# 存储器层次结构

## 考纲要求

1. 存储的分类
2. 层次化存储器的层次化结构
3. 班都提随机存取存储器
    1. SRAM存储器
    2. DRAM存储器
    3. Flash存储器
4. 主存储器
    1. DRAM芯片和内存条
    2. 多模块存储器
    3. 主存和CPU之间的连接
5. 外部存储器
    1. 磁盘存储器
    2. 固态硬盘SSD
6. 高速缓冲存储器Cache
    1. 基本原理
    2. 和主存之间的映射方式
    3. 主存块的替换算法
    4. 写策略

## 存储器分类

1. 按存储介质分类

    1. 半导体存储器（易失）：TTL、MOS
    2. 磁表面存储器（非易失）：
        1. 磁表面存储器：磁头、磁载体
        2. 磁芯存储器：硬磁材料、环状原件
        3. 光盘存储器：激光、磁光材料

2. 按存取方式分类

    1. 存期时间与物理地址无关（随机访问）
        - 随机存储器：程序执行过程中可读可写
        - 只读存储器：程序执行过程中只读
    2. 存取时间与物理地址有关（串行访问）
        - 顺序存取存储器（磁带）
        - 直接存取存储器（磁盘）

3. 按读写功能分类

    - ROM
        - MROM，PROM，EPROM，EEPROM
    - RAM

        - 静态RAM，动态RAM

4. 按信息的可保存性分类
    - 永久性，非永久性
5. 按存储器系统种的作用分类
    - 主存，辅存，缓存，控

# 存储器分级

速度快的，价格贵、容量小；反之便宜、容量大。

CPU <-> 缓存 <=> 主存 <=> 辅存

# 主存储器技术指标

- 存放一个机器字的存储单元，是字存储单元，对应的单元地址叫字地址
- 存放一个字节的存储单元，是字节存储单元，对应的单元地址叫字节地址
- 可编址的最小单位：按字寻址、按字节寻址
- 存储容量：一个存储器种可容纳的存储单元数量
- 存取时间（访问时间）：一次读操作命令发出到该操作完成，并将数据读出到数据总线上所经历的时间。通常取写操作时间等于读操作时间，称为存储器存取时间
- 存取周期：连续两次读操作所需的最小间隔时间。通常存取周期略大于存取时间
- 存储器带宽：单位时间内存储器能够存取的信息量

**主存储器SRAM和DRAM比较**：

| **类型特点** | **SRAM** | **DRAM** |
| --- | --- | --- |
| 存储信息 | 触发器 | 电容 |
| 破坏性读出 | 非 | 是 |
| 读出后是否需要重写 | 是 | 否 |
| 运行速度 | 快 | 慢 |
| 集成度 | 低 | 高 |
| 发热量 | 低 | 高 |
| 存储成本 | 高 | 低 |
| 易失/非易失 | 易失（断电后信息消失） | 易失（断电后信息消失） |
| 是否需要刷新 | 不需要 | 需要 |
| 送行列地址 | 同时送 | 分两次送 |
| - | 常用于Cache | 常用于主存 |

# SRAM存储器

静态读写存储器SRAM的存取速度块。而DRAM则是存储容量大。
SRAM用于cache，DRAM用于主存。

任何一个内部存储器都有三组信号线：ADC（A：地址线，D：数据线，C：控制线）。
地址线A -> | 译码驱动 存储矩阵 读写电路 | <-> 数据线D
控制线C负责片选CS和读写R/W

n位地址对应2^n个存储单元。

**SRAM逻辑结构**：
大部分SRAM采用双译码方式，采用二级译码：将地址分为x向、y向。

# DRAM存储器

DRAM用于主存。每个存储单元都能够实现写1、写0、读出1、刷新存储位元的1。

为了实现行列分时传送，需要行/列地址锁存器。

## 读/写周期

读/写周期的定义是从行选通信号RAS下降沿开始，到下一个RAS信号的下降沿为止（即连续两个读周期的时间间隔）——为了控制方便，读、写周期的时间相等。

## 刷新周期

在电荷漏掉前充电，保证存储信息不被破坏——这一充电过程称为再生或刷新。
DRAM一般≤2ms会刷新一次，采用"读出"方式、按行刷新，包括有：

1. 集中刷新
    - 读/写或维持结束后，再集中执行刷新，存取周期0.5μs；
    - 以128x128矩阵为例：
        - 死区：`0.5μs × 128 = 64μs`；
        - 死亡时间率：`128/4000 × 100% = 3.2%`。
2. 分散刷新
    - 存取周期为1μs，无死区；
    - 以128x128矩阵为例：`tc = tm + tr`，即0.5μs+0.5μs。
3. 异步刷新
    - 分散刷新与集中刷新结合；
    - 对于128x128的存储芯片，存取周期为0.5μs，则每间隔15.625μs刷新一行（通常取15.5μs）；
    - 每行间隔2ms刷新一次，死区0.5μs；
    - 若将刷新安排在指令译码阶段，则不会出现死区。

