

《计算机组成与结构》

教学大纲

安徽大学 计算机 学院

二 00 六年 4 月

《计算机组成与结构》教学大纲

I 前言

课程性质与设置目的要求

《计算机组成与结构》是计算机科学与技术、软件工程专业的一门主干课程，本课程以冯·诺依曼计算机模型作为教学起点，介绍计算机的组织结构和工作原理，剖析计算机的运算器、存储器、控制器和输入输出设备的组成、工作原理与相互关系。

设置本课程的目的是使学生对计算机的内部结构、功能部件、功能特征、性能以及交互方式全面掌握，使学生掌握计算机常用的逻辑器件、部件的原理、参数及使用方法，学懂简单、完备的单台计算机的基本组成原理，学习计算机设计中的入门性知识，掌握维护、使用计算机的基本技能。

学习本课程的要求是：通过本课程的学习和课程配套的系列实验之后，学生不应当把计算机看着一个执行程序的黑匣子，可以从计算机的组成原理和系统结构，完整说明计算机的工作过程，并能根据计算机的特征编写出更加高效的程序，为后继课程的学习奠定组成与结构的整机硬件工作的基础。

《计算机组成与结构》是计算机类各专业的一门必修的核心课程，从课程的地位来说，它在专业基础课和专业课之间起着承上启下的作用，应此一般安排在《数字逻辑》和《汇编语言》课程之后，后续课程还包括《计算机系统结构》、《微型机与接口技术》、《计算机网络》等。

本课程计划总学时 92 学时，5 学分，其中理论课 72 学时，4 学分，实验课 20 学时，1 学分。

选用教材：计算机组成与结构(第3版) 王爱英主编 清华大学出版社

教学手段：多媒体教学

教学方法：整个教学过程由授课、实验、作业和自学四个环节组成。

考核方法：闭卷考试

教学进程安排表：

周次	学时数	教学主要内容	教学环节	备注
1	2	第1章 计算机系统概论 2.1 三种基本逻辑操作及布尔代数的基本公式 2.2 逻辑函数的化简 2.3 逻辑门的实现	讲课	
1	2	2.4 计算机中常用的组合逻辑电路	讲课	
2	2	2.4 计算机中常用的组合逻辑电路 2.5 时序逻辑电路	讲课	
2	2	2.5 时序逻辑电路	讲课	
3	2	2.6 阵列逻辑电路	讲课	
3	2	3.1 数据的表示方法和转换 3.2 带符号的二进制数据在计算机中的表示方法及加减法运算	讲课	
4	2	3.2 带符号的二进制数据在计算机中的表示方法及加减法运算 3.3 二进制乘法运算	讲课	

4	2	3.3 二进制乘法运算	讲课+课堂讨论	
5	2	3.4 二进制除法运算	讲课+课堂讨论	
5	2	3.4 二进制除法运算	讲课+课堂讨论	
6	2	3.5 浮点数的运算方法 3.6 运算部件	讲课	
6	2	3.7 数据校验码		
7	2	课后习题	习题课	
7	2	4.1 主存储器处于全机中心地位 4.2 主存储器分类 4.3 主存储器的主要技术指标 4.4 主存储器的基本操作 4.5 读/写存储器	讲课	
8	2	4.5 读/写存储器	讲课	
8	2	4.6 非易失性半导体存储器 4.8 半导体存储器的组成与控制	讲课	
9	2	4.8 半导体存储器的组成与控制 4.9 多体交叉存储器	讲课	
9	2	5.1 指令系统的发展 5.2 指令格式	讲课	
10	2	5.3 数据表示 5.4 寻址方式	讲课	
10	2	5.5 指令类型 课后习题	讲课+习题课	
11	2	期中考查	期中测验	
11	2	试卷分析	课堂讨论	
12	2	6.1 计算机的硬件系统 6.2 控制器的组成	讲课	
12	2	6.2 控制器的组成 6.3 微程序控制计算机的基本工作原理	讲课	
13	2	6.3 微程序控制计算机的基本工作原理	讲课	
13	2	6.4 微程序设计技术	讲课	
14	2	6.4 微程序设计技术	讲课	
14	2	6.5 硬布线控制的计算机	讲课	
15	2	6.6 控制器的控制方式 6.7 流水线工作原理	讲课	
15	2	7.1 存储系统的层次结构 7.2 高速缓冲存储器	讲课	
16	2	7.2 高速缓冲存储器	讲课	
16	2	7.2 高速缓冲存储器	讲课	
17	2	7.3 虚拟存储器	讲课	
17	2	7.4 相联存储器	讲课	

		7.5 存储保护		
18	2	课后习题	习题课	
18	2	总复习	课堂讨论	

第1章 计算机系统概论

一、学习目的

了解计算机的语言和与硬件的层次关系；掌握冯·诺依曼计算机模型的思想、冯氏计算机的硬件组成和基本功能；了解计算机系统的层次结构，从软件、硬件两方面描述；了解电子计算机的发展简史；了解计算机的应用。

第一章计划 1 学时。

二、课程内容

1.1 计算机的语言

自然语言：人类相互交流信息所用的语言。

高级语言：一种和自然语言接近并能为计算机接受的语言。

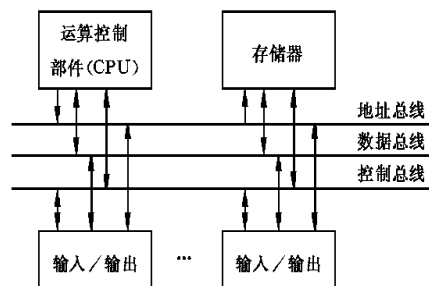
机器语言：机器能直接执行的语言（二进制，可直接进入内存，由CPU执行。）

汇编语言：符号式程序设计语言。（操作码+地址码）

1.2 计算机的硬件

组成计算机的基本部件有中央处理器 CPU（运算器和控制器）、存储器和输入、输出设备。

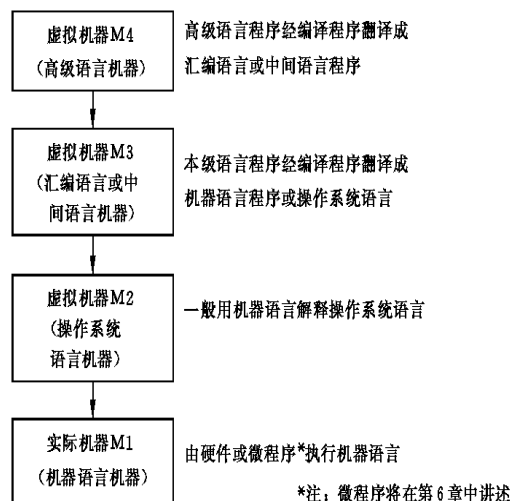
在计算机中，各部件间来往的信号可分成三种类型，即地址、数据和控制信号。通常这些信号是通过总线传送的，如图所示。



1.3 计算机系统的层次结构（从语言功能层次划分）

虚拟机：通过配置软件扩展功能后形成的与实际机无关的机器。它将提供给用户的功能抽象出来，脱离了物理机。

计算机系统的多级层次结构



1.4 电子计算机的发展简史

1.冯.诺依曼机的基本特点:

- 1) 计算机由运算器、控制器、存储器、输入设备和输出设备五部分组成。
- 2) 采用存储程序的方式，程序和数据放在同一存储器中，由指令组成的程序可以修改。
- 3) 数据以二进制码表示
- 4) 指令由操作码和地址码组成。
- 5) 指令在存储器中按执行顺序存放，由指令计数器指明要执行的指令所在的单元地址，一般按顺序递增。
- 6) 机器以运算器为中心，数据传送都经过运算器。

2.电子计算机的发展简史

3.计算机的六大分类

1.5 计算机的应用领域

三、重点、难点提示和教学手段

冯.诺依曼计算机模型的思想、冯氏计算机的硬件组成和基本功能。

四、思考与练习

冯.诺依曼结构的特点是什么?

