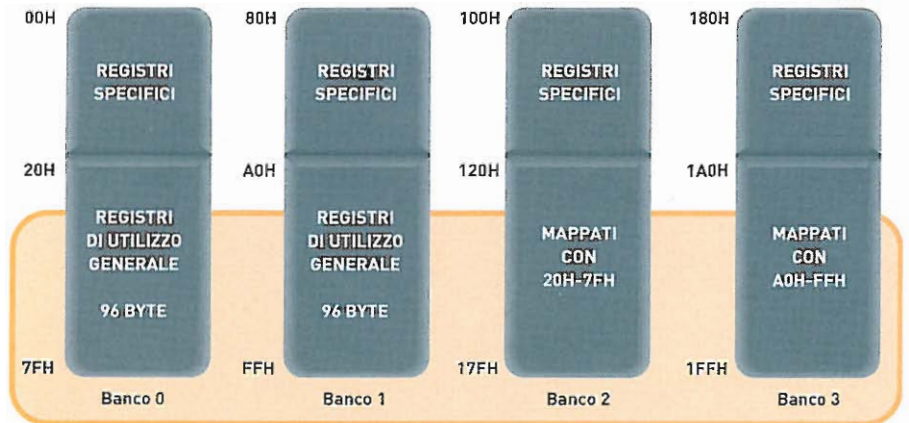


Istruzioni di gestione delle memorie dei dati

Nel Basic avevamo istruzioni specifiche per lavorare con le due memorie dei dati di cui disponiamo nei microcontroller PIC: la RAM e la EEPROM. In assembler invece, non c'è alcuna istruzione dedicata a questi processi, quindi avremo sempre più bisogno di conoscere il funzionamento interno del microcontroller per realizzare queste operazioni con le istruzioni standard.



La memoria RAM dei dati

In precedenza abbiamo già visto la relazione che riguarda la memoria RAM dei dati all'interno del percorso dei dati del microcontroller.

Questa memoria è divisa in zone chiamate "banchi". Secondo il modello di microcontroller avremo a disposizione più o meno banchi e più o meno registri all'interno di ognuno di essi. Nei PIC16F84 abbiamo solo 2 banchi, però nel caso dei PIC16F87x ne abbiamo 4. All'interno di questi banchi esistono registri chiamati specifici, che il microcontroller utilizza per gestire le risorse di cui dispone, e altri generali che l'utente-programmatore può utilizzare come desidera. Dato che il numero di registri specifici varia secondo il numero dei dispositivi del microcontroller, l'indirizzo di inizio dei registri generali cambia in relazione alle risorse stesse e questo va tenuto presente al momento di dichiarare le variabili: anche se il compilatore

Distribuzione della memoria RAM nei PIC16F873.

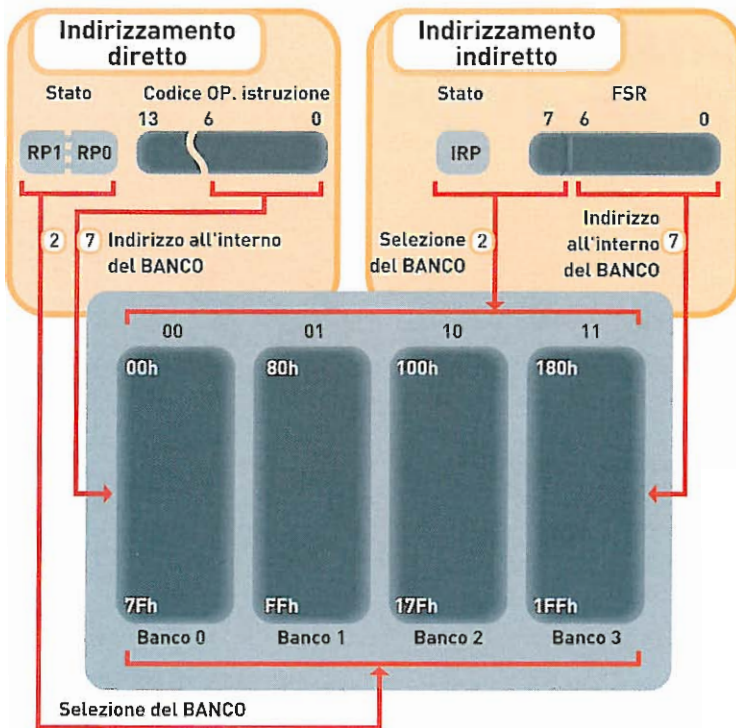
```

LIST      P-16F873          ; Pic che si vuole utilizzare
; Definizione dei registri specifici
PORTA    EQU    05          ; Definizione della Porta A
TRISA    EQU    05          ; Registri di configurazione della Porta A
STATUS   EQU    03          ; Definizione del registro STATUS
PORTB    EQU    26          ; Registro definito male
; Definizione di registri generali
TEMP1    EQU    10          ; Registro definito male
VAR1     EQU    20          ; Primo indirizzo libero da utilizzare
; Definizione di valori costanti
W        EQU    0           ; Destinazione, registro di lavoro W
F        EQU    1           ; Destinazione, registro di origine F
RPO      EQU    5           ; Bit di cambio di banco
RP1      EQU    6           ; Bit di cambio di banco
END
    
```

