

计算机组织结构

2 计算机的顶层视图

刘博涵

2022年9月9日



南京大學
NANJING UNIVERSITY

教材对应章节

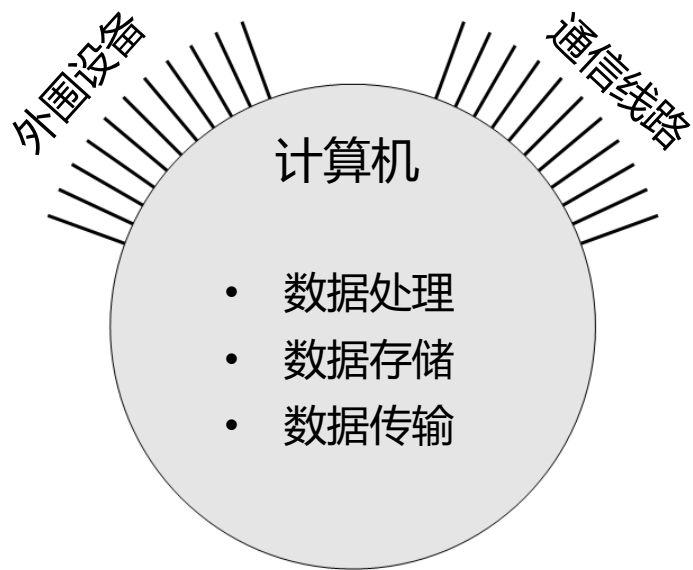


第1章 计算机系统概述

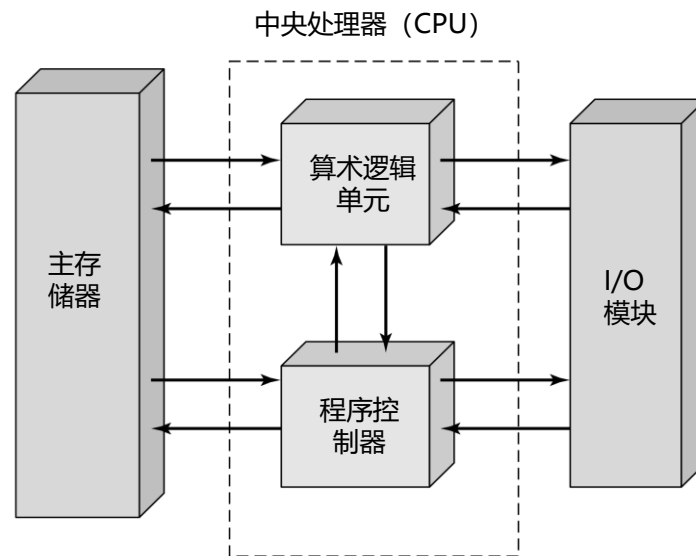


第3章 计算机功能和互连的顶层视图

计算机的不同视图

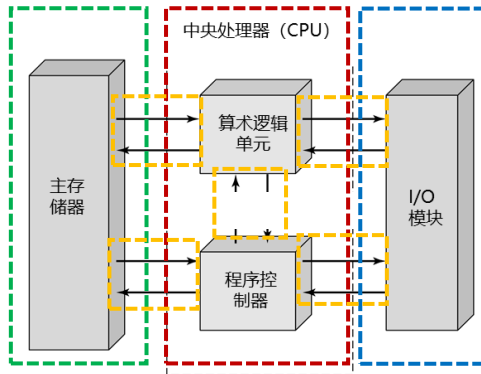


基本功能



冯·诺伊曼结构

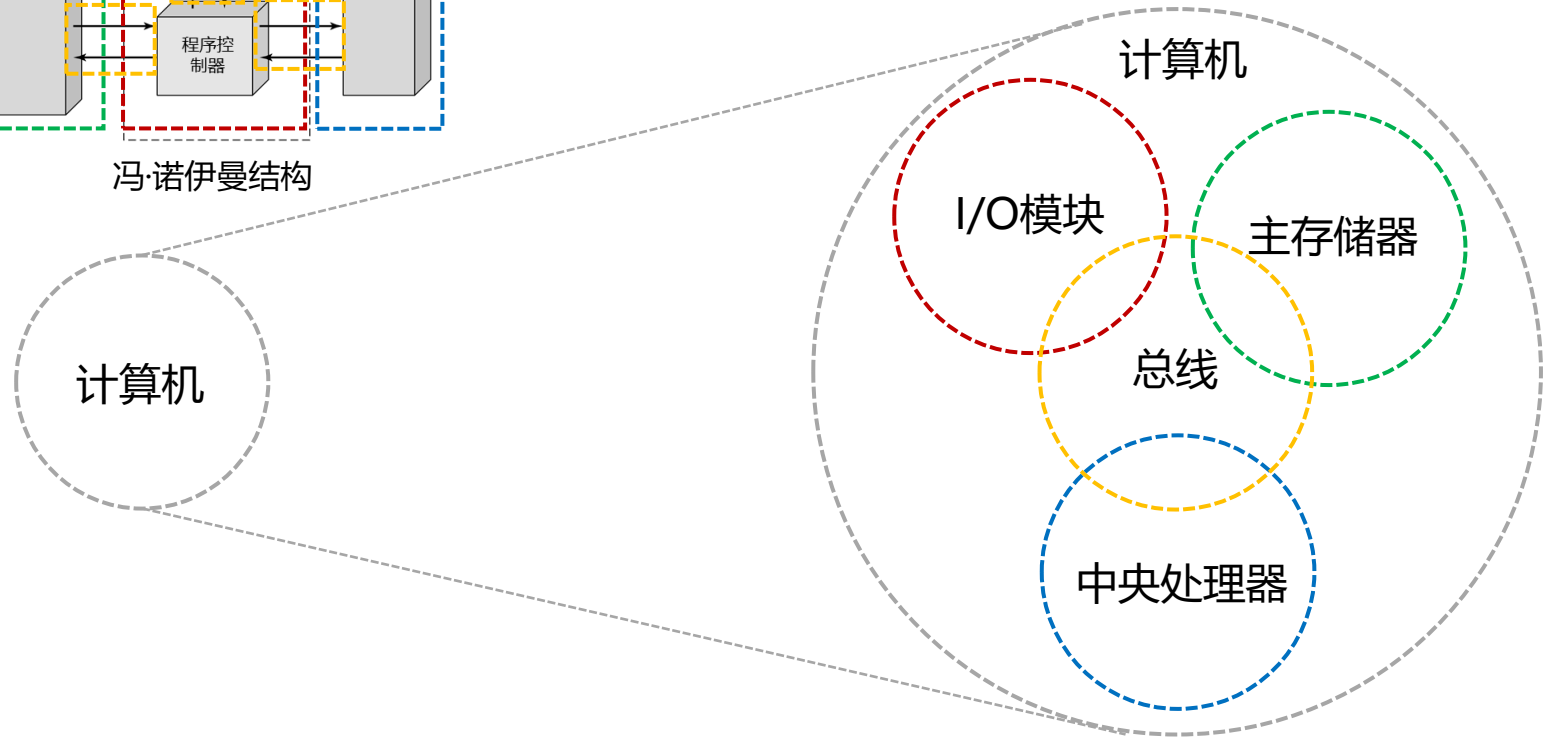
计算机顶层结构



冯·诺伊曼结构

早期，组件之间用**分散方式**相连

现在，组件之间大多用**总线方式**相连



- 每个组件的外部行为，即它与其他组件交换的数据和控制信号
- 互连结构和管理互连结构的使用所需要的控制

冯·诺伊曼结构最重要的思想

冯·诺伊曼的最重要的思想是 “**存储程序 (Stored-program)**”

