计算机组成与系统结构

第三章 多层次的存储器

吕昕晨

lvxinchen@bupt.edu.cn

网络空间安全学院

第三章 多层次的存储器

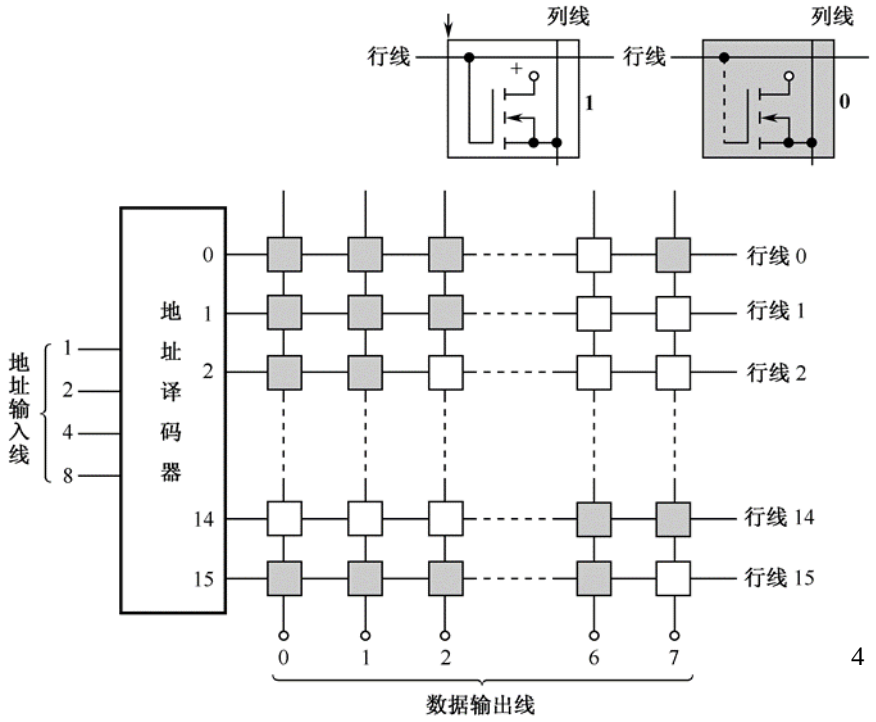
- 只读存储器（ROM）
- 并行存储器
- 虚拟存储器
- 奔腾系列机的虚存组织

只读存储器ROM

- 只读存储器（Read-Only Memory, ROM）
  - 只能读出，不能写入（不能随意写入）
  - 存储的原始数据，必须在它工作以前写入
  - 工作可靠，保密性强，在计算机系统中得到广泛的应用
- 主要有两类
  - 掩模ROM
    - 掩模ROM实际上是一个存储内容固定的ROM，由生产厂家提供产品
  - 可编程ROM：用户后写入内容，有些可以多次写入
    - 一次性编程的PROM
    - 多次编程的EPROM和E^2PROM

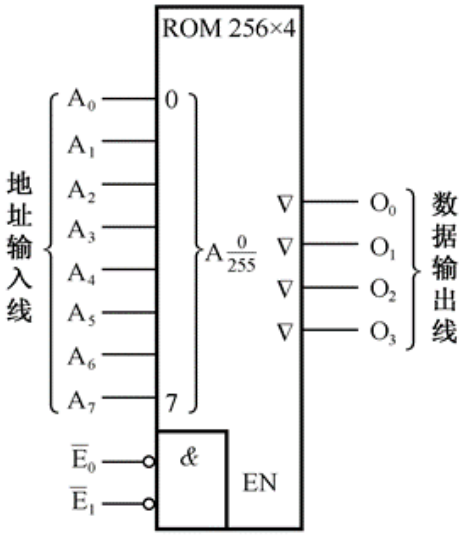
掩模ROM存储结构

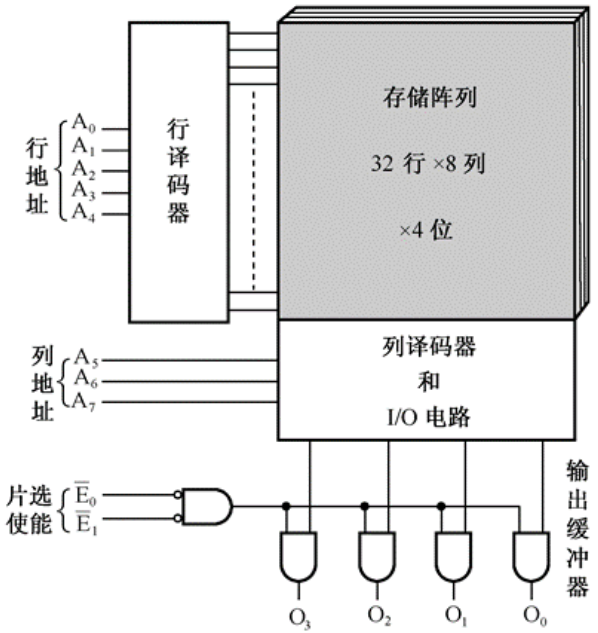
- 存储元
  - MOS管
  - 行选线与MOS管栅极是否连通（0/1）
  - 生产商制造ROM设置
- 阵列结构
  - 单译码结构
  - 16 * 8bit



掩模ROM芯片与逻辑框图

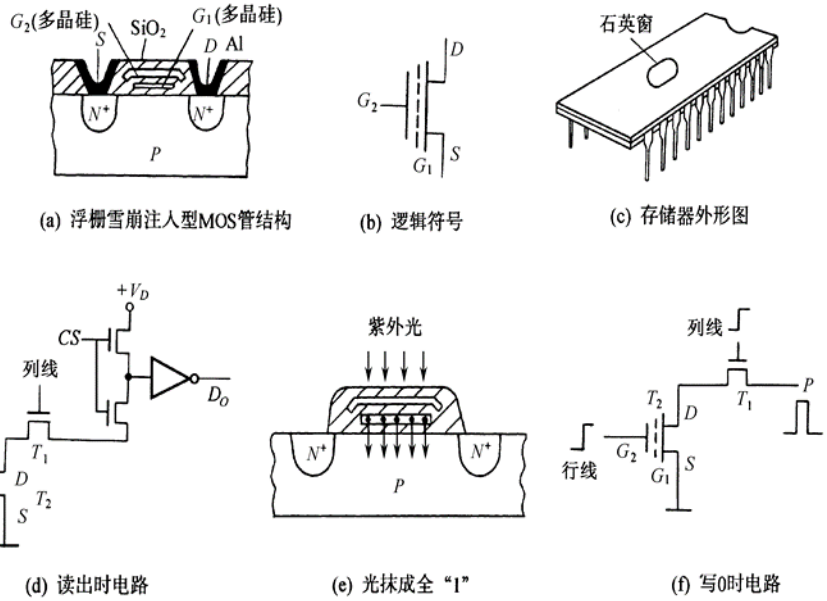
- 掩模ROM芯片
  - 256*4bit ROM芯片
  - 控制线E0/E1（与门连接），均为低电平可读出
  - 行列译码电路
  - 参考SRAM/DRAM扩展





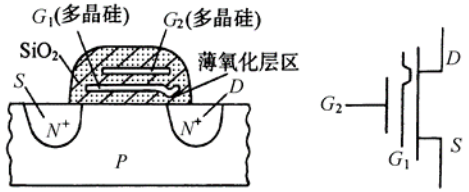
可编程EPROM

- EPROM：光擦除可编程可读存储器
  - 存储内容可以根据需要写入，当需要更新时将原存储内容抹去，再写入新的内容
- 存储元
  - 浮栅雪崩注入型MOS管
  - 光擦除方式（抹为全1），紫外光照射
  - 可读出，可写0

