CS/COE0447: Computer Organization and Assembly Language

Exception handling

Sangyeun Cho

Dept. of Computer Science
University of Pittsburgh

From last lecture

- We looked at the multi-cycle control scheme in detail
- Multi-cycle control can be implemented using FSM
- FSM is composed of some combinational logic and memory element

Today's topics

- Exceptions
  - Reset
  - Interrupt
  - Exceptions

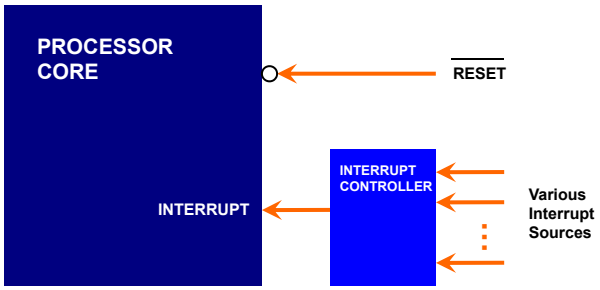
Terminology

- Exception
  - a condition causing the processor to escape from its normal control flow, to a pre-defined location (to deal with the condition)
  - Types
    - Reset (external)
    - Interrupt (external)
    - Exception (internal)
    - Software interrupt (internal)
- The exception mechanism plays a critical role in building a computer system
  - OS implementation
  - Time sharing system
  - Supporting a new instruction on an existing architecture

Exception types

- Reset
  - External (asynchronous to internal processing)
  - Puts the processor into a known, initial state
  - Jump to "reset vector" address
- Interrupt
  - External (asynchronous to internal processing)
  - Saves the current program counter and status register in special registers
  - Jump to "interrupt vector" address
- Exception
  - Internal (associated with some instruction; e.g., divide by 0)
  - Saves the current program counter and status register in special registers
  - Jump to "exception vector" address
- Software interrupt
  - Internal (associated with a special instruction called "software interrupt")
  - Used to call system functions
  - Used to emulate a special instruction
  - Used to support debugging software/hardware

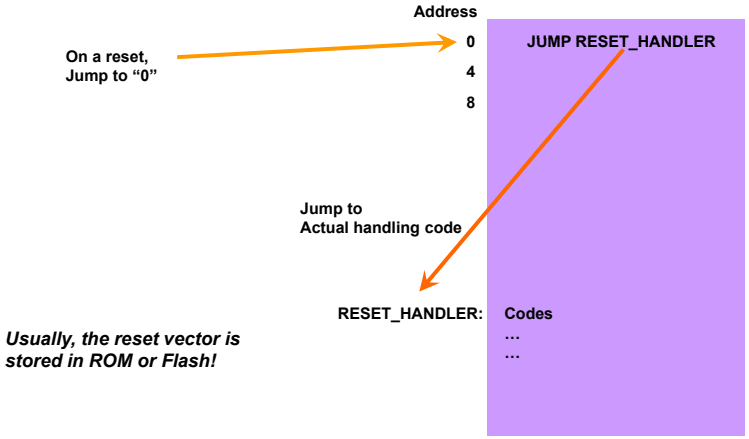
Interrupt & reset



Reset exception

- When you boot your machine;
- Internally, power-on generates a pulse (typically 1→0→1)
  - Then this "low" pulse is stretched by some logic to the point when
    - Several valid clocks have been generated and fed into the processor
    - Power level is stabilized
  - All the critical registers inside a processor will be initialized to a known value when this "reset pulse" is applied
  - After the pulse is applied, the processor, at its initial state, jumps to a pre-defined location (typically 0)
- Then, another jump instruction is usually executed
  - Now, jump to a location where boot code is
  - Boot code runs, initializes all other hardware, starts the operating system

Reset processing



Interrupt processing

- There are many interrupt sources
  - I/O devices
    - Keyboards
    - Mouse
    - UART (Universal Asynchronous Receiver-Transmitter)
    - Flash card controller (e.g., SmartMedia, memory stick, MMC, SD, CF, …)
  - DMA (Direct Memory Access)
  - Other peripheral devices
    - Timers
    - Watchdog timer
  - From outside the CPU
- These devices raise "interrupt" so that the CPU can stop what it has been doing and start paying attention to them
  - CPU saves the current PC and status automatically
  - Interrupt handler saves registers on stack
  - After interrupt is handled, interrupt handler restores the previous state and return to the previous job

Exception processing

- Here, "exception" means an internally-generated interrupt condition
  - E.g., Divide-by-Zero, undefined instruction
- On detecting these events, the processor saves the current PC and the state, and jumps to an exception handler by jumping to a vector
- Exception handler
  - Saves registers
  - Figures out why exception occurred
  - Does some appropriate actions
  - Restores registers
  - Returns to execute the original instruction
- Software interrupt can be thought of as a type of exception
  - To implement some system call

Exception vector

Address	Exception	Mode on entry	I state on entry	F state on entry
0x00000000	Reset	Supervisor	Set	Set
0x00000004	Undefined instruction	Undefined	Set	Unchanged
0x00000008	Software interrupt	Supervisor	Set	Unchanged
0x0000000C	Prefetch Abort	Abort	Set	Unchanged
Address	**Exception**	**Mode on entry**	**I state on entry**	**F state on entry**
0x00000010	Data Abort	Abort	Set	Unchanged
0x00000014	Reserved	Reserved	-	-
0x00000018	IRQ	IRQ	Set	Unchanged
0x0000001C	FIQ	FIQ	Set	Set

(Taken from ARM7 Technical Reference Manual)

Processor support

- Refer to Chapter 5.6
- Necessary functions
  - On detecting an interrupt or exception
    - Processor saves the associated PC in a special register
    - Processor saves the current state (status register) in a special register
    - Processor jumps to a pre-defined vector address
  - Instructions
    - To return to the exception-causing location
    - To manipulate special registers
  - Processor modes
    - Normal mode – user programs run in this mode
    - Privileged mode – interrupt handlers and operating system codes run in this mode
    - Processor switches from normal mode to privileged mode on exceptions/interrupts

To summarize…

- Exceptions and interrupts are an important part of any processor
- Modern operating system heavily depends on the hardware support for exceptions

- Interrupt mechanism allows efficient use of CPU cycles by getting hold of it only when it is necessary
- Implementing this interrupt mechanism usually is complicated and requires a lot of validation efforts in a processor design cycle