

## Fondamenti di Informatica B

### Lezione n.10

- CPU-MEMORIA
- CICLO PRINCIPALE CPU
- ARCHITETTURA BASE
- UNITÀ DI ESECUZIONE E UNITÀ DI CONTROLLO

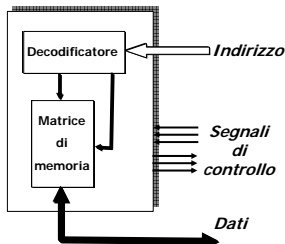
In questa lezione viene analizzata l'architettura di base di un sistema di elaborazione. Essa è formata da un'unità centrale di elaborazione (o CPU – *Central Processing Unit*) e dalla memoria che, nel modello di Von Neuman, immagazzina dati e istruzioni. Sono poi descritte le operazioni elementari svolte dalla CPU e la sua interazione con la memoria. Descriveremo poi le caratteristiche generali di una unità di controllo e le tecniche di realizzazione.

## Fondamenti di Informatica B

### Lezione n. 10

Alberto Broggi – Gianni Conte  
 A.A. 2005-2006

## La Memoria



La memoria è costituita da celle che memorizzano bit di informazione raggruppate in parole (8 bit) caratterizzate da un indirizzo. Sulla singola cella di memoria sono possibili le operazioni di lettura e scrittura.

Le linee di dato consentono l'ingresso e l'uscita delle informazioni dalla memoria.

Le linee di dato sono bidirezionali mentre le linee di indirizzo consentono di trasmettere l'informazione solo dalla CPU alla memoria.

## Il ciclo di base

Il ciclo base di una qualsiasi istruzione:

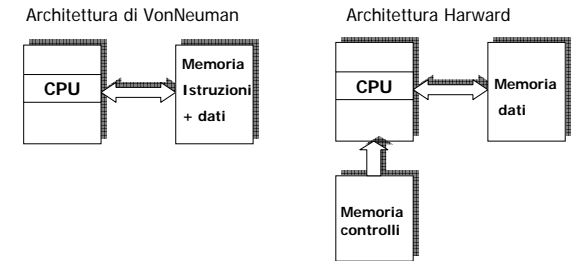
- **FETCH:** Lettura dell'istruzione in memoria.
- **ESECUZIONE:** Lettura dei dati necessari. Svolgimento dell'operazione.

La fase di FETCH è eguale per tutte le istruzioni che ovviamente si differenziano nella fase di ESECUZIONE.

Al termine della fase di esecuzione viene eseguita la fase di FETCH della istruzione successiva.

## Unità di elaborazione

Central Processing Unit + Memoria  
 Compito: Eseguire i comandi contenuti nella memoria.



## La sequenza delle istruzioni

Ogni istruzione contiene in modo esplicito o implicito l'indicazione dell'istruzione successiva.

*Modo esplicito:* il codice dell'istruzione contiene l'indirizzo della cella di memoria dove è contenuta l'istruzione da eseguire.

*Modo implicito:* l'indirizzo della istruzione successiva è contenuto in un registro interno alla CPU.

La sequenza delle istruzioni è definita dal programma.

## Interruzione

La CPU sorveglia gli altri elementi del sistema:

- Può sospendere la sequenza prevista e svolgere operazioni di maggiore urgenza
  - Al termine ritorna a svolgere la sequenza interrotta
- ⇒ INTERRUPT o INTERRUZIONE

Casi nei quali questo evento si può verificare:

- Mancanza della tensione di alimentazione
- Errori hardware
- Operazioni di I/O

Una interruzione sospende l'esecuzione del flusso principale di istruzioni e forza la CPU a svolgere una nuova sequenza di operazioni associata alla particolare richiesta

## CPU - Memoria

- La CPU (componente unico) deve essere realizzata con la migliore tecnologia possibile (anche costosa)
- La memoria principale (di grande capacità) deve essere realizzata con la tecnologia più conveniente (economicamente)

$t_{CPU}$  : tempo richiesto per la più veloce operazione della CPU  
(0,2 - 2 ns).

$t_M$  : intervallo minimo fra due accessi alla memoria  
(20 ns - 100 ns).

Questa differenza di prestazioni negli ultimi anni si è accentuata.

Le prestazioni delle CPU continuano ad crescere velocemente mentre il tempo di accesso delle memorie commerciali non è diminuito in modo significativo negli ultimi anni.

## Registri interni

Le operazioni fra registri interni avvengono con tempi dell'ordine di  $t_{CPU}$

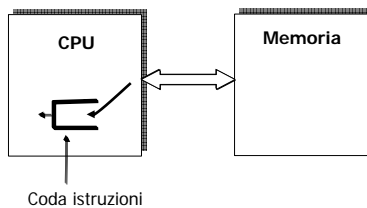
Ogni istruzione è trasformata in un certo numero di operazioni RTL interne alla CPU

Alcune CPU dispongono di centinaia di registri interni nei quali memorizzare dati temporanei

Questa tecnica utilizzata nelle architetture RISC

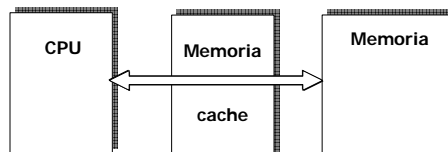
## CPU - Memoria

Altre tecniche architetturali consentono di accedere alla memoria (ad esempio per l'operazione di FETCH) contemporaneamente a fasi di esecuzione che non richiedono accessi alla memoria (PREFETCH)

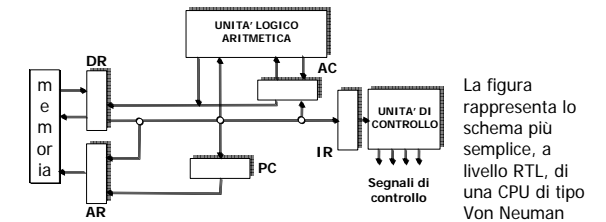


## CPU - Memoria

Per ridurre i tempi di accesso alla memoria si interpongono tra la memoria principale e la CPU memorie secondarie con prestazioni (tempi di accesso) più elevate. Questa metodologia verrà discussa in dettaglio in seguito.



