

Introduzione all'Architettura degli Elaboratori

Sommario

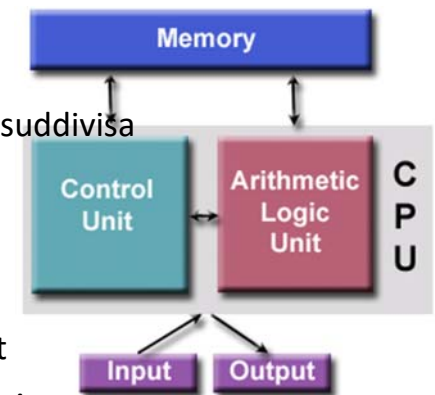
- Macchina di von Neumann
- Esecuzione dei programmi
- Rappresentazione dei dati
 - Dati numerici
 - Dati alfabetici

Il Modello di von Neumann (1)

- L'architettura di un computer è basata sul modello proposto da Janos (John) von Neumann alla metà degli anni '40 del secolo scorso

Il Modello di von Neumann (2)

- Prevede 3 entità logiche:
 - La memoria
 - La Central Processing Unit, suddivisa in
 - Control Unit
 - Arithmetic Logic Unit
 - I dispositivi di Input/Output
- Le informazioni (dati) viaggiano tra le componenti mediante un bus



Central Processing Unit (1)

- È la componente che acquisisce, interpreta ed esegue le istruzioni dei programmi
- Si compone a sua volta di
 - **Control Unit**, responsabile del prelievo e della decodifica delle istruzioni e dell'invio dei segnali di controllo
 - **Arithmetic Logic Unit**, per l'esecuzione delle operazioni aritmetiche e logiche
 - Alcune varianti del modello prevedono anche un **clock**

Central Processing Unit (2)

- Da un punto di vista operativo, la CPU fa uso di alcuni **registri**:
 - Elementi di memoria i cui valori possono essere acceduti in lettura e scrittura molto velocemente
 - Registro Istruzione Corrente (**CIR**) contiene l'istruzione in corso di esecuzione
 - Contatore di Programma (**PC**) contiene l'indirizzo della prossima istruzione del programma in esecuzione

Memory (1)

- È la memoria centrale, di lavoro
 - da non confondersi con la memoria di massa in cui vengono immagazzinati dati e programmi quando non in uso
- Contiene gli elementi che il computer sta usando nella elaborazione corrente, e precisamente
 - Le istruzioni del programma in corso di esecuzione
 - I dati necessari all'esecuzione di quel programma

Memory (2)

- Si può immaginare la memoria come costituita da tante celle, ognuna identificata univocamente da un proprio indirizzo

I/O

- Sono i dispositivi con cui rispettivamente
 - vengono forniti dati e programmi al computer
 - vengono prodotti dal computer i risultati dell'elaborazione

Elementi di supporto

- La macchina di von Neumann è una macchina astratta, un modello per realizzare macchine reali
- Per questo non vengono enfatizzati ulteriori elementi logici che pure sono necessari, come a esempio
 - i già citati bus e clock di sistema
 - le memorie di massa
 - le interfacce di rete
 - ...

Esecuzione dei Programmi (1)

- La macchina di von Neumann è in grado di eseguire programmi espressi in un opportuno **linguaggio macchina**
 - Per le macchine reali il linguaggio macchina è codificato secondo codici binari o esadecimali

Esecuzione dei Programmi (2)

- Si suppone che il programma da eseguire sia caricato nella Memory prima dell'esecuzione
 - da qualche memoria in cui è stato precedentemente registrato
 - fornito in input dal programmatore
 - ...
- Si suppone inoltre che il programma sia suddiviso logicamente in due parti
 - L'insieme di istruzioni che devono essere eseguite
 - I dati (di input/output e di supporto calcolati e temporanei) su cui le istruzioni operano

Il Ciclo Fetch-Execute (1)

- L'esecuzione del programma avviene ripetendo iterativamente le fasi di
 - **Acquisizione (fetch)** dell'istruzione da eseguire dalla Memory
 - **Interpretazione (decodifica)** dell'istruzione
 - **Esecuzione** dell'istruzione

Il Ciclo Fetch-Execute (2)

- Il contenuto del PC viene caricato nel CIR
- Il PC viene aggiornato con l'indirizzo dell'istruzione successiva
- L'istruzione del CIR viene decodificata ed eseguita
 - Eventualmente accedendo a dati

Rappresentazione dei Dati

- Tutte le informazioni (dati e istruzioni) in un computer sono rappresentate in forma binaria
 - Cioè come sequenza finita di simboli '0' e '1'
- L'unità di informazione è il **bit (binary digit)**
 - un bit può assumere solo i valori 0 oppure 1
- Una sequenza di 8 bit costituisce un **byte**
 - Un byte può assumere uno tra i 2^8 valori
00000000, 00000001, 00000010, ..., 11111111
- Con **word** si intende N byte
 - N dipendente dallo specifico contesto

Rappresentazione di Dati Numerici (1)

- Data la finitezza della macchina computer, tutte le informazioni che in esso si possono rappresentare sono necessariamente **finite**
- Quando, a esempio, si parlerà di
 - “insieme dei numeri interi” si intenderà un'approssimazione finita di \mathbb{N}
 - “insieme dei numeri reali” si intenderà un'approssimazione finita di \mathbb{R}

Rappresentazione di Dati Numerici (2)

