

# L'architettura dei sistemi informatici

---

*Fondamenti di Informatica T*

INFORMATICA:

**"Scienza della rappresentazione e dell'elaborazione automatica dell'informazione."**

ELABORATORE ELETTRONICO ("COMPUTER")

**È lo strumento per l'elaborazione automatica delle informazioni.**

**Le informazioni devono essere opportunamente rappresentate affinché l'elaboratore possa manipolarle.**

---

*Fondamenti di Informatica T*

# Rappresentazione delle Informazioni: i Numeri Naturali

---

*Fondamenti di Informatica T*

## Rappresentazione in base $p$

**Metodo posizionale:** ogni cifra ha un *peso*

Esempio:  $123 = 100 + 20 + 3$

- Di solito noi usiamo la *base* decimale
- Un numero generico di  $m$  cifre è rappresentato in base  $p$  dalla sequenza:  $a_n, a_{n-1}, a_{n-2}, \dots, a_0$

$a_n$  : cifra più significativa

$a_0$  : cifra meno significativa

$n = m-1$

$a_i \in \{0, 1, \dots, p-1\}$

---

*Fondamenti di Informatica T*

## Rappresentazione in base $p$

- Un numero naturale  $N$ , composto da  $m$  cifre, in base  $p$ , si esprime come:

$$N_p = a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p^1 + a_0 \cdot p^0 = \sum_{i=0}^n a_i \cdot p^i$$

- Esempio in base decimale ( $p=10$ ):  
 $587_{10} = 7 \cdot 10^0 + 8 \cdot 10^1 + 5 \cdot 10^2$
- Posso rappresentare i numeri nell'intervallo discreto:  
 $[0, p^m - 1]$

---

Fondamenti di Informatica T

## Rappresentazione in base due

- Base binaria:  $p=2$ ; cifre  $a_i \in \{0, 1\}$  chiamate *bit* (*binary digit*)
- Otto bit sono chiamati *byte*

**Esempio**, con  $m=5$ :

$$11011_2 = (1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0)_{10} = 27_{10}$$

- Posso rappresentare i numeri nell'intervallo discreto:  
 $[0, 2^m - 1]$
- Esempio con  $m=8$ :  
rappresento numeri binari:  $[00000000_2, 11111111_2]$ ,  
ovvero:  $[0, 255]$

---

Fondamenti di Informatica T

## Conversioni di base

- Per convertire da base due a base 10:
  - Usare la sommatoria illustrata nella slide precedente
- Per convertire da base dieci a base due:
  - Metodo delle divisioni successive:

---

Fondamenti di Informatica T

### Metodo delle divisioni successive

Esempio: Esprimere in base 2 il numero naturale  $215_{10}$

operazione	quoziente	resto	
215:2	53	1	$a_0$
53:2	26	1	$a_1$
26:2	13	0	$a_2$
13:2	6	1	$a_3$
6:2	3	1	$a_4$
3:2	1	1	$a_5$
1:2	0	1	$a_6$

$$\rightarrow 215_{10} = (1111011)_2$$

---

Fondamenti di Informatica T



## Sottrazione

- Le cifre sono 0 e 1 ed il prestito può essere solo 1

Prestito precedente	differenza	Risultato	Prestito
0	0 - 0	0	0
0	1 - 0	1	0
0	1 - 1	0	0
0	0 - 1	1	1
1	0 - 0	1	1
1	1 - 0	0	0
1	1 - 1	1	1
1	0 - 1	0	1

---

Fondamenti di Informatica T

## Basi ottale ed esadecimale

- Base ottale:  $p=8$ ;  $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$** 
  - Esempio:  $234_8 = (2 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0)_{10} = 156_{10}$
- Base esadecimale:  $p=16$ ;**  
 $a_i \in \{0, 1, 2, \dots, 9, A, B, C, D, E, F\}$ 
  - Esempio:  $B7F_{16} = (11 \cdot 16^2 + 7 \cdot 16^1 + 15 \cdot 16^0)_{10} = 2943_{10}$
  - Notare: "11" al posto di "B" e "15" al posto di "F", i loro equivalenti in base dieci
- Conversione dalla base 2 alla base 8 o 16: si considerano gruppi di tre o 4 cifre binarie
- Es: 001 010 110 111 si traduce ogni tripla nella corrispondente cifra ottale

---

Fondamenti di Informatica T

# Numeri Interi

---

*Fondamenti di Informatica T*

## Modulo e segno

- Uso un bit per codificare e memorizzare il segno: "1" significa numero negativo, "0" numero positivo. Esempio  $m=3$ :

Num. intero, base 10	Num. intero, base due, modulo e segno
-3	111
-2	110
-1	101
-0	100
+0	000
+1	001
+2	010
+3	011

---

*Fondamenti di Informatica T*

## NUMERI INTERI (con segno)

---

### Rappresentazione *in modulo e segno*

#### Due difetti principali:

- **occorrono *algoritmi speciali* per fare le operazioni**

- se si adottano le usuali regole, non è verificata la proprietà  $X + (-X) = 0$
- **occorrono regole (e quindi circuiti) ad hoc**

- ***due diverse rappresentazioni per lo zero***

$$\begin{array}{l} + 0 = 00000000 \\ \quad \quad \quad 10000000 \end{array} \qquad \begin{array}{l} - 0 = \\ \quad \quad \quad 00000000 \end{array}$$

Ad esempio:

$$\begin{array}{r} (+5) + (-5) = -10 ??? \\ +5 \quad 0 \ 0000101 \\ -5 \quad 1 \ 0000101 \\ \hline 0 \quad 1 \ 0001010 \end{array}$$

## NUMERI INTERI (con segno)

---

### Due possibilità:

- ***rappresentazione in modulo e segno (come quella usata negli esercizi di questo corso)***
  - semplice e intuitiva
  - ma inefficiente e complessa nella gestione delle operazioni → non molto usata in pratica
- ***rappresentazione in complemento a due***
  - meno intuitiva, costruita "ad hoc"
  - ma efficiente e capace di rendere semplice la gestione delle operazioni → largamente usata nelle architetture reali di CPU

## NUMERI INTERI (con segno)

---

### Rappresentazione in complemento a due

- *si vogliono poter usare le regole standard per fare le operazioni*
- in particolare, si vuole che
  - $X + (-X) = 0$
  - la rappresentazione dello zero sia *unica*
- *anche a prezzo di una notazione più complessa, meno intuitiva*

---

Fondamenti di Informatica T

## Rappresentazioni dei Numeri

---

- Il bit più significativo rappresenta il segno del numero: 0 per positivi, 1 per negativi
- La rappresentazione di un numero positivo si ottiene codificando il valore assoluto del numero con i bit restanti
- La rappresentazione di un numero negativo si ottiene in tre passi:
  - Si rappresenta in complemento a due il numero positivo con lo stesso valore assoluto del numero negativo da codificare
  - Si invertono tutti i bit
  - Si somma 1 al risultato ottenuto al passo precedente

---

Fondamenti di Informatica T

## Quindi:

---

- per calcolare il numero negativo  $-|v|$  occorre:
- ***prima invertire tutti i bit*** della rappresentazione di  $|v|$  ***poi aggiungere 1*** al risultato

---

Fondamenti di Informatica T

## Esempio

---

- Esempio con 4 bit a disposizione
- +5 -> 0101
- -5 -> :
  - La rappresentazione del complemento a due di +5 è 0101
  - Invertendo tutti i bit si ottiene 1010
  - Sommando 1 si ottiene 1011 -> -5

---

Fondamenti di Informatica T

## Ottenere un numero con segno data la rappresentazione in CPL<sub>2</sub>

---

- Se il primo bit è 0 il numero è positivo: per calcolarne il valore assoluto si esegue la conversione da binario a decimale
- Se il primo bit è 1 il numero è negativo:
  - Si ignora il primo bit
  - Si invertono i bit restanti
  - Si converte da binario a decimale
  - Si somma 1 al numero così ottenuto per ottenere il valore assoluto del numero negativo
  - Es. 1011. si esclude il primo bit. Invertendo 011 si ottiene 100 che è la codifica di 4, si aggiunge 1, quindi si ottiene il valore assoluto 5, il risultato è -5

---

Fondamenti di Informatica T

## Complemento a due (CPL<sub>2</sub>)

- Usando  $m$  bit:  $(-N)_{CPL_2} = (2^m - N_{10})_2$
- Esempio ( $m=3$ ):  $(-N)_{CPL_2} = (2^3 - N_{10})_2$

Num. intero base 10	Trasformazione	Num. intero, base 2, CPL <sub>2</sub> , $m=3$
-4	$8 - 4 = 4$	$4_{10} = 100$
-3	$8 - 3 = 5$	$5_{10} = 101$
-2	$8 - 2 = 6$	$6_{10} = 110$
-1	$8 - 1 = 7$	$7_{10} = 111$
0	nessuna	$0_{10} = 000$
1	nessuna	$1_{10} = 001$
2	nessuna	$2_{10} = 010$
3	nessuna	$3_{10} = 011$

---

Fondamenti di Informatica T

## Complemento a due (CPL<sub>2</sub>)

- Posso rappresentare i numeri nell'intervallo discreto:  
 $[-2^{m-1}, 2^{m-1} - 1]$ 
  - Asimmetria tra negativi e positivi
  - Esempio ( $m=8$ ):  $[-128, +127]$ , perché  $-2^7 = -128$  e  $2^7 - 1 = +127$
- Tutti i numeri negativi cominciano con il bit più significativo posto a "1", mentre tutti i positivi e lo zero iniziano con uno "0"

---

Fondamenti di Informatica T

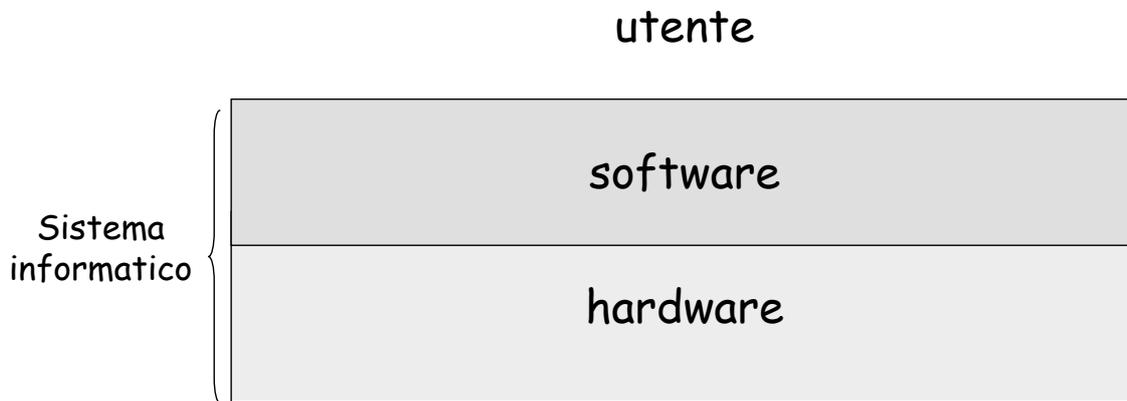
## Somma e sottrazione in CPL<sub>2</sub>

