

МИКРОКОНТРОЛЛЕРЫ PIC12C5XX

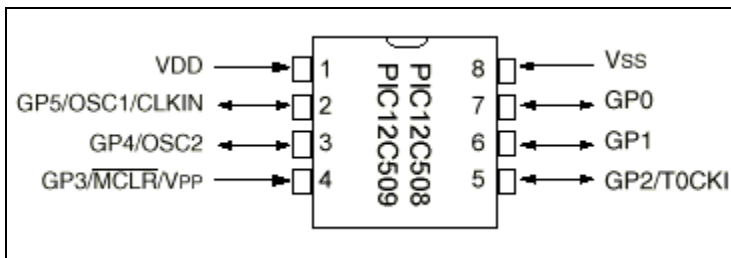
1.0 ОБЩЕЕ ОПИСАНИЕ

PIC12C5XX - семейство 8 битных, полностью статических CMOS микроконтроллеров. PIC12C5XX имеют RISC архитектуру с 33 командами. Все команды выполняются за один машинный цикл (4 периода тактового генератора) кроме команд переходов, которые выполняются за два цикла.

Микроконтроллер имеет следующие модули и особенности:

- Семь регистров специальных функций
- Двухуровневый стек
- Прямые, косвенные и относительные способы адресации для данных и команд
- Внутренний RC генератор на 4 MHz
- Встроенный автомат последовательного программирования
- 8-разрядный таймер / счетчик реального времени (TMR0)
- 8-разрядный программируемый предварительный делитель (ПД)
- Сброс по включению питания (POR)
- Таймер сброса (DRT)
- Сторожевой таймер (WDT) с собственным RC генератором
- Программируемая защита кода
- Режим пониженного энергопотребления SLEEP
- Выход из режима SLEEP при изменении уровня на внешнем выводе
- Возможность программного подключения внутренних подтягивающих резисторов
- Внутренний подтягивающий резистор на выводе MCLR
- Выбор типа генератора:
 - INTRC: Внутренний RC генератор 4 MHz
 - EXTRC: Внешний RC генератор
 - XT: Стандартный ВЧ резонатор
 - LP: низкочастотный генератор

PIC12C5XX конструктивно выполнены в 8 выводных корпусах (рис 1).



Сброс Устройства и сброс по включению питания (POR)

PIC12C5XX имеют внутренний таймер сброса (DRT) который исключает необходимость во внешней схеме сброса по включению питания, четыре типа конфигурации задающего генератора, режим SLEEP, сторожевой таймер и защиту кода программы от считывания.

ТАБЛИЦА 1-1: СЕМЕЙСТВО PIC12C5XX

Тип микроконтроллера		PIC12C508	PIC12C509
Частота	Максимальная рабочая частота(MHZ)	4	4
Память	Память программы (x 12 слов)	512	1024
	Память данных (байт)	25	41
	Модуль таймера	TMR0	TMR0
	Выход из режима SLEEP	Есть	Есть
	Количество выводов ввода\вывода	5	5
	Количество выводов на ввод	1	1
	Внутренние подтягивающие резисторы	Есть	Есть
	Напряжение питания	2,5 - 5,5	2,5 - 5,5
	Внутр. последоват. программирование	Есть	Есть
	Число команд	33	33
	Тип корпуса	8-pin DIP, SOIC	8-pin DIP, SOIC

2.0 ТИПЫ КРИСТАЛЛОВ PIC12C5XX

2.1 UV стираемые кристаллы

UV стираемая версия, предлагаемая в керамическом корпусе является оптимальной для разработки прототипа, т.к. допускает многократное перепрограммирование микросхемы.

Внимание: стирание программы также сотрет предварительно запрограммированное внутреннее значение калибровки внутреннего генератора.

2.2 Одноразово программируемые кристаллы

ОТР устройства являются разово программируемыми и используются для серийной продукции.

3.0 КРАТКИЙ АРХИТЕКТУРНЫЙ ОБЗОР

Микроконтроллеры PIC12C5XX используют только внутреннюю память программ

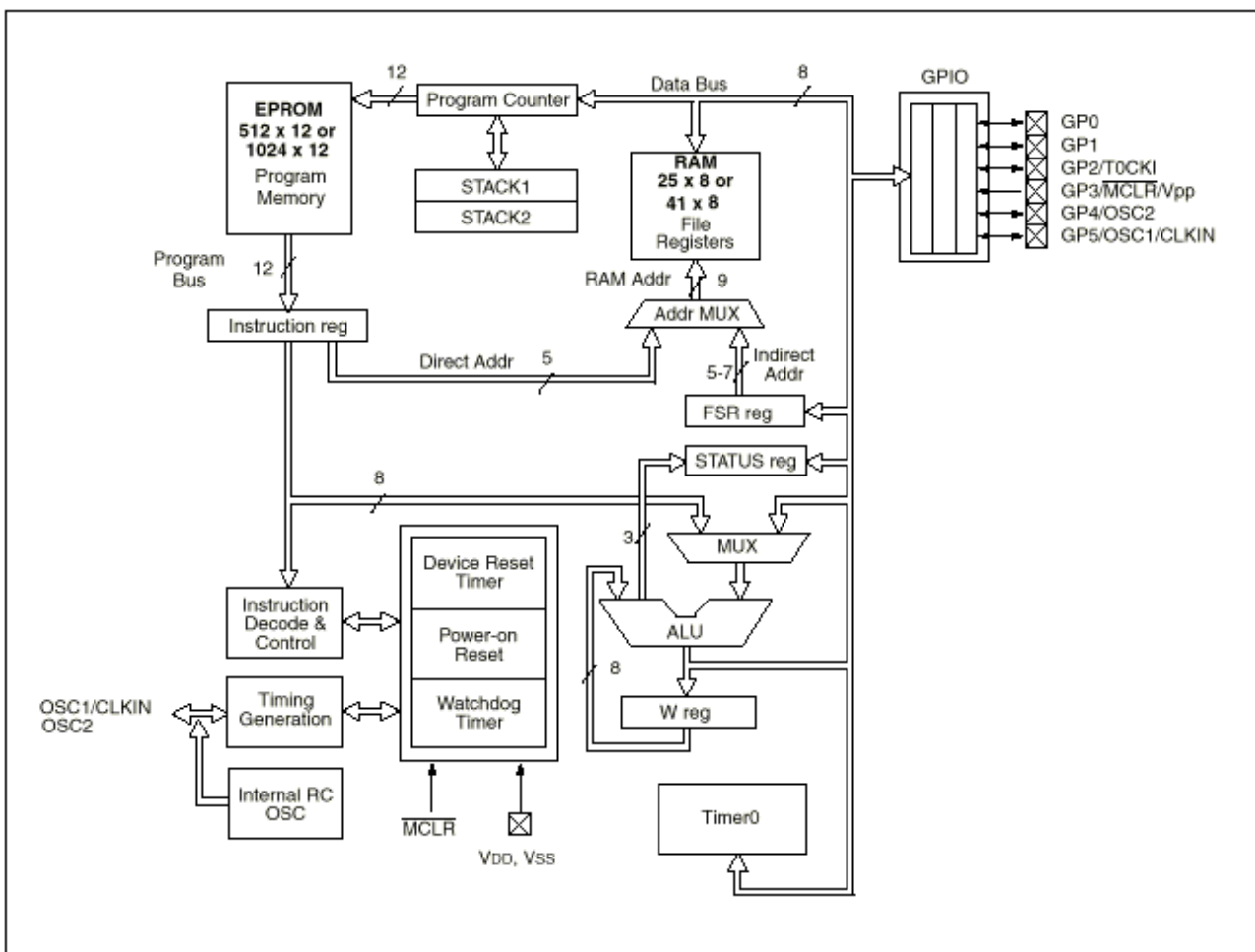
PIC12C508 адресует 512 x 12 памяти программы,

PIC12C509 адресует 1k x 12 памяти программы.

В PIC12C5XX команды - 12 битовые и выбираются за один цикл. Обращение к программе и данным осуществляется по отдельным шинам.

PIC12C5XX может непосредственно или косвенно адресовать память данных в которую входят регистры специальных функций и регистры RAM. PIC12C5XX имеет высоко ортогональную систему команд, которая делает возможным выполнить любую операцию на любом регистре, используя любой способ адресации. Это делает программирование на PIC12C5XX простым и эффективным. 8 разрядное ALU выполняет операции сложения, вычитания, сдвига и логических операций. В командах с двумя операндами, один операнд - регистр W, другим операндом является или регистр RAM, или непосредственно константа. В командах с одним операндом, операндом является или регистр W или регистр RAM. Регистр W - 8 битный, не адресуемый рабочий регистр используемый для операций ALU. В зависимости от выполненной команды, ALU может воздействовать на биты C, DC и Z в регистре STATUS.

