

计算机组成与体系结构 非官方复习提要（1）

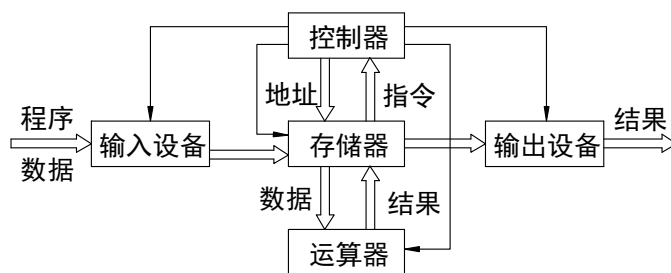
2015 年 4 月

※ 重要声明

1. 本资料不是官方复习资料，仅供自己参考使用，严禁用于非法用途；
2. 欢迎大家就本资料中未涉及的、讲解不清的地方进行修改与补充，别忘了认真看书。

第一章 绪论

1 Von Neumann 机的组成



2 Flynn 分类法

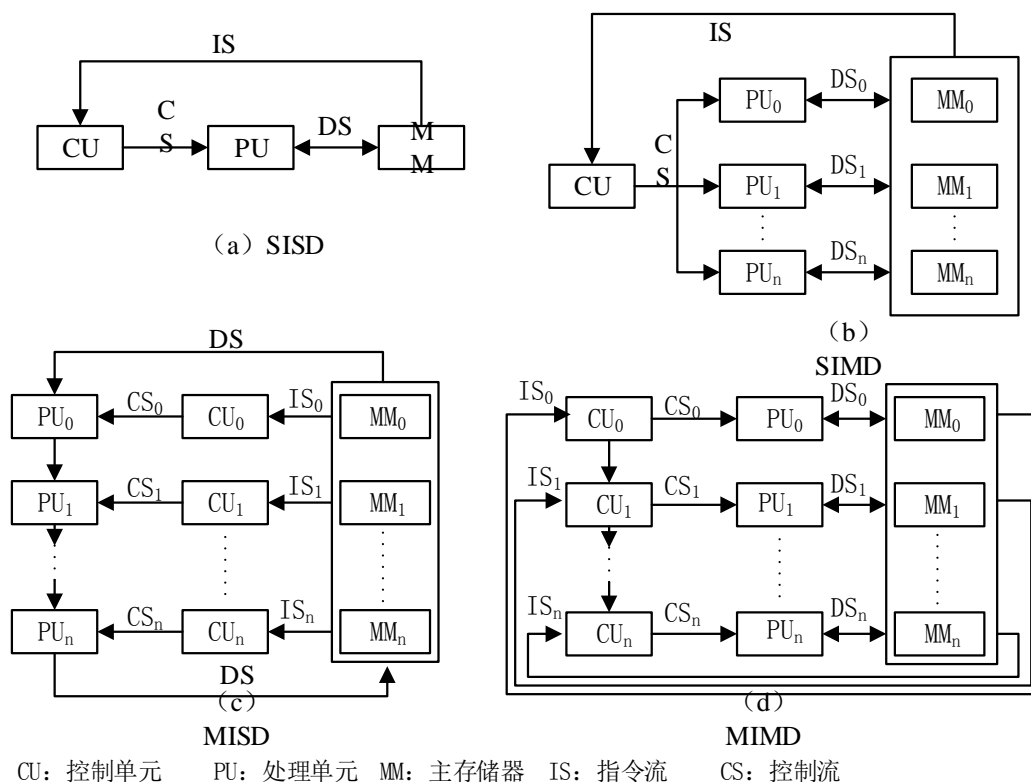


图1.5 Flynn分类法各类计算机示意图

3 Amdahl's Law

$$\text{加速比} = \frac{1}{\text{未改进部分的比例} + \sum_{k=1}^n \frac{\text{第}k\text{改进部分的比例}}{\text{第}k\text{改进部分的提升}}}$$

第二章 计算机系统的数据表示

4 原码、反码、补码和移码（以八位有符号整数表示为例）

X	原码	反码	补码	移码
-128	N/A	N/A	1 000 0000	0 000 0000
-127	1 111 1111	1 000 0000	1 000 0001	0 000 0001
.....				
-12	1 000 1010	1 111 0101	1 111 0110	0 111 0110
.....				
-1	1 000 0001	1 111 1110	1 111 1111	0 111 1111
-0	1 000 0000	1 111 1111	0 000 0000	1 000 0000
+0	0 000 0000	0 000 0000		
+1	0 000 0001	0 000 0001	0 000 0001	1 000 0001
.....				
+12	0 000 1010	0 000 1010	0 000 1010	1 000 1010
.....				
+127	0 111 1111	0 111 1111	0 111 1111	1 111 1111
表示范围	-127 到 +127	-127 到 +127	-128 到 +127	-128 到 +127

补码的移位性质

算术右移 k 位：除 2^k 运算，高位用符号位填充；

算术左移 k 位：乘 2^k 运算，低位用 0 填充，注意可能发生溢出，可通过变形补码（两位符号位）判断。

5 非数值数据的编码

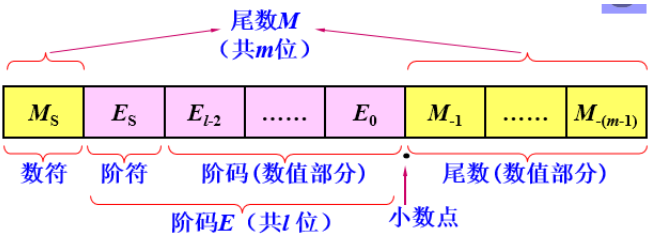
BCD 码：8421 码、余 3 码（8421 码+3）、Grey 码，注意是否压缩处理。

ASCII 码；

汉字编码：输入码、内码、输出码。

6 浮点数据表示

（1）教材提及的浮点数表示： $N = (-1)^{M_s} \times M \times B^E$ ，其中 B 隐含于定义，通常为 2。



要求：规格化，即尾数 $\frac{1}{2} \leq |M| < 1$ 。在尾数是补码的定义下，要求符号位与尾数数值部分最高位不一致。

不满足规格化的要求时，即变形补码为 11.1xxx 或 00.0xxx 形式，需要“左规”：尾数左移 1 位，阶码减 1，此时可能因为阶码低于允许下限引起“下溢”，一般当作机器零处理；

如果尾数的变形补码出现 01.xxx 或者 10.xxx 形式时，需要“右规”：尾数右移 1 位，阶码加 1，此时可能因为阶码超过允许上限引起“上溢”，此时进入异常处理。

规格化浮点数的表示范围（假定 $m=10, l=6$ ）

最大正数	0 111111 111111111	$(1 - 2^{-(m-1)}) \times 2^{(2^{(l-1)}-1)}$
最小正数	0 000000 100000000	$\frac{1}{2} \times 2^{-2^{(l-1)}}$
最大负数	1 000000 011111111	$-(\frac{1}{2} + 2^{-(m-1)}) \times 2^{-2^{(l-1)}}$
最小负数	1 111111 000000000	$-1 \times 2^{(2^{(l-1)}-1)}$

(2) IEEE 754 标准

主要特点：尾数用原码（符号位处于最高位），阶码用移码，便于数值比较；对于规格化浮点数，小数点前隐含一个“1”；

常用格式：单精度（32 位，其中 8 位阶码）与双精度（64 位，其中 11 位阶码）。

下表指出了单精度浮点数的表示形式和若干特殊值。

单精度格式位模式	IEEE 浮点数的值
$0 < e < 255$	规格化浮点数： $(-1)^s \times 2^{e-127} \times 1.f$
$e = 0, f \neq 0$	非规格化浮点数： $(-1)^s \times 2^{-126} \times 0.f$
$e = 0, f = 0$	依据符号位分为+0 和-0
$e = 255, f = 0$	依据符号位分为+INF 和-INF
$e = 255, f \neq 0$	NaN（不分符号位）

※ IEEE 754 标准中移码定义与 2.2 节的描述不一致（在标准定义中 bias 值为 128），不能直接通过符号位取反实现，切记！例如，1.0 可以表示成 0 01111111 000...00，即 1.0×2^0 ，而不是 0 10000000 000...00。

双精度浮点数的表示形式和若干特殊值与单精度类似，但是： $e_{\max} = 2047$ ， $bias = 1023$ 。

浮点数的舍入模式：就近舍入（四舍六入五取双）；朝 0 舍入（截断）；朝 $+\infty$ 舍入（上取整 ceil）；朝 $-\infty$ 舍入（下取整 floor）。

7 检错与纠错码

（1）奇偶校验码：确保一个字（含校验码本身）含 1 的 bit 数为奇数（奇校验）或偶数（偶校验）。只能检测每个字中所产生的奇数个错误，且不具备纠错能力。

（2）CRC（循环冗余校验）

发送端：对需要校验的数据左移 k 位，再按照模 2 除法（其实就是异或）的方法将其除以长度为 k 的生成多项式，将最终余数加到左移的被除数。

接收端：按照上面的方法进行除法，余数应当为 0。否则根据余数可判别出错的位置。

（3）Hamming Code

要检测不超过 k 个单位的错误，所有合法编码之间就至少具有 $k+1$ 的汉明距离；要能校正 k 个错误，编码方案的最小汉明距离必须大于 $2k+1$ 。

设欲检测的信息码为 n 位，增添 k 位检测位，组成 $n+k$ 位的校验码。为准确对错误定位并指出代码无错，检测位数 k 应满足： $2^k \geq n+k+1$ 。检测位有 2^k 状态，全 0 表示无错，余下的组合指示错误位的位置。

汉明码实现：以教材的（8,4）偶校验码为例：

a. 在 1、2、4、8 位留空，其余位依次填写数据（数组下标从 1 开始）；

b. 将所有位置编号写成 $n = \sum_{k=0}^3 a \cdot 2^k, a \in \{0,1\}$ 形式，把含 $2^k (k=0,1,2,3)$ 的所有位置进行偶校验。

c. 接收端进行类似的操作，把校验位拼接起来。如果得到的校验值非 0，就是第 i 位出错了。（ i 就是得到的校验值）

第三章 运算方法与运算器

8 定点数的加减法

(1) 补码加减法则与溢出判定（双符号位判决法，进位判决法，运算符号判决法等）

(2) 一位全加器 ($Z_i = X_i \oplus Y_i \oplus C_i$, $C_{i+1} = (X_i \cdot Y_i) + (X_i + Y_i) \cdot C_i = G_i + P_i \cdot C_i$)

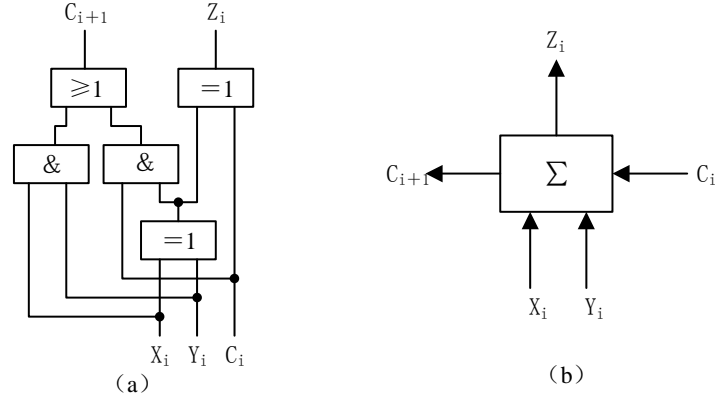


图3.1 一位全加器逻辑及其框图

(3) n 位加法器的实现

a. 行波进位（串行进位）加法器：总时延 $n\Delta t$

b. 先行进位（并行进位）加法器：总时延 $\tau + \Delta t$ （进位产生时间+生成和数的时间）

基本原理：可以将每一位展开成 $C_{i+k} = G_{i+k}^* + P_{i+k}^* C_i$ 的形式，可以通过逻辑门“提前”算出进位。

$$C_{i+1} = G_i + P_i C_i$$

$$C_{i+2} = G_{i+1} + P_{i+1} G_i + P_{i+1} P_i C_i$$

$$C_{i+3} = G_{i+2} + P_{i+2} G_{i+1} + P_{i+2} P_{i+1} G_i + P_{i+2} P_{i+1} P_i C_i$$

$$C_{i+4} = G_{i+3} + P_{i+3} G_{i+2} + P_{i+3} P_{i+2} G_{i+1} + P_{i+3} P_{i+2} P_{i+1} G_i + P_{i+3} P_{i+2} P_{i+1} P_i C_i$$

c. 组内并行、组间串行的进位链

d. 组内并行、组间并行的进位链（组内进位得到 G_{i+k}^*, P_{i+k}^* 按照类似方法进行组间进位）

(4) BCD 加减法：运算中某位 BCD 数（四位二进制数）相加的结果大于 9 或有向更高位的进位，则结果加 6（0110B），典型的 BCD 加法器实现电路：

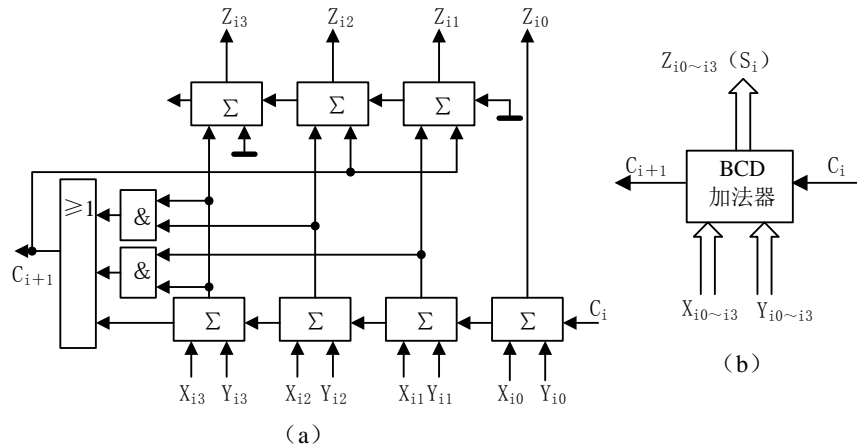


图3.4 BCD加法器

(5) 移码加减法：对该两移码求和差，再将结果的符号取反。

9 定点数的乘法

(1) 原码一位乘法

原则：乘积的符号为被乘数的符号位与乘数的符号位相异或；乘积的绝对值为被乘数的绝对值与乘数的绝对值之积。

操作类似于手工计算，但是是对被乘数进行移位（处于实现角度的考虑）。

(2) 原码二位乘法

Y_{i+1}	Y_i	C	操作
0	0	0	+0, 右移 2 位, $C=0$
0	0	1	+ X , 右移 2 位, $C=0$
0	1	0	+ X , 右移 2 位, $C=0$
0	1	1	+2 X , 右移 2 位, $C=0$
1	0	0	+2 X , 右移 2 位, $C=0$
1	0	1	- X , 右移 2 位, $C=1$
1	1	0	- X , 右移 2 位, $C=1$
1	1	1	+0, 右移 2 位, $C=1$

注：按定义 $Y_{i+1}Y_i=11$ 时应当+3|X|，但考虑性能方面需要调整。所有应当出现+3|X|的地方替换成-|X|，并且设 $C=1$ 。在运算中如果 $C=1$ ，需要额外在原来的基础上+|X|。初始 $C=0$ 。

(3) 补码一位乘法：校正法

a. 从补码表示的乘数最低位开始，若为 1 则加补码表示的被乘数 $[X]_{\text{补}}$ 。若为 0 则加 0。右移一位继续，直到乘数各位（除符号位）全都做完。

b. 若 $Y_0=1$ （乘数为负数），则加 $[-X]_{\text{补}}$ 。

(4) 补码一位乘法：Booth 法

乘数与被乘数均用补码表示，**连同符号位**一起参加运算。乘数最低位后增加一个附加位（用 A^{-1} 表示），初始设定为 0。**从附加位开始**操作完成运算。

y_i	y_{i-1}	$y_{i-1}-y_i$	操作
0	0	0	部分积+0, 右移 1 位
0	1	1	部分积+ $[X]_{\text{补}}$, 右移 1 位
1	0	-1	部分积+ $[-X]_{\text{补}}$, 右移 1 位
1	1	0	部分积+0, 右移 1 位

(5) 补码二位乘法：Booth 法

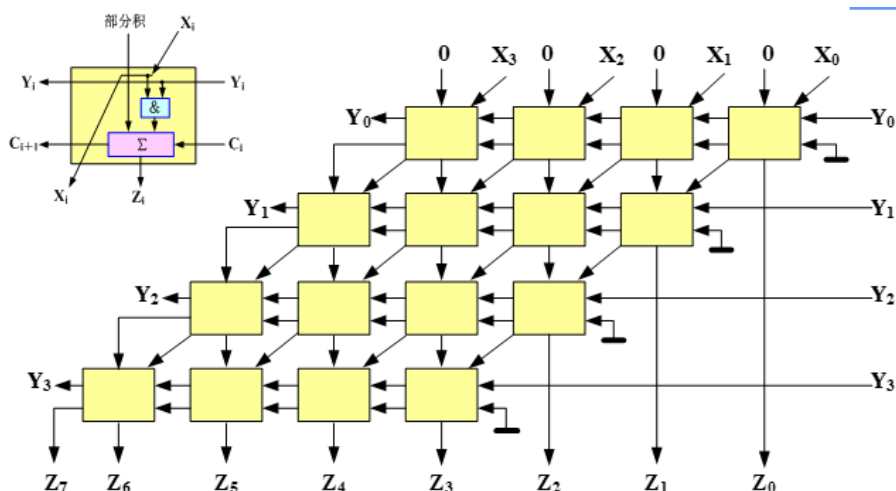
将 Y_i 与 Y_{i-1} 的状态比较和 Y_{i-1} 与 Y_{i-2} 的状态比较合在一起进行。由于乘数各位权值不一样，将一位乘法的两步合成一步，就有： $2(Y_{i-1}-Y_i)+(Y_{i-2}-Y_{i-1})=Y_{i-1}+Y_{i-2}-2Y_i$

Y_i	Y_{i-1}	Y_{i-2}	$Y_{i-1}+Y_{i-2}-2Y_i$	操作
0	0	0	0	+0, 右移 2 位
0	0	1	1	+ $[X]_{\text{补}}$, 右移 2 位
0	1	0	1	+ $[X]_{\text{补}}$, 右移 2 位
0	1	1	2	+2 $[X]_{\text{补}}$, 右移 2 位
1	0	0	-2	+2 $[-X]_{\text{补}}$, 右移 2 位
1	0	1	-1	+ $[-X]_{\text{补}}$, 右移 2 位
1	1	0	-1	+ $[-X]_{\text{补}}$, 右移 2 位
1	1	1	0	+0, 右移 2 位

与一位 Booth 法一样，只有一个附加位。乘数（含符号位）为偶数时右移 2 次的次数为 $n/2$ 次，最后一次右移 1 位；否则可以在乘数最后一位或多填充一个符号位，使其变为偶数。

(6) 阵列乘法器

a. 定点无符号阵列乘法器



B. 带符号阵列乘法器：将补码转换成不含符号位的原码的形式，即绝对值，然后进行乘法运算，再通过异或判断符号位

10 定点数的除法

(1) 原码除法的基本法则

前提条件：a. 除数 $\neq 0$ ；b. 定点纯小数时， $|被除数| < |除数|$ ；定点纯整数时， $|被除数| > |除数|$ 。（确保结果不超出允许表示的范围）

商的符号 = 被除数的符号 \oplus 除数的符号， $|商| = |被除数| \div |除数|$ 。

(2) 原码除法：恢复余数法

a. 符号位单独处理。

b. 对于定点纯小数，被除数左移一位，减除数，若够减，上商 1；若不够减，上商 0，同时加除数（恢复余数）。

c. 余数左移一位，减除数，若够减，上商为 1；若不够减，上商为 0，同时加除数（恢复余数）。重复上述过程直到除尽或精度达到要求。

d. 拼接商符得到商，即可获得除法的结果。

(3) 原码除法：加减交替法。操作要点：若余数 $R \geq 0$ ，则商上 1，左移一位，减除数；若余数 $R < 0$ ，则商上 0，左移一位，加除数。

(4) 补码除法

a. 如果被除数与除数同号，被除数减除数；如果被除数与除数异号，被除数加除数。运算结果称为余数。

b. 若余数与除数同号，上商为 1，余数左移一位，下次用余数减除数操作求商；若余数与除数异号，上商为 0，余数左移一位，下次用余数加除数操作求商。重复直至除尽或达到精度要求。

c. 商修正。在除不尽时，通常可用商的最低位恒置 1 进行修正来保证精度。

$[R_i]_{补}$ 与 $[Y]_{补}$	操作	$[R_{i+1}]_{补}$ 与 $[Y]_{补}$	上商	下一步操作	说明
同号	减 即 $+[-Y]_{补}$	同号(够减)	'1'	1, $+[-Y]_{补}$	
		异号(不够减)	'0'	1, $+ [Y]_{补}$	相当于恢复余数后左移一位,再 $+ [-Y]_{补}$
异号	加 即 $+ [Y]_{补}$	同号(不够减)	'1'	1, $+ [-Y]_{补}$	相当于恢复余数后左移一位,再 $+ [Y]_{补}$
		异号(够减)	'0'	1, $+ [Y]_{补}$	

(5) 阵列除法器

a. 判别进位的依据

在做无符号数减法时, 用被减数 + [减数]_补 来实现。若被减数 < 减数(不够减), 没有进位(借位); 若被减数 > 减数(够减), 有进位(借位)。

b. 可控加减单元 ($P=0$ 时输出为 Y ; 否则为 \bar{Y})

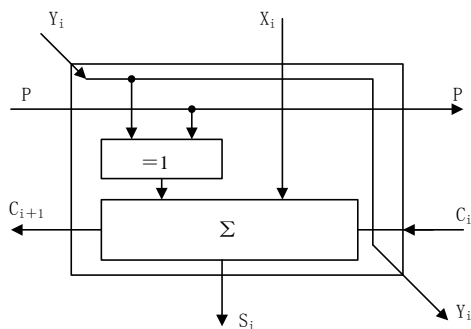
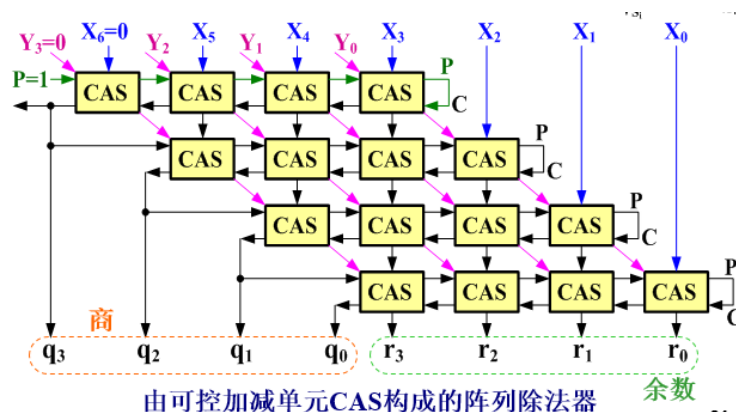


图3.17 可控加减单元CAS

c. 阵列乘法器框图



由可控加减单元CAS构成的阵列除法器

11 算术逻辑单元 ALU

(1) 单元电路组件

a. 寄存器

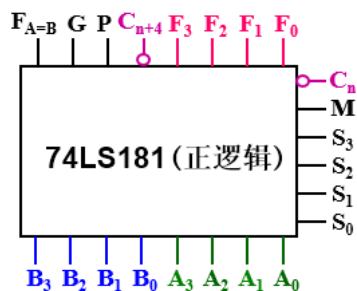
8D 锁存器（例如 74LS273，只能用于输出接口，否则会引发数据总线竞争）

带有三态门输出的锁存器（例如 74LS374）

b. 计数器

c. 移位寄存器（例如 74LS198）

(2) 算术逻辑部件（以 74LS181 为例，这是 4 位 ALU）



其中, A、B 均为输入字; F 为运算结果; S、M 为功能选择开关 (M=1 为逻辑运算, M=0 为算术运算)

(3) ALU 的级联工作（参见加法器部分）

(4) 运算器的基本结构：依据数据总线的多少分类有下面三种。

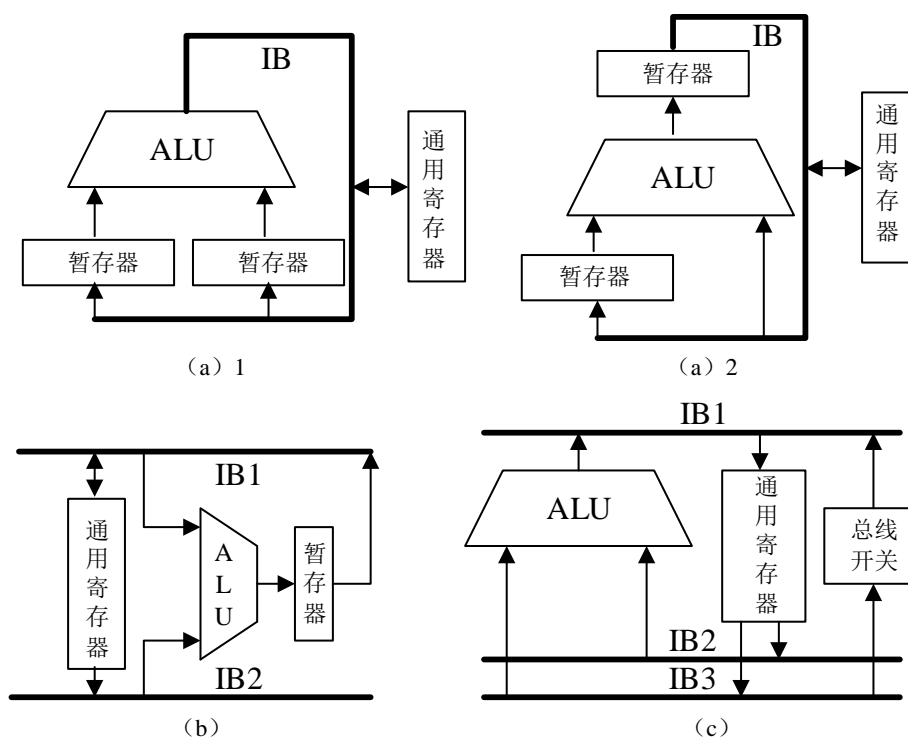


图3.26 运算器的三种基本结构

注意：总线总是分时工作的，任何时候只允许一个器件将其信号输出加到总线上。多于一个器件必然引起总线竞争。

同一个功能部件一次只能做一件事。

在双总线及三总线结构的运算器中需要多端口器件。

12 浮点运算

(1) 加减运算

a. 对阶（小数点对齐），原则是小阶对大阶，即小阶码每增加 1，其尾数右移一位，直到增大到与大阶码相同。

b. 尾数加减

c. 规格化（左规与右规）并判定溢出

注意：在对阶及规格化时需要将尾数右移，右移将丢掉尾数的最低位，这就出现舍入的问题。常见的方法：截（尾）断法；末位恒置 1 法；0 舍 1 入法。

(2) 乘除运算

a. 判断两浮点数是否为 0，若存在则直接返回 0 或（除法运算被除数）DIV 0 异常；

b. 阶码相加（减），并确认是否溢出；

c. 尾数乘（除），按照前述方法进行；

d. 规格化（左规与右规）并判定溢出，规格化操作只需要 1 次。

(3) 实现方法：软件方法；配专用浮点处理器；在处理器中设置浮点运算部件；浮点运算的流水线处理。

第四章 存储系统

13 存储系统概述

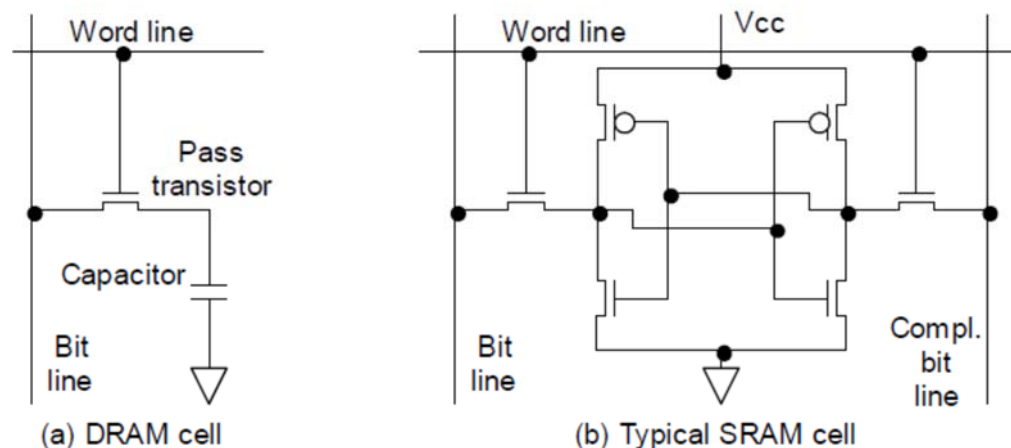
(1) 存储器分类依据：存储信息介质；计算机中的用途；存放信息的易失（挥发）性；存取方式（顺序存取、随机存取）；存储器的读写功能

(2) 存储器的性能指标：容量、速度（存取时间、存取周期、存储器带宽）、可靠性、功耗、价格。

14 内存储器：SRAM

(1) 译码结构：一维译码和二维译码

(2) 一种 SRAM 与 DRAM 单元电路



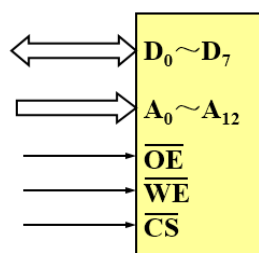
SRAM 的特点：只要不断电，信息一直保持；初始加电，状态随机；电路中总有晶体管导通，功耗大，集成度不高。

DRAM 的特点：必须定时刷新；初始加电，状态随机；在不进行读写及刷新时，电路中无晶体管导通：功耗低，集成度高。

教材给出了另外的 DRAM 电路，可供参考。

(3) CPU 与存储器的时序协调：在读取和写入过程中必须考虑数据和 RAM 的读写时延，选择芯片应当有 30% 余量。快速的 CPU 使用慢速的存储器时，需要采取措施。

(4) 内存接口（以基于 PC/XT ISA 总线的 8K×8 SRAM 芯片为例）



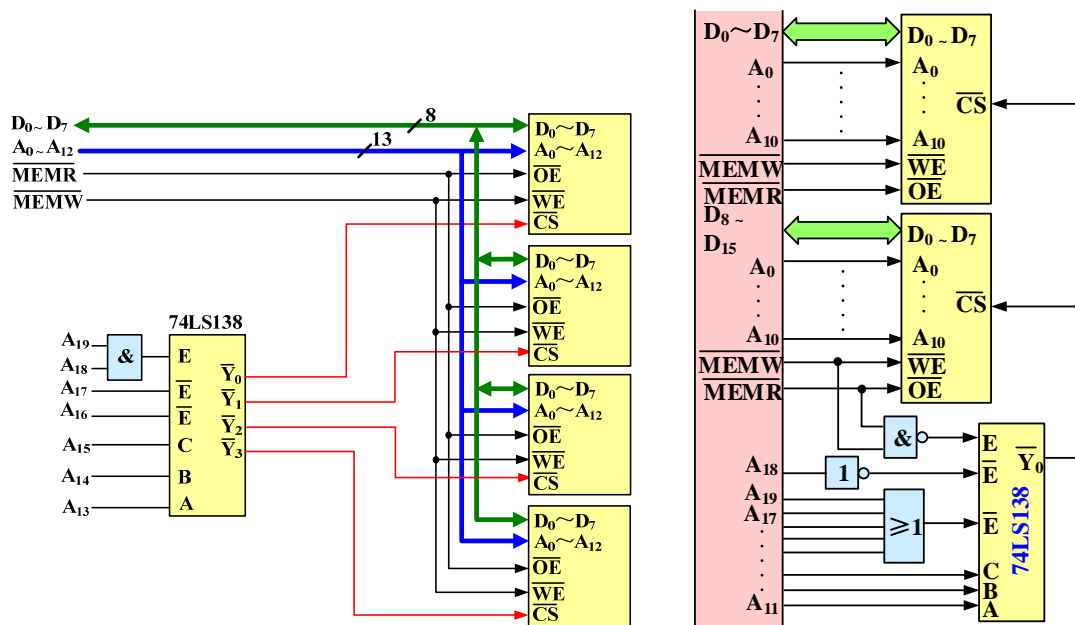
芯片的引脚： A_0 到 A_{12} 为地址信号输入线； D_0 到 D_7 为数据输入输出线。 \overline{CS} 是使能端，

低位有效； \overline{WE} 和 \overline{OE} 分别是写允许和输出允许信号。

内存与系统总线的连接

内存的字扩展（示例：利用 8K×8bit 的 SRAM，构成 32KB 内存。）

内存的位扩展（示例：利用 2K×8bit 的 SRAM，构成 2K×16bit 内存。）

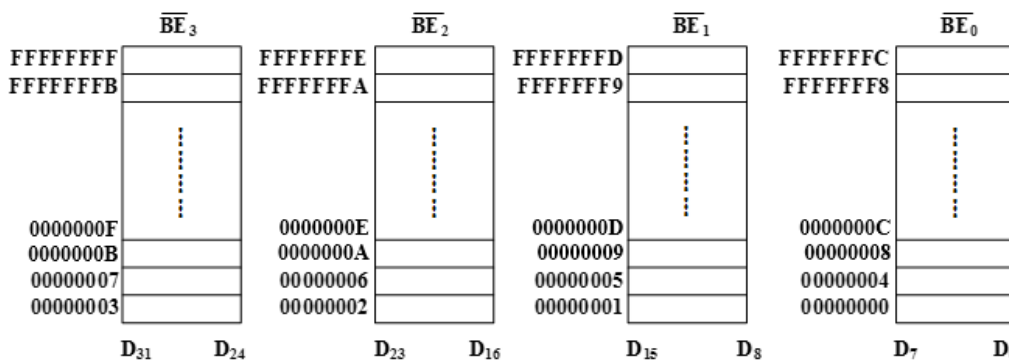


(5) 80x86 CPU 的内存接口

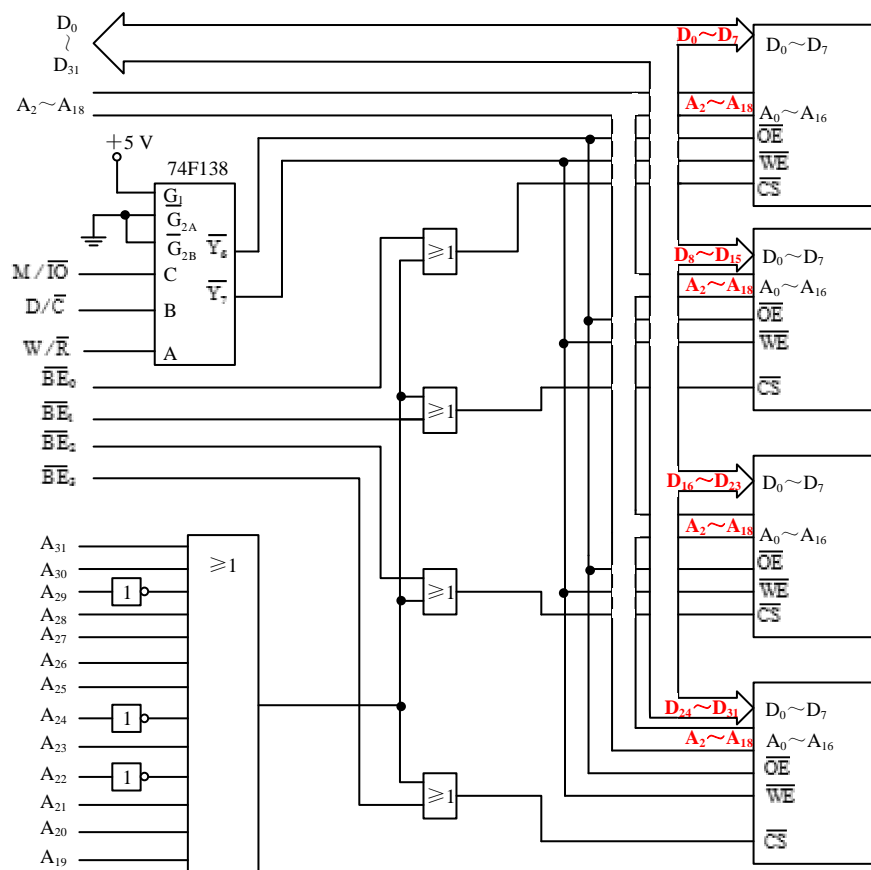
8086: 16 位处理器，内存数据线 16 位，奇偶分体（支持按字节或按字访问内存）。为了实现这个目标，用到 \overline{BHE} 信号。

\overline{BHE}	A_0	操作	所用数据线
0	0	从偶地址开始读/写一个字	AD15~AD0
0	1	从奇地址读/写一个字节	AD15~AD8
1	0	从偶地址读/写一个字节	AD7~AD0
1	1	无效	无

80386、80486: 32 位处理器，地址信号: $A_2 \sim A_{31}$; 存储体选择信号: $\overline{BE}_0 \sim \overline{BE}_3$ (对应 4 个存储体); 32 位数据信号: $D_0 \sim D_{31}$ 。通常利用 4 片容量为 128 K \times 8 bit 的 SRAM 芯片构成 512 KB 的内存。



80486 内存芯片与总线的连接如下图所示。



Pentium 及更新的系列: 与访问内存有关的是地址信号 $A_3 \sim A_{31}$ 、体选择信号 $BE_0 \sim BE_7$ 、64 位数据信号 $D_0 \sim D_{63}$ 、及控制信号 M/\overline{IO} 、 D/\overline{C} 和 W/\overline{R} 。

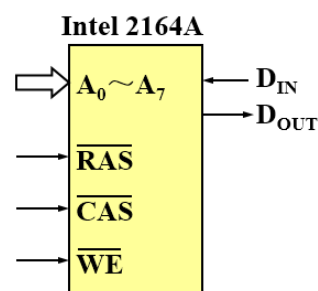
15 只读存储器

分类: 掩膜型 ROM、可编程 ROM (包括 PROM、EPROM、EEPROM、Flash 芯片)

16 内存储器: DRAM

(1) 一种 $64K \times 1b$ 芯片的接口: 注意: 对于 DRAM, 8 位地址线可通过列选和行选输入到内存芯片中, 达到 16 位地址空间的寻址。 \overline{RAS} 和 \overline{CAS} 分别为行、列地址锁存信号。

读出和写入数据的过程如下图所示。注意: 在图 4.14 中 \overline{WE}



在 \overline{CAS} 有效前已经变成低电平, 即所谓“提前写”, 此时 D_{out}

处于高阻态。相反情况下可先读出数据到 D_{out} , 再写入 D_{in} 数据, 即“读变写操作周期”。

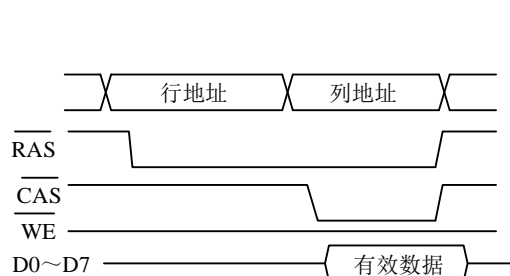


图4.13 DRAM的读出过程

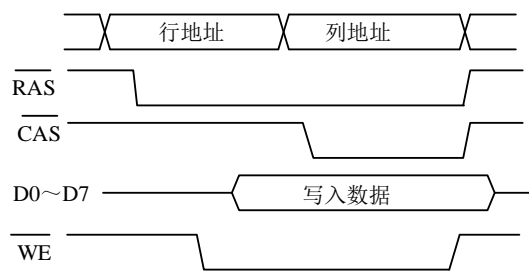


图4.14 DRAM 的写入过程

刷新：2 到 4ms 一次，过程：每次送出行地址加到芯片，利用 \overline{RAS} 有效即可实现。 \overline{CAS} 始终处于无效状态。

(2) 新型 DRAM 技术

SDRAM：与 CPU Clock 保持同步；多个存储体（Bank）；突发读写功能；

DDR：时钟的上升沿和下降沿都可以传输数据。

17 主存储器的校验

(1) 奇偶校验

(2) 交叉校验

当存储或传送数个字节时，对数据块进行纵横两个方向上的奇（偶）校验，故称为交叉校验。

可以发现并纠正数据块中的 1 位错误。该方法可以发现 2 位同时出错，但无法确定出错的位置，因而无法纠正 2 位错误。

(3) Hamming 码校验与纠错

18 其它存储器

(1) 多端口存储器：时间并行，用于提高访存速度

具有多(两)组独立的读写端口；允许多(两)个 CPU 或控制器从多(两)个端口同时异步访问存储单元。

存在访问冲突，即从多(两)个端口同时写同一个存储单元，发生冲突。解决方法：芯片内电路裁决。

(2) 多模块存储器：空间并行用于提高访存速度

单体多字：多模块构成单一存储体

一次访存并行读写 m 个字（数据总线/寄存器宽度为 m 字， m 个顺序地址分配在 m 个存储模块的相同存储单元上）。

访问的内容在存储体同一行，频带是单体单字存储器的 m 倍，否则，并行效率大为降低。

多体交叉：多个存储模块可并行独立工作

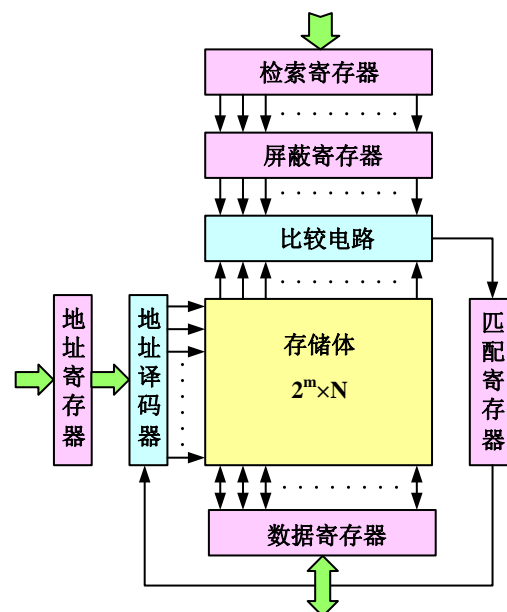
一次访存对 m 个存储模块的不同存储单元并行（或流水）读写 m 个字（数据总线/寄存器宽度为 1 字）。

频带是单体单字存储器的 m 倍。

(3) 相联存储器

把存储单元所存内容的某一部分作为检索项（即关键字项），去检索该存储器，并将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。

应用：Cache、Translation Look-ahead Buffer 等。



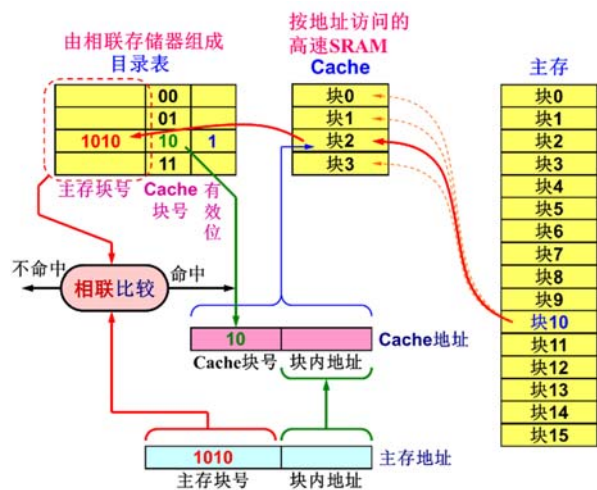
19 Cache

(1) 实现依据：程序、数据的局部性原理（时间局部性、空间局部性）；价格矛盾。

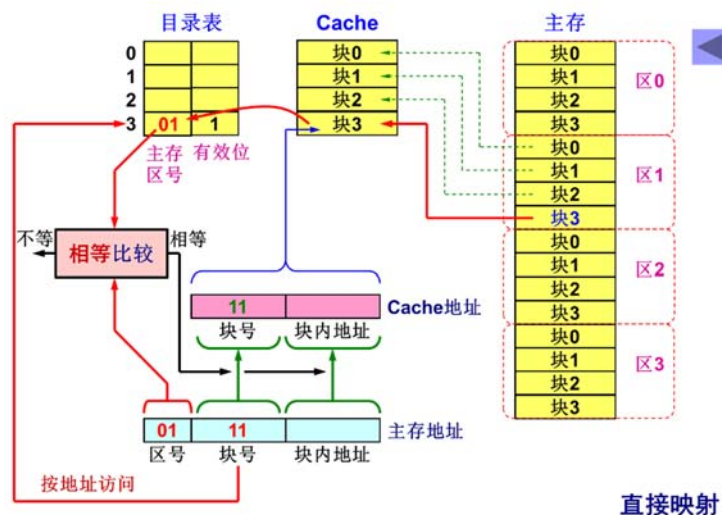
(2) 三种地址映射方式

a. 全相联

主存的任意一块可以映象到 Cache 中的任意一块。优点：块冲突概率低，Cache 空间利用率高。缺点：相联目录表容量大→成本高、查表速度慢。



全相联



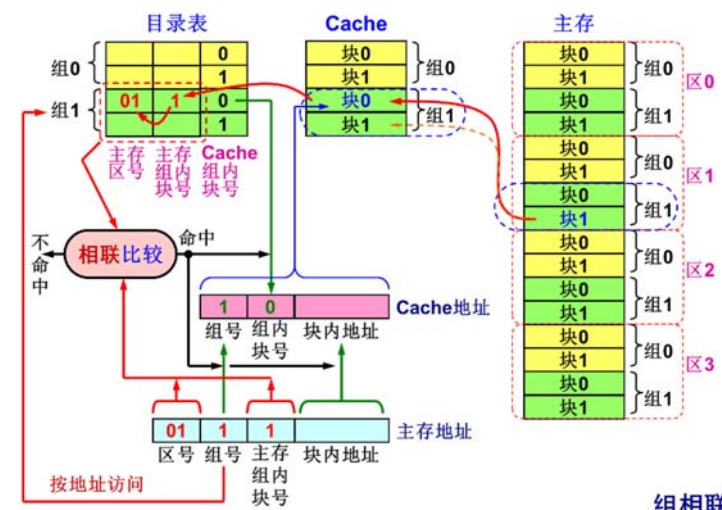
直接映射

b. 直接映射

主存的每一块只能映象到 Cache 的一个特定的块中。整个 Cache 地址与主存地址的低位部分完全相同。优点：硬件简单，不需要相联存储器，只要容量较小的按地址访问的区号标志表存储器与少量外比较电路。访问速度快（无需地址变换）。缺点：Cache 块冲突概率高；Cache 空间利用率很低。

c. 组相联

将主存和 Cache 都等分成相同大小的块，并将主存空间按照 Cache 大小等分成区，再将 Cache 和主存空间中的每一区都等分成大小相同的组，让主存各区中的某组中的任何一块均可直接映象装入到 Cache 中对应组的任何一块上。从主存的组到 Cache 的组之间采用直接映射方式，在两个对应的组内部采用全相联映射方式——组间直接，组内全相联。



组相联

(3) 常见替换算法（不适用于直接映射）

随机算法（RAND, Random Algorithm）

先进先出算法（FIFO, First-in First-out Algorithm）利用了历史信息，但没有反映局部性——最先调入的，可能也是要使用的。

最不使用算法（LFU, Least Frequently Used Algorithm）：计数器位数多，实现困难。

近期最少使用算法（LRU, Least Recently Used Algorithm）

最佳替换算法（OPT, Optimal replacement algorithm）：无法实现，其余算法的度量。

(4) 主存与 Cache 的一致性

Cache 类型	操作	访存操作	访存时间
写直达 Cache	读命中	只读 Cache	T_C
	写命中	写 Cache，同时写内存	T_C （隐藏 T_M ）
	读不命中	调入 Cache 块，再读 Cache	$T_B + T_C$
	写不命中	只写内存	T_M
写回 Cache	读命中	只读 Cache	T_C

	写命中	只写 Cache	T_C
	读不命中	调入 Cache 块, 再读 Cache	$T_B + T_C$
	写不命中	调入 Cache 块, 再写 Cache	$T_B + T_C$

(5) Cache 性能评价指标

定义: Cache 的访问周期为 T_C , 主存的访问周期为 T_M , 数据块调入 Cache 的块传输时间为 T_B , Cache 的命中率为 H 。

平均访问周期: $T = H \times T_C + (1 - H) \times T_M = H \times T_C + (1 - H) \times (T_B + T_C)$

加速比: $SP = T_M / T$ (命中率愈高, 加速比愈大。最大也只能接近 T_M / T_C)

成本: $C = (C_1 \times S_1 + C_2 \times S_2) / (S_1 + S_2)$

命中率与 Cache 容量的关系: $H = 1 - S^{-0.5}$

命中率与块大小的关系 (条件: 组相联, Cache 容量一定): 有极大值

两级 Cache 总未命中率 (总失效率): 总失效率 = 第一级失效率 \times 第二级失效率

20 Virtual Memory

(1) 动机与目的: 高速的主存容量满足不了要求; 多道程序运行 \rightarrow 各程序独立编址 \rightarrow 程序再定位 “逻辑地址 \rightarrow 实际地址”;

(2) 主要层次: 主存储器 + 联机工作的外部存储器 + 辅助硬件 + 系统软件。对于程序员, 可看作一个单一的存储器。

(3) 段式虚拟存储器: 将程序按逻辑意义分成段, 按段进行调入、调出和管理。

优点: 程序的模块化性能好。便于多道程序共享主存中的某些段。程序的动态链接和调度比较容易。便于按逻辑意义实现存储器的访问方式保护。

缺点: 地址变换所花费的时间长, 两次加法。段映象表庞大, 地址、段长字段太长。主存储器的利用率往往比较低——存储管理复杂; 段间 “零头”。对辅存 (磁盘存储器) 的管理比较困难。

(4) 页式虚拟存储器: 将主存空间和程序空间都机械等分成相同大小的页面, 按页顺序编号, 让程序的起点必须处在主存中某一个页面位置的起点上。任一主存单元的地址由实页号和页内位移两个字段组成。

优点: 主存储器的利用率比较高。页表相对比较简单, 使用硬件少。地址变换的速度比较快。对磁盘的管理比较容易。

缺点: 程序的模块化性能不好。页表很长, 需要占用很大的存储空间 (通过多级页表改进)

(5) 段页式虚拟存储器: 将程序按逻辑意义先分成段, 再让各段和实主存都机械等分成相同大小的页面。每道程序通过一个段表和相应的一组页表来进行程序在主存空间中的定位。

(6) 关于虚拟存储器的几点说明

多用户工作: 添加用户号字段, 每个用户使用不同的页 (段) 表

调度方式

替换算法: 与 Cache 一致

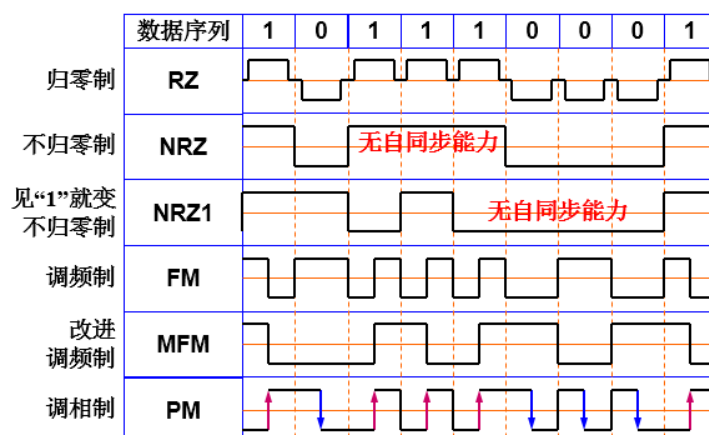
(7) Pentium 的虚拟存储器机制

21 外存储器

(1) 磁存储原理

记录原理：水平记录与垂直记录

记录方式：

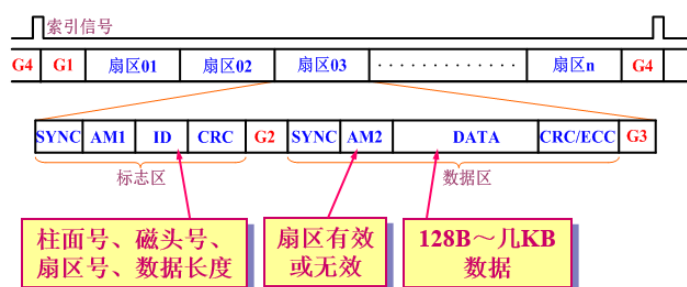


性能指标：自同步能力；编码效率（位密度÷最大磁化翻转次数，FM与PM为0.5，其余为1）

(2) 磁盘存储器

磁盘结构的划分：柱面、磁道、扇区

数据记录格式：



主要技术指标：

道密度：道/mm，道/英寸

位密度：bit/mm，bit/英寸，最靠近盘中心的磁道

存储容量

非格式化容量=位密度×内圈磁道周长×每个记录面的磁道数×记录面数

格式化容量=每个扇区的字节数×每道的扇区数×每个记录面的磁道数×记录面数

平均访问时间=平均寻道时间+平均等待时间

寻道时间：磁头移动到目标磁道（或柱面）所需要的时间，一般几ms~10ms。

等待时间：待读写的扇区旋转到磁头下方所用的时间，一般选用磁盘旋转一周所用时间的一半作为平均等待时间，在1ms~6ms之间。

转速：RPM（转/分钟）通常为5400RPM，7200RPM，10000RPM，15000RPM

数据传输率=每个扇区的字节数×每道扇区数×磁盘的转速（磁头找到数据的地址后，单位时间内写入或读出的字节数）

(3) RAID (独立冗余磁盘阵列)

种类	级	说 明	I/O 读写速度 (读/写)	数据传输率 (读/写)	典型应用
条带化	0	无冗余	大条带: 优秀	小条带: 优秀	高性能, 用于非关键性数据
镜像	1	镜像	良好/一般	一般/一般	系统盘、重要文件
并行处理	2	使用汉明码的数据冗余	差	优秀	无商业应用
	3	位交错奇偶校验	差	优秀	大容量的 I/O 请求, 如图像、CAD
独立存取	4	块交错奇偶校验	优秀/一般	一般/差	无商业应用
	5	块交错分布奇偶校验	优秀/一般	一般/差	高请求速度, 读集中, 数据查询
	6	块交错双分布奇偶校验	优秀/差	一般/差	要求极高可用性的应用

RAID level		Disk failures tolerated, check space overhead for 8 data disks	Pros	Cons	Company products
0	Nonredundant striped	0 failures, 0 check disks	No space overhead	No protection	Widely used
1	Mirrored	1 failure, 8 check disks	No parity calculation; fast recovery; small writes faster than higher RAID's; fast reads	Highest check storage overhead	EMC, HP (Tandem), IBM
2	Memory-style ECC	1 failure, 4 check disks	Doesn't rely on failed disk to self-diagnose	~ Log 2 check storage overhead	Not used
3	Bit-interleaved parity	1 failure, 1 check disk	Low check overhead; high bandwidth for large reads or writes	No support for small, random reads or writes	Storage Concepts
4	Block-interleaved parity	1 failure, 1 check disk	Low check overhead; more bandwidth for small reads	Parity disk is small write bottleneck	Network Appliance
5	Block-interleaved distributed parity	1 failure, 1 check disk	Low check overhead; more bandwidth for small reads and writes	Small writes → 4 disk accesses	Widely used
6	Row-diagonal parity, EVEN-ODD	2 failures, 2 check disks	Protects against 2 disk failures	Small writes → 6 disk accesses; 2× check overhead	Network Appliance

(4) 移动存储设备: 移动硬盘, 闪存盘 (U 盘), 固态硬盘

(5) 光存储器

分类: 只读光盘、一次写多次读光盘、可擦写光盘 (相变光盘 PCD、磁光盘 MO)

结构: 由透明的聚合物基片、铝合金或金反射层、漆膜保护层的圆盘构成。

技术指标: 尺寸、存储容量、传输速率、平均存取时间、缓冲区大小、接口类型、其它
数字通用光盘 DVD 的特点

CD 采用的激光波长为 780nm 而 DVD 采用红色激光, 波长为 650nm。波长愈短则聚焦点愈小。

DVD 的光道间距为 740nm, 而 CD 的光道间距为 1600nm, 故 DVD 光道更加紧凑。

DVD 定义一张光盘四种存储容量: 单面单层 4.7GB; 双面单层 9.4GB; 单面双层 8.5GB; 双面双层 17GB。而前面已提到 CD 的容量只有几百 MB。

兼容性: DVD-RAM、DVD+RW、DVD-RW