

# Exceptions

Jinkyu Jeong ([jinkyu@skku.edu](mailto:jinkyu@skku.edu))

Computer Systems Laboratory

Sungkyunkwan University

<http://csl.skku.edu>

# Exceptions and Interrupts

- “Unexpected” events requiring change in flow of control
  - Different ISAs use the terms differently
- Exception
  - Arises within the CPU
    - e.g., undefined opcode, overflow, syscall, ...
- Interrupt
  - From an external I/O controller
- Dealing with them without sacrificing performance is hard

\*\*\*STOP: 0x000000D1 (0x00000000, 0xF73120AE, 0xC0000008, 0xC0000000)

A problem has been detected and Windows has been shut down to prevent damage to your computer

DRIVER\_IRQL\_NOT\_LESS\_OR\_EQUAL

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

\*\*\*\* ABCD.SYS - Address F73120AE base at C0000000, DateStamp 36B072A3

Kernel1 Debugger Using: COM2 (Port 0x2F8, Baud Rate 19200)

Beginning dump of physical memory

Physical memory dump complete. Contact your system administrator or technical support group.

An example of exception... 😊

# Handling Exceptions

- In MIPS, exceptions managed by a System Control Coprocessor (CP0)
- Save PC of offending (or interrupted) instruction
  - In MIPS: Exception Program Counter (EPC)
- Save indication of the problem
  - In MIPS: Cause register
  - We'll assume 1-bit
    - 0 for undefined opcode, 1 for overflow
- Jump to handler at 8000 0180

# Handling Exceptions: An Alternate Mechanism

- **Vectored Interrupts**
  - Handler address determined by the cause
- **Example:**
  - Undefined opcode:       C000 0000
  - Overflow:                C000 0020
  - ....:                     C000 0040
- **Instructions either**
  - Deal with the interrupt, or
  - Jump to real handler

