Computer System Overview

CHAPTER 1

Basic Elements

- Processor
- Main Memory
  - volatile
- I/O modules
  - secondary memory devices (disks)
  - communications equipment
  - terminals
- System bus
  - communication among processors, memory, and I/O modules

Computer Component : Top-Level View

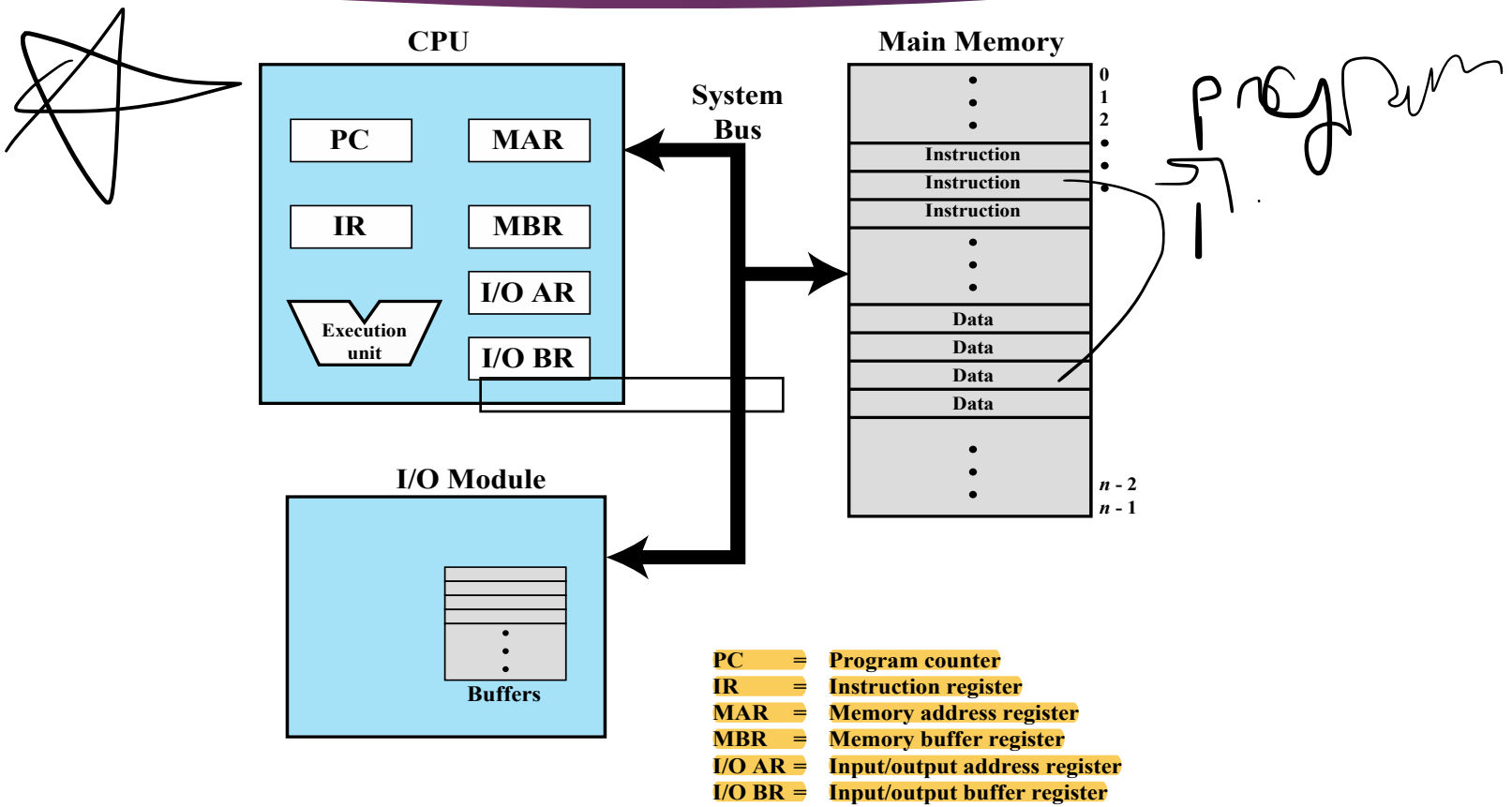


Figure 1.1 Computer Components: Top-Level View

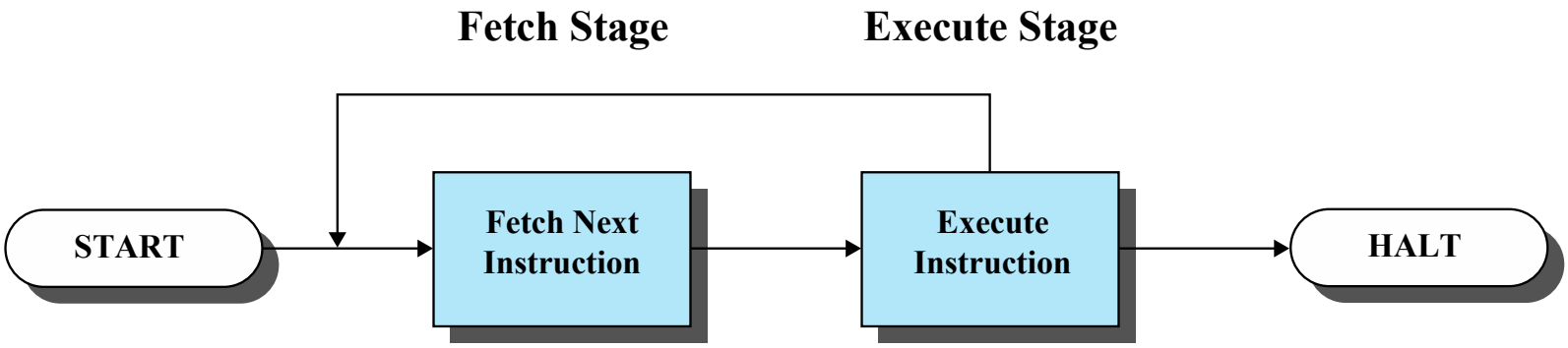
Instructions

- A program : a set of instructions and data

- Instructions
  - Processor-memory data transfer
  - Processor-I/O data transfer
  - Data processing
  - Control

