Device Management

设备管理

Chapter 4

电子科技大学

University of Electronic Science and Technology of China

内　　容

1. I/O Devices
2. Organization Of The I/O Function
3. OS 设备管理模块的结构
4. I/O Buffering
5. Disk Scheduling Policies

Learning objectives

By the end of this lecture you should be able to understand :

- 计算机 I/O 子系统的组成
- OS 设备管理模块的结构
- I/O 缓冲技术
- 磁盘调度策略
- RAID 技术

第41-42讲 设备管理概述和I/O控制方式

§4.1 I/O Devices

Categories of I/O Devices

- Human readable
- Machine readable
- Communication

Differences in I/O Devices

- Data Transfer Rate(数据传输速率)
- Application(应用)
- Complexity of Control (控制复杂性)
- Data Transfer Unit(数据传输单位)
- Data Representation(数据表示)
- Error Conditions(错误条件)

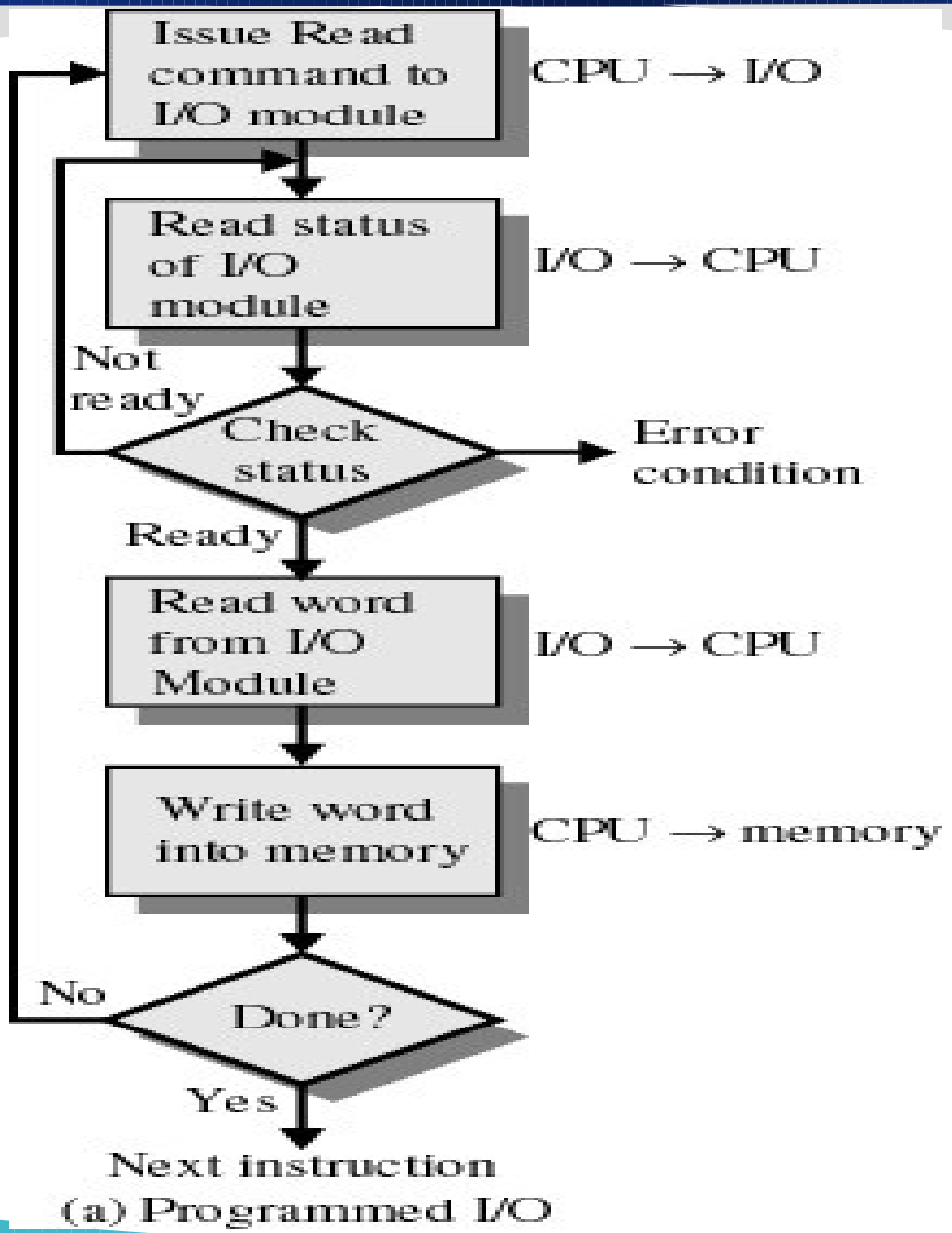
§4.2 Organization Of The I/O Function

Techniques for Performing I/O

- Programmed I/O(程序控制 I/O)
- Interrupt-Driven(中断驱动)
- DMA Control (DMA 控制)