第2章 计算机的逻辑部件

一、学习目的

1. 快速复习三种基本逻辑操作及布尔代数的基本公式、逻辑函数的化简和逻辑门的实现。
2. 掌握计算机中常用的组合逻辑电路，尤其是算术逻辑单元的组成、工作原理和先行进位的方法。
3. 了解时序逻辑电路。
4. 基本掌握阵列逻辑电路的组成、工作原理和解决问题的方案。

第二章计划 10 学时。

二、课程内容

2.1. 三种基本逻辑操作及布尔代数的基本公式

2.2. 逻辑函数的化简

1. 代数化简法
2. 卡诺图化简法

2.3. 逻辑门的实现

2.4. 计算机中常用的组合逻辑电路

逻辑电路的输出状态仅和当时的输入状态有关，而与过去的输入状态无关，称这种逻辑电路为组合逻辑电路。常见的组合电路有加法器、算术逻辑单元、译码器、数据选择器等。

1.加法器

1) 半加器

不考虑进位输入时，两数码 X_n , Y_n 相加称为半加。

半加和 H_n 的表达式为: $H_n = X_n \cdot \bar{Y}_n + \bar{X}_n \cdot Y_n = X_n \oplus Y_n$

2) 全加器

X_n , Y_n 及进位输入 C_{n-1} 相加称全加。

全加和 F_n 和进位输出 C_n 的表达式为:

$$F_n = X_n \bar{Y}_n \bar{C}_{n-1} + \bar{X}_n Y_n \bar{C}_{n-1} + \bar{X}_n \bar{Y}_n C_{n-1} + X_n Y_n C_{n-1} \quad (2.13)$$

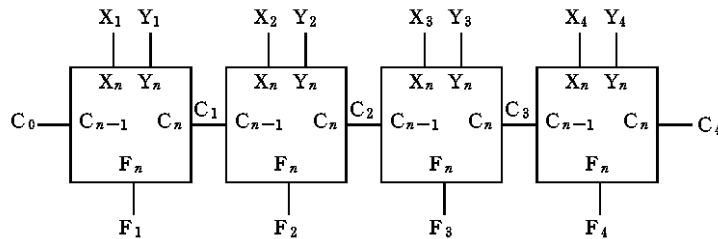
$$C_n = X_n Y_n \bar{C}_{n-1} + X_n \bar{Y}_n C_{n-1} + \bar{X}_n Y_n C_{n-1} + X_n Y_n C_{n-1} \quad (2.14)$$

全加器还可由两个半加器来形成。 F_n 是 A_n 、 B_n 相加再和 C_{n-1} 相加的结果，其表达式为：

$$F_n = X_n \oplus Y_n \oplus C_{n-1}$$

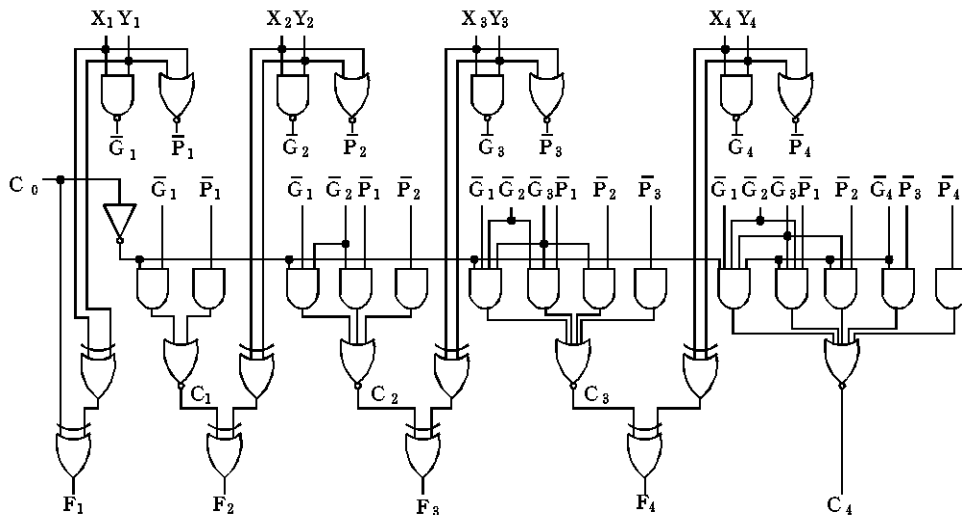
3) 串行进位加法器

将 n 个全加器相连可得 n 位加法器，其加法时间较长。因为其位间进位是串行传送的，本位全加和 F_i 必须等低位进位 C_{i-1} 来到后才能进行，加法时间与位数有关。



4) 超前进位加法器

为了提高加法器工作速度，采用“超前进位产生电路”来同时形成各位进位，从而实现快速加法。我们称这种加法器为超前进位加法器。



2. 算术逻辑单元

算术逻辑单元简称ALU,是一种功能较强的组合逻辑电路。它能进行多种算术运算和逻辑运算。ALU的基本逻辑结构是超前进位加法器，它是通过改变加法器的Gi和Pi来获得多种运算能力的。

如果把 16 位 ALU 中的每四位作为一组，用类似四位超前进位加法器(图 2.8)“位间快速进位”的形成方法来实现 16 位 ALU(由四片 ALU 组成)中的“组间快速进位”，那么就能得到 16 位快速 ALU。

3. 译码器

译码器有 n 个输入变量， 2^n 个(或少于 2^n 个)输出，每个输出对应于 n 个输入变量的一个最小项。当输入为某一组合时，对应的仅有一个输出为“0”(或为“1”)，其余输出均为“1”(或为“0”)。译码器的用途是把输入代码译成相应的控制电位，以实现代码所要求的操作。

4. 数据选择器

数据选择器又称多路开关，是以“与或”门或“与或非”门为主的电路。它能在选择信号的作用下，从多个输入通道中选择某一个通道的数据作为输出。

2. 5. 时序逻辑电路

时序逻辑电路不但与当前的输入状态有关，而且还与电路以前的输入状态有关。时序电路内必须有存储信息的记忆元件——触发器。触发器是构成时序电路的基础。

1. 触发器

1) 电位触发方式触发器 (D锁存器)

当触发器的同步控制信号 E 为约定“1”或“0”电平时, 触发器接收输入数据, 此时输入数据 D 的任何变化都会在输出 Q 端得到反映; 当 E 为非约定电平时, 触发器状态保持不变。鉴于它接收信息的条件是 E 出现约定的逻辑电平, 故称它为电位触发方式触发器, 简称电位触发器。

2) 边沿触发方式触发器 (D触发器)

触发器接收的是时钟脉冲 CP 的某一约定跳变(正跳变或负跳变)来到时的输入数据。在 CP=1 及 CP=0 期间以及 CP 非约定跳变到来时, 触发器不接收数据。

3) 主-从触发方式触发器 (J-K触发器)

主-从触发器基本上是由两个电位触发器级联而成的, 接收输入数据的是主触发器, 接收主触发器输出的是从触发器, 主、从触发器的同步控制信号是互补的(CP 和(CP)补)。

主从触发器由于有计数功能, 常用于组成计数器。

2. 寄存器和移位寄存器

寄存器是计算机的一个重要部件, 用于暂存数据、指令等。它由触发器和一些控制门组成。在寄存器中, 常用的是正边沿触发 D 触发器和锁存器。

在计算机中常要求寄存器有移位功能。如在进行乘法时, 要求将部分积右移; 在将并行传送的数转换成串行数时 also 需移位。有移位功能的寄存器称为移位寄存器。

双向四位移位寄存器。它有左移、右移、并行输入及保持功能, 采用主-从 R-S 触发器作寄存元件。

3. 计数器

计数器是计算机、数字仪表中常用的一种电路。

计数器按时钟作用方式来分, 有同步计数器和异步计数器两大类。

异步计数器中, 高位触发器的时钟信号是由低一位触发器的输出来提供的, 结构简单。

同步计数器中, 各触发器的时钟信号是由同一脉冲来提供的, 因此, 各触发器是同时翻转的, 它的工作频率比异步计数器高, 但结构较复杂。

计数器按计数顺序来分, 有二进制、十进制两大类。在计算机中较少使用异步计数器, 这里着重介绍有并行输入数据功能的正向同步十进制计数器。

同步计数器是采用快速进位方式来计数的, 触发器及实现快速进位的逻辑电路是它的核心。各触发器 J, K 表达式为

$$J_A = K_A = 1$$

$$J_B = K_B = Q_A \cdot \overline{Q_D}$$

$$J_C = K_C = Q_A \cdot Q_B$$

$$J_D = K_D = Q_A \cdot Q_B \cdot Q_C + Q_A \cdot Q_D$$

2.6 阵列逻辑电路

阵列逻辑电路近年来得到了迅速的发展。“阵列”是指逻辑元件在硅芯片上以阵列形式排列, 这种电路具有设计方便、芯片面积小、产品成品率高、用户自编程、减少系统的硬件规模等优点

