

# PIC16F87X

Однокристальные 8-разрядные  
FLASH CMOS микроконтроллеры  
компании Microchip Technology Incorporated

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Перевод основывается на технической документации DS30292C  
компании Microchip Technology Incorporated, USA.

© ООО «Микро-Чип»  
Москва - 2002

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

# PIC16F87X

## 28/40-Pin 8-Bit CMOS FLASH Microcontrollers

**Trademarks:** The Microchip name, logo, PIC, PICmicro, PICMASTER, PIC-START, PRO MATE, KEELOQ, SEEVAL, MPLAB and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Total Endurance, ICSP, In-Circuit Serial Programming, Filter-Lab, MXDEV, microID, *FlexROM*, *fuzzyLAB*, MPASM, MPLINK, MPLIB, PICDEM, ICEPIC, Migratable Memory, FanSense, ECONOMONITOR and SelectMode are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Term Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

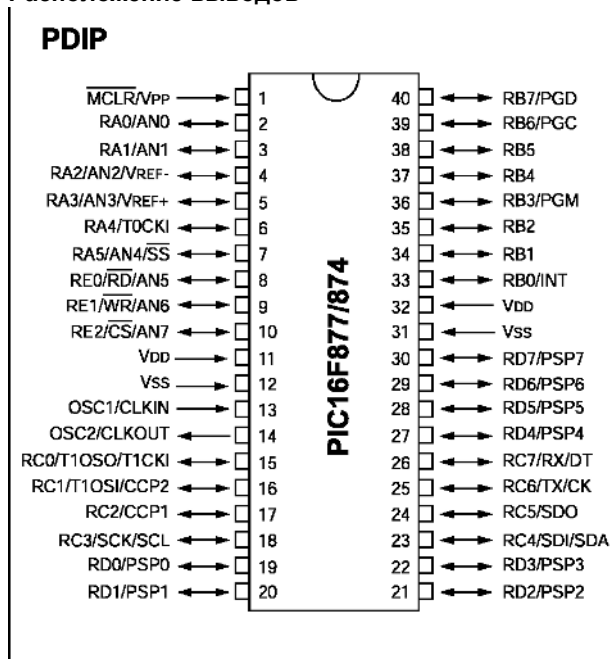
All other trademarks mentioned herein are property of their respective companies.

**Микроконтроллеры, описываемые в этом документе:**

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

**Характеристика микроконтроллеров:**

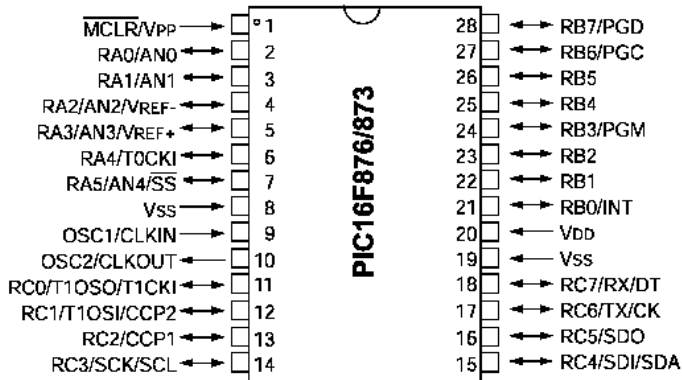
- Высокоскоростная RISC архитектура
- 35 инструкций
- Все команды выполняются за один цикл, кроме инструкций переходов, выполняемых за два цикла
- Тактовая частота:
  - DC - 20МГц, тактовый сигнал
  - DC - 200нс, один машинный цикл
- До 8к x 14 слов FLASH памяти программ  
До 368 x 8 байт памяти данных (ОЗУ)  
До 256 x 8 байт EEPROM памяти данных
- Совместимость по выводам с PIC16C73B/74B/76/77
- Система прерываний (до 14 источников)
- 8-уровневый аппаратный стек
- Прямой, косвенный и относительный режим адресации
- Сброс по включению питания (POR)
- Таймер сброса (PWRT) и таймер ожидания запуска генератора (OST) после включения питания
- Сторожевой таймер WDT с собственным RC генератором
- Программируемая защита памяти программ
- Режим энергосбережения SLEEP
- Выбор параметров тактового генератора
- Высокоскоростная, энергосберегающая CMOS FLASH/EEPROM технология
- Полностью статическая архитектура
- Программирование в готовом устройстве (используется два вывода микроконтроллера)
- Низковольтный режим программирования
- Режим внутрисхемной отладки (используется два вывода микроконтроллера)
- Широкий диапазон напряжений питания от 2.0В до 5.5В
- Повышенная нагрузочная способность портов ввода/вывода (25мА)
- Малое энергопотребление:
  - < 0.6 мА @ 3.0В, 4.0МГц
  - 20мкА @ 3.0В, 32кГц
  - < 1мкА в режиме энергосбережения

**Расположение выводов****Характеристика периферийных модулей:**

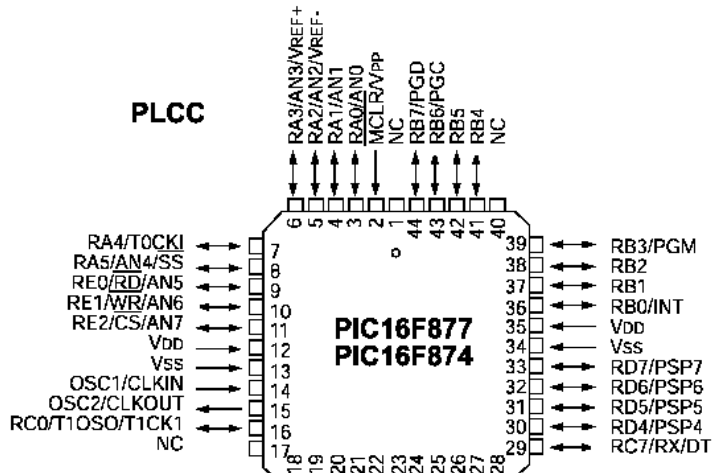
- Таймер 0: 8-разрядный таймер/счетчик с 8-разрядным программируемым предделителем
- Таймер 1: 16-разрядный таймер/счетчик с возможностью подключения внешнего резонатора
- Таймер 2: 8-разрядный таймер/счетчик с 8-разрядным программируемым предделителем и выходным делителем
- Два модуля сравнение/захват/ШИМ (CCP):
  - 16-разрядный захват (максимальная разрешающая способность 12.5нс)
  - 16-разрядное сравнение (максимальная разрешающая способность 200нс)
  - 10-разрядный ШИМ
- Многоканальное 10-разрядное АЦП
- Последовательный синхронный порт MSSP
  - ведущий/ведомый режим SPI
  - ведущий/ведомый режим I<sup>2</sup>C
- Последовательный синхронно-асинхронный приемопередатчик USART с поддержкой детектирования адреса
- Ведомый 8-разрядный параллельный порт PSP с поддержкой внешних сигналов -RD, -WR, -CS (только в 40/44-выводных микроконтроллерах)
- Детектор пониженного напряжения (BOD) для сброса по снижению напряжения питания (BOR)

**Расположение выводов**

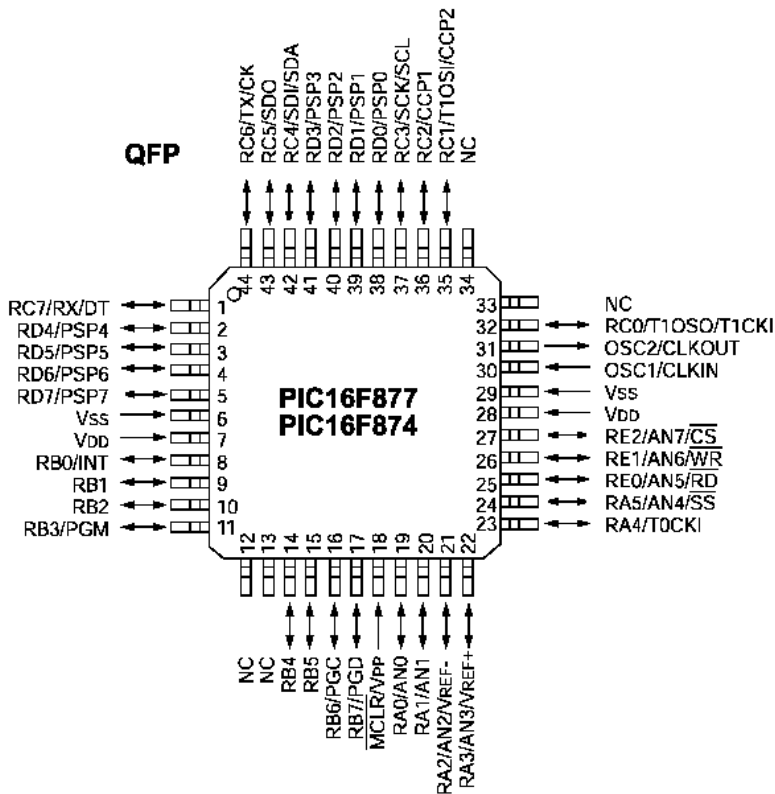
**PDIP, SOIC**



**PLCC**



**QFP**



Параметр	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Тактовая частота	DC - 20МГц	DC - 20МГц	DC - 20МГц	DC - 20МГц
Сброс (задержка сброса)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH память программ (14-разрядных слов)	4К	4К	8К	8К
Память данных (байт)	192	192	368	368
EEPROM память данных (байт)	128	128	256	256
Прерываний	13	14	13	14
Порты ввода/вывода	Порты А,В,С	Порты А,В,С,Д,Е	Порты А,В,С	Порты А,В,С,Д,Е
Таймеры	3	3	3	3
Модуль захват/сравнение/ШИМ	2	2	2	2
Модули последовательного интерфейса	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Модули параллельного интерфейса	-	PSP	-	PSP
Модуль 10-разрядного АЦП	5 каналов	8 каналов	5 каналов	8 каналов
Инструкций	35	35	35	35

## Содержание

<b>1.0 Введение .....</b>	<b>9</b>
<b>2.0 Организация памяти .....</b>	<b>14</b>
2.1 Организация памяти программ .....	14
2.2 Организация памяти данных .....	15
2.2.1 Регистры общего назначения .....	15
2.2.2 Регистры специального назначения .....	18
2.3 Регистры PCLATH и PCL .....	29
2.3.1 Вычисляемый переход .....	29
2.3.2 Стек .....	29
2.4 Страницы памяти программ .....	30
2.5 Косвенная адресация, регистры INDF и FSR .....	31
<b>3.0 Порты ввода/вывода .....</b>	<b>32</b>
3.1 Регистры PORTA и TRISA .....	32
3.2 Регистры PORTB и TRISB .....	34
3.3 Регистры PORTC и TRISC .....	36
3.4 Регистры PORTD и TRISD .....	38
3.5 Регистры PORTE и TRISE .....	39
3.6 Ведомый параллельный порт .....	41
<b>4.0 EEPROM память данных и FLASH память программ .....</b>	<b>43</b>
4.1 Регистры EECON1, EECON2 .....	43
4.2 Чтение из EEPROM памяти данных .....	45
4.3 Запись в EEPROM память данных .....	45
4.4 Чтение из FLASH памяти программ .....	46
4.5 Запись во FLASH память программ .....	46
4.6 Проверка записи .....	47
4.7 Защита от случайной записи .....	47
4.8 Операции при включенной защите кода .....	47
4.9 Защита записи FLASH памяти программ .....	47
<b>5.0 Модуль таймера TMR0 .....</b>	<b>49</b>
5.1 Прерывания от TMR0 .....	49
5.2 Использование внешнего источника тактового сигнала для TMR0 .....	50
5.3 Предделитель .....	50
<b>6.0 Модуль таймера TMR1 .....</b>	<b>51</b>
6.1 Работа TMR1 в режиме таймера .....	52
6.2 Работа TMR1 в режиме счетчика .....	52
6.3 Работа TMR1 в режиме синхронного счетчика .....	52
6.4 Работа TMR1 в режиме асинхронного счетчика .....	53
6.4.1 Чтение/запись TMR1 в асинхронном режиме .....	53
6.5 Генератор TMR1 .....	53
6.6 Сброс TMR1 триггером модуля CCP .....	53
6.7 Сброс регистров TMR1 (TMR1H, TMR1L) .....	54
6.8 Предделитель TMR1 .....	54
<b>7.0 Модуль таймера TMR2 .....</b>	<b>55</b>
7.1 Предделитель и выходной делитель TMR2 .....	56
7.2 Сигнал TMR2 .....	56

<b>8.0 ССР модуль (захват/сравнение/ШИМ)</b> .....	<b>57</b>
8.1 Режим захвата .....	59
8.1.1 <i>Настройка вывода модуля ССР</i> .....	59
8.1.2 <i>Настройка таймера TMR1</i> .....	59
8.1.3 <i>Обработка прерываний</i> .....	59
8.1.4 <i>Предварительный счетчик событий модуля ССР</i> .....	59
8.2 Режим сравнения .....	60
8.2.1 <i>Настройка вывода модуля ССР</i> .....	60
8.2.2 <i>Настройка таймера TMR1</i> .....	60
8.2.3 <i>Обработка прерываний</i> .....	60
8.2.4 <i>Триггер специального события</i> .....	60
8.3 Режим ШИМ .....	61
8.3.1 <i>Период ШИМ</i> .....	61
8.3.2 <i>Сквозность ШИМ</i> .....	62
8.3.3 <i>Последовательность настройки модуля ССР в ШИМ режиме</i> .....	62
<b>9.0 Модуль ведущего синхронного последовательного порта (MSSP)</b> .....	<b>64</b>
9.1 Режим SPI .....	68
9.1.1 <i>Режим ведущего SPI</i> .....	69
9.1.2 <i>Режим ведомого SPI</i> .....	70
9.2 Режим I <sup>2</sup> C .....	71
9.2.1 <i>Режим ведомого I<sup>2</sup>C</i> .....	72
9.2.2 <i>Поддержка общего вызова</i> .....	75
9.2.3 <i>Работа в SLEEP режиме</i> .....	75
9.2.4 <i>Эффект сброса</i> .....	75
9.2.5 <i>Режим ведущего I<sup>2</sup>C</i> .....	76
9.2.6 <i>Режим конкуренции</i> .....	76
9.2.7 <i>Поддержка режима ведущего I<sup>2</sup>C</i> .....	77
9.2.8 <i>Работа в режиме ведущего I<sup>2</sup>C</i> .....	77
9.2.8 <i>Генератор скорости обмена</i> .....	78
9.2.9 <i>Формирование бита START в режиме ведущего I<sup>2</sup>C</i> .....	79
9.2.10 <i>Формирование бита повторный START в режиме ведущего I<sup>2</sup>C</i> .....	80
9.2.11 <i>Передача данных в режиме ведущего I<sup>2</sup>C</i> .....	81
9.2.12 <i>Прием данных в режиме ведущего I<sup>2</sup>C</i> .....	83
9.2.13 <i>Формирование бита подтверждения в режиме ведущего I<sup>2</sup>C</i> .....	85
9.2.14 <i>Формирование бита STOP в режиме ведущего I<sup>2</sup>C</i> .....	85
9.2.15 <i>Синхронизация тактового сигнала</i> .....	86
9.2.16 <i>Режим конкуренции, арбитраж и конфликты шины</i> .....	86
9.3 Подключение к шине I <sup>2</sup> C .....	91
<b>10.0 Универсальный синхронно-асинхронный приемопередатчик (USART)</b> .....	<b>92</b>
10.1 Генератор частоты обмена USART BRG .....	94
10.1.1 <i>Выборка</i> .....	94
10.2 Асинхронный режим USART .....	96
10.2.1 <i>Асинхронный передатчик USART</i> .....	96
10.2.2 <i>Асинхронный приемник USART</i> .....	98
10.2.3 <i>Настройка 9-разрядного асинхронного приема с детектированием адреса</i> .....	100
10.3 Синхронный ведущий режим USART .....	102
10.3.1 <i>Передача синхронного ведущего</i> .....	102
10.3.2 <i>Прием синхронного ведущего</i> .....	104
10.4 Синхронный ведомый режим USART .....	105
10.4.1 <i>Передача синхронного ведомого</i> .....	105
10.4.2 <i>Прием синхронного ведомого</i> .....	106
<b>11.0 Модуль 10-разрядного АЦП</b> .....	<b>107</b>
11.1 <i>Временные требования к подключению канала АЦП</i> .....	110
11.2 <i>Выбор источника тактовых импульсов для АЦП</i> .....	111
11.3 <i>Настройка аналоговых входов</i> .....	111
11.4 <i>Аналого-цифровое преобразование</i> .....	112
11.4.1 <i>Выравнивание результата преобразования</i> .....	112
11.5 <i>Работа модуля АЦП в SLEEP режиме</i> .....	113
11.6 <i>Эффект сброса</i> .....	113

<b>12.0 Особенности микроконтроллеров PIC16F87X</b> .....	<b>114</b>
12.1 Биты конфигурации .....	114
12.2 Настройка тактового генератора.....	116
12.2.1 Режимы тактового генератора.....	116
12.2.2 Кварцевый/керамический резонатор.....	116
12.2.3 RC генератор.....	117
12.3 Сброс .....	118
12.4 Сброс по включению питания POR.....	119
12.5 Таймер включения питания PWRT .....	119
12.6 Таймер запуска генератора OST.....	119
12.7 Сброс по снижению напряжения питания BOR .....	119
12.8 Последовательность удержания микроконтроллера в состоянии сброса .....	119
12.9 Регистр PCON .....	119
12.10 Прерывания.....	124
12.10.1 Внешнее прерывание с входа RB0/INT.....	124
12.10.2 Прерывание по переполнению TMR0.....	125
12.10.3 Прерывание по изменению уровня сигнала на входах RB7:RB4.....	125
12.11 Сохранение контекста при обработке прерываний.....	125
12.12 Сторожевой таймер WDT .....	126
12.13 Режим энергосбережения SLEEP .....	127
12.13.1 Выход из режима SLEEP.....	127
12.13.2 Выход из режима SLEEP по прерыванию .....	127
12.14 Внутрисхемный отладчик ICD .....	128
12.15 Защита кода программы .....	128
12.16 Размещение идентификатора ID .....	128
12.17 Внутрисхемное программирование ICSP .....	129
12.18 Режим низковольтного программирования.....	129
<b>13.0 Система команд</b> .....	<b>130</b>
13.1 Подробное описание команд.....	132
<b>14.0 Поддержка разработчиков</b> .....	<b>139</b>
14.1 Интегрированная среда проектирования MPLAB-IDE .....	139
14.2 Ассемблер MPASM.....	140
14.3 С компиляторы MPLAB-C17 и MPLAB-C18 .....	140
14.4 Линкер MPLINK/ Организатор библиотек MPLIB .....	140
14.5 Программный симулятор MPLAB-SIM .....	140
14.6 Универсальный эмулятор MPLAB-ICE.....	140
14.7 ICEPIC.....	141
14.8 Внутрисхемный отладчик MPLAB-ICD .....	141
14.9 Универсальный программатор PRO MATE II .....	141
14.10 Программатор PICSTART .....	141
14.11 Демонстрационная плата PICDEM-1 .....	141
14.12 Демонстрационная плата PICDEM-2 для PIC16CXXX.....	141
14.13 Демонстрационная плата PICDEM-3 для PIC16CXXX.....	141
14.14 Демонстрационная плата PICDEM-17 .....	142
14.15 KeeLoq (с функциями программатора).....	142
<b>15.0 Электрические характеристики</b> .....	<b>144</b>
15.1 Электрические характеристики (Коммерческий, Промышленный).....	146
15.2 Электрические характеристики (Коммерческий, Промышленный).....	147
15.3 Электрические характеристики (Расширенный).....	148
15.4 Электрические характеристики (Расширенный).....	149
15.5 Символьное обозначение временных параметров .....	150
15.6 Временные диаграммы и спецификации .....	151
<b>16.0 Характеристики микроконтроллеров</b> .....	<b>164</b>
<b>17.0 Корпуса микроконтроллеров</b> .....	<b>175</b>
17.1 Описание обозначений на корпусах микроконтроллеров.....	175
17.2 Правила идентификации типа микроконтроллеров PIC16F87X .....	183



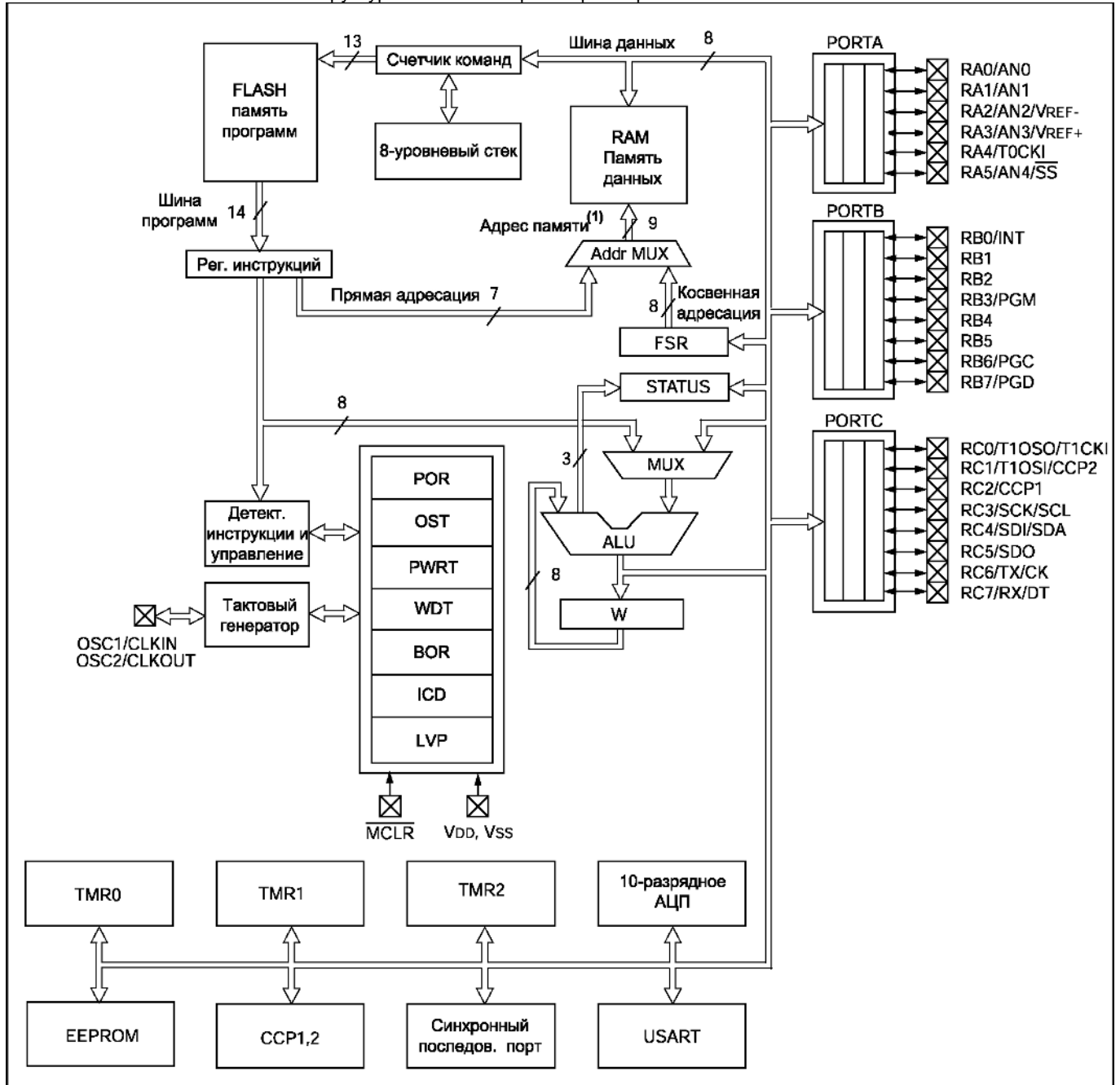
# 1.0 Введение

Дополнительную информацию по работе с микроконтроллерами PIC16F87X смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual". Файл Вы может получить у представителя Microchip Inc., на WEB узлах технической поддержки [www.microchip.com](http://www.microchip.com) и [www.microchip.ru](http://www.microchip.ru). Рекомендуется прочитать документацию "PICmicro™ Mid-Range Reference Manual" для лучшего понимания архитектуры микроконтроллеров и работы периферийных модулей.

В документации описывается работа четырех микроконтроллеров (PIC16F873, PIC16F874, PIC16F876 и PIC16F877). Микроконтроллеры PIC16F873/876 выпускаются в 28-выводном корпусе, а PIC16F874/877 в 40-выводном. В микроконтроллерах PIC16F873/876 ведомый параллельный порт не реализован.

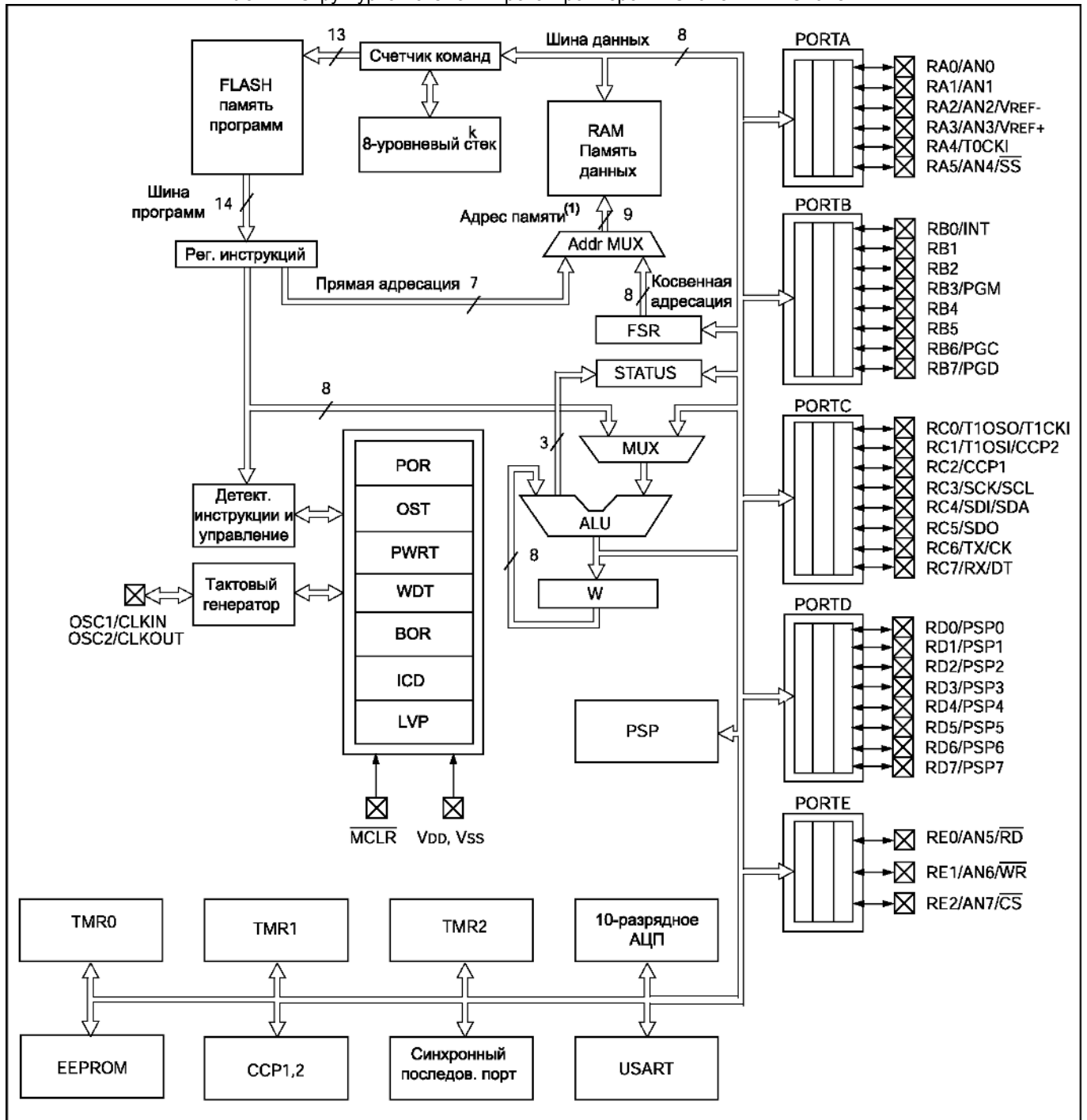
На рисунках 1-1, 1-2 показаны структурные схемы 28-выводных и 40-выводных микроконтроллеров, а таблицах 1-1, 1-2 представлен список выводов.

Рис. 1-1 Структурная схема микроконтроллеров PIC16F873 и PIC16F876



Примечание 1. Старшие биты адреса находятся в регистре STATUS.

Рис. 1-2 Структурная схема микроконтроллеров PIC16F874 и PIC16F877



Примечание 1. Старшие биты адреса находятся в регистре STATUS.

**Таблица 1-1** Назначение выводов микроконтроллеров PIC16F873 и PIC16F876

Обозначение вывода	№ вывода DIP	№ вывода SOIC	Тип I/O/P	Тип буфера	Описание
OSC1/CLKIN	9	9	I	ST/CMOS <sup>(3)</sup>	Вход генератора/ вход внешнего тактового сигнала
OSC2/CLKOUT	10	10	O	-	Выход генератора. Подключается кварцевый или керамический резонатор. В RC режиме тактового генератора на выходе OSC2 присутствует тактовый сигнал CLKOUT, равный F <sub>OSC</sub> /4.
-MCLR/V <sub>PP</sub>	1	1	I/P	ST	Вход сброса микроконтроллера или вход напряжения программирования. Сброс микроконтроллера происходит при низком логическом уровне сигнала на входе.
RA0/AN0	2	2	I/O	TTL	Двунаправленный порт ввода/вывода PORTA. RA0 может быть настроен как аналоговый канал 0 RA1 может быть настроен как аналоговый канал 1 RA2 может быть настроен как аналоговый канал 2 или вход отрицательного опорного напряжения RA3 может быть настроен как аналоговый канал 3 или вход положительного опорного напряжения RA4 может использоваться в качестве входа внешнего тактового сигнала для TMR0. Выход с открытым стоком. RA1 может быть настроен как аналоговый канал 1 или вход выбора микросхемы в режиме ведомого SPI
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/V <sub>REF-</sub>	4	4	I/O	TTL	
RA3/AN3/V <sub>REF+</sub>	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5-SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST <sup>(1)</sup>	
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST <sup>(2)</sup>	
RB7/PGD	28	28	I/O	TTL/ST <sup>(2)</sup>	
RC0/T1OSO/T1CKI	11	11	I/O	ST	Двунаправленный порт ввода/вывода PORTC. RC0 может использоваться в качестве выхода генератора TMR1 или входа внешнего тактового сигнала для TMR1. RC1 может использоваться в качестве входа генератора для TMR1 или вывода модуля CCP2. RC2 может использоваться в качестве вывода модуля CCP1. RC3 может использоваться в качестве входа/выхода тактового сигнала в режиме SPI и I <sup>2</sup> C. RC4 может использоваться в качестве входа данных в режиме SPI или вход/выход данных в режиме I <sup>2</sup> C. RC5 может использоваться в качестве выхода данных в режиме SPI. RC6 может использоваться в качестве вывода передатчика USART в асинхронном режиме или вывода синхронизации USART в синхронном режиме. RC6 может использоваться в качестве вывода приемника USART в асинхронном режиме или вывода данных USART в синхронном режиме.
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
V <sub>SS</sub>	8, 19	8, 19	P	-	Общий вывод для внутренней логики и портов ввода/вывода
V <sub>DD</sub>	20	20	P	-	Положительное напряжение питания для внутренней логики и портов ввода/вывода

Обозначения: I = вход, O = выход, I/O = вход/выход, P = питание, - = не используется,  
TTL = входной буфер 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 в режиме RC генератора и CMOS буфер в других режимах.

**Таблица 1-2** Назначение выводов микроконтроллеров PIC16F874 и PIC16F877

Обозначение вывода	№ выв. DIP	№ выв. PLCC	№ выв. QFP	Тип I/O/P	Тип буфера	Описание
OSC1/CLKIN	13	14	30	I	ST/CMOS <sup>(4)</sup>	Вход генератора / вход внешнего тактового сигнала
OSC2/CLKOUT	14	15	31	O	-	Выход генератора. Подключается кварцевый или керамический резонатор. В RC режиме тактового генератора на выходе OSC2 присутствует тактовый сигнал CLKOUT, равный $F_{osc}/4$ .
-MCLR/V <sub>PP</sub>	1	2	18	I/P	ST	Вход сброса микроконтроллера или вход напряжения программирования. Сброс микроконтроллера происходит при низком логическом уровне сигнала на входе.
RA0/AN0	2	3	19	I/O	TTL	Двухнаправленный порт ввода/вывода PORTA. RA0 может быть настроен как аналоговый канал 0 RA1 может быть настроен как аналоговый канал 1 RA2 может быть настроен как аналоговый канал 2 или вход отрицательного опорного напряжения RA3 может быть настроен как аналоговый канал 3 или вход положительного опорного напряжения RA4 может использоваться в качестве входа внешнего тактового сигнала для TMR0. Выход с открытым стоком. RA1 может быть настроен как аналоговый канал 1 или вход выбора микросхемы в режиме ведомого SPI
RA1/AN1	3	4	20	I/O	TTL	
RA2/AN2/V <sub>REF-</sub>	4	5	21	I/O	TTL	
RA3/AN3/V <sub>REF+</sub>	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	
RA5/-SS/AN4	7	8	24	I/O	TTL	
RB0/INT	33	36	8	I/O	TTL/ST <sup>(1)</sup>	
RB1	34	37	9	I/O	TTL	Двухнаправленный порт ввода/вывода PORTB. PORTB имеет программно подключаемые подтягивающие резисторы на входах. RB0 может использоваться в качестве входа внешних прерываний.  RB3 может использоваться в качестве входа для режима низковольтного программирования. Прерывания по изменению уровня входного сигнала. Прерывания по изменению уровня входного сигнала. Прерывания по изменению уровня входного сигнала или вывод для режима внутрисхемной отладки ICD. Тактовый вход в режиме программирования. Прерывания по изменению уровня входного сигнала или вывод для режима внутрисхемной отладки ICD. Вывод данных в режиме программирования.
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	
RB5	38	42	15	I/O	TTL	
RB6/PGC	39	43	16	I/O	TTL/ST <sup>(2)</sup>	
RB7/PGD	40	44	17	I/O	TTL/ST <sup>(2)</sup>	

Обозначения: I = вход, O = выход, I/O = вход/выход, P = питание, - = не используется,  
TTL = входной буфер 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 работе в режиме цифровых портов ввода/вывода. В режиме ведомого параллельного порта подключены входные буферы TTL (для совместимости с шиной микропроцессора).
4. Входной буфер с триггером Шмидта в режиме RC генератора и CMOS буфер в других режимах.

**Таблица 1-2** Назначение выводов микроконтроллеров PIC16F874 и PIC16F877 (продолжение)

Обозначение вывода	№ выв. DIP	№ выв. PLCC	№ выв. QFP	Тип I/O/P	Тип буфера	Описание
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	Двунаправленный порт ввода/вывода PORTC. RC0 может использоваться в качестве выхода генератора TMR1 или входа внешнего тактового сигнала для TMR1.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 может использоваться в качестве входа генератора для TMR1 или вывода модуля CCP2.
RC2/CCP1	17	19	36	I/O	ST	RC2 может использоваться в качестве вывода модуля CCP1.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 может использоваться в качестве входа/выхода тактового сигнала в режиме SPI и I <sup>2</sup> C.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 может использоваться в качестве входа данных в режиме SPI или вход/выход данных в режиме I <sup>2</sup> C.
RC5/SDO	24	26	43	I/O	ST	RC5 может использоваться в качестве выхода данных в режиме SPI.
RC6/TX/CK	25	27	44	I/O	ST	RC6 может использоваться в качестве вывода передатчика USART в асинхронном режиме или вывода синхронизации USART в синхронном режиме.
RC7/RX/DT	26	29	1	I/O	ST	RC6 может использоваться в качестве вывода приемника USART в асинхронном режиме или вывода данных USART в синхронном режиме.
RD0/PSP0 RD1/PSP1 RD2/PSP2 RD3/PSP3 RD4/PSP4 RD5/PSP5 RD6/PSP6 RD7/PSP7	19 20 21 22 27 28 29 30	21 22 23 24 30 31 32 33	38 39 40 41 2 3 4 5	I/O I/O I/O I/O I/O I/O I/O I/O	ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup>	Двунаправленный порт ввода/вывода PORTD или ведомый параллельный порт для подключения к шине микропроцессора
RE0-/RD/AN5 RE1-/WR/AN6 RE2-/CS/AN7	8 9 10	9 10 11	25 26 27	I/O I/O I/O	ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup> ST/TTL <sup>(3)</sup>	Двунаправленный порт ввода/вывода PORTE. RE0 может использоваться в качестве управляющего входа чтения PSP или аналогового канала 5. RE1 может использоваться в качестве управляющего входа записи PSP или аналогового канала 6. RE2 может использоваться в качестве управляющего входа выбора PSP или аналогового канала 7.
V <sub>SS</sub>	12,31	13,34	6, 29	P	-	Общий вывод для внутренней логики и портов ввода/вывода
V <sub>DD</sub>	11,32	12,35	7,28	P	-	Положительное напряжение питания для внутренней логики и портов ввода/вывода
NC	-	1,17, 28,40	12,13, 33,34	-	-	Эти выводы внутри микросхемы не подключены.

Обозначения: I = вход, O = выход, I/O = вход/выход, P = питание, - = не используется,  
TTL = входной буфер 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 работе в режиме цифровых портов ввода/вывода. В режиме ведомого параллельного порта подключены входные буферы TTL (для совместимости с шиной микропроцессора).
4. Входной буфер с триггером Шмидта в режиме RC генератора и CMOS буфер в других режимах.

## 2.0 Организация памяти

В микроконтроллерах PIC16F87X имеется три вида памяти. Память программ и память данных имеют отдельные шины данных и адреса, что позволяет выполнять параллельный доступ. Подробное описание работы с EEPROM памятью данных смотрите в разделе 4.0.

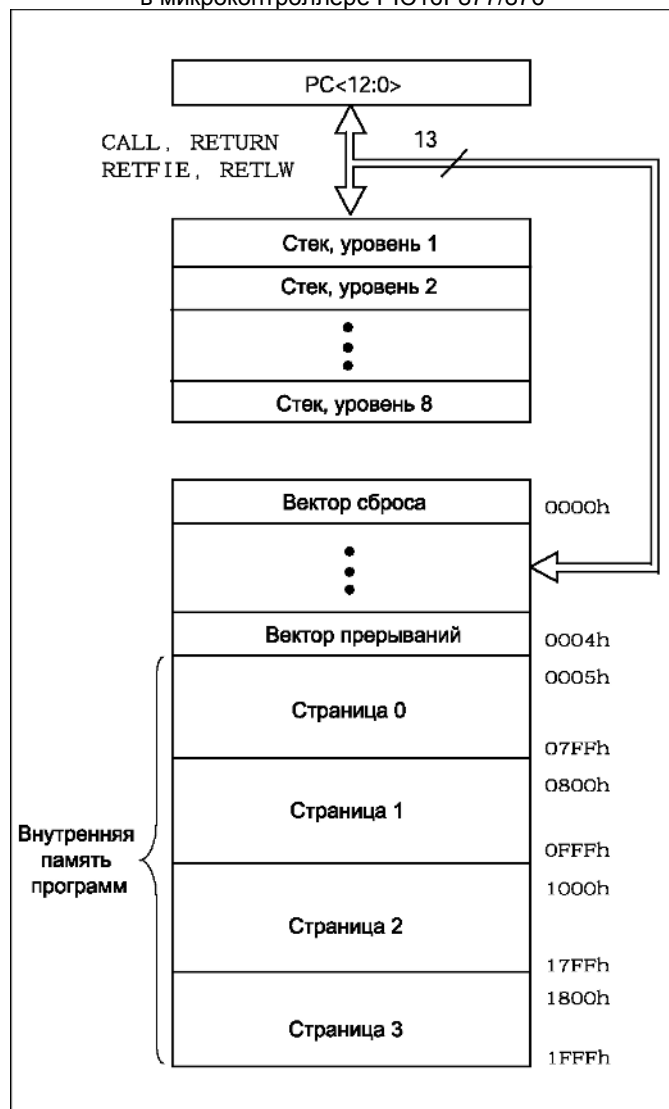
Дополнительную информацию по организации памяти смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual".

### 2.1 Организация памяти программ

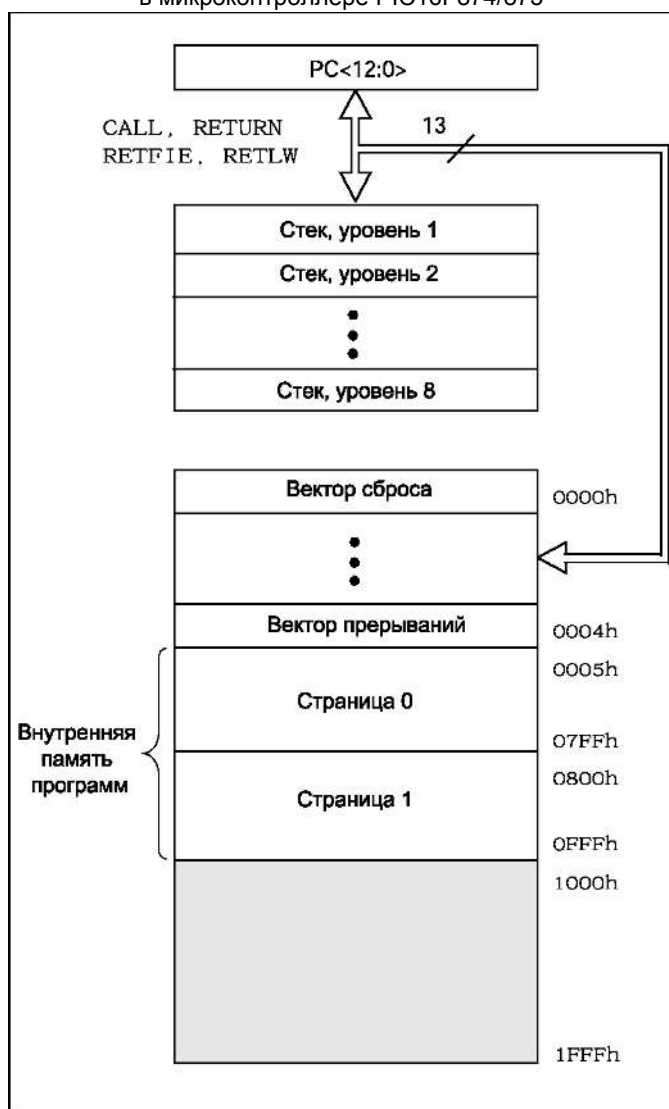
Микроконтроллеры PIC16F87X имеют 13-разрядный счетчик команд PC, способный адресовать 8К x 14 слов памяти программ. Физически реализовано FLASH памяти программ 8К x 14 в PIC16F877/876 и 4К x 14 в PIC16F873/874. Обращение к физически не реализованной памяти программ приведет к адресации реализованной памяти.

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

**Рис. 2-1** Организация памяти в микроконтроллере PIC16F877/876



**Рис. 2-2** Организация памяти в микроконтроллере PIC16F874/873



## 2.2 Организация памяти данных

Память данных разделена на четыре банка, которые содержат регистры общего и специального (SFR) назначения. Биты RP1 (STATUS<6>) и RP0 (STATUS<5>) предназначены для управления банками данных.

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

RP1: RP0	Банк
00	0
01	1
10	2
11	3

Объем банков памяти данных до 128 байт (7Fh). В начале банка размещаются регистры специального назначения, затем регистры общего назначения выполненные как статическое ОЗУ. Все реализованные банки содержат регистры специального назначения. Некоторые, часто используемые регистры специального назначения могут отображаться и в других банках памяти.

**Примечание.** Описание EEPROM памяти данных смотрите в разделе 4.0.

### 2.2.1 Регистры общего назначения

Обратиться к регистрам общего назначения можно прямой или косвенной адресацией, через регистр FSR.

Рис. 2-3 Карта памяти данных микроконтроллеров PIC16F877/876

Регистр косвенной адресации		Регистр косвенной адресации		Регистр косвенной адресации		Регистр косвенной адресации		Адрес
	00h		80h		100h		180h	
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h	
PCL	02h	PCL	82h	PCL	102h	PCL	182h	
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h	
FSR	04h	FSR	84h	FSR	104h	FSR	184h	
PORTA	05h	TRISA	85h		105h		185h	
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h	
PORTC	07h	TRISC	87h		107h		187h	
PORTD <sup>(1)</sup>	08h	TRISD <sup>(1)</sup>	88h		108h		188h	
PORTE <sup>(1)</sup>	09h	TRISE <sup>(1)</sup>	89h		109h		189h	
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch	
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh	
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Резерв <sup>(2)</sup>	18Eh	
TMR1H	0Fh		8Fh	EEADRH	10Fh	Резерв <sup>(2)</sup>	18Fh	
T1CON	10h		90h		110h		190h	
TMR2	11h	SSPCON2	91h		111h		191h	
T2CON	12h	PR2	92h		112h		192h	
SSPBUF	13h	SSPADD	93h		113h		193h	
SSPCON	14h	SSPSTAT	94h		114h		194h	
CCPR1L	15h		95h		115h		195h	
CCPR1H	16h		96h	Регистры общего назначения	116h	Регистры общего назначения	196h	
CCP1CON	17h		97h	16 байт	117h	16 байт	197h	
RCSTA	18h	TXSTA	98h		118h		198h	
TXREG	19h	SPBRG	99h		119h		199h	
RCREG	1Ah		9Ah		11Ah		19Ah	
CCPR2L	1Bh		9Bh		11Bh		19Bh	
CCPR2H	1Ch		9Ch		11Ch		19Ch	
CCP2CON	1Dh		9Dh		11Dh		19Dh	
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh	
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh	
	20h		A0h		120h		1A0h	
Регистры общего назначения		Регистры общего назначения		Регистры общего назначения		Регистры общего назначения		
96 байт		80 байт		80 байт		80 байт		
			EFh		16Fh		1EFh	
		Доступ к 70h-7Fh	F0h	Доступ к 70h-7Fh	170h	Доступ к 70h-7Fh	1F0h	
	7Fh		FFh		17Fh		1FFh	
Банк 0		Банк 1		Банк 2		Банк 3		

\* - не физический регистр  
Закрашенные участки памяти данных не реализованы, значение при чтении 00h

**Примечания:**

- Эти регистры не реализованы в PIC16F876
- Резервные регистры, пользователь не должен их использовать.



Рис. 2-4 Карта памяти данных микроконтроллеров PIC16F874/873

Регистр косвенной адресации		Регистр косвенной адресации		Регистр косвенной адресации		Регистр косвенной адресации		Адрес
	00h		80h		100h		180h	
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h	
PCL	02h	PCL	82h	PCL	102h	PCL	182h	
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h	
FSR	04h	FSR	84h	FSR	104h	FSR	184h	
PORTA	05h	TRISA	85h		105h		185h	
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h	
PORTC	07h	TRISC	87h		107h		187h	
PORTD <sup>(1)</sup>	08h	TRISD <sup>(1)</sup>	88h		108h		188h	
PORTE <sup>(1)</sup>	09h	TRISE <sup>(1)</sup>	89h		109h		189h	
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch	
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh	
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Резерв <sup>(2)</sup>	18Eh	
TMR1H	0Fh		8Fh	EEADRH	10Fh	Резерв <sup>(2)</sup>	18Fh	
T1CON	10h		90h		110h		190h	
TMR2	11h	SSPCON2	91h					
T2CON	12h	PR2	92h					
SSPBUF	13h	SSPADD	93h					
SSPCON	14h	SSPSTAT	94h					
CCPR1L	15h		95h					
CCPR1H	16h		96h					
CCP1CON	17h		97h					
RCSTA	18h	TXSTA	98h					
TXREG	19h	SPBRG	99h					
RCREG	1Ah		9Ah					
CCPR2L	1Bh		9Bh					
CCPR2H	1Ch		9Ch					
CCP2CON	1Dh		9Dh					
ADRESH	1Eh	ADRESL	9Eh					
ADCON0	1Fh	ADCON1	9Fh					
	20h		A0h		120h		1A0h	
Регистры общего назначения		Регистры общего назначения		Доступ к 20h-7Fh		Доступ к A0h-FFh		
96 байт		96 байт						
	7Fh		FFh		17Fh		1FFh	
Банк 0		Банк 1		Банк 2		Банк 3		

\* - не физический регистр  
Закрашенные участки памяти данных не реализованы, значение при чтении 00h

**Примечания:**

- Эти регистры не реализованы в PIC16F873
- Резервные регистры, пользователь не должен их использовать.

## 2.2.2 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 2-1** Регистры специального назначения

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	
<b>Банк 0</b>											
00h <sup>(3)</sup>	INDF	Обращение к регистру, адрес которого записан в FSR (не физический регистр)								0000 0000	
01h	TMR0	Регистр таймера 0								xxxx xxxx	
02h <sup>(3)</sup>	PCL	Младше биты счетчика команд PC								0000 0000	
03h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	-TO	-PD	Z	DC	C	0001 1xxx	
04h <sup>(3)</sup>	FSR	Регистр адреса при косвенной адресации								xxxx xxxx	
05h	PORTA	-	-	Зап. в вых. защелку PORTA, чтение сост. выв. PORTA							--0x 0000
06h	PORTB	Запись в выходную защелку PORTB, чтение состояния выводов PORTB								xxxx xxxx	
07h	PORTC	Запись в выходную защелку PORTC, чтение состояния выводов PORTC								xxxx xxxx	
08h <sup>(4)</sup>	PORTD	Запись в выходную защелку PORTD, чтение состояния выводов PORTD								xxxx xxxx	
09h <sup>(4)</sup>	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -xxx	
0Ah <sup>(1,3)</sup>	PCLATH	-	-	-	Старшие биты счетчика команд PC					---0 0000	
0Bh <sup>(3)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	
0Ch	PIR1	PSPIF <sup>(2)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	
0Dh	PIR2	-	(5)	-	EEIF	BCLIF	-	-	CCP2IF	-r-0 0--0	
0Eh	TMR1L	Младший байт 16-разрядного таймера 1								xxxx xxxx	
0Fh	TMR1H	Старший байт 16-разрядного таймера 1								xxxx xxxx	
10h	T1CON	-	-	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	
11h	TMR2	Регистр таймера 2								0000 0000	
12h	T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	
13h	SSPBUF	Буфер приемника MSSP / регистр передатчика								xxxx xxxx	
14h	SSPCON	WCOL	SSPOV	SSPEN	СКР	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	
15h	CCPR1L	Младший байт захвата/сравнения/ШИМ CCP1								xxxx xxxx	
16h	CCPR1H	Старший байт захвата/сравнения/ШИМ CCP1								xxxx xxxx	
17h	CCP1CON	-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	
19h	TXREG	Регистр данных передатчика USART								0000 0000	
1Ah	RCREG	Регистр данных приемника USART								0000 0000	
1Bh	CCPR2L	Младший байт захвата/сравнения/ШИМ CCP2								xxxx xxxx	
1Ch	CCPR2H	Старший байт захвата/сравнения/ШИМ CCP2								xxxx xxxx	
1Dh	CCP2CON	-	-	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	
1Eh	ADRESH	Старший байт результат преобразования АЦП								xxxx xxxx	
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON	0000 00-0	

Обозначения: - = не используется, читается как 0; u = не изменяется; x = не известно; q = зависит от условий, r = резерв. Затененные ячейки читаются как '0'.

### Примечания:

1. Старший байт счетчика команд PC программно не доступен. В регистре PCLATH сохраняются старшие биты <12:8>, переписываемые в старший байт счетчика команд.
2. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873/876 не реализованы, всегда должны равняться нулю.
3. Обращение к этим регистрам можно выполнить из любого банка.
4. Регистры PORTD, PORTE, TRISD, TRISE не реализованы в микроконтроллерах PIC16F873/876, читаются как '0'.
5. Резервные биты PIR2<6> и PIE2<6> при записи в регистр PIR2 всегда должны равняться нулю.

Таблица 2-1 Регистры специального назначения (продолжение)

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR
<b>Банк 1</b>										
80h <sup>(3)</sup>	INDF	Обращение к регистру, адрес которого записан в FSR (не физический регистр)								0000 0000
81h	OPTION_REG	-RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
82h <sup>(3)</sup>	PCL	Младше биты счетчика команд PC								0000 0000
83h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	-TO	-PD	Z	DC	C	0001 1xxx
84h <sup>(3)</sup>	FSR	Регистр адреса при косвенной адресации								xxxx xxxx
85h	TRISA	-	-	Направление выводов PORTA						--11 1111
86h	TRISB	Направление выводов PORTB								1111 1111
87h	TRISC	Направление выводов PORTC								1111 1111
88h <sup>(4)</sup>	TRISD	Направление выводов PORTD								1111 1111
89h <sup>(4)</sup>	TRISE	IBF	OBF	IBOV	PSPMODE	-	Направление выв. PORTE			0000 -111
8Ah <sup>(1,3)</sup>	PCLATH	-	-	-	Старшие биты счетчика команд PC					---0 0000
8Bh <sup>(3)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x
8Ch	PIE1	PSPIE <sup>(2)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000
8Dh	PIE2	-	(5)	-	EEIE	BCLIE	-	-	CCP2IE	-r-0 0--0
8Eh	PCON	-	-	-	-	-	-	-POR	-BOR	---- --qg
8Fh	-	Не реализовано								-
90h	-	Не реализовано								-
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000
92h	PR2	Регистр периода таймера 2								1111 1111
93h	SSPADDD	Регистр адреса / Регистр генератора скорости обмена								0000 0000
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000
95h	-	Не реализовано								-
96h	-	Не реализовано								-
97h	-	Не реализовано								-
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010
99h	SPBRG	Регистр генератора скорости USART								0000 0000
9Ah	-	Не реализовано								-
9Bh	-	Не реализовано								-
9Ch	-	Не реализовано								-
9Dh	-	Не реализовано								-
9Eh	ADRESL	Младший байт результат преобразования АЦП								xxxx xxxx
9Fh	ADCON1	ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000

Обозначения: - = не используется, читается как 0; u = не изменяется; x = не известно; q = зависит от условий, r = резерв. Затененные ячейки читаются как '0'.

**Примечания:**

1. Старший байт счетчика команд PC программно не доступен. В регистре PCLATH сохраняются старшие биты <12:8>, переписываемые в старший байт счетчика команд.
2. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873/876 не реализованы, всегда должны равняться нулю.
3. Обращение к этим регистрам можно выполнить из любого банка.
4. Регистры PORTD, PORTE, TRISD, TRISE не реализованы в микроконтроллерах PIC16F873/876, читаются как '0'.
5. Резервные биты PIR2<6> и PIE2<6> при записи в регистр PIR2 всегда должны равняться нулю.

Таблица 2-1 Регистры специального назначения (продолжение)

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR
<b>Банк 2</b>										
100h <sup>(3)</sup>	INDF	Обращение к регистру, адрес которого записан в FSR (не физический регистр)								0000 0000
101h	TMR0	Регистр таймера 0								xxxx xxxx
102h <sup>(3)</sup>	PCL	Младше биты счетчика команд PC								0000 0000
103h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	-TO	-PD	Z	DC	C	0001 1xxx
104h <sup>(3)</sup>	FSR	Регистр адреса при косвенной адресации								xxxx xxxx
105h	-	Не реализовано								-
106h	PORTB	Запись в выходную защелку PORTB, чтение состояния выводов PORTB								xxxx xxxx
107h	-	Не реализовано								-
108h	-	Не реализовано								-
109h	-	Не реализовано								-
10Ah <sup>(1,3)</sup>	PCLATH	-	-	-	Старшие биты счетчика команд PC					---0 0000
10Bh <sup>(3)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x
10Ch	EEDATA	Регистр данных, младший байт								xxxx xxxx
10Dh	EEADR	Регистр адреса, младший байт								xxxx xxxx
10Eh	EEDATH	-	-	Регистр данных, старший байт					xxxx xxxx	
10Fh	EEADRH	-	-	-	Регистр адреса, старший байт					xxxx xxxx
<b>Банк 3</b>										
180h <sup>(3)</sup>	INDF	Обращение к регистру, адрес которого записан в FSR (не физический регистр)								0000 0000
181h	OPTION_REG	-RBPV	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
182h <sup>(3)</sup>	PCL	Младше биты счетчика команд PC								0000 0000
183h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	-TO	-PD	Z	DC	C	0001 1xxx
184h <sup>(3)</sup>	FSR	Регистр адреса при косвенной адресации								xxxx xxxx
185h	-	Не реализовано								-
186h	TRISB	Направление выводов PORTB								1111 1111
187h	-	Не реализовано								-
188h	-	Не реализовано								-
189h	-	Не реализовано								-
18Ah <sup>(1,3)</sup>	PCLATH	-	-	-	Старшие биты счетчика команд PC					---0 0000
18Bh <sup>(3)</sup>	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x
18Ch	EECON1	EEPGRD	-	-	-	WREER	WREN	WR	RD	x--- x000
18Dh	EECON2	Регистр управления 2 (физически не реализован)								---- ----
18Eh	-	Резерв								-
18Fh	-	Резерв								-

Обозначения: - = не используется, читается как 0; u = не изменяется; x = не известно; q = зависит от условий, r = резерв. Затененные ячейки читаются как '0'.

**Примечания:**

1. Старший байт счетчика команд PC программно не доступен. В регистре PCLATH сохраняются старшие биты <12:8>, переписываемые в старший байт счетчика команд.
2. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873/876 не реализованы, всегда должны равняться нулю.
3. Обращение к этим регистрам можно выполнить из любого банка.
4. Регистры PORTD, PORTE, TRISD, TRISE не реализованы в микроконтроллерах PIC16F873/876, читаются как '0'.
5. Резервные биты PIR2<6> и PIE2<6> при записи в регистр PIR2 всегда должны равняться нулю.

**2.2.2.1 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 также не воздействуют на биты -TO и -PD. Поэтому, результат выполнения команды с регистром STATUS может отличаться от ожидаемого. Например, команда CLRFS STATUS сбросит три старших бита и установит бит Z (состояние регистра STATUS после выполнения команды 000и1ии, где и - 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 13.0.

**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: <b>IRP:</b> Бит выбора банка при косвенной адресации            1 = банк 2, 3 (100h – 1FFh)            0 = банк 0, 1 (000h - 0FFh)</p> <p>биты 6-5: <b>RP1:RP0:</b> 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: <b>-TO:</b> Флаг переполнения сторожевого таймера            1 = после POR или выполнения команд CLRWDT, SLEEP            0 = после переполнения WDT</p> <p>бит 3: <b>-PD:</b> Флаг включения питания            1 = после POR или выполнения команды CLRWDT            0 = после выполнения команды SLEEP</p> <p>бит 2: <b>Z:</b> Флаг нулевого результата            1 = нулевой результат выполнения арифметической или логической операции            0 = не нулевой результат выполнения арифметической или логической операции</p> <p>бит 1: <b>DC:</b> 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: <b>C:</b> 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><b>Примечание.</b> 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

**2.2.2.2 Регистр OPTION\_REG**

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

- Предварительным делителем TMR0/WDT;
- Активным фронтом внешнего прерывания RB0/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\_REG (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
<b>-RBPU</b>	<b>INTEDG</b>	<b>T0CS</b>	<b>T0SE</b>	<b>PSA</b>	<b>PS2</b>	<b>PS1</b>	<b>PS0</b>
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

**Примечание.** При использовании режима низковольтного программирования и включенных подтягивающих резисторах на PORTB необходимо сбросить в '0' 3-й бит регистра TRISB для выключения подтягивающего резистора на выводе RB3.

**2.2.2.3 Регистр INTCON**

Регистр INTCON доступен для чтения и записи, содержит биты разрешений и флаги прерываний: переполнение TMR0; изменения уровня сигнала на выводах PORTB; внешний источник прерываний RB0/INT.

**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
<p>бит 7: <b>GIE</b>: Глобальное разрешение прерываний 1 = разрешены все немаскированные прерывания 0 = все прерывания запрещены</p> <p>бит 6: <b>PEIE</b>: Разрешение прерываний от периферийных модулей 1 = разрешены все немаскированные прерывания периферийных модулей 0 = прерывания от периферийных модулей запрещены</p> <p>бит 5: <b>TOIE</b>: Разрешение прерывания по переполнению TMR0 1 = прерывание разрешено 0 = прерывание запрещено</p> <p>бит 4: <b>INTE</b>: Разрешение внешнего прерывания INT 1 = прерывание разрешено 0 = прерывание запрещено</p> <p>бит 3: <b>RBIE</b>: Разрешение прерывания по изменению сигнала на входах RB7:RB4 PORTB 1 = прерывание разрешено 0 = прерывание запрещено</p> <p>бит 2: <b>TOIF</b>: Флаг прерывания по переполнению TMR0 1 = произошло переполнение TMR0 (сбрасывается программно) 0 = переполнения TMR0 не было</p> <p>бит 1: <b>INTF</b>: Флаг внешнего прерывания INT 1 = выполнено условие внешнего прерывания на выводе RB0/INT (сбрасывается программно) 0 = внешнего прерывания не было</p> <p>бит 0: <b>RBIF</b>: Флаг прерывания по изменению уровня сигнала на входах RB7:RB4 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</p>							

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

**2.2.2.4 Регистр PIE1**

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

**Примечание.** Чтобы разрешить периферийные прерывания необходимо установить в '1' бит PEIE(INTCON<6>).

**Регистр PIE1 (адрес 8Ch)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>PSPIE<sup>(1)</sup></b>	<b>ADIE</b>	<b>RCIE</b>	<b>TXIE</b>	<b>SSPIE</b>	<b>CCP1IE</b>	<b>TMR2IE</b>	<b>TMR1IE</b>
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: **PSPIE<sup>(1)</sup>**: Разрешение прерывания записи/чтения ведомого параллельного порта  
1 = прерывание разрешено  
0 = прерывание запрещено

бит 6: **ADIE**: Разрешение прерывания по окончании преобразования АЦП  
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: **SSPIE**: Разрешение прерывания от модуля синхронного последовательного порта  
1 = прерывание разрешено  
0 = прерывание запрещено

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

**Примечание 1.** Бит PSPIE в микроконтроллерах PIC16F873/876 не реализован, всегда должен равняться нулю.



**2.2.2.5 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	R/W-0	R/W-0	R/W-0	R/W-0
<b>PSPIF<sup>(1)</sup></b>	<b>ADIF</b>	<b>RCIF</b>	<b>TXIF</b>	<b>SSPIF</b>	<b>CCP1IF</b>	<b>TMR2IF</b>	<b>TMR1IF</b>
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:	<b>PSPIF<sup>(1)</sup></b> : Флаг прерывания от ведомого параллельного порта 1 = произошла операция чтения или записи (сбрасывается программно) 0 = операции чтения или записи не происходило						
бит 6:	<b>ADIF</b> : Флаг прерывания от модуля АЦП 1 = преобразование АЦП завершено 0 = преобразование АЦП не завершено						
бит 5:	<b>RCIF</b> : Флаг прерывания от приемника USART 1 = буфер приемника USART полон 0 = буфер приемника USART пуст						
бит 4:	<b>TXIF</b> : Флаг прерывания от передатчика USART 1 = буфер передатчика USART пуст 0 = буфер передатчика USART полон						
бит 3:	<b>SSPIF</b> : Флаг прерываний от модуля MSSP 1 = выполнено условие возникновения прерывания от модуля MSSP (сбрасывается программно). Условия возникновения прерывания: <ul style="list-style-type: none"> <li>• SPI               <ul style="list-style-type: none"> <li>- Выполнен прием/передача данных.</li> </ul> </li> <li>• Ведомый I2C               <ul style="list-style-type: none"> <li>- Выполнен прием/передача данных.</li> </ul> </li> <li>• Ведущий I2C               <ul style="list-style-type: none"> <li>- Выполнен прием/передача данных.</li> <li>- Завершено формирование на шине бита START.</li> <li>- Завершено формирование на шине бита STOP.</li> <li>- Завершено формирование на шине бита повторный START.</li> <li>- Завершено формирование на шине бита подтверждения.</li> <li>- Обнаружено на шине формирование бита START (для режима с несколькими ведущими).</li> <li>- Обнаружено на шине формирование бита STOP (для режима с несколькими ведущими).</li> </ul> </li> </ul>						
	0 = условие возникновения прерывания от модуля MSSP не выполнено						
бит 2:	<b>CCP1IF</b> : Флаг прерывания от модуля CCP1 <u>Режим захвата</u> 1 = выполнен захват значения TMR1 (сбрасывается программно) 0 = захвата значения TMR1 не происходило <u>Режим сравнения</u> 1 = значение TMR1 достигло указанного в регистрах CCPR1H:CCPR1L(сбрасывается программно) 0 = значение TMR1 не достигло указанного в регистрах CCPR1H:CCPR1L <u>ШИМ режим</u> Не используется						
бит 1:	<b>TMR2IF</b> : Флаг прерывания по переполнению TMR2 1 = произошло переполнение TMR2 (сбрасывается программно) 0 = переполнения TMR2 не было						
бит 0:	<b>TMR1IF</b> : Флаг прерывания по переполнению TMR1 1 = произошло переполнение TMR1 (сбрасывается программно) 0 = переполнения TMR1 не было						
<b>Примечание 1.</b> Бит PSPIF в микроконтроллерах PIC16F873/876 не реализован, всегда должен равняться нулю.							

**2.2.2.6 Регистр PIE2**

Регистр PIE1 доступен для чтения и записи, содержит биты разрешения прерываний от модуля CCP2, возникновения коллизий на шине и окончания записи в EEPROM память данных.

**Регистр PIE2 (адрес 8Dh)**

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
-	<b>Резерв</b>	-	<b>EEIE</b>	<b>BCLIE</b>	-	-	<b>CCP1IE</b>
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>							
<p>бит 7: <b>Не реализован:</b> читается как '0'</p> <p>бит 6: <b>Резерв:</b> всегда должен равняться нулю</p> <p>бит 5: <b>Не реализован:</b> читается как '0'</p> <p>бит 4: <b>EEIE:</b> Разрешение прерывания по окончанию записи в EEPROM данных            1 = прерывание разрешено            0 = прерывание запрещено</p> <p>бит 3: <b>BCLIE:</b> Разрешение прерывания при возникновении коллизий на шине            1 = прерывание разрешено            0 = прерывание запрещено</p> <p>биты 2-1: <b>Не реализованы:</b> читаются как '0'</p> <p>бит 0: <b>CCP2IE:</b> Разрешение прерывания от модуля CCP2            1 = прерывание разрешено            0 = прерывание запрещено</p>							

**2.2.2.7 Регистр PIR2**

Регистр PIR1 доступен для чтения и записи, содержит флаги прерываний от модуля CCP2, возникновения коллизий на шине и окончания записи в EEPROM память данных.

**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 PIR2 (адрес 0Dh)**

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
-	<b>Резерв</b>	-	<b>EEIF</b>	<b>BCLIF</b>	-	-	<b>CCP1IF</b>
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'

бит 6: **Резерв:** всегда должен равняться нулю

бит 5: **Не реализован:** читается как '0'

бит 4: **EEIF:** Флаг прерывания по окончанию записи в EEPROM данных  
1 = запись в EEPROM данных завершена (сбрасывается программно)  
0 = запись в EEPROM данных не завершена или не была начата

бит 3: **BCLIF:** Флаг прерывания возникновения коллизий на шине  
1 = на шине обнаружены коллизии (только в режиме ведущего I<sup>2</sup>C)  
0 = коллизий не обнаружено

биты 2-1: **Не реализованы:** читаются как '0'

бит 0: **CCP2IF:** Флаг прерывания от модуля CCP2  
Режим захвата  
1 = выполнен захват значения TMR1 (сбрасывается программно)  
0 = захвата значения TMR1 не происходило  
Режим сравнения  
1 = значение TMR1 достигло указанного в регистрах CCP2H:CCP2L (сбрасывается программно)  
0 = значение TMR1 не достигло указанного в регистрах CCP2H:CCP2L  
ШИМ режим  
Не используется

**2.2.2.8 Регистр PCON**

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

- Сброс по включению питания (POR);
- Сброс по сигналу на выводе -MCLR;
- Сброс по переполнению сторожевого таймера WDT;
- Сброс по обнаружению снижения напряжения питания (BOR).

**Примечание.** При включении питания бит -BOR имеет непредсказуемое значение и не должен учитываться. Бит -BOR предназначен для обнаружения последующих сбросов микроконтроллера при снижении напряжения питания. Состояние бита -BOR также непредсказуемое, если работа детектора пониженного напряжения заблокирована в битах конфигурации при программировании микроконтроллера (BODEN=0).

**Регистр PCON (адрес 8Eh)**

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1
-	-	-	-	-	-	<b>-POR</b>	<b>-BOR</b>
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 7-2:**Не реализованы:** читаются как '0'

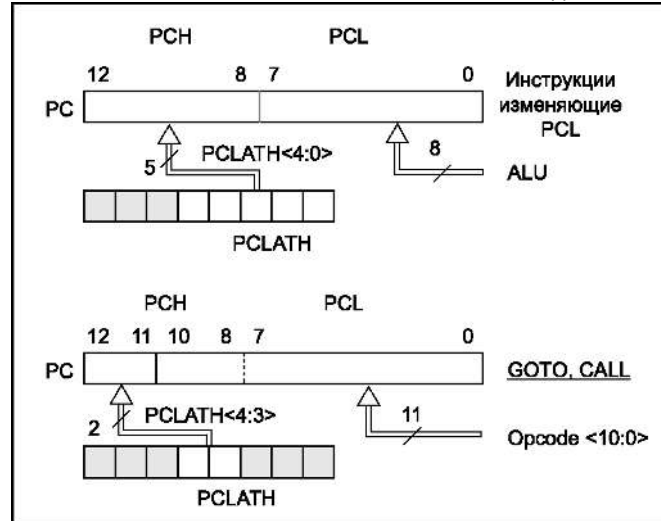
бит 1: **-POR:** Флаг сброса по включению питания  
1 = сброса по включению питания не было  
0 = произошел сброс микроконтроллера по включению питания (программно должен быть установлен в '1' для обнаружения сброса POR)

бит 0: **-BOR:** Флаг сброса по снижению напряжения питания  
1 = сброса по снижению напряжения питания не было  
0 = произошел сброс микроконтроллера по снижению напряжения питания (программно должен быть установлен в '1' для обнаружения сброса BOR)

## 2.3 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 PCH происходят через дополнительный регистр PCLATH. При любом виде сброса микроконтроллера счетчик команд PC очищается. На рисунке 2-5 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. 2-5 Запись значения в счетчик команд PC



### 2.3.1 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».

### 2.3.2 Стек

PIC16F87X имеют 8-уровневый 13-разрядный аппаратный стек (см. рисунки 2-1, 2-2). 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 обработки прерываний. Чтение из стека и запись в счетчик команд PC происходит при выполнении инструкций возвращения из подпрограммы или обработки 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.

## 2.4 Страницы памяти программ

Все микроконтроллеры PIC16F87X способны адресовать 8К слов памяти программ. Инструкции переходов (CALL и GOTO) имеют 11-разрядное поле для указания адреса, что позволяет непосредственно адресовать 2Кслов памяти программ. Для адресации верхних страниц памяти программ используются 2 бита в регистре PCLATH<4:3>. Перед выполнением команды перехода (CALL или GOTO) необходимо запрограммировать биты регистра PCLATH<4:3> для адресации требуемой страницы.

При выполнении инструкций возврата из подпрограммы, 13-разрядное значение для счетчика программ PC берется с вершины стека, поэтому манипуляция битами регистра PCLATH<3:4> не требуется.

**Примечание.** Содержимое регистра PCLATH не изменяется поле выполнения инструкции RETURN или RETFIE. Пользователь должен сам изменить значения регистра PCLATH для последующего выполнения команд GOTO и CALL.

В примере 2-1 показан переход со страницы 0 на страницу 1 памяти программ. Этот пример предполагает, что в подпрограмме сохраняется и восстанавливается значение регистра PCLATH.

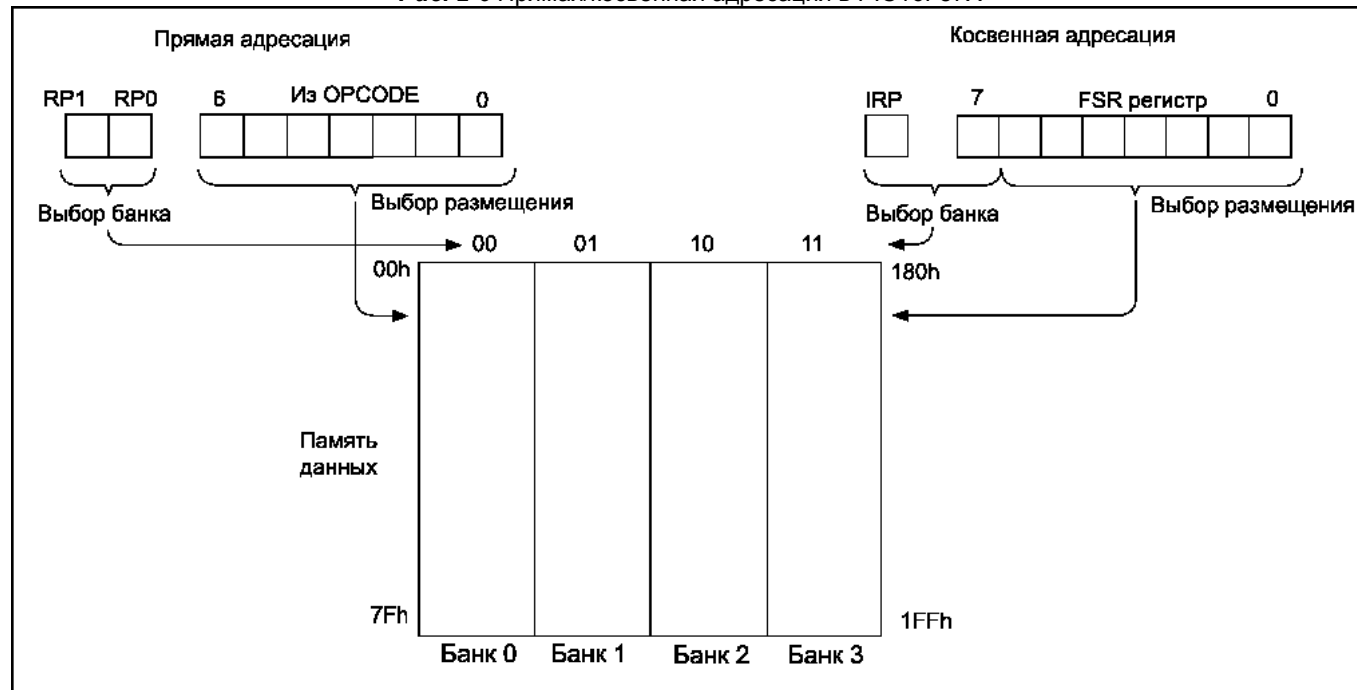
**Пример 2-1** Переход на 1 страницу памяти программ с 0 страницы

```
ORG    0x500
BSF    PCLATH,3    ; Выбор страницы 1 (800h-FFFh)
CALL   SUB1_P1     ; Переход на страницу 1 (800h-FFFh)
:
:
ORG    0x900
SUB1_P1:           ; Страница 1 (800h-FFFh)
:
RETURN          ; Возврат на страницу 0 (000h-7FFh)
```

## 2.5 Косвенная адресация, регистры INDF и FSR

Для выполнения косвенной адресации необходимо обратиться к физически не реализованному 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 2-6.

