



高等学校“十二五”规划教材

# 单片机原理及应用

DANPIANJI YUANLI JI YINGYONG

徐汉斌 熊才高 主 编



华中科技大学出版社  
<http://www.hustp.com>

TP368.1  
764

013033027

内容简介

高等学校“十二五”规划教材

# 单片机原理及应用

主 编 徐汉斌 熊才高  
副主编 刘璐玲 蔡 丽 肖凌俊



华中科技大学出版社  
中国·武汉



北航

C1640744

TP368.1  
764

## 内 容 简 介

本书围绕 MCS-51 这一经典单片机讲解了单片机开发的思想和方法,系统地介绍了 MCS-51 单片机内部的功能结构、软硬件资源的原理与应用,并介绍了 MCS-51 单片机系统的不同外部扩展方法。全书共 10 章,主要内容包括计算机基础知识、单片机的硬件结构和原理、汇编指令系统、单片机程序设计、单片机的中断与定时、单片机串行通信、单片机存储器扩展、单片机 I/O 口扩展、单片机 A/D 和 D/A 扩展、单片机综合应用。

本书提供的实例兼顾了教学与实际应用,大部分实例可以在稍加修改后直接应用于实际开发中,为实际应用提供了基本开发的“范本”。书中结合实例,讲解了单片机开发中的工具软件 Proteus 和 Keil 的用法,便于读者在实际中进行验证。本书最后提供了综合应用的实例,可作为相关专业学生和工程技术人员的参考资料。

本书可作为高等院校自动化、电气工程及其自动化、计算机应用、通信工程、控制工程、电子信息工程,以及机电类等专业的教材或教学参考书,也可供从事相关专业的技术人员参考。

### 图书在版编目(CIP)数据

单片机原理及应用/徐汉斌 熊才高 主编. —武汉:华中科技大学出版社,2013.4  
ISBN 978-7-5609-8658-6

I. 单… II. ①徐… ②熊… III. 单片微型计算机-高等学校-教材 IV. TP368.1

中国版本图书馆 CIP 数据核字(2013)第 011026 号

### 单片机原理及应用

徐汉斌 熊才高 主编

策划编辑:袁 冲

责任编辑:狄宝珠

责任校对:刘 竣

封面设计:李 嫒

责任监印:张正林

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321915

录 排:华中科技大学惠友文印中心

印 刷:华中科技大学印刷厂

开 本:787mm×1092mm 1/16

印 张:19.5

字 数:500 千字

版 次:2013 年 4 月第 1 版第 1 次印刷

定 价:39.00 元



本书若有印装质量问题,请向出版社营销中心调换  
全国免费服务热线:400-6679-118 竭诚为您服务  
版权所有 侵权必究

前

言

当今单片机-嵌入式技术已经渗入到我们生活的各个方面,广泛应用于工业控制、仪器仪表、办公设备、汽车电子、医疗仪器、家用电器和娱乐电子等领域。MCS-51 系列 8 位单片机,以其经典的架构、不断扩展的功能、开放的体系和易于实现的控制等特点,盛行 30 多年而不衰。本书围绕 MCS-51 这一经典单片机,系统介绍其内部功能结构、软硬件资源的原理与应用,以及以 51 单片机为核心如何进行扩展外部电路从而提升系统功能的方法。希望通过 51 单片机的学习,读者能够掌握单片机系统开发的基本思想和方法,而不仅仅只是掌握 51 单片机技术本身。

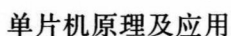
本着“扎实的基础、开放的思想、实战的能力”的思想,本书力求在内容取舍、顺序编排、实例组织和教学方法上有所改进,使读者能够快速理解单片机内部各功能模块的应用特点,掌握控制电路设计和程序开发的基本工具和方法。从系统功能需求出发,来构思系统硬件和软件的构成,综合硬件与软件各自的优势,对系统各部分构成进行选择,再到实现产品整体设计思想,进而提高综合运用计算机软硬件知识来解决实际问题的能力。

本书从应用角度出发,按照汇编和 C51 语言编程两条线;组织应用实例,让读者以汇编入手,理解程序对单片机硬件直接控制,最终过渡到用 C51 实现单片机系统编程。所采用的实例既兼顾讲课需要,又可以扩展为实际工程应用。同时书中提供了完整的开发实例,并讲解了开发的基本步骤和开发工具,方便读者通过实际应用掌握单片机应用系统的开发。

本书共分 10 章,主要内容包括计算机基础知识、单片机的硬件结构和原理、汇编指令系统、单片机程序设计、单片机的中断与定时、单片机串行通信、单片机存储器扩展、单片机 I/O 口扩展、单片机 A/D 和 D/A 扩展、单片机综合应用。

本书既可以作为高等院校自动化、电气工程及其自动化、计算机应用、通信工程、控制工程、电子信息工程,以及机电类等专业的教学用书,也可供有关院校师生和从事有关单片机应用与产品开发等工作的工程技术人员参考。读者阅读此书,需要掌握一些数字电路和 C 语言基础方面的知识。

本书由武汉理工大学徐汉斌、湖北工业大学商贸学院熊才高、武汉工程大学邮电信息学院刘璐玲、武汉东湖学院蔡丽、武汉理工大学华夏学院肖凌俊编著。第 1、4、10 章由徐汉斌编写,第 2、5 章由熊才高编写,第 3 章由肖凌俊编写,第 6、7 章由刘璐玲编写,第 8 章由熊才高、徐汉斌编写,第 9 章由蔡丽编写。全书由徐汉斌负责整理和统稿。



由于编者的水平有限,加之单片机及其应用技术也在不断发展,故书中难免存在不完善及欠妥之处,恳请同行及广大读者批评指正。

2012 年 9 月

|                               |      |
|-------------------------------|------|
| 第1章 计算机基础知识 .....             | (1)  |
| 1.1 计算机原理概述 .....             | (1)  |
| 1.2 二进制数在计算机中的应用 .....        | (3)  |
| 1.3 程序设计常用进制数 .....           | (9)  |
| 1.4 BCD 码与 ASCII 码 .....      | (11) |
| 本章小结 .....                    | (13) |
| 习题 .....                      | (13) |
| 第2章 单片机的硬件结构和原理 .....         | (15) |
| 2.1 80C51 单片机芯片的基本结构 .....    | (15) |
| 2.2 80C51 单片机的引脚功能 .....      | (16) |
| 2.3 80C51 单片机的内部组成和内部结构 ..... | (19) |
| 2.4 80C51 单片机的并行口结构与操作 .....  | (28) |
| 2.5 80C51 单片机的时序与低功耗 .....    | (34) |
| 2.6 80C51 单片机的最小系统 .....      | (38) |
| 本章小结 .....                    | (38) |
| 习题 .....                      | (39) |
| 第3章 汇编指令系统 .....              | (40) |
| 3.1 汇编指令格式构成 .....            | (40) |
| 3.2 80C51 单片机指令的寻址方式 .....    | (41) |
| 3.3 80C51 单片机指令分类 .....       | (45) |
| 本章小结 .....                    | (62) |
| 习题 .....                      | (62) |
| 第4章 单片机程序设计 .....             | (65) |
| 4.1 程序设计概述 .....              | (65) |
| 4.2 汇编语言的伪指令 .....            | (66) |
| 4.3 C51 程序设计基础 .....          | (68) |
| 4.4 程序基本结构形式 .....            | (75) |
| 4.5 80C51 单片机程序设计举例 .....     | (80) |
| 4.6 Keil 与 Proteus 设计工具 ..... | (86) |



|                             |       |
|-----------------------------|-------|
| 本章小结                        | (100) |
| 习题                          | (100) |
| <b>第5章 单片机的中断与定时</b>        | (103) |
| 5.1 中断概念                    | (103) |
| 5.2 80C51 单片机的中断控制系统        | (105) |
| 5.3 80C51 中断处理过程            | (108) |
| 5.4 中断的编程和应用举例              | (110) |
| 5.5 单片机的定时/计数器              | (114) |
| 本章小结                        | (126) |
| 习题                          | (126) |
| <b>第6章 单片机串行通信</b>          | (131) |
| 6.1 串行通信基础知识                | (131) |
| 6.2 单片机串行口                  | (135) |
| 6.3 I2C 通信实现                | (146) |
| 6.4 SPI 通信实现                | (155) |
| 本章小结                        | (157) |
| 习题                          | (158) |
| <b>第7章 单片机存储器扩展</b>         | (159) |
| 7.1 存储器分类                   | (159) |
| 7.2 单片机并行扩展                 | (160) |
| 7.3 存储器并行扩展                 | (165) |
| 7.4 存储器串行扩展                 | (174) |
| 本章小结                        | (177) |
| 习题                          | (177) |
| <b>第8章 单片机 I/O 口扩展</b>      | (178) |
| 8.1 用 TTL 芯片扩展简单的 I/O 接口    | (178) |
| 8.2 可编程并行 I/O 口 8255A       | (180) |
| 8.3 单片机键盘扩展                 | (192) |
| 8.4 单片机显示扩展                 | (198) |
| 本章小结                        | (207) |
| 习题                          | (207) |
| <b>第9章 单片机 A/D 与 D/A 扩展</b> | (208) |
| 9.1 模数与数模转换                 | (209) |
| 9.2 A/D 并行转换接口              | (223) |
| 9.3 D/A 并行转换接口              | (234) |
| 9.4 A/D 与 D/A 串行转换接口        | (241) |
| 本章小结                        | (248) |



|                               |       |
|-------------------------------|-------|
| 习题.....                       | (249) |
| <b>第 10 章 单片机综合应用</b> .....   | (252) |
| 10.1 单片机应用系统开发的基本方法.....      | (252) |
| 10.2 时钟秒表设计.....              | (257) |
| 10.3 A/D 和 D/A 综合应用系统设计 ..... | (283) |
| 本章小结.....                     | (301) |
| 习题.....                       | (301) |
| <b>参考文献</b> .....             | (302) |

# 第 1 章 计算机基础知识

## 【学习目标】

- (1) 理解计算机系统的基本构成及工作原理。
- (2) 掌握二进制、十进制和十六进制运算规则及三者间的转换方法。
- (3) 熟练运用数的不同编码表示方法及转换。

## 1.1 计算机原理概述

单片机实际上是将计算机的多种基本功能集中到单个芯片的大规模集成电路芯片,其工作原理与微型计算机基本相同,技术上也是相通的。因此可以说单片机也是计算机中的一种类型,通过了解计算机的基本构成及工作原理,也有助于单片机的学习。计算机系统包括硬件系统和软件系统,其组成框图如图 1.1 所示。

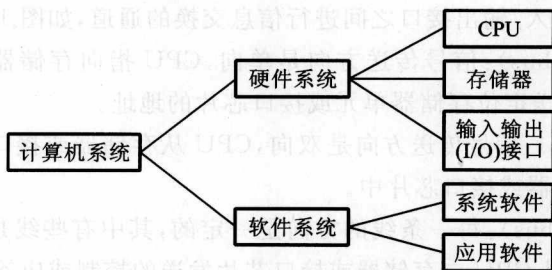


图 1.1 计算机系统组成框图

### 1.1.1 计算机硬件系统

#### 1. 冯·诺依曼体系

现代的计算机系统是建立在冯·诺依曼提出的以程序存储原理为基础的计算机理论之上的。而这一理论将计算机硬件体系结构划分为运算器、控制器、存储器、输入设备和输出设备五大部件。运算器主要完成算术和逻辑运算,控制器主要完成系统的控制信号的输入及控制信号的输出,以控制和协调系统的运行。运算器和控制器一般在中央处理单元

(CPU)中,是计算机系统的最终执行单元,按照一定的设计模式去执行程序,从而实现所要求的功能,因此,可以认为 CPU 就是一个自动执行单元。而 CPU 执行的程序需要预先准备好,也就是说,需要把编好的程序放到特定的地方存放好,让 CPU 按照规定模式自动读取程序,并执行程序,所以系统中就需要安排专门存放程序 and 数据的存储器。计算机系统又需要与外界进行交互,因而需要设计输入和输出部分,在此主要指输入(input)和输出(output)接口部分。输入接口完成外部信号接收,输出接口完成对外部设备的控制信号输出。计算机五大部分及系统三总线框图如图 1.2 所示。

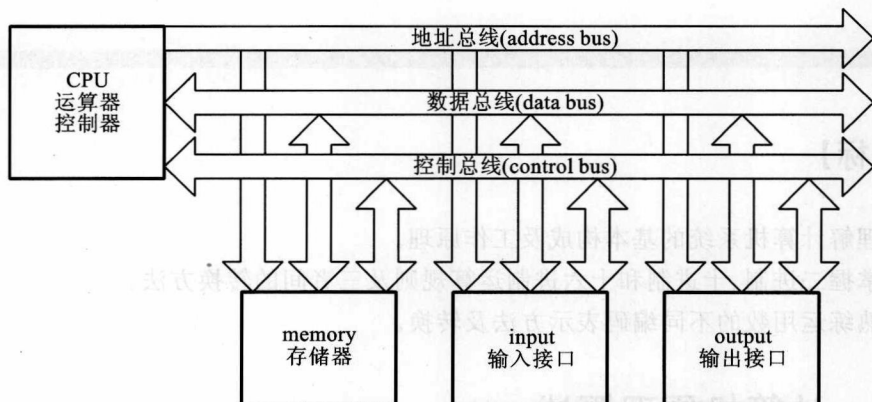


图 1.2 计算机五大部分及系统三总线框图

## 2. 计算机系统信息传送通道

计算机五大部分之间,需要经常进行信息(数据)的交换,为此在系统中设计了联系五大部分的传送通道,这个通道称为总线。所谓的总线就是指连接各个单元实现特定功能的一组连线。根据总线在系统中不同的作用,可将总线分为地址总线、数据总线和控制总线。它们是 CPU 与存储器、输入/输出接口之间进行信息交换的通道,如图 1.2 所示。

