Politecnico di Torino
Sede di Alessandria
Corso di informatica

Prof. Lorenzo Porcelli

e mail: genna18@iol.it

sito: users.iol.it/genna18

Organizzazione dei calcolatori

- Un computer digitale è una macchina in grado di risolvere problemi eseguendo istruzioni appositamente specificate.

- Una sequenza di istruzioni che specifica come risolvere un compito si chiama programma.

- Le istruzioni primitive di un calcolatore formano un linguaggio chiamato linguaggio macchina

Organizzazione dei calcolatori

Per risolvere un problema utilizzando un calcolatore digitale è necessario conoscere un linguaggio, comprensibile al calcolatore, attraverso cui descrivere la soluzione del problema come una serie di operazioni da svolgere.

Organizzazione dei calcolatori

Il linguaggio macchina (L0) è troppo complicato per essere utilizzato direttamente dagli uomini.

Bisogna progettare nuove istruzioni più facili da utilizzare. L'insieme di queste istruzioni crea un nuovo linguaggio (L1).

Organizzazione dei calcolatori

Il passaggio da L1 a L0 può avvenire in due modi:

- Traducendo tutto il programma scritto il L1 in una serie di istruzioni di L0. (**traduzione**)

- Scrivendo un programma in L0 che prenda programmi in L1 come dati di ingresso e li esegua esaminando ogni singola istruzione. (**interpretazione**)

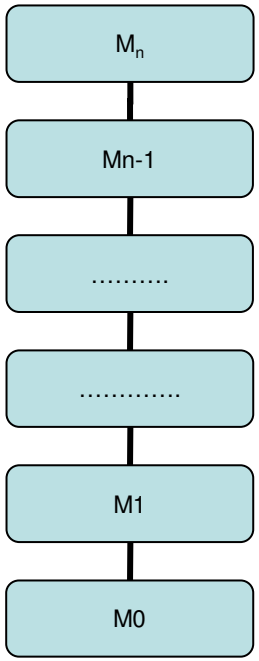
Organizzazione dei calcolatori

Macchina virtuale: calcolatore ipotetico definito attraverso un linguaggio macchina.

M1 è la macchina virtuale definita attraverso il linguaggio L1.

M2 è la macchina virtuale definita attraverso il linguaggio L2, che può essere più semplice da comprendere che il linguaggio L1. E così via …

Organizzazione dei calcolatori



I programmi della macchina M_n sono scritti in L_n

Vengono interpretati da una macchina a livello inferiore

o sono tradotti in un linguaggio macchina di un livello inferiore.

M_0 indica la macchina del livello digitale costituita da porte logiche.

