

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»**

МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА PIC16F87X

САМАРА 2007

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА PIC16F87X

*Утверждено редакционно-издательским советом
в качестве методических указаний к лабораторным работам*

САМАРА
Издательство СГАУ
2007

УДК 004.382.7(075)

Составитель ***И.В. Лофицкий***

Рецензент В. В. И в а н о в

МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА PIC16F87X: метод. указания к лаб. работам. / Сост. *И.В. Лофицкий*. – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2007. 44 с.

Рассмотрены вопросы организации архитектуры, системы команд микроконтроллеров PIC16F87X и отладочные средства фирмы MICROCHIP.

Предназначены для студентов специальности 210201 «Проектирование и технология РЭС».

Подготовлены на кафедре микроэлектроники.

ЛАБОРАТОРНАЯ РАБОТА №1

АРХИТЕКТУРА И СИСТЕМА КОМАНД МИКРО- КОНТРОЛЛЕРОВ PIC16F87X И ОТЛАДОЧНЫЕ СРЕДСТВА ФИРМЫ MICROCHIP

Цель работы: Изучение архитектуры и системы команд восьмиразрядных однокристальных микроконтроллеров фирмы **MICROCHIP** семейства **PIC16F87X**, знакомство с отладочной средой **MPLAB** и отладочным модулем **MPLAB-ICD Demo Board**.

1. АРХИТЕКТУРА И СИСТЕМА КОМАНД МИКРО- КОНТРОЛЛЕРОВ PIC16F87X

Основные особенности архитектуры высокопроизводительных восьмиразрядных однокристальных микроконтроллеров **PIC16F87X**:

АЛУ:

- высокоскоростная **RISC** архитектура арифметико-логического устройства;
- 35 инструкций;
- прямой, косвенный и относительный способы адресации;
- все команды выполняются за один командный цикл (4 такта) кроме инструкций переходов, выполняемых за два цикла;
- тактовая частота 0 – 20 МГц.

ПАМЯТЬ:

- 8192 14-разрядных слова памяти программ, выполненной по технологии **FLASH** (электрически стираемой ПЗУ);
- 368 8-разрядных байт памяти данных (ОЗУ);
- 256 8-разрядных байт памяти данных (электрически стираемое ПЗУ).

ПЕРИФЕРИЙНЫЕ МОДУЛИ:

- система прерываний (до 14 источников);
- 8-уровневый аппаратный стек;

- сторожевой таймер **WDT** с собственным RC генератором;
- режим энергосбережения;
- программный выбор параметров тактового генератора;
- 33 линии цифрового ввода-вывода (порты **A** – 6 разрядов; **B**, **C**, **D** – 8 разрядов; **E** – 3 разряда);
- повышенная нагрузочная способность портов ввода-вывода (25 мА);
- таймер 0: 8-разрядный таймер/счетчик с 8-разрядным предделителем;
- таймер 1: 16-разрядный таймер/счетчик с возможностью подключения внешнего резонатора;
- таймер 2: 8-разрядный таймер/счетчик с 8-разрядным предделителем и выходным делителем;
- два модуля сравнения/захват/ШИМ (**ССР**):
 - 16-разрядный захват (максимальная разрешающая способность 12.5 нс);
 - 16-разрядное сравнение (максимальная разрешающая способность 200 нс);
 - 10-разрядный ШИМ;
- многоканальный 10-разрядный АЦП;
- последовательный синхронный порт **MSSP**:
 - ведущий/ведомый режим **SPI**;
 - ведущий/ведомый режим **I²C**;
- последовательный синхронно-асинхронный приемопередатчик **USART** с поддержкой детектирования адреса;
- ведомый 8-разрядный параллельный порт (**PDP**) с поддержкой внешних сигналов **-RD**, **-WR**, **-CS**;
- детектор пониженного напряжения питания (**BOD**).

Структурная схема микроконтроллера **PIC16F788** приведена на рис.1.

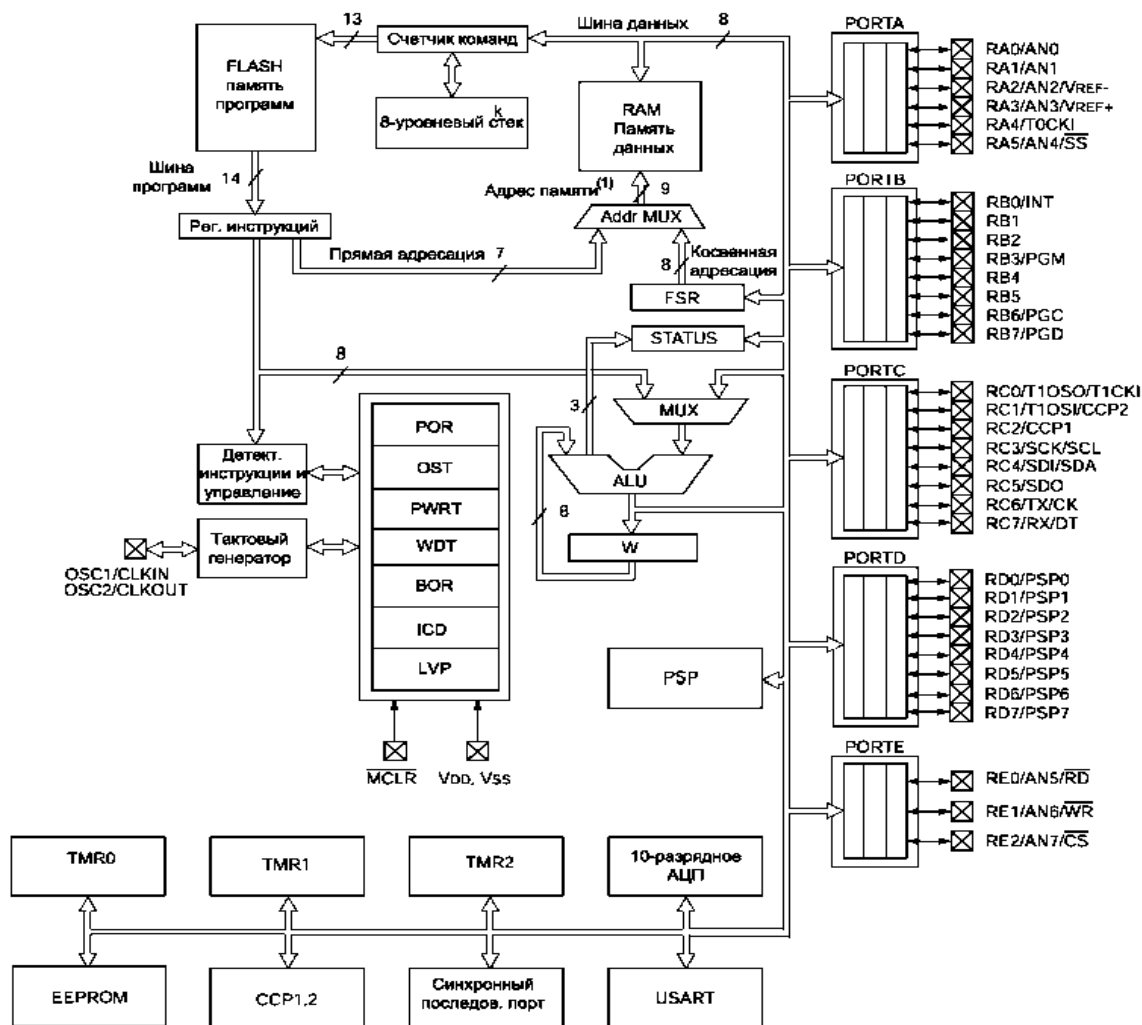


Рис. 1. Структурная схема микроконтроллера PIC16F788

1.1. Организация памяти программ

Семейство PIC16F7X имеет 13-разрядный счетчик команд, с помощью которого адресуется пространство $8K \times 14$ памяти программ. После начальной установки программный счетчик устанавливается на 0000h, а любое прерывание вызывает переход на адрес 0004h. Инструкции переходов (CALL и GOTO) имеют 11-разрядное поле для указания адреса, что позволяет непосредственно адресовать 2K слов памяти программ. Для адресации верхних страниц памяти программ используются 2 бита регистра PCLATH<4:3>. Перед выполнением команды перехода (CALL или GOTO) необходимо занести в соответствующие биты регистра PCLATH адрес требуемой страницы. При выполнении инструкций возврата из подпрограммы 13-разрядное значение для счетчика программ PC считывается из

вершины стека, поэтому манипуляция битами регистра PCLATH<4:3> не требуется. Организацию памяти программ поясняет рис.2.

Регистр косвенной адресации	00h	Регистр косвенной адресации	80h	Регистр косвенной адресации	100h	Регистр косвенной адресации	180h			
TMR0	01h	OPTION REG	81h	TMR0	101h	OPT CN 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 ¹⁾	08h	TRISD ¹⁾	88h		108h		188h			
PORTE ¹⁾	09h	TRISE ¹⁾	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	Резерв²⁾	18Eh			
TMR1H	0Fh		8Fh	EEADRH	10Fh	Резерв²⁾	18Fh			
T1CON	10h		90h		110h		190h			
TMR2	11h	SSPCON2	91h	Регистры общего назначения 16 байт	111h	Регистры общего назначения 16 байт	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		116h		196h			
CCP1CON	17h		97h		117h		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 96 байт	7Fh	Регистры общего назначения 80 байт	EFh F0h	Регистры общего назначения 80 байт	16Fh 170h	Регистры общего назначения 80 байт	1EFh 1F0h			
								Доступ к 70h-7Fh	Доступ к 70h-7Fh	Доступ к 70h-7Fh
Банк 0	Банк 1	Банк 2	Банк 3							

Рис.2. Карта памяти данных

1.2. Организация памяти данных

Память данных разделена на четыре банка, которые содержат регистры общего и специального (SFR) назначения. Биты RP1 и RP0 (регистр STATUS соответственно биты 6 и 5) предназначены для

управления банками данных. Ниже показано состояние управляющих битов при обращении к банкам памяти данных:

RP1:RP0	Банк
00	0
01	1
10	2
11	3

Объем одного банка памяти составляет 128 байт (7Fh). В начале каждого из банков размещены 32 8-разрядных регистра общего назначения. Остальные 96 регистров текущего банка используются пользователем как статическое ОЗУ данных. Некоторые, наиболее часто используемые регистры специальных функций, отображаются в банках памяти данных. Карта памяти данных применительно к **РІС16F788** приведена на рис. 2.

1.3. Регистры специальных функций

С помощью регистров специальных функций выполняется управление функциями арифметико-логического устройства и периферийными устройствами. Эти регистры реализованы как статическое ОЗУ. Основные сведения – адрес, уникальные имена регистров и отдельных битов, а также состояние после сброса приведены в приложении 2.

1.4. Стек

Микроконтроллеры семейства РІС16F87Х имеют 8-уровневый 13-разрядный аппаратный стек (рис. 3). Стек не имеет отображения на память программ и память данных. Значение счетчика команд заносится в вершину стека при выполнении инструкции перехода или обработки прерывания. Обратное чтение из стека в счетчик команд происходит при возвращении из подпрограммы или процедуры обработки прерываний. Стек работает как циклический буфер. После 8 записей в стек девятая запись заменяет первую.

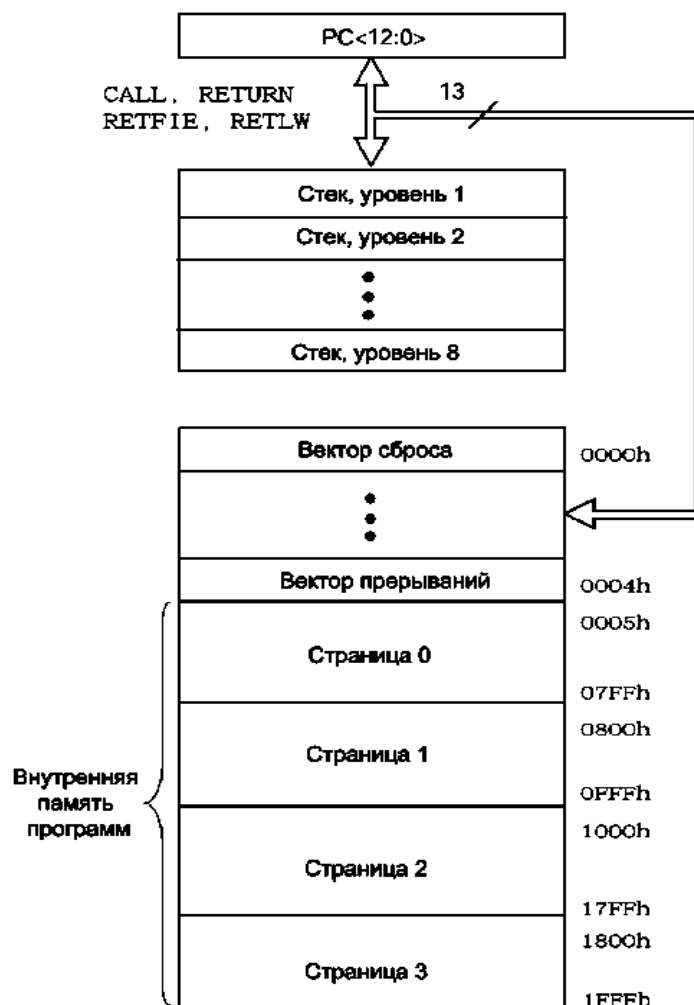


Рис.3. Аппаратный стек микроконтроллеров семейства PIC16F87X

2. СРЕДСТВА ОТЛАДКИ

Наиболее простым и функциональным инструментом для разработки программного обеспечения микроконтроллеров PICmicro является интегрированная среда разработки MPLAB-ICD фирмы MICROCHIP. MPLAB-ICD позволяет создавать/редактировать текст программ, преобразовывать текст программ в исполняемый код, а также осуществлять отладку и оптимизацию выполняемого кода. MPLAB-ICD включает в себя:

менеджер проекта MPLAB, используемый для создания и работы с файлами, относящимися к проекту;

редактор MPLAB, предназначенный для написания и редактирования исходного текста программы, шаблонов и файлов сценария линкера;

MPLAB-SIM симулятор, программно моделирующий выполнение программы в микроконтроллере;

MPLAB ICE эмулятор, обеспечивающий управляемую работу микроконтроллера в масштабе реального времени непосредственно в устройстве пользователя;

MPLAB ассемблер/компилятор/линкер/редактор библиотек, компилирующий (преобразующий в исполняемый код) текст программы, связывая различные модули и библиотеки;

программаторы *PRO MATE* и *PICSTART plus*, предназначенные для программирования встроенной FLASH-памяти программ микроконтроллеров.

