

# Note sull'architettura del calcolatore

---

## Informatica

---

- ◆ Scienza che studia la *rappresentazione* e l'*elaborazione* dell'informazione

# Algoritmo

---

## ◆ Realizzazione di un “compito”

Preparazione di una torta

Programmazione del DVD-recorder

Massimo Comun Divisore di due numeri

## ◆ Algoritmo

Sequenza ordinata di istruzioni, precisa e comprensibile, che porta alla realizzazione del compito

◆ PASSO 1: Fai qualcosa

◆ PASSO 2: Fai qualcosa

◆ :

◆ PASSO n: Fermati, hai finito

3

Ultimo aggiornamento: 26/02/2007

# Problema: consumo medio (km/l)

---

PASSO 1: **ACQUISISCI** i valori per *litri utilizzati*, *km alla partenza* e *km all'arrivo*

PASSO 2: **PONI** il valore di *distanza percorsa* a (*km all'arrivo* – *km alla partenza*)

PASSO 3: **PONI** il valore di *km al litro* a (*distanza percorsa / litri utilizzati*)

PASSO 4: **STAMPA** il valore di *km al litro*

PASSO 5: **FERMATI**

4

Ultimo aggiornamento: 26/02/2007

# Problema: Equazioni di 2° grado

---

$$ax^2 + bx + c = 0$$



5

Ultimo aggiornamento: 26/02/2007

# Algoritmo: Equazioni di 2° grado

---

PASSO 1: **ACQUISISCI** i valori per  $a$ ,  $b$  e  $c$

PASSO 2: **PONI** il valore di  $\Delta$  a  $b^2 - 4ac$

PASSO 3: **SE** il valore di  $\Delta$  è minore di zero allora

PASSO 4: **STAMPA** "Non esistono soluzioni reali"

**ALTRIMENTI**

PASSO 5: **SE** il valore di  $\Delta$  è uguale a zero allora

PASSO 6: **STAMPA** "Due soluzioni reali coincidenti"

PASSO 7: **PONI** il valore di  $x$  a  $(-b/2a)$

PASSO 8: **STAMPA** il valore di  $x$

**ALTRIMENTI**

PASSO 9: **STAMPA** "Due soluzioni reali distinte"

PASSO 10: **PONI** il valore di  $x_1$  a  $(-b - \sqrt{\Delta})/2a$

PASSO 11: **PONI** il valore di  $x_2$  a  $(-b + \sqrt{\Delta})/2a$

PASSO 12: **STAMPA** il valore di  $x_1$  e di  $x_2$

PASSO 13: **FERMATI**

6

Ultimo aggiornamento: 26/02/2007

# Problema: Test di Primalità

---

Il numero  $n$  è primo?



Ultimo aggiornamento: 26/02/2007

7

# Algoritmo: Test di Primalità

---

- PASSO 1: **ACQUISISCI** il valore per  $n$   
PASSO 2: **PONI** il valore di *primo* a "sì"  
PASSO 3: **PONI** il valore di  $d$  a 2  
PASSO 4: **RIPETI** i passi da 5 a 7 fintanto che il valore di *primo* è "sì" e  $d$  è minore di  $n$   
PASSO 5:     **SE**  $n$  è divisibile per  $d$   
PASSO 6:     **PONI** il valore di *primo* a "no"  
          **ALTRIMENTI**  
PASSO 7:     **PONI** il valore di  $d$  a  $(d+1)$   
PASSO 8: **SE** il valore di *primo* è "sì"  
PASSO 9:     **STAMPA** "È primo!"  
          **ALTRIMENTI**  
PASSO 10:    **STAMPA** "Non è primo!"  
PASSO 11: **FERMATI**

Ultimo aggiornamento: 26/02/2007

8

# Istruzioni (I)

---

## ◆ Istruzioni di elaborazione

**PONI** il valore della *variabile a espressione aritmetica*

**PONI** il valore della *variabile a “valore”*

## ◆ Istruzioni di trasferimento

ingresso

**ACQUISISCI** il *valore della variabile, ...*

uscita

**STAMPA** il *valore di variabile, ...*

**STAMPA** il *“messaggio”*

9

Ultimo aggiornamento: 26/02/2007

# Istruzioni (II)

---

## ◆ Istruzioni di controllo

### Istruzioni alternative (condizionali)

**SE** una *condizione* è vera allora

sequenza di istruzioni

**ALTRIMENTI**

sequenza di istruzioni

### Istruzioni ripetitive (cicliche)

**RIPETI** finché una *condizione* è vera

sequenza di istruzioni

10

Ultimo aggiornamento: 26/02/2007

# Algoritmo ed Automazione

---

- ◆ Se siamo capaci di specificare un algoritmo per risolvere un problema, allora
- ◆ siamo in grado di **automatizzare** la soluzione
- ◆ Bisogna saper costruire una macchina capace di effettuare i passi dell'algoritmo

11

Ultimo aggiornamento: 26/02/2007

# Algoritmo e calcolatore

---

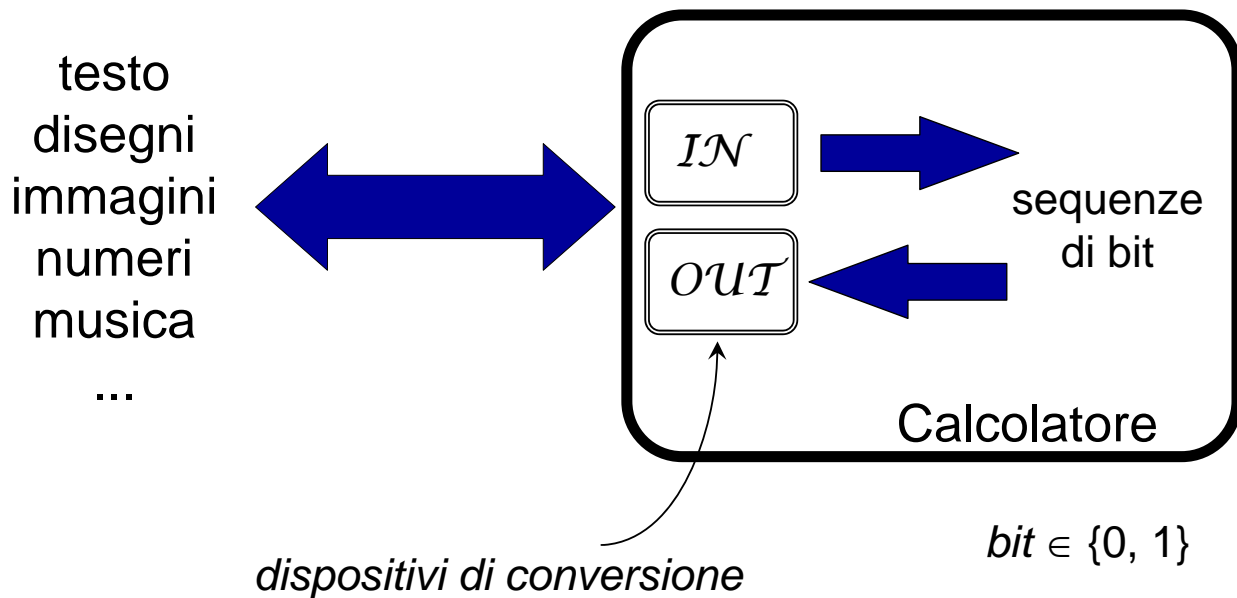
- ◆ *Calcolatore: macchina complessa in grado di elaborare informazioni in forma codificata*
- ◆ Un **calcolatore** è un esecutore di algoritmi in cui
  - un algoritmo viene descritto per mezzo di un **programma** ed
  - il programma è una sequenza di istruzioni espresse in un linguaggio "comprensibile" al calcolatore (**linguaggio di programmazione**)

12

Ultimo aggiornamento: 26/02/2007

# Calcolatore e Informazione

In un calcolatore i dati e le istruzioni sono codificati in forma **binaria**

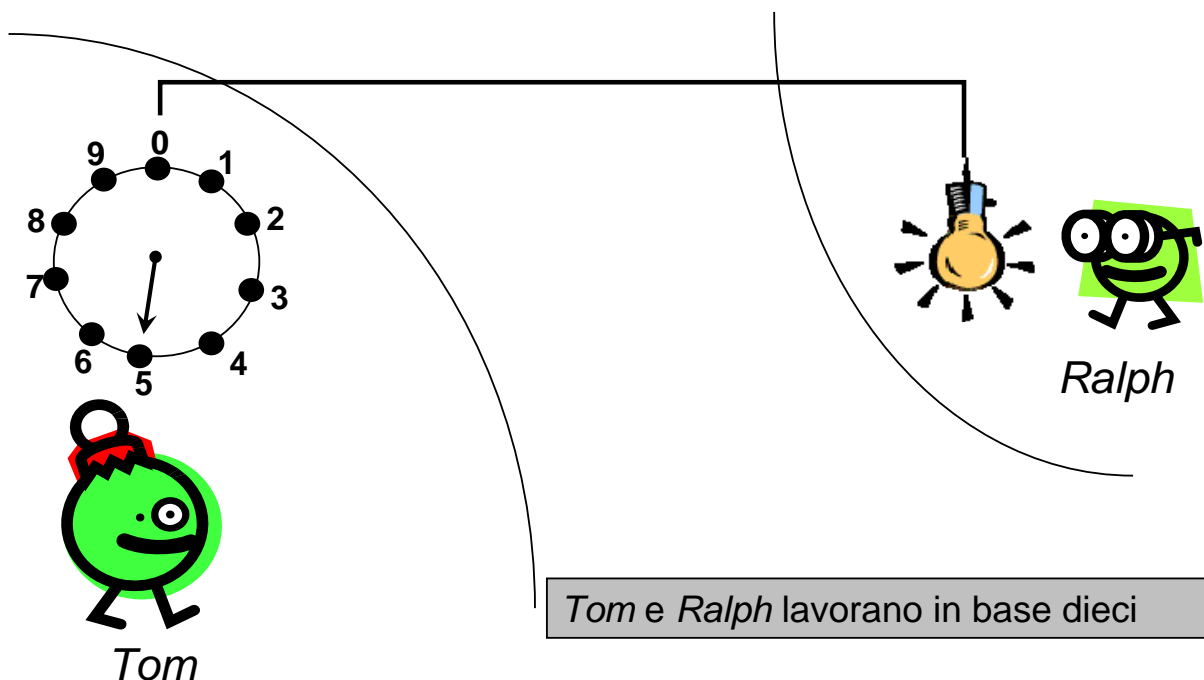


13

Ultimo aggiornamento: 26/02/2007

## Perché i bit (I)

*Tom* vuole trasmettere a *Ralph* un'informazione numerica {0..9}

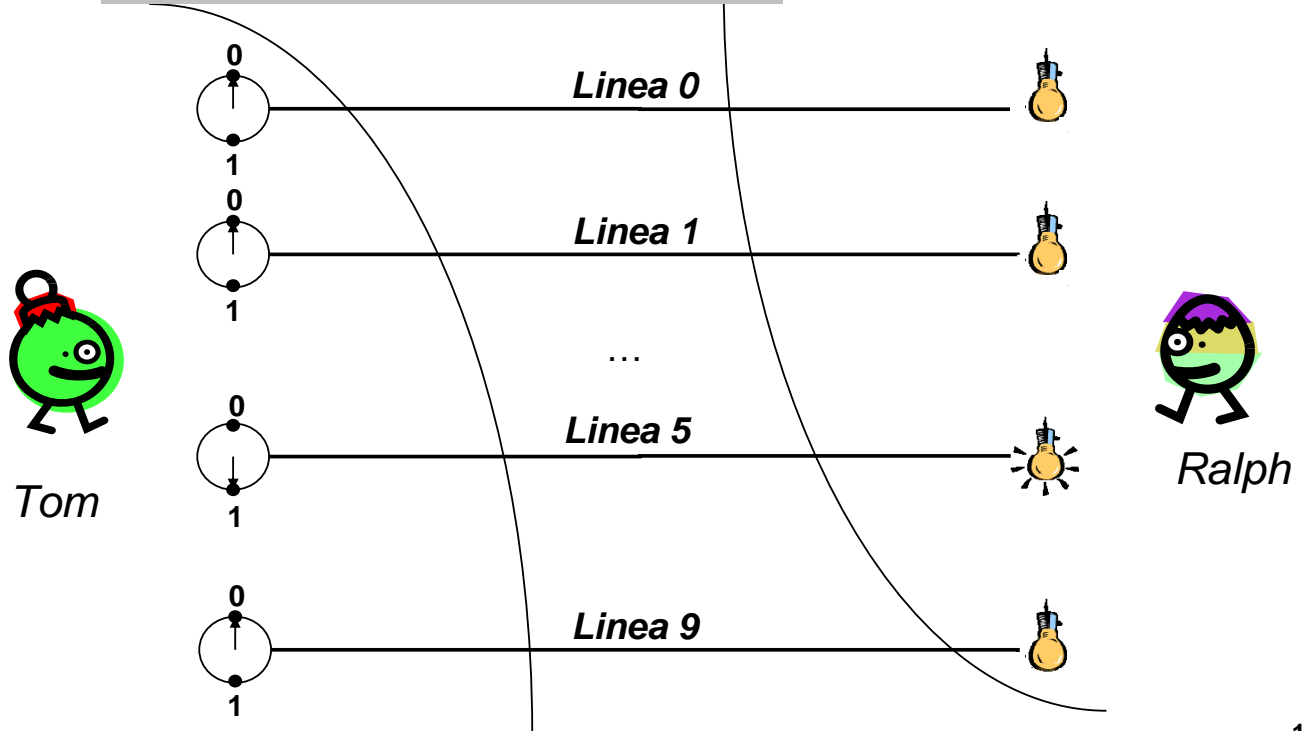


14

Ultimo aggiornamento: 26/02/2007

# Perché i bit (II)

Una soluzione migliore

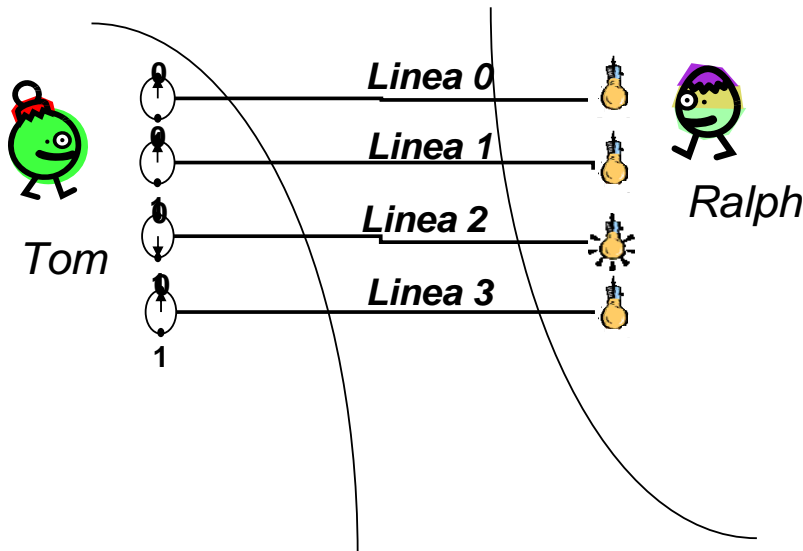


15

Ultimo aggiornamento: 26/02/2007

# Perché i bit (III)

Una soluzione ancora migliore



Cifra	Linea			
	3	2	1	0
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

16

Ultimo aggiornamento: 26/02/2007



# Rappresentazione Naturali (I)

## BASE DIECI

◆ Cifre: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

◆ Rappresentazione posizionale

$$(123)_{dieci} \text{ significa } 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

## BASE DUE

◆ Cifre: 0, 1

◆ Rappresentazione posizionale

$$(11001)_{due} \text{ significa}$$

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (25)_{dieci}$$

17

Ultimo aggiornamento: 26/02/2007

# Rappresentazione Naturali (II)

## ESPRESSIONE DI UN NUMERO NATURALE IN BASE DUE

Algoritmo del mod e div

Esempio:  $n = 47$

inizio

	QUOZIENTE	RESTO
	47	1
div 2	23	1
div 2	11	1
div 2	5	1
div 2	2	0
div 2	1	1
div 2	0	—

fine

$(101111)_{due}$

18

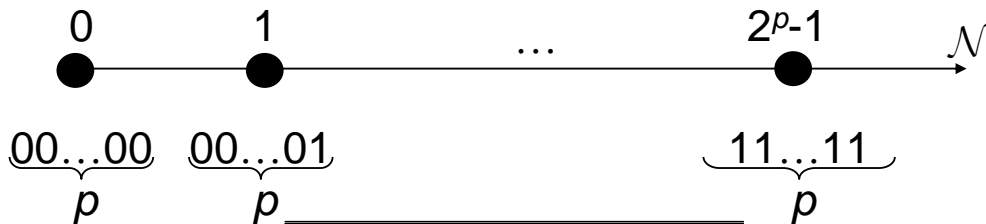
Ultimo aggiornamento: 26/02/2007

# Rappresentazione Naturali (III)

## ALCUNE PROPRIETÀ

Potenza della Base:  $2^p \leftrightarrow (1\underbrace{00\dots00}_p)_{due}$

Intervallo di rappresentabilità con  $p$  bit



$p$	Intervallo
8	[0, 255]
16	[0, 65535]
32	[0, 4294967295]