1. 只读存储器(read only memory, 简称 ROM)

ROM 是一类重要的阵列逻辑电路, 主要由全译码的地址译码器和存储单元体组成, 前者是一种“与”阵列, 后者则是“或”阵列, 它们都以阵列形式排列。存储体中写入的信息是由用户事先决定的, 因此是“用户可编程”的, 而地址译码器则是“用户不可编程”的。

2. 可程序逻辑阵列(programmable logic array, 简称 PLA)

是 ROM 的变种, 也可以说是一种新型的 ROM。它和 ROM 不同之处是 PLA 的与阵列、

或阵列都是用户可编程的。PLA 在组成控制器、存储固定函数以及实现随机逻辑中有广泛的应用。

3. 可程序阵列逻辑(programmable array logic,简称 PAL)

也是 ROM 的变种,它和 ROM 不同处是 PAL 的与阵列是用户可编程的,而或阵列是用户不可编程的。

4. 通用阵列逻辑(general array logic,简称 GAL)

是一种比 PAL 功能更强的阵列逻辑电路。在它的输出有一个逻辑宏单元,通过对它的编程,可以获得多种输出形式,从而使功能大大增强。

三、重点、难点提示和教学手段

重点: 1. 算术逻辑单元的组成、工作原理和先行进位的方法。

2. 阵列逻辑电路的组成、工作原理和解决问题的方案。

难点: 1. 算术逻辑单元的工作组成、控制方式、先行进位的原理。

2. 利用阵列逻辑电路解决问题的方案。

四、思考与练习

1. 74181是采用(1)进位方式的4 位并行加法器,74182是实现(2)进位的进位逻辑。若某计算机系统系统字长为64位,每4位构成一个小组,每4个小组构成一个大组,为实现小组内并行,大组内并行,大组间串行进位方式,共需要(3)片74181和(4)片74182.
2. 简化图2. 23,画出仅有清零功能的十进制计数器。

第3章 运算方法和运算部件

一、学习目的

1. 快速复习数值的表示方法和转换、带符号的二进制数据在计算机中的表示方法及加减法运算,使学生掌握加减法运算的溢出判断方式和定点数、浮点数的表示格式、表示范围等知识。2. 掌握二进制乘法、除法运算的控制流程和控制逻辑框图,了解快速乘法、除法的原理和实现方法。3. 掌握浮点数加减运算的基本步骤,了解乘除运算的基本方法。

4. 掌握定点运算部件的组成,了解浮点运算部件的组成。

5. 了解数据校验的原理,初步掌握海明校验、CRC 校验的原理和基本方法。

第三章计划 16 学时。

二、课程内容

3.1数值的表示方法和转换

1.数值型数据的表示和转换

1) 数制

2) 不同数制间的数据转换

3) 数据符号的表示

2.十进制数的编码与运算

1) 十进制数的编码与运算

2) 数字串在计算机内的表示与存储

3.2带符号的二进制数据在计算机中的表示方法及加减法运算

1.原码、补码、反码及其加减法运算

1) 原码表示法

①原码的定义

②原码的优缺点

2) 补码表示法

①补码的定义

②补码的优缺点

③求补码的简便方法:

正数的补码同原码, 负数补码—符号位:1,数值部分:绝对值求反末位加 1.

3) 反码表示法

①反码的定义

②反码表示的注意事项

4) 数据从补码和反码表示形式转换成原码

5) 整数的表示形式 小数点隐含位置不同

2.加减法运算的溢出处理

3.定点数和浮点数

进行算术运算时, 需指出小数点的位置, 根据小数点的位置是否固定, 在计算机中有两种数据格式: 定点数和浮点数两种表示方式.

1) 定点数

定点数是指小数点固定在某个位置上的数据, 一般有小数和整数两种表示形式。

①定点小数: 把小数点固定在数值部分的左边, 符号位的右边。记作: $X_s.X_1X_2...X_n$, 这个数是纯小数, 小数点位置是隐含的, 小数点并不需要真正地占据一个二进制数。

②定点整数

把小数点固定在数据数值部分的右边。记作: $X_sX_1X_2...X_n$, 这个数是纯整数。

2) 浮点数

浮点数是指小数点位置可浮动的数据, 通常以下式表示:

$$N=M \cdot R^E$$

其中, N为浮点数, M(mantissa)为尾数(纯小数), E(exponent)为阶码(整数), R(radix)称为“阶的基数(底)”, 而且R为一常数(与尾数的基数相同), 一般为2、8或16。在一台计算机中, 所有数据的R都是相同的, 于是不需要在每个数据中表示出来。

3) 计算机中数据的数值范围和精度

数值范围: 机器所能表示的一个数的最大值和最小值之间的范围。

数据精度: 指一个数的有效位数。

数值范围和数据精度是两个不同的概念。

3.3二进制乘法运算

1.定点数一位乘法

1) 定点原码一位乘法

规则如下:

①符号位单独处理, 同号为正, 异号为负。

②令乘数的最低位为判断位, 若为“1”, 加被乘数, 若为“0”, 不加被乘数(加0)

③累加后的部分积右移一位。

2) 定点补码一位乘法

补码乘法不能简单的套用原码乘法的算法, 因为补码的符号位是参加运算的。

①校正法

$$[XY]_{\text{补}} = X_{\text{补}}[0.Y_1Y_2.....Y_n] + [-X]_{\text{补}}Y_0$$

②比较法算法(布斯公式)

运算规则:

I. A、B 取双符号位, 符号参加运算;

II. C 取单符号位, 符号参加移位, 以决定最后是否修正;

III. C 末位设置附加位 C_{i+1} , 初值为 0, $C_{i+1}C_i$ 组成判断位, 决定运算操作;

IV. 需作 $n+1$ 次累加, n 次移位。

2.补码两位乘

1)根据BOOTH乘法推导出补码两位乘法规则如下:

- ①参加运算的数用补码表示
- ②符号位参加运算
- ③乘数最低位后面增加一位附加位 Y_{n+1} ,其初值为0
- ④根据乘数的最低三位 $Y_{n-1}Y_nY_{n+1}$ 的值决定每次应执行的操作
- ⑤移位按补码右移规则进行。

2)补码两位乘法操作 (P_{80} 表3.5)

当乘数由1位符号位和 n (奇数)位数据位组成时,求部分积的次数为 $(1+n)/2$,且最后一次的右移操作只右移1位。若数值位本身为偶数,则可采用下述方法之一

- ①可在乘数的末位补1个“0”,乘数的数据位就成为奇数,其值不变, ,求部分积的次数为 $1+(n+1)/2$,即 $n/2+1$,最后一次的右移操作只右移1位。
- ②乘数增加1位符号位,使总位数仍为偶数,此时求部分积的次数为 $n/2+1$,最后一次不再执行右移操作。

3.3.3阵列乘法器

3.4 二进制除法运算

1.定点除法运算

1)定点原码一位除法

- ①恢复余数法
- ②加减交替法

规则:当余数为正时,商上“1”,求下一位商的办法,是余数左移1位,再减去除数,当余数为负时,商上“0”,求下一位商的办法,是余数左移1位,再加上除数。此方法不用恢复余数,所以又称不恢复余数法,但若最后一次商为“0”,仍需恢复余数。

③说明

I.对定点小数除法,首先要比较除数和被除数的绝对值的大小,以检查是否出现商溢出的情况。

II.商的符号为相除二数的符号的半加和。

III.被除数的位数可以是除数的两倍,其低位的数值部分开始时放在商寄存器中。运算过程中,放被除数和商的寄存器同时移位,并将商寄存器中的最高位移到被乘数寄存器的最低位中。

IV.实现除法的逻辑电路与乘法的逻辑电路(图 3.5)极相似,但在 A 寄存器中放被除数/余数, B 寄存器中放除数, C 寄存器放商(如被除数为双倍长,在开始时 C 中放被除数的低位)。此外,移位电路应有左移 1 位的功能,以及将 $Y/[-Y]$ 补送 ALU 的电路。

V.余数的符号同被除数的符号。

2、 定点补码一位除法(加减交替法)

1) 规则如下:

- ①如果被除数与除数同号,用被除数减去除数;若两数异号,用被除数加上除数。如果所得余数与除数同号上商 1,若余数与除数异号,上商 0,该商即为结果的符号位。
- ②求商的数值部分。如果上次上商 1,将余数左移一位后减去除数;如果上次上商 0,将余数左移一位后加上除数。然后判断本次操作后的余数,如果余数与除数同号上商 1;若余数与除数异号上商 0。如此重复执行 $n-1$ 次(设数值部分有 n 位)。
- ③商的最后一位一般采用恒置1的办法,并省略了最低位+1的操作,此时最大误差为 $\pm 2^{-n}$ 。如果对商的精度要求较高,则可按规则② 再进行一次操作,以求得商的第 n 位。当除不尽时,若商为负,要在商的最低一位加1,使商从反码值转变成补码值;若商为正,最低位不需要加1。

2)商的校正原则:

①当除不尽时，若商为负，要在商的最低一位加1，使商从反码值转变成补码值；若商为正，最低位不需要加1。

②当除尽时，如果除数为正，商的最低一位不要加1，若除数为负，最低位加1。

3) 余数的校正原则：

①若商为正，当余数与被除数异号时，将余数加上除数进行校正。

②若商为负，当余数与被除数异号时，将余数减去除数进行校正。

注：除尽是指运算过程中任一步余数为0。

3.5 浮点的运算方法

1. 浮点加减运算

1) “对阶”——使两数阶码相等(对齐两数的小数点)

2) 尾数的加/减运算

3) 规格化操作

4) 舍入

5) 检查阶码是否溢出

2. 浮点数的乘除法运算

1) 乘法的步骤

①阶码相加——若阶码用移码表示，应注意要减去一个偏移量 2^n 。

②尾数相乘——与定点小数乘法算法相同。

③尾数结果规格化

④舍入

⑤判溢出

2) 除法的步骤

①尾数调整——保证 $M_x < M_y$

②阶码相减

③尾数相除

④舍入

⑤判溢出

3.6 运算部件

1. 定点运算部件

2. 浮点运算部件

通常由阶码运算部件和尾数运算部件组成，其各自的结构与定点运算部件相似。但阶码部分仅执行加减法运算。其尾数部分则执行加减乘除运算，左规时有时需要左移多位。为加速移位过程，有的机器设置了可移动多位的电路。

3.7 数据校验码

计算机系统中的数据，在读写、存取和传送的过程中可能产生错误。为减少和避免这类错误，一方面是精心设计各种电路，提高计算机硬件的可靠性；另一方面是在数据编码上找出路，即采用某种编码法，通过少量的附加电路，使之能发现某些错误，甚至能确定出错位置，进而实现自动纠错的能力。

1. 奇偶校验码

1) 奇偶校验码是一种开销最小，能发现数据代码中一位出错情况的编码，

2) 常用于存储器读写检查，或ASCII字符传送过程中的检查。

3) 编码方法：加一位校验位，使奇校验检验为奇数个1，偶校验为偶数个1。

2. 海明校验码

1) 特点：能检测出二位同时出错、亦能检测出一位出错并能自动纠错。

2) 实现原理：

若海明码的最高位号为 m ，最低位号为 1，即 $H_m H_{m-1} \dots H_2 H_1$ ，则此海明码的编码规律通常是：

①校验位与数据位之和为 m ，每个校验位 P_i 在海明码中被分在位号 2^{i-1} 的位置，其余各位为数据位，并按从低向高逐位依次排列的关系分配各数据位。

②海明码的每一位码 H_i (包括数据位和校验位本身) 由多个校验位校验，其关系是被校验的每一位位号要等于校验它的各校验位的位号之和。这样安排的目的，是希望校验的结果能正确反映出出错位的位号。

3. 循环冗余校验 (CRC) 码。

1) 特点：CRC 码可以发现并纠正信息串行读写、存储或传送过程中连续出现的一位、多位错误

2) 用途：在磁介质存储器读写和计算机之间通信方面得到广泛应用。

3) CRC 码的编码方法

①可将待编码的 k 位有效信息位组表达为多项式 $M(x)$ ：

$$M(x) = C_{k-1}x^{k-1} + C_{k-2}x^{k-2} + \dots + C_i x^i + \dots + C_1 x + C_0$$

式中 C_i 为 0 或 1。

②若将信息位组左移 r 位，则可表示为多项式 $M(x) \cdot x^r$ 。可以空出 r 位，以便拼接 r 位校验位。

③CRC 码是用多项式 $M(x) \cdot x^r$ 除以称为生成多项式 $G(x)$ (产生校验码的多项式) 所得余数作为校验位的。为了得到 r 位余数 (校验位)， $G(x)$ 必须是 $r+1$ 位。

④设所得余数表达为 $R(x)$ ，商为 $Q(x)$ 。将余数拼接在信息位组左移 r 位空出的 r 位上，就构成这个有效信息的 CRC 码。这个 CRC 码可用多项式表达为：

$$M(x) \cdot x^r + R(x) = [Q(x) \cdot G(x) + R(x)] + R(x) = [Q(x) \cdot G(x)] + [R(x) + R(x)] = Q(x) \cdot G(x)$$

因此所得 CRC 码可被 $G(x)$ 表示的数码除尽。

4) CRC 的译码与纠错

原理：将收到的循环校验码用约定的生成多项式 $G(x)$ 去除，如果码字无误则余数应为 0，如有某一位出错，则余数不为 0，不同位数出错余数不同。

纠错依据：余数与出错位的对应关系是不变的，只与码制和生成多项式有关。

5) 关于生成多项式

三、重点、难点提示和教学手段

重点：1. 加减法运算的溢出判断方式和定点数、浮点数的表示格式、表示范围等知识。

2. 二进制乘法、除法运算的控制流程和控制逻辑框图。

3. 浮点数加减运算的基本步骤。

4. 数据校验的原理。

难点：1. 二进制乘法、除法运算的控制流程和控制逻辑框图理。

2. 海明校验、CRC 校验的原理和基本方法。

四、思考与练习

1. 设机器字长 16 位，阶码 7 位，其中阶符 1 位；尾数 9 位，其中数符 1 位 (阶码底为 2)，若阶码和尾数均用补码表示，说明在尾数规格化和不规格化两种情况下，它所能表示的最大正数、非零最小正数，绝对值最大的负数，绝对值最小的负数各是哪几个数？写出机器数，并给出十进制值。若阶码用移码，尾数仍用补码，上述各值有变化吗？若有变化请列出。

2. 已知 $x = -0.1101$ ， $y = 0.0110$ ，用原码一位乘法求 $(x \cdot y)_{\text{原}} = ?$

3. 已知 $x = -0.1101$ ， $y = 0.0110$ ，用补码一位乘法求 $(x \cdot y)_{\text{补}} = ?$

4. $X = -0.10110$ ， $Y = 0.11111$ 用加减交替法原码一位除计算 X/Y 的商及余数。

5. $X = 0.10110$ ， $Y = 0.11111$ 用加减交替法补码一位除计算 X/Y 的商。

6. $X = -0.0100$ ， $Y = 0.1000$ ，用补码一位除，求 X/Y 的商。

7. $[X]$ 补=0.10011, $[Y]$ 补=1.01101,用补码两位乘运算,求乘积 $[X*Y]$ 补。
- 8 用原码两位乘方法求 $X*Y$ 。已知 $X=0.1011, Y=0.1101$ 。
- 9 设浮点数 X, Y , 阶码(补码形式)和尾数(原码形式)如下:
- X : 阶码 0001, 尾数 0.1010;
 Y : 阶码 1111, 尾数 0.1001。设基数为 2。
- (1) 求 $X+Y$ (阶码运算用补码, 尾数运算用补码)
(2) 求 $X*Y$ (阶码运算用移码, 尾数运算用原码一位乘)
(3) 求 X/Y (阶码运算用移码, 尾数运算用原码加减交替法)
- 10 如果采用偶校验, 下述两个数据的校验位的值是什么? (1) 0101010 (2) 0011011
- 11 设有 16 个信息位, 如果采用海明校验, 至少需要设置多少个校验位?应放在哪些位置上?
- 12 设有 8 位有效信息, 试为之编制海明校验线路。说明编码方法, 并分析所选方案具有怎样的检错与纠错能力。若 8 位信息为 01101101, 海明码是何值?