- **Somma:** come per i naturali
- **Sottrazione:**  $N_1 - N_2 = N_1 + (-N_2)_{\text{CPL}_2}$
- **Carry:**
  - Il carry non viene considerato!
- **Overflow:**
  - Se, sommando due interi di  $m$  bit dotati di segno concorde, ottengo un risultato di segno discorde (sempre considerando  $m$  bit), allora si ha un *overflow* (il risultato non è codificabile su  $m$  bit) e l'operazione è errata
  - L'overflow non può verificarsi se gli operandi sono di segno discorde

---

Fondamenti di Informatica T

# Sistema Informatico



---

*Fondamenti di Informatica T*

## HARDWARE

**Insieme delle componenti fisiche.**

E' composto da un insieme di *unità funzionali*:

- **CPU (central processing unit):** è il dispositivo che esegue l'elaborazione sulle informazioni e coordina/controlla tutte le unità funzionali
- **Memoria centrale:** contiene le informazioni (dati e istruzioni) necessarie alla cpu per la sua attività.
- **Periferiche:** dispositivi per la comunicazione tra il computer e l'ambiente esterno (tastiera, monitor, mouse, dischi, stampanti, interfacce di rete, modem, webcam, ecc.)
- **Bus:** collega tutte le unità funzionali



QuickTime™ e un decompressore sono necessari per visualizzare quest'immagine.

---

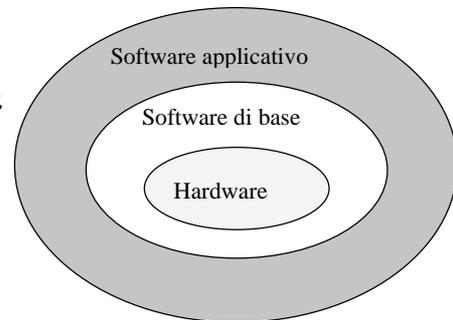
*Fondamenti di Informatica T*

## SOFTWARE

Programmi che vengono eseguiti dal sistema.

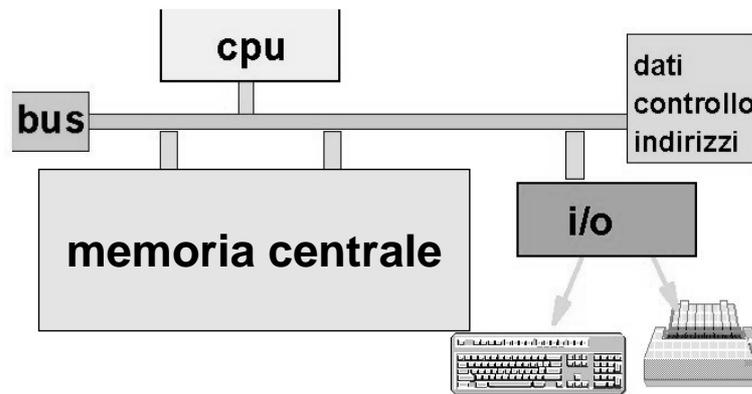
### Distinzione fra:

- Software di base (Sistema Operativo) programmi che presiedono alla gestione delle risorse hardware presentando all'utente una visione astratta e semplificata dell'elaboratore.
- Software applicativo:
  - Programmi di utilità che consentono lo svolgimento di particolari attività (es. word processing, web browser, fogli elettronici,...)



## Hardware: il modello di Von Neumann

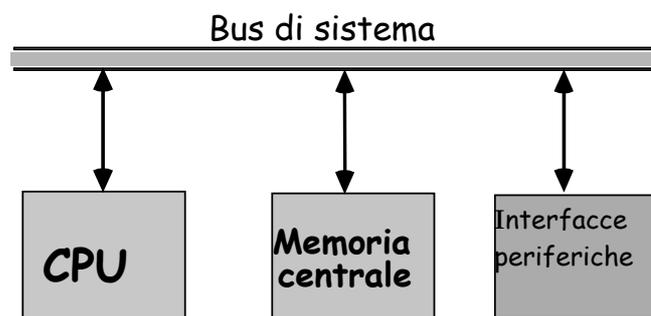
## ARCHITETTURA DI UN ELABORATORE



Ispirata al modello della **Macchina di Von Neumann** (Princeton, Institute for Advanced Study, anni '40).

*Fondamenti di Informatica T*

## MACCHINA DI VON NEUMANN

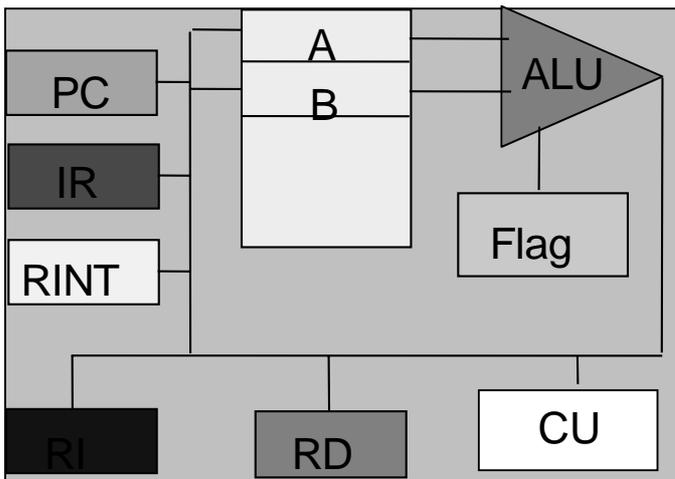


### UNITÀ FUNZIONALI fondamentali

- Processore (CPU)
- Memoria Centrale (RAM & ROM)
- Periferiche (ingresso / uscita)
- Bus di sistema

*Fondamenti di Informatica T*

## CPU



è il dispositivo che esegue l'elaborazione sulle informazioni e coordina/controlla tutte le unità funzionali

- ALU (Arithmetic & Logic Unit)
- CU (Unità di Controllo)
- Registri

Fondamenti di Informatica T

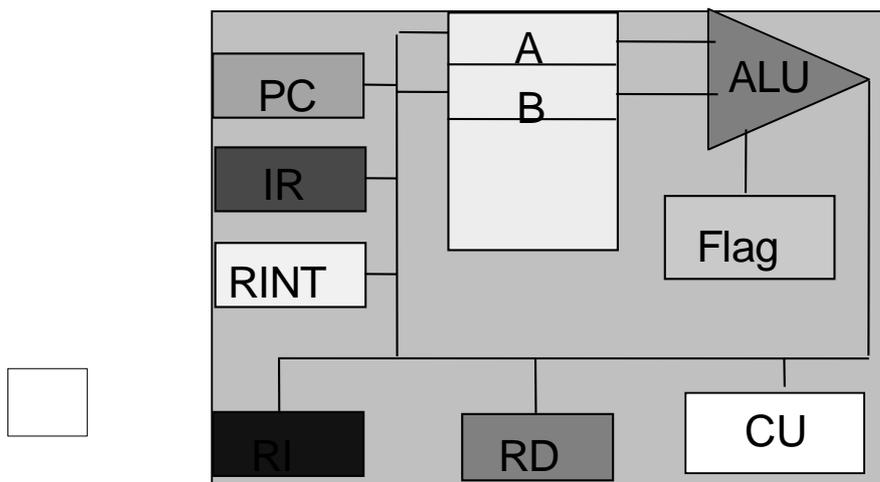
## MEMORIA CENTRALE



La memoria centrale è una collezione di celle *numerate*, che possono contenere **DATI** e **ISTRUZIONI**. Le istruzioni sono disposte in memoria in *celle di indirizzo crescente*.

Fondamenti di Informatica T

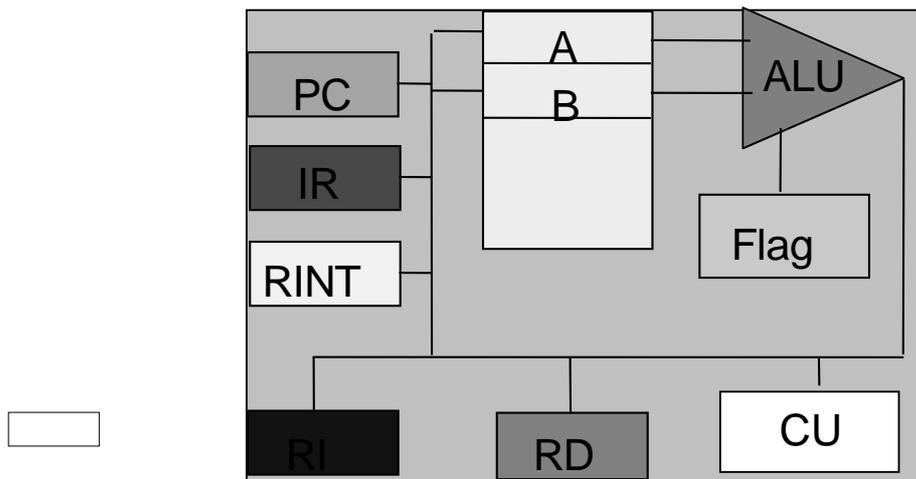
## UNITÀ DI ELABORAZIONE (CPU)



