

**Università degli Studi di Cassino**  
**Corso di Laurea in**  
**Ingegneria delle Telecomunicazioni**

**Corso di Calcolatori Elettronici I**  
(fino al 23/11)

**Corso di Calcolatori Elettronici**

**F.Tortorella**

*F. Tortorella*

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

**Contenuti del corso**

**Calcolatori  
Elettronici**

- Modello di programmazione del processore
- Programmazione in linguaggio assembly
- Elementi di progettazione logica dei circuiti
- Aritmetica e circuiti per l'aritmetica
- Architettura dell'unità centrale: data path e controllo
- Pipelining
- Gerarchia di memoria
- Sottosistema di I/O

**Calc. El. I**

*F. Tortorella*

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## **Testo adottato:**

Patterson, Hennessy

### **Struttura, organizzazione e progetto dei calcolatori**

2a edizione

Jackson Libri

## **Testi di consultazione:**

Hamacher, Vranesic, Zaky

Introduzione all'Architettura  
dei Calcolatori

McGraw-Hill Italia

Hennessy, Patterson

Architettura dei computer:

Un approccio quantitativo

2a edizione

Jackson Libri

*F. Tortorella*

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## **...chi siamo, da dove veniamo ?**

### **Fondamenti di Informatica I:**

- Strutture dati
- Costrutti di programmazione
- Algoritmi fondamentali

**C / C++**

### **Fondamenti di Informatica II:**

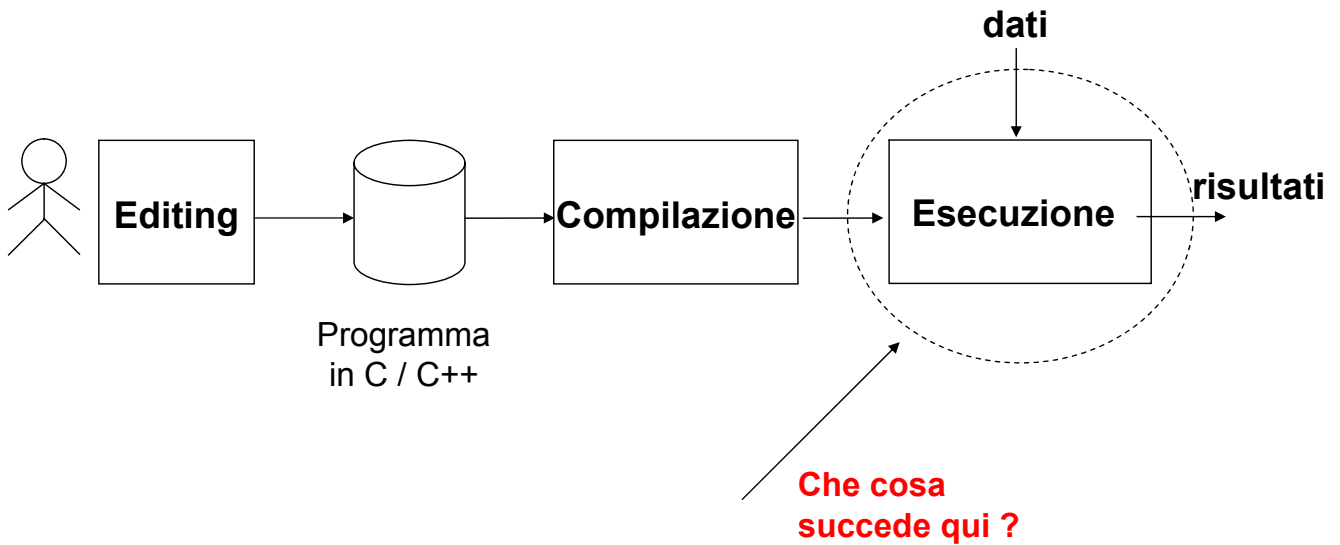
- Strutture dati complesse
- Ricorsione

*F. Tortorella*

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

# Una tipica sessione di lavoro

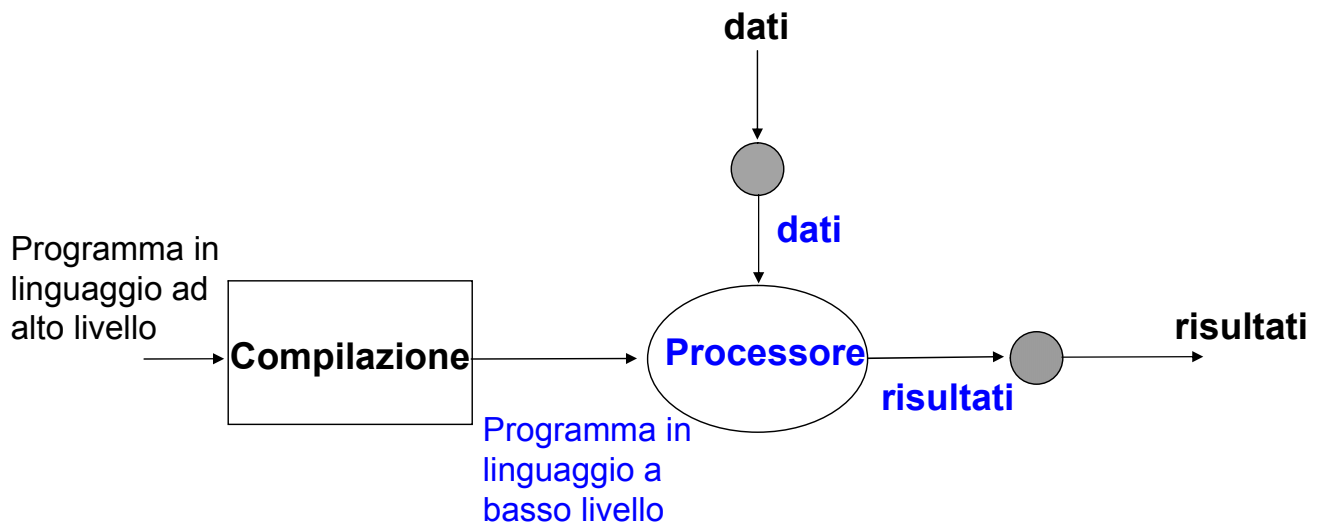


F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi di Cassino

# Che cosa succede sotto il coperchio ?

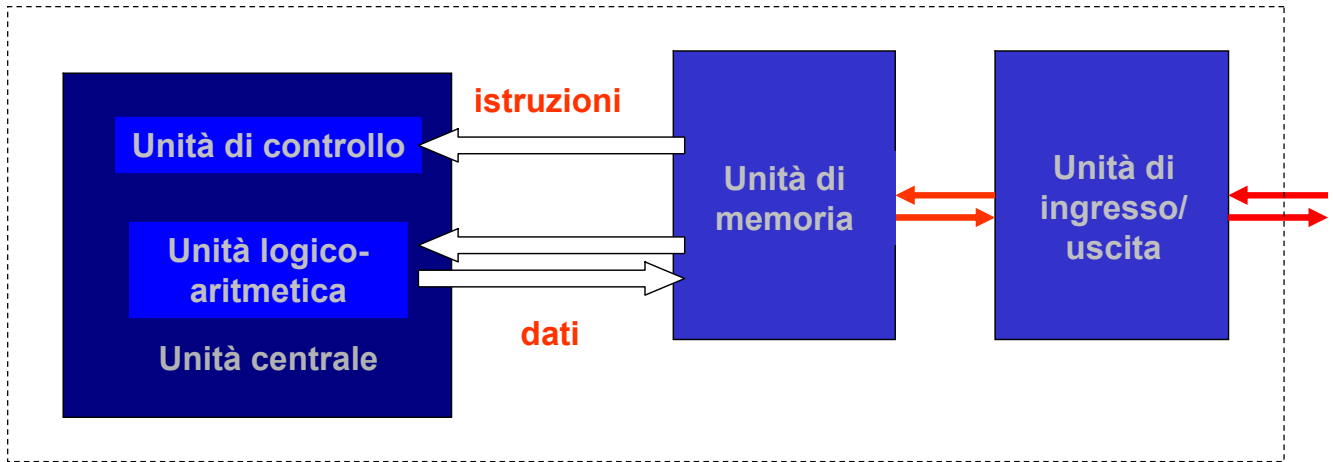


F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi di Cassino

## Organizzazione del calcolatore



Modello logico

singole componenti

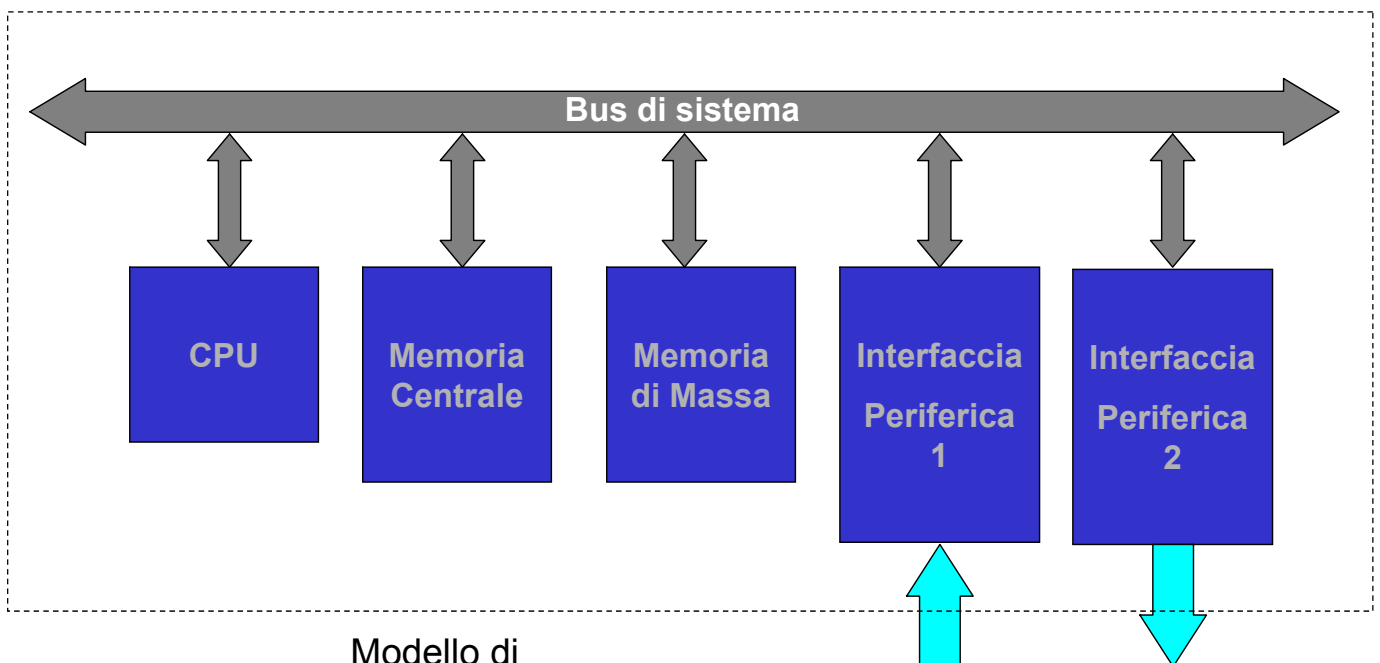
