Chapter 6

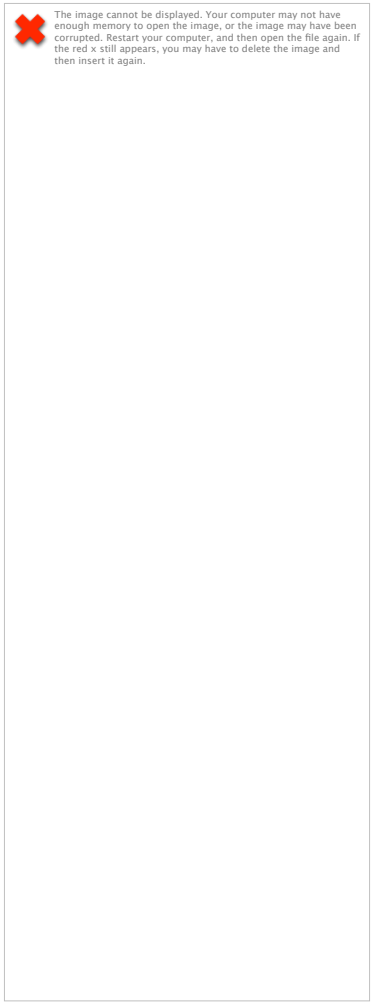
***Digital Design and Computer Architecture*, 2nd Edition**

David Money Harris and Sarah L. Harris

ELSEVIER

Chapter 6 :: Topics

- Introduction (done)
- Assembly Language (done)
- Machine Language (done)
- Programming (done)
- Addressing Modes (done)
- Lights, Camera, Action: Compiling, Assembling, & Loading (done)
- Odds and Ends (now)



Exceptions

- Unscheduled function call to *exception handler*
- Caused by:
  - Hardware, also called an *interrupt*, e.g., keyboard
  - Software, also called *traps*, e.g., undefined instruction
- When exception occurs, the processor:
  - Records the cause of the exception
  - Jumps to exception handler (at instruction address 0x80000180)
  - Returns to program

Exception Registers

- Not part of register file
  - **Cause**: Records cause of exception
  - **EPC** (Exception PC): Records PC where exception occurred
- `EPC` and `Cause`: part of Coprocessor 0
- Move from Coprocessor 0
  - `mfc0 $k0, EPC`
  - Moves contents of `EPC` into `$k0`

Exception Causes

Exception	Cause
Hardware Interrupt	0x00000000
System Call	0x00000020
Breakpoint / Divide by 0	0x00000024
Undefined Instruction	0x00000028
Arithmetic Overflow	0x00000030

Exception Flow

- Processor saves cause and exception PC in `Cause` and `EPC`
- Processor jumps to exception handler (0x80000180)
- Exception handler:
  - Saves registers on stack
  - Reads `Cause` register

    ```
    mfc0 $k0, Cause
    ```
  - Handles exception
  - Restores registers
  - Returns to program

