



CPU a pipeline: il concetto

Marco Tarini

1

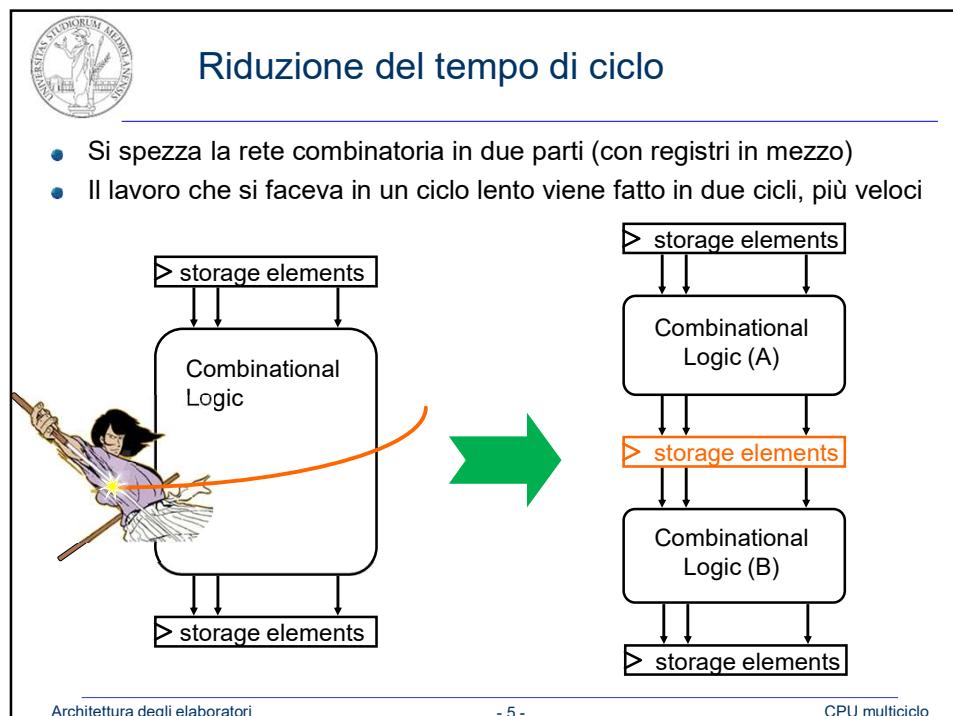


CPU monociclo (riassunto)

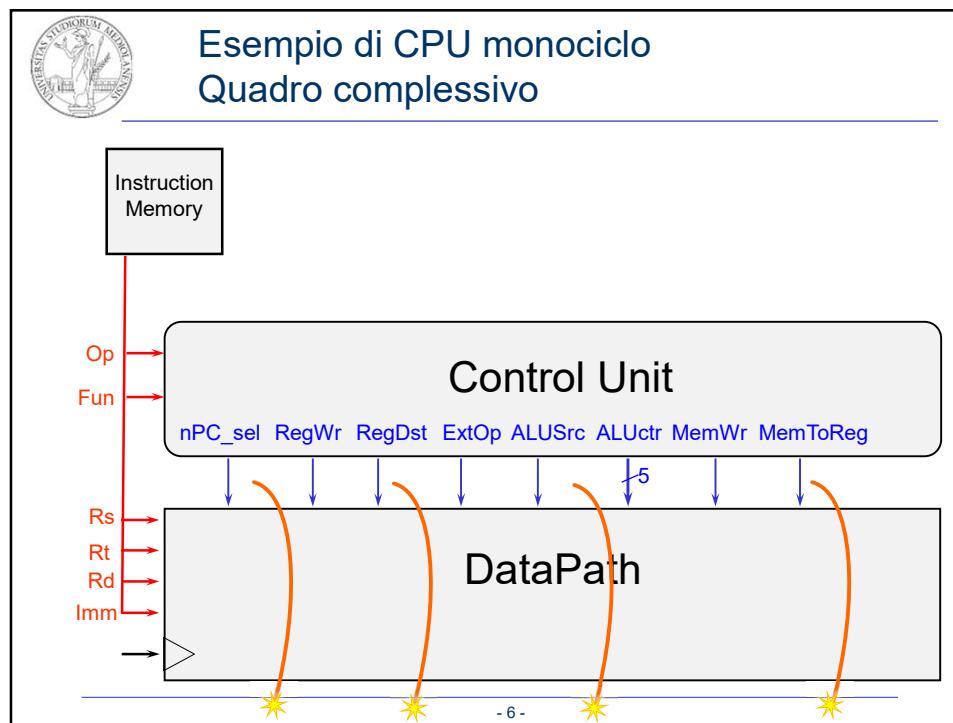
- Nel modello visto di CPU, ad ogni ciclo di clock si esegue tutta un'istruzione, comprese le fasi di
 - ▶ Fetch (accesso della memoria istruzioni al PC)
 - ▶ Decode (da parte della CU)
 - ▶ Estrazione dei valori dal banco dei registry
 - ▶ Eventuale operazione ALU, etc
 - ▶ Fino a memorizzare il risultato nei registri e nel PC (fase di "write back")
- Il limite di questa soluzione è che l'intero datapath è molto lungo
 - ▶ Periodo di clock lunghi sono richiesti, in modo da dare abbastanza tempo al datapath (e alla CU) di produrre i risultati finali da memorizzare (per es, in registri utente, PC, etc) per quando arriva il fronte del clock
- Una soluzione generale al problema di circuito combinatori troppo lunghi è "spezzarli" in sotto-circuiti, memorizzando i dati intermedi in nuovi appositi registri
 - ▶ in questa lezione panoramica, vediamo come applicare questa tecnica al datapath e più in generale alla CPU

