Exceptions

Exception e Interrupt

- un evento generato che distrugge la normale sequenza di istruzioni
- Usato normalmente per richiamare l'attenzione della CPU su eventi I/O asincroni (termine di una fase di scrittura DMA) senza usare un meccanismo di polling periodico e nel paging della memoria.
- Nelle CPU pipelined e superscalari interrompe il parallelismo di esecuzione delle istruzioni: un cattivo uso delle eccezioni/interrupt può provocare cadute di performance anche notevoli.

Exception vs Interrupt

- MIPS (distinzione tra interno ed esterno)
  - **Exception**: un evento generato all'**interno** della CPU che distrugge la normale sequenza di istruzioni
  - **Interrupt**: un evento generato all'**esterno** della CPU che distrugge la normale sequenza di istruzioni
- Intel
  - **Interrupt**: un evento che distrugge la normale sequenza di istruzioni (interno o esterno)

Exception vs Interrupt

Type of event	From where?	MIPS terminology
I/O device request	External	Interrupt
Invoke the operating system from user program	Internal	Exception
Arithmetic overflow	Internal	Exception
Using an undefined instruction	Internal	Exception
Hardware malfunctions	Either	Exception or interrupt

Coprocessore 0

- Gestisce le eccezioni e la memoria virtuale
- Sempre presente (mentre il coprocessore 1 dedicato al Floating Point può anche non essere presente)

Coprocessore 0

Exception handling registers in coprocessor 0

Register Number	Register Name	Usage
8	BadVAddr	Memory address where exception occurred
12	Status	Interrupt mask, enable bits, and status when exception occurred
13	Cause	Type of exception and pending interrupt bits
14	EPC	Address of instruction that caused exception

Instructions which access the registers of coprocessor 0

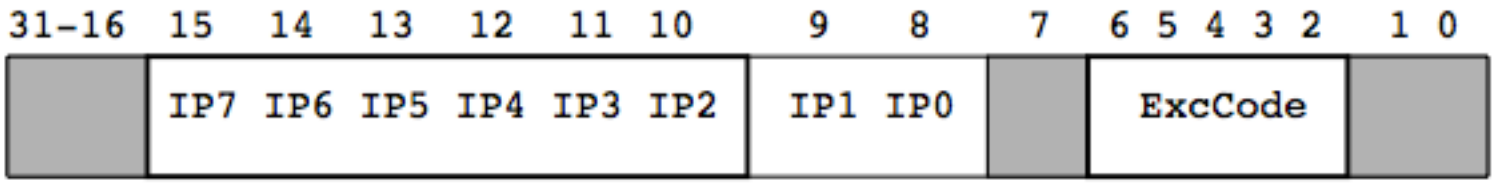
Instruction	Comment
`mfc0 Rdest, C0src`	Move the content of coprocessor's register `C0src` to `Rdest`
`mtc0 Rsrc, C0dest`	Integer register `Rsrc` is moved to coprocessor's register `C0dest`
`lwc0 C0dest, address`	Load word from `address` in register `C0dest`
`swc0 C0src, address`	Store the content of register `C0src` at `address` in memory

The BadVAddr register

- (**Bad V**irtual **Addr**ess) contiene l'indirizzo di memoria usato dall'istruzione che ha generato l'eccezione. Un accesso non allineato alla memoria genererà un eccezione e l'indirizzo non allineato sarà salvato in questo registro.

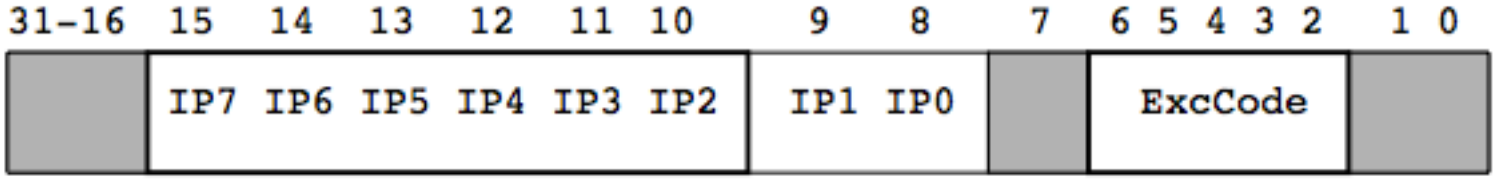
The Cause register

- Fornisce informationi su quali interrupts sono pendenti (IP0 to IP7) e sul tipo di exception che si è verificata. Il codice dell'exception è memorizzato nei bits 6-2



The Cause register

- IP0-IP1 sono usati per generare interrupt software, IP2-IP7 sono interrupt generati in hardware
- Il bit IP*i* è a 1 quando si è generato un interrupt di livello *i* e non è stato ancora servito.



