

# Architettura dell'elaboratore

Riprendiamo il discorso  
lasciato in sospeso  
ad inizio corso ...



---

---

---

---

---

---

---

---

## Riepilogando ...

- I programmi e i dati **risiedono** nella memoria secondaria
- Per essere eseguiti (i programmi) e usati (i dati) vengono **copiati** nella memoria principale
- Il processore è in grado di **eseguire in modo sequenziale** le istruzioni di cui sono composti i programmi

---

---

---

---

---

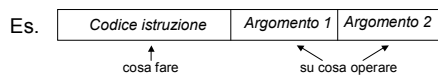
---

---

---

## Riepilogando ...

- I programmi sono fatti di **istruzioni elementari**
- Le istruzioni possono avere **formati diversi**



---

---

---

---

---

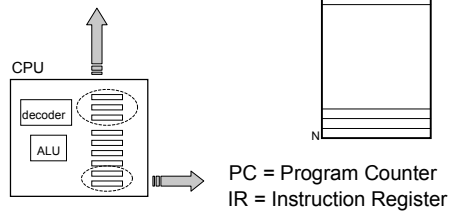
---

---

---

**Riepilogando ...**

MAR = Memory Address Register  
MDR = Memory Data Register  
[ RC = Registro di Controllo ]



---

---

---

---

---

---

---

---

**Ad ogni ciclo di clock**

1. Si legge dalla RAM la **prossima** istruzione da eseguire (**fetch**)
  
2. Si **decodifica** l'istruzione
  
3. Si **esegue** l'istruzione (**execute**)

---

---

---

---

---

---

---

---

**Ad ogni ciclo di clock**

- 1. Fetch**
  - i. si legge dalla RAM l'istruzione che si trova in RAM[PC] e si incrementa il PC
  
2. Si **decodifica** l'istruzione
  
3. Si **esegue** l'istruzione (**execute**)

---

---

---

---

---

---

---

---

**Ad ogni ciclo di clock**

**1. Fetch**

- i. MAR := PC
- ii. RC := "r"
- iii. MDR := RAM[MAR]
- iv. IR := MDR
- v. PC := PC + 1

2. Si **decodifica** l'istruzione

3. Si **esegue** l'istruzione (**execute**)

---

---

---

---

---

---

---

---

---

---

**Ad ogni ciclo di clock**

1. Fetch

**2. Si decodifica l'istruzione**

- > istruzioni di lettura in memoria
- > istruzioni di scrittura in memoria
- > istruzioni aritmetico / logiche
- > istruzioni di salto

**3. Si esegue l'istruzione**

- > a seconda del tipo di istruzione verranno intraprese azioni diverse

---

---

---

---

---

---

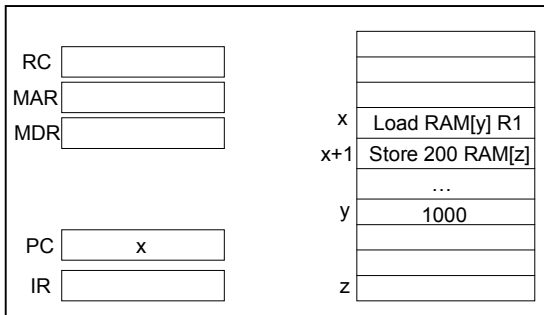
---

---

---

---

**Letture dalla memoria** (senza considerare il fetch dell'istruzione)



Descrizione ad alto livello, i dettagli al corso di Architetture ...