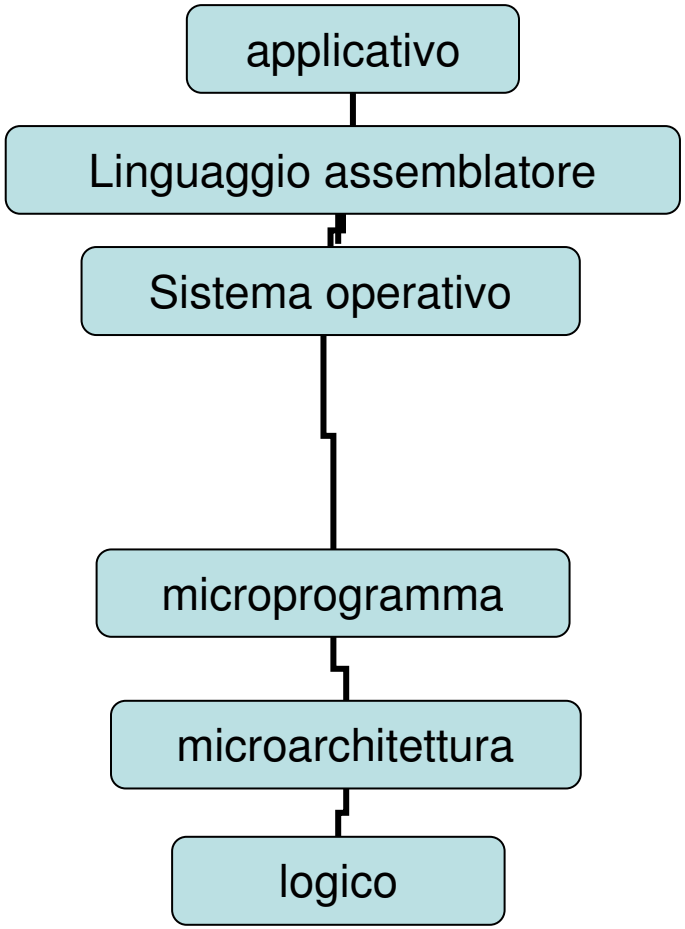
Organizzazione dei calcolatori

Ogni macchina virtuale ha un linguaggio macchina composto da tutte le istruzioni che quella macchina è in grado di comprendere ed eseguire.

Ogni macchina definisce un linguaggio *così come*

ogni linguaggio definisce una macchina.

Organizzazione di calcolatori



L5: linguaggi ad alto livello

L4: livello per programmatori

L3: livello ibrido. Contiene istruzioni livello sottostante. Gestione mem, multiproc.,

L2: livello instruction set. E' attivo interprete "sist. Op".

L1: A.L.U., registri, data path

L0: livello delle porte

Livello 0

Livello logico digitale: è composto da porte. Ogni porta ha uno o più input digitali (0, 1) e svolge operazioni tipo AND, OR.

Combinando tra loro porte è possibile realizzare registri.

Combinando più porte si formano i circuiti che implementano un calcolatore.

Livello 1

livello microarchitetturale: è composto dai registri (8..32) e da ALU (Unità logico aritmetica), in grado di svolgere operazioni. I registri sono collegati all'ALU dal *data path*.

Il funzionamento del data path è controllato da un microprogramma oppure direttamente dall'hardware.

Livello 2

E’ il livello **Instruction set**. I manuali dei produttori descrivono questo livello, le istruzioni a disposizione e il loro funzionamento.

Le istruzioni possono essere interpretate da un microprogramma al livello 1, o eseguite direttamente dall’hardware.

Livello 3

E’ il livello del **sistema operativo**. Gran parte delle istruzioni di questo livello appartengono anche al livello 2.

Le nuove istruzioni sono interpretate da un interprete del livello 2 che si chiama *sistema operativo.*

E’ un livello ibrido.

Livello 4

E’ il livello del **linguaggio assembler**.

E’ il primo livello pensato per i programmatori di applicazioni. Fornisce i meccanismi necessari per scrivere programmi per i livelli sottostanti.

Il linguaggio del livello 4 viene tradotto dall’assemblatore.

Livello 5

E’ il **livello applicativo**. Consiste di linguaggi, di alto livello, messi a disposizione dei programmatori per risolvere problemi.

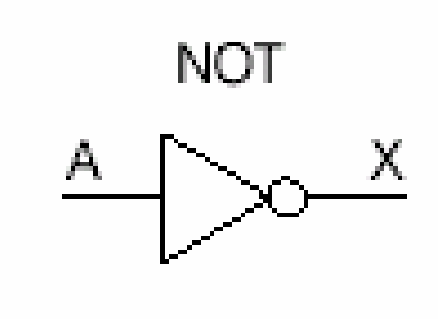
I linguaggi vengono tradotti dai compilatori.

Può anche fornire un interprete di dominio specifico, ad esempio la matematica, fornendo dati e operazioni facilmente comprensibili ad un esperto del settore.

Livello digitale

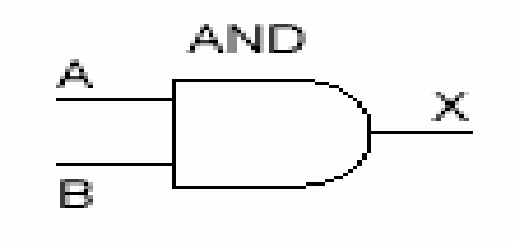
- In un circuito digitale ci sono solo due valori logici,0 e 1, associati a due tensioni diverse. Ad esempio 0..1 volt associato al valore 0, 2..5 volt associato al valore 1.
- Le porte sono dispositivi elettronici che possono calcolare alcune funzioni di questi segnali.
- Le porte rappresentano la base hardware.