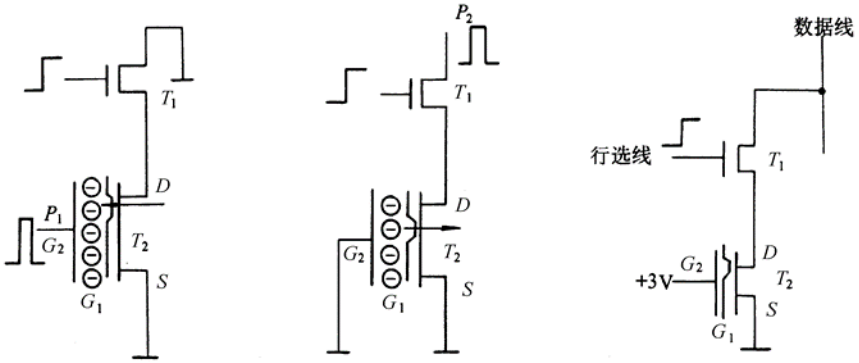


可编程E²PROM

- EEPROM/E²PROM：电擦除可编程可读存储器
- 存储元
  - 两个栅极的NMOS管
  - G1是控制栅，它是一个浮栅，无引出线
  - G2是抹去栅，它有引出线
  - 电擦除方式，在G2加20V正脉冲，存储1（20ms）
  - 可读出，可写0，抹1



(a) 结构图　　(b) 逻辑符号



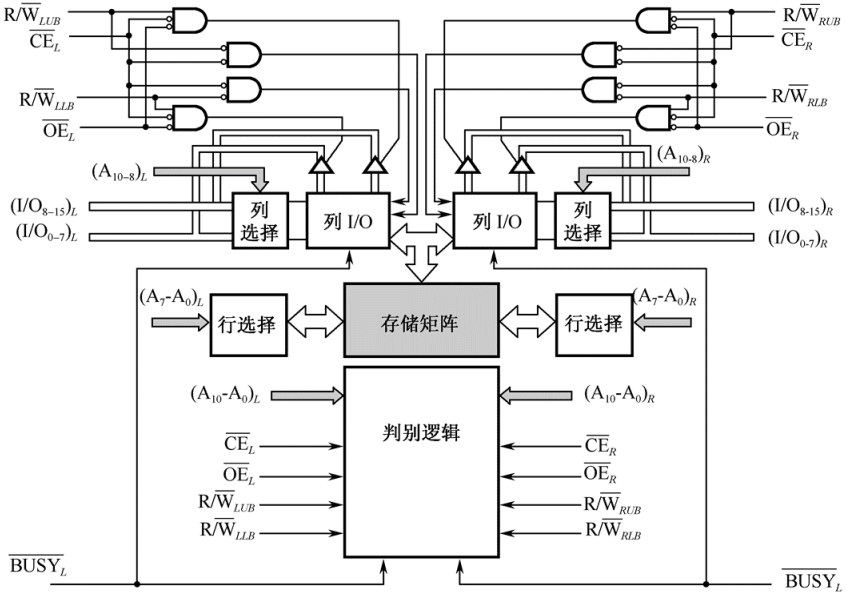
(c) 抹成全“1”　　(d) 写0时电路　　(e) 读出时电路

第三章 多层次的存储器

- 只读存储器（ROM）
- 并行存储器
  - 双端口存储器（空间并行）
  - 多体交叉存储器（时间并行/流水线）
- 虚拟存储器
- 奔腾系列机的虚存组织

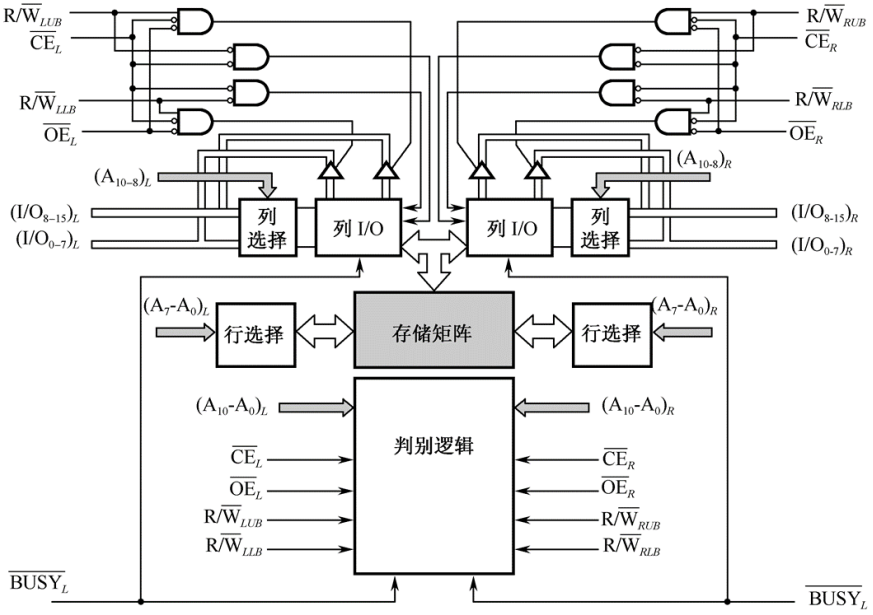
双端口存储器逻辑框图

- 双端口存储器
  - 同一个存储器具有两组相互独立的读写控制电路
  - 由于进行并行的独立操作，因而是一种高速工作的存储器
- 双端口存储器IDT7133的逻辑框图
  - 并行读写控制电路
  - 判别逻辑（冲突）



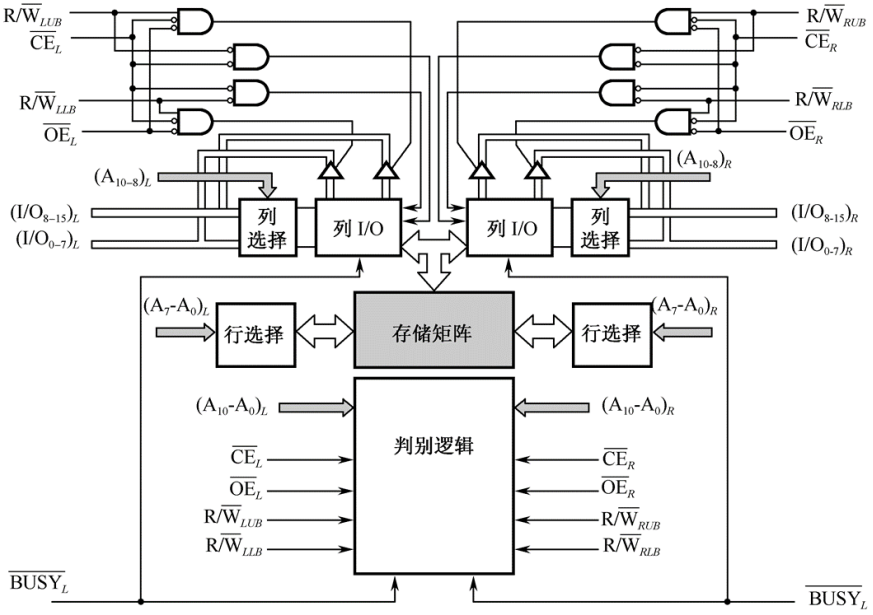
双端口存储器读写控制（1）

- 无冲突读写控制
  - 当两个端口的地址不相同时，在两个端口上进行读写操作，不会发生冲突
  - 每一个端口都有自己的片选控制(CE)和输出驱动控制(OE)
  - 读操作时，端口的OE(低电平有效)打开输出驱动器，由存储矩阵读出的数据就出现在I/O线上



