

计算机组成和体系结构

第九讲

四川大学网络空间安全学院

2020年4月26日

封面来自pickpik.com

版权声明

课件中所使用的图片、视频等资源版权归原作者所有。

课件原创内容采用 [创作共用署名—非商业使用—相同方式共享4.0国际版许可证\(Creative Commons BY-NC-SA 4.0 International License\)](#) 授权使用。

Copyright@四川大学网络空间安全学院计算机组成与体系结构课程组，2020



上期内容回顾

- 常见的存储器类型及特点
- 存储器层次结构及性能评价
- 高速缓存
 - 缓存的映射策略：直接映射、全相联映射、组相联映射
 - 缓存的替换策略、写策略
 - 指令和数据缓存、多级缓存
- 虚拟内存
 - 分页内存管理：页表，地址转换，TLB
 - 内存碎片和分段内存管理
- 存储器实例：Pentium处理器

本期学习目标

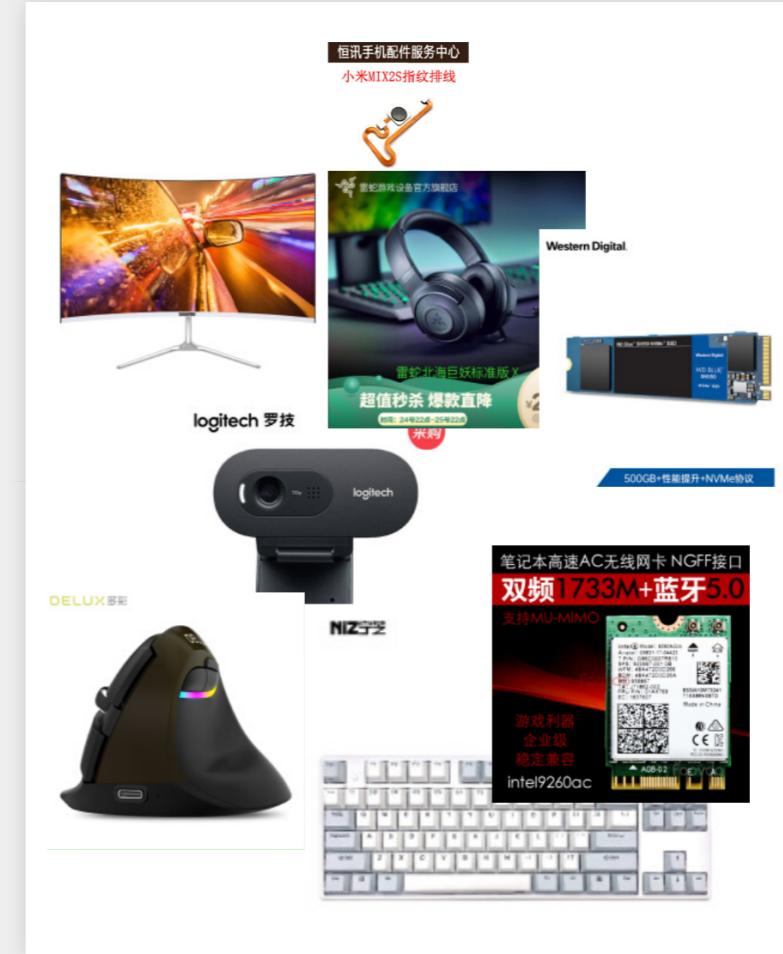
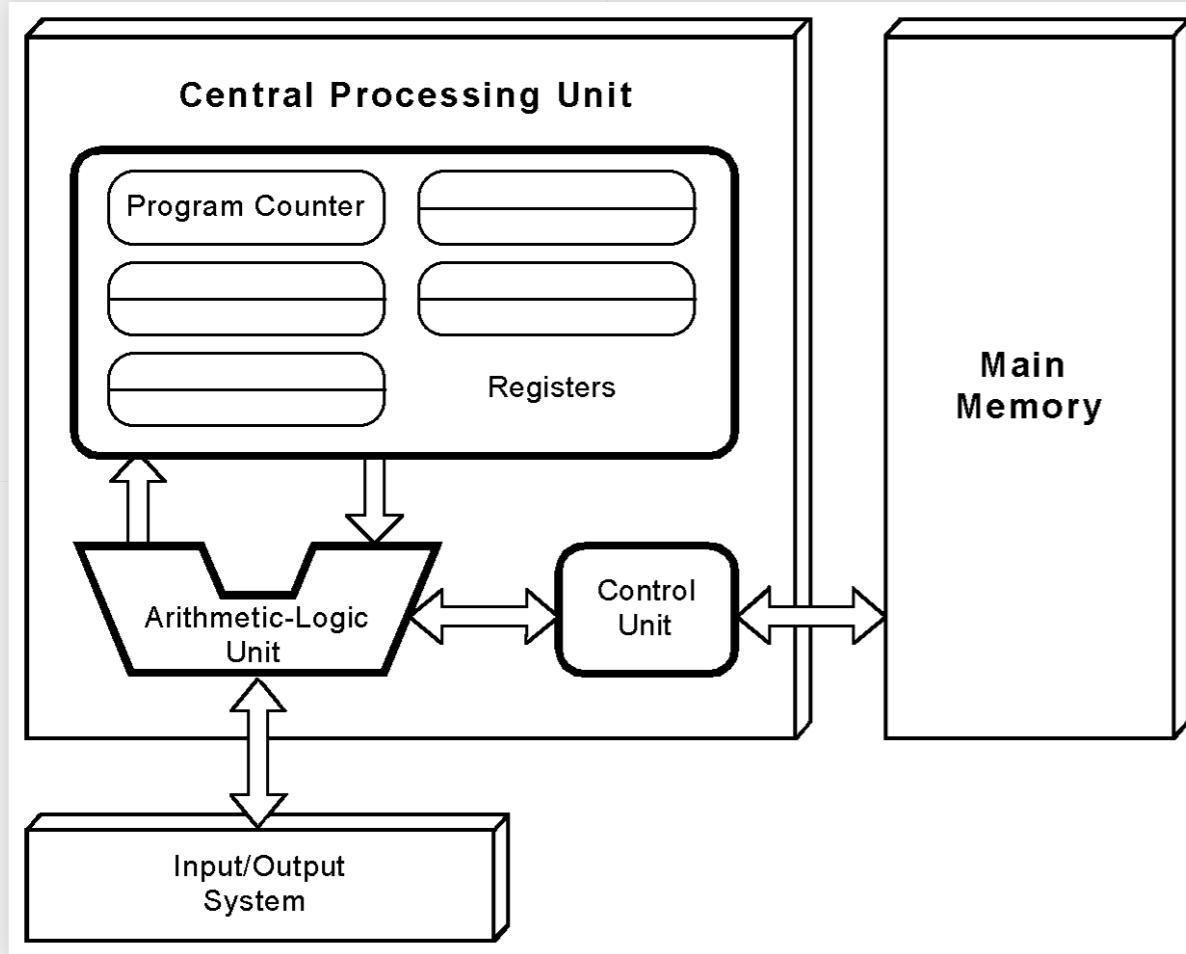
- 冯诺依曼模型的第三部分：输入输出系统
 - 子系统性能与计算机系统整体性能的关系：阿姆达尔定律
- I/O子系统组成和体系结构
 - I/O子系统示例
 - 如何控制I/O设备：五种控制方法、字符和块I/O、总线控制与时序图
- 数据传输模式：并行传输和串行传输

中英文缩写对照表

英文缩写	英文全称	中文全称
I/O	Input/Output	输入输出
DMA	Direct Memory Access	直接内存访问
RDMA	Remote Direct Memory Access	远程直接内存访问
HDD	Hard Disk Drive	硬盘
SSD	Solid State Drive	固态硬盘
RAID	Redundant Array of Independent Disks	独立磁盘冗余阵列

