

Report sull'Esercizio di Programmazione in Assembly 8088

Introduzione

Ho affrontato un esercizio di programmazione in linguaggio assembly per il microprocessore 8088. L'obiettivo era scrivere un programma che, presi due dati a e b dalla memoria, calcolasse l'espressione $(a + 3) * b$ ponendo il risultato nel registro accumulatore.

Traccia dell'Esercizio

- Scrivere un programma in linguaggio assembly 8088 che, presi due dati a e b in memoria, calcola l'espressione $(a + 3) * b$ ponendo il risultato nel registro accumulatore.

Soluzione Proposta

Ho scritto un programma assembly che segue questi passaggi:

- 1. Caricamento dei Valori dalla Memoria nei Registri:**
 - Carico a nel registro AX.
 - Carico b nel registro CX.
- 2. Calcolo dell'Espressione:**
 - Sommo 3 al valore di AX (che contiene a).
 - Moltiplico il risultato per il valore di CX (che contiene b).
- 3. Memorizzazione del Risultato:**
 - Il risultato della moltiplicazione è memorizzato nel registro AX, che funge da registro accumulatore.
- 4. Terminazione del Programma:**
 - Uso una chiamata di sistema per terminare il programma in ambiente DOS.

Codice Assembly

; Calcolo di $(a + 3) * b$

_EXIT = 1

.SECT .TEXT

start:

; Caricare 'a' nel registro AX

MOV AX, WORD PTR [a] ; AX <- a

; Aggiungere 3 ad AX

ADD AX, 3 ; AX <- AX + 3

; Caricare 'b' nel registro CX

MOV CX, WORD PTR [b] ; CX <- b

; Moltiplicare AX per CX

MUL CX ; AX <- AX * CX

; Salvare il risultato in RESULT

MOV WORD PTR [RESULT], AX ; RESULT <- AX

; Terminare il programma (per un ambiente DOS)

XOR CX, CX ; Resetta CX a 0

MOV AX, 0x4C00

INT 0x21

.SECT .DATA

a: .WORD 0x0005 ; Esempio valore per a

b: .WORD 0x0003 ; Esempio valore per b

RESULT: .WORD 0

Spiegazione Dettagliata del Codice

1. Definizione della Costante:

```
_EXIT = 1
```

- Definisco una costante `_EXIT` con valore 1, utilizzata per terminare il programma.

2. Sezione del Codice Eseguitibile:

```
.SECT .TEXT
```

```
start:
```

- Questa sezione contiene il codice eseguibile. L'etichetta `start:` indica l'inizio del programma.

3. Caricamento e Manipolazione dei Dati:

```
MOV AX, WORD PTR [a] ; AX <- a
```

```
ADD AX, 3 ; AX <- AX + 3
```

```
MOV CX, WORD PTR [b] ; CX <- b
```

```
MUL CX ; AX <- AX * CX
```

```
MOV WORD PTR [RESULT], AX ; RESULT <- AX
```

- **MOV AX, WORD PTR [a]:** Carica il valore di `a` nel registro `AX`.
- **ADD AX, 3:** Aggiunge 3 al valore nel registro `AX`.
- **MOV CX, WORD PTR [b]:** Carica il valore di `b` nel registro `CX`.
- **MUL CX:** Moltiplica il valore di `AX` per il valore di `CX` e mette il risultato in `AX`.
- **MOV WORD PTR [RESULT], AX:** Memorizza il risultato della moltiplicazione nella variabile `RESULT`.

4. Terminazione del Programma:

```
XOR CX, CX ; Resetta CX a 0
```

```
MOV AX, 0x4C00
```

```
INT 0x21
```

- **XOR CX, CX**: Resetta il registro CX a 0.
- **MOV AX, 0x4C00** e **INT 0x21**: Questi comandi sono utilizzati per terminare il programma in un ambiente DOS. 0x4C00 è il codice di uscita per il comando INT 0x21.

5. Sezione dei Dati:

`.SECT .DATA`

a: `.WORD 0x0005` ; Esempio valore per a

b: `.WORD 0x0003` ; Esempio valore per b

`RESULT: .WORD 0`

- Questa sezione definisce i dati utilizzati dal programma. a e b sono definiti con i valori di esempio 0x0005 (5 in decimale) e 0x0003 (3 in decimale) rispettivamente. RESULT è inizializzato a 0 e sarà utilizzato per memorizzare il risultato dell'operazione.

Testing del Codice

Per testare questo codice, è possibile usare un emulatore o un assembler per il microprocessore 8086. Uno degli strumenti popolari per questo scopo è DOSBox insieme a un assembler come MASM o TASM. Di seguito sono riportati i passaggi generali per testare il codice:

1. **Installazione di DOSBox**: Scarica e installa DOSBox dal sito ufficiale.
2. **Installazione di MASM**: Scarica MASM (Microsoft Macro Assembler) e configura DOSBox per utilizzarlo.
3. **Assemblare il Codice**: Scrivi il codice in un file con estensione `.asm` e utilizza MASM per assemblarlo.
4. **Eseguire il Programma**: Utilizza DOSBox per eseguire il file `.exe` generato.

Ecco un esempio di comandi che potresti utilizzare in DOSBox:

`masm codice.asm`

`link codice.obj`

`codice.exe`

Questo eseguirà il programma e permetterà di verificare che il calcolo è corretto.

Conclusione

Questo esercizio mi ha permesso di approfondire la programmazione in assembly per il microprocessore 8088/8086, imparando a caricare valori in registri, eseguire operazioni aritmetiche e memorizzare i risultati. Ho anche esplorato i metodi per testare e verificare il codice assembly utilizzando strumenti come DOSBox e MASM.