

Eccezioni

Datapath multi-cycle

Claudia Raibulet
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 utilizzata 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:**

- 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

- **Interruzioni:**

- 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 è 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

Interruzioni vettorizzate

Exception type	Exception vector address (in hex)
Undefined instruction	C000 0000 _{hex}
Arithmetic overflow	C000 0020 _{hex}

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 2 cause di eccezione come esempi:
 - **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

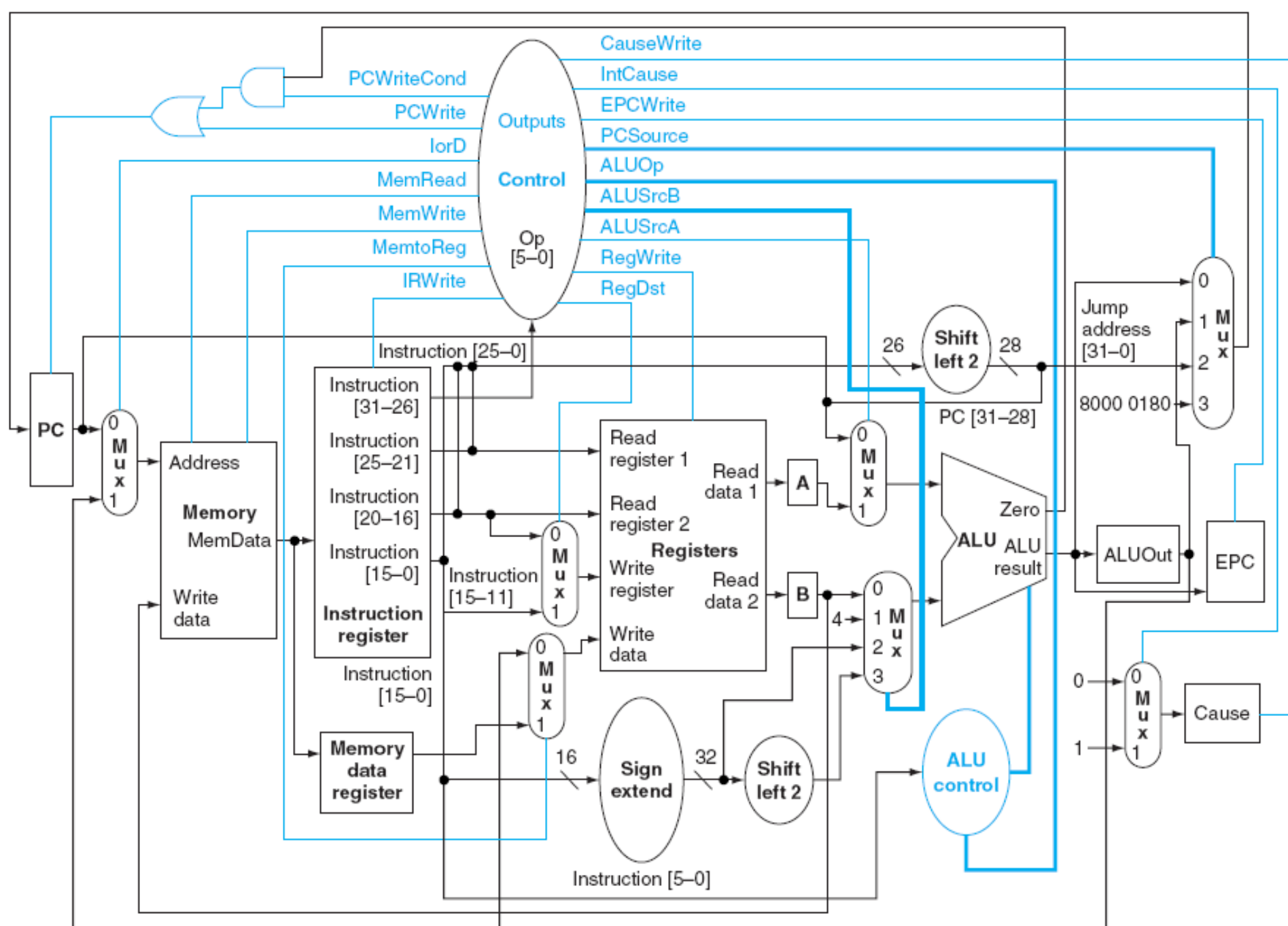
- **Passi da eseguire:**

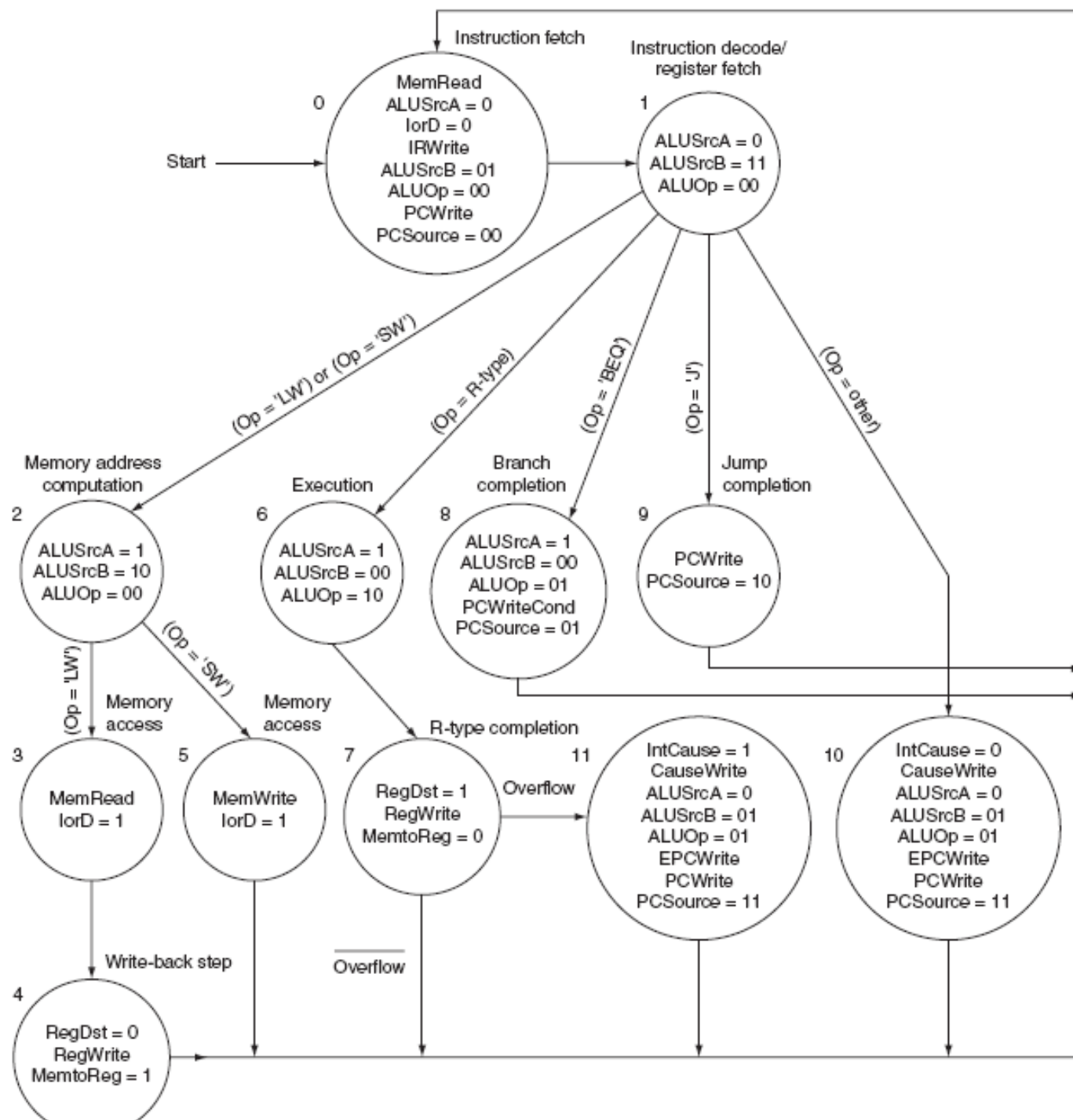
- Individuare l'evento inatteso, la causa dell'eccezione e salvarla in un registro dedicato denominato Cause
- Interrompere l'esecuzione corrente
- Salvare l'indirizzo dell'istruzione corrente nel EPC ($EPC \leftarrow PC-4$)
- Saltare a un gestore delle eccezione del SO che si trova a un indirizzo fisso 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 aritmetico:**
 - 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 perche' consideriamo 2 esempi di eccezioni)
 - Salvare in PC l'indirizzo del gestore delle eccezioni