I/O 1: 输入输出子系统与系统性能

输入输出子系统



输入输出子系统

输入输出子系统

- I/O子系统是计算机系统的重要组成部分

输入输出子系统

- I/O子系统是计算机系统的重要组成部分
- I/O子系统的性能对计算机的整体性能有重要的影响

输入输出子系统

- I/O子系统是计算机系统的重要组成部分
- I/O子系统的性能对计算机的整体性能有重要的影响



输入输出子系统

- I/O子系统是计算机系统的重要组成部分
- I/O子系统的性能对计算机的整体性能有重要的影响



量化局部优化对整体性能的影响

阿姆达尔定律(Amdahl's Law)

$$S = \frac{1}{(1 - f) + \frac{f}{k}}$$

- f : 优化的部分占总体性能的比例
- k : 优化的部分的加速比
- S : 系统的整体加速比

阿姆达尔定律定量描述了提高局部性能对整体性能的影响

阿姆达尔定律示例

假设视频会议应用30%的时间用于CPU解码，70%的时间用于等待网络传输视频数据，下面两种方法分别对应的加速比是多少：

1. 花¥1500升级CPU，性能提高30%
2. 花169升级宽带，从100Mbps升级到200Mbps

阿姆达尔定律示例

假设视频会议应用30%的时间用于CPU解码，70%的时间用于等待网络传输视频数据，下面两种方法分别对应的加速比是多少：

1. 花¥1500升级CPU，性能提高30%
 2. 花169升级宽带，从100Mbps升级到200Mbps
- 升级CPU: $f = 0.3, k = 1.3,$

$$S = \frac{1}{0.7 + \frac{0.3}{1.3}} \approx 1.074$$

阿姆达尔定律示例

假设视频会议应用30%的时间用于CPU解码，70%的时间用于等待网络传输视频数据，下面两种方法分别对应的加速比是多少：

1. 花¥1500升级CPU，性能提高30%
2. 花169升级宽带，从100Mbps升级到200Mbps

- 升级CPU: $f = 0.3, k = 1.3,$

$$S = \frac{1}{0.7 + \frac{0.3}{1.3}} \approx 1.074$$

- 升级宽带: $f = 0.7, k = 2,$

$$S = \frac{1}{0.3 + \frac{0.7}{2}} \approx 1.538$$

阿姆达尔定律的启示

$$S = \frac{1}{(1-f) + \frac{f}{k}}$$

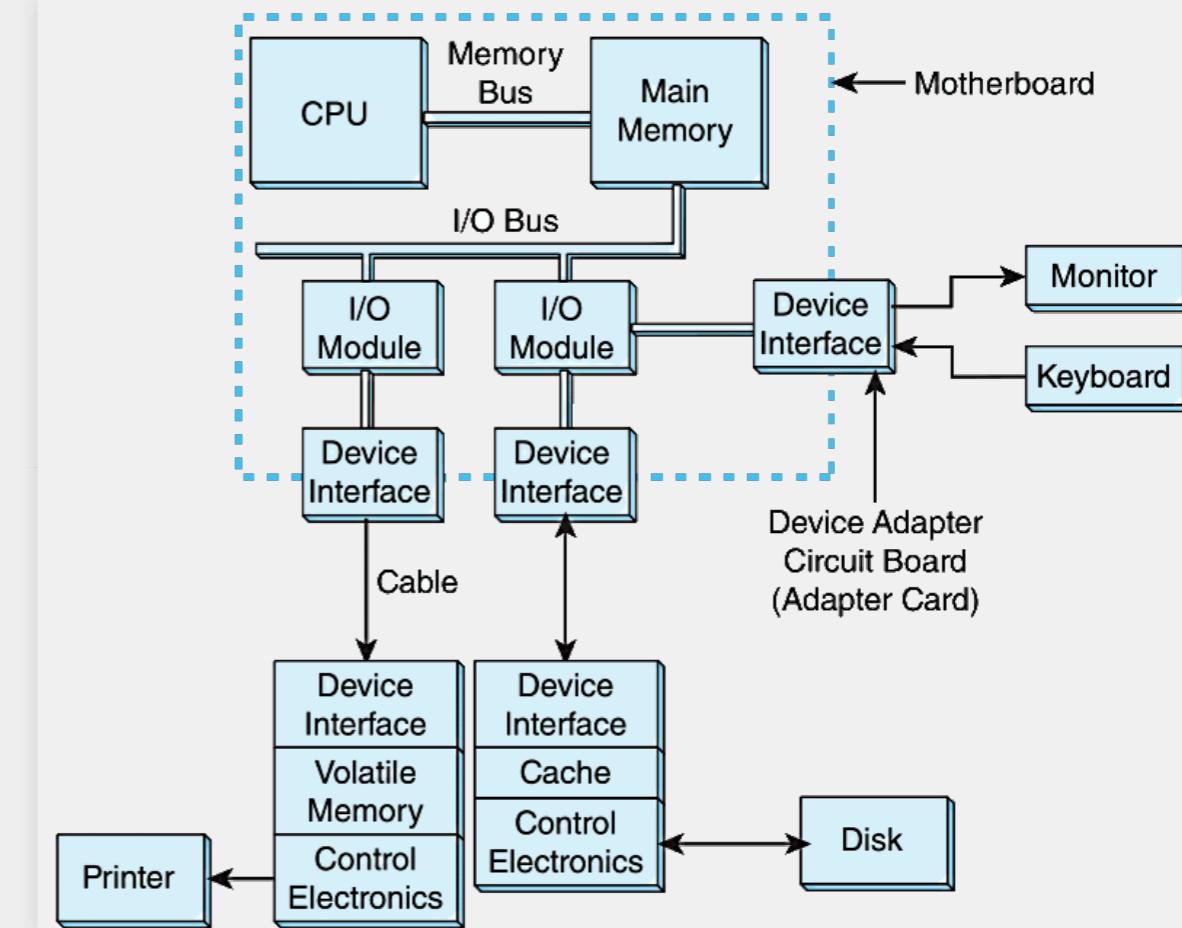
$$\frac{\partial S}{\partial f} = \frac{1 - \frac{1}{k}}{(1-f + \frac{f}{k})^2} > 0, \quad \frac{\partial S}{\partial k} = \frac{f}{(1-f + \frac{f}{k})^2 k^2} > 0$$

- 性能提高的部分占总体性能比例越高，对于整体性能的提高作用越明显
- 性能提高的部分提高的性能越高，对于整体性能的提高作用越明显
- 对系统优化的启示：优先优化系统中占比最高的部分

