

Lezione 3:

Architettura del calcolatore

Architettura di Von Neumann
BUS, CPU e Memoria centrale
Ciclo di esecuzione delle istruzioni

Architettura del calcolatore

◆ Il calcolatore è:

- uno strumento programmabile per rappresentare, memorizzare ed elaborare informazioni
- un **sistema**, costituito da molte componenti

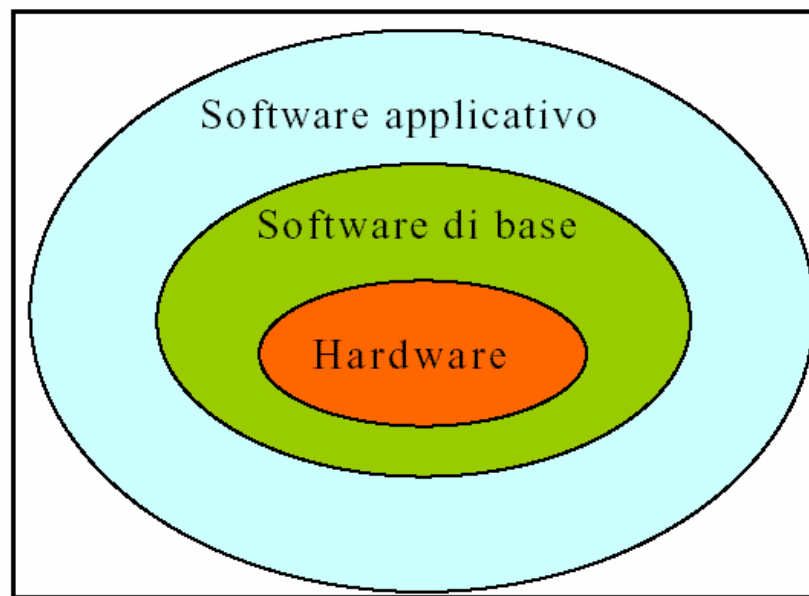
◆ Studiare l'architettura di un sistema significa:

- individuare ciascun componente del sistema
- comprendere i principi generali di funzionamento di ciascun componente
- comprendere come le varie componenti interagiscono

Architettura del calcolatore

◆ La prima decomposizione di un calcolatore è relativa a due macro-componenti:

- Hardware
- Software



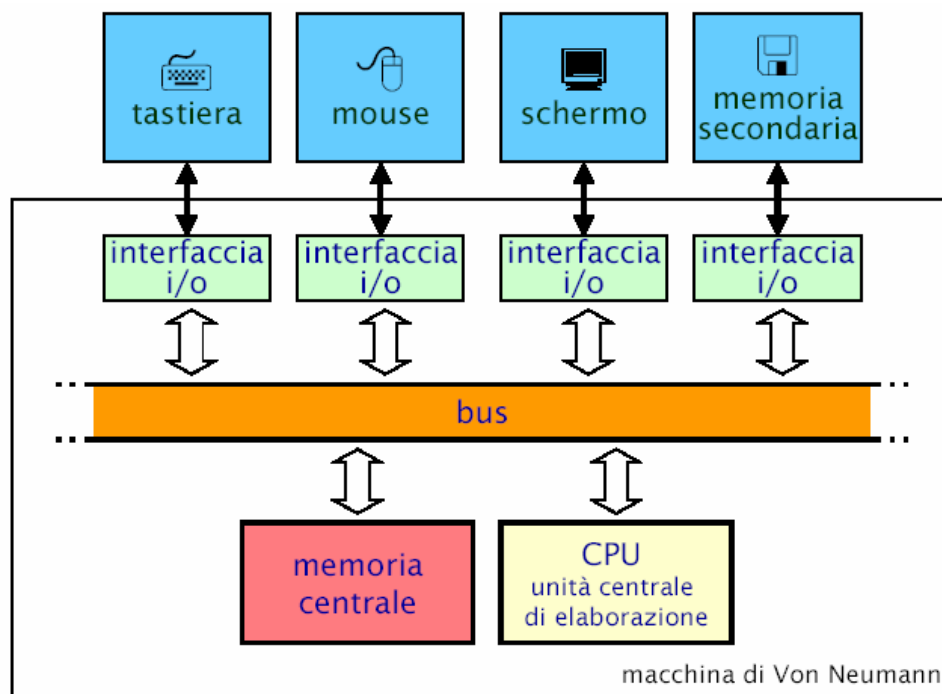
Hardware: architettura

- ◆ L'architettura dell'hardware di un calcolatore reale è molto complessa
- ◆ La **macchina di Von Neumann** è un modello semplificato dei calcolatori moderni
 - Von Neumann progettò, verso il 1945, il primo calcolatore con programmi memorizzabili anziché codificati mediante cavi e interruttori

Macchina di Von Neumann

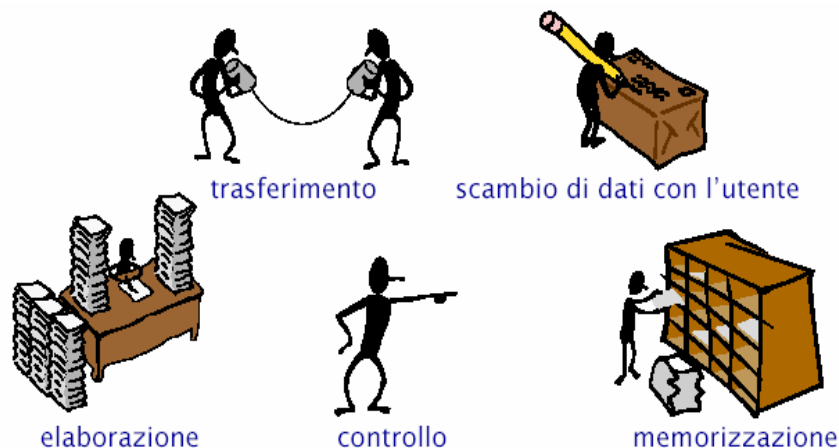
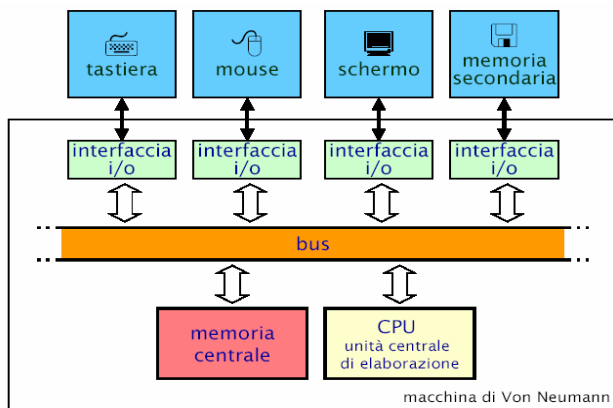
La macchina di Von Neumann è composta da 4 tipi di componenti funzionali:

- ◆ unità centrale di elaborazione (CPU – Central Processing Unit)
 - esegue istruzioni per l'elaborazione dei dati
 - svolge anche funzioni di controllo
- ◆ memoria centrale
 - memorizza e fornisce l'accesso a dati e programmi in esecuzione
- ◆ interfacce di ingresso e uscita
 - componenti di collegamento con le periferiche del calcolatore
- ◆ bus
 - svolge funzioni di trasferimento di dati e di informazioni di controllo tra le varie componenti funzionali



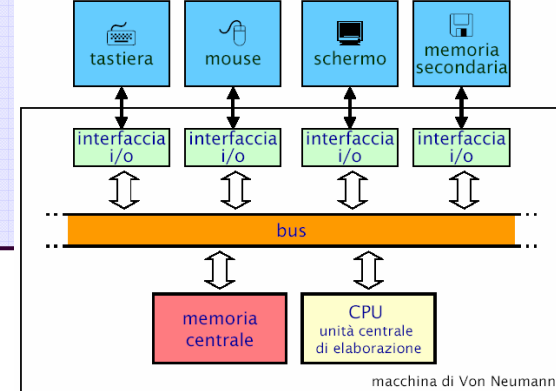
Macchina di Von Neumann

Ogni componente è specializzata nello svolgimento di una tipologia omogenea di funzionalità:



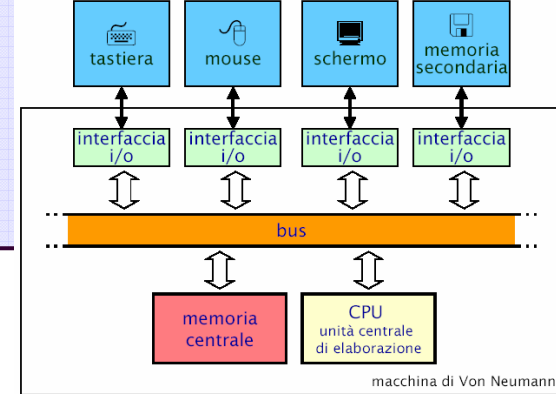
Eccezione: l'unità centrale di elaborazione svolge sia funzionalità di elaborazione sia di controllo

Trasferimento



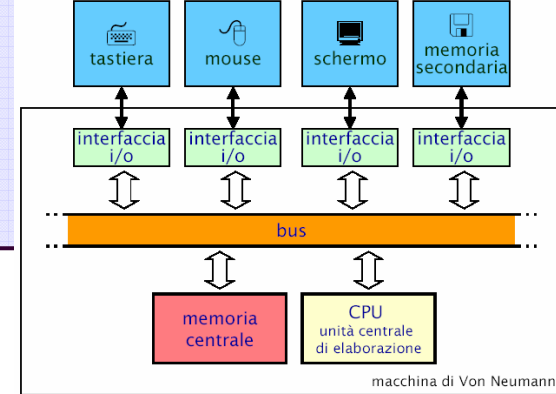
- ◆ Obiettivo: permettere lo scambio di informazioni tra le varie componenti funzionali del calcolatore
 - trasferimento dei dati e delle informazioni di controllo
- ◆ Due possibili soluzioni
 - collegare ciascun componente con ogni altro componente
 - collegare tutti i componenti a un unico canale (**bus**)
- ◆ L'utilizzo di un bus favorisce la modularità e l'espandibilità del calcolatore

Periferiche e interfacce di ingresso-uscita (I/O)



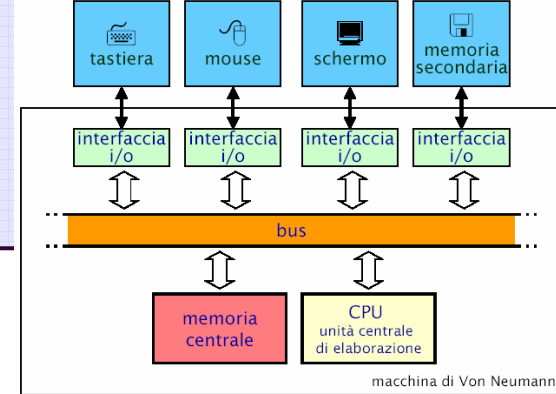
- ◆ Un calcolatore può essere collegato a vari dispositivi di ingresso e/o uscita (**periferiche**)
 - esempi: tastiera, mouse, schermo, stampanti, modem
 - anche le memorie di massa (es., unità disco e lettore di CD-ROM) sono considerate periferiche
- ◆ Nella macchina di Von Neumann, le periferiche non fanno parte del calcolatore
 - ogni periferica è controllata con un'opportuna **interfaccia**
 - una interfaccia ha il compito di tradurre i segnali interni del calcolatore in un formato comprensibile alla periferica stessa, e viceversa

Memorizzazione



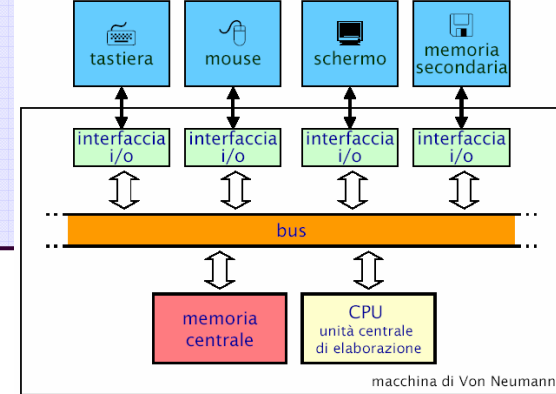
- ◆ Un calcolatore memorizza
 - i dati, che rappresentano informazioni di interesse
 - i programmi per l'elaborazione dei dati
- ◆ Unità responsabile della memorizzazione: **memoria**
 - una unità di memoria è organizzata in celle
 - a ciascuna cella è associato un indirizzo, che la identifica
 - ciascuna cella è in grado di memorizzare un singolo dato o una singola istruzione
- ◆ Un'unità di memoria fornisce due sole operazioni
 - memorizzazione di un valore in una cella (**scrittura**)
 - accesso al valore memorizzato in una cella (**lettura**)

