



UNIVERSITÀ
DEGLI STUDI DELLA
TUSCIA

INFORMATICA

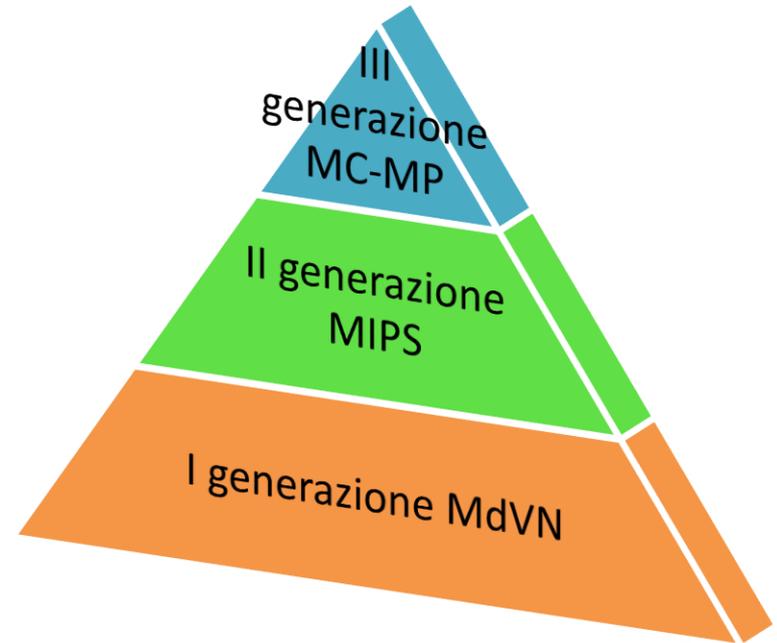
Elaboratore Elettronico

Franco Liberati
franco.liberati@unitus.it

Elaboratore Elettronico

Argomenti della lezione

- ❑ Elaboratore di prima generazione (macchine CISC)
 - ❑ Macchina di von Neumann
 - ❑ Unità di Controllo
 - ❑ Unità Logico-Aritmetica
 - ❑ Memoria Centrale
 - ❑ Unità di input Output
 - ❑ Struttura di interconnessione
- ❑ Elaboratore di seconda generazione (macchine RISC)
 - ❑ MIPS Project
- ❑ Elaboratore di terza generazione
 - ❑ Muti-processore
 - ❑ Multi-core





Elaboratori di I^a generazione

Elaboratore Elettronico

Prima generazione: Macchina di von Neumann

- ❑ Nel 1945 fu presentato un **modello di elaboratore** grazie ad un team di lavoro in cui erano presenti John von Neumann e Hermann Goldstine
- ❑ Il modello di “Macchina di von Neumann” (MvN, o “macchina di Princeton”) consentiva l’esecuzione di **programmi ad uso generico** (architettura adottata dal calcolatore EDSAC del 1949)
- ❑ Tale modello, perfezionato nei singoli componenti, è stato un punto di riferimento per la progettazione di un qualsiasi elaboratore elettronico fino agli anni 80 e 90 del XX secolo

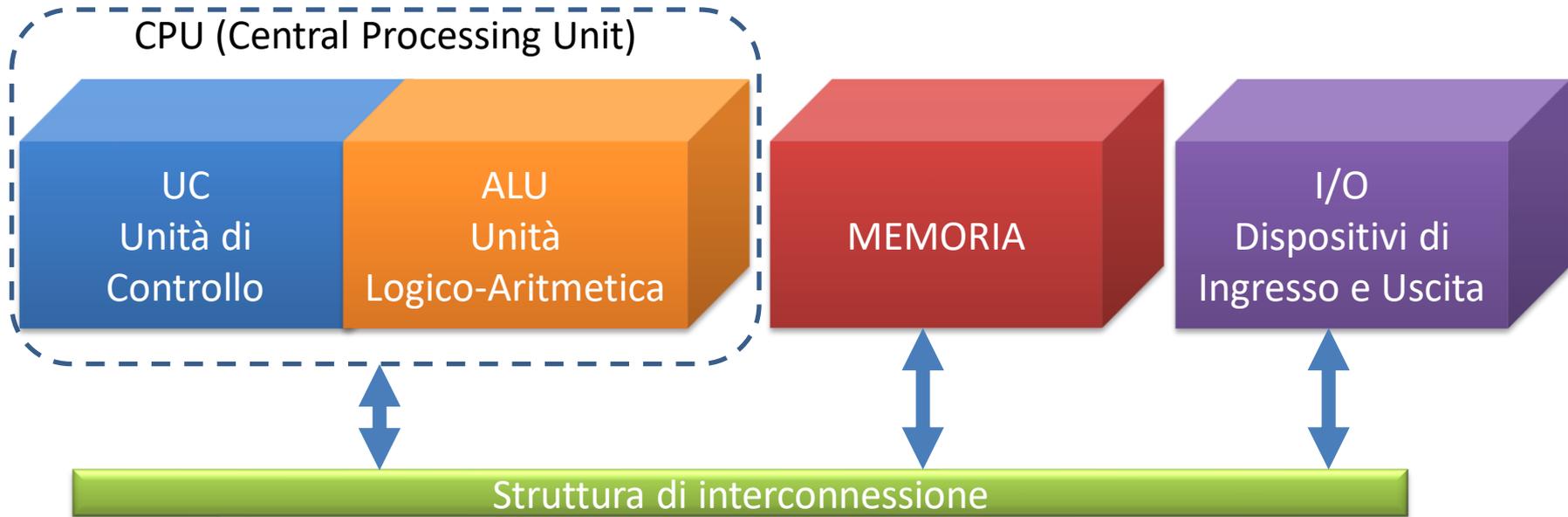


John von Neumann

Elaboratore Elettronico

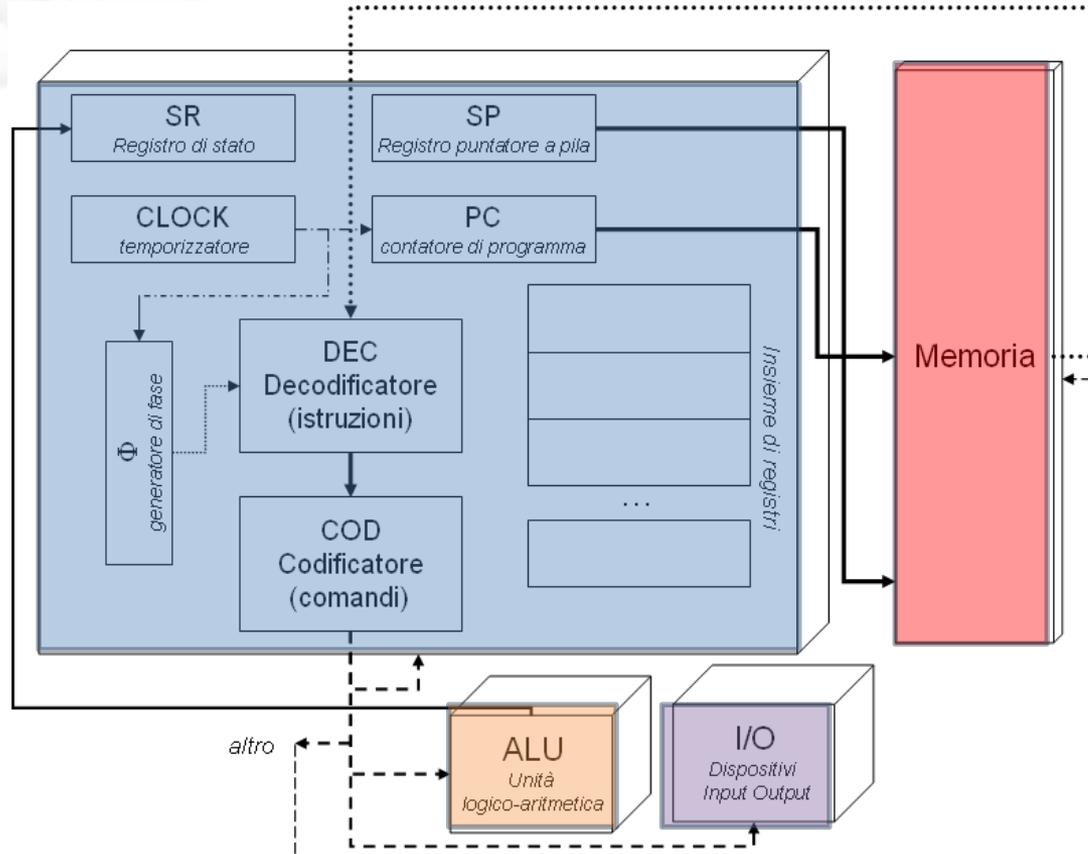
Macchina di von Neumann

Definizione. La Macchina di von Neumann è modello di elaboratore elettronico in grado di svolgere programmi ad uso generico costituito da quattro moduli (Unità di Controllo, Unità di Calcolo, Memoria Centrale, Dispositivi di Ingresso ed uscita) fondamentali ed una struttura atta ad interconnetterli



Elaboratore Elettronico

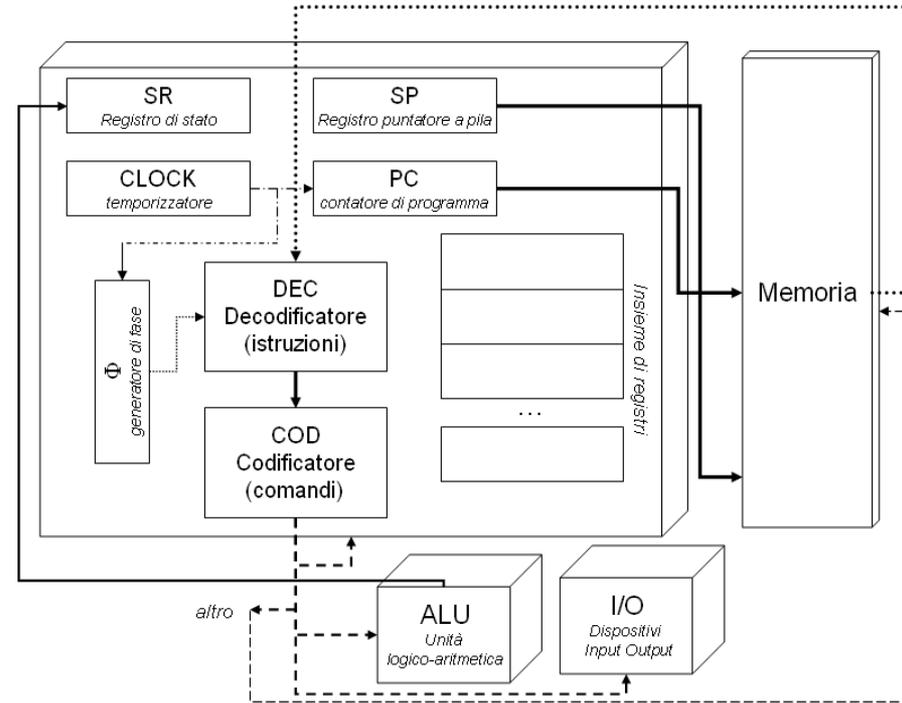
Macchina di von Neumann: dettaglio



Elaboratore Elettronico

Macchina di Von Neumann - Unità di Controllo

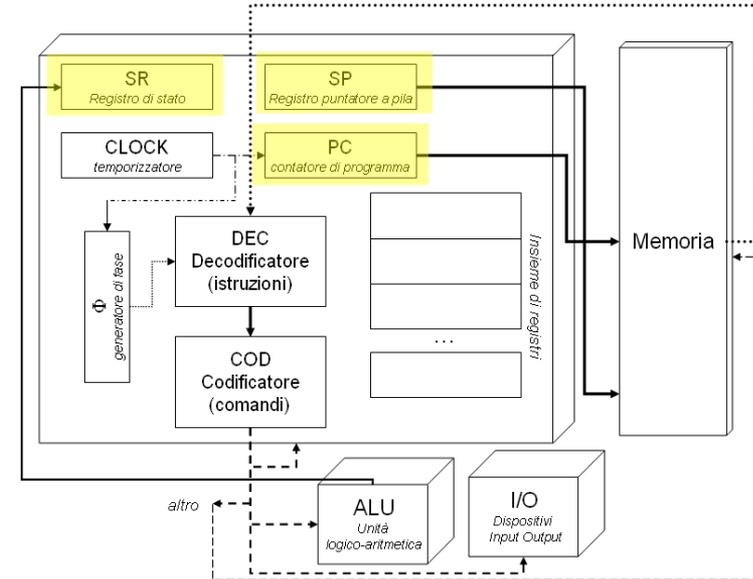
- ❑ L'**Unità di Controllo** (Control Unit, CU) è l'insieme dei circuiti elettronici predisposti a scandire le sequenze di operazioni elementari (i comandi) necessarie ad eseguire ogni singola istruzione
- ❑ Le **istruzioni** devono essere lette dalla Memoria Centrale (dove risiedono) e devono essere trasferite/copiate alla circuiteria interna all'Unità di Controllo
- ❑ La circuiteria dell'Unità di Controllo deve **riconoscere l'istruzione e generare i comandi** atti alla sua esecuzione (attivazione della struttura di interconnessione, passaggio dei dati e degli indirizzi in memoria, reperimento operandi, selezione del calcolo numerico,...)



Elaboratore Elettronico

Unità di Controllo: Registri ad uso speciale

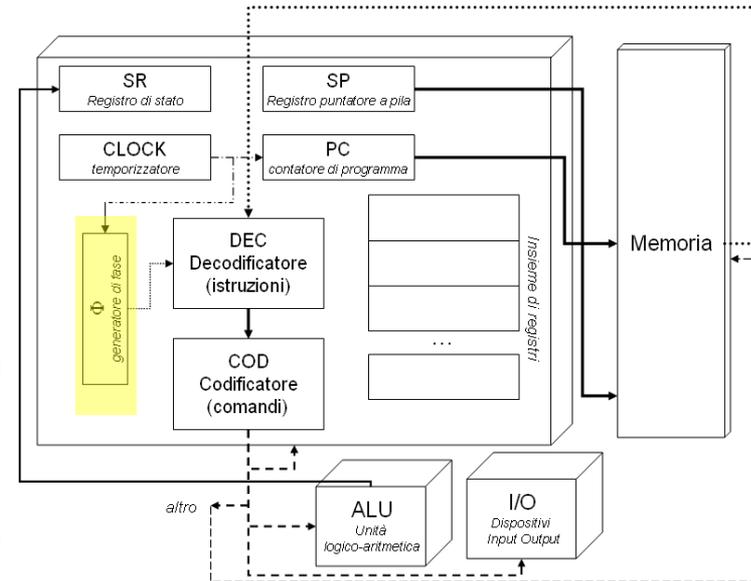
- ❑ I registri ad uso speciale sono :
 - ❑ Il **contatore di programma** (*program counter, PC*): è un registro contatore preselezionabile incrementato ad ogni impulso del clock, contenente l'indirizzo della cella di memoria dove è memorizzata l'istruzione da eseguire
 - ❑ Il **registro di stato** (*status register, SR* o *processor status word, PSW*): contiene informazioni che caratterizzano lo stato della CPU, tra cui, ad esempio, quelle relative all'ultima operazione eseguita (i *condition codes* provenienti dalla ALU)
 - ❑ Il **puntatore alla pila** (*stack pointer, SP*): contiene l'indirizzo della cima della pila (o canasta) cioè la zona di memoria usata per conservare dei dati utili all'elaborazione del programma (il passaggio di parametri)
- ❑ Il clock è un temporizzatore che emana segnali elettrici ad un intervallo di tempo. Il numero di colpi di clock al secondo definisce la **frequenza del processore** (dal 2018 si calcola in Ghz)



Elaboratore Elettronico

Unità di Controllo: Generatore di fase

- Il **generatore di fase** (tipicamente realizzato con un registro contatore) scandisce le fasi delle operazioni elementari eseguite dalla CU che possono essere così schematizzate:
 - Caricamento** (FETCH): lettura dalla memoria della parola puntata dal Program Counter. In questa fase la Memoria Centrale e il Codificatore presente nella CU si connettono per consentire il trasferimento dell'istruzione. Nel contempo il Program Counter si incrementa (in modo tale che al prossimo segnale del clock prenda l'istruzione successiva)
 - Decodifica** (DECODE): riconoscimento del tipo di istruzione e del modo di riferimento degli operandi. Non vi è alcuna connessione con componenti esterni perché il decodificatore è interno alla CU
 - Esecuzione** (EXECUTE): esecuzione dei comandi definiti dall'istruzione. Vi è connessione con tutte le unità richieste. Mentre le prime due fasi sono uguali per ogni istruzione, la fase di esecuzione varia in relazione all'operazione coinvolta (es.: se si effettua una operazione tra due valori deve essere attivata la ALU, se si deve riprodurre su carta un valore si deve attivare la Memoria Centrale che preserva il dato e la stampante)
 - In questa fase è incluso il **Reperimento Operandi** (LOAD): si prelevano i valori da elaborare
 - Spostamento** (MOVE) che ripristina il risultato nella destinazione opportuna



Elaboratore Elettronico

Unità di Controllo: Recupero e elaborazione di una istruzione

- ❑ Un **programma** è una successione di istruzioni
- ❑ Una **istruzione** ha una suddivisione in campi. Una prima parte (OPCODE) specifica il tipo di operazione da compiere e un'altra (DATA) i dati con i quali si deve operare o dove essi risiedono



ADD **#300, #100, #200**

Effettua la somma degli operandi presenti alle locazioni di memoria 100 e 200 e pone il risultato nella locazione 300

- ❑ Il prelievo e l'elaborazione dell'istruzione verte sullo **svolgimento di passi** elementari che sono ripetuti continuamente all'atto dell'accensione della macchina

FASE DI FETCH

Prelevare il valore contenuto nel PC e inviarlo in memoria per individuare la cella contenente l'istruzione

ISTR ← MEMORIA[PC]

Incremento del PC all'istruzione successiva

PC ← PC+1

Determinare il tipo dell'istruzione appena prelevata

DEC[ISTR]

Reperimento operandi

COD[ISTR_DATA]

