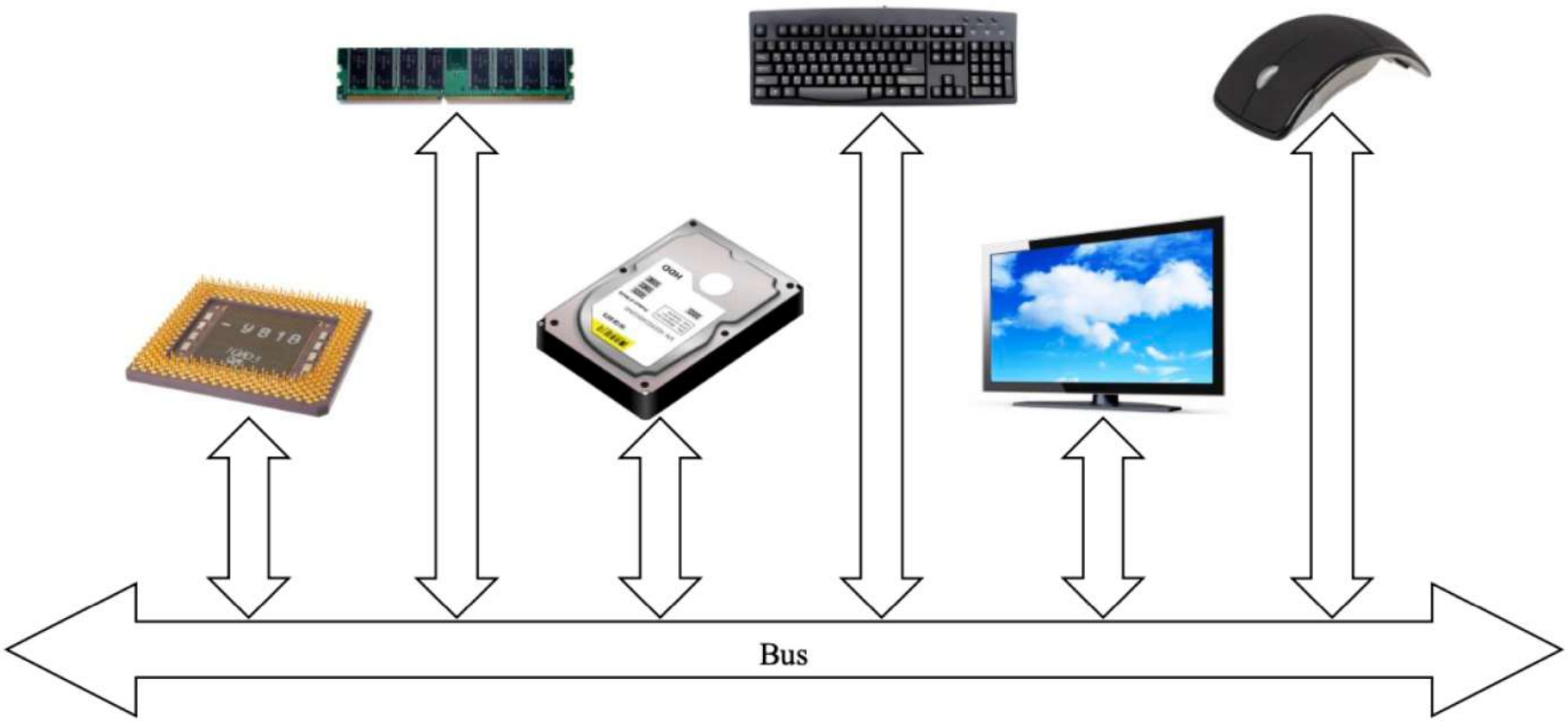
Organizzazione dei sistemi di calcolo

Componenti di un calcolatore digitale

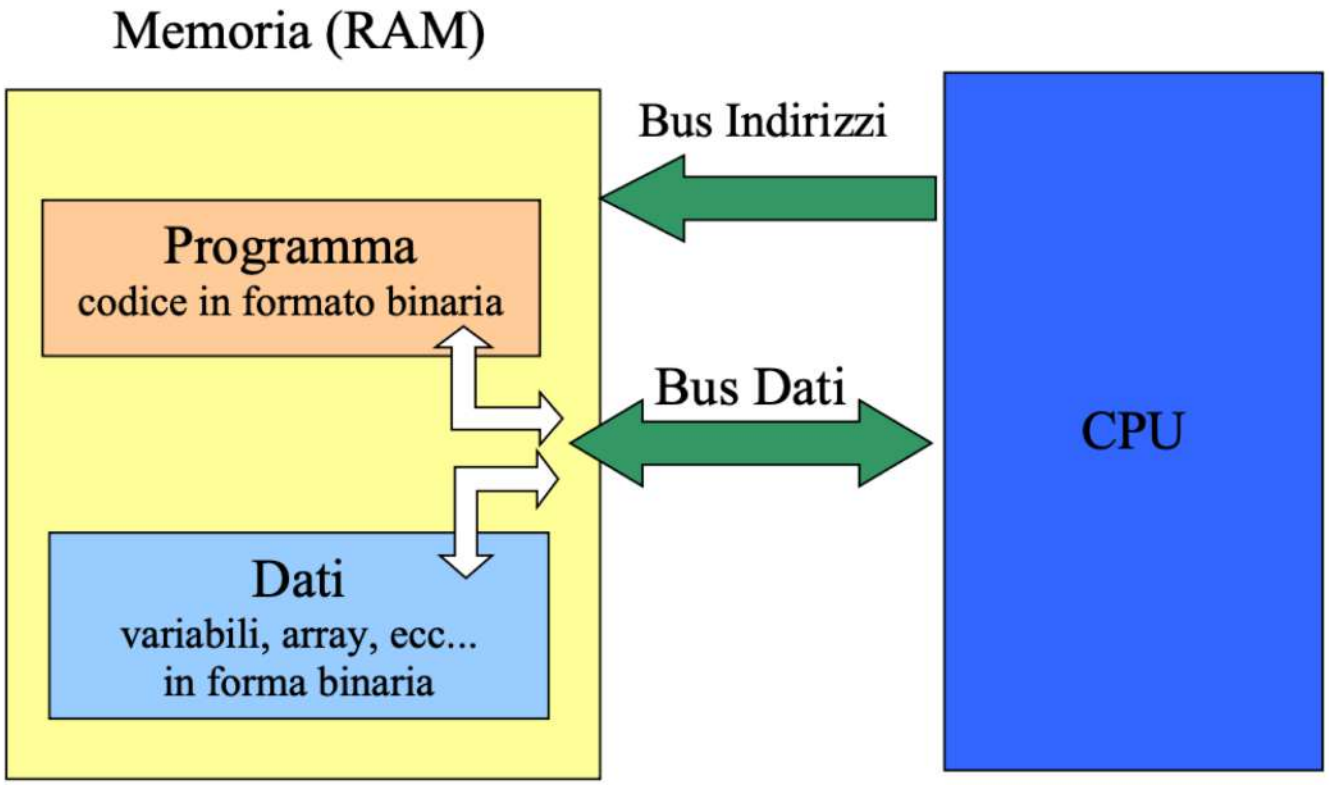
Un calcolatore digitale è un sistema composto da processori, memorie e dispositivi di input/output (I/O) collegati tra loro.



Architettura bus oriented

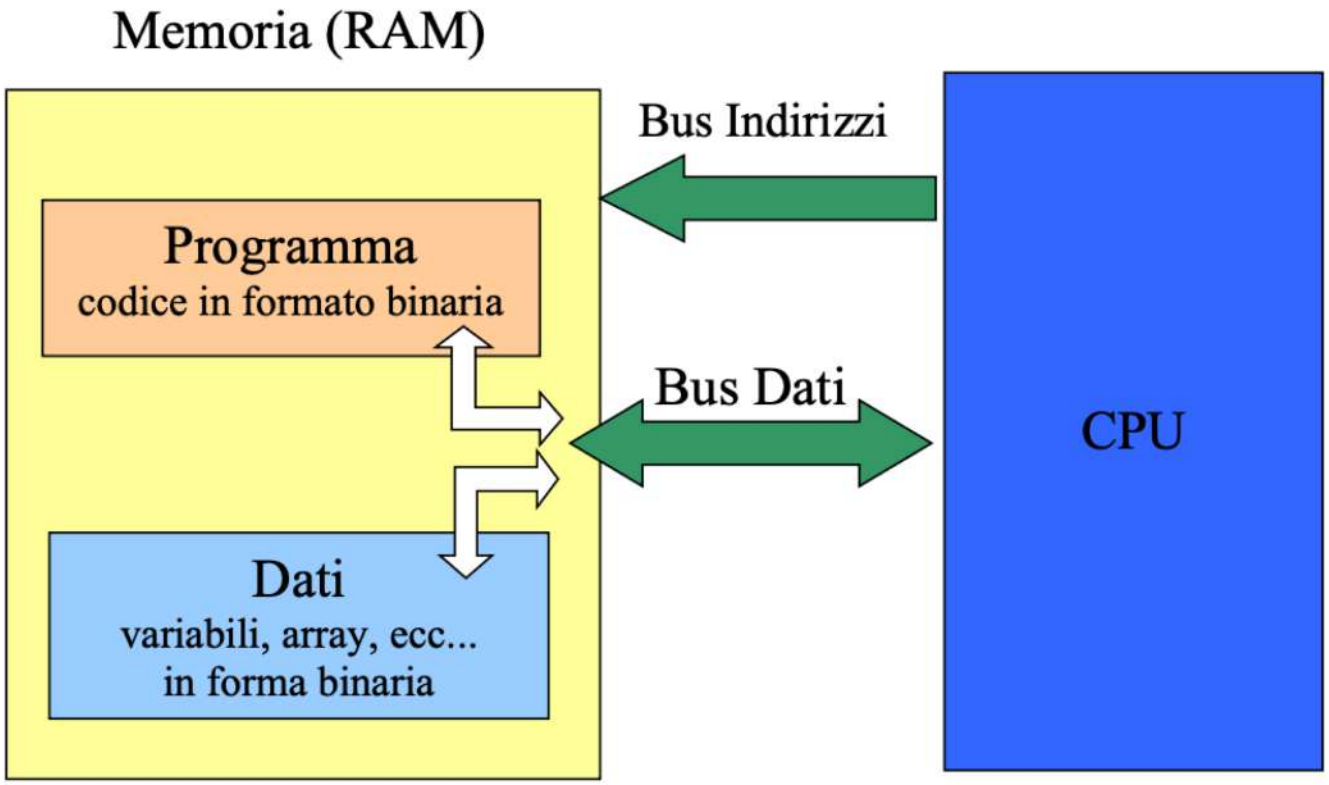
- Questa organizzazione del calcolatore digitale è detta "*bus oriented*".
- I componenti sono connessi fra loro mediante un **bus,** cioè un insieme di cavi paralleli sui quali vengono trasmessi indirizzi, dati e segnali di controllo.
- I bus possono essere esterni alla CPU, per connetterla alla memoria e ai dispositivi di I/O, oppure interni alla CPU stessa.

Architettura di Von Neumann



- Nell'architettura di Von Neumann, **la memoria viene usata non solo per i dati ma anche per i programmi**.
- Grazie a questa architettura, i programmi (su schede perforate) potevano essere caricati in memoria con un lettore di schede evitando complesse configurazioni/programmazioni con interruttori e cavi.

Architettura di Von Neumann



- **Programmi e Dati** al tempo di esecuzione sono caricati in memoria (codificati in forma binaria).
- Programmi e dati sono **trasferiti entrambi attraverso il bus dati**. Il Bus indirizzi è utilizzato dalla CPU per indicare alla memoria le locazioni dove risiedono le informazioni da trasferire.

