

# 计算机组成原理

## 总线系统

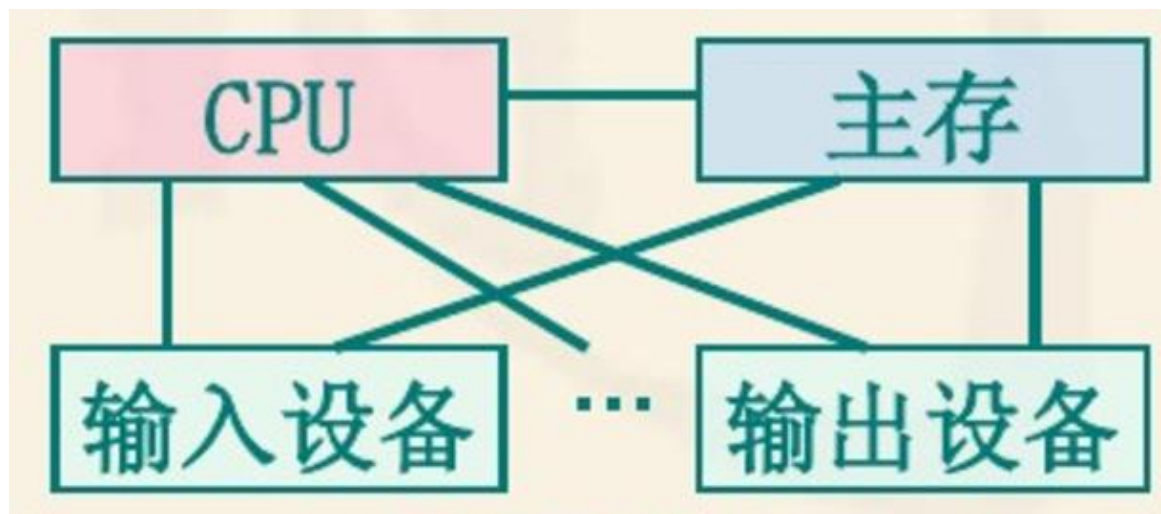
王浩宇,教授

haoyuwang@hust.edu.cn

<https://howiepku.github.io/>

# 为什么需要总线？

## ■ 计算机的若干功能部件之间如何连接？



分散连接方式

通信性能好，可扩展性能差

# 为什么需要总线？

## ■ 计算机的若干功能部件之间如何连接？

将各部件连到一组公共信息传输线上，通过这条公共线路进行数据传输



总线连接方式

**可扩展性好，易产生通信瓶颈**

# 总线设计需要考虑的问题

- 总线的设计影响着计算机系统的性能
- 如何解决不同速度的设备同时连接到总线上所造成的通信瓶颈？
- 不同类型的外围设备如何与计算机连接起来一起工作？
- 如何解决多个设备同时竞争总线控制权？
- 如何同步总线上不同设备的操作？

**总线的结构形态**

**总线接口**

**总线的仲裁**

**总线的定时和数据  
传送模式**

# 1. 总线概述和基本形态

- **总线的传统定义：** 构成计算机的互联机构，是多个系统功能部件之间进行数据传送的公共通路
  - 借助于总线连接，计算机在各系统功能部件之间实现**地址、数据和控制信息的交换**，并在争用资源的基础上进行工作
- **现代总线是指连接多个计算机内部功能部件或多个计算机的通信系统，总线既包括相关的硬件（总线控制器，总线接口），软件，也包括相关的通信协议**

# 总线的作用

- 是各功能部件间传递各类信息的通道
- 是系统中各部件间的物理接口，能够减少各部件通信的复杂程度
- 提供信息交换时所需的数据、地址、时序和控制信息
- 提供一个共同遵循的协议或标准
- 不应成为整个计算机性能的瓶颈
- 方便计算机系统的集成、扩展和进化

# 总线分类

## ■ 按照总线在计算机系统中所处的位置

- **片内总线**：芯片内部各组成部分之间的连接线
  - 内部寄存器连接，寄存器与运算器，控制器与执行部件的连接
- **系统总线/CPU总线**：CPU直接连接主存、IO模块等主要功能部件的信息传输线，与CPU引脚直接相连
  - 系统总线的传输能力对计算机整体性能影响非常大
- **IO总线**：连接计算机内部的中低速IO设备，通过桥接器与高速总线相连，将低速设备与高速总线相分离，提升总线系统性能
- **外部总线**：连接计算机与外部设备，使其在计算机系统或计算机与其他系统之间进行数据通信，也称通信总线
  - 不同部件的特征差异较大（连接方式，传输距离，工作方式等），外部总线种类繁多

# 总线分类

## ■ 按总线上传输的信息（信号线功能）分类

### ■ 地址线AB (Address Bus) : 地址总线

- 单向总线，用来标明发送或接收数据的设备编号和需要访问的设备内部单元的地址
- 其宽度表示地址空间范围

### ■ 数据线DB (Data Bus) : 数据总线

- 双向总线，是总线设备进行数据交换的通路
- 数据线宽度？ 每次传送的二进制位数

### ■ 控制线CB (Control Bus) : 控制总线

- 有出/有入，用于控制总线设备对总线的使用



# 总线的分类

## ■ 按总线信号是否有多个功能分类

- 专用总线 (Dedicated)
- 复用总线 (Multiplexed)

## ■ 复用型总线

- 定义了多种意义的信号或者连接到多个总线设备
- 使用较少的信号线，节省空间和成本
- 地址、数据线复用 A/D；读写复用 W/R

# 总线的分类

## ■ 按总线的定时方式分类

- 同步总线 (Synchronous)
  - 传输双方采用公共时钟进行同步
- 异步总线 (Asynchronous)
  - 采用应答机制进行同步

## ■ 按其他的标准进行分类

- 例如，按照总线的一次传输的数据位数分为1位、8位、16位、以及32位总线等

# 总线的特性

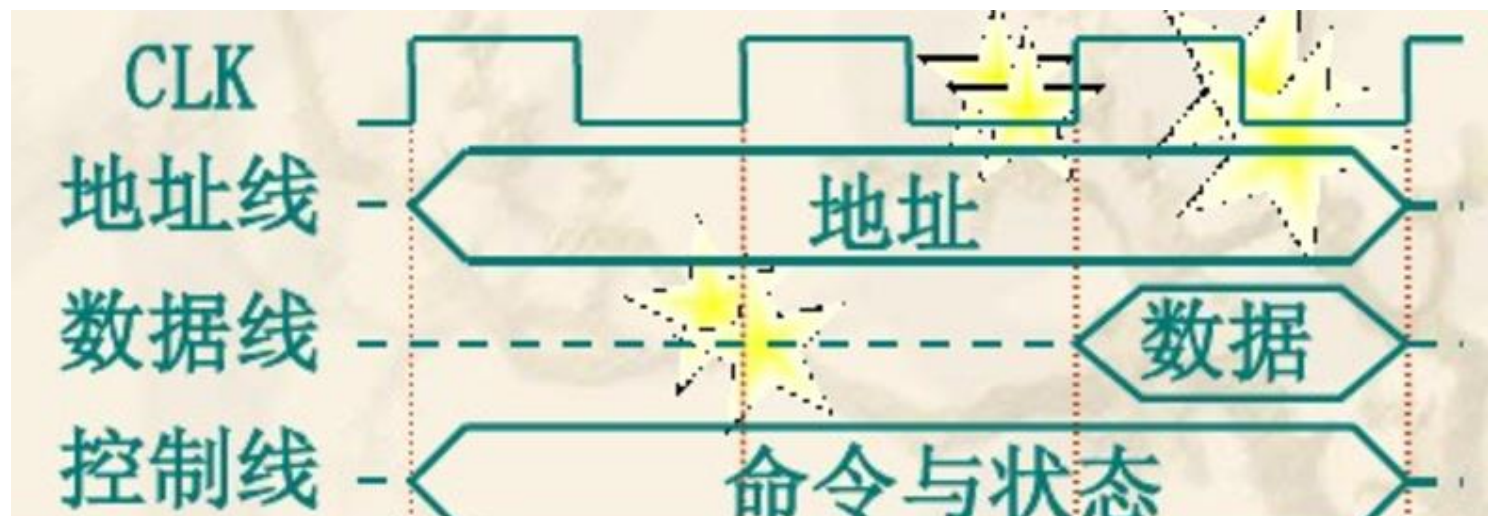
- 物理特性：总线上模块或设备**约定的物理连接方式**（根数、插头、插座形状，引脚排列方式、线宽）**总线标准化所需要**
- 功能特性：按照传输功能及方式需求，约定的信号线数量、功能、传输方向等



- 电气特性：信号线上表示逻辑信息的电压值及功率等

# 总线的特性

- 时间特性：又称逻辑特性，指传输过程中各信号线上信号有效的时长及前后顺序关系（时序特性）
  - 每根线在什么时间有效
  - 时间特性一般可用信号时序图来描述



# 总线的标准化

## ■ 总线的标准化

- 总线的目标就是连接部件，独立与其他部件，灵活组件系统，提高效率、降低成本
- 为了使不同厂家生产的相同功能部件可以互换使用，就需要进行系统总线的标准化工作。总线标准，如PCI、ISA等

## ■ 采用标准总线的优点

- 简化系统设计
- 简化系统结构，提高系统可靠性
- 便于系统的扩充和更新

# 总线的参数指标

- **总线频率：**反映总线工作的速率（ $f$ ），通常单位是MHz
  - 总线时钟周期的倒数，同步总线中双方拥有完全同步的时钟信号
  - 早期总线的时钟频率与CPU相同，但后来CPU发展较快，总线时钟独立于CPU时钟
- **总线宽度：**数据总线的位数（ $w$ ），单位是b（位）
  - 是计算机的一个重要指标，通常与处理器的字长相一致；
  - 在并行传输总线中，数据总线宽度直接决定了可以并发传输的位数
- **总线传输周期：**一次总线操作完成所需要的时间，包括总线申请阶段，寻址阶段，传输阶段和结束阶段
- **单时钟传输次数：**一个总线周期内传输数据的次数，通常为1
  - DDR：时钟上 下跳沿分别传输一次数据，该值为2
  - QDR：该值为4，总线内部时钟为两个相位相差90度的时钟
  - AGP 8X（Accelerated Graphics Port 8X）：单时钟传递八次

# 总线的参数指标

- **总线带宽：**总线的最大数据传输速率，即单位时间内总线上传输数据的位数，用每秒传输信息的字节数来衡量，单位MB/s或GB/s
  - 通常不考虑总线传输周期中总线申请和寻址等阶段的开销
  - **同步总线带宽 = 总线宽度\*总线时钟频率\*单时钟传输次数**
    - ISA总线（16位，带宽8MB/S或者16MB/S）
    - EISA总线（32位，带宽33MB/S）
    - VESA总线（32位，带宽133MB/S）
- **总线负载能力：**总线上能同时连接的设备数
  - 例如PCI总线插槽通常只能外接3个扩展设备
- **同步方式：**同步、异步
- **多路复用：**能否共用一条物理总线（分时传送）
- **信号线数量：**数据线、控制线和地址线的总和；
- **总线控制方式：**包括突发工作、自动配置、仲裁方式、逻辑方式、计数方式等

# 总线的参数指标

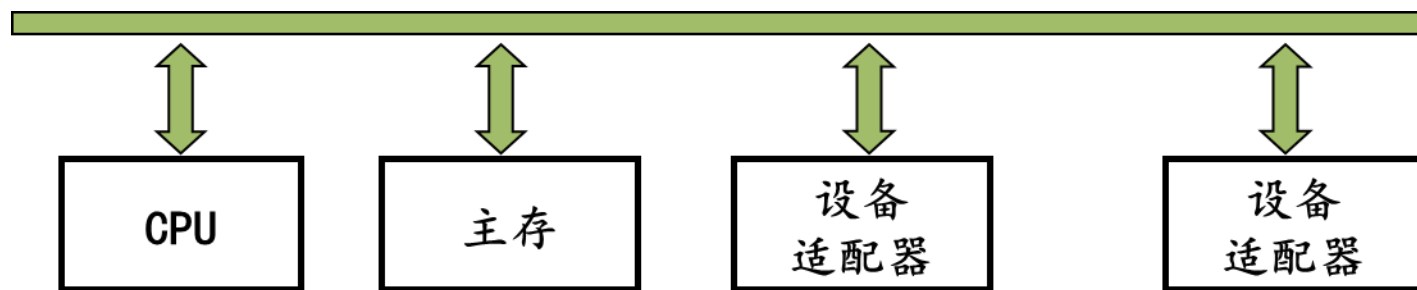
- 通常计算机系统都会追求总线的高带宽和高负载能力
- 但部分总线性能指标有可能是互相矛盾的
  - 例如并行总线时钟频率提高后，会引起线间串扰和时钟偏移问题，降低传输可靠性，从而影响总线带宽，另外还可能影响总线的负载能力
  - 在实际设计过程中会根据需求，技术，工艺水平等进行合理的这种考虑



# 总线的连接方式

## ■ 单总线结构

- 一条单一的系统总线来连接CPU、主存和IO设备
- 在单总线结构中，要求连接到总线上的逻辑部件必须高速运行，以便在某些设备需要使用总线时，能迅速获得总线控制权；而当不再使用总线时，能迅速放弃总线控制权



对IO设备的操作完全与主存一样  
判断指令的地址字段是主存还是外围设备

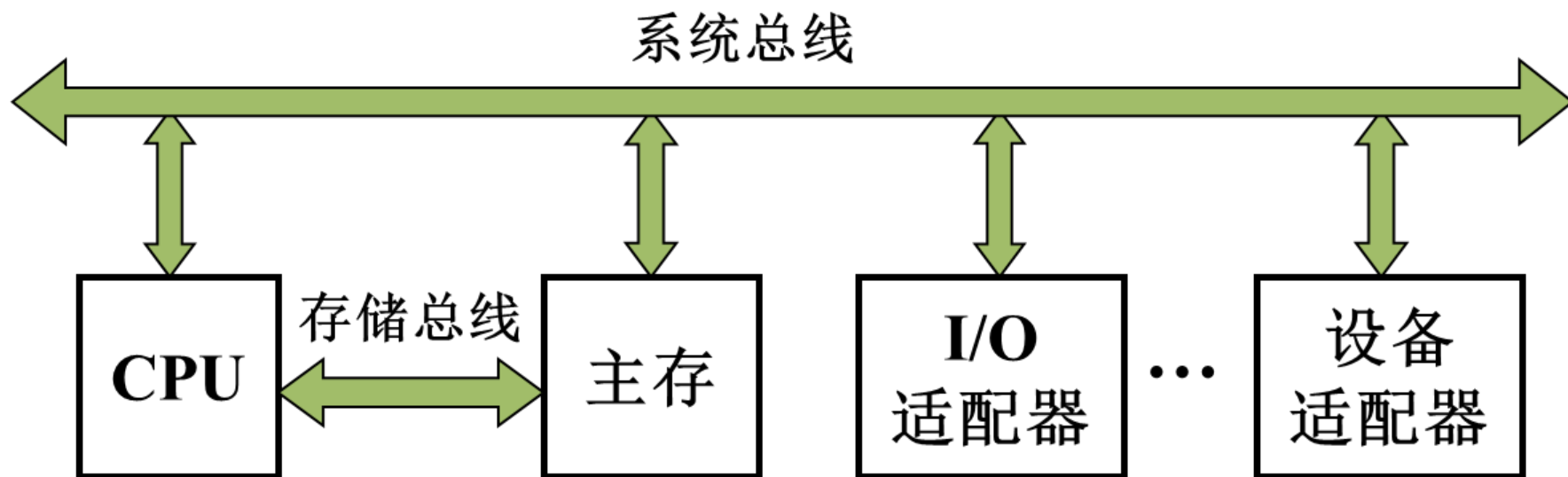
# 单总线结构

- 所有的高速设备和低速设备都挂在同一个总线上，且总线只能分时工作
  - 某一时间只能允许一对设备之间传送数据
- 结构简单，易于扩充
- 信息传送的效率和吞吐量受到极大限制

# 如何提高总线的效率和吞吐量？

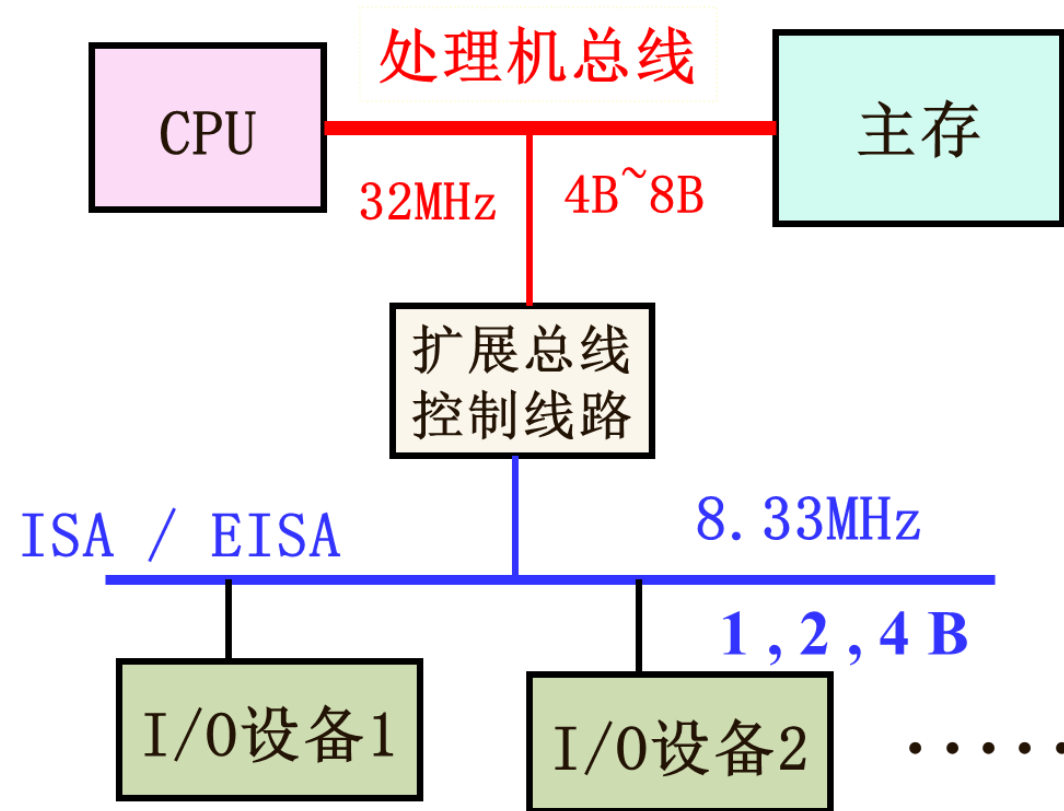
- 多个设备连接到总线上，总线的性能下降，随着总线上传输请求的增加，总线成为系统性能的瓶颈
- 总线上挂接设备数量的增长速度比总线数据传输率或者总线宽度的提高速度快得多
- 单纯依靠提高总线的数据传输率或使用更宽的总线已经不能解决总线的瓶颈问题