Schema del processore MIPS

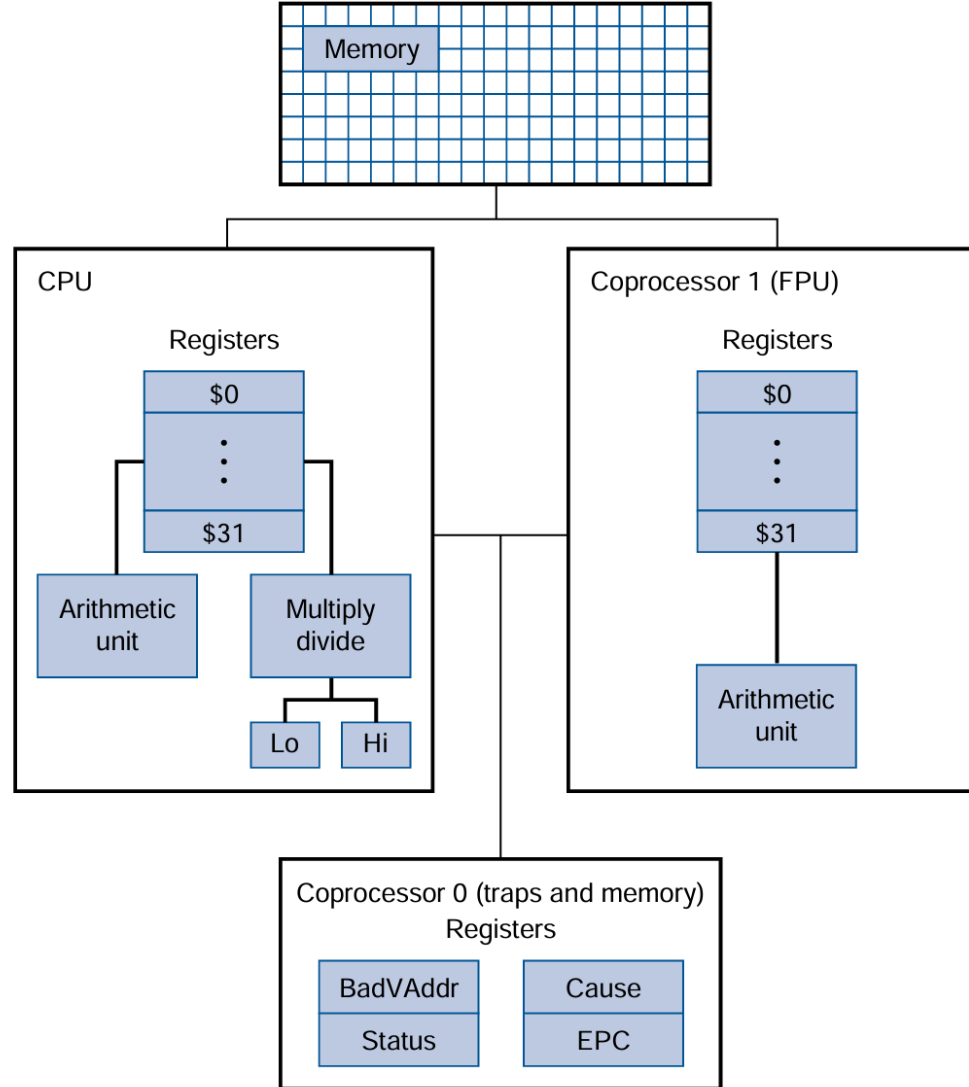


FIGURE A.10.1 MIPS R2000 CPU and FPU.