flussi di dati e istruzioni

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## Modello di von Neumann



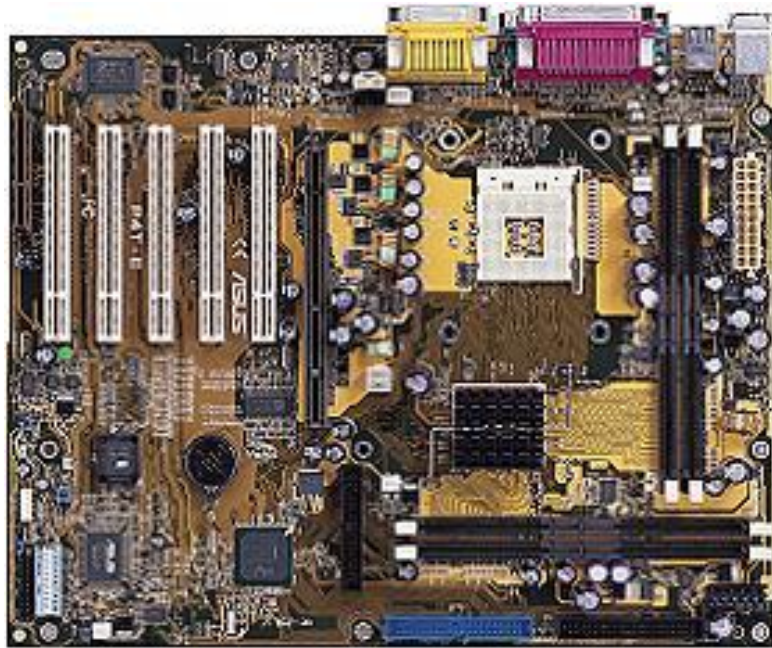
Modello di  
implementazione

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## ASUS P4T-E



F. Tortorella

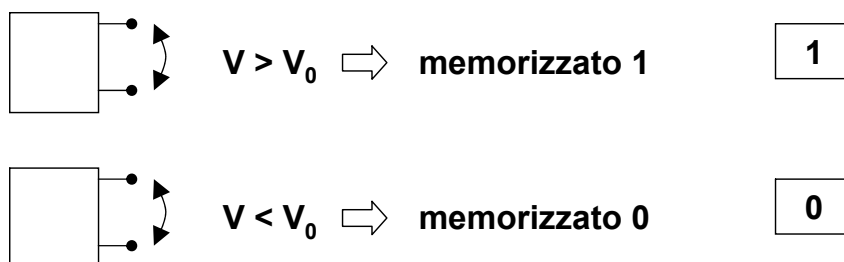
Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

### La memorizzazione dei dati e delle istruzioni

La più piccola unità di informazione memorizzabile (e quindi utilizzabile) è il **bit**, 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**).



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## 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.

### 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	2 bit → 4 stati
11	011	0011	3 bit → 8 stati
	100	0100	4 bit → 16 stati
	101	0101	...
	110	0110	
	111	0111	
		1000	
		1001	
		1010	
		1011	
		1100	
		1101	
		1110	
		1111	

## 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

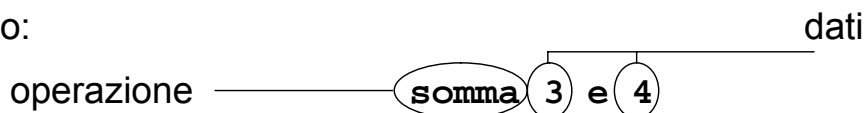
**La codifica implica una rappresentazione dei dati limitata e discreta**

## 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 ?

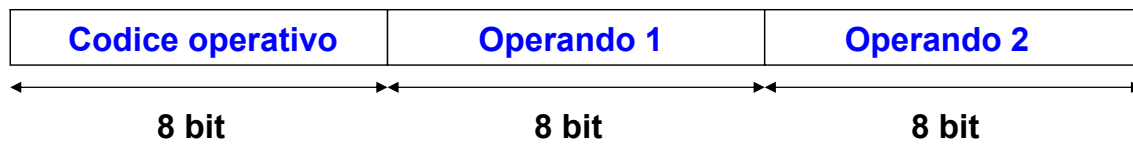
L'insieme delle diverse operazioni che l'unità centrale è in grado di eseguire è finito e quindi è possibile codificarlo con un certo numero di bit (**codice operativo**).

somma	0000
sottrai	0001
moltiplica	0010
dividi	0011
...	...

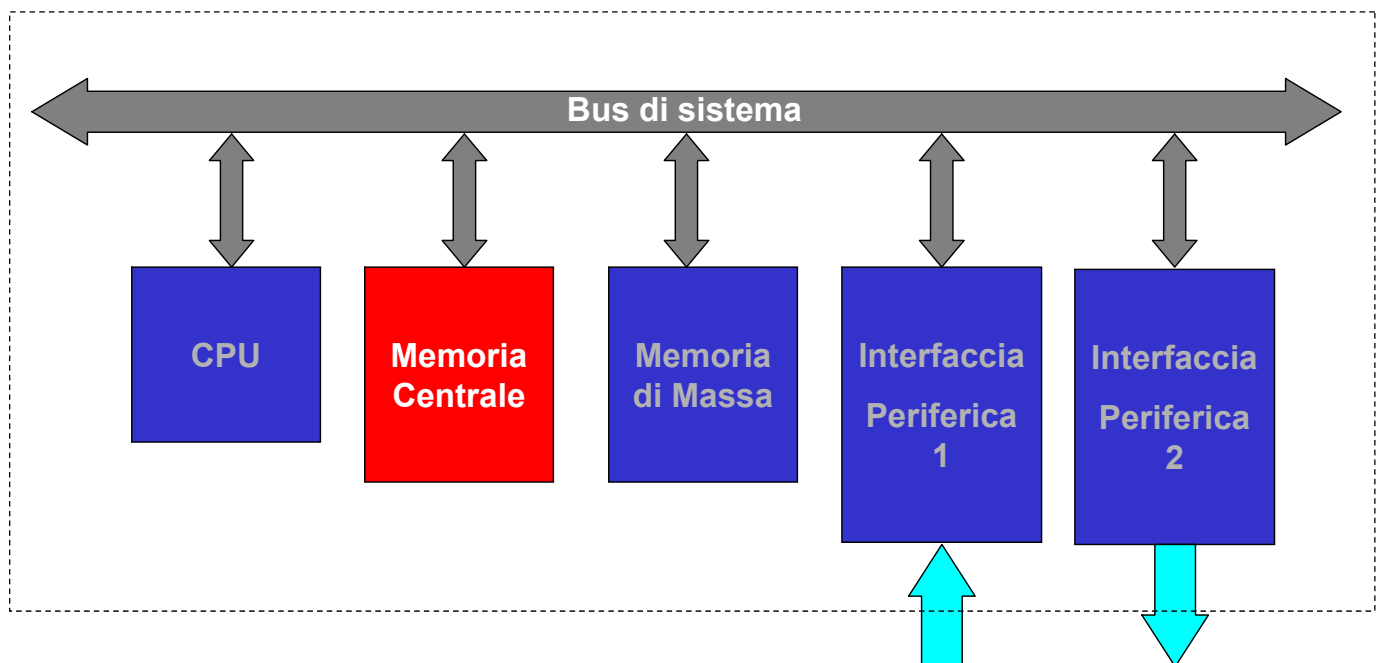


Una istruzione sarà quindi rappresentabile da una sequenza di bit divisa in due parti:

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



### Modello di von Neumann



## 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.

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.

Il parallelismo di accesso è definito dall'ampiezza del registro

0	01101101
1	10010110
2	00111010
3	11111101
	⋮
1022	00010001
1023	10101001

Quanti bit sono necessari per codificare un indirizzo ?