CPU



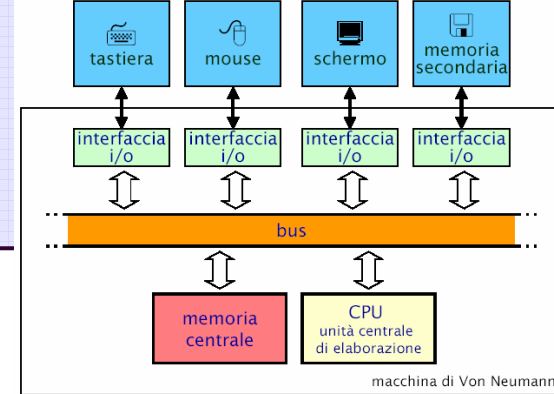
- ◆ L'**unità centrale di elaborazione** (CPU) o **processore** presiede all'esecuzione di un programma
 - il programma è memorizzato in memoria centrale come sequenza di istruzioni del linguaggio macchina
 - *Linguaggio macchina* o *assembly* è il linguaggio per cui la CPU si comporta da esecutore
- ◆ La CPU è composta da
 - unità di controllo
 - unità aritmetico-logica

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 imposta dall'orologio di sistema (**clock**)

Elaborazione



- ◆ L'elaborazione è svolta dall'**unità aritmetico-logica**, che è un componente dell'unità centrale di elaborazione
- ◆ Le istruzioni del linguaggio macchina corrispondono ad operazioni elementari di elaborazione
 - operazioni aritmetiche
 - operazioni relazionali (confronto tra dati)
 - operazioni su caratteri e valori di verità
 - altre operazioni numeriche
- ◆ 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

Componenti e funzionamento del calcolatore

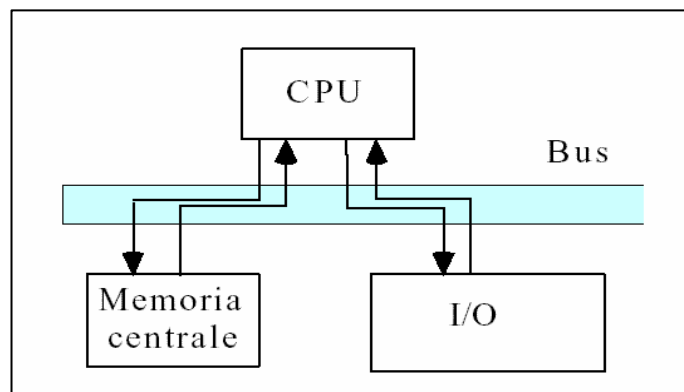
BUS di sistema

Caratteristiche del collegamento a BUS

- ◆ Semplicità
 - un'unica linea di connessione → costi ridotti di produzione
- ◆ Estendibilità
 - aggiunta di nuovi dispositivi molto semplice
- ◆ Standardizzabilità
 - regole per la comunicazione da parte di dispositivi diversi
- ◆ Lentezza
 - utilizzo in **mutua esclusione** del bus
- ◆ Limitata capacità
 - al crescere del numero di dispositivi collegati
- ◆ Sovraccarico del processore (CPU)
 - perchè funge da *master* sul controllo del bus

Bus di sistema

- ◆ Interconnette le componenti interne del calcolatore
 - CPU, memoria ed interfacce a periferiche (I/O, memoria di massa, ...)
- ◆ Collega due unità funzionali alla volta
 - una trasmette e l'altra riceve
- ◆ Funzionamento master/slave
 - la CPU (master) seleziona la connessione da attivare e ordina il trasferimento dei dati



Bus di sistema

- ◆ Il bus trasporta dati, indirizzi e comandi
- ◆ Componenti del bus (sottogruppi di linee):
 - Bus dati (data bus)
 - Bus indirizzi (address bus)
 - Bus comandi (command bus)
- ◆ **Bus dati** (data bus)
 - Serve per trasferire dati tra:
 - memoria centrale e registro dati (MDR) della CPU
 - periferiche e CPU (o memoria centrale)
 - Bidirezionale

Bus di sistema

◆ Bus indirizzi (address bus)

- Serve per trasmettere il contenuto del registro indirizzi (MAR) alla memoria (o a una periferica)
- si seleziona una cella per successive operazioni di lettura o scrittura
- Unidirezionale

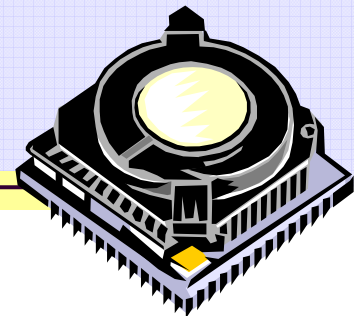
◆ Bus comandi (command bus)

- Serve per inviare comandi
 - verso la memoria (es: lettura o scrittura)
 - o verso una periferica (es.: stampa)
- Unidirezionale

