

# 章节导学

存储器

输入设备

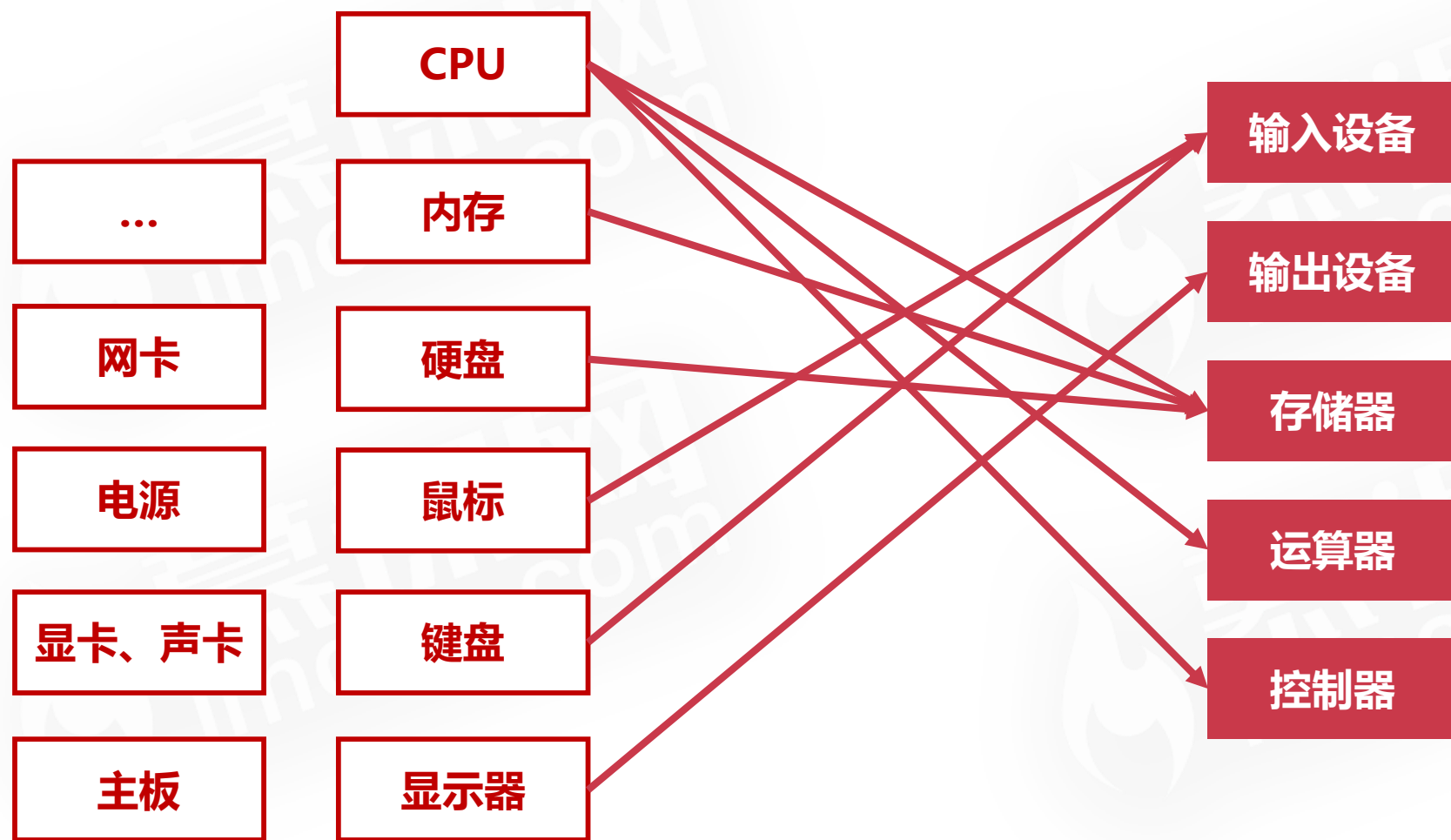
运算器

输出设备

控制器

冯诺依曼机

# 章节导学



计算机硬件

冯诺依曼机

# 章节导学

计算机的总线与IO设备

计算机的总线

计算机的输入/输出设备

计算机的存储器

计算机的存储器概览

计算机的高速存储器

计算机的主存储器  
与辅助存储器

计算机的CPU

计算机的指令系统

计算机的运算器

计算机的控制器

指令执行过程



# 计算机的总线

- ◆ 总线的概述
- ◆ 总线的仲裁

# 计算机的总线

## 总线的概述

- ◆ 总线的概述（是什么、什么用）
- ◆ 总线的分类

# 计算机的总线

## 总线的概述



USB

~~USB = "SB" ?~~

USB = Universal Serial Bus

通用串行总线

# 计算机的总线

## 总线的概述

- ◆ 提供了对外连接的接口
- ◆ 不同设备可以通过USB接口进行连接
- ◆ 连接的标准，促使外围设备接口的统一



通用串行总线（USB）

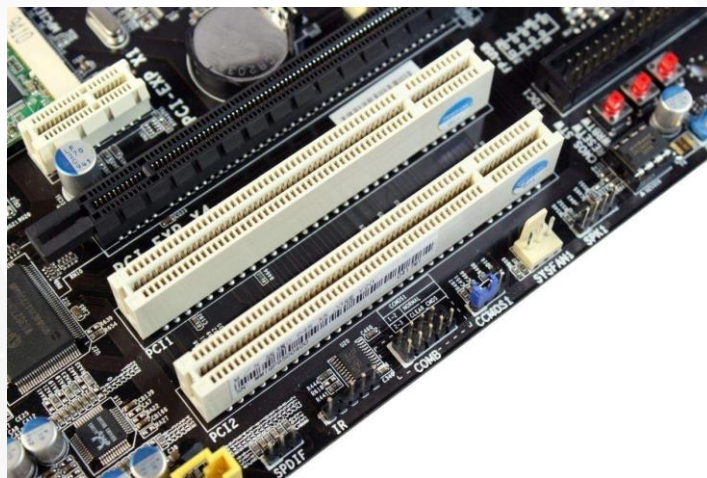


# 计算机的总线

## 总线的概述



USB



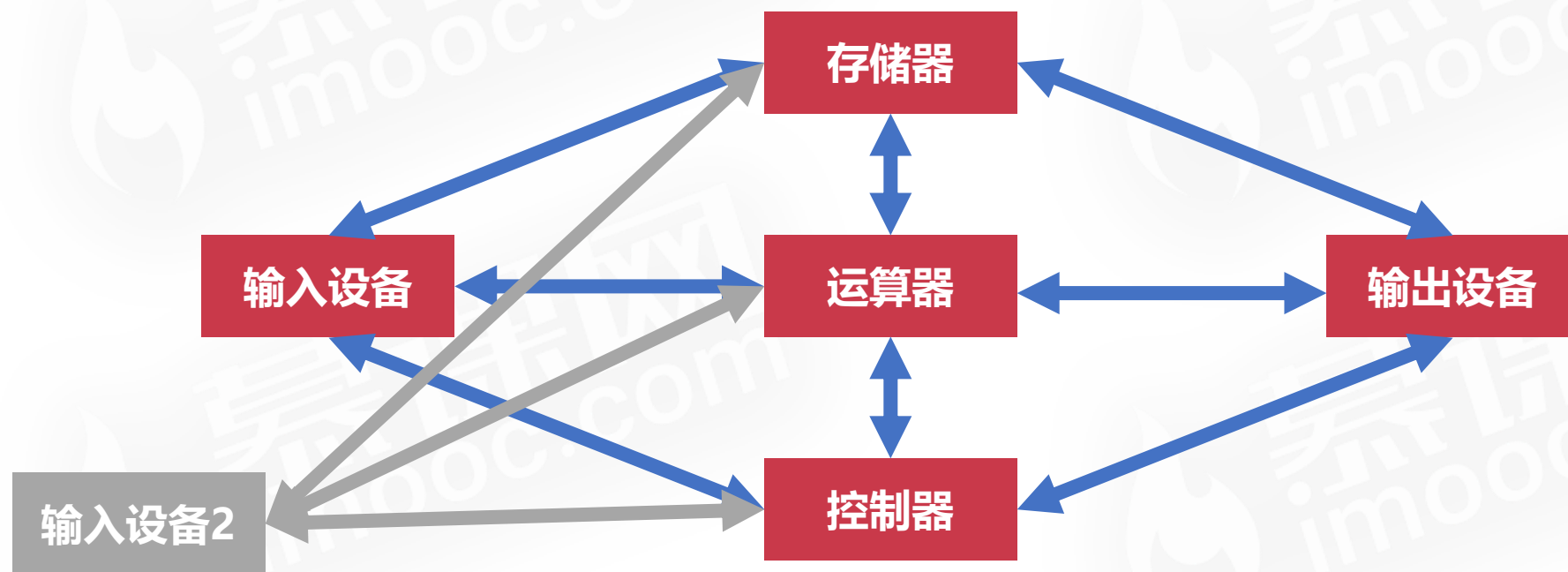
PCI总线

- ◆ ISA总线
- ◆ Thunderbolt总线
- ◆ ...

解决不同设备之间的通信问题

# 计算机的总线

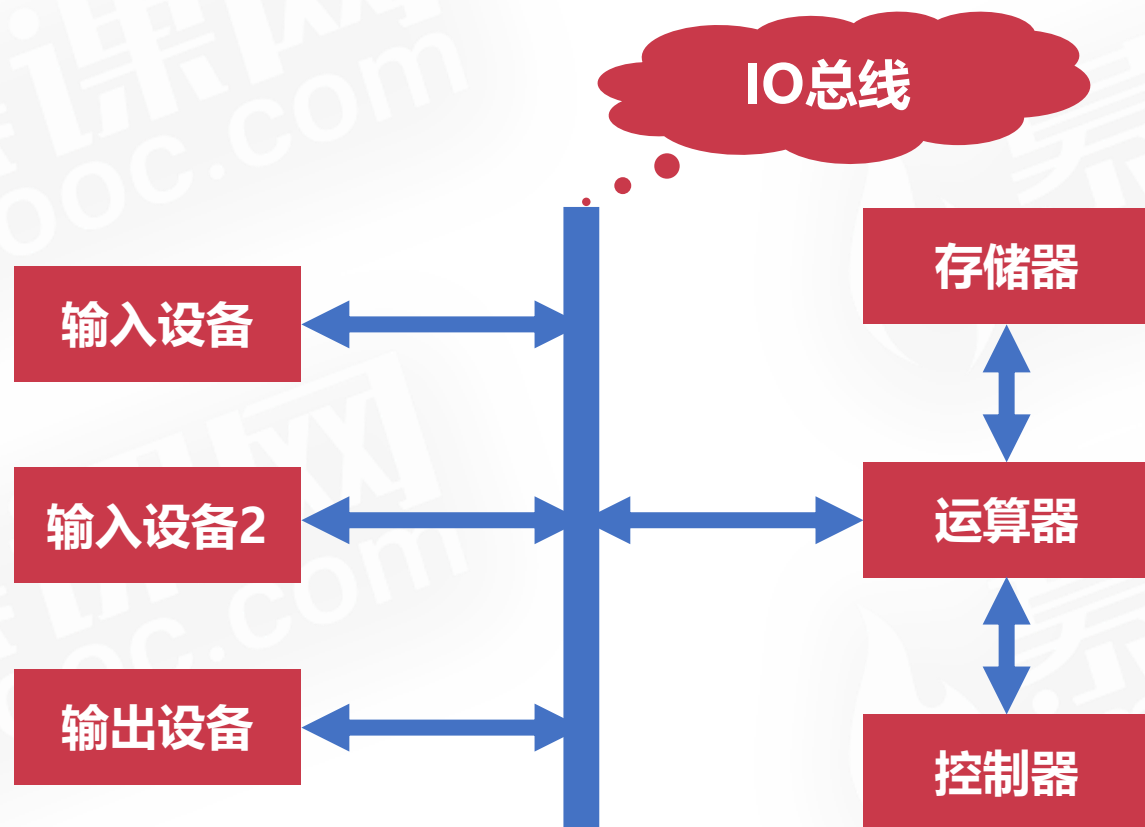
## 总线的概述



分散连接

# 计算机的总线

## 总线的概述



总线连接

# 计算机的总线

## 总线的概述

- ◆ 总线的概述（是什么、什么用）
- ◆ 总线的分类

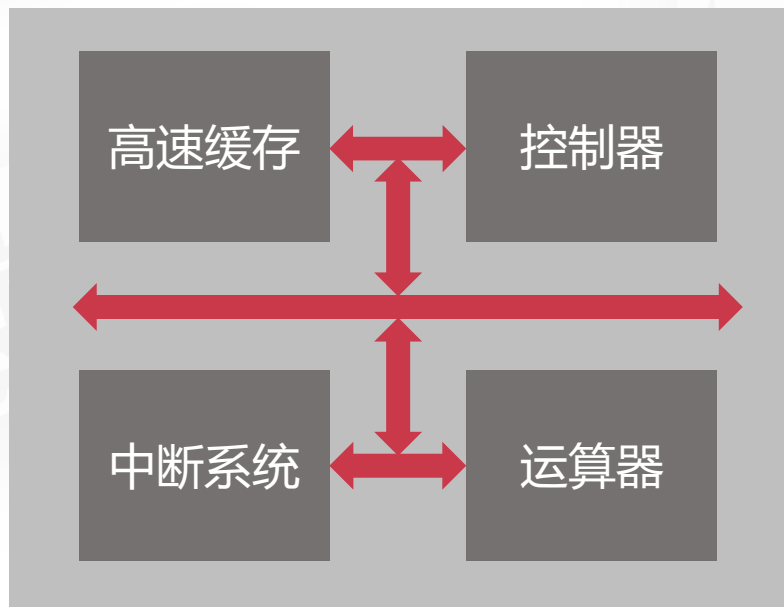
# 计算机的总线

## 总线的分类



# 计算机的总线

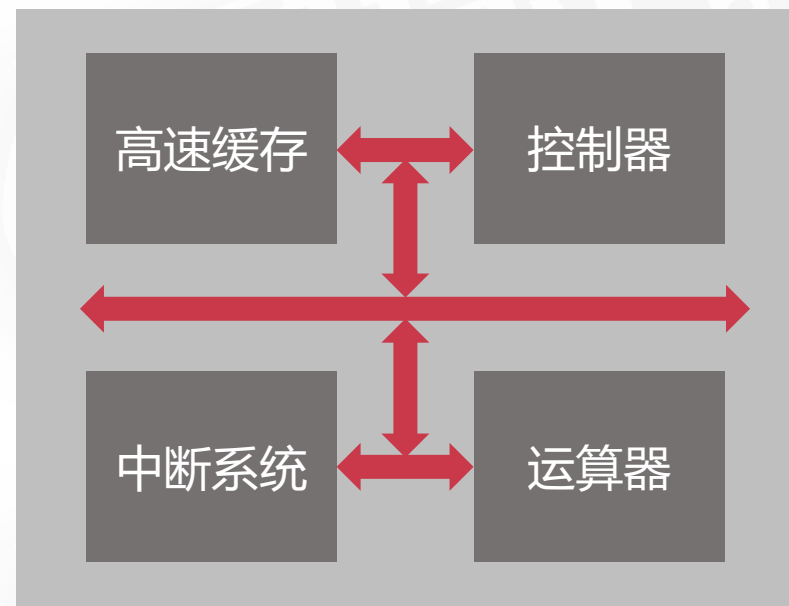
## 片内总线



# 总线的概述

## 片内总线

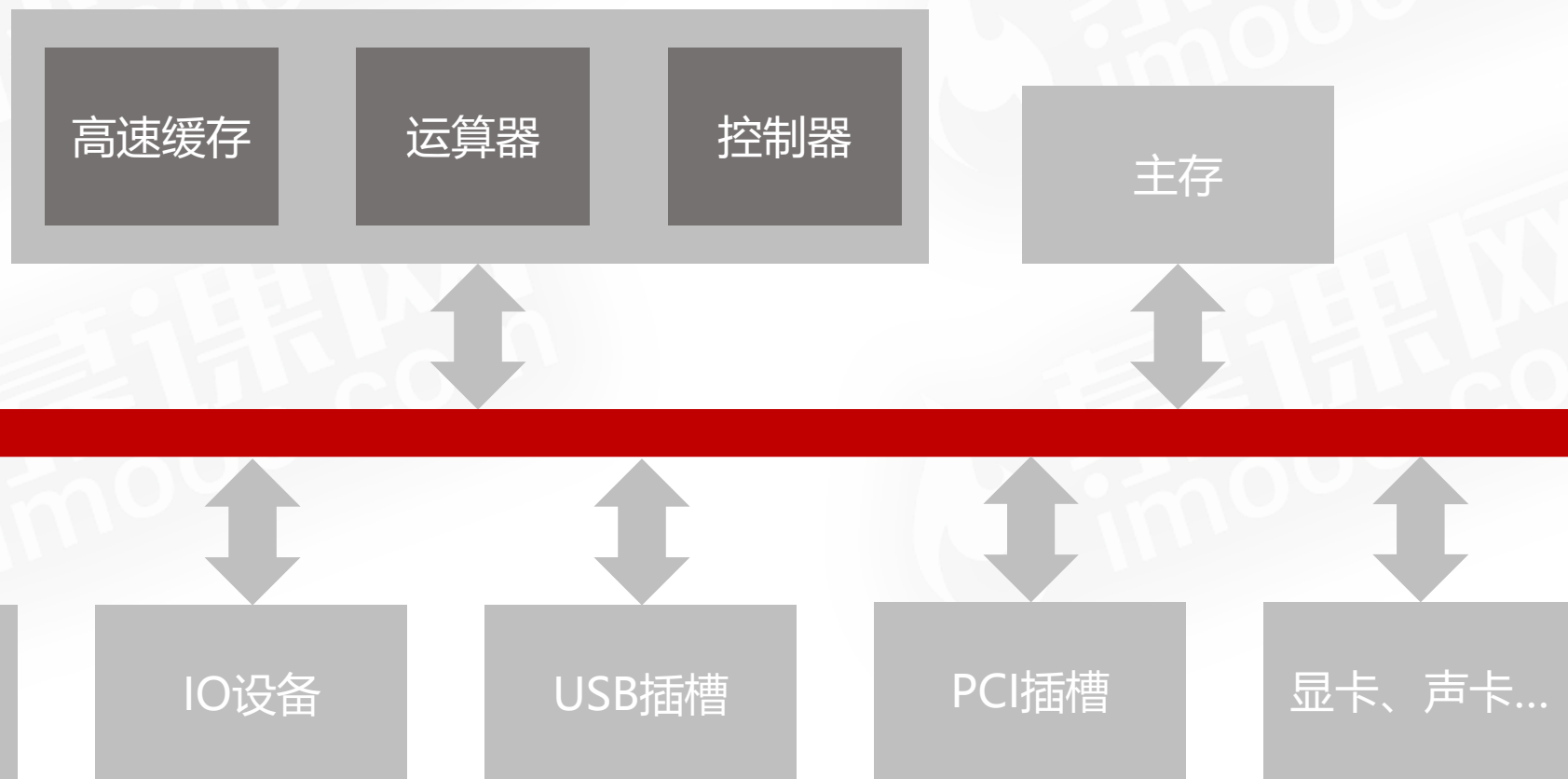
- ◆ 芯片内部的总线
- ◆ 寄存器与寄存器之间
- ◆ 寄存器与控制器、运算器之间



高集成度芯片内部的信息传输线

# 计算机的总线

系统总线





# 总线的概述

## 系统总线

- ◆ 数据总线
- ◆ 地址总线
- ◆ 控制总线

CPU、主内存、IO设备、各组件之间的信息传输线

# 总线的概述

## 数据总线

- ◆ 双向传输各个部件的数据信息
- ◆ 数据总线的位数（总线宽度）是数据总线的重要参数

一般与CPU位数相同（32位、64位）

# 总线的概述

## 地址总线

- ◆ 指定源数据或目的数据在内存中的地址
- ◆ 地址总线的位数与存储单元有关

地址总线位数= $n$ ，寻址范围： $0 \sim 2^n$

# 总线的概述

## 控制总线

- ◆ 控制总线是用来发出各种控制信号的传输线
- ◆ 控制信号经由控制总线从一个组件发给另外一个组件
- ◆ 控制总线可以监视不同组件之间的状态（就绪/未就绪）

