

PIC16CXX - это 8-разрядные микроконтроллеры с RISC архитектурой, (Reduced Instruction Set Computer - Компьютер с сокращенной системой команд), производимые фирмой Microchip Technology. Это семейство микроконтроллеров отличается низкой ценой, низким энергопотреблением и высокой скоростью. Микроконтроллеры имеют встроенное ЭПЗУ ("электрически перепрограммируемое постоянное запоминающее устройство") программы, ОЗУ данных и выпускаются в 18 и 28 выводных корпусах.

PIC OTP - это однократно программируемые пользователем контроллеры, предназначенные для полностью оттестированных и законченных изделий, в которых не будет проходить дальнейших изменений кода. Эти контроллеры выпускаются в дешевых пластиковых корпусах с предварительно заданным типом внешнего генератора - кварцевым или RC.

Для отладки программ и макетирования выпускается вариант контроллеров с ультрафиолетовым стиранием. Эти контроллеры допускают большое число циклов записи/стирания и имеют очень малое время стирания - обычно 1-2 минуты. Однако цена таких контроллеров существенно выше, чем однократно программируемых, поэтому их невыгодно устанавливать в серийную продукцию.

Для изделий, программа которых может меняться, либо содержит какие-либо переменные части, таблицы, параметры калибровки, ключи и т.д., выпускается электрически стираемый и перепрограммируемый контроллер PIC16C84. Он также содержит электрически перепрограммируемое ПЗУ данных. Именно такой контроллер мы и будем использовать для экспериментов.

СЕМЕЙСТВО PIC16CXX

Особенности и преимущества, которые выделяют эти микроконтроллеры среди других:

- Для применений, связанных с защитой информации, каждый PIC имеет **бит секретности**, который может быть запрограммирован для запрещения считывания программного кода и ПЗУ данных. При программировании сначала записывается программный код, проверяется на правильность записи, а затем устанавливается бит секретности. Если попытаться прочитать микросхему с установленным битом секретности, то для PIC16C5X старшие 8 разрядов кода будут считываться как 0, а младшие 4 разряда будут представлять собой скремблированные 12 разрядов команды. Для PIC16C84 аналогично 7 старших разрядов будут считываться нулями, а 7 младших разрядов будут представлять скремблированные 14 разрядов команды. Электрически перепрограммируемое ПЗУ данных PIC16C84 при установке бита защиты не может быть считано. На рисунке 1 представлены все выпускаемые в настоящее время микроконтроллеры и даны их краткие характеристики. (рис на стр 2-1)
- Микроконтроллеры семейства PIC имеют очень эффективную **систему команд**, состоящую всего из **35 инструкций**. Все инструкции **выполняются за один цикл**, за исключением условных переходов и команд, изменяющих программный счетчик, которые выполняются за 2 цикла. Один цикл выполнения инструкции состоит из 4 периодов тактовой частоты. Таким образом, при частоте 4 МГц, время выполнения инструкции составляет 1 мксек. **Каждая инструкция состоит из 14 бит**, делящихся на код операции и операнд (возможна манипуляция с регистрами, ячейками памяти и непосредственными данными). Система команд микроконтроллеров PIC16CXX представлена на рисунке 2. (таблица 7.2 на стр 2-569)
- **Высокая скорость выполнения команд** в PIC достигается за счет использования двухшинной Гарвардской архитектуры вместо традиционной одношинной Фон-Неймановской. Гарвардская архитектура основывается на наборе регистров с

разделенными шинами и адресным пространством для команд и для данных. Набор регистров означает, что все программные объекты, такие как порты ввода/вывода, ячейки памяти и таймер, представляют собой физически реализованные аппаратные регистры.

