# 操作系统

# 第1章 计算机系统概述
# Computer System Overview

**孙承杰**
**哈工大计算学部**

**E-mail： sunchengjie@hit.edu.cn**
**2025年秋季学期**

# Learning Objectives

- Describe the basic elements of a computer system and their interrelationship.
- Explain the steps taken by a processor to execute an instruction.
- Understand the concept of interrupts, and how and why a processor uses interrupts.
- List and describe the levels of a typical computer memory hierarchy.

# Outline

- Basic Elements
- Instruction Execution
- Interrupt/Interrupt Processing
- Memory Hierarchy
- I/O Techniques
- Muitiprocessor/multicore

# Basic Elements

Processor

I/O Modules

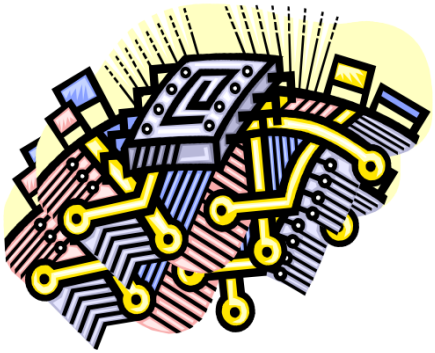
Main Memory

System Bus

# Processor

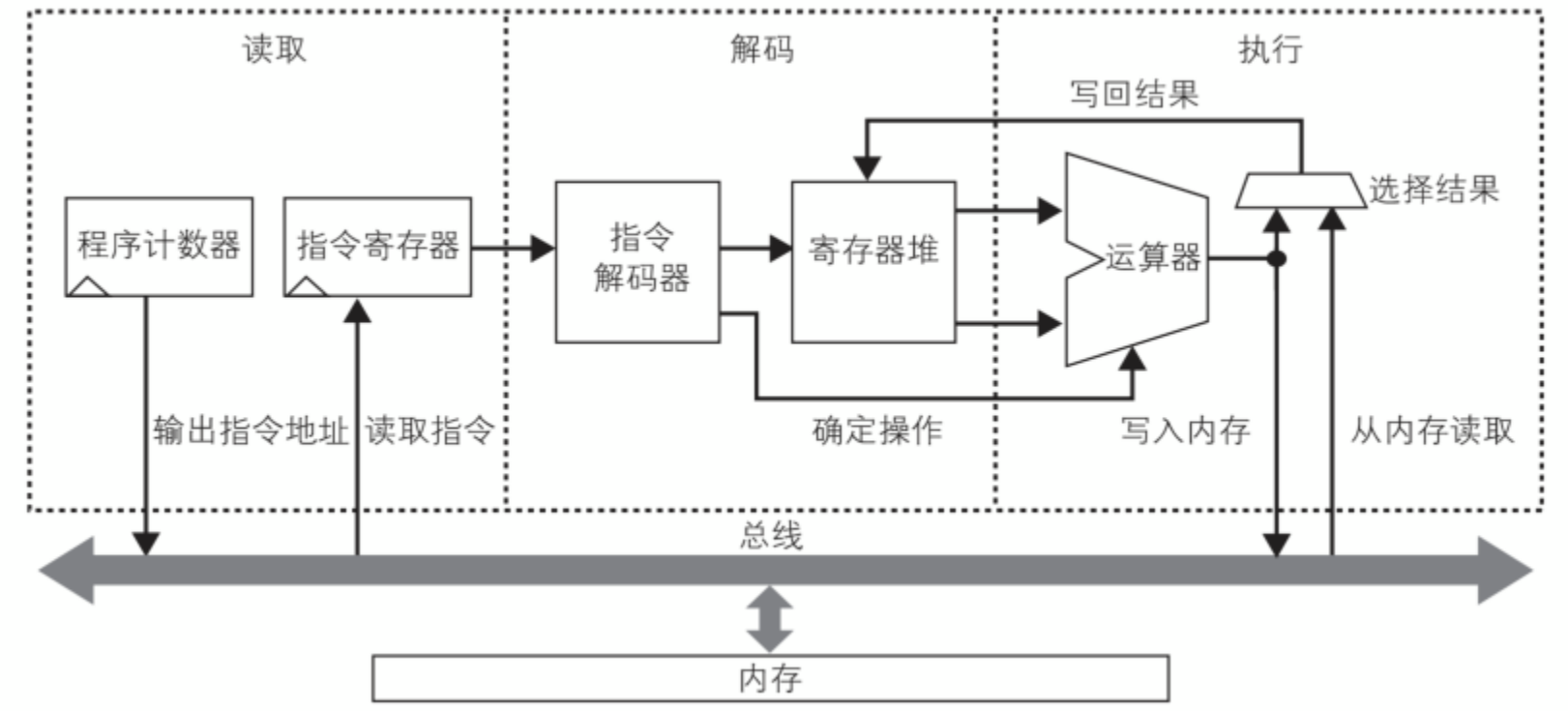
Controls the operation of the computer

Performs the data processing functions



Referred to as the *Central Processing Unit* (CPU)

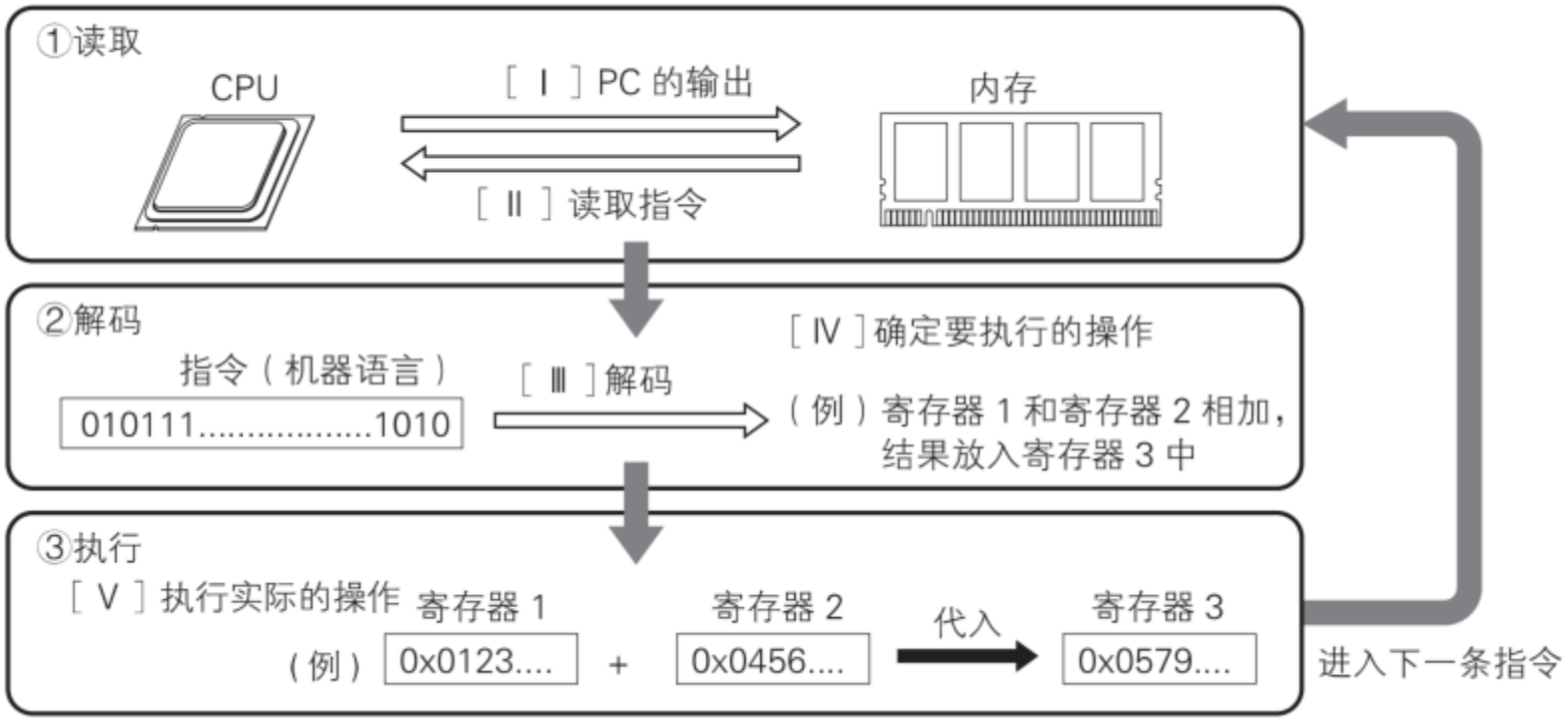
# CPU




CPU的内部构造

图片来自《CPU自制入门》

# CPU




CPU的处理流程

图片来自《CPU自制入门》

# CPU

**Processor Families:** 4004 8008 8080 8086 8088 80286 80386 DX 80486 DX 80486 DX2 80486 DX4 Atom Celeron Core 2 Duo Core 2 Quad Core 2 Quad Extreme Core Duo Core i3 Core i5 Core i7 Itanium Itanium 2 Pentium Pentium 4 Pentium 4 EE Pentium EE Pentium III Pentium M Pentium MMX Pentium Pro Xeon 80186 iAPX432 i860 i960 Core M Core2 Pentium II Pentium II Mobile http://cpudb.stanford.edu/manufacturers/9/processor_families/179 Core i7 Extreme Core 2 Extreme Xeon Phi Coprocessor Core Pentium D Core m5 Core m3 Core m7

**Microarchitectures:** Core:Merom Core:Conroe Core:Kentsfield Core:Penryn Core:Yorkfield Core:Wolfdale Madison Mckinley Merced Montecito Montvale Nehalem NetBurst:Willamette NetBurst:Northwood NetBurst:Prescott NetBurst:Cedar Mill NetBurst:Smithfield NetBurst:Presler P5 P6:Pentium II P6:Pentium III P6:Banias P6:Yonah Sandy Bridge 80386 DX 80486 DX Bonnel

**Code Names:** Pineview Tunnel Creek Stellarton Diamondville Silverthorne Conroe-L Mendocino Covington Clarkdale Allendale Conroe Wolfdale-3M Wolfdale Penryn-3M Merom Penryn Yorkfield Kentsfield Yonah Arrandale Lynnfield Clarksfield Bloomfield Gulftown Sandy Bridge Merced Mckinley P5 Prescott Cedar Mill Northwood Prescott 2M Presler Coppermine Tualatin Katmai Dothan Banias P55C N/A Woodcrest Clovertown Harpertown Nehalem-EP Westmere-EP Jasper Forest Cherry Trail Broadwell Ivy Bridge Bay Trail Irwindale Nehalem EX Lincroft Skylake Nehalem EP Haswell Dixon Avoton Haswell E Crystal Well Sandy Bridge-EN Sandy Bridge EN Braswell Cloverview Ivy Bridge EN Ivy Bridge EP Tukwila Montvale Knights Corner Moorefield Cedarview Nocona Potomac Cranford Cedarmill Paxville Dempsey Foster Prestonia Tulsa Gallatin Willamette Smithfield Tanner Cascades Madison Montecito Westmere EP Ivy Bridge E Broadwell E Dunnington Westmere EX Penwell Merrifield Gladden Sossaman Sandy Bridge E SoFIA LTE SoFIA 3G R SoFIA 3G Centerton Briarwood Poulson Tigerton Devil's Canyon Sandy Bridge EP Sandy Bridge-EP

**Processor Families:** 80386 80486 Athlon Athlon 64 FX Athlon 64 X2 Athlon II Athlon MP Athlon X2 Athlon XP K5 K6 K6-2 K6-III Opteron Phenom II Phenom Turion 64 Turion 64 X2 Am29000 Am29030 Am29035 Am29040 Am29050 A4 A10 A8 A6

**Microarchitectures:** K10 K5 K6 K6-2 K6-III K7 K8 K75

**Code Names:** DX Pluto Thunderbird ClawHammer SledgeHammer San Diego Toledo Windsor Manchester Brisbane Tyler Lion Propus Palomino Thoroughbred Kuma Barton SSA/5 Model 6 Little Foot Chomper Chomper Extended Sharptooth Venus Denmark Troy Italy Athens Egypt Santa Ana Budapest Santa Rosa Barcelona Shanghai Istanbul Heka Deneb Toliman Agena Lancaster Trinidad

**Processor Families:** Power2 Power3 Power3-II Power4+ Power5 Power6 Power7 PowerPC 7xx PowerPC 970 PowerPC 602 PowerPC 603 PowerPC 620 PowerPC 630 PowerPC 601 PowerPC 601v PowerPC 603e PowerPC 603ev PowerPC 604 PowerPC 604e PowerPC 604ev RS64 IV RS64-II RS64-III

**Microarchitectures:** Power6 PowerPC v2.00 PowerPC v2.02 PowerPC 601 PowerPC 602 PowerPC 603e PowerPC 604 PowerPC 620 PowerPC 630 PowerPC 7400 PowerPC 7450 PowerPC 7xx Power7

**Code Names:** P2SC RIOS2 Power2+ Giga Processor Arthur GP-UL Mach 5 Sstar Northstar Pulsar

**Processor Families:** microSPARC I SuperSPARC I SuperSPARC II UltraSPARC UltraSPARC II UltraSPARC IIe UltraSPARC IIi UltraSPARC III UltraSPARC III Cu UltraSPARC IIIi UltraSPARC T1 UltraSPARC T2

**Microarchitectures:** SuperSPARC I SuperSPARC II UltraSPARC UltraSPARC II UltraSPARC III UltraSPARC T1

**Code Names:** Tsunami Viking Voyager Spitfire Blackbird Sapphire-Black Hummingbird Sabre Sapphire-Red Phantom Cheetah Cheetah+ Jalepeno Niagara Niagara 2