Рис. 2-6 Прямая/косвенная адресация в PIC16F87X



**Примечание.** Карту памяти данных смотрите на рисунке 2-3 и 2-4.

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

### Пример 2-2 Косвенная адресация

```

BCF    STATUS, IRP    ; Установить банк 0,1
MOVLW 0x20            ; Указать первый регистр в O3Y
MOVWF  FSR
NEXT:
CLRF   INDF           ; Очистить регистр
INCF   FSR,F          ; Увеличить адрес
BTFSS  FSR,4          ; Завершить?
GOTO   NEXT           ; Нет, продолжить очистку

CONTINUE:
; Да

```

### 3.0 Порты ввода/вывода

Некоторые каналы портов ввода/вывода мультиплицированы с периферийными модулями микроконтроллера. Когда периферийный модуль включен, вывод не может использоваться как универсальный канал ввода/вывода.

Дополнительную информацию по работе с портами ввода/вывода смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual".

#### 3.1 Регистры PORTA и TRISA

PORTA – 6-разрядный порт ввода вывода. 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 (если выходная защелка подключена к выводу 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.

RA4 - имеет триггер Шмидта на входе и открытый сток на выходе, мультиплицирован с тактовым входом T0CKI. Все остальные каналы PORTA имеют TTL буфер на входе и полнофункциональные выходные КМОП буферы.

Каналы PORTA мультиплицированы с аналоговыми входами АЦП и аналоговым входом источника опорного напряжения  $V_{REF}$ . Биты управления режимов работы каналов порта ввода/вывода PORTA находятся в регистре ADCON1.

**Примечание.** После сброса по включению питания выводы настраиваются как аналоговые входы, а чтение дает результат '0'.

Биты регистра TRISA управляют направлением каналов PORTA, даже когда они используются как аналоговые входы. Пользователь должен удостовериться, что соответствующие каналы PORTA настроены на вход при использовании их в качестве аналоговых входов.

##### Пример 3-1 Инициализация PORTA

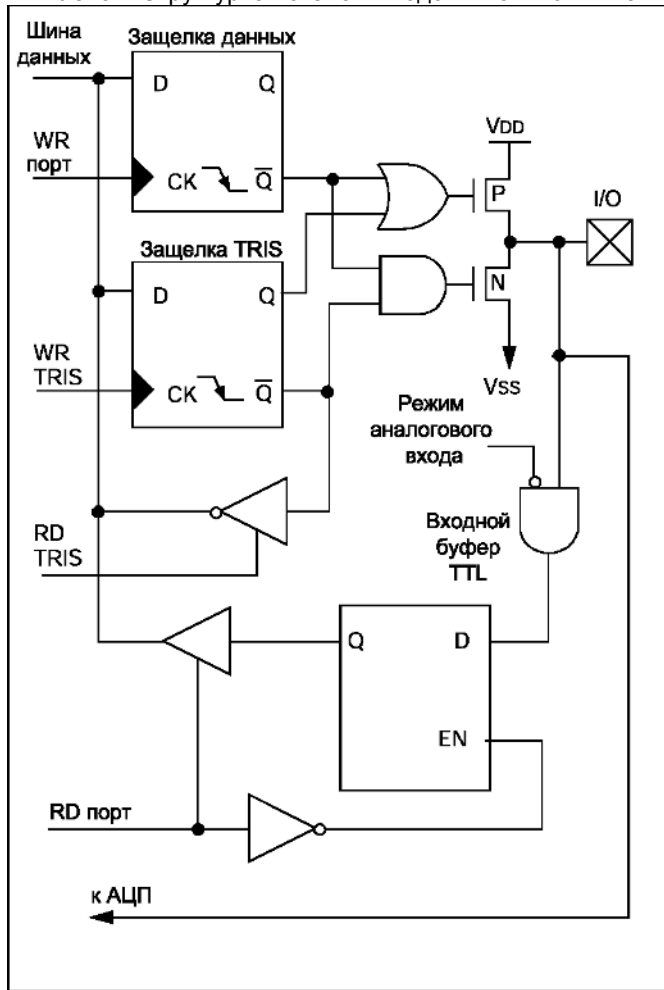
```
BCF      STATUS, RP1
BCF      STATUS, RP0    ; Выбрать банк 0
CLRF     PORTA          ; Инициализация защелок PORTA

BSF      STATUS, RP0    ; Выбрать банк 1
MOVLW   0X06            ;
MOVWF   ADCON1          ; Каналы PORTA – цифровые входы/выходы

MOVLW   0xCF            ; Значение для инициализации
                        ; направления каналов PORTA
MOVWF   TRISA           ; Настроить RA<3:0> как входы,
                        ; настроить RA<5:4> как выходы
                        ; Биты TRISA<7:6> всегда
                        ; читаются как '0'.
```

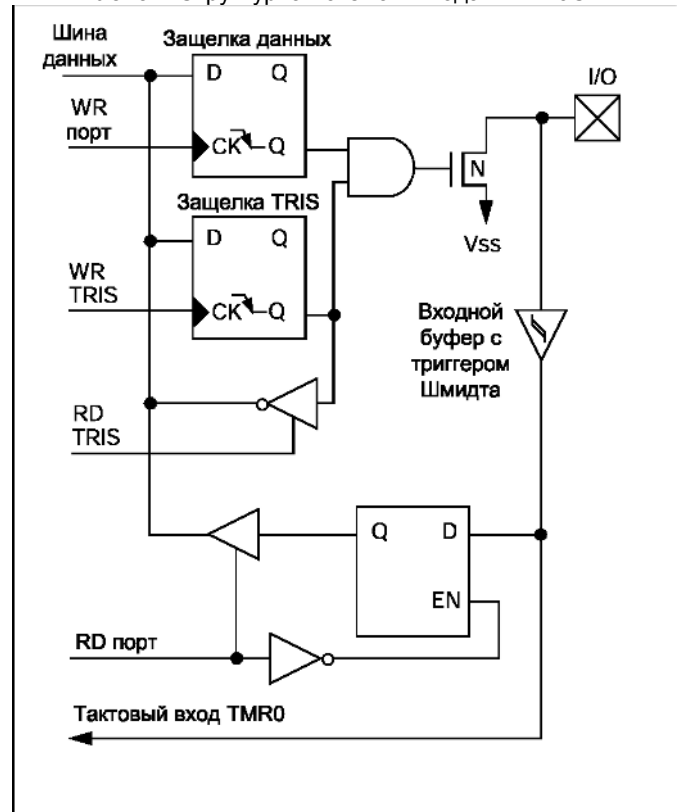


**Рис. 3-1** Структурная схема выводов RA3:RA0 и RA5



**Примечание.** Выводы портов имеют защитные диоды, подключенные к V<sub>DD</sub> и V<sub>SS</sub>.

**Рис. 3-2** Структурная схема вывода RA4/T0CKI



**Примечание.** Вывод имеет защитный диоды, подключенный только к V<sub>SS</sub>.

**Таблица 3-1** Функциональное назначение выводов PORTA

Обозначение вывода	№ бита	Тип буфера	Описание
RA0/AN0	бит 0	TTL	Двунаправленный порт ввода/вывода или аналоговый вход
RA1/AN1	бит 1	TTL	Двунаправленный порт ввода/вывода или аналоговый вход
RA2/AN2	бит 2	TTL	Двунаправленный порт ввода/вывода или аналоговый вход
RA3/AN3	бит 3	TTL	Двунаправленный порт ввода/вывода или аналоговый вход
RA4/T0CKI	бит 4	ST	Двунаправленный порт ввода/вывода, может использоваться как T0CKI, выход с открытым стоком
RA5-SS/AN4	бит 5	TTL	Двунаправленный порт ввода/вывода или вход выбора синхронного последовательного порта или аналоговый вход

Обозначение: ST = вход с триггером Шмидта; TTL = входной буфер TTL.

**Таблица 3-2** 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, BOR	Другие сбросы
05h	PORTA	-	-	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	-	-	Регистр направления данных PORTA						--11 1111	--11 1111
9Fh	ADCON1	ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000

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

**Примечание.** При работе модуля MSSP в ведомом режиме SPI с использованием вывода -SS, АЦП должно работать в одном из следующих режимов PCFG3:PCFG0 = 0100, 0101, 011x, 1101, 1110, 1111.

### 3.2 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 (если выходная защелка подключена к выводу микроконтроллера).

Три вывода PORTB мультиплицированы со схемой низковольтного программирования: RB3/PGM, RB6/PGC, RB7/PGD. Описание режима низковольтного программирования смотрите в разделе 12.0.

К каждому выводу PORTB подключен внутренний подтягивающий резистор. Бит -RBPU (OPTION\_REG <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 (INTCON<0>), и если разрешено, генерируется прерывание.

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

- Выполнить чтение или запись в PORTB, исключив несоответствие;
- Сбросить флаг RBIF в '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 рекомендуется использовать для определения нажатия клавиш, когда PORTB полностью задействован для реализации клавиатуры. Не рекомендуется опрашивать PORTB при использовании прерываний по изменению входного сигнала.

Прерывания по изменению сигнала на входах PORTB и 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](http://www.microchip.com) и [www.microchip.ru](http://www.microchip.ru)).

RB0/INT вход внешнего источника прерываний, настраиваемых битом INTEDG (OPTION\_REG<6>). Подробное описание использования прерываний INT смотрите в разделе 12.10.1.

Рис. 3-3 Структурная схема выводов RB3:RB0

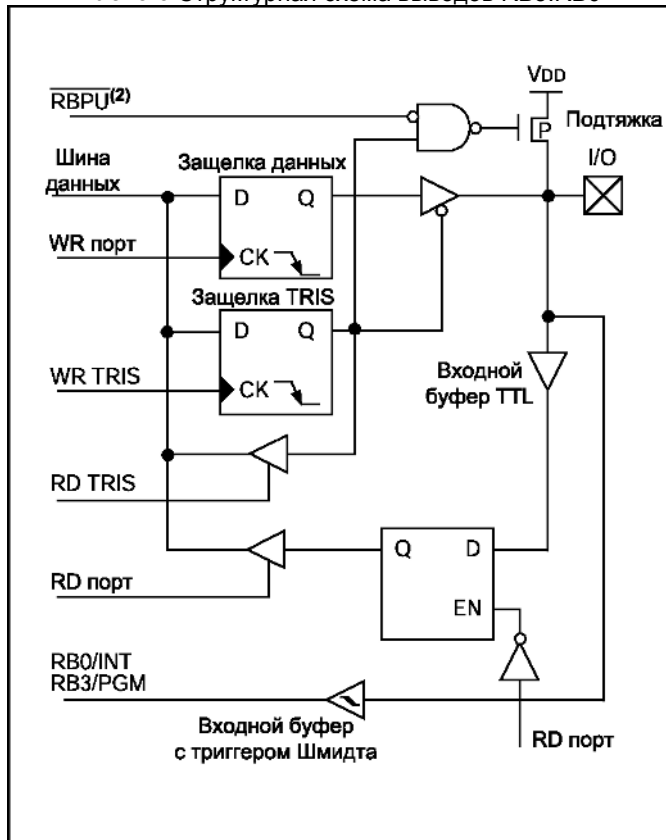
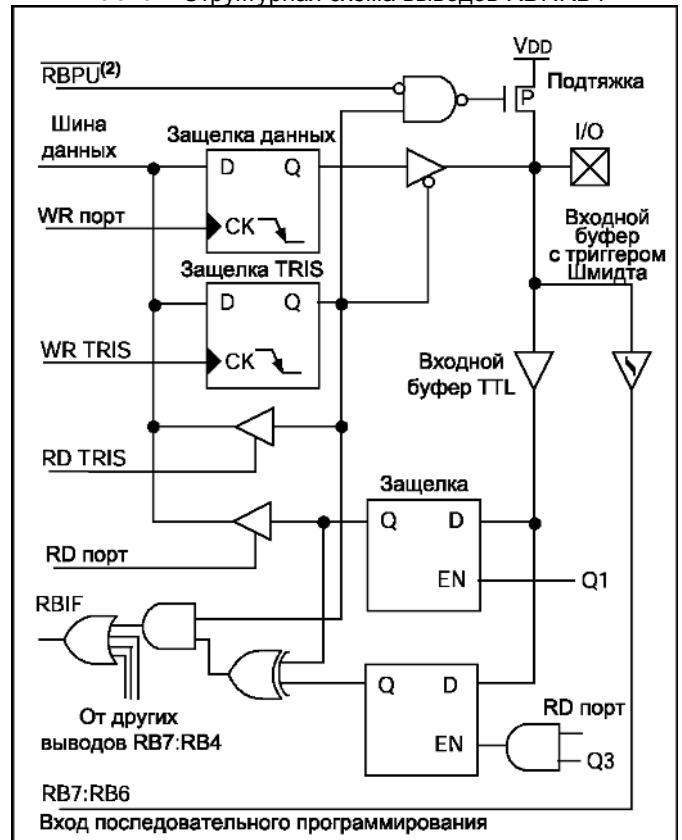


Рис. 3-4 Структурная схема выводов RB7:RB4



**Примечания:**

1. Выводы портов имеют защитные диоды, подключенные к  $V_{DD}$  и  $V_{SS}$ .
2. Для включения подтягивающих резисторов необходимо установить в '1' соответствующий бит TRIS и сбросить в '0' бит -RBPU (OPTION\_REG<7>).

**Таблица 3-3** Функциональное назначение выводов PORTB

Обозначение вывода	№ бита	Тип буфера	Описание
RB0/INT	бит 0	TTL/ST <sup>(1)</sup>	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора, вход внешнего прерывания.
RB1	бит 1	TTL	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора.
RB2	бит 2	TTL	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора.
RB3/PGM <sup>(3)</sup>	бит 3	TTL	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора или вход программирования в режиме LVP.
RB4	бит 4	TTL	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора и прерыванием по изменению входного сигнала.
RB5	бит 5	TTL	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора и прерыванием по изменению входного сигнала.
RB6/PGC	бит 6	TTL/ST <sup>(2)</sup>	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора и прерыванием по изменению входного сигнала. Тактовый вход в режиме программирования.
RB7/PGD	бит 7	TTL/ST <sup>(2)</sup>	Двунаправленный порт ввода/вывода с программным включением подтягивающего резистора и прерыванием по изменению входного сигнала. Вывод данных в режиме программирования.

Обозначение: ST = вход с триггером Шмидта; TTL = входной буфер TTL.

**Примечания:**

1. Входной буфер с триггером Шмидта при использовании внешних прерываний.
2. Входной буфер с триггером Шмидта при работе в режиме последовательного программирования.
3. Низковольтное программирование (LVP) ICSP разрешено по умолчанию, что отключает функцию цифрового порта ввода/вывода RB3. Для использования RB3 в качестве цифрового ввода/вывода необходимо выключить режим низковольтного программирования.

**Таблица 3-4** 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, BOR	Другие сбросы
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	Регистр направления данных PORTB								1111 1111	1111 1111
81h, 181h	OPTION_REG	-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.  
Затененные биты на работу не влияют.

### 3.3 Регистры PORTC и TRISC

PORTC – 8-разрядный двунаправленный порт ввода/вывода. Биты регистра TRISC определяют направление каналов порта. Установка бита в '1' регистра TRISC переводит выходной буфер в 3-е состояние. Запись '0' в регистр TRISC настраивает соответствующий канал как выход, содержимое защелки PORTC передается на вывод микроконтроллера (если выходная защелка подключена к выводу микроконтроллера).

Выходы PORTC мультиплицированы с несколькими периферийными модулями (см. таблицу 3-5). На каналах PORTC присутствует входной буфер с триггером Шмидта.

Когда модуль MSSP включен в режиме I<sup>2</sup>C, выходы PORTC<4:3> могут поддерживать уровни выходных сигналов по спецификации I<sup>2</sup>C или SMBus в зависимости от состояния бита CKE(SSPSTAT<6>).

При использовании периферийных модулей необходимо соответствующим образом настраивать биты регистра TRISC для каждого вывода PORTC (см. описание периферийных модулей). Некоторые периферийные модули отменяют действие битов TRISC принудительно настраивая вывод на вход или выход. В связи с чем не рекомендуется использовать команды "чтение - модификация - запись" с регистром TRISC.

Рис. 3-5 Структурная схема выводов RC7:RC5, RC2:RC0

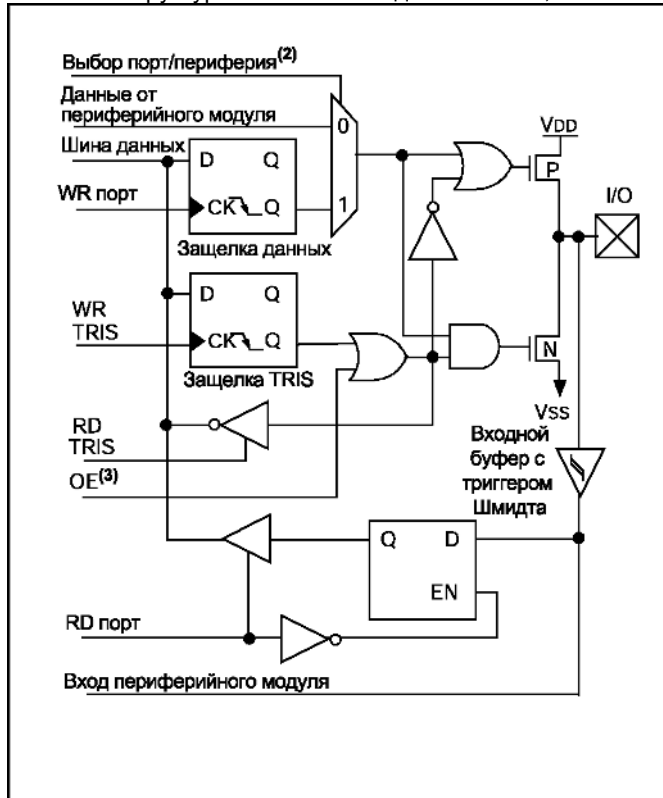
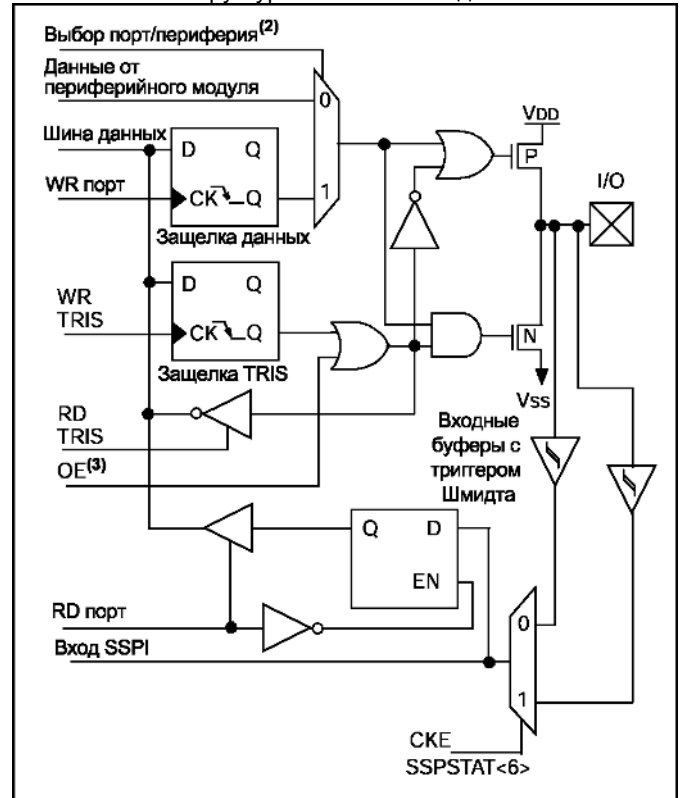


Рис. 3-6 Структурная схема выводов RC4:RC3



**Примечания:**

1. Выводы портов имеют защитные диоды, подключенные к V<sub>DD</sub> и V<sub>SS</sub>.
2. Сигнал режима канала – вывод используется периферийным модулем или цифровой порт ввода/вывода.
3. Сигнал разрешения (OE) от периферийного модуля, настраивать канал как выход.

**Таблица 3-5** Функциональное назначение выводов PORTC

Обозначение вывода	№ бита	Тип буфера	Описание
RC0/T1OSO/T1CKI	бит 0	ST	Двунаправленный порт ввода/вывода или выход генератора TMR1 / вход тактового сигнала для TMR1.
RC1/T1OSI/CCP2	бит 1	ST	Двунаправленный порт ввода/вывода или вход генератора TMR1 или вход захвата 2 / выход сравнения 2 / выход ШИМ 2.
RC2/CCP1	бит 2	ST	Двунаправленный порт ввода/вывода или вход захвата 1 / выход сравнения 1 / выход ШИМ 1.
RC3/SCK/SCL	бит 3	ST	Двунаправленный порт ввода/вывода или вход/выход тактового сигнала модуля MSSP в SPI, I <sup>2</sup> C режиме.
RC4/SDI/SDA	бит 4	ST	Двунаправленный порт ввода/вывода или вход данных в режиме SPI или вход/выход данных в режиме I <sup>2</sup> C.
RC5/SDO	бит 5	ST	Двунаправленный порт ввода/вывода или выход данных в режиме SPI.
RC6/TX/CK	бит 6	ST	Двунаправленный порт ввода/вывода или выход передатчика USART в асинхронном режиме или линия тактового сигнала USART в синхронном режиме.
RC7/RX/DT	бит 7	ST	Двунаправленный порт ввода/вывода или вход приемника USART в асинхронном режиме или линия данных USART в синхронном режиме.

Обозначение: ST = вход с триггером Шмидта.

**Таблица 3-6** Регистры и биты, связанные с работой PORTC

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	Регистр направления данных PORTC								1111 1111	1111 1111

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

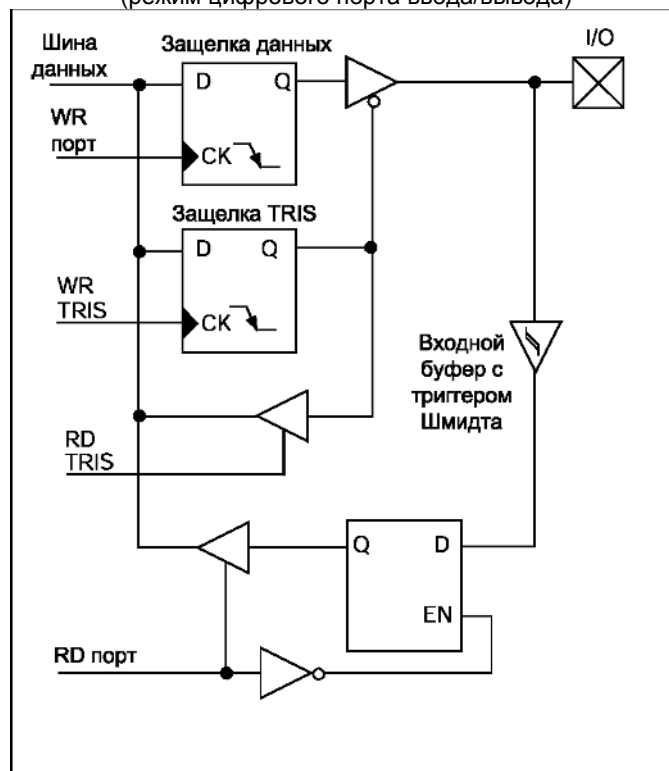
### 3.4 Регистры PORTD и TRISD

PORTD и TRISD не реализованы в микроконтроллерах PIC16F873 и PIC16F876.

PORTD – 8-разрядный двунаправленный порт ввода/вывода. Биты регистра TRISD определяют направление каналов порта.

PORTD может работать как 8-разрядный микропроцессорный порт (ведомый параллельный порт), если бит PSPMODE (TRISE<4>) установлен в '1'. В режиме ведомого параллельного порта ко входам подключены буферы TTL.

**Рис. 3-7** Структурная схема выводов PORTD  
(режим цифрового порта ввода/вывода)



**Примечание.** Выводы портов имеют защитные диоды, подключенные к  $V_{DD}$  и  $V_{SS}$ .

**Таблица 3-7** Функциональное назначение выводов PORTD

Обозначение вывода	№ бита	Тип буфера	Описание
RD0/PSP0	бит 0	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вывод ведомого параллельного порта бит 0.
RD1/PSP1	бит 1	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вывод ведомого параллельного порта бит 1.
RD2/PSP2	бит 2	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вывод ведомого параллельного порта бит 2.
RD3/PSP3	бит 3	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вывод ведомого параллельного порта бит 3.
RD4/PSP4	бит 4	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вывод ведомого параллельного порта бит 4.
RD5/PSP5	бит 5	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вывод ведомого параллельного порта бит 5.
RD6/PSP6	бит 6	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вывод ведомого параллельного порта бит 6.
RD7/PSP7	бит 7	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вывод ведомого параллельного порта бит 7.

Обозначение: ST = вход с триггером Шмидта; TTL = входной буфер TTL.

**Примечание 1.** В режиме цифрового ввода/вывода подключен буфер с триггером Шмидта, а в режиме ведомого параллельного порта подключен входной буфер TTL.

**Таблица 3-8** Регистры и биты, связанные с работой PORTD

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	Регистр направления данных PORTD								1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	-	Рег. напр. данных PORTE			0000 -111	0000 -111

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

Затененные биты на работу не влияют.

### 3.5 Регистры PORTE и TRISE

PORTE и TRISE не реализованы в микроконтроллерах PIC16F873 и PIC16F876.

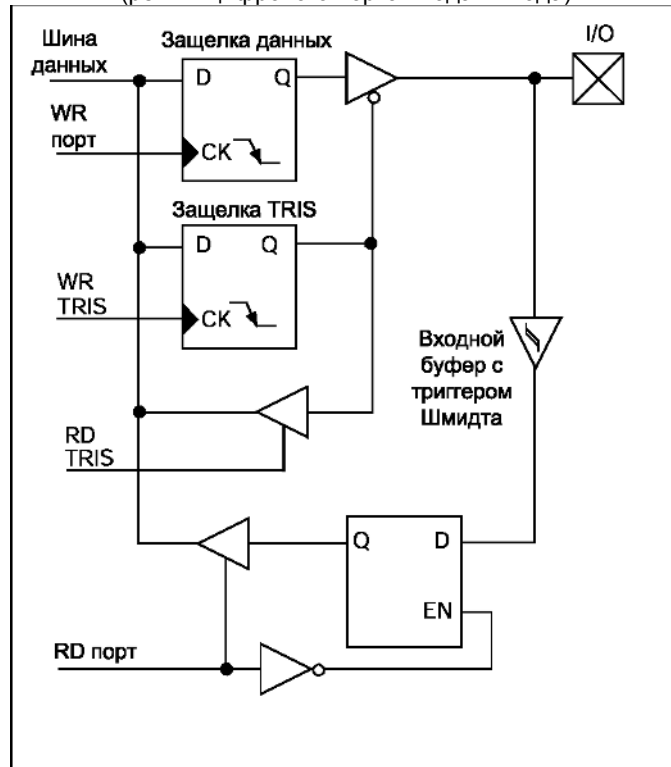
PORTE имеет три вывода (RE0/-RD/AN5, RE1/-WR/AN6, RE2/-CS/AN7), индивидуально настраиваемые на вход или выход. Выводы PORTE имеют входной буфер Шмидта.

Каналы PORTE станут управляемыми выводами ведомого параллельного порта, когда бит PSPMODE(TRISE<4>) установлен в '1'. В этом режиме биты TRISE<2:0> должны быть установлены в '1'. В регистре ADCON1 необходимо также настроить выводы PORTE как цифровые каналы ввода/вывода. В режиме ведомого параллельного порта к выводам PORTE подключены входные буферы TTL.

Выводы PORTE мультиплицированы с аналоговыми входами. Когда каналы PORTE настроены как аналоговые входы, биты регистра TRISE управляют направлением данных PORTE (чтение будет давать результат '0').

**Примечание.** После сброса по включению питания выводы настраиваются как аналоговые входы, а чтение дает результат '0'.

**Рис. 3-8** Структурная схема выводов PORTE (режим цифрового порта ввода/вывода)



**Примечание.** Выводы портов имеют защитные диоды, подключенные к  $V_{DD}$  и  $V_{SS}$ .

**Таблица 3-9** Функциональное назначение выводов PORTE

Обозначение вывода	№ бита	Тип буфера	Описание
RE0/-RD/AN5	бит 0	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вход управления чтением ведомого параллельного порта или аналоговый вход: -RD 1 = Ожидание 0 = Операция чтения. Защелка PORTD подключена к выводам PORTD (если -CS = 0)
RE1/-WR/AN6	бит 1	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вход управления записью ведомого параллельного порта или аналоговый вход: -WR 1 = Ожидание 0 = Операция записи. Данные с выводов PORTD сохраняются во внутренней защелке PORTD (если -CS = 0)
RE2/-CS/AN7	бит 2	ST/TTL <sup>(1)</sup>	Двунаправленный порт ввода/вывода или вход выбора микросхемы ведомого параллельного порта или аналоговый вход: -CS 1 = Микросхема не выбрана 0 = Микросхема выбрана

Обозначение: ST = вход с триггером Шмидта; TTL = входной буфер TTL.

**Примечание 1.** В режиме цифрового ввода/вывода подключен буфер с триггером Шмидта, а в режиме ведомого параллельного порта подключен входной буфер TTL.

**Регистр TRISE (адрес 89h)**

R-0	R-0	R-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
<b>IBF</b>	<b>OBF</b>	<b>IBOV</b>	<b>PSPMODE</b>	-	<b>BIT2</b>	<b>BIT1</b>	<b>BIT0</b>
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 управления и статуса ведомого параллельного порта**

- бит 7: **IBF**: Бит статуса приемного буфера  
1 = принят байт данных  
0 = байт данных не был получен
- бит 6: **OBF**: Бит статуса передающего буфера  
1 = предварительно записанный байт данных еще не прочитан  
0 = выходной буфер был прочитан
- бит 5: **IBOV**: Флаг переполнения приемного буфера  
1 = произошла новая запись, а предыдущий байт не был прочитан (сбрасывается программно)  
0 = переполнения не было
- бит 4: **PSPMODE**: Режим работы PORTD  
1 = PORTD работает как ведомый параллельный порт  
0 = PORTD работает в режиме цифровых каналов ввода/вывода
- бит 3: **Не реализован**: читается как '0'

**Биты управления направлением данных PORTE**

- бит 2: **BIT2**: Направление вывода RE2/-CS/AN7  
1 = вход  
0 = выход
- бит 1: **BIT1**: Направление вывода RE1/-WR/AN6  
1 = вход  
0 = выход
- бит 0: **BIT0**: Направление вывода RE0/-RD/AN5  
1 = вход  
0 = выход

**Таблица 3-10** Регистры и биты, связанные с работой PORTE

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
09h	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	-	Рег. напр. данных PORTE			0000 -111	0000 -111
9Fh	ADCON1	ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000

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



### 3.6 Ведомый параллельный порт

Ведомый параллельный порт (PSP) не реализован в микроконтроллерах PIC16F873 и PIC16F876.

PORTD работает как 8-разрядный параллельный порт (или порт микропроцессора), когда бит PSPMODE(TRISE<4>) установлен в '1'. В режиме ведомого данные асинхронно читаются или записываются внешними сигналами -RD (RE0/-RD) или -WR(RE1/-WR) соответственно.

Внешний микропроцессор может читать/записывать данные в PORTD. Операции чтения/записи выполняются при низком логическом уровне сигналов на входах -RD, -WR и низком логическом уровне сигнала на входе выбора микросхемы -CS. Биты TRISE (TRISE<2:0>) должны быть установлены в '1'. В регистре ADCON1 выводы RE2:RE0 должны быть настроены как цифровые каналы ввода/вывода (биты PCFG3:PCFG0).

Фактически существуют два 8-разрядных регистра: один регистр для приема данных, другой - для передачи. Пользователь записывает 8-разрядные данные в выходную защелку PORTD, а читает данные со входной защелки (обратите внимание, выходная и входная защелки имеют один и тот же адрес). В этом режиме значение битов регистра TRISE игнорируется, т.к. направлением данных управляет внешнее устройство.

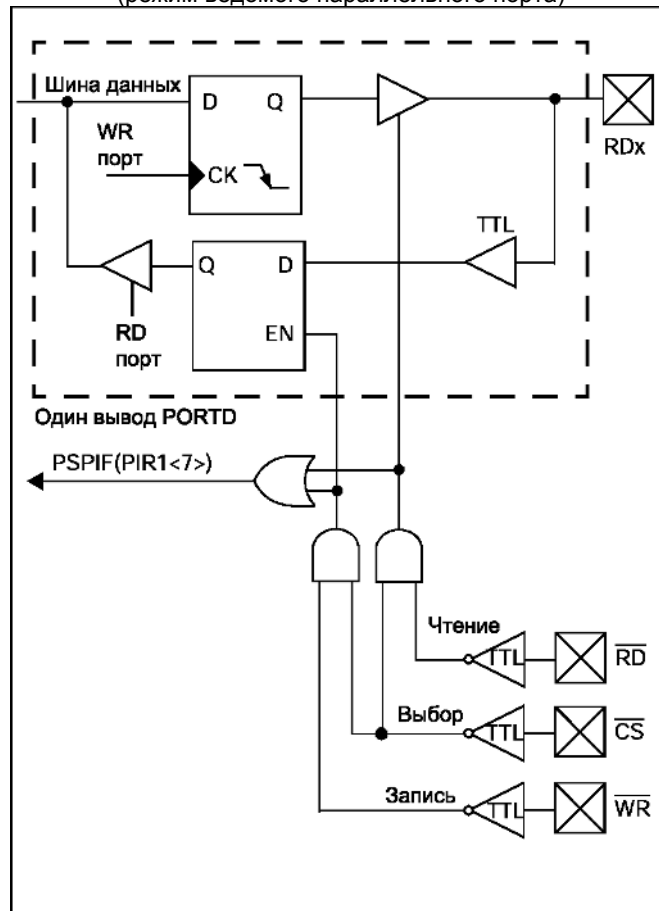
Запись в PSP происходит, если выходы -CS и -WR имеют низкий уровень сигнала. После перехода сигнала на выводе -CS или -WR в высокий уровень данные сохраняются во входной защелке на такте Q2 (см. рисунок 3-10). На такте Q4 устанавливаются в '1' бит IBF(TRISE<7>) и флаг прерываний PSPIF(PIR<7>). Бит IBF может быть сброшен в '0' только чтением регистра PORTD. Бит переполнения IBOV(TRISE<5>) устанавливается в '1', если произошла следующая запись в PSP, а предыдущий байт не был прочитан.

Чтение из PSP происходит, если выходы -CS и -RD имеют низкий уровень сигнала. Немедленно сбрасывается в '0' бит OBF(TRISE<6>), указывающий, что PORTD ожидает чтение внешней шиной (см. рисунок 3-11). После перехода сигнала на выводе -CS или -RD в высокий уровень устанавливается флаг прерывания PSPIF на такте Q4 (только после такта Q2), указывая, что чтение завершено. Бит OBF остается сброшенным в '0' пока не будут загружены новые данные в PORTD.

Когда режим PSP выключен, биты IBF и OBF равняются нулю, а предварительно установленный в '1' бит IBOV должен быть сброшен программно.

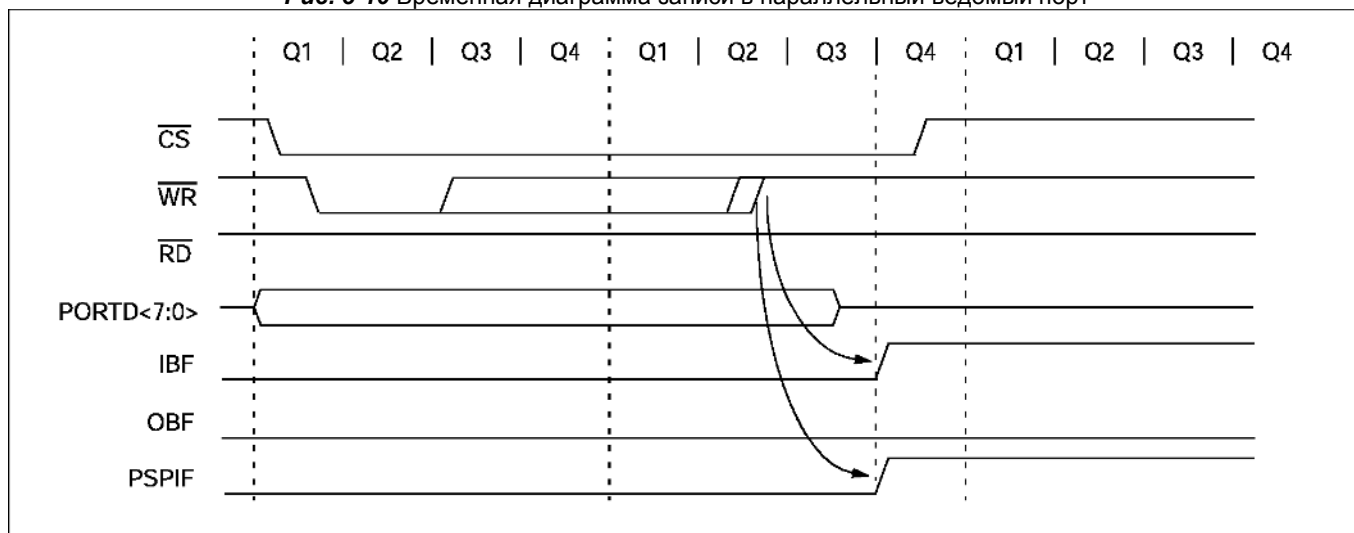
Флаг прерывания PSPIF устанавливается в '1' по завершению каждой операции чтения или записи (сбрасывается в '0' программно). Разрешить/запретить прерывания от модуля PSP можно установкой/сбросом бита PSPIE (PIE1<7>).

Рис. 3-9 Структурная схема выводов PORTD и PORTE (режим ведомого параллельного порта)

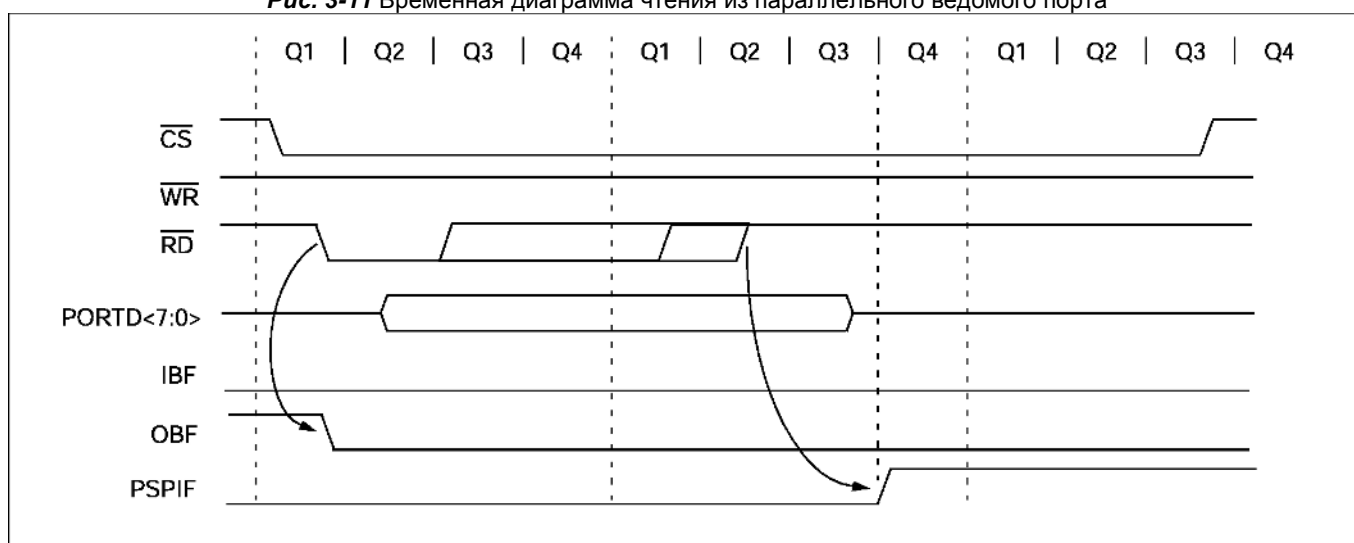


Примечание. Выводы портов имеют защитные диоды, подключенные к  $V_{DD}$  и  $V_{SS}$ .

**Рис. 3-10** Временная диаграмма записи в параллельный ведомый порт



**Рис. 3-11** Временная диаграмма чтения из параллельного ведомого порта



**Таблица 3-11** Регистры и биты, связанные с работой ведомого параллельного порта

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
08h	PORTD	Запись в выходную защелку PORTD, чтение состояния выводов PORTD								xxxx xxxx	uuuu uuuu
09h	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	-	Рег. напр. данных PORTE			0000 -111	0000 -111
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
9Fh	ADCON1	ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000

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

**Примечание\***. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

## 4.0 EEPROM память данных и FLASH память программ

Данные из EEPROM памяти и FLASH памяти программ могут быть прочитаны/перезаписаны в нормальном режиме работы микроконтроллера во всем диапазоне напряжения питания  $V_{DD}$ . Операции выполняются с одним байтом для EEPROM памяти данных и одним словом для FLASH памяти программ. Запись производится по принципу "стирание - запись" для каждого байта или слова. Сформированная кодом программы операция стирания не может выполнена при включенной защите записи.

Доступ к памяти программ позволяет выполнить вычисление контрольной суммы. Данные, записанные в памяти программ, могут использоваться в виде: 14-разрядных чисел, калибровочной информации, серийных номеров, упакованных 7-разрядных символов ASCII и т.д. В случае обнаружения недействительной команды в памяти программ, выполняется пустой цикл NOP.

Число циклов стирания/записи смотрите в разделе электрических характеристик (для EEPROM памяти данных параметр D120, для FLASH памяти программ - D130). Число циклов стирания/записи для FLASH памяти программ значительно ниже по сравнению с EEPROM памятью данных, поэтому EEPROM память данных должна использоваться для сохранения часто изменяемых данных. Время записи данных управляется внутренним таймером, оно зависит от напряжения питания, температуры и имеет небольшой технологический "разброс" (смотрите в разделе электрических характеристик параметры D122, D133).

При записи байта или слова автоматически стирается соответствующая ячейка, а затем выполняется запись. Запись в EEPROM память программ не воздействует на выполнение программы, а при записи во FLASH память программ выполнение программы останавливается на время записи. Нельзя обратиться к памяти программ во время цикла записи. В течение операции записи тактовый генератор продолжает работать, периферийные модули включены и генерируют прерывания, которые "ставятся в очередь" до завершения цикла записи. После завершения записи выполняется загруженная команда (из-за конвейерной обработки) и происходит переход по вектору прерываний, если прерывание разрешено и условие прерывания возникло во время записи.

Доступ к функциям записи/чтения EEPROM памяти данных и FLASH памяти программ выполняется через шесть регистров специального назначения:

- EEDATA;
- EEDATH;
- EEADR;
- EEADRH;
- EECON1;
- EECON2.

Операции чтения/записи EEPROM памяти данных не приостанавливают выполнение программы. В регистре EEADR сохраняется адрес ячейки EEPROM памяти данных. Данные сохраняются/читаются из регистра EEDATA. В микроконтроллерах PIC16F873/874 объем EEPROM памяти 128 байт. Старший бит адреса в регистре EEADR всегда должен равняться нулю, потому что не поддерживается циклическая адресация (т.е. ячейка с адресом 0x80 не отображается на 0x00). В PIC16F876/877 объем EEPROM памяти данных 256 байт (используются все 8-разрядов регистра EEADR).

Чтение FLASH памяти программ не влияет на выполнение программы, а во время операции записи выполнение программы приостановлено. В спаренных регистрах EEADRH:EEADR сохраняется 13-разрядный адрес ячейки памяти программ, к которой необходимо сделать обращение. Спаренные регистры EEADRH:EEADR содержат 14-разрядные данные для записи или отображают значение из памяти программ при чтении. Также как для EEPROM памяти данных, в регистры EEADRH:EEADR должен быть загружен адрес физически реализованной памяти программ (для PIC16F873/874 от 0000h до 0FFFh, для PIC16F876/877 от 0000h до 1FFFh), потому что циклическая адресация не поддерживается (т.е. ячейка с адресом 4000h не отображается на 0000h для PIC16F877).

### 4.1 Регистры EECON1, EECON2

Регистр EECON1 содержит биты управления косвенной записи/чтения EEPROM памяти данных и FLASH памяти программ. Регистр EECON2 физически не реализован, он используется только при операциях записи с целью предотвращения случайной записи.

Значение бита EEPGD в регистре EECON1 определяет тип памяти, к которой будет выполнено обращение. Если EEPGD = 0, то операции относятся к EEPROM памяти данных. Когда EEPGD=1, обращение происходит к FLASH памяти программ.

В операции чтения используется только один дополнительный бит RD, инициализирующий операцию чтения из указанной памяти. Установив бит RD в '1', значение ячейки памяти будет доступно в регистре данных. Бит RD не может быть сброшен программно в '0', он сбрасывается автоматически по окончании операции чтения. При чтении из EEPROM памяти данные будут доступны в регистре EEDATA в следующем машинном цикле, после установки бита RD. При чтении из FLASH памяти программ данные будут доступны в регистре EEDATA:EEDATH на втором машинном цикле, поле установки бита RD.

В операции записи используется два служебных бита WR, WREN и два бита статуса WRERR, EEIF. Бит WREN применяется для разрешения/запрещения операции записи (WREN = 0 - операция записи запрещена). Перед выполнением записи бит WREN необходимо установить в '1'. Бит WR предназначен для инициализации записи, он аппаратно сбрасывается в '0' по завершении операции записи. Флаг прерывания EEIF устанавливается в '1' по завершении записи. Этот флаг должен быть сброшен программно в '0' перед установкой бита WR.

Для EEPROM памяти данных:

После установки битов WREN, WR в '1' стирается, указанная в регистре EEADR, ячейка EEPROM памяти, а затем происходит запись данных из регистра EEDATA. Операция записи сопровождается выполнением кода программы. По завершении записи устанавливается флаг прерывания EEIF в '1'.

Для FLASH памяти программ:

После установки битов WREN, WR в '1' микроконтроллер приостанавливает выполнение программы. Стирается ячейка памяти программ, указанная в регистре EEADRH:EEADR, затем происходит запись данных из регистра EEDATH:EEDATA. По завершении записи устанавливается флаг прерывания EEIF в '1', а микроконтроллер продолжит выполнять код программы.

Бит WRERR указывает, что произошел сброс микроконтроллера PIC16F87X во время выполнения операции записи. Бит WRERR устанавливается в '1', если во время выполнения записи данных произошел сброс по сигналу -MCLR или по переполнению сторожевого таймера WDT в нормальном режиме. Проверив состояние бита WREER, пользователь может повторить запись (регистры EEDATA и EEADR не изменяют своего значения). Содержимое регистров данных, адреса и бит EEPGD не изменяется после сброса по сигналу -MCLR, по переполнению сторожевого таймера WDT в нормальном режиме.

**Регистр EECON1 (адрес 18Ch)**

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
<b>EEPGD</b>	-	-	-	<b>WRERR</b>	<b>WREN</b>	<b>WR</b>	<b>RD</b>
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: **EEPGD:** Бит выбора EEPROM память данных/FLASH память программ  
1 = FLASH память программ  
0 = EEPROM память данных  
(бит не должен изменяться во время выполнения операции чтения/записи)

биты 6-4:**Не используются:** читаются как '0'

бит 3: **WRERR:** Флаг ошибки записи данных  
1 = запись прервана (произошел один из сбросов: по сигналу -MCLR, по переполнению WDT в нормальном режиме)  
0 = запись завершена

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

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

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

## 4.2 Чтение из EEPROM памяти данных

Для чтения из EEPROM памяти необходимо только записать адрес в регистр EEADR и сбросить бит EEPGD в '0'. После установки в '1' бита RD данные будут доступны в регистре EEDATA на следующем машинном цикле. Данные в регистре EEDATA сохраняются до выполнения следующей операции чтения или записи в EEDATA.

Рекомендованная последовательность действий при чтении из EEPROM памяти данных:

1. Записать адрес в регистр EEADR. Проверьте, что записанный адрес корректен для данного типа микроконтроллера.
2. Сбросить в '0' бит EEPGD для обращения к EEPROM памяти данных.
3. Инициализировать операцию чтения установкой бита RD в '1'.
4. Прочитать данные из регистра EEDATA.

**Пример 4-1** Чтение из EEPROM памяти данных

```
BSF     STATUS,RP1      ;
BCF     STATUS,RP0      ; Выбрать банк 2
MOVF    ADDR,W          ; Записать адрес
MOVWF   EEADR           ; ячейки
BSF     STATUS,RP0      ; Выбрать банк 3
BCF     EECON1,EEPGD    ; Выбрать EEPROM память
BSF     EECON1,RD       ; Инициализировать чтение
BCF     STATUS,RP0      ; Выбрать банк 2
MOVF    EEDATA,W        ; W = EEDATA
```

## 4.3 Запись в EEPROM память данных

Запись данных в EEPROM память несколько сложнее чтения. Адрес ячейки EEPROM памяти и записываемые данные должны быть помещены в соответствующие регистры специального назначения, бит EEPGD сбрасывается в '0'. Бит WREN должен всегда равняться нулю, кроме непосредственной записи в EEPROM память. Бит WR может быть установлен в '1' только, если бит WREN был установлен в предыдущих командах, т.е. биты WR, WREN не могут устанавливаться в '1' одной командой. Бит WREN должен быть сброшен в '0' после инициализации записи (на процесс записи он не влияет).

Перед записью в EEPROM память должна быть выполнена обязательная последовательность команд, предотвращающая случайную запись. Обязательная последовательность выполняется при выключенных прерываниях.

Рекомендованная последовательность действий при записи в EEPROM память:

1. Если шаг 10 не был выполнен, то необходимо проверить, что не происходит запись (WR=0).
2. Записать адрес в регистр EEADR. Проверьте, что записанный адрес корректен для данного типа микроконтроллера.
3. Записать 8-разрядное значение в регистр EEDATA.
4. Сбросить в '0' бит EEPGD для обращения к EEPROM памяти данных.
5. Установить бит WREN в '1', разрешив запись в EEPROM память.
6. Запретить прерывания, если они разрешены.
7. Выполнить обязательную последовательность из пяти команд:
  - Запись значения 55h в регистр EECON2 (две команды, сначала в W затем в EECON2);
  - Запись значения AAh в регистр EECON2 (две команды, сначала в W затем в EECON2);
  - Установить бит WR в '1'.
8. Разрешить прерывания (если необходимо).
9. Сбросить бит WREN в '0'.
10. После завершения цикла записи сбрасывается в '0' бит WR, устанавливается в '1' флаг прерывания EEIF (сбрасывается программно). Если шаг 1 не выполняется, то необходимо проверить состояние битов EEIF, WR перед началом записи.

**Пример 4-2** Запись в EEPROM память данных

```
BSF     STATUS, RP1      ;
BSF     STATUS, RP0      ; Выбрать банк 3
BTFSC   EECON1, WR       ; Проверить завершения
GOTO    $-1              ; операции записи
BCF     STATUS, RP0      ; Выбрать банк 2
MOVF    ADDR, W          ; Указать адрес ячейки
MOVWF   EEADR           ;
MOVF    VALUE, W         ; Указать данные для записи
MOVWF   EEDATA          ;
BSF     STATUS, RP0      ; Выбрать банк 3
BCF     EECON1, EEPGD    ; Выбрать EEPROM память данных
BSF     EECON1, WREN     ; Разрешить запись в EEPROM память данных
BCF     INTCON, GIE      ; Запретить прерывания
MOVLW   0x55             ; Записать 55h в регистр EECON2
MOVWF   EECON2          ;
MOVLW   0xAA             ; Записать AAh в регистр EECON2
MOVWF   EECON2          ;
BSF     EECON1, WR       ; Инициализировать запись
BSF     INTCON, GIE      ; Разрешить прерывания
BCF     EECON1, WREN     ; Запретить запись в EEPROM память данных
```

#### 4.4 Чтение из FLASH памяти программ

Чтение из FLASH памяти программ очень похоже на процедуру чтения из EEPROM памяти данных, только необходимо выполнить две инструкции NOP после установки бита RD в '1'. Два пустых цикла NOP используются микроконтроллером для чтения данных из FLASH памяти программ и сохранения их в регистрах EEDATH:EEDATA. Данные в регистрах будут доступны после выполнения второй инструкции NOP. Данные в регистре EEDATH:EEDATA сохраняются до выполнения следующей операции чтения или записи в EEDATH:EEDATA.

Рекомендованная последовательность действий при чтении из FLASH памяти программ:

1. Записать адрес в регистры EEADRH:EEADR. Проверьте, что записанный адрес корректен для данного типа микроконтроллера.
2. Установить в '1' бит EEPGD для обращения к FLASH памяти программ.
3. Инициализировать операцию чтения установкой бита RD в '1'.
4. Выполнить две команды NOP, чтобы позволить микроконтроллеру произвести чтение из FLASH памяти программ.
5. Прочитать данные из регистров EEDATH:EEDATA.

**Пример 4-3** Чтение из FLASH памяти программ

```
BSF      STATUS,RP1      ;
BCF      STATUS,RP0      ; Выбрать банк 2
MOVF     ADDR, W         ; Записать адрес
MOVWF    EEADR           ; ячейки памяти программ
MOVF     ADDR, W         ;
MOVWF    EEADRH         ;

BSF      STATUS,RP0      ; Выбрать банк 3
BSF      EECON1,EEPGD    ; Выбрать FLASH память программ
BSF      EECON1,RD       ; Инициализировать чтение
NOP      ;
NOP      ; Две инструкции NOP
BCF      STATUS,RP0      ; Выбрать банк 2
MOVF     EEDATA, W       ; DATAL = EEDATA
MOVWF    DATAL           ;
MOVF     EEDATH, W       ; DATAH = EEDATH
MOVWF    DATAH         ;
```

#### 4.5 Запись во FLASH память программ

В течение операции записи выполнение программы останавливается, тактовый генератор продолжает работать, периферийные модули включены и генерируют прерывания, которые "ставятся в очередь" до завершения цикла записи. После завершения записи (параметр D133) микроконтроллер продолжает выполнять код программы с места остановки. Другим существенным отличием записи во FLASH память программ является наличие бита защиты WRT в слове конфигурации, предотвращающий любую запись в память программ (см. таблицу 4-1).

Запись данных во FLASH память программ несколько сложнее чтения. Адрес ячейки памяти программ и записываемые данные должны быть помещены в соответствующие регистры специального назначения, бит EEPG устанавливается в '1'. Бит WREN должен всегда равняться нулю, кроме непосредственной записи во FLASH память программ. Бит WR может быть установлен в '1' только, если бит WREN был установлен в предыдущих командах, т.е. биты WR, WREN не могут устанавливаться в '1' одной командой. Бит WREN должен быть сброшен программно в '0' после инициализации записи (на процесс записи он не влияет).

Перед записью во FLASH память программ должна быть выполнена обязательная последовательность команд, предотвращающая случайную запись. Обязательная последовательность выполняется при выключенных прерываниях. После обязательной последовательности должны размещаться две инструкции NOP, позволяющие микроконтроллеру произвести запись. Выполнение программы после записи начинается с инструкции, расположенной после двух команд NOP.

Рекомендованная последовательность действий при записи во FLASH память программ:

1. Записать адрес в регистры EEADRH:EEADR. Проверьте, что записанный адрес корректен для данного типа микроконтроллера.
2. Записать 14-разрядное значение в регистры EEDATH:EEDATA.
3. Установить в '1' бит EEPGD для обращения к FLASH памяти программ.
4. Установить бит WREN в '1', разрешив запись во FLASH память программ.
5. Запретить прерывания, если они разрешены.
6. Выполнить обязательную последовательность из пяти команд:
  - Запись значения 55h в регистр EECON2 (две команды, сначала в W затем в EECON2);
  - Запись значения AAh в регистр EECON2 (две команды, сначала в W затем в EECON2);
  - Установить бит WR в '1'.
7. Выполнить две команды NOP, чтобы позволить микроконтроллеру произвести запись во FLASH память программ.
8. Разрешить прерывания (если необходимо).
9. Сбросить бит WREN в '0'.

По завершении операции записи аппаратно сбрасывается в '0' бит WR и устанавливается в '1' флаг прерывания EEIF (флаг EEIF сбрасывается в '0' программно). Для определения окончания операции записи проверять бит WR и EEIF не обязательно, т.к. микроконтроллер не выполняет программу во время записи во FLASH память программ.

**Пример 4-4** Запись во FLASH память программ

```
BSF      STATUS,RP1      ;
BCF      STATUS,RP0      ; Выбрать банк 2
MOVF     ADDR,L,W        ; Записать адрес
MOVWF    EEADR           ; ячейки памяти программ
MOVF     ADDR,H,W        ;
MOVWF    EEADRH         ;
MOVF     VALUE,L,W       ; Значение записываемое
MOVWF    EEDATA         ; в память программ
MOVF     VALUE,H,W       ;
MOVWF    EEDATH         ;
BSF      STATUS, RP0     ; Выбрать банк 3
BSF      EECON1, EEPGD   ; Выбрать FLASH память программ
BSF      EECON1, WREN    ; Разрешить запись во FLASH память программ
BCF      INTCON, GIE     ; Запретить прерывания
MOVLW   0x55             ; Записать 55h в регистр EECON2
MOVWF    EECON2         ;
MOVLW   0xAA             ; Записать AAh в регистр EECON2
MOVWF    EECON2         ;
BSF      EECON1, WR      ; Инициализировать запись
NOP      ;
NOP      ;
BSF      INTCON, GIE     ; Разрешить прерывания
BCF      EECON1, WREN    ; Запретить запись во FLASH память программ
```

#### 4.6 Проверка записи

В микроконтроллерах PIC16F87X аппаратно не проверяется значение, сохраненное при записи. Рекомендуется проверять фактически записанное значение контрольным чтением. Особенно проверку записи необходимо выполнять при возможном исчерпании гарантированного числа циклов стирания/записи.

#### 4.7 Защита от случайной записи

Существуют условия при которых запись данных в EEPROM память или во FLASH память программ не будет выполнена. В микроконтроллерах PIC16F87X предусмотрены различные виды защиты от случайной записи. При включении питания сбрасывается в '0' бит WREN и во время счета таймера по включению питания PWRT (если он включен) запись запрещена. Обязательная последовательность команд, бит WREN предотвращают случайную запись при сбросе микроконтроллера или сбое программы.

#### 4.8 Операции при включенной защите кода

В PIC16F87X предусмотрена отдельная защита EEPROM памяти данных (один бит в слове конфигурации) и FLASH памяти программ (два бита в слове конфигурации). Данные могут быть свободно записаны/прочитаны из EEPROM памяти данных при любом значении бита CPD. Когда защита EEPROM памяти данных включена (CPD=0), чтение запрещено внешним протоколом ICSP вне зависимости от состояния битов защиты памяти программ. Это предотвращает чтение EEPROM памяти данных микроконтроллера.

Состояние битов CP0, CP1 нельзя изменить командами микроконтроллера. Биты защиты не влияют на операцию внутреннего чтения памяти программ. Однако состояние битов защиты и бита WRT будут создавать различные условия записи в память программ (см. таблицу 4-1).

Установленная защита памяти программ или EEPROM памяти данных может быть выключена только полным стиранием памяти микроконтроллера.

#### 4.9 Защита записи FLASH памяти программ

В слове конфигурации размещен бит защиты записи во FLASH память программ командами микроконтроллера. Состояние этого бита может быть изменено только в режиме программирования микроконтроллеров PIC16F87X по протоколу ICSP. Эта защита может быть выключена только полным стиранием памяти микроконтроллера. Защита записи не влияет на операцию чтения FLASH памяти программ.

**Таблица 4-1** Доступность операций чтения/записи FLASH памяти программ в зависимости от состояния битов защиты

Биты конфигурации			Область памяти	Внутреннее чтение	Внутренняя запись	ICSP чтение	ICSP запись
CP1	CP0	WRT					
0	0	x	Вся память программ	Есть	Нет	Нет	Нет
0	1	0	Незащищенная область	Есть	Нет	Есть	Нет
0	1	0	Защищенная область	Есть	Нет	Нет	Нет
0	1	1	Незащищенная область	Есть	Есть	Есть	Нет
0	1	1	Защищенная область	Есть	Нет	Нет	Нет
1	0	0	Незащищенная область	Есть	Нет	Есть	Нет
1	0	0	Защищенная область	Есть	Нет	Нет	Нет
1	0	1	Незащищенная область	Есть	Есть	Есть	Нет
1	0	1	Защищенная область	Есть	Нет	Нет	Нет
1	1	0	Вся память программ	Есть	Нет	Есть	Есть
1	1	1	Вся память программ	Есть	Есть	Есть	Есть

**Таблица 4-2** Регистры и биты, связанные с обращением к EEPROM памяти данных / FLASH памяти программ

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Dh	PIR2	-	(1)	-	EEIF	BCLIF	-	-	CCP2IF	-r-0 0--0	-r-0 0--0
8Dh	PIE2	-	(1)	-	EEIE	BCLIE	-	-	CCP2IE	-r-0 0--0	-r-0 0--0
10Dh	EEADR	Регистр адреса, младший байт								xxxx xxxx	uuuu uuuu
10Fh	EEADRH	-	-	-	Регистр адреса, старший байт					xxxx xxxx	uuuu uuuu
10Ch	EEDATA	Регистр данных, младший байт								xxxx xxxx	uuuu uuuu
10Eh	EEDATH	-	-	Регистр данных, старший байт					xxxx xxxx	uuuu uuuu	
18Ch	EECON1	EEPGD	-	-	-	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	Регистр управления 2 (физически не реализован)								-	-

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

Затененные биты на работу не влияют.

**Примечание 1.** Резервные биты. При обращении всегда должны равняться нулю.



## 5.0 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 и общего с WDT предделителя показана на рисунке 5-1.

Дополнительную информацию по работе модуля TMR0 смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual".

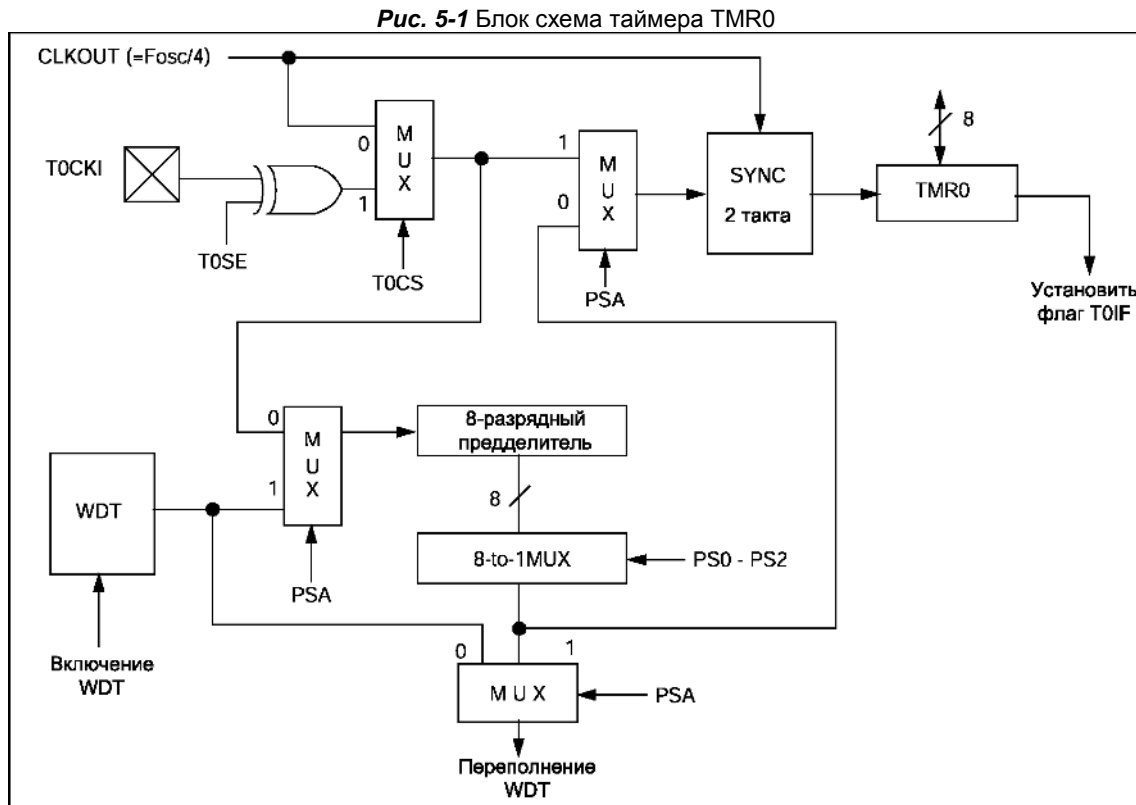
Когда бит T0CS сброшен в '0' (OPTION\_REG<5>), TMR0 работает от внутреннего тактового сигнала. Приращение счетчика TMR0 происходит в каждом машинном цикле (если предделитель отключен). После записи в TMR0 приращение счетчика запрещено два следующих цикла. Пользователь должен скорректировать эту задержку перед записью нового значения в TMR0.

Если бит T0CS установлен в '1' (OPTION\_REG<5>), TMR0 работает от внешнего источника тактового сигнала с входа RA4/T0CKI. Активный фронт внешнего тактового сигнала выбирается битом T0SE в регистре OPTION\_REG<4> (T0SE=0 – активным является передний фронт сигнала). Работа модуля TMR0 с внешним источником тактового сигнала будет рассмотрена в разделе 5.2.

Предделитель может быть включен перед WDT или TMR0, в зависимости от состояния бита PSA (OPTION\_REG<3>). Нельзя прочитать или записать новое значение в предделитель. Подробное описание работы с предделителем смотрите в разделе 5.3.

### 5.1 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 (INTCON<2>). Само прерывание может быть разрешено/запрещено установкой/сбросом бита T0IE (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.



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

## 5.2 Использование внешнего источника тактового сигнала для TMR0

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. 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.

## 5.3 Предделитель

8-разрядный счетчик может работать как предделитель TMR0 или выходной делитель WDT. 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\_REG<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 CLRWDТ сбросит предделитель вместе с WDT. Предделитель также очищается при сбросе микроконтроллера. Предделитель недоступен для чтения/записи.

**Примечание.** Запись в регистр TMR0 сбросит предделитель, если он подключен к TMR0, но не изменит его режима работы.

### Регистр OPTION\_REG (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:**

бит 6: **INTEDG:**

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

Таблица 5-1 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, BOR	Другие сбросы
01h,101h	TMR0	Регистр таймера 0								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	-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.  
Затененные биты на работу не влияют.

## 6.0 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:

- Режим таймера;
- Режим счетчика.

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 8.0).

Когда включен генератор тактовых импульсов (T1OSCEN=1), выходы RC1/T1OSI/CCP2 и RC0/T1OSO/T1CKI настроены как входы. Значение битов TRISC<1:0> игнорируется, а чтение данных с этих выводов дает результат '0'.

Управляющие биты TMR1 находятся в регистре T1CON.

Дополнительную информацию по работе модуля TMR1 смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual".

### 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	
-	-	<b>T1CKPS1</b>	<b>T1CKPS0</b>	<b>T1OSCEN</b>	<b>-T1SYNC</b>	<b>TMR1CS</b>	<b>TMR1ON</b>	
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'

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

бит 3: **T1OSCEN:** Включение тактового генератора TMR1  
1 = генератор включен  
0 = генератор выключен (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 RC0/T1OSO/T1CKI (активным является передний фронт сигнала)  
0 = внутренний источник Fosc/4

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

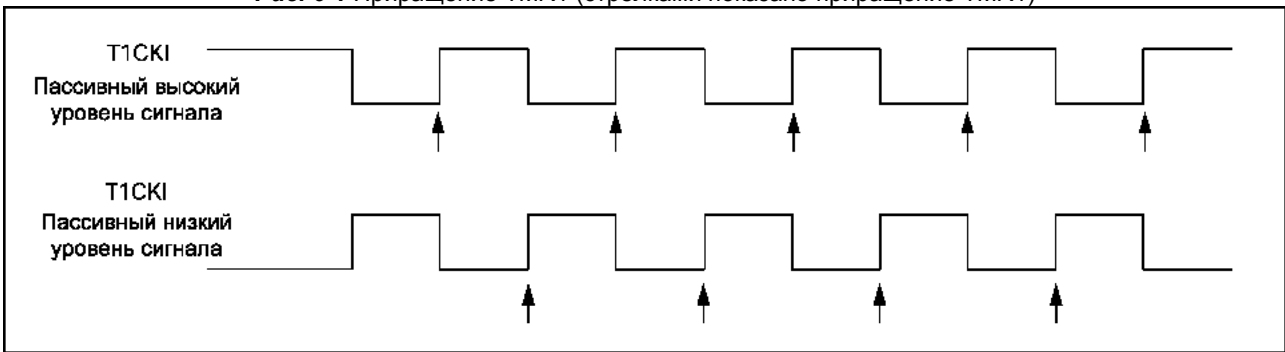
### 6.1 Работа TMR1 в режиме таймера

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.

### 6.2 Работа TMR1 в режиме счетчика

TMR1 может работать в синхронном или асинхронном режиме в зависимости от состояния бита TMR1CS. Когда TMR1 использует внешний тактовый сигнал, приращение таймера происходит по переднему фронту. Включив TMR1 в режим внешнего тактового сигнала счет начнется только после появления заднего фронта.

Рис. 6-1 Приращение TMR1 (стрелками показано приращение TMR1)



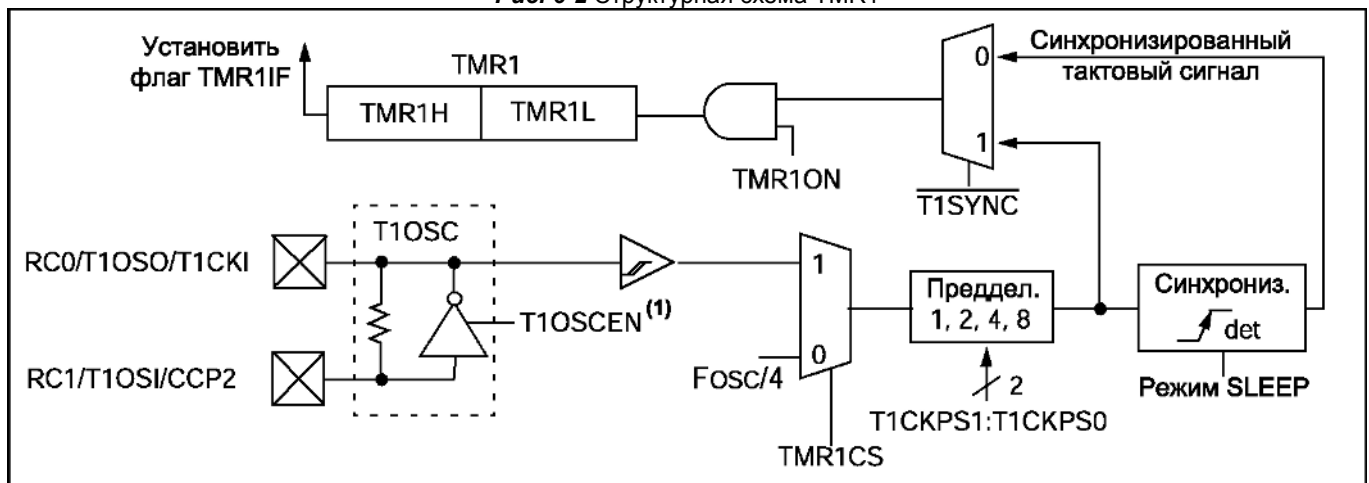
### 6.3 Работа TMR1 в режиме синхронного счетчика

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 RC1/T1OSI/CCP2 (если T1OSCEN=1) или RC0/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).

Рис. 6-2 Структурная схема TMR1



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

## 6.4 Работа TMR1 в режиме асинхронного счетчика

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 6.4.1).

В этом режиме TMR1 не может использоваться для захвата/сравнения данных модуля CCP.

### 6.4.1 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.

В примерах 12-2, 12-3 документации DS33023 "PICmicro™ Mid-Range Reference Manual" представлена рекомендованная последовательность операций чтения/записи 16-разрядного значения TMR1 в асинхронном режиме.

## 6.5 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 таблице 6-1 указаны рекомендуемые значения конденсаторов для генератора TMR1.

**Таблица 6-1** 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
<b>Ориентировочные значения</b>			

Протестированные резонаторы:		
32.768кГц	Epson C-001 R32.768K-A	±20 PPM
100кГц	Epson C-2 100.00 KC-P	±20 PPM
200кГц	STD XTL 200.000 kHz	±20 PPM

**Примечания:**

1. Большая емкость увеличивает стабильность генератора, но также увеличивает время запуска.
2. Каждый резонатор имеет собственные характеристики. Проконсультируйтесь у производителя резонаторов для правильного подбора внешних компонентов.

## 6.6 Сброс TMR1 триггером модуля CCP

Если модуль CCP1 или CCP2 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 CCPRxH:CCPRxL.

## 6.7 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 CCP1 или CCP2.

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.

## 6.8 Предделитель TMR1

Предделитель TMR1 очищается при записи в регистр TMR1L или TMR1H.

Таблица 6-2 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, BOR	Другие сбросы
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
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.

Затененные биты на работу не влияют.

**Примечание\***. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

## 7.0 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>).

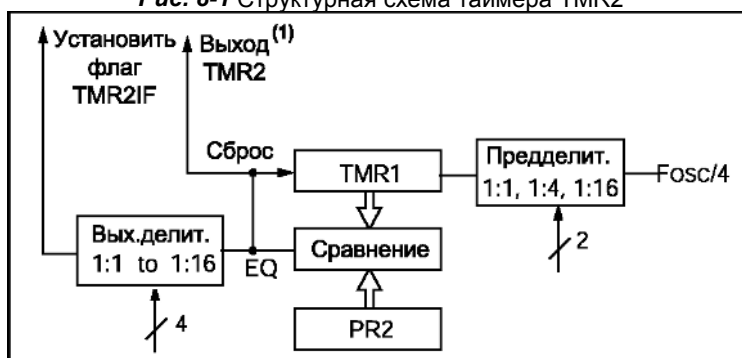
TMR2 считает, инкрементируя от 00h до значения в регистре PR2, затем сбрасывается в 00h на следующем машинном цикле. Регистр PR2 доступен для записи и чтения. После сброса значение регистра PR2 равно FFh.

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'.

Дополнительную информацию по работе модуля TMR2 смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual".

Рис. 8-1 Структурная схема таймера TMR2



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

### 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

## 7.1 Пределитель и выходной делитель TMR2

Счетчик пределителя 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.

## 7.2 Сигнал TMR2

Сигнал переполнения TMR2 (до выходного пределителя) поступает в модуль SSP для управления скоростью передачи данных.

**Таблица 7-1** 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, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
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.

Затененные биты на работу не влияют.

**Примечание\***. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.



## 8.0 CCP модуль (захват/сравнение/ШИМ)

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

- 16-разрядного регистра захвата данных;
- 16-разрядного регистра сравнения;
- Двух 8-разрядных (ведущий и ведомый) регистров ШИМ.

Работа модулей CCP1 и CCP2 идентична, за исключением функционирования триггера специального события. В таблице 8-1 и 8-2 указаны ресурсы, используемые модулем CCP. Далее будет описана работа модуля CCP1. Модуль CCP2 работает аналогично, отличия будут указаны отдельно.

### Модуль CCP1:

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. В режиме сравнения триггер специального события сбрасывает таймер TMR1.