I/O 2: I/O子系统体系结构

I/O子系统组成

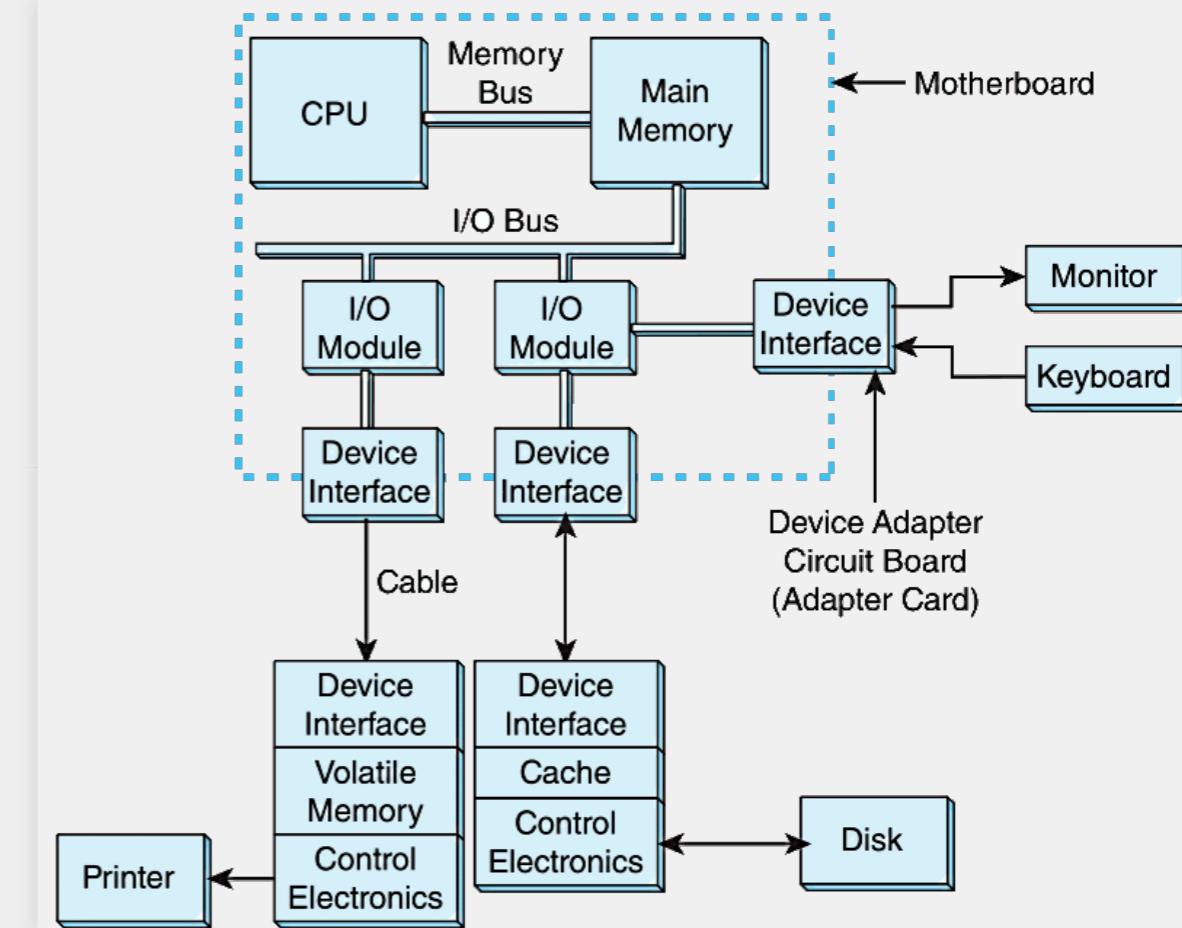
现代计算机的I/O子系统通常具有下面一些组件：



I/O子系统组成

现代计算机的I/O子系统通常具有下面一些组件：

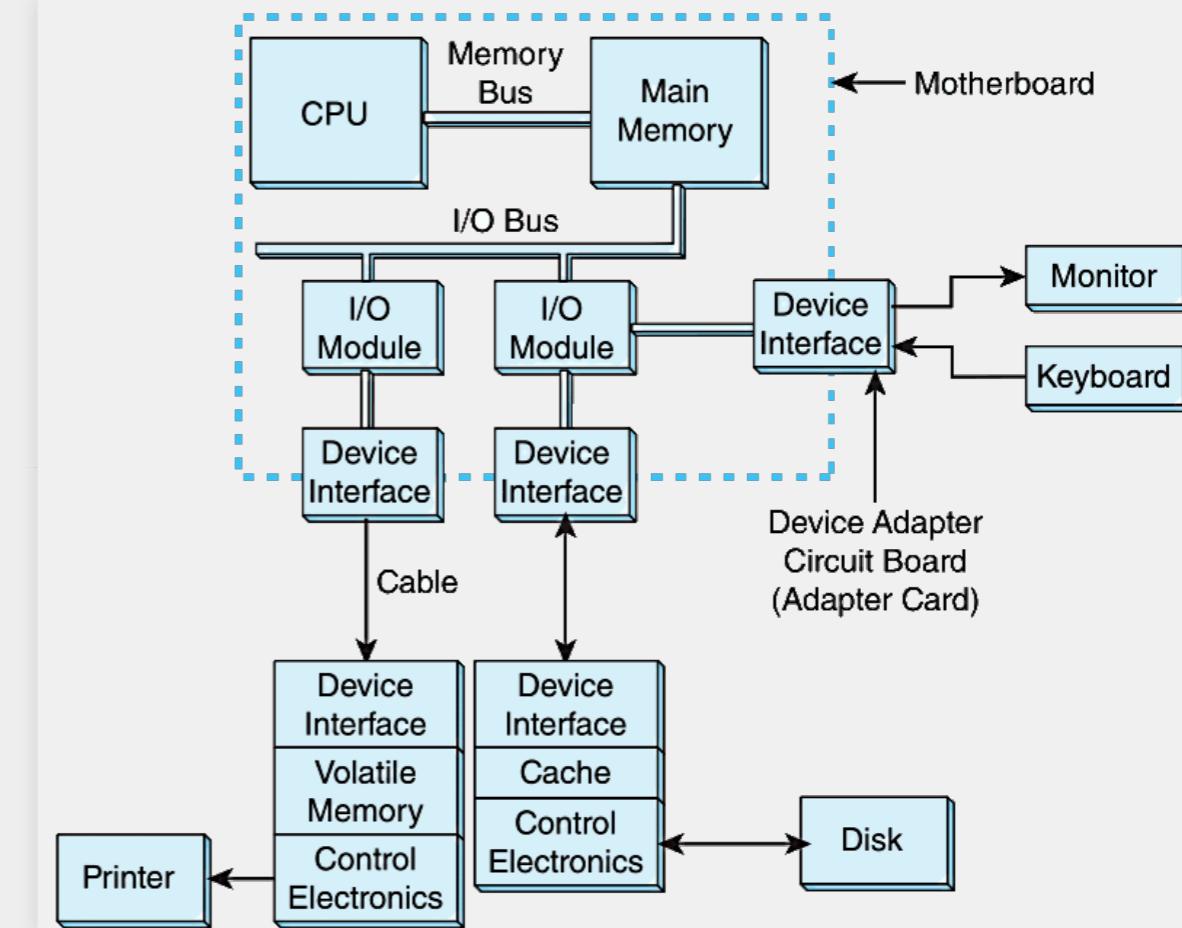
- 专用于I/O操作的主存数据块



I/O子系统组成

现代计算机的I/O子系统通常具有下面一些组件：

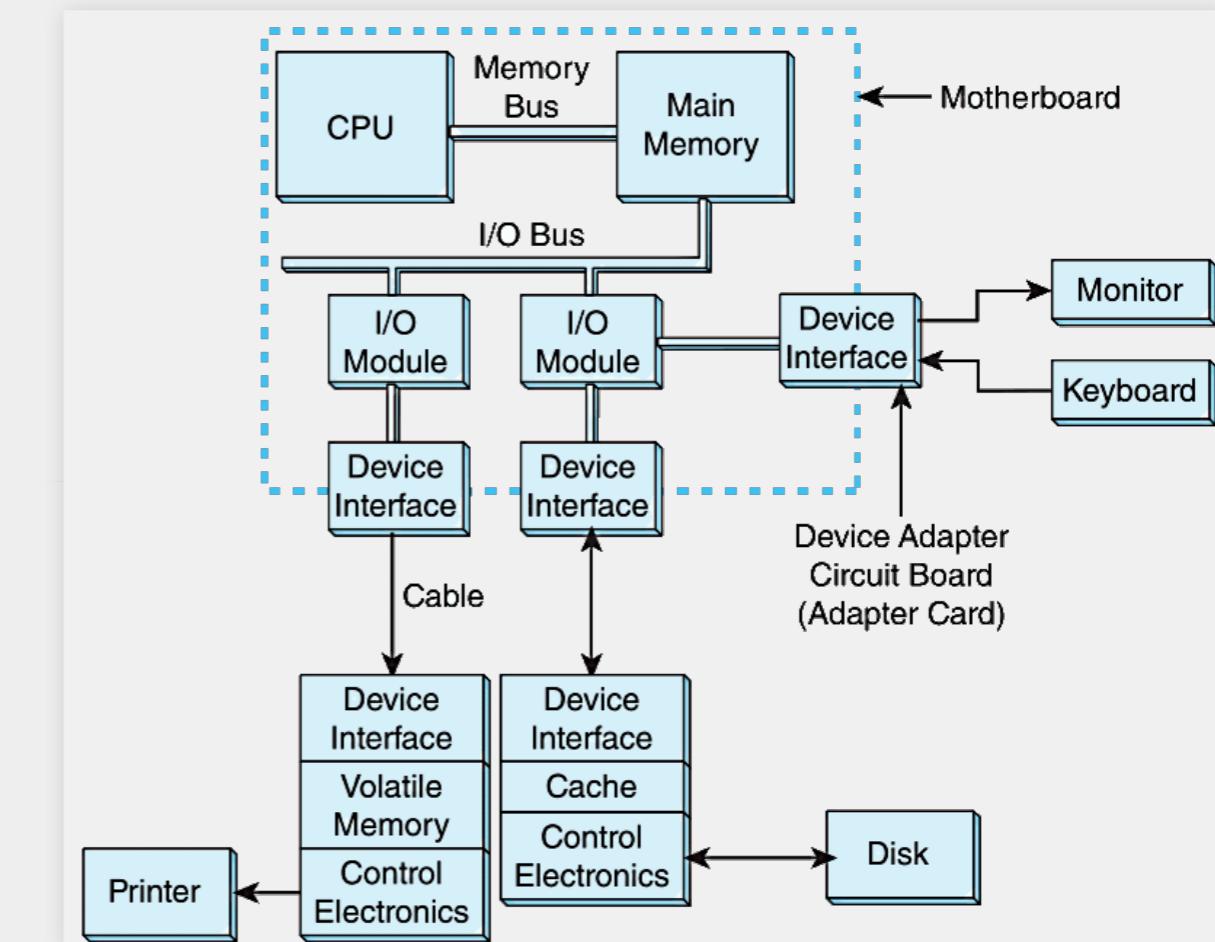
- 专用于I/O操作的主存数据块
