

# Assembler per PIC

rimanga come in figura.

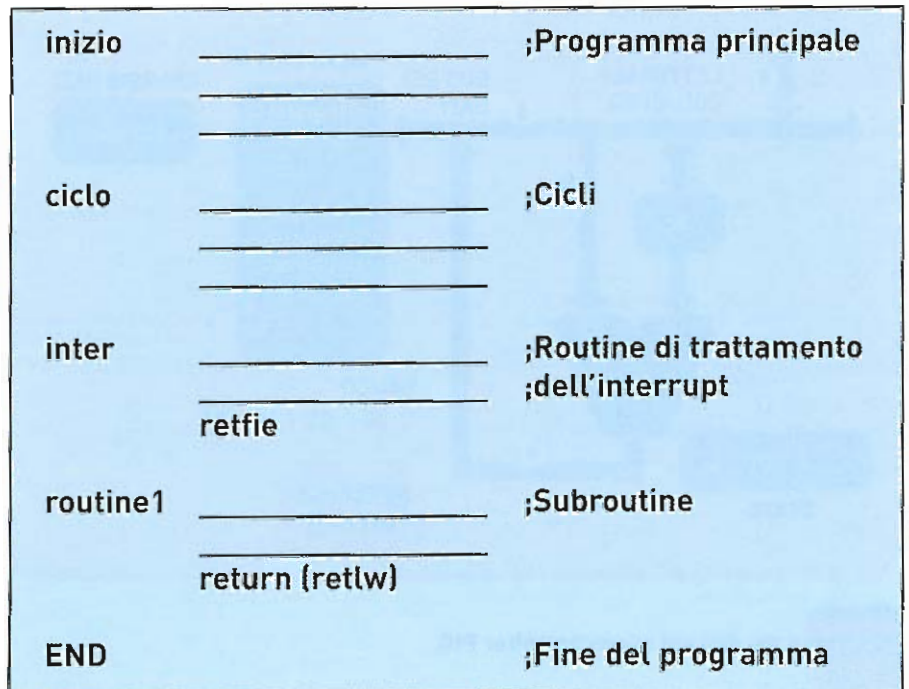
Reinizializzando un sistema con un PIC al suo interno, il PC (Program Counter) che punta in ogni momento l'istruzione da eseguire, punterà l'indirizzo 0 della memoria delle istruzioni.

Quindi è in quell'indirizzo che si deve trovare la prima istruzione e per questo si utilizza la prima direttiva `ORG`. Quando invece avviene un interrupt — abbiamo già visto che cos'è — il PC va all'indirizzo 4 per eseguire un altro frammento di codice. A questo indirizzo dobbiamo porre la prima istruzione della routine dedicata all'interrupt e, nel programma principale, si dovrà "saltare all'inizio di essa".

## Programma principale e routine

I tre paragrafi successivi possono scambiarsi le posizioni nel caso in cui si programmi in modo indipendente, quando cioè non si verifica la condizione in cui si termini da una parte e si prosegua dall'altra, ma si eseguono solo chiamate dirette. La figura rappresenta questo concetto con uno schema.

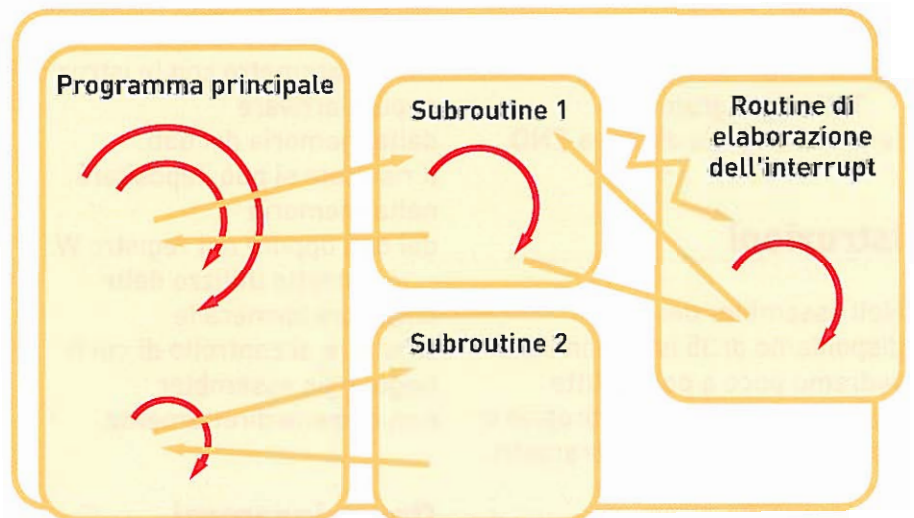
Al programma principale si arriva tramite l'indirizzo 0 della memoria di programma, e sarà formato da un ciclo con successivi cicli interni e terminerà in un altro ciclo in modo che non sia possibile passare da un ciclo all'altro. Alle subroutines si può giungere solamente tramite chiamate esplicite, o dal programma principale o dalle routines di elaborazione dell'interrupt, e si ritornerà al punto