Блок схема PIC12C5XX



Описание порта ввода-вывода

GP0 (вывод 7) ввода - вывода TTL/ST

Двунаправленный вывод. Имеет программно-устанавливаемые подтягивающие резисторы. Может программно устанавливать ся для выхода из режима SLEEP по изменению состояния на выводе.

GP1 (вывод 6) ввода - вывода TTL/ST

Двунаправленный вывод. Имеет программно-устанавливаемые подтягивающие резисторы. Может программно устанавливать ся для выхода из режима SLEEP по изменению состояния на выводе.

GP2/T0CKI (вывод 5) ввода - вывода ST

Двунаправленный вывод. Может быть сконфигурирован для внешнего тактирования TMR0 (T0CKI).

GP3/MCLR/VPP (вывод 4) ввод TTL

Входной вывод. Когда сконфигурирован как MCLR, этот вывод осуществляет сброс МК при подаче на него низкого уровня. Напряжение на выводе не должно превышать VDD. Может быть программно установлен для выхода из режима SLEEP по изменению состояния на выводе. Имеет программно-устанавливаемые подтягивающие резисторы. Если вывод сконфигурирован как вход сброса, то подтягивающий резистор подключается автоматически.

GP4/OSC2 (вывод 3) ввод - вывод TTL

Двунаправленный вывод. Программно может быть установлен как обычный разряд ввода-вывода или для подключения внешнего кварцевого резонатора в режиме XT или LP.

GP5/OSC1/CLKIN (вывод 2) ввод - вывод TTL/ST

Двунаправленный вывод (вход генератора) вход внешнего генератора. Вывод используется как вход-выход только при использовании внутреннего RC генератора, во всех других режимах как OSC1 вход..

VDD (вывод 1) P

Питание

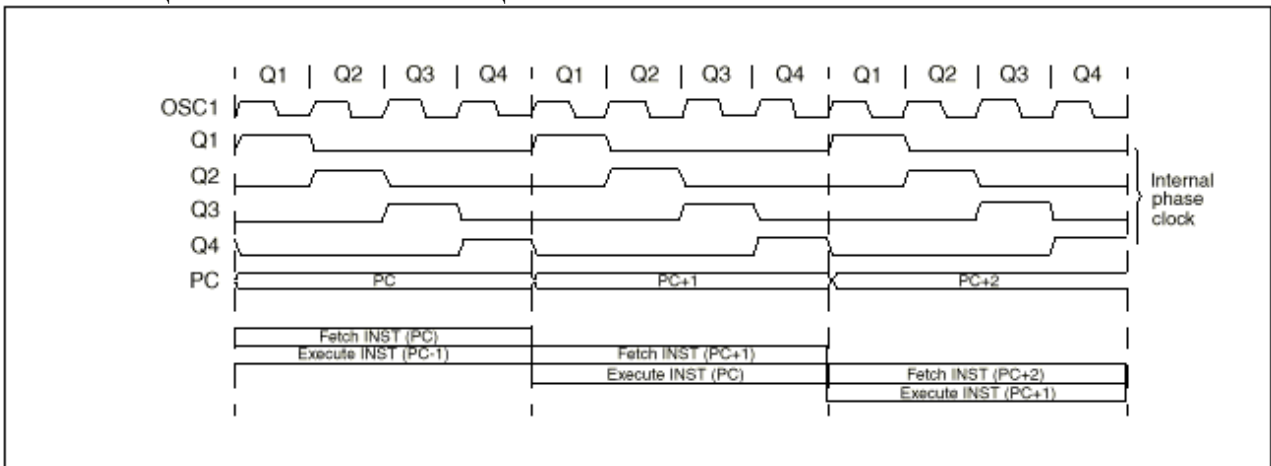
VSS (вывод 8) P

Общий

3.1 Синхронизация.

Сигнал задающего генератора делится на четыре, чтобы сформировать один машинный цикл (четыре такта) а именно Q1, Q2, Q3 и Q4. Счетчик программ увеличивается в такте Q1, команда выбирается из памяти программ и запирается в регистр команд в такте Q4. Команда декодируется и выполняется в течение следующего цикла. При выполнении текущей команды, из памяти программ осуществляется выборка следующей, за исключением команд переходов. Сигналы и поток выполнения команды показаны на рисунке 3-2 и 3-1.

РИС. 3-2: ЦИКЛ СИНХРОНИЗАЦИИ



Примечание: Fetch - Выборка, Execute - Выполнение

3.2 КОНВЕЙЕРНАЯ ОБРАБОТКА КОМАНД

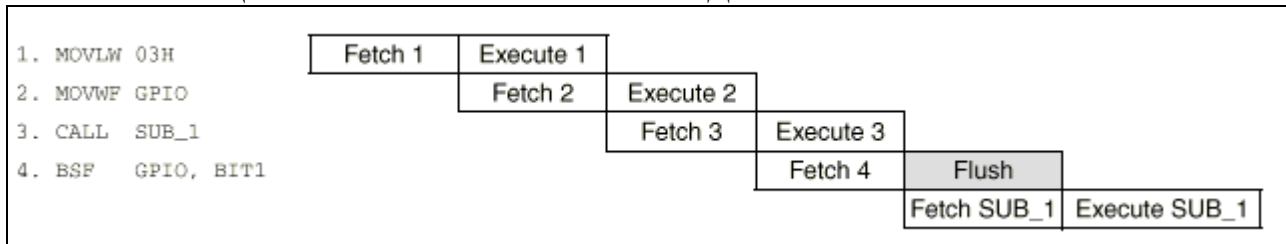
Цикл выполнения команды состоит из четырех Q1, Q2, Q3 и Q4 тактов задающего генератора. Выборка команды осуществляется за один цикл, а декодирование и выполнение за следующий. Однако, из-за конвейерной обработки, каждая команда действительно выполняется в одном цикле. Если команда изменяет счетчик программ (например, GOTO), то для ее выполнения требуются два цикла (Пример 3-1). Цикл выборки начинается с приращения счетчика программ (PC) в Q1. В цикле выполнения, выбранная команда защелкивается в регистре команд (IR) в цикле Q1. Затем команда декодируется и

4

PIC12C5XX

выполняется в течении тактов Q2, Q3, и Q4. Чтение операнда из памяти данных осуществляется в такте Q2 а запись в т акте Q4.

ПРИМЕР 3-1: ЦИКЛ ВЫПОЛНЕНИЯ КОМАНДЫ



Все команды выполняются за один цикл, кроме команд переходов, которым требуется два цикла.

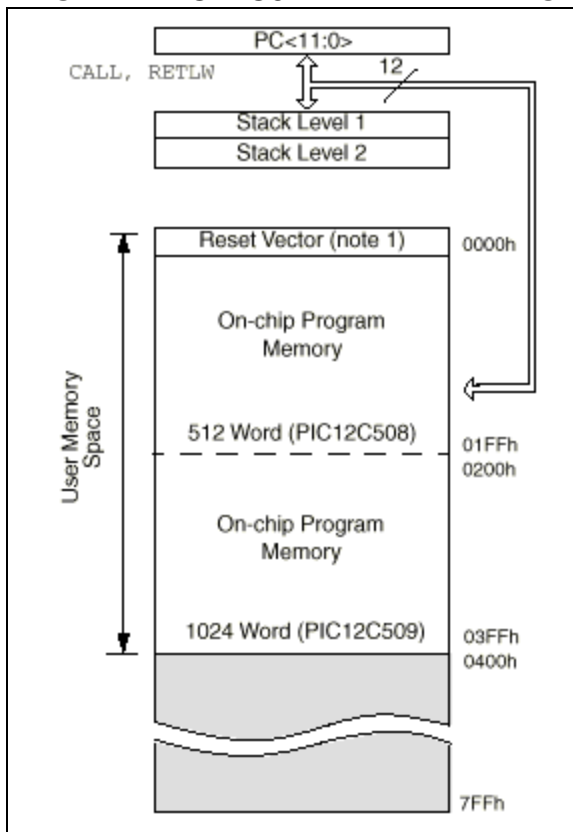
4.0 ОРГАНИЗАЦИЯ ПАМЯТИ

В PIC12C5XX Память организована в память программы и память данных. Для кристаллов имеющих более 512 байт памяти программы, используется страничная адресация. Выбор страницы памяти программ осуществляется сбросом/установкой бита "PA0" в регистре "STATUS". Для PIC12C509 с регистром памяти данных более чем 32 регистра, используется банковая схема. К банкам памяти данных обращаются, используя регистр FSR."

4.1 ОРГАНИЗАЦИЯ ПАМЯТИ ПРОГРАММ

PIC12C5XX имеет 12 битный счетчик команд (PC), способный к адресации 2k x 12 пространства памяти программы. Только первые 512 x 12 байт (0000h-01FFh) для PIC12C508 и 1k x 12 (0000h-03FFh) для PIC12C509 выполнены физически (рис.4-1)

РИС. 4-1: PIC12C5XX ПАМЯТЬ ПРОГРАММЫ И СТЕК



Примечание 1: Адрес 0000h становится вектором сброса. Адрес 01FFh для PIC12C508 и 03FFh для (PIC12C509) содержит значение калибровки генератора: MOVLW XX INTRC.

