

# Programmazione I

A.A. 2002-03

## Architettura dei Calcolatori

( Lezione V )

### Componenti hardware e loro schema funzionale

**Prof. Giovanni Gallo**

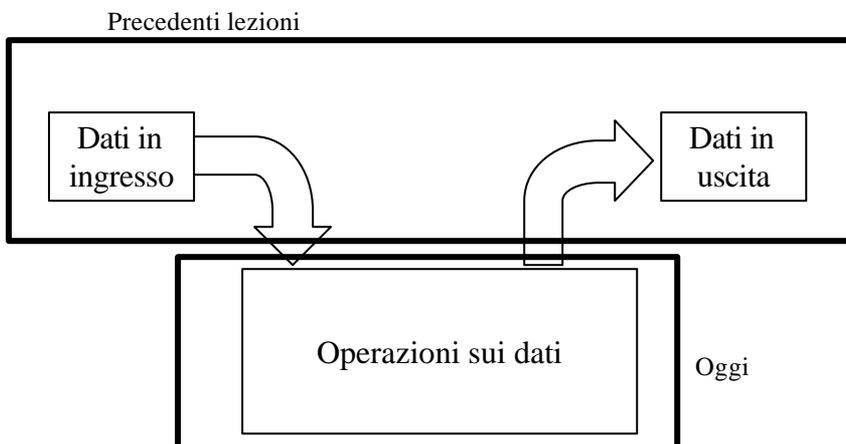
**Dr. Gianluca Cincotti**

Dipartimento di Matematica e Informatica

Università di Catania

e-mail : { [gallo](mailto:gallo@dmf.unict.it), [cincotti](mailto:cincotti@dmf.unict.it) } @dmf.unict.it

## Un vecchio modello ...



# *Molti i “modelli” operativi di computer*

---

- Macchine di Turing;
- Valutatori funzionali;
- Sistemi di Post-Markov;



Nel corso di  
“Fondamenti di Informatica”  
(III semestre)

- Macchina di von Neumann o macchina RAM.

È il modello astratto più vicino alla struttura dei computer che usiamo tutti i giorni: conoscere tale modello ci aiuta a “crearci” un modello mentale adeguato per programmare in maniera “imperativa”

## *Eccolo qui!*

---

Grafica non essenziale –  
omessa negli stampati per  
ridurre la size del file

# *La macchina computer*

---

- In generale, un computer:
  - esegue *operazioni* logiche e aritmetiche,
  - ha una *memoria* per conservare i dati.
  
- Un *programma* contiene la descrizione di tutte le operazioni da eseguire.

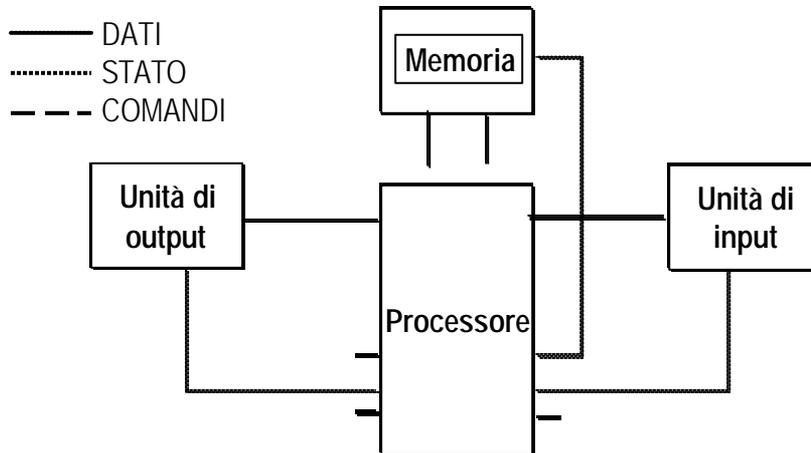
# *Architettura dei computers*

---

- In un computer possiamo distinguere tre unità funzionali:
  - Processore
    - fornisce la capacità di elaborazione delle informazioni,
  - Memoria (centrale e di massa),
  - Dispositivi di input/output,  
che comunicano attraverso un canale detto BUS
    - costituito da un insieme di linee elettriche digitali.
  