第4章 主存储器

一、学习目的

1. 了解主存储器处于全机中心地位、主存储器分类、主存储器的主要技术指标、主存储器的基本操作。
2. 掌握、存储器的组成、读/写过程的时序和再生产生的原因和实现方法。
3. 掌握半导体存储器的组成与控制, 了解多体交叉存储器的原理和编码方法。

第四章计划 8 学时。

二、课程内容

4.1 主存储器处于全机中心地位

- 1.正在运行的程序和数据存放于存储器中。**CPU**直接从存储器取指令或存取数据。
2. 采用DMA技术或输入输出通道技术, 在存储器和输入输出系统之间直接传输数据。
3. 多处理机系统采用共享存储器来存取和交换数据

4.2 主存储器分类

能用来作为存储器的器件和介质,除了其基本存储单元有两个稳定的物理状态来存储二进制信息以外,还必须满足一些技术上的要求。另外价格也是一个很重要的因素。

主存储器的类型:

- (1)随机存储器(random access memory, 简称 RAM)
- (2)只读存储器(read\only memory,简称 ROM)
- (3)可程序的只读存储器(programmable ROM,简称 PROM)
- (4)可擦除可程序只读存储器(erasable PROM,简称 EPROM)
- (5)可用电擦除的可编程只读存储器(electrically EPROM,简称 E²PROM)

4.3 主存储器的主要技术指标

主存储器的主要性能指标:主存容量、存储器存取时间和存储周期时间。

1.存储容量:

2.存取时间(存储器访问时间)(或读/写时间)

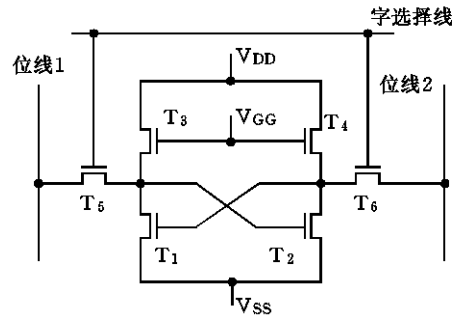
4.4 主存储器的基本操作

为了从存储器中取一个信息字, CPU 必须指定存储器字地址, 并进行“读”操作。CPU 需要把信息字的地址送到 AR, 经地址总线送往主存储器。同时, CPU 应用控制线(read)发一个“读”请求。此后, CPU 等待从主存储器发来的回答信号, 通知 CPU“读”操作完成。主存储器通过 ready 线做出回答, 若 ready 信号为“1”, 说明存储字的内容已经读出, 并放在数据总线上, 送入 DR。这时, “取”数操作完成。

为了“存”一个字到主存，CPU 先将信息字在主存中的地址经 AR 送地址总线，并将信息字送 DR。同时，发出“写”命令。此后，CPU 等待写操作完成信号。主存储器从数据总线接收到信息字并按地址总线指定的地址存储，然后经 ready 控制线发回存储器操作完成信号。这时，“存”数操作完成。

4.5 读/写存储器(随机存储器(RAM))

1. 静态存储器(SRAM)



1) 读操作

2) 写操作

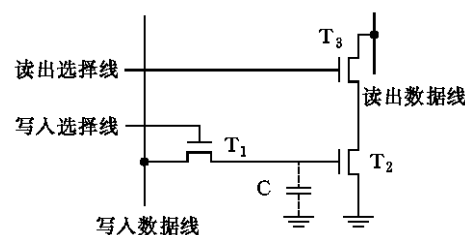
3) 开关特性

① 读周期的参数

② 写周期的参数

2. 动态存储器(DRAM)

1) 存储单元和存储器原理



2) 再生

DRAM是通过把电荷充积到MOS管的栅极电容或专门的MOS电容中去来实现信息存储的。但是由于电容漏电阻的存在，随着时间的增加，其电荷会逐渐漏掉，从而使存储的信息丢失。为了保证存储信息不遭破坏，必须在电荷漏掉以前就进行充电，以恢复原来的电荷。把这一充电过程称为再生，或称为刷新。对于DRAM，再生一般应在小于或等于2ms的时间内进行一次。SRAM则不同，由于SRAM是以双稳态电路为存储单元的，因此它不需要再生。

3. 时序图

工作方式：

1) 读工作方式

2) 写工作方式

3) 读-改写工作方式

4) 页面工作方式

5) 再生工作方式

4. DRAM与SRAM的比较

DRAM的优点：

1) 每片存储容量较大；引脚数少。

2) 价格比较便宜。

3) 所需功率大约只有SRAM的1 / 6。

DRAM作为计算机主存储器的主要元件得到了广泛的应用。

DRAM的缺点：

- 1)速度比SRAM要低。
- 2)DRAM需要再生，这不仅浪费了宝贵的时间，还需要有配套的再生电路，它也要用去一部分功率。

SRAM一般用作容量不大的高速存储器。

4.6 非易失性半导体存储器

易失性存储器(DRAM和SRAM):当掉电时,所存储的内容立即消失。

非易失性半导体存储器:即使停电,所存储的内容也不会丢失。

1. 只读存储器(ROM)
2. 可编程的只读存储器(PROM)
3. 可擦可编程的只读存储器(EPROM)
4. 可电擦可编程序只读存储器(E²PROM)
5. 快擦除读写存储器(Flash Memory)

4.8 多体交叉存储器

三、重点、难点提示和教学手段

重点: 1. 静、动态存储元的读/写原理,再生产生的原因和实现方法。

2. 存储器的字扩展、位扩展方式,存储器组成与控制。

难点: 1. 静、动态存储元的读/写原理。

2. 存储器组成与控制。

四、思考与练习

1. 有一个512K×16的存储器,由64K×1的2164RAM芯片构成(芯片内是4个128×128结构)。

(1) 总共需要多少个RAM芯片?

(2) 采用分散刷新方式,如单元刷新间隔不超过2ms,则刷新信号的周期是多少?

(3) 如采用集中刷新方式,设读/写周期 $T=0.1\mu s$,存储器刷新一遍最少用多少时间?

2. 某机器中,已知有一个地址空间为0000H~1FFFH的ROM区域,现在再用RAM芯片(8K×4)形成一个16K×8的RAM区域,起始地址为2000H,假设RAM芯片有CS和WE信号控制

端。CPU地址总线为A15~A0,数据总线为D7~D0,控制信号为 R/\overline{W} (读/写),MREQ(当存储器进行读或写操作时,该信号指示地址总线上的地址是有效的)。要求画出逻辑图。

第5章 指令系统

一、学习目的

1. 了解指令格式、数据表示。
2. 掌握不同寻址方式(编址方式)中部件之间的动作关系,可能的时间分配。
3. 了解指令类型、指令系统的兼容性和精简指令系统计算机(RISC)、复杂指令系统计算机(CISC)的有关概念、特性等。

第五章计划6学时。

二、课程内容

5.1 指令系统的发展

5.2 指令格式

计算机的指令格式与机器的字长、存储器的容量及指令的功能都有很大的关系。

设计指令格式的要求:

- (1)使指令能给出足够的信息
- (2)其长度又尽可能地与机器的字长相匹配,以便节省存储空间,缩短取指时间,提高机器

的性能。

1. 指令格式

- 1) 四地址指令
- 2) 三地址指令
- 3) 二地址指令
- 4) 一地址指令
- 5) 零地址指令

2. 指令操作码的扩展技术

3. 指令长度与字长的关系

5.3 数据表示

1. 操作数的类型

逻辑(布尔)数、定点数(整数)、浮点数(实数)、十进制数、字符串、数组等

2. 操作数的存储方式

3. 数据对齐方式

5.4 寻址方式(编址方式)

存放操作数的部件:ALU的某个寄存器、存储器、指令

程序的指令代码:一般在存储器中

寻址方式:根据指令中给出的地址码字段寻找真实的操作数以及下一条要执行的指令地址的方式。

1. 直接寻址

操作数的地址直接在指令中给出,操作数在存储器中

2. 寄存器寻址

操作数在寄存器中,寄存器号在指令中给出。

3. 基址寻址

在计算机中设置一个专用的基址寄存器,或由指令指定一个通用寄存器为基址寄存器,操作数的地址由基址寄存器的内容和指令的地址码A相加得到,地址码A通常被称为位移量(displacement),也可用其他方法获得位移量。