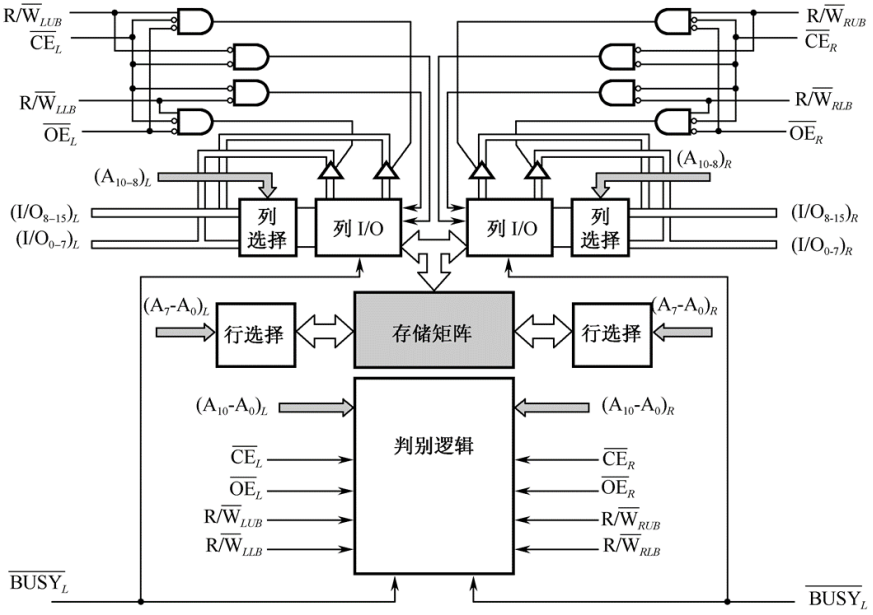
双端口存储器读写控制（2）

- 有冲突读写控制
  - 当两个端口同时存取存储器同一存储单元时，便发生读写冲突
  - 设置了BUSY标志
  - 片上的判断逻辑可以决定对哪个端口优先读写操作
  - 另一个被延迟的端口置BUSY标志(BUSY变为低电平)，即暂时关闭此端口



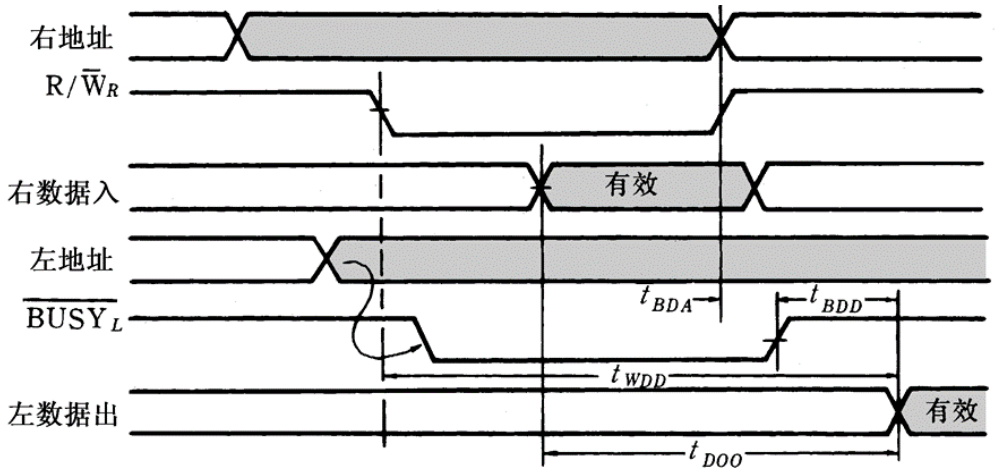
双端口存储器读写控制（3）

- 有冲突读写控制判断方法
- CE判断（片选有效判断）
  - 如果地址匹配且在CE之前有效，片上的控制逻辑在CEL和CER之间进行判断来选择端口
- 地址有效判断
  - 如果CE在地址匹配之前变低，片上的控制逻辑在左、右地址间进行判断来选择端口
- 判断标准：有效时间

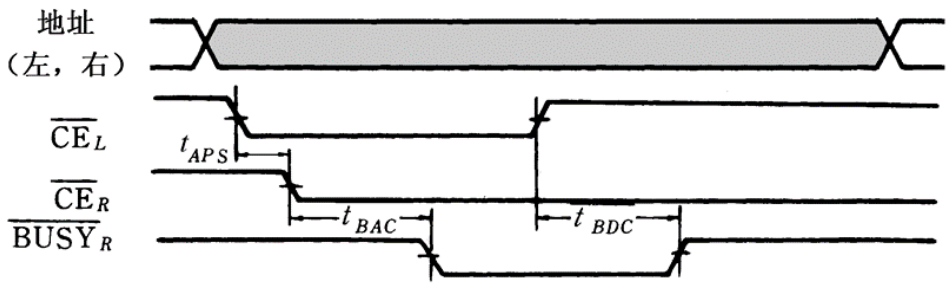


双端口存储器读写时序

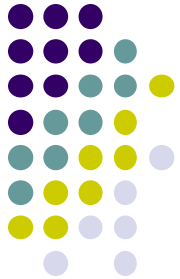
- 正常时序
  - 未发生冲突
  - 左读，右写



- 冲突时序
  - CE判断（地址先有效）
  - CEL先有效
  - 右端口被延迟



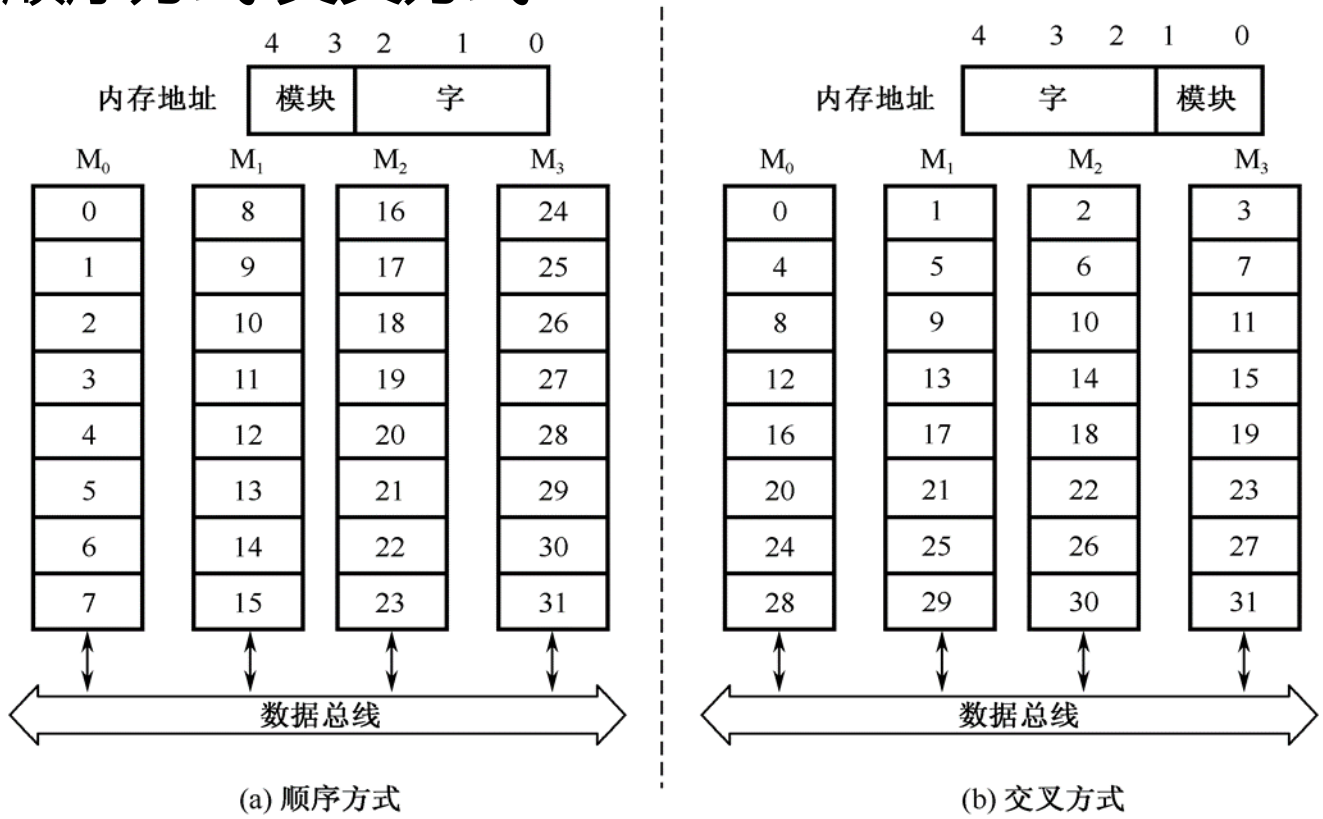
第三章 多层次的存储器



- 只读存储器（ROM)
- 并行存储器
  - 双端口存储器（空间并行)
  - 多体交叉存储器（时间并行/流水线)
- 虚拟存储器
- 奔腾系列机的虚存组织

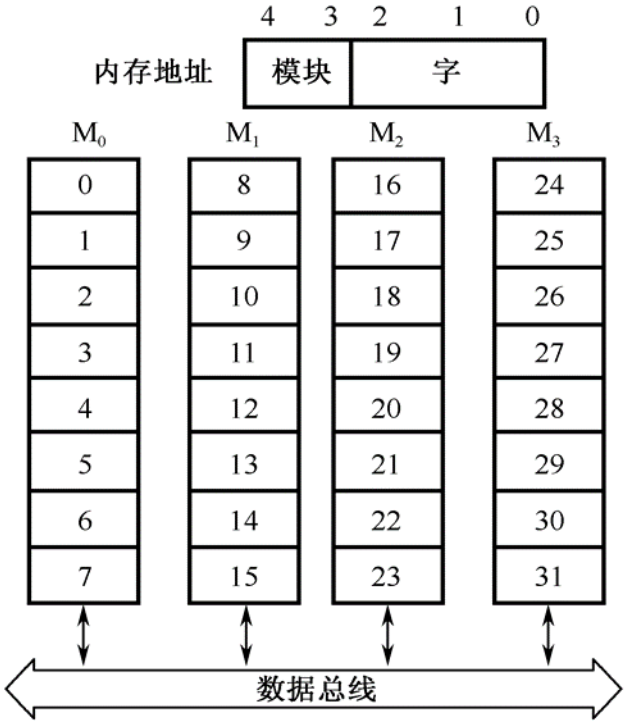
存储器的模块化组织

- 由若干个模块组成的主存储器：线性编址
- 编址方式
  - 顺序方式/交叉方式



顺序方式

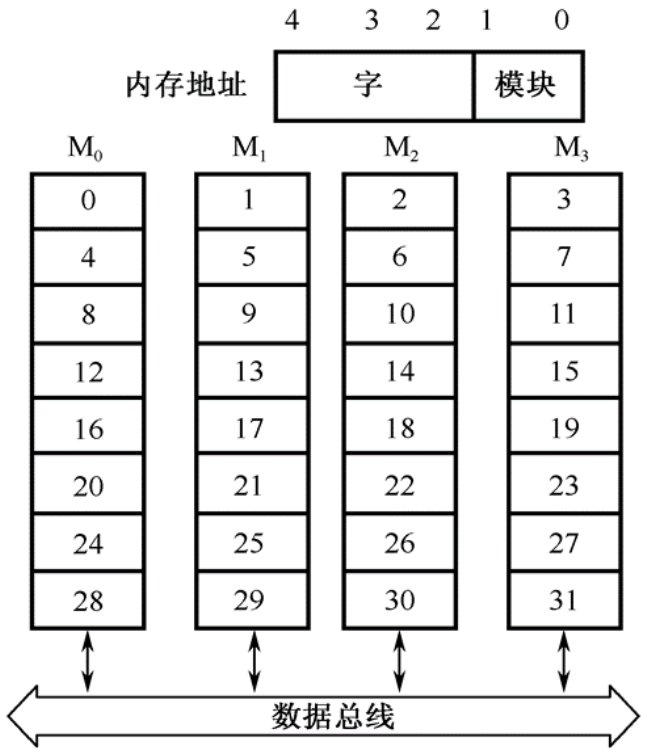
- 顺序方式
  - M0：0—7
  - M1：8 - 15
  - M2：16 - 23
  - M3：24 - 31
- 5位地址组织
  - 高位选模块，低位选块内地址
- 特点
  - 某个模块进行存取时，其他模块不工作
  - 优点是某一模块出现故障时，其他模块可以照常工作，通过增添模块来扩充存储器容量比较方便
  - 缺点是各模块串行工作，限制存储器带宽



(a) 顺序方式

交叉方式

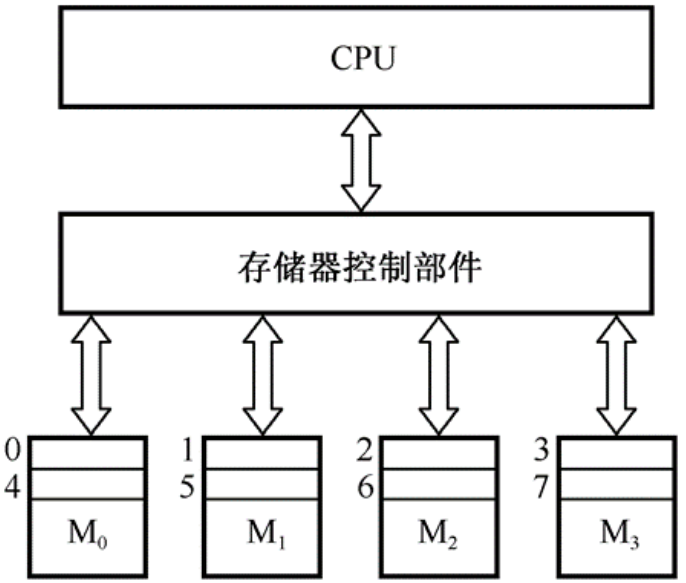
- 交叉方式：
  - M0：0，4,...除以4余数为0
  - M1：1，5,...除以4余数为1
  - M2：2，6,...除以4余数为2
  - M3：3，7,...除以4余数为3
- 5位地址组织
  - 高位选块内地址，低位选模块
- 特点
  - 连续地址分布在相邻的不同模块内，同一个模块内的地址都是不连续的
  - 优点是对连续字的成块传送可实现多模块流水式并行存取，大大提高存储器的带宽
  - 使用场合为成批数据读取