## 存储器容量库充

位扩展、字扩展

**字长位数扩展（位扩展）**：
用多片给定芯片扩展字长位数。
地址线和控制线公用，而数据线D单独分开连接。
所需芯片数 = 设计要求的存储容量 ÷ 选择芯片存储器容量

**字存储容量扩展（字扩展）**：
三组信号组中，给定芯片的地址总线和数据总线公用，控制总线C中R/W公用，使能端CS不公用，它由地址总线的高位段译码来决定片选信号。
所需芯片数 = 设计要求的存储器容量 ÷ 选择瞎拍存储器容量

**字位扩展**：
实际应用中往往需要同时扩展。
若存储器容量为M×N位，使用L×K位存储器芯片扩充，则总共需要：
(M/L) × (N/K)个L×K位存储器芯片。可以先位扩展，再字扩展。

# 只读存储器ROM和闪存存储器

ROM和RAM都支持随机存取，SRAM和DRAM均为易失性半导体存储器。ROM则即使断电也不会丢失。ROM结构简单，位密度高，非易失性，可靠性高。

## ROM

1. 掩模式只读存储器MROM：
    - 存储内容固定的ROM：行列选择交叉处有MOS管为1，无MOS管为0。
2. 一次可编程只读存储器PROM：
    - 一次性编程：单个原件中，熔丝断为0、熔丝未断为1。
3. 可擦除可编程只读存储器EPROM：
    - 光擦除可编程可读存储器，多次性编程；
    - 需要更新市江源存储内容抹去，允许多次重写；
    - 但写入时间有限，不能对个别单元单独擦除或重写，擦写时间较长。
4. EEPROM:
    - 电擦除可编程存储器；
    - 出厂时存储内容全为1，使用时可根据需求吧某些存储写0。

## FLASH

Flash存储元是在EPROM和EEPROM存储元的基础上发展的。

1. 闪速存储器FlashMemory：
    - 高密度非易失性的读写存储器，存储容量大；
    - 三种操作:
        1. 编程操作：实际上是写操作；
        2. 读取操作；
        3. 擦除操作：使全部存储单元变为1状态；
2. 固态硬盘SSD：
    - 由控制单元+存储单元(Flash)构成；
    - 可进行多次快速擦除重写；

- 速度快、功耗低、价格高，目前逐渐取代机械硬盘；
- 组成：
    1. 闪存翻译层：翻译逻辑块号，找到对应页（page）；
    2. 存储介质：多个闪存芯片，每个芯片包含多个块，每个块包含多个页（page）。
- 读写特性：
    - 以页为单位读写——相当于磁盘的“扇区”
    - 以块为单位擦除
    - 支持随机访问
    - 读取快，写入较慢；如果要写的页有数据，则无法写入，需要将块内其他页全部复制到新的块中，再写入新的页
- 与机械硬盘相比：
    - 读写速度快，随机访问性能高，无需通过移动磁旋臂控制到访问位置
    - 安静无噪音，耐摔抗震，耗能低，造价贵
    - 多次擦写一个块可能会损坏
- 磨损均衡技术：
    - 将擦除的任务均分在每个块上，提升使用寿命
    - 动态磨损均衡：写入时有限选择累计擦除次数少的新的块
    - 静态磨损均衡：SSD检测并自动进行数据分配、迁移，让老旧闪存块承担以读为主的存储任务，让较新闪存块承担更多写任务

# 并行存储器

为了提高CPU和主存间的数据传输率，除了缓存外，还可以使用并行存储器。
包括有：

- 空间并行技术：双端口存储器
- 时间并行技术：多体交叉存储器

## 双端口存储器

同一个存储器用两组相互独立的读写控制电路。

**两个端口对同一主存操作有以下情况**：

1. 同时对不同地址单元存储数据。
2. 对同一地址单元读出数据。
3. 同时对同一地址单元写入数据（有冲突）。
4. 同时对同一地址单元，一个写入、另一个读出（有冲突）。

**解决方法**：

设置BUSY标志，置BUSY信号为0，由判断逻辑界定暂时关闭一个端口（被延时），未被关闭的端口正常访问，被关闭端口延长一个很短的时间段后再访问。
有冲突读写控制判断方法：

