

Università degli Studi  
di Cassino

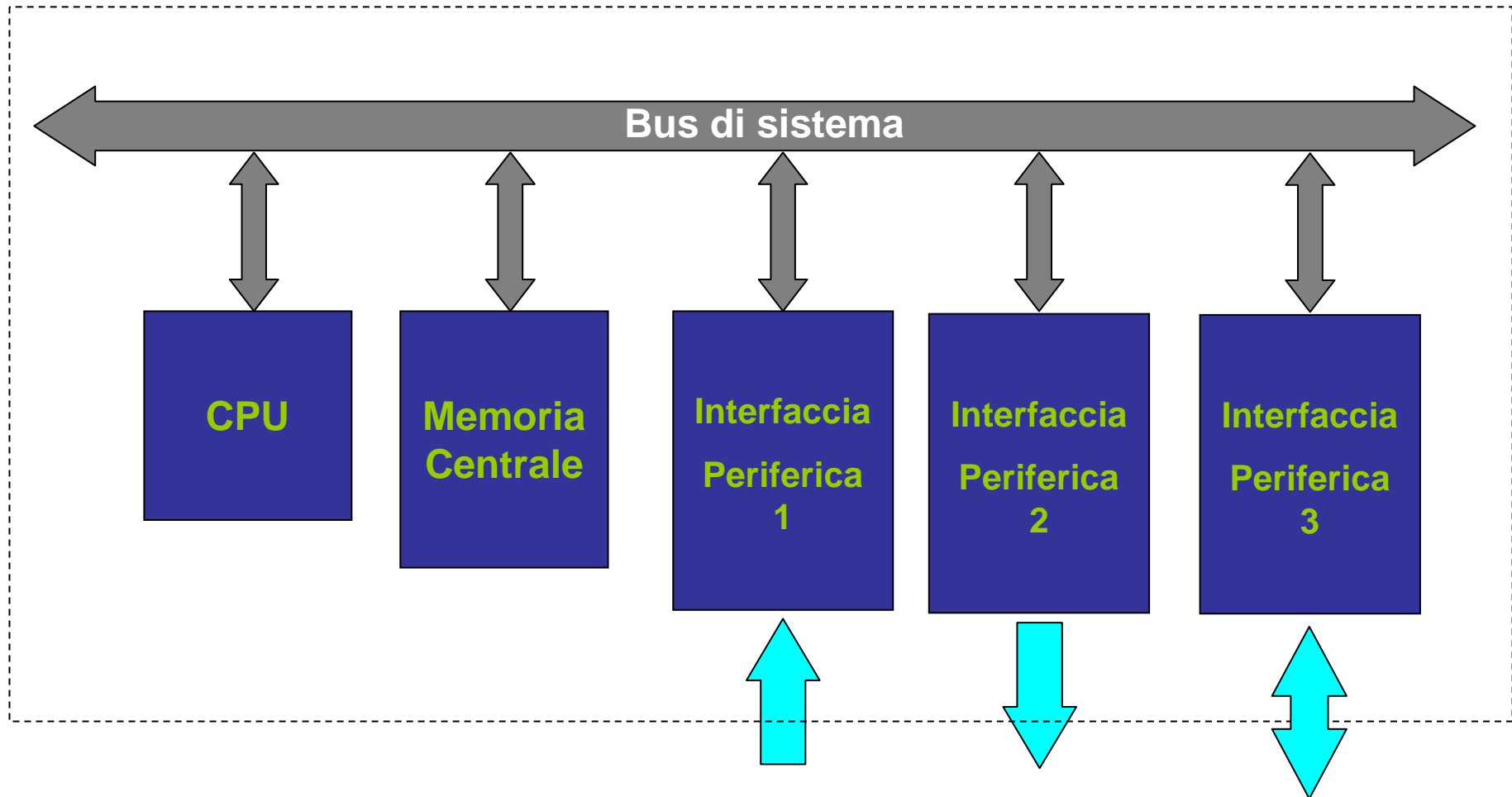
**Corso di Fondamenti di  
Informatica**

***Elementi di Architettura***

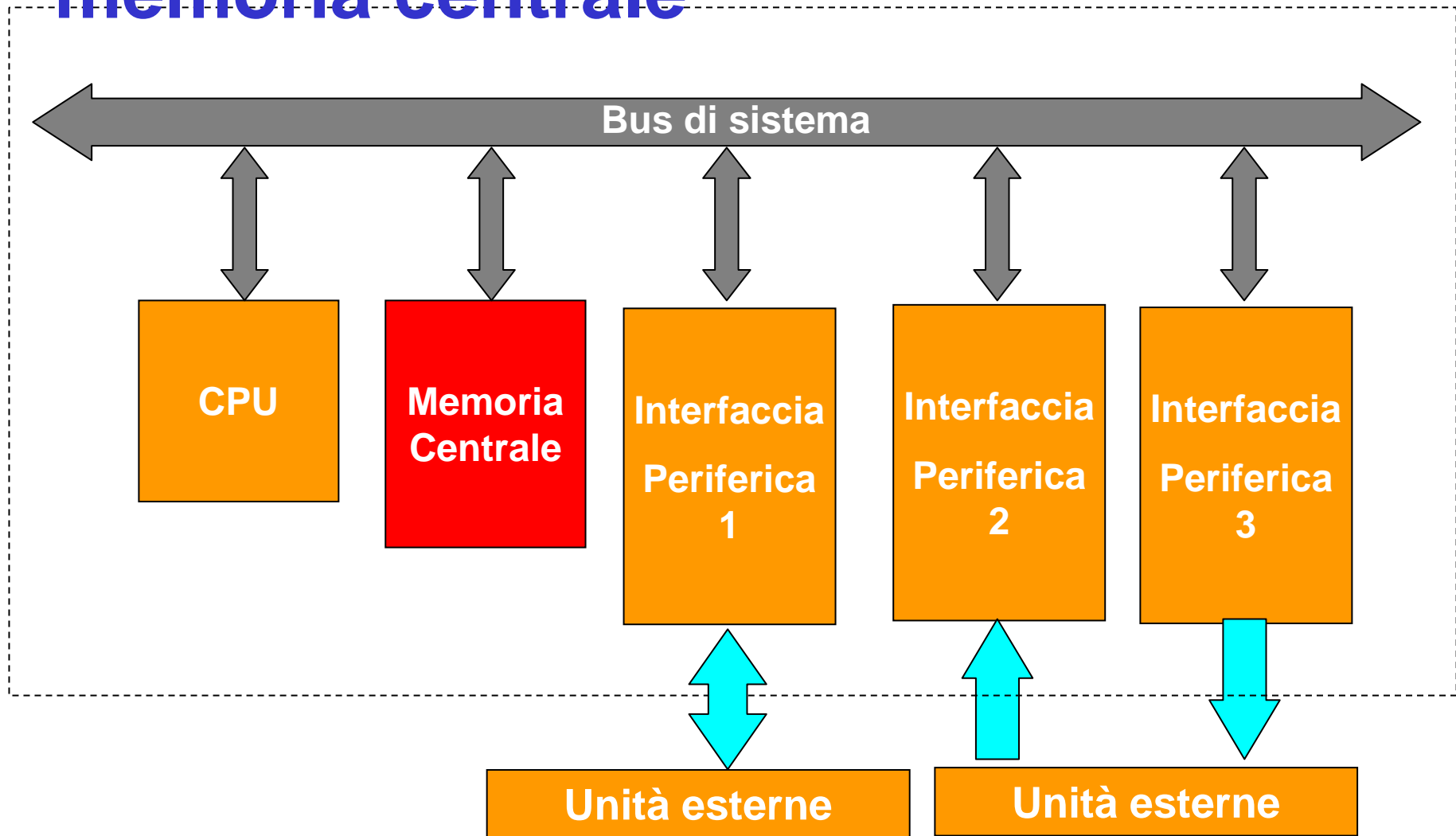
Anno Accademico 2009/2010

Francesco Tortorella

# Modello di von Neumann



# Modello di von Neumann: la memoria centrale



# 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

**Quanti bit sono necessari per codificare un indirizzo ?**

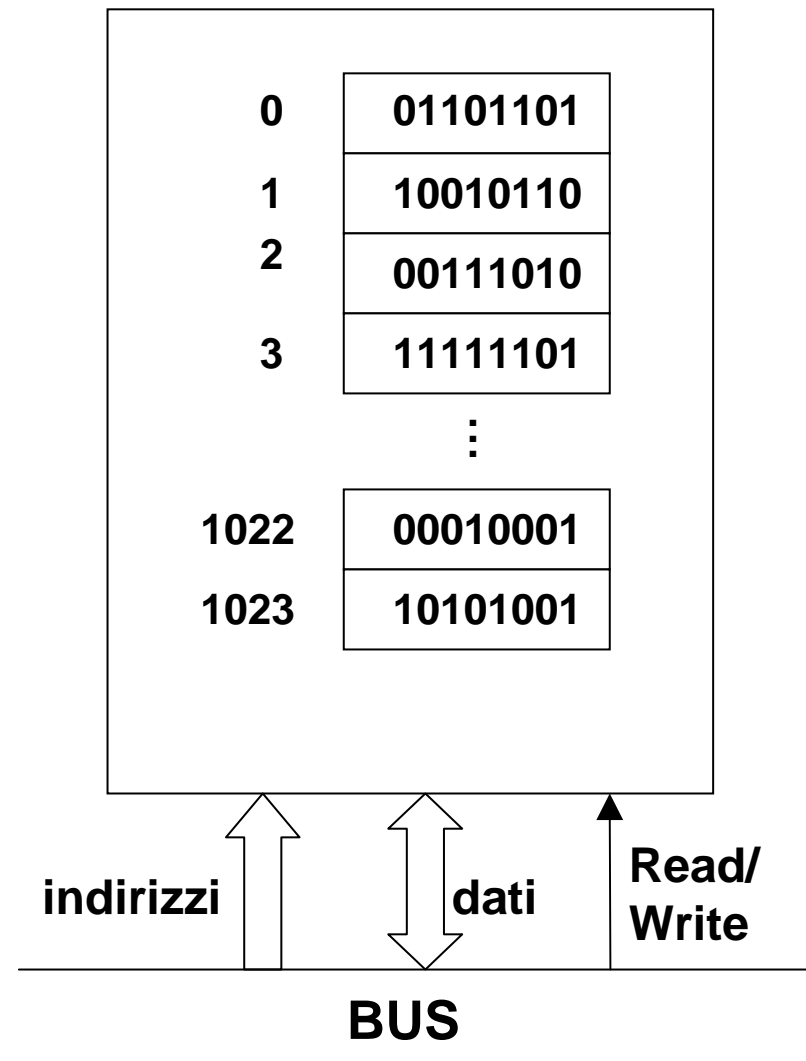
0	01101101
1	10010110
2	00111010
3	11111101
	⋮
1022	00010001
1023	10101001

# Organizzazione della memoria principale

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



# 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

# 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.
- **RAM dinamica o DRAM** (*Dynamic* Random Access Memory): Alta densità di integrazione, economica, lenta
  - *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

- **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.



# Organizzazione del Sistema di Memoria

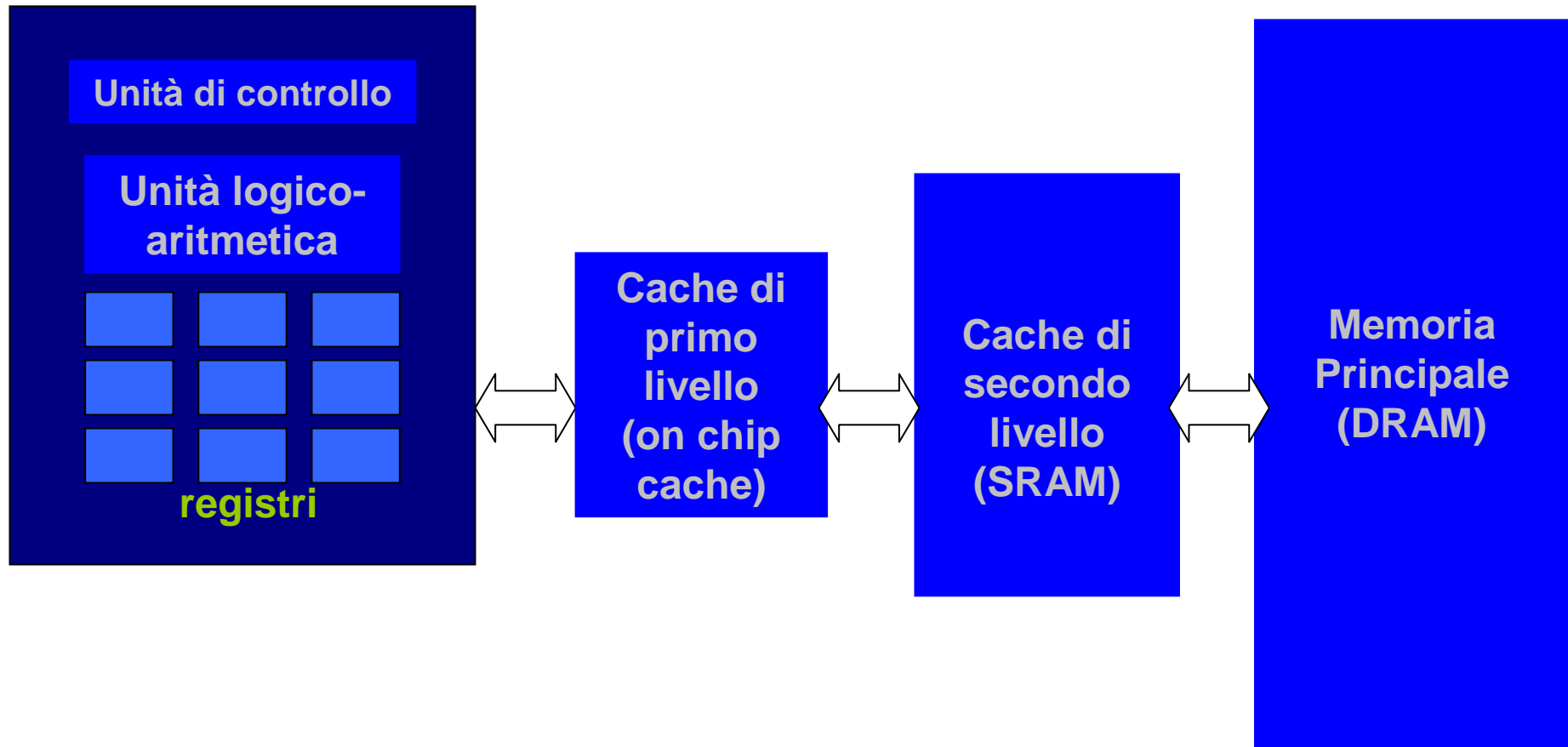
- Requisiti ideali di un sistema di memoria:
  - **capacità infinita**
  - **velocità infinita**
- Evidenza:
  - le memorie economiche (DRAM) sono lente
  - le memorie veloci (SRAM) sono costose e meno integrabili
- Come realizzare un sistema di memoria che sia capiente, economico e veloce ?