# 计算机的总线

- ◆ 总线的概述
- ◆ 总线的仲裁

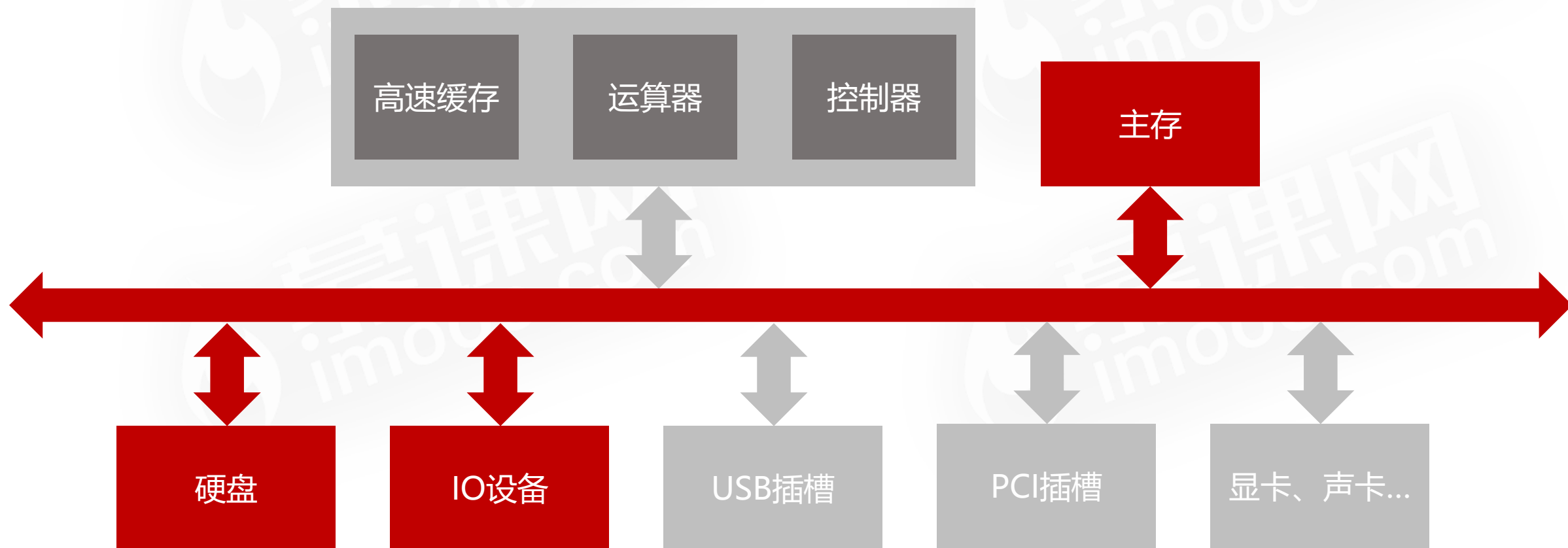
# 计算机的总线

## 总线的仲裁

- ◆ 为什么需要总线仲裁
- ◆ 总线仲裁的方法

# 计算机的总线

## 总线的仲裁



# 计算机的总线

## 总线的仲裁

- ◆ 为什么需要总线仲裁
- ◆ 总线仲裁的方法



# 计算机的总线

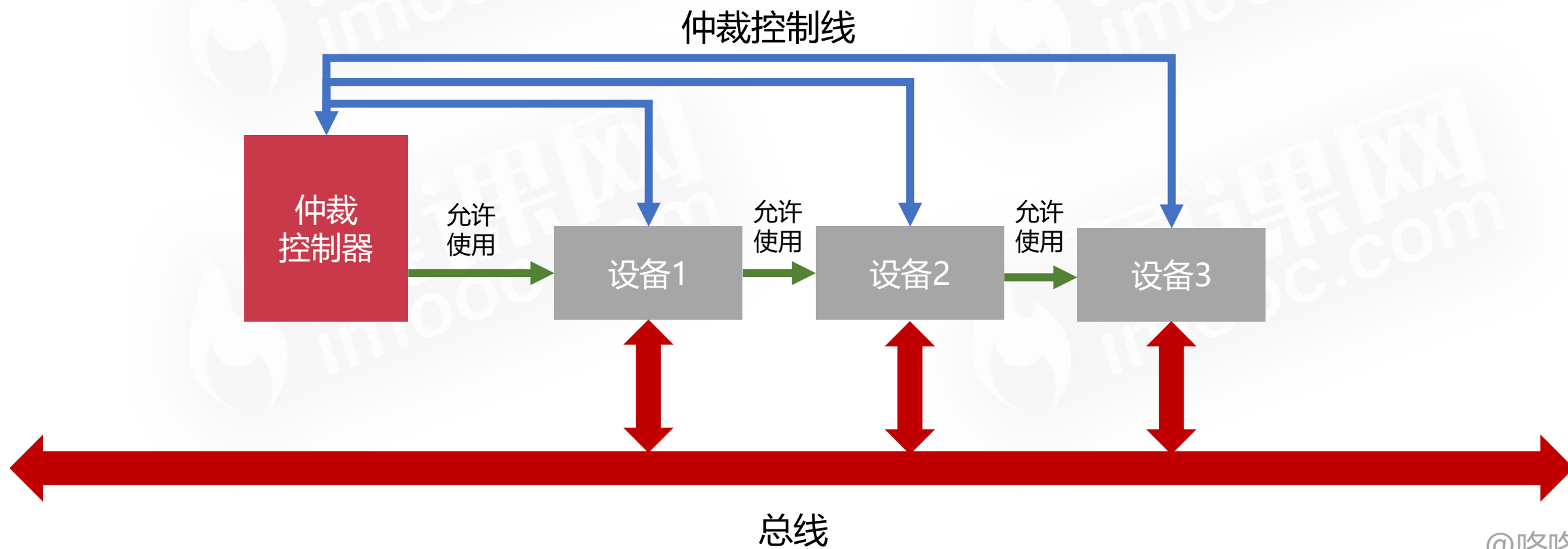
## 总线的仲裁方法

- ◆ 链式查询
- ◆ 计数器定时查询
- ◆ 独立请求

为了解决总线使用权的冲突问题

# 总线的仲裁

## 链式查询



# 总线的仲裁

## 链式查询

- ◆ 好处：电路复杂度低，仲裁方式简单
- ◆ 坏处：优先级低的设备难以获得总线使用权
- ◆ 坏处：对电路故障敏感

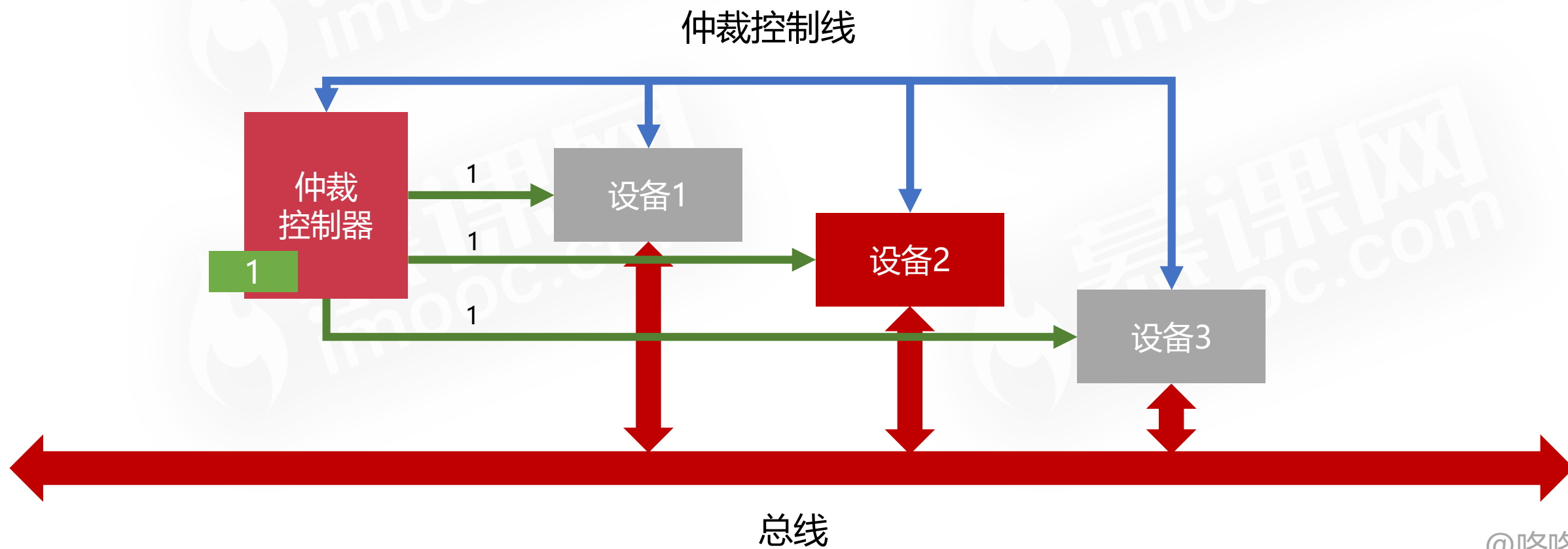
# 总线的仲裁

## 计时器定时查询

- ◆ 仲裁控制器对设备编号并使用计数器累计计数
- ◆ 接收到仲裁信号后，往所有设备发出计数值
- ◆ 计数值与设备编号一致则获得总线使用权

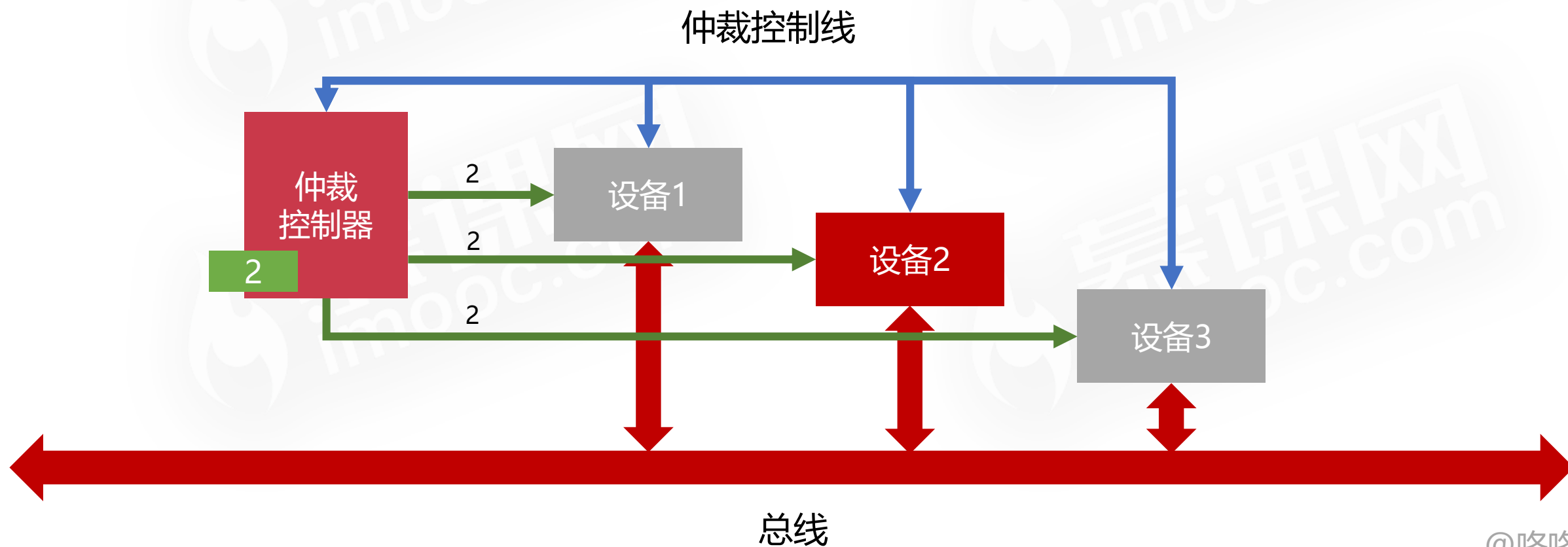
# 总线的仲裁

## 计时器定时查询



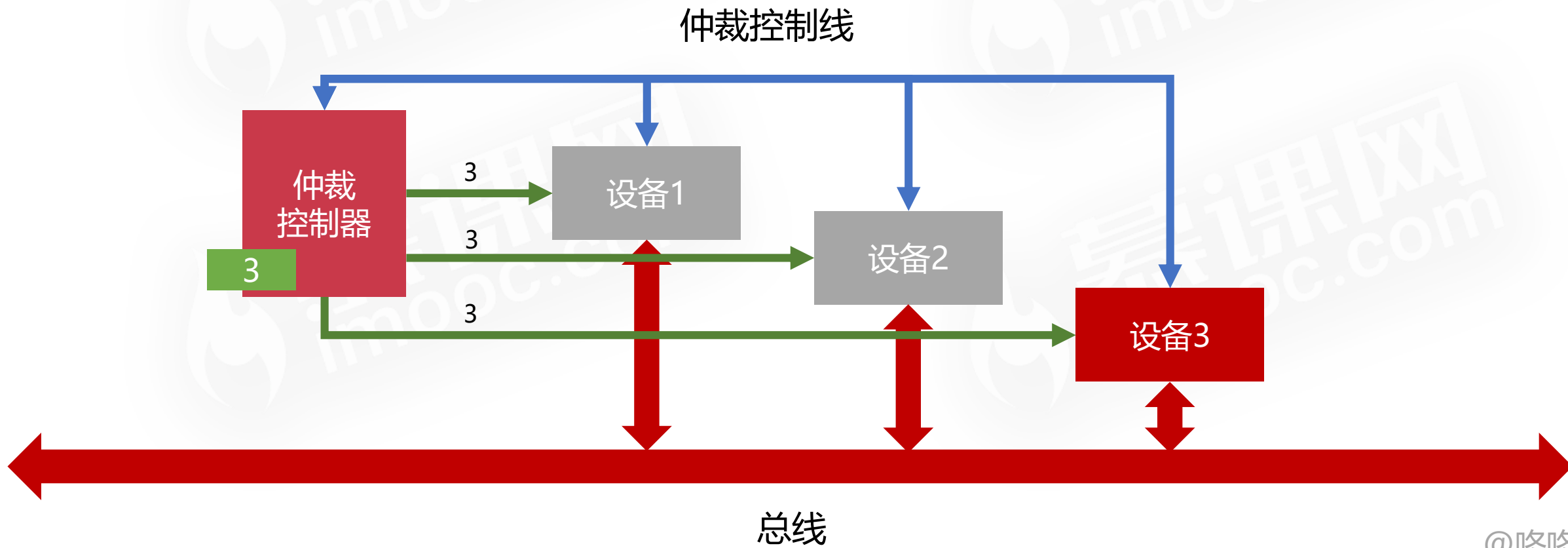
# 总线的仲裁

## 计时器定时查询



# 总线的仲裁

## 计时器定时查询



# 总线的仲裁

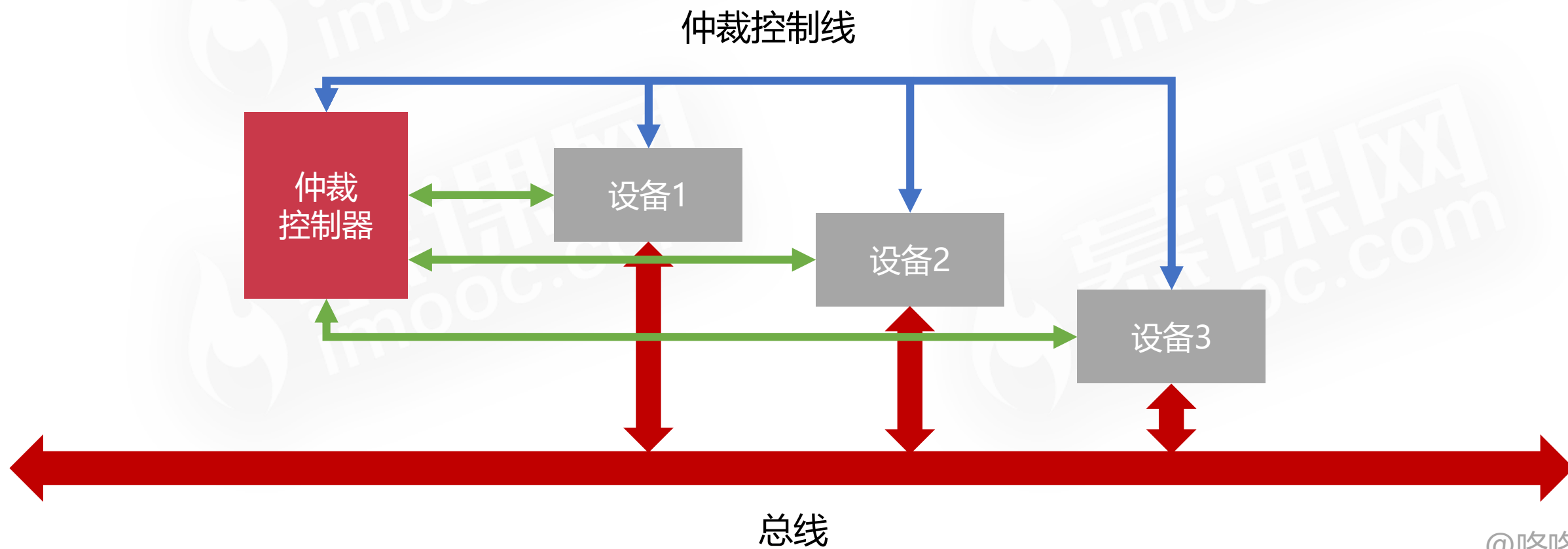
## 独立请求

- ◆ 每个设备均有总线独立连接仲裁器
- ◆ 设备可单独向仲裁器发送请求和接收请求
- ◆ 当同时收到多个请求信号，仲裁器有权按优先级分配使用权



# 总线的仲裁

独立请求



# 总线的仲裁

## 独立请求

- ◆ 好处：响应速度快，优先顺序可动态改变
- ◆ 设备连线多，总线控制复杂

# 计算机的总线

- ◆ 总线的概述
- ◆ 总线的仲裁



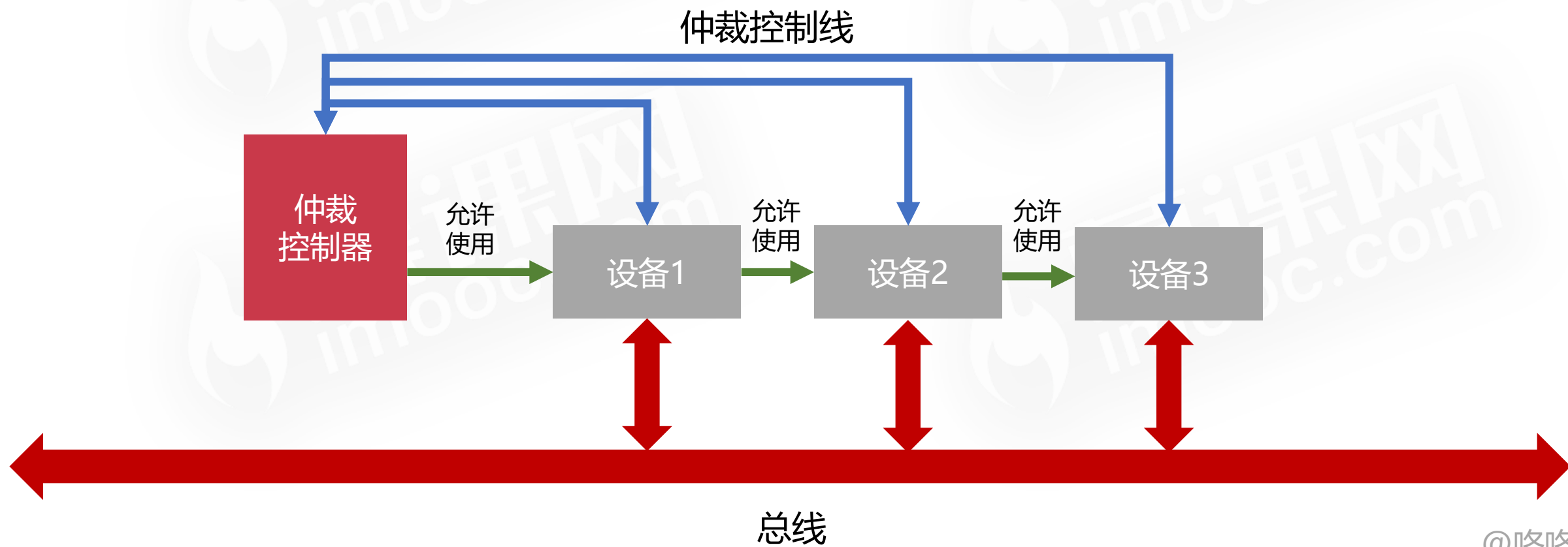
# 计算机的总线

## 总线的仲裁

- ◆ 串联优先级判别
- ◆ 并联优先级判别
- ◆ 循环优先级判别

# 总线的仲裁

## 串联优先级判别



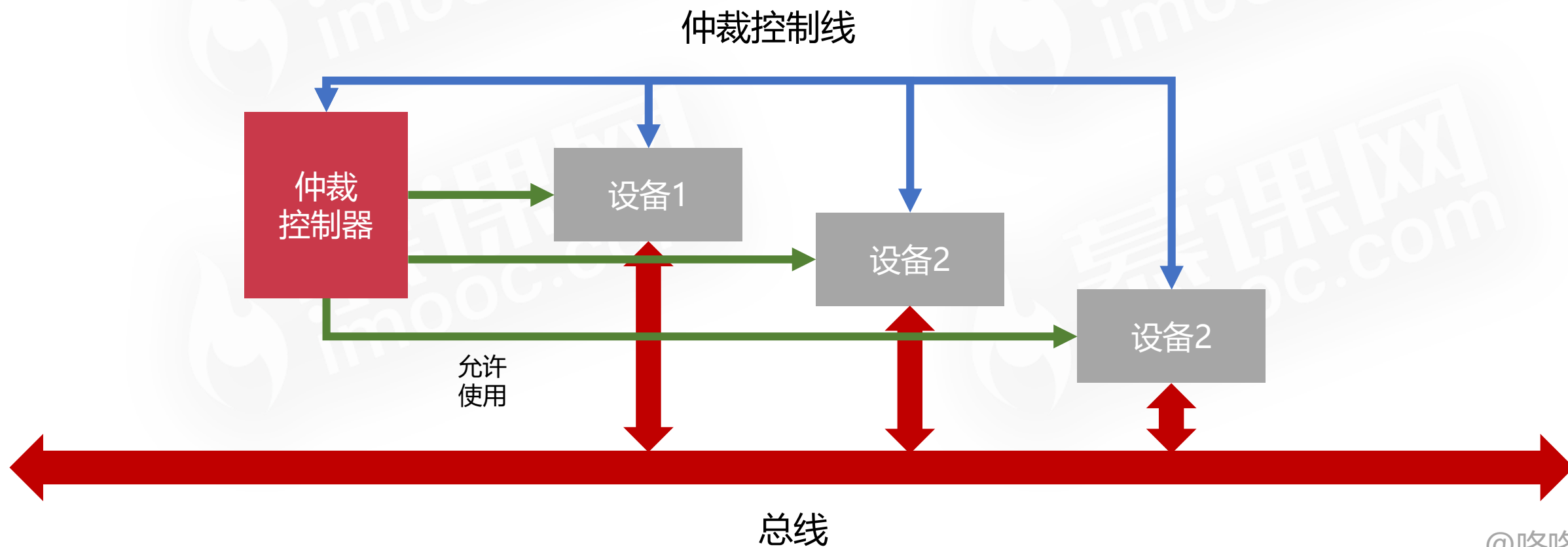
# 总线的仲裁

## 串联优先级判别

- ◆ 优先级高的设备经常得到总线使用权限
- ◆ 优先级低的设备长时间无法得到总线

# 总线的仲裁

## 并联优先级判别





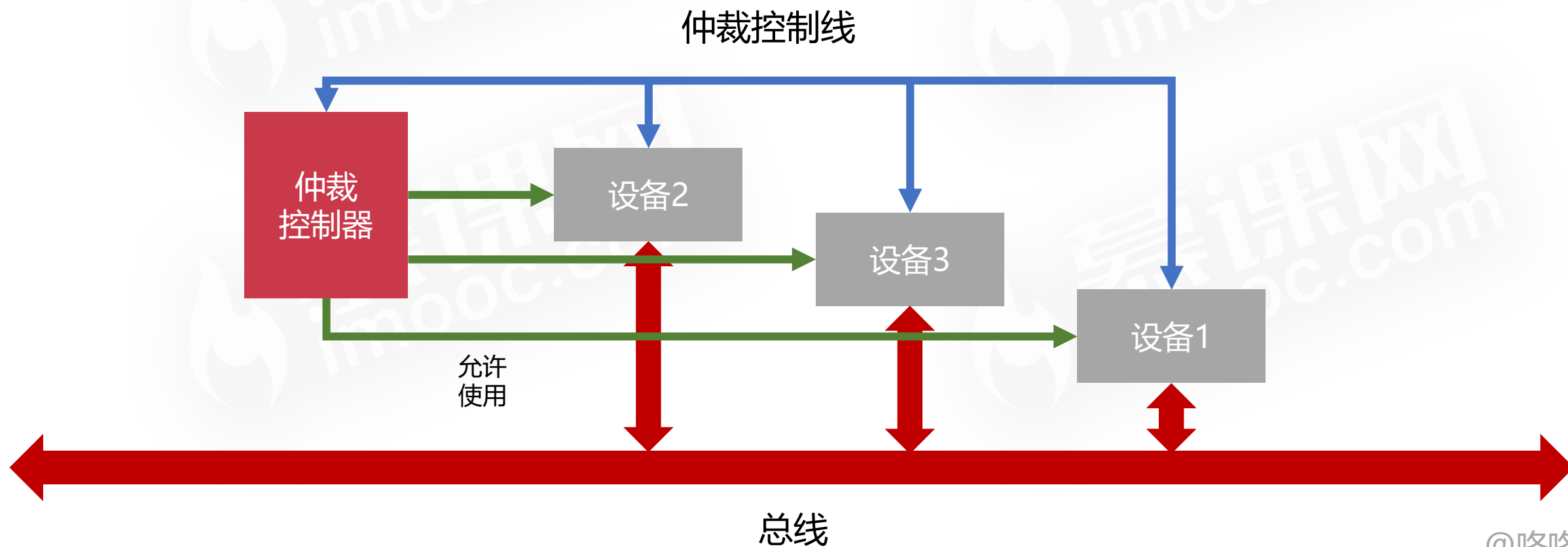
# 总线的仲裁

## 并联优先级判别

- ◆ 优先级高的设备经常得到总线使用权限
- ◆ 优先级低的设备长时间无法得到总线

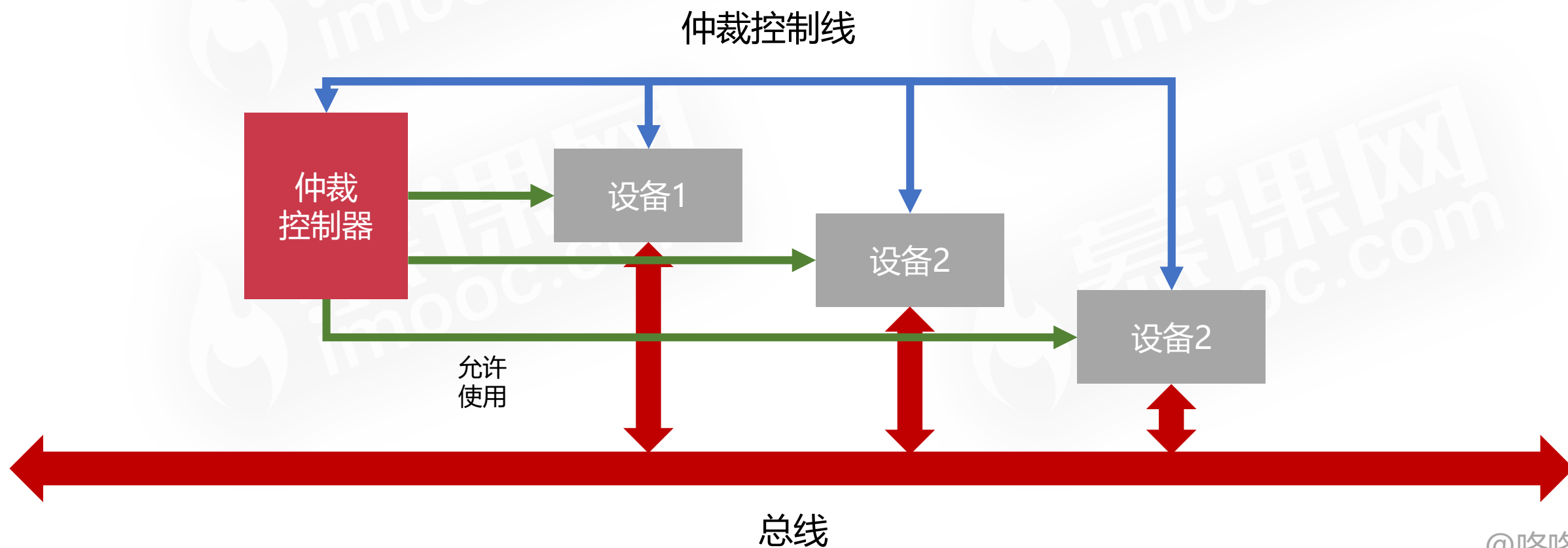
# 总线的仲裁

## 循环优先级判别



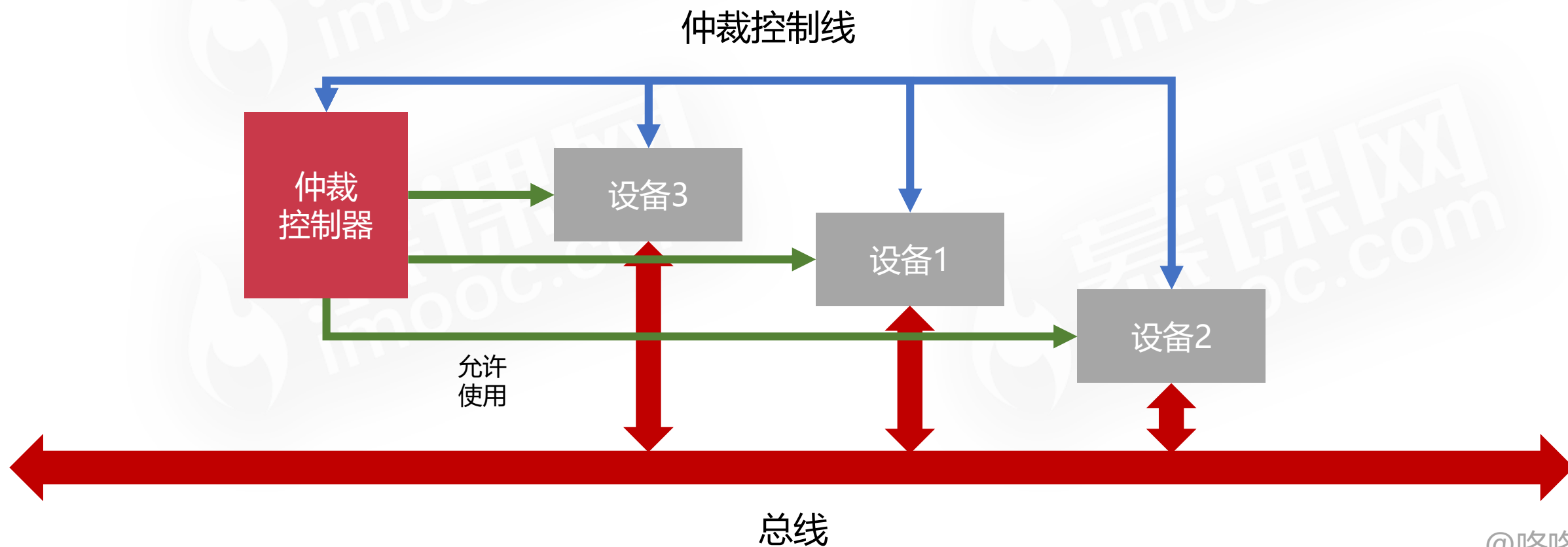
# 总线的仲裁

## 循环优先级判别



# 总线的仲裁

## 循环优先级判别



# 计算机的总线

- ◆ 总线的概述
- ◆ 总线的仲裁

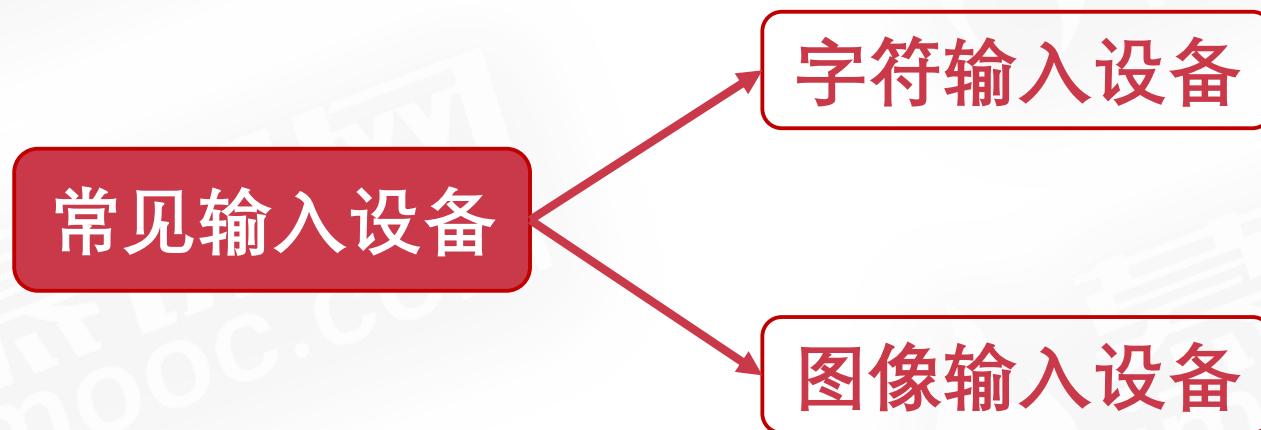


# 计算机的输入输出设备

- ◆ 常见的输入输出设备
- ◆ 输入输出接口的通用设计
- ◆ CPU与IO设备的通信

# 计算机的输入输出设备

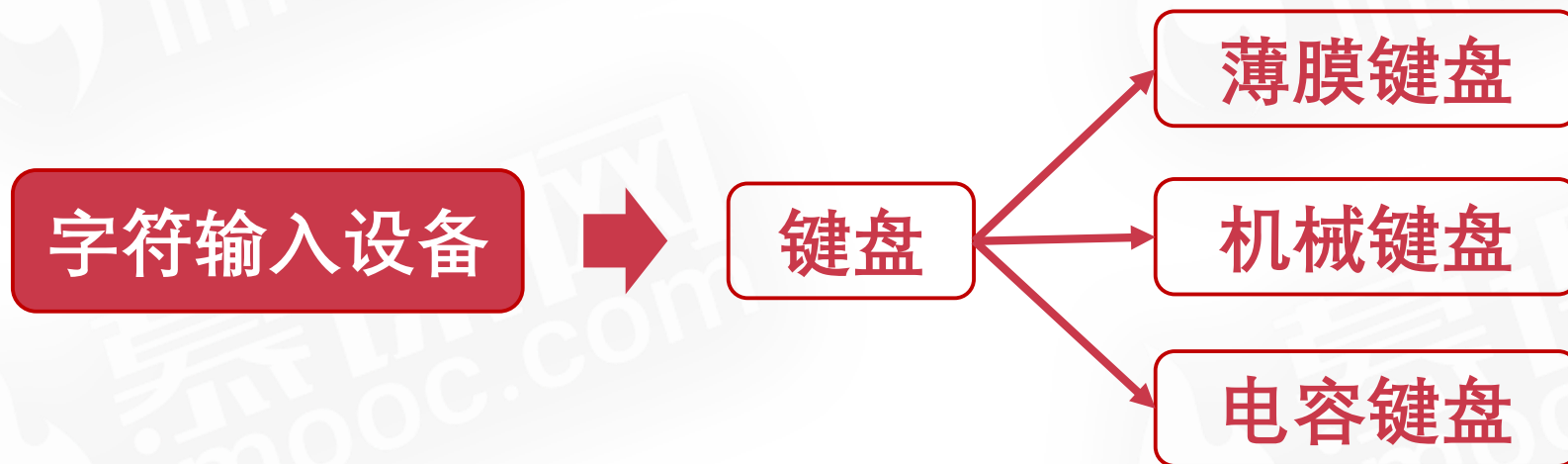
## 常见的输入输出设备





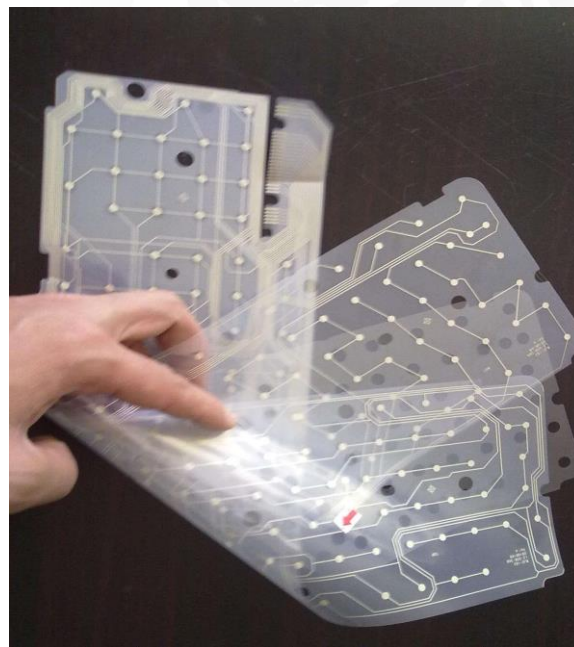
# 计算机的输入输出设备

## 常见的输入输出设备



# 计算机的输入输出设备

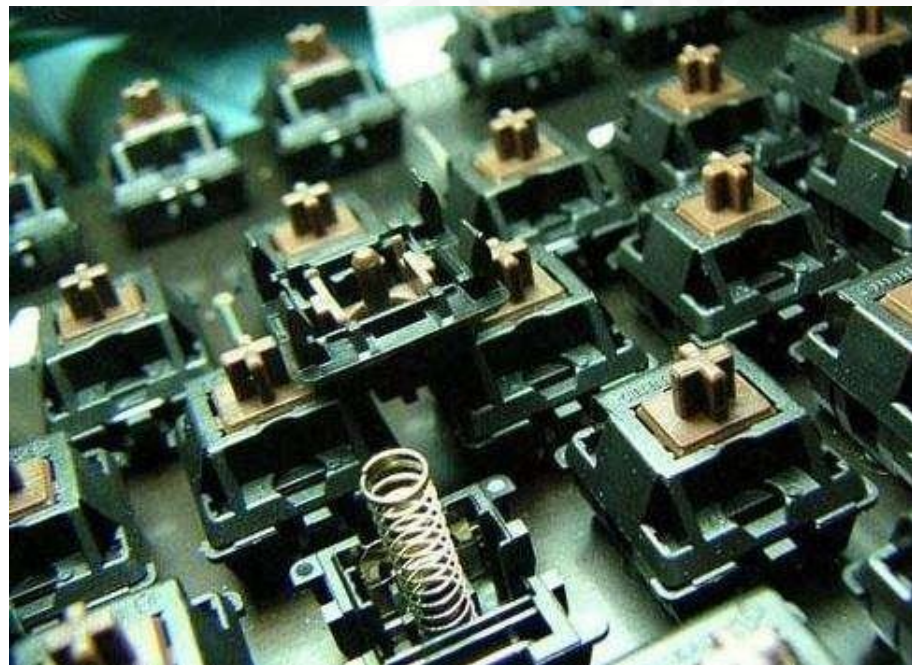
## 常见的输入输出设备



薄膜键盘

# 计算机的输入输出设备

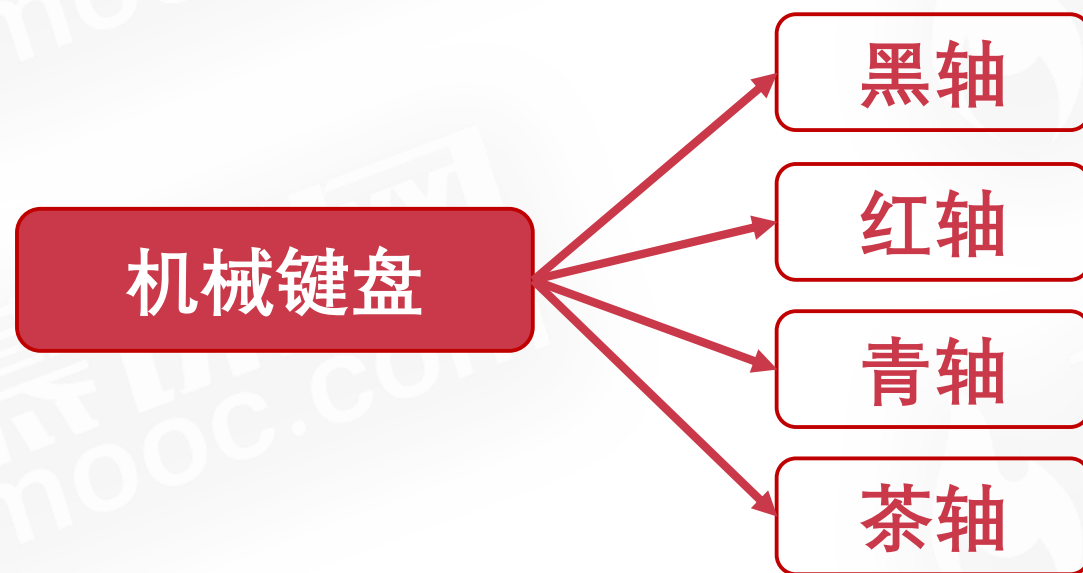
## 常见的输入输出设备



机械键盘

# 计算机的输入输出设备

## 常见的输入输出设备



段落感

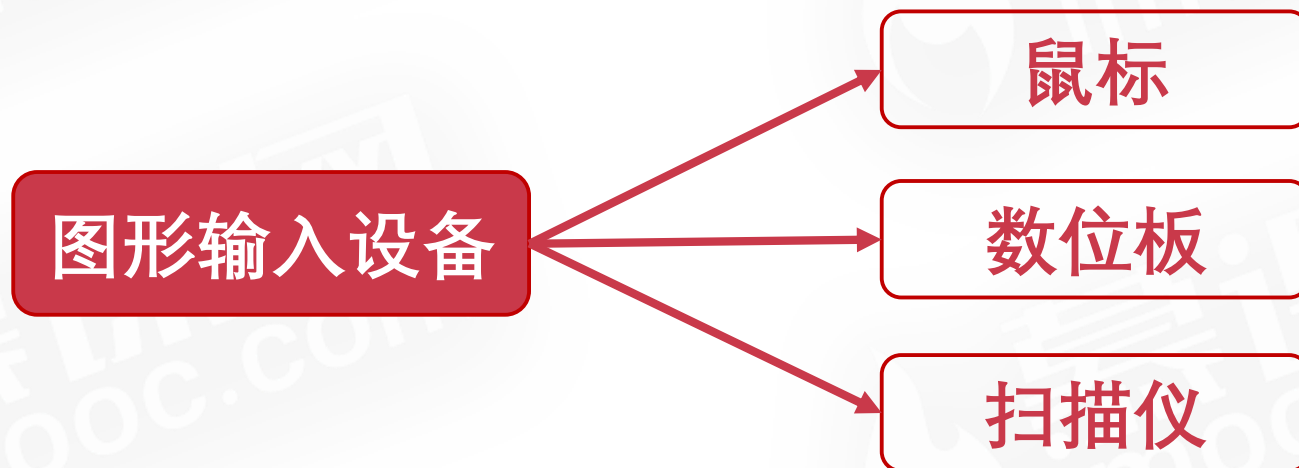
声音

压力

键程

# 计算机的输入输出设备

## 常见的输入输出设备





# 计算机的输入输出设备

## 常见的输入输出设备



鼠标

# 计算机的输入输出设备

## 常见的输入输出设备



数位板

- ◆ 输入板和压感笔
- ◆ 常用于绘图设计创作

# 计算机的输入输出设备

## 常见的输入输出设备



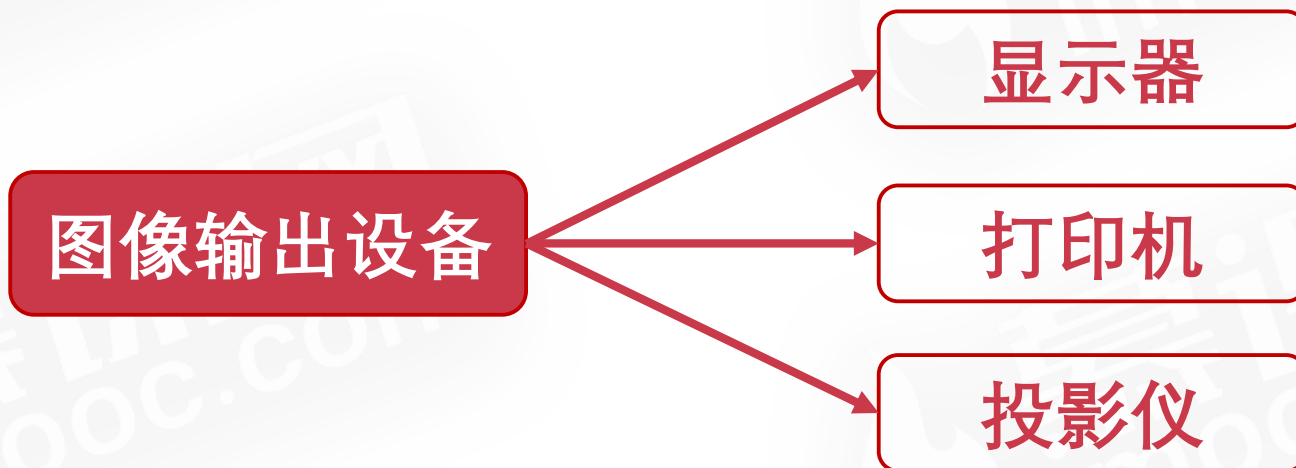
扫描仪

- ◆ 将图形信息转换为数字信号



# 计算机的输入输出设备

## 常见的输入输出设备



# 计算机的输入输出设备

## 常见的输入输出设备



CRT显示器



液晶显示器

# 计算机的输入输出设备

## 常见的输入输出设备



打印机

# 计算机的输入输出设备

## 常见的输入输出设备



投影仪

# 计算机的输入输出设备

- ◆ 常见的输入输出设备
- ◆ 输入输出接口的通用设计

# 计算机的输入输出设备

## 输入输出接口的通用设计

向设备发送数据？

读取数据？

设备有没有被占用？

设备是否已经启动？

设备是否已经连接？



# 计算机的输入输出设备

## 输入输出接口的通用设计

- ◆ 数据线
- ◆ 状态线
- ◆ 命令线
- ◆ 设备选择线

# 计算机的输入输出设备

## 输入输出接口的通用设计

- ◆ 是I/O设备与主机之间进行数据交换的传送线
- ◆ 单向传输数据线
- ◆ 双向传输数据线

数据线



# 计算机的输入输出设备

## 输入输出接口的通用设计

- ◆ IO设备状态向主机报告的信号线
- ◆ 查询设备是否已经正常连接并就绪
- ◆ 查询设备是否已经被占用

状态线

# 计算机的输入输出设备

## 输入输出接口的通用设计

- ◆ CPU向设备发送命令的信号线
- ◆ 发送读写信号
- ◆ 发送启动停止信号

命令线

# 计算机的输入输出设备

## 输入输出接口的通用设计

- ◆ 主机选择I/O设备进行操作的信号线
- ◆ 对连在总线上的设备进行选择

设备选择线

# 计算机的输入输出设备

- ◆ 常见的输入输出设备
- ◆ 输入输出接口的通用设计
- ◆ CPU与IO设备的通信

# 计算机的输入输出设备

## CPU与IO设备的通信

- ◆ 程序中中断
- ◆ DMA（直接存储器访问）

CPU速度与IO设备速度不一致

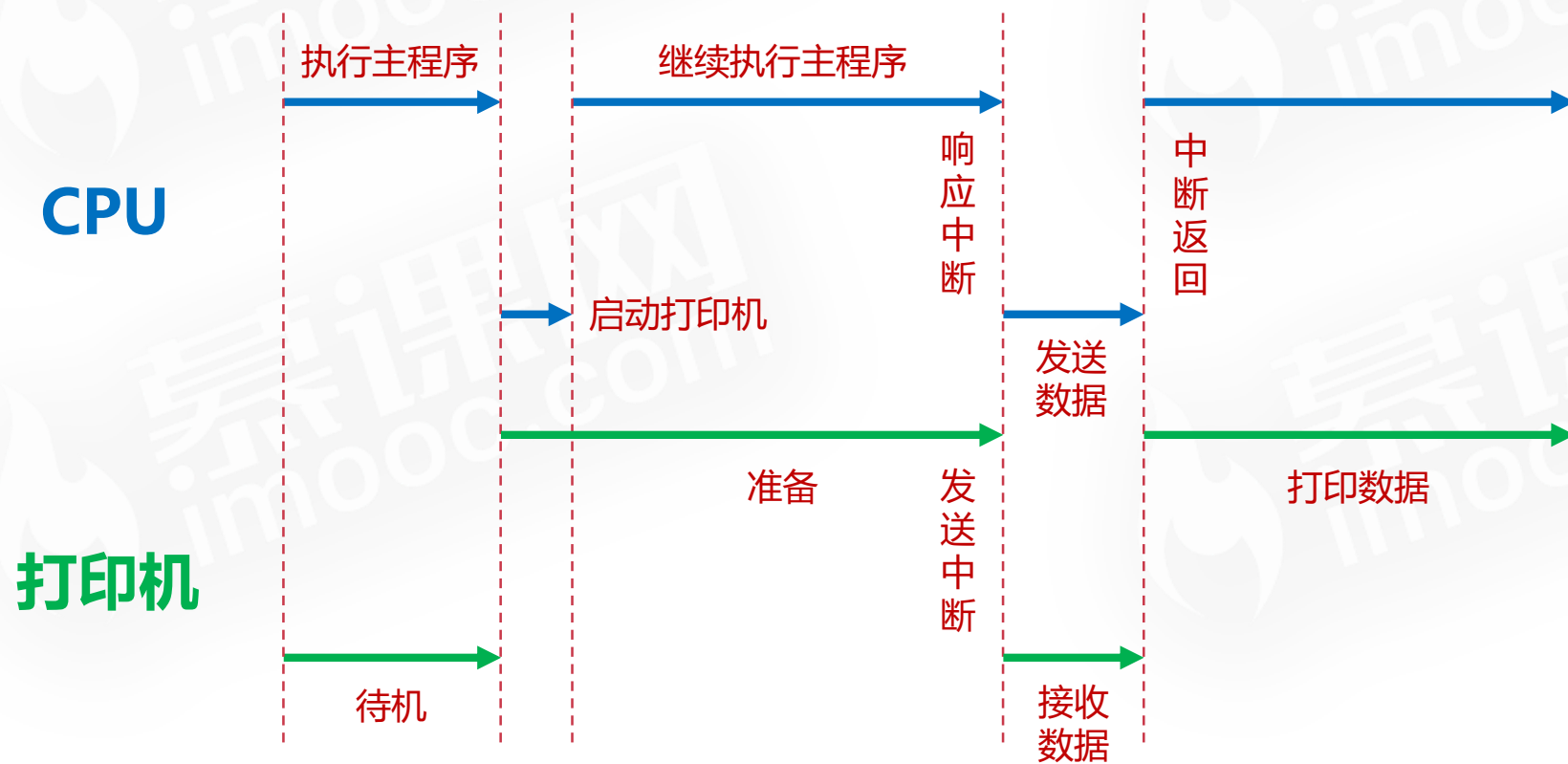
# 计算机的输入输出设备

## 程序中断

- ◆ 当外围IO设备就绪时，向CPU发出中断信号
- ◆ CPU有专门的电路响应中断信号

# 计算机的输入输出设备

## 程序中断



# 计算机的输入输出设备

## 程序中中断

提供低速设备通知CPU的一种异步的方式

CPU可以高速运转同时兼顾低速设备的响应



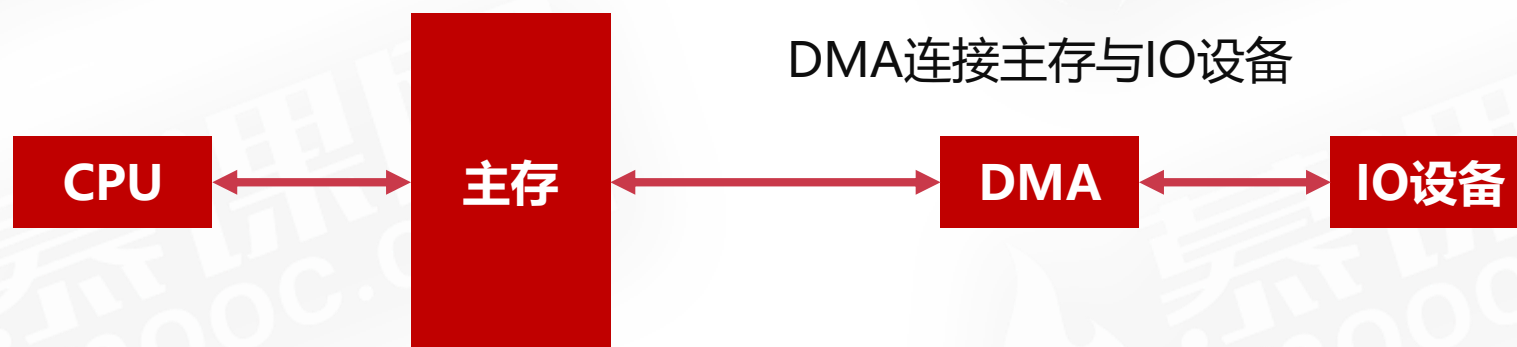
