

# Principi operativi dei computer

Capitolo 7

Fluency – Conoscere e usare l'informatica

# I soliti sospetti

Processore

Sistema operativo

Software

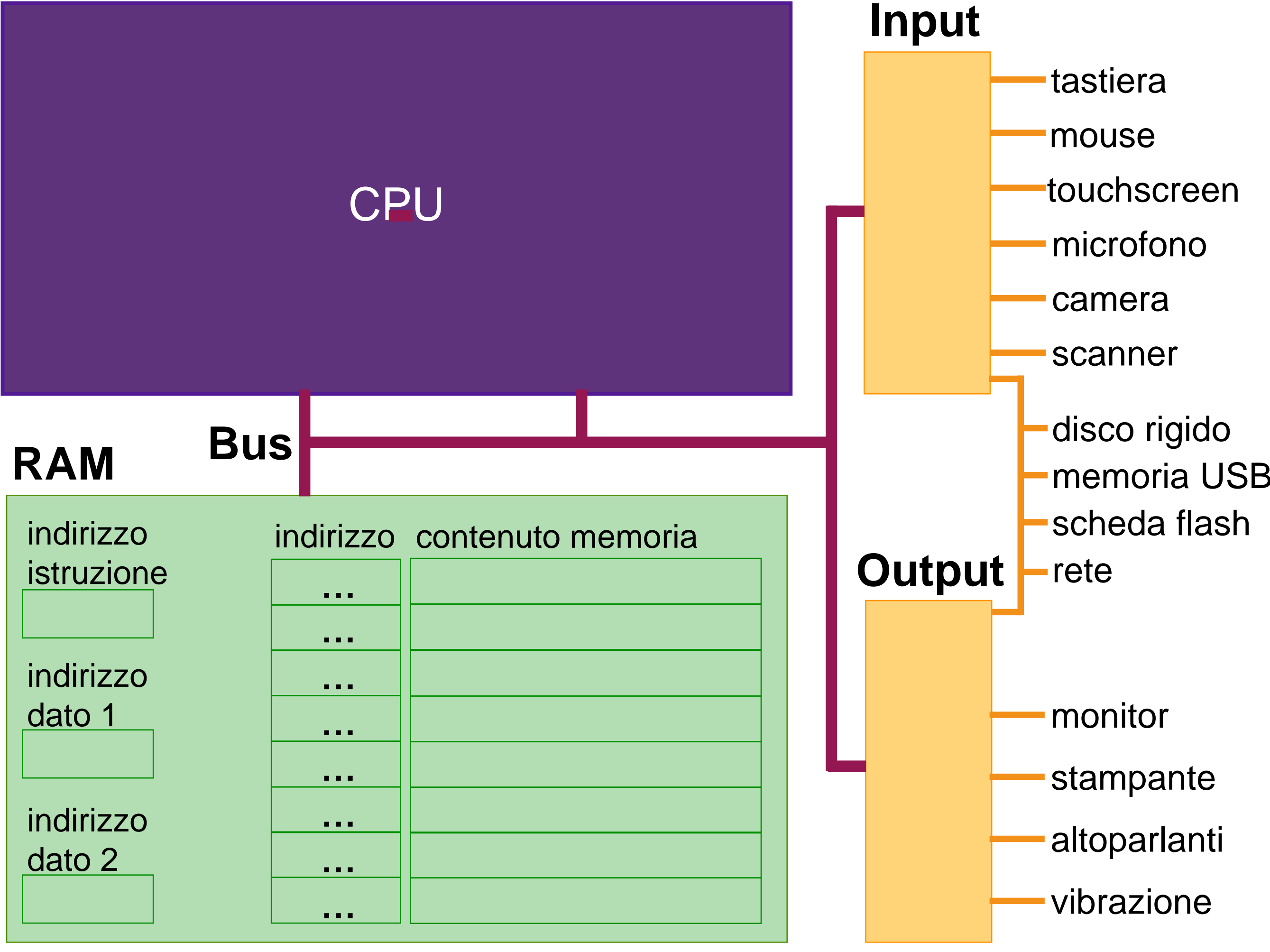
Istruzioni

Ciclo macchina

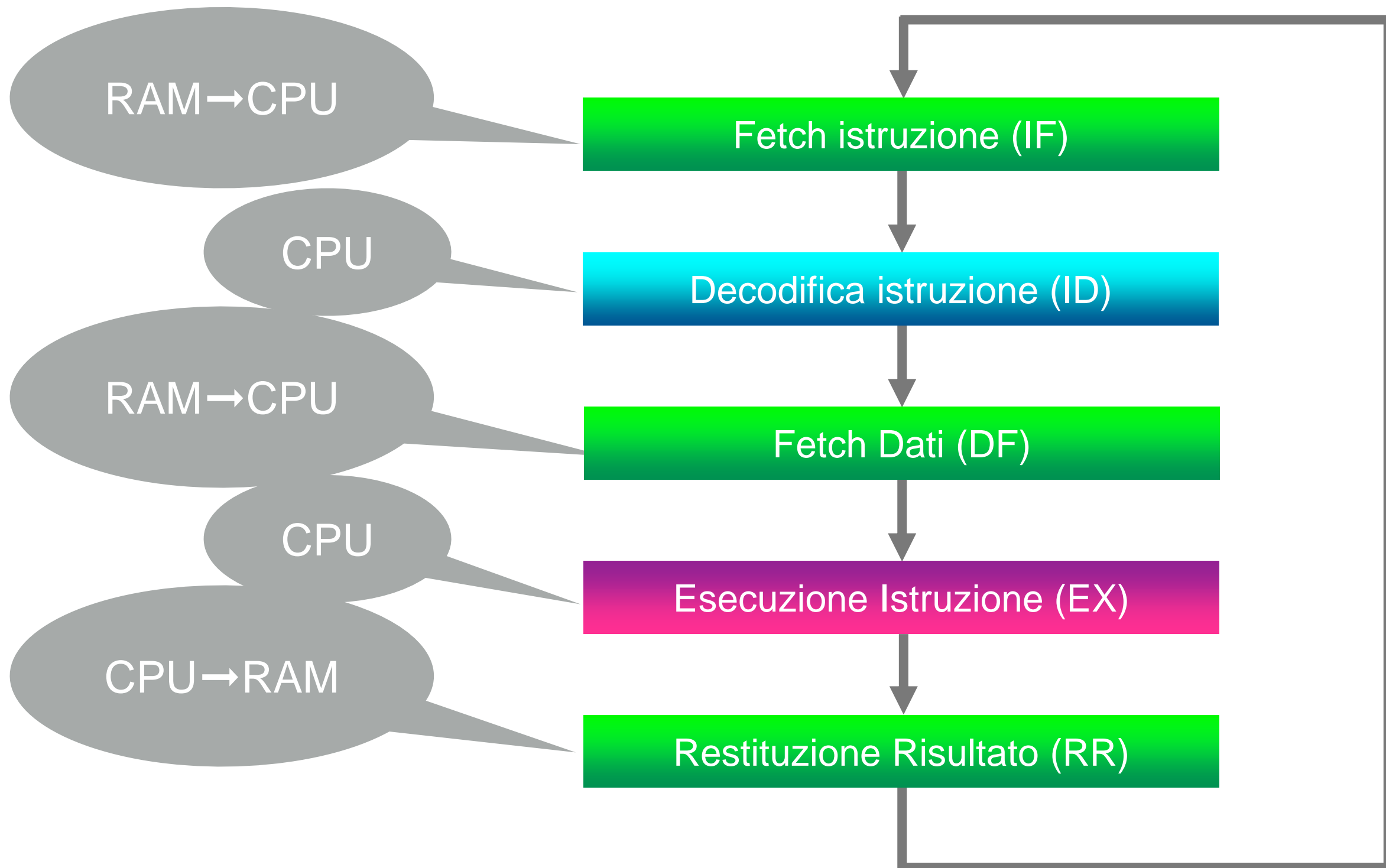
Memoria

Hardware componente

Com'è fatto un computer?



# Ciclo Macchina



# RAM

indirizzo istruzione	indirizzo	contenuto memoria
	...	
	...	
	...	
indirizzo dato 1	...	
	...	
	...	
indirizzo dato 2	...	
	...	
	...	

# RAM

Contiene

il programma in esecuzione

i dati su cui il programma stesso opera

# Caratteristiche della RAM

Locazioni discrete

ogni locazione è di 1 byte

Indirizzi

univoco per ogni locazione (intero a partire da 0)