注1:基址寻址主要用以解决程序在存储器中的定位(逻辑地址 物理地址)和扩大寻址空间(基址+位移量)等问题。

注2:通常基址寄存器中的值只能由系统程序设定,由特权指令执行,而不能被一般用户指令所修改,因此确保了系统的安全性。

4. 变址寻址

指令地址码部分给出的地址A和指定的变址寄存器X的内容通过加法器相加,所得的和作为地址从存储器中读出所需的操作数。这是几乎所有计算机都采用的一种寻址方式,当计算机中还有基址寄存器时,那么在计算有效地址时还要加上基址寄存器的内容。

5. 间接寻址

根据指令的地址码所取出的内容是操作数的地址或指令的地址,这种方式称为间接寻址或间址。

根据地址码取出的是寄存器地址还是存储器地址,间接寻址又可分为寄存器间接寻址和存储器间接寻址两种方式。

间接寻址有一次间址和多次间址两种情况。

对于存储器一次间址情况,需访问两次存储器才能取得数据,第一次从存储器读出操作数地址,第二次读出操作数。

6. 相对寻址

把程序计数器PC的内容(即当前执行指令的地址)与指令的地址码部分给出的位移量

(disp)之和作为操作数的地址或转移地址，称为相对寻址。

主要用于转移指令，执行本条指令后，将转移到(PC)+disp，(PC)为程序计数器的内容。

7. 立即数

所需的操作数由指令的地址码部分直接给出，就称为立即数(或直接数)寻址方式。

这种方式的特点是取指时，操作码和一个操作数同时被取出，不必再次访问存储器，提高了指令的执行速度，但不能修改，通常用于寄存器或存储单元赋初值或提供一个常数等。

5.5 指令类型

指令系统决定了计算机的基本功能，因此指令系统的设计是计算机系统设计中的一个核心问题。

1. 算术逻辑运算指令
2. 移位操作指令
3. 浮点运算指令
4. 十进制运算指令
5. 字符串处理指令
6. 数据传送指令
7. 转移类指令
8. 堆栈及堆栈操作指令
9. 输入输出(I/O)指令
10. 特权指令
11. 其他指令
 - (1) 向量指令
 - (2) 多处理机指令
 - (3) 控制指令

包括等待指令、停机指令、空操作指令、开中断、关中断、置条件码指令等。

三、重点、难点提示和教学手段

重点：不同寻址方式（编址方式）中部件之间的动作关系，可能的时间分配。

四、思考与练习

1 某指令系统指令长 16 位，每个操作数的地址码长 6 位，指令分为无操作数、单操作数和双操作数三类。若双操作数指令有 K 条，无操作数指令有 L 条，问单操作数指令最多可能有多少条？

2 基址寄存器的内容为 2000H(H 表示十六进制)，变址寄存器内容为 03A0H，指令的地址码部分是 3FH，当前正在执行的指令所在地址为 2B00H，请求出变址编址(考虑基址)和相对编址两种情况的访存有效地址(即实际地址)。

3 接上题

(1) 设变址编址用于取数指令，相对编址用于转移指令，存储器内存放的内容如下：

地址	内容
003FH	2300H
2000H	2400H
203FH	2500H
233FH	2600H
23A0H	2700H
23DFH	2800H
2B00H	063FH

请写出从存储器中所取的数据以及转移地址。

(2) 若采取直接编址，请写出从存储器取出的数据。

4 在下面有关寻址方式的叙述中, 选择正确答案填入()内。

根据操作数所在位置, 指出其寻址方式: 操作数在寄存器中, 为(A)寻址方式; 操作数地址在寄存器中称为(B)寻址方式; 操作数在指令中, 称为(C)寻址方式; 操作数地址(主存)在指令中, 为(D)寻址方式。操作数的地址, 为某一寄存器中的内容与位移量之和则可以是(E)、(F)、(G)寻址方式。

供选择的答案:

A、B、C、D、E、F、G:

① 直接② 寄存器③ 寄存器间接④ 基址⑤ 变址 ⑥ 相对⑦ 堆栈⑧ 立即数

第6章 中央处理部件CPU

一、学习目的

1. 了解计算机的硬件系统、分析计算机的加电及控制过程。
2. 掌握控制器的组成的主要部件, 并能分析各主要部件的功能和相互关系。
3. 掌握微程序控制计算机的基本工作原理, 微程序设计技术。
4. 分析硬布线控制的计算机的工作原理、时序分配方法。
5. 了解控制器的控制方式、流水线工作原理。

第六章计划 14 学时。

二、课程内容

6.1 计算机的硬件系统

1. Intel 80386微机系统
2. Intel 80386结构及外部连线

80386包括:

- 1) 指令部件: 完成取指及指令译码功能并产生控制信号;
- 2) 执行部件: 包括ALU、乘法部件、寄存器等;
- 3) 存储管理部件: 用来确定存储器地址。

6.2 控制器的组成

1. 控制器的功能

- 1) 取指令
- 2) 分析指令
- 3) 执行指令
- 4) 控制程序和数据的输入与结果输出
- 5) 对异常情况和某些请求的处理

2. 控制器的组成

- 1) 程序计数器(PC)
- 2) 指令寄存器(IR)
- 3) 指令译码器或操作码译码器
- 4) 脉冲源及启停线路
- 5) 时序控制信号形成部件

3. 指令执行过程

- 1) 组成控制器的基本电路
- 2) 指令执行过程举例
- ① 一条加法指令的执行过程:(时序图)
- ② 条件转移指令的执行过程

6.3 微程序控制计算机的基本工作原理

1. 微程序控制的基本概念

在计算机中，一条指令的功能是通过按一定次序执行一系列基本操作完成的，这些基本操作称为微操作。

微指令：在微程序控制的计算机中，将由同时发出的控制信号所执行的一组微操作称为微指令，所以微指令就是把同时发出的控制信号的有关信息汇集起来而形成的。将一条指令分成若干条微指令，按次序执行这些微指令，就可以实现指令的功能。组成微指令的微操作，又称微命令。

微程序：计算机的程序由指令序列构成，而计算机每条指令的功能均由微指令序列解释完成，这些微指令序列的集合就叫做微程序。

控制存储器：微程序是存放在存储器中的，由于该存储器主要存放控制命令(信号)与下一条执行的微指令地址(简称为下址)，所以被叫做控制存储器。一般计算机指令系统是固定的，所以实现指令系统的微程序也是固定的，于是控制存储器可以用只读存储器实现。执行一条指令实际上就是执行一段存放在控制存储器中的微程序。

2.实现微程序控制的基本原理

1) 控制信号

2) 微程序控制器

3) 时序信号及工作脉冲的形成

4) 电路配合中的常见问题

①电路延迟引起的波形畸变

②机器周期的确定

③时钟脉冲CLK和工作脉冲CP的标准性

5) 微程序控制计算机的工作过程简单的总结

①机器加电后，首先由reset信号在PC内置入开机后执行的第一条指令的地址，同时在微指令寄存器内置入一条“取指”微指令，并将其他一些有关的状态位或寄存器置于初始状态。当电压达到稳定值后，自动启动机器工作，产生节拍电位 T_1 、 T_2 和CP。

②机器开始执行程序，不断地取出指令、分析指令、执行指令。程序可以存放在固定存储器中，也可以利用一小段引导程序(在固存中)将要执行的程序和数据从外部设备调入主存。

6.4 微程序设计技术

在实际进行微程序设计时，要考虑下面三个问题：

(1)如何缩短微指令字长；

(2)如何减少微程序长度；

(3)如何提高微程序的执行速度。

1.微指令的编译法(编码译码方法)

1) 直接控制法

2) 字段直接编译法

3) 字段间接编译法

4) 常数源字段E

5) 其他

2.微程序流的控制

现行微指令：当前正在执行的微指令。

现行微地址：现行微指令所在的控制存储器单元的地址。

后继微指令：下一条要执行的微指令。

后继微地址：后继微指令所在的控制存储器单元地址。

微程序控制：当前微指令执行完毕后，产生后继微指令的微地址的方法。

产生后继微指令地址的几种方法

1)以增量方式产生后继微地址

A) 微程序入口：首条微指令，由专门硬件电路产生，控制实现取指操作；

B) 顺序执行：后继微地址=现行微地址（在微程序计数器中）+1

C) 转移地址：后继微地址=现行微地址+转移条件译码（相对转移量）

D) 由操作码产生后继地址

2) 增量与下址字段结合产生后继微地址

3) 多路转移方式

4) 微中断

3.微指令格式

1) 水平型微指令

特点:在一条微指令中定义并执行多个并行操作微命令。

在实际应用中，直接控制法、字段编译法(直接、间接编译法)经常应用在同一条水平型微指令中。从速度来看，直接控制法最快，字段编译法要经过译码，所以会增加一些延迟时间。

