Tratamiento de Excepciones en MIPS

Elías Todorovich

Arquitectura I - Curso 2013

Riesgos de Control

- Las direcciones del PC no son secuenciales (PC = PC + 4) en los siguientes casos:
  - Saltos condicionales (`beq`, `bne`)
  - Saltos incondicionales (`j, jal, jr`)
  - Excepciones
- Los riesgos de control son menos frecuentes que los riesgos de datos…
  - … pero no existe nada tan efectivo para los riesgos de control como el adelantamiento de datos para los riesgos de datos.

Riesgos de Control

- Posibles soluciones
  - Espera (*stall*, nop) (impacta en CPI)
  - Mover la decisión hacia una de las primeras etapas en el pipeline.
    - Eso reduce la cantidad de ciclos perdidos
  - Demorar la decisión (salto retardado)
    - Requiere que el compilador provea una solución
  - Predicción de saltos

Dos tipos de *stalls*

1. Se inserta el equivalente a una instrucción nop por hardware.
   - Se evita que avancen las instrucciones en las primeras etapas del pipeline (las últimas en el código) por un ciclo.
     - Esto se hace mediante control de escritura en registros del pipeline.
   - Se inserta el nop haciendo cero los bits de control en el registro de pipeline adecuado.
   - Las instrucciones en las últimas etapas del pipeline (las primeras en el codigo) progresan normalmente.

Dos tipos de *stalls*

2. Se reemplaza (flush) una instrucción en el pipeline por el equivalente a una instrucción nop.
   - Ejemplo: las instrucciones después de un salto incondicional `j`.
   - Se deben hacer cero los bits de control en la instrucción que se quiere reemplazar.

Excepciones

- Las excepciones son otra forma de riesgo de control. Algunos ejemplos:
  - *Overflow* en una instrucción te tipo R.
  - Tratar de ejecutar una instrucción no definida.
  - Una petición desde un dispositivo de E/S.
  - Un pedido de servicio al OS (por ejemplo un fallo de página).
  - Un fallo de hardware.

Tratamiento de Excepciones en MIPS

- El procesador segmentado detiene la ejecución de la instrucción errónea.
- Las instrucciones previas completan su ejecución.
- Se eliminan (*flush*) todas las instrucciones siguientes.
- Se carga en un registro especial el código de la causa de la excepción.
- Se guarda la dirección de la instrucción que genera la excepción.

HW

- Se salta a una dirección predeterminada.
  - Código de la rutina de tratamiento (*exception handler code*)
  - El software (OS) mira la causa de la excepción y procede en consecuencia.

SW

Dos tipos de Excepciones

1. Interrupciones – asíncronas a la ejecución del programa
   - Causadas por eventos externos.
   - Se pueden tratar **entre** instrucciones.
     - Se pueden completar las instrucciones en el pipeline antes de pasar el control a la rutina de servicio de la interrupción del OS.
   - Se suspende y reanuda el programa de usuario.

Dos tipos de Excepciones

- Excepciones (o *Traps*) – síncronas a la ejecución del programa
  - Causadas por eventos internos
  - La instrucción que genera la excepción **no termina** su ejecución.
  - Se pasa el control a la rutina de tratamiento de la excepción del OS.

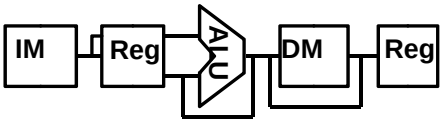
Rutina de servicio

- Leer *cause*, e invocar a la rutina correspondiente
- If reiniciable
  - Tomar acciones correctivas
  - Reintentar la ejecución del programa usando el EPC (-4)
- Si no…
  - El SO termina el programa (*kill*)
  - Retornar una indicación de la causa al usuario, incluyendo el EPC

Un mecanismo alternativo

- En otros procesadores se usan interrupciones vectorizadas
  - El SO sabe la causa de la excepción por la dirección de la rutina
- Ejemplo:
  - Opcode indefinido: C000 0000
  - Overflow: C000 0020
  - …: C000 0040
- Las 8 instrucciones entre una dirección y la siguiente:
  - Tratan la interrupción, o
  - Hacen un salto a una rutina más larga.