4.2 Организация Памяти Данных

Память Данных состоит из регистров, (байтов) оперативной памяти. Регистры разделены в две функциональных группы: регистры специальных функций и универсальные регистры.

Регистры специальных функций включают регистр TMR0, регистр счетчика команд (PC), регистр STATUS, регистр ввода - вывода PORT, и регистр косвенной адресации FSR. Специальные регистры используются и для управления конфигурацией порта ввода - вывода и ПД.

Универсальные регистры RAM используются для хранения данных.

PIC12C508, состоит из 7 регистров специальных функций и 25 регистров RAM (Рис 4-2)

PIC12C509, состоит из 7 регистров специальных функций, 25 универсальных регистров, и 16 регистров (универсальных), которые могут быть адресованы, используя банковую схему (рис. 4-3).

РИС.4-2: PIC12C508 Регистры специальных функций и RAM

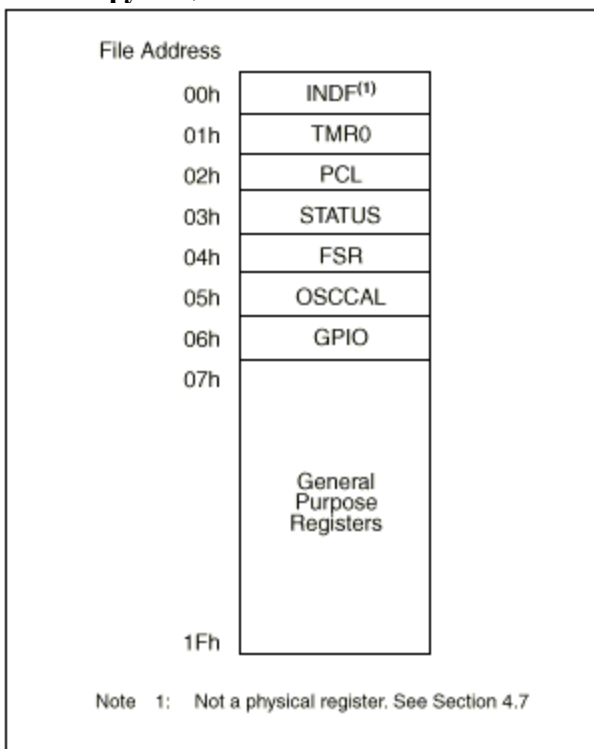
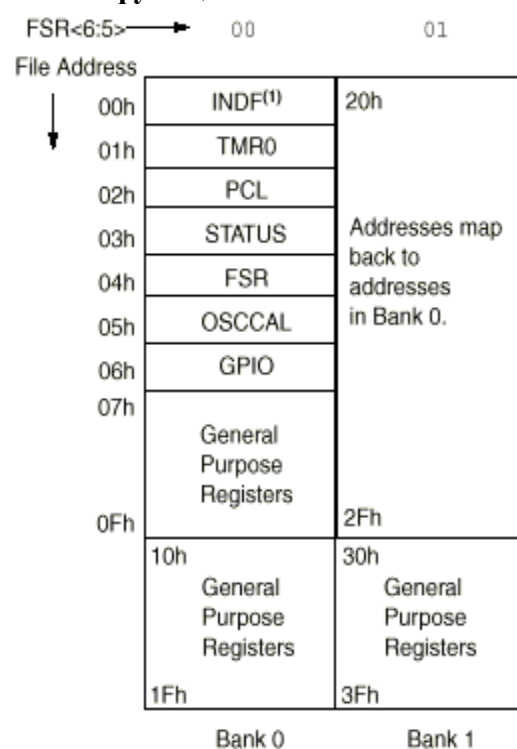


РИС.4-3: PIC12C509 Регистры Специальных функций и RAM



Примечание 1: Не физический регистр. См. Раздел 4.7

4.2.1 РЕГИСТРЫ ОПЕРАТИВНОЙ ПАМЯТИ (RAM)

К регистрам RAM обращаются или непосредственно или косвенно через регистр косвенной адресации FSR (Раздел 4.7).

4.2.2 РЕГИСТРЫ СПЕЦИАЛЬНЫХ ФУНКЦИЙ

Регистры специальных функций - регистры, используемые ЦЕНТРАЛЬНЫМ ПРОЦЕССОРОМ и периферийными модулями для управления операциями устройства (Таб.4-1) классифицированы в два набора:

Специальные регистры, связанные с "основными" функциями и описаны в этом разделе;

Регистры, связанные с периферийными устройствами описаны в разделе для каждого этого устройства.

ТАБЛИЦА 4-1: РЕГИСТРЫ СПЕЦИАЛЬНЫХ ФУНКЦИЙ (SFR)

Адрес	Имя	Бит7	Бит6	Бит5	Бит4	Бит3	Бит2	Бит1	Бит0
N/A	TRIS	Регистр управления ввода – вывода							
N/A	OPTION	Содержит служебные биты, для конфигурации Timer0, Timer0/WDT и ПД,а также разрешает прерывания и выход из режима SLEEP по перепаду входного уровня на входе							
00h	INDF	Использует содержание FSR для адресации памяти данных RAM (не физический регистр)							
01h	TMR0	таймер / счетчик 8-bit							
02h ⁽¹⁾	PCL	младшие 8 бит PC							
03h	STATUS	GPWUF	-----	PA0	TO	PD	Z	DC	C
04h	FSR	Косвенная память данных							
05h	OSCCAL	CAL7	CAL6	CAL5	CAL4	-----	-----	-----	-----
06h	GPIO	-----	-----	GP5	GP4	GP3	GP2	GP1	GP0

Примечание 1 : “-----” невыполнен, читается как '0'

2 : старший байт счетчика программ непосредственно не доступен. См. Раздел 4.5 для объяснения того, как обратиться к этим битам.

4.3 РЕГИСТР STATUS

Этот регистр содержит флаги состояния ALU при выполнении арифметических операций, при сбросе МК, и страницы ПД, бит выбора банка программ при адресации более 512 байт.

Регистр STATUS может быть адресом для любой команды, как и любой другой регистр. Если регистр STATUS - адресат для команды, которая воздействует на Z, DC или C биты, то запись в эти биты заблокирована. Эти биты устанавливаются или сбрасываются согласно логике устройства. Следовательно, результат команды с регистром STATUS как адресатом может быть иным чем предполагалось.

Например команда CLRFS STATUS очистит верхние три бита и установит Z бит. Это установит регистр STATUS в 000u u1uu (где u = неизменный бит).

Чтобы изменять биты регистра STATUS рекомендуется использовать команды BCF, BSF и MOVWF, т.к эти команды не воздействуют на биты Z, DC и C регистра STATUS.

РИС. 4-4: РЕГИСТР STATUS (АДРЕС: 03H)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
GPWUF	-	PA0	TO	PD	Z	DC	C
bit7							bit0

R = Читаемый бит

W = Перезаписываемый бит

n = Значение при сбросе POR

Bit7: GPWUF: GPIO бит сброса

1 = Устанавливается по сбросу при изменении уровня на выводе при выходе МК из режима SLEEP

0 = Сбрасывается по включению питания или при другом сбросе.

Bit6: Не используется

Bit5: PA0: бит выбора страницы памяти программ

1 = Страница 1 (200h - 3FFh) - PIC12C509

0 = Страница 0 (000h - 1FFh) - PIC12C508 и PIC12C509

Bit4: TO

1 = Устанавливается по включению питания, и по командам CLRWDT и SLEEP

0 = Сбрасывается при срабатывании WDT

Bit3: PD

1 = Устанавливается после включения питания или командой CLRWDT
 0 = Сбрасывается выполнением команды SLEEP

Bit2: Z: бит нуля

1 = Если результат арифм. или логич. операции равен нулю
 0 = Если результат арифм. или логич. операции не равен нулю

Bit1 DC бит переноса / заема цифры (команды ADDWF и SUBWF)

При выполнении команды ADDWF

1 = Если произошел перенос из 4-ого бита результата
 0 = Если перенос из 4-ого бита результата не происходил

При выполнении команды SUBWF

1 = Если заем из 4-ого бита результата не происходил
 0 = Если произошел заем из 4-ого бита результата

Bit0: C бит переноса / заема (команды ADDWF, SUBWF, RRF, RLF)

При выполнении команд ADDWF SUBWF RRF или RLF

1 = Перенос бита произошел 1 = заем бита не происходил
 Загрузки с LSB или MSB, соответственно
 0 = Перенос не происходил 0 = заем произошел

4.4 Регистр OPTION

Регистр OPTION содержит биты управления для конфигурации Timer0 Timer0 / WDT и (ПД). Выполняя команду OPTION, содержимое регистра W пересылается в регистр OPTION. При сбросе, биты регистра OPTION устанавливаются в "1".

