

Организация памяти в микроконтроллерах PIC16

(2-я редакция)

Статья основывается на технической документации DS33023a
компании Microchip Technology Incorporated, USA.

**© ООО «Микро-Чип»
Москва - 2001**

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

Организация памяти в микроконтроллерах PIC16

Статья основывается на технической документации DS33023a компании Microchip Technology Incorporated, USA.

1. Введение

Память в микроконтроллерах PIC16 разделена на два независимых блока: память программ и память данных. Каждый блок имеет собственную шину данных и шину адреса, позволяя организовать одновременный доступ к обоим типам памяти в течение одного машинного цикла.

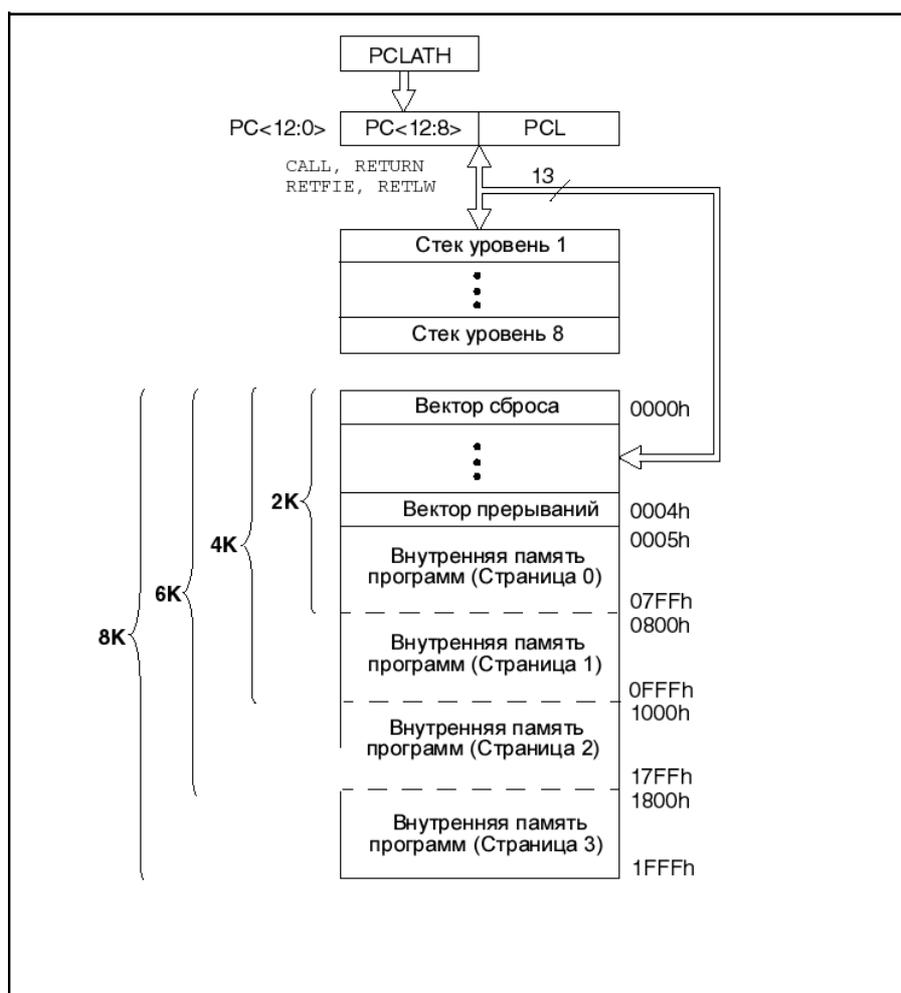
Память данных состоит из регистров общего (GPR) и специального (SFR) назначения. Регистры SFR, управляющие «ядром» микроконтроллера, будут описаны в данной статье. Описание регистров SFR, управляющие периферийными модулями, смотрите в соответствующем разделе документации на микроконтроллер.

