

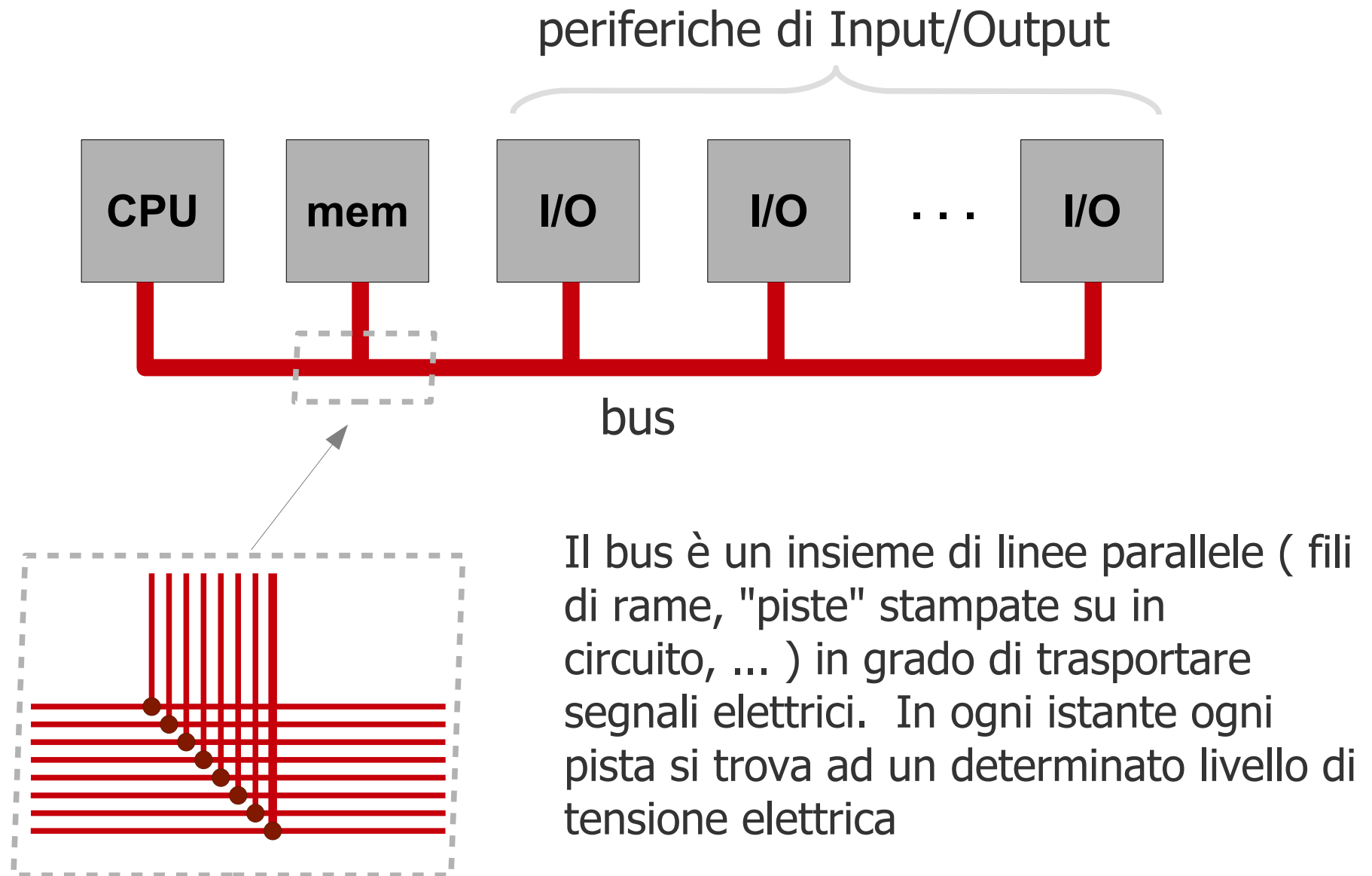
Architettura dei calcolatori

Architettura dei calcolatori

Un **computer** (calcolatore o elaboratore) è un sistema composto dai seguenti quattro componenti:

- **CPU**
- **Memoria principale (RAM)**
- **Periferiche di Input/Output**
 - Periferiche di input: tastiera, mouse, scanner, fotocamera
 - Periferiche di output: monitor, stampante
 - Periferiche di input/output: memoria di massa (hard disk, disco SSD), monitor touch-screen, scheda di rete, unità CD/DVD
- **Bus** = canale di comunicazione che permette lo scambio di informazioni fra componenti (CPU, memoria, periferiche)