(b) 交叉方式

多模块交叉存储器基本结构

- 四模块交叉存储器结构框图
  - 主存被分成4个相互独立、容量相同的模块M0，M1，M2，M3
  - 每个模块都有自己的读写控制电路、地址寄存器和数据寄存器，各自以等同方式与CPU传送信息
  - 在理想情况下
    - 如果程序段或数据块都是连续地在主存中存取
    - 主存的访问速度提升接近4倍



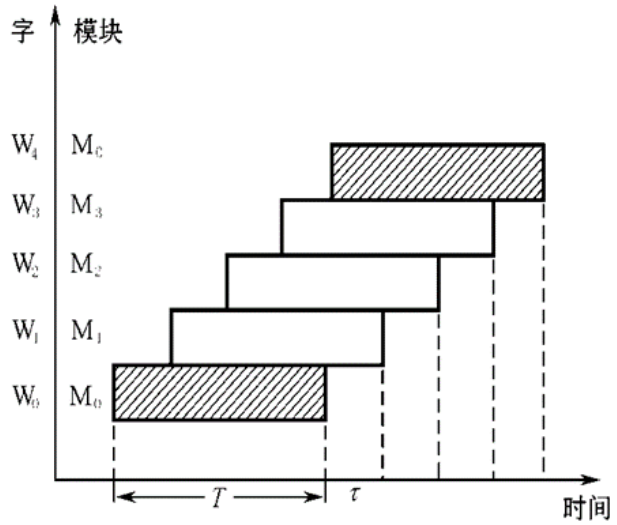
多模块交叉存储器效率

- 通常在一个存储器周期内
  - n个存储体必须分时启动
  - 各个存储体的启动间隔为 $t = T / n$
  - n为交叉存取度（顺序方式n=1，交叉方式为扩展芯片数）

- 整个存储器的存取速度有望提高n倍

$$t_{顺序} = xT$$

$$t_{交叉} = T + (x-1)t = T(\frac{x+n-1}{n})$$



例5 设存储器容量为32字，字长64位，模块数m=4，分别用顺序方式和交叉方式进行组织。存储周期T=200ns，数据总线宽度为64位，总线传送周期=50ns。
若连续读出4个字，问顺序存储和交叉存储的带宽各是多少?

解:

顺序存储器和交叉存储器连续读出m=4个字的信息总量都是:

$$q=64\text{（字长）}\times 4=256\ \text{bit}$$

顺序存储器和交叉存储器连续读出4个字所需的时间分别是:

$$t_{顺序}=mT=4\times 200ns=800ns=8\times 10^{-7}s$$

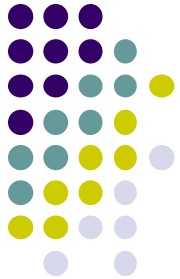
$$t_{交叉}=T+(m-1)T/n=200+3*200/4=350ns=35\times 10^{-7}s$$

顺序存储器和交叉存储器的带宽分别是:

$$W_{顺序}=q/t_{顺序}=256b\div(8\times 10^{-7})s=320Mb/s$$

$$W_{交叉}=q/t_{交叉}=256b\div(35\times 10^{-7})s=730Mb/s$$

第三章 多层次的存储器



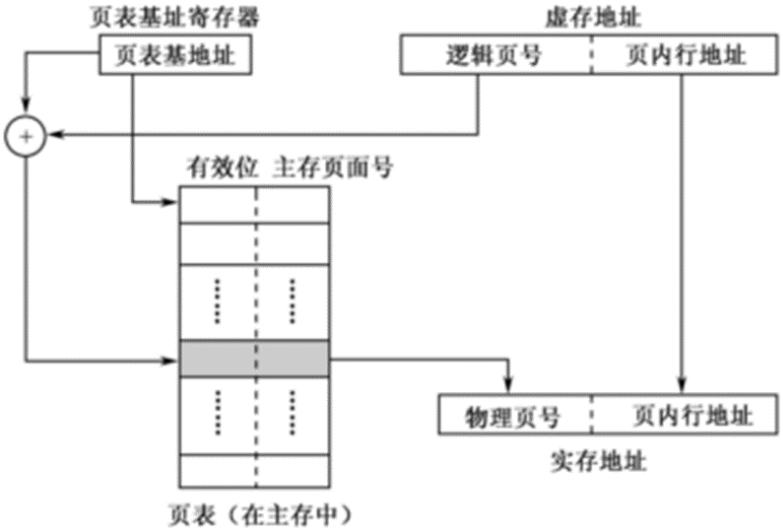
- 只读存储器（ROM)
- 并行存储器
- 虚拟存储器
  - 虚存地址映射方式：页式/段式/段页式
  - 虚存替换算法
- 奔腾系列机的虚存组织

实地址与虚地址

- 虚存空间的程序按照虚地址存放在辅存中
  - 程序运行时，由地址变换机构依据当时分配给该程序的实地址空间把程序的一部分调入实存
  - 每次访存时，首先判断该虚地址所对应的部分是否在实存中
    - 如果是，则进行地址转换并用实地址访问主存
    - 否则，按照某种算法将辅存中的部分程序调度进内存，再按同样的方法访问主存
- 虚地址空间可以远大于实地址空间
  - 存储容量扩展，内存大小不足，分批放入实存
- 虚地址空间也可以远小于实地址空间
  - 高性能服务器，实存空间过大，程序无法使用，较小虚存空间可缩短地址字段长度

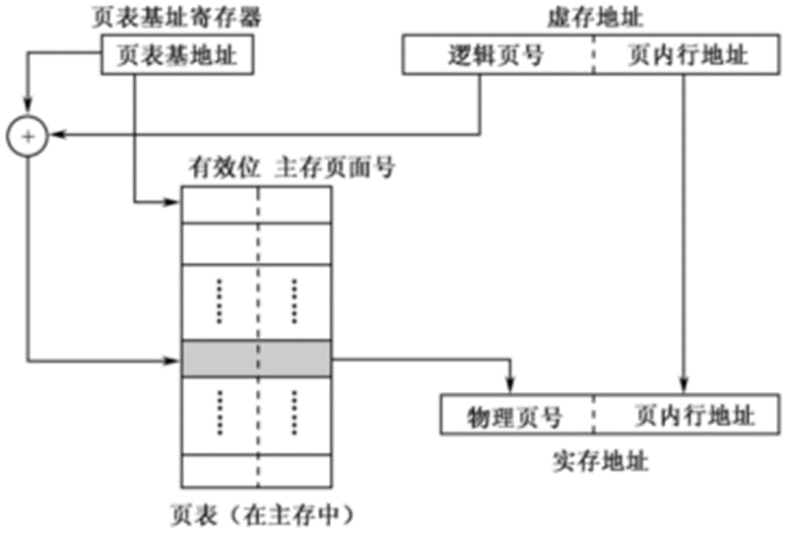
页式虚拟存储器

- 虚地址空间被分成等长的页（逻辑页）
- 主存空间也被分成同样大小的页（物理页）
- 地址字段：
  - 高字段为逻辑（物理）页号
  - 低字段为页内地址（偏移量）；实存地址也分两个字段
  - 通过页表可以把虚地址（逻辑地址）转换成物理地址。