Programmed I/O

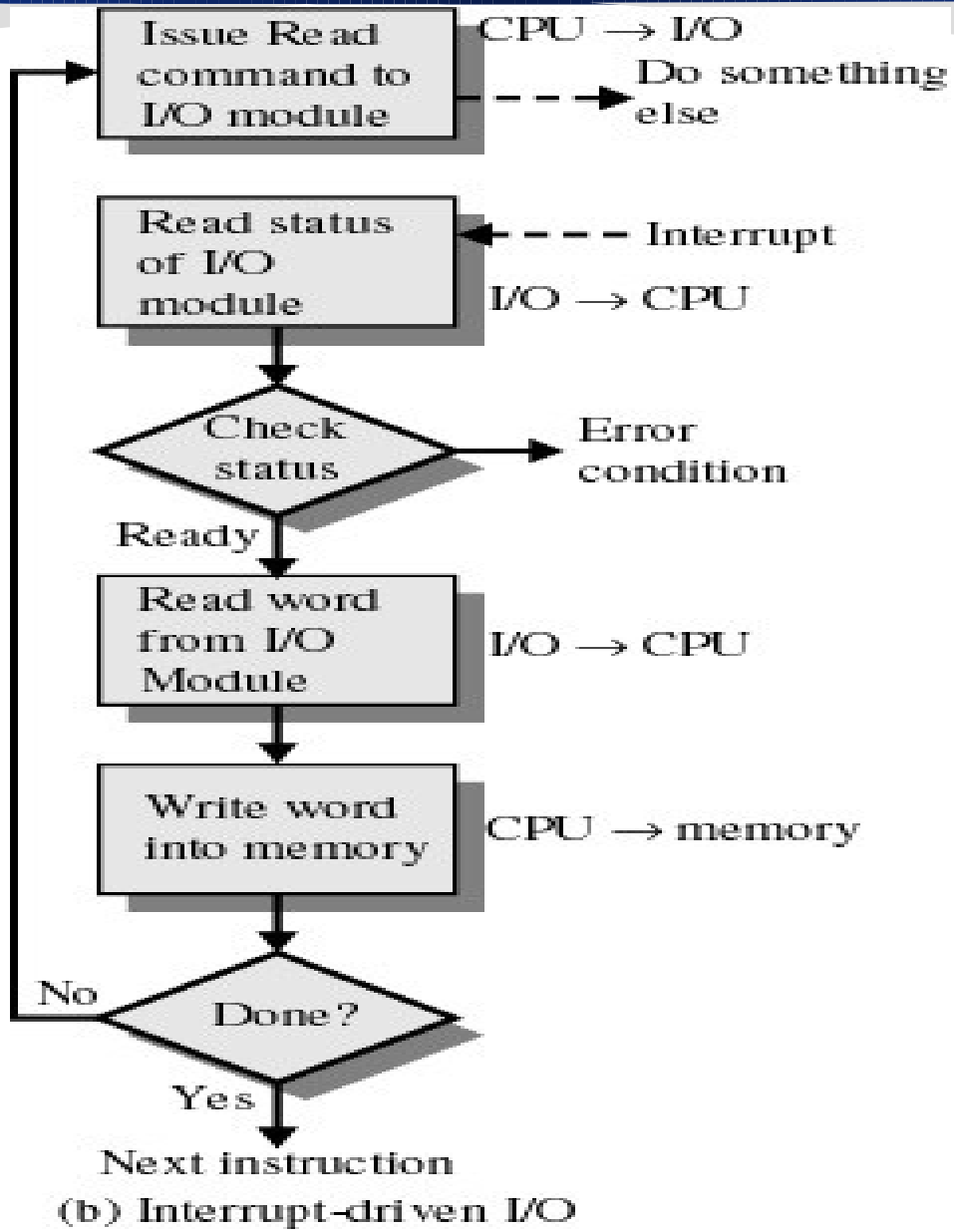
- The processor issues an I/O command to an I/O module, I/O module performs the I/O.
- Sets appropriate bits in the I/O status register.
- No interrupts occur.
- Processor checks status until operation is complete.
- Process is busy-waiting for the operation to complete.