### Модуль CCP2:

Регистр CCPR2 модуля CCP состоит из двух 8-разрядных регистров: CCPR2L (младший байт), CCPR2H (старший байт). В регистре CCP2CON находятся управляющие биты модуля CCP2, доступные для записи и чтения. В режиме сравнения триггер специального события сбрасывает таймер TMR1 и запускает преобразование АЦП (если АЦП включено).

Дополнительную информацию по работе модулей CCP смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual" и AN594 "Use the CCP Modules".

**Таблица 8-1** Использование таймеров модулями CCP

Режим модуля CCP	Таймер
Захват	TMR1
Сравнение	TMR1
ШИМ	TMR2

**Таблица 8-2** Взаимодействие двух модулей CCP

Режим CCPx	Режим CCPy	Взаимодействие
Захват	Захват	Базовый таймер TMR1
Захват	Сравнение	Модуль CCP, работающий в режиме сравнения, должен сбрасывать таймер TMR1 триггером специального события.
Сравнение	Сравнение	Модули CCP, работающие в режиме сравнения, должны сбрасывать таймер TMR1 триггером специального события.
ШИМ	ШИМ	Оба ШИМ имеют одинаковую частоту и фазу (базовый таймер TMR2)
ШИМ	Захват	Нет
ШИМ	Сравнение	Нет

## Регистр CCPxCON (адрес 17h/1Dh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
-	-	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
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: **CCPxX:CCPxY:** Младшие биты скважности ШИМ

Режим захвата  
Не используются

Режим сравнения  
Не используются

Режим ШИМ  
Два младших бита скважности. Восемь старших находятся в CCPxL.

биты 3-0: **CCPxM3:CCPxM0:** Режим работы модуля CCPx

- 0000 = модуль CCPx выключен (сброс модуля CCPx)
- 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 CCPxIF в '1')
- 1001 = сравнение, сбрасывает выходной сигнал (устанавливается флаг CCPxIF в '1')
- 1010 = сравнение, на выходной сигнал не влияет (устанавливается флаг CCPxIF в '1')
- 1011 = сравнение, триггер специальных функций (устанавливается флаг CCPxIF в '1'; на вывод CCPx не влияет). CCP1 - сброс таймера TMR1. CCP2 - сброс таймера TMR1, запуск преобразования АЦП (если АЦП включено).
- 11xx = ШИМ режим

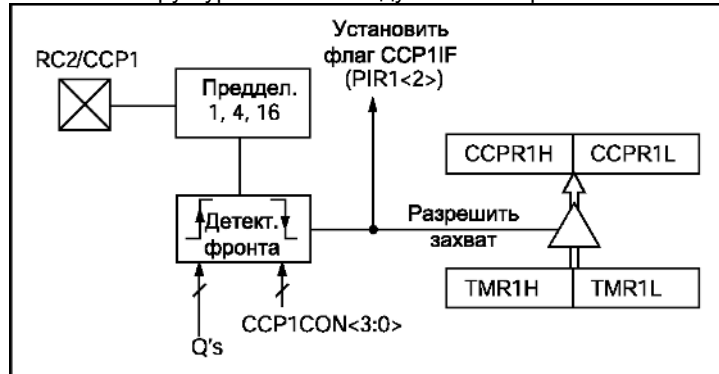
## 8.1 Режим захвата

При возникновении события захвата 16-разрядное значение счетчика TMR1 переписывается в регистры CCPR1L:CCPR1H модуля CCP1. Событием захвата может быть:

- Каждый задний фронт сигнала на входе RC2/CCP1;
- Каждый передний фронт сигнала на входе RC2/CCP1;
- Каждый 4-й передний фронт сигнала на входе RC2/CCP1;
- Каждый 16-й передний фронт сигнала на входе RC2/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. 8-1 Структурная схема модуля CCP1 в режиме захвата



### 8.1.1 Настройка вывода модуля CCP

Порт ввода/вывода RC2/CCP1 должен быть настроен на вход установкой бита TRISC<2> в '1'.

**Примечание.** Если порт ввода/вывода RC2/CCP1 настроен на выход, то захват может происходить командой из программы.

### 8.1.2 Настройка таймера TMR1

TMR1 должен работать в синхронизированном режиме, в случае использования внешнего тактового сигнала с вывода RC1/T1OSI/CCP2. В асинхронном режиме TMR1 модуль CCP1 работать не будет.

### 8.1.3 Обработка прерываний

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

### 8.1.4 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. Любой сброс микроконтроллера очищает счетчик событий.

В примере 8-1 показано как нужно производить переключение типа события, чтобы не вызвать ложное прерывание.