Esecuzione

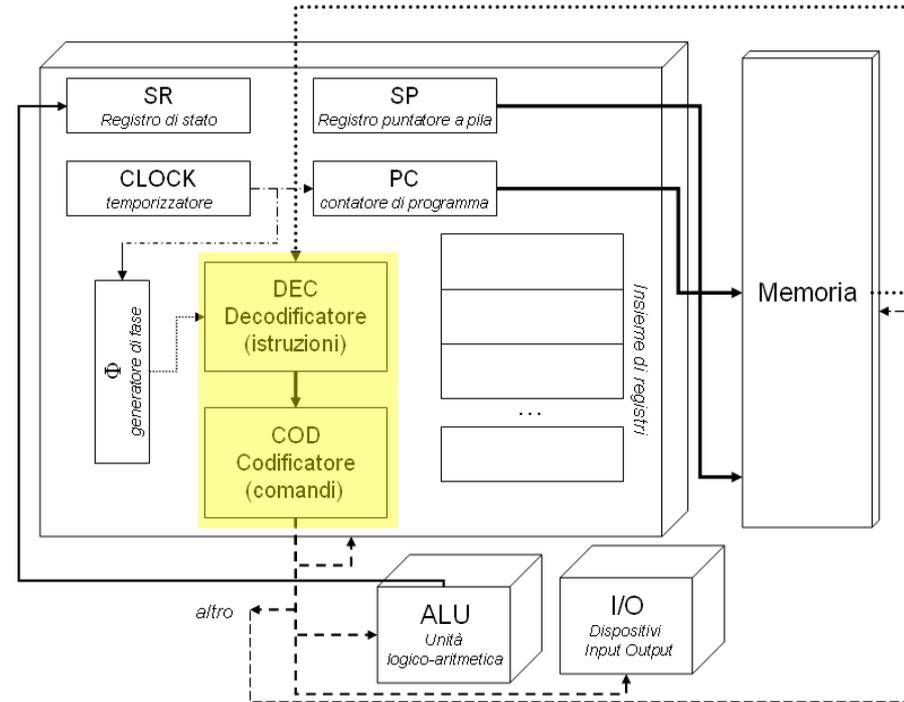
EXEC[ISTR_OPCODE]

Scrittura risultato o trasferimento operando elaborato

Elaboratore Elettronico

Unità di Controllo: Transcodificatore

- Una volta che l'istruzione è stata prelevata dalla Memoria Centrale e inviata all'Unità di Controllo è passata al **transcodificatore** (cioè il decodificatore delle istruzioni connesso col codificatore dei comandi) che riconosce l'istruzione e genera opportuni comandi per eseguire l'istruzione stessa
- Un decodificatore (**riconosce le istruzioni**) ha n linee di ingresso e 2^n linee di uscita
- Un codificatore (**genera comandi**) ha n linee di ingresso e $\log_2 n$ linee di uscita



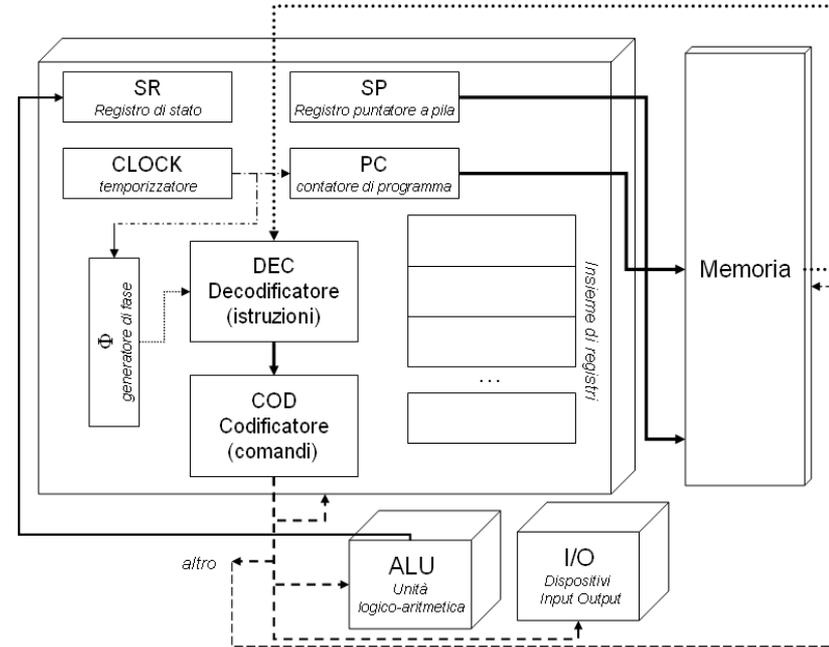
Elaboratore Elettronico

Unità di Controllo: *Esempio esecuzione istruzione*

□ Esempio

ADD #300,#100,#200

- ❖ L'istruzione viene letta dalla memoria e copiata nella CU per essere inviata al transcodificatore
- ❖ Il decodificatore, analizzando OPCODE, riconosce l'addizione
- ❖ Il codificatore, a sua volta, lancia dei segnali lungo le linee di ingresso della ALU che specificano il tipo di operazione che questa deve offrire
- ❖ Contestualmente il codificatore, analizzando il campo DATA, lancia dei segnali per prendere gli operandi in memoria alla locazione #100 e poi #200
- ❖ Una volta recuperati gli operandi si deve generare la connessione con la ALU disconnettendo la memoria
- ❖ La ALU esegue l'operazione in accordo ai segnali inviati dall'Unità di Controllo
- ❖ Si riattiva la linea tra la ALU e la Memoria Centrale per trasferire il risultato dalla ALU alla locazione #300



Elaboratore Elettronico

Unità di Controllo: *Esempio esecuzione istruzione*

li \$t2,102 *Copia nel registro ad uso generale \$t2 il valore 102*

PC

0000 0000 0000 0010



\$t1



\$t2



DEC

COD

Elaboratore Elettronico

Unità di Controllo: *Esempio esecuzione istruzione*

li \$t2,102 *Copia nel registro ad uso generale \$t2 il valore 102*

PC

0000 0000 0000 0010

+1



0101 | 0010 0110 0110

DEC

COD

2^{n-1}

\$t1

\$t2



Elaboratore Elettronico

Unità di Controllo: *Esempio esecuzione istruzione*

li \$t2,102 *Copia nel registro ad uso generale \$t2 il valore 102*

PC

0000 0000 0000 0011



0101 | 0010 0110 0110



\$t1

\$t2

0000 0000 0110 0110

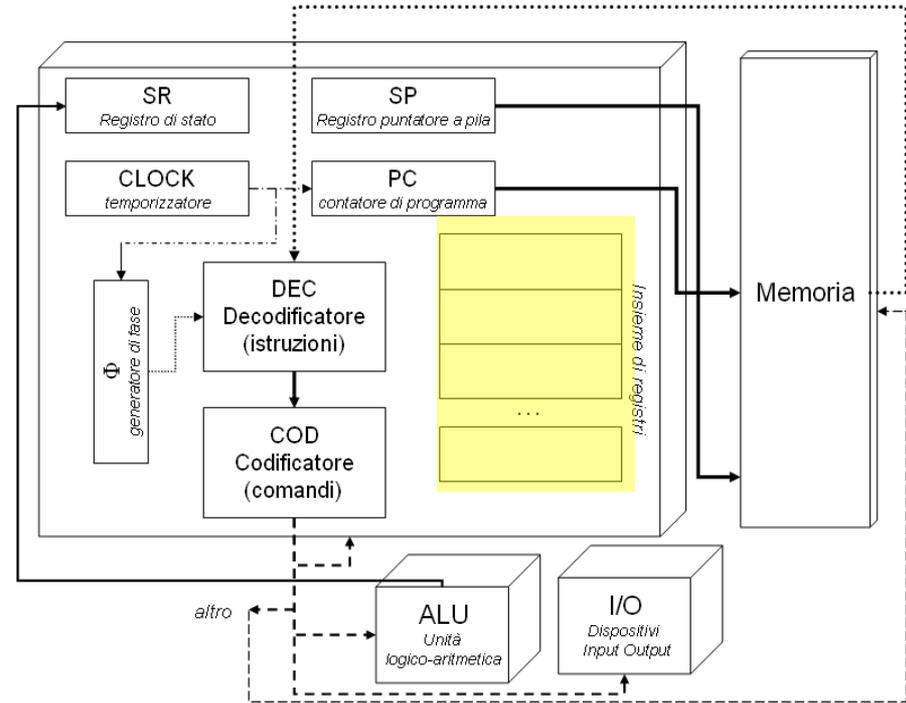
DEC

COD

Elaboratore Elettronico

Unità di Controllo: Registri ad uso generale

- ❑ L'insieme dei **registri ad uso generale**, denominato FR (*File Register*, archivio di registri), è utilizzato per memorizzare all'interno della CPU operandi e i risultati temporanei con lo scopo di diminuire il numero di accessi alla Memoria Centrale e di velocizzare il processo di elaborazione
- ❑ La dimensione di ciascun registro (o **parola**) è prefissata dal **progettista della macchina** (ad oggi ci sono architetture con parole di 64bit)

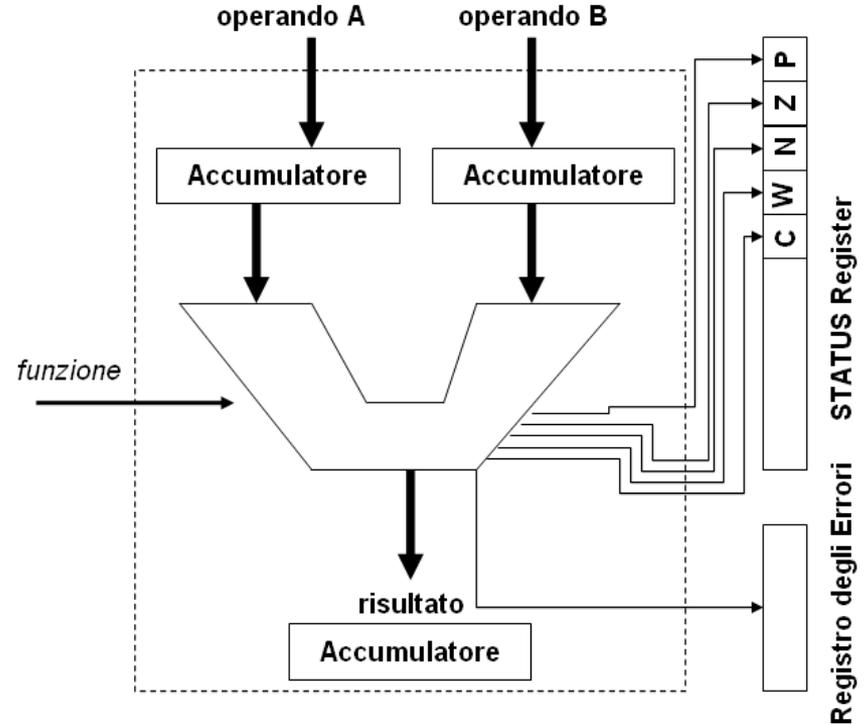


Più veloce ADD #300,#100,#200 o ADD \$t1,\$t2?

Elaboratore Elettronico

Unità di Calcolo: Funzionamento e Accumulatori

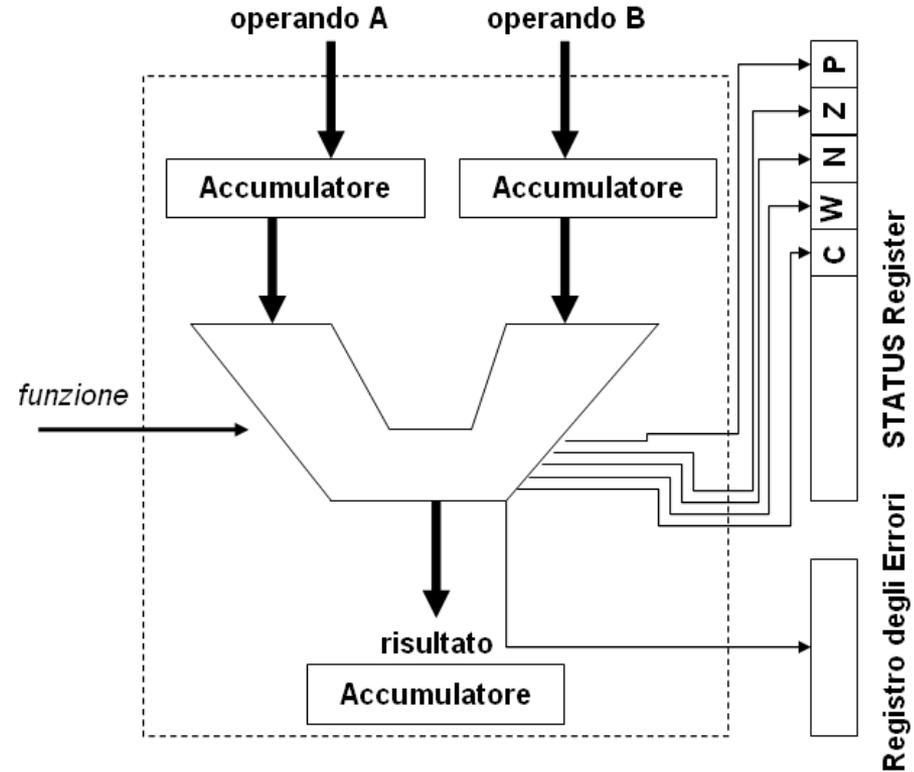
- ❑ L'unità logico-aritmetica è il componente che si occupa di effettuare operazioni logiche ed aritmetiche. Per il funzionamento di questa unità, di solito, sono impiegati un insieme di registri ad uso speciale che servono a contenere gli operandi e il risultato delle operazioni
- ❑ I registri speciali contenuti nella ALU, denominati **accumulatori**, sono trasparenti al programmatore (cioè il contenuto non può essere modificato dal programmatore mediante istruzioni) e svolgono la funzione di ospitare operandi e il risultato dopo l'esecuzione



Elaboratore Elettronico

Unità di Calcolo: Altri componenti

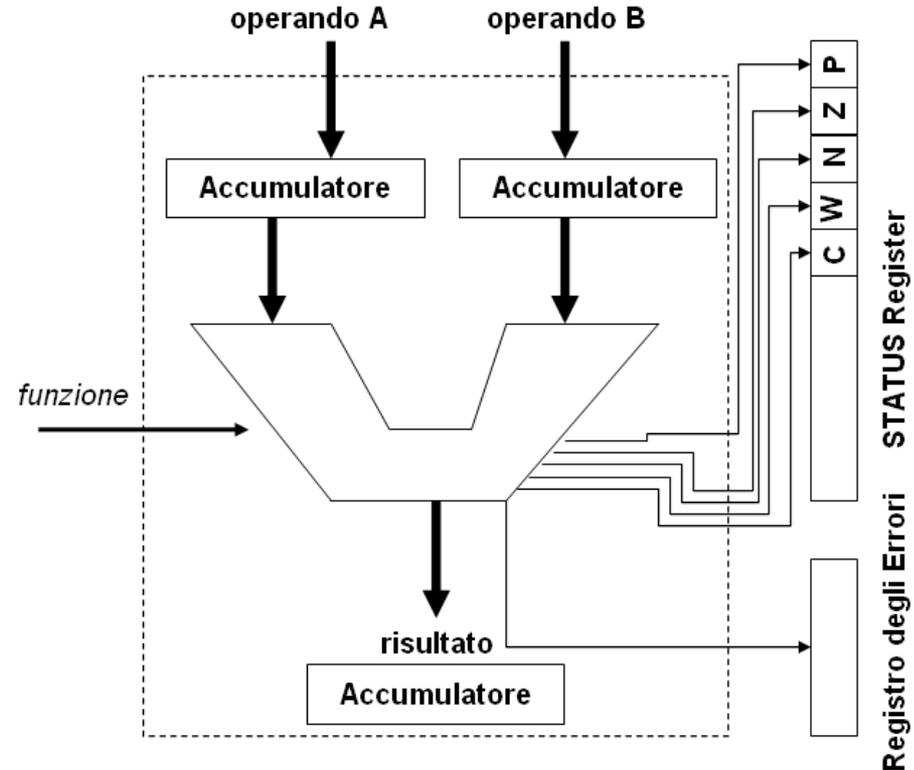
- ❑ Sono presenti delle linee di ingresso che individuano la funzione/operazione che deve essere attivata (le operazioni principali sono: ADD, SUB, MUL, DIV, NEG, AND, OR, COMP, TESTB, SHIFT) e delle linee di uscite su cui è ricondotto il risultato.
- ❑ La funzione/operazione si ricava dall'OPCODE dell'istruzione
- ❑ Inoltre ci sono delle linee di uscita denominate **condition code (codici di condizione o flags)** che riportano informazioni relative all'ultima operazione eseguita



Elaboratore Elettronico

Unità di Calcolo: Codici di Condizione

- ❑ Tra i **condition code** più importanti ci sono
 - ❑ **W - overfloW**: indica che l'ultima operazione ha dato luogo ad un risultato non rappresentabile (il valore risultante è non coerente)
 - ❑ ESEMPIO `ADD $t0, 4294967295, 10`
In `$t0` ci sarà il valore 9
 - ❑ **Z - Zero**: indica che l'operazione ha dato luogo al risultato zero
 - ❑ **N - Negative**: indica che l'operazione ha dato luogo ad un risultato negativo
 - ❑ **C - Carry**: indica che l'operazione ha dato luogo ad un trabocco
 - ❑ **I - Interrupt**: indica che è in corso una interruzione

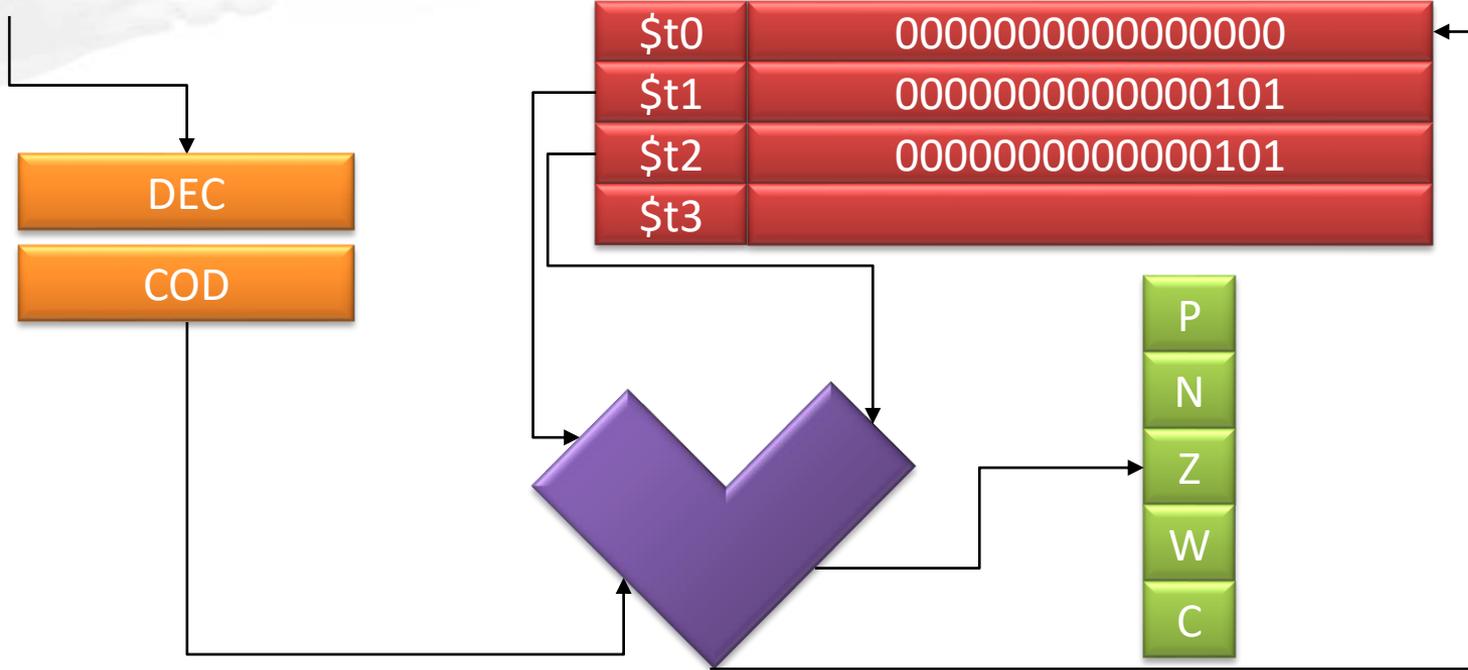


Elaboratore Elettronico

Unità di Calcolo: *Esempio esecuzione istruzione*

`sub $t0,$t1,$t2`

Sottrae l'operando contenuto nel registro ad uso generale \$t1 con quello presente in \$t2 e pone il risultato in \$t0



