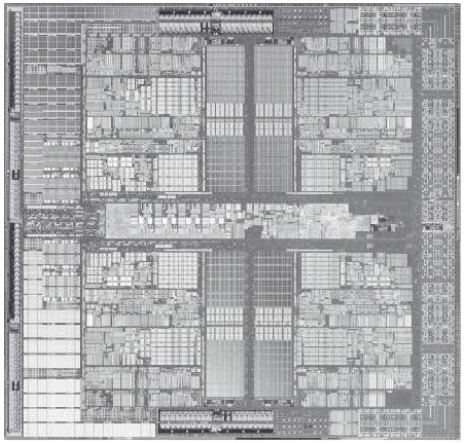
Università degli Studi di Cassino



Corso di Calcolatori Elettronici

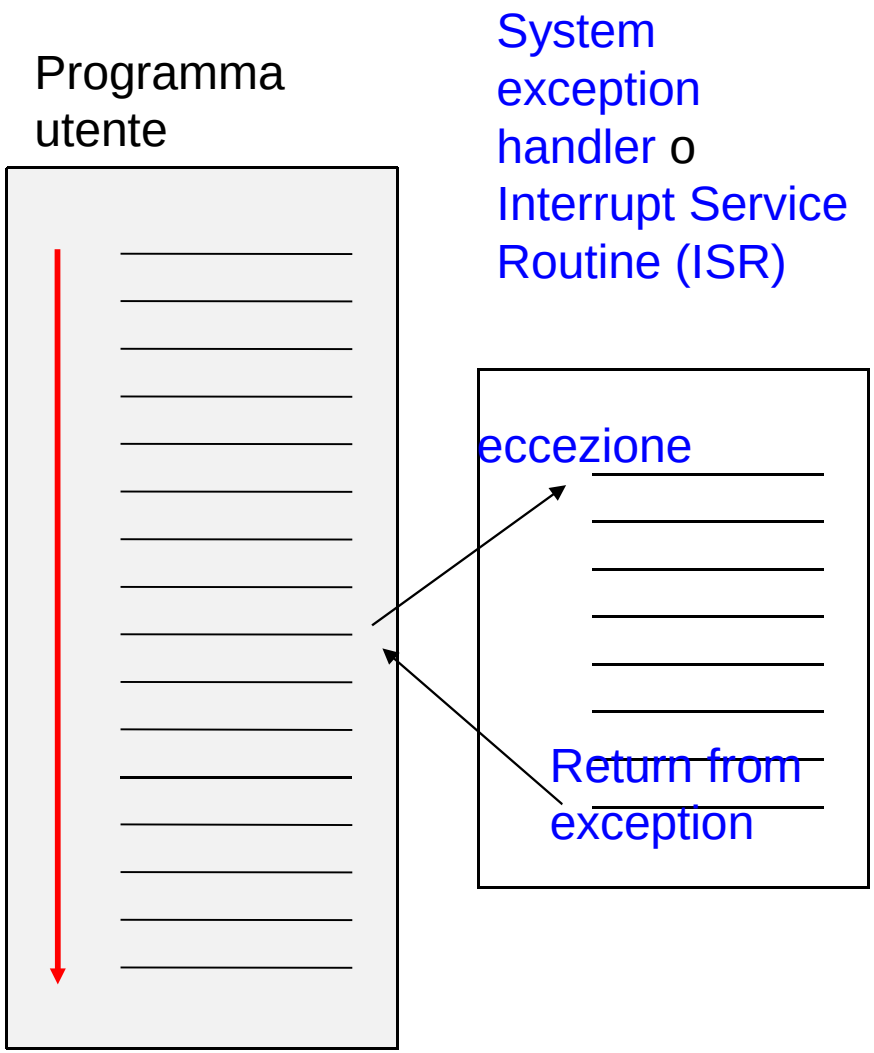
Eccezioni ed Interruzioni

Anno Accademico 2010/2011

Francesco Tortorella

Eccezioni

- Trasferimento imprevisto del controllo sulla base del verificarsi di un evento.
- Il sistema risponde con un'azione opportuna all'eccezione.
  - Necessario salvare l'indirizzo dell'istruzione in corrispondenza della quale si è generata l'eccezione
- Al termine viene restituito il controllo al programma interrotto
- Necessario salvare e poi ripristinare lo stato del programma interrotto.



Flusso di controllo normale: sequenza, branch, jump

Tipi di eccezioni

- **Interruzioni**
  - Causate da eventi esterni
  - Asincrone rispetto al programma in esecuzione
  - Possono essere gestite tra due istruzioni consecutive
  - Semplice sospensione e ripresa del programma utente

- **Eccezioni**
  - Causate da eventi interni
    - Condizioni di eccezione (overflow)
    - Errori (parity)
    - Faults (pagine non residenti in memoria)
  - Sincrone rispetto al programma in esecuzione
  - La condizione di eccezione deve essere risolta dall'handler
  - Se la condizione è risolubile, l'istruzione può essere rieseguita ed il programma continuato. In caso contrario, il programma deve essere terminato prematuramente.

