

# Assembler per PIC

MNEMONICO	PARAMETRI	SIGNIFICATO
bcf	F,B	Impostazione a 0 del bit B del registro F.
bsf	F,B	Impostazione a 1 del bit B del registro F.

Operazione di trasformazione di un bit.

MNEMONICO	PARAMETRI	SIGNIFICATO
movf	F,D	Movimento del valore del registro F a W (D=0) o su se stesso (D=1).
movwf	F	Movimento del valore del registro di lavoro W a F.
movlw	K	Movimento del valore di letterale K al registro di lavoro W.

Operazioni di movimento dei dati.

```

LIST    P=16F870           ; Pic che si vuole utilizzare
PORTA   EQU    05          ; Definizione della Porta A
TRISA   EQU    05          ; Registro di configurazione della Porta A
PORTB   EQU    07          ; Definizione della Porta C
STATUS  EQU    03          ; Definizione del registro di STATO
ORG     0                  ; Vector di Reset

; Programma principale
MAIN
    bsf    STATUS,5        ; Passaggio al banco 1
    movlw  b'0000000'      ; Un modo di impostare a 0 il registro W
    movwf  TRISA           ; Configurazione di tutti i pin della
                          ; Porta A come uscita

    bcf    STATUS,5        ; Passaggio al banco 0
    bsf    PORTA,3         ; Impostazione a 1 del bit 3 della Porta A
    bcf    TRISA,4         ; Impostazione a 0 del bit 4 della Porta A

    bsf    PORTA,7         ; Impostazione a 1 del bit 7 della Porta A
    movf   PORTA,0         ; Movimento del valore della Porta A a W

END
    
```

Programma per verificare le istruzioni viste.

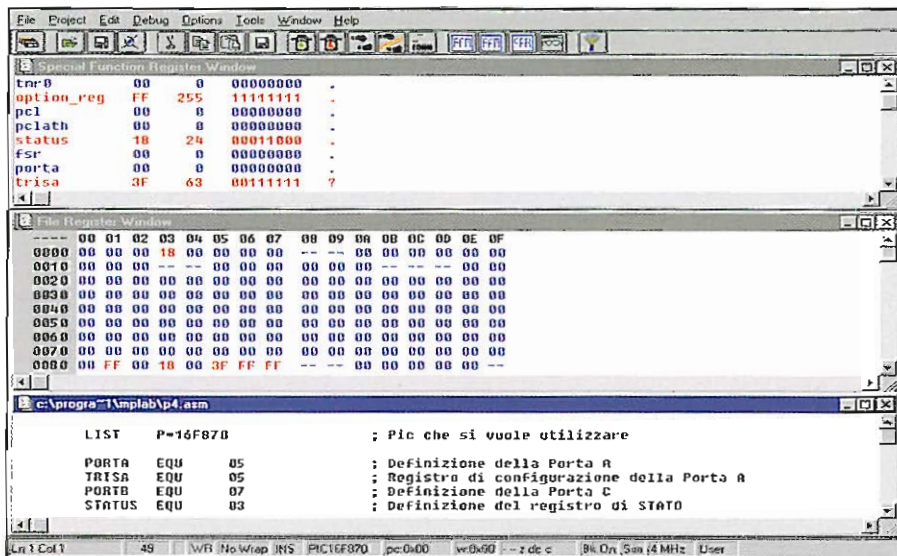
istruzioni per il movimento dei dati, in funzione dell'origine e della destinazione dell'informazione. Come si può vedere, non esiste alcuna istruzione che muova direttamente valori fra due registri, però si può utilizzare il registro di lavoro W come registro intermedio.

## Prova delle espressioni in MPLAB

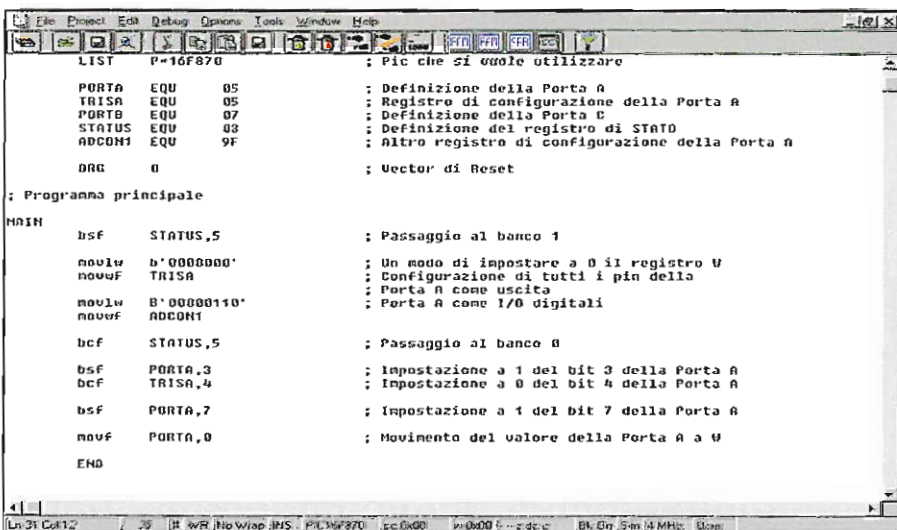
Proviamo a realizzare un esercizio che ci permetta di verificare le istruzioni viste in questo caso e fare qualche prova. Copiate il programma, assegnatelo a un progetto e compilatelo. Prima di iniziare con la simulazione fate attenzione ai commenti. Abbiamo definito la Porta A e i suoi registri di configurazione all'indirizzo 5, però all'indirizzo 7 abbiamo definito un altro registro che abbiamo chiamato PORTB, quando sappiamo che all'indirizzo 7 corrisponde la Porta C. Quindi l'indirizzo definito è di quest'ultima anche se si chiama in un altro modo.

Il registro STATUS occupa l'indirizzo 3 e ci serve per cambiare il banco. Dopo aver definito tutti i pin della Porta A come uscita realizziamo una serie di assegnazioni. Utilizziamo nuovamente il nome di un registro (TRISA) in un modo particolare, come se si trattasse di un altro e come se fossimo sul banco 0, TRISA (5) sta facendo realmente riferimento a PORTA. A questo punto possiamo aprire le finestre per la simulazione, bisogna saper comprendere le informazioni che ci vengono

# Assembler per PIC



Finestre utilizzate per la simulazione.



Nuovo programma con tre linee aggiunte.

fornite da queste finestre, altrimenti corriamo il rischio di confonderci. La finestra dei registri specifici ci mostra il valore di tutti i registri fondamentali, così come li chiama il compilatore.

Nella finestra dei registri specifici appaiono sia i registri specifici sia quelli generali che

abbiamo definito. Il valore degli uni e degli altri deve coincidere, però osservate che all'indirizzo 5 appare solo l'ultima definizione (TRISA), e all'indirizzo 7 il nome della porta sbagliato così come abbiamo previsto. Premete ora il tasto funzione F7 come abbiamo fatto in altri casi. Se avete copiato esattamente il

programma vedrete che la Porta A non modifica il suo valore con le istruzioni "bsf-bcf". Tuttavia se si cambia la Porta A, con la B o la C, tutto funziona correttamente. Qui si può vedere l'importanza della simulazione, perché a volte noi ci impegniamo a considerare una soluzione come corretta, però questa non funziona e la simulazione di solito non inganna. Modificate il programma aggiungendo le tre linee che appaiono nella figura in basso. La Porta A differisce dalla B e dalla C per il fatto che 5 delle sue linee possono essere configurate come analogiche, digitali o una miscela di entrambe. Questo si ottiene tramite il registro ADCON1, che per default ha valore 0: questo significa che la Porta A sarà un ingresso analogico, quindi ciò che stiamo cercando di fare nell'esercizio non concorda con questa definizione. Con il valore inserito in ADCON1 stiamo configurando la Porta A come digitale, e a questo punto possiamo lavorare su di essa. Realizziamo una nuova simulazione, e vedremo che ora tutto è corretto, compresa l'inattività della linea 7 della Porta A con l'istruzione "bsf PORTA,7", dato che la Porta A non ha la linea 7. Se cerchiamo di realizzare un montaggio con dei semplici LED sulla Porta A per vedere il programma, nella realtà ci accorgeremo che ci sono degli errori e che il collegamento alla linea RA4 non si accende. Questa linea è un po' particolare e ha bisogno di un piccolo cambio nell'hardware perché si possa comportare come uscita, ma questo lo vedremo nel prossimo capitolo.