19

Ultimo aggiornamento: 26/02/2007

## Il problema del riporto

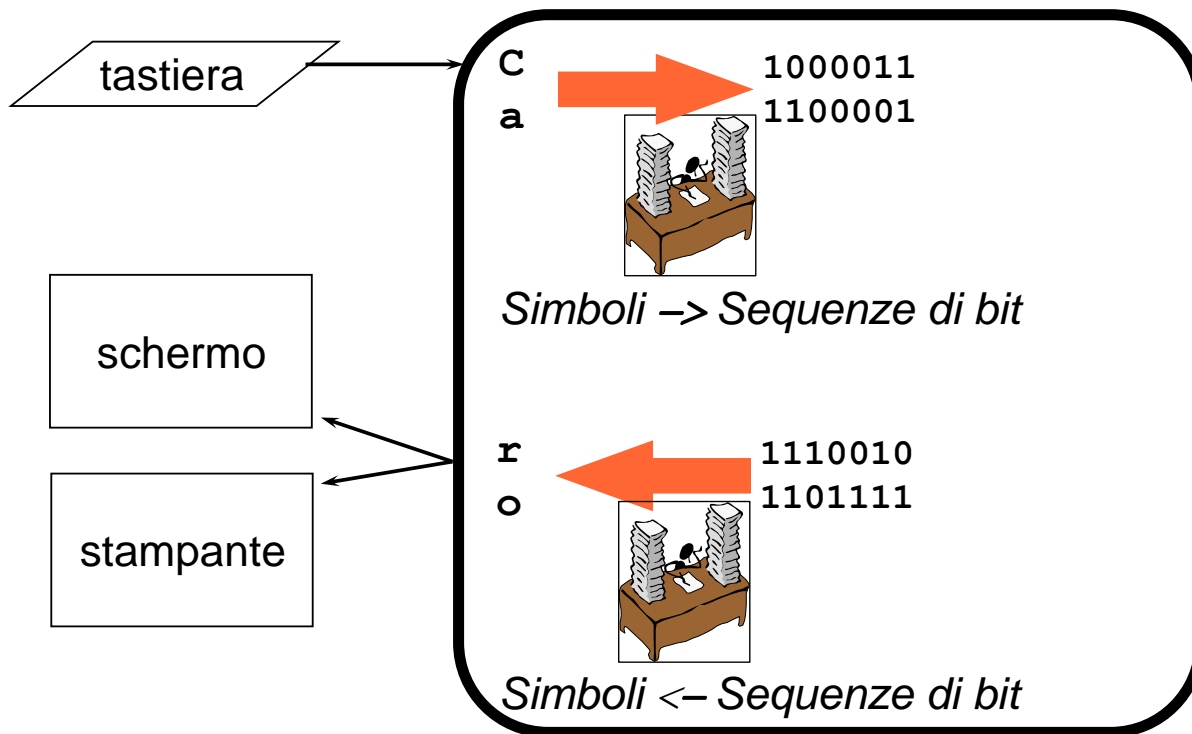
IL CALCOLATORE LAVORA CON NUMERO FINITO DI BIT

- ◆  $p = 16$  bit
- ◆  $A = 0111011011010101$  (30421)  
 $B = 1010100001110111$  (43127)
- ◆  $A + B$  è maggiore di  $2^p-1$  (65535) quindi non è rappresentabile su  $p$  bit, allora si dice che la somma ha dato luogo ad un riporto (*carry*)
- ◆ In generale, ci vogliono  $p+1$  bit per rappresentare la somma di due numeri di  $p$  bit

20

Ultimo aggiornamento: 26/02/2007

# Rappresentazione del testo

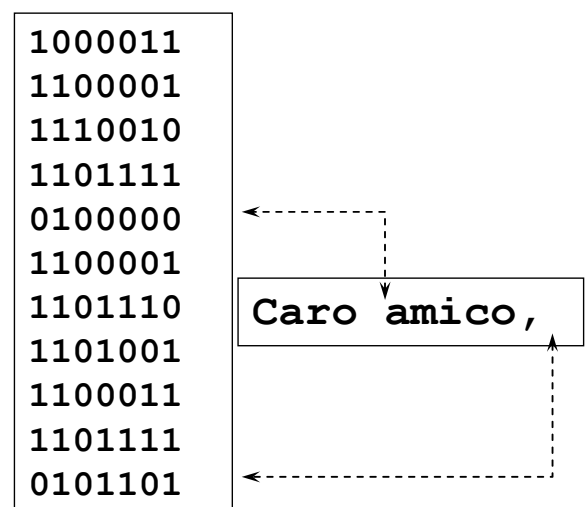


21

Ultimo aggiornamento: 26/02/2007

# Rappresentazione Testo (ASCII)

Simbolo	N.	Sequenze di bit
...	...	...
0	48	0110000
1	49	0110001
2	50	0110010
...	...	...
9	58	0111001
...	...	...
A	65	1000001
B	66	1000010
C	67	1000011
...	...	...
a	97	1100001
b	98	1100010
...	...	...
~	126	1111110
△	127	1111111



22

Ultimo aggiornamento: 26/02/2007

# Calcolatore e periferiche

---

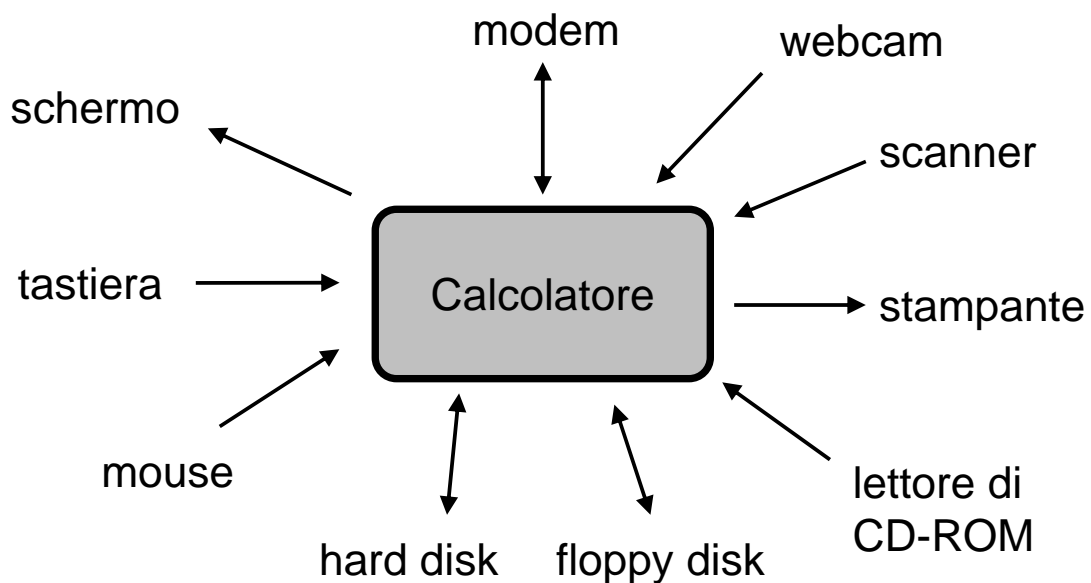


23

Ultimo aggiornamento: 26/02/2007

# Calcolatore e periferiche

---



24

Ultimo aggiornamento: 26/02/2007

# Calcolatori e periferiche

## ◆ Reti di calcolatori

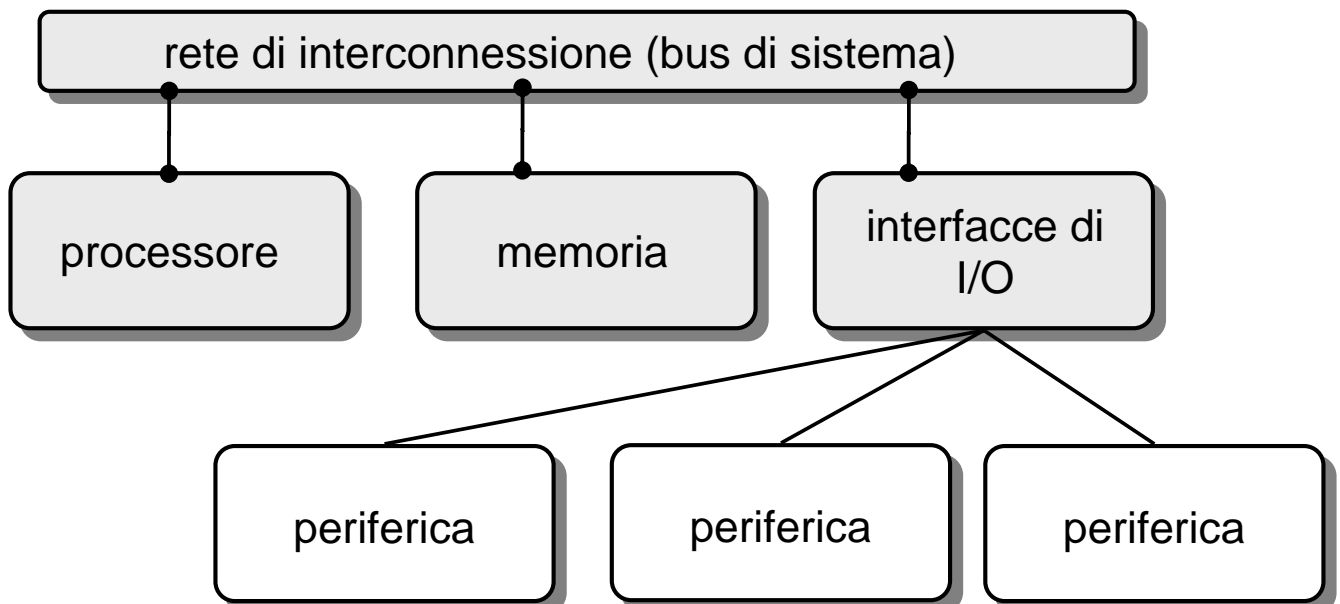


25

Ultimo aggiornamento: 03/03/2008

# Struttura logica

## Architettura di von Neumann (1946)



26

Ultimo aggiornamento: 26/02/2007

# Funzionamento (schema)

◆ La memoria contiene dati e programmi (istruzioni)

◆ Il processore ripete all'infinito le seguenti azioni:

- preleva una nuova istruzione dalla memoria
- la decodifica (capisce cosa deve fare!)
- la esegue (lo fa!)

L'esecuzione di un'istruzione può comportare

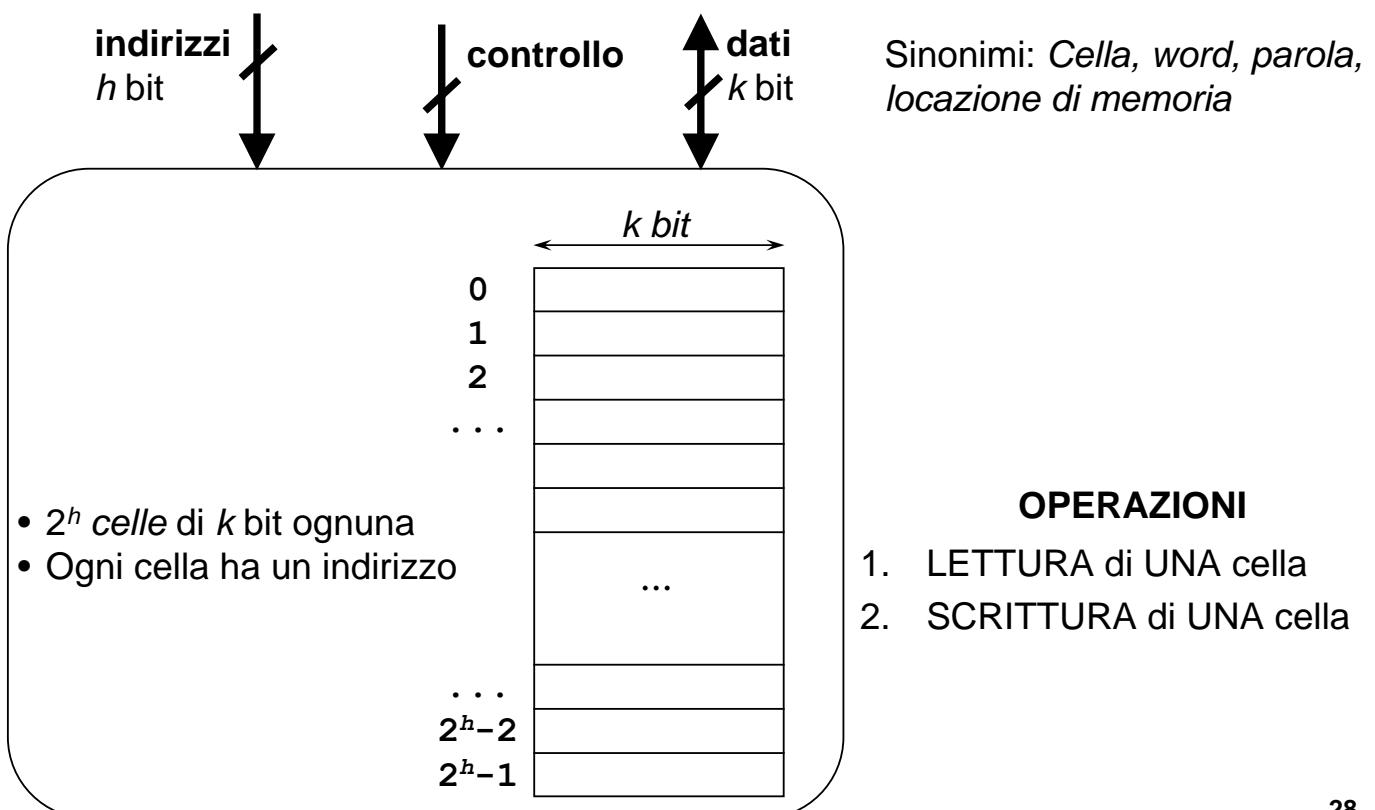
Elaborazione e/o Trasferimento (memoria ↔ processore,  
I/O ↔ processore)

◆ Le periferiche permettono al calcolatore di interagire con il mondo esterno

27

Ultimo aggiornamento: 26/02/2007

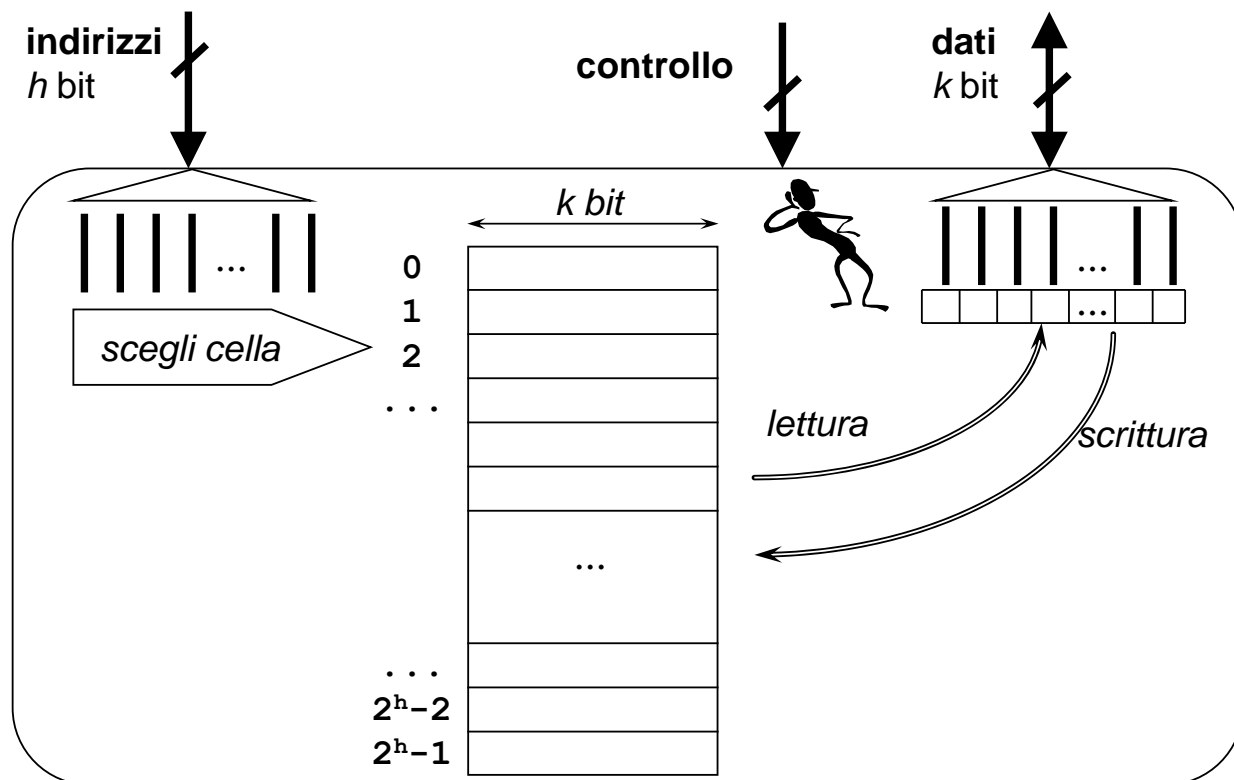
## Struttura logica della memoria (I)



Ultimo aggiornamento: 26/02/2007

28

# Struttura logica della memoria (II)



29

Ultimo aggiornamento: 26/02/2007

## Dimensione massima della memoria

- Valori tipici per  $k$  (in bit): 8 (byte), 16, 32, 64
- $h$  bit per gli indirizzi  $\Rightarrow 2^h$  celle