**Пример 8-1** 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
```

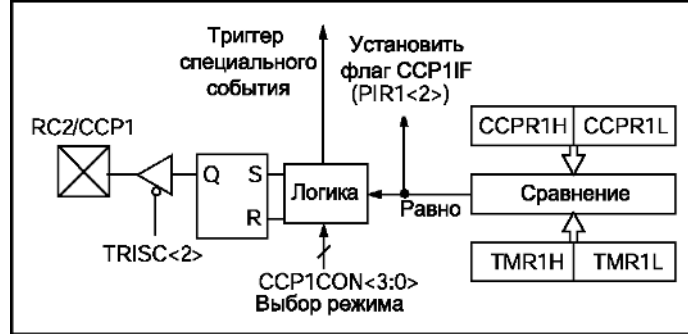
## 8.2 Режим сравнения

В этом режиме 16-разрядный регистр CCP1 сравнивается со значением TMR1. Как только значения в регистрах становятся одинаковыми, модуль CCP1 изменяет состояние вывода RC2/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 изменения состояния вывода устанавливается флаг прерывания CCP1IF в '1'.

Рис. 8-2 Структурная схема модуля CCP1 в режиме сравнения



### 8.2.1 Настройка вывода модуля CCP

Для изменения состояния вывода RC2/CCP1, он должен быть настроен на выход сбросом бита TRISC<2> в '0'.

**Примечание.** При очистке регистра CCP1CON на выводе RC2/CCP1 появится сигнал низкого уровня, что не является результатом сравнения или данными из выходной защелки PORTC.

### 8.2.2 Настройка таймера TMR1

TMR1 должен работать в синхронизированном режиме, в случае использования внешнего тактового сигнала с вывода RC1/T1OSI/CCP2. В асинхронном режиме TMR1 модуль CCP1 работать не будет.

### 8.2.3 Обработка прерываний

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

### 8.2.4 Триггер специального события

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

Триггер специального события CCP1 сбрасывает значения таймера TMR1 при каждом положительно выполненном сравнении. Регистр CCP1R является 16-разрядным программируемым регистром периода для TMR1.

Триггер специального события CCP2 сбрасывает значения таймера TMR1 и запускает преобразование АЦП (если модуль АЦП включен).

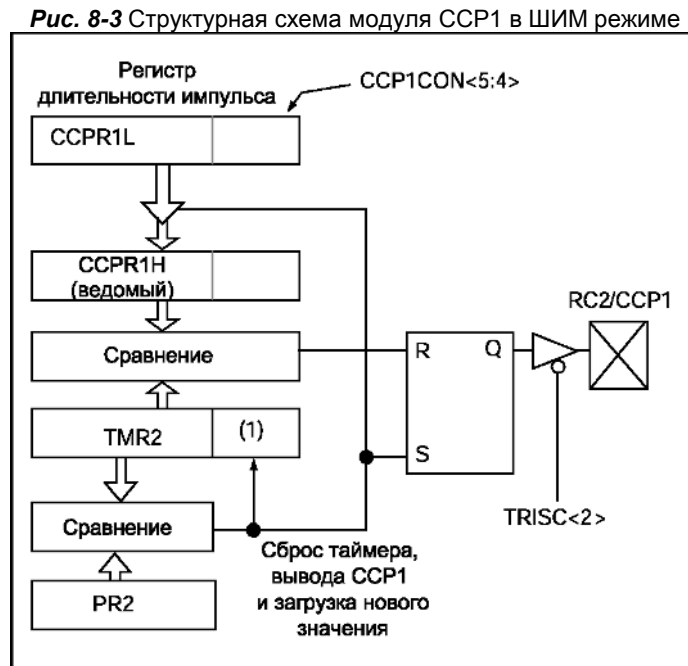
**Примечание.** Триггер специального события модулей CCP1 и CCP2 не устанавливает флаг прерывания TMR1IF (PIR1<0>) в '1'.

### 8.3 Режим ШИМ

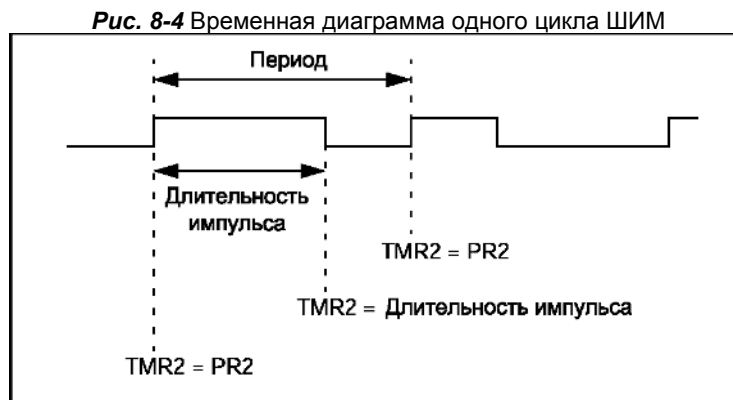
В ШИМ режиме модуля CCP1 вывод RC2/CCP1 используется в качестве выхода 10-разрядного ШИМ. Т.к. вход CCP1 мультиплицирован с цифровым каналом порта ввода/вывода, бит направления TRISC<2> должен быть сброшен в '0'.

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

На рисунке 8-3 показана структурная схема модуля CCP1 в ШИМ режиме. Пошаговое описание настройки модуля CCP1 в ШИМ режиме смотрите в разделе 8.3.3.



На рисунке 8-4 показана временная диаграмма одного цикла ШИМ (период ШИМ и длительность высокого уровня сигнала). Частота ШИМ есть обратная величина периоду (1/период).



#### 8.3.1 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 7.1) не влияет на частоту ШИМ. Он может использоваться для отсчета времени, когда необходимо изменить скважность ШИМ.

### 8.3.2 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> (CCPR1L:CCPCON<5:4>). Для вычисления длительности сигнала высокого уровня, воспользуйтесь следующей формулой:

$$\text{Длительность импульса ШИМ} = (\text{CCPR1L:CCPCON<5:4>}) \times T_{osc} \times (\text{коэффициент делителя TMR2})$$

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):

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

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

### 8.3.3 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 8-3** 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 8-4** Регистры и биты, связанные с работой модуля CCP в режиме захват/сравнение и модуля TMR1

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
0Dh	PIR2	-	(1)	-	EEIF	BCLIF	-	-	CCP2IF	-r-0 0--0	-r-0 0--0
8Dh	PIE2	-	(1)	-	EEIE	BCLIE	-	-	CCP2IE	-r-0 0--0	-r-0 0--0
87h	TRISC	Регистр направления PORTC								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 CCP1								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Старший байт захвата/сравнения/ШИМ CCP1								xxxx xxxx	uuuu uuuu
17h	CCP1CON	-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Младший байт захвата/сравнения/ШИМ CCP2								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Старший байт захвата/сравнения/ШИМ CCP2								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	-	-	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

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

Затененные биты на работу не влияют.

**Примечание\***. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

**Примечание 1**. Резервные биты. При обращении всегда должны равняться нулю.

**Таблица 8-5** Регистры и биты, связанные с работой модуля CCP в режиме ШИМ и модуля TMR2

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
0Dh	PIR2	-	(1)	-	EEIF	BCLIF	-	-	CCP2IF	-r-0 0--0	-r-0 0--0
8Dh	PIE2	-	(1)	-	EEIE	BCLIE	-	-	CCP2IE	-r-0 0--0	-r-0 0--0
87h	TRISC	Регистр направления PORTC								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	CCPR1L	Младший байт захвата/сравнения/ШИМ CCP1								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Старший байт захвата/сравнения/ШИМ CCP1								xxxx xxxx	uuuu uuuu
17h	CCP1CON	-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Младший байт захвата/сравнения/ШИМ CCP2								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Старший байт захвата/сравнения/ШИМ CCP2								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	-	-	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

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

Затененные биты на работу не влияют.

**Примечание\***. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

**Примечание 1**. Резервные биты. При обращении всегда должны равняться нулю.

## 9.0 Модуль ведущего синхронного последовательного порта (MSSP)

Модуль ведущего синхронного последовательного порта (MSSP) может использоваться для связи с периферийными микросхемами или другими микроконтроллерами. Периферийными микросхемами могут быть: EEPROM память, сдвиговые регистры, драйверы ЖКИ, АЦП и др. Модуль MSSP может работать в одном из двух режимах:

- Последовательный периферийный интерфейс (SPI);
- Inter-Integrated Circuit (I<sup>2</sup>C).

На рисунке 9-1 показана структурная схема модуля MSSP в режиме SPI, а на рисунках 9-5, 9-9 в двух разных режимах I<sup>2</sup>C.

Дополнительно смотрите техническую документацию AN734 "Using the PICmicro SSP for Slave I<sup>2</sup>C™ Communication" и AN735 "Using the PICmicro MSSP Module for I<sup>2</sup>C™ Communication".



**SSPSTAT (адрес 94h) Регистр статуса модуля MSSP**

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
<b>SMP</b>	<b>CKE</b>	<b>DI-A</b>	<b>P</b>	<b>S</b>	<b>R/-W</b>	<b>UA</b>	<b>BF</b>
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: **SMP:** Фаза выборки бита  
Ведущий режим SPI  
1 = опрос входа в конце периода вывода данных  
0 = опрос входа в середине периода вывода данных  
  
Ведомый режим SPI  
Для режима ведомого SPI этот бит всегда должен быть сброшен в '0'  
  
Ведущий или ведомый режим I<sup>2</sup>C  
1 = управление длительностью фронта выключено в стандартном режиме (100кГц и 1МГц)  
0 = управление длительностью фронта включено в скоростном режиме (400кГц)

бит 6: **CKE:** Выбор фронта тактового сигнала (см. рис. 9-2, 9-3 и 9-4)  
SPI режим, CKP=0  
1 = данные передаются по переднему фронту сигнала на выводе SCK  
0 = данные передаются по заднему фронту сигнала на выводе SCK  
  
SPI режим, CKP=1  
1 = данные передаются по заднему фронту сигнала на выводе SCK  
0 = данные передаются по переднему фронту сигнала на выводе SCK  
  
Ведущий или ведомый режим I<sup>2</sup>C  
1 = входные уровни соответствуют спецификации SMBus  
0 = входные уровни соответствуют спецификации I<sup>2</sup>C

бит 5: **DI-A:** Бит Данные/Адрес (только для режима I<sup>2</sup>C)  
1 = последний принятый или переданный байт является информационным  
0 = последний принятый или переданный байт является адресным

бит 4: **P:** Бит STOP (только для режима I<sup>2</sup>C)  
Этот бит сбрасывается в '0' когда модуль MSSP выключен, SSPEN=0.  
1 = указывает, что бит STOP был обнаружен последним (этот бит равен '0' после сброса)  
0 = бит STOP не является последним

бит 3: **S:** Бит START (только для режима I<sup>2</sup>C)  
Этот бит сбрасывается в '0' когда модуль MSSP выключен, SSPEN=0.  
1 = указывает, что бит START был обнаружен последним (этот бит равен '0' после сброса)  
0 = бит START не является последним

бит 2: **R/-W:** Бит чтения/записи (только для режима I<sup>2</sup>C)  
Значение бита действительно только после совпадения адреса и до приема бита START, STOP или -ACK.  
Ведомый режим I<sup>2</sup>C  
1 = чтение  
0 = запись  
  
Ведущий режим I<sup>2</sup>C  
1 = выполняется передача данных  
0 = передачи данных не происходит  
Логическое ИЛИ этого бита с битами SEN, RSEN, PEN, RCEN или ACKEN укажет на неактивное состояние модуля MSSP.

бит 1: **UA:** Флаг обновления адреса устройства (только для режима 10-разрядного I<sup>2</sup>C)  
1 = необходимо обновить адрес в регистре SSPADD  
0 = обновление адреса не требуется

бит 0: **BF:** Бит статуса буфера  
Прием (SPI и I<sup>2</sup>C режимы)  
1 = прием завершен, буфер SSPBUF полон  
0 = прием не завершен, буфер SSPBUF пуст  
  
Передача (только I<sup>2</sup>C режима)  
1 = выполняется передача данных (исключая биты -ACK и STOP), буфер SSPBUF полон  
0 = передача данных завершена (исключая биты -ACK и STOP), буфер SSPBUF пуст

**SSPCON (адрес 14h) Регистр управления модуля MSSP**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
<b>WCOL</b>	<b>SSPOV</b>	<b>SSPEN</b>	<b>СКР</b>	<b>SSPM3</b>	<b>SSPM2</b>	<b>SSPM1</b>	<b>SSPM0</b>	
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: **WCOL:** Бит конфликта записи

Ведущий режим  
1 = запись в SSPBUF была выполнена при не выполнении условий шины I<sup>2</sup>C  
0 = конфликта не было

Ведомый режим  
1 = была предпринята попытка записи в SSPBUF во время передачи предыдущего байта  
0 = конфликта не было

бит 6: **SSPOV:** Бит переполнения приемника

SPI режим  
1 = принят новый байт в то время как SSPBUF содержит предыдущие данные(байт в SSPSR будет потерян). В ведомом режиме пользователь должен прочитать содержимое регистра SSPBUF даже, если только передает данные. В ведущем режиме бит в '1' не устанавливается, т.к. каждая операция инициализируется записью в SSPBUF. (сбрасывается в '0' программно)  
0 = нет переполнения

I<sup>2</sup>C режим  
1 = принят новый байт в то время как SSPBUF содержит предыдущие данные. Значение бита не действительно при передаче данных. (сбрасывается в '0' программно)  
0 = нет переполнения

бит 5: **SSPEN:** Бит включения модуля MSSP

Когда модуль включен, соответствующие порты ввода/вывода настраиваются на выход или вход

SPI режим  
1 = модуль MSSP включен, выходы SCK, SDO, SDI, -SS используются модулем MSSP  
0 = модуль MSSP выключен, выходы работают как цифровые порты ввода/вывода

I<sup>2</sup>C режим  
1 = модуль MSSP включен, выходы SDA, SCL используются модулем MSSP  
0 = модуль MSSP выключен, выходы работают как цифровые порты ввода/вывода

бит 4: **СКР:** Бит выбора полярности тактового сигнала

SPI режим  
1 = пассивный высокий уровень сигнала  
0 = пассивный низкий уровень сигнала

Ведомый режим I<sup>2</sup>C  
Управление тактовым сигналом SCK  
1 = не управлять тактовым сигналом  
0 = удерживать тактовый сигнал в низком логическом уровне (используется для подготовки данных)

Ведущий режим I<sup>2</sup>C  
Не имеет значения

биты 3-0: **SSPM3:SSPM0:** Режим работы модуля MSSP

0000 = ведущий режим SPI, тактовый сигнал =  $F_{osc}/4$   
0001 = ведущий режим SPI, тактовый сигнал =  $F_{osc}/16$   
0010 = ведущий режим SPI, тактовый сигнал =  $F_{osc}/64$   
0011 = ведущий режим SPI, тактовый сигнал = выход TMR2 / 2  
0100 = ведомый режим SPI, тактовый сигнал с вывода SCK. Вывод -SS подключен к MSSP  
0101 = ведомый режим SPI, тактовый сигнал с вывода SCK. Вывод -SS не подключен к MSSP  
0110 = ведомый режим I<sup>2</sup>C, 7-разрядная адресация  
0111 = ведомый режим I<sup>2</sup>C, 10-разрядная адресация  
1000 = ведущий режим I<sup>2</sup>C, тактовый сигнал =  $F_{osc}/(4 * (SSPADD+1))$   
1011 = программная поддержка ведущего режима I<sup>2</sup>C (ведомый режим выключен)  
1110 = программная поддержка ведущего режима I<sup>2</sup>C, 7-разрядная адресация с разрешением прерываний по приему бит START и STOP  
1111 = программная поддержка ведущего режима I<sup>2</sup>C, 10- разрядная адресация с разрешением прерываний по приему бит START и STOP  
1001, 1010, 1100, 1101 = резерв

**SSPCON2 (адрес 91h) Регистр управления модуля MSSP**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>GCEN</b>	<b>ACKSTAT</b>	<b>ACKDT</b>	<b>ACKEN</b>	<b>RCEN</b>	<b>PEN</b>	<b>RSEN</b>	<b>SEN</b>
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: **GCEN:** Бит разрешения поддержки общего вызова (только для ведомого режима I<sup>2</sup>C)  
1 = разрешить прерывания при приеме в регистр SSPSR адреса общего вызова (0000h)  
0 = поддержка общего вызова выключена

бит 6: **ACKSTAT:** Бит статуса подтверждения (только для ведущего режима I<sup>2</sup>C)  
Передача ведущего I<sup>2</sup>C  
1 = подтверждения не было получено от ведомого  
0 = подтверждение от ведомого было получено

бит 5: **ACKDT:** Бит подтверждения (только для ведущего режима I<sup>2</sup>C)  
Прием ведущего I<sup>2</sup>C  
Значение этого бита передается при разрешении формирования бита подтверждения.  
1 = подтверждение  
0 = нет подтверждения

бит 4: **ACKEN:** Сформировать бит подтверждения (только для ведущего режима I<sup>2</sup>C)  
1 = на выводах SCL, SDA формируется бит ACKDT. Аппаратно сбрасывается в '0'  
0 = подтверждение не формируется

бит 3: **RCEN:** Разрешить прием данных (только для ведущего режима I<sup>2</sup>C)  
1 = разрешить прием данных с шины I<sup>2</sup>C  
0 = приемник выключен

бит 2: **PEN:** Сформировать бит STOP (только для ведущего режима I<sup>2</sup>C)  
1 = на выводах SCL, SDA формируется бит STOP. Аппаратно сбрасывается в '0'  
0 = бит STOP не формируется

бит 1: **RSEN:** Сформировать бит повторный START (только для ведущего режима I<sup>2</sup>C)  
1 = на выводах SCL, SDA формируется бит повторный START. Аппаратно сбрасывается в '0'  
0 = бит повторный START не формируется

бит 0: **SEN:** Сформировать бит START (только для ведущего режима I<sup>2</sup>C)  
1 = на выводах SCL, SDA формируется бит START. Аппаратно сбрасывается в '0'  
0 = бит START не формируется

**Примечание.** Для битов ACKEN, RCEN, PEN, RSEN, SEN. Если I<sup>2</sup>C модуль не находится в пассивном состоянии, то ни один из битов не может быть установлен в '1' (поставлен в очередь), не может быть выполнена запись в регистр SSPBUF (или запись в регистр SSPBUF заблокирована).

## 9.1 Режим SPI

В SPI режиме возможен одновременный синхронный прием/переда 8-разрядных данных. Модуль MSSP поддерживает четыре режима SPI с типовым использованием трех выводов микроконтроллера:

- Вход последовательных данных (SDI);
- Выход последовательных данных (SDO);
- Тактовый сигнал (SCK).

Дополнительно может быть задействован четвертый вывод для работы в режиме ведомого.

- Выбор ведомого (-SS).

При инициализации SPI необходимо определить параметры работы битами SSPCON<5:0>, SSPSTAT<7:6>. Управляющие биты определяют следующие параметры работы:

- Ведущий режим (SCK выход);
- Ведомый режим (SCK вход);
- Полярность тактового сигнала (пассивный уровень SCK);
- Фаза выборки входных данных (в середине или конце передачи бита);
- Активный фронт тактового сигнала (передний, задний);
- Частота тактового сигнала (только в ведущем режиме);
- Режим выбора ведомого (только в режиме ведомого).

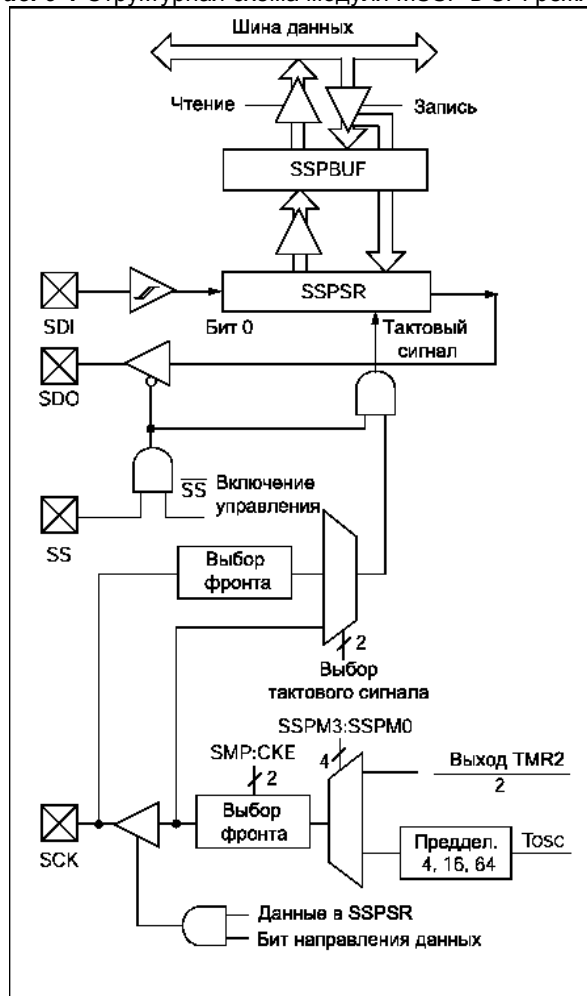
На рисунке 9-1 показана структурная схема модуля MSSP в SPI режиме.

Для включения модуля MSSP необходимо установить бит SSPEN (SSPCON<5>) в '1'. Для сброса или перенастройки режима SPI рекомендуется сбросить бит SSPEN в '0', выполнить изменения параметров работы, а затем вновь установить бит SSPEN в '1'. После включения MSSP в режиме SPI выводы SDI, SDO, SCK, -SS используются последовательным портом. Для корректной работы последовательного порта биты регистров TRIS должны быть настроены следующим образом:

- SDI, автоматически управляется SPI модулем;
- SDO, бит TRISC<5> должен быть сброшен в '0';
- SCK (ведущий режим), бит TRISC<3> = 0;
- SCK (ведомый режим), бит TRISC<3> = 1;
- -SS, бит TRISA<5> = 1, в регистре ADCON1 вывод RA5 должен быть настроен как цифровой порт ввода/вывода (см. раздел 11.0).

Любая нежелательная функция последовательного порта может быть выключена, настраивая соответствующие биты регистров направления данных TRIS.

Рис. 9-1 Структурная схема модуля MSSP в SPI режиме



### 9.1.1 Режим ведущего SPI

Ведущий шины может инициализировать передачу данных в любой момент, поскольку он генерирует тактовый сигнал, и определяет когда ведомый должен передать данные в соответствии с используемым протоколом (см. рисунок 9-2).

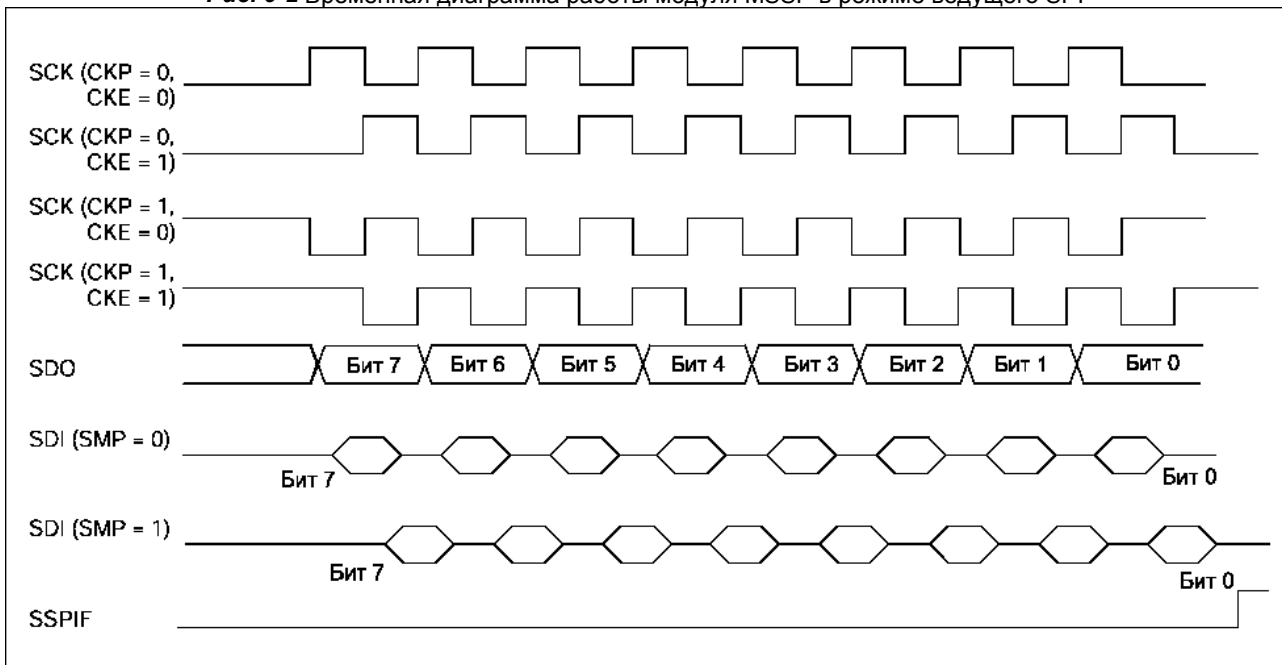
В режиме ведомого данные передаются/принимаются после их записи/чтения из регистра SSPBUF. Если в SPI режиме требуется только принимать данные, вывод SDO может быть заблокирован (настроен как вход). Данные с вывода SDI последовательно сдвигаются в регистр SSPSR с установленной скоростью. Каждый принятый байт загружается в регистр SSPBUF (как нормально полученный байт) с формированием прерываний и воздействием на соответствующие биты статуса. Эта функция может быть полезна при реализации "монитора шины".

Полярность тактового сигнала устанавливается битом CKP (SSPCON<4>), что позволяет получить различные методы передачи данных (см. рисунок 9-2). Данные всегда передаются старшим битом вперед. В ведущем режиме частота тактового сигнала выбирается программно:

- $F_{osc}/4$  (или  $T_{CY}$ );
- $F_{osc}/16$  (или  $4 \times T_{CY}$ );
- $F_{osc}/64$  (или  $16 \times T_{CY}$ );
- Выход таймера TMR2 / 2.
- Максимальная частота передачи данных 5МГц при тактовой частоте микроконтроллера 20МГц.

Бит СKE определяет по какому фронту тактового сигнала необходимо выполнять прием данных. Параметры выборки входных данных устанавливаются битом SMP. Поле загрузки принятых данных в регистр SSPBUF устанавливается флаг прерываний SSPIF в '1'.

**Рис. 9-2** Временная диаграмма работы модуля MSSP в режиме ведущего SPI



### 9.1.2 Режим ведомого SPI

В режиме ведомого данные передаются/принимаются по внешнему тактовому сигналу на выводе SCK. Когда принимается последний бит байта, устанавливается в '1' флаг прерываний SSPIF (PIR1<3>). Внешний тактовый сигнал должен удовлетворять требованиям длительности низкого и высокого логического уровня, описанным в разделе электрических характеристик.

В SLEEP режиме микроконтроллера ведомый может принимать/передавать данные. После приема данных микроконтроллер выходит из режима SLEEP.

#### Примечания:

1. В режиме ведомого SPI с поддержкой выбора ведомого по сигналу на выводе -SS (SSPCON<3:0>=0100), SPI модуль сброшен, если на выводе -SS напряжение питания  $V_{DD}$ .
2. В режиме ведомого SPI и CKE = 1, необходимо разрешить управление с вывода -SS.

Рис. 9-3 Временная диаграмма работы модуля MSSP в режиме ведомого SPI (CKE=0)

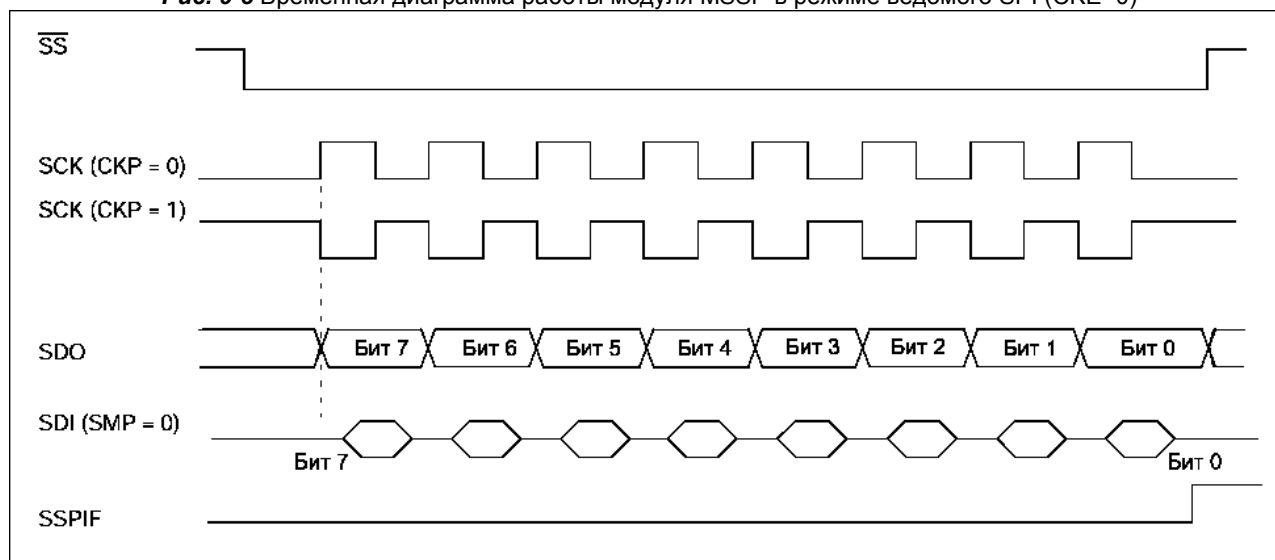


Рис. 9-4 Временная диаграмма работы модуля MSSP в режиме ведомого SPI (CKE=1)

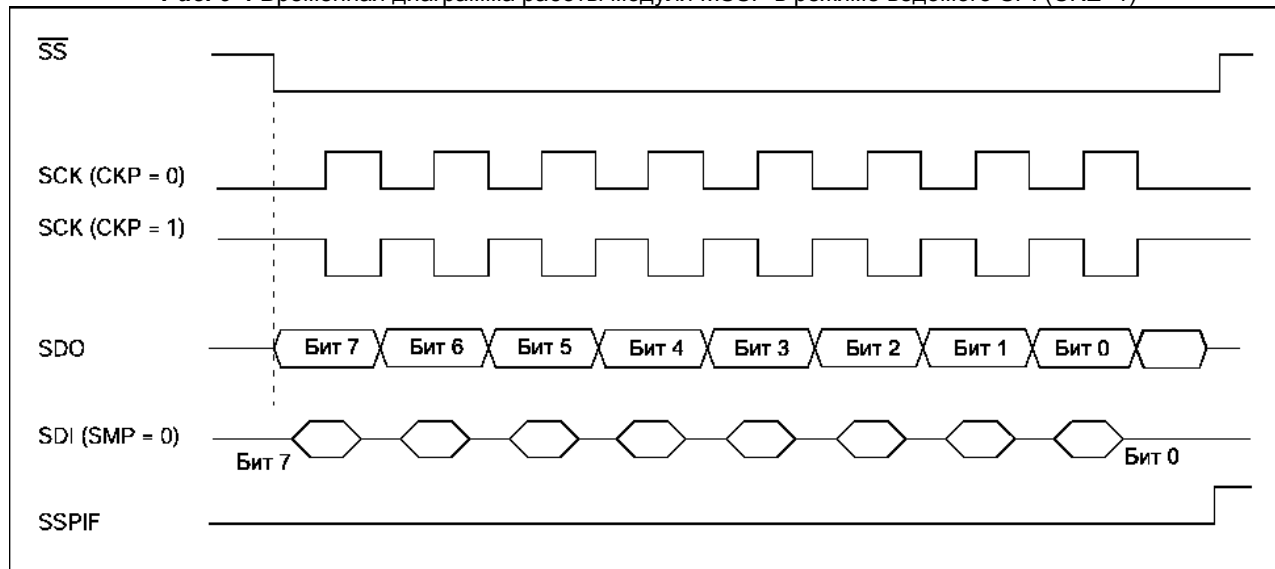


Таблица 9-1 Регистры и биты, связанные с работой модуля MSSP в режиме SPI

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	Буфер приемника MSSP / регистр передатчика								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000

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

**Примечание\*.** Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

## 9.2 Режим I<sup>2</sup>C

Модуль MSSP полностью поддерживает все функции ведущих и ведомых устройств, включая поддержку общего вызова, аппаратные прерывания по детектированию битов START и STOP для определения занятости шины I<sup>2</sup>C в режиме ведущего (при конкуренции на шине). В MSSP модуле реализована поддержка стандартного режима 7, 10-разрядной адресации.

Дополнительно смотрите техническую документацию AN578 "Use of the SSP Module in the I<sup>2</sup>C Multi-Master Environment".

Фильтр "glitch" подключен к выводам SDA и SCL, когда они настроены на вход. Фильтр работает в режимах 100кГц и 400кГц. В режиме 100кГц, когда выводы SDA и SCL настроены на выход, фильтр контролирует длительность формируемых сигналов в не зависимости от тактовой частоты микроконтроллера.

Для работы с шиной I<sup>2</sup>C используется два вывода SCL (сигнал синхронизации) и SDA (данные). Выводы SDA и SCL автоматически настраиваются при включении режима I<sup>2</sup>C. Включение модуля MSSP выполняется установкой бита SSPEN (SSPCON<5>) в '1'.

Для управления модулем MSSP в режиме I<sup>2</sup>C используется шесть регистров:

- SSPCON, регистр управления MSSP;
- SSPCON, регистр управления 2 MSSP;
- SSPSTAT, регистр статуса MSSP;
- SSPBUF, буфер приемника/передатчика;
- SSPSR, сдвиговый регистр (пользователю не доступен);
- SSPADD, регистр адреса.

В регистре SSPCON устанавливается требуемый режим I<sup>2</sup>C. С помощью четырех битов (SSPCON<3:0>) можно выбрать один из режимов I<sup>2</sup>C:

- Ведомый режим I<sup>2</sup>C, 7-разрядная адресация;
- Ведомый режим I<sup>2</sup>C, 10-разрядная адресация;
- Ведущий режим I<sup>2</sup>C, тактовый сигнал =  $F_{osc}/(4 * (SSPADD+1))$ ;
- Программная поддержка ведущего режима I<sup>2</sup>C (реализовано для совместимости с другими PICmicro).

При выборе любого режима I<sup>2</sup>C выводы SCL и SDA должны быть настроены на вход, установкой соответствующих битов регистра TRISC в '1'. После выбора режима I<sup>2</sup>C и установки бита SSPEN в '1' выводы SDA, SCL подключаются к модулю MSSP. Для нормальной работы модуля I<sup>2</sup>C к выводам SCL, SDA должны быть подключены внешние подтягивающие резисторы.

Бит SKE (SSPSTAT<6>) устанавливает уровни сигналов на выводах SCL, SDA в ведущем и ведомом режимах. Если SKE =1, то выходные уровни соответствуют спецификации SMBus. Когда SKE=0, выходные уровни соответствуют спецификации I<sup>2</sup>C.

Регистр SSPSTAT содержит биты статуса передачи данных: обнаружение на шине битов START (S) или STOP (P), флаг приема байта данных или адреса, указатель загрузки старшего байта 10-разрядного адреса, бит операции приема/передачи.

В регистре SSPBUF загружаются данные для передачи по шине I<sup>2</sup>C, и из него читаются принятые данные. Регистр SSPSR выполняет сдвиг принимаемых/передаваемых данных. При приеме данных регистры SSPBUF, SSPSR работают как двухуровневый буфер приемника. Буфер позволяет принимать следующий байт до чтения предыдущего принятого байта из регистра SSPBUF. Когда байт полностью загружен в SSPSR, он передается в регистр SSPBUF и устанавливается флаг прерывания SPIF в '1'. Если полностью принят следующий байт до чтения предыдущего байта из SSPBUF, то устанавливается бит SSPOV (SSPCON<6>) в '1', а байт в регистре SSPSR будет потерян.

В регистр SSPADD записывается адрес ведомого устройства. В 10-разрядном режиме пользователь должен сначала записывать старший байт адреса (1111 0 A9 A8 0). После соответствия старшего байта адреса необходимо загрузить младший байт адреса (A7:A0).

### 9.2.1 Режим ведомого I<sup>2</sup>C

В режиме ведомого I<sup>2</sup>C выводы SCL, SDA должны быть настроены на вход. Модуль MSSP автоматически изменит направление вывода SDA при передачи данных ведомым. Структурная схема модуля MSSP в режиме ведомого I<sup>2</sup>C показана на рисунке 9-5.

При совпадении адреса или после приема байта данных (если предварительно совпал адрес) аппаратно генерируется бит подтверждения (-ACK), а затем данные из регистра SSPSR загружаются в SSPBUF.

Существует несколько условий, при которых бит -ACK не формируется (эти условия могут возникать одновременно):

- Бит BF (SSPSTAT<0>) = 1 перед приемом данных;
- Бит переполнения SSPOV (SSPSTAT<6>) = 1 перед приемом данных.

Если бит BF = 1, то значение из SSPSR не переписывается в регистр SSPBUF, а биты SSPIF и SSPOV устанавливаются в '1'. В таблице 9-2 показаны операции после приема байта при различных значениях битов BF, SSPOV. В затененных ячейках показана ситуация, когда вовремя не был сброшен бит переполнения SSPOV в '0'. Заметьте, что бит BF аппаратно сбрасывается в '0' при чтении из регистра SSPBUF, а бит SSPOV необходимо сбрасывать в '0' программно.

Минимальная длительность логических уровней входного сигнала синхронизации SCL должна удовлетворять требованиям раздела электрических характеристик (см. параметры 100 и 101).

Рис. 9-5 Структурная схема модуля MSSP в режиме ведомого I<sup>2</sup>C

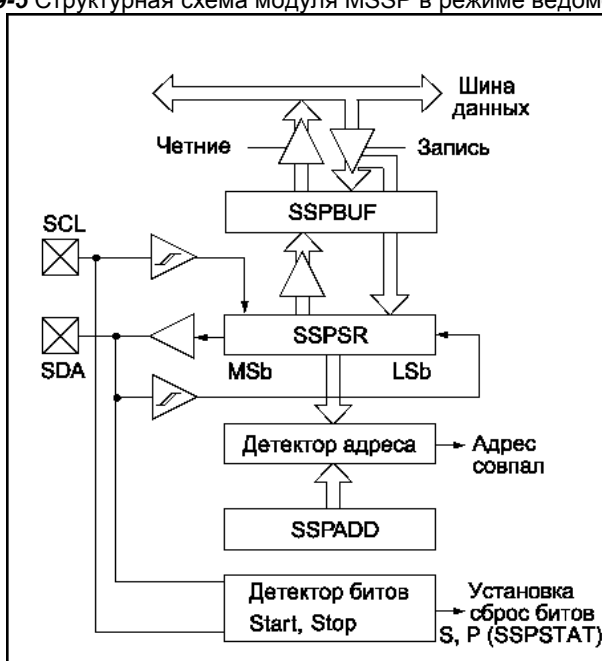


Таблица 9-2 Операции после приема байта при различных значениях битов BF, SSPOV

Биты статуса приемника		Запись из SSPSR в SSPBUF	Формирование бита -ACK	Установка флага прерываний SSPIF
BF	SSPOV			
0	0	Есть	Есть	Есть
1	0	Нет	Нет	Есть
1	1	Нет	Нет	Есть
0	1	Есть	Нет	Есть



### 9.2.1.1 Адресация

После включения модуля MSSP ожидается формирование на шине бита START. Получив бит START, принимается 8 бит в сдвиговый регистр SSPSR. Выборка битов происходит по переднему фронту синхронизирующего сигнала на выводе SCL. По заднему фронту восьмого такта сигнала SCL значение в регистре SSPSR<7:1> сравнивается с содержимым регистра SSPADD. Если значение адреса совпадает, а биты BF и SSPOV равны нулю, то выполняются следующие действия:

- Значение регистра SSPSR загружается SSPBUF по 8-му заднему фронту сигнала SCL;
- Устанавливается флаг BF в '1' (буфер полон) по 8-му заднему фронту сигнала SCL;
- Генерируется бит -ACK;
- Устанавливается флаг прерываний SSPIF (PIR1<3>) в '1' по 9-му заднему фронту сигнала SCL.

В режиме ведомого при 10-разрядной адресации необходимо принять два байта адреса. Пять старших бит первого байта определяют: является ли полученный байт первым байтом 10-разрядного адреса. Бит R/W(SSPSTAT<2>) должен быть настроен для приема второго байта адреса. Для 10-разрядной адресации первый байт адреса должен иметь формат '1111 0 A9 A8 0', где A9:A8 два старших бита адреса. Рекомендуемая последовательность действий при 10-разрядной адресации (шаги 7-9 для передачи ведомым):

- Принять старший байт адреса (устанавливаются биты SSPIF, BF и UA (SSPSTAT<1> в '1')).
- Записать младший байт адреса в регистр SSPADD (аппаратно сбрасывается бит UA в '0' и "отпускается" линия SCL).
- Выполнить чтение из регистра SSPBUF (сбрасывается бит BF в '0') и сбросить флаг SSPIF в '0'.
- Принять младший байт адреса (устанавливаются биты SSPIF, BF и UA (SSPSTAT<1> в '1')).
- Записать старший байт адреса в регистр SSPADD (аппаратно сбрасывается бит UA в '0' и "отпускается" линия SCL).
- Выполнить чтение из регистра SSPBUF (сбрасывается бит BF в '0') и сбросить флаг SSPIF в '0'.
- Принять бит повторный START.
- Принять старший байт адреса (устанавливаются биты SSPIF и BF в '1').
- Выполнить чтение из регистра SSPBUF (сбрасывается бит BF в '0') и сбросить флаг SSPIF в '0'.

**Примечание.** В 10-разрядном режиме после команды повторный START (шаг 7) не требуется обновлять значение в регистре SSPADD. В данном случае требуется соответствие только первого байта адреса.

### 9.2.1.2 Прием данных ведомым

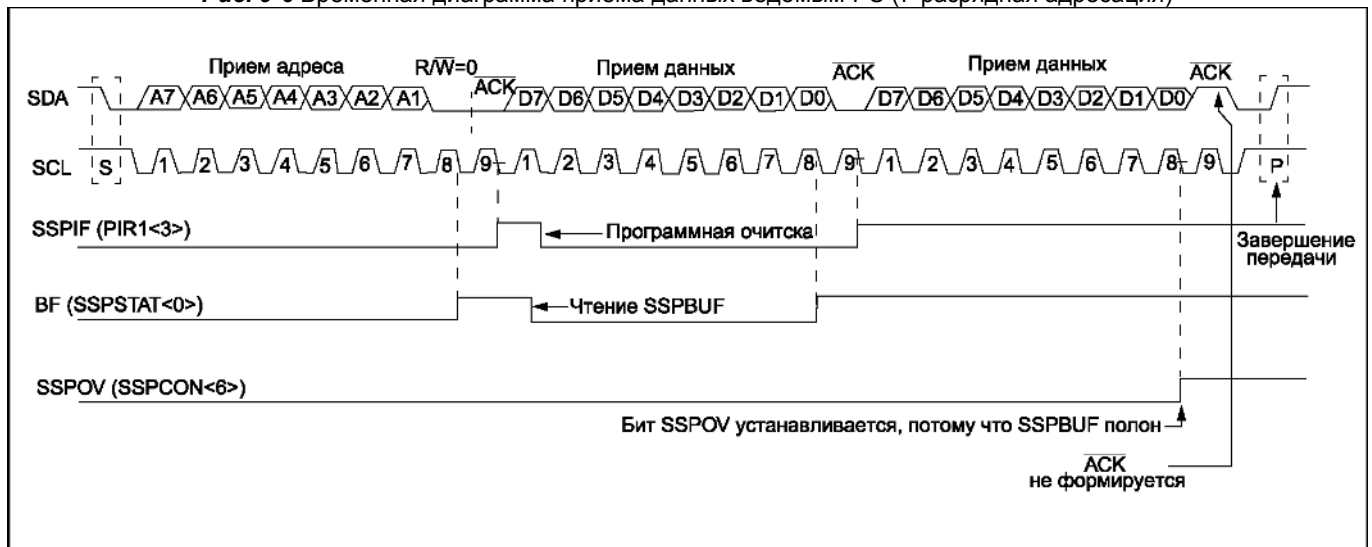
Если бит R/W в адресном байте равен нулю, а принятый адрес совпадает с адресом устройства, то бит R/W в регистре SSPSTAT сбрасывается в '0'. Принятый адрес загружается в регистр SSPBUF.

Если бит BF (буфер полон) или SSPOV (переполнение буфера) установлен в '1', то бит подтверждения -ACK не формируется. Эту ошибку необходимо обработать программно.

Прерывание от модуля MSSP генерируются при каждом принятом байте с шины I<sup>2</sup>C, установкой флага SSPIF (PIR<3>) в '1' (сбрасывается программно). Регистр SSPSTAT используется для определения типа принятого байта.

**Примечание.** Значение регистра SSPBUF будет обновлено, если SSPOV=1 BF=0. Если было выполнено чтение из регистра SSPBUF но не был сброшен бит SSPOV в '0', то бит -ACK не формируется.

Рис. 9-6 Временная диаграмма приема данных ведомым I<sup>2</sup>C (7-разрядная адресация)



**9.2.1.3 Передача данных ведомым**

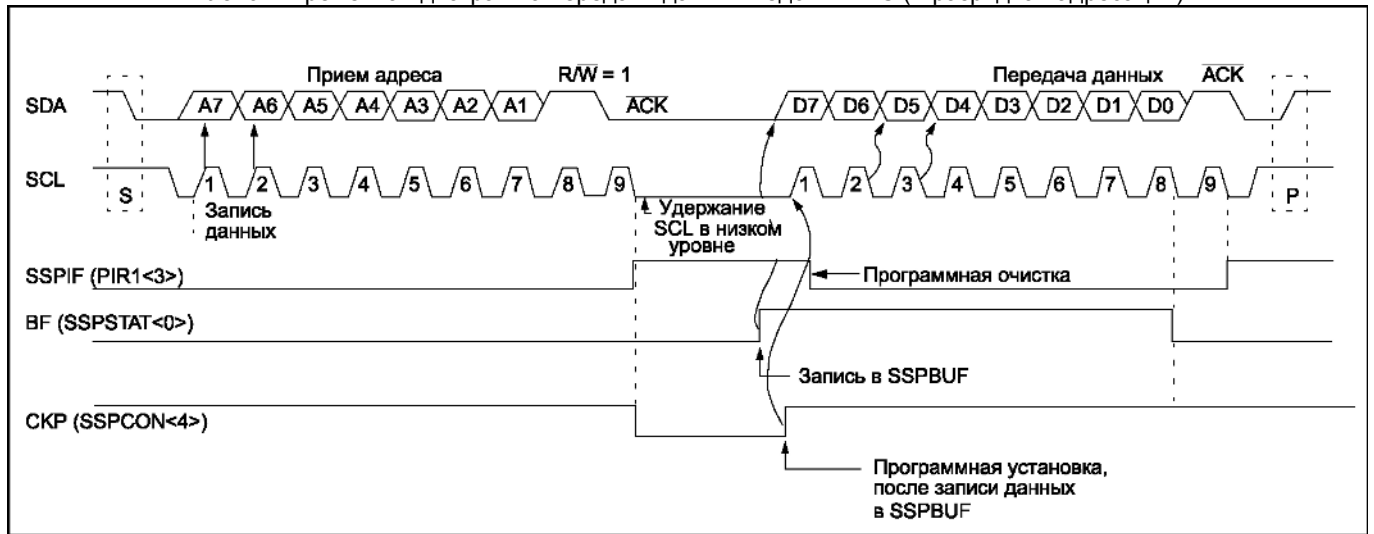
Если бит R/W в адресном байте равен '1', а принятый адрес совпадает с адресом устройства, то бит R/W в регистре SSPSTAT устанавливается в '1'. Принятый адрес загружается в регистр SSPBUF. Бит -ACK формируется девятым битом, после чего линия SCL удерживается в низком логическом уровне. Передаваемые данные должны быть записаны в регистр SSPBUF, после чего они автоматически переписываются в регистр SSPSR. После записи данных необходимо "отпустить" сигнал SCL установкой бита СКР(SSPCON<4>) в '1'. Ведущий шины контролирует состояние линии SCL, ожидая смены уровня сигнала. Восемь бит загруженных данных последовательно сдвигаются по заднему фронту сигнала SCL, что гарантирует достоверное значение данных на линии SDA (см. рисунок 9-7).

Модуль MSSP генерирует прерывание по каждому переданному байту, устанавливая бит SSPIF в '1' по заднему фронту девятого такта сигнала SCL. Флаг SSPIF должен быть сброшен программно. Регистр SSPSTAT используется для определения статуса передачи данных.

Ведущее устройство формирует бит подтверждения ACK на девятом такте сигнала SCL для каждого принятого байта. Если бит подтверждения ACK не сформирован (высокий уровень сигнала SDA), передача данных завершена. Логика ведомого устройства настраивается на обнаружение бита STATR.

Если бит подтверждения ACK был получен (низкий уровень сигнала SDA), в регистр SSPBUF необходимо записать новый байт для передачи. Линию SCL также необходимо "отпустить", установкой бита СКР в '1'.

**Рис. 9-7** Временная диаграмма передачи данных ведомым I<sup>2</sup>C (7-разрядная адресация)



### 9.2.2 Поддержка общего вызова

Процедура адресации на шине I<sup>2</sup>C такова, что первый после START байт определяет, к какому из ведомых устройств обращается ведущий шины. Исключением является адрес общего вызова, при использовании которого теоретически должны откликнуться все ведомые.

Адрес общего вызова – один из восьми зарезервированных адресов шины I<sup>2</sup>C, все биты которого равны нулю (в том числе и бит R/W).

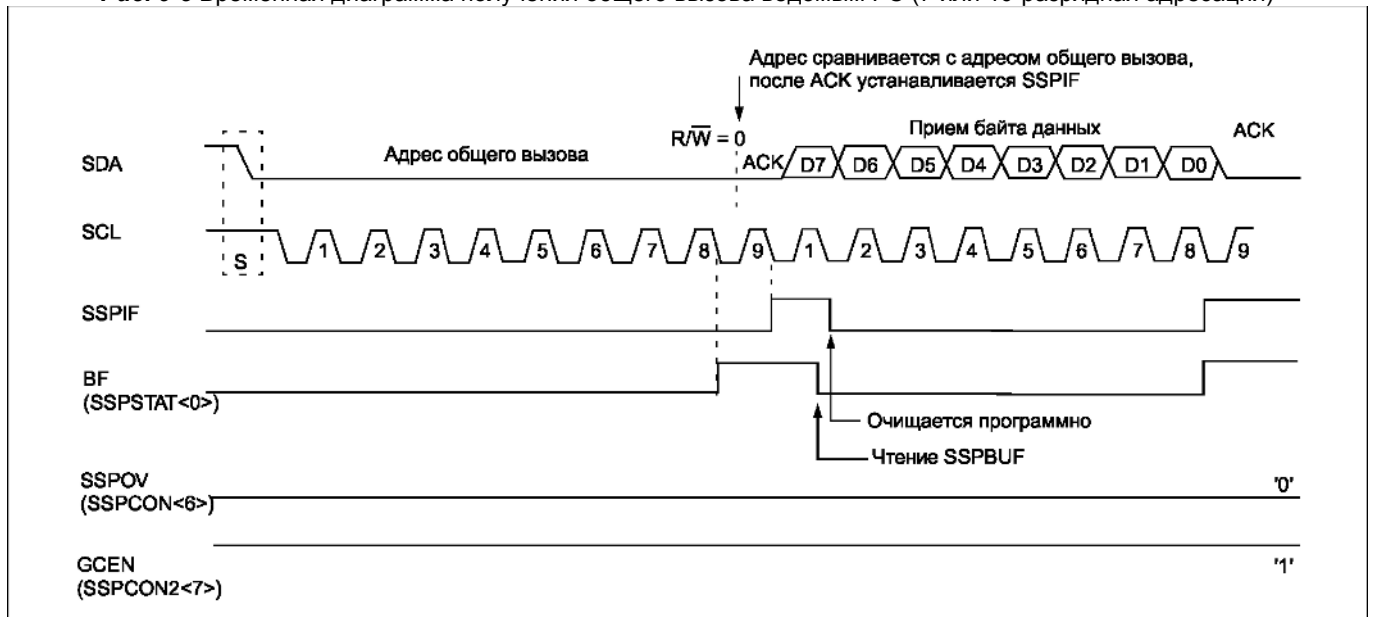
Распознавание адреса общего вызова включается установкой бита GCEN (SSPCON2<7>) в '1'. Следующий за START байт помещается в регистр SSPSR и сравнивается с содержимым SSPADD и фиксированным адресом общего вызова.

При получении адреса общего вызова, содержимое SSPSR переписывается в регистр SSPBUF (устанавливается бит BF в '1') по заднему фронту восьмого такта. На девятом такте формируется бит подтверждения (-ACK) и устанавливается флаг прерываний SSPIF в '1'.

Содержимое регистра SSPBUF позволяет определить получение общего вызова.

В 10-разрядном режиме требуется обновить содержимое регистра SSPADD для проверки соответствия младшего байта адреса после установки бита UA(SSPSTAT<1>) в '1'. Если получен адрес общего вызова в 10-разрядном режиме адресации при GCEN=1, то обновлять значение адреса не требуется. После формирование бита подтверждения ведущее устройство начнет принимать данные (см. рисунок 9-8).

Рис. 9-8 Временная диаграмма получения общего вызова ведомым I<sup>2</sup>C (7 или 10-разрядная адресация)



### 9.2.3 Работа в SLEEP режиме

Ведомый I<sup>2</sup>C может принимать адресные байты или байты данных в SLEEP режиме микроконтроллера. После приема байта микроконтроллер выходит из SLEEP режима, если разрешены прерывания от MSSP модуля.

### 9.2.4 Эффект сброса

При сбросе микроконтроллера модуль MSSP выключается, и прекращается любой обмен данными.

Таблица 9-3 Регистры и биты, связанные с работой модуля MSSP в режиме I<sup>2</sup>C

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
0Dh	PIR2	-	(1)	-	EEIF	BCLIF	-	-	CCP2IF	-r-0 0--0	-r-0 0--0
8Dh	PIE2	-	(1)	-	EEIE	BCLIE	-	-	CCP2IE	-r-0 0--0	-r-0 0--0
13h	SSPBUF	Буфер приемника MSSP / регистр передатчика								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSN	SEN	0000 0000	0000 0000
93h	SSPADD	Регистр адреса / Регистр генератора скорости обмена								0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000

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

Затененные биты на работу не влияют.

**Примечание\*.** Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

**Примечание 1.** Резервные биты. При обращении всегда должны равняться нулю.

### 9.2.5 Режим ведущего I<sup>2</sup>C

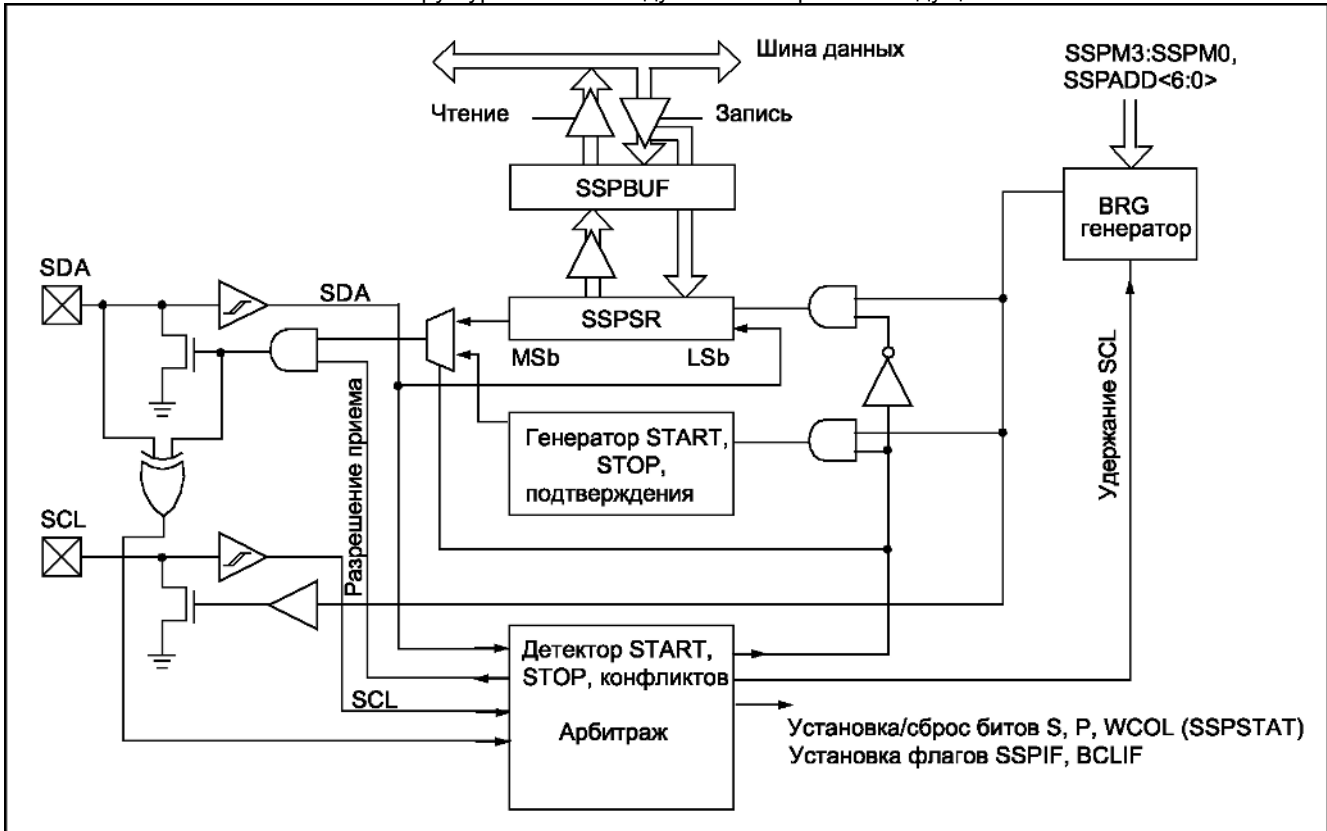
В режиме ведущего поддерживается генерация прерываний при обнаружении на шине битов START и STOP. Биты STOP (P) и START (S) в регистре SSPSTAT равны '0' после сброса микроконтроллера или при выключенном модуле MSSP. Шина находится в неактивном состоянии, если бит P=1 или оба бита S, P равны '0'.

В режиме ведущего выходы SCL, SDA управляются аппаратно.

Следующие события на шине I<sup>2</sup>C могут привести к установке флага прерываний SSPIF в '1':

- Выполнено условие START;
- Выполнено условие STOP;
- Передан/принят байт данных;
- Передан бит подтверждения;
- Выполнено условие повторный START.

Рис. 9-9 Структурная схема модуля MSSP в режиме ведущего I<sup>2</sup>C



### 9.2.6 Режим конкуренции

В режиме конкуренции, прерывания поле START и STOP позволяет определить, когда шина I<sup>2</sup>C свободна. Биты S и P сбрасываются в '0' при сбросе микроконтроллера или при выключении модуля MSSP. Управление шиной может быть перехвачено, когда бит P=1 или шина простаивает (S=0 и P=0). Если шина занята, можно разрешить прерывания от MSSP для обнаружения бита STOP на шине.

При конкуренции линия SDA должна проверяться на соответствия уровня, при ожидаемом высоком уровне на выходе. Эта проверка производится автоматически, а результат помещается в бит BCLIF.

Арбитраж на шине I<sup>2</sup>C может быть потерян во время:

- Передачи адреса;
- Передачи данных;
- Формирования бита START;
- Формирования бита повторный START;
- Формирования бита ACK.

## 9.2.7 Поддержка режима ведущего I<sup>2</sup>C

Ведущий режим включается соответствующей настройкой битов SSPM в регистре SSPCON и установкой в '1' бита SSPEN. После включения ведущего режима аппаратно могут выполняться следующие функции:

- Формирование бита START на линии SCL и SDA;
- Формирование бита повторный START на линии SCL и SDA;
- Запись в регистр SSPBUF инициализируется передача байта данных/адреса;
- Формирование бита STOP на линии SCL и SDA;
- Настройка порта I<sup>2</sup>C на прием данных;
- Формирование бита подтверждения ACK после приема байта на линии SCL и SDA.

**Примечание.** Модуль MSSP в ведущем режиме не имеет стека событий. Это означает, что пользователь не может к примеру инициировать передачу бита START и произвести запись в SSPBUF до того, как START будет завершен. При попытке осуществления подобной операции будет установлен бит WCOL в '1', указывая, что запись в регистр SSPBUF не произошла.

## 9.2.8 Работа в режиме ведущего I<sup>2</sup>C

Ведущий формирует на шине I<sup>2</sup>C тактовый сигнал и биты START, STOP. Текущий обмен данными завершается после формирования бита STOP или повторный START. Поскольку бит повторный START иницирует новый обмен данными, шина I<sup>2</sup>C остается занятой.

Передачик ведущего выдает данные на линию SDA, а тактовый сигнал на линию SCL. Первый передаваемый байт содержит 7-разрядный адрес приемника (при 7-разрядной адресации устройств) и бит направления данных R/-W=0. После каждого переданного 8-разрядного байта принимается бит подтверждения ACK. Биты SATR и STOP формируются для указания начала и завершения передачи данных.

В режиме приема ведущем на шину I<sup>2</sup>C сначала выдается байт, содержащий 7-разрядный адрес передатчика (при 7-разрядной адресации устройств) и бит направления данных R/-W = 1. Данные принимаются с линии SDA, а на линии SCL формирует тактовый сигнал. После каждого принятого байта формируется бит подтверждения. Биты SATR и STOP формируются для указания начала и завершения передачи данных.

Генератор скорости обмена BRG используется для установки требуемой частоты тактового сигнала на линии SCL – 100кГц, 400кГц или 1МГц. Значение для перезагрузки BRG берется из 7 младших бит регистра SSPADD. BRG начинает работу сразу после записи данных в регистр SSPBUF. Как только операция завершена (передан последний бит байта и принят бит подтверждения) генератор BRG останавливается, вывод SCL "отпускается".

Рекомендованная последовательность действий при передаче данных:

1. Инициировать START установкой бита SEN в регистре SSPCON2.
2. Ожидать прерывание (если оно разрешено) или установку бита SSPIF после завершения выполнения START.
3. Запись в SSPBUF иницируется передача адреса.
4. 7 бит адреса (при 7-разрядной адресации) и бит направления данных выдается на SDA.
5. Принять подтверждение ACK от приемника, результат записывается в бит ACKSTAT регистра SSPCON2.
6. По заднему фронту девятого такта устанавливается бит SSPIF в '1'.
7. Запись в SSPBUF иницируется передача данных.
8. 8 бит данных выдаются на SDA.
9. Принять подтверждение ACK от приемника, результат записывается в бит ACKSTAT регистра SSPCON2.
10. По заднему фронту девятого такта устанавливается бит SSPIF в '1'.
11. Инициировать STOP установкой бита PEN в регистре SSPCON2.
12. Ожидать прерывание (если оно разрешено) или установку бита SSPIF после завершения выполнения STOP.

### 9.2.8 Генератор скорости обмена

В ведущем режиме, значение для перезагрузки BRG берется из младших 7 бит регистра SSPADD (см. рисунок 9-10). После загрузки SSPADD в BRG, счетчик BRG считает, декрементируя до нуля (в тактах Q2 и Q4), и останавливается до следующей перезагрузки, которая не всегда производится автоматически. Если после окончания счета сигнал на линии SCL должен перейти в высокий уровень, перезагрузка производится только после этого перехода (см. рисунок 9-11).

**Примечание.** Скорость обмена =  $F_{osc} / (4 \times (SSPADD + 1))$

Рис. 9-10 Структурная схема генератора скорости обмена

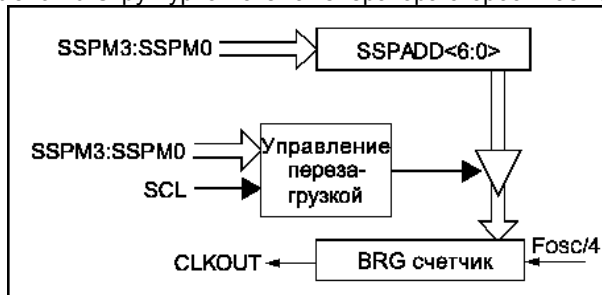
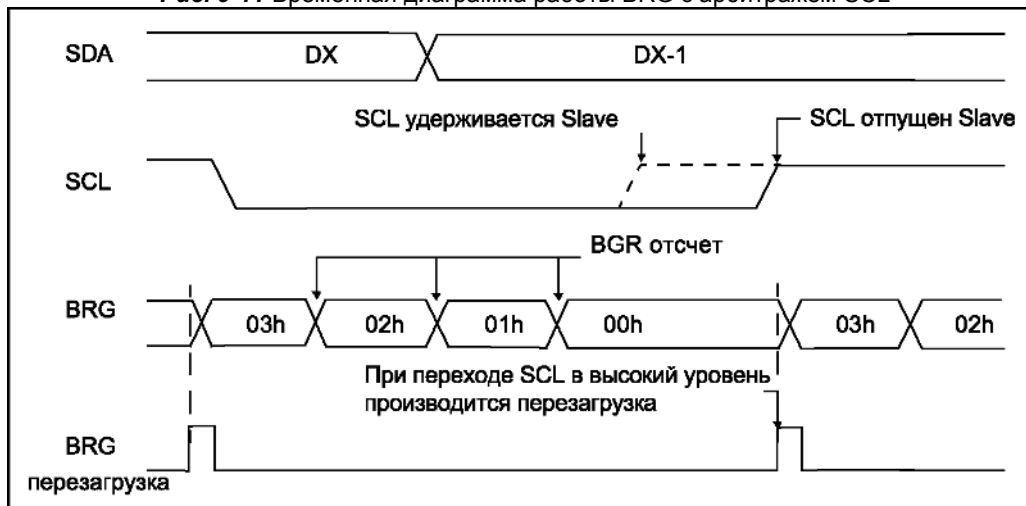


Рис. 9-11 Временная диаграмма работы BRG с арбитражем SCL



### 9.2.9 Формирование бита START в режиме ведущего I<sup>2</sup>C

Чтобы инициировать формирование бита START на шине I<sup>2</sup>C, необходимо установить бит SEN (SSPCON2<0>) в '1'. Если на линиях SCL и SDA высокий уровень сигнала, BRG загружается значением из регистра SSPADD <6:0> и начинает счет. Если по окончании отсчета времени T<sub>BRG</sub> сохраняется высокий уровень на SCL и SDA, сигнал SDA переводится в низкий логический уровень. Перевод SDA в низкий уровень, в то время когда на линии SCL высокий, и есть бит START на шине I<sup>2</sup>C. После формирования бита START устанавливается бит S и флаг прерывания SSPIF в '1', BRG загружается новым значением и начинает счет. По окончании счета бит SEN (SSPCON2<0>) автоматически сбрасывается в '0', генератор останавливается, на SDA остается низкий уровень сигнала. Формирование бита START завершено.

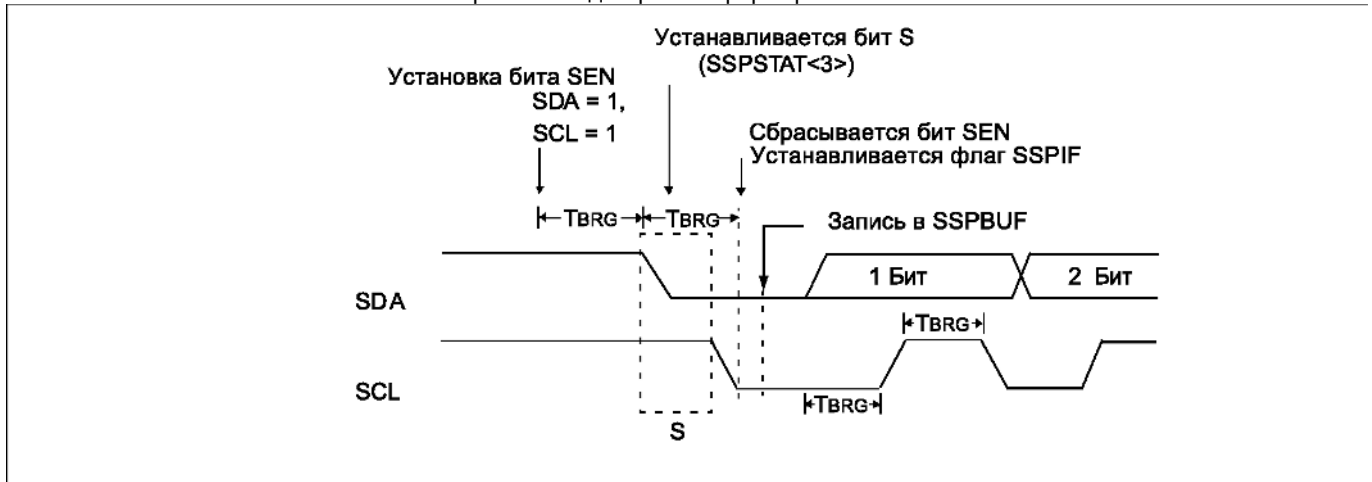
**Примечание.** Если в начале формирования бита START на SDA или SCL присутствует низкий уровень или во время выполнения START низкий уровень на SCL появляется раньше, чем на SDA, устанавливается флаг прерывания BCLIF (конфликт шины), выполнение START прекращается, MSSP переходит в состояние ожидания.

#### 9.2.9.1 Флаг WCOL

Если во время формирования бита START производится попытка записи в SSPBUF, устанавливается бит WCOL, а запись не происходит.

**Примечание.** Поскольку MSSP не имеет стека событий, установка любого из младших 5 битов регистра SSPCON2 до завершения формирования бита START запрещено.

**Рис. 9-12** Временная диаграмма формирования бита START



### 9.2.10 Формирование бита повторный START в режиме ведущего I<sup>2</sup>C

Чтобы инициировать формирование бита повторный START, необходимо установить бит RSEN (SSPCON2<1>) в то время, когда модуль MSSP находится в режиме ожидания. При включении формирования бита повторный START линия SCL переводится в низкий логический уровень. Когда на SCL устанавливается низкий уровень сигнала, BRG перезагружается содержимым регистра SSPADD<6:0> и начинает отсчет, при этом SDA "отпускается" в высокий уровень. Если по окончании счета BRG, на линии SDA сохраняется высокий уровень, SCL также "отпускается". BRG вновь перезагружается и начинает отсчет. Если по окончании отсчета времени  $T_{BRG}$  сохраняется высокий уровень на линиях SCL и SDA, сигнал SDA переводится в низкий уровень. После формирования бита повторный START на шине I<sup>2</sup>C устанавливается бит S (SSPSTAT<3>) в '1'. Флаг SSPIF не будет установлен в '1' до тех пор, пока BRG не перезагрузится новым значением и начнет счет.

**Примечания:**

1. Если бит RSEN установлен в '1' во время выполнения какой либо операции на шине, то не будет выполнено никаких действий.
2. Если на SDA низкий уровень при переходе SCL из низкого уровня в высокий или низкий уровень на SCL появляется раньше, чем на SDA, устанавливается флаг прерывания BCLIF (конфликт шины), формирование бита повторный START прекращается, MSSP переходит в состояние ожидания.

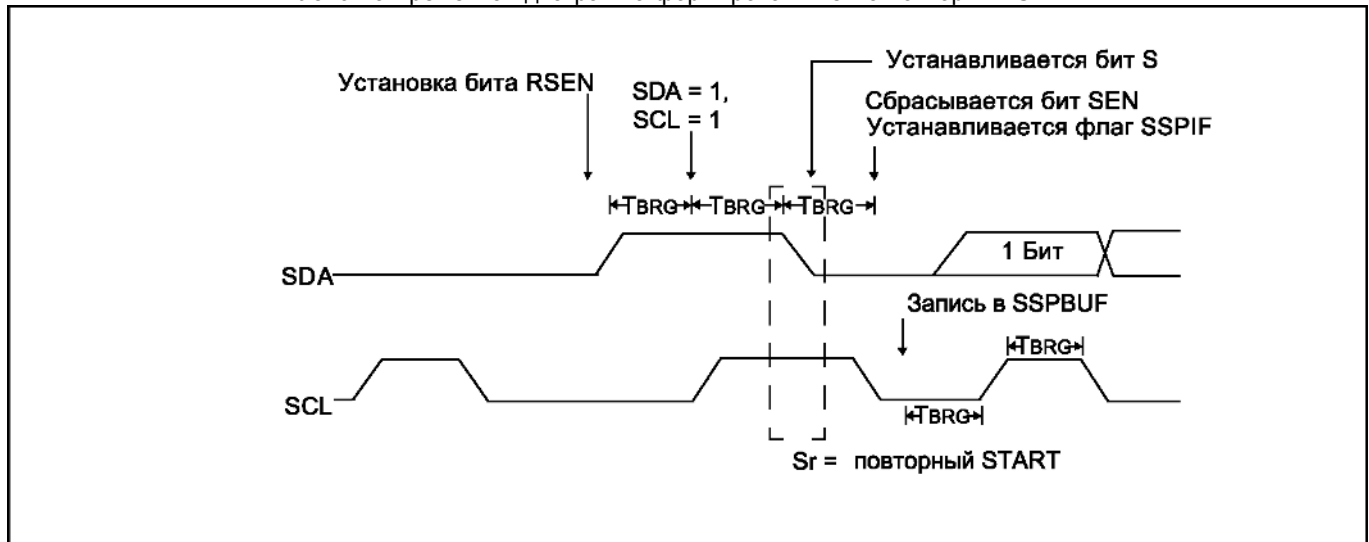
Сразу после установки бита SSPIF пользователь может загрузить регистр SSPBUF 7-разрядным адресом (для 7-разрядного режима адресации) или старшим байтом 10-разрядного адреса. По завершению передачи 8 бит и получения подтверждения ACK, можно передать байт данных или младший байт адреса.

#### 9.2.10.1 Флаг WCOL

Если во время формирования бита повторный START производится попытка записи в SSPBUF, устанавливается бит WCOL, а запись не происходит.

**Примечание.** Поскольку MSSP не имеет стека событий, установка любого из младших 5 битов регистра SSPCON2 до завершения формирования бита повторный START запрещено.

Рис. 9-13 Временная диаграмма формирования бита повторный START





### 9.2.11 Передача данных в режиме ведущего I<sup>2</sup>C

Для инициализации передачи байта данных, 7-разрядного адреса или любой части 10-разрядного адреса нужно просто записать байт в регистр SSPBUF. В результате чего установится бит BF в '1', а BRG начнет формировать сигнал для передачи данных. Каждый передаваемый бит будет выдаваться на SDA по заднему фронту сигнала SCL. Низкий уровень на SCL удерживается в течение одного периода BRG. Данные должны поступать на SDA до прихода переднего фронта на SCL (см. раздел временных характеристик). После отпускания SCL в высокий уровень на время  $T_{BRG}$  данные должны удерживаться на SDA в том же состоянии. По окончании передачи 8-го бита сбрасывается флаг BF в '0', а ведущий "отпускает" SDA с тем, чтобы принять бит подтверждения. По заднему фронту 9-го такта значение ACK записывается в бит ACKSTAT регистра SSPCON2. В этот же момент устанавливается флаг SSPIF в '1', а BRG отключается до следующей операции на шине оставляя низкий уровень на SCL и отпуская SDA (см. рисунок 9-14).

#### 9.2.11.1 Флаг BF

В режиме передачи данных бит BF (SSPSTAT<0>) аппаратно устанавливается в '1' после записи данных в регистр SSPBUF и аппаратно сбрасывается после передачи 8 бит данных.

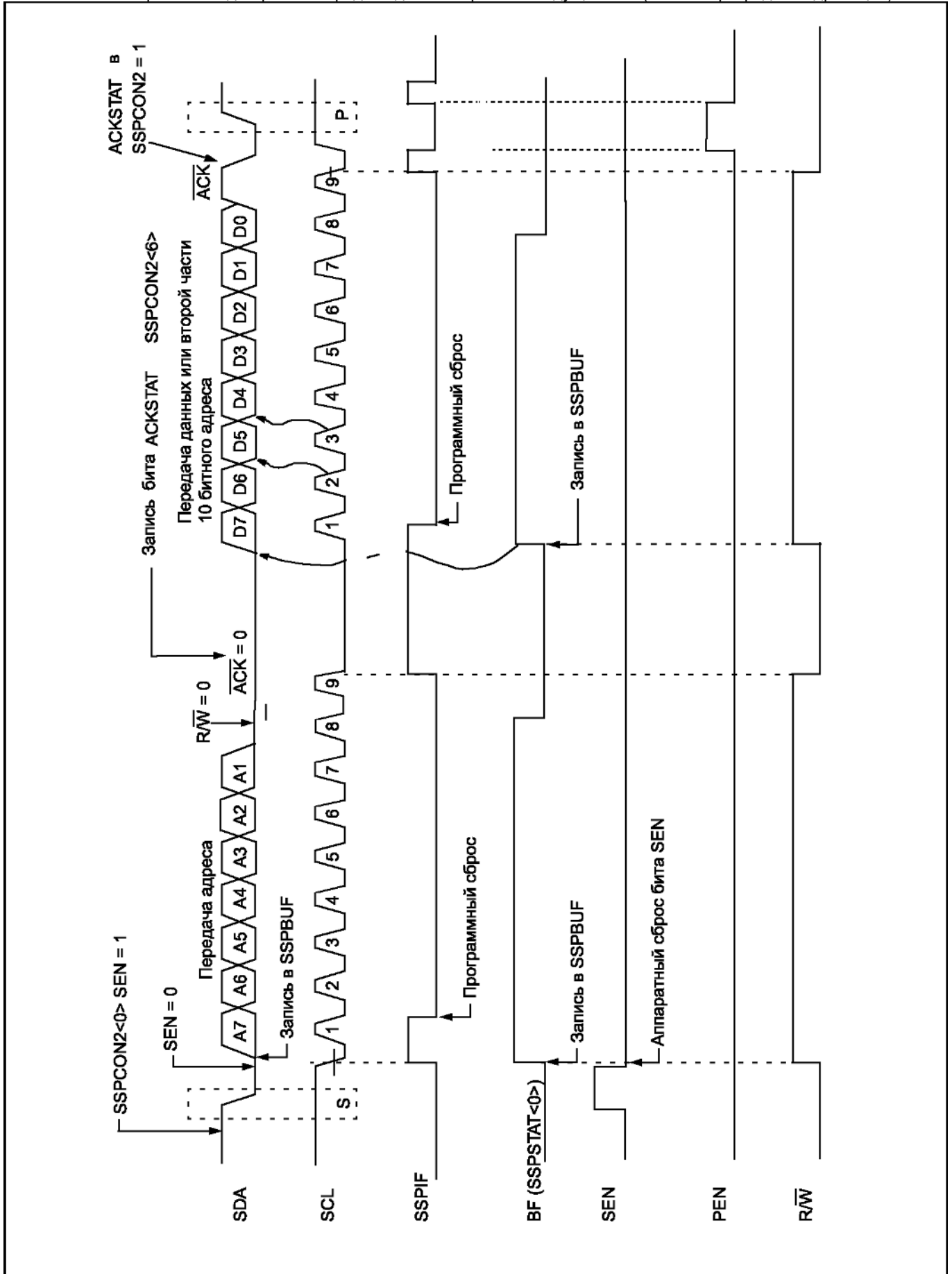
#### 9.2.11.2 Флаг WCOL

Если во время передачи данных производится попытка записи в регистр SSPBUF, устанавливается бит WCOL в '1', а запись не происходит. Бит WCOL сбрасывается программно.

#### 9.2.11.3 Флаг ACKSTAT

В режиме передачи данных бит ACKSTAT(SSPCON2<6>) равен нулю, если ведомый сформировал подтверждение. Ведомый посылает подтверждение, если он распознал адрес (включая общий вызов) или корректно принял данные.

Рис. 9-14 Временная диаграмма передачи данных в режиме ведущего I<sup>2</sup>C (7 или 10-разрядная адресация)



## 9.2.12 Прием данных в режиме ведущего I<sup>2</sup>C

Прием данных ведущем шины I<sup>2</sup>C разрешается установкой бита RCEN(SSPCON2<3>) в '1'.

**Примечание.** При установке бита RCEN в '1' модуль MSSP должен находиться в режиме ожидания.

BRG начинает формировать тактовый сигнал SCL, для приема данных в сдвиговый регистр SSPSR. Каждый бит данных будет приниматься с SDA по заднему фронту SCL. По заднему фронту 8-го такта, значение из SSPSR переписывается в SSPBUF, устанавливается бит BF и SSPIF в '1', BGR останавливается, удерживая SCL в низком уровне, а модуль MSSP переходит в режим ожидания. После чтения регистра SSPBUF аппаратно сбрасывается бит BF в '0'. По окончании приема, ведущий может сформировать бит подтверждения установкой бита ACKEN (SSPCON2<4>) в '1'.

### 9.2.12.1 Флаг BF

В режиме приема данных бит BF (SSPSTAT<0>) аппаратно устанавливается в '1' после загрузки данных в регистр SSPBUF и аппаратно сбрасывается после чтения регистра SSPBUF.

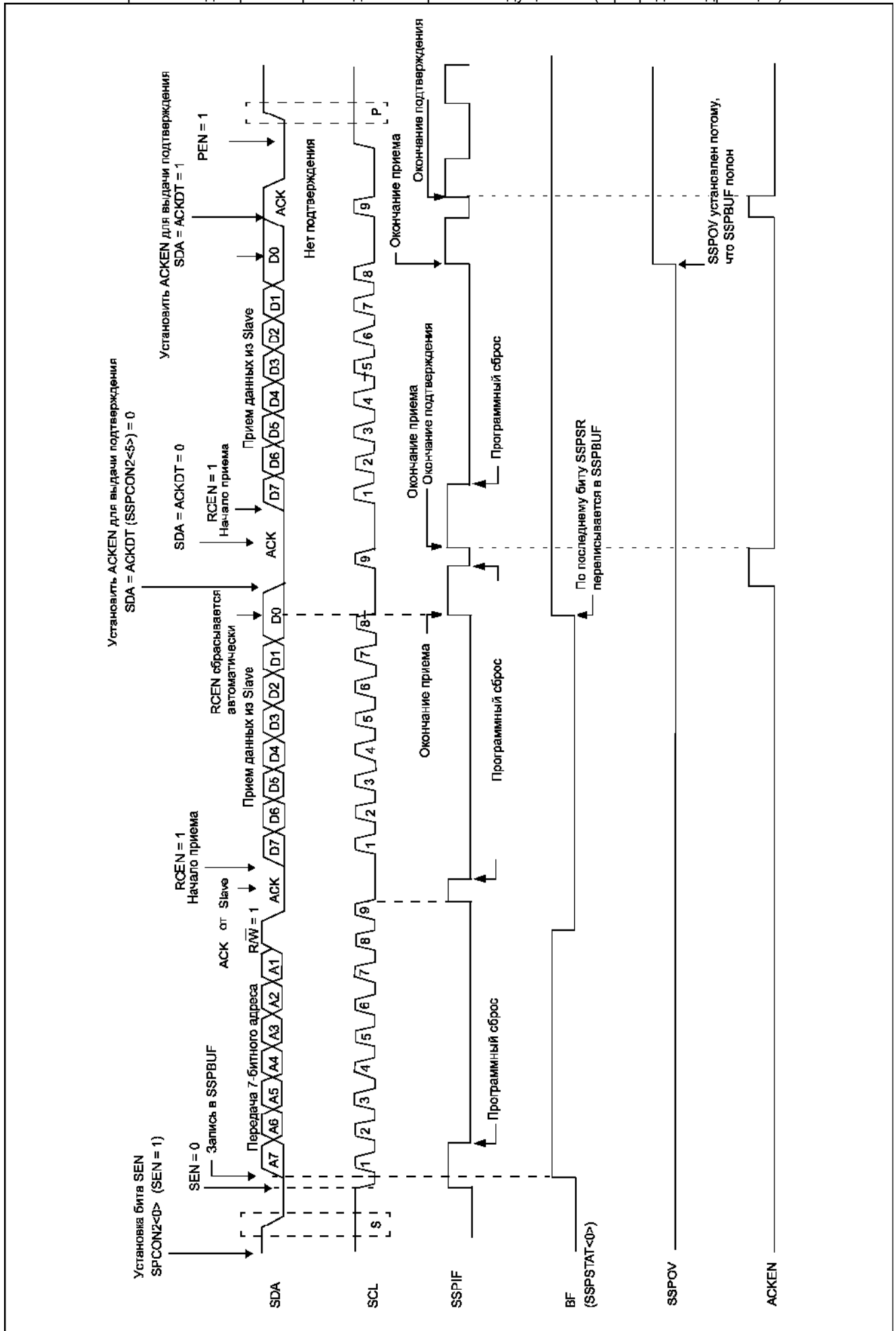
### 9.2.12.2 Флаг SSPOV

При приеме данных бит SSPOV устанавливается в '1', если в момент приема 8-го бита следующего байта бит BF=1 после приема предыдущего байта.

### 9.2.12.3 Флаг WCOL

Если во время приема данных производится попытка записи в регистр SSPBUF, устанавливается бит WCOL в '1', а запись не происходит. Бит WCOL сбрасывается программно.

Рис. 9-15 Временная диаграмма приема данных в режиме ведущего I<sup>2</sup>C (7-разрядная адресация)



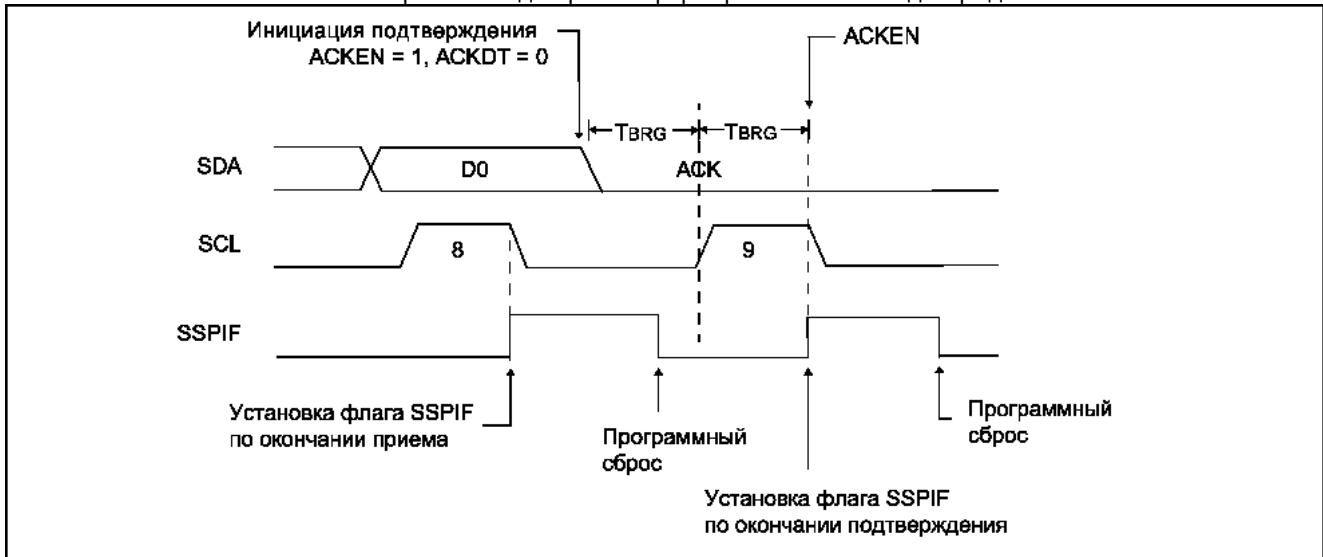
### 9.2.13 Формирование бита подтверждения в режиме ведущего I<sup>2</sup>C

Для инициализации формирования бита подтверждения на шине I<sup>2</sup>C необходимо установить бит ACKEN (SSPCON2<4>) в '1'. При установке этого бита на SCL выдается низкий уровень сигнала, а на SDA содержимое бита ACKDT. Если нужно подтвердить прием, бит ACKDT должен быть равен нулю. По окончании счета BRG линия SCL "отпускается". Как только SCL перейдет из низкого уровня в высокий, BRG опять начнет счет. После окончания счета SCL переводится в низкий уровень, бит ACKEN автоматически сбрасывается в '0', устанавливается флаг прерывания SSPIF в '1', BGR останавливается, а модуль MSSP переходит в режим ожидания (см. рисунок 9-16).

#### 9.2.13.1 Флаг WCOL

Если во время формирования бита подтверждения производится попытка записи в SSPBUF, устанавливается бит WCOL в '1', а запись не происходит.

Рис. 9-16 Временная диаграмма формирования бита подтверждения



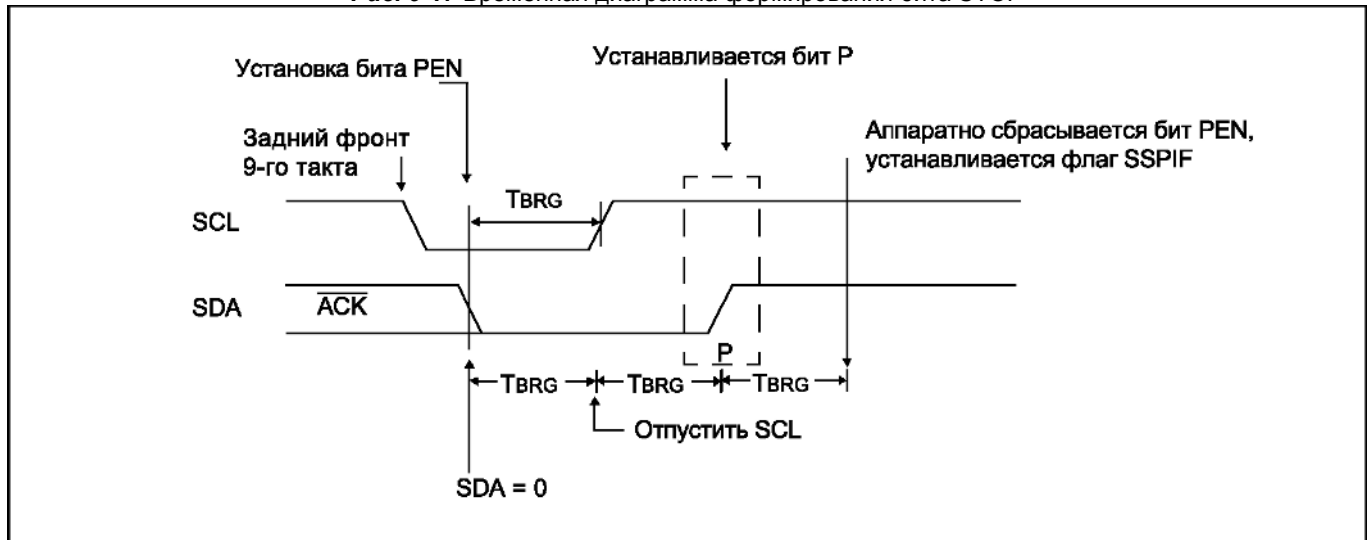
### 9.2.14 Формирование бита STOP в режиме ведущего I<sup>2</sup>C

Чтобы инициировать формирование бита STOP, необходимо установить бит PEN (SSPCON2<2>) в '1'. По окончании приема/передачи данных, после прохождения заднего фронта тактового сигнала на SCL удерживается низкий уровень сигнала. При установке бита PEN ведущий выдает низкий уровень на линию SDA, перезагружает BRG и начинает счет до нуля. По окончании счета линия SCL "отпускается". Через время T<sub>BRG</sub>, после установки высокого уровня на SCL, "отпускается" SDA. Когда на SDA появляется высокий уровень сигнала, устанавливаются биты P и SSPIF в '1', бит PEN автоматически сбрасывается в '0', а генератор BRG останавливается (см. рисунок 9-17).

#### 9.2.14.1 Флаг WCOL

Если во время формирования бита STOP производится попытка записи в SSPBUF, устанавливается бит WCOL в '1', а запись не происходит.

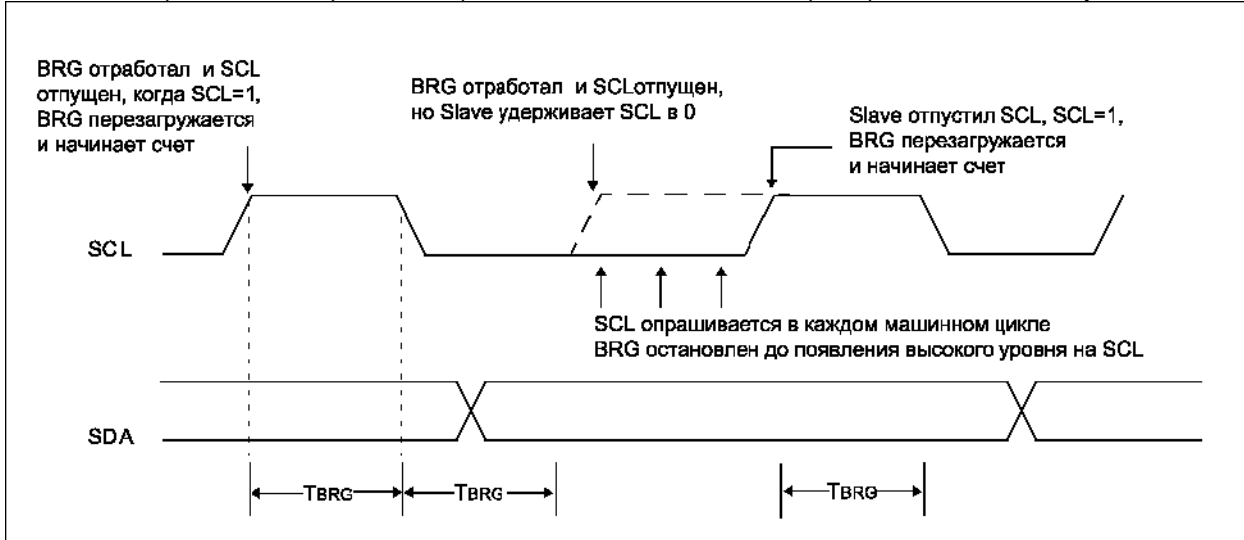
Рис. 9-17 Временная диаграмма формирования бита STOP



### 9.2.15 Синхронизация тактового сигнала

Синхронизация тактового сигнала производится каждый раз во время приема/передачи данных, формирования бита START или STOP и т.д. При "отпускании" ведущем SCL (SCL должен перейти в высокий уровень). В этот момент BRG приостанавливается пока на SCL не появится высокий уровень сигнала. При появлении сигнала высокого уровня на SCL генератор BRG перегружается значением из SSPADD<6:0> и начинает счет. Это гарантирует, что длительность высокого уровня сигнала на SCL всегда будет не меньше  $T_{BRG}$ , даже если другое устройство на шине удерживает тактовый сигнал.

Рис. 9-18 Временная диаграмма синхронизации тактового сигнала при передаче данных ведущем шиной



### 9.2.16 Режим конкуренции, арбитраж и конфликты шины

В режиме конкуренции необходимо поддерживать правила арбитража шины. Во время передачи адреса/данных на SDA ведущий может потерять арбитраж, если он формирует высокий уровень сигнала, а другой ведущий сформировал низкий уровень на SDA. При переходе SCL в высокий уровень, сигнал на SDA изменяться не может. Если на SDA ожидается высокий уровень, а в действительности низкий, значит возник конфликт шины. Обнаружив конфликт шины, ведущий устанавливает флаг прерывания BCLIF в '1', прекращает текущую операцию на шине и переводит порт I<sup>2</sup>C в режим ожидания (см. рисунок 9-19).

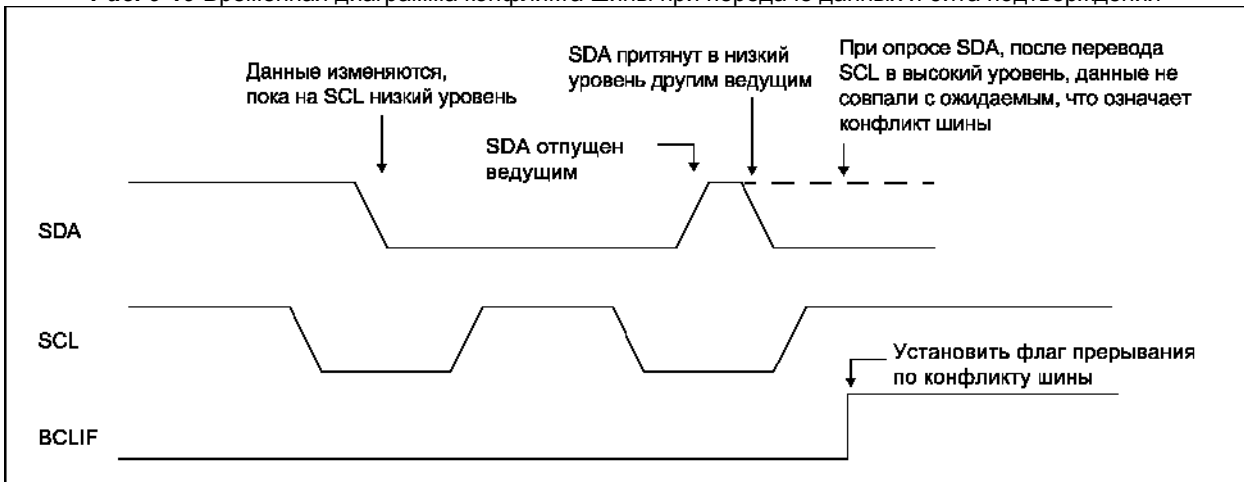
Если при возникновении конфликта шины выполнялась передача данных, она обрывается, устанавливается бит BF в '1', а линии SCL и SDA "отпускаются" в высокое состояние. В регистр SSPBUF может быть произведена запись, причем запись в SSPBUF инициирует передачу независимо от того, в какой момент передатчик отключился при возникновении конфликта шины. Если пользователь обрабатывает прерывания по конфликту шины, после освобождения шины он может продолжить обмен, сформировав бит START.

Если при возникновении конфликта выполнялось формирование бита START, повторный START, STOP или ACK, выполняемая операция обрывается, SCL и SDA "отпускаются", а соответствующий бит управления в SSPCON2 сбрасывается в '0'. Если пользователь обрабатывает прерывания по конфликту шины, после освобождения шины он может продолжить обмен, сформировав бит START.

Ведущий продолжает следить за состоянием шины, и при появлении бита STOP устанавливается флаг прерывания SSPIF в '1'.

В режиме конкуренции использование прерывания при обнаружении битов START и STOP позволяет определить занятость шины. Управление шиной может быть перехвачено при установленном бите P или сброшенных битах S и P.

Рис. 9-19 Временная диаграмма конфликта шины при передаче данных и бита подтверждения



### 9.2.16.1 Конфликт шины при формировании бита START

Во время формирования бита START конфликт шины возникает если:

- В начале START на SDA или SCL низкий уровень сигнала (см. рисунок 9-20);
- На SCL низкий уровень появляется раньше чем на линии SDA (см. рисунок 9-21).

Во время формирования бита START сигналы SCL и SDA продолжают отслеживаться. Если SCL или SDA имеют низкий уровень сигнала, то формирование бита START прекращается, устанавливается флаг BCLIF в '1', а модуль MSSP переходит в режим ожидания (см. рисунок 9-20).

Бит START начинается при наличии высокого уровня сигнала на линиях SCL и SDA. Если на SCL появляется низкий уровень раньше, чем на SDA, возникает конфликт шины, поскольку это подразумевает, что другой ведущий пытается в это время передать данные.

Если во время счета BRG на SDA появляется низкий уровень сигнала, BRG сбрасывается, а на SDA формируется низкий уровень раньше времени (см. рисунок 9-22). Если же на SDA высокий уровень, низкий уровень формируется в конце счета BRG. Генератор BRG перезагружается и считает до нуля. Если в это время на SCL появится низкий уровень, конфликт шины не возникает. В конце счета BRG SCL переводится в низкий уровень.

**Примечание.** Конфликт шины во время START не возникает, потому что два или более ведущих могут сформировать START одновременно, но при этом один из них первым переведет SDA в низкий уровень. Конфликт шины не возникает, поскольку ведущие могут продолжить арбитраж во время передачи адреса, данных, формировании бита повторный START и STOP.

**Рис. 9-20** Временная диаграмма конфликта шины во время формирования бита START (только SDA)

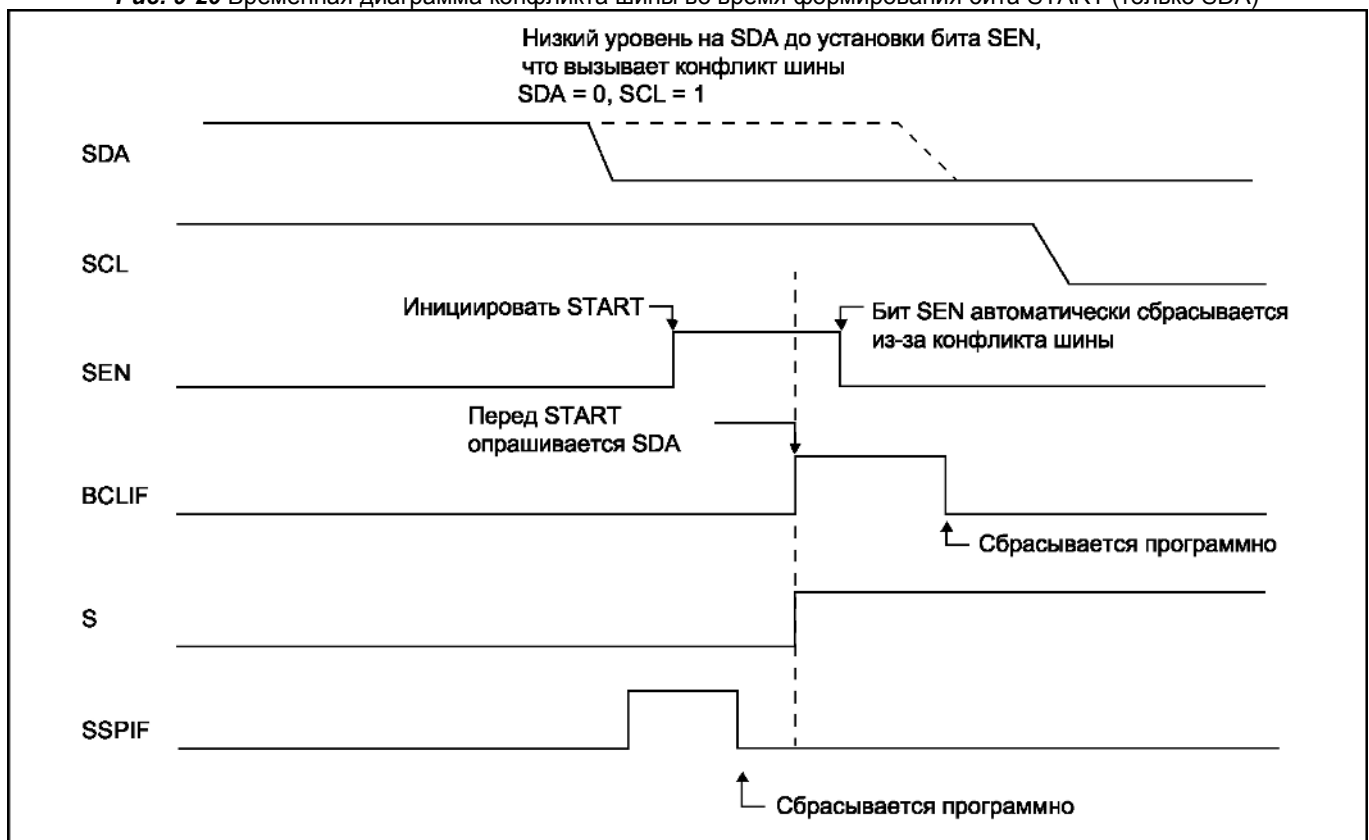


Рис. 9-21 Временная диаграмма конфликта шины во время формирования бита START (SCL=0)

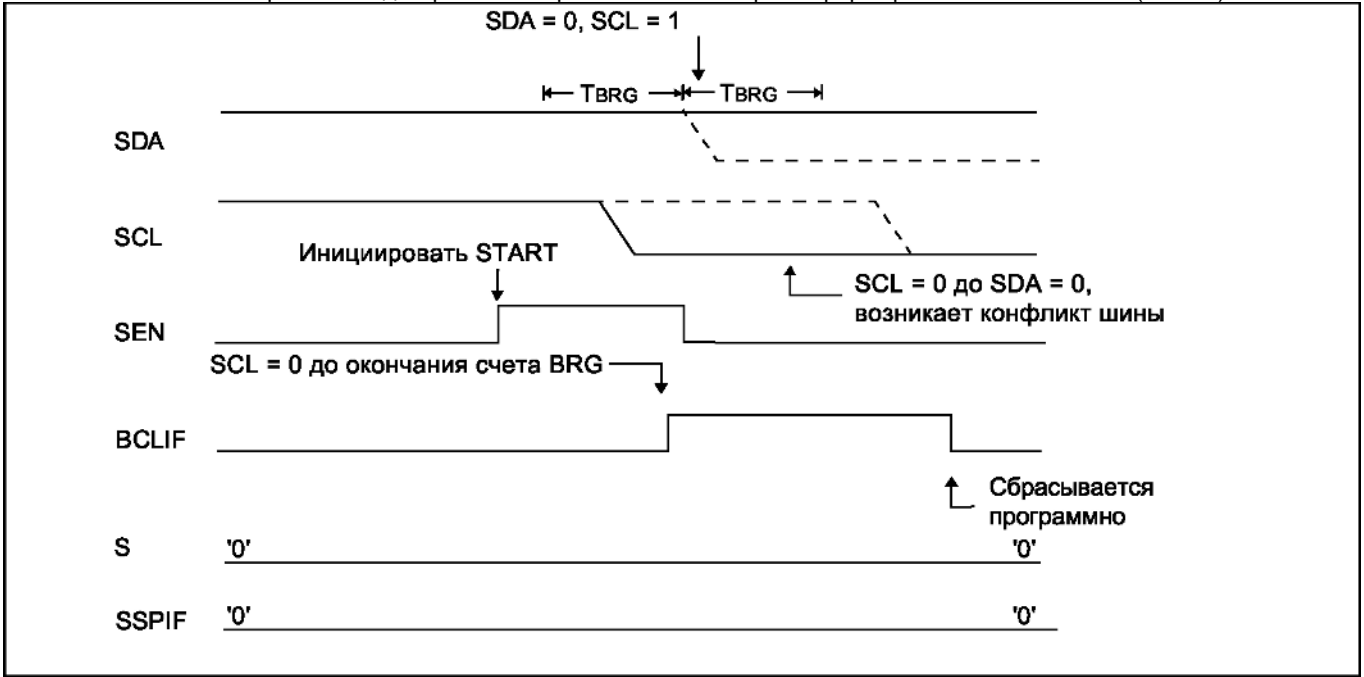
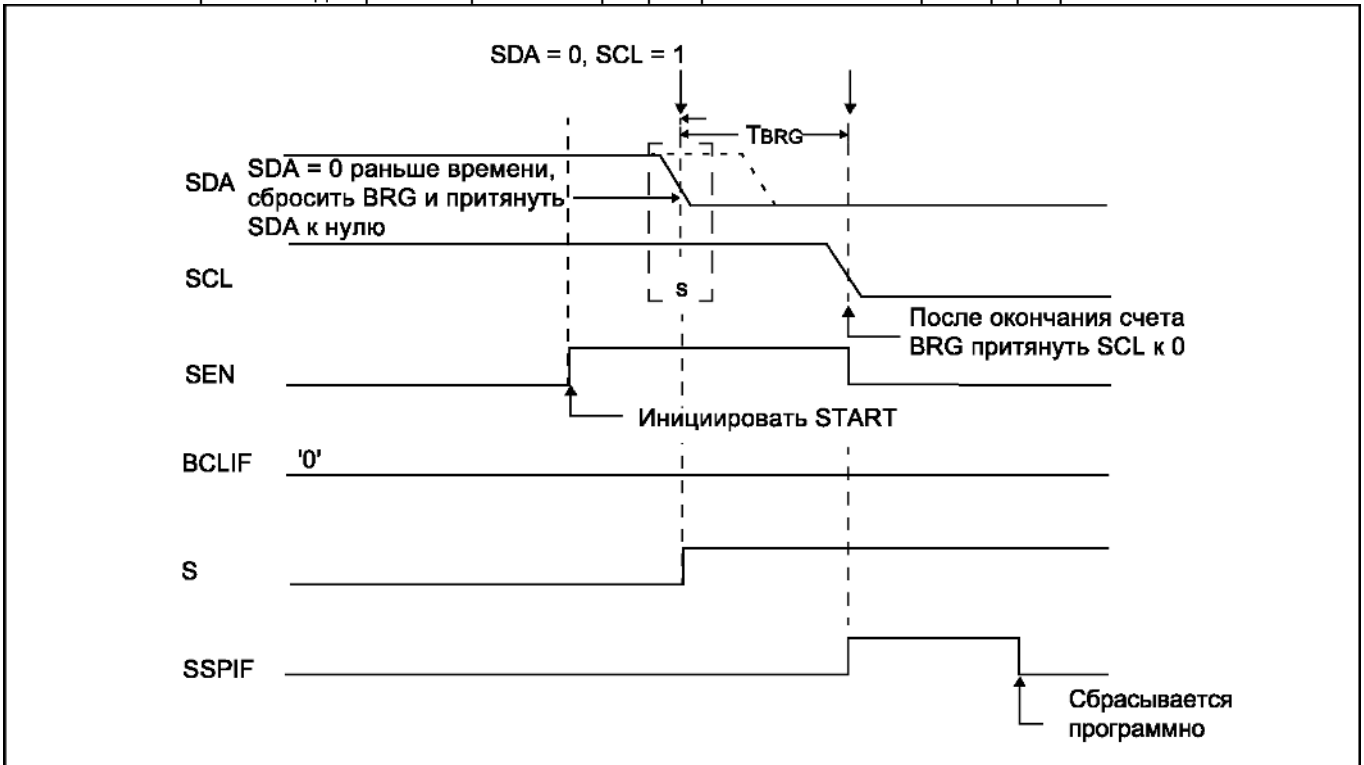


Рис. 9-22 Временная диаграмма сброса BRG при проверке линии SDA во время формирования бита START





**9.2.16.2 Конфликт шины при формировании бита повторный START**

Во время формирования бита повторный START конфликт шины возникает если:

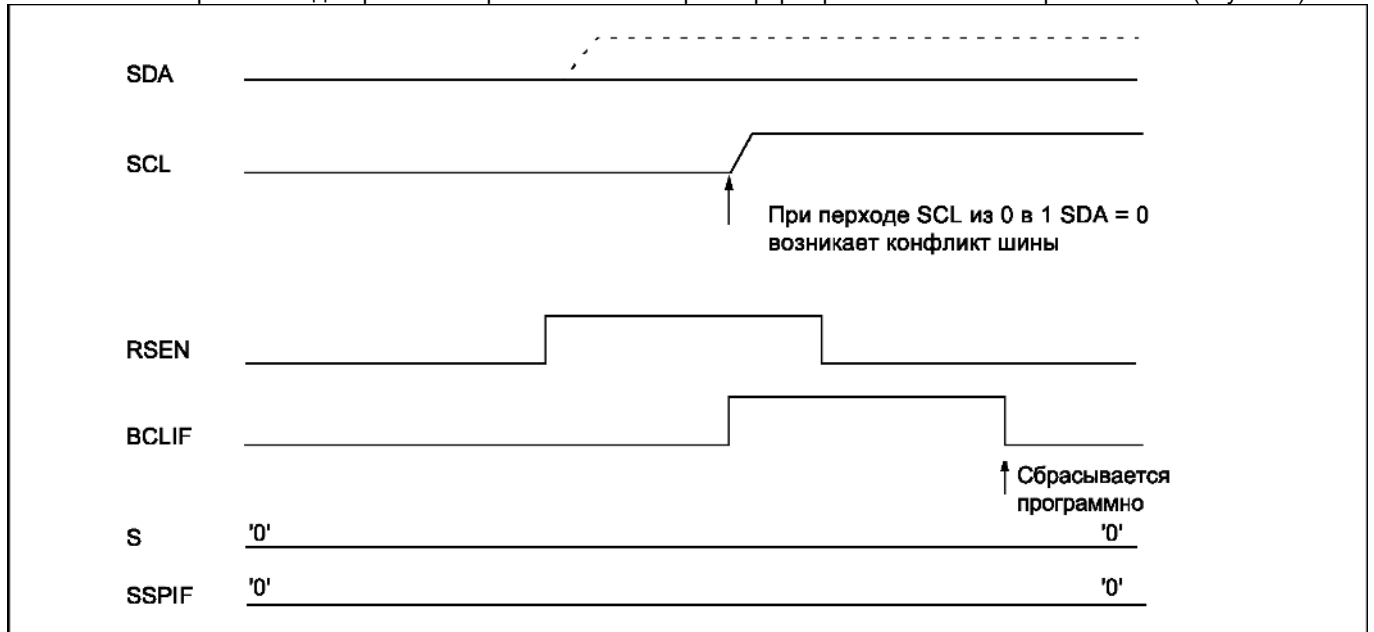
- На SDA низкий уровень при переходе SCL из низкого уровня в высокий (см. рисунок 9-23);
- SCL переходит в низкий уровень раньше SDA, что указывает на то, что другой ведущий пытается передать данные (см. рисунок 9-24).

После "отпускания" линии SDA сигнал на выводе должен перейти в высокий уровень, после чего BRG перезагружается и начинает счет. Затем "отпускается" SCL и при появлении на нем высокого уровня опрашивается SDA. Если на SDA низкий уровень сигнала, значит произошел конфликт шины, т.е. другой ведущий пытается передать данные. Если на SDA высокий уровень, то BRG снова перезагружается и начинается счет. Если SDA переходит в низкий уровень до окончания счета, конфликт шины не происходит, поскольку два или более ведущих могут пытаться получить доступ к шине одновременно.

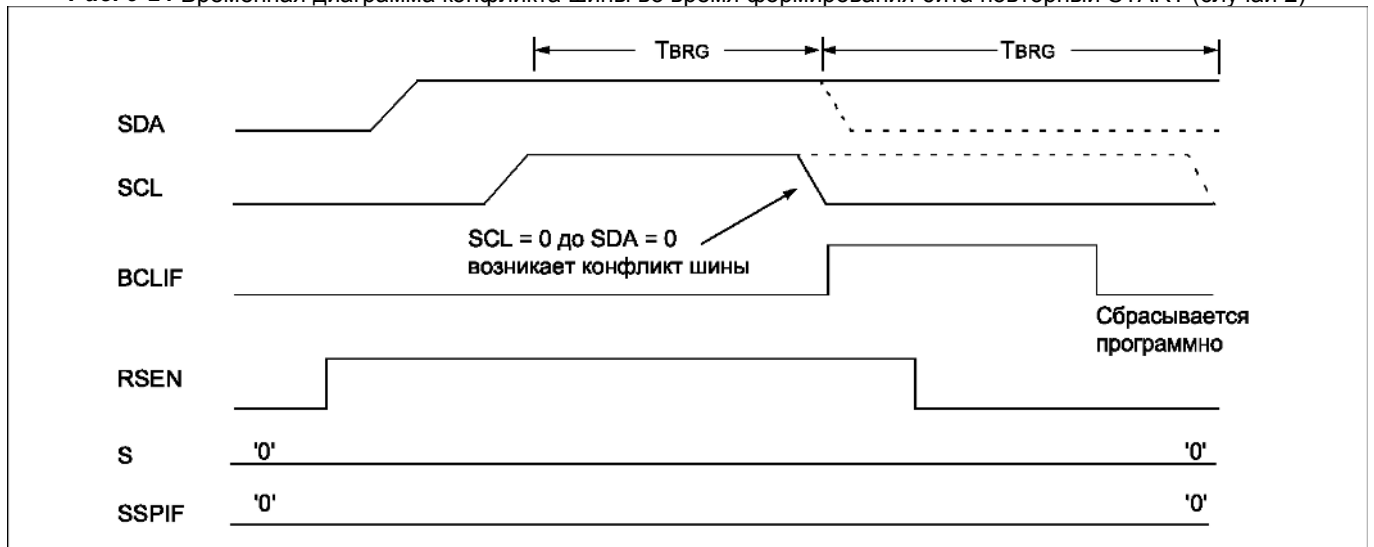
Если на линии SCL сигнал переходит в низкий уровень до окончания счета, а на SDA сохраняется высокий уровень, значит, произошел конфликт шины, т.е. другой ведущий пытается передать данные.

Если по окончании счета BGR на SCL и SDA высокий уровень, то SDA переводится в низкий уровень, а BRG перезагружается и начинает счет. По окончании счета, независимо от уровня сигнала на SCL он переводится в низкий уровень (см. рисунок 9-24).

**Рис. 9-23** Временная диаграмма конфликта шины во время формирования бита повторный START (случай 1)



**Рис. 9-24** Временная диаграмма конфликта шины во время формирования бита повторный START (случай 2)



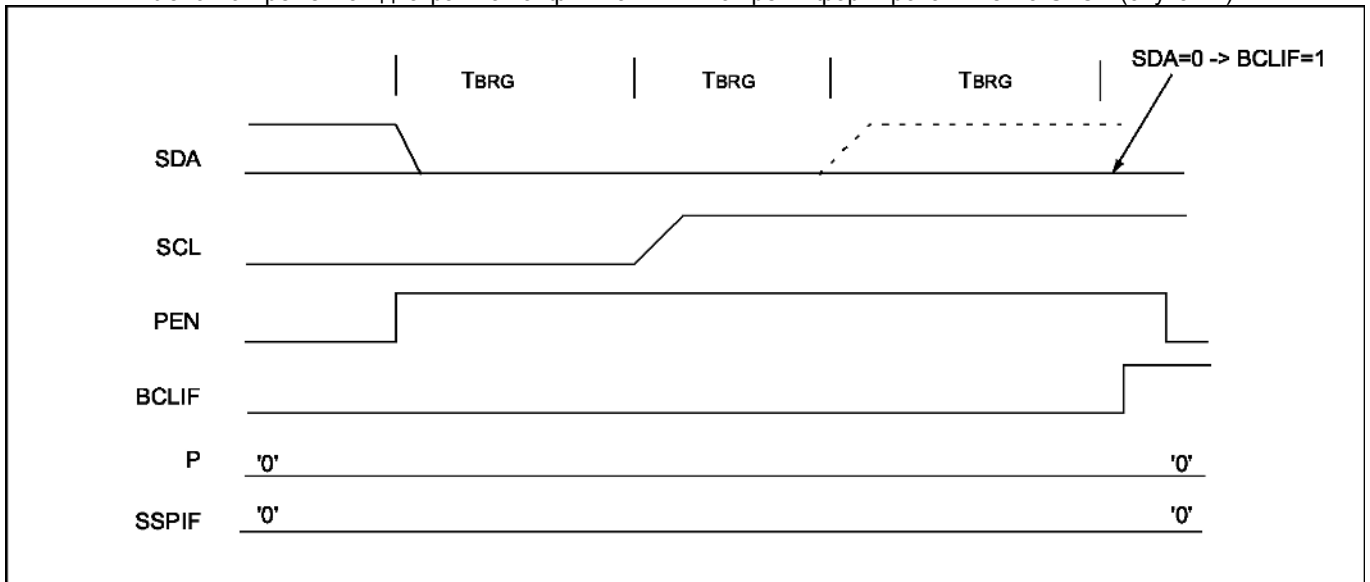
**9.2.16.3 Конфликт шины при формировании бита STOP**

Во время формирования бита STOP конфликт шины возникает если:

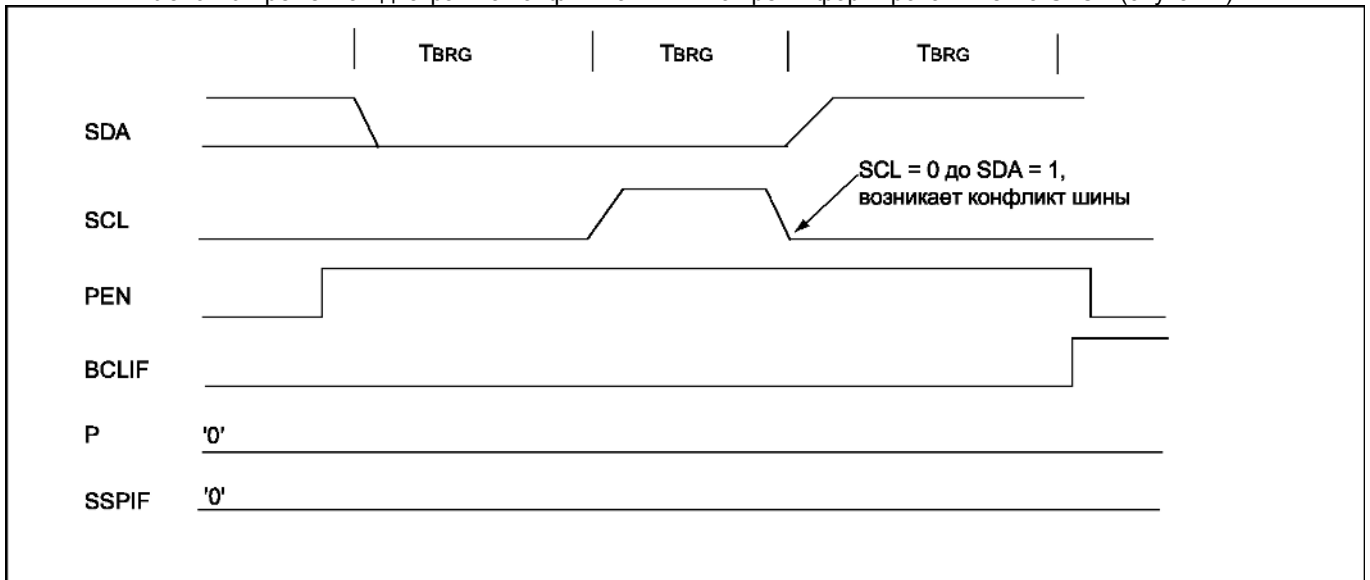
- После "отпускания" линии SDA и окончания счета BRG на SDA по-прежнему низкий уровень сигнала (см. рисунок 9-25);
- После "отпускания" линии SDA сигнал на SCL переходит в низкий уровень до того, как на SDA перейти в высокий уровень (см. рисунок 9-26).

Формирование бита STOP начинается с перевода линии SDA в низкий уровень, затем SCL "отпускается". После появления на SCL высокого уровня BRG перезагружается и начинает счет. По окончании счета SDA "отпускается", BRG перезагружается и снова начинает счет и опрашивает SDA. Если на нем низкий уровень или на SCL появился низкий уровень до перехода SDA в высокий, значит, произошел конфликт шины, т.е. другой ведущий пытается передать данные.

**Рис. 9-25** Временная диаграмма конфликта шины во время формирования бита STOP (случай 1)



**Рис. 9-26** Временная диаграмма конфликта шины во время формирования бита STOP (случай 2)



### 9.3 Подключение к шине I<sup>2</sup>C

Для стандартного режима I<sup>2</sup>C значение резисторов R<sub>p</sub> и R<sub>s</sub> (см. рисунок 9-27) зависит от следующих параметров:

- Напряжение питания;
- Емкость шины;
- Количество устройств на шине (входной ток + ток утечки).

Напряжение питания ограничивает минимальное значение сопротивления R<sub>p</sub>, из-за ограничения минимального тока стока 3мА при V<sub>OL</sub> max = 0.4В.

Например:

$$V_{DD} = 5\text{В} \pm 10\%$$

$$V_{OL} \text{ max} = 0.4 \text{ В при } 3\text{мА}$$

$$R_p \text{ min} = (5.5 - 0.4) / 0.003 = 1.7 \text{ кОм}$$

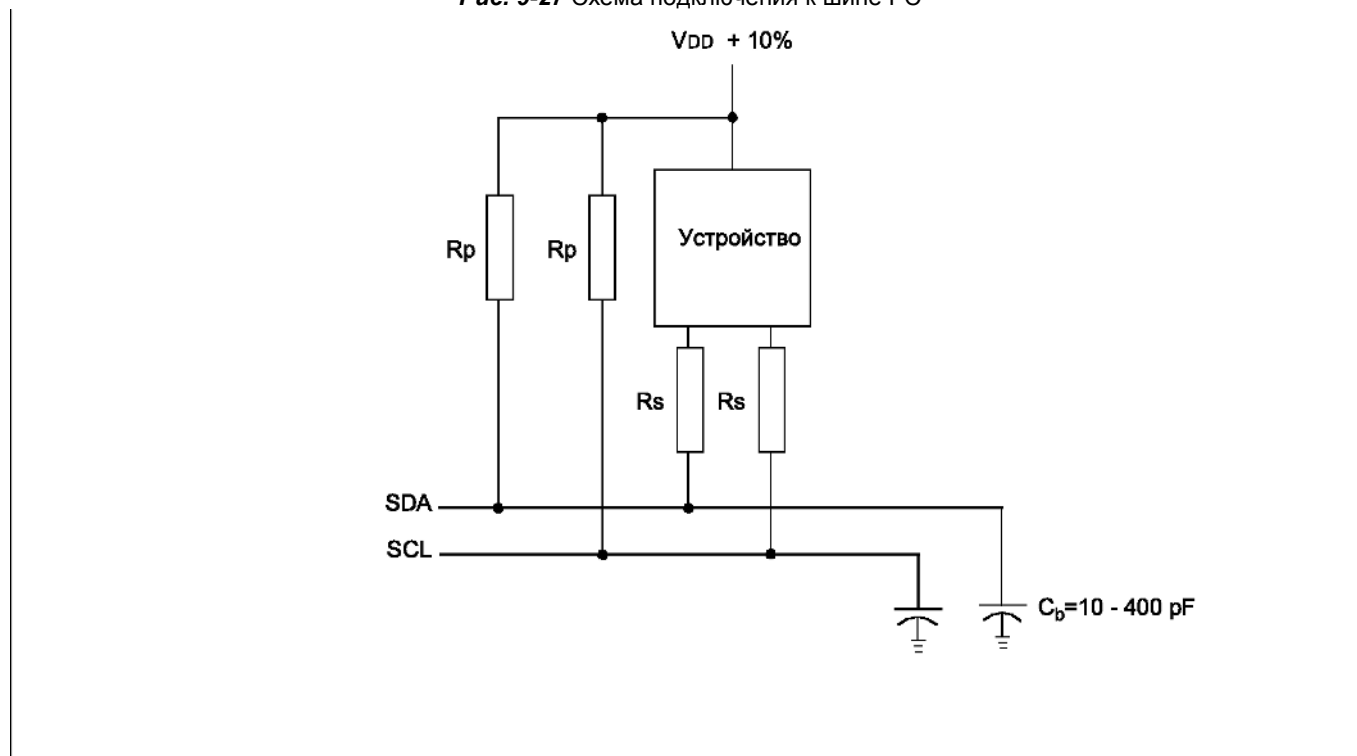
Максимальное значение R<sub>s</sub> определяется допустимым уровнем шума.

Емкость шины определяется суммарная емкостью проводников и выводов. Емкость определяет максимальное значение R<sub>p</sub> из-за допустимой длительности фронта.

Бит SMP в регистре SSPSTAT включает управление длительностью фронта SDA и SCL для того, чтобы фронты этих сигналов удовлетворяли спецификации при работе в скоростном режиме с частотой 400 кГц.

Устройства на шине I<sup>2</sup>C должны иметь один источник питания, к которому подключаются подтягивающие резисторы.

Рис. 9-27 Схема подключения к шине I<sup>2</sup>C



## 10.0 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 TRISC<7:6> должны быть установлены в '1' для использования выводов RC6/TX/CK и RC7/RX/DT в качестве портов универсального синхронно-асинхронного приемопередатчика. Модуль USART поддерживает режим детектирования 9-разрядного адреса для работы в сетевом режиме.

### 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 BRG  
0 = ведомый, внешний тактовый сигнал с входа CK

Асинхронный режим  
Не имеет значения

бит 6: **TX9:** Разрешение 9-разрядной передачи  
1 = 9-разрядная передача  
0 = 8-разрядная 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
<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:	<b>SPEN:</b> Разрешение работы последовательного порта 1 = модуль USART включен (выводы RC7/RX/DT, RC6/TX/CK подключены к USART) 0 = модуль USART выключен						
бит 6:	<b>RX9:</b> Разрешение 9-разрядного приема 1 = 9-разрядный прием 0 = 8-разрядный прием						
бит 5:	<b>SREN:</b> Разрешение одиночного приема <u>Синхронный режим</u> 1 = разрешен одиночный прием 0 = запрещен одиночный прием Сбрасывается в '0' по завершению приема. <b>Примечание.</b> В режиме ведомого не имеет значения  <u>Асинхронный режим</u> Не имеет значения						
бит 4:	<b>CREN:</b> Разрешение приема <u>Синхронный режим</u> 1 = прием разрешен (при установке бита CREN автоматически сбрасывается бит SREN) 0 = прием запрещен  <u>Асинхронный режим</u> 1 = прием разрешен 0 = прием запрещен						
бит 3:	<b>ADDEN:</b> Разрешение детектирования адреса <u>Асинхронный 9-разрядный прием (RX9=1)</u> 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.  <u>Асинхронный 8-разрядный прием (RX9=0)</u> Не имеет значения  <u>Синхронный режим</u> Не имеет значения						
бит 2:	<b>FERR:</b> Ошибка кадра, сбрасывается при чтении регистра RCREG 1 = произошла ошибка кадра 0 = ошибки кадра не было						
бит 1:	<b>OERR:</b> Ошибка переполнения внутреннего буфера, устанавливается в '0' при сбросе бита CREN 1 = произошла ошибка переполнения 0 = ошибки переполнения не было						
бит 0:	<b>RX9D:</b> 9-й бит принятых данных (может использоваться для программной проверки четности)						

## 10.1 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 10-1 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 10-1. Затем рассчитывается ошибка скорости обмена.

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 10-1** 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 10-2** 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, BOR	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.

### 10.1.1 Выборка

Сигнал с входа RC7/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. Если BRGH = 1 (выбран высокоскоростной режим), выборка производится на втором такте сигнала x4 тремя запросами).

**Таблица 10-3** 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	-	-	-	-	-	-	-	-	-
1,2	1,221	1,75	255	1,202	0,17	207	1,202	0,17	129
2,4	2,404	0,17	129	2,404	0,17	103	2,404	0,17	64
9,6	9,766	1,73	31	9,615	0,16	25	9,766	1,73	15
19,2	19,531	1,72	15	19,231	0,16	12	19,531	1,72	7
28,8	31,250	8,51	9	27,778	3,55	8	31,25	8,51	4
33,6	34,722	3,34	8	35,714	6,29	6	31,25	6,99	4
57,6	62,500	8,51	4	62,500	8,51	3	52,083	9,58	2
Максим.	312,5	-	0	250	-	0	156,250	-	0
Миним.	1,221	-	255	0,977	-	255	0,610	-	255

Скорость обмена (К)	Fosc = 4 МГц			Fosc = 3,6864 МГц		
	Реальная скорость	Ошибка %	Значение SPBRG (десят.)	Реальная скорость	Ошибка %	Значение SPBRG (десят.)
0,3	0,300	0	207	0,3	0	191
1,2	1,202	0,17	51	1,2	0	47
2,4	2,404	0,17	25	2,4	0	23
9,6	8,929	6,99	6	9,6	0	5
19,2	20,833	8,51	2	19,2	0	2
28,8	31,250	8,51	1	28,8	0	1
33,6	-	-	-	-	-	-
57,6	62,500	8,51	0	57,6	0	0
Максим.	62,500	-	0	57,6	-	0
Миним.	0,244	-	255	0,225	-	255

**Таблица 10-4** 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.)
0,3	-	-	-	-	-	-	-	-	-
1,2	-	-	-	-	-	-	-	-	-
2,4	-	-	-	-	-	-	2,441	1,71	255
9,6	9,615	0,16	129	9,615	0,16	103	9,615	0,16	64
19,2	19,231	0,16	64	19,231	0,16	51	19,531	1,72	31
28,8	29,070	0,94	42	29,412	2,13	33	28,409	1,36	21
33,6	33,784	0,55	36	33,333	0,79	29	32,895	2,10	18
57,6	59,524	3,34	20	58,824	2,13	16	56,818	1,36	10
Максим.	1250	-	0	1000	-	0	625	-	0
Миним.	4,883	-	255	3,906	-	255	2,441	-	255

Скорость обмена (К)	Fosc = 4 МГц			Fosc = 3,6864 МГц		
	Реальная скорость	Ошибка %	Значение SPBRG (десят.)	Реальная скорость	Ошибка %	Значение SPBRG (десят.)
0,3	-	-	-	-	-	-
1,2	1,202	0,17	207	1,2	0	191
2,4	2,404	0,17	103	2,4	0	95
9,6	9,615	0,16	25	9,6	0	23
19,2	19,231	0,16	12	19,2	0	11
28,8	27,798	3,55	8	28,8	0	7
33,6	35,714	6,29	6	32,9	2,04	6
57,6	62,500	8,51	3	57,6	0	3
Максим.	250	-	0	230,4	-	0
Миним.	0,977	-	255	0,9	-	255

## 10.2 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.

### 10.2.1 Асинхронный передатчик USART

Структурная схема асинхронного передатчика USART показана на рисунке 10-1. 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 устанавливается флаг прерывания TXIF (PIR1<4>). Прерывание может быть разрешено или запрещено битом TXIE (PIE1<4>). Флаг TXIF устанавливается независимо от состояния бита TXIE 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.

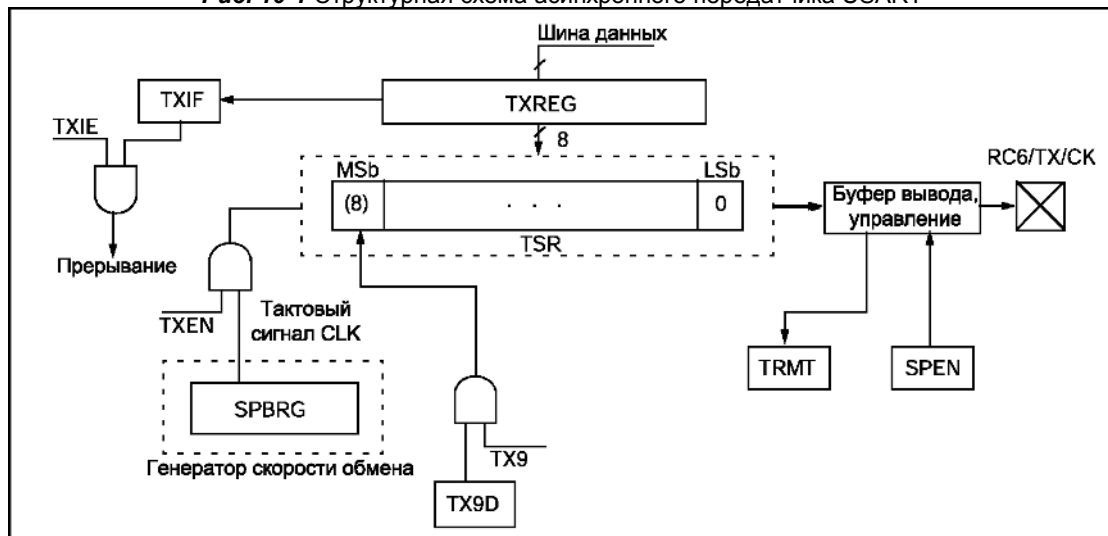
#### Примечания:

1. Регистр TSR не отображается на память и не доступен для чтения.
2. Флаг TXIF устанавливается в '1' только, когда бит TXEN=1 и сбрасывается автоматически в '0' после загрузки новых данных в регистр TXREG.

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 10-2). 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 10-3). Сброс бита TXEN в '0' вызовет немедленное прекращение передачи, сброс передатчика и перевод вывода RC6/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 (если он пуст).

Рис. 10-1 Структурная схема асинхронного передатчика USART





Рекомендованная последовательность действий для передачи данных в асинхронном режиме:

1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 10.1).
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.
8. Если используются прерывания, то биты GIE и PEIE в регистре INTCON должны быть установлены в '1'.

Рис. 10-2 Временная диаграмма асинхронной передачи данных

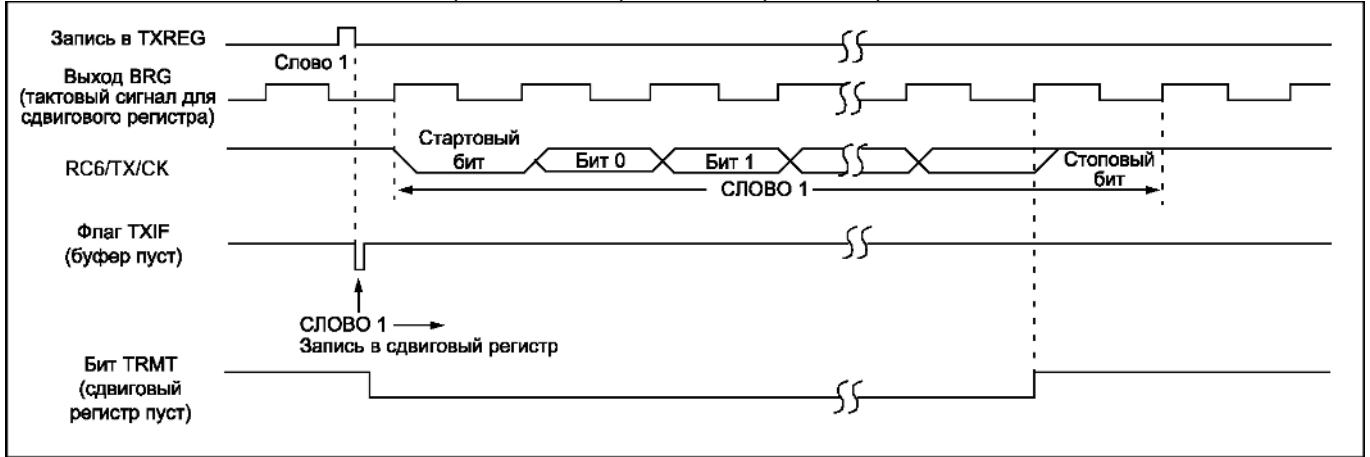


Рис. 10-3 Временная диаграмма слитной асинхронной передачи (последовательная передача двух байт)

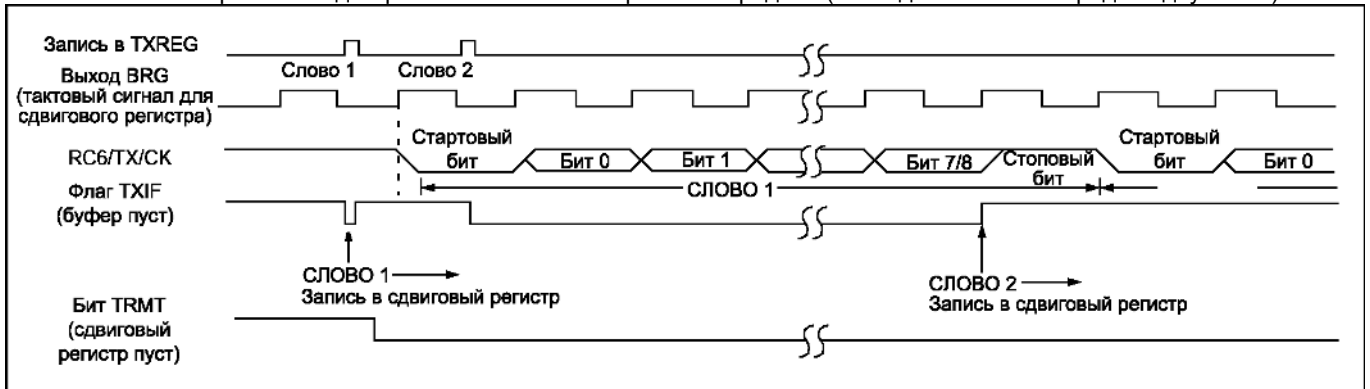


Таблица 10-5 Регистры и биты, связанные с работой передатчика USART в асинхронном режиме

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 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
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\*. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

### 10.2.2 Асинхронный приемник USART

Структурная схема асинхронного приемника USART показана на рисунке 10-4. Данные подаются на вход RC7/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 так же, как принятые данные. Рекомендуется сначала прочитать регистр RCSTA, затем RCREG, чтобы не потерять информацию RX9D и FERR.

Рис. 10-4 Структурная схема асинхронного приемника USART

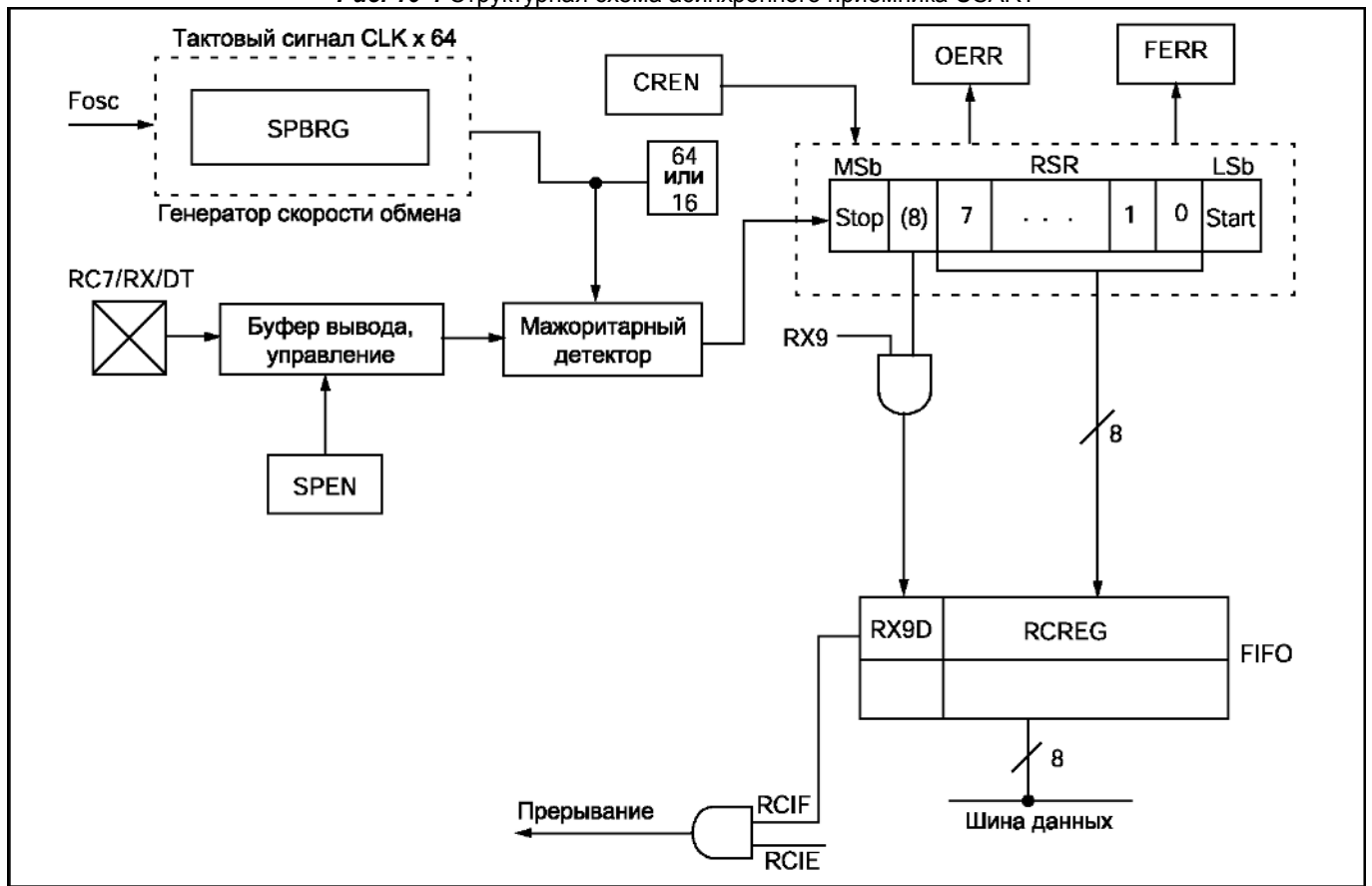
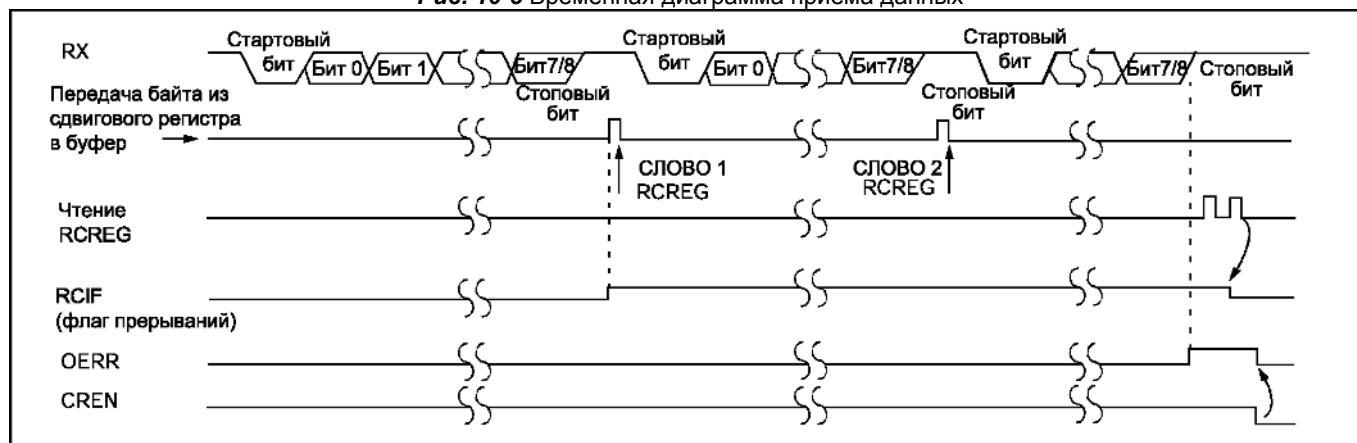


Рис. 10-5 Временная диаграмма приема данных



**Примечание к рисунку.** На временной диаграмме показан последовательный прием трех байт. Регистр RCREG (приемный буфер) читается после приема трех байт, поэтому устанавливается бит OERR в '1'.

Рекомендованные действия при приеме данных в асинхронном режиме:

1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 10.1).
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-разрядный, 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'.
10. Если используются прерывания, то биты GIE и PEIE в регистре INTCON должны быть установлены в '1'.

Таблица 10-6 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, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
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.

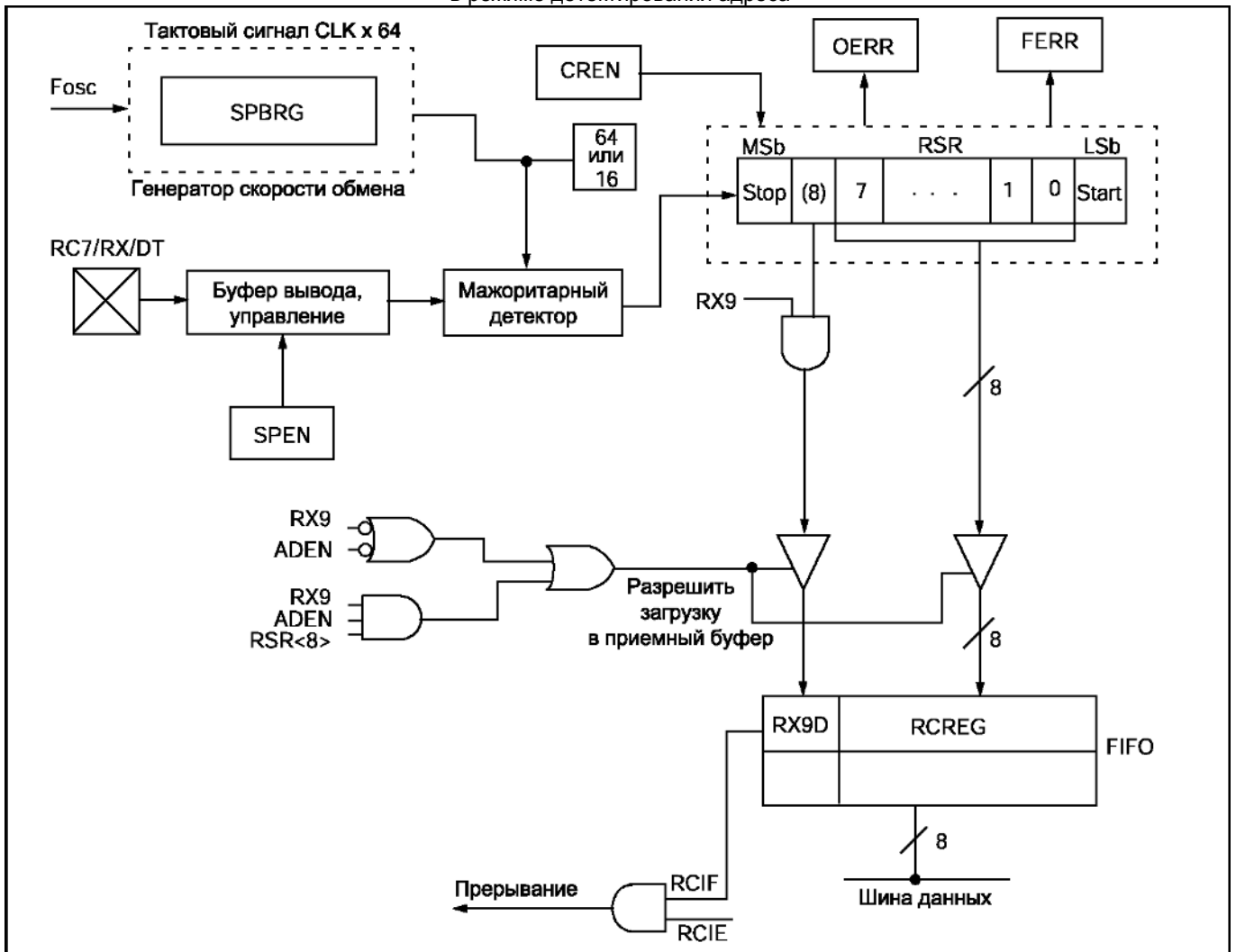
**Примечание\*.** Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

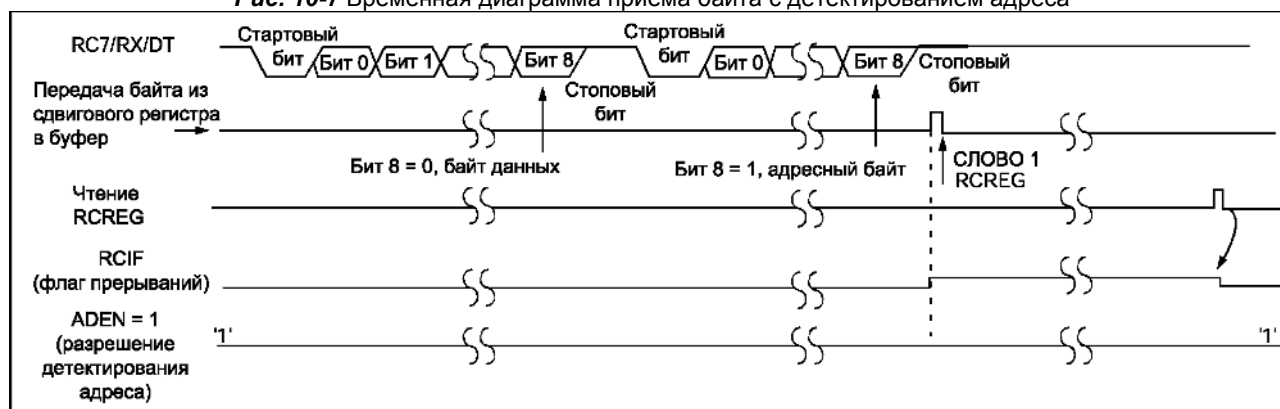
### 10.2.3 Настройка 9-разрядного асинхронного приема с детектированием адреса

Рекомендованная последовательность действия при использовании детектора адреса:

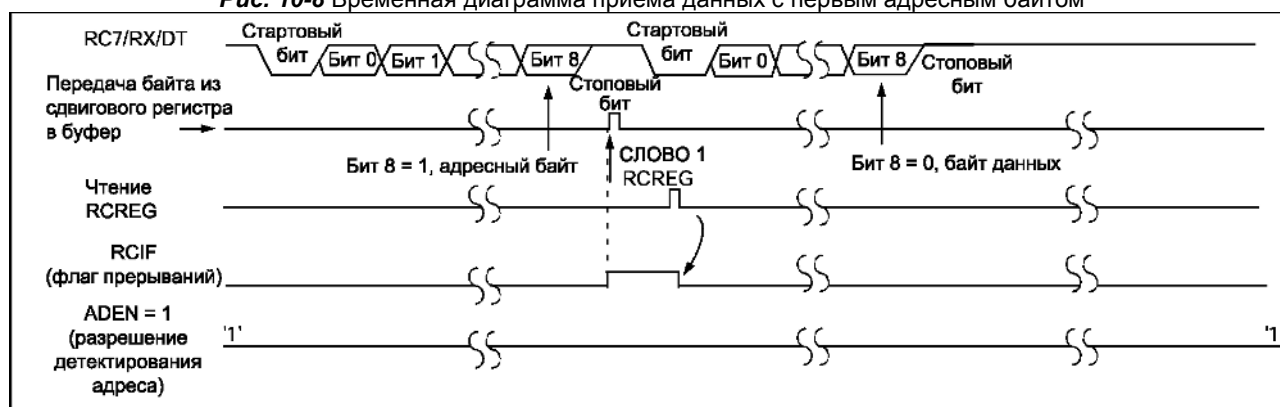
1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 10.1).
2. Выбрать асинхронный режим сбросом бита SYNC в '0' и установкой бита SPEN в '1'.
3. Если необходимо, разрешить прерывания установкой бита RCIE в '1'.
4. Установить бит RX9 в '1' для включения 9-разрядного 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.

**Рис. 10-6** Структурная схема асинхронного приемника USART в режиме детектирования адреса



**Рис. 10-7** 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. 10-8** 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.

**Таблица 10-7** 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, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
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\*.** Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

## 10.3 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 RC6/TX/CK и RC7/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'.

### 10.3.1 Передача синхронного ведущего

Структурная схема передатчика USART показана на рисунке 10-1. 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 (PIE1<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 10-9). Можно сначала загрузить данные в TXREG, и потом установить бит TXEN в '1' (см. рисунок 10-10). 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.

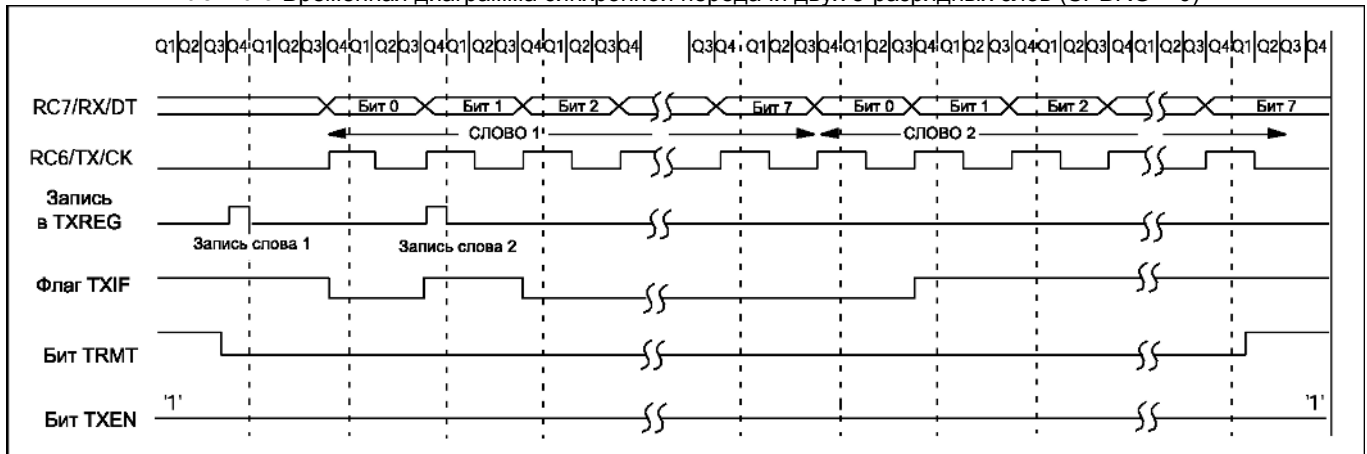
Сброс бита TXEN в '0' вызовет немедленное прекращение передачи, остановку логики передатчика и переведет выходы CK, DT в третье состояние. Установка бита CREN или SREN во время передачи вызовет ее прекращение и переведет вывод DT в третье состояние (для приема данных), а вывод CK останется выходом тактового 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, то получив слово бит SREN сбросится. 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'.

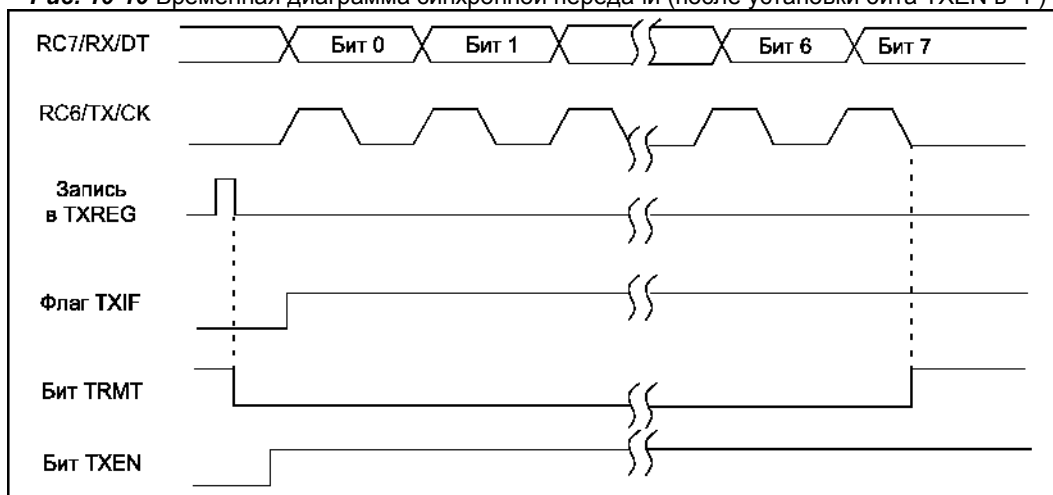
Для разрешения 9-разрядной передачи, необходимо установить бит TX9 (TXSTA<6>) в '1'. Девятый бит данных записывается в бит TX9D (TXSTA<0>). Девятый бит данных должен быть указан до записи в регистр TXREG, потому что данные, записанные в регистр TXREG, могут быть сразу загружены в сдвиговый регистр TSR.

Рекомендованная последовательность действий для передачи данных в синхронном ведущем режиме:

1. Установить требуемую скорость передачи с помощью регистра SPBRG и бита BRGH (см. раздел 10.1).
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.
8. Если используются прерывания, то биты GIE и PEIE в регистре INTCON должны быть установлены в '1'.

Рис. 10-9 Временная диаграмма синхронной передачи двух 8-разрядных слов (SPBRG = 0)



**Рис. 10-10** Временная диаграмма синхронной передачи (после установки бита TXEN в '1')**Таблица 10-8** 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, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 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
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\*.** Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

### 10.3.2 Прием синхронного ведущего

В синхронном режиме прием разрешается установкой битов CREN (RCSTA<4>) или SREN (RCSTA<5>) в '1'. Линия данных RC7/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 10.1).
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-разрядный, 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'.
11. Если используются прерывания, то биты GIE и PEIE в регистре INTCON должны быть установлены в '1'.

Рис. 10-11 Временная диаграмма синхронного приема в режиме ведущего (SREN = 1, SPBRG = 0)

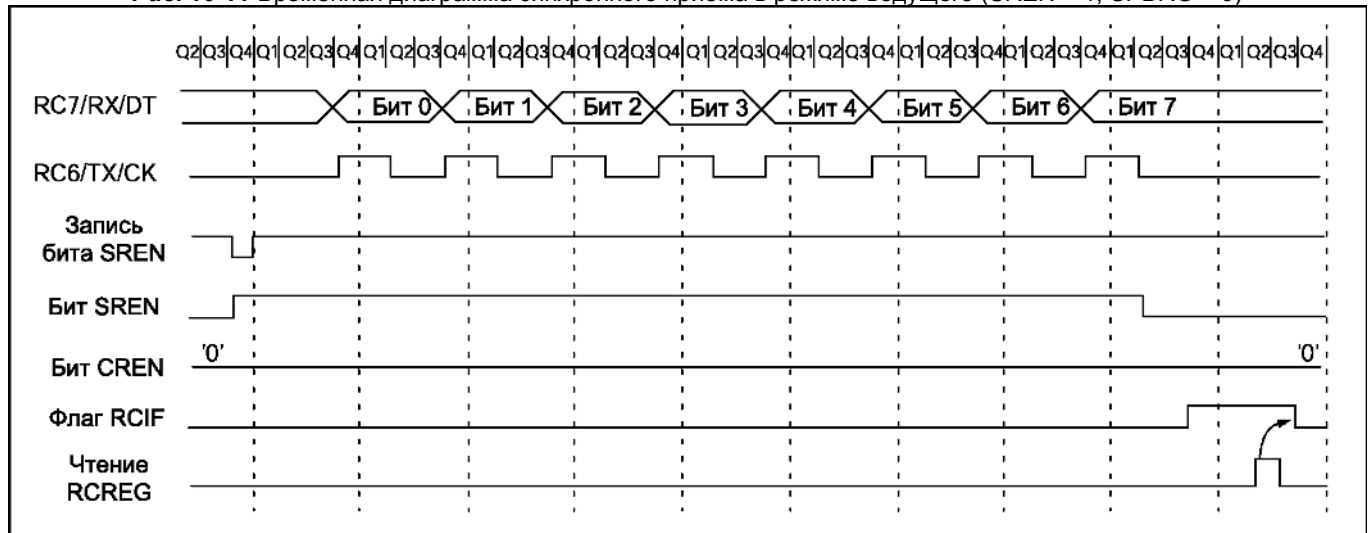


Таблица 10-9 Регистры и биты, связанные с работой приемника USART в синхронном ведущем режиме

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
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\*. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.



## 10.4 Синхронный ведомый режим USART

Режим ведомого отличается от ведущего тем, что микроконтроллер использует тактовый сигнал с входа RC6/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'.

### 10.4.1 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 SLEEP, выполняются следующие 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.
- Если используются прерывания, то биты GIE и PEIE в регистре INTCON должны быть установлены в '1'.

**Таблица 10-10** 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, BOR	Другие сбросы
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 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
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\*.** Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

### 10.4.2 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. 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-разрядный, 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'.
9. Если используются прерывания, то биты GIE и PEIE в регистре INTCON должны быть установлены в '1'.

**Таблица 10-11** 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, BOR	Другие сбросы
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
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\***. Биты PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не используются.

## 11.0 Модуль 10-разрядного АЦП

Модуль аналого-цифрового преобразования (АЦП) имеет пять каналов у 28-выводных микросхем и восемь каналов у 40/44-выводных микросхем.

Входной аналоговый сигнал через коммутатор каналов заряжает внутренний конденсатор АЦП  $C_{HOLD}$ . Модуль АЦП преобразует напряжение, удерживаемое на конденсаторе  $C_{HOLD}$  в соответствующий 10-разрядный цифровой код методом последовательного приближения. Источник верхнего и нижнего опорного напряжения может быть программно выбран с выводов  $V_{DD}$ ,  $V_{SS}$ , RA2 или RA3.

Допускается работа модуля АЦП в SLEEP режиме микроконтроллера, при этом в качестве источника тактовых импульсов для АЦП должен быть выбран RC генератор.

Для управления АЦП в микроконтроллере используется 4 регистра:

- Регистр результата ADRESH (старший байт);
- Регистр результата ADRESL (младший байт);
- Регистр управления ADCON0;
- Регистр управления ADCON1.

Регистр ADCON0 используется для настройки работы модуля АЦП, а с помощью регистра ADCON1 устанавливаются какие входы микроконтроллера будут использоваться модулем АЦП и в каком режиме (аналоговый вход или цифровой порт ввода/вывода).

Дополнительную информацию по работе с модулем АЦП смотрите в технической документации "PICmicro Mid-Range MCU Family Reference Manual" (DS33023).

### ADCON0 (адрес 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
<b>ADCS1</b>	<b>ADCS0</b>	<b>CHS2</b>	<b>CHS1</b>	<b>CHS0</b>	<b>GO/-DONE</b>	-	<b>ADON</b>
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: **ADCS1:ADCS0**: Выбор источника тактового сигнала  
 00 =  $F_{osc}/2$   
 01 =  $F_{osc}/8$   
 10 =  $F_{osc}/32$   
 11 =  $F_{RC}$  (внутренний RC генератор модуля АЦП)

биты 5-3: **CHS2:CHS0**: Выбор аналогового канала  
 000 = канал 0, (RA0/AN0)  
 001 = канал 1, (RA1/AN1)  
 010 = канал 2, (RA2/AN2)  
 011 = канал 3, (RA3/AN3)  
 100 = канал 4, (RA5/AN4)  
 101 = канал 5, (RE0/AN5)<sup>(1)</sup>  
 110 = канал 6, (RE1/AN6)<sup>(1)</sup>  
 111 = канал 7, (RE2/AN7)<sup>(1)</sup>

бит 2: **GO/-DONE**: Бит статуса модуля АЦП  
 Если **ADON=1**  
 1 = модуль АЦП выполняет преобразование (установка бита вызывает начало преобразования)  
 0 = состояние ожидания (аппаратно сбрасывается по завершению преобразования)

бит 1: **Не используется**: читается как '0'

бит 0: **ADON**: Бит включения модуля АЦП  
 1 = модуль АЦП включен  
 0 = модуль АЦП выключен и не потребляет тока

**Примечание 1.** Эти каналы не реализованы в микроконтроллерах PIC16F873/ PIC16F876.

**ADCON1 (адрес 9Fh)**

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>ADFM</b>	-	-	-	<b>PCFG3</b>	<b>PCFG2</b>	<b>PCFG1</b>	<b>PCFG0</b>
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: **ADFM**: Формат сохранения 10-разрядного результата  
1 = правое выравнивание, 6 старших бит ADRESH читаются как '0'  
0 = левое выравнивание, 6 младших бит ADRESL читаются как '0'

биты 6-4: **Не используются**: читаются как '0'

биты 3-0: **PCFG3:PCFG0**: Управляющие биты настройки каналов АЦП

PCGF3: PCGF0	AN7 <sup>(1)</sup> RE2	AN6 <sup>(1)</sup> RE1	AN5 <sup>(1)</sup> RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	V <sub>REF+</sub>	V <sub>REF-</sub>	Кан./ V <sub>REF</sub> <sup>(2)</sup>
0000	A	A	A	A	A	A	A	A	V <sub>DD</sub>	V <sub>SS</sub>	8/0
0001	A	A	A	A	V <sub>REF+</sub>	A	A	A	RA3	V <sub>SS</sub>	7/1
0010	D	D	D	A	A	A	A	A	V <sub>DD</sub>	V <sub>SS</sub>	5/0
0011	D	D	D	A	V <sub>REF+</sub>	A	A	A	RA3	V <sub>SS</sub>	4/1
0100	D	D	D	D	A	D	A	A	V <sub>DD</sub>	V <sub>SS</sub>	3/0
0101	D	D	D	D	V <sub>REF+</sub>	D	A	A	RA3	V <sub>SS</sub>	2/1
011x	D	D	D	D	D	D	D	D	V <sub>DD</sub>	V <sub>SS</sub>	0/0
1000	A	A	A	A	V <sub>REF+</sub>	V <sub>REF-</sub>	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	V <sub>DD</sub>	V <sub>SS</sub>	6/0
1010	D	D	A	A	V <sub>REF+</sub>	A	A	A	RA3	V <sub>SS</sub>	5/1
1011	D	D	A	A	V <sub>REF+</sub>	V <sub>REF-</sub>	A	A	RA3	RA2	4/2
1100	D	D	D	A	V <sub>REF+</sub>	V <sub>REF-</sub>	A	A	RA3	RA2	3/2
1101	D	D	D	D	V <sub>REF+</sub>	V <sub>REF-</sub>	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	V <sub>DD</sub>	V <sub>SS</sub>	1/0
1111	D	D	D	D	V <sub>REF+</sub>	V <sub>REF-</sub>	D	A	RA3	RA2	1/2

A = аналоговый вход      D = цифровой канал ввода/вывода

**Примечания:**

1. Эти каналы не реализованы в микроконтроллерах PIC16F873/ PIC16F876.
2. В этом столбце указывается число аналоговых каналов, доступных для выполнения преобразования, и число входов источника опорного напряжения.

В регистре ADRESH:ADRESL сохраняется 10-разрядный результат аналого-цифрового преобразования. Когда преобразование завершено, результат преобразования записывается в регистр ADRESH:ADRESL, после чего сбрасывается флаг GO/-DONE (ADCON0<2>) и устанавливается флаг прерывания ADIF. Структурная схема модуля АЦП показана на рисунке 11-1.

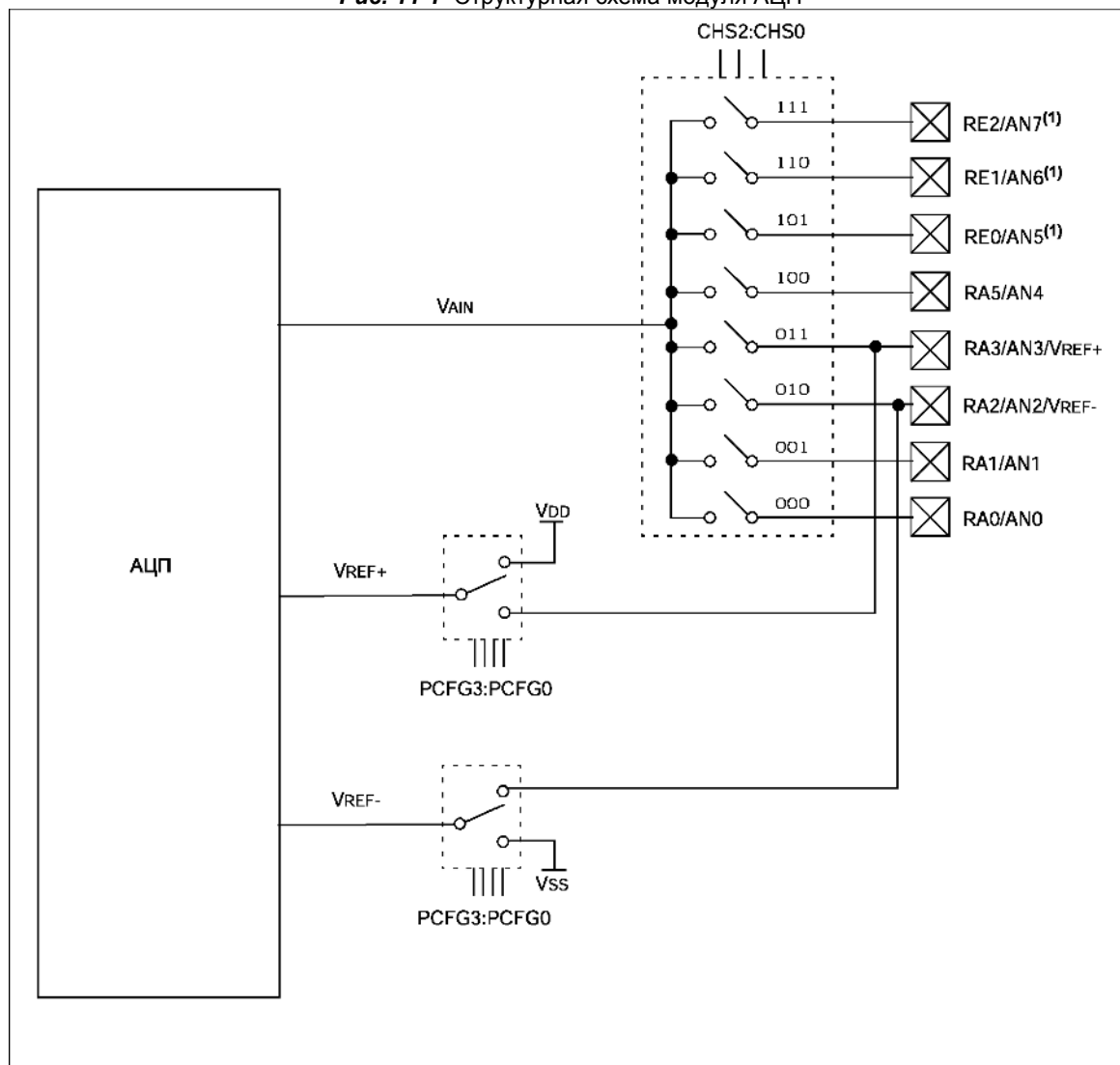
После включения и конфигурации АЦП выбирается рабочий аналоговый канал. Соответствующие биты TRIS аналоговых каналов должны настроить порт ввода/вывода на вход. Перед началом преобразования необходимо выдержать временную паузу, расчет которой приведен в разделе 11.1.

Рекомендованная последовательность действий для работы с АЦП:

1. Настроить модуль АЦП:
  - Настроить выходы как аналоговые входы, входы V<sub>REF</sub> или цифровые каналы ввода/вывода (ADCON1);
  - Выбрать входной канал АЦП (ADCON0);
  - Выбрать источник тактовых импульсов для АЦП (ADCON0);
  - Включить модуль АЦП (ADCON0).
2. Настроить прерывание от модуля АЦП (если необходимо):
  - Сбросить бит ADIF в '0';
  - Установить бит ADIE в '1';
  - Установить бит PEIE в '1';
  - Установить бит GIE в '1'.
3. Выдержать паузу, необходимую для зарядки конденсатора C<sub>HOLD</sub>.
4. Начать аналого-цифровое преобразование:
  - Установить бит GO/-DONE в '1' (ADCON0).
5. Ожидать окончания преобразования:
  - Ожидать пока бит GO/-DONE не будет сброшен в '0'; ИЛИ
  - Ожидать прерывание по окончании преобразования.

6. Считать результат преобразования из регистров ADRESH:ADRESL, сбросить бит ADIF в '0', если это необходимо.
7. Для следующего преобразования необходимо выполнить шаги начиная с пункта 1 или 2. Время преобразования одного бита определяется как время  $T_{AD}$ . Минимальное время ожидания перед следующим преобразованием должно составлять не менее  $2T_{AD}$ .

**Рис. 11-1** Структурная схема модуля АЦП



**Примечание к рисунку 1.** Эти каналы не реализованы в микроконтроллерах PIC16F873/ PIC16F876.

## 11.1 Временные требования к подключению канала АЦП

Для обеспечения необходимой точности преобразования, конденсатор  $C_{HOLD}$  должен успевать полностью заряжаться до уровня входного напряжения. Схема аналогового входа АЦП показана на рисунке 11-2. Сопротивления  $R_S$  и  $R_{SS}$  непосредственно влияют на время зарядки конденсатора  $C_{HOLD}$ . Величина сопротивления ключа выборки ( $R_{SS}$ ) зависит от напряжения питания  $V_{DD}$  (см. рисунок 11-2). Максимальное рекомендуемое значение внутреннего сопротивления источника аналогового сигнала 10кОм. При меньших значениях сопротивления источника сигнала - меньше суммарное время преобразования.

После того, как будет выбран один из нескольких аналоговых входных каналов, но прежде чем будет производиться преобразование, должно пройти определенное время. Для нахождения данного времени воспользуйтесь уравнением 11-1. Это уравнение дает результат с ошибкой в  $\frac{1}{2}$  LSB (2048 шагов АЦП). Ошибка в  $\frac{1}{2}$  LSB, это максимальная погрешность, позволяющая функционировать модулю АЦП с необходимой точностью.

Дополнительную информацию по расчету  $T_{ACQ}$  смотрите в технической документации "PICmicro Mid-Range MCU Family Reference Manual" (DS33023).

### Уравнение 11-1 Вычисление временной задержки

$$\begin{aligned} T_{ACQ} &= \text{Время задержки усилителя} + \text{Время заряда конденсатора } C_{HOLD} + \text{Температурный коэффициент} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\text{мкс} + T_C + [(Температура - 25^\circ\text{C})(0.05\text{мкс}/^\circ\text{C})] \end{aligned}$$

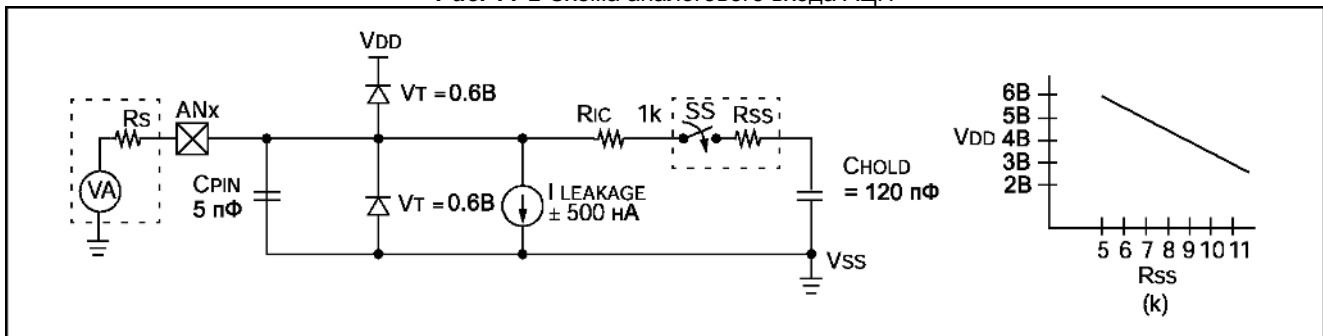
$$\begin{aligned} T_C &= - C_{HOLD} (R_{IC} + R_{SS} + R_S) \ln(1/2047) \\ &= - 120\text{пФ} (1\text{кОм} + 7\text{кОм} + 10\text{кОм}) \ln(0.0004885) \\ &= 16.47\text{мкс} \end{aligned}$$

$$\begin{aligned} T_{ACQ} &= 2\text{мкс} + 16.47\text{мкс} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05\text{мкс}/^\circ\text{C})] \\ &= 19.72\text{мкс} \end{aligned}$$

### Примечания:

1. Напряжение источника опорного напряжения  $V_{REF}$  не влияет на уравнение.
2. Конденсатор  $C_{HOLD}$  после преобразования не разряжается.
3. Максимальное рекомендуемое значение внутреннего сопротивления источника аналогового сигнала 10кОм. Это необходимо для компенсации внутреннего тока утечки.
4. После того, как преобразование завершено, необходимо программно обеспечить задержку не менее  $2.0T_{AD}$ , прежде чем начнете следующее преобразование. В течение этого времени конденсатор  $C_{HOLD}$  не подключен к выбранному входному каналу АЦП.

Рис. 11-2 Схема аналогового входа АЦП



### Обозначения:

$C_{PIN}$	- входная емкость;
$V_T$	- пороговое напряжение;
$I_{LEAKAGE}$	- ток утечки вывода;
$R_{IC}$	- сопротивление соединения;
$SS$	- переключатель защелки;
$C_{HOLD}$	- конденсатор защелки.

## 11.2 Выбор источника тактовых импульсов для АЦП

Время получения одного бита результата определяется параметром  $T_{AD}$ . Для 10-разрядного результата требуется как минимум  $12T_{AD}$ . Параметры тактового сигнала для АЦП определяются программно,  $T_{AD}$  может принимать следующие значения:

- $2T_{OSC}$ ;
- $8T_{OSC}$ ;
- $32T_{OSC}$ ;
- Внутренний RC генератор модуля АЦП (2-6мкс).

Для получения корректного результата преобразования необходимо выбрать источник тактового сигнала АЦП, обеспечивающий время  $T_{AD}$  не менее 1.6 мкс.

В таблице 11-1 указано максимальное значение тактовой частоты микроконтроллера для каждого режима синхронизирующего сигнала АЦП.

**Таблица 11-1** Максимальное значение  $F_{OSC}$  удовлетворяющие требованию к  $T_{AD}$  (для микроконтроллеров с нормальным диапазоном напряжения питания (F))

Выбор $T_{AD}$		$F_{OSC}$
Режим	ADCS1:ADCS0	Максимум
$2T_{OSC}$	00	1.25МГц
$8T_{OSC}$	01	5МГц
$32T_{OSC}$	10	20МГц
RC <sup>(1,2,3)</sup>	11	Примечание 1

### Примечания:

1. Типовое значение времени  $T_{AD}$  RC генератора АЦП равно 4мкс, может варьироваться от 2мкс до 6мкс.
2. Когда тактовая частота микроконтроллера больше 1МГц, рекомендуется использовать RC генератор АЦП только для работы в SLEEP режиме.
3. Для микроконтроллеров с расширенным диапазоном напряжений питания (LF) данные параметры смотрите в разделе электрических характеристик.

## 11.3 Настройка аналоговых входов

Регистры ADCON1, TRISA и TRISE отвечают за настройку выводов АЦП. Если выводы микросхемы настраиваются как аналоговые входы, то при этом должны быть установлены соответствующие биты в регистре TRIS. Если соответствующий бит сброшен в '0', вывод микросхемы настроен как цифровой выход со значениями выходных напряжений  $V_{OH}$  или  $V_{OL}$ .

Модуль АЦП функционирует независимо от состояния битов CHS2:CHS0 и битов TRIS.

### Примечания:

1. При чтении содержимого регистра порта нули будут установлены в тех разрядах, которые были настроены как аналоговые входы. Настроенные на цифровой вход каналы будут преобразовывать входные аналоговые уровни в цифровые, что однако не окажет влияния на точность преобразования.
2. Значения напряжений, подаваемых на выводы, настроены как аналоговые входы, включая выводы (AN7:AN0), могут влиять на ток потребления входного буфера, который может выйти за пределы значений, оговоренных в технической спецификации.

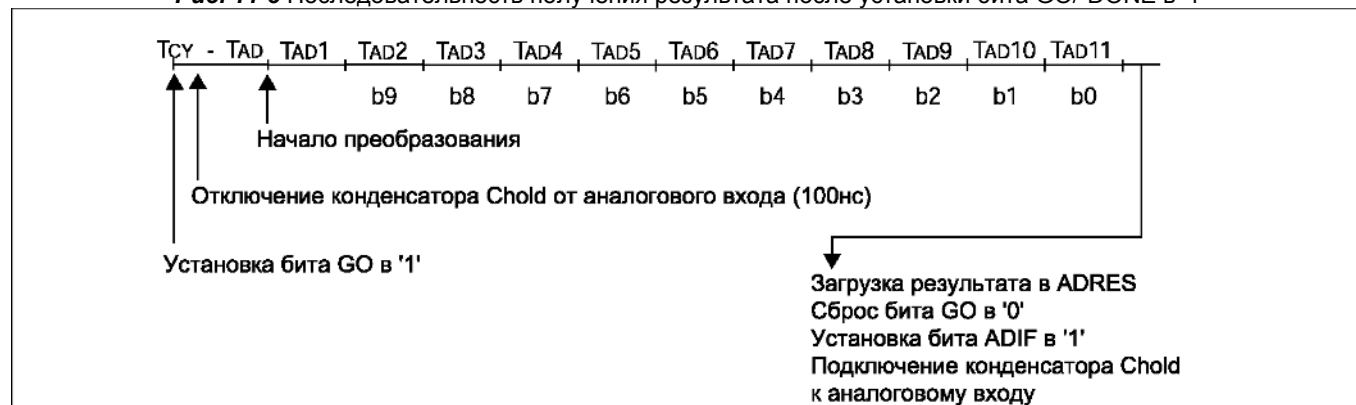
## 11.4 Аналого-цифровое преобразование

Сброс бита GO-DONE в '0' во время преобразования приведет к его прекращению. При этом регистры результата (ADRESH:ADRESL) не изменят своего содержимого. После досрочного завершения преобразования необходимо обеспечить временную задержку  $2T_{AD}$ . Выдержав требуемую паузу, можно начать новое преобразования установкой бита GO-DONE в '1'.

На рисунке 11-3 показана последовательность получения результата после установки бита GO-DONE в '1'.

**Примечание.** Бит GO-DONE и бит включения АЦП должны устанавливаться разными командами.

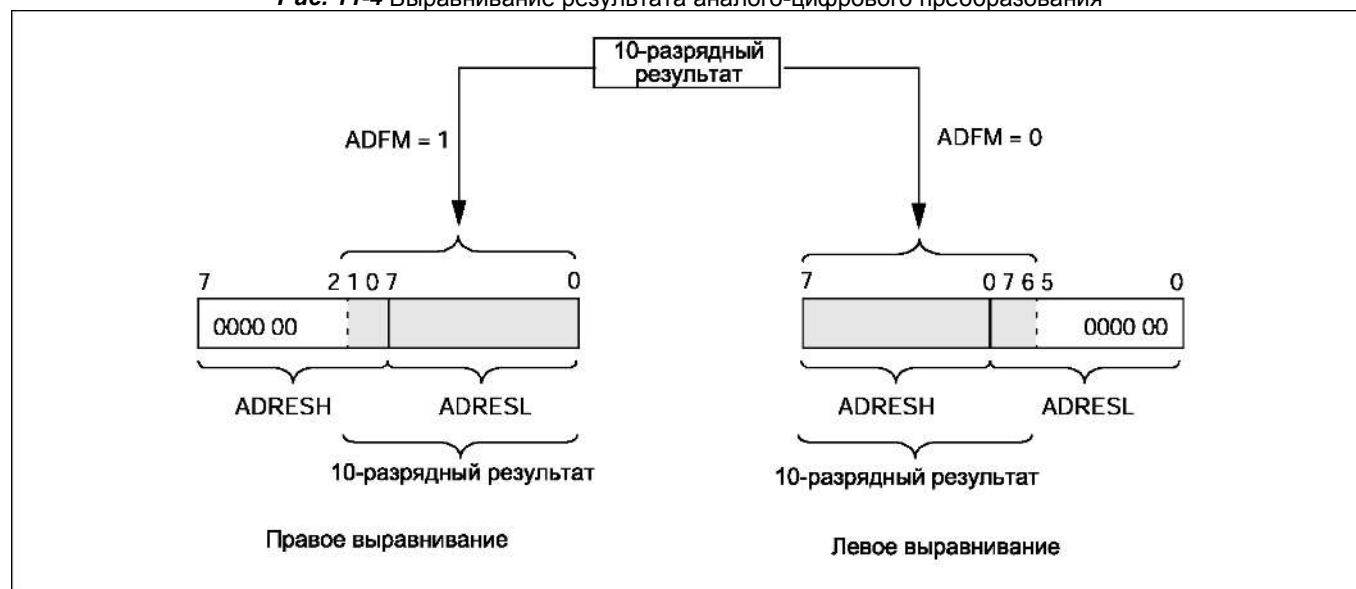
Рис. 11-3 Последовательность получения результата после установки бита GO-DONE в '1'



### 11.4.1 Выравнивание результата преобразования

10-разрядный результат преобразования сохраняется в спаренном 16-разрядном регистре ADRESH:ADRESL. Запись результата преобразования может выполняться с правым или левым выравниванием, в зависимости от значения бита ADFM (см. рисунок 11-4). Не задействованные биты регистра ADRESH:ADRESL читаются как '0'. Если модуль АЦП выключен, то 8-разрядные регистры ADRESH и ADRESL могут использоваться как регистры общего назначения.

Рис. 11-4 Выравнивание результата аналого-цифрового преобразования





## 11.5 Работа модуля АЦП в SLEEP режиме

Модуль АЦП может работать в SLEEP режиме микроконтроллера при условии, что источником импульсов преобразования АЦП будет внутренний RC генератор (ADCS1:ADCS0=11). При выборе RC генератора импульсов модуль АЦП сделает задержку в один машинный цикл перед началом преобразования. Это позволяет программе пользователя выполнить команду SLEEP, тем самым уменьшить “цифровой шум” во время преобразования. После завершения преобразования аппаратно сбрасывается бит GO/-DONE в '0', результат преобразования сохраняется в регистре ADRESH:ADRESL. Если разрешено прерывание от АЦП, то микроконтроллер выйдет из режима SLEEP. Если же прерывание было запрещено, то после преобразования модуль АЦП будет выключен, хотя бит ADON останется установленным.

Если был выбран другой источник тактовых импульсов АЦП (не внутренний RC генератор), то выполнение программой инструкции SLEEP прервет процесс преобразования и выключит модуль АЦП, оставив установленным бит ADON. Выключение модуля АЦП уменьшит ток потребления микроконтроллера.

**Примечание.** Для работы модуля АЦП в SLEEP режиме необходимо выбрать внутренний RC генератор (ADCS1:ADCS0=11), инструкция SLEEP должна быть выполнена сразу после команды, устанавливающей бит GO/-DONE в '1'.

## 11.6 Эффект сброса

При сбросе микроконтроллера значения всех его регистров устанавливаются по умолчанию. Сброс выключает модуль АЦП, а также останавливает процесс преобразования, если он был начат. Все выводы, используемые модулем АЦП, настраиваются как аналоговые входы.

Регистры ADRESH, ADRESL после сброса POR будут содержать неизвестное значение, а после остальных видов сброса не изменят своего значения.

Таблица 11-2 Регистры и биты, связанные с работой модуля АЦП

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR	Другие сбросы
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF*	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE*	ADIE	RCIE	TXIE	SSPIE	CCP1E	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	Старший байт результата преобразования								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	Младший байт результата преобразования								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/-DONE	-	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000
85h	TRISA	-	-	Регистр направления данных PORTA						--11 1111	--11 1111
05h	PORTA	-	-	Регистр защелки PORTA						--0x 0000	--0u 0000
89h*	TRISE	IBF	OBF	IBOV	PSPM	-	Регистр напр. PORTE			0000 -111	0000 -111
09h*	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -xxx	---- -uuu

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

**Примечание\*:** Эти регистры/биты в микроконтроллерах PIC16F873, PIC16F876 не реализованы.

## 12.0 Особенности микроконтроллеров PIC16F87X

В настоящее время, устройства работающие в режиме реального времени часто содержат микроконтроллер как основной элемент схемы. PIC16F87X имеют много усовершенствований повышающие надежность системы, снижающие стоимость устройства и число внешних компонентов. Микроконтроллеры PIC16F87X имеют режимы энергосбережения и возможность защиты кода программы.

Основные достоинства:

- Выбор тактового генератора
- Сброс:
  - сброс по включению питания (POR);
  - таймер включения питания (PWRT);
  - таймер запуска генератора (OSC);
  - сброс по снижению напряжения питания (BOR).
- Прерывания
- Сторожевой таймер (WDT)
- Режим энергосбережения (SLEEP)
- Защита кода программы
- Область памяти для идентификатора
- Внутрисхемное программирование по последовательному порту (ICSP)
- Режим низковольтного последовательного программирования
- Режим внутрисхемной отладки (ICD)

В микроконтроллерах PIC16F87X встроен сторожевой таймер WDT, который может быть выключен только в битах конфигурации микроконтроллера. Для повышения надежности сторожевой таймер WDT имеет собственный RC генератор.

Дополнительных 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. 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 RC позволяет уменьшить стоимость устройства, а режим LP снизить энергопотребление. Биты конфигурации микроконтроллера используются для указания режима его работы.

Дополнительную информацию по работе с микроконтроллерами PIC16F87X смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual".

### 12.1 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	DEBUG	-	WRT	CPD	LVP	BODEN	CP1	CP0	-PWRT	WDTE	FOSC1	FOSC0
-----	-----	-------	---	-----	-----	-----	-------	-----	-----	-------	------	-------	-------

Бит 13

Бит 0

биты 13-10: **CP1:CP0**: Биты защита памяти программ <sup>(2)</sup>

биты 5-4

11 = защита памяти программ выключена

10 = защищена память программ с адресами 1F00h-1FFFh (PIC16F877/876)

10 = защищена память программ с адресами 0F00h-0FFFh (PIC16F874/873)

01 = защищена память программ с адресами 1000h-1FFFh (PIC16F877/876)

01 = защищена память программ с адресами 0800h-0FFFh (PIC16F874/873)

00 = защищена память программ с адресами 0000h-1FFFh (PIC16F877/876)

00 = защищена память программ с адресами 0000h-0FFFh (PIC16F874/873)

бит 11: **DEBUG**: Бит включения режима внутрисхемной отладки

1 = внутрисхемная отладка выключена, выходы RB6 и RB7 работают как каналы ввода/вывода

0 = внутрисхемная отладка включена, выходы RB6 и RB7 используются отладчиком

бит 10: **Не реализован**: читается как '1'

бит 9: **WRT**: Бит разрешения записи во FLASH память программ

1 = разрешена запись во FLASH память программ через регистры управления EECON

0 = запрещена запись во FLASH память программ через регистры управления EECON

бит 8: **CPD**: Бит защиты EEPROM памяти данных

1 = защита памяти данных выключена

0 = защита памяти данных включена

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

1 = вывод RB3/PGM работает как PGM, режим низковольтного программирования включен

0 = вывод RB3/PGM работает как цифровой порт ввода/вывода, вывод HV используется для программирования микроконтроллера

бит 6: **BODEN**: Бит разрешения сброса по снижению напряжения питания <sup>(3)</sup>

1 = разрешен сброс BOR

0 = запрещен сброс BOR

бит 3: **-PWRT**: Бит разрешения работы таймера включения питания <sup>(3)</sup>

1 = PWRT выключен

0 = PWRT включен

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

1 = WDT включен

0 = WDT выключен

биты 1-0: **FOSC1:FOSC0**: Биты выбора режима тактового генератора

11 = RC генератор

10 = HS генератор

01 = XT генератор

00 = LP генератор

**Примечания:**

1. При стирании всей памяти микроконтроллера в слово конфигурации записывается значение 3FFFh.
2. Чтобы установить защиту памяти программ, все пары CP1:CP0 должны иметь одинаковое значение.
3. При возникновении сброса по снижению напряжения питания (BOR) автоматически запускается таймер PWRT, независимо от состояния бита -PWRT.

## 12.2 Настройка тактового генератора

### 12.2.1 Режимы тактового генератора

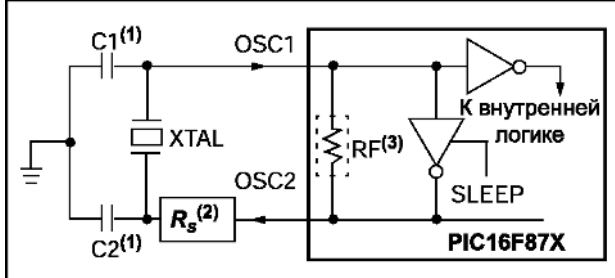
Микроконтроллеры PIC16F87X могут работать в одном из четырех режимов тактового генератора. Указать режим тактового генератора можно при программировании микроконтроллера в битах конфигурации (FOSC1:FOSC0):

- LP - низкочастотный резонатор;
- XT - обычный резонатор;
- HS - высокочастотный резонатор;
- RC - внешняя RC цепочка.

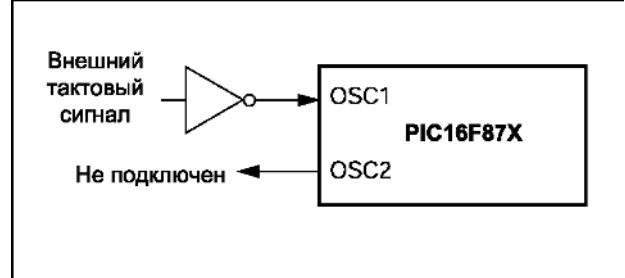
### 12.2.2 Кварцевый/керамический резонатор

В режимах тактового генератора XT, LP и HS кварцевый или керамический резонатор подключается к выводам OSC1/CLKIN, OSC2/CLKOUT (см. рисунок 12-1). Для микроконтроллеров PIC16F87X 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 12-2).

**Рис. 12-1** Подключение кварцевого/керамического резонатора в HS, XT и LP режиме тактового генератора



**Рис. 12-2** Подключение внешнего тактового сигнала в HS, XT и LP режиме тактового генератора



**Примечания:**

1. Смотрите таблицы 12-1, 12-2 для выбора емкости конденсаторов.
2. Для некоторых типов резонаторов может потребоваться последовательно включенный резистор.
3. Значение сопротивления RF варьируется в зависимости от выбранного режима генератора.

**Таблица 12-1** Параметры конденсаторов для керамического резонатора (оценочные значения)

Режим	Частота	OSC1(C1)	OSC2(C2)
XT	455 кГц	68-100пФ	68-100пФ
	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

Смотрите примечание к таблице 12-2

Резонаторы, используемые при тестировании		
455кГц	Panasonic EOF-A455K04B	±0.3%
2.0МГц	Murata Erie CDA2.00MG	±0.5%
4.0МГц	Murata Erie CDA4.00MG	±0.5%
8.0МГц	Murata Erie CDA8.00MT	±0.5%
16.0МГц	Murata Erie CDA16.00MX	±0.5%

Все резонаторы не имели паразитной емкости.

**Таблица 12-2** Параметры конденсаторов для кварцевого резонатора (оценочные значения)

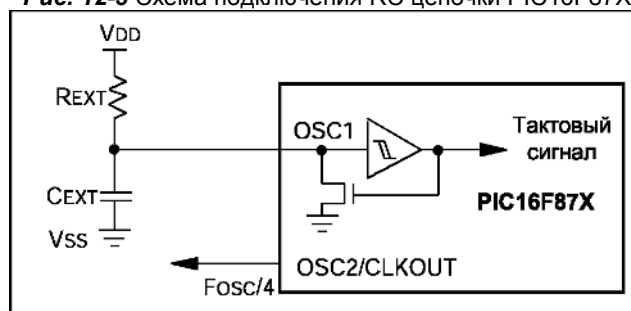
Режим	Частота	OSC1(C1)	OSC2(C2)	Резонаторы, используемые при тестировании		
LP	32 кГц	33пФ	33пФ	32кГц	Epson C-001R32.768K-A	±20PPM
	200 кГц	15пФ	15пФ	200кГц	STD XTL 200.000KHz	±20PPM
XT	200 кГц	47-68пФ	47-68пФ	1МГц	ECS ECS-10-13-1	±50PPM
	1 МГц	15пФ	15пФ	4МГц	ECS ECS-40-20-1	±50PPM
	4 МГц	15пФ	15пФ	8МГц	EPSON CA-301 8.000M-C	±30PPM
HS	4 МГц	15пФ	15пФ	20МГц	EPSON CA-301 20.000M-C	±30PPM
	8 МГц	15-33пФ	15-33пФ			
	20 МГц	15-33пФ	15-33пФ			

**Примечания:**

1. Большая емкость увеличивает стабильность генератора, но увеличивается и время запуска.
2. Значения емкости конденсаторов, указанные в таблице, являются оценочными, т.к. каждый резонатор имеет собственные характеристики. Проконсультируйтесь у производителя резонаторов для правильного подбора внешних компонентов.
3. Последовательный резистор Rs может потребоваться в HS и XT режиме для предотвращения возбуждения резонатора на низкой частоте.
4. При переходе к другим типам микроконтроллеров PICmicro необходимо проверить режим тактового генератора.

**12.2.3 RC генератор**

В приложениях, не требующей высокостабильной тактовой частоты, возможно использовать RC режим генератора, уменьшающий стоимость устройства. Частота RC генератора зависит от напряжения питания, значения сопротивления ( $R_{EXT}$ ), емкости ( $C_{EXT}$ ) и рабочей температуры. Дополнительно частота будет варьироваться в некоторых пределах из-за технологического разброс параметров кристалла. Различные паразитные емкости также будут влиять на частоту генератора, особенно при малых значениях  $C_{EXT}$ . Необходимо учитывать технологический разброс параметров внешних компонентов R и C. На рисунке 12-3 показана схема подключения RC цепочки к PIC16F87X.

**Рис. 12-3** Схема подключения RC цепочки PIC16F87X

Рекомендованные значения:  $3\text{кОм} \leq R_{EXT} \leq 100\text{кОм}$   
 $C_{EXT} > 20\text{пФ}$

### 12.3 Сброс

PIC16F87X различает следующие виды сбросов:

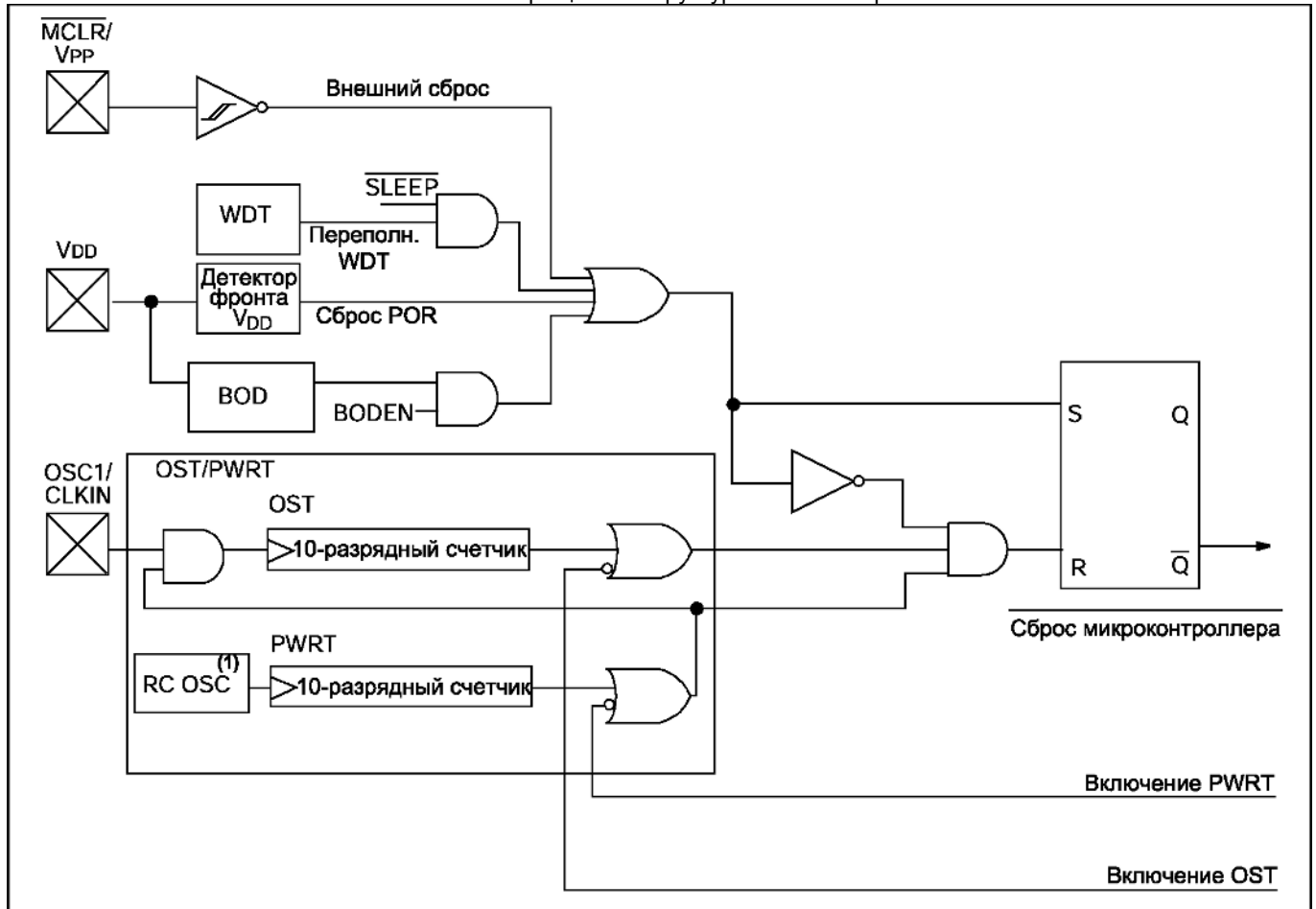
- Сброс по включению питания POR;
- Сброс по сигналу -MCLR в нормальном режиме работы;
- Сброс по сигналу -MCLR в SLEEP режиме;
- Сброс от WDT в нормальном режиме работы;
- Сброс от WDT в режиме SLEEP;
- Сброс по снижению напряжения питания BOR.

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, BOR, -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 12-4). Программное обеспечение может использовать эти биты для детектирования вида сброса микроконтроллера. Состояние регистров специального назначения после сброса смотрите в таблице 12-6.

Упрощенная структурная схема сброса показана на рисунке 12-4.

На входе -MCLR есть внутренний фильтр, не пропускающий короткие импульсы. Необходимо отметить, что сброс WDT не управляет выводом -MCLR.

Рис. 12-4 Упрощенная структурная схема сброса



Примечание 1. Это отдельный RC генератор.

## 12.4 Сброс по включению питания POR

Интегрированная схема POR удерживает микроконтроллер в состоянии сброса, пока напряжение  $V_{DD}$  не достигнет требуемого уровня (от 1.2В до 1.7В). 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".

## 12.5 Таймер включения питания PWRT

Таймер включения питания обеспечивает задержку в 72мс (номинальное значение) по сигналу схемы сброса POR. 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» параметр 33).

## 12.6 Таймер запуска генератора OST

Таймер запуска генератора обеспечивает задержку в 1024 такта генератора (вход OSC1) после окончания задержки от PWRT (если она включена). 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.

## 12.7 Сброс по снижению напряжения питания BOR

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  $V_{BOR}$ ) на время больше(или равно)  $T_{BOR}$  (см. параметр 35, 100мкс), произойдет сброс по снижению напряжения питания. Если длительность снижения напряжения питания меньше  $T_{BOR}$ , 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  $V_{BOR}$ . После нормализации напряжения питания микроконтроллер находится в состоянии сброса еще 72мс (см. параметр 33,  $T_{PWRT}$ ).

Если напряжение питания  $V_{DD}$  стало ниже  $V_{BOR}$  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  $V_{BOR}$  инициализирует PWRT, создавая задержку в 72мс. При включении схемы BOD всегда нужно включать таймер PWRT.

## 12.8 Последовательность удержания микроконтроллера в состоянии сброса

При включении питания выполняется следующая последовательность удержания микроконтроллера в состоянии сброса: сброс POR, задержка PWRT (если она разрешена), задержка OST (после завершения задержки PWRT). Полное время задержки изменяется в зависимости от режима работы тактового генератора и состояния бита -PWRTЕ.

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 12-5 показано состояние некоторых регистров специального назначения.

## 12.9 Регистр PCON

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

*Бит 0 - -BOR (детектор пониженного напряжения питания)*

Бит -BOR имеет неопределенное значение после сброса POR. Пользователь должен программно установить бит -BOR в '1' и проверять его состояние при возникающих сбросах микроконтроллера. Если -BOR =0, то произошел сброс по снижению напряжения питания (BOR). Бит -BOR 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 12-3** Время задержки при различных видах сброса

Режим генератора	Сброс POR		Сброс BOR	Выход из режима SLEEP
	-PWRTE=0	-PWRTE=1		
XT, HS, LP	72мс + 1024 T <sub>osc</sub>	1024 T <sub>osc</sub>	72мс + 1024 T <sub>osc</sub>	1024 T <sub>osc</sub>
RC	72мс	-	72мс	-

**Таблица 12-4** Состояние некоторых битов регистров STATUS/PCON

-POR	-BOR	-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	1	1	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 12-5** Состояние особых регистров после сброса

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



Таблица 12-6 Состояние регистров специального назначения после сброса

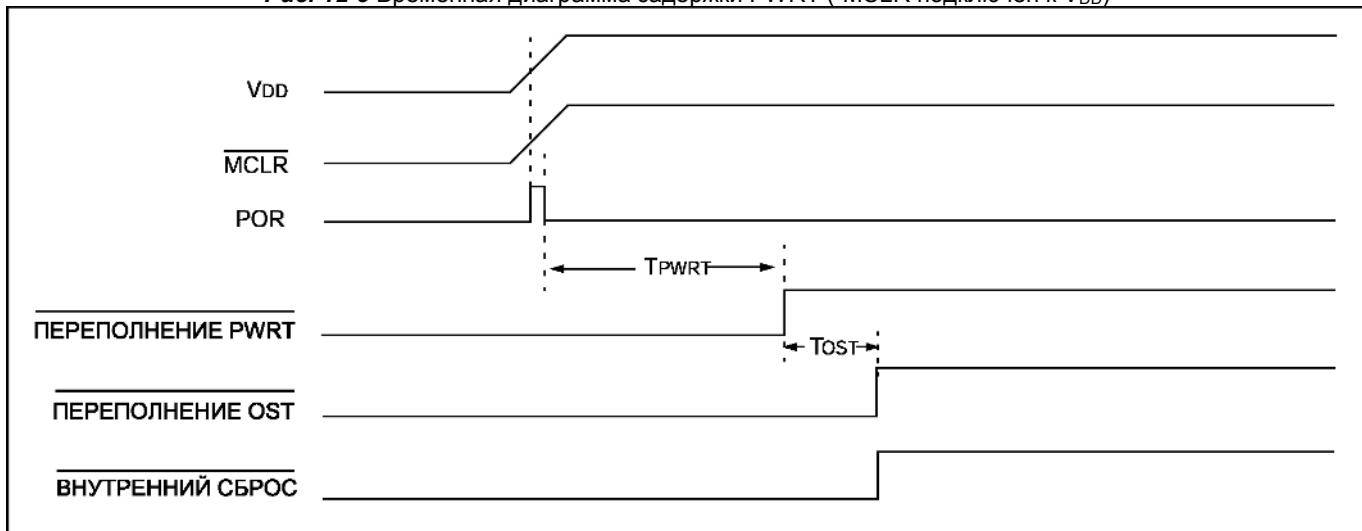
Регистр	Микроконтроллер				Сброс POR или BOR	Сброс -MCLR или WDT	Выход из режима SLEEP по прерыванию или WDT
W	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	873	874	876	877	-	-	-
TMR0	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	873	874	876	877	0000 0000	0000 0000	PC+ 1 <sup>(2)</sup>
STATUS	873	874	876	877	0001 1xxx	000q quuu <sup>(3)</sup>	uuuq quuu <sup>(3)</sup>
FSR	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	873	874	876	877	--0x 0000	--0u 0000	--uu uuuu
PORTB	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	873	874	876	877	---- -xxx	---- -uuu	---- -uuu
PCLATH	873	874	876	877	---0 0000	---0 0000	---u uuuu
INTCON	873	874	876	877	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	873	874	876	877	r000 0000	r000 0000	ruuu uuuu <sup>(1)</sup>
	873	874	876	877	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
PIR2	873	874	876	877	-r-0 0--0	-r-0 0--0	-r-u u--u <sup>(1)</sup>
TMR1L	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	873	874	876	877	--00 0000	--uu uuuu	--uu uuuu
TMR2	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
T2CON	873	874	876	877	-000 0000	-000 0000	-uuu uuuu
SSPBUF	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
CCPR1L	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	873	874	876	877	--00 0000	--uu uuuu	--uu uuuu
RCSTA	873	874	876	877	0000 000x	0000 000x	uuuu uuuu
TXREG	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
RCREG	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
CCPR2L	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	873	874	876	877	--00 0000	--uu uuuu	--uu uuuu
ADRESH	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	873	874	876	877	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
TRISA	873	874	876	877	--11 1111	--11 1111	--uu uuuu
TRISB	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
TRISC	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
TRISD	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
TRISE	873	874	876	877	0000 -111	0000 -111	uuuu -uuu
PIE1	873	874	876	877	r000 0000	r000 0000	ruuu uuuu
	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
PIE2	873	874	876	877	-r-0 0--0	-r-0 0--0	-r-u u--u
PCON	873	874	876	877	---- -qq	---- -uu	---- -uu
PR2	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
SSPADD	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	873	874	876	877	--00 0000	--00 0000	--uu uuuu
TXSTA	873	874	876	877	0000 -010	0000 -010	uuuu -uuu
SPBRG	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
ADRESL	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	873	874	876	877	0--- 0000	0--- 0000	u--- uuuu
EEDATA	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATH	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADRH	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
EECON1	873	874	876	877	x--- x000	u--- u000	u--- uuuu
EECON2	873	874	876	877	-	-	-

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

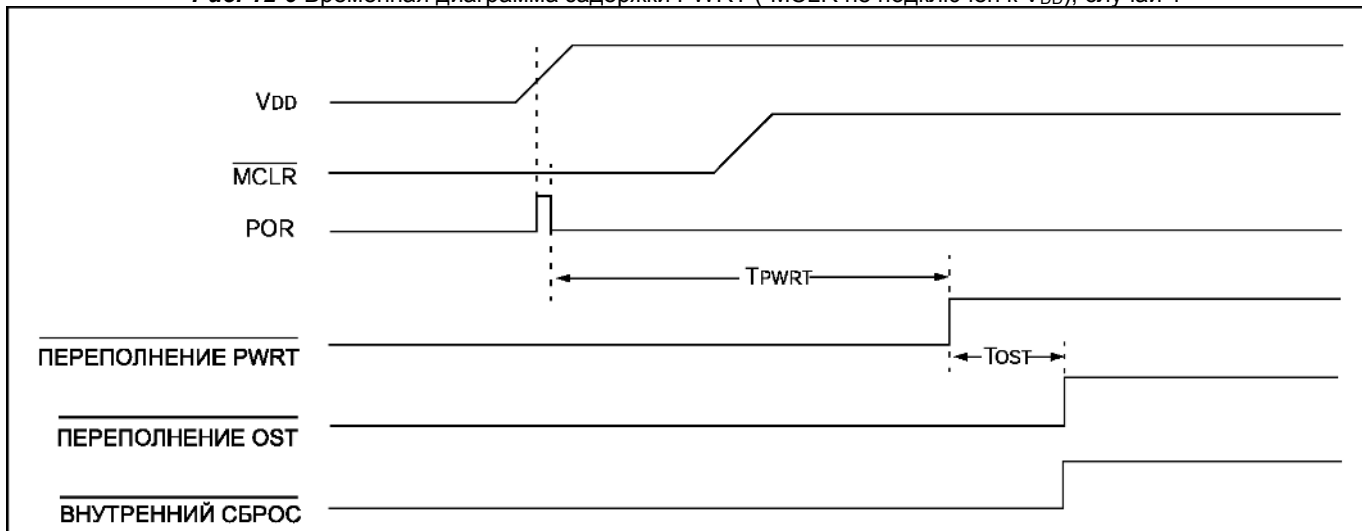
**Примечания:**

1. Один или несколько битов INTCON, PIR1 и/или будут изменены при выходе из режима SLEEP.
2. Если бит GIE=1 при выходе из режима SLEEP, в счетчик команд будет загружен вектор прерываний (0004h).
3. Смотрите в таблице 12-5 состояние битов регистра STATUS.

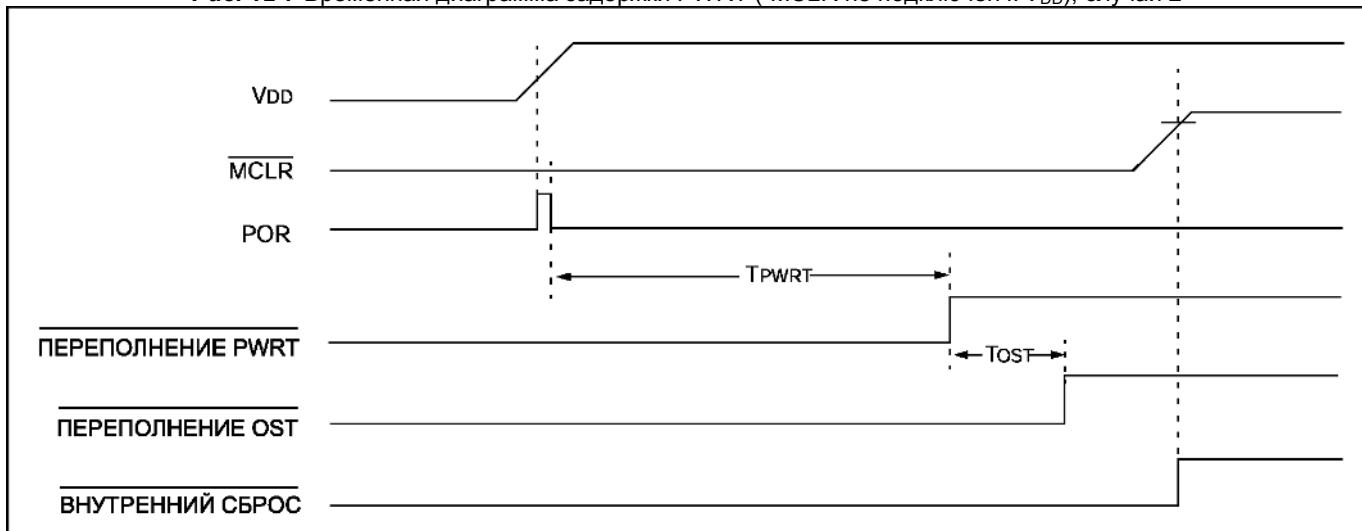
**Рис. 12-5** Временная диаграмма задержки PWRT (-MCLR подключен к V<sub>DD</sub>)

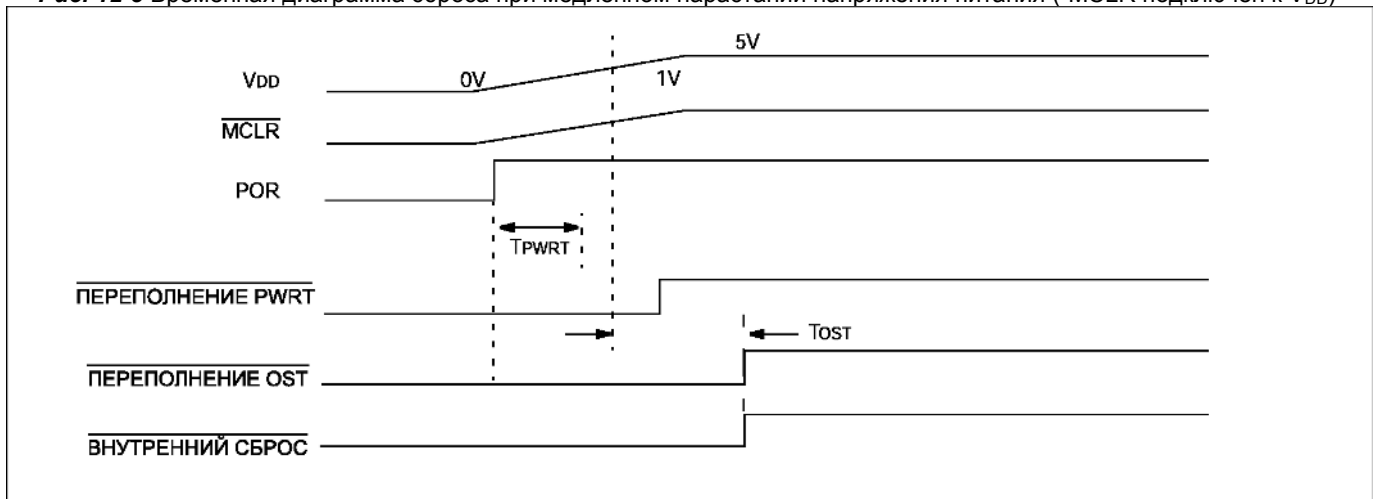


**Рис. 12-6** Временная диаграмма задержки PWRT (-MCLR не подключен к V<sub>DD</sub>), случай 1



**Рис. 12-7** Временная диаграмма задержки PWRT (-MCLR не подключен к V<sub>DD</sub>), случай 2



**Рис. 12-8** Временная диаграмма сброса при медленном нарастании напряжения питания (-MCLR подключен к V<sub>DD</sub>)

## 12.10 Прерывания

PIC16F87X имеют 14 источников прерываний. 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, PIE1 и PIE2. При сбросе микроконтроллера бит GIE сбрасывается в '0'.

При возвращении из подпрограмме 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.

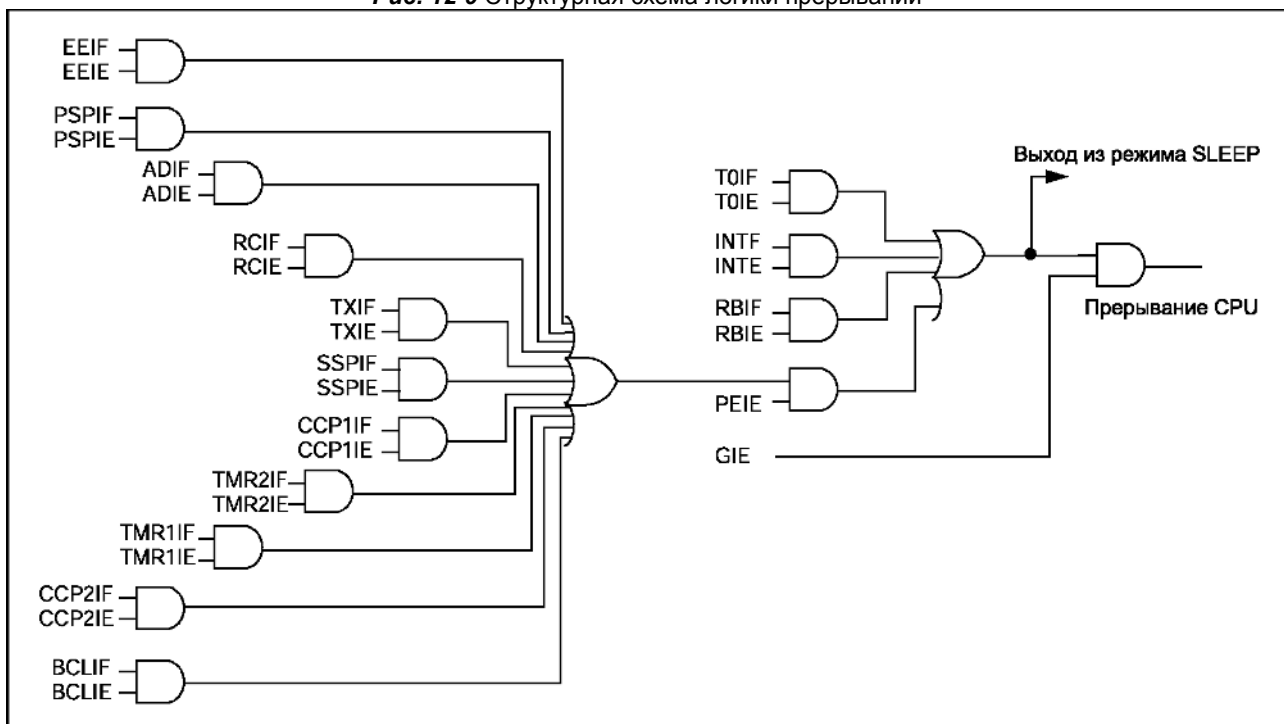
В регистрах PIR1, PIR2 содержатся флаги прерываний периферийных модулей микроконтроллера, а в регистрах PIE1, PIE2 соответствующие биты разрешения прерываний. В регистре INTCON находится бит разрешения прерываний от периферийных модулей.

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, оно одинаково для 1 и 2-х цикловых команд. Флаги прерываний устанавливаются независимо от состояния соответствующих битов маски и бита GIE.

**Примечание.** Индивидуальные флаги прерываний устанавливаются независимо от состояния соответствующих битов маски и бита GIE.

Рис. 12-9 Структурная схема логики прерываний



**Примечание.** PSPIE и PSPIF в микроконтроллерах PIC16F873, PIC16F876 не реализованы.

### 12.10.1 Внешнее прерывание с входа RB0/INT

Внешнее прерывание с входа RB0/INT происходит: по переднему фронту сигнала, если бит INTEDG (OPTION\_REG<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 на подпрограмму обработки прерываний после выхода из режима SLEEP. На рисунке 12-11 показана временная диаграмма выхода микроконтроллера из режима SLEEP по прерыванию с входа RB0/INT.

### 12.10.2 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 5.0.

### 12.10.3 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 3.2.

## 12.11 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.

Для PIC16F873/874, регистр W\_TEMP должен быть определен в обоих банках (0, 1) с одинаковым смещением относительно базового адреса банка (т.е., если регистр W\_TEMP определен в банке 0 с адресом 0x20, то он должен быть определен в банке 1 с адресом 0xA0). Регистры PCLATH\_TEMP и STATUS\_TEMP могут быть определены только в одном банке.

Т.к. старшие 16 байт каждого банка микроконтроллеров PIC16F876/877 доступны во всех банках, то регистры STATUS\_TEMP, PCLATH\_TEMP и W\_TEMP могут быть размещены в этой области. В примере 12-1 показан текст программы сохранения контекста.

**Пример 12-1** Сохранение и восстановление регистров STATUS, W и PCLATH

```

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
MOVF     PCLATH,W              ;Сохранить регистр PCLATH
MOVWF    PCLATH_TEMP           ;Сохранить регистр PCLATH
:
:
:                                ; Код программы обработки прерываний
:
MOVF     PCLATH_TEMP,W         ;Восстановить регистр PCLATH
MOVWF    PCLATH                ;Прочитать регистр STATUS_TEMP
SWAPF    STATUS_TEMP,W        ;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

```

### 12.12 Сторожевой таймер WDT

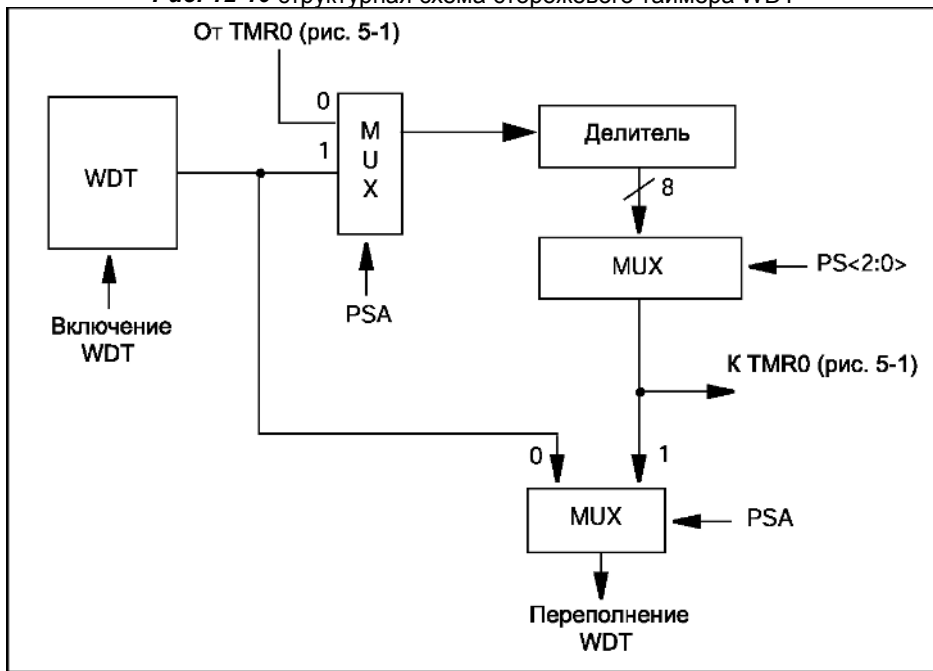
Встроенный сторожевой таймер WDT работает от отдельного RC генератора, не требующего внешних компонентов. Это позволяет работать сторожевому таймеру WDT при выключенном тактовом генераторе (выводы OSC1, OSC2) в SLEEP режиме микроконтроллера. В нормальном режиме работы при переполнении WDT происходит 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.

Время переполнения зависит от температуры, напряжения питания V<sub>DD</sub> и разброса технологических параметров микроконтроллера (см. раздел "электрические характеристики"). Если требуется большее время переполнения WDT, необходимо программно подключить предделитель в регистре OPTION\_REG с максимальным коэффициентом деления 1:128.

**Примечания:**

1. Команды CLRWDT и SLEEP сбрасывают сторожевой таймер и предделитель, если он подключен к WDT, откладывая сброс устройства.
2. Команда CLRWDT сбрасывают сторожевой таймер и предделитель, если он подключен к WDT, но не изменяет коэффициент деления предделителя.

**Рис. 12-10** структурная схема сторожевого таймера WDT



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

**Таблица 12-7** 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, BOR	Другие сбросы
2007h	Слово конф.	LVP	BOREN	CP1	CP0	-PWRTE	WDTE	FOSC1	FOSC0	uuuu uuuu	uuuu uuuu
81h,181h	OPTION_REG	-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.

## 12.13 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 -PD сбрасывается в '0', бит -TO 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 выводы 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 TOCKI 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.

### 12.13.1 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 пополнение WDT.

Список прерываний от периферийных модулей, которые могут вывести микроконтроллер из режима SLEEP:

1. Чтение/запись PSP (только для PIC16F874/877);
2. Переполнение TMR1 в режиме асинхронного счетчика;
3. Прерывание от модуля CCP;
4. Триггер специального события (TMR1 должен работать в режиме асинхронного счетчика);
5. Обнаружение START/STOP на шине I<sup>2</sup>C модулем MSSP;
6. Прием/передача байта в режиме ведомого SPI/I<sup>2</sup>C;
7. Прием/передача USART в ведомом синхронном режиме;
8. Завершение преобразования АЦП (когда используется внутренний RC генератор для АЦП);
9. Завершение записи в EEPROM.

Другие прерывания от периферийных модулей не могут вывести микроконтроллер из режима SLEEP.

При выполнении команды SLEEP происходит предвыборка следующей инструкции (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.

### 12.13.2 Выход из режима SLEEP по прерыванию

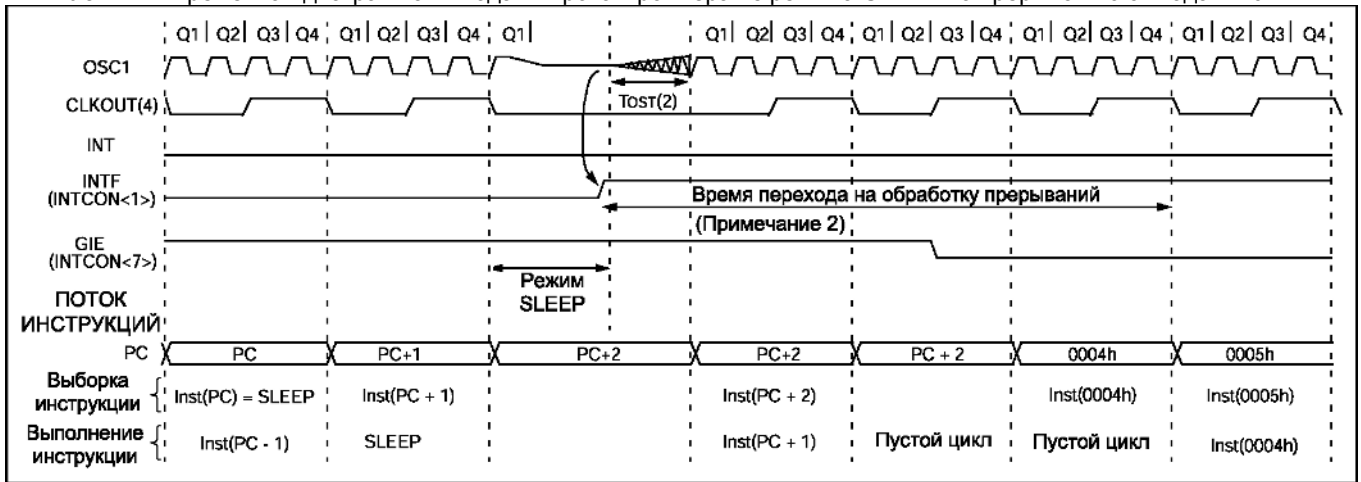
Когда бит глобального разрешения прерываний GIE сброшен в '0', а бит разрешения периферийных прерываний и соответствующий флаг прерывания установлен в '1', то возникнет одно из следующих событий:

- Если прерывание возникает перед выполнением команды SLEEP, то вместо инструкции SLEEP будет выполнен пустой цикл NOP, WDT и предделитель WDT не будут сброшены, бит -TO не будет установлен в '1', а бит -PD не будет сброшен в '0'.
- Если прерывание возникает в течение или после выполнения инструкции SLEEP, то микроконтроллер немедленно выйдет из режима SLEEP, а команда SLEEP выполняется полностью. WDT и предделитель WDT сброшены, бит -TO установлен в '1', бит -PD сброшен в '0'.

Даже если флаги прерываний были проверены перед выполнением команды SLEEP, они могут быть установлены в течение выполнения инструкции SLEEP. Для контроля полного выполнения команды SLEEP проверьте состояние бита -PD. Если -PD = 1, то вместо инструкции SLEEP был выполнен пустой цикл NOP.

Для гарантированного сброса WDT перед инструкцией SLEEP рекомендуется использовать команду CLRWDТ.

**Рис.12-11** 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). Для RC режима генератора задержка отсутствует.
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.

### 12.14 Внутрисхемный отладчик ICD

Если бит DEBUG в слове конфигурации равен нулю, то разрешен режим внутрисхемной отладки. Эта функция предоставляет простые функции отладки кода программы при использовании MPLAB ICD. Для работы отладчика используется часть ресурсов микроконтроллера, показанных в таблице 12-8.

**Таблица 12-8** Ресурсы, используемые для режима внутрисхемной отладки

Порты ввода/вывода	RB6, RB7
Стек	1 уровень
Память программ	Последние 100h слов
Память данных	0x070 (0xF0, 0x170, 0x1F0) 0x1EB - 0x1EF

Для использования режима внутрисхемной отладки схема устройства должна предусматривать возможность подключения к выводам -MCLR/VPP, VDD, GND, RB6 и RB7, аналогично режиму внутрисхемного программирования ICSP.

### 12.15 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.

### 12.16 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.



## 12.17 Внутрисхемное программирование ICSP

Микроконтроллеры PIC16F87X 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.

В режиме программирования ICSP при выполнении операции стирания всей памяти (включая снятие защиты) напряжение питания должно быть от 4.5В до 5.5В. Все остальные операции программирования могут быть выполнены во всем диапазоне напряжений питания.

Дополнительную информацию смотрите в спецификации программирования DS39025.

## 12.18 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 (не требуется подавать напряжение  $V_{IN}$  на вывод -MCLR). По умолчанию LVP=1, разрешая низковольтное программирование. При этом вывод RB3/PGM используется для низковольтного программирования и перестает быть цифровым портом ввода/вывода. Микроконтроллер переходит в режим программирования, когда на выводе RB3/PGM высокий уровень сигнала. Режим стандартного программирования по прежнему доступен (когда на выводе -MCLR напряжение  $V_{IN}$ ).

### Примечания:

1. Режим стандартного программирования всегда доступен, независимо от состояния бита LVP.
2. В режиме низковольтного программирования вывод RB3/PGM не может использоваться как цифровой порт ввода/вывода.
3. В режиме низковольтного программирования бит TRISB<3> должен быть сброшен в '0' для отключения подтягивающего резистора на входе.
4. В режиме низковольтного программирования вывод RB3/PGM не должен оставаться неподключенным. Если на RB3 высокий уровень сигнала, то микроконтроллер находится в режиме программирования.
5. Режим низковольтного программирования разрешен по умолчанию. Для выключения режима низковольтного программирования бит LVP в слове конфигурации должен равняться нулю.
6. Выключение режима LVP обеспечивает максимальную совместимость с микроконтроллерами PIC16CXXX.

Если режим низковольтного программирования не используется, бит LVP должен быть сброшен в '0', вывод RB3/PGM становится цифровым портом ввода/вывода. Бит LVP может быть изменен только в стандартном режиме программирования (когда на выводе -MCLR напряжение  $V_{IN}$ ). Когда бит LVP=0, возможен только стандартный режим программирования/проверки микроконтроллера.

В режиме программирования ICSP при выполнении операции стирания всей памяти (включая снятие защиты) напряжение питания должно быть от 4.5В до 5.5В. Все остальные операции программирования могут быть выполнены во всем диапазоне напряжений питания.

## 13.0 Система команд

Каждая команда микроконтроллеров PIC16F87X 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 13-2. Команды разделены на следующие группы: байт ориентированные команды, бит ориентированные команды, команды управления и операций с константами. Описание полей кода операции смотрите в таблице 13-1.

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 истинный 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 13-2. На рисунке 13-1 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 13-1** 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
Курсив	Термин, определяемый пользователем

Рис 13-1 Форма команд трех основных групп



Таблица 13-2 Список команд микроконтроллеров PIC16F87X

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

**Примечание.** Дополнительное описание команд микроконтроллеров PIC16F87X смотрите в технической документации DS33023 "PICmicro™ Mid-Range Reference Manual".

### 13.1 Подробное описание команд

<b>ADDLW</b>		<b>Сложить константу с W</b>	
Синтаксис:	<i>[label]</i>	ADDLW	k
Операнды:	$0 \leq k \leq 255$		
Операция:	$(W) + k \rightarrow (W)$		
Измен. флаги:	C, DC, Z		
Описание:	Содержимое регистра W складывается с 8-разрядной константой 'k'. Результат сохраняется в регистре W.		
<b>ADDWF</b>		<b>Сложение W и f</b>	
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		
Описание:	Сложить содержимое регистров W и 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.		
<b>ANDLW</b>		<b>Побитное 'И' константы и W</b>	
Синтаксис:	<i>[label]</i>	ANDLW	k
Операнды:	$0 \leq k \leq 255$		
Операция:	$(W) .AND. k \rightarrow (W)$		
Измен. флаги:	Z		
Описание:	Выполняется побитное 'И' содержимого регистра W и 8-разрядной константы 'k'. Результат сохраняется в регистре W.		
<b>ANDWF</b>		<b>Побитное 'И' W и f</b>	
Синтаксис:	<i>[label]</i>	ANDWF	f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$		
Операция:	$(W) .AND. (f) \rightarrow (dest)$		
Измен. флаги:	Z		
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'.		
<b>BCF</b>		<b>Очистить бит b в регистре f</b>	
Синтаксис:	<i>[label]</i>	BCF	f,b
Операнды:	$0 \leq f \leq 127$ $0 \leq b \leq 7$		
Операция:	$0 \rightarrow (f<b>)$		
Измен. флаги:	Нет		
Описание:	Очистить бит 'b' в регистре 'f'.		
<b>BSF</b>		<b>Установить бит b в регистре f</b>	
Синтаксис:	<i>[label]</i>	BSF	f,b
Операнды:	$0 \leq f \leq 127$ $0 \leq b \leq 7$		
Операция:	$1 \rightarrow (f<b>)$		
Измен. флаги:	Нет		
Описание:	Установить бит 'b' в регистре 'f'.		

<b>BTFS</b>	<b>Проверить бит b в регистре f, пропустить если 0</b>
Синтаксис:	<i>[label]</i> BTFS f,b
Операнды:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Операция:	пропустить если $(f < b) = 0$
Измен. флаги:	Нет
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.
<b>BTFS</b>	<b>Проверить бит b в регистре f, пропустить если 1</b>
Синтаксис:	<i>[label]</i> BTFS f,b
Операнды:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Операция:	пропустить если $(f < b) = 1$
Измен. флаги:	Нет
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.
<b>CALL</b>	<b>Вызов подпрограммы</b>
Синтаксис:	<i>[label]</i> CALL k
Операнды:	$0 \leq k \leq 2047$
Операция:	$(PC) + 1 \rightarrow TOS,$ $k \rightarrow PC<10:0>,$ $(PCLATH<4:3>) \rightarrow PC<12:11>$
Измен. флаги:	Нет
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.
<b>CLRF</b>	<b>Очистить f</b>
Синтаксис:	<i>[label]</i> CLRF f
Операнды:	$0 \leq f \leq 127$
Операция:	$00h \rightarrow (f)$ $1 \rightarrow Z$
Измен. флаги:	Z
Описание:	Очистить содержимое регистра 'f' и установить флаг Z
<b>CLRW</b>	<b>Очистить W</b>
Синтаксис:	<i>[label]</i> CLRW
Операнды:	Нет
Операция:	$00h \rightarrow (W)$ $1 \rightarrow Z$
Измен. флаги:	Z
Описание:	Очистить содержимое регистра W и установить флаг Z

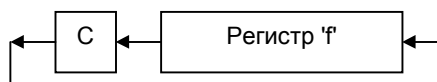
<b>CLRWDТ</b>	<b>Очистить WDT</b>
Синтаксис:	<i>[label]</i> CLRWDТ
Операнды:	Нет
Операция:	00h → WDT, 00h → предделитель WDT, 1 → -TO 1 → -PD
Измен. флаги:	-TO, -PD
Описание:	Инструкция CLRWDТ сбрасывает WDT и предделитель, если он подключен к WDT. В регистре STATUS устанавливает биты -TO и -PD.
<b>COMF</b>	<b>Инвертировать f</b>
Синтаксис:	<i>[label]</i> COMF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	(-f) → (dest)
Измен. флаги:	Z
Описание:	Инвертировать все биты в регистре 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.
<b>DECF</b>	<b>Вычесть 1 из f</b>
Синтаксис:	<i>[label]</i> DECF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	(f) - 1 → (dest)
Измен. флаги:	Z
Описание:	Декрементировать содержимое регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.
<b>DECFSZ</b>	<b>Вычесть 1 из f и пропустить если 0</b>
Синтаксис:	<i>[label]</i> DECFSZ f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	(f) - 1 → (dest); пропустить если результат равен 0
Измен. флаги:	Нет
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.
<b>GOTO</b>	<b>Безусловный переход</b>
Синтаксис:	<i>[label]</i> GOTO k
Операнды:	$0 \leq k \leq 2047$
Операция:	$k \rightarrow PC<10:0>$ , (PCLATH<4:3>) → PC<12:11>
Измен. флаги:	Нет
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.

<b>INCF</b>	<b>Прибавить 1 к f</b>
Синтаксис:	<i>[label]</i> INCF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	$(f) + 1 \rightarrow (\text{dest})$
Измен. флаги:	Z
Описание:	Инкрементировать содержимое регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.
<b>INCFSZ</b>	<b>Прибавить 1 к f и пропустить если 0</b>
Синтаксис:	<i>[label]</i> INCFSZ f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	$(f) + 1 \rightarrow (\text{dest})$ ; пропустить если результат равен 0
Измен. флаги:	Нет
Описание:	Инкрементировать 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.
<b>IORLW</b>	<b>Побитное 'ИЛИ' константы и W</b>
Синтаксис:	<i>[label]</i> IORLW k
Операнды:	$0 \leq k \leq 255$
Операция:	$(W) .OR. k \rightarrow (W)$
Измен. флаги:	Z
Описание:	Выполняется побитное 'ИЛИ' содержимого регистра W и 8-разрядной константы 'k'. Результат сохраняется в регистре W.
<b>IORWF</b>	<b>Побитное 'ИЛИ' W и f</b>
Синтаксис:	<i>[label]</i> IORWF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	$(W) .OR. (f) \rightarrow (\text{dest})$
Измен. флаги:	Z
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'.
<b>MOVF</b>	<b>Переслать f</b>
Синтаксис:	<i>[label]</i> MOVF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	$(f) \rightarrow (\text{dest})$
Измен. флаги:	Z
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.

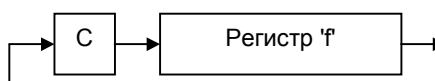
<b>MOVLW</b>	<b>Переслать константу в W</b>
Синтаксис:	<i>[label]</i> MOVLW k
Операнды:	$0 \leq k \leq 255$
Операция:	$k \rightarrow (W)$
Измен. флаги:	Нет
Описание:	Переслать константу 'k' в регистр W. В неиспользуемых битах ассемблер устанавливает '0'.
<b>MOVWF</b>	<b>Переслать W в f</b>
Синтаксис:	<i>[label]</i> MOVWF f
Операнды:	$0 \leq f \leq 127$
Операция:	$(W) \rightarrow (f)$
Измен. флаги:	Нет
Описание:	Переслать содержимое регистра W в регистр 'f'.
<b>NOP</b>	<b>Нет операции</b>
Синтаксис:	<i>[label]</i> NOP
Операнды:	Нет
Операция:	Нет операции
Измен. флаги:	Нет
Описание:	Нет операции
<b>RETFIE</b>	<b>Возврат из подпрограммы с разрешением прерываний</b>
Синтаксис:	<i>[label]</i> RETFIE
Операнды:	Нет
Операция:	TOS $\rightarrow$ PC 1 $\rightarrow$ GIE
Измен. флаги:	Нет
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.
<b>RETLW</b>	<b>Возврат из подпрограммы с загрузкой константы в W</b>
Синтаксис:	<i>[label]</i> RETLW k
Операнды:	$0 \leq k \leq 255$
Операция:	$k \rightarrow (W)$ TOS $\rightarrow$ PC
Измен. флаги:	Нет
Описание:	В регистр W загружается 8-разрядная константа. Вершина стека TOS загружается в счетчик команд PC. Инструкция выполняется за 2 цикла.
<b>RETURN</b>	<b>Возврат из подпрограммы</b>
Синтаксис:	<i>[label]</i> RETURN
Операнды:	Нет
Операция:	TOS $\rightarrow$ PC
Измен. флаги:	Нет
Описание:	Возврат из подпрограммы. Вершина стека TOS загружается в счетчик команд PC. Инструкция выполняется за 2 цикла.



<b>RLF</b>	<b>Циклический сдвиг f влево через перенос</b>
Синтаксис:	<i>[label]</i> RLF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	См. описание
Измен. флаги:	C
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'.



<b>RRF</b>	<b>Циклический сдвиг f вправо через перенос</b>
Синтаксис:	<i>[label]</i> RRF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	См. описание
Измен. флаги:	C
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'.



<b>SLEEP</b>	<b>Перейти в режим SLEEP</b>
Синтаксис:	<i>[label]</i> SLEEP
Операнды:	Нет
Операция:	00h → WDT 00h → предделитель WDT 1 → - TO 0 → - PD
Измен. флаги:	-TO, -PD
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.

<b>SUBLW</b>	<b>Вычесть W из константы</b>
Синтаксис:	<i>[label]</i> SUBLW k
Операнды:	$0 \leq k \leq 255$
Операция:	$k - (W) \rightarrow (W)$
Измен. флаги:	C, DC, Z
Описание:	Вычесть содержимое регистра W из 8-разрядной константы 'k'. Результат сохраняется в регистре W.

<b>SUBWF</b>	<b>Вычитать W из f</b>
Синтаксис:	<i>[label]</i> SUBWF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	$(f) - (W) \rightarrow (dest)$
Измен. флаги:	C, DC, Z
Описание:	Вычитать содержимое регистра W из регистра 'f'. Если d=0, результат сохраняется в регистре W. Если d=1, результат сохраняется в регистре 'f'.
<b>SWAPF</b>	<b>Поменять местами полубайты в регистре f</b>
Синтаксис:	<i>[label]</i> SWAPF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	$(f<3:0>) \rightarrow (dest<7:4>)$ $(f<7:4>) \rightarrow (dest<3:0>)$
Измен. флаги:	Нет
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'.
<b>XORLW</b>	<b>Побитное 'исключающее ИЛИ' константы и W</b>
Синтаксис:	<i>[label]</i> XORLW k
Операнды:	$0 \leq k \leq 255$
Операция:	$(W) .XOR. k \rightarrow (W)$
Измен. флаги:	Z
Описание:	Выполняется побитное 'исключающее ИЛИ' содержимого регистра W и 8-разрядной константы 'k'. Результат сохраняется в регистре W.
<b>XORWF</b>	<b>Побитное 'исключающее ИЛИ' W и f</b>
Синтаксис:	<i>[label]</i> XORWF f,d
Операнды:	$0 \leq f \leq 127$ $d \in [0,1]$
Операция:	$(W) .XOR. (f) \rightarrow (dest)$
Измен. флаги:	Z
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'.

## 14.0 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;
  - ICEPIC.
- Внутрисхемный отладчик:
  - MLAB-ICD для микроконтроллеров семейства PIC16F87X.
- Программаторы:
  - Универсальный программатор PRO MATE II;
  - Недорогой программатор PICSTART для начала работы с PICmicro.
- Недорогие демонстрационные платы:
  - PICDEM-1;
  - PICDEM-2;
  - PICDEM-3;
  - PICDEM-17;
  - KeeLoq.

### 14.1 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.

## 14.2 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.

## 14.3 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.

## 14.4 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.

## 14.5 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.

## 14.6 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.

## 14.7 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.

## 14.8 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 дает возможность быстро выполнить отладку программы, выполняя ее по шагам или в режиме реального времени.

## 14.9 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.

## 14.10 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.

## 14.11 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.

## 14.12 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 I<sup>2</sup>C, выводы для подключения ЖКИ и дополнительной клавиатуры.

## 14.13 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.

#### **14.14 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.

#### **14.15 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.



## 15.0 Электрические характеристики

### Максимально допустимые значения (\*)

Предельная рабочая температура .....	от -55°C до +125°C
Температура хранения .....	от -65°C до +150°C
Напряжение $V_{DD}$ относительно $V_{SS}$ .....	от -0.3В до +7.5В
Напряжение -MCLR относительно $V_{SS}$ .....	от 0В до +14В
Напряжение RA4 относительно $V_{SS}$ .....	от 0В до +8.5В
Напряжение на остальных выводах относительно $V_{SS}$ .....	от -0.3В до $V_{DD}+0.3В$
Рассеиваемая мощность <sup>(1)</sup> .....	1Вт
Максимальный ток вывода $V_{SS}$ .....	300мА
Максимальный ток вывода $V_{DD}$ .....	250мА
Входной запирающий ток $I_{IK}$ ( $V_I < 0$ или $V_I > V_{DD}$ ) .....	+20мА
Выходной запирающий ток $I_{OK}$ ( $V_O < 0$ или $V_O > V_{DD}$ ) .....	+20мА
Максимальный выходной ток стока канала ввода/вывода .....	25мА
Максимальный выходной ток истока канала ввода/вывода .....	25мА
Максимальный выходной ток стока портов ввода/вывода PORTA, PORTB и PORTE .....	200мА
Максимальный выходной ток истока портов ввода/вывода PORTA, PORTB и PORTE .....	200мА
Максимальный выходной ток стока портов ввода/вывода PORTC и PORTD .....	200мА
Максимальный выходной ток истока портов ввода/вывода PORTC и PORTD .....	200мА

**Примечание 1.** Потребляемая мощность рассчитывается по формуле:

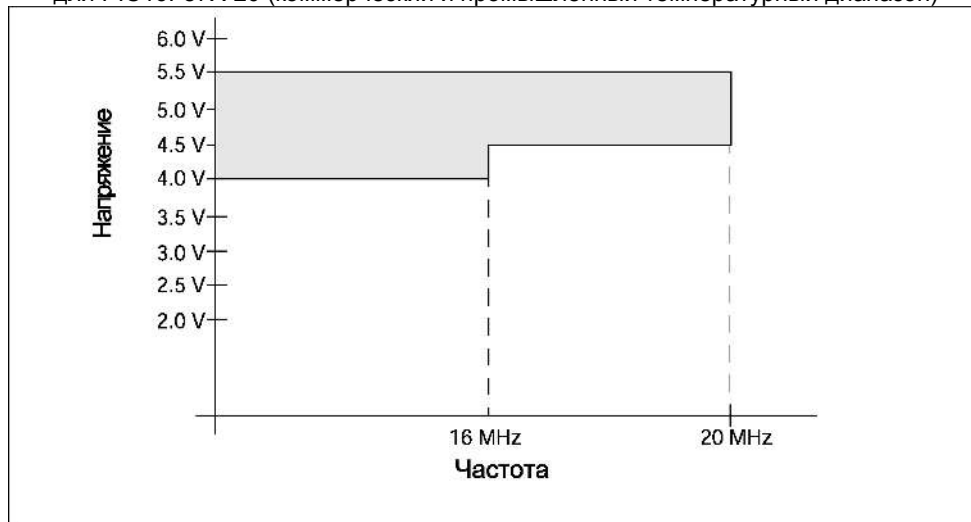
$$P = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

**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 недопустимых условиях может повлиять на его надежность.

**Примечание.** PORTD, PORTE не реализованы в микроконтроллерах PIC16F873/876.

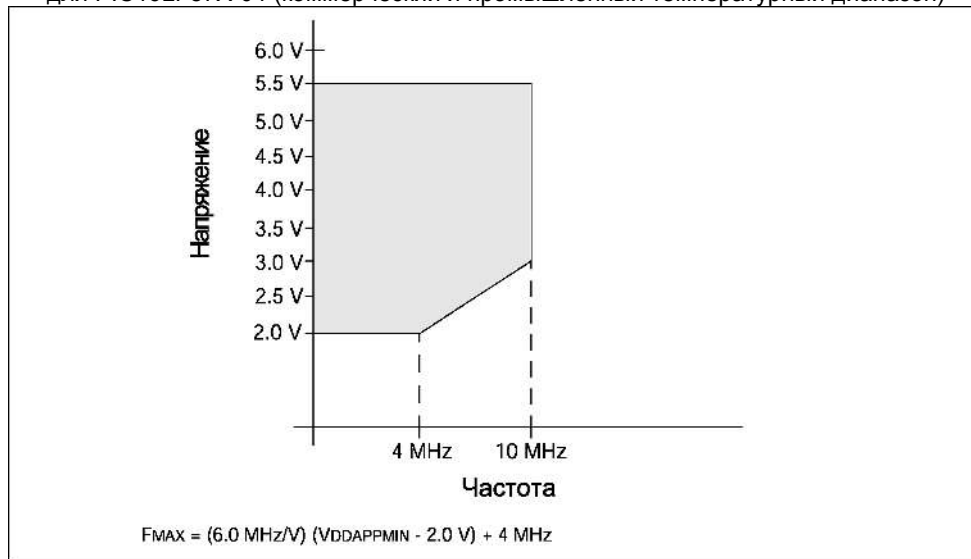
**Примечание.** Броски напряжения на выводе -MCLR ниже  $V_{SS}$  приводят к появлению больших токов (около 80мА), что может привести к срабатыванию защелки. Поэтому рекомендуется последовательно включать резистор сопротивлением от 500Ом до 1000Ом для подачи низкого уровня на этот вывод вместо непосредственного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

**Рис. 15-1** График рекомендованных комбинаций значений напряжения питания и тактовой частоты для PIC16F87X-20 (коммерческий и промышленный температурный диапазон)





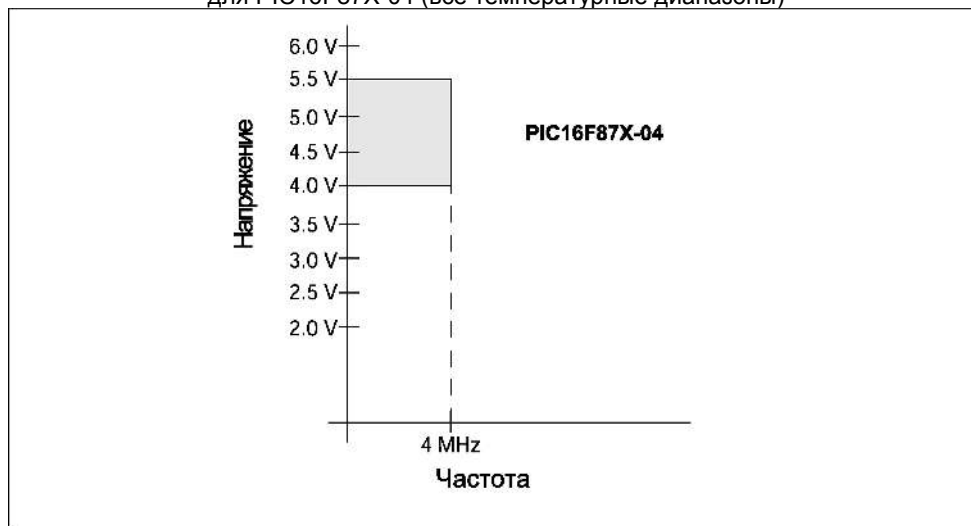
**Рис. 15-2** График рекомендованных комбинаций значений напряжения питания и тактовой частоты для PIC16LF87X-04 (коммерческий и промышленный температурный диапазон)



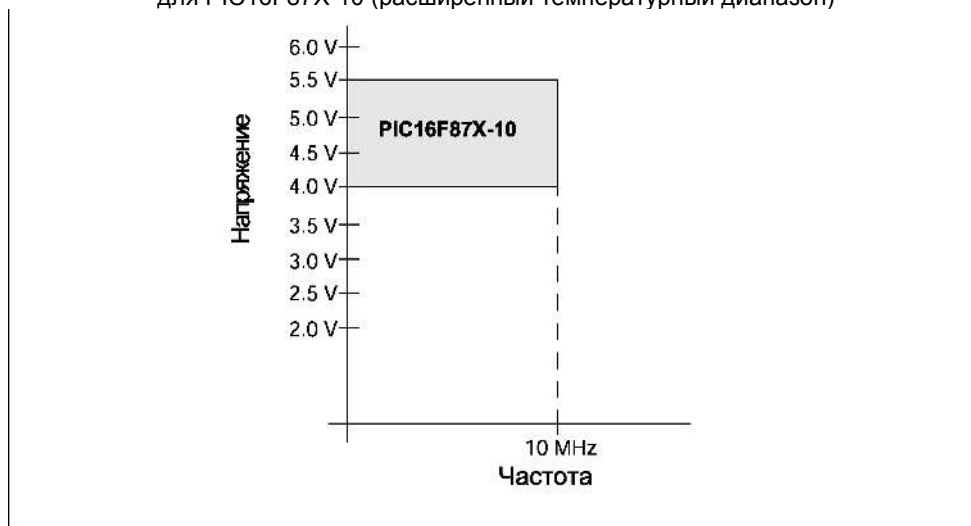
**Примечания:**

1.  $V_{DDAPP\text{MIN}}$  - минимальное напряжение питания микроконтроллера в устройстве.
2.  $F_{MAX}$  - максимальная частота 10 МГц.

**Рис. 15-3** График рекомендованных комбинаций значений напряжения питания и тактовой частоты для PIC16F87X-04 (все температурные диапазоны)



**Рис. 15-4** График рекомендованных комбинаций значений напряжения питания и тактовой частоты для PIC16F87X-10 (расширенный температурный диапазон)



**15.1 Электрические характеристики (Коммерческий, Промышленный)****PIC16F873/874/876/877-04 (Коммерческий, Промышленный)****PIC16F873/874/876/877-20 (Коммерческий, Промышленный)****PIC16LF873/874/876/877-04 (Коммерческий, Промышленный)**

PIC16LF873/874/876/877-04 (Коммерческий, Промышленный)		Стандартные рабочие условия (если не указано иное) Температурный диапазон: Коммерческий 0°C ≤ T <sub>A</sub> ≤ +70°C Промышленный -40°C ≤ T <sub>A</sub> ≤ +85°C						
PIC16F873/874/876/877-04 PIC16F873/874/876/877-20 (Коммерческий, Промышленный)		Стандартные рабочие условия (если не указано иное) Температурный диапазон: Коммерческий 0°C ≤ T <sub>A</sub> ≤ +70°C Промышленный -40°C ≤ T <sub>A</sub> ≤ +85°C						
№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание	
D001	V <sub>DD</sub>	Напряжение питания						
		16LF87X	2.0	-	5.5	B	LP, XT, RC режим генератора (от DC до 4МГц)	
D001 D001A		16F87X	4.0 4.5 V <sub>BOR</sub>	-	5.5 5.5 5.5	B B B	LP, XT, RC режим генератора HS режим генератора BOR включен, F <sub>MAX</sub> =4МГц <sup>(7)</sup>	
D002	V <sub>DR</sub>	Напряжение сохранения данных в ОЗУ <sup>(1)</sup>	-	1.5	-	B		
D003	V <sub>POR</sub>	Стартовое напряжение V <sub>DD</sub> для формирования POR	-	V <sub>SS</sub>	-	B	Смотрите раздел "сброс POR"	
D004	S <sub>VDD</sub>	Скорость нарастания V <sub>DD</sub> для формирования POR	0.05	-	-	B/мс	Смотрите раздел "сброс POR"	
D005	V <sub>BOD</sub>	Напряжение детектора BOD	3.7	4.0	4.35	B	Бит BODEN = 0	
	I <sub>DD</sub>	Ток потребления <sup>(2,5)</sup>						
D010		16LF87X	-	0.6	2.0	мА	XT, RC режим генератора F <sub>OSC</sub> = 4МГц, V <sub>DD</sub> =3.0В	
D010		16F87X	-	1.6	4	мА	RC режим генератора F <sub>OSC</sub> = 4МГц, V <sub>DD</sub> =5.5В	
D010A		16LF87X	-	20	35	мкА	LP режим генератора F <sub>OSC</sub> = 32кГц, V <sub>DD</sub> =3.0В, WDT выключен	
D013		16F87X	-	7	15	мА	HS режим генератора F <sub>OSC</sub> = 20МГц, V <sub>DD</sub> =5.5В	
D015	ΔI <sub>BOR</sub>	Ток потребления BOR <sup>(6)</sup>	-	85	200	мкА	Бит BODEN = 0, V <sub>DD</sub> =5В	
	I <sub>PD</sub>	Ток потребления в SLEEP режиме <sup>(3,5)</sup>						
D020		16LF87X	-	7.5	30	мкА	V <sub>DD</sub> =3.0В, WDT включен, от -40°C до +85°C	
D020		16F87X	-	10.5	42	мкА	V <sub>DD</sub> =4.0В, WDT включен, от -40°C до +85°C	
D021		16LF87X	-	0.9	5	мкА	V <sub>DD</sub> =3.0В, WDT включен, от 0°C до +70°C	
D021		16F87X	-	1.5	16	мкА	V <sub>DD</sub> =4.0В, WDT включен, от -40°C до +85°C	
D021A		16LF87X	-	0.9	5	мкА	V <sub>DD</sub> =3.0В, WDT включен, от -40°C до +85°C	
D021A		16F87X	-	1.5	19	мкА	V <sub>DD</sub> =4.0В, WDT включен, от -40°C до +85°C	
D023	ΔI <sub>BOR</sub>	Ток потребления BOR <sup>(6)</sup>	-	85	200	мкА	Бит BODEN = 0, V <sub>DD</sub> =5В	

\*\* - В столбце "Тип." приведены параметры при V<sub>DD</sub>=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<sub>DD</sub> 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<sub>DD</sub> проводилось в следующих условиях: внешний тактовый сигнал (меандр); каналы портов ввода/вывода в третьем состоянии и подтянуты к V<sub>DD</sub>; -MCLR = V<sub>DD</sub>; WDT выключен/выключен, указано в спецификации.
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<sub>DD</sub> или V<sub>SS</sub>.
4. В RC режиме генератора ток через внешний резистор не учитывается. Ток, протекающий через внешний резистор, может быть рассчитан по формуле I<sub>r</sub> = V<sub>DD</sub>/2R<sub>EXT</sub> (мА), где R<sub>EXT</sub> в кОм.
5. Генератор TMR1 дополнительно потребляет 20мкА (если включен). Этот параметр используется при разработке устройств, но не тестируется.
6. Δ ток - дополнительный потребляемый ток, если периферийный модуль включен. Этот ток должен быть добавлен к I<sub>DD</sub> или I<sub>PD</sub>.
7. Когда BOR включен, микроконтроллер будет нормально работать, пока напряжение питания не опустится ниже V<sub>BOR</sub>.

**15.2 Электрические характеристики (Коммерческий, Промышленный)****PIC16F873/874/876/877-04 (Коммерческий, Промышленный)****PIC16F873/874/876/877-20 (Коммерческий, Промышленный)****PIC16LF873/874/876/877-04 (Коммерческий, Промышленный)**

Рабочее напряжение питания $V_{DD}$ должно соответствовать значению, указанному в разделе 15.1		Стандартные рабочие условия (если не указано иное)					
		Температурный диапазон: Коммерческий $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ Промышленный $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$					
№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
	$V_{IL}$	Входное напряжение низкого уровня					
D030 D030A		Канал порта ввода/вывода ТТЛ буфер	$V_{SS}$	-	0.8	В	$V_{DD}$ = от 4.5В до 5.5В иначе
D031		Триггер Шмидта	$V_{SS}$	-	$0.15V_{DD}$	В	
D032		-MCLR, OSC1 (RC) <sup>(1)</sup>	$V_{SS}$	-	$0.2V_{DD}$	В	
D033		OSC1 (XT, HS, LP)	$V_{SS}$	-	$0.2V_{DD}$	В	
D034 D034A		Выводы RC3, RC4 Триггер Шмидта SMBus	$V_{SS}$ -0.5	-	$0.3V_{DD}$ 0.6	В В	
	$V_{IH}$	Входное напряжение высокого уровня					
D040 D040A		Канал порта ввода/вывода ТТЛ буфер	2.0 $0.25V_{DD}+0.8$	-	$V_{DD}$ $V_{DD}$	В В	$V_{DD}$ = от 4.5В до 5.5В иначе
D041		Триггер Шмидта	$0.8V_{DD}$	-	$V_{DD}$	В	
D042		-MCLR	$0.8V_{DD}$	-	$V_{DD}$	В	
D042A		OSC1 (XT, HS, LP)	$0.7V_{DD}$	-	$V_{DD}$	В	
D043		OSC1 (RC) <sup>(1)</sup>	$0.9V_{DD}$	-	$V_{DD}$	В	
D044 D044A		Выводы RC3, RC4 Триггер Шмидта SMBus	$0.7V_{DD}$ 1.4	-	$V_{DD}$ 5.5	В В	
D070	$I_{PURB}$	Ток через подтягивающие резисторы PORTB	50	250	400	мкА	$V_{DD} = 5.0\text{В}$ , $V_{PIN} = V_{SS}$ от $-40^{\circ}\text{C}$ до $+85^{\circ}\text{C}$
	$I_{IL}$	Входной ток утечки <sup>(2,3)</sup>					
D060		Порт ввода/вывода	-	-	$\pm 1$	мкА	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , 3-е сост. $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS, LP
D061		-MCLR, RA4/T0CKI	-	-	$\pm 5$	мкА	
D063		OSC1	-	-	$\pm 5$	мкА	
	$V_{OL}$	Выходное напряжение низкого уровня $V_{DD} = 4.5\text{В}$					
D080 D083		Канал ввода/вывода OSC2/CLKOUT (RC)	-	-	0.6 0.6	В В	$I_{OL} = 8.5\text{ мА}$ , $-40^{\circ}\text{C}$ до $+85^{\circ}\text{C}$ $I_{OL} = 1.6\text{ мА}$ , $-40^{\circ}\text{C}$ до $+85^{\circ}\text{C}$
	$V_{OH}$	Выходное напряжение высокого уровня $V_{DD} = 4.5\text{В}$					
D090 D092		Канал ввода/вывода <sup>(3)</sup> OSC2/CLKOUT (RC)	$V_{DD} - 0.7$ $V_{DD} - 0.7$	-	-	В В	$I_{OH} = -3.0\text{ мА}$ , $-40^{\circ}\text{C}$ до $+85^{\circ}\text{C}$ $I_{OH} = -1.3\text{ мА}$ , $-40^{\circ}\text{C}$ до $+85^{\circ}\text{C}$
D150*	$V_{OD}$	Напряжение на выходе с открытым стоком	-	-	8.5	В	RA4
		Емкостная нагрузка на выходах					
D100 D101	$C_{OSC2}$ $C_{IO}$	Вывод OSC2 Все каналы ввода/вывода и OSC2 в RC режиме	-	-	15 50	пФ пФ	XT, HS, LP
D102	$C_B$	SCL, SDA в режиме $I^2C$	-	-	400	пФ	
		EEPROM память данных					
D120 D121	$E_D$ $V_{DRW}$	Число циклов стирание/запись Напряжение питания для записи/чтения	100K $V_{MIN}$	-	- 5.5	С/3 В	5В @ $25^{\circ}\text{C}$ $V_{MIN}$ - минимальное напряжение питания
D122	$T_{DEW}$	Время цикла стирание/запись	-	4	8	мс	
		FLASH память программ					
D130 D131 D132A	$E_P$ $V_{PR}$ $V_{PEW}$	Число циклов стирание/запись Напряжение питания для чтения Напряжение питания для стирания/записи	1000 $V_{MIN}$ $V_{MIN}$	-	- 5.5 5.5	С/3	5В @ $25^{\circ}\text{C}$ $V_{MIN}$ - минимальное напряжение питания
D133	$T_{PEW}$	Время цикла стирание/запись	-	4	8		

\* - Эти параметры определены, но не протестированы.

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

- В RC режиме генератора на входе OSC1 включен триггер Шмидта. Не рекомендуется использовать внешний тактовый сигнал для PIC16F87X в RC режиме тактового генератора.
- Ток утечки на выводе -MCLR зависит от приложенного напряжения. Параметры указаны для нормального режима работы. В других режимах может возникнуть больший ток утечки.
- Отрицательный ток показывает, что он вытекает из вывода.

**15.3 Электрические характеристики (Расширенный)****PIC16F873/874/876/877-04 (Расширенный)****PIC16F873/874/876/877-10 (Расширенный)**

PIC16F873/874/876/877-04 PIC16F873/874/876/877-10 (Расширенный)		Стандартные рабочие условия (если не указано иное) Температурный диапазон: Расширенный $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$					
№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
D001 D001A	V <sub>DD</sub>	Напряжение питания	4.0 4.5 V <sub>BOR</sub>	-	5.5 5.5 5.5	B B B	LP, XT, RC режим генератора HS режим генератора BOR включен, F <sub>MAX</sub> =4МГц <sup>(7)</sup>
D002	V <sub>DR</sub>	Напряжение сохранения данных в ОЗУ <sup>(1)</sup>	-	1.5	-	B	
D003	V <sub>POR</sub>	Стартовое напряжение V <sub>DD</sub> для формирования POR	-	V <sub>SS</sub>	-	B	Смотрите раздел "сброс POR"
D004	S <sub>VDD</sub>	Скорость нарастания V <sub>DD</sub> для формирования POR	0.05	-	-	B/мс	Смотрите раздел "сброс POR"
D005	V <sub>BOD</sub>	Напряжение детектора BOD	3.7	4.0	4.35	B	Бит BODEN = 0
D010 D013	I <sub>DD</sub>	Ток потребления <sup>(2,5)</sup>	-	1.6 7	4 15	мА мА	RC режим генератора F <sub>OSC</sub> = 4МГц, V <sub>DD</sub> =5.5В HS режим генератора F <sub>OSC</sub> = 10МГц, V <sub>DD</sub> =5.5В
D015	ΔI <sub>BOR</sub>	Ток потребления BOR <sup>(6)</sup>	-	85	200	мкА	Бит BODEN = 0, V <sub>DD</sub> =5В
D020A D021B	I <sub>PD</sub>	Ток потребления в SLEEP режиме <sup>(3,5)</sup>	-	10.5 1.5	60 30	мкА мкА	V <sub>DD</sub> =4.0В, WDT включен V <sub>DD</sub> =4.0В, WDT выключен
D023	ΔI <sub>BOR</sub>	Ток потребления BOR <sup>(6)</sup>	-	85	200	мкА	Бит BODEN = 0, V <sub>DD</sub> =5В

\*\* - В столбце "Тип." приведены параметры при V<sub>DD</sub>=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<sub>DD</sub> 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<sub>DD</sub> проводилось в следующих условиях: внешний тактовый сигнал (меандр); каналы портов ввода/вывода в третьем состоянии и подтянуты к V<sub>DD</sub>; -MCLR = V<sub>DD</sub>; WDT выключен/выключен, указано в спецификации.
- 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<sub>DD</sub> или V<sub>SS</sub>.
- В RC 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<sub>EXT</sub> в кОм.
- Генератор TMR1 дополнительно потребляет 20мкА (если включен). Этот параметр используется при разработке устройств, но не тестируется.
- Δ ток - дополнительный потребляемый ток, если периферийный модуль включен. Этот ток должен быть добавлен к I<sub>DD</sub> или I<sub>PD</sub>.
- Когда BOR включен, микроконтроллер будет нормально работать, пока напряжение питания не опустится ниже V<sub>BOR</sub>.

**15.4 Электрические характеристики (Расширенный)****PIC16F873/874/876/877-04 (Расширенный)  
PIC16F873/874/876/877-10 (Расширенный)**

Рабочее напряжение питания $V_{DD}$ должно соответствовать значению, указанному в разделе 15.3		Стандартные рабочие условия (если не указано иное) Температурный диапазон: Расширенный $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$					
№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
	$V_{IL}$	Входное напряжение низкого уровня					
D030		Канал порта ввода/вывода	$V_{SS}$	-	0.8	B	$V_{DD}$ = от 4.5В до 5.5В иначе
D030A		ТТЛ буфер	$V_{SS}$	-	$0.15V_{DD}$	B	
D031		Триггер Шмидта	$V_{SS}$	-	$0.2V_{DD}$	B	
D032		-MCLR, OSC1 (RC) <sup>(1)</sup>	$V_{SS}$	-	$0.2V_{DD}$	B	
D033		OSC1 (XT, HS, LP)	$V_{SS}$	-	$0.3V_{DD}$	B	
D034		Выводы RC3, RC4	$V_{SS}$	-	$0.3V_{DD}$	B	
D034A		Триггер Шмидта SMBus	-0.5	-	0.6	B	
	$V_{IH}$	Входное напряжение высокого уровня					
D040		Канал порта ввода/вывода	2.0	-	$V_{DD}$	B	$V_{DD}$ = от 4.5В до 5.5В иначе
D040A		ТТЛ буфер	$0.25V_{DD}+0.8$	-	$V_{DD}$	B	
D041		Триггер Шмидта	$0.8V_{DD}$	-	$V_{DD}$	B	
D042		-MCLR	$0.8V_{DD}$	-	$V_{DD}$	B	
D042A		OSC1 (XT, HS, LP)	$0.7V_{DD}$	-	$V_{DD}$	B	
D043		OSC1 (RC) <sup>(1)</sup>	$0.9V_{DD}$	-	$V_{DD}$	B	
D044		Выводы RC3, RC4	$0.7V_{DD}$	-	$V_{DD}$	B	
D044A		Триггер Шмидта SMBus	1.4	-	5.5	B	
D070	$I_{PURB}$	Ток через подтягивающие резисторы PORTB	50	250	400	мкА	$V_{DD} = 5.0\text{В}, V_{PIN} = V_{SS}$
	$I_{IL}$	Входной ток утечки <sup>(2,3)</sup>					
D060		Порт ввода/вывода	-	-	$\pm 1$	мкА	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , 3-е сост. $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS, LP
D061		-MCLR, RA4/T0CKI	-	-	$\pm 5.0$	мкА	
D063		OSC1	-	-	$\pm 5.0$	мкА	
	$V_{OL}$	Выходное напряжение низкого уровня $V_{DD} = 4.5\text{В}$					
