



Corso di Laurea in Informatica  
Architettura degli elaboratori  
a.a. 2018-19



# Architettura degli Elaboratori 2018-2019

## Datapath multi-cycle Eccezioni

Claudia Raibulet  
[claudia.raibulet@unimib.it](mailto:claudia.raibulet@unimib.it)

## Eccezioni e interruzioni

- Durante l'esecuzione delle istruzioni si possono verificare eventi inattesi: eccezioni e/o interruzioni
- Eccezione:
  - Evento sincrono, generato all'interno del processore e provocato da problemi nell'esecuzione di un'istruzione
  - Esempi: overflow, istruzione non valida, errori, pagine non presenti in memoria, ...
- Interruzione:
  - Evento asincrono che giunge dall'esterno del processore
  - Di solito arriva da un'unità di I/O utilizzato per comunicare alla CPU il verificarsi di certi eventi
  - Esempi: la terminazione di un'operazione di I/O la cui esecuzione era stata richiesta dalla CPU, ...

## Eccezioni e interruzioni

- Eccezione:
  - Causate da eventi interni al processore
  - Sincrone rispetto al programma in esecuzione
  - La condizione di eccezione deve essere risolta da un gestore di eccezioni (e.g., handler di eccezioni)
  - Se la condizione di eccezione e' risolvibile, allora il programma puo' riprendere l'esecuzione; altrimenti, il programma termina prima della sua fine
- Interruzione:
  - Causate da eventi esterni al processore
  - Asincrone rispetto al programma in esecuzione
  - Sono gestite tra due istruzioni consecutive
  - Si sospende l'esecuzione del programma utente, si gestisce l'interruzione e poi si riprende l'esecuzione del programma utente

## Gestione di eccezioni e interruzioni

- Il controllo del processore deve gestire gli eventi inattesi
- Tutti i processori eseguono i seguenti passi per gestire un eccezione/interruzione:
  - Interruzione dell'esecuzione del programma corrente
  - Salvataggio parziale dello stato di esecuzione corrente (e.g., PC) – per riprendere eventualmente l'esecuzione dell'esecuzione del programma corrente se possibile
  - Salto a una routine del sistema operativo (SO) per gestire l'eccezione/interruzione (tale routine e' chiamata di solito handler o gestore delle eccezioni)
  - Esecuzione della routine del SO
  - Se possibile, ripristino dello stato di esecuzione del programma e continuazione dell'esecuzione del programma

- Problema: come capire l'evento inatteso verificatosi?
- Due possibili soluzioni:
  - Indirizzo fisso - Registro dedicato (e.g., Cause) - il controllo della CPU, prima di saltare all'handler del SO (a un indirizzo fisso), deve salvare in un registro interno un identificatore numerico del tipo di eccezione verificatosi. L'handler accederà al registro interno per determinare la causa dell'eccezione
  - Interruzioni vettorizzate - esistono handler diversi per eccezioni/interruzioni differenti. Il controllo della CPU sceglie l'handler corretto, saltando all'indirizzo corretto. A questo scopo, viene predisposto un vettore di indirizzi, uno per ogni tipo di eccezioni/interruzioni, da indirizzare tramite il codice numerico dell'eccezione/interruzione

## Gestione delle eccezioni

- Nel MIPS viene adottata la prima soluzione, usando un registro, denominato Cause, per memorizzare il motivo dell'eccezione
- L'indirizzo dell'istruzione corrente che ha causato l'eccezione viene salvato nel registro Exception Program Counter (EPC)