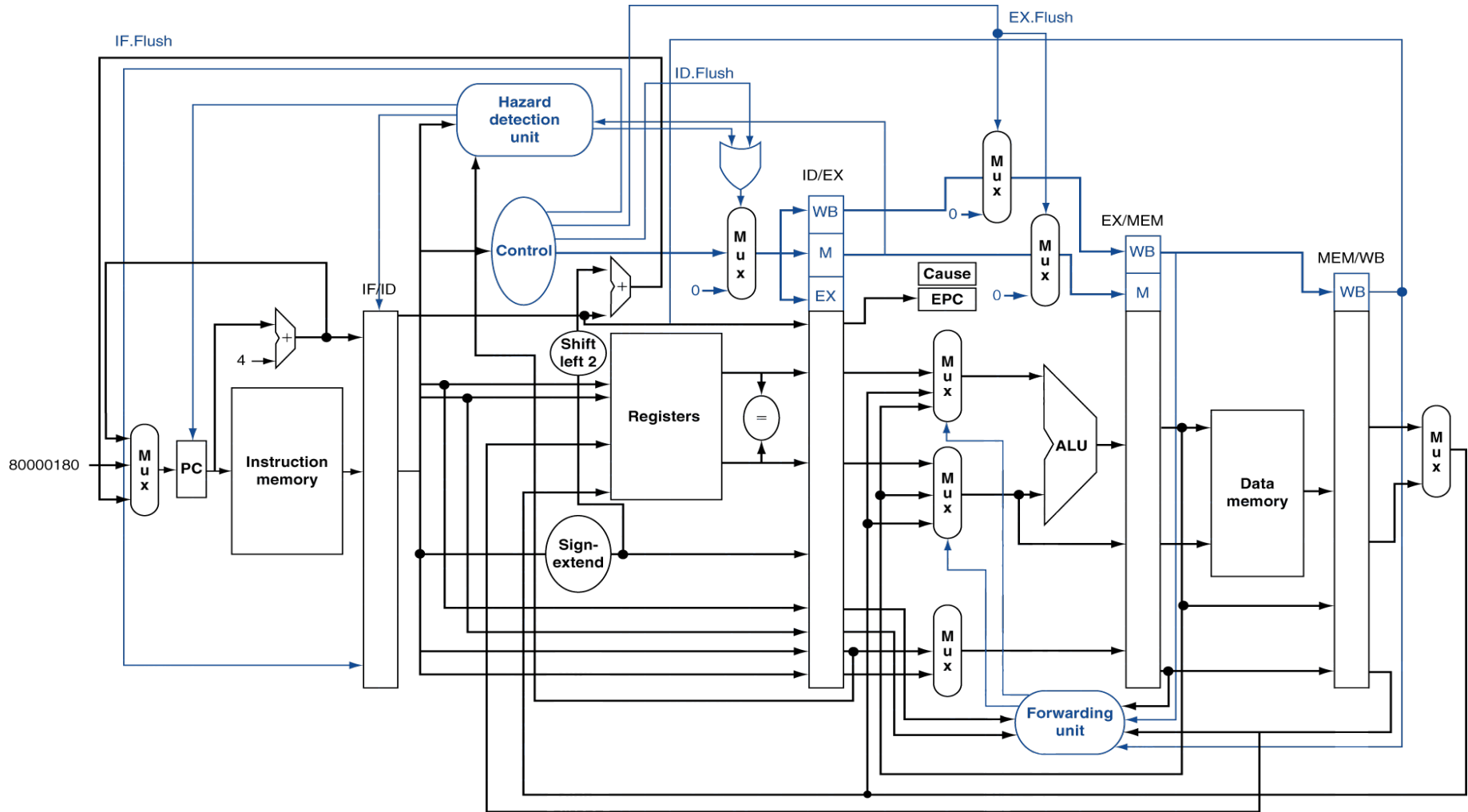
# Handling Exceptions: Handler Actions

- Read cause, and transfer to relevant handler
- Determine action required
- If restartable
  - Take corrective action
  - Use EPC to return to program
- Otherwise
  - Terminate program
  - Report error using EPC, cause, ...

# Exceptions in a Pipeline

- Another form of control hazard
- Consider overflow on add in EX stage  
add \$1, \$2, \$1
  - Prevent \$1 from being clobbered
  - Complete previous instructions
  - Flush add and subsequent instructions
  - Set Cause and EPC register values
  - Transfer control to handler
- Similar to mispredicted branch
  - Use much of the same hardware

# Pipeline with Exceptions





# Exception Properties

- Restartable exceptions
  - Pipeline can flush the instruction
  - Handler executes, then returns to the instruction
    - Refetched and executed from scratch
- PC saved in EPC register
  - Identifies causing instruction
  - Actually PC + 4 is saved
    - Handler must adjust

# Exception Example

- Exception on **add** in

40 sub	\$11,	\$2,	\$4
44 and	\$12,	\$2,	\$5
48 or	\$13,	\$2,	\$6
<b>4C add</b>	<b>\$1,</b>	<b>\$2,</b>	<b>\$1</b>
50 slt	\$15,	\$6,	\$7
54 lw	\$16,	50(\$7)	

