

Сводная таблица команд PIC12/16 XLP улучшенной группы 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 <i>R-R</i> <i>MOV Rn,Rn</i> (<i>MOV PORTn,PINn</i>) (<i>TEST f</i>) <i>MOV A,Rn</i> (<i>MOVFW f</i>) (<i>MOV A,@FSR</i>)	MOVWF f <i>MOV Rn,A</i> <i>MOVWF INDF = MOV @FSR,A</i> <i>JMP PCLATH:A</i>	COMF f,d <i>R-R</i> <i>CPL Rn</i> <i>MOV A,Rn</i> <i>CPL A</i> <i>JMP PCLATH:(NOT PCL)</i>	CLRF f <i>R-R</i> <i>CLR Rn</i> <i>CLR INDF = CLR @FSR</i> <i>JMP PCLATH:0x00</i>	CLRW <i>CLR A</i>
ADDLW k8 <i>ADD A,#k8</i>	SUBLW k8 <i>SUB #k8,A</i> <i>результат 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>	MOVLB k4 <i>MOV BSR,#k4</i> (<i>0<k4<15</i>)	MOVLPL k7 <i>MOV PCLATH,#k7</i> (<i>0<k7<127</i>)	ADDFSR m,k6 <i>ADD FSRmH:FSRmL,#k6</i> (<i>0<m<1</i>) (<i>-31<k6<31</i>)	ASRF f,d <i>R-R</i> <i>ASRF Rn</i> <i>MOV W,Rn</i> <i>ASRF W</i>
ADDWFC f,d <i>ADDC Rn,A</i> <i>ADDC A,Rn</i>	SUBWFB f,d <i>SUBB Rn,A</i> <i>MOV B,Rn</i> <i>SUBB B,A</i> <i>XCH A,B</i>	DECF f,d <i>R-R</i> <i>DEC Rn</i> <i>MOV A,Rn</i> <i>DEC A</i>	INCF f,d <i>R-R</i> <i>INC Rn</i> <i>MOV A,Rn</i> <i>INC A</i>	DECFSZ f,d <i>DJZ Rn,rstep</i> <i>MOV A,Rn</i> <i>DJZ A,rstep</i>	INCFSZ f,d <i>INC Rn</i> <i>JZ rstep</i> <i>MOV A,Rn</i> <i>INC A</i> <i>JZ rstep</i>	BCF f,b <i>CLR Rn.bit</i> <i>BCF INDF = CLR @FSR.bit</i>	BSF f,b <i>SET Rn.bit</i> <i>BSF INDF = SET @FSR.bit</i>	LSLFL f,d <i>R-R</i> <i>LSLFL Rn</i> <i>MOV W,Rn</i> <i>LSLFL W</i>	LSRFL f,d <i>R-R</i> <i>LSRFL Rn</i> <i>MOV W,Rn</i> <i>LSRFL W</i>
GOTO addr <i>JMP PCLATH<6:5:4:3>:addr</i>	CALL addr <i>CALL PCLATH<6:5:4:3>:addr</i>	RETLW k8 <i>MOV A,#k8</i> <i>RET</i>	RETURN <i>RET</i>	RETFIE <i>RETI</i>	SWAPF f,d <i>SWAP Rn</i> <i>MOV A,Rn</i> <i>SWAP A</i> <i>JMP PCLATH:(SWAP PCL)</i>	BTFSC f,b <i>JNB Rn.bit,rstep</i> <i>BTFSC INDF = JNB @FSR.bit,rstep</i>	BTFSS f,b <i>JB Rn.bit,rstep</i> <i>BTFSS INDF = JB @FSR.bit,rstep</i>	RLF f,d <i>RLC Rn</i> <i>MOV A,Rn</i> <i>RLC A</i>	RRF f,d <i>RRC Rn</i> <i>MOV A,Rn</i> <i>RRC A</i>
BRA adr <i>SJMP adr</i> (<i>-256<adr<255</i>)	CALLW <i>CALL PCLATH:A</i>	BRW <i>JMP PC+A</i>	RESET <i>software device RESET</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>	
MOVIW ++FSRn <i>INC FSRn</i> <i>LD A,@FSRn</i> (<i>0<n<1</i>)	MOVIW --FSRn <i>DEC FSRn</i> <i>LD A,@FSRn</i> (<i>0<n<1</i>)	MOVIW FSRn++ <i>LD A,@FSRn</i> <i>INC FSRn</i> (<i>0<n<1</i>)	MOVIW FSRn-- <i>LD A,@FSRn</i> <i>DEC FSRn</i> (<i>0<n<1</i>)	MOVIW k6[FSRn] <i>ADD FSR,#k6</i> <i>LD A,@FSRn</i> (<i>0<n<1</i>) (<i>-31<k<31</i>)					
MOVWI ++FSRn <i>INC FSRn</i> <i>ST @FSRn,A</i> (<i>0<n<1</i>)	MOVWI --FSRn <i>DEC FSRn</i> <i>ST @FSRn,A</i> (<i>0<n<1</i>)	MOVWI FSRn++ <i>ST @FSRn,A</i> <i>INC FSRn</i> (<i>0<n<1</i>)	MOVWI FSRn-- <i>ST @FSRn,A</i> <i>DEC FSRn</i> (<i>0<n<1</i>)	MOVWI k6[FSRn] <i>ADD FSR,#k6</i> <i>ST @FSRn,A</i> (<i>0<n<1</i>) (<i>-31<k<31</i>)					

k8 = 0 - 255
d = 0 (w) или 1 (f)
(по умолчанию d=1)

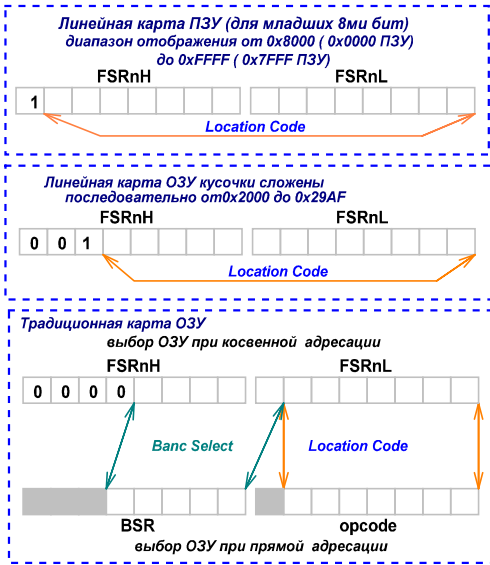
14bit
0<f<0x7F (127)
-256<adr<255
0<addr<2047
0<bit<7
5<r<7
0<n<1

rstep=пропуск команды

обращение к INDF в качестве f(Rn) выполняет обращение к регистру, адрес которого находится в соответствующем регистре FSR ... INDF,d = ... @FSR,d

запись в PCL вызывает автоматическую загрузку PCLATH -> PCH

при GOTO или CALL загружается только адрес страницы



Регистр W представлен в РСФ как РОН области core registers !

Rn=F=0-127 - адрес регистра в текущем банке (стандартная прямая адресация)

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 !!!

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

ПРИ ВЫЧИТАНИИ:	
Z=0, C=1	результат положительный
C=Z=1	результат равен 0
C=Z=0	результат отрицательный

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

MOVf и MOVWf модифицируют Z согласно содержимого регистра назначения !!!

rstep = переход на +1 команду (AVR)

STATUS				
-	-	-	Z	AC
-	-	-	DC	C

- на флаги не воздеиствует
- расширенная система команд
- 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)

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