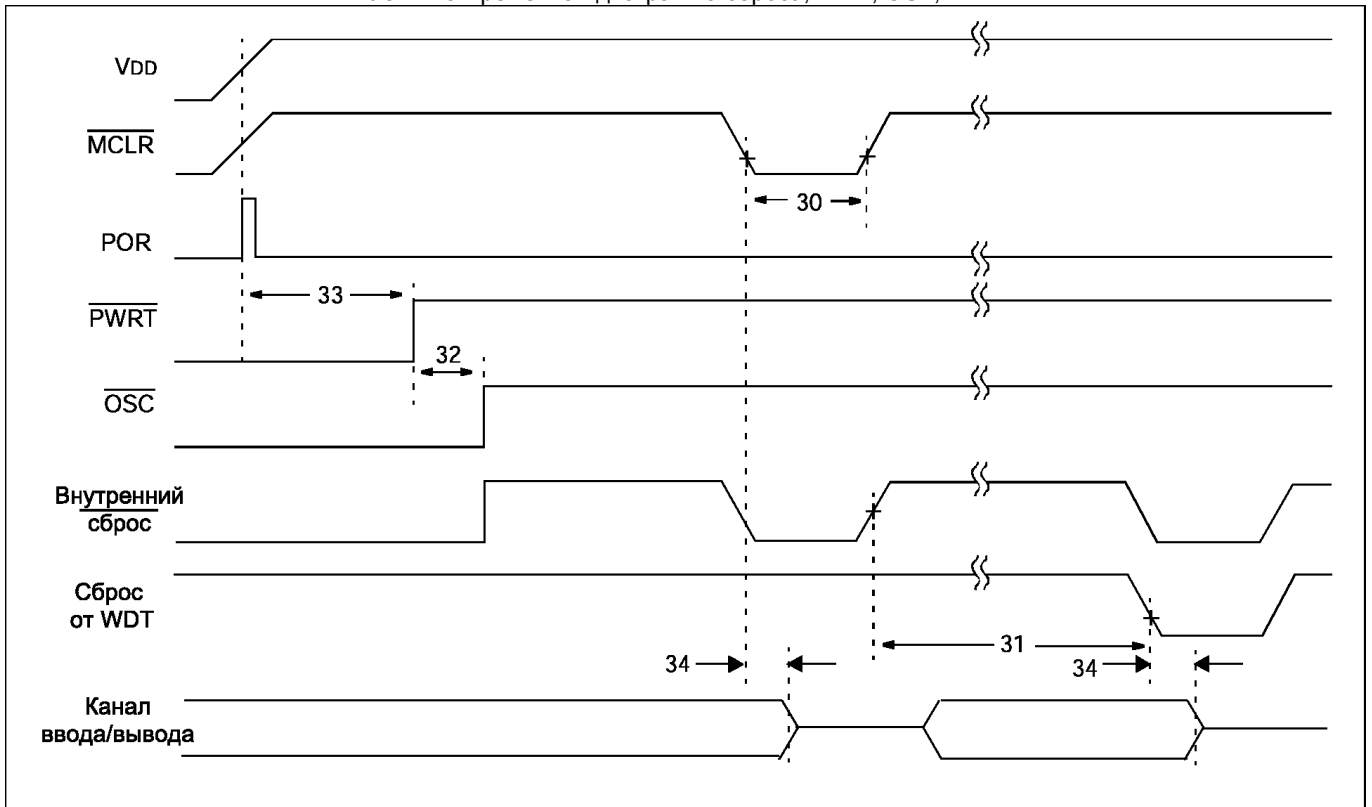


Рис. 17-9 Временная диаграмма работы BOD

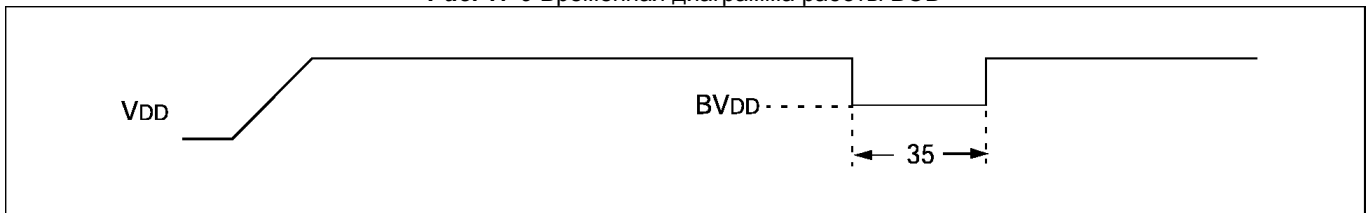


Таблица 17-6 Параметры сброса, WDT, OST, PWRT, BOD

| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
|--------|-------------------|---|-------------|----------------------|------------|----------|--|
| 30 | T _{MCLR} | Длительность импульса -MCLR | 2000 TBD | - TBD | - TBD | нс мс | V _{DD} =5В, -40°C до +85°C Расшир.темпер.диап. |
| 31 | T _{WDT} | Период переполнения WDT (без делителя) | 7 TBD | 18 TBD | 33 TBD | мс мс | V _{DD} =5В, -40°C до +85°C Расшир.темпер.диап. |
| 32 | T _{OST} | Период OST | - | 1024T _{OSC} | - | - | T _{OSC} = периоду OSC1 |
| 33* | T _{PWRT} | Период PWRT | 28 TBD | 72 TBD | 132 TBD | мс мс | V _{DD} =5В, -40°C до +85°C |
| 34 | T _{IOZ} | От сброса -MCLR или WDT до перевода каналов ввода/вывода 3-е состояние | - | - | 2.0 | мкс | |
| 35 | T _{BOD} | Длительность импульса BOD | 100 | - | - | мкс | V _{DD} ≤ BV _{DD} (D005) |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при V_{DD}=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