# Caratteristiche della RAM

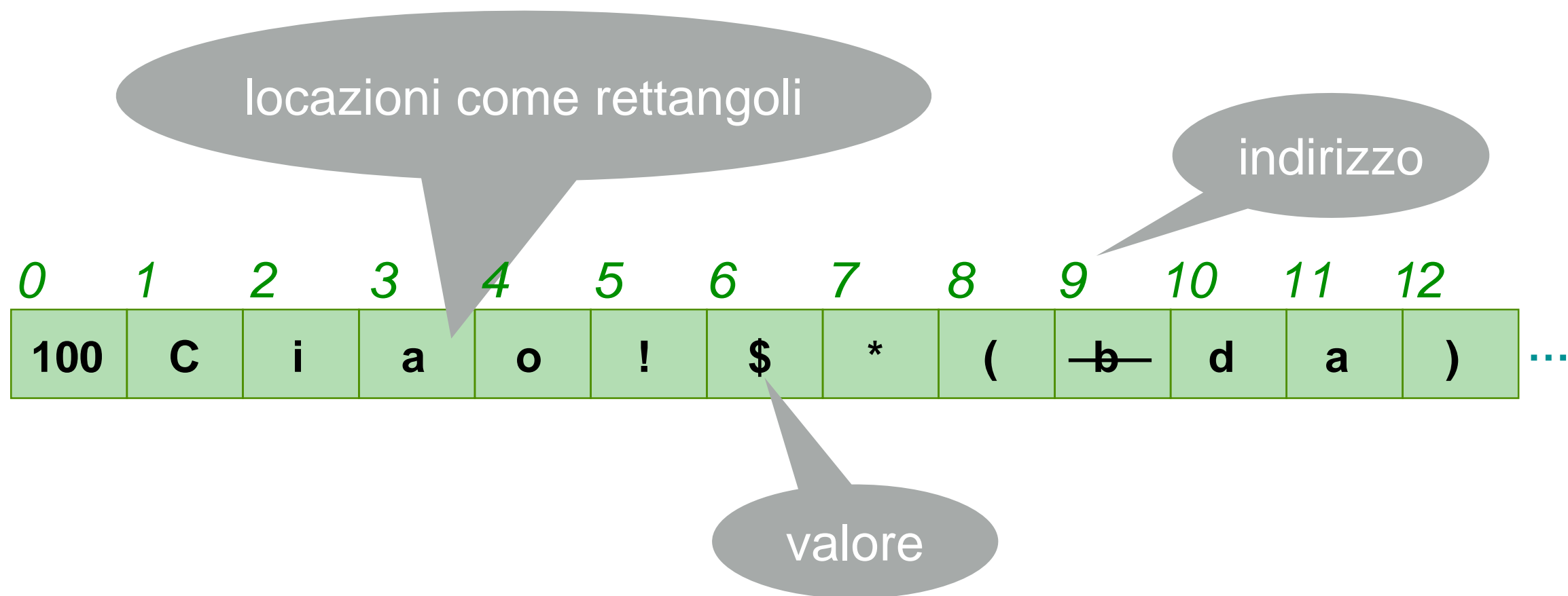
Valori

le locazioni di memoria memorizzano valori

Capacità finita

sia nel numero di locazioni

sia nella capacità di ogni locazione



# Locazioni da un byte

Ogni locazione contiene

una sequenza di 8 bit

Parola di memoria

blocchi di 4 byte usati come singola unità

# RAM

Accesso Causale o diretto

il computer può accedere direttamente a qualsiasi locazione di memoria

Ordine di grandezza: gigabyte (GB)

Avere molta memoria è preferibile

evita problemi di spazio per i programmi e i dati

# Controllo

istruzione

○ IF

○ ID

○ DF

○ EX

○ RR

program  
counter

```
graph LR; IF[IF] --> ID[ID]; ID --> DF[DF]; DF --> EX[EX]; EX --> RR[RR]; RR --> IF; PC[program counter] --> IF;
```

# ALU

dato 1	
dato 2 [   ]	
risultato	
indirizzo risult.	