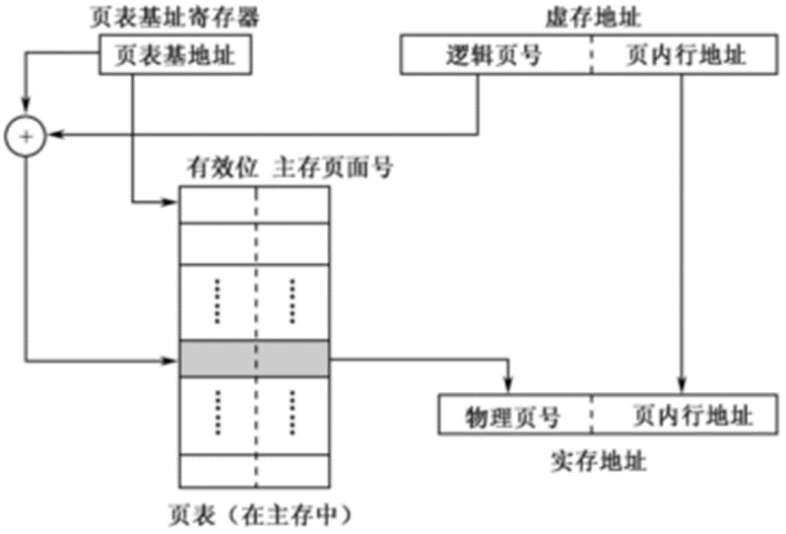
页式虚拟存储器

- 虚地址空间被分成等长的页（逻辑页）
- 主存空间也被分成同样大小的页（物理页）
- 地址字段：
  - 高字段为逻辑（物理）页号
  - 低字段为页内地址（偏移量）；实存地址也分两个字段
  - 通过页表可以把虚地址（逻辑地址）转换成物理地址。



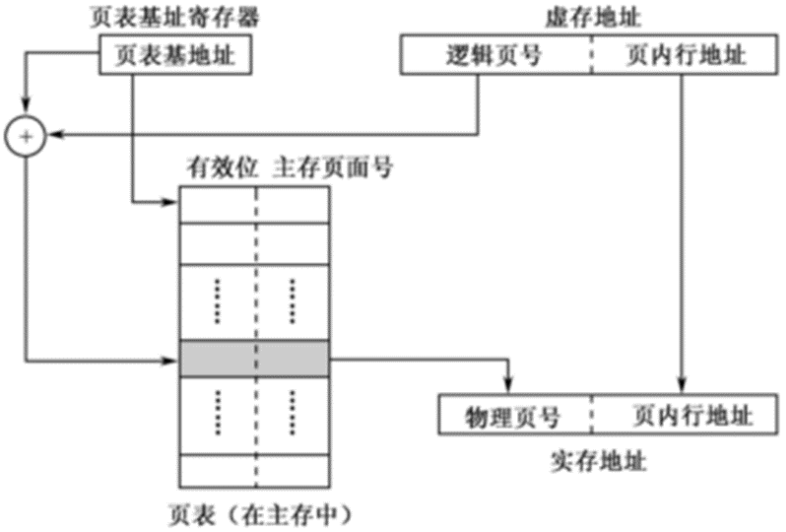
页式存地址映射

- 每个进程对应一个页表
- 页表表项内容
  - 虚存页面对应的主存页面的地址（物理页号）
  - 有效位：是否已调入主存
- 地址变换
  - 用逻辑页号作为页表内的偏移地址索引页表并找到相应物理页号，用物理页号作为实存地址的高字段，再与虚地址的页内偏移量拼接，就构成完整的物理地址
  - 现代的中央处理机通常有专门的硬件支持地址变换



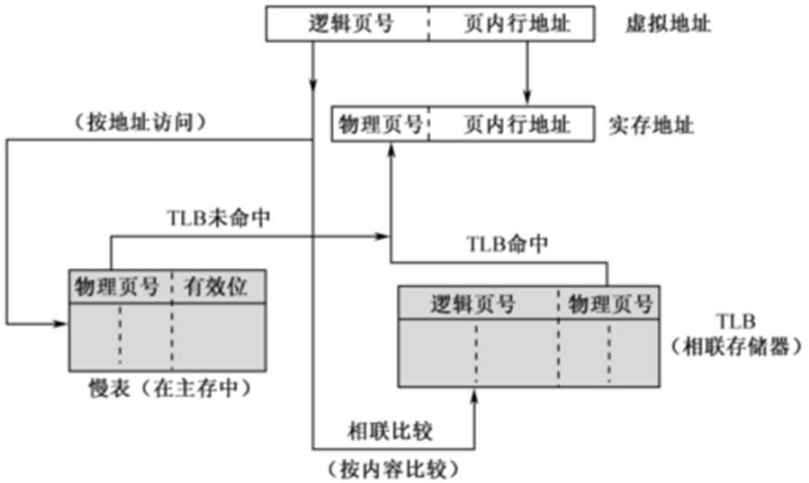
页式存地址映射

- 页表的长度是可变
  - 页表的基地址保存在寄存器中，页表本身则放在主存中
  - 由于虚存地址空间可以很大，因而每个进程的页表有可能非常长
    - 例如，如果一个进程的虚地址空间为2G字节，每页的大小为512字节，则总的虚页数为 $2^{31}/2^{9}=2^{22}$
  - 为了节省页表本身占用的主存空间，一些系统把页表存储在虚存中，因而页表本身也要进行分页



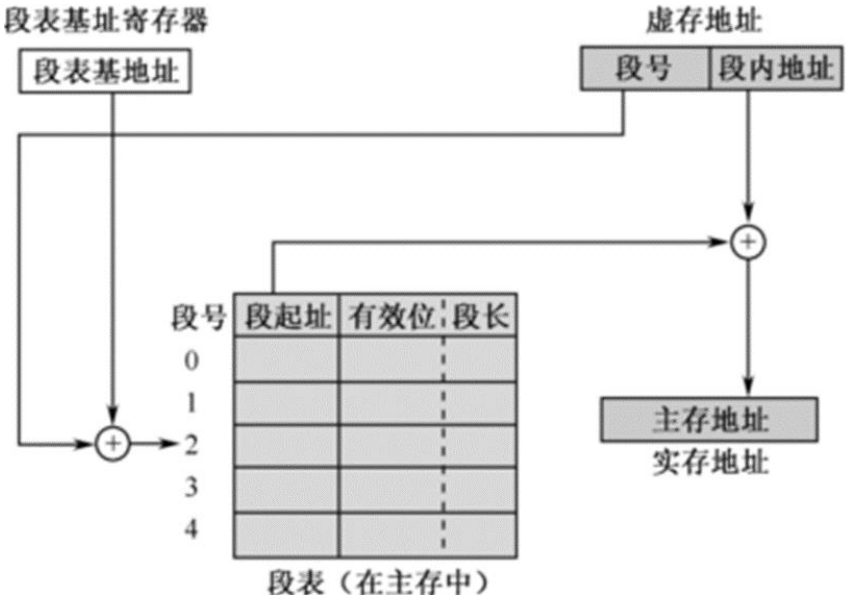
转换后援缓冲器

- 页表通常在主存中
  - 即使逻辑页已经在主存中，也至少要访问两次物理存储器才能实现一次访存（查找页表+物理地址）
  - 虚拟存储器的存取时间加倍
- 转换后援缓冲器（TLB）
  - 页表中的最活跃的部分存放在高速存储器中，组成快表
  - 保存在主存中的完整页表则称为慢表



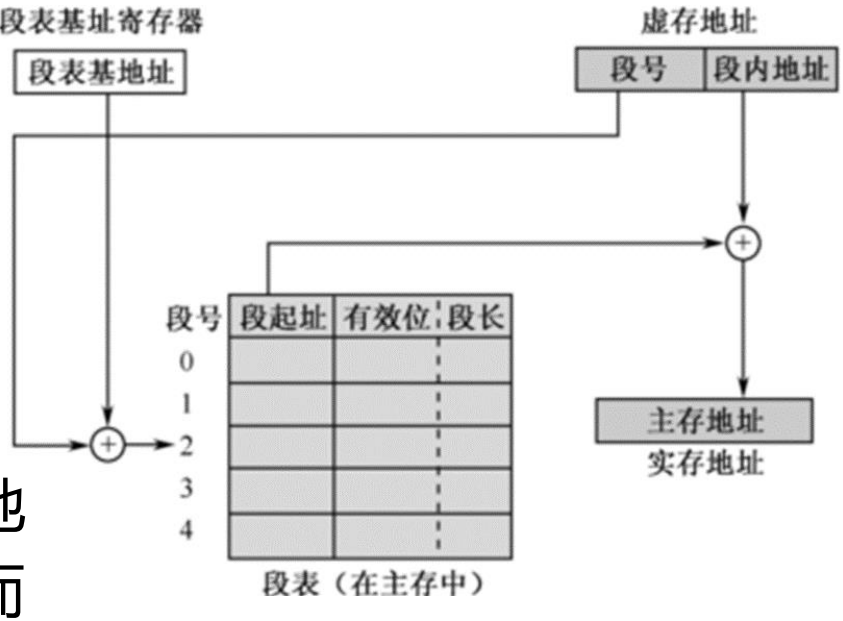
段式虚拟存储器

- 段是按照程序的自然分界划分的长度可以动态改变的区域
  - 子程序、操作数和常数等不同类型的数据划分到不同的段中
  - 每个程序可以有多个相同类型的段
- 段式虚拟存储系统
  - 虚地址由段号和段内地址（偏移量）组成
  - 虚地址到实主存地址的变换通过段表实现
  - 每个程序设置一个段表，段表的每一个表项对应一个段



段式虚拟存储器

- 表项至少包含下面三个字段：
  - 有效位：指明该段是否已经调入实存
  - 段起址：指明在该段已经调入实存的情况下，该段在实存中的首地址
  - 段长：记录该段的实际长度
- 保证访问某段的地址空间时，段内地址不会超出该段长度导致地址越界而破坏其他段
- 段表本身也是一个段，可以存在辅存中，但一般是驻留在主存中。



段式虚拟存储器的特点

- 优点
  - 段的逻辑独立性使其易于编译、管理、修改和保护，也便于多道程序共享
  - 段长可以根据需要动态改变，允许自由调度，以便有效利用主存空间
- 缺点
  - 因为段的长度不固定，主存空间分配比较麻烦
  - 容易在段间留下许多外碎片，造成存储空间利用率降低
  - 由于段长不一定是2的整数次幂，因而不能简单地像分页方式那样用虚地址和实地址的最低若干二进制位作为段内偏移量，并与段号进行直接拼接
  - 必须用加法操作通过段起址与段内偏移量的求和得到物理地址，需要更多的硬件支持

段页式虚拟存储器

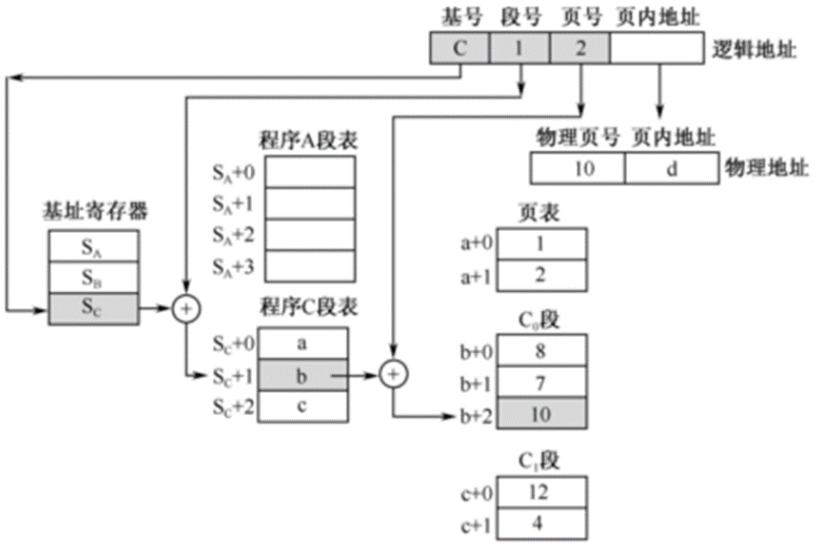
- 段页式虚拟存储器
  - 段式虚拟存储器和页式虚拟存储器的结合
  - 段号+页号+偏移量（基号—多任务操作系统）
- 实存被等分成页
  - 每个程序则先按逻辑结构分段，每段再按照实存的页大小分页，
  - 程序按页进行调入和调出操作，但可按段进行编程、保护和共享

（基号 N）	段号 S	段内逻辑页号 P	页内地址偏移量 D

段页式虚拟存储器例题（1）

[例12] 假设有三道程序，基号用A、B和C表示，其基址寄存器的内容分别为S_A、S_B和S_C。程序A由4个段构成，程序C由3个段构成。段页式虚拟存储系统的逻辑地址到物理地址的变换过程如图所示。

在主存中，每道程序都有一张段表，A程序有4段，C程序有3段，每段应有一张页表，段表的每行就表示相应页表的起始位置，而页表内的每行即为相应的物理页号。请说明虚实地址变换过程。



段页式虚拟存储器例题（2）

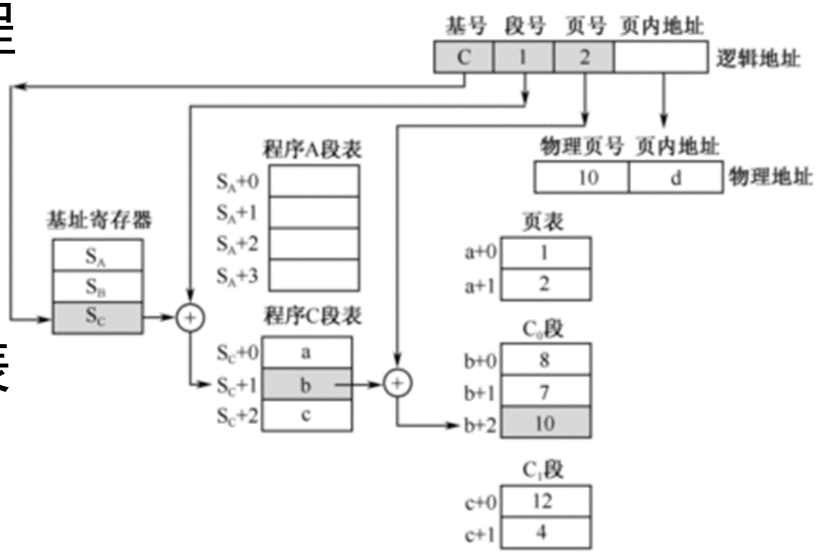
地址变换过程如下：

（1）由存储管理部件根据基号C找到段表基址寄存器表第C个表项，获得程序C的段表基址SC。再根据段号S(=1)找到程序C段表的第S个表项，得到段S的页表起始地址b。

（2）根据段内逻辑页号P(=2)检索页表，得到物理页号（图中为10）。

（3）物理页号与页内地址偏移量拼接即得物理地址

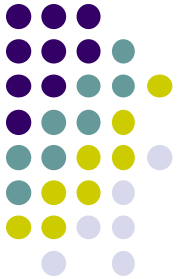
段页式虚拟存储器的缺点是在由虚地址向主存地址的映射过程中需要多次查表，复杂度较高