2.1. Начало работы с MPLAB-ICD

Рассмотрим работу в среде на основе конкретного примера. Рабочий стол среды состоит из (рис. 4):

1. Главное текстовое меню.
2. Графическое меню.
3. Рабочая область.
4. Линейка состояния.

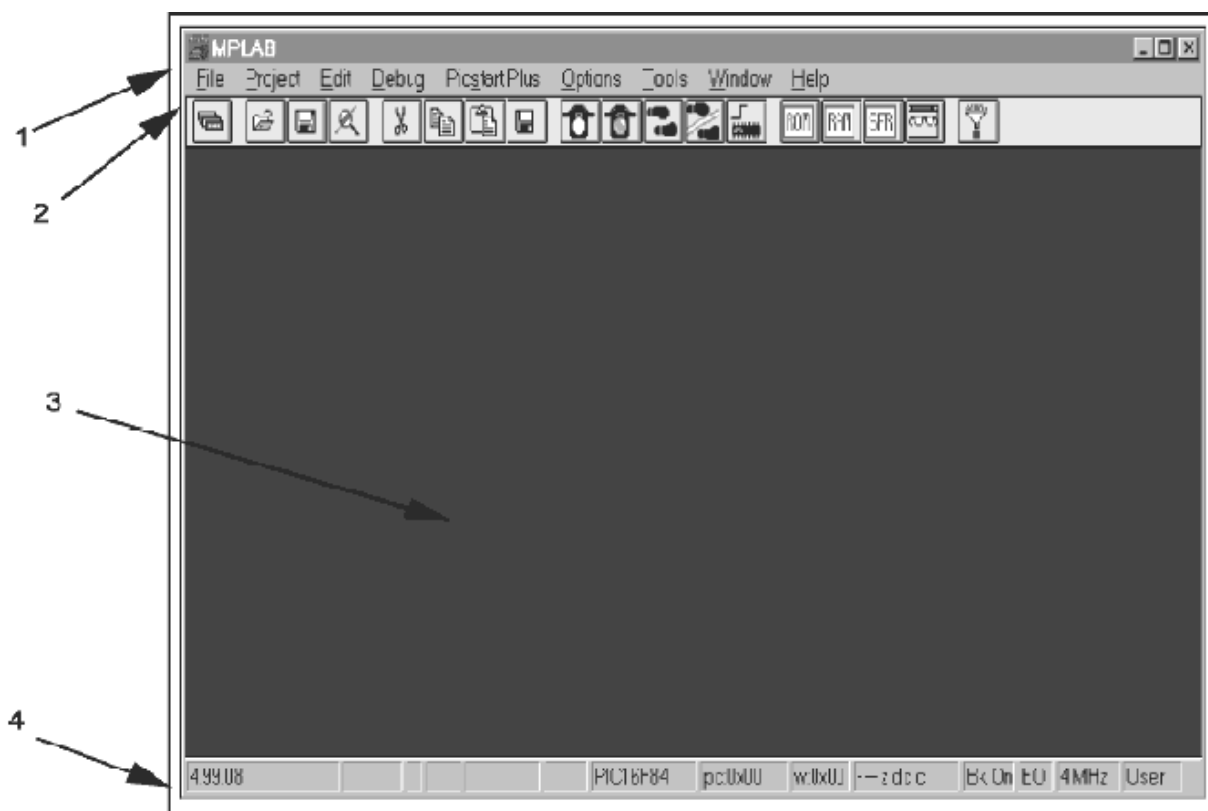


Рис. 4. Рабочий стол среды MPLAB-ICD

Выберите пункт **Options > Development Mode**, нажмите кнопку **Tools** для выбора инструментального средства и типа микроконтроллера, используемого в проекте (рис. 5).

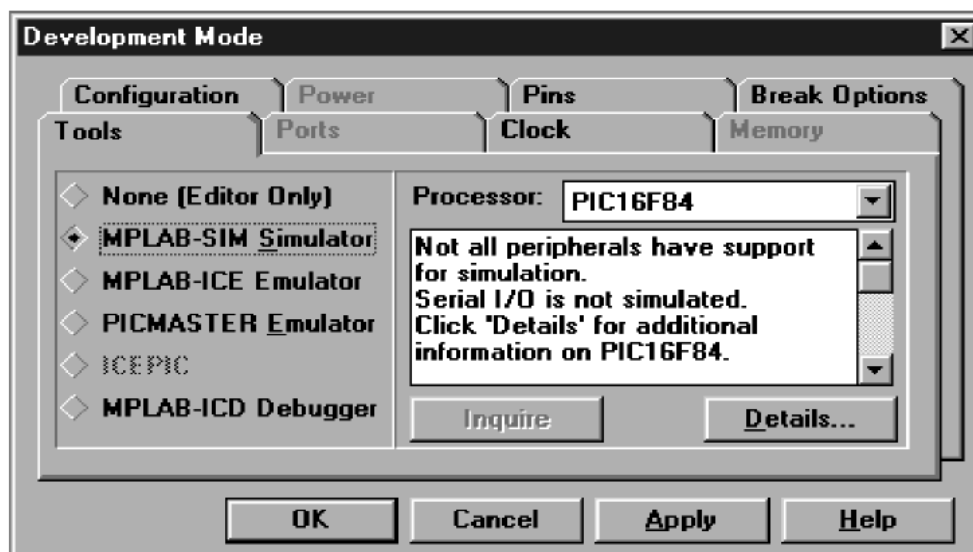


Рис.5. Окно выбора инструментального средства отладки

Выберите в окне **Tools** симулятор **MPLAB SIM** и микроконтроллер **PIC16F877**. Теперь, когда симулятор инициализирован, в линейке состояния включатся индикаторы “SIM” и “PIC16F877”.

2.2. Создание нового проекта

Симулятор работает с кодом программы (файл с именем, заданным пользователем, в дальнейшем – *<name>* и с расширением **.HEX**), который, в свою очередь, генерируется из текста исходного текста программы компилятором.

Выберите пункт меню **File > New**, и на экране появится сообщение (рис. 6).

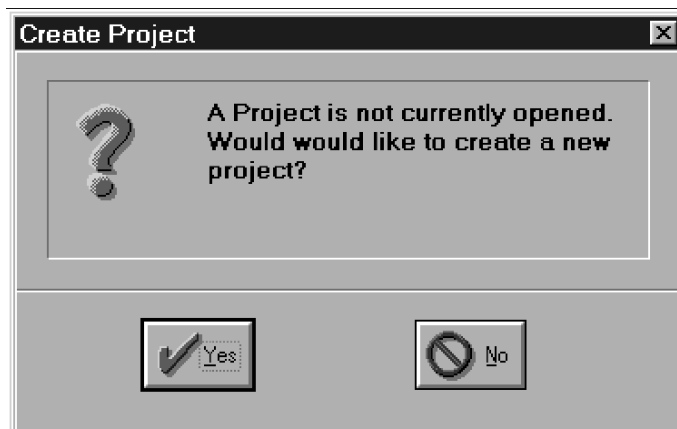


Рис. 6

Нажмите *Yes* и увидите следующее диалоговое окно (рис.7).

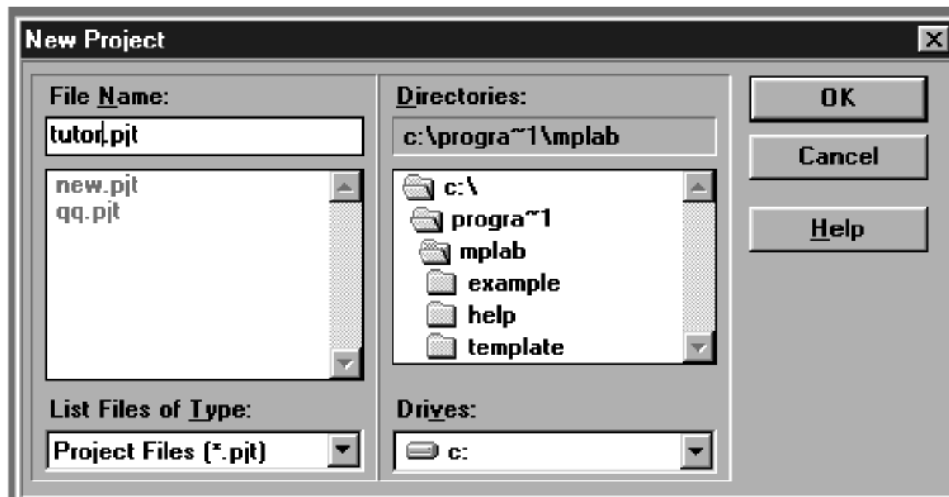


Рис. 7. Окно выбора файлов

В этом окне необходимо указать имя файла проекта и его местоположение. Подтверждение имени файла проекта приведет к следующему диалоговому окну (рис. 8).

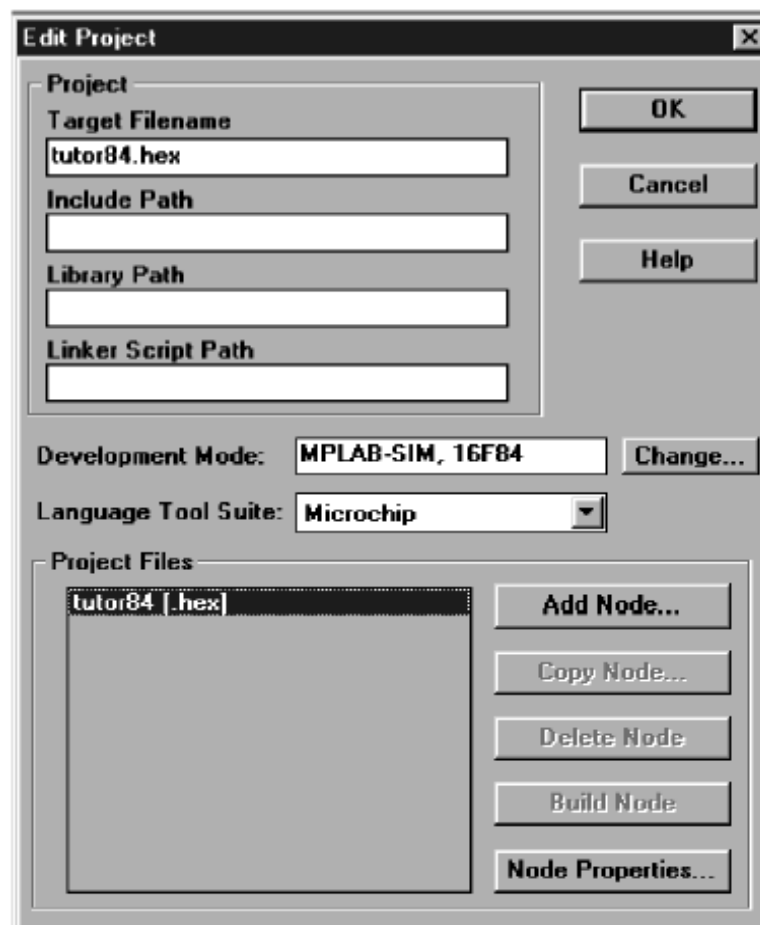


Рис.8. Окно редактирования параметров проекта

Нажав кнопку *Add Node*, вы увидите стандартный диалог выбора файлов (рис. 7), но уже применительно к файлу с расширением **.ASM**, содержащему текст программы на языке Ассемблера. Введите имя файла *<name>.asm* и нажмите **ОК**. Вы вернетесь к окну редактирования параметров проекта, где ниже файла кода *<name>.hex* должен появиться файл исходного текста программы *<name>.asm*. Нажатие кнопки **ОК** вернет Вас на рабочий стол среды **MPLAB-ICD** к открытому, но еще не названному файлу текста программы.