Livello digitale: porta not



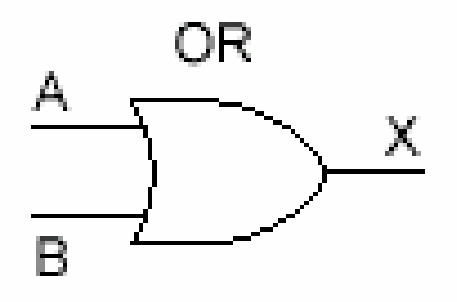
A	X
0	1
1	0

Livello digitale: porta and



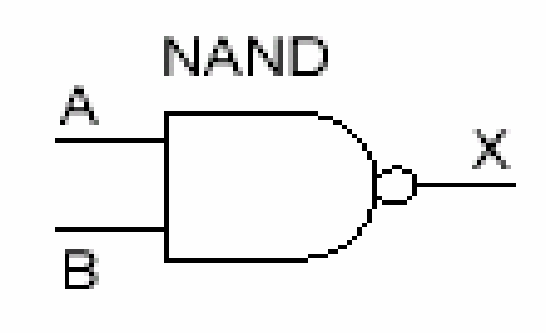
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Livello digitale: porta or



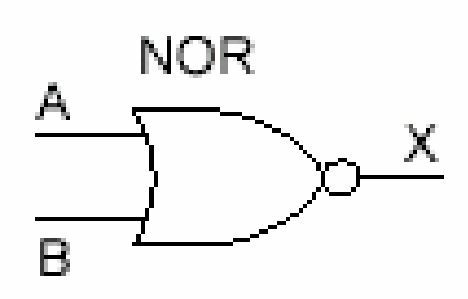
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Livello digitale: porta nand



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Livello digitale: porta nor



A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

algebra Booleana

- Le tecniche di composizione delle porte logiche in una rete sono derivate dall'*Algebra Booleana*, sviluppata dal matematico George Boole(1815-1864).
- Nel 1938 Shannon ha dimostrato come l'Algebra Booleana potesse essere presa a fondamento per la progettazione di circuiti logici digitali.

Algebra Booleana: identità

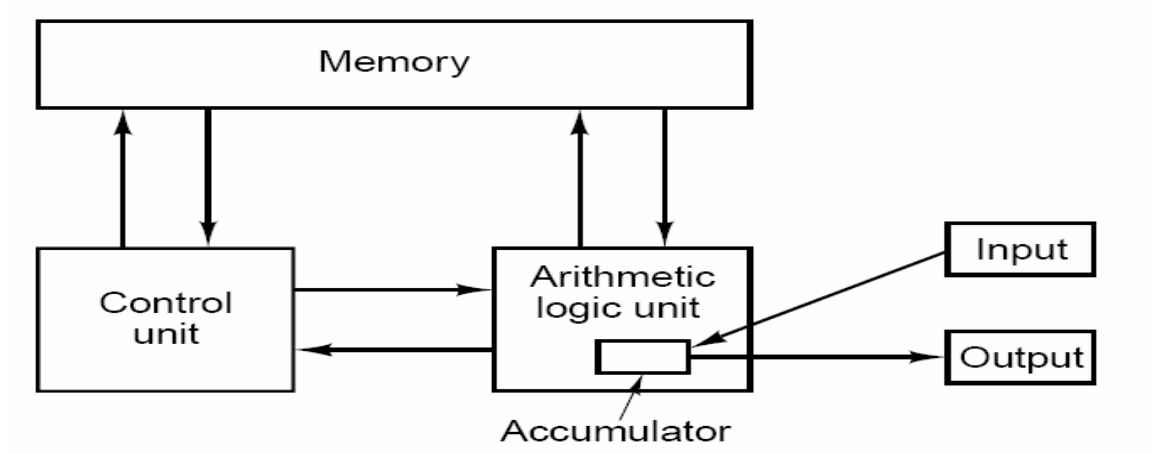
Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