(a) Programmed I/O

Interrupt-Driven I/O

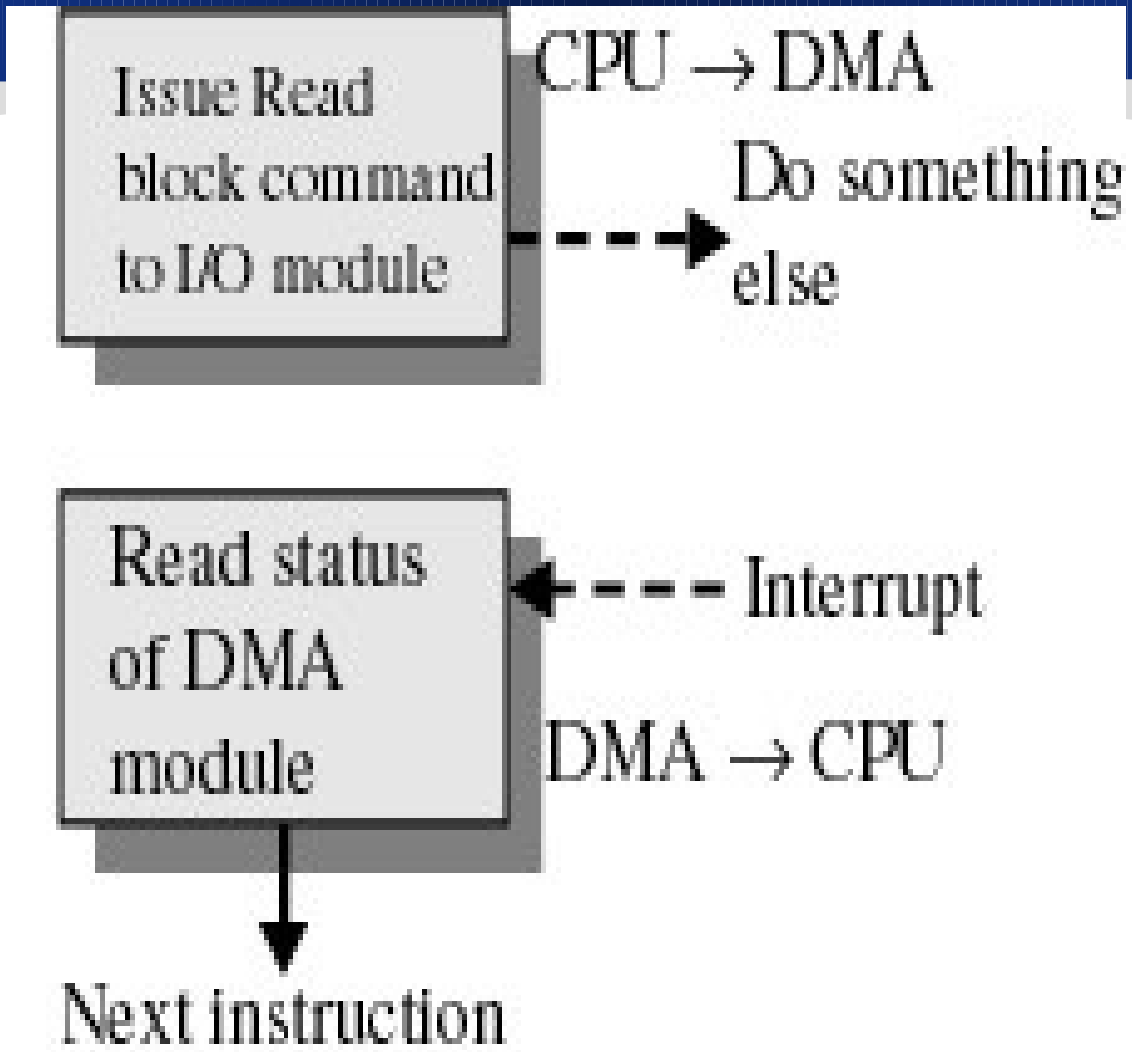
- Processor is interrupted when I/O module ready to exchange data.
- Processor is free to do other work.
- No needless waiting.
- Consumes a lot of processor time because every word read or written passes through the processor.



(b) Interrupt-driven I/O

Direct Memory Access (DMA)

- Transfers a block of data directly to or from memory.
- An interrupt is sent when the task is complete.
- The processor is only involved at the beginning and end of the transfer.



(c) Direct memory access

Direct Memory Access

- Takes control of the system from the CPU to transfer data to and from memory over the system bus.
- Cycle stealing is used to transfer data on the system bus.
- The instruction cycle is suspended so data can be transferred.
- The CPU pauses one bus cycle.
- No interrupts occur
  - Do not save context.