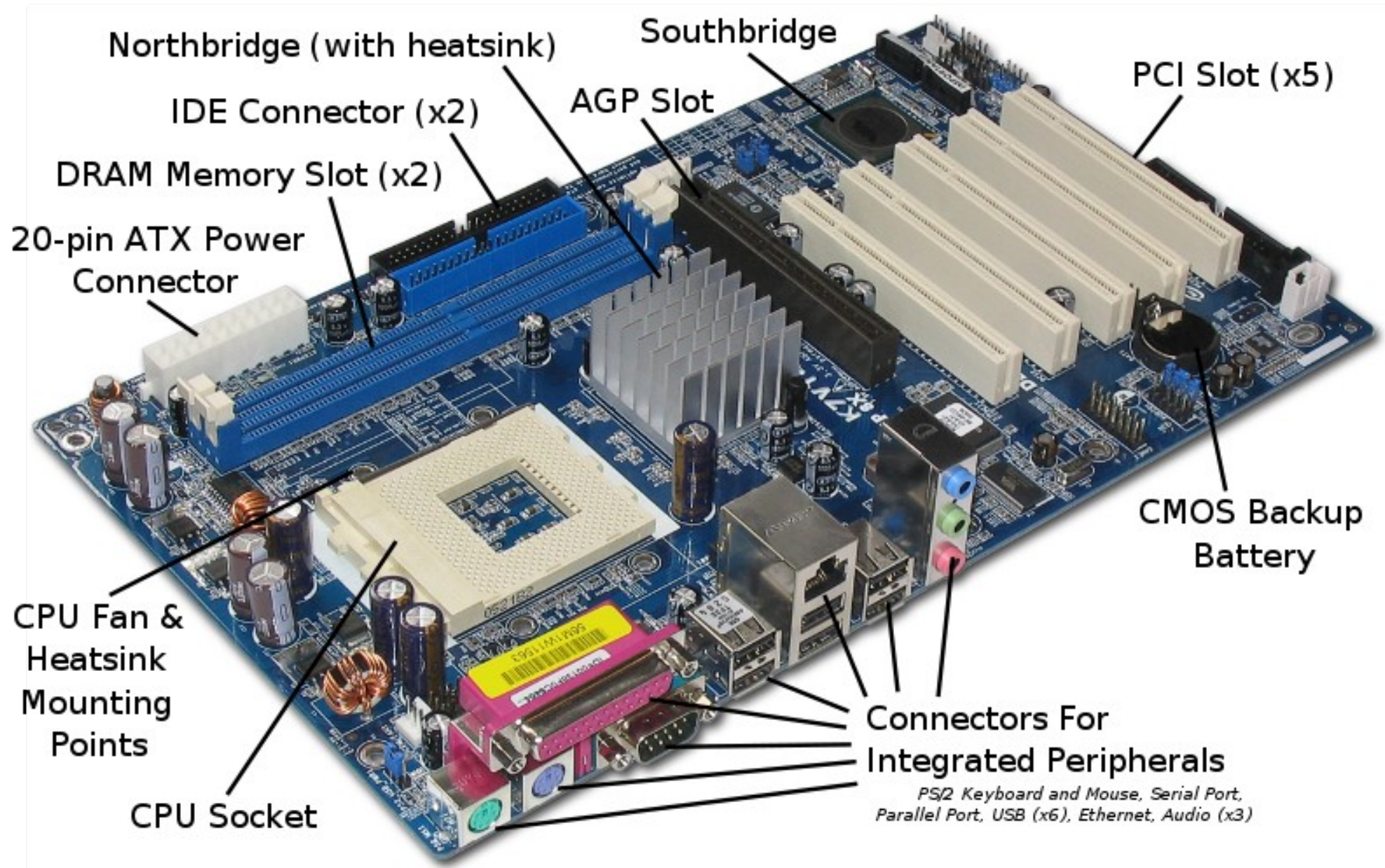
Architettura logica di un computer



Architettura dei calcolatori

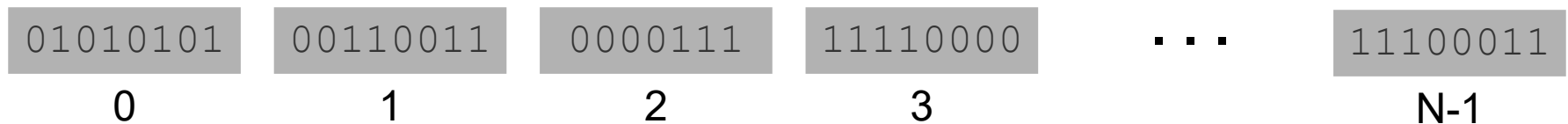
- La CPU è il "cervello" del computer: esegue tutte le operazioni richieste dai programmi (non solo operazioni aritmetiche, ma anche operazioni logiche, confronti, spostamento di dati da/verso la memoria o le periferiche)
- La memoria principale (RAM) è una memoria **volatile** (perde il contenuto quando non è alimentata)
- La RAM è una **memoria di lavoro**: il suo scopo è contenere i programmi e i dati utilizzati in un particolare istante, in modo che siano rapidamente accessibili da parte della CPU
- La CPU può eseguire un programma solo se questo è caricato in RAM (non può leggere un programma direttamente da disco)
- La memoria di massa (hard disk o disco SSD) è una memoria permanente (il contenuto permane anche a computer spento)

Uno sguardo all'interno di un PC



La memoria principale

- La memoria principale può essere pensata come una sequenza di **celle** (o **locazioni**) di memoria, ognuna in grado di contenere 8 bit

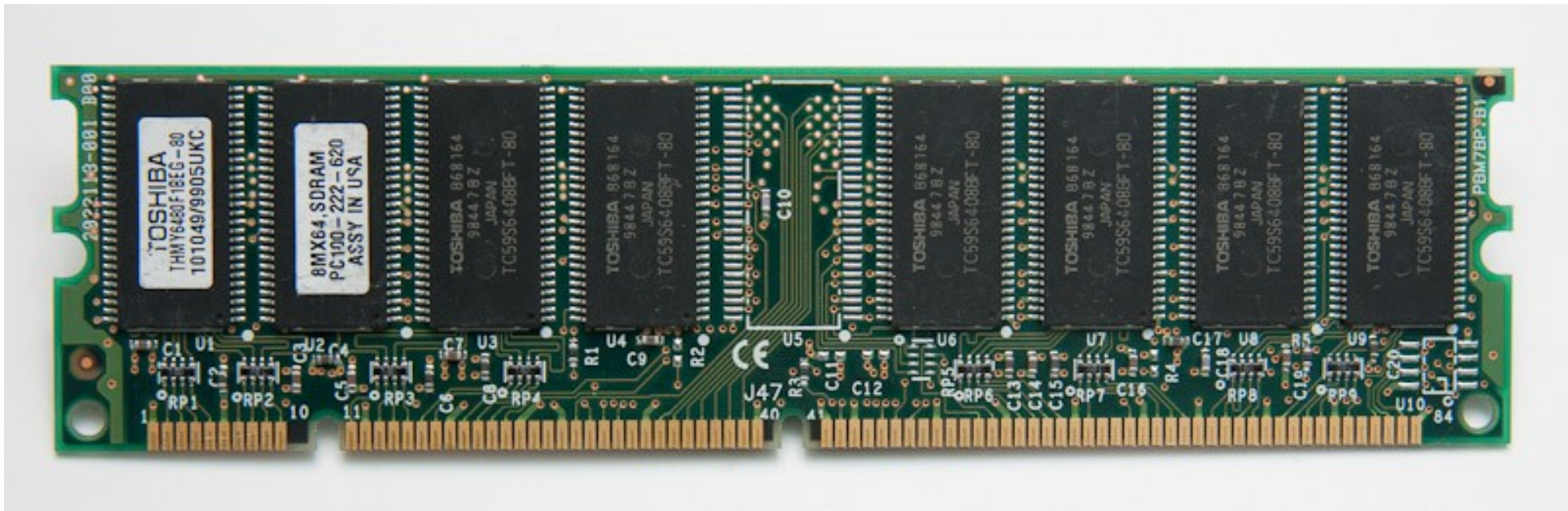


- Ad ogni cella è identificata da un numero progressivo, detto **indirizzo di memoria** della cella
- Esempio: computer con 8 GB di RAM

poichè $8 \text{ GB} = 8 * 2^{30} \text{ byte} = 8.589.934.592 \text{ byte}$

la prima cella avrà indirizzo 0 e l'ultima avrà indirizzo 8.589.934.591

Un modulo di memoria RAM



Comunicare con la memoria

- La CPU può richiedere alla memoria due tipi di operazioni (Lettura o Scrittura di una cella di memoria)
- La memoria principale comunica con il resto del sistema attraverso una serie di **pin** (piedini) o **linee**
- Attraverso questi pin la memoria può inviare o ricevere segnali binari di tensione (0=tensione bassa , 1=tensione bassa)
- Alcuni pin sono usati solo come **pin di input**: qualche componente esterno imposta (**scrive**) livello di tensione sul pin, e la memoria rileva (**legge**) il valore di tensione impostato
- Altri pin sono usati come **pin di output**: la memoria scrive un valore sul pin, e un componente esterno ne legge il valore
- Alcuni pin possono fungere sia da input che da output in momenti diversi

Linee per dati, indirizzi e controllo

I pin di una memoria si possono suddividere in:

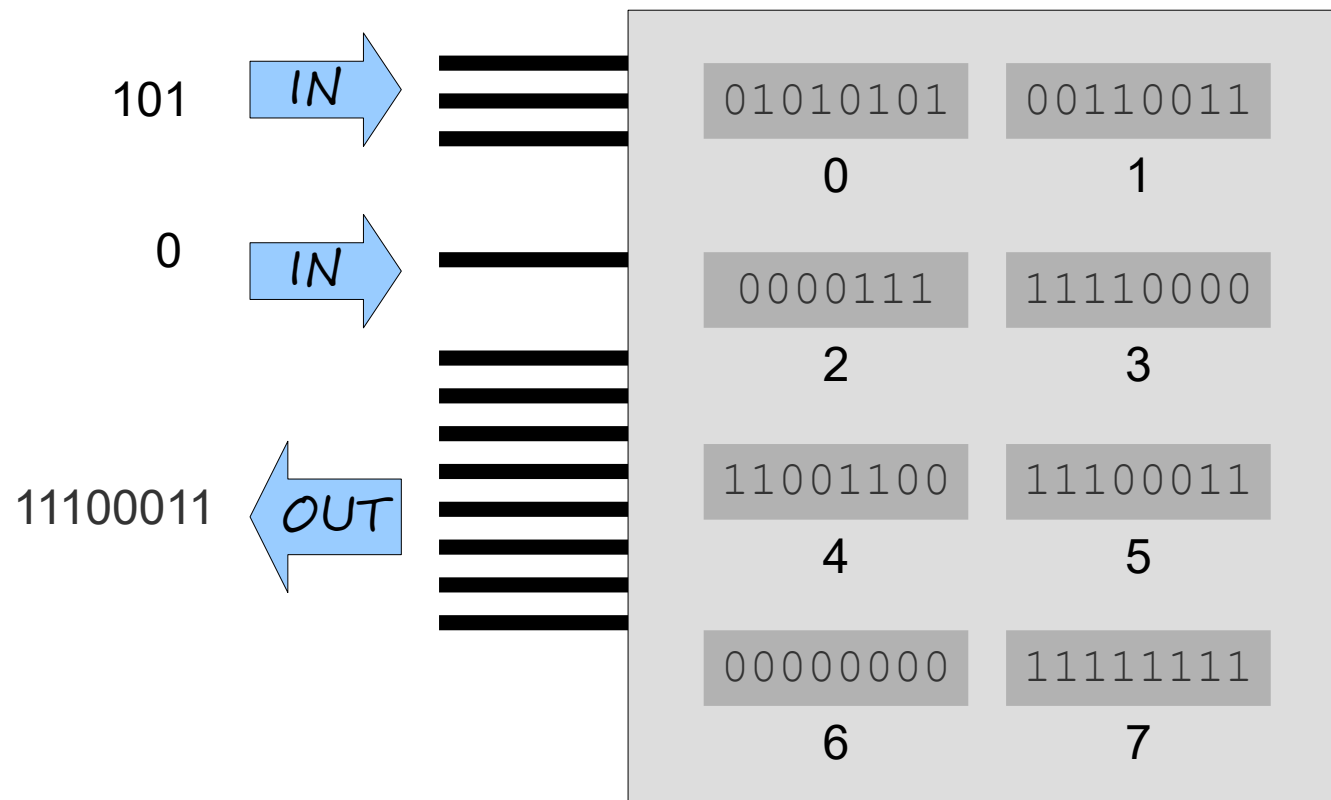
- Pin (o linee) di **controllo** (input)
 - Pin (o linee) di **indirizzo** (input)
 - Pin (o linee) **dati** (input/output a seconda dell'operazione)
-
- Le linee di controllo servono a indicare il tipo di operazione (lettura o scrittura) che vogliamo effettuare sulla memoria. Esempio (usando un solo pin di controllo):
0 = Lettura del contenuto di una cella di memoria
1 = Scrittura del contenuto di una cella di memoria

Linee dati, indirizzi e controllo

- Le linee di indirizzo servono a comunicare alla memoria qual è l'indirizzo (in binario!) della cella interessata dall'operazione di lettura o scrittura
- Le linee di dati servono per:
 - **restituire** al mondo esterno il contenuto della cella, se è stata richiesta un'operazione di lettura (in questo caso le linee dati sono utilizzate dalla memoria come linee di **output**)
 - **ricevere** dall'esterno il nuovo contenuto della cella, se è stata richiesta un'operazione di scrittura (in questo caso le linee dati sono utilizzate dalla memoria come linee di **input**)

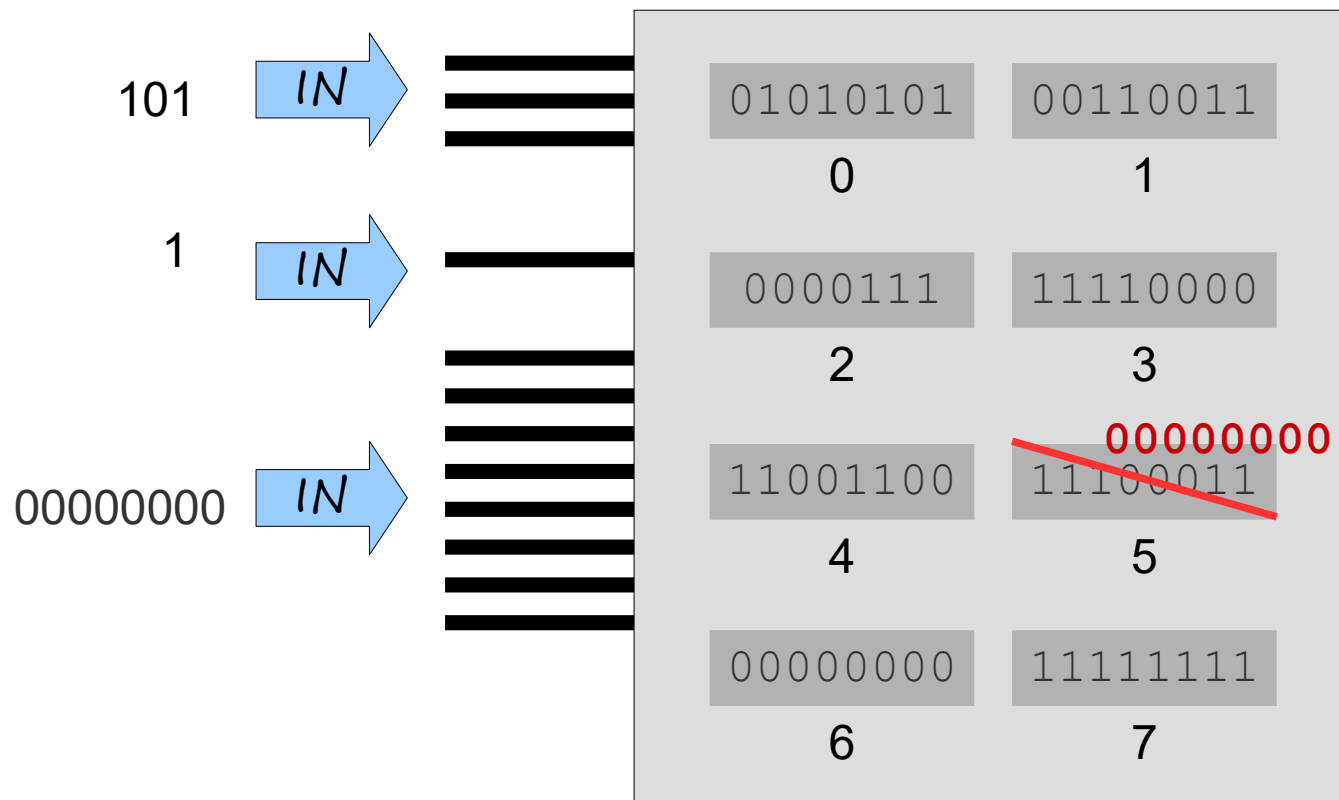
Esempio: operazione di lettura

- Supponiamo che la CPU voglia conoscere (cioè **leggere**) il contenuto della locazione di indirizzo 5 della RAM



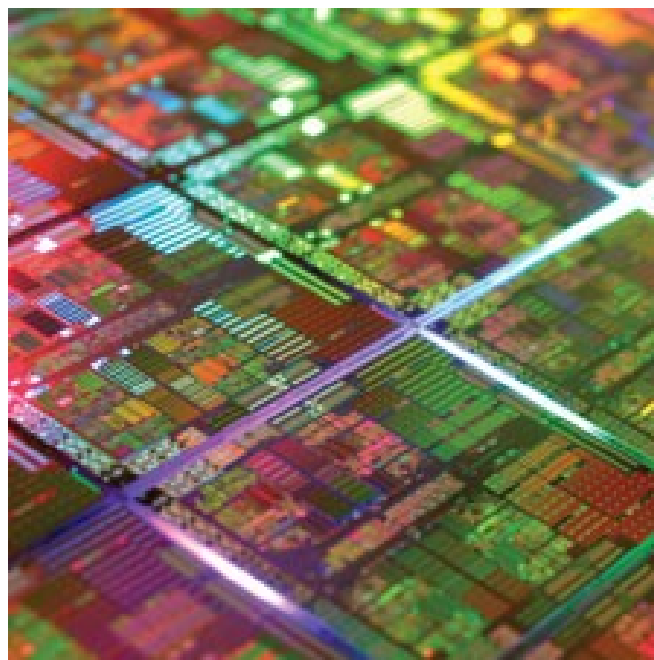
Esempio: operazione di scrittura

- Supponiamo che la CPU voglia conoscere (cioè **leggere**) il contenuto della locazione di indirizzo 5 della RAM



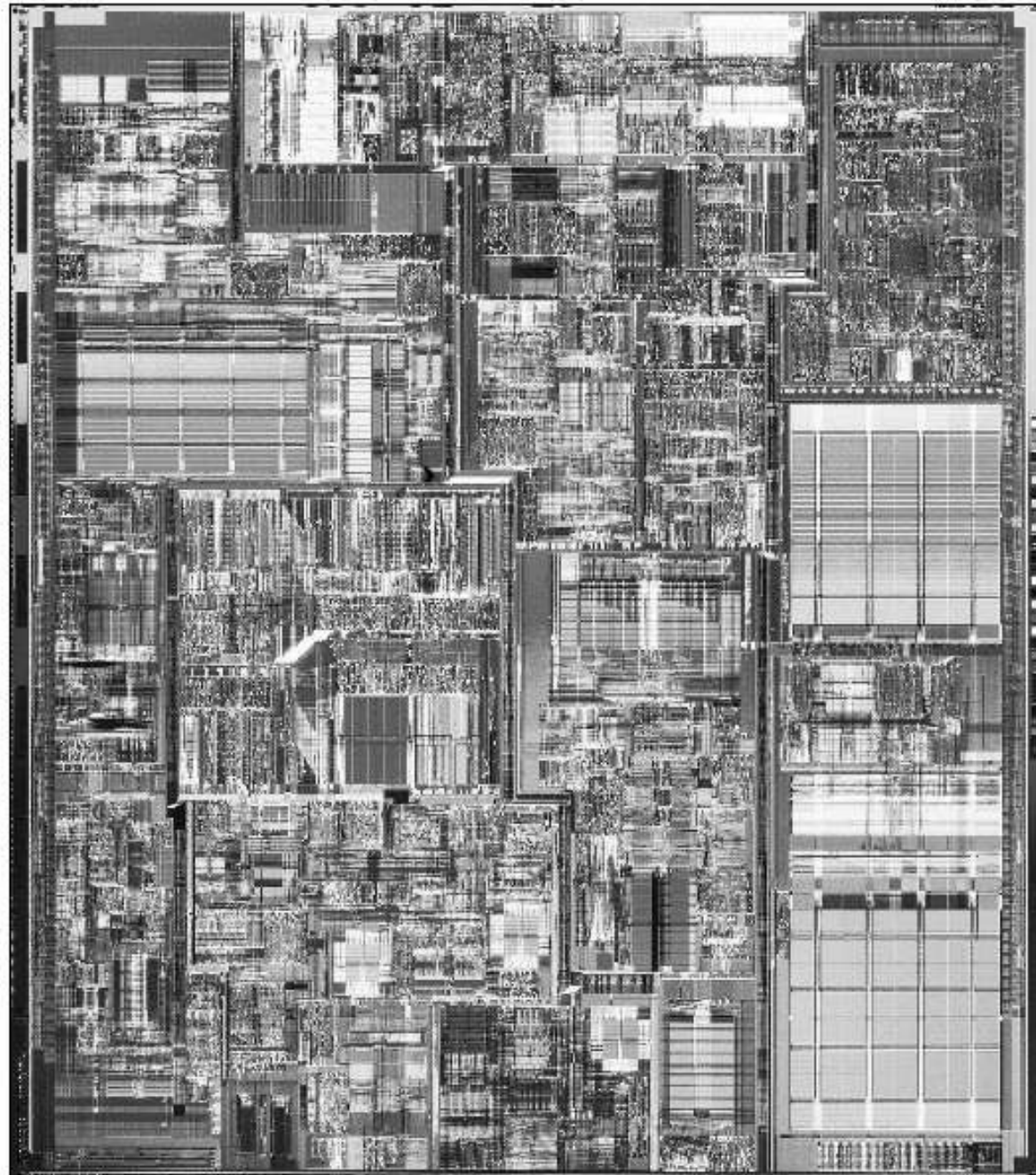
La CPU

- CPU = **Central Processing Unit** (unità centrale di elaborazione)
- Fisicamente, la CPU è un **chip** di silicio su cui vengono realizzati milioni di componenti elettronici detti **transistor**

