Architettura del Calcolatore

Modulo 1

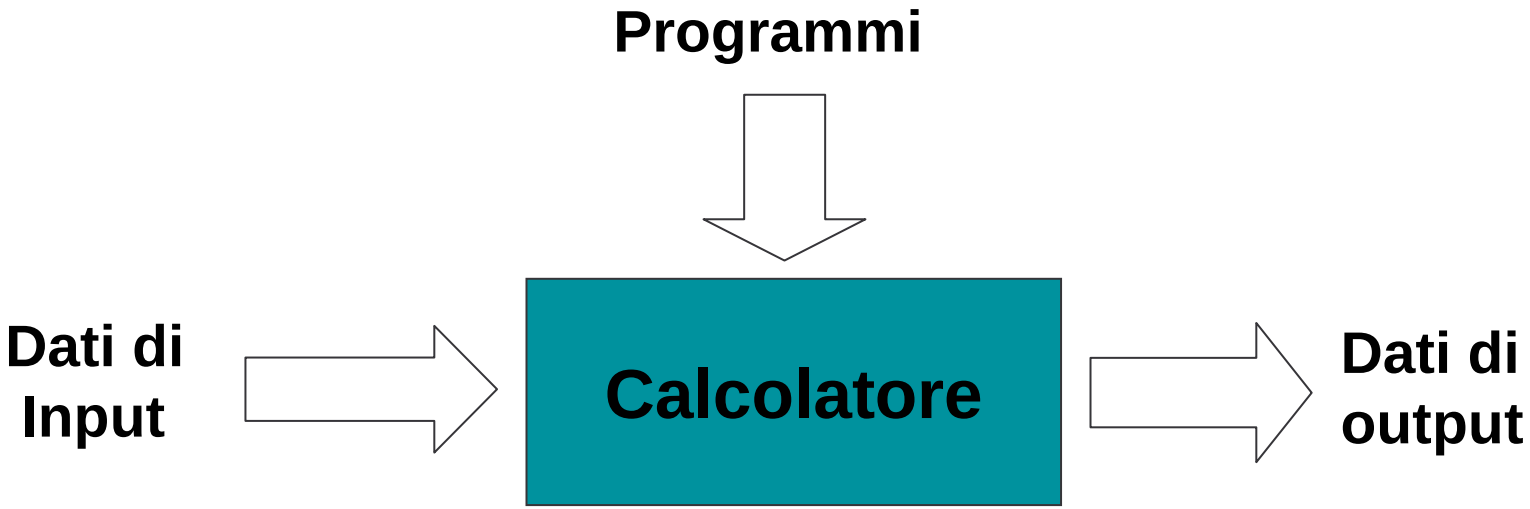
C. Marrocco

Università degli Studi di Cassino

Che cos’è un calcolatore ?

Un calcolatore può essere definito in diversi modi:

- una macchina di uso generale riconfigurabile.
- una macchina che esegue automaticamente un compito definito in base ad un programma specificato dall’utente.
- Una macchina che riceve dati in ingresso, li memorizza, li elabora sulla base di una lista di istruzioni, calcola i risultati, li memorizza e li fornisce in uscita.



Caratteristiche fondamentali di un calcolatore

macchina → **hardware**

struttura fisica del calcolatore, definita dall'insieme delle unità funzionali che la compongono e dalle loro interconnessioni

programma → **software**

insieme di istruzioni da eseguire secondo un ordine preciso, il cui effetto è la realizzazione di uno specifico compito

calcolatore = hardware + software

firmware

programmi per computer che sono permanentemente installati nel sistema. A differenza del software, il firmware non può essere alterato dall'utente, in quanto richiede modifiche o sostituzioni dell'hardware.

Definizione di calcolatore

Un calcolatore è un dispositivo fisico che implementa il funzionamento di una **macchina di Turing**.

Alan Turing (1912-1954) nel 1936 definisce una macchina formale per “risolvere” il problema di decisione di Hilbert.

Il problema di halt: data una generica macchina di Turing non è possibile stabilire se l’elaborazione avrà termine in un tempo finito. Una conseguenza di questo fatto è l’affermare l’impossibilità di risolvere in modo automatico una vasta categoria di problemi.

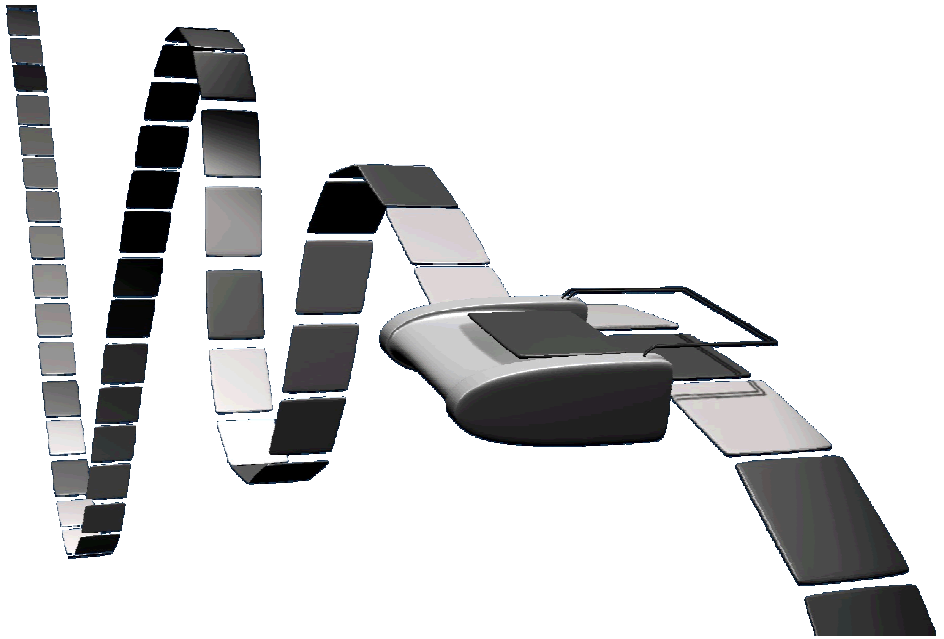
La **tesi di Church** ribadisce che non esiste architettura di calcolatore più potente della macchina di Turing. Praticamente che i problemi risolvibili non vengono influenzati dai mezzi di calcolo adottati.

La macchina di Turing

La macchina di Turing è il modello più semplice di calcolatore; consiste

in due diversi supporti di memoria:

- un **nastro** infinito suddiviso in celle contenenti ognuna un *singolo simbolo* tratto da un alfabeto finito; il nastro funge anche da *supporto per l'inserimento dei dati*, da *supporto di memoria* e da *supporto di uscita*;
- una singola cella contenente un elemento di un insieme finito detto **stato**.



