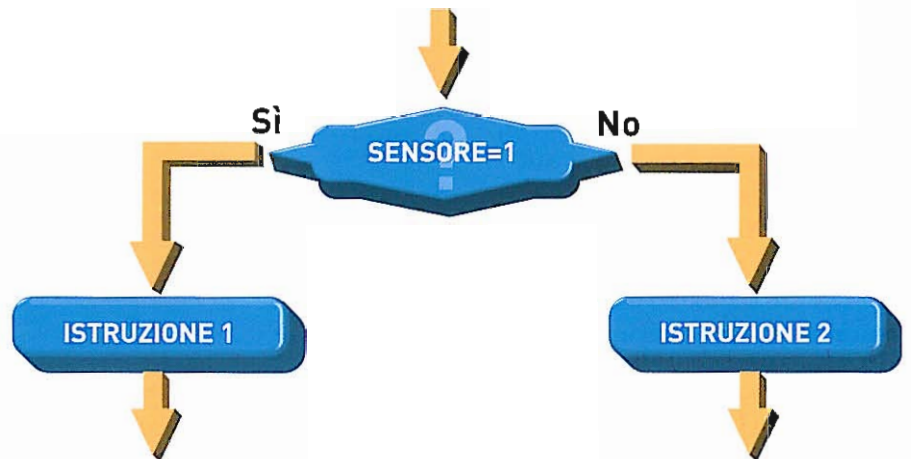


# Assembler per PIC

assembler; "s" significa salto (skip), però un salto speciale, piccolo, di una sola istruzione e la differenza è nella terminazione in "c" che significa zero (clear), o in "s" che significa uno (set). In questo modo, se traduciamo "letteralmente" il mnemonico di queste istruzioni, arriveremo a una frase del genere: "testa il valore di un bit e salta se è 1 (o 0)". Il registro può essere uno qualsiasi della memoria dei dati, sia di quelli specifici che di quelli generali, e il bit sarà un numero fra 0 e 7, corrispondente alla posizione del bit a cui puntare all'interno di questo registro. Il bit 0 è quello meno significativo (partendo da destra) e il bit 7 quello più significativo (si trova a sinistra). In questo caso bisogna anche tener presente che non tutti i registri hanno 8 bit (la porta A, per esempio, ne ha solo 6) e che non tutti i bit dei registri hanno un significato se considerati da soli. Quelli più comuni da verificare sono i flag e le linee delle porte definite come ingressi.

Nella figura della pagina precedente possiamo vedere come avviene la sequenza con l'istruzione di "btfsc". Se il bit verificato ha valore zero si salta un'istruzione, quindi si passa ad eseguire la seconda istruzione in sequenza, però se il bit ha valore 1 la sequenza continua tale e quale, e si esegue l'istruzione successiva e la seguente.

Lo stesso esempio servirà per l'istruzione "btfsc", però il valore del bit da verificare sarà 1. Sinora vi abbiamo indicato solo un piccolo passo per cambiare la sequenza del programma; se lo avete capito, possiamo già iniziare a formare delle strutture.



Organigramma di una struttura IF-THEN.

```
File Project Edit Debug PICSTART Plus Options Tools Window Help
LIST P = 16F873
RADIX HEX
INCLUDE "P16F873.INC"
UAR EQU 21
ORG 0
goto INIZIO ; Indirizzo di Reset
INIZIO btfsc UAR,0 ; Verifica il valore del bit 0 di UAR
goto LABEL1 ; Istruzione se vale 0
goto LABEL2 ; Istruzione se vale 1
LABEL1 goto LABEL1 ; Ciclo se vale 0
LABEL2 goto LABEL2 ; Ciclo se vale 1
END
Ln 24 Col 1 25 |WR|NoWrep|INS|PIC16F873 |pc.0x00 |w.0x00 |--z d c c |Bk On|Sim|4MHz|User
```

Esempio di traduzione dell'organigramma in codice assembler.

## Formazione delle strutture

La prossima cosa da fare sarà pensare alle strutture che vogliamo realizzare. Per fare questo la cosa migliore è partire dall'organigramma. La figura rappresenta un'istruzione che abbiamo utilizzato in molti casi.

Si tratta di una IF-THEN, in cui, a seconda del valore di una variabile, si segue un percorso oppure un altro. La traduzione in codice assembler è immediata. In questo caso la domanda all'interno del rombo si trasforma in un'istruzione "btfsc" e i due rami che ne derivano sono istruzioni "goto".



