

**Informatica**



# Informatica

## Scienza

Come la matematica, l'astronomia, la chimica...

Si basa sul metodo sperimentale: Osservo – faccio ipotesi – verifico

## Informatica

**Informazione + Automatica = Informatica**

Studia come rappresentare, elaborare e trasmettere le informazioni

# Cos'è l'informatica

	<b>IDEE</b>	<b>STRUMENTI</b>
<b>RAPPRESENTARE</b>	<b>CODICI</b>	<b>SUPPORTI</b>
<b>ELABORARE</b>	<b>ALGORITMI</b>	<b>CALCOLATORI</b>
<b>TRASMETTERE</b>	<b>PROTOCOLLI</b>	<b>RETI</b>

# Introduzione

## Rappresentare

Associare simboli a concetti astratti o oggetti concreti

## Elaborare

Trasformare le informazioni mediante un procedimento noto

## Trasmettere

Spostare l'informazione da un luogo ad un altro

# Introduzione

## Codice

Sequenza di simboli che è associata ad una informazione

## Algoritmo

Procedimento mediante il quale vengono elaborate le informazioni

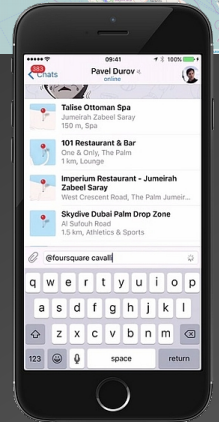
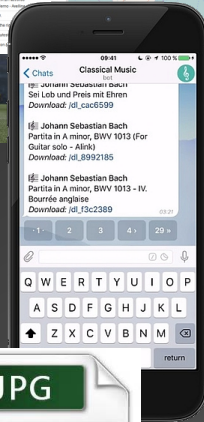
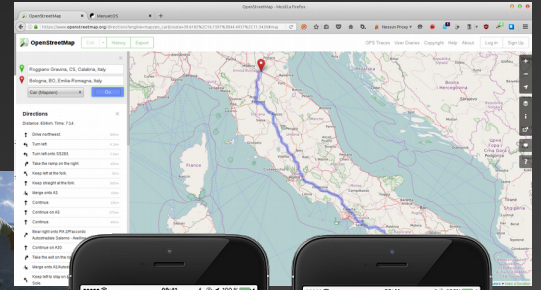
## Protocollo

Regole che devono essere rispettate per poter trasmettere le informazioni

# Introduzione



Gli strumenti:  
L'Hardware



Le idee:  
Il Software

# Attività

## Esperimento

Creiamo un codice per le lettere dell'alfabeto

Utilizziamo solo due simboli (sistema binario)

Otteniamo un codice per scambiare messaggi

## Osservazioni

Lunghezza dei codici

Difficoltà di decifrare il messaggio

Utilizzo di separatori (nuovo simbolo, non è più binario!)



# Codificare

## Cosa serve per codificare

Alfabeto

Regole

## Che caratteristiche deve avere un codice?

**Deve essere univoco**

Uno stesso codice non può rappresentare due informazioni diverse  
esempi: nickname, posta elettronica, numero di telefono

**Deve essere decifrabile**

Devo poter sempre risalire all'informazione partendo dal codice

## Codificare (da wikipedia)

In communications and information processing,

code is a system of rules to convert information—such as a letter, word, sound, image, or gesture—into another form or representation, sometimes shortened or secret, for communication through a channel or storage in a medium.

An early example is the invention of language, which enabled a person, through speech, to communicate what he or she saw, heard, felt, or thought to others.

But speech limits the range of communication to the distance a voice can carry, and limits the audience to those present when the speech is uttered.

The invention of writing, which converted spoken language into visual symbols, extended the range of communication across space and time