# 双总线结构



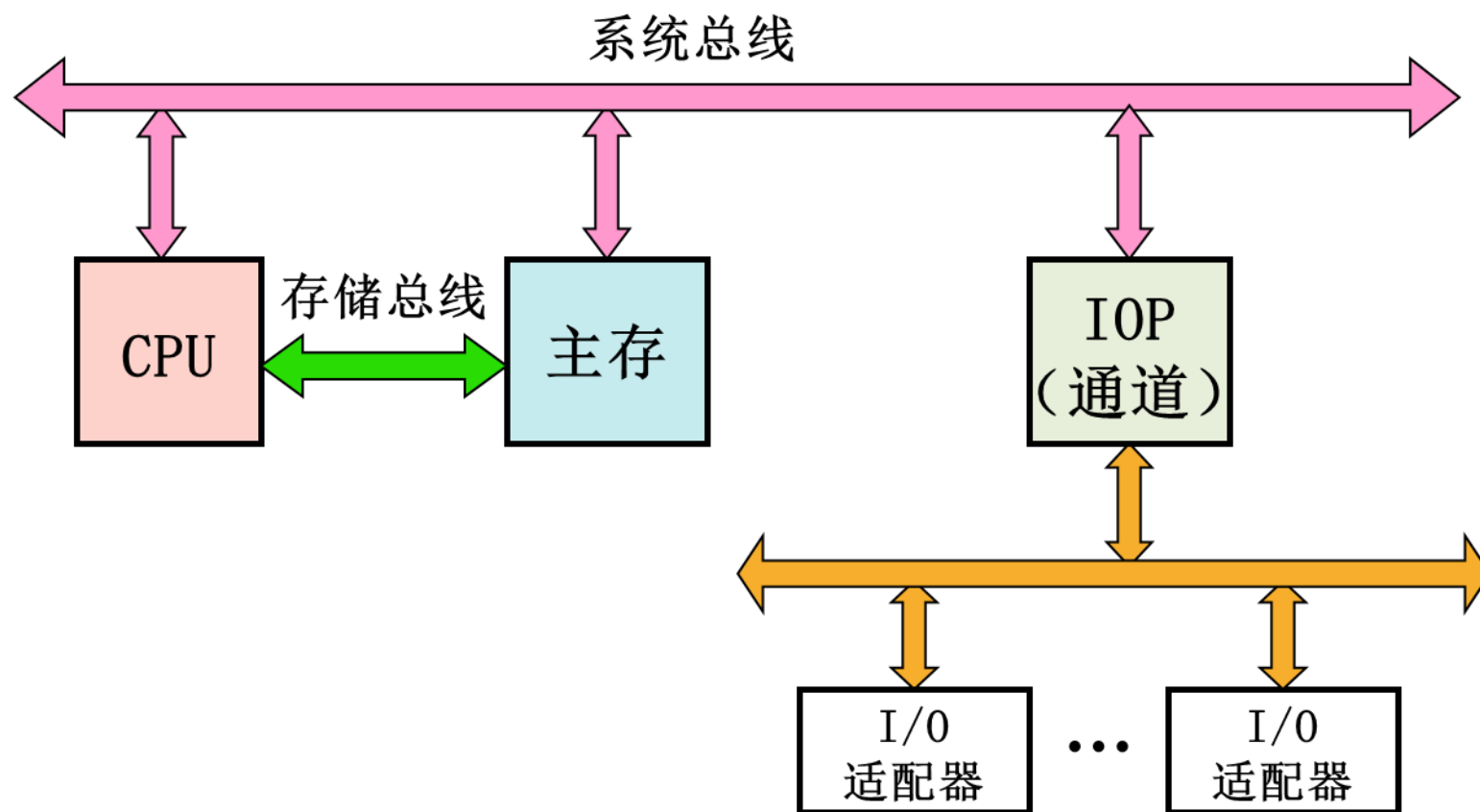
- CPU与内存有专用高速总线，减轻系统总线的负担；内存可通过系统总线与外设进行DMA操作，而不必经过CPU

# 双总线结构



**特点：**简单、易扩展、速度比单总线结构快。  
增加了硬件开销。

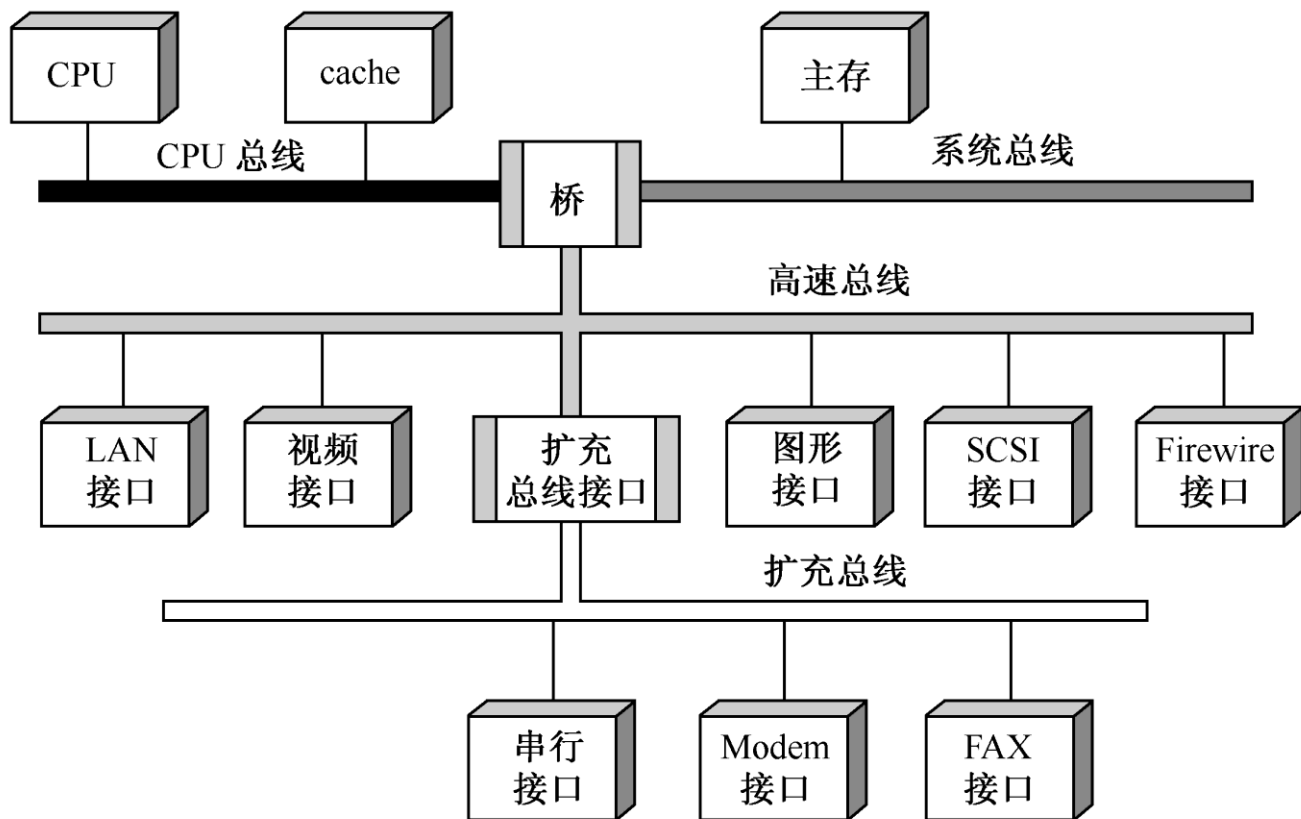
# 三总线结构



- 增加I/O总线，多个外设与通道间传递信息的通路
- 通道是一台具有特殊功能的处理器，统一管理外设及实现外设与内存的数据传送

# 总线的连接方式

- 多总线结构 **=》将高速、中速、低速设备连接到不同的总线上同时进行工作**
- 在CPU、主存、I/O之间互联采用多条总线

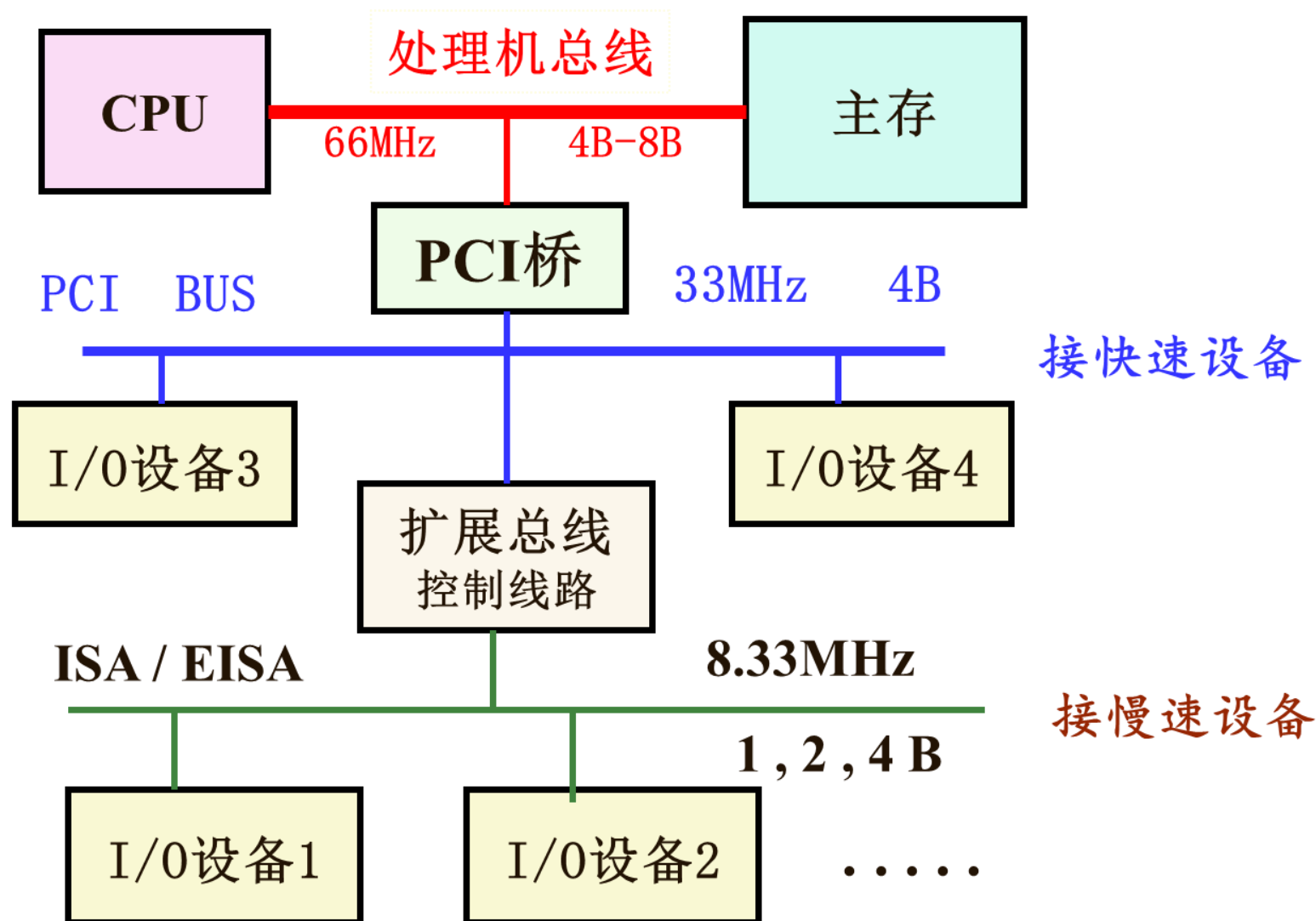


# 多总线结构

- **CPU总线：** CPU和cache之间采用
- **系统总线：** 主存连在其上
- **高速总线：** 连接高速外设，包括LAN（100Mb/s局域网）、视频接口、图形接口、SCSI接口、Fireware接口
- 高速总线通过扩充总线接口与扩充总线相连，扩充总线上可连接串行方式工作的I/O设备
- **桥：** 一种具有缓冲、转换、控制功能的逻辑电路，是不同速率总线之间的连接器件，起信号速度缓冲、电平转换、控制协议转换等作用



# 多总线结构示意图



# 总线结构对计算机系统性能的影响

## ■ 最大存储容量

- 在单总线系统中，最大主存容量必须小于计算机字长所决定的可能的地址总数
  - 32位地址总线最多允许访问多大的物理空间？
- 在多总线系统中，对主存和外设进行存取的判断是利用各自的指令操作码
  - 由于主存地址和外设地址出现于不同的总线上，所以存储容量不受外围设备多少的影响

# 总线结构对计算机系统性能的影响

## ■ 指令系统

- 在单总线系统中，访问主存和IO传送可使用相同的操作码及相同的指令，但它们使用不同的地址
- 在多总线系统中，CPU对存储总线、系统总线等必须有不同的指令系统

# 总线结构对计算机系统性能的影响

吞吐量：它是指流入、处理和流出系统的信息的速率。它主要取决于主存的存取周期和并行读出的位数。

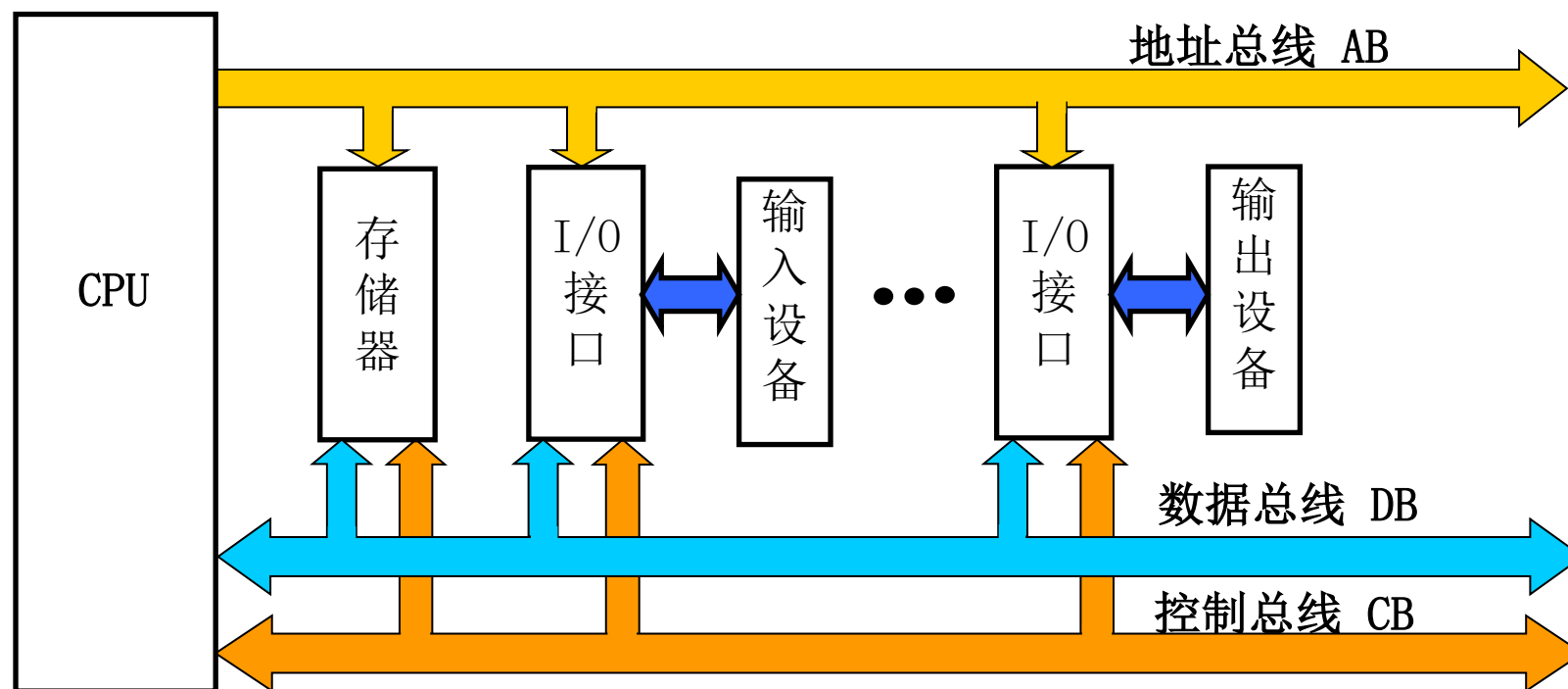
由于上述原因，采用双端口存储器可以增加主存的有效速度，可以更多的信息从内存输入输出。