- 2 -

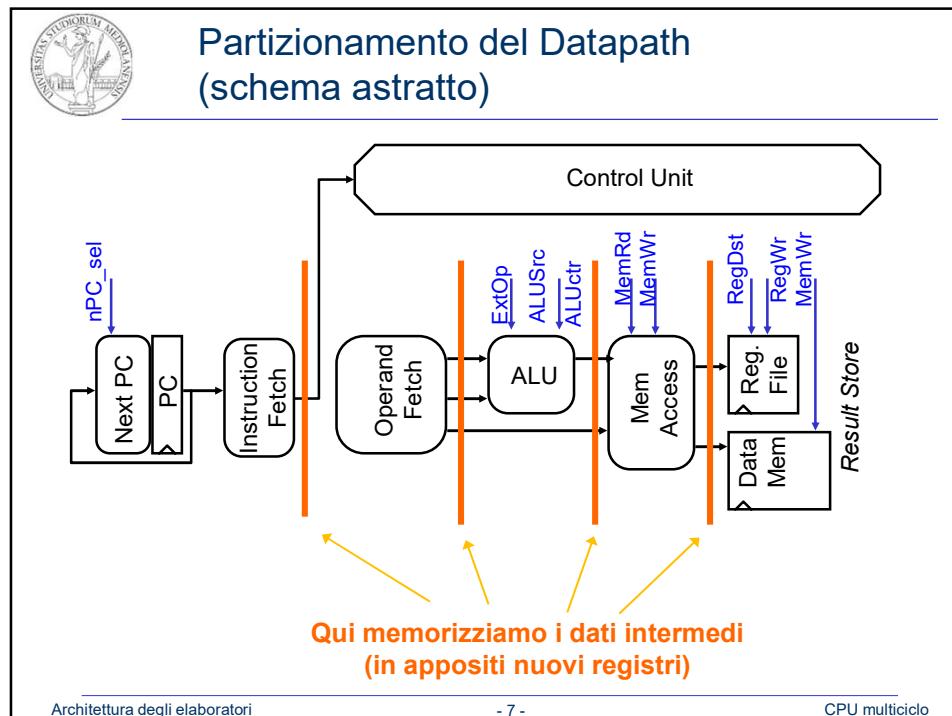
2



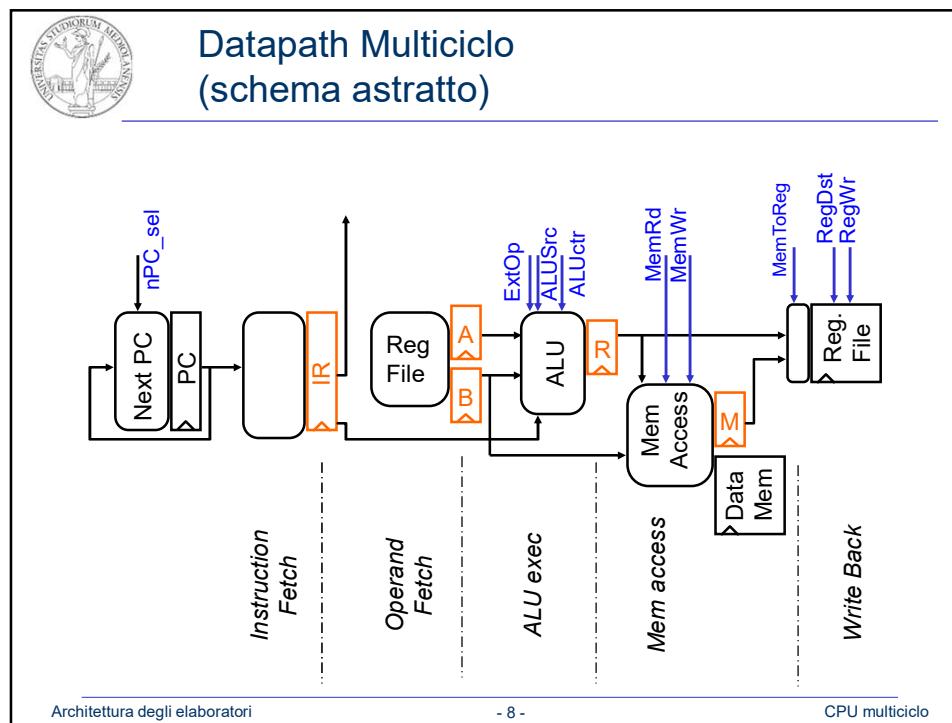
5



6



7



8



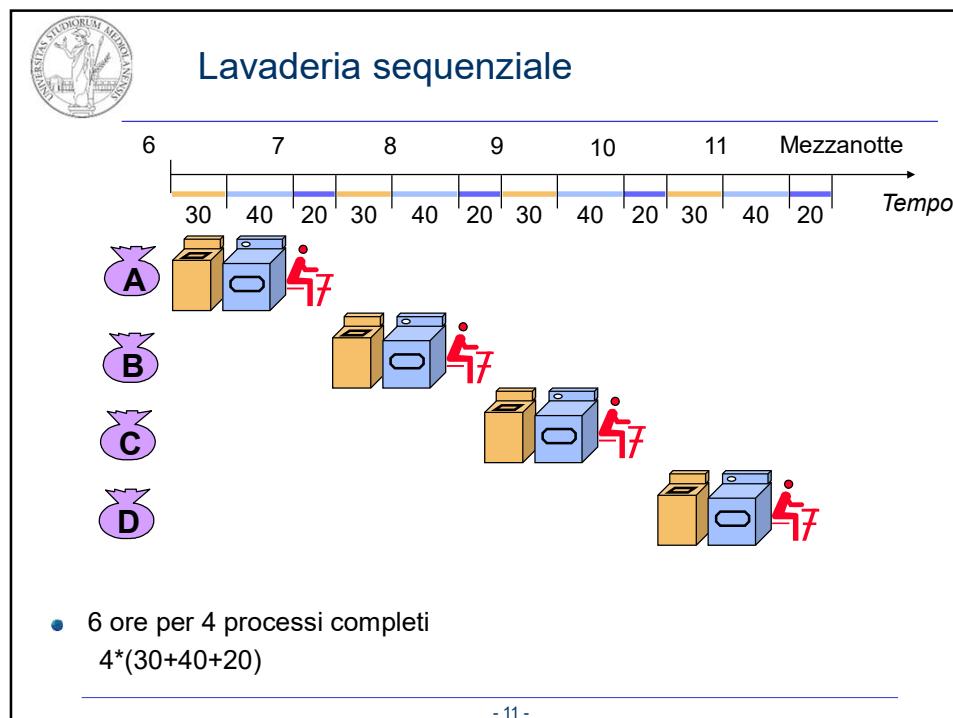
CPU multiciclo: vantaggi ulteriori

- Nella CPU a 1 ciclo, abbiamo dovuto replicare alcuni componenti del datapath, come:
 - ▶ RAM. Infatti:
 - per un'istruzione "LW", in uno stesso ciclo devo sia leggere un'istruzione dalla RAM (come per tutte le istruzioni) sia leggere un dato dalla RAM:
 - Nostra soluzione finora (poco adottata in realtà): una RAM ulteriore, separata dalla RAM dati, solo per le istruzioni :-(
 - ▶ ALU. Infatti:
 - per un'istruzione "BEQ", in uno stesso ciclo devo sia computare la condizione (di egualanza), sia eseguire una somma per calcolare il valore del nuovo PC
 - nostra soluzione finora (poco adottata in realtà): un addizionatore ulteriore, separato da quello nella ALU, solo per il computo del nuovo PC dopo il salto :-(
- Invece, nella CPU multi-ciclo, le stessa RAM e la stessa ALU possono assolvere funzioni diverse in sotto-fasi diverse di una stessa istruzione! :-)
- Il ulteriore grande opportunità consiste nello svolgere le fasi avanzate dell'istruzione corrente in parallelo con le fase iniziali dell'istruzione successiva,
 - ▶ Quest'idea porta al concetto di pipeline (catena di montaggio)
 - ▶ Vediamone il funzionamento, partendo da un esempio metaforico