F. Tortorella

Corso di Calcolatori Elettronici

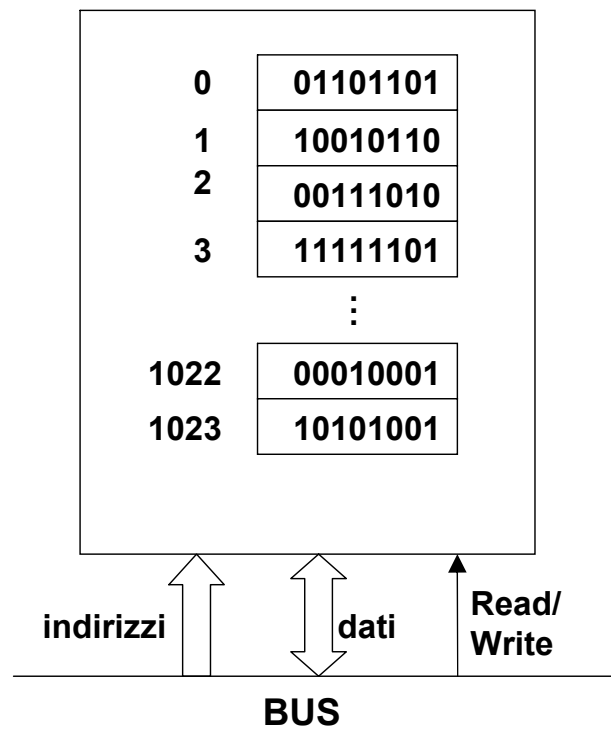
Università degli Studi di Cassino

## 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



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi di Cassino

## 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 → indirizzo
- che tipo di operazione si intende realizzare → 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 della memoria si misura in termini di byte  
(Megabyte =  $2^{20}$  byte Gigabyte =  $2^{30}$  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

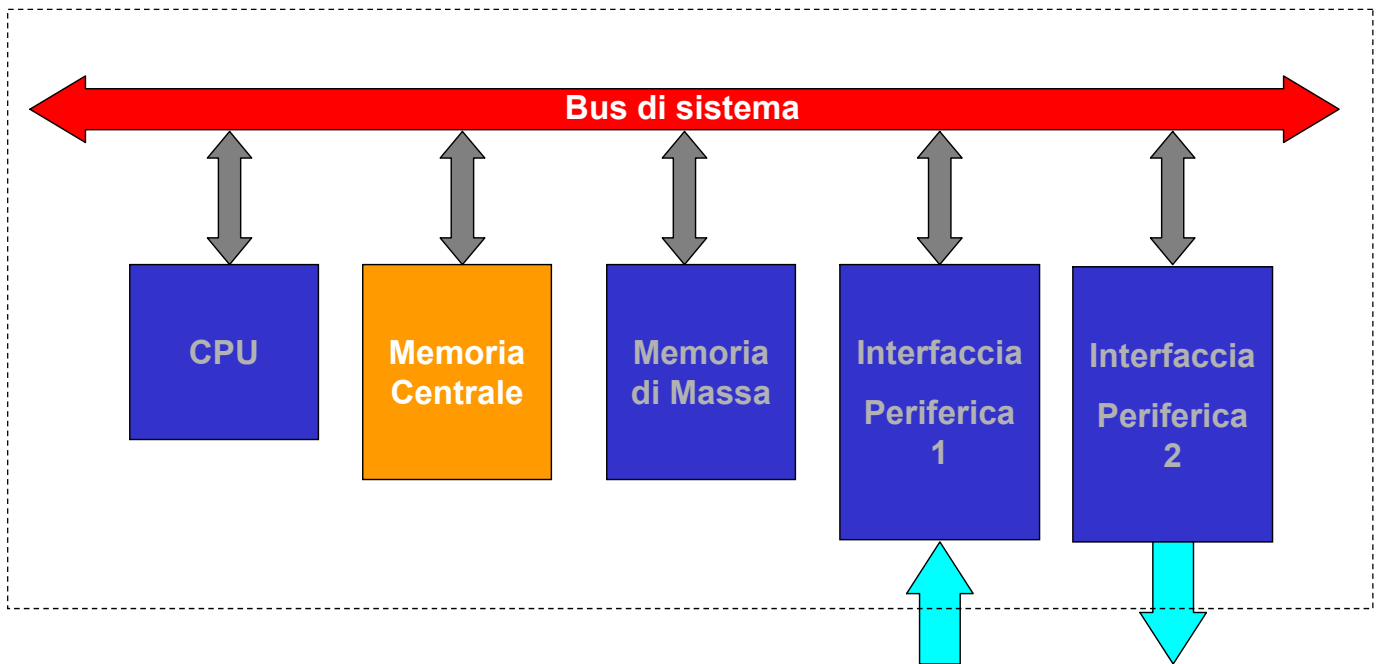
## Tipologie di memorie (2)

### 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.

## Modello di von Neumann



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

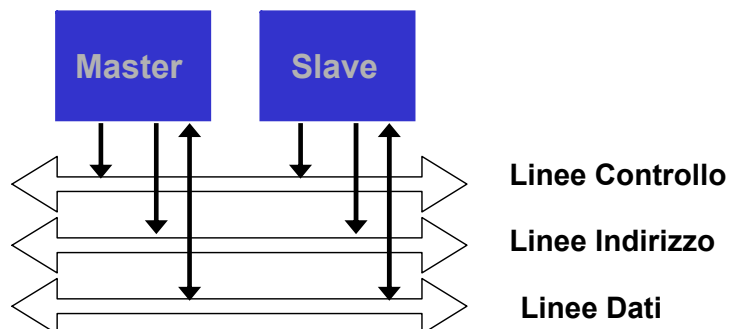
## Il bus

Forma un canale di comunicazione tra le varie unità del calcolatore.

Tipicamente è possibile un solo colloquio alla volta tra due unità: un **master**, che ha la capacità di controllare il bus ed inizia la comunicazione, ed uno **slave**, che viene attivato dal master.

Il bus è formato da un insieme di linee su cui viaggiano i segnali. Le linee si dividono in

- linee dati
- linee indirizzi
- linee controllo

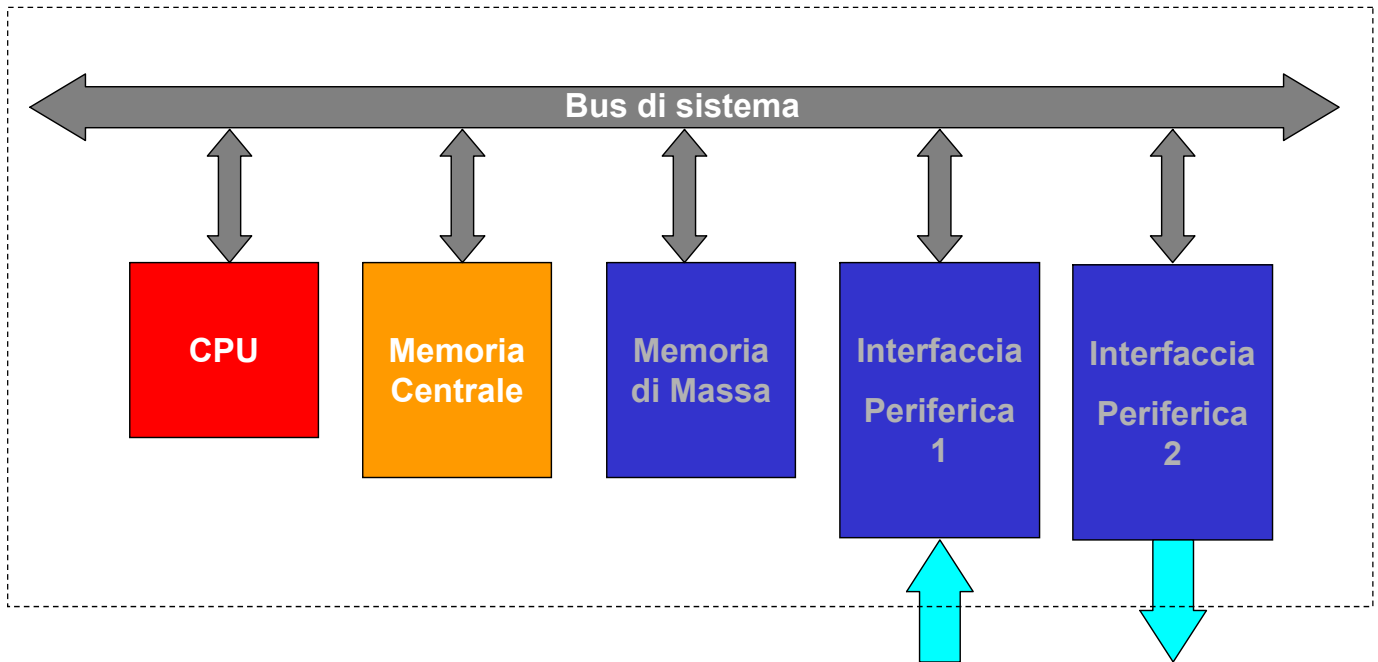


F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## Modello di von Neumann



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

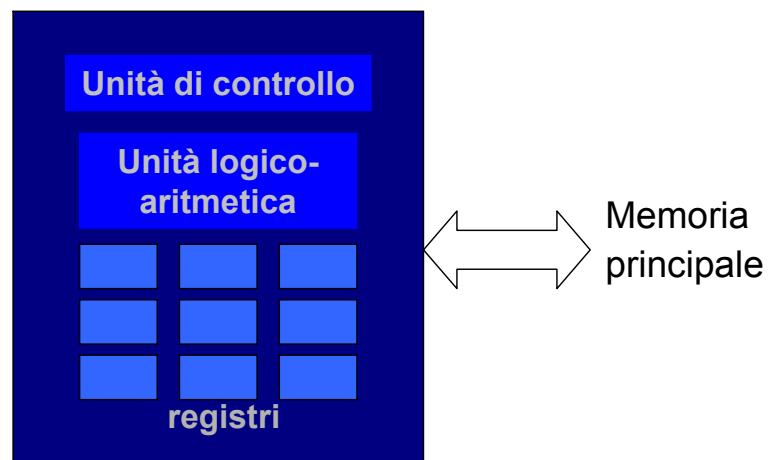
## CPU (Central Processing Unit)

### 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)

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## 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

L'esecuzione di una istruzione avviene attraverso alcune fasi:

Ciclo del processore

### 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

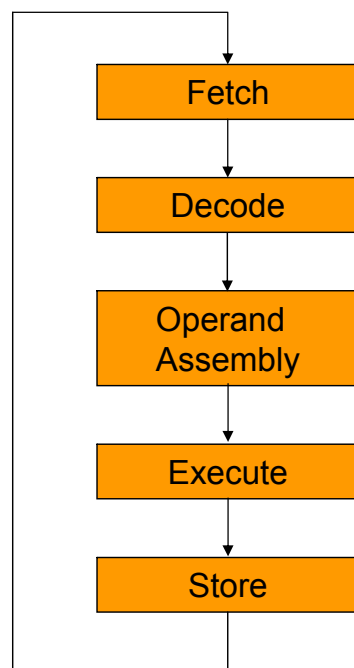
Viene memorizzato il risultato dell'operazione prevista dalla istruzione

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## 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

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

## L'Unità Logico Aritmetica

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)
- Istruction Register (IR)

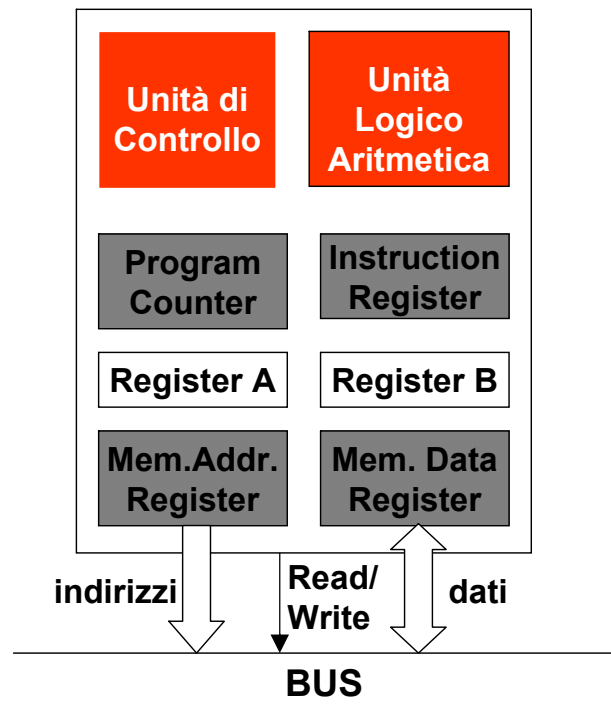
**I registri speciali non sono  
accessibili dalle istruzioni**



## 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).



## Trasferimento CPU-memoria

Qualunque sia il trasferimento da realizzare, la CPU (master) deve precisare l'indirizzo del dato da trasferire.

In queste operazioni, la memoria è comunque uno slave e "subisce" l'iniziativa della CPU, ricevendo da questa l'indirizzo del dato da trasferire e l'informazione sull'operazione da realizzare (lettura o scrittura)

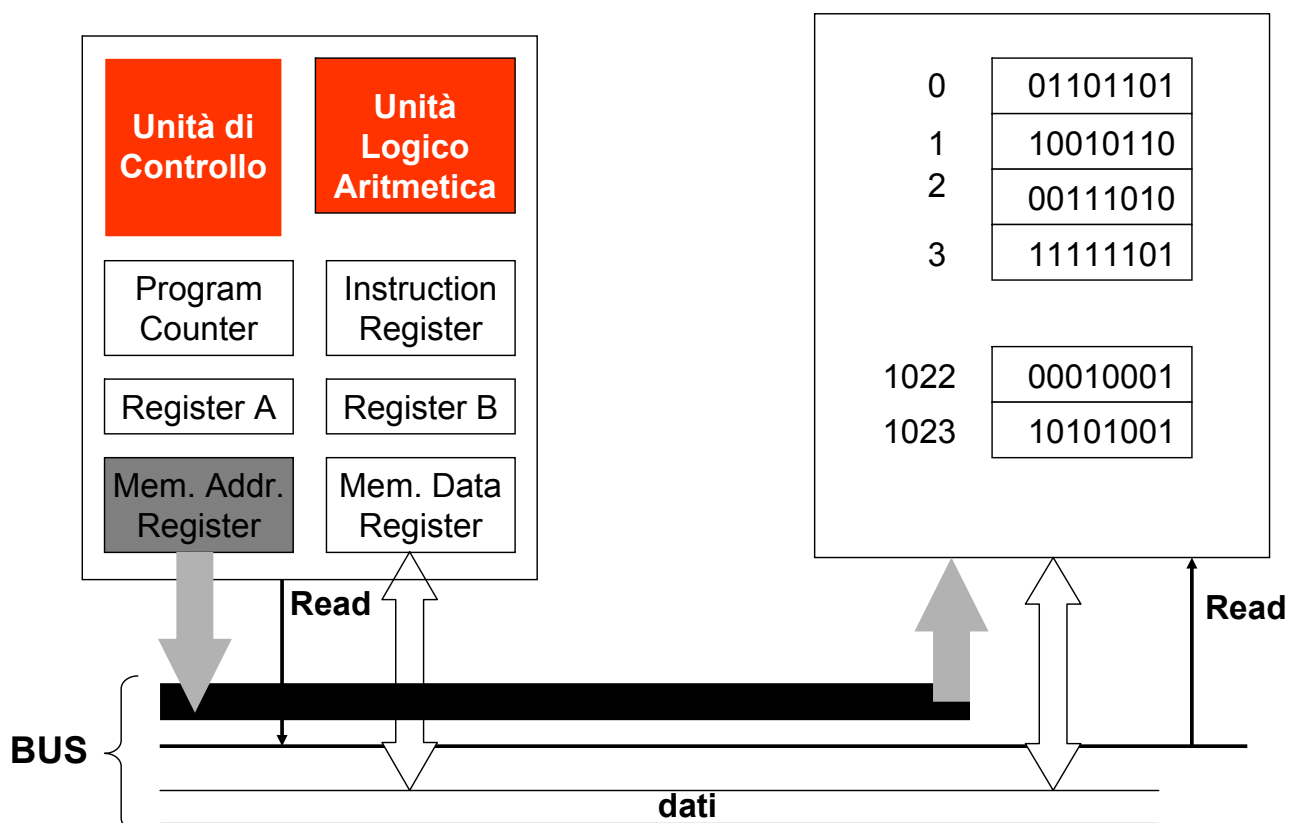
## Trasferimento memoria → CPU (lettura)

- 1) la CPU scrive l'indirizzo del dato da trasferire sul MAR che lo propagherà alle linee indirizzi del bus. Contemporaneamente, segnala sulle linee di controllo che si tratta di una lettura.
- 2) la memoria riceve, tramite il bus, l'indirizzo e l'indicazione dell'operazione da effettuare. Copia il dato dal registro individuato sulle linee dati del bus.
- 3) il dato richiesto, tramite le linee dati del bus, arriva al MDR della CPU. Da qui sarà spostato verso gli altri registri interni.