# La rappresentazione dell'informazione: il concetto di Bit

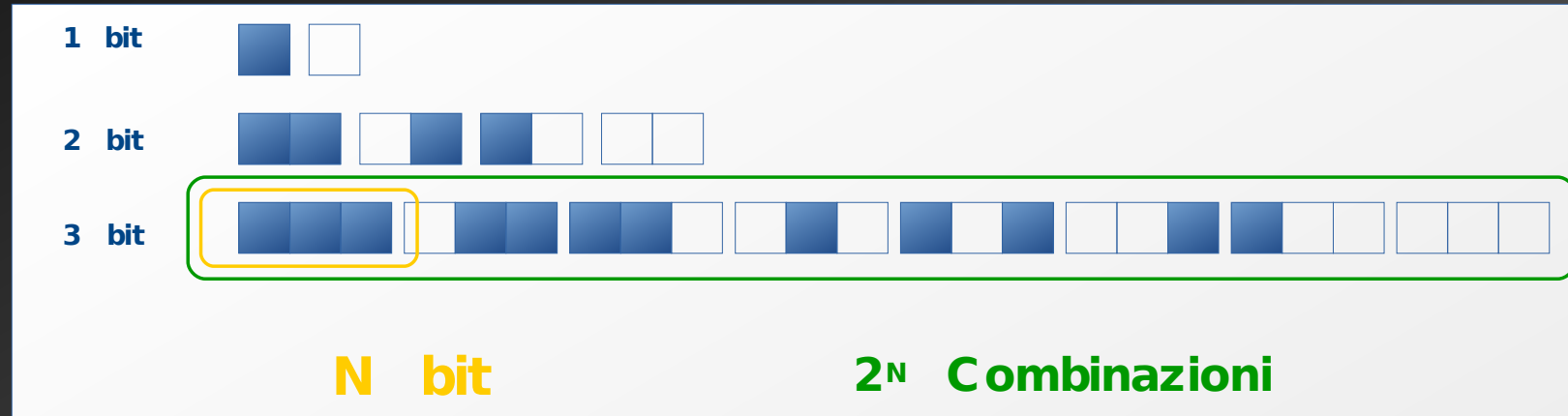
Bit: unità di misura elementare per la rappresentazione dell'informazione

## Livello logico

Il bit non ha una dimensione fisica, esprime la “lunghezza” dell'informazione

## Livello fisico

Per poter memorizzare i dati su un supporto bisogna trovare il modo per poterli scrivere. È il supporto ad avere una dimensione



# la rappresentazione dell'informazione: il concetto di Bit

Per assegnare in modo univoco (cioè senza ripetizioni) un codice ad N oggetti devo usare un numero di bit che mi permetta di ottenere un numero di combinazioni maggiore o uguale agli oggetti che voglio rappresentare.

## Esempio

Per codificare le 26 lettere dell'alfabeto devo usare almeno 5 bit perché

$$2^4 = 16 < 26$$

$$2^5 = 32 > 26$$

## Esempio

A 00000

B 00001

C 00010

D 00011

...

# La rappresentazione dell'informazione: il concetto di Bit

## Esempio

Una volta codificate le lettere dell'alfabeto, posso comporre parole utilizzando i codici di ogni singola lettera

BACCA = 000010000000010000100000

## Importante

Avendo utilizzato un numero fisso di bit per ogni lettera, riesco a capire il codice perché considero le sequenze di 1 e 0 a gruppi di 5. Se la codifica non utilizza un numero fisso di bit possono insorgere delle ambiguità. Ad esempio se

A 0

B 1

C 01

La sequenza 0110 può significare sia ABBA che CBA

# Codificare

## Codifica binaria

Perché il pc “conosce” solo ON/OFF cioè PASSA-CORRENTE/NON-PASSA-CORRENTE

## Importante per

### Capire il formato dei file

Per aprire un file, il programma deve sapere com'è fatto e cosa “significano” i bit!

### Dal bit al monitor

Per visualizzare un colore devo trovare un modo per rappresentarlo

### Caratteri, colori, suoni

ASCII, RGB, MIDI

# Equivalenze nel sistema binario

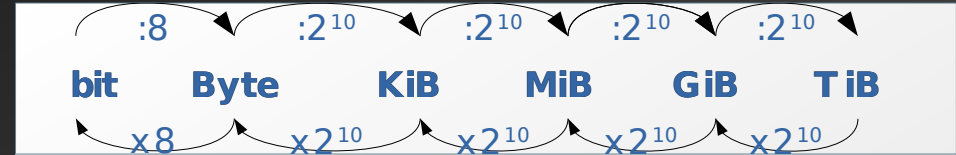
## Multipli e sottomultipli

bit    - Byte    - KiB    - MiB    - GiB    - TiB  
1    - 8    - 1024    - 1024    - 1024    - 1024

Attenzione: si usano le potenze del 2 ( $2^{10}$ ) e non del 10 ( $10^3$ ), ma le regole di conversione sono le stesse:

da sottomultiplo a multiplo: divido

da multiplo a sottomultiplo: moltiplico



Multipli del byte					
Prefissi SI			Prefissi binari		
Nome	Simbolo	Multiplo	Nome	Simbolo	Multiplo
kilobyte	kB	$10^3$	kibibyte	KiB	$2^{10}$
megabyte	MB	$10^6$	mebibyte	MiB	$2^{20}$
gigabyte	GB	$10^9$	gibibyte	GiB	$2^{30}$
terabyte	TB	$10^{12}$	tebibyte	TiB	$2^{40}$
petabyte	PB	$10^{15}$	pebibyte	PiB	$2^{50}$
exabyte	EB	$10^{18}$	exbibyte	EiB	$2^{60}$
zettabyte	ZB	$10^{21}$	zebibyte	ZiB	$2^{70}$
yottabyte	YB	$10^{24}$	yobibyte	YiB	$2^{80}$