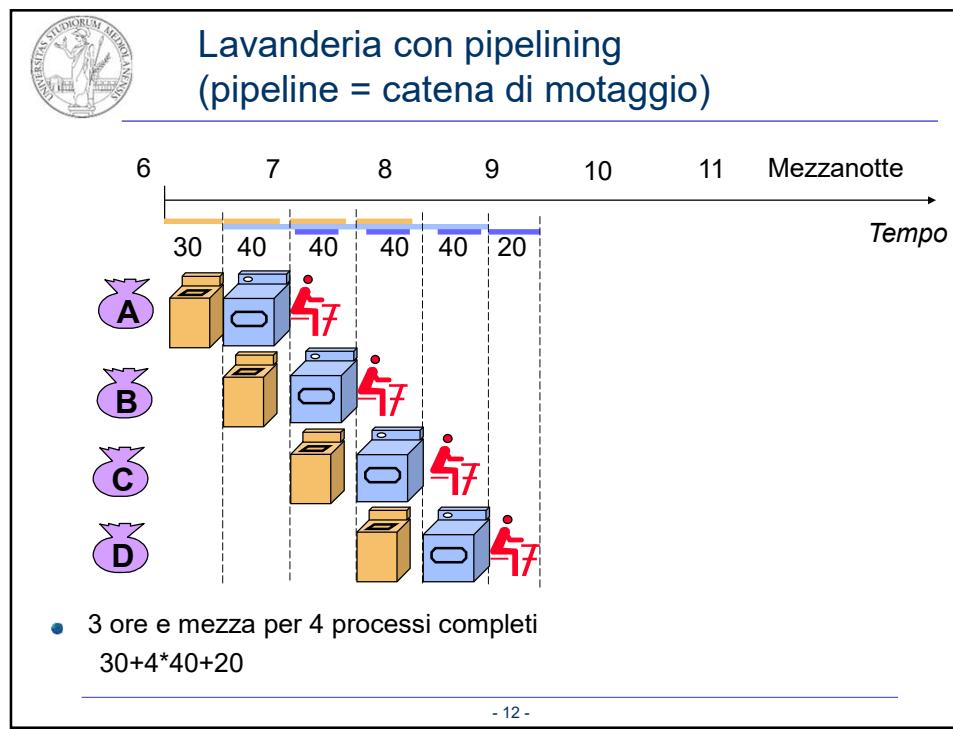


Il pipelining è una soluzione generale: esempio di una lavanderia

- Anna, Bernardo, Caterina e Davide hanno ciascuno dei panni da lavare, asciugare e piegare
 -
- ▶ Il lavaggio impiega 30 minuti
 -
- ▶ L'asciugamento impiega 40 minuti
 -
- ▶ Per la piegatura ci vogliono altri 20 minuti
 -



11



12

Pipelining: caratteristiche

- Il singolo task *non* viene velocizzato.
- L'intero processo è velocizzato
- Il tempo di una fase dipende dalla fase più lenta (detta «collo di bottiglia»)
- Diversi task che usano risorse diverse sono attivi in parallelo
- Più sono le fasi e migliore è l'efficienza complessiva (potenzialmente – vedi dopo)
- Fasi di lunghezza diversa riducono il miglioramento
- A regime, termino un intero bucato nella durata di una fase (x slot da 40min \rightarrow x bucati)

