



Architettura degli Elaboratori I - B

Concetti di Base e Storia

Introduzione

Daniel Riccio/Alberto Aloisio

Università di Napoli, Federico II

5 marzo 2018



Rif. Capitolo 1

ARM System-on-Chip Architecture, 2Ed, Addison Wesley

Sommario



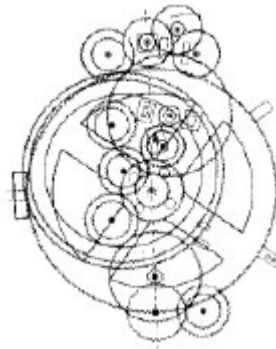
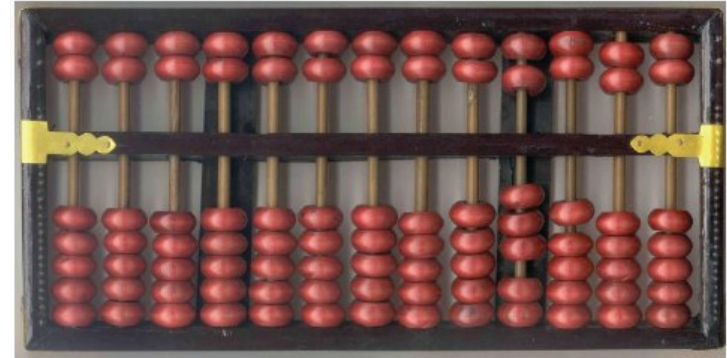
- ▶ Un po' di storia;
- ▶ Il Sistema Informatico;
- ▶ Memoria e indirizzamento;
- ▶ La CPU;
- ▶ Ciclo di esecuzione.



Un po' di storia

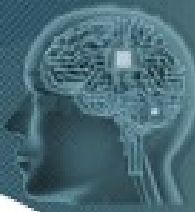
L' **Abaco cinese** (**3000 a.C.**) può essere considerato il primo modello matematico di calcolo. In questo modello si possono formalizzare le quattro operazioni (somma, sottrazione, prodotto e divisione intera).

<http://php.math.unifi.it/convegnostoria/materiali/GiacardiCiviltàExtraeuropee2.pdf>

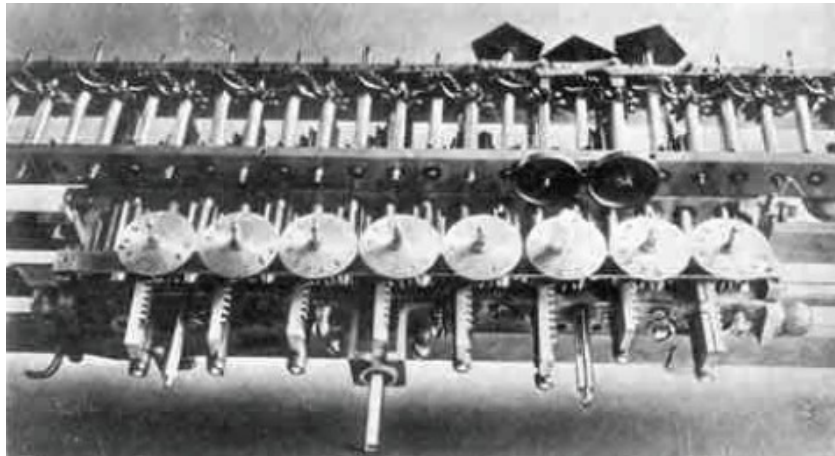


La **macchina di Antikythera** (**100 - 150 a.C.**) è un sofisticato planetario ritrovato in una isola greca. Esso è mosso da ruote dentate, che serviva per calcolare il sorgere del sole, le fasi lunari, i movimenti dei 5 pianeti allora conosciuti, gli equinozi, i mesi e i giorni della settimana.

Un po' di storia



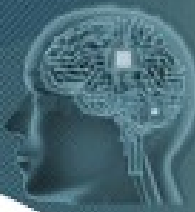
La **Pascalina** (1640) è un calcolatore meccanico per addizioni progettato da Blaise Pascal.



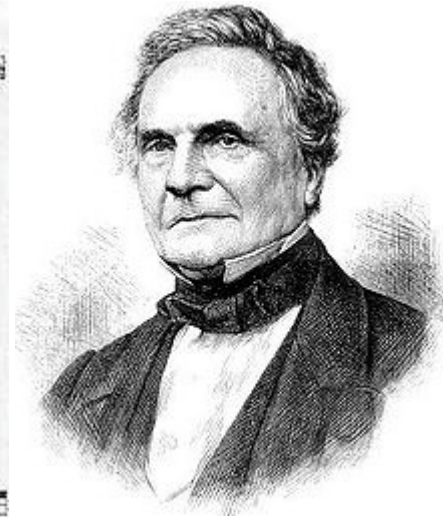
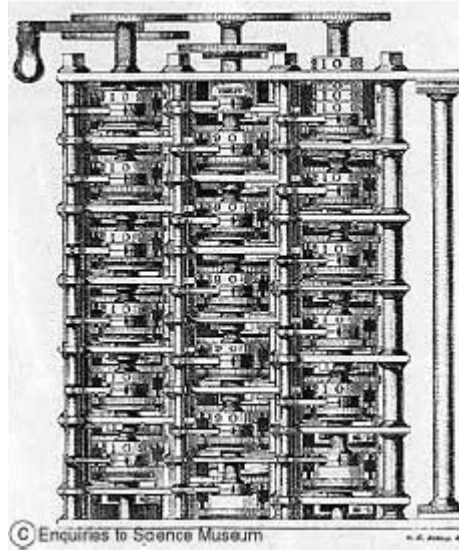
La **Stepped Reckoner** (1673) è una macchina per addizioni, sottrazioni, moltiplicazioni, divisioni e radice quadrata progettata da Gottfried Wilhelm von Leibniz.