### ALU (Arithmetic / Logic Unit)

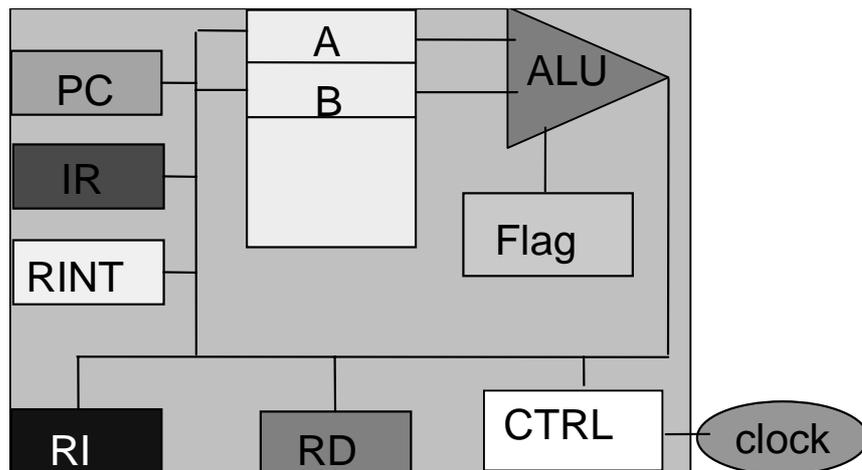
Esegue le operazioni aritmetiche e logiche elementari

## UNITÀ DI ELABORAZIONE (CPU)



**Unità di Controllo (Control Unit):** controlla e coordina l'attività della CPU. (In particolare, controlla il trasferimento dei dati tra memoria e registri e la decodifica e l'esecuzione delle istruzioni)

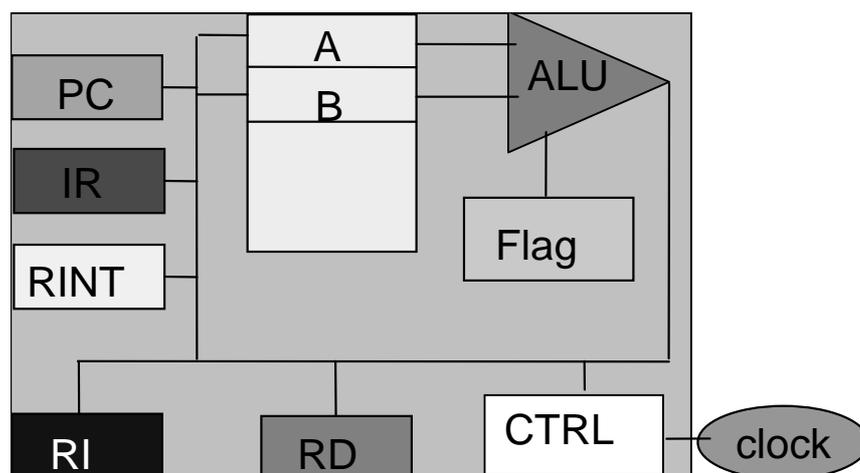
## UNITÀ DI ELABORAZIONE (CPU)



**Il clock dà la base dei tempi necessaria per mantenere il sincronismo fra le operazioni**

*Fondamenti di Informatica T*

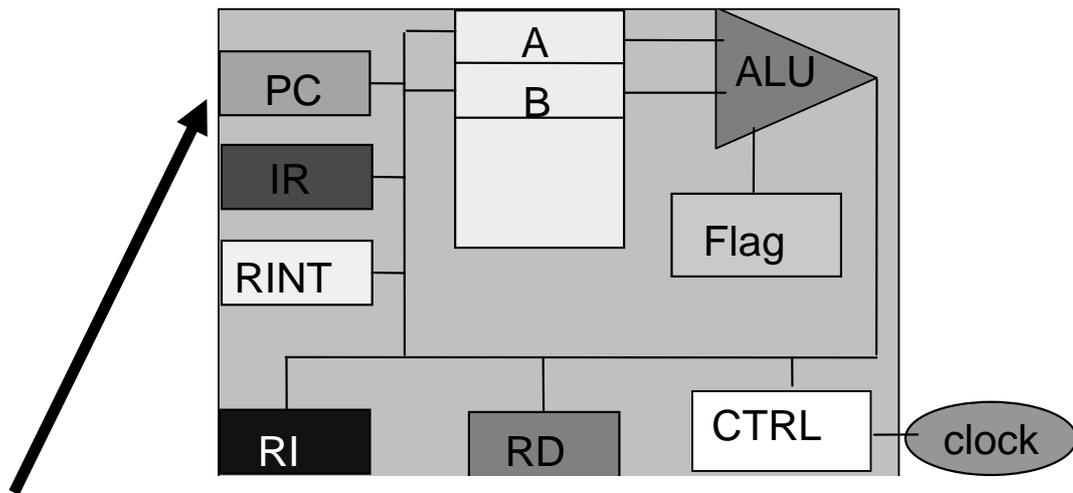
## UNITÀ DI ELABORAZIONE (CPU)



**I registri (qui A, B, PC, Flag,...) sono *locazioni* usate per memorizzare dati, istruzioni, o indirizzi *all'interno della CPU*. L'accesso ai registri è *molto veloce*.**

*Fondamenti di Informatica T*

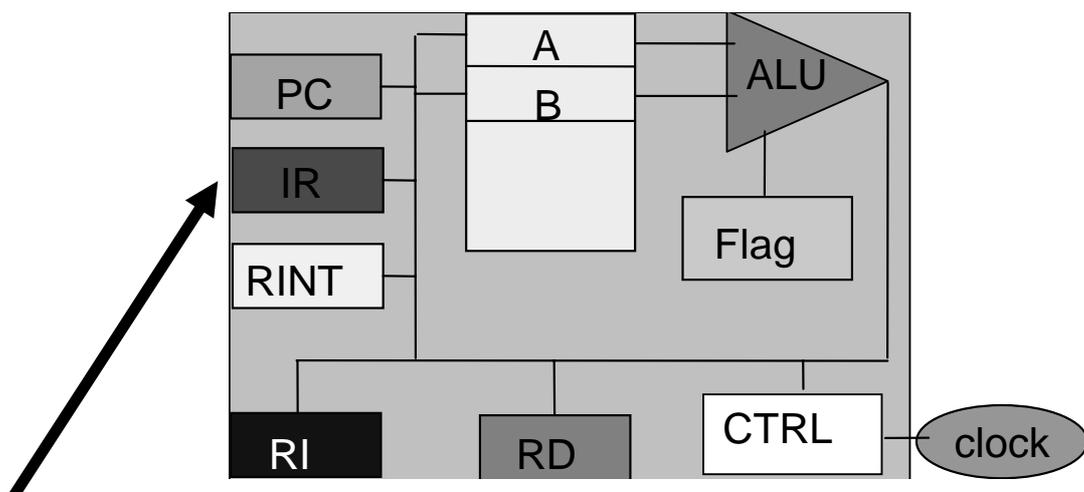
## I REGISTRI



### Program Counter (PC)

Indica l'indirizzo della cella di memoria che contiene la prossima istruzione da eseguire

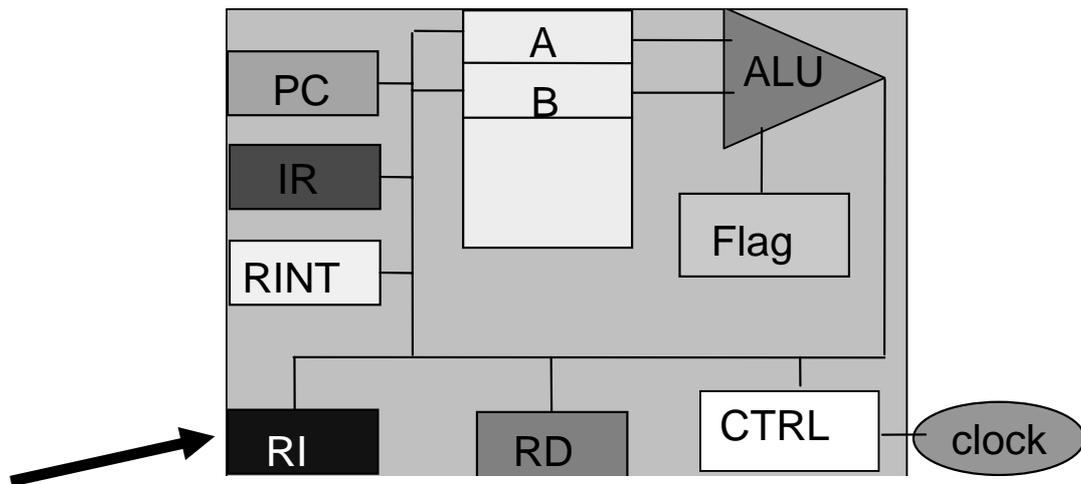
## I REGISTRI



### Instruction Register (IR)

Contiene l'istruzione da eseguire.

## I REGISTRI

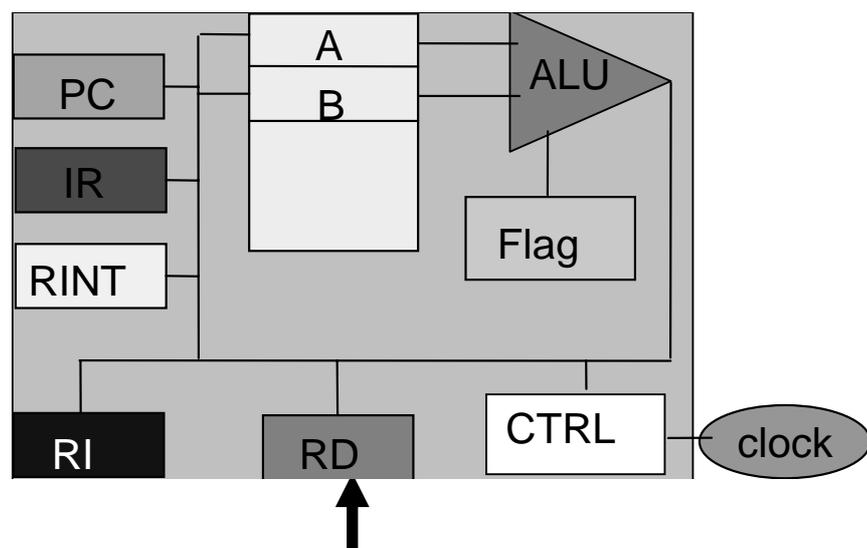


### **Registro Indirizzi (RI)**

Contiene l'indirizzo della cella di memoria da selezionare per il trasferimento di un dato