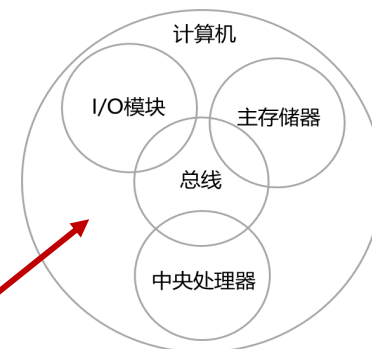
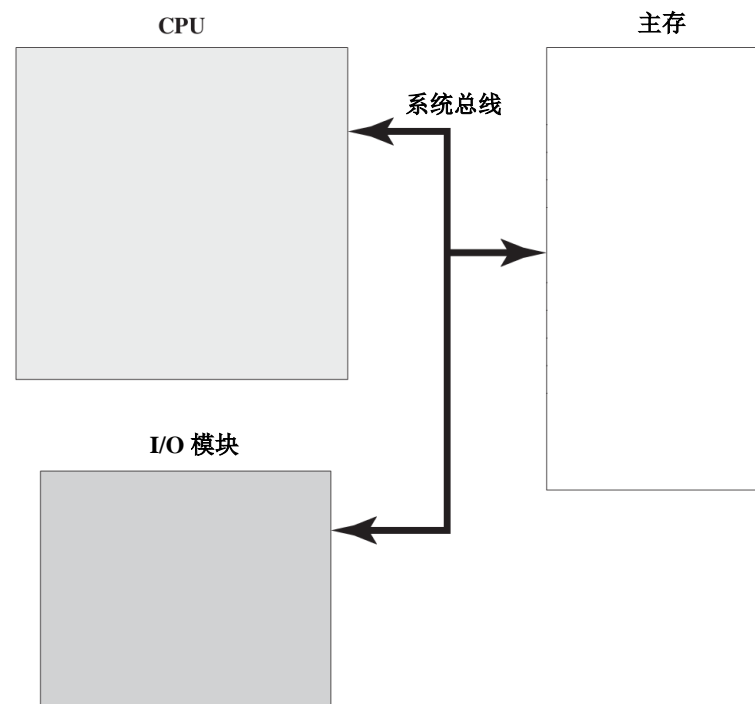
任何要计算机完成的工作都要先被编写成程序，然后将程序和原始数据送入主存并启动执行。一旦程序被启动，计算机应能在不需要操作人员干预下，自动完成逐条取出指令和执行指令的任务。

- 应该有个主存，用来存放程序和数据
- 程序由指令构成
- 应该有一个自动逐条取出指令的部件
- 应该有具体执行指令的部件
- 指令描述如何对数据进行处理
- 应该有将程序和原始数据输入计算机的部件
- 应该有将运算结果输出计算机的部件



冯·诺伊曼体系结构的关键概念

- 指令和数据存储在单个读写存储器中
- 主存中的内容按位置访问，无需考虑其中包含的类型
- CPU从一条指令到下一条指令以顺序方式执行（除非明确修改）
- 与CPU和内存交换从外部来源收集的数据
- 总线是连接两个或多个设备的通信通路



不成比例扩展效应
(Incommensurate Scaling)

回顾: CPU



Intel® Core™ Processors

Intel's highest-performance CPUs for laptops and desktops, delivering advanced responsiveness, connectivity and graphics.

13.3"

Retina display¹



Apple M1 chip

8-core

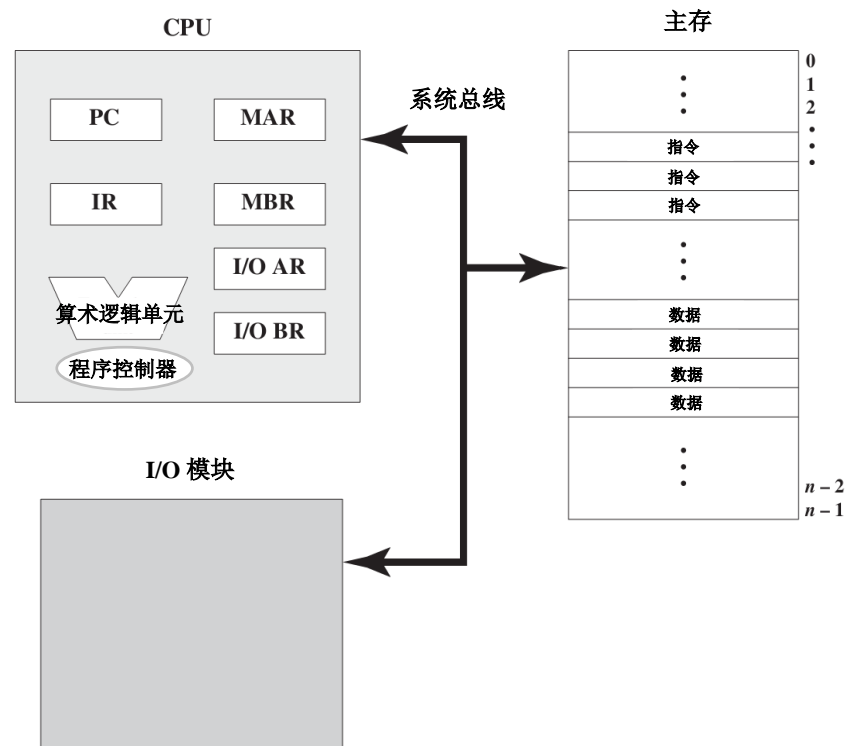
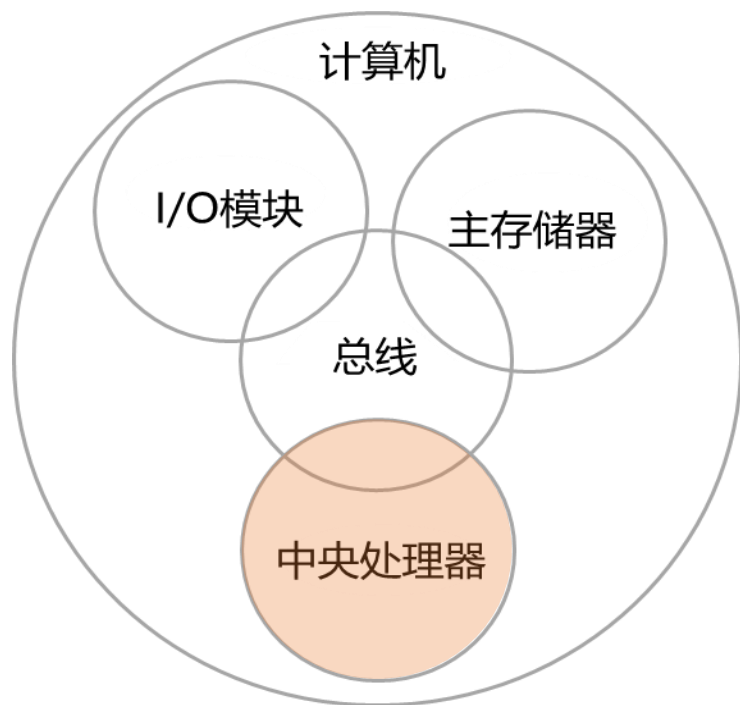
CPU

中央处理单元 (CPU) :获取并执行指令的计算机组成部分, 它由一个ALU、一个控制单元和多个寄存器构成。在单处理单元系统中, 它通常简称为**处理器**。

处理器: 含有一个或多个内核的物理硅片。处理器是计算机组件, 用于解释和执行指令。如果一个处理器包含多个内核, 则称其位**多核处理器**



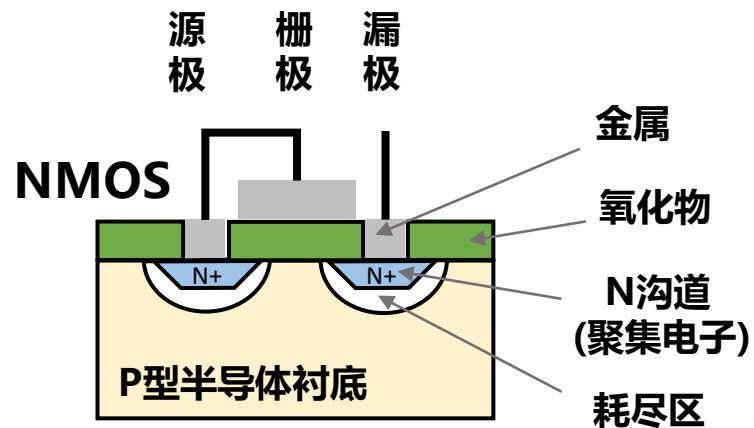
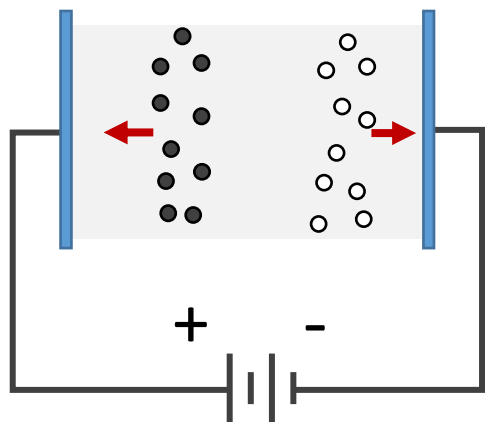
计算机组件: CPU