La CPU

Il compito della CPU è quello di eseguire i programmi immagazzinati nella memoria centrale leggendo le loro istruzioni ed eseguendole in sequenza.

Una CPU è composta da:

Unità di controllo: legge le istruzioni dalla memoria centrale e ne determina il tipo.

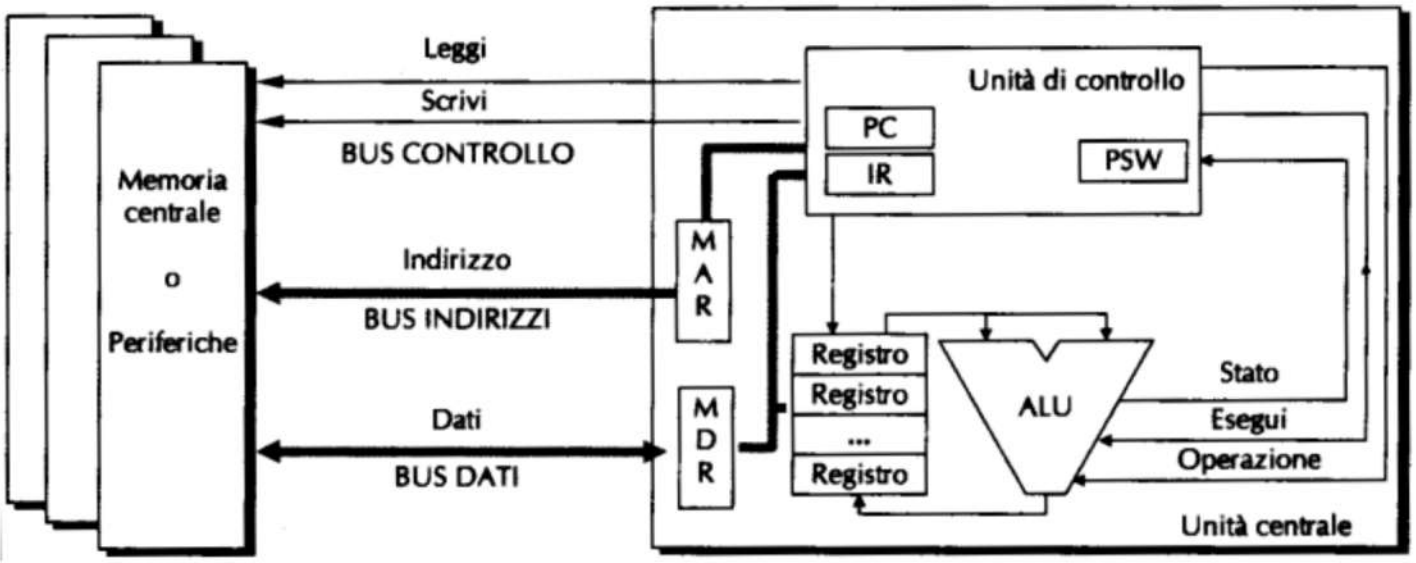
ALU: esegue le operazioni necessarie all'esecuzione delle istruzioni (AND,OR, addizione binaria).

Registri: sono una piccola memoria ad alta velocità utilizzata per memorizzare i risultati temporanei e le informazioni di controllo necessarie al funzionamento dell'ALU. I registri risiedono all'interno della CPU.

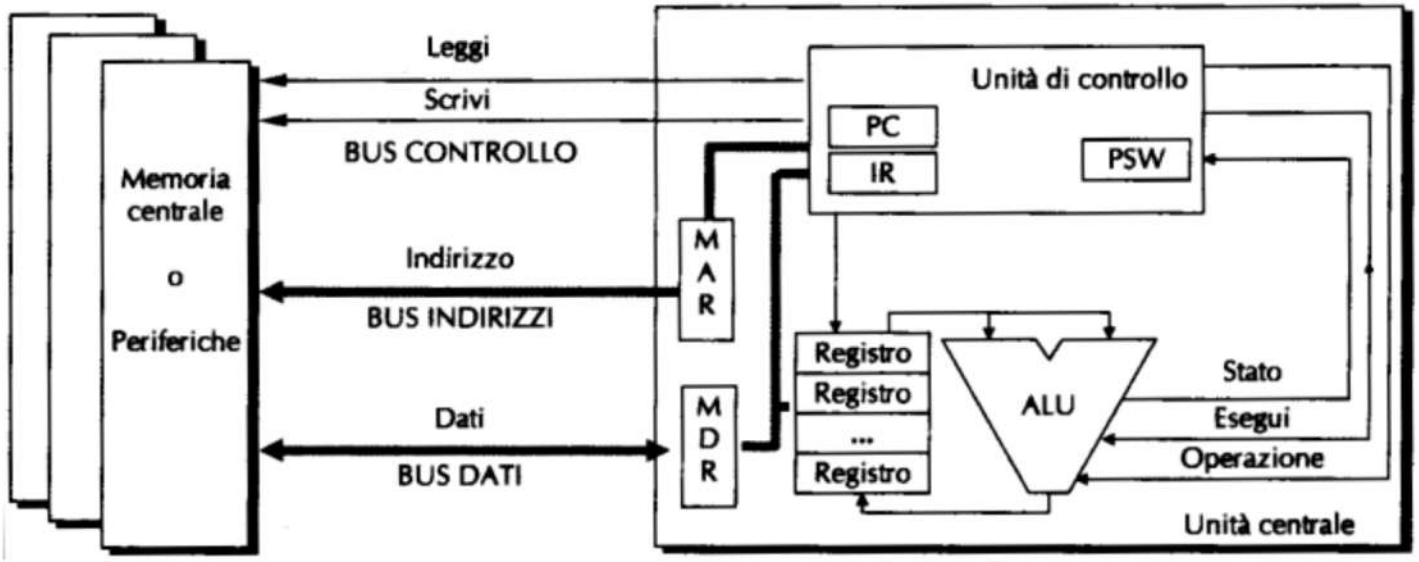
La CPU

- Normalmente i registri hanno tutti le stesse dimensioni, alcuni vengono utilizzati per compiti specifici altri sono general purpose.
- Il registro più importante è il **Program Counter** (PC) che indica la prossima istruzione da eseguire.
- L'**Instruction Register** (IR) è il registro che memorizza l'istruzione che si sta per eseguire.

All'interno della CPU i vari componenti sono a loro volta collegati tramite diversi bus.

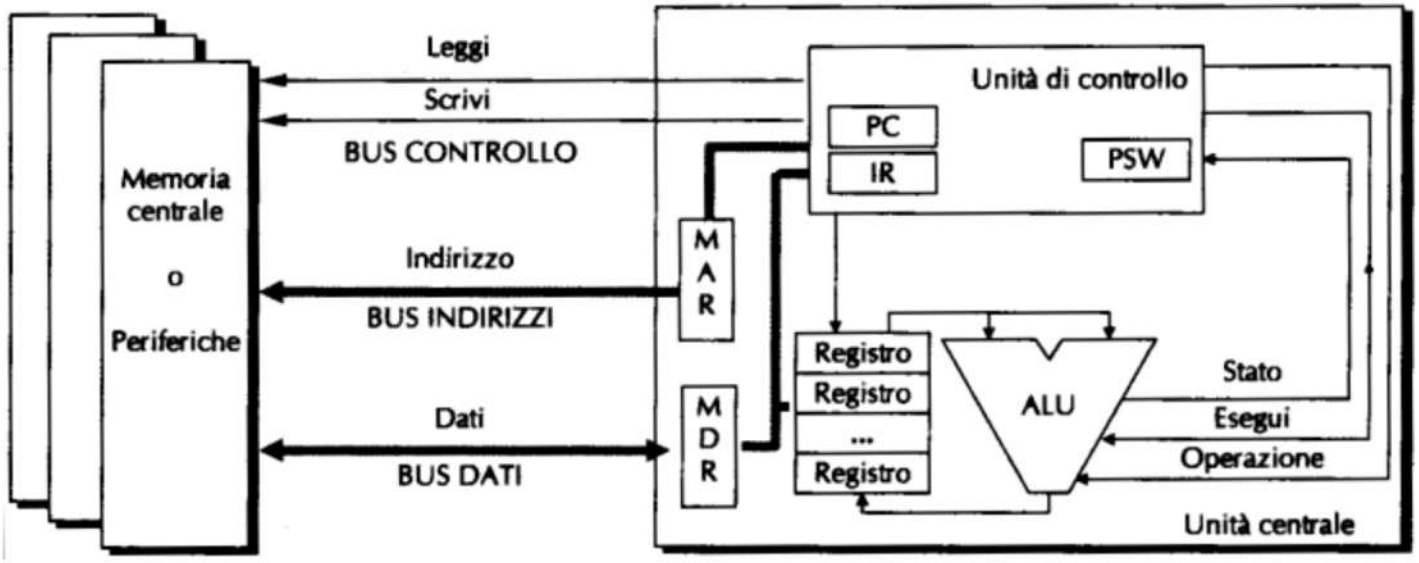


Esecuzione di una istruzione



1) la CPU mette il valore di PC (indirizzo della prossima istruzione da leggere dalla memoria) su MAR e attiva la linea Leggi;
2) la memoria attraverso il bus indirizzi accede a MAR e, una volta reperito quanto richiesto, lo scrive su MDR attraverso il bus dati;
3) la CPU copia su IR il valore di MDR e decodifica l'istruzione;
4) l'istruzione passa in esecuzione sulla ALU;

Esecuzione di una istruzione



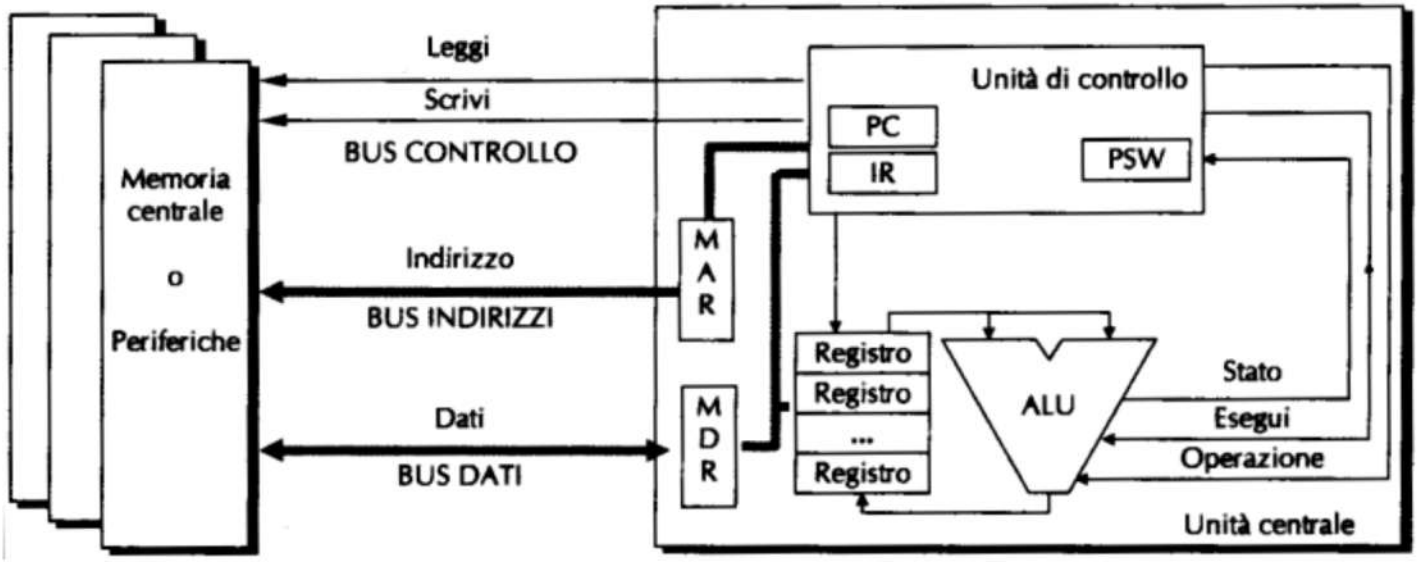
5) se l'istruzione prevede la lettura di operandi dalla memoria, questi devono essere caricati sui registri; per ciascun operando da reperire:

5.1) la CPU mette l'indirizzo dell'operando su MAR e attiva la linea Leggi;

5.2) la memoria attraverso il bus indirizzi accede a MAR e, una volta reperito quanto richiesto, lo scrive su MDR attraverso il bus dati;

5.3) la CPU copia sul registro destinazione il valore dell'operando che è in MDR;

Esecuzione di una istruzione



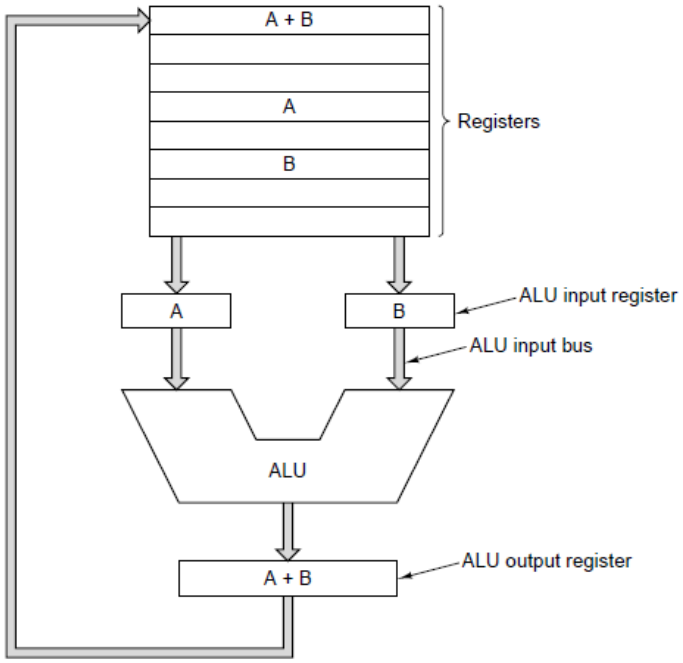
6) terminata l'esecuzione la CPU copia sul registro destinazione il valore prodotto dalla ALU; se è prevista scrittura in memoria del valore calcolato:

6.1) la CPU mette l'indirizzo della cella di destinazione su MAR e il risultato su MDR e attiva la linea Scrivi;

6.2) la memoria attraverso il bus indirizzi accede a MAR, attraverso il bus dati a MDR e, una volta reperito il valore in MDR, lo scrive sulla propria cella interna indicata da MAR;

7) Si ritorna al punto 1 dopo aver aggiornato il valore di PC (prossima istruzione da eseguire).

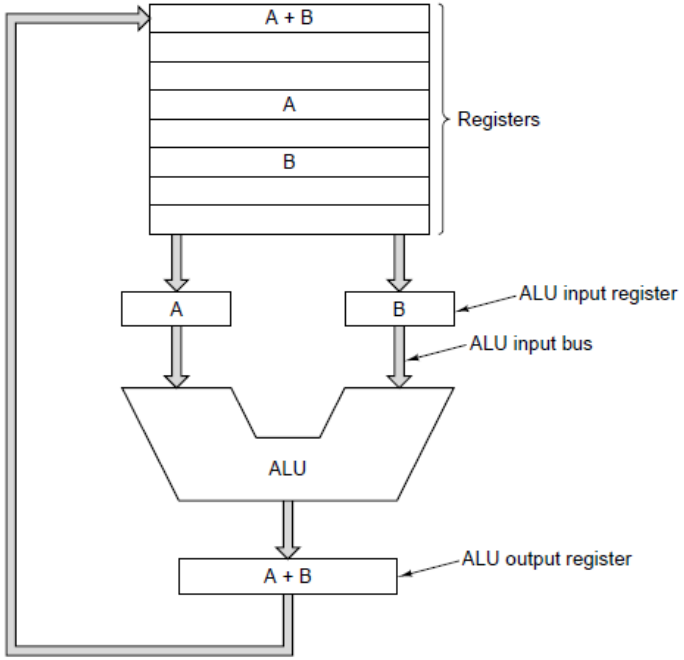
Il data path



In figura è rappresentato il classico **data path** per una CPU di Von Neumann. Il data path comprende l'ALU e i registri.

Il passaggio di due operandi attraverso la ALU e la memorizzazione del risultato in un nuovo registro viene detto **ciclo di data path**..

Il data path



Ogni istruzione ISA (assembly) viene eseguita in uno o più cicli di data-path; diversi cicli sono necessari per istruzioni complesse (es. divisione).

In architetture non parallele il ciclo di data path corrisponde al **ciclo di clock** (misurato in nanosecondi) ossia l'intervallo di tempo utilizzato per sincronizzare le diverse operazioni del processore.

La velocità con cui viene compiuto un ciclo di data-path contribuisce significativamente a determinare la velocità della CPU.

Numero di istruzioni al secondo

durata ciclo di data path = durata ciclo di clock = 1/F
(dove F è la frequenza di lavoro della CPU)

durata istruzione ISA = n × durata ciclo di data path
(*n* variabile per istruzioni diverse, ma anche per architetture diverse)

Istruzioni ISA per sec. = 1/durata istruzione ISA = F / n

Numero di istruzioni al secondo

CPU	Anno	Frequenza	Registri	Transistor	MIPS
4004	1971	0,74 MHz	4 bit	2.300 (10 μm)	0,07
8008	1972	0,5 MHz	8 bit	3.500 (10 μm)	0.05
8080	1974	2 MHz	8 bit	6.000 (6 μm)	0,29
8086	1978	8 MHz	16 bit	29.000 (3 μm)	0,66
80286	1982	12,5 MHz	16 bit	134.000 (1,5 μm)	2,66
Intel386	1985	33 MHz	32 bit	275.000 (1 μm)	10
Intel486	1989	50 MHz	32 bit	$1{,}2 \cdot 10^6$ (0,8 μm)	41
Pentium	1993	66 MHz	32 bit	$3{,}1 \cdot 10^6$ (0,8 μm)	112
Pentium Pro	1995	200 MHz	32 bit	$5{,}5 \cdot 10^6$ (0,35 μm)	541
Pentium II	1997	300 MHz	32 bit	$7{,}5 \cdot 10^6$ (0,35 μm)	813
Pentium III	1999	600 MHz	32 bit	$9{,}5 \cdot 10^6$ (0,25 μm)	1.105
Pentium 4	2000	1.5 GHz	32 bit	$42 \cdot 10^6$ (0,18 μm)	2.262
Pentium D	2005	3.2 GHz	64 bit	$230 \cdot 10^6$ (90 nm)	7.145
Core i7 (gen. 5)	2014	3.8 GHz	64 bit	$1{,}3 \cdot 10^9$ (14 nm)	298.190
Core i9 (gen. 9)	2018	4.7 GHz	64 bit	$3 \cdot 10^9$ (14 nm)	412.090

MIPS = **M**illion **I**nstructions **P**er **S**econd (riferito alle istruzioni ISA)

Esecuzione delle istruzioni

La CPU opera in modo ciclico, ripetendo le seguenti operazioni fino al termine dell'esecuzione del programma:

- **Caricamento** (**Fetch**): acquisizione dalla memoria di un'istruzione del programma.
- **Decodifica** (**Decode**): identificazione del tipo di operazione da eseguire.
- **Esecuzione** (**Execute**): effettuazione delle operazioni corrispondenti all'istruzione.

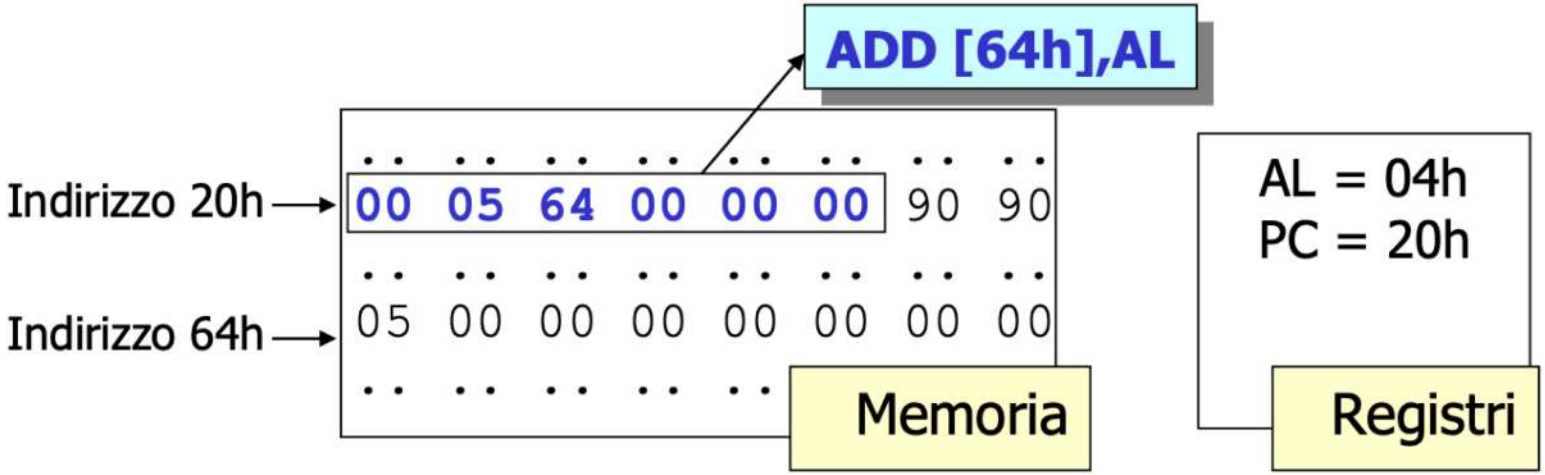
Esecuzione delle istruzioni

Più in dettaglio, l'esecuzione di ogni istruzione da parte della CPU richiede una serie di passi che, in linea di massima, possono essere così riassunti:

1) Leggi l'istruzione seguente dalla memoria e mettila nel IR
2) Incrementa il PC per indicare l'istruzione seguente
3) Decodifica l'istruzione appena letta
4) Se l'istruzione utilizza degli operandi (parole) determina dove si trovano (memoria/registri)
5) Se necessario metti gli operandi in registri della CPU
6) Esegui l'istruzione
7) Salva il risultato in un registro
8) Torna al punto 1

A sua volta il punto (6) può richiedere un insieme di sotto-passi a seconda della complessità dell'Instruction Set. La traduzione di una istruzione nei suoi passi elementari è effettuata da un **interprete** (microprogramma).

Esempio: somma di due elementi



Fetch:
viene letta in IR la sequenza 00 05 64 00 00 00 dalla memoria all'indirizzo correntemente puntato da PC (20h); PC viene incrementato.

Decode:
la sequenza 00 05 64 00 00 00 viene decodificata in ADD [64h], AL

Execute:
viene recuperato dalla memoria il contenuto all'indirizzo 64h; questo viene sommato con il valore di AL e nuovamente scritto all'indirizzo 64h. L'operazione richiede sicuramente più cicli di data-path, se non altro per la necessità di recuperare operandi dalla memoria.

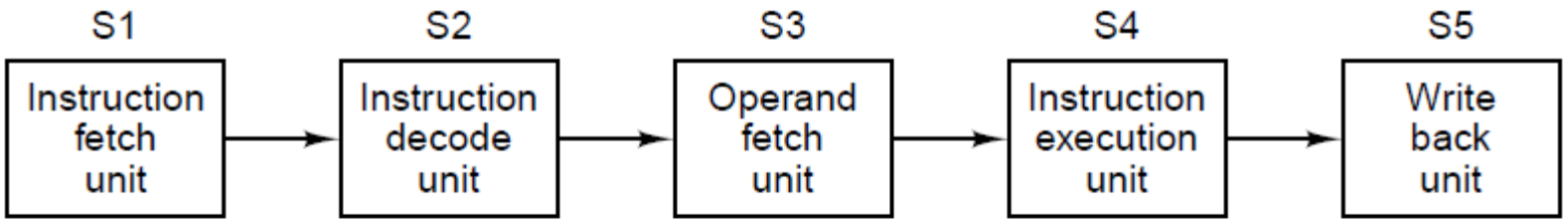
Prefetching

- Uno dei maggiori colli di bottiglia nella velocità di esecuzione delle istruzioni è rappresentato dal prelievo delle istruzioni dalla memoria.
- Una prima soluzione a questo problema fu la tecnica del **prefetching**: prelevare in anticipo le istruzioni dalla memoria, in modo da averle già a disposizione nel momento in cui dovessero rendersi necessarie.
- Le istruzioni venivano memorizzate in un insieme di registri chiamati **buffer di prefetch,** dai quali potevano essere prese nel momento in cui venivano richieste, senza dover attendere che si completasse una lettura della memoria.
- In pratica la tecnica di prefetching divide l'esecuzione dell'istruzione in due parti: il prelievo dell'istruzione e la sua esecuzione effettiva.

Pipelining

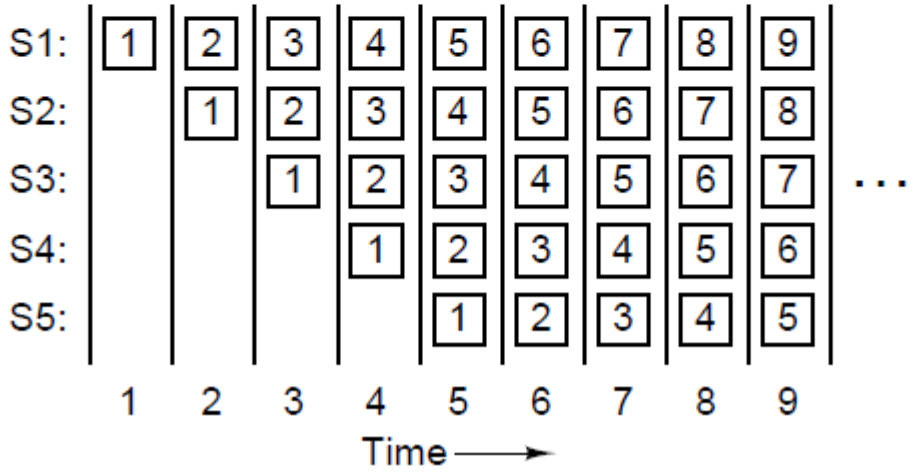
- Il concetto di **pipeline** è una evoluzione della strategia del prefetching: invece di dividere l'esecuzione di un'istruzione solamente in due fasi, la si divide in un numero maggiore di parti che possono essere eseguite in parallelo da componenti hardware dedicati.

- Esempio di pipeline a 5 stadi:



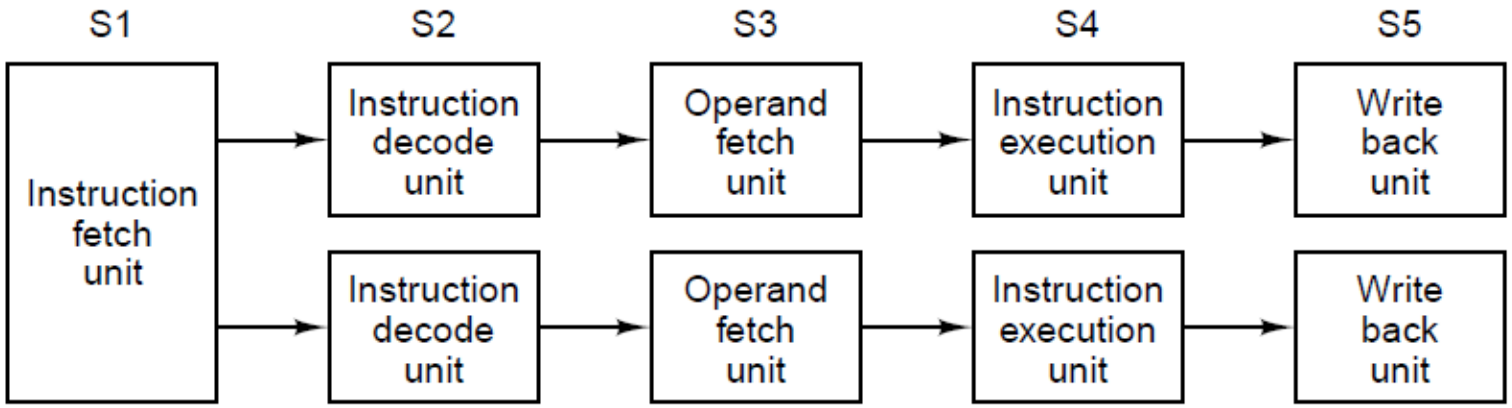
Pipelining

- **Clock 1**: lo stadio S1 sta lavorando sull'istruzione 1, prelevandola dalla memoria.
- **Clock 2**: S2 decodifica l'istruzione 1, mentre S1 preleva l'istruzione 2.
- **Clock 3**: S3 preleva gli operandi per l'istruzione 1, S2 decodifica l'istruzione 2 e S1 preleva l'istruzione 3.
- **Clock 4**: S4 esegue l'istruzione 1, S3 preleva gli operandi per l'istruzione 2, S2 decodifica l'istruzione 3 e S1 preleva l'istruzione 4.
- **Clock 5**: S5 scrive il risultato dell'istruzione 1, mentre gli altri componenti lavorano sulle istruzioni successive.



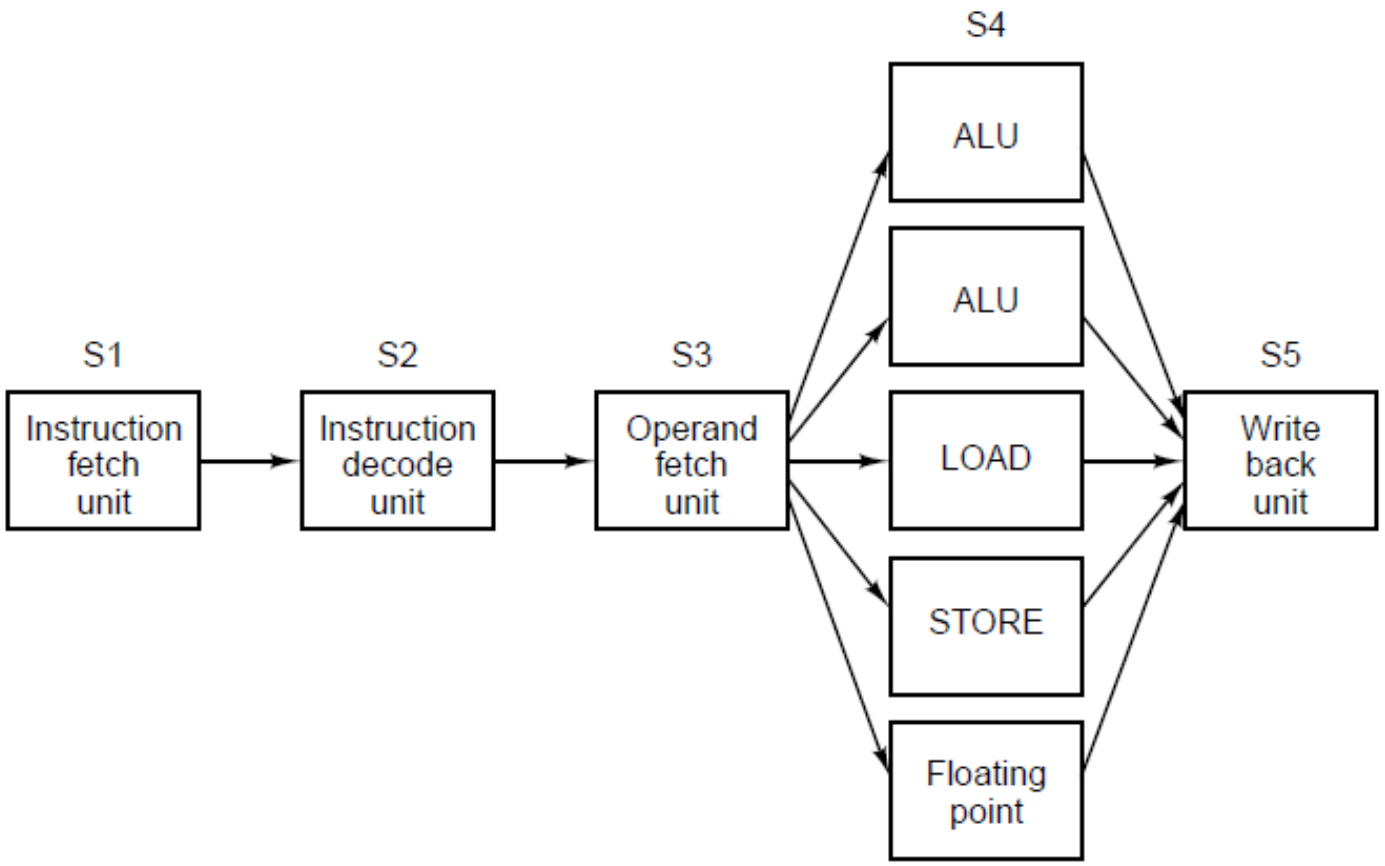
Doppia pipeline

- Nell'esempio seguente, una singola unità di *fetch* preleva due istruzioni alla volta e le inserisce in due pipeline, ognuna delle quali è dotata di una ALU.
- Affinché le due istruzioni possano essere eseguite in parallelo, non devono però esserci conflitti nell'uso delle risorse (i registri) e nessuna delle due istruzioni deve dipendere dal risultato dell'altra.
- Il compilatore deve occuparsi di gestire correttamente questa situazione, in quanto l'hardware non effettua alcun controllo e se le istruzioni sono incompatibili restituisce un risultato errato.

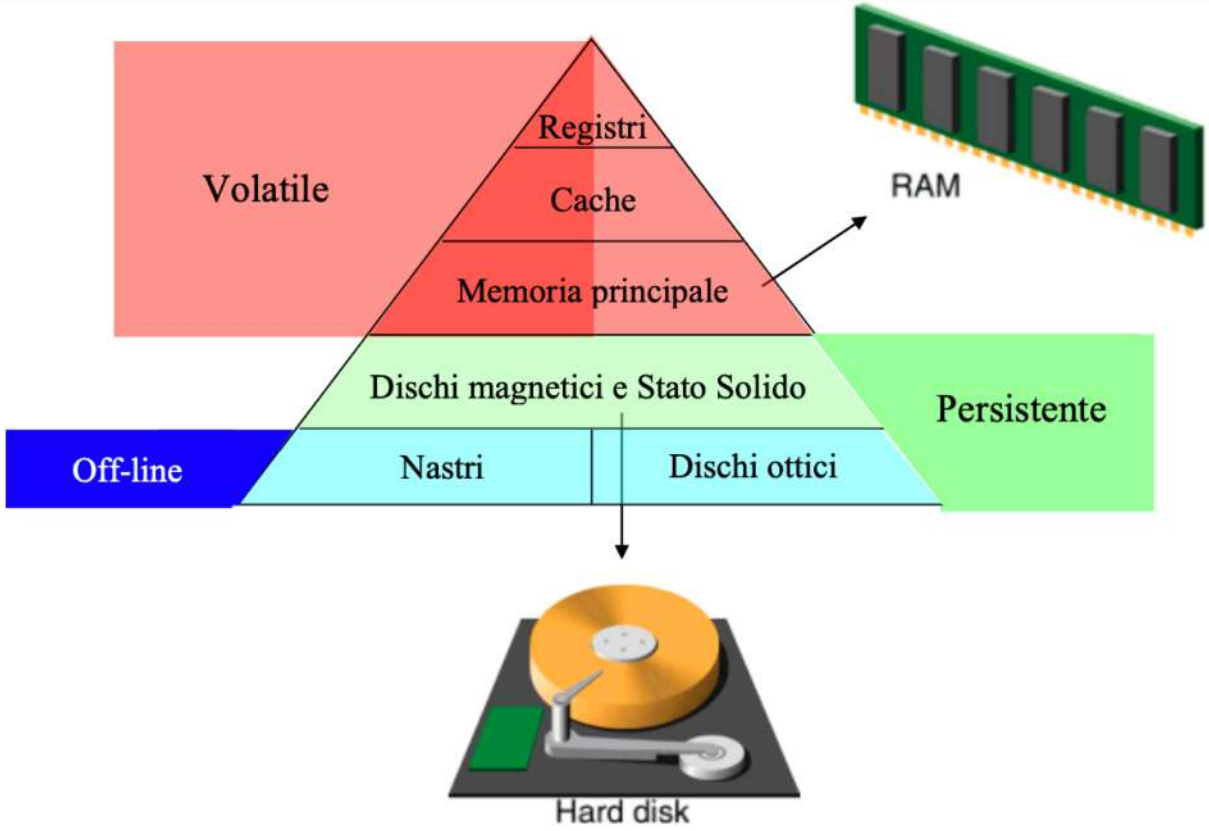


Architetture superscalari

- L'idea alla base delle architetture superscalari consiste nell'avere una singola pipeline, ma di associarle più unità funzionali:

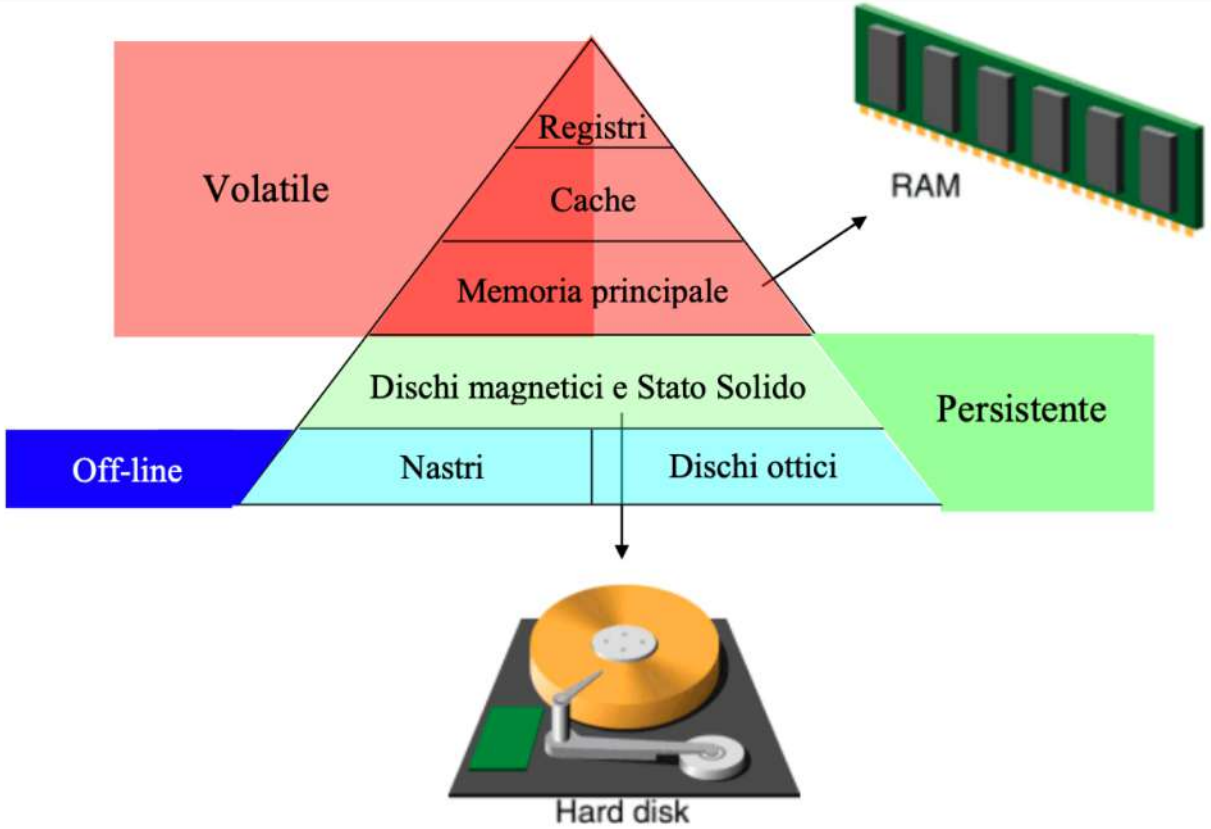


Le memorie



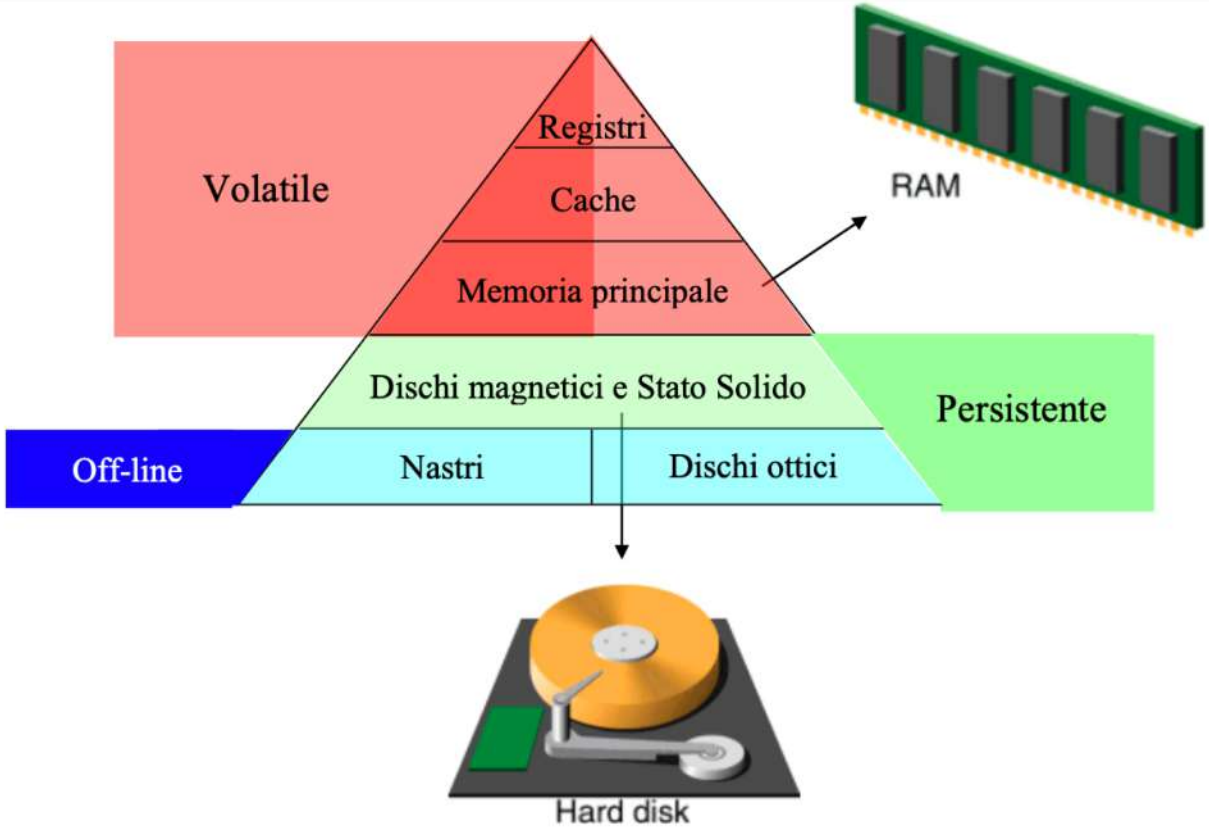
Le **memorie** sono le componenti del calcolatore in grado di memorizzare le informazioni: dati, programmi e risultati indispensabili per il suo funzionamento.

Le memorie



- Volatile: l'informazione rimane memorizzata fino a che il calcolatore è alimentato
- Persistente: l'informazione rimane memorizzata anche quando il calcolatore non è alimentato (spento)
- On-line: i dati sono sempre accessibili
- Off-line: il supporto deve essere montato per poter accedere ai dati

Le memorie



Il costo di memorizzazione per byte cresce salendo la piramide
La dimensione delle memorie cresce scendendo la piramide

Errore nei dati

La memorizzazione possono occasionalmente commettere errori a causa, ad esempio, di picchi di tensione elettrica (o difetti). Questi errori possono essere prevenuti utilizzando dei codici di **correzione degli errori**.

Distanza di Hamming: indica il numero di bit corrispondenti che differiscono in due parole.

D. Hamming=2

10011100
11010100

D. Hamming=4

11111110
01100111

Se due parole di codice hanno distanza di Hamming H, saranno necessari H errori (cambiamenti di stato di 1 bit) per convertire una nell'altra.

Identificazione degli errori nei dati

Bit di parità: semplice tecnica che non permette di correggere alcun errore, ma permette di identificare errori di un bit.

Funzionamento:

- Ad ogni parola viene aggiunto un bit di controllo
- Il bit di controllo vale 1 se il numero di bit a 1 della parola è dispari
- Il bit di controllo vale 0 se il numero di bit a 1 della parola è pari

10001110	100011100
10001010	100010101

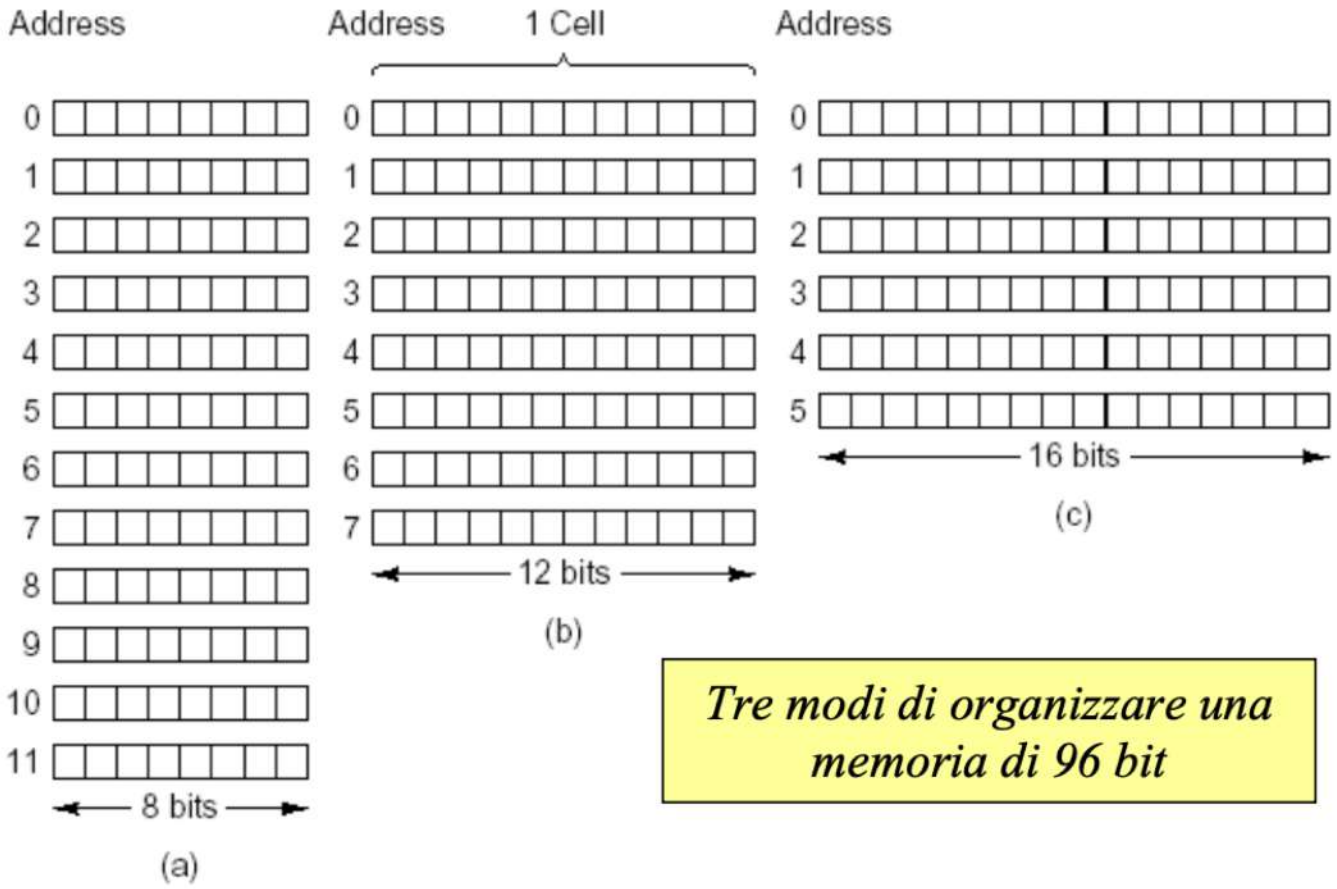
Organizzazione della memoria

La memoria principale è quella parte del calcolatore preposta a immagazzinare i programmi in esecuzione e i relativi dati. La memoria principale è volatile ossia mantiene le informazioni sino a quando il calcolatore è alimentato.

Le memorie si compongono di un numero di celle (o locazioni) ognuna delle quali è in grado di memorizzare una parte delle informazioni. Ogni cella è associata a un numero (**indirizzo**) che la identifica univocamente.

Tutte le celle di una memoria mantengono lo stesso numero di bit. Il numero di bit associato a un indirizzo è detto **parola**. La dimensione minima per una parola è il byte (8 bit) anche se normalmente i calcolatori moderni utilizzano parole più lunghe (32-64 bit).
La dimensione della parola determina anche le dimensioni dei registri e delle istruzioni.

Organizzazione della memoria

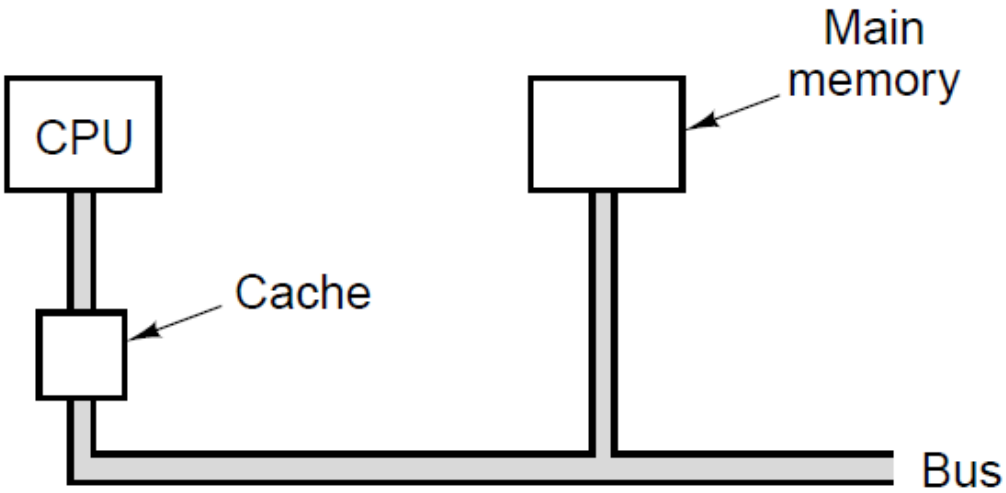


Tre modi di organizzare una memoria di 96 bit

Gli indirizzi di memoria sono espressi tramite numeri binari: se un indirizzo ha m bit il numero massimo di celle indirizzabili sarà 2^m.

Memoria cache

- La CPU è più veloce della memoria centrale, per cui quando la CPU invia una richiesta alla memoria, essa non otterrà la parola desiderata se non dopo molti cicli di CPU.
- Una soluzione per mitigare questo problema è rappresentata dalla **memoria cache**, la quale mantiene le parole di memoria usate più di frequente.
- Da un punto di vista logico la cache si trova tra la CPU e la memoria centrale (fisicamente può essere collocata in varie posizioni):



Memoria cache

- Quando la CPU necessita di una parola, la cerca nella cache e, solo nel caso in cui essa non sia presente, la richiede alla memoria centrale.
- E' possibile ridurre drasticamente il tempo medio di accesso se una frazione significativa delle parole è presente nella cache.
- Si può osservare che i programmi non accedono alle loro memorie in modo completamente casuale.
- **Principio di località**: se ad un certo istante la memoria fa un riferimento all'indirizzo *A* è molto probabile che il successivo riferimento alla memoria si troverà nelle vicinanze di *A.*
- Quando una parola viene referenziata, la parola stessa e alcune parole vicine sono portate dalla memoria centrale (più grande e lenta) all'interno della cache (più piccola e veloce), in modo che sia possibile accedervi velocemente in un secondo momento.

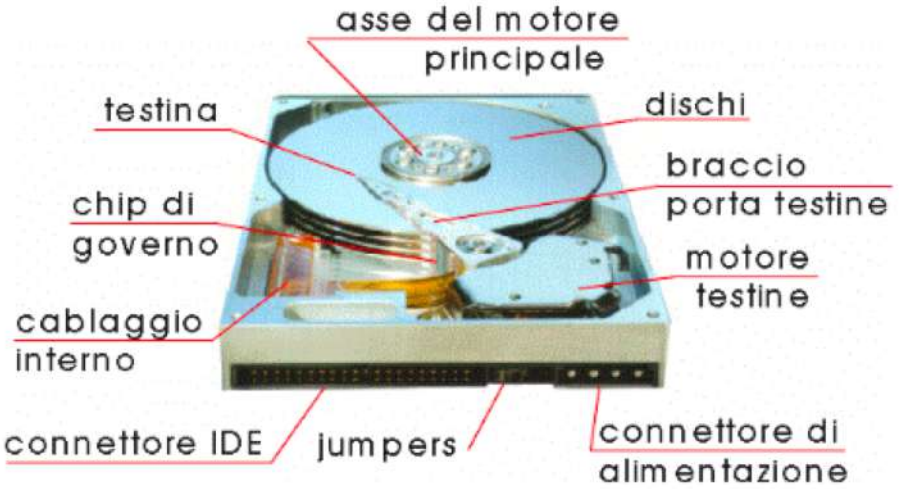
Assemblaggio della memoria

SIMM (**S**ingle **I**nline **M**emory **M**odule): i contatti dorati (in genere 72) si trovano solo su un lato della scheda e trasferiscono 32bit per ciclo di clock. Bus dati a 32 bit. Sono oramai obsolete.

DIMM (**D**ual **I**nline **M**emory **M**odule): i contatti dorati (da 168 a 288) si trovano su entrambi i lati della scheda. Bus dati a 64 bit: uso obbligato a partire dai processori Intel Pentium (bus dati 64 bit). La velocità dipende dal tipo di memoria (DDR, DDR2, DDR3, DDR4, DDR5). I moduli tipici per DDR4 sono da 8-16 GB.

Dischi magnetici

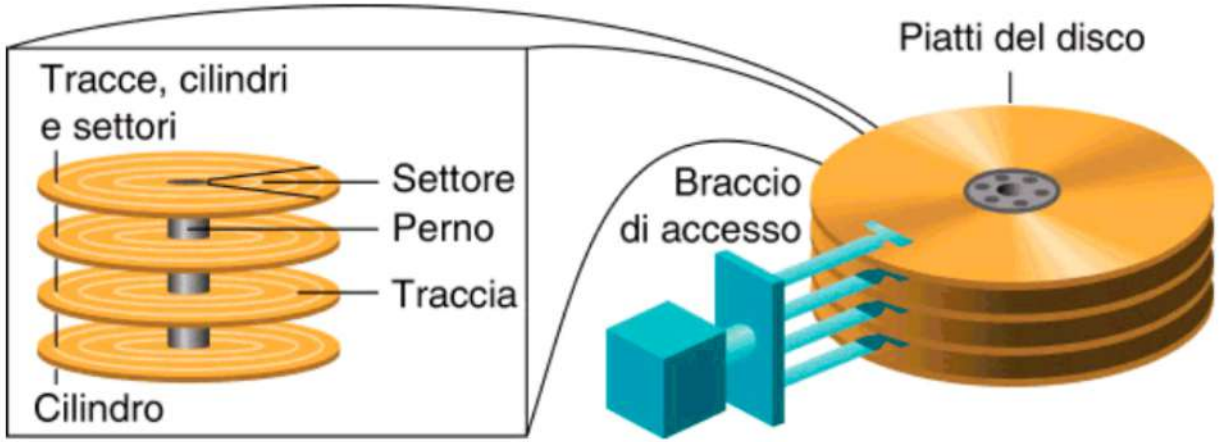
Un **hard disk** (HD) è un dispositivo elettro-meccanico per la conservazione di informazioni sotto forma magnetica, su supporto rotante a forma di piatto su cui agiscono delle testine di lettura/scrittura.

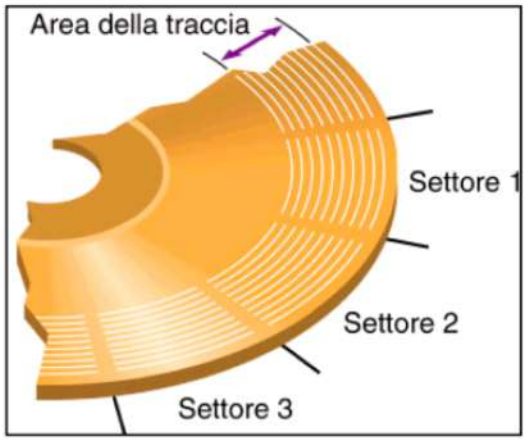


La **testina** di un disco, contenente un induttore, è sospesa sopra la superficie

- **Scrittura:** quando la corrente negativa o positiva passa attraverso la testina, viene magnetizzata la superficie appena sotto la testina e le particelle magnetiche si allineano verso sinistra o destra a seconda della polarizzazione della corrente.
- **Lettura:** quando una testina passa sopra un'area magnetizzata viene indotta una corrente positiva o negativa nella testina e ciò permette di rileggere i bit memorizzati precedentemente.

Dischi magnetici





Traccia: sequenza circolare di bit
Settore: porzione di traccia di dimensione fissa.
Ogni settore è composto da:

- Preambolo: necessario alla testina per sincronizzarsi prima della lettura/scrittura.
- Dati: l'insieme dei byte memorizzati nel settore, normalmente 512.
- Codice correzione errori

Cilindro: insieme delle tracce in una data posizione radiale

Dischi magnetici

Prima di poter leggere o scrivere la suddetta struttura (tracce, settori) deve essere creata. Questa operazione viene detta **formattazione**.

La capacità di un hard disk è in gran parte definita dalla densità di registrazione:

Densità lineare: è limitata dalla difficoltà di individuare le variazioni del campo magnetico sulla superficie del disco.

Densità di area: il primo hard disk IBM del 1956 aveva una densità d'area di circa 2 Kbits/in^2. Nel 2015 si sono superati 1.3 Tbits/in^2.

Prestazioni dei dischi magnetici

Seek time: è il tempo necessario per spostare le testine sul cilindro desiderato e può essere a sua volta suddiviso in tre fasi (speed up, coast e slow down). I costruttori forniscono in genere:

- Average Seek: 8-10 ms (di solito si riferisce a letture)
- Track-to-Track: 1 ms
- Full-stroke: 15-20 ms (dalla traccia più interna alla traccia più esterna)

Latency time: rappresenta il tempo necessario affinché il settore interessato all'operazione passi sotto la testina. Questo fattore è definito dalla velocità di rotazione del disco che varia da 3600 a 15.000 RPM (rotazioni per minuto)

RPM	Caso peggiore ms (un giro intero)	Caso medio ms (½ giro)
3.600	16.7	8.3
4.200	14.2	7.1
5.200	11.5	5.8
7.200	8.3	4.2
10.000	6.0	3.0
15.000	4.0	2.0

Prestazioni dei dischi magnetici

Con i tempi di trasferimento tipici della tecnologia attuale (100..300 MB/sec nel 2020) un settore di 512 byte richiede pochi μsec. Il tempo di seek e di latency caratterizzano quindi fortemente le prestazioni dell'hard disk.

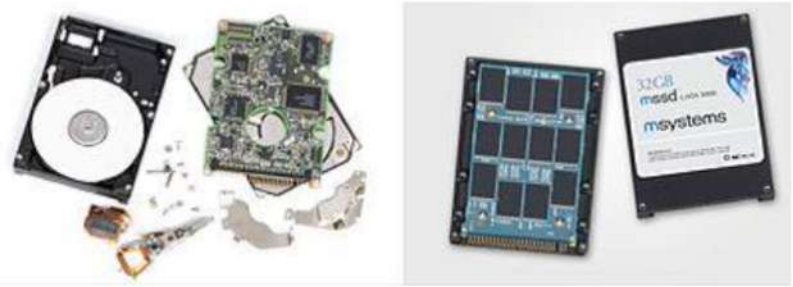
I tempi si riducono drasticamente leggendo/scrivendo più settori contigui.

Mentre la capacità dei dischi continua ad aumentare costantemente (nel 2022 sono disponibili dischi da 26 Terabyte), il tempo impiegato per la lettura di un intero disco tende a peggiorare. In generale la crescita del settore è molto più lenta rispetto a quella delle CPU.

Dischi SSD

Si tratta di dispositivi completamente elettronici, normalmente basati su memorie flash (NAND), e senza nessuna parte in movimento.

La comune denominazione disco allo stato solido è pertanto inadeguata.



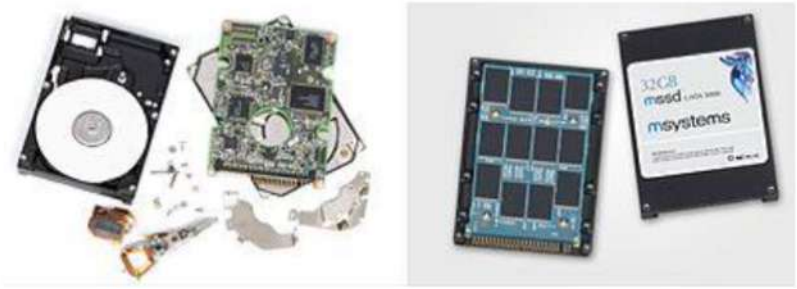
Pro (al 2023):

- tempo di accesso (seek) ridotto: in genere inferiore a 100 microsecondi, ovvero 30-100 volte inferiore ai 3..10 millisecondi di HD magnetici;
- maggiore velocità trasferimento dati: rispetto ai 100-500 MB/sec di un HD magnetico → da 500 MB/sec a oltre 7 GB/sec;
- minore possibilità di rottura e maggiore durata: le unità a stato solido hanno mediamente un tasso di rottura inferiore a quelli degli hard disk;
- rumorosità assente e minore produzione di calore;
- minori consumi durante le operazioni di lettura e scrittura (meno di ½);
- maggiore resistenza agli urti.
- maggiore capacità: disponibili SSD da 100TB, contro i ~20TB degli HD

Dischi SSD

Si tratta di dispositivi completamente elettronici, normalmente basati su memorie flash (NAND), e senza nessuna parte in movimento.

La comune denominazione disco allo stato solido è pertanto inadeguata.



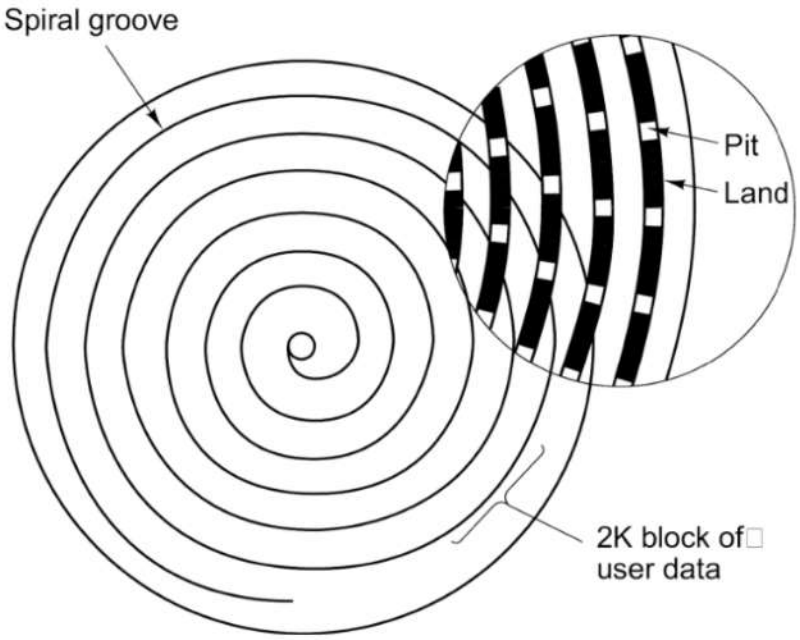
Contro (al 2023):

- un maggiore prezzo per bit: pari a circa quattro volte il costo di un disco rigido tradizionale (€60 per TB contro €15 per TB);
- una possibile minore durata dell'unità se usata per frequenti scritture: il numero massimo di riscritture dello stesso bit può andare da 3.000 a 100.000 cicli, a seconda del tipo di flash; i costruttori garantiscono, per ogni modello, un certo numero di TBW (Total Bytes Written), ad esempio 2400TB per un SSD da 4TB.

Dischi ottici

I **CD** (**C**ompact **D**isc) utilizzano un principio ottico, invece che magnetico, per la memorizzazione persistente di informazioni.

- Sono nati negli anni '80 (Philips e Sony) come supporto per la memorizzazione di musica.
- Le informazioni sono codificate per mezzo di fori (**Pit**) di 0,8 micron di diametro alternati con zone piane (**Land**) lungo un'unica spirale. Un passaggio Pit-Land o Land-Pit codifica un 1. L'assenza di variazioni codifica lo 0.



Dischi ottici

- Le informazioni sono lette tramite un raggio laser che viene riflesso diversamente al passaggio su pit e land.
- I CD vengono prodotti utilizzando uno stampo (ottenuto tramite erosione laser) su cui viene iniettata resina liquida che preserva gli incavi corrispondenti ai pit.
- Lungo la spirale i dati sono memorizzati con la stessa densità, quindi il CD ruota con velocità angolare non costante (da 530 a 200 giri/sec) per mantenere la medesima velocità lineare (120 cm/sec) nelle diverse aree del CD.

Dischi ottici

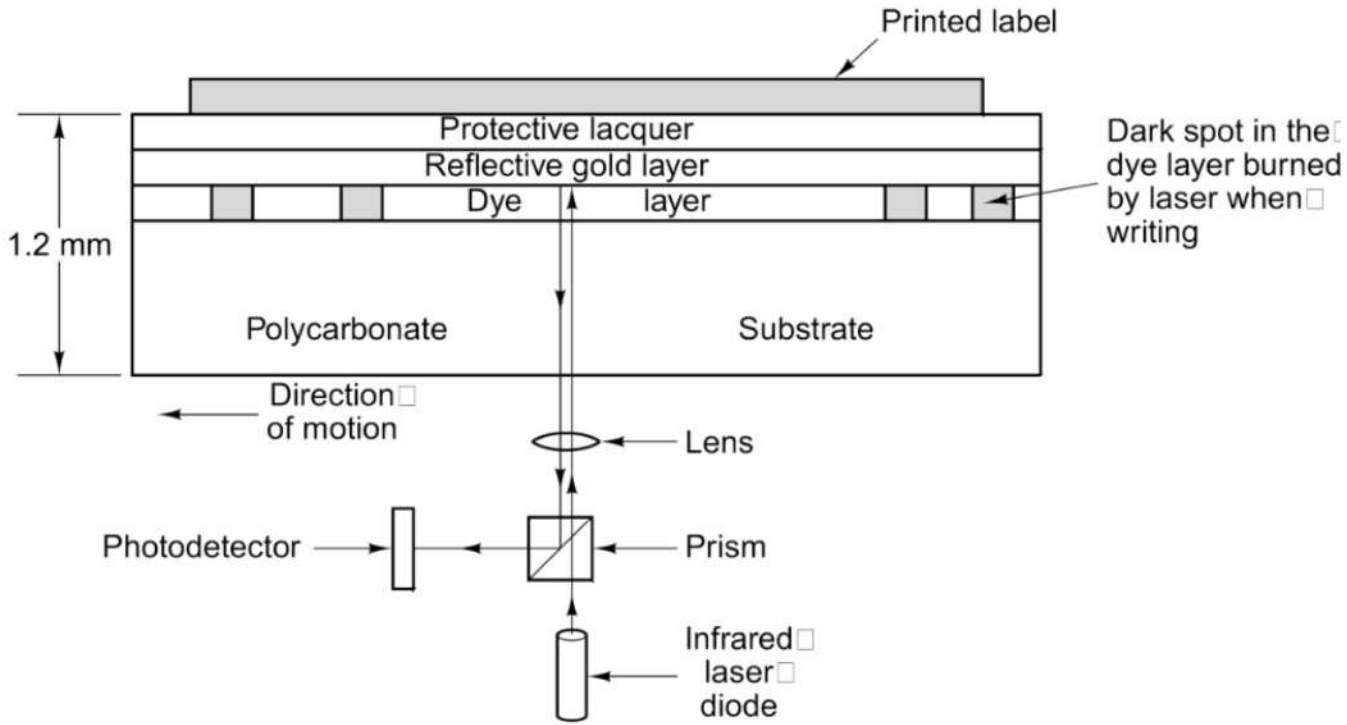
I **CD-ROM** (**C**ompact **D**isc-**R**ead **O**nly **M**emory) utilizzano la tecnologia dei CD per memorizzare dati informatici.

