

Appunti di Calcolatori Elettronici

Organizzazione dei sistemi di elaborazione

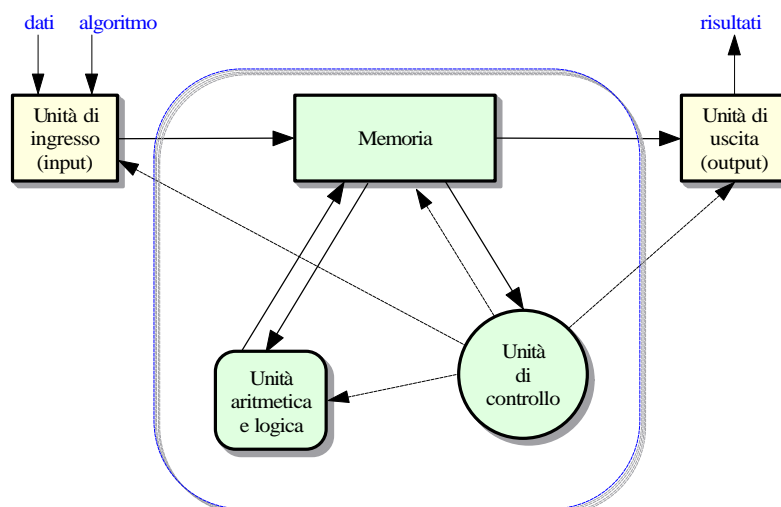
Introduzione.....	1
La CPU.....	2
Ciclo di esecuzione delle istruzioni	3
Insieme delle istruzioni	4
Organizzazione della CPU	4
La memoria.....	6
I bit.....	6
Gli indirizzi di memoria	7
Proprietà generali della memoria: RAM e ROM.....	8

Introduzione

Un **calcolatore digitale** è costituito da un sistema interconnesso dei seguenti dispositivi:

- *processori;*
- *memorie;*
- *dispositivi di I/O.*

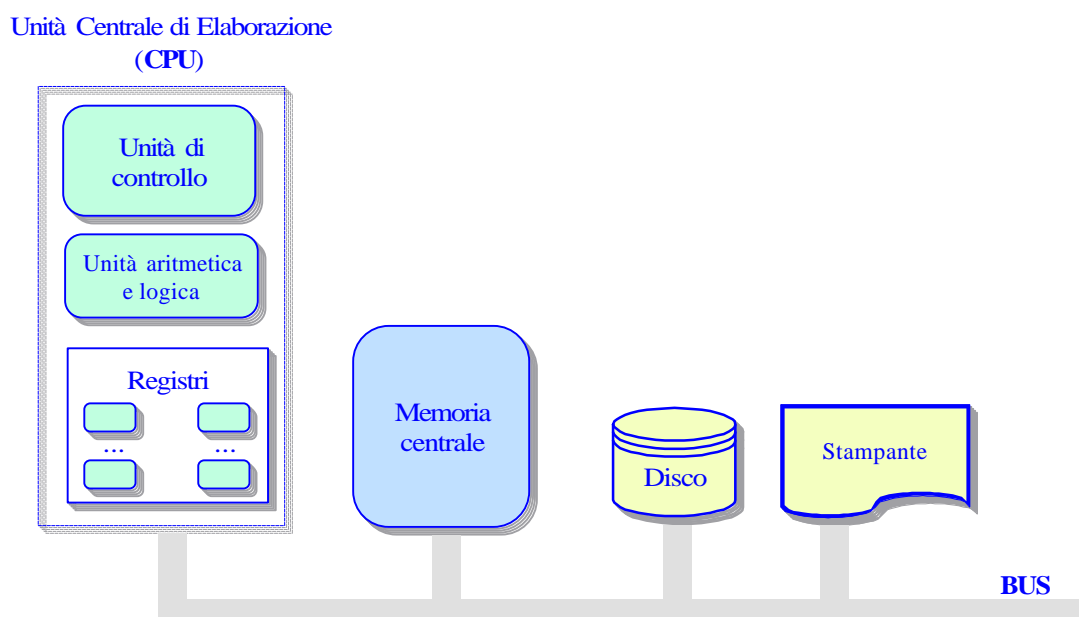
La figura seguente riporta una schematizzazione assolutamente generale di un *calcolatore automatico*:



L'insieme dell'*unità aritmetica e logica* e dell'*unità di controllo* è quello che oggi si definisce **processore**.

