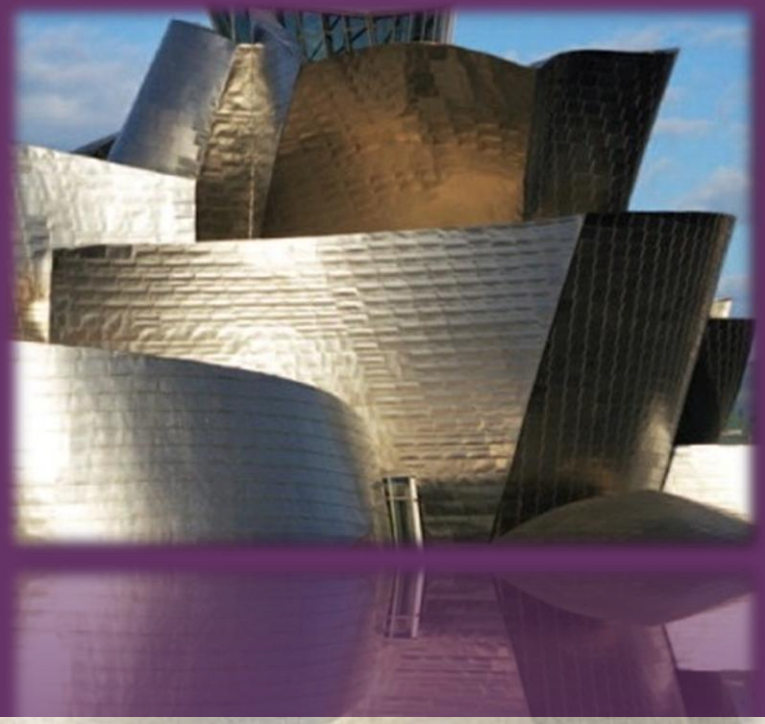
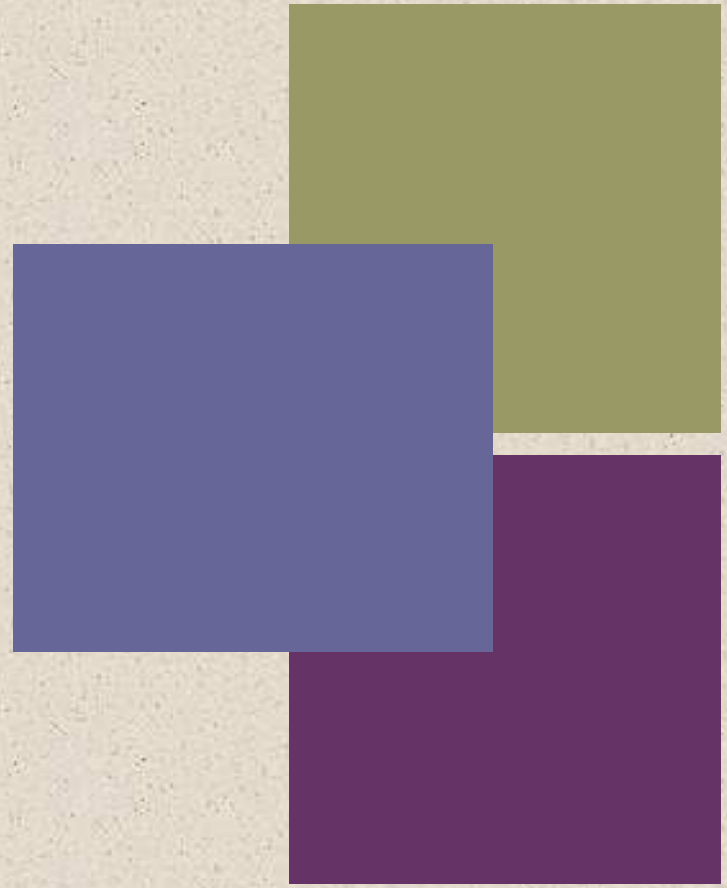


# William Stallings Computer Organization and Architecture 9th Edition



# Chapter 3

## A Top-Level View of Computer Function and Interconnection

# Classes of Interrupts

| | |
|---|---|
| **Program** | Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space. |
| **Timer** | Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis. |
| **I/O** | Generated by an I/O controller, to signal normal completion of an operation, request service from the processor, or to signal a variety of error conditions. |
| **Hardware failure** | Generated by a failure such as power failure or memory parity error. |