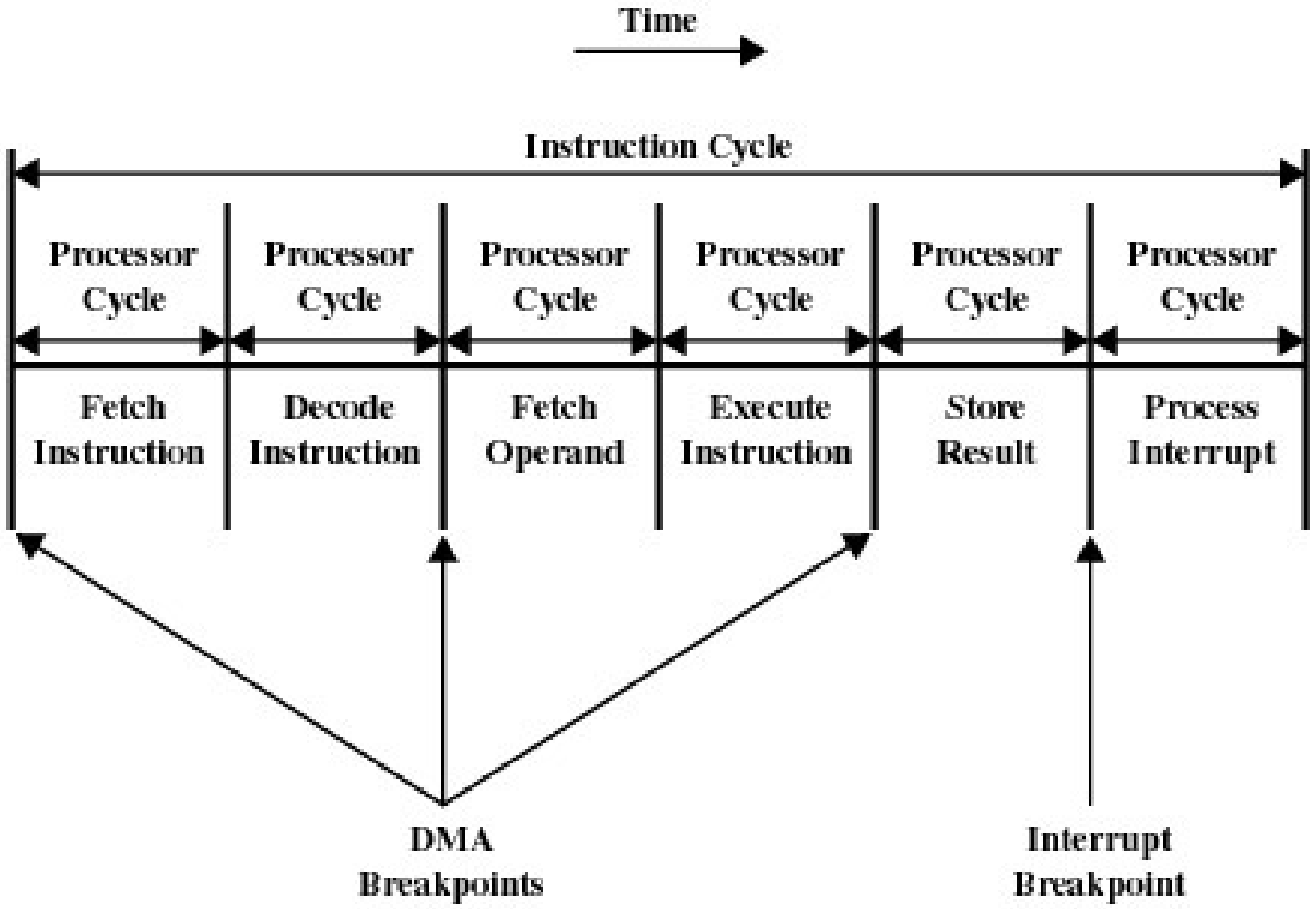


Figure 11.3 DMA and Interrupt Breakpoints During an Instruction Cycle

DMA Logic

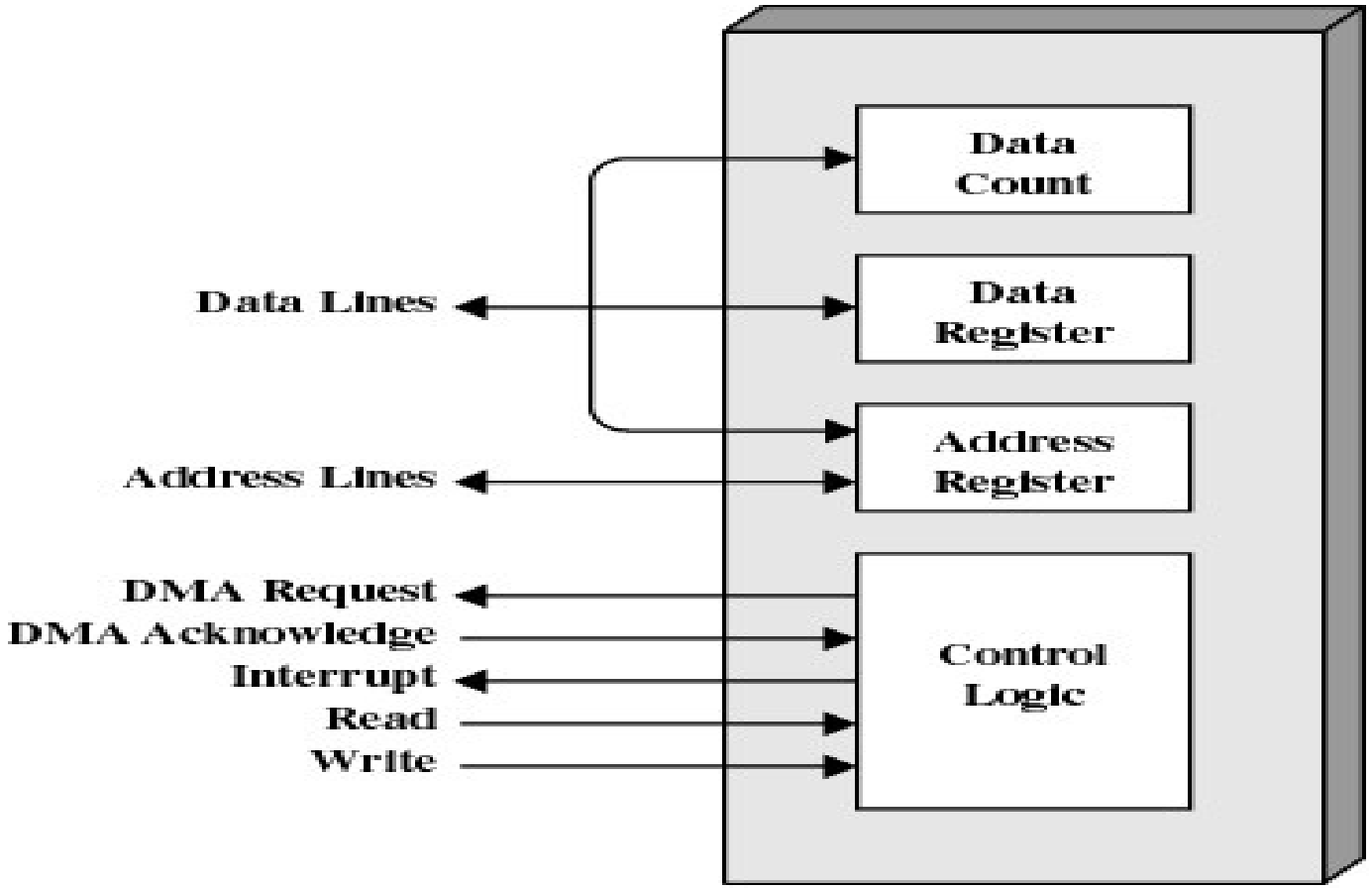


Figure 11.2 Typical DMA Block Diagram

DMA 控制器的工作原理（1）

- 当 CPU 将一块数据从某 I/O 设备读入内存某处或把一块数据从内存某处写入某 I/O 设备时
  - CPU 将把相关*数据块的长度*通过 DMA 的“ Data Lines”脚写入 DMA 的“ *Data Count*”寄存器