2.3. Создание нового исходного файла

Поставьте курсор в любое место открытого файла, выберите пункт меню **File > Save As...**, укажите имя и место расположения файла текста программы *<name>.asm* и нажмите **ОК**. Имя исходного файла должно совпадать с именем проекта.

2.4. Ввод исходного текста программы

Расположите курсор в начале файла *<name>.asm*. Введите следующий текст:

```
listp=16f877
include<p16f877.inc>
c1    equ    0x0c    ;адрес переменной c1 – 0x0c
      org    0x00    ;установить начало программы в век-
                    тор сброса 0x00

      nop
reset
      goto   start   ;переход на начало программы
      org    0x04    ;начало размещения программы в па-
                    мяти

start
      movlw  0x09    ;установить значение счетчика
      movwf  c1      ;записать его в регистр счетчика
loop
      incfsz c1, F   ;инкремент счетчика и пропуск
                    следующей команды
                    ;если результат равен нулю
```

goto	loop	;продолжить
goto	bug	;переинициализировать счет
end		

Обратите внимание на синтаксис текста и расположение меток, директив и комментариев. Подробное описание директив смотрите в документации *MPASM with MPLINK and MPLIB User's Guide*. В технической документации на микроконтроллер приведено детальное описание инструкций микроконтроллера.

Сохраните файл с помощью пункта меню *File > Save*.

2.5. Компиляция исходного файла

Компиляция (преобразование текста программы в машинный код) осуществляется после запуска компилятора с помощью меню *Project > Build All*. После компиляции на экране появится окно результатов. Так как в предпоследней строке текста приведенной выше программы была сделана ошибка, окно результатов будет содержать соответствующее сообщение о несуществующей метке “bug”. Двойное нажатие левой кнопки манипулятора перенесет курсор на строку в исходном тексте программы, где была допущена ошибка. После исправления ошибок и повторного запуска компилятора на экране появится окно результатов с сообщением об успешной компиляции “**Build completed successfully**”.

2.6. Запуск программы

Для инициализации симулятора выберите меню *Debug > Run > Reset*. Счетчик команд автоматически принимает значение 0x00. Затем с помощью меню *Debug > Run > Step* иницилируйте выполнение команд, помеченных темной полосой. Практически все пункты меню имеют горячие клавиши на клавиатуре и в графическом меню. Выбором пункта меню *Debug > Run > Run* или нажатием <F9> осуществите запуск программы с текущей позиции счетчика команд. Остановка программы произойдет после нажатия <F5> или *Debug > Run > Halt*, при этом указатель текущей инструкции примет первоначальный цвет, а информация в линейке состояния обновится.

2.7. Дополнительные окна

MPLAB-ICD поддерживает большое количество способов контроля выполнения программы как в режиме симулятора, так и при работе с аппаратными отладочными средствами. Для просмотра текущего значения регистров выберите меню **Window > File Registers**, что позволит отслеживать в отдельном окне состояние регистров микроконтроллера.

Для создания окна, отражающего отдельные (по выбору пользователя) регистры или биты, выберите пункт меню **Window > Watch Window > New Watch Window**. Если такое окно было создано ранее и сохранено на диске, выберите пункт меню **Window > Watch Window > Load Watch Window**. Укажите нужный файл, подтвердите выбор нажатием **ОК**. На экране появится диалоговое окно добавления переменных (рис. 9).

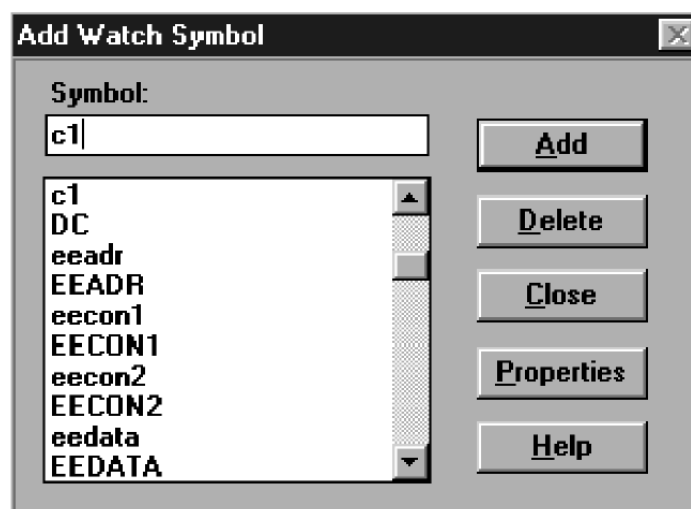


Рис. 9. Окно добавления переменных

Введите в поле **“Symbol:”** новую переменную **‘c1’** и нажмите **Add**. После выбора всех необходимых регистров аналогичным образом нажмите **Close**. На экране появится окно с выбранными регистрами, состояние которых будет изменяться в процессе работы симулятора.

Нажатием кнопки **<F5>** или **Debug > Run > Halt** остановите выполнение программы. При нажатии левой кнопки манипулятора на строке с текстом **‘movlw 0x09’** появится диалоговое окно (рис. 10).



Рис. 10. Меню определения точек остановки

Выберите пункт **Break Point(s)**, указательная линия изменит свой цвет, указывая на точку остановки. Нажмите кнопку <F6> или **Debug > Run > Reset** для сброса, а затем <F9> для начала выполнения программы. Выполнение программы будет прервано в точке остановки, при этом значение переменной 'c1' будет **0x00**. Только после выполнения той операции, которая была обозначена как точка остановки, значение переменной 'c1' изменится на 0x09.

3. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

- Ознакомиться с теоретической частью лабораторной работы.
- Запустить пакет MPLAB-ICD и, следуя рекомендациям п. 2, повторить все операции создания и отладки проекта.

ЛАБОРАТОРНАЯ РАБОТА №2

ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА В МИКРОКОНТРОЛЛЕРАХ СЕМЕЙСТВА PIC16F87X

Цель работы: Изучение работы портов параллельного ввода-вывода микроконтроллеров семейства **PIC16F87X**, закрепление навыков работы в отладочной среде **MPLAB** с отладочным модулем **MPLAB-ICD Demo Board**.

1. ПОРТЫ ВВОДА-ВЫВОДА МИКРОКОНТРОЛЛЕРОВ PIC16F87X

Семейство PIC16F87X включает в себя приборы PIC16F873, PIC16F874, PIC16F876 и PIC16F877, которые имеют ряд отличий, заключающихся в разном числе линий ввода-вывода. PIC16F874 и PIC16F877 имеют 33 линии ввода-вывода (6-разрядный PORTA, 8-разрядные PORTB, PORTC и PORTD и 3-разрядный PORTE), PIC16F873 управляет внешними устройствами по 22 линиям ввода-вывода (6-разрядный PORTA и 8 разрядные PORTB и PORTC). Основное назначение линий ввода-вывода – программное управление передачей данных, синхронизации процессов, а также опроса состояния внешних цепей и устройств.

Некоторые линии портов ввода-вывода мультиплицированы с периферийными модулями микроконтроллеров. При работе таких модулей соответствующие линии портов не могут использоваться как универсальные линии. Дополнительную информацию смотрите в технической документации DS33023 “PICmicro Mid-Range Reference Manual”.

1.1. Порт ввода-вывода PORTA

PORTA – 6-разрядный универсальный порт ввода-вывода, имеющий вспомогательный регистр TRISA, позволяющий настраивать отдельные линии как вход или выход. Запись логической ‘1’ в соответствующую (с тем же номером, что и отдельная линия) ячейку

этого регистра переводит соответствующий выходной буфер в 3-е (высокоимпедансное) состояние. Запись логического '0' в те же ячейки регистра TRISA определяет соответствующие линии порта как выходы. При этом содержимое регистра PORTA передается на соответствующие выводы микроконтроллера. Очевидно, что для корректного опроса состояния отдельной линии порта при вводе информации соответствующая ячейка регистра TRISA должна перевести буфер этой линии в 3-е состояние. Содержимое ячейки с тем же номером регистра PORTA в этом случае игнорируется. Важно помнить, что в микроконтроллерах семейства PIC16F87X линия RA4 имеет триггер Шмидта на входе и открытый сток на выходе. Эта линия мультиплицирована с тактовым входом таймера/счетчика TMR0. Все линии порта мультиплицированы с аналоговыми входами АЦП и аналоговым входом источника опорного напряжения. Структурная схема цепей, управляющих работой порта, приведена на рис. 1.

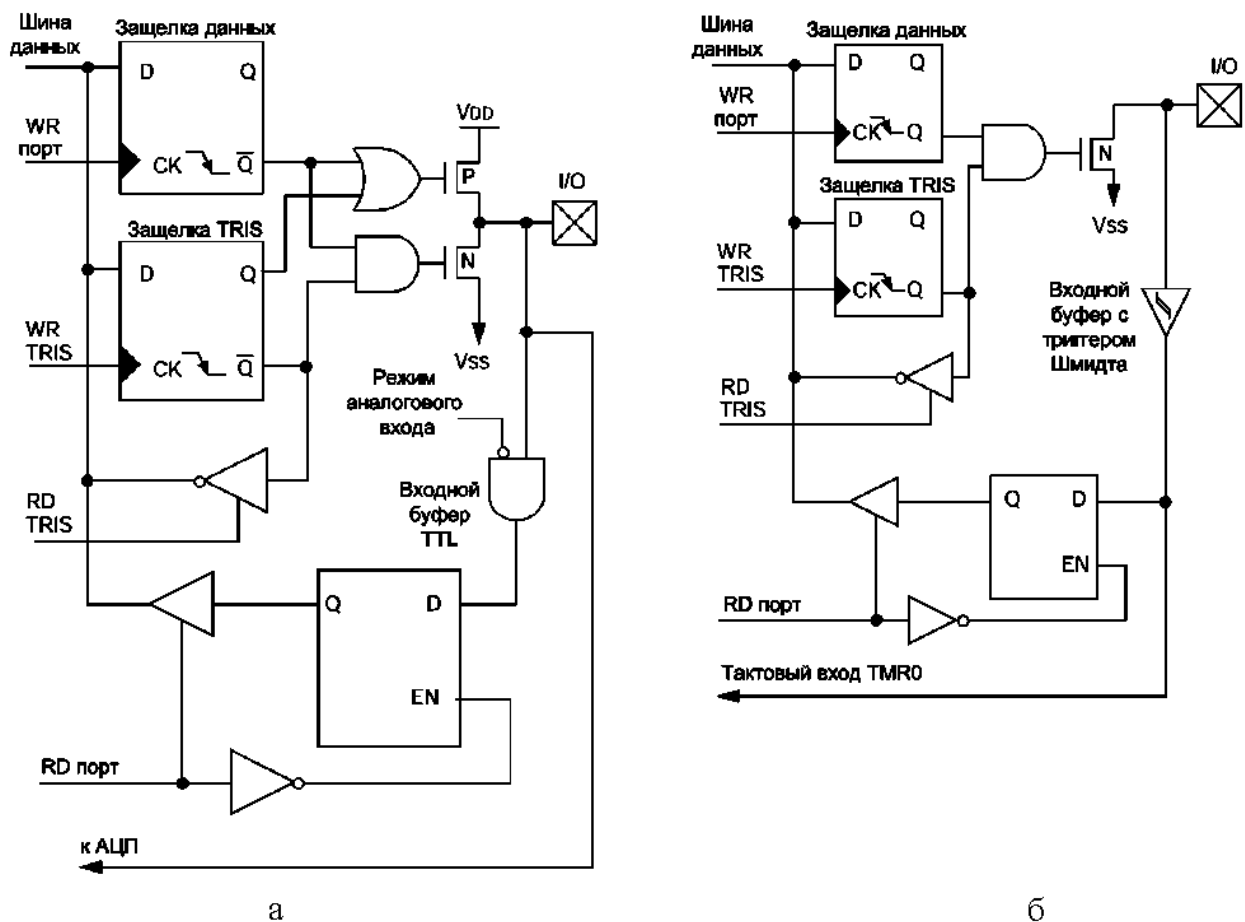


Рис.1. Структурная схема цепей управления линиями порта PORTA:
 а – линии RA<3:0> и RA<5>; б – линия RA<4>

После сброса по включению питания все выводы порта настраиваются как аналоговые и при опросе дают логический '0'. Следующий пример поясняет принцип настройки отдельных линий порта на ввод и вывод:

```

BCF      STATUS, RP1
BCF      STATUS, RP0      ;выбор банка 0
CLRF     PORTA            ;запись во все ячейки PORTA
                        ;логических '0'
BSF      STATUS, RP0      ;выбор банка 1
MOVLW    0x06             ;выключить компаратор
MOVWF    ADCON1           ;линии PORTA – цифровые входы/
                        ;выходы
MOVLW    0xCF
MOVWF    TRISA            ;RA<3:0> - цифровые входы
                        ;RA<5:4> - цифровые выходы
                        ;TRISA<7:6> - при опросе всегда '0'

```

1.2. Порт ввода-вывода PORTB

PORTB – 8-разрядный порт ввода-вывода, имеющий вспомогательный регистр TRISB, функционально аналогичный регистру TRISA, определяющий направление передачи данных. Структурная схема цепей управления линиями порта PORTB приведена на рис. 2, из которого видно, что помимо возможности перехода в 3-е состояние линии этого порта поддерживают внутрисхемную “подтяжку”, что позволяет реализовать простой интерфейс обслуживания клавиатуры. Три вывода порта RB3, RB6 и RB7 мультиплицированы со схемой низковольтного программирования, поэтому в отладочных модулях, предполагающих многократное перепрограммирование микроконтроллера, эти выводы не используются для ввода-вывода. Линии RB<7:4>, настроенные на вход, могут генерировать прерывания по признаку изменения логического уровня на входе.