Рис. 17-10 Временная диаграмма внешнего тактового сигнала для TMR0 и TMR1

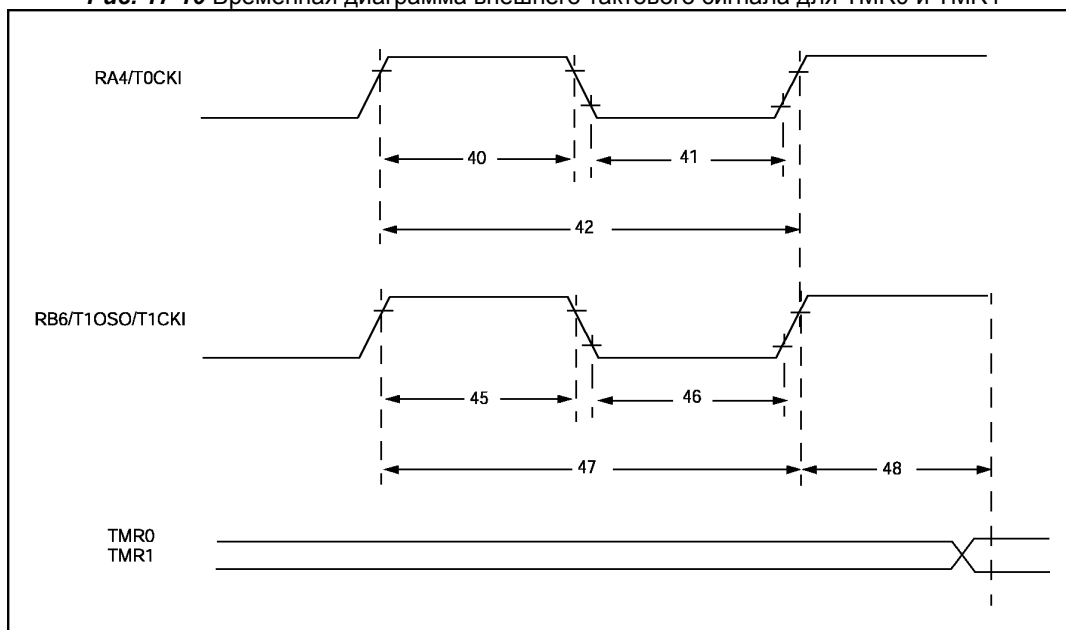


Таблица 17-7 Параметры внешнего тактового сигнала для TMR0 и TMR1

| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
|--------|-------|---|---|-----------------------|-----------------------|----------------------------|------------------|
| 40* | Tt0H | Длительность высокого уровня T0CKI Без делителя С делителем | $0.5T_{CY}+20$ 10 | - - | - - | нс нс | |
| 41* | Tt0L | Длительность низкого уровня T0CKI Без делителя С делителем | $0.5T_{CY}+20$ 10 | - - | - - | нс нс | |
| 42* | Tt0P | Период T0CKI | $(T_{CY}+40)/N$ | - | - | нс | N = коэфф.предд. |
| 45* | Tt1H | Длительность высокого уровня T1CKI Синхр.реж. без делителя Синхр.реж. с делел. 16F62X Синхр.реж. с делел. 16LF62X Асинхронный режим 16F62X Асинхронный режим 16LF62X | $0.5T_{CY}+20$ 15 25 30 50 | - - - - - | - - - - - | нс нс нс нс нс | |
| 46* | Tt1L | Длительность низкого уровня T1CKI Синхр.реж. без делителя Синхр.реж. с делел. 16F62X Синхр.реж. с делел. 16LF62X Асинхронный режим 16F62X Асинхронный режим 16LF62X | $0.5T_{CY}+20$ 15 25 30 50 | - - - - - | - - - - - | нс нс нс нс нс | |
| 47* | Tt1P | Период T1CKI Синхронный режим 16F62 Синхронный режим 16LF62 Асинхронный режим 16F62X Асинхронный режим 16LF62X | $(T_{CY}+40)/N$ $(T_{CY}+40)/N$ 60 100 | - - - - | - - - - | нс нс нс нс | N = коэфф.предд. |
| | Ft1 | Частота резонатора для TMR1 (T1OSCEN=1) | DC | - | 200 | кГц | |
| 48 | TCKE1 | Задержка от активного фронта тактового сигнала до приращения TMR1 | $2T_{osc}$ | - | $7T_{osc}$ | - | |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при $V_{DD}=5.0V$ @ $25^{\circ}C$, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