## Un sistema basato su una gerarchia di memoria

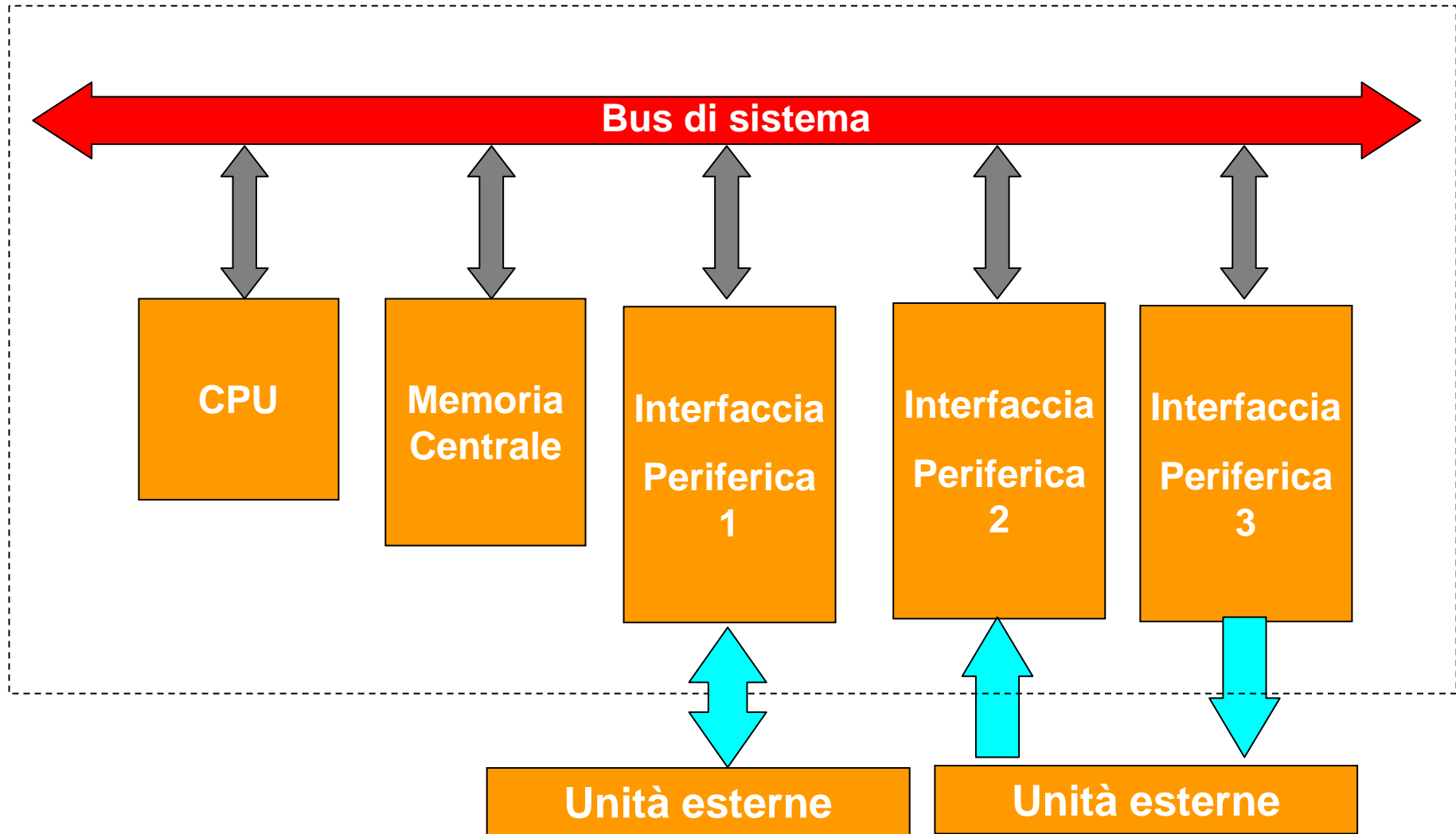
# La memoria cache

- 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.

# Sistema di memoria in un calcolatore attuale



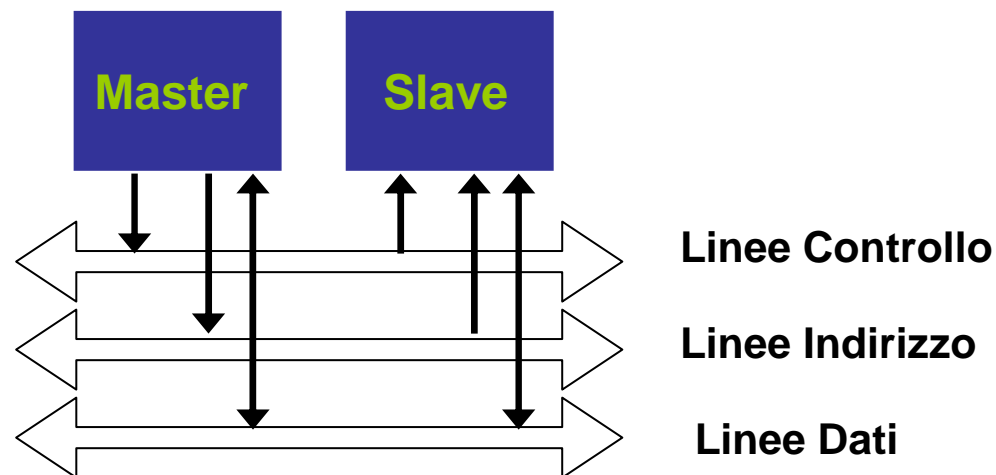
# Modello di von Neumann: il bus



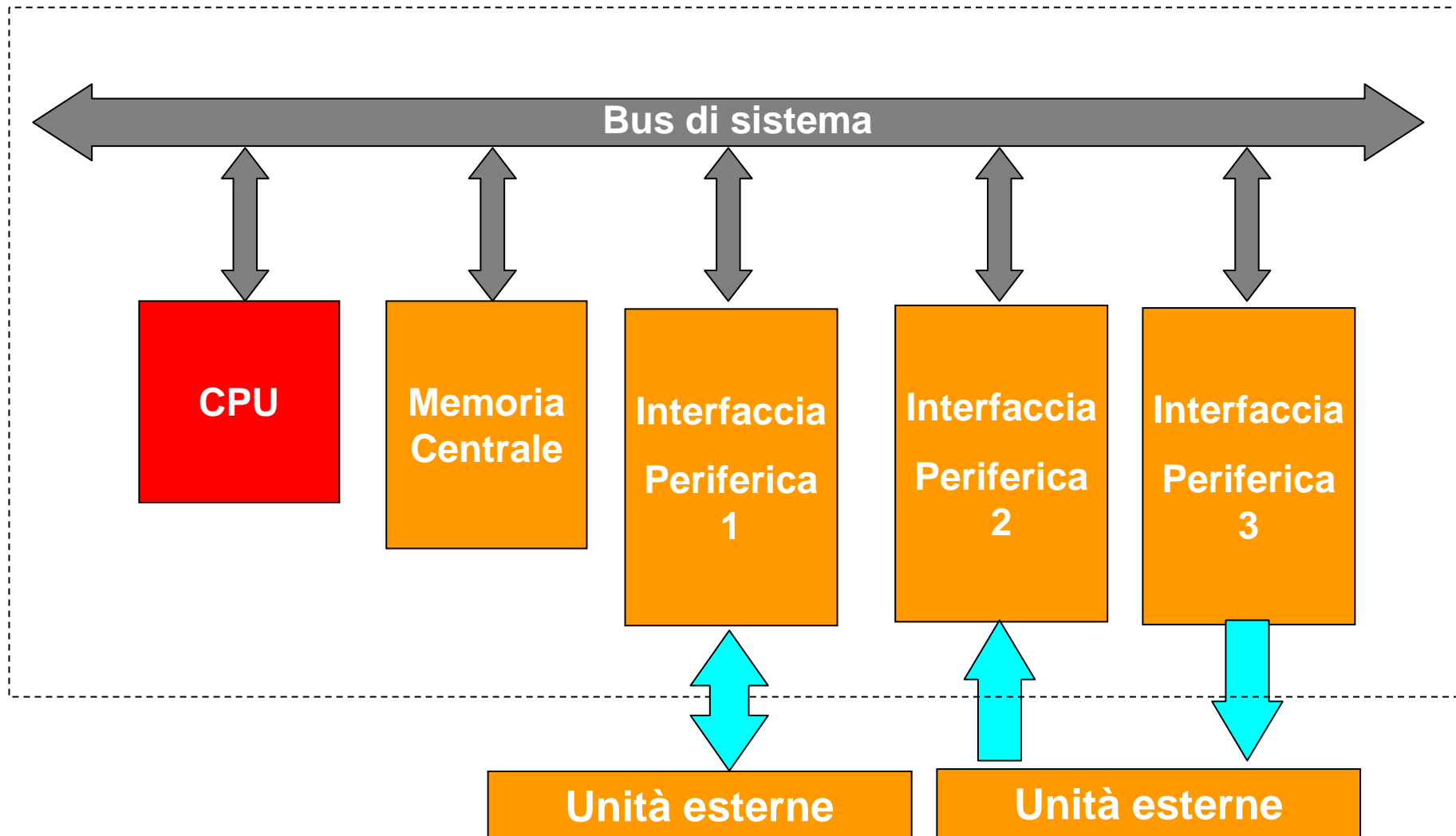
# Il bus

- 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:

- **linee dati**
- **linee indirizzi**
- **linee controllo**



# Modello di von Neumann: la CPU



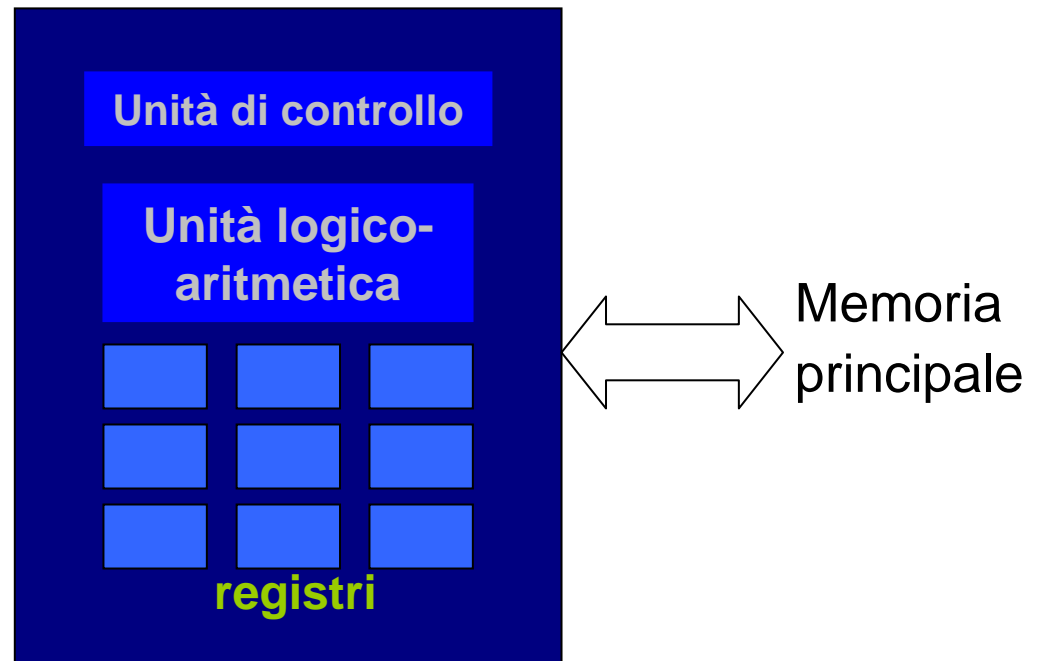
# CPU (Central Processing Unit)