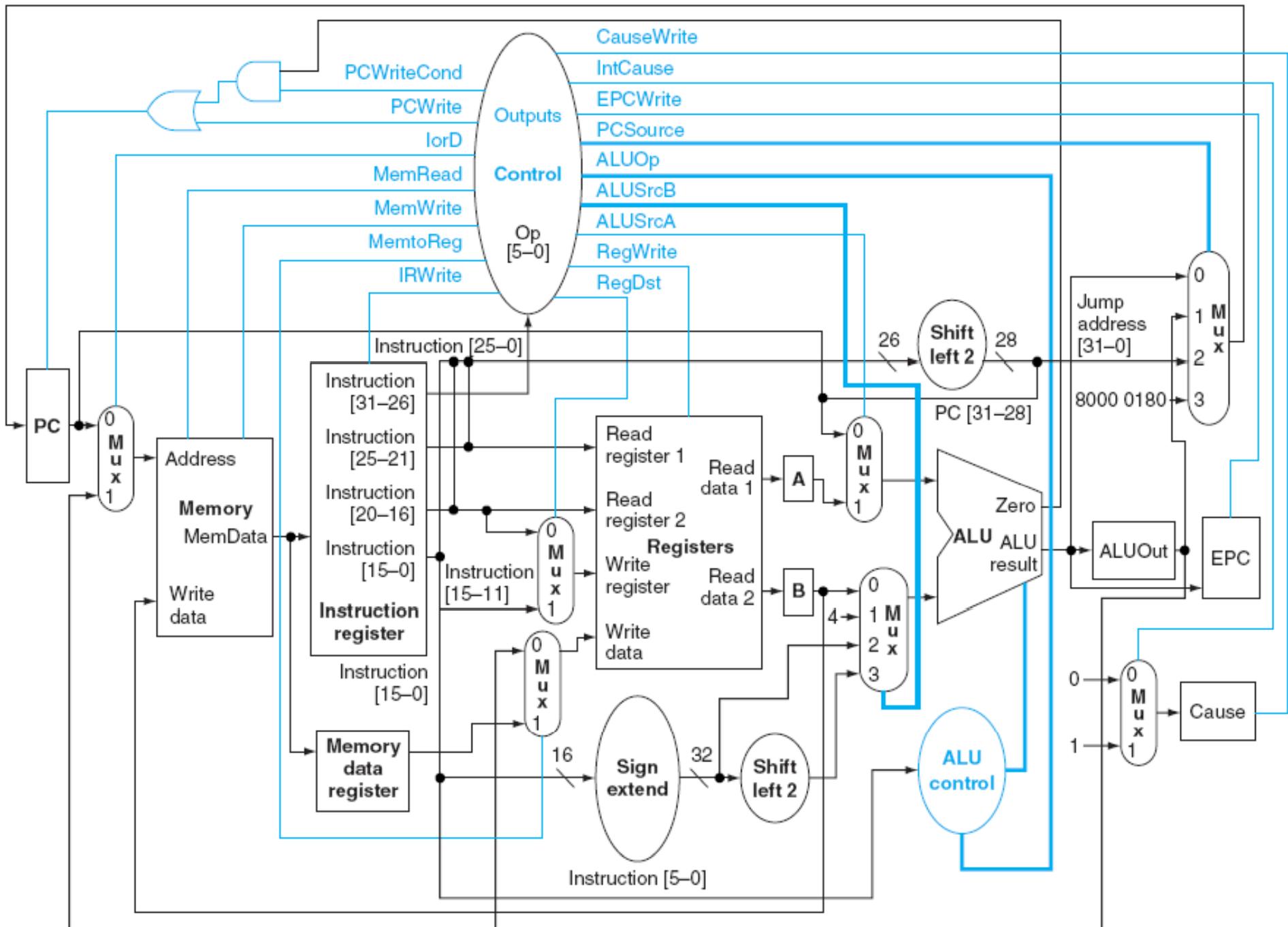
## Gestione delle eccezioni - esempi

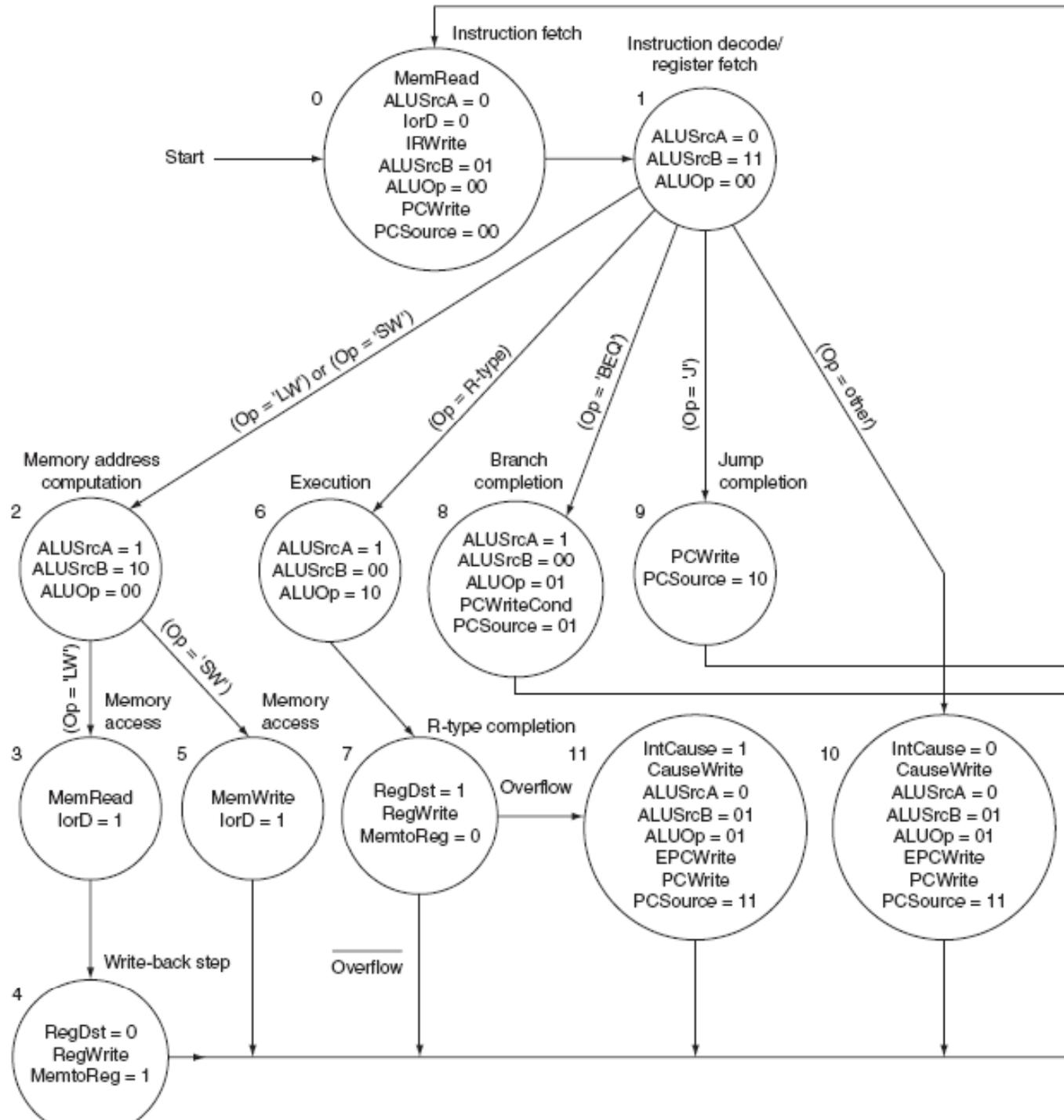
- Consideriamo solo 2 cause di eccezione:
  - Istruzione non valida (generata dall'unità di controllo a valle della decodifica dell'istruzione)
  - Overflow (generata dall'ALU nella fase execute di un'istruzione R-type)

## Gestione delle eccezioni in MIPS

- Passi da eseguire:
  - Individuare l'evento inatteso, la causa dell'eccezione e salvarla in un registro dedicato denominato Cause
  - Interrompere l'esecuzione corrente
  - Salvare l'indirizzo dell'istruzione corrente nel EPC (EPC  $\leftarrow$  PC-4)
  - Saltare a un gestore delle eccezioni del SO che si trova a un indirizzo fisso per gestire l'eccezione
- Osservazioni:
  - Il MIPS non salva nessun altro registro oltre al PC
  - E' compito della routine salvare altre porzioni dello stato corrente del programma se necessario
  - Esistono CPU dove questo salvataggio viene prima di saltare alla routine (e.g., approcci CISC)

- Istruzione non valida:
  - L'unità di controllo rileva tale eccezione sulla base del OPICODE di un'istruzione
- Overflow:
  - Segnale che arriva all'unità di controllo dalla ALU
- Da aggiungere 2 nuovi stati alla FSM del datapath
- Tali stati devono:
  - Salvare in EPC il valore PC - 4
  - Salvare nel registro Cause la causa dell'eccezione (0 o 1 in questo caso perché consideriamo 2 esempi di eccezioni)
  - Salvare in PC l'indirizzo del gestore delle eccezioni





## Da leggere

- **Chapter 5: The processor: Datapath and Control – disponibile sul sito elearning del corso – pag. 340 -346**