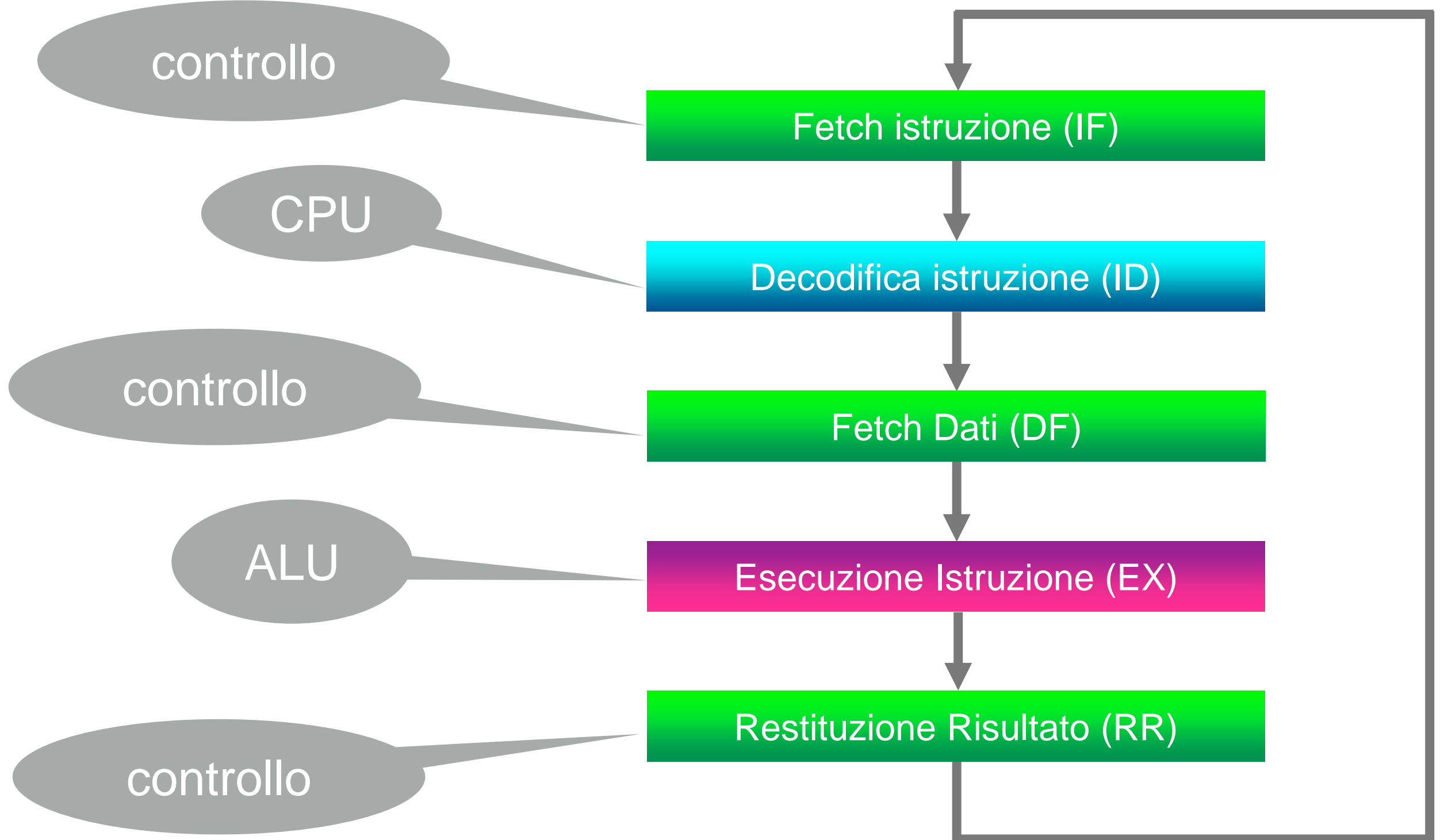
# RAM

indirizzo istruzione	indirizzo	contenuto memoria
	...	
	...	
indirizzo dato 1	...	
	...	
	...	
indirizzo dato 2	...	
	...	
	...	

# CPU

Controllo + ALU

Implementa il ciclo macchina via hardware



# Istruzione esempio



riferimento  
indiretto

**ADD 4000, 2000, 2080**

somma i numeri nelle locazioni 2080 e 2000

scrive il risultato nella locazione di memoria 4000



# Istruzione esempio

Il passo di Fetch Dati

deve estrarre i due valori

Il passo Restituzione Risultato

inserirà la somma nella locazione 4000

# ALU

Arithmetic Logic Unit

Esegue tutti i calcoli

il passo esecuzione istruzione

Circuito della ALU per la somma

Si sono altri circuiti

dedicati alla moltiplicazione, al confronto, ...

# Risultati operazione

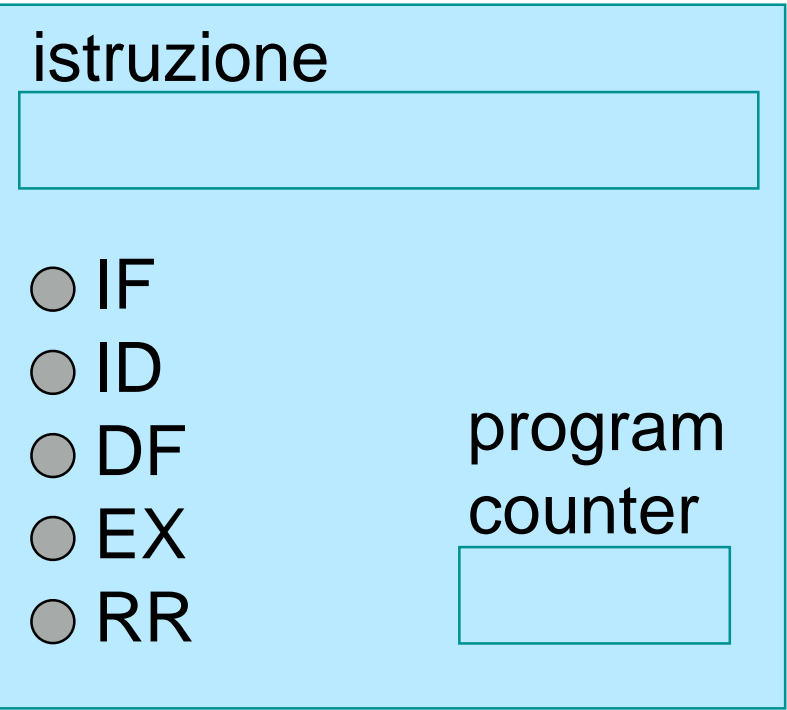
**ADD 4000, 2000, 2080**

$$\begin{array}{c} 2000 \\ \boxed{48} \end{array} + \begin{array}{c} 2080 \\ \boxed{2} \end{array} \longrightarrow \begin{array}{c} 4000 \\ \boxed{50} \end{array}$$