- Siamo abituati a pensare (e usare) i numeri interi in forma **decimale** e **posizionale**
 - Decimale, perché espressi in base 10, facendo uso di dieci simboli ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9')
 - Posizionale perché la posizione in cui compare un simbolo è fondamentale per la sua interpretazione (il numero 12 e il numero 21 sono rappresentati dagli stessi simboli '1' e '2', ma disposti in posizioni diverse)

Rappresentazione di Dati Numerici (3)

- Sia **N** un generico numero intero: in forma decimale e posizionale è rappresentato come la sequenza di **n** simboli (cifre decimali)

$$a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0$$

- dove le a_i sono le diverse cifre, con a_n cifra più significativa, e a_0 cifra meno significativa

- Se **N** è espresso in base **b**, allora vale

$$N_b = a_n * b^n + a_{n-1} * b^{n-1} + a_{n-2} * b^{n-2} + \dots + a_2 * b^2 + a_1 * b^1 + a_0 * b^0$$

Rappresentazione di Dati Numerici (4)

- Esempi
 - $485_{10} = (4 * 10^2 + 8 * 10^1 + 5 * 10^0)_{10}$
 - $3642_8 = (3 * 8^3 + 6 * 8^2 + 4 * 8^1 + 2 * 8^0)_{10} = (3 * 512 + 6 * 64 + 4 * 8 + 2 * 1)_{10} = (1536 + 384 + 32 + 2)_{10} = 1954_{10}$
 - $9C3_{16} = (9 * 16^2 + 12 * 16^1 + 3 * 16^0)_{10} = (9 * 256 + 12 * 16 + 3 * 1)_{10} = (2304 + 192 + 3)_{10} = 2499_{10}$
 - $10011001_2 = (1 * 2^7 + 0 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0)_{10} = (128 + 0 + 0 + 16 + 8 + 0 + 0 + 1)_{10} = 153_{10}$

Rappresentazione di Dati Numerici (5)

- Esempio di conversione di un numero decimale (342) in binario

$342 : 2 = 171$ con resto 0
 $171 : 2 = 85$ con resto 1
 $85 : 2 = 42$ con resto 1
 $42 : 2 = 21$ con resto 0
 $21 : 2 = 10$ con resto 1
 $10 : 2 = 5$ con resto 0
 $5 : 2 = 2$ con resto 1
 $2 : 2 = 1$ con resto 0
 $1 : 2 = 0$ con resto 1

diventa il bit meno significativo

$$342_{10} = 101010110_2$$

diventa il bit più significativo

Rappresentazione di Interi (1)

- Possiamo considerare due casi:
 - Interi senza segno - **unsigned integer**: si intendono solo i positivi
 - Interi con segno - **signed integer**: si intendono i positivi e i negativi
 - Immaginiamo che il computer possa utilizzare una word di N byte per rappresentare un integer (sia signed che unsigned): $N \text{ byte} = N * 8 \text{ bit} = W \text{ bit}$

Rappresentazione di Interi (2)

- Avendo a disposizione W bit è possibile rappresentare i valori binari da 0 a 2^W-1
- Nel caso si voglia rappresentare gli interi signed il bit più significativo viene utilizzato per indicare il segno
 - Convenzionalmente '0'='+' e '1'='-'
- In tal caso quindi con W bit si possono rappresentare gli interi compresi tra $-(2^{W-1}-1)$ e $+(2^{W-1}-1)$

Rappresentazione di Reali (1)

- Nell'ambito di un computer l'approssimazione di \mathbb{R} espressa da "numeri reali" indica più precisamente un'approssimazione (finita) dei numeri razionali
- Concettualmente, un numero "reale" è rappresentato dalla giustapposizione di due numeri
 - Se W bit sono disponibili, i primi W_i indicano la parte intera, i rimanenti W_f la parte frazionaria
- Tale rappresentazione è detta a **virgola fissa (fixed point)**

Rappresentazione di Reali (2)

- La modalità di rappresentazione più diffusa è quella a **virgola mobile (floating point)**
- Utilizza due valori
 - **Mantissa**, interpretato come numero frazionario tra -1 e +1
 - **Caratteristica**, usato come esponente
- Si basa sulla notazione esponenziale, secondo cui
$$r = m * b^n$$
 - dove r è il numero reale da rappresentare, m è la mantissa, b è la base di rappresentazione e n la caratteristica

Rappresentazione di Caratteri (1)

- Anche i caratteri sono rappresentati mediante codici binari
- Più in generale si parla di “simboli” per specificare oltre agli usuali caratteri alfabetici, anche i simboli che indicano
 - le cifre (decimali)
 - la punteggiatura
 - i simboli speciali (blank, carriage return, linefeed, ...)

Rappresentazione di Caratteri (2)

- Per poter codificare univocamente i caratteri è necessaria una corrispondenza biunivoca tra questi e un opportuno sottoinsieme degli interi
 - Standard ASCII (American Standard Code for Information Interchange)
 - UNICODE

Rappresentazione di Caratteri (3)

- Funzioni di trasferimento
 - $\text{ord}(c)$: numero d'ordine del simbolo c nella tavola di codifica
 - $\text{chr}(i)$: Simbolo il cui numero d'ordine è i
- Proprietà
 - $\text{ord}(\text{chr}(i))=i$ $\text{chr}(\text{ord}(c))=c$
 - SE $c1 < c2$ ALLORA $\text{ord}(c1) < \text{ord}(c2)$
- Relazione d'ordine totale
 - Coerente con i sottoinsiemi delle lettere e delle cifre