

计算机组成与体系结构第一章总结

1、电子计算机的分类：电子模拟计算机和电子数字计算机

特点：模拟计算机是数值由连续量来表示，运算过程连续；

数字计算机是按位运算，并且不连续地跳动计算；

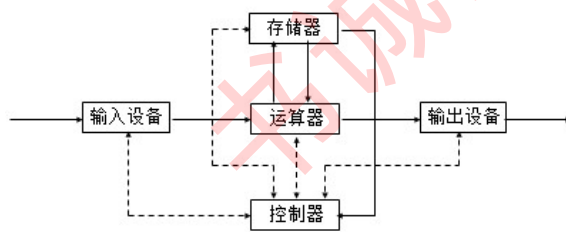
2、电子（数字）计算机分类

专用计算机和通用计算机（依据效率，速度，价格，运行的经济性和适应性划分）

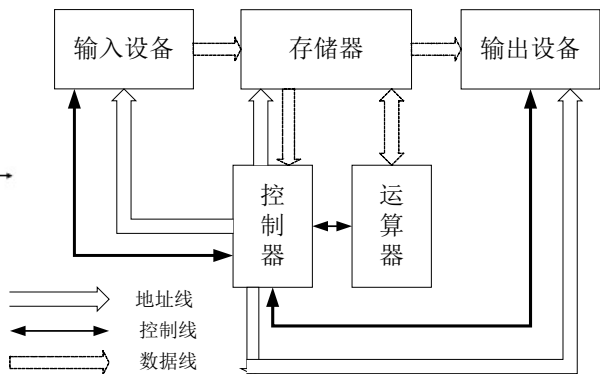
3、计算机系统由硬件和软件两大部分组成

4、硬件

（1）一般的计算机结构框图（运算器为中心）



一般的计算机结构框图（存储器为中心）



（2）冯·诺依曼计算机的特点

采用二进制表示机器指令和数据

硬件系统由运算器、控制器、存储器、输入设备和输出设备五大部分组成

程序和数据预先存放在存储器中，存储器按地址编址

操作时根据程序中指令的执行顺序，从存储器中取出指令或数据，由控制器解释，运算器完成运算

设计思想：采用二进制表示各种信息以及存储程序和程序控制。存储程序的概念是将解题程序（连同必须的原始数据）预先存入存储器；程序控制是指控制器依据所存储的程序控制全机自动、协调地完成解题任务。存储程序并按地址顺序执行。存储程序和程序控制统称为存储程序控制。

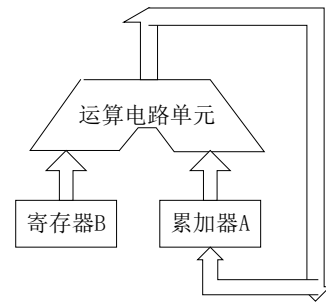
(3) 硬件各个主要部分

① 运算器: (如右图)

算术运算和逻辑运算

在计算机中参与运算的数是二进制的

运算器的长度一般是 8、16、32 或 64 位



② 存储器:

存储数据和程序

容量 —— 存储器所有存储单元的总数

(存储单元 (保存一个数 16 个触发器)、存储单元地址 (每个存储单元的编号)、容量单位)

内存储器 (ROM、RAM): 半导体存储器

外存储器: 计算机中配备了存储容量更大的磁盘存储器和光盘存储器

存储器单位: $2^{10} \text{ byte} = 1\text{K}$ $2^{10} \text{ K} = 1\text{M}$ $2^{10} \text{ M} = 1\text{G}$ $2^{10} \text{ G} = 1\text{T}$

③ 控制器:

指令和程序 (每一个基本操作叫做一条指令; 解决某一问题的一串指令序列叫做程序)

指令的形式 —— 操作码和地址码: 指令的内容由两部分组成, 操作的性质和操作数的

地址: 前者称为操作码, 后者称为地址码。

控制器的基本任务: 按照一定的顺序一条接一条取指令、指令译码、执行指令。

取指周期和执行周期: 控制器按照时序工作, 通常将分为两个过程:

取指周期: 从内存读出的信息流是指令流, 它流向控制器;

执行器周期: 从内存读出的信息流是数据流, 它由内存流向运算器。

PS: (出题) —— CPU 如何识别从主存取出来的二进制代码是指令还是数据?

从时间来说: 取指令事件发生在“取指周期”

取数据事件发生在“执行周期”

从空间来说: 取出的代码是指令, 一定送往指令寄存器 (控制器)

取出的代码是数据, 一定送往数据寄存器 (运算器)

④ 适配器与输入输出设备: 适配器是接口

PS: 控制器和运算器合称为中央处理器 CPU, CPU 和内存储器合称为计算机主机。

5、软件

分为系统软件和应用软件

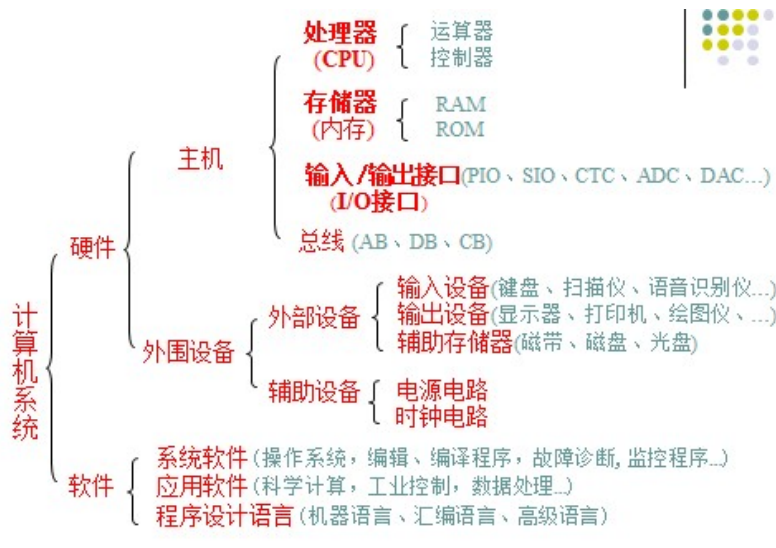
系统软件分类:

① 各种服务性程序 ② 语言程序 ③ 操作系统 ④ 数据库管理系统

系统软件发展历史:

手编程序 —— 汇编程序 —— 算法语言 —— 操作系统 —— 数据库

6、计算机系统组成框图汇总:



7、计算机系统的层次结构：（层次图如右）

第 0 级由**硬件**实现

第 1 级由**微程序**实现

第 2 级至第 6 级由**软件**实现

由软件实现的机器称为：虚拟机

第 2 级是**传统指令系统（机器语言）**机器

第 3 级是**操作系统**机器

操作系统是运行在第 2 级上的解释程序

第 4 级是**汇编语言**机器

第 5 级是**高级语言**机器

第 6 级是**应用语言**机器(图中未显示)

每一级都能进行程序设计

8、软件和硬件的逻辑等价性：

任何操作可以由软件来实现也可以有硬件来实现（设计计算机系统时，应考虑各个方面的因素：价格、速度、可靠性、存储容量、变更周期）

固件——功能上是软件，形态上是硬件

9、计算机体系结构的定义：

定义一：

Amdahl 于 1964 年在推出 IBM360 系列计算机时提出：程序员所看到的计算机系统的属性，即**概念性结构和功能特性**

定义二：

计算机系统结构主要研究**软硬件功能分配**和对**软硬件界面的确定**

10、改进计算机性能

基本方法：**加快经常性事件的执行速度**

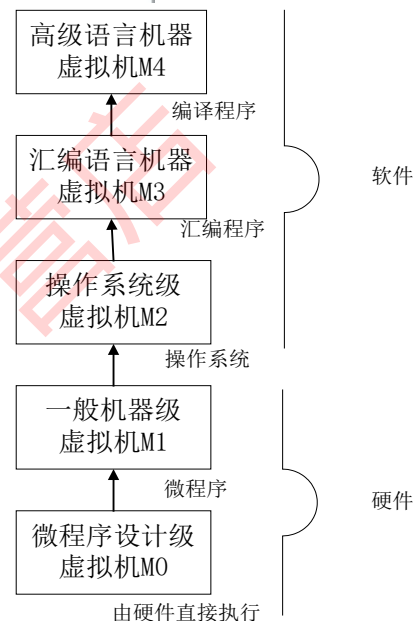
Amdahl 定律：系统中某一部件由于采用某种更快的执行方式后整个系统性能的提高与这种执行方式的使用频率或占总执行时间的比例有关。

在 Amdahl 定律中，**加速比**与**两个因素**有关：

$$F_e = \frac{\text{可改进部分占用的时间}}{\text{改进前整个任务的执行时间}}$$

$$S_e = \frac{\text{改进前改进部分的执行时间}}{\text{改进后改进部分的执行时间}}$$

$$S_n = \frac{T_0}{T_n} = \frac{1}{(1 - F_e) + \frac{F_e}{S_e}}$$



改进后整个系统的加速比为：

eg. 某一部件的处理速度加快到 10 倍，该部件的原处理时间仅为整个运行时间的 40%，
所以，其中—— $F_e=0.4$, $S_e=10$ ，代入公式即可

11、计算机系统的主要技术指标：

- (1) **机器字长**：运算器一次运行二进制位数
- (2) **运算速度**：描述一台计算机的运算速度的单位应该是每秒执行多少条指令
单位是 MIPS（百万条指令每秒）

$$V_m = \frac{1}{\sum_{i=1}^n f_i \cdot t_i}$$

式中，n—指令的种类

f_i —第 i 种指令在程序中出现的频度（%）

t_i —第 i 种指令的指令周期

对于带有浮点运算的计算机系统，常用每秒百万个浮点操作—— $MFLOPS = \frac{\text{浮点数操作次数}}{\text{执行时间} \times 10^6}$

- (3) **机器容量**：计算机的容量实际上是指机器内部主存储器的容量。
- (4) **吞吐量**：计算机系统的吞吐量是指计算机在单位时间内能处理的信息量。
- (5) **响应时间**：响应时间指从给定计算机输入到出现对应的输出之间的时间间隔。响应时间取决于用户输入的信息、系统特性以及在用户输入信息时系统正在处理的其他负载。

计算机组成与体系结构第二章总结

1、定点表示法概念

(1) 纯小数

① $x_0, x_1, x_2, x_3, \dots, x_{n-1}, x_n$ (x_0 表示符号位， x_1-x_n 为尾数)

② 表示范围：（表示数的范围是 $0 \leq |x| \leq 1-2^{-n}$ ）

$x=0.00\dots0$ $x=1.00\dots0$	$x=0$	正 0 和负 0 都是 0
$x=0.11\dots1$	$x=1-2^{-n}$	最大
$x=0.00\dots01$	$x=2^{-n}$	最接近 0 的正数
$x=1.00\dots01$	$x=-2^{-n}$	最接近 0 的负数
$x=1.11\dots1$	$x=-(1-2^{-n})$	最小

(2) 纯整数

① $x_0, x_1, x_2, x_3, \dots, x_{n-1}, x_n$ (x_0 为符号位，后面为尾数，小数点在最后)

②表示数的范围是 $0 \leq |x| \leq 2^n - 1$

2、浮点表示法概念

- (1) 格式: $N=R^E.M$ (基数 R 为固定值; E 为指数; M 为尾数)
 (2) 机器中表示

阶符	阶码 (纯整数)	数符	尾数 (纯小数)
----	----------	----	----------

3、数的机器码表示

- (1) 原码:
 注意符号位——0 正 1 负;
 有+0 和-0 之分, 字长 8 为的范围是-127 到+127;
- (2) 补码:
 正数补码与原码相同; 负数补码为其原码右数第一个 1 开始, 左边的都取反, 右边的都不变 (包括这个 1);
 无正负 0 之分;
 取值范围: $-2^n \sim 2^n - 1$ (定点整数) 定点小数: $1 \sim 1 - 2^{-n}$
 字长 8 位的范围是-128 到+127;
- (3) 反码:
 正数反码与原码, 补码相同; 负数补码符号位为 1, 其余与原码各位相反;
 有正负 0 之分;
 字长 8 位, 取值范围-127 到+127;
- (4) 移码:
 移码与补码符号位互为相反数, 其余各位相同;
 无正负 0 之分;
 字长 8 位, 取值范围与补码相同;

4、数据格式

- (1) 单精度 (32) 和双精度 (64);
 (2) 尾数用原码, 指数用移码;
 (3) 一个规格化的 32 位浮点数 x 的真值表示为 $x=(-1)^S \times (1.M) \times 2^{E-127}$

	31	30	23	22	0
32位浮点数	S	E	M		

- (4) 规格化的 64 位浮点数 x 的真值为: $x=(-1)^S \times (1.M) \times 2^{E-1023}$

	63	62	52	51	0
64位浮点数	S	E	M		

- (5) 规格化表示:
 尾数的最高位为非零数值的浮点数
- (6) 浮点数的范围:

格式	最小值	最大值
单精度	$E=1, M=0,$ $1.0 \times 2^{1-127} = 2^{-126}$	$E=254, f=.1111\cdots,$ $1.111\cdots 1 \times 2^{254-127}$ $= 2^{127} \times (2-2^{-23})$
双精度	$E=1, M=0,$ $1.0 \times 2^{1-1023} = 2^{-1022}$	$E=2046, f=.1111\cdots,$ $1.111\cdots 1 \times 2^{2046-1023}$ $= 2^{1023} \times (2-2^{-52})$

5、补码加减法

(1) 补码加法

公式: $[x+y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}} \pmod{2}$

(2) 补码减法

公式: $[x-y]_{\text{补}} = [x]_{\text{补}} - [y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}} \pmod{2}$

(3) 溢出问题

① 溢出检测方法一: 双符号位 (变形补码)

Sf_1	Sf_2	
0	0	正确 (正数)
0	1	上溢
1	0	下溢
1	1	正确 (负数)

Sf_1 、 Sf_2 表示两位符号位

② 溢出检测方法二: 单符号位

Cf	$C0$	
0	0	正确 (正数)
0	1	上溢
1	0	下溢
1	1	正确 (负数)

Cf 为符号位产生的进位, $C0$ 为最高有效位产生

(4) 基本加法和减法器

① 半加器 (不考虑低位进位) 和全加器 (考虑低位进位和向高位的进位)

一位全加器的逻辑表达式:

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_i B_i + B_i C_i + A_i C_i = A_i B_i + (A_i \oplus B_i) C_i = \overline{\overline{A_i B_i} \cdot \overline{(A_i \oplus B_i) C_i}}$$

6、定点乘法运算

(1) 定点原码乘法

被乘数A	a3	a2	a1	a0					
乘数B	×	b3	b2	b1	b0				
	<hr/>								
	a3b0	a2b0	a1b0	a0b0					
	a3b1	a2b1	a1b1	a0b1					
	a3b2	a2b2	a1b2	a0b2					
+	a3b3	a2b3	a1b3	a0b3					
<hr/>									
P7	P6	P5	P4	P3	P2	P1	P0		

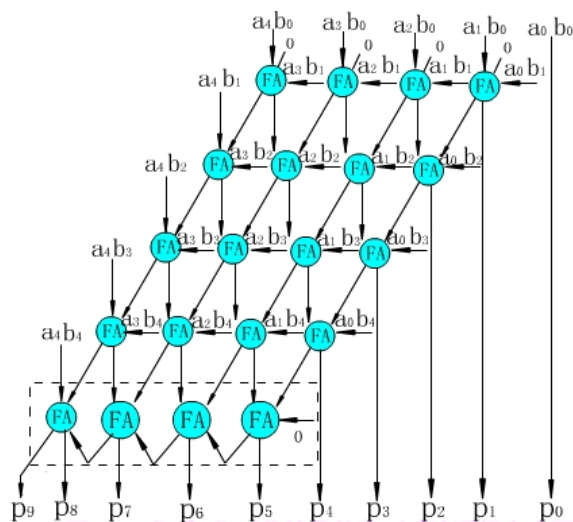


图2.5 5位乘5位不带符号的阵列乘法逻辑电路图

与门延时+ (n-1) 和运算延时+ (n-1) 进位延时
 $t_m = 2T + (n-1)6T + (n-1)2T = (8n-6)T$

(2) 定点补码乘法

原理：算前求补—乘法器—算后求补

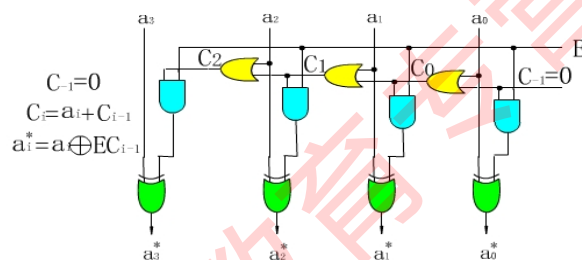


图2.6 对2求补器电路图

$E=0$ 时，输入和输出相等

$E=1$ 时，则从数最右端往左边扫描，直到第一个 1 的时候，该位和右边各位保持不变，左边各数值位按位取反

时间延迟分析：转换 $n+1$ 位带符号的时间延迟为 $t = n*2T + 5T$ ，其中 $n*2T$ 为或门延迟时间， $5T$ 为最高位与门和异或门的时延。

(3) 总结：

原码阵列乘法只需另加符号位的处理。

补码阵列乘法可对负操作数求补（得绝对值）后相乘；

再由乘积符号位决定是否对乘积求补即可

7、定点除法运算

(1) 原码除法：

符号位单独处理，尾数相除

	0.1 0 1 1	0.1 1 0 1	商q
0.1 0 1 1	— 0.1 0 0 1 0		$x(r_0)$ 被除数
	— 0.0 1 0 1 1		$2^{-1}y$ 除数右移1位,减除数
	0.0 0 1 1 1 0		r_1 得余数r1
	— 0.0 0 1 0 1 1		$2^{-2}y$ 除数右移1位,减除数
	0.0 0 0 1 1 1 0		r_2 得余数r2
	— 0.0 0 0 1 0 1 1		$2^{-3}y$ 除数右移1位,不减除数
	0.0 0 0 1 1 1 0 0		r_3 得余数r3
	— 0.0 0 0 0 1 0 1 1		$2^{-4}y$ 除数右移1位,减除数
	— 0.0 0 0 0 0 0 0 1		r_4 得余数r4

(2) 不恢复余数除法 (加减交替法):

计算步骤如下:

- 1、判断被除数 < 除数, 除数是否为 0;
- 2、减除数, 得到余数;
- 3、余数为负, 商为 0;
- 4、余数、商一起左移;
- 5、余数为正, 商为 1;
- 6、最后余数是否需要修正

	被除数	商	说明
$[x]_2 = 001011$	001011	00000	开始
$[y]_2 = 001101$	110011		-y (除数)
$[-y]_2 = 110011$	111110	00000	为负, 商0
	111100	00000	余数、商左移一位
	001101		+y (除数)
	001001	00001	为正, 商1
	010010	00010	余数、商左移一位
	110011		-y (除数)
	000101	00011	为正, 商1
	001010	00110	余数、商左移一位
	110011		-y (除数)
	111101	00110	为负, 商0
	111010	01100	余数、商左移一位
	001101		+y (除数)
	000111	01101	为正, 商1

Eg.

8、定点运算器的组成

(1) 基本组成包括:

算术逻辑运算单元 ALU: 核心部件

暂存器 (数据缓冲器): 用来存放参与计算的数据及运算结果, 它只对硬件设计者可见, 即只被控制器硬件逻辑控制或微程序所访问

通用寄存器堆: 用于存放程序中用到的数据, 它可以被软件设计者所访问。

内部总线: 用于连接各个部件的信息通道。

(2) 逻辑运算单元 ALU

片内先行进位(并行进位)

$$C_{n+1} = Y_0 + X_0 C_n$$

$$C_{n+2} = Y_1 + X_1 C_{n+1} = Y_1 + Y_0 X_1 + X_0 X_1 C_n$$

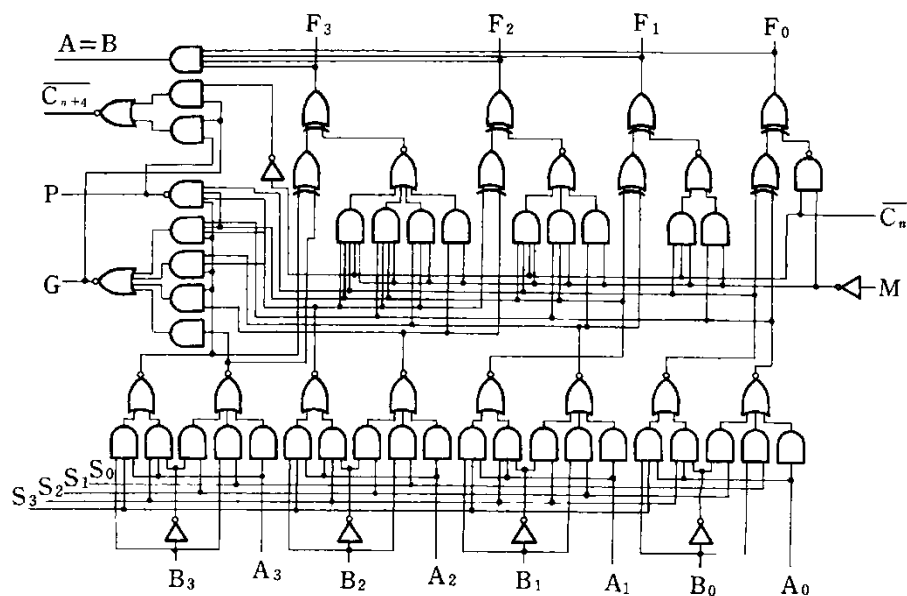
$$C_{n+3} = Y_2 + X_2 C_{n+2} = Y_2 + Y_1 X_1 + Y_0 X_1 X_2 + X_0 X_1 X_2 C_n$$

$$C_{n+4} = Y_3 + X_3 C_{n+3} = Y_3 + Y_2 X_3 + Y_1 X_2 X_3 + Y_0 X_1 X_2 X_3 + X_0 X_1 X_2 X_3 C_n$$

$$\text{令 } G = Y_3 + Y_2 X_3 + Y_1 X_2 X_3 + Y_0 X_1 X_2 X_3, \quad P = X_0 X_1 X_2 X_3$$

G 为进位发生输出, P 为进位传送输出

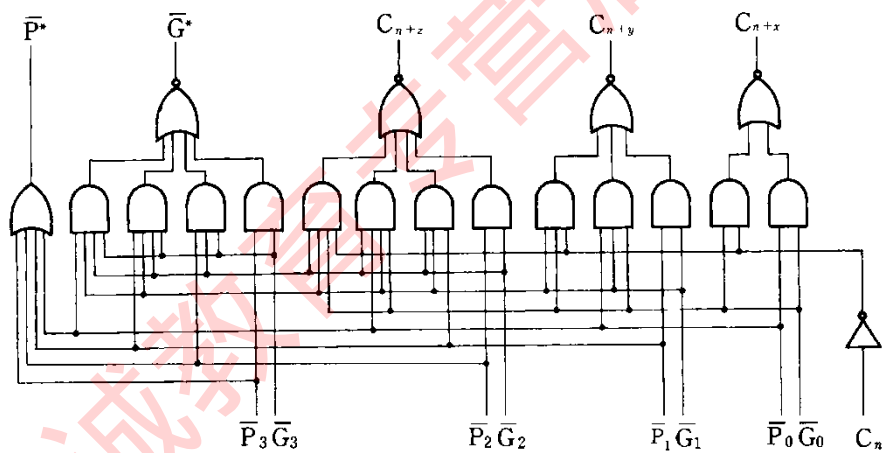
$$\text{则 } C_{n+4} = G + P C_n$$



74181ALU 逻辑图（总体）

$M=L$ 时，对进位信号没有影响，做算术运算

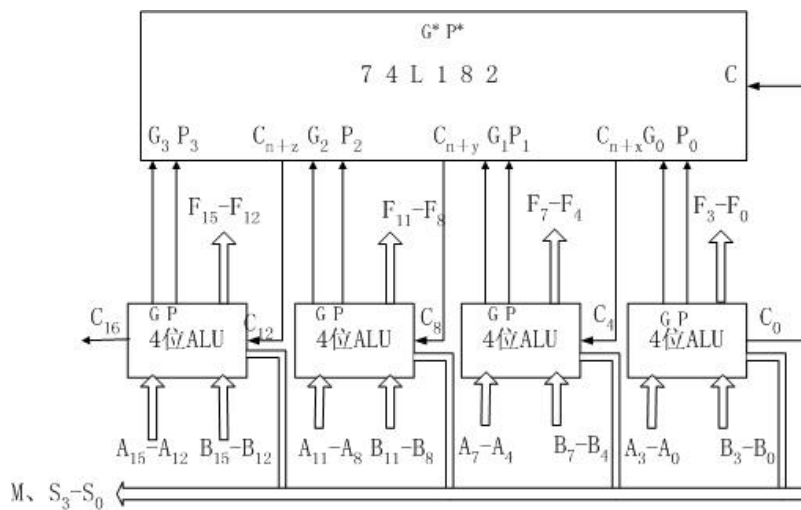
$M=H$ 时，进位门被封锁，做逻辑运算

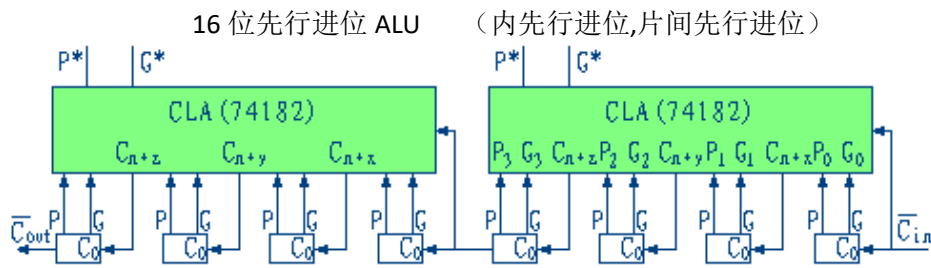


成组先行进位部件 CLA74182 逻辑图

G^* 为成组先行进位发生输出

P^* 为成组先行进位传递输出





32 位

(3) 数据总线

分类:

所处位置——内部总线 (CPU 内) 外部总线 (系统总线)

逻辑结构——单向传送总线 双向传送总线 (三态门, OG 门)

(4) 定点运算器的基本结构

①单总线结构的运算器

两个操作数分时经总线进入锁存器 A 和 B, 运算结果也通过单总线送回

②双总线结构的运算器

两个操作数通过各自的总线送加法器运算, 运算结果通过其中一总线送回

③三总线结构的运算器

两个操作数和操作结果通过各自的总线传送

9、浮点运算方法和浮点运算器

(1) 浮点加减运算

①两个浮点数 x 和 y, 它们分别为

$$x = 2^{E_x} \cdot M_x$$

$$y = 2^{E_y} \cdot M_y$$

E_x 和 E_y 分别为数 x 和 y 的阶码, M_x 和 M_y 为数 x 和 y 的尾数, 运算法则:

$$x \pm y = (M_x 2^{E_x - E_y} \pm M_y) 2^{E_y} \quad E_x \leq E_y$$

②运算步骤

- 0 操作数的检查, 看有无简化操作的可能;
- 比较阶码大小并完成对阶 (小阶向大阶对齐);
- 尾数进行加或减运算;
- 结果规格化并进行舍入处理

(1) 在浮点加减运算时, 尾数求和的结果也可以得到 **01.φ...φ** 或 **10.φ...φ**, 即两符号位不等, 此时将运算结果右移以实现规格化表示, 称为向右规格化。

(2) 结果是 **00.0..01.....** 或 **11.1...10...** 时, 则向左规格化

——舍入处理:

就近舍入(0 舍 1 入): 类似“四舍五入”, 丢弃的最高位为 1, 进 1, 否则为 0

——溢出判断和处理

阶码上溢, 超过最大值的正指数值, 一般将其认为是 $+\infty$ 和 $-\infty$ 。

阶码下溢, 超过最小值的负指数值, 则数值为 0

尾数上溢, 两个同符号尾数相加产生了最高位向上的进位。处理方法是尾数右移, 阶码加 1。

尾数下溢。尾数右移时, 最低位从最右端流出。进行要进行舍入处理。

(2) 浮点乘法运算

$$\textcircled{1} x = 2^{E_x} \cdot M_x \quad y = 2^{E_y} \cdot M_y$$

$$x * y = 2^{(E_x + E_y)} \cdot (M_x * M_y)$$

$$x / y = 2^{(E_x - E_y)} \cdot (M_x / M_y)$$

②运算步骤:

——阶码运算;

——尾数处理;

有两种方法(截尾法、舍入法)

舍入方法——

尾数用**原码**表示时

只要尾数最低为 1 或者移出位中有 1 数值位, 使最低位置 1; 0 舍 1 入

尾数用**补码**表示时 (p57 例 30)

丢失的位全为 0, 不必舍入。

丢失的最高位为 0, 以后各位不全为 0 时; 或者最高为 1, 以后各位全为 0 时, 不必舍入。

丢失的最高位为 1, 以后各位不全为 0 时, 则在尾数的最低位入 1 的修正操作。

(3) 浮点运算流水线

①原理:

一个任务分解为几个有联系的子任务。每个子任务由一个专门功能部件实现在流水线中的每个功能部件之后都要有一个**缓冲寄存器**, 或称为锁存器

流水线中**各段的时间应该尽量相等**, 否则将会引起“堵塞”和“断流”的现象
流水线需要有**装入时间和排空时间**, 只有流水线完全充满, 才能充分发挥效率

②一个具有 **k 级** 过程段的流水线处理 **n 个** 任务需要的时钟周期数为 **k+(n-1)** 所需要的时间为:

$$T_k = [k + (n - 1)] \times \tau$$

顺序完成的时间为: $T_L = n \times k \times \tau$

k 级线性流水线的加速比:

$$C_k = \frac{T_L}{T_k} = \frac{n \times k}{k + (n - 1)}$$

计算机组成与系统结构第三章总结

1、存储器概述

(1) 分类

按**存储介质**分类: 磁表面/半导体存储器

按**存取方式**分类: 随机/顺序存取(磁带)

按**读写功能**分类: **RAM**: 双极型/MOS;; **ROM**: MROM/PROM/EPROM/EEPROM/flash

按**信息的可保存性**分类: 永久性和非永久性的

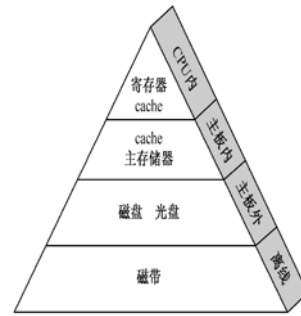
按**存储器系统中的作用**分类: 主/辅/缓

(2) 存储器的分级结构

a) 高速缓冲存储器简称 **cache**, 它是计算机系统中的一个高速小容量半导体存储器。

b) 主存储器简称**主存**, 是计算机系统的主要存储器, 用来存放计算机运行期间

的大量程序和数据。



c) 外存储器简称外存，它是大容量辅助存储器。

(3) 主存储器的技术指标

字存储单元：存放一个机器字的存储单元，相应的单元地址叫字地址。

字节存储单元：存放一个字节的单元，相应的地址称为字节地址。

存储容量：指一个存储器中可以容纳的存储单元总数。存储容量越大，能存储的信息就越多。

存取时间又称存储器访问时间：指一次读操作命令发出到该操作完成，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，故称为存储器存取时间。

存储周期：指连续启动两次读操作所需间隔的最小时间。通常，存储周期略大于存取时间，其时间单位为 ns。

存储器带宽：单位时间里存储器所存取的信息量，通常以位/秒或字节/秒做度量单位。

2、SRAM 存储器

(1) 主存（内部存储器）是半导体存储器。根据**信息存储的机理不同**可以分为两类：

静态读写存储器(SRAM)：存取速度快

动态读写存储器(DRAM)：存储容量不如 DRAM 大。

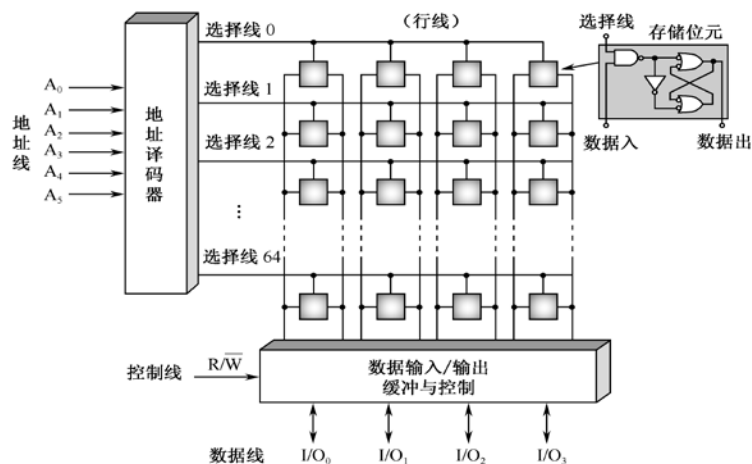
(2) 基本的静态存储元阵列

有三组信号线：

①地址线：本图中，有 6 条即 $A_0, A_1, A_2, A_3, A_4, A_5$ ；指定了存储器容量是 $2^6=64$ 个存储单元

②数据线：有 4 条即 $I/O_0, I/O_1, I/O_2, I/O_3$ 。指定了存储器的字长是 4 位，存储位元共有 $64*4=256$

③控制线：本例中 R/\overline{W} 。指定了对存储器**读**（高电平），还是**写**（低电平）



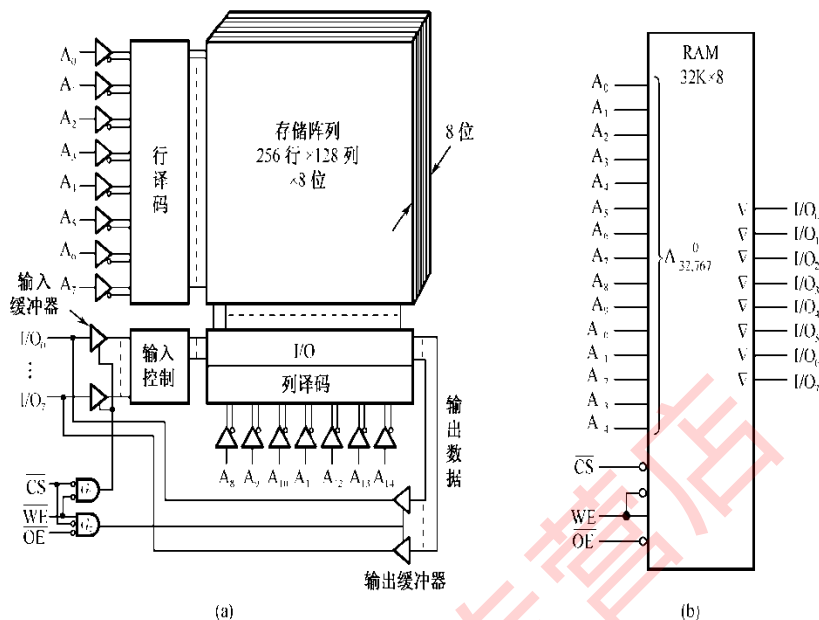
(3) 基本 SRAM 逻辑结构

多采用双译码方式；

32K*8 位，地址线 15 条，分为 x 方向 8 条，行译码输出 256 行；y 方向 7 条，列译码输出 128 列，存储阵列三维结构，即 256 行*128 列*8 位。

双向数据线有 8 条，即 I/O0---I/O7。

写入时，8 个输入缓冲器被打开，输出关闭，因而 8 条 I/O 数据线的的数据写入；读出时，8 个输出缓冲器被关闭，输入关闭，因而 8 条 I/O 数据线的的数据读出；



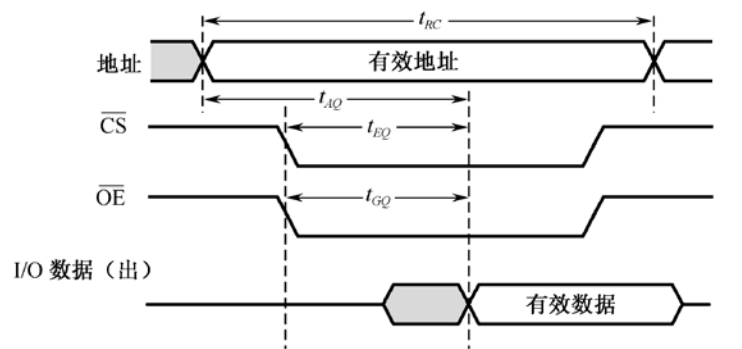
控制信号中 **CS** 是片选信号，CS 有效时（低电平），门 G1、G2 均被打开。**OE** 为读出使能信号，OE 有效时（低电平），门 G2 开启，当写命令 **WE=1** 时（高电平），门 G1 关闭，存储器进行读操作。写操作时，**WE=0**，门 G1 开启，门 G2 关闭。注意，门 G1 和 G2 是互锁的，一个开启时另一个必定关闭，这样保证了读时不写，写时不读。

(4) 存储器的读写周期

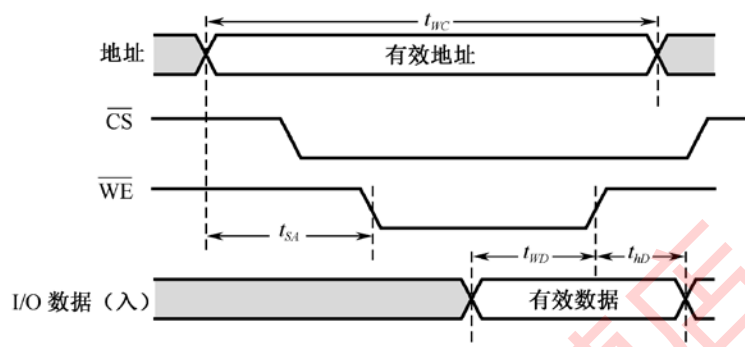
读周期——读出时间 T_{aq} 读周期时间 T_{rc}

写周期——写周期时间 T_{wc} 写时间 T_{wd}

存取周期——读周期时间 T_{rc} =写时间 T_{wd}



(a) 读周期 (\overline{WE} 高)



(b) 写周期 (\overline{WE} 低)

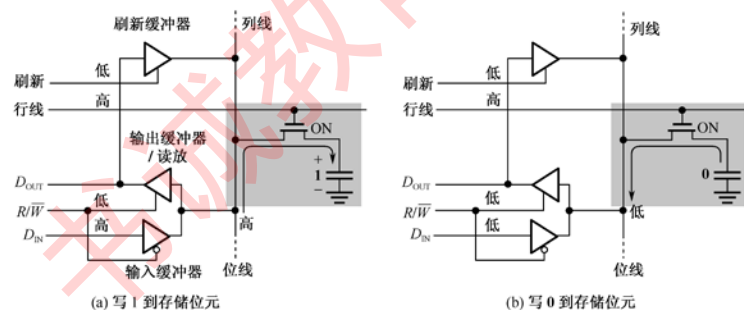
3、DRAM 存储器

(1) 记忆原理

SRAM 存储元是一个触发器，具有两个稳定状态；

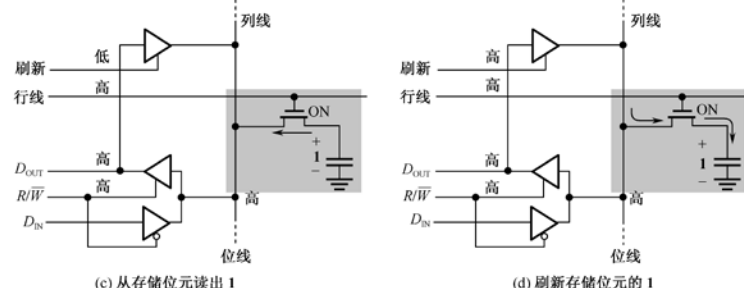
DRAM 存储元由一个 MOS 晶体管和电容器组成；

由于 (c) 中读出 1 是破坏性的，必须恢复存储位元中原存的 1；需要刷新



(a) 写 1 到存储位元

(b) 写 0 到存储位元

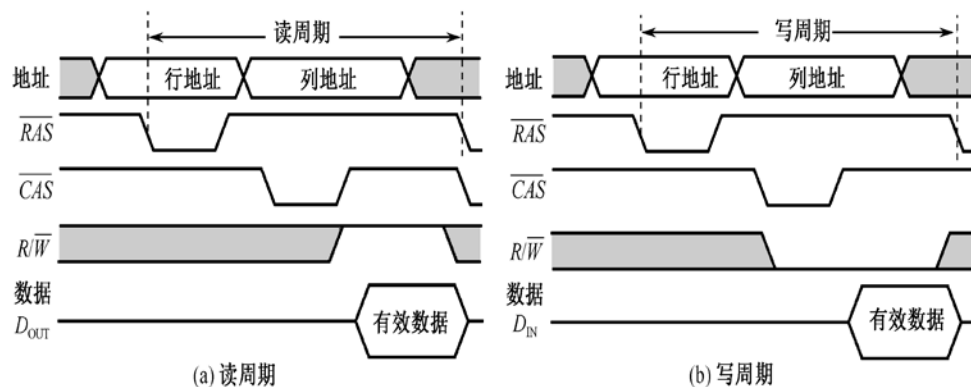


(c) 从存储位元读出 1

(d) 刷新存储位元的 1

(2) 读/写周期

定义：从行选通信号 **RAS** 下降沿开始，到下一个 **RAS** 信号的下降沿为止的时间，也即连续两个读周期的时间间隔。通常为控制方便，读周期和写周期时间相等。



(3) 刷新周期

刷新操作有两种刷新方式:

①**集中式刷新**: DRAM 的所有行在每一个刷新周期中都被刷新。

例如刷新周期为 8ms 的内存来说, 所有行的集中式刷新必须每隔 8ms 进行一次。为此将 8ms 时间分为两部分: 前一段时间进行正常的读/写操作, 后一段时间 (8ms 至正常读/写周期时间) 做为集中刷新操作时间。

★缺点: 有 1.024% 的时间, 即在 8ms 中有 $81.92 \mu s$ 不允许访存, CPU 要处于等待状态, 影响了计算机的工作效率

★优点: 控制逻辑简单, 设计容易实现

②**分散式刷新**: 每一行的刷新插入到正常的读/写周期之中。

★缺点: 控制逻辑复杂, 设计不易实现

★优点: 提高了计算机的工作效率

(4) 存储容量的扩充

①**字长位数扩充** ($1M \times 4$ 位 $\rightarrow 1M \times 8$ 位)