Algebra booleana: distributiva

A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

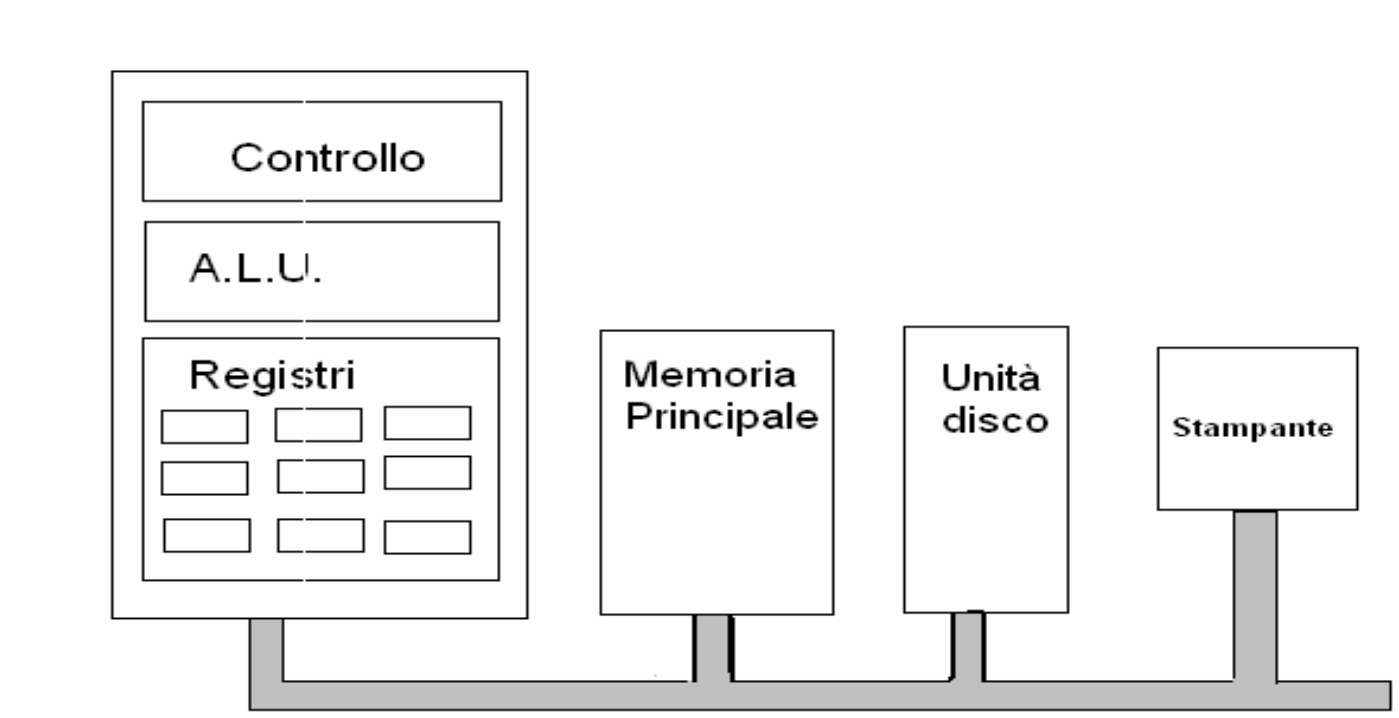
A	B	C	A	B + C	A(B + C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Macchina di von Neuman



Schema originale del primo calcolatore con il programma memorizzato in forma digitale, utilizzando l'aritmetica binaria. Dopo 60 anni è ancora alla base di moltissimi calcolatori.