Un po' di storia



La **Analithycal Engine** (1824) è una macchina “general-purpose” progettata da Babbage, le cui funzionalità dipendono da come è programmata.

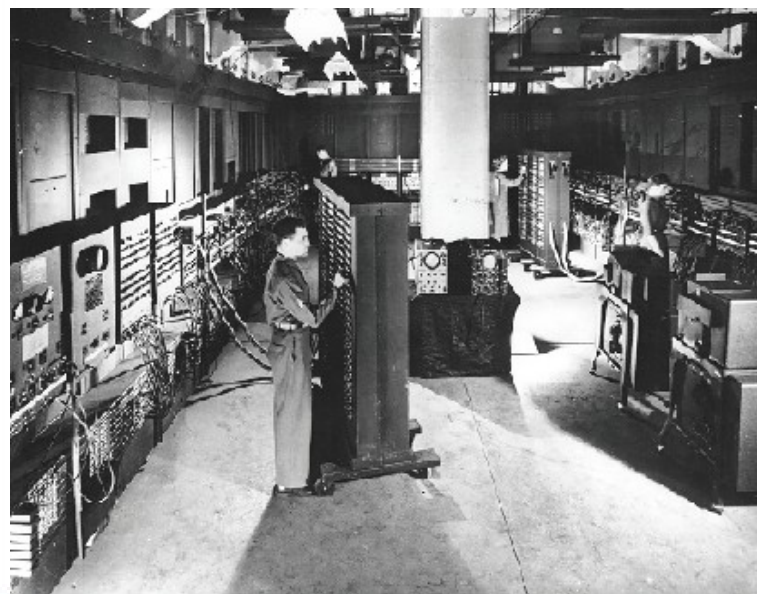


Ada Lovelace (1815-1852), collaborando con Babbage, scrive un algoritmo per calcolare i numeri di Bernoulli con la macchina di Babbage, oggi considerato il primo software della storia.



Un po' di storia

Le **Schede perforate** (1890) vengono introdotte da Herman Hollerit per automatizzare la tabulazione dei dati di un censimento.



L'**Electronic Numerical Integrator and Calculator** (ENIAC) (1946), è considerato primo calcolatore a valvole general-purpose programmabile progettato da Mauchly & Eckert (Univ. Pennsylvania).

Era programmato tramite inserimento di cavi e azionamento di interruttori.



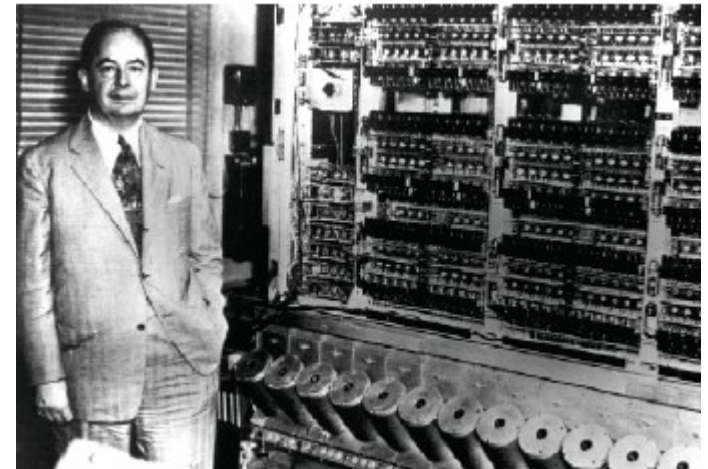
Un po' di storia



Lo **stored-program computer** (1945) viene introdotto da John von Neumann.

