

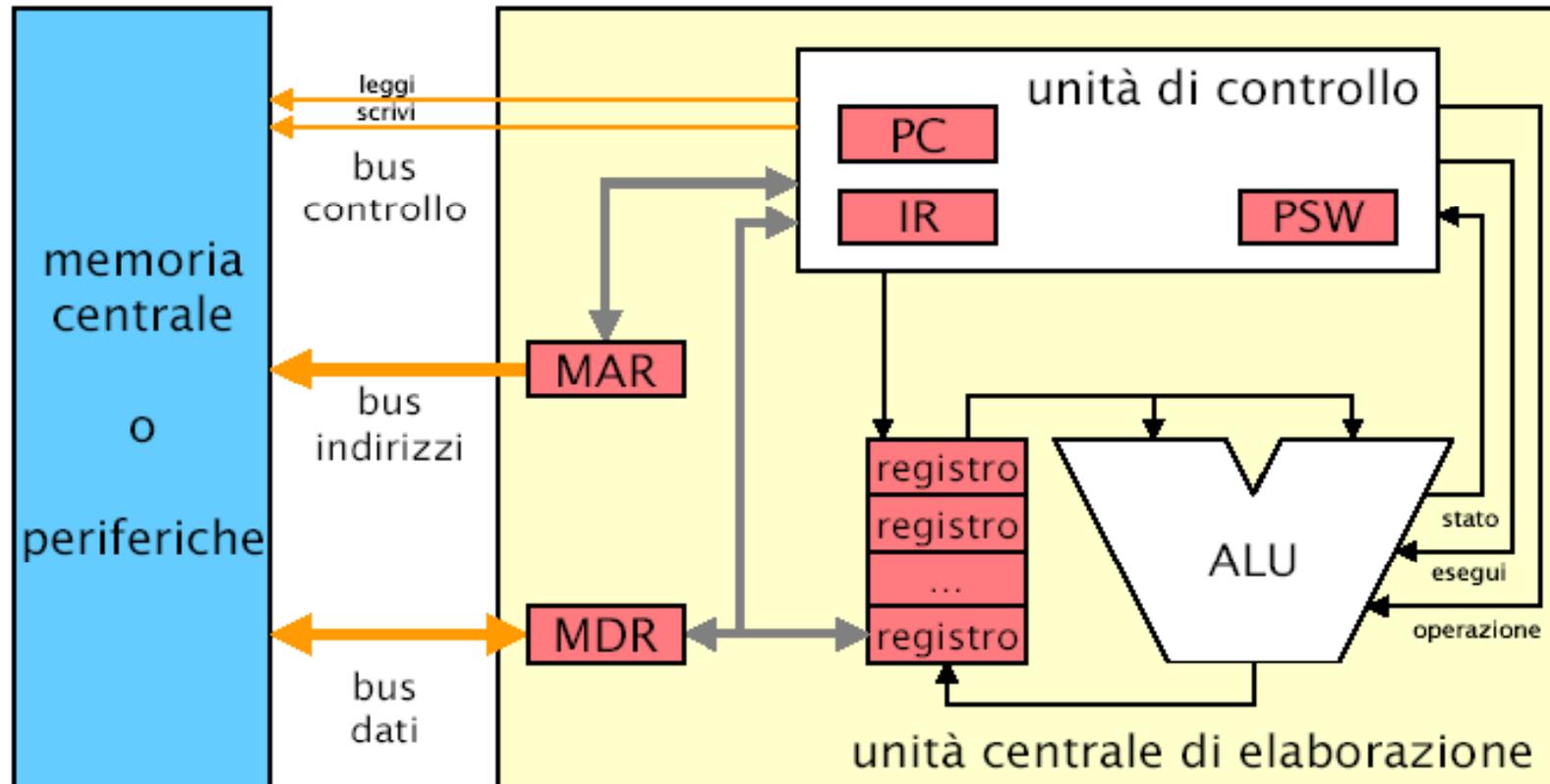
ELABORAZIONE

- ◆ Le istruzioni del **linguaggio macchina** corrispondono ad operazioni elementari di elaborazione
 - operazioni aritmetiche
 - operazioni relazionali (confronto tra dati)
 - operazioni booleane
 - altre operazioni
- ◆ Un calcolatore sa svolgere poche tipologie di operazioni elementari ma in modo molto efficiente
 - un calcolatore può eseguire decine o centinaia di milioni di istruzioni del linguaggio macchina al secondo
- ◆ L'elaborazione è svolta dall'**unità aritmetico-logica**, che è un componente dell'unità centrale di elaborazione