La CPU

L'organizzazione di un semplice calcolatore dotato di **bus** per il collegamento tra i dispositivi è mostrata nella seguente:



Organizzazione di un semplice calcolatore con una CPU e due dispositivi di I/O

- il cuore del sistema è costituito dalla **CPU** (*Unità centrale di elaborazione*), la quale consta a sua volta delle seguenti componenti:
 - **unità di controllo**
 - **unità aritmetica e logica (ALU)**
 - **registri interni;**
- in posizione fisicamente adiacente alla CPU c'è la **memoria centrale** (di tipo **RAM**, *Random Access Memory*, cioè ad accesso casuale uniforme);
- seguono i vari **dispositivi di Input** (disco, tastiera, mouse, scanner, ...) e di **Output** (video, stampante, ...).

Occupiamoci in particolare della CPU e delle sue componenti:

- l' **unità di controllo** ha il compito di prelevare le istruzioni dalla memoria principale e di determinarne il tipo, oltre che di sovrintendere al funzionamento generale del calcolatore;
- l' **unità aritmetico-logica** esegue le operazioni elementari (OR, AND, XOR, somma, moltiplicazione,...) richieste dalle istruzioni;
- i **registri interni** sono una specie di piccola memoria ad altissima velocità, usata dalla CPU per memorizzare i *risultati temporanei* delle proprie operazioni e certe *informazioni di controllo*; ognuno di questi registri ha una particolare funzione; ad esempio:
 - il registro più importante è il **Program Counter (PC)**, il quale punta alla prossima istruzione da eseguire ⁽¹⁾;
 - il registro delle istruzioni **Instruction Register (IR)** contiene di volta in volta l'istruzione da eseguire.

Ciclo di esecuzione delle istruzioni

La CPU esegue un **programma** (ossia una sequenza di istruzioni) attraverso la successione dei seguenti passi:

1. legge la nuova istruzione dalla memoria (usando l'indirizzo contenuto nel registro PC) e la pone nel registro delle istruzioni (IR); questa fase è detta di **fetch** dell'istruzione;
2. cambia il valore del registro PC, in modo che punti all'istruzione seguente;
3. determina il tipo di istruzione appena prelevata, leggendo i primi bit del registro IR (i quali recano il *codice operativo* dell'istruzione);
4. se l'istruzione usa dei dati (*operandi*) nella memoria, determina la loro posizione;
5. preleva i suddetti dati e li pone nei registri interni della CPU;
6. esegue l'istruzione, tramite l'uso dell'unità aritmetica e logica;

¹ Si noti che il nome "contatore di programma" è decisamente fuorviante, in quanto non ha niente a che fare con il conteggio di qualcosa.

7. memorizza il risultato nel "posto" previsto dalla istruzione (sia esso un registro o una locazione di memoria);
8. ritorna al passo 1 e comincia ad eseguire l'istruzione seguente.

Spesso, questa sequenza di passi viene detta **ciclo preleva-decodifica-esegui**, per evidenti motivi. Essa è alla base del funzionamento di tutti i calcolatori.