Dichiarazione delle differenti variabili e registri.

non ci segnalerà errori, infatti, si potrebbero generare dei mal funzionamenti, se durante il corso del programma faremo accesso a degli indirizzi specifici credendo che siano generali. Nei PIC16F84 la prima posizione libera è la 0Ch,

mentre nei PIC16F87x è la 20h. Anche se il programma della figura per ora non fa nulla, ci permette già di vedere che non si produce alcun errore al momento della compilazione. Però dobbiamo tener presente anche ciò che abbiamo



Origine degli indirizzi nei due tipi di indirizzamento.

FSR= 0Ch

**FOR 0Ch TO 27H
INDF= 0
FSR=FSR+1
NEXT**

Esempio di indirizzamento indiretto.

banchi. Tutto questo è conosciuto come indirizzamento diretto, dato che si accede direttamente al registro a cui si fa riferimento.

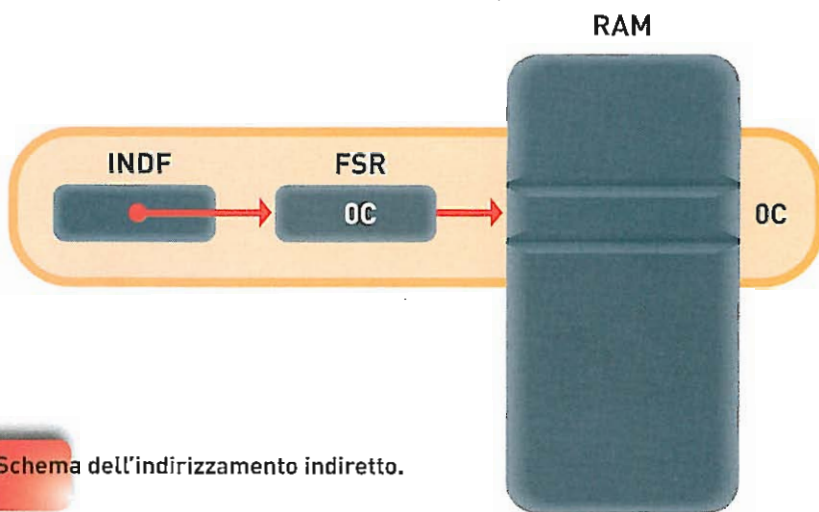
Indirizzamento indiretto della memoria RAM

In assembler si può accedere alla memoria RAM mediante "indirizzamento indiretto". La differenza con il precedente si trova nell'origine dell'indirizzo di accesso, che nell'indirizzamento indiretto si fa tramite i due registri INDF e FSR. INDF si utilizza come una specie di puntatore al registro FSR, in modo che quando si fa riferimento a esso si sta lavorando sulla posizione il cui indirizzo è contenuto all'interno di FSR. L'esempio in pseudo-codice della figura, rappresenta l'impostazione a 0 di una serie di registri consecutivi della memoria. FSR è caricato con l'indirizzo 0Ch, che sarà il primo da eseguire. Viene realizzato un ciclo fra gli indirizzi 0Ch e 27h per fare la stessa operazione su tutti.

Assegnare il valore 0 a INDF significa assegnarlo in realtà all'indirizzo che in quel momento sta puntando tramite SFR.

trattato negli altri capitoli, e cioè che nelle definizioni dei registri, delle costanti e delle variabili possiamo utilizzare i nomi che vogliamo, ma la loro corretta definizione dipenderà dall'utilizzo che ne faremo in seguito. Così ad esempio "TEMP1" è mal definito se realmente lo utilizzeremo come registro

generale, e lo stesso avviene con PORTB utilizzato per la porta B. Conosciamo già anche le istruzioni di gestione dei registri della RAM: le tre di movimento per passare i dati da un registro all'altro utilizzando un registro di lavoro W come intermedio, e le due di impostazione a 1 e a 0 bit per il cambio fra questi



Schema dell'indirizzamento indiretto.