$h$	Numero massimo di celle ( $2^h$ )	
16	65 536	64K
20	1 048 576	1M
22	4 194 304	4M
24	16 777 216	16M
32	4 294 967 296	4G

$$2^{10} = 1024 = 1K \text{ (kilo)}$$

$$2^{20} = 1\,048\,576 = 1 \text{ M (mega)}$$

$$2^{30} = 1\,073\,741\,824 = 1G \text{ (giga)}$$

$$2^{40} = 1\,099\,511\,627\,776 = 1T \text{ (tera)}$$

30

Ultimo aggiornamento: 26/02/2007

## ...tanto per fissare le idee

---

<b>Dimensione memoria</b>	<b>Ordine di grandezza</b>	<b>Informazione testuale</b>
1 byte	$10^0$	un carattere
1 kilobyte	$10^3$	una pagina scritta
1 megabyte	$10^6$	due o tre racconti
1 gigabyte	$10^9$	una biblioteca dipartimentale
1 terabyte	$10^{12}$	la biblioteca di qualche università

31

Ultimo aggiornamento: 26/02/2007

## RAM e ROM

---

### ◆ Random Access Memory (RAM)

- Ogni cella può essere acceduta indipendentemente dalle altre
- Il tempo di accesso è costante (30-70 ns)
- Volatile
- Costi: ~2 centesimi per Mbyte

### ◆ Read-Only Memory (ROM): unità RAM su cui si può *solo* leggere

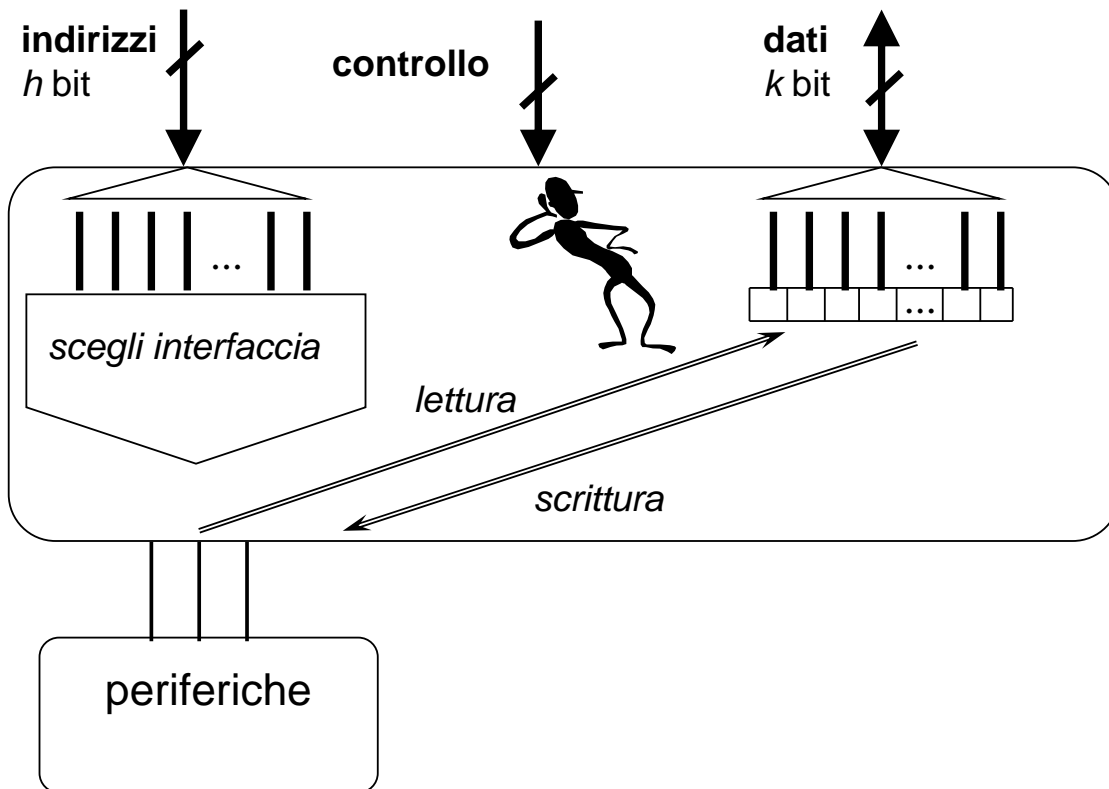
- Memoria protetta e permanente
- EROM, PROM, EPROM

32

Ultimo aggiornamento: 10/03/2007



# Struttura logica dello spazio di I/O



Ultimo aggiornamento: 26/02/2007

33

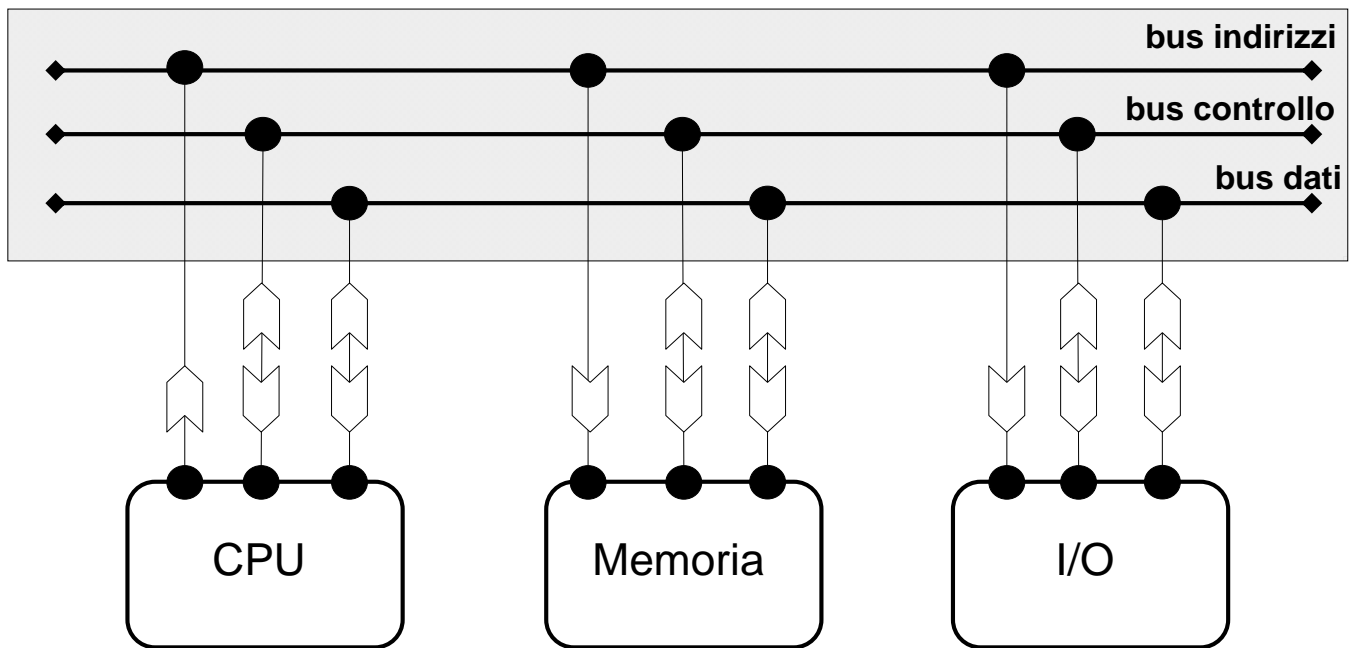
## Trasferimento dell'informazione

- ◆ Il bus è un insieme di *fili conduttori*, su ciascuno dei quali può essere impostata una *tensione*
- ◆ Su ciascun filo del bus, le tensioni "lecite" sono soltanto due
  - Tensione "alta" (5V, 3.3V)
  - Tensione "bassa" (0V)
- ◆ A ciascuna di queste tensioni è associato un **valore binario** (ad esempio: 5V=1, 0V=0)
- ◆ Possiamo quindi dire che, in un dato istante, su ciascuno dei fili del bus è presente un **bit**

Ultimo aggiornamento: 26/02/2007

34

# Struttura logica del Bus



35

Ultimo aggiornamento: 26/02/2007

## Le linee del bus

- ◆ **BUS DATI.** Fili su cui vengono trasferiti i dati tra l'elaboratore e le altre componenti
- ◆ **BUS INDIRIZZI.** Fili che specificano l'indirizzo della sorgente o della destinazione dei dati che si trovano sul Bus Dati.
- ◆ **BUS CONTROLLO.** Fili che specificano l'accesso e l'uso del Bus Dati e del Bus Indirizzi. Esempi:
  - Lavora:** filo che indica che bisogna lavorare
  - Memoria - I/O:** filo che indica "chi" deve lavorare
  - Leggi - Scrivi:** filo che indica "cosa" bisogna fare
  - Fatto:** filo che indica che il lavoro è stato fatto
  - Clock, Alimentazione...**

36

Ultimo aggiornamento: 26/02/2007

# Esempio di Temporizzazione

clock

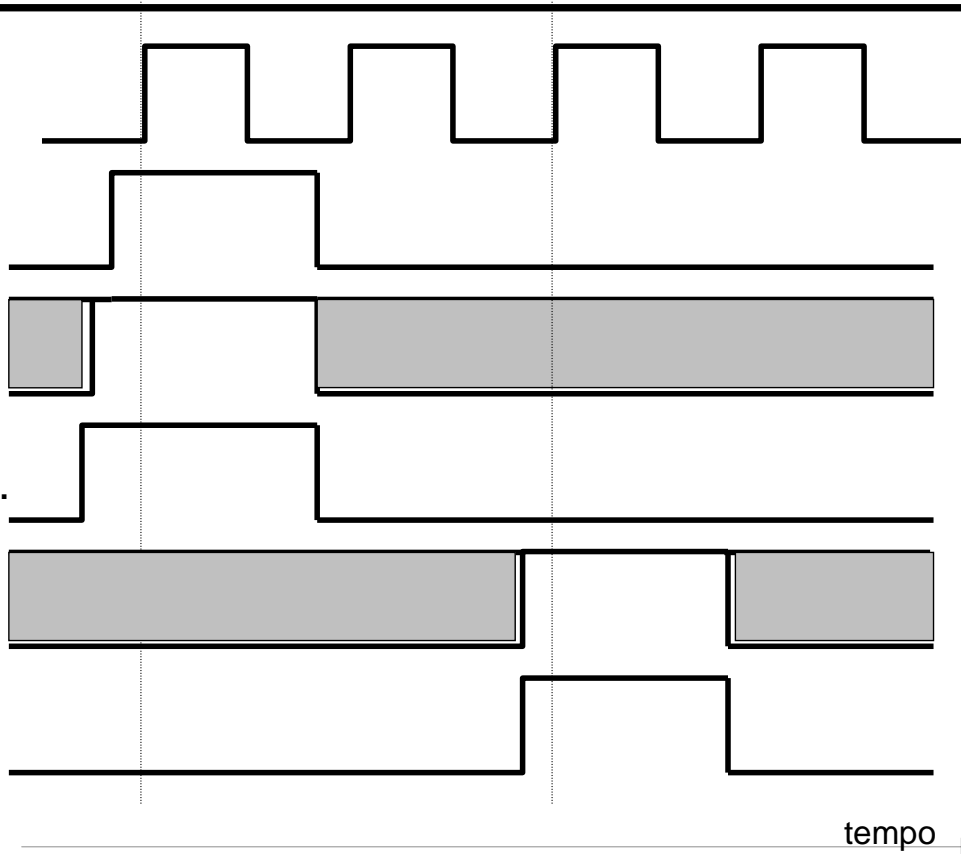
lavora

indirizzo

lett./scritt.

dati

fatto

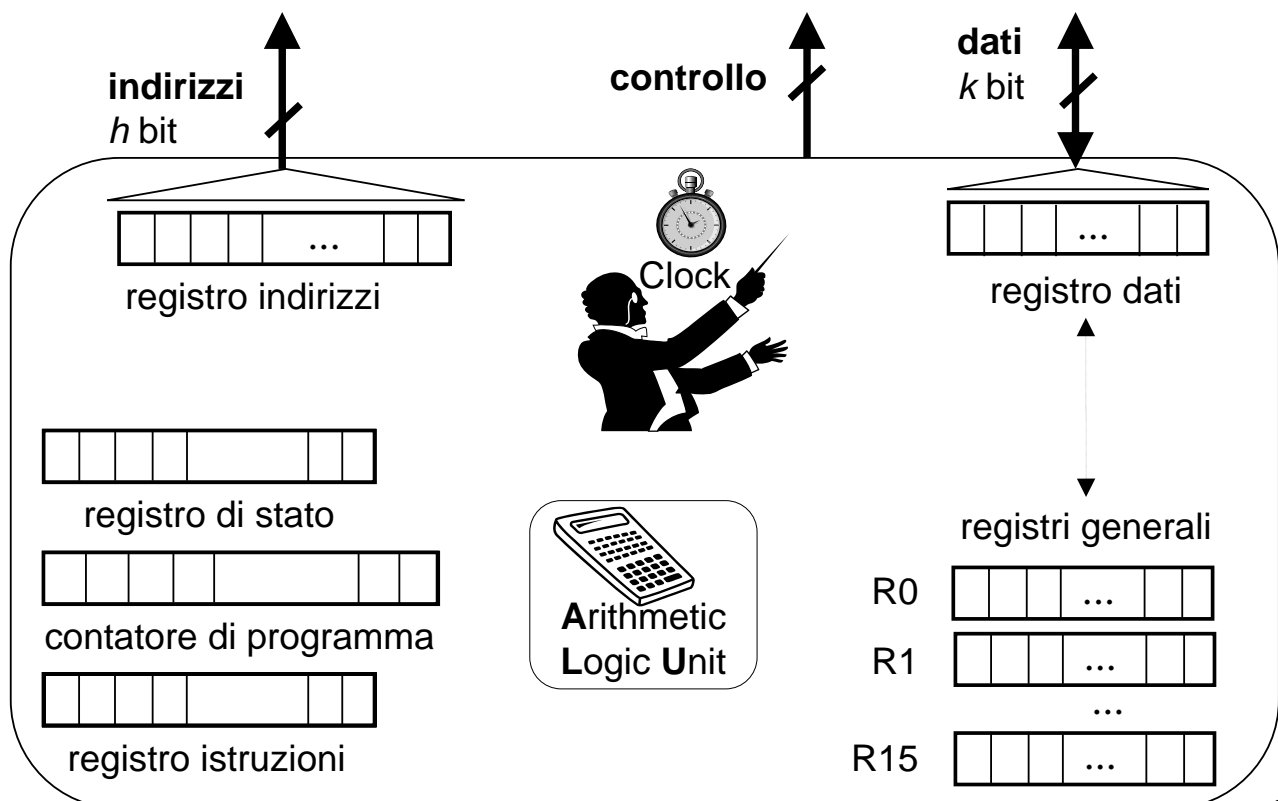


tempo

37

Ultimo aggiornamento: 26/02/2007

# Struttura logica del processore



38

Ultimo aggiornamento: 10/03/2007

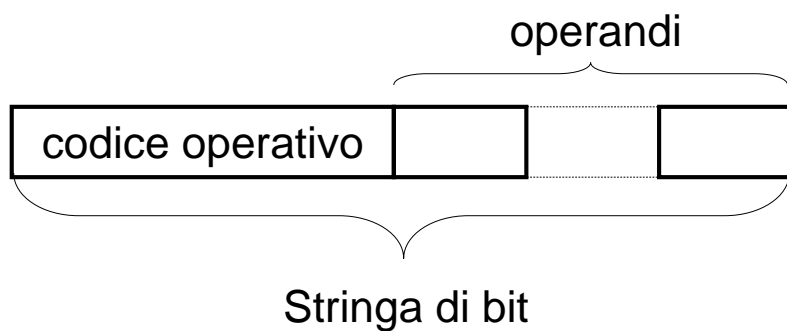
# Programmi ed Istruzioni

---

La funzione principale di un calcolatore consiste nell'esecuzione da parte del processore di un programma memorizzato in memoria

Un **programma** è una sequenza di istruzioni

Un **istruzione** è un'informazione codificata che specifica "cosa fare" e "dove" farlo



