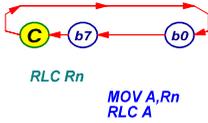
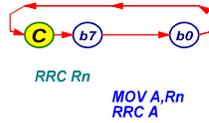


Сводная таблица 12/14 битовых команд PIC микроконтроллеров с аналогиями MCS51 & AVR Tiny&Mega.

ADDWF f,d <i>ADD Rn,A</i> ADD A,Rn <i>JMP PCLATH:(PCL+A)</i>	SUBWF f,d <i>SUB Rn,A</i> <i>MOV B,Rn</i> <i>SUB B,A</i> <i>XCH A,B</i> <i>JMP PCLATH:(PCL-A)</i>	ANDWF f,d <i>AND Rn,A</i> AND A,Rn <i>JMP PCLATH:(PCL AND A)</i>	IORWF f,d <i>OR Rn,a</i> OR a,Rn <i>JMP PCLATH:(PCL OR A)</i>	XORWF f,d <i>XOR Rn,A</i> XOR A,Rn <i>JMP PCLATH:(PCL XOR A)</i>	MOVF f,d R-R <i>MOV Rn,Rn</i> <i>(MOV PORTn, PINn)</i> <i>(TEST f)</i> <i>MOV A,Rn</i> <i>(MOVWF f)</i> <i>(MOV A,@FSR)</i>	MOVWF f ✦ <i>MOV Rn,A</i> <i>MOVWF INDF =</i> <i>MOV @FSR,A</i> <i>JMP PCLATH:A</i>	COMF f,d R-R <i>CPL Rn</i> <i>MOV A,Rn</i> <i>CPL A</i> <i>JMP PCLATH:(NOT PCL)</i>					
ADDLW k8 ADD A,#k8	SUBLW k8 SUB #k8,A <i>результат в регистре A !!!</i>	ANDLW k8 AND A,#k8	IORLW k8 OR A,#k8	XORLW k8 XOR A,#k8	MOVLW k8 ✦ MOV A,#k8	CLRW CLR A	CLRF f R-R <i>CLR Rn</i> <i>CLR INDF =</i> <i>CLR @FSR</i> <i>JMP PCLATH:0x00</i>					
DECF f,d R-R <i>DEC Rn</i> <i>MOV A,Rn</i> <i>DEC A</i>	INCF f,d R-R <i>INC Rn</i> <i>MOV A,Rn</i> <i>INC A</i>	DECFSZ f,d R-R ✦ <i>DJZ Rn,rstep</i> <i>MOV A,Rn</i> <i>DJZ A,rstep</i>	INCFSZ f,d R-R ✦ <i>INC Rn</i> <i>JZ rstep</i> <i>MOV A,Rn</i> <i>INC A</i> <i>JZ rstep</i>	RLF f,d R-R 	RRF f,d R-R 	BCF f,b R-R ✦ <i>CLR Rn.bit</i> <i>BCF INDF =</i> <i>CLR @FSR.bit</i>	BSF f,b R-R ✦ <i>SET Rn.bit</i> <i>BSF INDF =</i> <i>SET @FSR.bit</i>					
GOTO addr ✦ <i>JMP</i> <i>PCLATH<4:3>:addr</i>	CALL addr ✦ <i>CALL</i> <i>PCLATH<4:3>:addr</i>	RETLW k8 ✦ <i>MOV A,#k8</i> <i>RET</i>	RETURN ✦ <i>RET</i>	RETFIE ✦ <i>RETI</i>	SWAPF f,d R-R ✦ <i>SWAP Rn</i> <i>MOV A,Rn</i> <i>SWAP A</i> <i>JMP PCLATH:(SWAP PCL)</i>	BTFSC f,b R-R ✦ <i>JNB Rn.bit,rstep</i> <i>BTFSC INDF =</i> <i>JNB @FSR.bit,rstep</i>	BTFSS f,b R-R ✦ <i>JB Rn.bit,rstep</i> <i>BTFSS INDF =</i> <i>JB @FSR.bit,rstep</i>					
NOP ✦ NOP	CLRWDT ✦ WDR	SLEEP SLEEP	TRIS ✦ MOV TRIS(r),A	OPTION ✦ MOV OPTION,A	<div style="border: 1px solid black; padding: 5px;"> <p>кольцевой сдвиг W через C невозможен требуется использование вспомогательного POH</p> <p>FOКУСЫ С W</p> <pre>setb W,n = iorlw 1<<n clr W,n = andlw 0xff-(0xff&(1<<n)) inc W = addlw .1 dec W = addlw 0xff n= 0-7 или ИМЯ</pre> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>ПРИ ВЫЧИТАНИИ:</p> <table border="1"> <tr><td>Z=0, C=1</td><td>результат положительный</td></tr> <tr><td>C=Z=1</td><td>результат равен 0</td></tr> <tr><td>C=Z=0</td><td>результат отрицательный</td></tr> </table> </div>	Z=0, C=1	результат положительный	C=Z=1	результат равен 0	C=Z=0	результат отрицательный
Z=0, C=1	результат положительный											
C=Z=1	результат равен 0											
C=Z=0	результат отрицательный											

k8 = 0 - 255
d = 0 (w) или 1 (f)
(по умолчанию d=1)
12bit
0<f<0x1F (31)
14bit
0<f<0x7F (127)
0<addr<2047
0<bit<7
5<r<7
rstep=пропуск команды
обращение к INDF (0x00)
в качестве f(Rn)
выполняет обращение к регистру,
адрес которого находится в регистре FSR (0x04)
... INDF,d = ... @FSR,d

Rn=F=0-127 - адрес регистра в текущем банке (стандартная прямая адресация)
rstep = переход на +1 команду (AVR)

Обращение к INDF равноценно обращению к ячейке ОЗУ, адрес которой указан в FSR (@FSR для mcs51)

Варианты ассемблерной мнемоники:

КОП регистр,1 = КОП регистр,F = КОП регистр,f = КОП регистр (по умолчанию)

КОП регистр,0 = КОП регистр,W = КОП регистр,w

где "регистр" = объявленное имя регистра или его адрес в текущем банке ОЗУ

Для операций табличного чтения данных и относительных вычисляемых переходов за нулевой адрес (смещение = 0) принимается адрес ячейки памяти, следующей за командой, модифицирующей PCL !!! **addwf PCL,f при W=0xFF = "вечный цикл" !!!**
Применение команд, модифицирующих PCL требует особого внимания к размещению команд в адресном пространстве и величине результирующего значения в PCL, так как требуется корректное значение результата данных в паре PCLATH4-0:PCL !!! чтение PCL содержимого PCLATH не изменяет !!!

MOVF модифицирует Z
согласно содержимого
регистра назначения !!!

Когда регистр ввода/вывода используется для модификации самого себя (MOVWF PORTn,1)
то для записи будут использованы значения с вывода, а не из защелки порта.
/только если порт сконфигурирован как выходной/

шестнадцатеричное :	h'9f=0x9F
десятичное :	d'100=100
восьмеричное :	o'77'
двоичное :	b'01010011'
ASCII :	a'c='c'

- не рекомендуется к применению
- неприменимо в системе 12-бит
- аналог при d=1 (->f)
- при Rn=INDF (d=1)
- при d=1 адресат = PCL
- аналог при d=0 (->w)
- без альтернативы (A или Rn)
- ✦ - на флаги не воздействует

R-R — данная операция может выполняться непосредственно с регистром без участия аккумулятора

запись в PCL вызывает автоматическую загрузку PCLATH -> PCH

