

JISUANJI YINGJIAN JISHU JICHU

计算机 硬件技术 基础

施荣华 彭 军 主编



中国铁道出版社

TP360.3

S526 318

计算机硬件技术基础

施荣华 彭 军 主编

中 国 铁 道 出 版 社

2000年·北京

(京)新登字 063 号

内 容 简 介

本书是为了适应计算机基础教学的需要,为非计算机专业学生学习计算机硬件知识而编写的教材。全书共分十章,主要内容有计算机的基本知识及微型计算机系统的组成;16 位微处理器的结构、工作模式及操作功能;8086/8088 指令系统和汇编语言程序设计;输入/输出接口及常见控制器接口电路;A/D 和 D/A 转换器的原理及结构;IBM PC 微型计算机的组成和高档微型计算机系统的配置及安装。

本书深入浅出,循序渐进,选材适当,结构严谨,不仅可以作为高等学校的教材,也适用于各专业工程技术人员参考。

图书在版编目(CIP)数据

计算机硬件技术基础/施荣华,彭军主编. —北京:中国铁道出版社,2000

ISBN 7-113-03658-9

I. 计… II. 施… III. 电子计算机-硬件-基本知识-教材 IV. TP303

中国版本图书馆 CIP 数据核字(2000)第 01177 号

书 名:计算机硬件技术基础

著作责任者:施荣华 彭 军

出版·发行:中国铁道出版社(100054,北京市宣武区右安门西街 8 号)

策划编辑:殷小燕

责任编辑:殷小燕

封面设计:陈东山

印 刷:遵化市胶印厂

开 本:787×1092 1/16 印张:14.25 字数:356 千

版 本:2000 年 3 月第 1 版 2000 年 3 月第 1 次印刷

印 数:1—3000 册

书 号:ISBN 7-113-03658-9/TP·426

定 价:28.00 元

版权所有 盗印必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社发行部调换。

前 言

随着科学技术的迅速发展,计算机尤其是微型计算机应用技术不断普及,其应用范围已渗透到社会生产和生活的方方面面。在高等学校,各专业人员都十分重视微型计算机的应用,以解决本专业领域的实际问题。

一般认为,对于非计算机专业(非电类)人员来讲,不需学习计算机硬件方面的知识,而只要学习一两种高级语言,学会程序设计以及学习其它有关的专业应用软件就可以了,这话不无道理。但要注意到,随着计算机应用的不断深入,计算机应用系统的综合性增强了,我们与硬件打交道的机会增多了,这时如果对微型计算机的工作原理及硬件结构有一定的了解,我们将会如虎添翼,而且计算机教育作为一种文化教育,懂一点硬件方面的知识,必将丰富我们的知识结构,促进计算机应用水平的提高。

教育部对高等学校非计算机基础教学提出了三个层次的要求,即“计算机文化基础”、“计算机技术基础”和“计算机应用基础”。其中,在“计算机技术基础”这一层次上要求具有高级语言程序设计方面的知识,又具备计算机硬件方面的知识。“计算机程序设计”与“计算机硬件技术”各有偏重。前者可以脱离计算机硬件结构而专门讲述程序设计的技巧,后者着重讲述计算机的基本结构、内部信息流通和指令系统的基本原理,这也就是有关计算机硬件方面的知识。编写本书的目的就是为非计算机专业学生提供一本学习计算机硬件知识的教材。考虑到各种专业类型很多,要求各异,难以强求一致,因此,本书只能在最基本的原理方面作一定深度的介绍,故书名叫做《计算机硬件技术基础》。另外,考虑到非计算机专业学时的限制,篇幅适当。

全书共分十章。第一、二两章介绍计算机的基本知识及微型计算机系统的组成,篇幅不多,主要是帮助读者复习有关概念。第三章介绍 16 位微处理器的结构、工作模式及操作功能。重点分析 8086/8088CPU,同时对 80286,80386 等 CPU 作了简要论述。第四、五两章分别介绍 8086/8088 指令系统和汇编语言程序设计。第六、七两章分别介绍输入/输出接口及常见控制器接口电路。第八章介绍 A/D 和 D/A 转换器的原理及结构。第九章介绍 IBM PC 微型计算机的组成。第十章介绍 Pentium 等高档微型计算机系统的配置和安装。

本书深入浅出,循序渐进,选材适当,结构严谨,不仅可以作为高等学校的教材,也适用于各专业工程技术人员参考。

本书由施荣华、彭军主编。第一、三、六、七各章及附录由施荣华编写,第四、五、八、九各章由彭军编写,第二章由刘显雄编写,第十章由刘光瑜编写。参加全书编写及文字整理工作的还有刘卫国、王国才、盘善海、程诗杰、周跃飞、戴忠、曹尧谦等。