Рис. 17-11 Временная диаграмма захват/сравнение/ШИМ

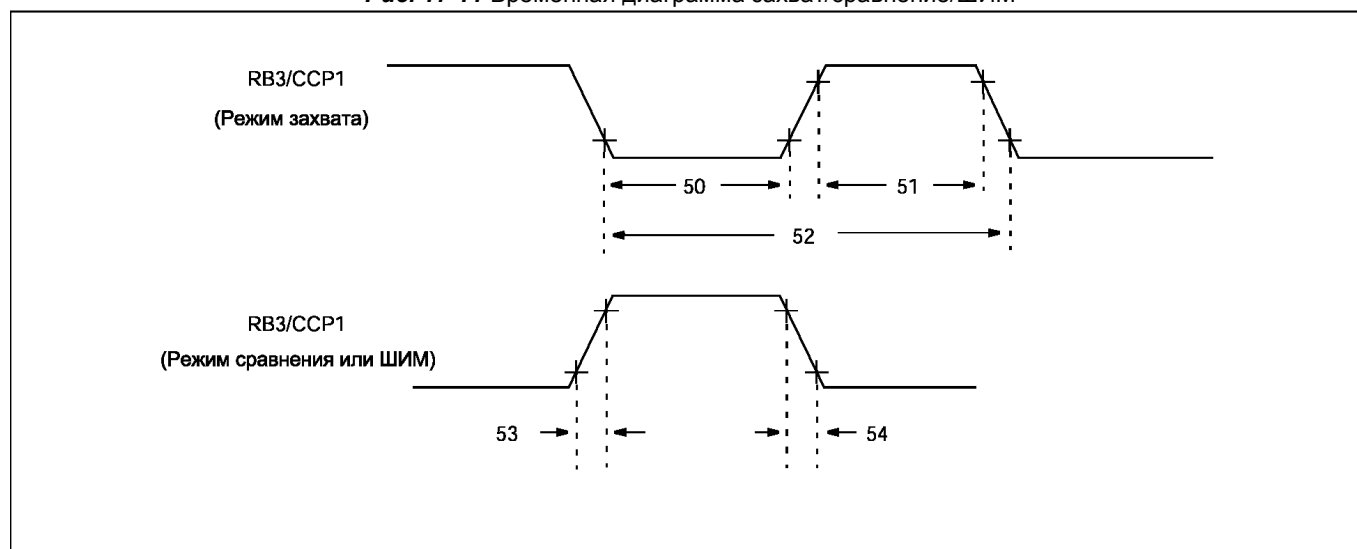


Таблица 17-8 Параметры захват/сравнение/ШИМ

| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
|--------|-------|---|------------------------------------|-------------|-------------|----------------|------------------|
| 50* | TccL | Сигнал низкого уровня CCP Без предделителя С предделителем 16F62X С предделителем 16LF62X | 0.5T _{CY} +20 10 20 | - - - | - - - | нс нс нс | |
| 51* | TccH | Сигнал высокого уровня CCP Без предделителя С предделителем 16F62X С предделителем 16LF62X | 0.5T _{CY} +20 10 20 | - - - | - - - | нс нс нс | |
| 52* | TccP | Период входного сигнала CCP | (3T _{CY} +40)/N | - | - | нс | N = коэфф.предд. |
| 53* | TccR | Время установление высокого уровня сигнала на выходе CCP 16F62X 16LF62X | | 10 25 | 25 45 | нс нс | |
| 54* | TccF | Время установление низкого уровня сигнала на выходе CCP 16F62X 16LF62X | | 10 25 | 25 45 | нс нс | |

* - Эти параметры определены, но не протестированы.

** - В столбце "Тип." приведены параметры при V_{DD}=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

Рис. 17-12 Временная диаграмма внешнего тактового сигнала TMR0

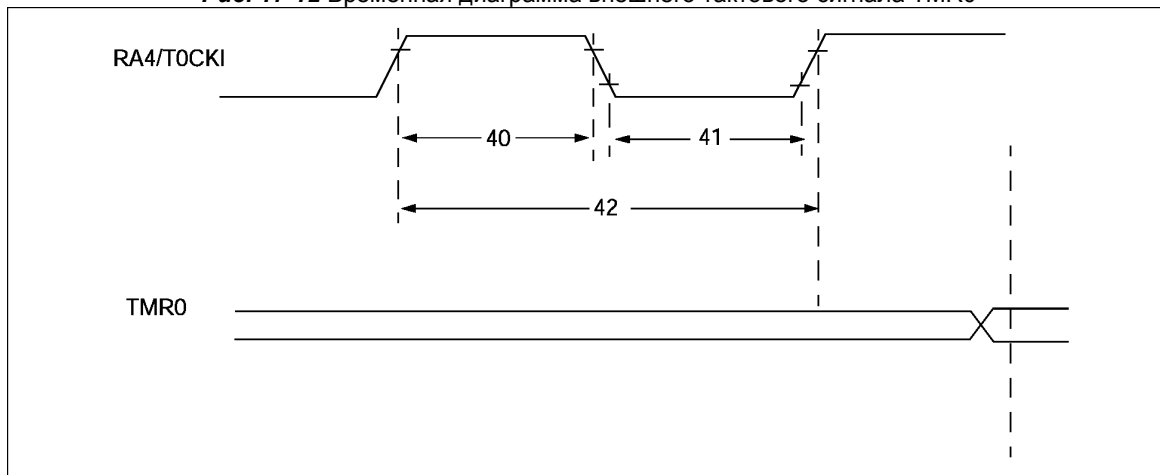


Таблица 17-9 Параметры внешнего тактового сигнала TMR0

| № пар. | Обоз. | Описание | Мин. | Тип.** | Макс. | Ед. | Примечание |
|--------|-------|------------------------------------|--------------------------|--------|-------|-----|------------------|
| 40 | Tt0H | Длительность высокого уровня T0CKI | 0.5T _{CY} +20* | - | - | нс | |
| | | С предделителем | 10 | - | - | нс | |
| 41 | Tt0L | Длительность низкого уровня T0CKI | 0.5T _{CY} +20* | - | - | нс | |
| | | С предделителем | 10 | - | - | нс | |
| 42 | Tt0P | Период T0CKI | (T _{CY} +40*)/N | - | - | нс | N = коэфф.предд. |

* - Эти параметры определены, но не протестированы.

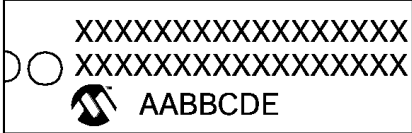
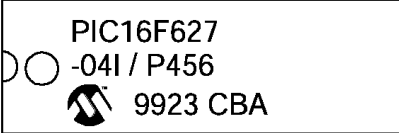
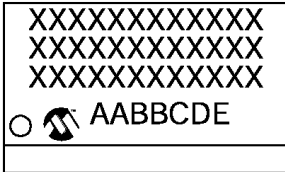
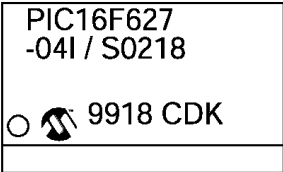
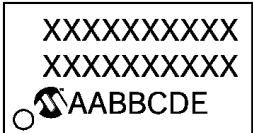
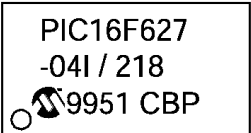
** - В столбце "Тип." приведены параметры при V_{DD}=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

18.0 Характеристики микроконтроллеров

На момент выполнения перевода данный раздел отсутствовал в оригинальной технической документации.

19.0 Корпуса микроконтроллеров

19.1 Описание обозначений на корпусах микроконтроллеров

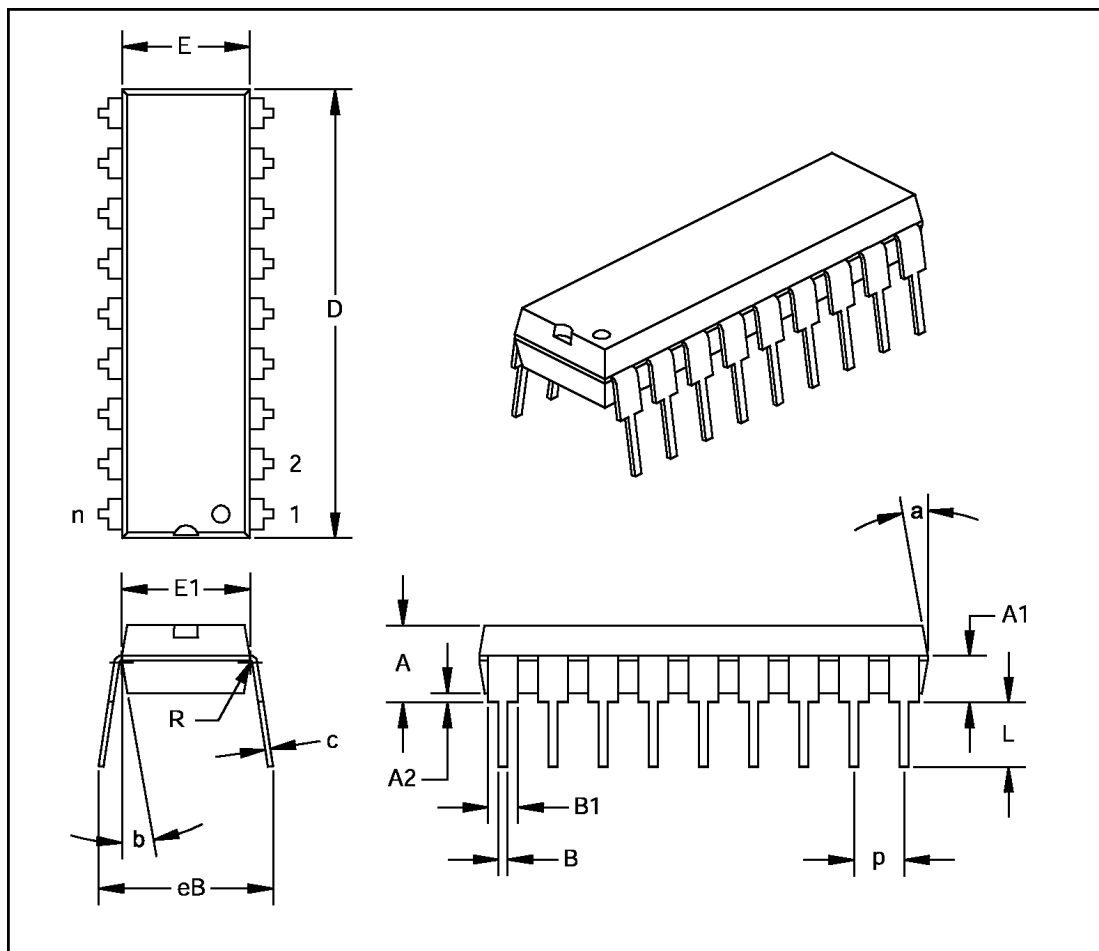
| | |
|---|--|
| <p>18-выводный PDIP</p>  | <p>Пример</p>  |
| <p>18-выводный SOIC</p>  | <p>Пример</p>  |
| <p>20-выводный SSOP</p>  | <p>Пример</p>  |

