



Linguaggio macchina



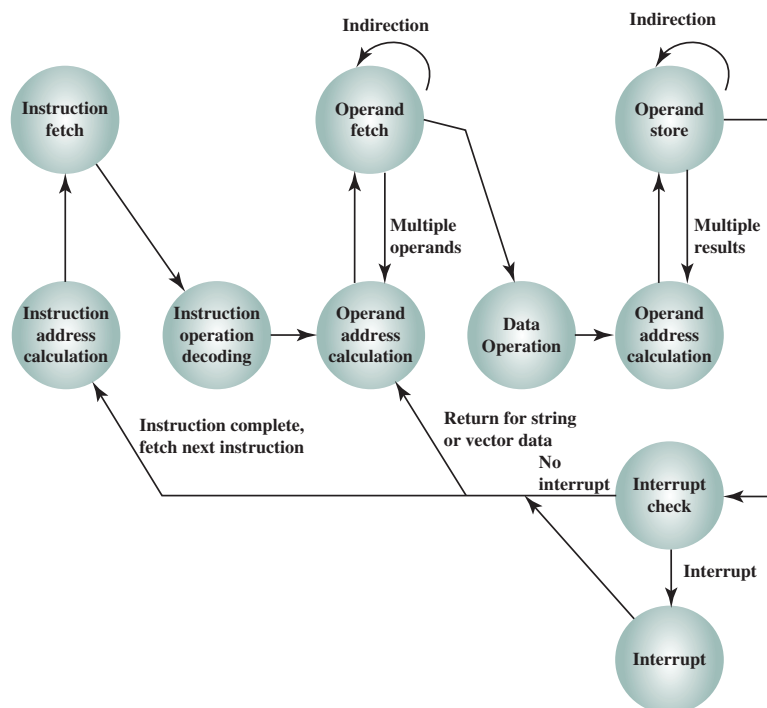
Caratteristiche e funzioni del set di istruzioni

- Linguaggio macchina
- Tipi di operandi
- Tipi di operazioni

Linguaggio macchina

- Insieme delle istruzioni (instruction set) che la CPU può eseguire

Ciclo della CPU



Elementi di un'istruzione macchina

- Codice operativo
 - Specifica l'operazione da eseguire
- Riferimento all'operando sorgente
 - Specifica l'operando che rappresenta l'input dell'operazione
- Riferimento all'operando risultato
 - Dove mettere il risultato
- Riferimento all'istruzione successiva

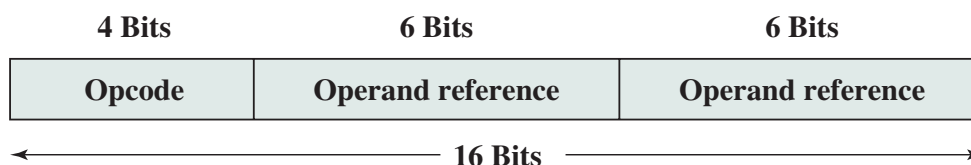
Dove sono gli operandi?

- Memoria centrale (o virtuale)
 - Si deve fornire l'indirizzo
- Registri della CPU
 - Ognuno ha un numero che lo identifica
- Dato immediato nella istruzione
- Dispositivi di I/O
 - Numero modulo o indirizzo di M

Rappresentazione delle istruzioni

- Istruzione = sequenza di bit
 - Divisa in campi
- Spesso viene usata una rappresentazione simbolica delle configurazioni di bit
 - es.: ADD, SUB, LOAD
- Anche gli operandi hanno una rappresentazione simbolica
 - Es.: ADD A,B

Esempio del formato di un'istruzione



Tipi di istruzioni

- Elaborazione dati
 - Istruzione aritmetiche e logiche, di solito sui registri della CPU
- Immagazzinamento dei dati in M o viceversa
- Trasferimento dei dati (I/O)
- Controllo del flusso del programma
 - Salto con o senza test

Quanti indirizzi sono necessari in una istruzione?