Componenti e funzionamento del calcolatore

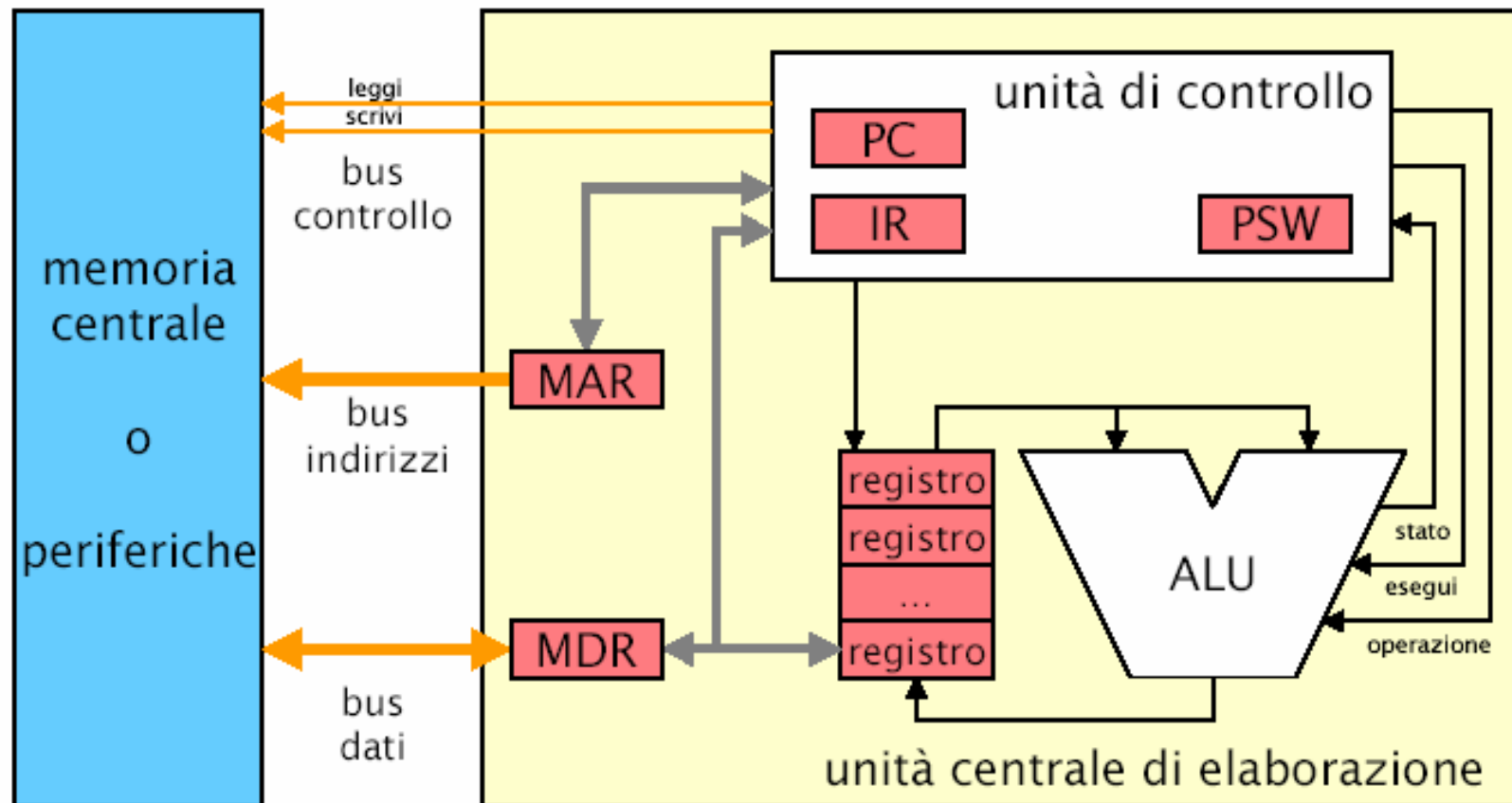
CPU

Unità centrale di elaborazione



- ◆ L'unità centrale di elaborazione è realizzata fisicamente sotto forma di **microprocessore**
 - i microprocessori sono dispositivi elettronici molto complessi
 - Es: un Pentium II è composto da oltre 7 milioni di transistor in un singolo circuito integrato (chip)
- ◆ L'unità centrale di elaborazione è costituita da
 - Unità di controllo
 - Unità Aritmetico-Logica
 - Registri

Struttura del microprocessore



Elementi di una CPU



◆ Unità di controllo

- Svolge funzioni di controllo, decide quali istruzioni eseguire.

◆ Unità aritmetico-logica

- esegue le operazioni aritmetico-logiche (+, -, ecc. , confronto).

◆ Registri

- **memoria ad alta velocità** usata per risultati temporanei e informazioni di controllo;
- il **valore massimo** memorizzabile in un registro è determinato dalle **dimensioni** del registro;
- esistono registri di uso generico e registri specifici:

Registri

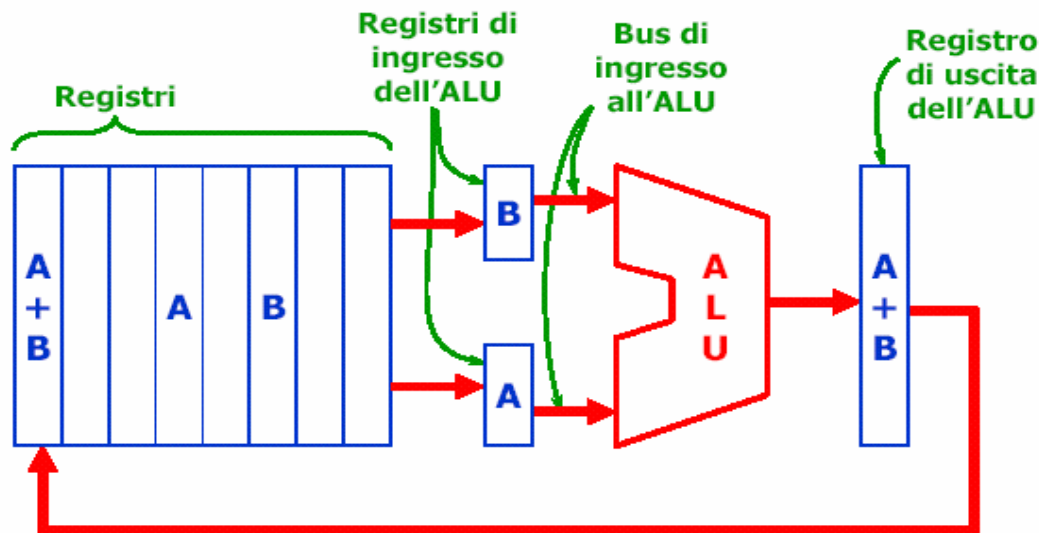
- ◆ Esistono registri di uso generico e registri specifici :
 - **PC**: contatore delle istruzioni (program counter)
 - contiene l'indirizzo della prossima istruzione da eseguire
 - **IR**: registro delle istruzioni (instruction register)
 - contiene l'istruzione che deve essere eseguita
 - **PSW**: parola di stato del processore
 - contiene informazioni, opportunamente codificate, sull'esito dell'ultima istruzione che è stata eseguita