## Occupazione di memoria per testi, immagini e suoni

### Esempi vari

Nei testi si codificano i caratteri

Nelle immagini si codificano i colori

Nei suoni si codifica

**Elaborare**



# Introduzione all'architettura del calcolatore

## **Esempio** *Preparazione di un piatto*

La ricetta è scritta su un foglio di carta

Leggo una frase

Prendo gli ingredienti che mi servono

Eseguo le operazioni indicate dalla ricetta  
utilizzando gli strumenti opportuni

La **ricetta** è scritta su un **foglio di carta**

**Leggo** una frase

**Prendo** gli **ingredienti** che mi servono

**Eseguo** le **operazioni** indicate dalla ricetta  
utilizzando gli **strumenti** opportuni

# Introduzione all'architettura del calcolatore

In maniera estremamente semplificata possiamo dire che un computer è costituito da due elementi:

**1) Una memoria che contiene i dati e le istruzioni (entrambi codificati in codice binario)**

**2) Un processore che esegue le operazioni**

Il processore legge i dati e le istruzioni contenute nella memoria ed esegue le operazioni necessarie per portarle a termine

## Esempio

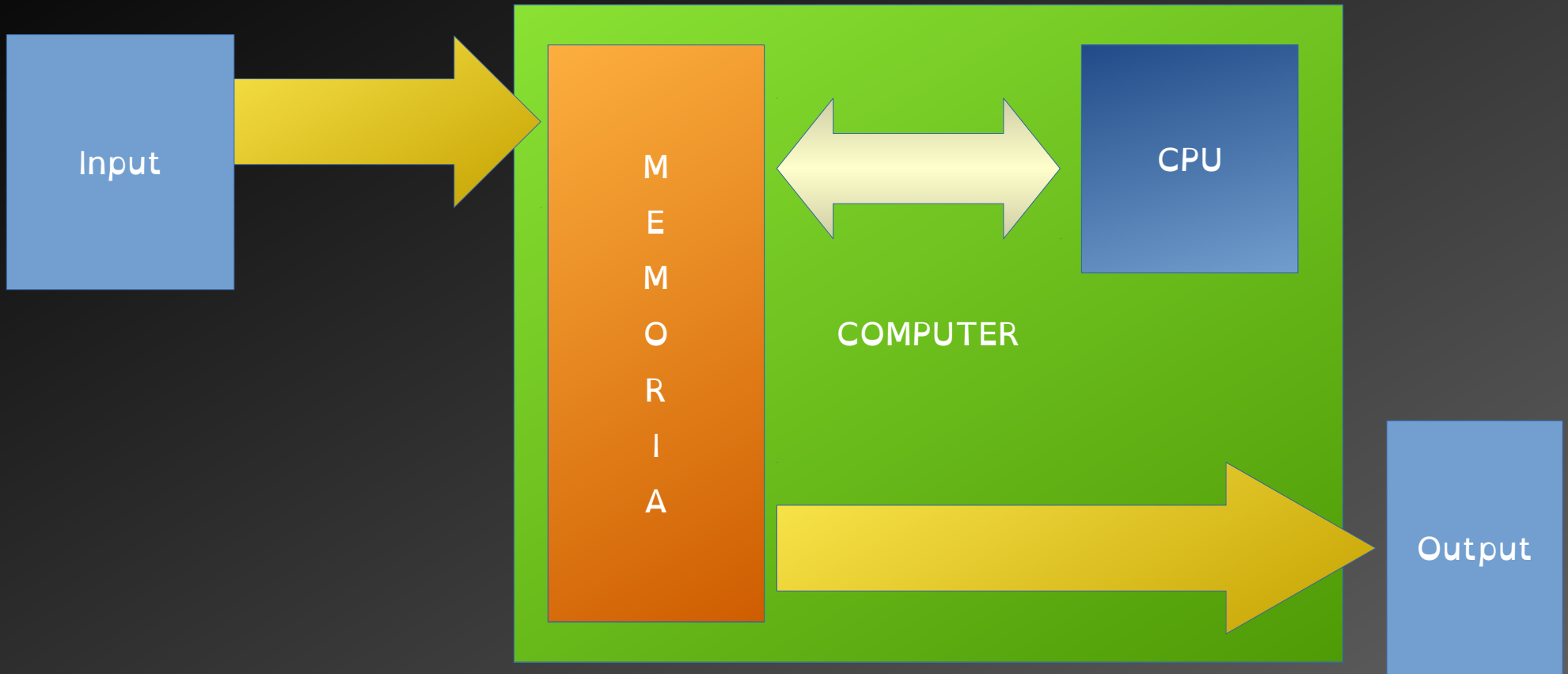
**1. A = 5**

**2. B = 7**

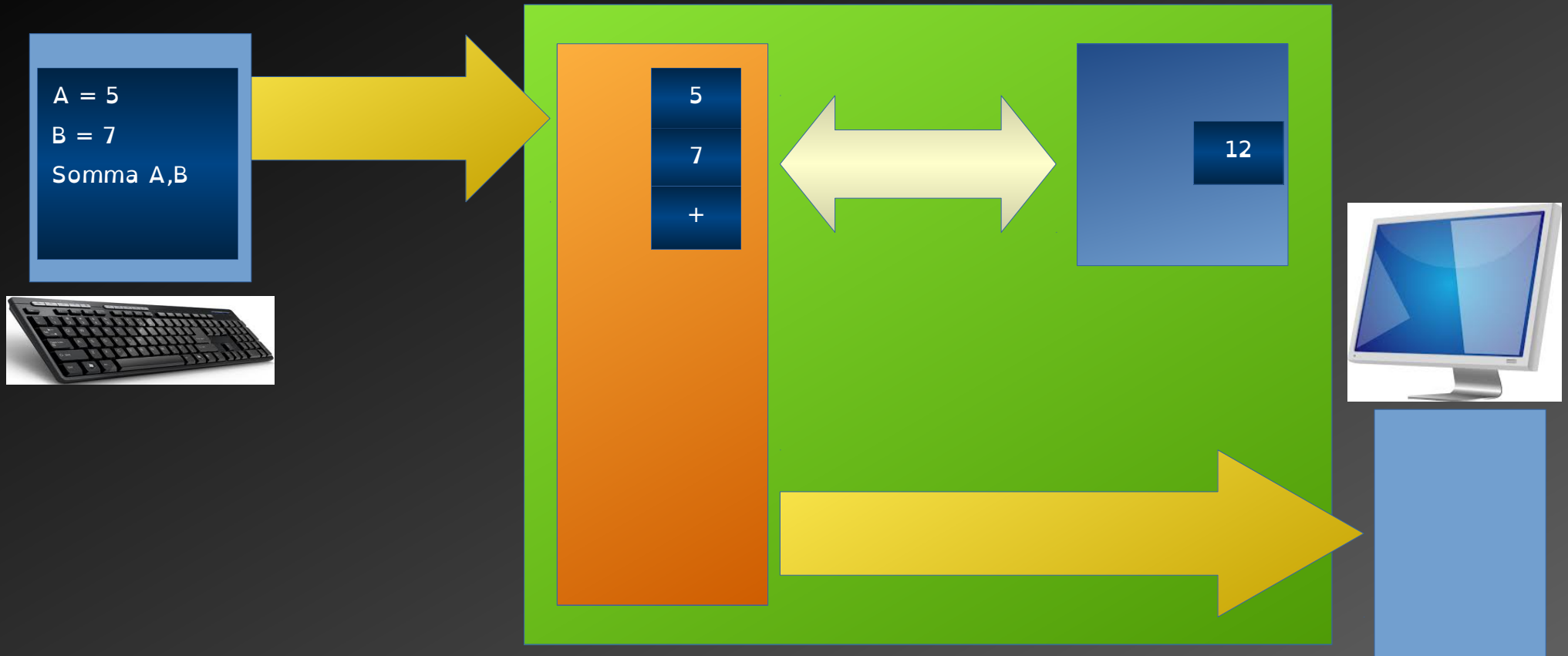
**3. Somma A e B**

**4. Visualizza il risultato sul monitor**

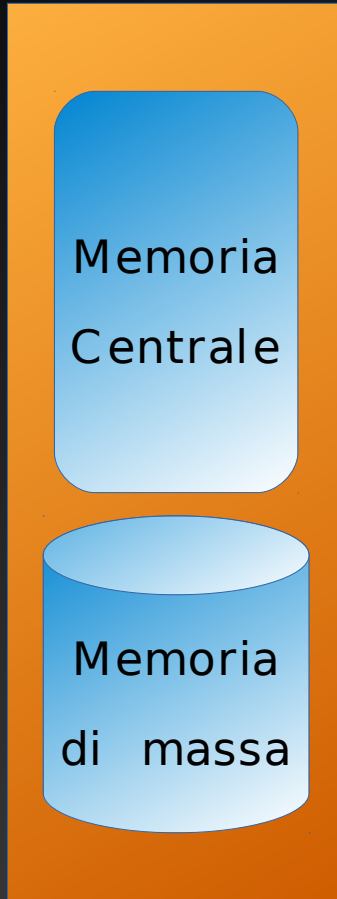
# Introduzione all'architettura del calcolatore



# Introduzione all'architettura del calcolatore



# Introduzione all'architettura del calcolatore: Memoria



La memoria di un computer si divide in due categorie:

- **Memoria centrale**
  - Contiene i dati e le istruzioni che vengono elaborati dalla CPU
- **Memoria di massa**
  - Serve per archiviare i dati e i programmi

# Introduzione all'architettura del calcolatore: Memoria Centrale (o RAM)

1	
2	
3	
4	
5	
6	
7	
8	



## Piccole

Se comparate alle memorie di massa

## Veloci

Dovendo fornire i dati direttamente alla CPU, si devono utilizzare dispositivi che trasferiscono i dati ad una velocità abbastanza elevata.

