

Pipeline

Definizione

Il **pipeline** è una tecnologia utilizzata nel datapath singolo ciclo che permette di elaborare più passi di diverse istruzioni in un ciclo di clock contemporaneamente.

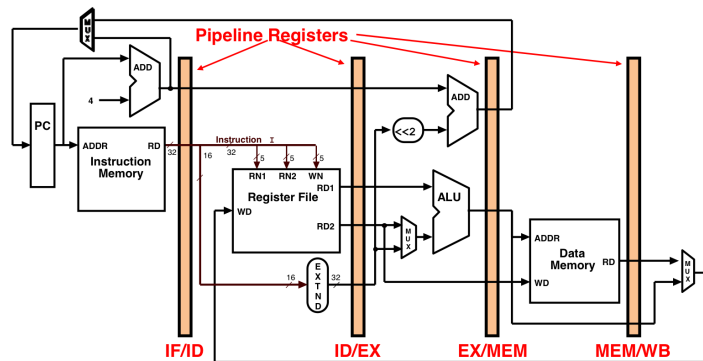
Step	Name	Description
Instruction Fetch	IF	Read an instruction from memory.
Instruction Decode	ID	Read source registers and generate control signals.
Execute	EX	Compute an R-type result or a branch outcome.
Memory	MEM	Read or write the data memory.
Writeback	WB	Store a result in the destination register.

Instruction	Steps required				
beq	IF	ID	EX		
R-type	IF	ID	EX		WB
sw	IF	ID	EX	MEM	
lw	IF	ID	EX	MEM	WB

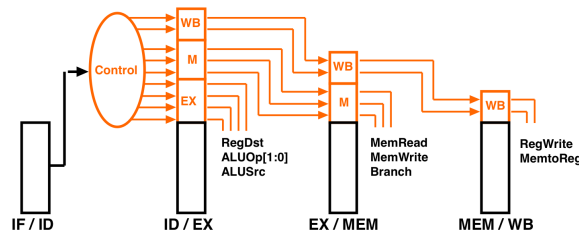
→ un'istruzione MIPS può avere al massimo 5 passi

Caratteristiche

Il processore pipeline sfrutta dei registri nei quali vengono salvati tutti i risultati ottenuti dopo un ciclo di clock. In questo modo i cicli successivi hanno a disposizione gli input necessari per l'output.



Per controllare un pipelined datapath, l'automata dei segnali fluisce passo con l'istruzione. Quindi trattando più istruzioni contemporaneamente, è necessario utilizzare i registri per salvare i risultati in ogni ciclo di clock.



Osservazione

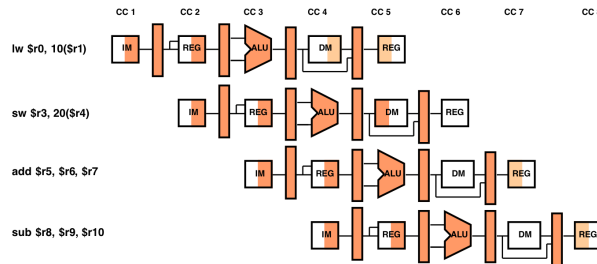
La durata di ogni ciclo è indicata dalla durata dell'operazione più lenta (solitamente MEM oppure ALU).

La scrittura su memoria/registro viene svolta nella prima metà del ciclo mentre la lettura nella seconda metà del ciclo.

Vantaggio → Più istruzioni in esecuzione nello stesso momento.

Svantaggio → **Hazard**

esempio



Pipeline hazards

Structural Hazard

Definizione

Viene utilizzata un'unità funzionale per due passi diversi contemporaneamente.

ES: memoria per istruzione e dati / register file

Soluzioni

memoria

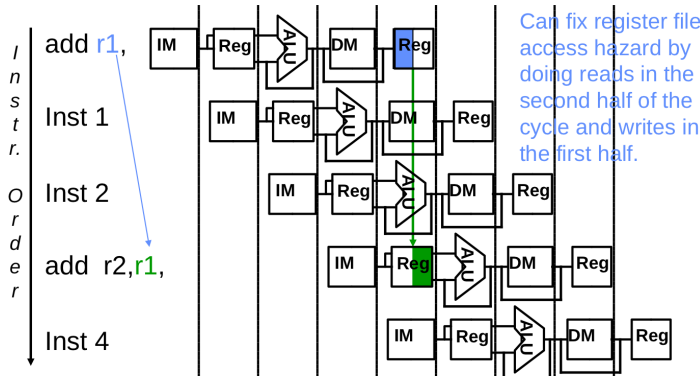
- I Delay del secondo accesso in un altro ciclo di clock
- II Memorie separate per istruzione e per dati.

register file

- I write before read → un ciclo di clock viene diviso a metà

Fronte salita : scrittura

Fronte discesa : lettura



Control Hazard

Definizione

Passare all'elaborazione della prossima istruzione prima della valutazione della condizione.

ES: l'istruzione subito dopo un branch / salti incondizionati

Soluzioni

- I Aspettare
- II controllare la decisione prima
- III Fare una predizione
- IV Rimandare il controllo

Data Hazard

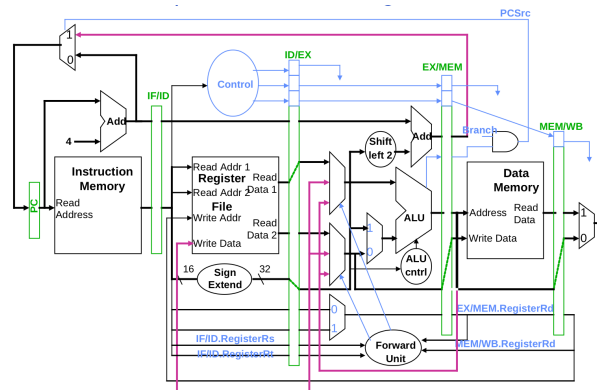
Definizione

Utilizzare un dato prima che sia pronto.

ES: utilizzo di registri non pronti / load

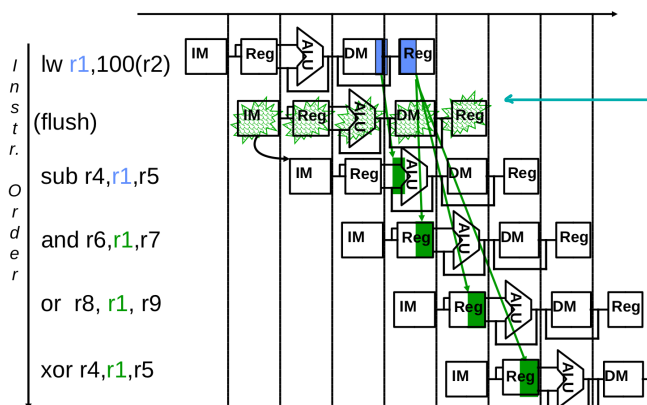
Soluzione

- I Aspettare
- II Forwarding → Aggiungere componenti al datapath per utilizzare il dato (appena è pronto) nella ALU.



hardware da aggiungere per l'input della ALU appena il dato è disponibile

esempio



serve perché serve il dato in r1 in un ciclo prima rispetto a quando il dato è pronto