2) 垂直型微指令

在微指令中设置有微操作码字段，采用微操作码编译法，由微操作码规定微指令的功能，称为垂直型微指令。

特点：不强调实现微指令的并行控制功能，通常一条微指令只要求能控制实现一二种操作。这种微指令格式与指令相似:每条指令有一个操作码；每条微指令有一个微操作码。

3) 水平型微指令与垂直型微指令的比较

①并行性、效率和灵活性：水平好，垂直差。

②指令执行时间：水平短，垂直长。

③水平：微指令字长，微程序短。

垂直：微指令字短，微程序长。

④用户使用：水平难，垂直易。

4. 微程序控制存储器和动态微程序设计

1) 微程序控制存储器

存储介质类型：RAM、ROM、EPROM等。

2) 动态微程序设计

能够根据用户的要求来改变微程序的设计称动态微程序设计。

用于动态微程序设计的控制存储器称为可写控制存储器(WCS)或用户控制存储器(UCS)。

3) 控制存储器的操作

①串行方式

②并行方式

4) 毫微程序设计的基本概念

5. 微程序设计语言

6.5 硬布线控制的计算机

1. 时序与节拍

一条指令的实现可分成:取指、计算地址、取数及执行等几个步骤。在微程序控制方式中，每一步由一条微指令实现，而硬布线控制方式则由指令的操作码直接控制并产生实现上述各步骤所需的控制信号。一条指令的每一步由一个机器周期实现的方法有两种：

法1. 两位计数器的译码输出产生的四个状态来表示当前所处的机器周期。

法2. 用四位触发器来分别表示四个周期，当机器处于某一周期时，相应的触发器处于“1”状态，而其余三个触发器则处于“0”状态，四位移位寄存器即可实现此功能。

注：

- 1) 实际机器指令有几十到几百条，一般根据指令功能列出每条指令的机器周期变化规律，最后归纳出几种情况，将情况相同的指令归为一类，然后列出表达式，画出逻辑图。
- 2) 延长某个机器周期时间的方法：封锁CP（简单）；控制计数器输入（复杂）。
- 3) 计数器译码器输出可能会产生毛刺，可通过改变触发器状态的顺序加以改善（此时真值表、表达式、逻辑图要作相应的修改）。

2. 操作控制信号的产生

1) 操作码译码器

2) 操作控制信号的产生

3. 控制器的组成

1) 程序计数器和中断控制逻辑

2) 译码器

3) 硬布线逻辑的实现途径

4. 硬布线控制逻辑设计中的若干问题—设计步骤

1) 指令操作码的代码分配

主要目标：简化控制部分的电路，减少延迟时间。

主要难点：

- ①CISC不定长操作码的译码困难；
- ②为后续升级系统预留指令位置困难。

2) 确定机器周期、节拍与主频

保证大部分指令在一个周期内完成，部分指令通过如“ready”信号等方式来增加机器周期来完成。

3) 根据指令功能，确定每一条指令所需的机器周期数以及每一周期所完成的操作

4) 综合所有指令的每一个操作命令(写出逻辑表达式，并化简之)

5. 硬布线控制与微程序控制的比较

1) 实现

微程序控制通过控制存储器内的代码来实现，调试、修改方便；

硬布线控制通过组合电路实现，调试、修改复杂。

2) 性能

微程序控制速度较慢；

硬布线控制速度很快，主要用于高速和RISC机器中。

6.6 控制器的控制方式

1. 同步控制方式

2. 异步控制方式

3. 联合控制方式

4. 人工控制方式

6.7 流水线工作原理

1. 流水线基本工作原理

2. 数据相关对流水线的影响

3. 程序转移对流水线的影响

1) 遇到转移指令时，下条指令不再预取（流水）。

2) 用猜测法选定分支中的一个，继续流水。若猜测正确，流水线继续执行；若不正确，返回分支点。

3) 中断处理方法：不精确断点法，将中断发生时已入流水线的指令继续执行完后再进入中断处理子程序。精确断点法，即刻进入中断处理子程序。

三、重点、难点提示和教学手段

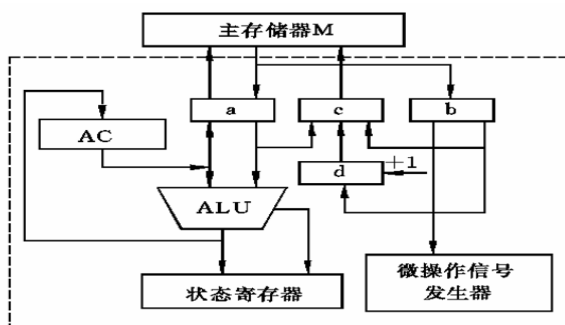
- 重点：1. 控制器的组成的主要部件，并能分析各主要部件的功能和相互关系。
 2. 微程序控制计算机的基本工作原理，微程序设计技术。
 3. 硬布线控制的计算机的工作原理、时序分配方法。
 4. 典型的 CPU 工作过程的分析。

- 难点：1. 微程序控制计算机的基本工作原理，微程序设计技术。
 2. 硬布线控制的计算机的工作原理、时序分配方法。

四、思考与练习

1. CPU 结构如图所示，其中有一个累加寄存器 AC、一个状态条件寄存器和其他四个寄存器，各部分之间的连线表示数据通路，箭头表示信息传送方向。要求：

- (1) 标明图中 a,b,c,d 四个寄存器的名称。
- (2) 简述指令从主存取到控制器的数据通路。
- (3) 简述数据在运算器和主存之间进行存/取访问的数据通路。



2. 设某计算机运算控制器逻辑图如图 6.8，控制信号意义见表 6.1，指令格式和微指令格式如下：

指令格式

操作码	rs,rd,	rs1	imm 或 disp
-----	--------	-----	------------

微指令格式

1	2	...	23	24	...	25
控制字段				下址字段		

其中 1—23 位代表的 1—23 号控制信号见表 6.1。

试写出下述三条指令的微程序编码：

- (1) JMP(无条件转移到(rs1)+disp)
- (2) Load(从(rs1)+disp 指示的内存单元取数，送 rs 保存)
- (3) Store(把 rs 内容送到(rs1)+disp 指示的内存单元)

提示：先列出各指令执行步骤和每步所需控制信号，最后再写出编码

3. 从供选择的答案中，选出正确答案填入()中

微指令分成水平型微指令和(A)微指令两类。(B)可同时执行若干个微操作，所以执行指令的速度比(C)快。

在实现微程序时，取下一条微指令和执行本条微指令一般是(D)进行的，而微指令之间是(E)执行的。

实现机器指令的微程序一般是存放在(F)中的，而用户可写的控制存储器则由(G)组成。

供选择的答案

A—C：① 微指令；② 微操作；③ 水平型微指令；④ 垂直型微指令。

D，E：① 顺序；② 重叠。

F，G：① 随机存储器(RAM)；② 只读存储器(ROM)。

4. 已知某机采用微程序控制方式，其控制存储器容量为 512×48 (位)。微指令字长为 48 位，

微程序可在整个控制存储器中实现转移，可控制微程序转移的条件共 4 个(直接控制)，微指令采用水平型格式，如图所示。

- (1) 微指令中的三个字段分别应为多少位?
- (2) 画出围绕这种微指令格式的微程序控制器逻辑框图。

微指令字段	判别测试字段	下地址字段
操作控制		顺序控制

5.从供选择的答案，选出正确答案，填入()中

某机采用两级流水线组织，第一级为取指、译码，需要 200ns 完成操作；第二级为执行周期，大部分指令能在 180ns 内完成，但有两类指令要 360ns 才能完成，在程序运行时，这类指令所占比例为 5—10%。