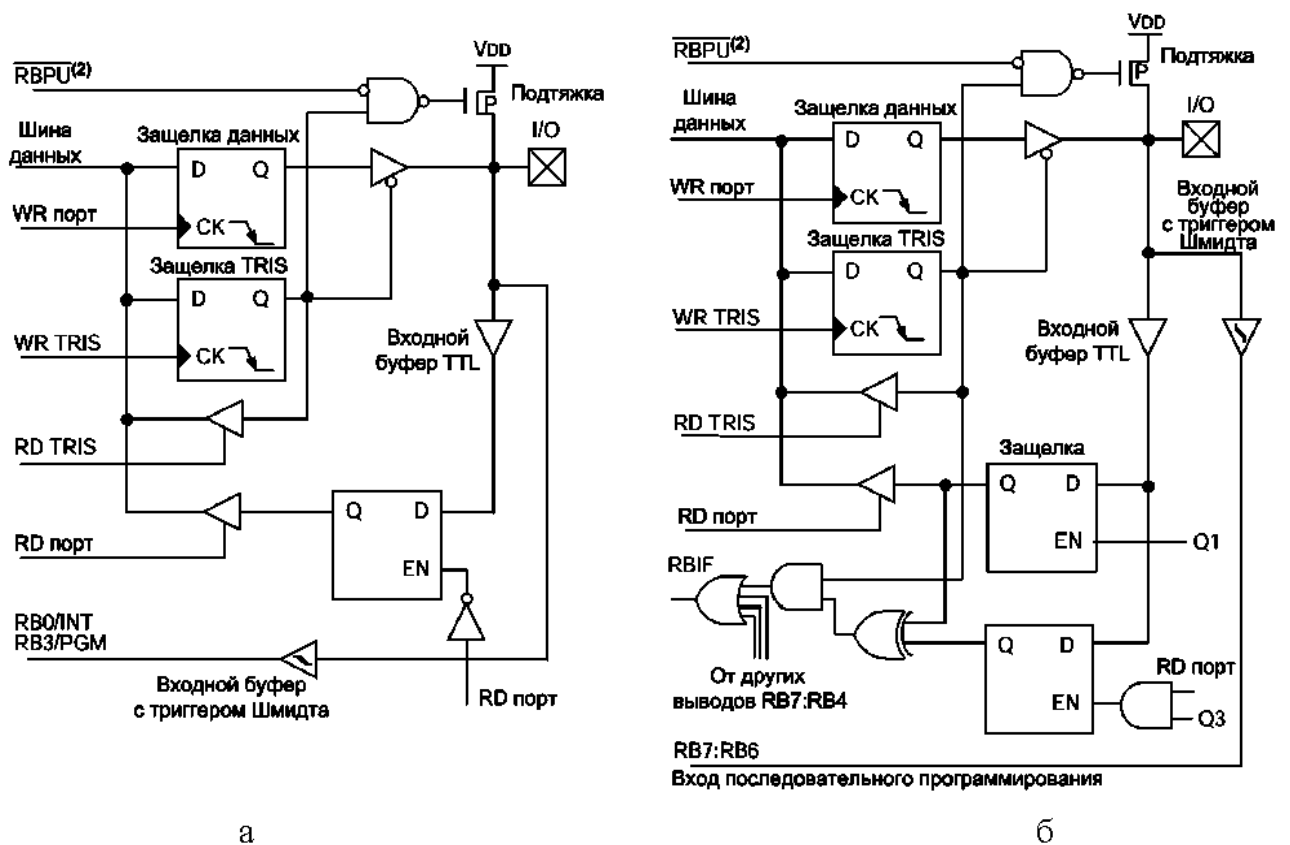


Рис.2. Структурная схема цепей управления линиями порта **PORTB**:
а – линии **RB<3:0>** ; б – линия **RB<7:4>**

1.3. Порт ввода-вывода **PORTC**

PORTC – 8-разрядный порт ввода-вывода, имеющий вспомогательный регистр **TRISC**, позволяющий настраивать отдельные линии как вход или выход. Запись логической ‘1’ в соответствующую (с тем же номером, что и отдельная линия) ячейку этого регистра переводит соответствующий выходной буфер в 3-е (высокоимпедансное) состояние. Запись логического ‘0’ в те же ячейки регистра **TRISC** определяет соответствующие линии порта как выходы. При этом содержимое регистра **PORTC** передается на соответствующие выводы микроконтроллера. Выводы **PORTC** мультиплицированы с несколькими периферийными модулями, при использовании которых необходимо соответствующим образом настраивать биты регистра **TRISC**. Некоторые периферийные модули модифицируют содержимое **TRISC**, настраивая отдельные линии на ввод или вывод. Структурная схема цепей управления линиями порта **PORTC** приведена на рис. 3.

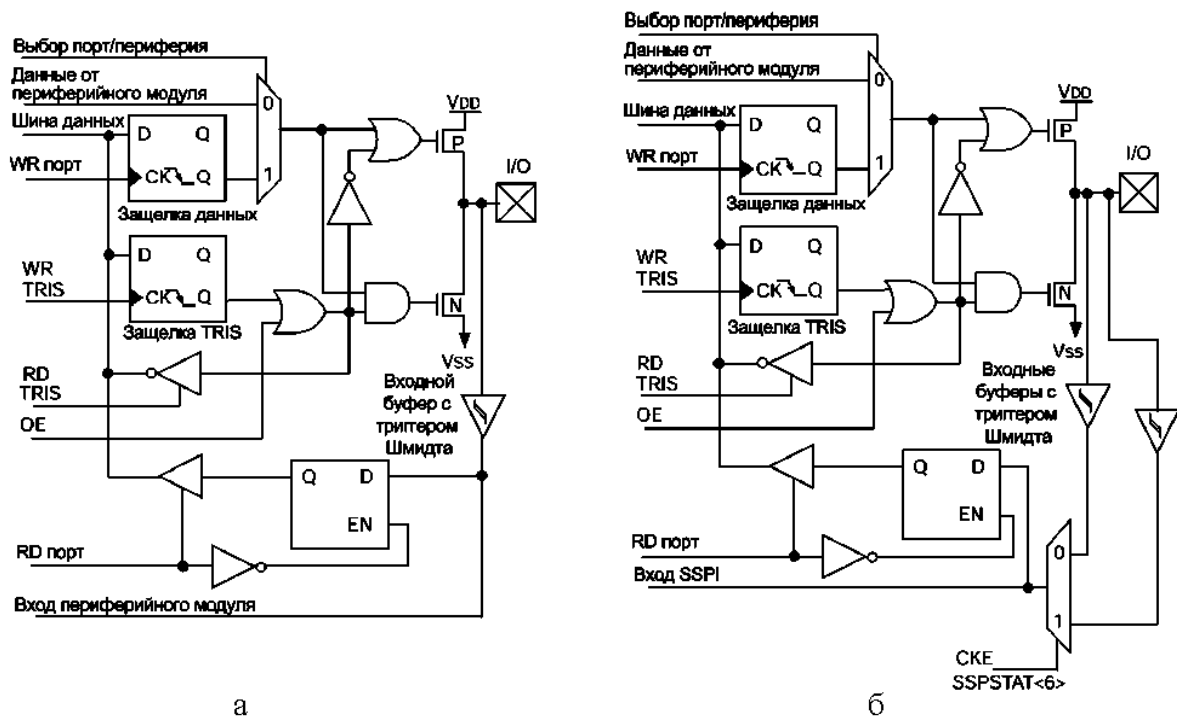


Рис.3. Структурная схема цепей управления линиями порта **PORTC**:
 а – линии **RC<7:5>** и **RC<2:0>**; б – линия **RC<4:3>**

1.4. Порт ввода-вывода **PORTD**

Регистры **POTRD** и **TRISD** не реализованы в микроконтроллерах **PIC16F873** и **PIC16F876**. В микроконтроллерах **PIC16F874** и **PIC16F877** **PORTD** – 8-разрядный двунаправленный порт, направление передачи которого управляется с помощью **TRISD** аналогично, как и **TRISA**, **TRISB** и **TRISC**. **PORTD** может работать как 8-разрядный микропроцессорный ведомый порт, если бит **PSPMODE (TRISE <4>)** установлен в '1'. Структурная схема цепей управления линиями порта **PORTD** приведена на рис. 4.

1.5. Порт ввода-вывода **PORTE**

PORTE и **TRISE** не реализованы в микроконтроллерах **PIC16F873** и **PIC16F876**. **PORTE** имеет три вывода индивидуально настраиваемых на ввод и вывод с помощью регистра **TRISE**. Если бит **PSPMODE<4>** установить в '1', то линии **RE0(-RD)**, **RE1(-WR)** и **RE2(-CS)** станут управляющими выводами ведомого параллельного порта **PORTD**. В этом режиме биты **TRISE<2:0>** должны быть установлены в '1'. Также в регистре **ADCON1** необходимо настроить линии **PORTE** как цифровые. Выводы **PORTE** мультиплицированы с

аналоговыми входами АЦП. После сброса по включению питания выводы настраиваются как аналоговые входы, а их чтение дает логические '0'. Структурная схема цепей управления линиями порта **PORTC** приведена на рис. 5.

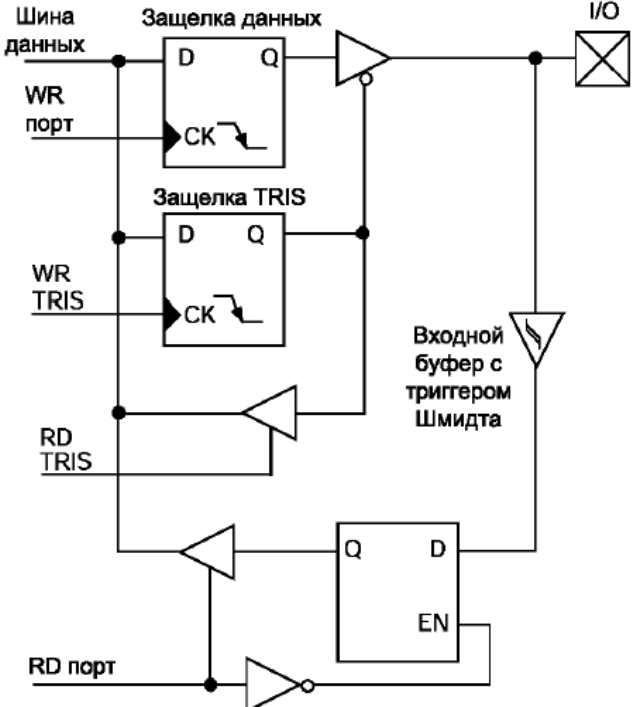


Рис. 4. Структурная схема цепей управления линиями порта **PORTD**

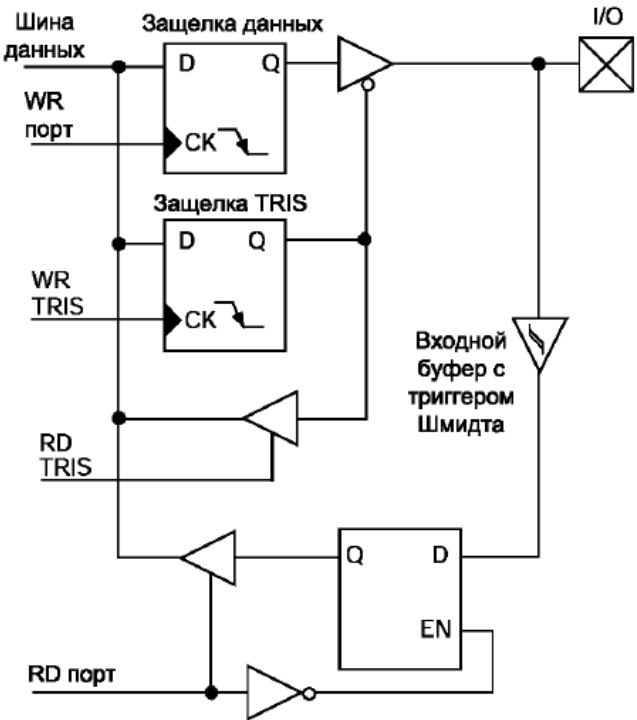
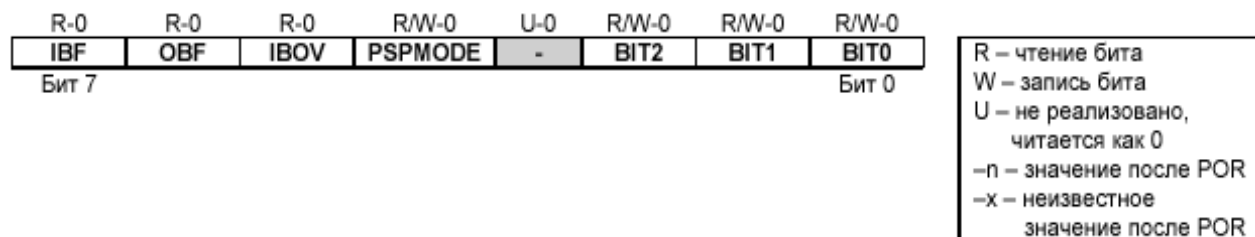


Рис.5. Структурная схема цепей управления линиями порта **PORTE**

Управление портом в режиме ввода-вывода и в режиме ведомого параллельного порта поясняет рис.6.