## TIPI DI ISTRUZIONE

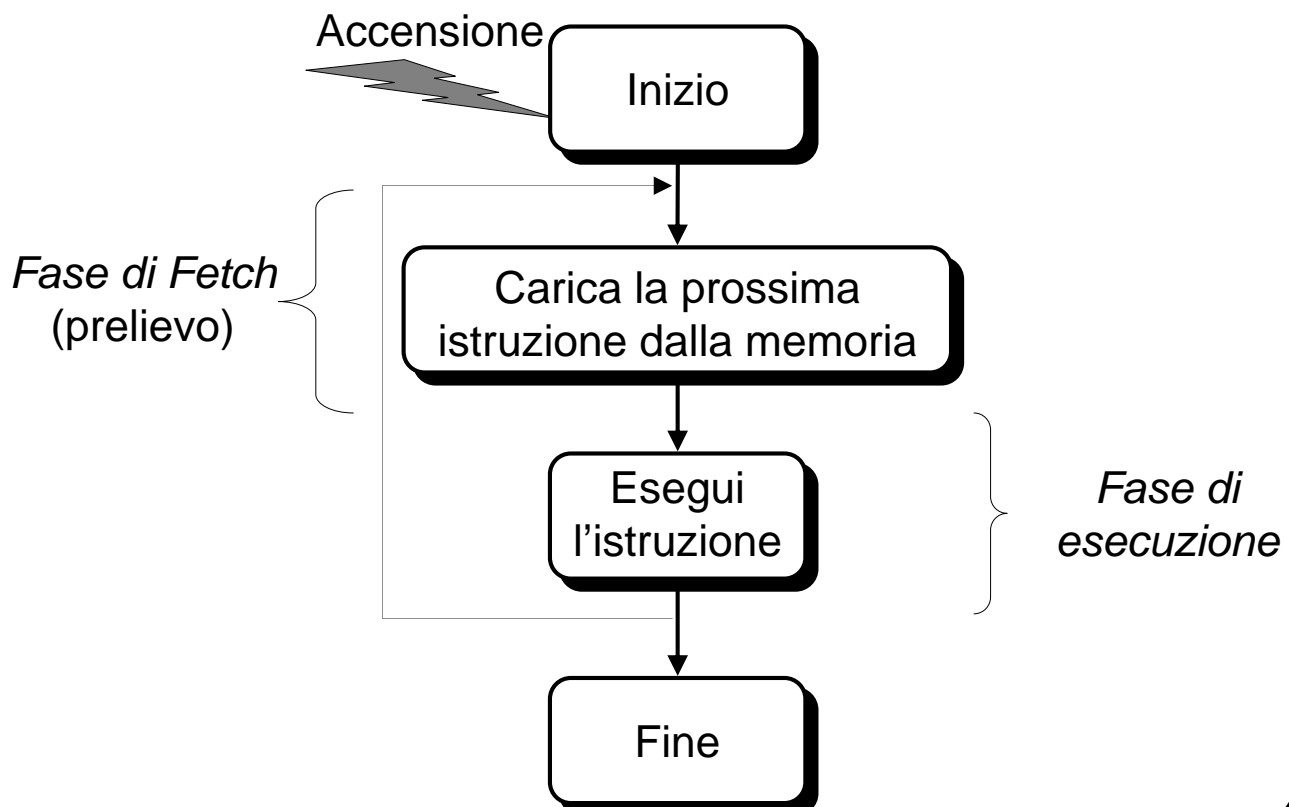
- Trasferimento
- Elaborazione
- Controllo

39

Ultimo aggiornamento: 10/03/2007

# Ciclo della CPU

---



40

Ultimo aggiornamento: 10/03/2007

# Fase di Esecuzione

---

- La fase di Esecuzione consiste in due sottofasi:
  1. Per prima cosa la CPU *decodifica* l'istruzione, cioè capisce cosa deve fare e dove deve farlo.
  2. Poi la CPU *esegue* l'istruzione, cioè fa quello che ha capito al passo precedente.
- Il passaggio dalla Fase di Esecuzione alla Fase Fine viene effettuato eseguendo un'istruzione particolare.

41

Ultimo aggiornamento: 10/03/2007

# Processore di esempio

---

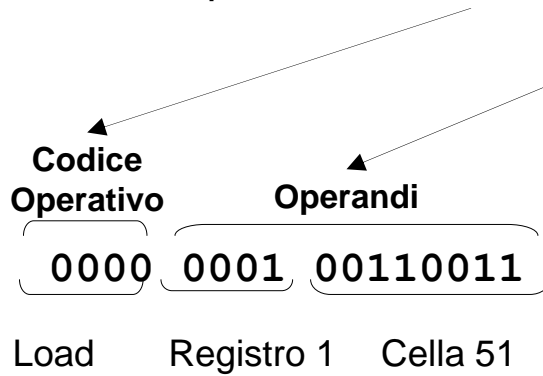
- ◆ Quello visto nei precedenti lucidi, con 16 registri generali R0, R1, ..., R15
- ◆ Il processore sa eseguire **16 operazioni diverse**
- ◆ Inserito in un sistema in cui c'è una memoria RAM di **256 celle**, ciascuna di **16 bit**  
 **$h=8, k=16$**
- ◆ Ogni istruzione (che sta in una cella di memoria) è una **sequenza di 16 bit**

42

Ultimo aggiornamento: 10/03/2007

# Esempio di Istruzione

Un'istruzione specifica: **cosa** fare e **dove** farlo



Carica (load) il contenuto della cella di indirizzo 51 nel registro R1

## Tipi di Istruzioni

16 Istruzioni; 16 Registri; 256 Parole di memoria

### TRASFERIMENTO

**LOAD** *Reg Mem* ( $M \rightarrow CPU$ )

**STORE** *Reg Mem* ( $M \leftarrow CPU$ )

**IN** *Reg IO* ( $I/O \rightarrow CPU$ )

**OUT** *Reg IO* ( $I/O \leftarrow CPU$ )

...

### ELABORAZIONE

**ADD** *Reg\_dest Reg\_sorg*

**SUB** *Reg\_dest Reg\_sorg*

**SET** *Reg\_dest Numero*

...

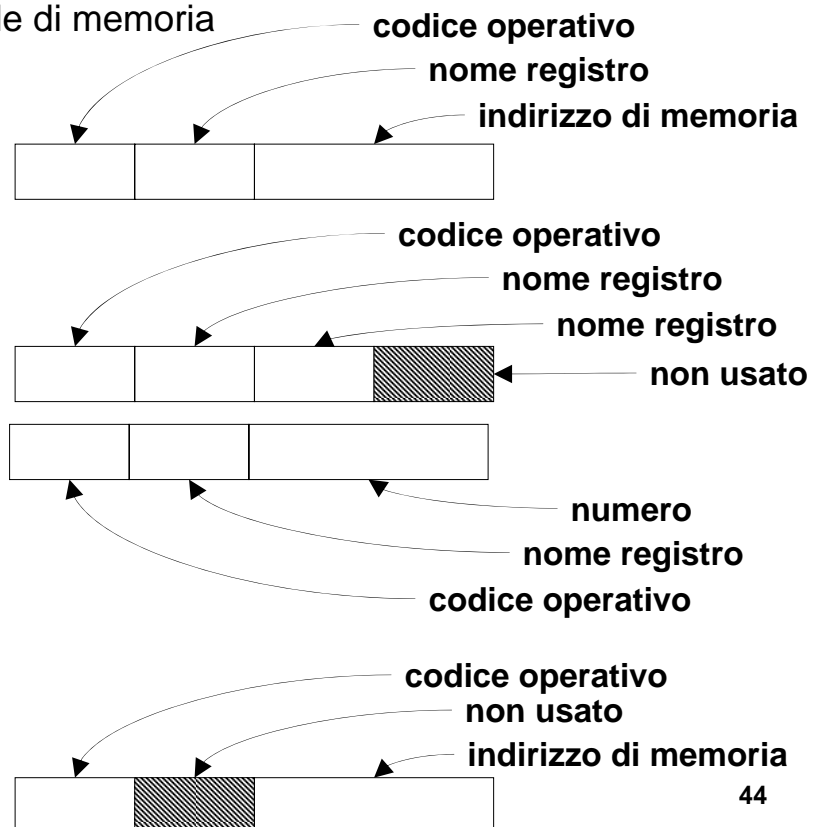
### CONTROLLO

**JMP** *Mem*

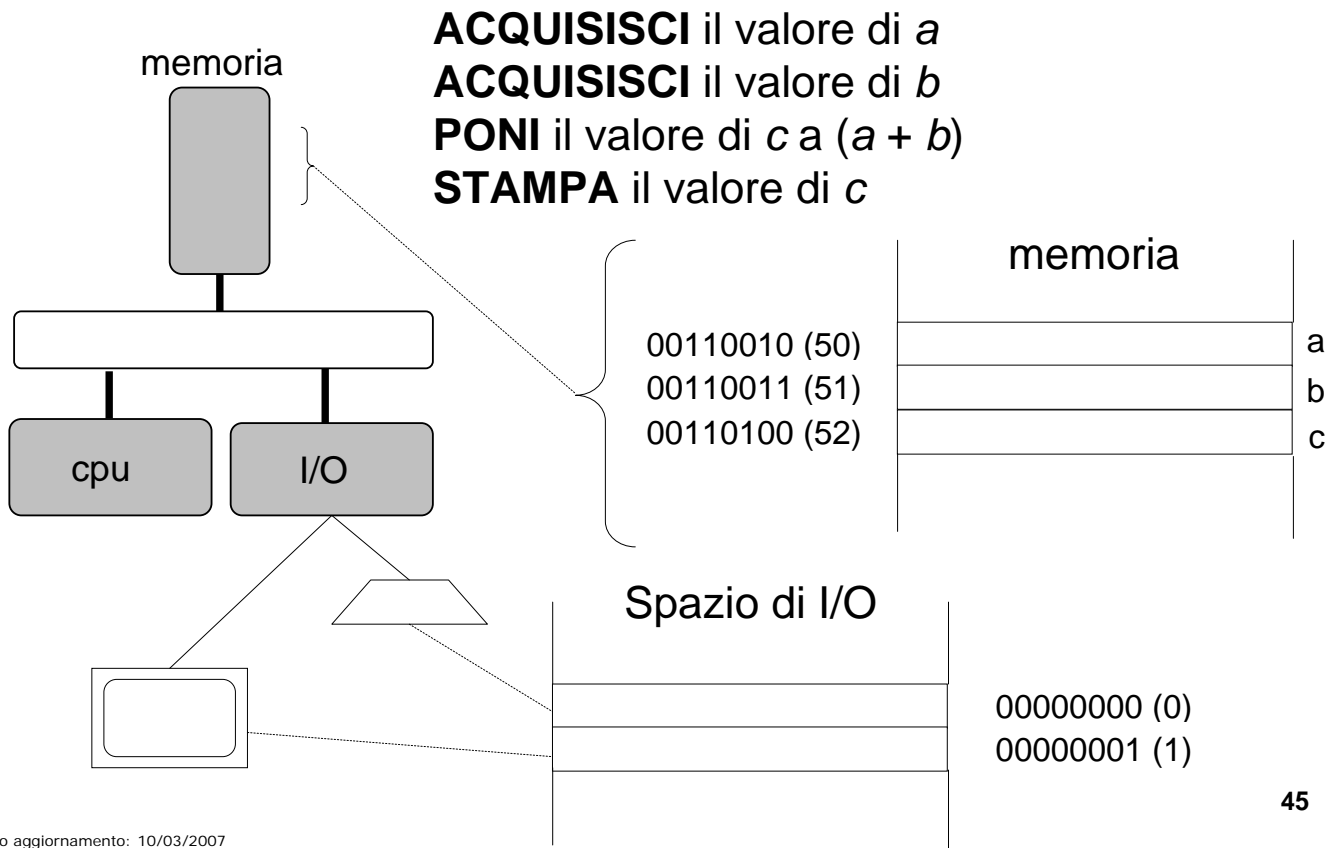
**JZ** *Mem*

**JC** *Mem*

...



# Programma (I)

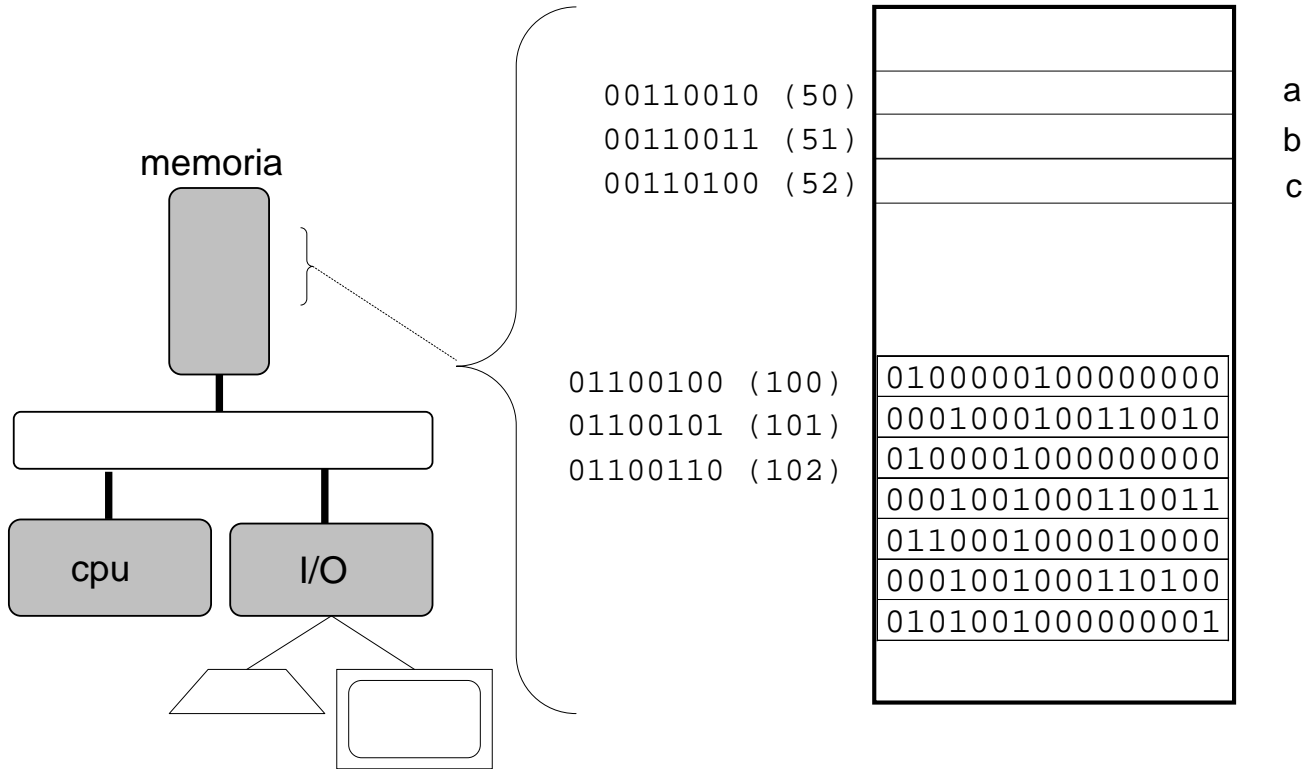


# Programma (II)

**ACQUISISCI** il valore di *a*  
**ACQUISISCI** il valore di *b*  
**PONI** il valore di *c* a (*a + b*)  
**STAMPA** il valore di *c*

IN MEMORIA	MNEMONICO
0100 0001 00000000	IN R1 0
0001 0001 00110010	STORE R1 50
0100 0010 00000000	IN R2 0
0001 0010 00110011	STORE R2 51
0110 0010 0001xxxx	ADD R2 R1
0001 0010 00110100	STORE R2 52
0101 0010 00000001	OUT R2 1

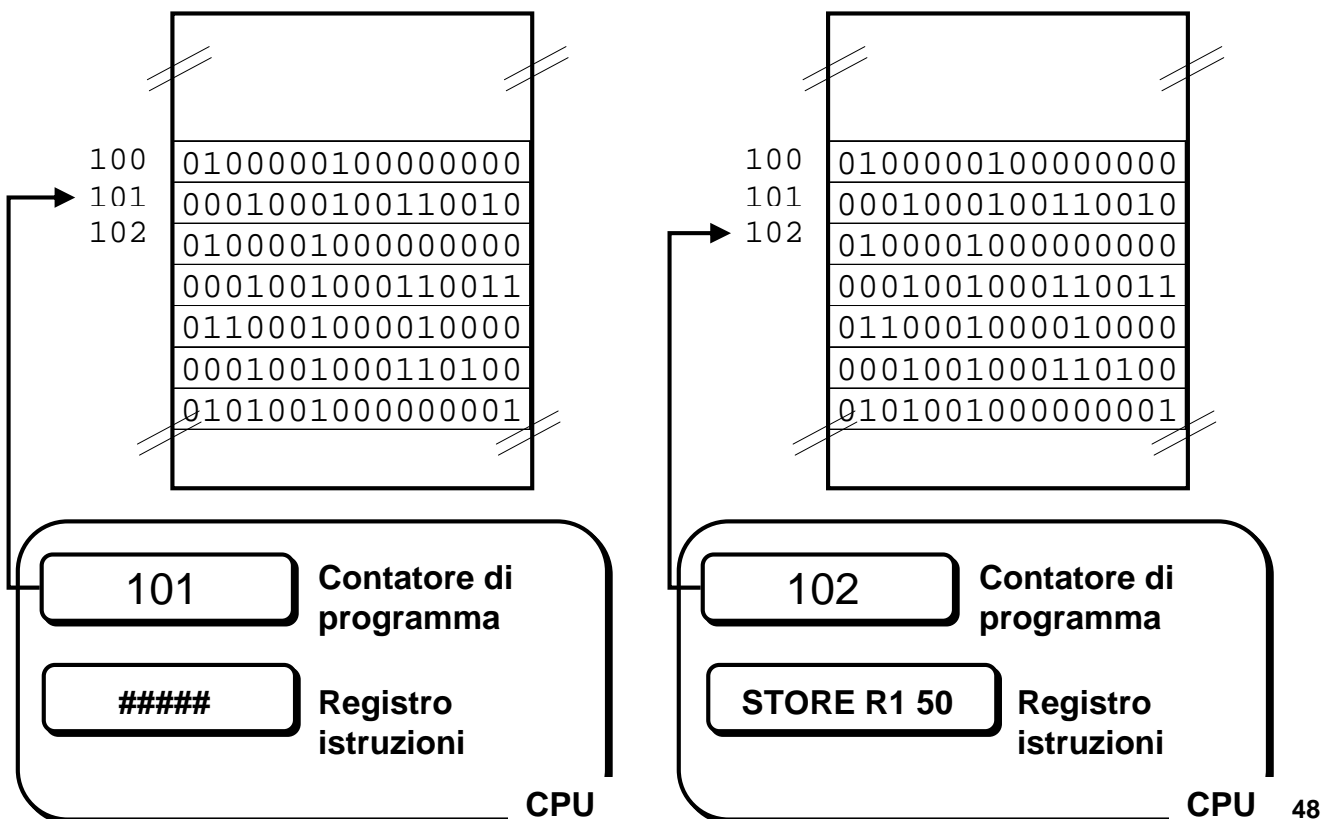
# Programma (III)