  - 把内存起始地址和 I/O 设备地址通过 DMA 的“ Data Lines”脚写入 DMA 的“ *Address Register*”寄存器
  - 通过 DMA 的 “ Read”脚或“ Write”脚向 DMA 发出读或写命令

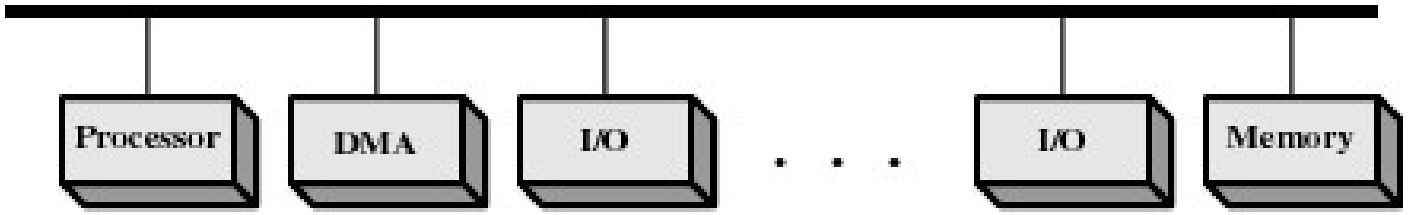
DMA 控制器的工作原理 （2）

- DMA 通过“ DMA Request” 脚 和 “ DMA Acknowledge” 脚控制相关的 I/O 设备
- 通过“ Data Lines” 脚与 I/O 设备以及内存交换数据
- 通过“ Address Lines” 脚寻址内存

DMA 控制器的工作原理 (3)

- 工作完毕， DAM 将通过“ Interrupt” 脚向 CPU 发出中断， CPU 对本次数据传输进行善后处理。
- DMA 将通过窃取 CPU 总线周期的方式获得总线控制权。

