

Autore : Demaldè Gastone

MPASM - Introduzione	1
Direttive assembler	1
END End Program Block.....	1
EQU Define an Assembler Constant.....	1
INCLUDE Include Additional Source File	2
LIST Listing Options	2
ORG Set Program Origin	2
Istruzioni assembler	3
BYTE-ORIENTED FILE REGISTER OPERATIONS	3
MOVF Move f.....	3
MOVWF Move W to f.....	3
NOP No Operation	3
BIT-ORIENTED FILE REGISTER OPERATIONS	4
BCF Bit Clear f	4
BSF Bit Set f.....	4
BTFSC Bit Test, Skip if Clear	4
BTFSS Bit Test f, Skip if Set	4
LITERAL AND CONTROL OPERATIONS	5
CALL Call Subroutine	5
GOTO Unconditional Branch	5
RETFIE Return from Interrupt.....	5
RETURN Return from Subroutine	5

MPASM - Introduzione

Direttive assembler

Sono comandi che si danno all'assemblatore, essi sono posti all'interno del codice assembler sorgente.

END End Program Block
Indica all'assemblatore la fine del programma.

EQU Define an Assembler Constant
Sintassi : <label> equ <expr>

Il valore di <expr> è assegnato a <label>. Esempio :

```
stop equ H'0001'
```

Dopo questa linea, scrivere **H'0001'** oppure **stop** è la stessa cosa.

INCLUDE Include Additional Source File

Il file specificato da tale direttiva viene letto come se fosse codice sorgente. L'effetto è lo stesso di avere l'intero testo del file incluso inserito nella posizione in cui si trova la linea **include**. Il carattere sharp '#' è opzionale.

Nei nostri sorgenti utilizzeremo :

```
include <p16f876.inc>
```

per includere il file che contiene l'associazione label-registro del micro 16F876.

Si possono avere fino a 6 livelli di include inseriti uno dentro l'altro. In totale ci possono essere fino a 256 file inclusi.

LIST Listing Options

Comando situato su di una unica linea. Con tale direttiva è possibile controllare il processo di assemblaggio o il formato del "listing file". Noi utilizzeremo l'opzione $p=<type>$ per definire il tipo di processore utilizzato :

```
list p=16f876
```

Specificando il tipo di processore con la direttiva **list**, l'ambiente MPLAB è in grado di segnalare un errore di compilazione se per caso è stata impostata la modalità di sviluppo (development mode) con un processore diverso.

ORG Set Program Origin

L'indirizzo del codice che segue è posto in una particolare locazione della memoria programma. Se non viene specificata alcuna direttiva **org**, il programma inizia dalla locazione 0.

Non si può utilizzare la direttiva **org** se si sta scrivendo del codice rilocabile.

Esempio :

```
org 0x004
```

Il codice che segue tale linea viene posto all'indirizzo 0x004 della memoria programma (inizio routine gestione interrupt).

Istruzioni assembler

BYTE-ORIENTED FILE REGISTER OPERATIONS

MOVF Move f

```
Syntax: [ label ] MOVF f,d
Operands: 0 ≤ f ≤ 127
          d ∈ [0,1]
Operation: (f) -> destination
Status Affected: Z
```

Il contenuto del registro 'f' è copiato sulla destinazione 'd'. Se 'd'=1 la destinazione è il registro stesso, se 'd'=0 la destinazione è l'accumulatore W.

Copiare un registro su se stesso è utile per settare il flag Z ("zero bit" posto in STATUS<2>) e controllare quindi se il valore del registro è nullo.

Nel file "p16f876.inc" sono definite due costanti W ed F :

```
W EQU H'0000'
F EQU H'0001'
```

Nel programma assembler, le istruzioni sono più leggibili se si utilizzano tali costanti al posto dei numeri 0 e 1.

Esempio : movf PORTA, W

Copia il registro PORTA nell'accumulatore W : PORTA -> W

MOVWF Move W to f

```
Syntax: [ label ] MOVWF f
Operands: 0 ≤ f ≤ 127
Operation: (W) → f
Status Affected: None
```

Copia l'accumulatore W sul registro 'f'.

Esempio : movwf PORTB

Copia l'accumulatore W sul registro PORTB : W -> PORTB

NOP No Operation

Nessuna operazione.

BIT-ORIENTED FILE REGISTER OPERATIONS**BCF** **Bit Clear f**

Syntax: [label] BCF f,b
Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
Operation: $0 \rightarrow f$
Status Affected: None

Azzera il bit 'b' del registro 'f'. Esempio : `bcf PORTA, mot`

BSF **Bit Set f**

Syntax: [label] BSF f,b
Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
Operation: $1 \rightarrow f$
Status Affected: None

Setta il bit 'b' del registro 'f'. Esempio : `bsf PORTB, mot`

BTFSC **Bit Test, Skip if Clear**

Syntax: [label] BTFSC f,b
Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
Operation: skip if $(f) = 0$
Status Affected: None

Se il bit 'b' del registro 'f' è '0', allora l'istruzione successiva è saltata.

Esempio : `btfsc PORTA, stop`

BTFSS **Bit Test f, Skip if Set**

Syntax: [label] BTFSS f,b
Operands: $0 \leq f \leq 127$
 $0 \leq b < 7$
Operation: skip if $(f) = 1$
Status Affected: None

Se il bit 'b' del registro 'f' è '1', allora l'istruzione successiva è saltata.

Esempio : `btfss PORTA, stop`

LITERAL AND CONTROL OPERATIONS

CALL Call Subroutine

Syntax: [label] CALL k
Operands: $0 \leq k \leq 2047$
Operation: $(PC)+1 \rightarrow TOS$,
 $k \rightarrow PC<10:0>$,
 $(PCLATH<4:3>) \rightarrow PC<12:11>$
Status Affected: None

Chiamata ad un sottoprogramma. I tredici bit dell'indirizzo di ritorno (PC+1) sono caricati sullo stack, poi gli undici bit specificati nell'argomento k sono caricati immediatamente nei bit PC<10:0>. I due bit più alti del program counter PC<12:11> sono caricati con PCLATCH<4:3>.

Questa istruzione dura due cicli macchina.

GOTO Unconditional Branch

Salto incondizionato verso un indirizzo posto all'interno di un banco di 2K locazioni di memoria programma. Per poter saltare in altro banchi occorre prima cambiare il contenuto dei bit 3 e 4 del registro PCLATCH.

Esempio :

```
goto inizio
```

Il program counter viene caricato con l'indirizzo associato alla label "inizio". La prossima istruzione eseguita si troverà in tale posizione.

Il vecchio valore del program counter non viene memorizzato.

Questa istruzione dura due cicli macchina.

RETFIE Return from Interrupt

Istruzione posta alla fine della routine di gestione degli interrupt. L'indirizzo memorizzato sulla cima dello stack viene caricato sul program counter e gli interrupt vengono riabilitati con il set automatico del bit GIE (Global Interrupt Enable bit posto in INTCON<7>).

Questa istruzione dura due cicli macchina.

RETURN Return from Subroutine

Syntax: [label] RETURN
Operands: None
Operation: $TOS \rightarrow PC$
Status Affected: None

Fine di un sottoprogramma. L'indirizzo contenuto sulla cima dello stack (TOS top of the stack) viene caricato sul program counter PC.

Questa istruzione dura due cicli macchina.