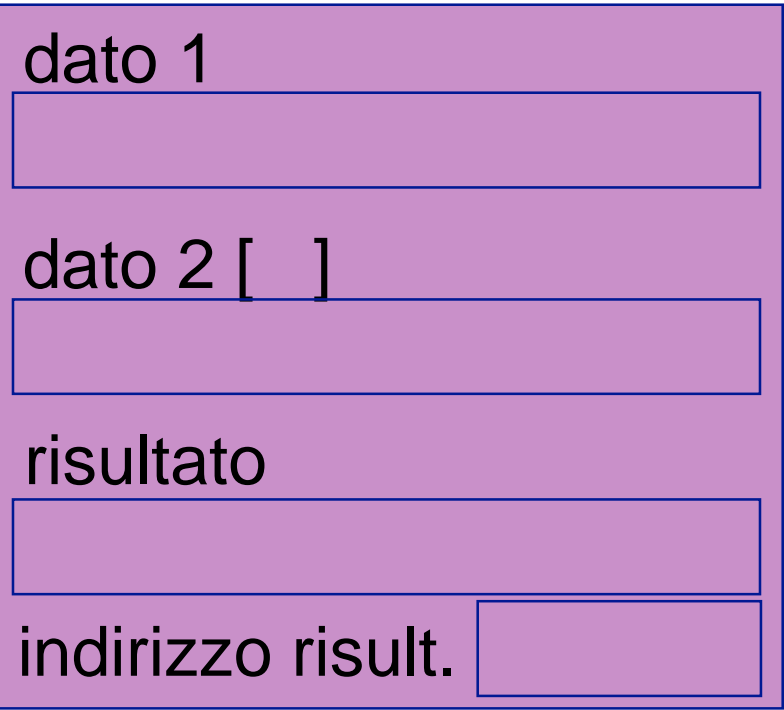
$$\begin{array}{c} 2000 \\ \boxed{9} \end{array} + \begin{array}{c} 2080 \\ \boxed{0} \end{array} \longrightarrow \begin{array}{c} 4000 \\ \boxed{9} \end{array}$$

$$\begin{array}{c} 2000 \\ \boxed{14} \end{array} + \begin{array}{c} 2080 \\ \boxed{14} \end{array} \longrightarrow \begin{array}{c} 4000 \\ \boxed{28} \end{array}$$

