Eccezioni e Pipeline

Architettura
degli Elaboratori e delle Reti,
Turno I

Alberto Borghese
Università degli Studi di Milano
Dipartimento di Scienze dell'Informazione
email: borghese@dsi.unimi.it

Eccezioni ed Interrput

Alterano il funzionamento di un programma (funzionalmente equivalenti ad una jump).

Eccezioni. Generamente internamente al processore (e.g. overflow).
Interrupt. Generate esternamente al processore (e.g. richiesta di attenzione da parte di una periferica).

Tipo di evento	Provenienza	Terminologia MIPS
Richiesta di un dispositivo di I/O	Esterna	Interrupt
Chiamata al SO da parte di un programma	Interna	Eccezione
Overflow aritmetico	Interna	Eccezione
Uso di un'istruzione non definita	Interna	Eccezione
Malfunzionamento dell'hardware	Entrambe	Eccezione o Interruzione

Hardware addizionale

Registro EPC: è un registro a 32 bit utilizzato per memorizzare l'indirizzo dell'istruzione coinvolta.

Registro causa: è un registro utilizzato per memorizzare la causa dell'eccezione; in MIPS sono 32 bit:
- bit 0 = 0 istruzione indefinita.
- bit 0 = 1 overflow aritmetico.

Segnali di controllo:
EPCWrite – scrittura nel registro EPC.
CausaWrite – scrittura nel registro Causa.
CausaInt – Dato per il registro Causa.

Set degli input della FSM modificato:
Aggiunta di Overflow.

Tipo di risposta ad un'eccezione

E' software (SO)

Vettorializzata: ciascuna eccezione rimanda ad un indirizzo diverso del SO. Gli indirizzi sono spaziati equamente (8 parole). Dall'indirizzo si può ricavare la causa dell'eccezione.

Tramite registro: detto registro **causa**. Il SO ha un unico entry point per la gestione delle eccezioni. La prima istruzione è di decodifica della causa dell'eccezione. L'entry point è forzato tramite PCSource = 11.

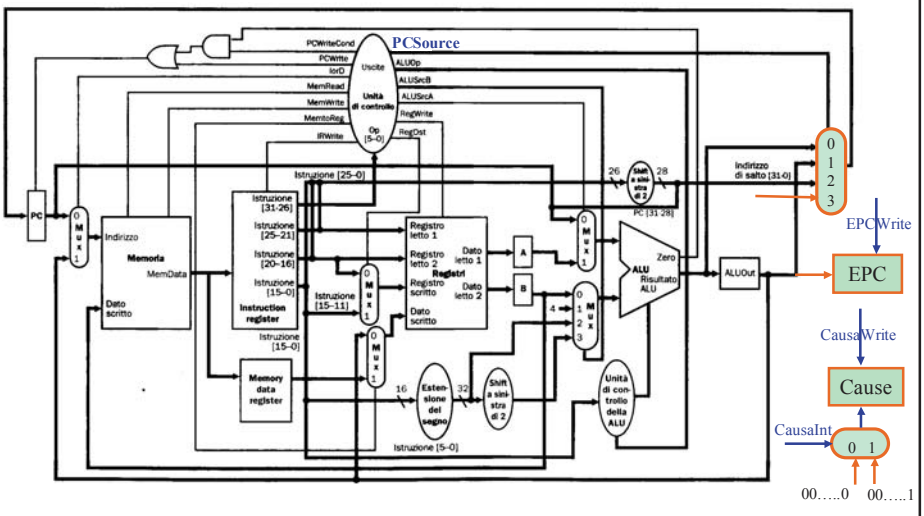
Azioni in risposta ad un'eccezione

Le eccezioni vengono gestite dal SO.

1) Salvataggio dell'indirizzo dell'istruzione incriminata (PC-4) nell'EPC.
2) Trasferimento del controllo al SO (questo potrà eventualmente terminare il programma segnalando errore).