- CPU从一条指令到下一条指令以顺序方式执行（除非明确修改）

PC	=	程序计数器
IR	=	指令寄存器
MAR	=	存储器地址寄存器
MBR	=	存储器缓冲寄存器
I/O AR	=	I/O地址寄存器
I/O BR	=	I/O缓冲寄存器

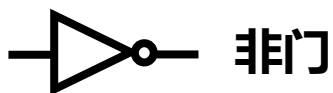
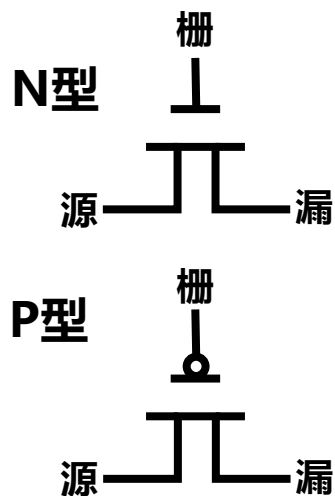
计算机组件: CPU



电场效应

场效应管

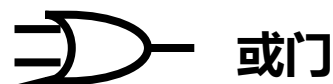
当电压**高**于阈值电压 **可以导通**
当电压**低**于阈值电压 **不能导通**



非门



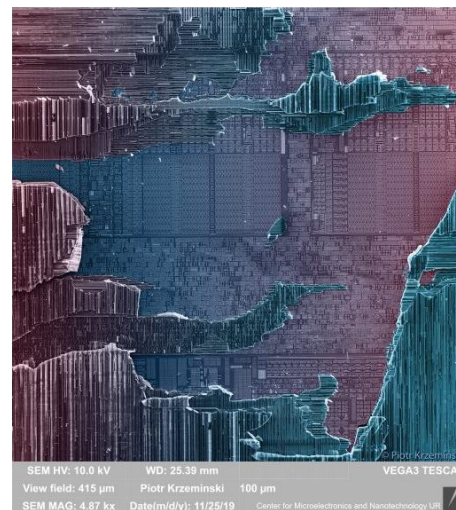
与门



或门

.....

芯片



显微镜下的CPU

<https://www.youtube.com/watch?v=7d1eyZBpLn8>



计算机组件: CPU

年份	CPU型号	制作工艺	晶体管数量
2000	奔腾4 Willamette	180nm	0.42亿
2010	酷睿i7-980X	32nm	11.7亿
2013	酷睿i7 4960X	22nm	18.6亿
2019	苹果A13	7nm	85亿
2020	苹果A14	5nm	118亿
2022	苹果M2	5nm	200亿

一直提高: **制作工艺**



几乎不变: **CPU的大小**

一直增加: **晶体管的数量**

能一直提高下去吗? **物理极限**

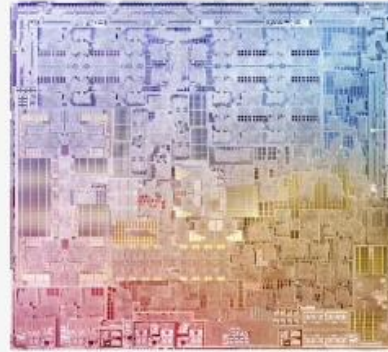
能通过无限增大CPU的**面积**来解决吗?



计算机组件: CPU



苹果 M1



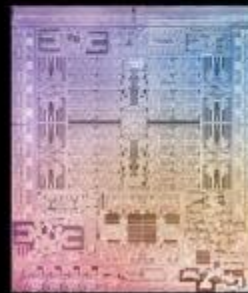
苹果 M2



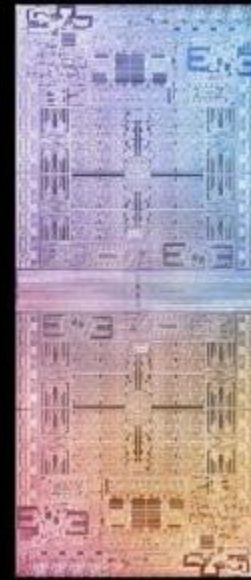
苹果 M1



苹果 M1 Pro



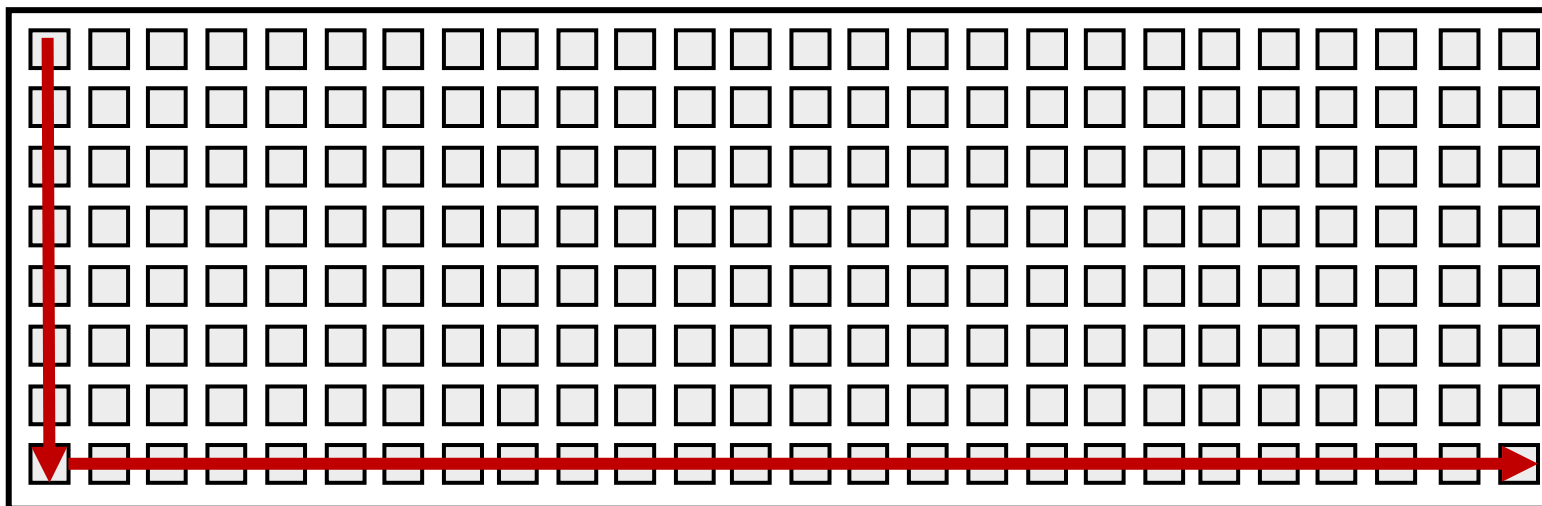
苹果 M1 Max



苹果 M1 Ultra

计算机组件: CPU

一个现代的CPU由几十上百亿个晶体管构成。每个晶体管都可以看成一个开关。



面积增大意味着，**互连延迟**增大。一个时钟周期需要大于最长互连延迟。

$$\text{5GHz的CPU的物理极限: } \frac{3 \times 10^8 \text{ m/s}}{5 \times 10^9 \text{ s}^{-1}} = 0.06 \text{ m}$$

晶体管数量:	M2 = 1.25 M1
单核性能:	M2 = 1.09 M1
多核性能:	M2 = 1.18 M1



问题1：CPU的频率不能无限提高

- 理论限制
 - MOS管开关、脉冲通过门电路需要时间
 - 为了信号同步，每个脉冲信号需要持续一定的时间
 -
- 制造限制
 - 芯片面积越来越大，导致连线延迟越来越大，需要保证信号在设计指定时钟周期内从芯片的一角到达另一角
 - 频率越高（即MOS管的开关频率也越高）会导致开关损耗也越高，CPU耗电和散热会提高

2002年：奔腾4处理器的主频为**3.06GHz**

2022年：12代酷睿i5处理器的主频为**3.7GHz**



解决1：改进CPU芯片结构

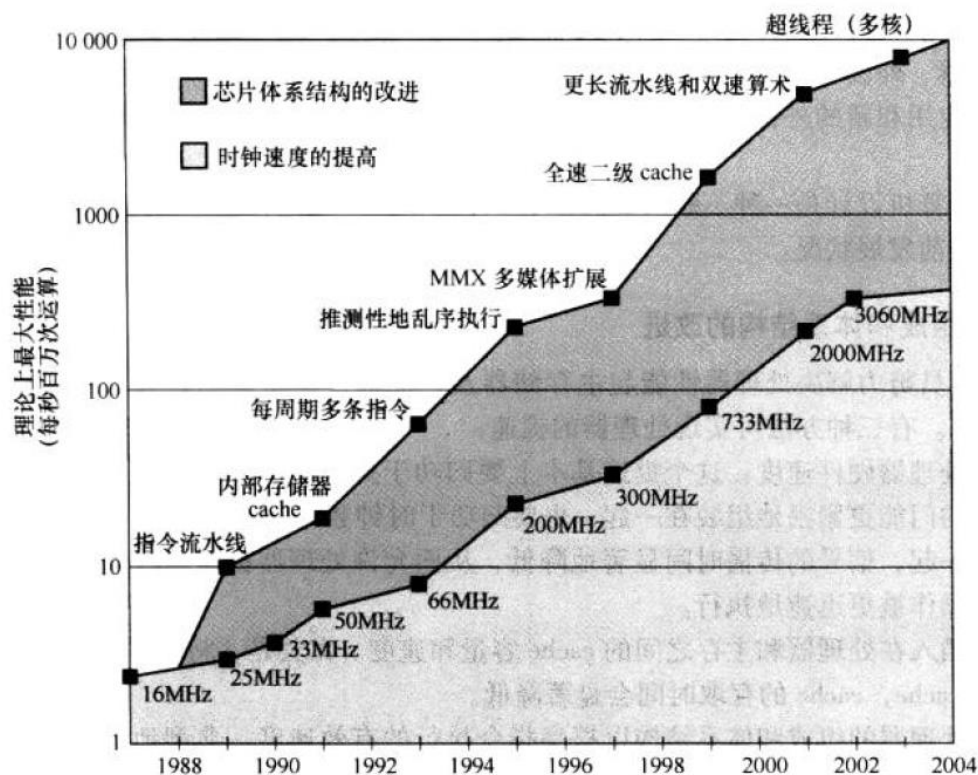


图 2-12 Intel 微处理器性能 [GIBB04]

领域定制
如面向人工智能的芯片

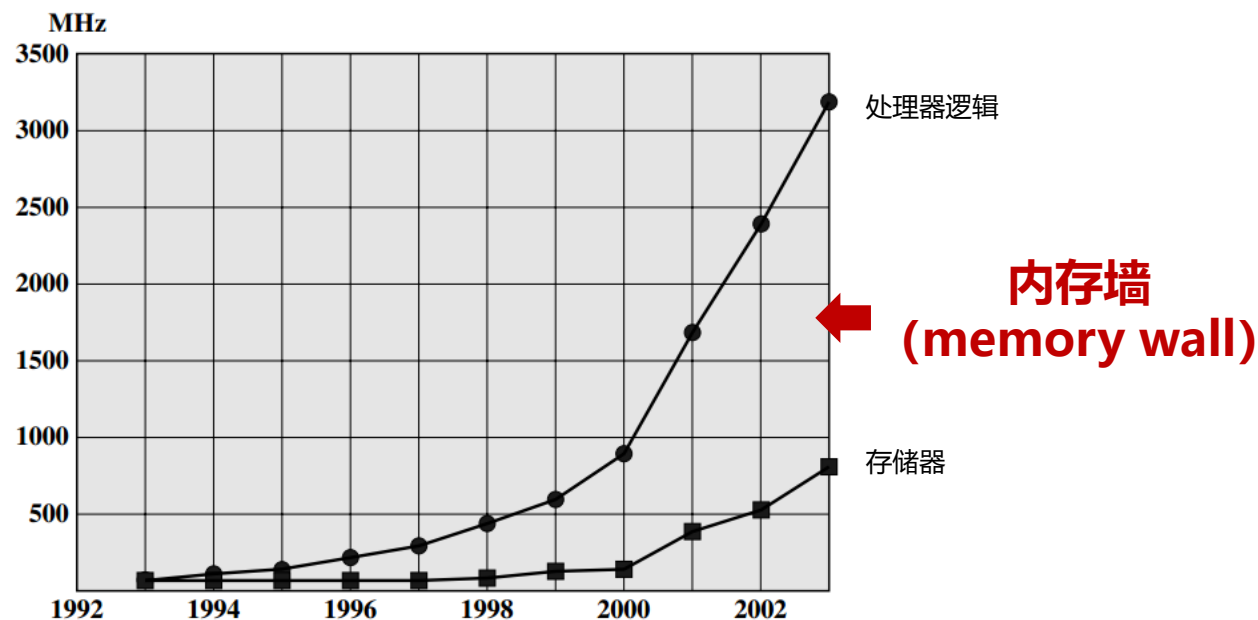
晶体管数量的增加为更先进、更复杂的体系结构提供了基础

- ✓ 第15讲：指令周期和指令流水线
- ✓ 第16讲：控制器



问题2：内存墙的存在

- 问题
 - 主存和CPU之间传输数据的速度跟不上CPU的速度



从1980年到2010年：

CPU的时钟频率提高了**2500**倍

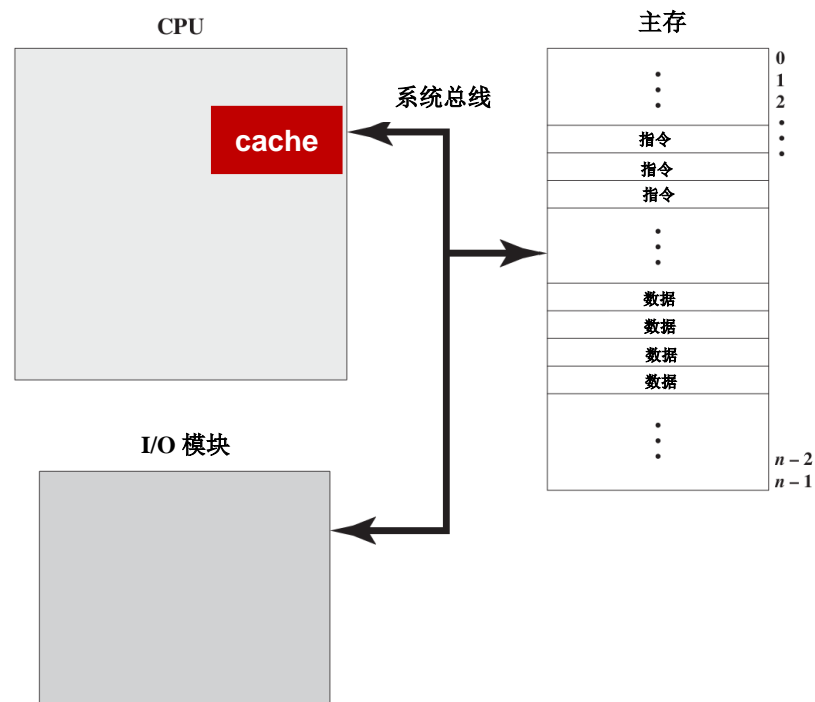
内存的访问速度提高了**9**倍，但存储空间提升了**125000**倍。



解决2：采用高速缓存（Cache）

- 解决方法

- 添加一级或多级缓存以减少存储器访问频率并提高数据传输速率
- 增大总线的数据宽度，来增加每次所能取出的位数
- ...