CONTROLLO

- ◆ Il coordinamento tra le varie parti del calcolatore è svolto dall'**unità di controllo**
 - è un componente dell'unità centrale di elaborazione
 - ogni componente del calcolatore esegue solo le azioni che gli vengono richieste dall'unità di controllo
- ◆ il controllo consiste nel coordinamento dell'esecuzione temporale delle operazioni
 - sia internamente all'unità di elaborazione sia negli altri elementi funzionali
 - il controllo avviene in modo sincrono rispetto alla scansione temporale impostata dall'orologio di sistema (**clock**)

CPU (CENTRAL PROCESSING UNIT)



REGISTRI

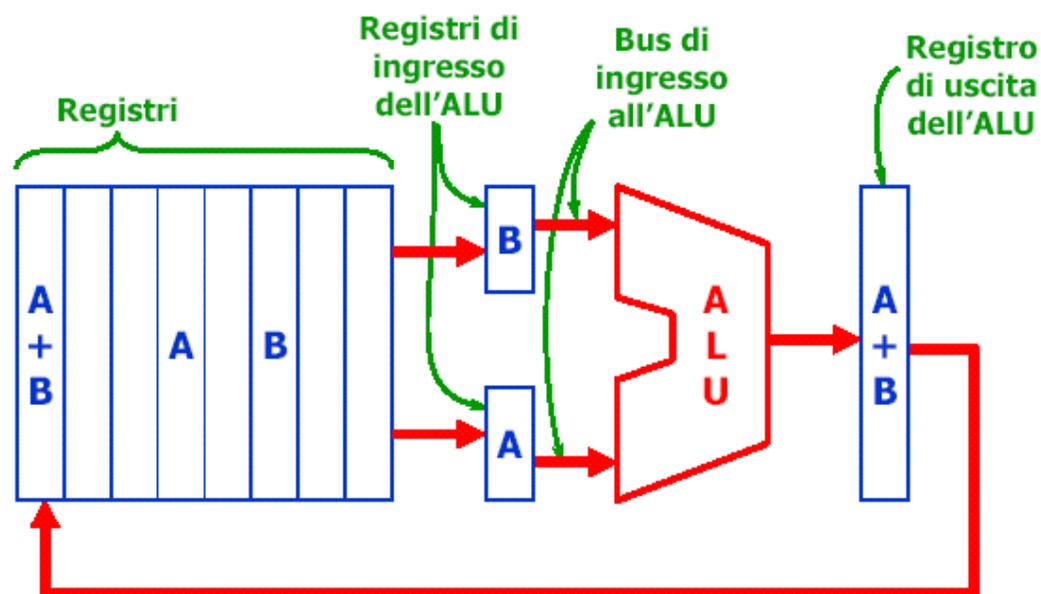
- ◆ Un microprocessore ha un numero limitato di celle di memoria (registri) con scopi specifici:
 - **PC: contatore delle istruzioni (program counter)**
 - contiene l'indirizzo della prossima istruzione da eseguire
 - **IR: registro delle istruzioni (instruction register)**
 - contiene l'istruzione che deve essere eseguita
 - **PSW: parola di stato del processore**
 - contiene informazioni, opportunamente codificate, sull'esito dell'ultima istruzione che è stata eseguita

REGISTRI

- **MAR: registro indirizzi della memoria**
 - indirizzo della cella di memoria che deve essere acceduta o memorizzata
- **MDR: registro dati della memoria**
 - dato che è stato acceduto o che deve essere memorizzato
- **registri generali**
 - per memorizzare gli operandi ed il risultato di una operazione

UNITA' ARITMETICO-LOGICA

- ◆ L'Unità Aritmetico-Logica (**ALU**) è costituita da un insieme di circuiti in grado di svolgere le operazioni di tipo aritmetico e logico
- ◆ La ALU legge i valori presenti in alcuni registri, esegue le operazioni e memorizza il risultato in un altro registro