由于作者水平有限,错误和不妥之处,希望批评指正。

作者

1999 年 11 月

第一章 计算机基础知识

第一节 计算机系统组成

一、计算机系统组成框图

计算机是依靠硬件和软件的协同工作来执行任一给定的任务。一个完整的计算机系统的整体构成如图 1-1 所示。

二、计算机硬件系统构成及其功能

计算机硬件的基本功能是接受计算机程序的控制来实现数据输入、运算、数据输出等一系列根本性的操作。实现这些功能所要求的基本硬件配置包括如下几个方面：

输入设备：负责把用户的信息输入到计算机中。

输出设备：负责从计算机中取出信息供用户查看。

存储器：负责存储程序和数据，并根据命令提供这些程序和数据。

运算器：负责数据的算术运算和逻辑运算。

控制器：负责对程序规定的控制信息进行分析，控制并协调输入、输出操作或内存访问。上述五大部件的关系如图 1-2 所示。

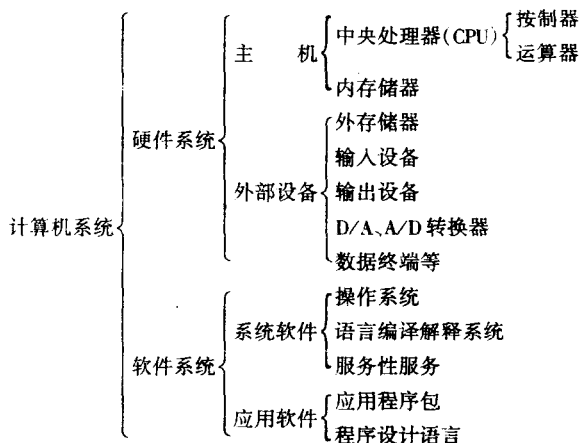


图 1-1 计算机系统组成框图

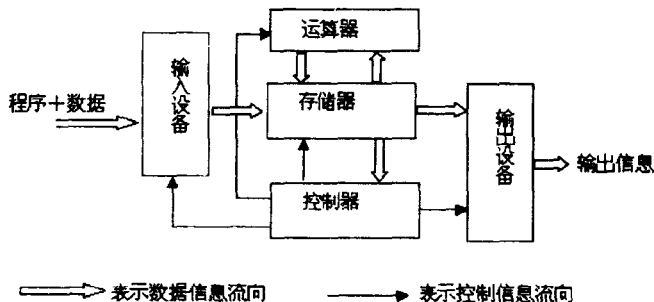


图 1-2 计算机硬件系统构成

在计算机中，基本上有两股信息在流动：一股是数据信息，即各种原始数据、中间结果、程序等，这些都由输入设备送到存储器中，在运算过程中，数据从存储器读入运算器进行运算，运

算的中间结果要么存入存储器,要么经输出设备输出;另一股是控制信息,由全机的指挥中心—控制器,根据程序的规定走向,发出控制信息并协调各部分的工作。

存储器通常又分为内存和外存两部分。内存容量小,但存储速度快。内存一般采用半导体存储器。外存储器容量大,但速度慢。微型机的外存储器常用的有磁盘和磁带,磁盘又可分为硬盘和软盘。

输入设备是计算机执行部件接受外界信息的中间媒介,最基本的输入设备便是键盘。还有其它各种所需的输入设备,如:鼠标器、光笔、数字化仪、图像扫描等等。

输出设备的作用是将机内信息传递到外部媒介,转化为某种为人所认识的表示形式。最基本的输出设备有字符显示器、图形显示器、打印机、绘图仪等。

三、计算机软件系统

(一)软件的概念

软件是相对于硬件而言的。它包括机器运行所需的各种程序及其有关资料,脱离软件的计算机硬件是不能做任何有意义工作的,它只是软件程序赖以运行的物质前提。因此,一台性能优良的计算机硬件系统能否发挥其应有的功能,取决于为之所设制的系统软件是否完善,应用软件是否丰富。由此可见,在使用、开发计算机时,不但要了解机器硬件系统的构成,还必须熟悉与之相应的各种软件。

(二)软件的分类

计算机软件一般可分为两种类别:即系统软件与应用软件。

系统软件是管理、监控和维护计算机资源的软件。它主要包括如下几方面:

- ①操作系统;
- ②各种程序设计语言及其解释程序和编译程序;
- ③机器的监控管理程序、调试程序、故障检查和诊断程序。

操作系统、程序设计语言以及服务程序,一般是由计算机厂家作为系统的一部分供给的。

应用软件是指用户利用计算机及其提供的系统软件为解决各种实际问题而编写的计算机程序。

(三)程序设计语言

编写计算机程序所用的语言称为程序设计语言。它是人与计算机之间交换信息的工具,是软件系统的重要组成部分。程序设计语言一般分为机器语言、汇编语言和高级语言三类。

(1)机器语言

机器语言是计算机硬件系统所识别的,不需翻译直接供机器使用的语言,也称为手编语言。通常随计算机型号不同而不同。机器语言中的每一条语句(即机器指令)实际上是一条二进制形式的指令代码,由操作码和地址码组成。机器语言编写难度大,调试修改繁琐,但执行速度快。

(2)汇编语言

汇编语言是一种面向机器的程序设计语言。在汇编语言中,用助记符代替操作码,用地址符号代替地址码,正是这种替代使得机器语言变得“符号化”,所以也称汇编语言为符号语言。

但使用这种语言编写的程序机器不能直接识别,要由一种起翻译作用的程序将其翻译成机器语言程序,机器方可执行。这一过程称之为“汇编”。

汇编语言程序比机器语言程序易读、易检查和修改,同时也保持了机器语言编程质量高、

执行速度快、占存储空间小的优点。但在编制复杂程度较高的程序时,汇编语言还存在着明显的局限性。尤其是这种语言程序依赖于具体的机型,故不具备通用性和可移植性。

(3)高级语言

高级语言是 50 年代中末期发展起来的面向问题的程序设计语言。高级语言的指令(或语句)一般都采用自然语言,并且使用与自然语言语法相近的自封闭语法体系,这使得程序更易阅读和理解。另一方面,高级语言的指令(或语句)是面向问题而不是面向机器的,这使得对问题及其求解的表述比汇编语言容易得多,并大大的简化了程序的编制和调试,使编程效率得以大幅度的提高。

高级语言的另一个显著特点是独立于具体的机器体系,因此,与汇编语言程序比较,通用性和可移植性大为提高。

目前世界上已有数百种高级语言,用的最多的是 FORTRAN、PASCAL、BASIC、C、PROLOG、LISP 语言等数十种。

(4)语言处理程序

除机器语言程序可以直接为机器识别之外,无论是汇编语言程序还是高级语言程序,要让机器识别,都必须经过“翻译”。所谓“翻译”是由一种特殊的程序把源程序转换成机器码,这种特殊的程序就是语言处理程序。语言处理程序可分为汇编程序、编译程序和解释程序。其中:汇编程序把汇编语言源程序“翻译”成机器语言程序,该过程叫“汇编”;编译程序把高级语言源程序“翻译”成目标程序,该过程叫“编译”;解释程序是逐条“翻译”执行高级语言程序的语句。编译程序得到的目标代码经连接后形成的可执行程序,执行速度比解释执行源程序要快,但是人机会话功能差,调试修改较复杂。

四、文件系统的基本概念

(一)文件的概念

文件是若干个逻辑记录构成的信息集合。每个文件都规定一个文件名,由此来进行读写操作。记录是作为逻辑单位顺序排列的一组相关数据项(又称字段)的集合,是构成文件的基本单位。数据项可为字符、数字或汉字组成的字符序列。记录有固定长、可变长和不可变长三种记录格式。对记录可以进行读出、更新、插入和删除四种操作。

文件系统是负责存取和管理文件的公共信息管理机构,具有对文件按名存取、采取保护及保密措施、实现信息共享、节省空间和时间开销等优点。

(二)文件的分类

计算机系统文件可按不同的方式进行分类。如按用途可分为系统文件、库文件或用户文件;按保护级别可分为只读文件、可读写文件和隐含文件;按信息流向可分为输入文件、输出文件和输入输出文件;按文件存放的时间可分为暂时文件、永久文件和档案文件;按存储设备的类型可分为磁带文件、磁盘文件;按记录的类型又可分为定长文件和变长文件等。

(三)文件的结构方式

文件的结构方式决定了记录的存储形式。常用的文件结构有顺序文件、随机文件、索引文件三种。

顺序文件是指将文件的记录按建立的时间先后顺序依次存放在连续的存储介质中。所产生的文件记录的逻辑次序与物理顺序是一致的。

随机文件是指将文件记录由程序指定的某一地址直接存取。文件记录的存取是在存储体

上绝对定位的,不必参照先前的存取操作。

索引文件是指将文件记录的逻辑号(记录号)与物理存储地址之间通过一个索引表联系在一起。记录在文件中的排列次序可以和物理位置顺序不一致,用户在使用时只要用索引表提供的关键数据项就可以进行检索。

(四)文件的使用

用户在使用文件时至少要知道文件名、文件存储介质和调用文件命令。

文件名是一个字符序列,一般由设备名、主文件名、扩展名组成。在微机上使用的文件存储设备主要是磁盘。文件的调用命令因机器型号和系统软件的不同在格式上存在差异,但是一般文件系统提供了建立文件(Create)、打开文件(Open)、读文件(Read)、写文件(Write)、关闭文件(Close)、删除文件>Delete)等基本操作命令。

第二节 计算机中的数制与编码

在计算机内部,信息的存储和处理都采用二进制,但二进制的位数太多,书写冗长,为了表达的方便,又引进了八进制和十六进制。而人们日常生活中又常常使用十进制。下面先介绍计算机中常用的几种进位数制及其转换,然后介绍字母、数字、符号的二进制编码方法。

一、进位计数制

在日常生活中,最常用的是十进制数,有时也会用到其它的进位制,例如,我国旧制中,16两为一市斤,英制16盎斯为一磅,采用的是十六进制;钟表计时中,60秒钟为一分钟,60分钟为一小时,属于六十进制。通过对各种进位计数制的分析,可以发现它们有以下两个基本特点。

(1)逢基数进一

所谓基数就是各种进位计数制中表示一位数所需的记数符号的数目。例如,十进制的基数为十,表示一位数需0~9共十个符号,进位规则是逢十进一;二进制基数是二,表示一位数需要0、1两个符号,进位规则是逢二进一;十六进制的基数为十六,需要0~9、10~15共十六个符号,进位规则是逢十六进一。十六进制中的计数10~15,一般用字母A~F来表示。

(2)每一个数位上的记数符号所表示的值是该符号和该数位权的乘积

在进位数制中,每一数位都有一个确定的位值称为该位的权。以666.66这个十进制数为例,个位的6表示其本身的数值;而十位的6表示其本身数值的十倍,即 6×10 ;百位数6表示其本身数值的100倍,即 6×10^2 ;小数点右边第一位小位的6表示0.6,即 6×10^{-1} ;第二位小数的6表示0.06,即 6×10^{-2} 。因此,十进制数各位的权为 10^i , i 沿着小数点左、右两边分别取0、1、2、...和-1、-2、-3...。十进制数666.66可以用多项式展开写成:

$$666.66 = 6 \times 10^2 + 6 \times 10^1 + 6 \times 10^0 + 6 \times 10^{-1} + 6 \times 10^{-2}$$