Gestione delle eccezioni

- Due esigenze vanno considerate per una corretta gestione delle eccezioni:
  - Salvare lo stato del programma interrotto
    - PC, altre informazioni
  - Individuare la routine da eseguire in funzione della causa che ha generato l'eccezione.
- Salvataggio dello stato
  - Push sullo stack (Vax, Motorola 68K, 80x86)
  - Salvataggio in registri speciali (MIPS)
  - Registri ombra (Motorola 88K)

Gestione delle eccezioni: indirizzamento dell'ISR

Diverse soluzioni:

- Interrupt vettorizzato (68K, 80x86,…)

Viene fornito un valore (vector) che è un indice in una tabella (vector table) che contiene gli indirizzi delle varie ISR:
PC=Mem[IVT_base+vector||00]. Causa→vector

- Tabella delle ISR (Sparc, 88K,…)

Viene fornito uno spiazzamento all'interno di un'area di memoria che contiene tutte le ISR:
PC=IT_base+offset||0000. Causa→offset

- Indirizzo fisso (MIPS)

Si salta ad un unico indirizzo: unica ISR che si occupa di identificare la causa e avviare la routine opportuna (effettivamente nel MIPS si considerano due entry).
PC=Exc_address

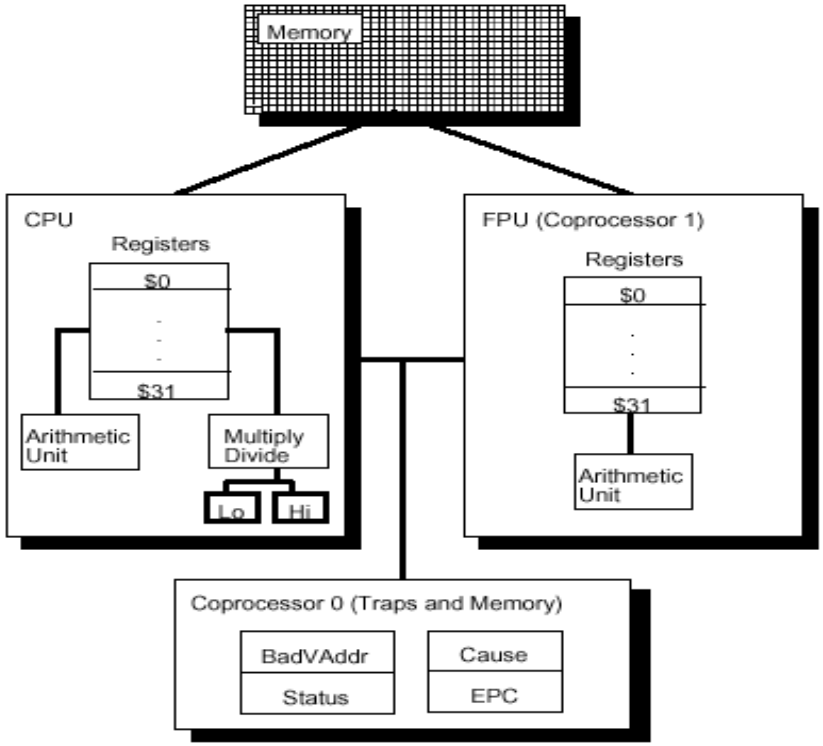
Modi di esecuzione user / kernel

- Due modi di esecuzione:
  - User
  - kernel
- Per ognuno dei due modi di esecuzione sono previste apposite aree dati e codice

- In modo kernel vengono eseguite le istruzioni del sistema operativo ed, in particolare, vengono gestite le eccezioni.

Gestione delle Eccezioni nel MIPS

- La gestione delle eccezioni e delle interruzioni è effettuata dal coprocessore 0
- 4 registri mantengono tutte le informazioni necessarie alla gestione
- E' possibile accedere ai registri tramite le istruzioni:
  lwc0, swc0, mfc0, mtc0