- 用于数据传输的总线



I/O子系统组成

现代计算机的I/O子系统通常具有下面一些组件：

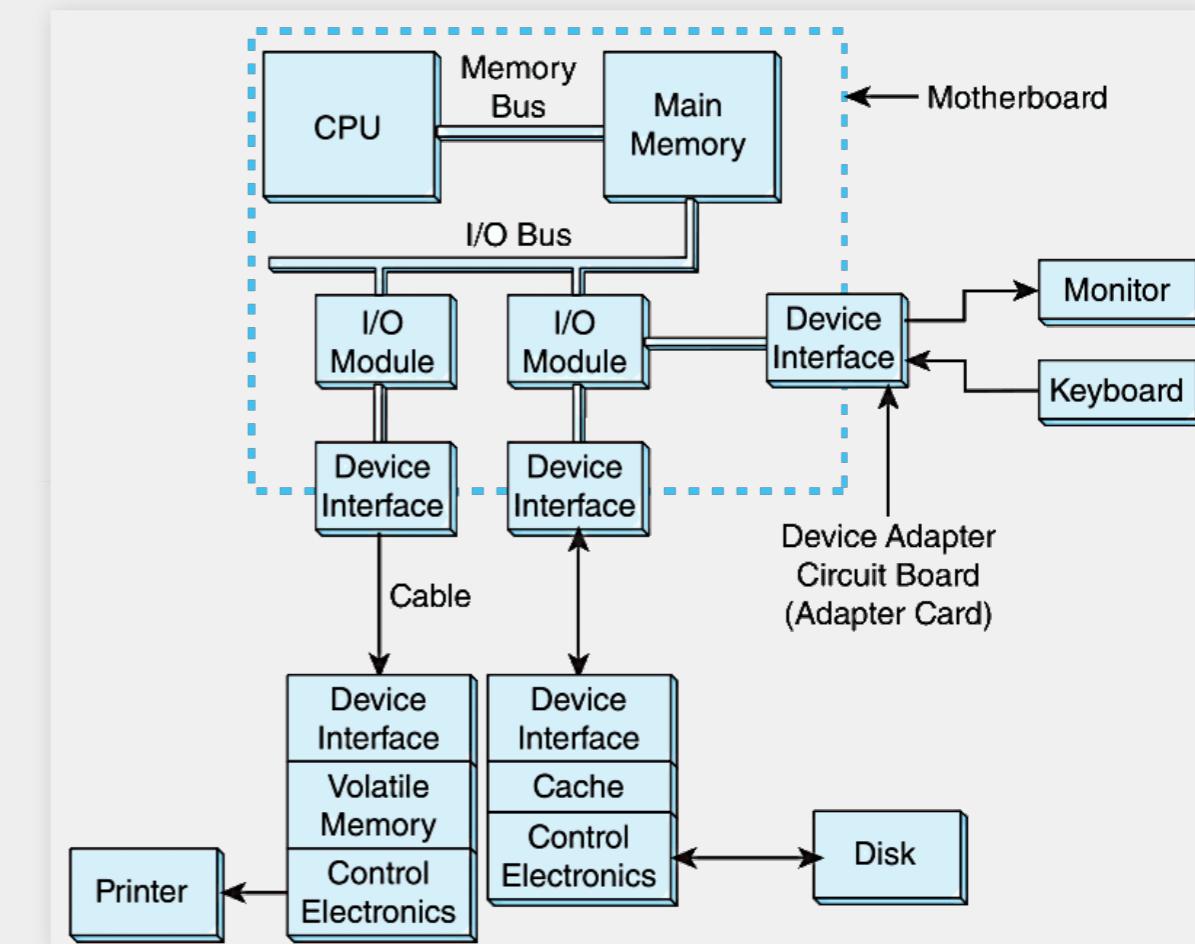
- 专用于I/O操作的主存数据块
- 用于数据传输的总线
- 主机和外围设备中的控制模块



I/O子系统组成

现代计算机的I/O子系统通常具有下面一些组件：

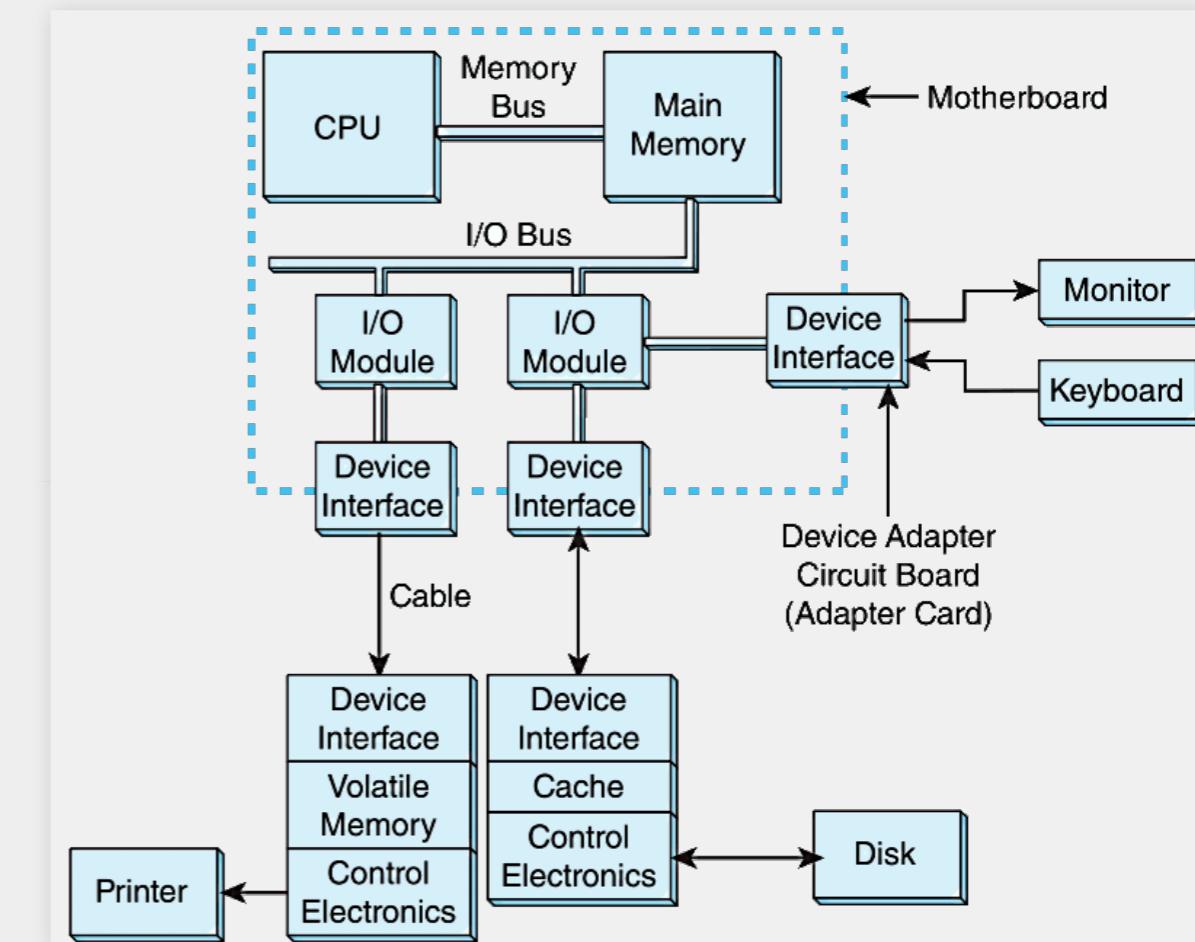
- 专用于I/O操作的主存数据块
- 用于数据传输的总线
- 主机和外围设备中的控制模块
- 连接外部设备的接口