Dato che anche i programmi sono rappresentabili come numeri digitali, esattamente come i dati, Von Neumann propone di porre *programmi e dati assieme in memoria*

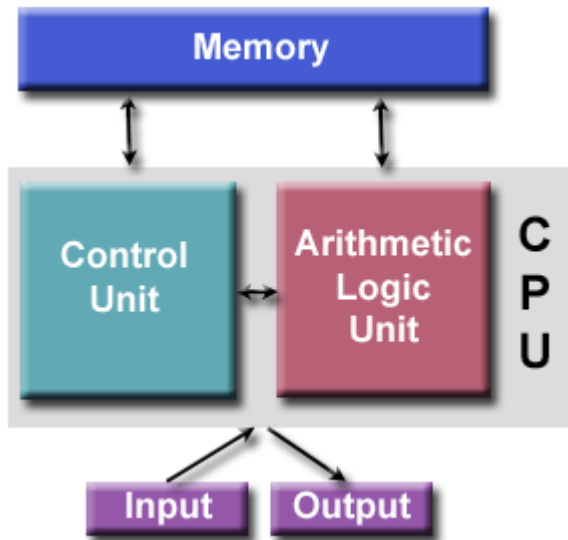
L'elaboratore assume la capacità di trattare qualsiasi informazione espressa in codice binario.



Le origini



Gli attuali computer general-purpose sono di tipo **stored program** (o a programma memorizzato), e rispecchiano il modello generale della macchina di **Von Neumann**.



John von Neumann, nato János Lajos Neumann (Budapest, 28 dicembre 1903 – Washington, 8 febbraio 1957), è stato un matematico, fisico e informatico ungherese naturalizzato statunitense.

Dal 1948, anno in cui il primo programma girava su un computer all'università di Manchester, la tecnologia ha conosciuto un incessante evoluzione:

- ▶ valvole;
- ▶ transistor e circuiti integrati (IC);
- ▶ very large scale integrated (VLSI) circuits.

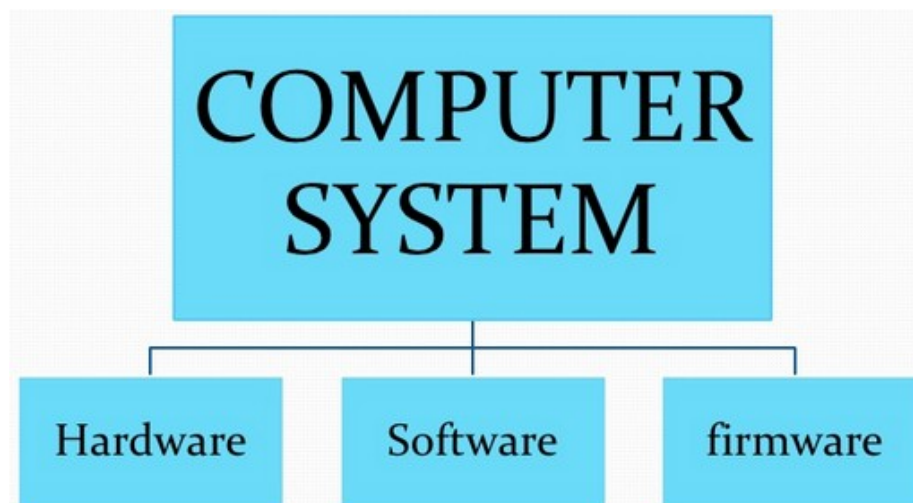
Il Sistema Informatico

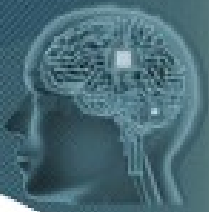


Prima scomposizione di un **sistema informatico**:

- ▶ **Hardware**: componenti fisici del sistema;
- ▶ **Software**: i programmi che vengono eseguiti dal sistema;

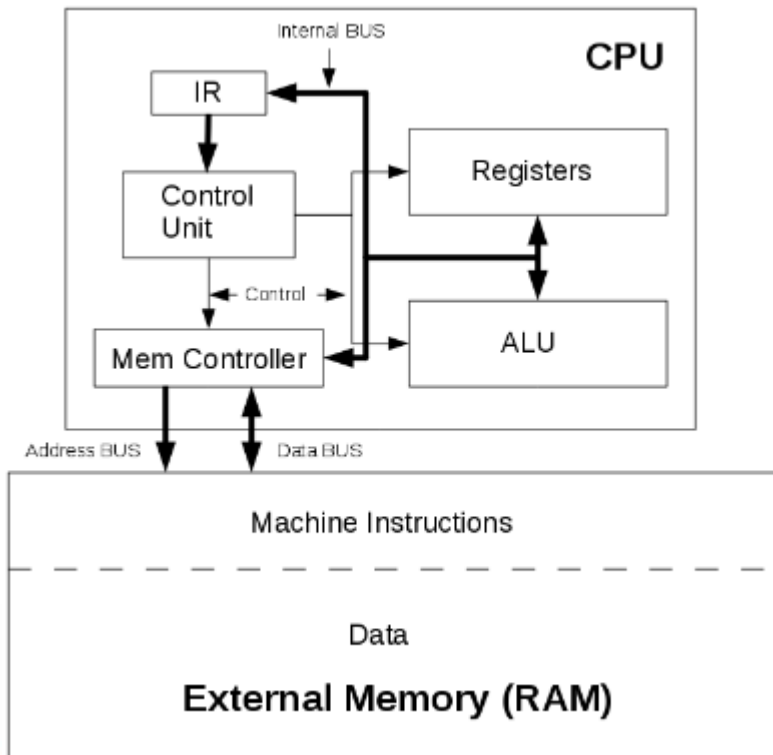
Il confine tra hardware e software in realtà non è sempre ben definito: in generale, vi è uno strato intermedio (il **firmware**) costituito da programmi direttamente mappati su circuiti elettronici.