F. Tortorella

Corso di Calcolatori Elettronici

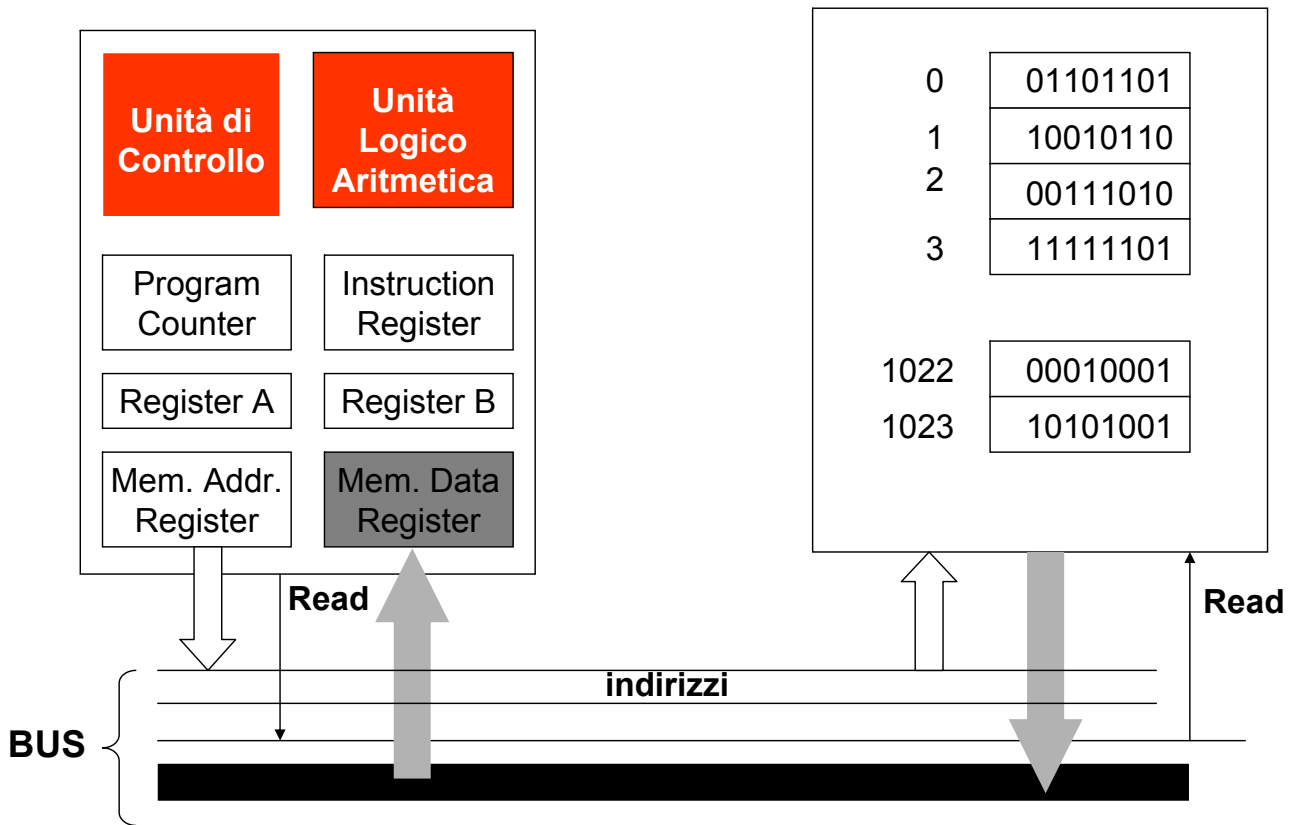
Università degli Studi  
di Cassino



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

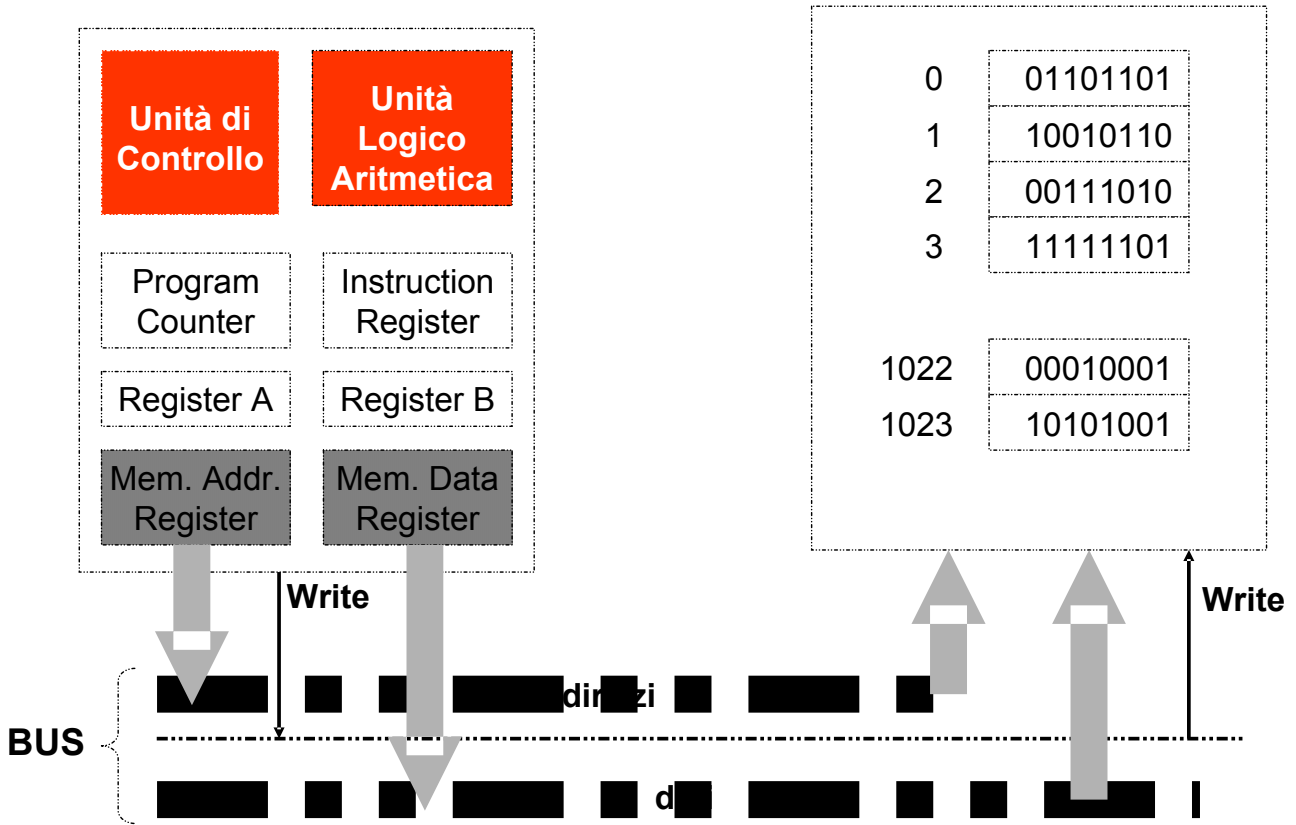
## Trasferimento CPU → memoria (scrittura)

- 1) la CPU scrive l'indirizzo del dato da trasferire sul MAR, mentre il dato viene copiato sul MDR. Il contenuto dei due registri viene propagato sulle linee indirizzi e dati del bus. Contemporaneamente, la CPU segnala sulle linee di controllo che si tratta di una scrittura.
- 2) la memoria riceve, tramite il bus, l'indirizzo, il dato e l'indicazione dell'operazione da effettuare. Copia il dato dalle linee dati del bus al registro individuato dall'indirizzo.

F. Tortorella

Corso di Calcolatori Elettronici

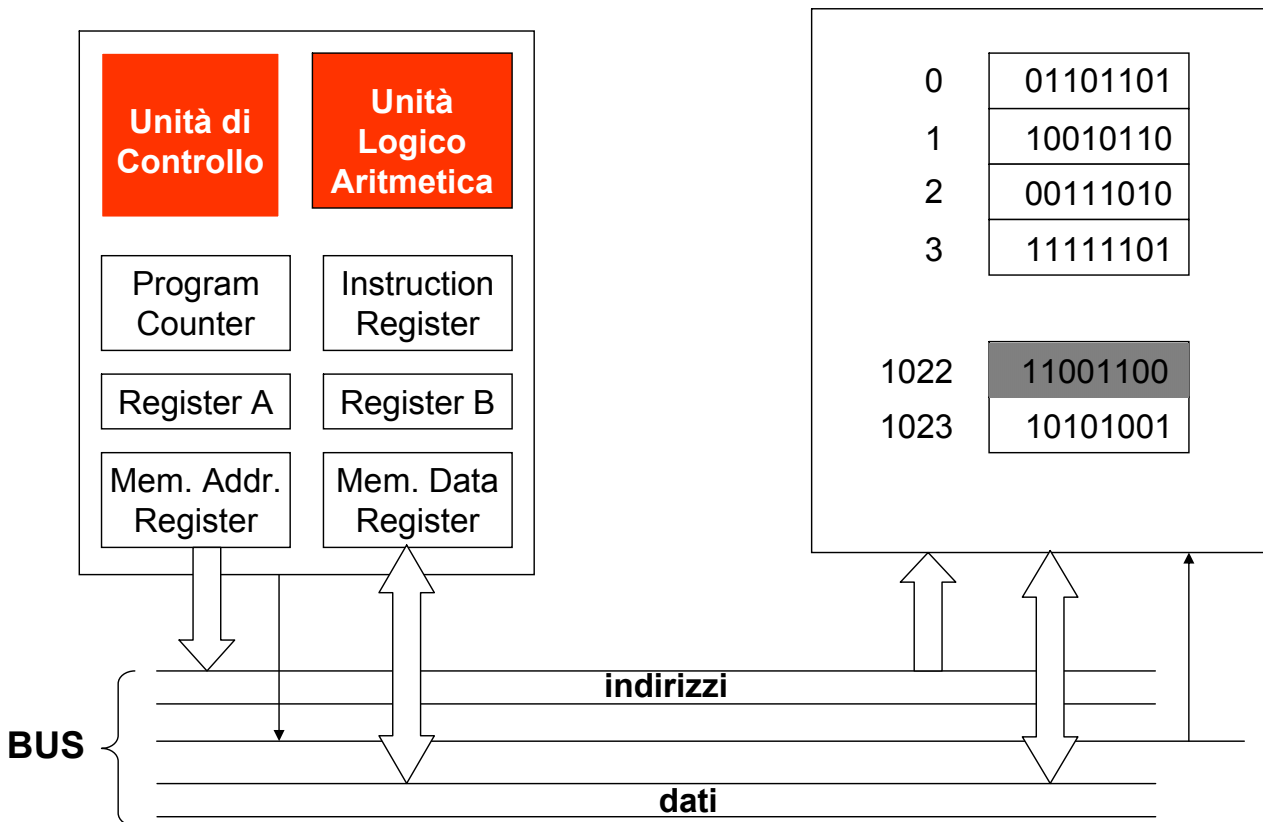
Università degli Studi  
di Cassino



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi di Cassino



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi di Cassino

## Esempio di esecuzione di una istruzione

Consideriamo un'istruzione del tipo:

ADD (1021),(1022),1023

Il cui significato è:

“somma i valori che trovi nei registri di memoria di indirizzo 1021 e di indirizzo 1022 e memorizza il risultato nel registro di indirizzo 1023”.

Supponiamo inoltre che l'istruzione si trovi memorizzata nel registro di memoria di indirizzo 3.

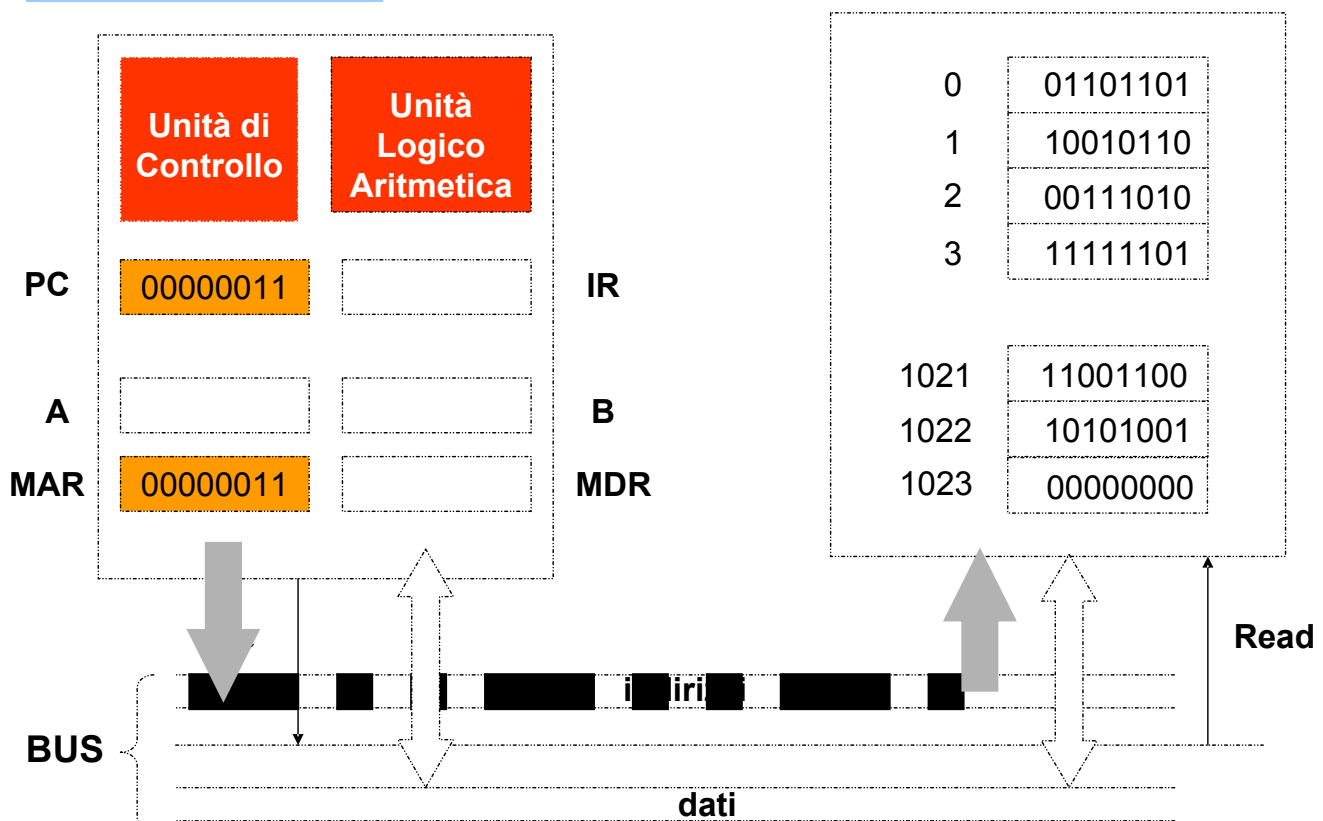
Consideriamo le varie fasi necessarie per l'esecuzione di questa istruzione...