I/O子系统组成

现代计算机的I/O子系统通常具有下面一些组件：

- 专用于I/O操作的主存数据块
- 用于数据传输的总线
- 主机和外围设备中的控制模块
- 连接外部设备的接口
- 主机和外围设备中进行数据通信的连接线



I/O控制方法

有五种常见的I/O控制方法

I/O控制方法

有五种常见的I/O控制方法

- 程序控制I/O(Programmed I/O): 用特殊寄存器保存I/O设备的状态, CPU轮询寄存器状态决定是否需要处理I/O事件

I/O控制方法

有五种常见的I/O控制方法

- 程序控制I/O(Programmed I/O): 用特殊寄存器保存I/O设备的状态，CPU轮询寄存器状态决定是否需要处理I/O事件
- 中断驱动I/O(Interrupt-driven I/O): I/O设备触发系统中断，由中断处理程序处理I/O事件

I/O控制方法

有五种常见的I/O控制方法

- 程序控制I/O(Programmed I/O): 用特殊寄存器保存I/O设备的状态，CPU轮询寄存器状态决定是否需要处理I/O事件
- 中断驱动I/O(Interrupt-driven I/O): I/O设备触发系统中断，由中断处理程序处理I/O事件
- 内存映射I/O(Memory-mapped I/O): 设备中的数据映射为内存地址空间的一部分

I/O控制方法

有五种常见的I/O控制方法

- 程序控制I/O(Programmed I/O): 用特殊寄存器保存I/O设备的状态，CPU轮询寄存器状态决定是否需要处理I/O事件
- 中断驱动I/O(Interrupt-driven I/O): I/O设备触发系统中断，由中断处理程序处理I/O事件
- 内存映射I/O(Memory-mapped I/O): 设备中的数据映射为内存地址空间的一部分
- 直接内存访问I/O(Direct Memory Access, DMA): 采用专用芯片处理I/O设备和主存之间的数据传输

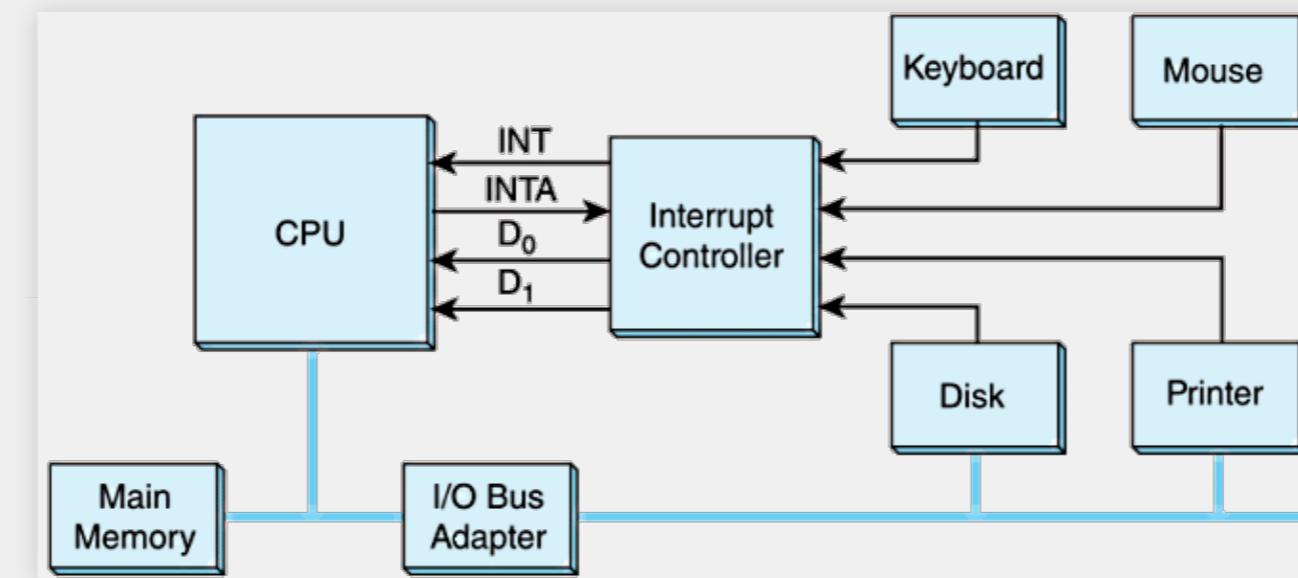
I/O控制方法

有五种常见的I/O控制方法

- 程序控制I/O(Programmed I/O): 用特殊寄存器保存I/O设备的状态，CPU轮询寄存器状态决定是否需要处理I/O事件
- 中断驱动I/O(Interrupt-driven I/O): I/O设备触发系统中断，由中断处理程序处理I/O事件
- 内存映射I/O(Memory-mapped I/O): 设备中的数据映射为内存地址空间的一部分
- 直接内存访问I/O(Direct Memory Access, DMA): 采用专用芯片处理I/O设备和主存之间的数据传输
- 通道I/O(Channel I/O): 采用专用芯片，将多个I/O路径合并为I/O通道

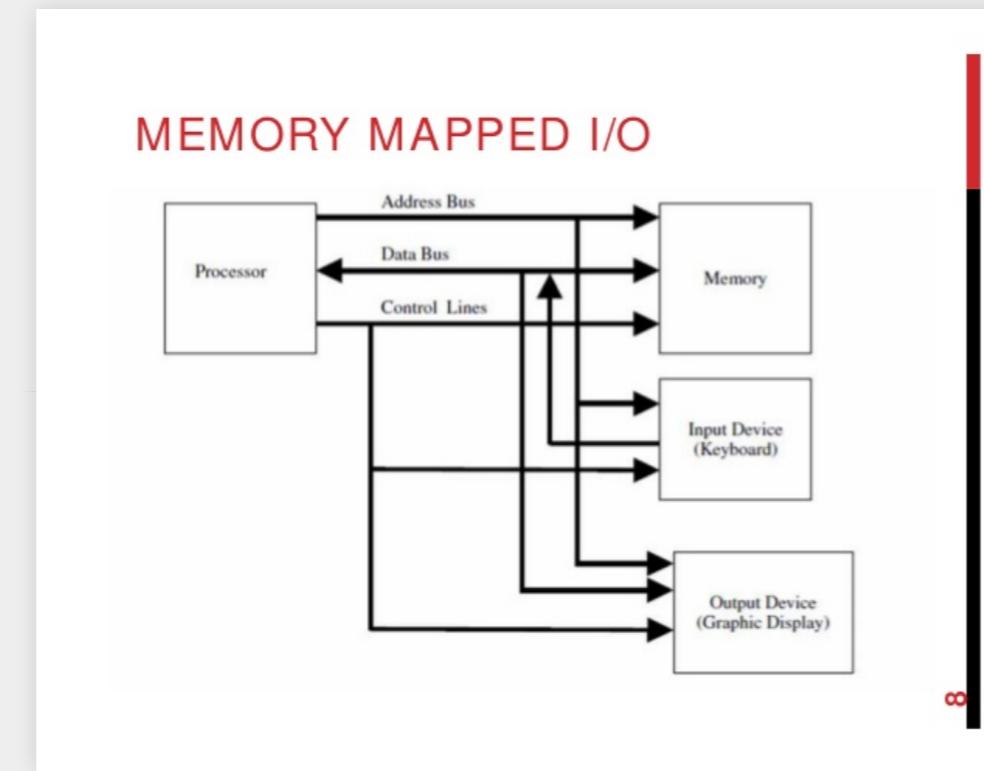
中断驱动I/O

- 多个I/O设备连接到中断处理器，当发生I/O事件时，中断处理器触发中断请求(INT)
- CPU准备好处理中断后回复中断应答(INTA)
- 中断处理器将事件相关信息发送给CPU



