

## 4.6 存储系统性能改进

常用来改进存储系统性能的措施：

- ✓ 更高速主存或加长存储器字长
- ✓ 双端口存储器
- ✓ 采用多级Cache
- ✓ 采用交叉存储器

...

## 4. 6. 1高速缓冲存储器Cache

### 1. 设置Cache的原因

- 为解决CPU和主存速度不匹配而采用的一项技术，使访问主存的平均速度接近于访问Cache的速度。
- 由硬件实现，对程序员透明。
- 已植入CPU内，两级以上的Cache系统。

## 2. Cache的前提条件

**局部性特征：**当CPU从主存中取指令和数据时，在一定时间内，地址范围常局限于主存的某个很小的区域内。

**因此，**把程序正在使用的部分预存在一个高速小容量的Cache中，使CPU的访存操作转换为访问Cache，从而使速度大大提高。

### 3. 主存与Cache的地址映射

#### (1) 直接映射

主存中每一个页只能映射到某一固定的Cache页中，直接映射有如下函数关系：

$$K=J \bmod 2^c$$

K为Cache的页号；

J为主存的页号；

C为Cache页号的位数。

直接映射的Cache组织，如图4-35示

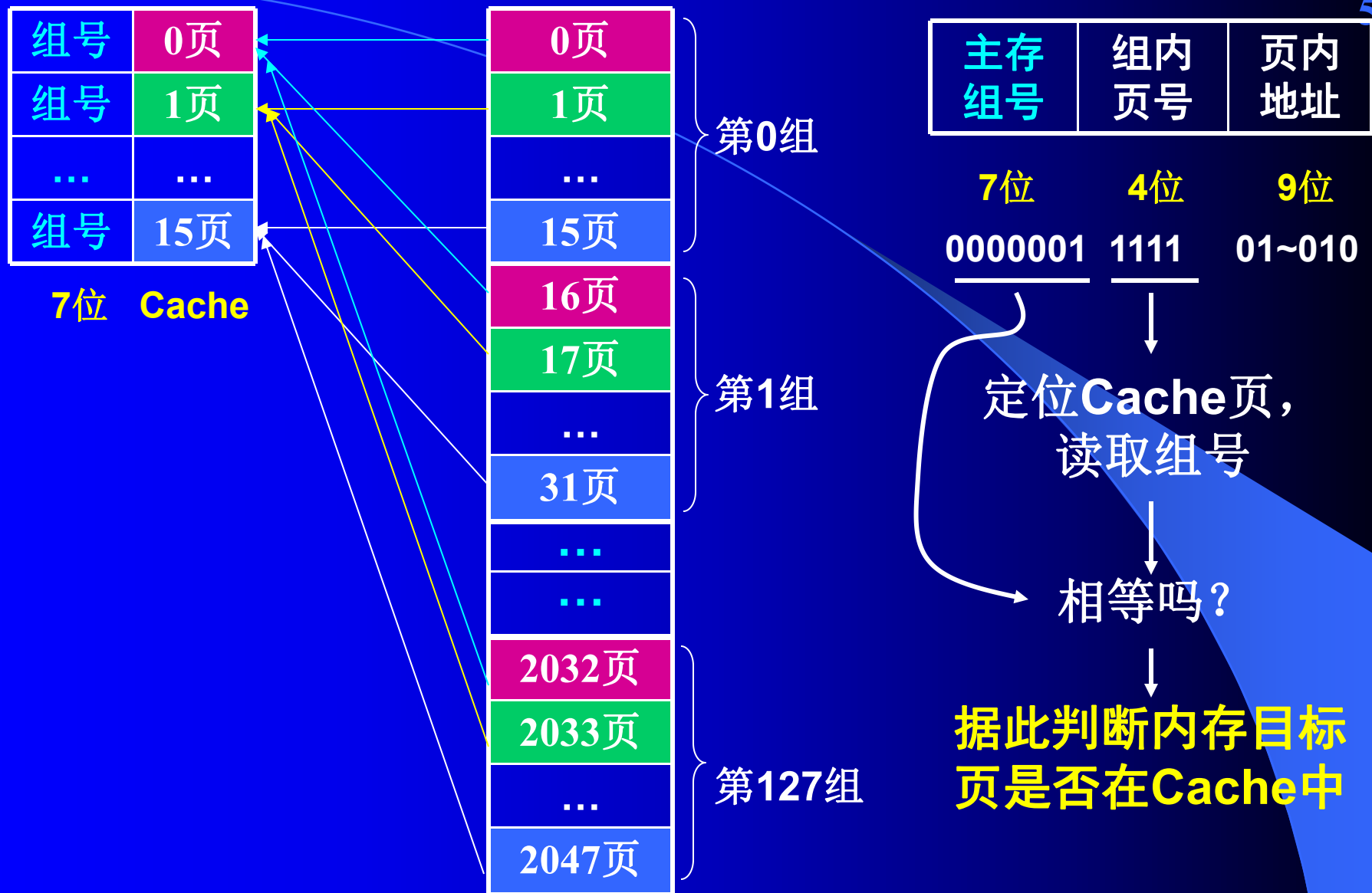


图4-35 直接映射  $C=4$ 、 $N_a=20$ ，每页512字节

## (2) 全相联映射

主存中任何一块都可以映射到Cache中的任何一块位置上，如图4-36示。

### 存在的缺点：

- ✓ Cache标记太长，判断时间太长。
- ✓ 其硬件复杂、成本高、实现起来比较困难。

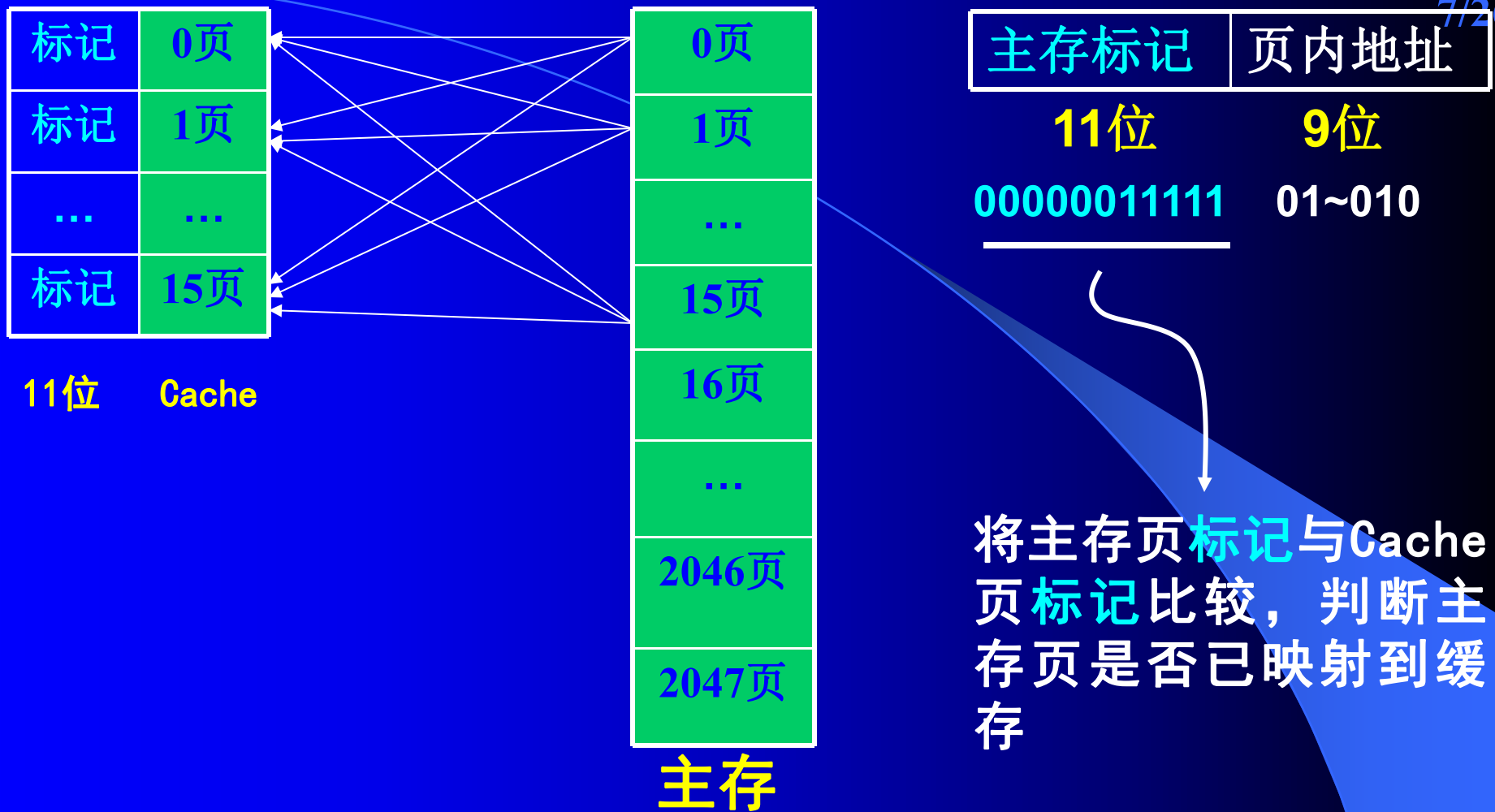


图4-36 全相联映射  $N_a=20$ ，每页512字节

### (3) 组相联映射

是一种**直接映射**和**全相联映射**的折衷方案。主存和Cache都分组，主存中组内的页数与Cache的分组数相同。

- 主存页号到Cache组号之间直接映射，主存页与Cache组内各页全相联映射。
- 判断的速度较快，硬件较简单、成本较低、比较容易实现。

组相联映射组织结构，如图4-37所示。



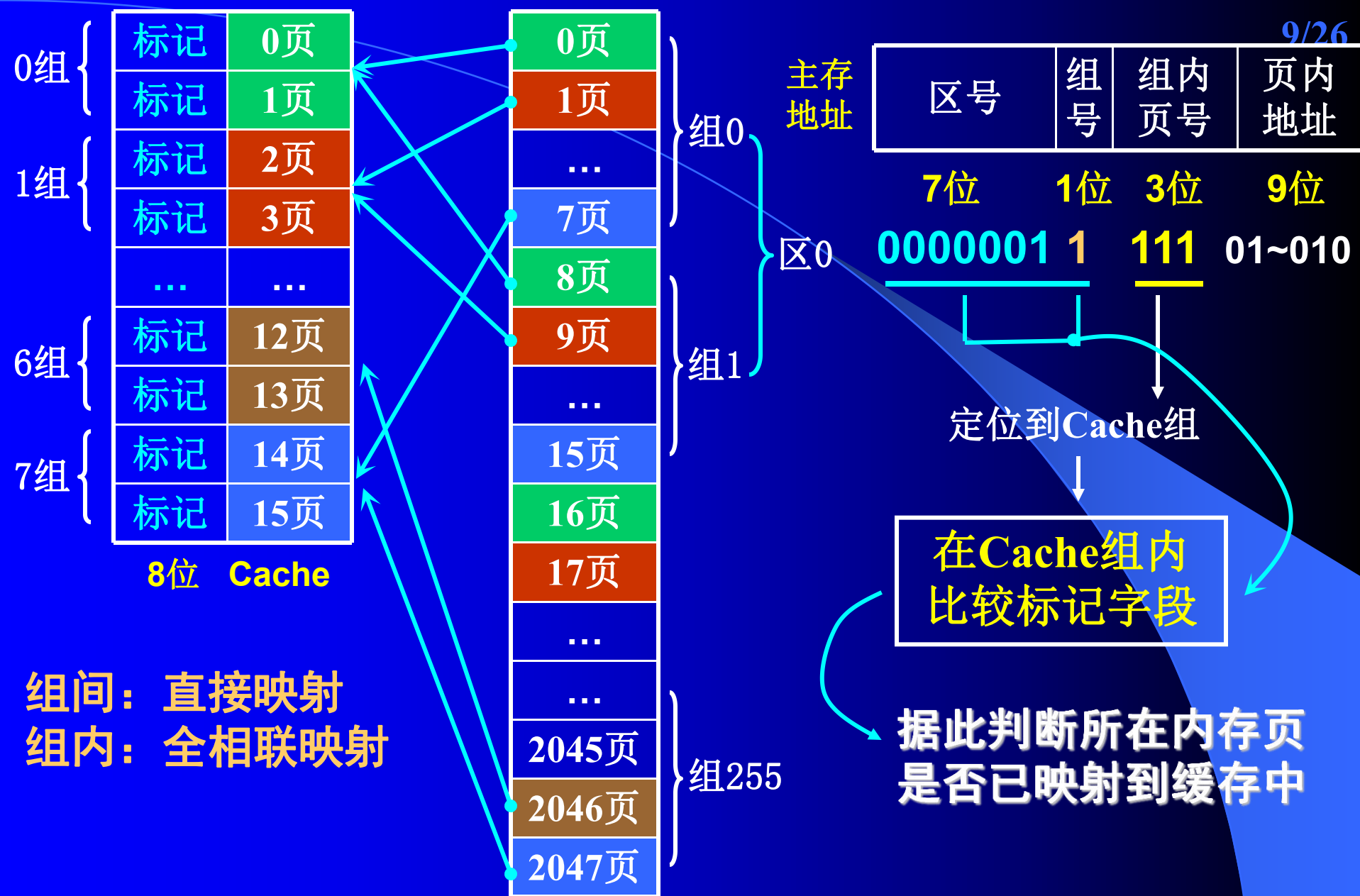


图4-37 组相联映射  $N_a=20$ ，每页512字节

## 4. 常用的替换算法

### (1) 最不经常使用 (LFU, Least-Frequently Used)

- 认为应将一段时间内被访问次数最少的那块从Cache中置换出去。
- 计数周期限定在对这些特定块两次替换之间的间隔时间内，不能严格反映近期访问情况。

### (2) 近期最少使用 (LRU, Least-Recently Used)

- 将近期内长久末被访问过的Cache块置换出去。
- 该算法保护了刚拷贝到Cache中的新数据块，符合Cache工作原理，使Cache有较高的命中率。

### (3) 随机替换

## 5. Cache的读/写操作

### ● Cache的写操作

当CPU发出写请求时，如果Cache命中，可有两种处理方案：

- ① Cache单元和主存单元同时写，使Cache和主存保持一致，称为**通写**(write-through)。
- ② 只修改Cache单元，并用标志将该块加以注明，直到该块从Cache中替换出来时才一次性写入主存，称为**回写**(write-back)。

