

Esercitazioni PIC Midrange.

Inserto_0: gli pseudo opcodes di MPASM.

Il Macro Assembler MPASM mette a disposizione numerosi automatismi in aiuto del programmatore, tra cui sono apprezzabili un certo numero di pseudo opcodes che possono semplificare la scrittura e lettura dei sorgenti.

Si tratta essenzialmente di macro predisposte e che agiscono principalmente attorno ai test sui flag dello **STATUS**.

Questi sono effettuabili con le forme:

bt fss STATUS, flag ; salta istruzione seguente se il flag è 1

bt fsc STATUS, flag ; salta istruzione seguente se il flag è 0

Però esse non sono di immediata comprensione, principalmente, ma non solo, per il programmatore che arrivi da altri processori, come i Motorola HC11, gli 8051, i Parallax, ecc.

In questi esistono istruzioni o forme di Assembly che si riferiscono direttamente ai flag dello **STATUS**. **MPASM** supporta alcune di queste forme.

Va ancora precisato che **non si tratta di opcodes, ovvero di istruzioni facenti parte del set del processore, ma di macro con label mnemonico** analogo a quelle delle famiglie di processori citati e che, in fase di compilazione, sono sostituite dal corrispondente opcode PIC.



Queste macro fanno parte degli automatismi del compilatore MPASM e possono non essere presenti in altri compilatori.

Dove servisse utilizzarle, e non fossero disponibili, è possibile definire le label relative come macro, secondo la tabella seguente. Ecco l'elenco dei principali:

opcode	arg	function	expanded
CLRC		Carry = 0	bcf STATUS, C
SETC		Carry = 1	bsf STATUS, C
SKPNC		skip if no Carry (= 0)	bt fsc STATUS, C
SKPC		skip if Carry (= 1)	bt fss STATUS, C
CLRDC		DC = 0	bcf STATUS, DC
SETDC		DC = 1	bsf STATUS, DC
SKPNDC		skip if no DC (= 0)	bt fsc STATUS, DC

SKPDC		skip if DC (= 1)	btfss	STATUS,DC
CLRZ		Z = 0	bcf	STATUS,Z
SETZ		Z = 1	bsf	STATUS,Z
SKPNZ		skip if no Z (= 0)	btfsc	STATUS,Z
SKPZ		skip if Z (= 1)	btfss	STATUS,Z
TSTF	addr	test file	movf	addr.f
MOVFW	addr	move file to w	movf	addr,w
B	addr	branch address	goto	addr
opcode	arg	function	expanded	
BC	addr	branch on Carry set	btfsc goto	STATUS,C addr
BNC	addr	branch if no Carry	btfss goto	STATUS,C addr
BDC	addr	branch on DC set	btfsc goto	STATUS,DC addr
BNDC	addr	branch if no DC	btfss goto	STATUS,DC addr
BZ	addr	branch on Zero set	btfsc goto	STATUS,Z addr
BNZ	addr	branch if no Zero	btfss goto	STATUS,Z addr
ADDCF	addr, dest		btfsc incf	STATUS,C addr,dest
SUBCF	addr, dest		btfsc decf	STATUS,C addr,dest
ADDDCF	addr, dest		btfsc incf	STATUS,DC addr,dest
SUBDCF	addr, dest		btfsc decf	STATUS,DC addr,dest
NEGF	addr, dest		comf incf	addr,f addr,dest
opcode	arg	function	expanded	
LGOTO	addr	goto a pagina diversa	bxof bxof goto	PCLATH3 PCLATH4 addr & 0x7FF

LCALL	addr	call a pagina diversa	bx f PCLATH3 bx f PCLATH4 call addr & 0x7FF
opcode	arg	function	expanded
banksel	label	Commuta banco	1 o 2 istruzioni
pagesel	label	Commuta pagina	1 o 2 istruzioni

banksel e **pagesel** lpossono essere compilate come 1 o due istruzioni a seconda del numero di banchi e pagine di cui dispone lo specifico processore utilizzato.

Le label non sono case sensitive e quindi possono essere scritte in maiuscolo o minuscolo.

Va considerata una certa cura nel loro uso. Ad esempio:

```
btfsc test, 3
CLRC
next instruction
```

è possibile, dato che **CLRC** origina un solo opcode. Quindi viene assemblato:

```
btfsc test, 3
bcf STATUS,C ; CLRC
next instruction
```

Ma:

```
btfsc test, 3
NEGF
next instruction
```

non funziona correttamente, in quanto il **bt**fsc salta l'istruzione successiva, ma lo pseudo opcode **NEGF** origina **DUE** istruzioni... :

```
btfsc test, 3
comf addr,f ; NEGF
incf addr,dest ; NEGF
next instruction
```

Quindi, il salto, se eseguito, non porta a **next**, ma a **incf** .

D'altronde si tratta della stessa precauzione che occorre considerare quando si utilizzano delle macro.

La scelta se usare questi pseudo opcodes oppure no, è una questione di stile personale nella stesura del sorgente: noi preferiamo utilizzarli ampiamente perchè, a nostro parere, semplificano sensibilmente la lettura di quanto scritto.