Coprocessore 0 - Registri

Registro	Numero	Significato
BadVAddr	8	Indirizzo di memoria il cui riferimento ha provocato l'eccezione
Status	12	Maschera delle interruzioni e bit di abilitazione
Cause	13	Tipo di eccezione e interrupt in attesa di servizio
EPC	14	Indirizzo dell'istruzione che ha causato l'eccezione

➤ Si usano le istruzioni

- mtc0 – move to c0
- mfc0 – move from c0

per accedere ai valori di tali registri

Registro Cause

- 8 bit per interrupt pendenti
- 5 bit per il codice dell'eccezione

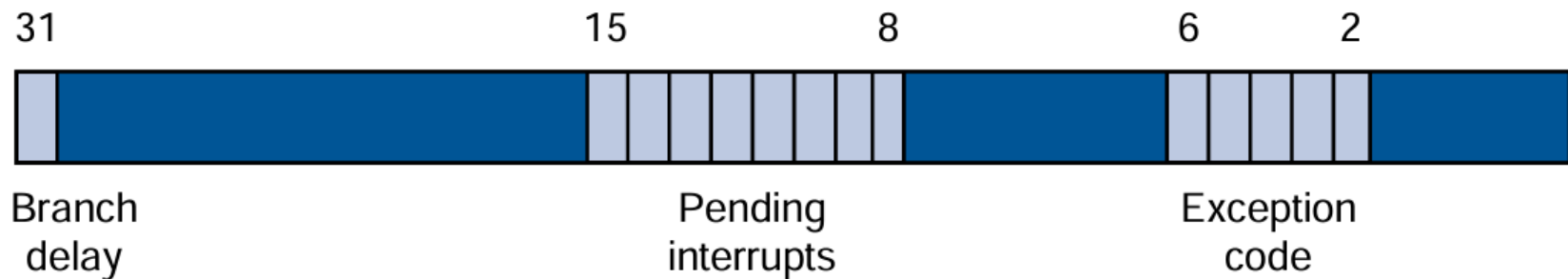


FIGURE A.7.2 The Cause register.