La macchina di Turing

La macchina di Turing può eseguire due operazioni in lettura:

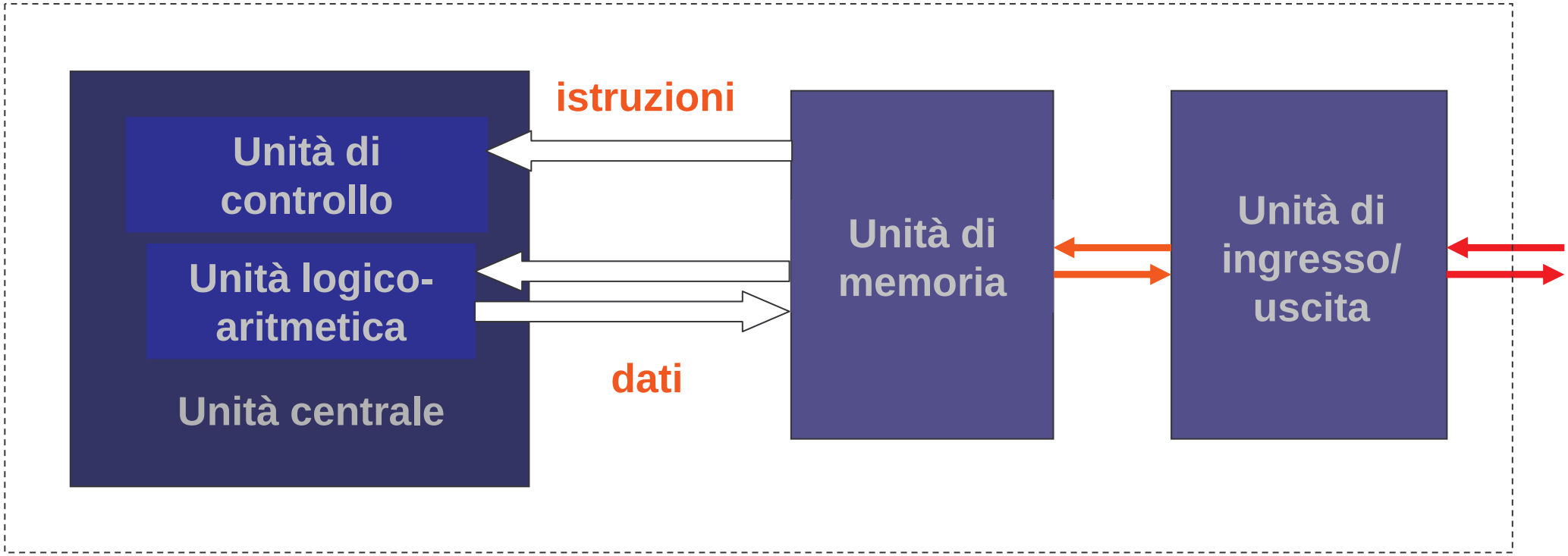
- legge un dato dal nastro;
- legge il suo stato.

E sulla base delle due letture:

- scrive un nuovo simbolo sul nastro al posto di quello letto;
- scrive un nuovo simbolo nella cella di stato;
- sposta la testina di lettura/scrittura di un posto a destra, a sinistra o la lascia dov'è.

In alternativa la macchina viene posta nello *stato di halt* e l'elaborazione termina.