*Fondamenti di Informatica T*

## I REGISTRI

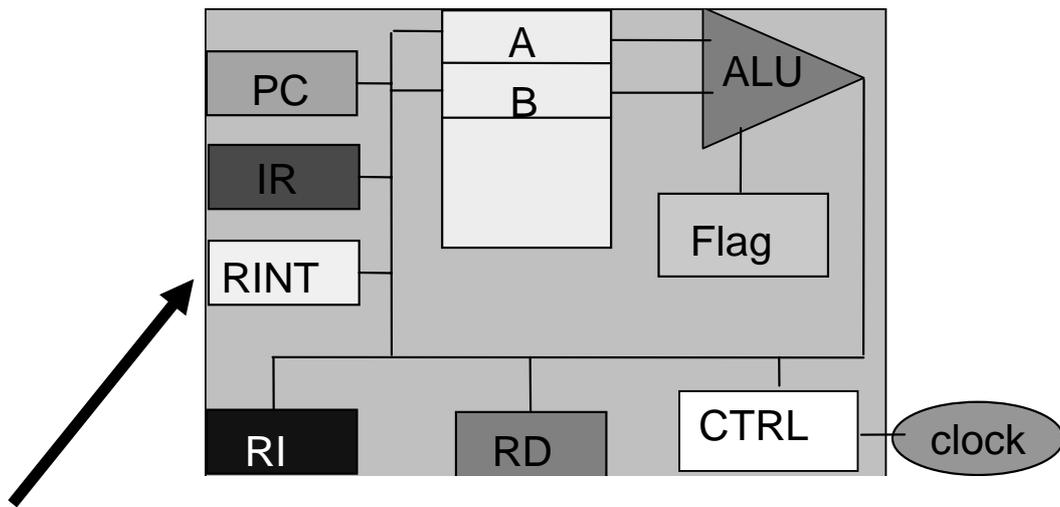


### **Registro Dati (RD)**

Contiene il dato attualmente oggetto di elaborazione

*Fondamenti di Informatica T*

## I REGISTRI

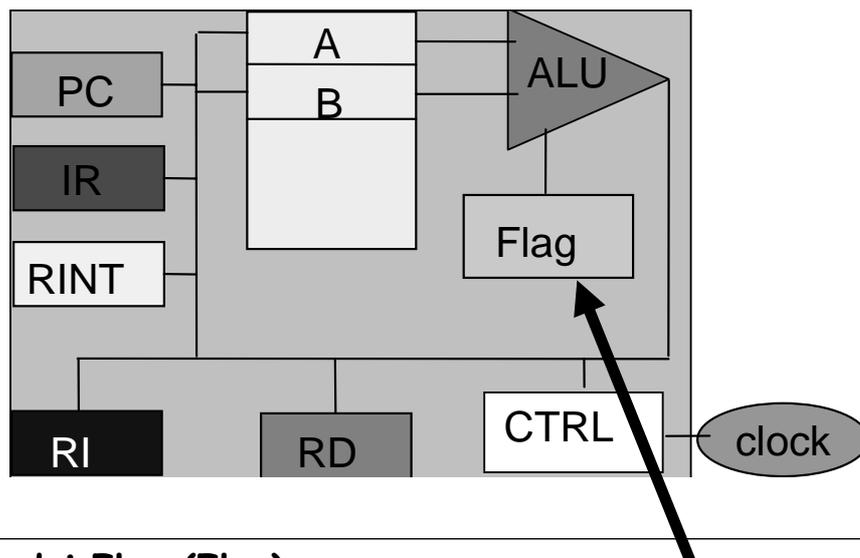


### Registro Interruzioni (RINT)

Serve per l'interazione con le periferiche e per la gestione di condizioni eccezionali.

*Fondamenti di Informatica T*

## I REGISTRI

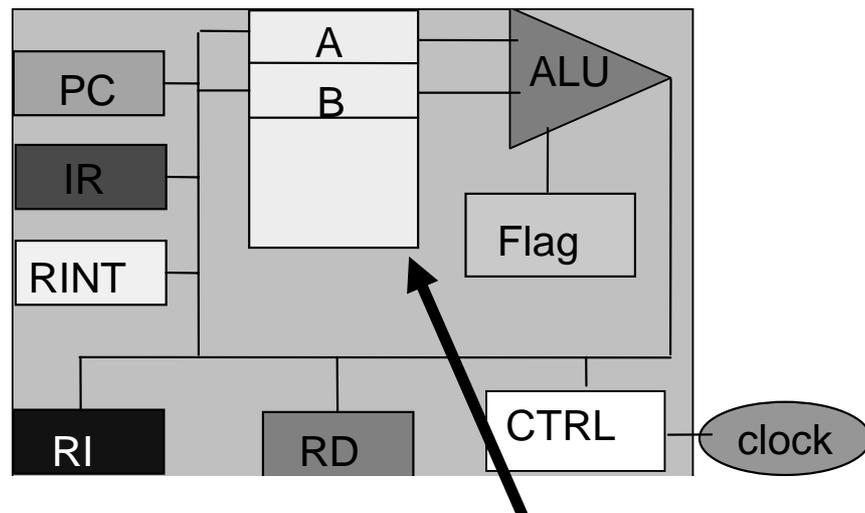


### Registro dei Flag (Flag)

Ogni flag indica la presenza/assenza di una proprietà (es. errori, riporti, ecc) nel risultato dell'ultima operazione eseguita dalla ALU. Altri bit riassumono lo stato del processore.

*Fondamenti di Informatica T*

## I REGISTRI

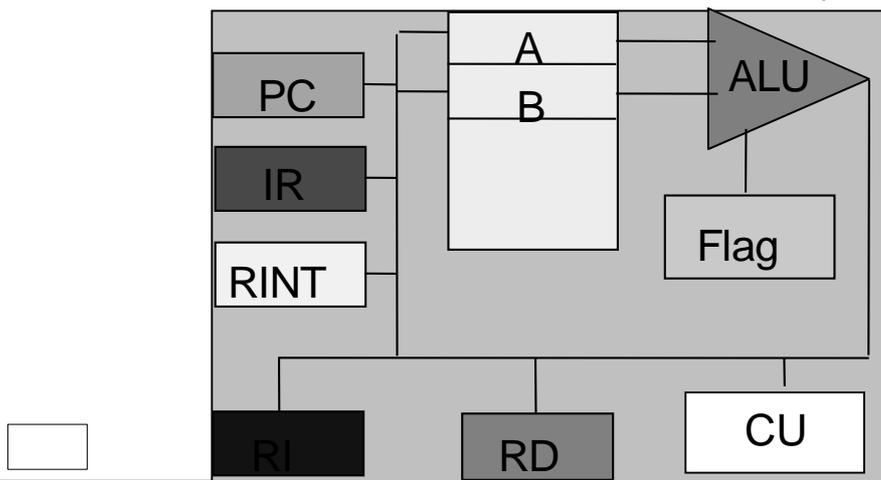


### Registri di uso generale (A, B, C, ...)

Sono usati per contenere dati (in particolare, operandi/risultati di operazioni aritmetico/logiche)

Fondamenti di Informatica T

## UNITÀ DI ELABORAZIONE (CPU)



L'unità di controllo fa funzionare l'elaboratore.

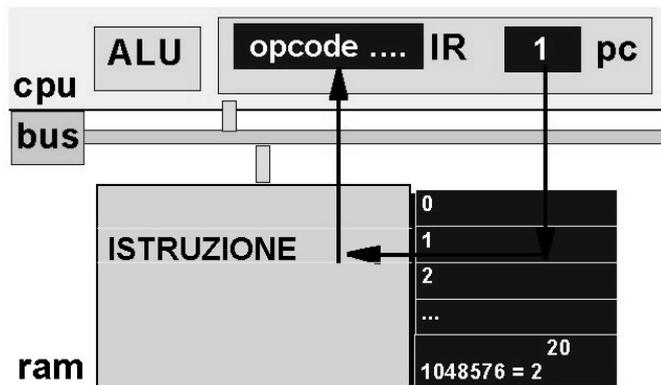
Da quando viene acceso a quando è spento, essa esegue in continuazione il ciclo di *prelievo / decodifica / esecuzione* (fetch / decode / execute).

Fondamenti di Informatica T

# IL CICLO fetch / decode / execute

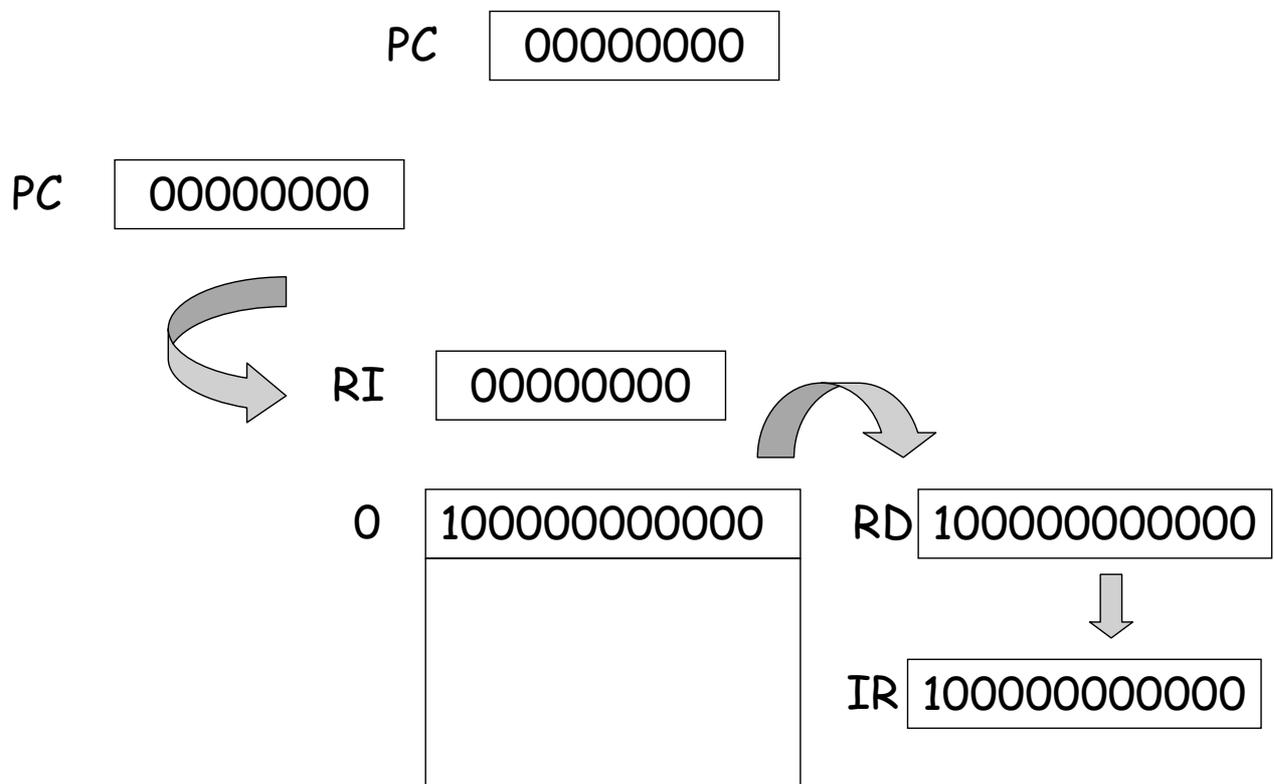
## FETCH

- si accede alla **prossima istruzione** (cella il cui indirizzo è contenuto nel registro **PC**) ...
- ... e la si porta dalla **memoria centrale**, memorizzandola nel **Registro Istruzioni (IR)**



