

CISC 3320

C05a: Interrupts

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Acknowledgement

- This slides are a revision of the slides by the authors of the textbook

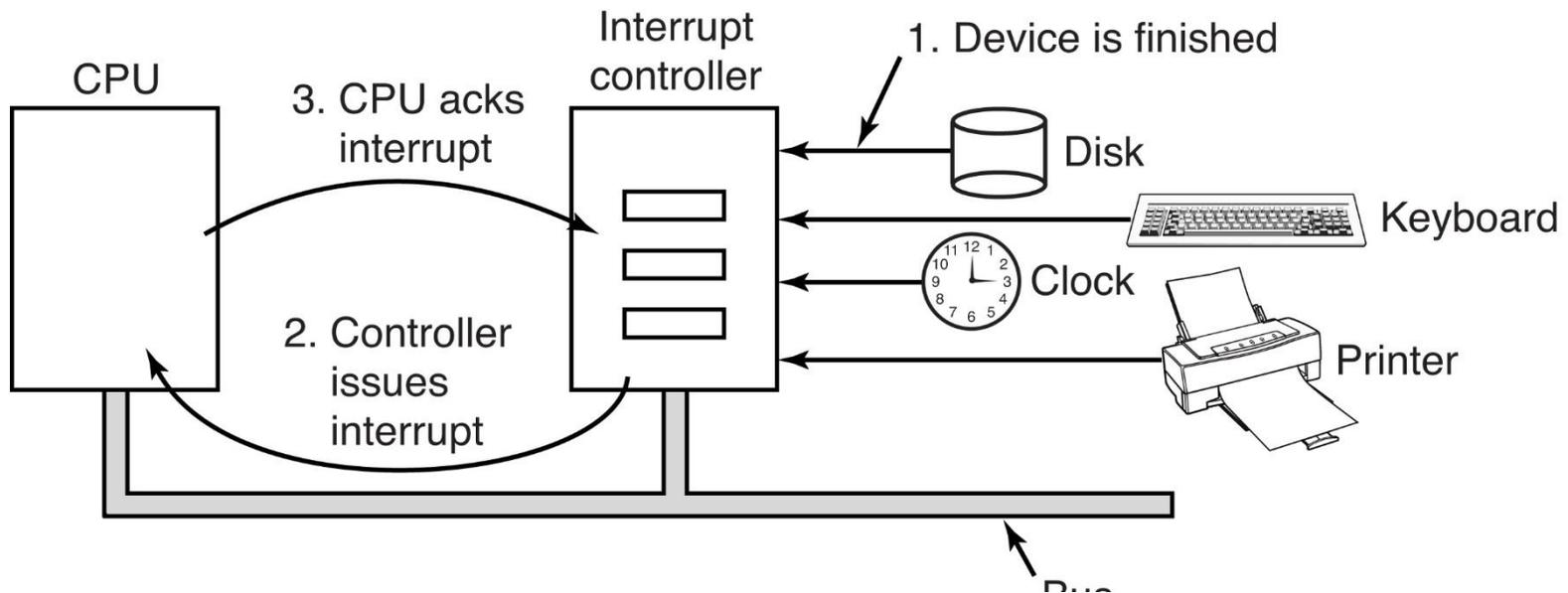
Outline

- Concept of interrupts
- Interrupt service routing, interrupt vector, and interrupt vector table
- Interrupt handling
- Interrupt design consideration