Registri

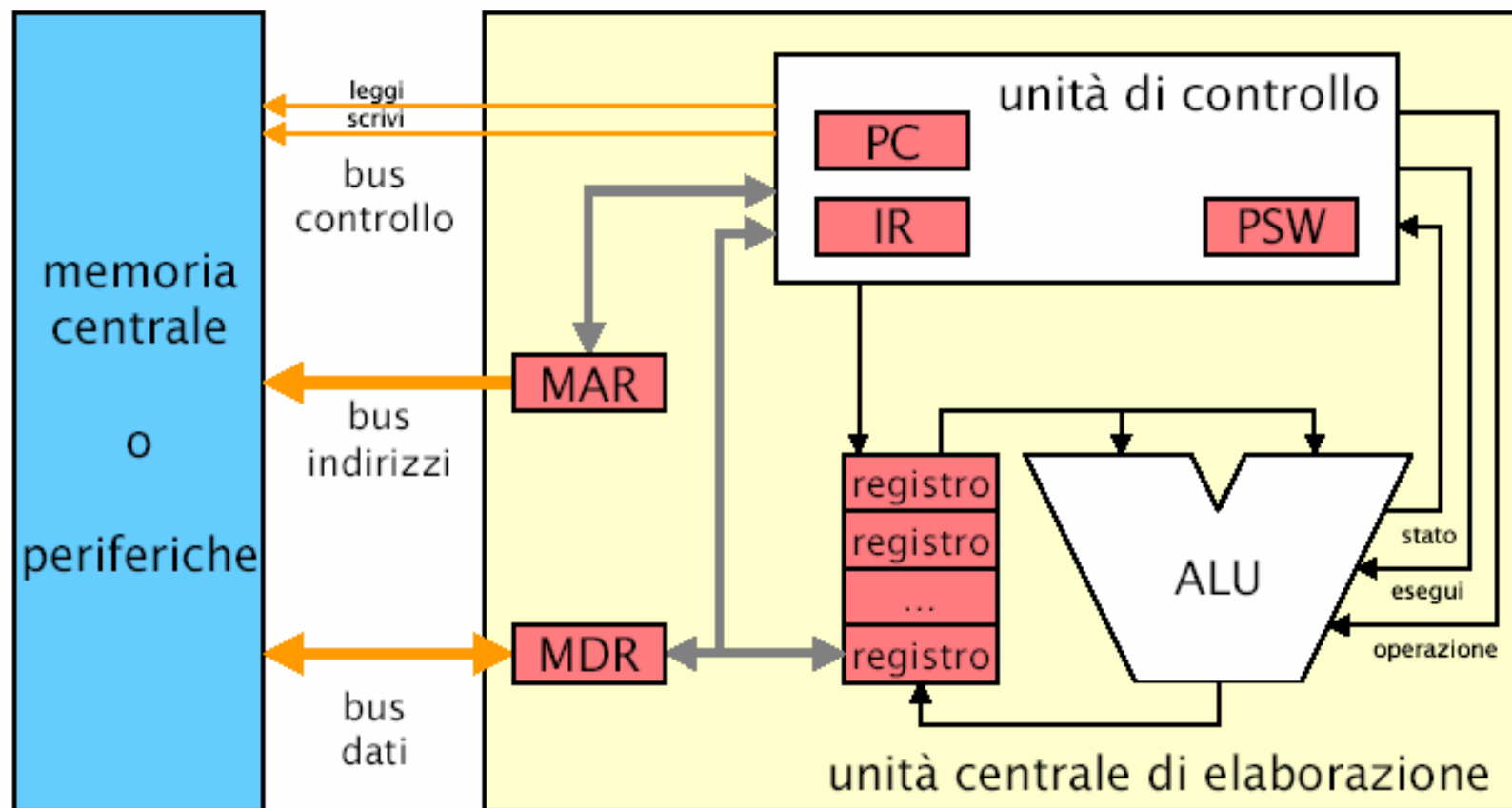
- **MAR:** registro indirizzi della memoria
 - indirizzo della cella di memoria che deve essere acceduta o memorizzata
- **MDR:** registro dati della memoria
 - dato che è stato acceduto o che deve essere memorizzato
- **registri generali**
 - per memorizzare gli operandi ed il risultato di una operazione

Unità 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



Struttura del microprocessore



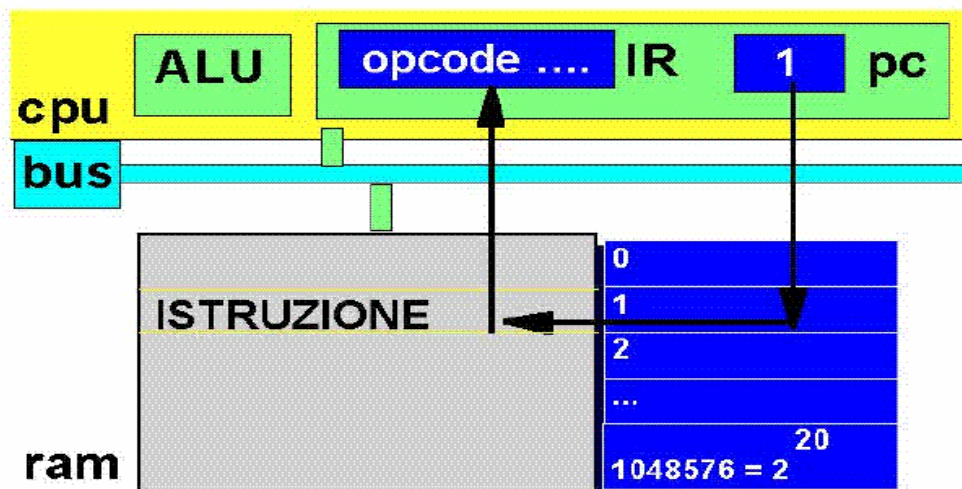
Ciclo di esecuzione delle istruzioni

- ◆ La CPU esegue un'istruzione mediante le tre seguenti operazioni di base:
 - **Fetch** (lettura)
 - **Decode** (decodifica)
 - **Execute** (esecuzione)
- ◆ Un programma è eseguito applicando ad ogni istruzione la sequenza fetch-decode-execute
 - *ciclo di esecuzione dell'istruzione* o *ciclo macchina* o *ciclo fetch-decode-execute*

Ciclo fetch-decode-execute

1) FETCH:

- si accede alla prossima istruzione, riferita dal registro contatore dell'istruzione (PC)
- si porta tale istruzione dalla memoria centrale al Registro Istruzioni (IR)



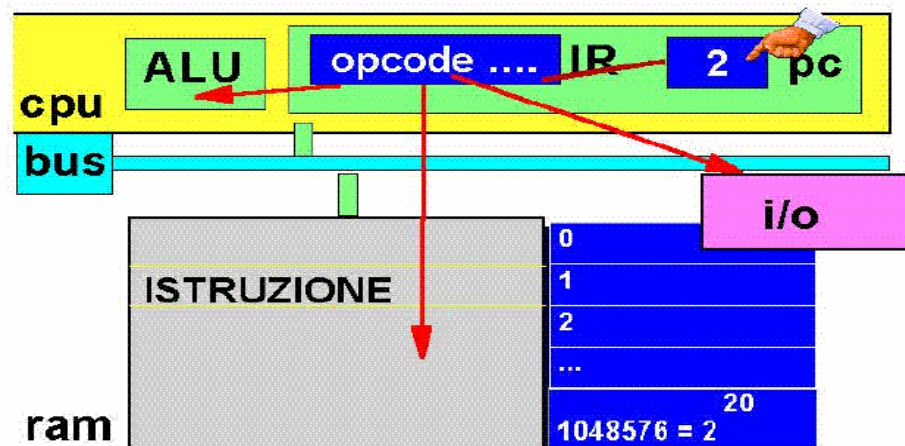
Ciclo *fetch-decode-execute*

2) **DECODE**: decodifica dell'istruzione

- si individua il tipo dell'operazione e gli operandi (dati) usati
- si trasferiscono i dati nei registri opportuni