## Volatili

Una volta interrotta l'alimentazione elettrica i dati in essa memorizzati vengono persi

# Introduzione all'architettura del calcolatore: Memorie di massa



## Grandi

Compare con le memorie centrali molto più capienti, cioè possono ospitare una quantità molto maggiore di dati.

## Lente

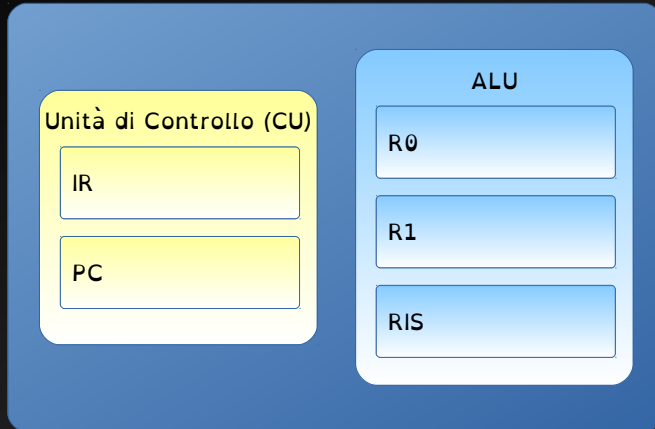
Non dovendo fornire dati direttamente alla CPU si possono utilizzare dispositivi che trasferiscono i dati a una velocità molto minore rispetto a quella utilizzata nello scambio di dati fra la RAM e la CPU.

## Persistenti

Una volta interrotta l'alimentazione elettrica i dati memorizzati non vengono persi ma sono nuovamente disponibili quando il dispositivo viene nuovamente alimentato.



# Introduzione all'architettura del calcolatore: Il processore



## UNITÀ DI CONTROLLO (CU)

Coordina le attività dei dispositivi

## ALU

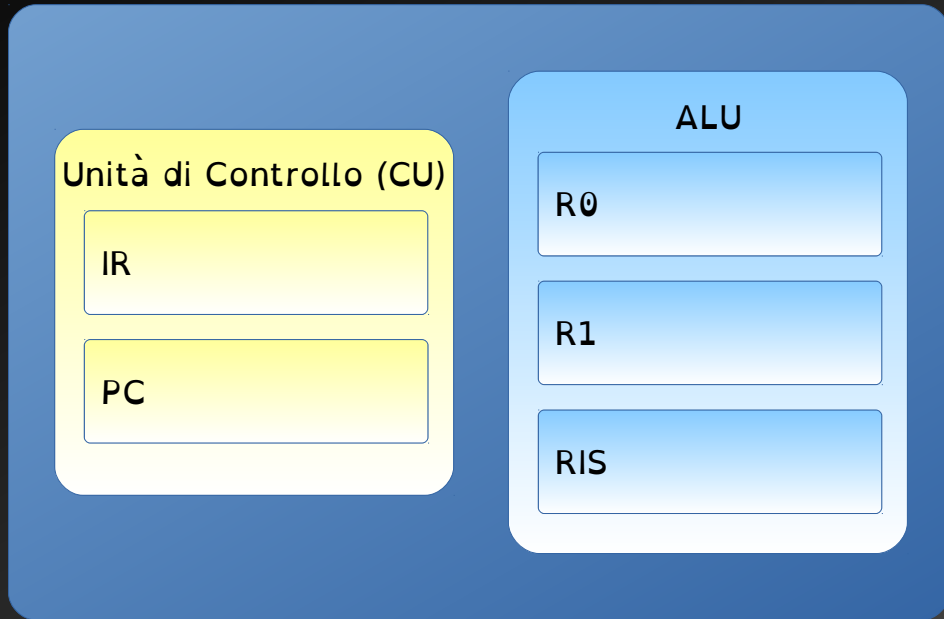
Effettua le operazioni matematiche e logiche

## REGISTRI

Memorizzano le informazioni sui calcoli che il processore deve eseguire o ha eseguito