## 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**

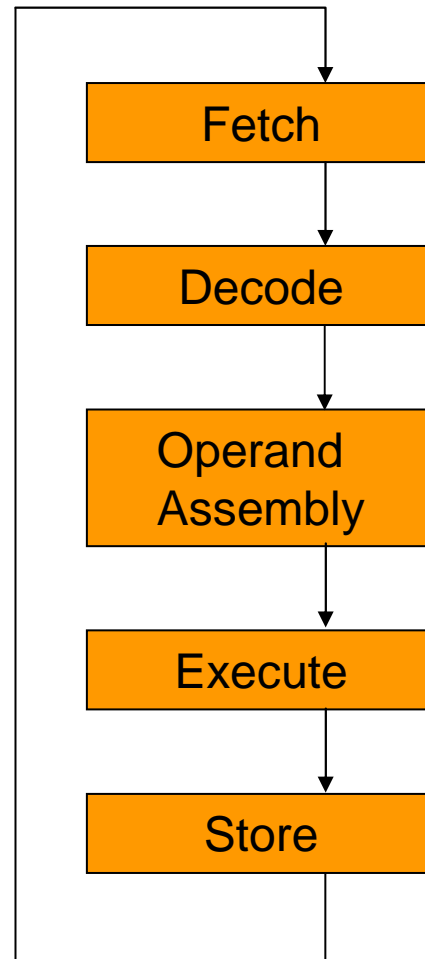
Viene portata a termine l'esecuzione dell'operazione prevista dalla istruzione

### **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

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**

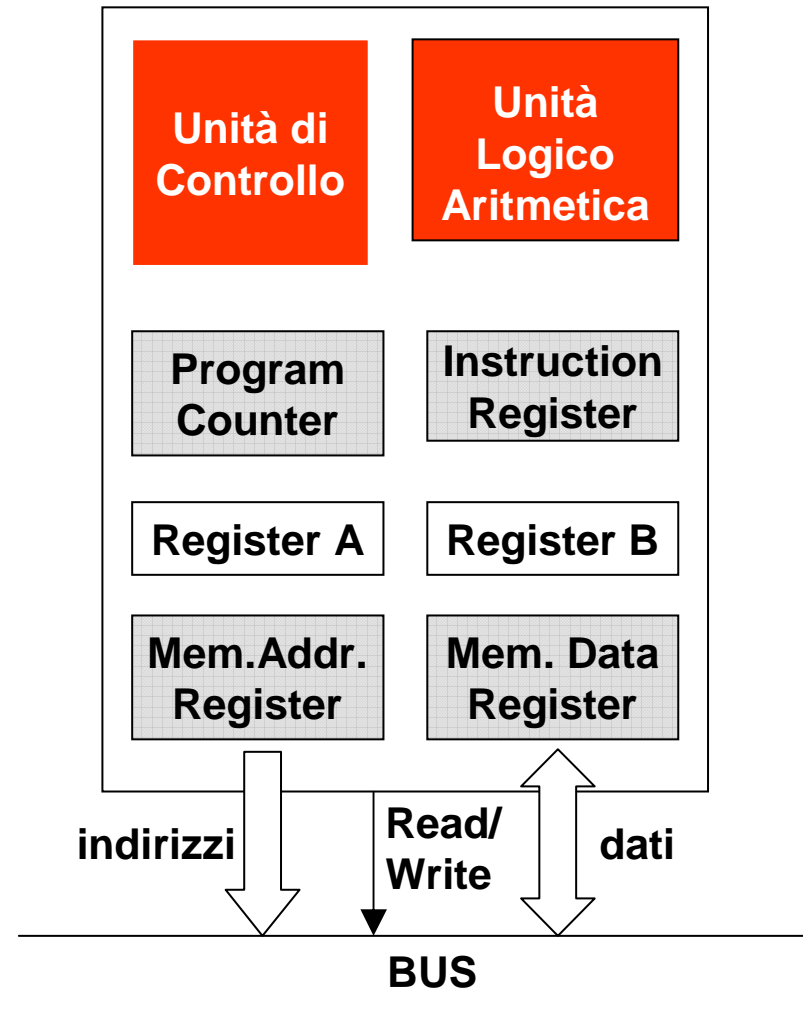
- Program Counter (PC)
- Mem. Address Reg. (MAR)
- Mem. Data Register (MDR)
- Istrunction Register (IR)

**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).

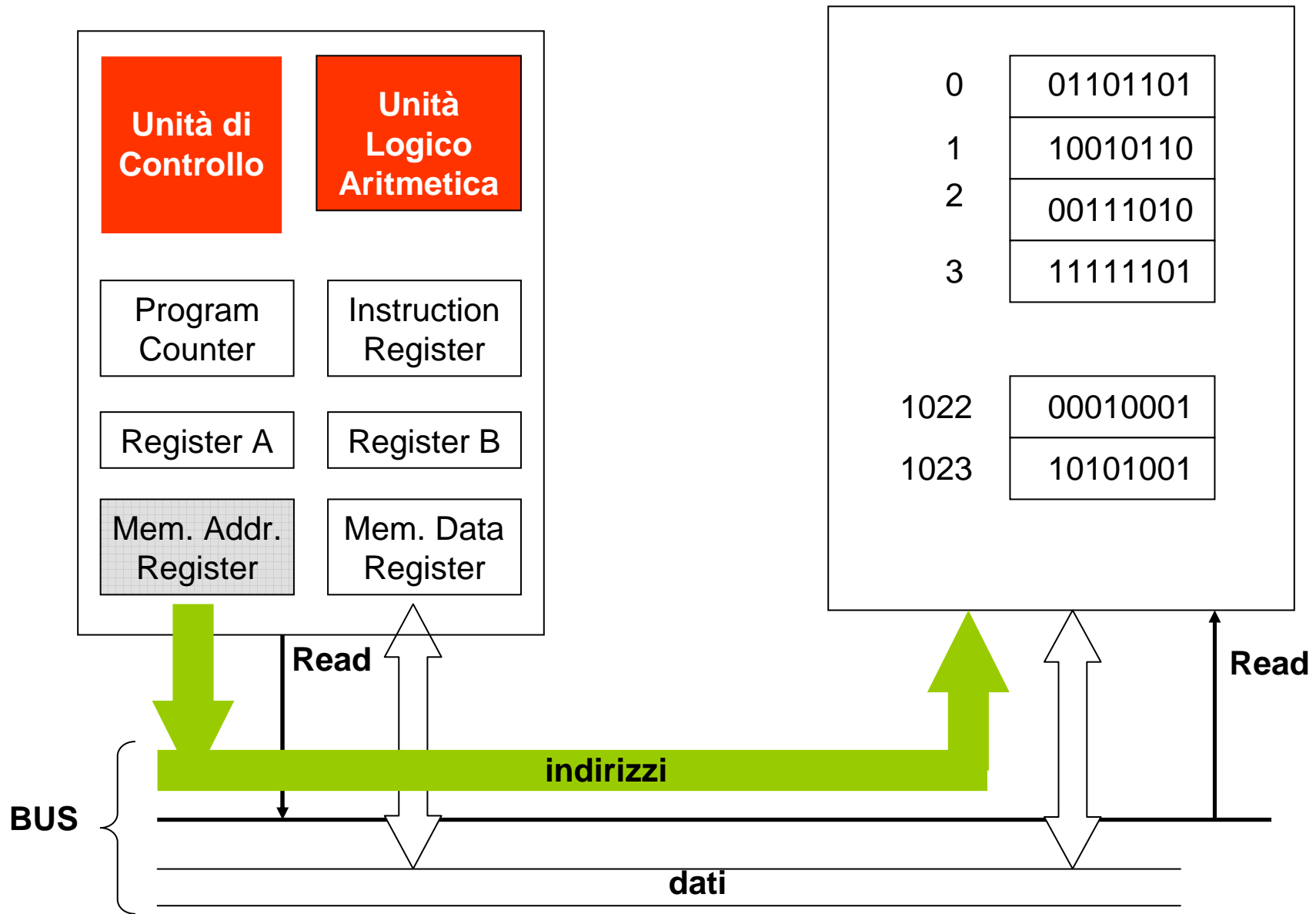


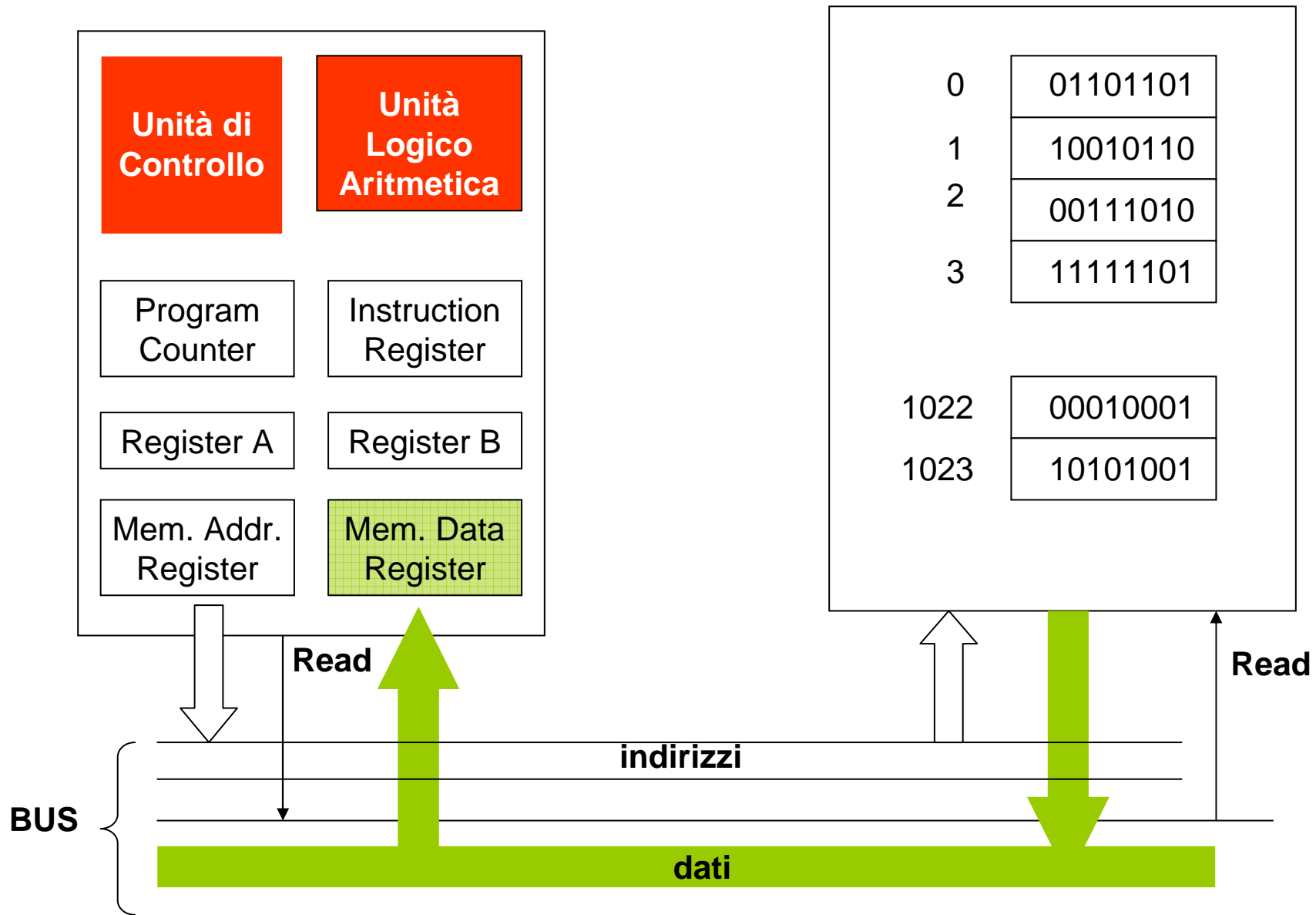
# Trasferimento CPU-memoria

- 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 registi interni.

