

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

ADDWF f,d <i>ADD Rn,A</i> <i>ADD A,Rn</i> <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> <i>AND A,Rn</i> <i>JMP PCLATH:(PCL AND A)</i>	IORWF f,d <i>OR Rn,a</i> <i>OR a,Rn</i> <i>JMP PCLATH:(PCL OR A)</i>	XORWF f,d <i>XOR Rn,A</i> <i>XOR A,Rn</i> <i>JMP PCLATH:(PCL XOR A)</i>	MOVF f,d R-R <i>MOV Rn,Rn</i> <i>(MOV PORTn,PINn)</i> (TEST f) <i>MOV A,Rn</i> (MOVFW f) <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 <i>ADD A,#k8</i>	SUBLW k8 <i>SUB #k8,A</i> <i>результат</i> <i>!!! в регистре A !!!</i>	ANDLW k8 <i>AND A,#k8</i>	IORLW k8 <i>OR A,#k8</i>	XORLW k8 <i>XOR A,#k8</i>	MOVLW k8 ✱ <i>MOV A,#k8</i>	CLRW <i>CLR A</i>	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 <i>RLC Rn</i> <i>MOV A,Rn</i> <i>RLC A</i>	RRF f,d R-R <i>RRC Rn</i> <i>MOV A,Rn</i> <i>RRC A</i>	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 ✱ <i>NOP</i>	CLRWDT <i>WDR</i>	SLEEP <i>SLEEP</i>	TRIS ✱ <i>MOV TRIS(r),A</i>	OPTION ✱ <i>MOV OPTION,A</i>	кольцевой сдвиг W через C невозможен требуется использование вспомогательного ПОН фактически с W 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		

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 !!! addw 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 (MOVFW PORTn,1)
то для записи будут использованы значения с вывода, а не из защелок 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