# Introduzione all'architettura del calcolatore: Il processore



## CONTATORE DI PROGRAMMA (PC)

Contiene il valore della casella da cui prelevare l'istruzione da eseguire

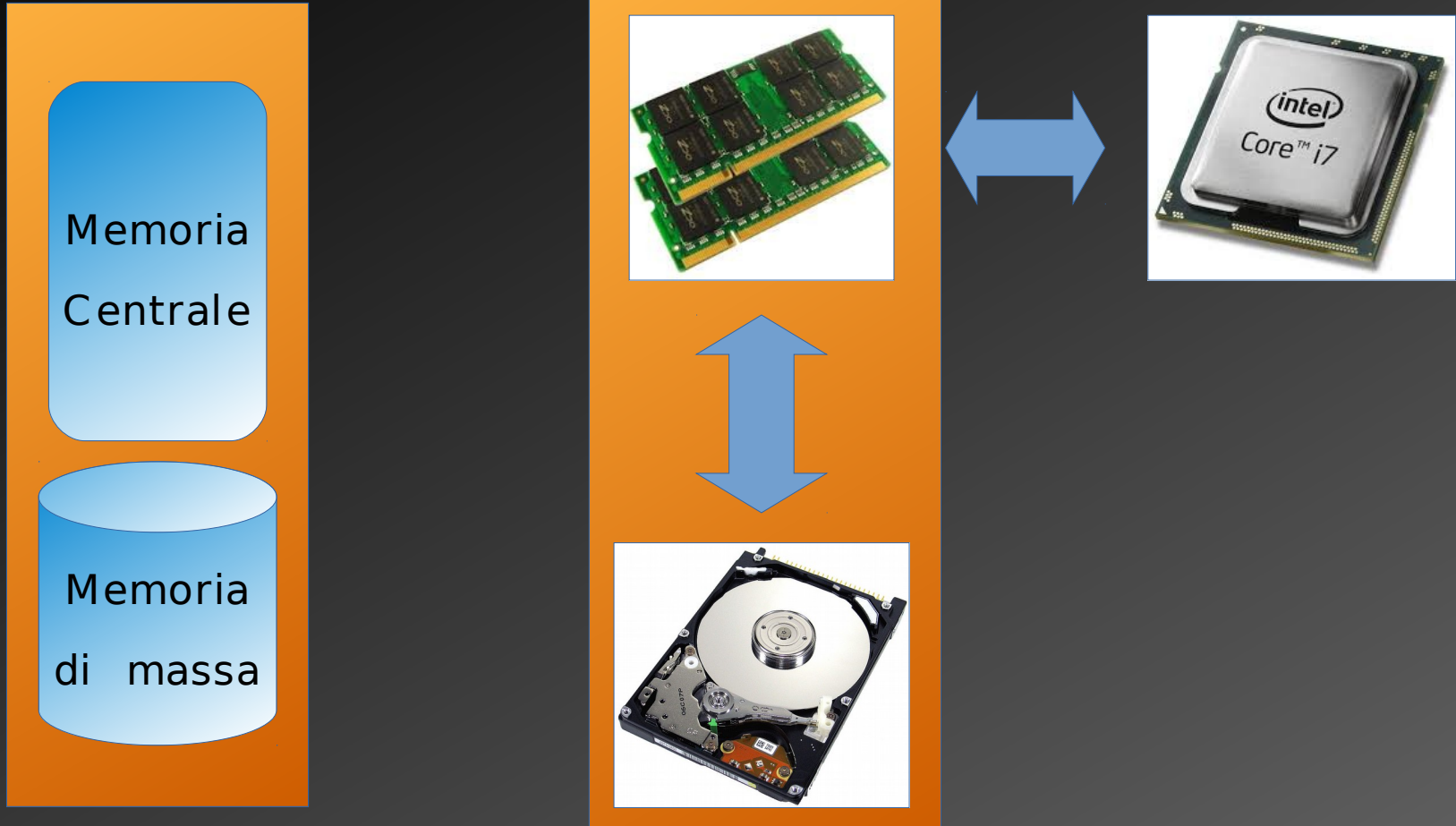
## REGISTRO ISTRUZIONI (IR)