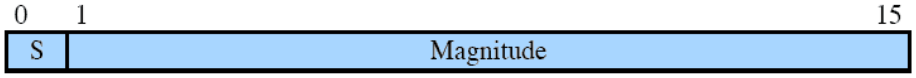
Instruction Execution

- Two steps
  - Processor reads instructions from memory
  - Processor executes each instruction
- Basic Instruction Cycle :



Example of Program Execution

(a) Instruction format



(b) Integer format

Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

(d) Partial list of opcodes

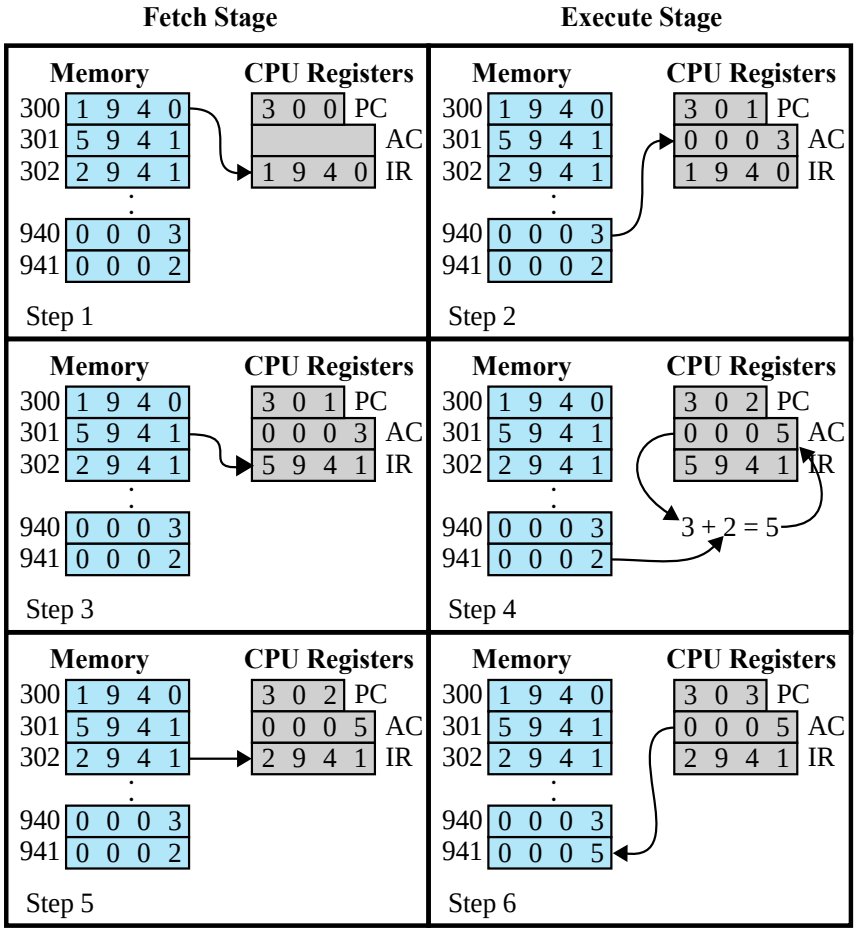
Figure 1.3 Characteristics of a Hypothetical Machine

16bit

Control and Status Registers

- Program Counter (PC)
  - Contains the address of an instruction to be fetched
- Instruction Register (IR)
  - Contains the instruction most recently fetched
- Program Status Word (PSW)
  - Condition codes
  - Interrupt enable/disable
  - Supervisor/user mode

Example of Program Execution (2)



Fetch Stage

- 첫 번째 fetch stage 시작 전 PC : 300, IR : 이전 실행 명령 주소

- Fetch stage
  - (1) MAR <- PC, PC++
  - (2) memory read, MBR <- (MAR에 저장된 번지의 명령)
  - (3) IR <- MBR

Execution Stage

- Execution stage #1
  - (1) 명령 분석
  - (2) MAR <- IR의 주소 부분
  - (3) memory read, MBR <- (MAR에 저장된 번지의 데이터)
  - (4) AC <- MBR

- Execution stage #2
  - (1) 명령 분석
  - (2) MAR <- IR의 주소 부분
  - (3) memory read, MBR <- (MAR에 저장된 번지의 데이터)
  - (4) AC <- AC + MBR

Execution Stage (2)

- Execution stage #3
  - (1) 명령 분석
  - (2) MAR <- IR의 주소 부분, MBR <- AC
  - (3) memory write, (MAR에 저장된 번지의 memory 공간) <- MBR

