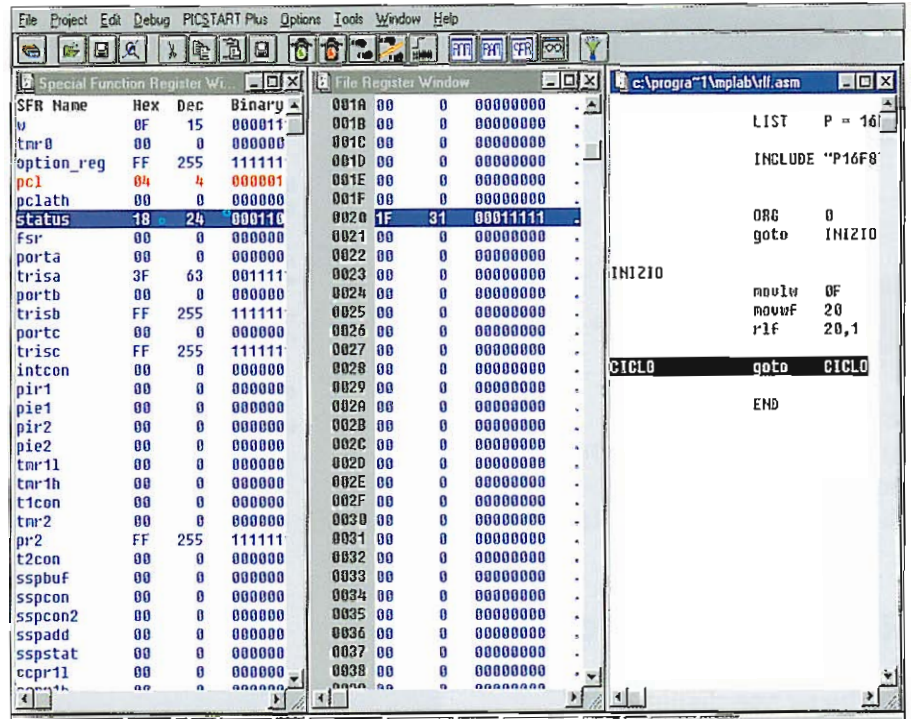


Assembler per PIC

spostare uno dei bit in una determinata posizione del registro (normalmente la più o la meno significativa); per realizzare qualche operazione, e poiché la rotazione è completa, il valore iniziale del carry non ha importanza perché lo si perderà comunque. Per conservare inalterato il valore del registro, non sarà possibile variare il valore del carry fra un'istruzione di rotazione e l'altra. Infine tenete sempre presente che una rotazione completa di un registro, in presenza di carry, comporta nove rotazioni invece di otto, come si potrebbe supporre.

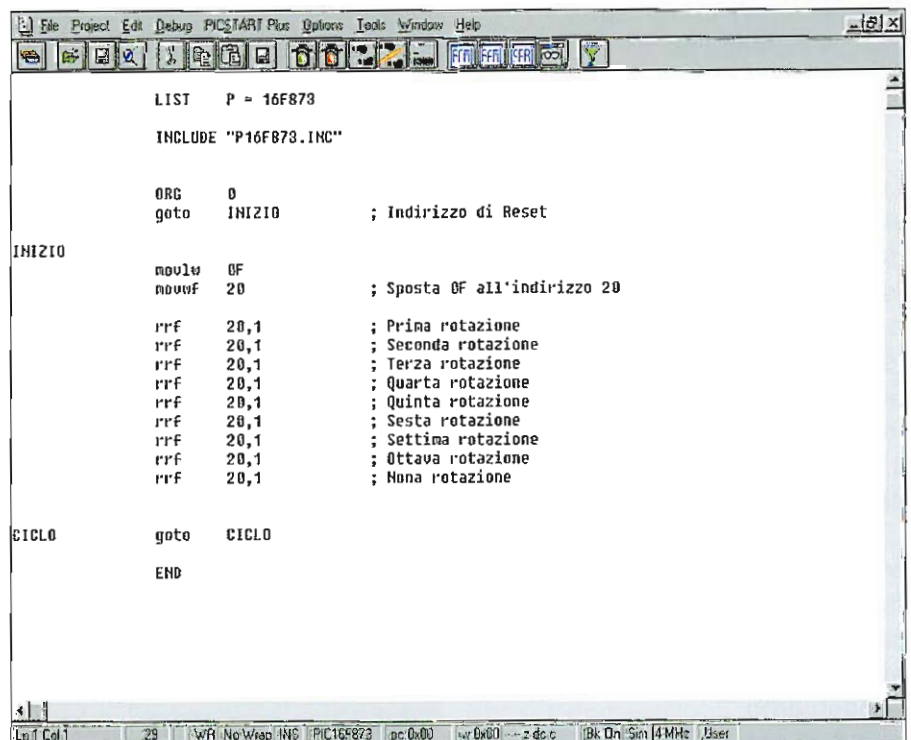


Istruzioni complementari

Dato che l'istruzione "rlf" ruota a sinistra e la "rrf" a destra, si possono considerare istruzioni complementari: se eseguiamo una di esse, e subito dopo eseguiamo l'altra, torneremo alla situazione iniziale. Questo avviene sempre, per qualsiasi valore iniziale del carry e qualsiasi ordine di esecuzione di entrambe le istruzioni. Anche il valore del carry rimane quello iniziale. Tuttavia se fra queste due istruzioni se ne inserisce qualcun'altra, quanto appena detto potrebbe anche non verificarsi.

Questo perché il carry, così come gli altri flag del registro Status, può subire modifiche dopo l'esecuzione di determinate istruzioni. Nella tabella riassuntiva delle

Risultato del programma quando il carry vale 1.



In una rotazione completa non ha importanza il valore del carry.



Organigramma di rotazione senza carry a destra.

Istruzioni presentata in uno dei capitoli precedenti, potrete verificare di quali istruzioni si tratta.

Rotazione senza carry

Anche se non disponiamo di alcuna istruzione di rotazione senza l'utilizzo del carry, a volte potrebbe essere utile. Questo sistema può sembrare complicato, perché richiede un po' di codici in più, però la sua realizzazione è semplice.

Così come mostra l'organigramma, si tratta di verificare il valore del bit più significativo (se la rotazione è a sinistra) o di quello meno significativo (se la rotazione è a destra). Secondo il valore di questi bit si imposterà il carry con valore 1 oppure 0 e poi si eseguirà la rotazione. L'organigramma si può trasformare in codice facilmente. Notate che in questo caso non abbiamo dato al registro un valore iniziale, supponiamo quindi di non conoscerlo. Per realizzare qualche prova, utilizzate l'opzione Window → Modify.

```

LIST    P = 16F873
INCLUDE "P16F873.INC"

ORG    0
goto   INIZIO    ; Indirizzo di Reset

INIZIO

movlw  0F
movwf  20        ; Sposta 0F all'indirizzo 20

rlf    20,1     ; Rotazione a sinistra del registro 20
rrf    20,1     ; Rotazione a destra del registro 20

CICLO  goto    CICLO
END
  
```

Due rotazioni complementari consecutive non modificano lo stato iniziale.

```

LIST    P = 16F873
INCLUDE "P16F873.INC"

ORG    0
goto   INIZIO    ; Indirizzo di Reset

INIZIO

btfsc  20,0     ; Test sul valore del bit 0
goto   UNO      ; Se vale 1
goto   ZERO     ; Se vale zero

UNO    bsf      03,0   ; Imposta il carry a 1
goto   ROT

ZERO   bcf      03,0   ; Imposta il carry a 0
goto   ROT

ROT    rrf      20,1   ; Rotazione a destra

CICLO  goto    CICLO
END
  
```

Programma di rotazione senza carry verso destra.