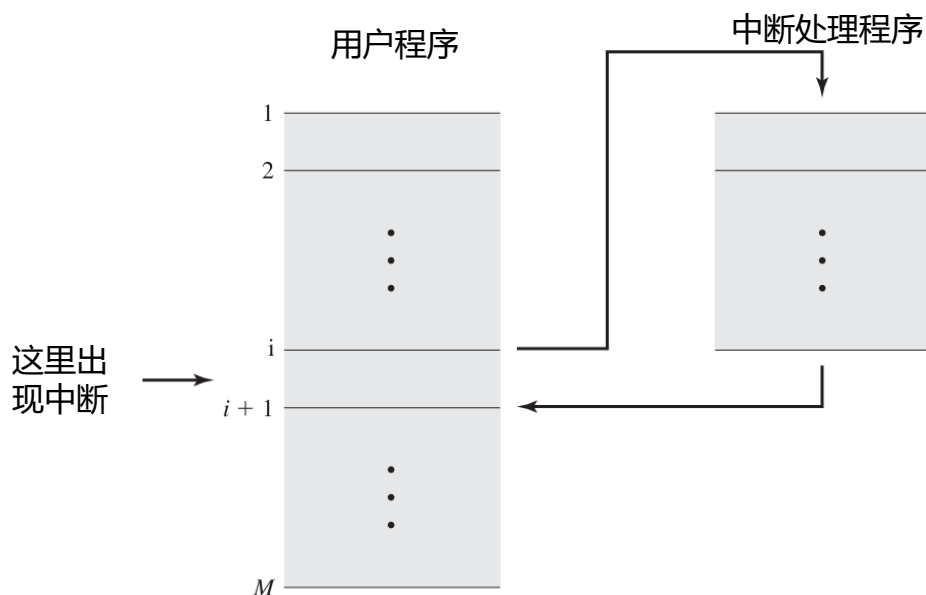


✓ 第08讲：高速缓冲存储器（Cache）



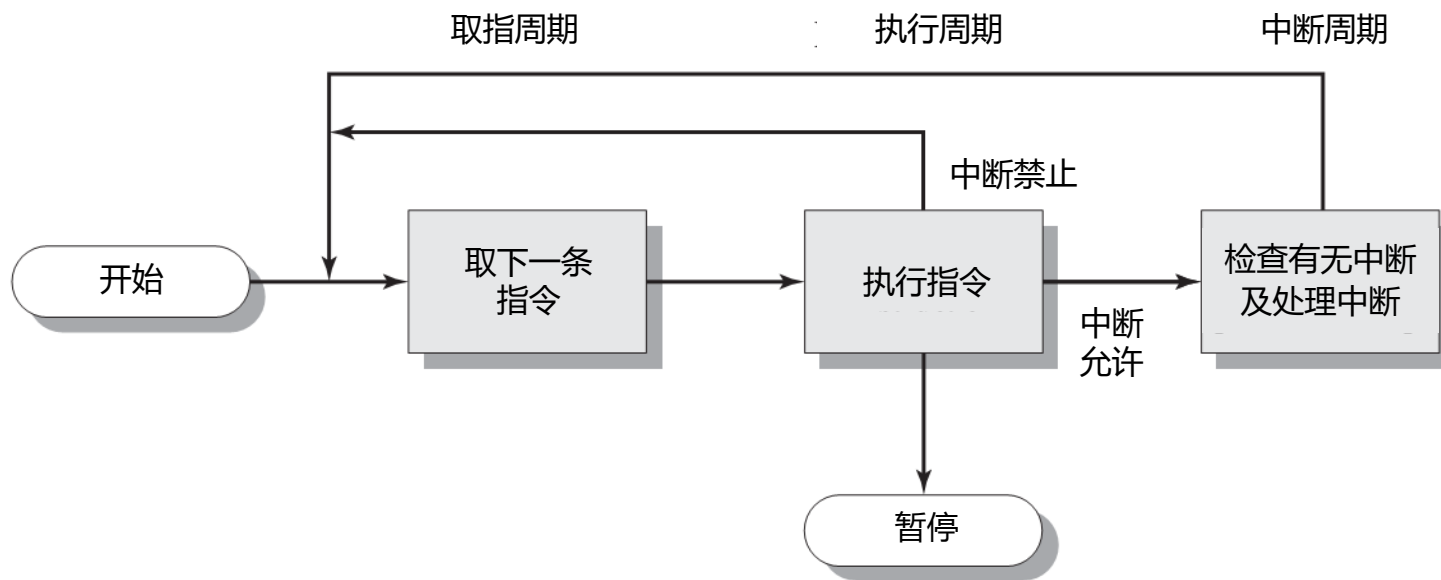
问题3：CPU等待I/O传输数据

- 问题
 - CPU 在等待 I/O 设备时保持空闲
- 解决方法
 - **中断**：其他模块（例如 I/O）可以中断正常处理顺序的机制



解决3：采用中断机制

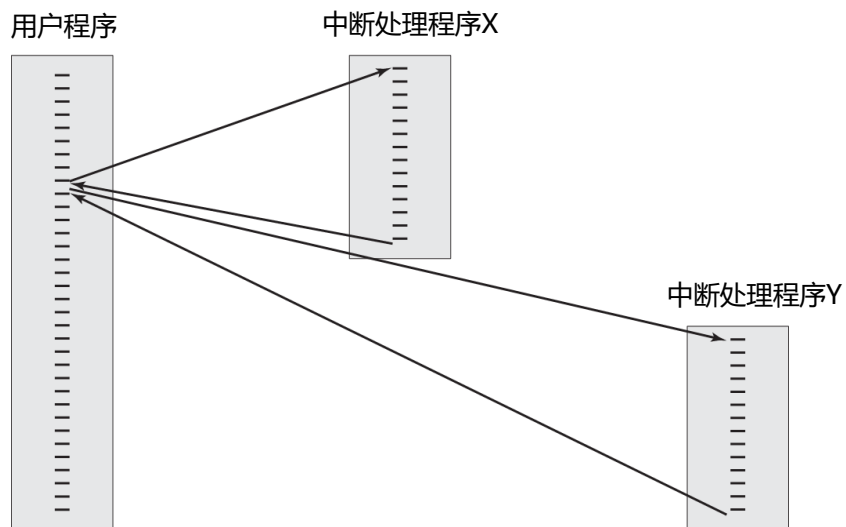
- 中断检测
 - 将中断周期加入指令周期



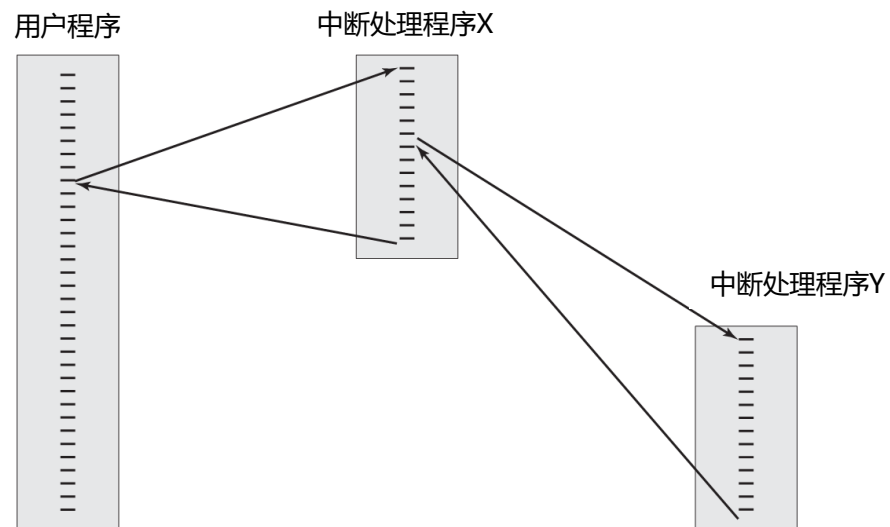
✓ 第14讲：指令系统



多重中断



顺序中断处理



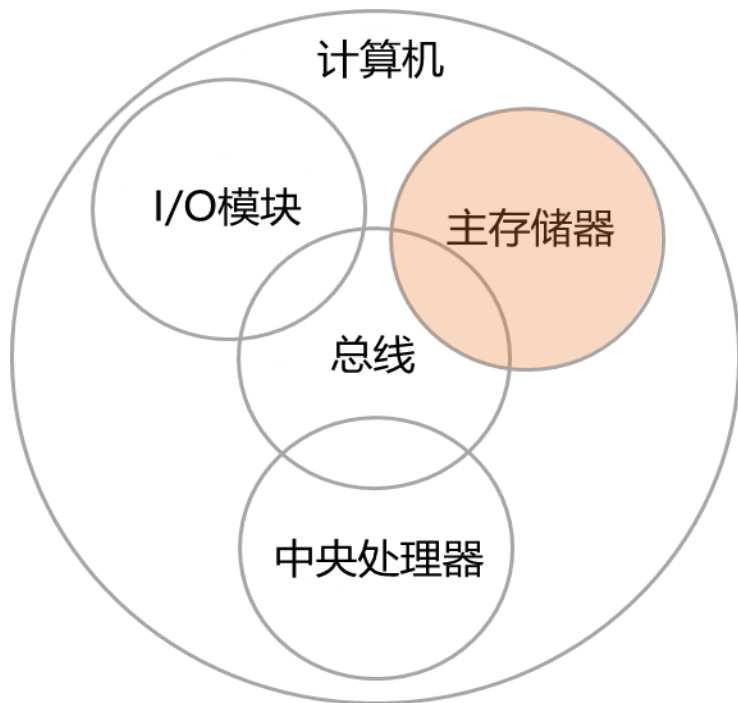
嵌套中断处理



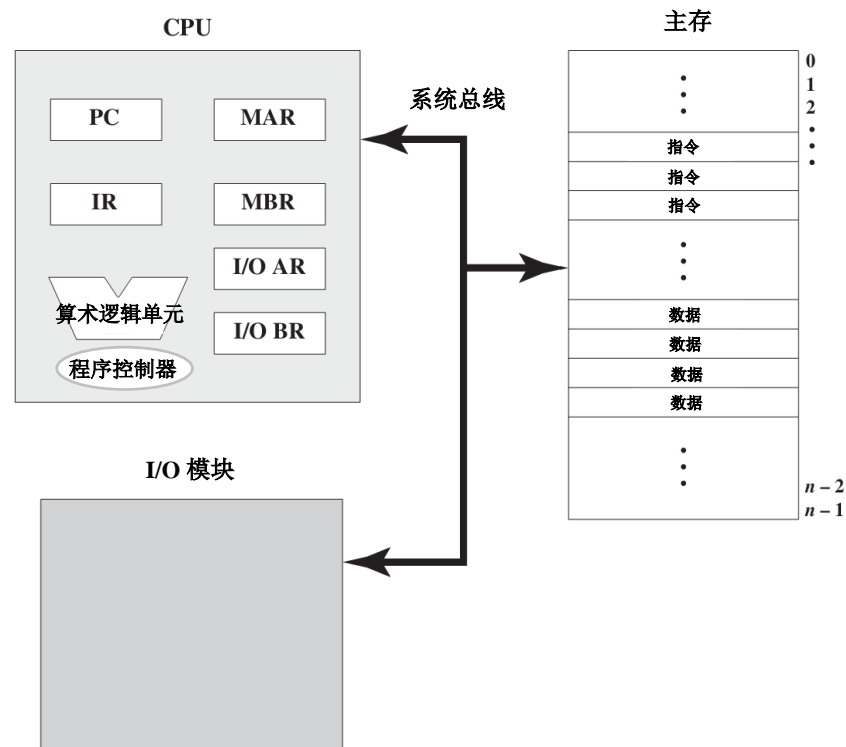
回顾: 存储器



计算机组件: 存储器



- 指令和数据存储在单个读写存储器中
