

存储系统

分类

按层次分 { 主存
外存
Cache

按存储介质分 { 磁表面存储器 (磁盘、磁带)
磁芯存储器
半导体存储器 (MOS型、双极型)
光存储器 (光盘)

按存取方式 { RAM { 静态
动态
ROM
串行访问存储器 { 顺序存取 (磁带)
直接存取 (磁盘、光盘)

性能指标

存储容量 = 存储字数 \times 字长

单位成本 每位价格 = 总成本 / 总容量

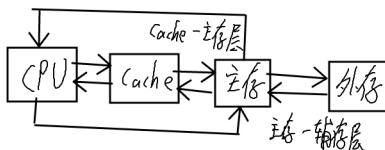
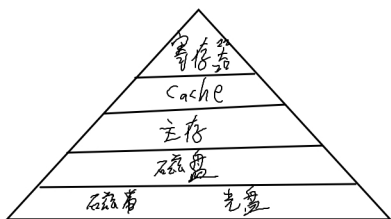
存储速度 数据传输率 = 数据宽度 / 存取周期

存取时间 T_a

存取周期 (读写周期) T_m

主存带宽 (数据传输率) B_m bit/s B/s 字/s

启动存取 { 存取完 { 不记存取
 T_a 恢复时间
 T_m



SRAM 与 DRAM

SRAM

存储元	触发器
破坏性读出	否
刷新	不要
送行地址	同时送
运行速度	快
集成度	低
存储成本	高
主要用途	Cache

DRAM

电容

是

要

分两边送

慢

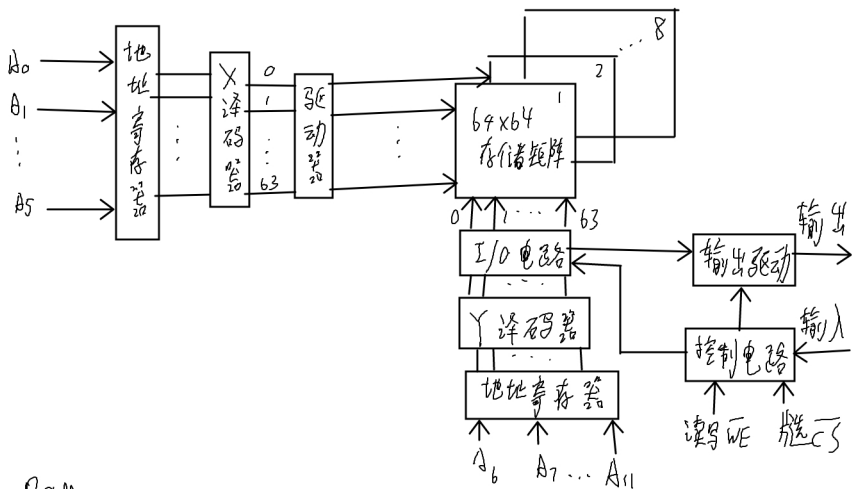
高

低

主存

刷新周期通常 2ms

方式 { 集中刷新
分散刷新
异步刷新



ROM

非易失性

ROM

MRom

PRom

EProm

Flash

SSD

掩模式 ROM

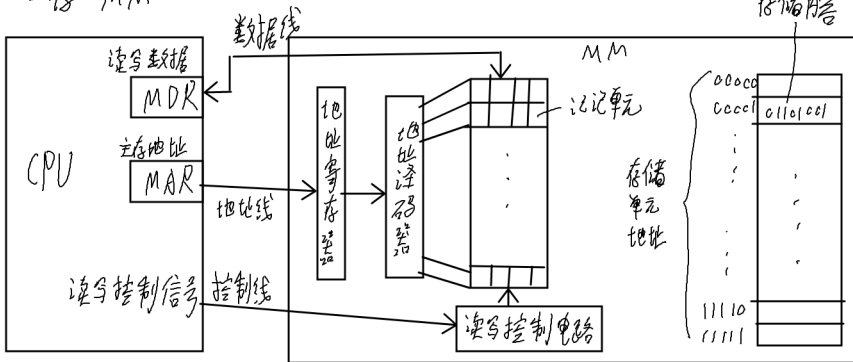
一次可编程 ROM

可擦除可编程 ROM

固态硬盘

主存 MM

存储单元

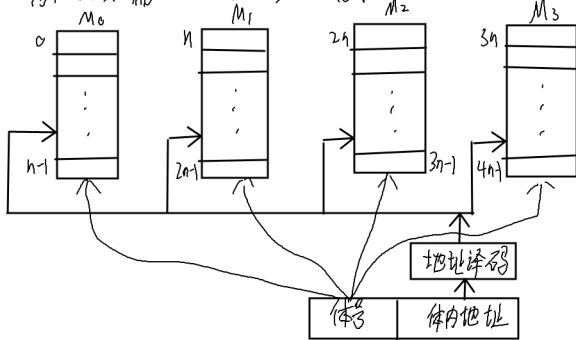


DRAM 行列分两次输入，地址引脚 = 地址线/2 (地址引脚复用)

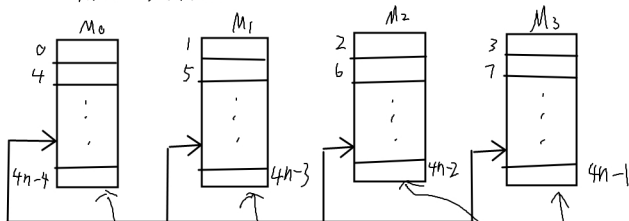
多模块存储器 { 单体多字
多体并行 { 高位交叉
空间并行技术 { 低位交叉
多体并行存储器
既能并行又能交叉工作

高位交叉编址(顺序方式)

存取方式仍是串行，不能提高吞吐量



低位交叉编址 (交叉式)

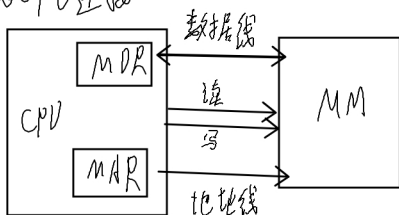


按模 m 交叉编址

模块号 = 单元地址 $\% m$

如 $0, m, \dots, (k-1)m$ 位于 M_0 .

主存与CPU连接



主存容量扩充 { 位扩展
字扩展
字位同时扩展

位扩展法

地址端、片选端、读写控制端 并联

数据端 分列引出

例 8片 $8K \times 1$ 组成 $8K \times 8$

字扩展法

地址线、数据线、读写控制线 并联

片选信号控制地址范围

例 4片 $16K \times 8$ 组成 $64K \times 8$

字位同时扩展法

例 8片 $16K \times 4$ 组成 $64K \times 8$

先每2片一组位扩展
 $16K \times 4 \rightarrow 16K \times 8$
再每4片一组字扩展
 $16K \times 8 \rightarrow 64K \times 8$

地址分配与译码

片选信号的产生

线选法

译码片选法

线选法

例 4片 $2K \times 8$ 组成 $8K \times 8$

$A_4 \sim A_{11}$ 组成片选信号

芯片	$A_{14} \sim A_{11}$
0#	1110
1#	1101
2#	1011
3#	0111

译码片选法

例 8片 8Kx8 组成 64Kx8

$A_{15}A_{14}A_{13} \begin{cases} 000 \rightarrow \#0 \\ 001 \rightarrow \#1 \\ \vdots \\ 111 \rightarrow \#7 \end{cases}$

外存

磁盘

存储地址 $\begin{cases} \text{磁头数 (记录面数)} \\ \text{柱面数 (磁道数)} \\ \text{扇区数 (块数)} \end{cases}$

磁盘按块存取

性能指标 $\begin{cases} \text{记录密度} \\ \text{磁盘容量} \end{cases}$

平均存取时间 = 寻道时间 + 旋转延迟时间 + 传输时间

数据传输率

$$D_r = rN$$

磁盘 r 转/s

每磁道 N 字节

磁盘地址

例 系统有4个驱动器, 每个驱动器带一个磁盘,
每个磁盘 256个磁道、16个盘面, 每个盘面 16个扇区
每个扇区要 18位二进制数编码

驱动器号 (2位)	柱面号 (8位)	盘面号 (4位)	扇区号 (4位)
-----------	----------	----------	----------

读写操作串行

磁盘阵列 RAID

多个独立物理磁盘组成一个独立逻辑盘

提高可靠性方式 { 镜像
校验

Cache

局部性原理 { 空间局部性 当前用的信息附近的信息可能还会用
时间局部性 当前用的信息可能不久后就要再用

CPU 读 { ① Cache 命中, 从 Cache 中读
② 未命中, 读的字所在的块(行)从主存调入 cache
③ Cache 满, 使用替换策略

Cache 与主存映射 (地址映射) { 直接映射
全相联映射
组相联映射

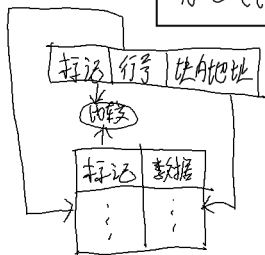
Cache 有 n 个有效位

直接映射

Cache 行号 = 主存块号 % 总行数 主存 2^n 块

Cache 总行数 = Cache 容量 / 块大小 $m = t \times c$

标记 (t 位)	行号 (c 位)	块内地址 (b 位)
----------	----------	------------



全相联映射

标记	块内地址
----	------

组相联映射

Cache 组号 = 主存块号 / 组数 Q

标记	组号	块内地址
----	----	------

替换算法 {
Rand
FIFO
LRU
LFU

写策略

Cache 命中处理方法 { 全写法
回写法

设置一位脏位

写不命中处理 { 写分配法
非写分配法

通常 写分配与回写 合用

非写分配与全写 合用

虚拟存储器

回写处理一致性问题

全相联映射

局部性原理

由OS和硬件共同实现

页式虚拟存储器

页表

有效位	脏位	引用位	物理页或磁盘地址
-----	----	-----	----------

页表寄存器

页表基址

帧号 (有效位)	物理页号

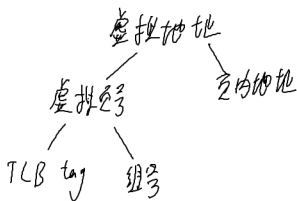
虚拟页号 | 页内地址 | 虚拟地址

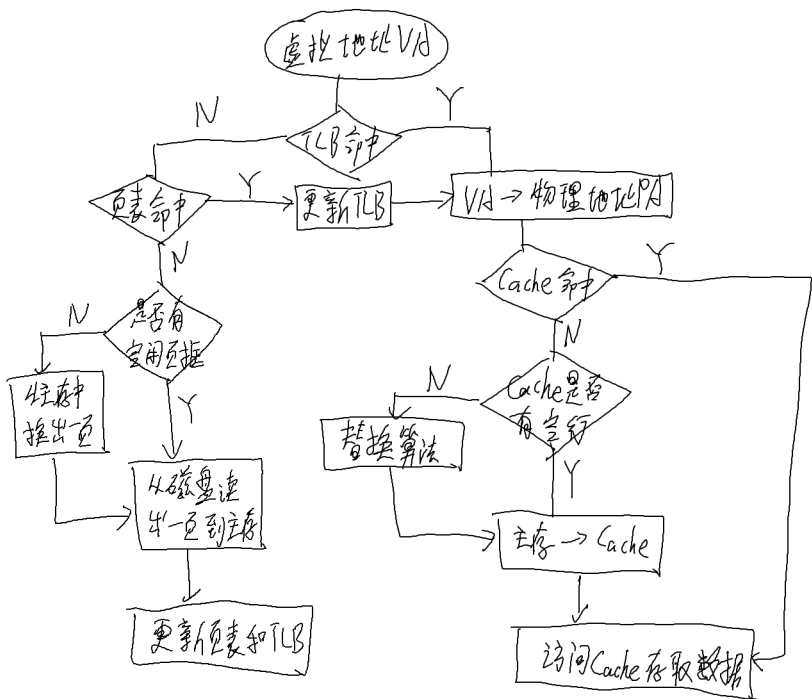
物理页号 | 页内地址 | 物理地址

页表在内存中

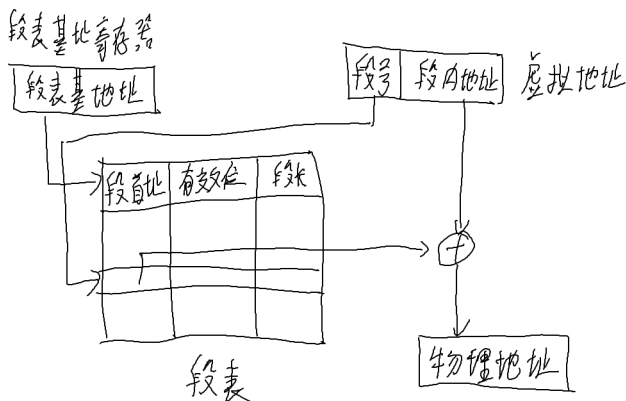
TLB快表

tag	有效位	物理页号
-----	-----	------





段表虚拟存储器



段页式虚拟存储器

先分段后分页