Organizzazione del calcolatore

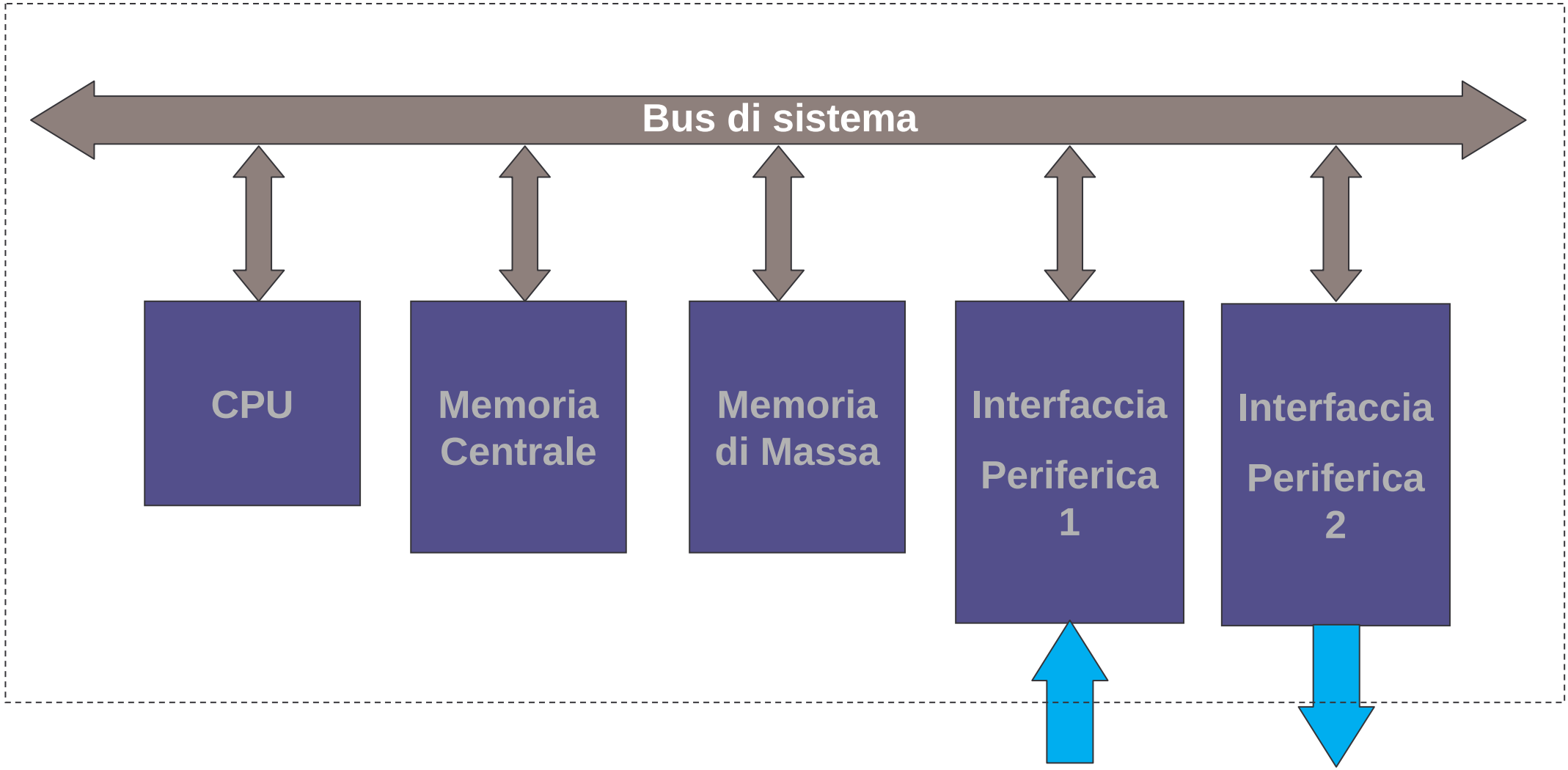


Modello logico

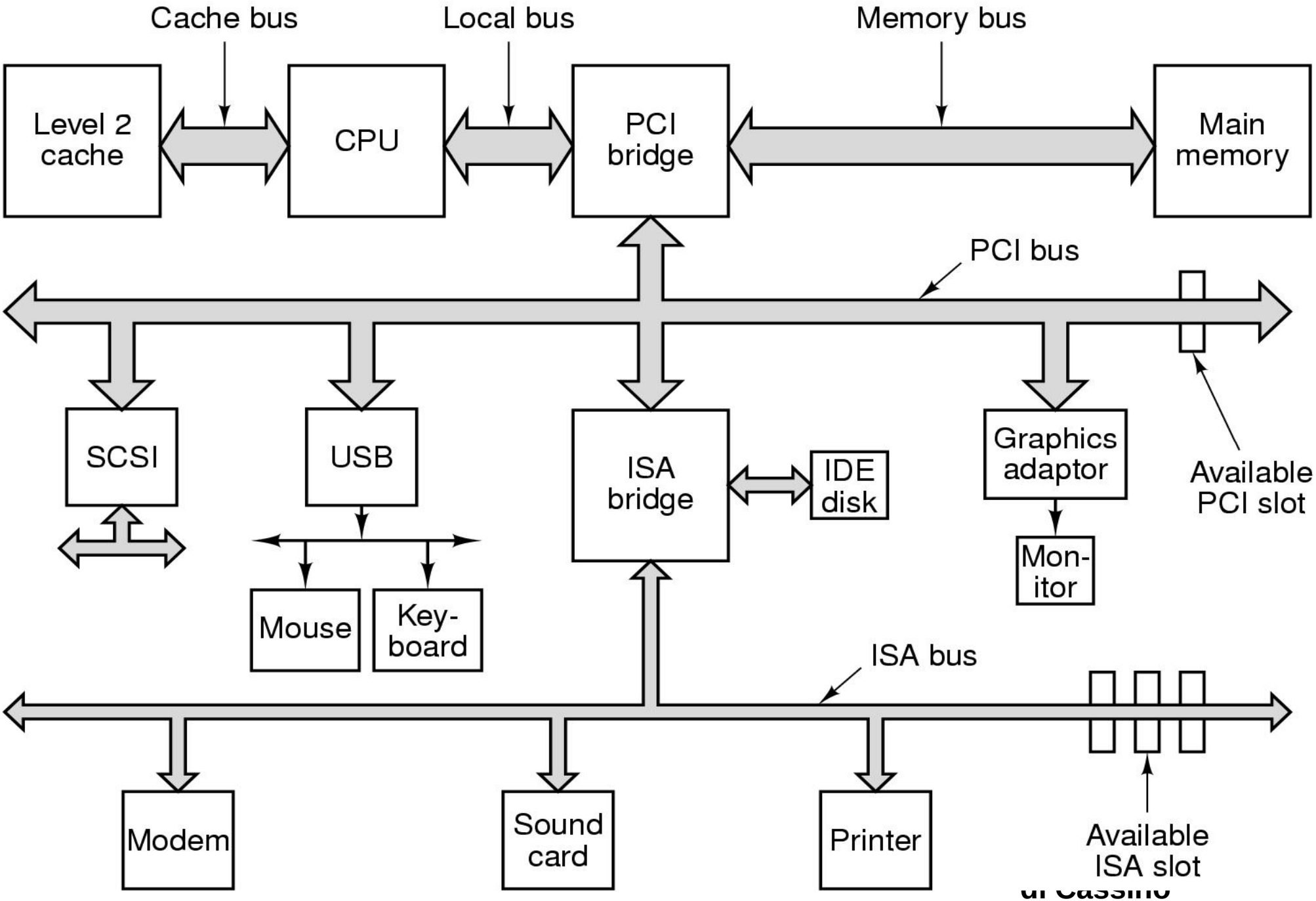
singole componenti

flussi di dati e istruzioni

Modello di von Neumann



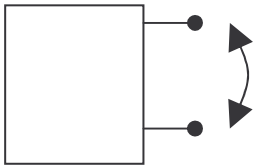
Un esempio


Cache bus
Local bus
Memory bus
Level 2 cache
CPU
PCI bridge
Main memory
PCI bus
SCSI
USB
ISA bridge
IDE disk
Graphics adaptor
Available PCI slot
Mon-itor
Mouse
Key-board
ISA bus
Modem
Sound card
Printer
Available ISA slot

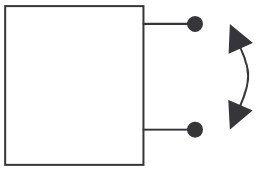
La memorizzazione dei dati e delle istruzioni

La più piccola unità di informazione memorizzabile (e quindi utilizzabile) è il **bit (Binary digIT)**, che può assumere valore 0 o 1.

Il dispositivo utilizzato per memorizzare un bit è un **elemento bistabile**, cioè un dispositivo elettronico che può assumere uno tra due stati stabili (es. due livelli differenti di tensione), ognuno dei quali viene fatto corrispondere a 0 o a 1 (**cella** di memoria).



$V > V_0$ ⇨ **memorizzato 1** **1**



$V < V_0$ ⇨ **memorizzato 0** **0**

Algebra di Boole

- Boole (1815-1864)
- Nel 1854 Boole delineò un sistema algebrico che venne successivamente denominato *algebra booleana*, nella quale le proposizioni vengono formalizzate con una notazione simbolica e le procedure di calcolo si possono condurre grazie a operatori matematici corrispondenti alle leggi della logica.

Operazioni possibili su una cella di memoria

Operazione di scrittura

La cella di memoria viene caricata con un determinato valore che permane memorizzato finchè:

- la cella viene alimentata elettricamente
- non si esegue un'altra operazione di scrittura che modifica il valore precedentemente memorizzato

Operazione di lettura

Si accede alla cella di memoria per consultarne il valore e copiarlo su un'altra cella di memoria, senza alterarne il contenuto

Nota

Non su tutte le celle di memoria sono possibili entrambe le operazioni di lettura e scrittura.

Con un solo bit è possibile gestire un'informazione binaria, cioè un'informazione che può specificare uno tra due valori possibili (es. un punto di un'immagine bianco o nero).

Quanti stati possibili può assumere un insieme di bit ?

00	000	0000
01	001	0001
10	010	0010
11	011	0011
	100	0100
	101	0101
	110	0110
	111	0111
		1000
		1001
		1010
		1011
		1100
		1101
		1110
		1111

2 bit $\rightarrow$ 4 stati

3 bit $\rightarrow$ 8 stati

4 bit $\rightarrow$ 16 stati

...

Il registro di memoria

Un insieme di N celle elementari può assumere uno tra 2^N stati possibili.

Un tale insieme è organizzato in un **registro** di memoria.

Il registro costituisce un supporto per la memorizzazione di un'informazione che può assumere uno tra 2^N valori possibili. In particolare un insieme di 8 bit forma un **byte**.

Sul registro sono possibili operazioni di lettura e scrittura che interessano contemporaneamente tutte le celle di memoria contenute nel registro.

Il problema della codifica

Un calcolatore può trattare diversi tipi di dati: numeri (interi, reali), testo, immagini, suoni, ecc. che vanno comunque memorizzati su registri di memoria.

È quindi necessario adottare una **codifica** del tipo di dato considerato: occorre, cioè,

mettere in corrispondenza biunivoca i valori del tipo con gli stati che può assumere il registro.

Esempio

registro da un byte $\Rightarrow 2^8 = 256$ stati possibili.

Che cosa è possibile codificare ?

Numeri naturali [0,255]

0 $\leftrightarrow$ 00000000
1 $\leftrightarrow$ 00000001
....
255 $\leftrightarrow$ 11111111

Numeri interi [-128,127]

-128 $\leftrightarrow$ 00000000
-127 $\leftrightarrow$ 00000001
0 $\leftrightarrow$ 10000000
+127 $\leftrightarrow$ 11111111

Numeri reali [0,1[

0.0000 $\leftrightarrow$ 00000000
0.0039 $\leftrightarrow$ 00000001
0.0078 $\leftrightarrow$ 00000010
....
0.9961 $\leftrightarrow$ 11111111

Caratteri

A $\leftrightarrow$ 01000001
a $\leftrightarrow$ 01100001

0 $\leftrightarrow$ 00110000
1 $\leftrightarrow$ 00110001

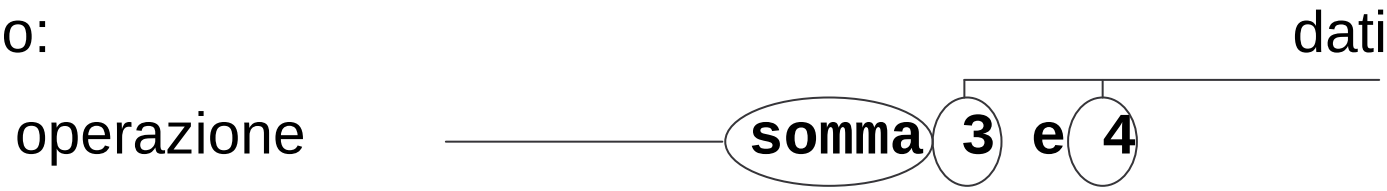
La codifica implica una rappresentazione dei dati limitata e discreta

Codifica delle istruzioni

Oltre ai dati, è necessario memorizzare anche le istruzioni, cioè le singole azioni elementari che l'unità centrale può eseguire.