    ```
    mfc0 $k0, EPC
    jr $k0
    ```

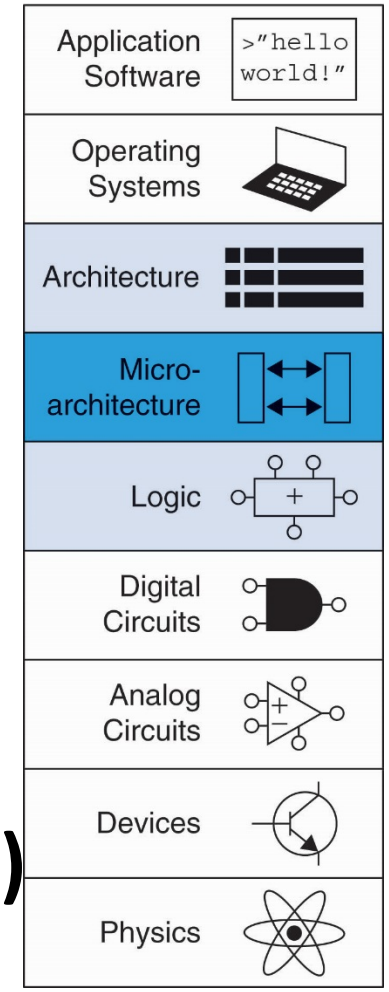
Chapter 7

***Digital Design and Computer Architecture*, 2nd Edition**

David Money Harris and Sarah L. Harris

Chapter 7 :: Topics

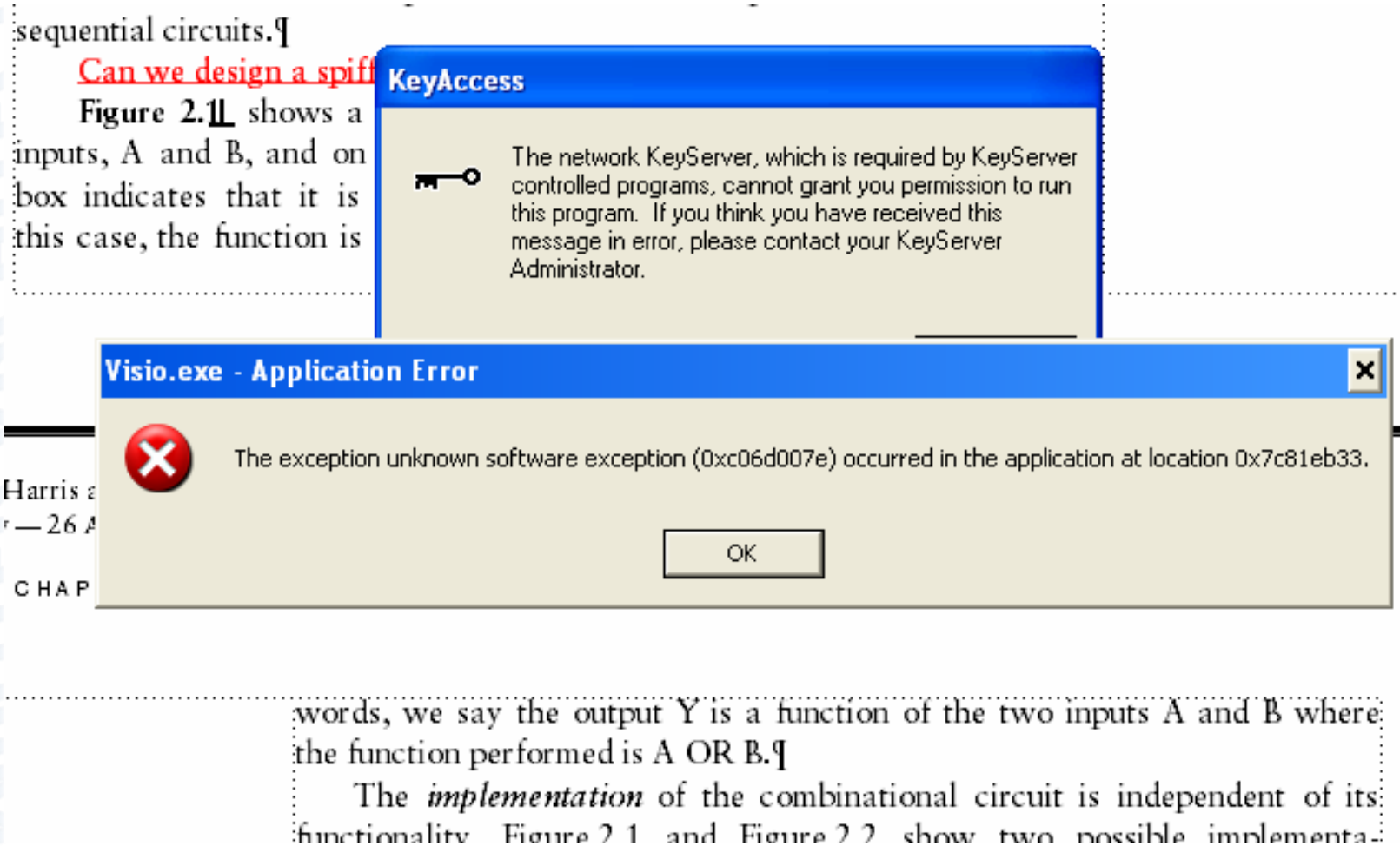
- **Introduction (done)**
- **Performance Analysis (done)**
- **Single-Cycle Processor (done)**
- **Multicycle Processor (done)**
- **Pipelined Processor (done)**
- **Exceptions (now)**
- **Advanced Microarchitecture (later)**



Review: Exceptions

- Unscheduled function call to *exception handler*
- Caused by:
  - Hardware, also called an *interrupt*, e.g. keyboard
  - Software, also called *traps*, e.g. undefined instruction
- When exception occurs, the processor:
  - Records cause of exception (`Cause` register)
  - Jumps to exception handler (0x80000180)
  - Returns to program (`EPC` register)