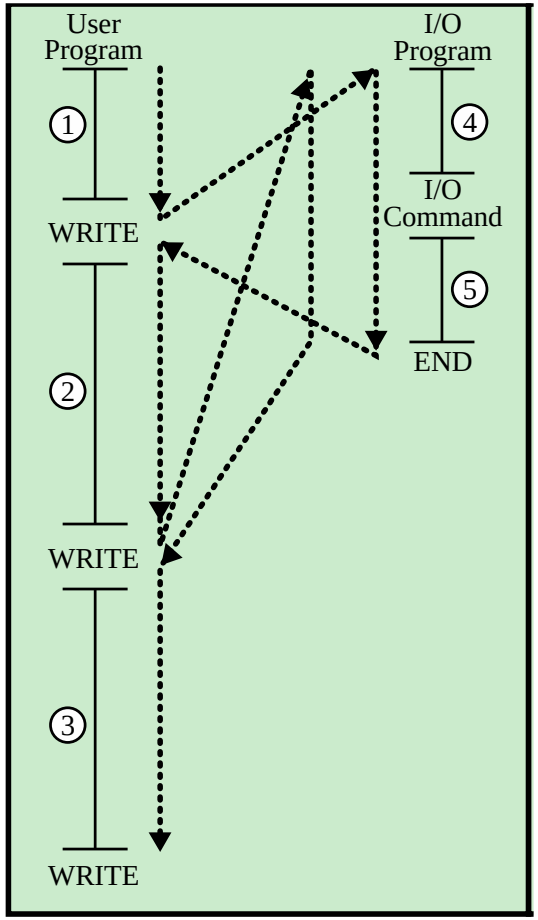
Interrupts

- Interrupt the normal sequencing of the processor
- Classes of Interrupts

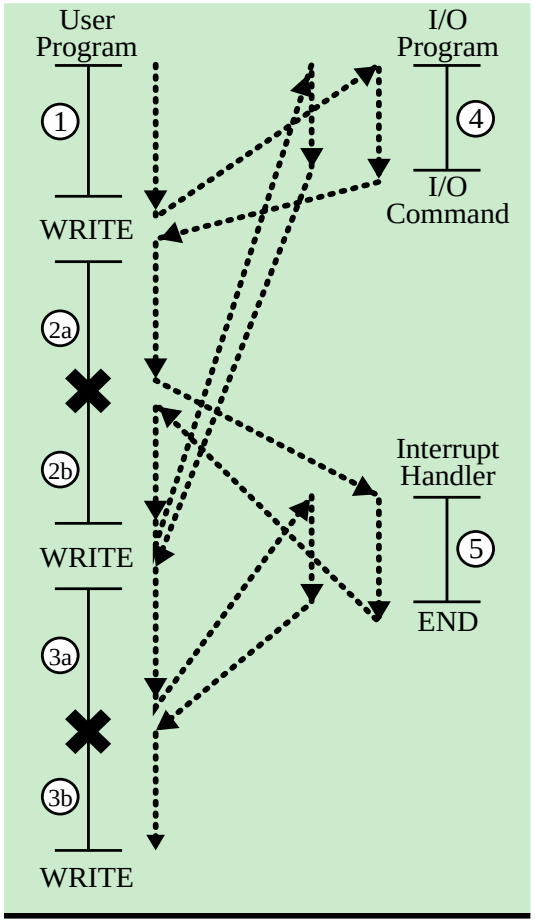
Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows 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 or to signal a variety of error conditions.
Hardware failure	Generated by a failure, such as power failure or memory parity error.

- Most I/O devices are slower than the processor
  - Processor must pause to wait for device

Program Flow of Control With or Without Interrupts (1)