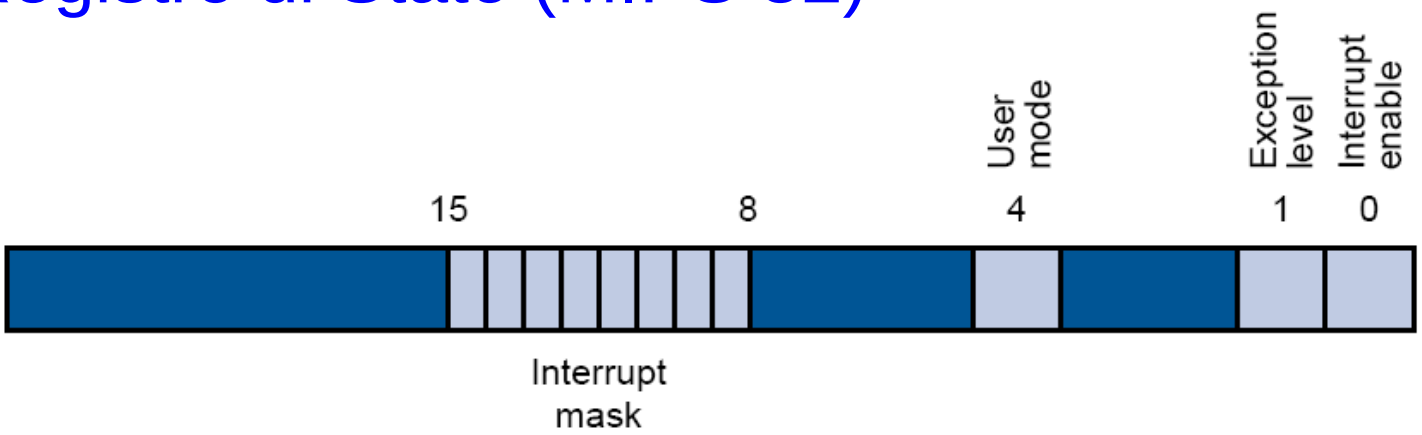
Registri del coprocessore 0

Register name	Register number	Usage
BadVAddr	8	memory address at which an offending memory reference occurred
Count	9	timer
Compare	11	value compared against timer that causes interrupt when they match
Status	12	interrupt mask and enable bits
Cause	13	exception type and pending interrupt bits
EPC	14	address of instruction that caused exception
Config	16	configuration of machine

EPC contiene l'indirizzo di ritorno dal servizio all'interruzione

BadVAddr contiene l'indirizzo di memoria cui è stato fatto riferimento

Registro di Stato (MIPS 32)



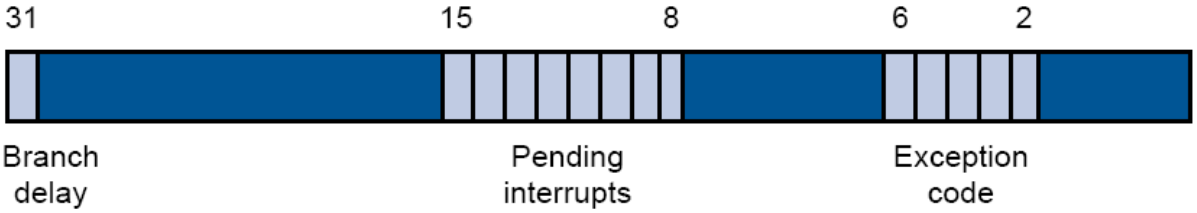
- L'interrupt mask contiene un bit per ciascuno degli 8 livelli possibili di interruzione: 6 hardware (10-15) e 2 software (8-9). Il livello è abilitato se il bit corrispondente è messo a 1.

- Il bit 0 (IEC) gestisce l'abilitazione generale delle interruzioni (0=disabilitate).

- Il bit 1 segnala se si sta servendo un'eccezione.

- Il bit 4 segnala se il processore è in modo kernel (0) o utente (1)

Registro Causa (MIPS 32)



I bit 8-15 corrispondono ai 6 livelli di interrupt hardware e 2 software: se un bit è a 1, c'è un'interruzione non ancora servita a quel livello.

I bit 2-6 descrivono la causa dell'eccezione secondo la codifica:

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

Interruzioni ed eccezioni

- Esistono diverse categorie di “interruzioni”:
  - Eccezioni
  - Interruzioni
  - Chiamate di sistema
  - Segnale RESET

Eccezioni

- Corrispondono ad eventi “anormali”, tipicamente errori che impediscono la corretta esecuzione dell’istruzione in corso.
- Non sono mascherabili.
- Tipi di eccezioni:
  - ADDRL Address error in lettura (load o istr. Fetch)
  - ADDRS Address error in scrittura
  - IBUS Bus error in istr. fetch
  - DBUS Bus error in data
  - RI Reserved instruction
  - CPU Coprocessor inaccessibile
  - OVF Overflow

Gestione delle eccezioni

- All'atto di un'eccezione il processore:
  - Salva l'indirizzo dell'istruzione (PC-4) in EPC
  - Salva la configurazione presente del registro di stato
  - Passa in modo kernel e disabilita le interruzioni
  - Salta all'indirizzo 0x80000180

Interruzioni

- Le richieste di interruzione sono eventi asincroni provenienti in genere dalle periferiche.
- Il processore ha 6 linee di interruzione esterne che possono essere mascherate globalmente o singolarmente. L'attivazione di una di queste linee è una richiesta di interruzione.
- Le interruzioni sono segnalate nel registro Causa e sono gestite al termine dell'esecuzione dell'istruzione in corso, se non sono mascherate
- Il processore passa in modo kernel e salta al gestore di interruzioni dopo aver salvato l'indirizzo di ritorno.

Gestione delle interruzioni

- All'atto di un'interruzione il processore:
  - Salva l'indirizzo (PC) in EPC
  - Salva la configurazione presente del registro di stato
  - Passa in modo kernel e disabilita le interruzioni
  - Salta all'indirizzo 0x80000180

Chiamate di sistema

- Due istruzioni per attivare chiamate di sistema, eseguibili in modo utente:
  - SYSCALL
  - BREAK
- L'istruzione SYSCALL permette ad un processo di chiedere un servizio al sistema, p. es. un'operazione di I/O
- Il codice identificativo del tipo di servizio richiesto e gli eventuali parametri devono essere preparati prima all'interno di registri generali
- L'istruzione BREAK è utilizzata più specificamente per inserire un punto di arresto all'interno di un programma
- In entrambi i casi, il processore passa in modo kernel e salta al gestore di interruzioni dopo aver salvato l'indirizzo di ritorno.