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

Рис.6. Назначение бит регистра **TRISE**

Режим параллельного ведомого порта (PSP) не реализован в микроконтроллерах PIC16F873 и PIC16F876. PORTD работает как 8-разрядный параллельный микропроцессорный порт, когда бит PSPMODE (TRISE<4>) установлен в '1'. В этом режиме данные читаются и записываются в PORTD асинхронно, с помощью внешних сигналов -RD (RE0 – Inverced Read), -WR (RE1 – Inverced Write) и -CS (RE2 – Inverced Chip Select). Принцип работы ведомого параллельного порта поясняет рис. 7.

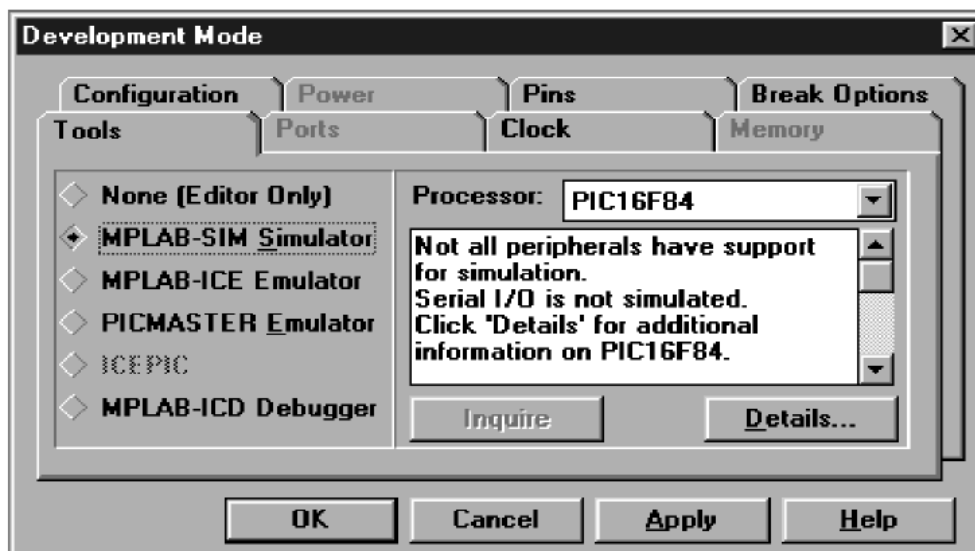


Рис.8. Окно выбора инструментального средства отладки



Рис. 9. Окно управления загрузчиком/программатором

Для ускорения внутрисхемной отладки следует до записи кода программы в поле возвращаемых значений регистров указать **‘Minimum & Watch windows’**. Откомпилированный исполняемый код программы будет загружен в память программ микроконтроллера после нажатия кнопки **‘Program’**. Когда загрузчик сообщит о завершении процесса программирования в окне **Status: Wait for User command** счетчик команд автоматически принимает значение 0x00. Затем с помощью меню **Debug > Run > Step** (или клавишей **F7**) иницируйте выполнение команд, помеченных темной полосой. Выбором пункта меню **Debug > Run > Run** или нажатием **<F9>** осуществите запуск программы с текущей позиции счетчика команд. Остановка программы произойдет после нажатия **<F5>** или **Debug > Run > Halt**.

ЛАБОРАТОРНАЯ РАБОТА №3

ИЗУЧЕНИЕ РАБОТЫ ВСТРОЕННОГО 10-БИТНОГО АЦП

Цель работы: Изучение работы портов встроенного 10-битного АЦП микроконтроллеров семейства PIC16F87X, закрепление навыков работы в отладочной среде MPLAB с отладочным модулем MPLAB-ICD Demo Board.

1. МОДУЛЬ АЦП МИКРОКОНТРОЛЛЕРОВ PIC16F87X

Встроенный аналого-цифровой 10-разрядный преобразователь имеет пять каналов у микроконтроллеров PIC16F873 и PIC16F876 (линии RA<5> и RA<3:0>) и восемь каналов у PIC16F874 и PIC16F877 (линии RE<2:0>, RA<5> и RA<3:0>). Напряжение, удерживаемое на линии соответствующего канала, преобразуется в 10-битный цифровой двоичный код с шагом квантования по уровню, зависящим от опорных напряжений VREF+ и VREF-, и составляющем (VREF+ и VREF-)/1024 В. В качестве источника опорных напряжений VREF+ и VREF- могут использоваться напряжение питания контроллера (VDD), общий вывод (земля VSS), а также напряжения на выводах RA2 или RA3. Для управления работой АЦП используются регистры ADCON0 и ADCON1, функции битовых полей которых применительно к микроконтроллерам PIC16F874 и PIC16F877 приведены на рис 1 и 2. Принцип выбора отдельных линий для преобразования входных напряжений и для опоры поясняется рис. 3.

Результаты аналого-цифрового преобразования сохраняются в регистрах ADRESH и ADRESL. Для корректной работы с модулем АЦП рекомендуется выполнять следующие операции:

1. Настройка АЦП:

- настроить линии портов PORTA и PORTE как аналоговые входы, выбрать линии опорного напряжения с помощью регистра ADCON1 (рис. 1);
- выбрать входной канал АЦП с помощью регистра ADCON0 (рис. 2);

- запустить АЦП, установив флаг GO/DONE (ADCON0<2>).
- 2. Разрешить прерывание от АЦП (только при использовании прерываний от АЦП):
 - сбросить флаг окончания преобразования ADIF (регистр PIR1<6>);
 - занести логическую '1' в ADIE (регистр ADIE<6>);
 - занести логическую '1' в PEIE (регистр INTCON<6>);
 - разрешить прерывания записью логической '1' в GIE (регистр INTCON<7>).
- 3. Обеспечить паузу не менее 20 мкс.
- 4. Запустить преобразование, установив флаг GO/DONE (ADCON0<2>).
- 5. Запустить цикл ожидания конца преобразования (по сбросу флага GO/DONE) или прерывания (если используется прерывание от АЦП).
- 6. Считать результаты преобразования из регистров ADRESH и ADRESL.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON
бит7							бит0

бит 7-6: **ADCS1:ADCS0**: выбор источника импульсов преобразования модуля АЦП

- 00 = Fosc/2
- 01 = Fosc/8
- 10 = Fosc/32
- 11 = FRC (источник импульсов – внутренний RC генератор)

бит 5-3: **CHS2:CHS0**: выбор рабочего 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)
- 110 = канал 6, (RE1/AN6)
- 111 = канал 7, (RE2/AN7)

бит 2: **GO/DONE**: бит состояния модуля АЦП

- 1 = модуль АЦП производит преобразование (установка бита вызывает начало преобразования)
- 0 = модуль АЦП преобразования нет (аппаратно сбрасывается по завершению преобразования)

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

бит 0: **ADON**: включение модуля АЦП

- 1 = включить модуль АЦП
- 0 = модуль АЦП выключен (что снижает общий ток потребления микроконтроллера)

Рис. 1. Регистр управления АЦП ADCON0

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0
бит7							бит0

бит 7: **ADFM**: Формат сохранения 10-битного результата
 1 = правое выравнивание. 6 старших бит ADRESH читаются как '0'
 0 = левое выравнивание. 6 младших бит ADRESL читаются как '0'

бит 6-4: **Не используется**: читается как '0'

бит 3-0: **PCFG3:PCFG0**: биты управления конфигурацией порта АЦП

PCFG3: PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	Vref+	Vref-
0000	A	A	A	A	A	A	A	A	VDD	VSS
0001	A	A	A	A	Vref+	A	A	A	RA3	VSS
0010	D	D	D	A	A	A	A	A	VDD	VSS
0011	D	D	D	A	Vref+	A	A	A	RA3	VSS
0100	D	D	D	D	A	D	A	A	VDD	VSS
0101	D	D	D	D	Vref+	D	A	A	RA3	VSS
011x	D	D	D	D	D	D	D	D	VDD	VSS
1000	A	A	A	A	Vref+	Vref-	A	A	RA3	RA2
1001	D	D	A	A	A	A	A	A	VDD	VSS
1010	D	D	A	A	Vref+	A	A	A	RA3	VSS
1011	D	D	A	A	Vref+	Vref-	A	A	RA3	RA2
1100	D	D	D	A	Vref+	Vref-	A	A	RA3	RA2
1101	D	D	D	D	Vref+	Vref-	A	A	RA3	RA2
1110	D	D	D	D	D	D	D	A	VDD	VSS
1111	D	D	D	D	Vref+	Vref-	D	A	RA3	RA2

A = аналоговый вход
 D = цифровой вход/выход

Рис. 2. Регистр управления АЦП ADCON1

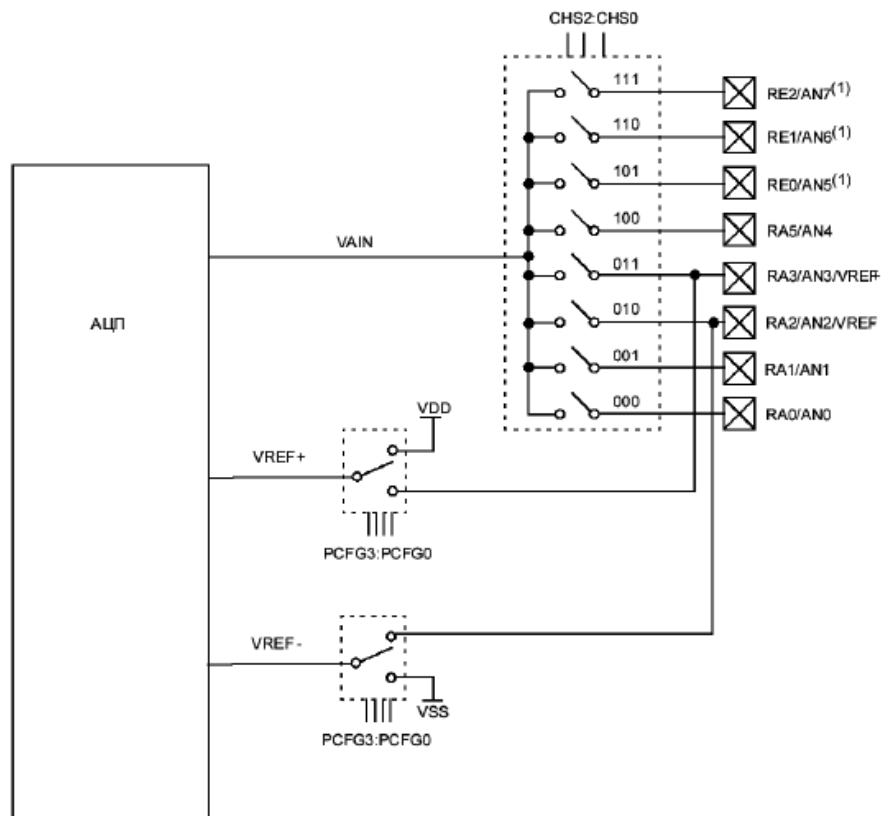


Рис.3. Блок-схема АЦП микроконтроллеров PIC16F874 и PIC16F877

При циклическом опросе АЦП повторяются только пункты 2-3-4-5-6 (только при использовании прерываний от АЦП) или 3-4-5-6. Формат чтения результатов преобразования из регистров ADRESH и ADRESL показан на рис. 4.