Внимание: Если какой-либо бит в регистре TRIS установлен в "0", выход из режима SLEEP по изменению состояния на этом выводе заблокирован, т. е. регистр TRIS отменяет контроль регистра OPTION над GPPU и GPWU. Обратите внимание: Если бит TOCS установлен в "1", то бит GP2 вынужден быть вводом даже если TRIS GP2 = "0".

РИСУНОК 4-5: РЕГИСТР OPTION

W-1	W-1	W-1	W-1	W-1	W-1	W-1	W-1	bit7
GPWU	GPPU	TOCS	T0SE	PSA	PS2	PS1	PS0	bit0

W = Перезаписываемый бит

U = Невыполненный бит

n = Значение при сбросе POR Таблица ссылок 4-1 сбросы.

Bit7: GPWU: Разрешение прерывания (или выход из SLEEP) при изменении входного уровня на выводах (GP0, GP1, GP3)

1 = Запрещено
 0 = Разрешено

Bit6: GPPU: Подтягивающие резисторы (GP0, GP1, GP3)

1 = Отключены
 0 = Подключены

Bit5: T0CS: Timer0 выбор источника тактирования

1 = С внешнего вывода GP2 / T0CKI
 0 = Внутренняя частота, Fosc/4

Bit4: T0SE: выбор фронта переключения Timer0

1 = Инкремент при переходе с "1" на "0" на выводе T0CKI

0 = Инкремент при переходе с “0” на “1” на выводе ТОСКИ

Bit3: PSA: Бит назначения предварительного делителя (ПД)

1 = Предварительный делитель включен после WDT

0 = Предварительный делитель включен перед Timer0

Bit2-0: PS2-PS0: Выбор коэф. деления ПД

Bit Value	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

4.5 Счетчик команд.

Счетчик команд (далее PC) содержит адрес следующей команды программы. При отсутствии команд перехода изменяющих PC, значение PC инкрементируется в каждом цикле команды

Для команды GOTO, биты 8:0 PC находятся в теле команды и защелкиваются в PC (PCL) на PC <7:0>. Бит 5 регистра STATUS обеспечивает информацию о выбранной странице и переписывается в бит 9 PC (Рис.4-6).

Для команды CALL, или любой команды, где PCL - адресат, биты 7:0 PC находятся в теле команды. Однако, PC <8> не исходит из слова команды, и всегда очищается (Рис.4-6).

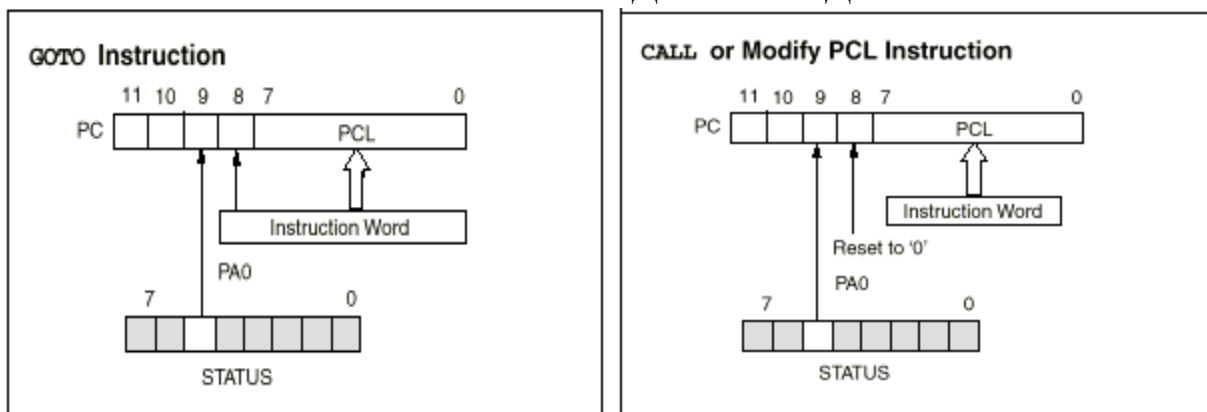
Команды, где PCL - адресат, или команды изменяющие PCL, включают MOVWF PC, ADDWF PC, и BSF PC,5.

Внимание: т.к. PC <8> очищается командой CALL, или любой командой изменяющей PCL, все обращения подпрограммы, или переходы ограничены первыми 256 адресами памяти программы (длинной 512 слов).

4.5.1 ЭФФЕКТЫ СБРОСА

После сброса (и при включении питания) PC адресует последнее расположение в последней странице т.е. кал ибровку генератора, а затем перейдет к адресу 00h, и начнет выполнять код пользователя. После сброса бит с траницы регистра STATUS очищен, что означает, что выбрана страницу 0. Следовательно, после сброса (по команде RESET), пока значение бита страницы не изменено команда GOTO будет осуществлять переход в п ределах страницы 0,.

РИСУНОК 4-6: ЗАГРУЗКА КОМАНД ПЕРЕХОДА PC -PIC12C508/C509



4.6 Стек

PIC12C5XX устройства имеют двухуровневый стек и широкие аппаратные средства с 12 битами, записи / чтения стека.

Команда CALL поместит текущее (актуальное) значение стека 1 в стек 2 и затем поместит в стек 1 текущее (актуальное) значение счетчика команд, увеличенное на единицу. Если выполнено более двух последовательных обращений к стеку, то будут сохранены только два последних адреса возврата. Команда RETLW загрузит содержимое стека в счетчик команд, и переписет содержимое из второго уровня в 1. Если выполнено более чем два последовательных обращения RETLW, стек будет заполнен адресом, предварительно сохраненным в уровне 2. Обратите внимание, что регистр W будет загружен литеральным значением, определенным в команде. Это особенно полезно для реализации таблиц поисковой таблицы данных внутри памяти программы.

4.7 Косвенная Адресация Данных, INDF и FSR Регистры

Регистр INDF - не физический регистр. Адресация INDF фактически адресует регистр, чей адрес содержится в регистре FSR (FSR - указатель). Это - косвенная адресация.

ПРИМЕР 4-1: КОСВЕННАЯ АДРЕСАЦИЯ

В регистре RAM с адресом 07h находится константа 10h

В регистре RAM с адресом 08h находится константа 0Ah

Загружаем величину 07h в регистр FSR (любое число загруженное в регистр. будет представлять собой адрес регистра RAM).

При чтении регистра INDF будет считано значение 10h находящееся в регистре RAM с адресом 07h

Производим инкремент регистра FSR (в регистре будет значение 08h – адрес следующей ячейки RAM)

Теперь при чтении регистра INDF будет считано значение 0Ah находящееся в регистре RAM с адресом 08h

При чтении косвенно самого регистра INDF будет получен результат 00h.

При записи косвенным образом в регистр INDF будет выполнена холостая операция (Nop), хотя биты регистра **STATUS** могут быть изменены.

Программа, для очистки оперативной памяти с адресами 10h-1Fh с использованием косвенной адресации показана в примере:

```

        movlw 0x10
        movwf FSR
NEXT    clrf INDF
        incf FSR,F
        btfsc FSR,4
        goto NEXT

```

Продолжение программы

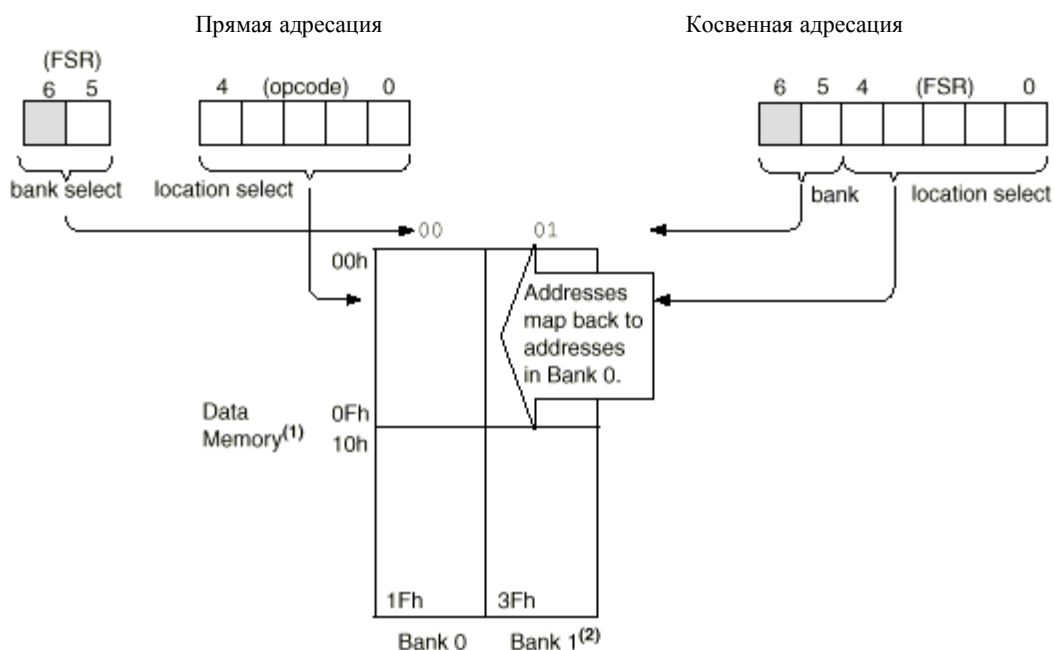
FSR-регистр используется вместе с регистром INDF для косвенной адресации области памяти данных и имеет 5 бит.

Биты <4:0> используются для адресации памяти данных, с адресами 00h-1Fh.

PIC12C508: Не использует биты FSR <6:5> которые читаются как "1".

PIC12C509: Использует бит FSR <5> для выбора банка "0" или "1". Бит FSR <6> не используется, читается как "1".

РИСУНОК 4-7: ПРЯМАЯ / КОСВЕННАЯ АДРЕСАЦИЯ



Примечание 1: См. Раздел 4.2.

Примечание 2: Только PIC12C509

5.0 ПОРТ ВВОДА/ВЫВОДА

Как и любой другой регистр, регистр ввода / вывода может быть записан и прочитан под управлением программы. Однако, при чтении, команды (например, `MOVWF GPIO,W`) данные всегда считываются с внешнего вывода, независимо от режима работы порта на ввод или вывод. При СБРОСЕ, все порты ввода / вывода настроены на ввод так как регистры управления ввода / вывода установлены.

5.1 GPIO

Порт GPIO- 8 разрядный регистр ввода/вывода. Используется только младшие 6 битов некоторые разряды на ввод или вывод. При выполнении альтернативных функций (GP5-GP0). Биты 7 и 6 отсутствуют и считываются как "0". Вывод GP3 является только входным. Слово конфигурации может устанавливать типовых функций (например A/D) в выводе будут читаться как "0" в течении чтения порта. Выводы GP0, GP1, и GP3 могут быть сконфигурированы с внутренними подтягивающими резисторами а также с возможностью выхода из режима SLEEP при изменении уровня на выводе. Если GP4 сконфигурирован как MCLR, то внутренний подтягивающий резистор всегда включен, и выхода из режима SLEEP при изменении уровня на выводе для этого вывода не будет т.к. произойдет сброс МК.

5.2 Регистр TRIS

11

PIC12C5XX

МК выполняя команду TRIS f загружает регистр управления драйвером ввода / вывода (PORT) содержимым регистра W, При записи “1”, вывод настраивается на ввод. При записи “0” вывод МК настраивается на вывод, и содержимое защелки помещается на выбранных выводах. Исключение - GP3, который является только входным и GP2, который может управляться регистром OPTION, см. Раздел 4.4.

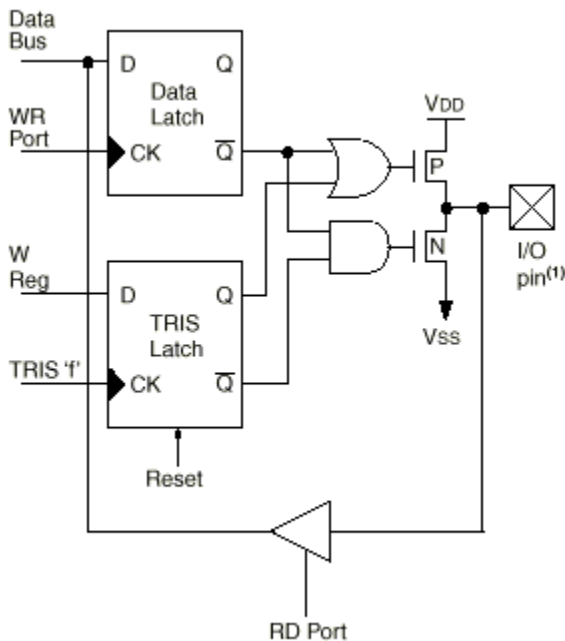
Регистры TRIS “только для прерывания” и установлены (заблокированные драйверы вывода) после СБРОСА.

Обратите внимание: при чтение портов считывается значение с вывода, а не из защелки выходных данных. То есть если драйвер вывода установлен на вывод и вывод имеет высокий выходной уровень, а внешняя схема поддерживает, на нем низкий уровень, то при чтение вывода в защелку будет записан “0”.

5.3 Связь с помощью интерфейса ввода / вывода

Эквивалентная схема для вывода порта ввода - вывода показывается на рисунке 5-1. Все выводы порта, за исключением GP3, который является только входным, могут использоваться для, операций ввода и вывода. Сигнал на выводе должен присутствовать до тех пор, пока не будет считан входной командой (например, MOVF GPIO, W). Данные защелкиваются в регистре-защелке и остаются неизменными, до следующего считывания порта. Чтобы использовать вывод порта как вывод, соответствующий бит направления в TRIS должен быть очищен (= 0). Для использования как ввод, соответствующий TRIS бит должен быть установлен. Любой вывод ввода - вывода (за исключением GP3) может программироваться индивидуально на ввод или вывод.

РИС. 5-1: Эквивалентная схема одного разряда ввода - вывода



Примечание 1: Выводы ввода / вывода имеют диоды защиты к VDD и VSS.

ТАБЛИЦА 5-1: РЕГИСТРЫ СВЯЗАННЫЕ С ПОРТОМ ВВОДА/ВЫВОДА

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on MCLR and WDT Reset	Value on Wake-up on Pin Change
N/A	TRIS	I/O control registers								--11 1111	--11 1111	--11 1111
N/A	OPTION	GPWU	GPPU	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111	1111 1111
03H	STATUS	GPWUF	—	PA0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	000q quuu	100q quuu
06h	GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx	--uu uuuu	--uu uuuu

5.4 Программирование ввода / вывода

5.4.1 ДВУНАПРАВЛЕННЫХ ПОРТОВ Ввода / вывода

Ряд команд МК выполняются в режиме чтение-модификация-запись (Ч-М-З). Например команды BCF и BSF считывают данные с внешних выводов порта, выполняют разрядную операцию над битом и записывают результат обратно, но в регистр-защелку порта. **Внимание:** необходимо иметь в виду, что не зависимо от того на ввод или на вывод работает тот или иной разряд порта, и не зависимо от того, команда, обращающаяся к порту осуществляет операцию над одним или несколькими разрядами этого порта, данные считываются не с защелки порта, а непосредственно с внешнего вывода, и результат помещается в регистр-защелку порта. Например, команда BSF GPIO,5 считывает данные с внешних выводов порта GPIO в CPU, затем установит в "1" bit5, и пишет новое значение байта в выходную защелку. При выполнении команд, осуществляющих режим Ч-М-З, необходимо учитывать следующее:

1. Если какой-либо разряд порта используется на вывод и на него из МК подан высокий уровень, а внешняя схема поддерживает низкий (например ключ на мощном транзисторе), то при выполнении команд Ч-М-З, с внешнего вывода будет считан низкий уровень, который и будет записан в регистр-защелку порта затирая ее предыдущее значение.

2. Если какой-либо разряд порта (например GP1) используется на вывод, а остальные разряды на ввод, то команда осуществляющая режим Ч-М-З (к примеру BCF GPIO,1) считывает данные с внешних выводов порта, выполняет операцию над разрядом GP1 и поместит результат в его регистр-защелку. При этом в регистры-защелки других разрядов будут занесены те логические уровни, которые находились на соответствующих внешних выводах во время считывания данных при выполнении этой команды. Пока не происходит переконфигурация порта, это не влияет на работу устройства в целом, в случае же перевода на вывод разрядов, работавших на ввод, на внешних выводах этих разрядов появятся значения находящиеся в регистрах-защелках которые были занесены туда при последнем считывании или после выполнения последней команды Ч-М-З.

Вывод: Перед очередной переконфигурацией разрядов порта необходимо или контролировать значения, занесенные в регистры-защелки, или просто внести их.

Внимание. На разряд порта работающий на вывод нельзя подключать нагрузки по схеме "монтажное ИЛИ" или "монтажное И" т.к. возникающие при этом большие токи нагрузки могут вывести микросхему из строя.

ПРИМЕР 5-1: Выполнение операции чтение-модификация-запись на порте

; Начальная установка порта

; GPIO <5:3> Входы

; GPIO <2:0> Выходы

GPIO защелка GPIO выводы

BCF GPIO,5 ;--01 -ppp --11 pppp

BCF GPIO,4 ;--10 -ppp --11 pppp

MOVLW 007h;

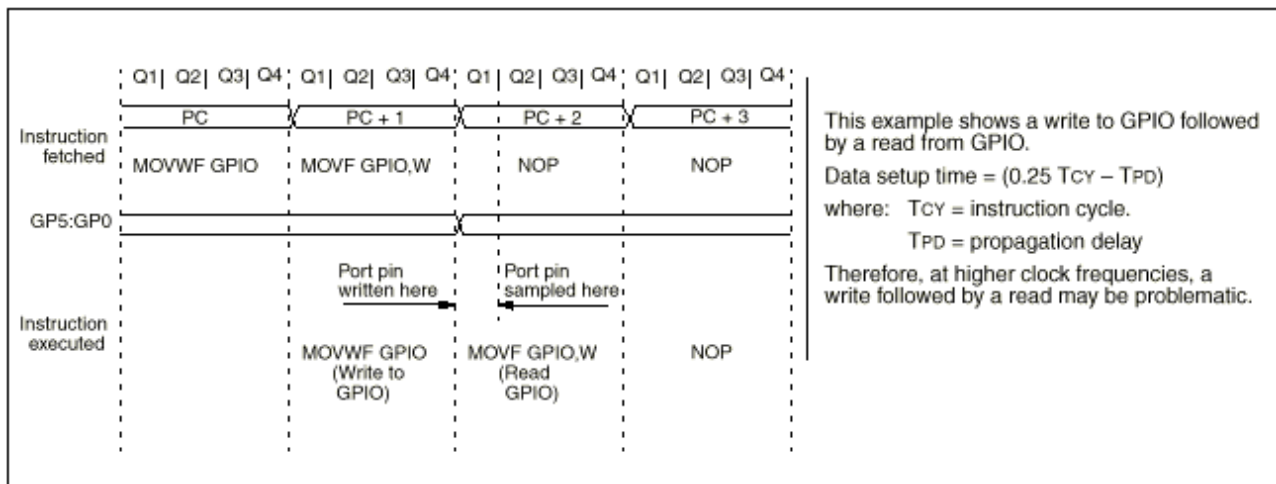
TRIS GPIO ;--10 -ppp --11 pppp

Обратите внимание: ожидалось что значение вывода будет --00 pppp, но вторая команда BCF защелкивает в GP5 высокий уровень.

5.4.2 ПОСЛЕДОВАТЕЛЬНЫЕ ОПЕРАЦИИ НА ПОРТАХ ввода / вывода

Данные на выводах порта ввода / вывода появляются в конце цикла команды, в то время как для чтения, данные должны быть установившимися в начале цикла команды (Рис. 5-2). В случае чтения порта, сразу же после записи в него, должна соблюдаться осторожность. Последовательность команд должна позволить напряжению вывода стабилизироваться прежде, чем следующая команда, считывающая данные с вывода в регистр W, будет выполнена. Иначе, предыдущее состояние вывода может быть считано до того, как установиться новое значение. Лучше отделить эти команды или NOP или другой командой, не обращающейся к этому порту ввода / вывода.

РИСУНОК 5-2: ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАЦИИ Ввода - вывода



6.0 МОДУЛЬ TIMER0 И РЕГИСТР TMR0

Модуль Timer0 имеет следующие особенности:

Читаемый и записываемый 8 битный регистр T/C

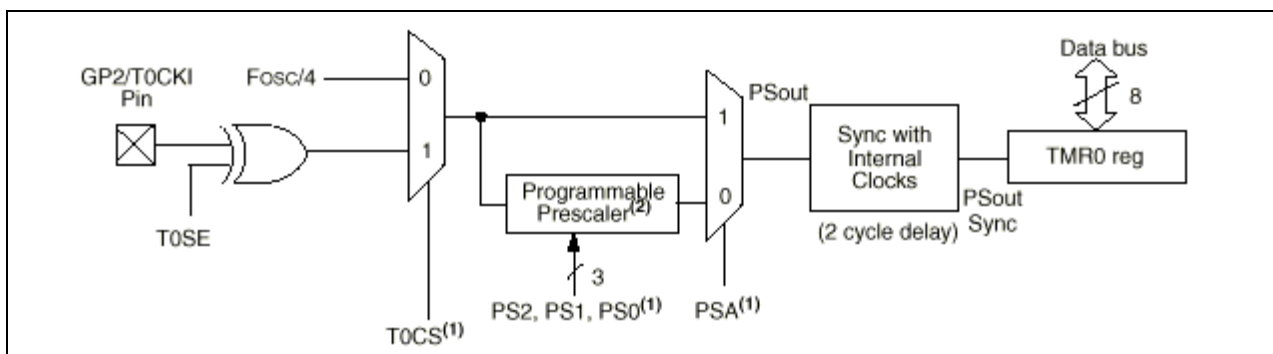
Программируемый 8 битный предварительный делитель (ПД) prescaler

Выбор внутреннего или внешнего сигнала тактирования

Выбор фронта переключения для внешнего сигнала

Упрощенная диаграмма модуля Timer0 показана на Рис. 6-1

РИСУНОК 6-1: БЛОК ДИАГРАММА МОДУЛЯ TIMER0



Примечание

1: Биты T0CS, T0SE, PSA, PS2, PS1 и PS0 размещены в регистре OPTION.

2: Предварительный делитель (ПД) разделен со сторожевым таймером (рис. 6-5).

Режим Таймера выбирается сбросом бита T0CS (OPTION 5). В режиме таймера, модуль Timer0 инкрементируется каждый машинный цикл (без ПД). После записи в регистр TMR0 его инкрементация задерживается на следующие два цикла (Рис 6-2 и Рис. 6-3). Подробное описание смотреть в руководстве по PIC16C8X. Пользователь может это использовать, для записи и коррекции значения регистра.

Режим счетчика выбирается установкой бита T0CS (OPTION < 5 >). В этом режиме, Timer0 инкрементируется по фронту или спаду входного сигнала на выводе T0CKI. Бит T0SE (OPTION < 4 >) определяет фронт переключения. Установка бита T0SE вызывает инкремент TC по спаду входного сигнала. Ограничения на внешний входной сигнал подробно рассмотрены в разделе 6.1.

Предварительный делитель не читаемый и не перезаписываемый модуль, который может использоваться как модулем Timer0 так и Сторожевым таймером, но не обими одновременно. Назначение предделителя управляется программой, битом PSA (OPTION < 3 >). Сброс бита PSA установит Предделитель перед Timer0 а установка - после сторожевого таймера Когда ПД включен перед Timer0 его коэффициент деления устанавливается в пределах 1:2, 1:4, ..., 1:256. Подробно работа Предделителя описана в разделе 6.2. Перечень регистров связанных с модулем Timer0 смотри в таблице 6-1.

РИСУНОК 6-2: ВНУТРЕННЯЯ СИНХРОНИЗАЦИЯ TIMER0 без ПД

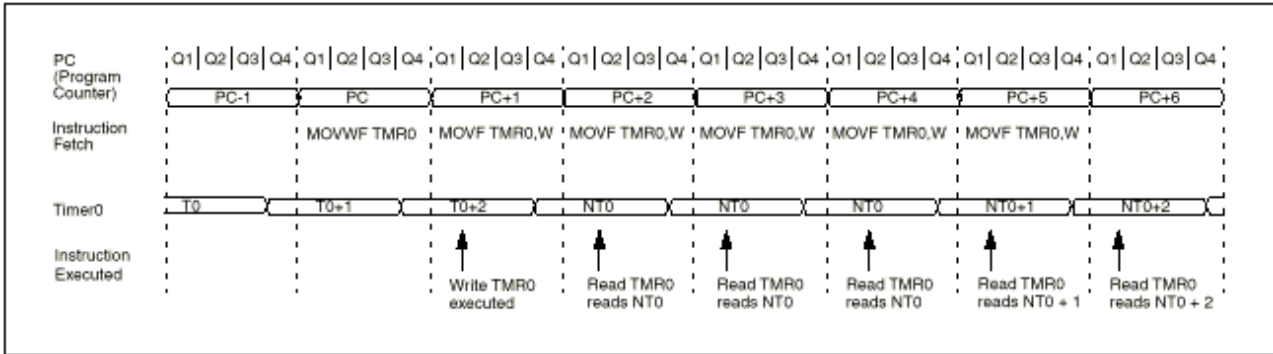


РИСУНОК 6-3: ВНУТРЕННЯЯ СИНХРОНИЗАЦИЯ TIMER0:С ПД 1:2

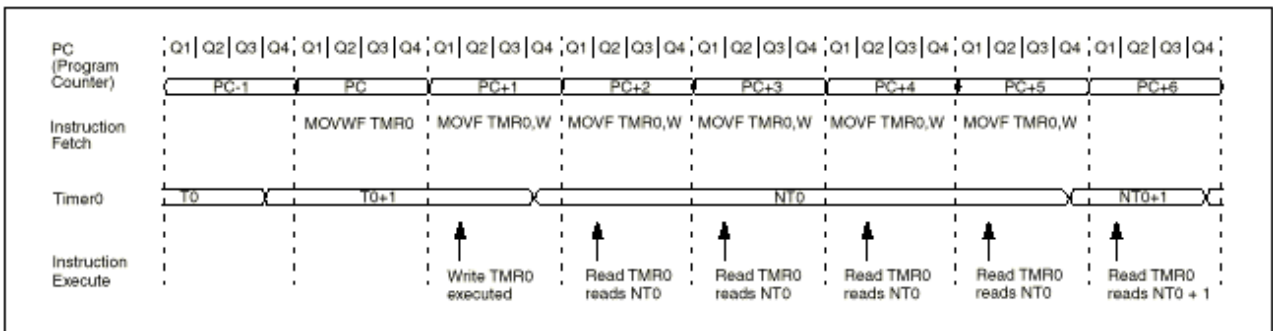


ТАБЛИЦА 6-1: РЕГИСТРЫ СВЯЗАННЫЕ С TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on MCLR and WDT Reset	Value on Wake-up on Pin Change
01h	TMR0	Timer0 - 8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu	uuuu uuuu
N/A	OPTION	GPWU	GPPU	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111	1111 1111
N/A	TRIS	I/O control registers								--11 1111	--11 1111	--11 1111