3) **EXECUTE**: esecuzione dell'istruzione

- si incrementa il registro contatore dell'istruzione (PC)
- ciascuna azione viene richiesta al componente opportuno



Istruzioni del linguaggio macchina

- ◆ Istruzioni per l'**elaborazione dei dati**
 - aritmetiche
 - logiche (AND, OR, NOT)
 - relazionali (maggiore, minore, uguale, ...)
- ◆ Istruzioni per il **controllo del flusso** di esecuzione
 - sequenza
 - selezione
 - ciclo
 - salto (*Jump*) ad una data istruzione
- ◆ Istruzioni per il **trasferimento** di informazioni
 - dati ed istruzioni fra CPU e memoria
 - dati fra CPU e dispositivi di I/O (tramite interfacce)

Clock

- ◆ L'orologio interno (**clock**) del microprocessore emette un segnale di sincronizzazione per tutto il sistema
 - si misura in cicli/secondo [Hz]
 - $400 \text{ MHz} = 400 \times 10^6 \text{ Hz} = 4 \times 10^8 \text{ Hz}$
 \Rightarrow ciclo è eseguito in $2.5 \times 10^{-9} \text{ s} = 2.5 \text{ ns}$
- ◆ Ad ogni impulso di clock si esegue un ciclo macchina
 - la velocità di un microprocessore dipende dalla frequenza del suo clock, ma non solo poiché l'esecuzione di un'istruzione può richiedere più cicli

Caratteristiche dei microprocessori

- ◆ Repertorio di istruzioni
 - L'insieme delle istruzioni che costituiscono il linguaggio macchina del processore (CISC, RISC, C-RISC)
- ◆ Frequenza di clock
- ◆ Ampiezza del bus
 - numero di bit nel bus interno del processore
- ◆ Co-processor
 - processori specializzati per operazioni complesse
 - (es: co-processore matematico)
- ◆ Memoria cache
 - memoria veloce interna al processore, che consente di accedere più velocemente ai dati da elaborare

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

Componenti e funzionamento del calcolatore

Memoria centrale

Memoria centrale

- ◆ La **memoria centrale** (o **principale**) è la componente in cui si immagazzinano e da cui si accedono dati e programmi
- ◆ È l'unico tipo di memoria che può essere acceduto direttamente dal processore
 - è costituita da **celle** (o **locazioni**)
 - ogni cella può contenere una quantità fissata di memoria (numero di bit), detta **parola**

Memoria centrale

- ◆ Ogni cella è caratterizzata da
 - **indirizzo**, un numero che identifica la cella e ne consente l'accesso
 - **valore**, la sequenza di bit memorizzata in essa
- ◆ La memoria fornisce le operazioni di:
 - **lettura**: consultazione del valore di una cella con un dato indirizzo
 - **scrittura**: modifica del valore di una cella con un dato indirizzo

Memoria centrale

- ◆ Le operazioni avvengono sotto il controllo della CPU

- La CPU seleziona una particolare cella di memoria ponendone l'indirizzo nel **Registro Indirizzi (MAR)**

- ◆ Se il Registro Indirizzi (MAR) è costituito da N bit, si possono indirizzare 2^N celle di memoria, da 0 a $2^N - 1$

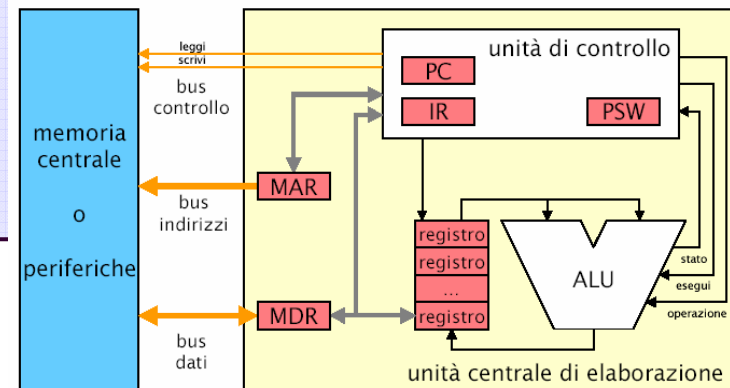
- Nei PC attuali il MAR è almeno di 32 bit

- ◆ Operazione di **lettura**:

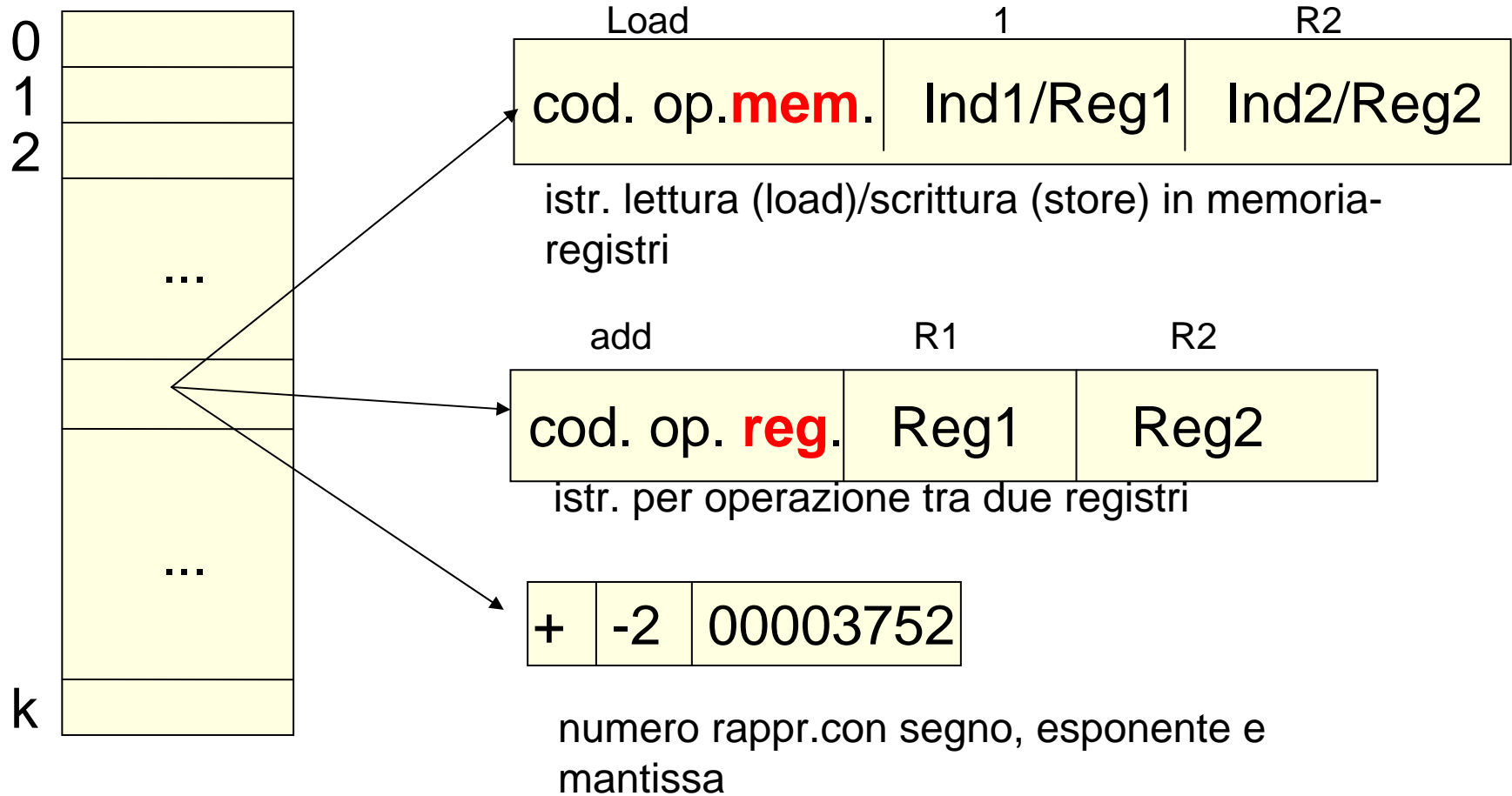
- copia nel Registro Dati (MDR) il contenuto della cella di memoria indirizzata dal Registro Indirizzi (MAR)