——三组信号线中, 地址线和控制线公用, 数据线, 单独分开连接

——公式: 所需芯片数量 = 设计要求的存储器容量 / 选择芯片存储器容量

②**字存储容量扩展** ($1M \times 8$ 位 $\rightarrow 2M \times 8$ 位)

——三组信号组中地址总线 and 数据总线公用, 控制总线中 R/W 公用, 使能端 EN 不能公用, 它由地址总线的高位段译码来决定片选信号。

——公式同上

(5) 高级的 DRAM 结构

①**FPM DRAM**: 快速页模式动态存储器

②**CDRAM**: 带高速缓冲存储器 (cache) 的动态存储器

在通常的 DRAM 芯片内又集成一个小容量的 RAM, 使 DRAM 芯片性能显著改进

③**SDRAM**: 同步型动态存储器

④**RDRAM**: 高速总线方式

4、只读存储器

(1) ROM 称为只读存储器; 主要分为两类:

掩膜 ROM 和 **可编程 ROM** (又分为一次性可编程 PROM 和多次可编程 EPROM, EEPROM);

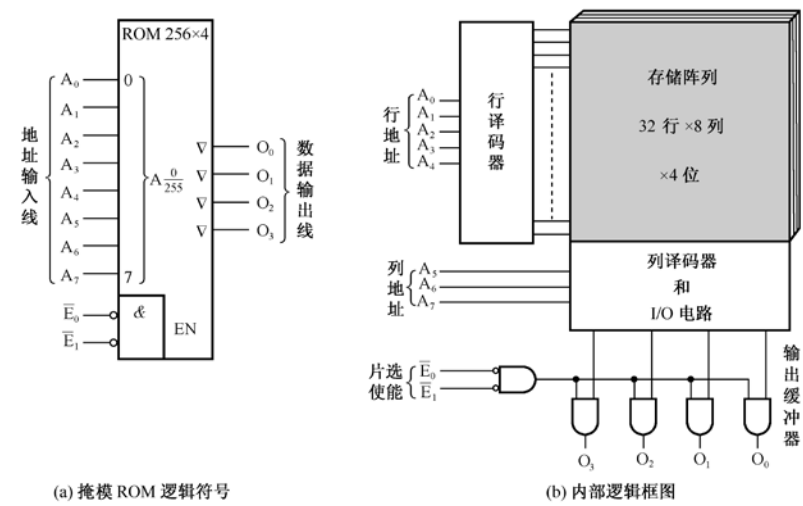
(2) 掩膜 ROM

有三组信号线: 地址线 8 条, 存储容量为 $2^8=256$;

数据线 4 条, 对应字长 4 比特;

控制线两条, E_0, E_1 , 二者是“与”的关系; 当允许 ROM 读出时,

两者相等且为低电平，输出缓冲器被打开，4 位数据 O3—O0 便输出；



- (3) 可编程 ROM
- ① EPROM——光擦除可编程可读存储器
 - ② EEPROM——电擦除可编程只读存储器

5、 闪存存储器

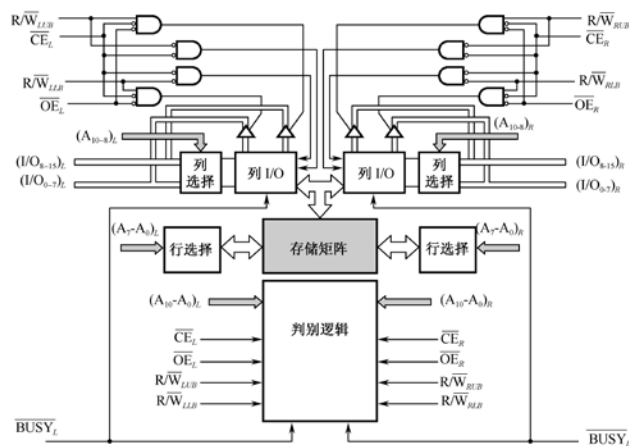
- (1) FLASH 存储器也称为闪存存储器
- (2) 基于 EPROM 存储元
- (3) FLASH 存储器的三个主要的基本操作：
 - ①编程操作：写操作。存储元初始状态全为“1”；控制栅加正电压，改写为“0”状态；控制栅不加正电压，存储元保持“1”状态。
 - ②读取操作：控制栅加正电压。存储元原存 1，浮空栅不带负电，开启晶体管；原存为 0，浮空栅带负电，晶体管不能开启导通；
 - ③擦除操作：EPROM 外部紫外光照射方式擦除，FLASH 点擦除。所有存储元中浮空栅上的负电荷全部洩放出去。故源极 S 加上正电压，与编程操作正好相反。
- (4) FLASH 存储器的阵列结构
有两种：
串行：NOR 可以随机存取；
并行：NAND, 可以顺序存取, 删除时按块操作。

6、 各种存储器的性能比较

存储器类型	非易失性	高密度	单晶体管存储元	在系统中的可写性
FLASH	√	√	√	√
SRAM	×	×	×	√
DRAM	×	√	√	√
ROM	√	√	√	×
EPROM	√	√	√	×
EEPROM	√	√	√	√

7、 并行存储器

- (1) 双端口存储器
 - ①逻辑结构



左端口和右端口：分别具有各自的地址线（A0—A10），数据线和控制线。

②无冲突读写控制

当两个端口的地址不相同，两个端口上进行读写操作，不发生冲突。任一端口被选中驱动时，就可对整个存储器进行存取，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 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。

③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，发生读写冲突。特设置 BUSY 标志。片上的判断逻辑可以决定对哪个端口优先进行读写操作，而对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。

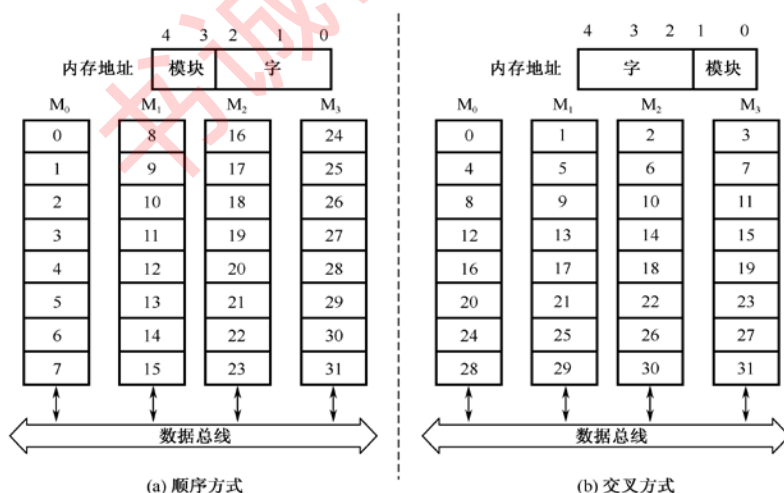
④有冲突读写判断

(1)如果地址匹配且在 CE 之前有效，片上的控制逻辑在 CE_L 和 CE_R 之间进行判断来选择端口(CE 判断)。

(2)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(地址有效判断)。

(2) 多模块交叉存储器

①存储方式：



顺序方式：高位选模块，低位选块内地址

某个模块进行存取时，其他模块不工作

优点：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：各模块串行工作，存储器的带宽受到了限制

交叉方式：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，同一个模块内的地址都是不连续的。

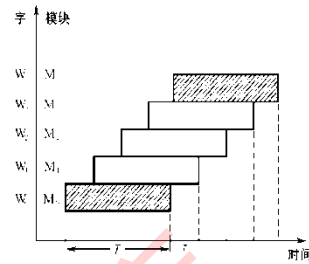
优点：对连续字的成块传送可实现多模块流水式并行存取，大大提高存储器的带宽。

使用场合为成批数据读取。

②基本结构：

通常在一个存储器周期内， m 个存储体必须分时启动，则各个存储体的启动间隔为 $\tau = T/m$ (m 为交叉存取度)

$$t_{\text{顺序}} = mT \quad t_{\text{交叉}} = T + (m-1)\tau$$



(T 存取一个字的存储周期，总线传送周期为)

8、cache 存储器

(1) 基本原理

1) cache 是介于 CPU 和主存之间的小容量存储器，但存取速度比主存快。主存容量配置几百 MB 的情况下，cache 的典型值是几百 KB。cache 能高速地向 CPU 提供指令和数据，从而加快了程序的执行速度。

2) 从功能上看，它是主存的缓冲存储器，由高速的 SRAM 组成。为追求高速，包括管理在内的全部功能由硬件实现，因而对程序员是透明的。

3) Cache 的设计依据:CPU 这次访问过的数据，下次有很大的可能也是访问附近的数据。

4) CPU 与 Cache 之间的数据传送是以字为单位,主存与 Cache 之间的数据传送是以块为单位

5) CPU 读主存时，便把地址同时送给 Cache 和主存，Cache 控制逻辑依据地址判断此字是否在 Cache 中，若在此字立即传送给 CPU，否则，则用主存读周期把此字从主存读出送到 CPU，与此同时，把含有这个字的整个数据块从主存读出送到 cache 中。

(2) Cache 命中率

①在一个程序执行期间，设 N_c 表示 cache 完成存取的总次数， N_m 表示主存完成存取的总次数， h 定义为命中率，则有 $h = N_c / (N_c + N_m)$

②若 t_c 表示命中时的 cache 访问时间， t_m 表示未命中时的主存访问时间， $1-h$ 表示未命中率，则 cache/主存系统的平均访问时间 t_a 为： $t_a = h \cdot t_c + (1-h) \cdot t_m$

③ $r = t_m / t_c$ 表示主存慢于 cache 的倍率， e 表示访问效率，则有

$$e = t_c / t_a = t_c / (h \cdot t_c + (1-h) \cdot t_m) = 1 / (h + (1-h) \cdot r) = 1 / (r + (1-r) \cdot h)$$

(3) 主存与 cash 的地址映射

① 全相联映射方式

★★映射方法——多对多

★★将地址分为两部分（块号和字），在内存块写入 Cache 时，同时写入块号标记

★★优点：冲突概率小，cache 利用高

★★缺点：比较器难实现，需要一个访问速度很快代价高的相联存储器

★★适用于小容量的 Cache

② 直接映射方式

★★映射方法——一对多

★★ $i = j \bmod m$ （其中 cache 行号为 i ，主存块号为 j ， m 为 cache 总行数）

主存的 $0, m, 2m, \dots$ 映射到 cache 的第 0 行；

主存的 $1, m+1, 2m+1, \dots$ 映射到 cache 的第 1 行；

★★地址分为三部分（tag，行号，字）

★★优点：硬件简单，成本低； 缺点：冲突概率高

★★适用于较大容量的 Cache

③ 组相联映射方式

★★前两者的结合（一对多映射）

★★Cache 分组，组间采用直接映射方式，组内采用全相联的映射方式

★★Cache 分组 U ，组内容量 V

★★组号： $q = j \bmod u$ （主存第 j 块内容拷贝到 Cache 的 q 组中的某行）

★★ $v=1$ ，则为直接相联映射方式； $u=1$ ，则为全相联映射方式

(4) 替换策略

① LFU（最不经常使用）：被访问的行计数器增加 1，换值小的行，不能反映近期 cache 的访问情况

② LRU（近期最少使用）：被访问的行计数器置 0，其他的计数器增加 1，换值大的行，符合 cache 的工作原理

④ 随机替换：从特定的行位置中随机地选取一行换出即可。

LFU（最不经常使用）					LRU（近期最少使用）					
	说明	1块	2块	3块	4块	说明	1块	2块	3块	4块
a	a进入	1	0	0	0	a进入	0	1	1	1
b	b进入	1	1	0	0	b进入	1	0	2	2
c	c进入	1	1	1	0	c进入	2	1	0	3
d	d进入	1	1	1	1	d进入	3	2	1	0
b	命中	1	2	1	1	命中	4	0	2	1
b	命中	1	3	1	1	命中	5	0	3	2
c	命中	1	3	2	1	命中	6	1	0	3
c	命中	1	3	3	1	命中	7	2	0	4
d	命中	1	3	3	2	命中	8	3	1	0
d	命中	1	3	3	3	命中	9	4	2	0
a	命中	2	3	3	3	命中	0	5	3	1
e	替换a	1	0	0	0	替换b	1	0	4	2

(5) 写操作策略

① 写回法：换出时，对行的修改位进行判断，决定是写回还是舍掉。

② 全写法：写命中时，Cache 与内存一起写

③ 写一次法：与写回法一致，但是第一次 Cache 命中时采用全写法

计算机组成与体系结构第四章总结

1、指令：就是要计算机执行某种操作的命令

2、计算机的指令有微指令、机器指令和宏指令之分

微指令是微程序级的命令，它属于硬件；

宏指令：由若干条机器指令组成的软件指令，它属于软件；

机器指令：介于微指令与宏指令之间，通常简称为指令，每一条指令可完成一个独立的

算术运算或逻辑运算操作。

3、一台计算机中所有机器指令的集合，称为这台计算机的**指令系统**

4、指令系统的发展情况

(1) **复杂指令系统计算机**：简称 CISC

(2) **精简指令系统计算机**：简称 RISC

5、指令系统的性能要求

完备性：完备性是指用汇编语言编写各种程序时，指令系统直接提供的指令足够使用，而不必用软件来实现。

有效性：有效性是指利用该指令系统所编写的程序能够**高效率地运行**。

规整性：规整性包括指令系统的**对称性、匀齐性、指令格式和数据格式的一致性**。

兼容性：系列机各机种之间具有相同的基本结构和共同的基本指令集，因而指令系统是兼容的，即**各机种上基本软件可以通用**。

6、指令格式

(1) **机器指令用机器字**来表示。表示一条指令的机器字，称为**指令字**，简称**指令**。

(2) **指令格式**，指令字用二进制代码表示的结构形式，由**操作码字段**（表征指令的操作特性与功能）和**地址码字段**（指定参与操作的操作数的地址）组成。



