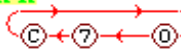
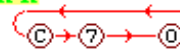


Сводная таблица 12/14 битовых команд PIC микроконтроллеров с аналогиями MCS51 & AVR Tiny & Mega.

ADDWF f, d ADD Rn, A ADD A, Rn JMP PCLATH:(PCL+A)	SUBWF f, d SUBB Rn, A MOV B, Rn SUB B, A XCH A, B JMP PCLATH:(PCL-A)	ANDWF f, d AND Rn, A AND A, Rn JMP PCLATH:(PCL and A)	IORWF f, d OR Rn, A OR A, Rn JMP PCLATH:(PCL or A)	XORWF f, d XOR Rn, A XOR A, Rn JMP PCLATH:(PCL xor A)	MOVF f, d MOV Rn, Rn (TEST Rn) (MOV PORTn, PINn) MOV A, Rn MOVF INDF, 0 = MOV A, @FSR JMP PCLATH:A	MOVWF f * MOV Rn, A MOVWF INDF = MOV @FSR, A JMP PCLATH:A	COMF f, d CPL Rn MOV A, Rn CPL A JMP PCLATH:PCL
ADDLW k ADD A, #k	SUBLW k SUB #d, A ! результат в регистре A	ANDLW k AND A, #k	IORLW k OR A, #k	XORLW k XOR A, #k	MOVLW k * MOV A, #k	CLRWF CLR A	CLRF f CLR Rn CLR INDF = CLR @FSR JMP PCLATH:0x00
DECf f, d DEC Rn MOV A, Rn DEC A	INCF f, d INC Rn MOV A, Rn INC A	DECFSZ f, d * DJZ Rn, rstep MOV A, Rn DJZ A, rstep	INCFSZ f, d * INC Rn JZ rstep MOV A, Rn INC A JZ rstep	RLF f, d  RLC Rn MOV A, Rn RLC A	RRF f, d  RRC Rn MOV A, Rn RRC A	BCF f, b * CLR Rn.b BCF INDF.b = CLR @FSR.b	BSF f, b * SET Rn.b BSF INDF.b = SET @FSR.b
GOTO addr * AJMP addr адрес страницы в ! PCLATH !	CALL addr * ACALL addr адрес страницы в ! PCLATH !	RETLW k * MOV A, #k RET	RETURN * RET	RETFIE * RETI	SWAPF f, d * SWAP Rn MOV A, Rn SWAP A JMP PCLATH:(SWAP PCL)	BTFSF f, b * JNB Rn.b, rstep BTFSF INDF.b = JNB @FSR.b, rstep	BTSS f, b * JB Rn.b, rstep BTSS INDF.b = JB @FSR.b, rstep
NOP * NOP	CLRWDI WDR	SLEEP SLEEP	TRIS r * MOV TRIS(r), A	OPTION * MOV OPTION, A	кольцевой сдвиг W через C невозможен - требуется использование вспомогательного PCH		

k=0-255
d= 0 или 1
(по умолчанию d=1)
12bit
0<f<1F(31)
14bit
0<f<7F(127)
0<addr<2047
0<b<7
5≤r≤7
rstep - пропуск команды

- не рекомендуется к применению
обращение к INDF(0x00) в качестве f(Rn) выполняет обращение к регистру, адрес которого находится в регистре FSR(0x04)
... indf, d=
... @fsr, d

запись в PCL вызовет автоматическую загрузку PCLATH в PCH
PCLATH
7 6 5 4 3 2 1 0

адрес страницы памяти программ при командах GOTO и CALL

Rn=F0-127 - адрес регистра в текущем банке.

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

Когда регистр порта ввода/вывода используется для модификации самого себя (MOVF PORTn, I), то для записи будут использованы значения с выводов, а не из защепок порта.

! только если порт сконфигурирован как входной!

Варианты ассемблерной мнемоники:

MOVF модифицирует 2 Коп регистр, l = Коп регистр, F = Коп регистр, f = Коп регистр (по умолчанию)

согласно содержимого !!!

Коп регистр, 0 = Коп регистр, W = Коп регистр, w

регистр назначения

регистр-объявленное имя

rstep=переход на +1 команду (AVR)

регистр или его адрес в ОЗУ

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

шестнадцатеричное h'9f' = 0x9f
десятичное d'100' = .100
восьмеричное o'77'
двоичное b'01010011'
ASCII a'c' = 'c'

setb W, n = n=0-7 или имя
iorwf 1<<.n
clr W, n =
andlw 0xff-(0xff&(1<<.n))
inc W =
addlw .1
dec W =
addlw 0xff

при вычитании:
C=1 > результат положительный
C=Z=1 результат = 0
C=Z=0 результат отрицательный

- не применимо в системе 12бит
- аналог при d=1
- при Rn=INDF (d=1)
- d=1 адресат=PCL
- аналог при d=0
- без альтернативы (A или Rn)
* - на флаги не воздействует



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 данных в паре PCLATH 4-0:PCL

чтение PCL содержащее PCLATH не изменяет