Nello specificare un'istruzione, bisogna precisare l'operazione da compiere e i dati coinvolti nell'operazione.

Esempio:



Come rappresentare le operazioni ?

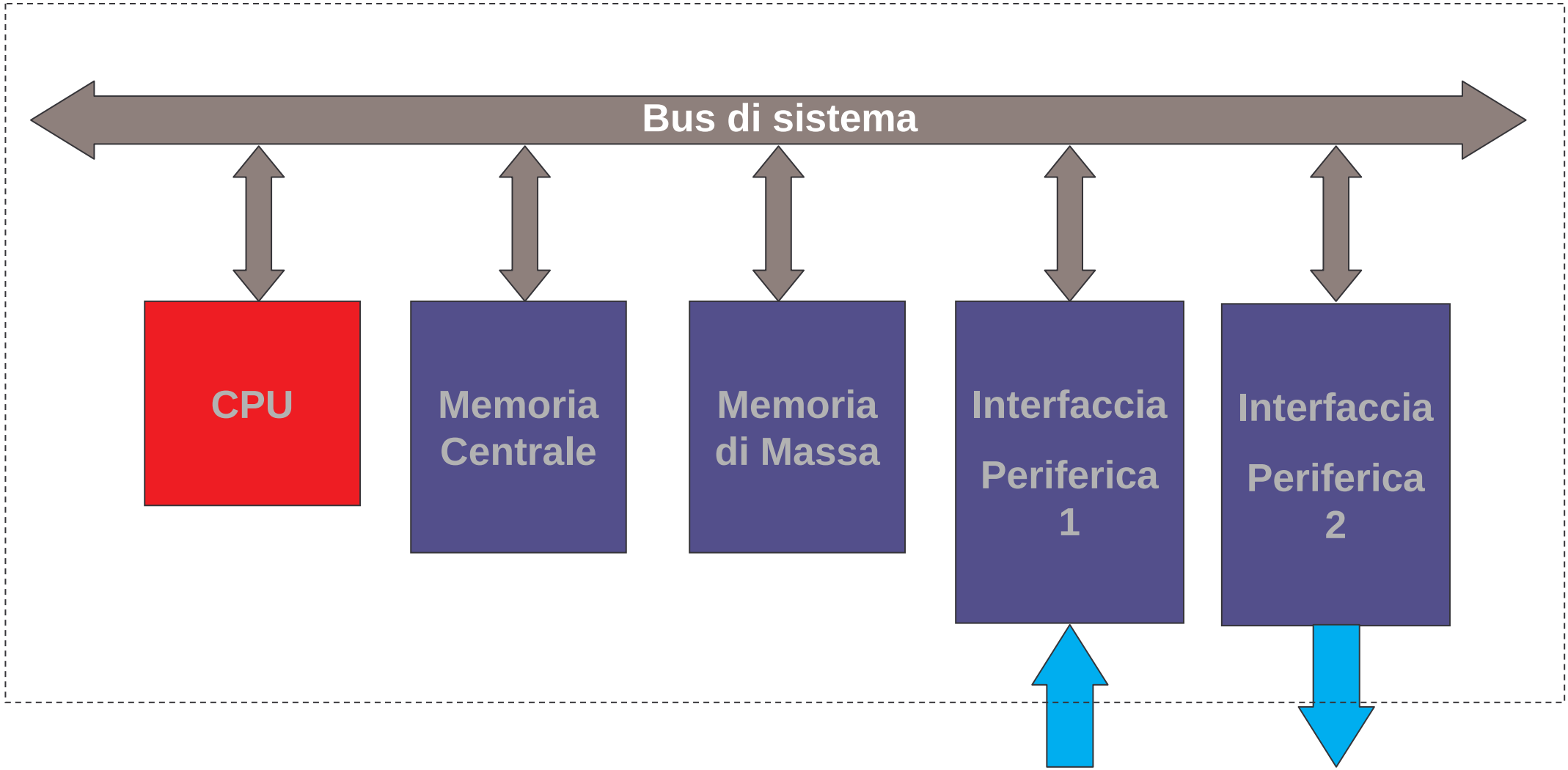
L'insieme delle diverse operazioni che l'unità centrale è in grado di eseguire è finito e quindi è possibile codificarlo con un certo numero di bit (**codice operativo**).

somma	**0000**
sottrai	**0001**
moltiplica	**0010**
dividi	**0011**
…	…

Una istruzione sarà quindi rappresentabile da una sequenza di bit divisa in due parti:

- **un codice operativo**
- **un campo operandi (1, 2 o più operandi)**

Modello di von Neumann



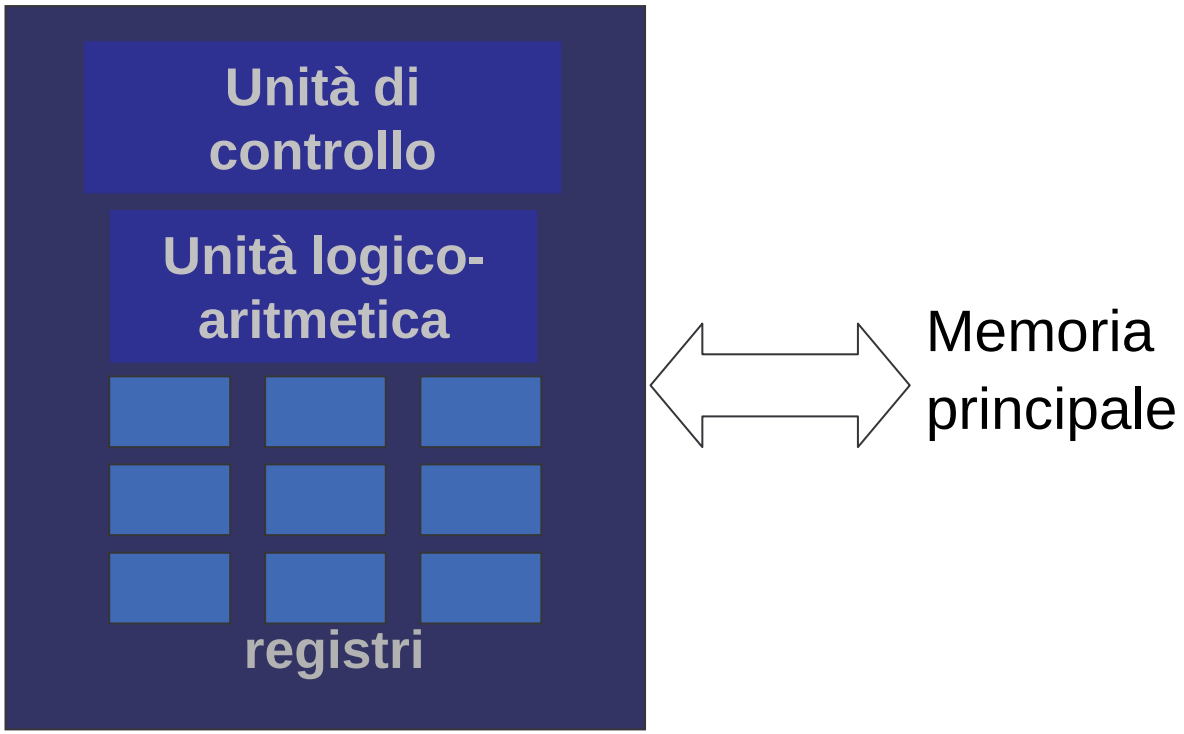
CPU (Central Processing Unit)