- Al fine di evitare la perdita di bit durante la lettura ogni byte viene codificato da un **simbolo** di 14 bit. Nei bit in eccesso viene inserito un codice per la correzione dell'errore.
- Un gruppo di 42 simboli viene denominato **frame**. Ogni frame contiene 192 bit di dati e 396 bit di correzione errore e controllo.
- Un gruppo di 98 frame viene denominato **settore di CD-ROM**.
- Ogni settore di CD-ROM inizia con un preambolo di 16 byte che permette di identificare il settore e la modalità di registrazione dei dati.
- Per motivi di compatibilità tra le diverse piattaforme i CD-ROM utilizzano un proprio file system (standard di organizzazione dei file all'interno del CD) denominato High Sierra.

Dischi ottici

I **CD-R** (**C**ompact **D**isc-**R**ecordables), nati a metà degli anni '90 svolgono le stesse funzioni dei CD-ROM ma sono registrabili dagli utenti senza l'utilizzo dello stampo.

- Diversamente dai CD la riflettività di pit e land viene ottenuta "bruciando" tramite un raggio laser uno strato di materiale colorato inserito tra il policarbonato e lo strato riflettente.



Dischi ottici

I **DVD** (**D**igital **V**ideo **D**isk o **D**igital **V**ersatile **D**isk) nascono alla fine degli anni '90 come supporto per la memorizzazione di video digitali.

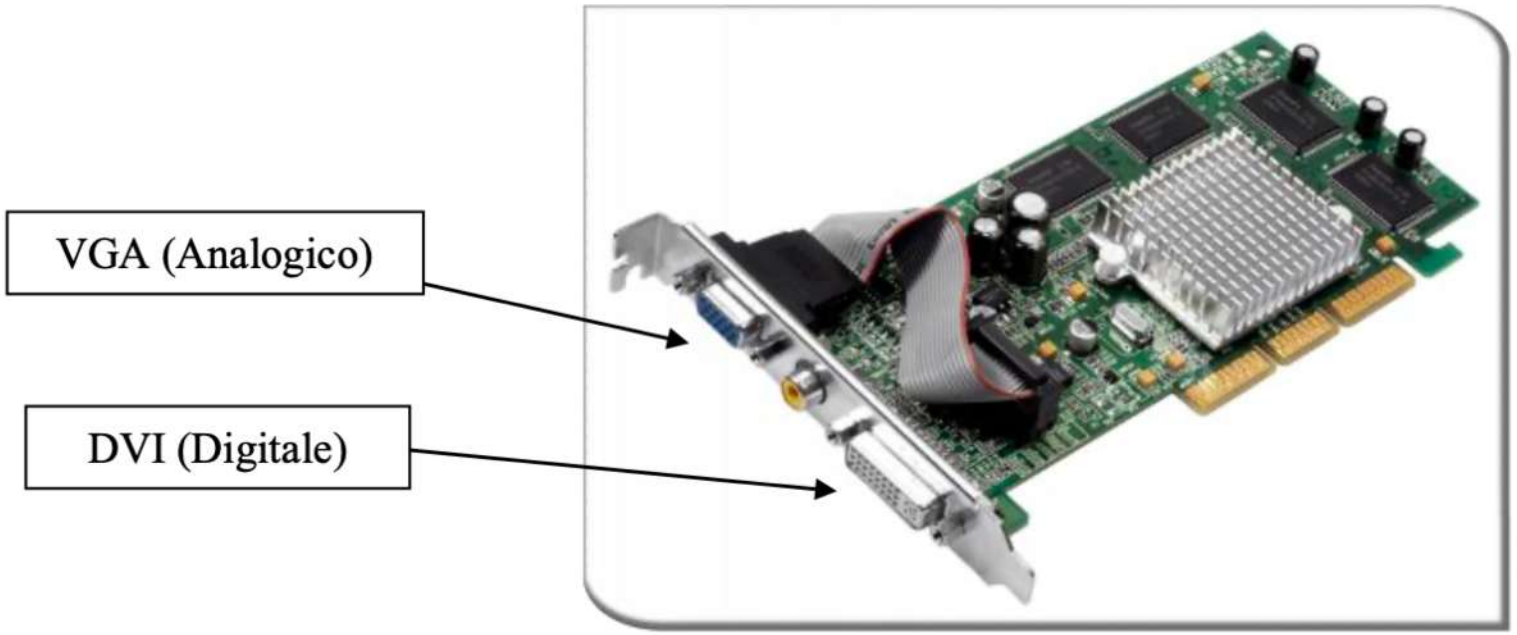
- I DVD utilizzando lo stesso progetto dei CD-ROM inserendovi alcune innovazioni:
  - Pit più piccoli (0,4 micron invece di 0,8)
  - Spirale più serrata (0,74 micron tra ogni traccia invece di 1,6)
  - Raggio laser rosso (0,65 micron invece di 0,78)
- Permettono di memorizzare fino a 4,7 GB (133 minuti di video digitale in formato MPEG-2)

Poiché lo spazio a disposizione non è mai sufficiente (e per altri motivi legati al business del mercato cinematografico) sono stati introdotti 4 formati:

Formato	Capacità (GB)
Lato unico-Strato unico	4,7
Lato unico-Strato doppio	8,5
Lato doppio-Strato unico	9,5
Lato doppio-Strato doppio	17

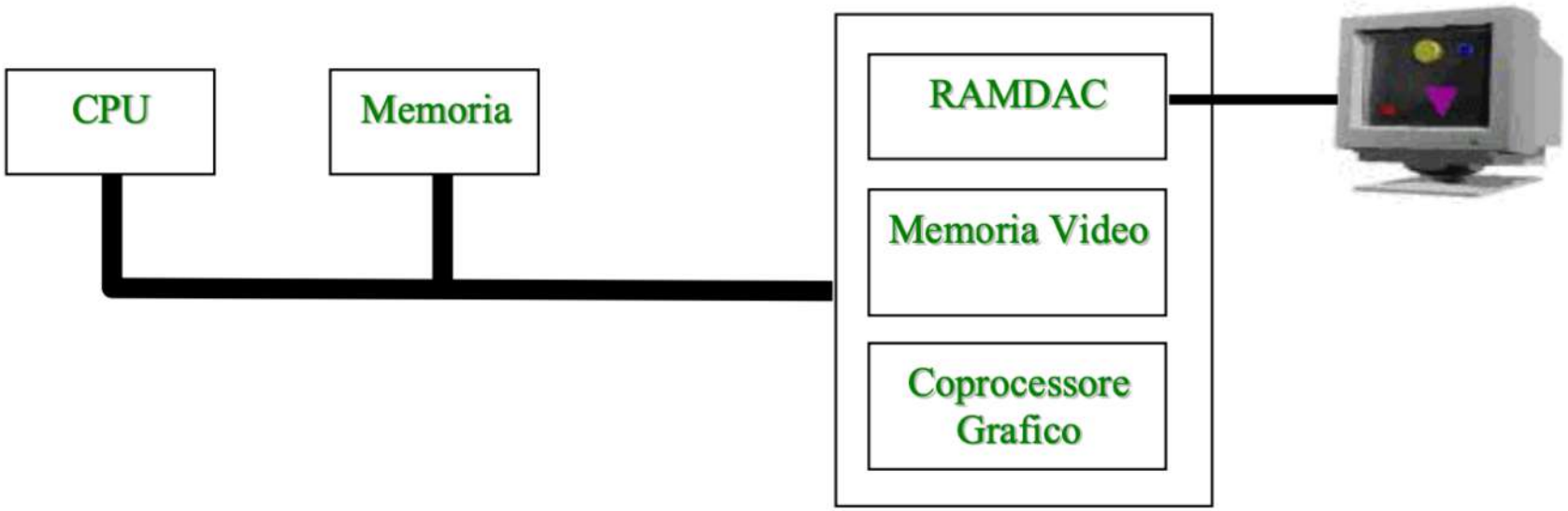
Schede grafiche

Il compito della **scheda grafica** (o scheda video) è quello di tradurre la rappresentazione dell'immagine prodotta dal processore in un formato visualizzabile dal monitor. Ogni volta che il video deve essere ridisegnato la scheda video legge la rappresentazione digitale dell'immagine dalla memoria e invia al monitor il segnale che permette di rappresentare il colore di ogni singolo pixel.



Schede grafiche

Quando i monitor (così come i televisori) erano basati su tubo catodico (CRT), i loro segnali di ingresso erano analogici e la scheda grafica aveva il compito di convertire la rappresentazione digitale dell'immagine da visualizzare in segnali analogici (es. uscita VGA). La traduzione vera e propria del segnale è realizzata da un componente denominato **RAMDAC** (**R**andom **A**ccess **M**emory **D**igital-**A**nalog **C**onverter).



Schede grafiche

Oggi quasi tutti i monitor sono LCD e prevedono input digitale. Pertanto il ruolo primario della scheda grafica non è più quello di tradurre il segnale in formato analogico, ma di:

- adattare l'immagine adeguando colori e risoluzione (attenzione al blurring!)
- mettere a disposizione una memoria locale da utilizzare come buffer
- supportare l'accelerazione grafica (es. 3D).

I principali formati di output sono:

- Video Graphics Array (**VGA**): standard analogico introdotto nel 1987 e progettato per monitor a tubo catodico, ma utilizzato, per compatibilità, anche da diversi monitor LCD. Problemi di rumore elettrico, distorsione dell'immagine e alcuni errori nella valutazione dei pixel.

Schede grafiche

- Digital Visual Interface (**DVI**): introdotto nei monitor LCD. Risolve i problemi di VGA facendo corrispondere a ogni pixel dell'output un pixel dello schermo, in quanto ne riconosce la risoluzione nativa. Trasporta anche (in piedini specifici) segnali analogici VGA.

- High-Definition Multimedia Interface (**HDMI**): pubblicato nel 2003, questo standard, ha come obiettivo la sostituzione degli standard precedenti e può trasportare anche dati audio.

- DisplayPort (**DP**): pubblicato nel 2006, questo standard è utilizzato principalmente per collegare una sorgente video digitale a un monitor, ma può trasportare anche audio e altri tipi di dati.

Schede grafiche

Potenza di calcolo: la porzione di tempo di CPU dedicato al calcolo dell'immagine video è aumentato proporzionalmente al crescere dell'utilizzo di interfacce grafiche da parte delle applicazioni. Al fine di limitare lo "spreco" di tempo CPU le schede video sono state dotate di processori dedicati denominati **GPU** (Graphical Processing Unit). Il processore grafico può essere utilizzato sia per la grafica 3D, sia per la grafica raster: la CPU del calcolatore non calcola la posizione e il colore di tutti i pixel da disegnare, ma invia un comando all'acceleratore indicandogli cosa deve essere disegnato. Esempio disegna *rettangolo blu centrato in 100, 150 e di dimensioni 20×20*.

Per poter sfruttare le capacità di accelerazione delle schede grafiche è necessario che le applicazioni conoscano le funzioni da esse implementate. A causa dell'elevato numero e delle differenze tra le schede grafiche è stato necessario sviluppare delle librerie standard (OpenGL, Direct3D) che permettano ai programmatori di prescindere dalla scheda grafica utilizzata.

Schede grafiche

Le GPU delle attuali schede grafiche (prodotte da ATI ed Nvidia) sono costituita da un numero **molto elevato di core** in grado di operare parallelamente (alcune migliaia). Da ciò deriva un'enorme capacità di calcolo che ha destato l'interesse dei programmatori. Attraverso linguaggi specifici (es. Cuda) possono oggi eseguire algoritmi (non grafici) sulla GPU.