

Capitolo Quarto	2
Le direttive di assemblaggio di ASM 68000	2
Premessa	2
1. Program Location Counter e direttiva ORG	2
2. Dichiarazione di dati: le direttive DS e DC	3
2.1 Direttiva DS	3
2.2 Direttiva DC	4
3. La chiusura statica del programma: direttiva END	5
4. Uso di file esterni per l'assemblaggio: direttiva INCLUDE	5
5. Altre direttive	6
5.1 Direttiva EQU	6
5.2 Direttiva SET	6
5.3 Direttiva REG	6

Capitolo Quarto

Le direttive di assemblaggio di ASM 68000

Premessa

In questo capitolo si descrivono le direttive riconosciute dal linguaggio assembler che la Motorola ha definito per il proprio processore MC68000.

1. Program Location Counter e direttiva ORG

Per definire gli indirizzi (relativi) di memoria nei quali saranno allocati programma e dati, il programma assembler gestisce una variabile interna a 32 bit, detta *Program Location Counter* (PLC), il cui valore è incrementato ogni volta che un'istruzione è tradotta, di un numero pari alla dimensione in byte del codice macchina dell'intera istruzione.

Una apposita pseudo-istruzione, direttiva ORG (Set Origin), è utilizzata per inizializzare o per assegnare un valore a PLC. Il valore è espresso come operando della direttiva, secondo la seguente sintassi:

ORG *indirizzo*

ove *indirizzo* è un numero (costante o espressione).

Ad esempio:

```
ORG    $3000
```

nel significato: la prima istruzione che segue avrà l'indirizzo §3000.

```
ORG    INDIRIZZO_BASE+200
```

nel significato: la prima istruzione che segue avrà l'indirizzo della variabile INDIRIZZO_BASE incrementato di 200.

Il simbolo speciale '*' sta ad indicare il valore del Program Location Counter e può essere utilizzato nelle espressioni. Ad esempio:

```
ORG    *+$1000
```

nel significato: la prima istruzione che segue avrà un indirizzo pari a quello corrente di PLC incrementato di §1000 o, in altri termini, lascia §1000 locazioni di memoria libere.

La direttiva ORG è tipicamente presente all'inizio del codice sorgente, per definire l'indirizzo base a partire dal quale il codice macchina verrà caricato in memoria. Altre direttive ORG possono essere utilizzate all'interno del pro-

gramma (ad esempio per allocare i dati in una zona della memoria separata dall'area in cui è allocato il codice).

2. Dichiarazione di dati: le direttive DS e DC

Per introdurre in un programma una (o più) variabili, dando ad essa un nome ed eventualmente assegnando loro un valore iniziale definito già al tempo dell'assemblaggio si usano le direttive DS (*Define Storage*) e DC e DC (*Define Constant*).

Preliminarmente si ricorda che il 68000 gestisce dati di tipo Byte (.B), Word (.W, tipo di default) e Longword (.L) e si pone in evidenza un vincolo della sua architettura: gli indirizzi dei dati di tipo W e L debbono essere pari (l'indirizzo è calcolato in byte). In altri termini, un dato che sia più lungo di un byte deve iniziare ad un indirizzo pari. Le direttive in esame realizzano automaticamente tale vincolo "allineando" l'indirizzo mediante eventuale incremento di 1 unità a PLC.

2.1 Direttiva DS

La direttiva DS (*Define Storage*) definisce il nome di una variabile, le assegna una locazione ed uno spazio di memoria. Ad esempio, la dichiarazione

```
ALFA DS.B 1
```

introduce una variabile di nome ALFA, di tipo byte, posta in memoria all'indirizzo PLC. A tale scopo ASM compie le seguenti azioni:

- 1) immettere in tabella simboli la coppia (ALFA, valore di PLC);
- 2) incrementa PLC di 1 (dimensione in byte del dato immeso)

Più in generale, la dichiarazione si presenta nella forma:

```
[etichetta] DS[[.L] | [.W] | [.B]] espressione
```

ed ASM preliminarmente allinea PLC ad un indirizzo pari (in caso di W o L) e quindi incrementa PLC di una quantità pari al valore specificato nel campo operando (come costante o espressione numerica) moltiplicata per 1, 2 o 4, a seconda che si tratti di dato di tipo B, W o L.

Si noti che di norma l'etichetta dell'istruzione è non-vuota, anche se sintatticamente potrebbe esserlo. In realtà, un'etichetta vuota viene usata se si vuole usare la dichiarazione soltanto per allineare PLC ad indirizzo pari:

```
DS.W 0
```

2.2 Direttiva DC

La direttiva DC ha un effetto analogo a DS, ma, contrariamente a quest'ultima, inizializza l'area di memoria riservata con la dichiarazione. In altri termini, essa è usata per allocare in memoria sequenze di valori costanti o stringhe di caratteri. La sintassi della direttiva è:

[etichetta] DC[[.L] | [.W] | [.B]] <elemento>
[[,<elemento>]...,<elemento]]

ove il singolo <elemento> può essere un'espressione o una stringa di caratteri. Si noti che mentre con la direttiva DS la dimensione dell'area riservata è definita dal numero presente nel campo-operandi, con la DC è il numero degli elementi della lista che lo definisce.