Funzione:

eseguire i programmi immagazzinati in memoria principale prelevando le istruzioni (e i dati relativi), interpretandole ed eseguendole una dopo l'altra

E' formata da:

- **unità di controllo**
- **unità logico aritmetica**
- **registri**



La CPU è inoltre caratterizzata dall'insieme delle istruzioni che può eseguire (instruction set)

L’unità di controllo (1/2)

E’ l’unità che si occupa di dirigere e coordinare le attività interne alla CPU che portano all’esecuzione di una istruzione

Ciclo del processore

L’esecuzione di una istruzione avviene attraverso alcune fasi:

Fetch
L’istruzione da eseguire viene prelevata dalla memoria e trasferita all’interno della CPU

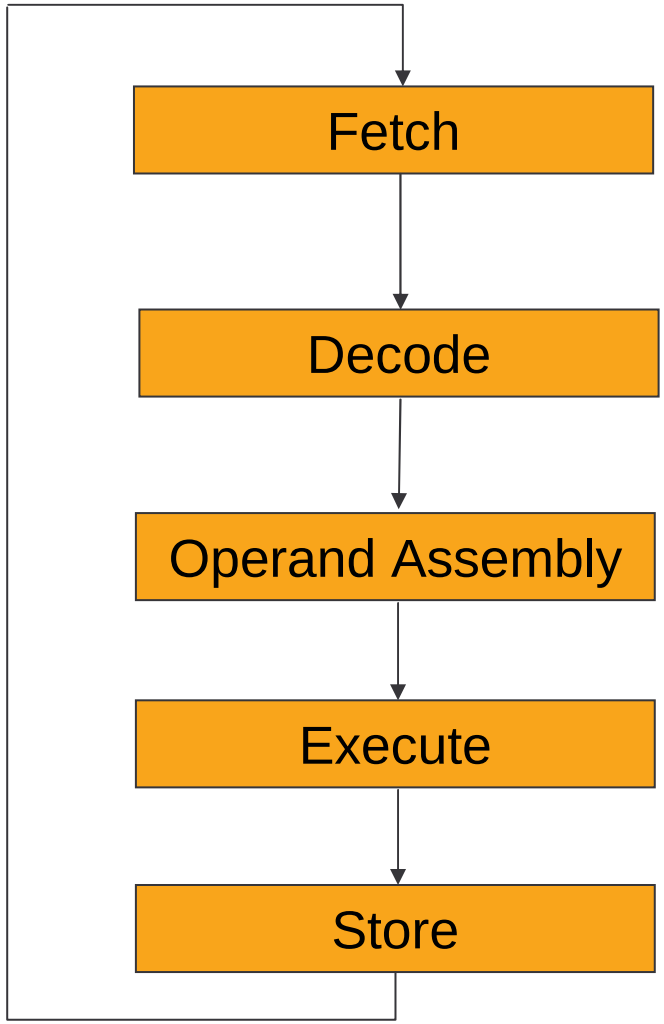
Decode
L’istruzione viene interpretata e vengono avviate le azioni interne necessarie per la sua esecuzione

Operand Assembly
Vengono prelevati dalla memoria i dati su cui eseguire l’ operazione prevista dalla istruzione

Execute
Viene portata a termine l’esecuzione dell’operazione prevista dalla istruzione

Store
Viene memorizzato il risultato dell’operazione prevista dalla istruzione

L’unità di controllo (2/2)



L’unità di controllo realizza in ciclo le fasi per eseguire la sequenza di istruzioni che costituiscono il programma.

L’Unità Logico Aritmetica (ALU)

E’ l’unità che si occupa di realizzare le operazioni logiche ed aritmetiche eventualmente richieste per eseguire un’istruzione

Operazioni Aritmetiche

ADD

SUB

MUL

DIV

REM

SET

Operazioni Logiche

CMP

AND

OR

NOT

I registri

Hanno la funzione di memorizzare all'interno della CPU dati e istruzioni necessari all'esecuzione

- **Registri generali**
- **Registri speciali**
  - Program Counter (PC)
  - Mem. Address Reg. (MAR)
  - Mem. Data Register (MDR)
  - Instruction Register (IR)
  - Interrupt Register (INTR)

I registri speciali non sono accessibili dalle istruzioni

Clock

- **La CPU è sincronizzata da un orologio interno che procede a velocità costante (clock)**
- **I "clock ticks" definiscono gli istanti possibili per la progressione dei singoli passi eseguiti dal processore:**

- **tempo di ciclo= intervallo tra due ticks = secondi per ciclo**
- **clock rate (frequenza) = cicli al secondo (1 Hz. = 1 ciclo/sec)**

1 MegaHertz= 1MHz = 10^6 cicli/sec

1 GigaHertz = 1GHz = 10^9 cicli/sec

Frequenze di clock maggiori indicano CPU più veloci

Tipi di processori

- **Architetture CISC** (*Complex Set Instruction Set Computers*):
  1. Set di istruzioni del linguaggio macchina grande.
  2. Istruzioni sintatticamente omogenee (con uno, nessuno o più operandi e diversi modi di indirizzamento) che implica un linguaggio più semplice.
  3. Ogni istruzione può essere molto complessa e richiedere più cicli di clock e una struttura hardware complessa.

  Esempi sono il **Motorola 680x0**, **Intel 80x86**, **Pentium**.

- **Architetture RISC** (*Reduced Instruction Set Computers*):
  1. Set di poche istruzioni molto semplici.
  2. Istruzioni eseguite direttamente senza traduzione in microcodice con un unico ciclo di clock che implica una maggiore velocità.
  3. Ottimizzazione delle istruzioni usate più di frequente.

  Esempi sono l'**IBM/Motorola Power PC**, **Sun SPARC**, **Digital Alpha**, **MIPS**.

Estensioni dell'architettura di Von Neumann

Per aumentare la velocità della CPU si ricorre a delle modifiche dell'architettura di Von Neumann, tese ad introdurre varie forme di parallelismo:

1. Uso di processori dedicati (**coprocessori**) capaci di eseguire solo alcuni compiti specifici in parallelo ad altre istruzioni eseguite dalla CPU.

2. Uso di processori dedicati (**canali di ingresso/uscita**) per trasferire grandi quantità di dati dalla memoria di massa a quella centrale.