CICLO DI ESECUZIONE DELLE ISTRUZIONI

- ◆ L'unità di controllo esegue un'istruzione mediante le tre seguenti operazioni di base:
 - **Fetch** (lettura)
 - **Decode** (decodifica)
 - **Execute** (esecuzione)
- ◆ Fetch-Decode-Execute
 1. Prendi l'istruzione corrente, vale a dire quella individuata dal registro PC, e mettila nel registro IR (**fetch**)
 2. Incrementa il PC in modo che contenga l'indirizzo dell'istruzione successiva
 3. Determina il tipo di istruzione da eseguire (**decode**)
 4. Se l'istruzione necessita di un dato in memoria determina dove si trova e caricalo in un registro della CPU
 5. Esegui l'istruzione (**execute**)
 6. Torna al punto 1

ESEMPIO FETCH-DECODE-EXECUTE

1000

Load	3568	R1
Add	R1	R2
Store	R1	3568
Jump	1000	

1001

1002

1003

3568

44

Registri CPU

R1

R2

PC

IR

MAR

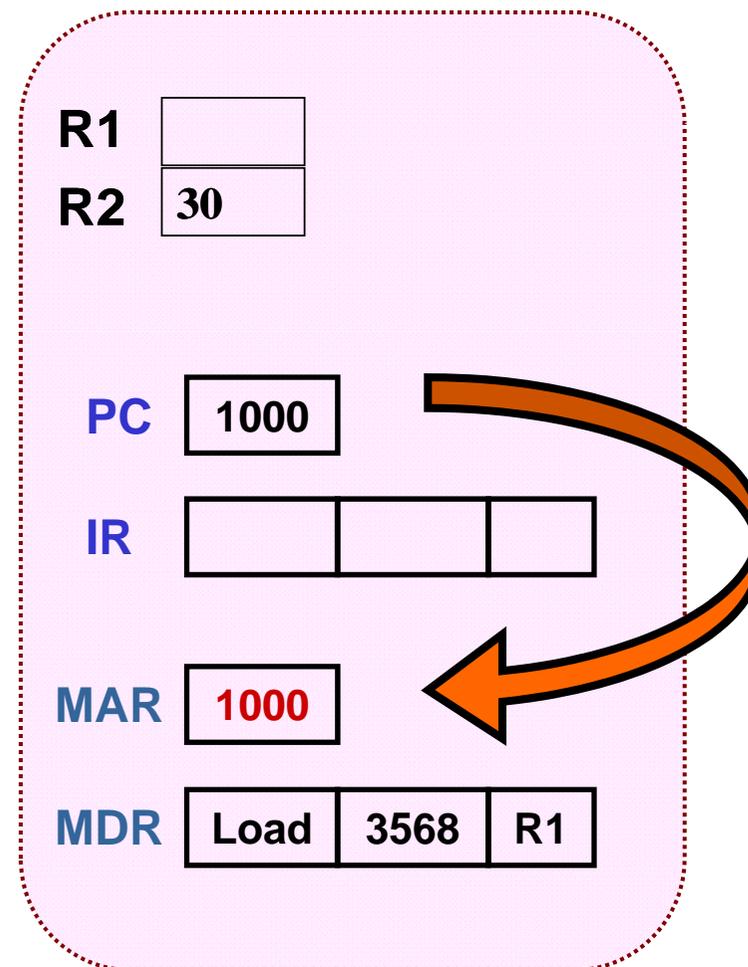
MDR

ESECUZIONE ISTRUZIONE 1000: FETCH (1/2)