Si copia il contenuto di PC in RI. Si accede alla locazione di memoria indirizzata e si estrae la parola memorizzandola nel RD. Il contenuto del registro dati poi viene trasferito sul registro istruzione corrente

Fondamenti di Informatica T

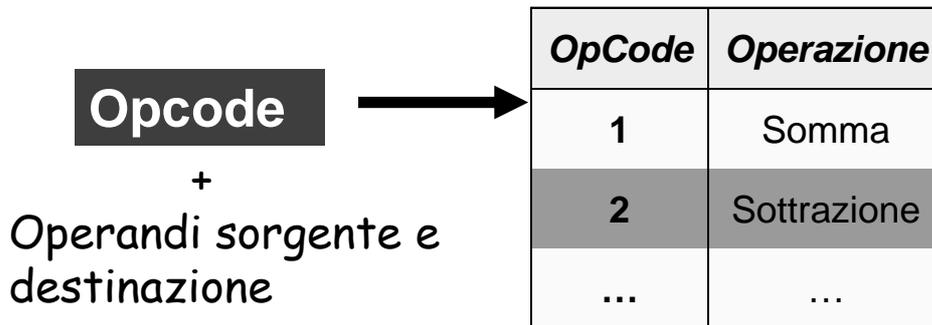


Fondamenti di Informatica T

## IL CICLO fetch / decode / execute

### DECODE

- si decodifica il tipo dell'istruzione in base al suo *OpCode* (codice operativo)



Istruzione a tre operandi (es. somma, due addendi e risultato)

Istruzione a due operandi (es. istruzione di trasferimento)

Istruzioni a un operando (es. istruzione di salto)

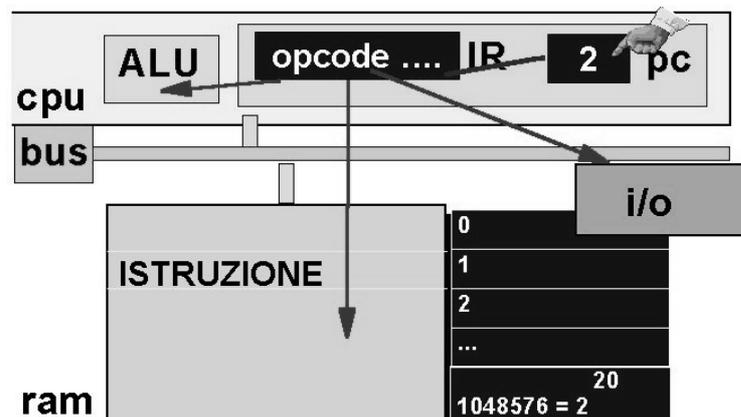
Istruzione senza operandi (es. Istruzione di halt)

Fondamenti di Informatica T

## IL CICLO fetch / decode / execute

### EXECUTE

- si individuano i dati usati dall'istruzione
- si trasferiscono tali dati nei registri opportuni
- si esegue l'istruzione.



Fondamenti di Informatica T

## IL CICLO fetch / decode / execute

### Attenzione:

Istruzioni particolari possono *alterare il prelievo delle istruzioni da celle consecutive*:

- istruzioni di salto
- istruzioni di chiamata a sotto-programmi
- istruzioni di interruzione/terminazione

---

*Fondamenti di Informatica T*

## MULTITASKING

Poiché i registri compendiano *tutto lo stato dell'elaborazione* di un certo processo,

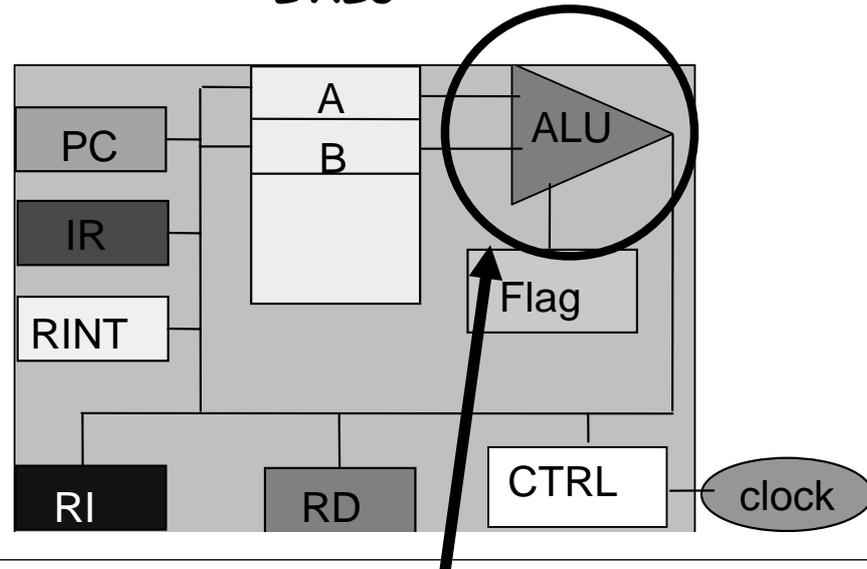
- salvando in memoria il contenuto di tutti i registri è possibile *interrompere l'esecuzione di un programma* per passare a svolgerne un altro
- ripristinando dalla memoria il contenuto di tutti i registri precedentemente salvati è possibile *ripristinare lo stato dell'esecuzione di un programma interrotto*, riprendendone l'esecuzione *come se nulla fosse accaduto*.

Questa possibilità è ciò che consente a un sistema operativo di eseguire più compiti "allo stesso tempo": **MULTITASKING**

---

*Fondamenti di Informatica T*

## L'ALU



Esegue operazioni aritmetiche, logiche e confronti sui dati della memoria centrale o dei registri.

Fondamenti di Informatica T

## L'ALU (segue)

### ESEMPIO SEMPLICE:

ALU in grado di eseguire **somma**, **sottrazione**, **prodotto**, **divisione** con due operandi contenuti nei registri A e B.

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



Possibile alterazione di due bit nel registro **Flag**:  
**carry** (riporto) e **overflow**

Fondamenti di Informatica T

## LA MEMORIA CENTRALE

### INDIRIZZAMENTO

- E' l'attività con cui l'elaboratore seleziona una particolare cella di memoria
- Per farlo, l'elaboratore pone l'indirizzo della cella desiderata nel Registro Indirizzi (RI).
  - se il RI è lungo  $N$  bit, si possono indirizzare  $2^N$  celle di memoria (numerate da 0 a  $2^N-1$ )
  - esempio:  $N=10 \rightarrow 2^N = 1024$  celle.
- Oggi, RI è lungo tipicamente 32 bit  
→ **SPAZIO INDIRIZZABILE di 4 Gb**

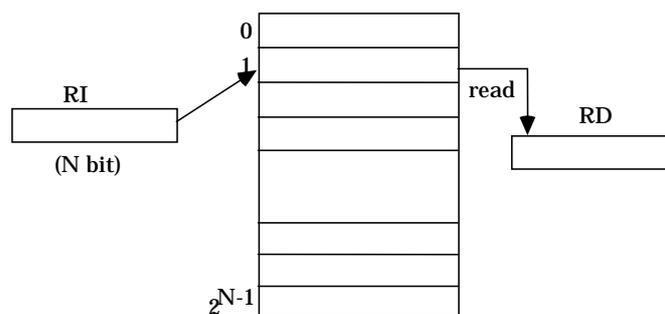
---

Fondamenti di Informatica T

## LA MEMORIA CENTRALE (2)

### OPERAZIONI

- **Lettura (Read)**: il contenuto della cella di memoria indirizzata dal Registro Indirizzi è copiato nel Registro Dati.



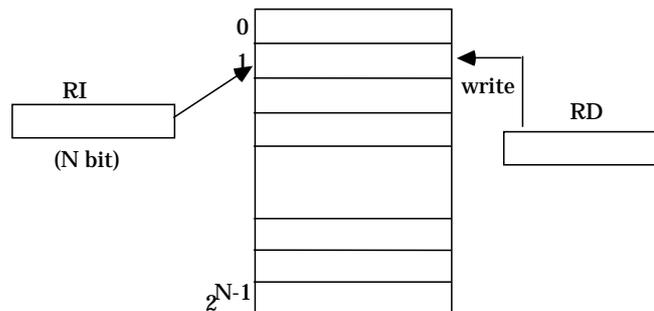
---

Fondamenti di Informatica T

## LA MEMORIA CENTRALE (3)

### OPERAZIONI

- **Scrittura (*Write*)**: il contenuto del Registro Dati è copiato nella cella di memoria indirizzata dal Registro Indirizzi.



---

Fondamenti di Informatica T

### Prestazioni CPU e Memoria

#### **PROBLEMA:**

Sebbene la RAM sia veloce, non è **abbastanza** veloce da "star dietro" ai moderni processori.

#### **CONSEGUENZA:**

il processore *perde tempo* ad aspettare l'arrivo dei dati dalla RAM.