- 主存中的内容按位置访问，无需考虑其中包含的类型



PC	=	程序计数器
IR	=	指令寄存器
MAR	=	存储器地址寄存器
MBR	=	存储器缓冲寄存器
I/O AR	=	I/O地址寄存器
I/O BR	=	I/O缓冲寄存器

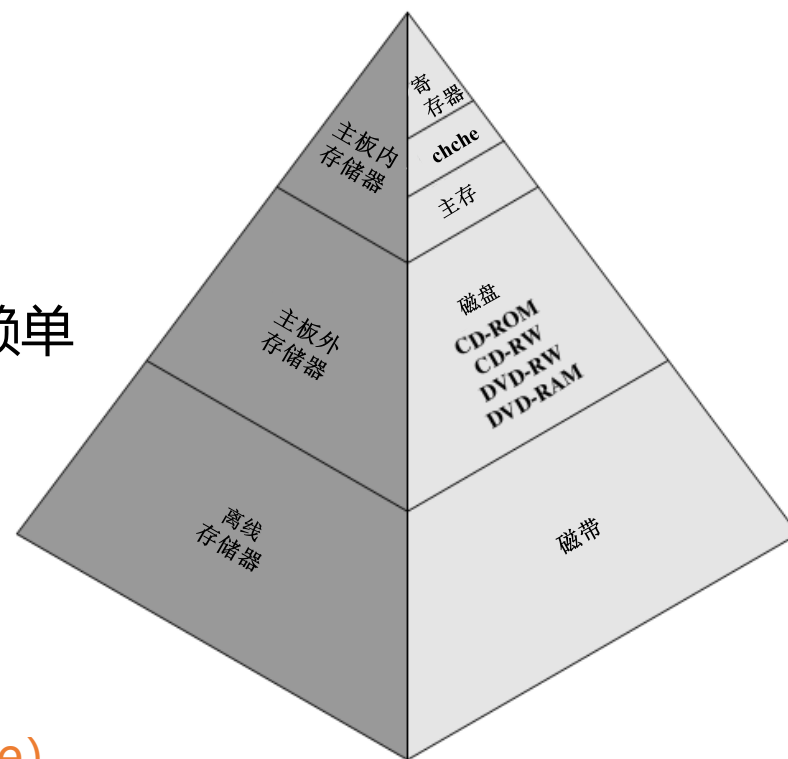
问题4：兼顾存储容量、速度和成本

- 约束
 - 容量：越大越好
 - 速度：跟上处理器
 - 成本：相对于其他组件合理
- 约束之间的关系
 - 更短的访问时间，更高的每比特成本



解决4：层次式存储结构

- 需求
 - 大容量数据存储
 - 高速性能
- 解决方案
 - 使用存储器层次结构而不是依赖单个存储器组件



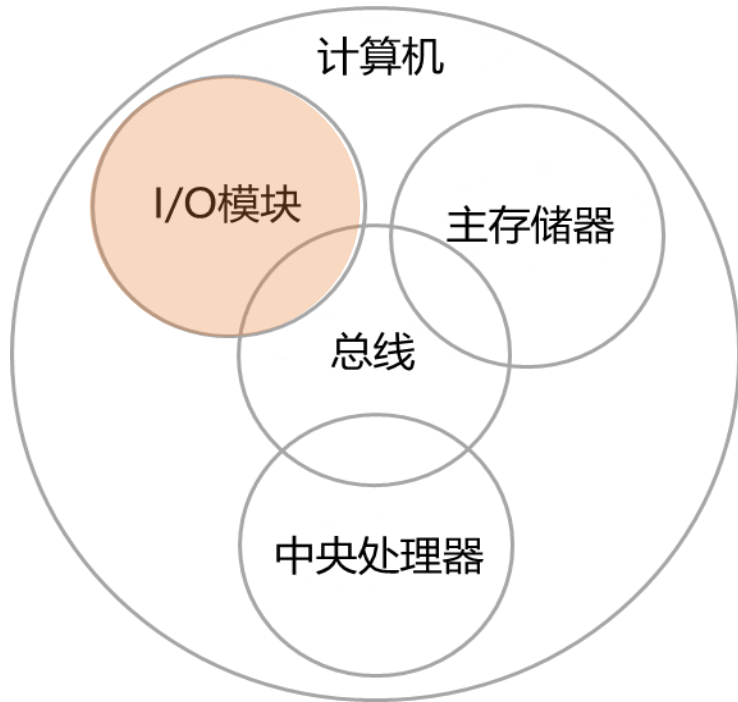
- ✓ 第07讲：内部存储器
- ✓ 第08讲：高速缓冲存储器 (Cache)
- ✓ 第09讲：外部存储器
- ✓ 第10讲：数据校验码
- ✓ 第11讲：磁盘冗余阵列 (RAID)
- ✓ 第12讲：虚拟存储器