1. CE判断：若地址匹配再CE之前有效，片上的控制逻辑在CEL和CER之间判断选择端口。
2. 地址有效判断：若CE再地址匹配前遍地，片上的控制逻辑在左、右地址之间进行判断来选择端口。

## 多模块交叉模拟器

主存被分为多个相互独立、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、数据寄存器，个自己等同方式与CPU传送信息。
理想状态下（程序段或数据块都是连续地在贮存中存取）将大幅提高主存访问速度。

**高位选模块，低位选块内地址**：
当某一模块故障，其他模块可以照常工作。但由于各个模块串行工作，带宽受限。

**高位选块内地址，低位选模块**：
连续地址分布在相邻的不同模块内（同一模块内的地址都不连续）。对连续字的成块传送可实现多模块流水式并行存取，提高带宽。

# Cache

Cache采用高速SRAM组成，解决CPU和主存的速度不匹配的问题，全由硬件调度，对用户透明。

CPU与Cache之间的数据传送与字为单位，主存与Cache之间则是块为单位。
CPU读取主存使，把地址同时传送给Cache和主存：若判断地址在Cache中，则直接传送给CPU；否则用主存读周期把字送到CPU。同时把整个数据块从主存送到Cache。

## 为什么需要Cache

**程序访问局部性**：
对局部范围存储器地址频繁访问，而对此范围以外的地址则较少访问的现象称为“程序访问的局部性”。
**空间局部性**：在较近的未来要用道德信息（指令、数据），很可能与正在使用的信息在存储空间上是临近的。
**时间局部性**：在较近的未来要用到的信息很可能是现在正在使用的信息。

## Cache性能指标

**命中率h**：
在性能上使主存的平均读出时间尽可能接近Cache的读出时间，
Nc为Cache中的访问时间，Nm为在主存中的访问时间 `h = Nc / (Nc + Nm)`

**Cache-主存系统的平均访问时间ta**：
tc为命中时Cache访问时间，tm为命中时主存访问时间，则：
`ta = h*tc + (1-h)tm`

**效率e**：
效率和命中率有关，
`e = 访问Cache / 平均访问时间 * 100%`
h为Cache命中率，tc为Cache访问时间，则：
`e = tc / (h*tc+(1-h)*tm) * 100%`

## 主存与Cache的地址映射

Cache容量很小，保存的只是主存中的一个子集。Cache与主存的数据交换以块为单位。
地址映射则是将主存地址定位于Cache中。
映射时需要将主存和Cache划分围同样大小的块。

**全相连的映射方式**：
主存块放在Cache的任意位置，是一种最灵活但成本最高的方式。

Cache完全满了才会替换，需要在全局选择替换对应块。

**直接映射方式**：
映射方式一对多：`i = j mod m`。
若对应位置非空，则毫无选择地直接替换。

**组相连映射方式**：
全相连和直接映射的结合：`q = j mod u`，主存第j块的内容靠背到Cache的q组某行。
分组内满了才需要替换，需要在分组内选择替换对应块。

### Cache替换策略

从主存中调入新的字块，若其位置一杯其它字块占有，则需要替换旧字块。
最优情况是使被替换的字块是下一段时间内估计最少使用的。

常见替换算法有：先进先出FIFO，近期最少使用LRU，最不经常使用LFU，随机替换。

**先进先出FIFO**：
把最先调入Cache的字块替换，不记录各字块使用情况，实现容易，开销小。

**近期最少使用LRU**：
被访问的行数计数器置0，其它行的计数器+1。近期最少使用的字块被替换（计数器值最大的行）。符合Cache工作原理。

**最不经常使用LFU**：
被访问的行进行计数器+1，同一时间访问次数最少的数据被替换。不能访问近期Cache的访问情况

**随机替换**：
随机选取一行换出。

### Cache写操作策略

- **写命中**
  - *全写法*
    - 数据必须同时写入Cache和主存，一般使用写缓冲。访问次数增加，速度慢，但更保证数据一致。
  - *写回法*
    - 只修改Cache内容，不立即写入主存，只有被换出才写回主存。减少了访问次数，但存在数据不一致的隐患。
- **写不命中**
  - *写分配法*
    - 把主存快调入Cache，在Cache中修改。通常搭配写回法。
  - *非写分配法*
    - 只写入主存，不调入Cache。通常搭配全写法，只有读未命中时调入Cache。