D080A		Канал ввода/вывода	-	-	0.6	B	$I_{OL} = 7.0\text{ мА}$
D083A		OSC2/CLKOUT (RC)	-	-	0.6	B	$I_{OL} = 1.2\text{ мА}$
	$V_{OH}$	Выходное напряжение высокого уровня $V_{DD} = 4.5\text{В}$					
D090A		Канал ввода/вывода <sup>(3)</sup>	$V_{DD} - 0.7$	-	-	B	$I_{OH} = -2.5\text{ мА}$
D092A		OSC2/CLKOUT (RC)	$V_{DD} - 0.7$	-	-	B	$I_{OH} = -1.0\text{ мА}$
D150*	$V_{OD}$	Напряжение на выходе с открытым стоком	-	-	8.5	B	RA4
		Емкостная нагрузка на выходах					
D100	$C_{OSC2}$	Вывод OSC2	-	-	15	пФ	XT, HS, LP
D101	$C_{IO}$	Все каналы ввода/вывода и OSC2 в RC режиме	-	-	50	пФ	
D102	$C_B$	SCL, SDA в режиме $I^2C$	-	-	400	пФ	
		EEPROM память данных					
D120	$E_D$	Число циклов стирание/запись	100K	-	-	C/3	5В @ 25°C
D121	$V_{DRW}$	Напряжение питания для записи/чтения	$V_{MIN}$	-	5.5	B	$V_{MIN}$ - минимальное напряжение питания
D122	$T_{DEW}$	Время цикла стирание/запись	-	4	8	мс	
		FLASH память программ					
D130	$E_P$	Число циклов стирание/запись	1000	-	-	C/3	5В @ 25°C
D131	$V_{PR}$	Напряжение питания для чтения	$V_{MIN}$	-	5.5	B	$V_{MIN}$ - минимальное напряжение питания
D132A	$V_{PEW}$	Напряжение питания для стирания/записи	$V_{MIN}$	-	5.5	B	$V_{MIN}$ - минимальное напряжение питания
D133	$T_{PEW}$	Время цикла стирание/запись	-	4	8	мс	

\* - Эти параметры определены, но не протестированы.

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

- В RC режиме генератора на входе OSC1 включен триггер Шмидта. Не рекомендуется использовать внешний тактовый сигнал для PIC16F87X в RC режиме тактового генератора.
- Ток утечки на выводе -MCLR зависит от приложенного напряжения. Параметры указаны для нормального режима работы. В других режимах может возникнуть больший ток утечки.
- Отрицательный ток показывает, что он вытекает из вывода.

## 15.5 Символьное обозначение временных параметров

Символьное обозначение временных параметров имеет один из следующих форматов:

- |                  |                |  |
|------------------|----------------|--|
| 1. $T_{ppS2ppS}$ | 3. $T_{CC:ST}$ | (только спецификация I <sup>2</sup> C) |
| 2. $T_{ppS}$     | 4. $T_S$       | (только спецификация I <sup>2</sup> C) |

<b>T</b>			
F	Частота	T	Время

Строчные символы (pp) и их значение

<b>pp</b>			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	-RD
cs	-CS	rw	-RD или -WR
di	SDI	sc	SCK
do	SDO	ss	-SS
dt	Входные данные	t0	T0CKI
io	Канал ввода/вывода	t1	T1CKI
mc	-MCLR	wr	-WR

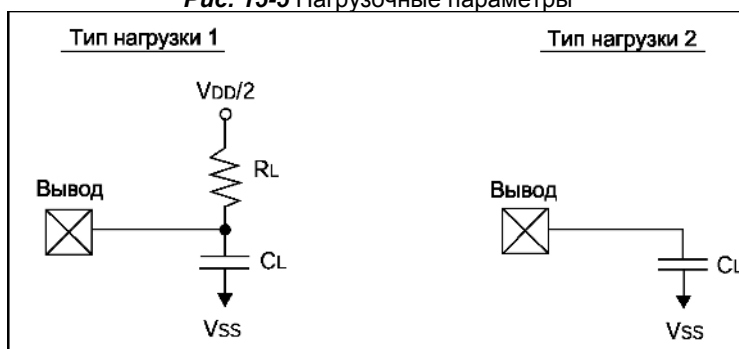
Прописные символы и их значение

<b>S</b>			
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
<b>Только I<sup>2</sup>C</b>			
AA	Доступ вывода	High	Высокий уровень
BUF	Шина свободна	Low	Низкий уровень

$T_{CC:ST}$  (только спецификация I<sup>2</sup>C)

<b>CC</b>			
HD	Удержание	SU	Установка
<b>ST</b>			
DAT	Сохранение данных на входе	STO	Условие STOP
STA	Условие START	Low	Низкий уровень

Рис. 15-5 Нагрузочные параметры



$R_L = 464\text{Ом}$

$C_L = 50\text{пФ}$  (для всех выводов, кроме OSC2, включая PORTD и PORTE в режиме портов ввода/вывода)

$C_L = 15\text{пФ}$  (для вывода OSC2)

**Примечание.** PORTD, PORTE не реализованы в микроконтроллерах PIC16F873/876.

## 15.6 Временные диаграммы и спецификации

Рис. 15-6 Временная диаграмма внешнего тактового сигнала

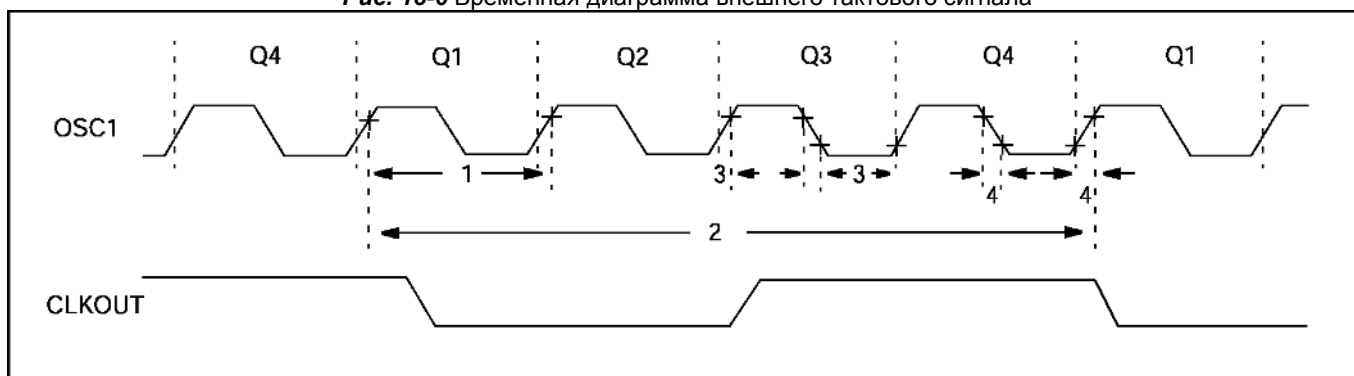


Таблица 15-1 Параметры внешнего тактового сигнала

№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
	F <sub>osc</sub>	Частота внешнего тактового сигнала <sup>(1)</sup>	DC	-	4	МГц	XT, RC
			DC	-	4	МГц	HS режим (-04)
			DC	-	10	МГц	HS режим (-10)
			DC	-	20	МГц	HS режим (-20)
			Dc	-	200	кГц	LP режим
		Частота генератора <sup>(1)</sup>	DC	-	4	МГц	RC режим
			0.1	-	4	МГц	XT режим
			4	-	10	МГц	HS режим (-10)
			4	-	20	МГц	HS режим (-20)
			5	-	200	кГц	LP режим
1	T <sub>osc</sub>	Период внешнего тактового сигнала <sup>(1)</sup>	250	-	-	нс	XT, RC режим
			250	-	-	нс	HS режим (-04)
			100	-	-	нс	HS режим (-10)
			50	-	-	нс	HS режим (-20)
			5	-	-	мкс	LP режим
		Период генератора <sup>(1)</sup>	250	-	-	нс	RC режим
			250	-	10000	нс	XT режим
			250	-	-	нс	HS режим (-04)
			100	-	250	нс	HS режим (-10)
			50	-	250	нс	HS режим (-20)
5	-	-	мкс	LP режим			
2	T <sub>cy</sub>	Время выполнения инструкции <sup>(1)</sup>	200.0	T <sub>cy</sub>	DC	нс	T <sub>cy</sub> = 4/F <sub>osc</sub>
3	T <sub>osL</sub> , T <sub>osH</sub>	Длительность высокого/низкого уровня CLKIN (OSC1)	100	-	-	нс	XT режим
			2.5	-	-	мкс	LP режим
			15	-	-	нс	HS режим
4	T <sub>osR</sub> , T <sub>osF</sub>	Длительность переднего/заднего фронта внешнего тактового сигнала (OSC1)	-	-	25	нс	XT режим
			-	-	50	нс	LP режим
			-	-	15	нс	HS режим

\*\* - В столбце "Тип." приведены параметры при V<sub>DD</sub>=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 большему потребляемому току. Все микроконтроллеры проверены в режиме "Мин." при внешнем тактовом сигнале на выводе OSC1/CLKIN.

Рис. 15-7 Временная диаграмма CLKOUT и каналов ввода/вывода

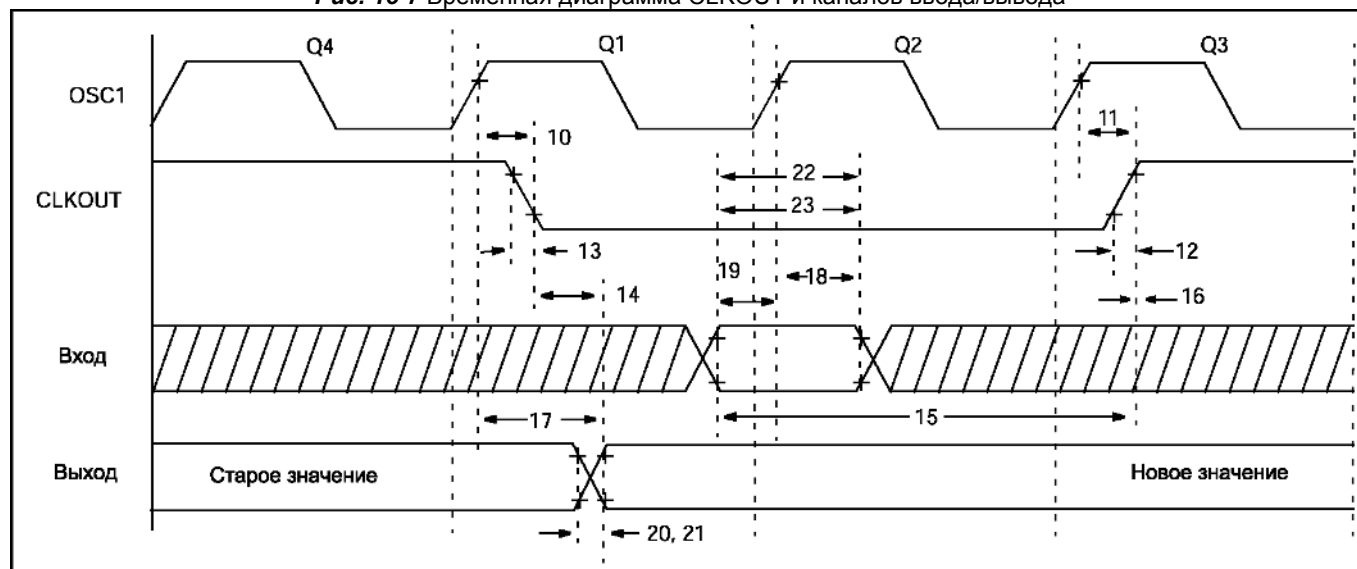


Таблица 15-2 Параметры CLKOUT и каналов ввода/вывода

№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
10*	TosH2ckL	От OSC1 ↑ до CLKOUT ↓	-	75	200	нс	(1)
11*	TosH2ckH	От OSC1 ↑ до CLKOUT ↑	-	75	200	нс	(1)
12*	TckR	CLKOUT длит. переднего фронта	-	35	100	нс	(1)
13*	TckF	CLKOUT длит. заднего фронта	-	35	100	нс	(1)
14*	TckL2ioV	От CLKOUT ↓ до установл. выхода	-	-	0.5T <sub>cy</sub> +20	нс	(1)
15*	TioV2ckH	От установл. входа до CLKOUT ↑	T <sub>osc</sub> +200	-	-	нс	(1)
16*	TckH2ioI	Удержание входа после CLKOUT ↑	0	-	-	нс	(1)
17*	TosH2ioV	От OSC1 ↑ до установл. выхода	-	100	255	нс	
18*	TosH2ioI	Удержание входа после OSC1 ↑	F	100	-	-	нс
		LF	200	-	-	нс	
19*	TioV2osH	Переход в режим входа относ. OSC1↑	0	-	-	нс	
20*	TioR	Длительность переднего фронта на выходе порта ввода/вывода	F	-	10	40	нс
		LF	-	-	145	нс	
21*	TioF	Длительность заднего фронта на выходе порта ввода/вывода	F	-	10	40	нс
		LF	-	-	145	нс	
22***	Tinp	Длит. высокого/низкого уровня INT	T <sub>cy</sub>	-	-	нс	
23***	Trbp	Длит. высокого/низкого уровня RB7:RB4	T <sub>cy</sub>	-	-	нс	

\* - Эти параметры определены, но не протестированы.

\*\* - В столбце "Тип." приведены параметры при V<sub>DD</sub>=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

\*\*\* - Асинхронные события, не связанные с внутренним тактовым сигналом.

**Примечание 1.** Измерения проведены в RC режиме генератора, где CLKOUT = 4 x T<sub>osc</sub>.



Рис. 15-8 Временная диаграмма сброса, WDT, OST, PWRT

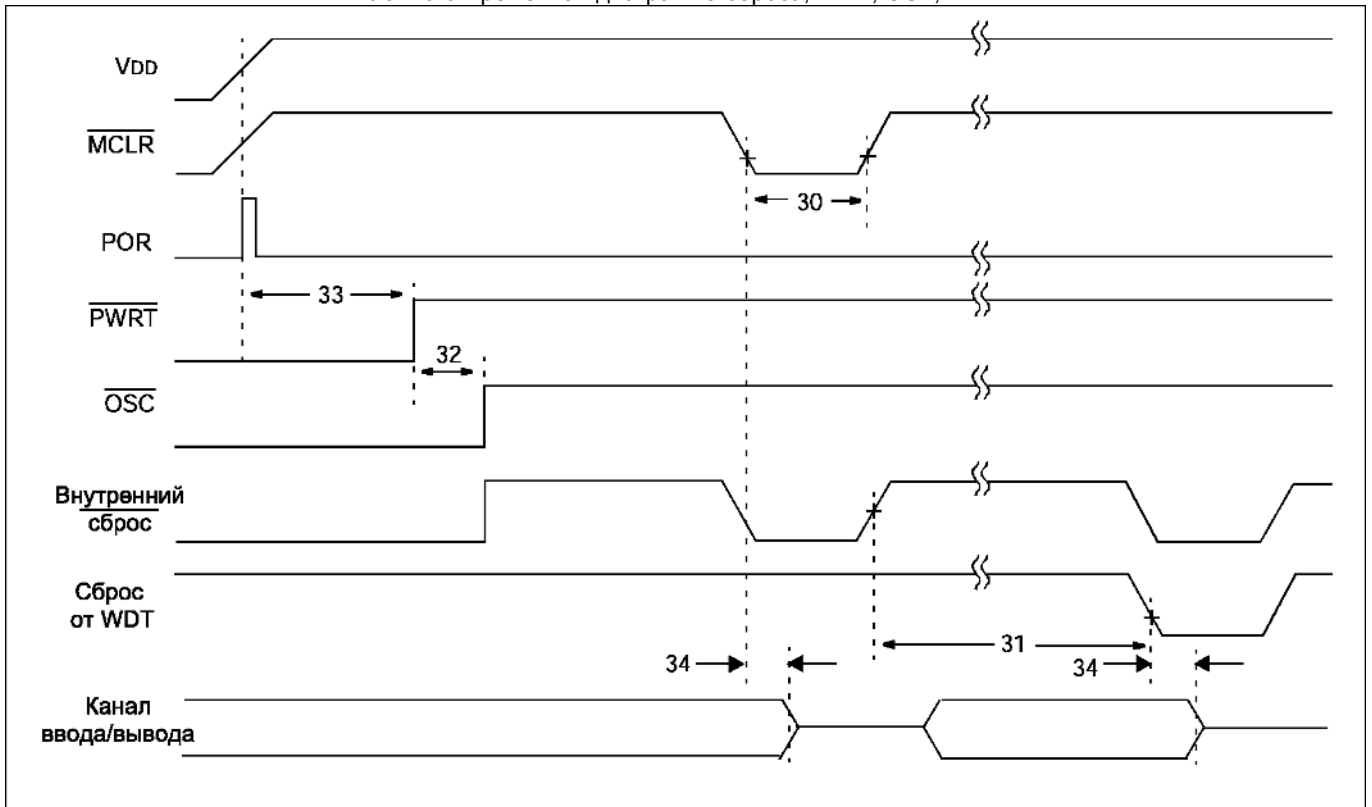


Рис. 15-9 Временная диаграмма работы BOR

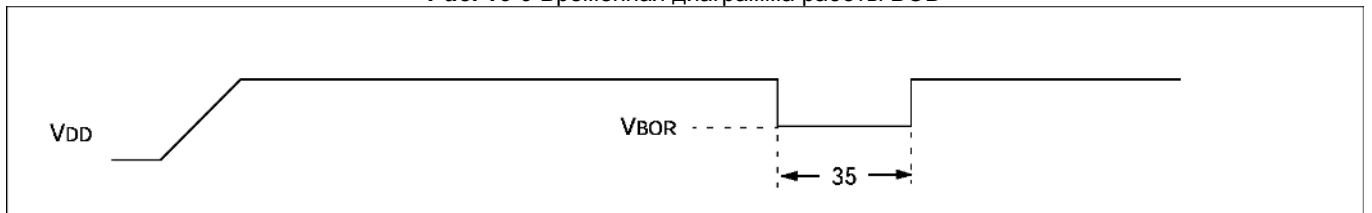


Таблица 15-3 Параметры сброса, WDT, OST, PWRT, BOR

№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
30	Tmcl	Длительность импульса -MCLR	2	-	-	мкс	V <sub>DD</sub> =5В, -40°C до +85°C
31*	Twdt	Период переполнения WDT (без предделителя)	7	18	33	мс	V <sub>DD</sub> =5В, -40°C до +85°C
32	Tost	Период OST	-	1024T <sub>OSC</sub>	-	-	T <sub>OSC</sub> = период OSC1
33*	Trprt	Период PWRT	28	72	132	мс	V <sub>DD</sub> =5В, -40°C до +85°C
34	T <sub>IOZ</sub>	От сброса -MCLR или WDT до перевода каналов ввода/вывода 3-е состояние	-	-	2.1	мкс	
35	T <sub>BOR</sub>	Длительность импульса BOR	100	-	-	мкс	V <sub>DD</sub> ≤ V <sub>BOR</sub> (D005)

\* - Эти параметры определены, но не протестированы.

\*\* - В столбце "Тип." приведены параметры при V<sub>DD</sub>=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

Рис. 15-10 Временная диаграмма внешнего тактового сигнала для TMR0 и TMR1

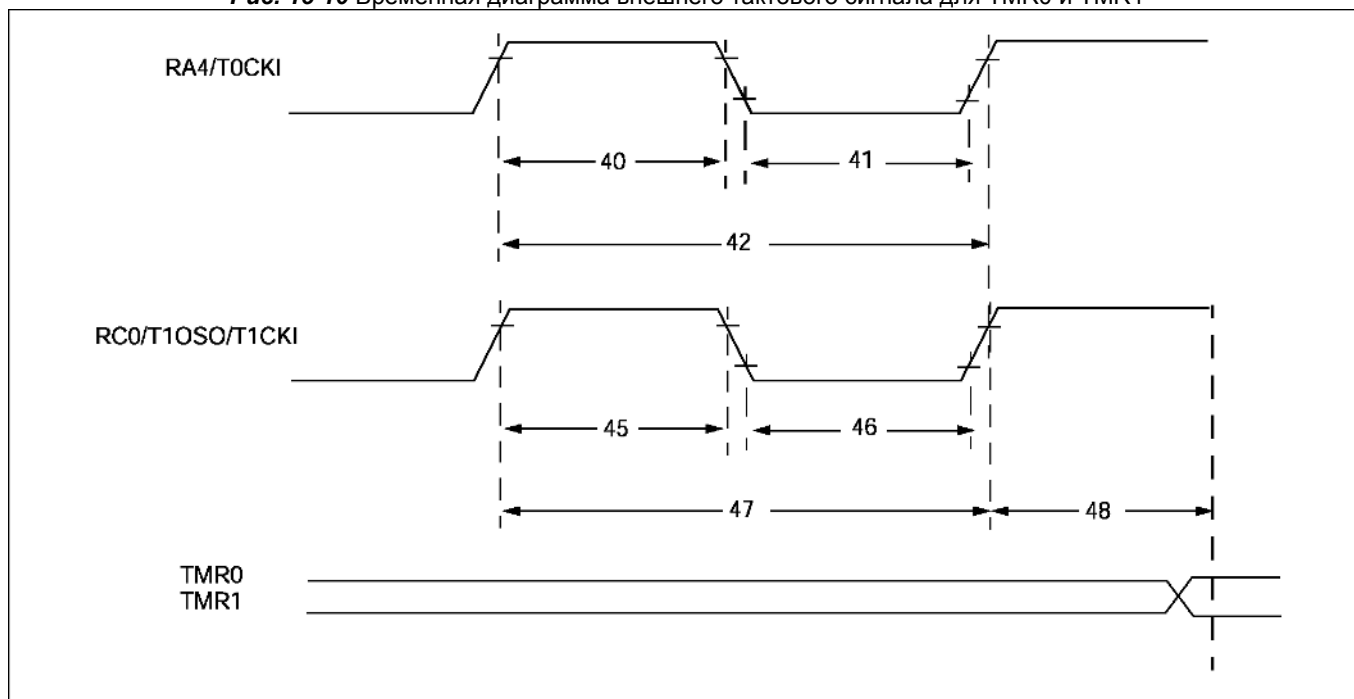


Таблица 15-4 Параметры внешнего тактового сигнала для TMR0 и TMR1

№ пар.	Обоз.	Описание		Мин.	Тип.**	Макс.	Ед.	Примечание	
40*	Tt0H	Длительность высокого уровня T0CKI	Без делителя	$0.5T_{CY}+20$	-	-	нс	Также должен выполняться параметр 42	
41*	Tt0L		С делителем	10	-	-	нс		
42*	Tt0P	Период T0CKI	Без делителя	$T_{CY}+40$	-	-	нс		
			С делителем	20 или $(T_{CY}+40)/N$	-	-	нс		N = коэфф.предд.
45*	Tt1H	Длительность высокого уровня T1CKI	Синхр.реж. без делит.	$0.5T_{CY}+20$	-	-	нс	Также должен выполняться параметр 47	
			Синхр. режим с делит.	F	15	-	-		нс
				LF	25	-	-		нс
			Асинхронный режим	F	30	-	-		нс
LF	50	-		-	нс				
46*	Tt1L	Длительность низкого уровня T1CKI	Синхр.реж. без делит.	$0.5T_{CY}+20$	-	-	нс	Также должен выполняться параметр 47	
			Синхр. режим с делит.	F	15	-	-		нс
				LF	25	-	-		нс
			Асинхронный режим	F	30	-	-		нс
LF	50	-		-	нс				
47*	Tt1P	Период T1CKI	Синхронный режим	F	30 или $(T_{CY}+40)/N$	-	-	нс	N = коэфф.предд.
				LF	50 или $(T_{CY}+40)/N$	-	-	нс	N = коэфф.предд.
			Асинхронный режим	F	60	-	-	нс	
				LF	100	-	-	нс	
	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. 15-11 Временная диаграмма захват/сравнение/ШИМ

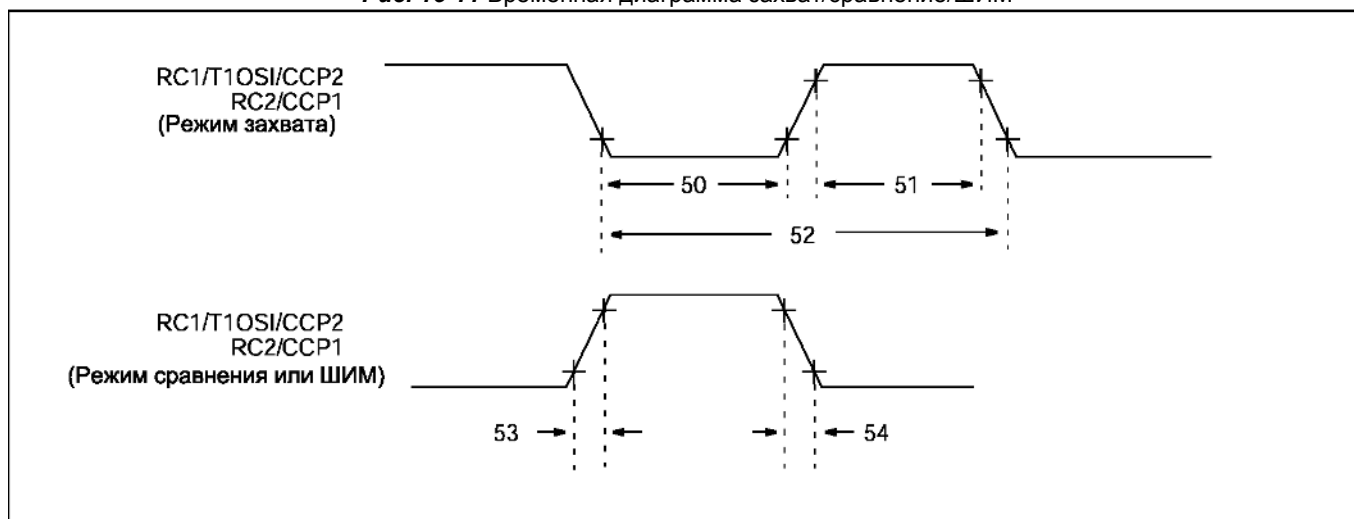


Таблица 15-5 Параметры захват/сравнение/ШИМ (CCP1 и CCP2)

№ пар.	Обоз.	Описание		Мин.	Тип.**	Макс.	Ед.	Примечание	
50*	ТссL	Сигнал низкого уровня CCP1 и CCP2	Без делителя	$0.5T_{CY}+20$	-	-	нс		
			С делителем	F	10	-	-		нс
				LF	20	-	-		нс
51*	ТссL	Сигнал высокого уровня CCP1 и CCP2	Без делителя	$0.5T_{CY}+20$	-	-	нс		
			С делителем	F	10	-	-		нс
				LF	20	-	-		нс
52*	ТссP	Период входного сигнала CCP		$(3T_{CY}+40)/N$	-	-	нс	N = коэфф.предд.	
53*	ТссR	Время установление высокого уровня сигн. на вых. CCP1 и CCP2	F	-	10	25	нс		
			LF	-	25	45	нс		
54*	ТссF	Время установление низкого уровня сигн. на вых. CCP1 и CCP2	F	-	10	25	нс		
			LF	-	25	45	нс		

\* - 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. 15-12 Временная диаграмма работы ведомого параллельного порта (только PIC16F874/877)

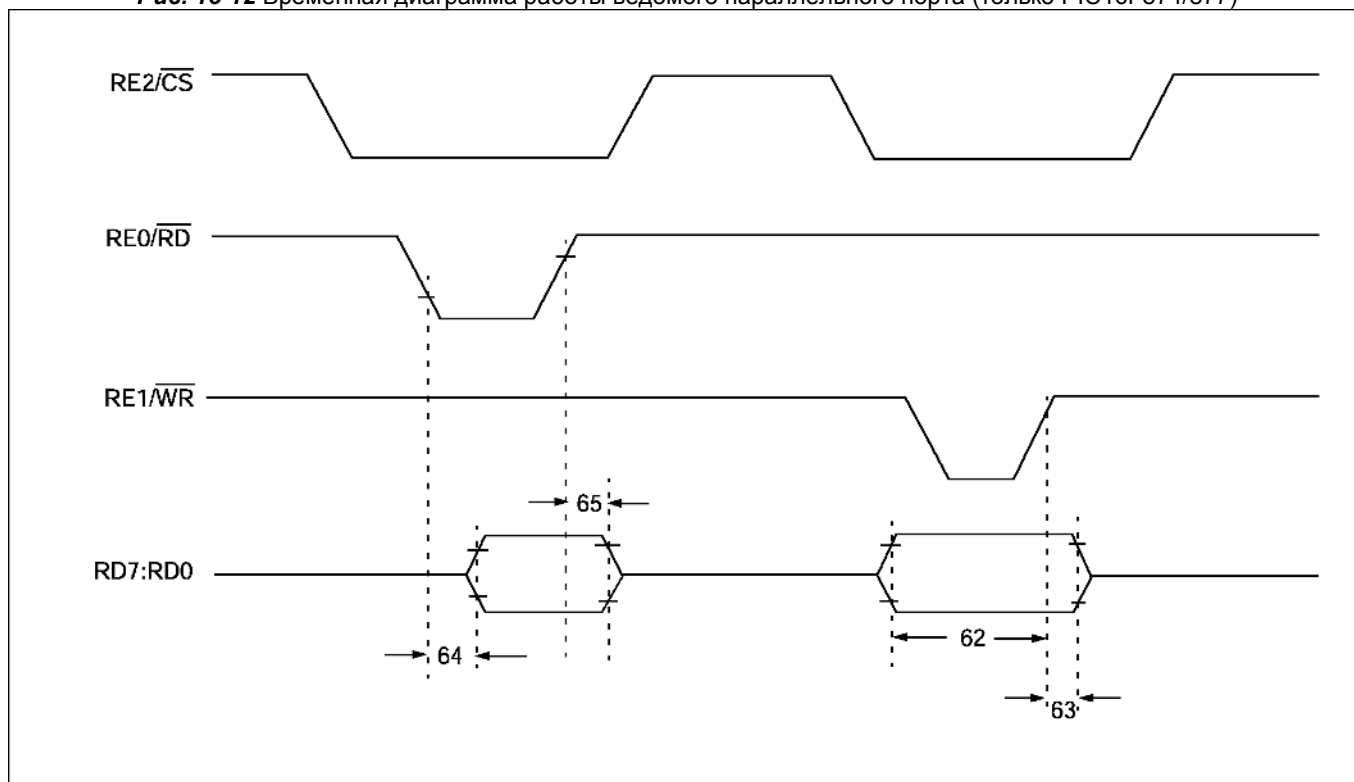


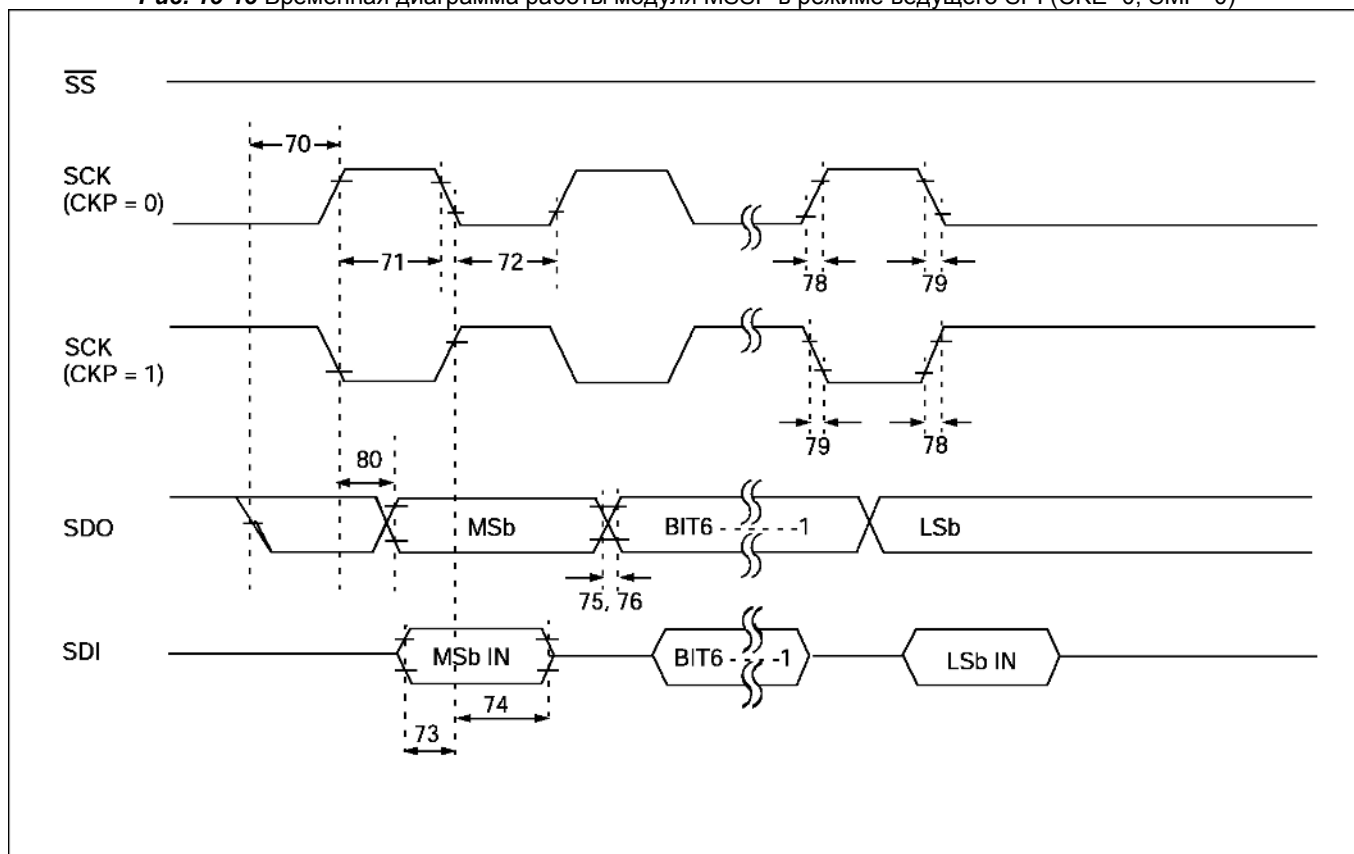
Таблица 15-6 Параметры работы ведомого параллельного порта (только PIC16F874/877)

№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
62	TdtV2H	Установка данных перед -WR↑ или -CS↑	20 25	- -	- -	нс нс	Только для расшир. диап.
63*	TwrH2dtl	Удержание данных после -WR↑ или -CS↑	F: 20 LF: 35	- -	- -	нс нс	
64	TrdL2dtV	Формирование данных после -RD↓ и -CS↓	- -	- -	80 90	нс нс	Только для расшир. диап.
65	TrdH2dtl	Неправильные данные после -RD↑ или -CS↑	10	-	30	нс	

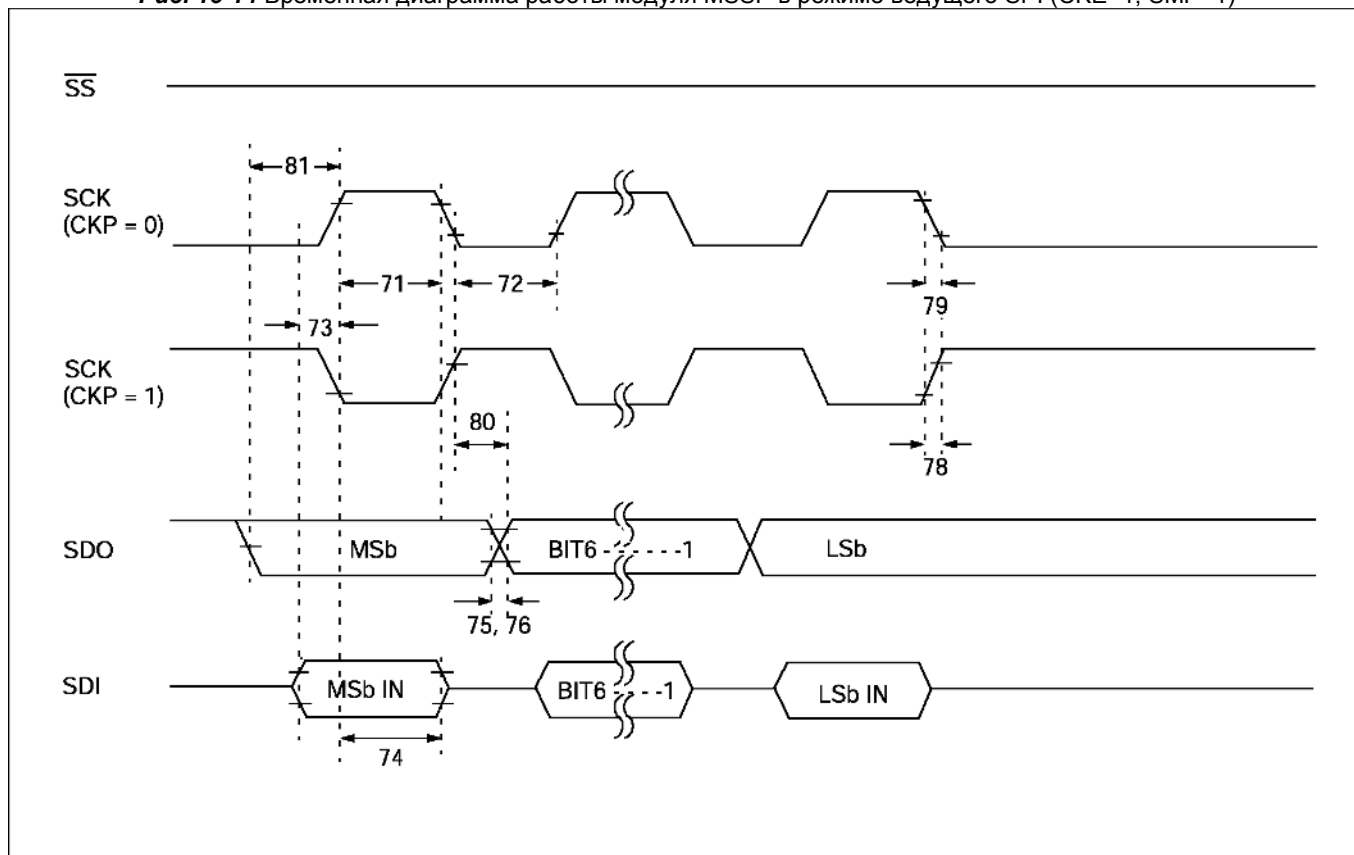
\* - Эти параметры определены, но не протестированы.

\*\* - 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.

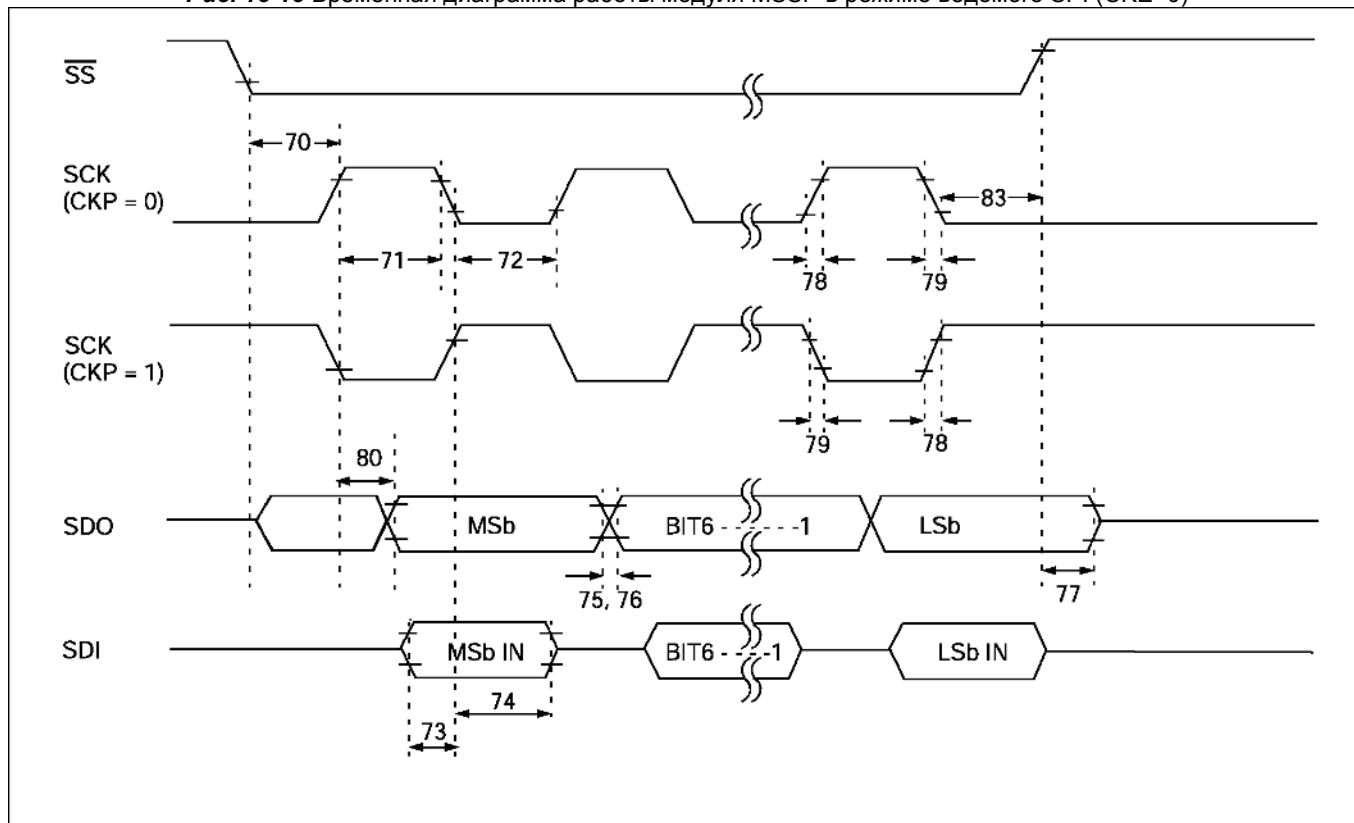
**Рис. 15-13** Временная диаграмма работы модуля MSSP в режиме ведущего SPI (CKE=0, SMP=0)



**Рис. 15-14** Временная диаграмма работы модуля MSSP в режиме ведущего SPI (CKE=1, SMP=1)



**Рис. 15-15** Временная диаграмма работы модуля MSSSP в режиме ведомого SPI (CKE=0)



**Рис. 15-16** Временная диаграмма работы модуля MSSSP в режиме ведомого SPI (CKE=1)

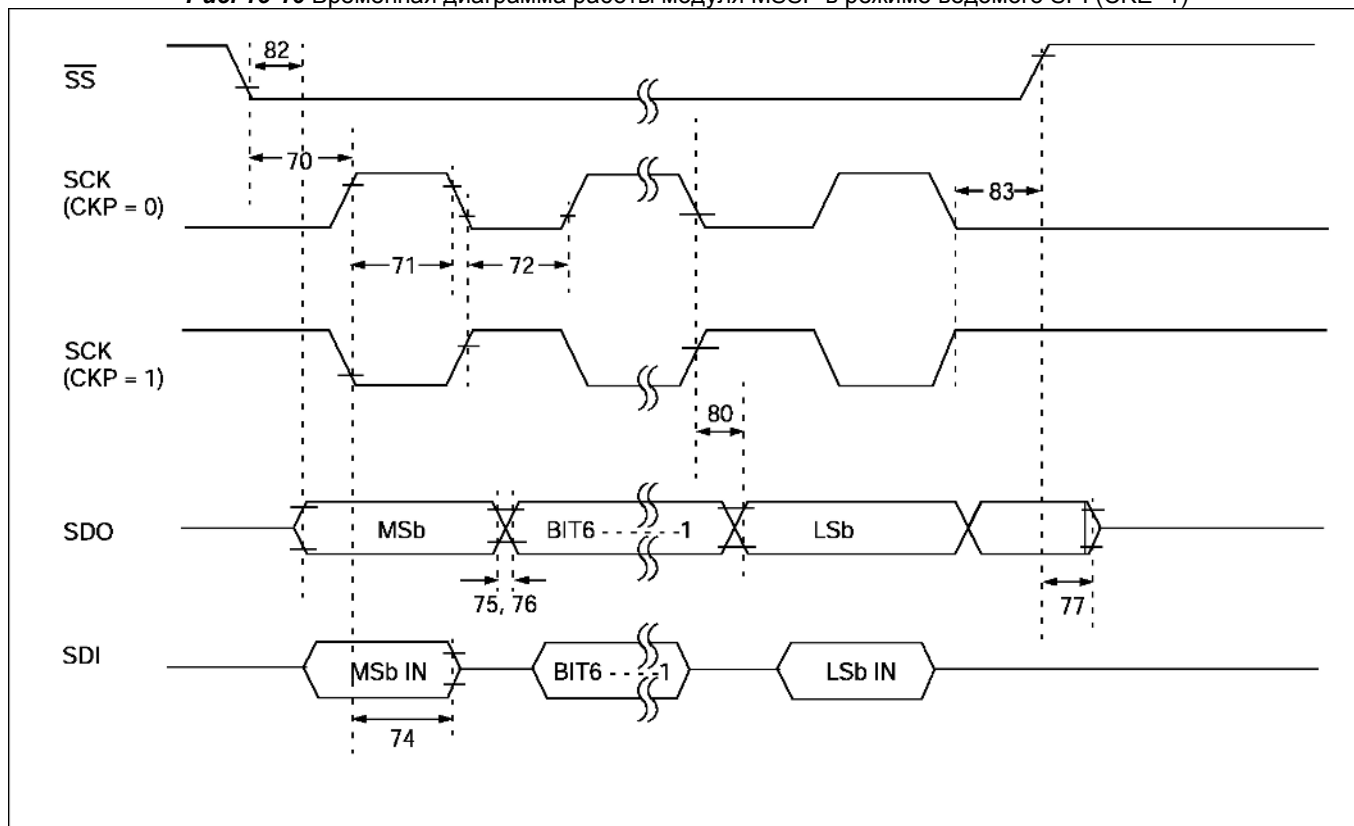
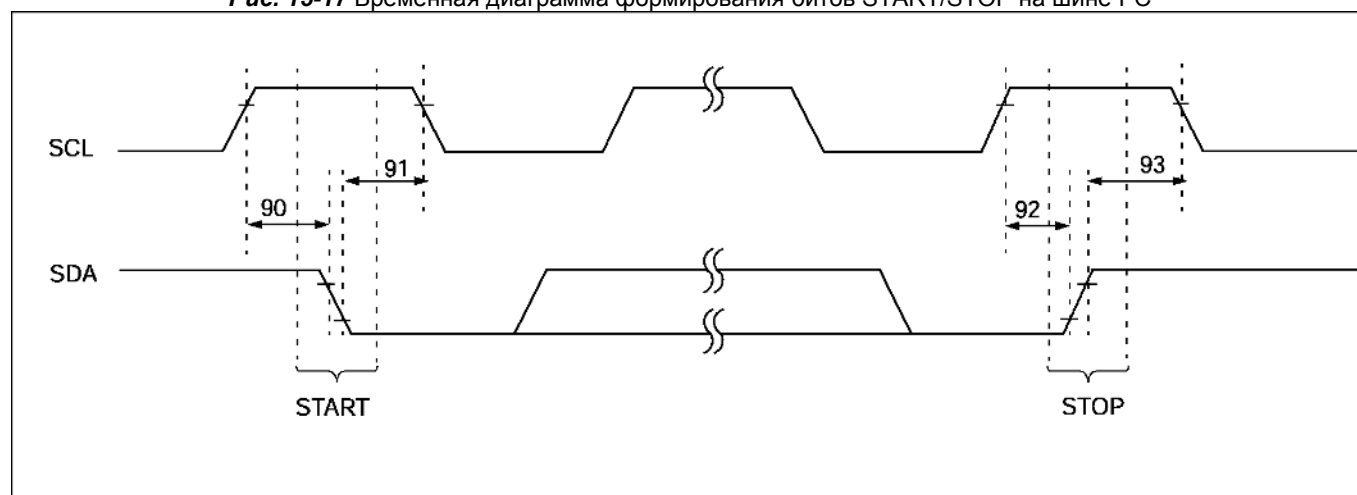


Таблица 15-7 Параметры работы модуля MSSP в режиме SPI

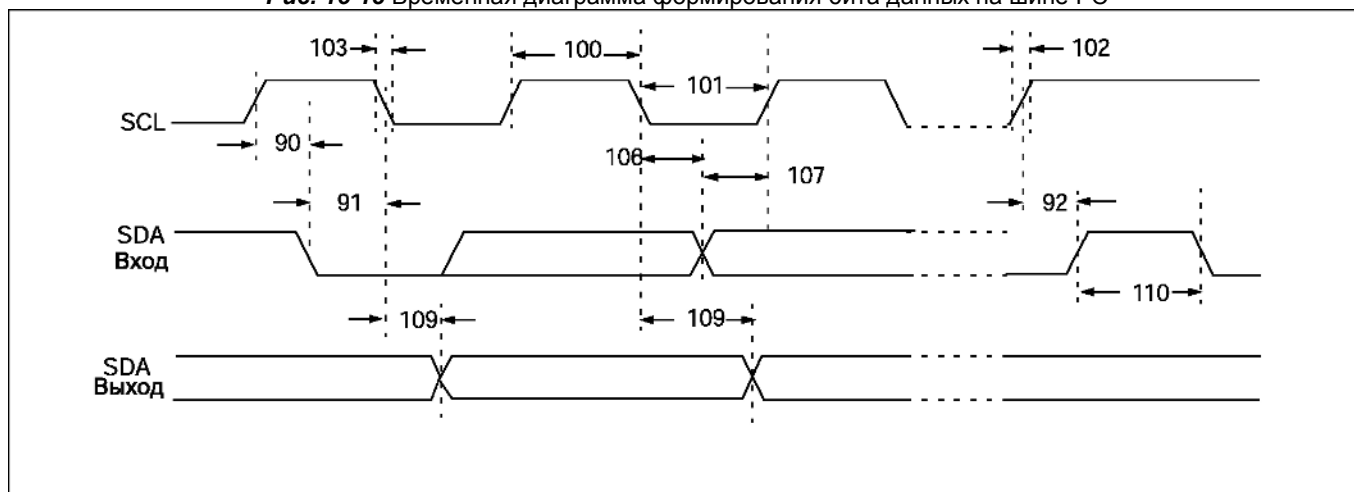
№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
70*	TssL2sch, TssL2scL	-SS↓ перед SCK↑ или SCK↓	$T_{CY}$	-	-	нс	
71*	Tsch	Высокий ур. сигн. SCK (ведомый реж.)	$T_{CY} + 20$	-	-	нс	
72*	TscL	Низкий ур. сигн. SCK (ведомый реж.)	$T_{CY} + 20$	-	-	нс	
73*	TdiV2sch, TdiV2scL	Установка данных на входе SDI относительно фронта SCK	100	-	-	нс	
74*	Tsch2diL, TscL2diL	Удержание данных на входе SDI относительно фронта SCK	100	-	-	нс	
75*	TdoR	Длительность переднего фронта на выходе SDO	F LF	10 25	25 50	нс нс	
76*	TdoF	Длительность заднего фронта на SDO	-	10	25	нс	
77*	TssH2doZ	Перевод SDO в 3-е сост. после SS↑	10	-	50	нс	
78*	TscR	Длительность переднего фронта на SCK (режим ведущего)	F LF	10 25	25 50	нс нс	
79*	TscF	Длит. заднего фронта на SCK (ведущий)	-	10	25	нс	
80*	Tsch2doV, TscL2doV	Достоверные данные на SDO после фронта SCK	F LF	- -	50 145	нс нс	
81*	TdoV2sch, TdoV2scL	Установка данных на выходе SDO после фронта SCK	$T_{CY}$	-	-	нс	
82*	TssL2doV	Достов. данные на вых. SDO после SS↓	-	-	50	нс	
83*	Tsch2ssh, TscL2ssh	SS↑ после фронта SCK	$1.5T_{CY} + 40$	-	-	нс	

\* - 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. 15-17 Временная диаграмма формирования битов START/STOP на шине I<sup>2</sup>CТаблица 15-8 Параметры формирования битов START/STOP на шине I<sup>2</sup>C

№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание	
90	Tsu:sta	Установка условия START	Режим 100 кГц	4700	-	-	нс	Только при формировании бита повторный START
			Режим 400 кГц	600	-	-		
91	Thd:sta	Удержание условия START	Режим 100 кГц	4000	-	-	нс	После этого форм. первый импульс тактового сигнала
			Режим 400 кГц	600	-	-		
92	Tsu:sto	Установка условия STOP	Режим 100 кГц	4700	-	-	нс	
			Режим 400 кГц	600	-	-		
93	Thd:sto	Удержание условия STOP	Режим 100 кГц	4000	-	-	нс	
			Режим 400 кГц	600	-	-		

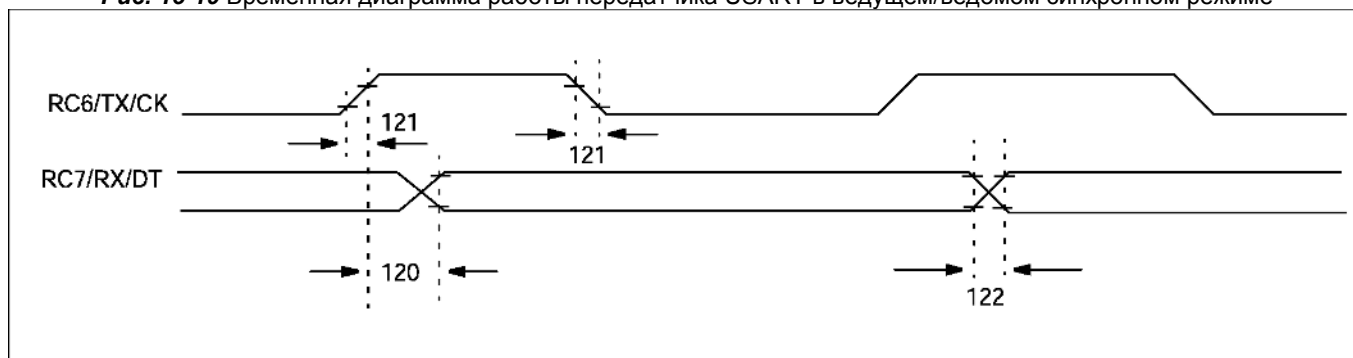
Рис. 15-18 Временная диаграмма формирования бита данных на шине I<sup>2</sup>CТаблица 15-9 Параметры формирования бита данных на шине I<sup>2</sup>C

№ пар.	Обоз.	Описание	Мин.	Макс.	Ед.	Примечание	
100	Thigh	Длительность высокого уровня тактового сигнала	Режим 100 кГц	4.0	-	мкс	Мин. F <sub>OSC</sub> 1.5МГц
			Режим 400 кГц	0.6	-	мкс	Мин. F <sub>OSC</sub> 10МГц
			Модуль SSP	0.5T <sub>СУ</sub>	-		
101	Tlow	Длительность низкого уровня тактового сигнала	Режим 100 кГц	4.7	-	мкс	Мин. F <sub>OSC</sub> 1.5МГц
			Режим 400 кГц	1.3	-	мкс	Мин. F <sub>OSC</sub> 10МГц
			Модуль SSP	0.5T <sub>СУ</sub>	-		
102	Tr	Долит. переднего фронта на SDA и SCL	Режим 100 кГц	-	1000	нс	
			Режим 400 кГц	20 + 0.1 C <sub>b</sub>	300	нс	10пФ ≤ C <sub>b</sub> ≤ 400пФ
103	Tf	Долит. заднего фронта на SDA и SCL	Режим 100 кГц	-	300	нс	
			Режим 400 кГц	20 + 0.1 C <sub>b</sub>	300	нс	10пФ ≤ C <sub>b</sub> ≤ 400пФ
90	Tsu:sta	Установка условия START	Режим 100 кГц	4.7	-	мкс	Только при формировании бита повторный START
			Режим 400 кГц	0.6	-	мкс	
91	Thd:sta	Удержание условия START	Режим 100 кГц	4.0	-	мкс	После этого форм. первый импульс тактового сигнала
			Режим 400 кГц	0.6	-	мкс	
106	Thd:dat	Удержание данных на входе	Режим 100 кГц	0	-	нс	
			Режим 400 кГц	0	0.9	мкс	
107	Tsu:dat	Установка данных на входе	Режим 100 кГц	250	-	нс	Примечание 2
			Режим 400 кГц	100	-	нс	
92	Tsu:sto	Установка условия STOP	Режим 100 кГц	4.7	-	мкс	
			Режим 400 кГц	0.6	-	мкс	
109	Taa	Достоверность сигнала на выходе	Режим 100 кГц	-	3500	нс	Примечание 1
			Режим 400 кГц	-	-	нс	
110	Tbuf	Время не занятости шины	Режим 100 кГц	4.7	-	мкс	Задержка перед новой передачей
			Режим 400 кГц	1.3	-	мкс	
	C <sub>b</sub>	Емкостная нагрузка линии	-	400	пФ		

**Примечания:**

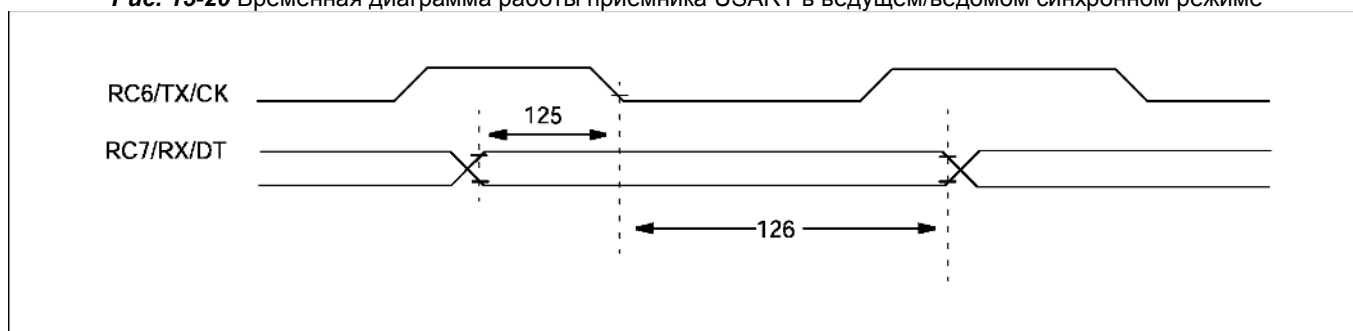
1. Необходимо выдерживать эту минимальную задержку относительно заднего фронта SCL, чтобы избежать ложное формирование битов START и STOP.
2. Устройства с высокоскоростным режимом обмена (400кГц) могут использоваться в стандартном режиме (100кГц), но требование Tsu:dat ≥ 250нс необходимо выполнять. Это условие автоматически будет выполняться, если не возникает удержания линии SCL в низком логическом уровне. Если возникает удержание линии SCL в низком логическом уровне, то необходимо сформировать бит данных на SDA Tr.max + Tsu:dat = 1000 + 250 = 1250 нс (согласно спецификации I<sup>2</sup>C) прежде, чем SCL будет "отпущена".



**Рис. 15-19** Временная диаграмма работы передатчика USART в ведущем/ведомом синхронном режиме**Таблица 15-10** Параметры работы передатчика USART в ведущем/ведомом синхронном режиме

№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
120	TckH2dtV	Действ. данные после перехода такт. сигнала в высокий уровень	F	-	-	80	нс
			LF	-	-	100	нс
121	Tckrf	Длительность заднего/переднего фронта такт. сигн. (ведущий)	F	-	-	45	нс
			LF	-	-	50	нс
122	Tdtrf	Длительность переднего/заднего фронта данных	F	-	-	45	нс
			LF	-	-	50	нс

\*\* - 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. 15-20** Временная диаграмма работы приемника USART в ведущем/ведомом синхронном режиме**Таблица 15-11** Параметры работы приемника USART в ведущем/ведомом синхронном режиме

№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание
125	TdtV2ckL	Установка данных после CK↓	15	-	-	нс	
126	TckL2dl	Удержание данных после CK↓	15	-	-	нс	

\*\* - 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.

**Таблица 15-12 PIC16F87X-04 (Коммерческий, Промышленный, Расширенный)  
 PIC16F87X-10 (Расширенный)  
 PIC16F87X-20 (Коммерческий, Промышленный)  
 PIC16LF87X-04 (Коммерческий, Промышленный)**

№ пар.	Обоз.	Описание	Мин.	Тип**	Макс.	Ед.	Примечание	
A01	N <sub>R</sub>	Разрядность	-	-	10	бит	$V_{REF} = V_{DD} = 5.12В,$ $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A03	E <sub>IL</sub>	Интегральная погрешность	-	-	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12В,$ $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A04	E <sub>DL</sub>	Дифференциальная погрешность	-	-	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12В,$ $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A06	E <sub>OFF</sub>	Ошибка смещения	-	-	$< \pm 2$	LSb	$V_{REF} = V_{DD} = 5.12В,$ $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A07	E <sub>GN</sub>	Ошибка усиления	-	-	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12В,$ $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A10	-	Монотонность <sup>(3)</sup>	Гарантируется			-	$V_{SS} \leq V_{AIN} \leq V_{REF}$	
A20	V <sub>REF</sub>	Опорное напряжение (V <sub>REF+</sub> -V <sub>REF-</sub> )	2.0	-	V <sub>DD</sub> + 0.3	В	Минимальное значение для 10-разрядного АЦП	
A21	V <sub>REF+</sub>	Положительное опорное напр.	AV <sub>DD</sub> - 2.5	-	AV <sub>DD</sub> + 0.3	В		
A22	V <sub>REF-</sub>	Отрицательное опорное напр.	AV <sub>SS</sub> - 0.3	-	V <sub>REF+</sub> - 2.0	В		
A25	V <sub>AIN</sub>	Аналоговый вход	V <sub>SS</sub> - 0.3	-	V <sub>REF</sub> + 0.3	В		
A30	Z <sub>AIN</sub>	Сопrotивление источника сигн.	-	-	10.0	кОм		
A40	I <sub>AD</sub>	Потребляемый ток АЦП	F	-	220	-	мкА	Среднее потребление при включенном АЦП <sup>(1)</sup>
			LF	-	90	-	мкА	
A50	I <sub>REF</sub>	Потребляемый ток от источника опорного напряжения <sup>(2)</sup>	10	-	1000	-	мкА	Во время выборки V <sub>AIN</sub> . Основано на дифференц. значении заряда C <sub>HOLD</sub> до V <sub>AIN</sub> . Во время преобразования.
			-	-	10	-	мкА	

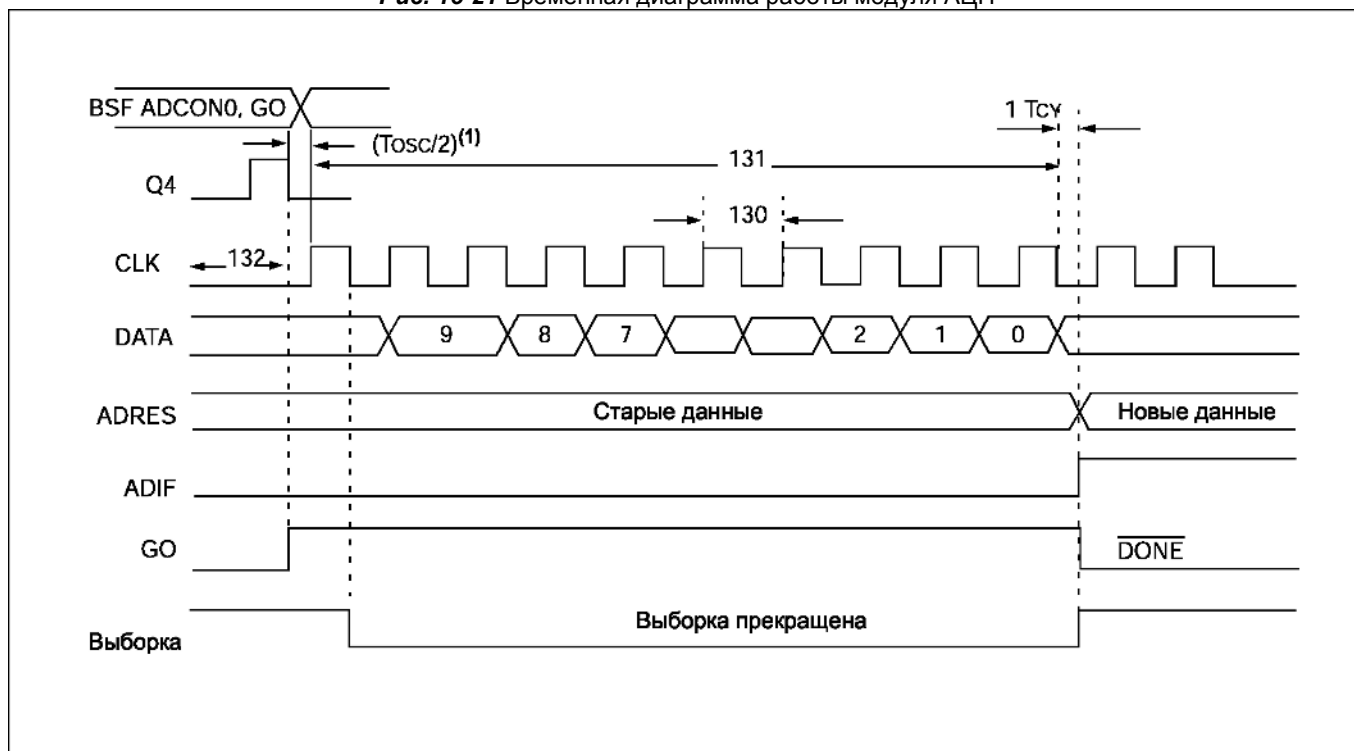
\* - Эти параметры определены, но не протестированы.

\*\* - В столбце "Тип." приведены параметры при V<sub>DD</sub>=5.0В @ 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

**Примечания:**

1. Выключенный модуль АЦП не потребляет тока, кроме токов утечки.
2. Ток со входа RA3 или V<sub>DD</sub> в зависимости от выбранного источника опорного напряжения.
3. Результат АЦП никогда не уменьшается с увеличением напряжения на входе и не имеет кодов отсутствия напряжения.

Рис. 15-21 Временная диаграмма работы модуля АЦП



**Примечание.** Если используется внутренний RC генератор для АЦП, то добавляется время  $T_{CY}$  перед запуском АЦП, позволяющее выполнить команду SLEEP.

Таблица 15-13 Параметры работы модуля АЦП

№ пар.	Обоз.	Описание	Мин.	Тип.**	Макс.	Ед.	Примечание	
130	$T_{AD}$	Период тактового сигнала АЦП	F	1.6	-	-	мкс	Основа $T_{OSC}$ , $V_{REF} \geq 3.0$ В
			LF	3.0	-	-	мкс	Основа $T_{OSC}$ , $V_{REF} \geq 2.0$ В
			F	2.0	4.0	6.0	мкс	RC генератор АЦП
			LF	3.0	6.0	9.0	мкс	RC генератор АЦП
131	$T_{CNV}$	Время преобразования <sup>(1)</sup>		-	12	$T_{AD}$		
132	$T_{ACQ}$	Время выборки	(2)	40	-	мкс	Примечание 3	
			10*	-	-	мкс		
134	$T_{GO}$	Старт преобразования относительно Q4	-	$T_{OSC}/2^{***}$	-	-	Примечание 4	

\* - Эти параметры определены, но не протестированы.

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

\*\*\* - Оценочное значение.

**Примечания:**

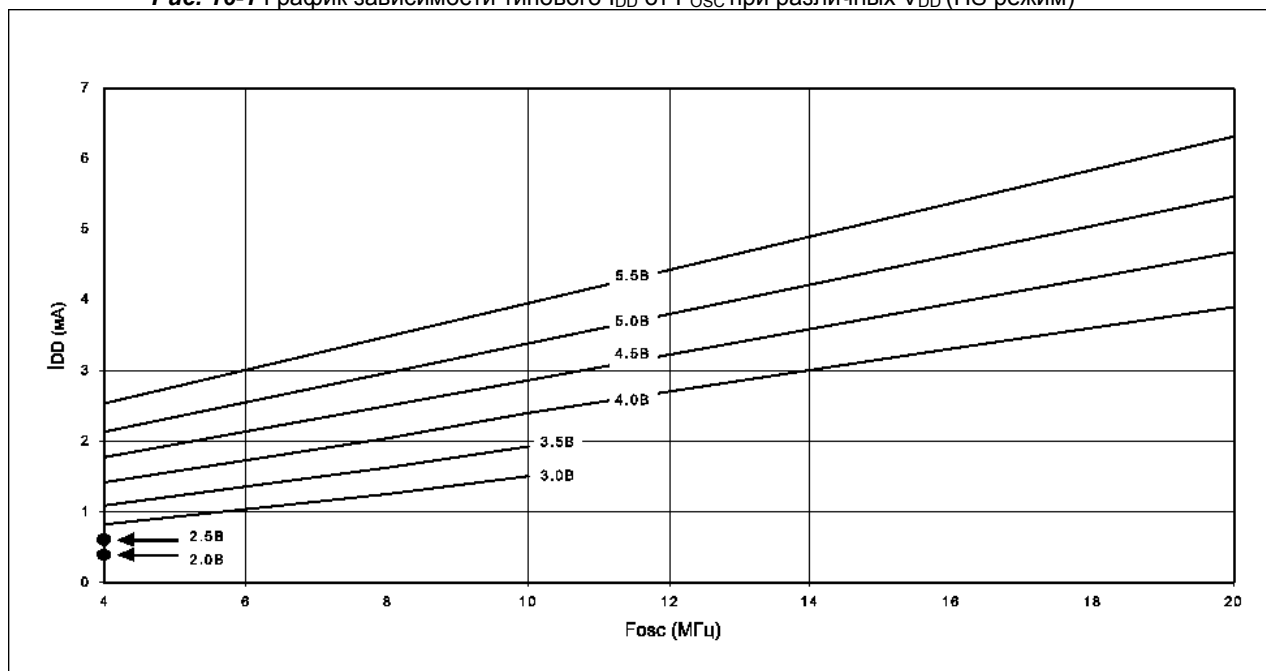
1. Регистр ADRESL может быть прочитан в следующем цикле.
2. Смотрите раздел 11.1 для выбора минимального значения.
3. Минимальное время - задержка усилителя. Может использоваться, если напряжение на входе изменилось не более, чем на 1 LSb (т.е. 20мВ @ 5.12В) от последнего измерения.
4. Если используется внутренний RC генератор для АЦП, то добавляется время  $T_{CY}$  перед запуском АЦП, позволяющее выполнить команду SLEEP.

## 16.0 Характеристики микроконтроллеров

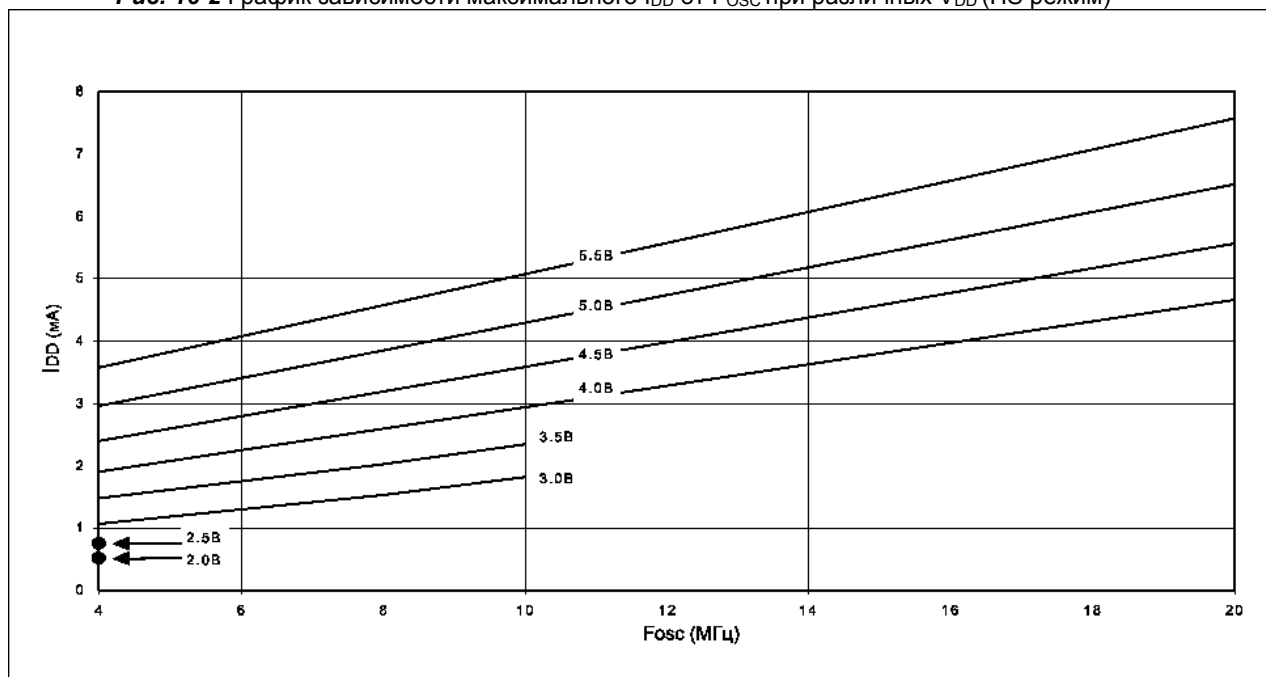
Графики в этом разделе не проверены и предназначены только для оценки при разработке устройств. В некоторых графиках представлены данные вне рабочего диапазона (в частности для напряжения питания  $V_{DD}$ ). Это только информационные данные.

Данные, представленные в этом разделе, являются среднестатистическим результатом испытаний большого числа микроконтроллеров в течение длительного времени. Типовое значение подразумевает среднее, а минимальное и максимальное - соответственно (среднее -  $3\sigma$ ) и (среднее +  $3\sigma$ ), где  $\sigma$  - стандартный разброс.

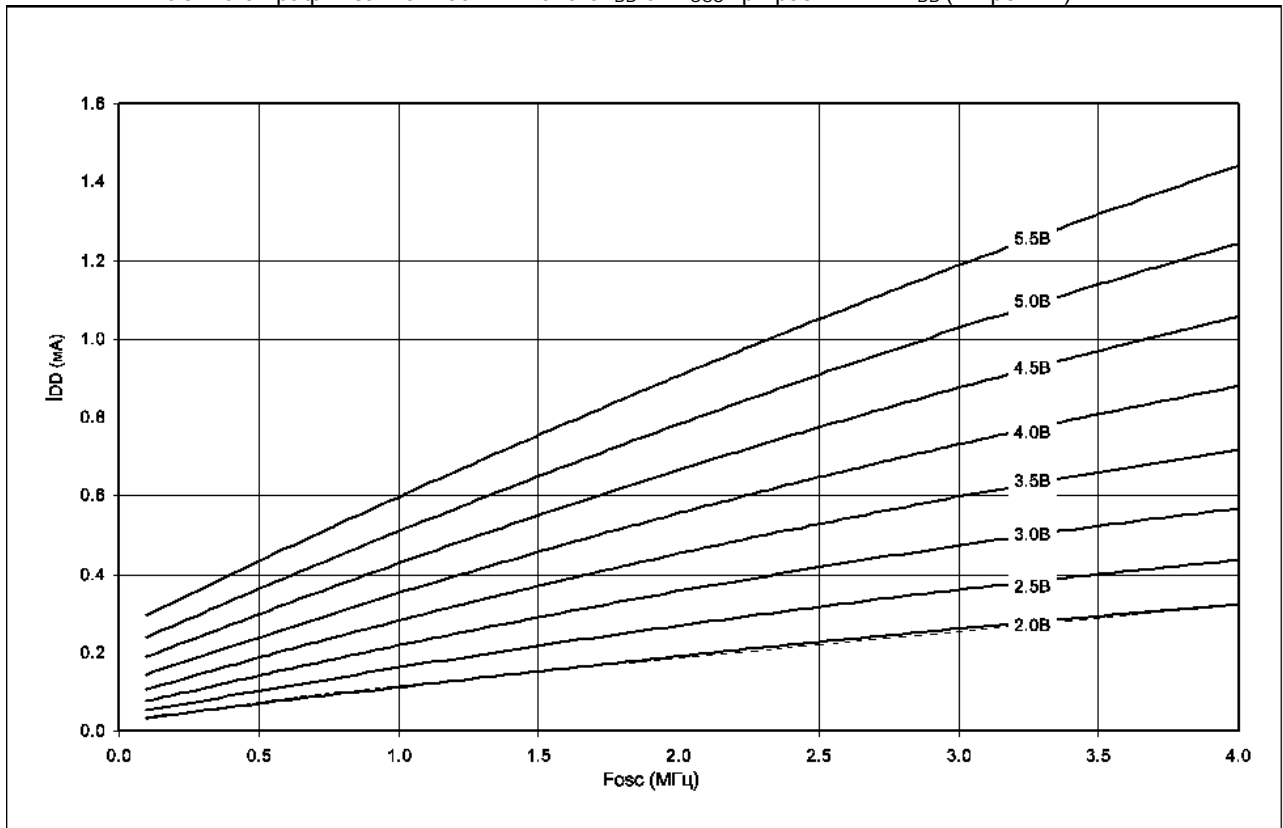
**Рис. 16-1** График зависимости типового  $I_{DD}$  от  $F_{OSC}$  при различных  $V_{DD}$  (HS режим)



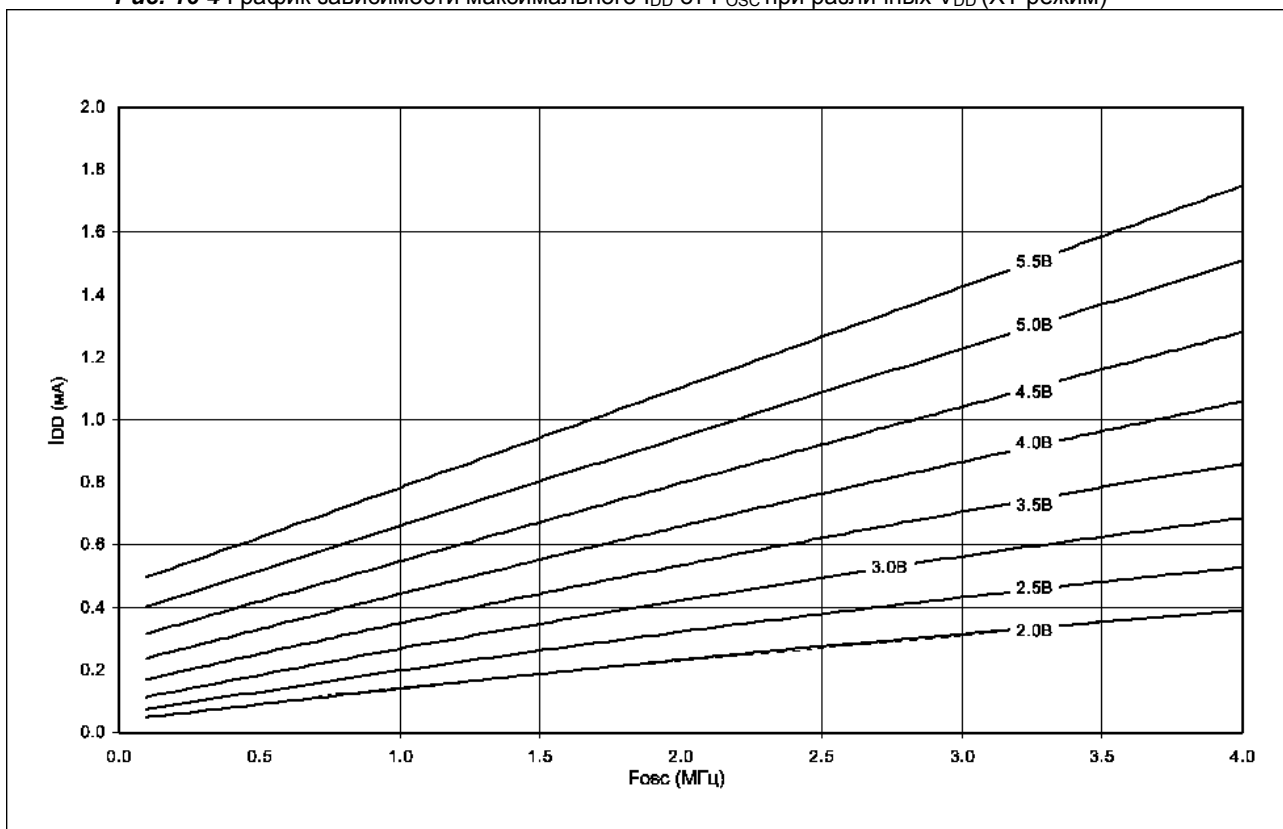
**Рис. 16-2** График зависимости максимального  $I_{DD}$  от  $F_{OSC}$  при различных  $V_{DD}$  (HS режим)



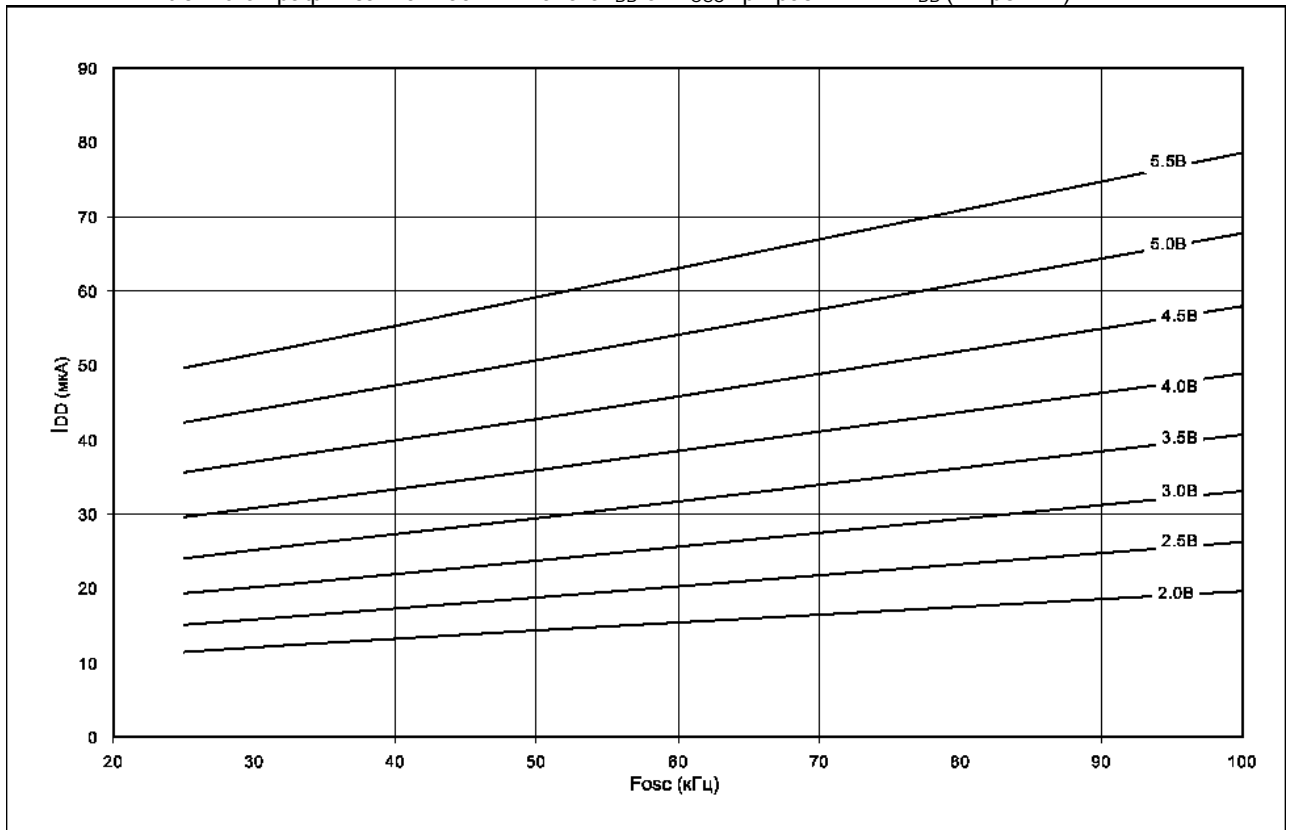
**Рис. 16-3** График зависимости типового  $I_{DD}$  от  $F_{OSC}$  при различных  $V_{DD}$  (XT режим)



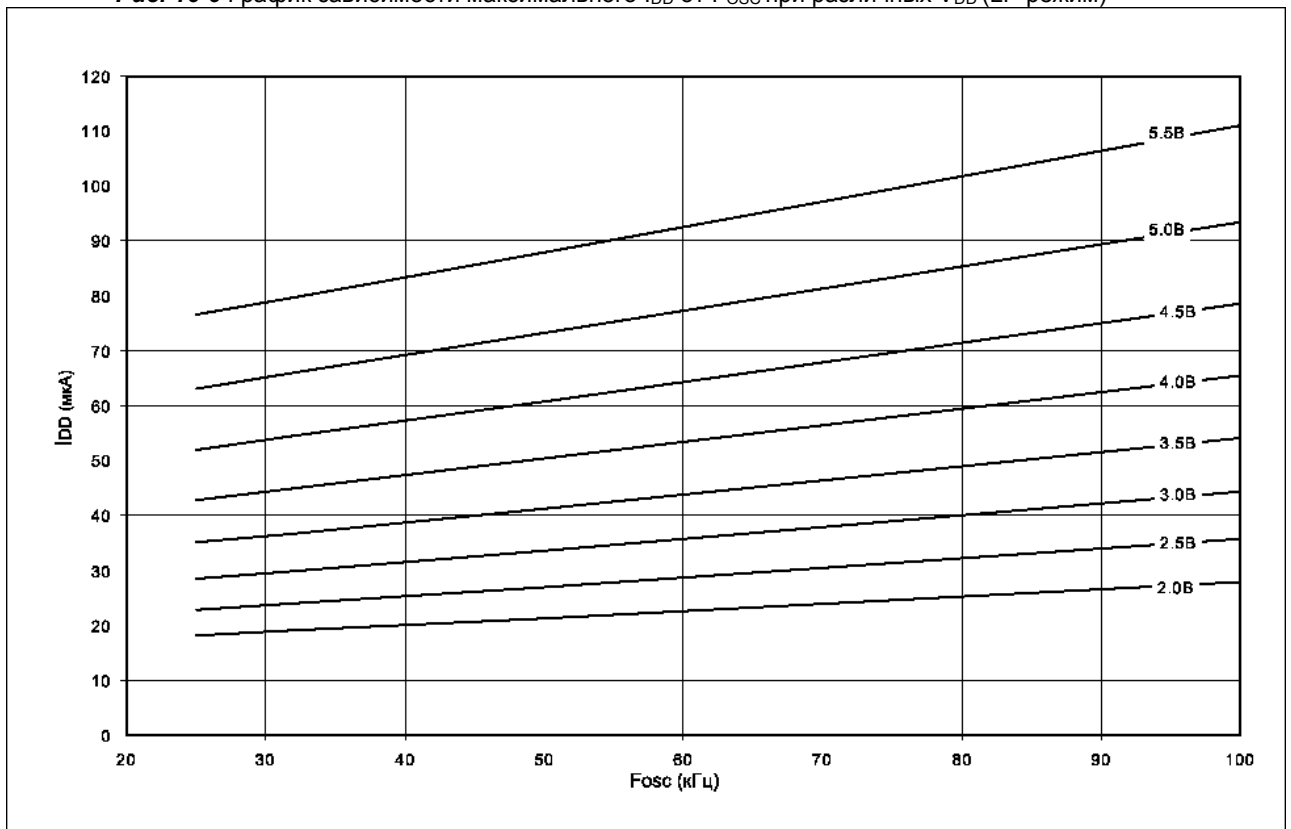
**Рис. 16-4** График зависимости максимального  $I_{DD}$  от  $F_{OSC}$  при различных  $V_{DD}$  (XT режим)



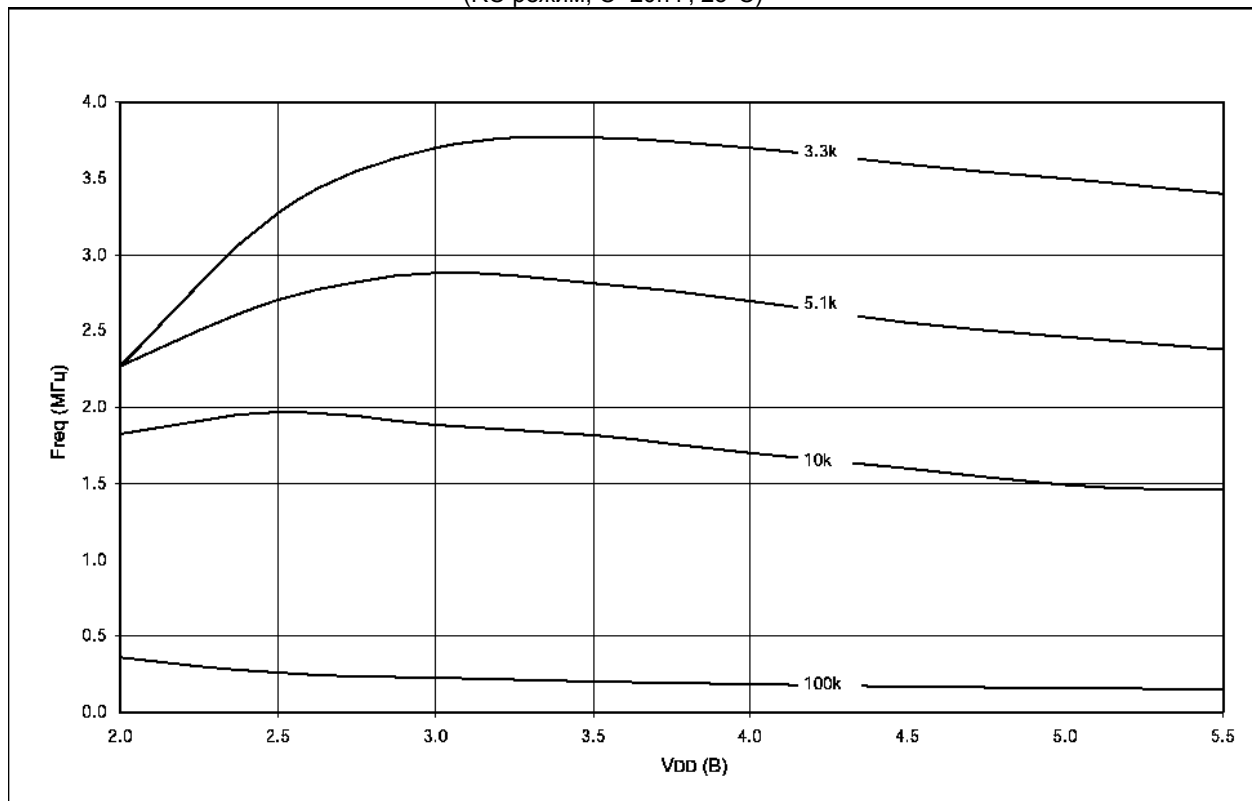
**Рис. 16-5** График зависимости типового  $I_{DD}$  от  $F_{OSC}$  при различных  $V_{DD}$  (LP режим)



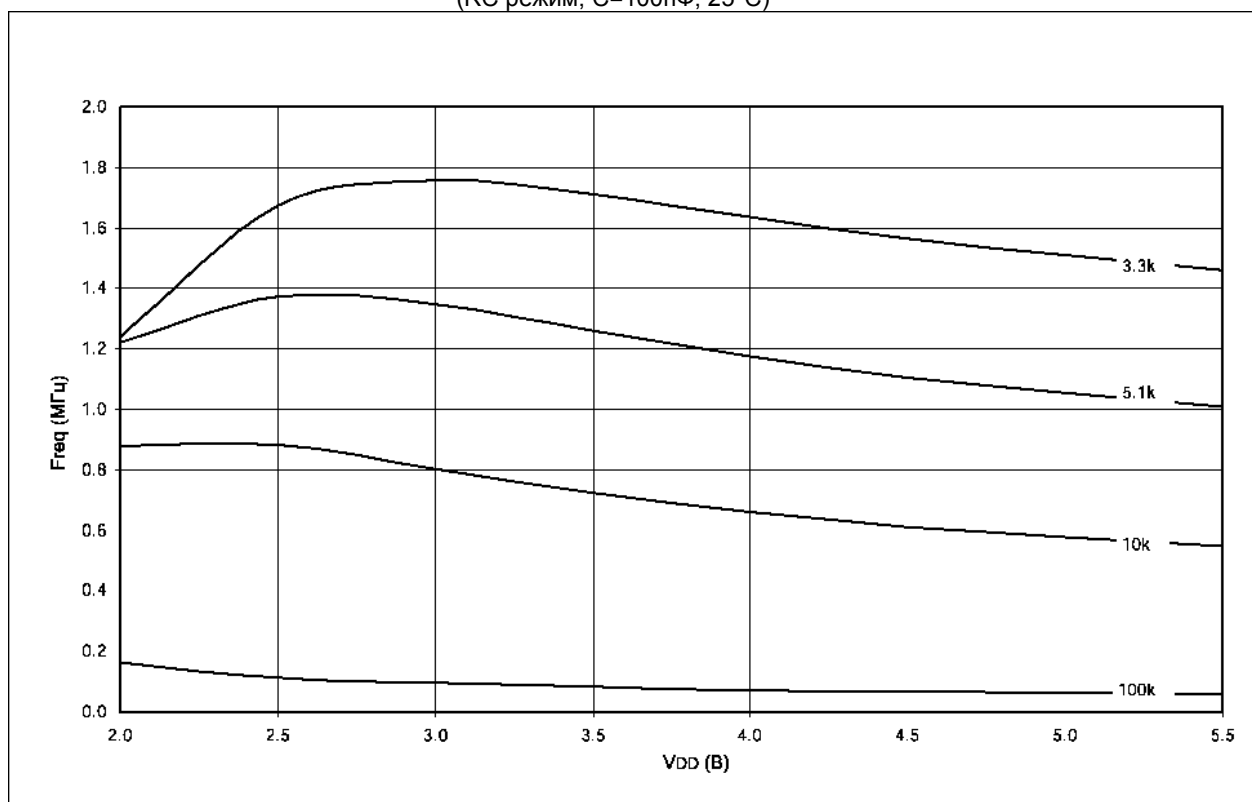
**Рис. 16-6** График зависимости максимального  $I_{DD}$  от  $F_{OSC}$  при различных  $V_{DD}$  (LP режим)



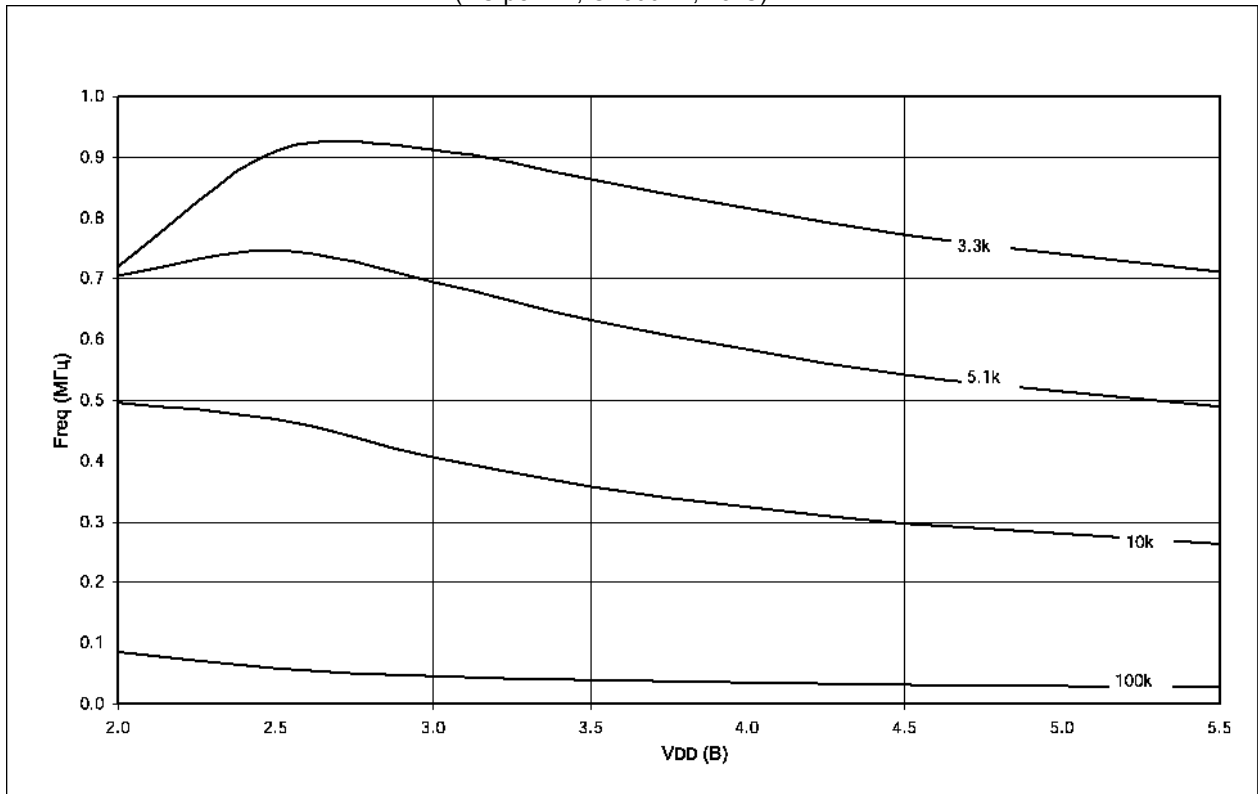
**Рис. 16-7** График зависимости  $F_{OSC}$  от  $V_{DD}$  при различных R (RC режим, C=20пФ, 25°C)



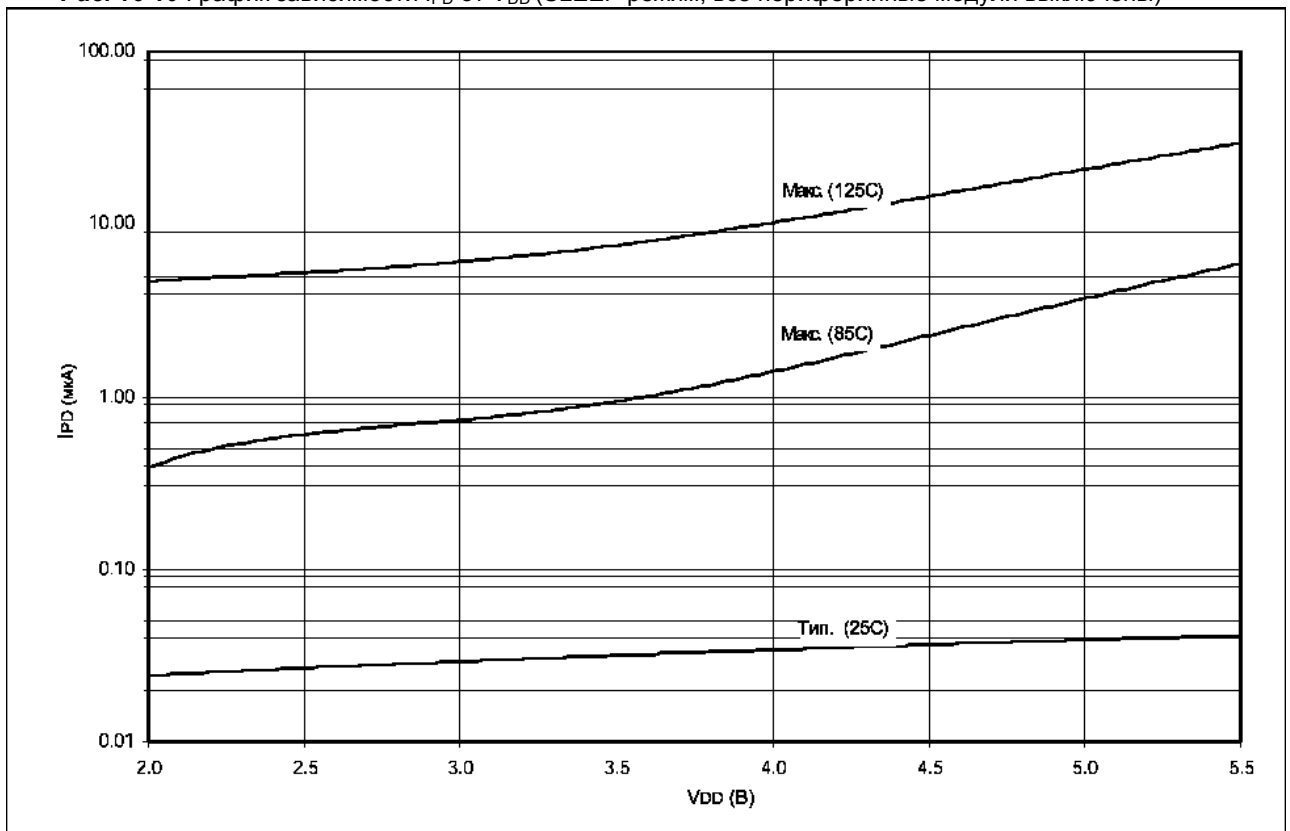
**Рис. 16-8** График зависимости  $F_{OSC}$  от  $V_{DD}$  при различных R (RC режим, C=100пФ, 25°C)



**Рис. 16-9** График зависимости  $F_{OSC}$  от  $V_{DD}$  при различных R (RC режим, C=300пФ, 25°C)

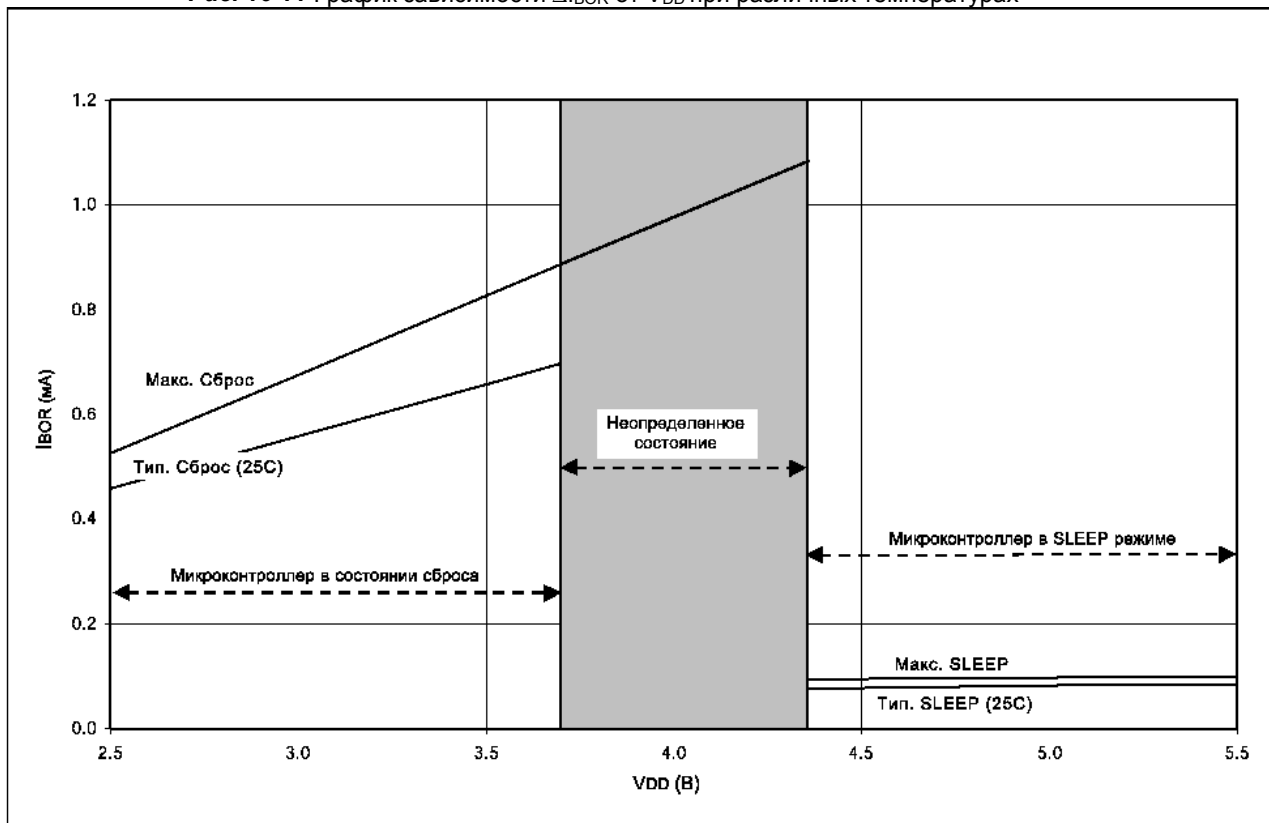


**Рис. 16-10** График зависимости  $I_{PD}$  от  $V_{DD}$  (SLEEP режим, все периферийные модули выключены)

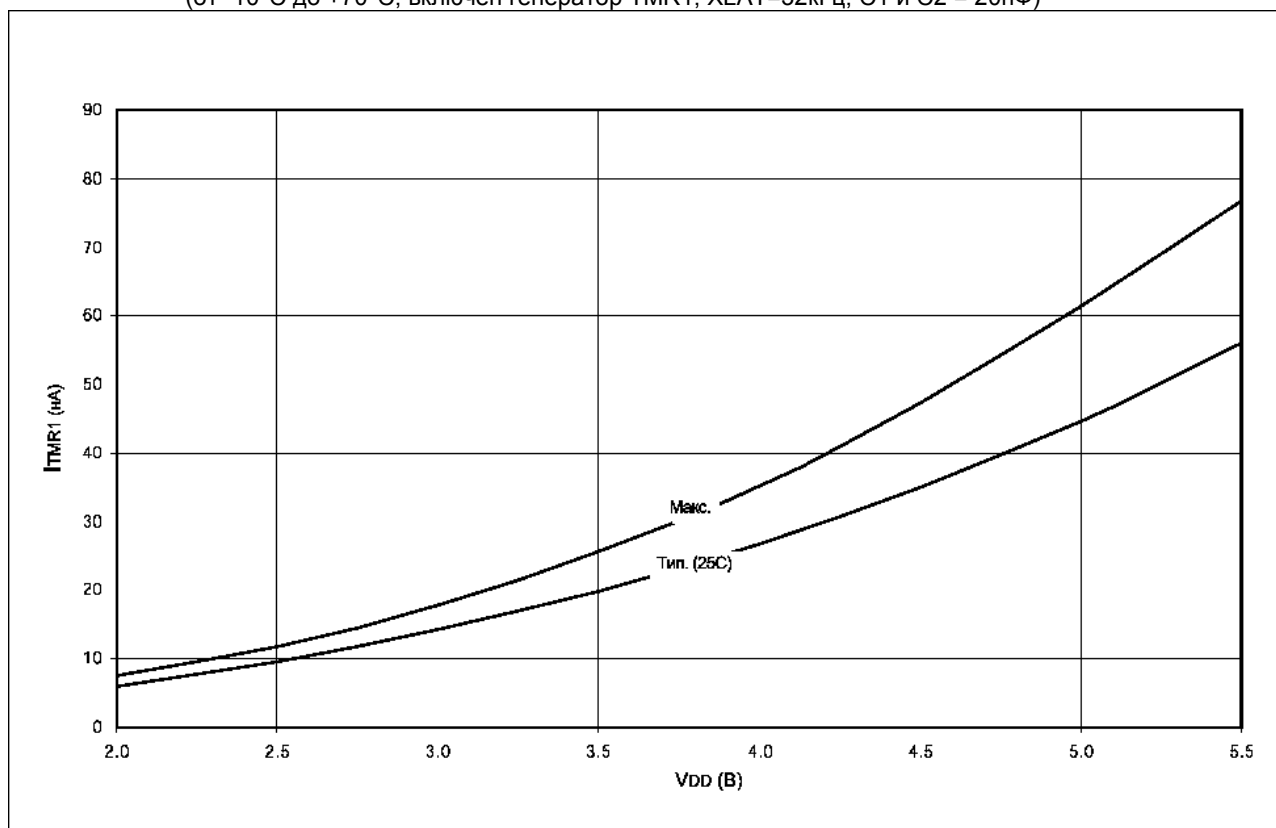


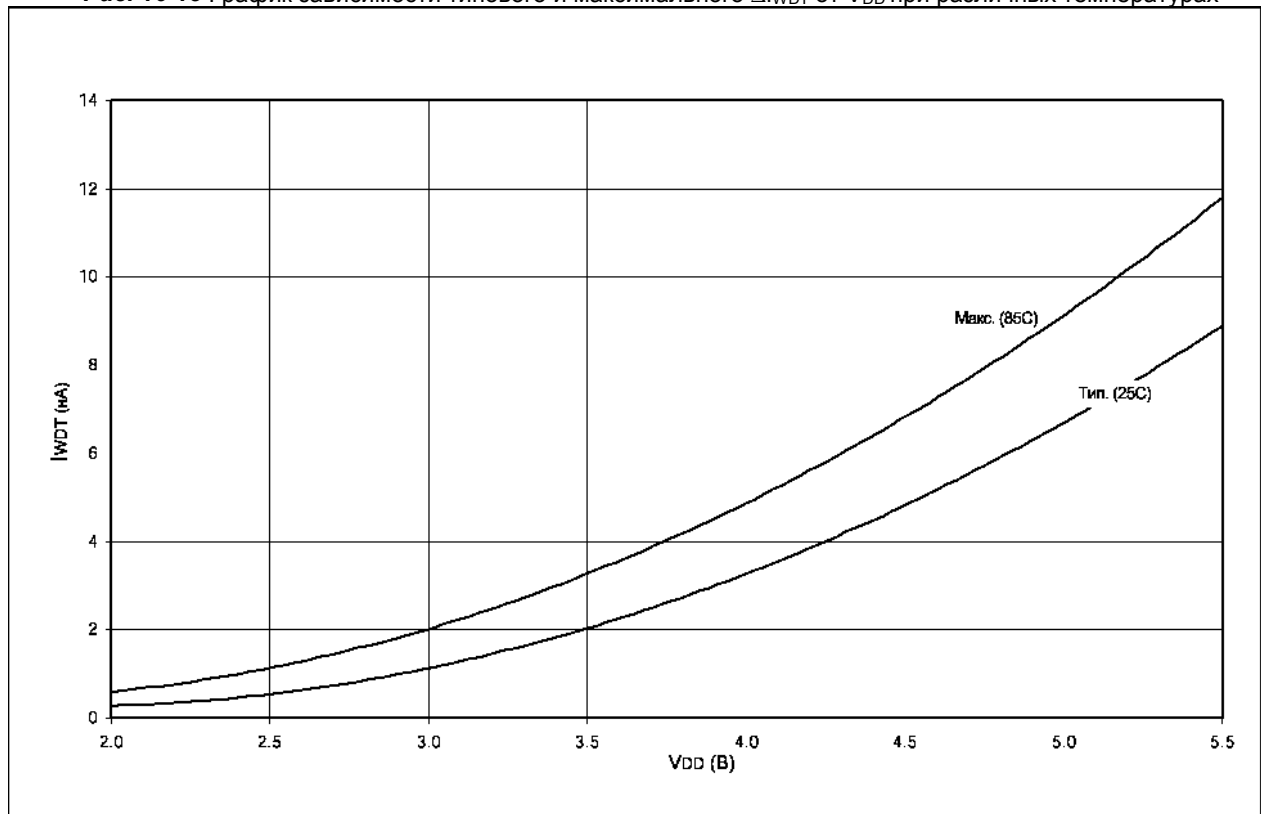
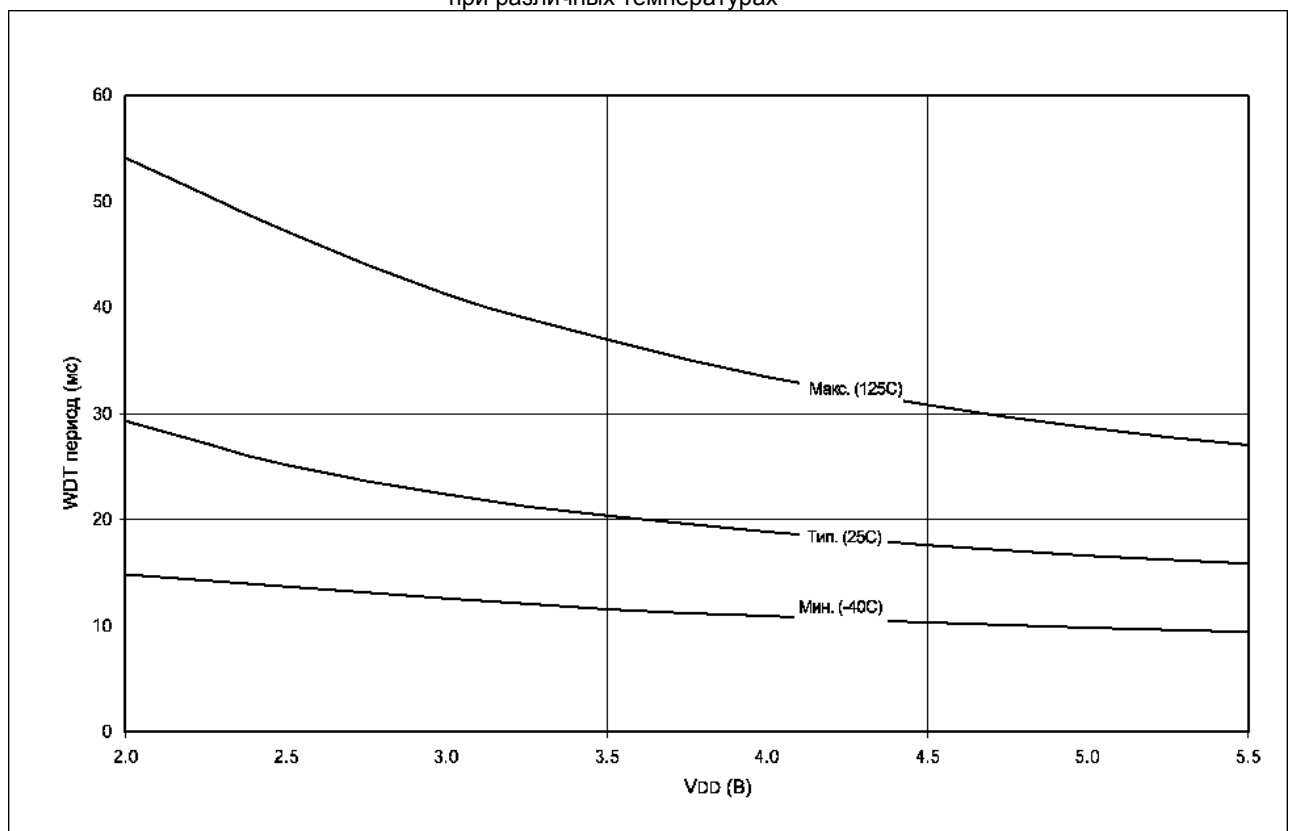


**Рис. 16-11** График зависимости  $\Delta I_{BOR}$  от  $V_{DD}$  при различных температурах

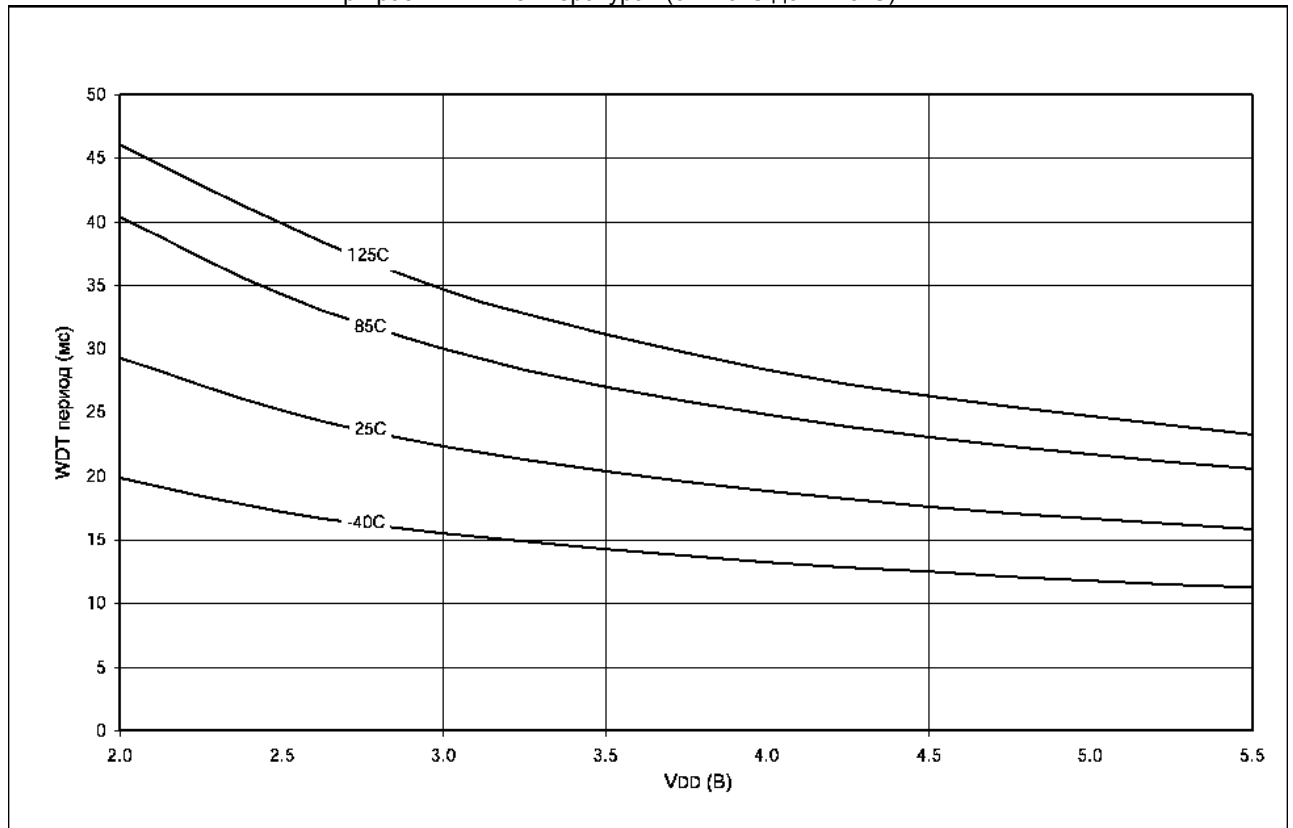


**Рис. 16-12** График зависимости типового и максимального  $\Delta I_{TMR1}$  от  $V_{DD}$  при различных температурах (от  $-10^{\circ}\text{C}$  до  $+70^{\circ}\text{C}$ , включен генератор TMR1,  $XLAT=32\text{кГц}$ ,  $C1$  и  $C2 = 20\text{пФ}$ )

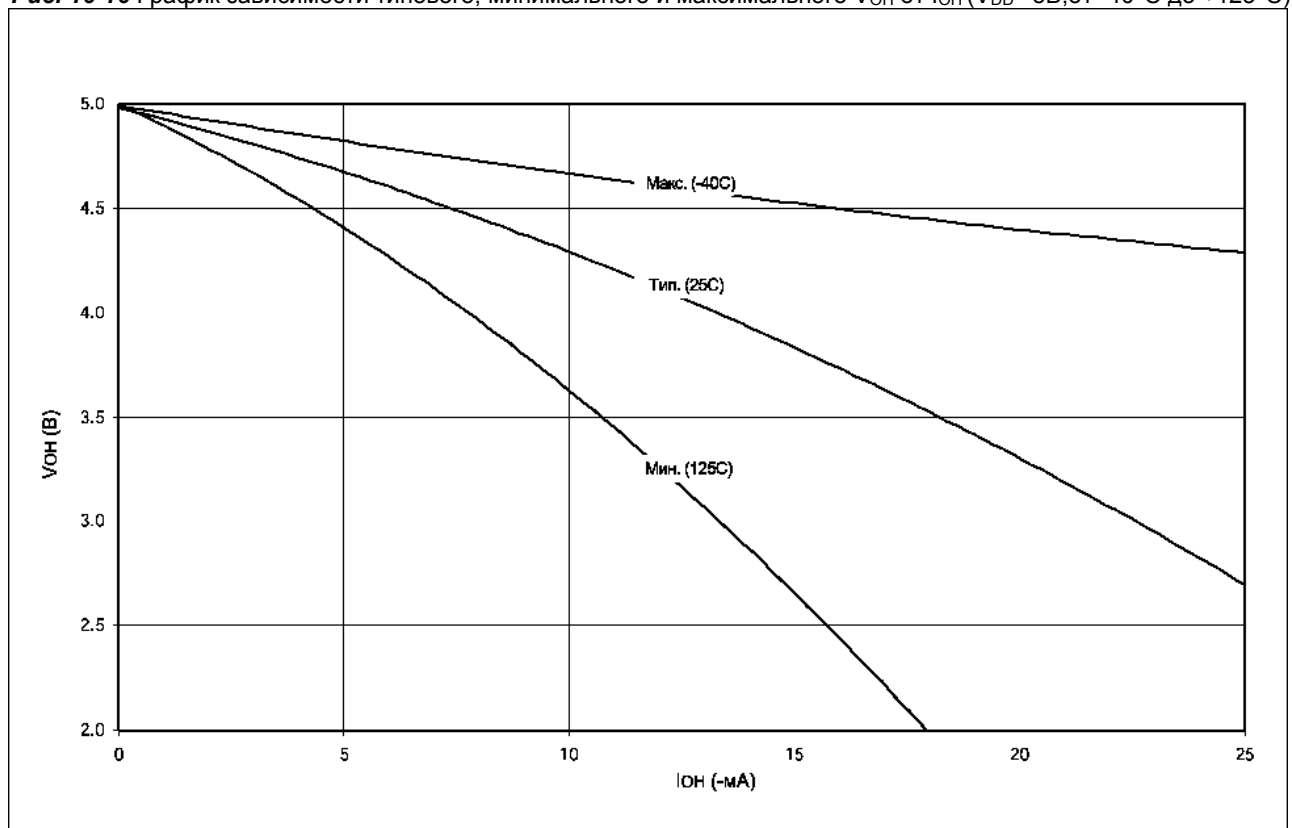


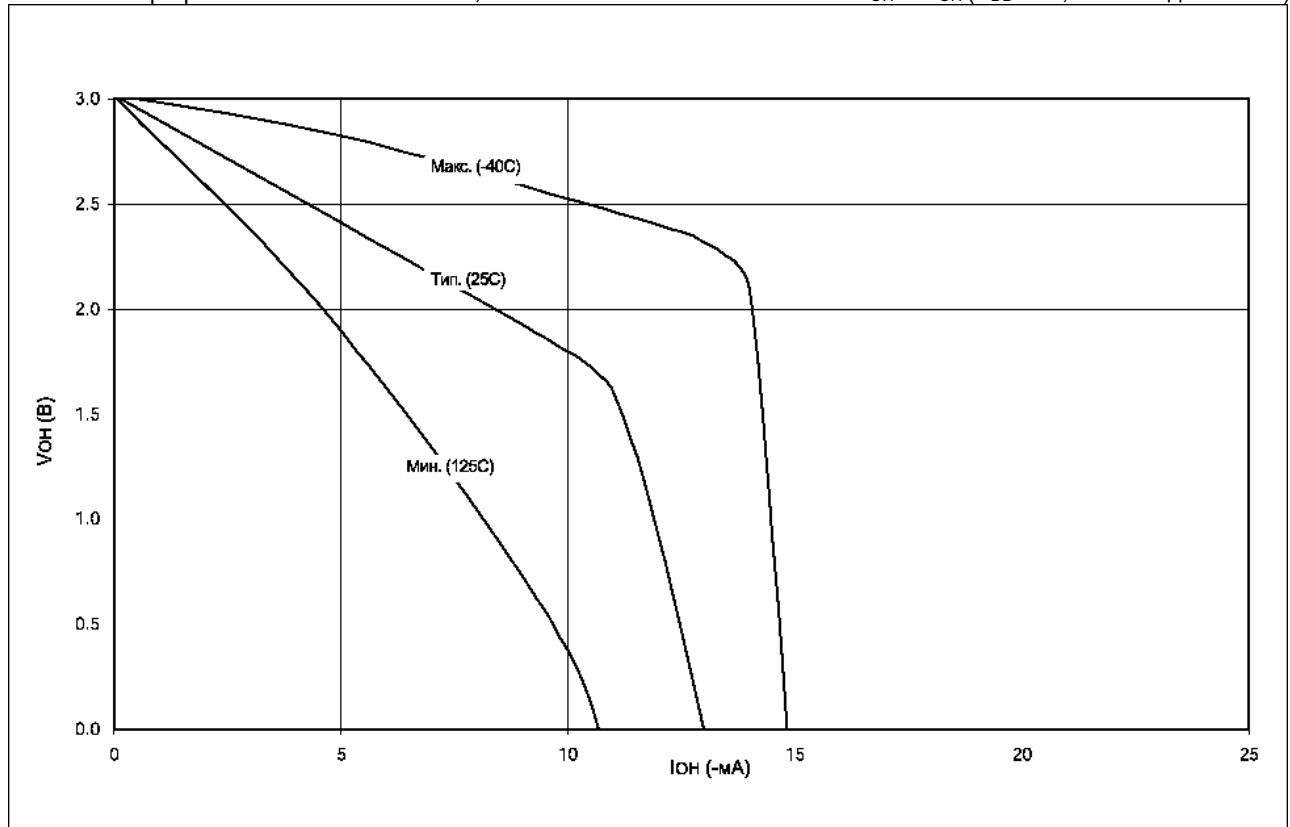
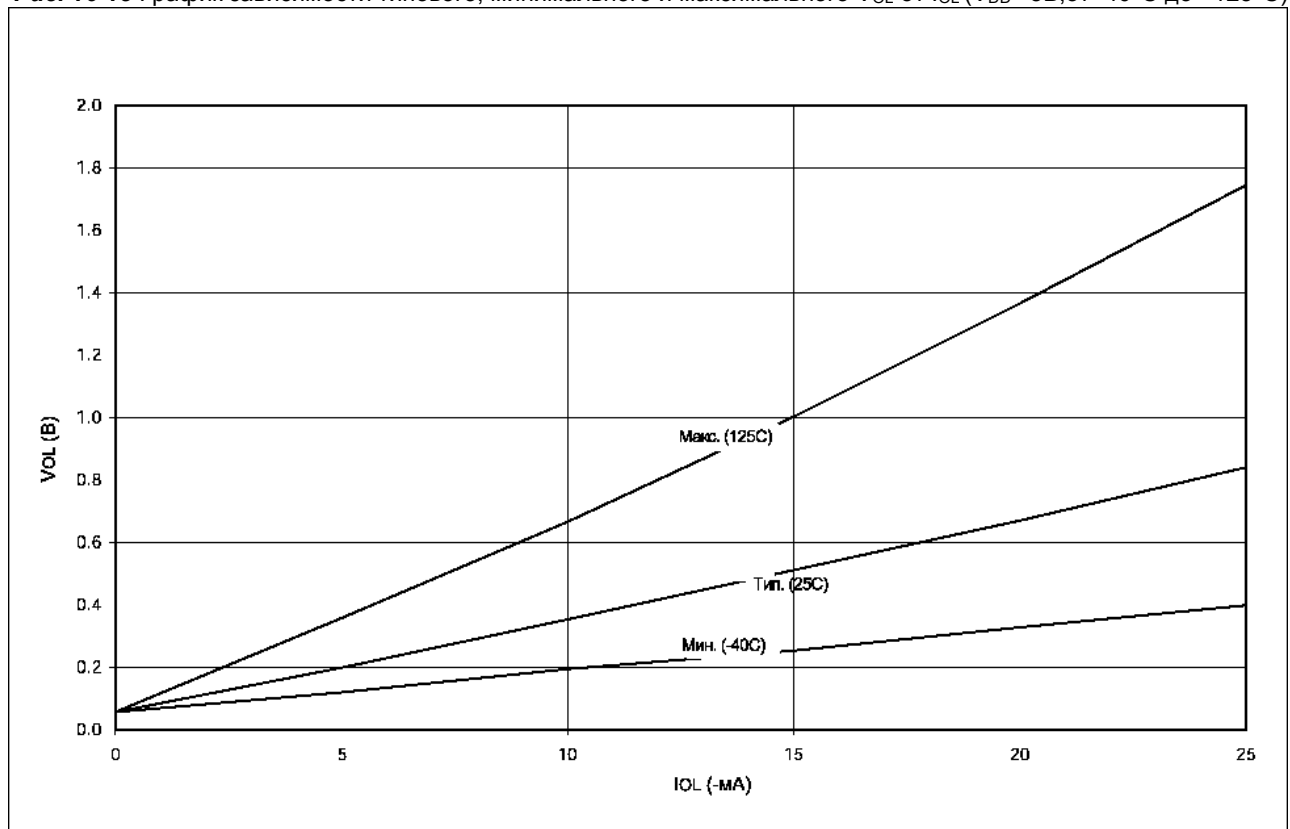
**Рис. 16-13** График зависимости типового и максимального  $\Delta I_{WDT}$  от  $V_{DD}$  при различных температурах**Рис. 16-14** График зависимости типового, минимального и максимального периода WDT от  $V_{DD}$  при различных температурах

**Рис. 16-15** График зависимости среднего периода WDT от  $V_{DD}$  при различных температурах (от  $-40^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ )

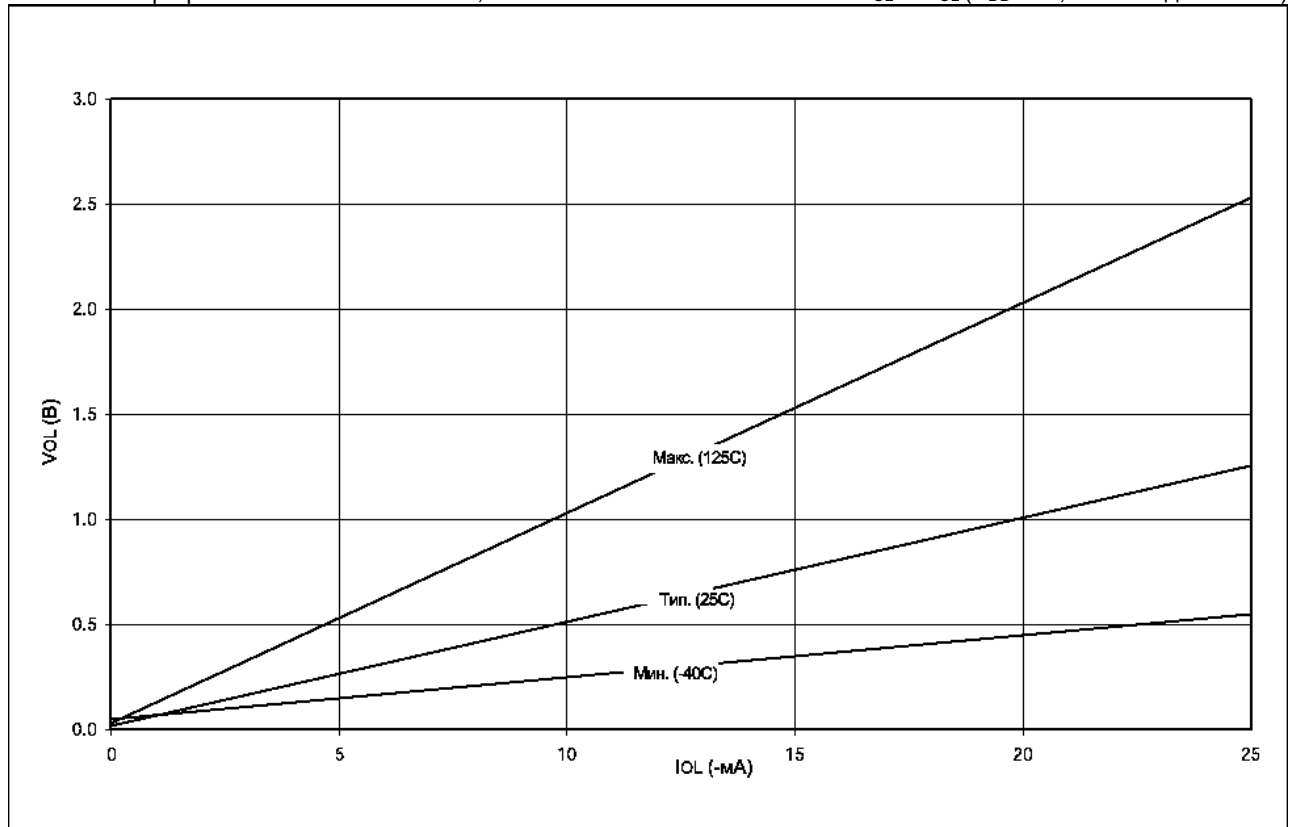


**Рис. 16-16** График зависимости типового, минимального и максимального  $V_{OH}$  от  $I_{OH}$  ( $V_{DD}=5\text{В}$ , от  $-40^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ )

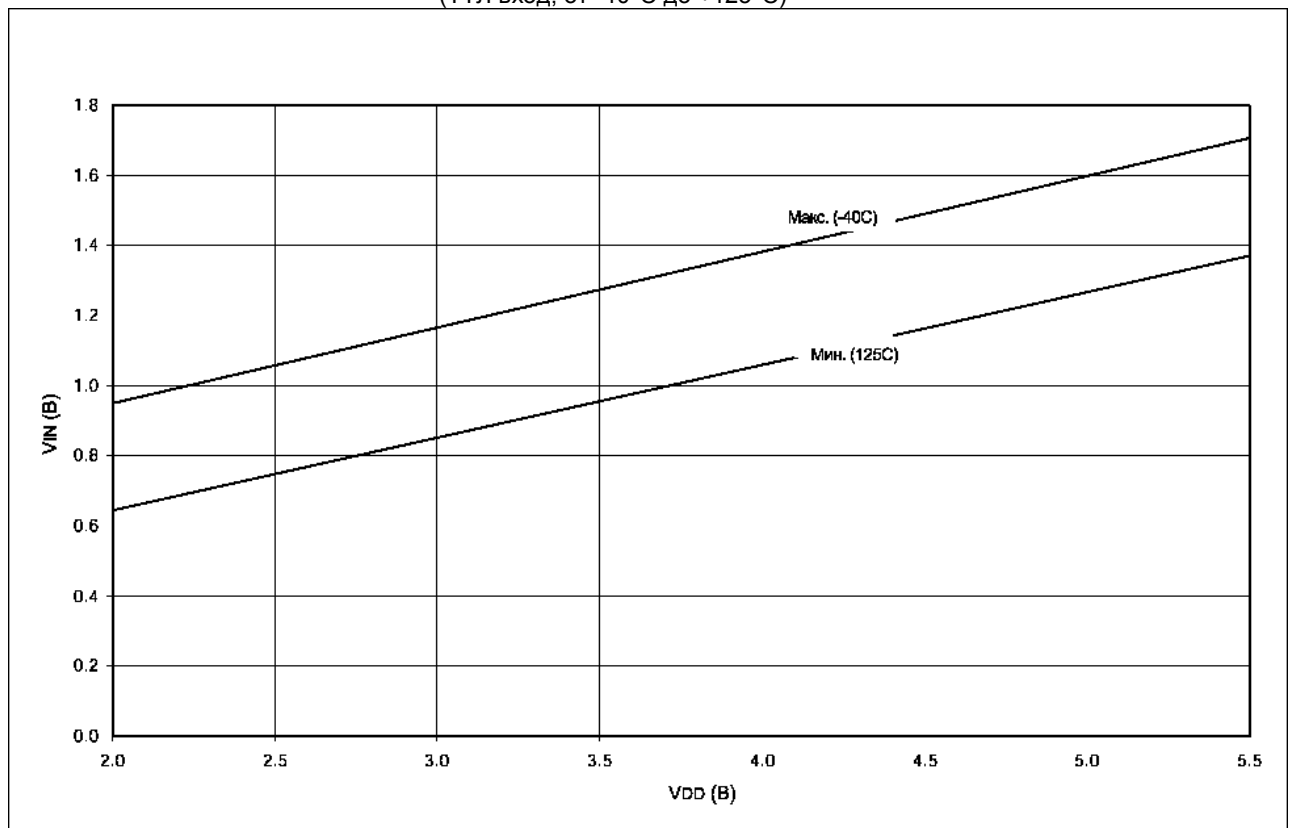


**Рис. 16-17** График зависимости типового, минимального и максимального  $V_{OH}$  от  $I_{OH}$  ( $V_{DD}=3В$ , от  $-40^{\circ}C$  до  $+125^{\circ}C$ )**Рис. 16-18** График зависимости типового, минимального и максимального  $V_{OL}$  от  $I_{OL}$  ( $V_{DD}=5В$ , от  $-40^{\circ}C$  до  $+125^{\circ}C$ )

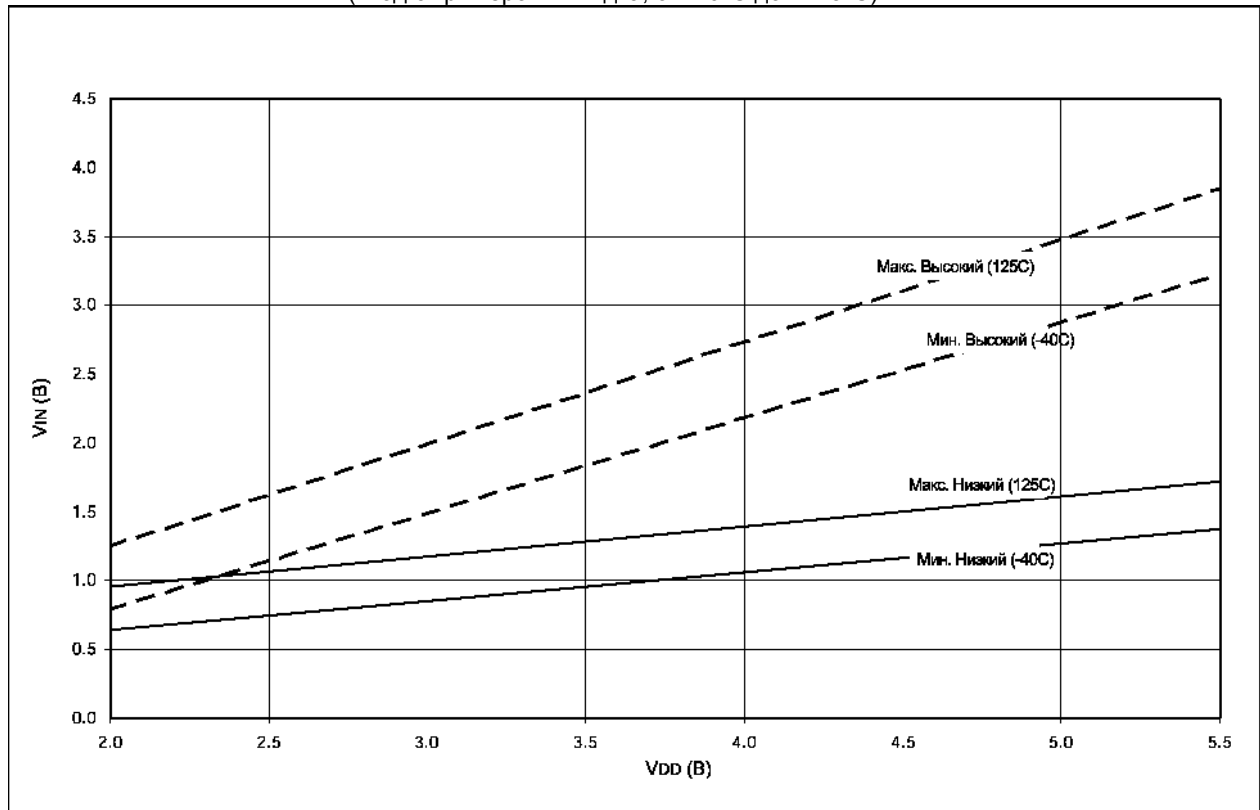
**Рис. 16-19** График зависимости типового, минимального и максимального  $V_{OL}$  от  $I_{OL}$  ( $V_{DD}=3В$ , от  $-40^{\circ}C$  до  $+125^{\circ}C$ )



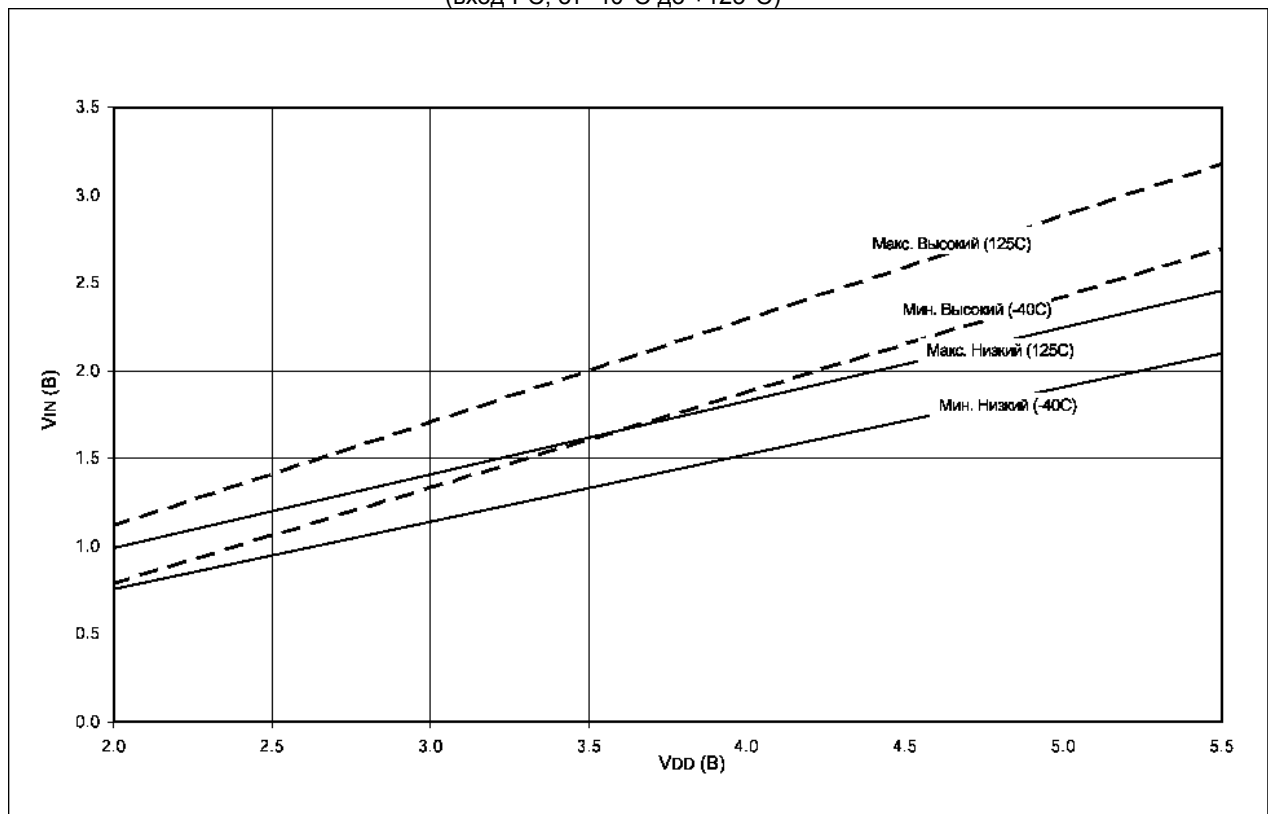
**Рис. 16-20** График зависимости минимального и максимального  $V_{IN}$  от  $V_{DD}$  (ТТЛ вход, от  $-40^{\circ}C$  до  $+125^{\circ}C$ )



**Рис. 16-21** График зависимости минимального и максимального  $V_{IN}$  от  $V_{DD}$   
(вход с триггером Шмидта, от  $-40^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ )



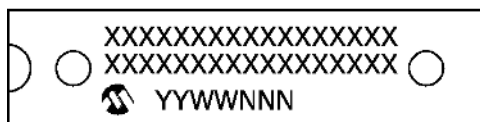
**Рис. 16-22** График зависимости минимального и максимального  $V_{IN}$  от  $V_{DD}$   
(вход  $I^2C$ , от  $-40^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ )



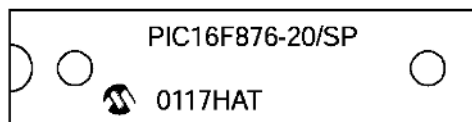
## 17.0 Корпуса микроконтроллеров

### 17.1 Описание обозначений на корпусах микроконтроллеров

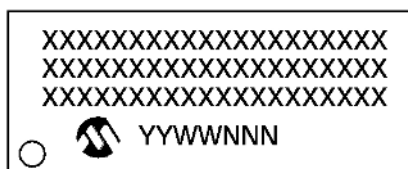
#### 28 - выводный PDIP



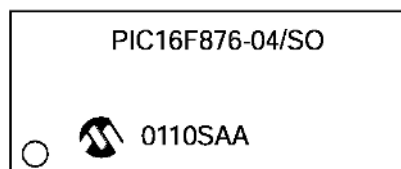
#### Пример



#### 28 - выводный SOIC



#### Пример



Обозначения:

XX..X	Тип микроконтроллера*
YY	Две цифры года изготовления
WW	Две цифры номера недели изготовления считая с 1 января.
NNN	Алфавитно-цифровой код

**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.

**Описание обозначений на корпусах микроконтроллеров (продолжение)**

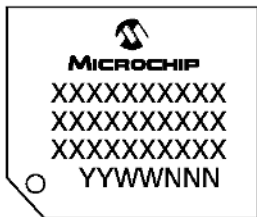
**40 - выводный PDIP**



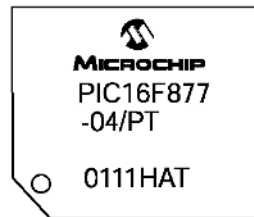
**Пример**



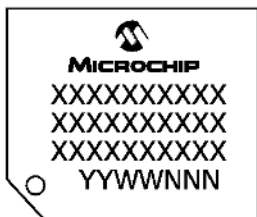
**44 - выводный TQFP**



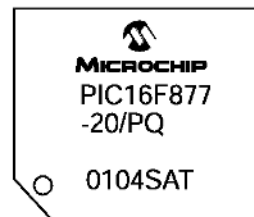
**Пример**



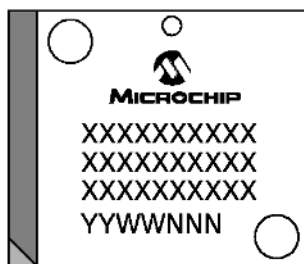
**44 - выводный MQFP**



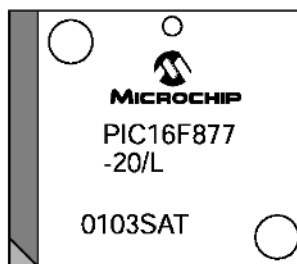
**Пример**



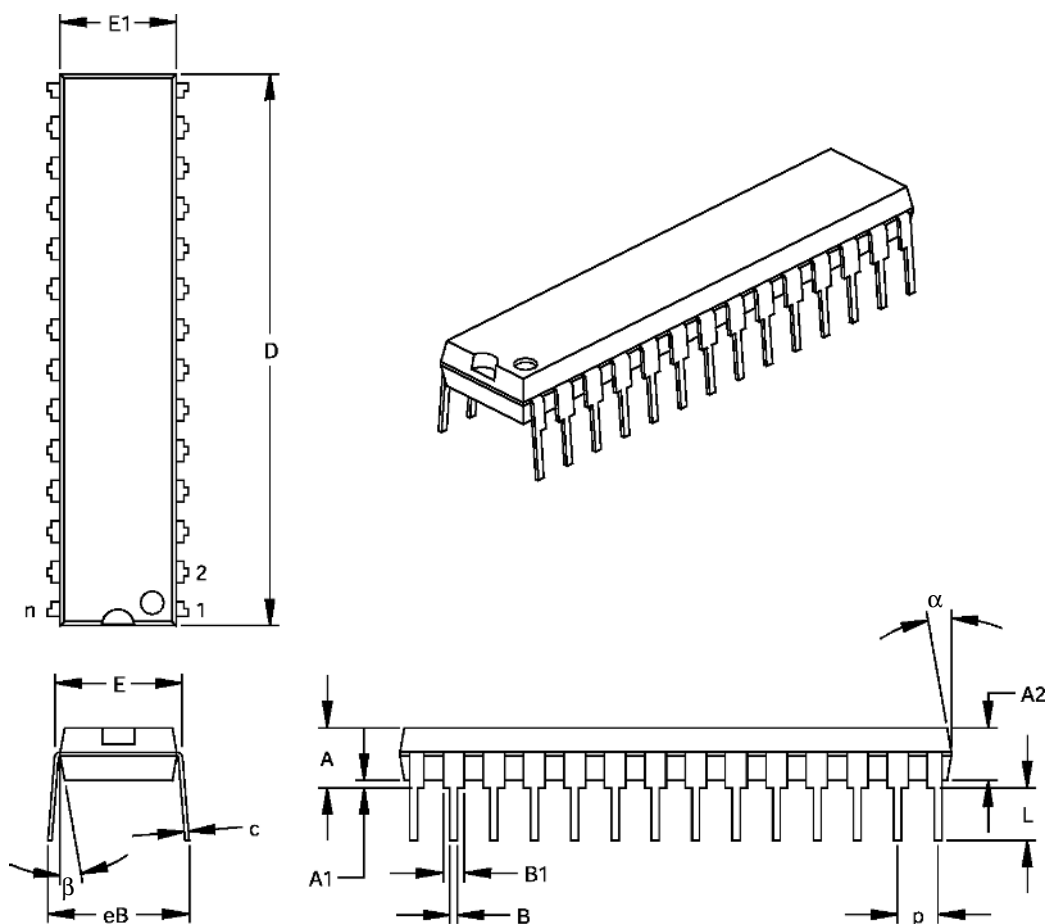
**44 - выводный PLCC**



**Пример**





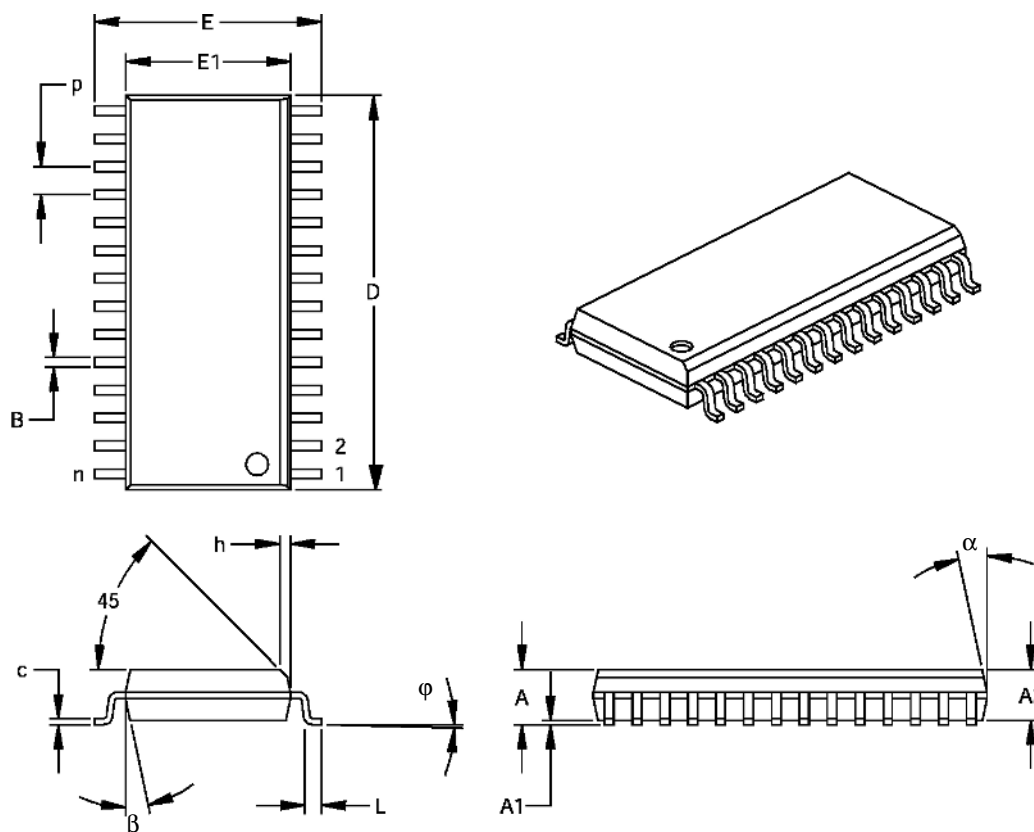
**Тип корпуса: 28-выводный PDIP - 300mil**

Единицы измерения		Дюймы*			Миллиметры		
Пределы размеров		Мин.	Ном.	Макс.	Мин.	Ном.	Макс.
Число выводов	n		28			28	
Расстояние между выводами	p		0.100			2.54	
Высота корпуса	A	0.140	0.150	0.160	3.56	3.81	4.06
Толщина корпуса	A2	0.125	0.130	0.135	3.18	3.30	3.43
Расстояние между корпусом и платой	A1	0.015			0.38		
Ширина корпуса с выводами	E	0.300	0.310	0.352	7.62	7.87	8.26
Ширина корпуса	E1	0.275	0.285	0.295	6.99	7.24	7.49
Длина корпуса	D	1.345	1.365	1.385	34.16	34.67	35.18
Длина нижней части вывода	L	0.125	0.130	0.135	3.18	3.30	3.43
Толщина вывода	c	0.008	0.012	0.015	0.20	0.29	0.38
Ширина верхней части вывода	B1	0.040	0.053	0.065	1.02	1.33	1.65
Ширина нижней части вывода	B	0.016	0.019	0.022	0.41	0.48	0.56
Полная ширина корпуса с выводами	eB	0.320	0.350	0.430	8.13	8.89	10.92
Угол фаски верхней части корпуса	alpha	5	10	15	5	10	15
Угол фаски нижней части корпуса	beta	5	10	15	5	10	15

\* Основные размеры.

Эквивалент JEDEC: MO-095

Примечание. Параметры D и E1 не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм).

**Тип корпуса: 28-выводный SOIC - 300mil**

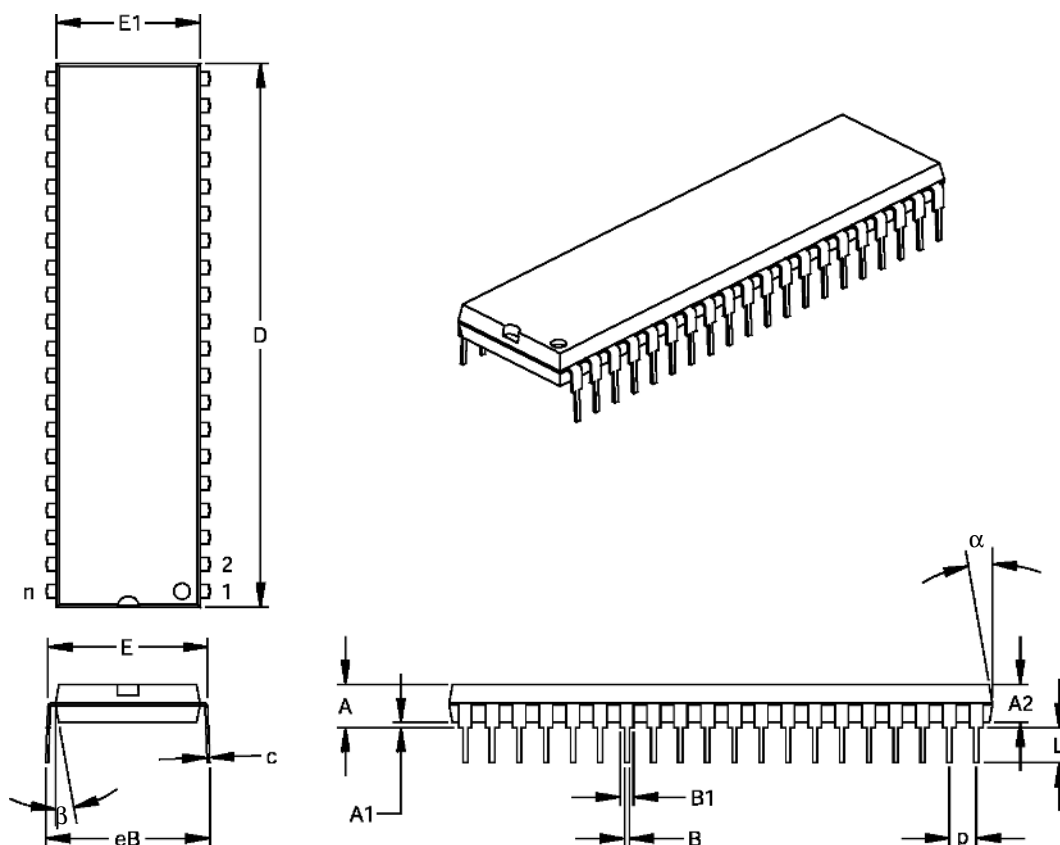
Единицы измерения		Дюймы*			Миллиметры		
Пределы размеров		Мин.	Ном.	Макс.	Мин.	Ном.	Макс.
Число выводов	n		28			28	
Расстояние между выводами	p		0.050			1.27	
Толщина корпуса с выводами	A	0.093	0.099	0.104	2.36	2.50	2.64
Толщина корпуса	A2	0.088	0.091	0.094	2.24	2.31	2.39
Расстояние между корпусом и платой	A1	0.004	0.008	0.012	0.10	0.20	0.30
Ширина корпуса с выводами	E	0.394	0.407	0.420	10.01	10.34	10.67
Ширина корпуса	E1	0.288	0.295	0.299	7.32	7.49	7.59
Длина корпуса	D	0.695	0.704	0.712	17.65	17.87	18.08
Размер ориентирующей фаски	h	0.010	0.020	0.029	0.25	0.50	0.74
Длина нижней части вывода	L	0.016	0.033	0.050	0.41	0.84	1.27
Угол наклона нижней части вывода	φ	0	4	8	0	4	8
Толщина вывода	c	0.009	0.011	0.013	0.23	0.28	0.33
Ширина вывода	B	0.014	0.017	0.20	0.36	0.42	0.51
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.

Эквивалент JEDEC: MS-013

Примечание. Параметры D и E1 не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм).

## Тип корпуса: 40-выводный PDIP - 600mil



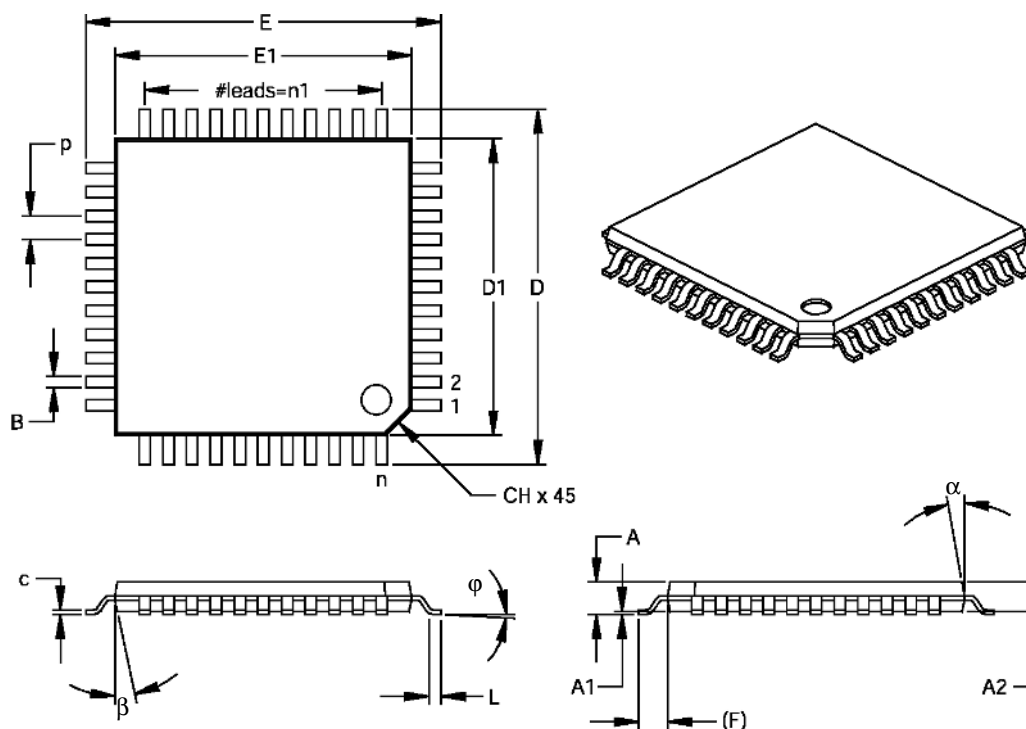
Единицы измерения		Дюймы*			Миллиметры		
Пределы размеров		Мин.	Ном.	Макс.	Мин.	Ном.	Макс.
Число выводов	n		40			40	
Расстояние между выводами	p		0.100			2.54	
Высота корпуса	A	0.160	0.175	0.190	4.06	4.45	4.83
Толщина корпуса	A2	0.140	0.150	0.160	3.56	3.81	4.06
Расстояние между корпусом и платой	A1	0.015			0.38		
Ширина корпуса с выводами	E	0.595	0.600	0.625	15.11	15.24	15.88
Ширина корпуса	E1	0.530	0.545	0.560	13.46	13.84	14.22
Длина корпуса	D	2.045	2.058	2.065	51.94	52.26	52.45
Длина нижней части вывода	L	0.120	0.130	0.135	3.05	3.30	3.43
Толщина вывода	c	0.008	0.012	0.015	0.20	0.29	0.38
Ширина верхней части вывода	B1	0.030	0.050	0.070	0.76	1.27	1.78
Ширина нижней части вывода	B	0.014	0.018	0.022	0.36	0.46	0.56
Полная ширина корпуса с выводами	eB	0.620	0.650	0.680	15.75	16.51	17.27
Угол фаски верхней части корпуса	alpha	5	10	15	5	10	15
Угол фаски нижней части корпуса	beta	5	10	15	5	10	15

\* Основные размеры.

Эквивалент JEDEC: MO-011

Примечание. Параметры D и E1 не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм).

## Тип корпуса: 44-выводный TQFP



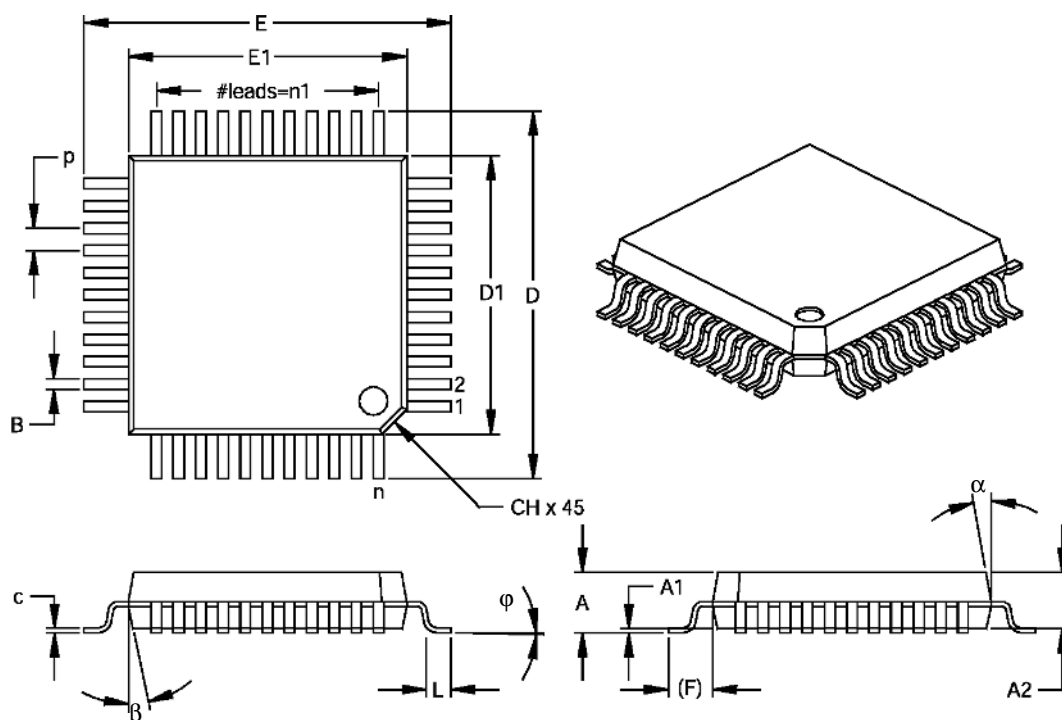
Единицы измерения		Дюймы			Миллиметры*		
Пределы размеров		Мин.	Ном.	Макс.	Мин.	Ном.	Макс.
Число выводов	n		44			44	
Расстояние между выводами	p		0.031			0.80	
Число выводов с одной стороны	n1		11			11	
Толщина корпуса с выводами	A	0.039	0.043	0.047	1.00	1.10	1.20
Толщина корпуса	A2	0.037	0.039	0.041	0.95	1.00	1.05
Расстояние между корпусом и платой	A1	0.002	0.004	0.006	0.05	0.10	0.15
Длина нижней части вывода	L	0.018	0.024	0.030	0.45	0.60	0.75
Длина вывода	(F)		0.039		1.00		
Угол наклона нижней части вывода	φ	0	3.5	7	0	3.5	7
Ширина корпуса с выводами	E	0.463	0.472	0.482	11.75	12.00	12.25
Длина корпуса с выводами	D	0.463	0.472	0.482	11.75	12.00	12.25
Ширина корпуса	E1	0.390	0.394	0.398	9.90	10.00	10.10
Длина корпуса	D1	0.390	0.394	0.398	9.90	10.00	10.10
Толщина вывода	c	0.004	0.006	0.008	0.09	0.15	0.20
Ширина вывода	B	0.012	0.015	0.017	0.30	0.38	0.44
Размер ориентирующей фаски	CH	0.025	0.035	0.045	0.64	0.89	1.14
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.

Эквивалент JEDEC: MS-026

Примечание. Параметры D и E1 не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм).

## Тип корпуса: 44-выводный MQFP



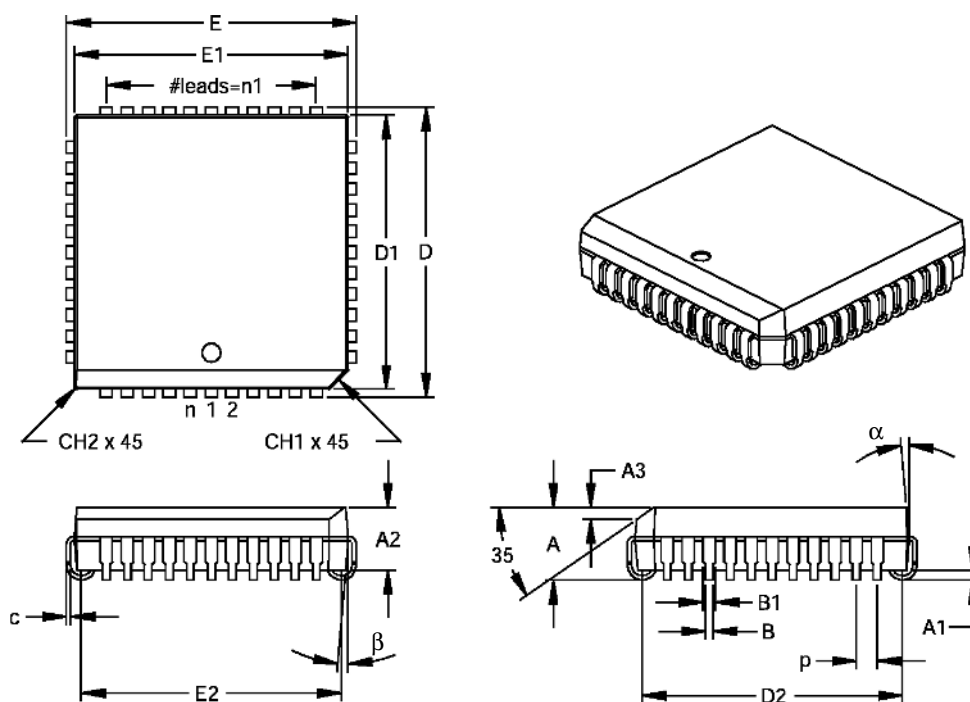
Единицы измерения		Дюймы			Миллиметры*		
Пределы размеров		Мин.	Ном.	Макс.	Мин.	Ном.	Макс.
Число выводов	n		44			44	
Расстояние между выводами	p		0.031			0.80	
Число выводов с одной стороны	n1		11			11	
Толщина корпуса с выводами	A	0.079	0.086	0.093	2.00	2.18	2.35
Толщина корпуса	A2	0.077	0.080	0.083	1.95	2.03	2.10
Расстояние между корпусом и платой	A1	0.002	0.006	0.010	0.05	0.15	0.25
Длина нижней части вывода	L	0.029	0.035	0.041	0.73	0.88	1.03
Длина вывода	(F)		0.063			1.6	
Угол наклона нижней части вывода	φ	0	3.5	7	0	3.5	7
Ширина корпуса с выводами	E	0.510	0.520	0.530	12.95	13.20	13.45
Длина корпуса с выводами	D	0.510	0.520	0.530	12.95	13.20	13.45
Ширина корпуса	E1	0.390	0.394	0.398	9.90	10.00	10.10
Длина корпуса	D1	0.390	0.394	0.398	9.90	10.00	10.10
Толщина вывода	c	0.005	0.007	0.009	0.13	0.18	0.23
Ширина вывода	B	0.012	0.015	0.018	0.30	0.38	0.45
Размер ориентирующей фаски	CH	0.025	0.035	0.045	0.64	0.89	1.14
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.

Эквивалент JEDEC: MS-022

Примечание. Параметры D и E1 не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм).

## Тип корпуса: 44-выводный PLCC



Единицы измерения Пределы размеров		Дюймы*			Миллиметры		
		Мин.	Ном.	Макс.	Мин.	Ном.	Макс.
Число выводов	n		44			44	
Расстояние между выводами	p		0.050			1.27	
Число выводов с одной стороны	n1		11			11	
Толщина корпуса с выводами	A	0.165	0.173	0.180	4.19	4.39	4.57
Толщина корпуса	A2	0.145	0.153	0.160	3.68	3.87	4.06
Расстояние между корпусом и платой	A1	0.020	0.028	0.035	0.51	0.71	0.89
Высота ориентирующей фаски	A3	0.024	0.029	0.34	0.61	0.74	0.86
Размер ориентирующей фаски	CH1	0.040	0.045	0.050	1.02	1.14	1.27
Размер ориентирующей фаски	CH2	0.000	0.005	0.010	0.00	0.13	0.25
Ширина корпуса с выводами	E	0.685	0.690	0.695	17.40	17.53	17.65
Длина корпуса с выводами	D	0.685	0.690	0.695	17.40	17.53	17.65
Ширина корпуса	E1	0.650	0.653	0.656	16.51	16.59	16.66
Длина корпуса	D1	0.650	0.653	0.656	16.51	16.59	16.66
Расстояние между выводами	E2	0.590	0.620	0.630	14.99	15.75	16.00
Расстояние между выводами	D2	0.590	0.620	0.630	14.99	15.75	16.00
Толщина вывода	c	0.008	0.011	0.013	0.20	0.27	0.33
Ширина верхней части вывода	B1	0.026	0.029	0.032	0.66	0.74	0.81
Ширина нижней части вывода	B	0.013	0.020	0.021	0.33	0.51	0.53
Угол фаски верхней части корпуса	alpha	0	5	10	0	5	10
Угол фаски нижней части корпуса	beta	0	5	10	0	5	10

\* Основные размеры.

Эквивалент JEDEC: MO-047

Примечание. Параметры D и E1 не включают выступы. Выступы в сторону не должны превышать 0.010"(0.254мм).

## 17.2 Правила идентификации типа микроконтроллеров PIC16F87X

Чтобы определить параметры микроконтроллеров воспользуйтесь ниже описанным правилом.

<u>PART№</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Микроконтроллер	Температурный диапазон	Корпус	Образец
<p>Микроконтроллер  PIC16F87X<sup>(1)</sup>, PIC16F87XT<sup>(2)</sup>, <math>4.0V \leq V_{DD} \leq 5.5V</math>  PIC16LF87X<sup>(1)</sup>, PIC16LF87XT<sup>(2)</sup>, <math>2.0V \leq V_{DD} \leq 5.5V</math></p> <p>Максимальная тактовая частота  04 = 4МГц  10 = 10МГц  20 = 20МГц</p> <p>Температурный диапазон  - = от 0°C до +70°C  I = от -40°C до +85°C  E = от -40°C до +125°C</p> <p>Корпус  PQ = MQFP  PT = TQFP  SO = SOIC  SP = тонкий PDIP  P = PDIP  L = PLCC</p>			
<p><b>Пример</b></p> <ol style="list-style-type: none"> <li><b>PIC16F877-20/P 301</b> = коммерческий температурный диапазон, корпус PDIP, 20МГц, нормальный диапазон напряжения питания, код QTP 301.</li> <li><b>PIC16LF876-04I/SO</b> = промышленный температурный диапазон, корпус SOIC, 4МГц, расширенный диапазон напряжения питания.</li> <li><b>PIC16F877-10E/P</b> = расширенный температурный диапазон, корпус PDIP, 10МГц, нормальный диапазон напряжения питания.</li> </ol> <p><b>Примечания:</b></p> <ol style="list-style-type: none"> <li>F = CMOS FLASH; LF = CMOS FLASH с расширенным напряжением питания.</li> <li>T = для работы в условиях вибрации, только корпуса SOIC, PLCC, MQFP, TQFP.</li> </ol>			

## 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](mailto: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](mailto:sales@microchip.ru)

На сайте

[www.microchip.ru](http://www.microchip.ru)

Вы можете узнать последние новости нашей фирмы, найти техническую документацию и информацию по наличию комплектующих на складе.