- ◆ Operazione di **scrittura (store)**

- copia il contenuto del Registro Dati (MDR) nella cella di memoria indirizzata dal Registro Indirizzi (MAR)



Memoria centrale

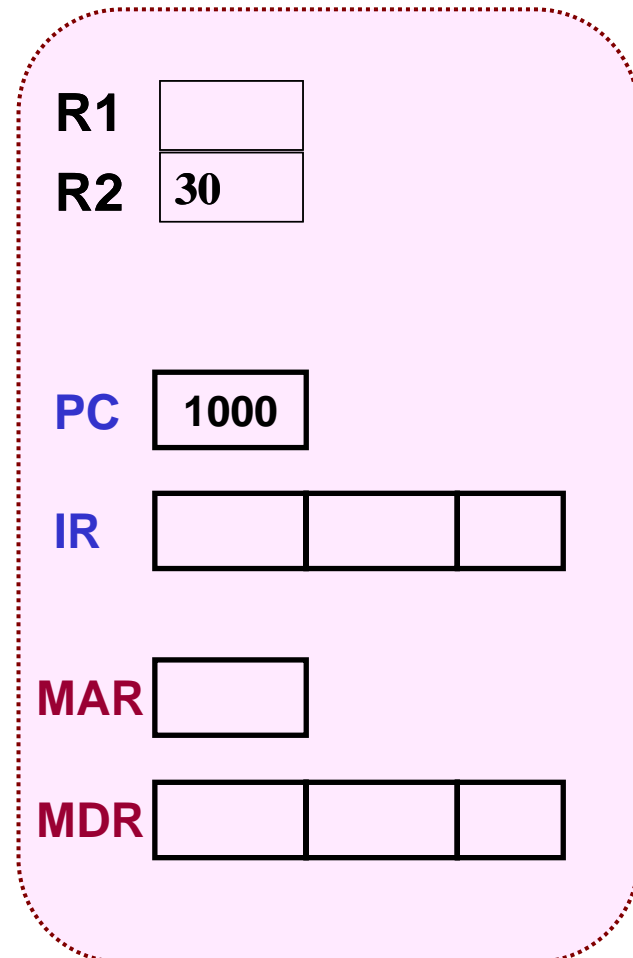


Esempio di esecuzione di un programma in linguaggio macchina

Registri CPU

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

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

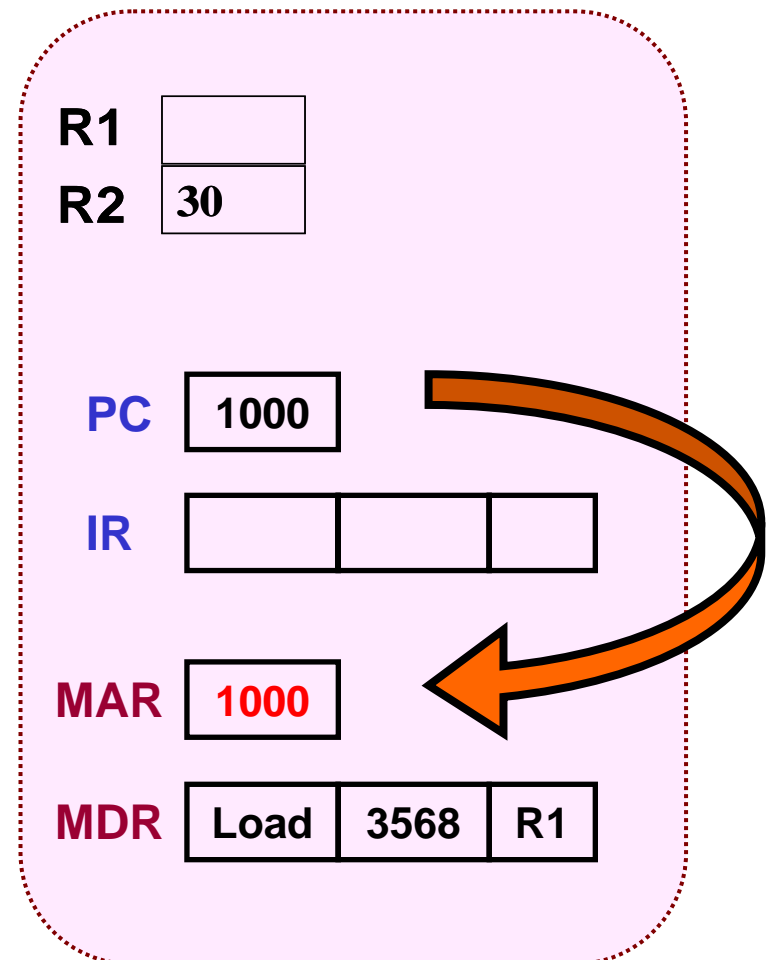


Esecuzione istruzione *1000*: fetch

Registri CPU

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

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



Esecuzione istruzione *1000* (2): fetch

1000

Load	3568	R1
-------------	-------------	-----------

1001

Add	R1	R2
------------	-----------	-----------

1002

Store	R1	3568
--------------	-----------	-------------

1003

Jump	1000
-------------	-------------

3568

44

Registri CPU

R1

R2

30

PC

1000

IR

Load	3568	R1
-------------	-------------	-----------

MAR

1000

MDR

Load	3568	R1
-------------	-------------	-----------

Esecuzione istruzione **1000** (3): 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**

