

## Creazione e gestione di tabelle

In precedenza abbiamo visto il concetto e la necessità di lavorare con le tabelle dei dati in diverse occasioni. Il LetPicBasicLite permette la creazione di una tabella di dati tramite programma. Il LetPicBasicPlus aggiunge inoltre la gestione della tabella nella memoria di programma. Anche l'assembler ha istruzioni che permettono la creazione e la gestione delle tabelle dei dati.

L'obiettivo finale è lo stesso, anche se il programmatore dovrà essere cosciente di alcuni aspetti per non incorrere in errori.

### Definizione di una tabella di dati

Per definire le tabelle di dati nell'assembler si utilizza l'istruzione "addwf" già vista e "retlw" che è nuova. Questa istruzione funziona come un ritorno da subroutine, ritornando all'indirizzo indicato in cima allo stack, però con la particolarità di lasciare in W il valore che utilizza come parametro. Questa istruzione non si trova da sola, ma fa parte di una struttura come quella mostrata nella figura. La forma della struttura è sempre la stessa: un nome o una etichetta, DISPLAY in questo caso, un'istruzione "addwf" con i parametri PCL e 1, e tante istruzioni "retlw" quanti dati ha la tabella. I dati sono i parametri dell'istruzione e possono essere di

MNEMONICO	PARAMETRI	SIGNIFICATO
retlw	K	Restituisce l'indirizzo indicato in cima allo stack, lasciando il valore "K" sul registro di lavoro W.

### Utilizzo delle istruzioni "retlw".

```

INCLUDE "P16F873.INC"
ORG 0
goto INIZIO ; Indirizzo di Reset
INIZIO
goto INIZIO
DISPLAY
addwf PCL,1
retlw h'00'
retlw b'11010011'
retlw d'78'
END
    
```

### Programma esempio di definizione di una tabella.

tipo diverso, sempre mantenendo la dimensione di 8 bit.

Nell'esempio sono riportati un valore esadecimale, uno binario e un altro decimale.

In questo programma si definisce solamente la tabella, ma non si fa nulla con essa, perché non ha etichetta. Il programma rimane in un ciclo infinito senza fare nulla.

### Lavoro con le tabelle di dati

Come già succedeva nel LetPicBasicLite o Plus, non è sufficiente definire una tabella, ma bisogna anche utilizzarla. Per questo si utilizza l'istruzione "call". Questa istruzione, come fa anche "goto", salta all'indirizzo

indicato dal parametro "etichetta" ma lascia in cima allo stack l'indirizzo di ritorno, in modo che lo si possa recuperare in seguito.

Il nome della tabella definita sarà "etichetta". Prima di chiamare la tabella, tuttavia, bisogna caricare su W un valore, così come mostra il programma della figura, dove il valore inserito è 1.

PCL è la parte bassa del PC, il contatore di programma, in modo che una volta arrivati all'etichetta il PC ha come valore l'indirizzo di questa etichetta. Se si somma il valore del PC con il valore precedentemente caricato su W e il risultato si introduce in PCL (secondo parametro 1) si realizzerà un salto a valle dell'etichetta, di valore pari a quanto indicato da W. In altre parole W funziona come indice della tabella.

Bisogna tener presente che questa somma incrementa di 1 il suo valore, dato che il PC si incrementa dopo ogni istruzione per puntare a quella successiva, quindi se in W carichiamo uno zero punteremo alla prima istruzione "retlw", se carichiamo un 1 alla seconda, e così via. L'indice della tabella inizia quindi con il valore 0. Quando si esegue l'istruzione "retlw" corrispondente si ritornerà all'istruzione successiva a quella di chiamata, lasciando su W il valore indicato. Il processo è indicato passo a passo nel grafico della figura riportata nella pagina successiva.

MNEMONICO	PARAMETRI	SIGNIFICATO
call	etichetta	Salta l'indirizzo puntato da "etichetta" scrivendo in cima allo stack l'indirizzo di ritorno.

### Utilizzo dell'istruzione "call".

```

File Project Edit Debug PICSTART Plus Options Tools Window Help
[Icons]
LIST P = 16F873
RADIX HEX
INCLUDE "P16F873.INC"
ORG 0
goto INIZIO ; Indirizzo di Reset
INIZIO
movlw d'01'
call DISPLAY
goto INIZIO
DISPLAY
addwf PCL,1
retlw h'00'
retlw b'11010011'
retlw d'78'
END
Ln 24 Col 1 | 24 | WR | No Wrap | INS | PIC16F873 | pc:0x00 | w:0x00 | --z dc c | Bk On | Sim | 4 MHz | U3
    
```

### Programma esempio di una chiamata a una tabella.

## Precauzioni per il lavoro con le tabelle

Il lavoro con le tabelle è semplice, sempre che si seguano alcune minime precauzioni. La prima da tener presente è che stiamo utilizzando il registro di lavoro W, sia per passare un parametro alla tabella sia per

ottenere un valore di ritorno, e il registro non può essere diverso, quindi se vogliamo prendere diversi valori da una tabella fra una chiamata e un'altra, bisogna spostare il valore di W a un altro registro per non perderlo, o realizzare immediatamente l'operazione che si desidera con esso. Un altro inconveniente