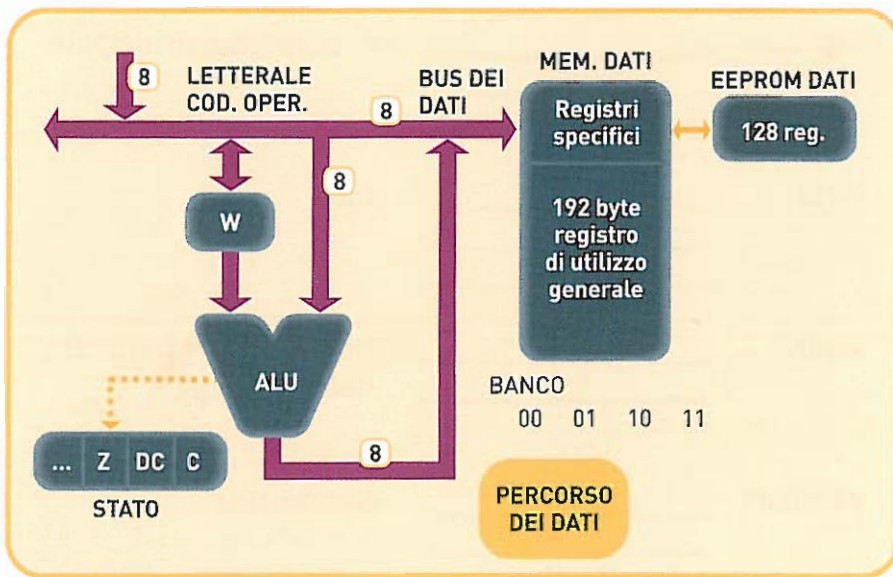
Questi paragrafi possono avere le loro posizioni intercambiabili.

di partenza dopo che la routine sarà terminata. L'istruzione di ritorno può essere "return" o "retlw", secondo il tipo di subroutine. Esiste solamente una routine di

trattamento per l'interrupt, e non esiste nel programma principale alcuna istruzione che richiami questa routine; dato che si tratta di una interruzione, interromperà il programma



Schema di funzionamento di un programma.



**Percorso dei dati nei microcontroller PIC.**

in qualunque momento. Al termine si tornerà al punto da cui è stato interrotto il programma. Questo si può verificare sia che ci si trovi nel programma principale oppure in una subroutine. L'istruzione di ritorno dall'interrupt è "retfie". Sia le subroutines normali che quelle dedicate all'interrupt, possono contenere cicli al loro interno.

Tutto il programma terminerà con la direttiva END.

## Istruzioni

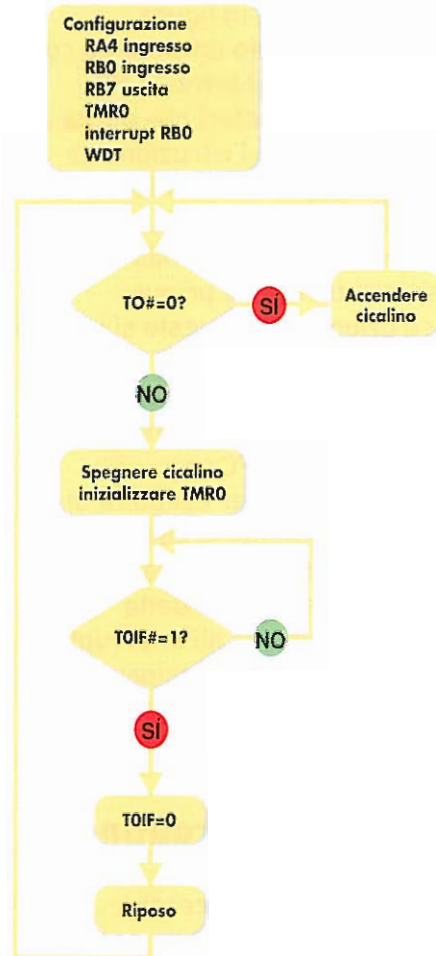
Nell'assembler dei PIC disponiamo di 35 istruzioni, che vedremo poco a poco. Tutte queste hanno un nome proprio o mnemonico e 0, 1 o 2 parametri. Come potrete vedere, le istruzioni sono molto semplici da comprendere e da utilizzare, però proprio per questo, al contrario di ciò che succede con

i linguaggi di alto livello, dovremo conoscere la struttura interna dei PIC. La ALU (Unità Logico Aritmetica) è quella incaricata di organizzare le operazioni aritmetiche logiche. Uno dei suoi dati di ingresso arriverà dal registro di lavoro W, che non appartiene alla memoria dei dati, e l'altro può essere un valore letterale fornito come parametro con le istruzioni oppure arrivare dalla memoria dei dati. Il risultato si può depositare nella memoria dei dati oppure nel registro W.

Il corretto utilizzo delle istruzioni formerà le strutture di controllo di cui il linguaggio assembler non dispone direttamente.

## Organigrammi

Per fare in modo che gli organigrammi possano essere



**Organigramma che precede un programma assembler per PIC.**

successivamente tradotti in modo facile in assembler, devono essere un po' più ristretti degli organigrammi comuni. Avremo un unico punto di ingresso e un unico punto di uscita, avremo inoltre un riquadro di inizio per l'inizializzazione delle variabili e i rombi per la presa delle decisioni, i quali potranno avere unicamente due rami.

Ogni subroutine e la routine dedicata agli interrupt avrà, se il sistema lo richiede, organigrammi indipendenti.