47

Ultimo aggiornamento: 10/03/2007

# Fase di fetch



48

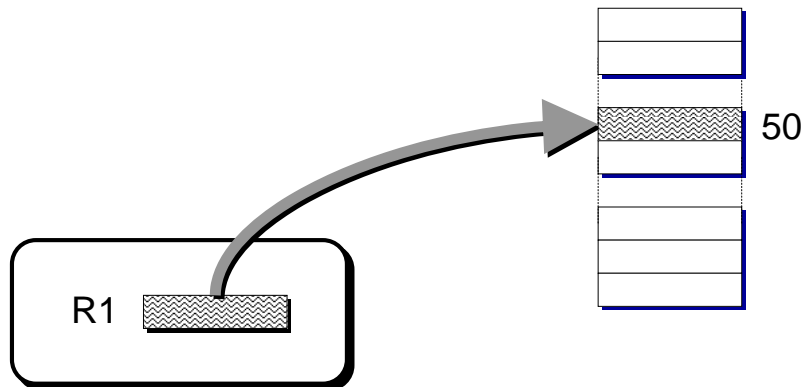
Ultimo aggiornamento: 10/03/2007



# Fase di esecuzione

## Esecuzione di **STORE R1 50**

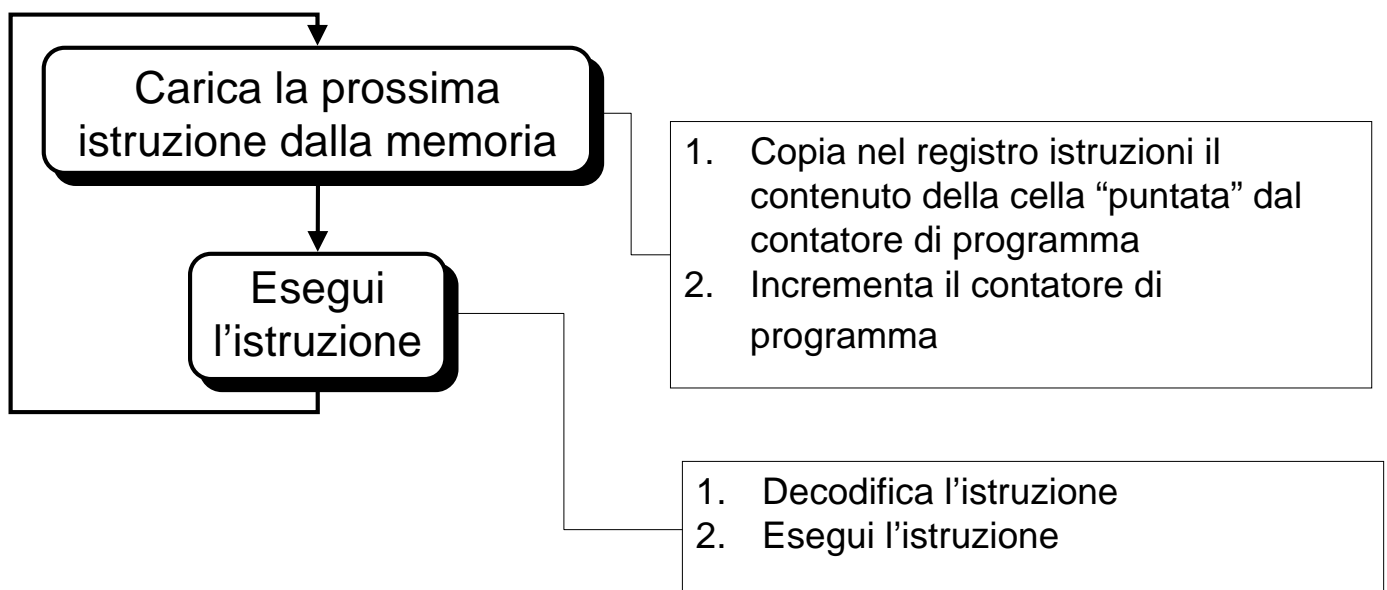
1. Il processore esegue un'operazione di scrittura in memoria all'indirizzo 50 (0011 0010)
2. I contenuti scritti nella cella di memoria 50 sono letti dal registro generale R1



49

Ultimo aggiornamento: 10/03/2007

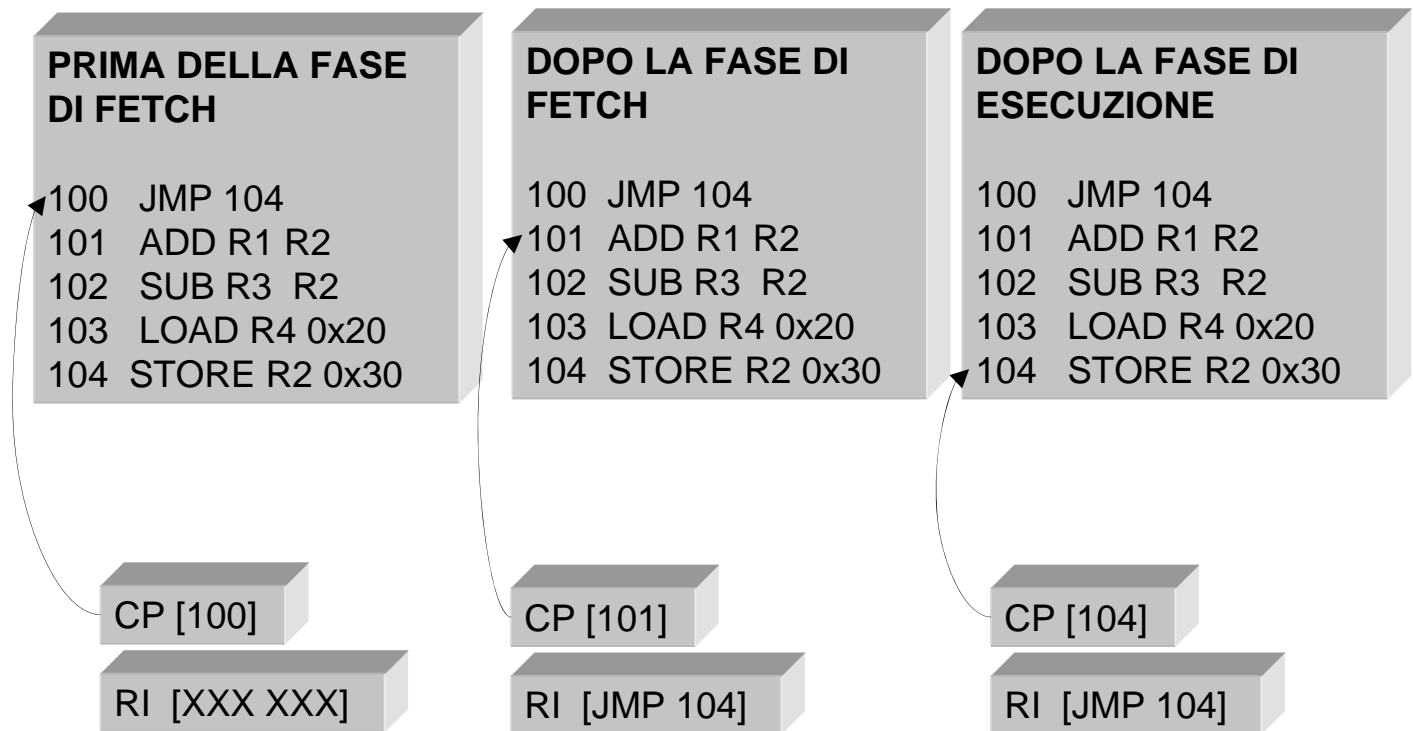
# Ciclo della CPU (ripreso)



50

Ultimo aggiornamento: 10/03/2007

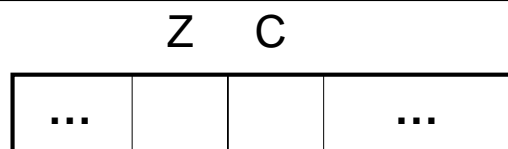
# Istruzioni di controllo



51

Ultimo aggiornamento: 10/03/2007

# Registro di stato



Alcune istruzioni di elaborazione modificano i bit del Registro di stato

- Se l'ultima istruzione di elaborazione che è stata eseguita ha prodotto un valore pari a zero, allora il bit Z vale 1; altrimenti vale 0
- Il bit C riporta il valore del carry (riporto/prestito) dell'ultima istruzione di elaborazione che è stata eseguita
- Alcune istruzioni di controllo (salti condizionati) eseguono il salto se i bit del registro di controllo assumono certi valori
- JZ = salta se il bit Z vale 1
- JC = salta se il bit C vale 1

52

Ultimo aggiornamento: 10/03/2007

# Salto: Esempio I

---

(Supponiamo per ipotesi  $a > 0$ )

**SE**  $b$  è divisibile per  $a$

**PONI** il valore di  $c$  a 1

**ALTRIMENTI**

**PONI** il valore di  $c$  a 0

**Variabile**      **Indirizzo**

$a$                       48

$b$                       49

$c$                       50

INDIRIZZO	ISTRUZIONE
100	LOAD R0 48
101	LOAD R1 49
102	SUB R1 R0
103	JZ 106
104	JC 108
105	JMP 102
106	SET R2 #1
107	JMP 109
108	SET R2 #0
109	STORE R2 50

53

Ultimo aggiornamento: 10/03/2007

# Salto: Esempio II

---

(Supponiamo per ipotesi  $b > 0$ )

1. **ACQUISISCI** il valore di  $a$

2. **ACQUISISCI** il valore di  $b$

3. **RIPETI** i passi 4-5 *finché*  $b > 0$

4. **PONI** il valore di  $a$  ad  $(a + 1)$

5. **PONI** il valore di  $b$  a  $(b - 1)$

6. **PONI** il valore di  $c$  ad  $a$

**Variabile**      **Indirizzo**

$a$                       48

$b$                       49

$c$                       50

INDIRIZZO	ISTRUZIONE
100	IN R1 0
101	STORE R1 48
102	IN R2 0
103	STORE R2 49
104	SET R3 #1
105	ADD R1 R3
106	SUB R2 R3
107	JZ 109
108	JMP 105
109	STORE R1 50
110	OUT R1 1

54

Ultimo aggiornamento: 10/03/2007

# Linguaggi ad alto livello

## Esempio di programma in linguaggio C++: test di primalità

```
#include <iostream.h>

void main() {
    int n;           // numero da testare
    int primo = 1;  // indica se n è primo
    int d = 2;      // divisore corrente

    cin >> n;       // ACQUISISCI

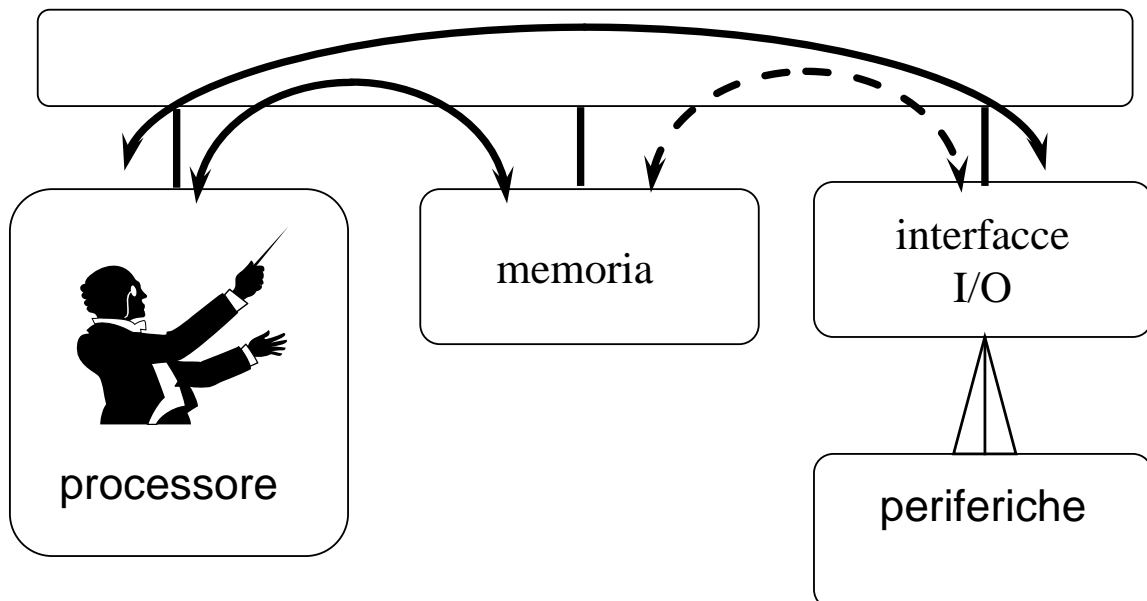
    while ( (d < n) && // RIPETI
           (primo == 1) )
        if ( n % d == 0 ) // SE
            primo = 0;    // PONI
        else // ALTRIMENTI
            d = d+1;
    if ( primo == 1)
        cout << "E' PRIMO"; // STAMPA
    else
        cout << "NON E' PRIMO!";

}
```

55

Ultimo aggiornamento: 10/03/2007

## Trasferimento dell'informazione



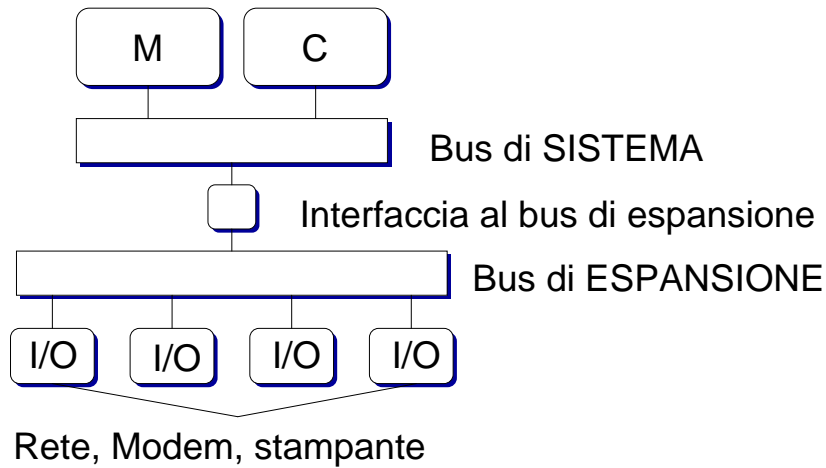
56

Ultimo aggiornamento: 10/03/2007

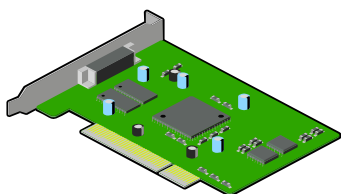
# Gerarchie di Bus

Se un bus viene collegato a molti dispositivi si avrà un calo delle prestazioni a causa dei seguenti motivi:

- ❑ Aumenta la lunghezza del bus e quindi aumenta il tempo di propagazione delle informazioni lungo il bus
- ❑ La domanda totale dei dati da trasferire tende ad avvicinarsi alla capacità del bus