Collegamenti HW per la risposta ad un'eccezione

Salvataggio del PC: PC = PC – 4 PC → EPC; Selezione della causa

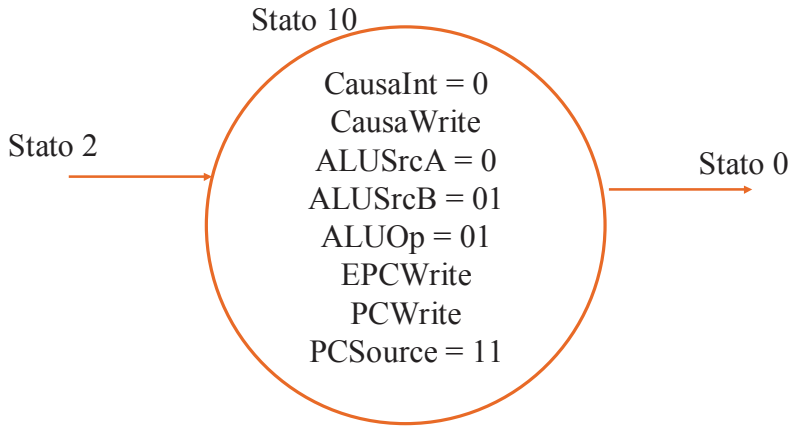


Modifiche alla FSM della CPU

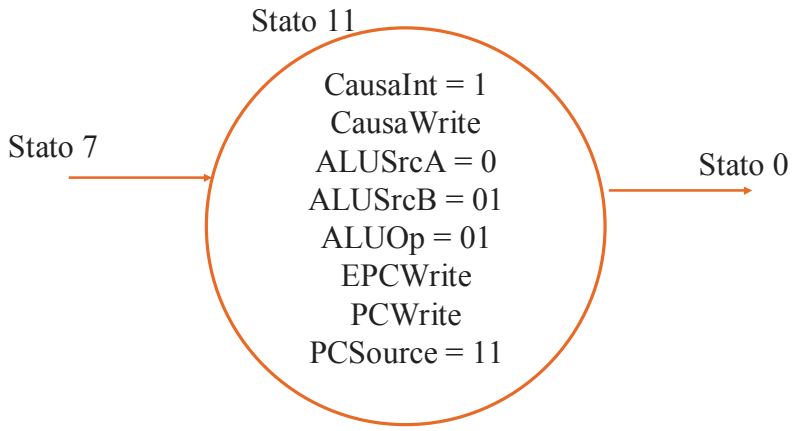
Istruzione indefinita.
Non esiste, al passo 2 (decodifica), uno stato futuro valido.
Si aggiunge un nuovo stato futuro indicato con 'altro', è lo Stato corrispondente al verificarsi dell'eccezione.

Overflow aritmetico.
Al passo 4 di esecuzione dell'operazione, lo stato futuro è scelto in funzione del segnale di overflow (input alla FSM). Ho già eseguito la somma e sto scrivendo il risultato nel register file.