Come esempio, si consideri il seguente frammento di codice assembly:

```
1:          ORG   $1000
2:          DC.W $1,$2,$3
3:          DC.L $FF
4:  ST1     DC.B 'Hello'
5:  ST2     DC.B 'world',0
6:          DC.B $4
7:          DC.B $5
8:          DC.W $6
```

ASM produce un codice oggetto che, quando sarà caricato in memoria, darà luogo alla situazione illustrata nella figura seguente:

0000 1000	00	01	00	02
0000 1004	00	03	00	00
0000 1008	00	FF	48	65
0000 100C	6C	6C	6F	77
0000 1010	6F	72	6C	64
0000 1014	00	04	05	00
0000 1018	00	06	00	00

Osservando la figura precedente, è possibile fare le seguenti considerazioni:

- § le costanti numeriche sono rappresentate su 1, 2 o 4 byte, a seconda del suffisso usato nella direttiva DC;
- § le stringhe di caratteri sono rappresentate come sequenze di byte; ciascun byte ha il valore del codice ASCII corrispondente ad un carattere (linee 4 e 5 del programma);
- § per ottenere stringhe terminate dal carattere con codice ASCII 0, come quelle utilizzate dalle librerie C standard, occorre allocare esplicitamente la costante 0 nel byte successivo a quello in cui è memorizzato l'ultimo carattere della stringa (linea 5);
- § se si usa una direttiva DC.W o DC.L per allocare una stringa, verranno eventualmente allocati degli zeri in coda alla stringa, in modo da rendere la dimensione allocata un multiplo di 2 o di 4 byte rispettivamente;
- § la direttiva DC.B alloca un valore (o una sequenza di valori) a partire dall'indirizzo corrente del Program Location Counter, anche se quest'ultimo ha un valore dispari (linee 6 e 7). Di contro, le direttive DC.W e DC.L, nel caso in cui il PLC abbia un valore dispari, ne incrementano il valore di 1 (linea 8)

3. La chiusura statica del programma: direttiva END

La direttiva END è utilizzata per contrassegnare la fine del sorgente assembly. ASM ignora tutto ciò che si trova nel file sorgente dopo la linea contenente questa direttiva. In ASM68K, alla direttiva END deve essere associato come operando l'*entry point* del programma (ovvero l'indirizzo della prima istruzione da eseguire).

4. Uso di file esterni per l'assemblaggio: direttiva INCLUDE

Così come avviene in linguaggi ad alto livello, una buona organizzazione del lavoro di programmazione ed assemblaggio consiste nel riusare in programmi diversi i medesimi file già usati altrove. A tale scopo si usa la direttiva

INCLUDE File_Esterno

Essa fa sì che ASM prosegua il processo di assemblaggio a partire dalla prima linea del codice sorgente di File_Esterno (espresso mediante un path). Quando l'assemblatore raggiunge la fine del file esterno, continua il processo di

assemblaggio del file sorgente originario, a partire dalla linea successiva alla direttiva INCLUDE.

5. Altre direttive

5.1 Direttiva EQU

La direttiva EQU (*Equate Symbol*) è utilizzata per definire un simbolo che sostituisca per ASM un valore.

etichetta **EQU** *espressione*

La stringa utilizzata come etichetta è il nome a cui l'assemblatore associa il valore specificato come operando della direttiva. Ad esempio:

```
                  ORG    $1000
ENTRY1           EQU    *
```

La direttiva EQU assegna al simbolo ENTRY1 il valore esadecimale \$1000.

Si noti che:

- § il valore del simbolo introdotto con la direttiva EQU non può essere alterato durante l'assemblaggio;
- § l'espressione operando di EQU non può contenere riferimenti in avanti (cioè simboli definiti in linee di programma seguenti).

5.2 Direttiva SET

La direttiva SET (*Set Symbol*) è simile ad EQU, ma consente di modificare il valore del simbolo mediante una ulteriore direttiva SET:

etichetta **SET** *espressione*

5.3 Direttiva REG

La direttiva REG (*Register Range*) associa ad un simbolo una lista di registri. Tale simbolo potrà essere utilizzato nel resto del programma sorgente come operando per l'istruzione MOVEM, per operazioni di load e store multiple. L'operando di questa direttiva consiste in una lista di nomi di registri (Registri Dato o Registri Indirizzo) separati dal carattere '/'. Ciascun elemento della lista può essere o un singolo registro, o una sequenza di registri individuata mediante il primo e l'ultimo registro separati dal carattere '-'. Registri e sequenze di registri possono essere specificati secondo un ordine qualunque. Un esempio d'uso della direttiva è il seguente:

```
my_regs          REG    D0-D3/D7/A1-A3
```