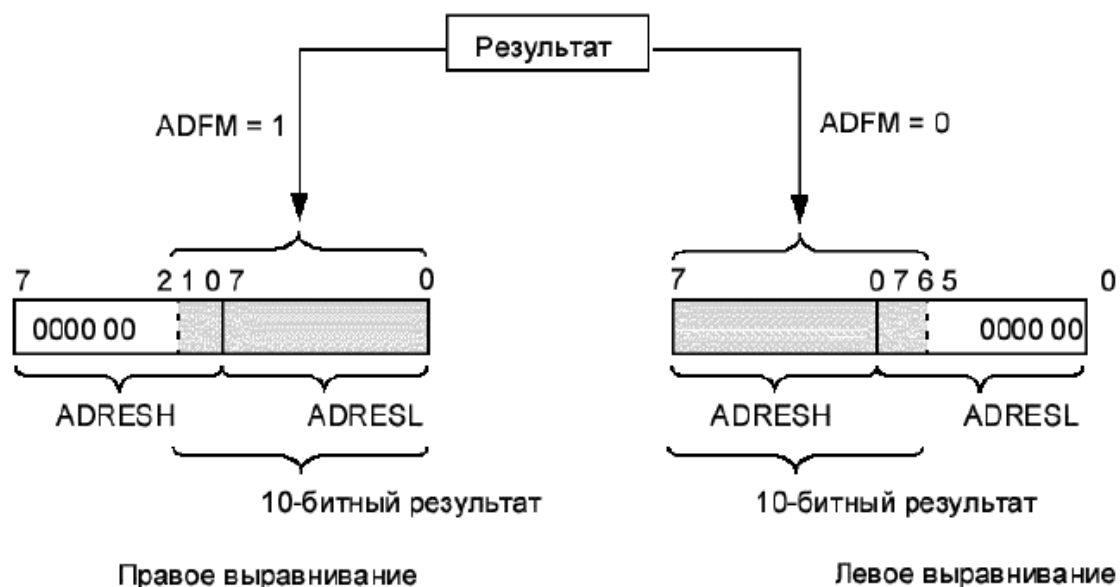


Рис. 4. Формат чтения результатов аналого-цифрового преобразования

Время 10-битного преобразования зависит от выбранного в регистре ADCON0 источника тактовых импульсов и коэффициента деления тактовых импульсов микроконтроллера для формирования импульсов аналого-цифрового преобразования. Согласно рис. 1 для формирования импульсов преобразования TAD доступны четыре варианта:

- $TAD = 2 T_{osc}$;
- $TAD = 8 T_{osc}$;
- $TAD = 32 T_{osc}$;
- Встроенный RC-генератор.

Здесь T_{osc} – период тактового сигнала микроконтроллера. Для 10-битного преобразования необходимо время, равное 12 TAD. При этом для достоверного преобразования время TAD должно быть не менее 1,6 и не более 8 мкс. Период тактовых импульсов встроенного RC-генератора колеблется в пределах 2 – 6 мкс. В табл. 1 приведены

значения TAD для различных тактовых частот микроконтроллера и коэффициентов деления T_{ad} /T_{osc}.

Таблица 1. Зависимость времени T_{ad} от рабочей частоты микроконтроллера (в затененных ячейках указаны T_{ad}, выходящие за пределы рекомендуемых значений)

Источник импульсов АЦП		Рабочая частота			
Источник	ADCS1:ADCS0	20МГц	5МГц	1.25МГц	333.33кГц
2TOSC	00	100нСек	400нСек	1.6 мкСек	6мкСек
8TOSC	01	400нСек	1.6 мкСек	6.4 мкСек	24мкСек
32TOSC	10	1.6 мкСек	6.4 мкСек	25.6 мкСек	96мкСек
RC-генератор	11	2-6мкСек	2-6мкСек	2-6мкСек	2-6мкСек

2. Порядок выполнения работы

- Ознакомиться с вводной частью лабораторной работы и схемой электрической принципиальной отладочного модуля **MPLAB-ICD Demo Board** (см. приложение 1).
- Получить задание у преподавателя.
- Запустить пакет **MPLAB-ICD**, создать и отладить проект в режиме симуляции выполнения программы микроконтроллера.
- Осуществить внутрисхемную отладку проекта, загрузив исполняемый код программы в память команд микроконтроллера.

ЛАБОРАТОРНАЯ РАБОТА №4

ИЗУЧЕНИЕ РАБОТЫ ТАЙМЕРОВ И СИСТЕМЫ ПЕРЕРЫВАНИЙ

Цель работы: Изучение работы таймеров TMR0, TMR1 и TMR2, а также системы прерываний микроконтроллеров семейства PIC16F87X, закрепление навыков работы в отладочной среде MPLAB с отладочным модулем MPLAB-ICD Demo Board.

1. МОДУЛЬ TMR0 МИКРОКОНТРОЛЛЕРОВ PIC16F87X

TMR0 – 8-разрядный таймер/счетчик с 8-разрядным программируемым предделителем, поддерживающий возможность чтения и записи текущего значения, выбора внутреннего или внешнего источника тактового сигнала, выбора активного фронта внешнего тактового сигнала и вызова аппаратного прерывания по переполнению. Блок-схема модуля TMR0 и общего с WDT предделителя приведена на рис. 1. При сбросе бита TOCS (OPTION_REG<5>) TMR0 работает как таймер, т.е. осуществляет подсчет машинных циклов микроконтроллера (следующих с частотой $1/(4 \times T_{osc})$). После записи текущего значения в TMR0 приращение счетчика запрещено два последующих цикла. При установке бита TOCS, TMR0 работает как счетчик внешних тактовых импульсов, поступающих на вывод RA4/TOCK1. При этом активный фронт внешнего тактового сигнала (передний или задний) определяется битом TOSE (если OPTION_REG<4>=0, активным является передний фронт). Программно недоступный предделитель включается перед WDT или TMR0 в зависимости от состояния бита PSA (OPTION_REG<3>).

Значения коэффициентов деления восьмиразрядного предделителя определяются комбинацией, заносимой в битовые поля PS<2:0> регистра OPTION_REG согласно табл. 1.

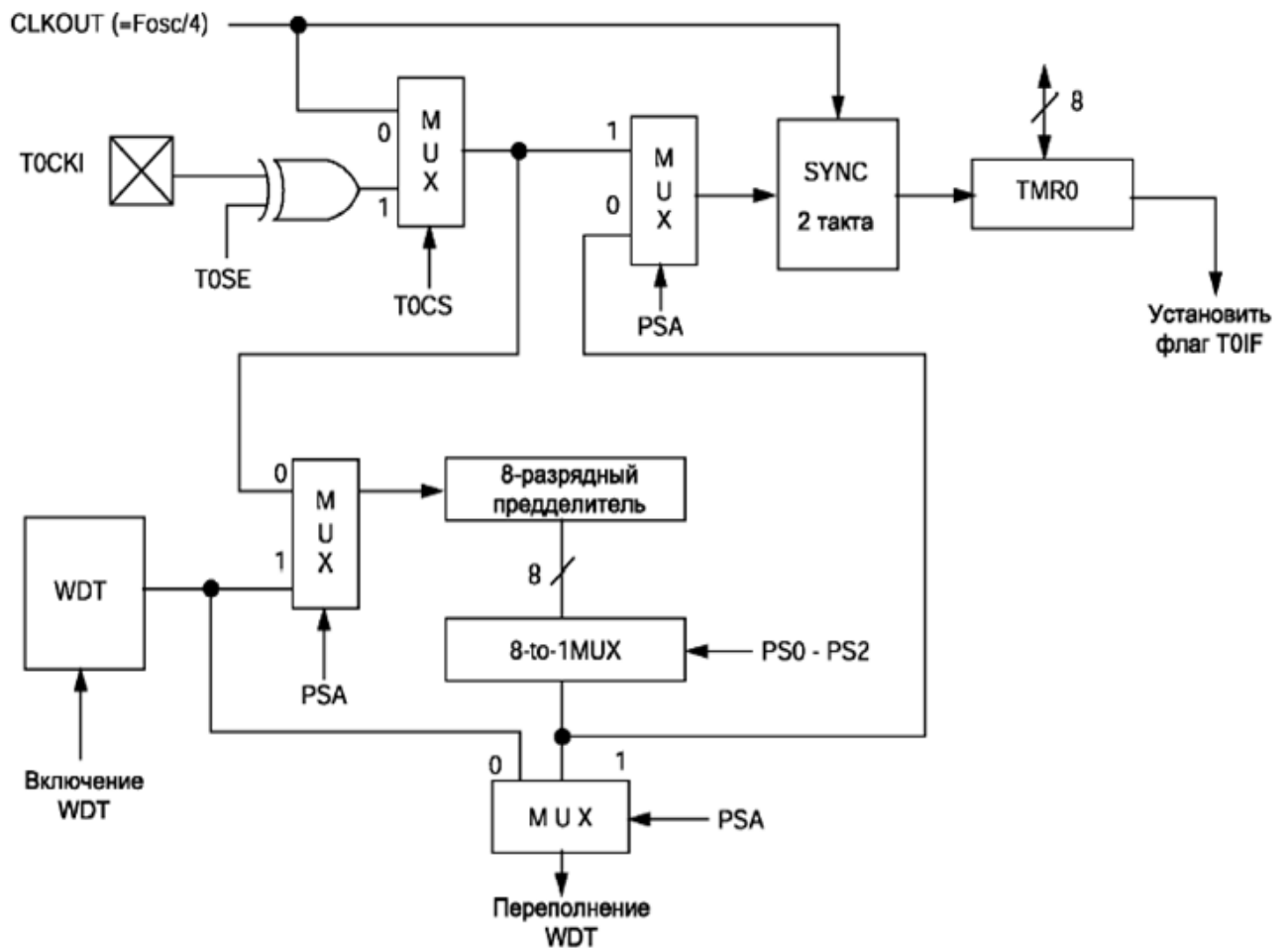


Рис. 1. Блок-схема модуля TMR0

Таблица 1. Коэффициенты деления 8-разрядного предделителя для модулей WDT и TMR0

Значения PS<2:0>	Коэффициент деления для TMR0	Коэффициент деления 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

2. МОДУЛЬ TMR1 МИКРОКОНТРОЛЛЕРОВ PIC16F87X

TMR1 – 16-разрядный таймер/счетчик, состоящий из двух 8-разрядных регистров (TMR1H и TMR1L), доступных для чтения и записи. Инкрементирование счетчика происходит с каждым тактовым импульсом, источником которого, в режиме таймера, является предделитель (делитель частоты машинных циклов), либо внешний источник, подключенный к выводу RC0 (в режиме таймера). Модуль TMR1 также поддерживает возможность подсчета импульсов, генерируемых встроенным RC-генератором, пассивные элементы которого подключаются к выводам RC0 – RC1. Блок-схема модуля TMR1 приведена на рис. 2.

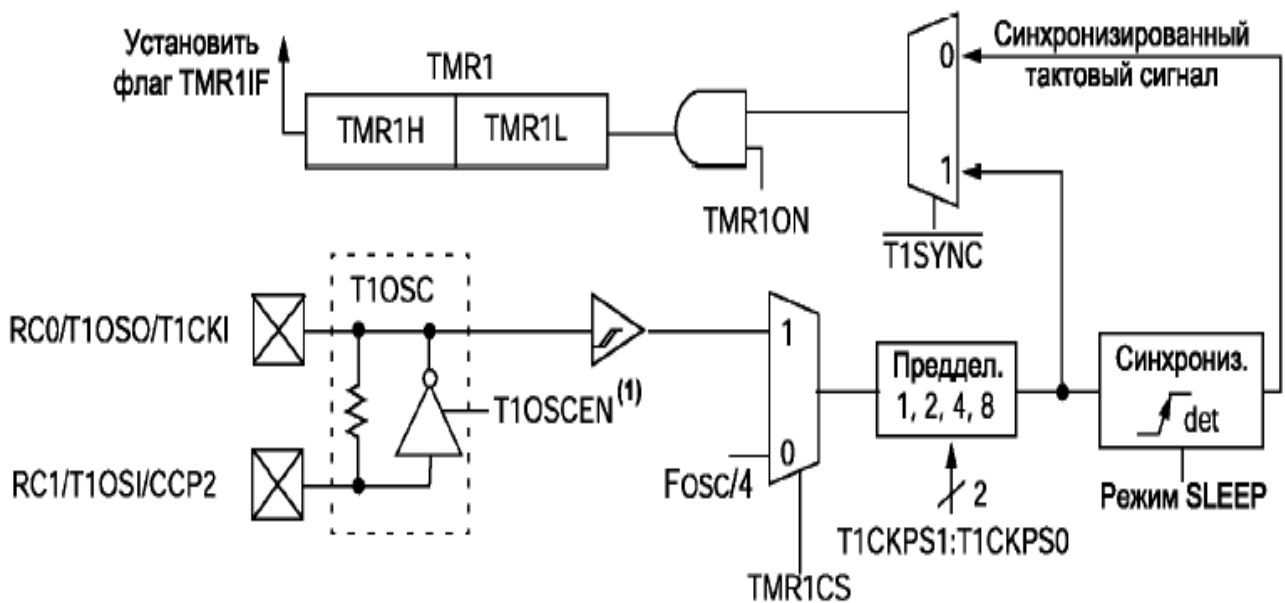


Рис. 2. Блок-схема модуля TMR1

Для управления работой модуля TMR1 используется регистр T1CON, функции битовых полей которого применительно к микроконтроллерам PIC16F874 и PIC16F877 приведены на рис 3.

После переполнения 16-разрядного счетчика (максимальное значение FFFFh) флаг прерывания TMR1IF в регистре PIR1<0> устанавливается в '1', а счетчик сбрасывается в 0000h.