# ● Cache的读操作

## ① 旁路式读 (Look-Aside)

CPU向Cache和主存同时发读命令和地址。

Cache命中，则Cache回送数据并中断读主存命令；

Cache未命中，则直接访问主存读取数据。

## ② 通过式读(Look-Through)

CPU首先向Cache发读命令和地址。

Cache命中，则从Cache中读出数据；

Cache未命中，再将读命令和地址传给主存并读主存。

**讨论：命中率、平均访问时间和访问效率？**

## 4.6.2 虚拟存储器

### 1. 虚拟存储器的基本概念

- 建立在主存—外存层次上的由操作系统存储管理软件及附加硬件装置（存储器管理部件MMU）组成的存储体系。
- 它以透明的方式给用户提供了一个访问速度接近主存储器，而存储空间比实际主存空间大得多的存储器。
- 此时程序的逻辑地址称为虚地址，程序的逻辑地址空间称为虚地址空间。

## 2. 页式虚拟存储器

主存和外存统一分页后进行管理。

- 建立一张虚地址页号与实地址页号的对照表，称为**页表**，记录程序的虚页面调进主存时被安排在主存中的位置。
- 硬件中设置一个**页表基址寄存器**，存放当前所运行程序的**页表的起始地址**。
- 页表中的每一行对应一个虚页号，称为一个**登记项**。
- 活跃的页表存放于专用存储器，称为**快表**。

页表基址寄存器

页表起始地址

虚页号

页内地址

页 表

虚地址

页表行地址

实页号

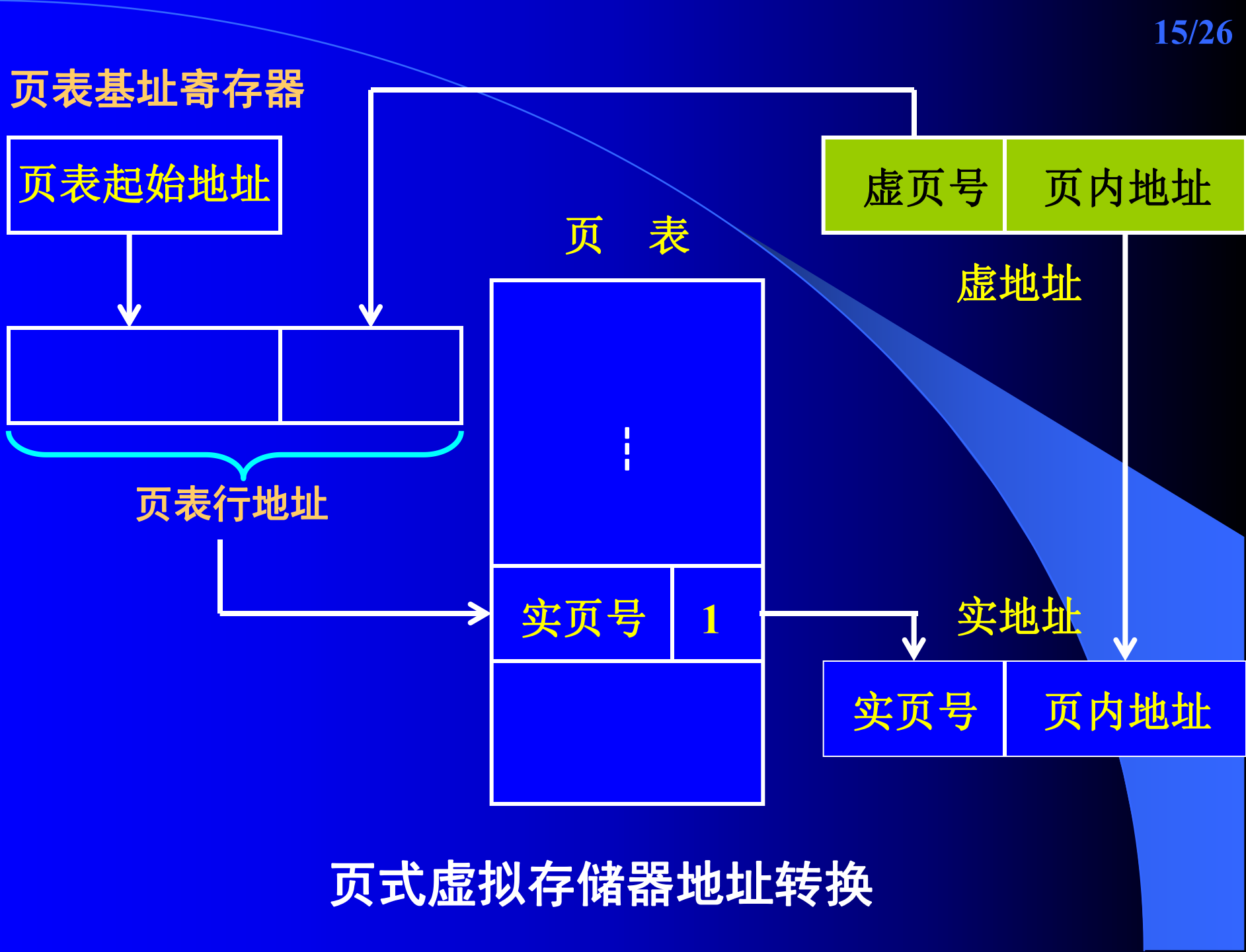
1

实地址

实页号

页内地址

页式虚拟存储器地址转换



### 3. 段式虚拟存储器

外存中程序分段（按照代码段、数据段和共享段等）进行管理。

- 为了将虚拟地址变换成主存实地址，需要一个**段表**。
- 每个程序段在段表中都占有一登记项，内容有：**段号、段起点、段长、装入位**等。
- 虚实地址变换，如后图所示。