Macchina di von Neuman



Architettura: unità centrale

La CPU si compone di parti distinte:

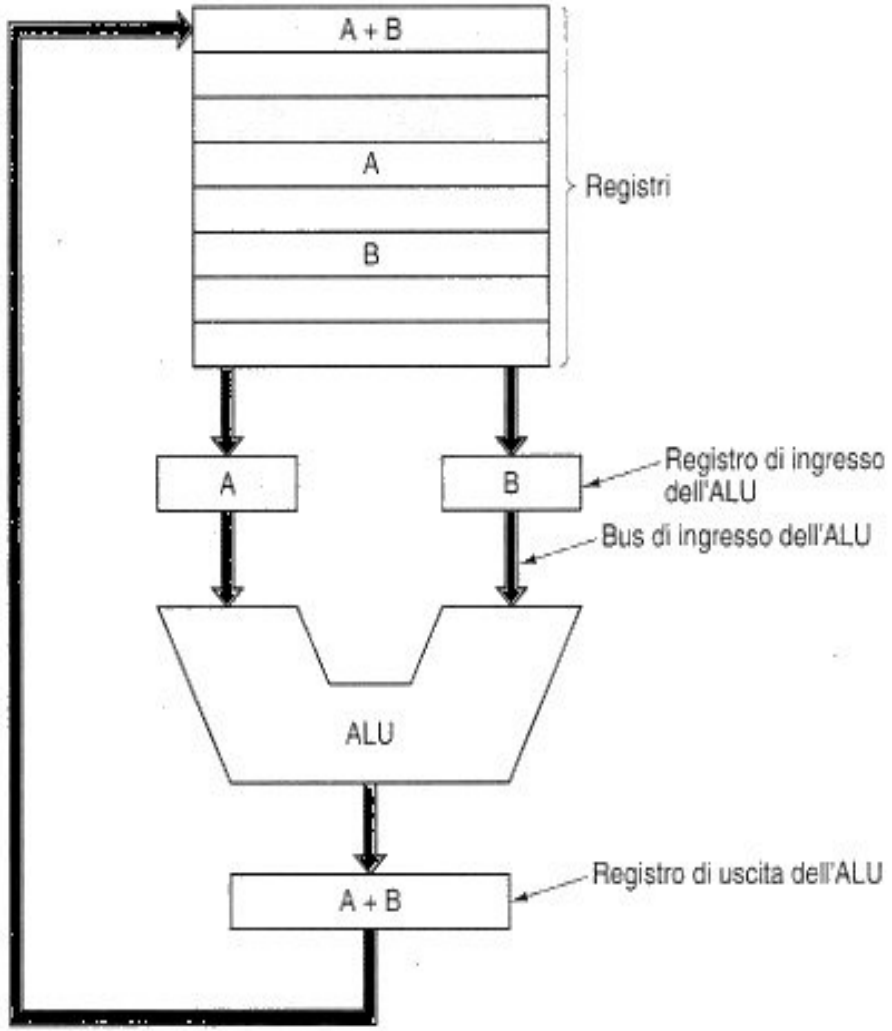
- Unità di controllo: legge le istruzioni dalla memoria e determina il tipo
- Unità aritmetico-logica: esegue le operazioni (addizione, or)
- Registri: piccola memoria ad alta velocità,per memorizzare risultati temporanei o informazioni di controllo

unità centrale: data path

Descrive l'organizzazione interna di una cpu;

comprende registri, alu, bus di collegamento.

Nell'esempio l'alu ha due registri di ingresso ed uno di uscita.



Cpu: esecuzione delle istruzioni

Ciclo fetch-decode-execute

1. Prendi una istruzione dalla memoria e mettila nel registro istruzioni
2. Cambia il p.c. per indicare istruzione seguente
3. Riconosci istruzione appena letta
4. Se l'istruzione usa una parola di memoria determina dove si trova
5. Metti la parola, se necessario, in un registro
6. Esegui l'istruzione
7. Torna al punto 1

C.P.U.: tipologia

Esistono due tipologie di Cpu:

C.I.S.C. **C**omplex **I**nstruction **S**et **C**omputer

La potenza della CPU è data, oltre che dal clock, dal numero di istruzioni e dalla loro complessità

R.I.S.C. **R**educed **I**nstruction **S**et **C**omputer

La potenza della CPU è data dalla velocità di esecuzione delle istruzioni

R.I.S.C.

La CPU è dotata di istruzioni semplici, che di conseguenza vengono direttamente eseguite dall'hardware in un solo ciclo del data path. Ogni istruzione viene pertanto messa in esecuzione molto velocemente.