---

---

---

---

---

---

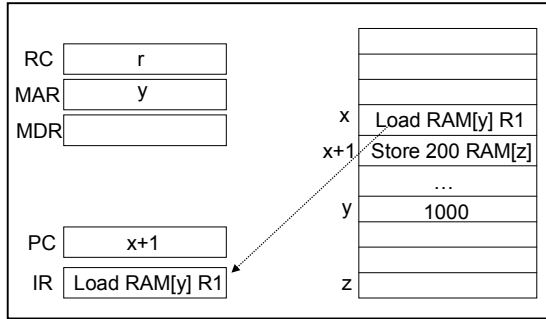
---

---

---

---

**Letture dalla memoria** (senza considerare il fetch dell'istruzione)



*Descrizione ad alto livello, i dettagli al corso di Architetture ...*

---

---

---

---

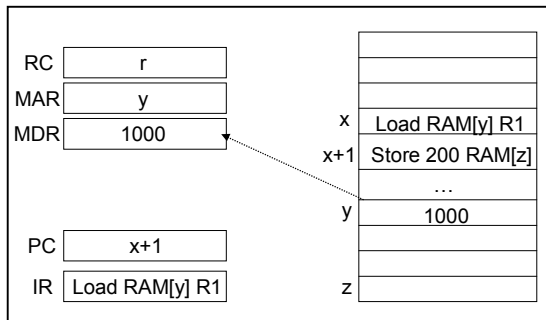
---

---

---

---

**Letture dalla memoria** (senza considerare il fetch dell'istruzione)




---

---

---

---

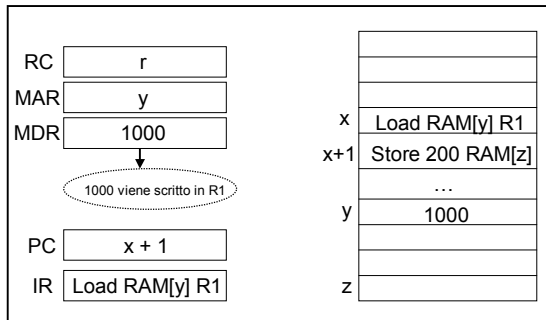
---

---

---

---

**Letture dalla memoria** (senza considerare il fetch dell'istruzione)




---

---

---

---

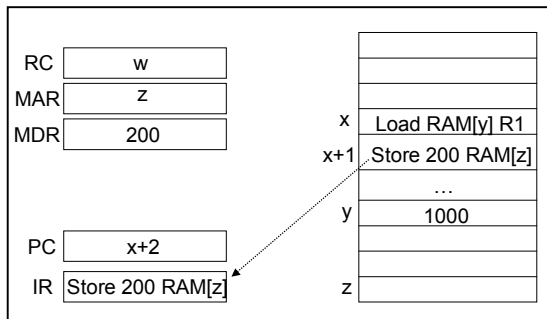
---

---

---

---

**Scrittura in memoria** (senza considerare il fetch dell'istruzione)




---

---

---

---

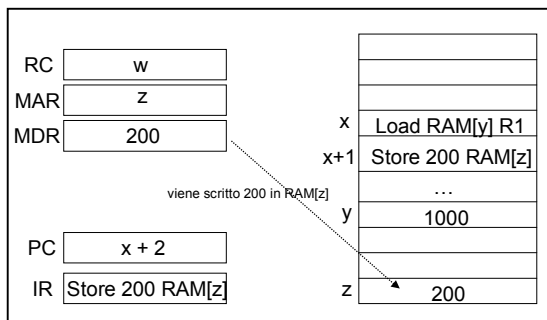
---

---

---

---

**Scrittura in memoria** (senza considerare il fetch dell'istruzione)




---

---

---

---

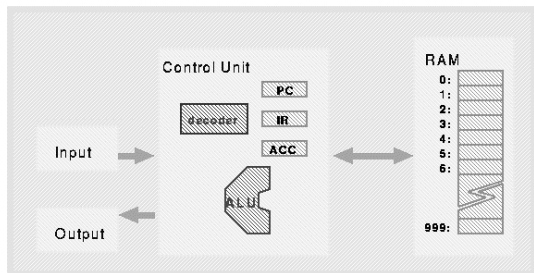
---

---

---

---

**La macchina di Von Neumann**




---

---

---

---

---

---

---

---

**La macchina di Von Neumann**

1. **RAM:** 1000 celle di memoria
2. **Input:** dispositivo per l'introduzione di valori interi
3. **Output:** dispositivo per l'output di valori interi
4. **Control Unit:** realizza il funzionamento della macchina

---

---

---

---

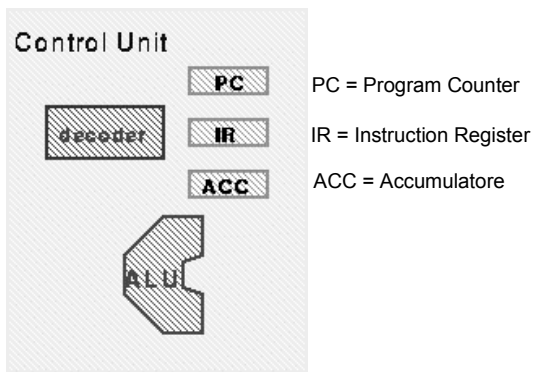
---

---

---

---

**La macchina di Von Neumann**



---

---

---

---

---

---

---

---

**Istruzioni della macchina: somma**

1. **codice:** 0     **parametro:** N
2. **descrizione:** somma tra due interi (il valore contenuto in ACC e il valore contenuto nella cella di memoria specificata dal parametro)
3. **esecuzione:** ACC := ACC + RAM[N]

:= indica l'operatore di assegnazione (= in C)

---

---

---

---

---

---

---

---

**Istruzioni della macchina: differenza**

- 1. **codice:** 1      **parametro:** N
- 2. **descrizione:** differenza tra due interi (il valore contenuto in ACC e il valore contenuto nella cella di memoria specificata dal parametro)
- 3. **esecuzione:** ACC := ACC - RAM[N]