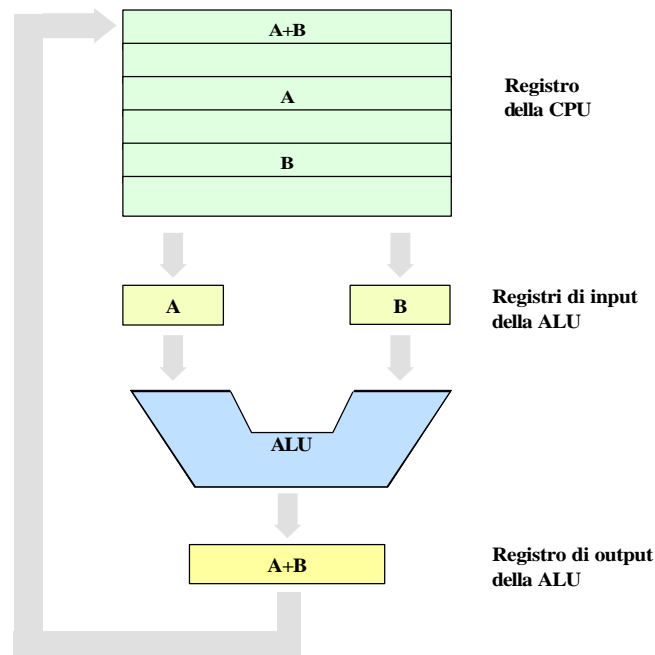
Insieme delle istruzioni

Il gruppo di tutte le istruzioni disponibili per un programmatore, ad un determinato livello, si chiama **insieme di istruzioni** di quel livello. Il numero di istruzioni contenute in tale insieme varia sia da livello a livello sia da macchina a macchina. Un grosso insieme di istruzioni non necessariamente è migliore di uno piccolo; al contrario, spesso vale il concetto opposto, dato che un grosso insieme di istruzioni significa che tali istruzioni sono piuttosto specifiche, mentre invece i *compilatori dei linguaggi ad alto livello* (come il Pascal o il C) lavorano meglio su macchine con insiemi di istruzioni piccoli e ben scelti.

Le macchine con insiemi di istruzioni molto piccoli sono dette macchine **RISC** (*Reduced Instruction Set Computers*) e si contrappongono alle macchine **CISC** (*Complex Instruction Set Computers*): esempi di macchine CISC sono i processori Intel a partire dal 80386, oppure i più vecchi IBM 360, Motorola 780030 o il DEC VAX. Le macchine RISC non usano la microprogrammazione e sono molto veloci.

Organizzazione della CPU

L'organizzazione interna di una parte della classica **CPU di Von Neumann** è mostrata dettagliatamente nella figura seguente, facendo riferimento ad un esempio semplice di somma tra due numeri:



Cammino dei dati di una tipica macchina di Von Neumann

Quando l'*unità aritmetica e logica* deve eseguire una operazione tra due dati, il **cammino** di tali dati, secondo lo schema di Von Neumann, è il seguente:

- i due dati vengono prelevati dai *registri interni della CPU* (oppure dalla memoria);
- vengono quindi inviati a due **registri di input** per l'ALU;
- l'ALU esegue l'operazione richiesta sui dati contenuti in tali registri e deposita il risultato in un **registro di output** (sempre specifico della ALU), dal quale i dati verranno successivamente trasportati nella destinazione finale (che potrà essere un registro della CPU, come in figura, oppure una locazione della memoria centrale).

A proposito della posizione degli operandi di una istruzione e del risultato di una operazione su di essi, si distinguono tre categorie di istruzioni:

- **istruzioni da memoria a registro**, che permettono il passaggio delle parole dalla memoria ai registri della CPU, dove possono essere usate come input dell'ALU, quali operandi di istruzioni;

- **istruzioni da registro a registro**, che prendono i due operandi dai registri della CPU, li portano nei registri di input della ALU, eseguono l'operazione su di essi e memorizzano il risultato di nuovo in un registro della CPU;
- **istruzioni da memoria a memoria**, che portano gli operandi dalla memoria direttamente ai registri di input dell'ALU, eseguono l'operazione e riscrivono il risultato direttamente in memoria.

Il cammino dei dati è il cuore della maggior parte delle CPU, in quanto, in un certo senso, definisce ciò che la macchina può fare.

La memoria

La **memoria** è quella parte del calcolatore in cui sono immagazzinati i *programmi* ed i *dati*. Senza una memoria in cui i processori possano leggere e scrivere informazioni, non esisterebbero calcolatori digitali con programmi memorizzati.

I bit

L'unità fondamentale della memoria è la cifra binaria, detta **bit**. Ogni bit può contenere uno 0 oppure un 1. E' dunque l'unità più semplice possibile da memorizzare ed elaborare.