Source: http://cpudb.stanford.edu/

# CPU


Source: https://www.loongson.cn/EN

# CPU

- **LA664系列**
- LA464系列
- LA364系列
- LA264系列
- LA132系列


**龙芯3A6000**

龙芯3A6000是龙芯第四代微架构处理器，面向高端嵌入式计算机、桌面、服务器等应用。采用自主…

信息化 桌面/终端 →


**龙芯3A6000工业级**

龙芯3A6000工业级芯片是面向工控应用领域的通用处理器，基于龙芯自主指令系统LoongArch的…

工控/嵌入式 CPU →


**龙芯3C6000/S**

龙芯 3C6000/S 是基于 LA664 处理器核设计的面向服务器领域最新一代 16 核 32 线程通用处理器,…

信息化 服务器 →


**龙芯3C6000/D**

龙芯 3C6000/D 是基于双 3C6000 硅片设计的最新一代 32 核通用处理器,通过龙链互连技术在基板…

信息化 服务器 →


**龙芯3C6000/Q**

龙芯 3C6000/Q 是基于四 3C6000 硅片设计的 64 核通用处理器,通过龙链互连技术在基板上实现高速…

信息化 服务器 →


**龙芯3B6000**

龙芯 3B6000 是基于 LA664 处理器核的 8 核 16 线程处理器,面向高端桌面及工作站应用。采…

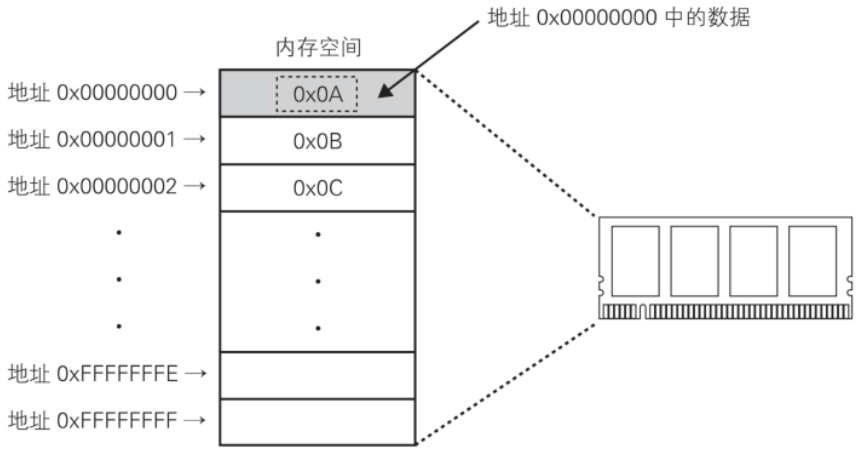
信息化 桌面/终端 →



Source: https://www.loongeco.cn/product/LoongArch

# Main Memory

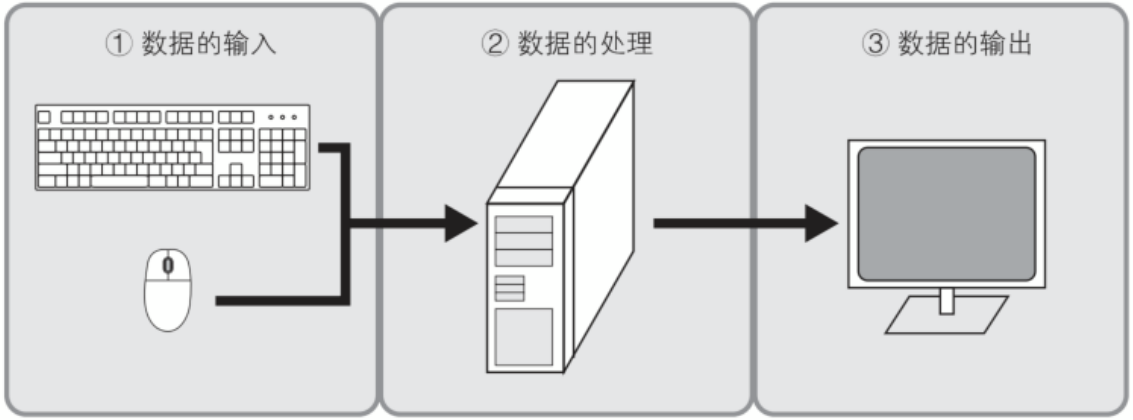
- Volatile （易失的）
  - The contents of the memory are lost when the computer is shut down
  - the contents of disk memory are retained even when the computer system is shut down
- Referred to as real memory or primary memory




图片来自《CPU自制入门》

# I/O modules

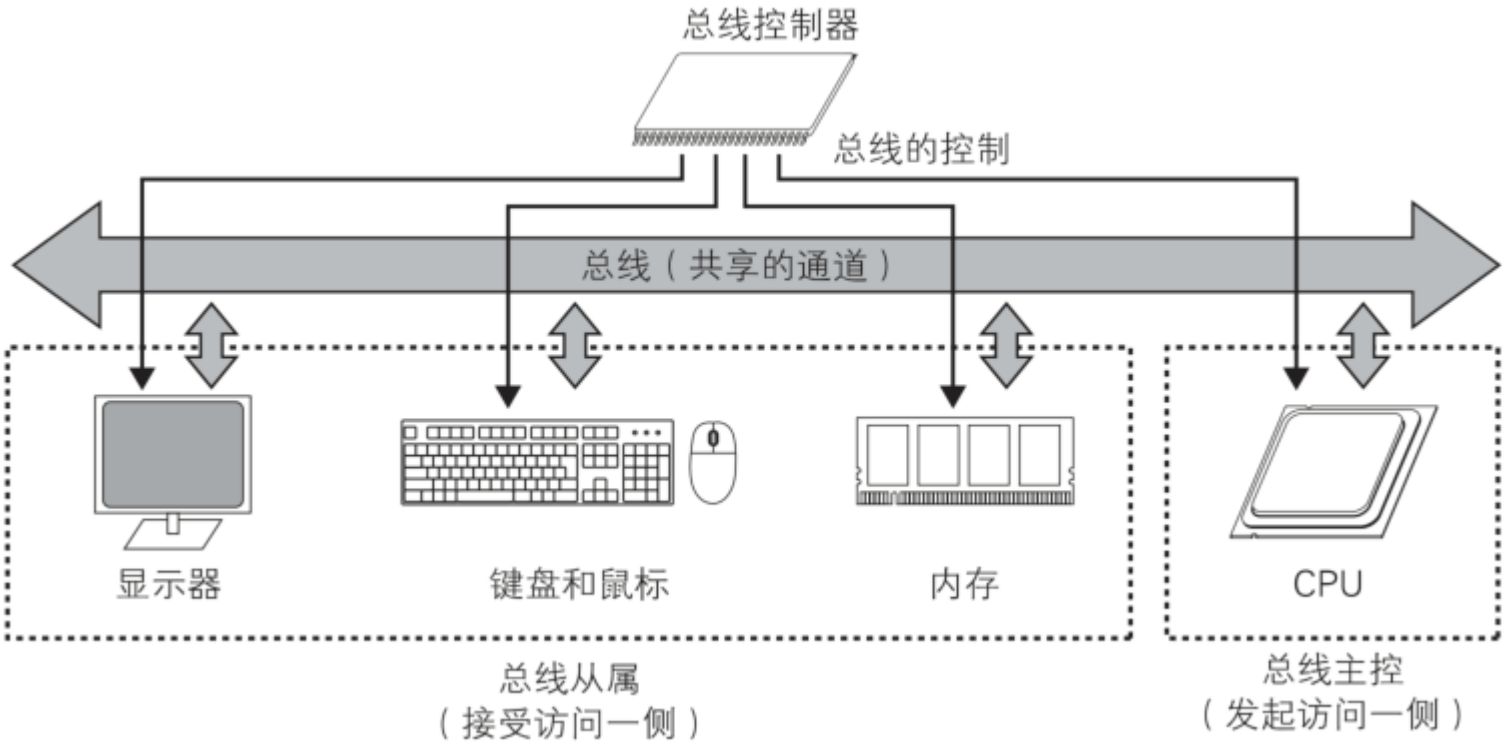
- Move data between the computer and its external environment.
- The external environment: secondary memory devices (e.g., disks), communications equipment, and terminals.




图片来自《CPU自制入门》

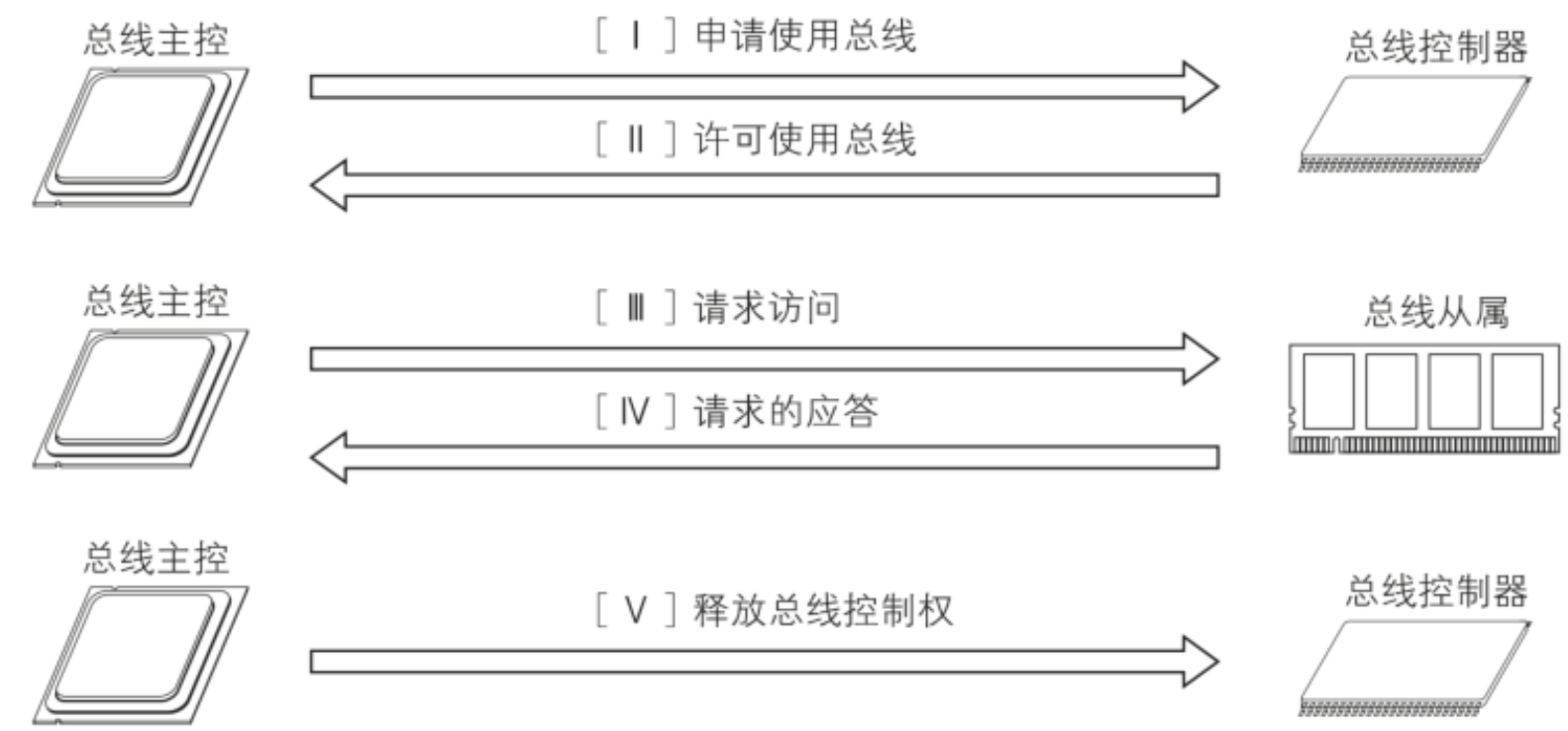
# System Bus

- Provides for communication among processors, main memory, and I/O modules.



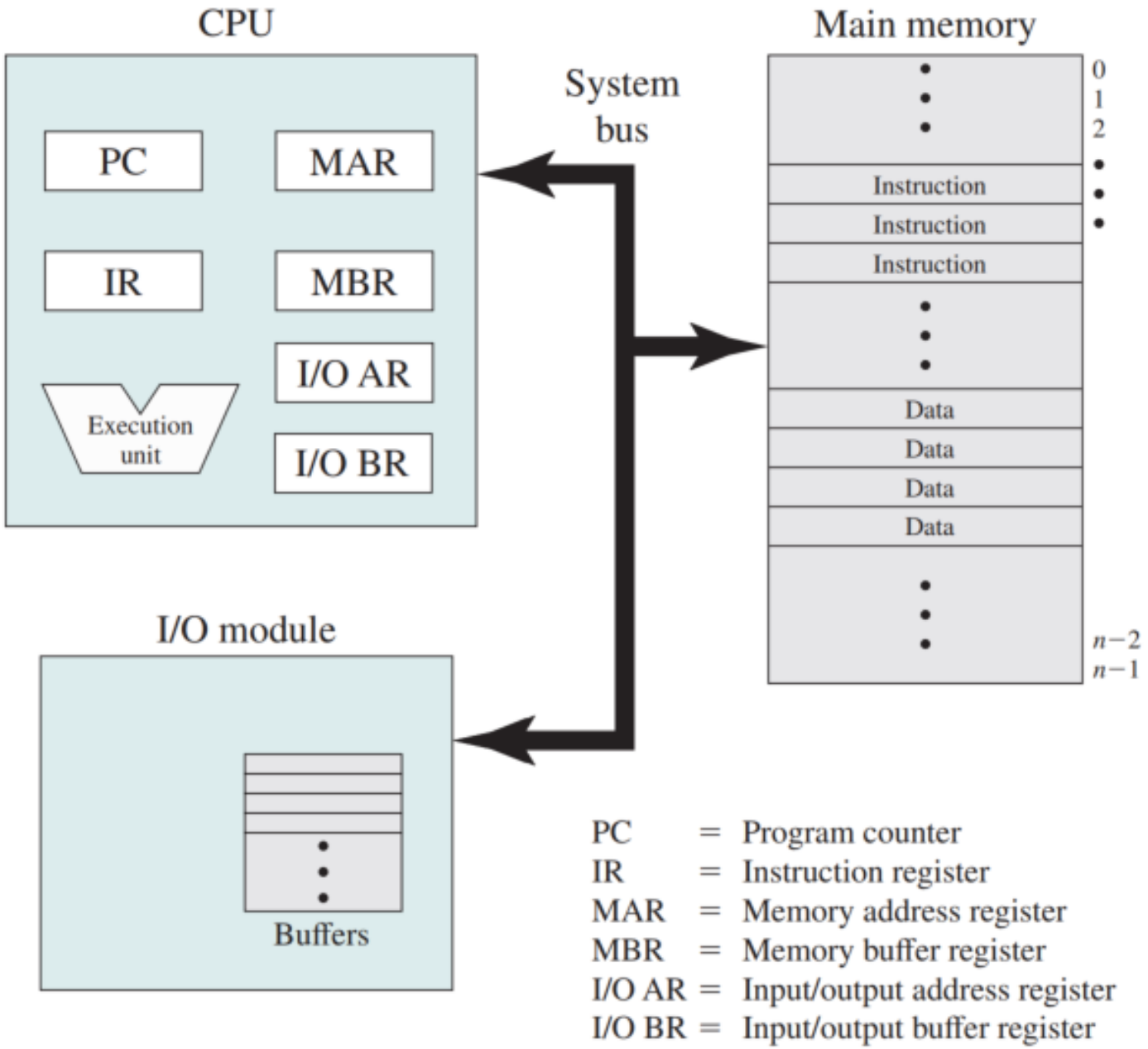

图片来自《CPU自制入门》

# System Bus




总线传输示例

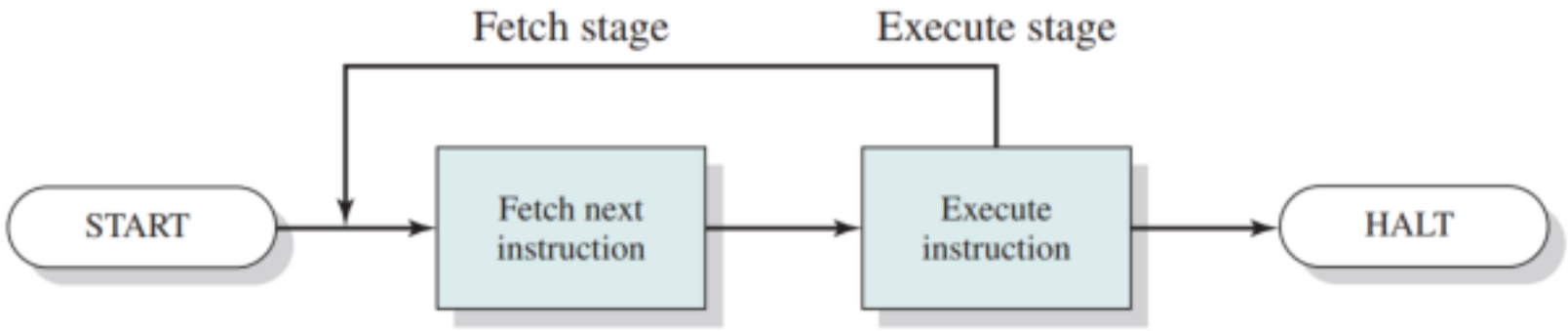
图片来自《CPU自制入门》




**Computer Components: Top-Level View**

# Instruction Execution

- A program consists of a set of instructions stored in memory
- Instruction processing consists of two steps:
  - reads (fetches) instructions from memory
  - executes each instruction