Il Sistema Informatico

La **memoria centrale** contiene sia le **istruzioni** che i **dati** (informazioni necessarie per eseguire un programma).
Ha dimensioni limitate ed è volatile (cioè le informazioni memorizzate vengono perse allo spegnimento del computer).



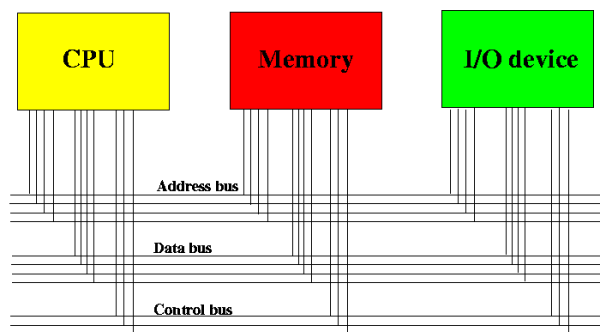
La **CPU** contiene i dispositivi elettronici in grado di acquisire, interpretare ed eseguire le istruzioni di ogni programma, trasformando i dati. Le istruzioni vengono eseguite in sequenza. Dati ed istruzioni vengono trasferiti da (e verso) la memoria centrale.

Il Sistema Informatico



Le **interfacce** verso le periferiche consentono uno scambio di informazioni fra l'elaboratore e l'esterno (**ingresso/uscita, memoria secondaria**).

Fra le **periferiche**, in particolare, la memoria secondaria (o memoria di massa) viene utilizzata per memorizzare grandi quantità di informazioni in modo persistente. Ha dimensioni elevate, ma l'accesso è meno rapido, rispetto alla memoria centrale.



Il **bus di sistema** collega questi elementi funzionali.

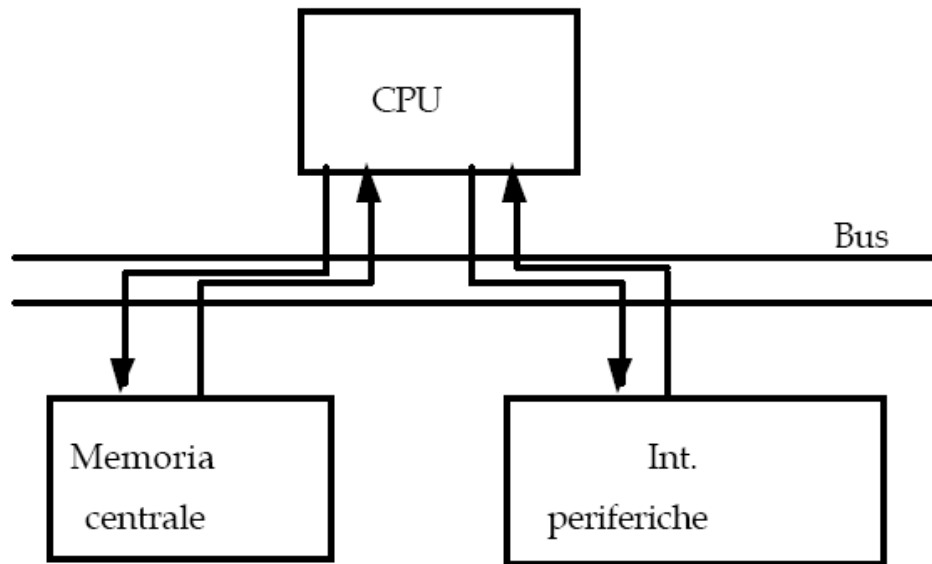
Fornisce il supporto fisico per la trasmissione dei dati tra i vari elementi.



II BUS

Interconnette la CPU, la memoria centrale e le interfacce verso dispositivi periferici (I/O, memoria di massa, etc.)

Collega due unità funzionali alla volta: una trasmette e l'altra riceve. Il trasferimento avviene sotto il controllo della CPU (Control Unit).



Su questo supporto (spesso costituito da più linee) viaggiano dati, indirizzi e comandi. Si distinguono spesso tali linee in:

bus dati (data bus)

bus indirizzi (address bus)

bus comandi (command bus)



La memoria centrale

Contiene dati ed istruzioni relativi al programma in esecuzione.

