

Eccezioni

Mon, 9 May

Un'eccezione è un cambio di esecuzione rispetto al **default**.

Quando un'operazione della sequenza prevista presenta errori o malfunzionamenti, avviene un cambiamento nel percorso necessario a gestire tale eccezione.

La gestione di un'eccezione è a sua volta un **software**.

Le eccezioni si dividono in due tipi:

- ▼ **Eventi Sincroni (Exception)**
Errori nel software, ad esempio un'istruzione sbagliata.
Sono sincronizzate rispetto all'esecuzione del programma, quindi rispetto al clock.
- ▼ **Eventi Asincroni (Interrupt)**
Eventi o errori che vengono "da fuori", ad esempio eventi relativi all I/O
Non sono sincronizzate con la corrente esecuzione, vengono gestite tramite due istruzioni consecutive.

Gestione delle Eccezioni

Gestione a Livello Software

MIPS fornisce ed imposta un **registro fisso dedicato** chiamato `Cause` che indicherà, tramite *identificatore numerico*, il **tipo** di eccezione.

Il *Datapath* "conosce" l'**indirizzo fisso** dei software di gestione delle eccezioni. Esistono architetture che implementano *diversi gestori*, e il processore "sceglie" che software lanciare per gestire il particolare tipo di eccezione.

È inoltre necessario un ulteriore registro dedicato, **Exception Program Counter (EPC)**, per salvare lo stato dell'esecuzione corrente nel momento in cui avviene l'eccezione, prima di spostarsi sull'esecuzione del gestore.

MIPS non salva lo stato di nessun altro registro, per cui il corretto ritorno all'esecuzione non è garantito di default: deve essere il gestore a gestire la consistenza di questi dati.

Gestione a Livello Hardware

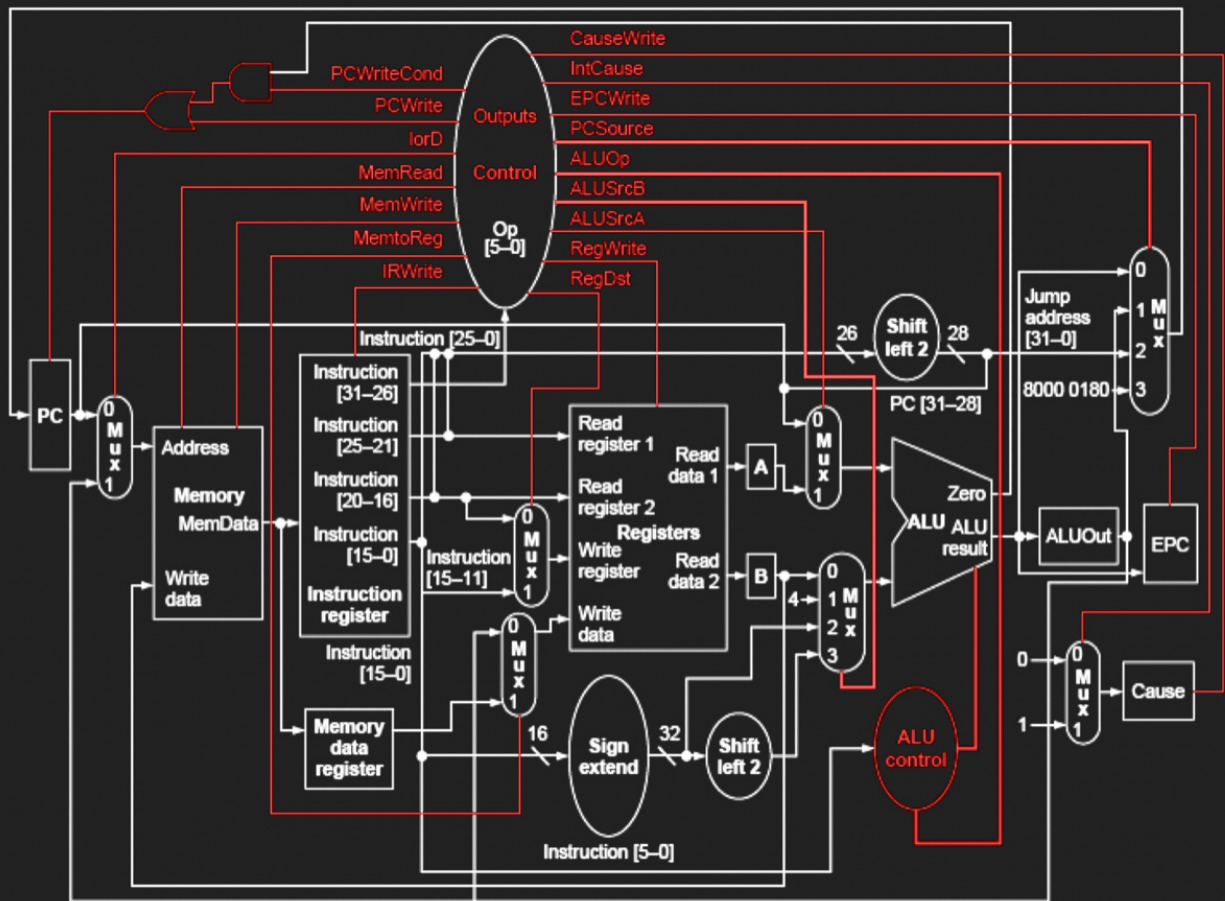
Consideriamo due tipi di eccezioni:

- ▼ **Istruzione non valida**
Necessita di un controllo sull'OP Code dell'istruzione
- ▼ **Overflow**
Necessita modifiche hardware aggiun

Avvenuta l'eccezione dobbiamo:

- Individuarne il tipo e aggiornare il registro Cause.
- Interrompere l'esecuzione corrente.
- ▼ **Salvare l'istruzione corrente**
 $EPC = PC - 4$
È necessario salvare l'istruzione precedente a quella salvata nel PC, perchè tale registro è stato incrementato di 4 durante la fase di fetch.
- Lanciare un gestore di eccezioni del SO

A livello **hardware**, per le eccezioni di tipo *overflow*, è necessario cablare un **segnale** che entra nella *rete di controllo principale* per effettuare la gestione dell'eccezione.



Struttura Software di Gestione delle Eccezioni

[Appendice A.7](#)

- ▼ Guida ai valori del registro **Cause**

0	Int	Interrupt (hardware)
4	AdEL	address error (load or fetch)
5	AdES	address error (store)
6	IBE	bus error on instruction fetch
7	DBE	bus error in data load or store
8	Sys	syscall exception
9	Bp	breakpoint exception
10	R1	reserved instruction exception
11	CpU	coprocessor unimplemented
12	0v	arithmetic overflow exception
13	Tr	trap
15	FPE	floating point exception

Per gestire gli interrupt i fa uso del registro dedicato **Status**.

I due registri hanno una serie di bit [8 – 15] denominati "**Interrupt Mask**" legati da una *corrispondenza uno a uno*. Tali bit indicano i **dispositivi** all'origine dell'*interrupt*:

- *Cause* indica quale dispositivo **ha generato** l'interrupt.
- *Status* indica quali dispositivi **possono generare** l'interrupt.

Tramite *Status* è dunque possibile *regolare* quali dispositivi possono mandare i *segnali* di *interrupt*.