- Память данных (ОЗУ) для PIC16CXX имеет разрядность **8 бит**, память программ (ППЗУ) имеет разрядность 12 бит для PIC16C5X и 14 бит для PIC16CXX. Использование Гарвардской архитектуры позволяет достичь высокой скорости выполнения битовых, байтовых и регистровых операций. Кроме того, Гарвардская архитектура допускает **конвейерное выполнение инструкций**, когда одновременно выполняется текущая инструкция и считывается следующая. В традиционной же Фон-Неймановской архитектуре команды и данные передаются через одну разделяемую или мультиплексируемую шину, тем самым ограничивая возможности конвейеризации. На рисунке 3 показана блок-схема двухшинной структуры PIC16CXX. (рис В на стр 2-536)

Как Вы можете видеть, внутренние физические и логические компоненты, из которых состоит PIC16CXX аналогичны любому другому микроконтроллеру, с которым Вы могли работать до сих пор. Поэтому писать программы для PIC не сложнее, чем для любого другого процессора. Гарвардская архитектура и большая разрядность команды позволяют сделать код для PIC значительно более компактным, чем для других микроконтроллеров и существенно повысить скорость выполнения программ.

НАБОР РЕГИСТРОВ PIC

Все программные объекты, с которыми может работать PIC, представляют собой физические регистры. Чтобы понять, как работает PIC, нужно разобраться с тем, какие регистры у него существуют и как с каждым из них работать. На рисунке 4 показаны все регистры PIC16C84. (рис 3.7.1 на стр. 2-541)

НАБОР ОПЕРАЦИОННЫХ РЕГИСТРОВ

Этот набор состоит из регистра косвенной адресации (f0), регистра таймера/счетчика (f1), программного счетчика (f2), регистра слова состояния (f3), регистра выбора (f4) и регистров ввода/вывода (f5,f6).

f0 РЕГИСТР КОСВЕННОЙ АДРЕСАЦИИ *INDO*

Регистр косвенной адресации f0 физически не существует. Он использует регистр выбора f4 для косвенной выборки одного из 64 возможных регистров. Любая команда, использующая f0, на самом деле обращается к регистру данных, на который указывает f4.

f1 РЕГИСТР ТАЙМЕРА/СЧЕТЧИКА *TMR0*

Регистр таймера/счетчика TMR0 может быть записан и считан как и любой другой регистр. TMR0 может увеличиваться по внешнему сигналу, подаваемому на вывод RTCC, или по внутренней частоте, соответствующей частоте команд. Основное применение таймера/счетчика - подсчет числа внешних событий и измерение времени. Сигнал от внешнего или внутреннего источника также может быть предварительно поделен при помощи встроенного в PIC программируемого делителя.

f2 ПРОГРАММНЫЙ СЧЕТЧИК *PCL*

Программный счетчик (РС) используется для генерации последовательности адресов ячеек ПЗУ программы, содержащих 14-разрядные команды. РС имеет разрядность 13 бит, что позволяет прямо адресовать 8Кх14 ячеек ПЗУ. Для PIC16C84 однако, только 1К ячеек физически доступно. Младшие 8 разрядов РС могут быть записаны и считаны через регистр f2, старшие 5 разрядов загружаются из регистра PCLATCH, имеющего адрес 0Ah.

f3 РЕГИСТР СЛОВА СОСТОЯНИЯ *STATUS*

Регистр слова состояния похож на регистр PSW, существующий в большинстве микропроцессоров. В нем находятся бит переноса, десятичного переноса и нуля, а также биты режима включения и биты страничной адресации.

f4 РЕГИСТР ВЫБОРА *FSR*

Как было уже сказано, регистр выбора FSR используется вместе с регистром косвенной адресации f0 для косвенной выборки одного из 64 возможных регистров. Физически задействовано 36 регистров ОЗУ пользователя, расположенных по адресам 0Ch-2Fh и 15 служебных регистров, расположенных по различным адресам.