Example Exception



Exception Registers

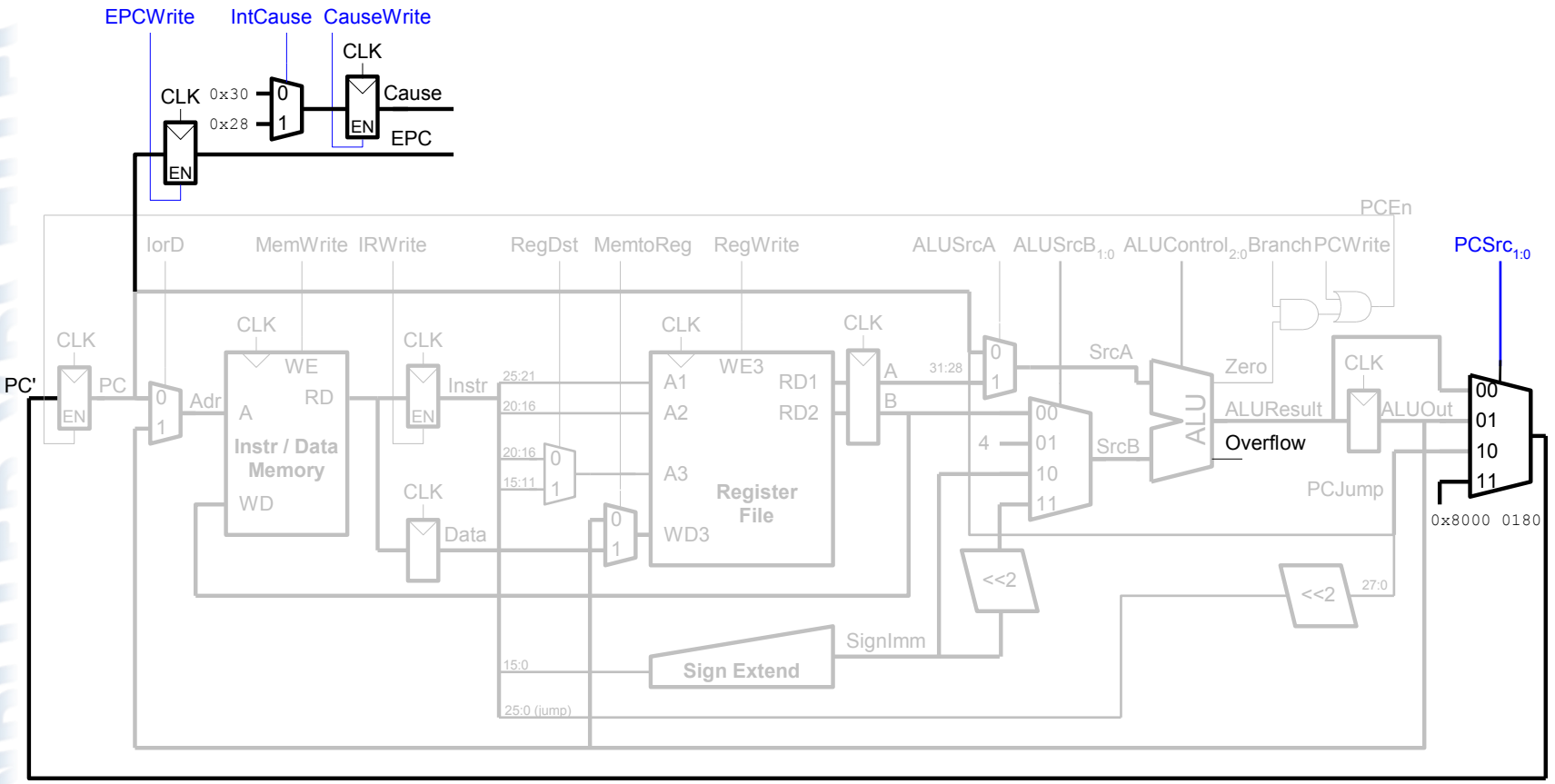
- Not part of register file
  - `Cause`
    - Records cause of exception
    - Coprocessor 0 register 13
  - `EPC` (Exception PC)
    - Records PC where exception occurred
    - Coprocessor 0 register 14
- Move from Coprocessor 0
  - `mfc0 $t0, Cause`
  - Moves contents of `Cause` into `$t0`