Gestione delle chiamate di sistema

- All'atto dell'esecuzione di una chiamata di sistema il processore:
  - Salva l'indirizzo di ritorno (PC) in EPC
  - Salva la configurazione presente del registro di stato
  - Passa in modo kernel e disabilita le interruzioni
  - Salta all'indirizzo 0x80000180

Reset

- Il processore possiede una linea RESET la cui attivazione produce il salto incondizionato alla routine di inizializzazione
- L'attivazione della linea RESET è simile ad una settima linea di interruzione esterna con le seguenti importanti differenze:
  - non è mascherabile
  - non è necessario salvare un indirizzo di ritorno
  - il codice per la gestione del reset si trova all'indirizzo 0xBFC00000
- All'atto dell'attivazione del RESET il processore
  - Passa in modo kernel e disabilita le interruzioni
  - Salta all'indirizzo 0xBFC00000

Istruzioni per la gestione di eccezioni

- **eret** (exception return)
  - Salta all'indirizzo contenuto in EPC
- **break** *code* (es. break 3)
  - Genera una eccezione di breakpoint parametrizzata da *code*. *code*=1 è riservato al debugger. Il code si estrae dall'istruzione indirizzata da EPC
- **syscall**
  - Genera una system call. Il codice del servizio è contenuto in $v0 ($2)
- **nop** (no operation)
  - Istruzione senza alcun effetto

Fasi del servizio delle eccezioni

- In seguito al verificarsi di un'eccezione
  - L'istruzione in esecuzione, e tutte quelle successive in esecuzione nella pipeline, sono abortite
  - Le informazioni utili alla gestione e al ripristino dell'esecuzione del processo interrotto sono salvate nei registri del coprocessore 0
  - La modalità di esecuzione del processore commuta a kernel
  - Il controllo è trasferito a un ***exception handler*** posto all'indirizzo **0x80000180** (kernel), che esamina la causa dell'eccezione e la gestisce

Exception handler

```
.ktext 0x80000180
mov  $k1, $at      # Save $at register
sw   $a0, save0    # Handler is not re-entrant and can't use
sw   $a1, save1    # stack to save $a0, $a1
                   # Don't need to save $k0/$k1

mfc0    $k0, $13        # Move Cause into $k0

srl     $a0, $k0, 2     # Extract ExcCode field
andi    $a0, $a0, 0xf

beqz    $a0, done       # Branch if ExcCode is Int (0)

mov     $a0, $k0        # Move Cause into $a0
mfc0    $a1, $14        # Move EPC into $a1
jal     print_excp      # Print exception error message
```

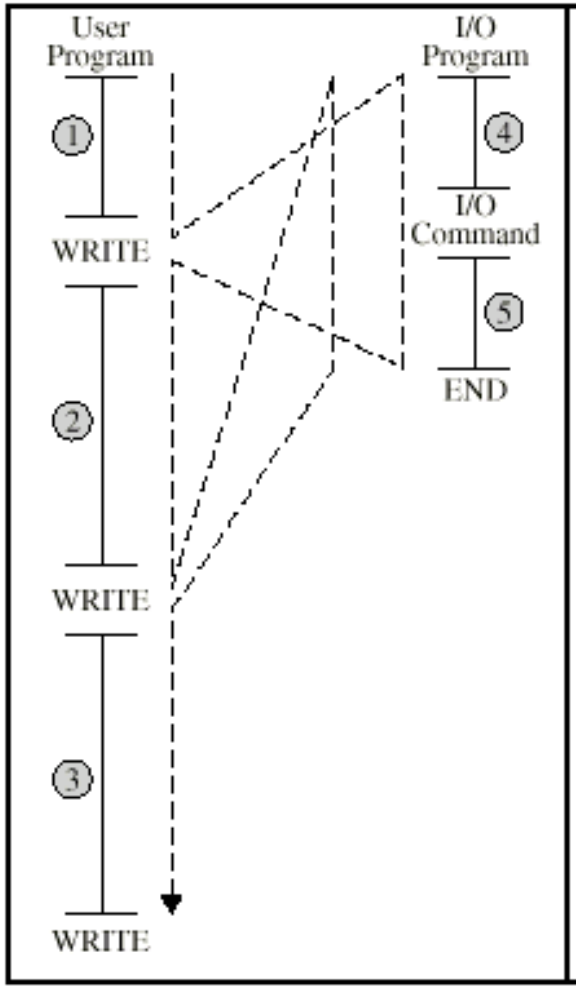
Exception handler

```
done:   mfc0    $k0, $14        # Bump EPC
        addiu   $k0, $k0, 4     # Do not reexecute
                                # faulting instruction
        mtc0    $k0, $14        # EPC

        mtc0    $0, $13         # Clear Cause register

        mfc0    $k0, $12        # Fix Status register
        andi    $k0, 0xfffd     # Clear EXL bit
        ori     $k0, 0x1        # Enable interrupts
        mtc0    $k0, $12
        lw      $a0, save0      # Restore registers
        lw      $a1, save1
        mov     $at, $k1

        eret                    # Return to EPC

        .kdata
save0:  .word 0
save1:  .word 0
```