同理,对于J进制,各数位的权为 J^i , i 的取值同上。设J进制数 $N = (K_{n-1}K_{n-2} \dots K_1K_0K_{-1}K_{-2} \dots K_{-m})_J$,则可将它表示成按权展开的多项式之和:

$$\begin{aligned} N &= K_{n-1}J^{n-1} + \dots + K_0J^0 + K_{-1}J^{-1} + \dots + K_{-m}J^{-m} \\ &= \sum_{i=n-1}^{-m} K_i J^i \end{aligned}$$

例如:八进制数123.45按权展开式为:

$$(123.45)_8 = 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1} + 5 \times 8^{-2}$$

又如:十六进制数 6CF.7A 按权展开为:

$$(6CF.7A)_{16} = 6 \times 16^2 + 12 \times 16^1 + 15 \times 16^0 + 7 \times 16^{-1} + 10 \times 16^{-2}$$

从以上分析可以看出,各种进位计数制的基本规律是共同的。表 1-1 所示给出了常用的几种进位计数制之间的关系。

表 1-1 常用进位计数制之间的对应关系

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
二进制	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000
八进制	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

为了书写方便,常用 B(Binary)表示二进制数,O(Octal)表示八制进数,H(Hexadecimal)表示十六进制数,D(Decimal)表示十进制数。

二、数制间的转换

计算机内部处理时使用二进制数,它的每一个数位只有 0 和 1 两个符号,因此,可用任何具有两个不同稳定状态的元件来表示。显然,制造具有两个稳定状态的元件比制造具有多个(如十个)稳定状态的元件要容易得多。如开关的闭合与断开,晶体管的截止与导通等等。只要我们规定一种状态表示“1”,另一种状态表示“0”,就能用于表示二进制数。然而,人们习惯于十进制数,要把它输入到计算机中参加运算,必须将其转换成二进制数,计算机运算的结果输出时,又要将二进制数转换成十进制数来显示或打印。这种数制之间的相互转换过程在计算机内频繁地进行着。当然,计算机内有专门的程序自动完成这些转换工作,但学习计算机仍有必要掌握数制转换的基本方法。

(一)十进制数与二、八、十六进制数之间的转换

(1)任意进制数转换成十进制

这种转换比较方便,只需将待转换的任意进制位按权展开成一个多项式,然后求该多项式的和就可以了。

例 1-1 将二进制数 1011111.11 转换成十进制数。

$$\begin{aligned} (1011111.11)_2 &= 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= (95.75)_{10} \end{aligned}$$

或写成:1011111.11B = 95.75D

例 1-2 将八进制数 156.72 转换成十进制数。

$$\begin{aligned} (256.72)_8 &= 2 \times 8^2 + 5 \times 8^1 + 6 \times 8^0 + 7 \times 8^{-1} + 2 \times 8^{-2} \\ &= (174.906)_{10} \end{aligned}$$

例 1-3 将十六进制数 D39A.3F 转换十进制数。

$$\begin{aligned} (D39A.3F)_{16} &= 13 \times 16^3 + 3 \times 16^2 + 9 \times 16^1 + 10 \times 16^0 + 3 \times 16^{-1} + 15 \times 16^{-2} \\ &= (54170.2461)_{10} \end{aligned}$$

(2)十进制数转换为二、八、十六进制数

十进制数转换成其它进位制数需要将整数部分和小数部分分别进行转换,然后把所得的

结果用小数点连接起来。假设把十进制数转换成 J 进制数(J 为 2、8 或 16),转换方法如下:

整数部分采用除 J 取余法:将 J 逐次去除待转换的十进整数,直至商小于基数 J 为止。每次所得的余数即为 J 进制数的数符,先得到的余数在低位,后得到的余数排在高位。

小数部分采用乘 J 取整法:逐次用 J 去乘待转换的十进制小数,将每次得到的整数部分依次记为 J 进制小数的 K_{-1} 、 K_{-2} 、...、 K_{-m} 。值得注意的是,并非每一个十进制小数都能用有限位的 J 进制小数来等值表示,也就是说“乘 J 取整”可以无限进行下去,此时只能用有限位 J 进制小数来表示近似值。

例 1-4 将十进制数 84.8125 转换成二进制数。其转换过程如图 1-3 所示。

整数部分:除 2 取余	小数部分:乘 2 取整
$ \begin{array}{r} 2 \overline{) 84} \\ 2 \overline{) 42} \quad \text{--- } 0(K_0) \\ 2 \overline{) 21} \quad \text{--- } 0(K_1) \\ 2 \overline{) 10} \quad \text{--- } 1(K_2) \\ 2 \overline{) 5} \quad \text{--- } 0(K_3) \\ 2 \overline{) 2} \quad \text{--- } 1(K_4) \\ 2 \overline{) 1} \quad \text{--- } 0(K_5) \\ 0 \quad \text{--- } 1(K_6) \end{array} $	$ \begin{array}{r} 0.8125 \\ \times \quad 2 \\ \hline 1.6250 \quad \text{--- } 1(K_1) \\ 0.625 \\ \times \quad 2 \\ \hline 1.250 \quad \text{--- } 1(K_2) \\ 0.25 \\ \times \quad 2 \\ \hline 0.50 \quad \text{--- } 0(K_3) \\ \times \quad 2 \\ \hline 1.0 \quad \text{--- } 1(K_4) \end{array} $