1000	Load	3568	R1
1001	Add	R1	R2
1002	Store	R1	3568
1003	Jump	1000	

3568	44
------	-----------

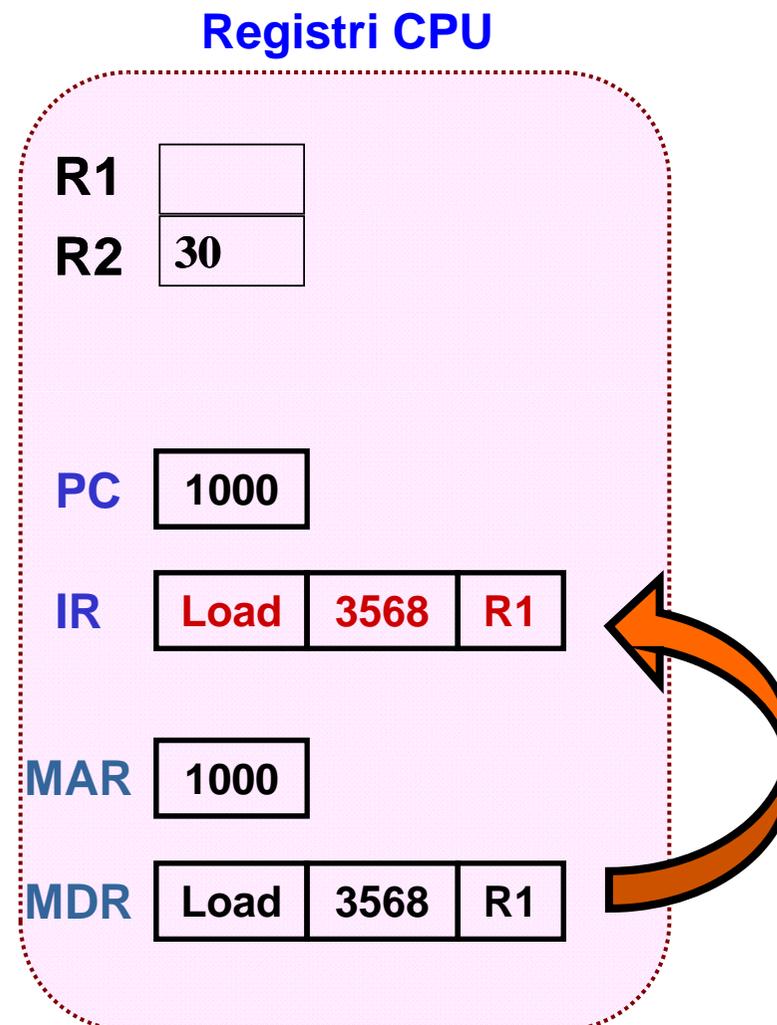
Registri CPU



ESECUZIONE ISTRUZIONE 1000: FETCH (2/2)