Exception Causes

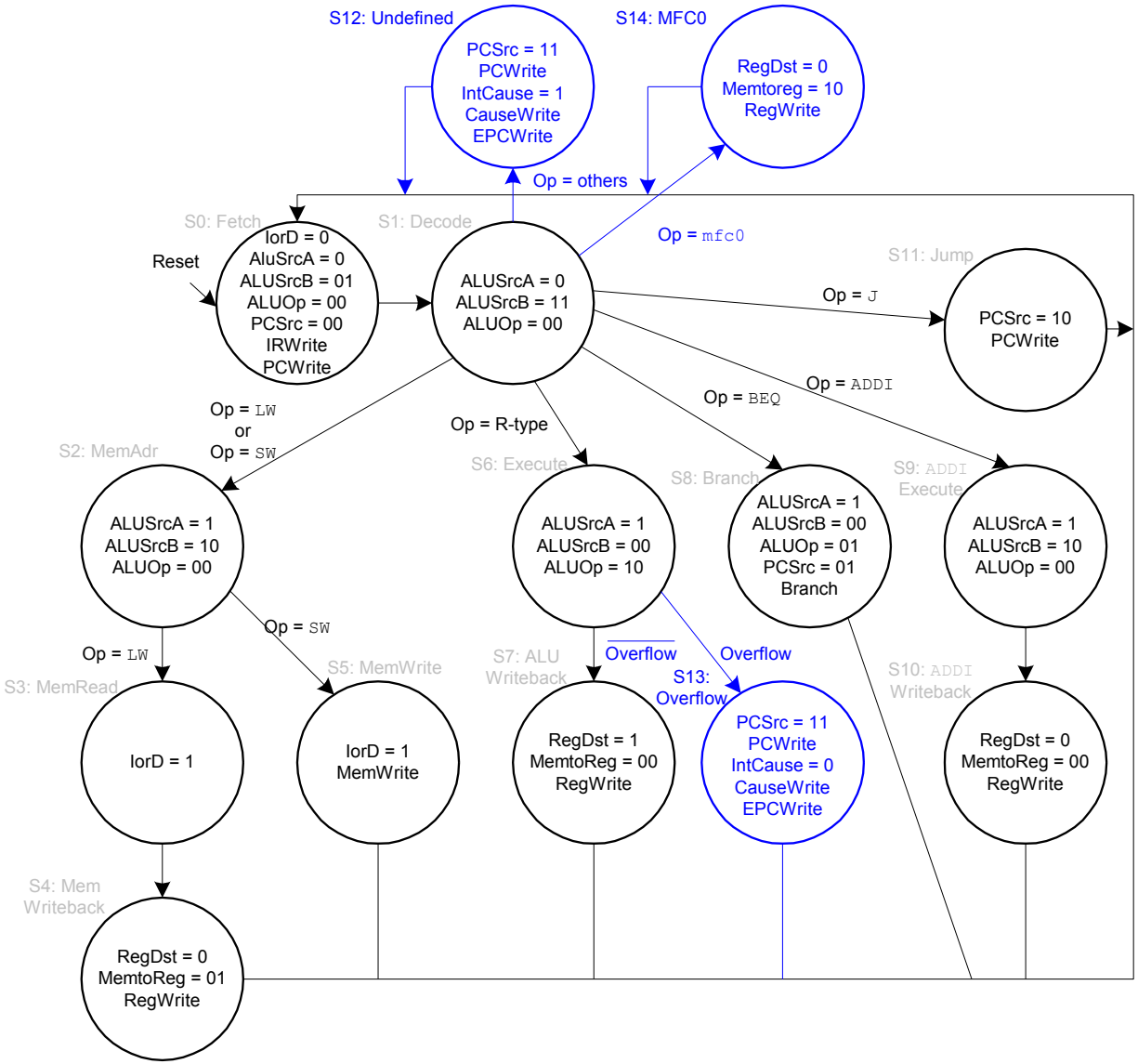
Exception	Cause
Hardware Interrupt	0x00000000
System Call	0x00000020
Breakpoint / Divide by 0	0x00000024
Undefined Instruction	0x00000028
Arithmetic Overflow	0x00000030

Extend multicycle MIPS processor to handle last two types of exceptions

Exception Hardware: EPC & Cause



Control FSM with Exceptions



Exception Hardware: `mfc0`