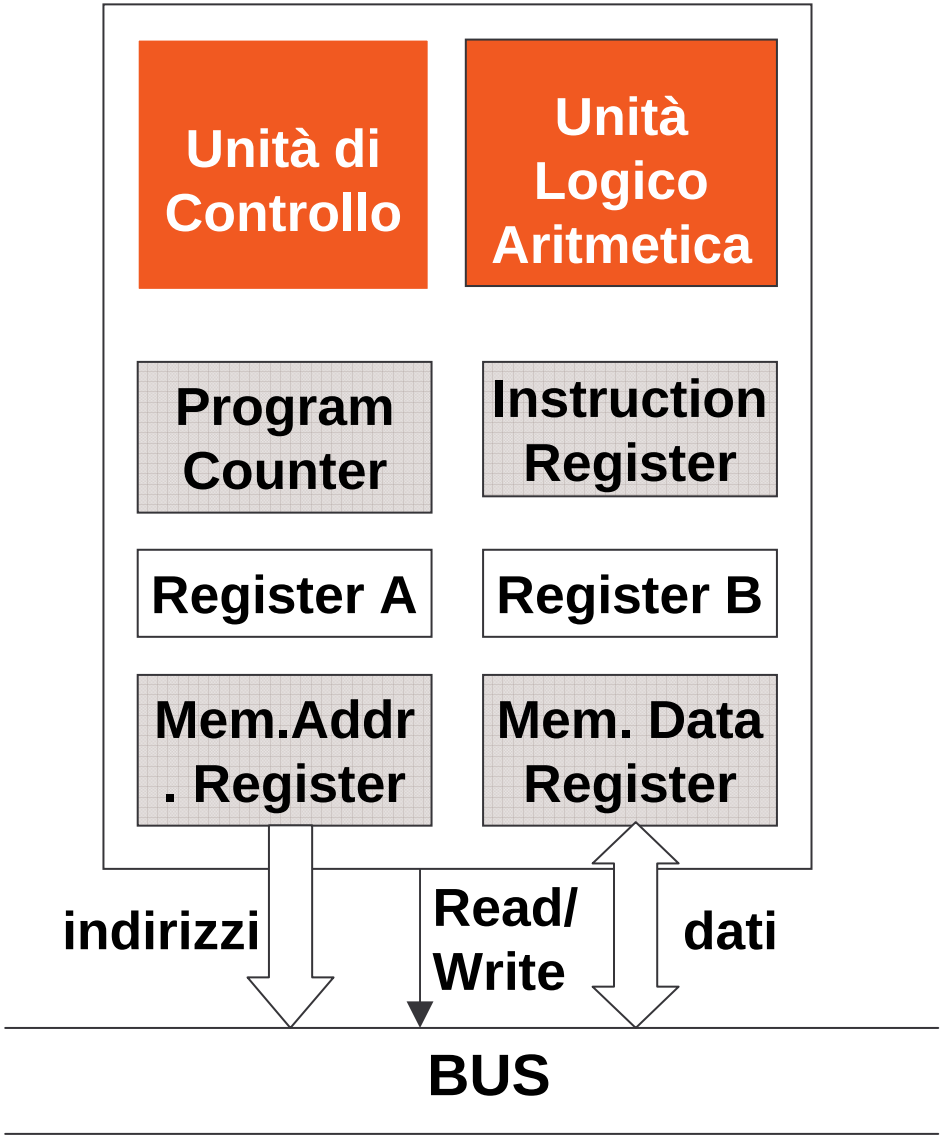
3. Modifiche alla struttura dei processori in modo da poter eseguire separatamente e in parallelo le varie fasi di un'istruzione (**pipelining**), eseguire più istruzioni dello stesso processo in contemporanea (**hyperthreading**) o costruire processori con due core fisici diversi (**dual core processor**)**.**

4. **Architetture multiprocessore** per costruire architetture dotate di più processori indipendenti.

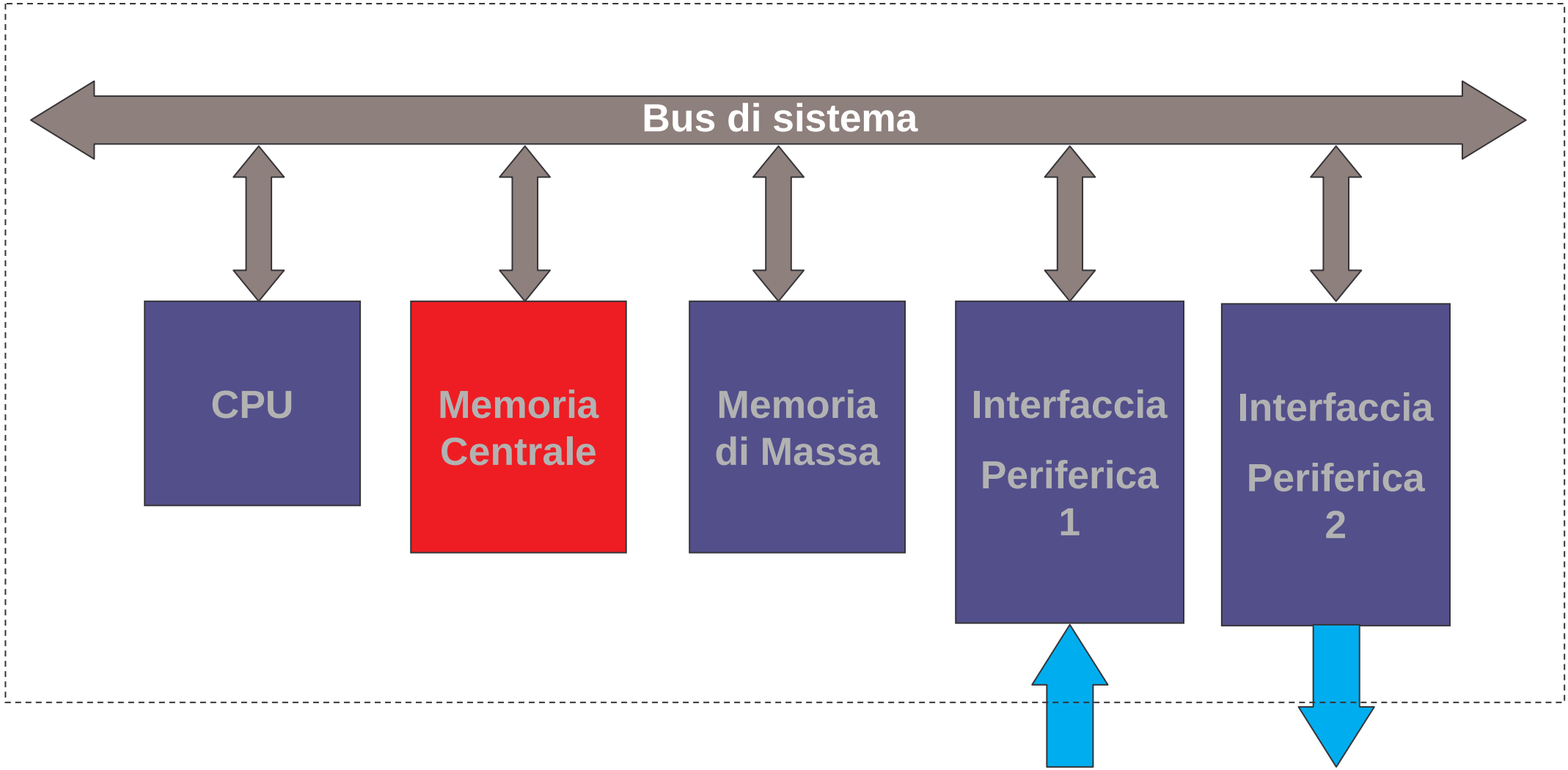
Connessione della CPU con il sistema

I vari componenti interni della CPU sono comunicanti tramite connessioni interne.

La CPU è connessa al resto del sistema tramite il BUS (linee indirizzi, dati e controllo).