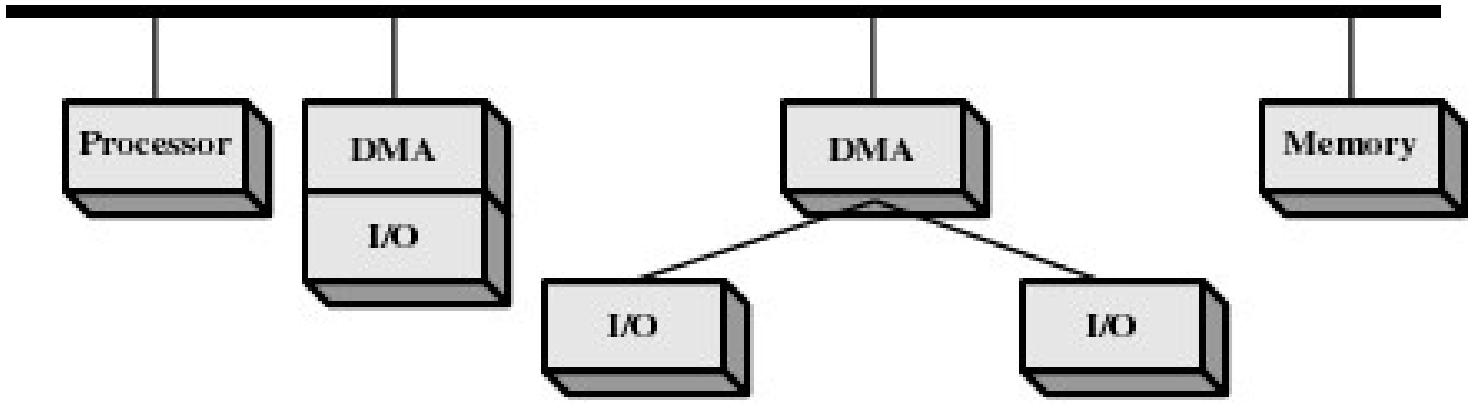
DMA



(a) Single-bus, detached DMA

Figure 11.4 Alternative DMA Configurations

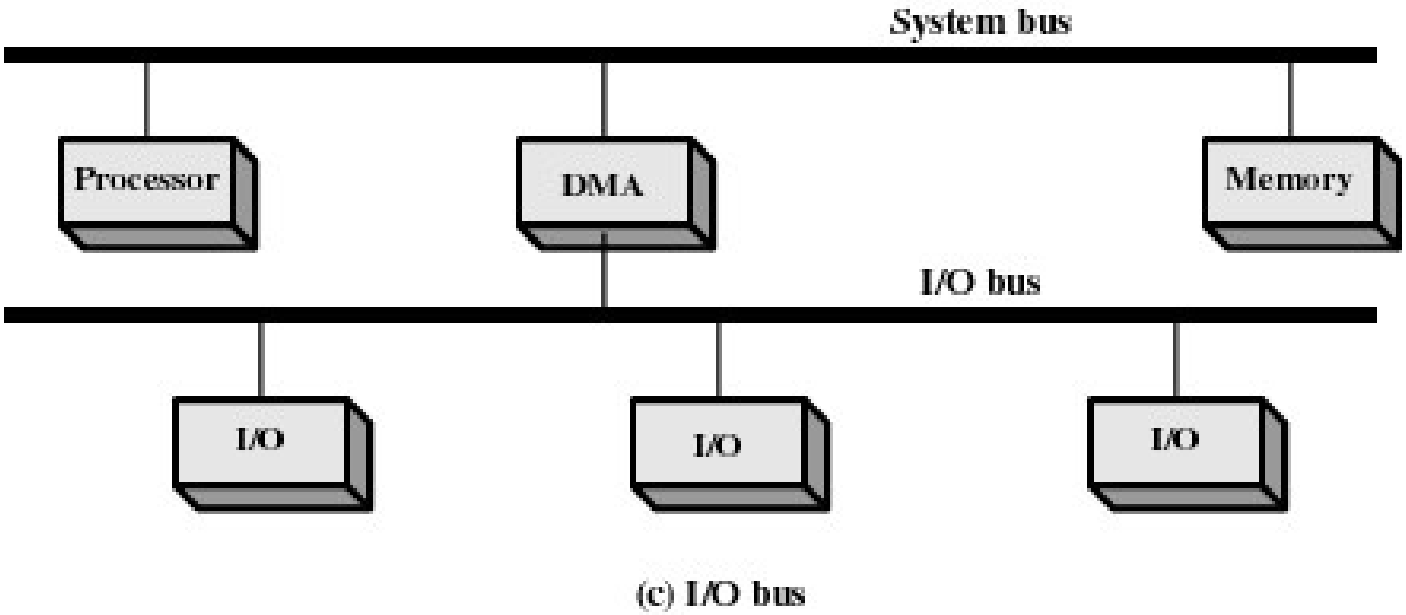
DMA



(b) Single-bus, Integrated DMA-I/O

Figure 11.4 Alternative DMA Configurations

DMA



(c) I/O bus

Figure 11.4 Alternative DMA Configurations

Evolution of the I/O Function

- Processor directly controls a peripheral device.
- Controller or I/O module is added
  - Processor uses *programmed I/O* without interrupts .
  - Processor does not need to handle details of external devices.

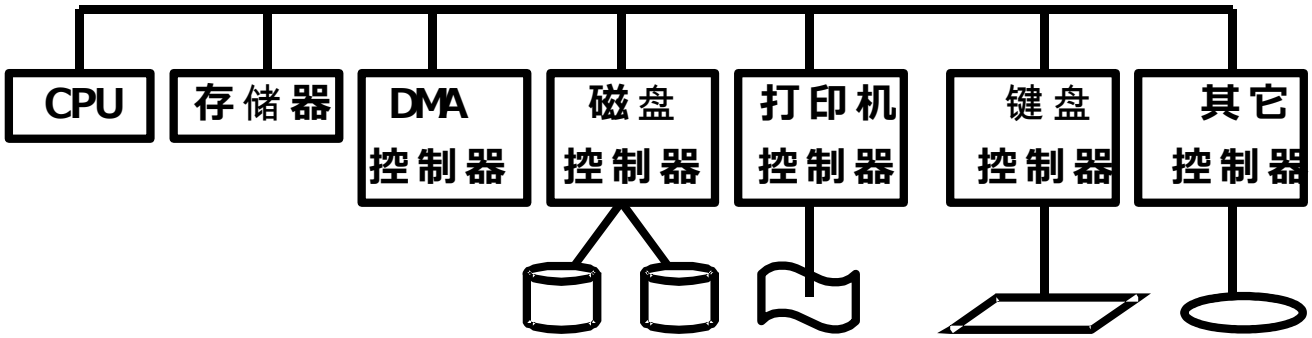
Evolution of the I/O Function

- **Controller or I/O module with *interrupts***
  - **Processor does not spend time waiting for an I/O operation to be performed.**
- ***Direct Memory Access***
  - **Blocks of data are moved into memory without involving the processor.**
  - **Processor involved at beginning and end only.**