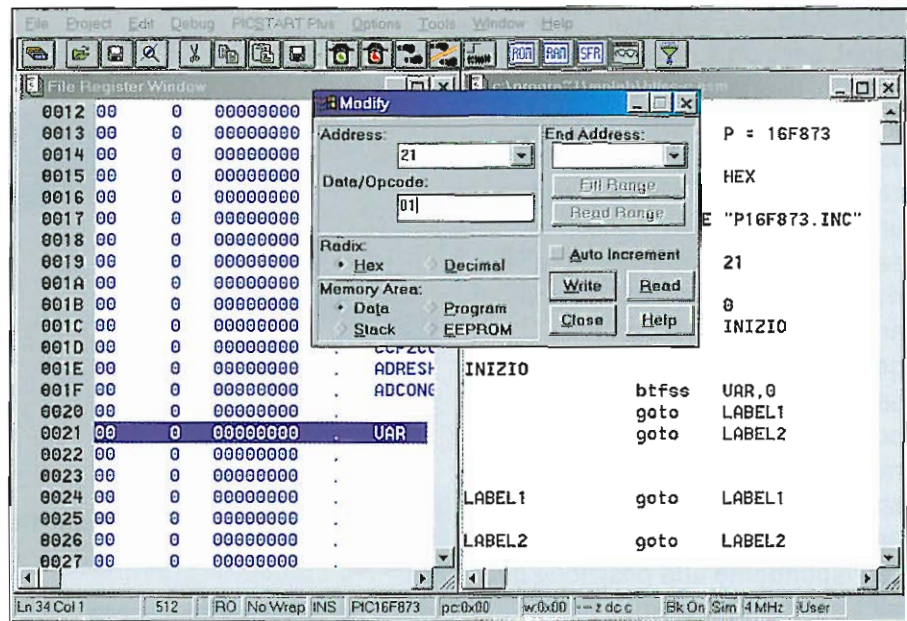
Nella figura è riportato un esempio di questa traduzione. È stato definito un registro in una posizione libera della memoria, ed è stato eseguito un test sul valore del suo bit 0. A seconda di questo l'istruzione salta o meno una posizione, quindi utilizzando delle istruzioni "goto" si possono dividere due programmi indipendenti.

In questi programmi sono stati implementati cicli infiniti. Per verificare l'esempio dobbiamo solo aprire la finestra dei registri generali e posizionarla in modo che veda il registro "21", quello che abbiamo definito. Secondo il valore dei bit 0 di questo registro, la sequenza del programma, quando lo si esegue passo a passo, sarà una oppure l'altra. Per cambiare il valore di questo bit utilizzate Window→Modify.

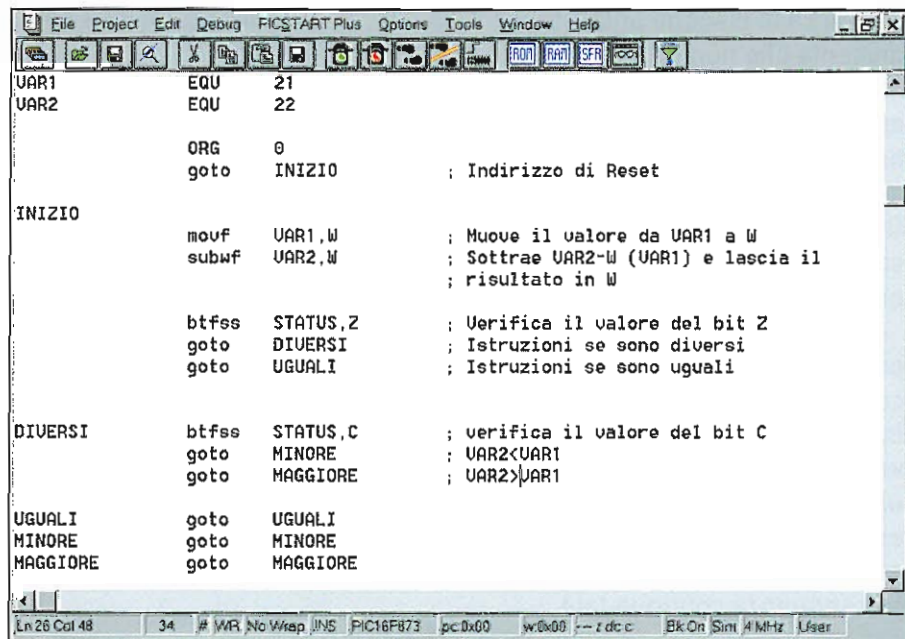
## Registri e bit da utilizzare

Come vedete la cosa complicata non è utilizzare queste istruzioni, ma sapere quali istruzioni e quali bit utilizzare nei vari casi. Vediamo qualche consiglio. Se, come nell'esempio, la domanda dell'organigramma fa riferimento a un sensore, normalmente non dovremmo avere problemi nell'identificarlo, dato che il registro sarà una porta di ingresso/uscita e il bit la linea dove si collega il sensore, sempre di ingresso.

Se stiamo lavorando con dei dispositivi interni, abbiamo



Verifica del funzionamento del programma.



Esempio di domanda per dati completi.

già visto che normalmente si verifica il flag associato.

Anche qui si tratta di un bit all'interno di un registro determinato, ed entrambi

sono facilmente identificabili. Dobbiamo solo avere l'accortezza di lavorare sui banchi corrispondenti a questi registri.