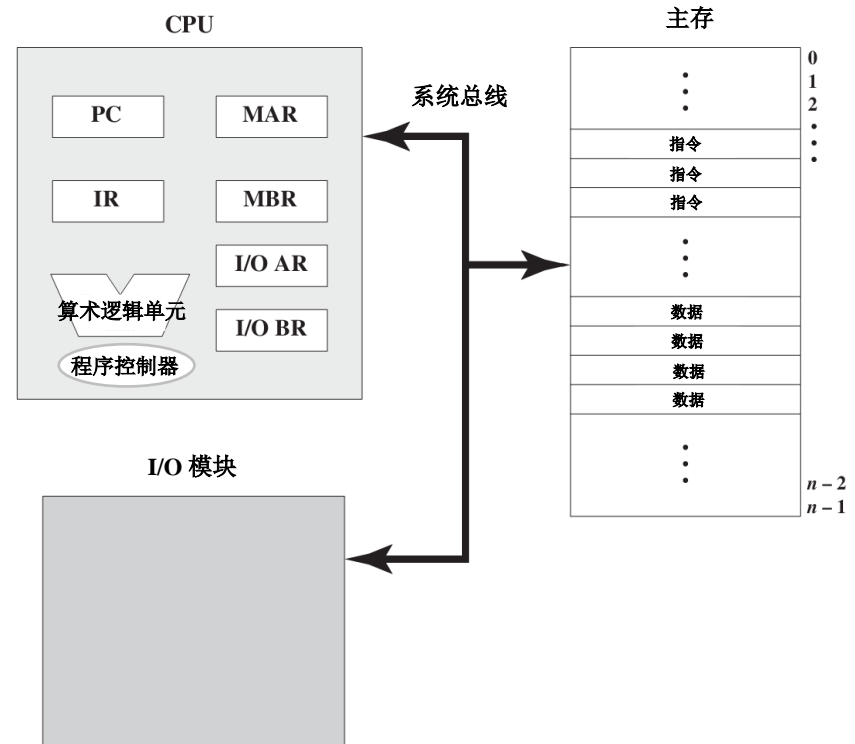
回顾: I/O



计算机组件: I/O模块



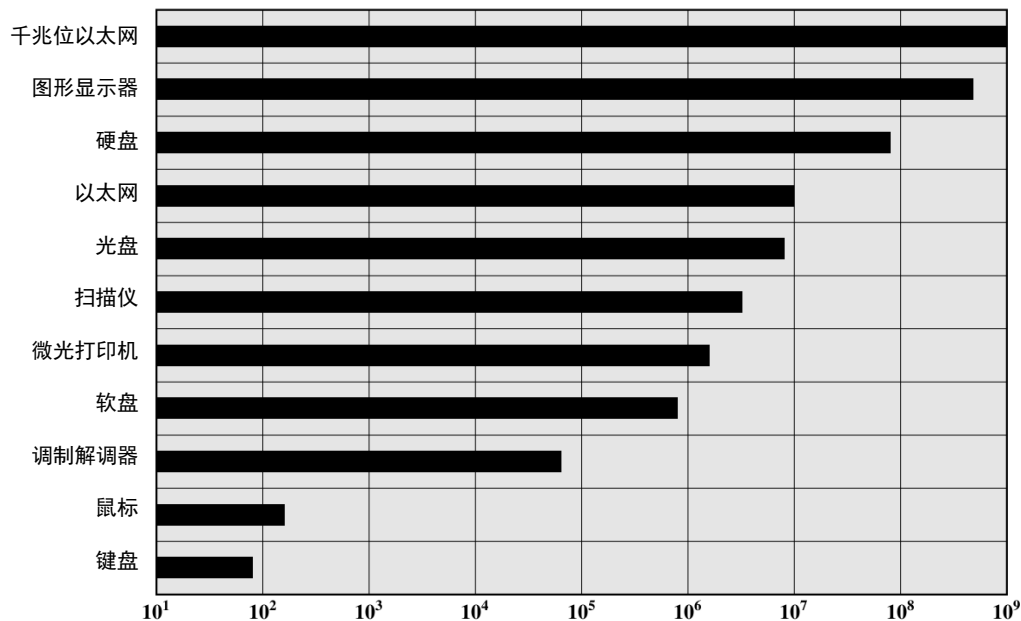
- 与 CPU 和内存交换从外部来源收集的数据



PC	=	程序计数器
IR	=	指令寄存器
MAR	=	存储器地址寄存器
MBR	=	存储器缓冲寄存器
I/O AR	=	I/O地址寄存器
I/O BR	=	I/O缓冲寄存器

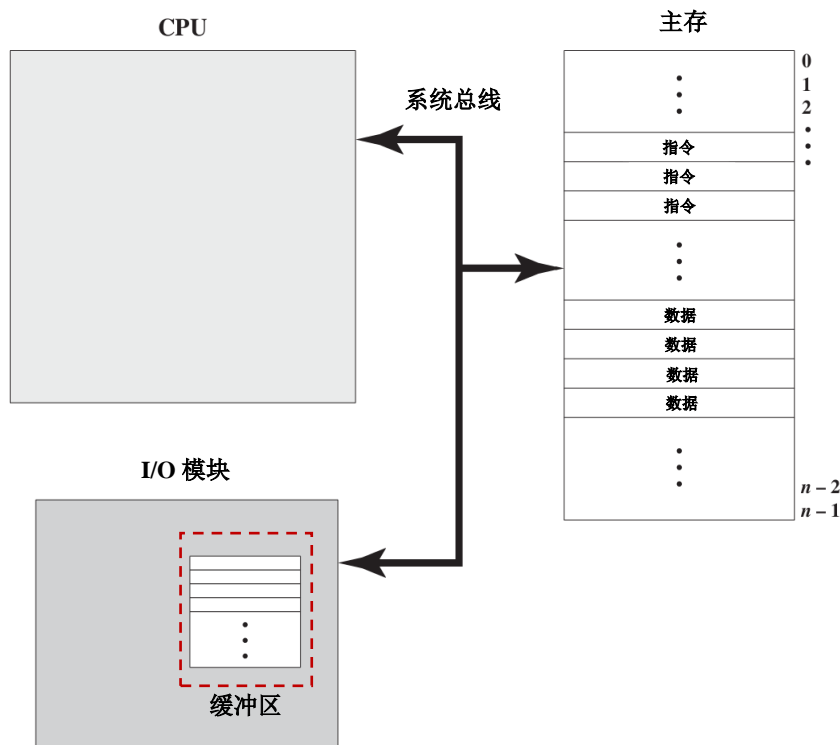
问题5：I/O设备传输速率差异大

- 问题
 - I/O性能跟不上CPU速度的提升



解决5：采用缓冲区和改进I/O操作技术

- 解决方法
 - 设立缓冲区
 - 新的接口技术
 - 不同的I/O操作技术
 - ...