三总线系统中，CPU的一部分功能下放给通道，由通道对外围设备统一管理并实现外围设备与内存之间的数据传送，因而系统的吞吐能力比单总线系统强。

# 总线的内部结构

- **早期总线的内部结构：** 它实际上是处理器芯片引脚的延伸，是处理器与I/O设备适配器的通道。

这种简单的总线一般也由50~100条线组成，这些线按其功能可分为三类：地址线、数据线和控制线。



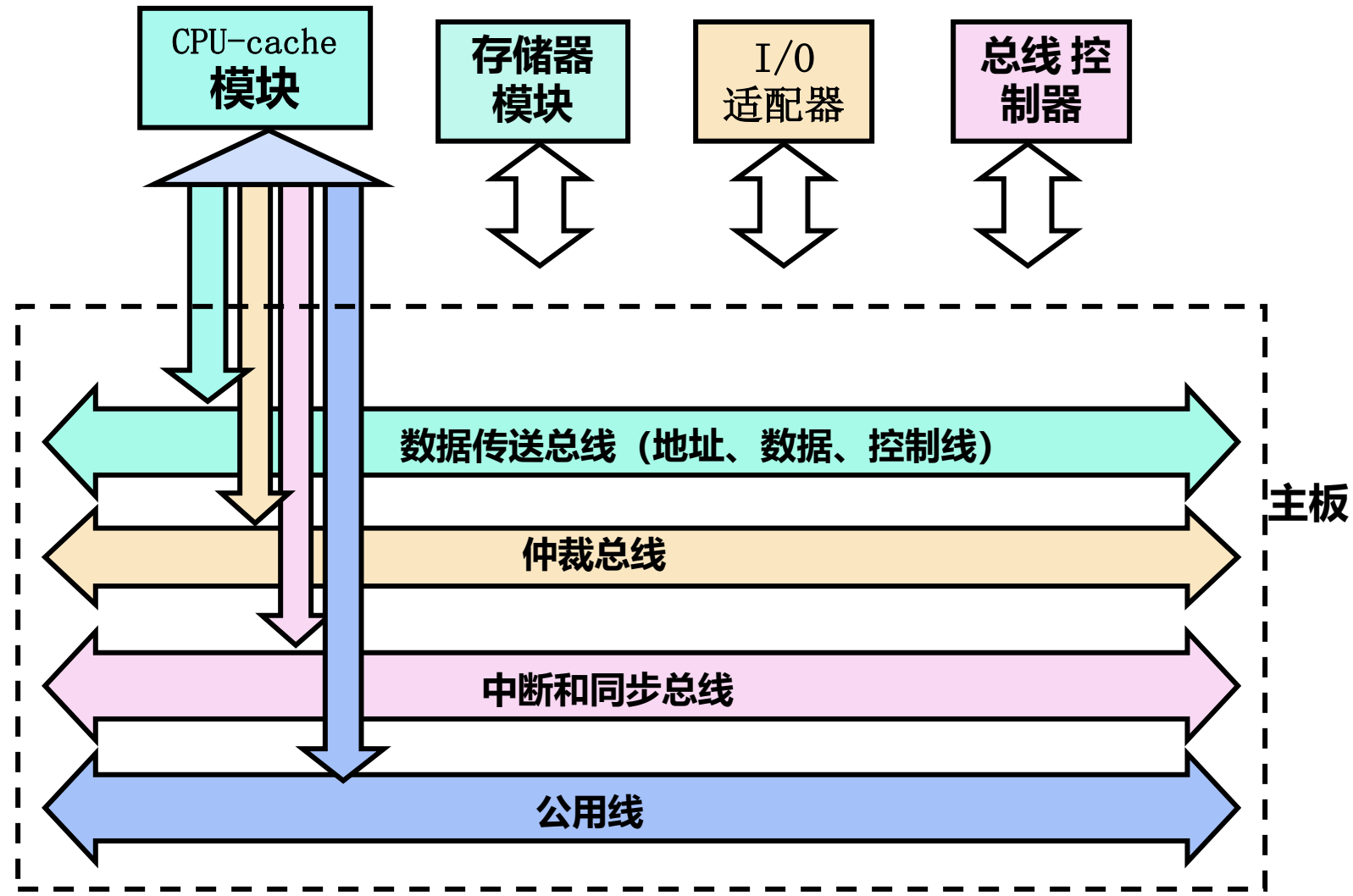
# 总线的内部结构

早期总线结构的不足之处在于：

- CPU是总线上唯一的主控者。即使后来增加了具有简单仲裁逻辑的DMA控制器以支持DMA传送，但仍不能满足多CPU环境的要求
- 总线信号是CPU引脚信号的延伸，故总线结构紧密与CPU相关，通用性较差