Примечание.: ячейки, не используемые Timer0: = невыполненно, X = неизвестно u = неизменяемо.

6.1 ИСПОЛЬЗОВАНИЕ TIMER0 С ВНЕШНИМ СИГНАЛОМ

Если для тактирования используется внешний сигнал, то он должен удовлетворять некоторым требованиям для синхронизации с внутренней тактовой частотой. Кроме того, между перепадом на выводе T0CK1 и реальным увеличением счетчика TMR0 есть некоторая задержка.

6.1.1 СИНХРОНИЗАЦИЯ ВНЕШНИМ СИГНАЛОМ

Если ПД не используется, то внешний входной сигнал совпадает с выходом Предделителя. Синхронизация T0CK1 с внутренней тактовой частотой осуществляется фиксацией выхода ПД в тактах Q2 и Q4 внутренней тактовой последовательности (Рисунок 6-4). Следовательно, необходимо на входе T0CK1 сохранять как высокий так и низкий уровень в течении по крайней мере 2 TOSC (и маленькая задержка RC 20 ns). Когда используется, ПД, внешний входной сигнал делится асинхронным счетчиком ПД поэтому выходной сигнал является симметричным. Определяя требования к входному сигналу, в этом случае необходимо учесть работу асинхронного счетчика. Сигнал на входе T0CK1, должен иметь период по крайней мере 4 TOSC (и задержку 40 ns) деленный на коэффициент деления ПД. Единственным ограничением длительности высокого и низкого уровней сигнала на входе T0CK1 в этом случае является минимальная длительность 10 ns.

6.1.2 ЗАДЕРЖКА УВЕЛИЧЕНИЯ TIMER0

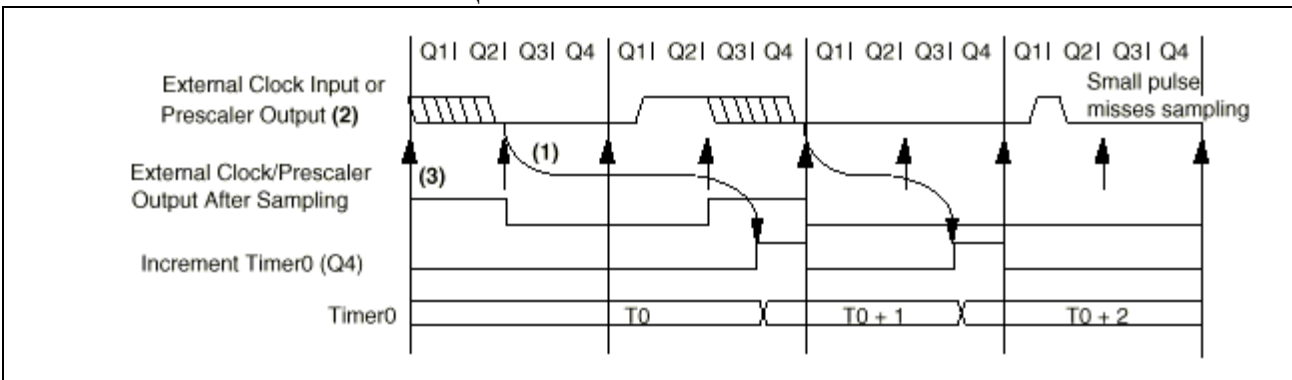
Так как выход Предделителя синхронизирован внутренним сигналом, то имеется небольшая задержка во врем

ени, между перепадом сигнала на выводе TOCK1 и моментом увеличения содержимого Timer0 (рис. 6-4)

6.1.3 ЭФФЕКТ РЕГИСТРА OPTION НА GP2 TRIS

если бит TOCS регистра OPTION установлен (внешняя частота на вход TIMER0) то разряд GP2 порта настраивается на вход, независимо от состояния бита GP2 регистра TRIS.

РИС. 6-4: СИНХРОНИЗАЦИЯ TIMER0 С ВНЕШНИМ СИГНАЛОМ



Примечание 1: Задержка между изменением входного сигнала и увеличения Timer0 составляет от $3T_{osc}$ до $7T_{osc}$. (скважность $Q = T_{osc}$). Следовательно, ошибка в измерении интервала между двумя фронтами входа T/C равна максимум $4T_{osc}$.

2: Внешний сигнал, если ПД не выбран, либо выход Предделителя.

3: Стрелки ↑ указывают момент фиксации.

6.2 ПРЕДВАРИТЕЛЬНЫЙ ДЕЛИТЕЛЬ (prescaler)

8 битный счетчик, для Timer0 доступен как ПД, а для Сторожевого Таймера (WDT) как счетчик переполнения. (Раздел 7.6). Для простоты, в данном описании этот счетчик упоминается как ПД (prescaler). ПД может использоваться или Timer0 модулем или WDT, но не одновременно. Таким образом, назначение ПД для T/C означает, что WDT работает без ПД, и наоборот.

Биты PSA и PS0: PS1: PS2: (OPTION3:0) определяют назначение и коэффициент деления ПД. Когда ПД установлен перед T/C, все команды, записывающие в регистр TMR0 (например, CLRF 1, MOVWF 1, BSF 1, и т.д.) сбрасывают ПД. Когда ПД включен после WDT, CLRWDТ команда сбросит ПД вместе с WDT. ПД не является читаемым, или записываемым модулем. При СБРОСЕ, ПД содержит все нули.

1.1.1 ПЕРЕКЛЮЧЕНИЕ НАЗНАЧЕНИЯ ПД

Пример 6-1 переключение ПД с Timer0 на WDT.

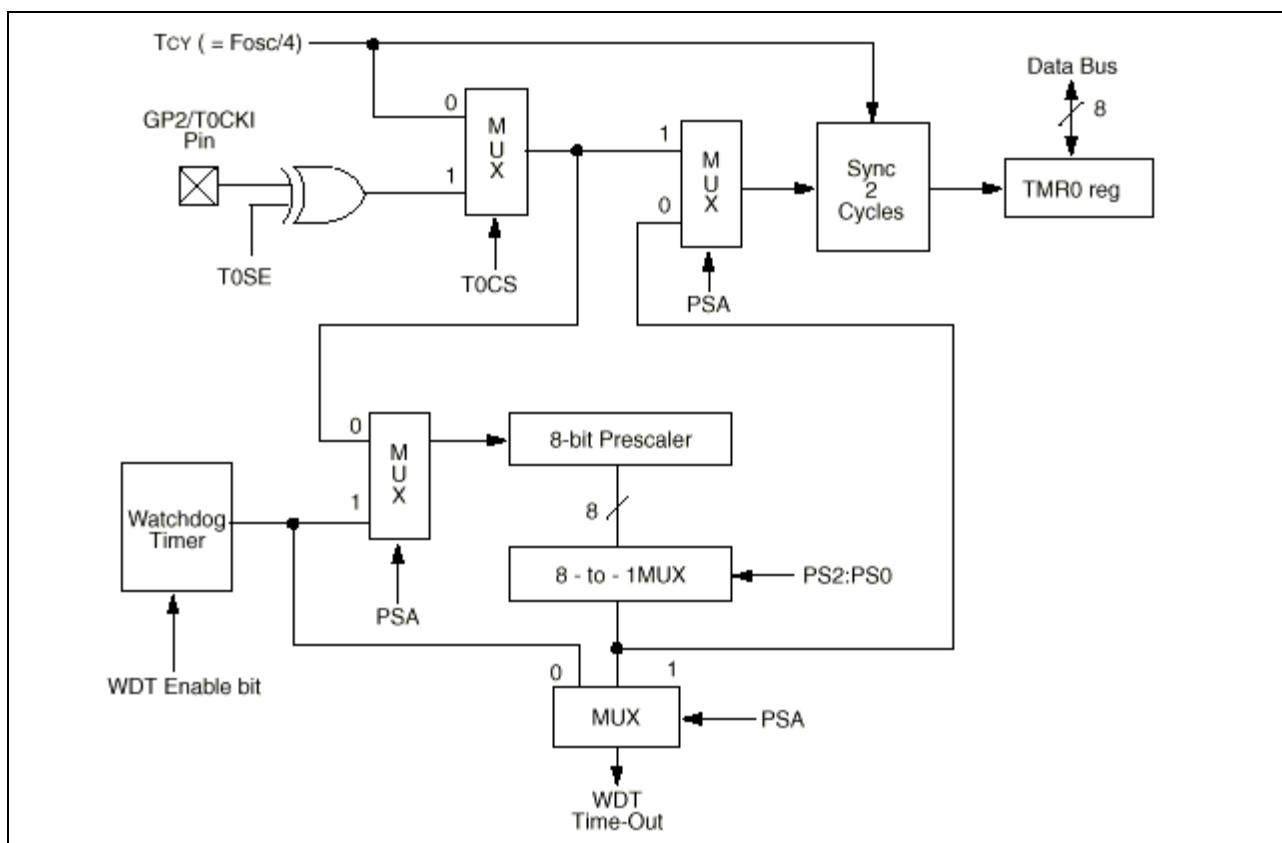
1. CLRWDТ
2. CLRF TMR0
3. MOVLW '00xx1111'b
4. OPTION
5. CLRWDТ
6. MOVLW '00xx1xxx'b
7. OPTION