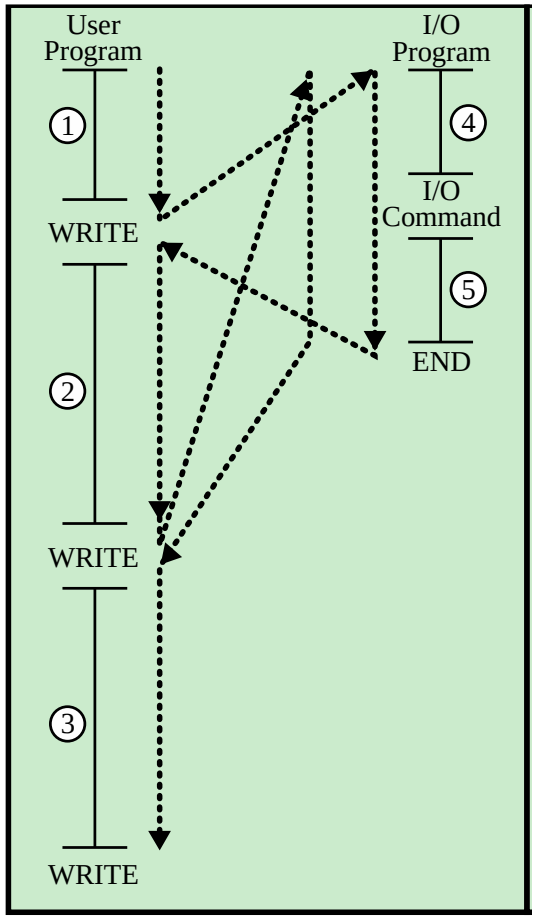


(a) No interrupts

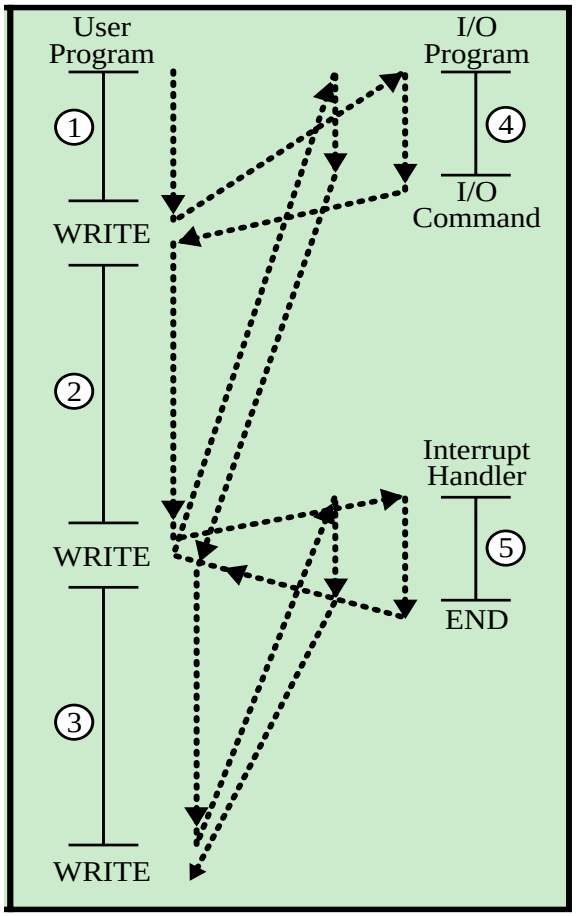


(b) Interrupts; short I/O wait

Program Flow of Control With or Without Interrupts (2)



(a) No interrupts



(c) Interrupts; long I/O wait

Interrupt Handler

- Program to service a particular I/O device
- Generally part of the operating system

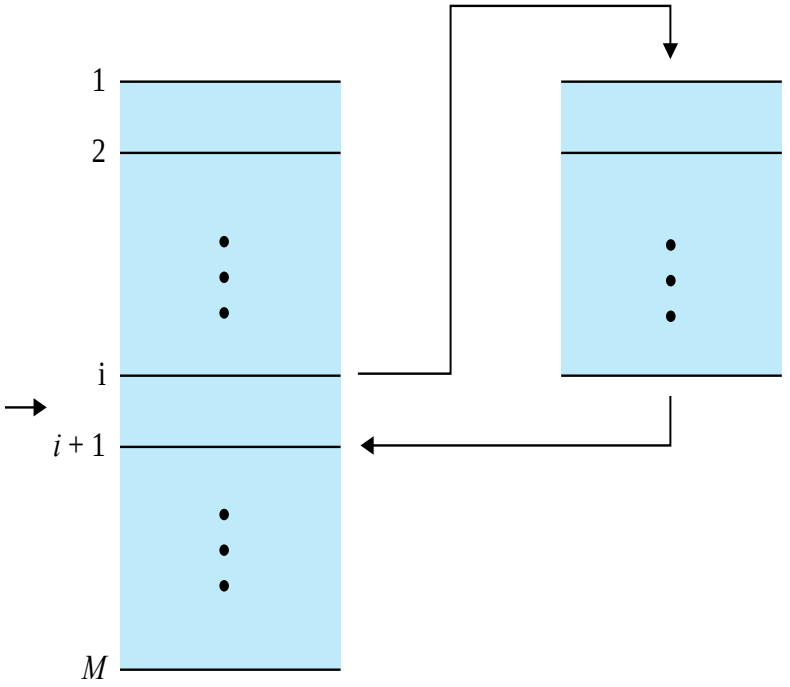


Figure 1.6 Transfer of Control via Interrupts

Instruction Cycle with Interrupts

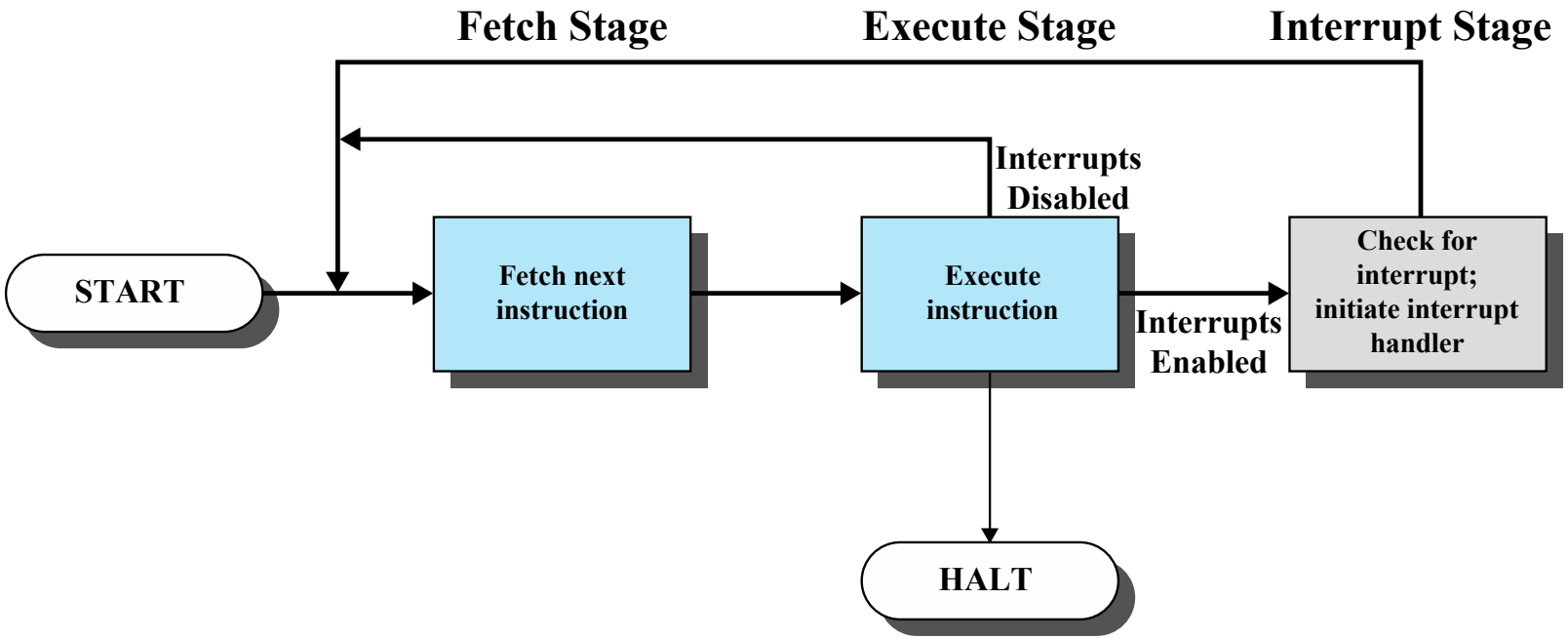


Figure 1.7 Instruction Cycle with Interrupts

Timing Diagram : Short I/O Wait

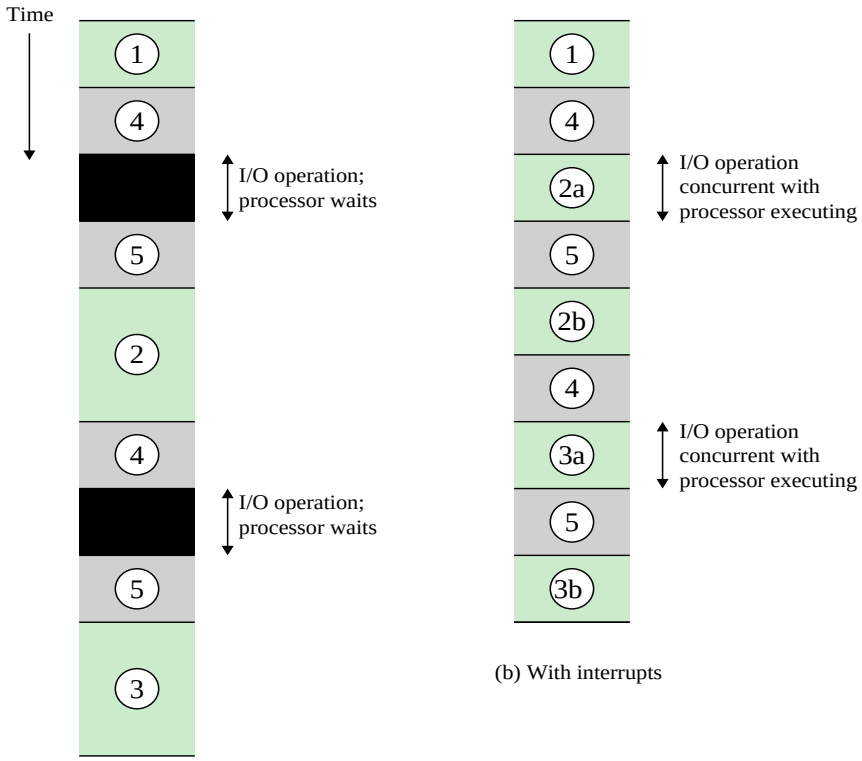


Figure 1.8 Program Timing: Short I/O Wait

Timing Diagram : Long I/O Wait

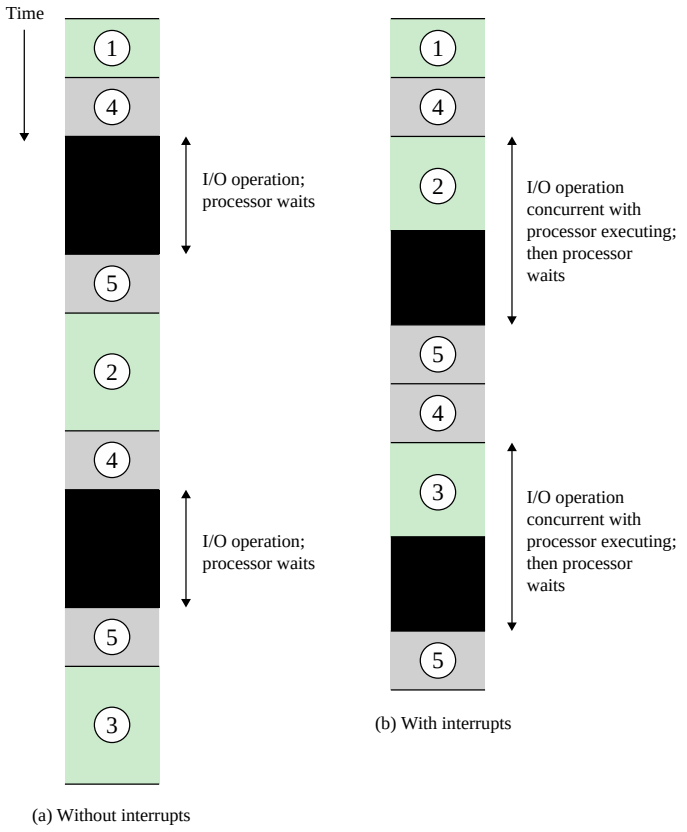
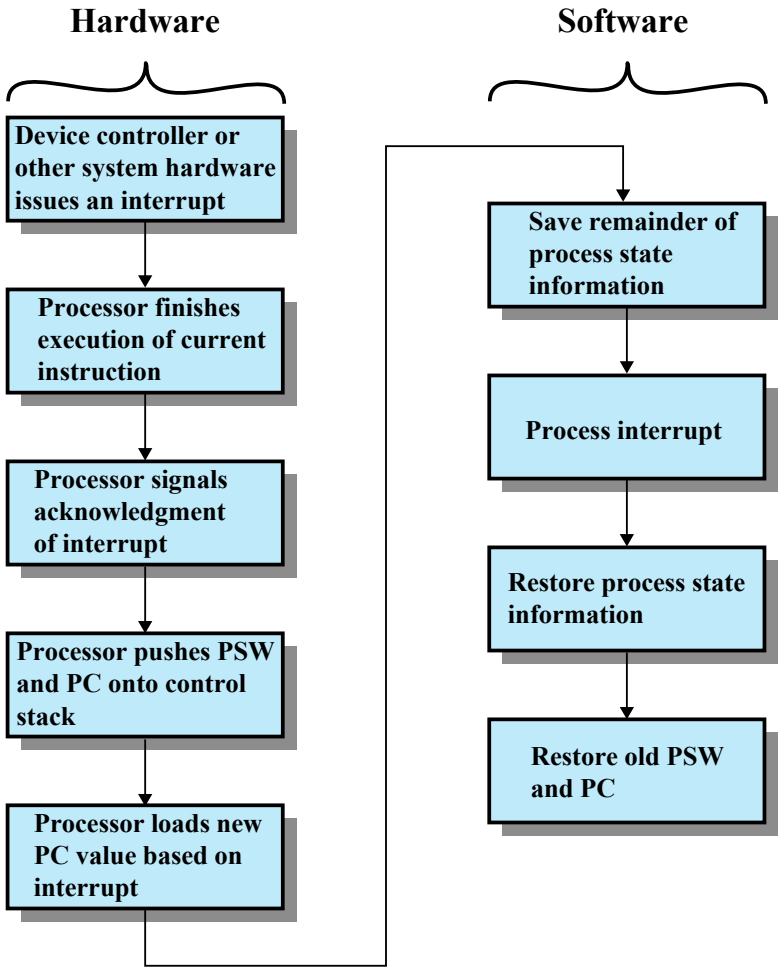