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

### Fase FETCH



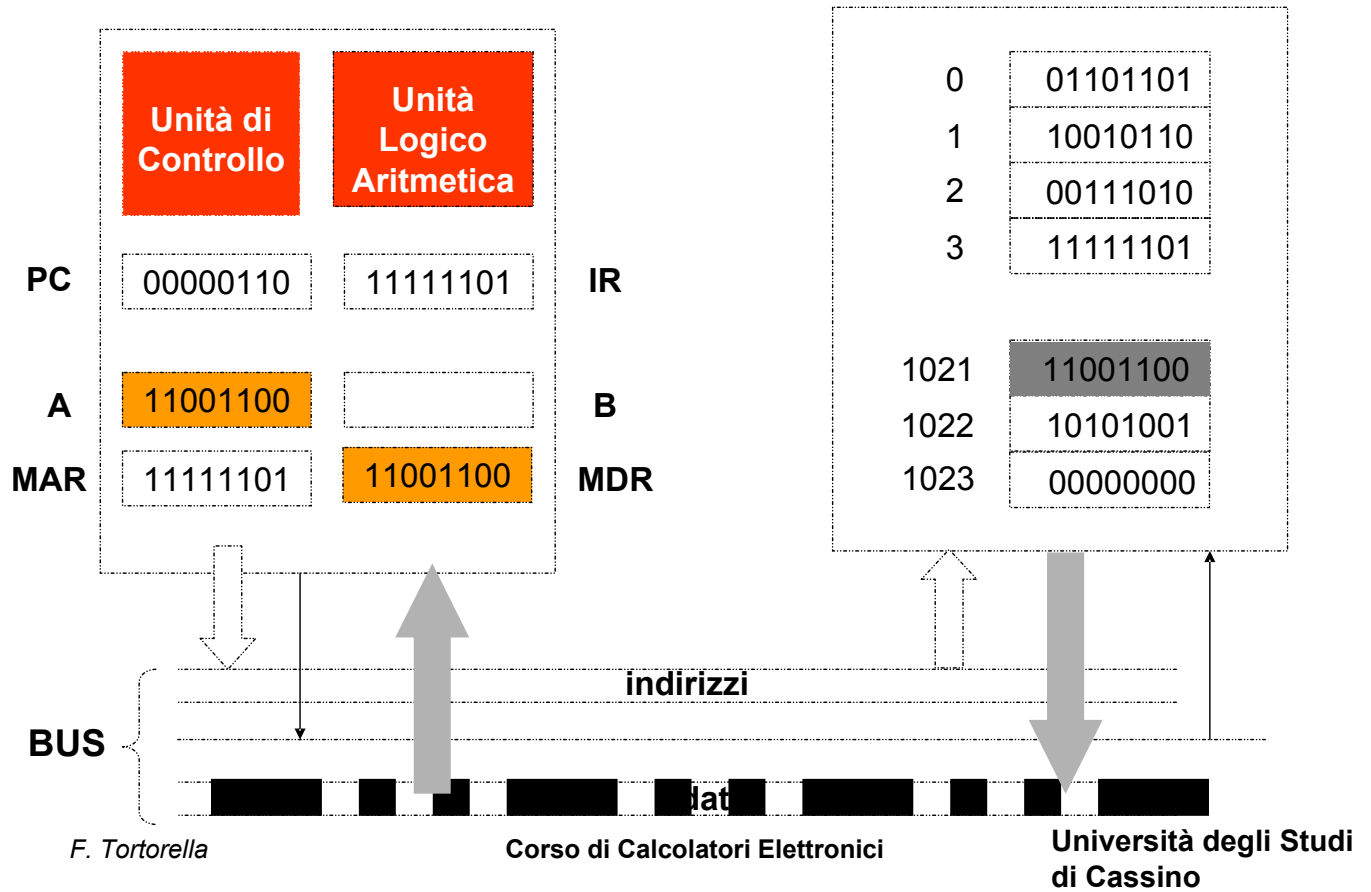
F. Tortorella

Corso di Calcolatori Elettronici

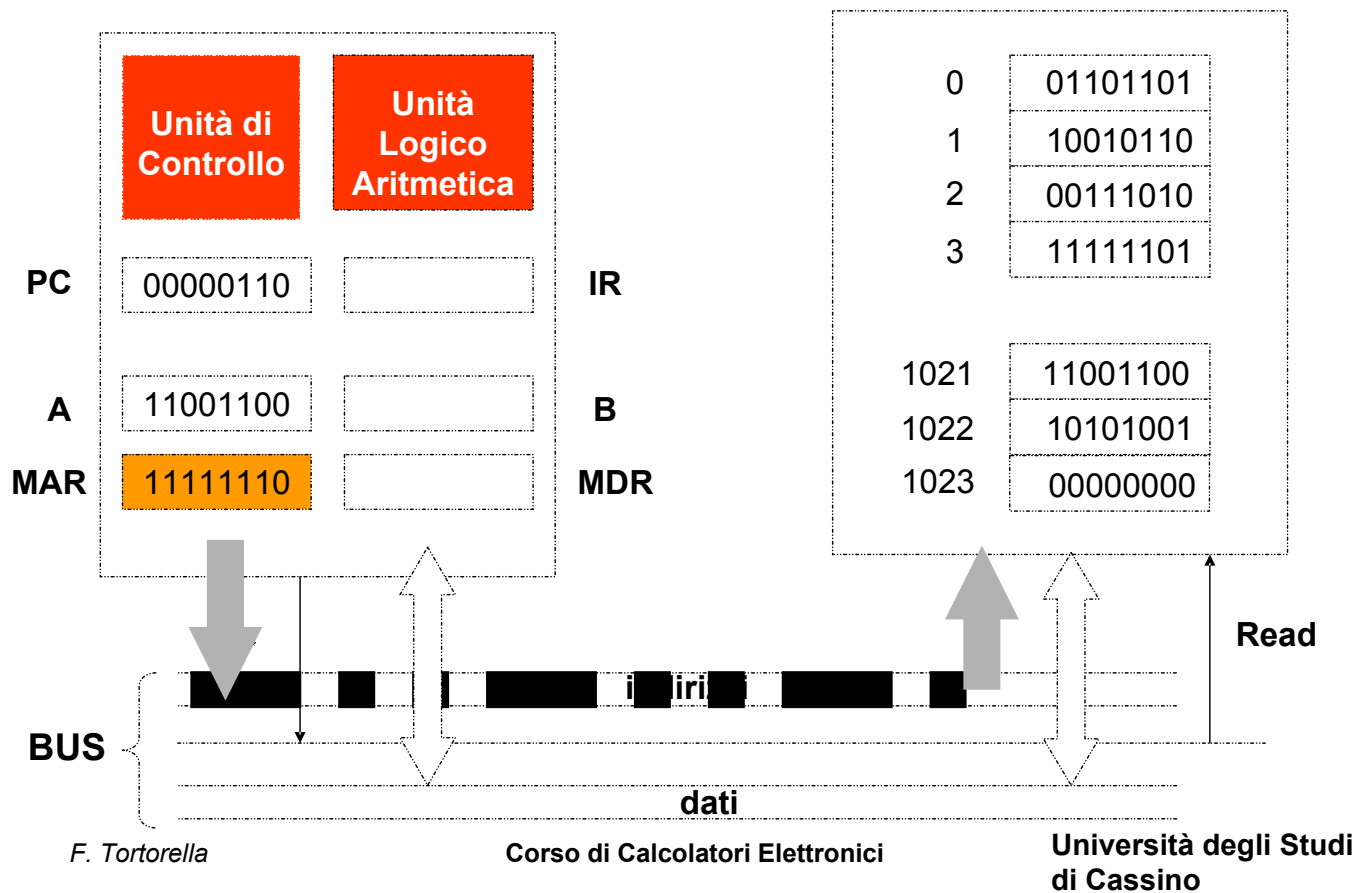
Università degli Studi  
di Cassino



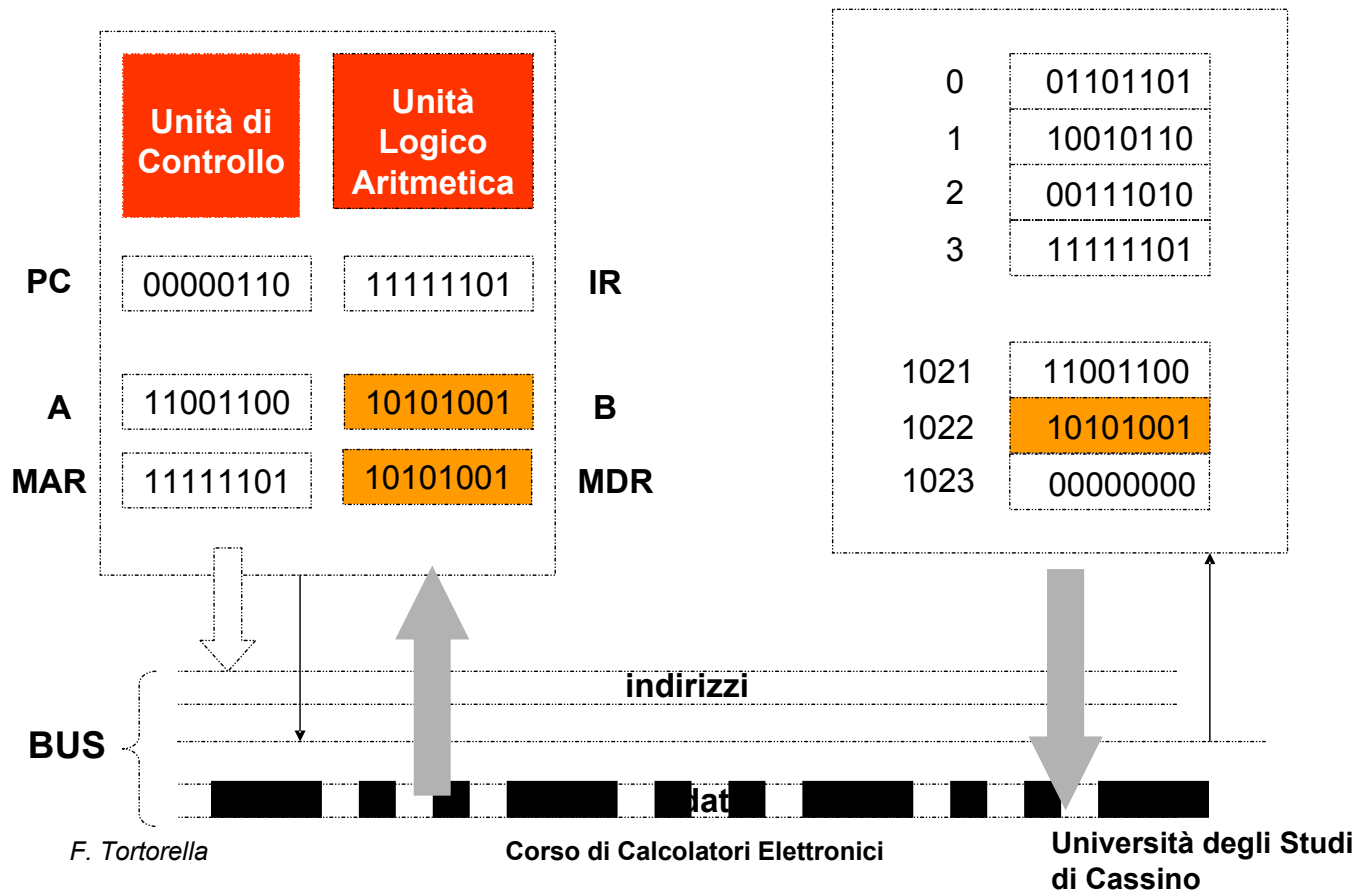
## Fase OPERAND ASSEMBLY



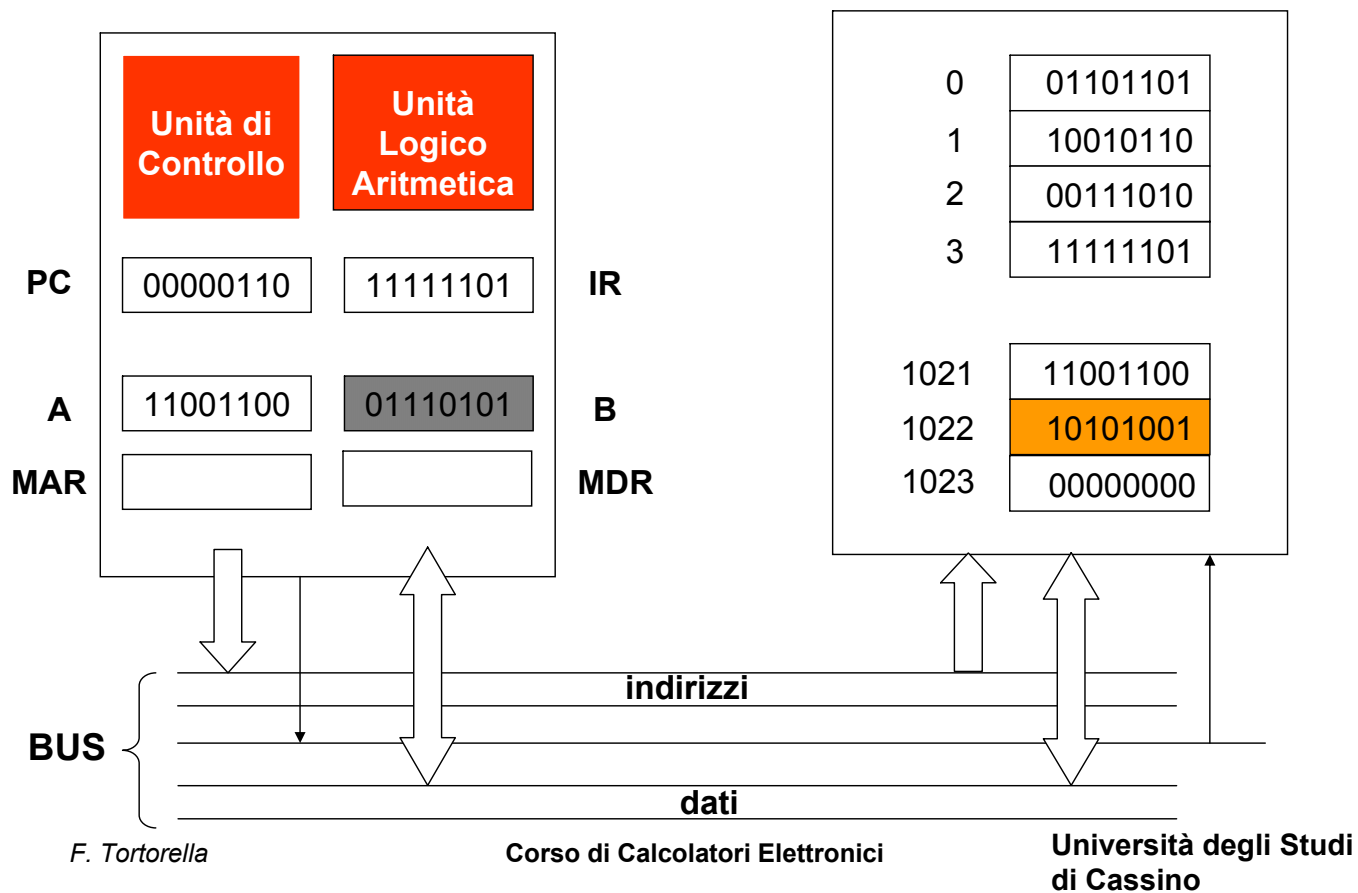
## Fase OPERAND ASSEMBLY



## Fase OPERAND ASSEMBLY

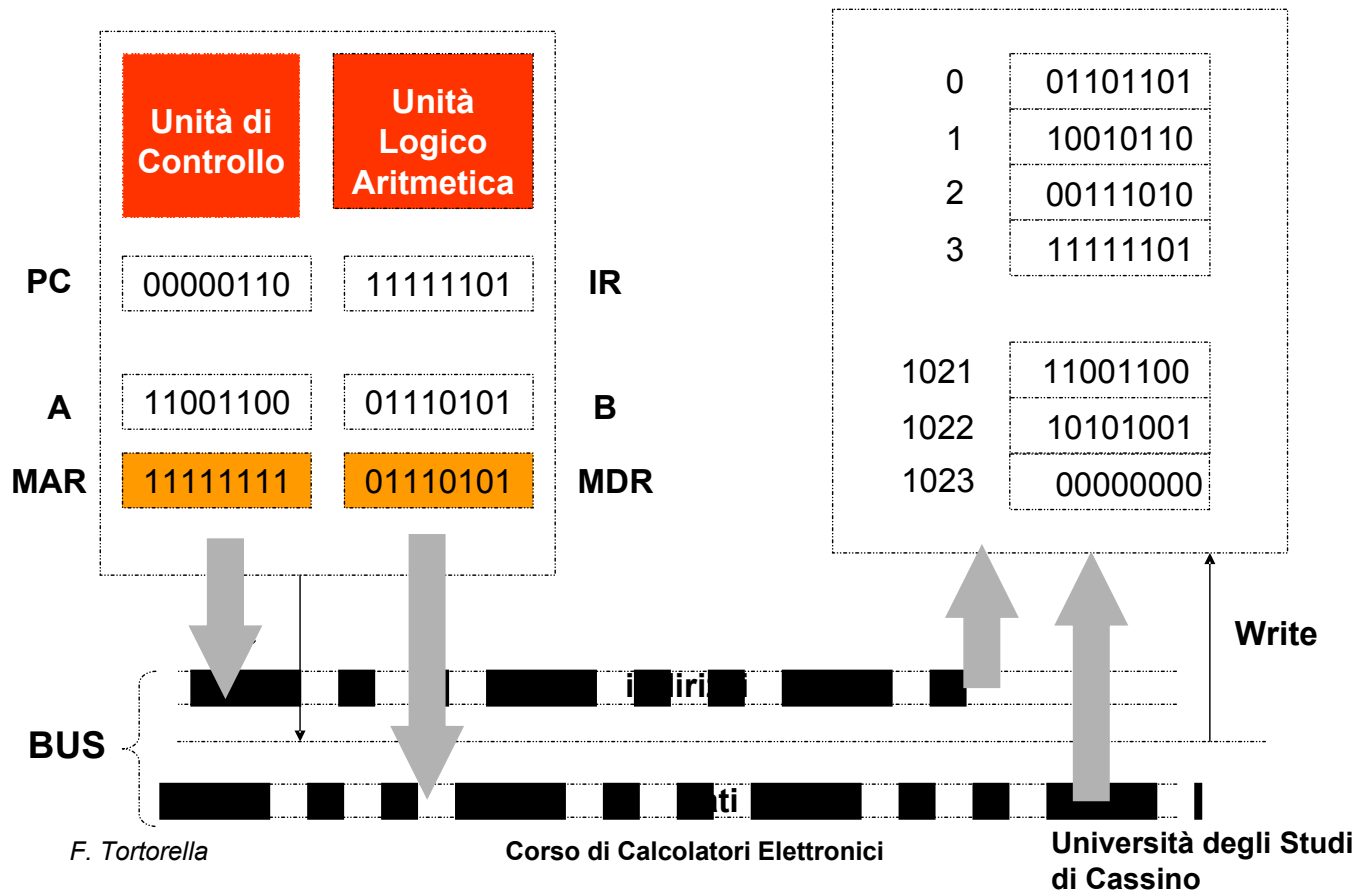


## Fase EXECUTE

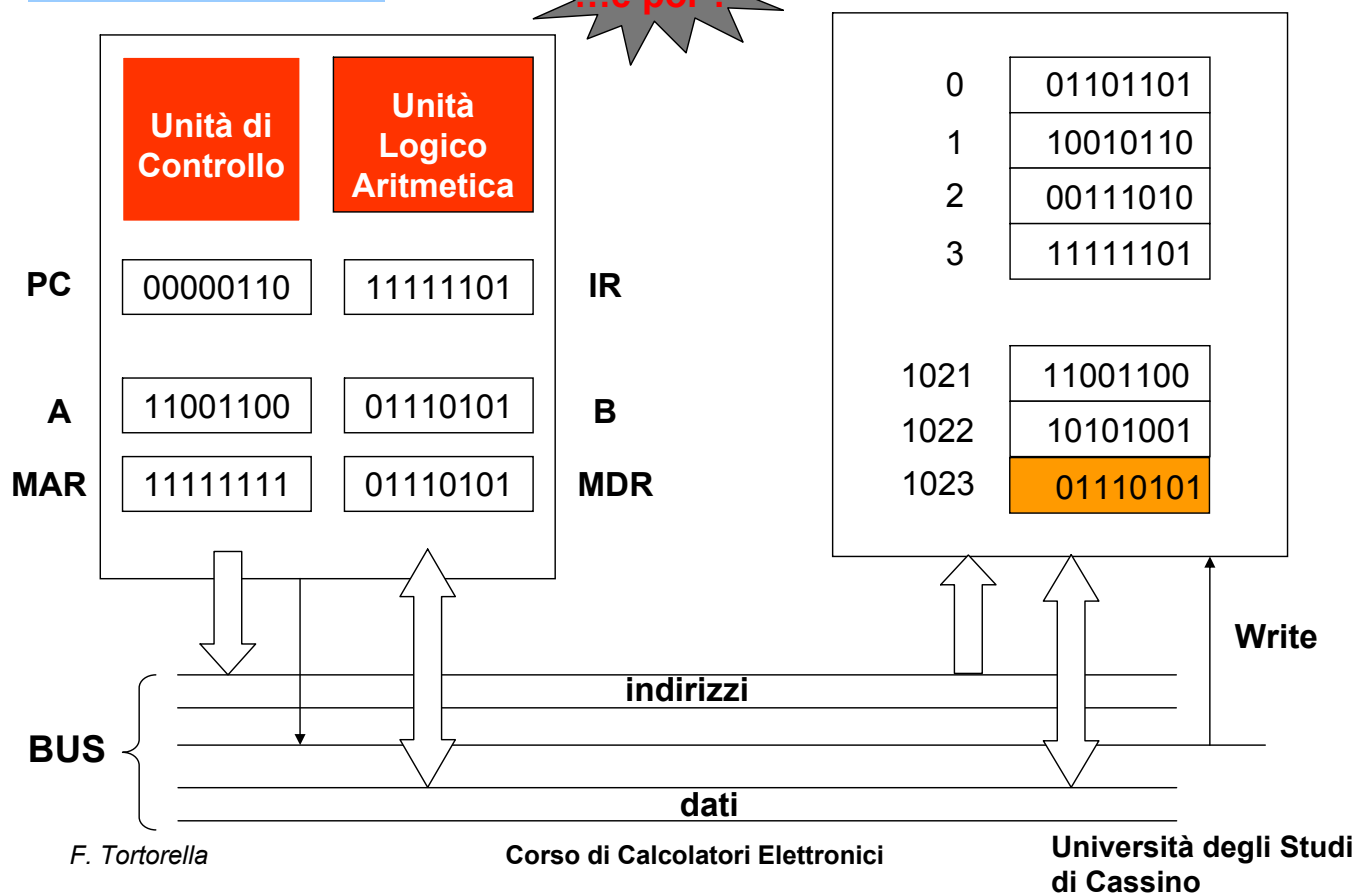




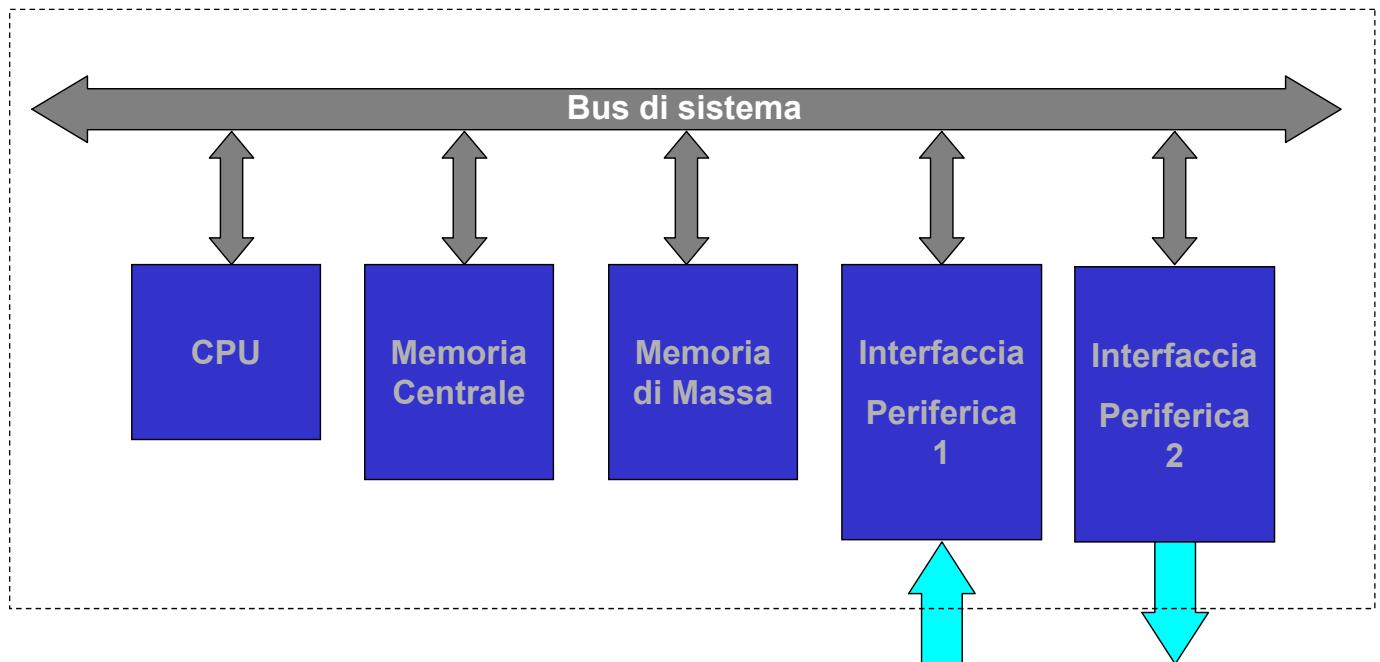
## Fase STORE



## Fase STORE



# Modello di von Neumann



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

Un calcolatore basato sul modello di von Neumann permette l'esecuzione di un *programma*, cioè di una sequenza di istruzioni descritte nel linguaggio interpretabile dal calcolatore.  
Ma quali sono le caratteristiche di tale linguaggio ?

- è codificato tramite sequenze di bit
- accede ai dati tramite gli indirizzi di memoria o i registri interni della CPU