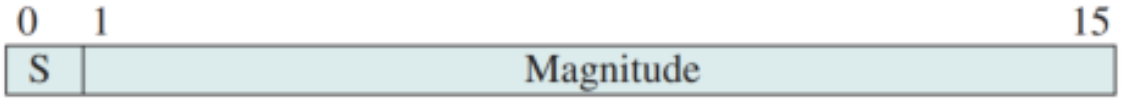

Basic Instruction Cycle

# Instruction Register (IR)

- The fetched instruction is loaded into the Instruction Register (IR)
- The processor interprets the instruction and performs the required action
  - Processor-memory
  - Processor-I/O
  - Data processing
  - Control

# Characteristics of a Hypothetical Machine


(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

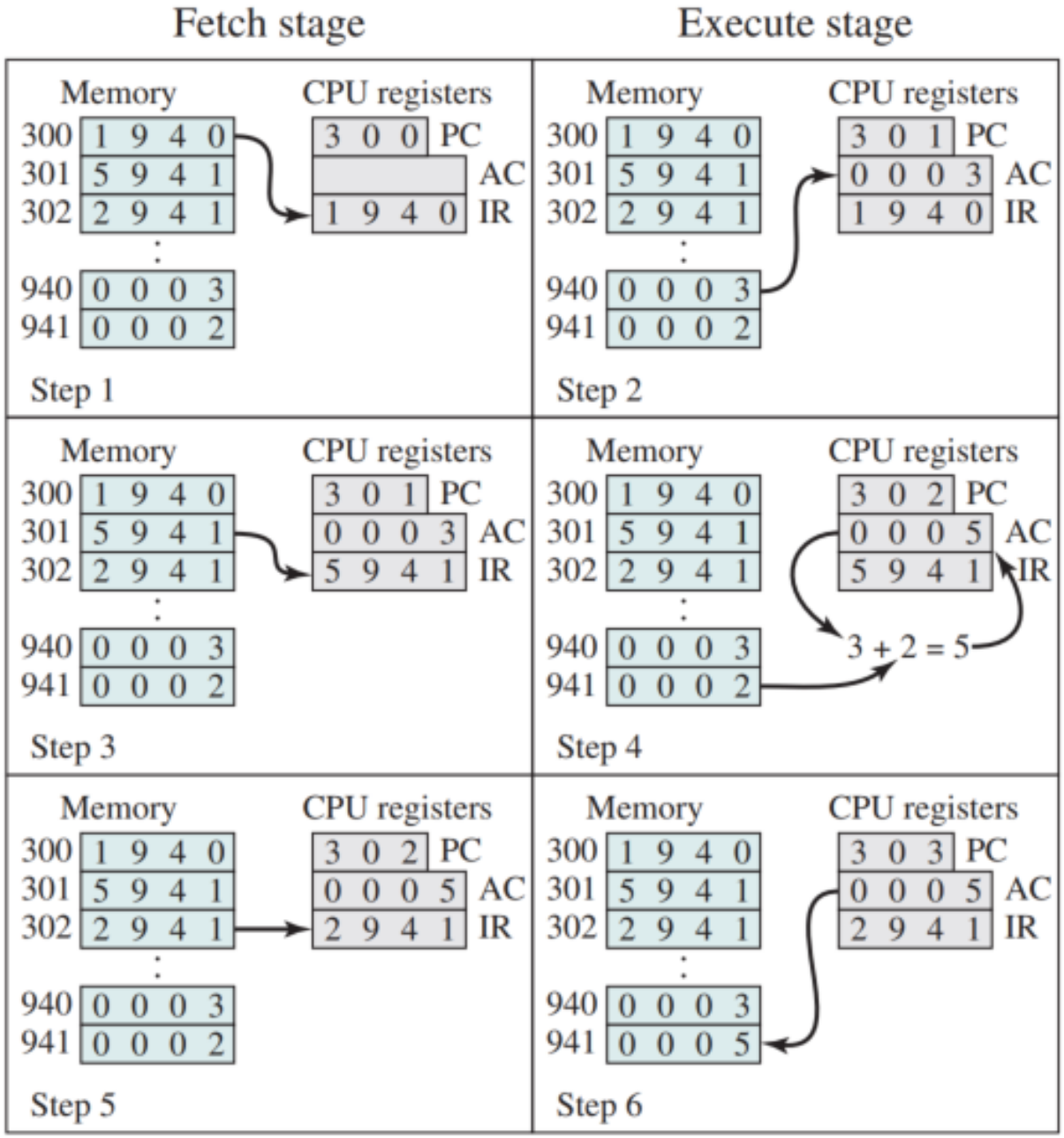
0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

(d) Partial list of opcodes

load 940

add 941

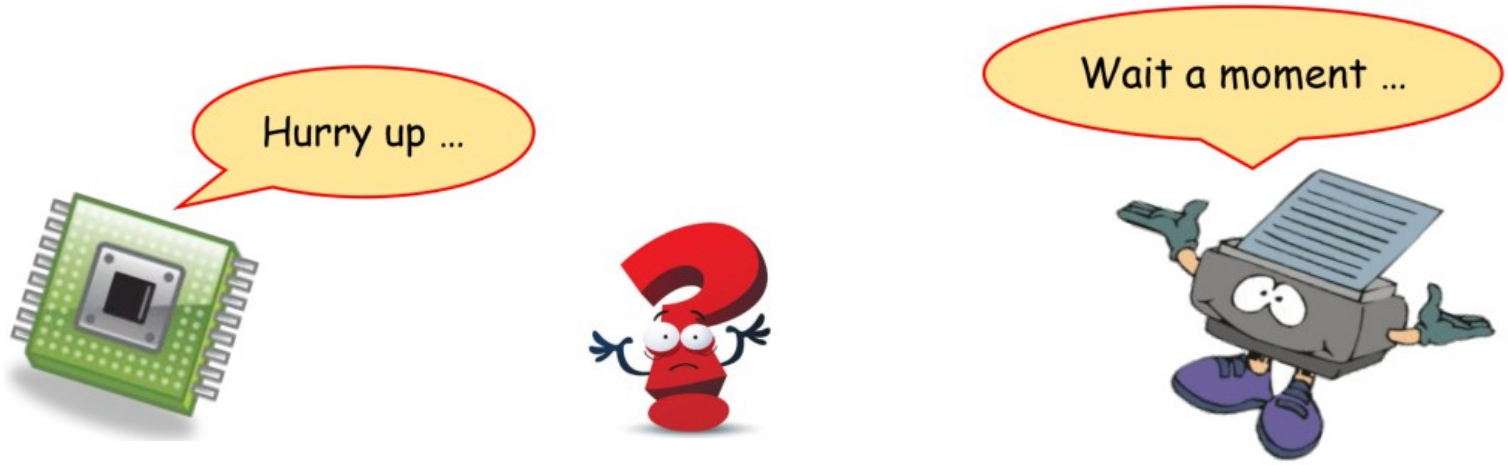
store 941




**Example of Program Execution (contents of memory and registers in hexadecimal)**

# Interrupts

- Interrupt the normal sequencing of the processor
- Provided to improve processor utilization
  - most I/O devices are slower than the processor
  - processor must pause to wait for device
  - wasteful use 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, 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 or to signal a variety of error conditions.

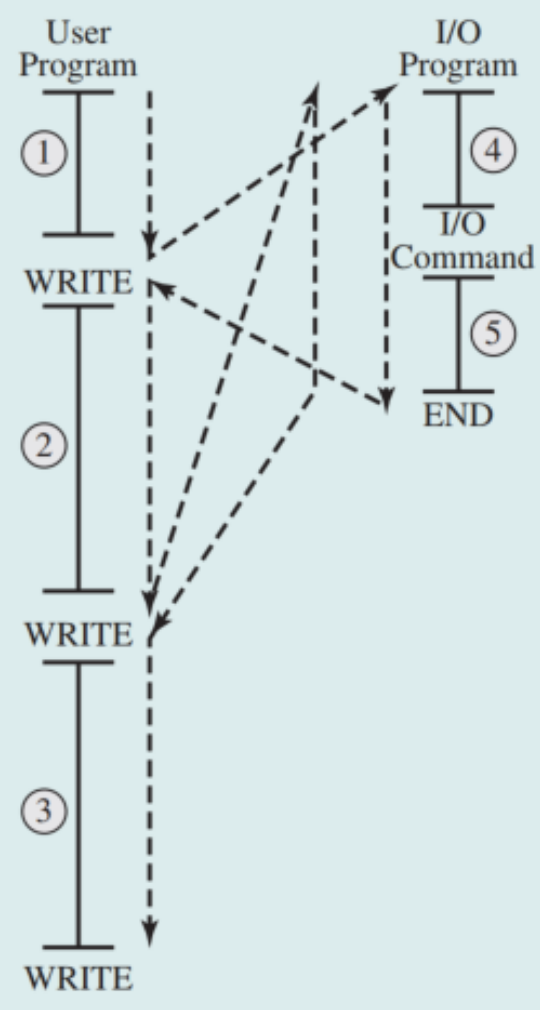
**Hardware failure**

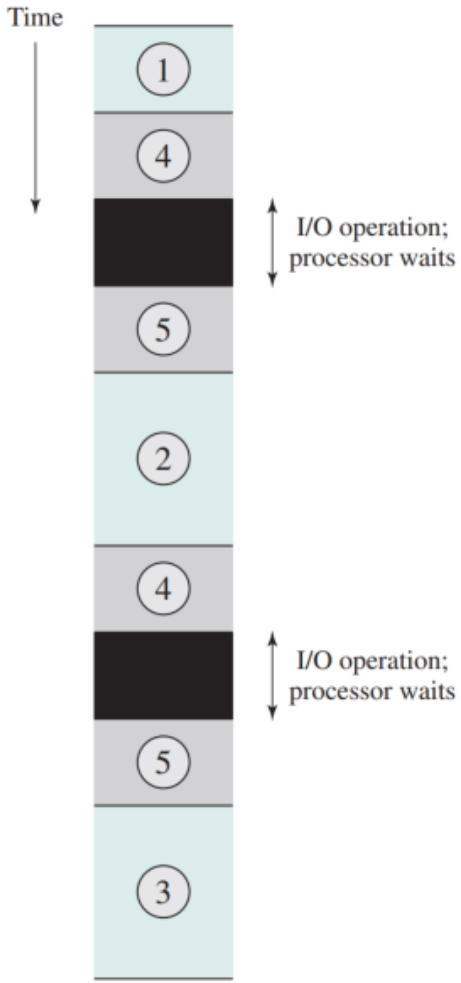
- Generated by a failure, such as power failure or memory parity error.

# I/O Devices Slower Than the Processor

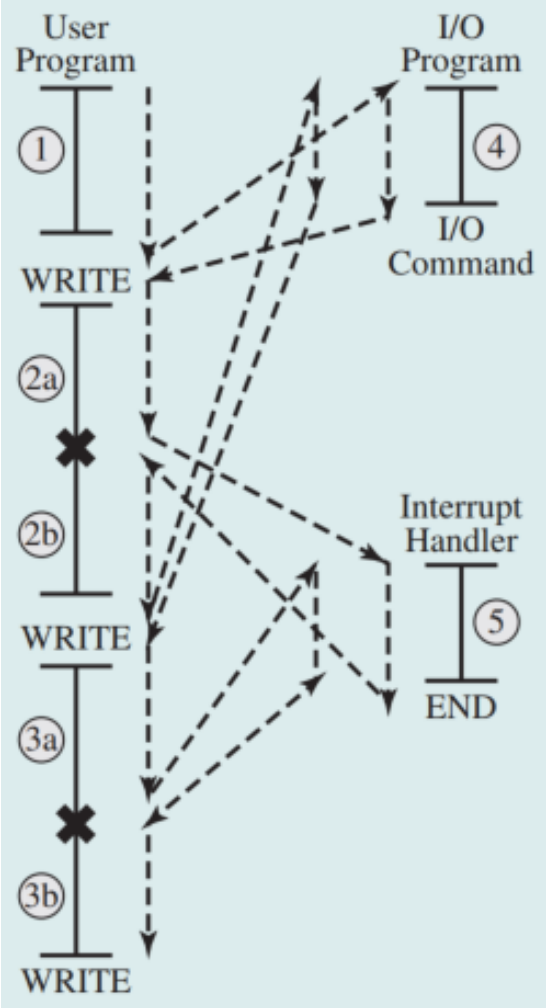
- An example
  - consider a PC that operates at 1 GHz, which would allow roughly $10^9$ instructions per second.
  - A typical hard disk has a rotational speed of 7200 revolutions per minute for a half-track rotation time of 4 ms
    - 4 million times slower than the processor.