## 虚拟存储器

将主存和辅存的地址空间统一编址，合成一个更大的地址空间。

虚存的替换算法和Cache类似，也有先进先出FIFO，近期最少使用LRU，最不经常使用LFU，随机替换等。

- 虚地址：用户编程时用的地址称为虚拟地址和逻辑地址，对应的存储空间称为虚存空间或逻辑地址空间。
- 实地址：物理内存访问的地址，其对应存储空间是物理存储空间或主存空间。
- 程序的再定位：程序进行虚地址到实地址转换的过程。
- 虚存的访问过程：由地址变换机构依据当时分配给程序的实地址空间，把程序的一部分调入实存。每次访问时都将判断虚地址对应部分是否在实存中：是，则地址转换并用实地址访问主存；否，则按指定算法将辅存中的部分程序调度进内存，再按同样方式访问主存。
- Cache与虚存的异同：
    - 主存-辅存的访问机制和Cache-主存访问机制类似。这是由Cache、主存、辅存构成的三级存储体系中的两个层次。
    - Cache-主存构成了系统内存；主存-辅存依靠软硬件的支持构成了虚拟存储器

### 页式虚拟存储器

**页式虚存地址映射**：
虚地址空间和主存空间分别分为等长大小的页（逻辑页和物理页）。
虚地址分为两个字段：高字段为逻辑页号，低字段为页内地址（偏移量）；
实存地址分为两个字段：高字段为物理页号，地字段为页内地址。
通过页表把虚地址转为物理地址。

**页式虚拟存储器的地址映射过程**：
每个进程对应一个页表，表项的内容包含该虚存页面所在的主存页面的物理页号。页表的基地址存在寄存器，页表本身放在主存。

**快/慢表**：
页表通常在主存，即使逻辑页在主存，也至少访问两次物理存储器才实现一次访存。
可对页表本身实行二级缓存，把页表中最活跃的部分放在高速存储器，组成**快表**。
专用于页表缓存的高速存储部件通常称为转换后援缓冲器TLB。存在主存里完整的页表则是**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。
通常把子程序、操作数、常数等不同类型数据分到不同的段，每个程序可以有多个相同类型的段。

虚地址由段号和段内地址（偏移量）组成。虚地址到实地址主存地址的变换通过段表实现。

每个程序设置一个段表，每一个表项对应一个段。每个表项包含：有效位，段起止，段长。

### 段页式虚拟存储器

页式和段式的结合。
实存被分为页，每个程序先按逻辑结构分段，每段再按实存的页大小来分页，程序按页进行调入调出操作，但可按段进行编程、保护、共享。

## 错题补充

- 不能采取随机存取的存储器：CD-ROM

- 存取周期为200ns的64K×32位的四体并行低位交叉存储器，在200ns内，存储器能够向CPU提供128位的二进制信息（每个模块提供32位）

- 对于四体低位交叉存储器，读取6个连续地址单元中的存储字，需要2个完整的存储周期（同理读取3个连续地址单元的，需要1个完整的周期）

- 4M×8位的DRAM芯片，其地址引脚数是11、数据引脚数是8。
  DRAM的特点是地址复用，所以在计算地址引脚数时需要÷2（对于4M，有`log2(4*1024*1024) = 22`，所以地址引脚数是11；
  数据引脚数就是该芯片的位数。

- 使用四体交叉编制存储器，存储器总线上出现的主存地址序列为：8005、8006、8007、8008、8001、8002、8003、8004、8000。可能发生访问冲突的是8000和8004。
  对各个地址做取余处理得到下表：

| - | - | - | - |
|---|---|---|---|
| - | 5 | 6 | 7 |
| 8 | 1 | 2 | 3 |
| 4 | - | - | - |
| 0 | - | - | - |

- 计算机主存按字节编址，由4个64M×8位的DRAM芯片采用交叉编址方式构成，并与宽度为32位的存储器总线相连，主存每次最多读写32位数据。若double型变量x的主存地址为804001AH，则读取x需要的存储周期数是3。
  x的起始位是A(1010)，所以对于该四体交叉编制存储器，x从第3个开始；而double占据8个字节。所以综上存储周期为3。

- 对于一个`8192*8192*8位`的DRAM芯片，芯片内缓冲有8192*8位。

- 用若干2K×4位的芯片组成一个8K×8位的存储器，则地址0BIFH所在的芯片最小地址是什么？
  8K = 2^13，所以要13根地址线；单个芯片2K需要11根地址线；
  所以前2位代表4组选片地址，后11位代表片内地址；
  所以对于`0BIDH = 000(0 1)(011 0001 1111)`，其所在的最小地址为`0000 1000 0000 0000 = 0800H`。