MODULE 4: Computer Organization and MARIE

Lecture 4.3
Instruction Processing

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech

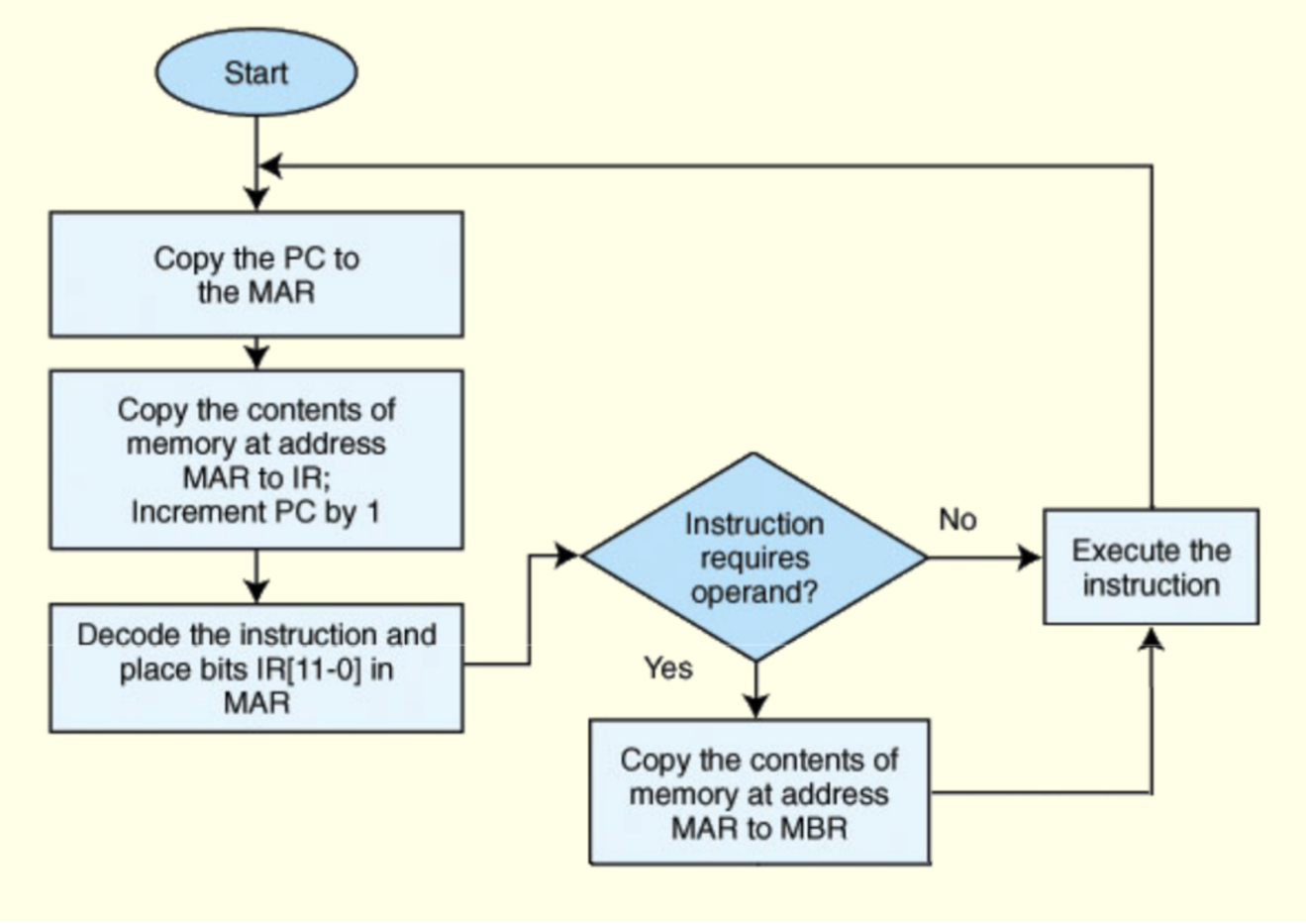
Lecture 4.3 Objectives

- Describe the fetch-decode-execute cycle
- Explain when traps and interrupts are used
- Describe how an interrupt is processed
- Enumerate the steps that must be taken by a trap handler and an interrupt service routine

Fetch-Decode-Execute Cycle (1)

- The fetch-decode-execute cycle is the series of steps that a computer carries out when it runs a program
- We first have to fetch an instruction from memory, and place it into the IR
- Once in the IR, it is decoded to determine what needs to be done next
- If a memory value (operand) is involved in the operation, it is retrieved and placed into the MBR
- With everything in place, the instruction is executed

Fetch-Decode-Execute Cycle (2)