Evolution of the I/O Function

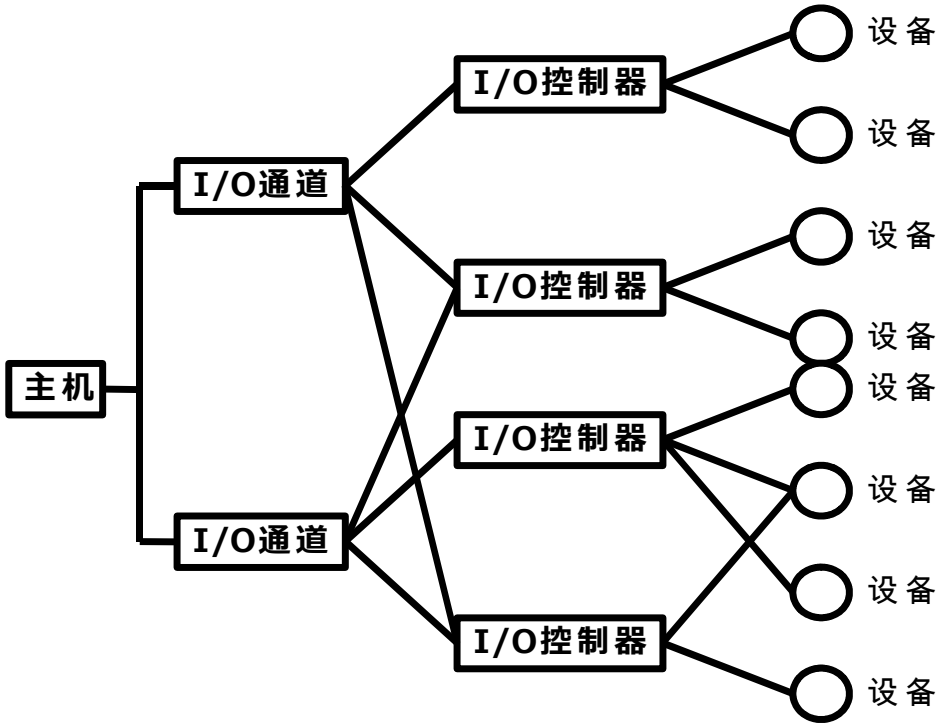
- I/O module is a separate processor： *I/O channel.*
- I/O processor
  - I/O module has its own local memory.
  - It’s a computer in its own right.

计算机 I/O 子系统的组成（1）



微型和小型计算机 I/O子系统的组成

计算机 I/O 子系统的组成（2）



中型和大型计算机 I/O子系统的组成

§4.3 OS 设备管理模块的结构

Operating System Design Issues

Efficiency

- Most I/O devices extremely slow compared to main memory.
- Use of multiprogramming allows for some processes to be waiting on I/O while another process executes.
- I/O cannot keep up with processor speed.
- Swapping is used to bring in additional Ready processes which is an I/O operation.

Operating System Design Issues

Generality

- Desirable to handle all I/O devices in a uniform manner.
- Hide most of the details of device I/O in lower-level routines so that processes and upper levels see devices in general terms such as read, write, open, close, lock, unlock.

Device Independence （设备无关性）

即，应用软件所引用的用于实现 I/O 操作的设备与计算机 I/O 子系统中实际安装的设备没有固定的联系。

Logical I/O Device

应用软件引用的用于实现 I/O 操作的设备。

从应用软件的角度看，逻辑 I/O 设备是一类具有相同或相似属性的物理 I/O 设备的抽象。

逻辑 I/O 设备的分类

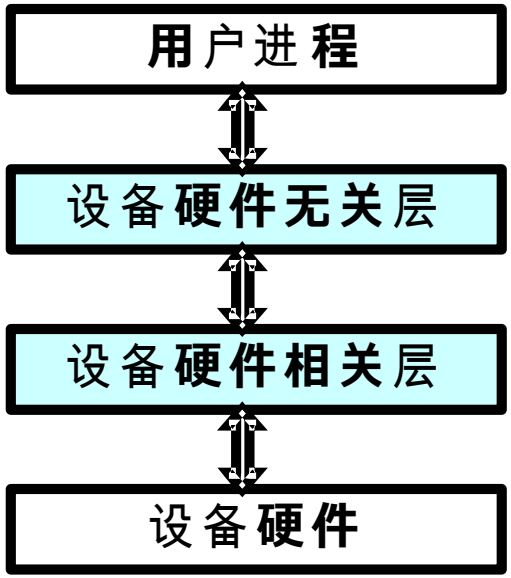
- **字符设备**
  - 也叫面向流的设备 (Stream-Oriented Device)；
  - 应用软件以*字符*为单位读写此类逻辑 I/O 设备。
- **块设备**
  - 也叫面向块的设备 (Block-Oriented Device)；
  - 应用软件以*块*为单位读写此类逻辑 I/O 设备。