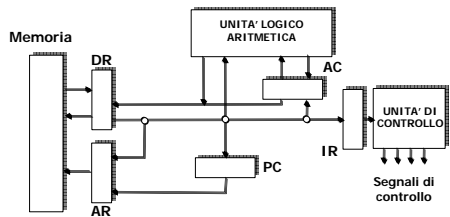
## CPU



La figura rappresenta lo schema più semplice, a livello RTL, di una CPU di tipo Von Neuman

- DR: *Data Register* - Memorizza i dati provenienti e diretti alla memoria
- AR: *Address Register* - Memorizza gli indirizzi da comunicare alla memoria
- ACC: *Accumulatore* - Immagazzina dati in ingresso e uscita all'ALU
- PC: *Program Counter* o Contatore di Programma - Memorizza l'indirizzo della istruzione da eseguire
- IR: *Instruction Register* - Memorizza il codice dell'istruzione da eseguire
- ALU: *Aritmetic Logic Unit* - Esegue le operazioni aritmetico-logiche

## Fase di Fetch



Fase di Fetch comune a tutte le istruzioni:

- ⇒ AR ← PC ;
- ⇒ DR ← M(AR) ; Lettura in memoria dell'istruzione
- ⇒ IR ← DR ; Codice di istruzione all'unità di controllo
- ⇒ PC ← PC + 1 ; Si prepara il PC per la lettura della prossima istruzione

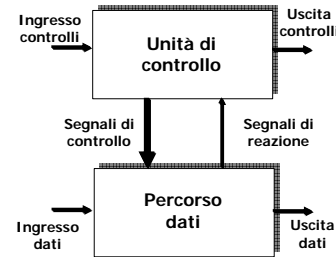
Segue la esecuzione dell'istruzione

## Struttura di base

### ESECUZIONE-CONTROLLO.

Una CPU è formata da:

- Unità di esecuzione o percorso dati (data path)
- Unità di controllo (a più livelli):
  - della sequenza delle istruzioni,
  - della esecuzione delle istruzioni.



## Tecniche di implementazione

### Controllo cablato

- Circuito sequenziale speciale che realizza la funzione desiderata
- Meno costoso e più conveniente per sistemi semplici
- Di difficile progettazione per sistemi complessi

### Controllo a microprogramma

- Microistruzioni che contengono direttamente le indicazioni per l'attivazione dei segnali
- Progettazione concorrente di parte operativa e controllo
- Progettazione strutturata
- Semplicità per l'aggiornamento dell'HW

## Controllo cablato

### GENERAZIONE DI SEQUENZE

Da un diagramma di flusso (o un programma, ...) che definisce il comportamento della parte operativa si deve generare una sequenza temporale di vettori:

$$C_1, C_2, \dots, C_n \text{ con } C_i = \{c_1, c_2, \dots, c_m\}$$

ai tempi:  $t_1, t_2, \dots, t_n$

Metodi di progetto:

1. Generatori di sequenze basati su elementi di ritardo (asincroni) o contatori (sincroni)
2. Macchine a stati finiti (progetto tradizionale di circuiti sequenziali)

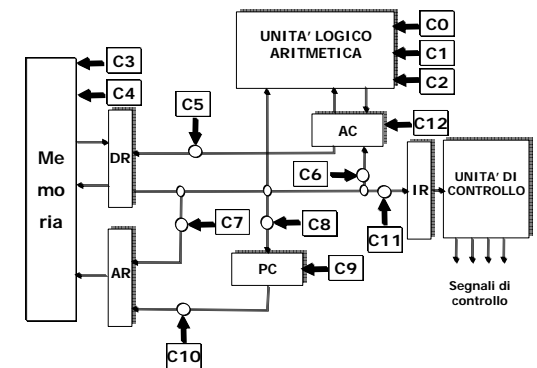
## Unità di controllo CPU

Si consideri una semplice CPU con le seguenti caratteristiche:

- 8 istruzioni a un indirizzo
- Unità aritmetica che svolge le seguenti funzioni: ADD, COMP, AND

LOAD X	AC	← M(X)
STORE X	M(X)	← AC
ADD X	AC	← AC + M(X)
AND X	AC	← AC ∧ M(X)
JUMP X	PC	← X
COMP	AC	← $\overline{AC}$
RSHIFT		Scorr. destra di AC

## Semplice CPU



## Segnali di controllo

---

C <sub>0</sub>	AC	← AC + DR
C <sub>1</sub>	AC	← $\overline{AC \wedge DR}$
C <sub>2</sub>	AC	← $\overline{AC}$
C <sub>3</sub>	DR	← M(AR) (read)
C <sub>4</sub>	M(AR)	← DR (write)
C <sub>5</sub>	DR	← AC
C <sub>6</sub>	AC	← DR
C <sub>7</sub>	AR	← DR(ADDR)
C <sub>8</sub>	PC	← DR(ADDR)
C <sub>9</sub>	PC	← PC + 1
C <sub>10</sub>	AR	← PC
C <sub>11</sub>	IR	← DR(OP)
C <sub>12</sub>	Scorr. destra di AC	