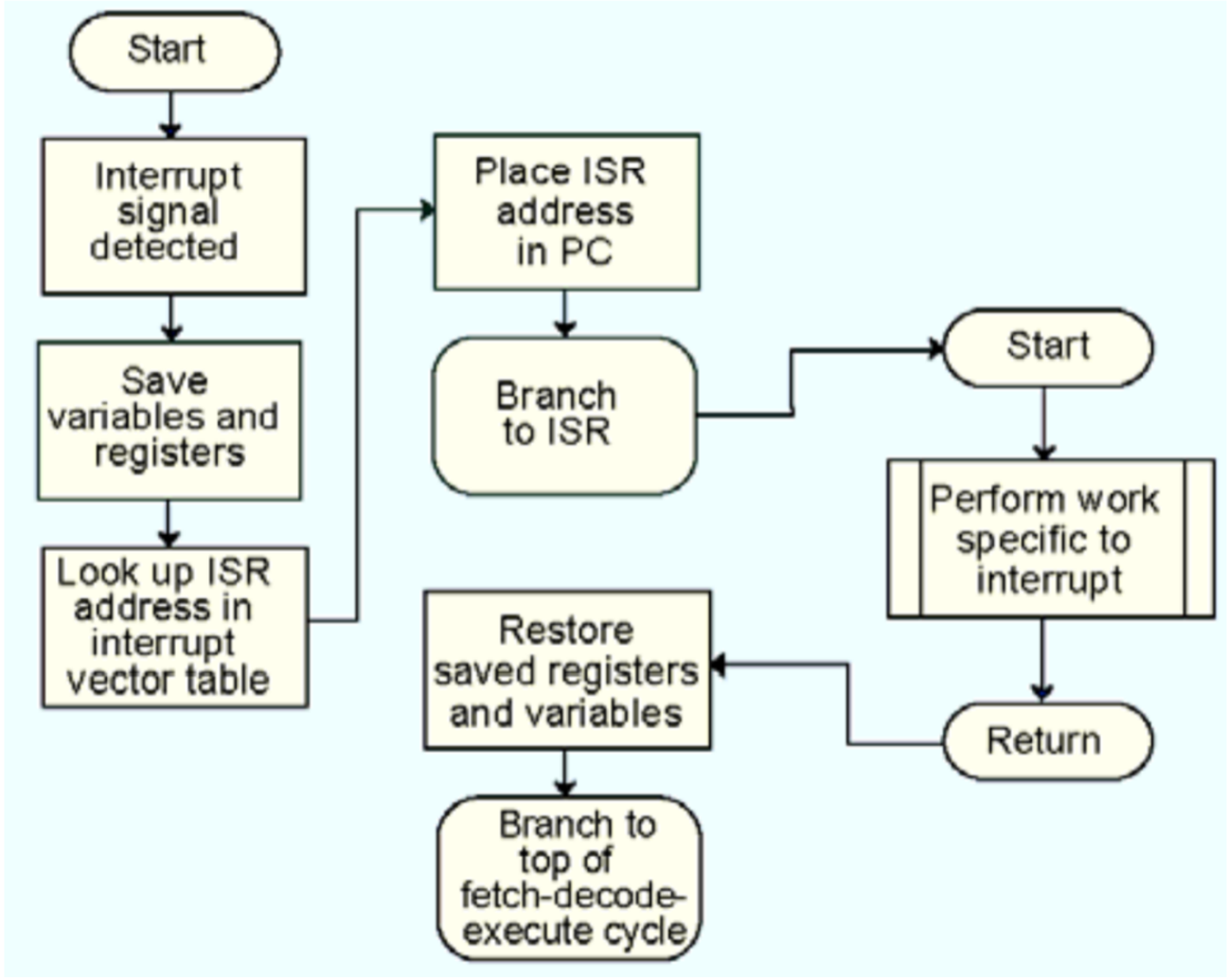
Traps

- A trap is a software procedure that is automatically invoked by the processor hardware due to some abnormal condition
  - Caused as part of an instruction execution, hence it is synchronous with the processor's operation
  - Examples are illegal instruction, divide by 0, overflow, array bounds violation, etc.
- Reduce the need for extensive software-only checking in applications and the operating system
- Traps may be used to trap "illegal" instructions that can then be executed in software
  - Used for floating point instructions on a processor without floating point hardware

Interrupts

- An interrupt is a software procedure that is automatically invoked by the processor hardware due to some hardware exception
  - Caused by hardware, perhaps external, hence it is asynchronous with the processor's operation
  - Causes a branch to an interrupt service routine (ISR)
- Interrupts occur when:
  - A user break (e.g., Control+C) is issued
  - I/O is requested by the user or a program
  - A critical error occurs

Interrupt Processing



CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Describe the fetch-execute-decode cycle
- Describe the difference between a trap and an interrupt

If you have any difficulties, please review the lecture video before continuing.