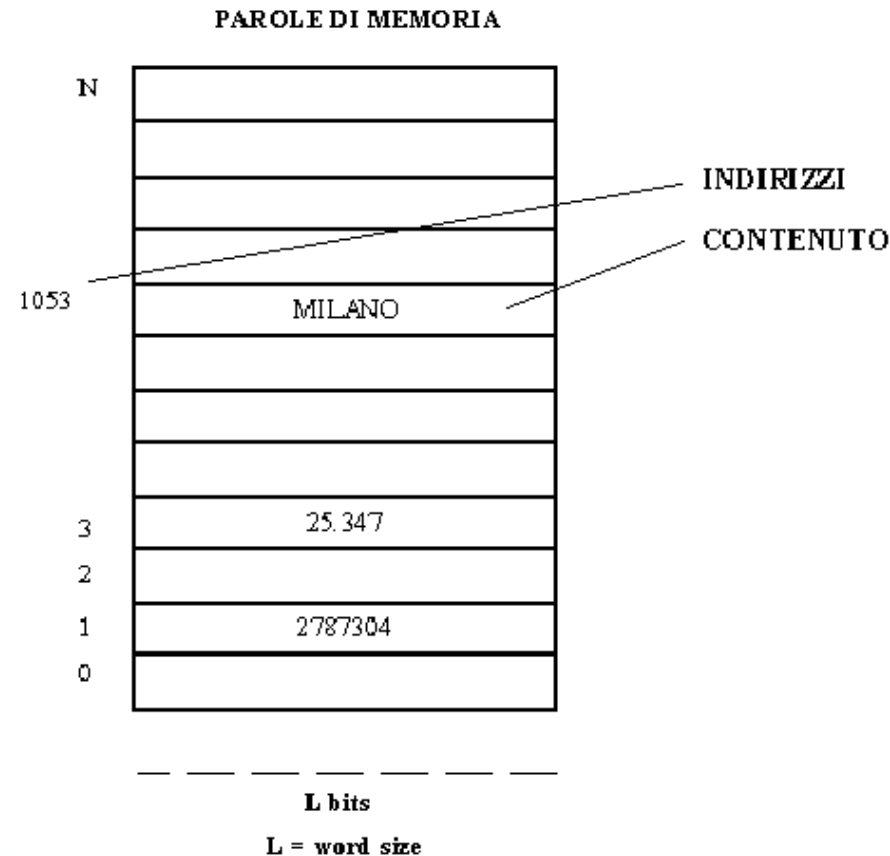
Ha una dimensione limitata.

È un passaggio obbligato per dati e istruzioni: la CPU può scambiare direttamente informazioni soltanto con la memoria centrale.

È **volatile** e, in generale, di dimensioni ridotte.

È una sequenza di celle di memoria, ciascuna contenente una sequenza di bit chiamata parola (**word**) di dimensione prefissata.

Ogni parola è caratterizzata da un indirizzo che la individua univocamente.





Concetto di memorizzazione

I sistemi di elaborazione sono realizzati con tecnologia digitale. Le informazioni sono rappresentate mediante segnali elettrici a 2 valori di tensione {**Vlow**, **Vhigh**} (oppure {**0**,**1**}).

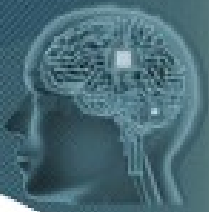
Per questo motivo, l'unità logica di memorizzazione (e, in generale, di rappresentazione delle informazioni) è il **bit** (binary digit):

Un bit è una grandezza il cui dominio di variazione è composto dai due valori {0,1}.

il **byte** equivale ad 8 bit

Potenze in base 2 del byte (con esponente multiplo intero di 10):

Kilobyte	2^{10} byte	1024 byte($\sim 10^3$)	KB
Megabyte	2^{20} byte	1048576 byte($\sim 10^6$)	MB
Gigabyte	2^{30} byte	$\sim 10^9$ byte	GB
Terabyte	2^{40} byte	$\sim 10^{12}$ byte	TB
Petabyte	2^{50} byte	$\sim 10^{15}$ byte	PB



Memorie Fisiche

RAM: Random Access Memory (ad accesso casuale): memoria di tipo volatile. Su di essa si possono svolgere operazioni di lettura e di scrittura.

ROM: Read Only Memory (a sola lettura): non volatile e non scrivibile; in essa vengono contenuti i dati e programmi per inizializzare il sistema.

CACHE. È un tipo di memoria particolarmente veloce ma molto costosa. Per questo motivo ha dimensioni limitate. Utile per memorizzare i dati usati più frequentemente.

PROM: Programmable ROM. Si può scrivere soltanto una volta, mediante particolari apparecchi (detti programmatori di ROM).