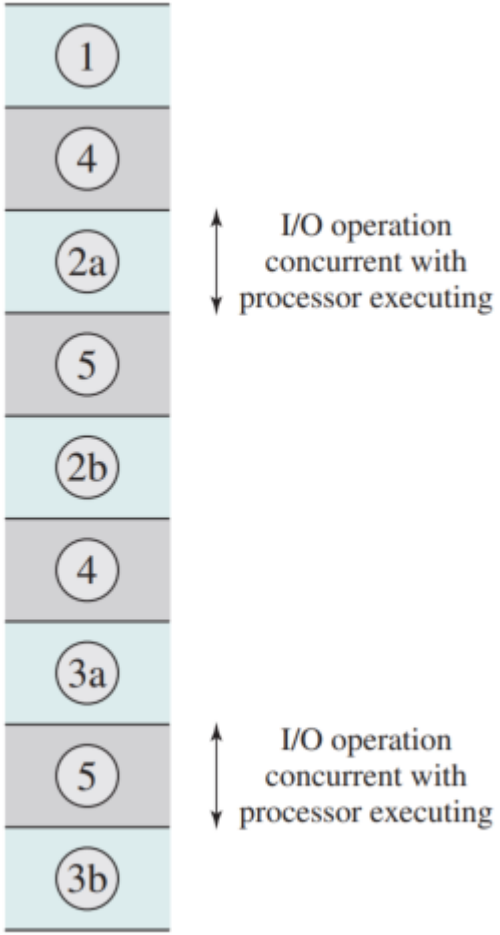
# Flow of Control Without Interrupts



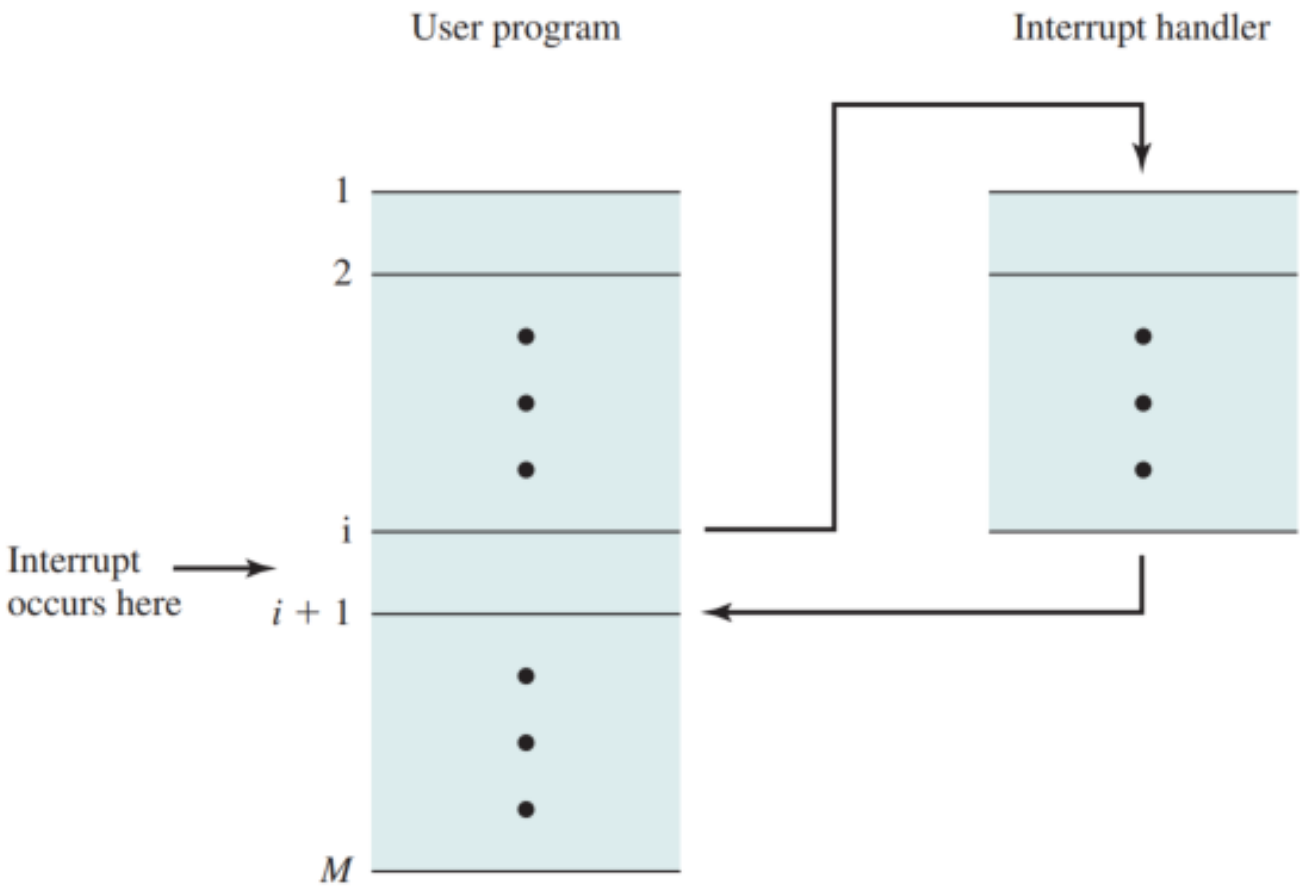
# Interrupts: short I/O wait



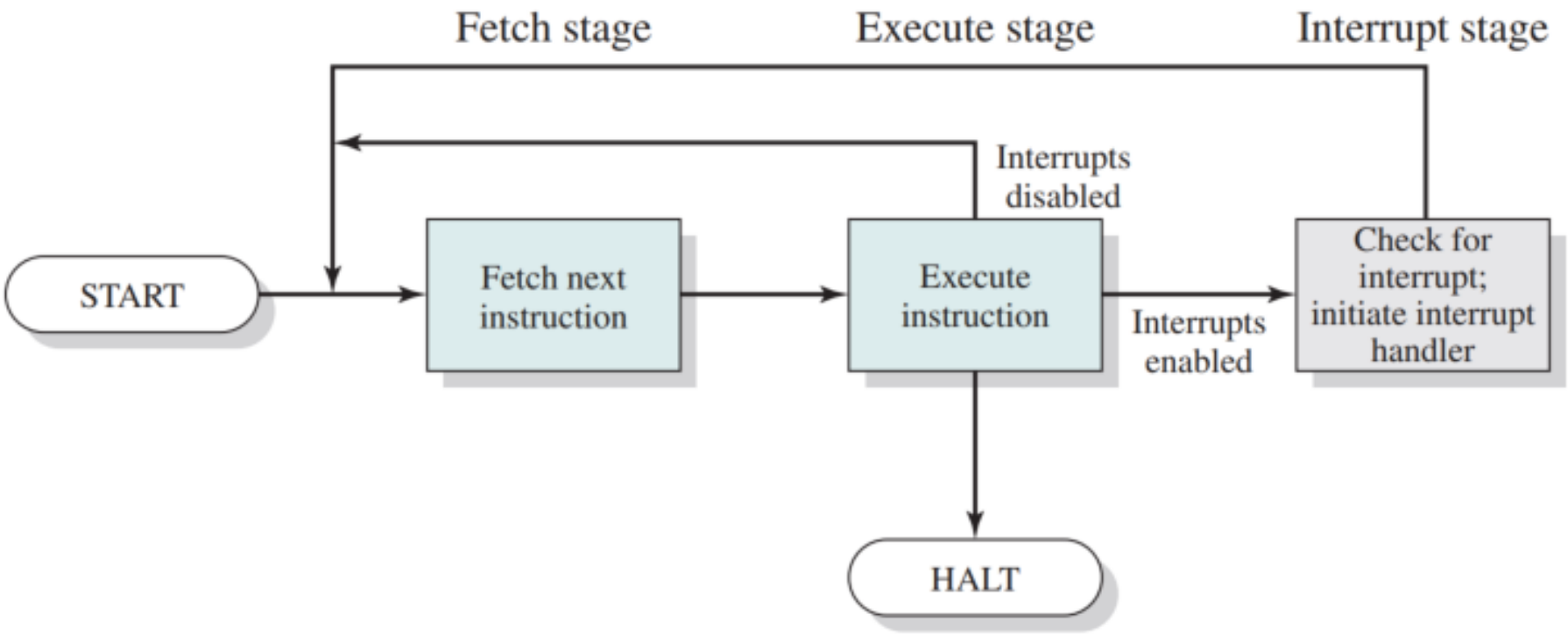




the second "I/O concurrent with processor executing" should be assigned to label (3a) not label (5)

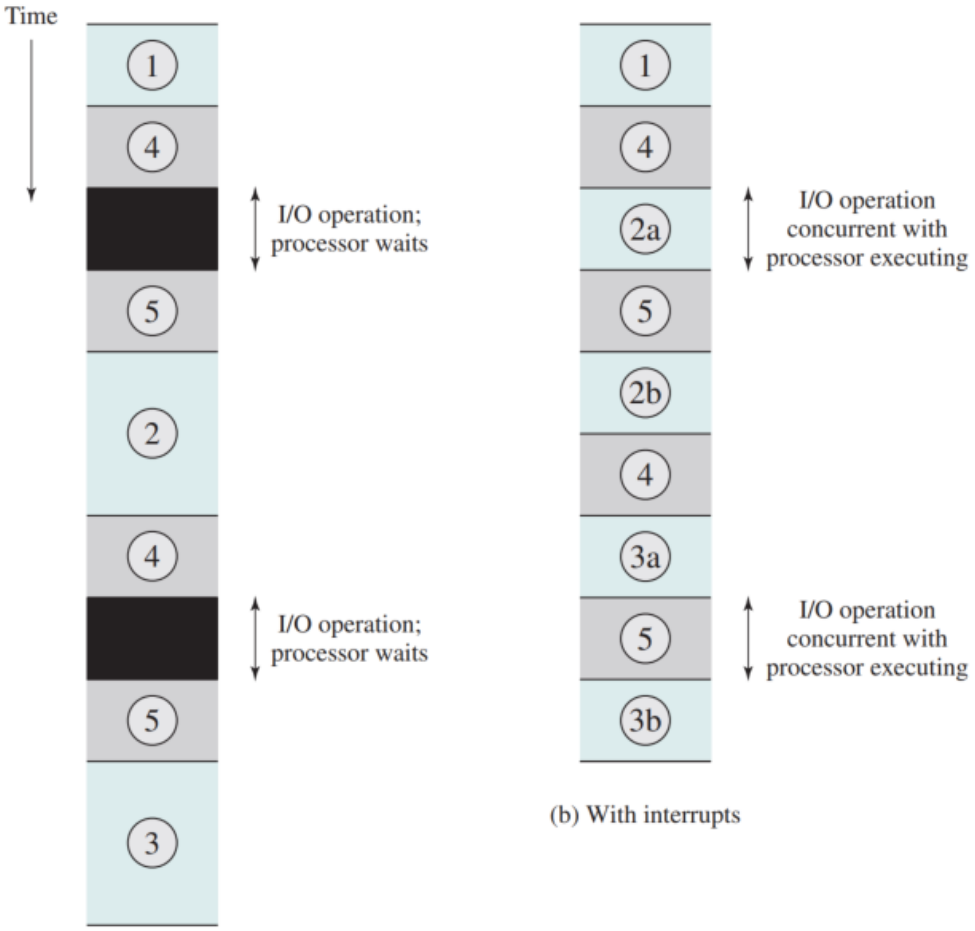
# Transfer of Control via Interrupts



# Instruction Cycle with Interrupts

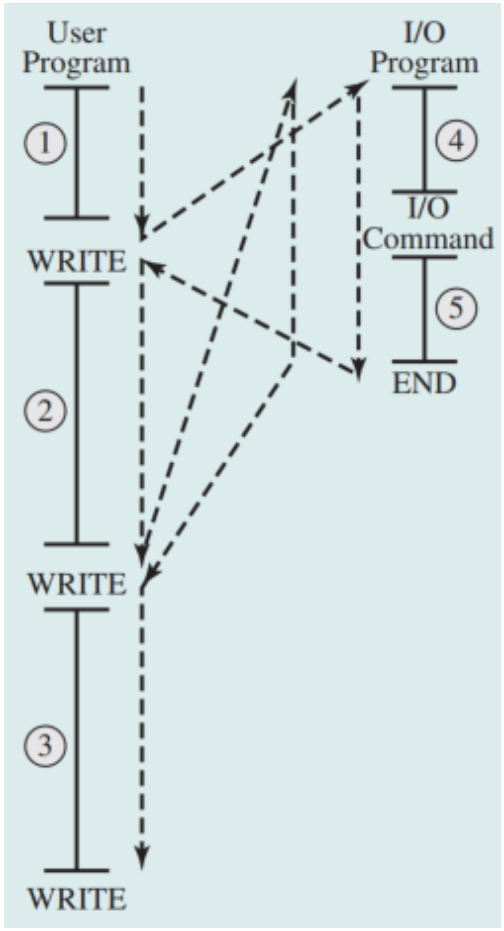


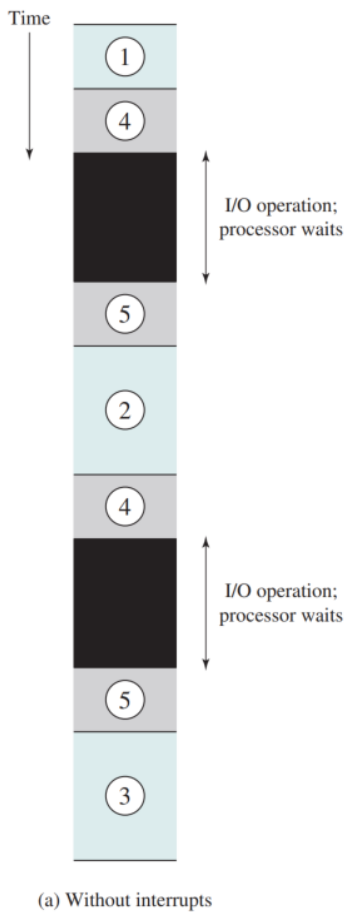
# Program Timing: Short I/O Wait



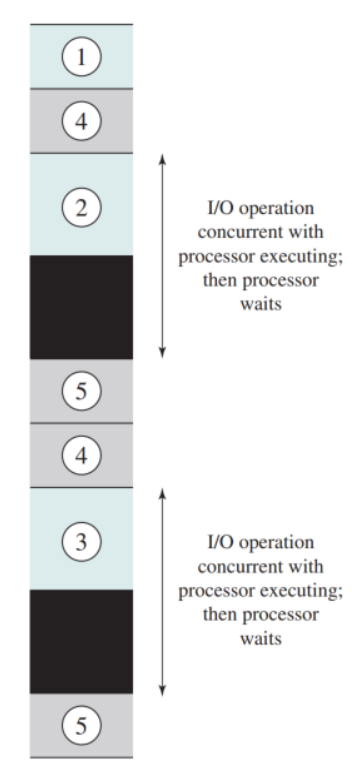

the second "I/O concurrent with processor executing" should be assigned to label (3a) not label (5)