2. Организация памяти программ

Микроконтроллеры PIC16 имеют 13-разрядный счетчик команд, способный адресовать 8К x 14 памяти программ, и 14-разрядную шину данных памяти программ. Все инструкции микроконтроллера состоят из 14-разрядного слова, поэтому микроконтроллер с объемом памяти программ 8К x 14 может содержать 8К инструкций. Это позволяет легко определить достаточность памяти программ для желаемого приложения.

Вся память программ разделена на 4 страницы по 2Кслов каждая (0000h–07FFh, 0800h–0FFFh, 1000h–17FFh, 1800h–1FFFh). В зависимости от типа микроконтроллера, только некоторая часть доступной памяти программ реализована аппаратно (смотрите техническую документацию на конкретный микроконтроллер).

На рисунке показана карта памяти программ и 8-уровневый аппаратный стек.



Примечания.

1. Не во всех микроконтроллерах полностью реализовано адресное пространство памяти программ.
2. Калибровочная информация может размещаться в памяти программ.

Для перехода между страницами памяти программ необходимо изменить старшие биты регистра счетчика команд PC, записью в регистр специального назначения PCLATH (старший байт счетчика команд). Изменив значение

регистра PCLATH, и выполнив команду ветвления, счетчик команд PC пересечет границу страницы памяти программ без дополнительного вмешательства пользователя.

Для устройств, имеющих память программ меньше 8Кслов, обращение к памяти программ выше фактически существующего значения приведет к циклической адресации. Например, в микроконтроллере с памятью программ 4Кслов и попытке перехода по адресу 17FFh – переход будет выполнен по адресу 07FFh. В микроконтроллерах с памятью программ 2Кслова управление страницами памяти не требуется.

2.1. Вектор сброса

В любом микроконтроллере PICmicro сброс приведет к очистке счетчика команд (PC), устанавливая адрес 0h. Адрес 0000h называется «адрес вектора сброса», т.к. будет выполнен переход по этому адресу при сбросе микроконтроллера. Вместе со счетчиком команд (PC) очищается регистр PCLATH, устанавливая рабочую страницу памяти программ 0.

2.2. Вектор прерываний

Когда возникает разрешенное прерывание, в счетчик команд PC записывается адрес 0004h, называемый «адрес вектора прерываний», при этом значение регистра PCLATH не изменяется.

Если в подпрограмме обработки прерываний требуется выполнять команды ветвления, то необходимо предварительно записать в регистр PCLATH значение, определяющее нужную страницу памяти программ. Прежде чем регистр PCLATH будет изменен, его значение должно быть сохранено в другом регистре памяти данных, а затем восстановлено перед выходом из подпрограммы обработки прерываний.

2.3. Калибровочная информация

В некоторых типах микроконтроллеров, во время заключительного заводского испытания в памяти программ сохраняется калибровочная информация. Использование калибровочной информации позволяет приложению получать наилучшие результаты работы. Как правило, калибровочная информация сохраняется в конце памяти программ набором инструкций RETLW.

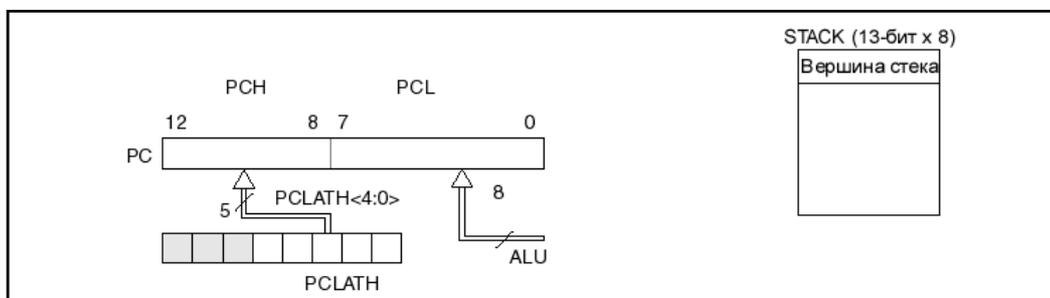
Примечание. Для микроконтроллеров, в которых перед программированием очищается вся память программ, рекомендуется сначала прочитать калибровочную информацию, а затем запрограммировать микроконтроллер.