(3) 操作码

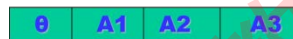
① 指令的操作码 OP 表示该指令应进行什么性质的操作

② 位数一般取决于**计算机指令系统的规模**

(4) 地址码

① 一般操作数有**被操作数**，**操作数**以及**操作结果**这三种数，形成了三地址指令格式。

② 三地址指令



操作码 0，第一操作数 A1，第二操作数 A2，结果 A3

③ 二地址指令



操作码 0 第一操作数 A1 第二操作数 A2

从操作数的物理地址归结为三种类型：

SS 存储器-存储器类型——访问内存的指令格式（**最慢**）

RS 寄存器-存储器类型——既要访问内存，又要访问寄存器

RR 寄存器-寄存器类型——访问寄存器的指令格式（**最快**）

④ 一地址指令



操作码 0 第一操作数 A1

⑤ 零地址指令



操作码 0，“停机”、“空操作”、“清除”等控制类指令。

(5) 指令字长度

指令字长度：一个指令字中包含二进制代码的位数

机器字长：计算机能直接处理的二进制数据位数，通常与主存单元的位数一致。

分类：

指令字长度等于机器字长称为**单字长指令**；等于半个机器字长度，称为**半字长指令**；等于两个机器字长度的指令，称为**双字长指令**。

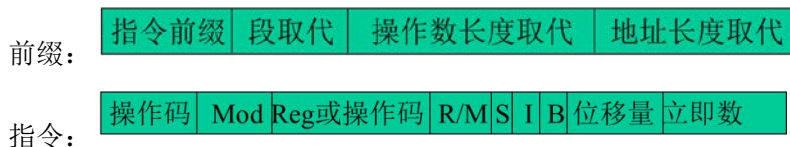
(6) 指令助记符

为了便于书写和阅读程序，每条指令通常用 **3 个或 4 个英文缩写字母**表示。

(7) Pentium 指令格式

① **指令长度可变**，最短 1 个字节，最长 12 个字节，典型的 CISC 指令系统

② **可选前缀** (0~4)、**操作码** (1~2)、一个由 **mod-R/M 字节** 和一个 **SIB 比例变址字节** 组成的**地址指定器**、一个可选的**位移量** (0~4) 和一个可选的**立即数字段** (0~4) 构成



指令前缀: 包括 **LOCK(锁定)前缀**和**重复前缀**。LOCK 前缀用于多 CPU 环境中对共享存储器的排他性访问。重复前缀用于字符串的重复操作

段取代: 用于改变默认段寄存器的情况

操作数长度取代和地址长度取代: 用于在保护模式下决定操作数和指令长度

操作码是必须的，1~2 个字节

Mod、Reg、R/M 为共 1 个字节，是可选的

mod-R/M 指定的操作数是在 **R** 中还是在 **M** 中。

mod (2 位) **R/M** (3 位) 的 32 种值构成了 8 种寄存器方式和 24 种变址方式

Reg 或 **OP** (3 位) 指定另一个操作数 (寄存器) 或用作 **OP** 的补充。

S、I、B 共 1 个字节，分别是比例系数、变址寄存器号、基址寄存器号，可选

SS (2) 指定比例变换的因子，**Index** (2) 指定变址寄存器，**Base** (3) 指定基址寄存器

位移量: 可以是 0, 1, 2, 4 个字节

立即数: 可以是 0, 1, 2, 4 个字节

7、操作数类型

地址数据:地址实际上也是一种形式的数据。

数值数据:计算机中普遍使用的三种类型的数值数据。

字符数据:文本数据或字符串，目前广泛使用 ASCII 码。

逻辑数据:一个单元中有几位二进制 bit 项组成，每个 bit 的值可以是 1 或 0。当数据以这种方式看待时，称为逻辑性数据。

8、指令和数据的寻址方式

(1) **寻址方式**:指 CPU 根据指令中给出的地址码字段寻找相应的操作数的方式，它与计算机硬件结构紧密相关，而且对指令的格式和功能有很大的影响

(2) 指令的寻址方式

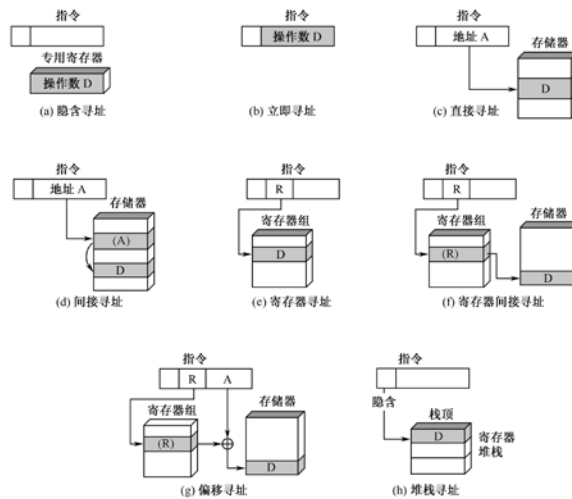
① **顺序寻址方式**

必须使用程序计数器 PC 来计数指令的顺序号

② **跳跃寻址方式**

(3) 操作数基本寻址方式

根据**操作数**放在不同的地方，从而派生各种不同的寻址方式



① 隐含寻址

指令中**隐含**着操作数的地址

② 立即寻址

指令的地址字段指出的不是操作数的地址，而是**操作数本身**
提高了指令的执行速度；操作数不能修改；大小有限；灵活性差

③ 直接寻址

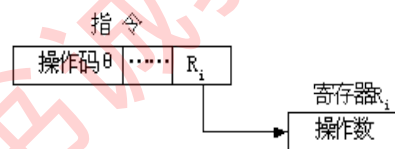
指令中地址码字段给出的**地址 A** 就是**操作数的有效地址 EA**
操作数地址不能修改；绝对寻址方式；

④ 间接寻址

指令的地址码部分给出的**地址 A** 不是操作数的地址，而是**存放操作数地址的主存单元的地址 $EA=(A)$**
至少两次访问主存，降低了执行速度

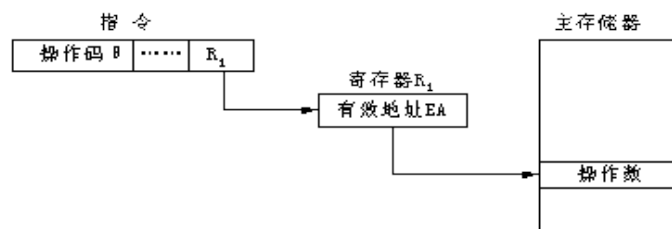
⑤ 寄存器寻址

指令的地址码部分给出 **CPU** 内某一通用寄存器的编号，指令的操作数存放在相应的**寄存器**中，即 **$EA=R_i$** （RR 型指令）



⑥ 寄存器间接寻址

操作数放在主存储器中，而操作数的**地址**放在某一通用寄存器中，然后在指令的地址码部分给出该通用寄存器的编号，这时有 **$EA=(R_i)$**



⑦ 偏移寻址——直接寻址和寄存器间接寻址方式的结合

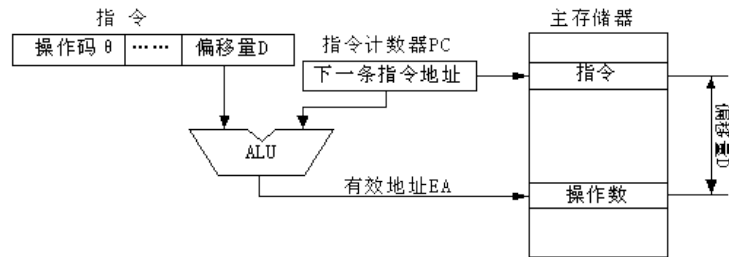
指令中有两个地址字段，至少一个是显示的。

一个地址字段的形式地址 A 直接使用；

另外一个地址字段，或基于操作码的一个隐含引用，某个专用寄存器。

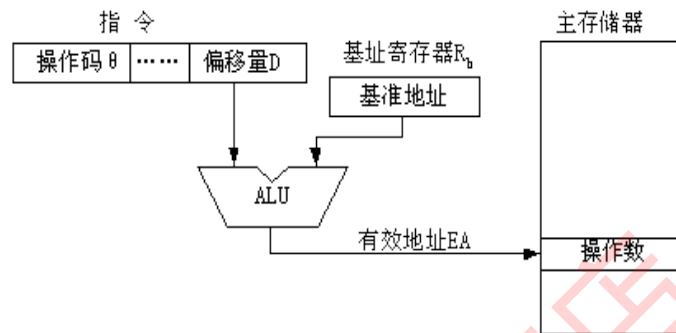
a) 相对寻址

专用寄存器是程序计数器 PC， $EA=A+(PC)$



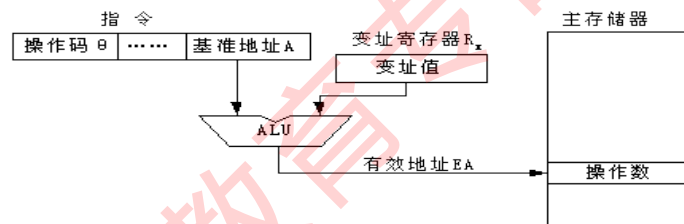
b) 基址寻址

基准地址放在基址寄存器 R_b ， $EA=(R_b)+D$

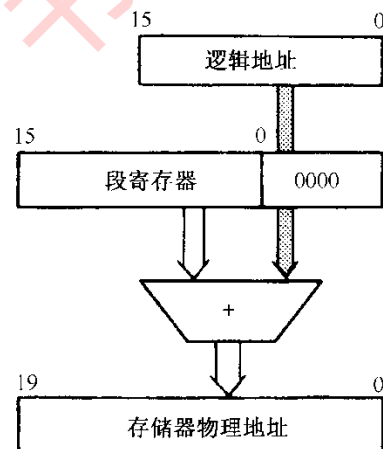


c) 变址寻址

将指令的地址码部分给出的基准地址 A 与 CPU 内某特定的变址寄存器 R_x 中的内容相加，以形成操作数的有效地址，即： $EA=A+(R_x)$



⑧ 段寻址



⑨ 堆栈寻址

寄存器堆栈和存储器堆栈
先进后出

PUSH, 压入栈顶地址, 指示器减 1; POP, 栈顶地址弹出, 指示器加 1

9、典型指令

(1) 指令系统应当有**数据处理、数据存储、数据传送、程序控制**四大类指令

数据传送类指令

一般传送指令: MOV AX, BX

数据交换指令: XCHG

堆栈操作指令: PUSH, POP

运算类指令

算术运算指令、逻辑运算指令、移位指令

程序控制类指令

程序控制类指令用于控制程序的**执行方向**, 并使程序具有**测试、分析与判断**的能力。

输入和输出指令、字符串处理指令、特权指令、其他指令

计算机组成与体系结构第五章总结

1、CPU 的功能与组成

(1) CPU 的功能:

取指令 $\xrightarrow{\text{操作控制、时间控制}}$ 执行指令

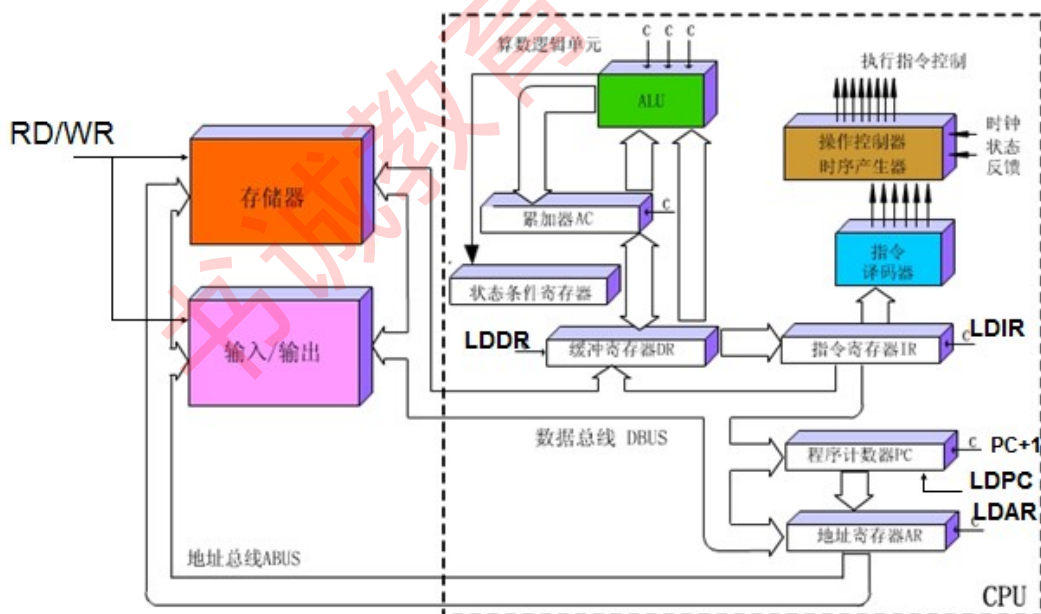
指令控制 (程序的顺序控制)

操作控制 (一条指令有若干操作信号实现)

时间控制 (指令各个操作实施时间的定时)

数据加工 (算术运算和逻辑运算)

(2) CPU 的基本组成:



① 早期 cpu 由**运算器**和**控制器**组成

② **运算器:**

算术逻辑单元 ALU、通用寄存器、数据缓冲器 DR 和状态条件寄存器 PSW 组成
主要执行所有的算术运算; 执行所有的逻辑运算, 并进行逻辑测试

③ **控制器:** 有**同步控制、异步控制和混合控制**等三种不同的控制方式

a) **组成:** 程序计数器、指令寄存器、数据缓冲器、地址寄存器、通用寄存器、状

态寄存器、时序发生器、指令译码器、总线（数据通路）

b) 程序计数器 PC

用来存放正在执行的指令的地址或接着将要执行的下一条指令的地址；

顺序执行时，每执行一条指令，PC 的值应加 1；

要改变程序执行顺序的情况时，一般由转移类指令将转移目标地址送往 PC，可实现程序的转移。

c) 指令寄存器 IR

指令寄存器用来存放从存储器中取出的待执行的指令；

在执行该指令的过程中，指令寄存器的内容不允许发生变化，以保证实现指令的全部功能。

d) 指令译码器 ID

暂存在指令寄存器中的指令只有在其操作码部分经译码后才能识别出是一条什么样的指令；

译码器经过对指令进行分析和解释，产生相应的控制信号提供给时序控制信号形成部件。

e) 机器周期、工作节拍、脉冲及启停控制线路

由脉冲源产生一定频率的脉冲信号作为整个机器的时钟脉冲

f) 时序控制信号形成部件

时序控制信号形成部件又称微操作信号发生器，真正控制各部件工作的微操作信号是由指令部件提供的操作信号、时序部件提供的时序信号、被控制功能部件所反馈的状态及条件综合形成的。

g) 地址形成部件

根据指令的不同寻址方式，用来形成操作数的有效地址；

功能就是指令流出的控制，实质上就是对取指令的控制；

指令分析与执行的控制，对指令流中的每条指令进行分析解释，根据指令的操作性质和寻址方式形成操作数的地址，然后根据该操作数的地址找到相应的存储单元，并从中取出指令执行过程中要用到的操作数，最后还要形成相应的操作控制信号序列，通过运算器、存储器及输入 / 输出设备的动作，来实现这条指令的功能；