# Program Timing: Long I/O Wait

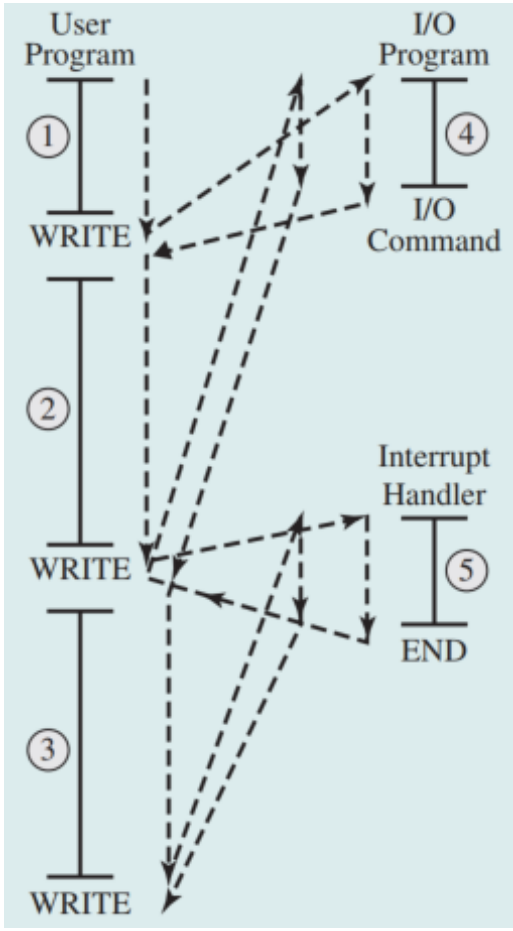


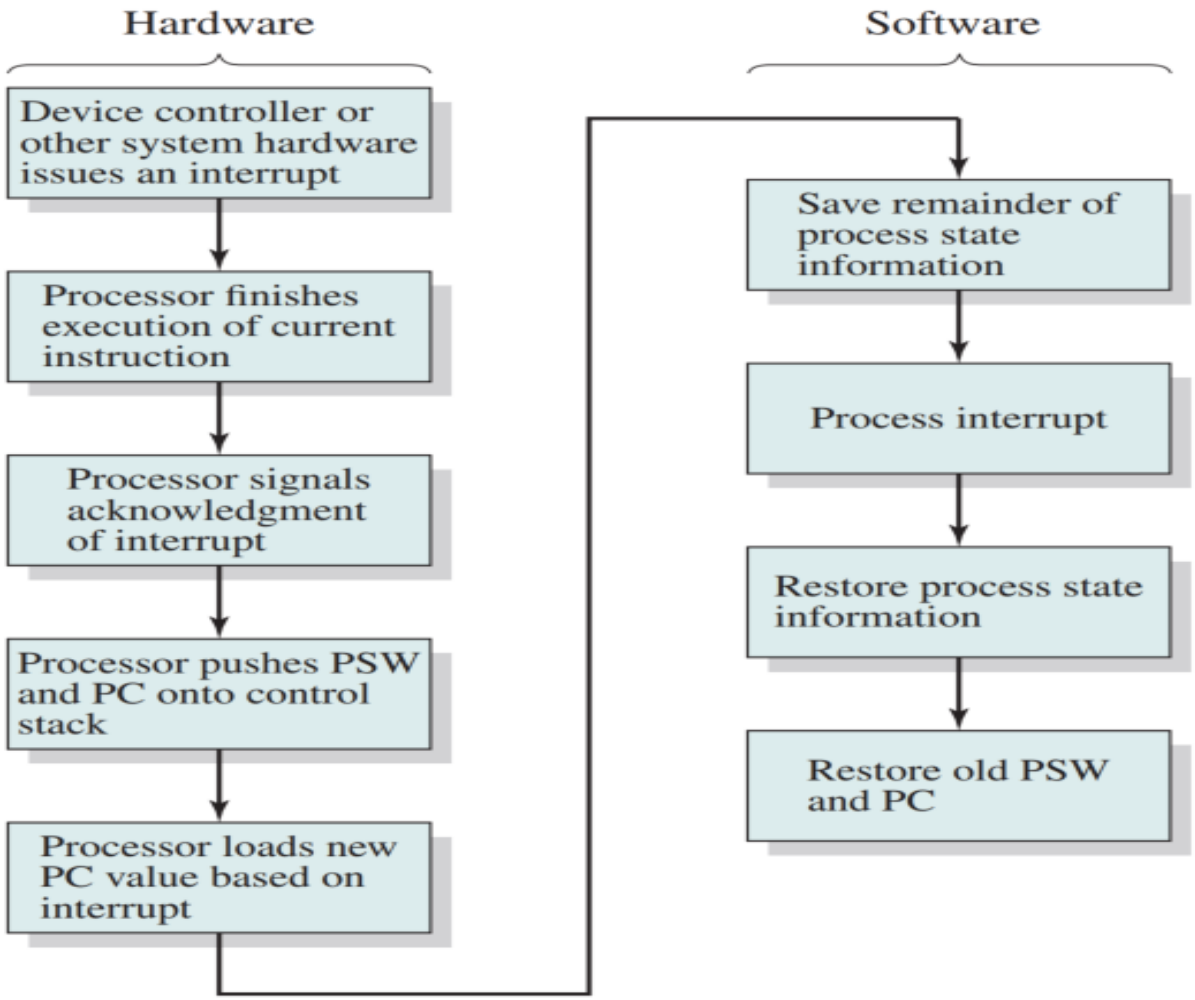

(a) Without interrupts

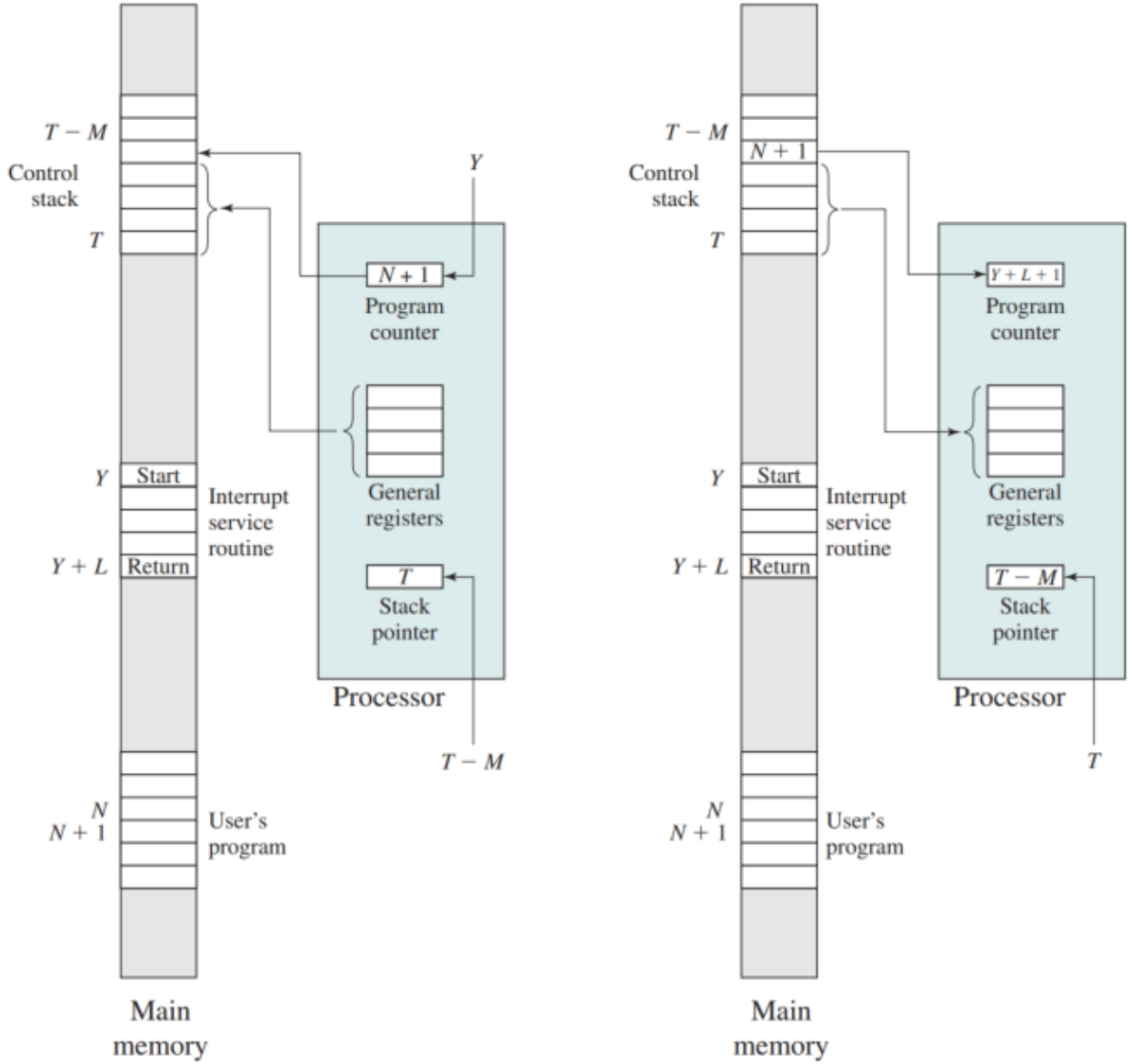



(b) With interrupts



# Simple Interrupt Processing






# Changes in Memory and Registers for an Interrupt

# Multiple Interrupts

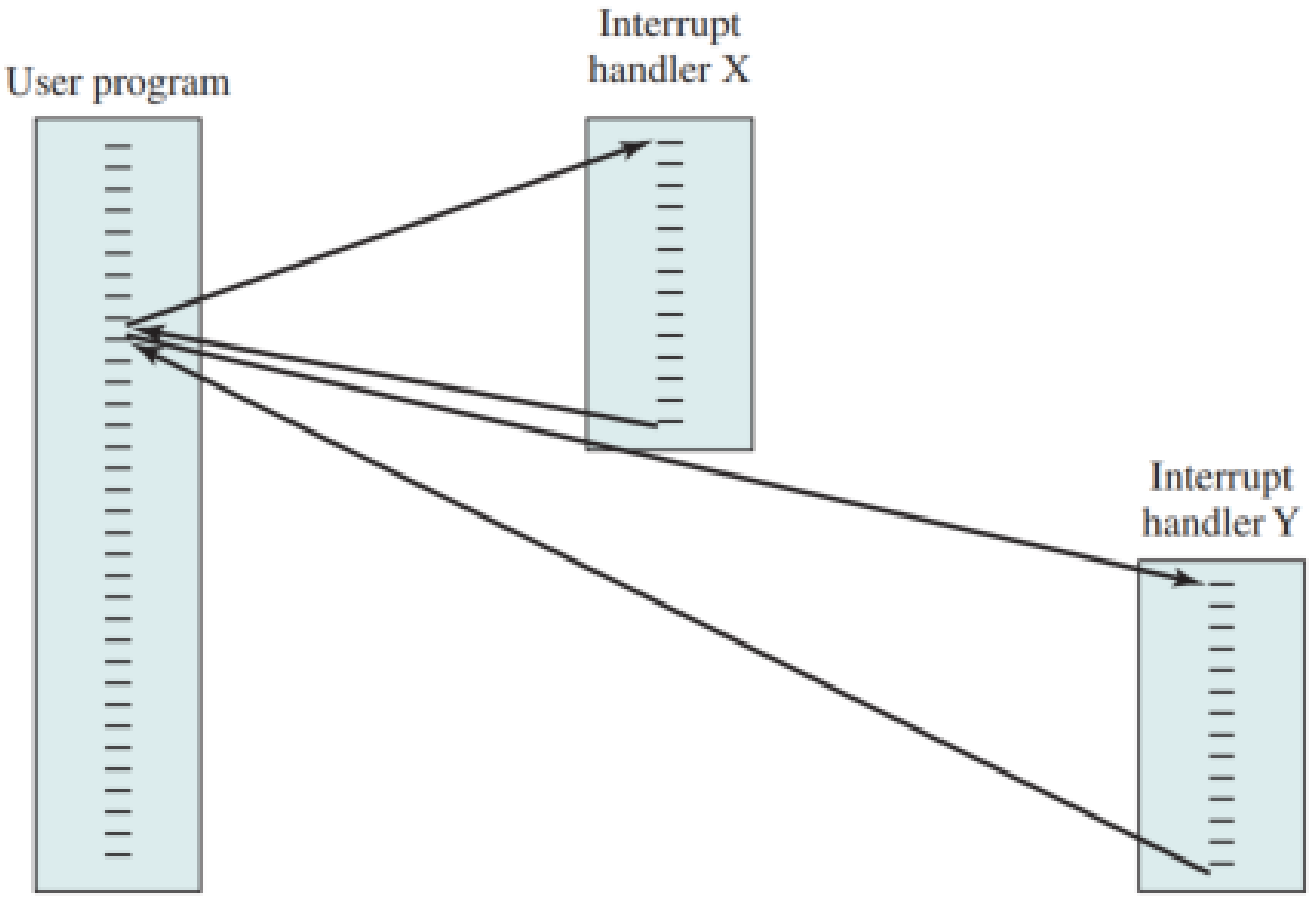
one or more interrupts can occur while an interrupt is being processed

- Example: receiving data from a communications line, and printing results at the same time

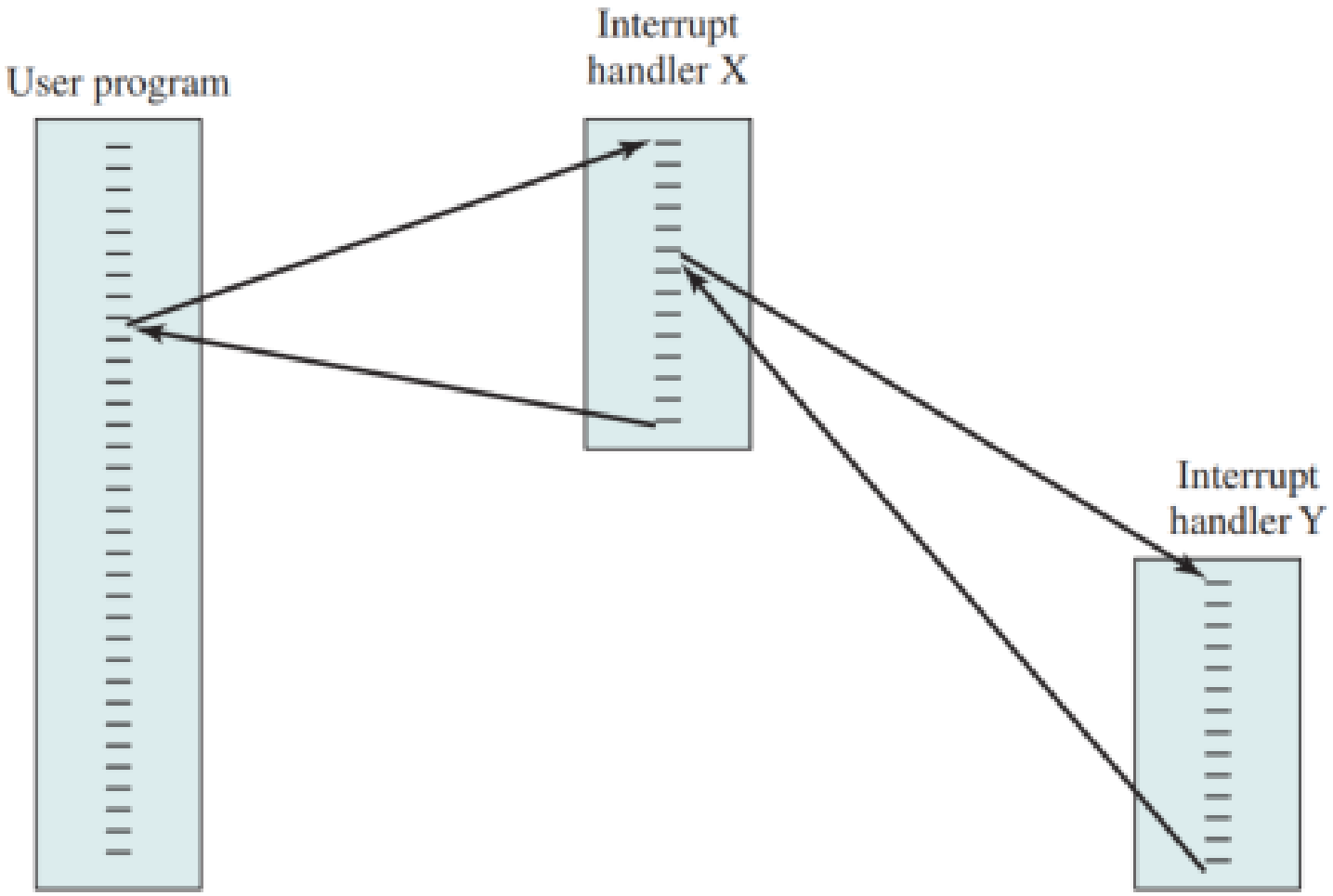
Two approaches:

- disable interrupts while an interrupt is being processed
- define priorities for interrupts and to allow an interrupt of higher priority to cause a lower-priority interrupt handler to be interrupted

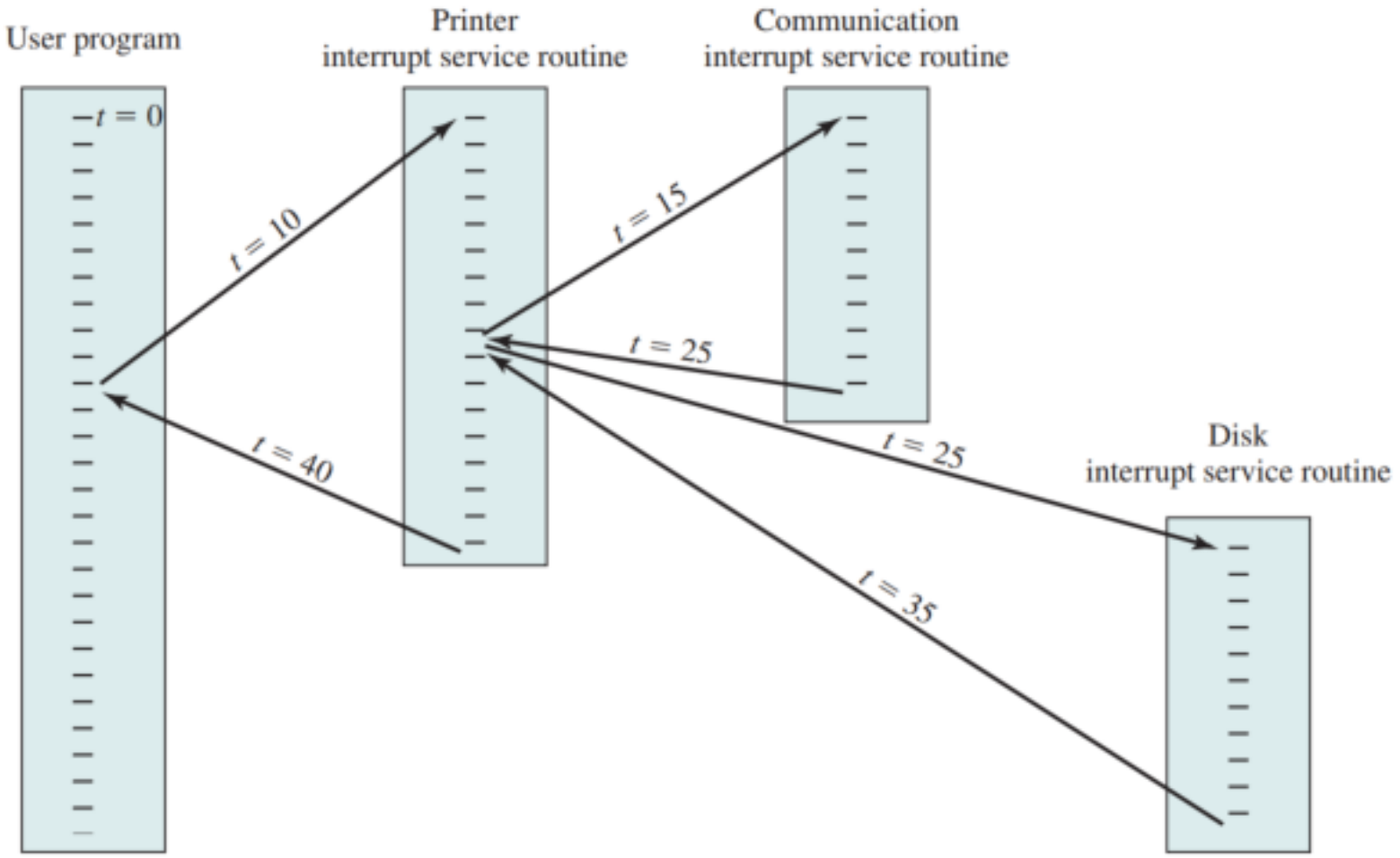
# Sequential interrupt processing



# Nested interrupt processing



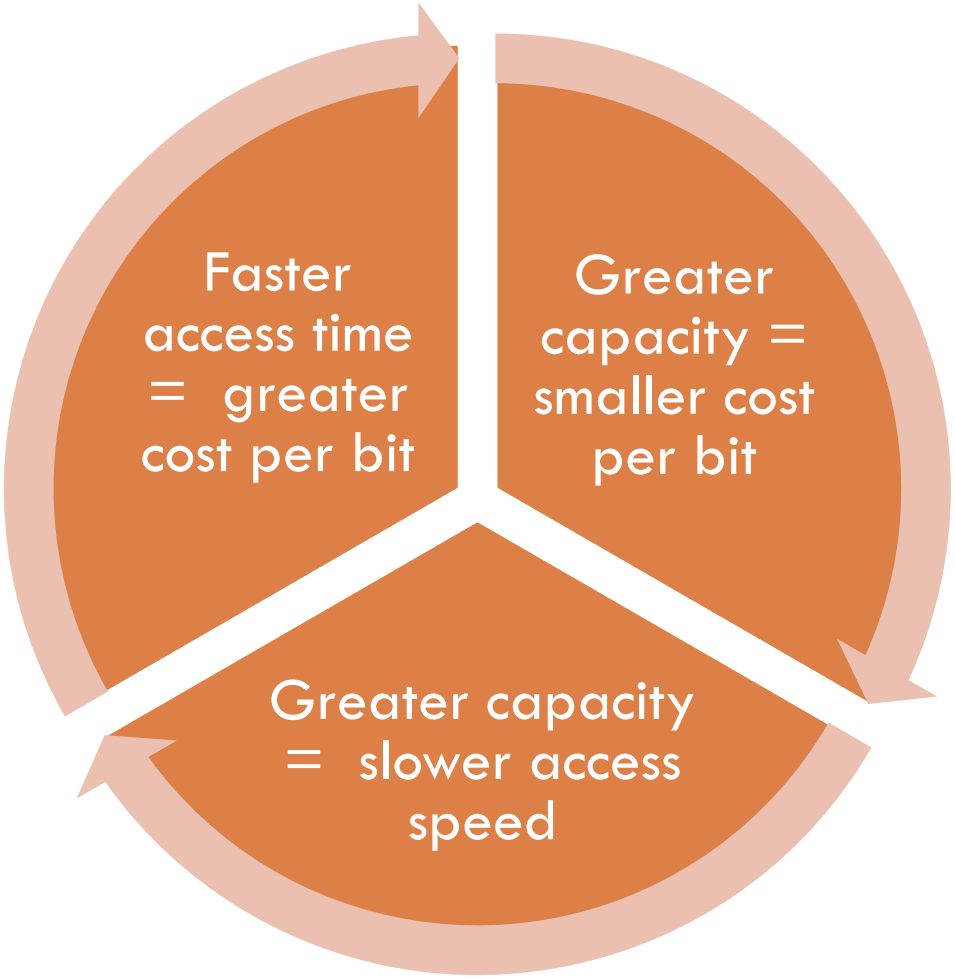
# Example Time Sequence of Multiple Interrupts



# Memory Hierarchy

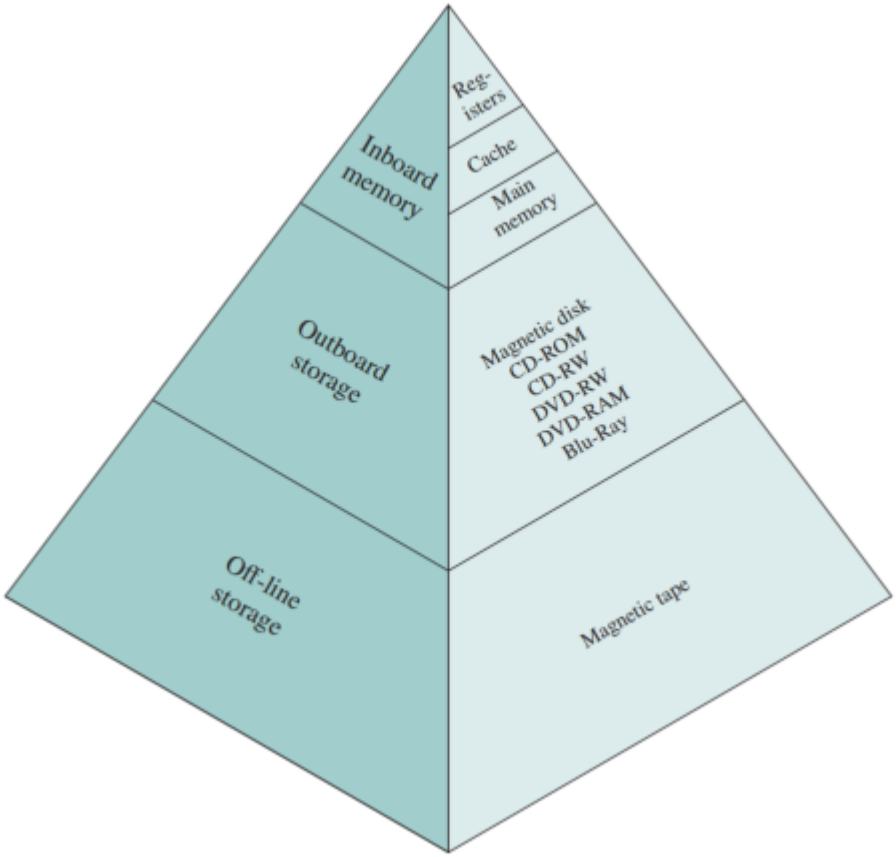
- The design constraints on a computer’s memory
  - Amount -- How much?
  - Speed -- How fast?
  - Expense(cost) -- How expensive?
- The memory must be able to keep up with the processor
- The cost of memory must be reasonable in relationship to other components

# Memory Relationships



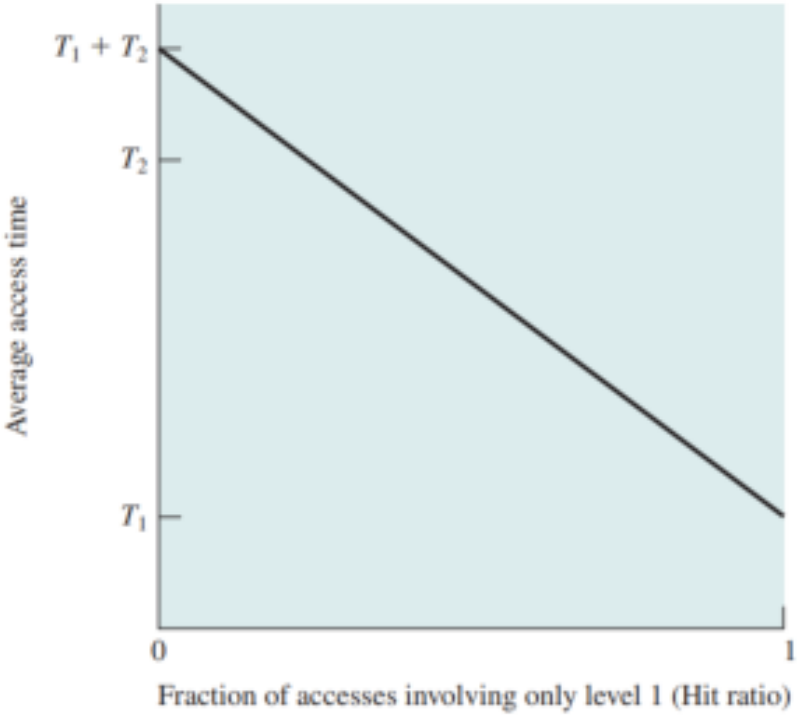
# The Memory Hierarchy

- Going down the hierarchy:
  a. Decreasing cost per bit
  b. Increasing capacity
  c. Increasing access time
  d. Decreasing frequency of access to the memory by the processor



# Performance of a Simple Two-Level Memory

- Hit ratio H:
  - H is define as the fraction of all memory accesses that are found in the faster memory (e.g., the cache)




Suppose 95% of the memory accesses are found in the cache ($H$ = 0.95).
The average time to access a byte can be expressed as:

**(0.95)(0.1 us) + (0.05)(0.1 us + 1 us) = 0.095 + 0.055 = 0.15 us**

# Principle of Locality

- Memory references by the processor, for both instructions and data, tend to cluster.
- Organize data across the hierarchy such that the percentage of accesses to each successively lower level is substantially less than that of the level above.
- Can be applied across more than two levels of memory

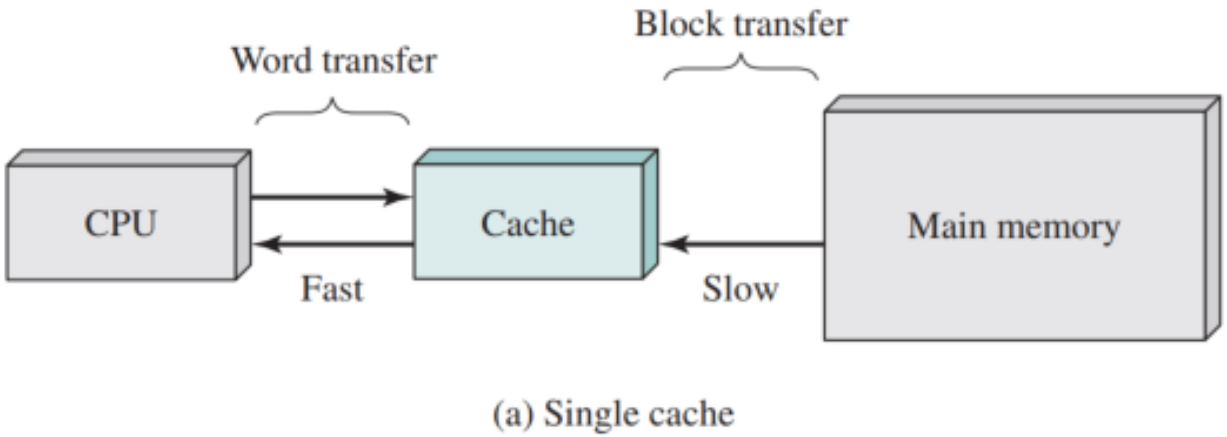
# Secondary Memory

- Also referred to as auxiliary memory
  - External
  - Nonvolatile
  - Used to store program and data files

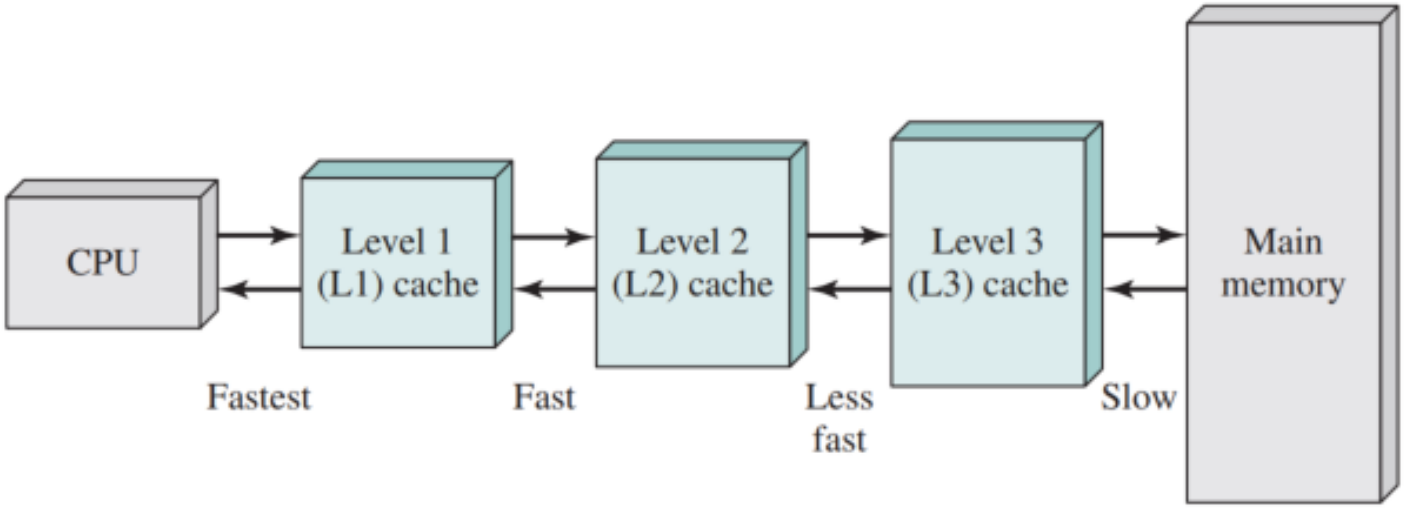
# Cache Memory

- Invisible to the OS
- Interacts with other memory management hardware
- Processor must access memory at least once per instruction cycle
- Processor execution is limited by memory cycle time
- Exploit the principle of locality with a small, fast memory

# Cache and Main Memory

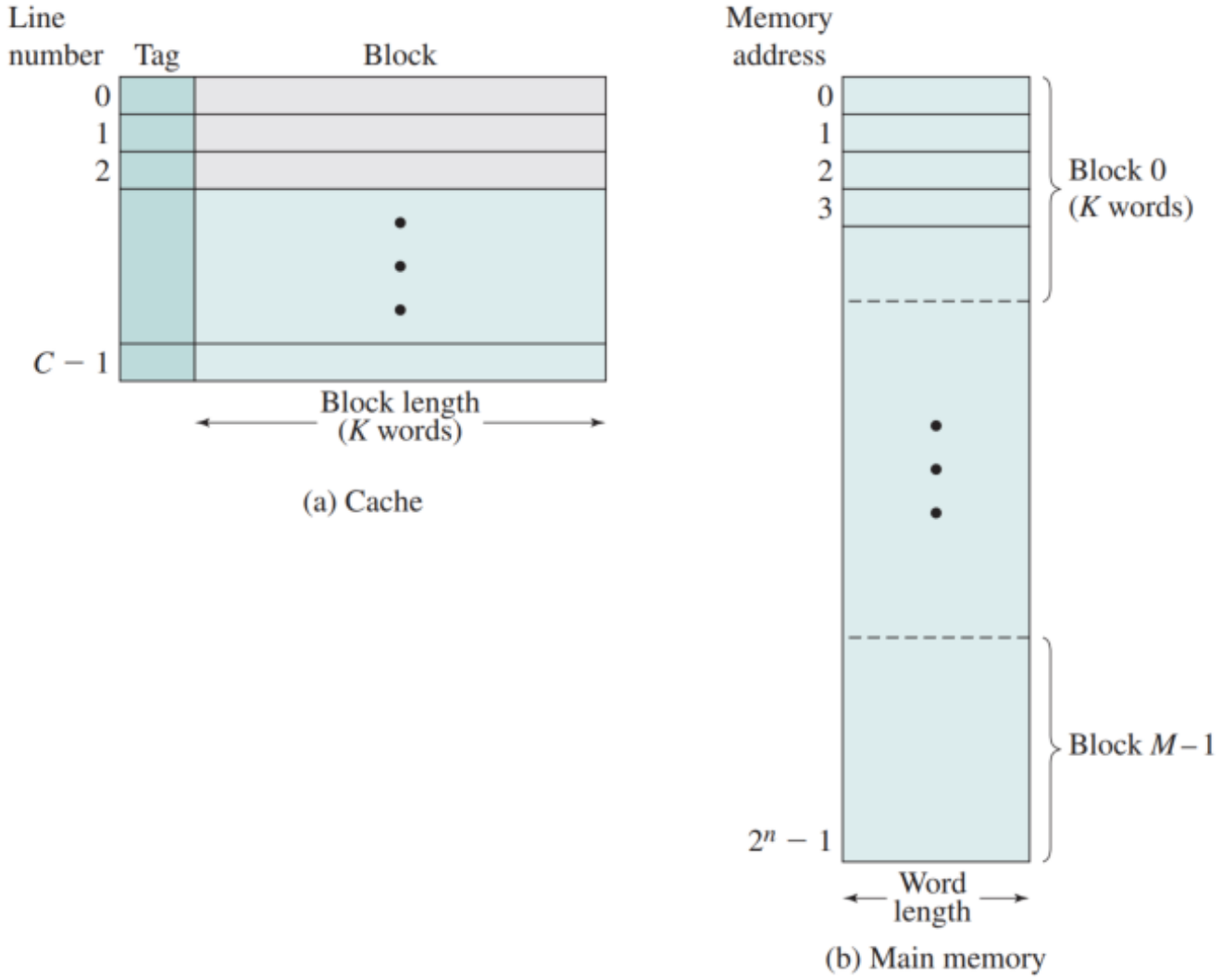



(a) Single cache



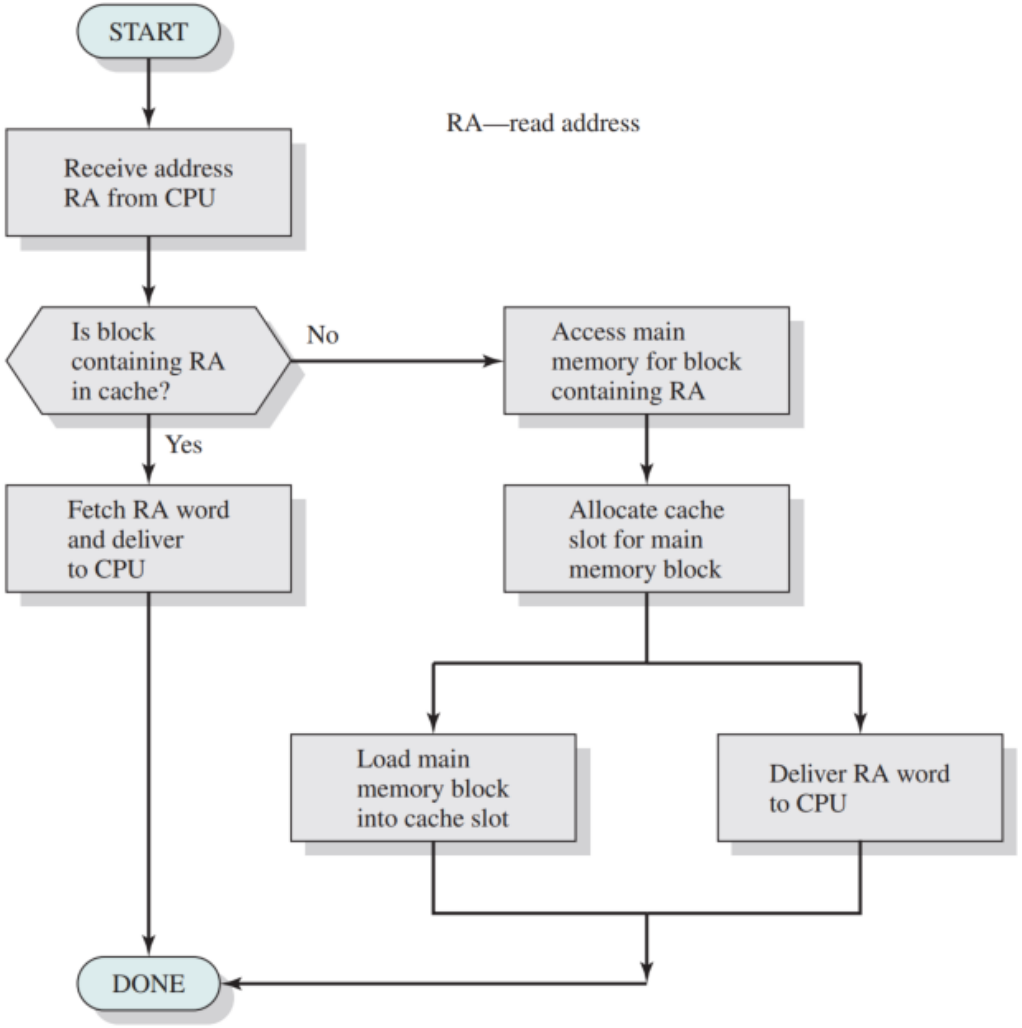

(b) Three-level cache organization

# Cache/Main Memory Structure



# Cache Read Operation

- Mapping function
- Replacement algorithm
- Write policy
- Number of cache levels