2.4. Счетчик команд PC

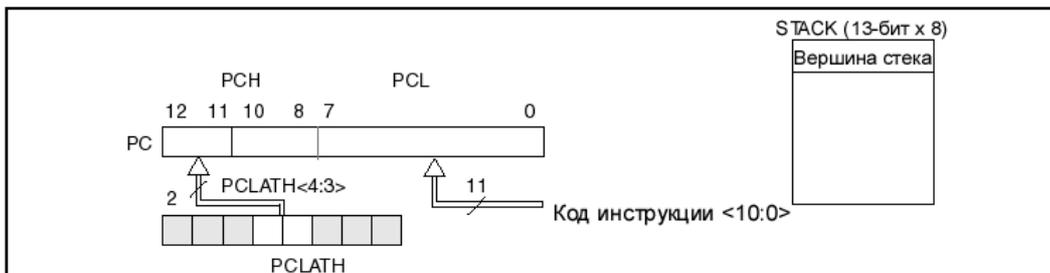
13-разрядный регистр счетчика команд PC указывает адрес выполняемой инструкции. Младший байт счетчика программ PCL доступен для чтения и записи. Старший байт PCH, содержащий <12:8> биты счетчика команд PC, не доступен для чтения и записи. Все операции с регистром PCH происходят через дополнительный регистр PCLATH.

На рисунках показано 4 примера изменения значения счетчика команд PC.

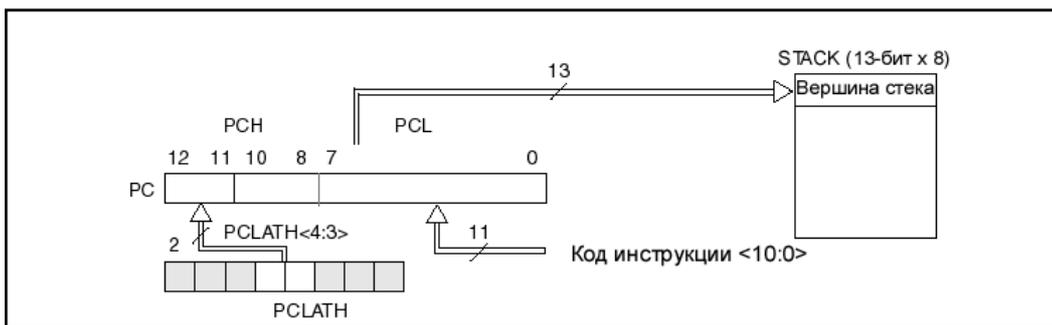
1. Непосредственная запись значения в регистр PCL (PCLATH<4:0> -> PCH)



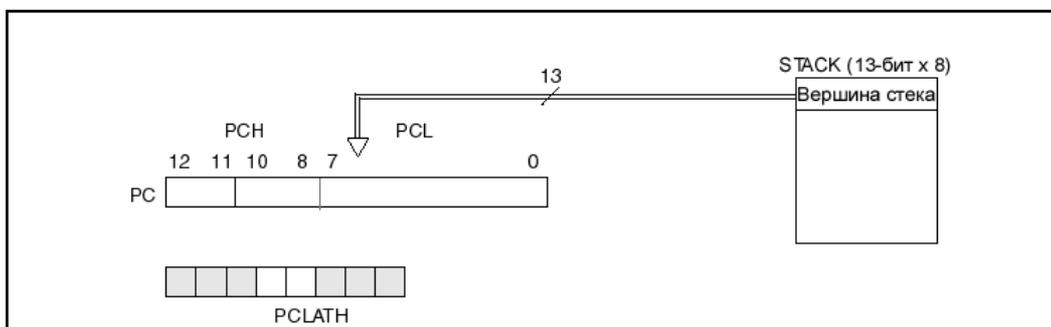
2. Изменение значения PC при выполнении инструкции GOTO (PCLATH<4:3> -> PCH)



3. Изменение значения PC при выполнении перехода к подпрограмме CALL (PCLATH<4:3> -> PCH), причем старое значение PC сохраняется в вершине стека



4. Возвращение из подпрограммы (RETURN, RETFIE или RETLW), счетчик команд загружается значением с вершины стека



Примечание. Регистр PCLATH не изменяется при изменении PCH.