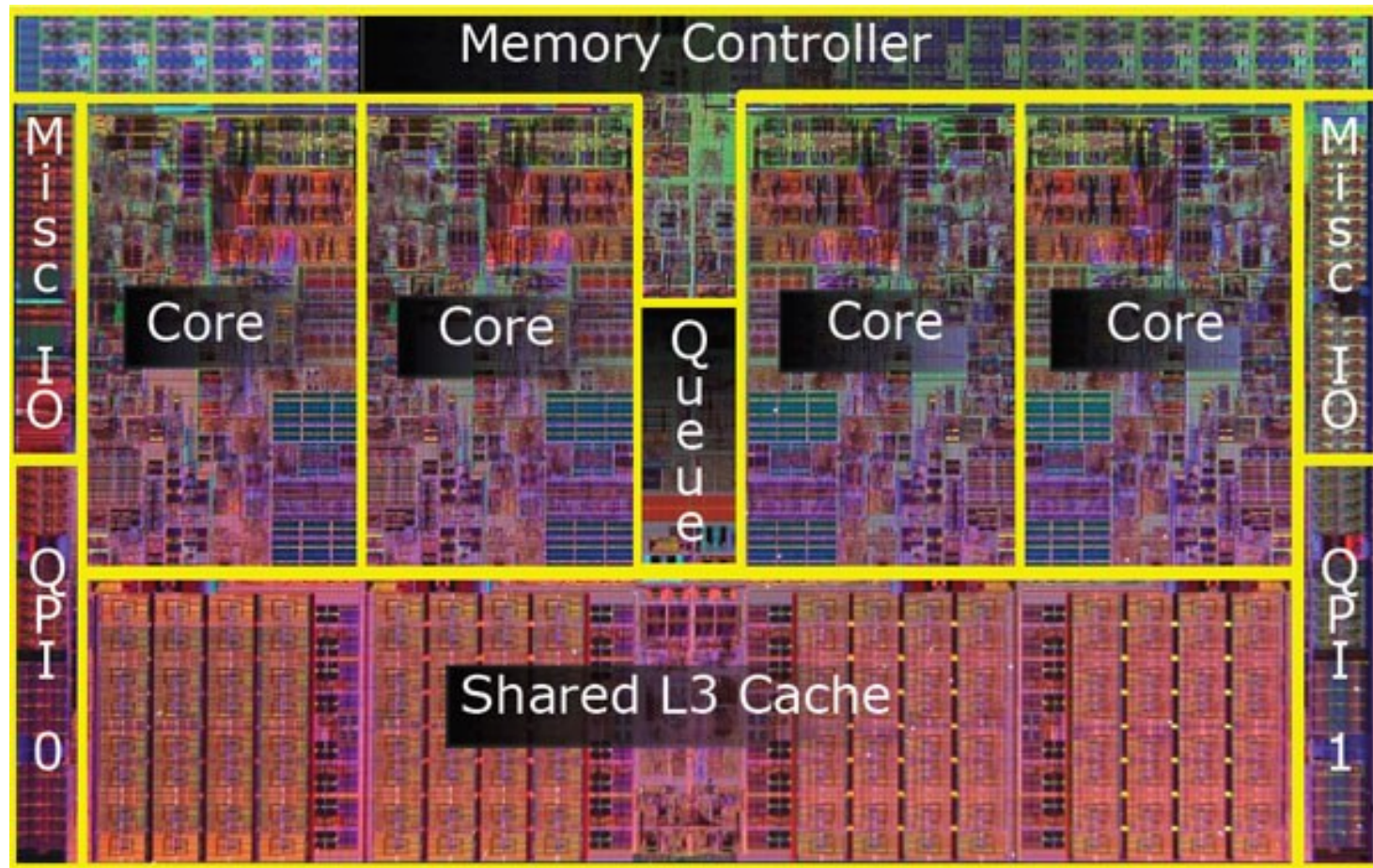


- Il transistor è il componente fondamentale dell'elettronica digitale (è il "mattone di base" con cui costruire porte logiche e quindi memorie, unità aritmetiche, e infine CPU)

Intel Pentium 4



Intel Core i-7



Legge di Moore

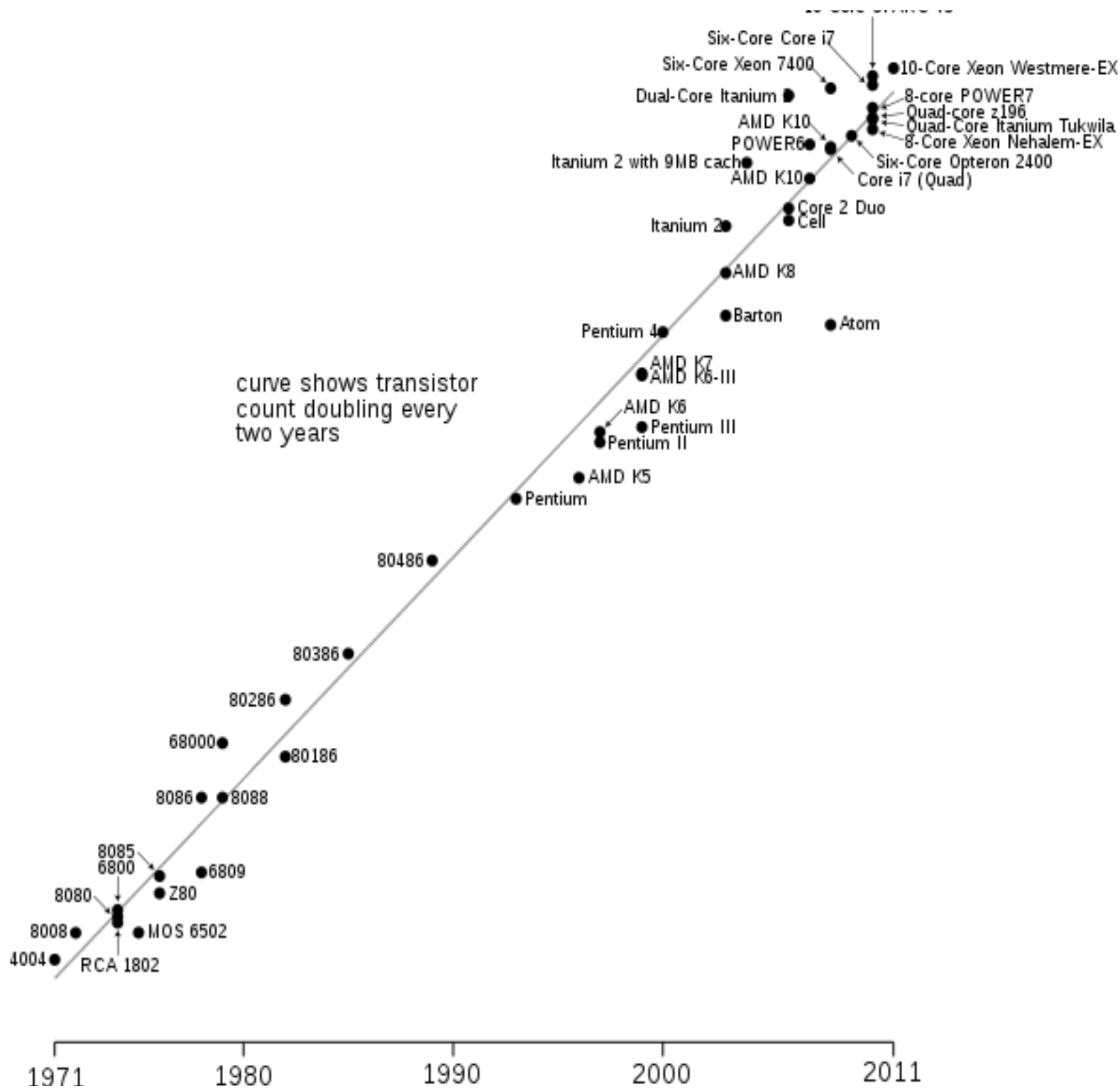
- La **legge di Moore** è una legge empirica che afferma che, con il progredire della tecnologia, il numero di transistor memorizzabili in un chip raddoppia ogni circa due anni circa (secondo alcuni, ogni 18 mesi)
- Se si produce un grafico in cui si dispongono sull'asse x le data di introduzione sul mercato di vari processori, e sull'asse y le quantità di transistor in essi contenuti, si nota che il numero di transistor aumenta in modo esponenziale

Legge di Moore

- La **legge di Moore** è una legge empirica che afferma che, con il progredire della tecnologia, il numero di transistor memorizzabili in un chip raddoppia ogni circa due anni circa (secondo alcuni, ogni 18 mesi)
- Se si produce un grafico in cui si dispongono sull'asse x le data di introduzione sul mercato di vari processori, e sull'asse y le quantità di transistor in essi contenuti, si nota che il numero di transistor aumenta in modo esponenziale