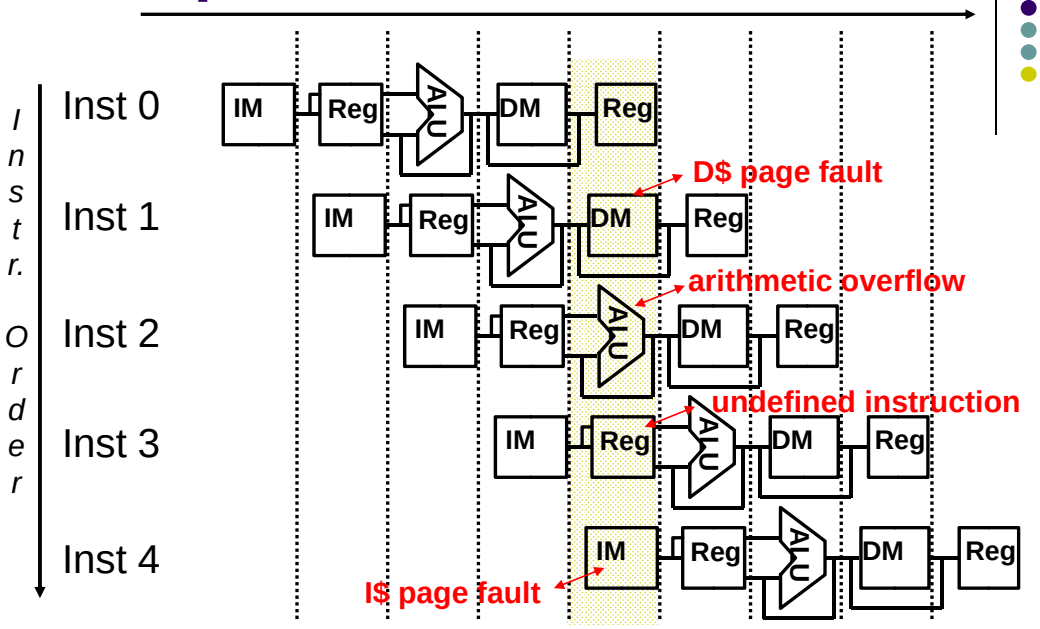
¿Dónde ocurren las excepciones?



	Etapa(s)	Sincrono
• Arithmetic overflow	EX	si
• Instruction no definida	ID	si
• Fallo de pagina o TLB	IF, MEM	si
• Petición de I/O	cualquiera	no
• Falla de Hardware	cualquiera	no

- Cuidado que pueden ocurrir múltiples excepciones en un sólo ciclo de reloj.