Figure 1.9 Program Timing: Long I/O Wait

Simple Interrupt Processing



Changes in Memory and Registers

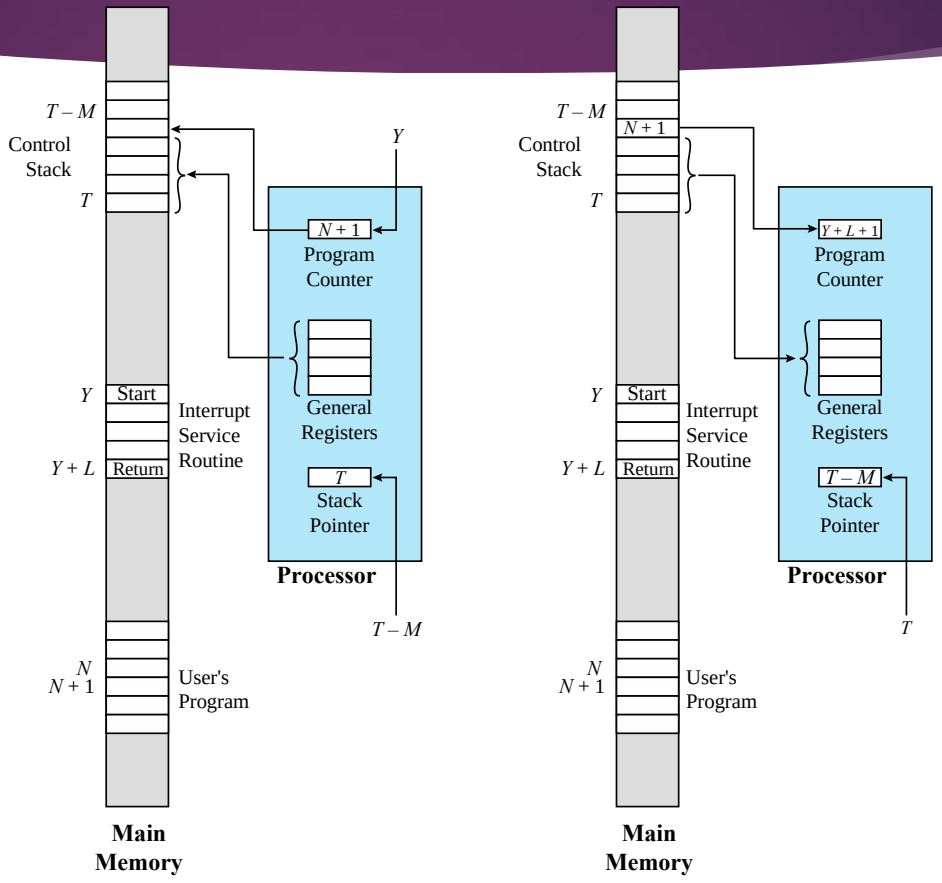


Figure 1.11 Changes in Memory and Registers for an Interrupt

Memory Hierarchy

- Major constraints in memory
  - amount (capacity)
  - Speed (access time)
  - Expense
- Going Down the Hierarchy
  - Decreasing cost per bit
  - Increasing capacity
  - Increasing access time
  - Decreasing frequency of access of the memory by the processor