# Program Flow Control

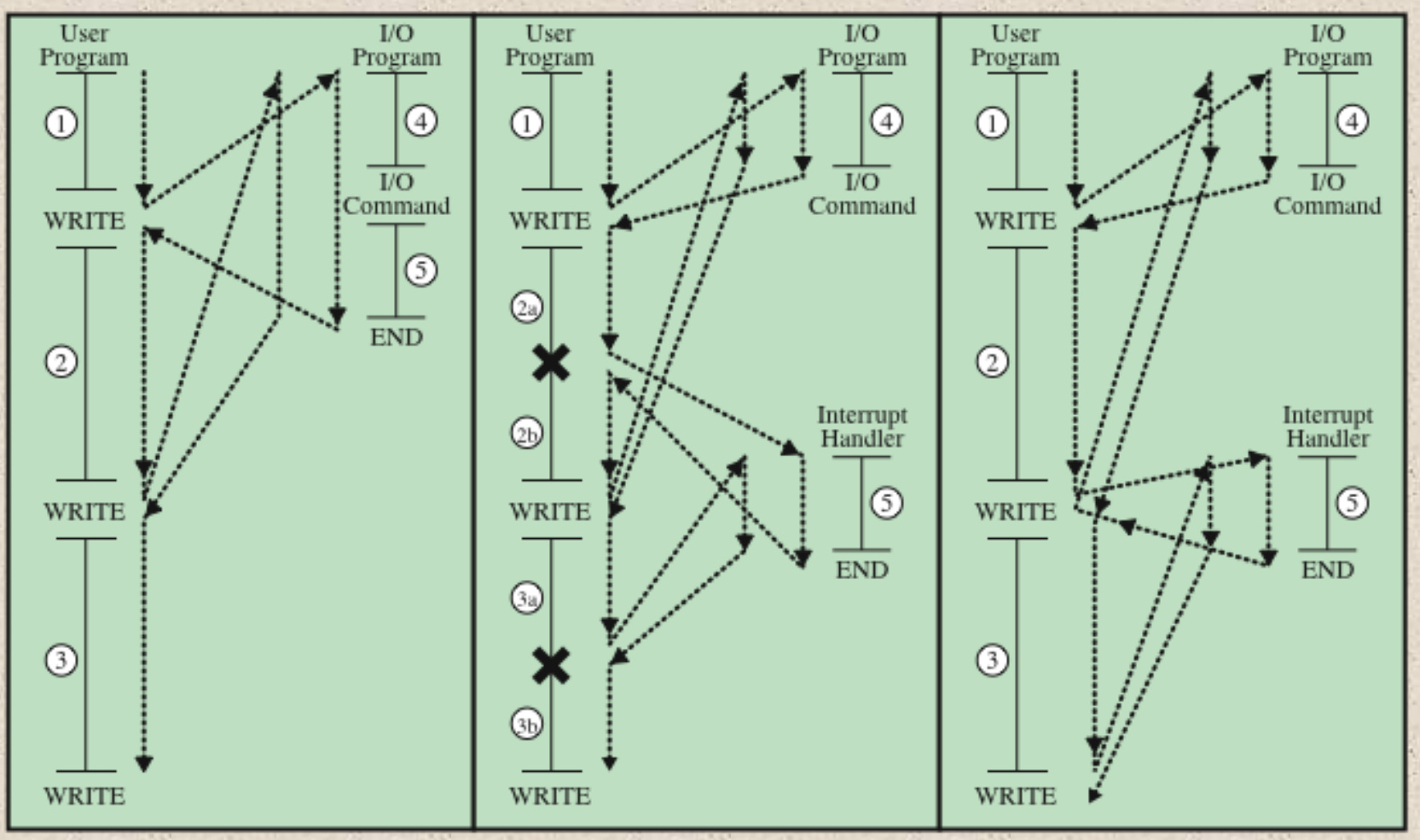



**Figure 3.7 Program Flow of Control