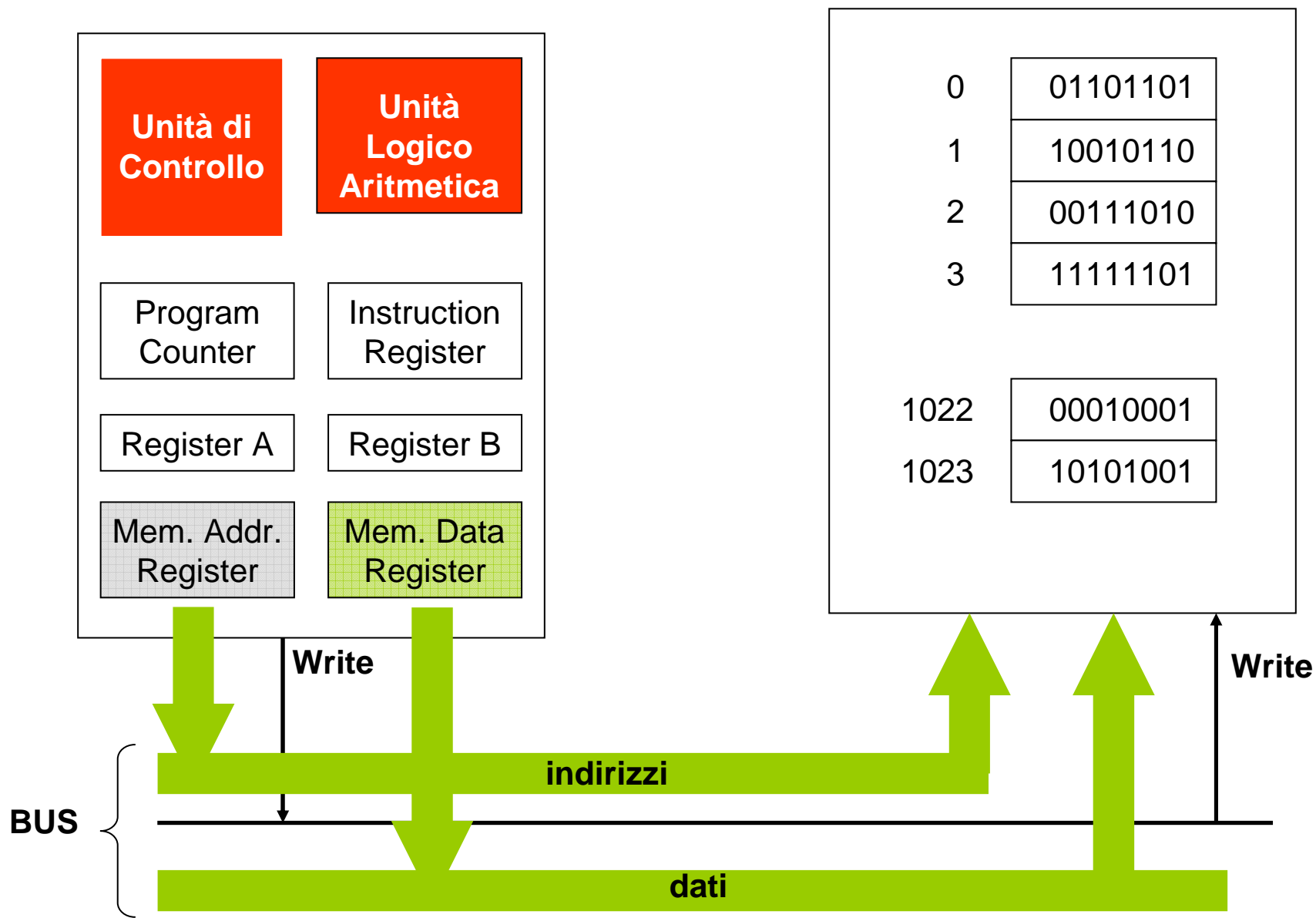


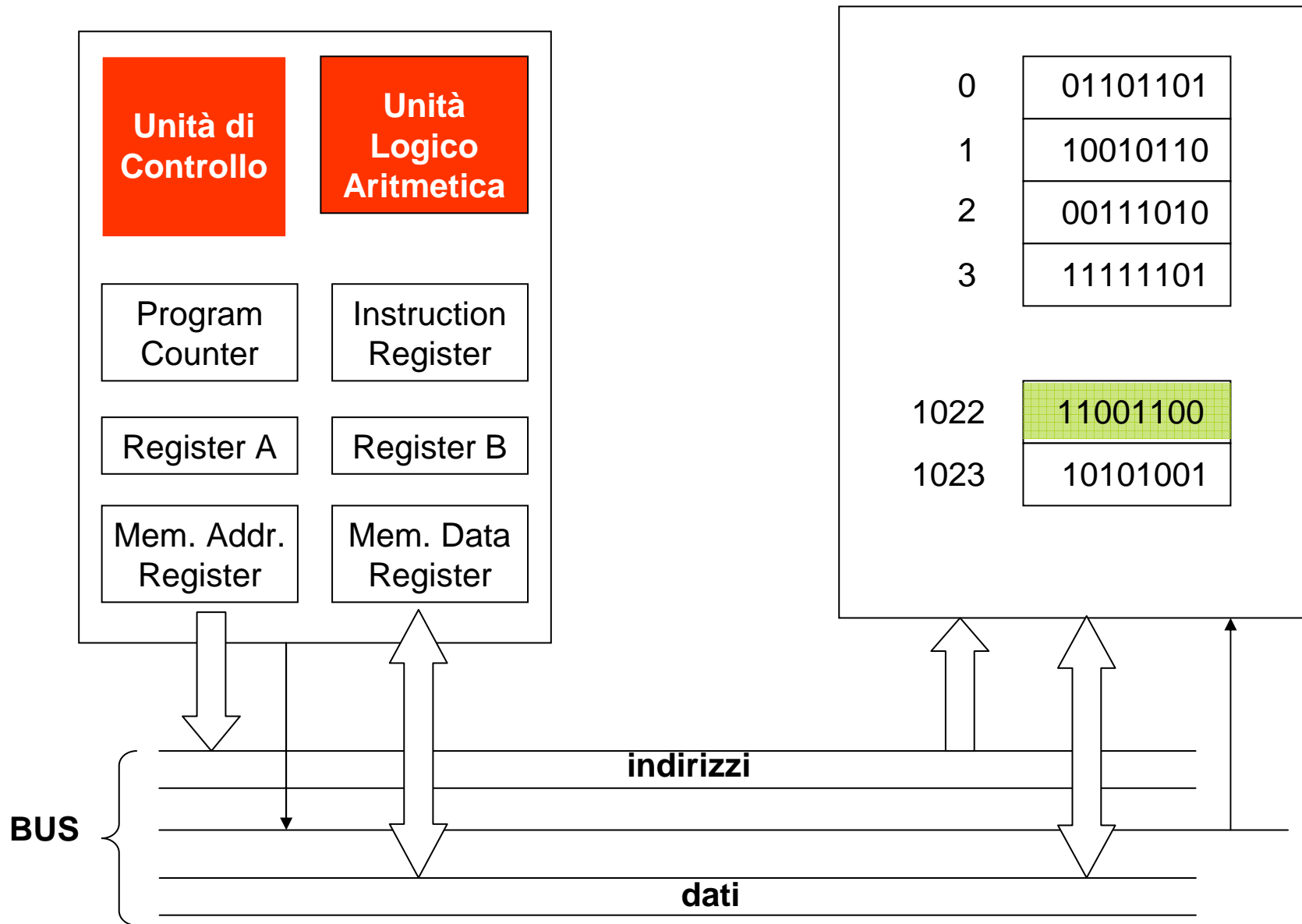




# Trasferimento CPU → memoria (scrittura)

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.





# Esempio di esecuzione di una istruzione

Consideriamo un'istruzione del tipo:

ADD (1021),(1022),1023

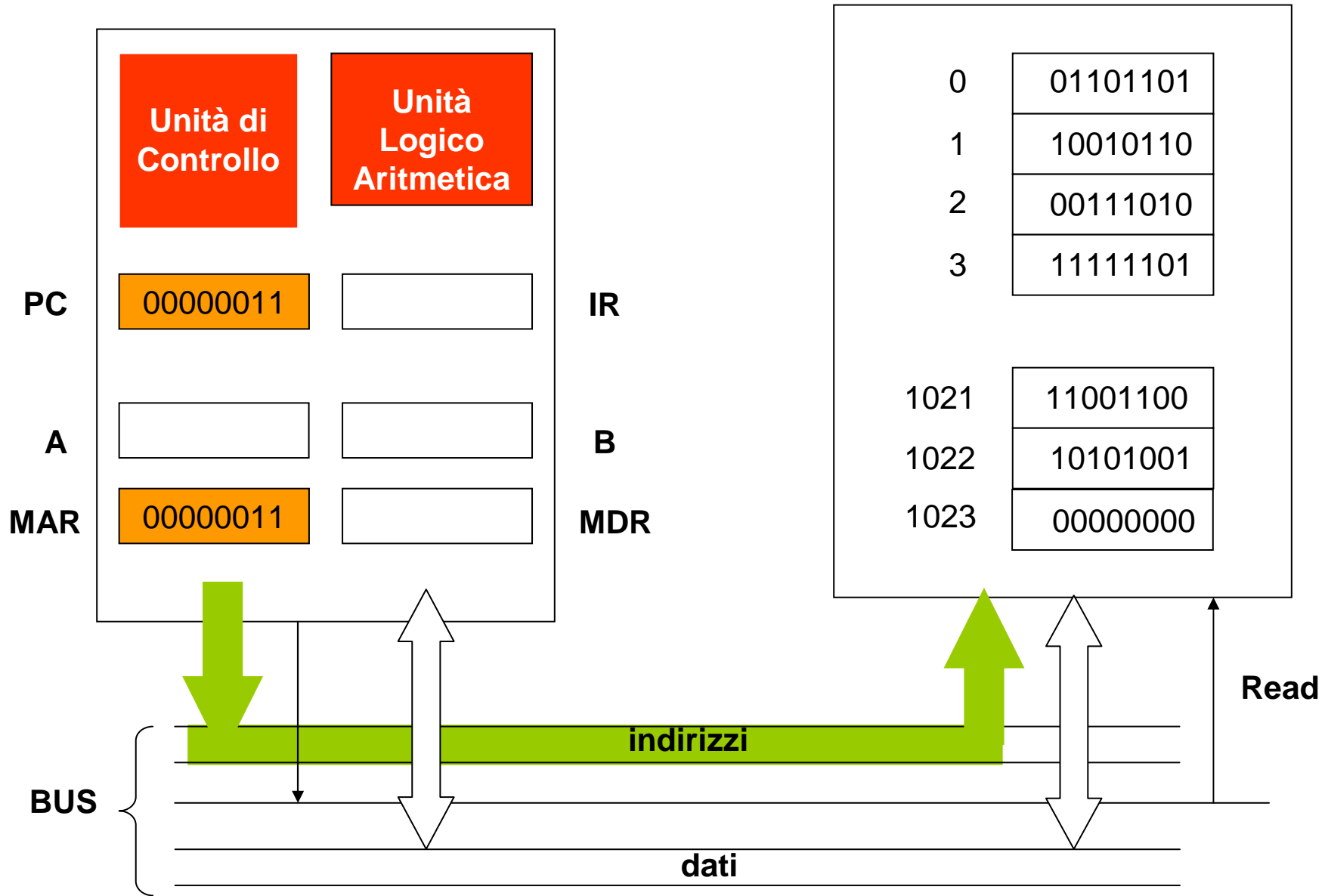
Il cui significato è:

“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”.

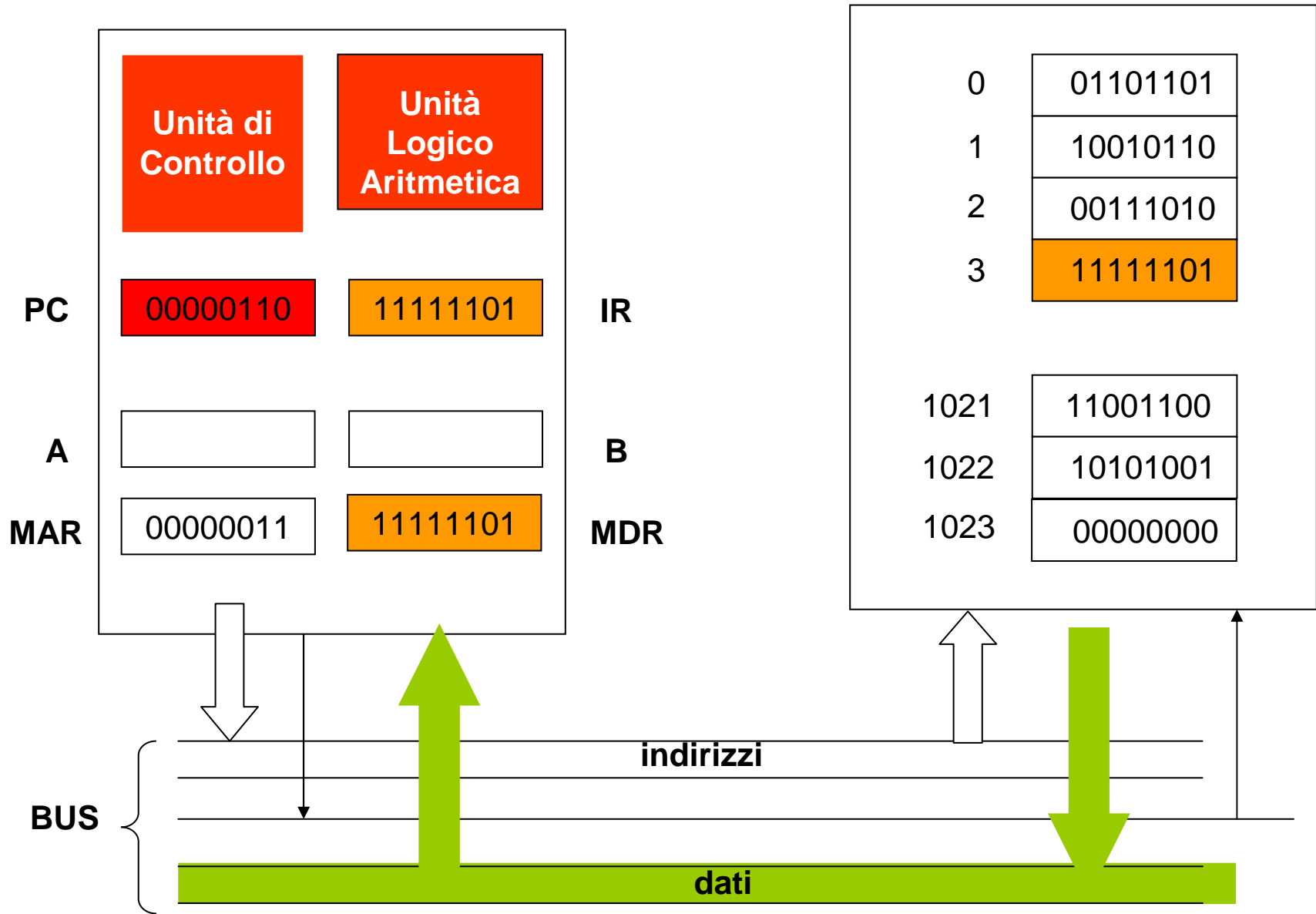
Supponiamo inoltre che l'istruzione si trovi memorizzata nel registro di memoria di indirizzo 3.