f5, f6 РЕГИСТРЫ ВВОДА/ВЫВОДА *PORTA*, *PORTB*

Регистры f5 и f6 соответствуют двум портам ввода/вывода, имеющимся у PIC16C84. Порт А имеет 5 разрядов RA4-RA0, которые могут быть индивидуально запрограммированы как входы или выходы при помощи регистра TRISA, имеющего адрес 85h. Порт В имеет 8 разрядов RB7-RB0 и программируется при помощи регистра TRISB, имеющего адрес 86h. Задание 1 в разряде регистра TRIS программирует соответствующий разряд порта как вход. При чтении порта считывается непосредственное состояние вывода, при записи в порт запись происходит в буферный регистр.

f8, f9 РЕГИСТРЫ ЭППЗУ *EEDATA*, *EEADR*

PIC16C84 имеет встроенное электрически перепрограммируемое ПЗУ размером 64 байта, которое может быть считано и записано при помощи регистра данных EEDATA и регистра адреса EEADR. Запись нового байта длится около 10 мсек и управляется встроенным таймером. Управление записью и считыванием осуществляется через регистр EECON1, имеющий адрес 88h. Для дополнительного контроля за записью служит регистр EECON2, имеющий адрес 89h.

РЕГИСТРЫ ОБЩЕГО НАЗНАЧЕНИЯ

Регистры общего назначения представляют собой статическое ОЗУ, расположенное по адресам 0Ch-2Fh. Всего в PIC16C84 можно использовать 36 ячеек ОЗУ.

СПЕЦИАЛЬНЫЕ РЕГИСТРЫ *W*, *INTCON*, *OPTION*

К ним относятся рабочий регистр W, используемый в большинстве команд в качестве регистра аккумулятора и регистры INTCON и OPTION. Регистр прерываний INTCON (адрес 0Bh) служит для управления режимами прерывания и содержит биты разрешения прерываний от различных источников и флаги прерываний. Регистр режимов OPTION (адрес 81h) служит для задания источников сигнала для предварительного делителя и таймера/счетчика, а также для задания коэффициента деления предварительного делителя, активного фронта сигнала для RTCC и входа прерывания. Кроме того при помощи регистра OPTION могут быть включены нагрузочные резисторы для разрядов порта В, запрограммированных как входы.

СТОРОЖЕВОЙ ТАЙМЕР WDT

Сторожевой таймер WDT предназначен для предотвращения катастрофических последствий от случайных сбоев программы. Он также может быть использован в приложениях, связанных со счетом времени, например, в детекторе пропущенных импульсов. Идея использования сторожевого таймера состоит в регулярном его сбрасывании под управлением программы или внешнего воздействия до того, как закончится его выдержка времени и не произойдет сброс процессора. Если программа работает нормально, то команда сброса сторожевого таймера CLRWDТ должна регулярно выполняться, предохраняя процессор от сброса. Если же микропроцессор случайно вышел за пределы программы (например, от сильной помехи по цепи питания) либо заиклился на каком-либо участке программы, команда сброса сторожевого таймера скорее всего не будет выполнена в течение достаточного времени, и произойдет полный сброс процессора, инициализирующий все регистры и приводящий систему в рабочее состояние.

Сторожевой таймер в PIC16C84 не требует каких-либо внешних компонентов и работает на встроенном RC генераторе, причем генерация не прекращается даже в случае отсутствия тактовой частоты процессора. Типовой период сторожевого таймера 18 мсек. Можно подключить предварительный делитель на сторожевой таймер и увеличить его период вплоть до 2 сек.

Еще одной функцией сторожевого таймера служит включение процессора из режима пониженного энергопотребления, в который процессор переводится командой SLEEP. В этом режиме PIC16C84 потребляет очень малый ток - около 1 мкА. Перейти из этого режима в рабочий режим можно или по внешнему событию нажатию кнопки, срабатыванию датчика, или по сторожевому таймеру.