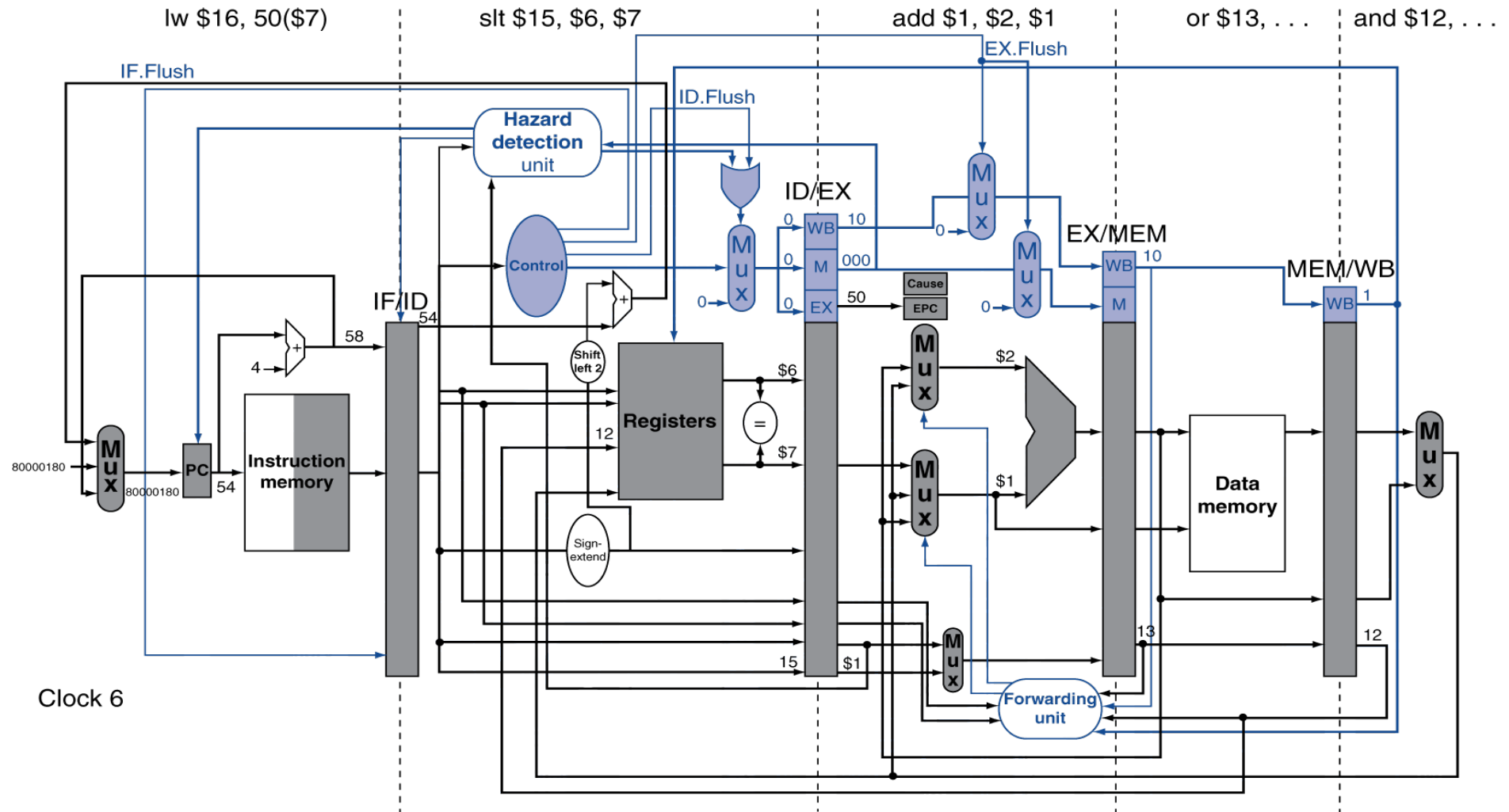
...