Without and With Interrupts**

# Transfer of Control via Interrupts

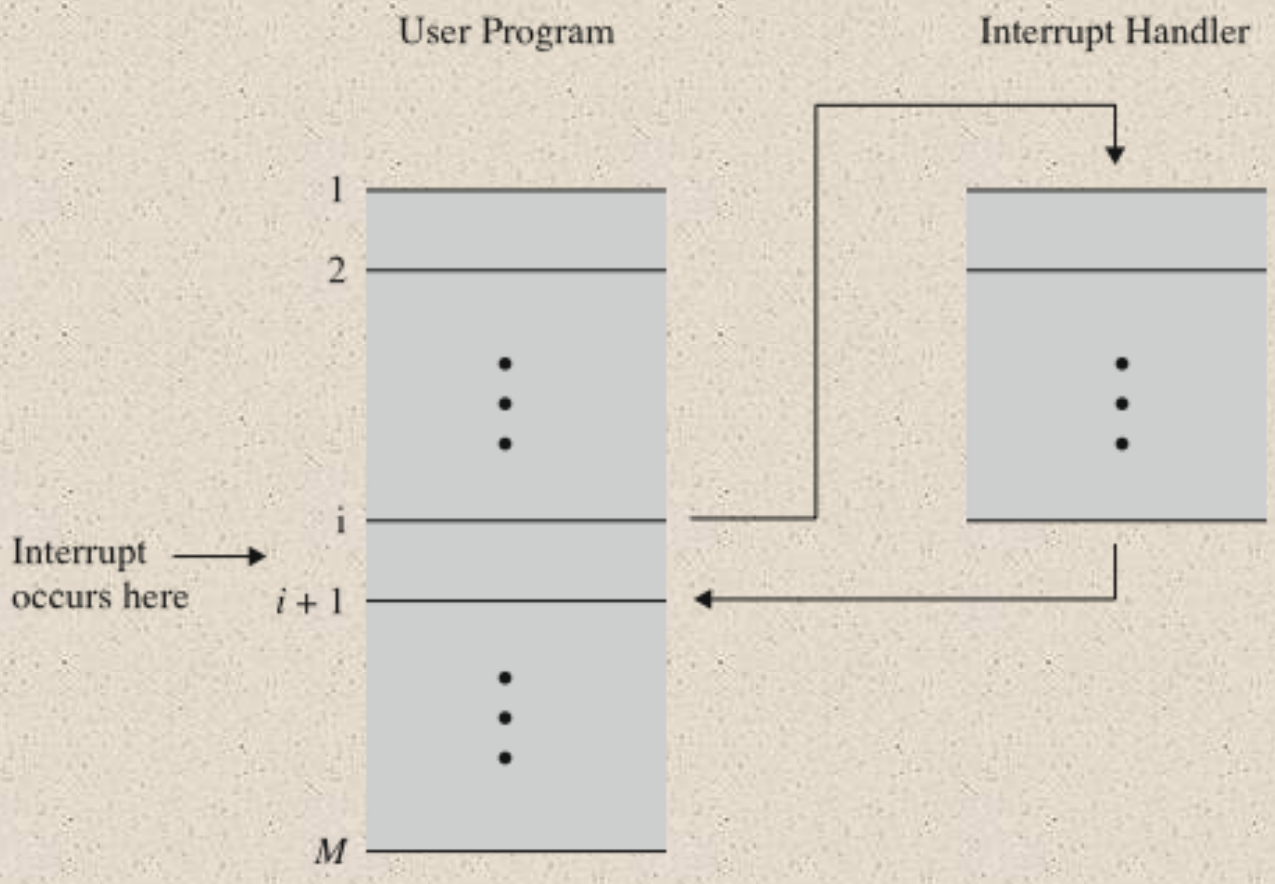