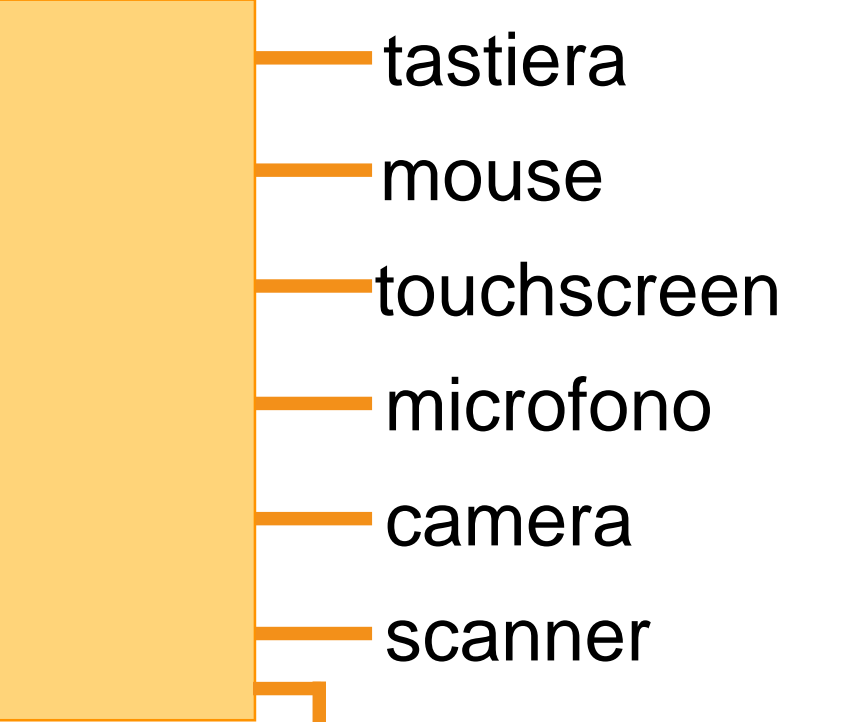
# Controllo



# ALU



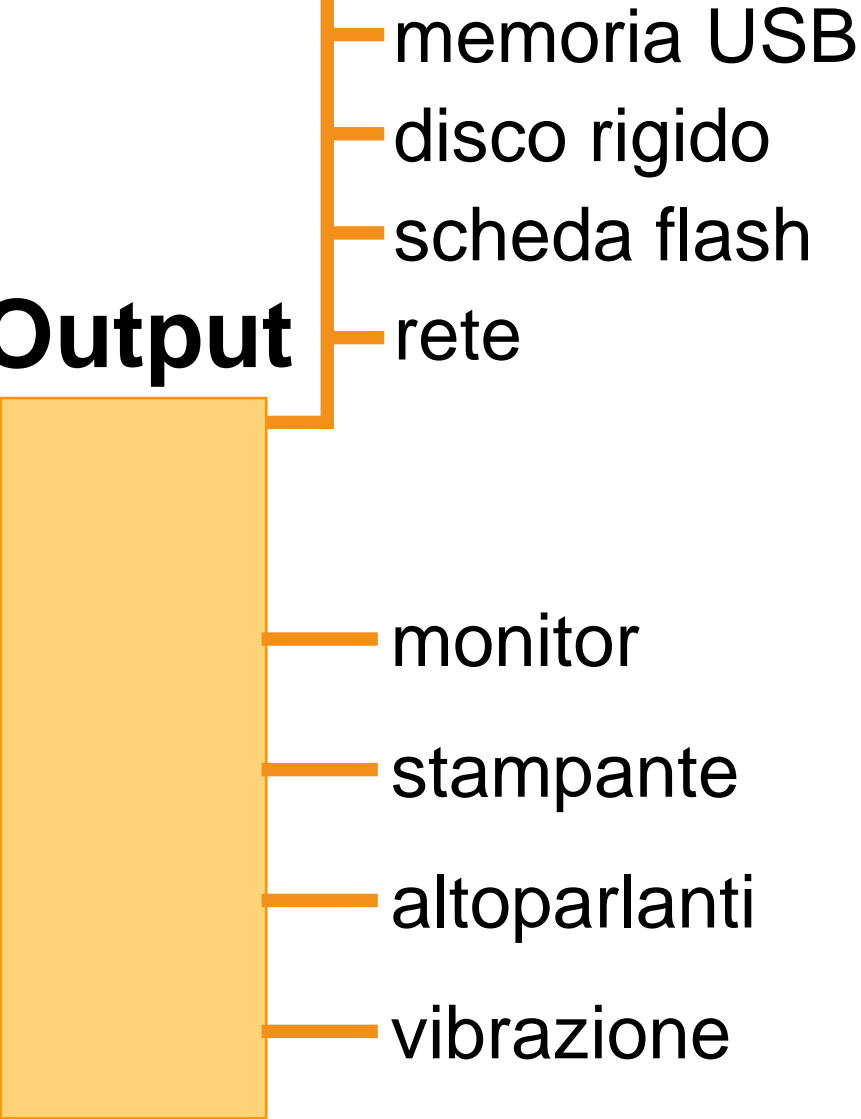
# Input



# RAM

indirizzo istruzione	indirizzo	contenuto memoria
<div></div>	...	
	...	
indirizzo dato 1	...	
<div></div>	...	
	...	
indirizzo dato 2	...	
<div></div>	...	

# Output



# Input e output

Le periferiche

si collegano alle porte di input/output

non sono parti vere e proprie del computer

specializzate per codificare o decodificare i dati scambiati col mondo esterno

# Hard disk

Periferica alfa

Chiavi USB

hanno

archivio

vita “attesa”

Obsolescenza della  
tecnologia

# Le periferiche

La tastiera

trasforma le battute sui tasti in formato binario

Il monitor

rappresenta i dati contenuti nella memoria video

# Driver delle periferiche

Le periferiche sono “stupide”

Driver

traduzione tra fenomeno fisico e segnale binario

Il computer fa tutto il resto

interpreta il segnale binario riportato dalla periferica

prepara l'output



# Controllo

istruzione

IF

ID

DF

EX

RR

program counter

# ALU

dato 1

dato 2 [   ]

risultato

indirizzo risult.

# Input

# RAM

# Bus

indirizzo istruzione

indirizzo dato 1

indirizzo dato 2

indirizzo

...

...

...

...

...

...

...

...

...

...

contenuto memoria

# Output

# BUS

Tutti i trasferimenti di dati passano attraverso il bus

Come una autostrada a più corsie

ampiezza

velocità

“Collo di bottiglia” delle prestazioni del computer

# Un “PC” nel PC

Program Counter

Qual'è la prossima l'istruzione da eseguire?

l'indirizzo è memorizzato nell'unità di controllo

# Aggiornamento del PC

Al fetch di una nuova istruzione

il PC è incrementato di 4

Al prossimo fetch

il PC “punta” all’istruzione giusta

# Decodifica delle istruzioni

Esecuzione di un programma

il computer interpreta i nostri comandi

espressi nel suo proprio linguaggio

# ADD 800, 428, 884

somma

indirizzo 1<sup>o</sup>  
addendo

Esecuzione di ADD 800, 428, 884

indirizzo  
risultato

indirizzo 2<sup>o</sup>  
addendo

# Controllo

istruzione

- ☐ IF
- ☐ ID
- ☐ DF
- ☐ EX
- ☐ RR

program  
counter

**800**

# ALU

dato 1

dato 2 [   ]

risultato

indirizzo result.