1000	Load	3568	R1
1001	Add	R1	R2
1002	Store	R1	3568
1003	Jump	1000	

3568	44
------	----



ESECUZIONE ISTRUZIONE 1000: DECODE & EXECUTE

1000	Load	3568	R1
1001	Add	R1	R2
1002	Store	R1	3568
1003	Jump	1000	

3568	44
------	-----------

Registri CPU

R1 **44**
R2 **30**

PC **1001**

IR **Load 3568 R1**

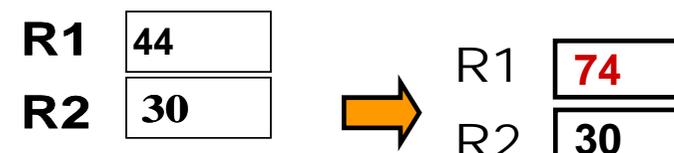
MAR **1000**

MDR **Load 3568 R1**

ESECUZIONE ISTRUZIONE 1001

1000	Load	3568	R1
1001	Add	R1	R2
1002	Store	R1	3568
1003	Jump	1000	

3568	44
------	----

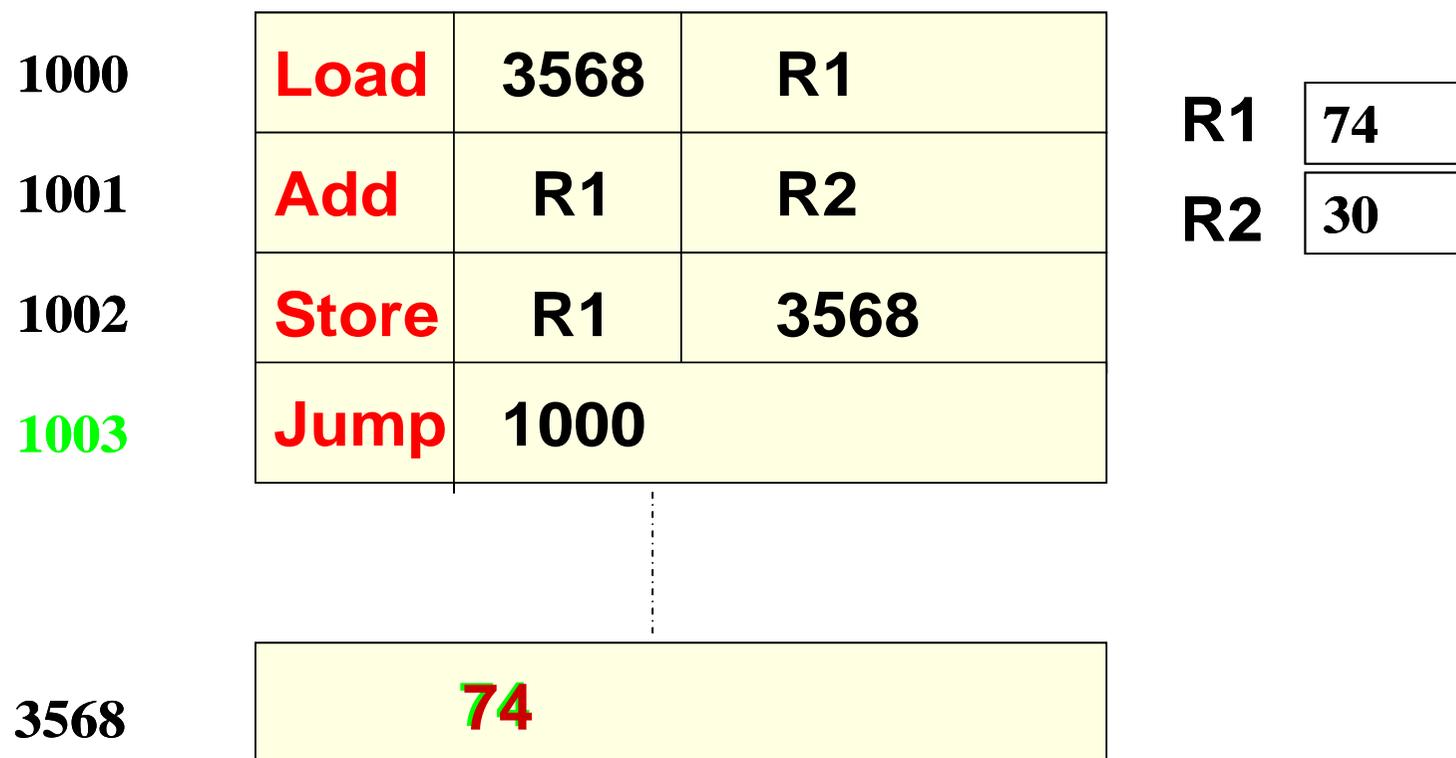


NB.:

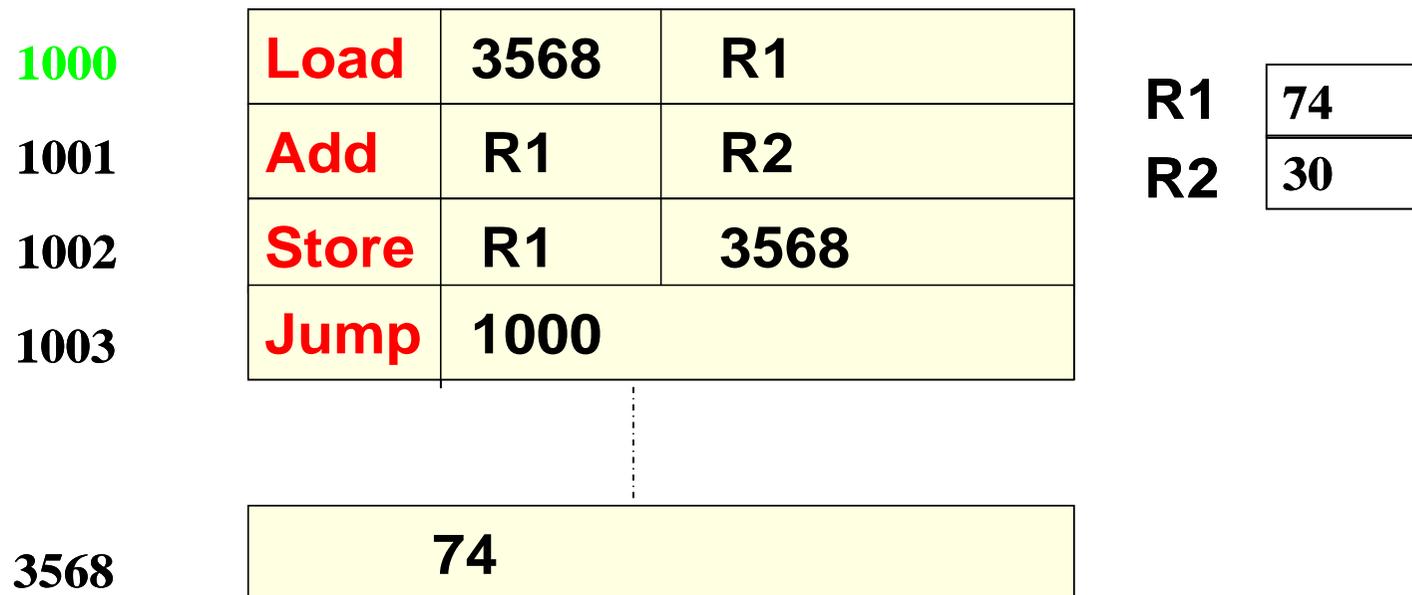
E' stata attivata la ALU

$$R1 = 44 + 30 = 74$$

ESECUZIONE ISTRUZIONE 1002

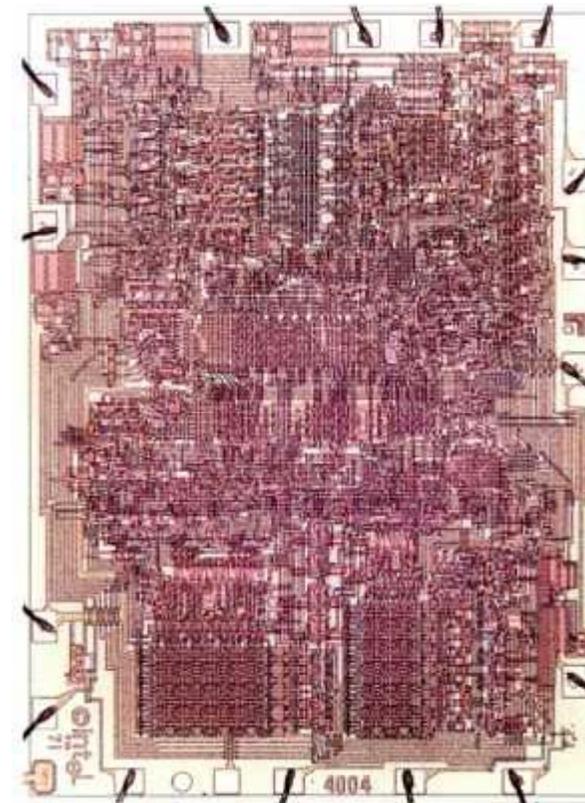


ESECUZIONE ISTRUZIONE 1003



MICROPROCESSORE

- ◆ Nel 1971, tre ingegneri della Intel, Federico Faggin, Ted Hoff e Stan Mazer costruirono il primo microprocessore.
- ◆ Riuscirono ad inserire 2.250 transistor su una piastrina di silicio di 4×3 mm, che formavano il cuore di un intero calcolatore in grado di elaborare in parallelo 4 bit.
- ◆ Da allora sono seguiti numerosi altri modelli, sempre più sofisticati e potenti che, grazie al loro basso costo, hanno determinato l'attuale enorme diffusione dei calcolatori.