- **Handler**

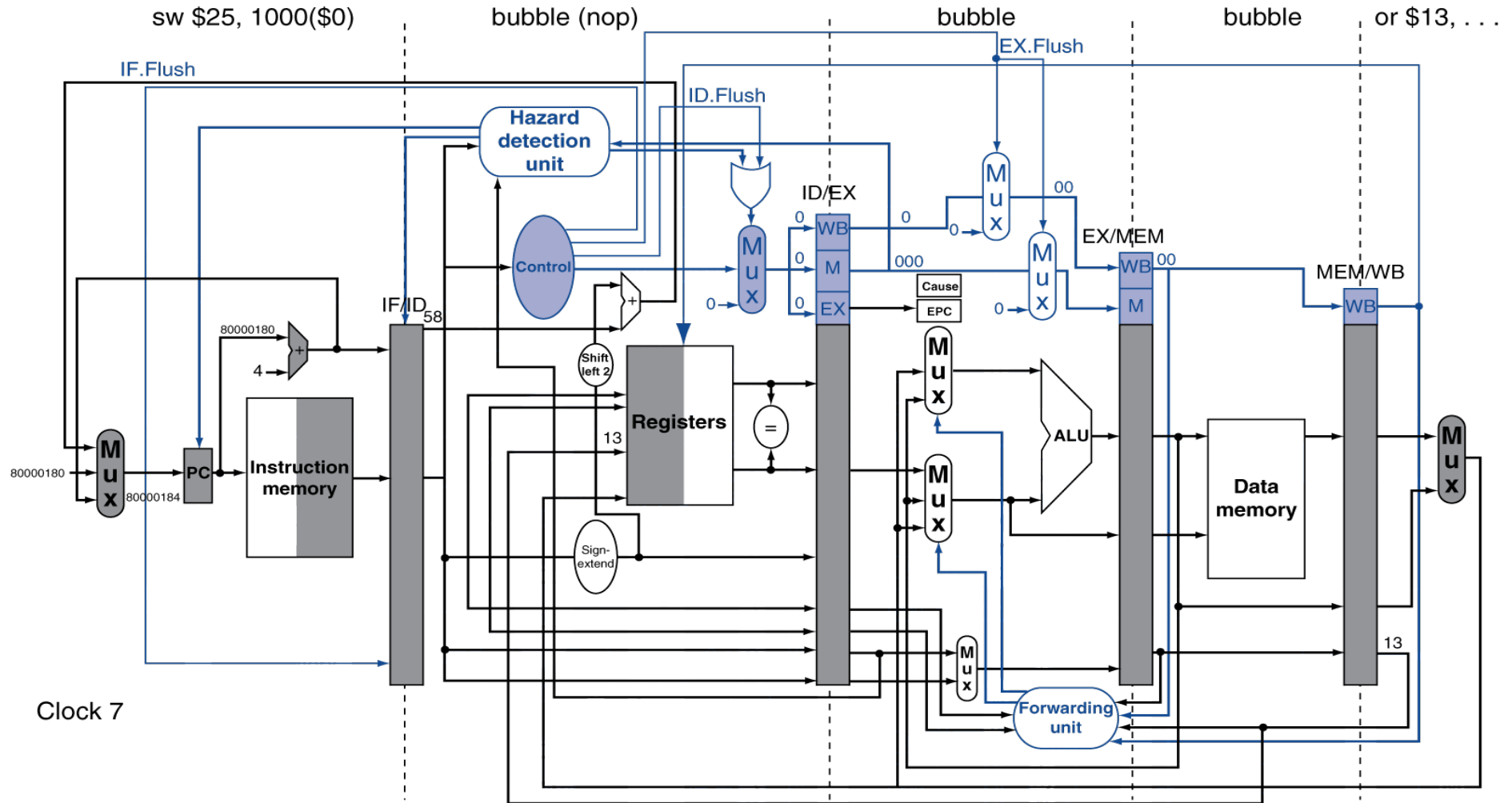
80000180	sw	\$25,	1000(\$0)
80000184	sw	\$26,	1004(\$0)

...

# Exception Example



# Exception Example



# Multiple Exceptions

- Pipelining overlaps multiple instructions
  - Could have multiple exceptions at once
- Simple approach: deal with exception from earliest instruction
  - Flush subsequent instructions
  - “Precise” exceptions
- In complex pipelines
  - Multiple instructions issued per cycle
  - Out-of-order completion
  - Maintaining precise exceptions is difficult!

# Imprecise Exceptions

- Just stop pipeline and save state
  - Including exception cause(s)
- Let the handler work out
  - Which instruction(s) had exceptions
  - Which to complete or flush
    - May require “manual” completion
- Simplifies hardware, but more complex handler software
- Not feasible for complex multiple-issue out-of-order pipelines