内存映射I/O

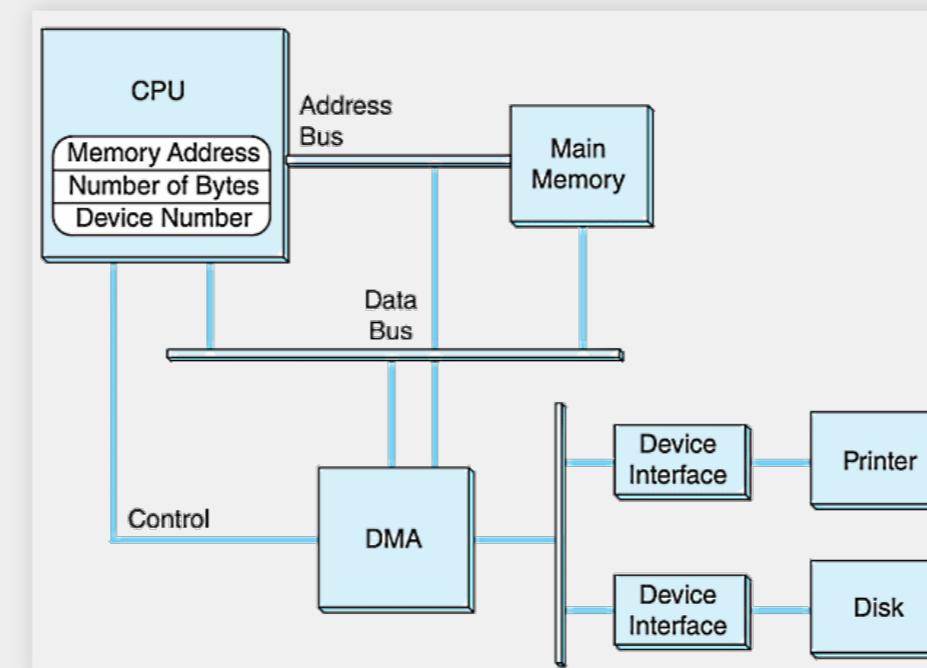
- I/O设备与内存共享地址空间
- I/O设备内部的存储映射到内存的地址空间
- CPU通过访问内存的方式直接访问I/O设备的内部存储



图片来源: <https://www.slideshare.com>

直接内存访问

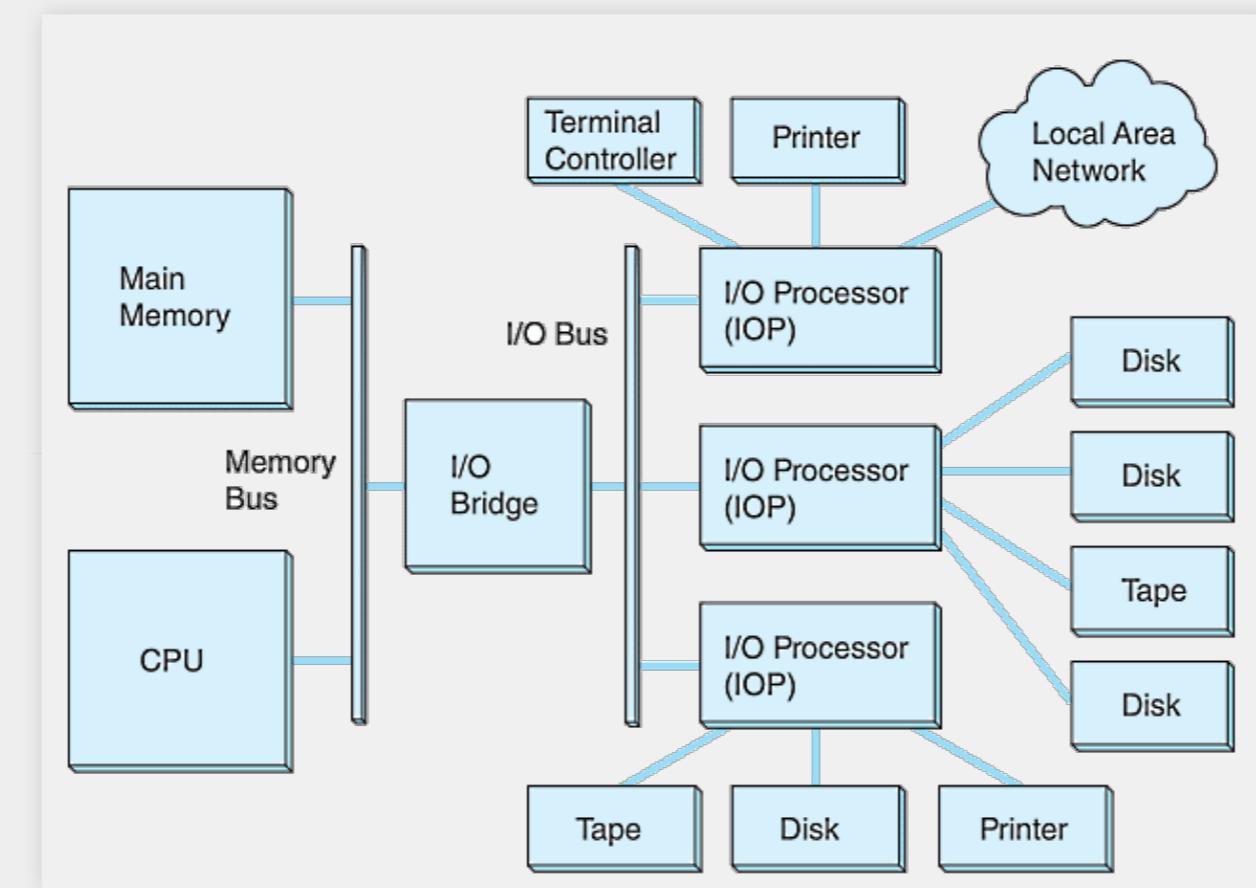
- 将数据从I/O设备中传输到内存中的工作采用专用DMA处理器完成
- CPU不需要执行I/O数据传输的指令，需要指定数据传输的信息（例如数据源和数据目的地、数据大小等）
- CPU和DMA共享总线，如何共享：独占式共享、交替共享、周期窃取



通道控制I/O

- 将多个I/O设备(称为I/O路径)合并为一个I/O通道(channel)，每个通道由一个或多个输入输出处理器(I/OP)管理
- 合并的方法：对于慢速设备，采用多路复用通道(multiplexor)；对于快速设备，采用选择器通道(selector)
- I/O通道之间可以采用I/O桥直接通信
- I/OP和DMA处理器的主要区别是智能程度：I/OP可完成更复杂的处理工作

通道控制I/O示例



I/O的数据传输

- 字符I/O (Character I/O): 按字符为单位进行I/O操作
 - 通常采用中断驱动I/O处理
 - 例子: 键盘
- 块I/O (Block I/O): 按数据块为单位进行I/O操作
 - 通常采用DMA或者通道
 - 例子: 磁盘等存储介质