Contiene l'istruzione che il processore deve eseguire

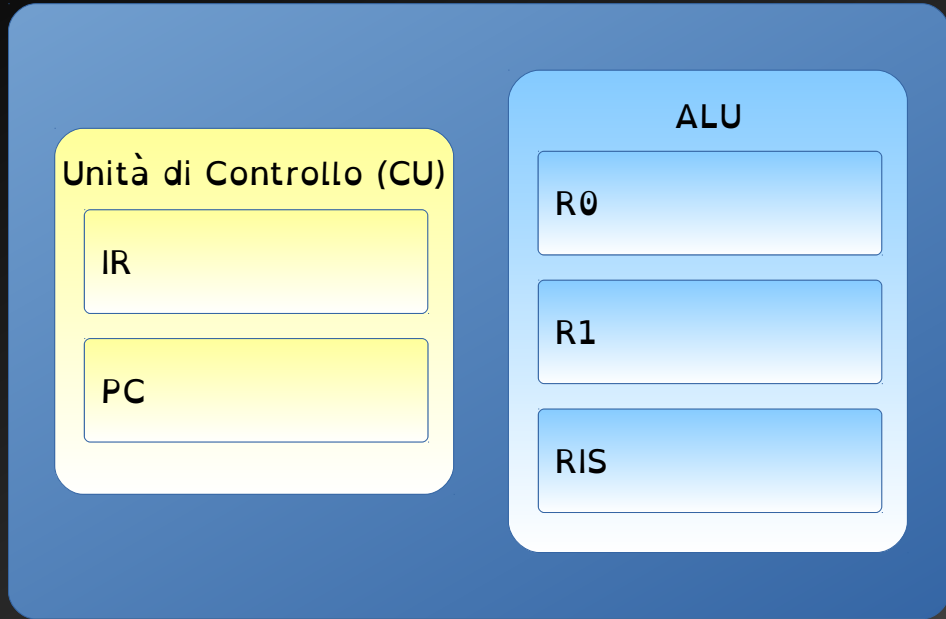
## R0, R1, RIS

Memorizzano i valori dei dati da elaborare e il loro risultato

# Introduzione all'architettura dei calcolatori



# L'elaborazione delle informazioni (I)



1	
2	
3	
4	
5	
6	
7	
8	

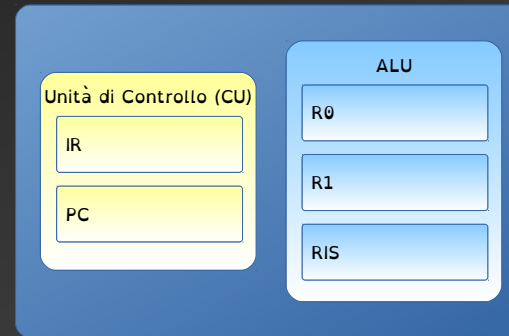
## L'elaborazione delle informazioni (II)

Per poter eseguire un programma, è necessario codificare istruzioni da comunicare al computer.

Una volta creato, per poter essere eseguito, il programma deve essere caricato nella memoria centrale.

Quando un programma si trova nella RAM, il processore riesce ad eseguirlo, ripetendo le seguenti azioni:

- 1) Preleva dalla memoria l'istruzione indicata in PC e la copia in IR
- 2) La esegue
- 3) Aumenta di 1 il valore nel PC e ripete



1	
2	
3	
4	
5	
6	
7	
8	

# Esempio I

Supponiamo di avere un computer che riconosca le seguenti istruzioni

**SOMMA casella casella casella**

**INPUT casella**

**OUTPUT casella**

**STOP**

Le istruzioni hanno il seguente significato

**SOMMA casella casella casella**

- Copia il valore della prima casella in R0
- Copia il valore della seconda casella in R1
- Mette in RIS la somma di R0 e R1
- Copia nella terza casella il valore di RIS

**INPUT casella**

- Acquisisce un numero e lo mette nella casella indicata

**OUTPUT casella**

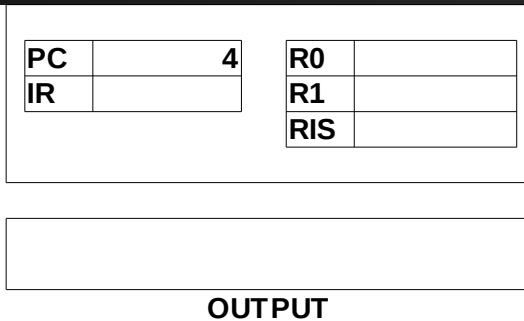
- Visualizza il valore della casella indicata

**STOP**

- Termina l'esecuzione del programma

# Esempio I

1	
2	
3	
4	INPUT 1
5	INPUT 2
6	SOMMA 1 2 3
7	OUTPUT 3
8	STOP
9	
10	
11	
12	
13	
14	
15	



Consideriamo il programma caricato in memoria e vediamo cosa succede quando viene eseguito dal computer.

All'interno di PC si trova il valore 4, la CPU andrà quindi nella cella 4 e preleverà l'istruzione che sarà copiata in IR

# Esempio 1

1	
2	
3	
4	INPUT 1
5	INPUT 2
6	SOMMA 1 2 3
7	OUTPUT 3
8	STOP
9	
10	
11	
12	
13	
14	
15	

PC	4	R0	
IR	INPUT 1	R1	
		RIS	

OUTPUT

1		5
2		
3		
4	INPUT 1	
5	INPUT 2	
6	SOMMA 1 2 3	
7	OUTPUT 3	
8	STOP	
9		
10		
11		
12		
13		
14		
15		

PC		4		
IR	INPUT 1			

R0			
R1			
RIS			

OUTPUT

Una volta “dentro” il processore, l’istruzione sarà eseguita.