图 1-3 十进制数转换成二进制数

将整数部分和小数部分的转换结果用小数点连接起来,最后得:

$$(84.8125)_{10} = (1010100.1101)_2$$

例 1-5 将十进制数 678 转换成八、十六进制数。其转换过程如图 1-4 所示。

$$(678)_{10} = (1246)_8 \quad (678)_{10} = (2A6)_{16}$$

除 8 取余:	除 16 取余:
$ \begin{array}{r} 8 \overline{) 678} \\ 8 \overline{) 84} \quad \text{--- } 6(K_0) \\ 8 \overline{) 10} \quad \text{--- } 4(K_1) \\ 8 \overline{) 1} \quad \text{--- } 2(K_2) \\ 0 \quad \text{--- } 1(K_3) \end{array} $	$ \begin{array}{r} 16 \overline{) 678} \\ 16 \overline{) 42} \quad \text{--- } 6(K_0) \\ 16 \overline{) 2} \quad \text{--- } 10(\text{用 A 表示})(K_1) \\ 0 \quad \text{--- } 2(K_2) \end{array} $

图 1-4 十进制数转换成八进制与十六进制数

(二)二进制数与八、十六进制数之间的转换

由于一位八进制数正好对应于三位二进制数,一位十六进制数正好对应于四位二进制数,所以二进制与八、十六进制数之间存在着简单而直接的关系,它们之间的转换比较简单。

(1)二进制数转换为八进制与十六进制数

分别采用三位一并法和四位一并法:即以小数点为界,分别向左、右两个方向每三位或四位合为一组,不足三位或四位的以 0 补齐,然后对每三位或四位二进制数用相应的八进制或十六进制数符来表示。

例 1-6 将二进制数 11101.01101 分别转换成八进制与十六进制数。

011	101.	011	010	0001	1101.	0110	1000
↓	↓	↓	↓	↓	↓	↓	↓
3	5	3	2	1	D	3	8

$$(11101.01101)_2 = (35.32)_8 \qquad (11101.01101)_2 = (1D.38)_{16}$$

(2) 八进制数与十六进制数转换成二进制数

与上述转换过程相反,分别采用一分为三和一分为四法:将每一个八进制数符用三位二进制数来表示,每一个十六进制数符用四位二进制数来表示即可。

例 1-7 将八进制数 345.6, 十六进制数 DE.FC 分别转换成二进制数。

3	4	5	·	6	D	E	·	F	C
↓	↓	↓		↓	↓	↓		↓	↓
011	100	101		110	1101	1110		1111	1100

$$(357.4)_8 = (11100101.11)_2 \qquad (DE.FC)_{16} = (11011110.111111)_2$$

三、算术运算和逻辑运算

计算机中的算术运算包括加、减、乘、除四则运算,逻辑运算有逻辑与运算(逻辑乘法)、逻辑或运算(逻辑加)、逻辑非运算(逻辑否定)等。

(一) 算术运算

(1) 二进制加法

二进制数的加法运算法则:

$$\begin{aligned} 0+0 &= 0 & 0+1 &= 1 \\ 1+0 &= 1 & 1+1 &= 10 \text{ (本位为 0, 向高位进位为 1)} \end{aligned}$$

例 1-8 二进制数 1101 与 1001 相加。

其运算过程为:

$$\begin{array}{r} \text{被加数} \quad 1101 \\ \text{加数} \quad + 1001 \\ \hline \text{和} \quad 10110 \end{array}$$

$$\text{所以 } (1101)_2 + (1001)_2 = (10110)_2$$

由此可见,两个二进制数相加时,每一位最多有 3 个数(本位被加数、加数和来自低位的进位)相加,按二进制数的加法运算规则得到本位相加的和及向高位的进位。

(2) 二进制减法

二进制数的减法运算法则:

$$\begin{aligned} 0-0 &= 0 & 1-0 &= 1 \\ 1-1 &= 0 & 10-1 &= 1 \text{ (有借位)} \end{aligned}$$

例 1-9 二进制数 11000011 与 00101100 相减。

其运算过程为:

$$\begin{array}{r} \text{被减数} \quad 11000011 \\ \text{减数} \quad - 00101100 \\ \hline \text{差} \quad 10010111 \end{array}$$

$$\text{所以 } (11000011)_2 - (00101100)_2 = (10010111)_2$$

由此可见,两个二进制数相减时,每一位最多有 3 个数(本位被减数、减数和向高位的借位)参与运算,按二进制数的减法运算法则得到本位相减的差数和向高位的借位。

(3) 二进制乘法

二进制数的乘法运算法则:

$$\begin{array}{ll} 0 \times 0 = 0 & 0 \times 1 = 0 \\ 1 \times 0 = 0 & 1 \times 1 = 1 \end{array}$$

例 1-10 二进制数 1110 与 1101 相乘,其运算过程为:

$$\begin{array}{r} \text{被乘数} \quad 1111 \\ \text{乘数} \quad \times) \quad 1001 \\ \hline \text{部分积} \left\{ \begin{array}{r} 1111 \\ 0000 \\ 0000 \\ 1111 \end{array} \right. \\ \hline \text{积} \quad 10000111 \end{array}$$

$$\text{所以 } (1111)_2 \times (1101)_2 = (10000111)_2$$

由此可见,在两个二进制数相乘的过程中,每一个部分积取决于乘数,若乘数的相应位为 1,则部分积就是被乘数;若乘数的相应位为 0,则部分积为全 0;乘数有几位,就有几个部分积。在计算机中,二进制数的乘法是由加被乘数和移位两种操作来实现的。

(4) 二进制除法

二进制数的除法运算法则:

$$\begin{array}{ll} 0 \div 0 = 0 & 0 \div 1 = 0 \\ 1 \div 0 \text{ 无意义} & 1 \div 1 = 1 \end{array}$$

例 1-11 二进制数 100111 与 110 相除。

$$\begin{array}{r} 110 \\ 110 \overline{) 100111} \\ \underline{110} \\ 111 \\ \underline{110} \\ 11 \end{array}$$

$$\text{所以 } (100111)_2 \div (110)_2 = (110)_2 \cdots \cdots \text{余}(11)_2$$

与乘法类似,二进制数的除法可以由减除数和移位两种操作实现。

(二) 逻辑运算

逻辑运算是计算机的基本功能之一。逻辑量用来表示一个条件的判断结果,如“ $X > 0$ ”、“M 是偶数”、“ $A = B$ ”等都是表示一个条件。每个条件的判断结果只可能取两个值:真(条件满足)或假(条件不满足)。由于计算机只对数字 0 和 1 进行运算,所以很容易用它们来表示逻辑量,即用“1”表示“真”,“0”表示“假”。注意,这里的 0 和 1 并不表示数字的大小,而是表示条件的两种可能性。逻辑运算有三种:与运算、或运算以及非运算。

(1) 与运算

逻辑与运算也叫逻辑乘法。通常用符号“ $\times$ ”或“ $\cdot$ ”或“ $\wedge$ ”表示与运算。“与”的运算法则为:

$$0 \times 0 = 0 \quad 0 \times 1 = 0$$

$$1 \times 0 = 0 \quad 1 \times 1 = 1$$

由此可以看出,进行逻辑与运算时,只有当两者都是 1(真)时,其结果才为 1,其余情况下结果均为 0(假)。请注意,上述与运算虽与算术乘法形式相似,但含义并不相同。

例 1-12 按位进行逻辑与运算。设 $A = (10101001)_2$, $B = (11000011)_2$, 求 $F = A \times B$ 。

$$\begin{array}{r} 10101001 \\ \times 11000011 \\ \hline 10000001 \end{array}$$

所以, $F = (10000001)_2$

根据 F 和 B 的内容,可以判断出 A 右边的第一位、第二位是否为 1,因为按照逻辑与的运算法则,只有运算数同时为 1 时,结果才为 1。可见,按位与运算的典型用法是可检测出一个二进制串的哪些位为 1,或取一个二进制串的某几位。

(2)或运算

逻辑或运算又称为逻辑加法,常用符号“+”或“V”表示。“或”的运算法则如下:

$$0 + 0 = 0 \quad 0 + 1 = 1$$

$$1 + 0 = 1 \quad 1 + 1 = 1$$

顾名思义,“或”就是或者的意思。当两个逻辑量中有一个为 1,或运算的结果就为 1。

例 1-13 按位进行逻辑或运算。设 $A = (10010100)_2$, $B = (00111001)_2$, 求 $F = A + B$ 。

$$\begin{array}{r} 10010100 \\ + 00111001 \\ \hline 10111101 \end{array}$$

所以 $F = (10111101)_2$

从上述运算结果可以看出,如果将运算结果 F 送回 A 中,则 B 中为 1 的位能使 A 中相应为 0 的位修正为 1。如 A 从右起的第 1、4、6 位本来是 0,由于 B 数的相应位为 1,或运算的结果使 A 第 1、4、6 位变 1。按位或运算的典型用法是将一个二进制串的某几位置 1。

(3)非运算

逻辑非运算又称为逻辑否定,通常在逻辑量上加一横线表示“非”运算。非运算的法则如下:

$$\overline{0} = 1 \quad \overline{1} = 0$$

例 1-14 按位进行逻辑非运算。设 $A = (11010100)_2$, 求 $F = \overline{A}$ 。

$$F = \overline{A} = \overline{(11010100)_2} = (00101011)_2$$

四、二进制编码

我们知道,在计算机中信息的存储和处理必须以二进制形式表示。对于十进制数可以把它转换成相应的二进制数,也可以将十进数的每一位变成二进制形式,即二进制编码的十进制数(BCD 码);对于字符则要用二进制编码来表示,字符的二进制编码有 ASCII 码,EB CDIC 码等,目前在微机系统中用得最多的是 ASCII 码。

1. BCD 码

BCD(Binary Coded Decimal)码是一种用四位二进制数表示一位十进制数的编码,也称二—十进制编码。它即具有二进制数的形式,又保持了十进制的特点,可以作为人机联系时的一种