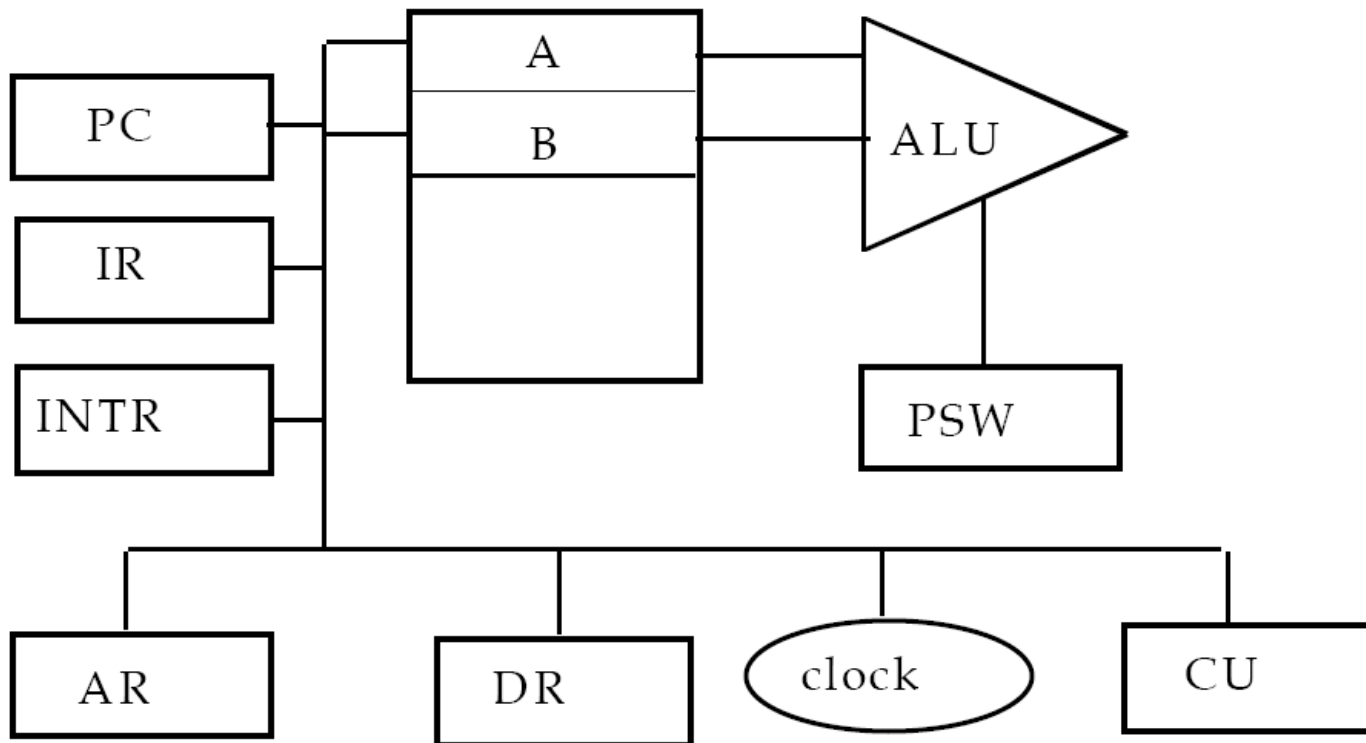
EPROM = Erasable-programmable ROM (è cancellabile sottoponendola a raggi ultravioletti).

Firmware : è costituito dal software memorizzato nelle ROM (codice microprogrammato).



Central Processing Unit (CPU)

La **Central Processing Unit**, o CPU, controlla l'elaborazione.



Central Processing Unit (CPU)



L'unità aritmetico-logica (**Arithmetic Logic Unit**, **ALU**) che esegue le operazioni elementari necessarie per l'esecuzione delle istruzioni.

L'unità di controllo (**Control Unit**, **CU**), controlla e coordina l'attività della CPU. In particolare, è responsabile del trasferimento e della decodifica delle istruzioni dalla memoria centrale ai registri della CPU.

Il **clock** (orologio) “cadenza” le operazioni elementari, permettendo il sincronismo delle operazioni.

Un **registro** (ad esempio, **A**, **B**, **PC**, **PSW**, etc.) è una locazione utilizzata per memorizzare dati, istruzioni, o indirizzi all'interno della CPU. L'accesso ai registri è molto veloce.

Nei moderni sistemi di elaborazione la CPU è realizzata da un unico **circuito integrato** (**chip**): il microprocessore.



Central Processing Unit (CPU)

DR (Data Register), registro dati: viene utilizzato per trasferire dati da e verso la memoria centrale.

AR (Address Register), registro indirizzi: viene utilizzato per memorizzare l'indirizzo della cella corrente nella memoria centrale (la sorgente/destinazione del trasferimento di dati).

PC (Program Counter), registro "contatore" del programma. Contiene l'indirizzo della prossima istruzione da eseguire.

IR (Instruction Register), registro istruzione corrente. Contiene, istante per istante, l'istruzione che è attualmente in esecuzione.

Registri accumulatori o di lavoro (ad esempio, A,B,..). Contengono operandi e risultati delle operazioni svolte dalla ALU.

INTR (Interrupt Register), registro interruzioni. Contiene informazioni sullo stato di funzionamento delle periferiche.