Обозначения:

| | |
|--|---|
| XX..X | Тип микроконтроллера* |
| AA | Две цифры даты изготовления |
| BB | Две цифры номера недели изготовления считая с 1 января. |
| C | Код завода изготовителя |
| | O = внешний представитель |
| | C = 5" |
| | S = 6" |
| | H = 8" |
| D | Номер версии |
| E | Код завода или страны, в которой кристалл был упакован в корпус |
| Примечание. Если тип микроконтроллера не помещается в одну строку, то он будет перемещен на другую строку, ограничивая число доступных символов для информации заказчика. | |

* Стандартная маркировка OTP микросхем состоит из: типа микроконтроллера, код года, код недели, код завода изготовителя, код упаковщика кристалла в корпус. Изменение маркировки микросхемы выполняется за отдельную плату. Для QTP микроконтроллеров стоимость маркировки входит в цену микросхем QTP.

Тип корпуса: 18-выводный PDIP K04-007

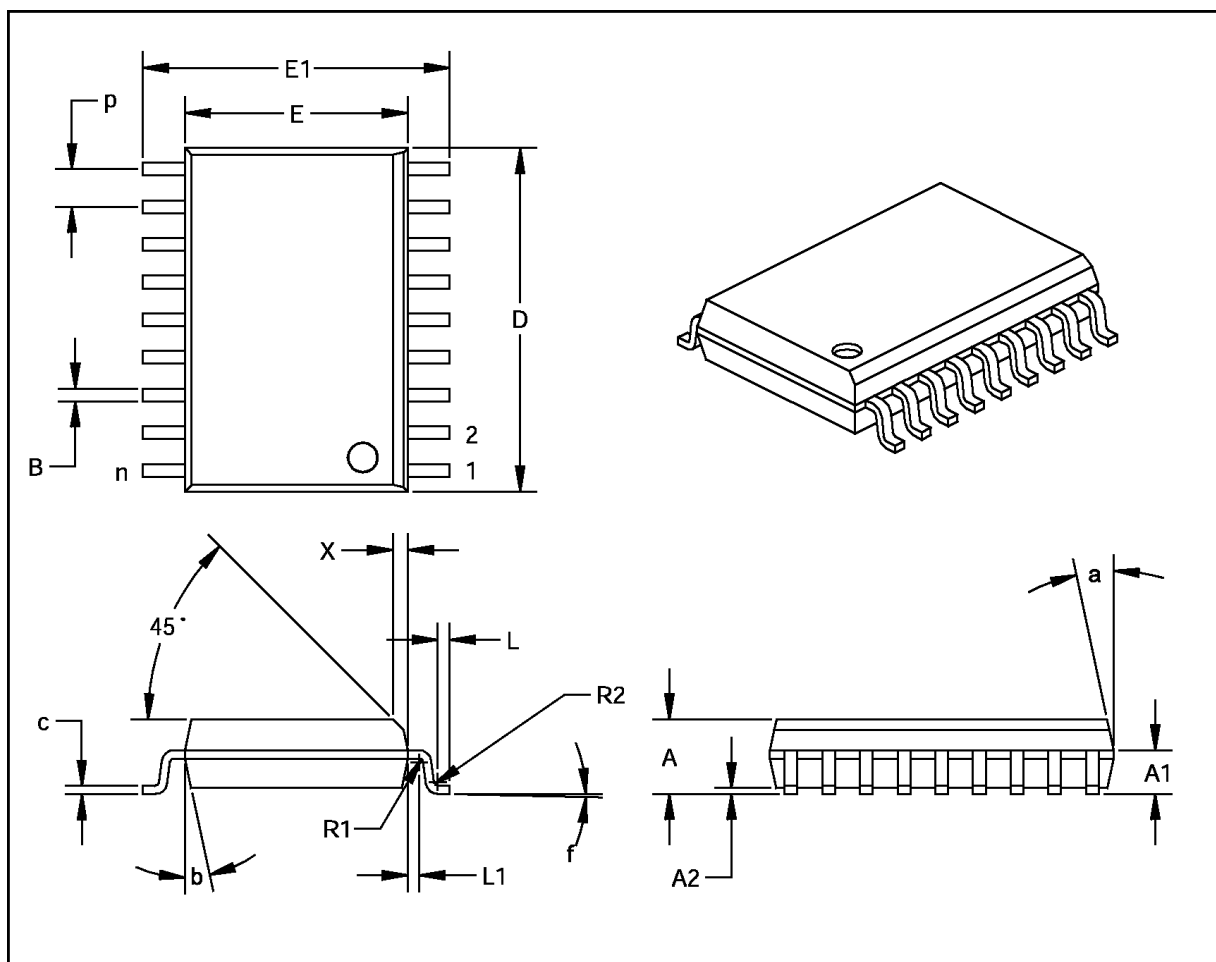


| Единицы измерения | | Дюймы* | | | Миллиметры | | |
|------------------------------------|----------|--------|-------|-------|------------|-------|-------|
| | | Мин. | Ном. | Макс. | Мин. | Ном. | Макс. |
| Пределы размеров | | | | | | | |
| Число выводов | n | | 18 | | | 18 | |
| Расстояние между выводами | p | | 0.100 | | | 2.54 | |
| Ширина нижней части вывода | B | 0.013 | 0.018 | 0.023 | 0.33 | 0.46 | 0.58 |
| Ширина верхней части вывода | B1** | 0.055 | 0.060 | 0.065 | 1.40 | 1.52 | 1.65 |
| Радиус сгиба вывода | R | 0.000 | 0.005 | 0.010 | 0.00 | 0.13 | 0.25 |
| Толщина вывода | c | 0.005 | 0.010 | 0.015 | 0.13 | 0.25 | 0.38 |
| Толщина корпуса | A | 0.110 | 0.155 | 0.155 | 2.79 | 3.94 | 3.94 |
| Толщина нижней части корпуса | A1 | 0.075 | 0.095 | 0.115 | 1.91 | 2.41 | 2.92 |
| Расстояние между корпусом и платой | A2 | 0.000 | 0.020 | 0.020 | 0.00 | 0.51 | 0.51 |
| Длина нижней части вывода | L | 0.125 | 0.130 | 0.135 | 3.18 | 3.30 | 3.43 |
| Длина корпуса | D*** | 0.890 | 0.895 | 0.900 | 22.61 | 22.73 | 22.86 |
| Ширина корпуса | E*** | 0.245 | 0.255 | 0.265 | 6.22 | 6.48 | 6.73 |
| Ширина корпуса без фаски | E1 | 0.230 | 0.250 | 0.270 | 5.84 | 6.35 | 6.86 |
| Полная ширина корпуса с выводами | eB | 0.310 | 0.349 | 0.387 | 7.87 | 8.85 | 9.83 |
| Угол фаски верхней части корпуса | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Угол фаски нижней части корпуса | β | 5 | 10 | 15 | 5 | 10 | 15 |

* Основные размеры.

** Параметр B1 не включает в себя возможные выступы. Выступ в сторону не должен превышать 0.003"(0.076мм) или не более 0.006"(0.152мм) параметра B1.

*** Параметры D и E не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм) или не более 0.020"(0.508мм) параметров D и E.

Тип корпуса: 18-выводный SOIC K04-051 - 300mil.