C.I.S.C.

La Cpu è dotata di molte istruzioni, anche molto complesse, che quindi vengono interpretate da un microprogramma poiché altrimenti i costi di realizzazione sarebbero troppo elevati. L'esecuzione delle singole istruzione occupa più cicli del data path, quindi le singole istruzioni sono più lente in esecuzione, ma sono molto più potenti.

Risc - Cisc

Poiché in un programma la maggior parte delle istruzioni sono semplici, e poiché spesso i compilatori non sfruttano al meglio le istruzioni complesse a disposizione le prestazioni dei processori risc sono molto superiori.

Intel ha modificato con il 486 il processore cisc in modo che contenesse un piccolo processore risc che esegue direttamente le istruzioni semplici. La velocità rispetto ai risc è solo lievemente inferiore ma il vantaggio è la compatibilità con tutti i processori della famiglia.

Architettura: memoria principale

Vengono immagazzinati i programmi e i dati.

L'unità di base è il bit.

8bit formano un **byte**.

I byte vengono raggruppati in **parole**.

Un calcolatore a 32 bit ha una parola di 4 byte, registri interni di 32 bit, istruzioni a 32 bit.

Architettura: memoria principale

La memoria è composta da celle, tutte con uguale numero di bit.

Ogni cella ha un indirizzo.

Il numero di celle indirizzabili è 2^m, dove m è il numero di bit dell'indirizzo.

Se una memoria ha k celle significa che servono M bit per indirizzarla dove

$$2^M >= k$$

Architettura: memoria secondaria

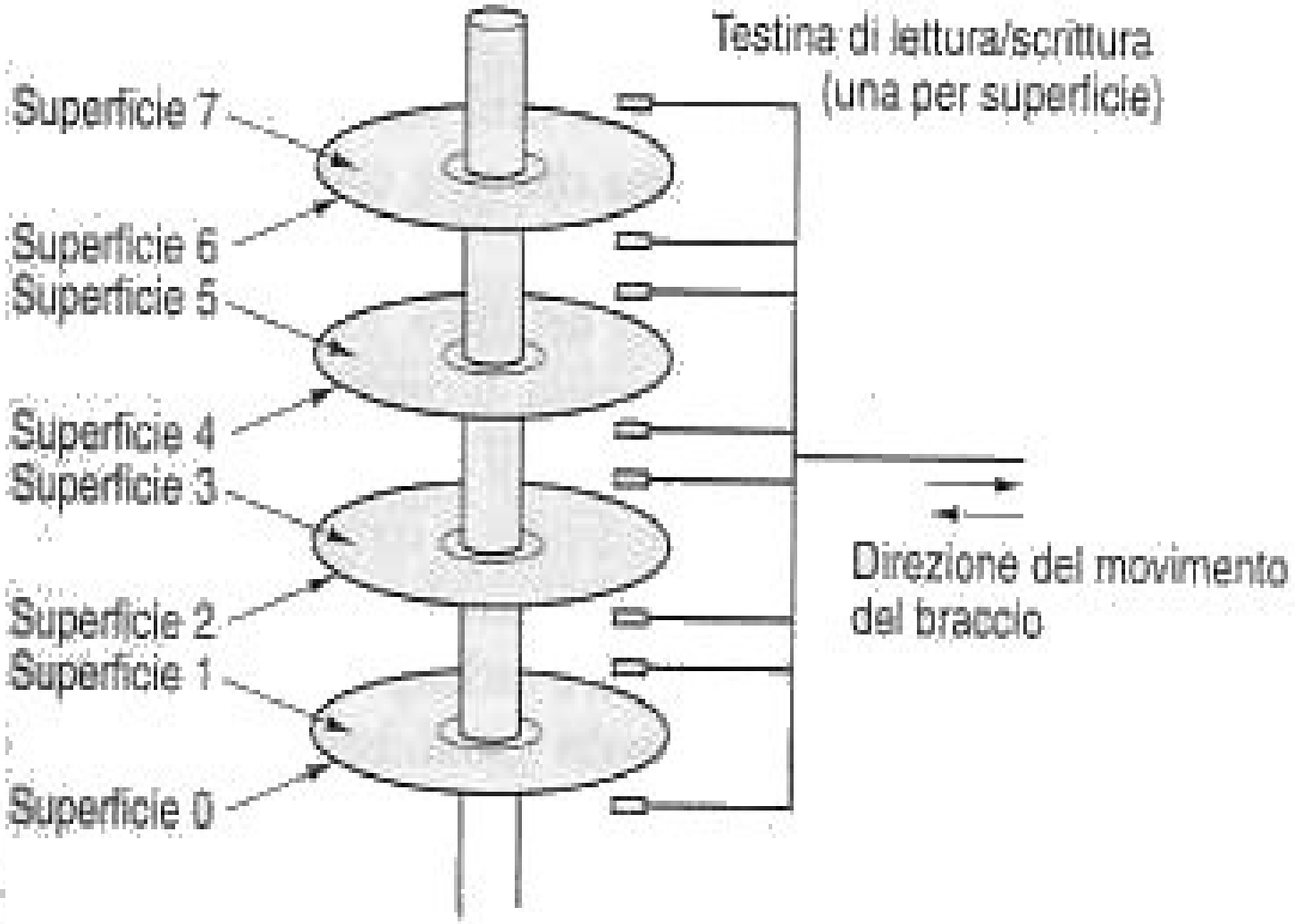
Il disco magnetico è ancora il supporto di riferimento quando si parla di memoria secondaria.

