

Estructura de Computadores

Tema 8. Excepciones e Interrupciones

Excepciones e Interrupciones

- ▶ Mecanismo del procesador que altera el flujo de control normal de un programa
 - ▶ Sin que intervenga ninguna instrucción de salto

Excepciones e Interrupciones

- ▶ Mecanismo del procesador que altera el flujo de control normal de un programa
 - ▶ Sin que intervenga ninguna instrucción de salto
- ▶ Excepciones
 - ▶ Eventos **internos** causados por la ejecución de una instrucción
 - ▶ Fallo de página, acceso a memoria no alineado, overflow...

Excepciones e Interrupciones

- ▶ Mecanismo del procesador que altera el flujo de control normal de un programa
 - ▶ Sin que intervenga ninguna instrucción de salto
- ▶ Excepciones
 - ▶ Eventos **internos** causados por la ejecución de una instrucción
 - ▶ Fallo de página, acceso a memoria no alineado, overflow...
- ▶ Interrupciones
 - ▶ Eventos **externos** al procesador producidos por dispositivos de entrada/salida

Rutina de Servicio de Excepciones (RSE)

- ▶ RSE (exception handler)
 - ▶ Código del sistema operativo que se encarga de gestionar las excepciones e interrupciones

Rutina de Servicio de Excepciones (RSE)

- ▶ RSE (exception handler)
 - ▶ Código del sistema operativo que se encarga de gestionar las excepciones e interrupciones
- ▶ Al detectar una excepción o interrupción:
 1. Se finaliza o se cancela la instrucción en curso
 2. Se salta a la RSE
 - ▶ Se escribe en el PC la dirección inicial de la RSE
 3. Se ejecutan las instrucciones de la RSE
 4. Se aborta o se reanuda la ejecución del programa

Tipos de excepciones e interrupciones

- ▶ Excepciones producidas al ejecutar una instrucción que viola alguna restricción
 - ▶ Ejemplos:
 - ▶ Overflow aritmético
 - ▶ Acceso a memoria no alineado
 - ▶ División por cero en coma flotante
 - ▶ Instrucción reservada

Tipos de excepciones e interrupciones

- ▶ Excepciones producidas al ejecutar una instrucción que viola alguna restricción
 - ▶ Ejemplos:
 - ▶ Overflow aritmético
 - ▶ Acceso a memoria no alineado
 - ▶ División por cero en coma flotante
 - ▶ Instrucción reservada
 - ▶ La instrucción se interrumpe sin escribir en registros o memoria
 - ▶ Normalmente la RSE aborta la ejecución del programa
 - ▶ En algunos casos se puede tratar:
 - ▶ La RSE resuelve el problema y escribe el resultado en el registro destino de la instrucción que causó la excepción
 - ▶ La RSE reanuda la ejecución saltando a la instrucción siguiente a la que causó la excepción

Tipos de excepciones e interrupciones

- ▶ Excepciones de fallo de página y fallo de TLB
 - ▶ Se producen al traducir una dirección virtual a física en un acceso a memoria

Tipos de excepciones e interrupciones

- ▶ Excepciones de fallo de página y fallo de TLB
 - ▶ Se producen al traducir una dirección virtual a física en un acceso a memoria
 - ▶ En caso de fallo de TLB:
 - ▶ La RSE copia la entrada de la tabla de páginas en la TLB

Tipos de excepciones e interrupciones

- ▶ Excepciones de fallo de página y fallo de TLB
 - ▶ Se producen al traducir una dirección virtual a física en un acceso a memoria
 - ▶ En caso de fallo de TLB:
 - ▶ La RSE copia la entrada de la tabla de páginas en la TLB
 - ▶ En caso de fallo de página:
 - ▶ La RSE copia la página del disco a un marco de página libre en memoria
 - ▶ También se actualizan la tabla de páginas y el TLB

Tipos de excepciones e interrupciones

- ▶ Excepciones de fallo de página y fallo de TLB
 - ▶ Se producen al traducir una dirección virtual a física en un acceso a memoria
 - ▶ En caso de fallo de TLB:
 - ▶ La RSE copia la entrada de la tabla de páginas en la TLB
 - ▶ En caso de fallo de página:
 - ▶ La RSE copia la página del disco a un marco de página libre en memoria
 - ▶ También se actualizan la tabla de páginas y el TLB
 - ▶ La RSE finaliza saltando a la instrucción que causó la excepción

Tipos de excepciones e interrupciones

- ▶ Excepciones de *trap* y llamada al sistema
 - ▶ Excepción producida por una instrucción especial del ISA para acceder a un servicio del sistema operativo

Tipos de excepciones e interrupciones

- ▶ Excepciones de *trap* y llamada al sistema
 - ▶ Excepción producida por una instrucción especial del ISA para acceder a un servicio del sistema operativo
 - ▶ Similar a una llamada a subrutina
 - ▶ Paso de parámetros y retorno de resultados

Tipos de excepciones e interrupciones

- ▶ Excepciones de *trap* y llamada al sistema
 - ▶ Excepción producida por una instrucción especial del ISA para acceder a un servicio del sistema operativo
 - ▶ Similar a una llamada a subrutina
 - ▶ Paso de parámetros y retorno de resultados
 - ▶ Instrucción `syscall` en MIPS

Tipos de excepciones e interrupciones

- ▶ Excepciones de *trap* y llamada al sistema
 - ▶ Excepción producida por una instrucción especial del ISA para acceder a un servicio del sistema operativo
 - ▶ Similar a una llamada a subrutina
 - ▶ Paso de parámetros y retorno de resultados
 - ▶ Instrucción `syscall` en MIPS
 - ▶ Al finalizar la RSE, se continúa ejecutando el programa a partir de la instrucción siguiente a la que produjo la excepción

Tipos de excepciones e interrupciones

- ▶ Interrupciones
 - ▶ Causadas por dispositivos de E/S (entrada/salida)

Tipos de excepciones e interrupciones

- ▶ Interrupciones
 - ▶ Causadas por dispositivos de E/S (entrada/salida)
 - ▶ No se atienden inmediatamente si hay una instrucción en ejecución
 - ▶ Quedan registradas en un registro de peticiones pendientes

Tipos de excepciones e interrupciones

▶ Interrupciones

- ▶ Causadas por dispositivos de E/S (entrada/salida)
- ▶ No se atienden inmediatamente si hay una instrucción en ejecución
 - ▶ Quedan registradas en un registro de peticiones pendientes
- ▶ Para su gestión la RSE suele transferir datos entre la CPU y el dispositivo de E/S

Tipos de excepciones e interrupciones

▶ Interrupciones

- ▶ Causadas por dispositivos de E/S (entrada/salida)
- ▶ No se atienden inmediatamente si hay una instrucción en ejecución
 - ▶ Quedan registradas en un registro de peticiones pendientes
- ▶ Para su gestión la RSE suele transferir datos entre la CPU y el dispositivo de E/S
- ▶ El programa interrumpido continúa su ejecución a partir de la siguiente instrucción

Tipos de excepciones e interrupciones

► Interrupciones

- Causadas por dispositivos de E/S (entrada/salida)
- No se atienden inmediatamente si hay una instrucción en ejecución
 - Quedan registradas en un registro de peticiones pendientes
- Para su gestión la RSE suele transferir datos entre la CPU y el dispositivo de E/S
- El programa interrumpido continúa su ejecución a partir de la siguiente instrucción
- El sistema operativo puede planificar otros procesos mientras dura la operación de E/S

El Coprocesador del Sistema CP0

- ▶ El tratamiento de una excepción lo realiza el sistema operativo con la ayuda de soporte de la arquitectura

El Coprocesador del Sistema CP0

- ▶ El tratamiento de una excepción lo realiza el sistema operativo con la ayuda de soporte de la arquitectura
- ▶ En MIPS tenemos el Coprocesador del Sistema (CP0)
 - ▶ Controla las excepciones y la traducción rápida de direcciones virtuales con TLB

El Coprocesador del Sistema CP0

- ▶ El tratamiento de una excepción lo realiza el sistema operativo con la ayuda de soporte de la arquitectura
- ▶ En MIPS tenemos el Coprocesador del Sistema (CP0)
 - ▶ Controla las excepciones y la traducción rápida de direcciones virtuales con TLB
- ▶ Banco de registros específico
 - ▶ Referenciados mediante su número en el código ensamblador
 - ▶ No se pueden utilizar en ninguna de las instrucciones estudiadas hasta ahora

El Coprocesador del Sistema CP0

- ▶ El tratamiento de una excepción lo realiza el sistema operativo con la ayuda de soporte de la arquitectura
- ▶ En MIPS tenemos el Coprocesador del Sistema (CP0)
 - ▶ Controla las excepciones y la traducción rápida de direcciones virtuales con TLB
- ▶ Banco de registros específico
 - ▶ Referenciados mediante su número en el código ensamblador
 - ▶ No se pueden utilizar en ninguna de las instrucciones estudiadas hasta ahora
- ▶ Instrucciones para mover datos al CP0
 - ▶ `mfc0 rt, c0_rd # Copia c0_rd del CP0 en rt`
 - ▶ `mtc0 rt, c0_rd # Copia rt en c0_rd del CP0`

Registros del CP0

- ▶ Registro EPC (Exception Program Counter, \$14)
 - ▶ Registro de lectura/escritura
 - ▶ Antes de llamar a la RSE, el procesador guarda el valor actual del PC en el registro EPC
 - ▶ En caso de excepción, se guarda el PC de la instrucción que produce la excepción
 - ▶ En caso de interrupción, se guarda el PC de la siguiente instrucción a ejecutar

EPC (\$14)

Adreça (PC) de la instrucció interrompuda

Registros del CP0

- ▶ Registro Status (\$12)

- ▶ Bit EXL (Exception Level)

- ▶ Si vale 0 el procesador está en modo usuario
- ▶ Si vale 1 el procesador está en modo sistema
- ▶ En modo sistema se puede acceder a las direcciones del SO y ejecutar instrucciones privilegiadas
- ▶ Si vale 1 (modo sistema) todas las interrupciones están deshabilitadas

- ▶ 8 bits de IM (Interrupt Mask)

- ▶ 1 bit para cada tipo de interrupción (lectura/escritura)
- ▶ Si el bit vale 1, las interrupciones de ese tipo están habilitadas



-
- The diagram shows the Cause register (Cause (\$13)) with a total width of 32 bits. It is divided into two main sections:
 - Interrupt Pending (IP):** The first 15 bits, represented by a solid grey block.
 - ExcCode:** The next 16 bits, represented by a white block with vertical lines. Above this section, the bit positions 15, 8, 6, and 2 are marked.



Códigos de excepción

- Ejemplos de valores del campo *ExcCode* del registro Cause

Núm	Nom	Causa
0	Int	interrupció d'un dispositiu hardware d'E/S
1	Mod	fallada de TLB per pàgina modificada (primera escriptura a pàgina)
2	TLBL	fallada de TLB (o fallada de pàgina) per lectura
3	TLBS	fallada de TLB (o fallada de pàgina) per escriptura
4	AdEL	error d'adreça per lectura (accés mal alineat, o a espai de sistema no permès)
5	AdES	error d'adreça per escriptura (accés mal alineat, o a espai de sistema no permès)
6	IBE	error de bus (instruction fetch), p.ex. adreça física inexistent
7	DBE	error de bus (dades de load/store), p.ex. adreça física inexistent
8	Sys	crida al sistema (causada per la instrucció syscall)
9	Bp	breakpoint (causada per la instrucció break)
10	RI	instrucció reservada. p.ex. codis d'operació inexistents
11	CpU	coprocessador no implementat, p.ex. accés a CP0 en mode usuari
12	Ov	overflow aritmètic d'enters (instruccions add, sub, addi)
13	Tr	trap (causada per una instrucció de trap)
15	FPE	excepció de coma flotant: consultar detalls als registres del coprocessador CP1

Acciones hardware en caso de excepción o interrupción

- ▶ En caso de excepción:
 - ▶ La instrucción en curso se aborta
 - ▶ No se escribe en registros o memoria, no se incrementa el PC

Acciones hardware en caso de excepción o interrupción

- ▶ En caso de excepción:
 - ▶ La instrucción en curso se aborta
 - ▶ No se escribe en registros o memoria, no se incrementa el PC
- ▶ En caso de interrupción:
 - ▶ Se finaliza la ejecución de la instrucción en curso
 - ▶ Se comprueba que el dispositivo i ha hecho una petición ($IP_i = 1$) y que está habilitado ($EXL=0$ y $IM_i = 1$)

Acciones hardware en caso de excepción o interrupción

- ▶ En caso de excepción:
 - ▶ La instrucción en curso se aborta
 - ▶ No se escribe en registros o memoria, no se incrementa el PC
- ▶ En caso de interrupción:
 - ▶ Se finaliza la ejecución de la instrucción en curso
 - ▶ Se comprueba que el dispositivo i ha hecho una petición ($IP_i = 1$) y que está habilitado ($EXL=0$ y $IM_i = 1$)
- ▶ Se guarda el PC en el registro EPC

Acciones hardware en caso de excepción o interrupción

- ▶ En caso de excepción:
 - ▶ La instrucción en curso se aborta
 - ▶ No se escribe en registros o memoria, no se incrementa el PC
- ▶ En caso de interrupción:
 - ▶ Se finaliza la ejecución de la instrucción en curso
 - ▶ Se comprueba que el dispositivo i ha hecho una petición ($IP_i = 1$) y que está habilitado ($EXL=0$ y $IM_i = 1$)
- ▶ Se guarda el PC en el registro EPC
- ▶ Se escribe la causa de la excepción en el campo ExcCode del registro Cause
 - ▶ En caso de varias excepciones simultáneas, se establece un sistema de prioridades

Acciones hardware en caso de excepción o interrupción

- ▶ Se pone a 1 el bit EXL del registro Status (Modo sistema)

Acciones hardware en caso de excepción o interrupción

- ▶ Se pone a 1 el bit EXL del registro Status (Modo sistema)
- ▶ El procesador escribe en el PC la dirección base de la RSE

Acciones hardware en caso de excepción o interrupción

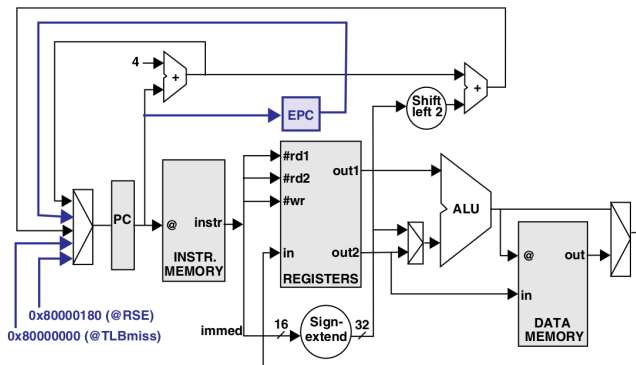
- ▶ Se pone a 1 el bit EXL del registro Status (Modo sistema)
- ▶ El procesador escribe en el PC la dirección base de la RSE
- ▶ Dos rutinas de gestión de excepciones en MIPS:
 - ▶ RSE (rutina genérica): $PC = 0x80000180$
 - ▶ TLBmiss (para fallos de TLB): $PC = 0x80000000$

Acciones hardware en caso de excepción o interrupción

- ▶ Se pone a 1 el bit EXL del registro Status (Modo sistema)
- ▶ El procesador escribe en el PC la dirección base de la RSE
- ▶ Dos rutinas de gestión de excepciones en MIPS:
 - ▶ RSE (rutina genérica): $PC = 0x80000180$
 - ▶ TLBmiss (para fallos de TLB): $PC = 0x80000000$
- ▶ Ambas rutinas están en el espacio de direcciones del SO
 - ▶ Solo se pueden ejecutar en modo sistema

Acciones hardware en caso de excepción o interrupción

- Posible diseño de un procesador MIPS incluyendo el soporte para excepciones



Acciones software (RSE)

1. La RSE ha de preservar el estado del programa interrumpido
 - ▶ Se guardan **TODOS** los registros de enteros en la pila (excepto \$k0 y \$k1)
 - ▶ No es necesario salvar los registros de coma flotante

Acciones software (RSE)

1. La RSE ha de preservar el estado del programa interrumpido
 - ▶ Se guardan **TODOS** los registros de enteros en la pila (excepto \$k0 y \$k1)
 - ▶ No es necesario salvar los registros de coma flotante
2. Se identifica la causa de la excepción (ExcCode en el registro Cause) y se salta a la subrutina específica que realiza el tratamiento

Acciones software (RSE)

1. La RSE ha de preservar el estado del programa interrumpido
 - ▶ Se guardan **TODOS** los registros de enteros en la pila (excepto \$k0 y \$k1)
 - ▶ No es necesario salvar los registros de coma flotante
2. Se identifica la causa de la excepción (ExcCode en el registro Cause) y se salta a la subrutina específica que realiza el tratamiento
3. Se ejecuta la subrutina específica, que puede:
 - ▶ Abortar el programa
 - ▶ Bloquearlo, dando paso a otros programas
 - ▶ Solucionar el problema y continuar la ejecución del programa

Acciones software (RSE)

4. En caso de syscall, ajustar el EPC sumándole 4 (siguiente instrucción)

Acciones software (RSE)

4. En caso de syscall, ajustar el EPC sumándole 4 (siguiente instrucción)
5. Restaurar todos los registros salvados en la pila

Acciones software (RSE)

4. En caso de syscall, ajustar el EPC sumándole 4 (siguiente instrucción)
5. Restaurar todos los registros salvados en la pila
6. Retornar, usando la instrucción **eret** (exception return):
 - ▶ Pone EXL=0: pasa a modo usuario y habilita las interrupciones
 - ▶ Copia EPC al PC: salta a la dirección de retorno guardada en EPC