# 总线的内部结构

## ■ 当代流行的总线内部结构

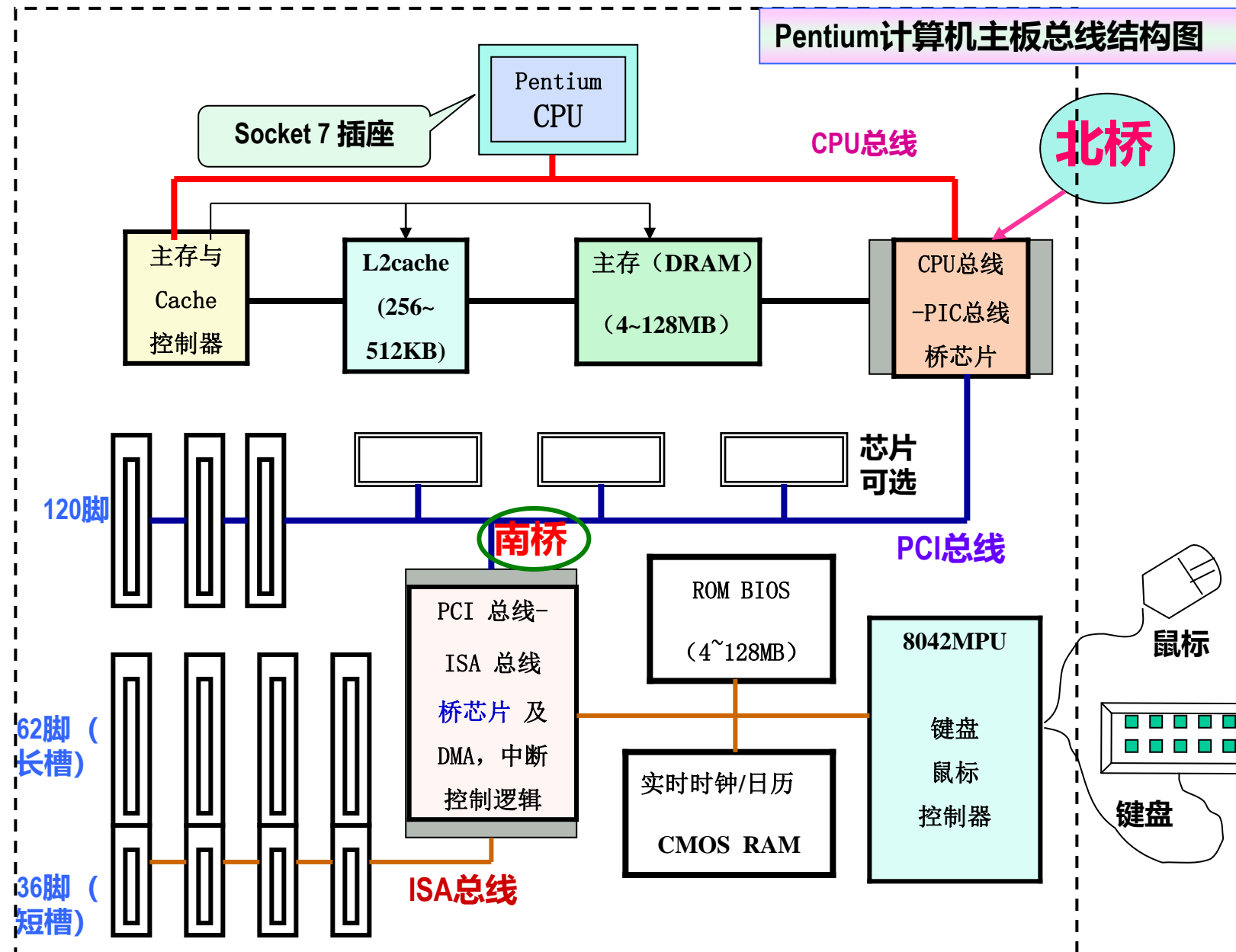


# 总线结构实例

- 大多数计算机采用了**分层次**的多总线结构。在这种结构中，速度差异较大的设备模块使用不同速度的总线，而速度相近的设备模块使用同一类总线

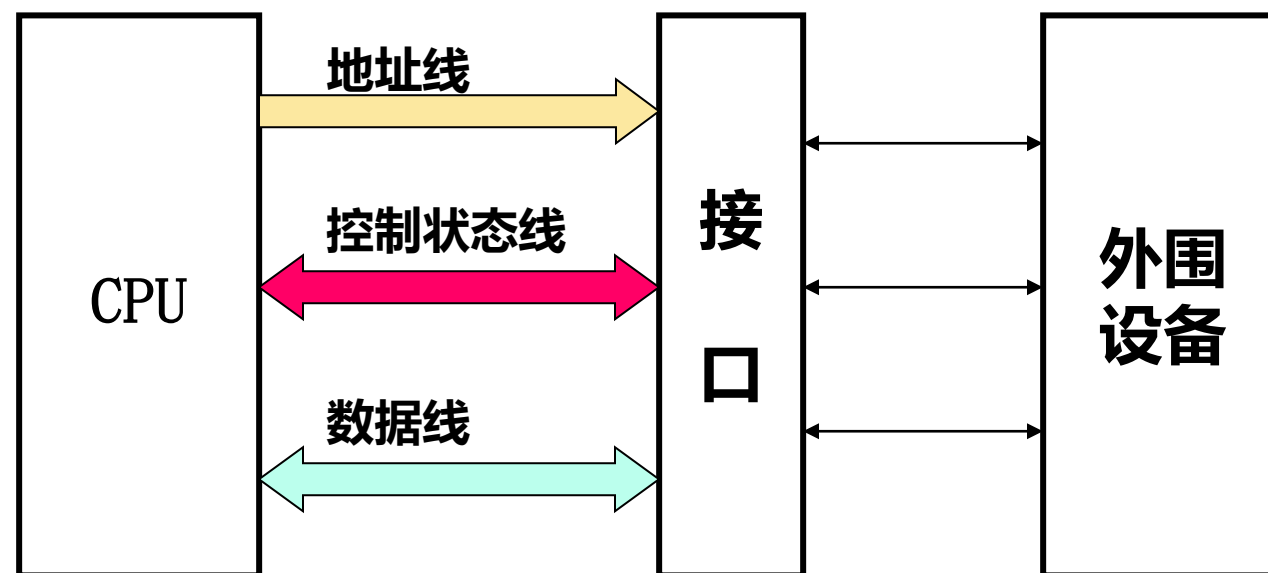


# 总线结构实例



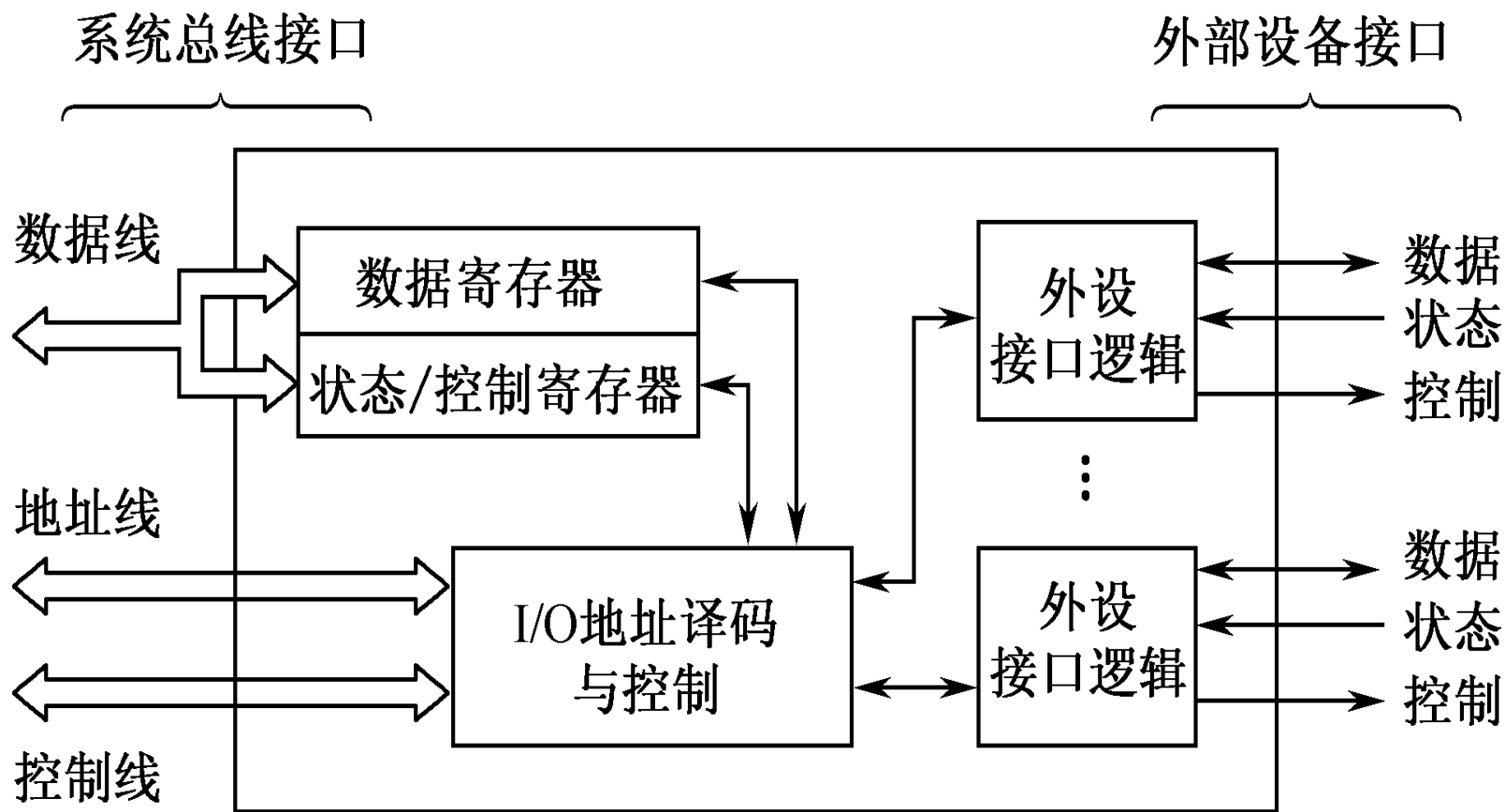
## 2. 总线接口

- 接口即I/O设备适配器，具体指CPU和主存、外围设备之间通过总线进行连接的逻辑部件
- 接口部件在它动态连接的两个部件之间起着转换器作用，以便实现彼此之间的信息传送



# 总线接口

## ■ 接口模块的结构框图



# 总线接口

## ■ 接口的典型功能

- **控制**：靠指令信息来控制外围设备的动作，如启动、关闭设备等
- **缓冲**：补偿各种设备在速度上的差异
- **状态**：监视外围设备的工作状态并保存状态信息
- **转换**：完成任何要求的数据转换，使数据能够在外围设备和CPU之间正确传送
- **整理**：完成一些特别的功能，例如修改字计数器
- **程序中断**：发生一个中断请求信号到CPU

# 总线接口

- 适配器的两面性，必有两个接口
  - 一个是和系统总线的接口，CPU和适配器的数据交换一定是并行方式
  - 二是和外设的接口，适配器和外设的数据交换可能是并行方式，也可能是串行方式。根据外围设备供求串行数据或者并行数据方式的不同，适配器也分为**串行数据接口**和**并行数据接口**两大类

## 3. 总线仲裁

### ■ 总线主设备和从设备

- 总线主设备：**发起**总线传输的模块/设备
  - 能够申请并获得总线使用权的设备
  - 具有控制总线的能力，发起总线事务
- 总线从设备：**响应**总线传输的模块/设备
  - 不具有申请总线使用权的设备
  - 被总线事务激活的模块或设备

# 总线的仲裁

- 连接到总线上的功能模块有主动和被动两种形态
  - 主方可以启动一个总线周期，从方只能响应主方请求。每次总线操作，只能有一个主方，但是可以有多个从方
- 为了解决多个主设备同时竞争总线控制权，必须具有总线仲裁部件，以某种方式选择其中一个主设备作为总线的下一次主方
  - 优先级或者公平策略
- **总线占用期**：主方持续控制总线的时间。
- 按照总线仲裁电路的位置不同，仲裁方式分为集中式和分布式