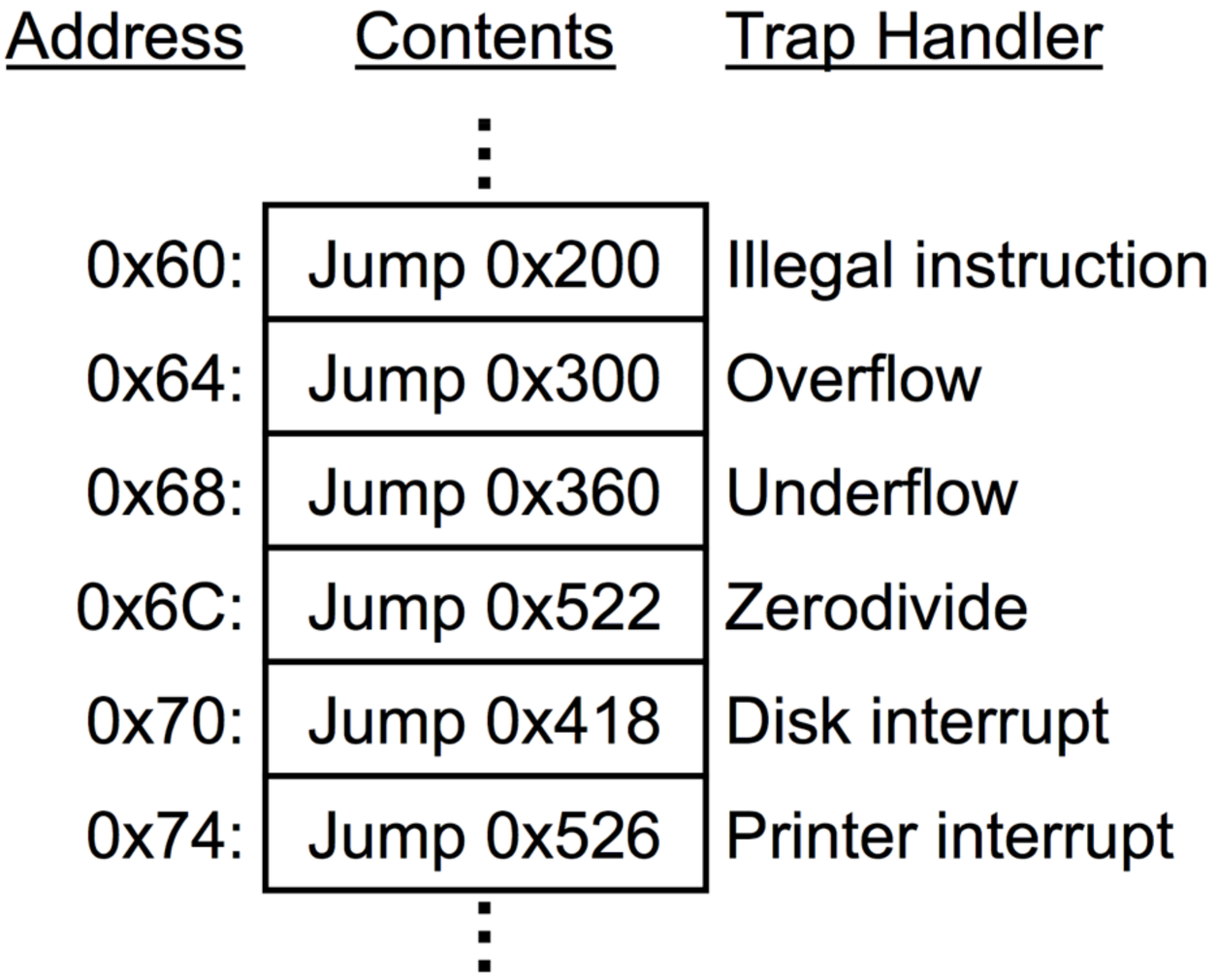
Maskable and Non-maskable Interrupts

- For general-purpose systems, it is common to disable all interrupts during the time in which an interrupt is being processed
    - Typically, this is achieved by setting a bit in the flags register
- Interrupts that are ignored in this case are called maskable
- Non-maskable interrupts are those interrupts that must be processed in order to keep the system in a stable condition

Branch Table (1)

- A branch table is a table that allows quick access to trap and interrupt service routines
- Entries may be:
  - One-word jump instructions to the handler
  - One-word addresses of the handler
- Indexed by vector number that corresponds to the type of trap or that is provided by a device that generates an interrupt

Branch Table (2)



Trap Handlers, ISRs

- Trap or interrupt causes processor to:
  - Save return address (PC)
  - Save program status register
- Handlers and service routines:
  - Save any additional registers that will be altered, usually on the stack
  - Performs task that is specific to the trap or interrupts
  - Restores registers that were saved
  - Returns

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Describe how an interrupt is processed
- Enumerate the steps that must be taken by a trap handler and an interrupt service routine

If you have any difficulties, please review the lecture video before continuing.

Summary

- The fetch-decode-execute cycle is the series of steps that a computer carries out when it runs a program
- A trap is a software procedure that is automatically invoked by the processor hardware due to some abnormal condition
- An interrupt is a software procedure that is automatically invoked by the processor hardware due to some hardware exception
- A branch table is a table that allows quick access to trap and interrupt service routines
- A trap handler and an interrupt service routine save the PC and processor status information before branching to a specified location in memory

MODULE 4: Computer Organization and MARIE

Lecture 4.3
Instruction Processing

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech