

6.存储器层次结构

6.1 存储技术

6.1.1 随机访问存储器

随机访问存储器(Random-Access Memory, RAM) 分为两类：静态和动态。静态SRAM比动态DRAM更快。SRAM用来作为高速缓存存储器，既可以在CPU芯片上，也可以在片下。DRAM用来作为主存以及图形系统的帧缓冲区。

1.SRAM

SRAM将每个为存储在一个双稳态的存储器单元里。每个单元用一个晶体管电路来实现。
由于SRAM存储器单元的双稳态特性，只要有电，他就会永远保持它的值。

2.DRAM

DRAM将每个位存储为一个电容的充电。DRAM存储单元对干扰非常敏感。内存系统必须周期性地读出，然后重写来刷新内存每一位。有些系统使用纠错码。

3.传统的DRAM

DRAM芯片中的单元（位）被分成d个超单元，每个超单元都由w个DRAM单元组成。一个dxw地DRAM总共存储了dw位信息。

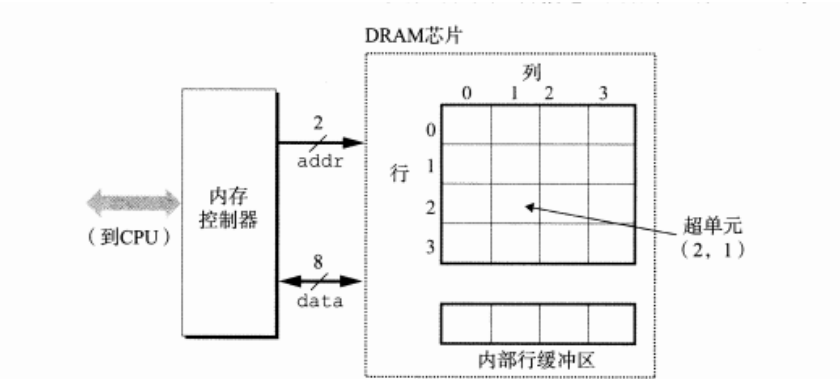
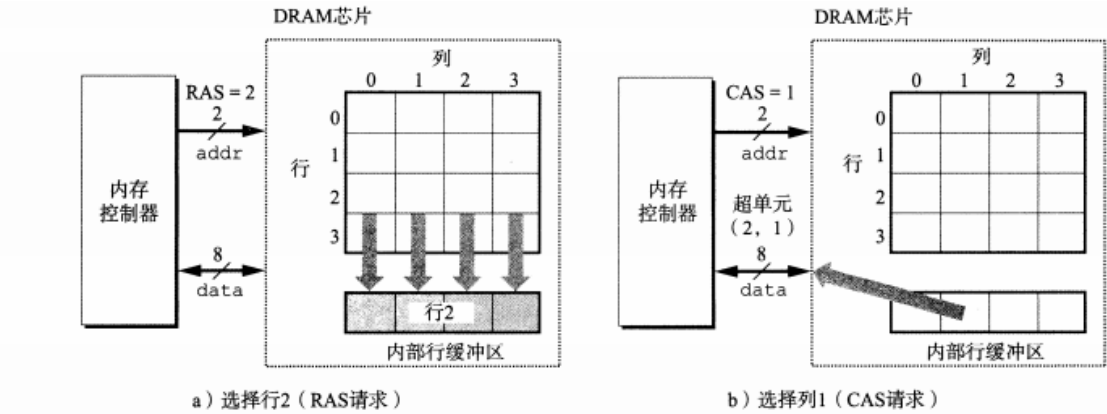


图 6-3 一个 128 位 16x8 的 DRAM 芯片的高级视图

获取一个超单元的过程：

内存控制器先发送行地址，然后再发送列地址。DRAM把超单元 (i, j) 地内容发回给控制器作为响应。
在内部相应行缓冲区中一次将一行全部取出。



4.内存模块

每个超单元存储主存的一个字节，而用相应的超单元地址为 (i, j) 的8个超单元来表示主存中字节地址A处的64位字。
要取出内存地址A处的一个字，内存控制器将A转换成一个超单元地址 (i, j) ，并将它发哦是那个到内存模块，然后内存模块再将i和j广播到每个DRAM。作为响应，每个DRAM输出他的 (i, j) 超单元的8位内容。模块中的电路收集这些输出，并把它们合并成一个64位字，再返回给内存控制器。

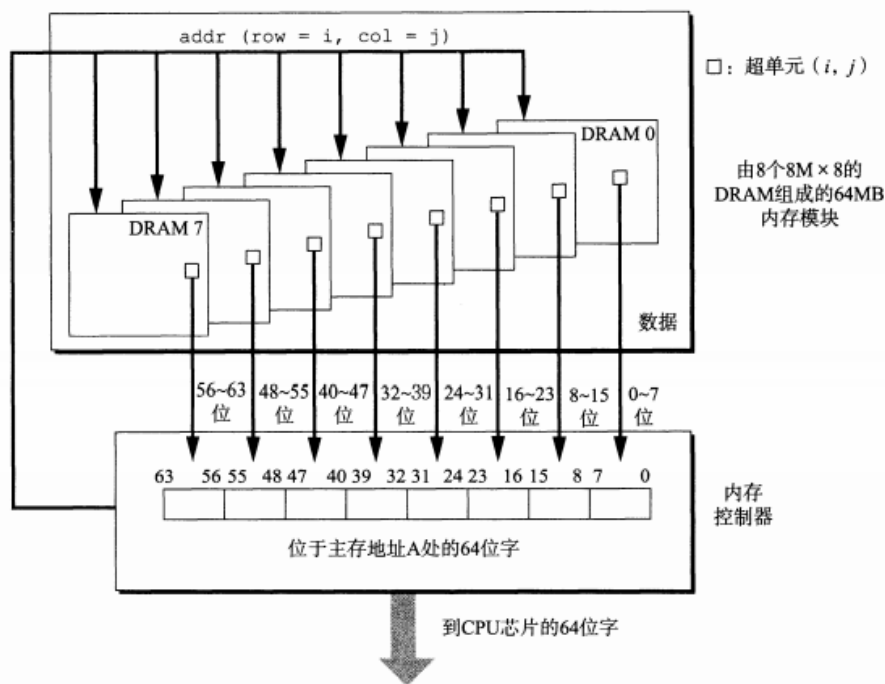


图 6-5 读一个内存模块的内容

5.增强的DRAM

- ①快页模式DRAM：一次取多个同行不同列的超单元；
- ②扩展数据输出DRAM，是FPM DRAM的增强形式，允许各个CAS信号在时间上更紧密一些。
- ③同步DRAM (SDRAM) 能比异步的存储器更快地输出它的超单元的内容。
- ④双倍数据率同步DRAM (DDR SDRAM)
- ⑤视频RAM(VRAM)写的同时用帧缓冲区中的像素刷新屏幕（读）。

6.非易失性存储器

ROM只读存储器。

可编程ROM(PROM)只能被编程一次。（熔丝只能用高电流熔断一次）

7.访问主存

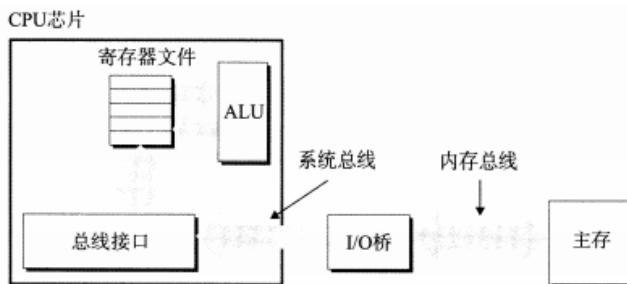


图 6-6 连接 CPU 和主存的总线结构示例

6.1.2 磁盘存储

1.磁盘构造

磁盘由盘片构成，每个盘片有两面或者称为表面，每个表面是由一组称为磁道的同心圆组成的，每个磁道被划分为一组扇区。每个扇区包含相等数量的相同的数据位。扇区之间由一些间隙分隔开，间隙中不存储数据。

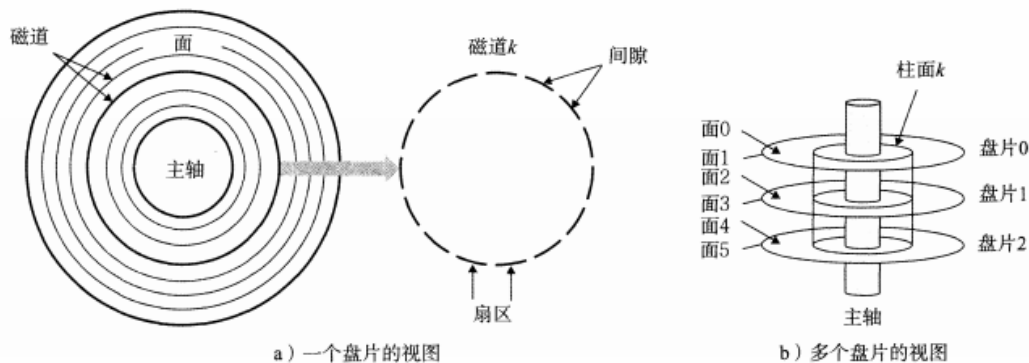


图 6-9 磁盘构造

2. 磁盘容量

记录密度：磁道一英尺的段中可以放入的位数；

磁道密度：从盘片中心出发半径上一英寸的段内可以有的磁道数；

面密度：记录密度与磁道密度的乘积。

为了让扇区容纳的数据量更合理，柱面的集合被分割成不相交的子集合，称为记录区，每个区包含一组连续的柱面，一个区中的每个柱面的每条磁道都有相同数量的扇区，由最里面的磁道所能包含的扇区数决定。

下面的公式给出了一个磁盘的容量：

$$\text{磁盘容量} = \frac{\text{字节数}}{\text{扇区}} \times \frac{\text{平均扇区数}}{\text{磁道}} \times \frac{\text{磁道数}}{\text{表面}} \times \frac{\text{表面数}}{\text{盘片}} \times \frac{\text{盘片数}}{\text{磁盘}}$$

例如，假设我们有一个磁盘，有 5 个盘片，每个扇区 512 个字节，每个面 20 000 条磁道，每条磁道平均 300 个扇区。那么这个磁盘的容量是：

$$\begin{aligned} \text{磁盘容量} &= \frac{512 \text{ 字节}}{\text{扇区}} \times \frac{300 \text{ 扇区}}{\text{磁道}} \times \frac{20\,000 \text{ 磁道}}{\text{表面}} \times \frac{2 \text{ 表面}}{\text{盘片}} \times \frac{5 \text{ 盘片}}{\text{磁盘}} \\ &= 30\,720\,000\,000 \text{ 字节} \\ &= 30.72 \text{ GB} \end{aligned}$$

注意，制造商是以千兆字节 (GB) 或兆兆字节 (TB) 为单位来表达磁盘容量的，这里 $1\text{GB}=10^9$ 字节， $1\text{TB}=10^{12}$ 字节。

3. 磁盘操作

磁盘以扇区大小的块来读写数据。对扇区的访问时间有三个主要的部分：寻道时间、旋转时间和传送时间。

寻道时间：为了读取某个目标扇区的内容。传动臂将读写头定位到目标扇区的磁道上的时间。

旋转时间：等待目标扇区的第一个位旋转到读写头下方的时间。最坏的情况是刚转走而要等待一周。

$$T_{\text{max rotation}} = \frac{1}{\text{RPM}} \times \frac{60\text{s}}{1\text{min}}$$

平均旋转时间 $T_{\text{avg rotation}}$ 是 $T_{\text{max rotation}}$ 的一半。

$$T_{\text{avg transfer}} = \frac{1}{\text{RPM}} \times \frac{1}{(\text{平均扇区数} / \text{磁道})} \times \frac{60\text{s}}{1\text{min}}$$

传送时间：

例子：

参数	值
旋转速率	7200RPM
$T_{\text{avg seek}}$	9 ms
每条磁道的平均扇区数	400

对于这个磁盘，平均旋转延迟 (以 ms 为单位) 是

$$T_{\text{avg rotation}} = 1/2 \times T_{\text{max rotation}} = 1/2 \times (60\text{s}/7200 \text{ RPM}) \times 1000 \text{ ms/s} \approx 4 \text{ ms}$$

平均传送时间是

$$T_{\text{avg transfer}} = 60/7200 \text{ RPM} \times 1/400 \text{ 扇区} / \text{磁道} \times 1000 \text{ ms/s} \approx 0.02 \text{ ms}$$

总之，整个估计的访问时间是

$$T_{\text{access}} = T_{\text{avg seek}} + T_{\text{avg rotation}} + T_{\text{avg transfer}} = 9 \text{ ms} + 4 \text{ ms} + 0.02 \text{ ms} = 13.02 \text{ ms}$$

4. 逻辑磁盘块

将扇区逻辑化为 B 个扇区大小的逻辑块序列。控制器上的固件执行一个快速查找表，将一个逻辑块号翻译成一个 (盘面，磁道，扇区) 的三元组。

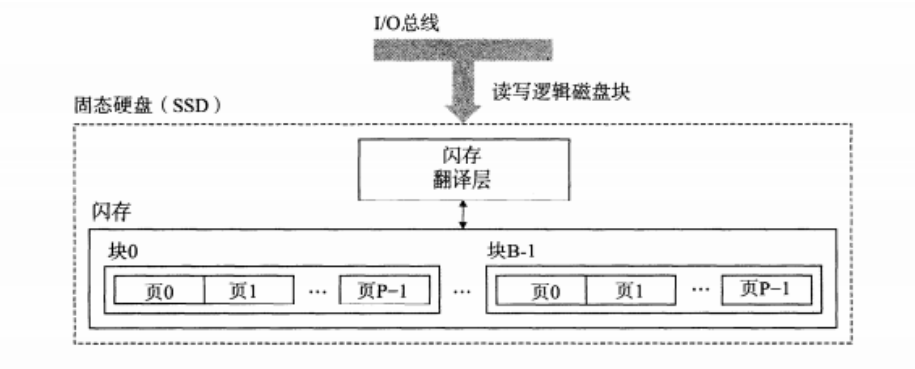
5. 连接 I/O 设备

6.访问磁盘

设备可以自己执行读或写总线事务而不需要CPU干涉的过程，称为直接内存访问(DMA)。这种数据传送称为DMA传送。

6.1.3 固态硬盘

SSD是一种基于山村的存储技术。



随机读比随机写性能好。SSD一个潜在的缺陷是底层闪存会磨损。

6.2 局部性

6.2.1 对程序数据引用的局部性

每个k个元素进行访问，就称步长为k的引用模式。随着步长的增加，空间局部性下降。

6.2.2 取指令的空部性

循环体越小，循环迭代次数越多，局部性越好。

6.3 存储器体系结构

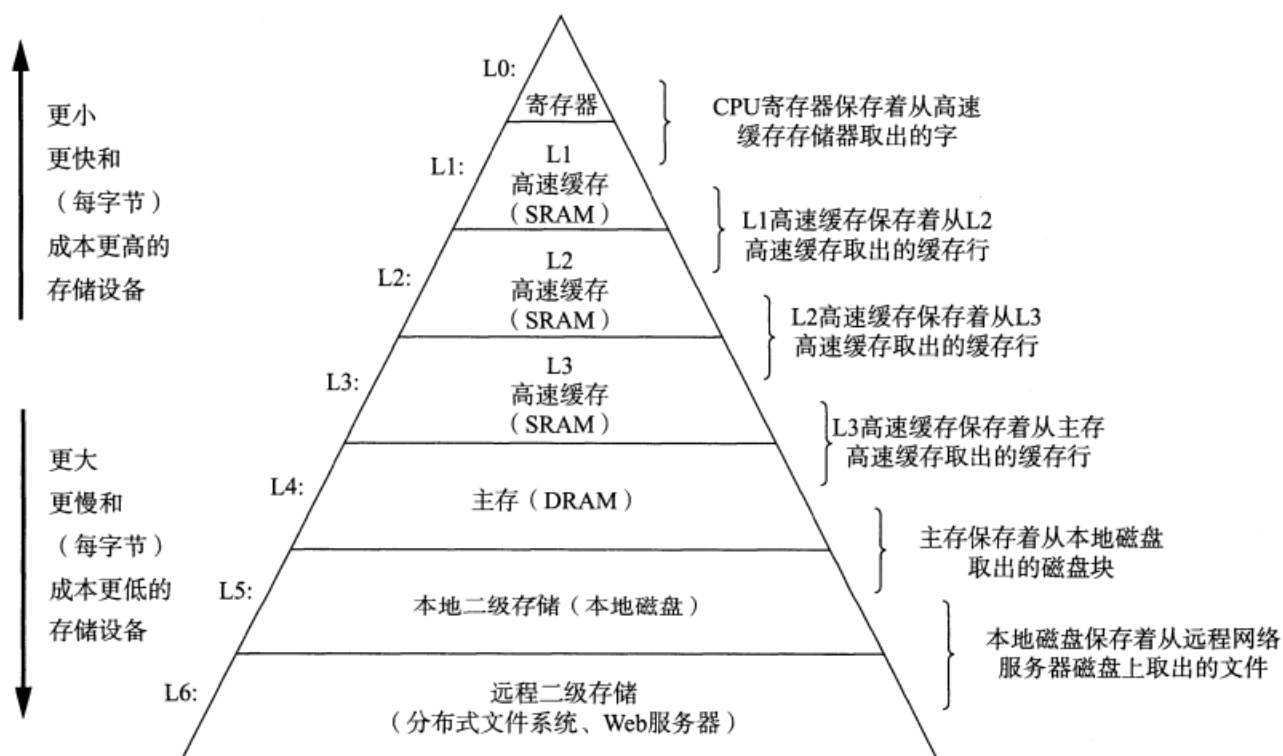


图 6-21 存储器层次结构

6.3.1 存储器层次结构中的缓存

1. 缓存命中
2. 缓存不命中
3. 缓存不命中的种类

一个空的缓存被称为冷缓存，此类不命中称为强制性不命中或冷不命中。

限制性的放置策略引起的不命中称为冲突不命中。

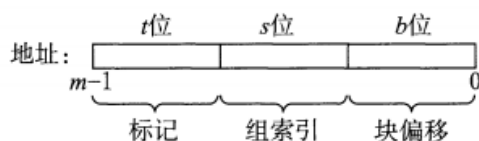
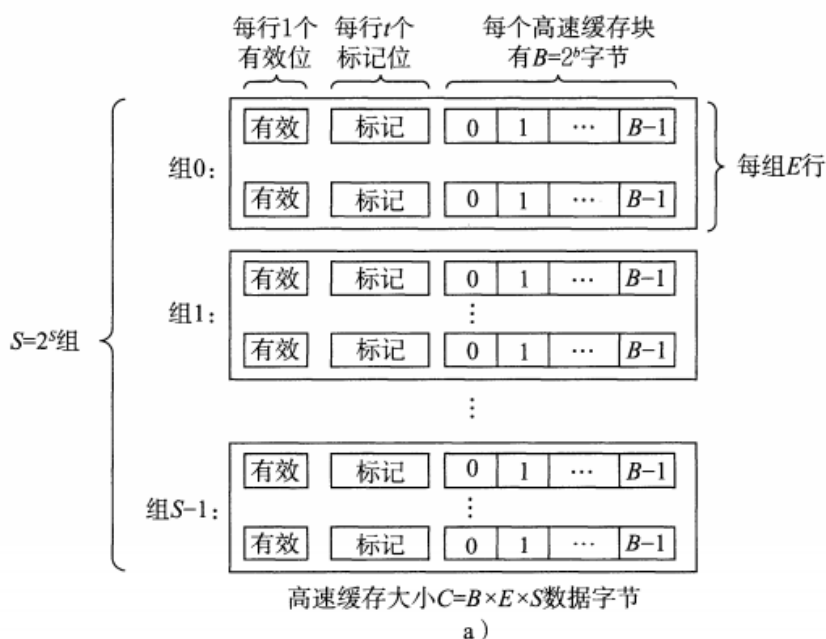
当工作集的大小超过缓存的大小时，缓存就会经历容量不命中。

4. 缓存管理

编译器管理寄存器文件，L1、L2、L3层的缓存完全由缓存中的硬件逻辑来管理。DRAM主存由OS和CPU上的地址翻译硬件共同管理。

6.4 高速缓存存储器

6.4.1 通用的高速缓存存储器组织结构



高速缓存的结构用元组 (S, E, B, m) 来描述。(S=2的 s 次方, 为组的数量, 每组包含E个高速缓存行, 每行由 $B=2$ 的 b 次方个数据块组成, $t=m-b-s$ 个标记位) C是所有块的大小的和。标记为和有效位不包括在内。C=SEXEB
组索引在中间, 标记在最高位, 不然多路时连续的内存块会被划分为同一组

6.4.2 直接映射高速缓存

每组一行。

抽取出被请求的字有步骤: 1.组选择 2.行匹配 3.字抽取

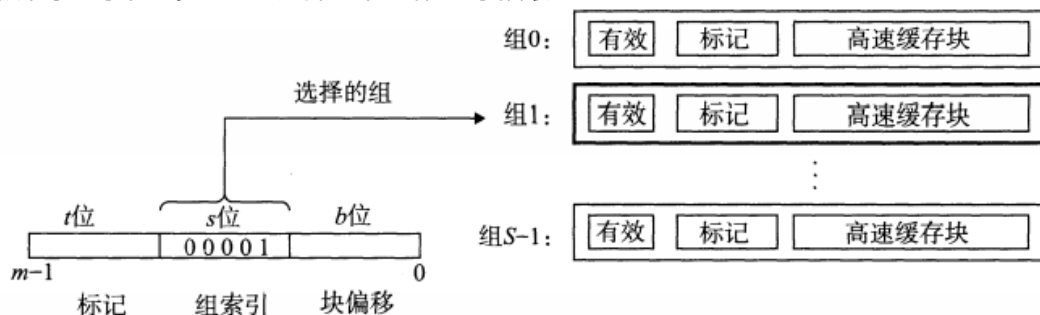


图 6-28 直接映射高速缓存中的组选择

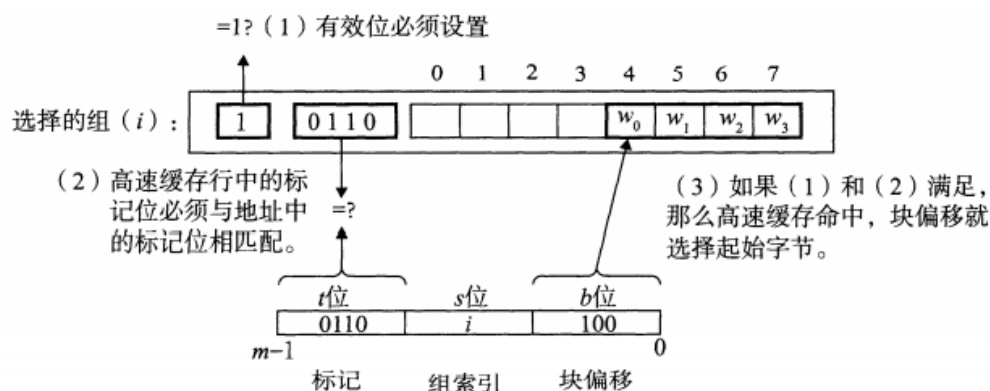


图 6-29 直接映射高速缓存中的行匹配和字选择。在高速缓存块中, w_0 表示字 w 的低位字节, w_1 是下一个字节, 依此类推

6.4.3 组相联高速缓存

6.4.4 全相联高速缓存

没有组索引。

6.4.5 有关写的问题

直写：立即将w的高速缓存块写回到紧接着的低一层中

写回：等到替换算法要驱逐这个更新过的块时再写回。缺点是要增加一个修改位。

处理写不命中：

写分配：从低一层加载到高速缓存中。

非写分配：避开高速缓存，直接把这个字写到低一层中。

6.4.7 高速缓存参数的性能影响

不命中率、命中率、命中时间、不命中处罚。