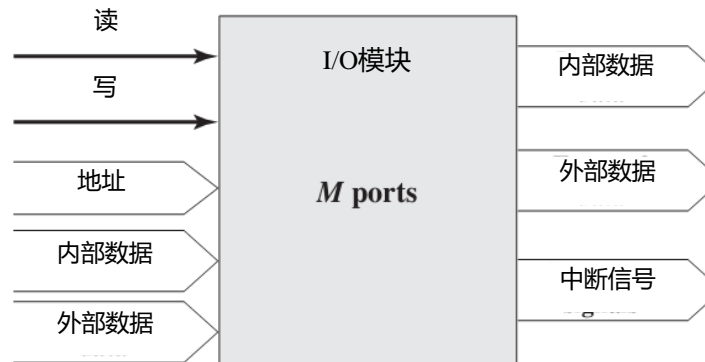
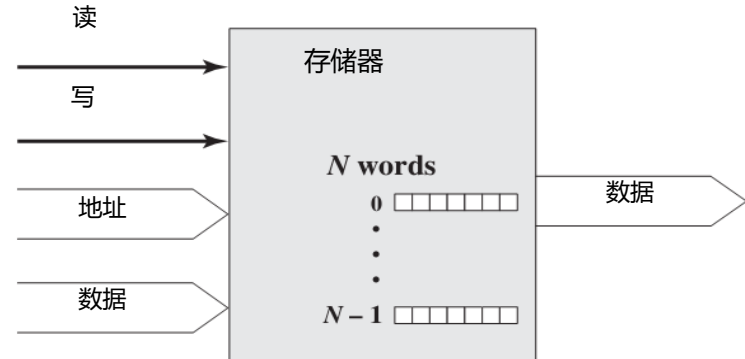
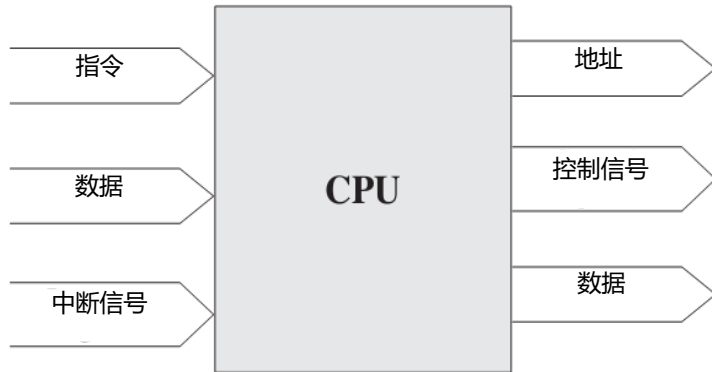
回顾: 总线

总线的两大基本特征是什么？

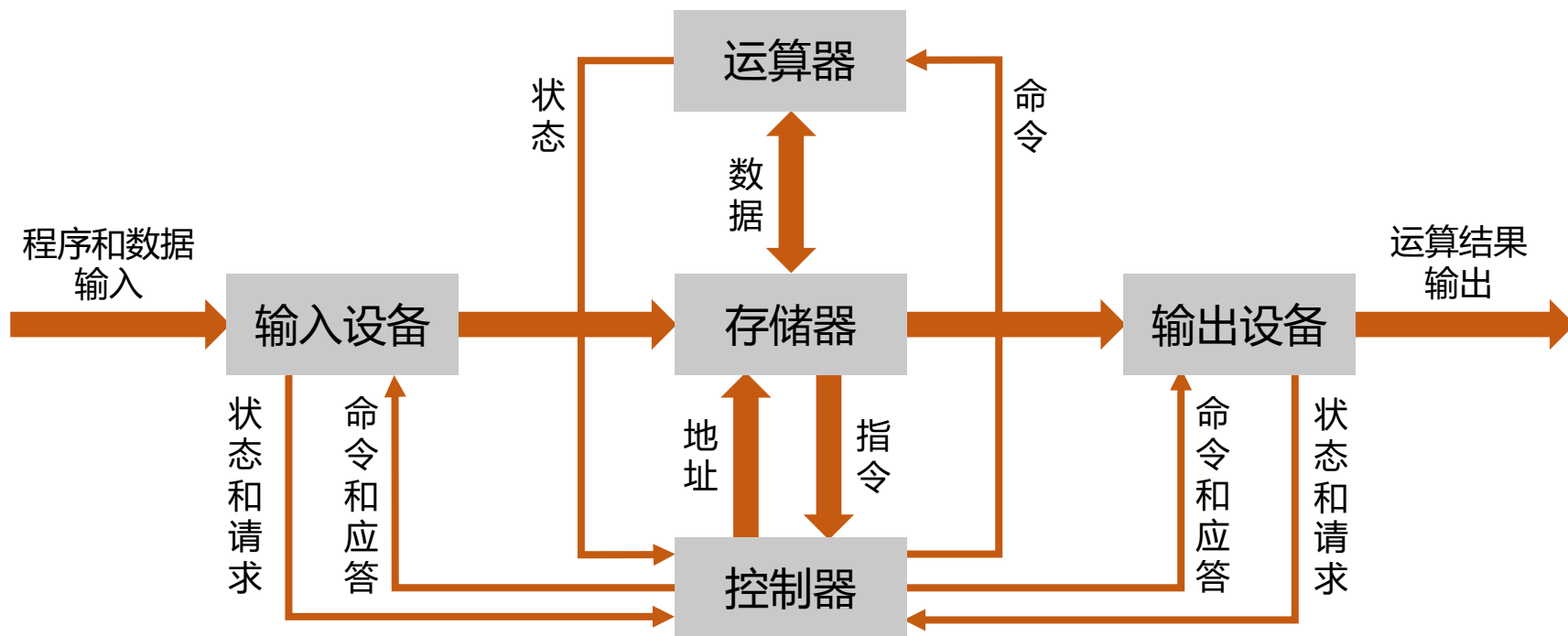
- **共享：**多个部件连接在同一组总线上，各个部件之间都通过该总线进行数据交换。
- **分时：**同一时刻，总线上只能传输一个部件发送的信息。



计算机组件: 总线



问题6：计算机部件互连复杂

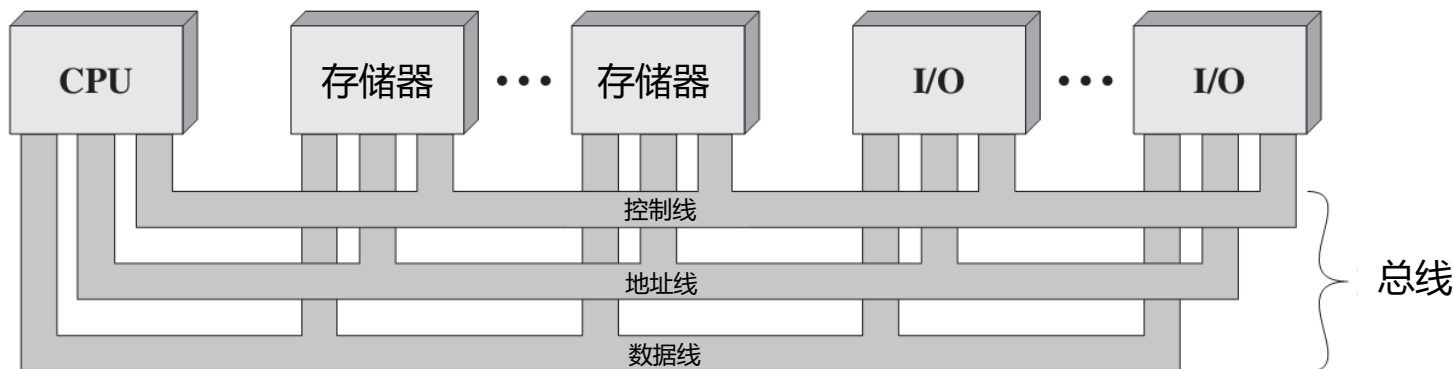


早期，组件之间用**分散方式**相连

解决6：采用总线

- 数据传输类型

- 控制线：控制数据线和地址线的访问和使用
- 地址线：指定数据总线地址I/O端口上数据的来源或去向
- 数据线：在系统模块之间传送数据



共享传输介质
简化互连布局 and 处理器控制

✓ 第13讲：总线



总结

- 计算机的顶层视图
 - 基本功能，冯·诺伊曼结构
- 计算机体系结构遇到的问题及解决方案
 - CPU的频率不能无限提高 → 改进CPU芯片结构
 - 内存墙的存在 → 采用高速缓存（Cache）
 - CPU等待I/O传输数据 → 采用中断机制
 - 兼顾存储容量、速度和成本 → 层次式存储结构
 - I/O设备传输速率差异大 → 采用缓冲区和改进I/O操作技术
 - 计算机部件互连复杂 → 采用总线



谢谢

bohanliu@nju.edu.cn



南京大學
NANJING UNIVERSITY