Interruzioni come strumento per la gestione di operazioni di sistema

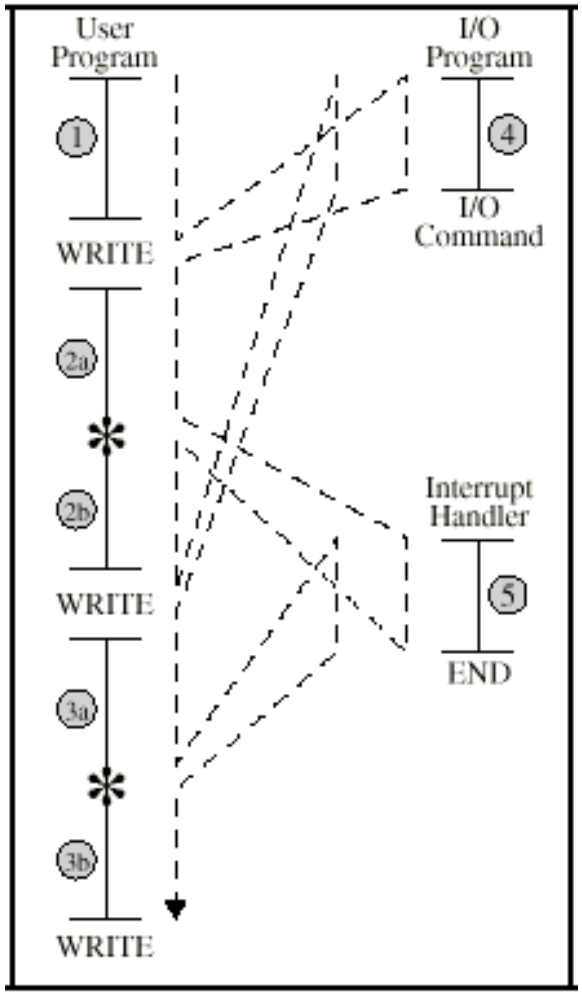
- Le interruzioni possono essere usate per migliorare l'efficienza dell'elaborazione.
- Permettono di liberare il processore da compiti gravosi di sincronizzazione.
- Sono utili soprattutto per gestire le operazioni realizzate da componenti che hanno tempi di risposta molto superiori a quelli del processore (es. dispositivi di I/O).

Flusso di controllo



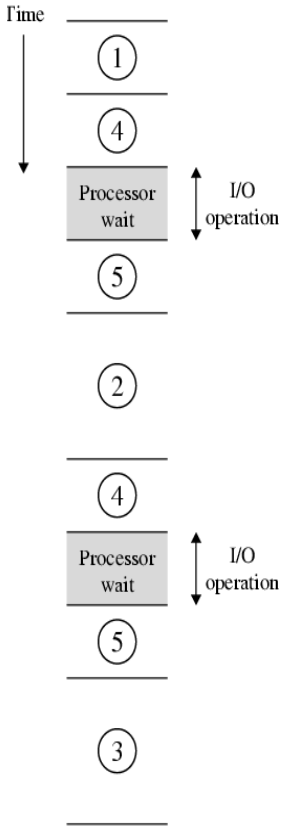
- Gestione di un'operazione di I/O senza interruzioni
- Il processore
  - avvia l'operazione
  - attende la fine dell'operazione
  - chiude l'operazione
  - continua l'esecuzione del programma

Flusso di controllo

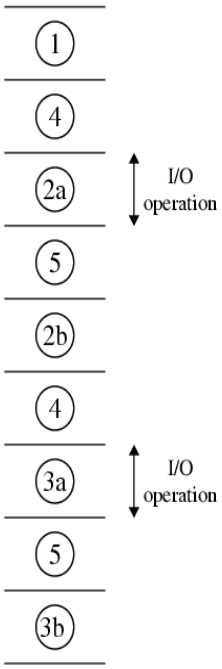


- Gestione di un'operazione di I/O con interruzioni
- Operazione breve (termina prima della richiesta di I/O successiva)
- Il processore
  - avvia l'operazione
  - continua l'esecuzione del programma
  - al termine dell'operazione viene interrotta l'esecuzione del programma e viene gestita la chiusura dell'operazione
  - riprende l'esecuzione

Diagramma temporale

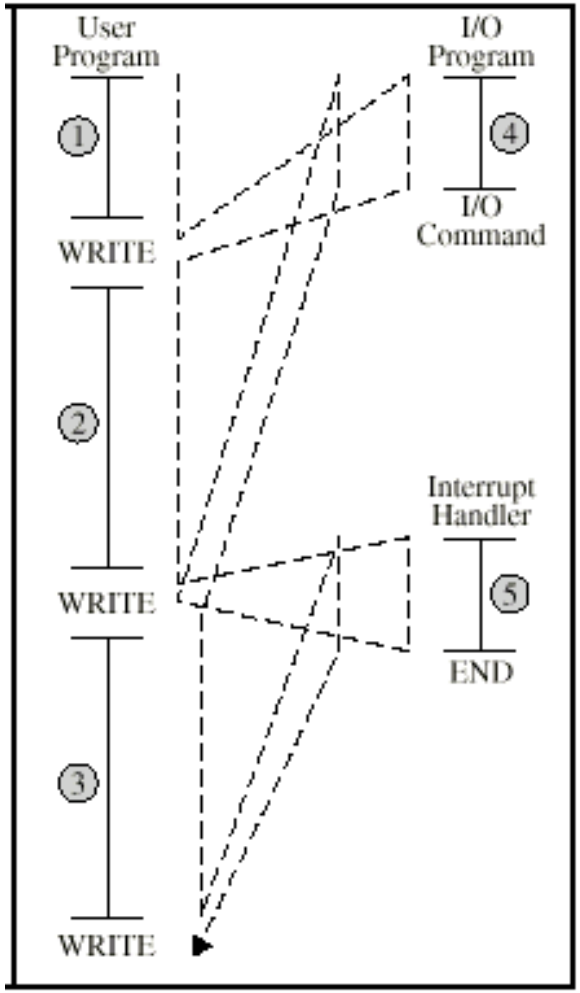


Senza interruzioni



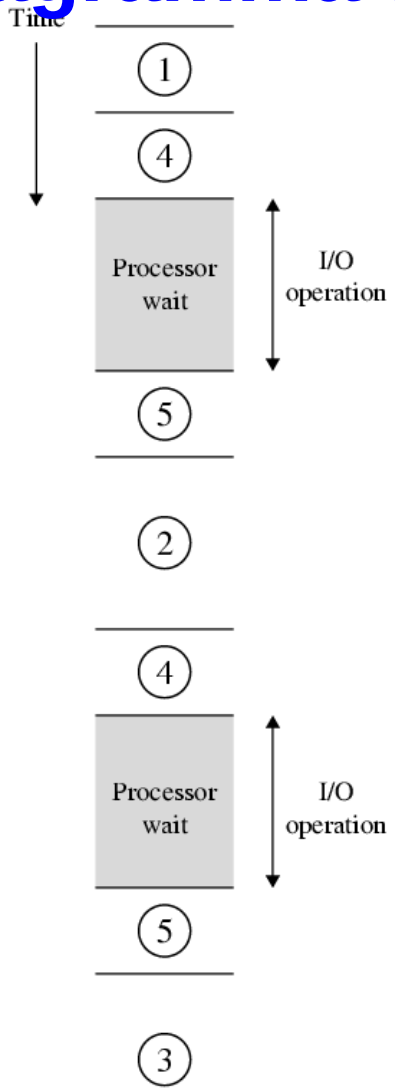
Con interruzioni

Flusso di controllo

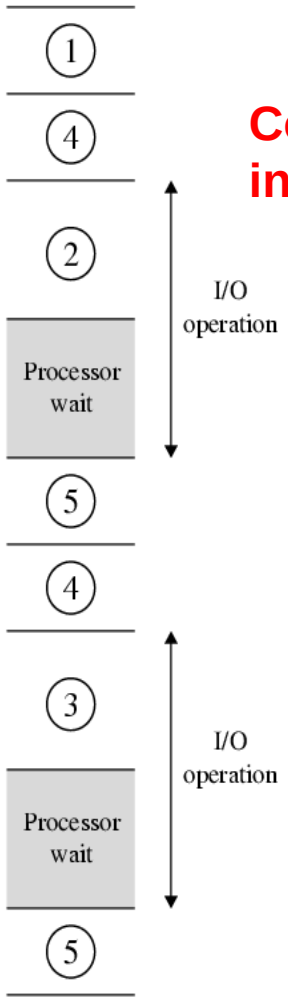


- Gestione di un'operazione di I/O con interruzioni
- Operazione lunga (non termina prima della richiesta di I/O successiva)
- Il processore
  - avvia l'operazione
  - continua l'esecuzione del programma
  - alla richiesta successiva l'esecuzione del programma viene sospesa, si attende il termine dell'operazione precedente e si avvia l'operazione successiva
  - riprende l'esecuzione