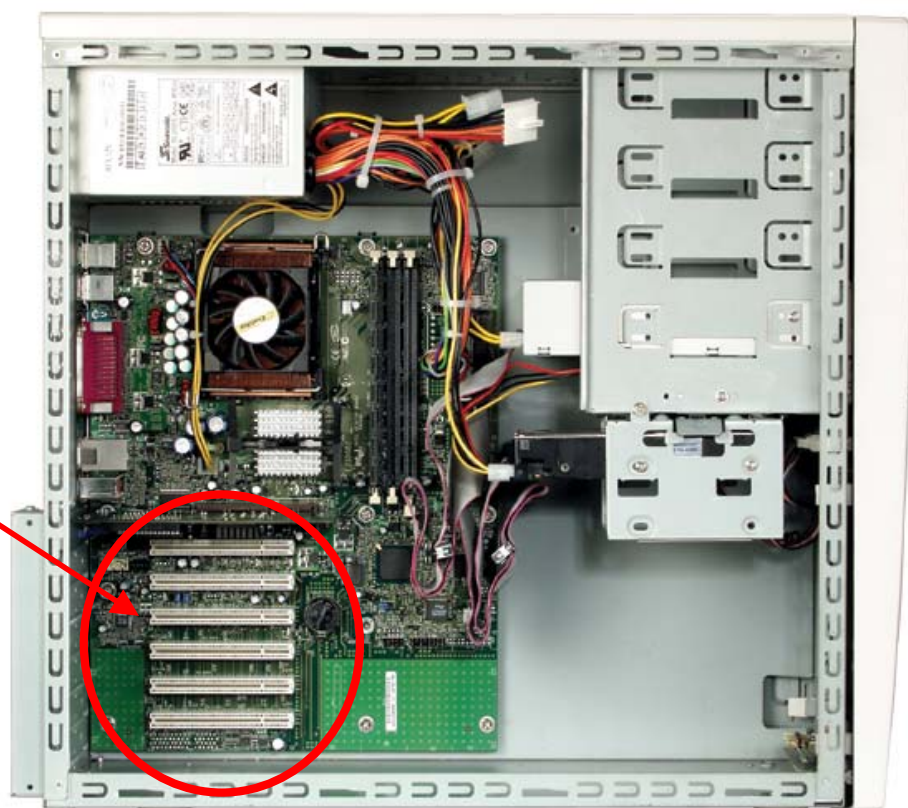


# Bus di espansione e slot

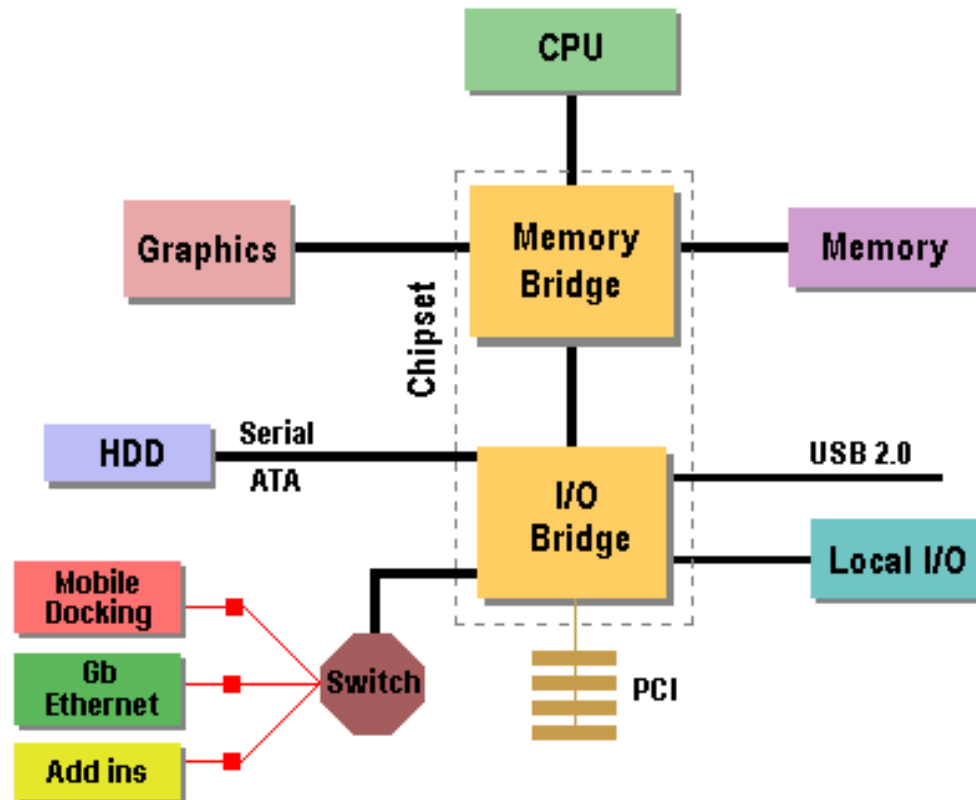


Interfaccia  
(scheda)

slot



# Bus di espansione e slot



Ultimo aggiornamento: 10/03/2007

59

## Volatilità della memoria

- ◆ La memoria RAM immagazzina le informazioni sotto forma di *carica di un condensatore*
- ◆ In assenza di tensione, le informazioni vanno perse  
*Volatilità della memoria*
- ◆ Sono necessari dispositivi che mantengano l'informazione anche in assenza di tensione (memorie *di massa, permanenti, non-volatili*)

Il fenomeno fisico con il quale l'informazione viene memorizzata sarà necessariamente diverso da quello della RAM

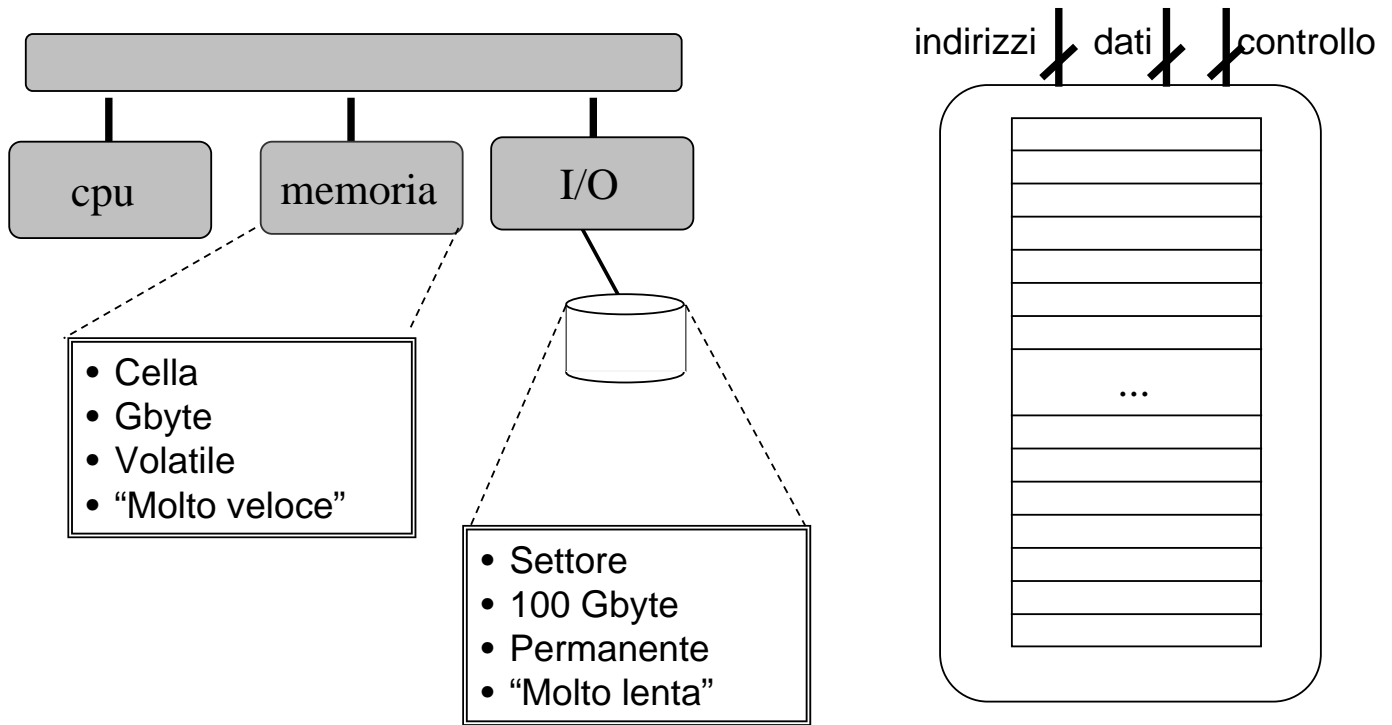
Floppy disk, hard disk:  
CD-ROM, DVD  
(Compact) Flash

dischi **magnetici**  
dischi **ottici**  
memorie a stato solido

Ultimo aggiornamento: 16/03/2007

60

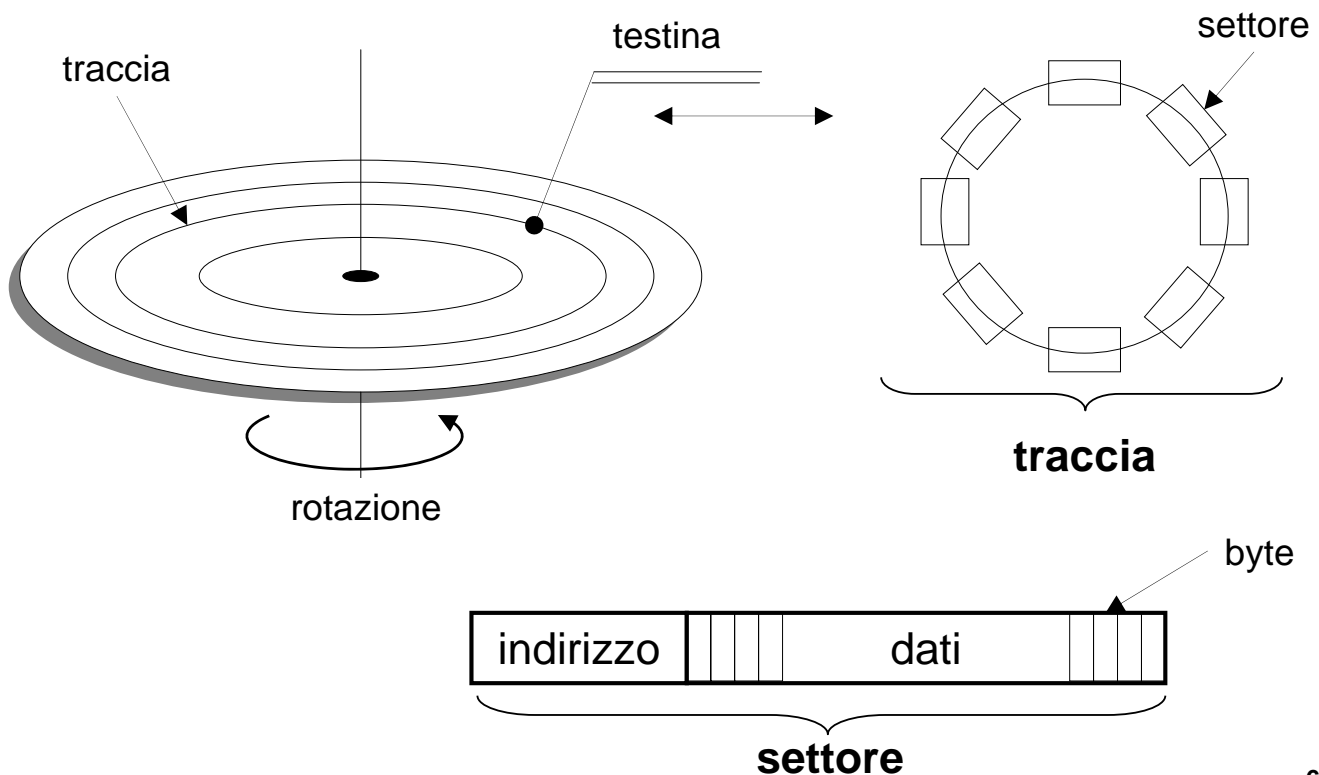
# Struttura logica del disco



61

Ultimo aggiornamento: 16/03/2007

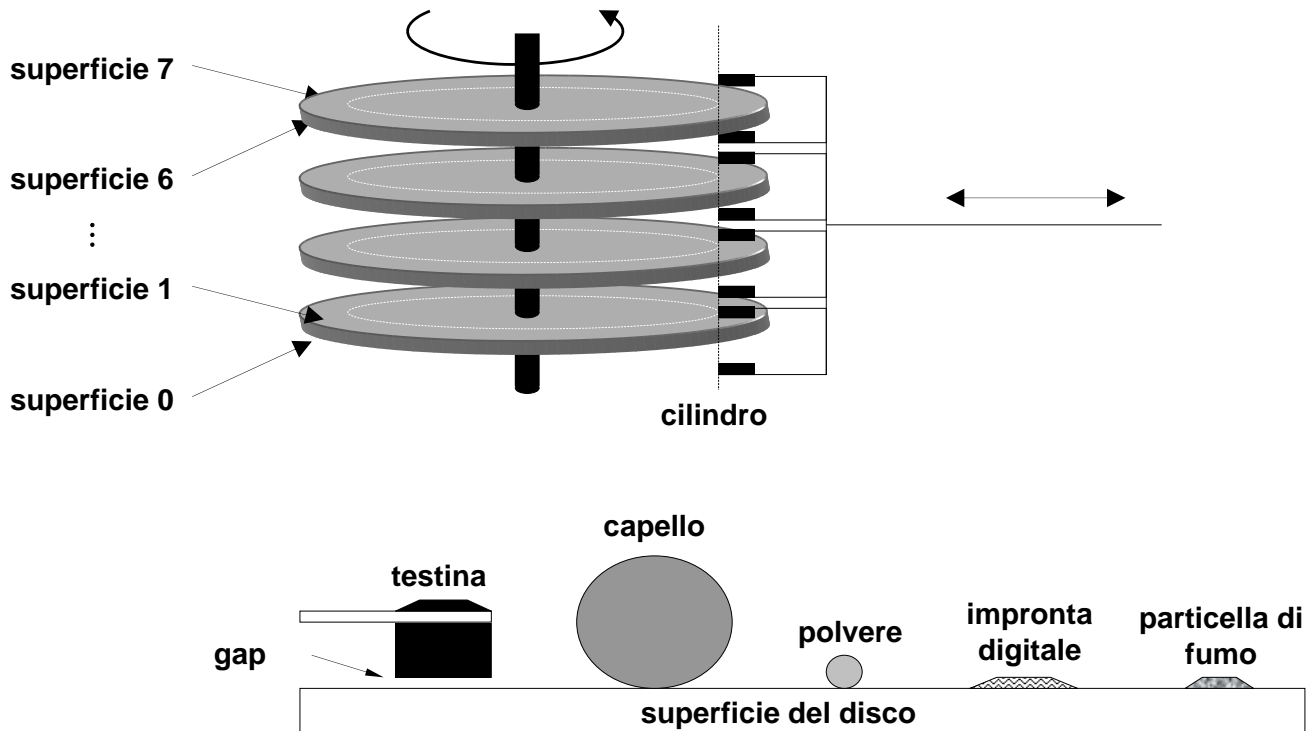
# Struttura fisica del disco (I)



62

Ultimo aggiornamento: 16/03/2007

# Struttura fisica del disco (II)



63

Ultimo aggiornamento: 16/03/2007

# Tempo di accesso al disco

## TRE COMPONENTI

- ◆ **tempo di ricerca:** tempo per posizionare la testina sulla traccia
- ◆ **latenza:** tempo necessario affinché il settore cominci a ruotare sotto la testina
- ◆ **tempo di trasferimento:** tempo necessario affinché l'intero settore passi sotto la testina

Il tempo di accesso è *variabile* e dipende dal settore scelto e dallo stato corrente del disco

64

Ultimo aggiornamento: 16/03/2007



# Tempo di accesso al disco: Esempio

---

- **Velocità di rotazione** = 7200 giri/min → 8.33 ms/giro
- **Tempo di spostamento** (del braccio sulla traccia adiacente) = 0.02 ms
- **Numero tracce** = 1000
- **Settori/traccia** = 50
- **Byte/settore** = 512

	Tempo di accesso		
	Migliore (ms)	Peggiora (ms)	Medio (ms)
<i>Tempo di ricerca</i>	0	$999 \times 0.02 = 19.98$	$500 \times 0.02 = 10$
<i>Latenza</i>	0	8.33	$8.33/2 = 4.17$
<i>Trasferimento</i>	$8.33/50 = 0.17$	$8.33/50 = 0.17$	$8.33/50 = 0.17$
<i>Totale</i>	0.17	28.48	14.34

65

Ultimo aggiornamento: 16/03/2007

# Memoria RAM vs. Hard Disk

---

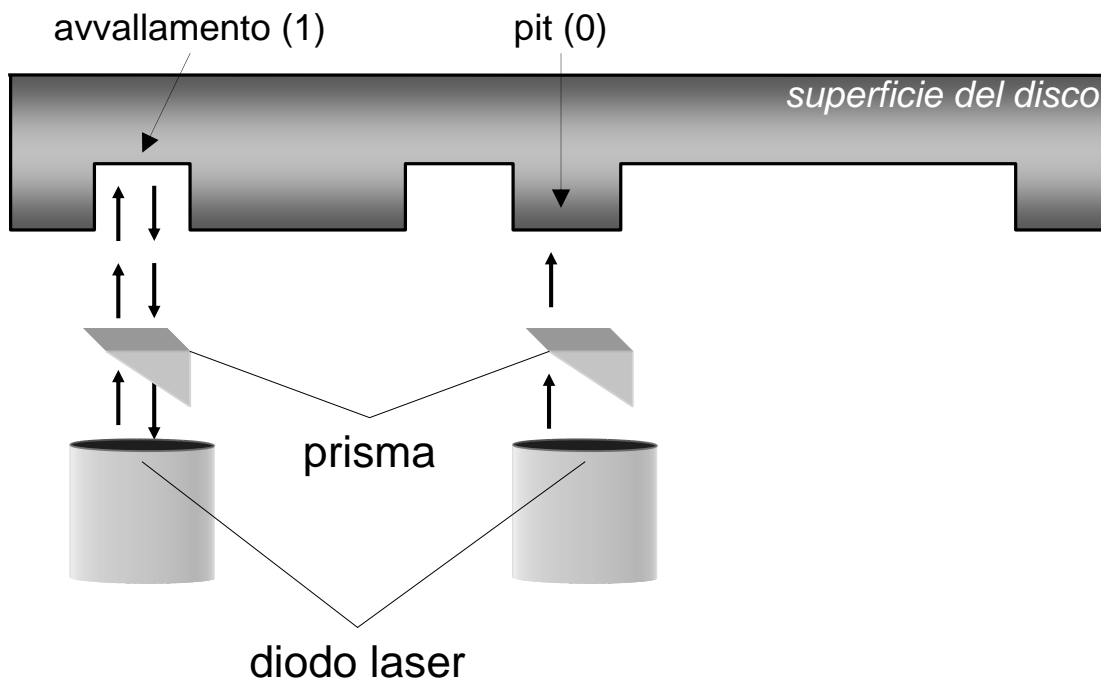
	MEMORIA RAM	DISCO
<i>Unità di accesso</i>	cella (8, 16, 32 bit)	settore (512 byte)
<i>In un PC...</i>	1-2 Gbyte	80-320 Gbyte
<i>Volatile/permanente</i>	Volatile	Permanente
<i>Tempo di accesso</i>	30-70 ns	0.2-20 ms
<i>Costo</i>	20-80 €/Gbyte	0.2-0.4 €/Gbyte

66

Ultimo aggiornamento: 31/03/2008

# CDROM

---



67

Ultimo aggiornamento: 16/03/2007

## CDROM e DVD

---

- ◆ L'informazione è memorizzata sfruttando il medesimo principio
- ◆ Rispetto al CDROM, un DVD ha tracce più vicine ed avvallamenti/pit di dimensione minore
  - Maggior capacità a parità di superficie
  - Necessaria maggior precisione nella lettura
- ◆ CDROM: 650/800 Mbyte (74-90m di musica)
- ◆ DVD: 4.7/9.6 Gbyte (1,5h di filmato hi-def)

68

Ultimo aggiornamento: 16/03/2007

# Memorie Compact Flash

---

- ◆ Strutturalmente simili a memorie ROM, con possibilità di *cancellare* blocchi di memoria (Flash-EEPROM)
- ◆ Connesse tramite bus plug-and-play USB (Universal Serial Bus)
- ◆ Capacità: 1GB - 4GB
- ◆ Tempi di accesso paragonabili a quelli di un HD
- ◆ Costo contenuto (<10 euro/GB)



69

Ultimo aggiornamento: 31/03/2008

# Dispositivi di I/O

---

- ◆ Memorie di massa (dischi, CD-ROM)  
*rappresentano le informazioni in forma binaria*
  - accesso diretto  
hard-disk, floppy-disk, CD-ROM
- ◆ Tastiera, mouse (in), schermi, stampanti (out)  
*Convertono le informazioni da una forma ad un'altra*
  - Utente umano -> computer (ingresso)*
  - Computer -> utente umano (uscita)*

70

Ultimo aggiornamento: 16/03/2007

# Tastiera (semplificata)

---

- ◆ Sotto i tasti c'è una *matrice* di contatti elettrici
- ◆ La pressione di un tasto viene ricondotta ad una coppia (riga,colonna)
- ◆ Un *programma* si occupa di tradurre l'informazione (riga, colonna) in un codice ASCII esteso
- ◆ Tale programma può essere configurato per interagire con tastiere *diverse* (italiana, USA, etc.)