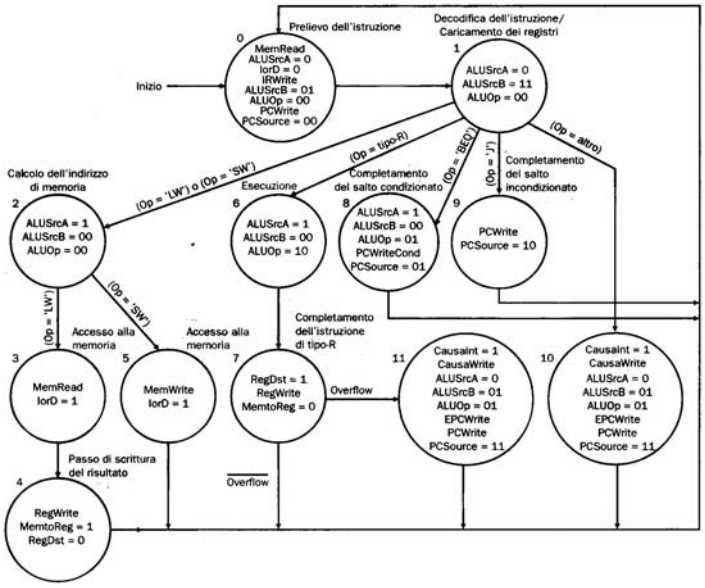
Stato OpCode not valid



Stato Overflow



FSM per la CPU multi-ciclo con gestione delle eccezioni



Il coprocessore 0

Contiene indirizzi per la gestione delle eccezioni e per la gestione della memoria

Nome del registro	Numero del registro in coprocessore 0	Utilizzo
Bad/Addr	8	Registro contenente l'indirizzo di memoria a cui si è fattoriferimento
Stato	12	Maschera delle interruzioni e bit di abilitazione
Causa	13	Tipo dell'interruzione e but dell'interruzione pendente
EPC	14	Registro contenente l'indirizzo dell'istruzione che ha causato l'interruzione.

Il coprocessore 0 - Istruzioni

Set di istruzioni che lavora sul coprocessore 0:

lwc0 $<reg_c0> <offset>($reg)
swc0 $<reg_c0> <offset>($reg)

mfc0 $<reg>, $<reg_c0> pseudo-istruzione
mtc0 $<reg_c0>, $<reg> pseudo-istruzione

La pipeline

Intuizione della pipe-line

Anna, Bruno, Carla e Dario devono fare il bucato.

Devono lavare, asciugare, piegare e mettere via ciascuno un carico di biancheria

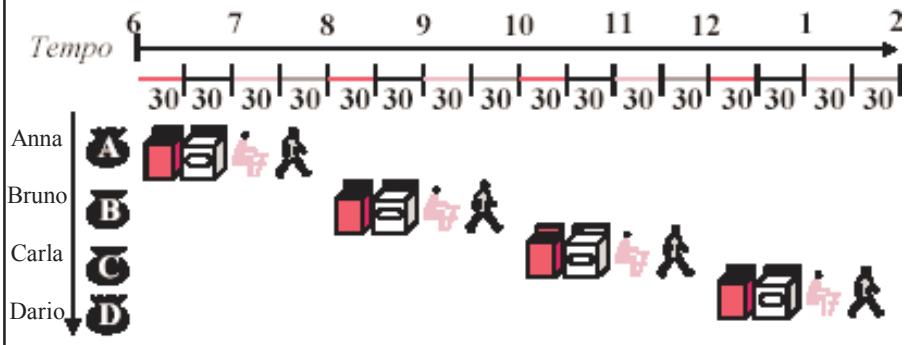
La lavatrice richiede 30 minuti.

L'asciugatrice richiede 30 minuti.

Stirare richiede 30 minuti.

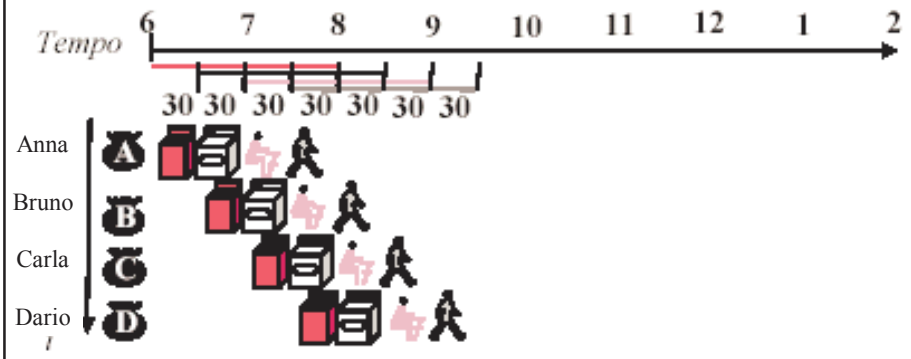
Piegare e mettere via richiede 30 minuti.