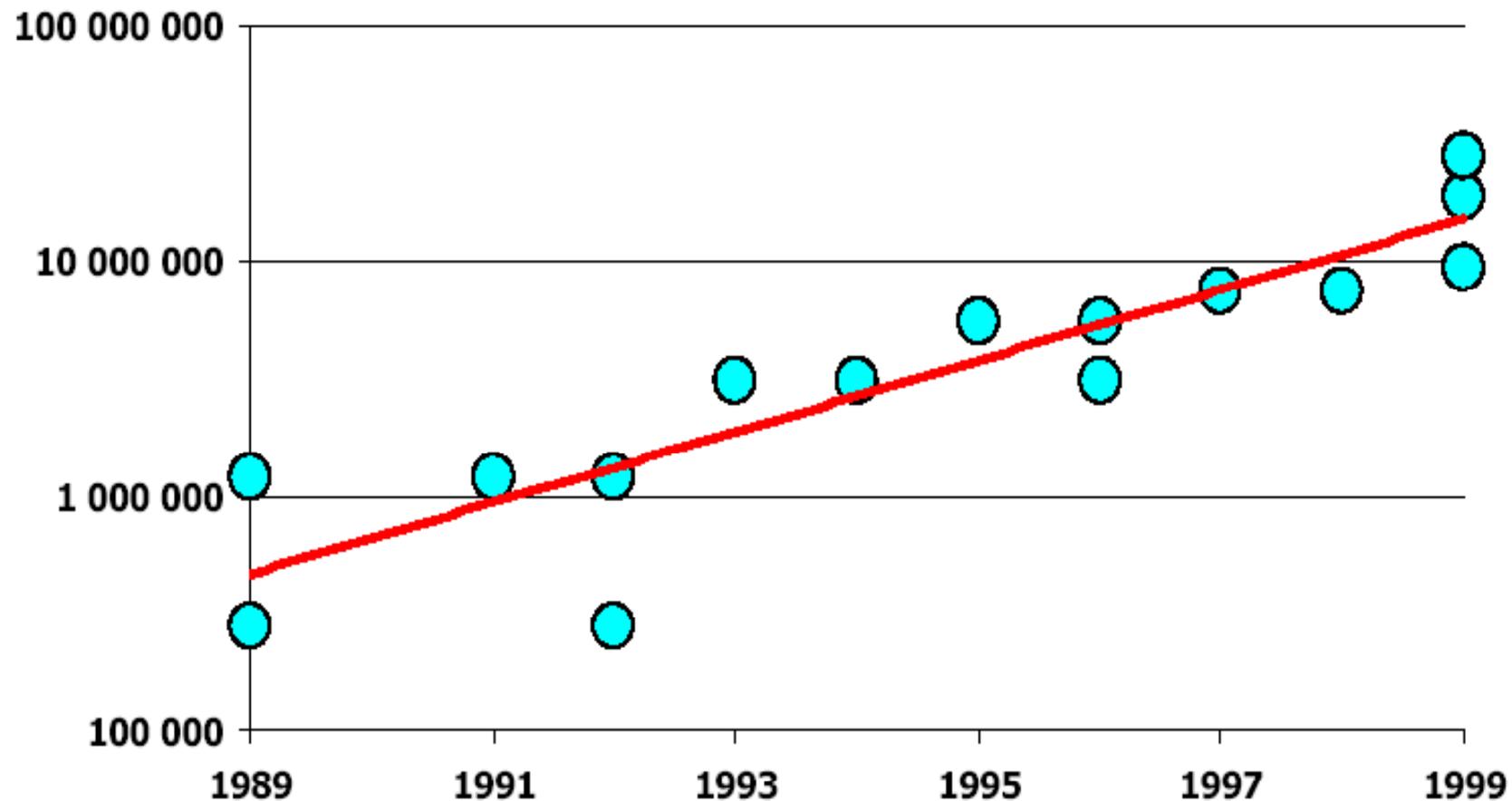
**Ingrandimento dell'Intel
4004 (1971)**

EVOLUZIONE DEI MICROPROCESSORI

CPU	Anno	Frequenza (MHz)	Dimensione registri / bus dati	Numero di transistor
8086	1978	4.77 — 12	8 / 16	29 000
80286	1982	8 — 16	16 / 16	134 000
80386	1986	16 — 33	32 / 32	275 000
80386 SX	1988	16 — 33	32 / 16	275 000
80486	1989	33 — 50	32 / 32	1 200 000
Pentium	1993	60 — 200	32 / 64	3 100 000
Pentium II	1997	233 — 400	32 / 64	7 500 000
Pentium III	1999	450 — 1133	32 / 64	24 000 000
Pentium 4	2000	1600 — 2000	32 / 64	42 000 000

NUMERO DI TRANSISTOR (CPU INTEL)

Il numero di transistor per cm² raddoppia ogni 18 mesi!



MEMORIA

◆ Caratterizzazione

- Capacità
- Costo per bit
- Velocità di accesso
 - Velocità di trasferimento (bit/s o Byte/s, dati trasferiti/secondo)
- Volatilità

◆ Categorie

- Memoria centrale
- Memoria di massa

MEMORIA

- ◆ Tipo di accesso
 - Sequenziale (nastro)
 - Casuale (memoria centrale)
 - Misto (disco)

- ◆ RAM (*Random Access Memory*)
 - Memoria centrale, Volatile, Lettura/scrittura, Accesso casuale