Пример 6-2 переключение ПД с WDT на T/C

Эта последовательность должна быть выполнена даже в том случае если WDT заблокирован.

CLRWDT команда должна быть выполнена перед переключением ПД.

```
CLRWDT
MOVLW 'xxxx0xxx
OPTION
```

РИСУНОК 6-5: ДИАГРАММА БЛОКА TIMER0/WDT ПД.

Примечание: Биты T0CS, T0SE, PSA, PS2:PS0 находятся в регистре OPTION.

7.0 ОСОБЕННОСТИ МИКРОКОНТРОЛЛЕРА

Семейство микроконтроллеров PIC12C5XX имеет следующие особенности:

выбор Генератора

Сброс

Сброс Мощности (POR)

Таймер Сброса Устройства (DRT)

Режим пониженного энергопотребления SLEEP

Выход из режима SLEEP при изменении уровня на выводе

Сторожевой Таймер (WDT)

Защиту Кода

17

PIC12C5XX

Наличие ИДЕНТИФИКАТОРА

Внутрисхемное программирование

PIC12C5XX имеет Сторожевой Таймер (WDT), который может быть запрещен только через бит конфигурации WDTE. Для надежности, Таймер работает от собственного RC генератора. При выборе XT или LP генератора, всегда имеется задержка, обеспечиваемая Устройством Таймера Сброса (DRT), поддерживающим МК в режиме сброса, пока кварцевый генератор не войдет в устойчивый режим работы. При использовании INTRC или EXTRC генераторов имеется 18 тактов задержки генератора только при включении питания. С этим таймером большинство прикладных программ не нуждается ни в какой внешней схеме сброса.

Режим SLEEP необходим для того, чтобы вводить Микроконтроллер в режим пониженного энергопотребления. Такой режим необходим для экономии энергии источника питания, когда МК находится в режиме ожидания, а также во время холостой работы. Выход из режима SLEEP осуществляется либо путем изменения уровня на внешних выводах сконфигурированных на вход, либо сбросом при переполнении Сторожевого Таймера (если последний был разрешен программой).

Микроконтроллер позволяет выбрать один из 4 типов генераторов, включая IXTRC. Это внутренний 4 MHz RC генератор не требующий внешних частотозадающих цепей, что упрощает конструкцию устройства выполненного на PIC12C5XX и снижает его стоимость, однако более экономичным является LP генератор. Набор битов конфигурации используется, для выбора типа генератора.

7.1 Биты Конфигурации

Слово конфигурации PIC12C5XX состоит из 5 битов. Биты программируются пользователем. Два бита - для выбора типа генератора, и по одному для режима сброса, Сторожевого Таймера и защиты кода (Рисунок 7-1).

РИСУНОК 7-1: СЛОВО КОНФИГУРАЦИИ) ДЛЯ PIC12C508 ИЛИ PIC12C509

-	-	-	-	-	-	MCLR	CP	WDT	FOSC	FOSC
						E		E	1	0

bit11
bit0

Регистр CONFIGURATION

Адрес(1): FFFh

Биты 11-5: Не выполнены

Бит 4: MCLRE:

1 = MCLR внешний сброс (через вывод)

0 = MCLR связан с VDD, (Внутренне)

Бит 3: CP: бит защиты Кода.

1 = Защита Кода есть

0 = Защита Кода нет

Бит 2: WDTE: Сторожевой таймер

1 = WDT включен

0 = WDT выключен

Бит 1-0: FOSC1: FOSC0: биты выбора типа Генератора

1 1 = EXTRC - внешний RC генератор

1 0 = INTRC - внутренний RC генератор

0 1 = XT генератор

0 0 = LP генератор

Примечание 1: Биты регистра устанавливаются при программировании, программой они не адресуются.

7.2 Конфигурации Генератора

7.2.1 ТИПЫ ГЕНЕРАТОРОВ

PIC12C5XX может использовать один из четырех типов генераторов. Пользователь программирует два бита (FOSC1: FOSC0) конфигурации, чтобы выбрать один из этих четырех типов:

LP: Кварц Малой мощности

XT: Кварцевый Резонатор

INTRC: Внутренний генератор 4 MHz

EXTRC: Внешний RC генератор

7.2.2 КВАРЦЕВЫЕ ГЕНЕРАТОРЫ

В XT или LP режимах, кварцевый или керамический резонатор связан с выводами GP5/OSC1/CLKIN и GP4/OSC2 (Рисунок 7-2).

PIC12C5XX требует использования кварцевого резонатора с параллельным резонансом. В режимах XT и LP, микроконтроллер может тактироваться внешним сигналом через вывод GP5/OSC1/CLKIN (Рисунок 7-3).

РИСУНОК 7-2: Генератор XT или LP ТИПА

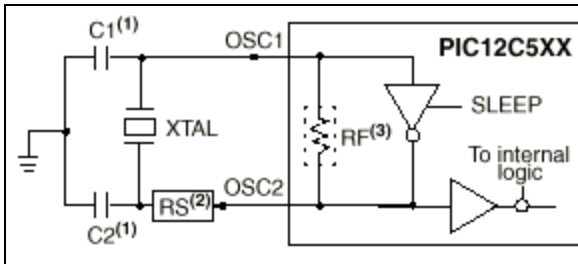


РИСУНОК 7-3: ТАКТИРОВАНИЕ МК ВНЕШНИМ СИГНАЛОМ

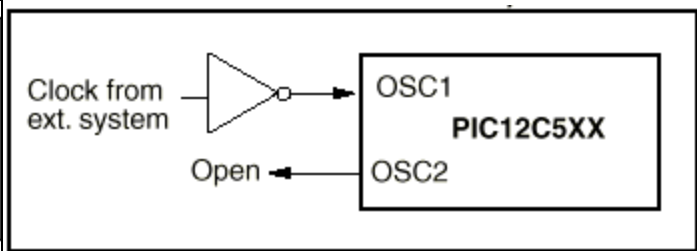


Рисунок 7-1: Выбор конденсаторов для керамических резонаторов.

Osc Type	Resonator Freq	Cap. Range C1	Cap. Range C2
XT	4.0 MHz	30 pF	30 pF

Рисунок 7-2: Выбор конденсаторов для кварцевых резонаторов.

Osc Type	Resonator Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz ⁽¹⁾	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF

Примечание 1: Рекомендуется для VDD > 4.5V, C1 = C2 » 30 pF

Эти значения ориентировочны, так как каждый резонатор имеет собственные характеристики.

Резисторы могут требоваться в режиме XT, чтобы избежать перевозбуждения резонатора с низким уровнем возбуждения.

7.2.5 ВНУТРЕННИЙ RC ГЕНЕРАТОР 4 MHz

Внутренний RC генератор работает на частоте 4 MHz.

Кроме того, команда калибровки программируется в верхнюю часть памяти, которая содержит значение калибровки для внутреннего RC генератора. Это значение программируется как MOVLW XX команда, где XX - значение калибровки. Это загрузит регистр W значением калибровки после сброса, и PC затем перейдет к программе пользователя по адресу 0x000.

Обратите внимание: Стирание программы также сотрет предварительно запрограммированное внутреннее значение калибровки для внутреннего генератора.