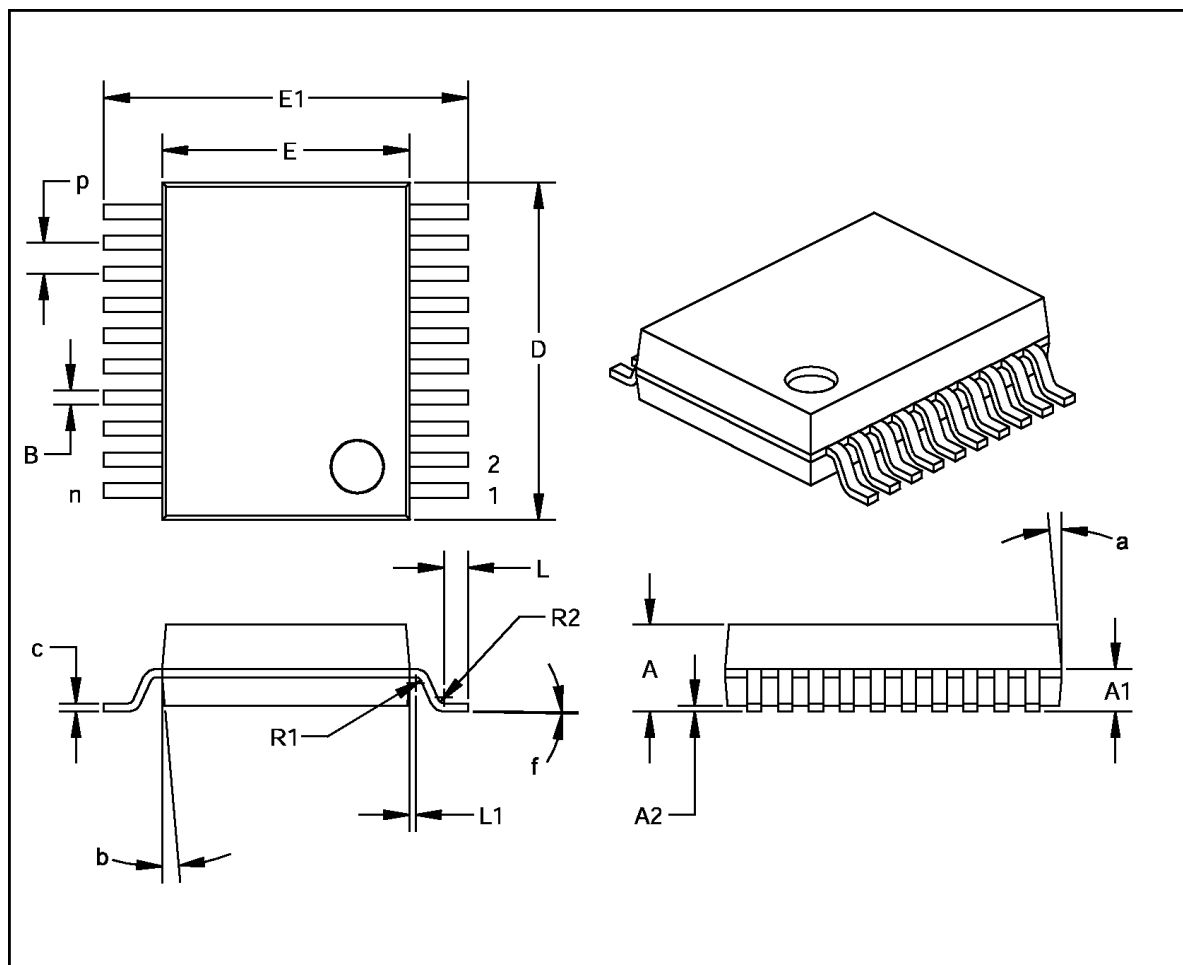
| Единицы измерения | | Дюймы* | | | Миллиметры | | |
|------------------------------------|------|--------|-------|-------|------------|-------|-------|
| | | Мин. | Ном. | Макс. | Мин. | Ном. | Макс. |
| Пределы размеров | | | | | | | |
| Число выводов | n | | 18 | | | 18 | |
| Расстояние между выводами | p | | 0.050 | | | 1.27 | |
| Толщина корпуса | A | 0.093 | 0.099 | 0.104 | 2.36 | 2.50 | 2.64 |
| Толщина нижней части корпуса | A1 | 0.048 | 0.058 | 0.068 | 1.22 | 1.47 | 1.73 |
| Расстояние между корпусом и платой | A2 | 0.004 | 0.008 | 0.011 | 0.10 | 0.19 | 0.28 |
| Длина корпуса | D*** | 0.450 | 0.456 | 0.462 | 11.43 | 11.58 | 11.73 |
| Ширина корпуса | E*** | 0.292 | 0.296 | 0.299 | 7.42 | 7.51 | 7.59 |
| Ширина корпуса с выводами | E1 | 0.394 | 0.407 | 0.419 | 10.01 | 10.33 | 10.64 |
| Размер ориентирующей фаски | X | 0.010 | 0.020 | 0.029 | 0.25 | 0.50 | 0.74 |
| Радиус изгиба верхней части вывода | R1 | 0.005 | 0.005 | 0.010 | 0.13 | 0.13 | 0.25 |
| Радиус изгиба нижней части вывода | R2 | 0.005 | 0.005 | 0.010 | 0.13 | 0.13 | 0.25 |
| Длина нижней части вывода | L | 0.011 | 0.016 | 0.021 | 0.28 | 0.41 | 0.53 |
| Угол наклона нижней части вывода | φ | 0 | 4 | 8 | 0 | 4 | 8 |
| Длина верхней части вывода | L1 | 0.010 | 0.015 | 0.020 | 0.25 | 0.38 | 0.51 |
| Толщина вывода | c | 0.009 | 0.011 | 0.012 | 0.23 | 0.27 | 0.30 |
| Ширина вывода | B** | 0.014 | 0.017 | 0.019 | 0.36 | 0.42 | 0.48 |
| Угол фаски верхней части корпуса | α | 0 | 12 | 15 | 0 | 12 | 15 |
| Угол фаски нижней части корпуса | β | 0 | 12 | 15 | 0 | 12 | 15 |

* Основные размеры.

