Lecture 10 – the Branch Delay Slot and Exceptions

Karl R. Wilcox
Karl@cs.rhul.ac.uk

Objectives

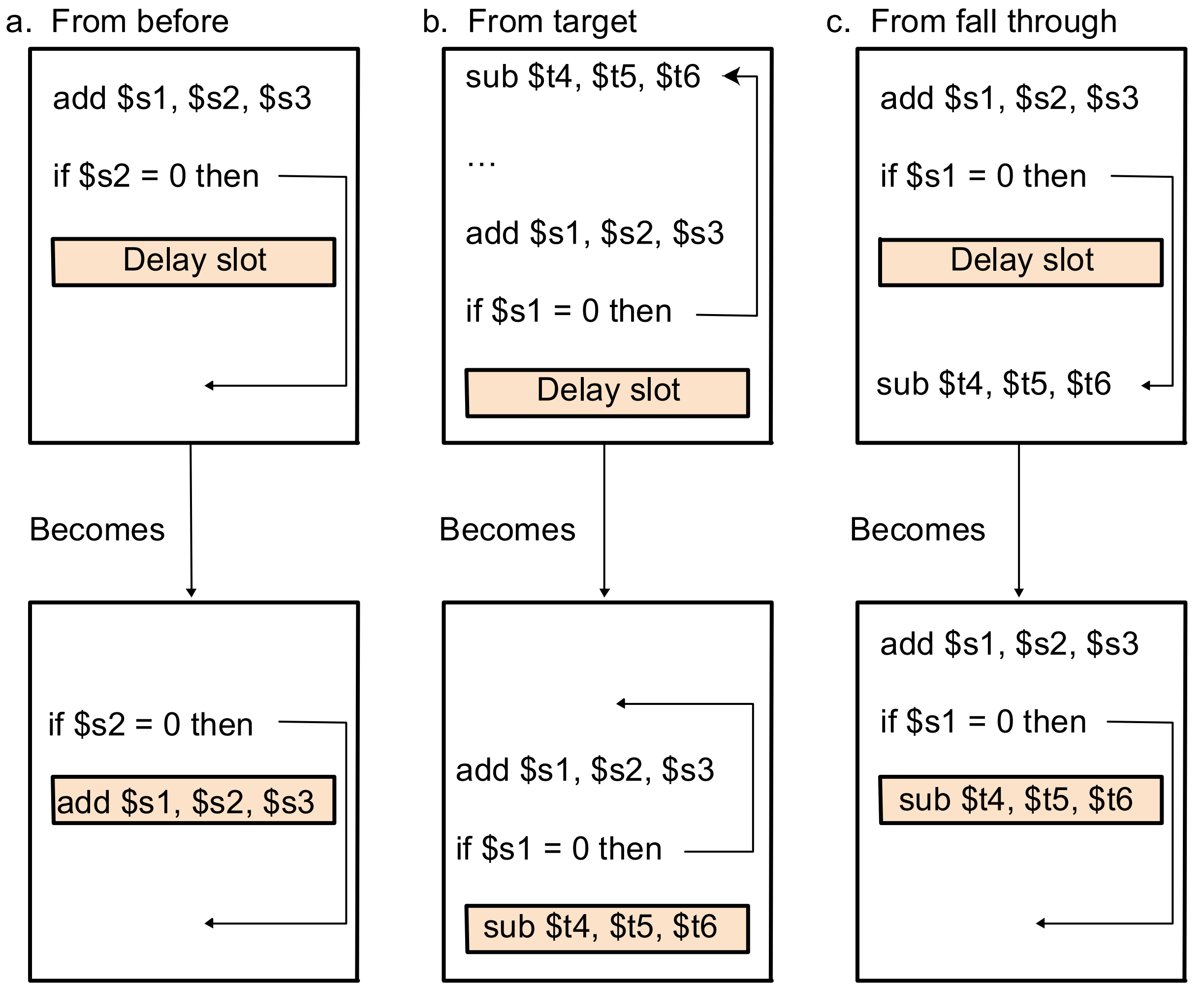
- In this lecture we will cover
  - The MIPS Branch Delay Slot
  - Handling exceptions (interrupts)

(diagrams are from Patterson & Hennessy)

The MIPS Branch Delay Slot

- Last lecture we looked at improvements to processing conditional branches
- Even with these improvements, in the MIPS processor there is still one instruction in the pipeline when the branch outcome is known
- Rather than undo this instruction, call it a “Branch Delay Slot” and let the compiler use it!
  - In about 50% of cases, a useful instruction can be inserted
  - If not, use a NO-OP

Using the Branch Delay Slot



Exceptions

- **Exceptions can occur for multiple reasons**
  - **I/O Device request (e.g. transfer complete)**
  - **A user program invoking the operating system (software interrupt)**
  - **Undefined instruction**
  - **Arithmetic overflow**
  - **Hardware malfunction**