Ejemplo sencillo de RSE (I)

```
.ktext 0x80000180          # secció "kernel text"
RSE:
# Salvar TOTS els registres incloent $hi, $lo però no $k0, $k1, $sp
addiu $sp, $sp, -128
sw    $1, 0($sp)
sw    $2, 4($sp)
sw    $3, 8($sp)
sw    $4, 12($sp)
. . .
sw    $31, 124($sp)

# Passar paràmetres Cause i EPC a una rutina
mfc0  $a0, $13             # Passa Cause Register (amb ExcCode)
mfc0  $a1, $14             # Passa EPC
jal   handler_dispatcher   # invocarà el handler que correspongui
                                # i decidirà on ha de retornar la RSE
mtc0  $v0, $14             # copiar resultat a EPC
```

Ejemplo sencillo de RSE (II)

```
# Restaurar registres i pila
addiu $sp, $sp, -128
lw    $1, 0($sp)
lw    $2, 4($sp)
lw    $3, 8($sp)
lw    $4, 12($sp)
. . .
lw    $31, 124($sp)
addiu $sp, $sp, 128

# Retornar al programa d'usuari, posant EVL=0 i PC=EPC
eret
```

	Afirmació	V	F
1.-	Si la CPU rep una petició d'interrupció per part d'un dispositiu d'E/S, la instrucció en curs s'interromp sense arribar a produir cap resultat, i la CPU passa a executar la rutina RSE.		
2.-	En MIPS, la traducció d'adreces virtuals a físiques en una instrucció <code>lw</code> pot arribar a produir fins a quatre excepcions.		
3.-	Quan cerquem la traducció d'un número de pàgina (VPN) al TLB, perquè hi hagi un encert de TLB cal trobar una entrada amb el mateix VPN i que tingui el bit de presència <code>V=1</code> .		
4.-	Les escriptures del bit de Dirty del TLB segueixen una política d'escriptura immediata.		
5.-	Les fallades de TLB en el MIPS provoquen una excepció i s'executa la RSE genèrica.		
6.-	En MIPS, quan s'acaba d'executar la RSE, es torna a la instrucció següent a la que ha provocat l'excepció o la interrupció.		
7.-	La sincronització dels dispositius d'entrada/sortida en un computador, es gestiona exclusivament per interrupcions.		
8.-	En MIPS, una instrucció <code>lb</code> mai pot produir una excepció per accés no alineat.		
9.-	En un computador que disposa de memòria virtual amb TLB sempre cal fer dos accessos a memòria principal per cada referència: un a la Taula de Pàgines i l'altre a l'adreça de la referència.		
10.-	L'excepció de divisió per zero pot ser inhibida a través del camp <code>Interrupt Mask</code> .		

	Afirmació	V	F
1.-	Si en una cache canviem la política d'escriptura <i>immediata amb assignació a retardada amb assignació</i> , sense cap més canvi, el nombre total de fallades no canvia.		
2.-	En memòria virtual paginada, sempre que reemplacem de la memòria física una pàgina cal escriure-la en disc.		
3.-	En un sistema amb memòria virtual, la mida total d'un programa i les seves dades no pot excedir la capacitat de la memòria física.		
4.-	Si un cert número enter de 8 bits es representa en Ca2 per 0x8B, el mateix número es representa en Ca1 per 0x8C		
5.-	Si el resultat d'un càlcul en format normalitzat de coma flotant causa underflow, es pot reduir l'error absolut de precisió si el podem expressar en format denormal.		
6.-	A l'inici de la rutina genèrica de servei d'excepcions de MIPS (RSE) aquesta sols ha de salvar a la pila aquells registres segurs que es modifiquin durant l'execució de la RSE.		
7.-	Un programa en mode usuari pot copiar un registre qualsevol de la CPU al coprocessador CP0 per mitjà de la instrucció <code>mtc0</code> .		
8.-	El tractament de les excepcions al MIPS es fa únicament en dues rutines de servei, una per a la fallada de TLB i una altra per a la resta d'excepcions.		
9.-	En el MIPS, el camp IM (Interrupt Mask) del registre Status usat en la gestió de les interrupcions serveix per indicar les peticions d'interrupció que no han de ser ateses quan s'acabi l'execució de la instrucció actual.		
10.-	La codificació <i>en excés</i> dels exponents dels nombres en coma flotant permet que la comparació de les seves magnituds es pugui fer amb un comparador de nombres naturals.		