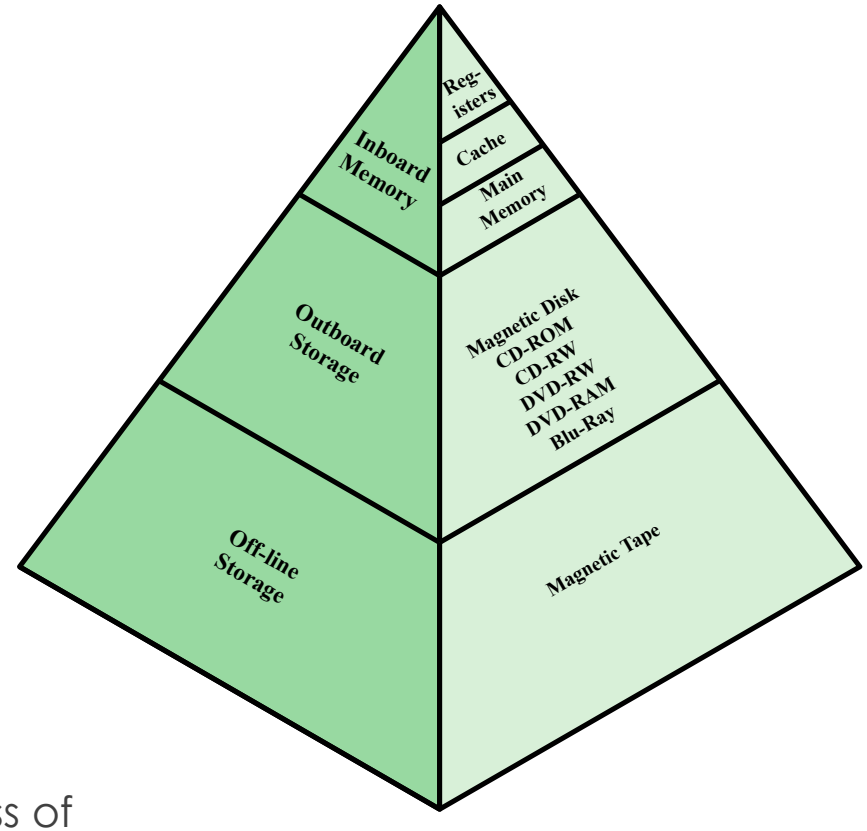


Figure 1.14 The Memory Hierarchy

Memory Hierarchy (2)

- The Key to the success of memory hierarchy is the decreasing frequency of access at lower levels.

- Locality of reference
  - Memory reference by the processor, for both instruction and data, tend to cluster

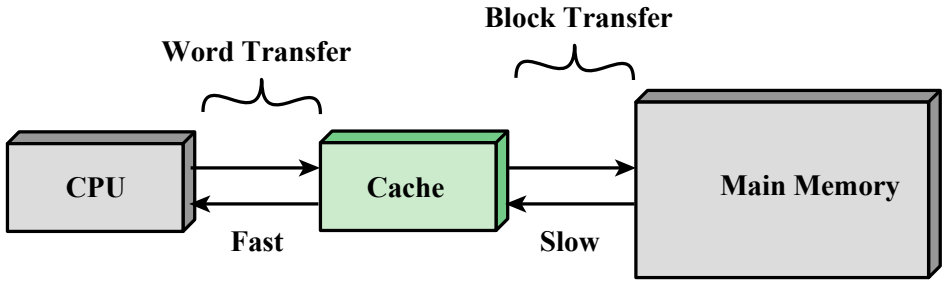
Cache Memory

- Processor speed is faster than memory speed
- A small, fast memory between the processor and main memory

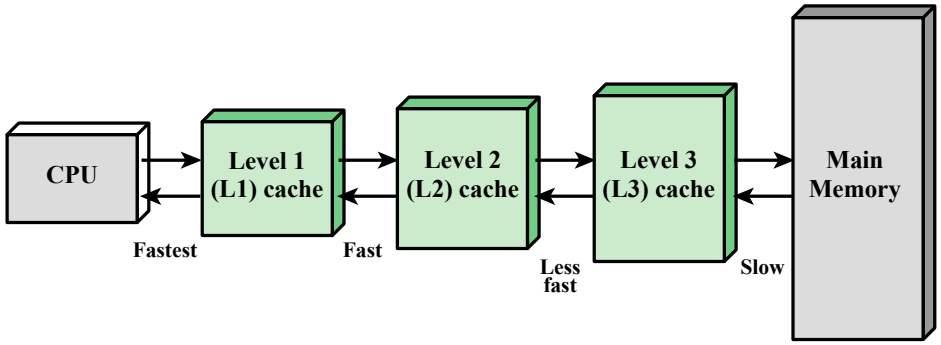
- Cache contains a copy of a portion of main memory
- Increase the speed of memory
- Exploit the principle of locality

- Invisible to operating system

Cache Memory (2)



(a) Single cache



(b) Three-level cache organization

Figure 1.16 Cache and Main Memory

I/O Techniques

Three techniques are possible for I/O operations:

- Programmed I/O
- Interrupt-Driven I/O
- Direct Memory Access (DMA)

Direct Memory Access

- Transfers the entire block of data directly to and from memory without going through the processor
  - processor is involved only at the beginning and end of the transfer
  - processor executes more slowly during a transfer when processor access to the bus is required
- More efficient than interrupt-driven or programmed I/O

Symmetric Multiprocessors (SMP)