Transistor count

2,600,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
2,300

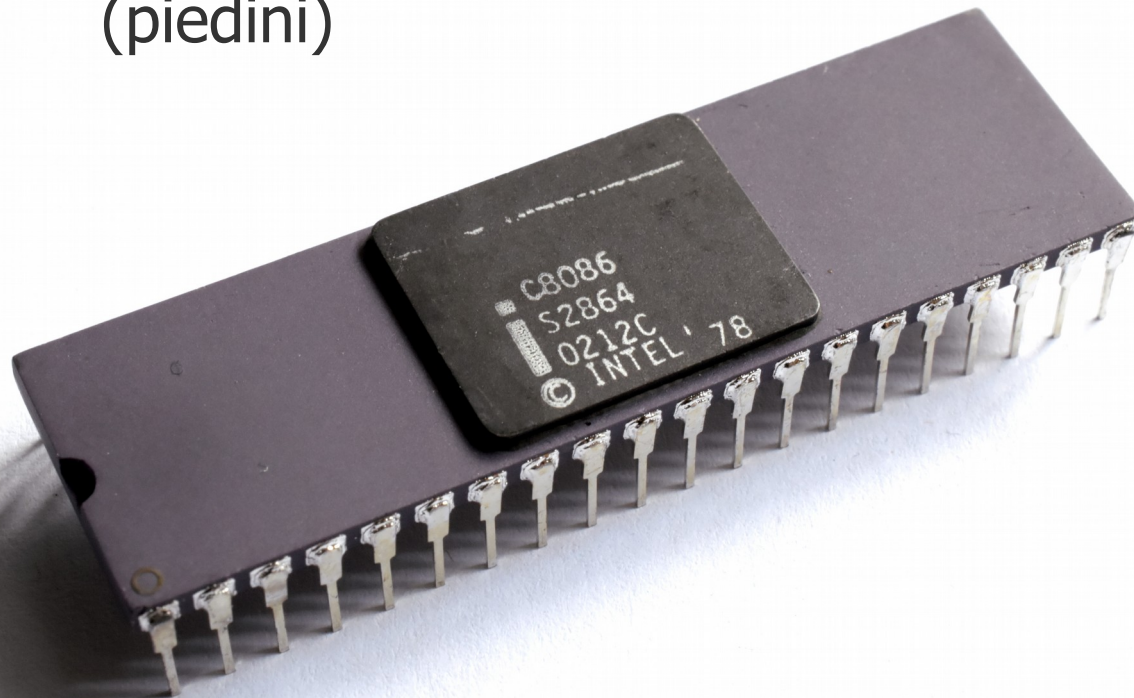


Legge di Moore

- La legge afferma che la densità dei transistor (numero di transistor per cm^2) aumenta quindi in modo esponenziale
- Ciò significa che, a parità di dimensioni del chip, ogni anno l'evoluzione delle tecnologie costruttive consente di realizzare su quel chip processori più complessi o memorie più capienti
- Problema: la dimensione minima dei transistor deve sottostare a limiti fisici (dimensioni degli atomi, surriscaldamento del chip, temperature di fusione...)
- La legge di Moore sta quindi perdendo di validità

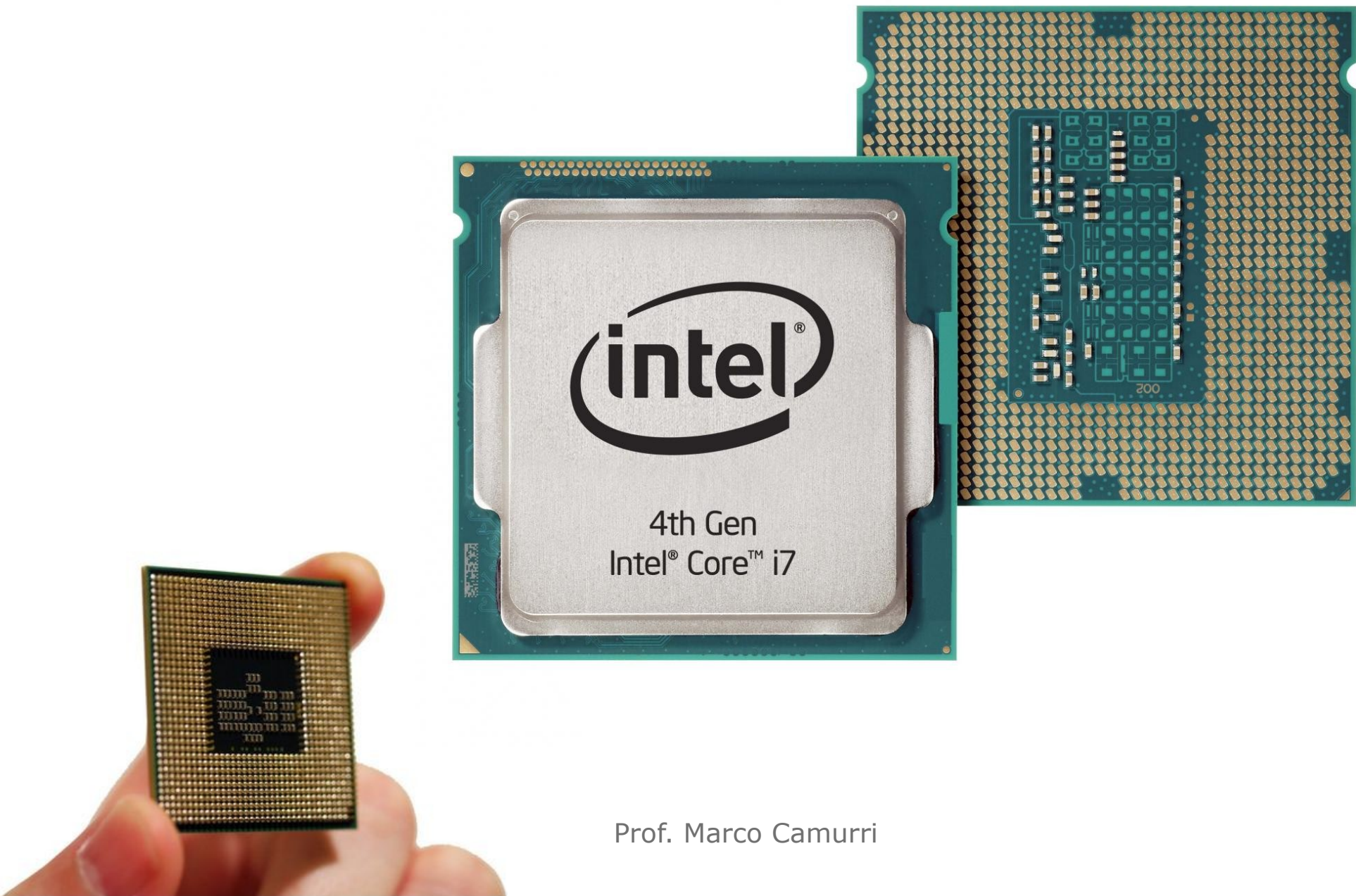
Piedinatura della CPU

- Il chip della CPU è inserito in **package** (contenitore) e comunica con il resto del sistema attraverso un insieme di **pin** (piedini)



				MAX MODE	MIN MODE
Vss (GND)	1	40	Vcc (5V)		
AD14	2	39	AD15		
AD13	3	38	A16/S3		
AD12	4	37	A17/S4		
AD11	5	36	A18/S5		
AD10	6	35	A19/S6		
AD9	7	34	$\overline{BHE}/S7$		
AD8	8	33	$\overline{MN}/\overline{MX}$		
AD7	9	32	\overline{RD}		
AD6	10	31	$\overline{RQ}/\overline{GT0}$	HOLD	
AD5	11	30	$\overline{RQ}/\overline{GT1}$	HLDA	
AD4	12	29	\overline{LOCK}	\overline{WR}	
AD3	13	28	$\overline{S2}$	\overline{MIO}	
AD2	14	27	$\overline{S1}$	$\overline{DT}/\overline{R}$	
AD1	15	26	$\overline{S0}$	\overline{DEN}	
AD0	16	25	QS0	ALE	
NMI	17	24	QS1	\overline{INTA}	
INTR	18	23	\overline{TEST}		
CLK	19	22	READY		
Vss (GND)	20	21	RESET		

Intel Core i7



Pin di input e output

- Attraverso i pin la CPU può inviare o ricevere segnali binari di tensione (0=tensione bassa , 1=tensione alta)
- Alcuni pin sono usati solo come **pin di input**: qualche componente esterno imposta un livello di tensione sul pin, e la CPU rileva il valore di tensione impostato
- In questo caso il componente esterno **scrive** un valore binario sul pin, e la CPU **legge** il valore di quel pin
- Altri pin sono usati come **pin di output**: la CPU scrive un valore sul pin, e un componente esterno ne legge il valore