# 集中式仲裁

集中式仲裁中每个功能模块有两条线连到中央仲裁器：一条是送往仲裁器的总线请求信号线 **BR**，一条是仲裁器送出的总线授权信号线 **BG**。

集中式仲裁又可分为：

链式查询方式；

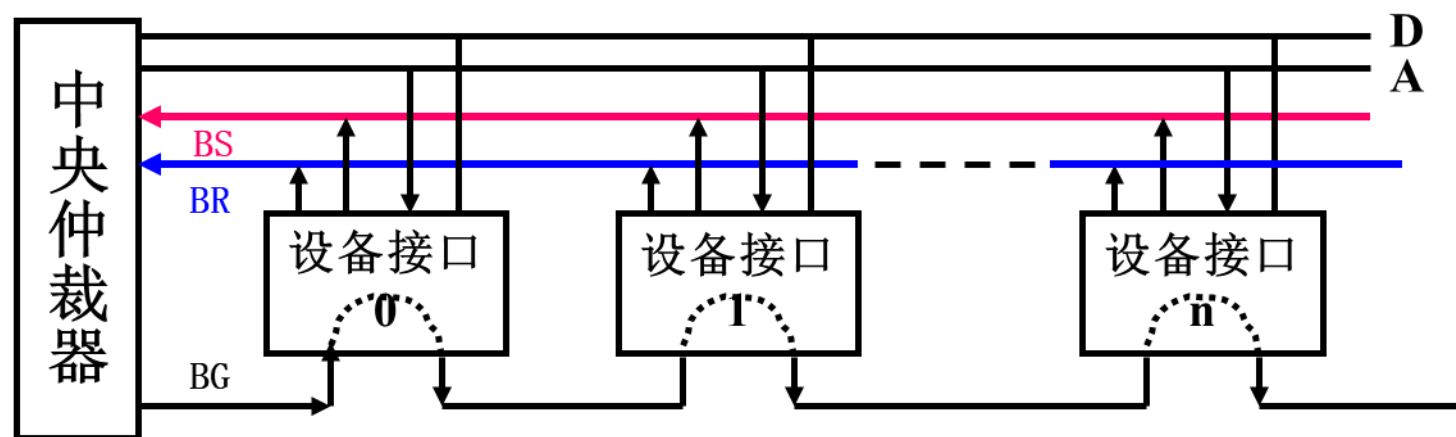
计数器定时查询方式；

独立请求方式。



# 集中式仲裁：链式查询

- **链式查询方式：**离中央仲裁器最近的设备具有最高优先权，离总线控制器越远，优先权越低



链式查询方式

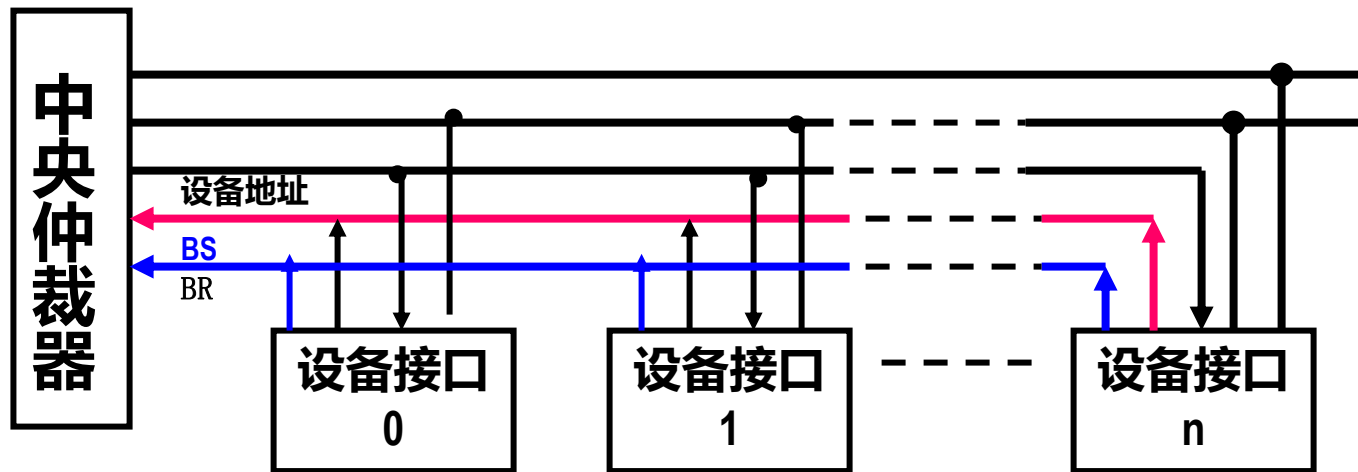
- 优点：只用很少几根线就能按一定优先次序实现总线控制，且这种链式结构很容易扩充设备
- 缺点：对询问链的电路故障敏感，优先级固定

# 集中式仲裁：计数器定时查询

中央仲裁器接到请求信号以后，在BS线为“0”的情况下让计数器开始计数，计数值通过一组地址线发向各设备。

每个设备接口都有一个设备地址判别电路，当地址线上的计数值与请求总线的设备地址相一致时，该设备置“1”BS线，获得了总线使用权，此时中止计数查询。

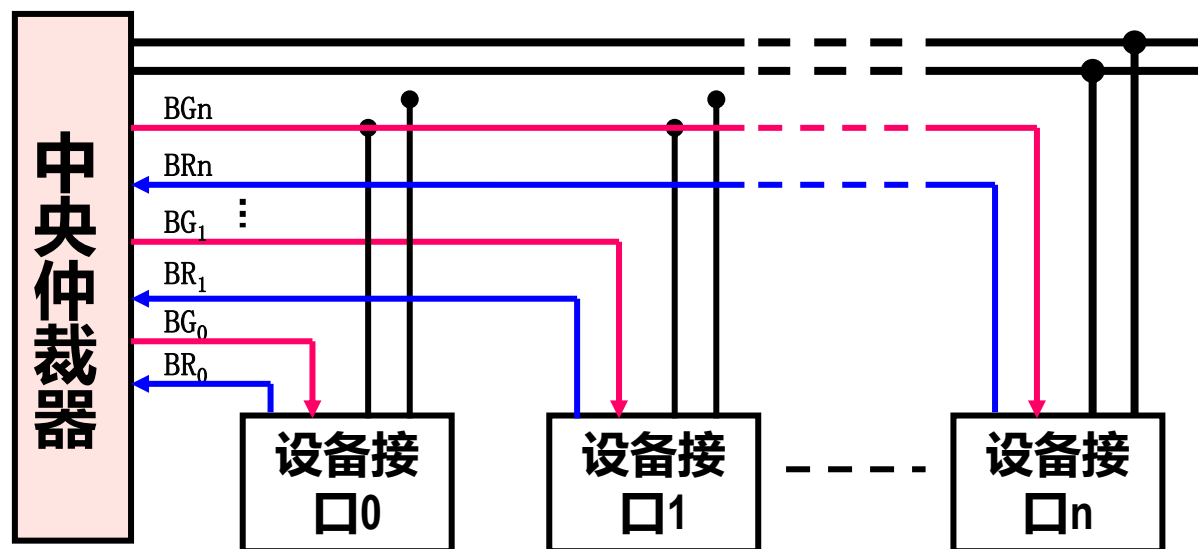
**优点：**计数器的初值也可用程序来设置，优先次序可以方便地改变。



计数器定时查询方式

# 集中式仲裁：独立请求方式

- 在独立请求方式中，每一个共享总线的设备均有一对总线请求线 $BR_i$ 和总线授权线 $BG_i$
- 当设备要求使用总线时，便发出该设备的请求信号。总线仲裁器中有一个排队电路，它根据一定的优先次序决定首先响应哪个设备的请求，给设备以授权信号 $BG_i$



# 集中式仲裁：独立请求方式

- 独立请求方式的优点：
  - **响应时间快**：确定优先响应的设备所花费的时间少，不用一个设备接一个设备地查询
  - **对优先次序的控制相当灵活**：它可以预先固定，例如BR0优先级最高，BR1次之…BRn最低；也可以通过程序来改变优先次序；还可以用屏蔽（禁止）某个请求的办法，不响应来自无效设备的请求。
- 但控制线数量多，总线控制更为复杂
- **当代总线标准普遍采用独立请求方式**

# 集中式仲裁

- 链式查询中仅用 **根线** 确定总线使用权属于哪个设备
- 在计数查询中大致用 **根线**，其中 $n$ 是允许接纳的最大设备数
- 独立请求方式需采用 **根线**

# 分布式仲裁

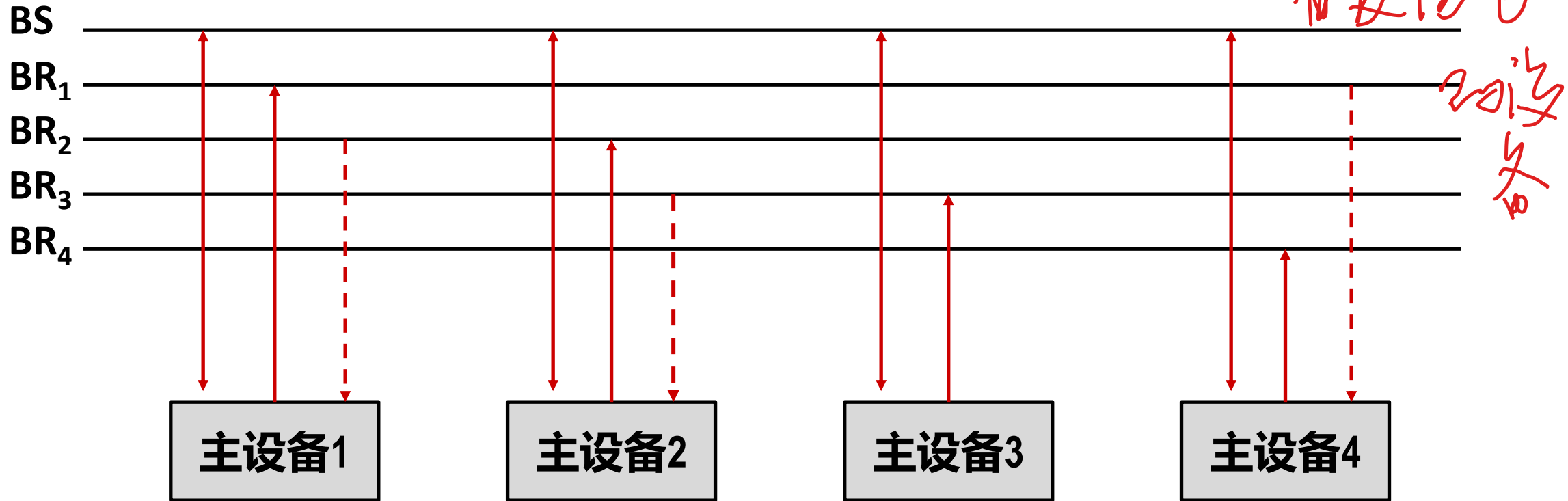
- 分布式仲裁控制逻辑分散在总线的各设备中，不需要中央仲裁器，每个主设备都有自己的仲裁号和仲裁器
- 自举分布式仲裁
- 并行竞争仲裁
- 冲突检测分散式仲裁