OS 设备管理模块的分层结构

- 为了实现“通用性”这一设计目标，大多数 OS 的设备管理模块采用*分层结构*。
- 在分层结构中，上层可以屏蔽下层的实现细节

OS 设备管理模块的分层结构模型

- **典型的两层结构：**
  - 设备硬件无关层，实现*设备映射功能*，把逻辑 I/O 设备映射到物理 I/O 设备
  - 设备硬件相关层，实现*设备驱动功能*，控制物理 I/O 设备以便完成实际的 I/O 操作



OS设备管理模块的分层结构模型

电子科技大学
University of Electronic Science and Technology of China

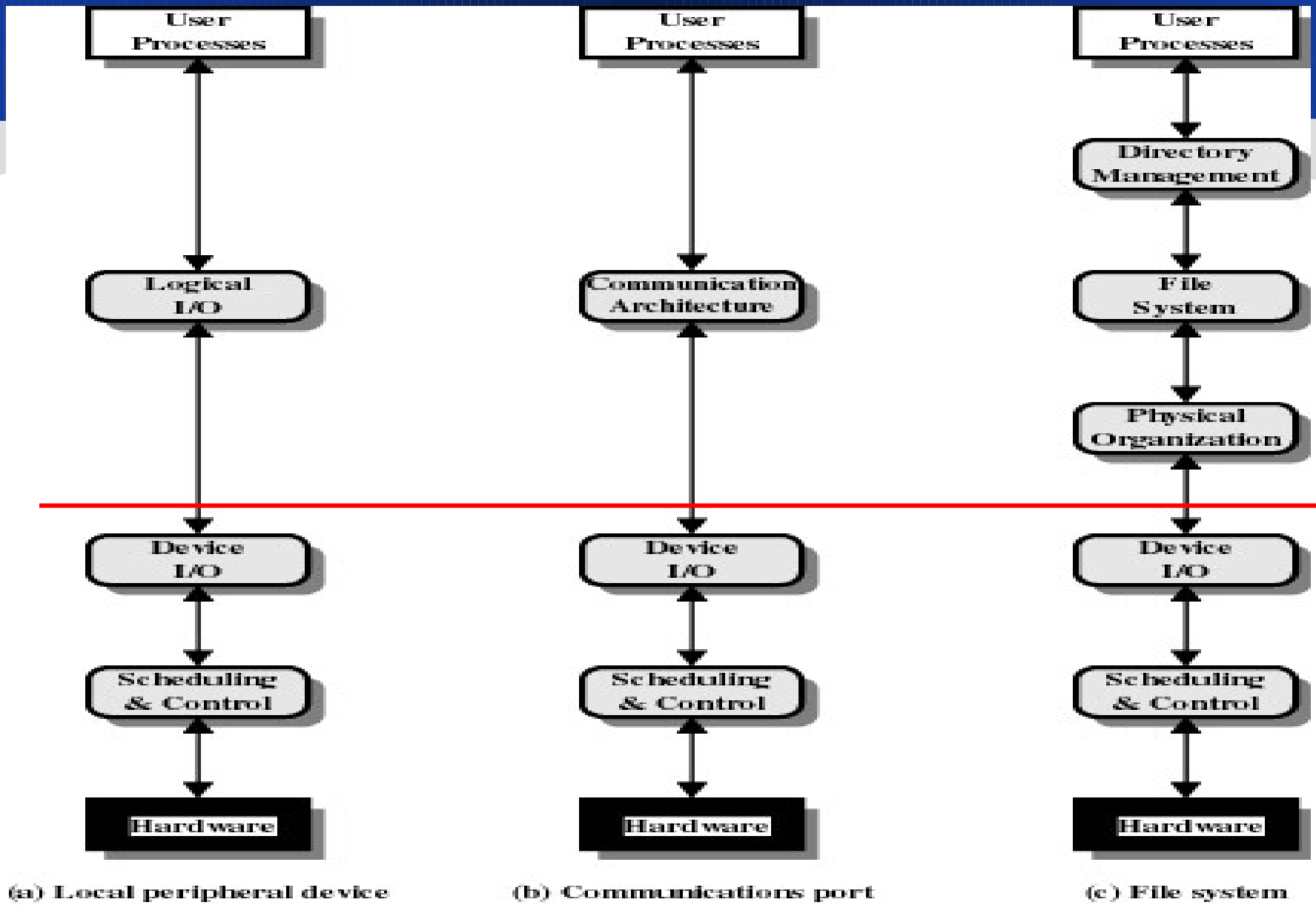


Figure 11.5 A Model of I/O Organization