Funzione e prestazioni della CPU

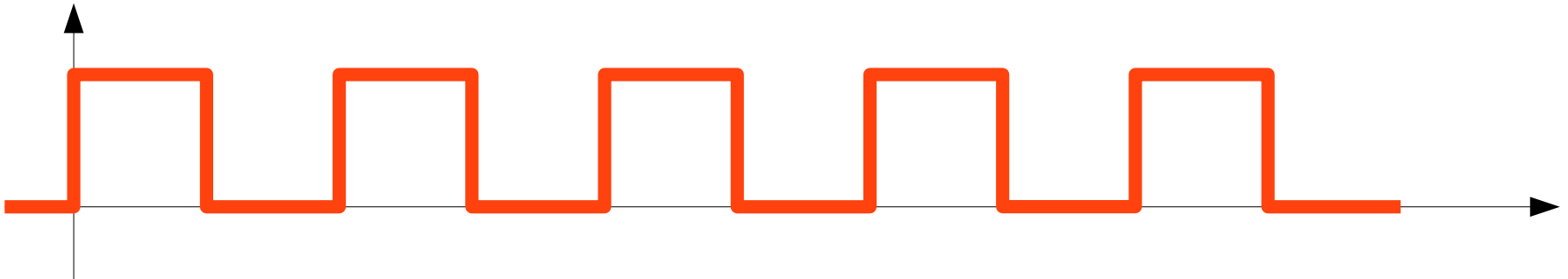
- La CPU è il componente che esegue le **istruzioni** di cui sono composti i programmi
- Le istruzioni eseguibili dalla CPU sono dette **istruzioni macchina**
- La velocità di una CPU può essere misurata in **MIPS** (**Million Instruction Per Second**)
- A volte si usa scorrettamente la frequenza di clock per confrontare la velocità di CPU diverse... (vedi slide successive)

Frequenza di clock

- Per eseguire le istruzioni, la CPU deve attraversare numerosi stati (si può modellare la CPU come un automa a stati finiti molto complesso!)
- La frequenza con cui la CPU è in grado di cambiare stato è detta **frequenza di clock** e si misura in **Hertz** (Hz)
- 1 Hz = 1 volta al secondo
- Frequenze tipiche delle CPU attuali: 2-3 GHz
2 GHz = 2 miliardi di volte al secondo

Segnale di clock

- La frequenza di clock della CPU è stabilita da un circuito esterno (oscillatore) che genera un segnale periodico (segnale che si ripete "uguale a sè stesso" periodicamente nel tempo)



- La frequenza con cui il segnale si ripete è la frequenza di clock
- Questo segnale arriva alla CPU attraverso un apposito pin di input, e detta il "ritmo" di funzionamento della CPU
- la CPU può cambiare stato ad ogni **fronte di salita** del segnale

Clock e prestazioni della CPU

- Confrontare la velocità di due CPU diverse basandosi solo sulla frequenza di clock è SBAGLIATO perchè:
 - Per eseguire un'istruzione, una CPU deve attraversare un certo numero di stati, e quindi "utilizza" un certo numero di cicli di clock
 - Per eseguire la stessa istruzione, CPU diverse utilizzano in generale un numero diverso di cicli di clock

Cicli di clock e prestazioni

■ Problema

La CPU1 ha un clock a 3 GHz, e utilizza 3 cicli di clock per istruzione

La CPU2 ha un clock a 2.5 GHz, e utilizza 2 cicli di clock per istruzione

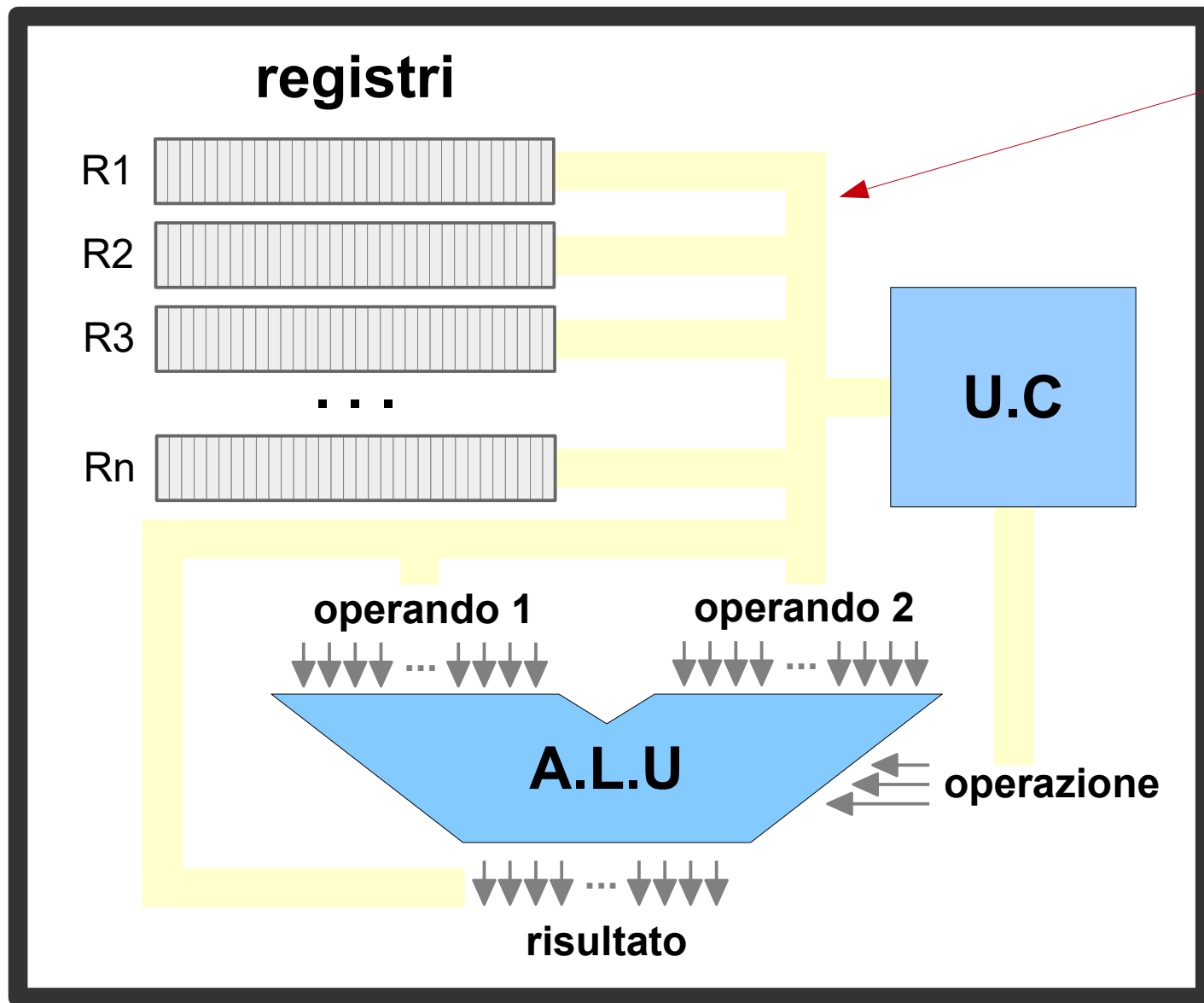
Quale CPU è più veloce (esegue più istruzioni al secondo) ?