Diagramma temporale



Senza interruzioni



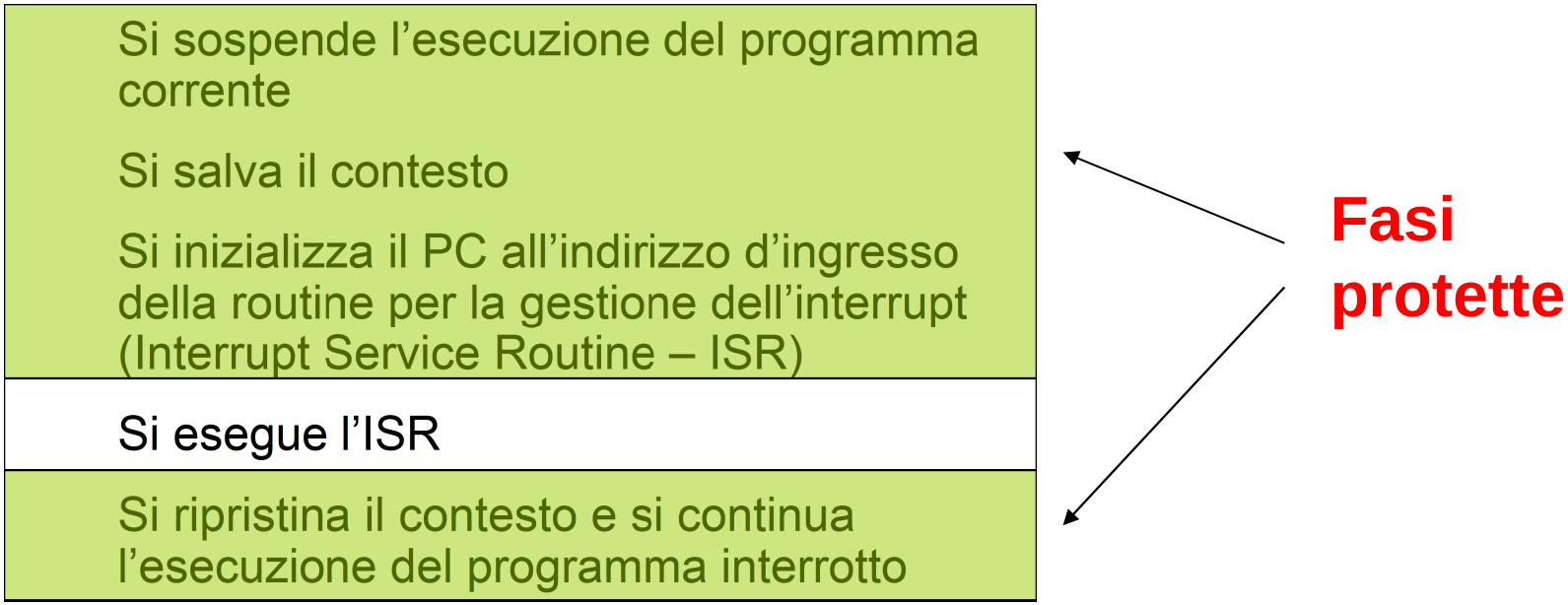
Con interruzioni

Interruzioni multiple

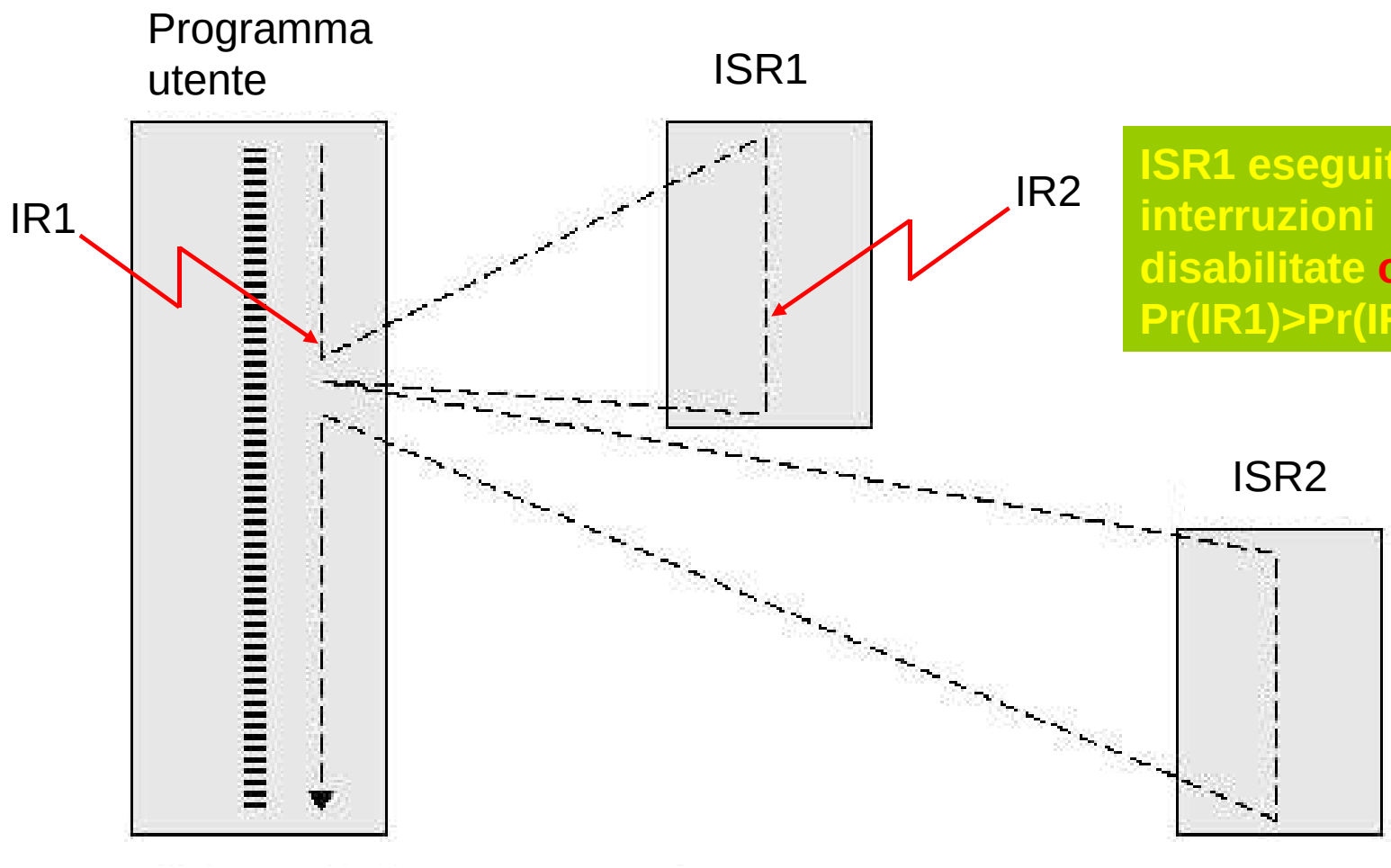
- Una richiesta di interruzione può giungere mentre si sta già servendo un'interruzione. Cosa fare ?
- Disabilitazione delle interruzioni
  - Il processore ignora ulteriori richieste di interruzione mentre sta servendo un'interruzione
  - Le richieste non accolte restano in attesa e sono verificate dopo che la prima interruzione è stata servita
  - Le richieste di interruzione sono gestite in sequenza, in ordine di arrivo
- Definizione di priorità
  - Vengono definite delle classi di priorità (su quale base ?) e ciascuna delle possibili richieste di interruzione viene assegnata ad una classe
  - Le ISR relative ad interruzioni a bassa priorità possono essere interrotte da richieste di interruzione a più alta priorità
  - Il processore torna ad eseguire la ISR interrotta dopo aver servito l'interruzione a più alta priorità
  - Una richiesta di interruzione resta in attesa se il processore sta servendo un'interruzione a priorità più alta