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

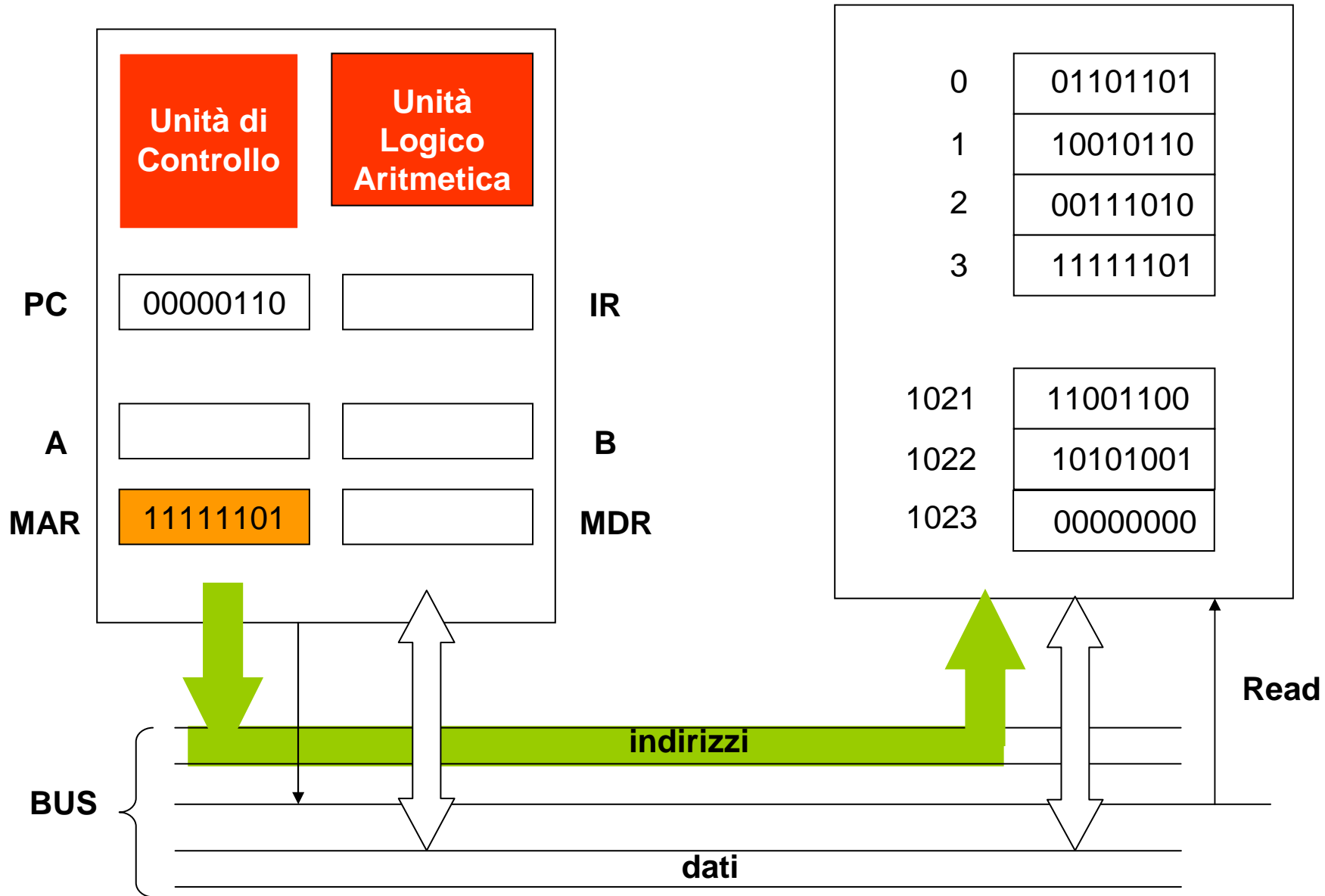
# Fase FETCH



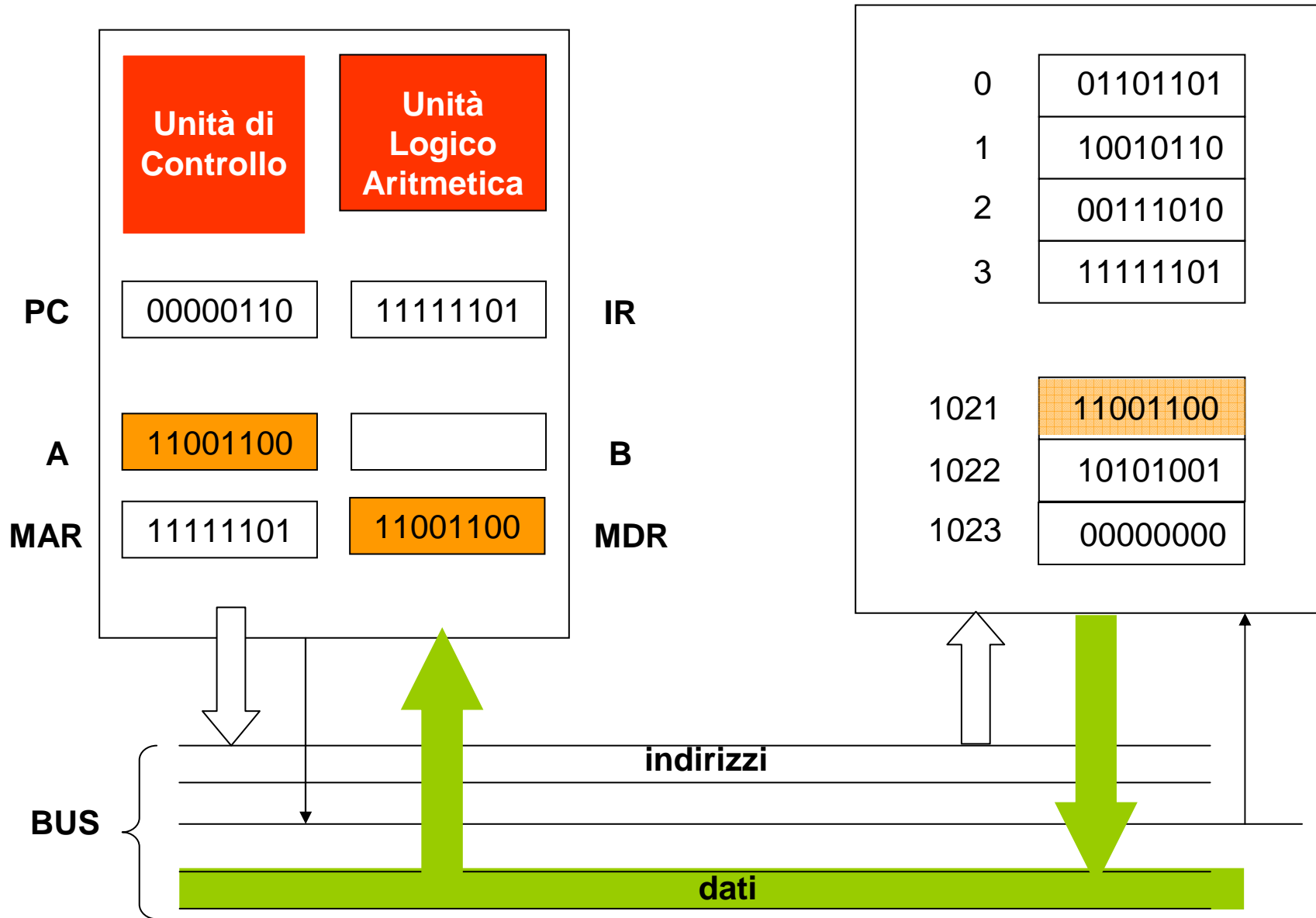
# Fase FETCH



# Fase OPERAND ASSEMBLY

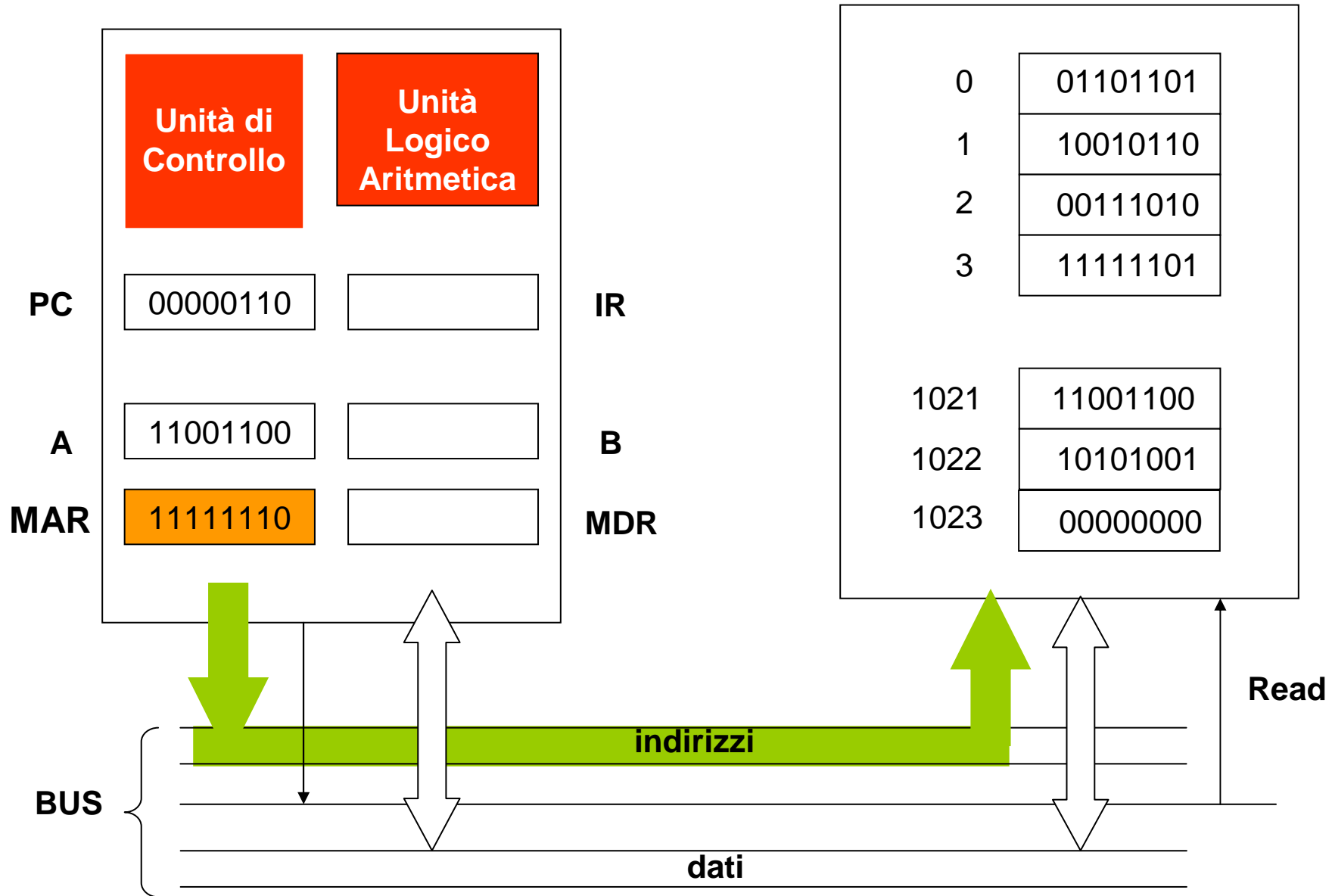


# Fase OPERAND ASSEMBLY

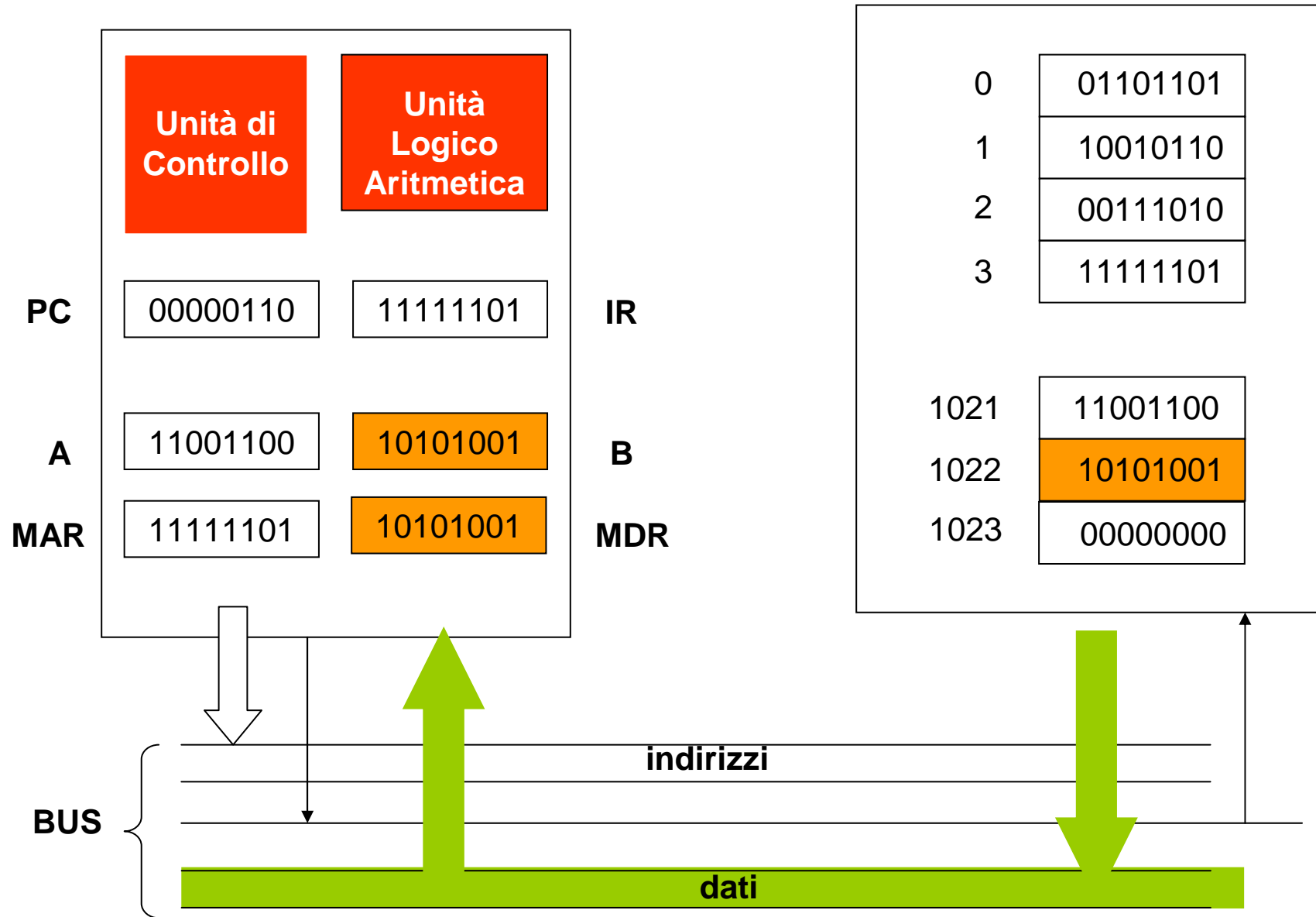




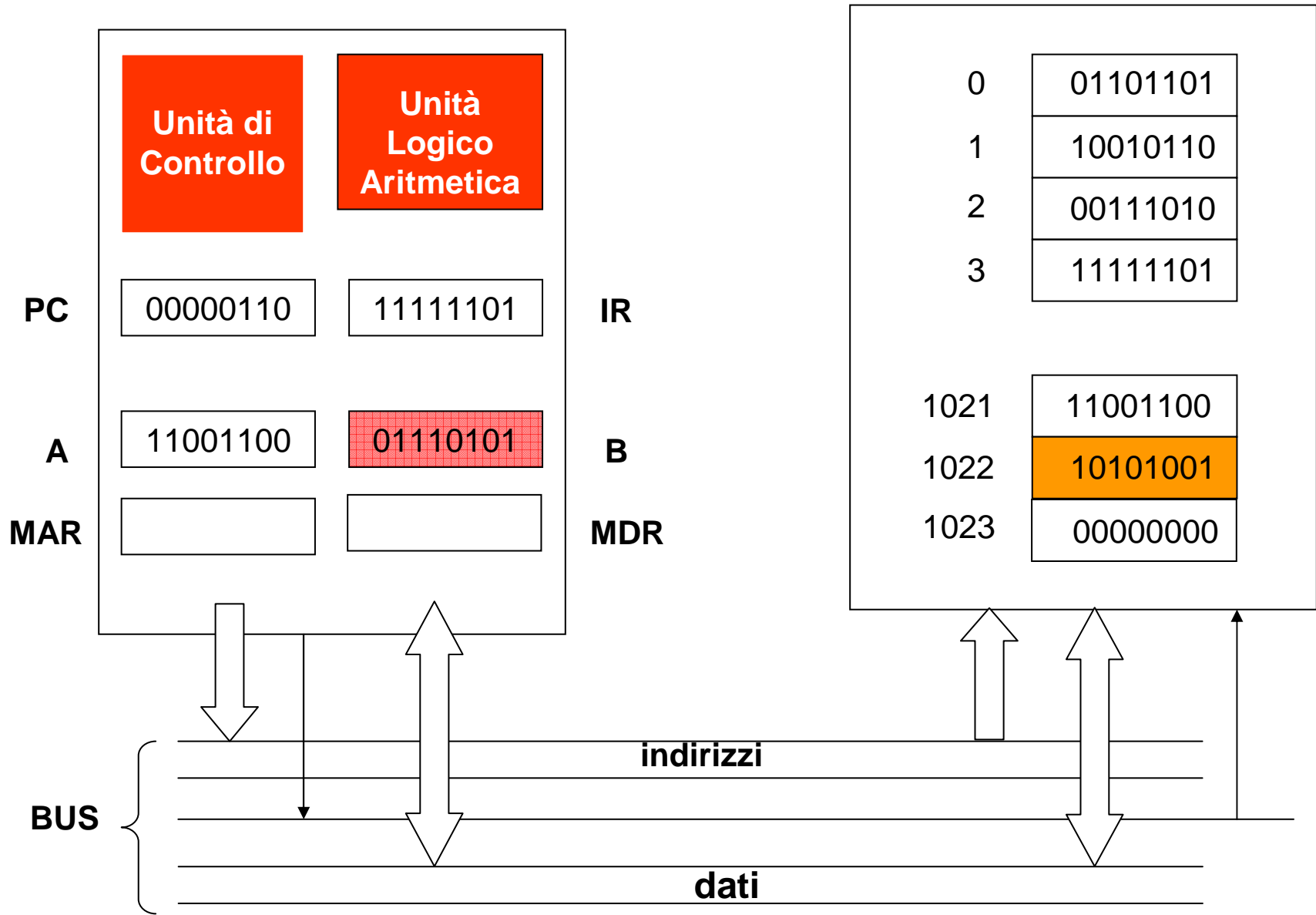
# Fase OPERAND ASSEMBLY



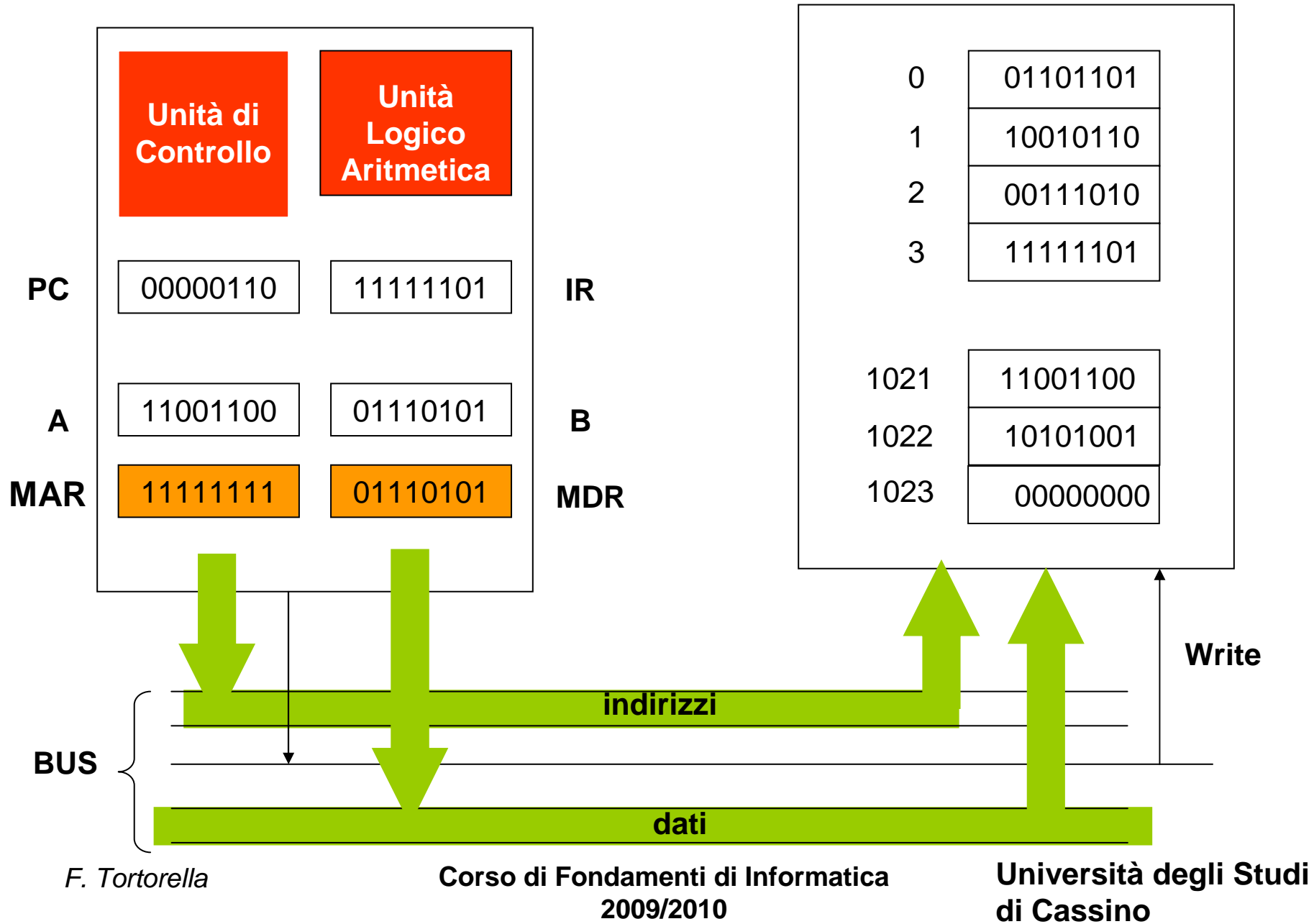
# Fase OPERAND ASSEMBLY



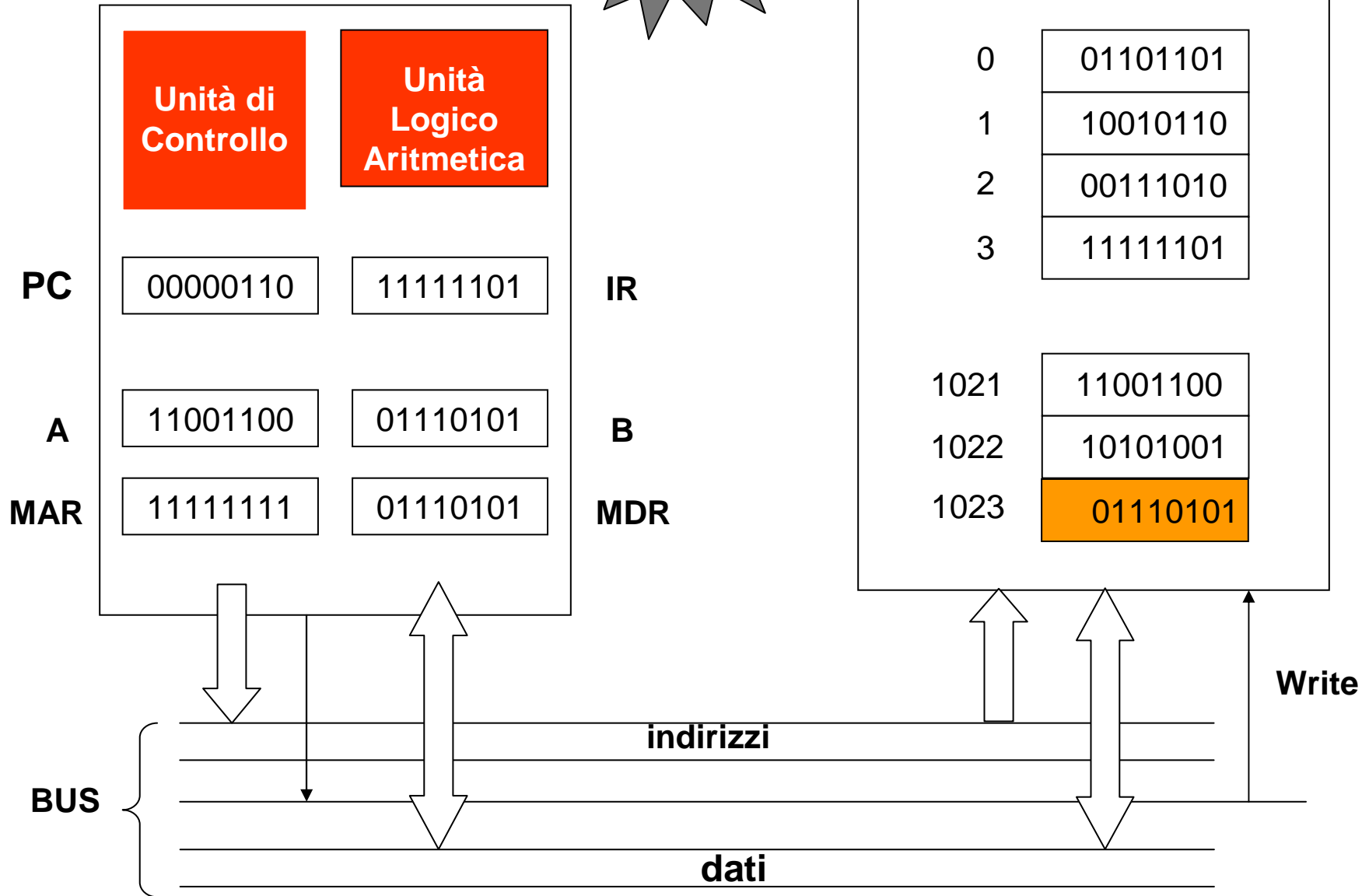
# Fase EXECUTE



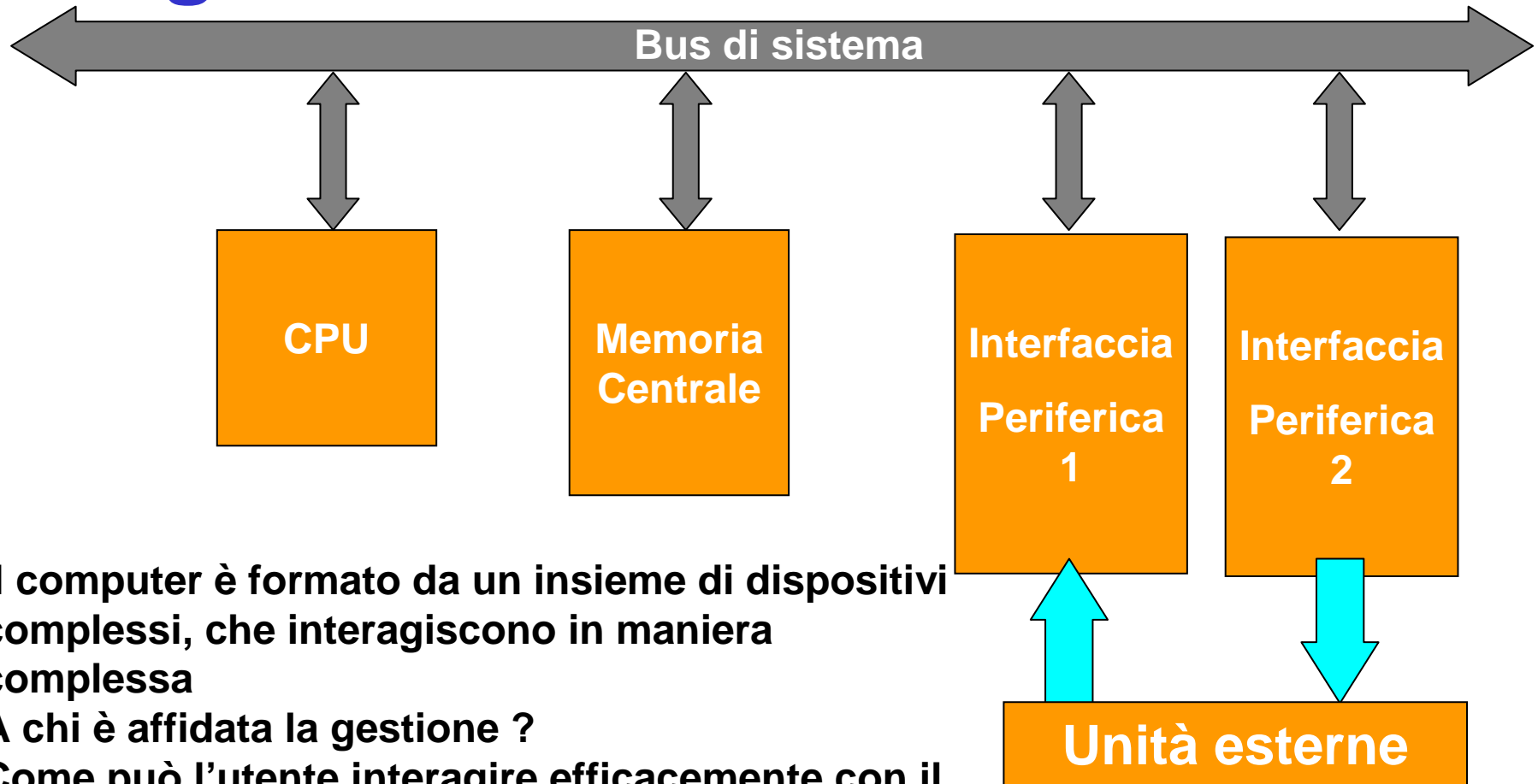
# Fase STORE



# Fase STORE



# Chi gestisce ?



**Il computer è formato da un insieme di dispositivi complessi, che interagiscono in maniera complessa**

**A chi è affidata la gestione ?**

**Come può l'utente interagire efficacemente con il sistema ?**

# L'hardware non basta...

- L'utente di un computer non può interagire direttamente con l'hardware perché:
  - è troppo complesso da gestire
  - offre dei servizi di livello estremamente basso
  - richiede conoscenze estremamente specialistiche
  - l'architettura hardware può essere estremamente diversa da computer a computer
- Il Sistema Operativo è un software apposito che offre all'utente gli strumenti per svolgere le operazioni necessarie e gestire le risorse a disposizione

# Sistema Operativo

- Il Sistema Operativo è uno strato software che
  - opera direttamente sull'hardware
  - isola dai dettagli dell'architettura hardware
  - fornisce un insieme di funzionalità di alto livello
- Gli obiettivi dell'impiego del S.O. sono:
  - Convenienza: rende l'utilizzo del computer più semplice ed intuitivo
  - Efficienza: permette di impiegare le risorse del sistema in maniera più efficiente
  - Capacità di evoluzione: permette l'introduzione di nuove funzionalità e/o il miglioramento delle risorse hardware senza interferire con il servizio svolto



# Sistema Operativo

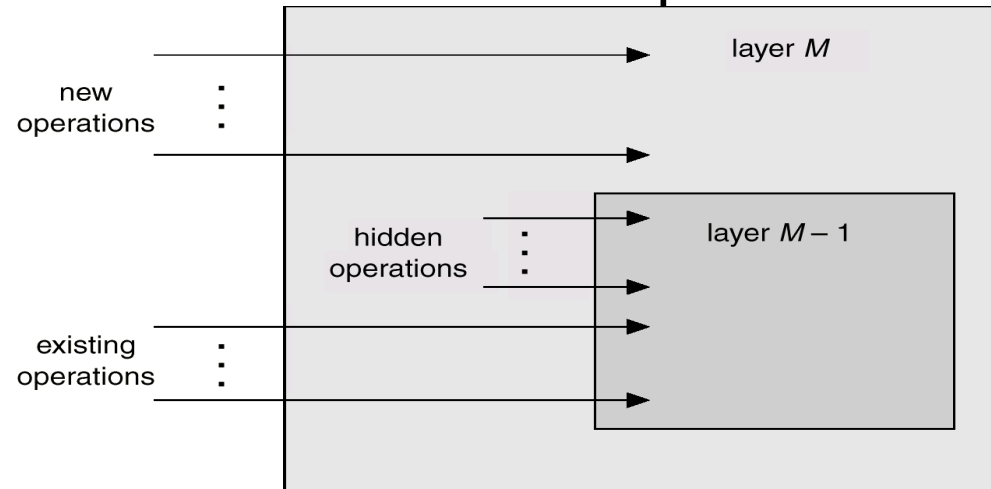
- Il S.O. è un software indispensabile, è il primo ad essere eseguito all'accensione del computer e resta sempre in RAM
- Esempi di S.O. diffusi:
  - MS Windows 95, 98, NT, 2000, XP
  - Mac OS
  - OS/2
  - Linux
- I programmi sono realizzati per essere eseguiti su un certo S.O.

# I servizi del Sistema Operativo

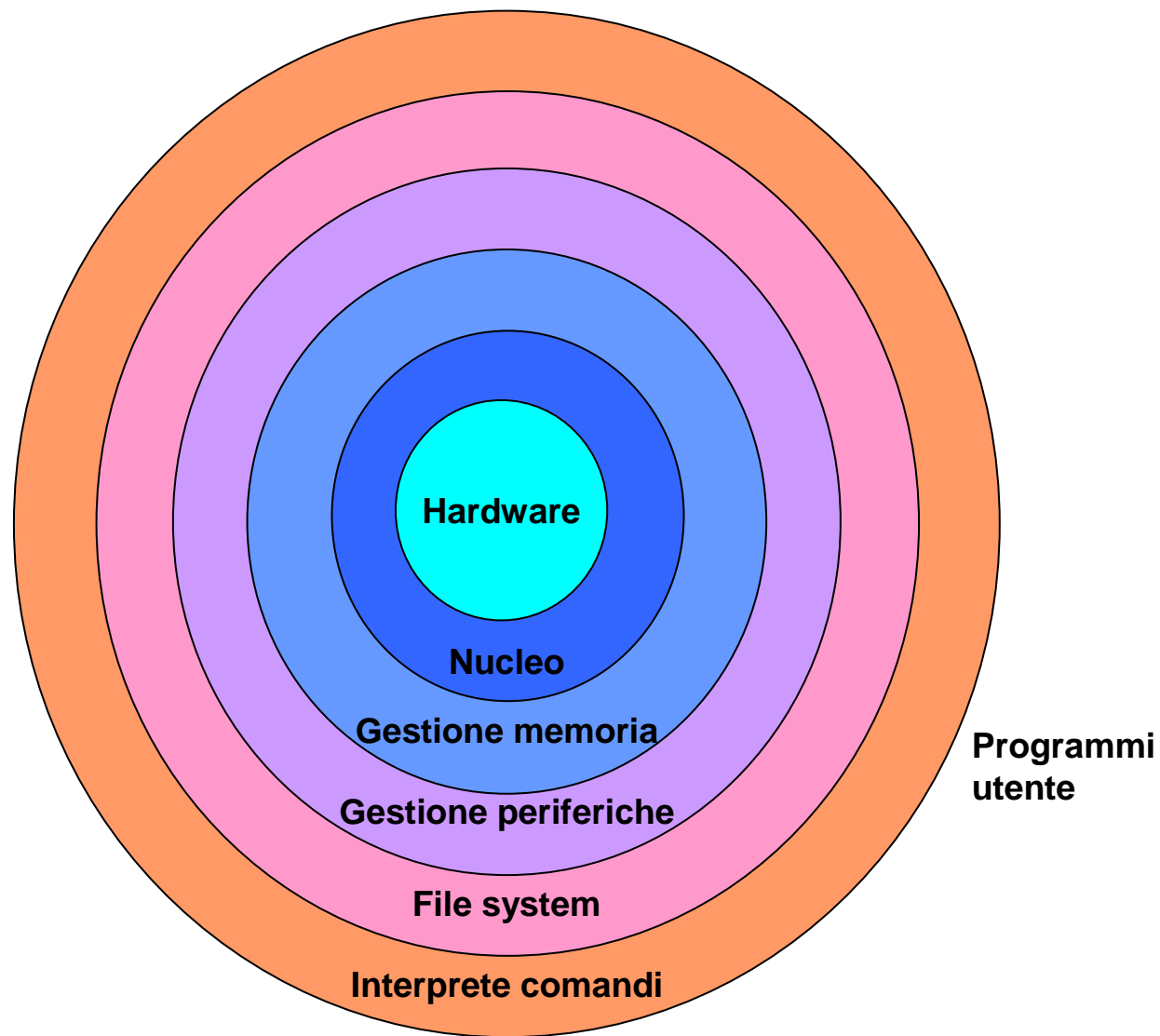
- Sviluppo di programmi
- Esecuzione dei programmi
  - Multitasking
- Accesso ai dispositivi di I/O e di memoria di massa
- Accesso controllato ai file
  - Organizzazione logica dei dischi
- Accesso al sistema
  - Criteri di protezione