I/O总线

- 在体系结构中，I/O和内存采用同样的总线

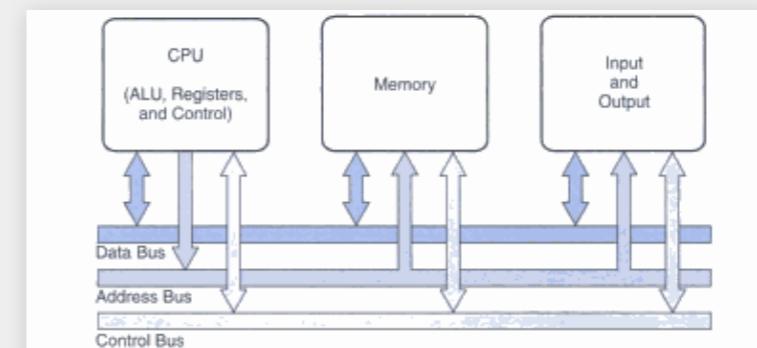


FIGURE 1.5 The Modified von Neumann Architecture, Adding a System Bus

I/O总线

- 在体系结构中，I/O和内存采用同样的总线
- 实际上通常将内存总线和I/O总线分开
 - 内存总线通常是同步(synchronous)的，每个时钟周期都可以进行内存访问
 - I/O总线通常是异步(asynchronous)的

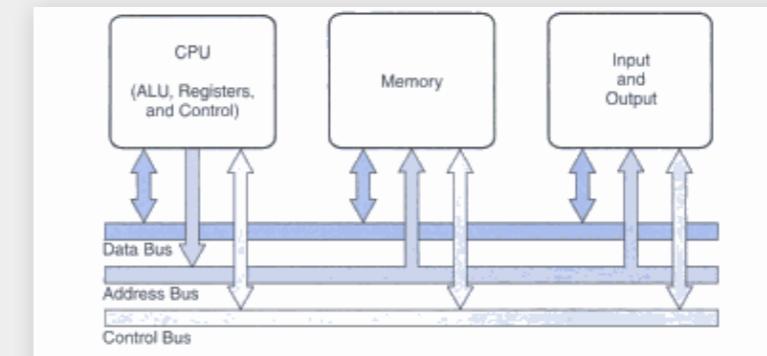
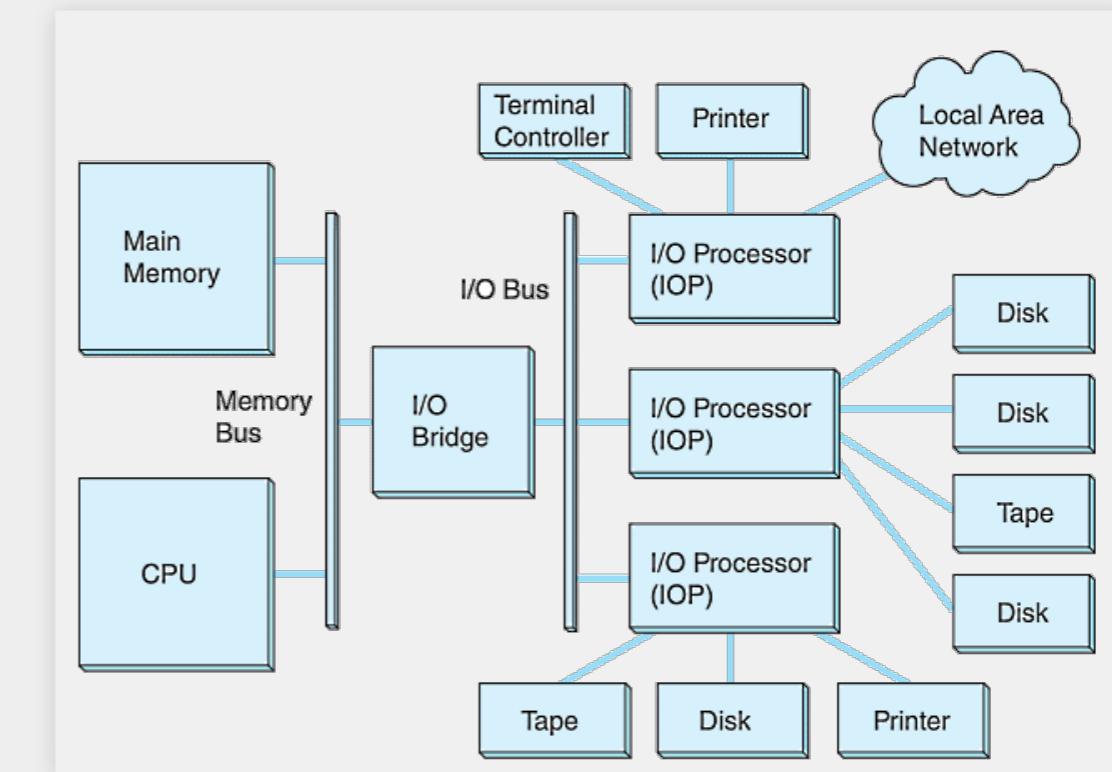
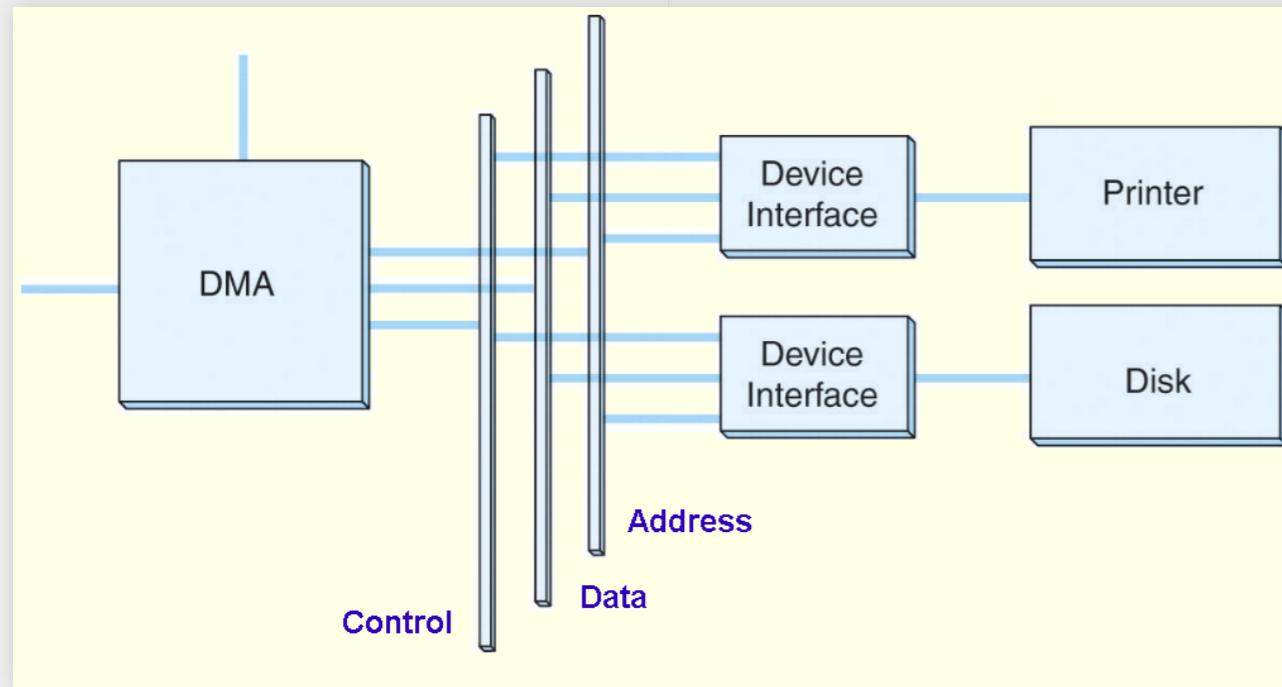
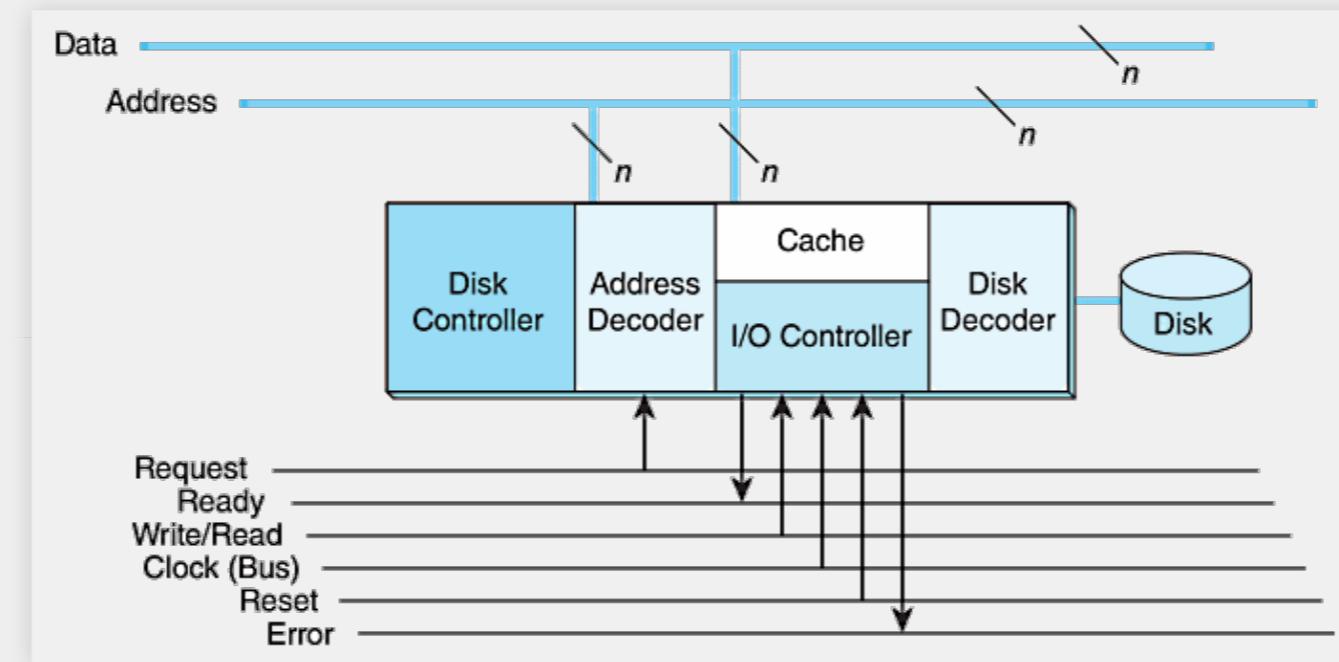