第三章 多层次的存储器

- 只读存储器（ROM)
- 并行存储器
- 虚拟存储器
  - 虚存地址映射方式：页式/段式/段页式
  - 虚存替换算法
- 奔腾系列机的虚存组织

虚存的替换算法



- 当从辅存调页至主存而主存已满时，虚存替换
- 虚存替换与cache替换算法类似
  - FIFO算法、LRU算法、LFU算法
- 不同点
  - Cache的替换全部靠硬件实现，而虚拟存储器的替换有操作系统的支持
  - 虚存缺页对系统性能的影响比cache未命中要大得多，因为调页需要访问辅存，并且要进行任务切换
  - 虚存页面替换的选择余地很大，属于一个进程的页面都可替换

虚存的替换算法例题

[例2] 假设主存只允许存放a、b、c三个页面，逻辑上构成a进c出的FIFO队列。某次操作中进程访存的序列是0,1,2,4,2,3,0,2,1,3,2（虚页号）。若分别采用FIFO算法、FIFO+LRU算法，请用列表法分别求两种替换策略情况下主存的命中率。

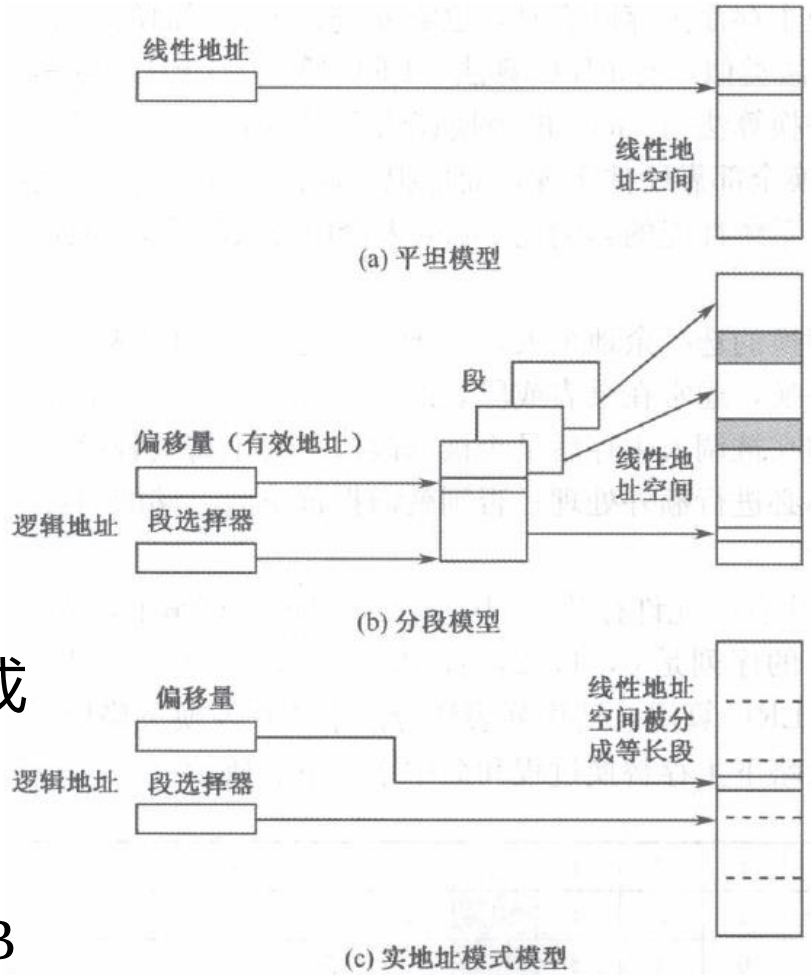
页面访问序列		0	1	2	4	②	3	0	②	1	3	②	命中率
FIFO算法	a	0	1	2	4	4	3	0	2	1	3	3	2/11=18.2%
	b		0	1	2	②	4	3	0	2	1	1	
	c			0	1	1	2	4	3	0	2	②	
						命中						命中	
FIFO+LRU算法	a	0	1	2	4	②	3	0	②	1	3	②	3/11=27.3%
	b		0	1	②	4	2	3	0	2	1	3	
	c			0	1	1	4	②	3	0	②	1	
						命中			命中			命中	

第三章 多层次的存储器

- 只读存储器（ROM)
- 并行存储器
- 虚拟存储器
- 奔腾系列机的虚存组织

存储器模型

- 平坦存储器模型
  - 内存被组织成单一、连续的地址空间，称为“线性地址空间”
  - 包括代码、数据、堆栈等
- 分段存储器模型
  - 每个程序均使用独立地址空间（段）
  - 段长最大2^{32}B
  - 逻辑地址由段选择器+偏移量组成
- 实地址模式存储器
  - 与早期8086处理机兼容模式
  - 线性地址空间被分段，段长64KB
  - 线性地址空间最大长度2^{20}B

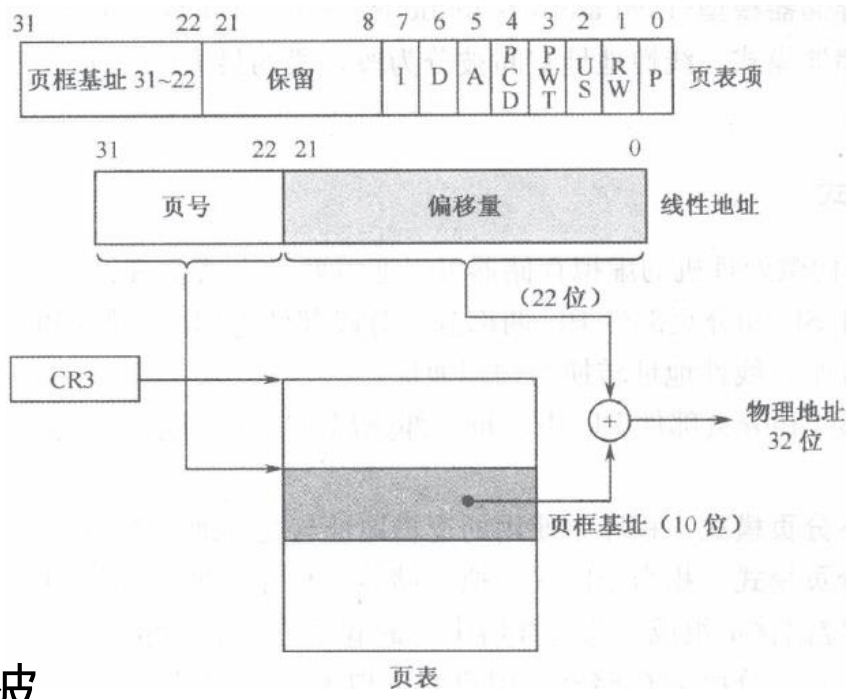


虚地址模式

- IA32体系结构微处理机的虚拟存储器方式
  - 分段和分页
  - 可独立打开或关闭
- 存储管理部件包括分段部件SU和分页部件PU两部分
  - 分段部件将程序中使用的虚地址转换成线性地址
  - 分页部件则将线性地址转换为物理地址
- 总体模式
  - 不分段不分页模式
  - 分段不分页模式
  - 不分段分页模式
  - 分段分页模式

分页模式下的地址转换

- 页大小
  - 80386/486：4KB（两级页表）
  - 奔腾处理器：4MB（单级页表）
- 地址划分（32位）
  - 低22位A0~A21：页内偏移量（4MB）
  - 高10位A22~A31：页号
- 页表项
  - I：页大小（4KB/4MB）
  - P：有效位，是否装入主存
  - A：已访问位，装入主存后是否被访问过
  - R/W：读写控制位



作业

- 第三章（2）
  - 3-8，3-9，3-13，3-15
  - 3-19，3-21，3-22