**Figure 3.8 Transfer of Control via Interrupts**

# Instruction Cycle With Interrupts

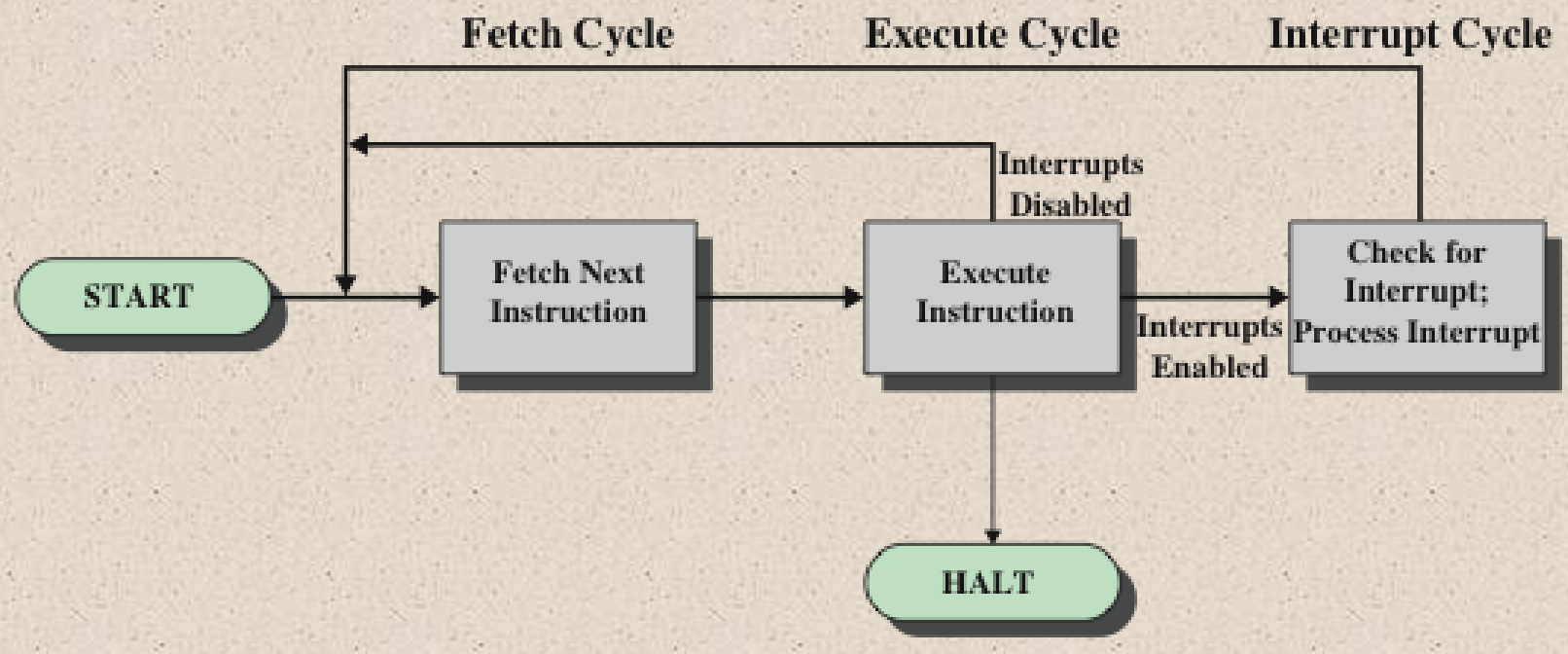



Figure 3.9 Instruction Cycle with Interrupts

# Program Timing: Short I/O Wait

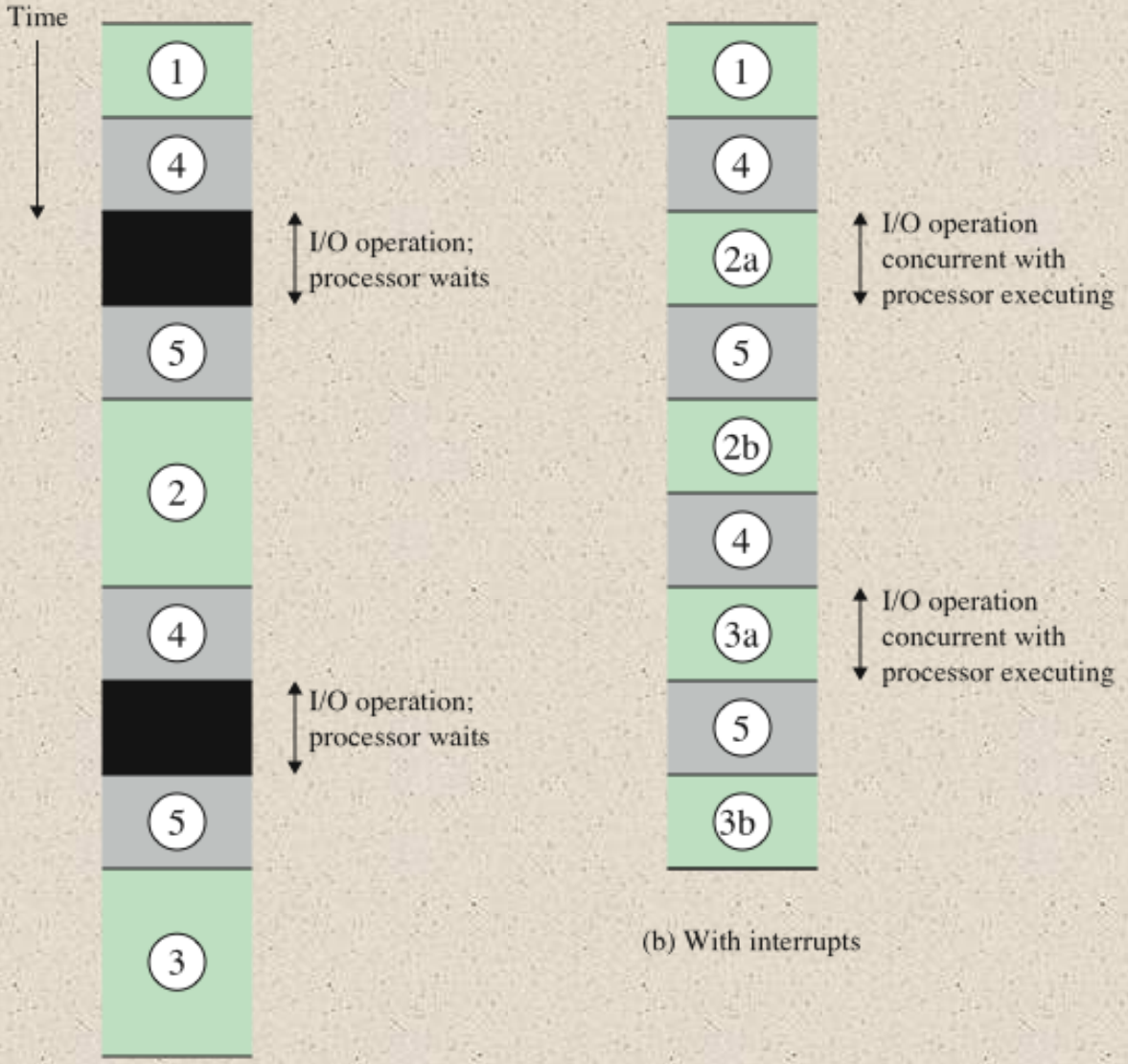
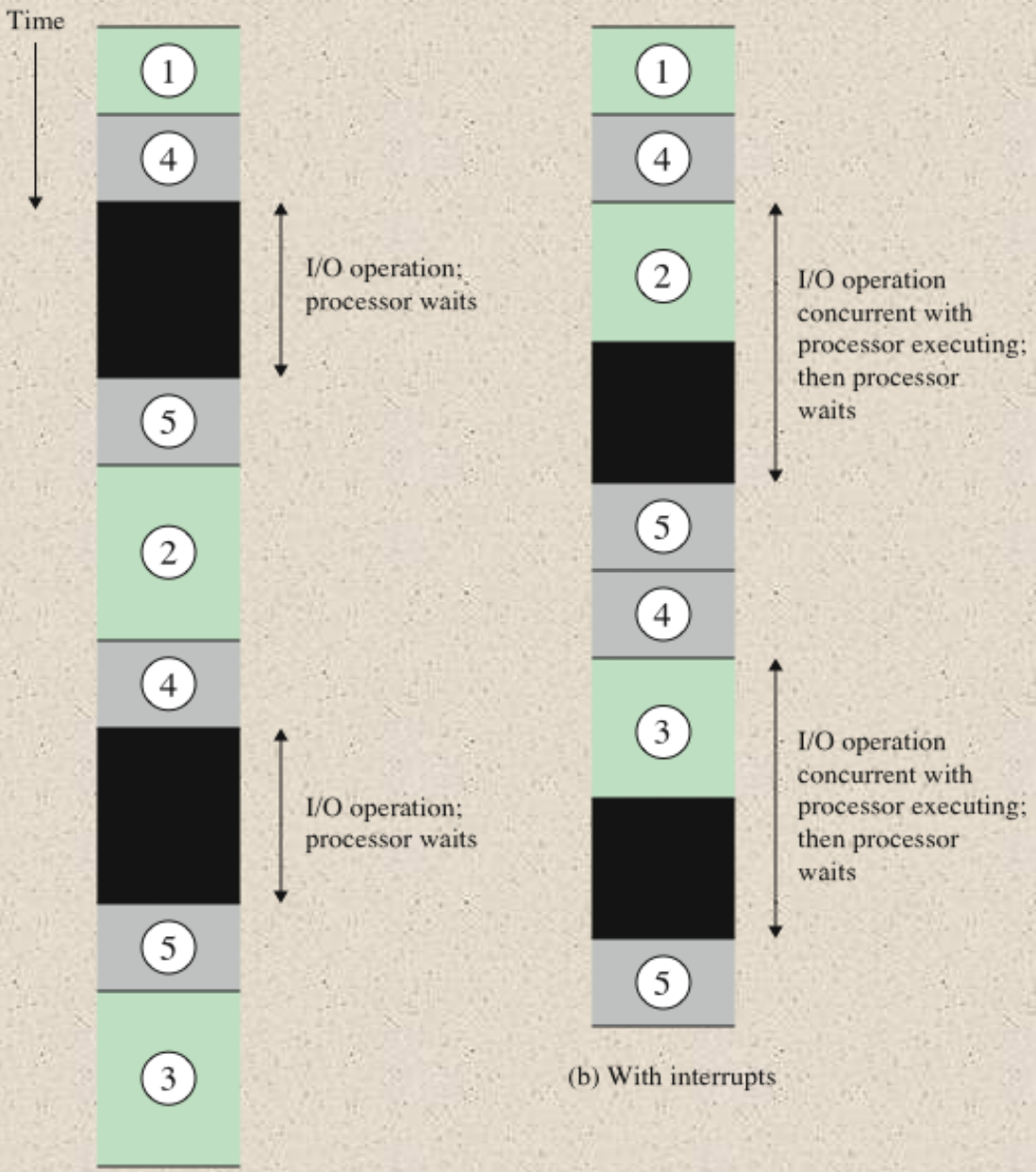