# 计算机的输入输出设备

## DMA（直接存储器访问）

- ◆ DMA直接连接主存与IO设备
- ◆ DMA工作时不需要CPU的参与

# 计算机的输入输出设备

## DMA（直接存储器访问）



当主存与IO设备交换信息时，不需要中断CPU

可以提高CPU的效率

# 计算机的输入输出设备

## DMA（直接存储器访问）

- ◆ 硬盘
- ◆ 外置显卡

# 计算机的输入输出设备

- ◆ 常见的输入输出设备
- ◆ 输入输出接口的通用设计
- ◆ CPU与IO设备的通信



# 计算机存储器概览

- ◆ 存储器的分类
- ◆ 存储器的层次结构

# 计算机存储器概览

## 存储器的分类



# 计算机存储器概览

## 存储器的分类





# 计算机存储器概览

- ◆ 存储器的分类
- ◆ 存储器的层次结构

# 计算机存储器概览

## 存储器的层次结构

读写速度

5400转

7200转

存储容量

2T

4G

价格



# 计算机存储器概览

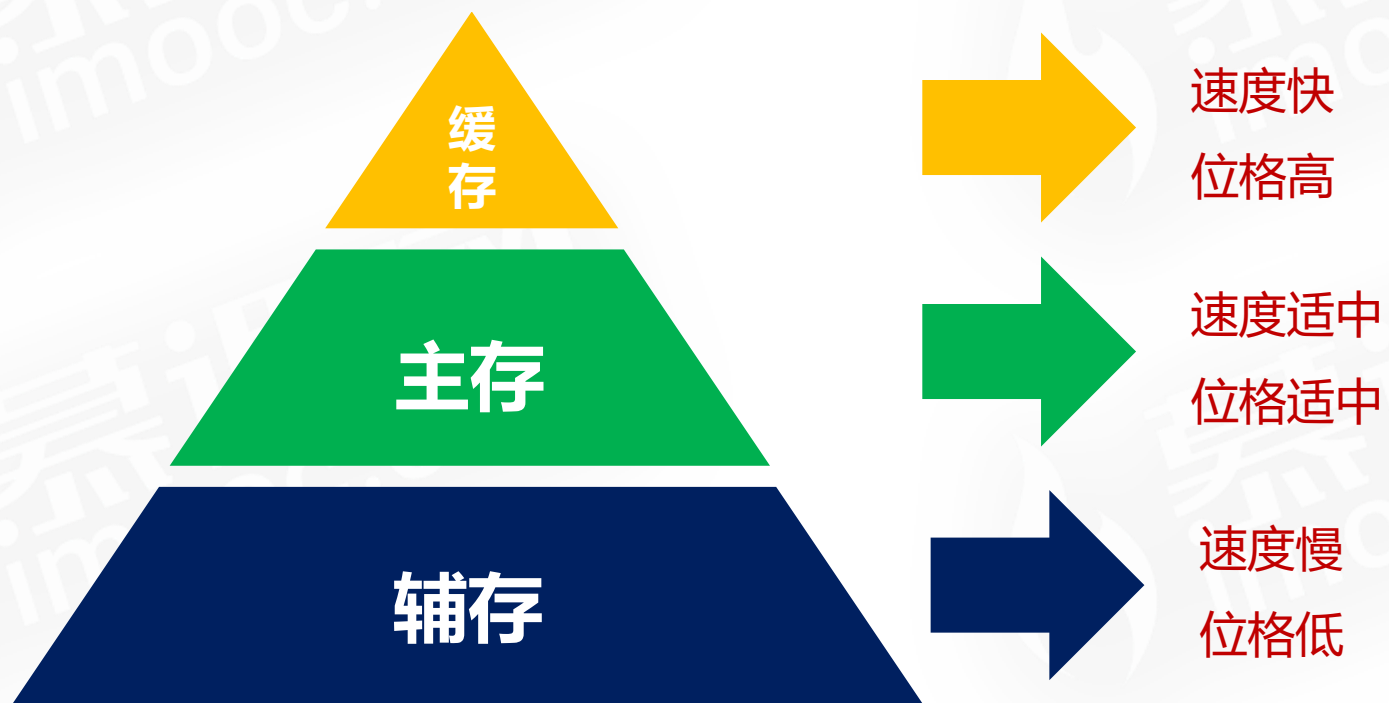
## 存储器的层次结构



容量+价格=>位价：每比特位价格

# 计算机存储器概览

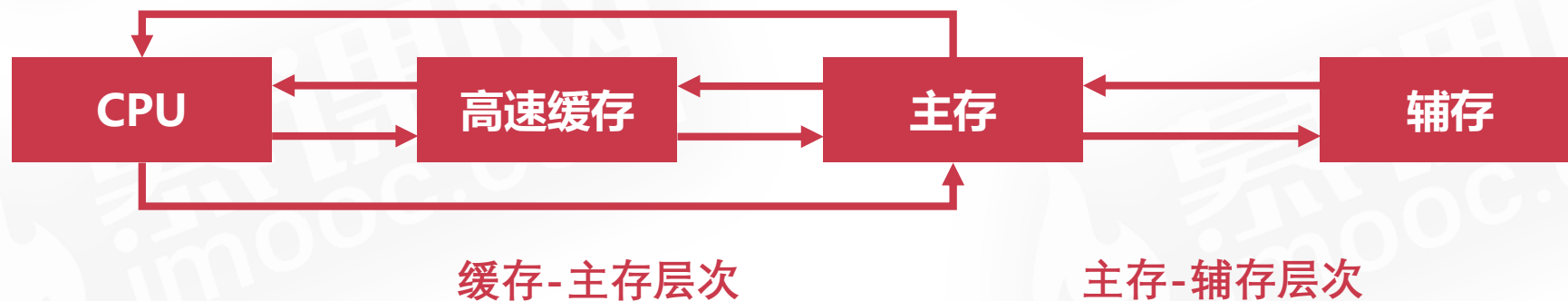
## 存储器的层次结构



存储器的层次结构

# 计算机存储器概览

## 存储器的层次结构



存储器的层次结构

# 计算机存储器概览

## 存储器的层次结构

- ◆ 原理：局部性原理
- ◆ 实现：在CPU与主存之间增加一层速度快（容量小）的Cache
- ◆ 目的：解决主存速度不足的问题

缓存-主存层次

# 计算机存储器概览

## 存储器的层次结构

局部性原理是指CPU访问存储器时，无论是存取指令还是存取数据，所访问的存储单元都趋于聚集在一个较小的连续区域中。

局部性原理

# 计算机存储器概览

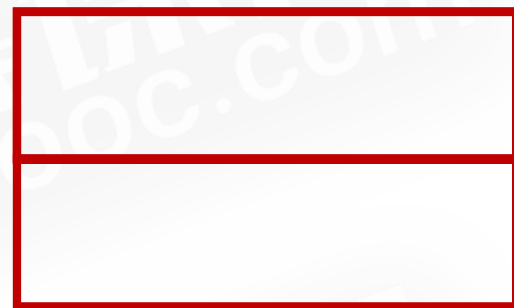
## 存储器的层次结构

只需要把这段内存  
置换至缓存即可。

程序经常访问的内存



局部性原理



...



内存



# 计算机存储器概览

## 存储器的层次结构

- ◆ 原理：局部性原理
- ◆ 实现：主存之外增加辅助存储器（磁盘、SD卡、U盘等）
- ◆ 目的：解决主存容量不足的问题

主存-辅存层次

# 计算机存储器概览

- ◆ 存储器的分类
- ◆ 存储器的层次结构



# 计算机的主存储器与辅助存储器

计算机断电，内存数据丢失

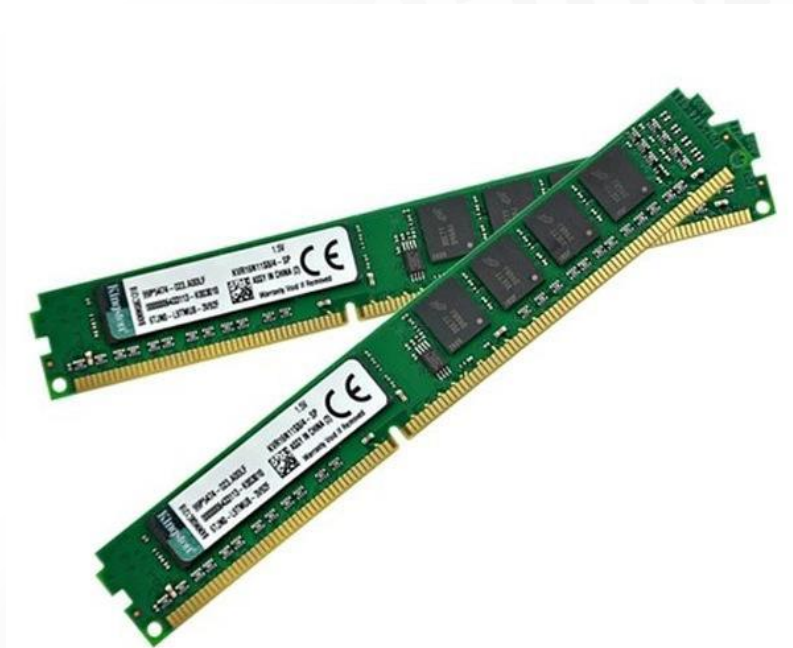
计算机断电，磁盘数据不会丢失



# 计算机的主存储器与辅助存储器

- ◆ 主存储器——内存
- ◆ 辅助存储器——磁盘

# 计算机的主存储器与辅助存储器



主存储器

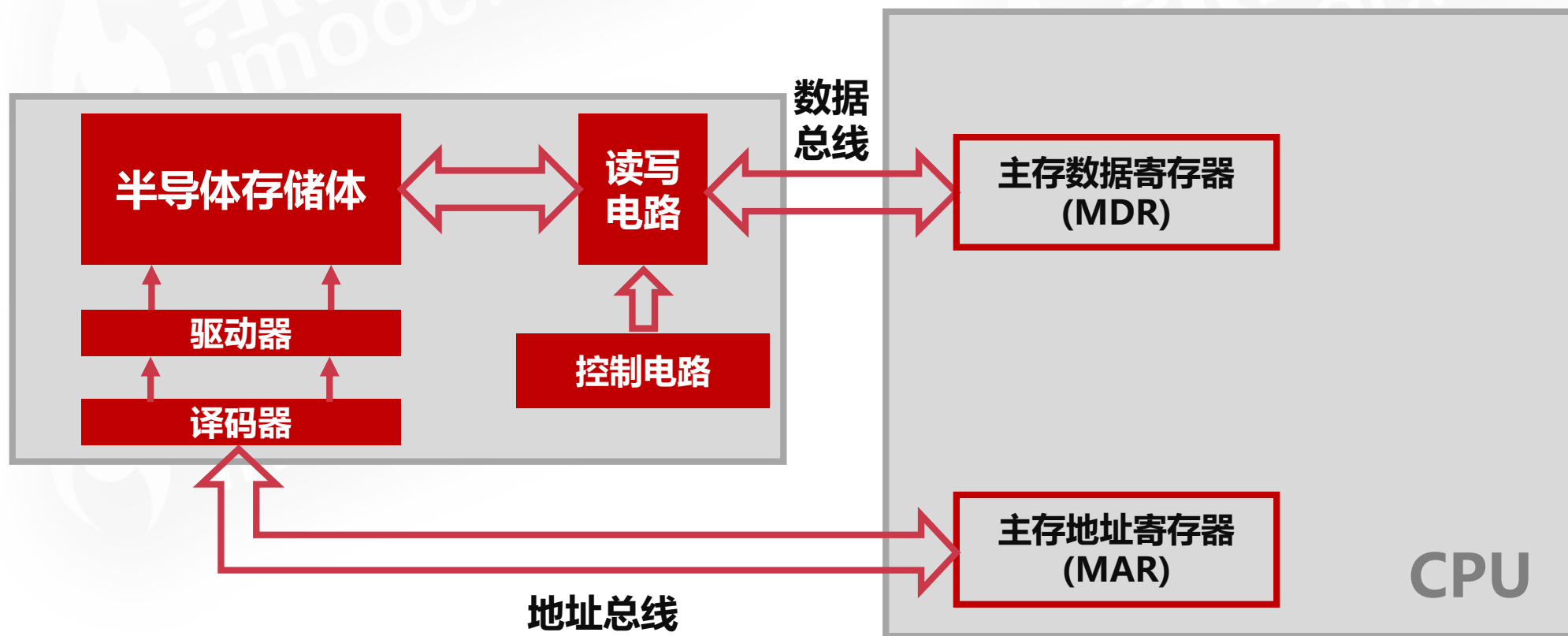
# 计算机的主存储器与辅助存储器

## 主存储器——内存

- ◆ RAM（随机存取存储器：Random Access Memory）
- ◆ RAM通过**电容**存储数据，必须隔一段时间刷新一次
- ◆ 如果掉电，那么一段时间后将丢失所有数据

# 计算机的主存储器与辅助存储器

## 主存储器——内存





# 计算机的主存储器与辅助存储器

## 主存储器——内存

32位系统

$$2^{32} = 4 \times 2^{30} = 4GB$$

64位系统

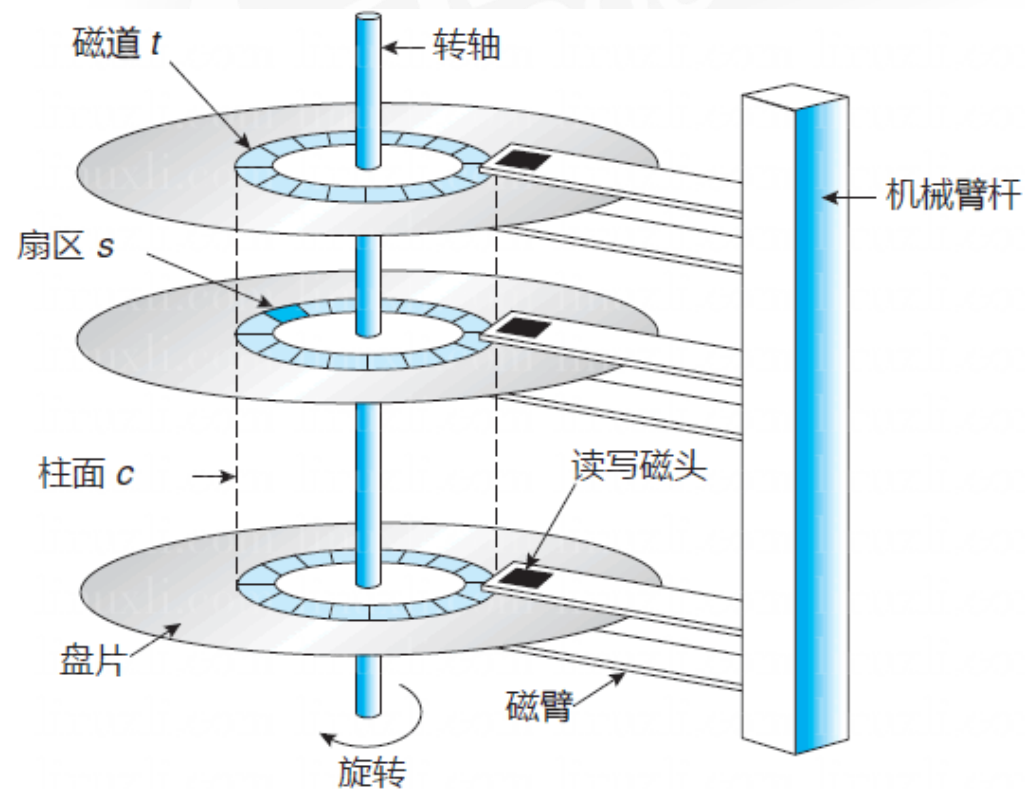
$$2^{64} = 2^{34} \times 2^{30} = 2^{34}GB$$