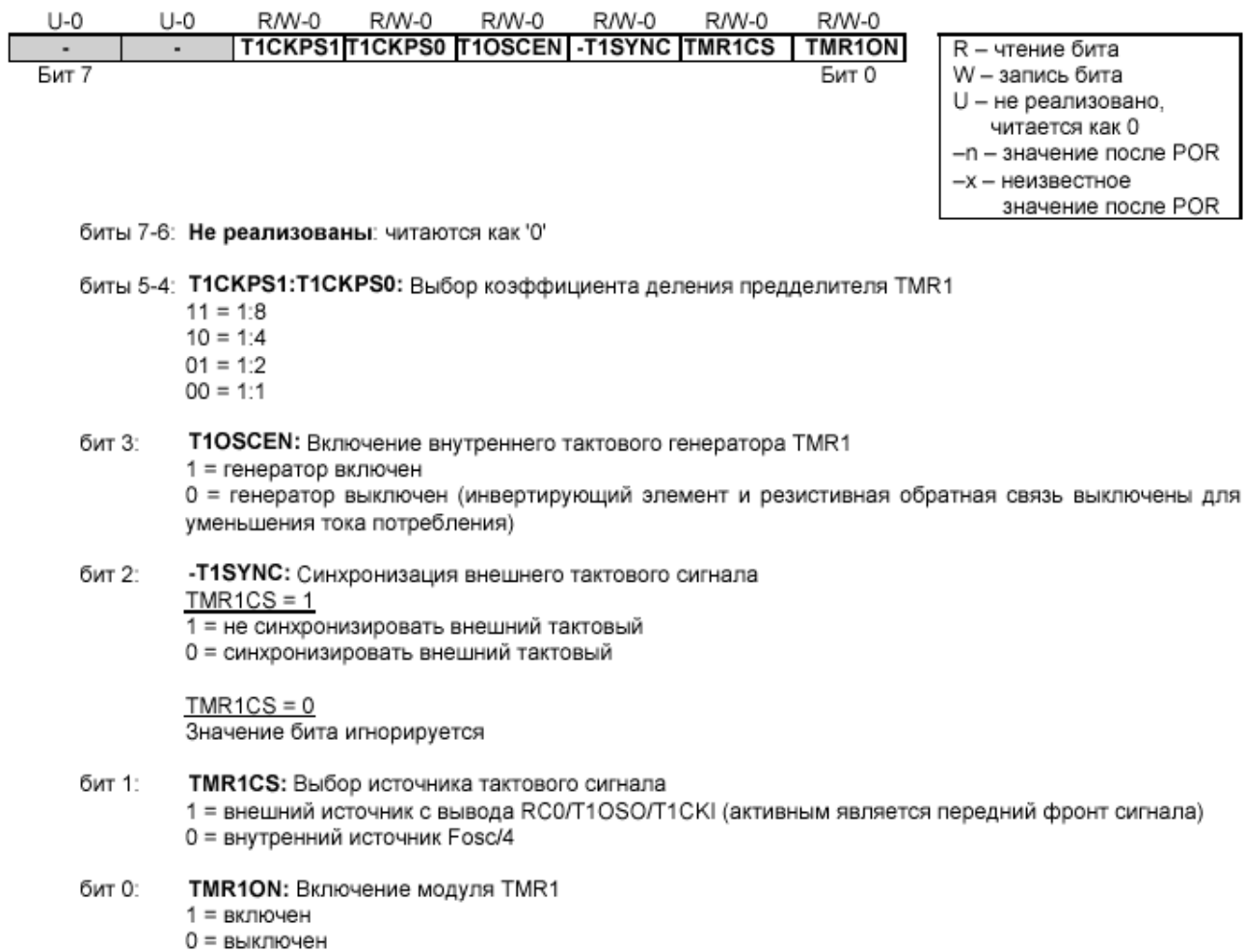


Рис. 3. Регистр управления режимами работы модуля TMR1

3. МОДУЛЬ TMR2 МИКРОКОНТРОЛЛЕРОВ PIC16F87X

TMR2 – 8-разрядный таймер с программируемым делителем, выходным делителем и 8-разрядным регистром периода PR2, позволяющим организовать программно управляемый ШИМ-контур. Регистры, связанные с работой TMR2, доступны для записи и чтения и сбрасываются в '0' при любом виде сброса. Блок-схема модуля TMR1 приведена на рис. 4.

Входной тактовый сигнал ($1/(4 \times T_{osc})$) поступает через делитель с программируемым коэффициентом деления (1:1, 1:4 или 1:16), определяемый битами T2CKPS1 и T2CKPS0 (T2CON<1:0>). TMR2 инкрементируется с каждым машинным циклом от 00h до значения, записанного в PR2, после чего сбрасывается в 00h. После сброса значение регистра PR2 равно FFh.

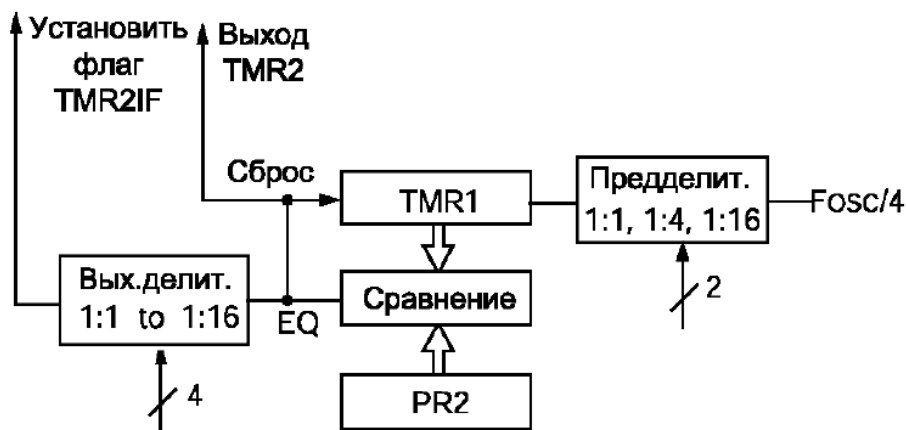
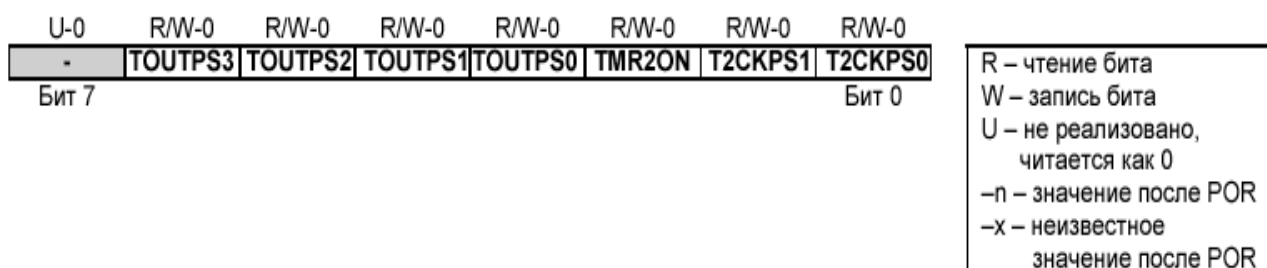


Рис. 4. Блок-схема модуля TMR2

Сигнал переполнения TMR2 проходит через выходной 4-разрядный делитель с программируемым коэффициентом деления (от 1:1 до 1:16) для установки флага TMR2IF в регистре PIR1<1>. Для уменьшения энергопотребления таймер TMR2 может быть выключен сбросом бита TMR2ON (T2CON<2>) в '0'. Функции битовых полей регистра T2CON приведены на рис. 5.



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

биты 6-3: **TOUTPS3:TOUTPS0:** Выбор коэффициента выходного делителя TMR2

0000 = 1:1
0001 = 1:2
:
:
1111 = 1:16

бит 2: **TMR2ON:** Включение модуля TMR2

1 = включен
0 = выключен

биты 1-0: **T2CKPS1:T2CKPS0:** Выбор коэффициента деления предделителя TMR2

00 = 1:1
01 = 1:4
1x = 1:16

Рис. 5. Регистр управления режимами работы модуля TMR1

4. СИСТЕМА ПРЕРЫВАНИЙ МИКРОКОНТРОЛЛЕРОВ СЕМЕЙСТВА PIC16F87X

Микроконтроллеры семейства **PIC16F87X** имеют 14 источников прерываний. Регистр **INTCON** содержит флаги отдельных прерываний, биты их разрешения и бит глобального разрешения прерываний. Если бит **GIE** (**INTCON**<7>) установлен в '1', то все прерывания разрешены. Этот бит после сброса установлен в '0'. При возникновении прерывания поднимается соответствующий флаг в регистре **INTCON**, и, если прерывание разрешено, вызывается процедура обработки соответствующего прерывания и сбрасывается флаг **GIE**, запрещая вызов других прерываний. По окончании процедуры обработки прерывания после выполнения команды **REFIE** этот флаг возвращается в '1'. При переходе на процедуру обработки прерывания текущее значение из счетчика команд заносится в стек, а в счетчик команд загружается единый вектор **0004h**. Определение источника прерываний осуществляется программно посредством проверки флагов, поэтому в процессе настройки прерываний эти флаги следует сбросить с помощью соответствующих битовых полей в регистрах **INTCON**, **PIE1** и **PIE2**. Следует помнить, что флаги прерываний устанавливаются независимо от состояния бита **GIE**. Функции битовых полей регистров **INTCON**, **PIE1** и **PIE2** приведены на рис. 6, 7, 8.

5. Порядок выполнения работы

- Ознакомиться с вводной частью лабораторной работы и схемой электрической принципиальной отладочного модуля **MPLAB-ICD Demo Board** (см. приложение 1).
- Получить задание у преподавателя.
- Запустить пакет **MPLAB-ICD**, создать и отладить проект в режиме симуляции выполнения программы микроконтроллера.
- Осуществить внутрисхемную отладку проекта, загрузив исполняемый код программы в память команд микроконтроллера.

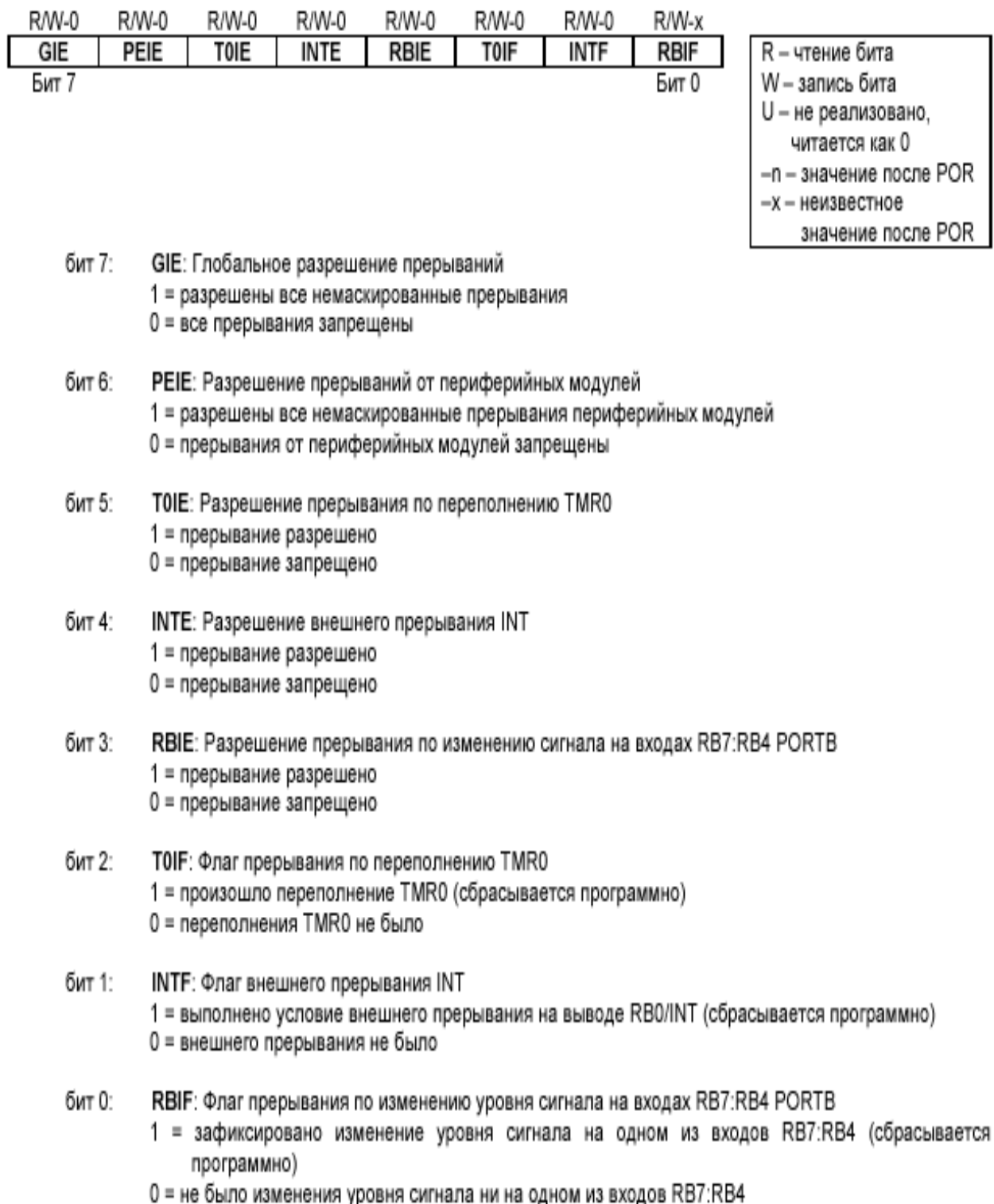


Рис. 6. Регистр управления прерываниями INTCON

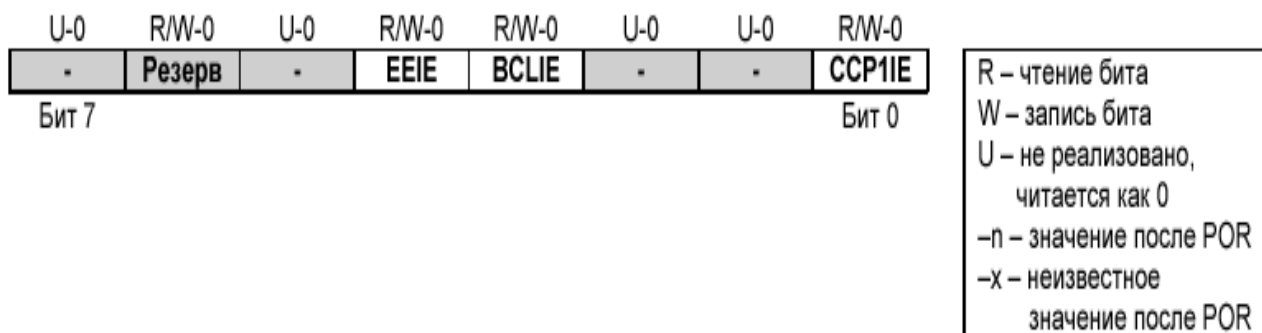
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
Бит 7							Бит 0

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

- бит 7: **PSPIE⁽¹⁾**: 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.