- Indirizzi necessari:
 - Un indirizzo per ogni operando (1 o 2)
 - Uno per il risultato
 - Indirizzo istruzione successiva
- Quindi al massimo quattro indirizzi
 - Ma molto raro, e sarebbe molto dispendioso
- Di solito 1, 2 o 3 per gli operandi/risultati

Numero di indirizzi

■ 1 indirizzo

- ☐ il secondo indirizzo è implicito
- ☐ di solito si tratta di un registro (accumulatore)
- ☐ situazione tipica nei primi calcolatori

Numero di indirizzi

■ Zero indirizzi

- ☐ tutti gli indirizzi sono impliciti
- ☐ utilizza una pila (stack)
 - Ad esempio, $c = a + b$ è realizzato come segue
 - push a
 - push b
 - add
 - pop c

Numero di indirizzi

Number of Addresses	Symbolic Representation	Interpretation
3	OP A, B, C	$A \leftarrow B \text{ OP } C$
2	OP A, B	$A \leftarrow A \text{ OP } B$
1	OP A	$AC \leftarrow AC \text{ OP } A$
0	OP	$T \leftarrow (T - 1) \text{ OP } T$

AC = accumulator

T = top of stack

(T - 1) = second element of stack

A, B, C = memory or register locations

Numero di indirizzi

- Meno indirizzi → istruzioni più elementari (e più corte), quindi CPU meno complessa
 - Però più istruzioni per lo stesso programma → tempo di esecuzione più lungo
- Più indirizzi → istruzioni più complesse
- Indirizzo di M o registro: meno bit per indicare un registro
- RISC (Reduced Instruction Set Computer) verso CISC (Complex Instruction Set Computer)
 - Pentium sono CISC, PowerPC (Apple, IBM, Motorola) sono RISC

Numero di indirizzi

Instruction	Comment
SUB Y, A, B	$Y \leftarrow A - B$
MPY T, D, E	$T \leftarrow D \times E$
ADD T, T, C	$T \leftarrow T + C$
DIV Y, Y, T	$Y \leftarrow Y \div T$

(a) Three-address instructions

Instruction	Comment
MOVE Y, A	$Y \leftarrow A$
SUB Y, B	$Y \leftarrow Y - B$
MOVE T, D	$T \leftarrow D$
MPY T, E	$T \leftarrow T \times E$
ADD T, C	$T \leftarrow T + C$
DIV Y, T	$Y \leftarrow Y \div T$

(b) Two-address instructions

Instruction	Comment
LOAD D	$AC \leftarrow D$
MPY E	$AC \leftarrow AC \times E$
ADD C	$AC \leftarrow AC + C$
STOR Y	$Y \leftarrow AC$
LOAD A	$AC \leftarrow A$
SUB B	$AC \leftarrow AC - B$
DIV Y	$AC \leftarrow AC \div Y$
STOR Y	$Y \leftarrow AC$

(c) One-address instructions

Figure 12.3 Programs to Execute $Y = \frac{A - B}{C + (D \times E)}$

Progettare un insieme di istruzioni

- Repertorio
 - quante e quali operazioni
- Tipi di dato
 - su quali dati
- Formato
 - lunghezza, numero indirizzi, dimensione campi, ...
- Registri
 - numero dei registri della CPU indirizzabili dalle istruzioni
- Indirizzamento
 - modo di specificare gli indirizzi degli operandi

Tipi degli operandi

- Indirizzi (interi senza segno)
- Numeri
 - Limite al modulo
 - Limite alla precisione
- Caratteri
- Dati logici

Numeri

- Interi (virgola fissa)
- Virgola mobile
- Quando ci sono soprattutto operazioni di I/O, si usano i decimali impaccati
 - Cifra decimale = 4 bit (0=0000, 1=0001, 2=0010, ..., 8=1000, 9=1001)
 - Inefficiente: solo 10 delle 16 configurazioni vengono usate
 - Es.: 246 = 0010 0100 0110
 - Più lunga della notazione binaria, ma evita la conversione

Caratteri

- Codice ASCII (American Standard Code for Information Exchange)
- Carattere = 7 bit → 128 caratteri in totale
- Caratteri alfabetici + caratteri di controllo
- Di solito 8 bit: un bit per controllo di errori di trasmissione (controllo di parità)
 - Settato in modo che il numero totale di bit a 1 sia sempre pari (o sempre dispari)
 - Es.: 00011100 → ottavo bit a 1
 - Se si ricevono 8 bit con n.ro dispari di 1, c'è stato un errore di trasmissione

Dati logici

- n bit, invece che un singolo dato
- Per manipolare i bit separatamente



Tipi di operandi

- Indirizzi
- Numeri
 - Interi o virgola mobile
- Caratteri
 - Es.: Codice ASCII (American Standard Code for Information Exchange): 7 bit per ogni carattere (128 caratteri), ottavo bit per controllo (settato in modo che il numero totale di 1 sia pari)
- Dati logici
 - Sequenza di bit invece che un singolo dato