指令流向的控制，指令流向的控制即下条指令地址的形成控制；

数据缓冲器、状态条件寄存器；

(3) CPU 中的主要寄存器



① 数据缓冲寄存器 DR

暂时存放 ALU 的运算结果，或由数据存储器读出的一个数据字，或来自外部接口的一个数据字；

② 指令寄存器 IR

用来保存当前正在执行的一条指令；

③ 程序计数器 PC

又称为**指令计数器**；

④ 数据地址寄存器 AR

保存当前 CPU 所访问的数据 cache 存储器中单元的地址

⑤ 通用寄存器(R0—R3)

当 ALU 执行算数或逻辑运算时，为 ALU 提供一工作区

⑥ 状态字寄存器 PSW

保存由算数指令和逻辑指令运算或测试结果建立的各种条件代码

(4) 操作控制器和时序产生器

① 数据通路：许多寄存器之间传送信息的通路

② 操作控制器：为数据通路的建立提供各种操作信号。操作信号提供的依据是指令操作码和时序信号，主要有三种类型：

a) 硬布线控制类型

采用组合逻辑技术；

时序控制信号形成部件是由门电路组成的复杂树形网络；

速度快，但时序控制信号形成部件的结构不规整,难以实现设计自动化

b) 微程序控制类型

采用存储逻辑；

把微操作信号代码化，使每条机器指令转化成为一段微程序并存入一个专门的存储器(控制存储器)中，微操作控制信号由微指令产生；

具有设计规整、调试、维修以及更改、扩充指令方便的优点，易于实现自动化设计；

指令执行速度比组合逻辑控制器慢；

c) 混合类型

PLA 控制器，吸收前两种的设计思想实现

③ 时序产生器

提供定时和时序信号

2、指令周期

(1) 基本概念

①指令周期：指取指令、分析指令到执行完该指令所需的全部时间。

②机器周期：(CPU 周期)

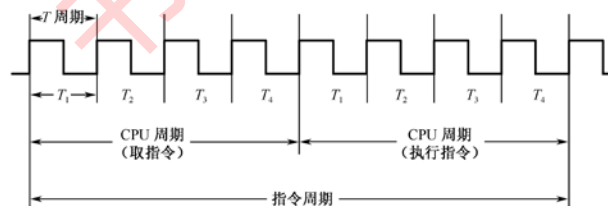
把一条指令周期划分为若干个机器周期，每个机器周期完成一个基本操作；

不同的指令，可能包含不同数目的机器周期；

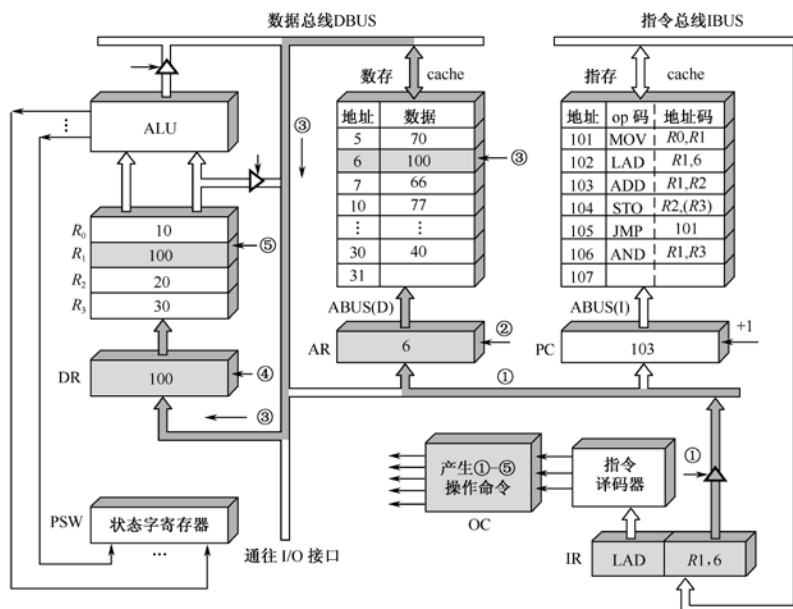
一个机器周期中，包含若干个时钟周期；

③时钟周期：处理操作的最基本单位

把一个机器周期分为若干个相等的时间段，每一个时间段称为一个节拍。



(2) 几种指令的指令周期



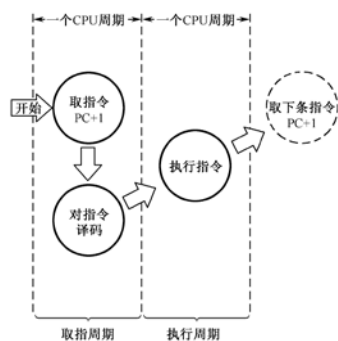
① MOV 指令

取值周期:

- ① 程序计数器 PC 中装入第一条指令地址 101 (八进制);
- ② PC 的内容被放到指令地址总线 ABUS (I) 上, 对指存进行译码, 并启动读命令;
- ③ 从 101 号地址读出的 MOV 指令通过指令总线 IBUS 装入指令寄存器 IR;
- ④ 程序计数器内容加 1, 变成 102, 为取下一条指令做好准备;
- ⑤ 指令寄存器中的操作码 (OP) 被译码;
- ⑥ CPU 识别出是 MOV 指令, 至此, 取指周期即告结束。

执行周期:

- ① 操作控制器 (OC) 送出控制信号到通用寄存器, 选择 R1 (10) 作源寄存器, 选择 R0 作目标寄存器;
- ② OC 送出控制信号到 ALU, 指定 ALU 做传送操作;
- ③ OC 送出控制信号, 打开 ALU 输出三态门, 将 ALU 输出送到数据总线 DBUS 上。注意, 任何时候 DBUS 上只能有一个数据。
- ④ OC 送出控制信号, 将 DBUS 上的数据打入到数据缓冲寄存器 DR (10)
- ⑤ OC 送出控制信号, 将 DR 中的数据 10 打入到目标寄存器 R0, R0 的内容由 00 变为 10。至此, MOV 指令执行结束。



②LAD 指令

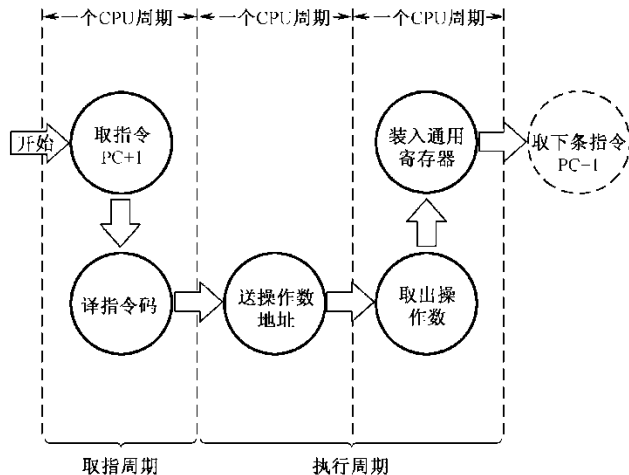
取值周期:

与 MOV 指令取值周期中一样, 只是 PC 指令地址为 102, 从指令存储器读出 “LDA R1, 6” 指

令放入 IR 中，然后 PC+1，使 PC 内容为 103，为下条指令 ADD 做准备

执行周期：2 个 CPU 周期

- ① 操作控制器 OC 发出控制命令打开 IR 输出三态门，将指令中的直接地址码 6 放到数据总线 DBUS 上；
- ② OC 发出操作命令，将地址码 6 装入数存地址寄存器 AR；
- ③ OC 发出读命令，将数存 6 号单元中的数 100 读出到 DBUS 上；
- ④ OC 发出命令，将 DBUS 上的数据 100 装入缓冲寄存器 DR；
- ⑤ OC 发出命令，将 DR 中的数 100 装入通用寄存器 R1，原来 R1 中的数 10 被冲掉。至此，LAD 指令执行周期结束。



③ADD 指令

取值周期：与 MOV 相同；

执行周期：一个 CPU 周期

- ① 操作控制器 OC 送出控制命令到通用寄存器，选择 R1 做源寄存器，R2 做目标寄存器；
- ② OC 送出控制命令到 ALU，指定 ALU 做 R1 (100) 和 R2 (20) 的加法操作；
- ③ OC 送出控制命令，打开 ALU 输出三态门，运算结果 120 放到 DBUS 上；
- ④ OC 送出控制命令，将 DBUS 上数据打入缓冲寄存器 DR；ALU 产生的进位信号保存状态字寄存器在 PSW 中。
- ⑤ OC 送出控制命令，将 DR (120) 装入 R2，R2 中原来的内容 20 被冲掉。至此 ADD 指令执行周期结束。

④STO 指令

取值周期：与 MOV 相同

执行周期：2 个 CPU 周期

- ① 操作控制器 OC 送出操作命令到通用寄存器，选择 (R3) = 30 做数据存储器的地址单元；
- ② OC 发出操作命令，打开通用寄存器输出三态门（不经 ALU 以节省时间），将地址 30 放到 DBUS 上；
- ③ OC 发出操作命令，将地址 30 打入 AR，并进行数存地址译码；
- ④ OC 发出操作命令到通用寄存器，选择 (R2) = 120，作为数存的写入数据；
- ⑤ OC 发出操作命令，打开通用寄存器输出三态门，将数据 120 放到 DBUS 上。
- ⑥ OC 发出操作命令，将数据 120 写入数存 30 号单元，它原先的数据 40 被冲掉。至此，STO 指令执行周期结束。

注意：DBUS 是单总线结构，先送地址 (30)，后送数据 120，必须分时传送

⑤JMP 指令

取值周期：与 MOV 相同

执行周期：一个 CPU 周期

① OC 发生操作控制命令，打开指令寄存器 IR 的输出三态门，将 IR 中的地址码 101 发送到 DBUS 上；

② CC 发出操作控制命令，将 DBUS 上的地址码 101 打入到程序计数器 PC 中，PC 中的原先内容 106 被更换。于是下一条指令不是从 106 号单元取出，而是转移到 101 号单元取出。至此 JMP 指令执行周期结束。

(3) 用方框图语言表示指令周期

①方法：

指令系统设计（模型机的五指令系统）

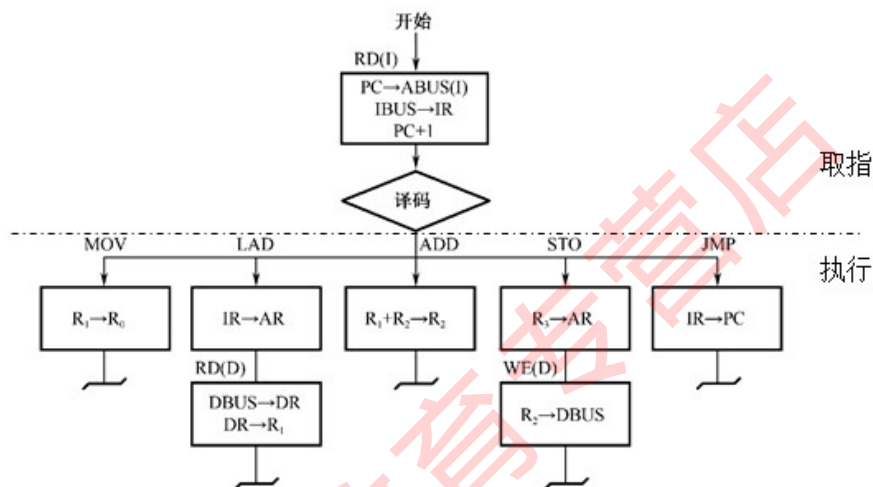
方框——按 CPU 周期

方框内内容——数据通路操作或控制操作

菱形符号——判别或测试

~——公操作

前边所讲述的 5 种操作的框图描述



②

3、时序列产生器和控制方式

(1) 时序信号的作用和体制

作用：

CPU 中的控制器用它指挥机器的工作；

用时序信号/周期信息来辨认从内存中取出的是指令（取指）还是数据（执行）

一个 CPU 周期中时钟脉冲对 CPU 的动作有严格的约束

操作控制器发出的各种信号是时间（时序信号）和空间（部件操作信号）的函数。

硬布线控制器的体制：

主状态周期---节拍电位---节拍脉冲 三级体制；

一个节拍电位表示一个 CPU 周期的时间；

主状态周期包含若干个节拍电位；用一个触发器的状态持续时间来表示；

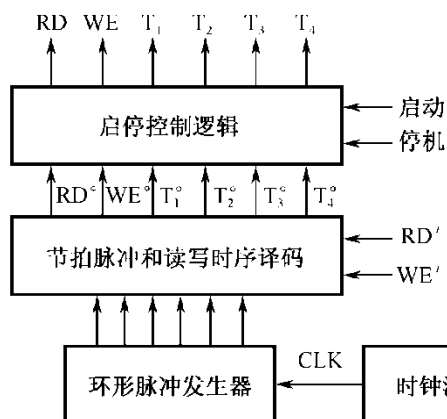
微程序控制器的体制：

节拍电位—节拍脉冲二级体制

(2) 时序信号产生器：

①功能：产生时序信号

②构成：



a) 时钟源:

用来为环形脉冲发生器提供频率稳定且电平匹配的方波时钟脉冲信号;

b) 环型脉冲发生器:

产生一组有序的间隔相等或者不等的脉冲序列;

毛刺产生原因: 电路内部原因以及寄存器参数的影响

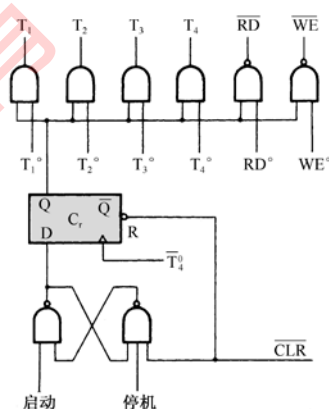
避免方法: 采用循环移位寄存器

c) 节拍脉冲和存储器读写时序:

d) 启停控制逻辑:

当运行触发器为“1”时, 打开时序电路。当计算机启动时, 一定要从第 1 个节拍脉冲前沿开始工作。

当运行触发器“0”时, 关闭时序产生器。停机时一定要在第 4 个节拍脉冲结束后关闭时序产生器。



(3) 控制方式:

控制方式: 控制不同操作序列时序信号的方法。常用的有以下三种控制方式:

①同步控制方式 (指令的机器周期和时钟周期数不变)

完全统一的机器周期执行各种不同的指令

采用不定长机器周期

中央控制于局部控制的结合

②异步控制方式

每条指令需要多长时间就占多长时间

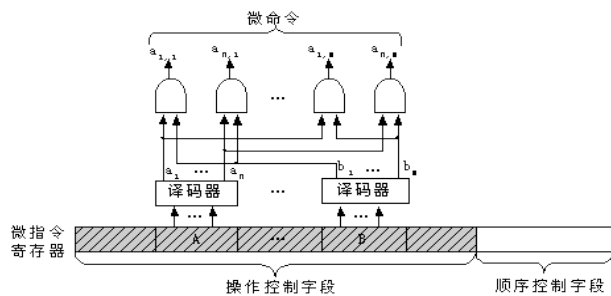
③联合控制方式

大部分指令在固定的周期内完成, 少数难以确定的操作采用异步方式

机器周期的节拍脉冲固定, 但是各指令的机器周期数不固定 (微程序控制器采用)

三、混合表示法:

前两种结合。一个字段的某些编码不能独立地定义某些微命令，而需要与其他字段的编码来联合定义



编码注意：字段编码法中操作控制字段并非是任意的，必须要遵循如下的原则：

①把互斥性的微命令分在同一段内，兼容性的微命令分在不同段内。这样不仅有助于提高信息的利用率，缩短微指令字长，而且有助于充分利用硬件所具有的并行性，加快执行的速度。

②应与数据通路结构相适应。

③每个小段中包含的信息位不能太多，否则将增加译码线路的复杂性和译码时间。

④一般每个小段还要留出一个状态，表示本字段不发出任何微命令。因此当某字段的长度为三位时，最多只能表示七个互斥的微命令，通常用 **000** 表示不操作。

②后继微地址的形成方法：

一、计数器方式：

方法：

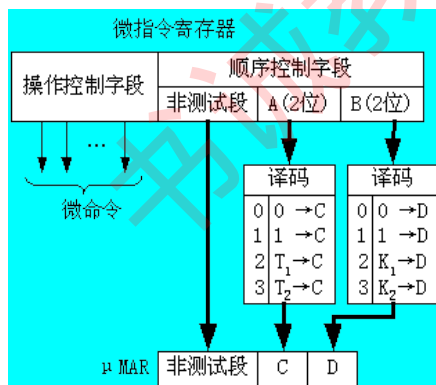
——微程序顺序执行时，其后继微地址就是现行微地址加上一个增量(通常为 1)；

——当微程序遇到转移或转子程序时，由微指令的转移地址段来形成转移微地址。

——在微程序控制器中也有一个微程序计数器 μPC ，一般情况下都是将微地址寄存器 μMAR 作为 μPC

二、多路转移方式：

状态条件/测试/微指令中微地址/操作码



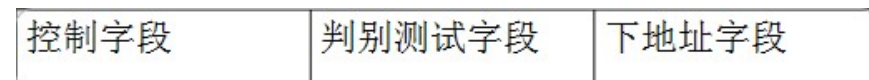
测试段如果只有一位，则微地址将产生两个分支，若有两位，则最多可产生四个分支，依此类推，测试段为 n 位最多可产生 2^n 个分支

③微指令格式：

微指令格式分为两类：水平型微指令和垂直型微指令

(1) 水平型微指令：

一次能定义并能并行执行多个微命令的微指令。



优点：

微指令字较长，速度越快。
微指令中的微操作有高度的并行性。
微指令译码简单。
控制存储器的纵向容量小，灵活性强。

缺点：

微指令字比较长，明显地增加了控制存储器的横向容量。
水平微指令与机器指令差别很大，一般要熟悉机器结构、数据通路、时序系统以及指令执行过程的人才能进行微程序设计，这对用户来说是很困难的

(2) 垂直型微指令：

一次只能执行一到二个微命令的微指令称为垂直型微指令

15	13 12	8 7	3 2	0
000	源寄存器编址	目标寄存器编址	其他	

微指令字短，一般为 10~20 位左右。
微指令的并行微操作能力有限，一条微指令一般只包含一个微操作命令。
微指令译码比较复杂。全部微命令用一个微操作控制字段进行编码，微指令执行时需行完全译码。
垂直微指令字较短，使控制存储器的横向容量少
不能充分利用数据通路具有多种并行操作能力
设计用户只需注意微指令的功能，而对微命令及其选择、数据通路的结构则不用过多地考虑，因此，便于用户编制微程序。而且，编制的微程序规整、直观，便于实现设计的自动化；

(3) 二者比较：

水平型并行操作能力强，效率高，灵活性强；
水平型微指令执行一条指令的时间短；
由水平型微指令解释指令的微程序，有微指令字较长而微程序短的特点；
水平型微指令用户难以掌握，垂直型微指令与指令比较相似，比较容易掌握；

计算机组成与体系结构第七章总结

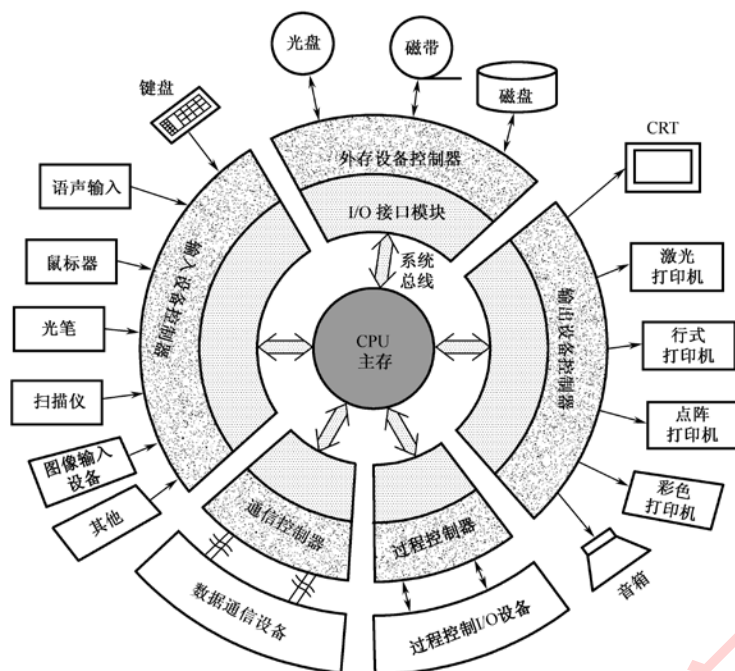
1、外围设备概述

外围设备大体分为**输入设备、输出设备、外存设备、数据通信设备、过程控制设备**五大类

(1) 外围设备三个基本组成部分：

- (1)存储介质**，它具有保存信息的物理特征。例如磁盘就是一个存储介质的例子，它是用记录在盘上的磁化元表示信息。
- (2)驱动装置**，它用于移动存储介质。例如，磁盘设备中，驱动装置用于转动磁盘并进行定位。
- (3)控制电路**，它向存储介质发送数据或从存储介质接受数据。例如，磁盘读出时，控制电路把盘上用磁化元形式表示的信息转换成计算机所需要的电信号，并把这些信号用电缆送给计算机主机。

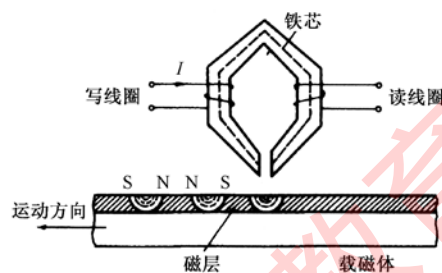
(2) 外围设备分类：



2、磁盘存储设备

(1) 磁记录原理:

磁表面存储器的读写原理



写操作:

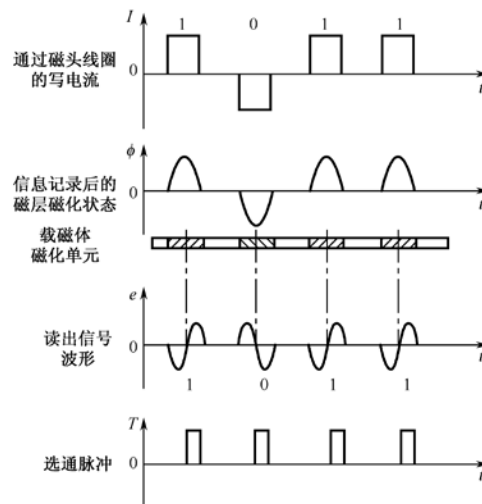
当写线圈中通过一定方向的脉冲电流时，铁芯内就产生一定方向的磁通。

读操作:

当磁头经过载磁体的磁化元时，由于磁头铁芯是良好的导磁材料，磁化元的磁力线很容易通过磁头而形成闭合磁通回路。不同极性的磁化元在铁芯里的方向是不同的。

记录方式的写读波形图

通过电磁变换，利用磁头写线圈中的脉冲电流，可把一位二进制代码转换成载磁体存储元不同剩磁状态；反之，通过磁电变换，利用磁头读出线圈，可将由存储元的不同剩磁状态表示的二进制代码转换成电信号输出。这就是磁表面存储器存取



信息的原理。

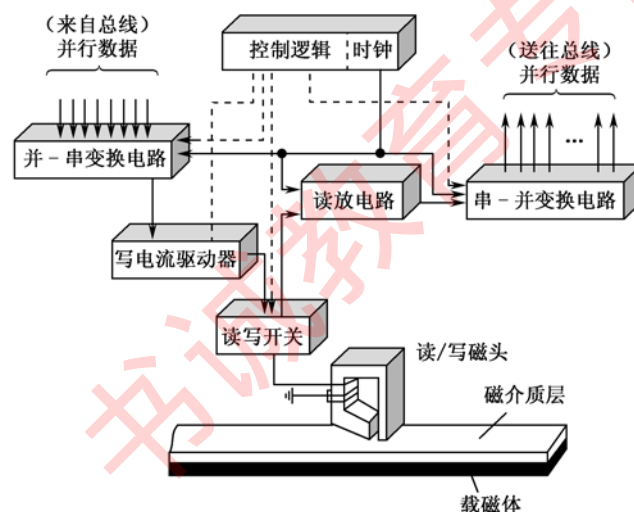
磁层上的存储元被磁化后，它可以供多次读出而不被破坏。当不需要这批信息时，可通过磁头把磁层上所记录的信息全部抹去，称之为写“0”。通常，写入和读出是合用一个磁头，故称之为读写磁头。每个读写磁头对应着一个信息记录磁道。

(2) 磁盘的组成和分类：

下图表示了存取功能的逻辑结构，主要由**磁记录介质**，**磁盘控制器**，**磁盘驱动器**三大部分组成：

磁盘控制器包括控制逻辑与时序、数据并—串变换电路和串—并变换电路

磁盘驱动器包括写入电路与读出电路、读写转换开关、读写磁头与磁头定位伺服系统



组成：写入时，将计算机并行送来的数据取至并串变换寄存器，变为串行数据，然后一位一位地由写电流驱动器作功率放大并加到写磁头线圈上产生电流，从而在盘片磁层上形成按位的磁化存储元。**读出时**，当记录介质相对磁头运动时，位磁化存储元形成的空间磁场在读磁头线圈中产生感应电势，此读出信息经放大检测就可还原成原来存入的数据。由于数据是一位一位串行读出的，故要送至串—并变换寄存器变换为并行数据，再并行送至计算机。

分类：硬磁盘按照**盘片结构**，分成**可换盘片式**与**固定盘片式**两种；**磁头**也分为**可移动磁头**和**固定磁头**两种。

可移动磁头固定盘片磁盘机：一片或一组盘片固定在主轴上，盘片不可更换。盘片没面只有一个磁头，存取数据时磁头沿盘面径向移动。

固定磁头盘片机：磁头位置固定，磁盘的每一个磁道对应一个磁头，盘片不可更换。存取速度快，省去磁头找道时间，但结构复杂。

可移动磁头可换盘片磁盘机：盘片可以更换，磁头可沿盘面径向移动。盘片可以脱机保存，同种型号的盘片具有互换性。

温彻斯特磁盘机：简称**温盘**，先进技术研制的可移动磁头固定盘片磁盘机。密封组合式的硬磁盘。防尘性能好，可靠性高，对使用环境要求不高。

(3) 磁盘启动器和控制器

一、**磁盘驱动器**是一种精密的电子和机械装置；

温盘驱动器，还要求在**超净环境**下组装；

主要由**定位驱动系统、主轴系统**和**数据转换系统**组成；

定位驱动系统——驱动部件，传动部件，运载部件组成。主要采用步进电机和音圈电机两种驱动方式。

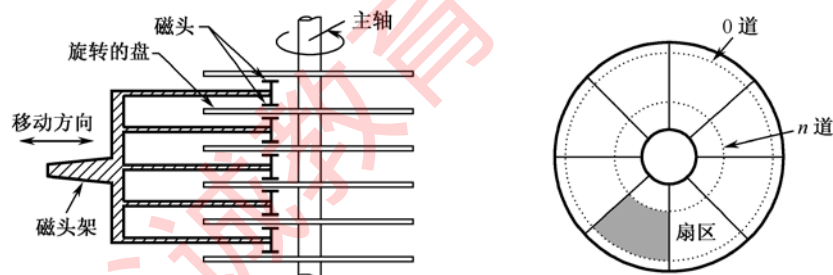
主轴系统——安装盘片，驱动他们以额定转速稳定旋转。主要部件是主轴电机和有关控制电路。