Si compone di uno o più *piatti di alluminio* con un rivestimento magnetizzabile.

Il diametro era inizialmente di 50 cm ma ora hanno dimensioni molto inferiori (3 – 12 cm).

La testina è sospesa sopra il disco ma non a contatto, sostenuta da un cuscinetto d'aria, ed opera attraverso l'induzione elettromagnetica.

Disco magnetico



Disco magnetico: terminologia

Una **traccia** è una sequenza di bit scritta mentre il disco compie una rotazione completa.

Ogni traccia è suddivisa in **settori** di lunghezza fissa. Ognuno è preceduto da un **preambolo** che permette alla testina di sincronizzarsi prima della lettura/scrittura.

I dati sono seguiti da un codice di **correzione d'errore**.

Tra due settori consecutivi si trova un piccolo spazio (**intersector gap**).

La capacità formattata di un disco è circa 15% inferiore rispetto alla capacità non formattata.

Disco magnetico: terminologia

Ogni disco ha braccia mobili su cui sono posizionate le **testine**.

I dischi sono composti da piatti multipli impilati verticalmente su cui si muovono contemporaneamente le testine.

L'insieme delle tracce in una data posizione radiale si chiama **cilindro**.

Dischi: prestazioni

Le prestazioni di un disco dipendono:

- Posizionamento del braccio sulla posizione radiale corretta (**seek**)
- Attesa che il settore desiderato si presenti sotto la testina (**latenza di rotazione**). Dipende dalla velocità di rotazione
- Il **tempo di trasferimento** dipende sia dalla velocità di rotazione sia dalla densità lineare.

I primi due tempi sono dell'ordine dei msec, il terzo, per un settore, è dell'ordine dei µsec

Dischi: problemi da risolvere

- Differenza tra velocità massima di trasferimento per un periodo limitato di tempo e velocità di trasferimento per un periodo illimitato di tempo
- I dischi ruotando si scaldano e si espandono cambiando la geometria: necessità di ricalibrarsi periodicamente.
- La velocità angolare del disco è costante, le tracce esterne sono più lunghe di quelle interne:
  - O cambia la densità di memorizzazione
  - O cambia il numero di settori passando dall'interno all'esterno

Dischi: controller

Ad ogni disco è associato un **controllore di disco**.

Il controller accetta comandi quali read, write, format, controlla il movimento del braccio, individua/corregge gli errori, trasforma gli 8 bit paralleli letti dalla memoria in un flusso seriale di bit e viceversa. Alcuni controller hanno buffering di settori multipli o caching di settori in previsione di nuove letture. Alcuni controller sono dotati di vere e proprie CPU.