Рис. 7. Регистр разрешения/запрещения прерываний от функциональных модулей PIE1



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

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

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

бит 4: **EEIE:** Разрешение прерывания по окончании записи в EEPROM данных
1 = прерывание разрешено
0 = прерывание запрещено

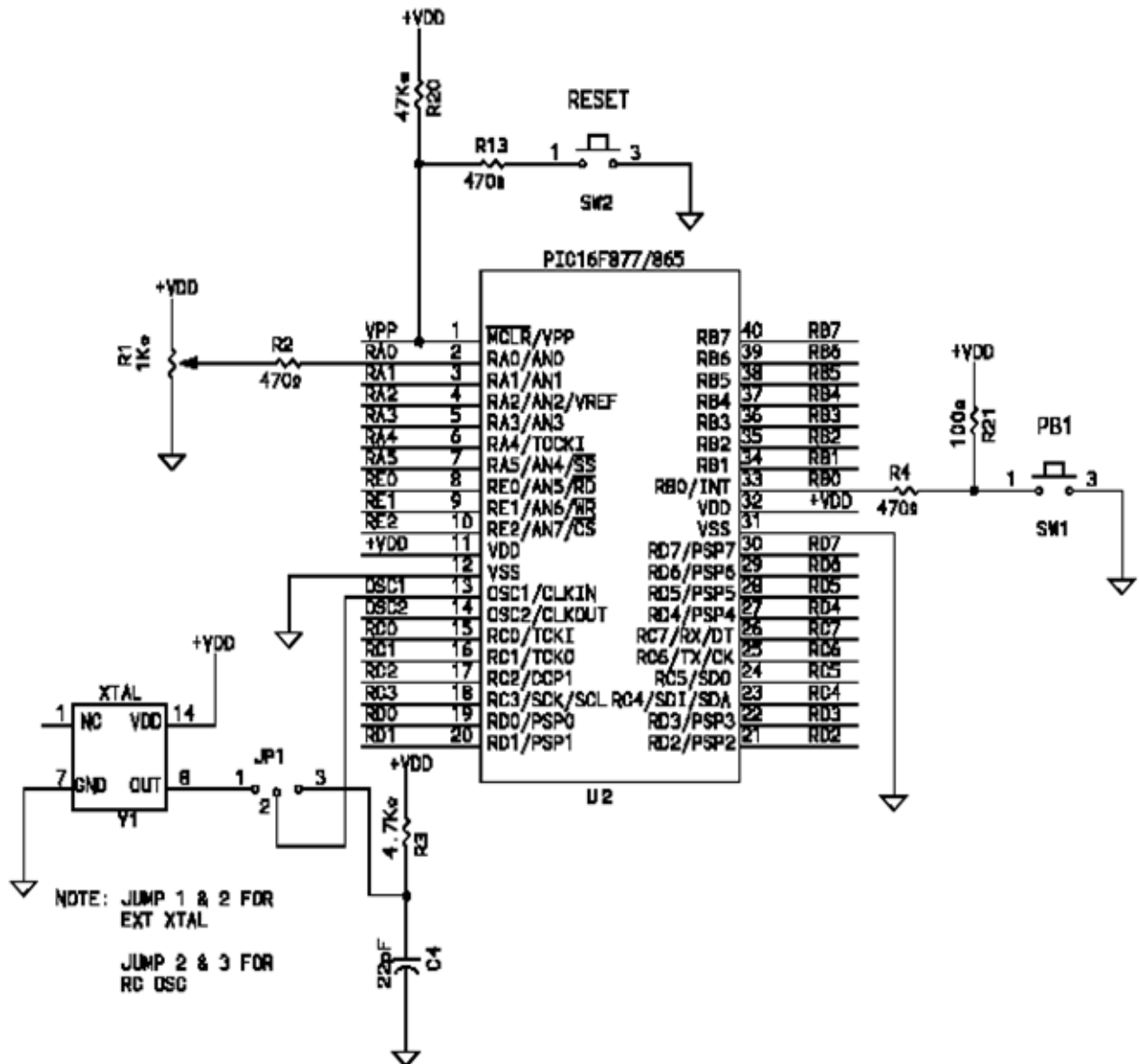
бит 3: **BCLIE:** Разрешение прерывания при возникновении коллизий на шине
1 = прерывание разрешено
0 = прерывание запрещено

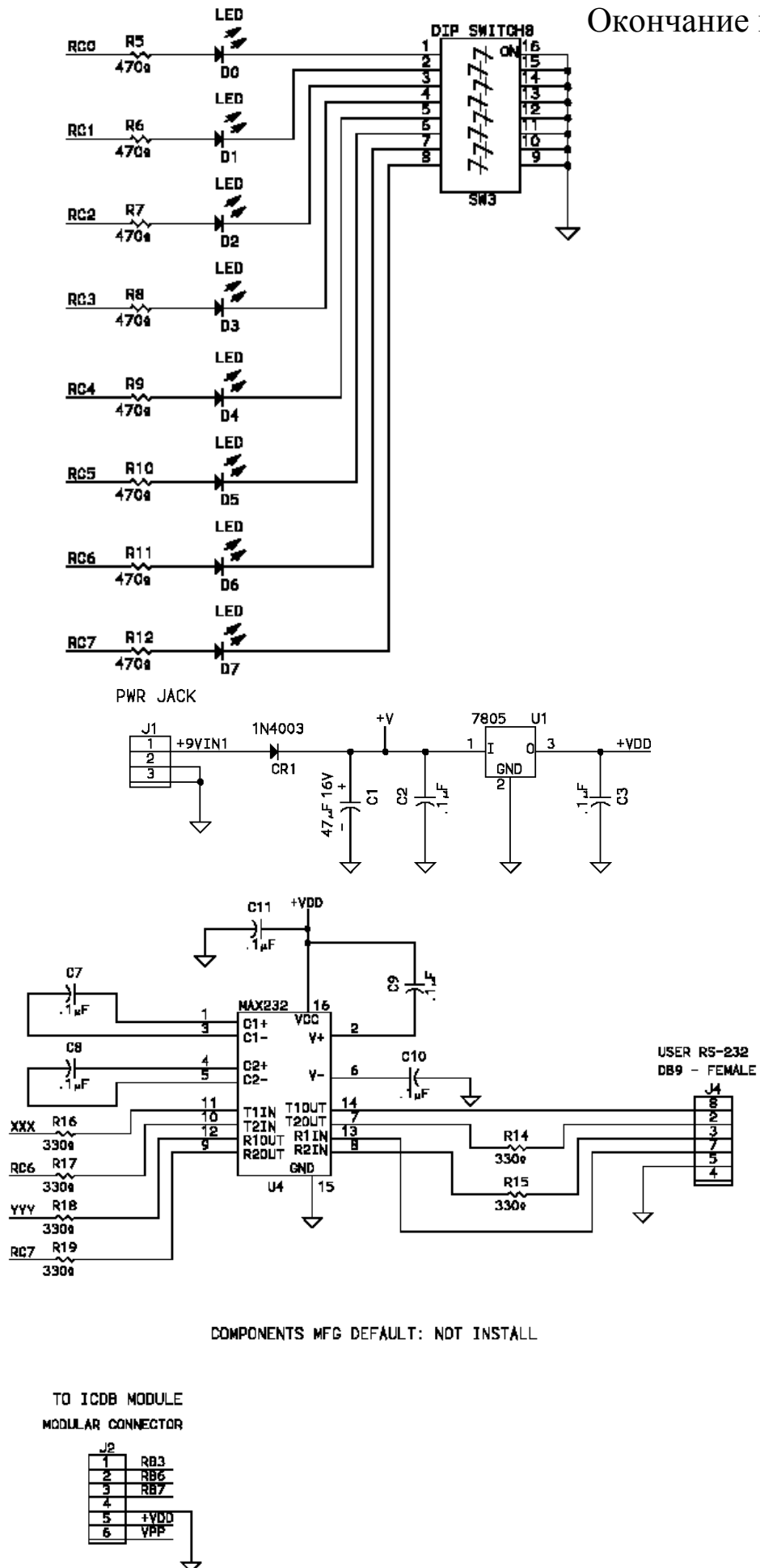
биты 2-1: **Не реализованы:** читаются как '0'

бит 0: **CCP2IE:** Разрешение прерывания от модуля CCP2
1 = прерывание разрешено
0 = прерывание запрещено

Рис. 6. Регистр разрешения/запрещения прерываний от функциональных модулей PIE2

Схема электрическая принципиальная
отладочного модуля MPLAB-ICD Demo Board





Область регистров специальных функций

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR
Банк 0										
00h	INDF	Обращение к регистру, адрес которого записан в FSR (не физический регистр)								0000 0000
01h	TMR0	Регистр таймера 0								xxxx xxxx
02h	PCL	Младше биты счетчика команд PC								0000 0000
03h	STATUS	IRP	RP1	RP0	-TO	-PD	Z	DC	C	0001 1xxx
04h	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	PORTD	Запись в выходную защелку PORTD, чтение состояния выводов PORTD								xxxx xxxx
09h	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -xxx
0Ah	PCLATH	-	-	-	Старшие биты счетчика команд PC					---0 0000
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000
0Dh	PIR2	-	(5)	-	EEIF	BCLIF	-	-	CCP2IF	-x-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	CKP	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

Банк 1											
80h	INDF	Обращение к регистру, адрес которого записан в FSR (не физический регистр)								0000	0000
81h	OPTION_REG	-RBPV	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111	1111
82h	PCL	Младше биты счетчика команд PC								0000	0000
83h	STATUS	IRP	RP1	RP0	-TO	-PD	Z	DC	C	0001	1xxx
84h	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	TRISD	Направление выводов PORTD								1111	1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	-	Направление выв. PORTE			0000	-111
8Ah	PCLATH	-	-	-	Старшие биты счетчика команд PC					---0	0000
8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000	000x
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000	0000
8Dh	PIE2	-	(5)	-	EEIE	BCLIE	-	-	CCP2IE	-x-0	0--0
8Eh	PCON	-	-	-	-	-	-	-	-POR	-BOR	---- --qq
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

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Сброс POR, BOR
Банк 2										
100h	INDF	Обращение к регистру, адрес которого записан в FSR (не физический регистр)								0000 0000
101h	TMR0	Регистр таймера 0								xxxx xxxx
102h	PCL	Младше биты счетчика команд PC								0000 0000
103h	STATUS	IRP	RP1	RP0	-TO	-PD	Z	DC	C	0001 1xxx
104h	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	PCLATH	-	-	-	Старшие биты счетчика команд PC					---0 0000
10Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	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
Банк 3										
180h	INDF	Обращение к регистру, адрес которого записан в FSR (не физический регистр)								0000 0000
181h	OPTION_REG	-RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
182h	PCL	Младше биты счетчика команд PC								0000 0000
183h	STATUS	IRP	RP1	RP0	-TO	-PD	Z	DC	C	0001 1xxx
184h	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	PCLATH	-	-	-	Старшие биты счетчика команд PC					---0 0000
18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x
18Ch	EECON1	EEPGD	-	-	-	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'.

Учебное издание

МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА PIC16F87X

Методические указания к лаб. работам

Составитель *Лофицкий Игорь Вадимович*

Редактор Н. С. К у п р и я н о в а
Компьютерная верстка Т. Е. П о л о в н е в а

Подписано в печать 25. 05.07 г. Формат 60x84 1/16.

Бумага офсетная. Печать офсетная.

Усл. печ. л. 2,75.

Тираж 150 экз. Заказ . Арт. С-38/2007.

Самарский государственный
аэрокосмический университет.
443086 Самара, Московское шоссе, 34.

Издательство Самарского государственного
аэрокосмического университета.
443086 Самара, Московское шоссе, 34.