- ogni istruzione può compiere solo azioni molto semplici
- non gestisce direttamente i tipi di dati di interesse
- è strettamente legato alla particolare macchina su cui è definito

**Non a caso viene definito *linguaggio macchina***

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi  
di Cassino

Se si volesse implementare un dato algoritmo attraverso la scrittura di un programma in L.M. sarebbe quindi necessario:

- conoscere dettagliatamente tutti i codici operativi e la loro codifica
- decidere in quali registri (di memoria o interni alla CPU) vadano memorizzati i dati
- determinare, per ogni singola operazione richiesta dall'algoritmo, la sequenza di istruzioni in linguaggio macchina che la realizzano
- definire un'opportuna tecnica di codifica per ogni tipo di dati considerato
- limitarsi a utilizzare solo i calcolatori per cui esista una tale competenza, tenendo comunque presente che il programma scritto per un certo calcolatore non è eseguibile su altre macchine

**Impresa difficile, ma non impossibile**

*F. Tortorella*

Corso di Calcolatori Elettronici

**Università degli Studi  
di Cassino**

### **Linguaggio di programmazione**

- linguaggio formale, con costrutti precisi per la definizione dei dati e delle operazioni
- gestione completa dei tipi fondamentali; possibilità di definire tipi strutturati
- costrutti che realizzano le principali azioni elaborative richieste

### **Calcolatore**

- linguaggio rigido e complicato
- gestione dei tipi quasi nulla
- istruzioni estremamente semplici

*F. Tortorella*

Corso di Calcolatori Elettronici

**Università degli Studi  
di Cassino**

## Linguaggio di programmazione

- linguaggio formale, con costrutti precisi per la definizione dei dati e delle operazioni
- gestione completa dei tipi fondamentali; possibilità di definire tipi strutturati
- costrutti che realizzano le principali azioni elaborative richieste



orientato al linguaggio (*front end*)

## Compilatore

orientato alla macchina (*back end*)



## Calcolatore

- linguaggio rigido e complicato
- gestione dei tipi quasi nulla
- istruzioni estremamente semplici