■ Soluzione

■ CPU1: $3 \text{ GHz} / 3 = 1$ miliardo di istruzioni al secondo

■ CPU2: $2.5 \text{ GHz} / 2 = 1.25$ miliardi di istruzioni al secondo

Struttura interna della CPU



Bus interno:
consente, ad esempio, di portare il contenuto dei registri alla ALU

Componenti interni della CPU

Ogni CPU contiene i seguenti componenti fondamentali:

- Registri
- Unità aritmetico logico (A.L.U)
- Unità di controllo (C.U)
- Bus interno che collega i componenti precedenti

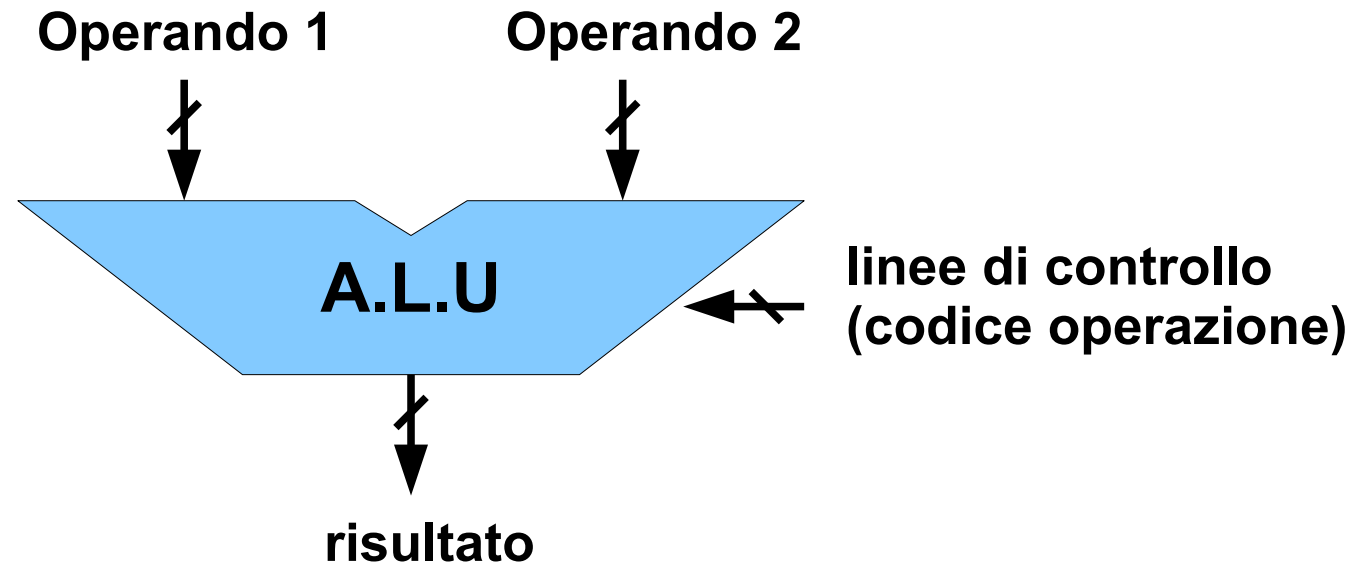
Struttura interna della CPU

- Ogni CPU contiene un certo numero di registri
- Ogni registro è una memoria in grado di memorizzare un numero fisso di bit (valori tipici: 8, 16, 32 o 64 bit)
- Il numero di bit memorizzabili in un singolo registro è detto **parallelismo del processore** e determina l'architettura del processore (architettura a 32 bit, architettura a 64 bit, ecc..)
- Esempio: "CPU a 64 bit" significa "CPU con registri interni a 64 bit"

ALU

- ALU = Arithmetic and Logic Unit
- La ALU è la componente della CPU in grado di eseguire le operazioni **aritmetiche** (somme, sottrazioni, moltiplicazioni, divisioni, ...) e **logiche** (NOT, AND, OR, XOR, ...)
- La ALU è una **rete logica combinatoria** (cioè un circuito "senza memoria" o "senza stato") che riceve in input:
 - le due sequenze di bit da elaborare (**operandi**)
 - l'operazione da eseguire sulle sequenze (somma, AND, ecc..)e fornisce in output il risultato dell'operazione
- Poichè la ALU è una rete combinatoria, l'**output corrente** dipende esclusivamente dall'**input corrente**

ALU



- Nota: per convenzione una freccia barrata indica che essa rappresenta un certo numero di linee (e non una sola linea)

Numero di linee di controllo della ALU

- il numero di linee di controllo della ALU dipende dal numero di operazioni supportate dalla ALU
- Con N linee di controllo è possibile codificare 2^N operazioni (Motivo: N bit danno luogo a 2^N configurazioni diverse)
- Esempio: per gestire 5 operazioni (SOMMA, SOTTRAZIONE, AND, OR, NOT) servono 3 linee di controllo

000 = SOMMA

011 = OR

110 = non usata

001 = SOTTRAZIONE

100 = NOT

111 = non usata

010 = AND

101 = non usata

Esempio: ALU a 8 bit con 4 operazioni

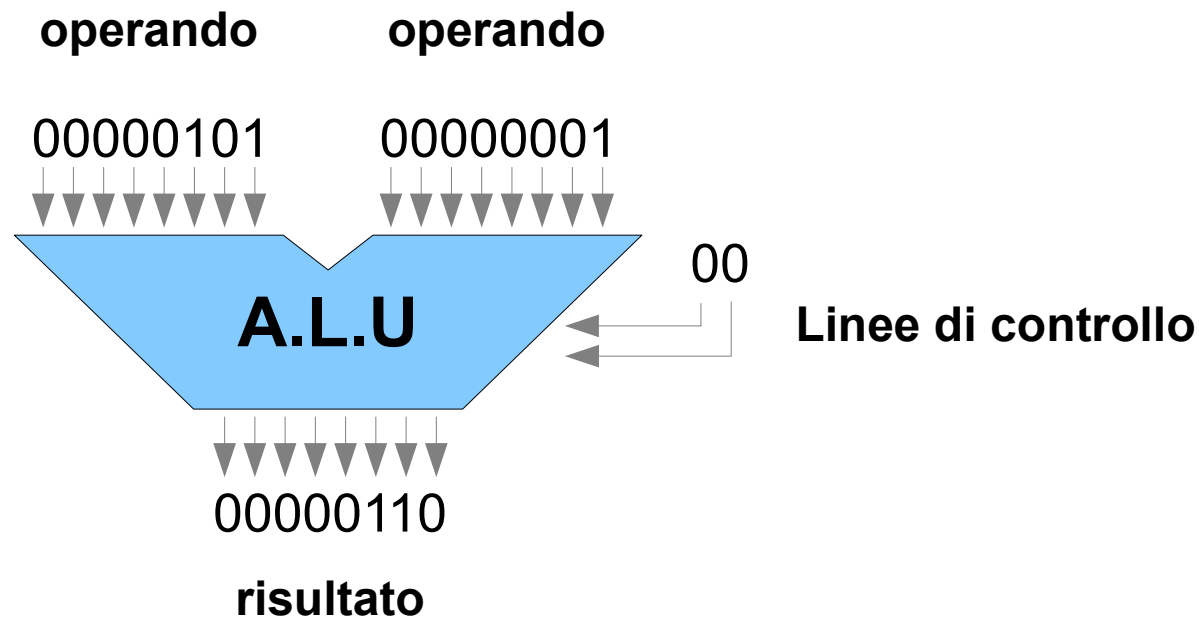
- "ALU a 8 bit" significa che gli operandi sono di 8 bit ciascuno
- Per gestire 4 operazioni occorrono almeno 2 linee di controllo

00 = SOMMA

01 = SOTTRAZIONE

10 = AND

11 = OR



Unità di controllo

- L'unità di controllo è il componente della CPU che si occupa di generare tutti i segnali di controllo necessari a prelevare le istruzioni dalla memoria principale ed eseguirle
- Esempio: se la CPU deve eseguire una sottrazione tra il registro R1 e il registro R2, è l'unità di controllo a generare i segnali necessari affinché i contenuti dei due registri arrivino agli ingressi della ALU, e affinché la ALU esegua una sottrazione