
Assemblaggio per il processore Motorola 68000



Daniele Paolo Scarpazza
daniele.scarpazza@elet.polimi.it

Politecnico di Milano
Ultimo aggiornamento: 16 Maggio 2005

Assemblaggio

- A prima vista sembra un problema banale:
“una riga di assembly = una istruzione”
- in realtà: piccoli problemi, fra cui *riferimenti in avanti*,
esempio (frammento di codice che calcola il valore assoluto):

```
    ...  
    CMP.L   #0, D0      * test whether D0 >= 0  
    BGE     LABEL_POS  * jump to LABEL_POS if true  
    NEG.L   D0          * else invert D0  
LABEL_POS  ...
```

per assemblare una istruzione di salto in avanti
devo già conoscere a quale indirizzo corrisponde l'etichetta.

- Soluzioni possibili (con vantaggi e svantaggi):
 - assemblaggio in due passate (l'input viene letto 2 volte)
 - caricamento di tutto il listato in memoria in una forma intermedia,
elaborazione della forma intermedia;

Primo passo

- Costruzione della tabella dei simboli:
 - coppie <nome, valore> di etichette e costanti;
 - l'assemblatore mantiene l'ILC (*instruction location counter*), variabile che contiene l'indirizzo dell'istruzione che sta assemblando in quell'istante;
 - inizializzo ILC a 0 all'inizio dell'assemblaggio;
 - ad ogni direttiva del tipo **ORG \$xxx**, sposto ILC alla locazione **xxx**;
 - ad ogni istruzione letta, incremento ILC della dimensione di quella istruzione;
 - man mano che incontro le etichette, aggiungo nella tabella dei simboli la coppia:
<nome etichetta, valore attuale di ILC>;

Primo passo: tabelle usate

- Tabella dei simboli:
 - potrebbe contenere informazioni aggiuntive (lunghezza del campo dati, informazioni di rilocazione, ...)
- Tabella degli opcode e degli ingombri:
 - indica come tradurre uno mnemonico in uno degli opcode corrispondenti a seconda degli argomenti;
 - confronta: Hamacher et al.,
Introduzione all'architettura dei calcolatori, Appendice A;
 - confronta: Motorola M68000 Family, Programmer's Ref. Manual,
Section 8, Instruction format summary;

Determinazione dell'ingombro

- Ingombri delle istruzioni che manipolano dati
 - Istruzione = 1 parola contenente opcode + 0–4 parole di estensione;
 - La parola è lunga 16 bit;
 - Gli opcode sono brevi e lasciano spazio per codificare:
 - modi di indirizzamento del primo e del secondo operando;
 - se il primo/secondo operando sono registri, il numero di registro;
 - Costanti, spiazamenti e indirizzi si codificano in parole di estensione;
 - Dimensioni possibili:
 - delle costanti: 8, 16 bit (1 parola), 32 bit (2 parole);
 - degli spiazamenti: 8, 16 bit (1 parola);
 - indirizzi di memoria: 16 bit (1 parola) o 32 bit (2 parole);
 - Eccezioni: ospitano l'argomento nella stessa parola dell'opcode:
 - ADDQ, SUBQ prevedono costanti a 3 bit;
 - MOVEQ permette una costante di 8 bit;

Esempi di ingombro

Istruzione	parola opcode	parola arg1	parola arg2	parole totali
MOVEQ.B #\$01, D1	1	0	0	1
MOVE.B #\$12, D1	1	1	0	2
MOVE.W #\$0123, D1	1	1	0	2
MOVE.L #\$01234567, D1	1	2	0	3
MOVE.X A1, D1	1	0	0	1
MOVE.X (A1), D1	1	0	0	1
MOVE.X (A1), D1	1	0	0	1
MOVE.X (A1), D1	1	0	0	1
MOVE.X \$0123 (A1), D1	1	1	0	2
MOVE.X \$01 (A1, D2.Y), D1	1	1	0	2
MOVE.X \$01 (A1, D2.Y), \$23 (A3, A4.Y),	1	1	1	3
MOVE.X \$0123.W, D1	1	1	0	2
MOVE.X \$01234567.L, D1	1	2	0	3
MOVE.X \$0123.W, \$456789AB.L	1	1	2	4
MOVE.W #\$0123, \$456789AB.L	1	1	2	4
MOVE.X \$01234567.L, \$89ABCDEF.L	1	2	2	5
MOVE.L #\$01234567, \$89ABCDEF.L	1	2	2	5
ADD.X D1, D2	1	0	0	1
ADD.X \$0123.W, D2	1	1	0	2
ADDX. \$01234567.L, D2	1	1	1	3
LSL.X #1, D1	1	0	0	1

Determinazione dell'ingombro (salti)

- Ingombri delle istruzioni di salto:
 - Istruzione = 1 parola contenente opcode + 0–2 parole di estensione;
 - La parola è lunga 16 bit;
 - Gli opcode sono brevi e lasciano spazio per codificare:
 - modo di indirizzamento della destinazione del salto;
 - se la destinazione è in un registro, il numero di registro;
 - la condizione di salto;
 - uno spiazzamento corto da 8 bit per i salti di tipo Bcc (non modalità registro base; i salti di tipo J);
 - Indirizzi e altri tipi di spiazzamenti vanno in parole di estensione;
 - Dimensioni possibili:
 - degli spiazzamenti: 8 bit + registro base (1 parola), 16 bit (1 parola);
 - indirizzi di memoria: 16 bit (1 parola) o 32 bit (2 parole);

Esempi di ingombro (salti)

Istruzione	parola opcode	parole dest	parole totali
JMP (A1)	1	0	1
BRA \$12 (PC)	1	0	1
Bcc \$12 (PC)	1	0	1
BRA \$1234 (PC)	1	1	2
Bcc \$1234 (PC)	1	1	2
JMP \$12 (PC, A1)	1	1	2
JMP \$1234	1	1	2
JMP \$12345678	1	2	3

Secondo passo

- Generazione e emissione del codice oggetto:
 - opcode e lunghezza di ogni istruzione sono già noti
 - completamento includendo i valori delle etichette
 - emissione del codice oggetto
- È a disposizione un esempio completo di traduzione e assemblaggio per un programma C costituito da funzioni multiple.