Spesso si dice che i calcolatori usano l'**aritmetica binaria** perché è più *efficiente*; in realtà, l'aritmetica binaria gode di due pregi: l'*efficienza* e l'*affidabilità*. Cominciamo proprio dall'**affidabilità**: l'informazione digitale può essere memorizzata distinguendo tra i diversi valori di qualche grandezza fisica continua (o, meglio, *analogica*), come la tensione o la corrente; quanti più valori devono essere distinti, tanto meno separazione esisterà tra valori adiacenti e quindi tanto meno la memoria sarà affidabile; allora, nel caso del sistema di numerazione binaria, i valori da distinguere sono solo due, il che lo rende il sistema più affidabile possibile per la codifica delle informazioni.

Talvolta, si pubblicizzano alcuni calcolatori, come i grandi mainframe IBM, dicendo che sono dotati dell'**aritmetica decimale** oltre che di quella binaria. Il "trucco", in questa affermazione, è semplice: si usano 4 bit per memorizzare una singola cifra decimale, secondo una tecnica chiamata **BCD** (*Binary-Coded Decimal*).

Evidentemente, dato che 4 bit forniscono $2^4=16$ combinazioni, 10 di queste sono usate per le 10 cifre decimali (da 0 a 9), mentre le rimanenti 6 rimangono inutilizzate. Così facendo, ad esempio, il numero 1944 sarà codificato nel modo seguente:

sistema decimale: 0001 1001 0100 0100
 sistema binario: 0000011110011000

In entrambi i casi, sono stati usati 16 bit, ma con una differenza fondamentale: i 16 bit del formato decimale possono memorizzare solo i numeri da 0 a 9999, permettendo cioè solo 10000 combinazioni, mentre invece i 16 bit binari consentono di rappresentare 65536 diverse combinazioni, ossia più di 6 volte tanto. Per questa ragione si dice che l'aritmetica binaria è più efficiente.

Gli indirizzi di memoria

La memoria principale di un calcolatore è costituita da un numero di **celle** (o **locazioni**), ognuna delle quali può immagazzinare un *elemento di informazione*. Ogni cella è individuata da un proprio numero (detto **indirizzo**) con il quale i programmi possono riferirsi ad essa. Se una memoria ha N celle, allora tali celle avranno indirizzi compresi tra 0 e N-1.

cella #0
cella #1
cella #2
cella
cella #N-2
cella #N-1

Schematizzazione della divisione in celle di una memoria, numerate in modo progressivo da 0 ad N-1, dove N è il numero di celle

Ogni cella di memoria contiene lo stesso numero di bit. Se una cella contiene K bit, allora essa può contenere una qualsiasi delle 2^K combinazioni diverse di tali bit. Le *celle adiacenti* hanno ovviamente indirizzi consecutivi.

I calcolatori che usano l' **aritmetica binaria** esprimono anche gli **indirizzi di memoria** tramite numeri binari. Ad esempio, se un indirizzo di memoria binario è

costituito da M bit, significa che il numero massimo di celle *direttamente indirizzabili* è 2^M . Il numero di bit nell'indirizzo è dunque collegato al numero massimo di celle direttamente indirizzabili, ma invece non dipende affatto dal numero di bit per ogni cella. Ad esempio, una memoria con 2^{12} celle ciascuna da 8 bit avrebbe bisogno di indirizzi a 12 bit così come una memoria con 2^{12} celle da 60 bit: ciò che conta è il numero di celle e non la loro dimensione.

La cella è l'unità indirizzabile più piccola. Negli ultimi anni, molti costruttori hanno standardizzato celle a 8 bit, dette **byte**. A loro volta, i byte sono raggruppati in **parole**. Il numero di byte costituenti una parola varia da calcolatore e calcolatore: ad esempio, un calcolatore con una parola a 16 bit ha evidentemente 2 byte/parola, mentre invece un calcolatore con una parola a 32 bit ha evidentemente 4 byte/parola (questo è il caso oggi più comune).

La ragione per l'uso delle parole è che molte istruzioni operano su parole intere: un esempio classico è la somma tra due parole. Di conseguenza, una **macchina a 16 bit** (ossia con parole a 16 bit) avrà registri a 16 bit e istruzioni per manipolare parole a 16 bit, mentre invece una **macchina a 32 bit** (ossia con parole a 32 bit) avrà registri a 32 bit e istruzioni per spostare, sommare, sottrarre e manipolare parole a 32 bit.