# 计算机的主存储器与辅助存储器

- ◆ 主存储器——内存
- ◆ 辅助存储器——磁盘

# 计算机的主存储器与辅助存储器

## 辅助存储器——磁盘

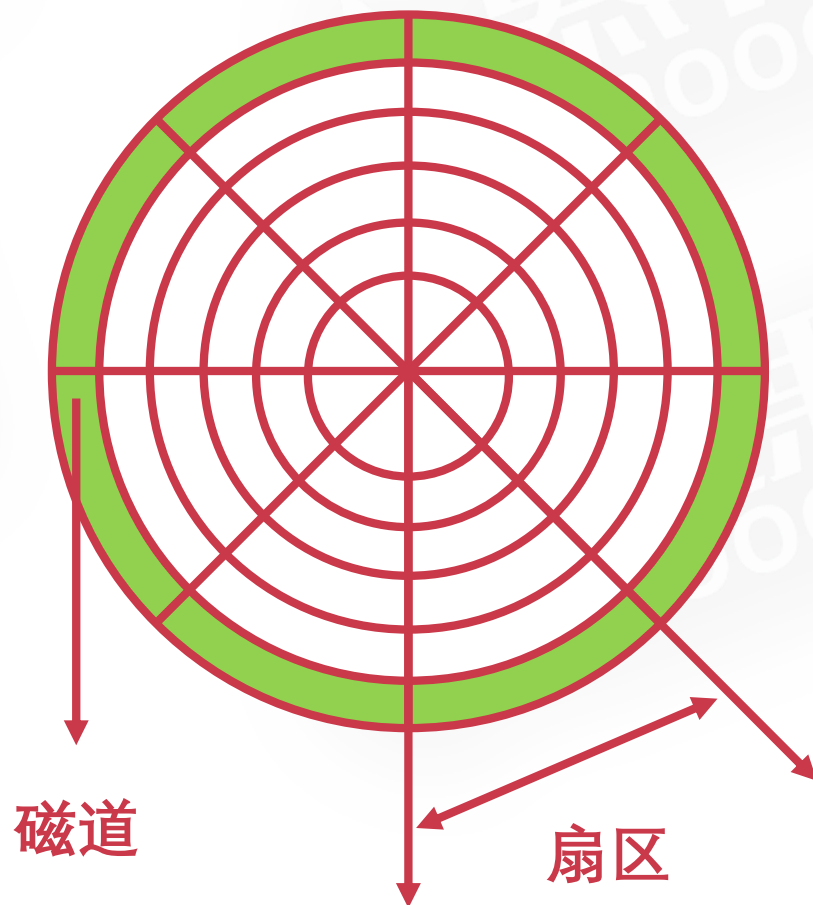


# 计算机的主存储器与辅助存储器

## 辅助存储器——磁盘

磁头位置

磁头方向



# 计算机的主存储器与辅助存储器

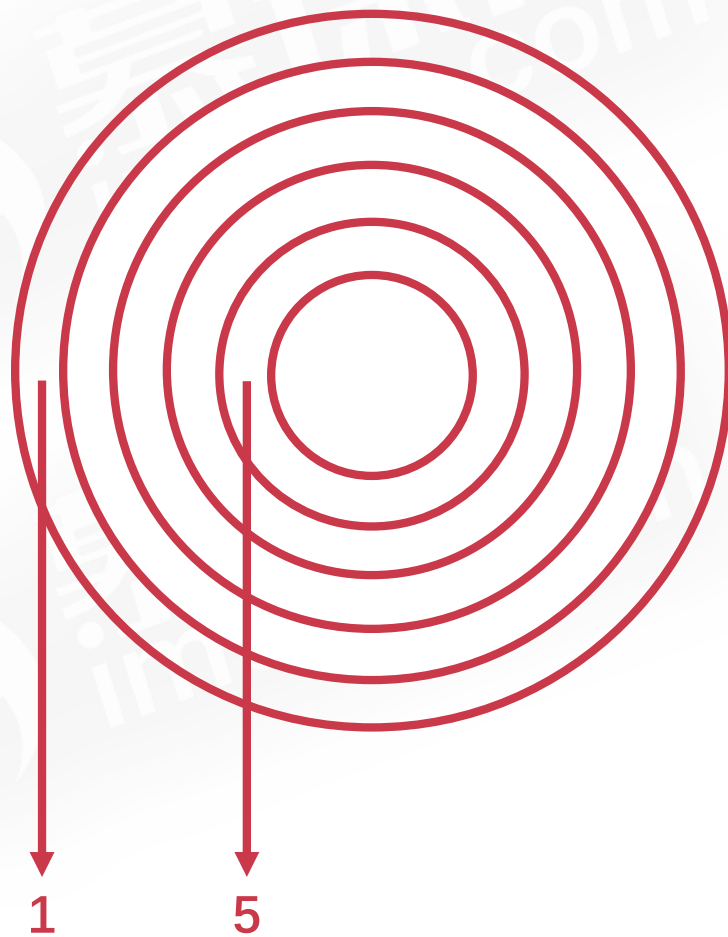
## 辅助存储器——磁盘

- ◆ 表面是可磁化的硬磁特性材料
- ◆ 移动磁头径向运动读取磁道信息

# 计算机的主存储器与辅助存储器

## 辅助存储器——磁盘

- ◆ 先来先服务算法
- ◆ 最短寻道时间优先
- ◆ 扫描算法（电梯算法）
- ◆ 循环扫描算法



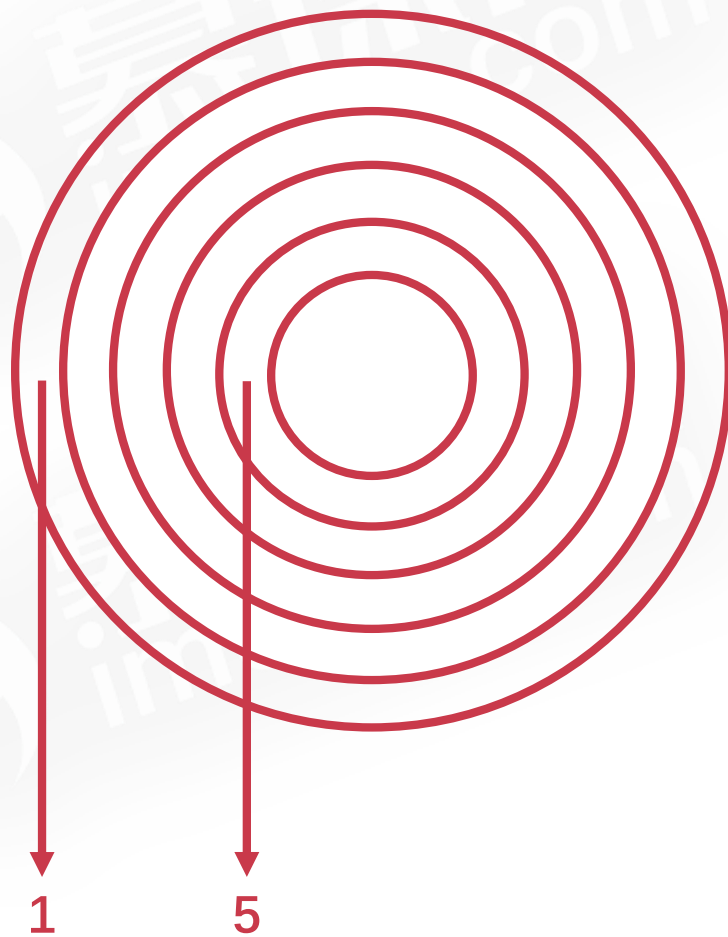
# 计算机的主存储器与辅助存储器

## 辅助存储器——磁盘

磁头在磁道4

磁头方向向外

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 先来先服务算法

- ◆ 按顺序访问进程的磁道读写需求

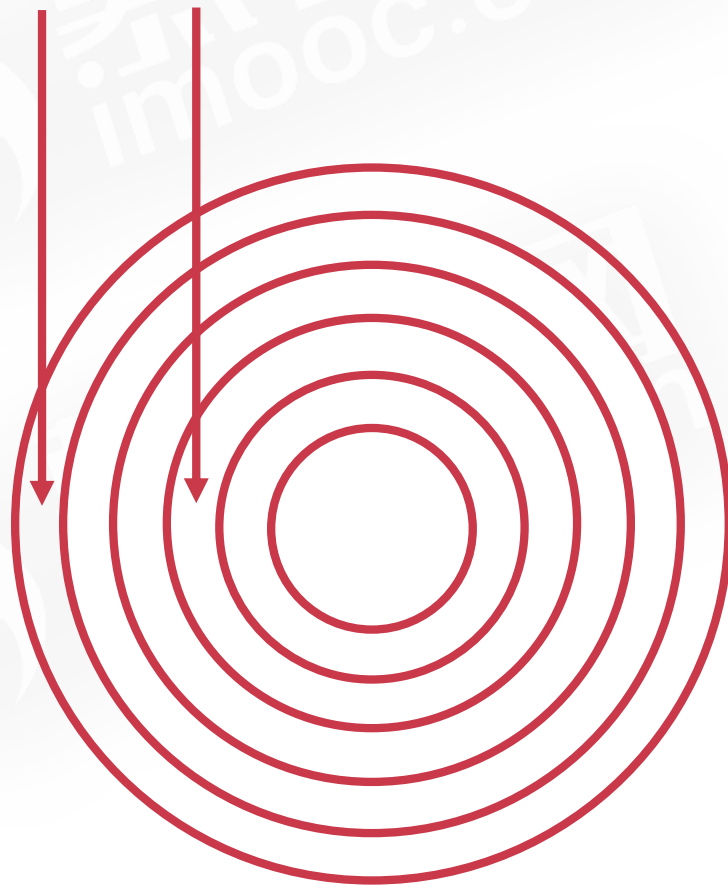


# 计算机的主存储器与辅助存储器

## 先来先服务算法

1 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 先来先服务算法

1 =》

4 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

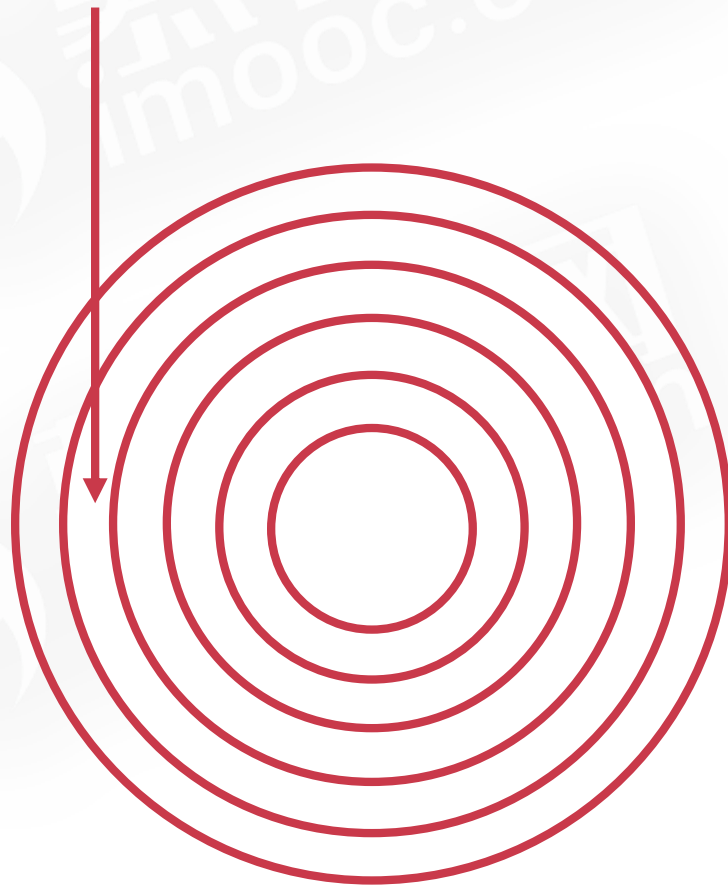
## 先来先服务算法

1 =》

4 =》

2 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 先来先服务算法

1 =》

4 =》

2 =》

3 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 先来先服务算法

1 =》

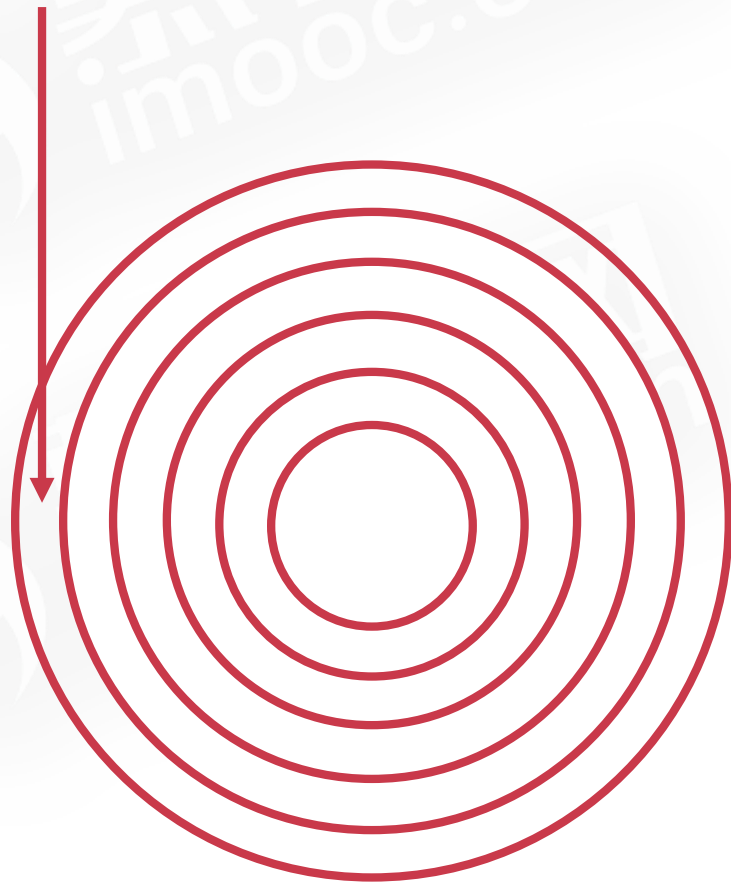
4 =》

2 =》

3 =》

1 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 先来先服务算法

1 =》

4 =》

2 =》

3 =》

1 =》

5 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 最短寻道时间优先算法

- ◆ 与磁头当前位置有关
- ◆ 优先访问离磁头最近的磁道

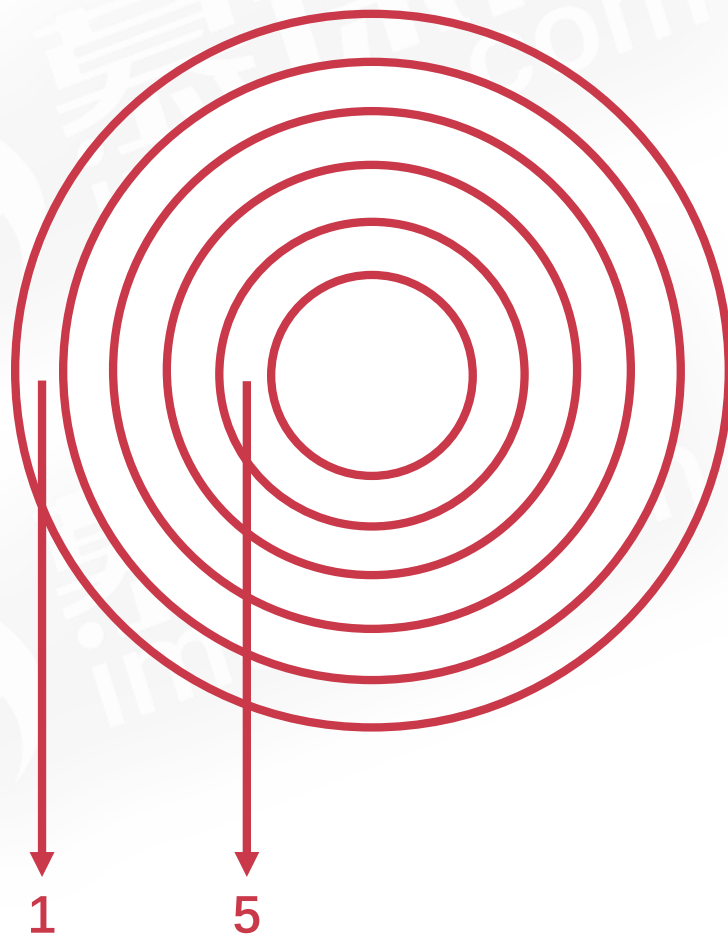
# 计算机的主存储器与辅助存储器

## 辅助存储器——磁盘

磁头在磁道4

磁头方向向外

现读取磁道：1 4 2 3 1 5





# 计算机的主存储器与辅助存储器

## 最短寻道时间优先算法

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 最短寻道时间优先算法

4 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 最短寻道时间优先算法

4 =》

5 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 最短寻道时间优先算法

4 =》

5 =》

3 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 最短寻道时间优先算法

4 =》

5 =》

3 =》

2 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 最短寻道时间优先算法

4 =》

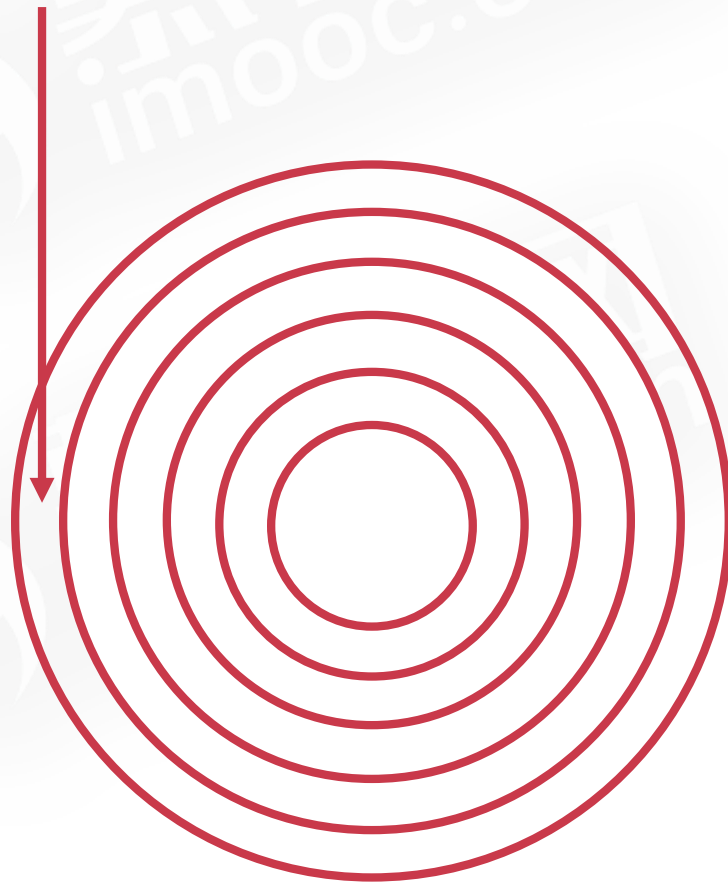
5 =》

3 =》

2 =》

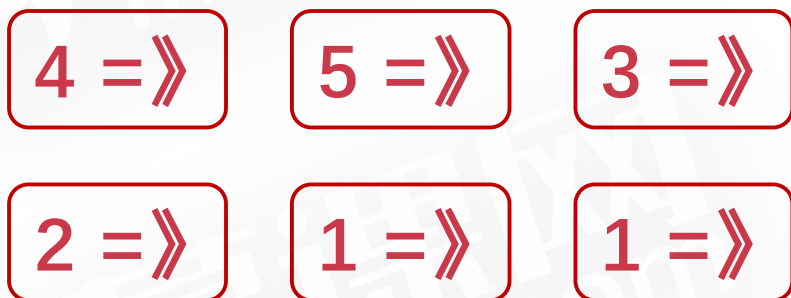
1 =》

现读取磁道：1 4 2 3 1 5

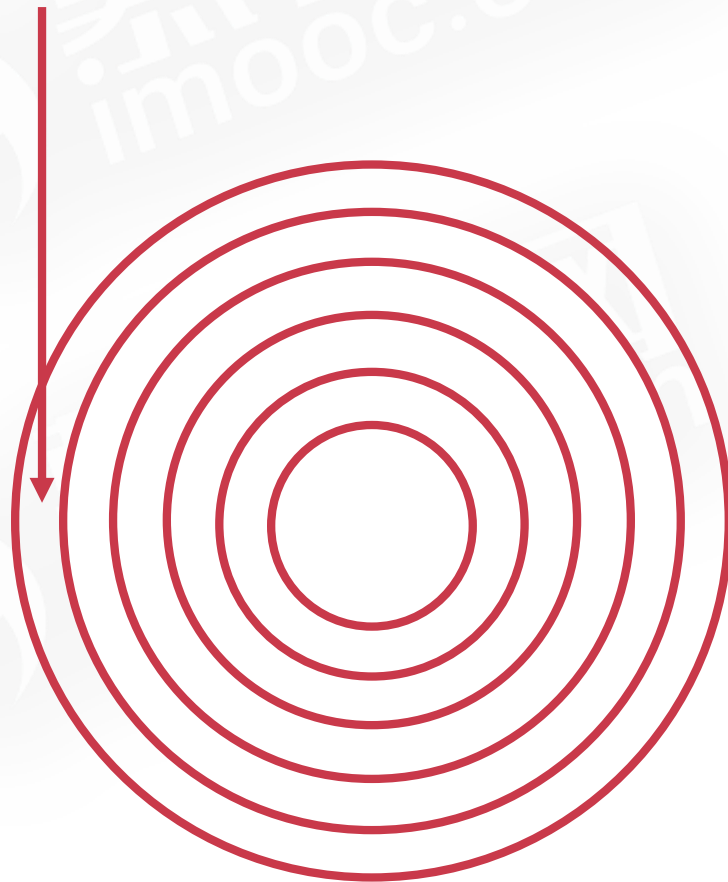


# 计算机的主存储器与辅助存储器

## 最短寻道时间优先算法



现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 扫描算法（电梯算法）

- ◆ 每次只往一个方向移动
- ◆ 到达一个方向需要服务的尽头再反方向移动



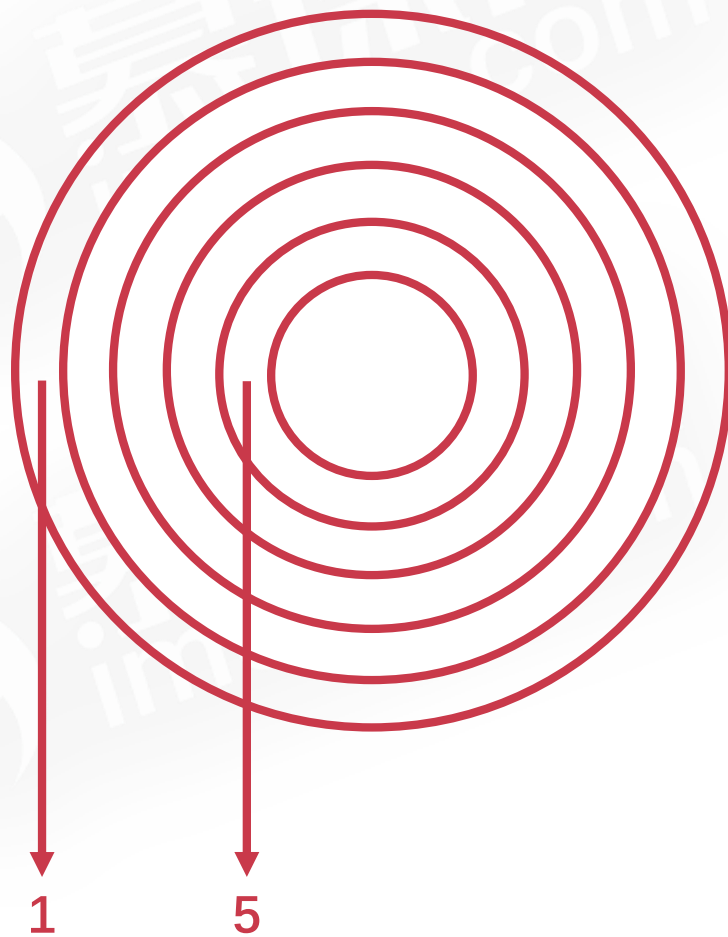
# 计算机的主存储器与辅助存储器

## 扫描算法（电梯算法）

磁头在磁道4

磁头方向向外

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 扫描算法（电梯算法）

4 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 扫描算法（电梯算法）

4 =》

3 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 扫描算法（电梯算法）

4 =》

3 =》

2 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 扫描算法（电梯算法）

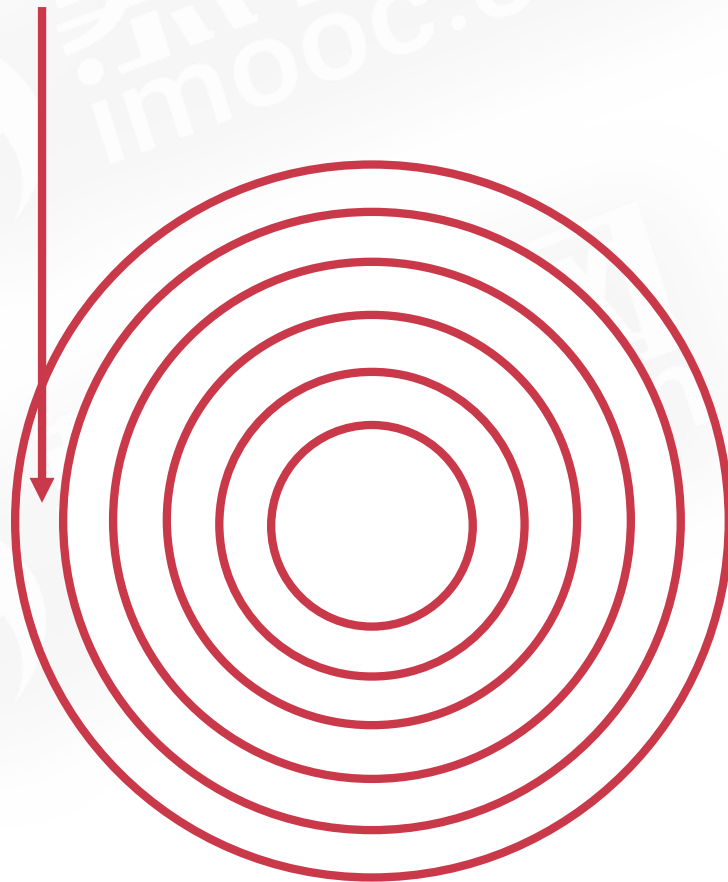
4 =》

3 =》

2 =》

1 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 扫描算法（电梯算法）

4 =》

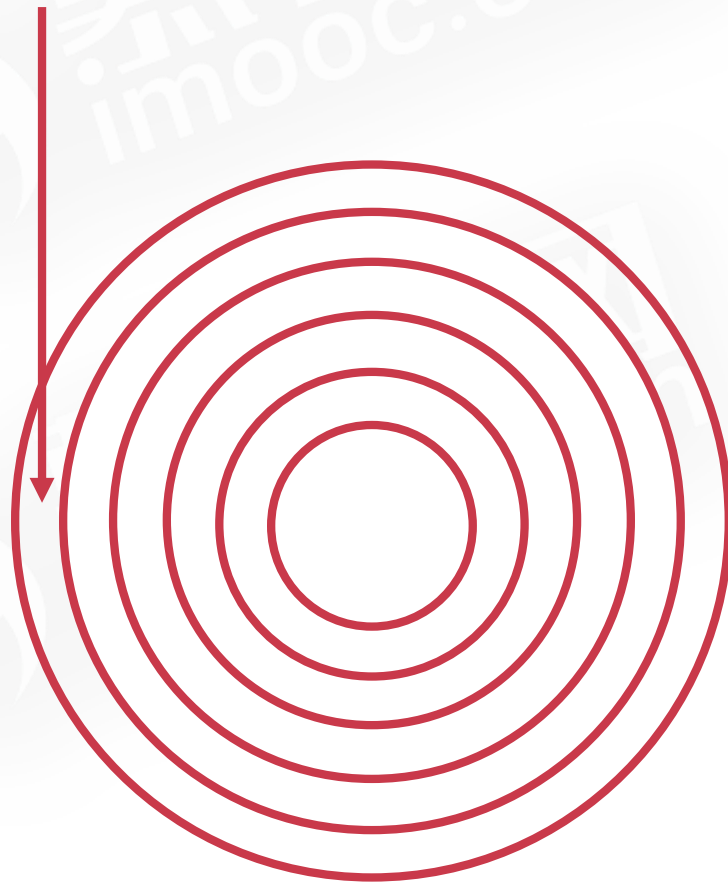
3 =》

2 =》

1 =》

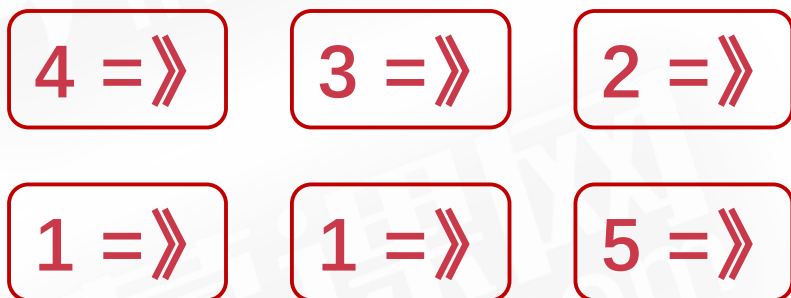
1 =》

现读取磁道：1 4 2 3 1 5



# 计算机的主存储器与辅助存储器

## 扫描算法（电梯算法）



现读取磁道：1 4 2 3 1 5

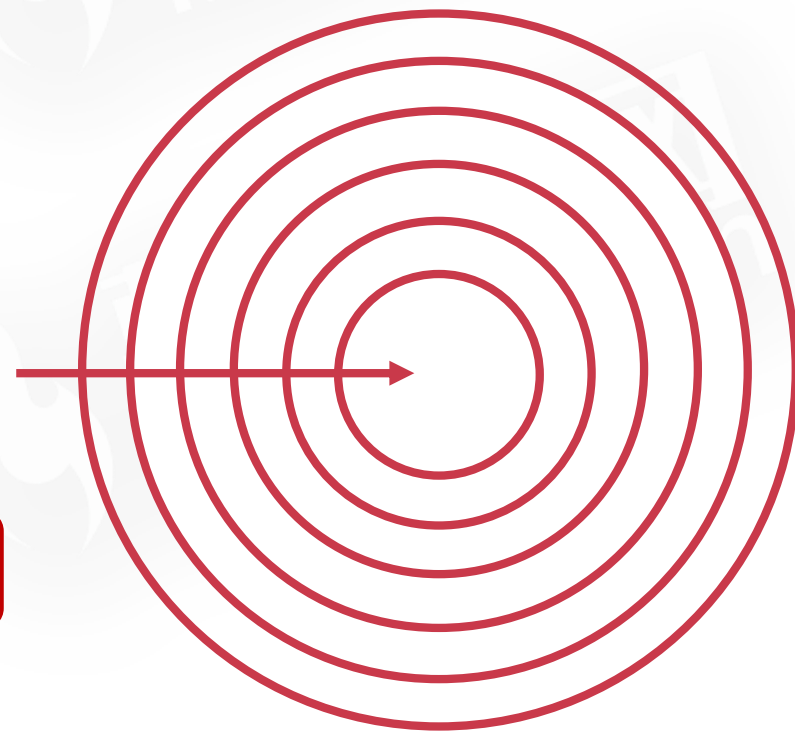


# 计算机的主存储器与辅助存储器

## 循环扫描算法



假设磁头在磁道4，现读取磁道：1 4 2 3 1 5



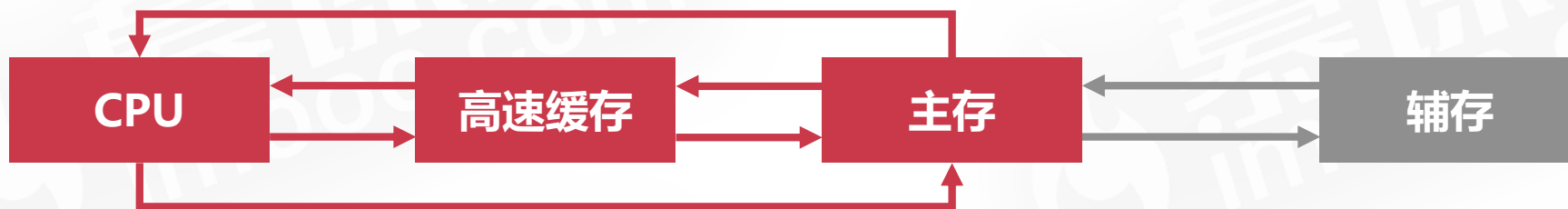


# 计算机的主存储器与辅助存储器

- ◆ 主存储器——内存
- ◆ 辅助存储器——磁盘



# 计算机的高速缓存



缓存-主存层次

CPU与主存的速度不匹配

存储器的层次结构



# 计算机的高速缓存

- ◆ 高速缓存的工作原理
- ◆ 高速缓存的替换策略

# 计算机的高速缓存

## 高速缓存的工作原理

**字：**

是指存放在一个存储单元中的二进制代码组合

**字块：**

存储在连续的存储单元中而被看作是一个单元的一组字

# 计算机的高速缓存



...



主存( $2^n$ 个字)

- ◆ 一个字有32位
- ◆ 一个字块共B个字
- ◆ 主存共M个字块

$$B * M = \text{主存总字数}$$

$$B * M * 32 = \text{主存总容量(bits)}$$

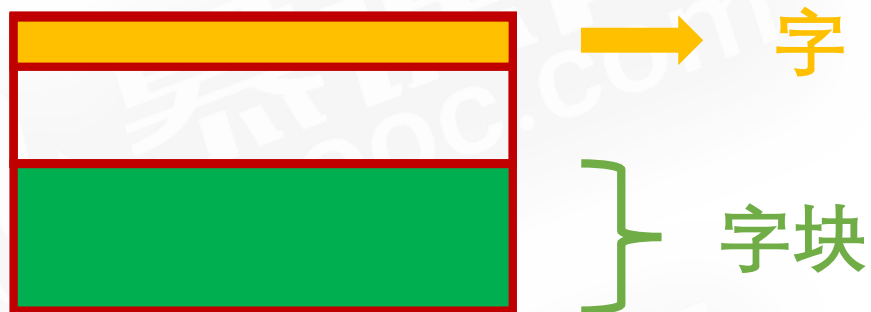
# 计算机的高速缓存



主存( $2^n$ 个字)

- ◆ 字的地址包含两个部分
- ◆ 前 $m$ 位指定字块的地址
- ◆ 后 $b$ 位指定字在字块中的地址

# 计算机的高速缓存



...



主存( $B \times M$ )

- ◆ 一个字有32位
- ◆ 一个字块共B个字
- ◆ 主存共M个字块



$$2^m = M \quad 2^b = B$$



# 计算机的高速缓存

例子：假设主存用户空间容量为4G，字块大小为4M，字长为32位，则对于字地址中的块地址m和块内地址b的位数，至少应该是多少？

$$4G = 4096M$$

字块数： $4096 \div 4 = 1024$

字块地址m： $\log_2 1024 = 10$

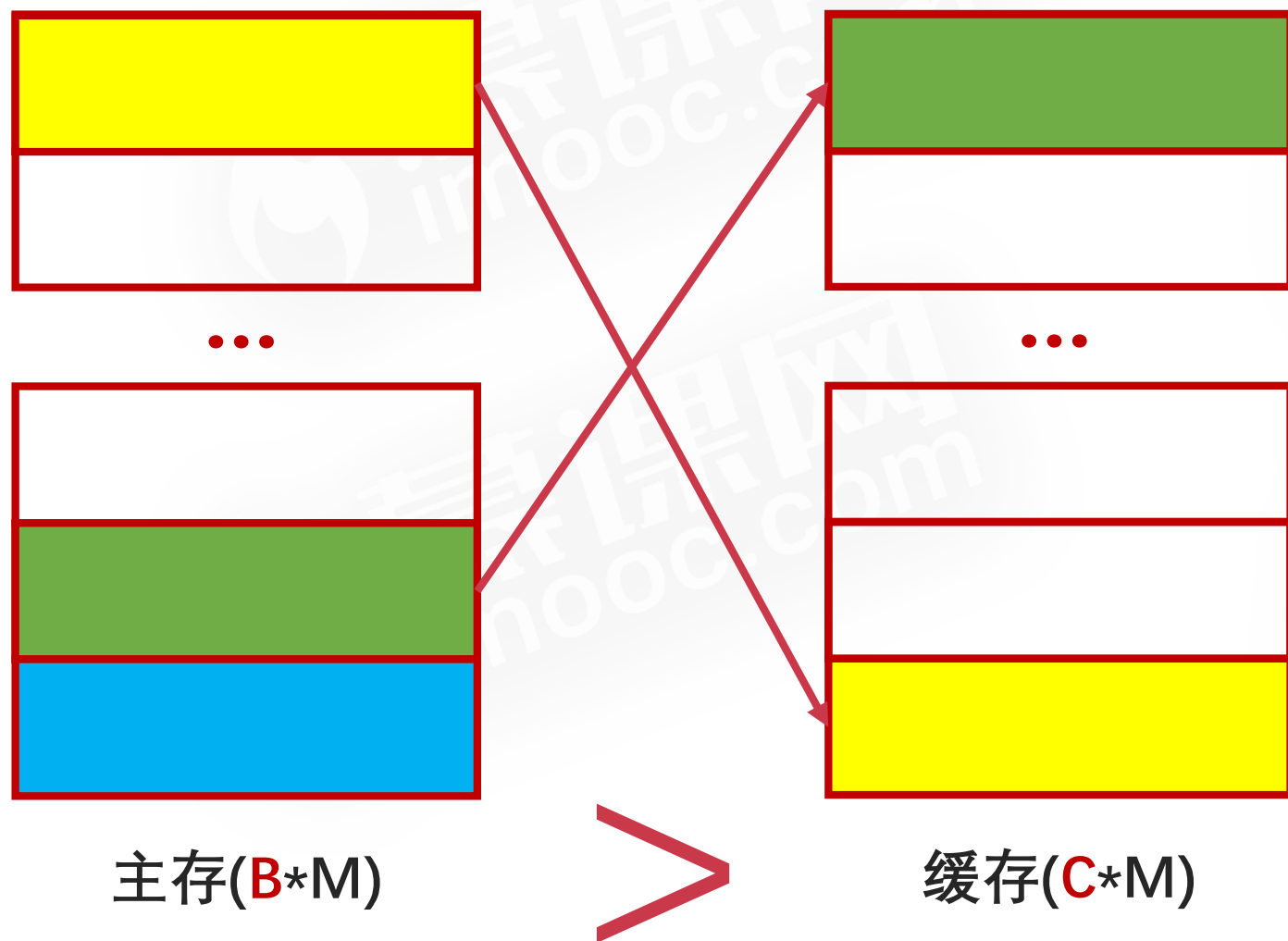
块内字数： $4M \div 32bit = 1048576$

块内地址b： $\log_2 1048576 = 20$

$$m \geq 10 \quad b \geq 20$$



# 计算机的高速缓存



◆ 存储的逻辑结构类似

◆ 缓存的容量较小

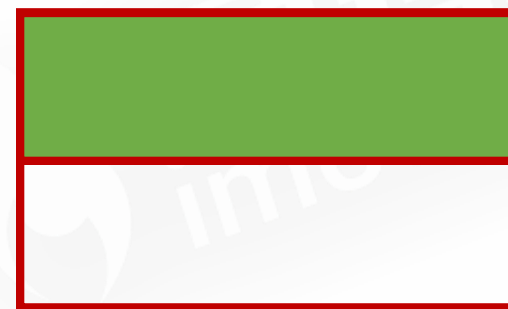
◆ 缓存的速度更快

# 计算机的高速缓存

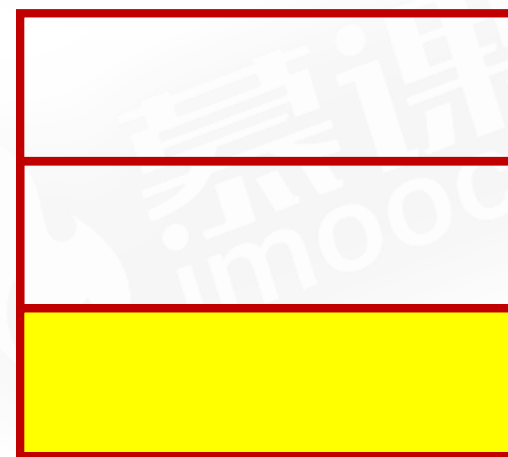
- ◆ 一个字有32位
- ◆ 一个字块共B个字
- ◆ 缓存共C个字块



$$2^c = C \quad 2^b = B$$



...



缓存(C\*M)

# 计算机的高速缓存



- ◆ CPU需要的数据在缓存里
- ◆ CPU需要的数据不在缓存里
- ◆ 不在缓存的数据需要去主存拿

命中率

# 计算机的高速缓存

## 高速缓存的工作原理

- ◆ 命中率是衡量缓存的重要性能指标
- ◆ 理论上CPU每次都能从高速缓存取数据的时候，命中率为1

访问主存次数： $N_m$

访问Cache次数： $N_c$

$$h = \frac{N_c}{N_c + N_m}$$

命中率

# 计算机的高速缓存

## 高速缓存的工作原理

访问效率:  $e$

$$h = \frac{N_c}{N_c + N_m}$$

访问主存时间:  $t_m$       访问缓存时间:  $t_c$

访问Cache-主存系统平均时间:  $t_a = ht_c + (1 - h)t_m$

$$e = \frac{t_c}{t_a} = \frac{t_c}{ht_c + (1 - h)t_m}$$

# 计算机的高速缓存

例子：假设CPU在执行某段程序时，共访问了Cache命中2000次，访问主存50次，已知Cache的存取时间为50ns，主存的存取时间为200ns，求Cache-主存系统的命中率、访问效率和平均访问时间。

$$h = \frac{N_c}{N_c + N_m} = \frac{2000}{2000 + 50} = 0.97$$



# 计算机的高速缓存

例子：假设CPU在执行某段程序时，共访问了Cache命中2000次，访问主存50次，已知Cache的存取时间为50ns，主存的存取时间为200ns，求Cache-主存系统的命中率、访问效率和平均访问时间。

$$e = \frac{t_c}{t_a} = \frac{t_c}{ht_c + (1 - h)t_m}$$
$$= \frac{50}{0.97 * 50 + (1 - 0.97)200} = 0.917 = 91.7\%$$





# 计算机的高速缓存

例子：假设CPU在执行某段程序时，共访问了Cache命中2000次，访问主存50次，已知Cache的存取时间为50ns，主存的存取时间为200ns，求Cache-主存系统的命中率、访问效率和平均访问时间。

$$0.97 * 50 + (1 - 0.97)200 = 54.5ns$$



# 计算机的高速缓存

## 高速缓存的工作原理

命中率: 
$$h = \frac{N_c}{N_c + N_m}$$

访问效率: 
$$e = \frac{t_c}{t_a} = \frac{t_c}{ht_c + (1 - h)t_m}$$

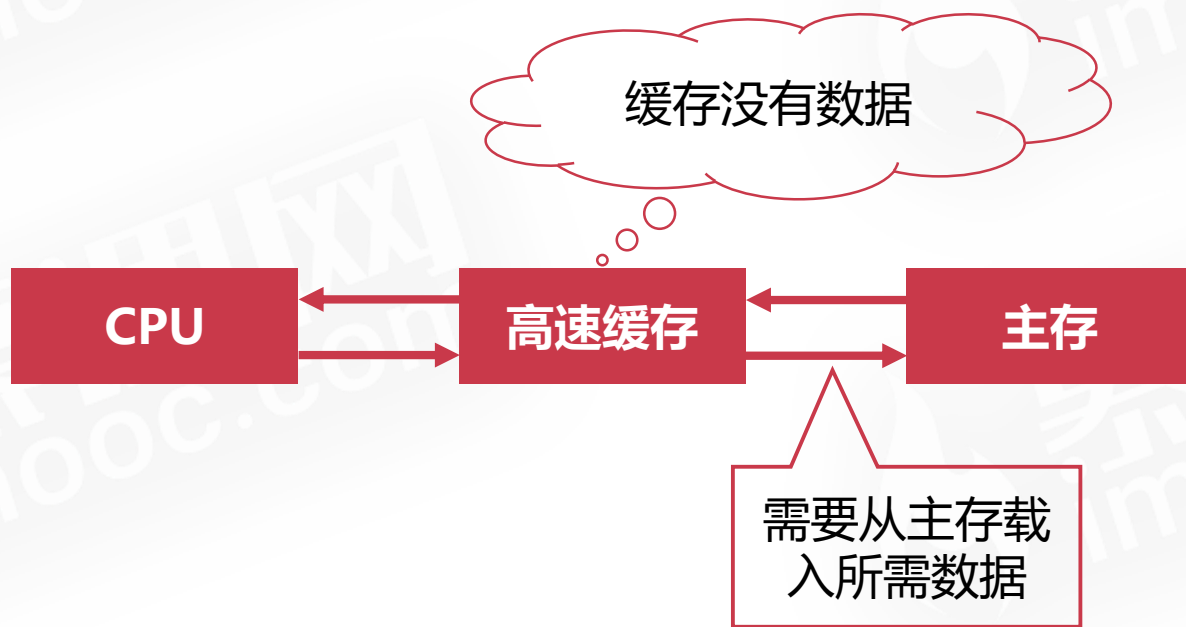
需要性能良好的缓存替换策略

# 计算机的高速缓存

- ◆ 高速缓存的工作原理
- ◆ 高速缓存的替换策略

# 计算机的高速缓存

## 高速缓存的替换策略



高速缓存的替换时机

# 计算机的高速缓存

## 高速缓存的替换策略

- ◆ 随机算法
- ◆ 先进先出算法(FIFO)
- ◆ 最不经常使用算法(LFU)
- ◆ 最近最少使用算法(LRU)

# 计算机的高速缓存

## 先进先出算法(FIFO)

- ◆ 把高速缓存看做是一个先进先出的队列
- ◆ 优先替换最先进入队列的字块

# 计算机的高速缓存



先进先出算法(FIFO)

# 计算机的高速缓存

## 最不经常使用算法(LFU)

- ◆ 优先淘汰最不经常使用的字块
- ◆ 需要额外的空间记录字块的使用频率



# 计算机的高速缓存

缓存	1	2	3	4	5	6	7	8
频率	0	0	0	0	0	0	0	0



访问: 2

缓存	1	2	3	4	5	6	7	8
频率	0	1	0	0	0	0	0	0



访问: 6

缓存	1	2	3	4	5	6	7	8
频率	0	1	0	0	0	1	0	0

最不经常使用算法(LFU)

# 计算机的高速缓存

缓存	1	2	3	4	5	6	7	8
频率	7	4	2	1	8	4	2	6



缓存	1	2	3	4	5	6	7	8
频率	7	4	2	1	8	4	2	6

4		缓存	1	2	3	9	5	6	7	8
		频率	7	4	2	1	8	4	2	6

最不经常使用算法(LFU)

# 计算机的高速缓存

## 最近最少使用算法 (LRU)

- ◆ 优先淘汰一段时间内没有使用的字块
- ◆ 有多种实现方法，一般使用双向链表
- ◆ 把当前访问节点置于链表前面（保证链表头部节点是最近使用的）

# 计算机的高速缓存

## 最近最少使用算法 (LRU)

(1) 1

(2) 2、1

(4) 4、2、1

(7) 7、4、2、1

(5) 5、7、4、2 [1]

(4) 4、5、7、2

(6) 6、4、5、7 [2]

(1) 1、6、4、5 [7]

(6) 6、1、4、5

(7) 7、6、1、4 [5]

(4) 4、7、6、1

(1) 1、4、7、6

假设缓存4个字块，（）表示使用的字块，[]表示淘汰的字块

# 计算机的高速缓存

- ◆ 高速缓存的工作原理
- ◆ 高速缓存的替换策略

慕课网  
imooc.com

慕课网  
imooc.com

慕课网  
imooc.com

慕课网  
imooc.com

# 计算机的指令系统

- ◆ 机器指令的形式
- ◆ 机器指令的操作类型
- ◆ 机器指令的寻址方式

# 计算机的指令系统

## 机器指令的形式

- ◆ 机器指令主要由两部分组成：操作码、地址码

操作码字段

地址码字段



# 计算机的指令系统

## 机器指令的形式

操作码字段

地址码字段

- ◆ 操作码指明指令所要完成的操作
- ◆ 操作码的位数反映了机器的操作种类

$$2^8 = 256$$

# 计算机的指令系统

## 机器指令的形式

操作码字段

地址码字段

- ◆ 地址码直接给出操作数或者操作数的地址
- ◆ 分三地址指令、二地址指令和一地址指令

# 计算机的指令系统

## 机器指令的形式

操作码(OP)

addr1

addr2

addr3

$(addr1)OP(addr2) \rightarrow (addr3)$

三地址指令

# 计算机的指令系统

## 机器指令的形式

操作码(OP)

addr1

addr2

$(addr1)OP(addr2) \rightarrow (addr1)或(addr2)$

二地址指令

# 计算机的指令系统

## 机器指令的形式

操作码(OP)

addr1

$(addr1)OP \rightarrow (addr1)$

$(addr1)OP(ACC) \rightarrow (addr1)$

一地址指令

# 计算机的指令系统

## 机器指令的形式

- ◆ 在机器指令中无地址码
- ◆ 空操作、停机操作、中断返回操作等

零地址指令

# 计算机的指令系统

- ◆ 机器指令的形式
- ◆ 机器指令的操作类型

# 计算机的指令系统

## 机器指令的操作类型

- ◆ 寄存器之间、寄存器与存储单元、存储单元之间传送
- ◆ 数据读写、交换地址数据、清零置一等操作

数据传输



# 计算机的指令系统

## 机器指令的操作类型

- ◆ 操作数之间的加减乘除运算
- ◆ 操作数的与或非等逻辑位运算

算术逻辑操作

# 计算机的指令系统

## 机器指令的操作类型

- ◆ 数据左移（乘2）、数据右移（除2）
- ◆ 完成数据在算术逻辑单元的必要操作

移位操作

# 计算机的指令系统

## 机器指令的操作类型

- ◆ 等待指令、停机指令、空操作指令、中断指令等

控制指令

# 计算机的指令系统

## 机器指令的操作类型

移位操作

数据传输

控制指令

算术逻辑操作

# 计算机的指令系统

- ◆ 机器指令的形式
- ◆ 机器指令的操作类型
- ◆ 机器指令的寻址方式

# 计算机的指令系统

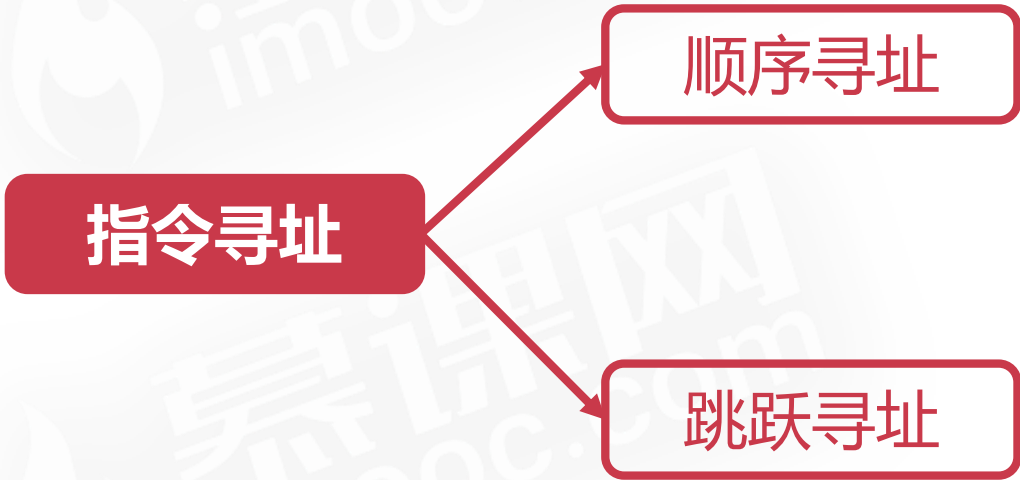
## 机器指令的寻址方式

指令寻址

数据寻址

# 计算机的指令系统

## 机器指令的寻址方式

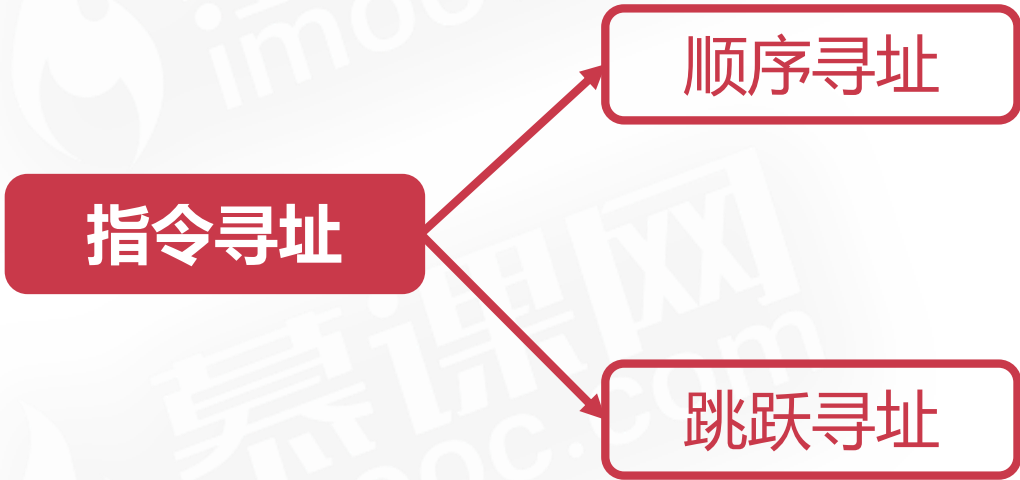


地址	指令
101	MOV R0,R1
102	LAD R1,6
103	ADD R1,R2
104	AND R1, R3
105	JMP 102



# 计算机的指令系统

## 机器指令的寻址方式



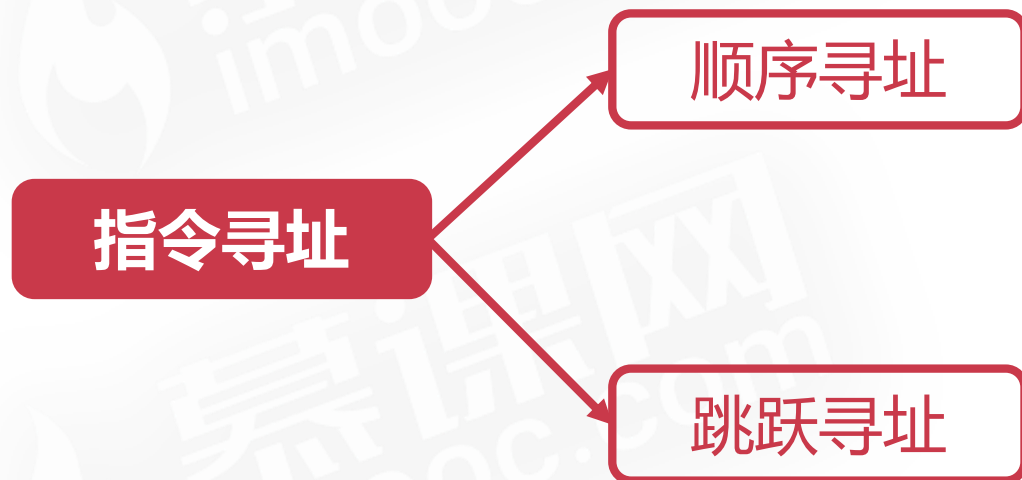
地址	指令
101	MOV R0,R1
102	LAD R1,6
103	ADD R1,R2
104	AND R1, R3
105	JMP 102





# 计算机的指令系统

## 机器指令的寻址方式

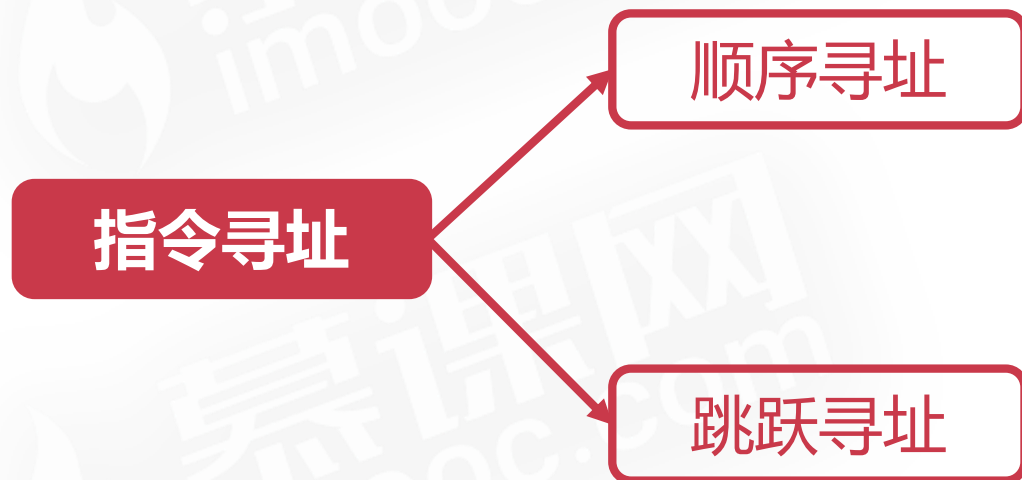


地址	指令
101	MOV R0,R1
102	LAD R1,6
103	ADD R1,R2
104	AND R1, R3
105	JMP 102



# 计算机的指令系统

## 机器指令的寻址方式

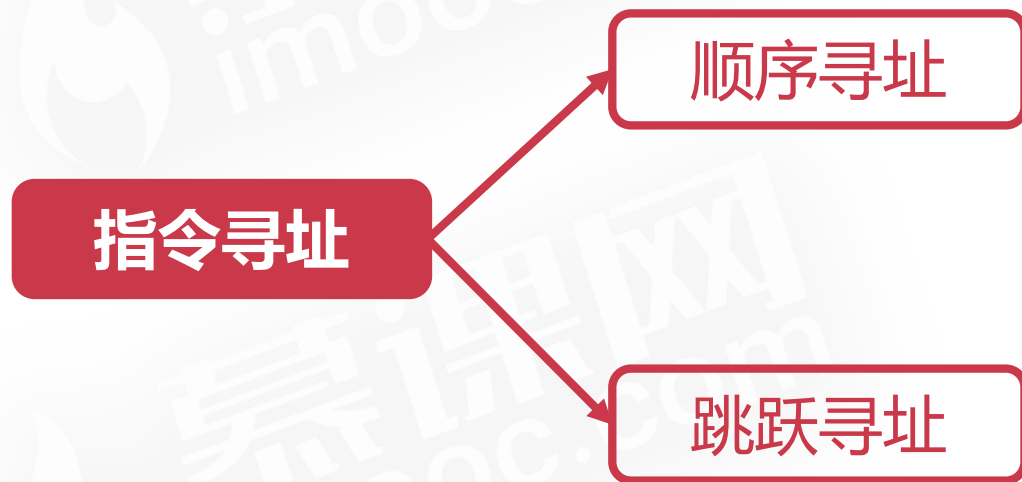


地址	指令
101	MOV R0,R1
102	LAD R1,6
103	ADD R1,R2
104	AND R1, R3
105	JMP 102



# 计算机的指令系统

## 机器指令的寻址方式

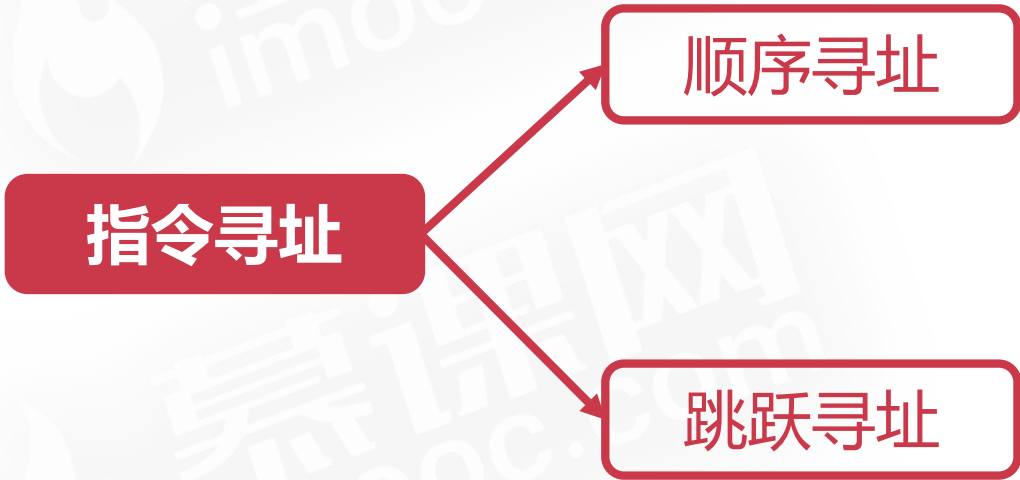


地址	指令
101	MOV R0,R1
102	LAD R1,6
103	ADD R1,R2
104	AND R1, R3
105	JMP 102



# 计算机的指令系统

## 机器指令的寻址方式

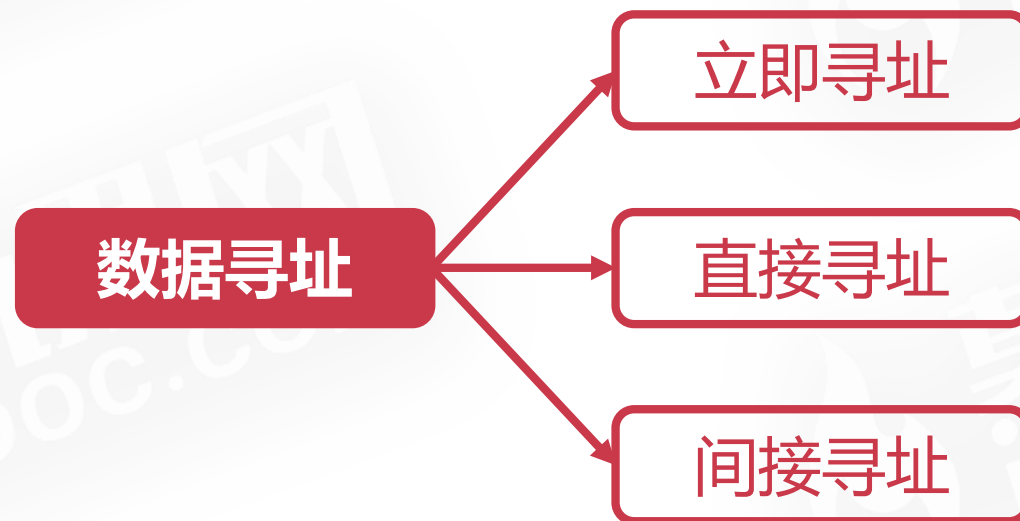


地址	指令
101	MOV R0,R1
102	LAD R1,6
103	ADD R1,R2
104	AND R1, R3
105	JMP 102



# 计算机的指令系统

## 机器指令的寻址方式



# 计算机的指令系统

## 机器指令的寻址方式

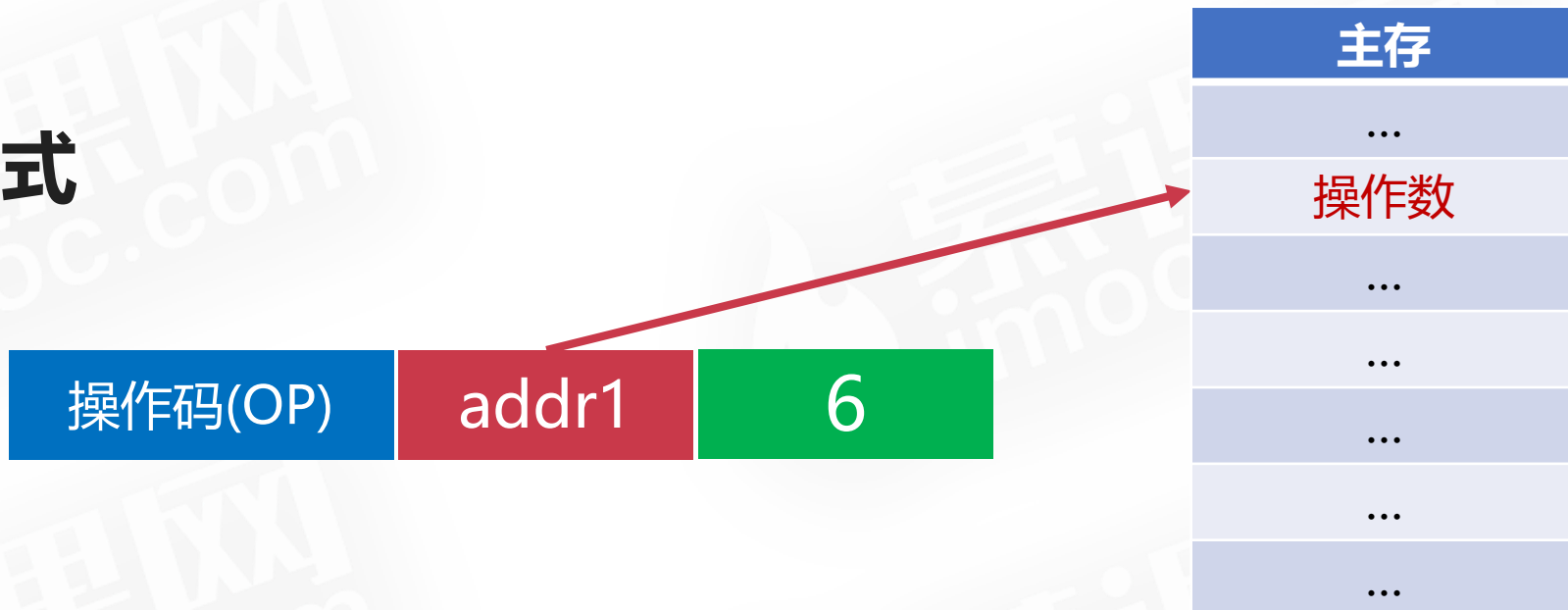


- ◆ 指令直接获得操作数
- ◆ 无需访问存储器

立即寻址

# 计算机的指令系统

## 机器指令的寻址方式



- ◆ 直接给出操作数在主存的地址
- ◆ 寻找操作数简单，无需计算数据地址

直接寻址

# 计算机的指令系统

## 机器指令的寻址方式



- ◆ 指令地址码给出的是操作数地址的地址
- ◆ 需要访问一次或多次主存来获取操作数

间接寻址



# 计算机的指令系统

## 机器指令的寻址方式

寻址方式	优点	缺点
立即寻址	速度快	地址码位数限制操作数表示范围
直接寻址	寻找操作数简单	地址码位数限制操作数寻址范围
间接寻址	操作数寻址范围大	速度较慢