71

Ultimo aggiornamento: 16/03/2007

# Mouse

---

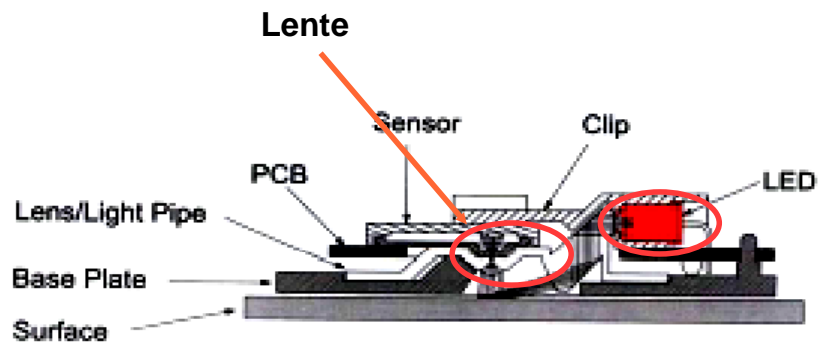
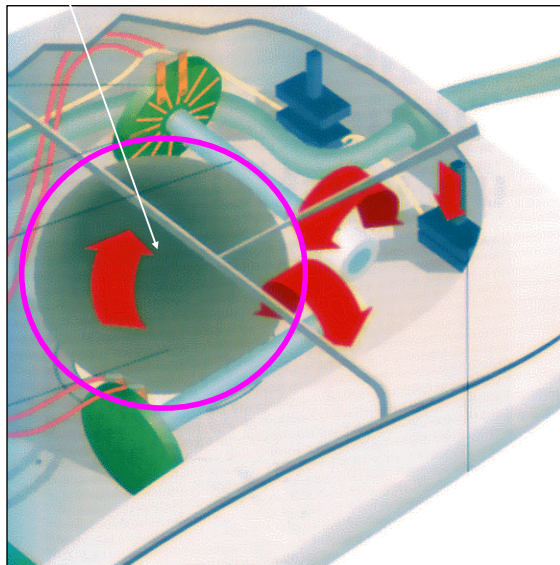
- ◆ Consente di associare il movimento di un oggetto su un piano al movimento di un puntatore sullo schermo  
Utile in ambienti grafici
- ◆ Il Mouse converte in impulsi elettrici  
Il **moto** di una sfera (mouse **meccanici**)  
La differenza di immagini successive (mouse **ottici**)

72

Ultimo aggiornamento: 16/03/2007

# Mouse meccanico ed ottico

Pallina di gomma

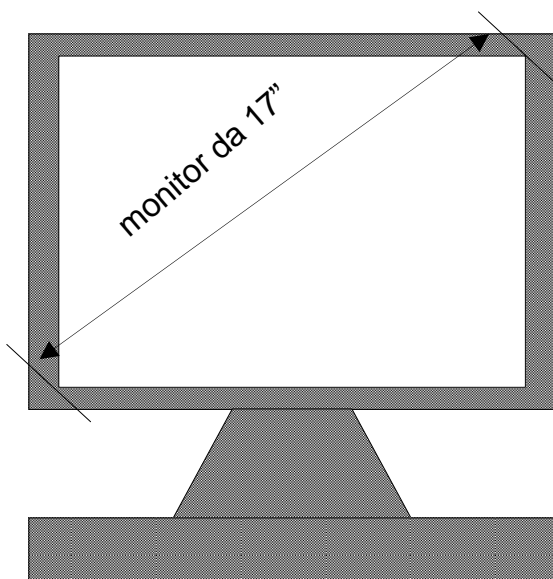


Ultimo aggiornamento: 16/03/2007

73

# Schermo

## PARAMETRI



- **Dimensioni** (lunghezza della diagonale)

- **Nitidezza** (dimensione dei punti, risoluzione, velocità di aggiornamento)

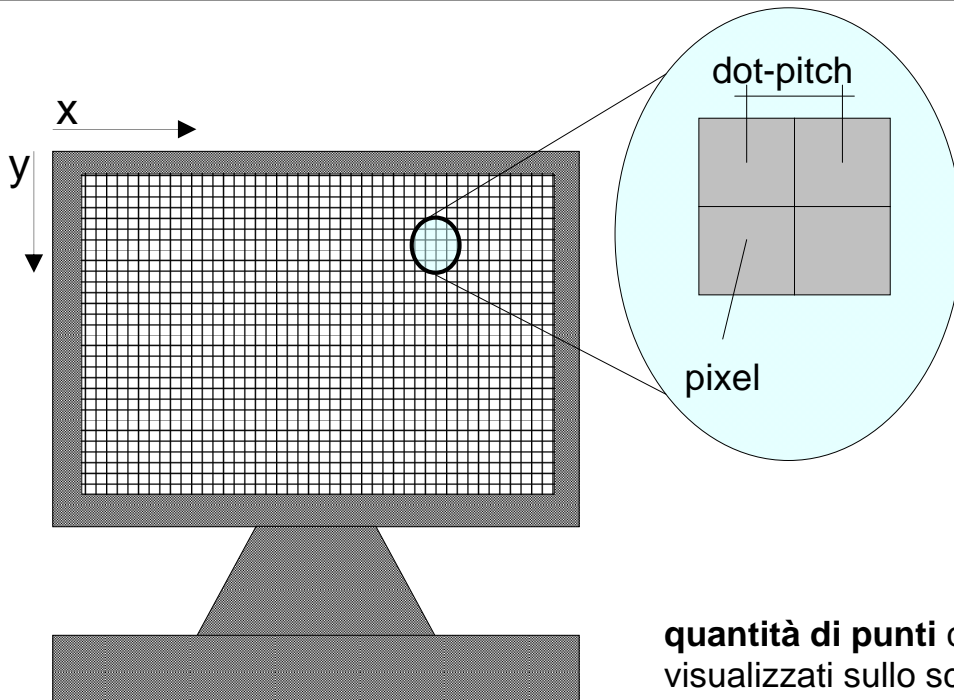
- **Tipo di schermo** (CRT, LCD)

- **Potenzialità di visualizzazione** (SVGA, XVGA)

Ultimo aggiornamento: 11/05/2007

74

# Pixel



**Dimensione dei punti (DOT-PITCH):** distanza tra i centri di due pixel adiacenti

Esempio: 0,28: 28 centesimi di mm

**alternativamente**

Numero di pixel per unità di misura lineare (72-100 dpi, *dots per inch*) (**risoluzione**)

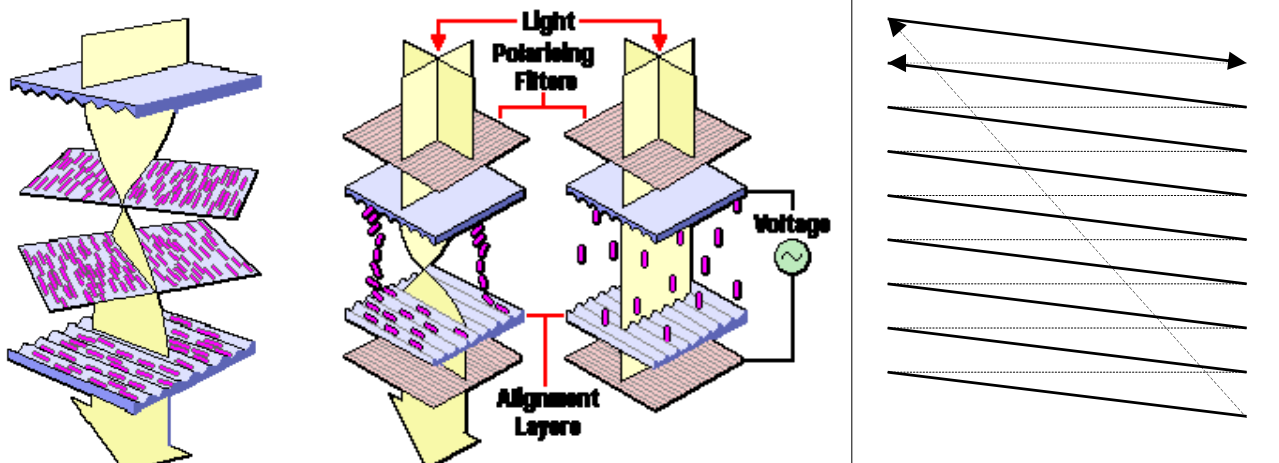
**quantità di punti** che possono essere visualizzati sullo schermo del monitor

Esempio: 640x480 (VGA), 800x600 (SVGA), 1024x768 pixel (XGA)

75

Ultimo aggiornamento: 11/05/2007

# Liquid Cristal Display



scansione non-interlacciata

**Frequenza di aggiornamento (verticale):** numero di volte al secondo che i pixel vengono aggiornati per conservarne la luminosità (es.: 50-100 volte il secondo)

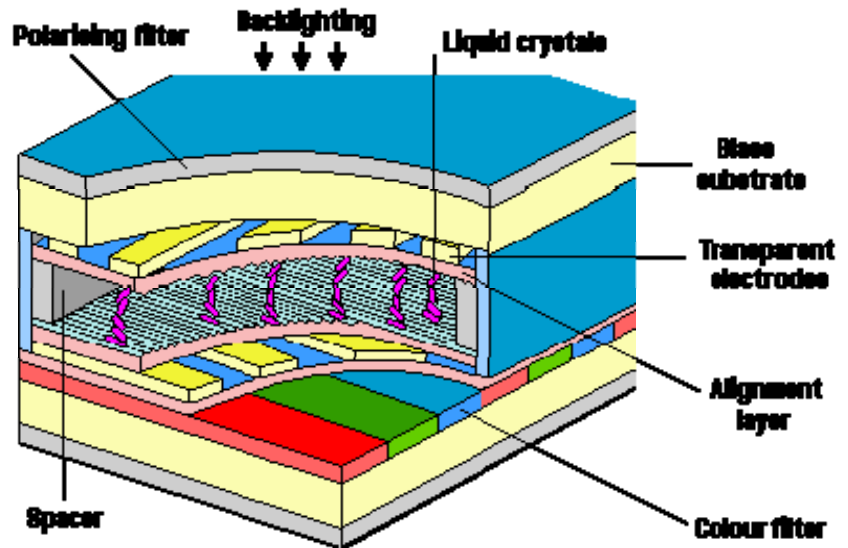
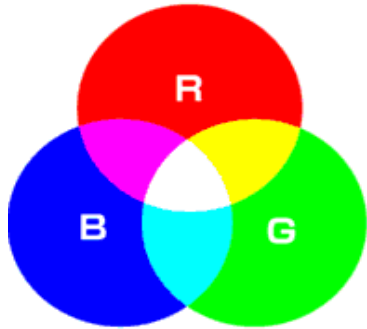
76

Ultimo aggiornamento: 05/05/2008

# LCD a colori

---

- ◆ Tre celle per ogni pixel, una per ognuno dei colori fondamentali (R,G,B)



77

Ultimo aggiornamento: 05/05/2008

# Schermo

---

**Frequenza di aggiornamento/rinfresco (verticale):** numero di volte al secondo che i pixel vengono aggiornati (50 – 110 Hz)

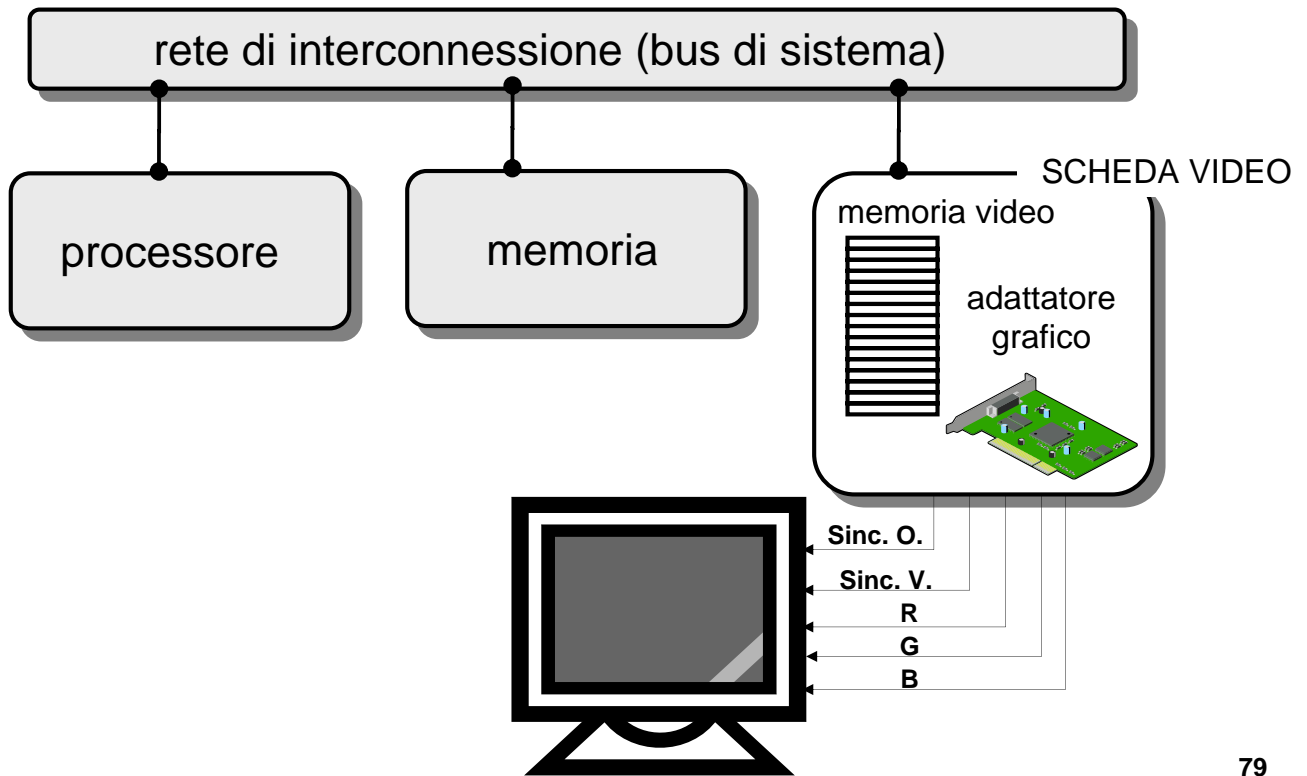
**Frequenza orizzontale:** numero di righe orizzontali che possono essere dipinte in un secondo (28-58 KHz)

**Banda video:** il numero di pixel contigui che possono essere colorati con colori diversi in un secondo (ovvero la frequenza con cui è possibile cambiare l'intensità del pennello). Valore tipico 200 MHz

