

ОПИСАНИЕ КОНТРОЛЛЕРА PIC16F84.....	2
Обзор характеристик.....	2
Структурная схема PIC16F84.....	5
Разводка ножек.....	6
Назначение ножек.....	6
Обзор регистров и ОЗУ.....	7
TMR0 таймер/счетчик.....	9
Организация встроенного ПЗУ.....	13
Данные в EEPROM.....	14
Организация прерываний.....	16
Обзор регистров/портов.....	19
Обзор команд и обозначения.....	22
Условия сброса.....	24
Watch Dog таймер.....	26
Типы генераторов.....	26
Регистр OPTION_REG.....	30
Конфигурационное слово.....	32
Индивидуальная метка.....	32
Защита программ от считывания.....	32
Режим пониженного энергопотребления.....	33

Описание контроллера PIC16F84

PIC16F84 относится к семейству КМОП микроконтроллеров. Отличается тем, что имеет внутреннее 1К x 14 бит EEPROM для программ, 8-битовые данные и 64байт EEPROM памяти данных. При этом отличаются низкой стоимостью и высокой производительностью. Пользователи, которые знакомы с семейством PIC16C5X могут посмотреть подробный список отличий нового от производимых ранее контроллеров. Все команды состоят из одного слова (14 бит шириной) и исполняются за один цикл (1 мкс при 4 МГц), кроме команд перехода, которые

выполняются за два цикла (2 мкс). PIC16F84 имеет прерывание, срабатывающее от четырех источников, и восьмиуровневый аппаратный стек. Периферия включает в себя 8-битный таймер/счетчик с 8-битным программируемым предварительным делителем (фактически 16 - битный таймер) и 13 линий двунаправленного ввода/вывода. Высокая нагрузочная способность (25 мА макс. входной ток, 20 мА макс. выходной ток) линий ввода/вывода упрощают внешние драйверы и, тем самым, уменьшается общая стоимость системы. Разработки на базе контроллеров PIC16F84 поддерживаются ассемблером, программным симулятором, внутрисхемным эмулятором (только фирмы Microchip) и программатором.

Серия PIC16F84 подходит для широкого спектра приложений от схем высокоскоростного управления автомобильными и электрическими двигателями до экономичных удаленных приемопередатчиков, показывающих приборов и связанных процессоров. Наличие ПЗУ позволяет подстраивать параметры в прикладных программах (коды передатчика, скорости двигателя, частоты приемника и т.д.).

Малые размеры корпусов, как для обычного, так и для поверхностного монтажа, делает эту серию микроконтроллеров пригодной для портативных приложений. Низкая цена, экономичность, быстродействие, простота использования и гибкость ввода/вывода делает PIC16F84 привлекательным даже в тех областях, где ранее не применялись микроконтроллеры. Например, таймеры, замена жесткой логики в больших системах, сопроцессоры.

Следует добавить, что встроенный автомат программирования EEPROM кристалла PIC16F84 позволяет легко подстраивать программу и данные под конкретные требования даже после завершения ассемблирования и тестирования. Эта возможность может быть использована как для тиражирования, так и для занесения калибровочных данных уже после окончательного тестирования.

Обзор характеристик

Высокоскоростной RISC процессор

- ◆ только 35 простых команд;
- ◆ все команды выполняются за один цикл(1 мкс), кроме команд перехода, выполняющихся за два цикла;
- ◆ рабочая частота 0 Гц ... 4 МГц (min 1 мкс цикл команды);

- ◆ 14- битовые команды;
- ◆ 8- битовые данные;
- ◆ 1024 x 14 электрически перепрограммируемой программной памяти на кристалле (EEPROM);
- ◆ 36 x 8 регистров общего использования;
- ◆ 15 специальных аппаратных регистров SFR;
- ◆ 64 x 8 электрически перепрограммируемой EEPROM памяти для данных;
- ◆ восьмиуровневый аппаратный стек;
- ◆ прямая, косвенная и относительная адресация данных и команд;
- ◆ четыре источника прерывания:
 - внешний вход INT,
 - переполнение таймера TMR0,
 - прерывание при изменении сигналов на линиях порта В,
 - по завершению записи данных в память EEPROM.

Периферия и Ввод/Вывод

- ◆ 13 линий ввода-вывода с индивидуальной настройкой;
- ◆ входной/выходной ток для управления светодиодами.
 - макс. входной ток - 20 мА. ,
 - макс. выходной ток - 25 мА.,
- ◆ TMR0: 8 - битный таймер/счетчик TMR0 с 8-битным программируемым предварительным делителем.

Специальные свойства

- ◆ автоматический сброс при включении;
- ◆ таймер включения при сбросе;

- ◆ таймер запуска генератора;
- ◆ **WatchDog** таймер (WDT) с собственным встроенным генератором, обеспечивающим повышенную надежность;
- ◆ **EEPROM** бит секретности для защиты кода;
- ◆ экономичный режим **SLEEP**;
- ◆ выбираемые пользователем биты для установки режима возбуждения встроенного генератора:
 - RC генератор : RC;
 - обычный кварцевый резонатор : XT;
 - высокочастотный кварцевый резонатор : HS;
 - экономичный низкочастотный кристалл : LP;
- ◆ встроенное устройство программирования EEPROM памяти программ и данных; используются только две ножки.

КМОП технология

- ◆ экономичная высокоскоростная КМОП EPROM технология;
- ◆ статический принцип в архитектуре;
- ◆ широкий диапазон напряжений питания и температур:
 - коммерческий: 2.0 ... 6.0 В, 0...+70С,
 - промышленный: 2.0 ... 6.0 В, -40...+70С,
 - автомобильный: 2.0 ... 6.0 В, 40...+125С;
- ◆ низкое потребление:
 - 2 мА типично для 5В, 4МГц,
 - 15 мкА типично для 2В, 32КГц,
 - 1 мкА типично для SLEEP режима при 2В.

Типы корпусов и исполнений

Обозначения корпусов для кристаллов PIC16F84. Тип корпуса указывается в Маркировке при заказе микросхем. Корпуса бывают только с 18 Выводами.

PDIP - Обычный пластмассовый двухрядный корпус

SOIC - Малогабаритный DIP корпус для монтажа на поверхность

Маркировка при заказе

Обозначение микросхем складывается из следующих полей:

Фирм.номер/Частота генератора/Темпер.диапазон/Корпус/Примеч

Фирм.номер : PIC16F84 Vdd range 4...6 V,

PIC16LC84 Vdd range 2...6 V;

Частота генератора: 04 ---> 4 mHz (*большинство 4 МГц. приборов работает до 10 МГц.!*),

10 ---> 10mHz;

Температурный диапазон бывает:

- от 0С до +70С,

I от-40С до +85С,

E от-40С до +125С;

Корпус обозначается:

P - обычный пластмассовый DIP,

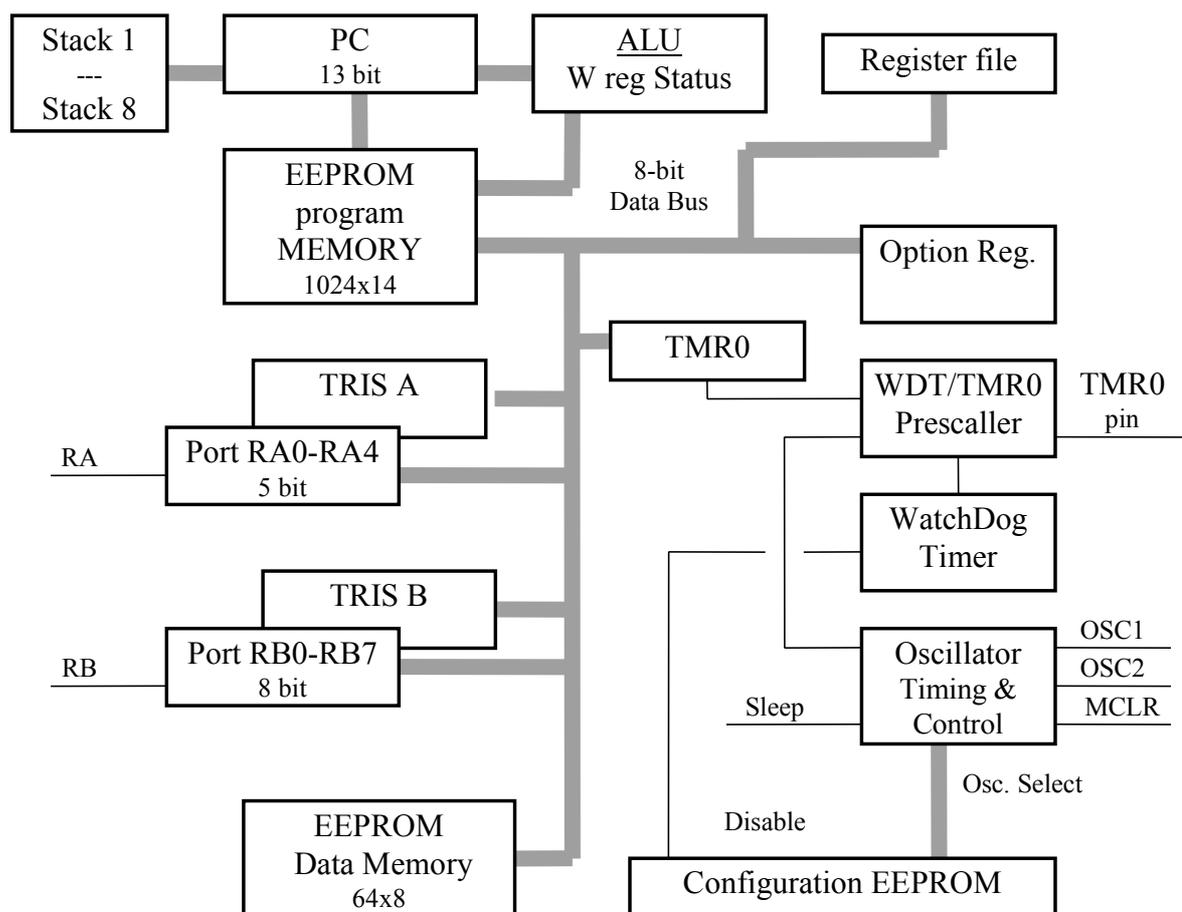
SO -300 mil SOIC.