Elaboratore Elettronico

Unità di Calcolo: Codici di Condizione

- ❑ I condition code sono importanti perché consentono di realizzare i salti forzando il contenuto del program counter (cioè offrono al programma di intraprendere decisioni)

Problema. Si vogliono confrontare due valori e se sono uguali si memorizza il valore 200 altrimenti si memorizza il valore 100

```
000:      li $t0,5
001:      li $t1,5
002:      sub $t2,$t0,$t1
003:      btz #006
004:      li $t3,100
005:      j #007
006:      li $t3,200
007:
```

li (*load immediate*) mette in numero in un registro ad uso generico (non si accede in Memoria Centrale perché l'operando è già incluso nell'istruzione)

sub (*subtraction*) svolge la sottrazione tra i valori contenuti nei registri \$t0 e \$t1 e pone il risultato nel registro \$t2

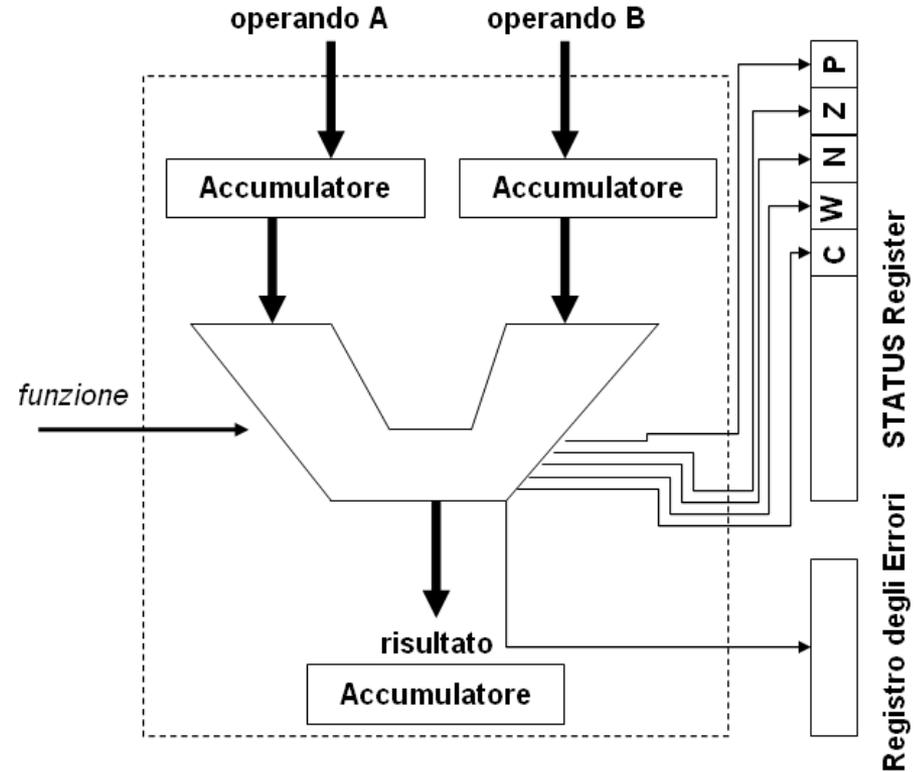
btz (*test bit z and brunch*) - SALTO CONDIZIONATO. Se il flag **Z** è 1 sovrascrive il valore del program counter con l'indirizzo (in questo caso #006)

j (*jump*) SALTO INCONDIZIONATO. Sovrascrive il valore del program counter con l'indirizzo seguente (in questo caso #007) Senza valutare i condition code

Elaboratore Elettronico

Unità di Calcolo: Registro degli errori

- Infine c'è un registro speciale detto **registro degli errori** nel quale sono riportate situazioni non risolvibili (es.: divisione per zero, radice quadrata di un numero negativo)



Elaboratore Elettronico

Unità di Calcolo: Esempi



istruzione	Significato	overflow	Negative	Zero
li \$t1, 45	#pone il valore 45 nel registro \$t1	0	0	0
li \$t2, 15	#pone il valore 15 nel registro \$t2	0	0	0
l li \$t3,-30	#pone il valore -30 nel registro \$t3	0	0	0
add \$t4,\$t1,\$t2	#Somma il valore del registro \$t1 con quello contenuto nel registro \$t2 e pone il risultato in \$t4	0	0	0
sub \$t4,\$t1,\$t2	#Sottrae il valore del registro \$t2 con quello contenuto nel registro \$t1 e pone il risultato in \$t4	0	0	0
add \$t5,\$t4,\$t3	#Somma il valore del registro \$t4 con quello contenuto nel registro \$t3 e pone il risultato in \$t5	0	0	1



Elaboratore Elettronico

Unità di Calcolo: Esempi

Istruzione	Significato	overflow	Negative	Zero
li \$t1, 4294967295	#pone il massimo intero memorizzabile nel registro \$t1	0	0	0
li \$t2, 1	#pone il valore 15 nel registro \$t2	0	0	0
mul \$t4,\$t1,3	#moltiplica il valore contenuto in \$t1 per 3 e pone il risultato in \$t4	1	0	0
add \$t4,\$t1,\$t2	#Somma il valore del registro \$t1 con quello contenuto nel registro \$t2 e pone il risultato in \$t4	1	0	1

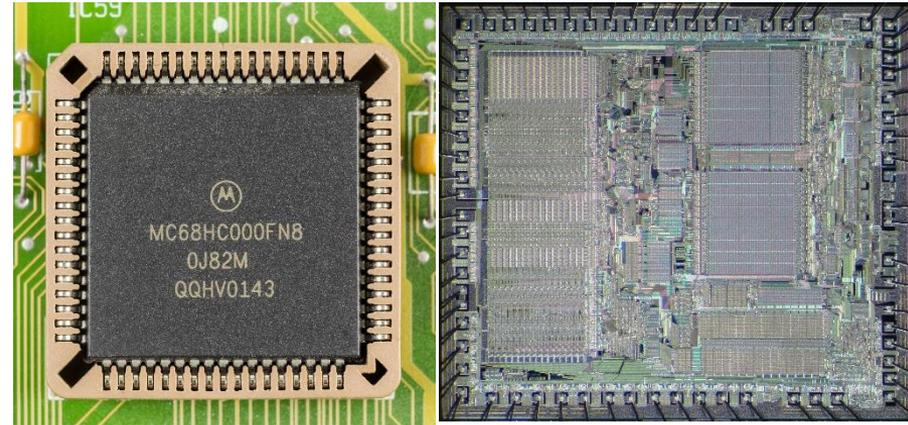
Elaboratore Elettronico

CPU

- ❑ La **CU** e la **ALU** identificano la **CPU** (*Central Processing Unit*) dell'elaboratore elettronico (il 'cuore' della macchina)
- ❑ Un processore si caratterizza:
 - ❑ **Potenza di calcolo:** il numero di operazioni che si possono elaborare in un secondo (determinato dalla frequenza di clock e dalla complessità delle istruzioni)
 - ❑ **Il processo di produzione:** la tecnica con cui si realizzano i componenti elementari (le porte logiche) con i quali si realizzano i circuiti elettronici (oggi si producono porte logiche da 50 a 22 nanometri). Migliore è il processo di produzione più componenti possono essere realizzate e si ha un minore consumo di energia e di dissipazione di calore
 - ❑ Dimensione della parola: maggiore è la lunghezza dei registri maggiore sono gli operandi rappresentabili e le istruzioni della macchina
 - ❑ L'overclock: una procedura che permette di aumentare la frequenza del processore (non è concessa su tutti i modelli).



PROCESSORE PENTIUM DI INTEL



PROCESSORE 68000 MOTOROLA: esterno (sinistra) interno (destra)

Elaboratore Elettronico

Memoria Centrale: Generalità



Definizione. La **Memoria Centrale** è una memoria volatile ad accesso casuale (RAM, *random access memory*) costituita da tante **celle** (o locazioni) ciascuna delle quali può immagazzinare una stringa binaria di lunghezza finita n

- ❑ Il termine **accesso casuale** indica si può scrivere o leggere il contenuto di una cella in tempo unitario indipendentemente dalla sua posizione

Le celle della Memoria Centrale sono numerate in sequenza e tale numero prende il nome di **indirizzo** della cella

- ❑ L'indirizzo specifica univocamente una cella

I dati archiviati in Memoria Centrale possono essere: **istruzioni, operandi o indirizzi**

INDIRIZZO

CELLA

000
001
002
003
004
005
006
007



....

....

Elaboratore Elettronico

Memoria Centrale: Generalità

- ❑ La Memoria Centrale presenta delle **aree riservate**, in cui risiedono programmi basilari utili al funzionamento della macchina (Sistema Operativo) o nelle quali si svolgono operazioni particolari (stack, zona per trasferimento I/O,...) e infine un'**area di uso comune**, dove risiedono programmi e dati

SISTEMA OPERATIVO

AREA AD USO
COMUNE
(programmi)

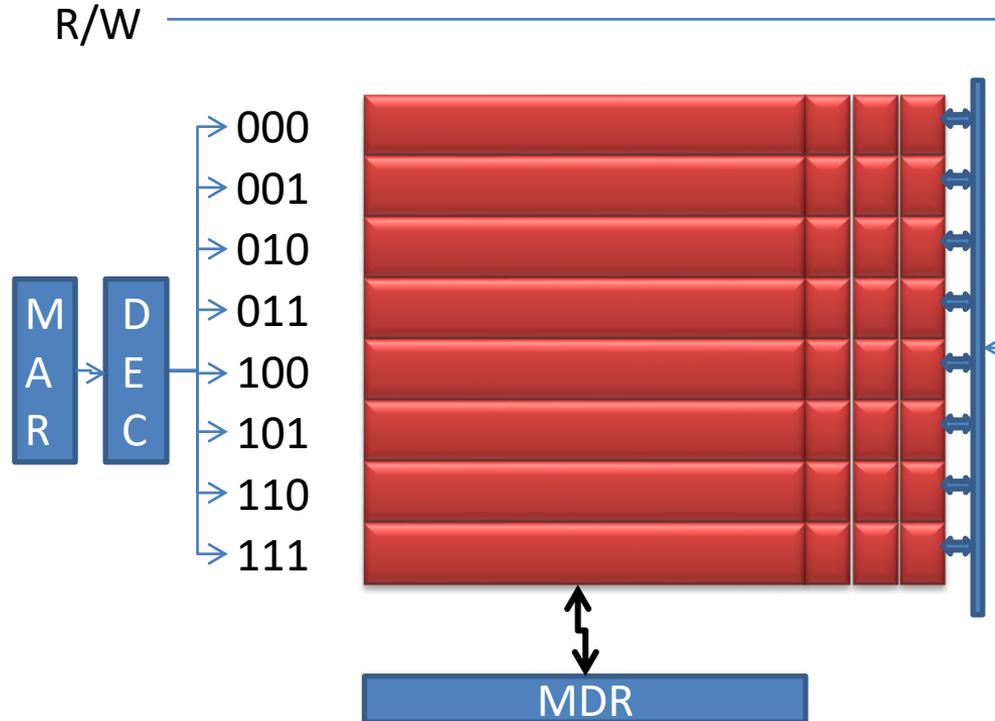
I/O

Elaboratore Elettronico

Memoria Centrale: Componenti

- ❑ Per poter operare sulla Memoria Centrale è necessario che ci siano:
 - ❑ **linee di ingresso** che specificano un indirizzo (in alcuni testi si fa riferimento al registro **MAR**, *memory address register*)
 - ❑ **linee di uscita** per poter inviare o trasferire il dato (in alcuni testi si fa riferimento al registro **MDR**, *memory data register*)
 - ❑ **un segnale di controllo** (generato dalla CU) per la lettura o la scrittura del dato

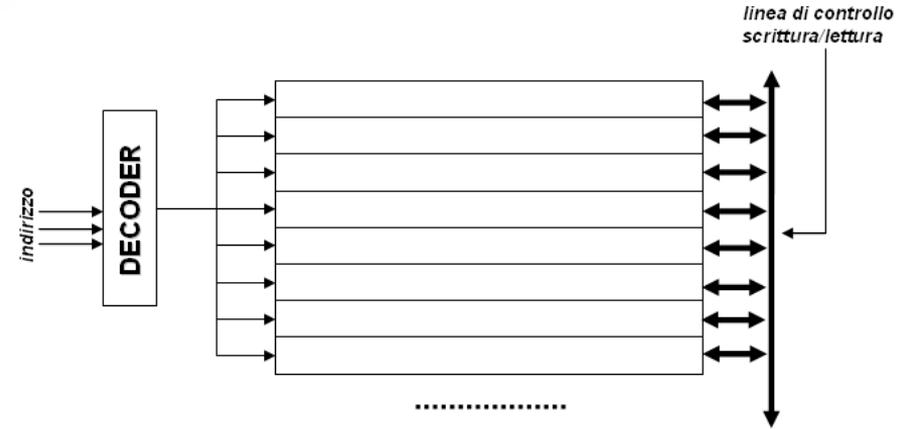
Osservazione. Oltre alle celle riservate a preservare gli operandi ci possono essere bit supplementari per: la protezione da scrittura della cella, le impostazioni del bit di parità,....



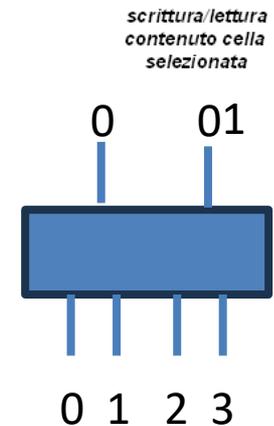
Elaboratore Elettronico

Memoria Centrale: Organizzazione fisica

- ❑ È prevista, pertanto, una architettura costituita da un **decodificatore** che riceve in ingresso l'indirizzo della cella di memoria alla quale si vuole accedere ed una linea che abilita questa a porre il suo contenuto in uscita dalla memoria o trascrivere in essa il dato da memorizzare
- ❑ Un decodificatore ha n linee di ingresso e 2^n linee di uscita. Per ogni valore in ingresso si seleziona una ed una sola linea di uscita



Linea di uscita (segnale di selezione)	Linee di entrata (indirizzo)
0	00
1	01
2	10
3	11



scrittura/lettura contenuto nella selezionata

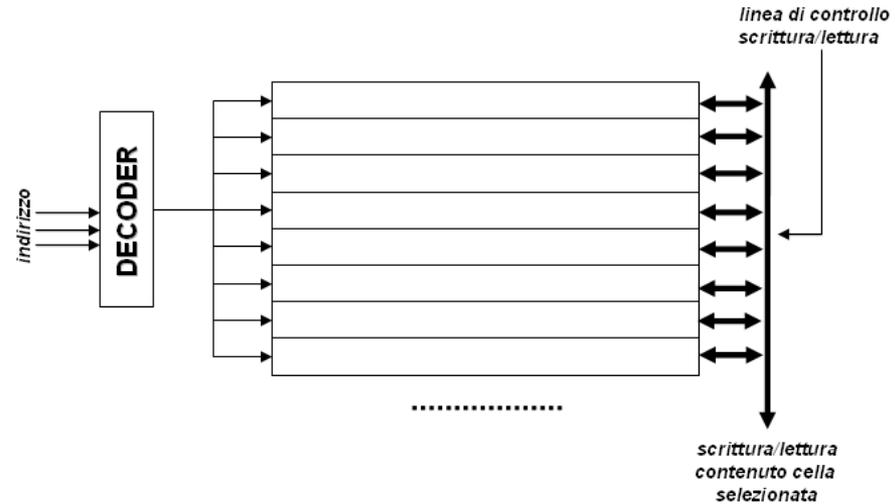
Elaboratore Elettronico

Memoria Centrale: Organizzazione fisica (limiti)

- Una organizzazione di questo tipo è **impraticabile** nel caso in cui la parola che specifica l'indirizzo abbia una grande dimensione. Con un indirizzo di lunghezza m è possibile far riferimento a 2^m celle di memoria. Nel caso in cui il valore di m sia abbastanza grande (superi la decina di unità) una architettura gestita da un singolo decodificatore è da escludere

Osservazione. Dal 2010 si usano sistemi con parole di lunghezza 32; pertanto è possibile indirizzare $2^{32} = 4294967296$ celle di memoria. Un decoder che abbia 32 linee di ingresso e più di 4 miliardi di linee di uscita è una possibilità implementativa da escludere.