OS and Interrupts

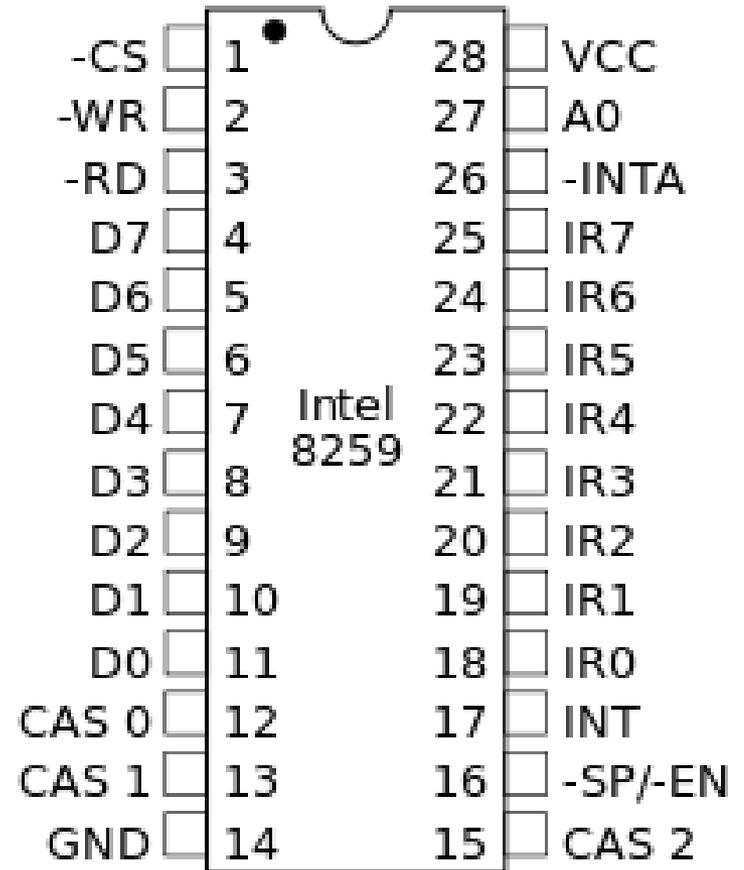
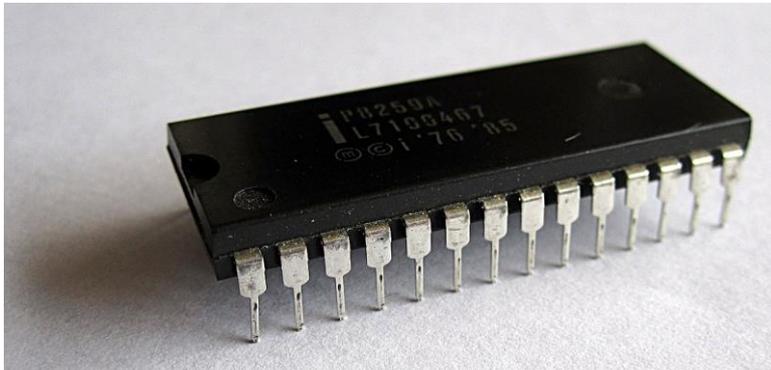
- An operating system is interrupt driven
 - Timers
 - I/O
 - ...

How an Interrupt Happens?



- [Figure 5-5 in Tanenbaum & Bos, 2014]

Example: CPU & Interrupt Controller



Interrupts

- Interrupt transfers control to the interrupt service routine generally
- Two sources of interrupts
 - External (hardware-generated) interrupts: interrupts are generally caused by hardware
 - Software generated interrupts: a trap or exception is a software-generated interrupt caused either by an error or a user request
- Interrupt vector (interrupt descriptor by Intel)
 - Interrupt service routine: interrupt handler, a program processes the interrupt
 - Interrupt vector table: consists of interrupt vectors
 - Interrupt vector: the address of an interrupt handler
- Interrupt architecture must save the address of the interrupted instruction

Interrupt Vectors

- Address to interrupt routines
 - Some PC event/interrupt-vector numbering

Vector Number	Description
0	Divide Error
1	Debug Exception
...	
6	Invalid Opcode
...	
32-255	Maskable Interrupts (device generated)