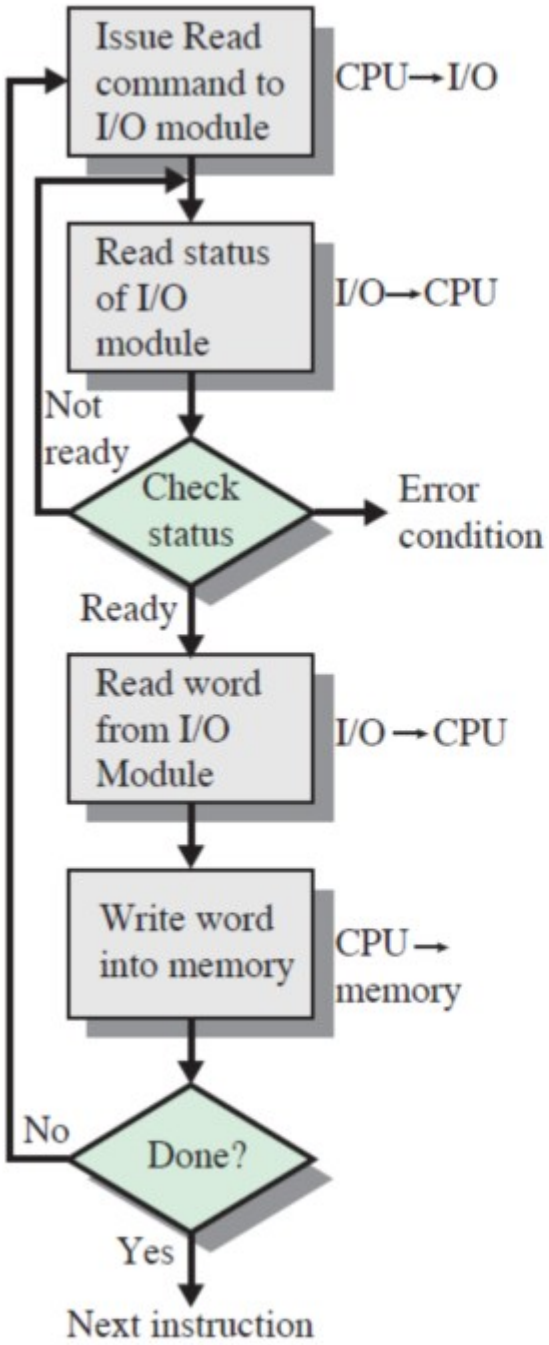


# I/O Techniques

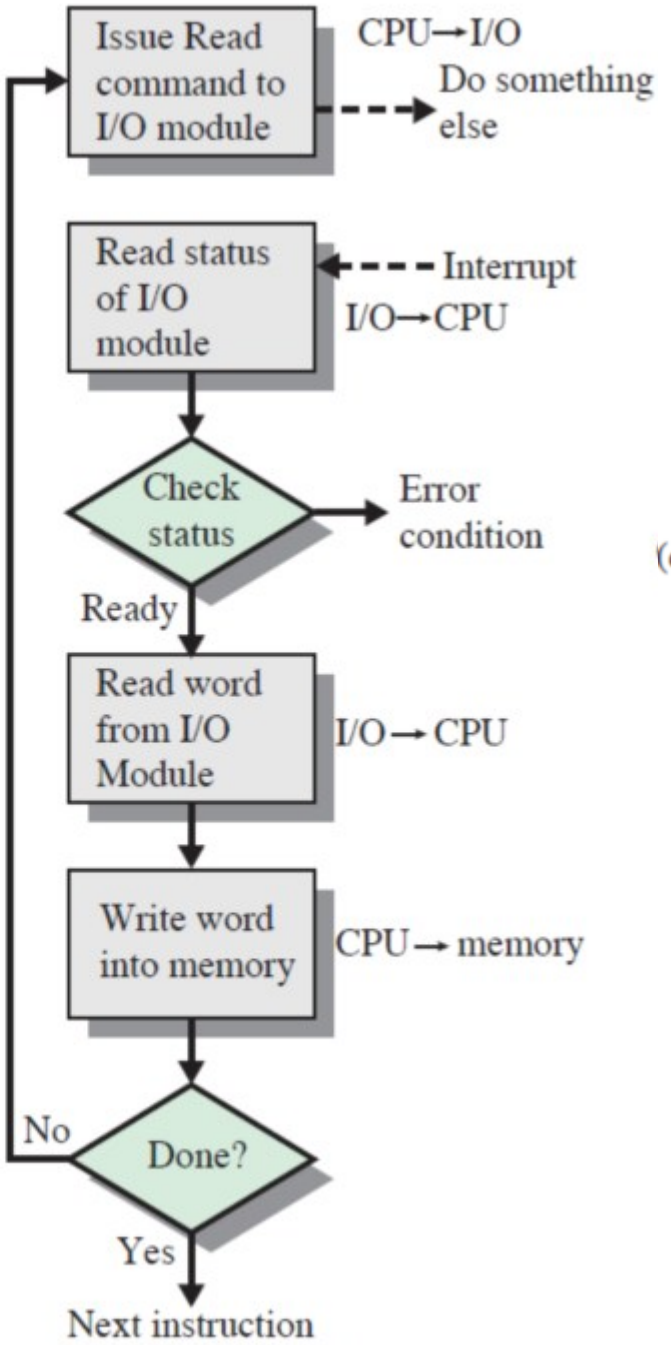
- Three techniques are possible for I/O operations
  - programmed I/O
  - interrupt-driven I/O
  - direct memory access (DMA)

# DMA

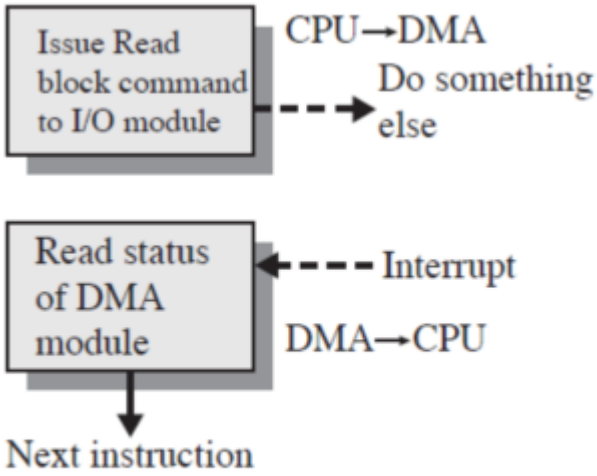
- It allows certain hardware subsystems to access main system memory independently of CPU
  - performed by a separate module on the system bus
  - incorporated into an I/O module
- It issues a command to the DMA module containing
  - whether a read or write is requested
  - the address of the I/O device involved
  - the starting location in memory to read/write
  - the number of words to be read/written




(a) Programmed I/O




(b) Interrupt-driven I/O




(c) Direct memory access

# Compare three techniques

**Programmed/interrupt-driven I/O**

☒ Transfer rate is limited by the speed with which the processor can test and service a device

☒ The processor is tied up in managing an I/O transfer, a number of instructions must be executed for each I/O transfer

**DMA**

☑ Processor is involved only at the beginning and end of the transfer

☑ More efficient

☒ Processor executes more slowly during a transfer when processor access to the bus is required

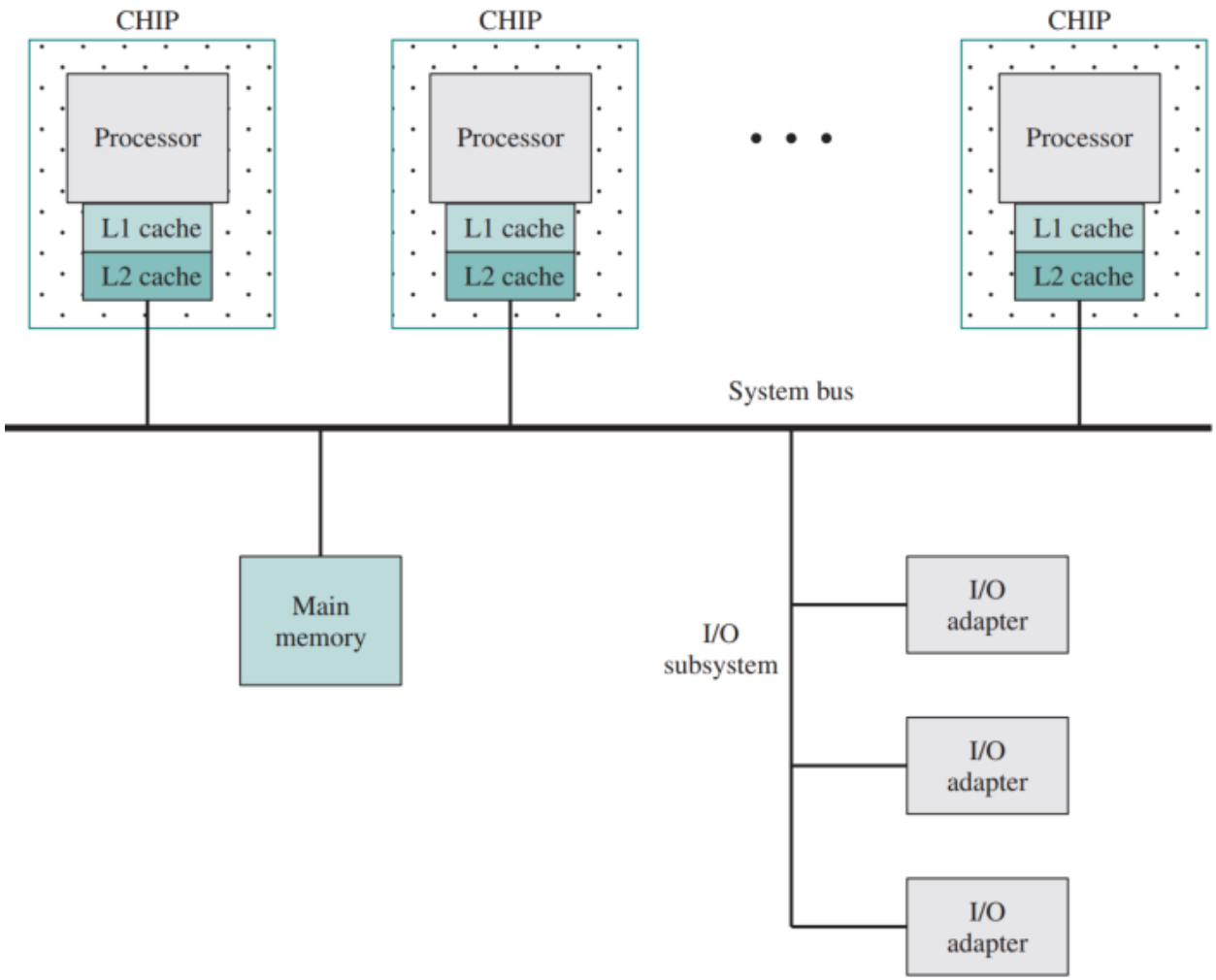
# Symmetric Multiprocessors

- An SMP can be defined as a stand-alone computer system with the following characteristics
  1. There are two or more similar processors of comparable capability.
  2. These processors share the same main memory and I/O facilities and are interconnected by a bus or other internal connection scheme, such that memory access time is approximately the same for each processor.
  3. All processors share access to I/O devices, either through the same channels or through different channels that provide paths to the same device.
  4. All processors can perform the same functions (hence the term symmetric).
  5. The system is controlled by an integrated operating system that provides interaction between processors and their programs at the job, task, file, and data element levels.

# Symmetric Multiprocessors

- An SMP organization has a number of potential advantages over a uniprocessor organization
  - **Performance:** If the work to be done by a computer can be organized such that some portions of the work can be done in parallel, then a system with multiple processors will yield greater performance than one with a single processor of the same type.
  - **Availability:** In a symmetric multiprocessor, because all processors can perform the same functions, the failure of a single processor does not halt the machine. Instead, the system can continue to function at reduced performance.
  - **Incremental growth:** A user can enhance the performance of a system by adding an additional processor.
  - **Scaling:** Vendors can offer a range of products with different price and performance characteristics based on the number of processors configured in the system.

# Symmetric Multiprocessor Organization



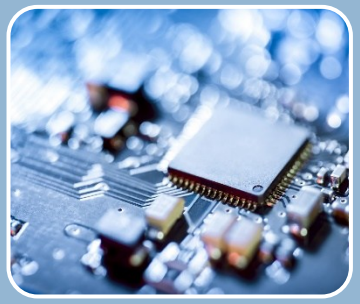
# Multicore Computer

- Also known as a chip multiprocessor
- Combines two or more processors (cores, 核) on a single piece of silicon (die, 片)
  - each core consists of all of the components of an independent processor
- In addition, multicore chips also include L2 cache and in some cases L3 cache

# Intel Core i7-5960X

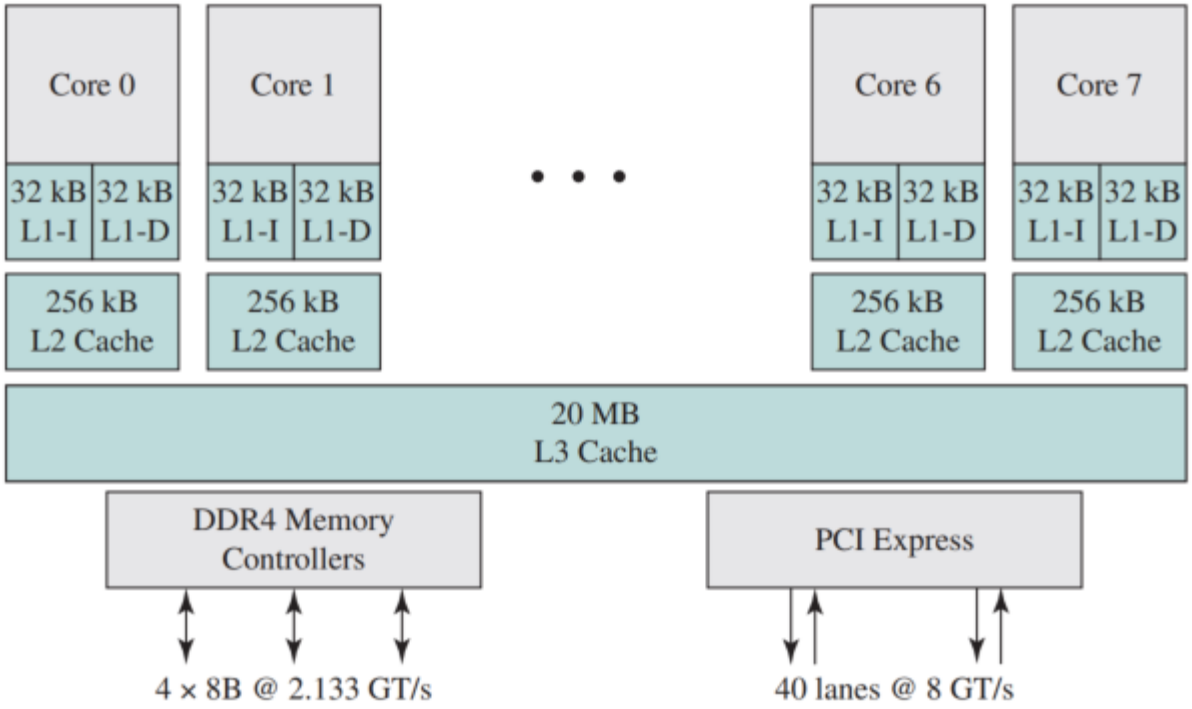
- Supports two forms of external communications to other chips

**DDR4 memory controller**
brings the memory controller for the DDR (double data rate) main memory onto the chip.
With the memory controller on the chip, the Front Side Bus is eliminated

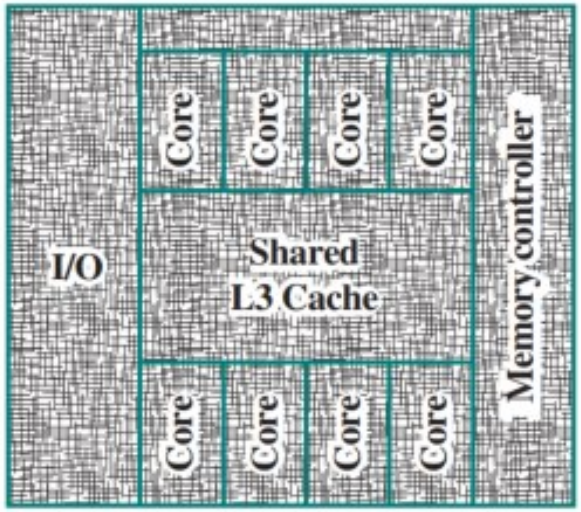


The **PCI Express**
a peripheral bus and enables high-speed communications among connected processor chips.
The PCI Express link operates at 8 GT/s (transfers per second). At 40 bits per transfer, that adds up to 40 GB/s.

# Intel Core i7-5960X Block Diagram




(a) Block diagram




(b) Physical layout on chip

# Summary

- Basic Elements
  - processor, main memory, I/O modules, system bus
- Instruction execution
  - processor-memory, processor-I/O, data processing, control
- Interrupt/Interrupt Processing
- Memory Hierarchy
  - Cache/cache principles and designs
- I/O Techniques
- Multiprocessor/multicore