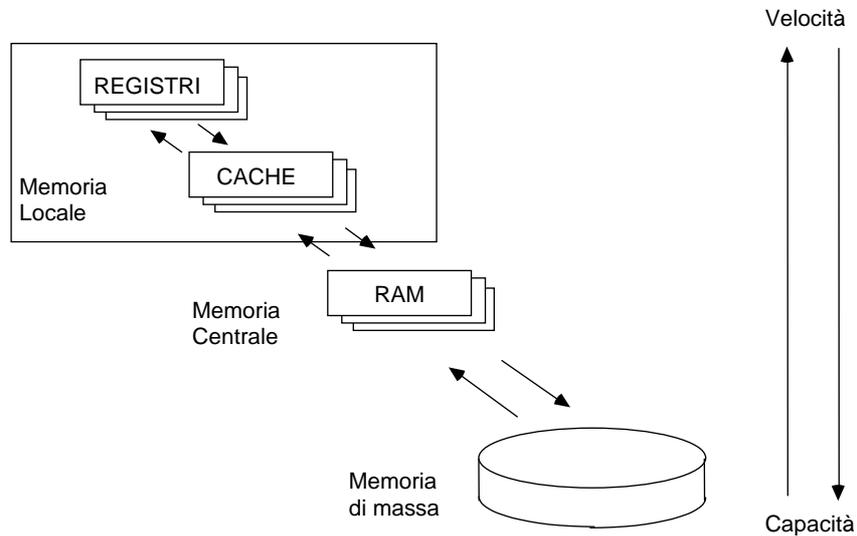
---

Fondamenti di Informatica T

Soluzione: usare gerarchie di memorie con costi e prestazioni diversi

- memoria di massa
- memoria centrale
- memoria cache

### GERARCHIA DELLE MEMORIE



Fondamenti di Informatica T

### MEMORIE CACHE

#### **SOLUZIONE:**

Inserire tra processore e RAM una *memoria particolarmente veloce* dove tenere i dati usati più spesso (*memoria cache*)

In questo modo,

- ◆ la prima volta che il microprocessore carica dei dati dalla memoria centrale, tali dati vengono caricati anche sulla cache
- ◆ le volte successive, i dati possono essere letti dalla cache (veloce) invece che dalla memoria centrale (più lenta)

Fondamenti di Informatica T

## MEMORIE CACHE

### DUBBIO:

Ma se abbiamo memorie così veloci, *perché non le usiamo per costruire tutta la RAM?*

**Semplice...**  
**perché costano molto!!**

**OGGI**, la cache è spesso già presente dentro al processore (cache di I° livello), e altra può essere aggiunta (cache di II° livello)

Fondamenti di Informatica T

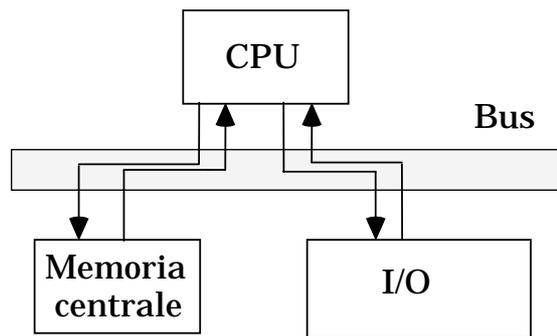
## BUS DI SISTEMA



Il Bus di Sistema interconnette la CPU, la memorie e le interfacce verso dispositivi periferici (I/O, memoria di massa, etc.)

Fondamenti di Informatica T

## BUS DI SISTEMA (2)



Il Bus collega due unità funzionali alla volta:

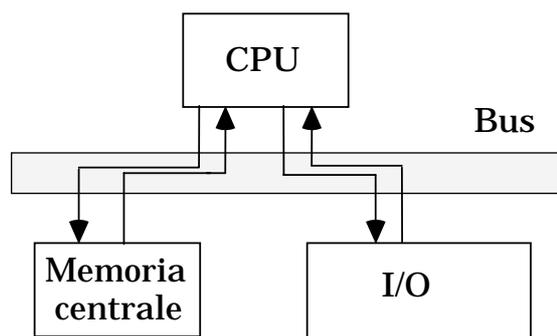
una trasmette...

... e l'altra riceve.

Il trasferimento dei dati avviene o sotto il controllo della CPU, o mediante accesso diretto alla memoria (DMA).

*Fondamenti di Informatica T*

## BUS DI SISTEMA (3)



Il Bus è in realtà un insieme di linee diverse:

bus **dati** (data bus)

bus **indirizzi** (address bus)

bus **comandi** (command bus)

*Fondamenti di Informatica T*

## BUS DI SISTEMA (4)

### **BUS DATI**

bidirezionale

serve per trasmettere dati dalla memoria o viceversa.

### **BUS INDIRIZZI**

unidirezionale

serve per trasmettere il contenuto del registro indirizzi alla memoria  
(si seleziona una specifica cella su cui viene eseguita o un'operazione di lettura o una operazione di scrittura)

---

*Fondamenti di Informatica T*

## BUS DI SISTEMA (5)

### **BUS COMANDI**

- **Unidirezionale [bidirezionale]**
- tipicamente usato per *inviare comandi verso la memoria* (es: lettura o scrittura) o *verso una periferica* (es. stampa verso la stampante)
- [può essere usato per *inviare comandi verso il processore* nel caso di DMA (o interfacce di I/O)]

---

*Fondamenti di Informatica T*

## Esempi:

Per effettuare l'operazione di lettura la CPU:

1. Carica l'indirizzo della locazione da cui vuole leggere sul bus degli indirizzi;
2. Chiede di effettuare una lettura in memoria inviando un segnale sulla linea dei comandi MEM READ

Il banco di memoria interessato all'operazione:

1. Legge il dato contenuto nella locazione selezionata e lo carica sul bus dei dati
2. Indica alla CPU la presenza del dato sul bus dei dati inviando un segnale sulla linea comandi READY

## DISPOSITIVI DI MEMORIA: tecnologie

### DISPOSITIVI FISICI

- **RAM:** Random Access Memory (ad accesso casuale): su di essa si possono svolgere operazioni sia di lettura che di scrittura
- **ROM:** Read Only Memory (a sola lettura): non volatili e non scrivibili dall'utente (che la ordina con un certo contenuto); in esse sono contenuti i dati e programmi per inizializzare il sistema
- **PROM:** Programmable ROM. Si possono scrivere soltanto una volta, mediante particolari apparecchi (detti programmatori di PROM).

## DISPOSITIVI DI MEMORIA (segue)

### DISPOSITIVI FISICI (segue)

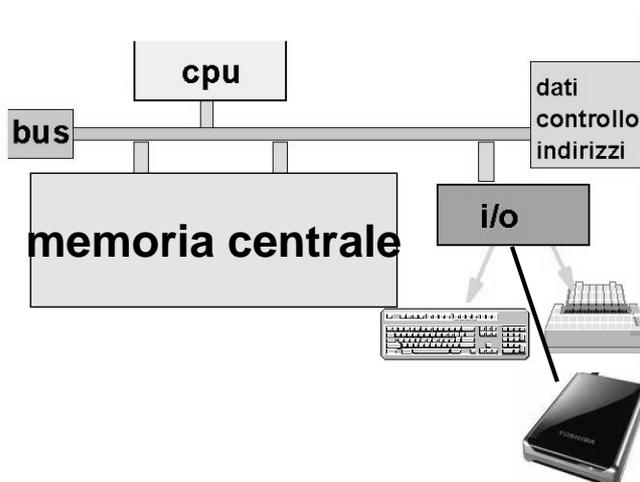
- **EPROM**: Erasable-Programmable ROM (si cancellano sottoponendole a raggi ultravioletti).
- **EEPROM**: Electrically-Erasable-PROM (si cancellano elettricamente).

**Il *Firmware*** è costituito da software memorizzato su ROM, EPROM, etc. (codice microprogrammato).

---

Fondamenti di Informatica T

### INTERFACCE DI I/O



Una interfaccia (controller) è un dispositivo che consente all'elaboratore di comunicare con una periferica (tastiere, mouse, dischi, terminali, stampanti, ...).

---

Fondamenti di Informatica T

# LA MEMORIA SECONDARIA

## MEMORIA SECONDARIA ("DI MASSA")

È un particolare dispositivo di I/O che:

- memorizza **grandi quantità** di informazioni
- **persistente** (le informazioni non si perdono spegnendo la macchina)
- **accesso molto meno rapido** della memoria centrale (millisecondi contro nanosecondi / differenza  $10^6$ )

### Esempi:

- Dischi (tecnologia magnetica)
- CD/DVD (tecnologia ottica)
- Memoria flash: PENNE USB, schede.. (tecnologia elettronica ssd)

---

*Fondamenti di Informatica T*

## DISPOSITIVI MAGNETICI

- L'area del dispositivo è suddivisa in **micro-zone**
- Ogni micro-zona memorizza una **informazione elementare** sotto forma di **stato di magnetizzazione**:
  - **area magnetizzata / area non magnetizzata**
- Ai due possibili stati di magnetizzazione vengono associate le due cifre binarie **0 e 1**  
**bit** (Binary digIT)
- Quindi, **ogni micro-zona memorizza 1 bit**
- Per memorizzare informazioni più complesse si considerano *collezioni di bit*.

**BYTE** (collezione di **8 bit**) e suoi multipli

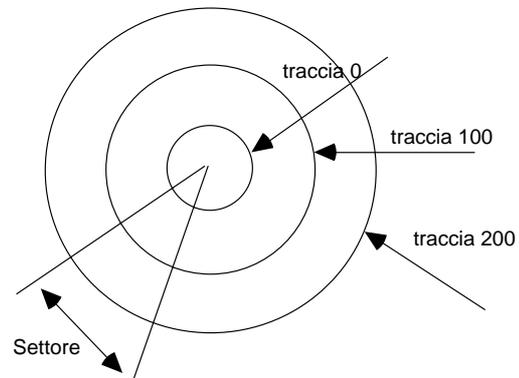
---

*Fondamenti di Informatica T*

## DISCHI MAGNETICI

Un disco consiste in un certo numero di **piatti** con **due superfici** che ruotano attorno ad un perno centrale.

Ogni superficie dispone di una propria **testina di lettura / scrittura**.

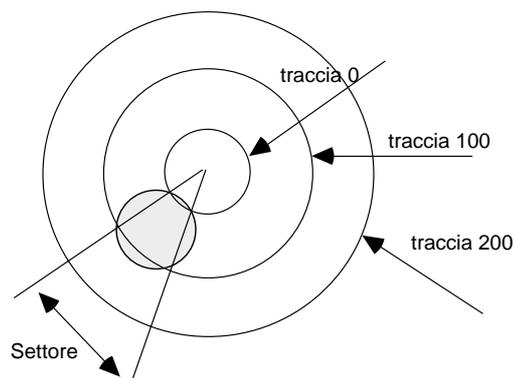


Le superfici sono organizzate in **cerchi concentrici (tracce)** e in **spicchi di ugual grandezza (settori)**.  
Le tracce equidistanti dal centro formano un **cilindro**.

*Fondamenti di Informatica T*

## DISCHI MAGNETICI (segue)

Ogni **blocco** del disco è identificato con la terna **<superficie, traccia, settore>**.  
Per effettuare il trasferimento dei dati in memoria centrale occorre disporre di un'area di memoria (**buffer**) di dimensioni pari al blocco.