- 13 -

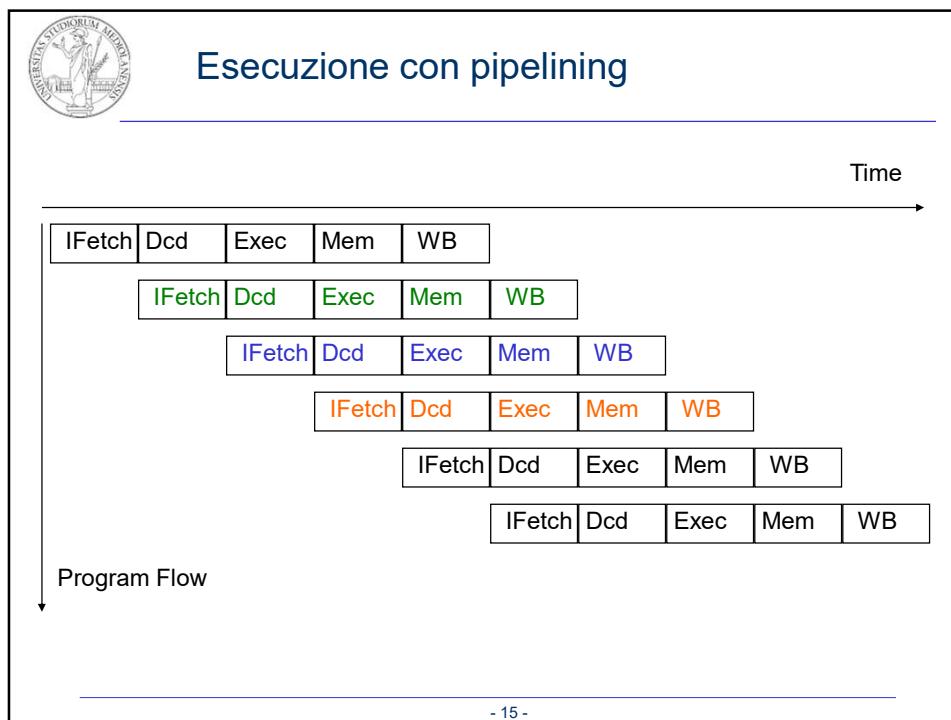
13

Fasi di esecuzione di una istruzione in una CPU pipelined

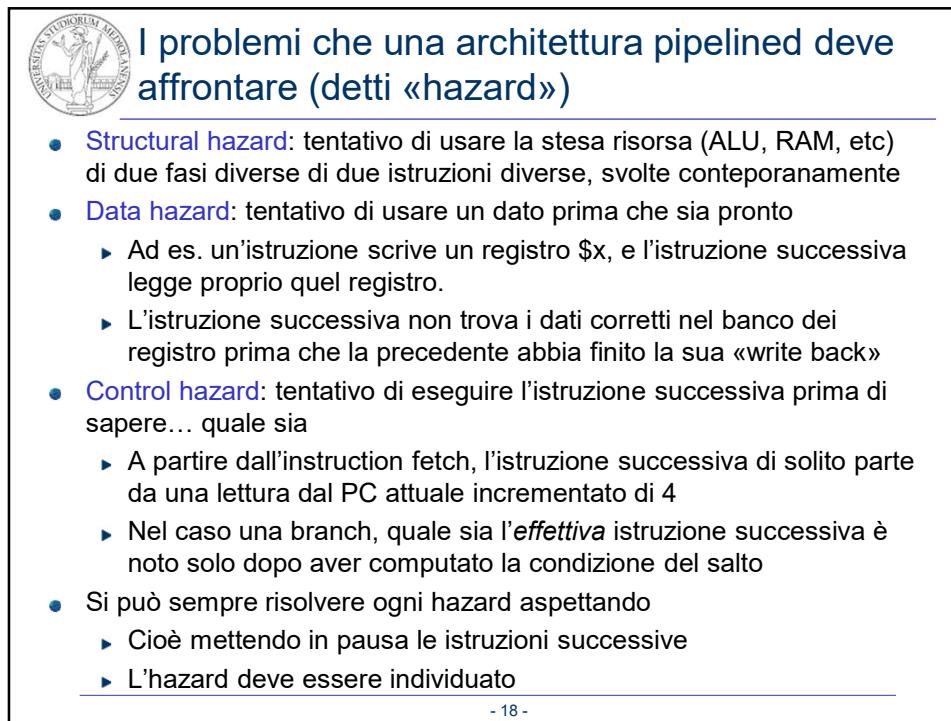
- Instruction Fetch
 - Accedere alla RAM per ottenere il valore dell'istruzione corrente
 - Registro intermedio prodotto: contiene l'**istruzione** (questo registro è detto Instruction Register, o IR)
- Decode
 - La CU produce i **bit di controllo**
 - Registro intermedio prodotto: tutti i controlli
- Operand fetch
 - Il banco di registri produce gli operandi della ALU
 - Registri intermedi prodotti: due nuovi appositi registri interni A e B, con gli operandi
- Exec
 - La ALU elabora ed espone in uscita il risultato dell'operazione logica o matematica
 - Registri intermedi prodotti: un registro R contenente il risultato
- Memory access
 - La memoria produce la parola letta (per le istruzioni che lo richiedono)
 - Registro intermedio: un altro apposito registro M
- Write back
 - Scrive dove appropriato il nuovo PC, i valori computati, la parola da scrivere in RAM

- 14 -

14



15



18