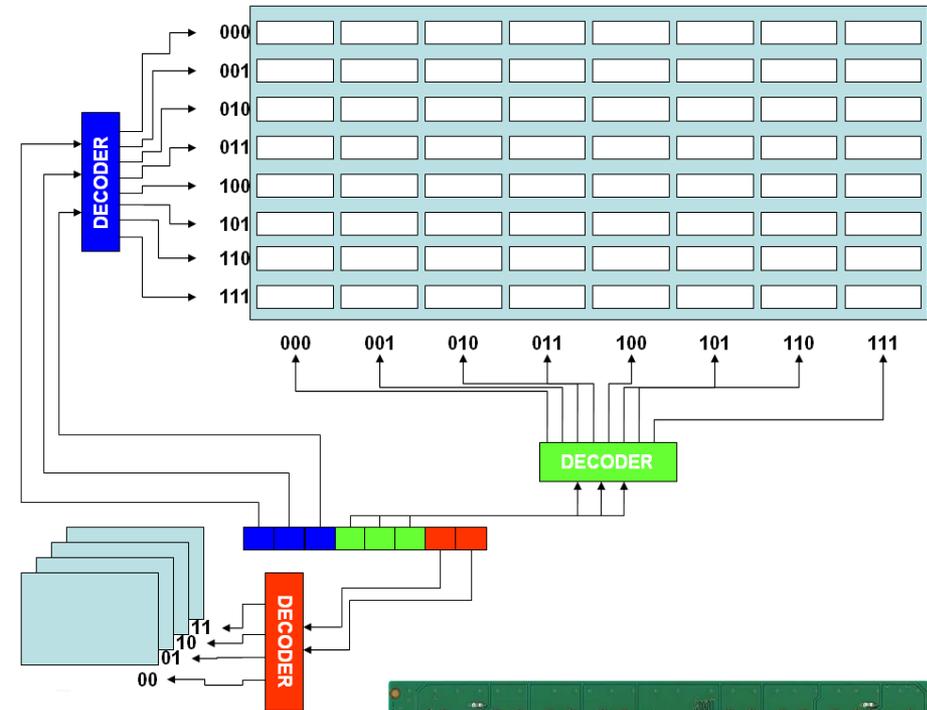
Inoltre dal 2014 i sistemi impiegano parole da 64bit



Elaboratore Elettronico

Memoria Centrale: Organizzazione fisica (soluzione)

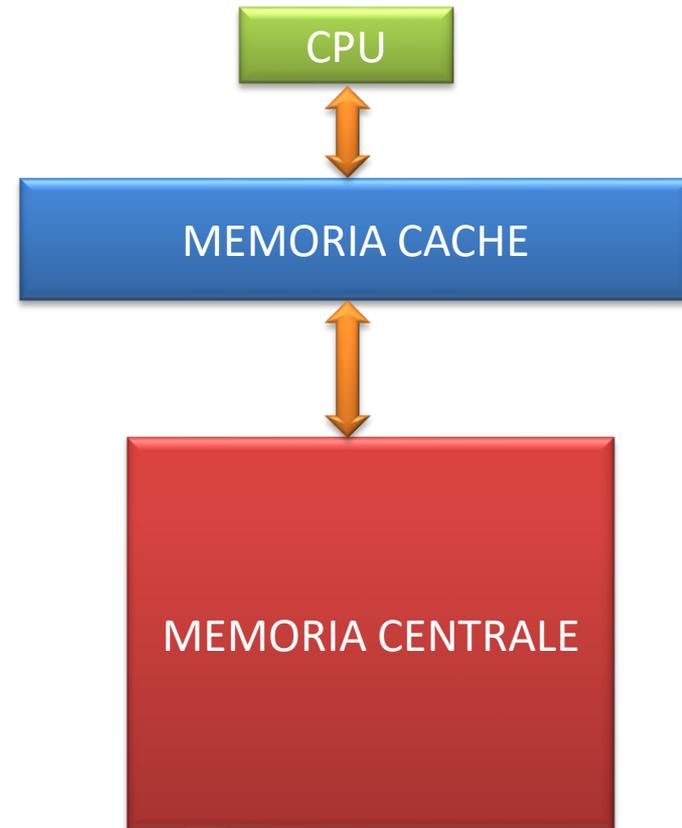
- ❑ Si ricorre ad una **suddivisione logica della memoria** (*multi-dimensione*)
- ❑ La memoria è suddivisa logicamente in **banchi** (o piastre) e **blocchi**
- ❑ L'indirizzo è suddiviso in campi aventi significato diverso
 - ❑ Ad esempio nel caso di un solo banco con quattro blocchi avremo una suddivisione in tre campi : il primo campo indicante il banco e gli altri due campi la posizione in cui risiede la cella
- ❑ In questo modo il risparmio in termini di linee di connessione tra decoder e celle di memoria è del 90%-99%



Elaboratore Elettronico

Memoria Cache

- ❑ Per migliorare le prestazioni si introduce una **memoria cache** tra la CPU e la Memoria Centrale
- ❑ La memoria cache è più piccola (KB, cioè mille byte, fino a MB, ovvero un milione di byte) della Memoria Centrale (GB, un miliardo di byte) ed è fatta con componenti più rapidi (e più costosi)
- ❑ La memoria cache è prossima alla CPU (questo impedisce il ritardo del prelievo dei dati residenti in memoria)
- ❑ Ad intervalli regolari la Memoria Centrale trasferisce un gruppo di istruzioni del programma in esecuzione alla memoria Cache che è direttamente elaborato dal processore



Elaboratore Elettronico

Input/Output: generalità

Definizione. Le **Unità di input/output** (dispositivi di Ingresso/uscita o periferiche) consentono di collegare l'elaboratore, ed in particolare la Memoria Centrale, con il mondo esterno (persone o altri dispositivi)

- Esistono numerosi tipi di periferiche con caratteristiche molto varie che comportano problemi relativi alla conversione tra rappresentazione interna ed esterna dell'informazione



Elaboratore Elettronico

Input/Output: caratteristiche

- ❑ Le **velocità di trasferimento** sono inferiori a quelle possibili all'interno delle altre componenti (CPU, memoria) a causa della natura tecnologica di fabbricazione (componenti meccanici,...) e quindi l'uso delle periferiche comporta **problemi di sincronizzazione ed adattamento della velocità**

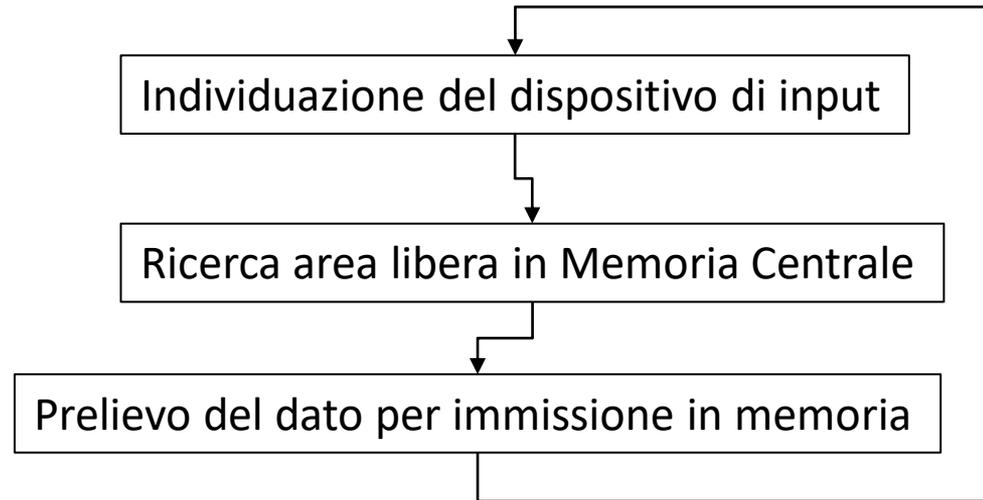
Dispositivo	Comportamento	Dati scambiati (KB/s)
Tastiera	Input	0.01
Mouse	Input	0.02
Stampante ad aghi	Output	1
Floppy	Storage	50
Stampante laser	Output	100
Disco	Storage	10.000
LAN	Input/Output	10.000
Display	Output	30.000

Elaboratore Elettronico

Input/Output: protocollo

- ❑ Nel momento in cui c'è un trasferimento dati è opportuno che il dispositivo coinvolto e il processore operino in modo coordinato mediante un insieme di regole: **il protocollo**

PROTOCOLLO INPUT

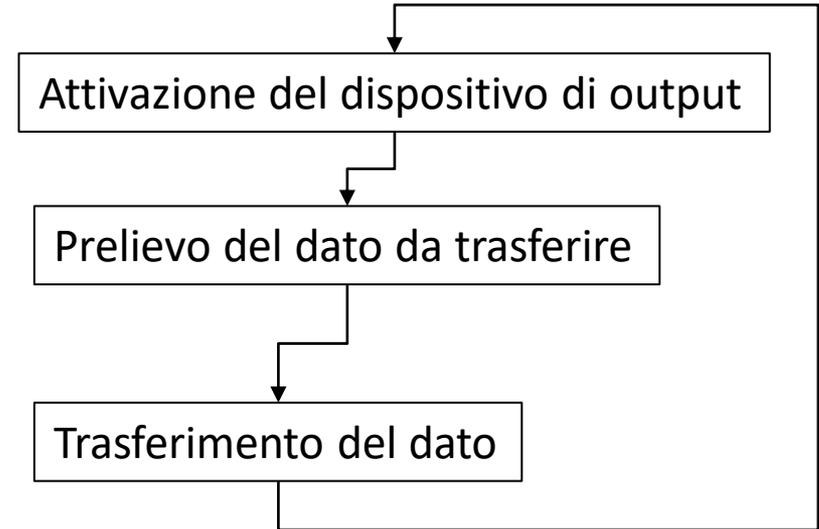


Elaboratore Elettronico

Input/Output: protocollo

- ❑ Nel momento in cui c'è un trasferimento dati è opportuno che il dispositivo coinvolto e il processore operino in modo coordinato mediante un insieme di regole: il **protocollo**

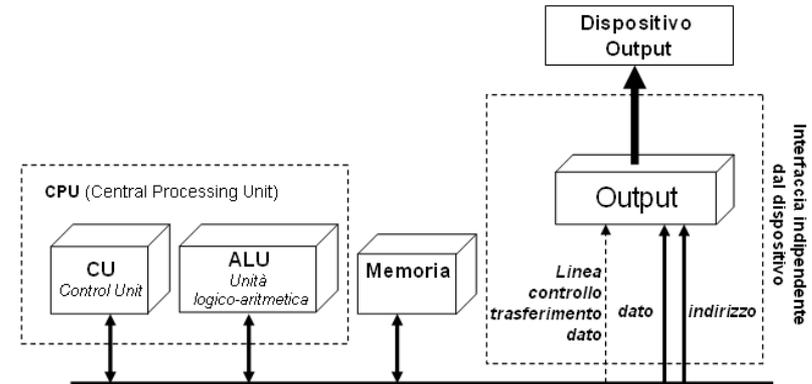
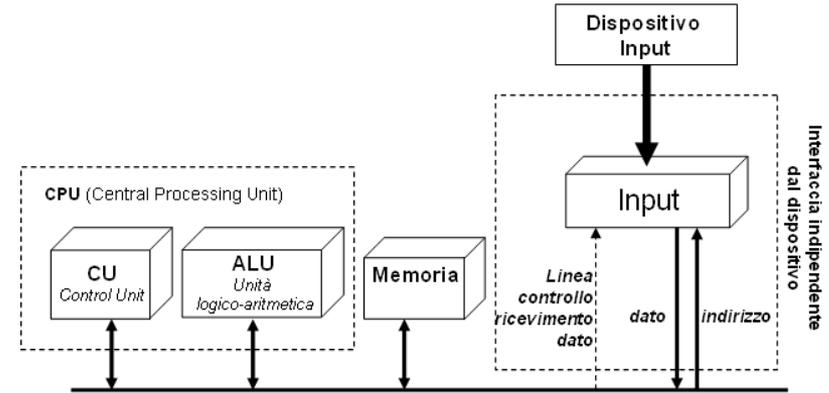
PROTOCOLLO OUTPUT



Elaboratore Elettronico

Input/Output: protocollo

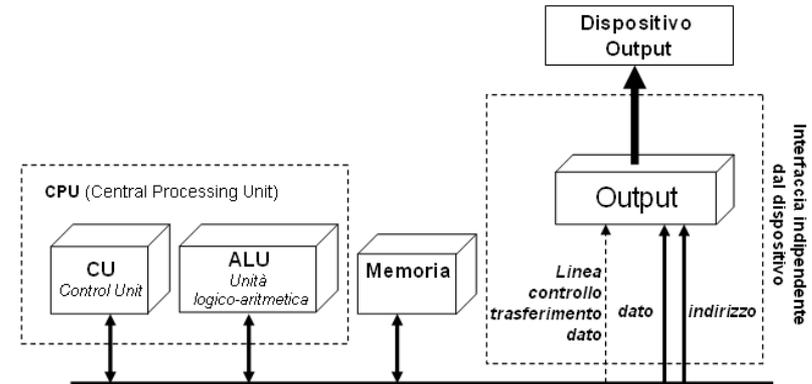
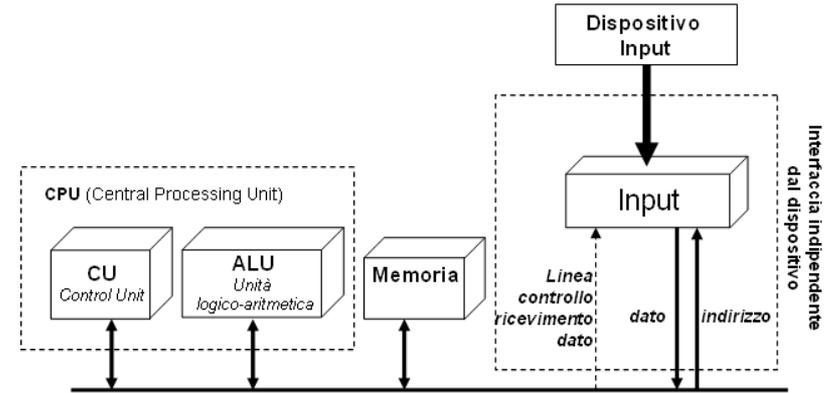
- ❑ Per interagire con il processore ogni dispositivo deve essere dotato di un **modulo di I/O** (o modulo di interfaccia I/O), cioè un circuito che colloquia con il processore inviando e ricevendo (tramite un bus di I/O) i segnali che, secondo il protocollo, controllano le operazioni di trasferimento dalla/verso la Memoria Centrale



Elaboratore Elettronico

Input/Output: protocollo

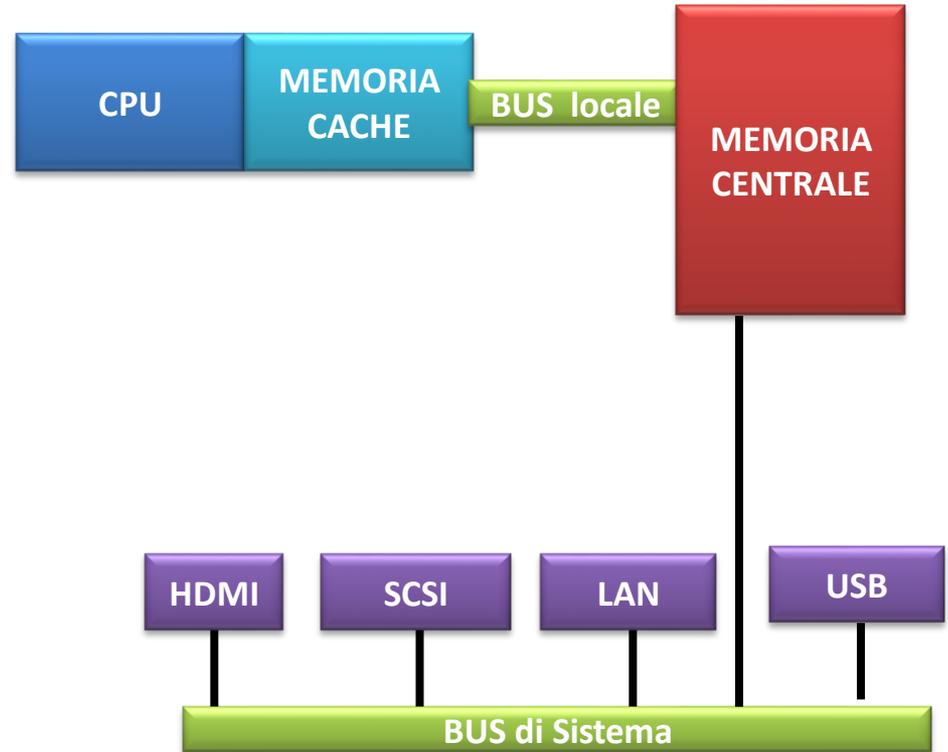
- Il protocollo di I/O pertanto è caratteristico dell'elaboratore, in quanto determinato dal modo di operare del processore stesso, cioè dall'insieme di istruzioni di cui il processore può disporre per i trasferimenti
- Questo vuol dire che **i diversi dispositivi** esterni collegati allo stesso calcolatore **devono rispettare tutti lo stesso protocollo di I/O, indipendentemente dalla natura delle informazioni trasferite a dalla struttura fisica del dispositivo.** Solamente in seguito il dispositivo darà il giusto significato al codice ricevuto



Elaboratore Elettronico

Interconnessione (BUS)

- ❑ Per interconnettere le unità della MdVN si usa il **bus**
- ❑ Il bus è un fascio di fili (da 50 a 100) di rame dotati di connettori (*buffer tri-state*) che consentono la connessione tra la CPU, la Memoria Centrale e le periferiche di I/O
- ❑ Nei primi modelli si usava un solo bus (**mono-bus** o uni-bus)
- ❑ Attualmente si usano più bus (**multi bus**): uno specifico tra la CPU e la Memoria Centrale e (almeno) un altro bus tra la Memoria Centrale e le periferiche





Elaboratore di II^a generazione

Elaboratore Elettronico

MIPS Project

- ❑ Alla fine degli anni 70 del XX secolo, un gruppo di ricercatori dell'Università di Berkeley definirono una nuova architettura (progetto *Reduced Instruction Set Computer*, o RISC) su cui era possibile elaborare un insieme di istruzioni completo ma minimo
- ❑ Tra i pionieri ci furono i ricercatori della Stanford University, tra cui David Patterson e John Hennessy
- ❑ I due accademici idearono una nuova architettura, il **Microprocessor without Interlocked Pipelined Stages** (MIPS), in cui ogni istruzione era eseguita alla fine di un singolo ciclo macchina



Elaboratore Elettronico

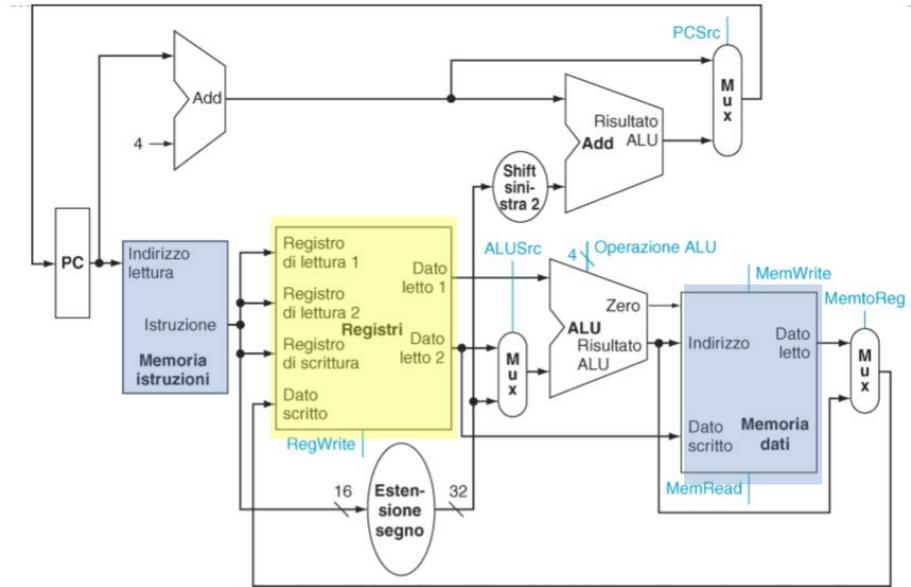
MIPS Project: Architettura

Le modifiche fisiche fondamentali rispetto alla Macchina di von Neumann furono:

- uno **sdoppiamento della Memoria Centrale** in due parti disgiunte, una riservata agli operandi e l'altra alle istruzioni

- la presenza di **molte registri ad uso generale** all'interno della CPU

- Poiché ogni istruzione è eseguita alla fine di un singolo colpo di clock, sebbene con un po' di approssimazione, si può affermare che in tale architettura se il processore ha una frequenza di 1Ghz si svolgono 1 miliardo di operazioni ogni secondo



Elaboratore Elettronico

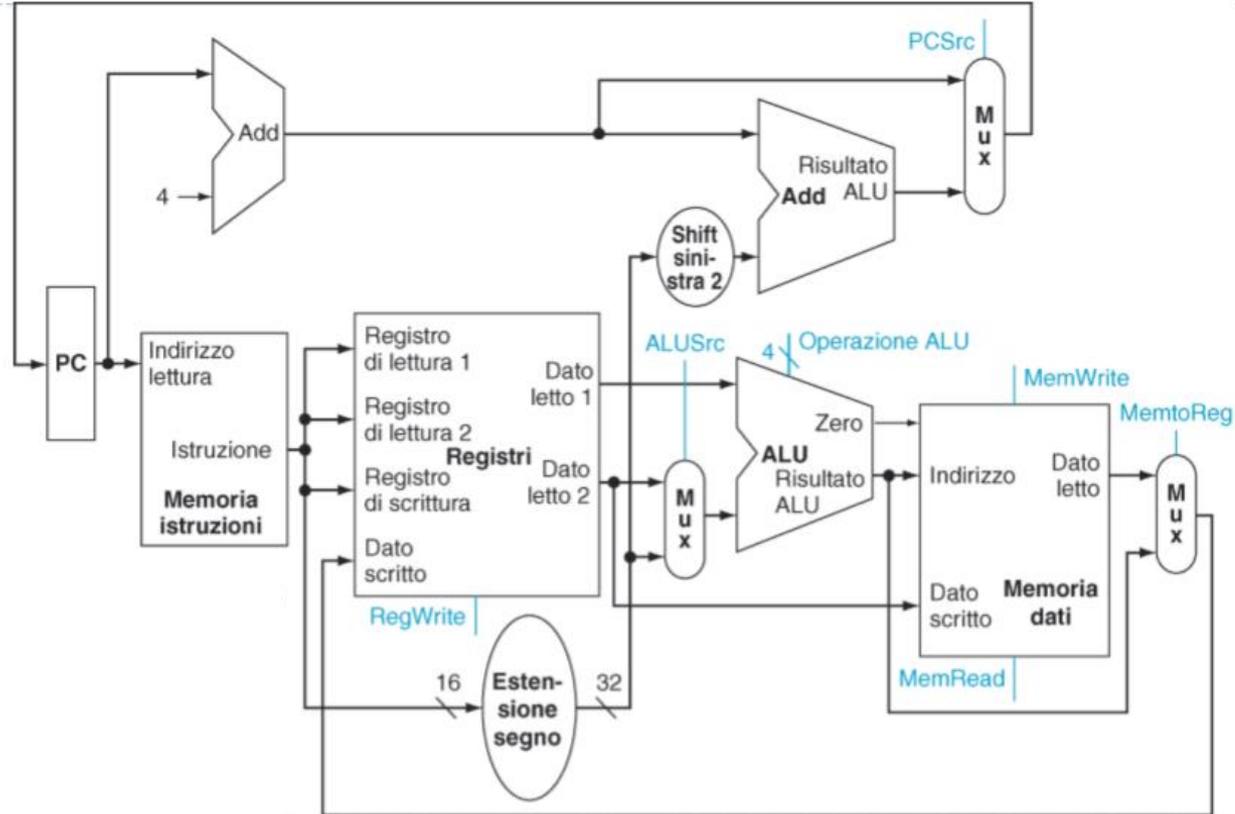
MIPS Project: Architettura (dettaglio)

add \$t3,\$t1,\$t2

li \$t3,\$t1,5

lw \$t1, #300

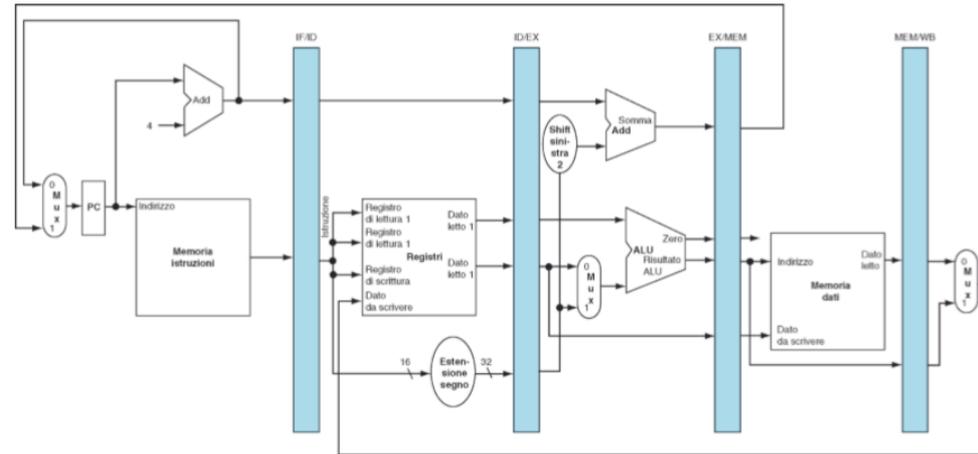
add #300, #200,#100



Elaboratore Elettronico

MIPS Project: Canalizzazione (o pipeline)

- Il MIPS, in seguito, fu migliorato per applicare la **canalizzazione** (pipeline): le istruzioni non erano più eseguite sequenzialmente ma - rendendo indipendenti le fasi in cui si preleva, codifica ed esegue una singola istruzione - si procedeva alla loro sovrapposizione, migliorando le prestazioni complessive della macchina



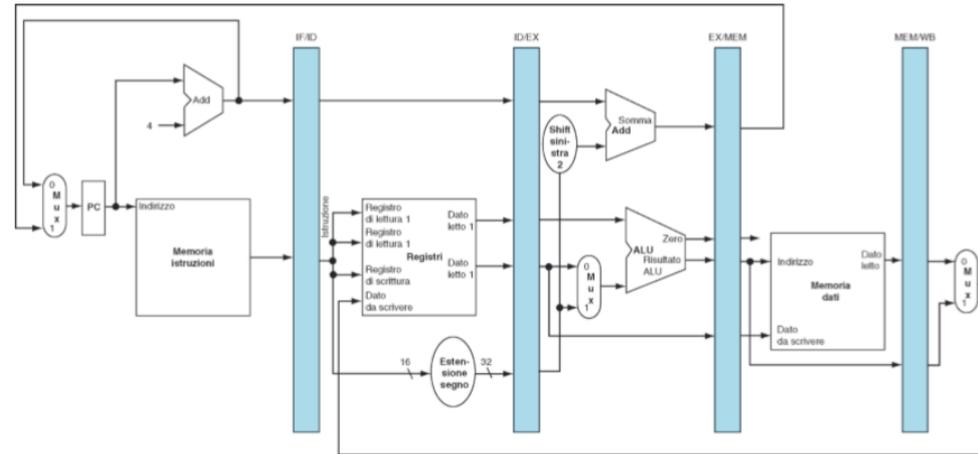
Elaboratore Elettronico

MIPS Project: Canalizzazione (o pipeline)

Definizione. Il **pipeline**, o **canalizzazione**, è una tecnica che consiste nella scomporre un circuito elettrico (nel particolare una rete combinatoria) in un sotto circuiti più semplici mediante l'inserimento di opportuni **registri di disaccoppiamento (interlock)**

❑ Questo accorgimento permette di aumentare la frequenza di clock e svolgere più operazioni in **parallelo**

❑ **NB: le istruzioni sono svolte in pseudo-parallelismo**



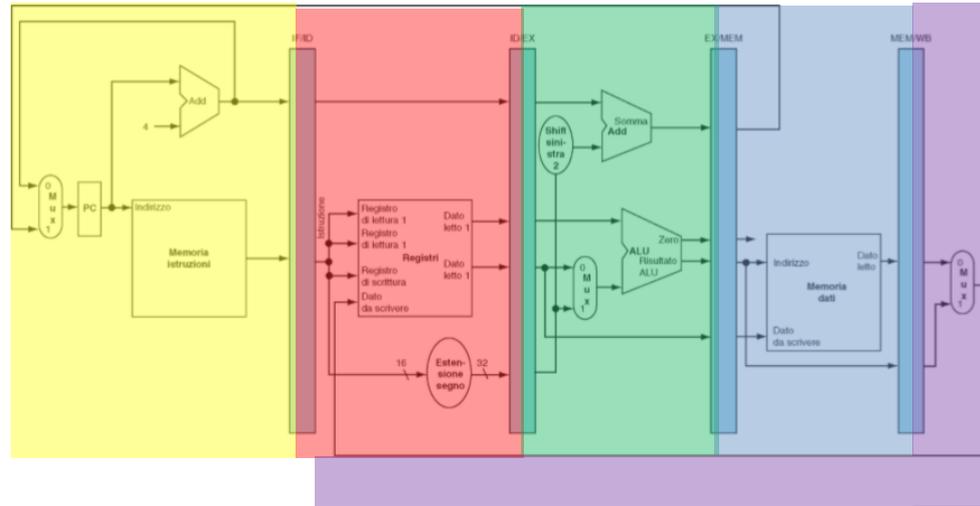
Elaboratore Elettronico

MIPS Project: Canalizzazione (o pipeline)

- L'esecuzione di una istruzione, in una macchina RISC (es.: MIPS), è di solito suddivisa in cinque **fasi** (o **sezioni** o **stage**):

- ❖ • **F – Fetch**: prelievo istruzione (con incremento del Program Counter)
- ❖ • **D – Decode**: decodifica/riconoscimento istruzione
- ❖ • **E – Execution**: esecuzione istruzione
- ❖ • **M – Memory**: accesso in memoria per scrittura o lettura (*load* o *store*)
- ❖ • **WB – Write Back**: quando il risultato dell'ALU o quello letto dalla memoria viene messo nel registro destinazione

- In ogni momento **solo una unità funzionale è attiva** e le altre non sono impegnate



Elaboratore Elettronico

MIPS Project: Macchina NON canalizzata

Non c'è una suddivisione delle unità funzionali:

- Fetch:** Memoria Istruzioni (e aggiornamento PC)
- Decode:** Blocco registri (e CU)
- Execute:** ALU
- MEM:** Memoria dati
- Write Back:** Blocco registri

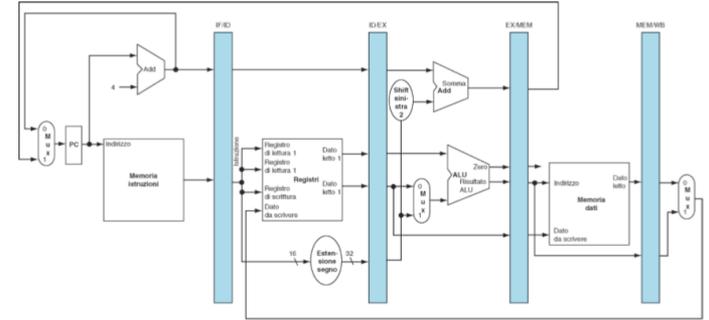
Esecuzione sequenziale

Δt	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
Istruzione 1	F	D	E	M	WB																					
Istruzione 2						F	D	E	M	WB																
Istruzione 3											F	D	E	M	WB											
Istruzione 4																F	D	E	M	WB						
Istruzione 5																						F	D	E	M	WB

Elaboratore Elettronico

MIPS Project: Macchina canalizzata

In una **macchina canalizzata**, cioè con pipeline, ogni unità funzionale elabora la fase che gli corrisponde e poi passa all'istruzione successiva



Esecuzione sequenziale

Δt	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Istruzione 1	F	D	E	M	WB																
Istruzione 2		F	D	E	M	WB															
Istruzione 3			F	D	E	M	WB														
Istruzione 4				F	D	E	M	WB													
Istruzione 5					F	D	E	M	WB												

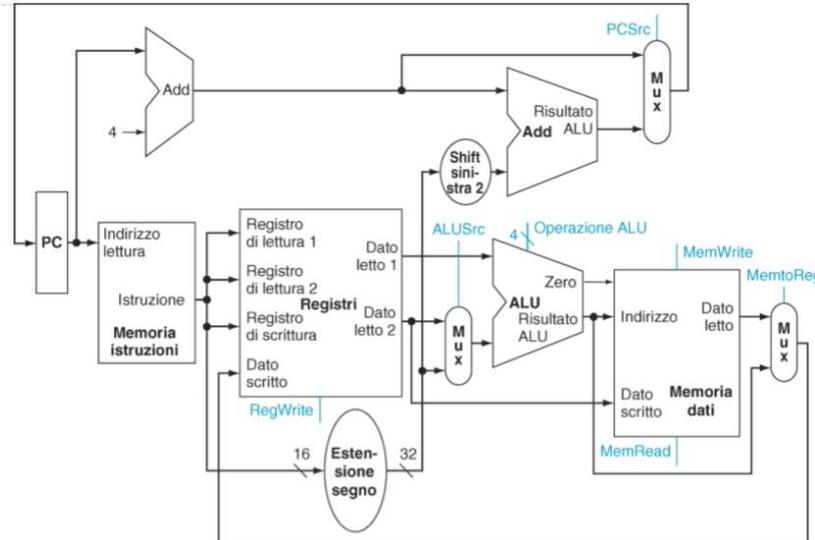
In questo modo una volta che la canalizzazione arriva a **regime**, cioè quando tutte le unità funzionali sono occupate e cioè si svolgono fino a n istruzioni contemporaneamente in n fasi diverse, alla fine di una fase si completa una istruzione

Elaboratore Elettronico

MIPS Project: Definizione

Definizione: Una macchina RISC (es.: MIPS) è una architettura nella quale la Memoria Centrale è suddivisa in due parti disgiunte una contenente le istruzioni e l'altra i dati

Con questa conformazione la macchina è in grado di svolgere, nella sua forma più semplice, una istruzione in un solo colpo di clock e nella forma canalizzata (mantenendo la stessa frequenza di clock di una macchina non canalizzata) più istruzioni



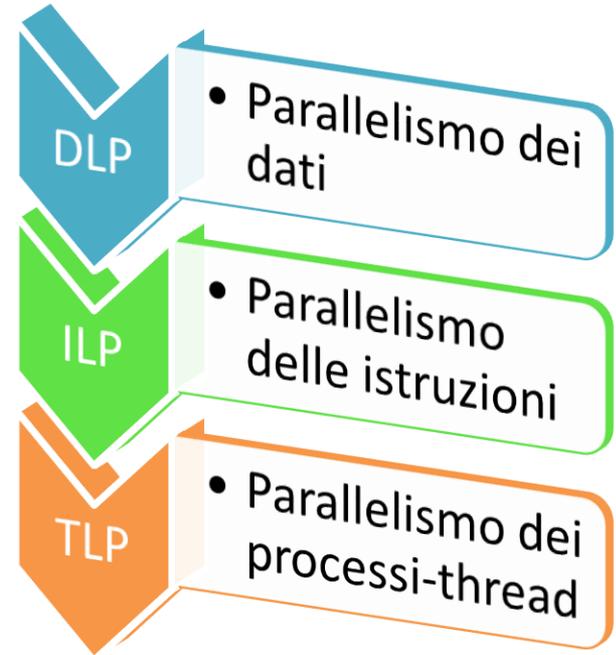


Elaboratori di III^a generazione

Elaboratore Elettronico

Tipi di parallelismo

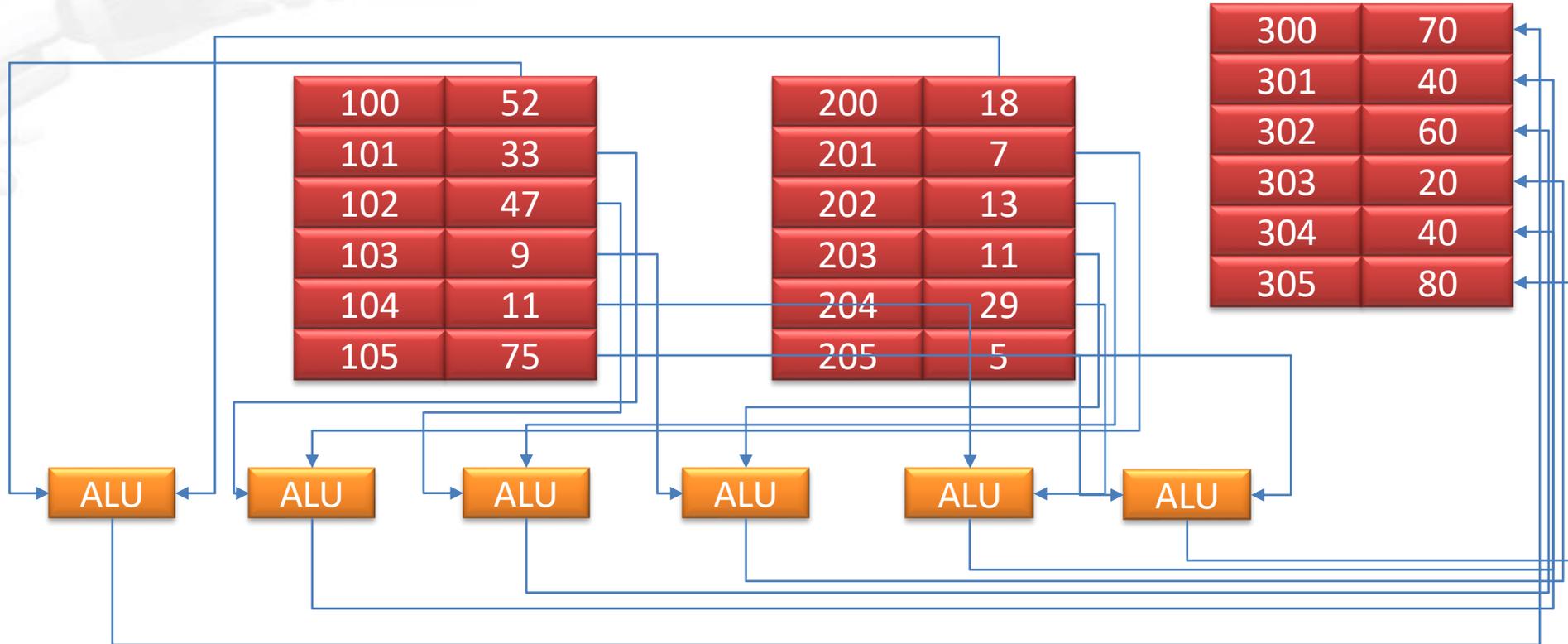
- ❑ Non riuscendo più ad aumentare significativamente le prestazioni della macchina (la frequenza di clock si attestò all'ordine di grandezza del GHz), le case di produzione decisero di puntare su architetture parallele
- ❑ Una **architettura parallela** (definita da George S. Almasi e Allan Gottlieb nel 1989 come: *“Insieme di elementi di elaborazione che cooperano e comunicano per risolvere velocemente problemi di dimensioni considerevoli, talvolta intrattabili su macchine sequenziali”*) è caratterizzata da **tre livelli di parallelismo**:
 - ❑ **Data Level Parallelism (DLP)**: i dati, su cui il programma lavora, sono distribuiti ed elaborati contemporaneamente da più processori
 - ❑ **Instruction Level Parallelism (ILP)**: le istruzioni, che compongono il programma, sono distribuite ed eseguite contemporaneamente da più processori.
 - ❑ **Thread Level Parallelism (TLP)**: i processi/thread sono eseguiti in parallelo da più processori