ПРИМЕРЫ:

PIC16C84-04/P - 4 mHz, коммерческое исполнение в PDIP корпусе,
норм.диапазон Vdd

PIC16LC84-04I/SO - 4 mHz, исполнение для промышленности, расширенный диапазон питания, корпус=SOIC

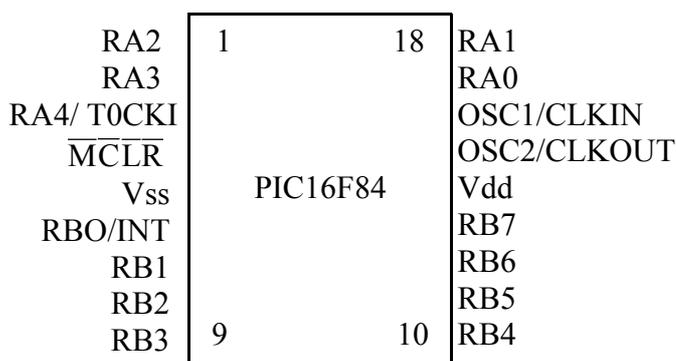
PIC16C84-10E/P - исполнение для автомобилей, 10 mHz, PDIP, норм. питание

Структурная схема PIC16F84

Архитектура основана на концепции отдельных шин и областей памяти для данных и для команд (Гарвардская архитектура). Шина данных и память данных (ОЗУ) - имеют ширину 8 бит, а программная шина и программная память (ПЗУ) имеют ширину 14 бит. Такая концепция обеспечивает простую, но мощную систему команд, разработанную так, что битовые, байтовые и регистровые операции работают с высокой скоростью и с перекрытием по времени выборок команд и циклов выполнения. 14-битовая ширина программной памяти обеспечивает выборку 14-битовой команды в один цикл. Двухступенчатый конвейер обеспечивает одновременную выборку и исполнение команды. Все команды выполняются за один цикл, исключая команды переходов. В PIC16F84 программная память объемом 1К x 14 расположена внутри кристалла. Исполняемая программа может находиться только во встроенном ПЗУ.

Разводка ножек

Функциональное назначение выводов см. "Обозначения Выводов" или Структурную Схему. Типы корпусов PDIP и др. см. "Корпуса".



Назначение ножек

Обозначения ножек и их функциональное назначение

Обозначение	Нормальный режим	Режим записи EEPROM
RA0 – RA3	Двунаправленные линии ввода/вывода. Входные уровни TTL	
RA4/T0CKI	Вход через триггер Шмитта. Ножка порта ввода/вывода с открытым стоком или вход частоты для таймера/счетчика TMR0	
RB0/INT	Двунаправленная линия порта ввода/ вывода или внешний вход прерывания Уровни TTL	
RB1 – RB5	Двунаправленные линии ввода/ вывода. Уровни TTL	
RB6	Двунаправленные линии ввода/ вывода. Уровни TTL.	Вход тактовой частоты для EEPROM
RB7	Двунаправленные линии ввода/ вывода. Уровни TTL.	Вход/выход EEPROM данных.
\overline{MCLR} /V _{pp}	Низкий уровень на этом входе генерирует сигнал сброса для контроллера. Активный низкий.	Сброс контроллера Для режима EEPROM- подать V _{pp} .
OSC1/CLKIN	Для подключения кварца,	

Обозначение	Нормальный режим	Режим записи EEPROM
	RC или вход внешней тактовой частоты	
OSC2/CLKOUT	Генератор, выход тактовой частоты в режиме RC генератора, в остальных случаях - для подкл.кварц	
Vdd	Напряжение питания	Напряжение питания
Vss	Общий(земля)	Общий

Обзор регистров и ОЗУ

Область ОЗУ организована как 128 x 8. К ячейкам ОЗУ можно адресоваться прямо или косвенно, через регистр указатель FSR (04h). Это также относится и к EEPROM памяти данных-констант.

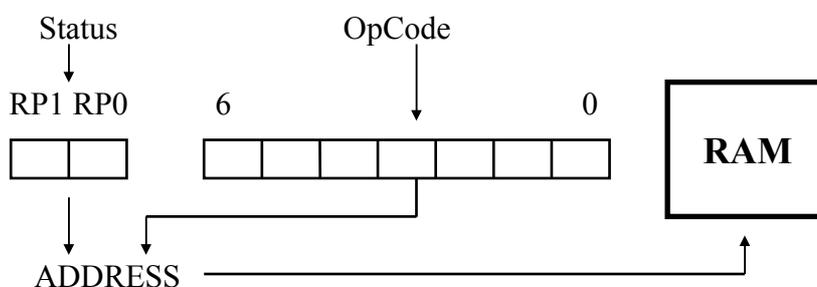
	Page 0	Page 1	
00	Indirect add.		80
01	TMR0	OPTION_REG	81
02	PCL		82
03	STATUS		83
04	FSR		84
05	PORT A	TRISA	85
06	PORT B	TRISB	86
07			87
08	EEDATA	EECON1	88
09	EEADR	EECON2	89
0A	PCLATH		8A
0B	INTCON		8B
0C	36 регистров общего назначения		8C
2F		Тоже	AF
30	Не существует		B0
7F			FF

В регистре статуса (03h) есть биты выбора страниц, которые позволяют обращаться к четырем страницам будущих модификаций этого кристалла. Однако для

PIC16F84 память данных существует только до адреса 02Fh. Первые 12 адресов используются для размещения регистров специального назначения. Регистры с адресами 0Ch-2Fh могут быть использованы, как регистры общего назначения, которые представляют собой статическое ОЗУ. Некоторые регистры специального назначения продублированы на обеих страницах, а некоторые расположены на странице 1 отдельно. Когда установлена страница 1, то обращение к адресам 8Ch-AFh фактически адресует страницу 0. К регистрам можно адресоваться прямо или косвенно.

Прямая адресация.

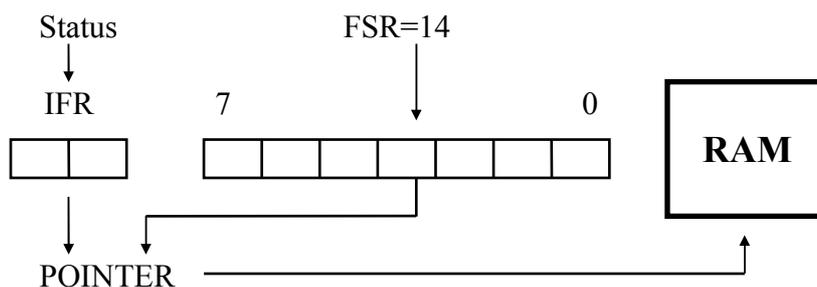
Когда производится прямая 9-битная адресация, младшие 7 бит берутся как прямой адрес из кода операции, а два бита указателя страниц (RP1,RP0) из регистра статуса (03h).



Косвенная адресация

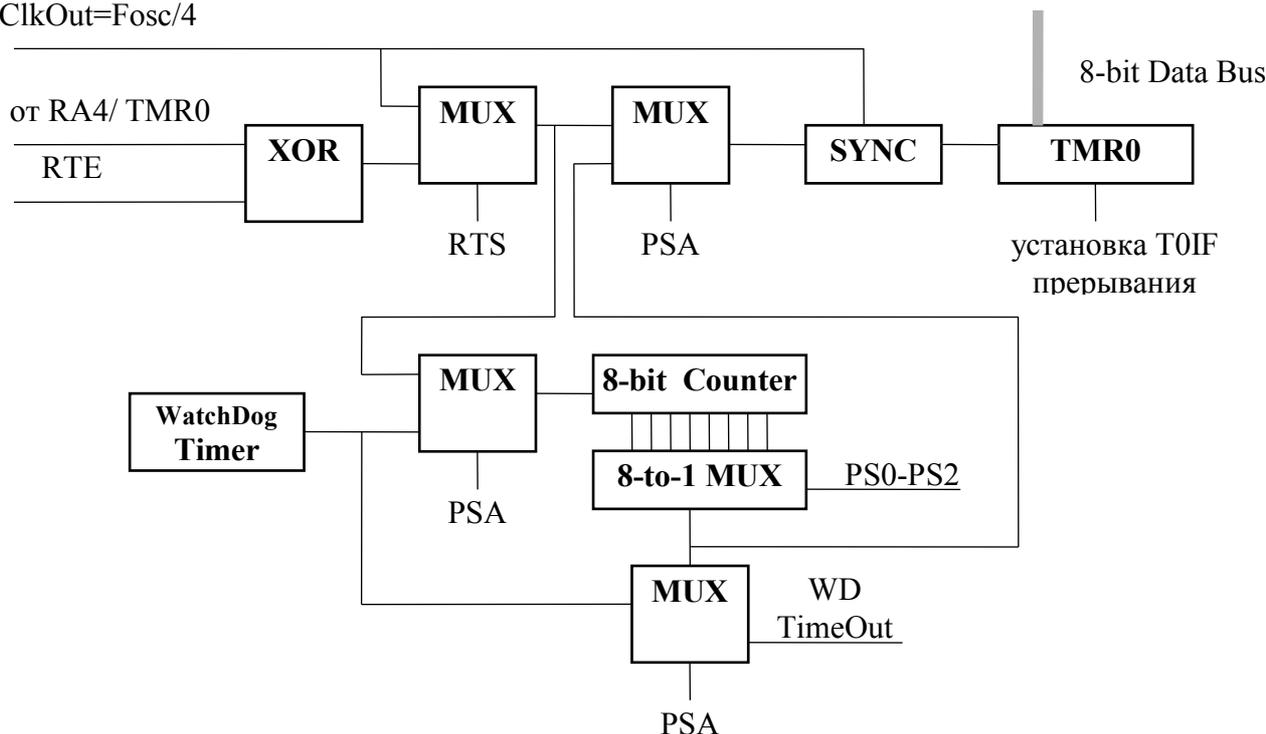
FSR (04h)- Указатель косв.адресации

Любая команда, которая использует INDF (адрес 00h) в качестве регистра фактически обращается к указателю, который хранится в FSR (04h). Чтение косвенным образом самого регистра INDF даст результат 00h. Запись в регистр INDF косвенным образом будет выглядеть как NOP, но биты статуса могут быть изменены. Необходимый 9-битный адрес формируется объединением содержимого 8-битного FSR регистра и бита IRP из регистра статуса.



TMR0 таймер/счетчик

$$\text{ClkOut} = \text{Fosc} / 4$$



Режим таймера выбирается путем сбрасывания в ноль бита T0CS, который находится в регистре OPTION_REG. В режиме таймера TMR0 будет инкрементироваться от ВНУТРЕННЕГО ИСТОЧНИКА частоты каждый командный цикл (без прескаллера). После записи информации в TMR0, инкрементирование его начнется после двух командных циклов. Такое происходит со всеми командами, которые производят запись или чтение-модификацию-запись TMR0 (напр. MOVF f1, CLRF f1). Избежать этого можно при помощи записи в TMR0 скорректированного значения. Если TMR0 нужно проверить на равенство нулю без остановки счета, следует использовать инструкцию MOVF f1,W. Режим счетчика выбирается путем установки в единицу бита T0CS, который находится в регистре OPTION_REG. В этом режиме TMR0 будет инкрементироваться либо положительным, либо отрицательным фронтом на ножке RA4/T0CKI от ВНЕШНЕГО источника. Направление фронта определяется управляющим битом T0SE в регистре OPTION_REG. При T0SE=0 будет выбран передний фронт. Прескаллер может быть использован или совместно с TMR0, или с Watchdog таймером. Вариант подключения делителя контролируется битом PSA в регистре OPTION_REG. При PSA=0 делитель будет подсоединен к TMR0. Содержимое делителя программе недоступно. Коэффициент деления - программируется. Прерывание по TMR0 вырабатывается тогда, когда происходит переполнение TMR0 таймера/счетчика при переходе от FFh к 00h. Тогда

устанавливается бит запроса T0IF в регистре INTCON<2>. Данное прерывание можно замаскировать битом T0IE в регистре INTCON<5>. Бит запроса T0IF должен быть сброшен программно при обработке прерывания. Прерывание по TMR0 не может вывести процессор из SLEEP, так как таймер в этом режиме отключен.

Проблемы с таймером

Проблемы могут возникнуть при счете внешних сигналов. Эти сигналы стробируются внутренним сигналом синхронизации, см. схему SYNC. Образуется некоторая задержка между фронтом входного сигнала и моментом инкрементирования TMR0. Стробирование производится после прескаллера. Выход прескаллера опрашивается дважды в течение каждого командного цикла, чтобы определить положительный и отрицательный фронты входного сигнала. Поэтому сигнал Psout должен иметь высокий и низкий уровень не менее двух периодов синхронизации.

Когда прескаллер не используется, Psout повторяет входной сигнал, поэтому требования к нему следующие:

$$Trth = \text{TMR0 high time} \geq 2 \text{ tosc} + 20\text{ns}$$

$$Trtl = \text{TMR0 low time} \geq 2 \text{ tosc} + 20\text{ns}.$$

Когда прескаллер используется, на вход TMR0 подается сигнал, поделенный на число, установленное в счетчике делителя. Сигнал после прескаллера всегда симметричен.

$$Psout \text{ high time} = Psout \text{ low time} = N * Trt / 2, \text{ где } Trt - \text{входной период TMR0,}$$

N- значение счетчика делителя (2,4...256).

В этом случае требования к входному сигналу можно выразить так:

$$N * Trt / 2 \geq 2 \text{ tosc} + 20\text{ns} \text{ или } Trt \geq (4\text{tosc} + 40\text{ns}) / N.$$

Когда используется прескаллер, низкий и высокий уровень сигнала на его входе должны быть не менее 10 нс. Таким образом общие требования к внешнему сигналу, когда делитель подключен таковы:

$$Trt = \text{TMR0 period} \geq (4\text{tosc} + 40\text{ns}) / N$$

$$Trth = \text{TMR0 high time} \geq 10\text{ns}$$

$$Trtl = \text{TMR0 low time} \geq 10\text{ns}.$$

Так как выход делителя синхронизируется внутренним сигналом тактовой частоты, то существует небольшая задержка между появлением фронта внешнего сигнала и временем фактического инкремента TMR0. Эта задержка находится в диапазоне между $3 \cdot t_{osc}$ и $7 \cdot t_{osc}$. Таким образом измерение интервала между событиями будет выполнено с точностью $4 \cdot t_{osc}$ (1 мкс при кварце 4 МГц).

Регистр статуса

Регистр статуса (03h) содержит арифметические флаги АЛУ, состояние контроллера при сбросе и биты выбора страниц для памяти данных. Регистр доступен для любой команды так же, как любой другой регистр. Однако, биты \overline{TO} и \overline{PD} устанавливаются аппаратно и не могут быть записаны в статус программно. Это следует иметь в виду при выполнении команды с использованием регистра статуса. Например, команда CLRF 03h обнулит все биты, кроме битов \overline{TO} и \overline{PD} , а затем установит бит Z=1. После выполнения этой команды регистр статуса может и не иметь нулевое значение (из-за битов \overline{TO} и \overline{PD}) 03h=000??100. Поэтому рекомендуется для изменения регистра статуса использовать только команды битовой установки BCF, BSF, MOVWF, которые не изменяют остальные биты статуса. Воздействие всех команд на биты статуса можно посмотреть в “Описании команд”.

Программные флаги статуса

Размещение флагов в регистре статуса следующее:

	b7	b6	b5	b4	B3	b2	b1	b0
03h=	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C

C - Флаг переноса/заема:

Для команд ADDWF и SUBWF. Этот бит устанавливается, если в результате операции из самого старшего разряда происходит перенос. Вычитание осуществляется путем прибавления дополнительного кода второго операнда. При выполнении команд сдвига этот бит всегда загружается из младшего или старшего бита сдвигаемого источника.

;-----Пример

;SUBWF Example #1

 clrf 0x20 ;f(20h)=0

```
movlf    1      ;wreg=1
subwf    0x20   ;f(20h)=f(20h)-wreg=0-1=FFh
          ;Carry=0: Результат отрицательный.
;SUBWF   Example #2
movlw    0xFF
movwf    0x20   ;f(20h)=FFh
clrw     ;wreg=0
subwf    0x20   ;f(20h)=f(20h)-wreg=FFh-0=FFh
          ;Carry=1: Результат положительный.
```

DC - Флаг десятичного переноса/заема:

Для команд ADDWF и SUBWF. Этот бит устанавливается, если в результате операции из четвертого разряда происходит перенос. Механизм установки десятичного бита переноса “DC” тот же самый, отличается тем, что отслеживается перенос из четвертого бита.

Z - Флаг нулевого результата:

Устанавливается, если результатом арифметической или логической операции является ноль.

\overline{PD} - Power Down (режим хранения данных) :

Устанавливается в “1” при включении питания или команде CLRWDT. Сбрасывается в “0” командой SLEEP.

\overline{TO} - Time Out. Флаг срабатывания Watchdog таймера:

Устанавливается в “1” при включению питания и командами CLRWDT, SLEEP. Сбрасывается в “0” по завершению выдержки времени таймера WDT.

RP1, RP0 - Биты выбора страницы памяти данных при прямой адресации

RP1,RP0:

00= Страница 0 (00h-7Fh),

01= Страница 1 (80h-FFh),

10= Страница 2 (100h-17Fh),

11= Страница 3 (180h-1FFh).

На каждой странице расположено 128 байт. В кристалле PIC16F84 используется только RP0. В этом кристалле RP1 может использоваться просто как бит общего назначения чтения/записи. Однако надо помнить, что в последующих разработках он будет использоваться.

IRP - Бит выбора страницы памяти данных при косвенной адресации

IRP:

0= Страницы 0,1 (00h-FFh),

1= Страница 2,3 (100h-1FFh).

Этот бит в кристалле PIC16F84 использовать не имеет смысла. Поэтому его можно использовать как бит общего назначения чтения/записи. Однако, надо помнить, что в последующих разработках он будет использоваться.

Аппаратные флаги статуса

Аппаратные Биты статуса \overline{TO} (Time Out) и \overline{PD} (Power Down).

По состоянию битов регистра статуса \overline{TO} и \overline{PD} , можно определить, чем был вызван “Сброс”:

- просто включением питания,
- срабатыванием таймера Watchdog,
- выходом из режима пониженного энергопотребления (Sleep) в результате срабатывания Watchdog таймера,
- по внешнему сигналу \overline{MCLR} .

На состояние этих битов могут повлиять только следующие события:

События	\overline{TO}	\overline{PD}	Примечание
Включение	1	1	
WDT таймер	0	X	Не воздействует на бит \overline{PD}
Sleep команда	1	0	
CLRWDT команда	1	1	

Примечание: Событие WDT timeout происходит независимо от состояния \overline{TO} бита. Sleep команда выполняется независимо от состояния \overline{PD} бита.

Ниже показаны состояния битов \overline{TO} и \overline{PD} после “Сброса”

\overline{TO}	\overline{PD}	“Сброс” был вызван следующими событиями
0	0	Выход из Sleep по завершению задержки Watchdog
0	1	Завершение задержки Watchdog
X	0	Выход из Sleep по внешнему сигналу \overline{MCLR}
1	1	Включение питания
X	X	“0”-импульс на входе \overline{MCLR}

Примечание: Биты \overline{TO} и \overline{PD} сохраняют текущее состояние до тех пор, пока не произойдет одно из событий перечисленных в таблице. “0” - импульс низкого уровня на входе кристалла \overline{MCLR} не изменяет состояние битов \overline{TO} и \overline{PD} .

Организация встроенного ПЗУ

Программный счетчик в PIC16F84 имеет ширину 13 бит и способен адресовать 8Кx14бит объема программной памяти. Однако, физически на кристалле имеется только 1Кx14 памяти (адреса 0000h-03FFh). Обращение к адресам выше 3FFh фактически есть адресация в тот же первый килобайт. Вектор сброса находится по адресу 0000h, вектор прерывания находится по адресу 0004h.

PC <12:0>	
Stack Level 1	
Stack Level 2	
.....	
Stack Level 8	
Reset Vector	0000h
Interrupt Vector	0004h
	0005h
On-Chip Program Memory	03FFh
	0400h
	1FFFh

EEPROM PIC16F84 рассчитан на ограниченное число циклов стирания/записи. Чтобы записать в программную память, кристалл должен быть переведен в специальный режим при котором на ножку \overline{MCLR} подается напряжение программирования V_{prg} , а питание V_{dd} должно находиться в пределах 4.5 В ...5.5В. PIC16F84 непригоден для применений, в которых часто модифицируется программа.

Запись в программную память осуществляется побитно, последовательно с использованием только двух ножек. Паспортное значение количества записей в программную память PIC16F84 равно 100(min) и 1000 (tip), практически-же микросхемы выдерживают несколько тысяч перезаписей. Число перезаписей памяти данных (64 байта) достигнет 1000000.

РС и адресация ПЗУ

(02h). Программный счетчик

Ширина программного счетчика - 13 бит. Младший байт программного счетчика (PCL) доступен для чтения и записи и находится в регистре 02h. Старший байт программного счетчика (PCH) не может быть прямым образом прочитан или записан. Старший байт программного счетчика может быть записан через PCLATH регистр, адрес которого 0Ah. В зависимости от того, загружается ли в программный счетчик новое значение во время выполнения команд CALL, GOTO, или в младший байт программного счетчика (PCL) производится запись, - старшие биты программного счетчика загружаются из PCLATH разными способами так, как показано на рисунке.

Случай команд GOTO, CALL



Случай команд, у которых результат помещается в 02h



Стек и возвраты из подпрограмм

Кристалл PIC16F84 имеет восьмиуровневый аппаратный стек шириной 13 бит. Область стека не принадлежит ни к программной области, ни к области данных, а указатель стека пользователю недоступен. Текущее значение программного счетчика посылается в стек, когда выполняется команда CALL или производится обработка

прерывания. При выполнении процедуры возврата из подпрограммы (команды RETLW, RETFIE или RETURN), в программный счетчик выгружается содержимое стека. Регистр PCLATH (0Ah) не изменяется при операциях со стеком.

Данные в EEPROM

Долговременная Память данных EEPROM.

Память данных EEPROM позволяет прочитать и записать байт информации. При записи байта автоматически стирается предыдущее значение и записываются новые данные (стирание перед записью). Все эти операции производит встроенный автомат записи EEPROM. Содержимое ячеек этой памяти сохраняется при выключении питания. Кристалл PIC16F84 имеет память данных 64x8 EEPROM бит, которая позволяет запись и чтение во время нормальной работы (во всем диапазоне питающих напряжений). Эта память не принадлежит области регистров ОЗУ. Доступ к ней осуществляется через два регистра: EEDATA (08h), который содержит в себе восьмибитовые данные для чтения/записи и EEADR (09h), который содержит в себе адрес ячейки к которой идет обращение. Дополнительно имеется два управляющих регистра: EECON1 (88h) и EECON2 (89h).

При считывании данных из памяти EEPROM необходимо записать требуемый адрес в EEADR регистр и затем установить бит RD EECON1<0> в единицу. Данные появятся в следующем командном цикле в регистре EEDATA и могут быть прочитаны. Данные в регистре EEDATA защелкиваются.

При записи в память EEPROM, необходимо сначала записать требуемый адрес в EEADR регистр и данные в EEDATA регистр. Затем выполнить специальную последовательность команд, производящую непосредственную запись:

```
movlw    55h
movwf    EECON2
movlw    AAh
movwf    EECON2
bsf      EECON1,WR    ;установить WR бит, начать запись
```

Во время выполнения этого участка программы, все прерывания должны быть запрещены для точного выполнения временной диаграммы. Время записи - примерно 10мс. Фактическое время записи будет изменяться в зависимости от напряжения,

температуры и индивидуальных свойств кристалла. В конце записи бит WR автоматически обнуляется, а флаг завершения записи EEIF, он же запрос на прерывание, устанавливается. Для предотвращения случайных записей в память данных предусмотрен специальный бит WREN в регистре EECON1. Рекомендуется держать бит WREN выключенным, кроме тех случаев, когда нужно обновить память данных. Более того, кодовые сегменты, которые устанавливают бит WREN и те, которые выполняют запись должны храниться на различных адресах, чтобы избежать случайного выполнения их обоих при сбое программы.

Управление EEPROM

Управляющие регистры для EEPROM

Название	Функция	Адрес	Значение после включения
EEDATA	EEPROM регистр данных	08h	XXXX XXXX
EEADR	EEPROM регистр адреса	09h	XXXX XXXX
EECON1	EEPROM 1 управляющий регистр	88h	0000 X000
EECON2	EEPROM 2 управляющий регистр	89h	-

Регистры EECON1 и EECON2.

Регистр EECON1 (адрес 88h) - это управляющий регистр шириной пять бит. Младшие пять бит физически существуют, а старшие три бита читаются всегда как "0"

Регистр EECON1 Адрес 88h Сброс при включении 0000X000

-	-	-	EEIF	WRERR	WREN	WR	RD
7	6	5	4	3	2	1	0

Управляющие биты RD и WR запускают соответственно чтение и запись. Они могут быть установлены только программно. Сбрасываются аппаратно по завершения операций чтения/записи. Запрет программного сброса бита WR предотвращает преждевременное окончание записи.

RD - Бит чтения.

RD =1 : Запускает чтение памяти данных EEPROM. Чтение занимает

один цикл. Устанавливается программно. Обнуляется аппаратно.

WR - Бит записи.

WR =1 : Запускает запись в память данных EEPROM.

Устанавливается программно. Обнуляется аппаратно.

WREN - Бит разрешения записи в память данных EEPROM.

WREN = 1: Разрешена запись.

WREN = 0: Запрещена запись.

После включения питания WREN обнуляется.

Флаг ошибки WRERR устанавливается, когда процесс записи прерывается сигналом сброса /MCLR или сигналом сброса от WDT таймера. Рекомендуем проверять этот флаг WRERR и при необходимости производить перезапись данных, данные и адрес которых сохраняются в регистрах EEDATA и EEADR.

WRERR - Флаг ошибки записи.

WRERR = 1: Флаг устанавливается, когда операция записи

преждевременно прерывается сигналом сброса /MCLR (во время обычного режима или режима SLEEP) или сигналом сброса WDT во время обычного режима.

EEIF - Флаг завершения записи.

EEIF = 1: Флаг устанавливается, когда завершена запись.

Соответствующий бит разрешения прерывания - EEIE в регистре INTCON.

Организация прерываний

Прерывания в PIC16F84 могут быть от четырех источников:

внешнее прерывание с ножки RB0/INT,

прерывание от переполнения счетчика/таймера TMR0,

прерывание по окончании записи данных в EEPROM,

прерывание от изменения сигналов на ножках порта RB<7:4>.

Все прерывания имеют один и тот же вектор/адрес - 0004h. Однако, в управляющем регистре прерываний INTCON записывается: от какого именно

источника поступил запрос прерывания. Записывается соответствующим битом-флагом. Такое прерывание может быть замаскировано индивидуально или общим битом. Единственным исключением является прерывание по концу записи в EEPROM. Этот флаг находится в другом регистре EECON1. Бит общего разрешения/запрещения прерывания GIE (INTCON <7>) разрешает (если=1) все индивидуально незамаскированные прерывания или запрещает (если=0) их. Каждое прерывание в отдельности может быть дополнительно разрешено/запрещено установкой/сбросом соответствующего бита в регистре INTCON.

Бит GIE обнуляется при сбросе. Когда начинает обрабатываться прерывание, бит GIE обнуляется, чтобы запретить дальнейшие прерывания, адрес возврата посылается в стек, а в программный счетчик загружается адрес 0004h. Время реакции на прерывание для внешних событий, таких как прерывание от ножки INT или порта В, составляет приблизительно пять циклов. Это на один цикл меньше, чем для внутренних событий, таких как прерывание по переполнению от таймера TMR0. Время реакции всегда одинаковое. В подпрограмме обработки прерывания источник прерывания может быть определен по соответствующему биту в регистре флагов. Этот флаг-бит должен быть программно сброшен внутри подпрограммы. Флаги запросов прерываний не зависят от соответствующих маскирующих битов и бита общего маскирования GIE.

Команда возврата из прерывания RETFIE завершает прерывающую подпрограмму и устанавливает бит GIE, чтобы опять разрешить прерывания.

Регистр запросов и масок

Управляющий регистр прерываний (INTCON) и его биты

Адрес: 0Bh Значение при сбросе - 0000 0000

GIE	EEIE	RTIE	INIE	RBIE	RTIF	INTF	RBIF
-----	------	------	------	------	------	------	------

RBIF -Флаг прерывания от изменения на порту RB.

Флаг устанавливается, когда сигнал на входе RB<7:4> изменяется.

Флаг сбрасывается программным способом.

INTF - Флаг прерывания INT.

Флаг устанавливается, когда на ножке INT появляется сигнал от

внешнего источника прерывания. Сбрасывается программным способом.

T0IF - Флаг прерывания от переполнения TMR0.

Флаг устанавливается, когда TMR0 переполняется.

Флаг сбрасывается программным способом.

RBIE - Бит разрешения/запрещения RBIF прерывания.

RBIE = 0 : запрещает RBIE прерывание.

RBIE = 1 : разрешает RBIE прерывание.

INTE - Бит разрешения/запрещения INT прерывания.

INTE = 0 : запрещает INT прерывание.

INTE = 1 : разрешает INT прерывание.

T0IE Бит разрешения/запрещения T0IF прерывания.

T0IE = 0 : запрещает T0IF прерывание.

T0IE = 1 : разрешает T0IF прерывание.

EEIE - Бит разрешения/запрещения прерывания EEPROM записи.

EEIE = 0 : запрещает EEIF прерывание.

EEIE = 1 : разрешает EEIF прерывание.

GIE Бит разрешения/запрещения всех прерываний.

GIE = 0 : запрещает прерывания

GIE = 1 : разрешает прерывания

Он сбрасывается автоматически при следующих обстоятельствах:

по включению питания,

по внешнему сигналу \overline{MCLR} при нормальной работе,

по внешнему сигналу \overline{MCLR} в режиме SLEEP,

по окончанию задержки таймера WDT при нормальной работе,

по окончанию задержки таймера WDT в режиме SLEEP.

Внешнее прерывание

Внешнее прерывание на ножке RB0/INT осуществляется по фронту: либо по нарастающему (если бит INTEDG=1 в регистре OPTION_REG), либо по спадающему фронту (если INTEDG=0). Когда фронт обнаруживается на ножке INT, то бит запроса INTF устанавливается (INTCON <1>). Это прерывание может быть замаскировано установкой управляющего бита INTE в ноль (INTCON <4>). Бит запроса INTF должен быть очищен прерывающей программой перед тем, как опять разрешить это прерывание. Прерывание INT может вывести процессор из режима SLEEP, если перед входом в этот режим бит INTE был установлен в единицу. Состояние бита GIE также определяет: будет ли процессор переходить на подпрограмму прерывания после просыпания из режима SLEEP.

Прерывание от TMR0

Переполнение счетчика TMR0 (FFh->00h) установит бит запроса T0IF (INTCON<2>). Это прерывание может быть разрешено/запрещено установкой/сбросом бита маски T0IE (INTCON<5>).

Сброс запроса T0IF - дело программы обработки.

Прерывание от порта RB

Любое изменение сигналов на четырех входах порта RB<7:4> установит бит RBIF (INTCON<0>). Это прерывание может быть разрешено/запрещено установкой/сбросом бита маски RBIE (INTCON<3>). Сброс запроса RBIF - дело программы обработки.

Прерывание от EEPROM

Флаг запроса прерывания по окончании записи в EEPROM, EEIF (EECON1<4>) устанавливается по окончании автоматической записи данных в EEPROM. Это прерывание может бОыть замаскировано сбросом бита EEIE (INTCON<6>).

Сброс запроса EEIF - дело программы обработки.

Обзор регистров/портов

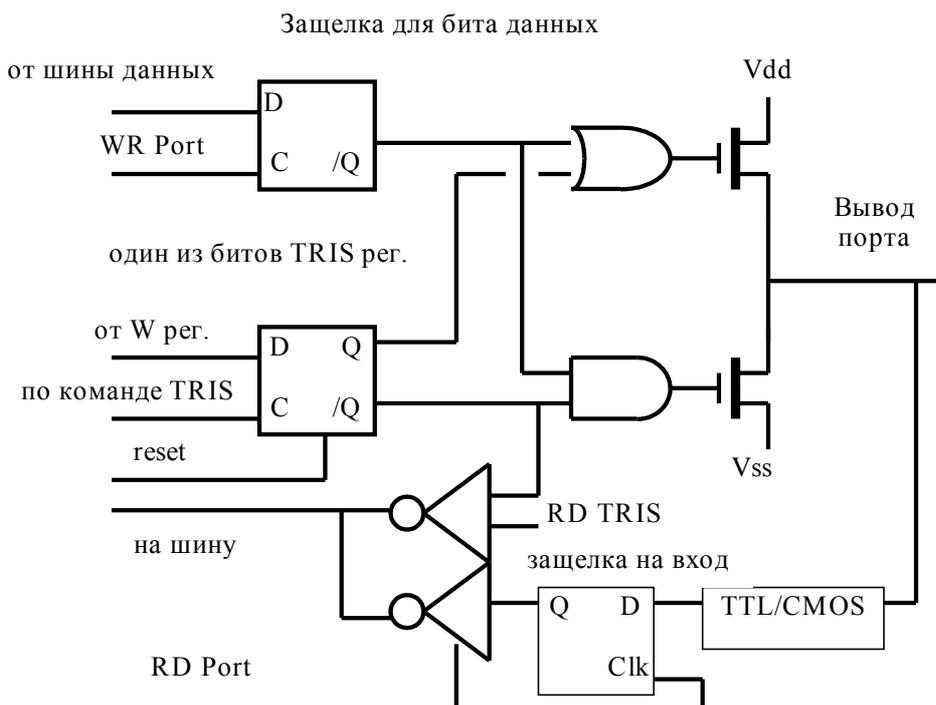
Кристалл имеет два порта: 5 бит порт RA и 8 бит порт RB с побитовой индивидуальной настройкой на ввод или на вывод.

Схема линии порта А

Порт А - это порт шириной 5 бит, соответствующие ножки кристалла RA<4:0>. Линии RA<3:0> двунаправленные, а линия RA4 - выход с открытым стоком. Адрес регистра порта А - 05h. Относящийся к порту А управляющий регистр TRISA расположен на первой странице регистров по адресу 85h. TRISA<4:0> - это регистр шириной 5 бит. Если бит управляющего TRISA регистра имеет значение единица, то соответствующая линия будет устанавливаться на ввод. Ноль переключает линию на вывод и одновременно выводит на нее содержимое соответствующего регистра защелки.

Название ножки	#	Функция ножки	Другие функции
RA0	0	Порт ввода/вывода. Входные уровни TTL.	
RA1	1	Порт ввода/вывода. Входные уровни TTL.	
RA2	2	Порт ввода/вывода. Входные уровни TTL.	
RA3	3	Порт ввода/вывода. Входные уровни TTL..	
RA4/T0CKI	4	Порт ввода/вывода. Выход-открытый коллектор. Вход - триггер Шмитта.	Вход внешнего тактового сигнала для TMR0

Ниже дана схема портов RA0..RA3



Все биты портов имеют защитные диоды, подключенный к Vdd и Vss.

Вывод RA4/T0CKI имеет несколько другую схему. Она приведена ниже

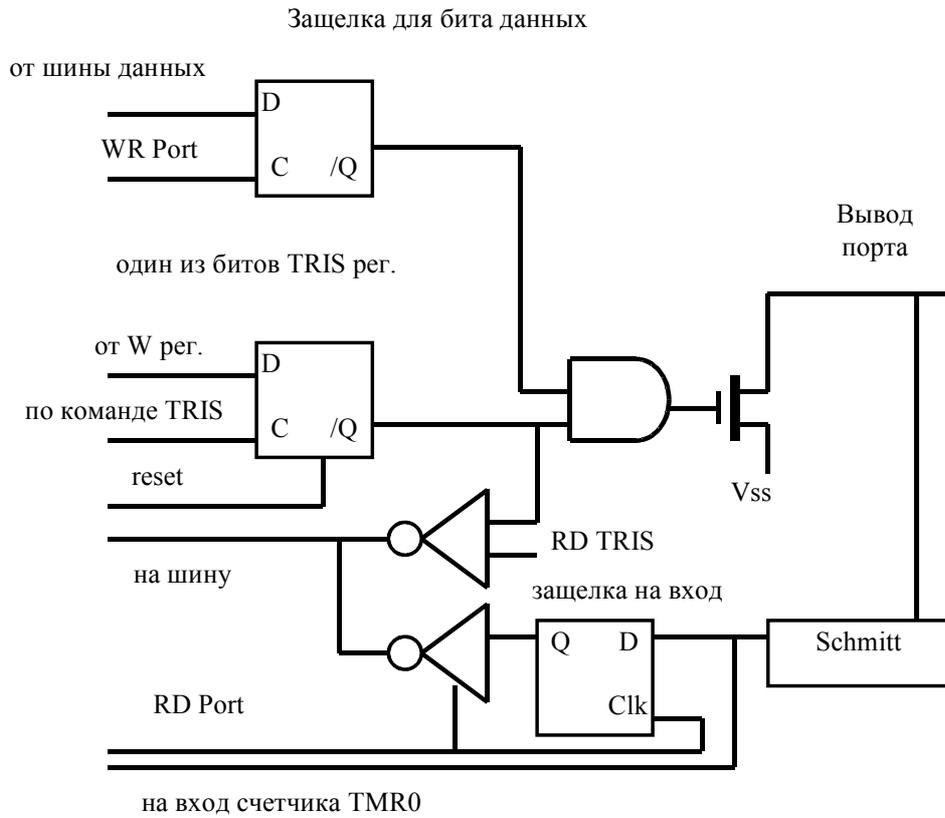


Схема линии порта В

Порт В - это двунаправленный порт, шириной в восемь бит (адрес регистра 06h). Относящийся к порту В управляющий регистр TRISB расположен на первой странице регистров по адресу 86h. Если бит управляющего TRISB регистра имеет значение единица, то соответствующая линия будет устанавливаться на ввод. Ноль переключает линию на вывод и одновременно выводит на нее содержимое соответствующего регистра защелки. У каждой ножки порта В имеется небольшая активная нагрузка (около 100мкА) на линию питания. Она автоматически отключается, если эта ножка запрограммирована как вывод. Более того, управляющий бит \overline{RPBU} OPTION_REG <7> может отключить ($\overline{RPBU}=1$) все нагрузки. Сброс при включении питания также отключает все нагрузки. Четыре линии порта В (RB<7:4>) имеют способность вызвать прерывание при изменении значения сигнала на любой из них. Если эти линии настроены на ввод, то они опрашиваются и защелкиваются в цикле чтения Q1. Новая величина входного сигнала сравнивается со старой в каждом командном цикле. При несовпадении значения сигнала на ножке и в защелке, генерируется высокий уровень. Выходы детекторов “несовпадений” RB4, RB5, RB6, RB7 объединяются по ИЛИ и генерируют прерывание RBIF (запоминаемое в INTCON<0>). Любая линия, настроенная как вывод, не участвует в этом сравнении. Прерывание может вывести кристалл из режима SLEEP. В подпрограмме обработки прерывания следует сбросить запрос прерывания одним из следующих способов:

- 1) Запретить прерывания при помощи обнуления бита RBIE INTCON<3>.
- 2) Прочитать порт В. Это завершит состояние сравнения.
- 3) Обнулить бит RBIF INTCON<0>.

Прерывание по несовпадению и программно устанавливаемые внутренние активные нагрузки на этих четырех линиях могут обеспечить простой интерфейс например с клавиатурой, с выходом из режима SLEEP по нажатию клавиш.

Ножка RB0 совмещена с входом внешнего прерывания INT.

Название ножки	#	Функция ножки	Другие функции
RB0	0	Порт ввода/вывода. Входные уровни TTL и внутренняя программируемая активная нагрузка.	Вход внешнего прерывания

Название ножки	#	Функция ножки	Другие функции
RB1	1	Порт ввода/вывода. Входные уровни ТТЛ и внутренняя программируемая активная нагрузка.	
RB2	2	Порт ввода/вывода. Входные уровни ТТЛ и внутренняя программируемая активная нагрузка.	
RB3	3	Порт ввода/вывода. Входные уровни ТТЛ и внутренняя программируемая активная нагрузка.	
RB4	4	Порт ввода/вывода. Входные уровни ТТЛ и внутренняя программируемая активная нагрузка.	Прерывание при изменении
RB5	5	Порт ввода/вывода. Входные уровни ТТЛ и внутренняя программируемая активная нагрузка.	Прерывание при изменении
RB6	6	Порт ввода/вывода. Входные уровни ТТЛ и внутренняя программируемая активная нагрузка.	Прерывание при изменении
RB7	7	Порт ввода/вывода. Входные уровни ТТЛ и внутренняя программируемая активная нагрузка.	Прерывание при изменении

Проблемы с портами

Проблемы при организации двунаправленных портов

Некоторые команды внутренне выполняются как чтение+запись. Например, команды BCF и BSF считывают порт целиком, модифицируют один бит и выводят результат обратно. Здесь необходима осторожность. Например, команда BSF для бита 5 регистра 06h (PORTB) сначала считывает все восемь бит. Затем выполняются действия над битом 5 и новое значение байта целиком записывается в выходные защелки. Если другой бит регистра 06h используется в качестве двунаправленного ввода/вывода (скажем бит 0) и в данный момент он определен как входной, входной сигнал на этой ножке будет считан и записан обратно в выходную защелку этой-же ножки, затирая ее предыдущее состояние. До тех пор пока эта ножка остается в режиме ввода, никаких проблем не возникает. Однако, если позднее линия 0 переключится в режим вывода, ее состояние будет неопределенным.

На ножку, работающую в режиме вывода, не должны “наваливаться” внешние источники токов (“монтажное И”, “монтажное ИЛИ”). Результирующие большие токи могут повредить кристалл.

Последовательное обращение к портам ввода/вывода.

Запись в порт вывода происходит в конце командного цикла. Но при чтении, данные должны быть стабильны в начале командного цикла. Будьте внимательны в операциях чтения, следующих сразу за записью в тот же порт. Здесь надо учитывать инерционность установления напряжения на выводах. Может потребоваться программная задержка, чтобы напряжение на ножке (зависит от нагрузки) успело стабилизироваться до начала исполнения следующей команды чтения.

Обзор команд и обозначения

Каждая команда PIC16F84 - это 14-битовое слово, которое разделено по смыслу на следующие части: - 1. код операции, -2. поле для одного и более операндов, которые могут участвовать или нет в этой команде. Система команд PIC16F84 включает в себя байт-ориентированные команды, бит-ориентированные, операции с константами и команды передачи управления.

Для байт-ориентированных команд “f” обозначает собой регистр, с которым производится действие; “d” - бит определяет, куда положить результат. Если “d”=0, то результат будет помещен в W регистр, при “d”=1 результат будет помещен в “f”, упомянутом в команде. Для бит-ориентированных команд “b” обозначает номер бита, участвующего в команде, а “f” -это регистр, в котором этот бит расположен.

Для команд передачи управления и операций с константами, “k” обозначает восьми или одиннадцатибитную константу.

Все команды выполняются в течение одного командного цикла. В двух случаях исполнение команды занимает два командных цикла: -1. проверка условия и переход, -2.изменение программного счетчика как результат выполнения команды. Один командный цикл состоит из четырех периодов генератора. Таким образом, для генератора с частотой 4 МГц время исполнения командного цикла будет 1 мкс.

Байт ориентированные команды

Мнемокод		Название команды	Флаги	Примеч.
ADDWF	f,d	Сложение W с f	C,DC,Z	2,3
ANDWF	f,d	Логическое И W и f	Z	2,3
CLRF	f	Сброс регистра f	Z	3
CLRWF		Сброс регистра W	Z	
COMF	f,d	Инверсия регистра f	Z	2,3
DECF	f,d	Декремент регистра f	Z	2,3
DECFSZ	f,d	Декремент f, пропустить команду, если 0		2,3
INCF	f,d	Инкремент регистра f	Z	2,3
INCFZ	f,d	Инкремент регистра f, пропустить команду, если 0		2,3
IORWF	f,d	Логическое ИЛИ W и f	Z	2,3
MOVF	f,d	Пересылка регистра f	Z	2,3
MOVWF	f	Пересылка W в f		3
NOP		Холостая команда		
RLF	f,d	Сдвиг f влево через перенос	C	2,3
RRf	f,d	Сдвиг f вправо через перенос	C	2,3
SUBWF	f,d	Вычитание W из f	C,DC,Z	2,3
SWAPF	f,d	Обмен местами тетрад в f		2,3
XORWF	f,d	Исключающее ИЛИ W и f	Z	2,3
ADDLW	k	Сложение константы с W.	C,DC,Z	
ANDLW	k	Логическое И константы и W	Z	
IORLW	k	Логическое ИЛИ константы и W	Z	
SUBLW	k	Вычитание W из константы.	C,DC,Z	
MOVLW	k	Пересылка константы в W		
XORLW	k	Исключающее ИЛИ константы и W	Z	
OPTION		Загрузка W в OPTION REG регистр		1
TRIS	F	Загрузка TRIS регистра		1

Бит ориентированные команды

Мнемокод		Название команды	Флаги	Примеч.
BCF	f,d	Сброс бита в регистре f		2,3
BSF	f,d	Установка бита в регистре f		2,3
BTFSF	f,b	Пропустить команду, если бит равен 0		
BTFSB	f,b	Пропустить команду, если бит равен 1		

Переходы

Мнемокод		Название команды	Флаги	Примеч.
CALL	k	Вызов подпрограммы		.
CLRWDT		Сброс Watchdog таймера	TO,PD	

Мнемокод		Название команды	Флаги	Примеч
GOTO	k	Переход по адресу		.
RETLW	k	Возврат из подпрограммы с загрузкой константы в W		
RETFIE		Возврат из прерывания.		
RETURN		Возврат из подпрограммы.		
SLEEP		Переход в режим SLEEP	TO,PD	

Замечания и пояснения

Примечание 1: Команды TRIS и OPTION помещены в перечень команд для совместимости с семейством PIC16C5X. Их использование не рекомендуется. В PIC16F84 регистры TRIS и OPTION доступны для чтения и записи как обычные регистры с номером. Предупреждаем, что эти команды могут не поддерживаться в дальнейших разработках PIC16CXX.

Примечание 2: Когда модифицируется регистр ввода/вывода, например MOVF 6,1, значение, используемое для модификации считывается непосредственно с ножек кристалла. Если значение защелки вывода для ножки, запрограммированной на вывод равно “1”, но внешний сигнал на этом выводе “0” из-за “навала” снаружи, то будет считываться “0”.

Примечание 3: Если операндом этой команды является регистр f1 (и, если допустимо, d=1), то делитель, если он подключен к TMR0, будет обнулен.

Условия сброса

В PIC16F84 существуют различия между вариантами сбросов:

- 1) Сброс по включению питания.
- 2) Сброс по внешнему сигналу \overline{MCLR} при нормальной работе.
- 3) Сброс по внешнему сигналу \overline{MCLR} в режиме SLEEP.
- 4) Сброс по окончанию задержки таймера WDT при нормальной работе.
- 5) Сброс по окончанию задержки таймера WDT в режиме SLEEP.

Некоторые из спец.регистров при сбросе не инициализируются. Они имеют случайное состояние при включении питания и не изменяются при других видах сбросов. Другая часть спец.регистров инициализируются в “состояние сброса” при всех видах сброса, кроме сброса по окончанию задержки таймера WDT в режиме

SLEEP. Просто этот сброс рассматривается как временная задержка в нормальной работе. Есть еще несколько исключений. Программный счетчик всегда сбрасывается в ноль (0000h). Биты статуса \overline{TO} и \overline{PD} устанавливаются или сбрасываются в зависимости от варианта сброса. Эти биты используются программой для определения природы сброса. Их значения после сброса приведены в таблице.

Состояние регистров после сброса представлено в таблице. Там приняты обозначения: u -не изменяется, x -неизвестно, - не исполняется, читается как '0'. ? - будет установлен по окончанию записи в EEPROM

W	-	Xxxx xxxx	uuuu uuuu				
INDF	00h						
TMR0	01h	Xxxx xxxx	uuuu uuuu				
PCL	02h	0000	0000	PC+1	0000	0000	PC+1
STATUS	03h	0001 1xxx	0000 1uuu	Uuu0 0uuu	000u uuuu	0001 0uuu	uuu1 0uuu
FSR	04h	Xxxx xxxx	uuuu uuuu				
PORT A	05h	Xxxx xxxx	uuuu uuuu				
PORT B	06h	Xxxx xxxx	uuuu uuuu				
TRIS A	85h	---1 ---1	---1 1111	---u uuuu	---1 1111	---1 1111	---u uuuu
TRIS B	86h	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111	uuuu uuuu
OPTION_	81h	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111	uuuu uuuu
REG							
EEDATA	08h	Xxxx xxxx	uuuu uuuu				
EEADR	09h	Xxxx xxxx	uuuu uuuu				
EECON1	88h	---0 0000	---0 ?000	---u uuuu	---0 ?000	---0 ?000	---u uuuu
EECON2	89h						
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 0000	uuuu uuuu	0000 000u	0000 0000	uuuu uuuu

Алгоритм сброса при включении питания

Кристалл PIC16F84 имеет встроенный детектор включения питания. Таймер запуска начинает счет выдержки времени после того, как напряжение питания пересекло уровень около 1,2...1,8 Вольт. По истечении выдержки около 72мс считается, что напряжение достигло номинала и запускается другой таймер выдержка на стабилизацию кварцевого генератора. Программируемый бит конфигурации позволяет разрешать или запрещать выдержку от встроенного таймера запуска. Выдержка запуска меняется от экземпляров кристалла, от питания и температуры. См. характеристики постоянного тока.

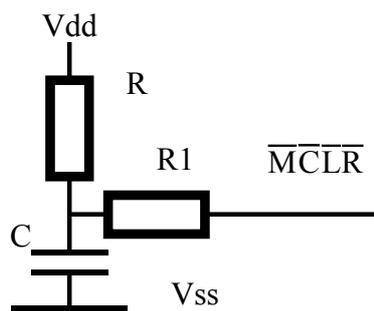
Таймер на стабилизацию генератора отсчитывает 1024 импульса от начавшего работу генератора. Считается, что кварцевый генератор за это время

вышел на режим. При использовании RC генераторов- выдержка на стабилизацию не производится.

Затем включается таймер ожидания внешнего сброса \overline{MCLR} . Это необходимо для тех случаев, когда требуется синхронно запустить в работу несколько PIC контроллеров через общий для всех сигнал \overline{MCLR} . Если такого сигнала не поступает, то через время T_{ost} вырабатывается внутренний сигнал сброса и контроллер начинает ход по программе. Время T_{ost} программируется битами конфигурации в EEPROM.

Здесь существует проблема, когда V_{dd} нарастает слишком медленно и все выдержки на запуск, а питание еще не достигло своего минимального значения V_{dd} (min) работоспособности. В таких случаях рекомендуем использовать внешние RC цепочки для сброса по \overline{MCLR} .

Ниже приведена такая цепочка



Здесь можно применить диод для быстрого разряда конденсатора при выключении питания. Рекомендуется резистор $R < 40 \text{ кОм}$, тогда на нем будет падать не более 0,2В. Резистор $100 \text{ Ом} < R1 < 1 \text{ кОм}$ ограничит ток в ножку \overline{MCLR} .

Watch Dog таймер

Watchdog таймер представляет собой полностью независимый встроенный RC генератор, который не требует никаких внешних цепей. Он будет работать, даже если основной генератор остановлен, как это бывает при исполнении команды SLEEP. Таймер вырабатывает сигнал сброса. Выработка таких сбросов может быть запрещена путем записи нуля в специальный бит конфигурации WDTE. Эту операцию производят на этапе прожига микросхем.

Выдержка времени WDT

Номинальная выдержка WDT составляет 18 мс (без использования делителя). Она зависит от температуры, напряжения питания, от особенностей типов микросхем. Если требуются большие задержки, то к WDT может быть подключен встроенный делитель с коэффициентом деления до 1:128; который программируется путем записи в регистр OPTION_REG. Здесь могут быть реализованы выдержки до 2.5 секунд. Команды “CLRWDТ” и “SLEEP” обнуляют WDT и делитель, если он подключен к WDT. Это запускает выдержку времени сначала и предотвращает на некоторое время выработку сигнала сброс. Если сигнал сброса от WDT все же произошел, то одновременно обнуляется бит “ \overline{TO} ” в регистре статуса (f3). В приложениях с высоким уровнем помех, содержимое регистра OPTION_REG подвержено сбою. Поэтому регистр OPTION_REG должен обновляться через равные промежутки времени. Следует учесть, что наихудшей комбинацией является: Vdd=min, температура=max и max коэффициент деления делителя - это приводит к самой большой выдержке времени, она может достигать нескольких секунд.

Типы генераторов

Кристаллы PIC16F84 могут работать с четырьмя типами встроенных генераторов. Пользователь может запрограммировать два конфигурационных бита (FOSC1 и FOSC0) для выбора одного из четырех режимов: RC, LP, XT, HS. Кристаллы PIC16... могут также тактироваться и от внешних источников. Генератор, построенный на кварцевых или керамических резонаторах, требует периода стабилизации после включения питания. Для этого, встроенный таймер запуска генератора держит устройство в состоянии сброса примерно 18 мс после того, как сигнал на \overline{MCLR} ножке кристалла достигнет уровня логической единицы. Таким образом, внешняя цепочка RC, связанная с ножкой \overline{MCLR} во многих случаях не требуется.

Встроенные генераторы работоспособны при определенных номиналах питающего напряжения:

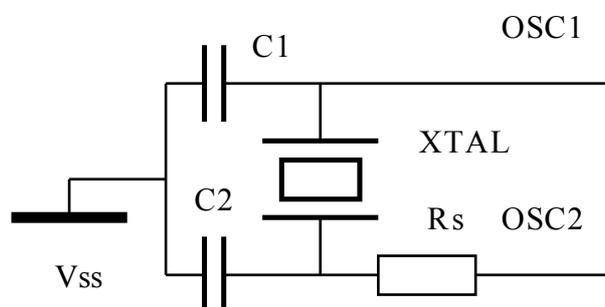
Vdd	OSC mode	Max Freq
2..3 V	RC	2 mHz
	LP	200 kHz
3..6 V	RC	4 mHz

Vdd	OSC mode	Max Freq
	XT,LP	200 kHz
4,5...5,5 V	HS	10 MHz

ВНИМАНИЕ! При частотах ниже 500 кГц, внутренний генератор может генерировать сбойный импульс на гармониках, когда переключается бит 0 порта А. Этого не происходит при использовании внешнего генератора или при встроенном RC генераторе.

Генератор на кварцах

PIC16F84-XT, -HS или -LP требуют подключения кварцевого или керамического резонатора к выводам OSC1 и OSC2.



Маркировка следующая:

XT - стандартный кварцевый генератор,

HS - высокочастотный кварцевый генератор,

LP - низкочастотный генератор для экономичных приложений.

Резистор Rs может потребоваться для генератора "HS", особенно при частотах ниже 20 МГц для гашения гармоник. Он также может потребоваться в режиме XT с резонатором типа AT strip-cut.

Выбор конденсаторов для керамического резонатора.

Тип генератора	Частота резонатора	Конденсатор C1=C2
XT	455 kHz	150-330 pF

Тип генератора	Частота резонатора	Конденсатор C1=C2
	2 MHz	20-330 pF
	4 MHz	20-330 pF
HS	10 MHz	20-200 pF

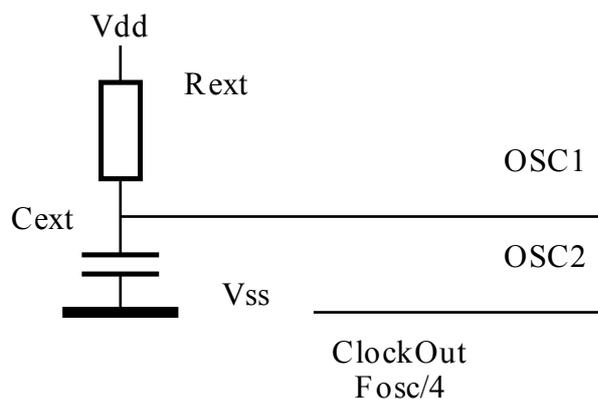
Выбор конденсатора для кварцевого генератора

Тип генератора	Частота	Конденсатор C1	Конденсатор C2
LP	32 kHz	30 pF	30 - 50 pF
	100 kHz	15 pF	15 pF
	200 kHz	0-15 pF	0-15 pF
XT	100 kHz	15-30 pF	200-300 pF
	200 kHz	15-30 pF	100-200 pF
	455 kHz	15-30 pF	15-100 pF
	1 MHz	15-30 pF	15-30 pF
	2 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	10 MHz	15 pF	15 pF

Более высокая емкость будет увеличивать стабильность генератора, но также будет увеличивать время запуска. Значения приведены для ориентировки. В режимах HS и XT, чтобы избежать гармоник может потребоваться последовательный резистор Rs.

RC генератор.

Когда не предъявляются требования к быстродействию и к точности по времени, ОTR кристалл, например PIC16F84-RC, позволяет сэкономить деньги и реализовать простой RC



Частота есть функция питающего напряжения, значений резистора R_{ext} , конденсатора C_{ext} и температуры. Кроме того, частота генератора будет незначительно изменяться от партии к партии. На частоту генерации влияет собственная емкость корпуса кристалла, ее влияние заметно для малых значений C_{ext} . Нужно принять во внимание также дрейф R и C элементов.

Для значений R_{ext} ниже 2.2 кОм генератор может работать нестабильно или не заводиться. При очень больших значениях R_{ext} (напр. 1 МОм) генератор становится чувствительным к помехам, влажности и монтажным утечкам тока. Рекомендуемая величина R_{ext} находится между 5 кОм и 100 кОм. Хотя генератор работоспособен и при отсутствии внешнего конденсатора ($C_{ext} = 0$), мы рекомендуем использовать емкость более 20 пФ для увеличения стабильности работы. С малой C_{ext} , или вообще без нее, частота генератора сильно зависит от монтажных емкостей. Разброс будет тем больше, чем больше величина R (так как влияние токов утечки на частоту RC генератора сильнее при больших R) и чем меньше величина C (так как в этом случае сильнее проявляется влияние монтажных емкостей). Сигнал с частотой генератора, деленной на 4, присутствует на ножке OSC2/CLKOUT, и может быть использован для целей тестирования или синхронизации других схем.

Внешнее возбуждение



Регистр *OPTION_REG*

Регистр *OPTION_REG* (адрес 81h) доступен для чтения и записи и содержит различные управляющие биты, которые определяют конфигурацию делителя, куда он подключен: к *TMR0* или *WDT*, знак фронта внешнего прерывания *INT* и внешнего сигнала для *TMR0*, подключение активной нагрузки на порту *RB*.

Регистр *OPTION_REG* Адрес 81h

Значение при включении питания= FFh

7	6	5	4	3	2	1	0
\overline{RPBU}	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

настройки делителя

PS2-PS0	Коэффициенты деления <i>TMR0</i>	Коэффициенты деления <i>WDT</i>
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

PSA - бит, подключающий делитель к

TMR0 (0) ,

WDT (1).

T0SE- фронт переднего сигнала *TMR0*

0- инкремент по нарастающему фронту на ножке *TMR0*,

1- инкремент по спадающему фронту на ножке TMR0.

T0CS - источник сигнала для TMR0

0- сигнал от внутреннего генератора,

1- внешний сигнал на ножке TMR0.

INTEDG - фронт сигнала INT

0- прерывание по спадающему фронту на ножке INT,

1- прерывание по нарастающему фронту на ножке INT.

\overline{RBPU} - инверсный бит подключения активной нагрузки к порту В

0- нагрузки подключены по алгоритму работы порта В,

1- нагрузки отключены.

Подключения делителя частоты

Один и тот же восьмибитный счетчик может быть включен либо перед TMR0 либо после Watchdog таймера. Отметим, что делитель работает только с одним из этих устройств. Повторяем, если делитель работает с TMR0, это значит, что в данный момент он не работает с Watchdog таймером и наоборот. Схему включения счетчика (см. в разделе TMR0: TMR0).

Биты PSA и PS0-PS2 в регистре OPTION_REG определяют, с каким устройством работает делитель и настройку коэффициента деления. Когда делитель подключен к TMR0, все команды, которые записывают в TMR0 (напр., CLRF 16 MOVWF 1, BSF 1,x... и т.д.) будут обнулять делитель. Когда он подключен к Watchdog таймеру, то делитель вместе с Watchdog таймером будет обнулять команда CLRWDT. Содержимое делителя программе недоступно.

Подключение делителя - программно управляемое. Ниже представлен фрагмент программы переключения с TMR0 на WDT

1. `MOVLW B'xx0x0xxx'` ;выбрать внутреннюю синхронизацию и новое значение для делителя. Если новое значение делителя равно `000` или `001`,
2. `OPTION` ;то надо временно выбрать другое

- ;значение делителя.
3. CLRWF 1 ;обнулить TMR0 и делитель.
 4. MOVLW В`xxxx1xxx` ;выбрать WDT, не изменяя значения
;делителя.
 5. OPTION
 6. CLRWDT ;обнулить WDT и делитель.
 7. MOVLW В`xxxx1xxx` ;выбрать новое значение для делителя.
 8. OPTION

Пункты 1 и 2 требуются только тогда, когда к TMR0 был подключен внешний источник импульсов. Пункты 7 и 8 требуются тогда, когда в коэффициент деления надо установить `000` или `001`.

Переключение делителя с WDT на TMR0

1. CLRWDT ;обнулить WDT и делитель.
2. MOVLW В`xxxx0xxx` ;выбрать TMR0, новое значение для делителя
;источник сигнала.
3. OPTION

Указанными программами следует пользоваться, даже если WDT запрещен.

Конфигурационное слово

Кристалл PIC16F84 имеет пять битов конфигурации, которые хранятся в EEPROM и устанавливаются на этапе программирования кристалла. Эти биты могут быть запрограммированы (читается как `0`) или оставлены незапрограммированными (читается `1`) для выбора подходящего варианта конфигурации устройства. Эти биты расположены в EEPROM памяти по адресу 2007h. Пользователю следует помнить, что этот адрес находится ниже области кодов и недоступен программе.

Ячейка EEPROM конфигурации.

13	4	3	2	1	0	бит.адрес
----	---	---	---	---	---	-----------

CP	PWRTE	WDTE	FOSC1	FOSC0	2007h
----	-------	------	-------	-------	-------

FOSC0, FOSC1 - биты выбора типа генератора

00	LP генератор
01	XT генератор
10	HS генератор
11	RS генератор

WDTE - бит разрешения работы WatchDog Timer

1	- разрешен
0	- запрещен

PWRTE - бит разрешения выдержки времени после включения питания

1	- выдержка есть
0	- выдержки нет

Индивидуальная метка

Кристалл PIC16F84 имеет четыре слова, расположенные по адресу (2000h-2003h) Они предназначены для хранения идентификационного кода (ID) пользователя, контрольной суммы или другой информации. Как и слово конфигурации, они могут быть прочитаны или записаны только с помощью программатора. Доступа по программе к ним нет.

Если кристалл защищен, пользователю рекомендуется использовать для идентификации только младшие семь бит каждого ID слова, а в старший бит записывать `0`. Тогда ID слова можно будет прочитать даже в защищенном варианте.

Защита программ от считывания

Программный код, который записан в кристалл, может быть защищен от считывания при помощи установки бита защиты (CP) в слове конфигурации в ноль. Содержимое программы не может быть прочитано так, что с ним можно было бы работать. Кроме того, при установленном бите защиты становится невозможным изменять программу. Тоже относится и к содержимому памяти данных EEPROM.

Если установлена защита, то бит CP можно стереть только вместе с содержимым кристалла. Сначала будет стерта EEPROM программная память и память данных и в последнюю очередь бит защиты кода CP.

Проверка кристалла с установленной защитой.

При считывании защищенного кристалла, чтение любого адреса памяти даст результат, похожий на 0000000XXXXXXX (двоичный код), где X- это 0 или 1. Чтобы проверить сохранность памяти в защищенном кристалле, следуйте правилам:

- запрограммируйте и проверьте работу исправного кристалла.
- установите защиту кода программы и считайте содержимое программной памяти в файл-эталон.
- проверяйте любой защищенный кристалл путем сравнения его программной памяти с содержимым этого эталона.

Память данных EEPROM не может быть проверена после установки бита защиты.

Режим пониженного энергопотребления.

Вход в режим SLEEP осуществляется командой SLEEP. По этой команде, если WDT разрешен, то он сбрасывается и начинает счет времени, бит “ \overline{PD} ” в регистре статуса (f3) сбрасывается, бит “ \overline{TO} ” устанавливается, а встроенный генератор выключается. Порты ввода/вывода сохраняют состояние, которое они имели до входа в режим SLEEP. Для снижения потребляемого тока в этом режиме, ножки на вывод должны иметь такие значения, чтобы не протекал ток между кристаллом и внешними цепями. Ножки на ввод должны быть соединены внешними резисторами с высоким или низким уровнем, чтобы избежать токов переключения, вызываемых плавающими высокоомными входами. То же и про TMR0. Ножка \overline{MCLR} должна быть под напряжением V_{ihms} .

Выход из режима SLEEP

осуществляется в результате следующих событий:

Внешний сброс - импульс низкого уровня на на ножке \overline{MCLR} ,

Сброс при срабатывании WDT (если он разрешен),

Прерывания. (Прерывание с ножки INT, прерывание при изменении порта В, прерывание при завершении записи данных EEPROM).

При первом событии происходит сброс всего устройства. Два других события предполагают продолжение выполнения программы.

Бит “ \overline{PD} ” в регистре статуса (03h), который устанавливается при включении, но обнуляется командой “SLEEP”, может быть использован для определения состояния процессора до “просыпания”: или процессор был в режиме “SLEEP”(горячий старт), или было просто выключено питание (холодный старт). Бит “ \overline{TO} ” позволяет определить, чем был вызван выход из режима SLEEP: или внешним сигналом на ножке \overline{MCLR} , или срабатыванием WDT.

Чтобы устройство вышло из режима SLEEP через прерывание, это прерывание должно быть разрешено установкой соответствующей маски в регистре INTCON. При выходе из режима SLEEP будет выполняться фоновая программа, если общая маска запрещает все прерывания (GIE=0). Если GIE=1, то будет выполняться подпрограмма обработки прерываний.