La lavanderia sequenziale



In totale vengono richieste 8 ore.

Lavanderia con pipe-line



In totale vengono richieste 3.5 ore.

Osservazioni sulla pipe-line

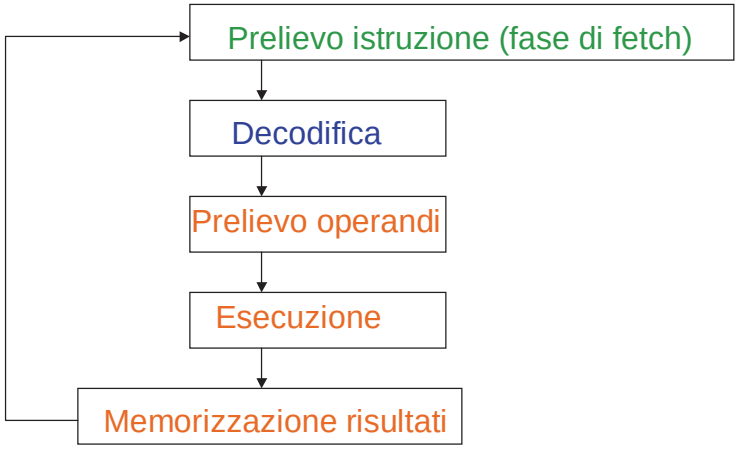
Il tempo di ciascuna operazione elementare non viene ridotto.

Gli stadi della pipe-line lavorano in contemporanea perché utilizzano unità funzionali differente.

Le unità funzionali lavorano sequenzialmente (in passi successivi) su istruzioni successive.

Viene aumentato il throughput.

Ciclo di esecuzione di un'istruzione

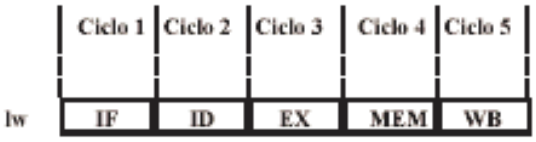


Le istruzioni richiedono 5 passi → 5 stadi della pipe-line

Utilizzo unità funzionali della CPU

Passo esecuzione	ALU	Memoria	Register File
Fase fetch	Yes	Yes	NO
Decodifica	Yes (salto)	NO	Yes
Exec I - beq	Yes	NO	NO
Exec I - j	NO	NO	NO
Exec I - R	Yes	NO	NO
Exec II - R	Yes	NO	Yes
Exec I - sw	Yes (calcolo indirizzo)	NO	NO
Exec II - sw	NO	Yes	NO
Exec I - lw	Yes (calcolo indirizzo)	NO	NO
Exec II - lw	NO	Yes	NO
Exec III - lw	NO	NO	Yes

I 5 stadi della pipeline



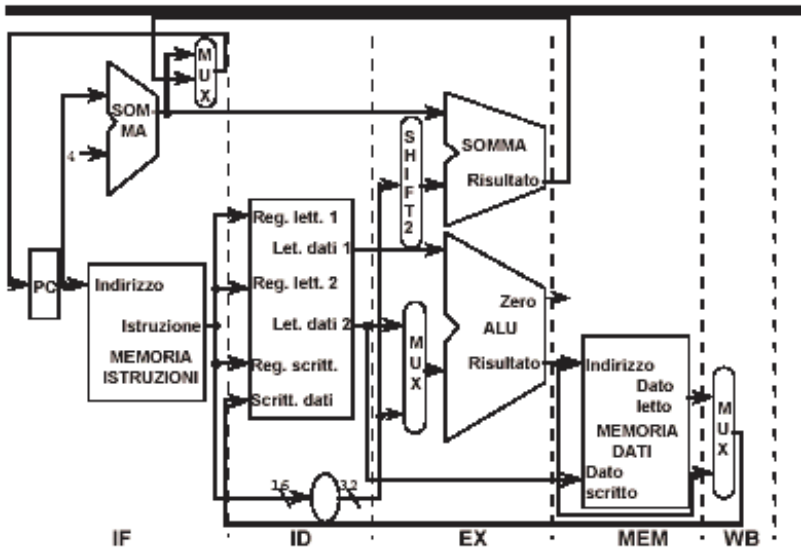
Nomi degli stadi di pipeline:

- **IF: prelievo istruzione**
- **ID: decodifica istruzione/lettura registri**
- **EX: esecuzione**
- **MEM: accesso a memoria**
- **WB: riscrittura**

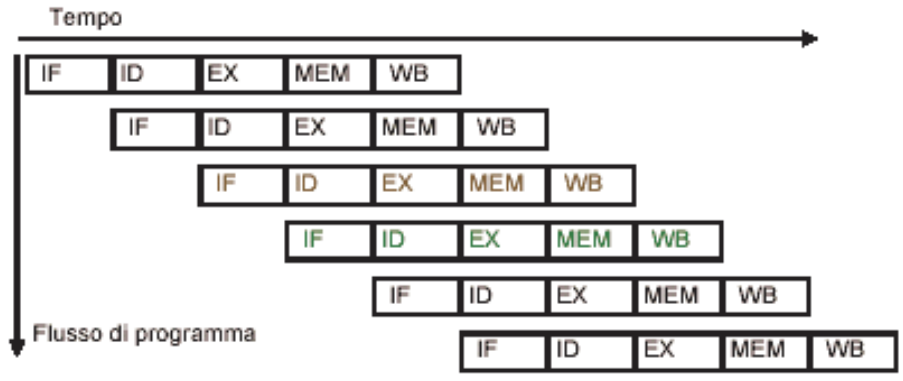
Durata dei passi:

Memoria (istruzioni, dati) = 2ns.
ALU = 2ns.
Lettura/Scrittura registri = 1ns.

Torniamo alla CPU singolo ciclo



Rappresentazione convenzionale della pipeline

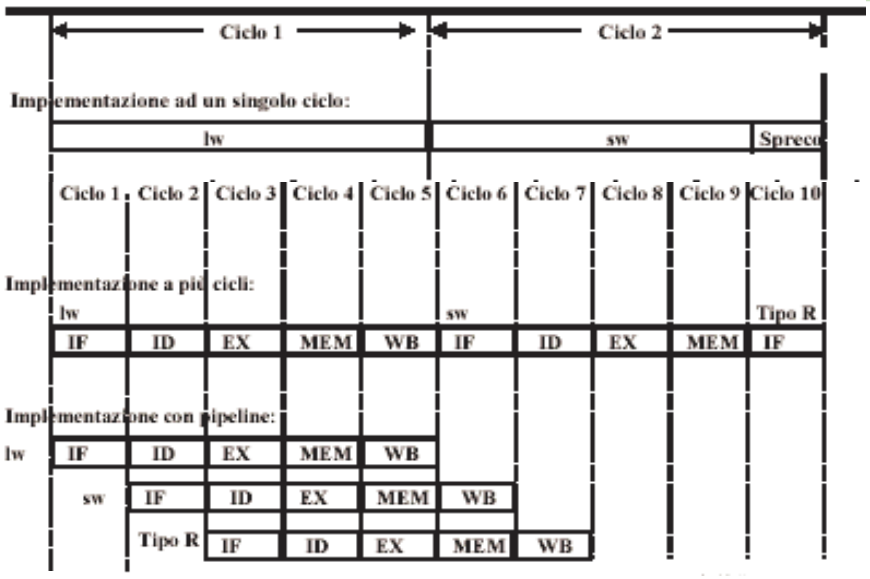


Rappresentazione grafica della pipeline

Esempio: add $s0, $t0, $t1

I rettangoli grigi a destra indicano lettura, a sinistra indicano scrittura. I componenti bianchi, indicano il loro non utilizzo.