- Macchina di von Neumann.

## *Schema a blocchi di un elaboratore*

---



## *Hardware vs. Software*

---

- *L'hardware* denota la struttura fisica del computer, costituita di norma da componenti elettronici che svolgono specifiche funzioni nel trattamento dell'informazione.
- Il *software* denota l'insieme delle istruzioni che consentono all'hardware di svolgere i propri compiti (programmi).

## *Il linguaggio del processore*

---

- Ogni modello di microprocessore ha un proprio *linguaggio macchina* diverso da quello di altri microprocessori.
  - Il linguaggio macchina specifica tutte e sole le *istruzioni macchina* che possono essere eseguite dal microprocessore.

## *Le istruzioni macchina*

---

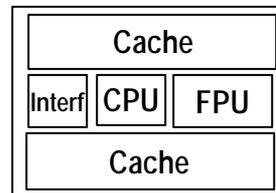
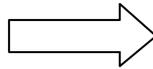
- Un' *istruzione macchina* costituisce un'operazione primitiva che il processore è in grado di svolgere.
  - È composta da:
    - *Codice operazione* (indica cosa fare),
    - *Uno o due operandi* (su cosa operare),
    - *Destinazione* del risultato (dove memorizzarlo).
- Un *programma* è costituito da una sequenza ordinata di *istruzioni macchina* espresse in codice binario.

10011010
01010101
01110010
10010111

# Processore

---

- Composto da blocchi con funzionalità diverse:
  - CPU (Central Processing Unit),
  - FPU (Floating Point Unit),
  - Cache,
  - Interfacce varie.
- Se integrato su un unico chip prende il nome di *microprocessore*.



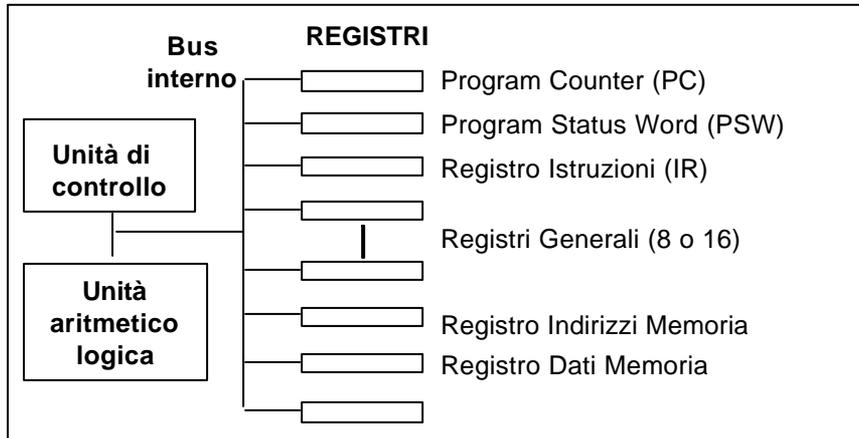
# Central Processing Unit (CPU)

---

- Svolge tutte le operazioni di:
  - elaborazione numerica,
  - controllo e coordinamento di tutte le attività.
- Si suddivide in:
  - Unità logico-aritmetica (ALU).
  - Unità di controllo (CU).
  - Registri.

# Componenti della CPU

---



# Arithmetic Logic Unit (ALU)

---

- Svolge tutti i calcoli *logici* ed *aritmetici* (complementazione, somma intera, confronto, etc).
  - Opera direttamente sui registri generali.
- E' costituita da circuiti elettronici in grado di eseguire la *somma* di due numeri binari contenuti in due registri oppure di eseguire il *confronto* tra due numeri.