PSW (Program Status Word), i cui bit forniscono informazioni sul risultato dell'ultima operazione eseguita dalla ALU (overflow, zero, carry, segno)



Indirizzamento

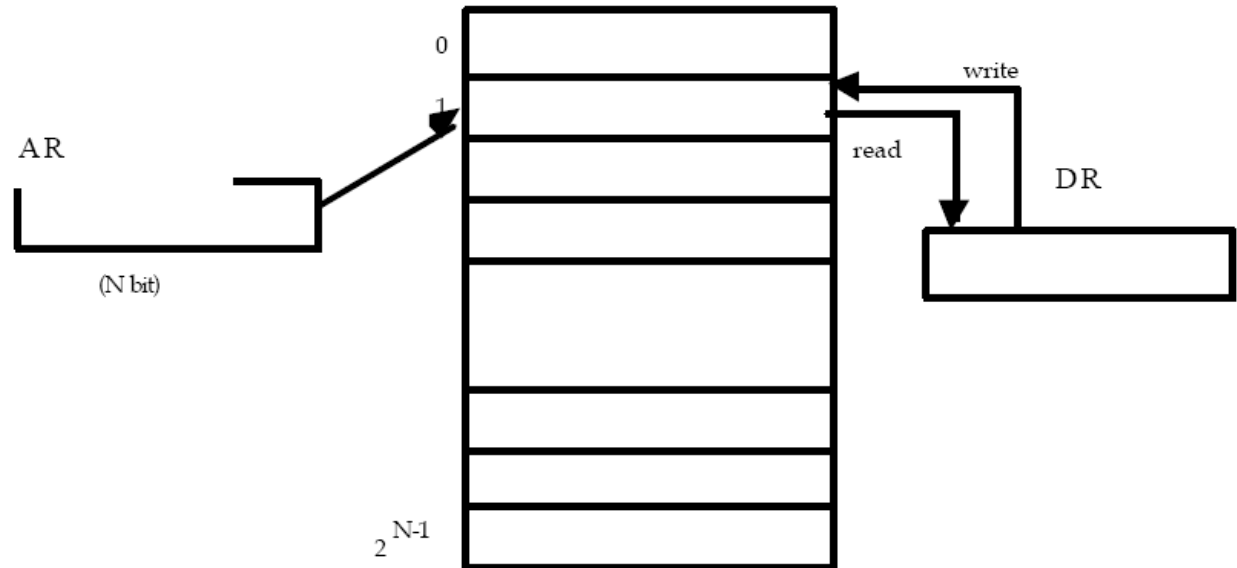
Se il registro indirizzi **AR** è lungo **N** bit, si possono indirizzare 2^N celle di memoria (con indirizzo da **0** a $2^N - 1$).

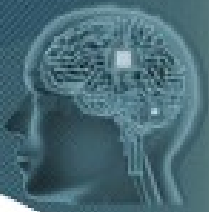
Indirizzata una cella attraverso AR, si possono eseguire operazioni di lettura e di scrittura da e verso il **registro dati DR**.

Una operazione di lettura trasferisce il contenuto della cella di memoria indirizzata dall'**AR** nel **DR**.

Una operazione di scrittura trasferisce il contenuto del **DR** nella cella di memoria indirizzata dall'**AR**.

Le operazioni vengono coordinate dalla **Control Unit (CU)**.





Central Processing Unit (CPU)

L'unità di controllo ha il compito di:

- reperire dalla memoria centrale le istruzioni di un programma (**fetch**);
- interpretarle (**decode**);
- farle eseguire (**execute**);
- registrare in memoria centrale eventuali risultati (**store**).

La CPU si comporta quindi come segue:

ripeti

FETCH di una istruzione

DECODE dell'istruzione

EXECUTE dell'istruzione

STORE eventuali risultati

finché istruzione = HALT oppure ERRORE



Ciclo di Esecuzione

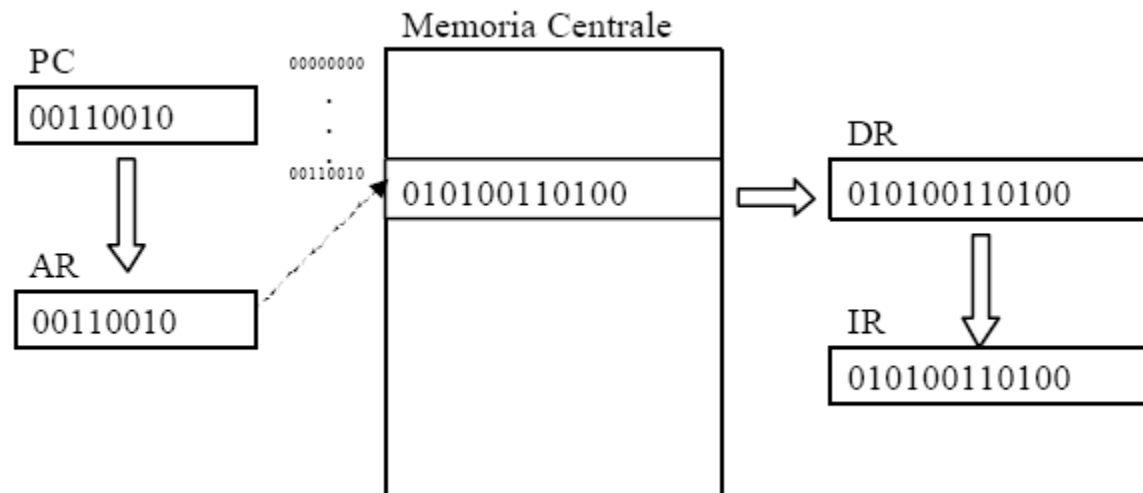
FETCH

Il contenuto del registro **PC** viene trasferito nel registro **AR**. Il registro **PC** contiene l'indirizzo della prossima istruzione da eseguire.