FIGURE 1.5 The Modified von Neumann Architecture, Adding a System Bus



I/O总线管理示例



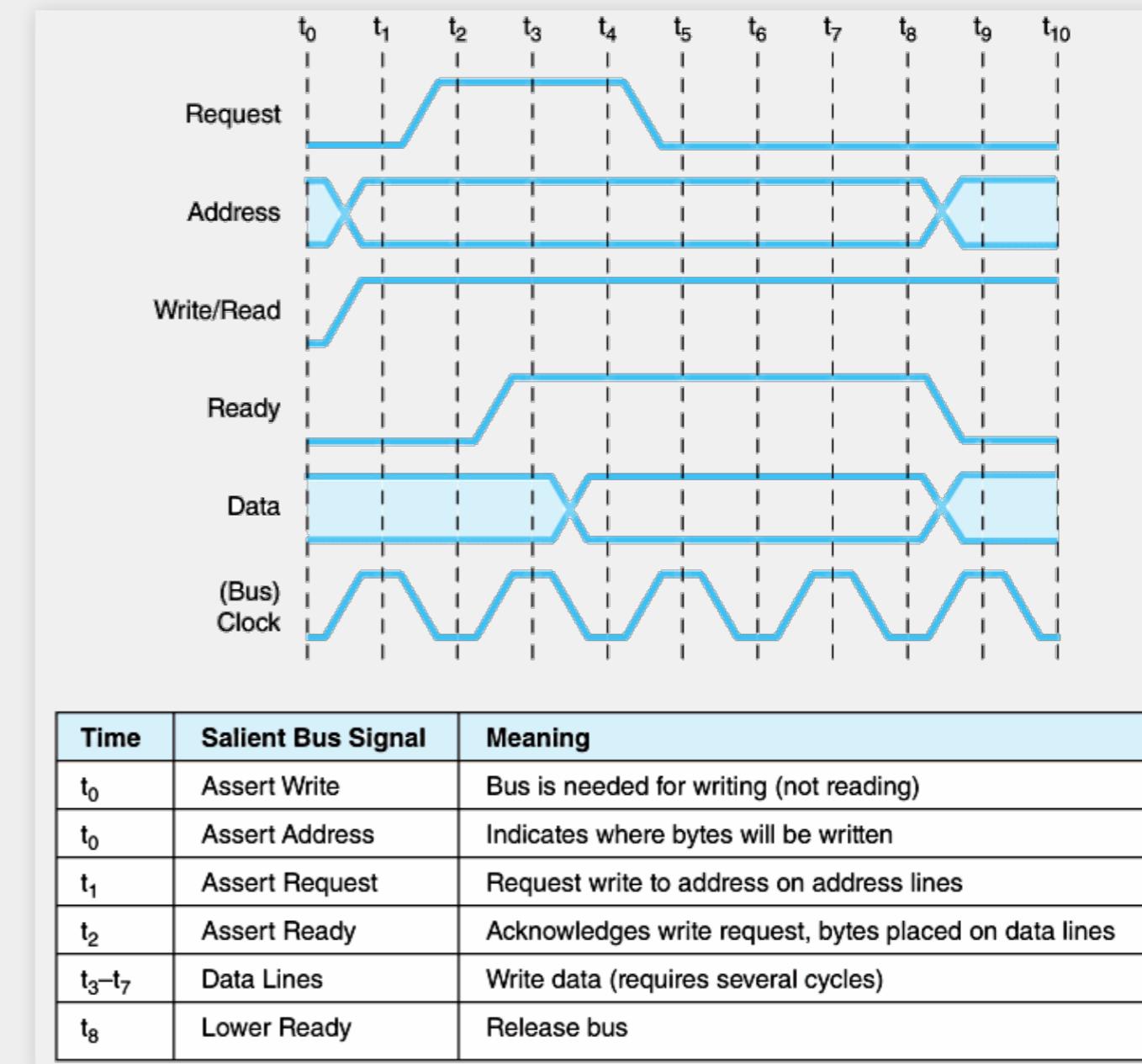
访问磁盘数据的过程:

1. 通知磁盘准备访问，磁盘控制器通过地址总线确定是访问自己，准备读取访问的数据地址
2. 磁盘准备好访问目标地址后，发送就绪信号
3. DMA发送数据
4. 数据传输结束后，复位相关信号

时序图

通常采用时序图(timing diagram)来帮助分析异步总线操作

- 电位通常不是立即上升、下降
- 信号只能在时钟上升、下降时改变
- 数据线和地址线成对出现，阴影表示不稳定状态



数据传输模式

- 并行传输：多位数据同时进行传输
- 串行传输：每个周期只传输一位
- 串行传输比并行传输
 - 需要的元件更少，通常接口尺寸也更小
 - 对于衰减和干扰的抗性更好
 - 传输的距离和速度可以更快

数据传输模式

- 并行传输：多位数据同时进行传输
- 串行传输：每个周期只传输一位
- 串行传输比并行传输
 - 需要的元件更少，通常接口尺寸也更小
 - 对于衰减和干扰的抗性更好
 - 传输的距离和速度可以更快



图片来源: <https://jd.com>

数据传输模式

- 并行传输：多位数据同时进行传输
- 串行传输：每个周期只传输一位
- 串行传输比并行传输
 - 需要的元件更少，通常接口尺寸也更小
 - 对于衰减和干扰的抗性更好
 - 传输的距离和速度可以更快



图片来源: <https://jd.com>

第九讲结束

本期内容总结

- 冯诺依曼模型的第三部分：输入输出系统
 - 子系统性能与计算机系统整体性能的关系：阿姆达尔定律
- I/O子系统组成和体系结构
 - I/O子系统示例
 - 如何控制I/O设备：五种控制方法、字符和块I/O、总线控制与时序图
- 数据传输模式：并行传输和串行传输



Q & A