**Figure 3.10 Program Timing: Short I/O Wait**




# Program Timing: Long I/O Wait

**Figure 3.11 Program Timing: Long I/O Wait**

# Instruction Cycle State Diagram With Interrupts

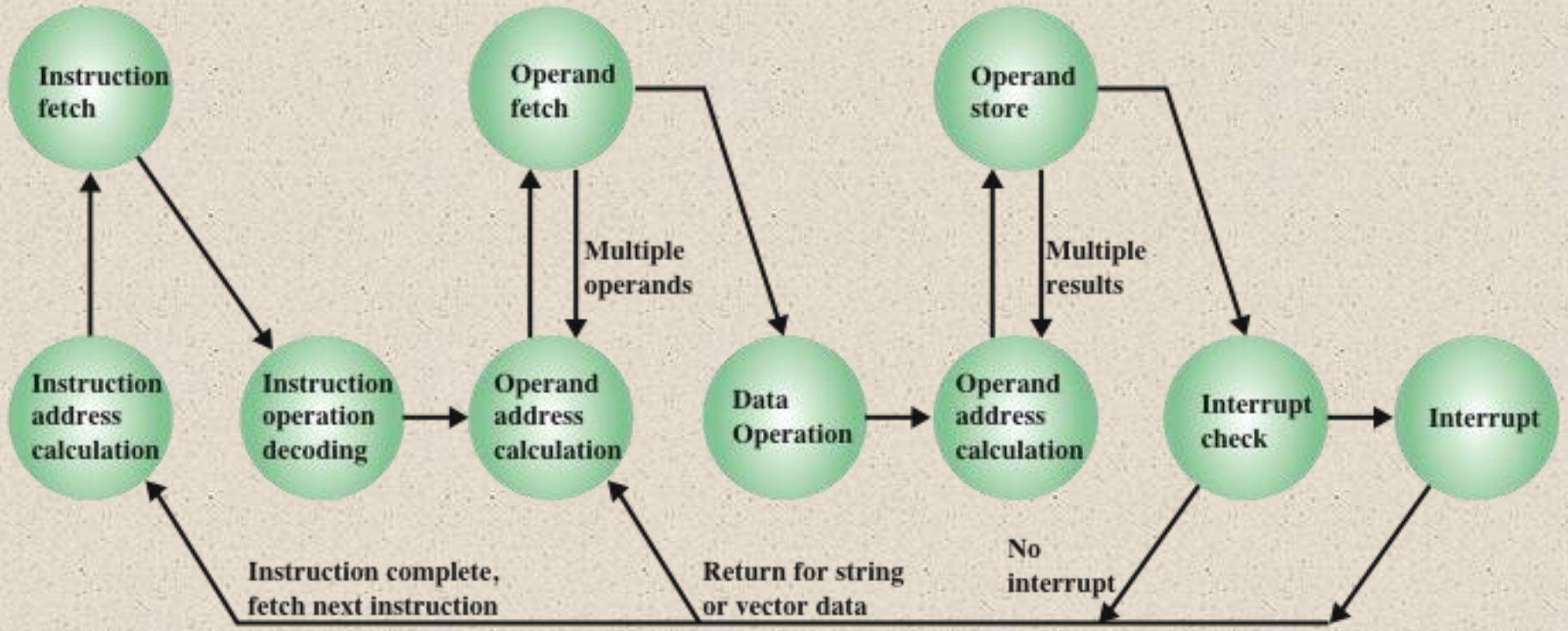
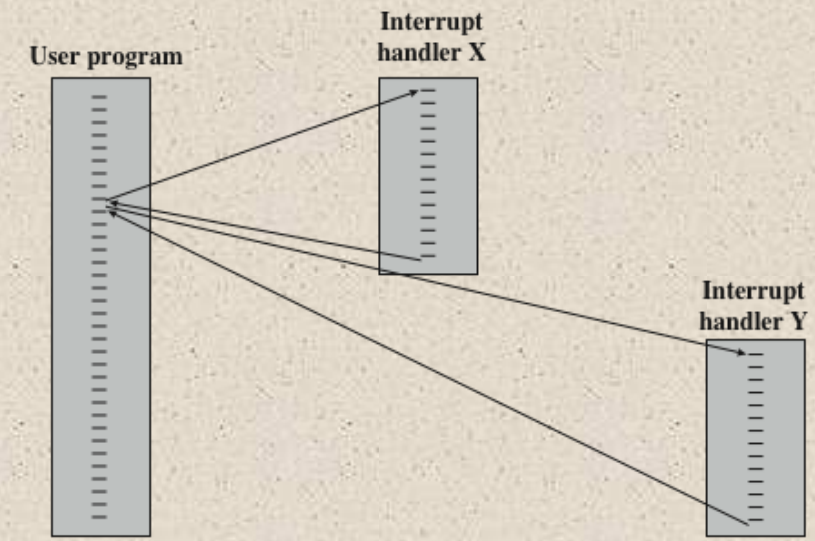