# Registri

---

- Memoria locale usata per memorizzare:
  - dati acquisiti dalla memoria centrale o dalla unità di input,
  - risultati delle operazioni eseguite dall' ALU.
- Numero limitato: tipicamente da 8 a 256.
- Unità di memoria estremamente veloci.
- Le dimensioni di un registro sono una caratteristica fondamentale del processore: 16, 32, 64 bit.

# Registri (cont.)

---

- I registri contengono dati ed informazioni che vengono immediatamente elaborate.
- Esistono due tipi di registri:
  - i *registri speciali* utilizzati dalla CU per scopi particolari,
  - i *registri di uso generale* (registri aritmetici).

## *Registri speciali*

---

- Il *Program Counter* (PC) contiene l'indirizzo di memoria che contiene la prossima istruzione da eseguire.
- L' *Instruction Register* (IR) contiene l'istruzione attualmente in esecuzione.
- Il *registro di stato* (PSW) contiene delle informazioni sullo stato di esecuzione del processore.
  - In particolare esso può segnalare eventuali errori che possono avvenire durante l'esecuzione di un programma.

## *Control Unit (CU)*

---

- E' la parte più importante del processore:
  - In base:
    - al *programma* in esecuzione ed
    - allo stato di tutte le unità,decide l'operazione da eseguire ed emette gli ordini relativi.
  - In pratica, esegue le istruzioni macchina.

## *L'unità di controllo al lavoro ...*

---

Grafica non essenziale –  
omessa negli stampati per  
ridurre la size del file

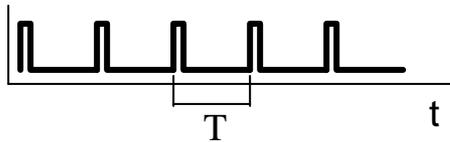
## *Floating Point Unit (FPU)*

---

- In molti elaboratori si può avere una FPU specializzata in operazioni matematiche complesse (numeri reali).
- Svolge tutte le operazioni che la ALU della CPU non è in grado di svolgere:
  - Somma/sottrazione reale,
  - Moltiplicazione intera/reale,
  - Divisione,
  - Funzioni matematiche complesse.
- Nei calcolatori di generazioni precedenti era esterna alla CPU (Coprocesore matematico).

## *Il clock*

- Ogni elaboratore contiene un circuito di temporizzazione (*clock*) che genera un riferimento temporale comune per tutti gli elementi del sistema.
  - $T = \textit{periodo}$  di clock
  - $f = \textit{frequenza}$  di clock ( $= 1/T$ )



- Frequenze tipiche delle ultime generazioni:  
 $f > 1000 \text{ MHz}$ ,  $T < 10^{-9}$  secondi.

## *Tempistica delle istruzioni*

- Un'istruzione macchina viene eseguita dal microprocessore svolgendo una sequenza di *operazioni elementari*.
  - Un *ciclo-macchina* è il tempo richiesto per svolgere un'operazione elementare.
    - È un multiplo del periodo del clock.
  - Il numero di operazioni elementari necessario a portare a compimento un'istruzione macchina è dell'ordine di 7-10, dipendentemente dal tipo di istruzione.

## *Velocità del microprocessore*

---

- La velocità di elaborazione di un processore dipende dalla frequenza del clock.
  - I processori attuali hanno valori di frequenza di clock che varia tra gli 8 MHz ed i 1800 MHz.

## *Memoria*

---

- Viene utilizzata per conservare dati e programmi.
- Si suddivide in:
  - **Memoria di lavoro** (memoria principale).
    - Memoria in grado di conservare dinamicamente dati e programmi che il processore sta utilizzando.
      - ❖ **Ram (Random Access Memory)**
      - ❖ **Rom (Read Only Memory)**
  - **Memoria magazzino** (memoria di massa).

## Organizzazione della memoria principale

---

- La memoria è organizzata funzionalmente in **locazioni** indipendenti.
- Ad ogni locazione è associato un **indirizzo**
  - cioè, un numero progressivo a partire da 0.

00101101	0
10011001	1
10010110	2
	3
10010101	4
	5
	6
	7
	8

## Organizzazione della memoria principale (cont.)

---

- Ogni locazione ha la stessa dimensione: 8, 16, 32, o 64 bit.
  - Ogni locazione viene anche detta **parola** (**word**) e costituisce un blocco unico.
- Le uniche operazioni che si effettuano sulla memoria sono **lettura** e **scrittura**.
  - Una locazione di memoria può contenere un dato o un'istruzione.

00101101	0
10011001	1
10010110	2
	3
10010101	4
	5
	6
	7
	8

## *Indirizzi di memoria*

---

- L'indirizzo di una locazione di memoria è un numero intero codificato in binario.
  - Ogni computer utilizza un numero di bit costante per rappresentare gli indirizzi.
- Maggiore è il numero di bit utilizzati, maggiore sarà il numero di locazioni indirizzabili: ***Spazio di indirizzamento.***

## *Indirizzi di memoria (cont.)*

---

- Se l'elaboratore utilizza:
  - 16 bit per l'indirizzo, la memoria conterrà fino a 65.536 locazioni (cioè 64 KB di memoria).
  - 32 bit per l'indirizzo, la memoria potrà contenere fino a 4.294.967.296 locazioni (cioè 4 GB di memoria).

## Schema funzionale

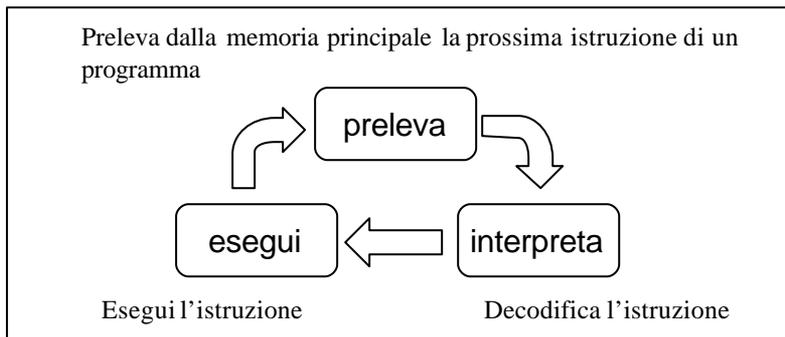
Programma e dati sono caricati in memoria.

- 1) La CU *preleva* un'istruzione dalla memoria.
- 2) L'istruzione viene *decodificata*.
- 3) L'istruzione viene *eseguita*.
- 4) La CU passa all'istruzione successiva e cioè al punto 1).

## Schema funzionale (cont.)

➤ Il processore esegue in continuazione il ciclo

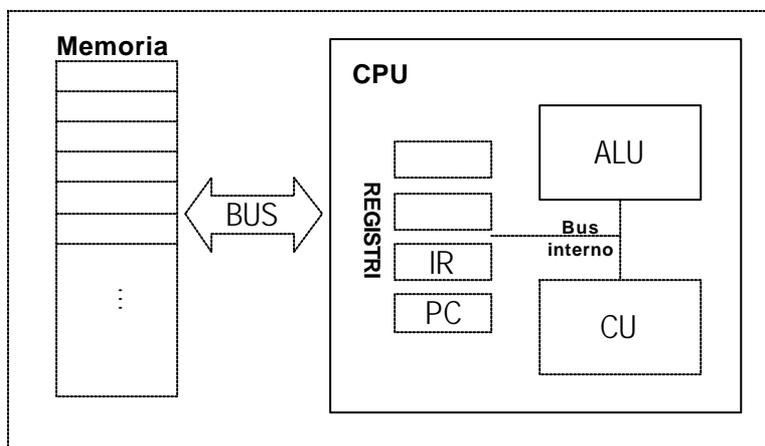
- *preleva – interpreta – esegui*.

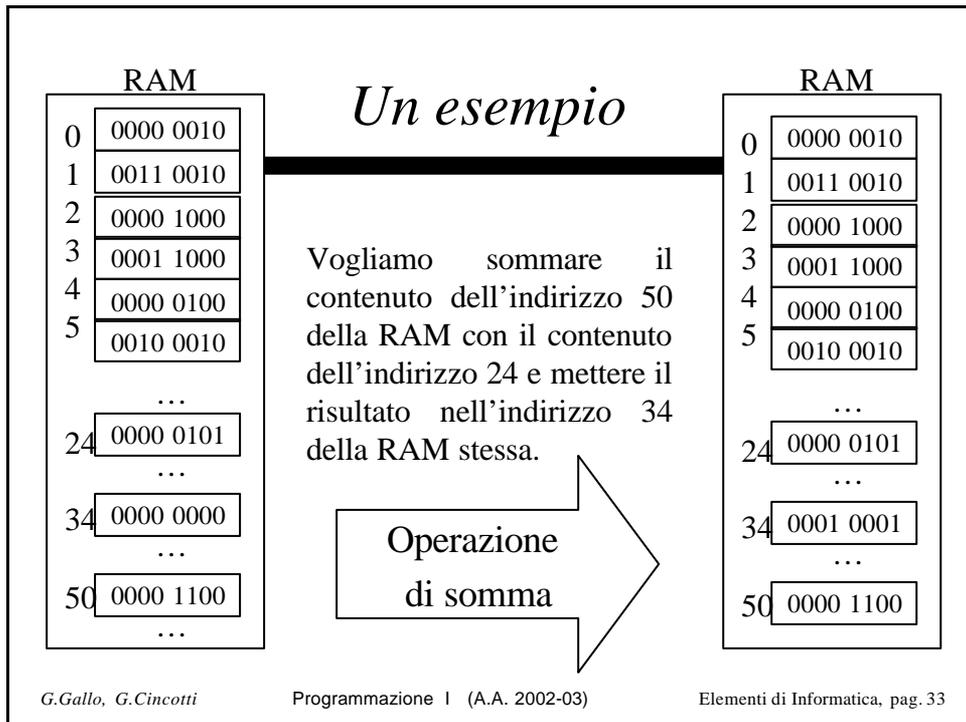


## Schema funzionale (dettagliato)

1. La CU *preleva* l'istruzione che si trova all'indirizzo di memoria contenuto nel PC e la trasferisce nell'IR (fase di *fetch*).
2. L'istruzione in IR viene *decodificata* ed *eseguita*.
  - L'*esecuzione* consiste nell'invio da parte della CU di opportuni comandi all'unità interessata:
    - Calcoli → ALU,
    - Acquisizione/Stampa → Unità di I/O,
    - Trasferimento dati → Memoria.
3. La CU *incrementa* opportunamente il PC ed il processo si ripete passando all'istruzione successiva in memoria (si torna quindi al passo 1).

## Schema della CPU





## *L'operazione di somma tra due numeri in memoria non è elementare!*

➤ La sequenza di operazioni da fare è:

- Copia il contenuto della word 50 dalla RAM al registro ACC (accumulatore);
- Prendi il contenuto della word 24 ed incrementa ACC di tale valore;
- Scrivi il contenuto del registro ACC nella parola 34 della RAM.

G.Gallo, G.Cincotti
Programmazione I (A.A. 2002-03)
Elementi di Informatica, pag. 34

## *Un linguaggio macchina “fittizio”*

---

Solo a scopo didattico ipotizziamo che alcune istruzioni siano “*codificate*” nel nostro microprocessore come segue:

Copia dalla RAM al registro ACC:	0 0 0 0 0 0 1 0
Copia il valore in ACC nella RAM :	0 0 0 0 0 1 0 0
Somma in ACC un valore nella RAM:	0 0 0 0 1 0 0 0

## *Ecco il nostro programma...*

---

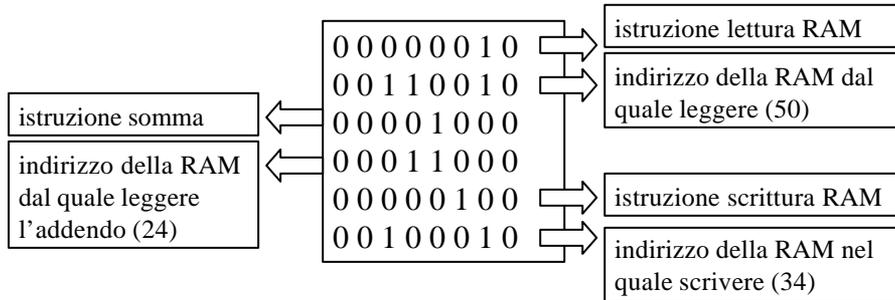
0 0 0 0 0 0 1 0
0 0 1 1 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 1 1 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 1 0

Facile ?

Forse ... ma poco comprensibile  
per un povero umano!

Le cose diventano più chiare se commentiamo il codice.  
**SCRIVERE COMMENTI AI PROPRI PROGRAMMI E' PARTE ESSENZIALE DELL'ATTIVITA' DI PROGRAMMAZIONE.**

# Codice commentato

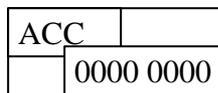
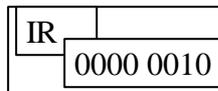
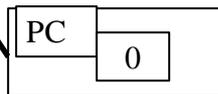


Il programma viene inizialmente caricato in RAM; il PC viene inizializzato all'indirizzo della prima istruzione.

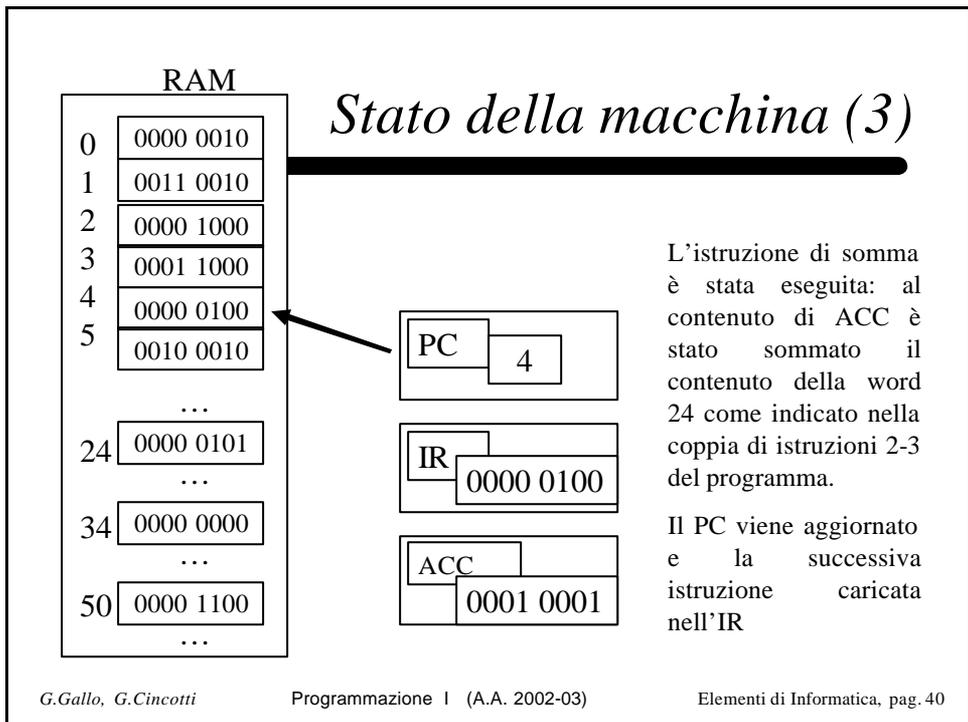
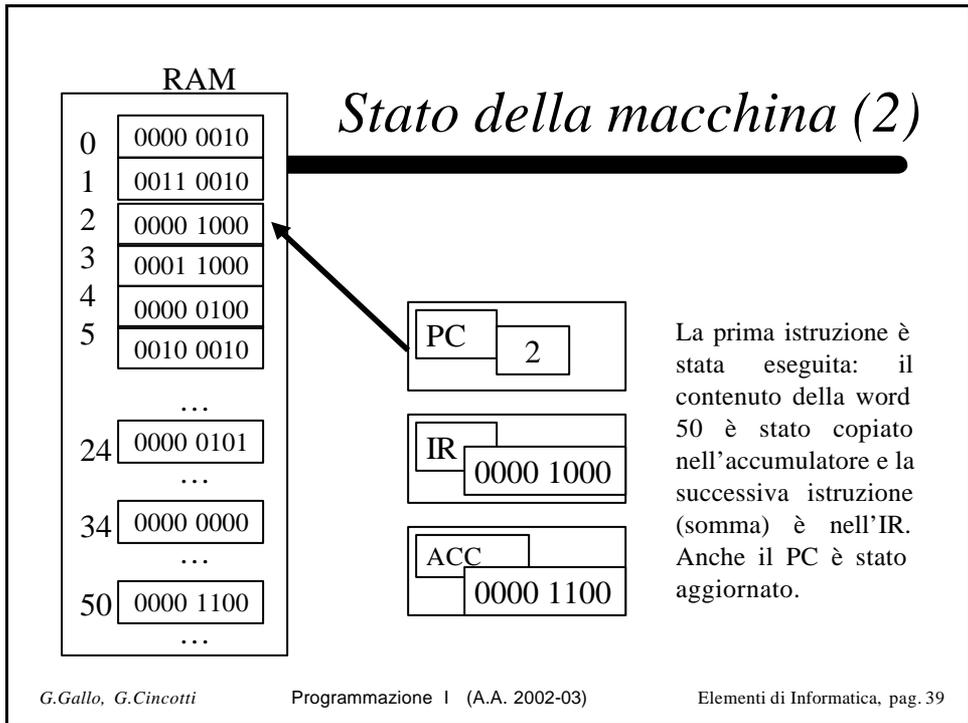
## RAM