段表基址寄存器

段表起始地址

虚地址

段号

段内地址

段 表

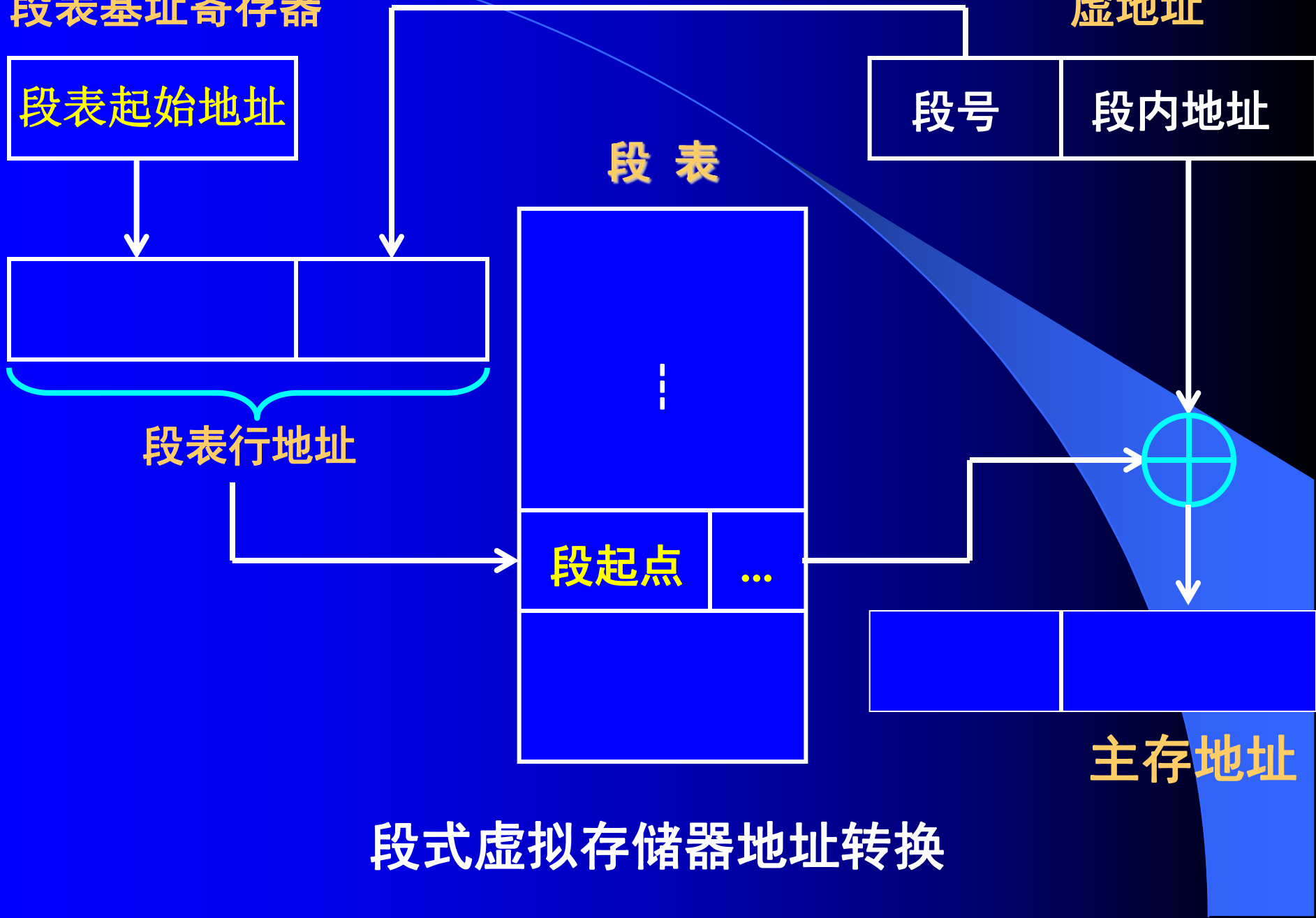
段表行地址

段起点

...

主存地址

段式虚拟存储器地址转换



## 4. 段页式虚拟存储器

每个程序按模块分段，每段再划分为页，页面大小与实存页面相同；

- 虚地址格式：段号+页号+页内地址；
- 实地址格式：页号+页内地址；
- 每个程序有一张段表，每段对应有一张页表；
- 要经过两级查表才能完成地址转换，耗时长；

## 4. 6. 3双端口存储器

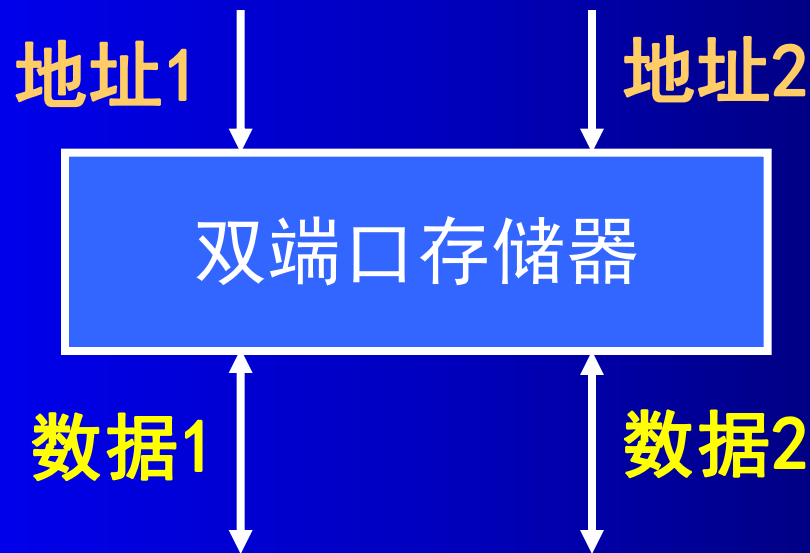
### [主存速度与CPU处理速度存在差距]

低速主存无法为CPU提供快速服务。如需同时需要用访问2个主存单元时，更限制了CPU高速处理信息能力的发挥。

为了使CPU缩短等待时间，通常采取一些加速CPU和存储器之间数据传输的特殊措施，如**双端口存储器**等。

# 1. 双端口存储器的逻辑结构

**双端口存储器：** 同一个存储器具有两组独立的读写控制线路，两个端口分别具有各自的地址线、数据线和控制线，可进行独立的存取操作。



## 2. 无冲突读写控制

当两端口地址不同时，在两端口上进行读写，不会发生冲突，可同时进行读写。

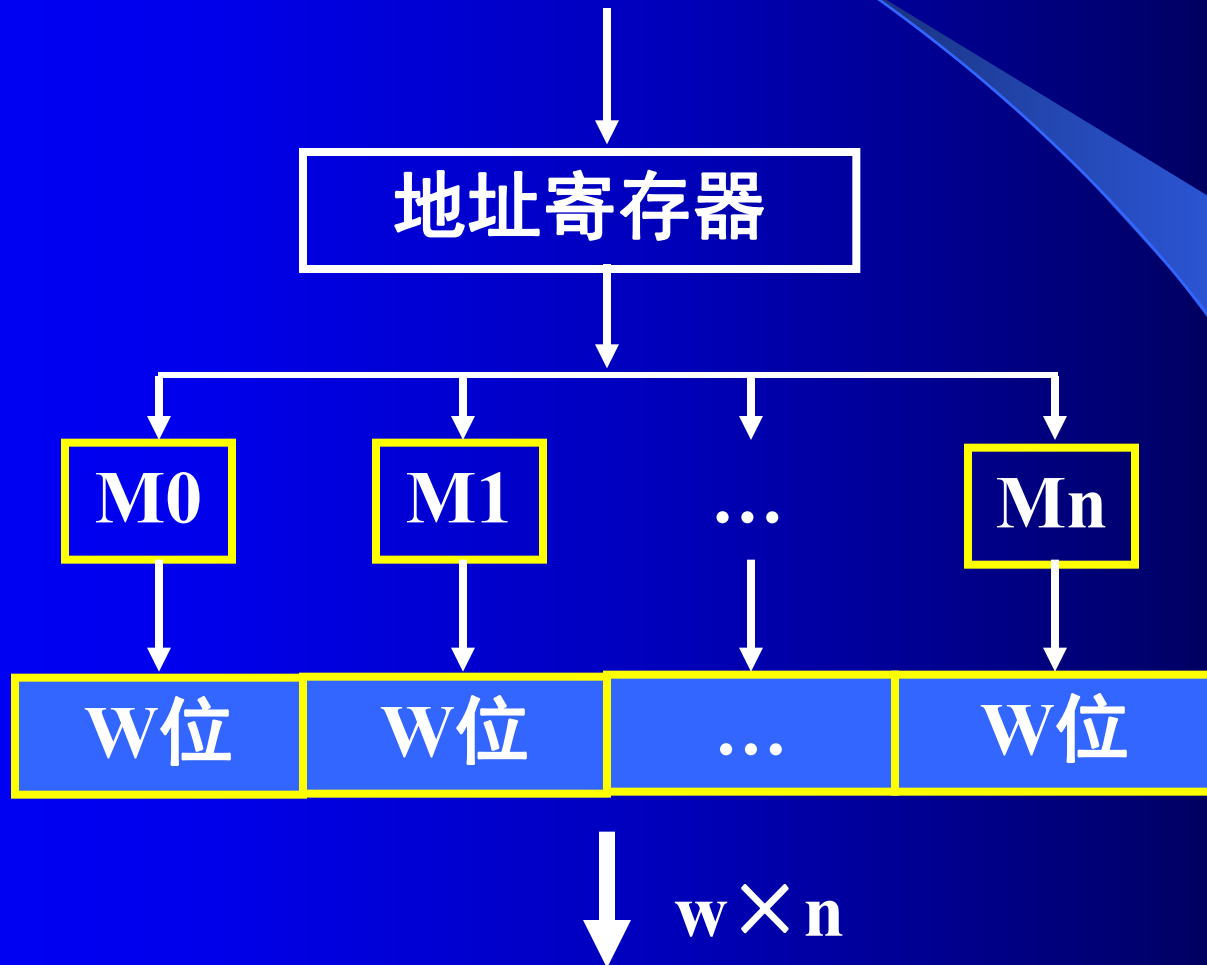
## 3. 有冲突读写控制

**[问题]**当两个端口同时存取同一存储单元时，会发生端口间的读写冲突。

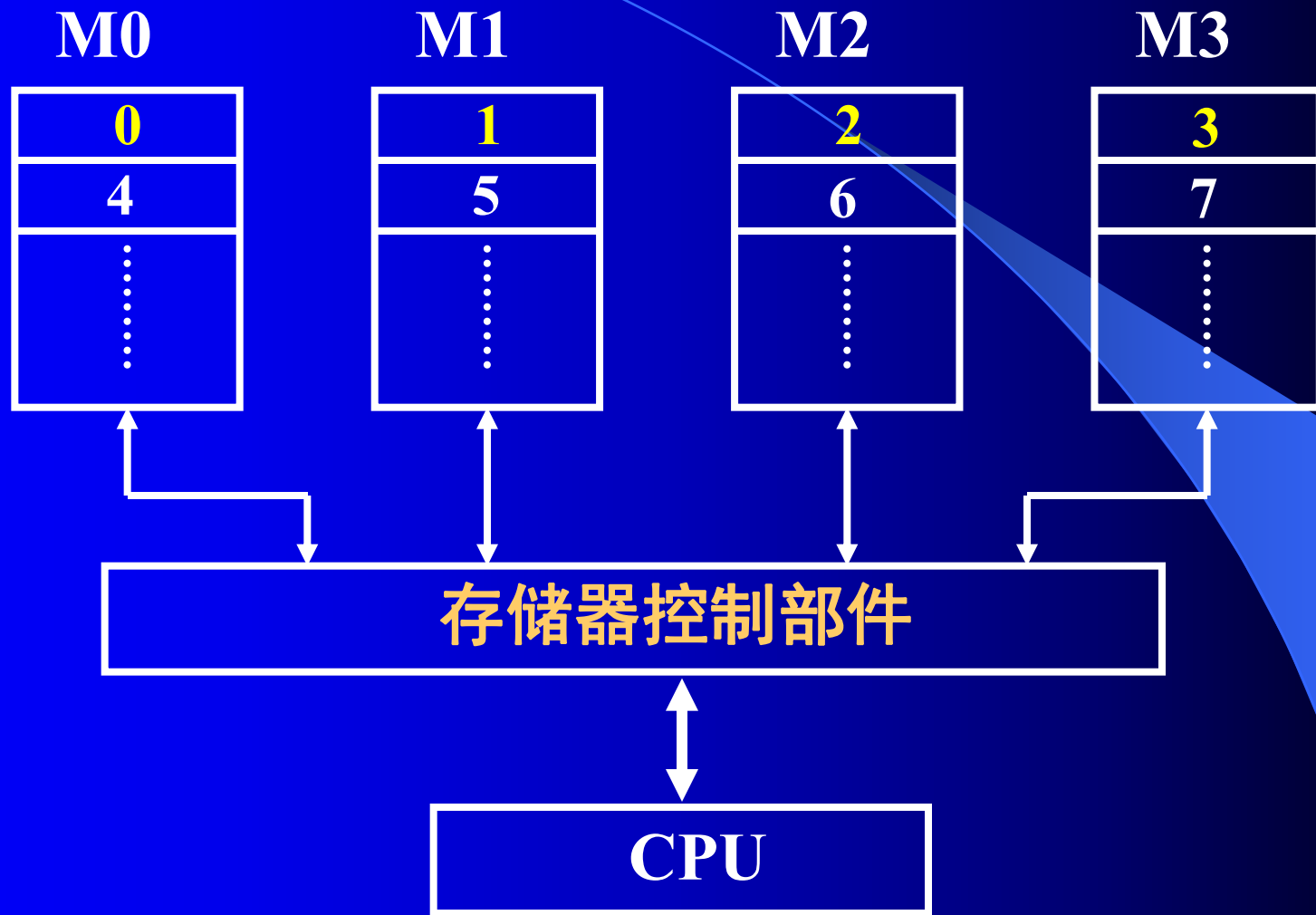
**[解决方法]**设置BUSY标志，采用仲裁逻辑，由芯片上的判断逻辑决定由哪个端口优先进行读写操作，而暂时关闭另一个被延迟的端口。