# 计算机的指令系统

- ◆ 机器指令的形式
- ◆ 机器指令的操作类型
- ◆ 机器指令的寻址方式

慕课网  
imooc.com

慕课网  
imooc.com

慕课网  
imooc.com

慕课网  
imooc.com

# 计算机的控制器



控制器是协调和控制计算机运行的

# 计算机的控制器

## 程序计数器

- ◆ 程序计数器用来存储下一条指令的地址
- ◆ 循环从程序计数器中拿出指令
- ◆ 当指令被拿出时，指向下一条指令

# 计算机的控制器

## 时序发生器

- ◆ 电气工程领域，用于发送时序脉冲
- ◆ CPU依据不同的时序脉冲有节奏的进行工作

# 计算机的控制器

## 指令译码器

- ◆ 指令译码器是控制器的主要部件之一
- ◆ 计算机指令由操作码和地址码组成
- ◆ 翻译操作码对应的操作以及控制传输地址码对应的数据

# 计算机的控制器

## 指令寄存器

- ◆ 指令寄存器也是控制器的主要部件之一
- ◆ 从主存或高速缓存取计算机指令



# 计算机的控制器

## 主存地址寄存器

- ◆ 保存当前CPU正要访问的内存单元的地址

# 计算机的控制器

## 主存数据寄存器

- ◆ 保存当前CPU正要读或写的主存数据

# 计算机的控制器

## 通用寄存器

- ◆ 用于暂时存放或传送数据或指令
- ◆ 可保存ALU的运算中间结果
- ◆ 容量比一般专用寄存器要大

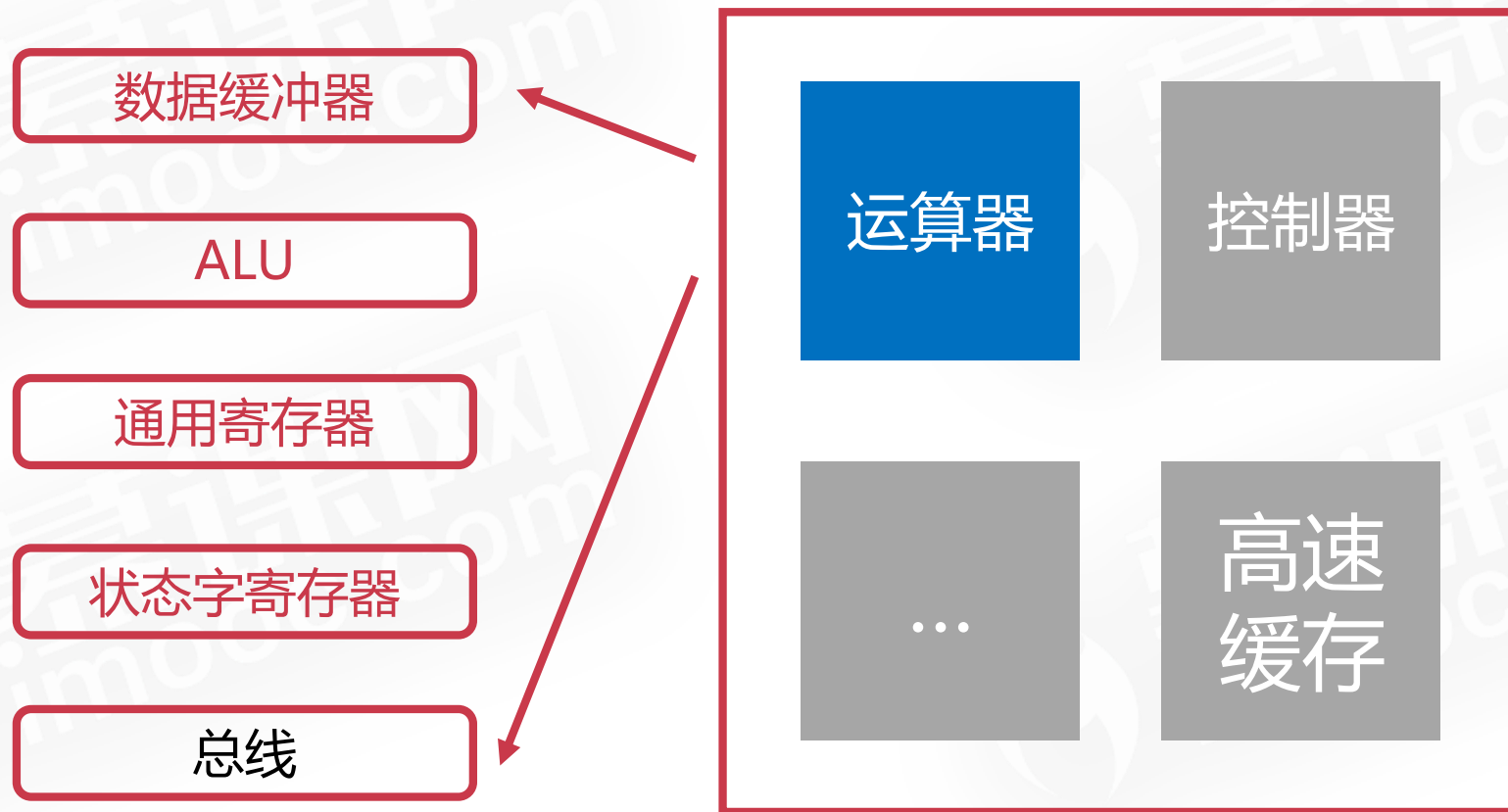
# 计算机的控制器



控制器是协调和控制计算机运行的



# 计算机的运算器



运算器是用来进行数据运算加工的

# 计算机的运算器

## 数据缓冲器

- ◆ 分为输入缓冲和输出缓冲
- ◆ 输入缓冲暂时存放外设送过来的数据
- ◆ 输出缓冲暂时存放送往外设的数据

# 计算机的运算器

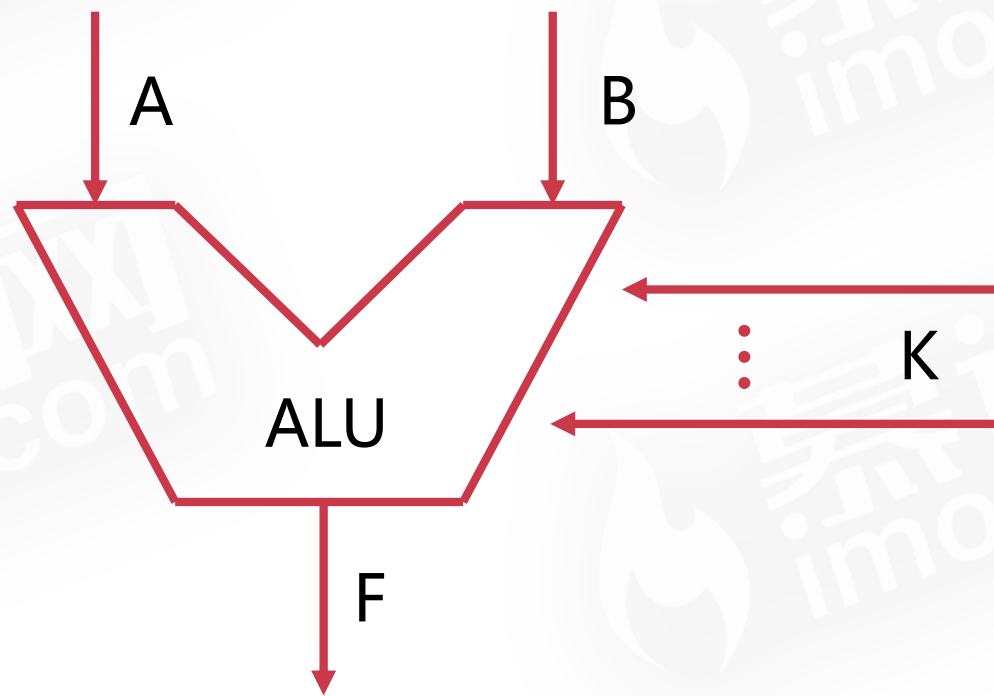
## ALU

- ◆ ALU：算术逻辑单元，是运算器的主要组成
- ◆ 常见的位运算（左右移、与或非等）
- ◆ 算术运算（加减乘除等）



# 计算机的运算器

ALU



# 计算机的运算器

## 状态字寄存器

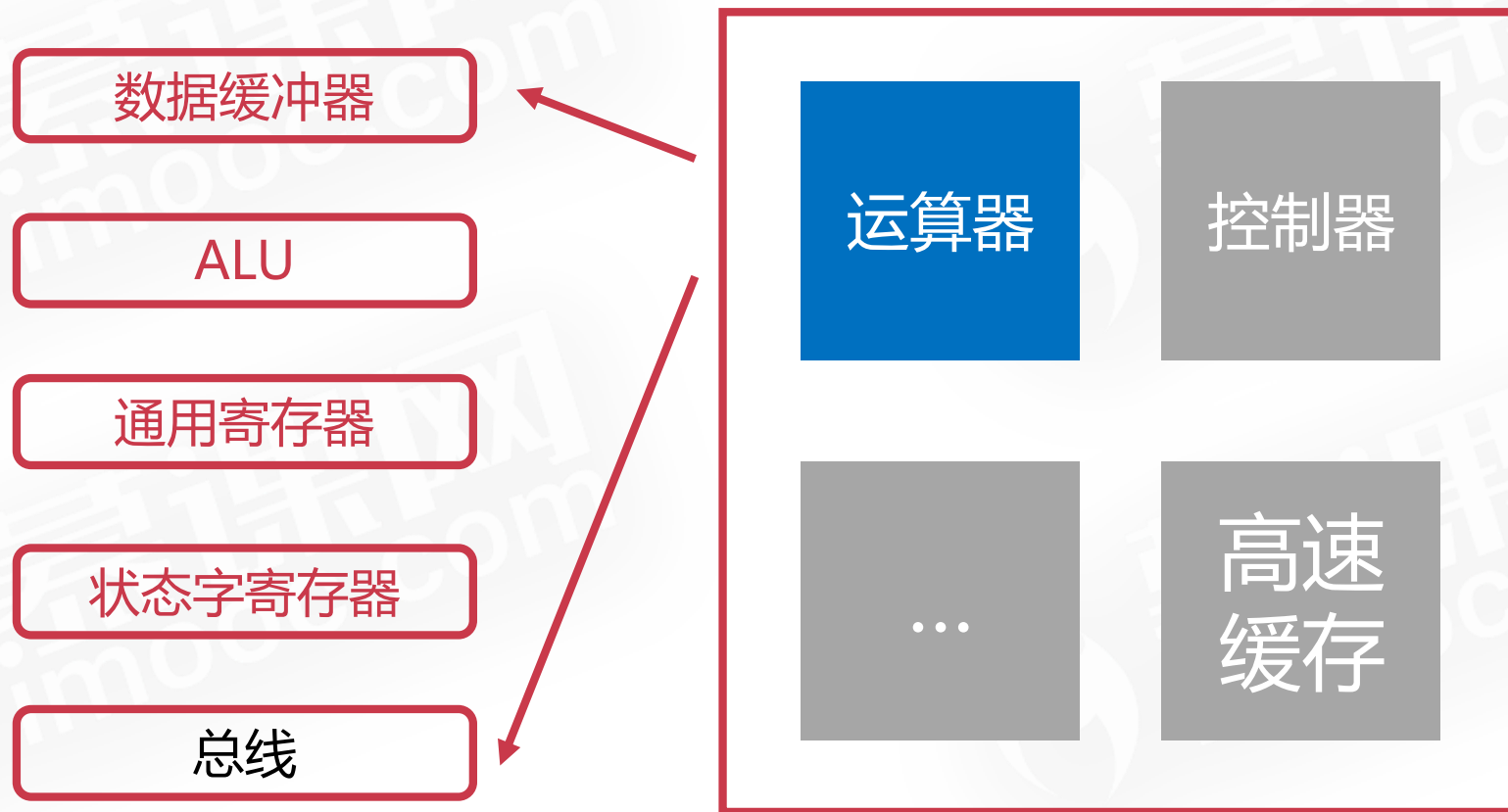
- ◆ 存放运算状态（条件码、进位、溢出、结果正负等）
- ◆ 存放运算控制信息（调试跟踪标记位、允许中断位等）