---

---

---

---

---

---

---

---

**Istruzioni della macchina: lettura**

- 1. **codice:** 2      **parametro:** nessuno
- 2. **descrizione:** legge un valore numerico dal dispositivo di ingresso e lo copia in ACC
- 3. **esecuzione:** ACC := input

---

---

---

---

---

---

---

---

**Istruzioni della macchina: scrittura**

- 1. **codice:** 3      **parametro:** nessuno
- 2. **descrizione:** scrive il valore contenuto in ACC sul dispositivo di uscita
- 3. **esecuzione:** output := ACC

---

---

---

---

---

---

---

---

**Istruzioni della macchina: memorizza**

- 1. codice:** 4      **parametro:** N
- 2. descrizione:** memorizza il valore contenuto in ACC nella cella di memoria specificata dal parametro
- 3. esecuzione:** RAM[N] := ACC

---

---

---

---

---

---

---

---

**Istruzioni della macchina: carica**

- 1. codice:** 5      **parametro:** N
- 2. descrizione:** carica in ACC il valore contenuto nella cella specificata dal parametro
- 3. esecuzione:** ACC := RAM[N]

---

---

---

---

---

---

---

---

**Istruzioni della macchina: salto**

- 1. codice:** 6      **parametro:** N
- 2. descrizione:** copia il parametro nel registro PC
- 3. esecuzione:** PC := N

---

---

---

---

---

---

---

---



**Istruzioni della macchina: salto condizionato**

- 1. codice:** 7      **parametro:** N
- 2. descrizione:** se il valore di ACC è zero, copia il parametro nel registro PC
- 3. esecuzione:** if (ACC == 0) then PC := N

---

---

---

---

---

---

---

---

**Istruzioni della macchina: end**

- 1. codice:** 8      **parametro:** nessuno
- 2. descrizione:** ferma l'interpretazione del programma
- 3. esecuzione:** ferma l'interpretazione del programma

---

---

---

---

---

---

---

---

**Codifica delle istruzioni come numeri interi**

- 1. Moltiplico per 1000** il codice dell'istruzione
  - 2. Sommo** il valore dell'eventuale **parametro**
- NB:** il parametro dovrà essere compreso tra 0 e 999

---

---

---

---

---

---

---

---

**Codifica delle istruzioni come numeri interi**



Esempi

1. Voglio **scrivere nella cella 300**

- i. istruzione 4       $4 * 1000 = 4000$
- ii. parametro 300     $4000 + 300 = 4300$

2. Voglio **sommare al contenuto di ACC il valore della cella 167**

- i. istruzione 0       $0 * 1000 = 0$
- ii. parametro 167     $0 + 167 = 167$

---

---

---

---

---

---

---

---

**Codifica delle istruzioni come numeri interi**



Esempi

1. **6899**

- i.  $6899 / 1000 = 6 + 899$
- ii. istruzione 6 (salto)
- iii. parametro 899 (PC := 899)

2. **8000**

- i.  $8000 / 1000 = 8$
- ii. istruzione 8 (end)

---

---

---

---

---

---

---

---

**Un programma per la somma di due numeri**



		RAM	
PC	0	0	2000
IR	0	1	4050
ACC	0	2	2000
		3	50
		4	3000
		5	8000
			...
		50	...

---

---

---

---

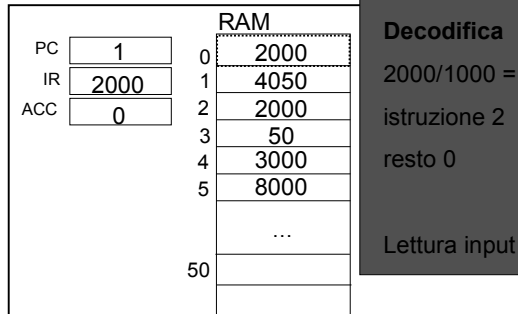
---

---

---

---

**Un programma per la somma di due numeri**



**Decodifica**  
 2000/1000 =  
 istruzione 2  
 resto 0  
 Lettura input

---

---

---

---

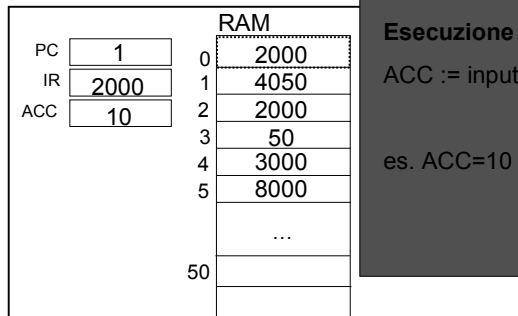
---

---

---

---

**Un programma per la somma di due numeri**



**Esecuzione**  
 ACC := input  
 es. ACC=10

---

---

---

---

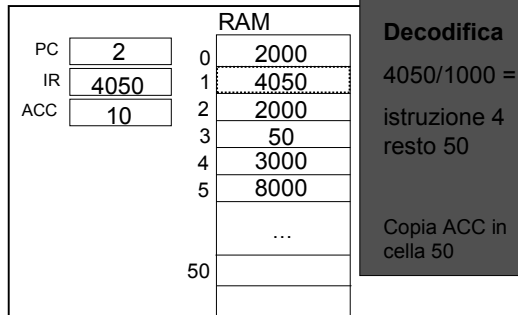
---

---

---

---

**Un programma per la somma di due numeri**



**Decodifica**  
 4050/1000 =  
 istruzione 4  
 resto 50  
 Copia ACC in  
 cella 50

---

---

---

---

---

---

---

---

**Un programma per la somma di due numeri**

PC	2	RAM	
IR	4050	0	2000
ACC	10	1	4050
		2	2000
		3	50
		4	3000
		5	8000
			...
		50	10

**Esecuzione**  
RAM[50] := 10

---

---

---

---

---

---

---

---

**Un programma per la somma di due numeri**

PC	3	RAM	
IR	2000	0	2000
ACC	10	1	4050
		2	2000
		3	50
		4	3000
		5	8000
			...
		50	10

**Decodifica**  
2000/1000 =  
istruzione 2  
resto 0  
Lettura input

---

---

---

---

---

---

---

---

**Un programma per la somma di due numeri**

PC	3	RAM	
IR	2000	0	2000
ACC	40	1	4050
		2	2000
		3	50
		4	3000
		5	8000
			...
		50	10

**Esecuzione**  
ACC := input  
es. ACC=40

---

---

---

---

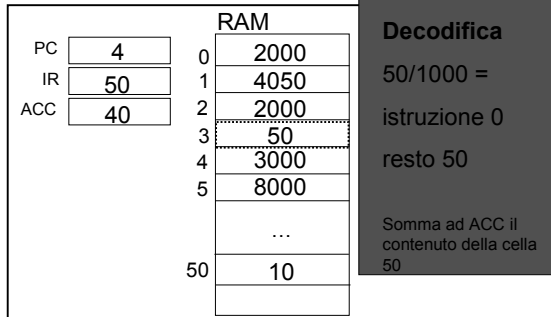
---

---

---

---

**Un programma per la somma di due numeri**




---

---

---

---

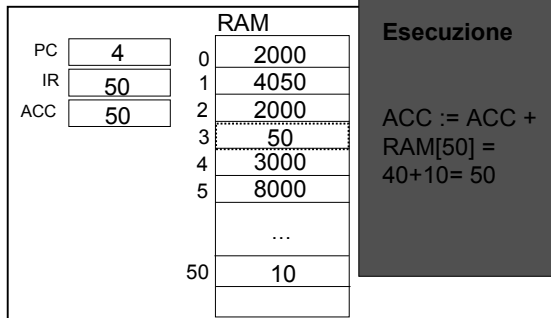
---

---

---

---

**Un programma per la somma di due numeri**




---

---

---

---

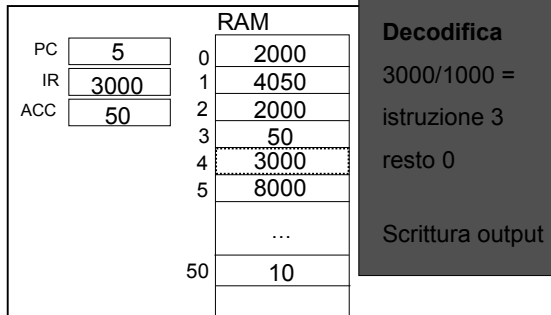
---

---

---

---

**Un programma per la somma di due numeri**




---

---

---

---

---

---

---

---

**Un programma per la somma di due numeri**

PC	5	RAM	0	2000
IR	3000		1	4050
ACC	50		2	2000
			3	50
			4	3000
			5	8000
				...
			50	10

**Esecuzione**  
output := ACC = 50

---

---

---

---

---

---

---

---

---

---

**Un programma per la somma di due numeri**

PC	6	RAM	0	2000
IR	8000		1	4050
ACC	50		2	2000
			3	50
			4	3000
			5	8000
				...
			50	10

**Decodifica e esecuzione**  
8000/1000 =  
istruzione 8  
resto 0  
**Fine**

---

---

---

---

---

---

---

---

---

---

**Un programma per ripetere N volte la somma di 2 numeri**

RAM	0	2000	ACC := input
	1	7009	if (ACC == 0) then PC := 9
	2	4008	RAM[8] := ACC
	3	2000	ACC := input
	4	7009	if (ACC == 0) then PC := 9
	5	8	ACC := ACC + RAM[8]
	6	3000	output := ACC
	7	6000	PC := 0
	8	0	risultato della somma
	9	8000	End
		...	

---

---

---

---

---

---

---

---

---

---