Proprietà generali della memoria: RAM e ROM

Le memorie che possono essere sia scritte sia lette sono chiamate **RAM (Random Access Memory)**, ossia *memoria ad accesso casuale*), che in realtà è un termine improprio (per quanto ormai diffusamente utilizzato), dato che tutti i chip di memoria sono accessibili in maniera casuale.

Le RAM si dividono in due varietà:

- le **RAM statiche (SRAM)** sono costruite internamente in modo che il loro contenuto venga mantenuto fin quando sono alimentate;
- le **RAM dinamiche (DRAM)** sono sostanzialmente delle matrici di piccoli condensatori, ognuno dei quali può essere caricato o scaricato, in modo da memorizzare uno 0 oppure un 1. Dato che la carica elettrica tende a disperdersi, ogni bit di una DRAM deve essere rinfrescato ogni pochi millisecondi se si vuole che i dati non svaniscano. Dato che deve essere la *logica esterna* a curare il "rinfrescamento", le memorie DRAM richiedono

una interfaccia più complessa rispetto alle SRAM; a fronte di questo svantaggio, le DRAM hanno il pregio di capacità generalmente maggiori rispetto alle SRAM. Alcune DRAM hanno inoltre la logica di rinfrescamento direttamente sul chip, il che permette sia una grande capacità sia una interfaccia semplice: si parla in questo caso di **chip quasi-statici**.

Le RAM non sono comunque l'unico tipo di chip di memoria; in molte applicazioni, infatti, il programma da eseguire e parti dei dati necessari devono rimanere memorizzati anche quando la corrente viene disinserita; oltre a questo, molti programmi devono necessariamente rimanere inalterati una volta installati. Questi requisiti hanno portato allo sviluppo delle memorie **ROM (Read-Only Memory)**, ossia *memorie a sola lettura*): il contenuto di queste memorie non può essere cambiato o cancellato in alcun modo. Il contenuto di una ROM viene inserito durante la costruzione del chip, esponendo un materiale fotosensibile attraverso una maschera che contiene la configurazione desiderata di bit e poi incidendo la superficie esposta (oppure quella non esposta).

Le ROM sono molto più economiche delle RAM, specialmente se ordinate in grandi quantità per sostenere le spese della creazione della maschera. Tuttavia, queste memorie sono evidentemente poco flessibili proprio perché non modificabili. Le cosiddette memorie **PROM (Programmable ROM)**, ossia *ROM programmabili*) sono state allora introdotte per facilitare lo sviluppo di nuovi prodotti basati su ROM: il chip è come una ROM, ma può essere programmato (anche se una sola volta) sul campo.

Lo sviluppo successivo di questi chip è stato quello delle memorie **EPROM (Erasable PROM)**, ossia *PROM cancellabili*): questi chip non solo sono programmabili come le PROM, ma sono anche cancellabili. In poche parole, quando la cosiddetta "finestra al quarzo" di una EPROM viene esposta ad una forte luce ultravioletta per 15 minuti, tutti i bit vengono posti ad 1. Nei casi in cui sono necessari molti cambiamenti durante il ciclo di progettazione, la EPROM è senz'altro più economica della PROM, proprio perché può essere riutilizzata.

Ancora meglio delle EPROM sono le memorie **EEPROM (Electrically Erasable PROM)**, ossia *PROM cancellabili elettricamente*), dette anche **EAROM (Electrically Alterable ROM)**, ossia *ROM alterabili elettricamente*): queste memorie possono essere cancellate semplicemente applicando degli impulsi elettrici anziché inserendole in una camera speciale per l'esposizione alla luce ultravioletta.

Segnaliamo che una EEPROM differisce da una RAM poiché sia la scrittura di un byte sia la cancellazione di un byte richiedono un tempo migliaia di volte più lungo; al contrario, i tempi di accesso per la lettura delle memorie ROM in generale sono comparabili con quelli delle RAM.

Autore: **Sandro Petrizzelli**
e-mail: sandry@iol.it
sito personale: <http://users.iol.it/sandry>