L'operazione di partizionamento del disco in tracce e settori prende il nome di **FORMATTAZIONE**

*Fondamenti di Informatica T*

## DISPOSITIVI OTTICI

### **1984, CD-ROM (Compact-Disk Read-Only Memory)**

- Capacità: > 600 MB
- Velocità di trasferimento:
  - originariamente 150 KB / s ( "1X" )
  - oggi 24, 32, 40 volte tanto...

### **1984, WORM (Write Once Read Many)**

- Sono dischi ottici scrivibili (una sola volta)
- Parenti stretti dei CD audio (CD-DA, 1982)
- Accesso diretto ai settori (capacità 2.048 KB)

---

*Fondamenti di Informatica T*

## DISPOSITIVI OTTICI (segue)

### **1986, CD - I (Compact-Disk Interactive)**

- Per memorizzare immagini, filmati, grafica, suono, testi e dati (*multimedialità*).

Ormai il CD è il principale mezzo per lo scambio di grandi quantità di informazioni

- installazione di nuovi programmi di utilità
- archiviazione di immagini, suoni, opere multimediali
- copie di riserva (backup)
- distribuzione di materiale pubblicitario o "di prova"

**Affidabilità: fino a 10-15 anni.**

---

*Fondamenti di Informatica T*

## Il presente

### 1997, DVD (Digital Video Disk)

- Evoluzione del CD-ROM
- Capacità fino a 4,7 GB
- Velocità di trasferimento molto elevata

Adatto per film e opere pesantemente multimediali.

### Flash memory stick :

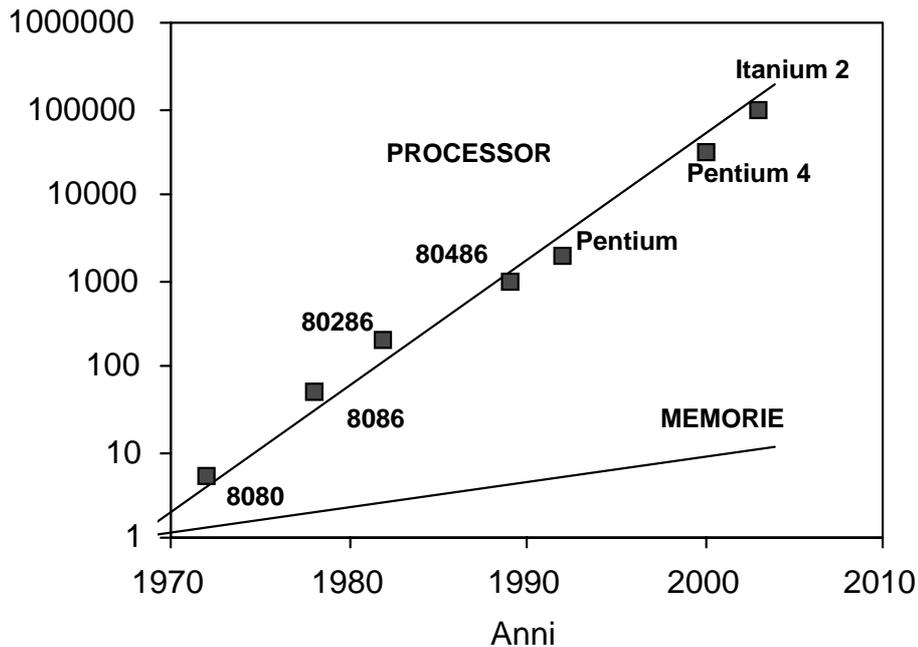
- memorie persistenti che possono essere riscritte/cancellate più volte
- capacità : fino a 16 GB



## CAPACITÀ DELLE MEMORIE

Tipo di memoria	Capacità
Memoria centrale	16-512 Mbyte
Dischi magnetici	1 GByte -100 Gbyte
Dischi floppy	1.4 Mbyte
Nastri (bobina)	20-400 Mbyte
Nastri (cassetta)	200-5000 Mbyte
Dischi ottici	650 Mbyte - 17 GByte

# Legge di Moore



la legge di Moore mostra il crescente divario tra la velocità dei processori e quella delle memorie

Fondamenti di Informatica T

## OLTRE la macchina di Von Neumann

- **Problema:** nella Macchina di Von Neumann le operazioni sono *strettamente sequenziali*.
- Altri modelli architetturali prevedono forme di *parallelismo*:
  - **processori dedicati (coprocessori)** al calcolo numerico, alla gestione della grafica, all'I/O.
  - **esecuzione in parallelo** delle varie fasi di un'istruzione: mentre se ne esegue una, si acquisiscono e decodificano le istruzioni successive (*pipeline*)
  - **architetture completamente diverse:** sistemi multi-processore, macchine dataflow, LISP machine, reti neurali.

Fondamenti di Informatica T

# Il Software

---

*Fondamenti di Informatica T*

## TECNOLOGIA DIGITALE

CPU, memoria centrale e dispositivi sono realizzati con **tecnologia elettronica digitale**.

Dati ed operazioni vengono codificati a partire da due valori distinti di grandezze elettriche:

- tensione alta ( $V_H$ , 5V)
- tensione bassa ( $V_L$ , 0V)

A tali valori vengono convenzionalmente **associate le due cifre binarie 0 e 1**:

- **logica positiva:**  $1 \square V_H$ ,  $0 \square V_L$
- **logica negativa:**  $0 \square V_H$ ,  $1 \square V_L$

---

*Fondamenti di Informatica T*

## TECNOLOGIA DIGITALE (segue)

Dati ed operazioni vengono codificati tramite **sequenze di bit**

**01000110101 ....**

CPU è in grado di operare soltanto in aritmetica binaria, effettuando operazioni *elementari*:

- somma e differenza
- scorrimento (shift)
- ...

Lavorando direttamente sull'hardware, l'utente è **forzato a esprimere i propri comandi al livello della macchina, tramite sequenze di bit.**

*Fondamenti di Informatica T*

## IL SOFTWARE

### Software:

insieme di programmi eseguibili dal computer.

**Organizzazione a strati**, ciascuno con funzionalità di livello più alto rispetto a quelli sottostanti

Concetto di **macchina virtuale**



*Fondamenti di Informatica T*

## IL FIRMWARE

### **Firmware:**

il confine fra hardware e software.

**È uno strato di *micro-programmi*, scritti dai costruttori dell'hardware, che agiscono direttamente al di sopra dello strato hardware**

Sono memorizzati su una speciale *memoria centrale permanente* (ROM, EPROM, ...)

---

*Fondamenti di Informatica T*

## IL SISTEMA OPERATIVO

Programma che opera *al di sopra* dell'hardware fornendo un ambiente di esecuzione per i programmi, nascondendo i dettagli relativi alle caratteristiche fisiche delle componenti e alla loro gestione.

Per lo stesso elaboratore, spesso **si può scegliere tra *diversi sistemi operativi***, con diverse caratteristiche.

### **Esempi:**

- Windows (95 / 98, NT, XP, Vista...)
- Unix
- Linux
- MacOS X...

---

*Fondamenti di Informatica T*

## FUNZIONI DEL SISTEMA OPERATIVO

Le funzioni messe a disposizione dal S.O. dipendono dalla complessità del sistema di elaborazione:

- Interazione con l'utente ->interpretazione ed esecuzione di comandi
- gestione delle risorse disponibili:
  - Cpu: assegnazione delle cpu ai diversi programmi,
  - Memoria centrale : allocazione della memoria ai programmi
  - Dispositivi: accesso alle periferiche indipendente dalle caratteristiche hw dei dispositivi; organizzazione e gestione della memoria di massa.
- gestione di un sistema multi-utente/multitasking:
  - concorrenza delle attività`
  - protezione
  - una macchina astratta (o virtuale) per ogni utente:

**Ogni utente "vede" l'elaboratore solo tramite il Sistema Operativo  
→ il S.O. realizza una "macchina virtuale"**

---

*Fondamenti di Informatica T*

## FUNZIONI DEL SISTEMA OPERATIVO

### **Conseguenza:**

S.O. diversi possono realizzare *diverse macchine virtuali* sullo stesso elaboratore fisico

### **Interazione con l'utente:**

Attraverso il S.O. il livello di interazione fra utente ed elaboratore viene elevato:

- senza S.O.:           sequenze di bit
- con S.O.:             comandi, programmi, dati, forniti in modo testuale o grafico

---

*Fondamenti di Informatica T*

## RUOLO DEL SISTEMA OPERATIVO

Il S.O. traduce le richieste dell'utente in opportune sequenze di istruzioni, a loro volta trasformate in corrispondenti sequenze di bit per la macchina fisica.



e viceversa:



---

*Fondamenti di Informatica T*

## RUOLO DEL SISTEMA OPERATIVO

**Qualsiasi operazione di accesso a risorse della macchina implicitamente richiesta dal comando di utente viene esplicitata dal S.O.**

### **Esempi:**

- accesso a memoria centrale
- accesso ai dischi
- I/O verso video, tastiera, ...

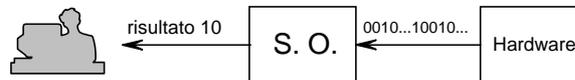
---

*Fondamenti di Informatica T*

## ESEMPIO



e viceversa:



<u>Utente:</u> "esegui progr1"	<u>Sistema Operativo:</u> - input da tastiera - ricerca codice di "progr1" su disco - carica in memoria centrale codice e dati <elaborazione>
<u>Utente:</u> "stampa 10"	<u>Sistema Operativo:</u> - output su video

Fondamenti di Informatica T

## CLASSIFICAZIONE dei S.O.

In base al numero di utenti:

- **Mono-utente (*mono-user*):** un solo utente alla volta può utilizzare il sistema
- **Multi-utente (*multi-user*):** più utenti possono interagire contemporaneamente con la macchina.

Nel caso di più utenti contemporanei, il **Sistema Operativo** deve fornire a ciascuno l'astrazione di **un sistema "dedicato"**.

Fondamenti di Informatica T

## CLASSIFICAZIONE dei S.O.

In base al numero di programmi in esecuzione:

- **Mono-programmato (*mono-tasking*):** si può eseguire *un solo programma* per volta
- **Multi-programmato (*multi-tasking*):** il S.O. è in grado di portare avanti contemporaneamente l'esecuzione di più programmi (pur usando una sola CPU).

Nel caso di multi-programmazione il S.O. deve gestire la suddivisione del tempo della CPU fra i vari programmi.