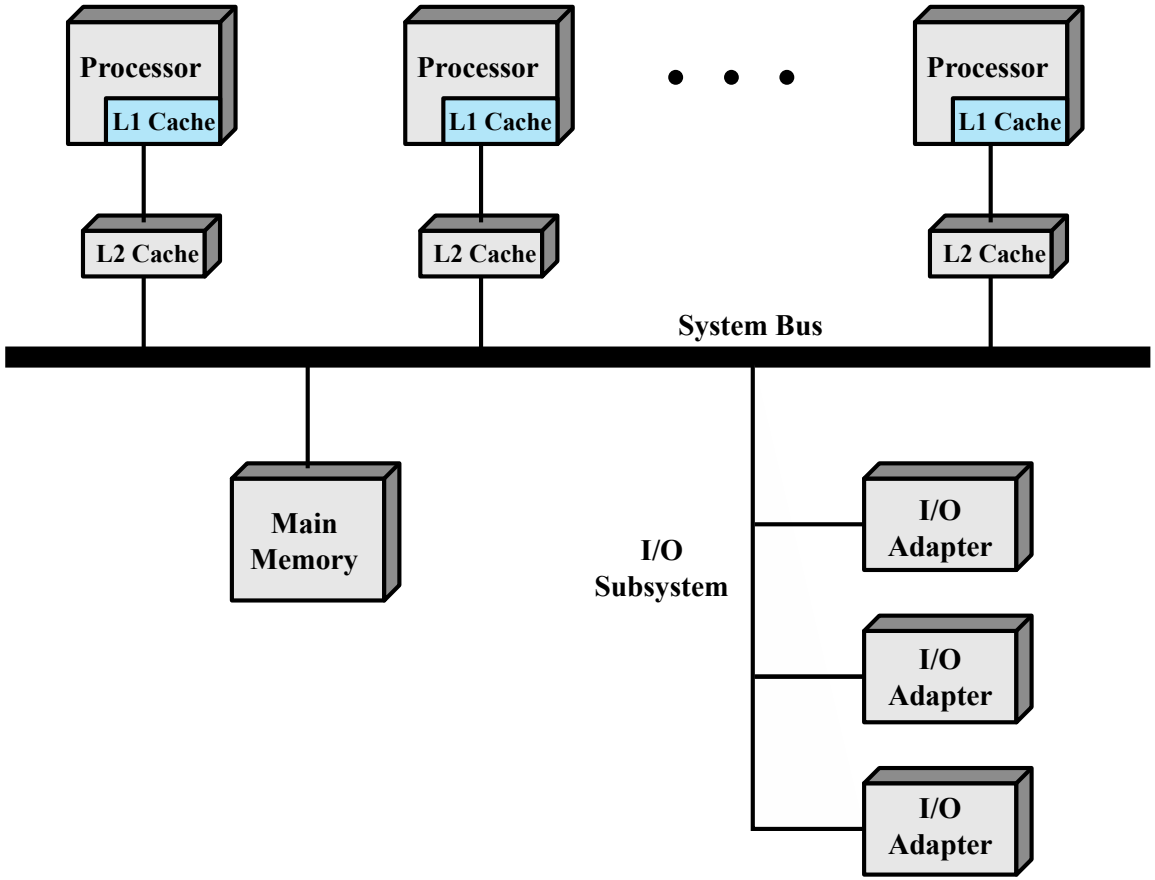


Figure 1.19 Symmetric Multiprocessor Organization

Multicore Computer

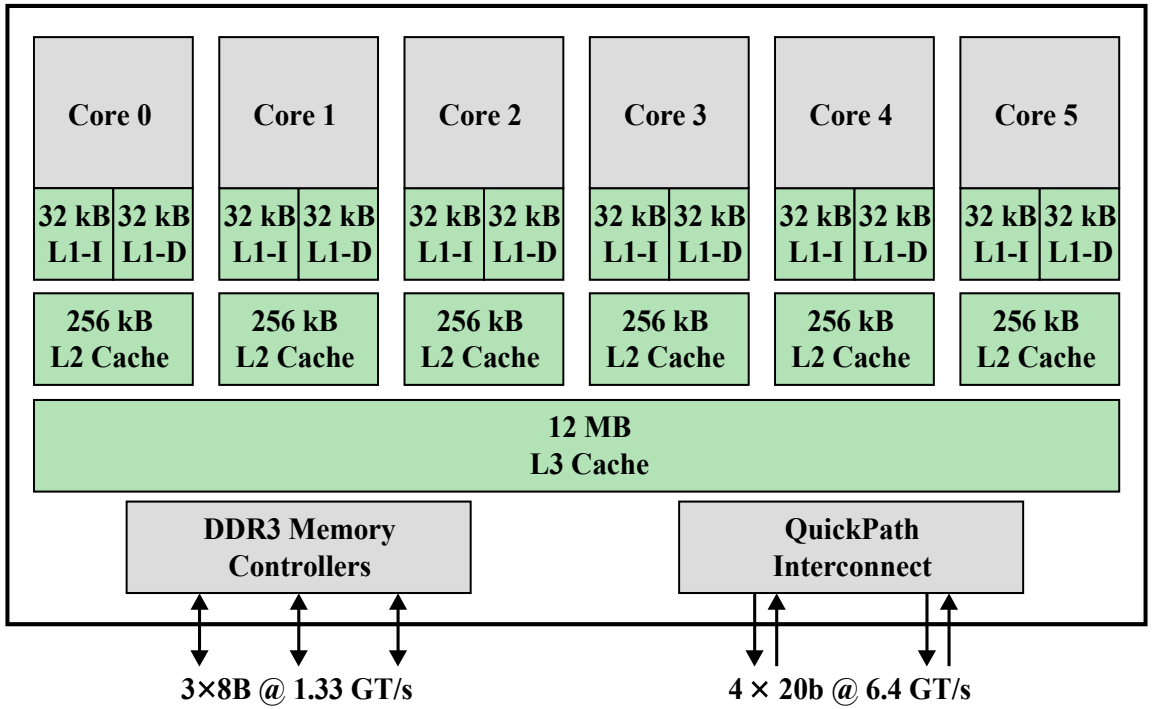


Figure 1.20 Intel Core i7-990X Block Diagram