

CS 203A
Advanced Computer Architecture

Lecture 7

Exception Handling, Multicycle operations

Instructor: Jun Yang

Review

- Pipeline hazards
 - Structural hazard
 - Data hazard
 - Control hazard
- Difficulties in pipeline
 - Exception handling

Outline

- Difficulties in pipeline
 - Exception handling
 - ISA complications
- Multicycle FP operations

Interrupts (3)

• Implementation issues:

- Difficult case - internal interrupt and must be restartable.
Actions:
 - ❖ Force a trap in IF in next cycle;
 - ❖ Disable all writes in pipeline following the faulting instruction;
 - ❖ Save restart state (PC of faulting instruction). Problem: if fault inst. is in branch delay slot and branch is taken! Must save PCs of delay slot size + 1.
- Pipelined implementations: k concurrent interrupts:

IF	Page fault, misaligned access, memory protection.
ID	Illegal opcode.
EX	Arithmetic interrupt.
MEM	Same as in IF.

Interrupts (4)

Example:

	T+1	T+2	T+3	T+4	T+5	T+6
LW	IF	ID	EX	MEM	WB	
ADD		IF	ID	EX	MEM	WB

- Easy case: LW page faults at MEM and ADD has exception at EX. Service page fault first, restart, exception will reappear.
- Difficult case: LW has page fault in MEM and ADD page faults in IF, out of order interrupts.
 - ❖ Precise approach: service interrupts in order
 - Create an interrupt status vector for each instruction showing stage of occurrence of interrupt.
 - Check for interrupts at MEM to WB transition, before update to state.
 - ❖ Less precise approach: service interrupts out-of order.

ISA can be hard for exceptions and pipelining

- Instructions that modify machine state in the middle of execution may cause imprecise exceptions.
 - VAX string copy inst. make the working storage in register.
 - In general, providing precise handling is costly and difficult.
- Complex instructions with memory addressing and multiple operands may cause multiple interrupts in a single instruction.
- Long running inst. make the pipeline design difficult because it introduces enormous number of hazards and forwarding conditions.
 - *Microinstruction*: a simple instruction used in sequences to implement a more complex instruction set (Intel, VAX 8800 etc.)
 - ❖ Easy for pipelining.
 - ❖ They look like MIPS!

WAW Example in Reasonable Code

```
BNEZ    R1, foo
DIV.D   F0, F2, F4; moved into delay slot from fall through
```

```
.....
.....
```

```
foo:   L.D   F0, qrs
```

Implications - 3

C. Hazard detection before issue: made easier by separate FP and INT reg files

- ❖ Check for structural hazards: wait for DIV operation to complete before issuing a new one, make sure reg write port is available before issuing instruction.
- ❖ Check for RAW hazards: wait until all src registers are not listed as dest registers of pending instructions.
- ❖ Check for WAW hazards: any inst in A1, ... A4, D, M1, ... M7 that has same dest as current inst: stall current inst.

D. Forwarding: from EX/MEM, A4/MEM, M7/MEM, D/MEM, or MEM/WB to source registers.

Interrupts

• Precise Interrupts

- Problem: out of order completion - cause imprecise exception.

A. Ignore problem:

- ❖ used in 60's and 70's, mostly by supercomputers;
- ❖ virtual memory and IEEE FP standards require precise interrupts;
- ❖ many recent CPUs support a two modes operation: precise or not, switch via compiler, precise is much slower (e.g. Alpha 21064, 21164, Power 1 & 2, MIPS R8000).

B. Buffer results:

- ❖ until all preceding inst have completed.
- ❖ Might need a lot of buffers, results from finished but not retired inst must be bypassed to subsequent instructions: complex logic.

Interrupts - 2

Variants:

- ❖ History file (Cyber 180/990) keeps original values of reg.
- ❖ Future file: register values kept in future file until instruction is retired. Main register file is updated from future file, it holds correct state. Variant used in PowerPC 620 & R8000.

C. Software approach:

somewhat imprecise.

Example: I1 long running instruction, interrupts
 I2 ... In-1 unfinished inst
 In finished inst

Trap handling routine can finish I1 ... In-1 in software.

MIPS-like pipeline simplification: If I2 ... In are all integer inst, then if In is finished so are I2 ... In-1. So we only worry about FP inst. Used in Sun SPARC.

Interrupts - 3

D. Hybrid approach:

- ❖ allow inst issue iff it is certain that no pending inst can interrupt.
- ❖ If interrupt then all prior inst can complete and no inst after can.
- ❖ Means stalling pipeline when condition cannot be certain.
- ❖ Used in MIPS R2000/3000, R4000 and Intel Pentium.