

Lezione 1 – Motivazioni del ricorso alle strutture *pipeline*

Architettura degli elaboratori

Modulo 5 - Principali linee di evoluzione
architeturale

Unità didattica 2 - Strutture *pipeline*

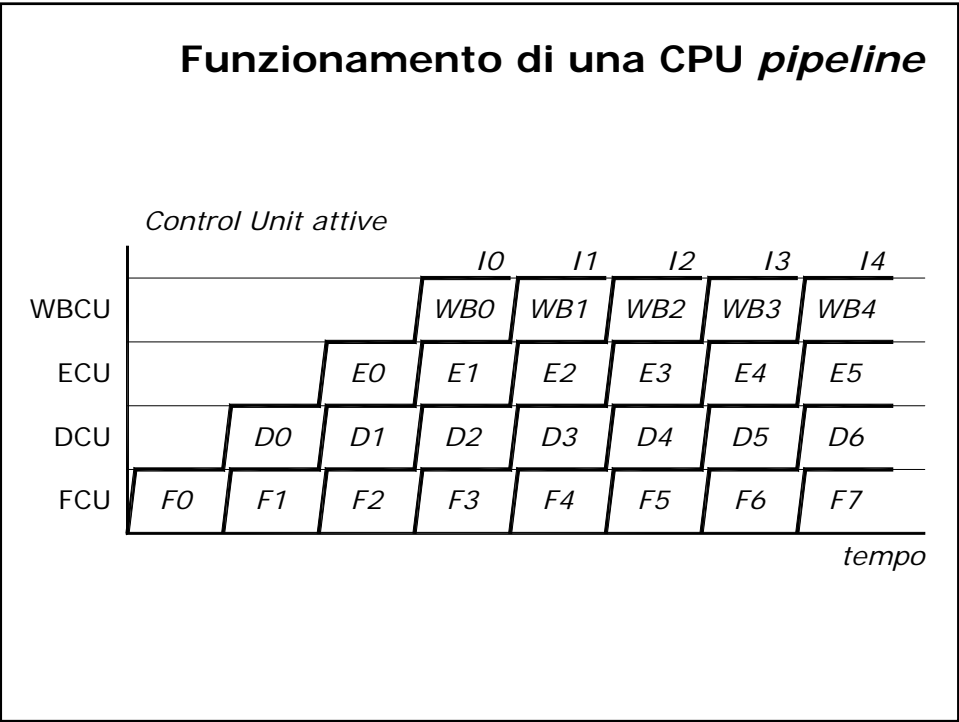
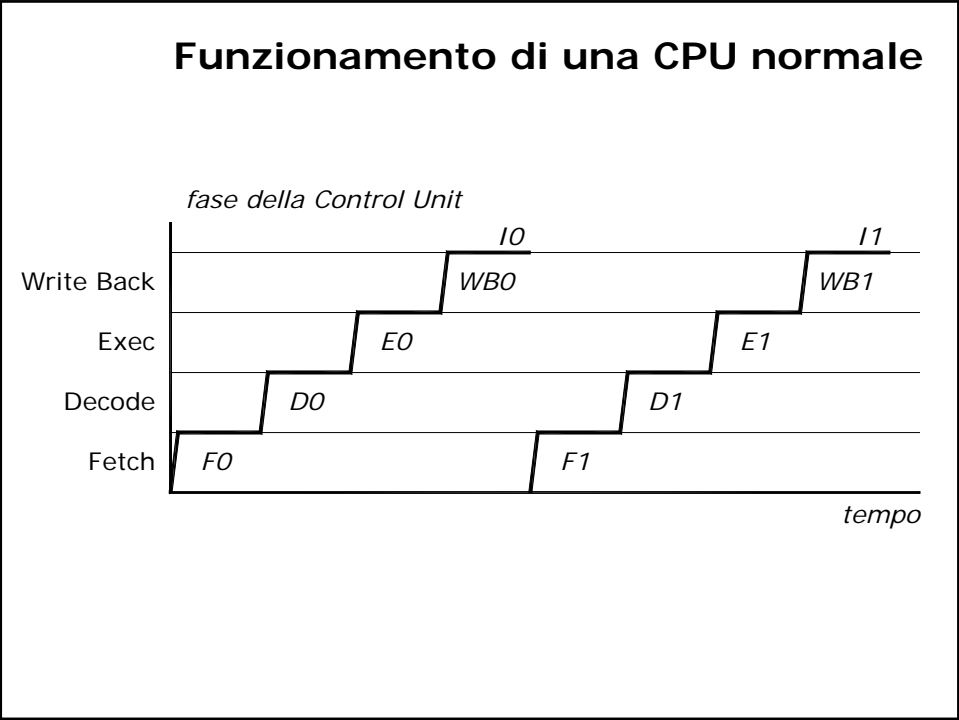
Nello Scarabottolo

Università degli Studi di Milano - Ssri - CDL ONLINE

Formulazione del problema

- **La CPU passa la sua esistenza a eseguire istruzioni macchina;**
- **ogni istruzione macchina viene portata a termine in diverse fasi:**
 - Fetch;
 - Decode;
 - Exec;
 - Write Back;
- **c'è poca sovrapposizione fra gli elementi coinvolti nelle varie fasi.**

Adottiamo una catena di montaggio (*pipeline*)



Efficacia della *pipeline*

Dopo il transitorio di avviamento, il tasso completamento istruzioni (**ICR**: *Instruction Completion Rate*) delle due CPU è:

$$ICR_{\text{pipeline}} = NS \times ICR_{\text{normal}} \quad [\text{MIPS}]$$

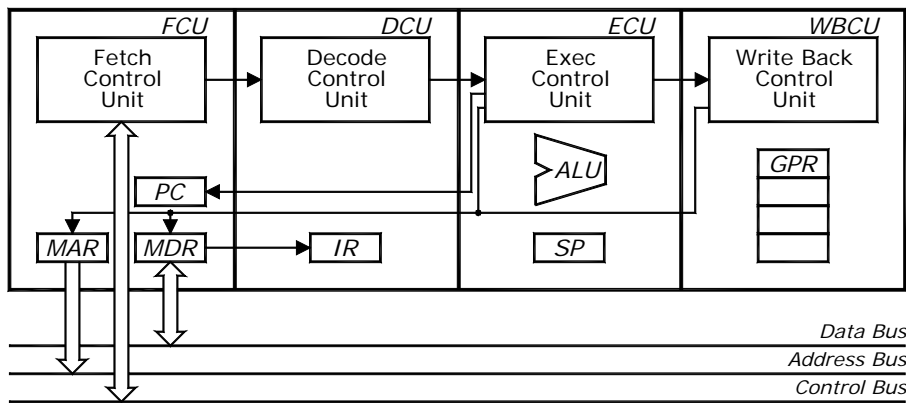
dove **NS** = n° di Stadi (*Number of Stages*) della *pipeline*.

In realtà, le fasi non hanno tutte durata esattamente uguale:

$$ICR_{\text{pipeline}} < NS \times ICR_{\text{normal}} \quad [\text{MIPS}]$$

La singola istruzione impiega circa lo stesso tempo.

Struttura di una CPU *pipeline*



In sintesi...

- Una struttura a catena di montaggio (*pipeline*) consente di velocizzare l'esecuzione aumentando il numero di istruzioni macchina completate per unità di tempo [**MIPS**].
- L'incremento di prestazioni è dovuto a un intervento **architeturale**:
 - la tecnologia di realizzazione dell'integrato - quindi la frequenza di lavoro - non cambia.

Vedremo quali sono i problemi nell'esecuzione di un programma reale.

Chiusura

**Fine della
lezione**