In questo caso l’istruzione da eseguire indica che bisogna prendere un numero in input e metterlo nella cella 1

Successivamente, il PC viene aumentato di 1

# Esempio I

1	5			
2				
3				
4	INPUT 1			
5	INPUT 2			
6	SOMMA 1 2 3			
7	OUTPUT 3			
8	STOP			
9				
10				
11				
12				
13				
14				
15				

PC	5	R0	
IR		R1	
		RIS	

OUTPUT

1	5			
2				
3				
4	INPUT 1			
5	INPUT 2			
6	SOMMA 1 2 3			
7	OUTPUT 3			
8	STOP			
9				
10				
11				
12				
13				
14				
15				

PC	5	R0	
IR	INPUT 2	R1	
		RIS	

OUTPUT

1	5			
2	20			
3				
4	INPUT 1			
5	INPUT 2			
6	SOMMA 1 2 3			
7	OUTPUT 3			
8	STOP			
9				
10				
11				
12				
13				
14				
15				

PC	5	R0	
IR	INPUT 2	R1	
		RIS	

OUTPUT

Viene prelevata l'istruzione  
nella cella 5

Viene eseguita l'istruzione  
all'interno della CPU

Successivamente si aumenta  
di 1 il valore di PC



# Esempio I

1	5
2	20
3	
4	INPUT 1
5	INPUT 2
6	SOMMA 1 2 3
7	OUTPUT 3
8	STOP
9	
10	
11	
12	
13	
14	
15	

PC	6	R0	
IR	SOMMA 1 2 3	R1	
		RIS	

OUTPUT

1	5
2	20
3	
4	INPUT 1
5	INPUT 2
6	SOMMA 1 2 3
7	OUTPUT 3
8	STOP
9	
10	
11	
12	
13	
14	
15	

PC	6	R0	5
IR	SOMMA 1 2 3	R1	20
		RIS	25

OUTPUT

1	5
2	20
3	25
4	INPUT 1
5	INPUT 2
6	SOMMA 1 2 3
7	OUTPUT 3
8	STOP
9	
10	
11	
12	
13	
14	
15	

PC	6	R0	5
IR	SOMMA 1 2 3	R1	20
		RIS	25

OUTPUT

Viene prelevata l'istruzione  
nella cella 6

Viene eseguita l'istruzione  
all'interno della CPU

Successivamente si aumenta  
di 1 il valore di PC

# Esempio I

1	5
2	20
3	25
4	INPUT 1
5	INPUT 2
6	SOMMA 1 2 3
7	OUTPUT 3
8	STOP
9	
10	
11	
12	
13	
14	
15	

PC	7	R0	5
IR	OUTPUT 3	R1	20
		RIS	25

OUTPUT
--------

1	5
2	20
3	25
4	INPUT 1
5	INPUT 2
6	SOMMA 1 2 3
7	OUTPUT 3
8	STOP
9	
10	
11	
12	
13	
14	
15	

PC	7	R0	5
IR	OUTPUT 3	R1	20
		RIS	25

3
---

OUTPUT

Successivamente viene aumentato di 1 il PC

In questo caso l'istruzione successiva è STOP e il programma si ferma

1	5
2	20
3	25
4	INPUT 1
5	INPUT 2
6	SOMMA 1 2 3
7	OUTPUT 3
8	STOP
9	
10	
11	
12	
13	
14	
15	

PC	8	R0	5
IR		R1	20
		RIS	25

3
---

OUTPUT

# Esercizio

1	
2	
3	
4	
5	INPUT 1
6	INPUT 2
7	INPUT 4
8	SOMMA 1 2 3
9	SOMMA 3 4 3
10	SOMMA 3 3 3
11	OUTPUT 3
12	
13	
14	
15	

PC	5	R0	
IR		R1	
		RIS	

--

OUTPUT

1	
2	
3	
4	
5	
6	INPUT 1
7	INPUT 2
8	MOLTIPLICA 1 1 1
9	MOLTIPLICA 2 2 2
10	SOMMA 1 2 3
11	RADQ 3 4
12	OUTPUT 4
13	
14	
15	

PC	6	R0	
IR		R1	
		RIS	

OUTPUT
--------

## MOLTIPLICA

Prende il valore contenuto nella prima casella

Prende il valore contenuto nella seconda casella

Calcola il risultato

Mette il risultato nella terza casella

## RADQ

Prende il valore della prima casella

Calcola il risultato dell'operazione radice quadrata

Mette il risultato nella seconda casella