# 自举分布式仲裁

- **原理：每个设备独立地决定是否是否是最高优先级请求者**
- **每个设备有一根自己的总线请求线，每个需要总线请求的设备在各自的总线请求线上发出请求信号，同时接收其他设备的总线请求信号**
  - 若没有接收到优先级比自己高的设备的总线请求信号
    - 且此时“总线忙”信号无效，则该设备可以立即使用总线，并发出“总线忙”信号以阻止其他设备使用总线。
  - 若检测到其他优先级更高的设备也请求使用总线
    - 则该设备放弃本次对总线的请求。

# 自举分布式仲裁

看是否有设备发信号，有的话就放弃，  
 3214  
 看发信号  
 到设备



- 图中四个主设备的优先级顺序是？

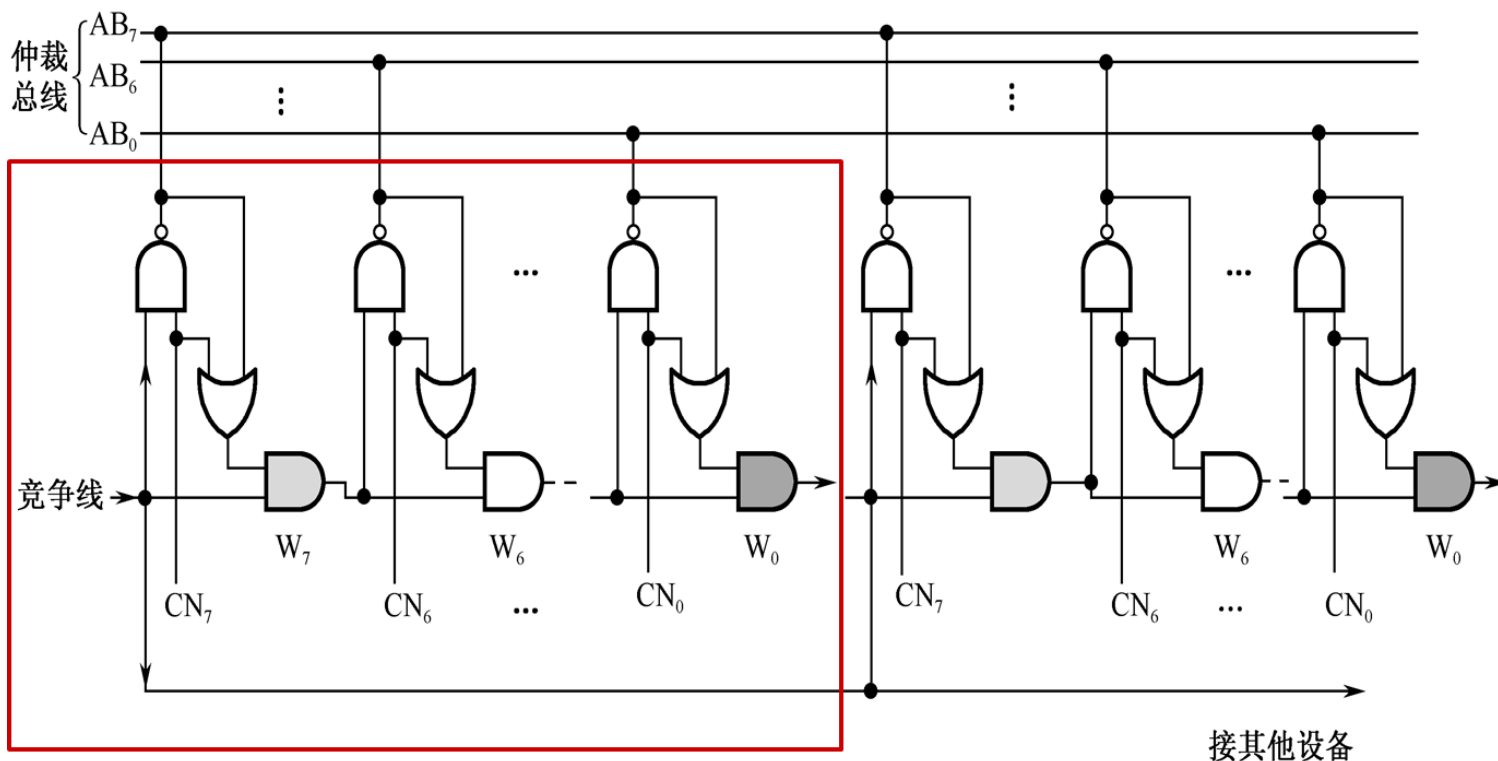


# 并行竞争仲裁

- 每个主设备具有专属的仲裁号和仲裁器
- 当它们有总线请求时，把它们的仲裁号发送到共享仲裁总线上，每个仲裁器将仲裁总线上得到的号与自己的号进行比较。
- 如果仲裁总线上的号大，则它的总线请求不予响应，并撤消它的仲裁号。最后，获胜者的仲裁号保留在仲裁总线上
- 分布式仲裁是以**优先级仲裁策略**为基础

# 并行竞争仲裁示意图

从高位到低比较



- $CN_7 \dots CN_0$  设备仲裁号，为1则竞争请求
- 所有参加竞争的主设备将仲裁号CN取反后打到仲裁总线AB上，实现“线或”逻辑
- $Ab_i=0$ 说明总线此位目前有总线请求

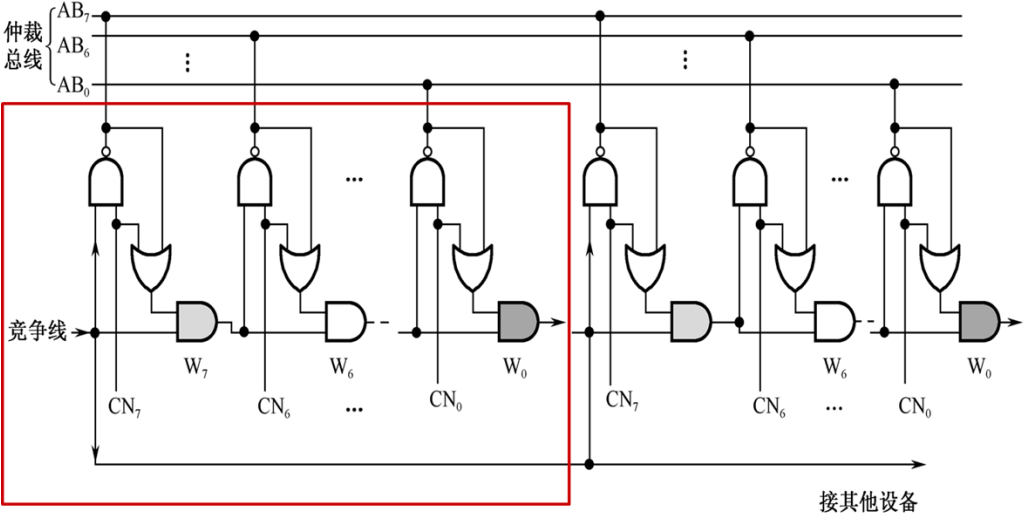
有一个1就为0

- ① 如 $CN_i=0$ ,  $Ab_i=0$ , 则竞争失败,  $W_i=0$
- ② 如 $CN_i=0$ ,  $Ab_i=1$ , 则此位无竞争,  $W_i=1$
- ③ 如 $CN_i=1$ ,  $Ab_i=1$ , 则竞争成功,  $W_i=1$
- ④ 如 $CN_i=1$ ,  $Ab_i=0$ , 则继续竞争下一位,  $W_i=1$

# 并行竞争仲裁示意图

- 假设两个设备同时要求使用总线，仲裁号分别是00000101和00001010
- 最终留在仲裁线上的号为00001010

裁决号1		裁决号2		裁决线电平	裁决线逻辑
cn	AB	cn	AB		
0	高	0	高	高	0
0	高	0	高	高	0
0	高	0	高	高	0
0	高	0	高	高	0
0	高	1	低	低	1
1	高	0	高	高	0
0	高	1	低	低	1
1	高	0	高	高	0



每个单独考虑 而不是两个一起比较

# 冲突检测分散式仲裁

## ■ 基本思想

- 当某个设备需要使用总线时，它首先检查一下有没有其他设备正在使用总线
  - 如果没有，那它就置总线忙，然后使用总线
- 若两个设备同时检测到总线空闲，则可能会同时使用总线，此时发生冲突
  - 一个设备在传输过程中，它会侦听总线以检测是否发生了冲突
  - 当冲突发生时，两个设备都停止传输，延迟一个随机时间后再重新请求
- 一般在网络通信总线上，如Ethernet总线