2.4.1. Вычисляемый переход

Вычисляемый переход может быть выполнен командой приращения к регистру PCL (например, ADDWF PCL). При выполнении вычисляемого перехода следует заботиться о том, чтобы значение PCL не пересекло границу блока памяти (каждый блок 256 байт).

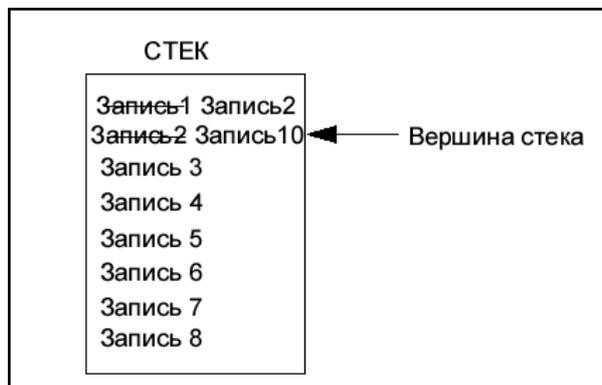
Примечание. При записи значения в регистр PCL, автоматически происходит перезапись 5 младших бит из регистра PCLATH<4:0> в регистр PCH.

2.5. Аппаратный стек

Стек поддерживает до 8 уровней вложенности подпрограмм пользователя, включая обработку прерываний. В стеке сохраняется адрес возврата в основную программу.

В микроконтроллерах семейства PIC16 используется 8-уровневый 13-разрядный аппаратный стек. Стек не имеет отображения на память программ и память данных, нельзя записать или прочитать данные из стека. Значение счетчика команд заносится в вершину стека при выполнении инструкций перехода на подпрограмму (CALL) или обработки прерываний. Чтение из стека и запись в счетчик команд PC происходит при выполнении инструкций возвращения из подпрограммы или обработки прерываний (RETURN, RETLW, RETFIE), при этом значение регистра PCLATH не изменяется.

После 8 записей в стек, девятая запись запишется на место первой, а десятая запись заменит вторую и так далее.



- Примечания.**
1. В микроконтроллерах не имеется никаких указателей о переполнении стека.
 2. В микроконтроллерах не предусмотрено команд записи/чтения из стека, кроме команд вызова/возвращения из подпрограмм (CALL, RETURN, RETLW и RETFIE) или условий перехода по вектору прерываний.

2.6. Страницы памяти программ

Инструкции переходов в микроконтроллерах PIC16 имеют 11-разрядное поле для указания адреса, что позволяет непосредственно адресовать 2Кслов памяти программ. Некоторые микроконтроллеры имеют память программ более 2Кслов. Для адресации верхних страниц памяти программ используются 2 бита в регистре PCLATH<4:3>. Перед выполнением команды перехода (CALL или GOTO) необходимо запрограммировать биты регистра PCLATH<4:3> для адресации требуемой страницы.

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

Примечания.

1. В микроконтроллерах с объемом памяти программ до 2Кслов биты регистра PLATH<4:3> игнорируются. Не рекомендуется их использовать, как биты общего назначения, т.к. может быть потребность переноса программы на микроконтроллер с большим объемом памяти программ.

2. В микроконтроллерах с объемом памяти программ до 4Кслов бит регистра PLATH<4> игнорируется. Не рекомендуется его использовать, как бит общего назначения, т.к. может быть потребность переноса программы на микроконтроллер с большим объемом памяти программ.

Пример перехода со страницы 0 на страницу 1 памяти программ. Этот пример предполагает, что в подпрограмме сохраняется и восстанавливается значение регистра PCLATH.

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

3. Организация памяти данных

Память данных разделяется на регистры двух типов:

- регистры специального назначения (SFR), управляют работой микроконтроллера;
- регистры общего назначения (GPR), для хранения данных программы.

Память данных разделена на банки, содержащие регистры общего и специального назначения. Регистры специального назначения размещаются в разных банках памяти данных для того, чтобы была возможность организовать более 96 байт ОЗУ. Регистры специального назначения предназначены для управления периферийными модулями и функциями микроконтроллера. Управление банками памяти выполняется битами в регистре STATUS<7:5>. На рисунке представлена карта памяти данных одного из микроконтроллеров PIC16.

Адрес		Адрес		Адрес		Адрес	
INDF	00h	INDF	80h	INDF	100h	INDF	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	PORTF	107h	TRISF	187h
PORTD	08h	TRISD	88h	PORTG	108h	TRISG	188h
PORTE	09h	TRISE	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch		10Ch		18Ch
PIR2	0Dh	PIE2	8Dh		10Dh		18Dh
TMR1L	0Eh	PCON	8Eh		10Eh		18Eh
TMR1H	0Fh	OSCCAL	8Fh		10Fh		18Fh
T1CON	10h		90h		110h		190h
TMR2	11h		91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPATAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRES	1Eh		9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
Регистры общего назначения (2)	20h	Регистры общего назначения ⁽³⁾	A0h	Регистры общего назначения ⁽³⁾	120h	Регистры общего назначения ⁽³⁾	1A0h
		Отображаются на банк 0 70h - 7Fh ⁽⁴⁾	EFh	Отображаются на банк 0 70h - 7Fh ⁽⁴⁾	16Fh	Отображаются на банк 0 70h - 7Fh ⁽⁴⁾	1EFh
			F0h		170h		1F0h
	7Fh		FFh		17Fh		1FFh
Банк0		Банк1		Банк2 ⁽⁵⁾		Банк3 ⁽⁵⁾	

Примечания к рисунку.

1. Регистры, названия которых выделены жирным текстом будут присутствовать в каждом типе микроконтроллеров.
2. В некоторых типах микроконтроллеров часть банка памяти данных может быть не реализована. Чтение по адресу не существующего ОЗУ будет давать результат 00h.
3. Эта часть ОЗУ может быть не реализована. Чтение по адресу не существующего ОЗУ будет давать непредсказуемый результат.
4. В некоторых типах микроконтроллеров этот банк отображается на банк 0, для регистров общего назначения.
5. Банк может быть не реализован. Чтение по адресу не существующего ОЗУ будет давать результат 00h.
6. Регистры общего назначения могут быть расположены в области регистров специального назначения.

Чтобы передать данные из одного регистра в другой, необходимо использовать дополнительный регистр W. Эта операция выполняется двумя командами за два машинных цикла микроконтроллера.

Обращение к памяти данных может быть выполнено прямой или косвенной адресацией регистров:

- прямая адресация, для указания банка памяти данных необходимо использовать биты PR1:PR0 регистра STATUS;
- косвенная адресация, адрес регистра сохраняется в FSR, а в бите IRP регистра STATUS указывается к какой паре банков памяти данных выполняется обращение (Банк0/Банк1 или Банк2/Банк3).

3.1. Регистры общего назначения (GRP)

Регистры общего назначения размещаются в разных банках памяти данных. Эти регистры не инициализируются при сбросе по включению питания и имеют неизвестное значение, а при всех остальных сбросах микроконтроллера не изменяют своего значения.

Обращение к регистрам общего назначения может быть выполнено прямой или косвенной адресацией. В некоторых микроконтроллерах существуют регистры общего назначения, адресуемые к одной и той же ячейке ОЗУ, независимо от текущего банка памяти данных. Обратите внимание на эти регистры, т.к. они расположены в общем ОЗУ.

3.2. Регистры специального назначения (SFR)

Регистры специального назначения используются для управления ядром и периферийными модулями микроконтроллера, и реализованы как статическое ОЗУ. Описание регистров SFR, управляющие периферийными модулями, смотрите в соответствующем разделе документации на микроконтроллер.

Регистры специального назначения размещены в различных банках памяти данных, а некоторые из регистров отображаются во всех банках.

Переключение рабочего банка памяти выполняется настройкой битов PR0:PR1 регистра STATUS. При сбросе микроконтроллера в регистры специального назначения записывается определенное значение (см. техническую документацию на соответствующий микроконтроллер).

Обращение к регистрам специального назначения может быть выполнено прямой или косвенной адресацией.

Примечание. В области регистров специального назначения могут размещаться регистры общего назначения.

3.3. Банки памяти данных

Память данных разделена на 4 банка. Каждый банк содержит регистры специального назначения (в начале адресного пространства банка) и регистры общего назначения. Переключение между банками осуществляется с помощью битов:

- PR1:PRO в регистре STATUS, при прямой адресации;
- IRP в регистре STATUS, при косвенной адресации.

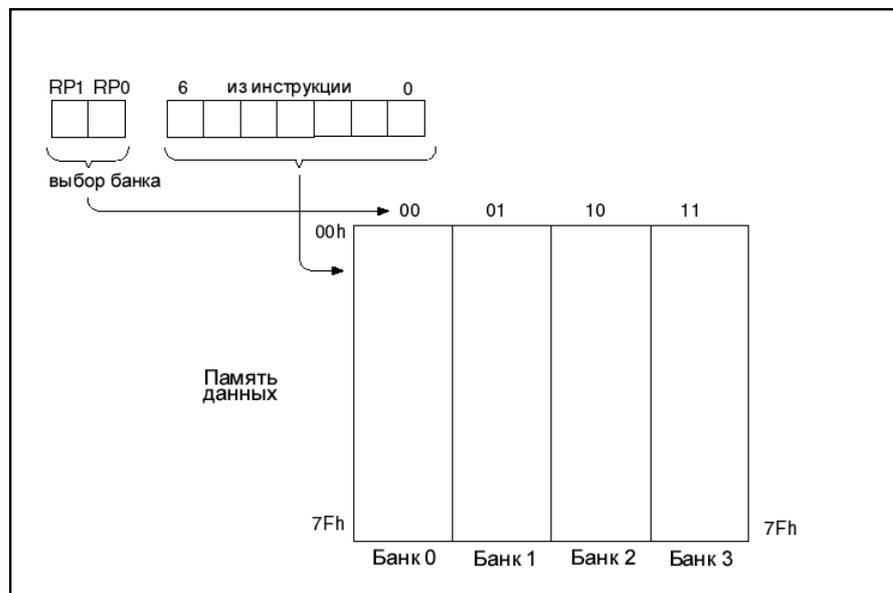
Доступ к банку	Прямая адресация (PR1:PR0)	Косвенная адресация IRP
0	0 0	0
1	0 1	
2	1 0	1
3	1 1	

Вся память данных выполнена по технологии статического ОЗУ с максимальный размер банка памяти 7Fh (128байт). В начале банка располагаются регистры специального назначения, за ними регистры общего назначения. Некоторые часто используемые регистры специального назначения банка 0 отображаются в других банках памяти для упрощения программного обеспечения и получения быстрого доступа к ним.

В процессе разработки новых микроконтроллеров, адреса размещения регистров специального назначения в памяти данных, претерпели некоторые изменения. Организация памяти данных, показанная на рисунке в начале раздела, является стандартом для всех новых микроконтроллеров. Последние 16 байт всех банков памяти данных, показанных на карте, отображаются на банк 0, что должно упростить программное обеспечение, работающее с банками памяти данных. Регистры, имена которых выделены полужирным текстом, присутствуют во всех микроконтроллерах.

Состав регистров специального назначения, их адреса в памяти данных и объем регистров общего назначения для конкретного типа микроконтроллера смотрите в технической документации на соответствующий микроконтроллер.

Механизм прямой адресации памяти данных



Пример программы переключения банков при прямой адресации.

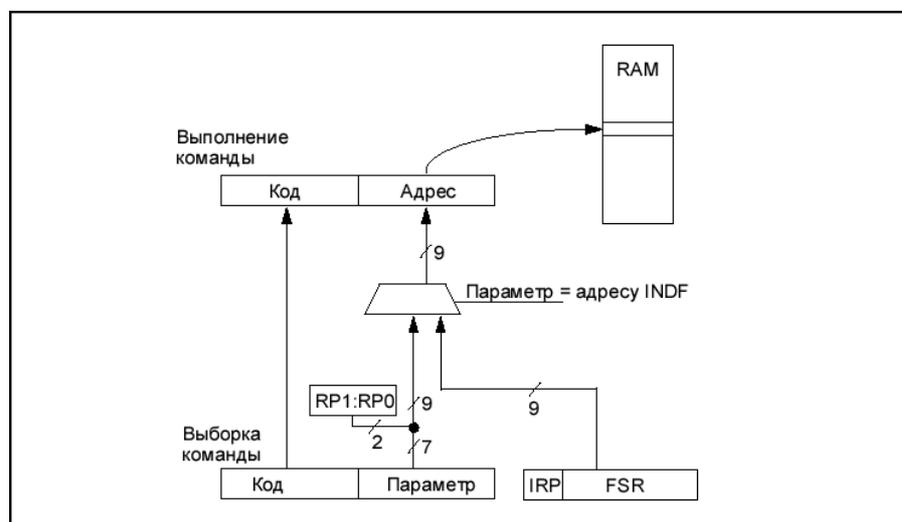
```

CLRf  STATUS      ; Очистка STATUS регистра (Банк 0)
:
BSF   STATUS, RP0  ; Банк 1
:
BCF   STATUS, RP0  ; Банк 0
:
MOVLW 0x60        ; Установить RP0 и RP1 в STATUS регистре
XORWF STATUS, F   ; (Банк 3)
:
BCF   STATUS, RP0  ; Банк 2
:
BCF   STATUS, RP1  ; Банк 0

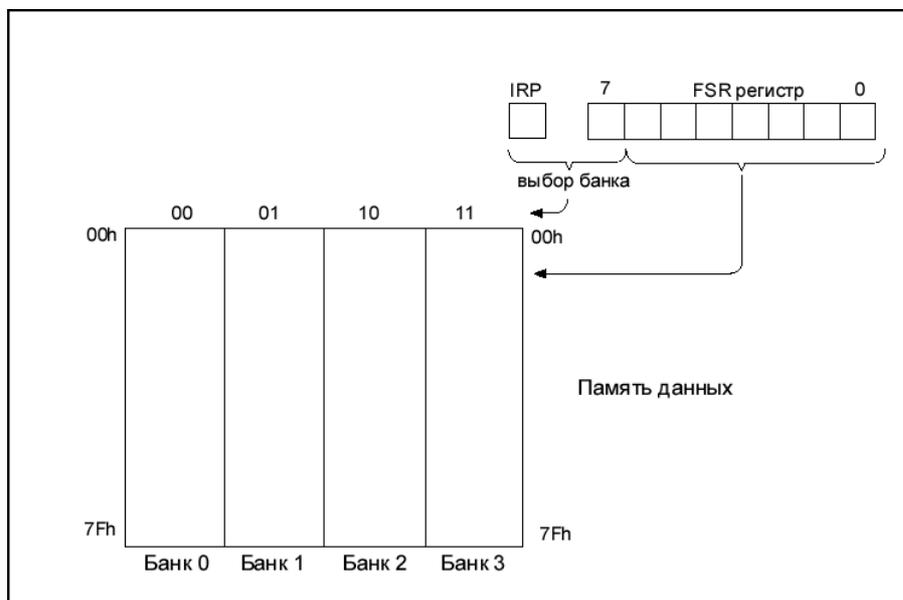
```

3.4. Косвенная адресация, регистры INDF и FSR

Косвенная адресация – такой режим адресации регистров, при котором в команде адрес памяти данных не указывается. Регистр специального назначения FSR используется в качестве указателя адреса в памяти данных, доступный для записи/чтения пользователем. Этот режим адресации может быть полезен при обращении к таблицам данных. На рисунке показан механизм записи значения в регистр косвенной адресацией памяти данных.