数据转换系统——控制数据的写入和读出，包括磁头、磁头选择电路、读写电路以及索引、区标电路等

二、**磁盘控制器**——主机与磁盘驱动器之间的接口。采用成批交换数据方式。**两个方面的接口：**一个是与主机的接口，控制外存与主机**总线**之间交换数据（**系统级接口**）；一个是与设备的接口，根据主机命令控制设备的操作（**设备级接口**）。

(4) 磁盘上信息分布

盘片的上下两面都能记录信息，通常把**磁盘片表面**称为**记录面**。记录面上一系列同心圆称为**磁道**。每个盘片表面通常有几百到几千个磁道，**每个磁道**又分为若干个**扇区**。从图中看出，**外面扇区比里面扇区面积要大**。磁盘上的这种磁道和扇区的排列称为**格式**。



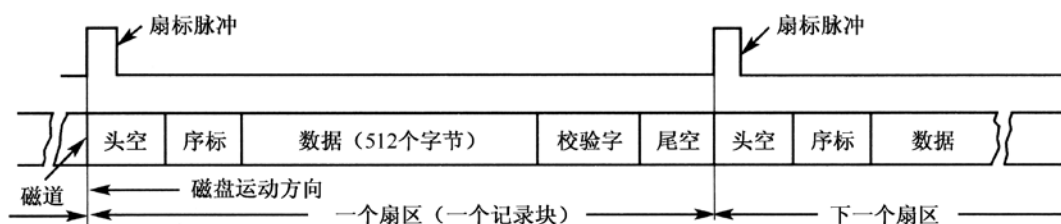
磁道的编址从外向内编号，最外一个同心圆叫**0 磁道**，最里面叫**n 磁道**；**n 磁道**里面的圆面积不用来记录信息；

在磁道上，信息按区存放，各个区存放的字或字节数相同；

读写操作是以**扇区**为单位；每一个扇区记录一个记录块；

每个扇区开始时由磁盘控制器产生一个扇区脉冲。标志一个扇区的开始。每个记录块由头部空白段、序标段、数据段、检验字段和尾部空白段组成。

数据在磁盘上的记录格式



(5) 磁盘存储器的技术标准

①**存储密度**：存储密度分道密度、位密度和面密度

道密度：沿磁盘半径方向单位长度上的磁道数，单位为道/英寸。

位密度：磁道单位长度上能记录的二进制代码位数，单位为位/英寸。

面密度：位密度和道密度的乘积，单位为位/平方英寸。

②**存储容量**：一个磁盘存储器所能存储的字节总数，称为磁盘存储器的存储容量

③**存取时间**：存取时间是指从发出读写命令后，磁头从某一起始位置移动至新的记录位置，到开始从盘片表面读出或写入信息加上传送数据所需要的时间。取决于以下三个因素决定：

一个是将磁头定位至所要求的磁道上所需的时间，称为**找道时间**；

第二个是找道完成后至磁道上需要访问的信息到达磁头下的时间，称为**等待时间**，这两个时间都是随机变化的，因此往往使用平均值来表示，平均找道时间是最大找道时间与最小找道时间的平均值。平均等待时间和磁盘转速有关，它用磁盘旋转一周所需时间的一半来表示。

第三个是**数据传送时间**。

④**数据传输率**：磁盘存储器在单位时间内向主机传送数据的字节数，叫数据传输率，传输率与**存储设备和主机接口逻辑**有关。从主机接口逻辑考虑，应有足够快的传送速度向设备接收/发送信息。从存储设备考虑，假设磁盘**旋转速度为 n 转/秒**，**每条磁道容量为 N 个字节**，则数据传输率：

$$Dr=nN(\text{字节/秒}) \text{ 或 } Dr=D \cdot v(\text{字节/秒})$$

3、磁盘存储设备的技术发展

自行阅读 P217---P219

磁盘阵列 RAID 是多台磁盘存储器组成的大容量外存系统，它实现数据的并行存储、交叉存储，单独存储，改善了 I/O 性能，增加了存储容量，是一种先进的硬磁盘体系结构。各种可移动硬盘的诞生，是磁盘先进技术的又一个重要进展。

4、磁带存储设备

磁带机的记录原理与磁盘机基本相同，只是它的载磁体是一种带状塑料，叫做**磁带**；

写入时可通过磁头把**信息代码记录在磁带上**；

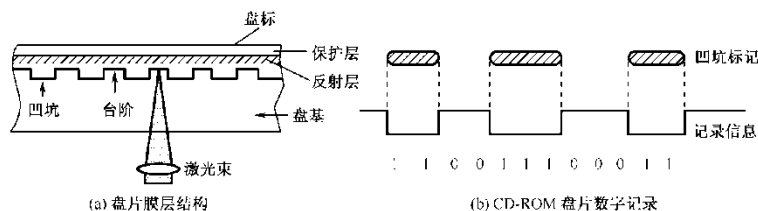
磁带存储设备由**磁带机和磁带**两部分组成，它通常作为**海量存储设备**的数据备份；

速度比磁盘速度慢——磁带上的数据采用顺序访问方式，而磁盘则采用随机访问方式。

5、光盘和磁光盘存储设备

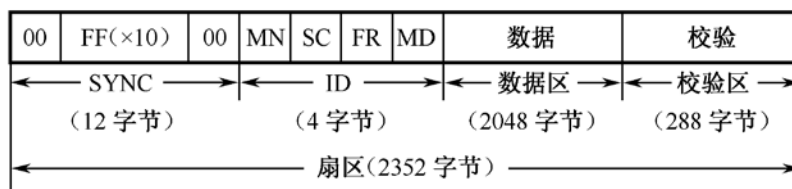
(1) 光盘存储设备：

只读型光盘系统都基于一个**共同原理**，即光盘上的信息以坑点形式分布，有坑点表示为“1”，无坑点表示为“0”，一系列的坑点(存储元)形成信息记录道。



——光盘的记录信息以**凹坑方式永久性存储**。读出时，当激光束聚焦点照射在凹坑上将发生衍射，反射率低；而聚焦点照射在凸面上时大部分光将返回。根据反射光的光强变化并进行光 电转换，即可读出记录信息。

信息记录轨迹称为**光道**。光道上划分出一个个**扇区**，它是光盘的**最小可寻址单位**。



扇区分为 4 个区域：2 个全 0 字节和 2 个全 1 字节组成**同步 (SYNC) 区**，标志开始；4 字节的**扇区标识 (ID) 区**用于说明此扇区的地址和工作模式；以分 (MN)，秒 (SC) 和分数秒 (FR， $1/75s$) 时间值作为地址。

ID 区的 MD 为模式控制，用于控制数据区和校验区的使用。共有三种模式：

0 规定数据区和校验区的全部 2336 个字节都是 0，不用于记录数据，用于光盘的导入区和导出区；

1 规定 288 字节的校验区为 4 字节的检测码 (EDC)、8 字节的保留域 (未定义) 和 276 字节的纠错码 (ECC)，有 2048 字节的数据并有很强的检测和纠错能力；

2 规定 288 字节的校验区也用于存放数据，用于保存声音，图像等数据。

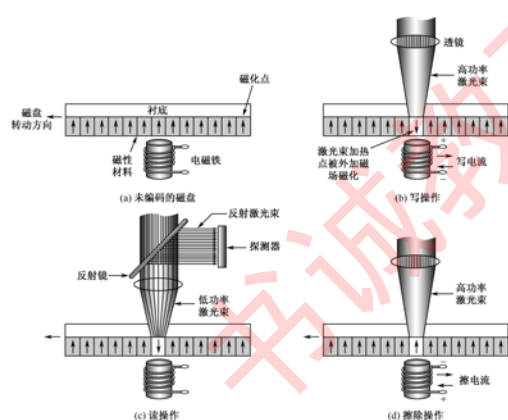
(2) 磁光盘存储设备：

采用磁场技术和激光技术相结合的产物。

重写型光盘，随机写入、擦除或重写信息；

基本工作原理是：利用**热磁效应**写入数据：当激光束将磁光介质上的记录点加热到居里点温度以上时，外加磁场作用改变记录点的磁化方向，而不同的磁化方向可表示数字“0”和“1”。利用磁光克尔效应**读出数据**：当激光束照射到记录点时，记录点的磁化方向不同，会引起反射光的偏振面发生不同结果，从而检测出所记录的数据“1”或“0”。

四种基本操作情况：



图(a)表示未编码的磁盘，例如所有磁化点均存“0”。

图(b)表示写操作：高功率激光束照射加热点 (记录点)，磁头线圈中外加电流后产生的磁场使其对应的记录点产生相反的磁性微粒，从而写入“1”。

图(c)表示读操作：低功率的激光束反射掉相反极性的磁性粒子且使它的极性变化。如果这些粒子没有被反射掉，则反射激光束的极性是不变化的。

图(d)表示擦除操作：高功率激光束照射记录点，外加磁场改变方向，使磁性粒子恢复到原始极性。

MO 盘介质材料发生的物理特性改变是可逆变化，因此信息是可重写的。

6、显示设备

(1) 显示设备的分类与有关概念：

以可见光的形式传递和处理信息的设备叫**显示设备**

分类：

器件：阴极射线管 CRT、液晶显示器 LCD、等离子体……

——**CRT 中又可以分类：**

扫描方式：光栅扫描和随机扫描

分辨率：高分辨率和低分辨率

显示颜色：单色和彩色

显示屏幕大小：14、15、17、19 等

显示内容：字符/图形显示器、图象显示器……

有关概念：

——**分辨率：**显示器所能显示的像素个数。像素越密，分辨率越高，图象越清晰。它取决于显像管荧光粉的粒度，荧光屏的尺寸以及 CRT 电子束的聚焦能力。

——**灰度级：**像素点的亮暗差别（黑白），在彩色显示器中表现为颜色的不同（彩色）。灰度级越多，图象层次越清楚越逼真。它取决于每个像素对应的刷新存储器的位数以及 CRT 本身的性能。

——**刷新：**电子束打在荧光粉上引起的发光只能维持几十毫秒的时间。因此必须让电子束反复不断地扫描整个屏幕，该过程称为刷新。刷新频率越高，显示越没有闪烁。50Hz（至少）

——**刷新存储器（视频存储器、显存）：**为刷新提供信号的存储器。容量取决于分辨率和灰度级。如 1024×768 ，32 位真彩色，需要 $1024 \times 768 \times 32/8B = 3MB$ ，其存取周期必须满足刷新频率的要求。设上例中要求刷新频率为 75Hz，则刷新存储器的总带宽为 $75 \times 3MBPS = 225MBPS$

——**随机扫描：**电子束在需要显示字符和图形的地方扫描。速度快，图象清晰。驱动系统复杂，价格昂贵。

——**光栅扫描：**电子束扫描整个屏幕（从上到下，从左到右）

——**图形：**没有亮暗层次变换的线条图，一般用计算机表示和生成

——**图像：**最初就具有亮暗层次的图，多来自客观世界

（2）字符/图形显示器

① 字符显示

显示系统由显示卡和和显示器组成；

显示字符的方法以点阵为基础；

字符点阵存放于字符发生器（ROM）中；

字符窗口：每个字符在屏幕上所占的点数，包括字符点阵和字符间隔

屏幕上所有字符窗口的 ASCII 均存放于 VRAM 中字符发生器的高位地址来自于 ASCII 码，低位地址来自于光栅地址计数器的输出 RA3~RA0

VRAM 的地址由水平地址计数器和垂直地址计数器决定

VRAM 输出的 ASCII 码作为 ROM 的高位地址，低位地址来自于光栅地址计数器

② 图形显示

——**随机图形显示器**

工作原理：将所显示图形的一组坐标点和绘图命令组成显示文件存放在缓冲存储器，缓存中的显示文件送矢量(线段)产生器，产生相应的模拟电压，直接控制电子束在屏幕上的移动。

优点：分辨率高(可达 4096×4096 个像素)，显示的曲线平滑。

缺点：当显示复杂图形时，会有闪烁感。

——**光栅图形显示器**

工作原理:相邻像素串接法

刷新存储器与分辨率及灰度的关系

VRAM 中存放一帧图形的形状信息，它的地址和屏幕上的地址一一对应。

$VRAM = \text{分辨率} \times \text{灰度级}$

DDA(Digital Differential Analysis)数据插补, 将显示文件变成像素信息

优点: 通用性强, 灰度层次多, 色调丰富, 显示复杂图形时无闪烁现象; 所产生的图形有阴影效应、隐藏面消除、涂色等功能。目前流行的显示器。

(3) 图像显示设备

图形是用计算机表示和生成的图, 称为**主观图像**;

图像所处理的对象多半来自客观世界, 称为**客观图像**;

光栅扫描方式, 分辨率在 256×256 或 512×512 像素; 灰度级在 64---256 级;

图像显示器有两种类型:

——一种是简单图像显示器, 仅仅显示送来的数字图像, 显示器不作处理;

——图形处理子系统: 专用计算机, 图形工作站

(4) VESA 显示模式

①MDA,CGA,EGA,VGA,VESA,SVGA,TVGA

MDA: PC 机最早使用的显示标准。单色字符显示适配器, 9×14 点阵的字符窗口, 满屏显示 80 列 \times 25 行字符, 分辨率为 720×350 像素;

VGA: 兼容字符和图形两种显示方式。字符窗口为 9×16 点阵, 图形方式下分辨率为 640×480 , 16 种颜色;

Super-VGA: 兼容 VGA 显示方式, 支持 1280×1024 像素光栅, 刷新频率达 75MHz

AVGA: super-VGA, 有了图形加速器硬件

②显示适配器:

刷新存储器、显示控制器、ROM BIOS 三部分组成:

——刷新存储器: 存储容量取决于设定的显示工作方式。

——ROM BIOS: 用于 DOS 操作系统

——**显示控制器: 适配器的核心。**

给显示器提供 **GRB** 三色信号及同步信号

CPU 将主存已经修改好的内容在扫描回程的消隐期送到**刷新存储器**。

先进的显示适配器具有**图形加速能力**: 位和块传送; 画线; 填域; 颜色扩充

7、输入设备和打印设备 (详情关注教材 P229---P232)

(1) 常用的输入设备: 图形输入 (键盘输入, 鼠标器输入)、图像输入、声音输入等

(2) 打印输出: 最基本输出形式。硬拷贝设备。分为**击打式**和**非击打式**两类。