Modello di von Neumann

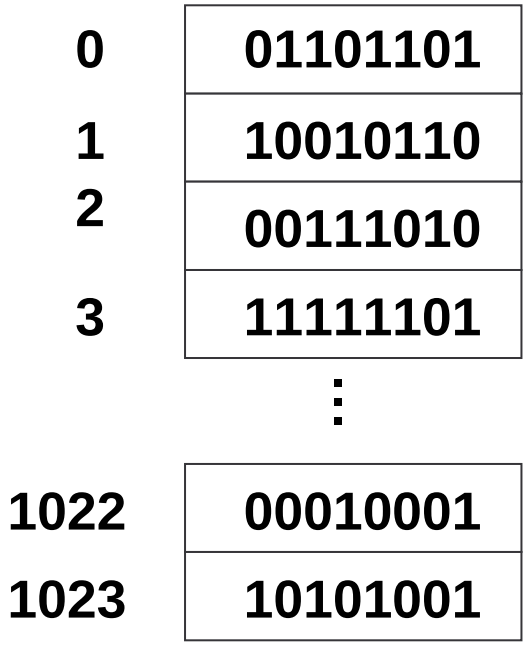


Organizzazione della memoria principale

La memoria principale è organizzata come un insieme di registri di uguale dimensione, ognuno dei quali è identificato tramite un numero progressivo ad esso associato, detto indirizzo che rappresenta la posizione di quel registro rispetto al primo registro.

Il contenuto dei registri non è immediatamente riconoscibile: non c'è distinzione esplicita tra istruzioni e dati e tra dati di tipo diverso.

Una istruzione o un dato possono risiedere su più registri consecutivi, se la dimensione del registro di memoria non è sufficiente.

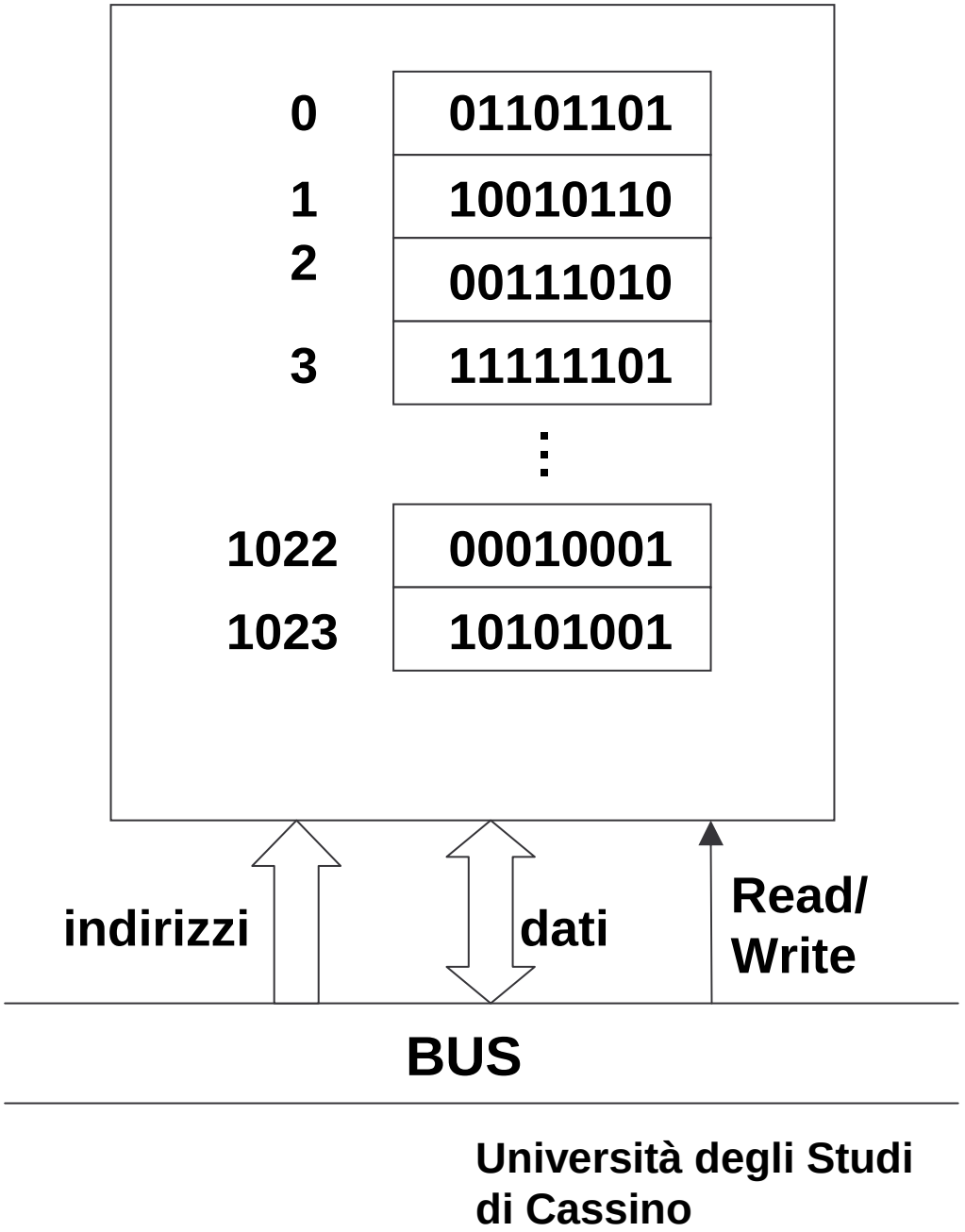


Organizzazione della memoria principale (2)

Il modulo di memoria principale è connesso al resto del sistema tramite il BUS.

In particolare, sono presenti tre gruppi di linee:

- **linee indirizzi**
- **linee dati**
- **linee Read/Write**

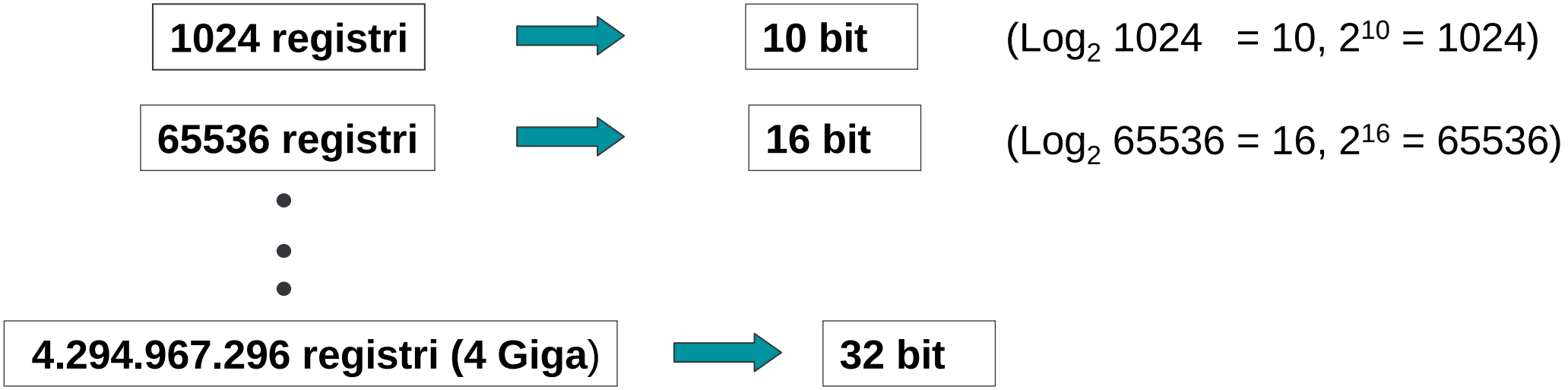


Quanti bit sono necessari per codificare un indirizzo ?

- Se la memoria contiene N registri abbiamo bisogno di N indirizzi diversi.
- Di quanti bit ho bisogno per rappresentare N indirizzi diversi?