根据上述情况，机器周期(即一级流水线时间)应选为(A)。两条执行周期长的指令采用(B)的方法解决。

供选择的答案

A: ①180ns; ②190ns; ③200ns; ④360ns。

B: ①机器周期选为 360ns; ②用两个机器周期完成。

第7章 存储系统

一、学习目的

1. 掌握存储系统的层次结构，分析层次结构的目的和实现方式。
2. 掌握高速缓冲存储器的原理、基本结构和 cache 的存储器组织。
3. 掌握虚拟存储器信息传送单位和存储管理和虚拟存储器工作的全过程。4. 了解相联存储器和存储保护。

第七章计划 10 学时。

二、课程内容

7.1 存储系统的层次结构

1. 主存与辅存之间的关系
2. 主存和高速缓存之间的关系

7.2 高速缓冲存储器 (cache)

1. cache 存储器工作原理——程序访问的局部性

在较短时间内由程序产生的地址往往集中在存储器逻辑地址空间的很小范围内。(指令分布的连续性和循环程序及子程序的多次执行)

这种对局部范围的存储器地址频繁访问，而对此范围以外的地址则访问甚少的现象就称为程序访问的局部性。

时间局部性：如果一个信息项正在被访问，那么在近期它很可能还会被再次访问。(程序循环、堆栈)

空间局部性：在最近的将来将用到的信息很可能与现在正在使用的信息在空间地址上是临近的。(指令顺序执行、数组存放)

2. cache 存储器组织

1) 地址映像

①直接映像

这是一种多对一的映射关系，但一个主存块只能映象到 Cache 的一个特定块位置上去。

②全相联映像

全相联映像方式是最灵活但成本最高的一种方式。

③组相联映像

组相联映像方式是直接映像和全相联映像方式的一种折衷方案。组内全相联，组间直接映像。

组相联映像方式的性能与复杂性介于直接映像与全相联映像两种方式之间。当 $r=0$ 时，是直接映像方式；当 $r=c$ 时，是全相联映像方式。

主存中的一块能对应到 Cache 中一个特定组的任意一行。若组中有 n 个块，则称其为 n 路组相联。

直接映象和全相联映象是组相联的特例。

2) 替换算法

①FIFO

②LRU

③随机替换法

3. cache 存储器举例

4. 多层次 cache 存储器

1) 指令 cache 和数据 cache

2) 多层次 cache 结构

3) cache 的一致性问题

①数据 cache 存在 cache 一致性问题。(数据 cache 有写入操作，且有多种写入方案，为了提高计算机处理速度，在每次写入时，并不同时修改 L1, L2 和主存储器的内容，造成了数据的不一致。)

②指令 cache 不存在 cache 一致性问题(程序是不允许修改的)。

7.3 虚拟存储器

1. 虚拟存储器概述

1) 主存—辅存层次与 cache—主存层次的比较

2) 主存—辅存层次信息传送单位和存储管理

2. 页式虚拟存储器

3. 段页式虚拟存储器

4. 虚拟存储器工作的全过程

5. Pentium 处理机的虚拟存储器

Pentium 存储器结构有很大灵活性，根据其段表和页表是否设置可以有 4 种组合情况。

1) 无段表和无页表的存储器。非虚拟存储器其逻辑地址即为物理地址，可减少复杂性，在高性能的控制机中经常被采用。

2) 无段表和有页表的存储器。页式虚拟存储器，此时存储器的管理和保护是通过页面转换实现的。

3) 有段表和无页表的存储器。段式虚拟存储器。

4) 有段表和有页表的存储器。段页式虚拟存储器。

6. 存储管理部件(MMU)

现代计算机一般都有辅助存储器，但具有辅存的存储系统不一定是虚拟存储系统。虚拟存储系统有两大特点：

1) 允许用户用比主存空间大得多的空间来访问主存。

2) 每次访存都要进行虚实地址的转换。

为了实现逻辑地址到物理地址的转换，并在页面失效时(即被访问的页面不在主存)进入操作系统环境，设置了由硬件实现的存储管理部件 MMU，而整个虚拟存储器的管理是由 MMU 部件与操作系统共同完成的。

7.4 相联存储器

相联存储器不按地址访问存储器，而按所存数据字的全部内容或部分内容进行查找(或

检索)。

7.5 存储保护

1. 存储区域保护

- 1) 页表保护
- 2) 键方式
- 3) 环保护方式

2. 访问方式保护

对主存信息的使用可以有三种方式：读(R)、写(W)和执行(E)，“执行”指作为指令来用。所以，相应的访问方式保护就有 R, W, E 三种以及由这三种方式形成的逻辑组合。

访问方式保护可以和上述区域保护结合起来使用。例如，在界限寄存器中加一位访问方式位；采用环式保护和页式保护时将访问方式位放在页表和段表中，使得同一环内或同一段内的各页可以有不同的访问方式，从而增强了保护的灵活性。

三、重点、难点提示和教学手段

重点：1. 层次结构的目的和实现方式。

2. 高速缓冲存储器的原理、基本结构和 cache 的存储器组织。

3. 虚拟存储器信息传送单位和存储管理和虚拟存储器工作的全过程。

难点：1. cache 的存储器组织。

2. 虚拟存储器工作的全过程。

四、思考与练习

1 设某流水线计算机有一个指令和数据合一的 cache，已知 cache 的读 / 写时间为 10ns，主存的读 / 写时间为 100ns，取指的命中率为 98%，数据的命中率为 95%，在执行程序时，约有 1/5 指令需要存 / 取一个操作数，为简化起见，假设指令流水线在任何时候都不阻塞。问设置 cache 后，与无 cache 比较，计算机的运算速度可提高多少倍？

2 接上题，如果采用哈佛结构(分开的指令 cache 和数据 cache)，运算速度可提高多少倍？

3 设某计算机的 cache 采用 4 路组相联映像，已知 cache 容量为 16KB，主存容量为 2MB，每个字块有 8 个字，每个字有 32 位。请回答：

(1) 主存地址多少位(按字节编址)，各字段如何划分(各需多少位)？

(2) 设 cache 起始为空，CPU 从主存单元 0, 1, ..., 100。依次读出 101 个字(主存一次读出一个字)，并重复按此次序数读 11 次，问命中率为多少？若 cache 速度是主存的 5 倍，问采用 cache 与无 cache 比较速度提高多少倍？

4 设某计算机采用直接映像 cache，已知容量为 4096B。

(1) 若 CPU 依次从主存单元 0, 1, ..., 99 和 4096, 4097, ..., 4195 交替取指令，循环执行 10 次，问命中率为多少？

(2) 如 cache 存取时间为 10ns，主存存取时间为 100ns，cache 命中率为 95%，求平均存取时间。

5 某程序对页面要求的序列为 P3P4P2P6P4P3P7P4P3P6P3P4P8P4P6。

(1) 设主存容量为 3 个页面，求 FIFO 和 LRU 替换算法时各自的命中率(假设开始时主存为空)。

(2) 当主存容量增加到 4 个页面时，两替换算法各自的命中率又是多少？

6 下面是有关存储保护的描述。请从本题最后列出的供选择答案中选择应填入()处的正确答案：

为了保护系统软件不被破坏以及在多道程序环境下，防止一个用户破坏另一用户的程序而采取下列措施：

(1) 不准在用户程序中使用“设置系统状态”等指令。此类指令是(A)指令。

(2) 在段式管理存储器中设置(B)寄存器，防止用户访问不是分配给这个用户的存储区

域。

(3) 在环保护的主存中，把系统程序 and 用户程序按其允许访问存储区的范围分层；假如规定内层级别高，那么系统程序应在(C)，用户程序应在(D)。内层(E)访问外层的存储区。

(4) 为了保护数据及程序不被破坏，在页式管理存储器中，可在页表内设置 R(读)、W(写)及 F 位。(F)位为 1，表示该页内存放的是程序代码。

供选择的答案：

A, B: ① 特权；② 特殊；③ 上、下界；④ 系统。

C, D: ① 内层；② 外层；③ 内层或外层。

E: ① 允许；② 不允许。

F: ① M(标志)；② P(保护)；③ E(执行)；④ E(有效)。

参考书目

- 1) 计算机组成原理
白中英 主编 (第三版) 科学出版社
- 2) 计算机组成原理
蒋本珊 主编 北京航空航天大学出版社
- 3) 计算机组成原理
王诚 主编 (第二版) 清华大学出版社
- 4) 计算机组成原理
胡越民 主编 经济出版社
- 5) 计算机组成原理
唐朔飞 编著 高等教育出版社
- 6) 计算机组成原理教程 (第 3 版)
张基温 编著 清华大学出版社