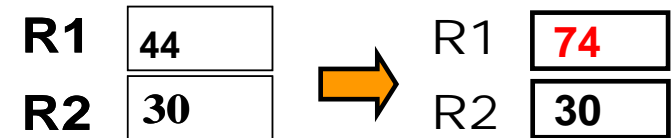
MAR **1000**

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

1000	Load	3568	R1	R1	74
1001	Add	R1	R2	R2	30
1002	Store	R1	3568		
1003	Jump	1000			

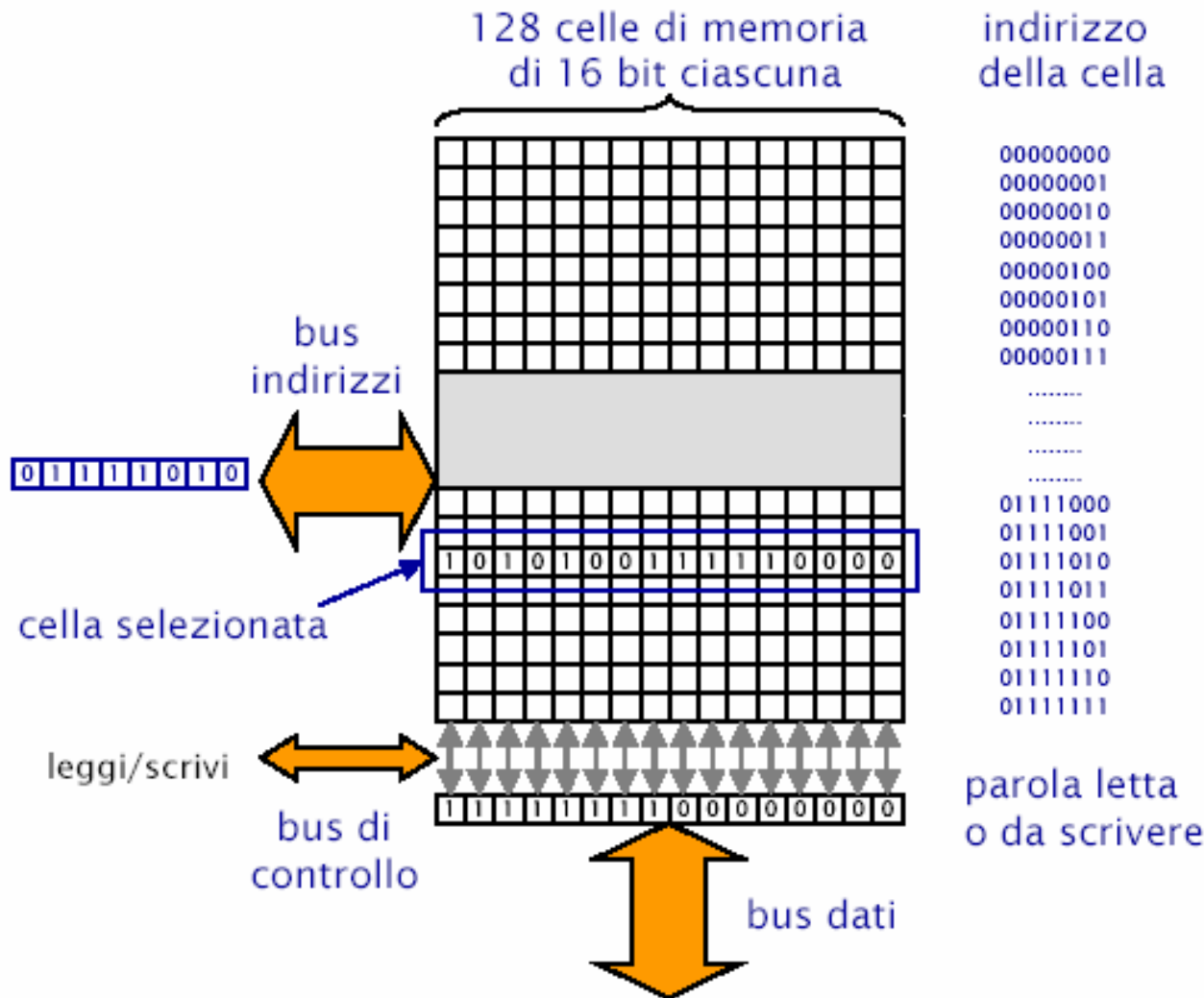
3568	74
------	----

Esecuzione istruzione *1003*

1000	Load	3568	R1	<div>R174</div> <div>R230</div>
1001	Add	R1	R2	
1002	Store	R1	3568	
1003	Jump	1000		

3568	74
------	----

Struttura semplificata della memoria centrale



Memorie RAM e memorie ROM

- ◆ Le memorie RAM (random access memory)
 - possono essere accedute sia in lettura che in scrittura
 - sono volatili
 - i dati memorizzati vengono persi allo spegnimento del calcolatore
- ◆ La memorie ROM (read only memory)
 - sono persistenti
 - mantengono il contenuto anche senza alimentazione
 - permettono solo la lettura dei dati (o programmi)
 - memorizzano alcuni programmi di sistema (firmware)
- ◆ Evoluzioni delle memorie ROM:
 - PROM (scritte una sola volta)
 - EPROM (scritte più volte)

Memoria cache

- ◆ Memoria “intermedia” fra registri e RAM
 - La RAM ha tempi di accesso molto alti rispetto alla velocità dei microprocessori e ne ritarda l’elaborazione
- ◆ Memorizza il contenuto di celle della RAM che potrebbero essere acceduti nuovamente dalla CPU
 - sfrutta la località dei programmi (90%-10%)
- ◆ Strategia di utilizzo:
 - la prima volta che la CPU carica dati dalla memoria centrale, questi sono caricati anche sulla cache
 - le volte successive, i dati possono essere letti dalla cache invece che dalla memoria centrale (più lenta)
- ◆ Tipi di memoria cache:
 - cache di I° livello: contenuta nel microprocessore
 - cache di II° livello: aggiungibile successivamente