** Параметр B не включает в себя возможные выступы. Выступ в сторону не должен превышать 0.003"(0.076мм) или не более 0.006"(0.152мм) параметра B.

*** Параметры D и E не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм) или не более 0.020"(0.508мм) параметров D и E.

Тип корпуса: 20-выводный SSOP K04-072 - 5.30mm.



| Единицы измерения | | Дюймы | | | Миллиметры* | | |
|------------------------------------|------|-------|-------|-------|-------------|------|-------|
| | | Мин. | Ном. | Макс. | Мин. | Ном. | Макс. |
| Пределы размеров | | | | | | | |
| Число выводов | n | | 20 | | | 20 | |
| Расстояние между выводами | p | | 0.026 | | | 0.65 | |
| Толщина корпуса | A | 0.068 | 0.073 | 0.078 | 1.73 | 1.86 | 1.99 |
| Толщина нижней части корпуса | A1 | 0.026 | 0.036 | 0.046 | 0.66 | 0.91 | 1.17 |
| Расстояние между корпусом и платой | A2 | 0.002 | 0.005 | 0.008 | 0.05 | 0.13 | 0.21 |
| Длина корпуса | D*** | 0.278 | 0.283 | 0.289 | 7.07 | 7.20 | 7.33 |
| Ширина корпуса | E*** | 0.205 | 0.208 | 0.212 | 5.20 | 5.29 | 5.38 |
| Ширина корпуса с выводами | E1 | 0.301 | 0.306 | 0.311 | 7.65 | 7.78 | 7.90 |
| Радиус изгиба верхней части вывода | R1 | 0.005 | 0.005 | 0.010 | 0.13 | 0.13 | 0.25 |
| Радиус изгиба нижней части вывода | R2 | 0.005 | 0.005 | 0.010 | 0.13 | 0.13 | 0.25 |
| Длина нижней части вывода | L | 0.015 | 0.020 | 0.025 | 0.38 | 0.51 | 0.64 |
| Угол наклона нижней части вывода | φ | 0 | 4 | 8 | 0 | 4 | 8 |
| Длина верхней части вывода | L1 | 0.000 | 0.005 | 0.010 | 0.00 | 0.13 | 0.25 |
| Толщина вывода | c | 0.005 | 0.007 | 0.009 | 0.13 | 0.18 | 0.22 |
| Ширина вывода | B** | 0.010 | 0.012 | 0.015 | 0.25 | 0.32 | 0.38 |
| Угол фаски верхней части корпуса | α | 0 | 5 | 10 | 0 | 5 | 10 |
| Угол фаски нижней части корпуса | β | 0 | 5 | 10 | 0 | 5 | 10 |