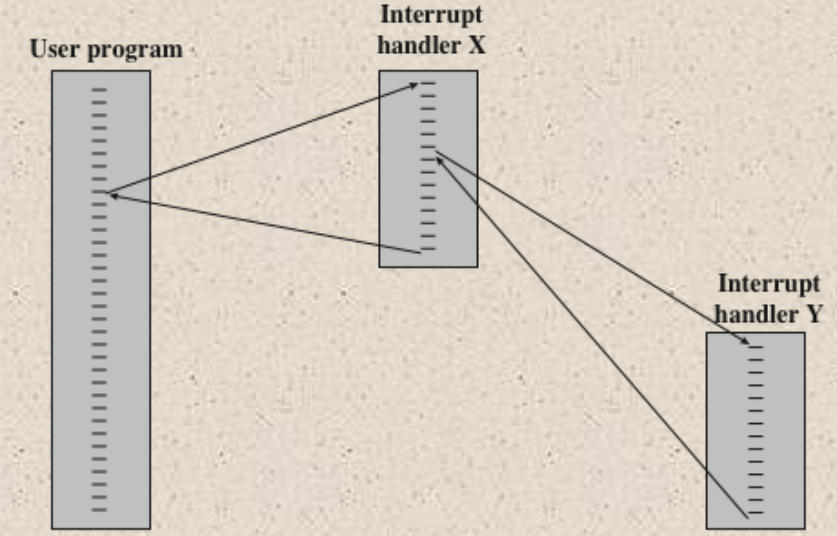

Figure 3.12 Instruction Cycle State Diagram, With Interrupts

# Transfer of Control

# Multiple Interrupts




(a) Sequential interrupt processing




(b) Nested interrupt processing

Figure 3.13 Transfer of Control with Multiple Interrupts

# Time Sequence of Multiple Interrupts

Example

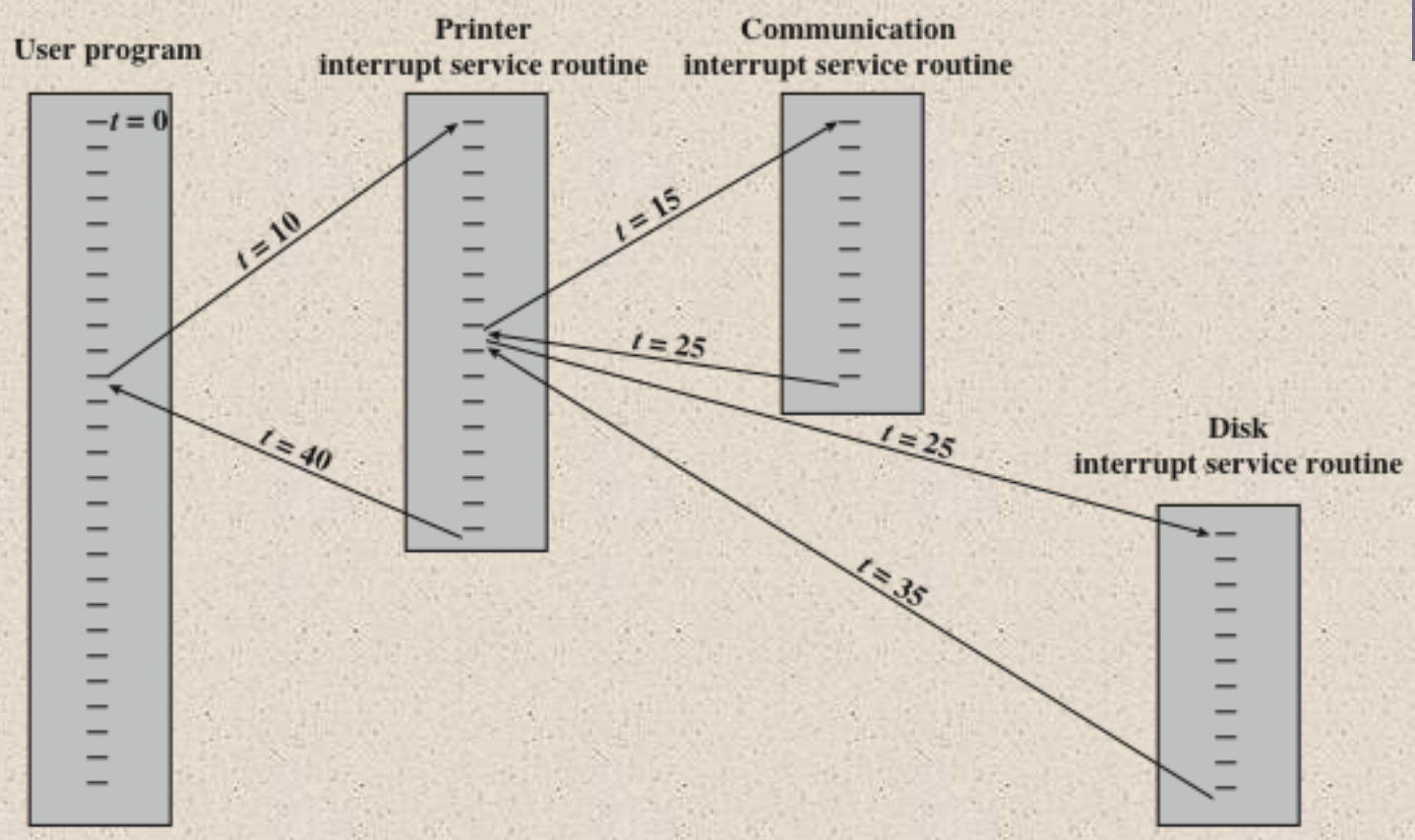



**Figure 3.14 Example Time Sequence of Multiple Interrupts**