ТАКТОВЫЙ ГЕНЕРАТОР

Для микроконтроллеров семейства PIC возможно использование четырех типов тактового генератора:

- ХТ кварцевый резонатор
- HS высокочастотный кварцевый резонатор
- LP микропотребляющий кварцевый резонатор
- RC цепочка

Задание типа используемого тактового генератора осуществляется в процессе программирования микросхемы. В случае задания вариантов ХТ, HS и LP к микросхеме подключается кварцевый или керамический резонатор либо внешний источник тактовой частоты, а в случае задания варианта RC - резистор и конденсатор. Конечно, керамический и, особенно, кварцевый резонатор значительно точнее и стабильнее, но если высокая точность отсчета времени не нужна, использование RC генератора может уменьшить стоимость и габариты устройства.

СХЕМА СБРОСА

Микроконтроллеры семейства PIC используют внутреннюю схему сброса по включению питания в сочетании с таймером запуска генератора, что позволяет в большинстве ситуаций обойтись без традиционного резистора и конденсатора. Достаточно просто подключить вход MCLR к источнику питания. Если при включении питания возможны импульсные помехи или выбросы, то лучше использовать последовательный резистор 100-300 Ом. Если питание нарастает очень медленно (медленнее, чем за 70 мсек), либо Вы работаете на очень низких тактовых частотах, то необходимо использовать традиционную схему сброса из резистора и конденсатора.

Система команд PIC-контроллеров серии PIC16C8X

Каждая команда представляет собой 14-разрядное слово, содержащее поле кода операции OPCODE и поле операндов. Система команд включает в себя

команды работы с байтами,
команды работы с битами,
команды управления и операции с константами.

Для команд работы с байтами "f" обозначает регистр, с которым производится действие, а бит "d" определяет регистр назначения. При "d"=0 результат помещается в регистр W, при "d"=1 результат помещается в регистр "f", заданный в команде.

Для команд работы с битами "b" обозначает номер бита, участвующего в команде, а "f" - регистр, в котором этот бит расположен.

Для команд управления и операций с константами "k" обозначает 8- или 11-битовую константу или идентификатор.

Все команды выполняются в течение одного командного цикла, кроме следующих двух случаев:

Переход по проверке условия, если результат проверки условия - истина.
Изменение счетчика команд как результат выполнения команды.

В этих случаях команда выполняется за два цикла с выполнением второго цикла как NOP. Один командный цикл состоит из четырех периодов генератора. Таким образом, для генератора с частотой 4 МГц время выполнения команды составит 1 мкс. Если выполняется переход по проверке условия или в результате выполнения команды изменился счетчик команд, время выполнения этой команды при тактовой частоте 4 МГц составит 2 мкс.

Принятые обозначения

f:	Адрес регистра
W:	Рабочий регистр
b:	Номер бита в 8-ми разрядном регистре
k:	Константа
x:	Не используется. Ассемблер формирует код с x=0
d:	Регистр назначения:
d=0	- результат в регистре W
d=1	- результат в регистре f. (По умолчанию d=1)
label:	Имя метки
TOS:	Вершина стека
PC:	Счетчик команд
TO:	Тайм-аут
PD:	Выключение питания
dest:	Регистр назначения: рабочий регистр W или регистр, заданный в команде
[]:	Необязательные параметры
():	Содержание
-->:	Присвоение
<>:	Битовое поле
O:	Из набора

Обозначение	Функция	Циклы	Код команды	Биты состояния	Примечания
ADDLW	Сложение константы и W	1	11 111x kkkk kkkk	C, DC, Z	
ADDWF	Сложение W с f	1	00 0111 dfff ffff	C, DC, Z	1, 2
ANDLW	Логическое И константы и W	1	11 1001 kkkk kkkk	Z	
ANDWF	Логическое И W и f	1	00 0101 dfff ffff	Z	1, 2
BCF	Сброс бита в регистре f	1	01 00bb bfff ffff		1, 2
BSF	Установка бита в регистре f	1	01 01bb bfff ffff		1, 2
BTFSC	Пропустить команду, если бит в f равен нулю	1 (2)	01 10bb bfff ffff		3
BTFSS	Пропустить команду, если бит в f равен единице	1 (2)	01 11bb bfff ffff		3
CALL	Вызов подпрограммы	2	10 0kkk kkkk kkkk		
CLRF	Сброс регистра f	1	00 0001 1fff ffff	Z	2
CLRW	Сброс регистра W	1	00 0001 0xxx xxxx	Z	
CLRWDT	Сброс сторожевого таймера WDT	1	00 0000 0110 0100	TO, PD	
COMF	Инверсия регистра f	1	00 1001 dfff ffff	Z	1, 2
DECF	Декремент регистра f	1	00 0011 dfff ffff	Z	1, 2
DECFSZ	Декремент f, пропустить команду, если 0	1 (2)	00 1011 dfff ffff		1, 2, 3
GOTO	Переход по адресу	2	10 1kkk kkkk kkkk		
INCF	Инкремент регистра f	1	00 1010 dfff ffff	Z	1, 2
INCFSZ	Инкремент f, пропустить команду, если 0	1 (2)	00 1111 dfff ffff		1, 2, 3
IORLW	Логическое ИЛИ константы и W	1	11 1000 kkkk kkkk	Z	
IORWF	Логическое ИЛИ W и f	1	00 0100 dfff ffff	Z	1, 2
MOVF	Пересылка регистра f	1	00 1000 dfff ffff	Z	1, 2
MOVLW	Пересылка константы в W	1	11 00xx kkkk kkkk		
MOVWF	Пересылка W в f	1	00 0000 1fff ffff		
NOP	Холостая команда	1	00 0000 0xx0 0000		
OPTION	Загрузка регистра OPTION	1	00 0000 0110 0010		
RETFIE	Возврат из прерывания	2	00 0000 0000 1001		
RETLW	Возврат из подпрограммы с загрузкой константы в W	2	11 01xx kkkk kkkk		
RETURN	Возврат из подпрограммы	2	00 0000 0000 1000		
RLF	Сдвиг f влево через перенос	1	00 1101 dfff ffff	C	1, 2
RRF	Сдвиг f вправо через перенос	1	00 1100 dfff ffff	C	1, 2
SLEEP	Переход в режим SLEEP	1	00 0000 0110 0011	TO, PD	
SUBLW	Вычитание W из константы	1	11 110x kkkk kkkk	C, DC, Z	
SUBWF	Вычитание W из f	1	00 0010 dfff ffff	C, DC, Z	1, 2
SWAPF	Обмен местами тетрадь wf	1	00 1110 dfff ffff		1, 2
TRIS	Загрузка регистра TRIS	1	00 0000 0110 0fff		
XORLW	Исключающее ИЛИ константы и W	1	11 1010 kkkk kkkk	Z	
XORWF	Исключающее ИЛИ W и f	1	00 0110 dfff ffff		1, 2

Примечание:

1. Если модифицируется регистр ввода/вывода (например, MOVF PORTB,1), то используется значение, считываемое с выводов. Например, если в выходной защелке порта, включенного на ввод, находится "1", а внешнее устройство формирует на этом выводе "0", то в этом разряде данных будет записан "0".
2. Если операндом команды является содержимое регистра TMRO (и, если допустимо, d=1), то предварительный делитель, если он подключен к TMRO, будет сброшен.
3. Если в результате выполнения команды изменяется счетчик команд, или выполняется переход по проверке условия, то команда выполняется за два цикла. Второй цикл выполняется как NOP.

PIC-контроллеры

<http://rf.atnn.ru/pic.html>

Система команд PIC-контроллеров серии PIC16C8X

<http://www.gaw.ru/html.cgi/txt/doc/micros/pic/asm14/start.htm>

Самоучитель по программированию PIC контроллеров для начинающих:

http://www.ikarab.narod.ru/Kea_20.html