0	0000 0010
1	0011 0010
2	0000 1000
3	0001 1000
4	0000 0100
5	0010 0010
...	
24	0000 0101
...	
34	0000 0000
...	
50	0000 1100
...	

## Stato della macchina (1)

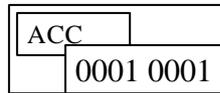
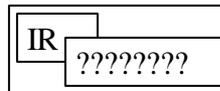
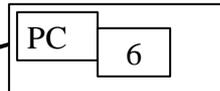


La prima istruzione (lettura dalla RAM) viene messa in IR. Il contenuto della cella 1 indica in quale cella andare a leggere il numero da mettere nell'accumulatore.



## Stato della macchina (4)

RAM	
0	0000 0010
1	0011 0010
2	0000 1000
3	0001 1000
4	0000 0100
5	0010 0010
...	
24	0000 0101
...	
34	0001 0001
...	
50	0000 1100
...	



L'istruzione di scrittura nella parola di indirizzo 34 è stata eseguita.

Il PC viene aggiornato ma non ci sono istruzioni da eseguire.

Il programma è terminato.

## Tipi di processore

- Ogni processore possiede un “set” di istruzioni macchina che costituiscono i programmi.
- Una distinzione fondamentale fra i processori è quella che li differenzia in:
  - CISC: Complex Instructions Set Computer.
  - RISC: Reduced Instructions Set Computer.

## *Tipi di processore (cont.)*

---

- La differenza è nel set di istruzioni.
  - I CISC hanno un *linguaggio macchina* formato da un numero elevato di istruzioni, anche complesse.
    - Es.: Intel x86, Motorola 68000.
  - I RISC sono dotati di istruzioni più semplici e in numero minore.
    - Es.: PowerPC, Sparc.

---

# *Fine*