Fasi protette

Anche se subentra una richiesta di interruzione a più alta priorità, ci sono alcune fasi del servizio di un'interruzione (salvataggio e ripristino del contesto) che non possono essere interrotte

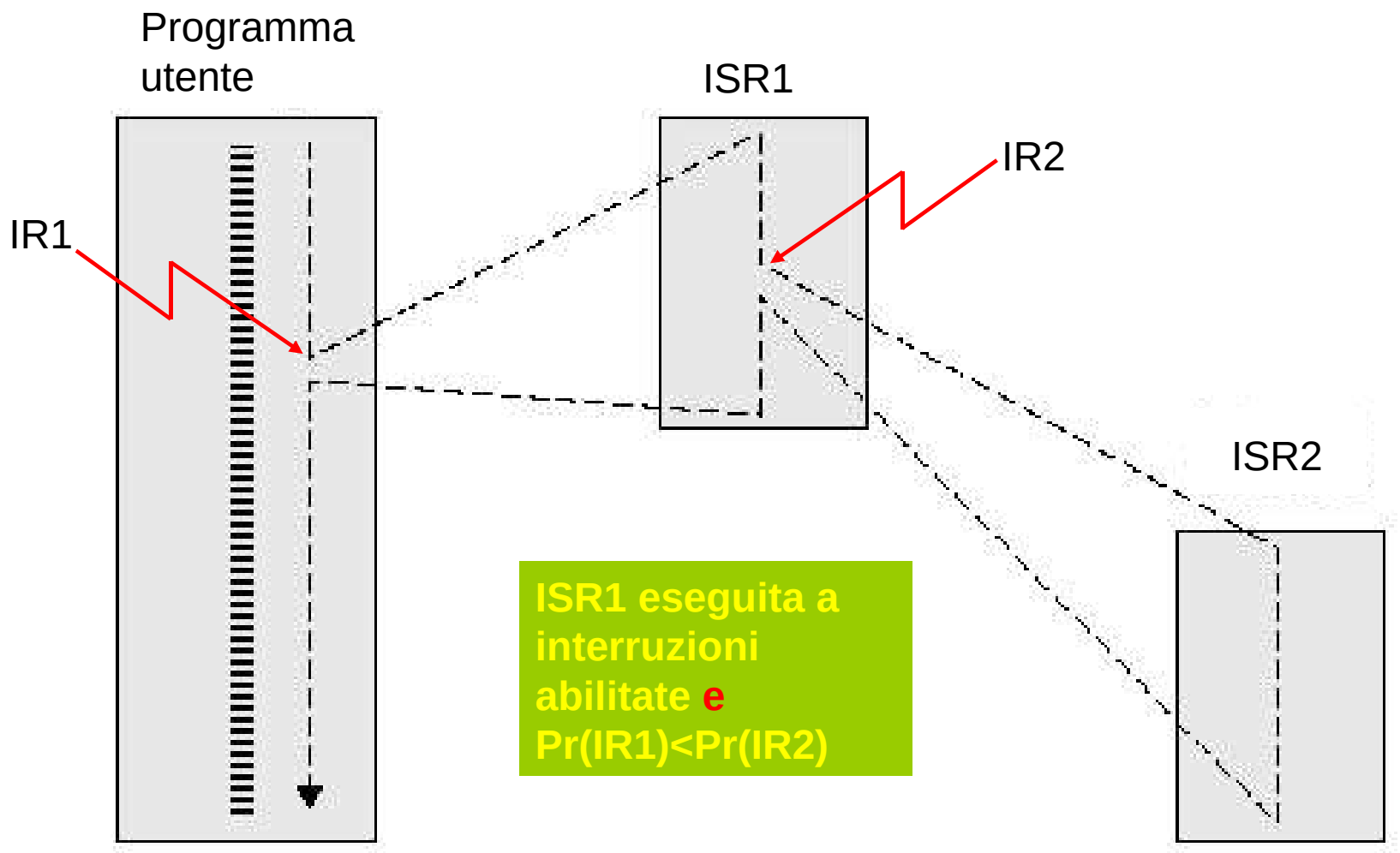


Interruzioni multiple servite in sequenza



ISR1 eseguita a interruzioni disabilitate o Pr(IR1)>Pr(IR2)

Interruzioni multiple innestate



ISR1 eseguita a interruzioni abilitate e Pr(IR1)<Pr(IR2)

Interruzioni innestate – sequenza temporale