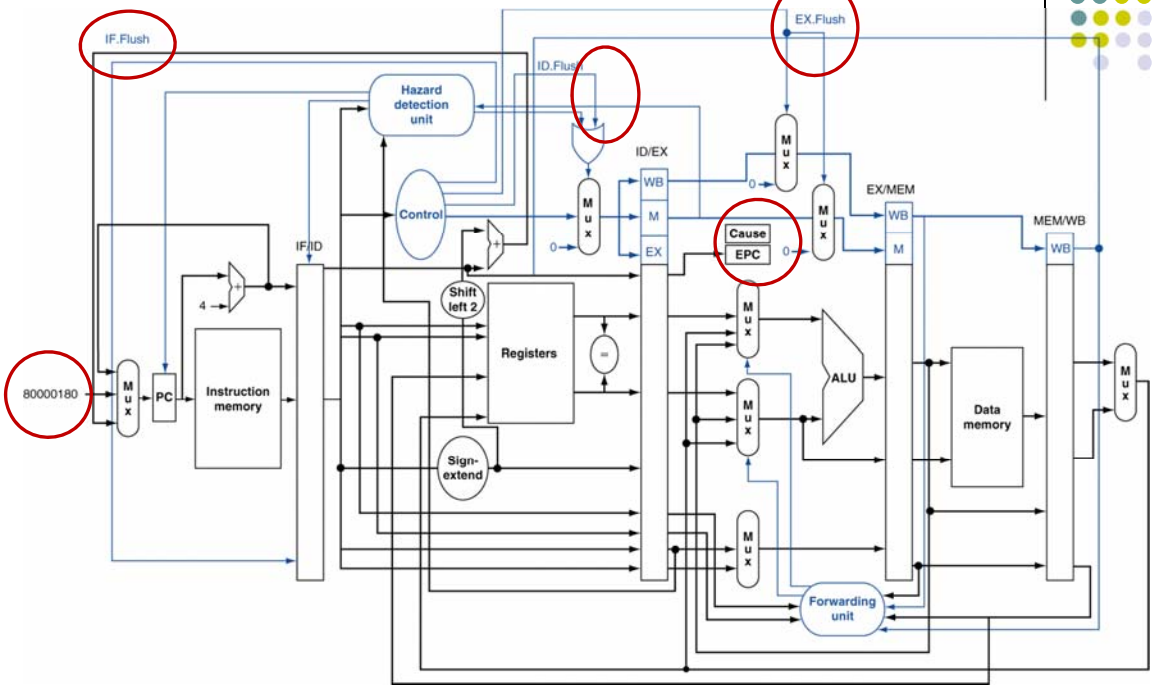
Excepciones Simultáneas



- El procesador ordena las excepciones de manera que la instrucción más adelantada se interrumpa primero.

HW adicional para Excepciones

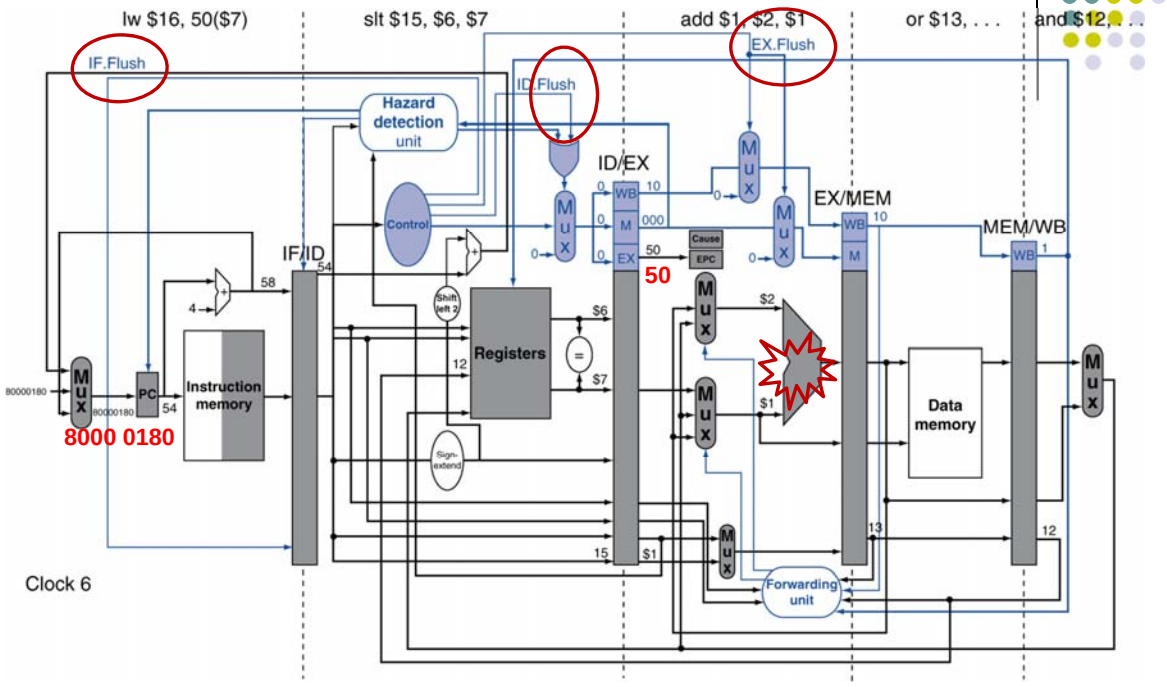
- Registro de Causa
- Registro EPC (*Exception* PC)
  - Se guarda la dirección de la instrucción conflictiva
- Se debe poder cargar el PC con la dirección de la rutina de tratamiento de excepciones
  - El multiplexor al PC tiene una entrada adicional con un valor fijo, la dirección del *exception handler* - (por ejemplo 8000 0180_{hex})
- *Flushing* de la instrucción conflictiva y las subsiguientes.