# Struttura del Sistema Operativo

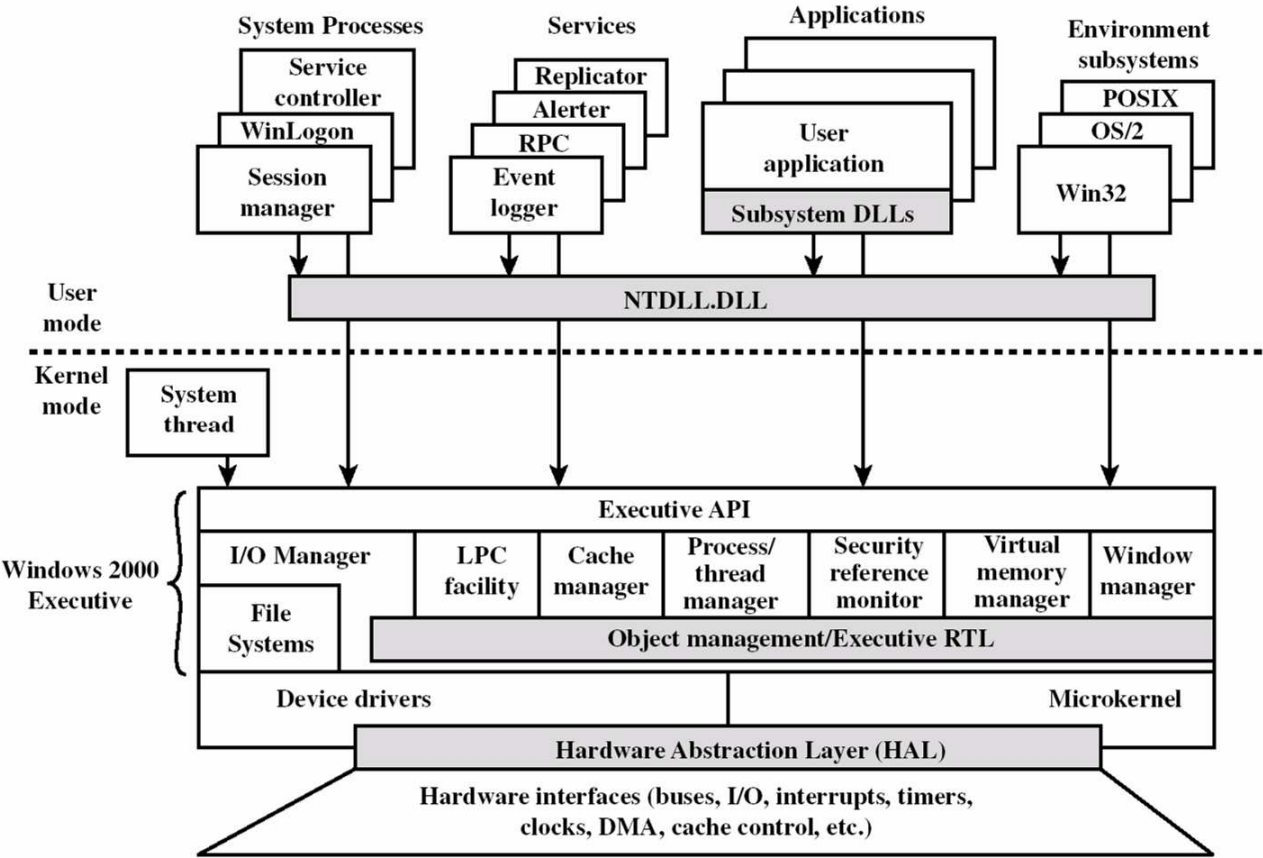
- A causa della loro complessità i Sistemi Operativi sono di solito strutturati come una serie di livelli (architettura a buccia di cipolla)
- Ogni livello realizza un certo sottoinsieme di funzioni
- Ogni livello realizza una macchina virtuale, che nasconde i meccanismi implementativi e offre un insieme ben definito di funzionalità ai livelli superiori



# Struttura del Sistema Operativo



# Esempio: struttura di Windows



# L'interprete di comandi

- L'interprete dei comandi (o shell) è la parte più esterna del SO con la quale l'utente interagisce
- Attraverso l'interprete, l'utente richiede un servizio al SO (esecuzione di un programma, copia di un file, ecc.)
  - l'utente inserisce un comando in accordo ad una certa sintassi
  - l'interprete riconosce il comando ed attiva una serie di funzioni dei livelli interni
  - l'esecuzione di tali funzioni di fatto porta al compimento del comando richiesto
- L'interprete si presenta
  - con un'interfaccia di tipo alfanumerico
  - con un'interfaccia di tipo grafico (GUI)

# L'interprete di comandi

Interfaccia alfanumerica (finestra DOS)

comandi

```
04/02/2002 11.46      1.433 gdbtk.ini
18/12/2001 09.17      <DIR>  My Music
24/12/2001 18.37      <DIR>  Program Files
03/02/2002 23.44      <DIR>  Programmi
20/12/2001 14.58      <DIR>  sploff
19/01/2002 02.30      <DIR>  sprann
20/12/2001 14.34      <DIR>  TEMP
03/02/2002 23.45      <DIR>  WINNT
          4 File      755,089 byte
          11 Directory 25,467,158,528 byte disponibili

C:\>date
Data corrente: mer 06/02/2002
Immettere nuova data: (gg/mm/aa)

C:\>dir /w
Il volume nell'unit  C non ha etichetta.
Numero di serie del volume: DC17-0F42

Directory di C:\

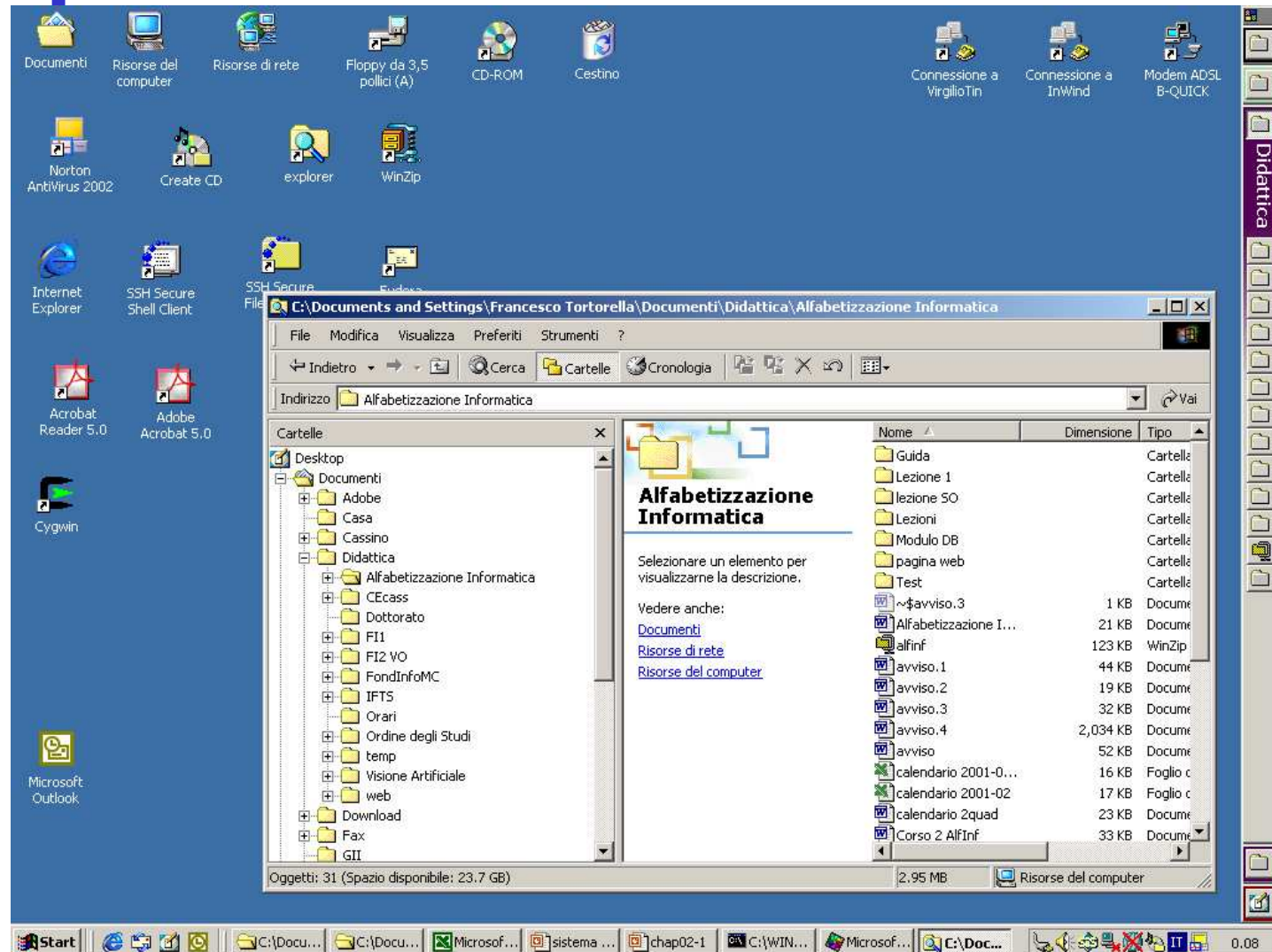
[Acrobat3]          AdobeWeb.log      dbec.log
dbec.ps            [Didattical]     [Documents and Settings]
[Download]         gdbtk.ini         [My Music]
[Program Files]   [Programmi]      [sploff]
[sprann]           [TEMP]           [WINNT]
          4 File      755,089 byte
          11 Directory 25,467,158,528 byte disponibili

C:\>
```

uscite

# L'interprete di comandi

Interfaccia  
GUI



F. Tortorella

Corso di Fondamenti di Informatica  
2009/2010

Università degli Studi  
di Cassino



# Il File System

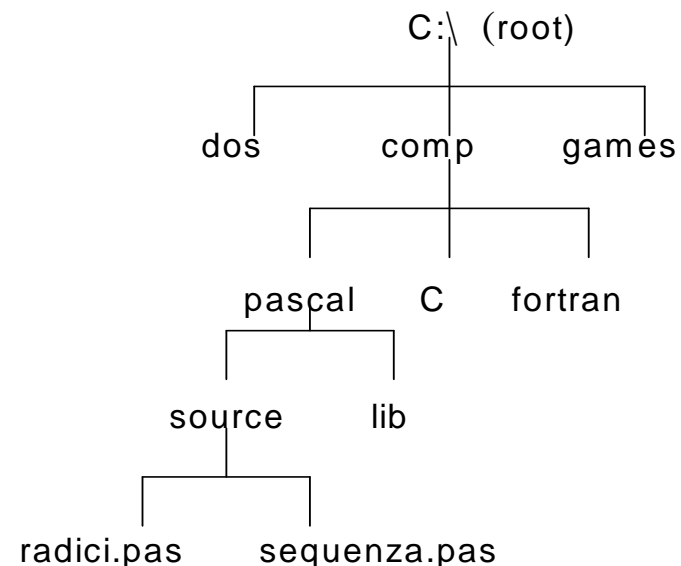
- Le unità di memoria di massa forniscono il supporto fisico per la memorizzazione permanente dei dati, e presentano caratteristiche estremamente diverse a seconda della casa costruttrice e del tipo di unità.
- Il File System offre una visione logica uniforme della memorizzazione dei dati basata su un'unità di memoria logica, il file, definita indipendentemente dalle caratteristiche fisiche delle particolari unità.
- Il file è un insieme di informazioni, correlate e registrate nella memoria di massa, identificato da un nome, che può essere formato da più sottoparti.
  - nome: si riferisce ai contenuti del file
  - estensione: si riferisce al tipo del file
- Dal punto di vista dell'utente, un file è la più piccola porzione (logica) di memoria secondaria: i dati, cioè, possono essere scritti nella memoria secondaria solo all'interno di un file.

# Contenuti dei file

- Le informazioni registrate all'interno di un file sono di due tipi:
- dati veri e propri
  - programmi eseguibili
  - testi
  - immagini
  - dati numerici
  - ...
- attributi di interesse per l'utente
  - dimensione del file
  - data di creazione e/o ultima modifica
  - permessi di accesso

# Organizzazione logica dei files

- L'insieme dei file presenti in memoria di massa è organizzato secondo una struttura gerarchica ad albero, in cui i nodi intermedi costituiscono le *directory* (che raggruppano altri files e directory secondo un criterio di omogeneità), mentre le foglie rappresentano i file.
- All'interno di tale struttura, un particolare file è univocamente identificato dal *path* (o percorso) che localizza la directory in cui il file è memorizzato.



# Organizzazione fisica dei files

- Da un punto di vista fisico, la registrazione del file sul disco viene realizzata dal sistema operativo disponendo il contenuto del file su un insieme di cluster possibilmente contigui.
- La registrazione dei dati è organizzata in maniera sequenziale, per cui le operazioni di lettura e scrittura possono avvenire solo a partire dall'inizio e procedendo verso la fine.
- La successione dei blocchi nei quali sono memorizzati i bytes che lo compongono può essere strutturata come:

