

第3章 系统总线

大纲要求

(一) 总线概述

1. 总线的基本概念
2. 总线的分类
3. 总线的组成及性能指标

(二) 总线仲裁

1. 集中仲裁方式
2. 分布仲裁方式

(三) 总线操作和定时

1. 同步定时方式
2. 异步定时方式

第三章 系统总线

1

3.1 总线的基本概念

2

3.2 总线的分类

3

3.3 总线特性及性能指标

4

3.4 总线结构

5

3.5 总线控制

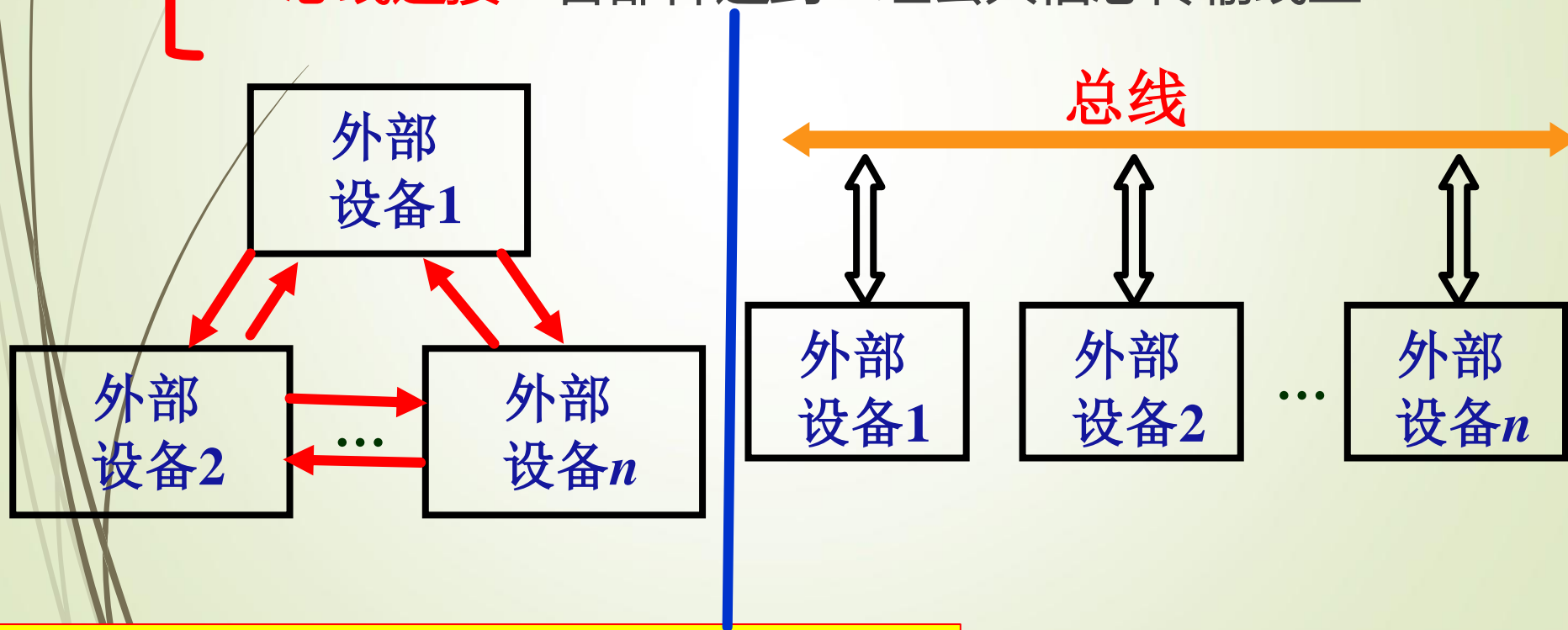
3.1 总线的基本概念

为什么要用总线

计算机系统部件之间的连接方式

分散连接：各部件之间通过单独的连线连接（早期方式）

总线连接：各部件连到一组公共信息传输线上



3.1 总线的基本概念

一、为什么要用总线

- ➡ 减少各部件之间的连接线;
- ➡ 方便系统的构成、扩充或更新。

各部件和总线采用挂接形式，增加一个部件或去掉一个部件对整个系统的结构不会造成什么影响。

总线技术使计算机的设计、生产走向标准化。

3.1 总线的基本概念

二、什么是总线

总线是连接各个部件的信息传输线

是各个部件共享的传输介质

可以看作信息传输的**高速公路**

三、总线上信息的传送

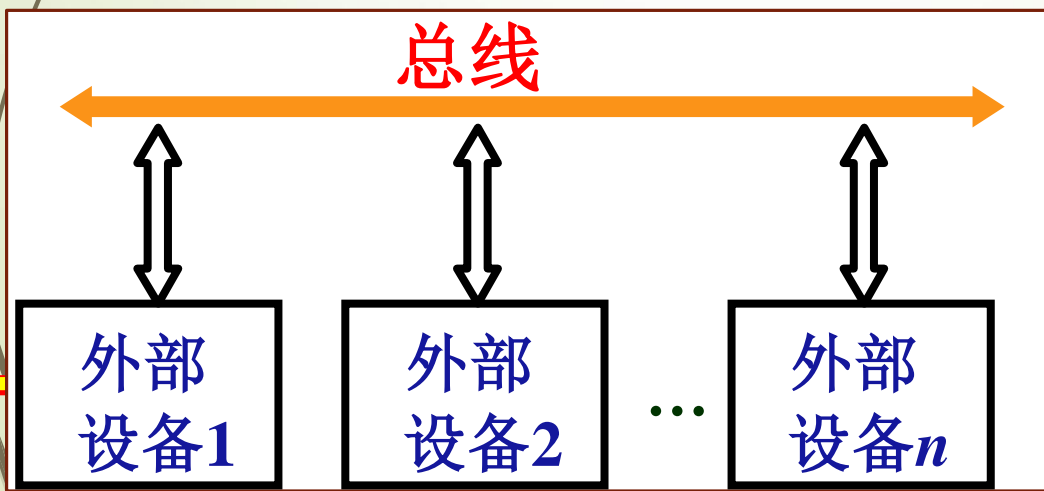
串行

[Home](#)
[About Us](#)
[Contact Us](#)
[Privacy Policy](#)
[Terms of Service](#)
[FAQ](#)
[Blog](#)
[Partners](#)
[Press](#)
[Careers](#)

并行

【例】挂载在总线上的多个部件()。

- ☐ A 只能分时向总线发送数据，并只能分时从总线接收数据
- ☒ B 只能分时向总线发送数据，但可同时从总线接收数据
- ☐ C 可同时向总线发送数据，并同时从总线接收数据
- ☐ D 可同时向总线发送数据，但只能分时从总线接收数据



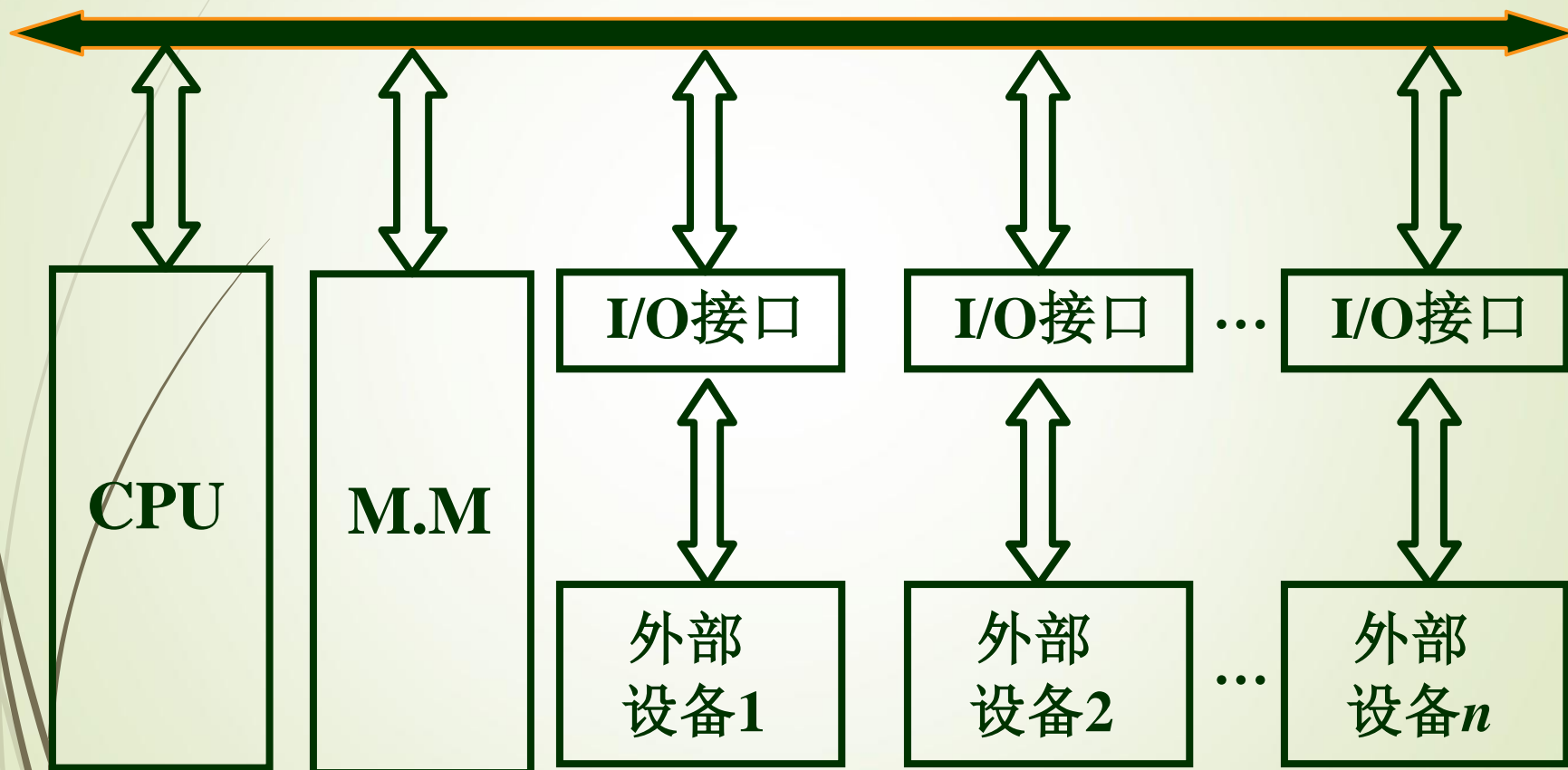
提交

六、总线结构计算机举例

1. 单总线结构框图

优点：结构简单
缺点：快速设备和慢速设备挂在一条总线上。

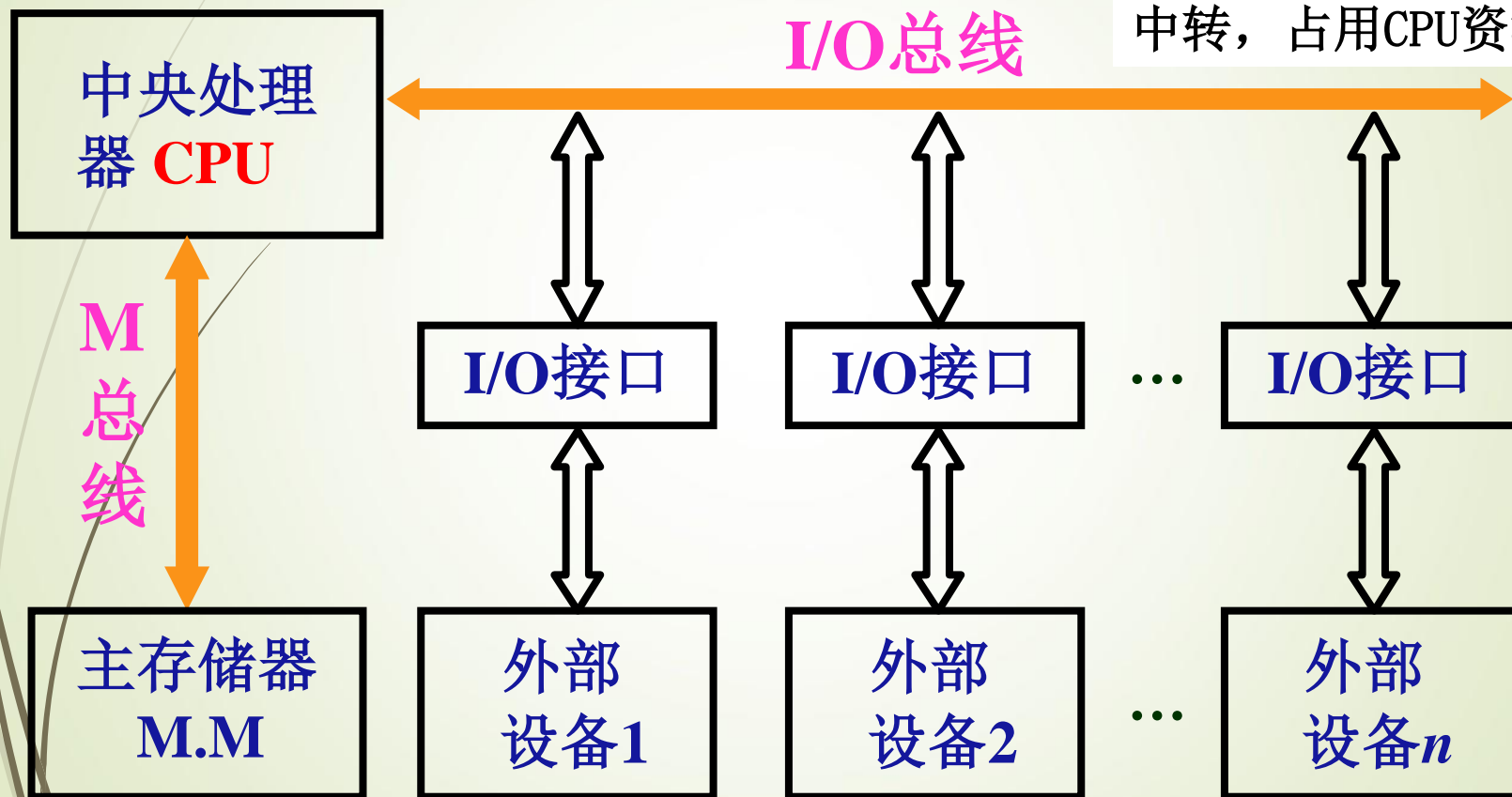
单总线（系统总线）



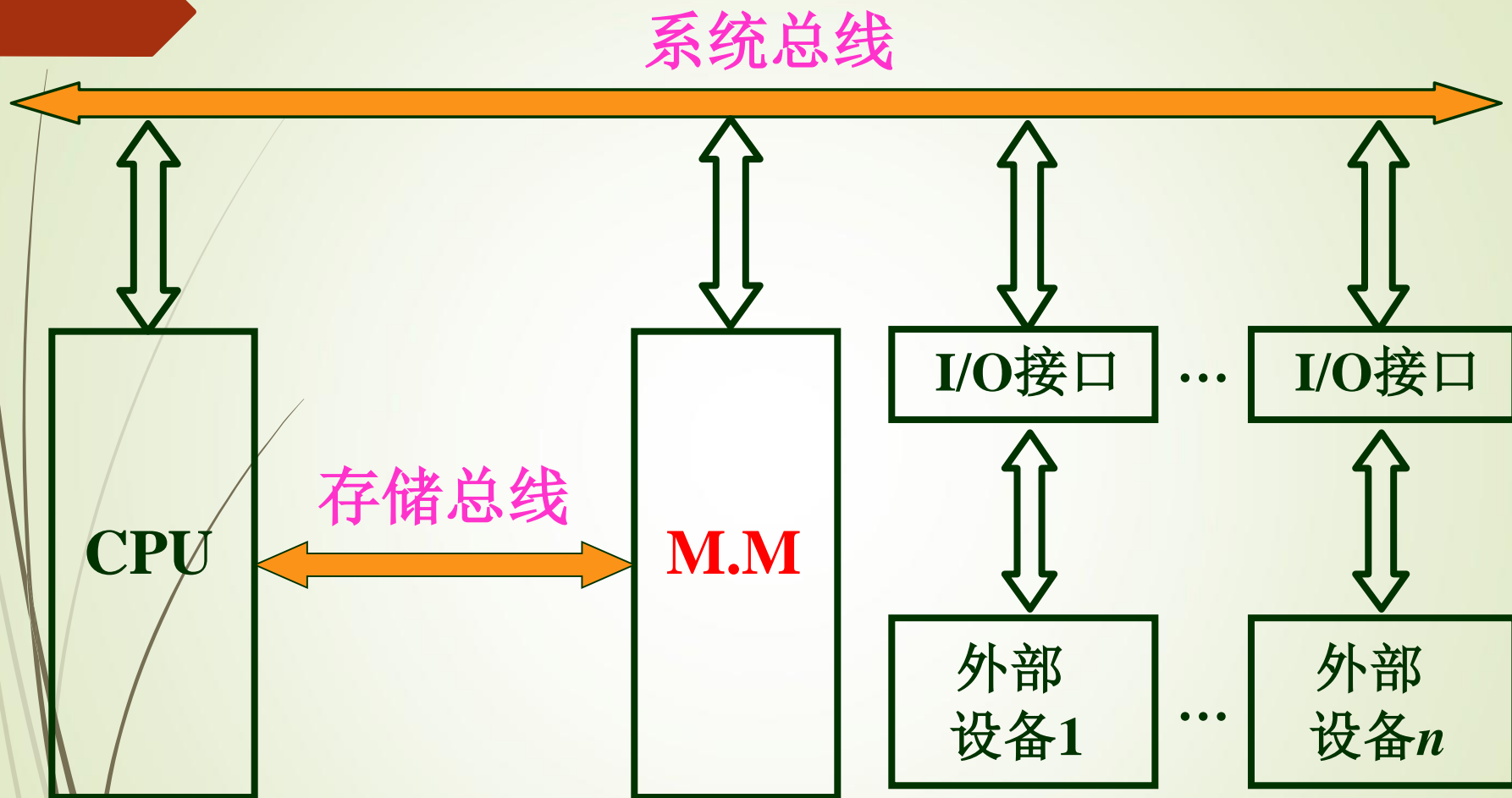
六、总线结构计算机举例

2. 面向 CPU 的双总线结构框图

优点：快速设备和慢速设备挂在两条总线上。
缺点：结构复杂，外设1送数据给主存要经过CPU中转，占用CPU资源。



3. 以存储器为中心的双总线结构框图



优点：快速设备和慢速设备挂在两条总线上。外设1送数据给主存不需要经过CPU中转。

缺点：结构复杂。

3.2 总线的分类

1. 片内总线

芯片内部 的总线

2. 系统总线

计算机各部件之间 的信息传输线

数据总线

双向 与机器字长、存储字长有关

地址总线

单向 与存储地址、I/O地址有关

存储单元个数 = $2^{\text{地址总线根数}}$

控制总线

有入 有出, 发出各种控制信号, 监视各部件状态

中断请求、
总线请求

存储器读、存储器写
总线允许、中断确认

【例】使用ISA总线（20位地址线）允许寻址的主存空间有多大？使用PCI总线（32位地址线）允许寻址的主存空间又有多大？（ ）

- ☒ A 1MB, 4GB
- ☐ B 4GB, 1MB
- ☐ C 1MB, 4MB
- ☐ D 4GB, 1GB

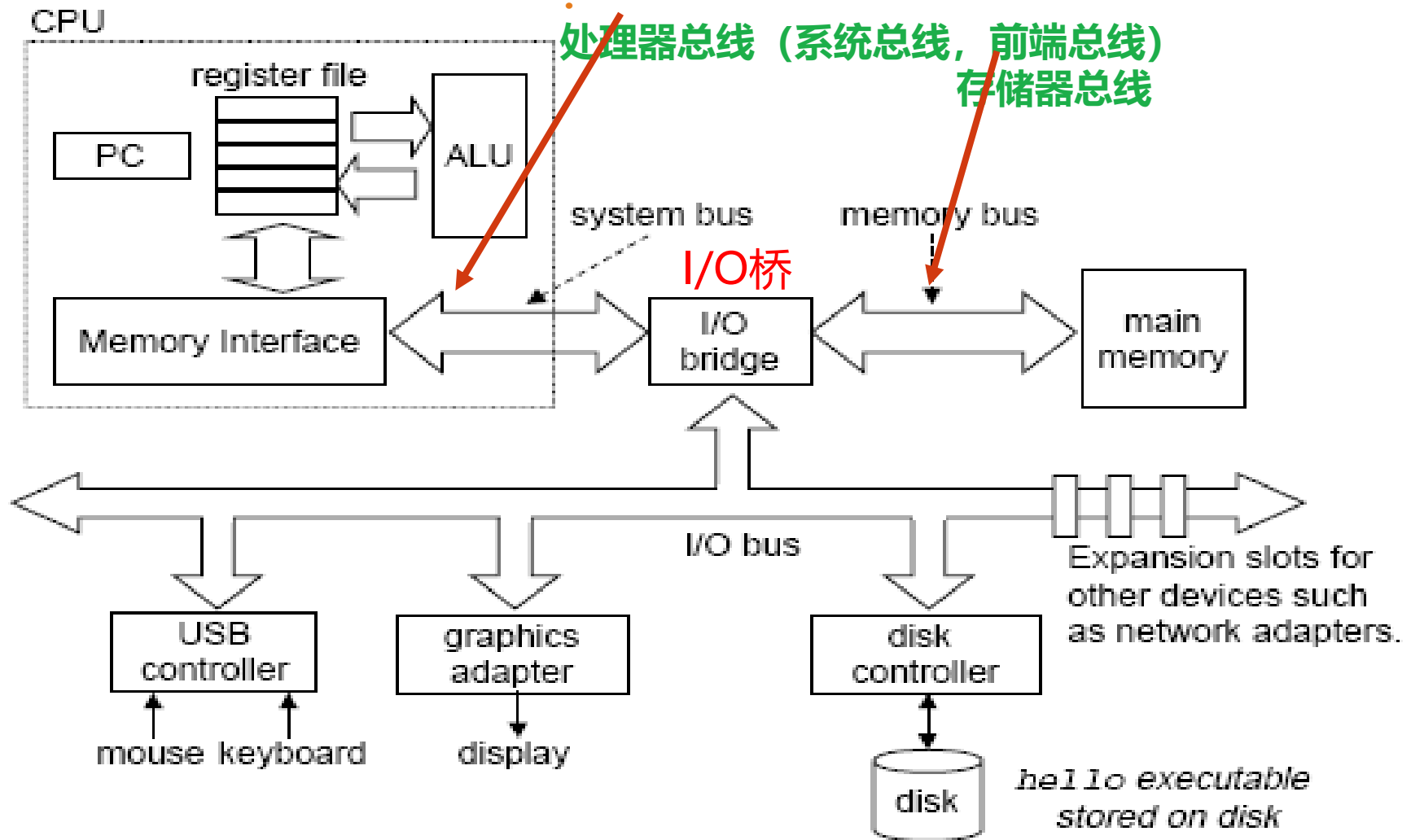
提交

注：Intel公司在推出845、850等芯片组时，对“System Bus”有专门的定义，将处理器总线称为前端总线(Front Bus)或系统总线

Intel 体系结构中特指的“系统总线”

北桥芯片组把处理器-存储器总线分成了两个总线

处理器总线 (系统总线, 前端总线)
存储器总线



3.2 总线的分类

3. 通信总线（外部总线、I/O 总线）

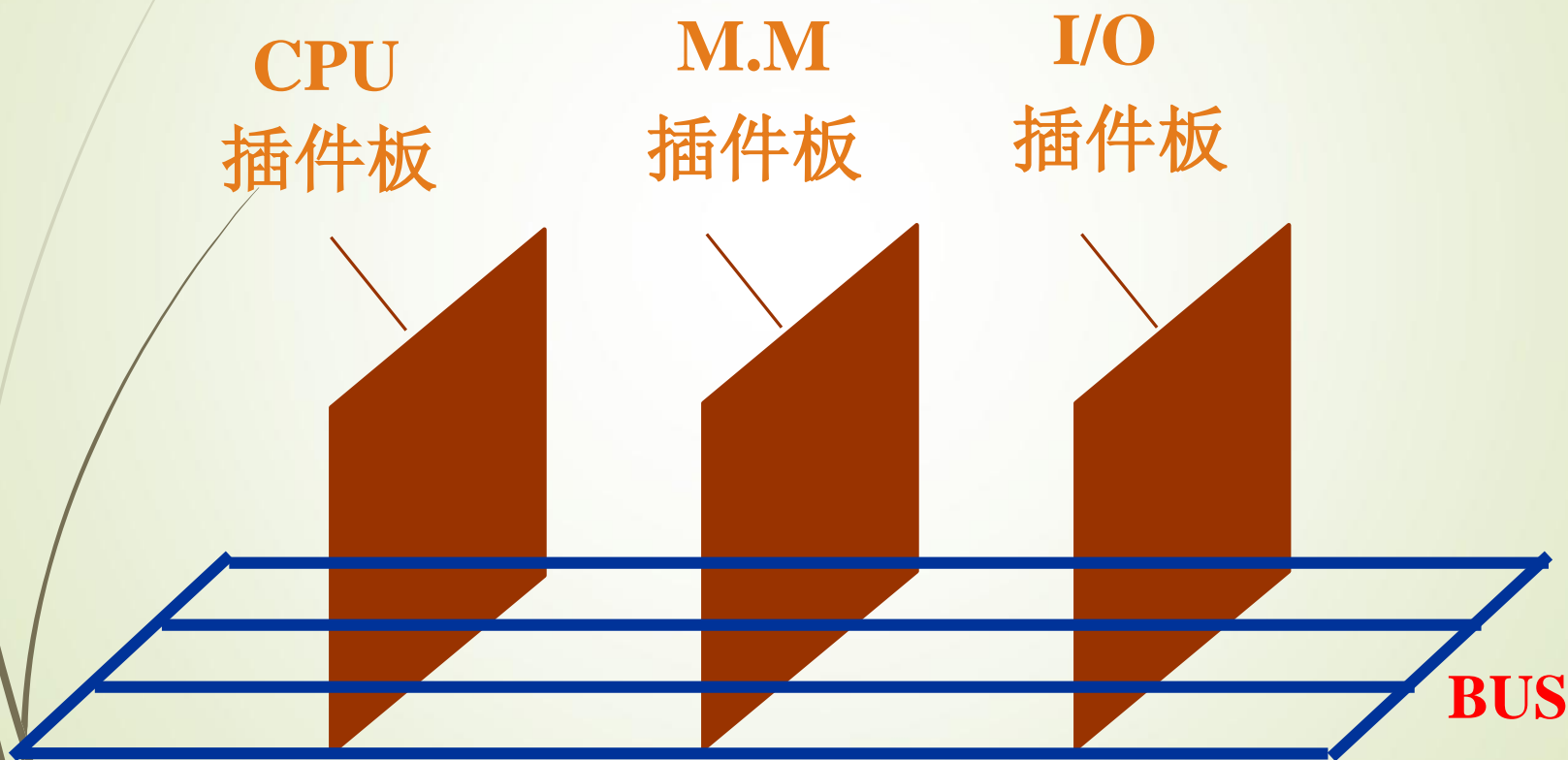
用于 计算机系统之间 或 计算机系统
与其他系统（如控制仪表、移动通信等）
之间的通信

-通常是**电缆式总线**，如**SCSI、RS-232、USB**等

传输方式 { **串行通信总线**
并行通信总线

3.3 总线特性及性能指标

一、总线物理实现



3.3 总线特性及性能指标

1. 总线周期 一次总线操作所需的时间
2. 总线工作频率 协调总线上各种操作的时钟频率
3. 总线宽度 数据线的根数
4. 总线带宽
(Bus Bandwidth) 每秒传输的最大字节数 (MBps)

(总线最大数据传输率、传输带宽)

总线带宽 = (总线宽度/8) × 总线工作频率 (MHz)
5. 时钟同步/异步 同步、不同步

3.3 总线特性及性能指标

6. 总线复用

地址线 与 数据线 复用

7. 信号线数

地址线、数据线和控制线的 总和

8. 总线控制方式

突发工作、自动配置、仲裁方式等

9. 其他指标

负载能力、电源电压（5 or 3.3）、
总线宽度扩展性

某总线在一个总线周期中并行传送4个字节的数据，假设一个总线周期等于一个时钟周期，总线时钟频率为33MHz，总线带宽是多少？如果一个总线中并行传送64位数据，总线时钟频率升为66MHz，总线带宽是多少？（ ）

- ☐ A 132 MBps , 4224 MBps
- ☐ B 1056 MBps , 4224 MBps
- ☒ C 132 MBps , 528 MBps
- ☐ D 33 MBps , 66 MBps

提交

名词解释

- 波特率**：每秒钟通过信道传输的码元数（二进制位数），单位为bps(位/秒)，记为波特。(每秒内线路状态的改变次数)
- **比特率**：每秒钟通过信道传输的信息量（有效数据位数）
 - **猝发式数据传输**（burst mode）：是一种总线传输方式，即在一个总线周期传输存储地址连续的多个消息。
 - **消息传输**：将总线需要传送的数据信息、地址信息和控制信息等合成一个固定的数据结构，以猝发方式进行传输。
 - **总线协议**：总线通信同步方式规定了实现总线数据传输的定时规则，即总线协议。
 - **总线事务**(Bus Transaction)：从请求总线到完成总线使用的操作序列。如：存储器读、存储器写、I/O读、I/O写、读指令、中断响应等

总线的功能：沟通计算机各部件信息传递，并使不同厂商提供的产品能互换组合。

总线标准：

- 正式公布的标准：由国际电工委员会(IEC)、国际电报电话咨询委员会 (CCITT)、电气电子工程师协会 (IEEE)、国际电信联盟标准化部门(ITU-T)国际标准化组织 (ISO) 等国际组织正式确定和承认，并有严格的定义。
- 实际的工业标准：首先由某一厂家提出而又得到其它厂家广泛使用，这类标准可能还没有经过正式、严格的定义，也可能经过一段时间后交给有关组织讨论而被确定为正式标准。

- **总线标准**是系统与各模块、模块与模块之间的一个互连的标准界面。

- **总线标准规范：**

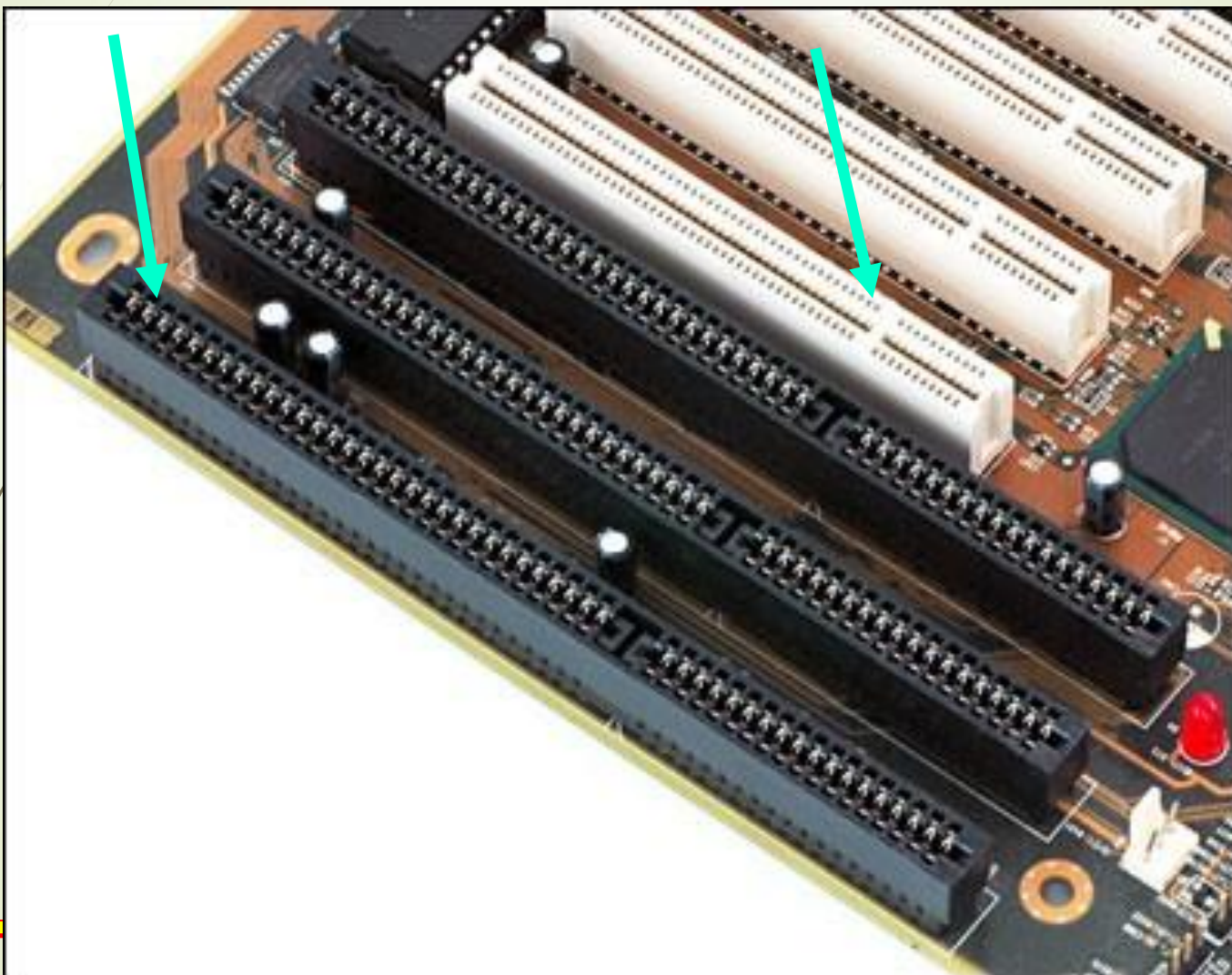
- **机械结构规范：**确定模板尺寸、总线插头，边沿连接器等的规格及位置。
- **功能规范：**确定各引脚的名称、定义、功能与逻辑关系。
- **电气规范：**规定信号工作时的高低电平、动态转换时间、负载能力以及最大额定值。

常用工业标准总线：

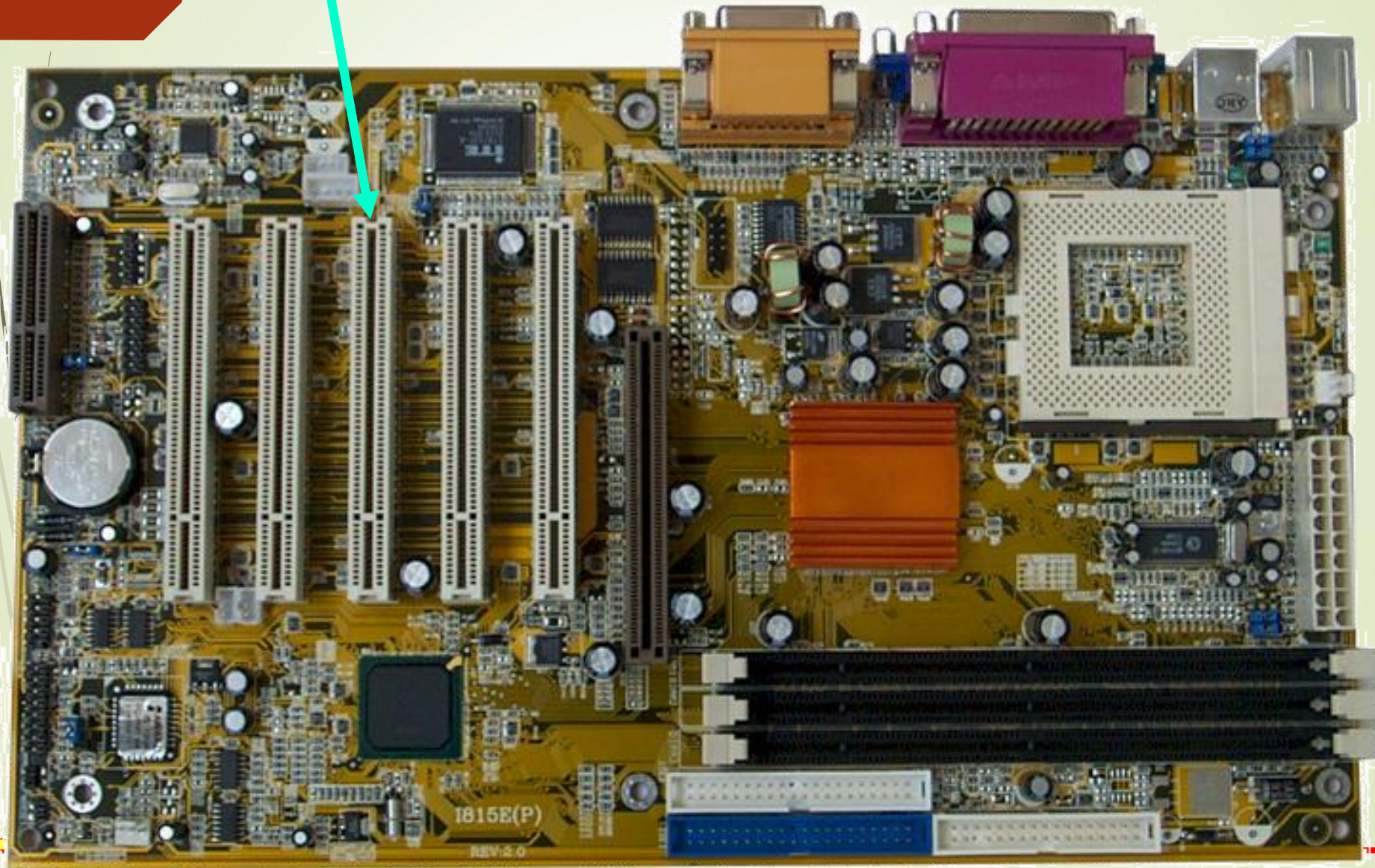
总线标准	数据线	总线时钟	带宽
ISA	16	8MHz(独立)	16MBps
EISA	32	8MHz(独立)	33MBps
VESA	32	32MHz(CPU)	133MBps
PCI	32/64	33/66MHz(独立)	132/528MBps
AGP	32	66.7/133MHz(独立)	266MBps(AGP 1x) 533MBps(AGP 2x) 1GBps(AGP 4x) 2.1GBps(AGP 8x)
RS-232	串行通信		数据终端设备和数据通信设备之间的接口
USB	串行接口		1.5~12Mbps(USB1.0) 480Mbps(USB2.0)
PCI Express	8,点对点串行连接的设备连接方式	2.5 GHz	支持双向多通道，最大带宽可达256Gbps https://blog.csdn.net/CocoWu892

ISA总线

PCI总线扩展槽



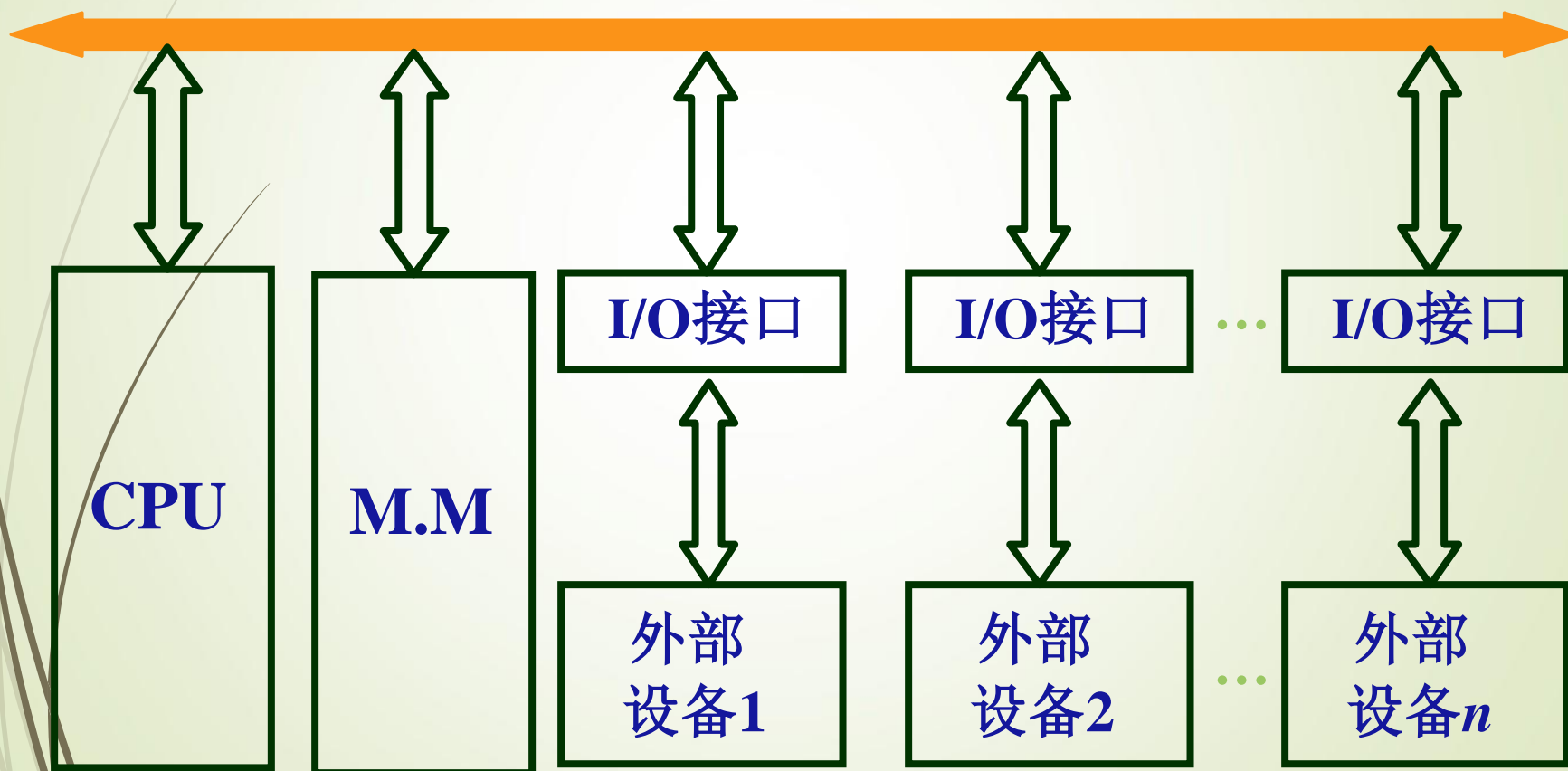
PCI总线扩展槽



3.4 总线结构

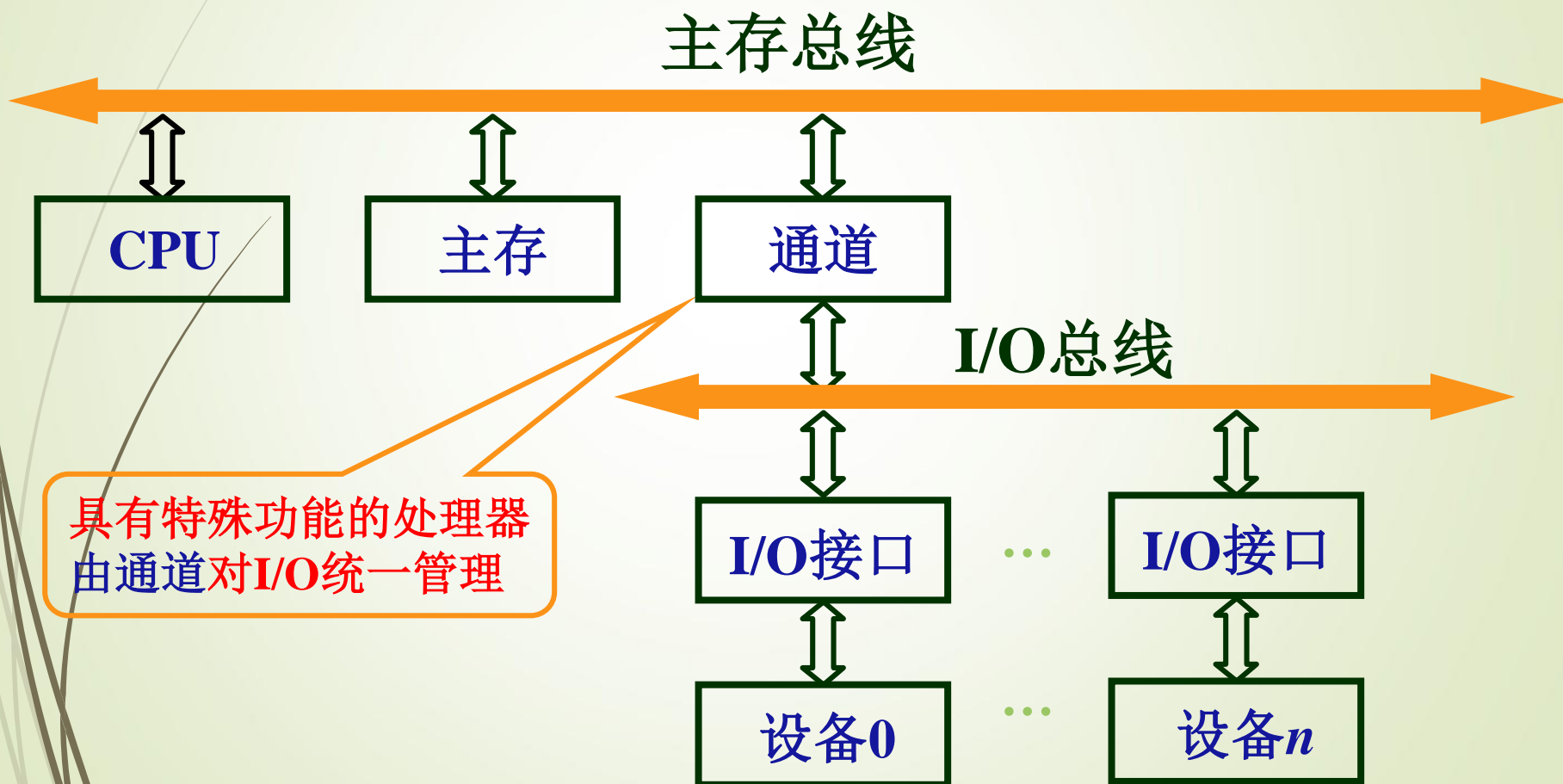
一、单总线结构

单总线（系统总线）

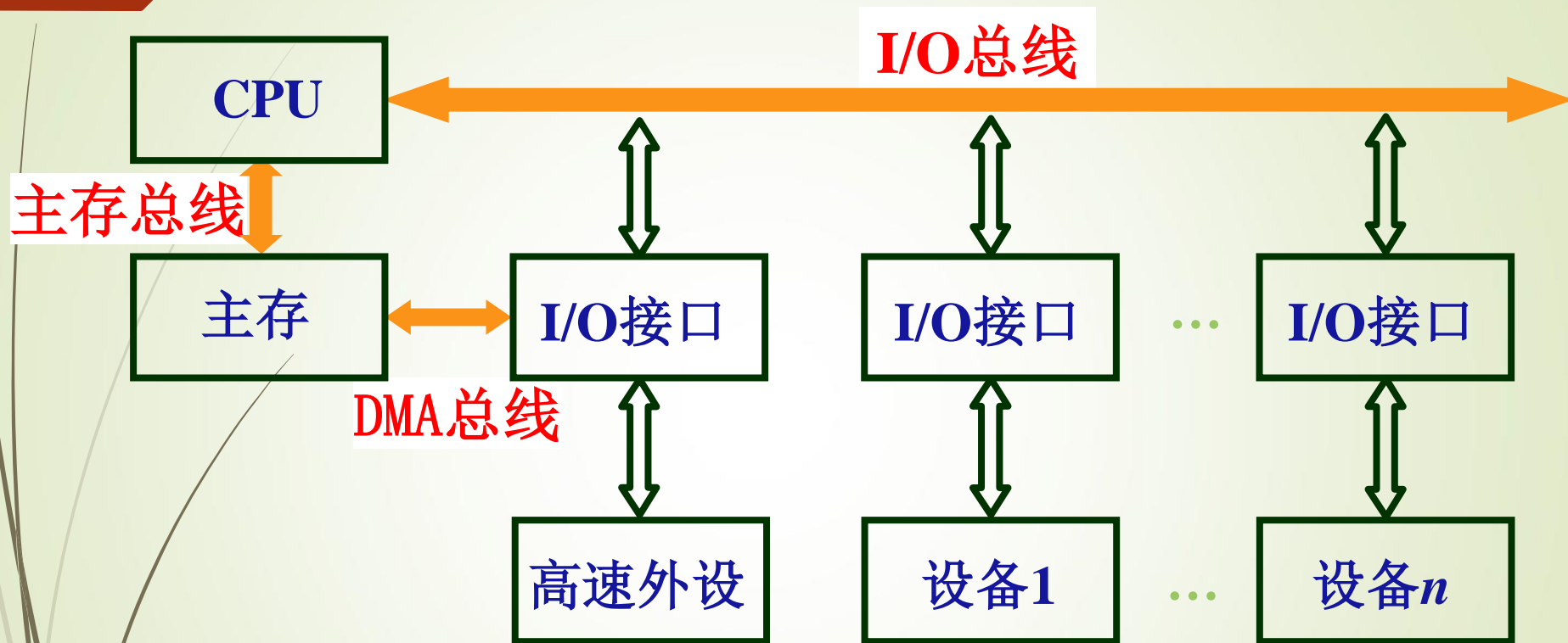


二、多总线结构

1. 双总线结构

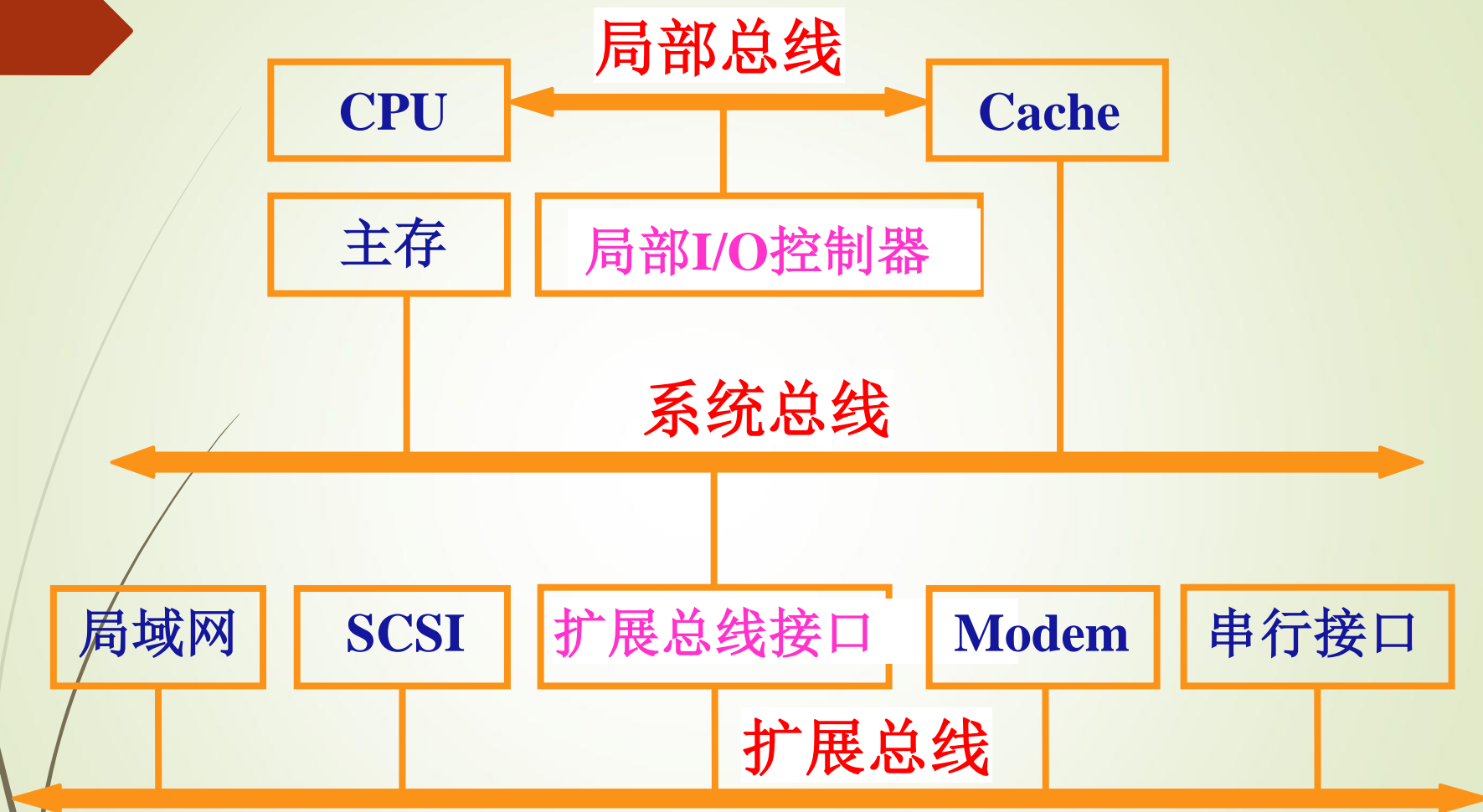


2. 三总线结构

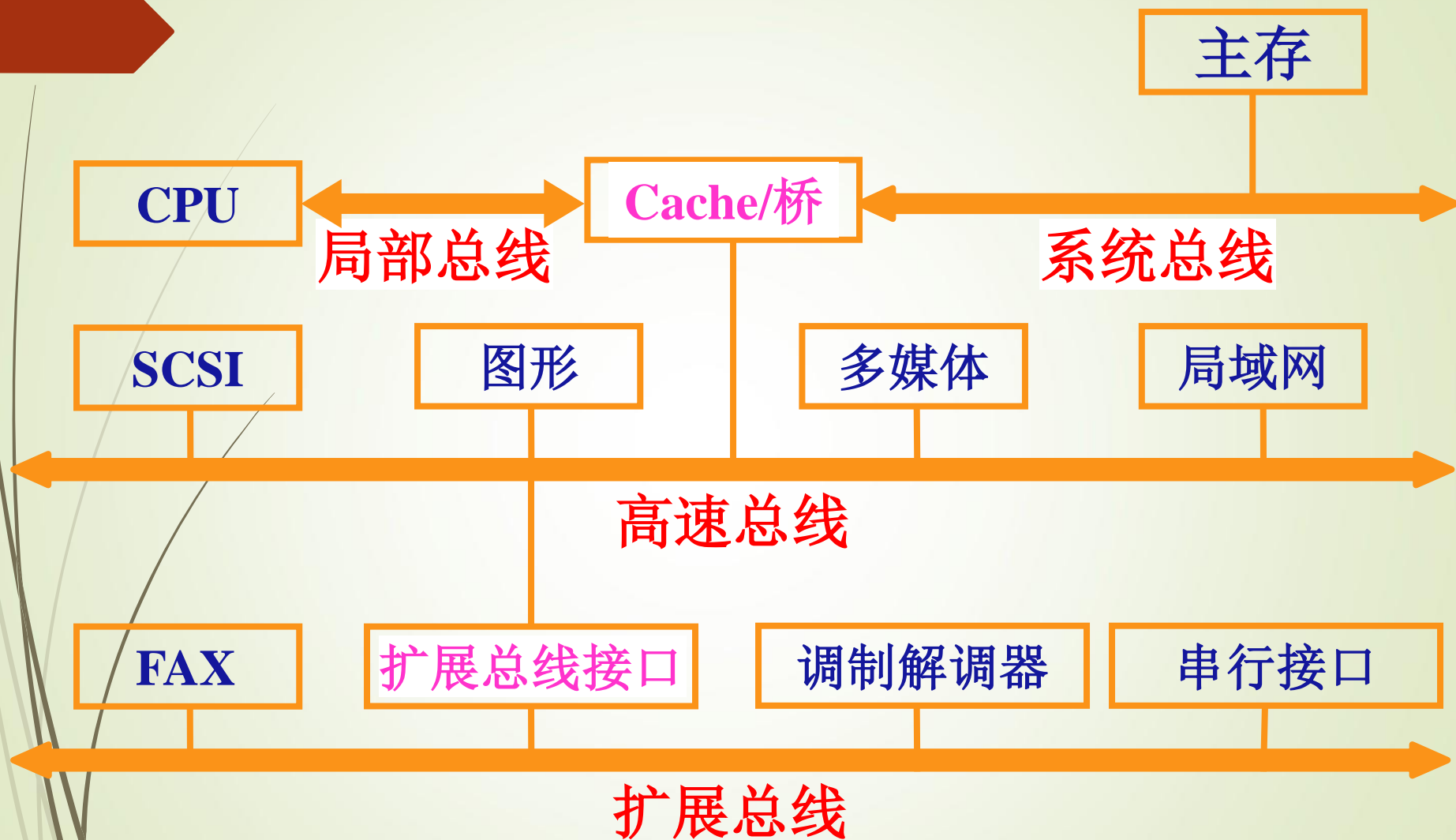


- 任一时刻只能使用一种总线

3. 三总线结构的又一形式

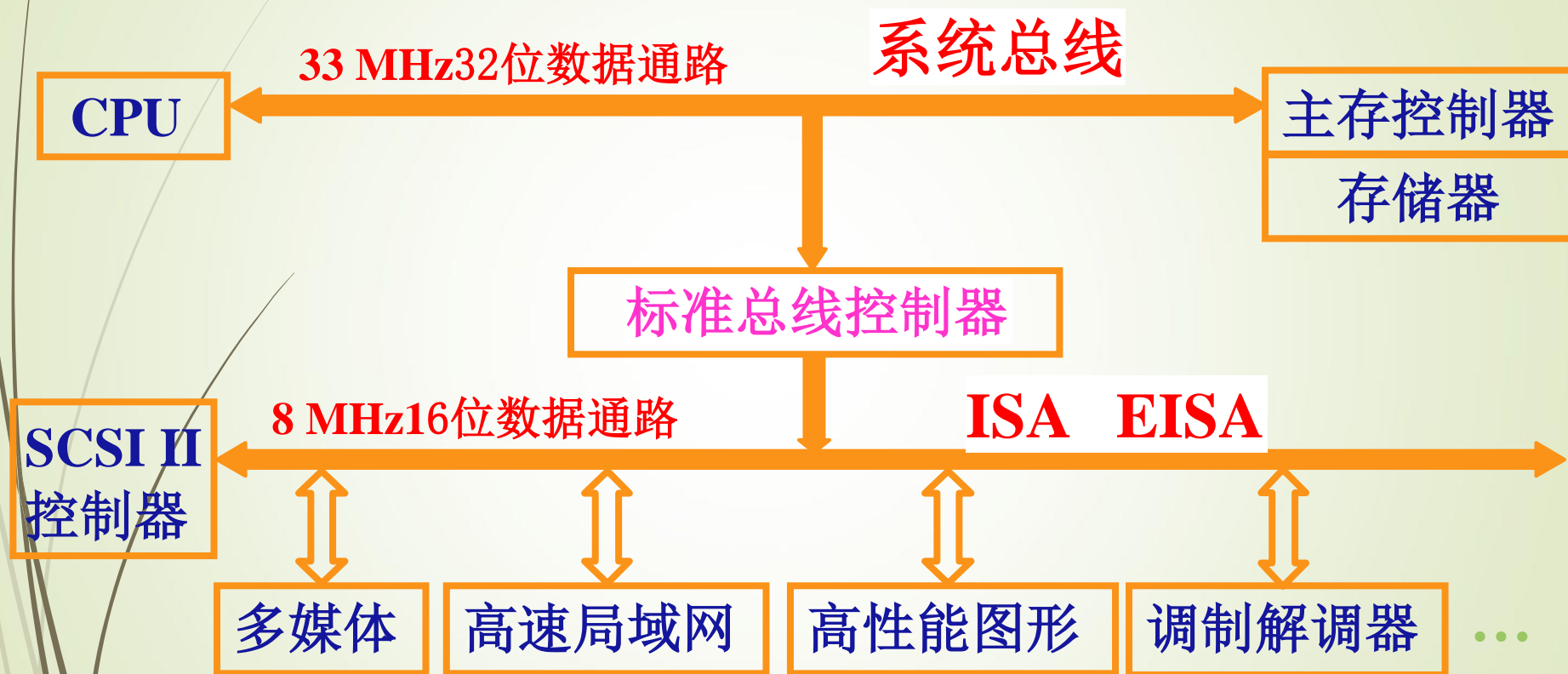


4. 四总线结构

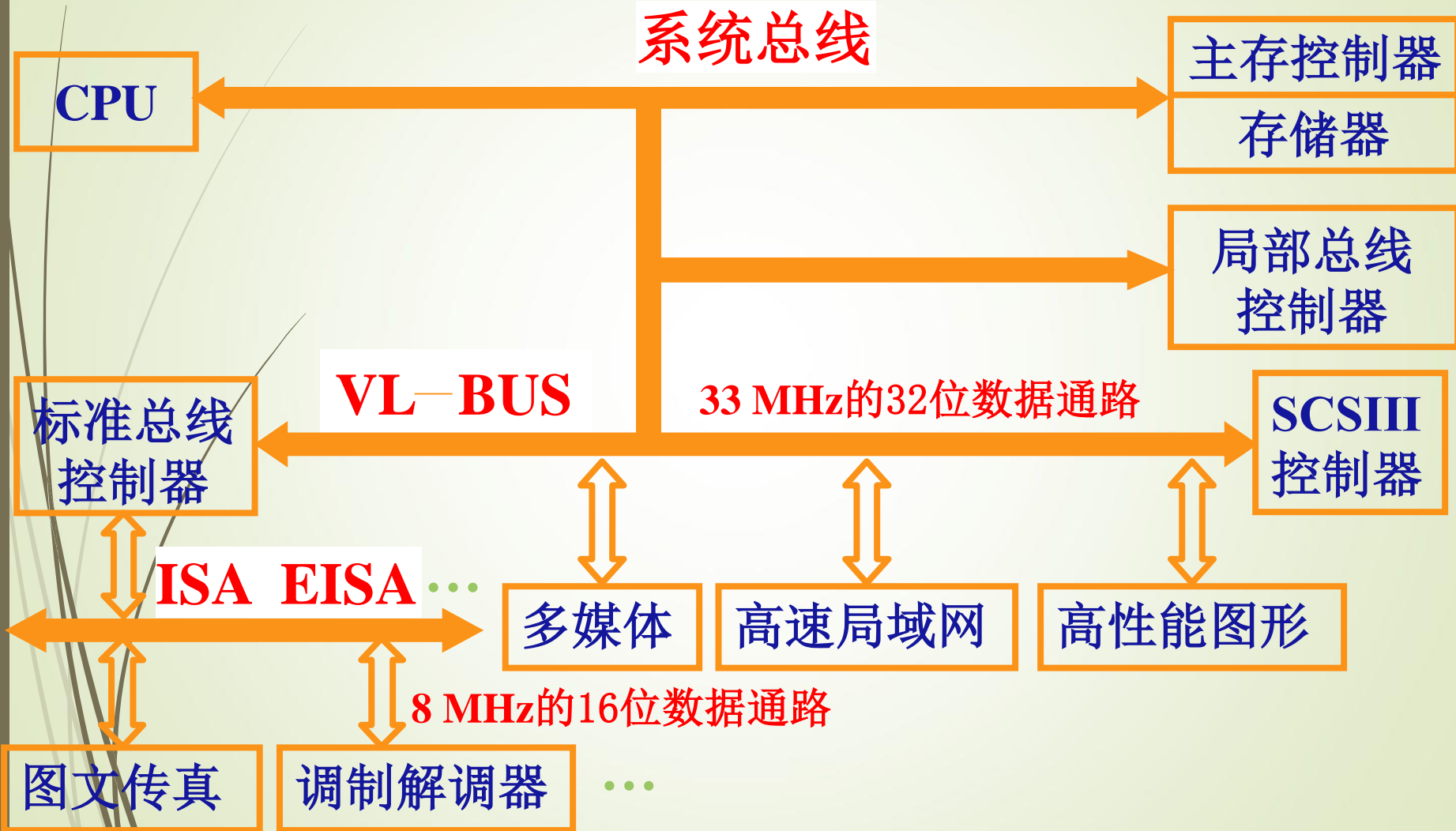


三、总线结构举例

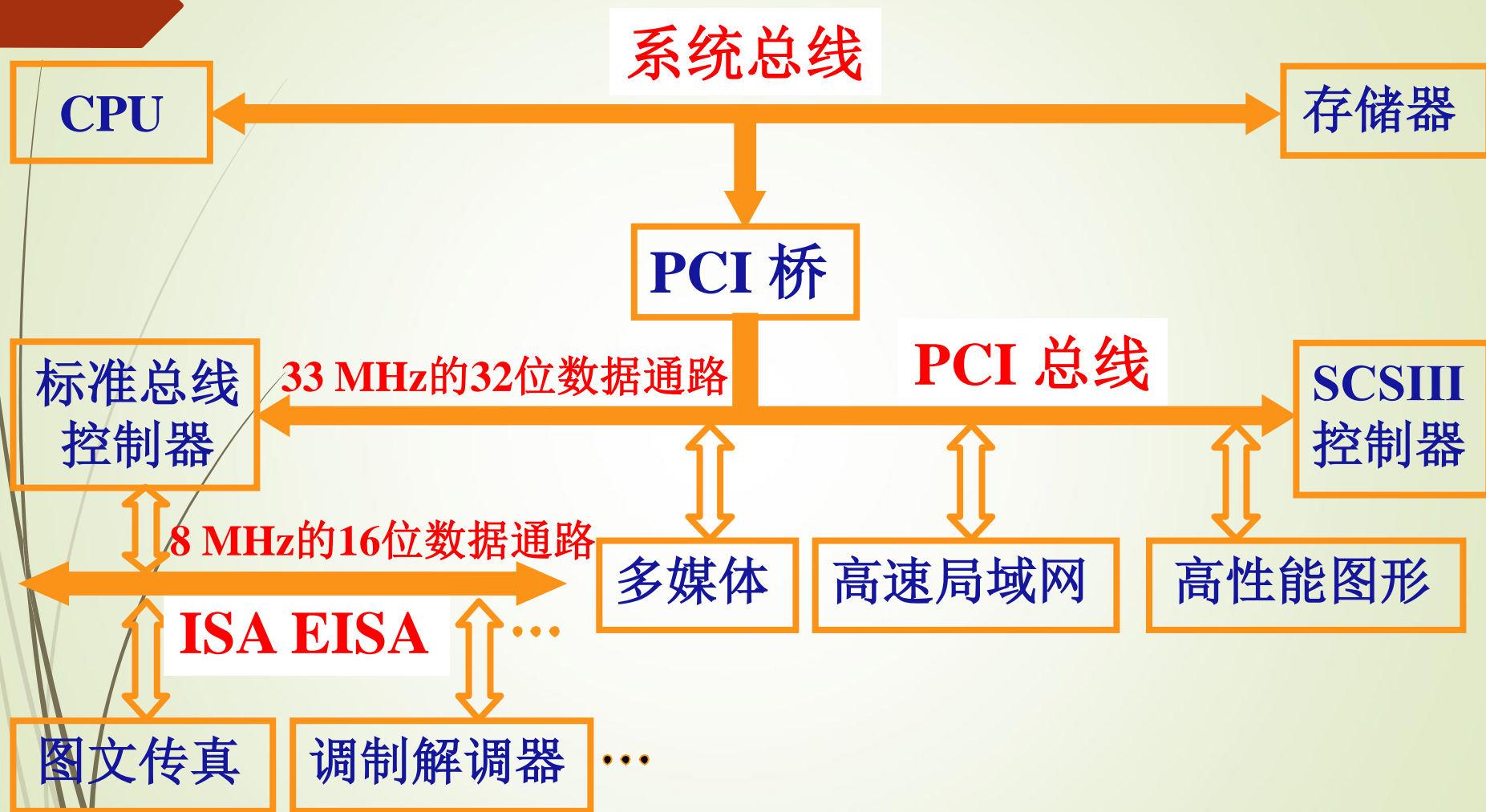
1. 传统微型机总线结构



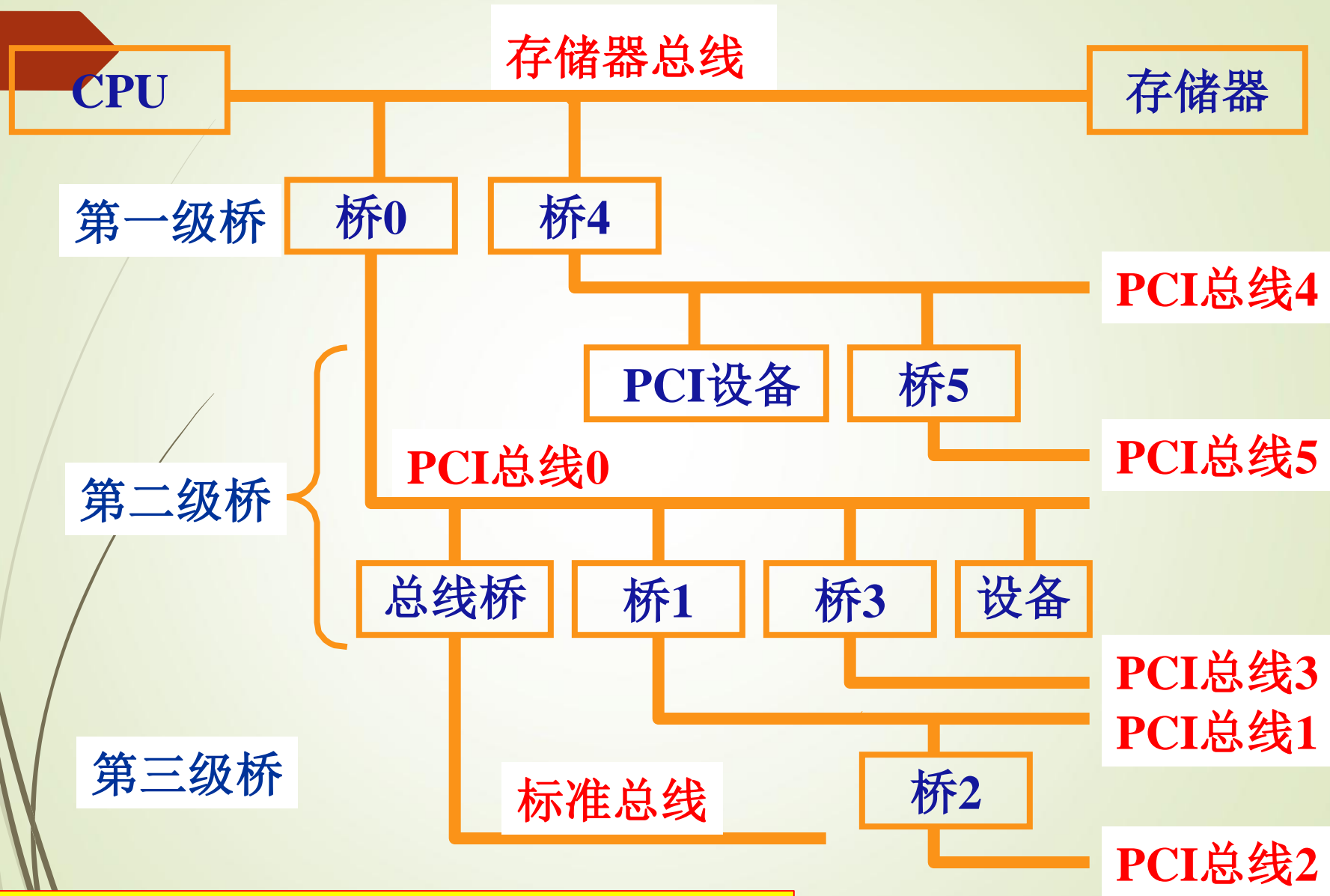
2. VL-BUS局部总线结构



3. PCI 总线结构



4. 多层 PCI 总线结构



3.5 总线控制

一、总线判优控制（仲裁）

1. 基本概念

- **主设备(模块)** 对总线有 控制权
- **从设备(模块)** 响应 从主设备发来的总线命令
- **总线仲裁** 在总线争用时，决定（选出）具有总线控制权的设备或模块的过程

3.5 总线控制

一、总线判优控制（仲裁）

1. 基本概念

• 总线判优控制

集中式

链式查询(菊花链, Daisy chain)

计数器定时查询(Query by a counter)

独立请求方式(Centralized, Parallel)

自举式 (Self-selection)

分布式

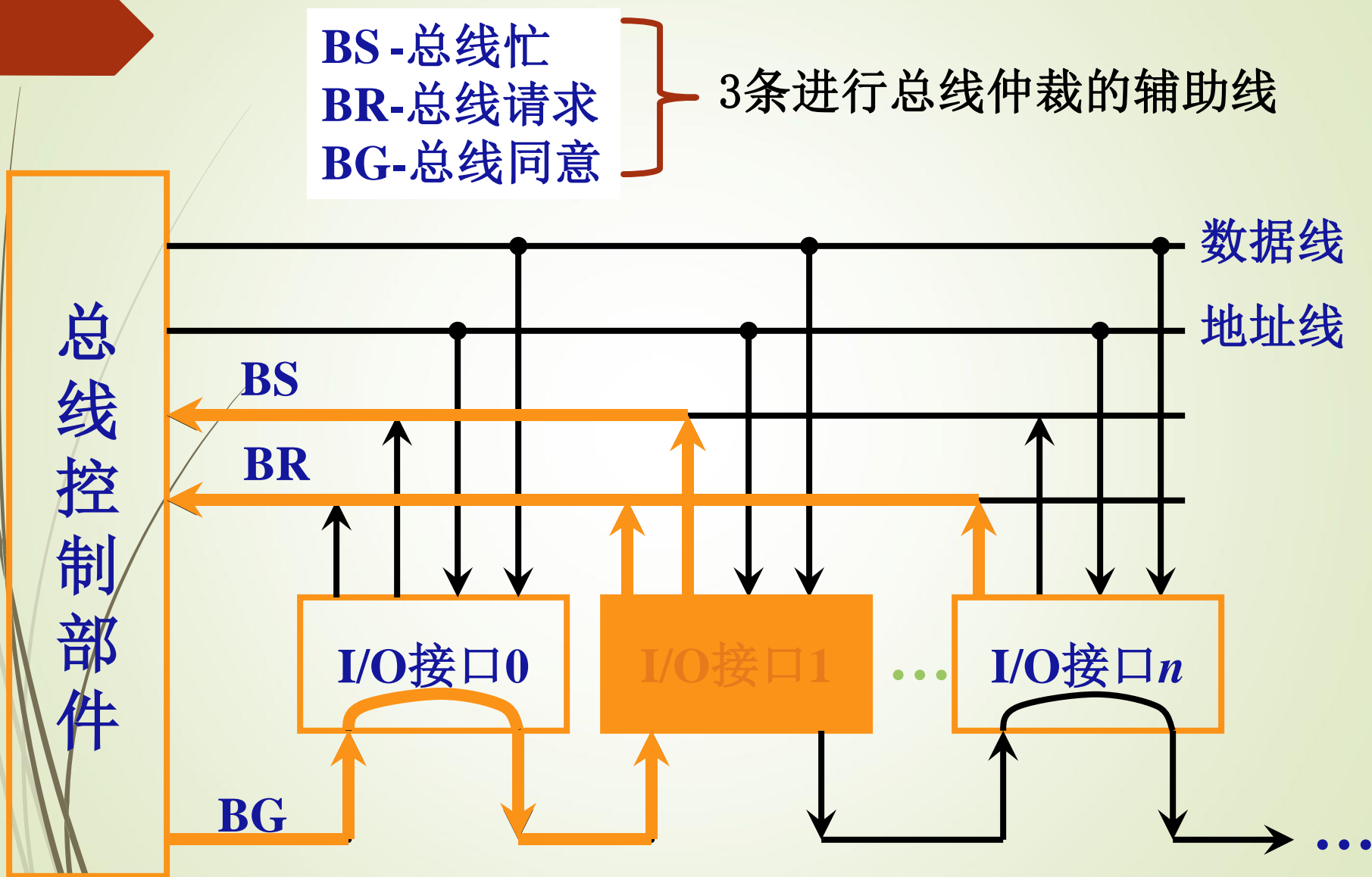
冲突检测 (Collision detection)

- ° 集中式：将控制逻辑做在一个专门的总线控制器或总线裁决器中，通过将所有总线请求集中起来利用一个特定的裁决算法进行裁决
- ° 分布式：没有专门的总线控制器，其控制逻辑分散在各个部件或设备中

裁决方案应在以下两个因素间进行平衡

- 等级性(Priority)—具有高优先级的设备应该先被服务
- 公平性(Fairness)—即使具有最低优先权的设备也不能永远得不到总线使用权

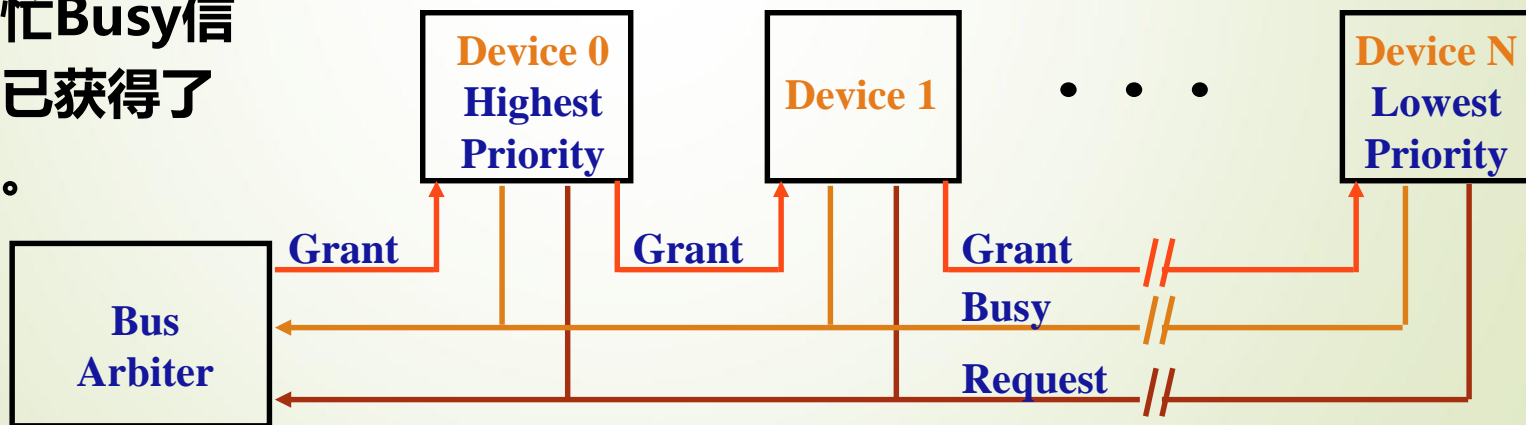
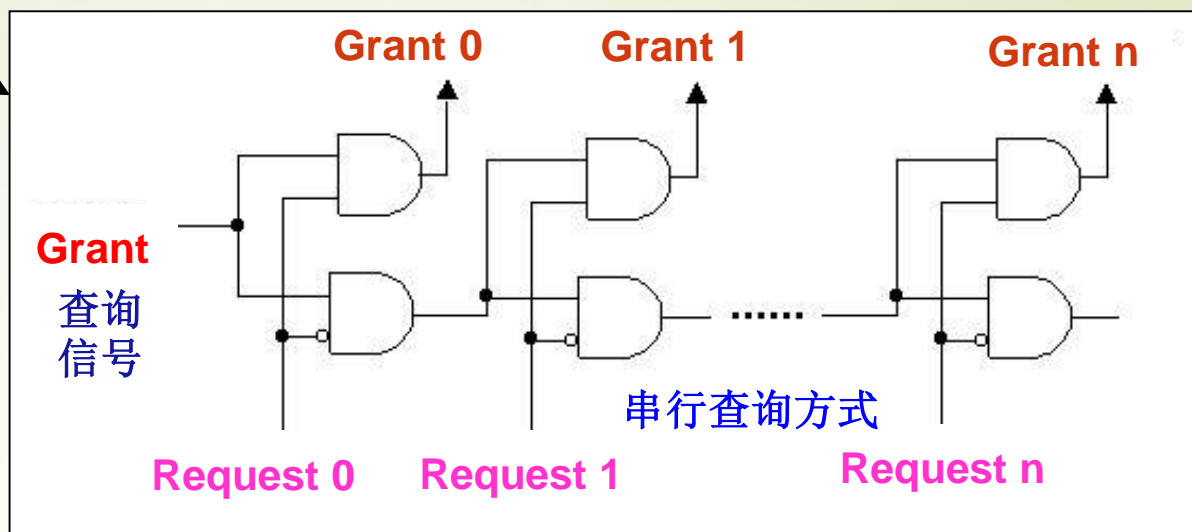
2. 链式查询方式



菊花链总线裁决

菊花链查询 电路

Grant从最高优先权的设备依次向最低优先权的设备串行相连。如果到达的设备有总线请求，则Grant信号就不再往下传，该设备建立总线忙Busy信号，表示它已获得了总线使用权。

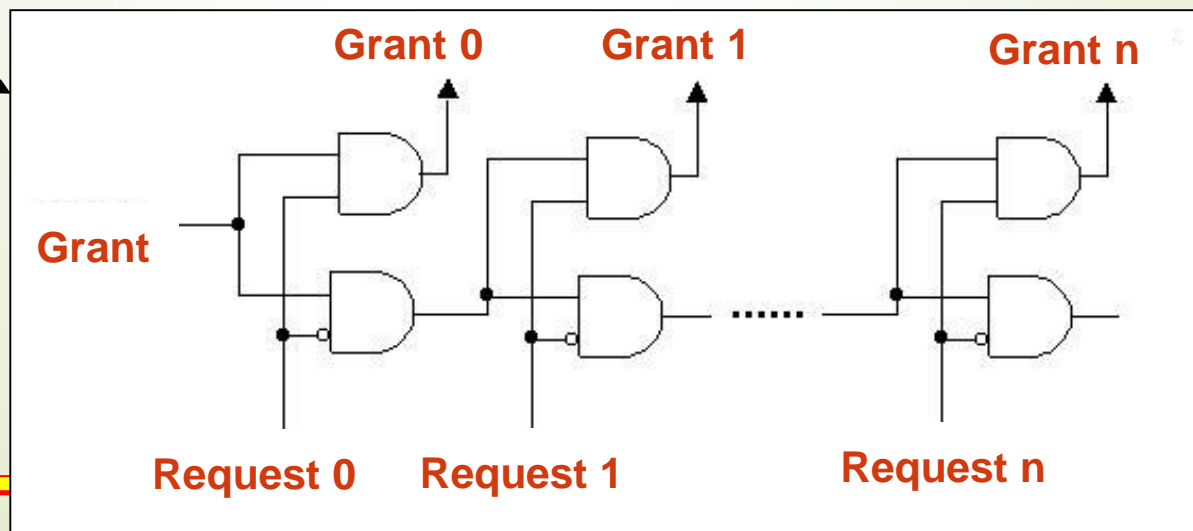


相当于“击鼓传花”

总线同意信号截留电路的设计

grant	request ₀	request ₁	...	request _n	grant ₀	grant ₁	...	grant _n
1	1	X	X	X	1	0	0	0
1	0	1	X	X	0	1	0	0
...	
1	0	0	0	1	0	0	0	1

菊花链查询
电路



2. 链式查询方式

- 优点:

- ➡ 线数少（3根）,设备扩充容易

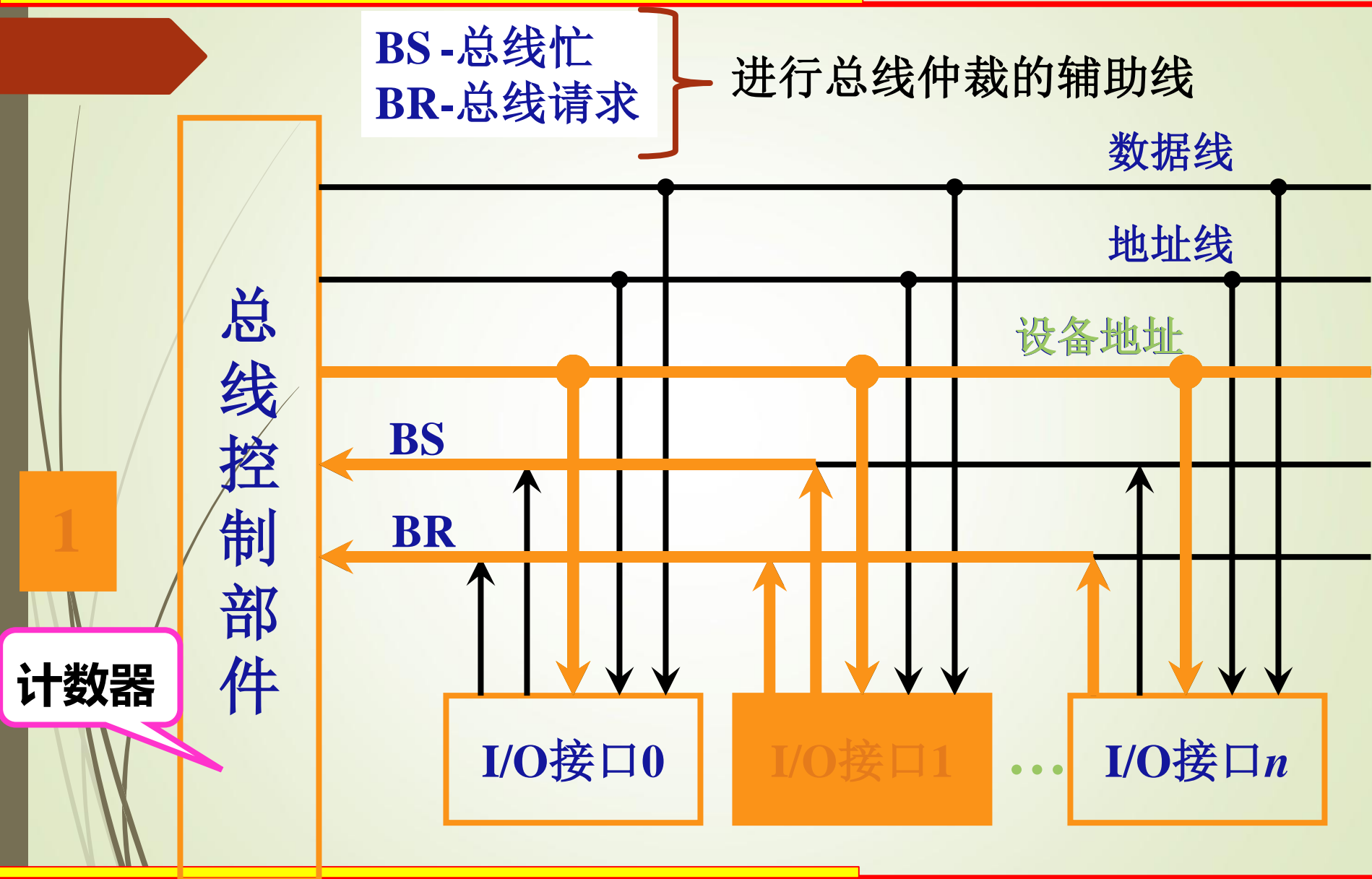
- 缺点:

- ➡ 离总线控制器最近的设备具有最高优先权，优先级低的设备难获得请求。

- ➡ 对电路故障敏感（如果BG线故障）。

- ➡ 菊花链的使用限制了总线速度。

3. 计数器定时查询方式



相当于“点名报到”

3. 计数器定时查询方式

计数器

从“0”开始 与链式优先级相同

从终止点开始 循环优先级，各个部件
优先级相等

计数器初始值可以由程序设定。

地址线数： $\log_2 n$

因为 n 个设备，需要 $\log_2 n$ 根地址线与之对应，
每个设备还有一个唯一的编码（地址）。

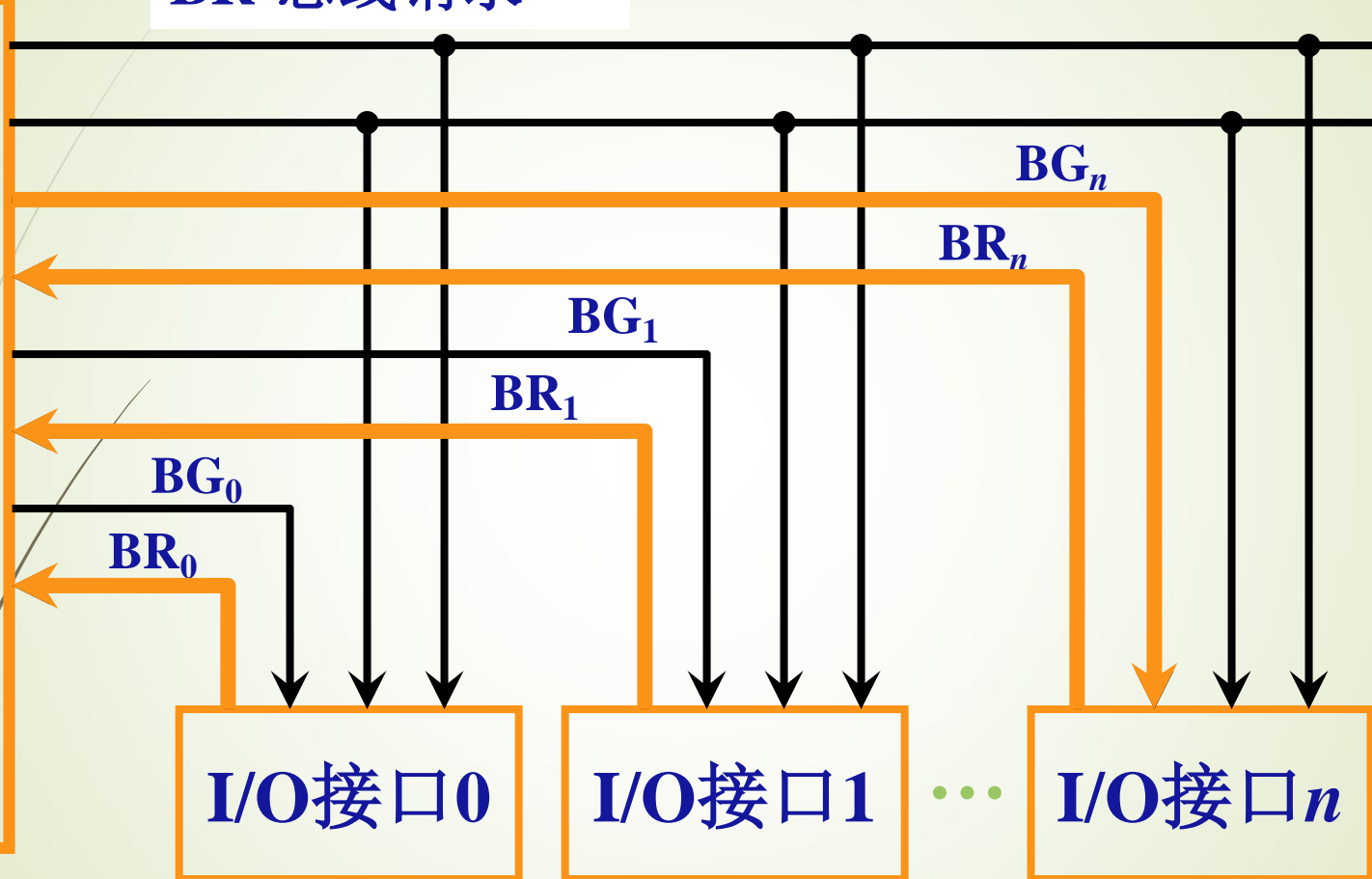
4. 独立请求方式

相当于“领导说了算”

BG-总线同意
BR-总线请求

总线控制部件

数据线
地址线



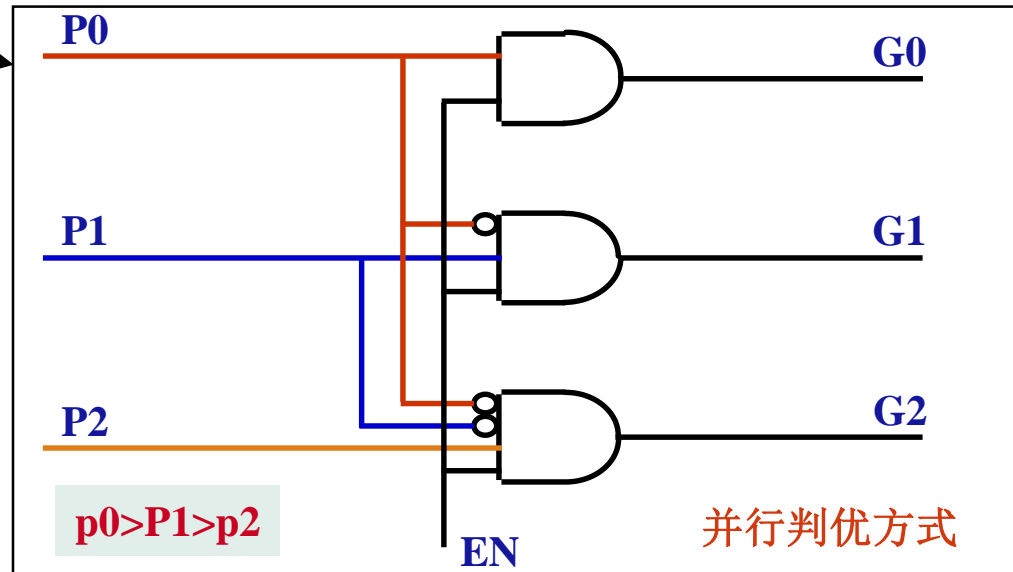
排队器

裁决算法：有排队器。控制器可采用固定的并行判优算法、平等的循环菊花链算法、动态优先级算法（如：最近最少用算法、先来先服务算法）等。

独立请求方式裁决

并行判优电路

p0、P1、p2优先级怎样？

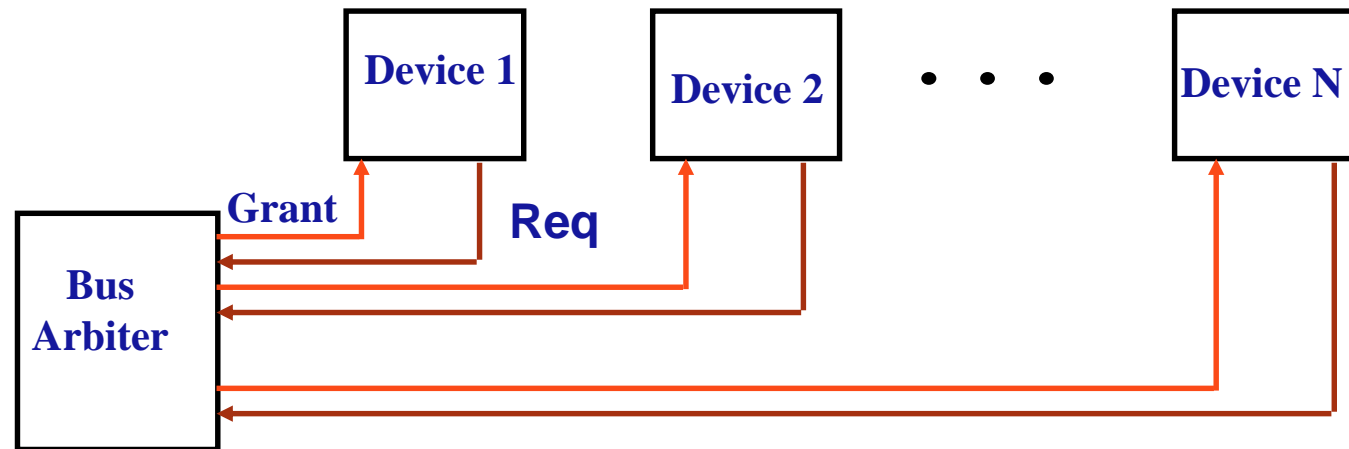


- 各设备都有一对总线请求线**Req**和总线允许线

Grant。

- 当某设备要使用总线时，就通过对应的总线请求线将请求信号送到总线控制器。

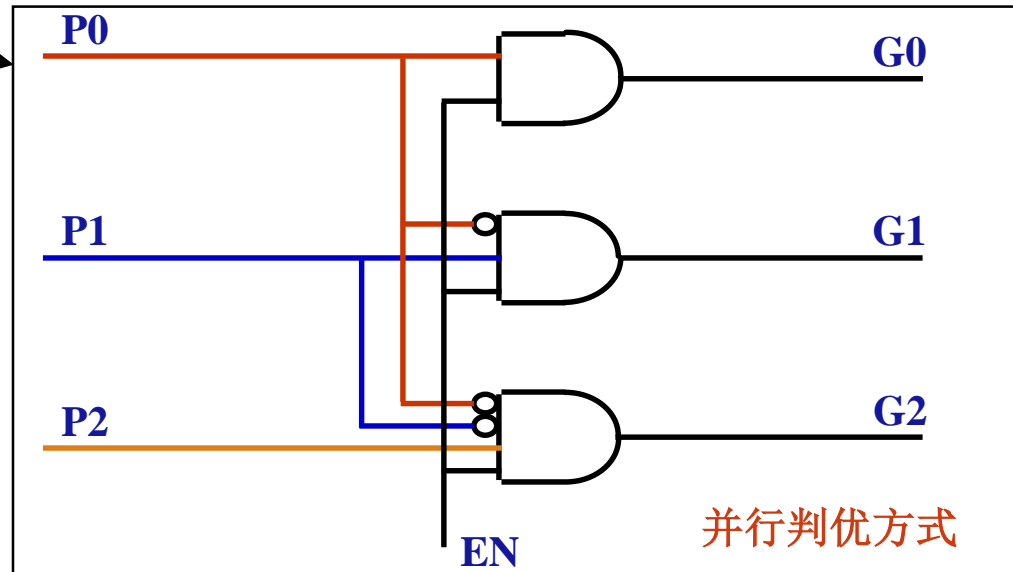
- 总线控制器中有一个判优电路，可根据各设备的优先级确定选择哪个设备。控制器可给各请求线以固定的优先级，也可编程设置



独立请求方式裁决

并行判优电路

p0、P1、p2优先级怎样？



EN	P0	P1	P2	G0	G1	G2
1	1	X	X	1	0	0
1	0	1	X	0	1	0
1	0	0	1	0	0	1

$p0 > P1 > p2$

4. 独立请求方式

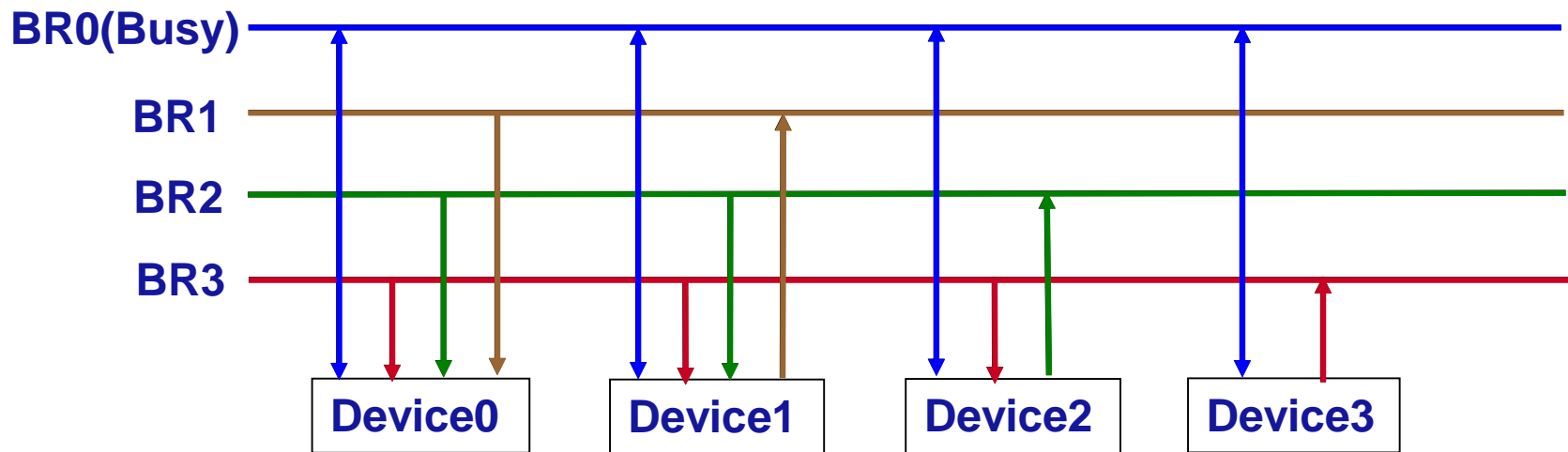
- 优点:

- ➡ 响应速度快
- ➡ 优先级控制灵活

- 缺点:

- ➡ 线数多（ $2n$ 根辅助线），控制复杂

5、自举分布式裁决



- 优先级固定，各设备独立决定是否是自己是最高优先级请求者
- 需请求总线的设备在各自对应的总线请求线上送出请求信号
- 在总线裁决期间每个设备将比自己优先级高的请求线上的信号取回分析：
 - 若有总线请求信号，则本设备不能立即使用总线
 - 若没有，则可以立即使用总线，并通过总线忙信号阻止其他设备使用总线
 - 最低优先级设备可以不需要总线请求线，为什么？
 - 需要较多连线用于请求信号，所以，许多总线用数据线DB作为总线请求线
N个设备要多少请求信号？ N条！
 - NuBus (MacintoshII 中的底板式总线)、SCSI总线等采用该方案

上图中的优先级 (优先级)是什么？

设备3>设备2>设备1>设备0

6、冲突检测方式裁决

基本思想：

- 当某个设备要使用总线时，它首先检查一下是否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，它会监听总线以检测是否发生了冲突；
- 当冲突发生时，两个设备都会停止传输，延迟一个随机时间后再重新使用总线
 - 该方案一般用在网络通信总线上，如：Ethernet总线等。

二、总线通信控制

1. 目的 解决通信双方 协调配合 问题

2. 总线周期

申请分配阶段 主模块申请，总线仲裁决定

寻址阶段 主模块向从模块 给出地址 和 命令

传数阶段 主模块和从模块 交换数据

结束阶段 主模块 撤销有关信息

3. 总线通信的四种方式(总线定时方式)

同步通信
(synchronous)

由 统一时标 控制数据传送

其控制线上有一个时钟和一个固定的
与该时钟相关的通信协议

异步通信
(asynchronous)

采用 应答方式，没有公共时钟标准
握手协议 (handshaking protocol)

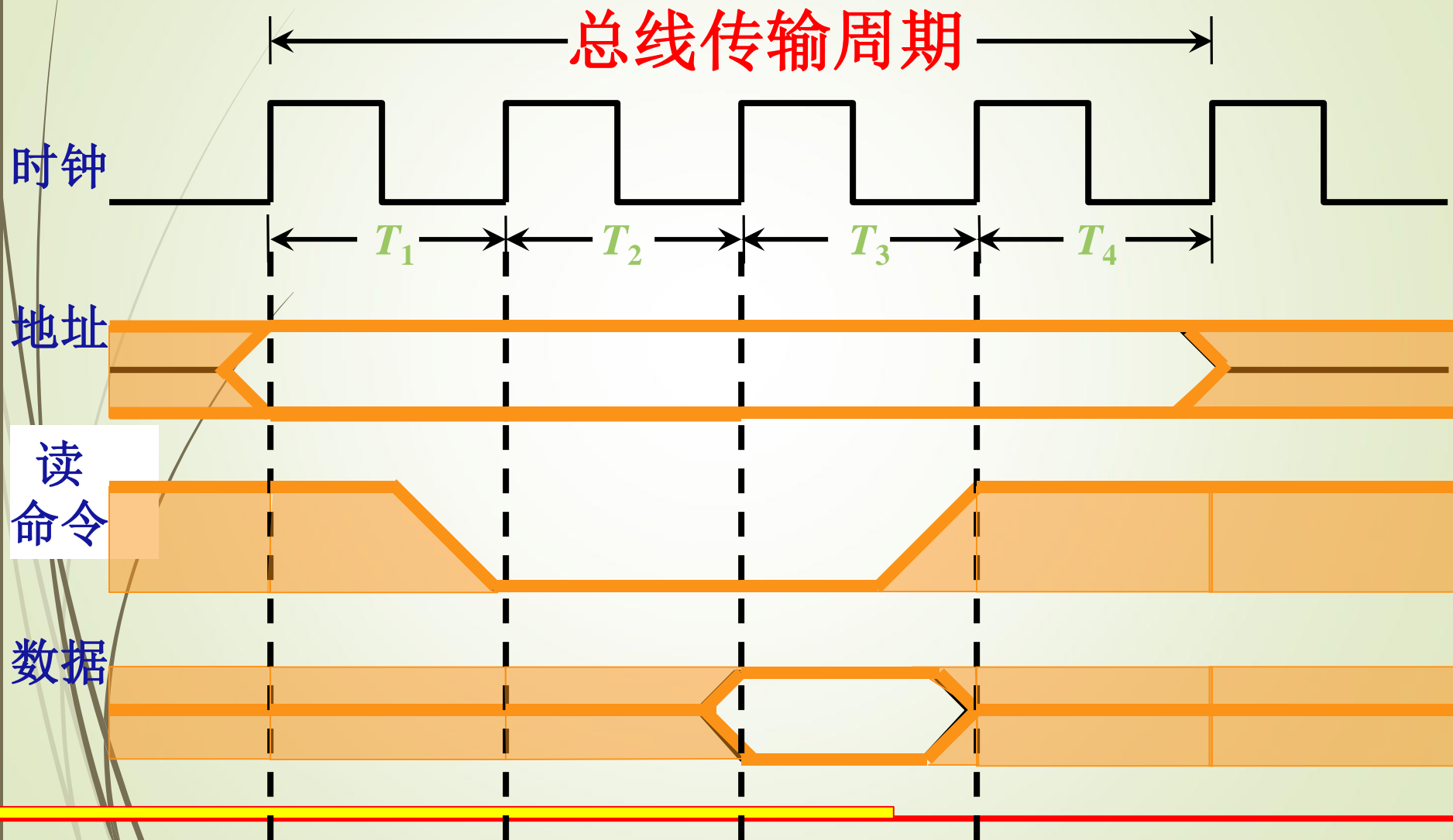
半同步通信

同步、异步结合

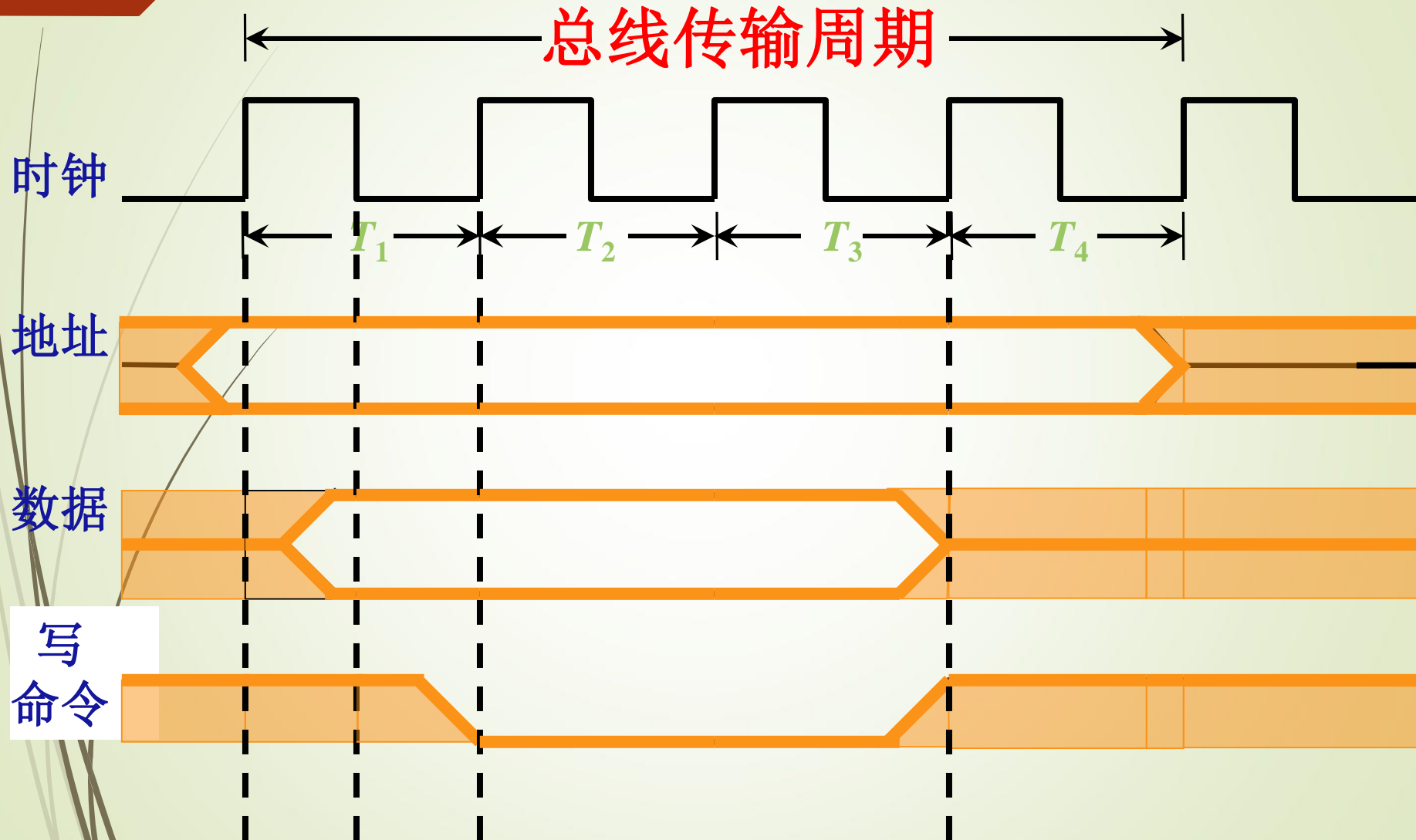
分离式通信
(拆分总线事务,
Split Bus Transaction)

充分 挖掘 系统 总线每瞬间 的 潜力

(1) 同步式数据输入



(2) 同步式数据输出



同步通信

➡ 优点

- ➡ 规定明确、统一，模块间的配合简单一致。
- ➡ 控制逻辑少而速度快

➡ 缺点

- ➡ 主从模块时间配合属于强制性“同步”，必须在限定时间内完成规定要求。
- ➡ 对所有从模块用同一时限，就必须按照最慢的部件来设计公共时钟，影响了总线的工作效率，也给设计带来了局限性，缺乏灵活性。

(3) 异步通信

- ➡ 非时钟定时，**没有一个公共的时钟标准**。因此，能够连接带宽范围很大的各种设备。总线能够加长而不用担心时钟偏移（**clock skew**）问题。
- ➡ 采用**应答方式**，允许各模块速度不一致。
- ➡ 主模块发出**请求信号**（**Request**）时，一直等待从模块反馈回来的**应答信号**（**Acknowledge**），才开始通信。这需要主从模块之间增加两条**应答线**（即握手交互信号线 **Handshaking**）。
- ➡ 分为不互锁、半互锁和全互锁

(3) 异步通信

请求
回答

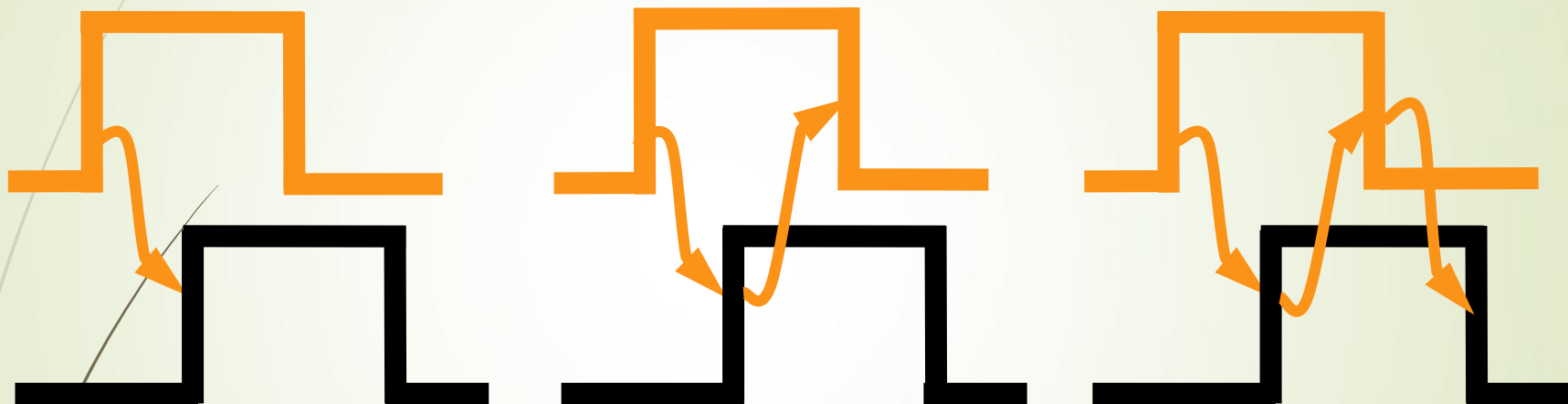
主设备

从设备

不互锁

半互锁

全互锁



(3) 异步通信

■ 优点:

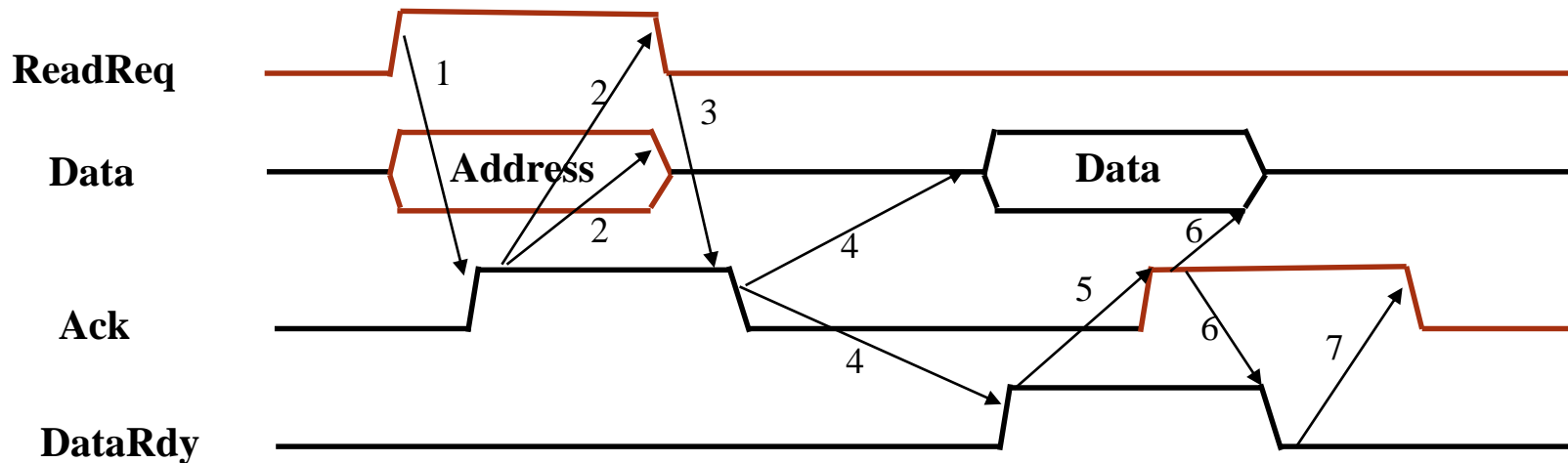
- 灵活, 可挂接各种具有不同工作速度的设备

■ 缺点:

- ① 对噪声较敏感 (任何时候都可能接收到对方的应答信号)
- ② 接口逻辑较复杂

Handshaking Protocol(握手协议)

一个总线事务：地址阶段 + 数据阶段 + ... + 数据阶段



❖ Three control lines

- ReadReq: 请求读内存单元
(地址信息同时送到地址/数据线上)
- DataRdy: 表示已准备好数据
(数据同时送到地址/数据线上)
- Ack: ReadReq or DataRdy的回答信号

❖ 上述为read过程, 但write操作基本类似

ReadReq和Ack之间的握手过程
完成地址信息的传输

DataRdy和Ack之间的握手过程
完成数据信息的传输

一共有多少次握手? 7次

是全互锁方式!

(3) 异步通信

- 异步串行通信用波特率来衡量数据传输率

例1.在异步串行传输系统中，假设每秒可传输20个数据帧，一个数据帧包含1个起始位、7个数据位、1个奇校验位、1个结束位。计算其波特率和比特率。

解：波特率 $= (1 + 7 + 1 + 1) \times 20 = 200\text{bps}$

比特率 $= 20 \times 7 = 140\text{bps}$

在异步串行传输系统中，假设每秒可传输20个数据帧，一个数据帧包含1个起始位、8个数据位、1个奇校验位、1个结束位。计算其波特率和比特率分别为多少？

- ☐ A 11bps, 220bps
- ☐ B 220bps, 11bps
- ☐ C 140bps, 220bps
- ☒ D 220bps, 160bps

提交

在异步串行传输系统中，若每个数据帧包含1个起始位、8个数据位、1个奇校验位、1个结束位。波特率为1200bps，求比特率为多少？

- ☒ A 872.72bps
- ☐ B 1000bps
- ☐ C 13200bps
- ☐ D 9600bps

提交

(4) 半同步通信 (同步、异步 结合)

同步 发送方 用系统 时钟前沿 发信号

接收方 用系统 时钟后沿 判断、识别

异步 允许不同速度的模块和谐工作

增加一条 “等待” 响应信号 $\overline{\text{WAIT}}$

以输入数据为例的半同步通信时序

T_1 主模块发地址

T_2 主模块发命令

T_w 当 $\overline{\text{WAIT}}$ 为低电平时，等待一个 T

T_w 当 $\overline{\text{WAIT}}$ 为低电平时，等待一个 T

⋮

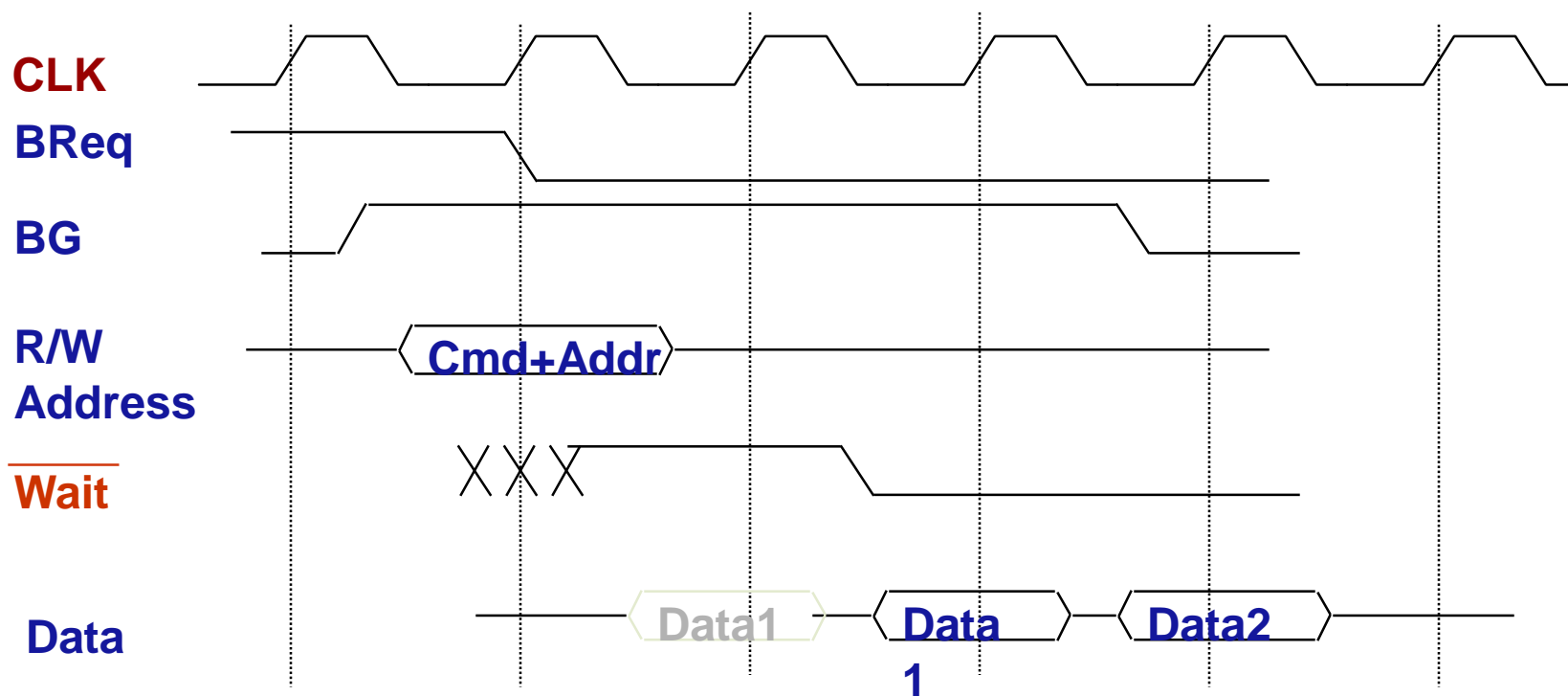
T_3 从模块提供数据

T_4 从模块撤销数据，主模块撤销命令

半同步总线

为解决异步方式对噪声敏感的问题，在异步总线中引入时钟信号

就绪和应答等握手信号 (如: **Wait**信号、**TRDY**和**IRDY**信号等) 都在时钟的上升沿有效
信号的有效时间限制在时钟到达的时刻，而不受其他时间的信号干扰



- ❖ 通过 “**Wait**”信号从设备告知主设备何时数据有效
- ❖ 结合了同步和异步的优点。既保持了 “所有信号都由时钟定时” 的特点，又允许 “不同速度设备共存于总线”

上述三种通信的共同点

一个总线传输周期（以输入数据为例）

- 主设备发地址、命令 占用总线
- 从设备准备数据 不占用总线 总线空闲
- 从设备向主设备发数据 占用总线

(5) 分离式通信

充分挖掘系统总线每瞬间的潜力

一个总线传输周期

子周期1 主设备申请 占用总线，使用完后
即 放弃总线 的使用权

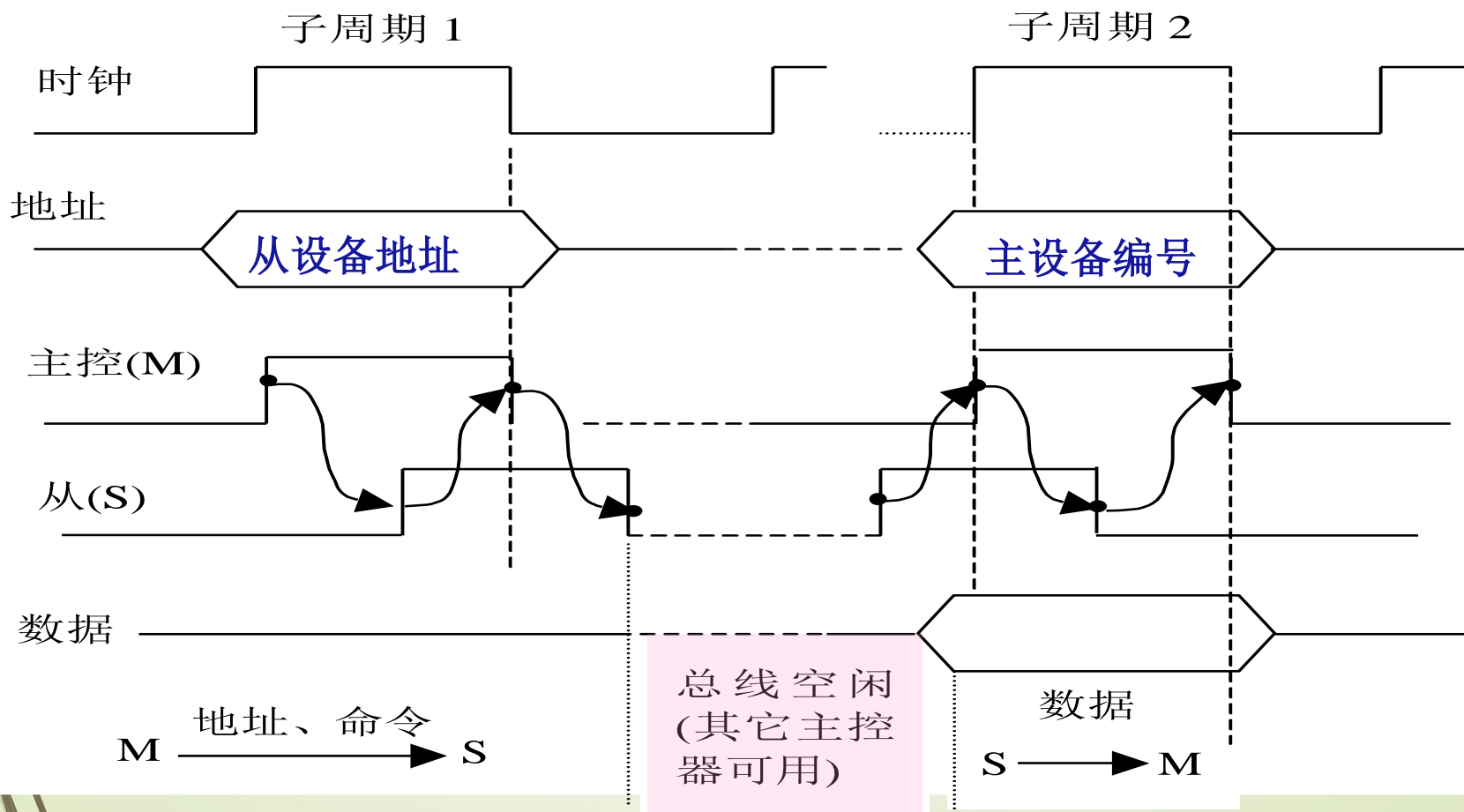
子周期2 从设备申请 占用总线，将各种信
息送至总线上

主设备

Split Bus Transaction(拆分总线事务)

将一个事务分成两个子过程:

- 过程1: 主控设备A获得总线使用权后, 将请求的事务类型、地址及其他信息(如A的标识等)发到总线, 从设备B记下这些信息。A发完信息后立即释放总线, 其他设备便可使用总线
- 过程2: B收到A发来的信息后, 按照A的要求准备数据, 准备好后, B便请求使用总线, 获使用权后, B将A的编号及所需数据送到总线, A便可接收



分离式通信特点

1. 各设备有权申请占用总线
2. 采用同步方式通信，不等对方回答
3. 各设备准备数据时，不占用总线
4. 总线被占用时，无空闲

充分发挥了总线的有效占用



Thank You !

附录例子

例1：同步和异步总线的最大带宽比较

举例：假定同步总线的时钟周期为50ns，每次总线传输花1个时钟周期，异步总线每次握手需要40ns，两种总线的数据都是32位宽，存储器的取数时间为200ns。要求求出从该存储器中读出一个字时两种**总线的带宽**。

分析如下：

同步总线的步骤和时间为：

(1) 发送地址和读命令到存储器：50ns

(2) 存储器读数据：200ns

(3) 传送数据到设备：50ns

如果存储器读为**230ns**，则结果为多少？

总时间为**350ns**， $4B/350ns=11.4MB/s$

所以总时间为300ns，故最大总线带宽为 $4B/300ns$ ，即：13.3MB/s。

异步总线的步骤和时间为：

第1步为：40ns；

第2、3、4步为： $\text{Max}(3 \times 40ns, 200ns) = 200ns$ ；

(第2、3、4步都和存储器访问时间重叠)

第5、6、7步为： $3 \times 40ns = 120ns$ 。

总时间为360ns，故最大带宽为 $4B/360ns = 11.1MB/s$

由此可知：同步总线仅比异步快大约**20%**。要获得这样的速度，异步总线上的设备和存储器必须足够快，以使每次在**40 ns**内能完成一个子过程

例2：数据块大小对带宽的影响

假定有一个系统具有下列特性：

- (1) 系统支持4~16个32位字的块访问。
- (2) 64位同步总线，时钟频率为200MHz，每个64位数据传输需一个时钟周期，地址发送到存储器需1个时钟周期。
- (3) 在每次总线操作(事务)间有两个空闲时钟周期。
- (4) 存储器访问时间对于开始的4个字是200ns，随后每4个字是20ns。

假定读出数据在总线上传送的同时，随后4个字的存储器读操作也在重叠进行

一个总线事务由一个地址传送后跟一个数据块传送组成，请求出分别用4-字块和16-字块方式读取256个字时的持续**带宽**和**等待时间**。并且求出两种情况下每秒钟内的**有效总线事务数**。

举例-数据块大小对带宽的影响

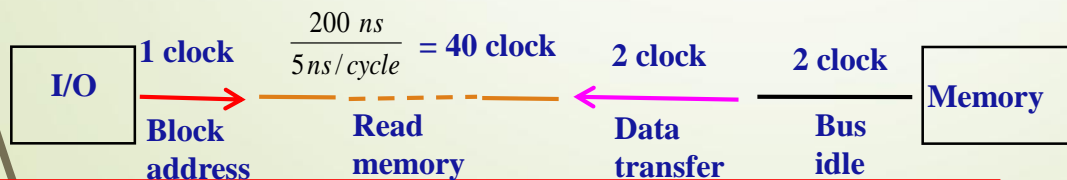
分析 4-字块传送情况：

对于4-字块传送方式，**一次总线事务由一个地址传送后跟一个4-字块的数据传送组成**。也即每个总线事务传送一个4个字的数据块。

每个数据块所花时间为：

- (1) 发送一个地址到主存花1个时钟周期
- (2) 从主存读4个字花： $200\text{ns}/(5\text{ns}/\text{Cycle})=40$ 个时钟周期
(一个周期是 $10^9\text{ns}/200\text{MHz}=1000/200=5\text{ns}$)
- (3) 4个字（128位）的传输需2个时钟周期
(一个64位数据传输需1个时钟周期)
- (4) 在这次传送和下次之间有2个空闲时钟周期

所以一次总线事务总共需45个周期，256个字需 $256/4=64$ 个事务，所以整个传送需 $45 \times 64 = 2880$ 个时钟周期，因而总等待时间为： $2880 \text{周期} \times 5\text{ns}/\text{周期} = 14400\text{ns}$ 。每秒钟的总线事务数为： $64 \times (1\text{s}/14400\text{ns}) = 4.44\text{M}$ 个。
总线带宽为： $(256 \times 4\text{B})/14400\text{ns} = 71.11\text{MB/s}$ 。



Latency = 2880 clock cycles
Bandwidth = 71.11MB /sec

举例-数据块大小对带宽的影响

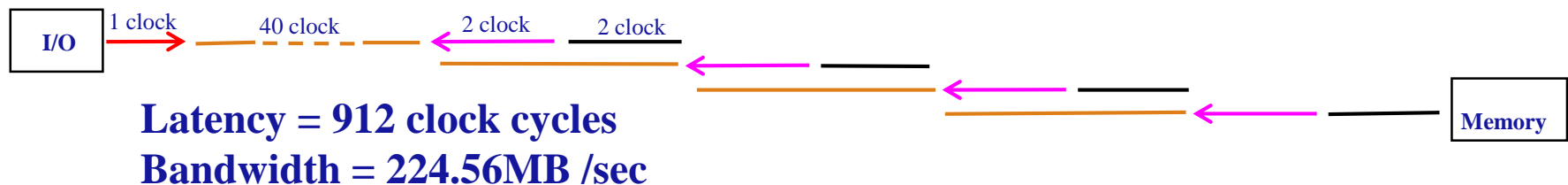
分析 16-字块传送情况：

对于16-字块传送，一次总线事务由一个地址传送后跟一个16-字块的数据传送组成。
也即每个总线事务传送一个16个字的数据块。

第一个4-字所花时间为：

- (1) 发送一个地址到主存花1个时钟周期
- (2) 从主存读开始的4字花： $200\text{ns}/(5\text{ns}/\text{Cycle})=40$ 个时钟周期
- (3) 4个字需2个时钟周期，在传输期间存储器开始读取下一个4字
- (4) 在本次和下次之间有2个空闲时钟，此期间下一个4字已读完

所以，16字中其余三个4字只要重复上述最后两步。因此对于16-字块传送，一次总线事务共需花费的周期数为： $1+40+4 \times (2+2) = 57$ 个周期，256个字需 $256 / 16=16$ 个事务，因此整个传送需 $57 \times 16 = 912$ 个时钟周期。故总等待时间为： $912\text{周期} \times 5\text{ns} / \text{周期}=4560\text{ns}$ 。几乎仅是前者的1/3。每秒钟的总线事务个数为： $16 \times (1\text{s} / 4560\text{ns}) = 3.51\text{M}$ 个。总线带宽为： $(256 \times 4\text{B}) \times (1\text{s}/4560\text{ns}) = 224.56\text{MB/s}$ ，比前者高3.6倍。由此可见，大数据块传输的优势非常明显。



增加同步总线带宽的措施

- 提高时钟频率
- Data bus width(增加数据线宽度)
 - 能同时传送更多位
 - Example: SPARCstation 20's memory bus 有 128 bit
 - Cost: more bus lines
- Block transfers(允许大数据块传送)
 - 背对背总线周期, 也称为突发(Burst)传输方式
 - 只要开始送一次地址, 后面连续送数据
 - Cost: (a)增加复杂性
(b)延长响应时间
- Split Bus Transaction(拆分总线事务)
 - 一次总线事务时间延长, 但整个系统带宽增加
 - Cost: (a) 增加复杂性
(b) 延长响应时间
- 不采用分时复用方式
 - 地址和数据可以同时送出
 - Cost(代价): (a) more bus lines, (b) 增加复杂性

关于I/O总线标准

I/O总线是各类I/O控制器与CPU、内存之间传输数据的一组公用信号线，这些信号线在物理上与主板扩展槽中插入的扩展卡（I/O控制器）直接连接。

➡ I/O总线是标准总线，I/O总线标准有：

- ➡ ISA / EISA总线：（已逐步被淘汰）
- ➡ Multibus总线：（已逐步被淘汰）
- ➡ PCI总线：目前PC机所用的主流标准
- ➡ PCI-Express(高速PCI总线)：目前PC机所用的主流标准

➡ I/O总线的带宽

- ➡ 总线的数据传输速率(MB/s) =
数据线位数/8 × 总线工作频率（MHz） × 每个总线周期的传输次数