# Input

# RAM

Bus

indirizzo  
istruzione

**800**

indirizzo  
dato 1

indirizzo  
dato 2

indirizzo	contenuto memoria
...	
800	<b>ADD 4000, 2000, 2080</b>
...	
2000	<b>30</b>
...	
2080	<b>12</b>
...	
4000	

# Output

# Controllo

istruzione

**ADD 4000, 2000, 2080**

- IF
- ID
- DF
- EX
- RR

program  
counter

**800**

# ALU

dato 1

dato 2 [   ]

risultato

indirizzo risult.

# Input

# RAM

## Bus

indirizzo  
istruzione

**800**

indirizzo  
dato 1

indirizzo  
dato 2

indirizzo    contenuto memoria

...

**800**

...

**2000**

...

**2080**

...

**4000**

**ADD 4000, 2000, 2080**

**30**

**12**

# Output



# Controllo

istruzione

**ADD 4000, 2000, 2080**

- ☐ IF
- ☒ ID
- ☐ DF
- ☐ EX
- ☐ RR

program  
counter

**800**

# ALU

dato 1

dato 2 [ **+** ]

risultato

indirizzo result.

**4000**

# Input

# RAM

## Bus

indirizzo  
istruzione

indirizzo  
dato 1

**2000**

indirizzo  
dato 2

**2080**

indirizzo

...

800

...

2000

...

2080

...

4000

contenuto memoria

**ADD 4000, 2000, 2080**

**30**

**12**

# Output

# Controllo

istruzione

- ☐ IF
- ☐ ID
- ☒ DF
- ☐ EX
- ☐ RR

program  
counter  
**804**

# ALU

dato 1

**30**

dato 2 [**+**]

**12**

risultato

indirizzo result.

**4000**

# Input

# RAM

## Bus

indirizzo  
istruzione

indirizzo  
dato 1

**2000**

indirizzo  
dato 2

**2080**

indirizzo	contenuto memoria
...	
800	<b>ADD 4000, 2000, 2080</b>
...	
2000	<b>30</b>
...	
2080	<b>12</b>
...	
4000	

# Output

# Controllo

istruzione

- ☐ IF
- ☐ ID
- ☐ DF
- ☒ EX
- ☐ RR

program  
counter

804

# ALU

dato 1

30

dato 2 [ + ]

12

risultato

42

indirizzo result.

4000

# Input

# RAM

Bus

indirizzo  
istruzione

indirizzo  
dato 1

indirizzo  
dato 2

indirizzo	contenuto memoria
...	
800	<b>ADD 4000, 2000, 2080</b>
...	
2000	<b>30</b>
...	
2080	<b>12</b>
...	
4000	

# Output

# Controllo

istruzione

- ☐ IF
- ☐ ID
- ☐ DF
- ☐ EX
- ☒ RR

program  
counter

804

# ALU

dato 1

30

dato 2 [ + ]

12

risultato

42

indirizzo risult.

4000

# Input

# RAM

## Bus

indirizzo  
istruzione

indirizzo  
dato 1

indirizzo  
dato 2

indirizzo	contenuto memoria
...	
800	<b>ADD 4000, 2000, 2080</b>
...	
2000	30
...	
2080	12
...	
4000	42

indirizzo	contenuto memoria
...	
800	<b>ADD 4000, 2000, 2080</b>
...	
2000	30
...	
2080	12
...	
4000	42

# Output

# Velocità del ciclo macchina

In un secondo esegue un enorme numero di istruzioni

Il clock del computer

determina la velocità del ciclo macchina

misurato in Hertz (Hz)

ODG del miliardo di cicli al secondo (GHz)

# Software visto dal computer

Vede un oggetto binario (codice macchina)

una sequenza di parole (word, gruppi di 4 byte)

l'unico che comprende e sa eseguire

```
...  
10001111 10010100 00000011 01110100  
10001111 10011000 00000001 10101100  
00000010 10011000 10100000 00100000  
10101111 10010100 00000001 10010000  
...
```

# Il linguaggio assembly

Alternativa al linguaggio macchina

usa lettere e numeri

Di più facile comprensione per le persone

Traducibile automaticamente in linguaggio macchina

# Assemblare

Il computer scandisce il programma assembly

quando incontra una parole chiave cerca in una tabella la corrispondente sequenza binaria

assembla le varie parti dell'istruzione

“costruisce” l'istruzione macchina



# Linguaggi di alto livello

La maggior parte del software è scritta così

Prima compilato in linguaggio assembly

Poi assemblato per ottenere un file binario

# Compilare

Il processo è svolto da un'applicazione

Dai costrutti del linguaggio di alto livello all'assembly

Scritto in un linguaggio di alto livello (es. C o Java)

# Eseguire un'applicazione

Le istruzioni macchina vengono trasferite dal disco nella memoria RAM

Il ciclo macchina esegue le istruzioni

Tutte le istruzioni del computer sono eseguite dai circuiti contenuti nell'ALU

# Sistemi operativi

Offrono le operazioni base per l'uso del computer non supportate direttamente dall'hardware

I tre più usati per personal computer:

Microsoft Windows

Apple MacOS X

Linux (Unix)

# GUI

Sono “impacchettati” e forniti con l’OS

bordi delle finestre,

le barre di scorrimento,

i pulsanti,

i puntatori, ecc.