中间表示方法,也可以用它直接进行运算。

BCD 码又有多种,其中最常用的 BCD 码是 8421 码,这是一种有权码,从最高位起,四位数的权分别为 8、4、2、1。由于它与十进制数的二进制表示相对应,故又称为自然(Natural)BCD 码(NBCD),如表 1-2 所示。由于 10 到 15 这六个数在 8421 码中没有意义,因此若用 8421 码参加运算,一旦在运算中出现超过 9 的数(如 1010、1011 等),就必须设法调整,才能保证结果正确。8421 码比较直观,便于识别。

表 1-2 8421 码表

十进制数	8421 码	十进制数	8421 码
0	0000	8	1000
1	0001	9	1001
2	0010	10	0001 0000
3	0011	11	0001 0010
4	0100	12	0001 0011
5	0101	13	0001 0011
6	0110	14	0001 0100
7	0111	15	0001 0101

例 1-15 写出十进制数 3456.789 的 BCD 码表示形式。

$$(3456.789)_{10} = (0011\ 0100\ 0101\ 0110.0111\ 1000\ 1001)_{\text{BCD}}$$

2. ASCII 码

ASCII 码(American Standard Code for Information Interchange)即美国标准信息交换码,它是国际上广泛采用的一种字符编码。ASCII 码由七位二进制码组成,如表 1-3 所示。其中包括 34 个控制字符(第 0 列和第 1 列共 32 个,再加上空格字符 SP 和删除字符 DEL)和 94 个图形字符,共计 128 个字符。

表中列 b_7 、 b_6 、 b_5 为高 3 位,行 b_4 、 b_3 、 b_2 、 b_1 为低 4 位,组合起来即构成了七位二进制码,即 $b_7b_6b_5b_4b_3b_2b_1$ 。例如,要查 B 的 ASCII 码,应先找到其所在的行和列:

第 2 行,即 $b_4b_3b_2b_1 = 0010$;第 4 列,即 $b_7b_6b_5 = 100$

因此,B 的 ASCII 码是 1000010,八进制形式为 102,十六进制形式为 42,十进制形式为 66。

虽然 ASCII 码是七位编码,但由于一个字节是 8 个二进制位。因此,实际使用中,ASCII 码占一个字节,最高位一般约定为 0。

ASCII 码的组成有它的规律性,从表 1-3 中可以看出,按 ASCII 码值从小到大的顺序,依次是控制字符、数字(0~9)、大写字母(A~Z)、小写字母(a~z)、其中空格最小。

五、汉字编码

汉字在计算机中通常用两个字节的编码来表示。我国制定了“中华人民共和国国家标准信息交换汉字编码”,代号“GB2312-80”。这种编码称为国标码。在该标准编码字符集中共收录了汉字和图形符号 7445 个,其中一级汉字 3755 个,二级汉字 3008 个,图形符号 682 个。

表 1-3 ASC II 码表

b ₄ b ₃ b ₂ b ₁	列 行	b ₇	0	0	0	0	1	1	1	1
		b ₆	0	0	1	1	0	0	1	1
		b ₅	0	1	0	1	0	1	0	1
			0	1	2	3	4	5	6	7
0000	0	32 个 控 制 字 符		SP	0	@	P	,P		
0001	1			!	1	A	Q	a	q	
0010	2			"	2	B	R	b	r	
0011	3			#	3	C	S	c	s	
0100	4			\$	4	D	T	d	t	
0101	5			%	5	E	U	e	u	
0110	6			&	6	F	V	f	v	
0111	7			'	7	G	W	g	w	
1000	8			(	8	H	X	h	x	
1001	9			)	9	I	Y	i	y	
1011	A			*	:	J	Z	j	z	
1011	B			+	:	K	[	K	?	
1100	C			,	<	L	\	l	!	
1101	D			-	=	M	]	m	:	
1110	E			●	>	N	^	m	~	
1111	F			/	?	O	-	o	DEL	

国标 GB2312-80 规定,全部国标汉字及符号组成一个 94×94 的矩阵。在此正方形的矩阵中,每一行称为一个“区”,每一列称为一个“位”。这样,就组成了一个有 94 个区(01~94 区),每个区内有 94 个位(01~94 位)的汉字字符集。区码和位码简单地组合在一起(即两位区码居高位,两位位码居低位)就形成了“区位码”。区位码可以唯一确定某一个汉字或符号,反之任何一个汉字或符号都对应唯一的区位码。如汉字“啊”的区位码是“1601”(即在 16 区的第 1 位),符号“、”的区位码为“0102”。

所有汉字及符号的 94 个区划分为如下四个组:

①1~15 区:图形符号区,其中 1~9 区为标准区,10~15 区为自定义区。

②16~55 区:一级常用汉字区共 3755 个汉字。该区的汉字按汉语拼音排序,同音字按笔划顺序排序,55 区的 90~94 位未定义汉字。

③56~87 区:二级非常用汉字区共 3008 个汉字。按该区的汉字部首排序。

④88~94 区:自定义汉字区。

汉字的内码是从上述区码和位码的基础上演变而来的。它是在计算机内部进行存储、传输所使用的汉字代码。

区码或位码的范围都在 1~94 内,如果直接用它作内码就与基本 ASCII 码冲突,所以汉字的两位内码以如下的运算规则确定:

高位内码 = 区码 + 20H + 80H

低位内码 = 位码 + 20H + 80H

在上述的运算规则中加 20H 应理解为避开基本 ASCII 码的控制码,加 80H 意在把最高二进制位置 1,以与基本的 ASCII 码(其最高二进制位为 0)相区别。高位内码和低位内码的取值范围均在 161 ~ 255(即 A1H ~ FEH)。

第三节 基本逻辑电路

计算机所执行的逻辑运算是依靠逻辑电路完成的。逻辑电路由基本门电路组成。基本门电路包括:与门、或门及非门。其真值表如表 1-4 所示。

表 1-4 基本门电路真值表

输入端 A B		与门 A·B	或门 A + B	非门 $\bar{A}$ B	
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

(一)与门、或门及非门电路

(1)与门

与门是具有两个或两个以上输入端和一个输出端,且有逻辑乘功能的电路。与门电路如图 1-5(a)所示。只有 A、B 两个输入端都是高电位时,电阻 R 方无电流通过,这样 X 端为高电位。符号表示如图 1-5(b)所示。

(2)或门

或门是具有两个或两个以上输入端和一个输出端且具有逻辑加功能的电路。或门电路如图 1-5(c)所示。只要 A、B 中有一个高电位,X 端即为高电位;两个都是低电位时,X 端才是低电位。符号表示如图 1-5(d)所示。

(3)非门

非门又称反相器,是具有逻辑非功能的电路,电路图如图 1-5(e)所示。当 A 端为低电位时,三极管内基本无电流,X 端为高电位。反之,若 A 端为高电位时,三极管导通,X 端为低电位。符号表示如图 1-5(f)所示。

表 1-5 半加器真值表

输入变量 A B		输出变量 S C	
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

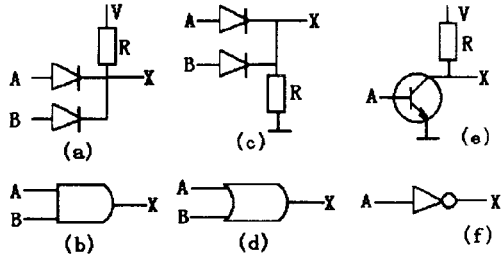


图 1-5 基本门电路

(二)逻辑电路的实现

这里通过对计算机运算器中的重要部分——半加法器的设计,来理解如何用基本门电路

实现逻辑运算的机理。半加器的真值表如表 1-5 所示。

由表 1-5 可知:

$$S = \bar{A} \cdot B + A \cdot \bar{B} = (A + B)(\overline{A \cdot B})$$

$$C = A \cdot B$$

变换后 S 和 C 都含有项 $A \cdot B$, 这样使设计得以简化。其逻辑电路如图 1-6 所示。

只要从 A、B 两端输入代表 0、1 的低、高电位, 即可在输出端 S 和 C 上得到运算后的值。它同样是用低、高电位来表示 0、1。

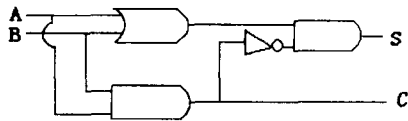


图 1-6 半加器电路

(三)基本逻辑部件

与门、或门、非门是逻辑电路的最基本形式,但在实际应用中,通常是利用它们之间的不同组合构成复合门

来进行运算和控制的。这是因为单独的二极管门通过多级传输,输出信号幅度将减小,波形的前沿产生衰变,而且带负载能力也差,所以,往往在二极管门的输出端接上三极管,这样就改善了逻辑电路的性能。当采用了晶体管放大元件后,就出现了与非、或非、与或非等复合门电路。在计算机中应用最多的由复合门电路组成的部件有触发器、寄存器、计数器及译码器等。

(1)触发器

它是一种存储部件,以低电平或高电平的方式存储一位二进制代码。

(2)寄存器

由 N 个触发器组成的部件可存储 N 位二进制代码,称之为 N 位寄存器。通常,将能够存放代码的寄存器称为缓冲寄存器,在进行算术运算和某些逻辑运算时,常常要把寄存器中的数向右移。这就需要具有移位功能的寄存器。这种寄存器称之为移位器。

(3)计数器

用来对输入的脉冲进行计数的部件,一般由若干个触发器组成。

(4)译码器

将寄存器中存放的数码翻译成另一种控制代码的部件。

第四节 计算机常用的基本术语

在学习使用计算机的时候,常会遇到有关的一些名词术语。了解这些术语是学习、使用计算机的前提。此处介绍几个最常见的基本术语:

(1)硬件:组成一台计算机所有固定装置的总称叫计算机的硬件。硬件是计算机工作的物质基础,是计算机软件发挥作用施展其技能的舞台。它是看得见摸得着的硬设备。例如:计算机的主机、打印机、显示器等都是计算机的硬件。

(2)软件:计算机软件是指指挥计算机工作的各种程序的集合。软件是计算机的灵魂,它的任务是发挥和扩大机器的功能,提高机器的使用效率。

(3)程序:完成一定处理功能的指令集合称为程序。在人们使用计算机时,必须把要解决的问题按处理步骤编成一条条的指令,这些指令必须是计算机能识别和执行的。

(4)指令:指挥计算机进行基本操作的命令。一条指令包括二部分内容:操作码和操作数。操作码,它指明操作的性质;操作数,它完成操作所需要的数。

(5)指令系统:计算机所能执行的全部指令叫计算机的指令系统。指令系统在一定程度上