# 计算机的运算器

## 通用寄存器

- ◆ 用于暂时存放或传送数据或指令
- ◆ 可保存ALU的运算中间结果
- ◆ 容量比一般专用寄存器要大

# 计算机的运算器



运算器是用来进行数据运算加工的



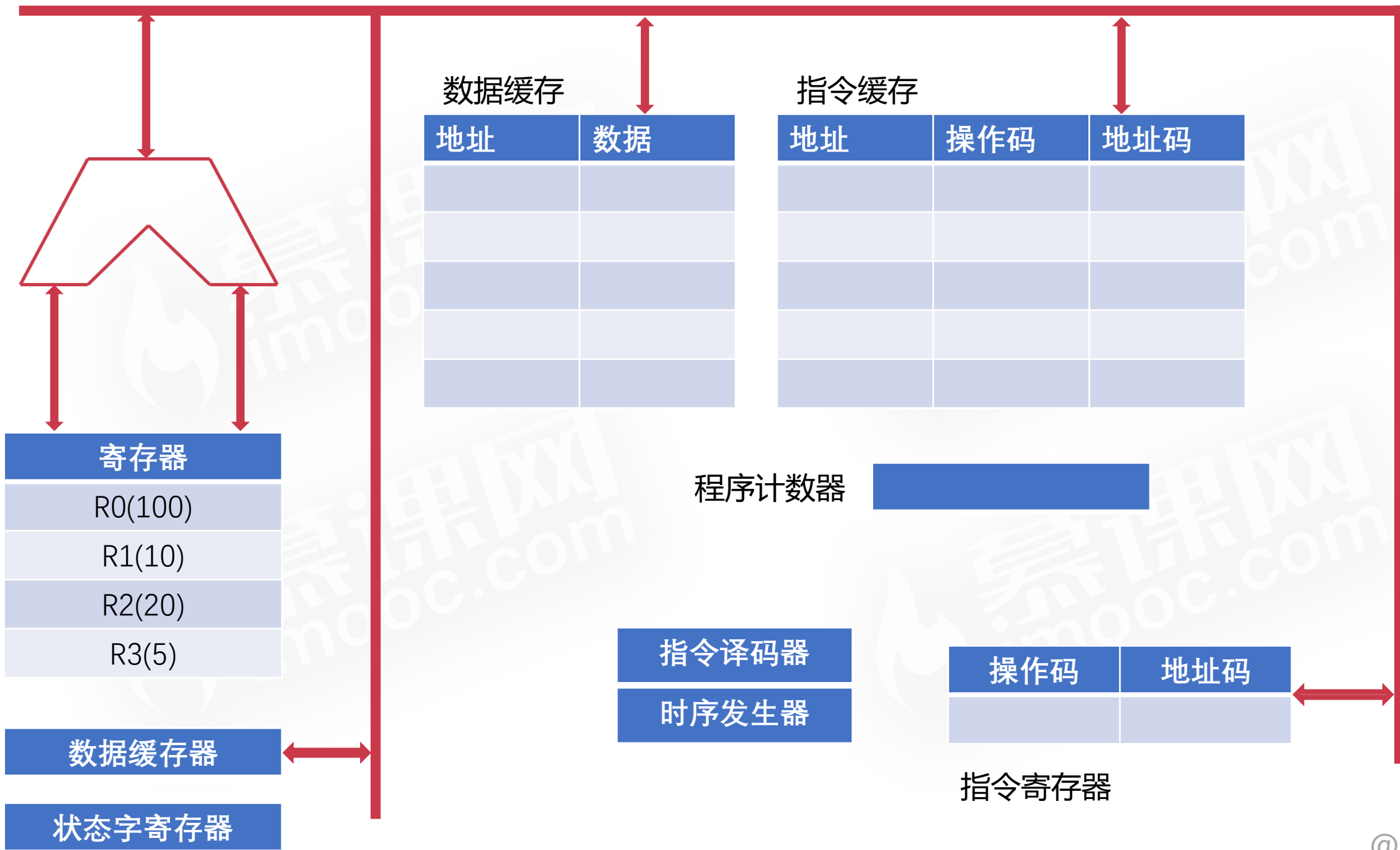
# 计算机指令的执行过程

- ◆ 指令执行过程
- ◆ CPU的流水线设计

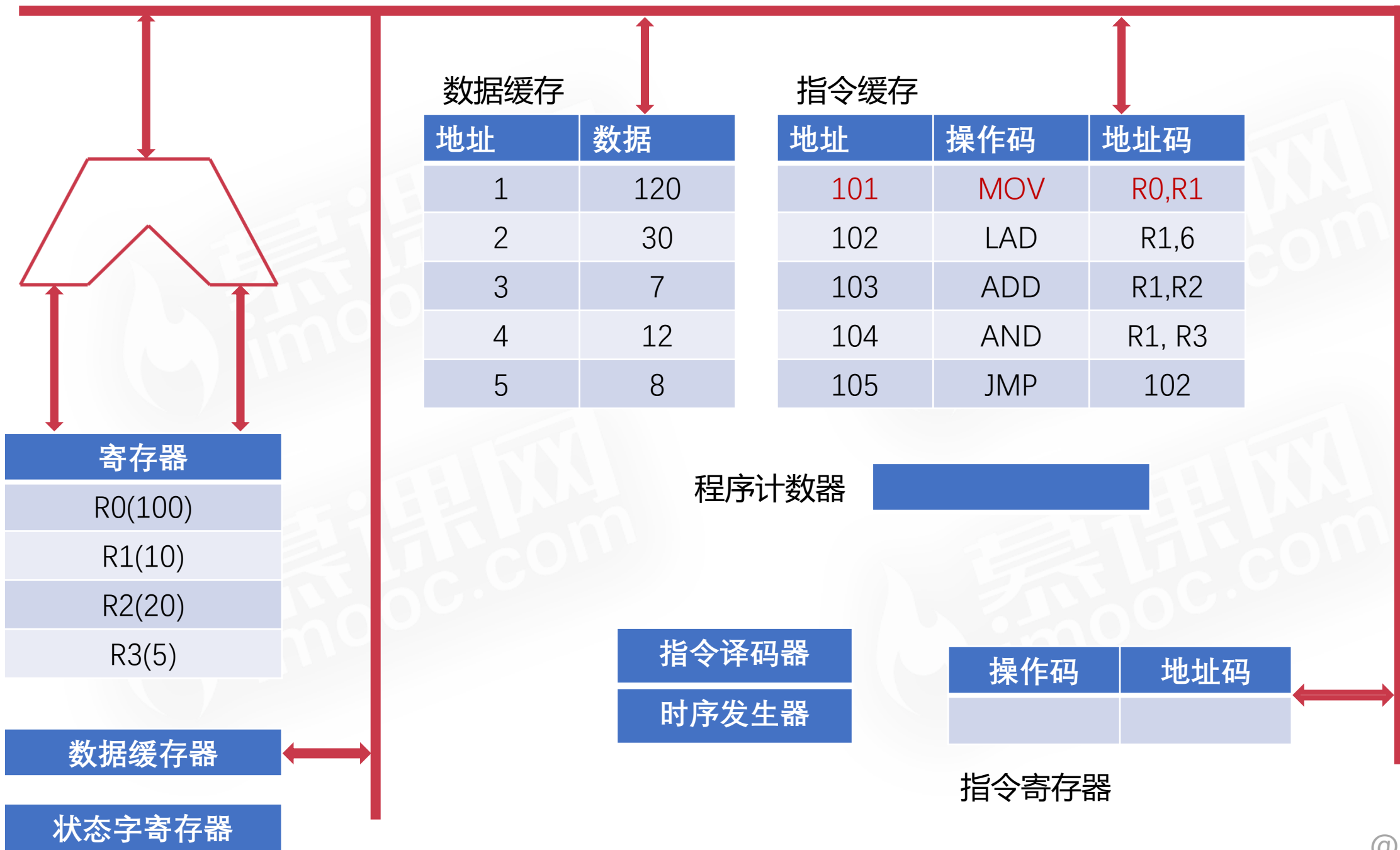
# 计算机指令的执行过程

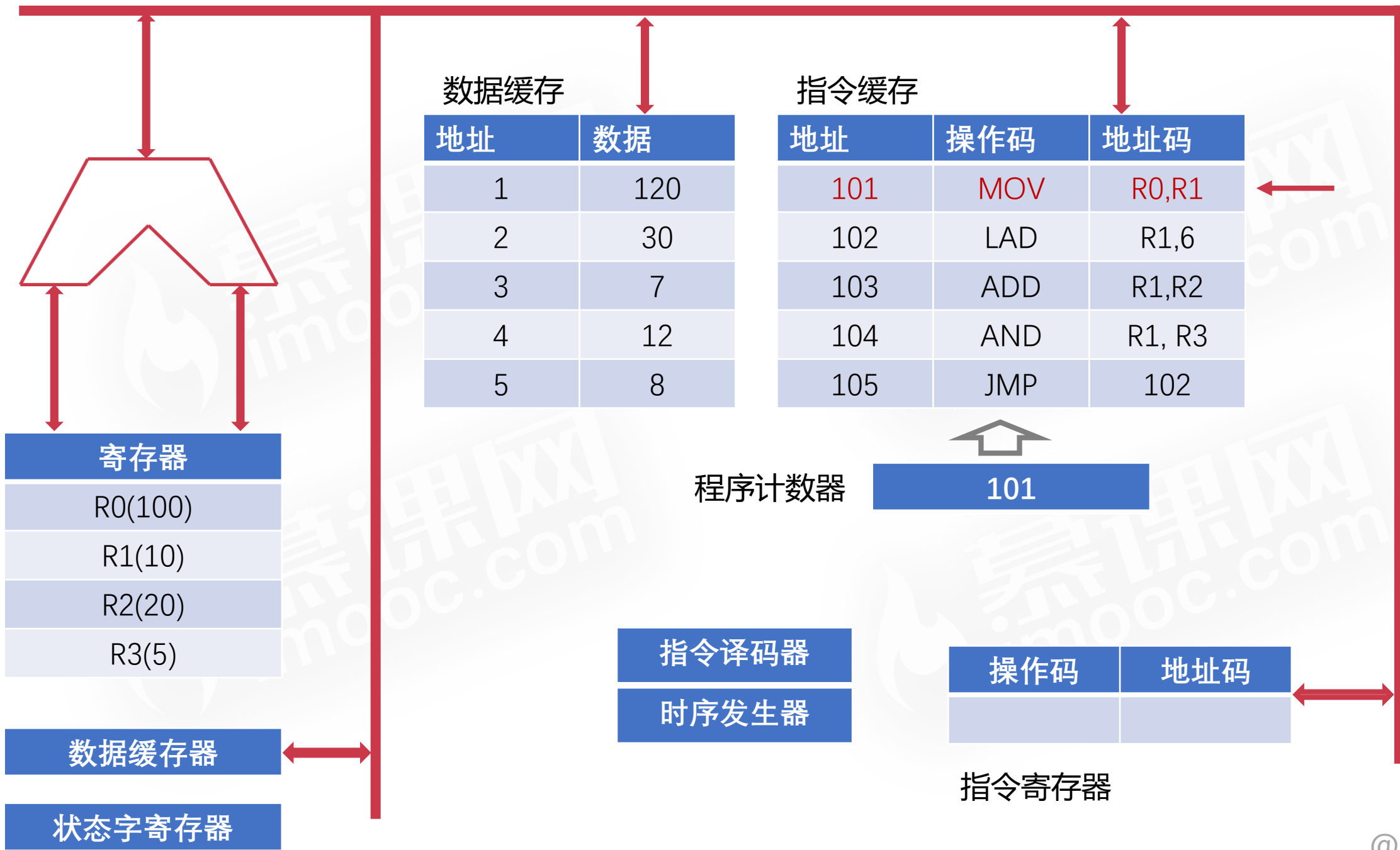
## 指令执行过程

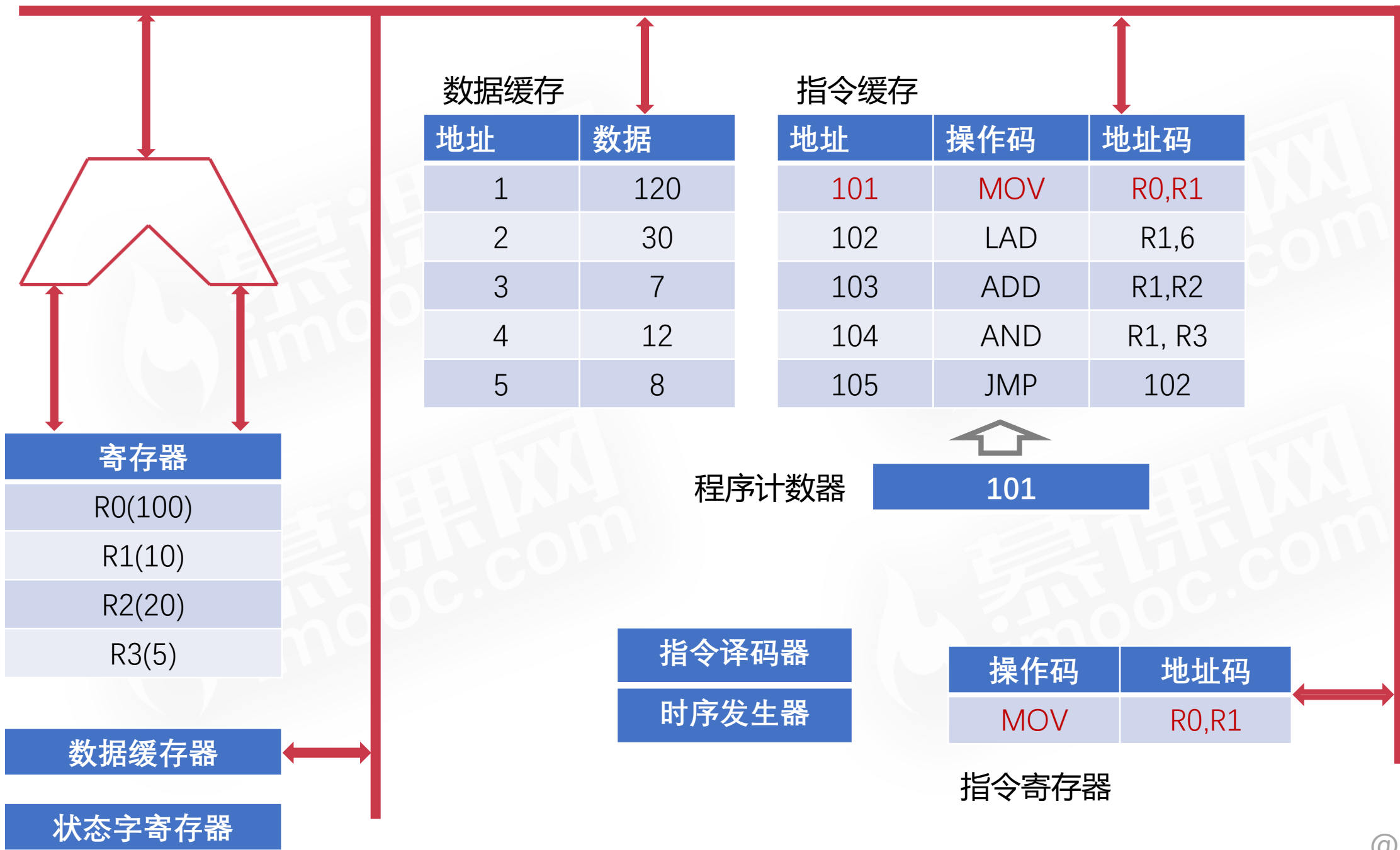


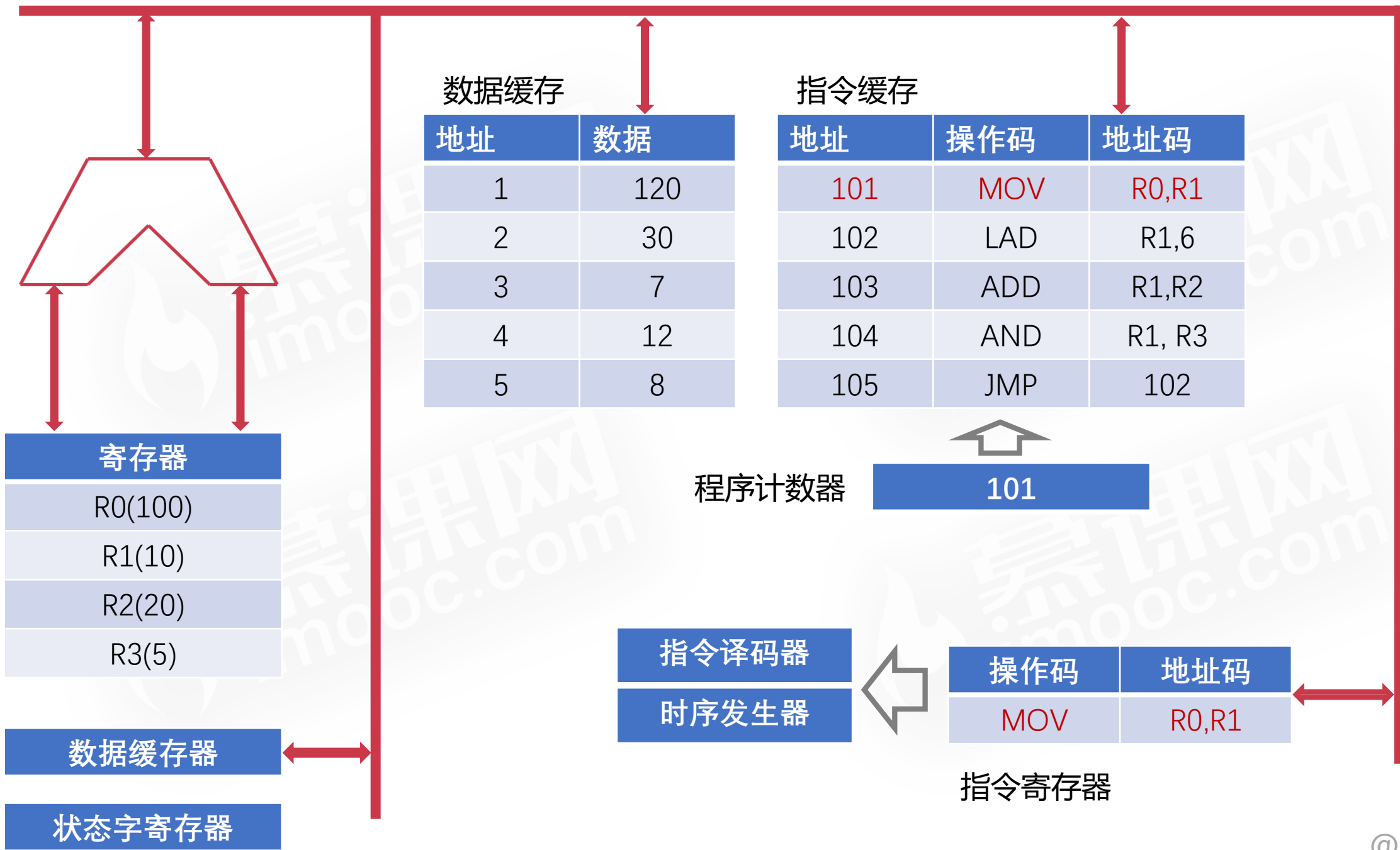


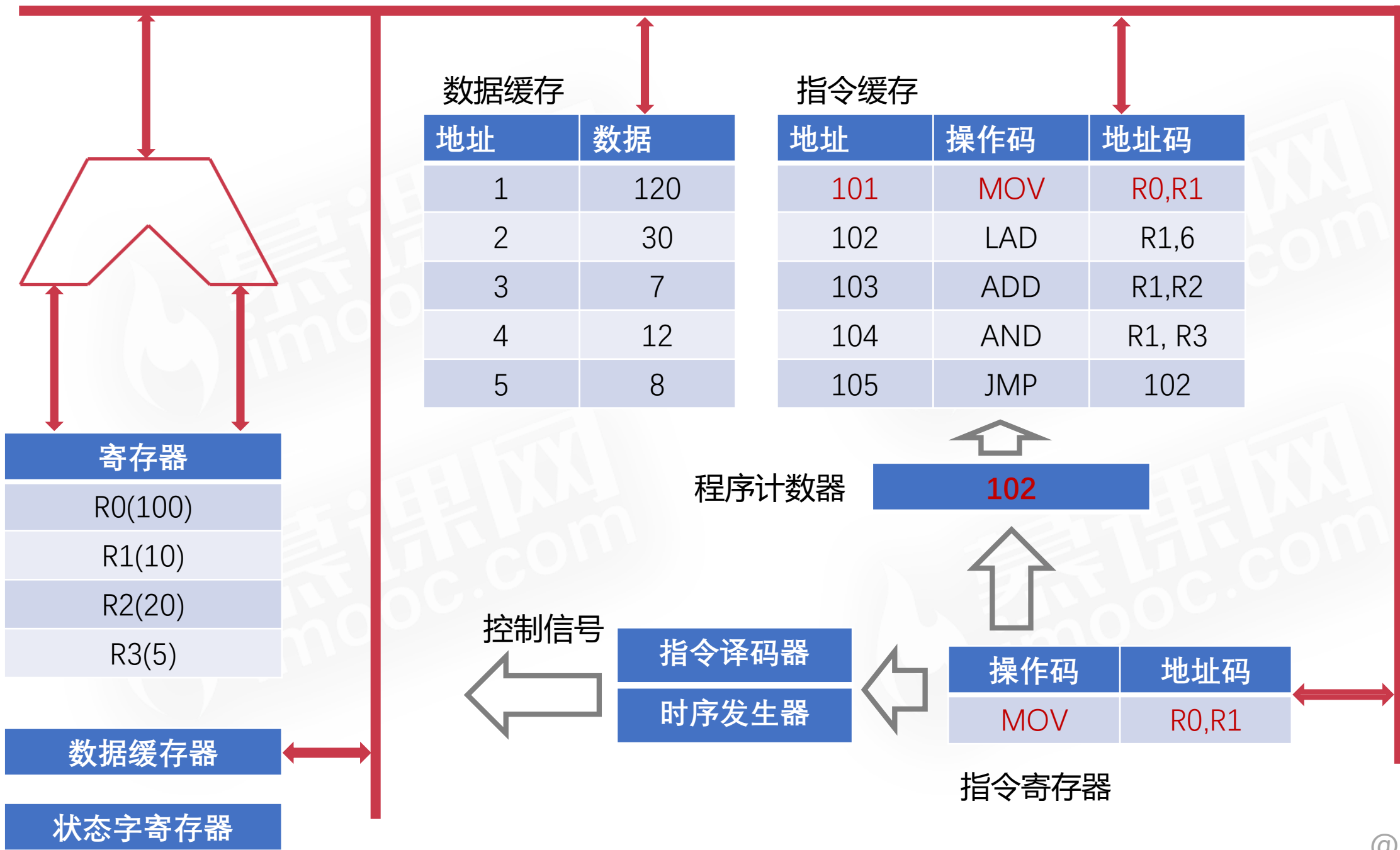


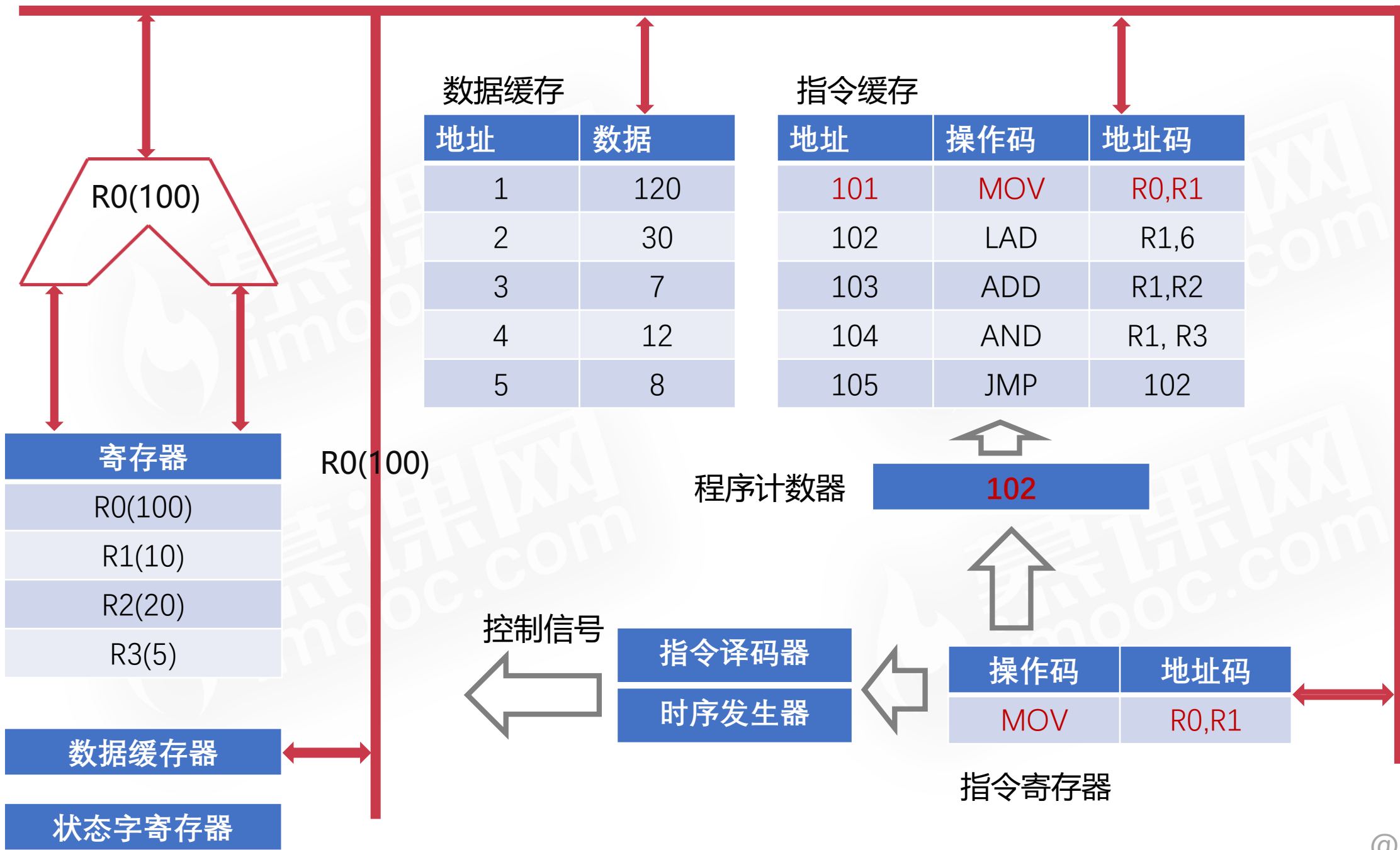


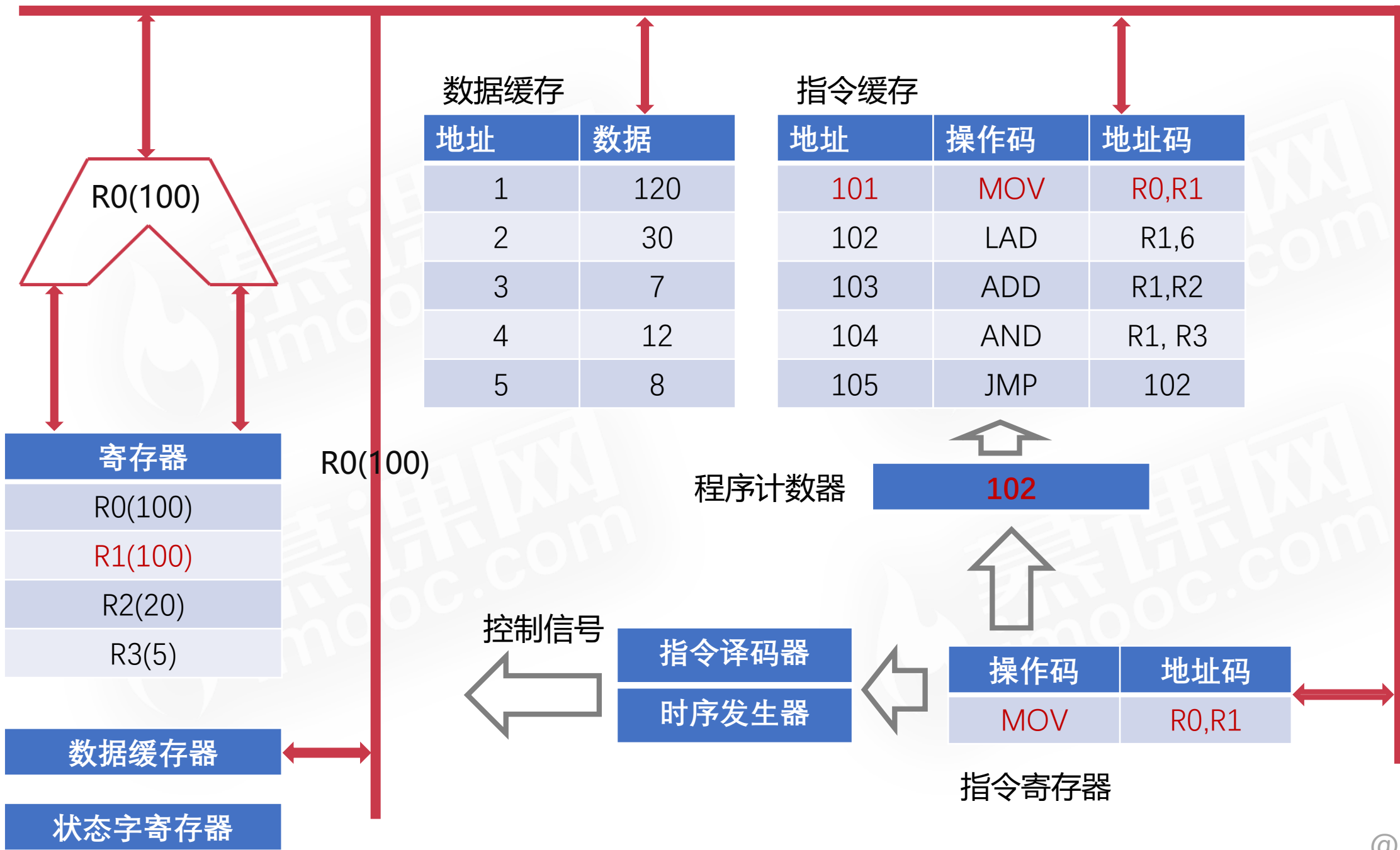












# 计算机指令的执行过程

## 指令执行过程





# 计算机指令的执行过程



**CPU的综合利用率并不高**

# 计算机指令的执行过程

- ◆ 指令执行过程
- ◆ CPU的流水线设计

# 计算机指令的执行过程

## CPU的流水线设计

- ◆ 类似工厂的装配线
- ◆ 工厂的装配线使得多个产品可以同时被加工
- ◆ 在同一个时刻，不同产品均位于不同的加工阶段

# 计算机指令的执行过程

## CPU的流水线设计

取指令

分析指令

执行指令

取指令

分析指令

执行指令

取指令

分析指令

执行指令

取指令

分析指令

执行指令

# 计算机指令的执行过程

## CPU的流水线设计

指令	时间片	时间片	时间片	时间片	时间片	时间片	时间片	时间片	时间片
1	取指令	分析指令	执行指令						
2		取指令	分析指令	执行指令					
3			取指令	分析指令	执行指令				
4				取指令	分析指令	执行指令			
5					取指令	分析指令	执行指令		
6						取指令	分析指令	执行指令	
7							取指令	分析指令	执行指令

# 计算机指令的执行过程

## CPU的流水线设计

串行执行m条指令:  $T_1 = 3t \times m$

流水线执行m条指令:  $T_2 = t \times (m + 2)$

$$H = \frac{T_2}{T_1} = \frac{t \times (m + 2)}{3t \times m} = \frac{1}{3} + \frac{1}{3m}$$

# 计算机指令的执行过程

- ◆ 指令执行过程
- ◆ CPU的流水线设计





# 计算机的异构计算设备

- ◆ 图像处理器(GPU)
- ◆ 现场可编程门阵列(FPGA)

# 计算机的异构计算设备

## 图像处理器(GPU)

- ◆ 又称显示核心、视觉处理器、显示芯片
- ◆ 擅长执行复杂的数学和几何计算

# 计算机的异构计算设备

## 图像处理器(GPU)

	CPU	GPU
结构	复杂	简单
性能	通用计算能力强	专有计算能力强
功耗	较低	较高
体积	较小	较大
用途	计算机的核心设备	计算机的附加设备

# 计算机的异构计算设备

## 图像处理器(GPU)



集成显卡



独立显卡

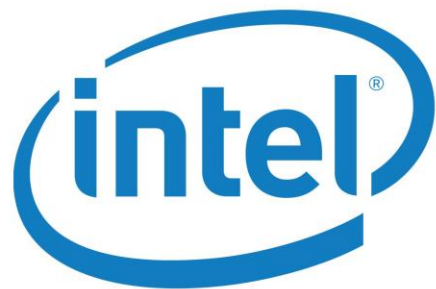
# 计算机的异构计算设备

## 图像处理器(GPU)

	集成显卡	独立显卡
兼容性	与主板耦合高，不便拆卸	与主板独立，方便拆卸
性能	性能适中	性能较高
功耗	功耗较低	功耗较高
价格	价格较低	价格较高，独立价格
应用场景	普通PC场景	科学计算、AI等
体积	体积较小，与主板融合	体积较大，台式机

# 计算机的异构计算设备

图像处理器(GPU)



INTEL



NVIDIA

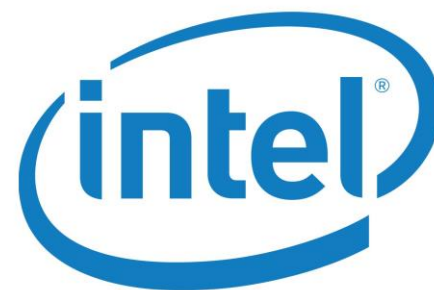


AMD

# 计算机的异构计算设备

## Intel

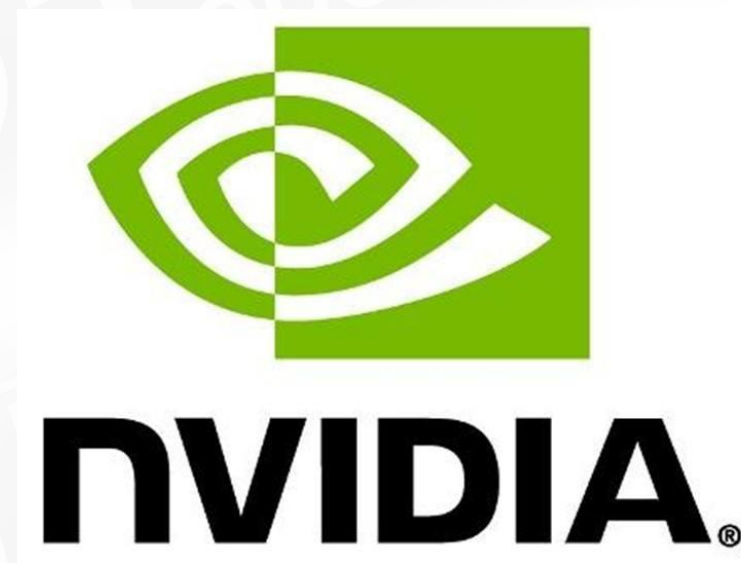
- ◆ 主要是集成显卡，配合自家的CPU
- ◆ 占据GPU市场的60%以上



# 计算机的异构计算设备

## Nvidia

- ◆ 世界上最大的独立显卡厂商
- ◆ Geforce系列、Quadro系列、Tesla系列





# 计算机的异构计算设备

## AMD

- ◆ 自AMD收购ATI后，市场进一步扩大
- ◆ 世界上第二大的独立显卡厂商
- ◆ 是常见游戏机Xbox、PS4等的GPU厂商



# 计算机的异构计算设备

## 现场可编程门阵列(FPGA)

- ◆ 是一种专用集成电路，诞生于1985年
- ◆ 电路级别的设备，属于硬件层面的开发
- ◆ AI人工智能的大热，FPGA逐渐走进大众的视野

# 计算机的异构计算设备

## 现场可编程门阵列(FPGA)



# 计算机的异构计算设备

## 现场可编程门阵列(FPGA)



# 计算机的异构计算设备

## 现场可编程门阵列(FPGA)



FPGA

# 计算机的异构计算设备

## 现场可编程门阵列(FPGA)

- ◆ AI的大规模参数训练
- ◆ 音视频、图像处理等复杂计算

# 计算机的异构计算设备

## 现场可编程门阵列(FPGA)

	GPU	FPGA
价格	价格较高	价格较低
功耗	功耗较高	功耗较低
兼容性	PCIe接口	PCIe接口
灵活性	灵活性较高	灵活性低
编程要求	要求较低	要求较高

# 计算机的异构计算设备

- ◆ 图像处理器(GPU)
- ◆ 现场可编程门阵列(FPGA)



慕课网  
imooc.com

慕课网  
imooc.com

慕课网  
imooc.com

慕课网  
imooc.com