# Caratteristiche

## Multitasking

può eseguire più processi per volta

gestione dei programmi in esecuzione

## Multuser

più utenti in “contemporanea”

# Time sharing

Suddivide il tempo in intervalli

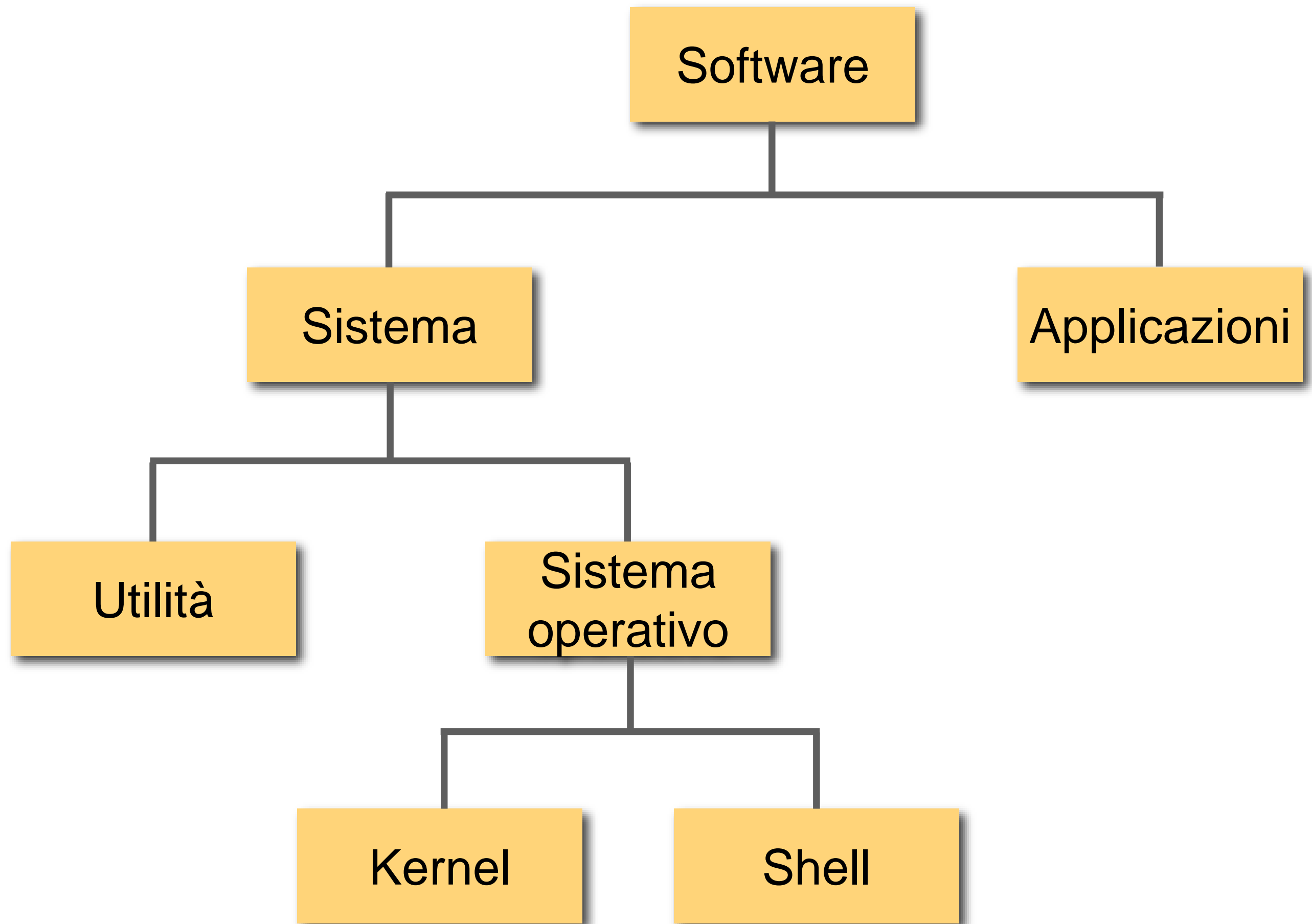
frazioni di secondo

Ad ogni intervallo:

un solo programma è in esecuzione

ciclicamente esegue tutti i programmi

L'utente ha l'impressione di contemporaneità





# Shell

Interfaccia con l'utente

contiene anche le GUI

Speso confuso con il sistema operativo

# Kernel

Parte interna del sistema operativo

driver

file manager

memory manager

# File manager

Gestisce i file nella memoria di massa

permessi di accesso

allocazione

icone

# Memory manager

Gestisce la RAM e i processi in esecuzione

Paginazione

Memoria virtuale

fine

Capitolo 7

Fluency – Conoscere e usare l'informatica