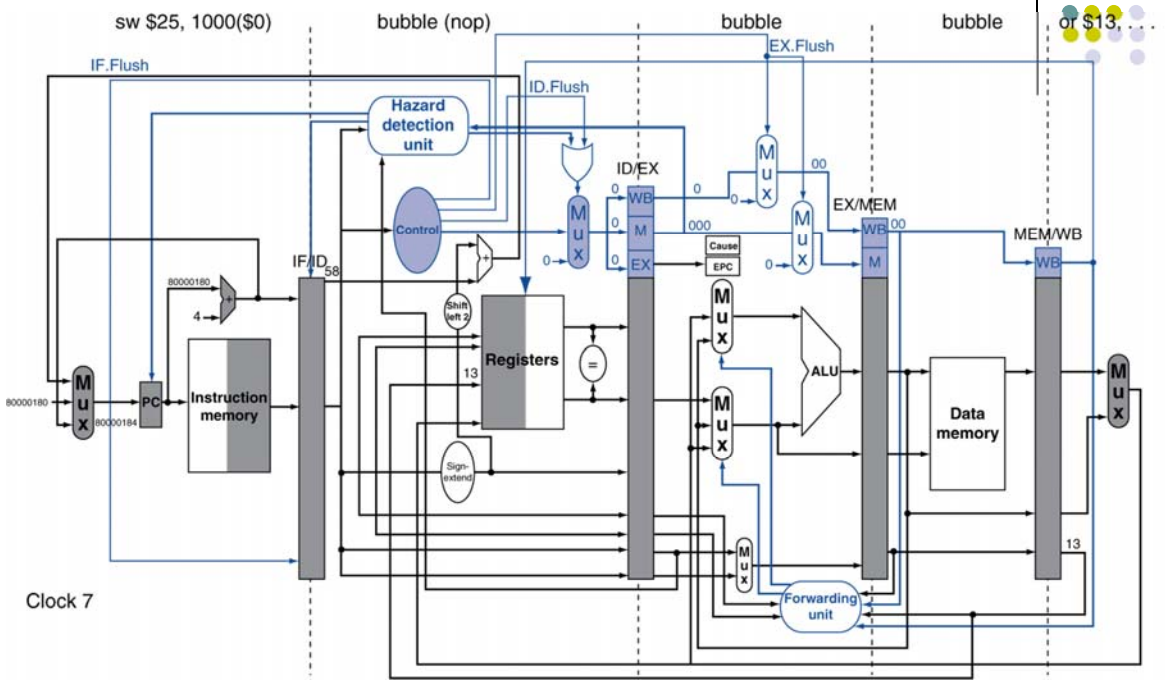


Example (Patterson, pp. 388)

```
00000044    and   $12,  $2,   $5
00000048    or    $13,  $2,   $6
0000004C    add   $1,   $2,   $1
00000050    slt   $15,  $6,   $7
00000054    lw    $16,  50($7)
…
80000180    sw    $25, 1000($0)
80000184    sw    $26, 1004($0)
```

add genera overflow, pero no debe actualizarse $1 con el nuevo valor.


lw $16, 50($7)
slt $15, $6, $7
add $1, $2, $1
or $13, . . .
and $12, . . .
IF.Flush
ID.Flush
EX.Flush
Hazard detection unit
Control
ID/EX
EX/MEM
MEM/WB
IF/ID
WB
M
EX
Cause
EPC
50
Registers
Instruction memory
Data memory
Forwarding unit
Sign-extend
Shift left 2
PC
8000 0180
Clock 6


sw $25, 1000($0)
bubble (nop)
bubble
bubble
or $13, . . .
IF.Flush
ID.Flush
EX.Flush
Hazard detection unit
Control
ID/EX
EX/MEM
MEM/WB
IF/ID
WB
M
EX
Cause
EPC
Registers
ALU
Instruction memory
Data memory
Forwarding unit
Sign-extend
Shift left 2
PC
Clock 7

Contexto de Diseño

- El control es la parte más difícil en el diseño de un procesador
  - La más difícil de hacer bien
  - La más difícil de hacer rápida
- La parte más difícil en el diseño de la unidad de control es la implementación de las excepciones.

Resumen de MIPS segmentado

- Todos los procesadores modernos usan pipelines para tener un CPI tendiendo a 1.
- El periodo de reloj en un procesador segmentado está limitado por la etapa más lenta.
- Se deben tratar los riesgos de datos y de control.
  - Caminos de adelantamiento.
  - Espera, flushing de instrucciones (sube el CPI).
  - Predicción de saltos.

Tratamiento de Excepciones en MIPS

Elías Todorovich

Arquitectura I - Curso 2013