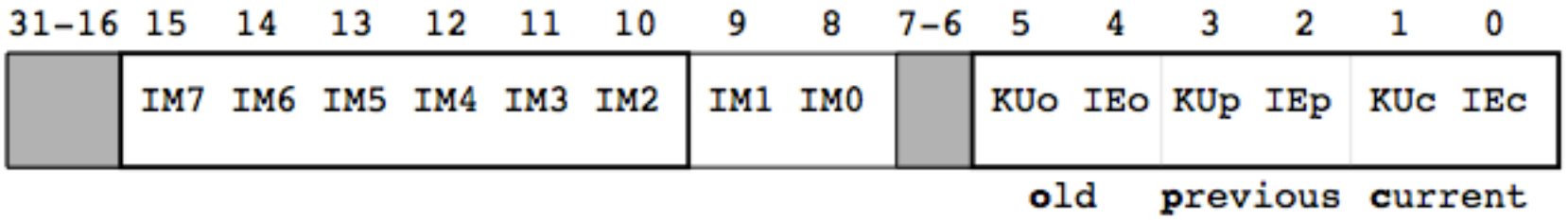
The Cause register

Exception codes[a] implemented by SPIM

Code	Name	Description
0	INT	Interrupt
4	ADDRL	Load from an illegal address
5	ADDRS	Store to an illegal address
6	IBUS	Bus error on instruction fetch
7	DBUS	Bus error on data reference
8	SYSCALL	`syscall` instruction executed
9	BKPT	`break` instruction executed
10	RI	Reserved instruction
12	OVF	Arithmetic overflow

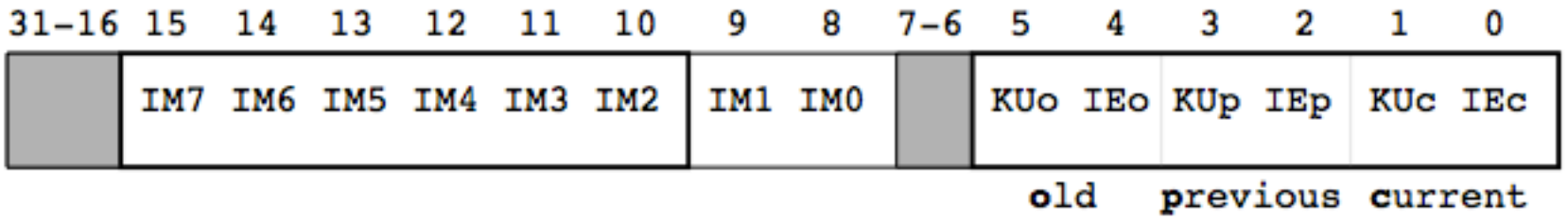
The Status register

- Contiene un interrupt mask nei bits 15-10 e uno insieme di bits 5-0 per controllare lo stato.
- Se il bit IM*i* è settato a 1 allora l'interrupt di livello *I* genera un'interruzione per la CPU



The Status register

- Se IE*c* è settato a 1 allora le Interruzioni sono abilitate.
- Se KUc è settato a 1 allora la CPU è in *user mode*, altrimenti è in *kernel mode.*



Kernel vs User mode

- La CPU entra in kernel mode quando si verifica un’excepion o un imterrupt.
- Il Coprocessor 0 può essere usato solo in kernel mode.
- La memoria superiore (da 0x8000.0000 in poi) può essere usata solo in kernel mode.

The Status register

- I bit 0-5 implementano un semplice sistema a tre livelli per ricordare lo stato precendente all'exception.
- Quando si verifica un exception:

  KUo,IEo ← KUp,IEp ← KUc,IEc ← 0,0

- Quando si ritorna da un'eccezione con l'istruzione **rfe**

  KUo,IEo → KUp,IEp → KUc,IEc

The EPC register

- **Exception** (interna): Viene servita immediatamente interrompendo l'istruzione che era in esecuzione al momento dell'exception. EPC memorizza l'indirizzo dell'istruzione che era in esecuzione quando si è verificata l'exception.
- **Interrupt** (esterna): viene servito alla fine dell'istruzione in esecuzione al momento della chiamata. EPC memorizza l'indirizzo dell'istruzione seguente quella in esecuzione al momento dell'interrupt.

References

- Capitolo 5.6 di Computer Organization and design Patterson-Hennessy
- http://www.cs.iit.edu/~virgil/cs470/Labs/Lab7.pdf