Elaboratore Elettronico

Parallelismo dati (DLP)

ADDVECT #300,#100,#200,6



Elaboratore Elettronico

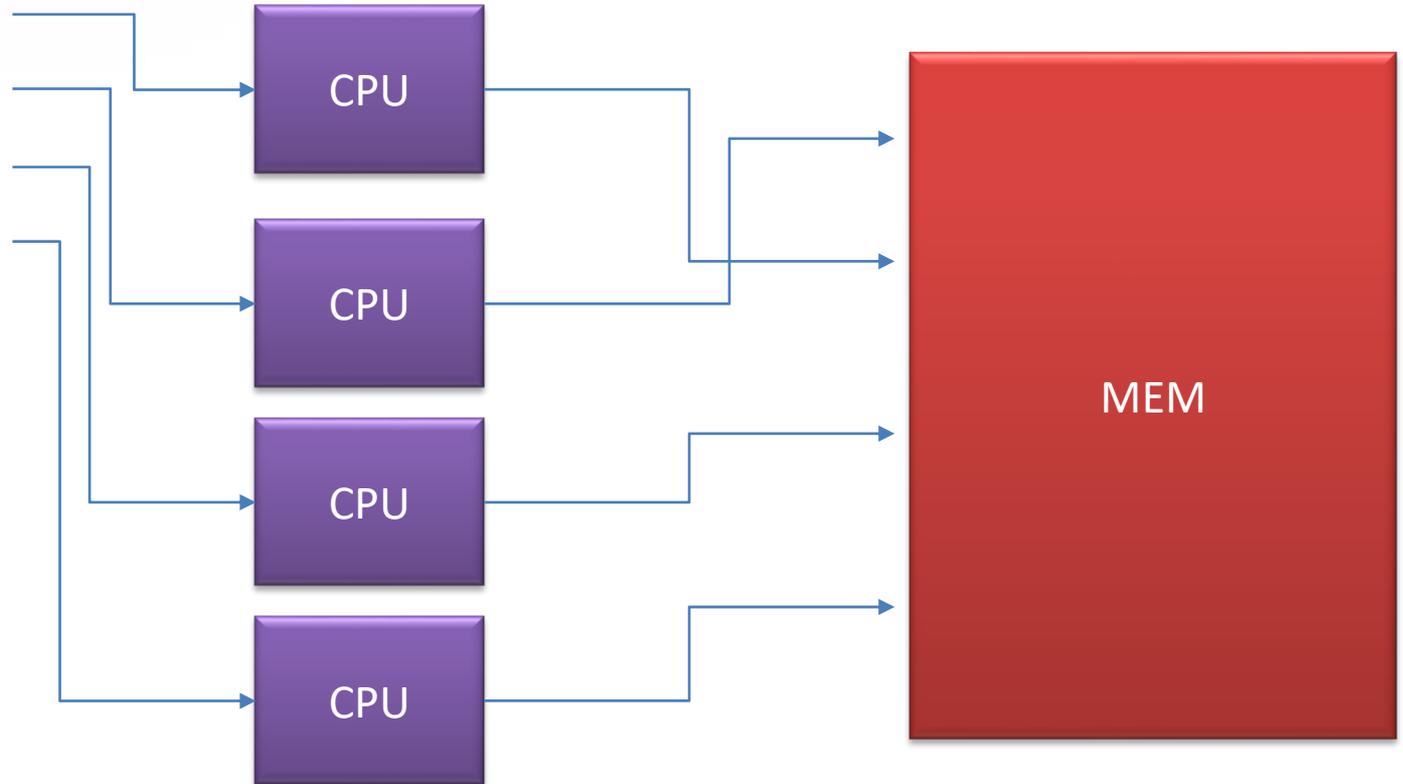
Parallelismo istruzioni (ILP)

ADD #300,#100,#200

MUL #100,#90,#80

DIV #500,#430,#110

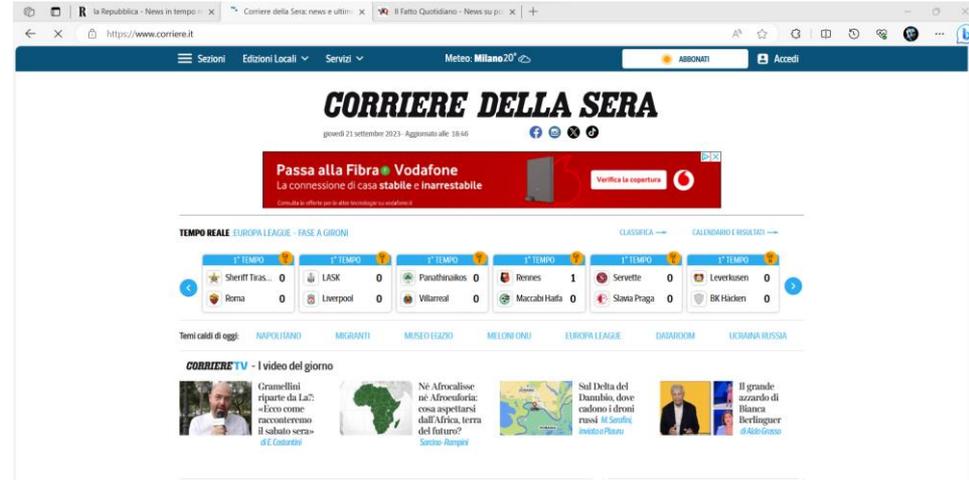
SUB #600,#550,#590



Elaboratore Elettronico

Tipi di parallelismo: thread

- ❑ Un **thread** (abbreviazione di *thread of execution*, filo dell'esecuzione) è un sottoprocesso (ovvero una parte del processo) che è eseguito in maniera concorrente ed indipendente dal processo stesso
- ❑ Il termine inglese rende bene l'idea, in quanto si rifà visivamente al concetto di fune composta da vari fili attorcigliati: se la fune è il processo in esecuzione, allora i singoli fili che la compongono sono i thread

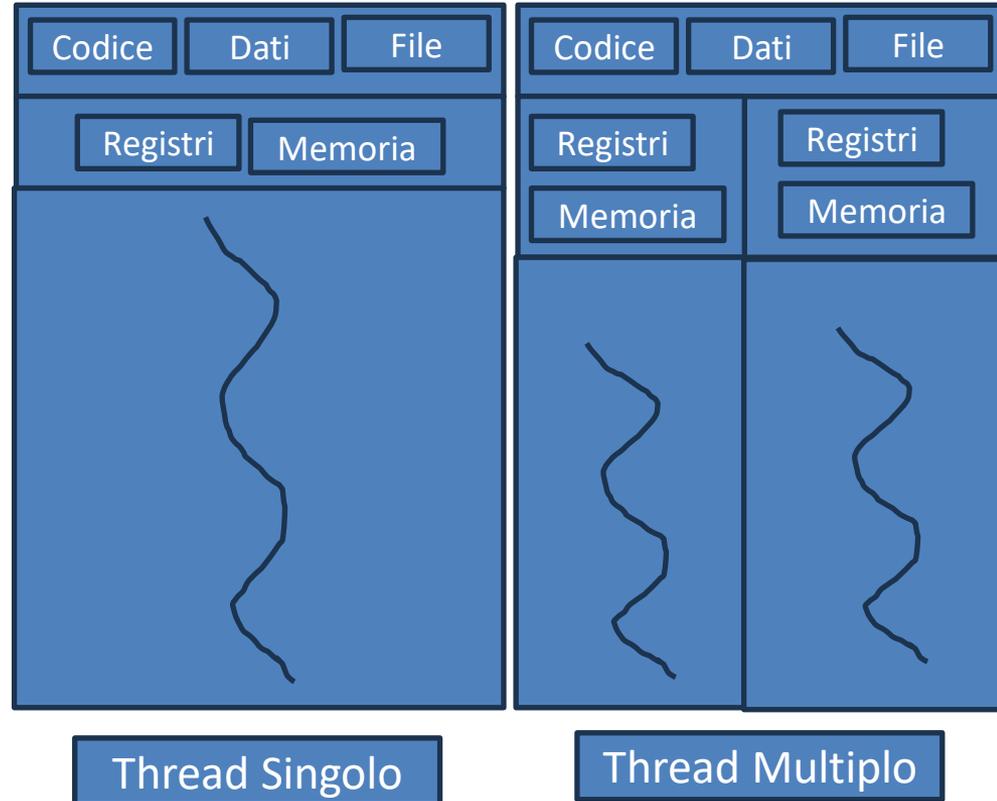


The screenshot shows the homepage of the Corriere della Sera website. At the top, there is a navigation bar with 'Sezioni', 'Edizioni Locali', and 'Servizi'. The main header features the newspaper's name 'CORRIERE DELLA SERA' and the date 'giovedì 21 settembre 2023'. Below the header is a red banner for 'Passa alla Fibra Vodafone'. The main content area displays a 'TEMPO REALE' section for the 'EUROPA LEAGUE - FASE A GRUPPI'. It shows a table of football matches with columns for '1° TEMPO' and '2° TEMPO'. The matches listed are: Sheriff Tiras. vs LASK, Roma vs Liverpool, Panathinaikos vs Villarreal, Rennes vs Maribor, Servette vs Sparta Praga, and Leverkusen vs BK Häcken. Below the table, there are sections for 'Temi caldi di oggi' and 'CORRIERE TV - I video del giorno'.

Elaboratore Elettronico

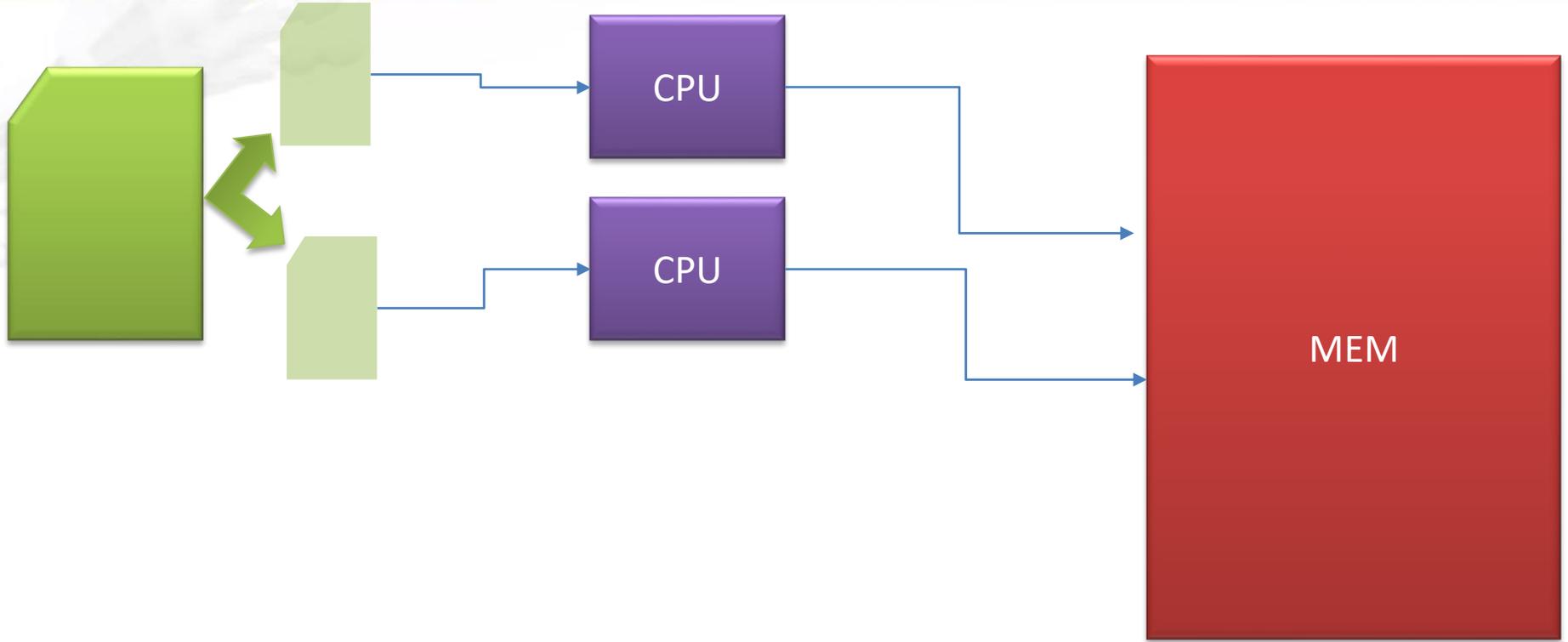
Tipi di parallelismo: thread

- ❑ Un processo ha sempre almeno un **thread** (se stesso), ma in alcuni casi un processo può avere più thread che vengono eseguiti in parallelo
- ❑ Una differenza sostanziale fra thread e processi consiste nel modo con cui essi condividono le risorse: mentre i processi sono di solito fra loro indipendenti, utilizzando diverse aree di memoria ed interagendo soltanto mediante appositi meccanismi di comunicazione messi a disposizione dal sistema, al contrario i thread di un processo tipicamente condividono le medesime informazioni di stato, la memoria ed altre risorse di sistema



Elaboratore Elettronico

Parallelismo dati (TLP)



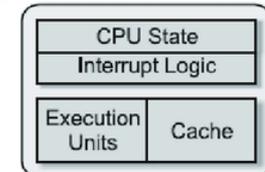
Elaboratore Elettronico

Multiprocessore - Multicore

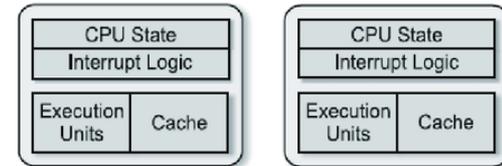
Il **parallelismo dei processi** ottenuto mediante più elaboratori, i **multi processori** (*multi processor*), o più CPU sullo stesso chip, i **multi nuclei** (*multi-core*)

Definizione. Il **multiprocessore** è una architettura con due o più CPU indipendenti che consente l'elaborazione simultanea dei programmi

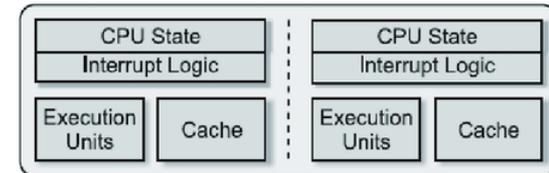
Definizione. Un **multicore** è un processore con due o più CPU simili ma indipendenti (chiamate *core*) che sono in grado di leggere ed eseguire le istruzioni di programma



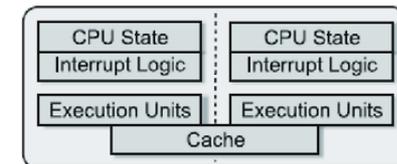
PROCESSORE SINGOLO



MULTIPROCESSORE



MULTICORE



MULTICORE A MEMORIA CONDIVISA

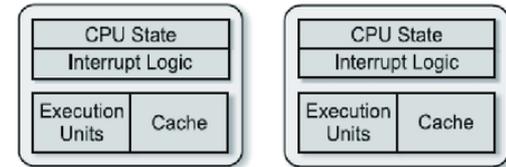
Elaboratore Elettronico

Multiprocessore – Multicore (metafora)

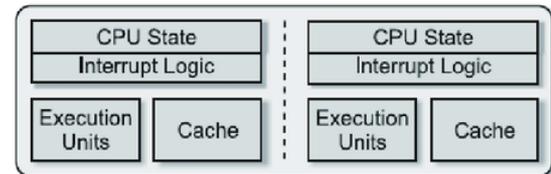
❑ Un processore **multicore** è paragonabile ad un sito produttivo con all'interno due o più operai (i core) addetti a costruire un certo prodotto (ovvero eseguire calcoli informatici richiesti)

❑ In un sistema **multiprocessore** i medesimi operai sono suddivisi in diversi siti produttivi, più o meno distanti tra loro

❑ La differenza sostanziale tra i due scenari è l'ottimizzazione dei "tempi morti" necessari per il trasporto del prodotto semilavorato da un sito produttivo all'altro (oltre ai pericoli di incidenti che possono esserci durante il trasferimento)



MULTIPROCESSORE



MULTICORE

Elaboratore Elettronico

Multiprocessore

❑ Tra il 2001 e il 2002, Sunnyvale presentò il **microprocessore Athlon XP**, per gli elaboratori desktop, e Athlon MP, per i server, entrambi dotati di più di un microprocessore per scheda madre e transistori di dimensione di 0.13 micrometri

❑ In generale un **sistema multiprocessore** offre:

- ☑ maggiore affidabilità rispetto ad uno multicore. Un multiprocessore ha più CPU. Il guasto in uno non influirà sulle altre CPU. È più tollerante ai difetti
- ☒ Più costoso di un multicore

ATHLON XP (Scheda madre)



ATHLON XP (CPU)



Elaboratore Elettronico

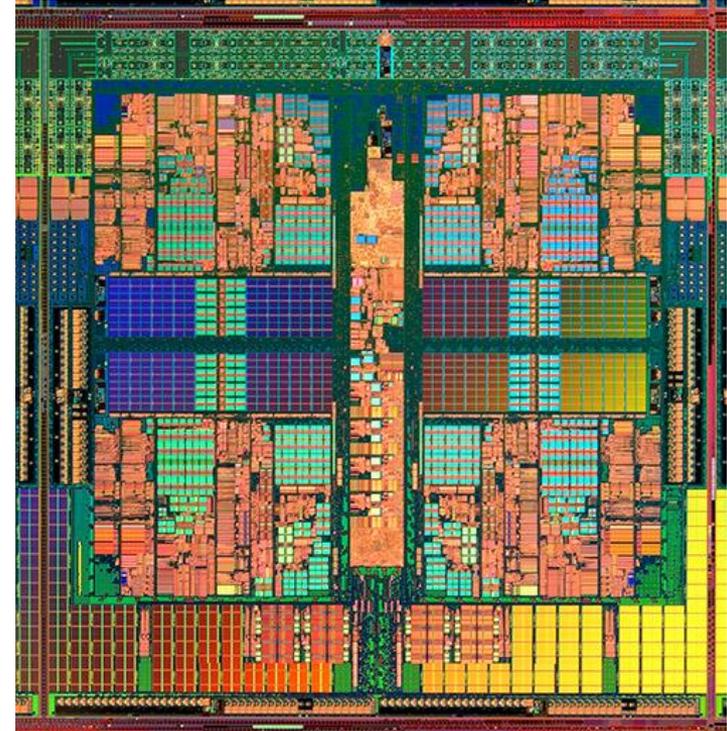
Multicore

Tra il 2001 e il 2002, Intel propose il dual core **Pentium D**

In generale **un sistema multicore** offre:

- ha meno traffico dati rispetto a un multiprocessore perché tutti i core sono integrati in un singolo chip
- Più economico di un multiprocessore
- Meno tollerante ai guasti rispetto ad un multiprocessore
- Mediamente più lento rispetto ad un multiprocessore

INTEL CORE 2 QUAD



Elaboratore Elettronico

Multicore moderni: Intel i9-i7-i5

	Core i9-9900K	Core i7-9700K	Core i5-9600K
Socket	LGA1151(v2)		
Cores/Threads	8 cores, 16 threads	8 cores, 8 threads	6 cores, 6 threads
Clock Speed	3.60 GHz	3.60 GHz	3.70 GHz
Boost Clock	5.00 GHz	4.90 GHz	4.60 GHz
L2 Cache/core	256 KB	256 KB	256 KB
L3 Cache	16 MB, shared	12 MB, shared	9 MB, shared
Memory Channels	Dual-channel	Dual-channel	Dual-channel
PCIe Lanes	16	16	16
TDP	95W	95W	95W

Elaboratore Elettronico

Scheda madre

Definizione. Una **scheda madre** (*main board o mother board*) è un pannello in grado di ospitare i circuiti elettronici e i collegamenti utili per interconnettere e far funzionare i componenti interni principali di un elaboratore elettronico
Tra i componenti ci sono l'alloggiamento del processore e della memoria, i bus di espansione, le interfacce verso le periferiche esterne (mouse, tastiera, USB), la batteria di sistema, la ventola di raffreddamento, la memoria per il firmware



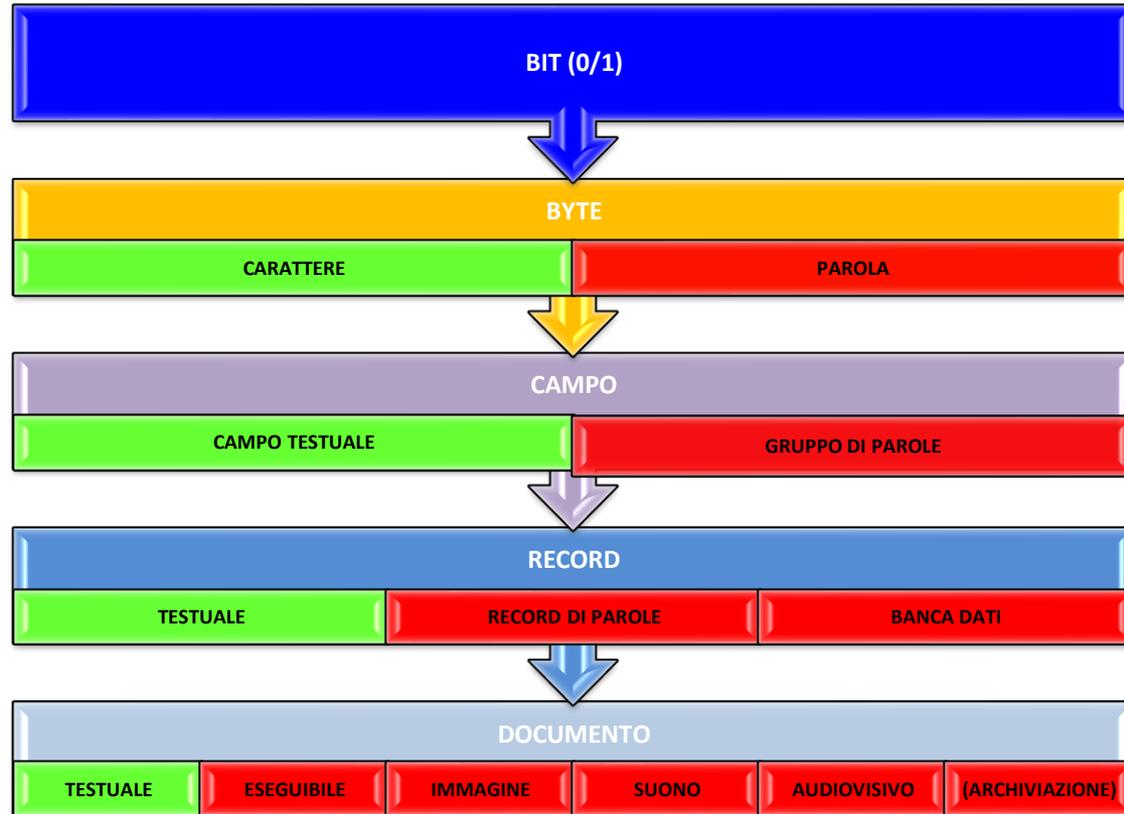


**L'informazione gestita negli
elaboratori elettronici**

Elaboratore Elettronico

Documento digitale: file

- ❑ I primi elaboratori operavano manipolando numeri (naturali ed interi)
- ❑ Le istruzioni ed i numeri sulle quali dovevano essere applicate erano prelevati dalle periferiche (tessera perforata, nastro magnetico, disco magnetico) e inviati in Memoria Centrale per essere elaborati
- ❑ Con il passare del tempo si gestirono messaggi più articolati tanto da definire una gerarchia dell'informazione digitale



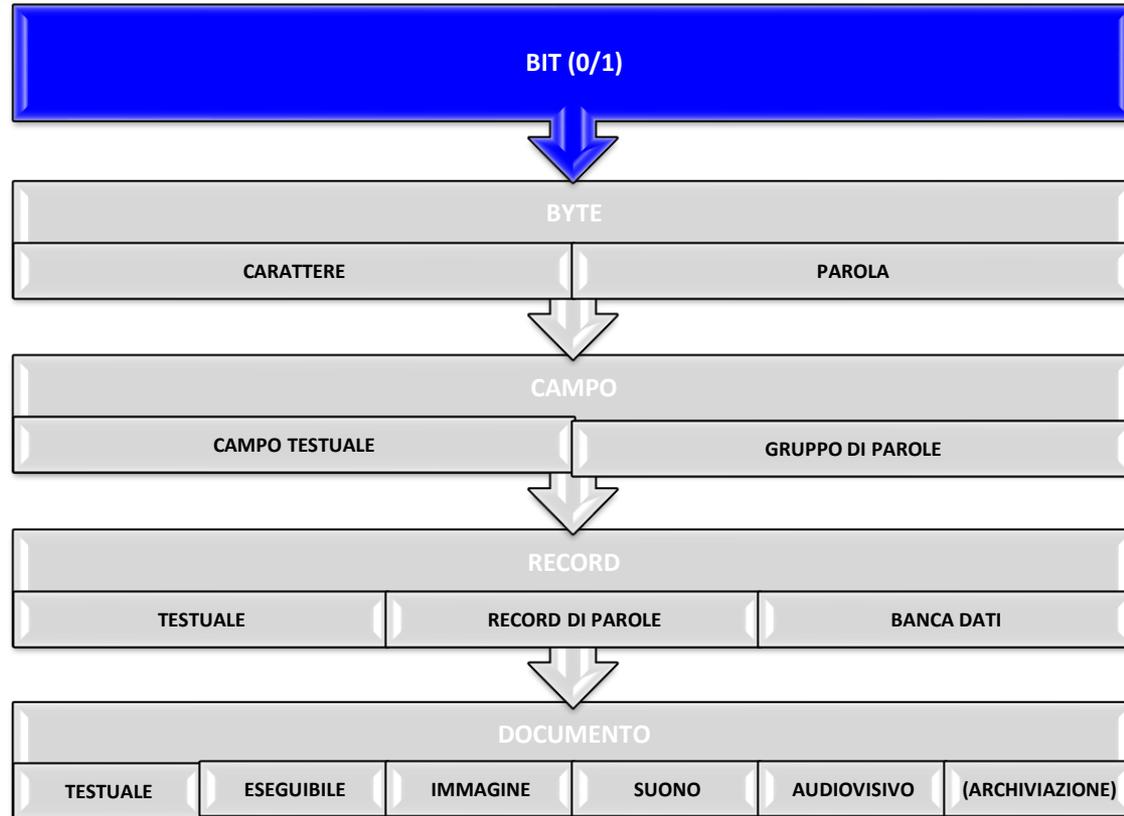
Elaboratore Elettronico

Documento digitale: file

❑ Il **bit** è la più piccola unità di rappresentazione di informazione digitale ed assume il valore 0 o 1

❑ Otto bit formano un **byte** ed è la quantità minima standard che un elaboratore elettronico è in grado di gestire
Altre grandezze utili sono:

Quantità	Simbolo	Nome
0/1	b	Bit
8 bit	B	Byte
1024 byte	KB	Kilobyte
1048576 byte	MB	Megabyte
1073741824byte	1GB	Gigabyte
1099511628000byte	1TB	Terabyte

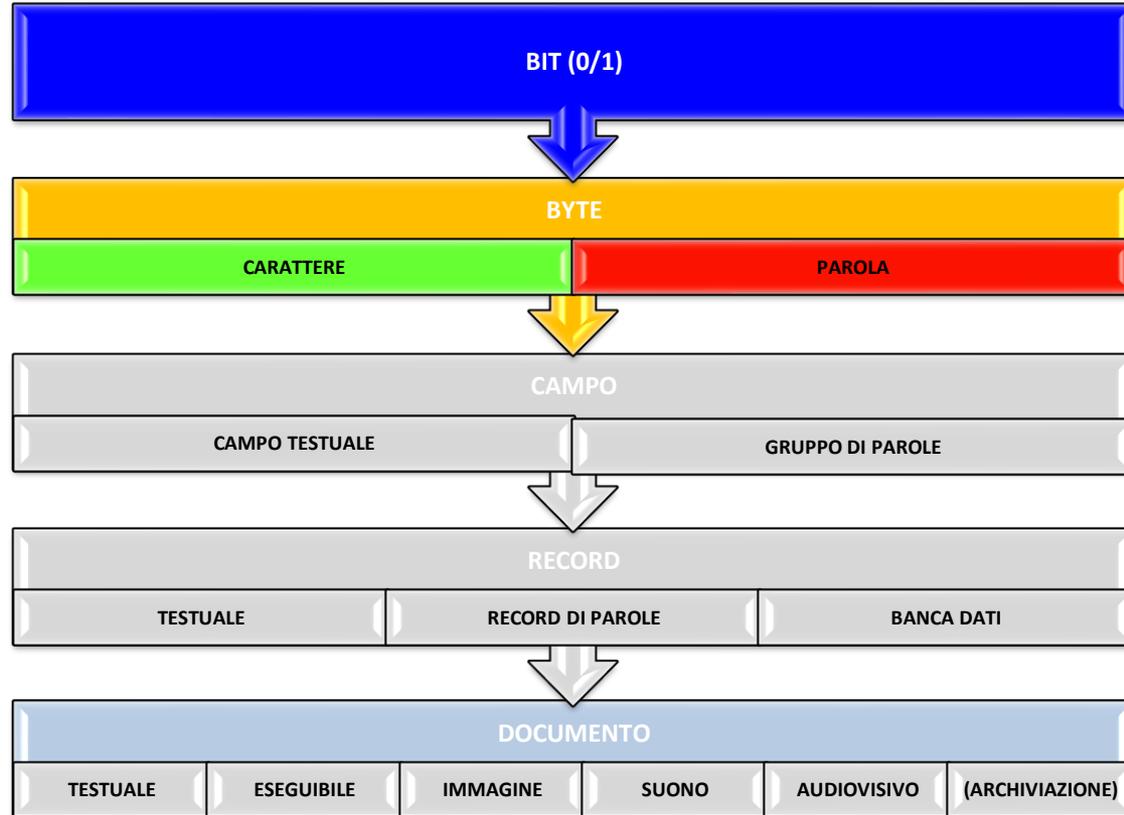


Elaboratore Elettronico

Documento digitale: file

❑ Un byte al quale è associato un segno testuale prende il nome di **carattere** (character). Un esempio è il valore numerico 102 espresso in un byte, cioè $(01100110)_2$, che rappresenta la lettera 'f' nel codice testuale ASCII

❑ Un byte che non ha una corrispondenza testuale è una **parola** il cui significato varia in relazione al dominio di applicazione: un operando, nel campo numerico; un colore nel caso di immagini digitali, un campione di un'onda sonora per un suono elettronico

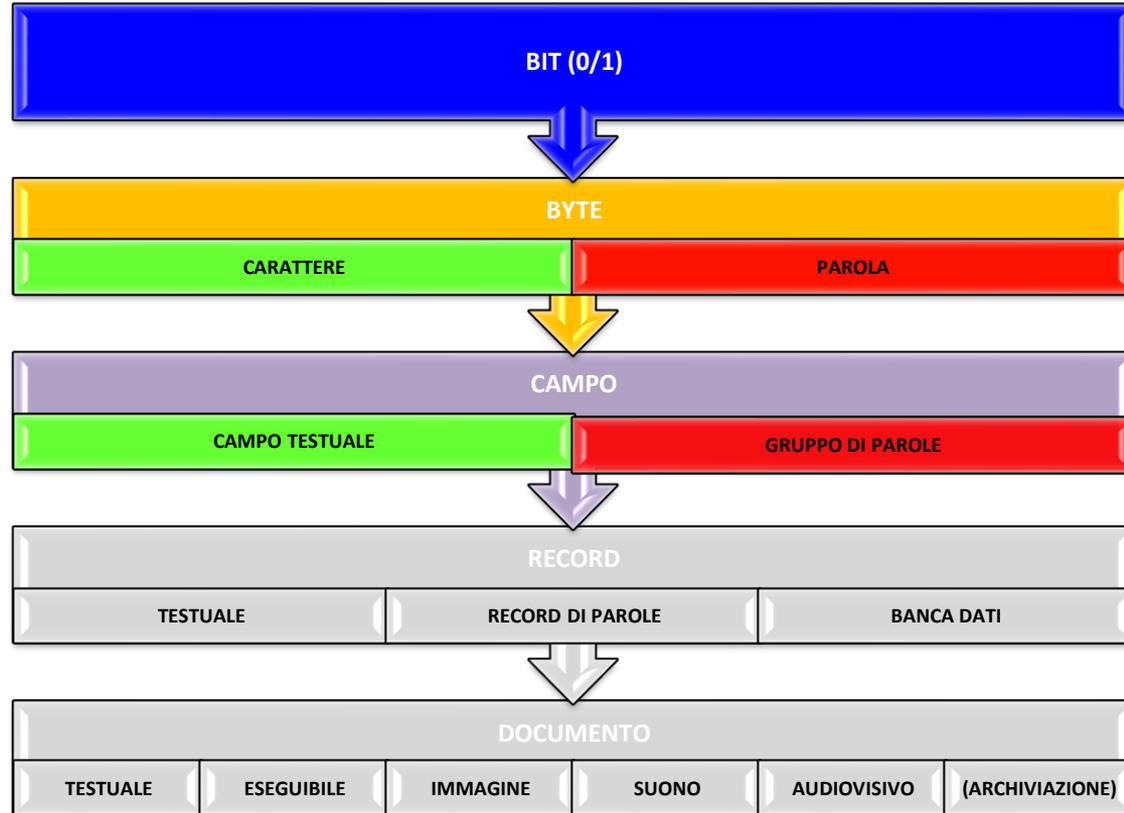
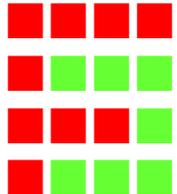


Elaboratore Elettronico

Documento digitale: file

Il **campo** è un raggruppamento di caratteri o di parole
Ad esempio, il campo “1947-05-13”
è associabile alla data di nascita di
una persona

Il gruppo di parole
AA0000 AA0000 AA0000 AA0000
A0000 00FF00 00FF00 00FF00 AA0000
AA0000 AA0000 00FF00 AA0000
00FF00 00FF00 00FF00
può identificare una immagine che
rappresenta la lettera F

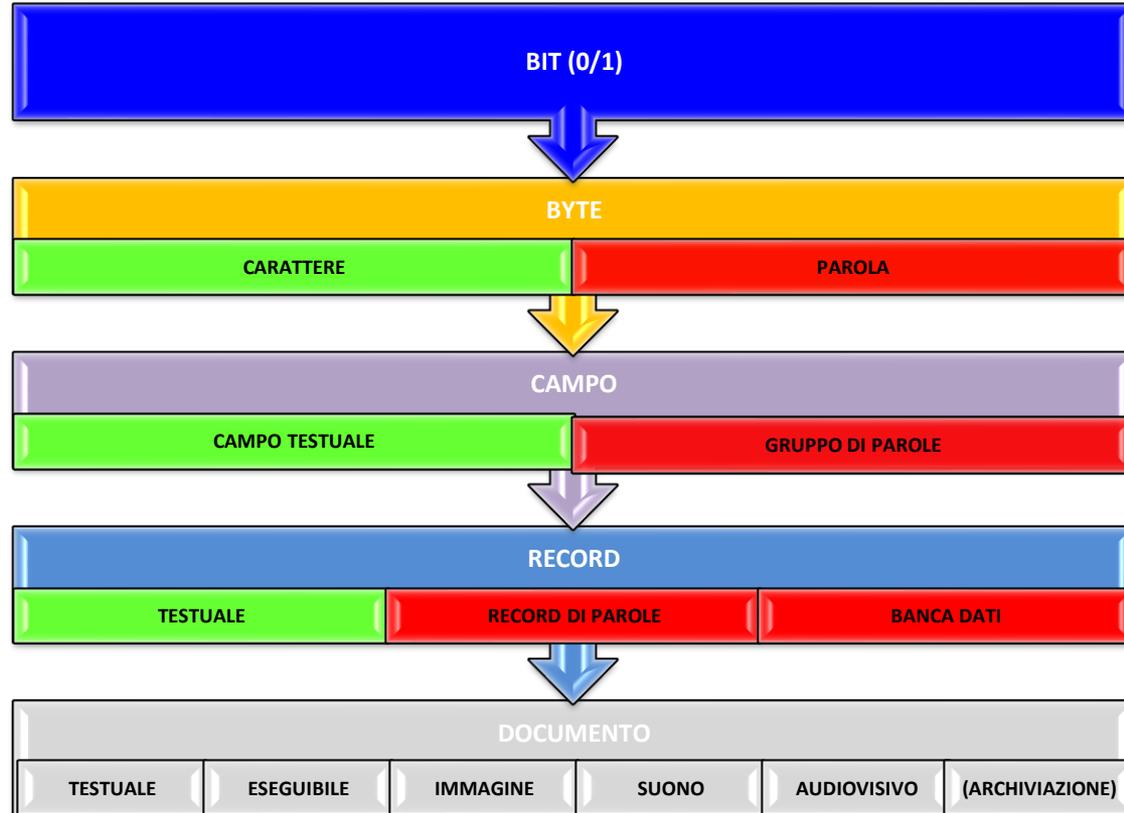


Elaboratore Elettronico

Documento digitale: file

□ Il **record** è una collezione di campi

Ad esempio il record LIBRO è composto dai campi TITOLO, AUTORE, DATA, EDITORE
E una istanza può essere "Il Codice da Vinci" "Dan Brown" "2000" "Random House "



Elaboratore Elettronico

Documento digitale: file

- Più record con relazioni tra di loro formano una tabella di una **banca dati**

STUDENTE

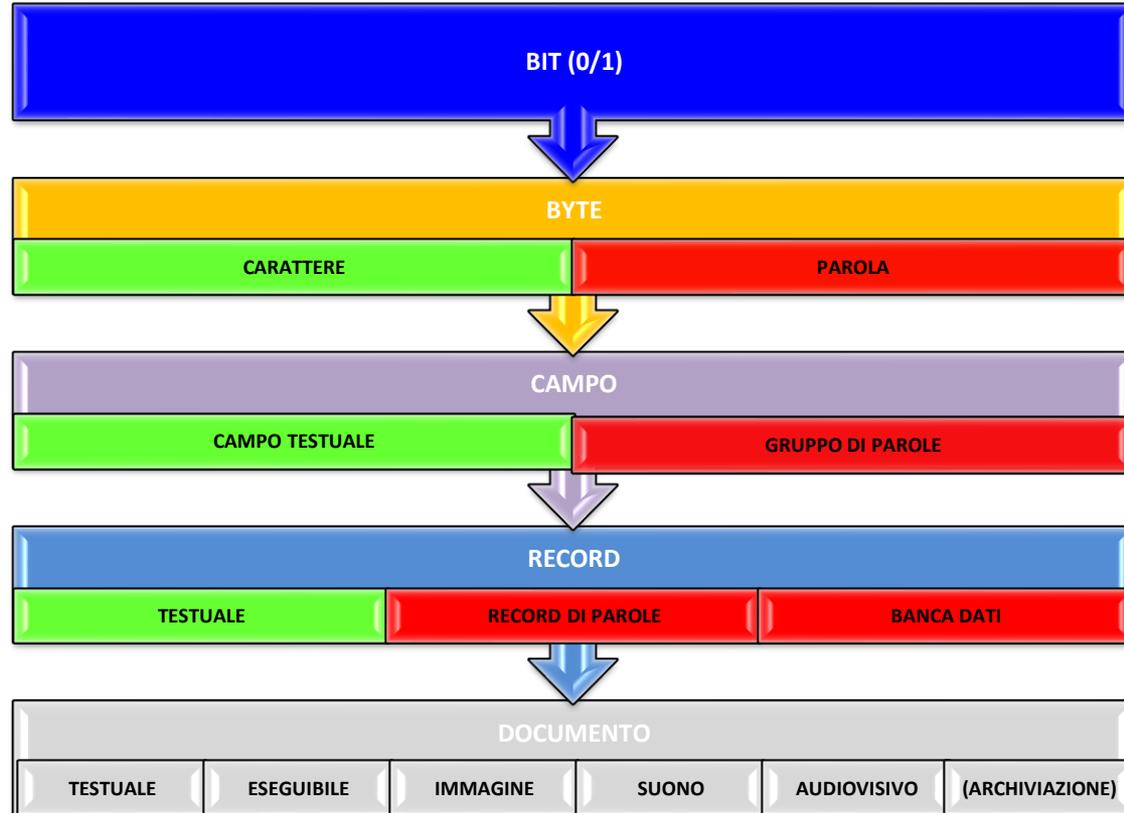
Matricola, Nome, Cognome,
Indirizzo, Città, Facoltà

ESAMI STUDENTE

Matricola, Id_Corso, Voto

CORSO

CodiceCorso, Nome, Sinossi,
Docente

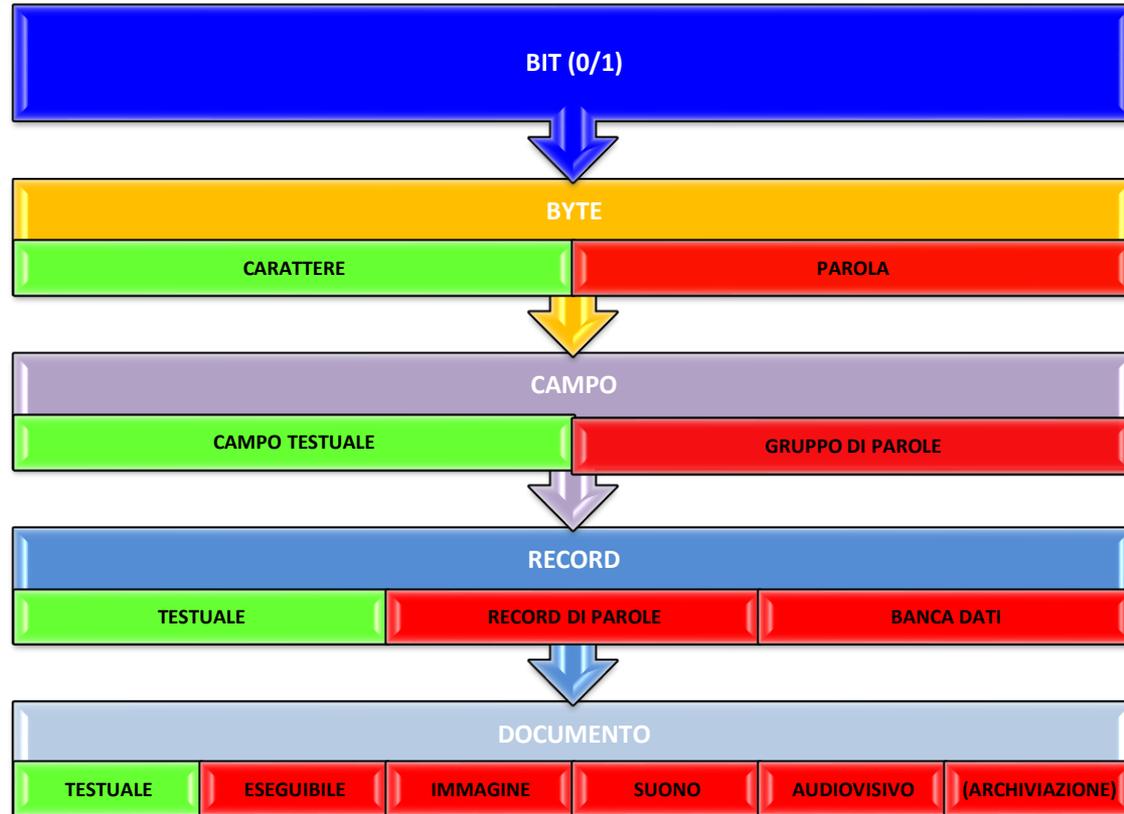


Elaboratore Elettronico

Documento digitale: file

❑ Un **documento digitale** (o file) è una collezione di record omogeni, correlati e con un ordine prestabilito

❑ Un documento digitale in informatica ha due caratterizzazioni: **testuale** e **binario**. Nel primo caso si tratta di una collezione di record testuali; nel secondo è un insieme di record avente un significato (numerico, immagine, suono, audiovisivo, multimediale) esplicitato mediante campi aggiuntivi essenziali (metadati tecnici)

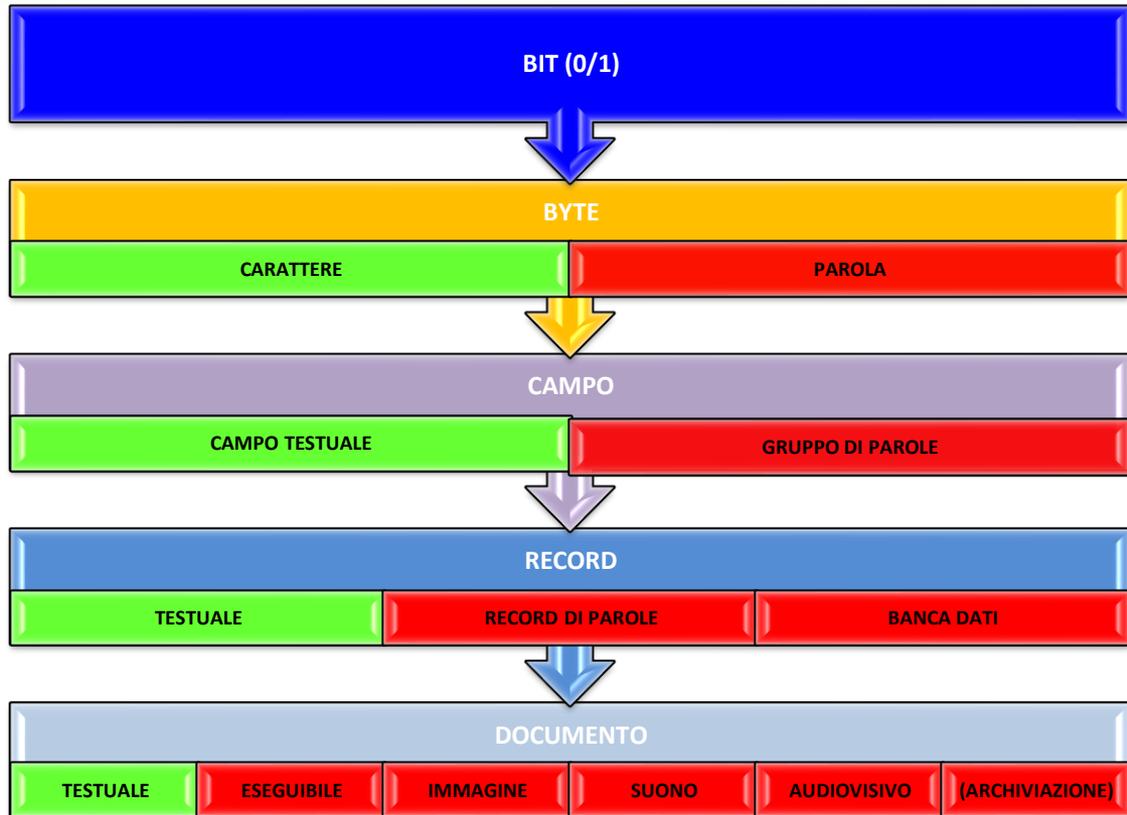


Elaboratore Elettronico

Documento digitale: file

- ❑ Il record binario necessita di informazioni ausiliarie (metadati tecnici) per stabilire la classe di appartenenza del documento (immagine, video, audio) e ne consente la riproduzione con il corretto applicativo
- ❑ La totalità dei record e i metadati tecnici costituiscono il **formato** del documento

Formato	Nome	Tipo di parola
Immagine	JPEG	Binario
Audio	MP3	Binario
Video	MP4	Binario
Testo	TXT	Testuale
Testo con marcatori	HTML	Testuale



Elaboratore Elettronico

Documento digitale: file (esempio formato immagine BMP)



Altezza x Larghezza

780 x 749

30C x 2ED

https://hexed.it

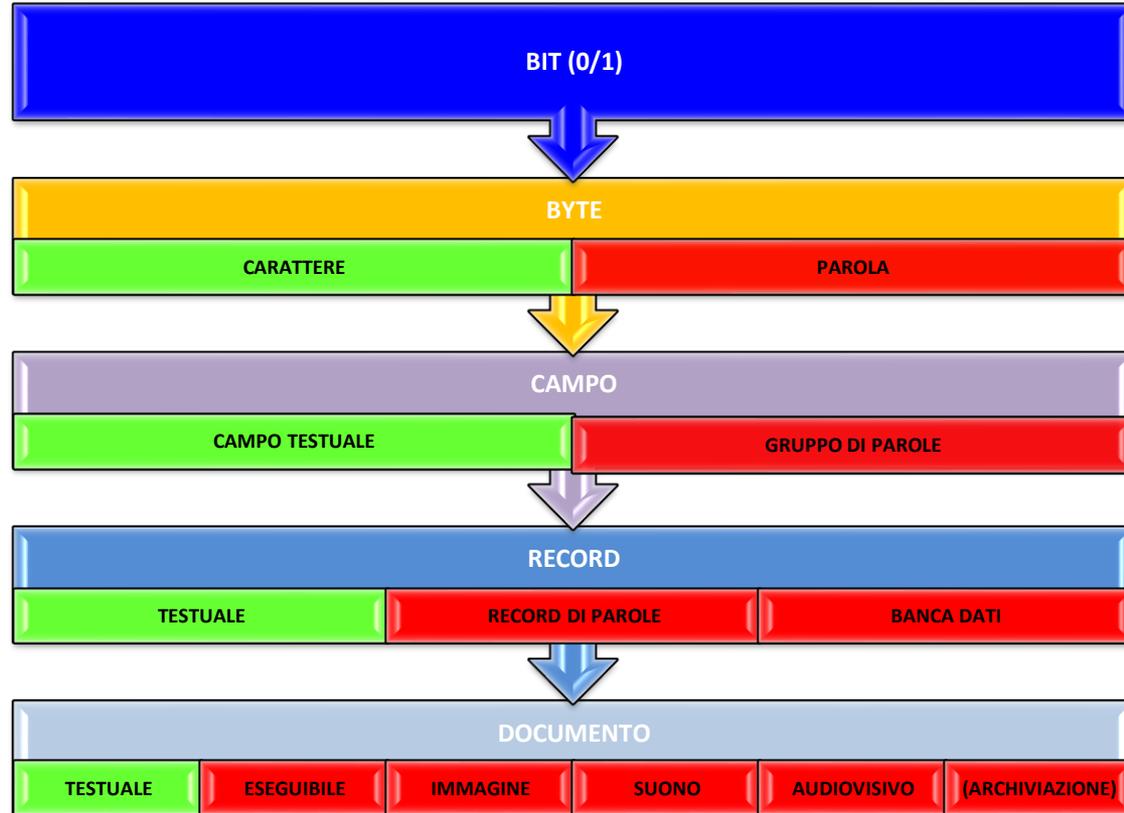
Nuovo file Apri file Salva con nome Annulla Ripeti Strumenti Traduci Impostazioni Aiuto

Informazioni File		-Senza Titolo-	Viterbo.jpg x	Viterbo.bmp x
Nome File	Viterbo.bmp	00000000	42 4D B6 A8 23 00 00 00	00 00 46 00 00 00 38 00
Dim. File	2.336.950 Bytes (2.283 KiB)	00000010	00 00 0C 03 00 00	ED 02 30 00 01 00 20 00 03 00
Ispettore dati (Little-endian)		00000020	00 00 70 A8 23 00 13 0B	00 00 13 0B 00 00 00 00
		00000030	00 00 00 00 00 00 00 00	FF 00 00 FF 00 00 FF 00
Tipo		00000040	00 00 00 00 00 FF 44 5A	5F FF 56 6C 71 FF 60 7D
		00000050	82 FF 7C 99 9E FF 85 A2	A9 FF 7D 99 A0 FF 6E 88
Intero a 8-bit	66	00000060	8F FF 75 8F 95 FF 6E 8B	8F FF 72 91 92 FF 51 67
Intero a 16-bit	19778	00000070	6C FF 3F 51 58 FF 56 6C	72 FF 55 6F 76 FF 5D 72
Intero a 24-bit	11947330	00000080	7A FF 54 6B 73 FF 5D 78	82 FF 61 77 82 FF 58 7B
Intero a 32-bit	2830519618	00000090	7F FF 5E 7C 81 FF 51 6B	71 FF 55 70 74 FF 52 71
Intero a 64-bit (+)	153154374978	000000A0	74 FF 4D 70 74 FF 4D 73	77 FF 59 82 85 FF 61 87
Intero a 64-bit (±)	153154374978	000000B0	8C FF 65 85 8B FF 48 62	68 FF 34 48 4D FF 2C 3D
Virg. mob. a 16-bit	21,03125	000000C0	40 FF 27 36 39 FF 29 38	3A FF 2B 3A 3C FF 25 2E
Virg. mob. a 32-bit	-2,0239564e-14	000000D0	32 FF 1E 2A 2E FF 2C 3B	3C FF 2B 3A 3D FF 2A 39
Virg. mob. 64-bit	7,5668315186918019e-313	000000E0	3C FF 3D 4B 51 FF 24 37	3E FF 3D 50 57 FF 81 A0
LEB128 (+)	66	000000F0	A9 FF 94 B0 B7 FF 93 AB	B1 FF 9A B0 B6 FF 94 AC
LEB128 (±)	-62	00000100	B2 FF 92 AA B0 FF 8B A2	AA FF 82 97 9F FF 81 94
Data/Ora MS-DOS	22/05/2064 09:42:04 Local	00000110	9C FF 83 96 9E FF 7C 8F	9E FF 75 87 8E FF 84 96
Data/Ora OLE 2.0	30/12/1899 00:00:00.000 UTC	00000120	9D FF 82 94 9B FF 85 97	9E FF 75 87 8E FF 5A 66
Data/Ora UNIX	11/09/2059 15:26:58 UTC	00000130	6C FF 38 44 4A FF 36 44	4A FF 33 40 48 FF 36 43
Data/Ora Macintosh HFS	10/09/1993 17:26:58 Local	00000140	4B FF 34 42 48 FF 32 42	4E FF 3A 4A 50 FF 35 49
Data/Ora Macintosh HFS+	10/09/1993 15:26:58 UTC	00000150	4E FF 39 4C 51 FF 38 48	4E FF 36 44 4A FF 38 46
UTF-8 Character	B	00000160	4C FF 33 40 48 FF 3A 47	4F FF 3C 49 51 FF 25 38
Binario	<input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	00000170	3F FF 44 5A 60 FF 5E 77	7B FF 6A 84 8A FF 80 9D
		00000180	A2 FF CC D6 FF AA C4	D2 FF 88 A2 B2 FF 84 99
		00000190	A8 FF 86 9E AA FF 8F AA	B4 FF 93 AF B6 FF 88 A2
		000001A0	A8 FF 84 9C A2 FF 84 9C	A2 FF 85 9D A3 FF 85 9D
		000001B0	A3 FF 83 9B A1 FF 7D 95	9B FF 7A 90 96 FF 71 84
		000001C0	8B FF 7D 90 97 FF 76 88	8F FF 6B 7D 84 FF 75 88
		000001D0	8D FF 7C 8F 94 FF 55 68	6D FF 35 48 4D FF 37 47
		000001E0	4D FF 34 44 4A FF 35 43	49 FF 35 43 49 FF 39 49

Elaboratore Elettronico

Documento digitale: file

- ❑ Nella tipologia binaria si distingue il **documento eseguibile** (*executable file*), cioè una sequenza di record che hanno il significato di istruzioni, operandi e indirizzi e che concretizzano un applicativo informatico, cioè un programma eseguibile da un elaboratore elettronico
- ❑ Gli applicativi informatici sono realizzati non solo per l'elaborazione numerica ma anche per la creazione, la visualizzazione, l'elaborazione e il salvataggio dei documenti digitali





Fine