## CLASSIFICAZIONE dei S.O.

**Esempi:**

- **MS-DOS:** monoutente, monoprogrammato
- **Windows95/98:** monoutente, multiprogrammato
- **Windows XP, Vista:** mono/multiutente, multiprogrammato
- **UNIX e Linux:** multiutente, multiprogrammato

## PROGRAMMI APPLICATIVI

Risolvono problemi specifici degli utenti:

- *word processor*: elaborazione di testi
- *fogli elettronici*: gestione di tabelle, calcoli e grafici
- *database*: gestione di archivi
- *suite* (integrati): collezione di applicativi capaci di funzionare in modo integrato come un'applicazione unica.

- Sono scritti in **linguaggi di programmazione** di alto livello
- Risentono in misura ridotta delle caratteristiche della architettura dell'ambiente sottostante (*portabilità*)

---

Fondamenti di Informatica T

## AMBIENTI DI PROGRAMMAZIONE

È l'insieme dei programmi che consentono la scrittura, la verifica e l'esecuzione di nuovi programmi (*fasi di sviluppo*).

**Sviluppo di un programma:**

- Affinché un programma scritto in un qualsiasi linguaggio di programmazione sia comprensibile (e quindi eseguibile) da un calcolatore, occorre **tradurlo** dal linguaggio originario al linguaggio della macchina.
- Questa operazione viene normalmente svolta da speciali programmi, detti **traduttori**.

---

Fondamenti di Informatica T

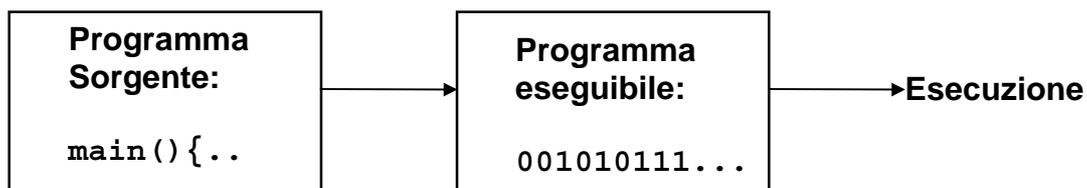
## TRADUZIONE DI UN PROGRAMMA

PROGRAMMA	TRADUZIONE
main() { int A; ... A=A+1; if....	00100101  11001.. 1011100..

Il traduttore converte

- **il testo** di un algoritmo scritto in un particolare linguaggio di programmazione (**sorgenti**)
- nella corrispondente **rappresentazione in linguaggio macchina** (programma **eseguibile**).

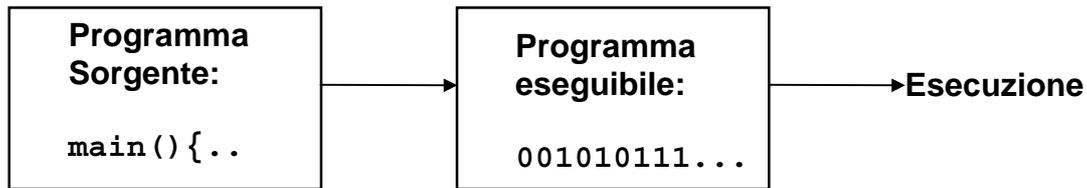
## SVILUPPO DI PROGRAMMI



Due categorie di traduttori:

- i **Compilatori** traducono l'intero programma (senza eseguirlo!) e producono in uscita il programma convertito in linguaggio macchina
- gli **Interpreti** traducono ed eseguono immediatamente ogni singola istruzione del *programma sorgente*.

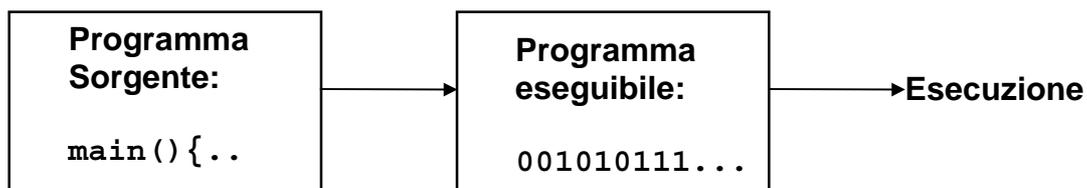
## SVILUPPO DI PROGRAMMI (segue)



Quindi:

- **nel caso del compilatore**, lo schema precedente viene percorso *una volta sola* prima dell'esecuzione
- **nel caso dell'interprete**, lo schema viene invece attraversato *tante volte quante sono le istruzioni* che compongono il programma.

## SVILUPPO DI PROGRAMMI (segue)



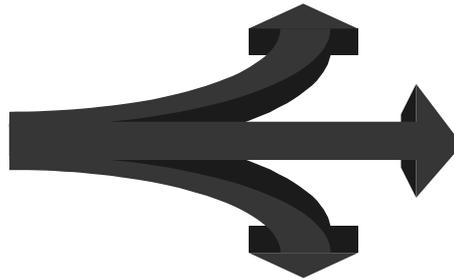
L'esecuzione di un programma *compilato* è più veloce dell'esecuzione di un programma *interpretato*

## AMBIENTI DI PROGRAMMAZIONE

### COMPONENTI

- **Editor**: serve per creare file che contengono **testi** (cioè sequenze di caratteri).  
In particolare, l'editor **consente di scrivere il programma sorgente**.

E poi...



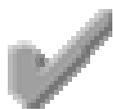
---

Fondamenti di Informatica T

## AMBIENTI DI PROGRAMMAZIONE

### I° CASO: COMPILAZIONE

- **Compilatore**: opera la **traduzione di un programma sorgente** (scritto in un linguaggio ad alto livello) in un **programma oggetto** direttamente eseguibile dal calcolatore.



**PRIMA** si traduce *tutto il programma*  
**POI** si esegue la versione tradotta.

---

Fondamenti di Informatica T

## AMBIENTI DI PROGRAMMAZIONE (2)

### I° CASO: COMPILAZIONE (segue)

- **Linker:** (*collegatore*) nel caso in cui la costruzione del programma oggetto richieda l'unione di *più moduli* (compilati separatamente), il linker provvede a **collegarli** formando un unico *programma eseguibile*.
- **Debugger:** ("*spulciatore*") consente di **eseguire passo-passo** un programma, **controllando via via quel che succede**, al fine di **scoprire ed eliminare errori** non rilevati in fase di compilazione.

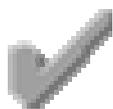
---

Fondamenti di Informatica T

## AMBIENTI DI PROGRAMMAZIONE (3)

### II° CASO: INTERPRETAZIONE

- **Interprete:** *traduce ed esegue* direttamente *ciascuna istruzione* del *programma sorgente*, *istruzione per istruzione*.  
È alternativo al compilatore (raramente sono presenti entrambi).



**Traduzione ed esecuzione sono *intercalate*, e avvengono *istruzione per istruzione*.**

---

Fondamenti di Informatica T

## PERSONAL COMPUTER

**PC** (ex "IBM-COMPATIBILI")

Usano processori della famiglia *Intel 80x86*:

- 8086
- 80286
- ...
- Pentium
- Pentium MMX
- Pentium II
- Pentium III
- Pentium IV



**Le prestazioni dipendono da:**

- frequenza dell'orologio di sistema (*clock*)
- dimensione della RAM
- velocità/parallelismo delle linee dati/comandi (*bus*)

---

*Fondamenti di Informatica T*

## ALTRI SISTEMI DI CALCOLO

### **Workstation**

sistemi con capacità di supportare più attività contemporanee, spesso dedicati a più utenti. Prestazioni normalmente superiori a quello di un tipico Personal Computer.

### **Mainframe**

Macchine capaci di servire decine di utenti contemporaneamente, collegati tramite terminali

### **Super-calcolatori**

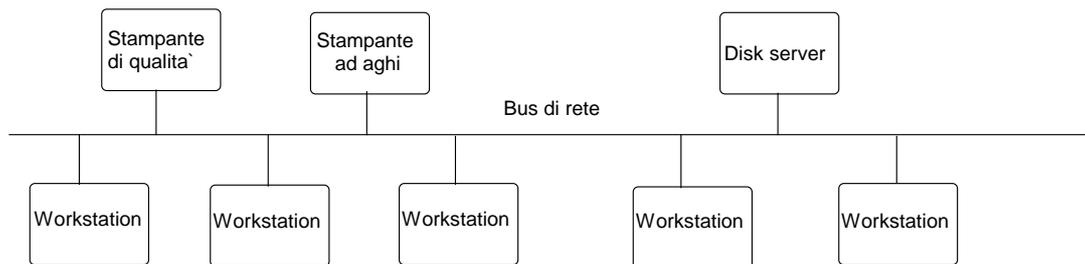
Hanno molti processori, grandi memorie di massa e servono tipicamente centinaia o migliaia di terminali.

---

*Fondamenti di Informatica T*

## RETI DI CALCOLATORI

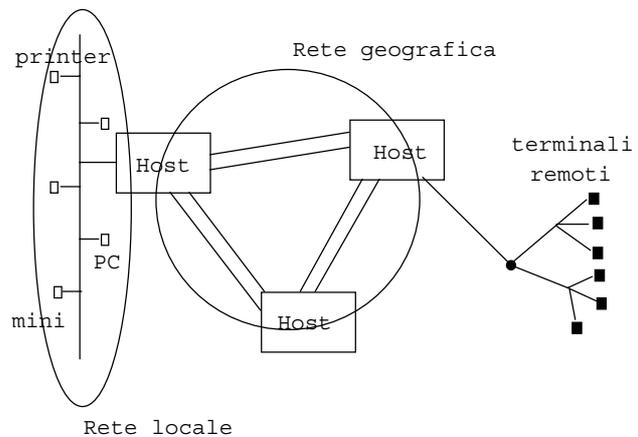
- **Reti Locali:**  
collegano elaboratori *fisicamente vicini* (nello stesso ufficio o stabilimento).
- **LAN (Local Area Network)**



Fondamenti di Informatica T

## RETI DI CALCOLATORI (segue)

- **Reti geografiche:**  
collegano elaboratori medio-grandi situati anche a *grande distanza*.
- **WAN (Wide Area Network)**



Fondamenti di Informatica T

## INTERNET: la rete delle reti

- **Internet:** la rete risultante dalla interconnessione mondiale di tutte le reti.
- Milioni di elaboratori ("siti") collegati a **ragnatela**
- **World-Wide Web (WWW)**