I vantaggi della pipeline



Pipeline per l'istruzione lw

Passo esecuzione	ALU	Memoria	Register File
IF (Fase fetch)	Yes	Yes	NO
ID (Decodifica)	Yes (salto)	NO	Yes
EX (Esecuzione)	Yes (calcolo indirizzo)	NO	NO
MEM (Accesso memoria)	NO	Yes	NO
WB (riscrittura)	NO	NO	Yes

Passi della prima istruzione lw	IF	ID	EX	MEM	WB	
.....						
lw $t0, 8($t2)	Mem, ALU	RF	ALU	Mem	RF	
lw $t1, 12($t2)		Mem, ALU	RF	ALU	Mem	RF
.....						

Miglioramento delle prestazioni

Il miglioramento massimo è una riduzione del tempo di un fattore pari al numero di stadi della pipe-line.

Nell'esempio precedente (2 istruzioni di lw), il tempo richiesto con la pipe-line è di 12ns contro i 20ns senza pipe-line. Il miglioramento teorico, prevedrebbe 4ns. Allora?

Throughput!! Miglioramento relativo al lavoro globale.

Lo stallo della pipeline

Passi della prima istruzione lw	IF	ID	EX	MEM	WB	
.....						
lw $t0, 8($s0)	Mem, ALU	RF	ALU	Mem	RF	
lw $t1, 12($s0)		Mem, ALU	RF	ALU	Mem	RF
lw $t2, 16($s0)			Buco	Buco	Buco	Mem,ALU
.....						

I buchi (o bubble) inducano degli istanti di clock in cui non può essere eseguita l'istruzione successiva ➔ La pipeline va in stallo.

MIPS e pipe-line

Fase di fetch semplificata: tutte le istruzioni hanno la stessa lunghezza.

Numero ridotto di formati, i registri sono sempre nella stessa posizione (si può decodificare il codice operativo e leggere i registri).

Non ci sono operazioni sui dati in memoria: se utilizzo la ALU per effettuare dei calcoli, non dovrò accedere alla memoria. Se utilizzo la ALU per calcolare l'indirizzo, accederò alla memoria nell'istante successivo.

Allineamento delle istruzioni al byte.

Problemi? (Hazard = azzardo)

Se non posso eseguire un'istruzione nell'istante di clock immediatamente successivo sono in presenza di un *hazard* (criticità).

Passi della prima istruzione lw	IF	ID	EX	MEM	WB	
.....						
lw $t0, 8($t2)	Mem, ALU	RF	ALU	Mem	RF	
lw $t1, 12($t2)		Mem, ALU	RF	ALU	Mem	RF
lw $t1, 12($t2)			???			
.....						

Criticità

Strutturali:
•Dovrei utilizzare la stessa unità funzionale due volte nello stesso passo.
•Le unità funzionali non sono in grado di supportare le istruzioni (nelle diverse fasi) che devono essere eseguite in un determinato ciclo di clock.

Controllo:
• Dovrei prendere una decisione (sull'istruzione successiva) prima che l'esecuzione dell'istruzione sia terminata (e.g. branch).

Dati:
• Dovrei eseguire un'istruzione in cui uno dei dati è il risultato dell'esecuzione di un'istruzione precedente.

add $s0, $t1,$t1
add $s2, $s0, $t3

Introduzione sulla pipeline

La tecnica di progettazione mediante pipeline sfrutta il parallelismo tra le istruzioni poste in un flusso sequenziale.

Non aumenta la velocità di esecuzione della singola istruzione, ma del lavoro nel suo complesso (throughput).

L'impossibilità di iniziare ad eseguire un'istruzione determina uno stallo della pipeline.

Criticità in una pipeline sono suddivisibili in: strutturali, di controllo e sui dati. La soluzione a tutte le criticità potrebbe essere … aspettare.