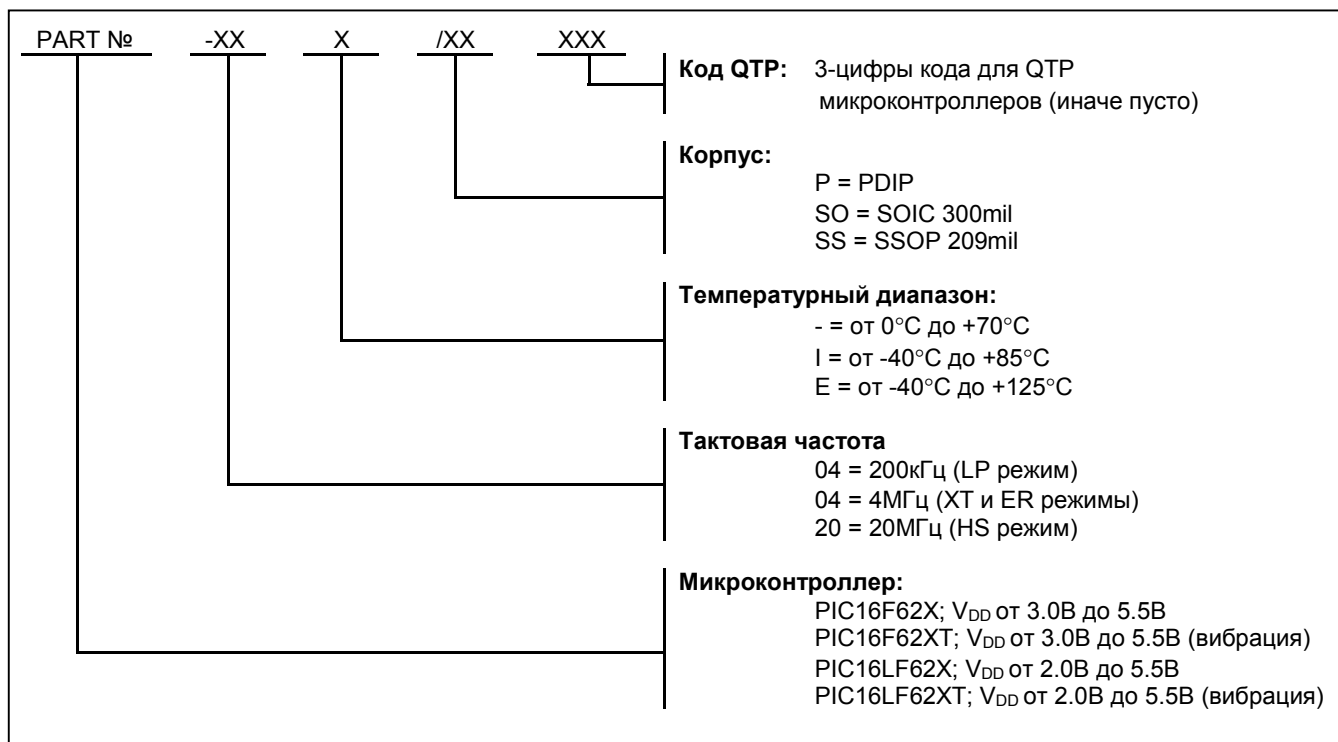
* Основные размеры.

** Параметр B не включает в себя возможные выступы. Выступ в сторону не должен превышать 0.003"(0.076мм) или не более 0.006"(0.152мм) параметра B.

*** Параметры D и E не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм) или не более 0.020"(0.508мм) параметров D и E.

19.2 Правила идентификации типа микроконтроллеров PIC16F62X

Чтобы определить параметры микроконтроллеров воспользуйтесь ниже описанным правилом.



Пример 19-1

PIC16F627-04/P 301 = коммерческий температурный диапазон, корпус PDIP, 4МГц, нормальный диапазон напряжения питания, код QTP 301.

Пример 19-2

PIC16LF627-04I/SO = промышленный температурный диапазон, корпус SOIC, 200кГц, расширенный диапазон напряжения питания.

Уважаемые господа!

ООО «Микро-Чип» поставляет полную номенклатуру комплектующих фирмы **Microchip Technology Inc** и осуществляет качественную техническую поддержку на русском языке.

С техническими вопросами Вы можете обращаться по адресу support@microchip.ru

По вопросам поставок комплектующих Вы можете обращаться к нам по телефонам:

(095) 963-9601

(095) 737-7545

и адресу sales@microchip.ru

На сайте

www.microchip.ru

Вы можете узнать последние новости нашей фирмы, найти техническую документацию и информацию по наличию комплектующих на складе.