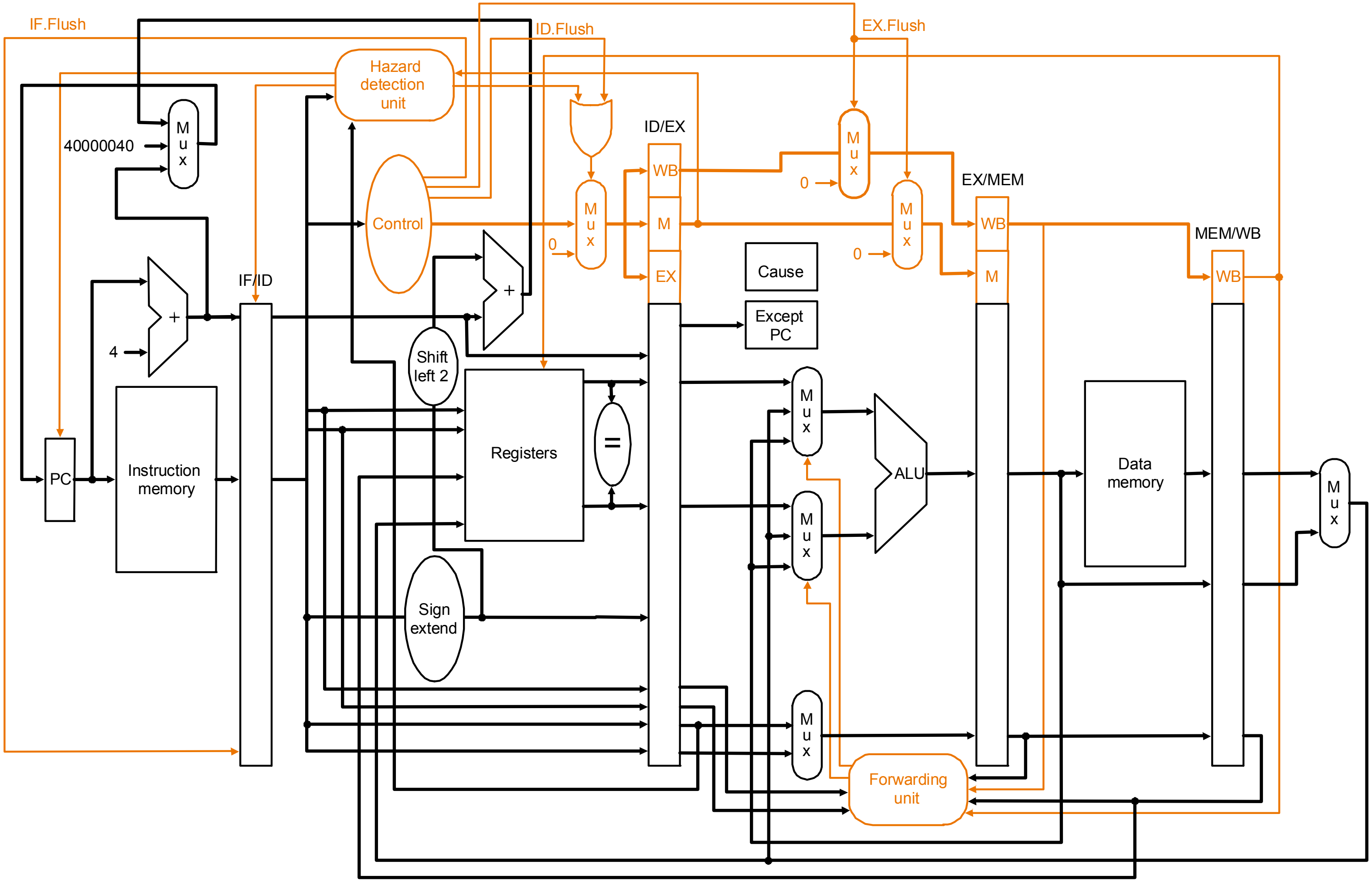
Which Instruction To Interrupt?

- Some instructions cause the interrupt
  - Undefined instruction
  - Arithmetic overflow
- Other interrupts are not instruction specific but must still be serviced
  - With 5, overlapping instructions in the pipeline, where do we stop?

Instruction Caused Interrupts

- **We must stop execution as soon as the interrupt occurs**
  - **Because other instructions in the pipeline may be dependent on the “faulty” instruction**
- **We “flush” the pipeline**
  - **Needs new control signals on multiplexors**
  - **Causes no registers to be updated**
- **We put the address of the offending instruction in EPC register and source in the cause register**
- **We call the instruction handling routine**

Interrupt Hardware



Royal Holloway University of London

Non-Instruction Based Interrupts

- There is some flexibility in when to service the interrupt
  - Known as *imprecise exceptions*
  - Still put the "interrupted" instruction address into EPC
    - So we can restart execution from EPC + 4

- Choices
  - Wait for the pipeline to empty
  - Select a "simple" instruction to stop on
  - Flush the pipeline

Summary

- The Branch Delay Slot “legislates” away a potential pipeline hazard
  - The problem is for the compiler to sort out
- Exception handling can be complicated in a pipeline architecture
  - Requires extra hardware and control lines
- Pipelining is a generally applicable “design pattern”
  - Many of the problems (and solutions) are also applicable in other situations

Next Lecture

- Other processor architectures -
  - Superscalar
  - Dynamic pipelining
  - Vector
  - Parallel