Risposta: $\text{Log}_2 N$

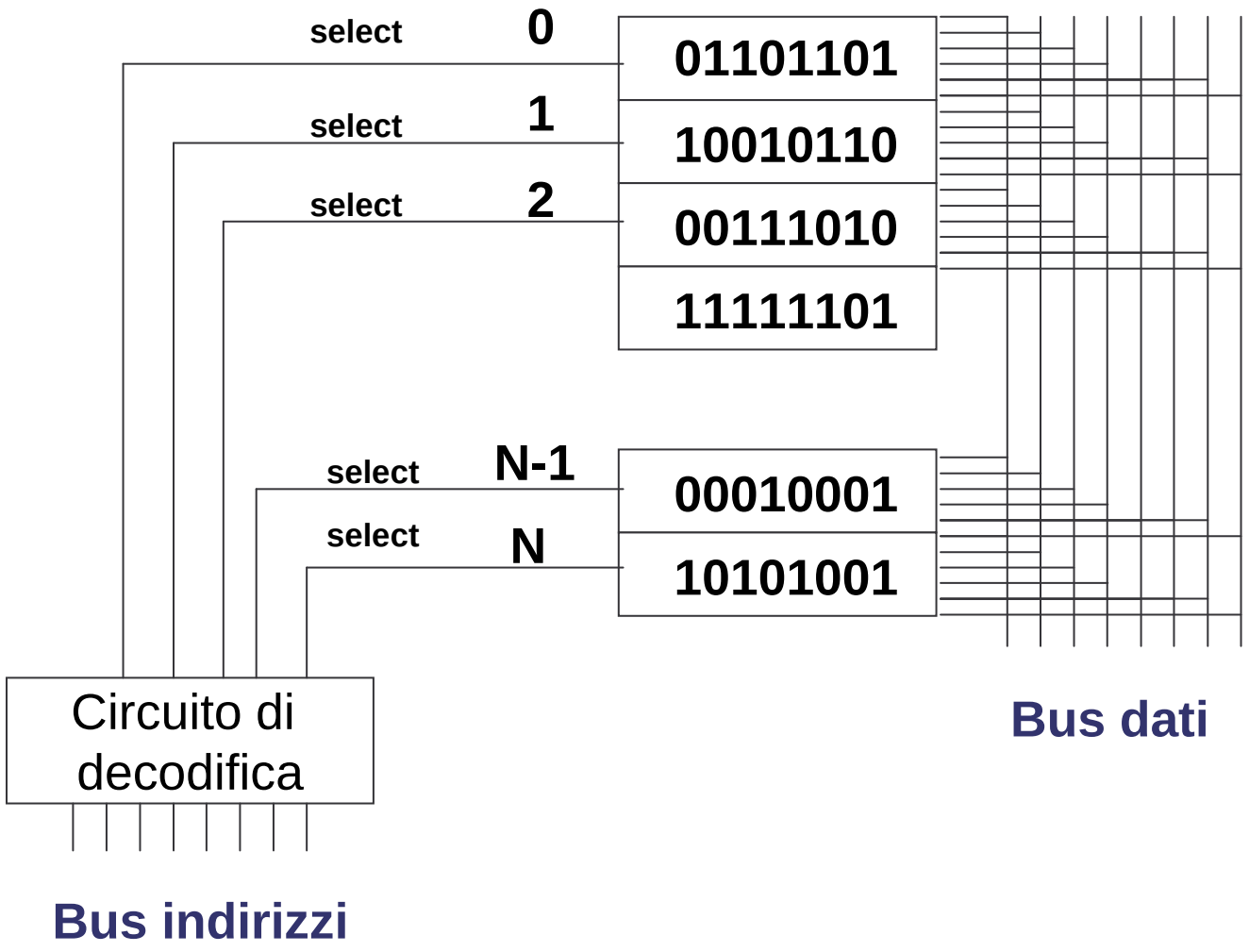
Es.



Decodifica degli indirizzi

Ogni singolo registro attiva le uscite quando viene attivato l'ingresso select.

Il circuito di decodifica è un circuito elettronico che ha $\log_2 N$ ingressi e N uscite.



Operazioni sulla memoria principale

Le operazioni possibili sul modulo di memoria principale sono orientate ai registri:

- scrittura di un valore in un registro
- lettura del valore di un registro

In ogni operazione è quindi necessario specificare:

- su quale registro si intende compiere l'operazione $\rightarrow$ indirizzo
- che tipo di operazione si intende realizzare $\rightarrow$ Read/Write
- in caso di scrittura, quale sia il valore da memorizzare

Parametri della memoria principale

Capacità

Fornisce una misura della quantità di informazione che è possibile memorizzare. Questa dipende dall'ampiezza dei singoli registri e dal numero di registri contenuti.
La capacità delle memoria si misura in termini di byte
(**Megabyte** = 2^{20} byte **Gigabyte** = 2^{30} byte **Terabyte** = 2^{40} byte)

Tempo di accesso

E' il tempo minimo che intercorre tra due operazioni (accessi) in memoria.
Dipende dalla tecnologia di realizzazione della memoria.
Si misura in termini di secondi (nanosecondi = 10^{-9} secondi).

Tipologie di memorie

Memorie RAM

Con le memorie viste finora si possono realizzare operazioni sia di lettura che di scrittura. Tali memorie si indicano come memorie **RAM** (*Random Access Memory*) ed hanno la caratteristica di mantenere il loro contenuto finché è presente l'alimentazione.

Esistono due tipi di memoria RAM:

RAM dinamica o DRAM (*Dynamic Random Access Memory)*

Alta densità di integrazione, economica, lenta, bassa potenza alimentazione

Dynamic: è necessario rigenerare i contenuti periodicamente (refresh)

RAM statica o SRAM (*Static Random Access Memory)*

Bassa densità di integrazione, costosa, veloce, alta potenza alimentazione

Static: il contenuto viene mantenuto finché è presente l'alimentazione

Tipologie di memorie (2)

Memorie ROM

All'interno del calcolatore, alcuni programmi e dati (es. i programmi per l'avvio all'accensione) devono rimanere memorizzati anche quando l'alimentazione viene a mancare. Questi sono, inoltre, programmi e dati che, una volta memorizzati, non devono essere più modificati.

Per questo tipo di esigenze si utilizzano memorie **ROM** (*Read Only Memory*), i cui contenuti sono inseriti una volta per sempre all'atto della loro costruzione e non possono più essere modificati o cancellati.

Oggi esistono delle ROM che sono programmabili **EEPROM** (**E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory)

Organizzazione del Sistema di Memoria

Requisiti ideali di un sistema di memoria:

capacità infinita

velocità infinita

Evidenza:

→ le memorie capienti ed economiche (DRAM) sono lente

→ le memorie veloci (SRAM) sono costose e meno integrabili

Come realizzare un sistema di memoria che sia capiente, economico e veloce?

Un sistema basato su una gerarchia di memoria

La memoria cache

Il sistema di memoria è composto da moduli di memoria con caratteristiche diverse e organizzati a livelli.

Tra CPU e memoria principale viene posto un modulo di memoria intermedio (**cache**), ad accesso veloce, ma di capienza limitata.

I dati memorizzati sono distribuiti sui vari moduli e possono essere trasferiti tra moduli adiacenti.

La distribuzione è realizzata in maniera da cercare di memorizzare i dati e le istruzioni richiesti più frequentemente nella cache, in modo che la CPU possa accedervi velocemente.

Caratteristiche della cache

- Il tempo di propagazione del segnale (è un vincolo per il tempo di accesso) è minore.
- Una memoria grande ha bisogno di indirizzi grandi (maggiori tempi di decodifica).

Il miglioramento delle prestazioni dovuto alla memoria cache si basa sul **principio di località del riferimento**:

i dati usati più di recente saranno utilizzati ancora nel prossimo futuro

Sistema di memoria in un calcolatore attuale