Для выполнения косвенной адресации необходимо обратиться к регистру INDF. Обращение к регистру INDF фактически вызовет действие с регистром с адресом, указанным в FSR. Косвенное чтение регистра INDF (FSR=0) даст результат 00h. Косвенная запись в регистр INDF не вызовет никаких действий (вызывает воздействия на флаги АЛУ в регистре STATUS). 9-бит косвенного адреса IRP сохраняется в регистре STATUS<7>. Пример 9-разрядной косвенной адресации показан на рисунке.



В примере программы показано использование косвенной адресации для очистки памяти данных (диапазон адресов 20h – 2Fh) минимальным числом команд микроконтроллера. Подобный метод может использоваться для передачи блока данных в регистр TXREG передатчика USART. Начальный адрес блока данных, подготовленного для передачи, может быть легко изменен в соответствии с требованиями программы.

```

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

CONTINUE:
; Да

```

4. Пример программы инициализации регистров общего назначения.

```

Bank0_LP
    CLRF    STATUS    ; Очистить регистр STATUS (Банк 0)
    MOVLW  0x20      ; 1-й адрес регистра GPR
    MOVWF  FSR       ; записать в регистр косвенного адреса

    CLRF    INDF0    ; Очистить регистр GPR с адресом в регистре FSR
    INCF   FSR       ; Следующий регистр GPR
    BTFSS  FSR, 7    ; Очистка регистров GPR в этом банке завершена? (FSR = 80h, C = 0)
    GOTO   Bank0_LP ; НЕТ, продолжать очистку
                ;
                ; Следующий банк (Банк 1)
                ; (** Только для микроконтроллеров с банком 1 **)
                ;

    MOVLW  0xA0      ; 1-й адрес регистра GPR
    MOVWF  FSR       ; записать в регистр косвенного адреса

Bank1_LP
    CLRF    INDF0    ; Очистить регистр GPR с адресом в регистре FSR
    INCF   FSR       ; Следующий регистр GPR
    BTFSS  STATUS, C ; Очистка регистров GPR в этом банке завершена? (FSR = 00h, C = 1)
    GOTO   Bank1_LP ; НЕТ, продолжать очистку
                ;
                ; Следующий банк (Банк 2)
                ; (** Только для микроконтроллеров с банком 2 **)
                ;

    BSF    STATUS, IRP ; Выбор банков 2 и 3
                ; для косвенной адресации

    MOVLW  0x20      ; 1-й адрес регистра GPR
    MOVWF  FSR       ; записать в регистр косвенного адреса

Bank2_LP
    CLRF    INDF0    ; Очистить регистр GPR с адресом в регистре FSR
    INCF   FSR       ; Следующий регистр GPR
    BTFSS  FSR, 7    ; Очистка регистров GPR в этом банке завершена? (FSR = 80h, C = 0)
    GOTO   Bank2_LP ; НЕТ, продолжать очистку
                ;
                ; Следующий банк (Банк 3)
                ; (** Только для микроконтроллеров с банком 3 **)
                ;

    MOVLW  0xA0      ; 1-й адрес регистра GPR
    MOVWF  FSR       ; записать в регистр косвенного адреса

Bank3_LP
    CLRF    INDF0    ; Очистить регистр GPR с адресом в регистре FSR
    INCF   FSR       ; Следующий регистр GPR
    BTFSS  STATUS, C ; Очистка регистров GPR в этом банке завершена? (FSR = 00h, C = 1)
    GOTO   Bank3_LP ; НЕТ, продолжать очистку
    :       ; ДА, все регистры GPR очищены

```

Статья основывается на технической документации DS33023a
компании Microchip Technology Incorporated, USA.

Уважаемые господа!

ООО «Микро-Чип» поставляет полную номенклатуру комплектующих фирмы **Microchip Technology Inc** и осуществляет качественную техническую поддержку на русском языке.

С техническими вопросами Вы можете обращаться по адресу support@microchip.ru

По вопросам поставок комплектующих Вы можете обращаться к нам по телефонам:
(095) 963-9601
(095) 737-7545
и адресу sales@microchip.ru

На сайте

www.microchip.ru

Вы можете узнать последние новости нашей фирмы, найти техническую документацию и информацию по наличию комплектующих на складе.