Avviene un'operazione di lettura dalla memoria centrale. Il contenuto della cella di memoria corrispondente all'indirizzo contenuto nell'**AR** viene trasferito nel **DR** tramite il bus di sistema.

Il contenuto del **DR** è trasferito all'**IR**.

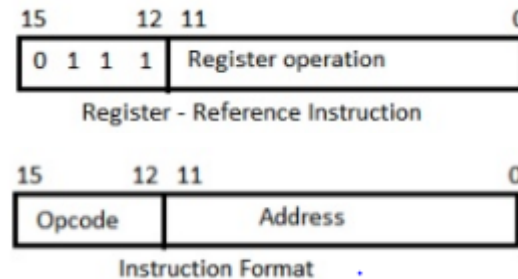
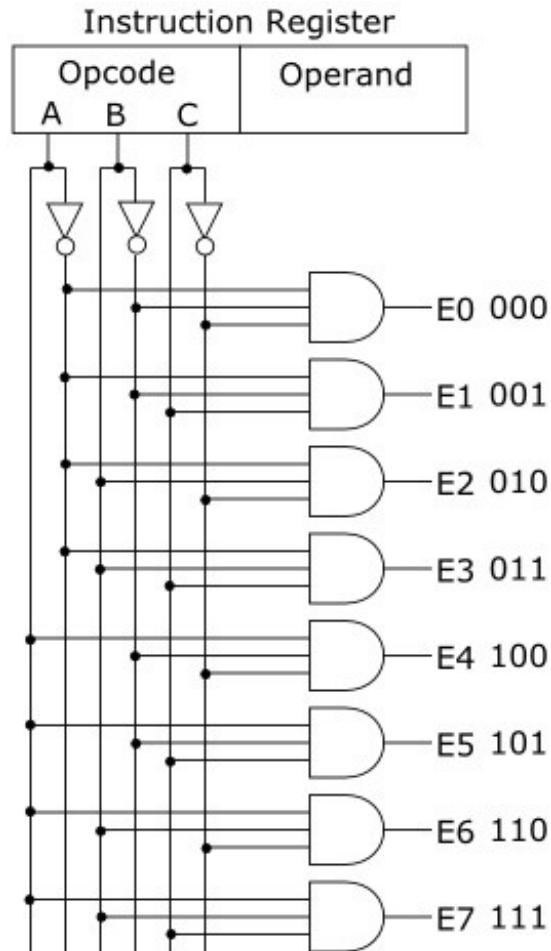
Il valore del registro **PC** viene incrementato di **1**, in modo da contenere l'indirizzo dell'istruzione successiva (in alcuni casi il valore di **PC** può essere diverso in base all'esecuzione di particolari istruzioni).





Ciclo di Esecuzione

DECODE: questa è la fase di interpretazione dell'istruzione.



Parte del contenuto dell'IR contiene il codice operativo dell'istruzione (ad esempio, l'operazione di lettura).

La parte rimanente contiene l'indirizzo dell'operando (ad esempio, quale dato leggere).



Ciclo di Esecuzione

EXECUTE: questa fase è differente per ogni operazione.

Consiste nell'esecuzione dell'operazione stessa. Operazioni possibili sono:

- il trasferimento dati da e verso la memoria e da o verso le periferiche;
- operazioni supportate dalla **ALU** e dai registri;
- l'accesso alla memoria centrale per l'individuazione delle celle di memoria avviene sempre attraverso l'**AR**;
- la lettura di un dato contenuto in una specifica cella di memoria comporta il trasferimento del contenuto della cella nel **DR**;
- la scrittura di un dato in una specifica cella di memoria comporta il trasferimento del contenuto del **DR** nella specifica cella di memoria;
- l'interazione con le periferiche avviene sempre attraverso il **DR**. In caso di operazione aritmetica, recuperato l'operando nel **DR** (tramite **AR** e accesso alla memoria centrale), questo viene trasferito verso uno dei registri accumulatori (**A** o **B**).

STORE: questa fase viene svolta quando viene richiesta la memorizzazione di un valore (eventualmente un calcolo aritmetico o un dato immesso da tastiera).



La ALU

Realizza le operazioni aritmetiche e logiche necessarie per l'esecuzione delle istruzioni.

ALU a due operandi (contenuti nei registri A e B) in grado di eseguire le operazioni aritmetiche (somma, sottrazione, prodotto, divisione):

1. I due operandi vengono **caricati** nei registri A e B;
2. La ALU viene attivata in funzione da un **codice operativo** inviato dalla CU che specifica il tipo di operazione;
3. Nel registro A viene caricato il **risultato** dell'operazione eseguita dalla ALU;
4. Il registro **PSW** riporta sui suoi bit indicazioni sul risultato dell'operazione (riporto, overflow, etc.).