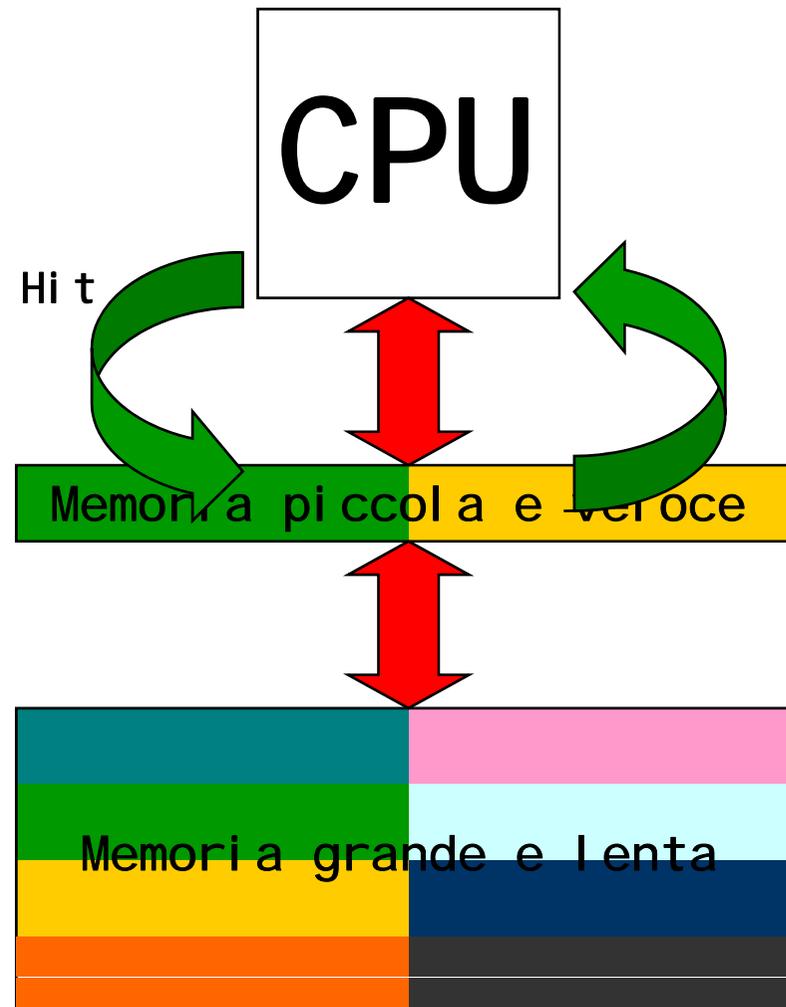
- ◆ ROM (*Read Only Memory*)
EPROM (*Erasable Programmable ROM*)
 - Permanente, Sola lettura (ROM) / Cancellabili e riscrivibili (EPROM)
 - BIOS

PRINCIPIO DI LOCALITA'

- ◆ **Località spaziale:**
quando si accede all'indirizzo **A**, è molto probabile che gli accessi successivi richiedano **celle vicine ad A**.
 - le istruzioni del codice vengono in genere lette da locazioni consecutive della memoria
 - Gli accessi a strutture dati sono “vicini”
- ◆ **Località temporale:**
quando si accede all'indirizzo **A**, è molto probabile che negli accessi successivi si richieda **di nuovo la cella A**.
 - cicli di istruzioni accedono ripetutamente alle stesse locazioni di memoria
 - istruzioni vicine tendono ad utilizzare le stesse variabili

GERARCHIA DI MEMORIA

- ◆ Obiettivi
 - Dimensione elevata
 - Velocità elevata
 - Costo limitato
- ◆ I dati prelevati dalla MGL vengono conservati il **più a lungo possibile**
- ◆ Quando si copia un dato dalla MGL si copiano anche i **dati vicini**
- ◆ h , **hit ratio** (frequenza di successo) pari al 99%.
 $1 - h$, **miss ratio**



GERARCHIA DI MEMORIA

- ◆ Obiettivi
 - Dimensione elevata
 - Velocità elevata
 - Costo limitato
- ◆ I dati prelevati dalla MGL vengono conservati il **più a lungo possibile**
- ◆ Quando si copia un dato dalla MGL si copiano anche i **dati vicini**
- ◆ $h = 0.99, T_1 = 1, T_2 = 10$
 $T_{medio} = T_1 h + T_2 (1 - h) = 1.09!$