Tabella delle eccezioni

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

Registro Status

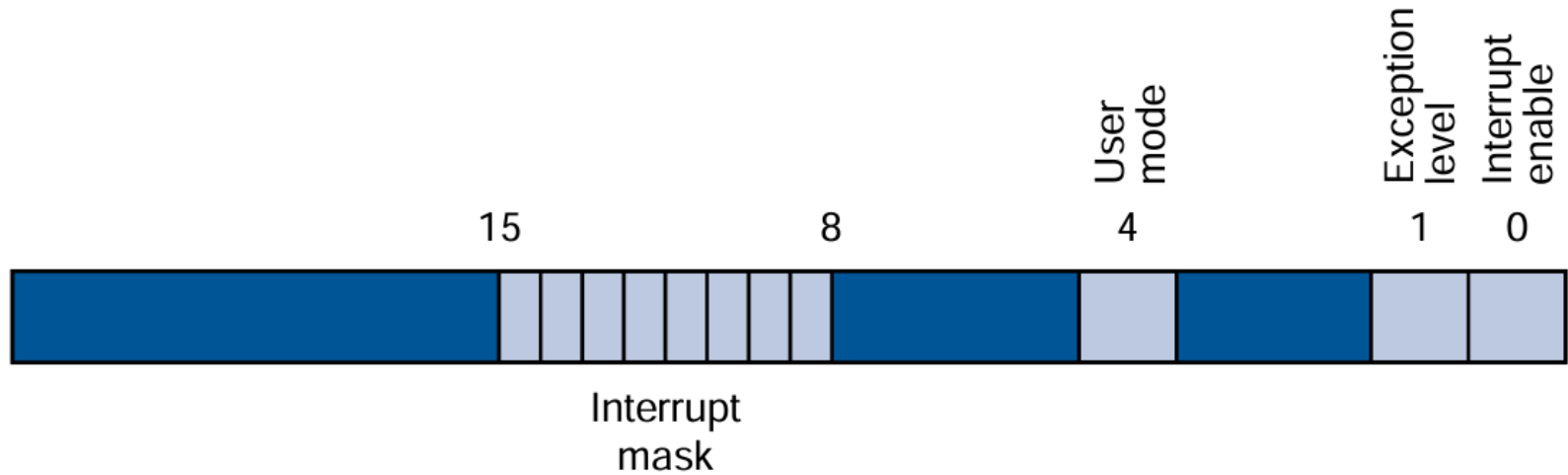


FIGURE A.7.1 The Status register.

Esempio – overflow aritmetico

Consideriamo che il programma è memorizzato partendo dall'indirizzo 0x00400028.

.text

main:

Genera un overflow aritmetico addizionando "con segno"
due numeri positivi la cui somma NON e' rappresentabile
in complemento a 2

li \$t0, 0x7fffffff # massimo positivo in complemento a 2
li \$t1, 2 # aggiungo abbastanza per superare il massimo
positivo
add \$a0, \$t0, \$t1 # cosa succede dopo aver eseguito questa
istruzione?

li \$v0, 1 # stampa il risultato dell'operazione aritmetica
syscall

jr \$ra # fine

Esempio – Overflow Aritmetico

- Il contenuto dei registri quanto si solleva l'eccezione:

PC = 80000180 EPC = 00400030 Cause = 00000030 BadVAddr= 00000000
Status = 3000ff12 HI = 00000000 LO = 00000000

➤ Il messaggio scritto nel Message Window:

[0x0040002c] 0x34090002 ori \$9, \$0, 2 ; 13: li \$t1, 2 # aggiungo
abbastanza per superare il massimo positivo
[0x00400030] 0x01092020 add \$4, \$8, \$9 ; 14: add \$a0, \$t0, \$t1 # cosa
succede dopo aver eseguito questa istruzione ?

Exception occurred at PC=0x00400030
Arithmetic overflow

- **Si noti che quando avviene l'eccezione:**
 - appare un messaggio sulla Message Window "Exception occurred ..."; tale stampa è effettuata internamente dal simulatore e NON dal gestore delle eccezioni che non ha ancora iniziato l'esecuzione
 - PC assume il valore 0x80000180, l'indirizzo iniziale del gestore delle eccezioni (Exceptions.s)
 - EPC assume il valore 0x00400030, l'indirizzo della istruzione che ha provocato l'eccezione
 - Il registro Cause assume il valore 0x00000030

Registri EPC e BadVAddr

- **EPC**
 - Dopo che è avvenuta una eccezione il registro EPC contiene l'indirizzo dell'istruzione in corso di esecuzione (PC) quando è avvenuta l'eccezione
- **BadVAddr**
 - Se avviene una eccezione a seguito di un accesso illegale alla memoria, questo registro contiene l'indirizzo della locazione alla quale il programma ha tentato di accedere

Esercizio

```
.data
dato:
    .word 42
    .text
main:
    la    $t0, dato           # indirizzo in data segment
    lw    $t1, 0($t0)         # lettura da locazione
    lw    $t2, 3($t0)         # lettura da locazione nop
    jr    $ra                 # fine
```

➤ **Considerando che il programma è memorizzato partendo dall'indirizzo 0x00400024 e i dati dall'indirizzo 0x10010000 si chiede:**

- Quale istruzione genera l'eccezione?
- Cosa indica il registro Cause?
- Quali valori assumono i registri EPC e BadVAddr?

Da leggere

- **Chapter 5: The processor: Datapath and Control – disponibile sul sito elearning del corso – pag. 340 -346**
- **Appendice A: Capitolo A7**