## 4.6.4 并行存储器

### 1. 单体多字并行主存系统



## 2. 多体交叉并行主存系统



四分体并行存储系统

## 4.6.5 相联存储器 葛

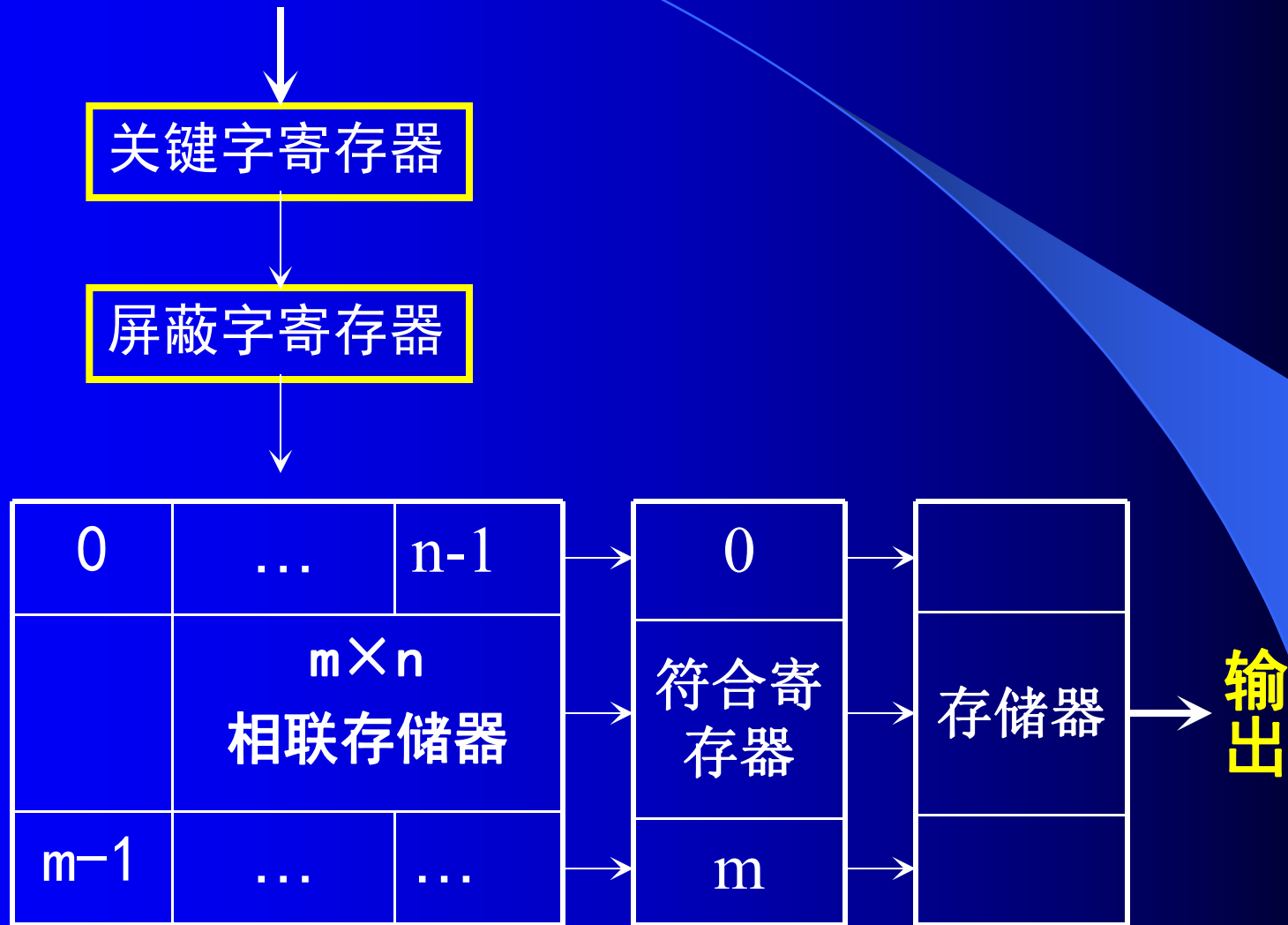
### 1. 基本原理

根据存储单元所存内容的一部分作为检索项(即关键字项)，去检索存储器，并对存储器中与该检索项符合的存储单元内容进行读出或写入。

- ✓根据存储的**内容**进行寻址。
- ✓寻址字段叫做**关键字**。
- ✓存储的内容：**关键字+数据**，其中关键字是地址，数据则是被读写的信息。



## 2. 相联存储器的组成



### 3. 相联存储器的应用

在计算机系统中，相联存储器 (**Associative Memory**) 主要用于存放需要快速查找的内容，如：虚拟存储器中存放**段表**、**页表**和**快表**等；在高速缓冲存储器中，相联存储器作为存放 cache 的行地址之用。