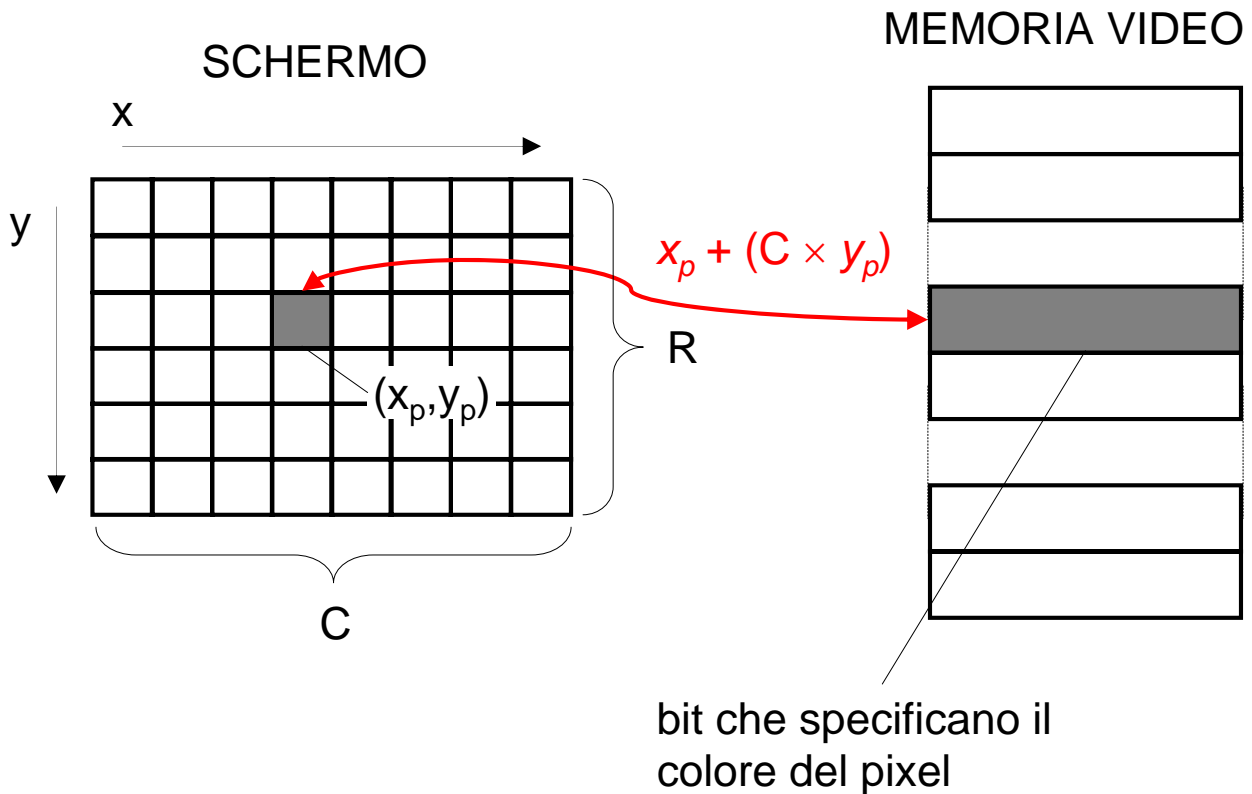
78

Ultimo aggiornamento: 11/05/2007

# Scheda Video (semplificata)



# Memoria Video (semplificata)





# Adattatore Grafico (semplificato)

---

- L'AG legge ad intervalli di tempo regolari le locazioni della MV
- Una volta che ha letto il contenuto di una locazione associata ad un certo pixel, l'AG compie le seguenti azioni:
  1. Considera i bit letti come la codifica del colore che dovrà caratterizzare il pixel
  2. Genera tre segnali corrispondenti ai tre colori fondamentali
  3. Trasmette tali segnali al monitor tramite i fili R, G e B

81

Ultimo aggiornamento: 11/05/2007

## Memoria Video: esempio

---

$C$  = Numero dei colori

$P$  = Numero di pixel

$n$  = bit di indirizzo per la Memoria Video

$d$  = dimensione di una parola di Memoria Video

$$n = \lceil \log_2 P \rceil$$

$$d = \lceil \log_2 C \rceil$$

$N$  = Numero locazioni della Memoria Video =  $2^n$  ( $N \geq P$ )

### ESEMPIO

Modalità Grafica:  $640 \times 480$ ; COLORI: 256  $\Rightarrow$

$P = 307200 \Rightarrow$

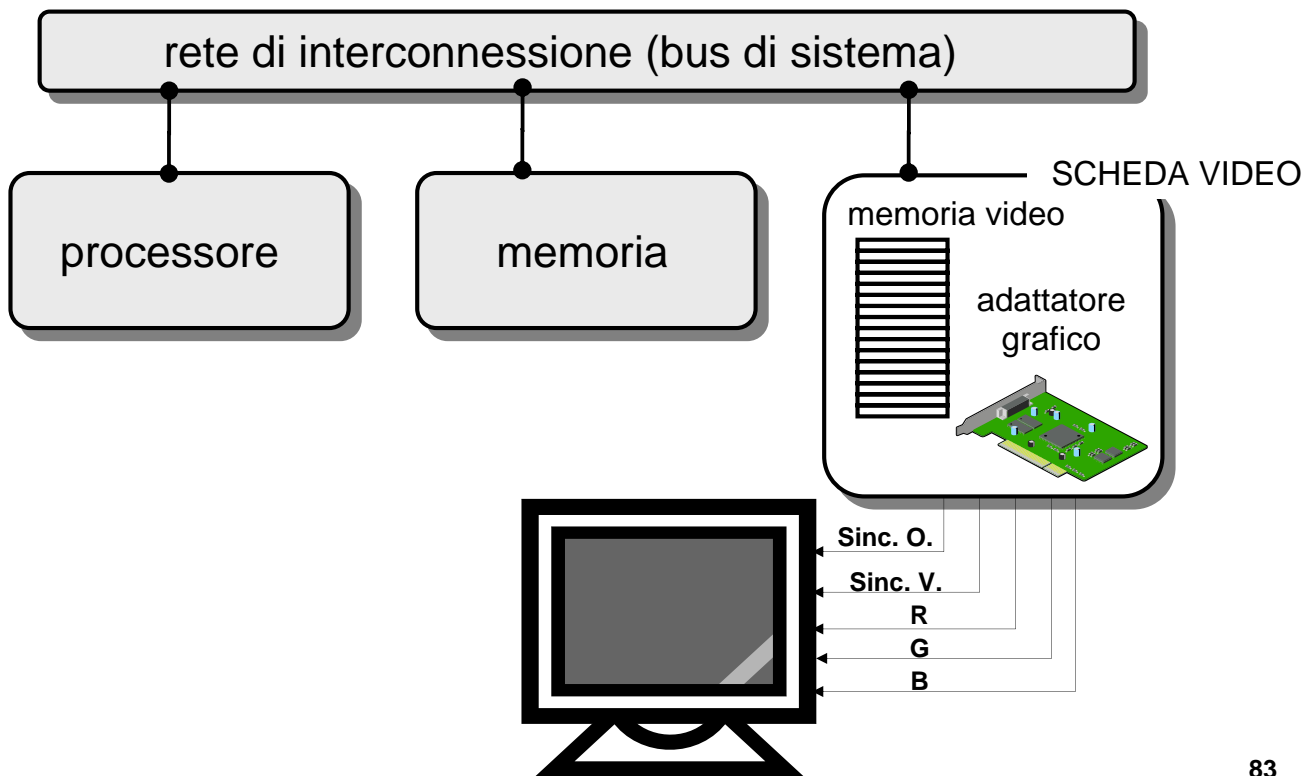
$n = 19$ ;  $d = 8 \Rightarrow$  Memoria Video di 512 Kbyte

82

Ultimo aggiornamento: 11/05/2007

# Pixel logici e pixel fisici

---



Ultimo aggiornamento: 11/05/2007

83

# Visualizzare bitmap

---

- ◆ La memoria video contiene, cella per cella, le informazioni di colore da visualizzare in ogni pixel dello schermo
- ◆ Se devo visualizzare un'immagine bitmap true color, basta che la trasferisca nella memoria video "così com'è"
- ◆ Il formato bitmap true color è il più semplice che si può immaginare per memorizzare un'immagine, ed è quello che richiede meno operazioni per la visualizzazione sullo schermo

Ultimo aggiornamento: 11/05/2007

84

## Visualizzare immagini compresse

---

- ◆ Un'immagine compressa (e.g., GIF) non può essere copiata direttamente nella memoria video
- ◆ Deve prima essere **decompressa**
- ◆ Migliore è l'algoritmo di compressione, più piccola è la dimensione, ed in genere maggiore è il numero di operazioni (e quindi il tempo) necessario per decomprimerla
- ◆ Chi fa la decompressione?
  - Normalmente, la CPU
  - Se la scheda grafica contiene un processore, è possibile che ad esso sia "subappaltata" parte del lavoro

85

Ultimo aggiornamento: 11/05/2007

## Visualizzare un'immagine vettoriale

---

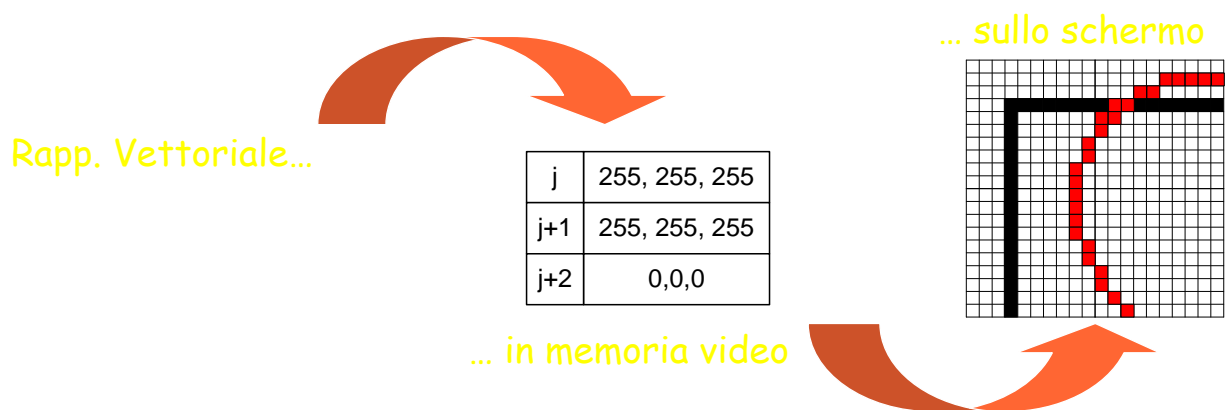
- ◆ Per visualizzare sullo schermo (periferica raster) un'immagine descritta in modo vettoriale, è necessario creare in memoria video il raster che la contiene
  - Processo di **rasterizzazione (rendering)**
  - Per ogni pixel logico (cella di memoria video), devo decidere un colore, calcolato sulla base della descrizione vettoriale di **tutti** gli oggetti contenuti in un'immagine

86

Ultimo aggiornamento: 11/05/2007

# Rasterizzazione in memoria video

---



- ◆ Se ci sono molti oggetti, è necessario fare molti conti per stabilire il colore di ogni singolo pixel

87

Ultimo aggiornamento: 11/05/2007

# Rasterizzazione in memoria video

---

- ◆ Chi esegue la rasterizzazione in memoria video?

La CPU

Le schede video moderne hanno un *processore grafico*. Tale processore è particolarmente adatto a svolgere compiti del genere in modo efficiente, alleggerendo in tal modo il carico della CPU (che nello stesso tempo può fare altre cose)

Un processore grafico può rasterizzare una notevole quantità di poligoni al secondo

88

Ultimo aggiornamento: 11/05/2007

# Esempio

---

## ◆ Scheda Video NVIDIA GeForce4 Ti 4800

Processore grafico NfiniteFX II Engine, in grado di rasterizzare 45 milioni di triangoli al secondo

Ciò significa che si possono rasterizzare in tempo reale scene animate (24 fotogrammi al secondo) composte da ~2 milioni di triangoli

Immagine vettoriale  
inserita sopra uno  
sfondo raster



Ultimo aggiornamento: 11/05/2007

89

# Stampanti

---

Differiscono per:

## ◆ Tecnologia di stampa

Getto di inchiostro

Laser

## ◆ Capacità cromatica

Monocromatica

Colore

## ◆ Velocità di stampa

Numero di pagine al minuto

## ◆ Risoluzione

>600 dpi

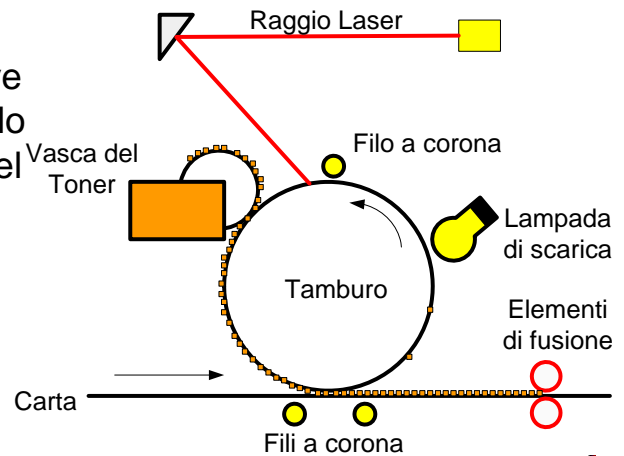


Ultimo aggiornamento: 11/05/2007

90

# Stampanti laser

- ◆ Un raggio laser, tramite specchi che lo deflettono, scandisce per righe successive un tamburo e ne carica elettricamente solo i punti che dovranno corrispondere ai pixel da stampare
- ◆ Contemporaneamente un foglio di carta passa vicino ad un filo percorso da corrente e si carica elettrostaticamente
- ◆ Il tamburo, ruotando, *pesca* toner da una vaschetta
- ◆ Il toner, a carica opposta al tamburo, si fissa ad esso solo nei punti caricati dal raggio laser
- ◆ Quando la carta passa a contatto con il tamburo, dato che la carta è molto più carica, il toner viene attirato dalla carta

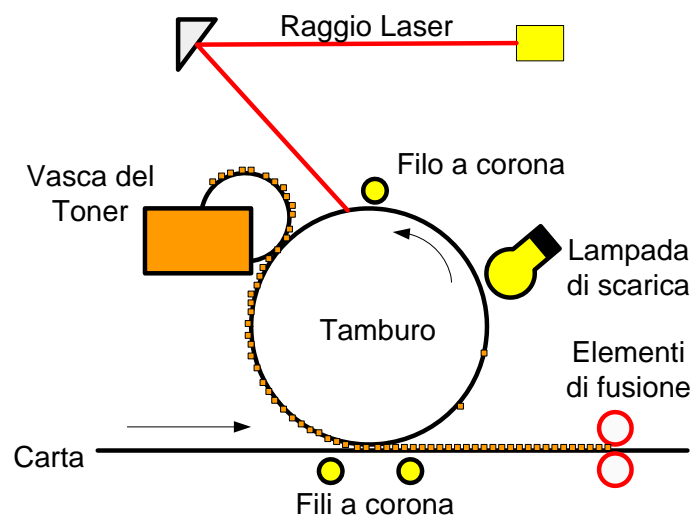


91

Ultimo aggiornamento: 11/05/2007

# Stampanti laser

- ◆ Il tamburo, continuando a ruotare, passa vicino alla *lampada di scarica* che lo scarica completamente
- ◆ La carta passa invece dal sistema di fusione che, con una combinazione di calore e pressione, fissa in permanenza, fondendolo, il toner alla carta



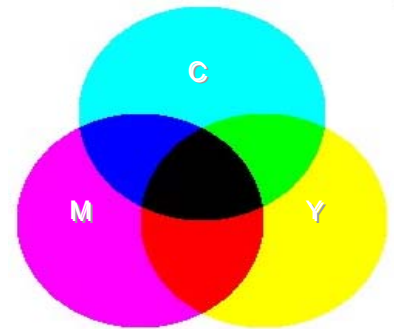
92

Ultimo aggiornamento: 11/05/2007

# Stampanti a colori

---

- ◆ Nelle stampanti a colori, si usa un modello di colore basato su quattro colori primari  
Cyan, Magenta, Yellow, Black (CMYK)
- ◆ In realtà, basterebbero i primi tre  
Nero usato per convenienza e precisione
- ◆ Il processo appena descritto viene ripetuto 4 volte, ogni volta con un inchiostro di colore differente



Ultimo aggiornamento: 11/05/2007

## Il processo di stampa

---

- ◆ Come si passa da una *pagina* (ad esempio) di testo *all'insieme di operazioni* che devono essere effettuate sulla stampante per ottenere la pagina stampata?  
Il software che elabora il documento (p.e. Word) interagisce con il **driver** della stampante  
Tale interazione avviene secondo criteri *standard*, che **non** dipendono dal modello di stampante  
Il driver invia alla stampante i comandi nella maniera ad essa comprensibile  
Tale interazione **dipende** dal modello di stampante. Infatti stampanti *diverse* hanno driver *diversi*

Ultimo aggiornamento: 11/05/2007

# Stampanti PostScript

---

- ◆ Hanno dei processori che sanno eseguire comandi scritti in un particolare linguaggio (PostScript)

```
2028 y(and)g(dev)o(ex)e(up)q(dates)i(using)f Fq(\033)i
Fy(will)d(also)i(not)g(carry)f(o)o(v)o(er)f(directly)l(.)949
2783 y(17)p eop
%%Page: 16 48
16 47 bop 164 98 a Fz(Up)r(dating)237 204 y Fy(The)23
b(\026)-27 b Fq(x)p Fy(,)20 b Fq(B)s Fy(,)g Fq(\015)502
```

- ◆ Per mandare una pagina di testo, invece che convertirla in una matrice di punti posso darle una descrizione molto più sintetica in PS e lasciare che la stampante la interpreti