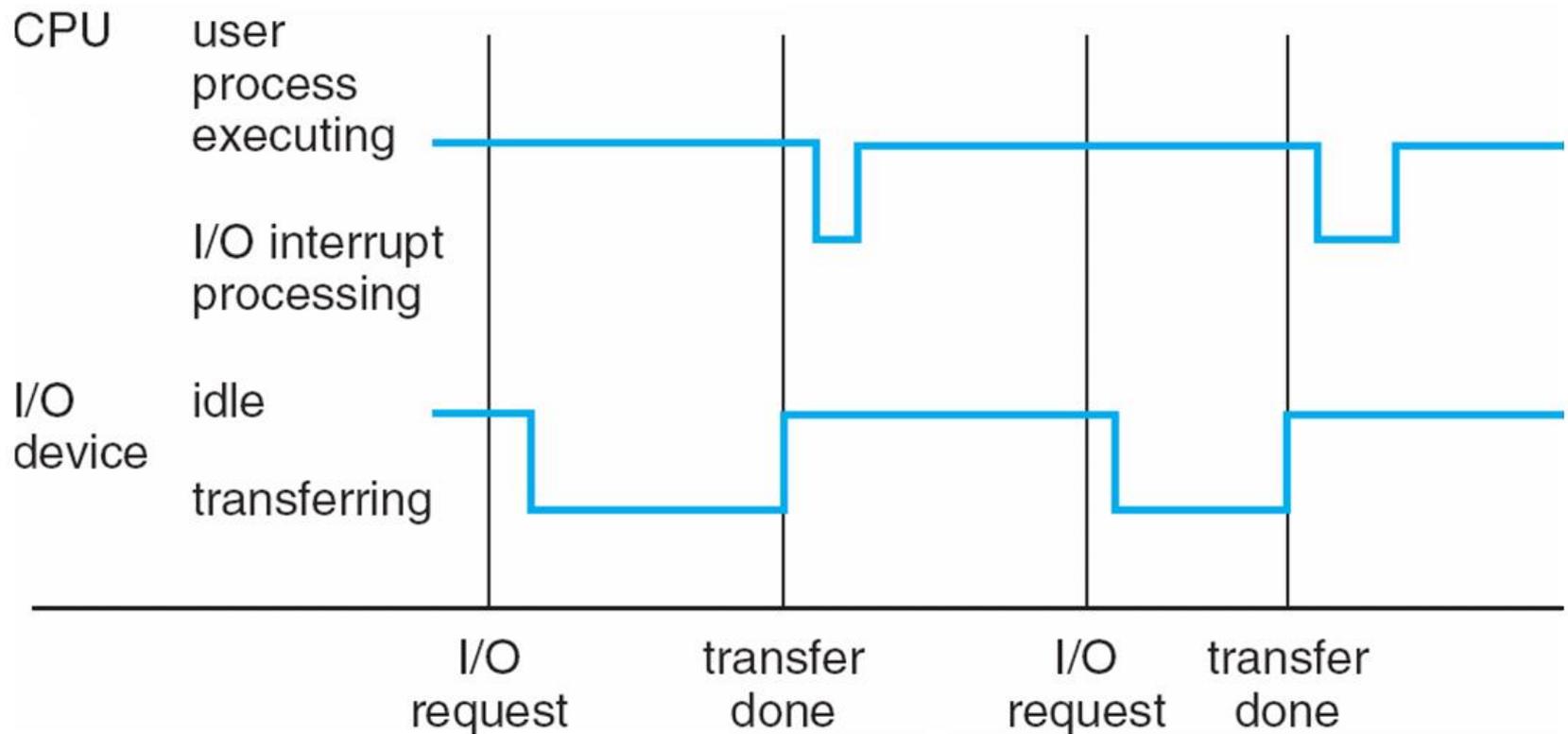
Handling Interrupt

- CPU senses its interrupt-request line after each instruction
- When it is "lit", CPU saves the current state
 - Example: push registers PSW and PC to the stack
- CPU jumps to the interrupt-handler routine at a fixed address in the memory
- Interrupt-handler routine completes its task and restore the CPU state
 - Pop the registers from the stack

Design Consideration: Interrupts

- Maskable and nonmaskable interrupts
- Interrupt priorities and interrupt chaining
- Exceptions and software interrupts (traps)
- Precise and imprecise interrupts

Interrupt Timeline: I/O Interrupts



Exceptions and Interrupts

- Interrupt mechanism also used for exceptions
 - Terminate process, crash system due to hardware error
- Page fault executes when memory access error
 - System call executes via trap to trigger kernel to execute request
- Multi-CPU systems can process interrupts concurrently
 - If operating system designed to handle it
- Used for time-sensitive processing, frequent, must be fast

Questions?

- Reviewed the concept of interrupts