**地址总线(address bus):**信号传送方向是单向,CPU 指向存储器或输入/输出接口芯片,CPU 通过该总线传送定位存储器单元或接口芯片的地址。

**数据总线(data bus):**信号传送方向是双向,CPU 从存储器或接口芯片读取数据,CPU 也可以将数据写入存储器或接口芯片中。

**控制总线(control bus):**每一条线的方向是一定的,其中有些线是接口芯片向 CPU 发送的请求信号,有些线是 CPU 向存储器或接口芯片发送的控制或应答信号。

### 1.1.2 计算机软件系统

只有硬件而没有软件的计算机是无法完成任何工作的。计算机总是不断从存储器中读取预先放在存储器中的程序,按照程序执行各种操作,最终完成各项任务,常将这些程序称为软件。一般计算机系统使用的软件,按其功能可划分为两大类:系统软件和应用软件。系统软件主要是为各种应用软件提供一个平台,如操作系统、数据库管理系统、工具软件和各种程序设计语言等;应用软件是指为解决计算机应用问题而编写的具有专门用途的程序,例如,科学计算、图形图像处理、计算机辅助设计及制造、计算机辅助教学、文字处理等应用



软件。

但是并不是所有计算机系统都必须有系统软件。单片机技术主要用于嵌入式系统中,一般嵌入式系统以单片机为核心,根据功能要求进行硬件扩展,系统的软件是专用的,即根据系统要求相应设计专门的控制软件。

### 1.1.3 计算机的工作过程

计算机的工作流程,就是在 CPU 的控制下从存储器中读取程序指令并逐条执行的过程。指令的执行一般分为两个阶段:读取指令阶段和执行指令阶段。

#### 1. 读取指令阶段

在 CPU 的控制下,由内部程序指示器形成指令存储地址,从存储器对应地址中读出要执行的指令,将该指令送到指令寄存器中保存。

#### 2. 执行指令阶段

在 CPU 的控制下,由指令译码器(也称微代码译码器)对指令进行译码,产生完成该指令操作所需要的各种定时和控制信号,并执行该指令所规定的操作。

程序中的每条指令的执行都分为读取指令阶段和执行指令阶段,这一过程一直进行到程序执行结束为止。因此,计算机可以看做是自动执行预先存入存储器的程序的全自动机器,如何安排好程序决定了系统最终完成功能的好坏。

## 1.2 二进制数在计算机中的应用

由于计算机是建立在二进制基础之上的,而且计算机的电路逻辑和处理方法也都是按二进制的原则实现的,因此在计算机中所有指令、数据、字符和地址的表示,以及它们的存储、处理和传送,都是以二进制形式进行的。计算机中处理的任何信息最终都将转化成二进制存储,因此有必要先了解二进制的相关知识。

### 1.2.1 二进制数

#### 1. 进位计数制

进位计数制是一种常用的计数方法,计算机中也用进位计数制表示数,其中最常用的进位计数制有二进制、十六进制和十进制三种。那么进位计数制是如何表示的呢?下面以大家熟悉的十进制数为例进行说明。例如,十进制数:

888.88

同样的数字符号 8,但从左向右第 1 个 8 的值是 800,第 2 个 8 的值是 80,以下依次为 8、0.8 和 0.08。因此,可把这个数的值展开成如下幂级数形式:

$$888.88 = 8 \times 10^2 + 8 \times 10^1 + 8 \times 10^0 + 8 \times 10^{-1} + 8 \times 10^{-2}$$

式中,各位数的值都是该位数数码乘以一个不同的幂系数,通常把这个幂系数称作该位数的权。数码的位置不同,权的大小也不同,但它们都是 10 的幂次形式。其中整数位的权幂是该数码所在位数减 1,从低位向高位依次为  $10^0, 10^1, 10^2, \dots$ ;而小数位的权幂是该位数数码的

位数的负数,从高位向低位依次为  $10^{-1}, 10^{-2}, 10^{-3}, \dots$ 。对于十进制数,共有 10 个数码,即 0、1、2、3、4、5、6、7、8、9,数码的个数称为基数,因此十进制的基数为 10,逢 10 进位,各位数的权是 10 的幂次形式。分析可得进位计数制具有如下特点。

- (1) 数码的个数等于进位计数制的基数。
- (2) 逢基数进位。
- (3) 数字的权与其位置有关,且为基数幂的形式。

## 2. 二进制数(binary)

二进制数只有两个数码 0 和 1,因此其基数为 2,逢 2 进位,各位数的权是 2 的幂次形式。例如,二进制数 1011,其值为十进制数 11。该值是由下式计算出来的:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11$$

由此可以看出,二进制数与十进制数表示同一个数值时,十进制的位数少,二进制的位数多。十进制表示起来更简洁,但是计算机最终执行是以二进制来进行的,这是由计算机系统逻辑电路构成而决定的,因此二进制是计算机运行的基础。

### 1.2.2 计算机中二进制数的存储单位

在计算机中,表示数据或信息全部用的是二进制数,数据一般存储在存储器中,常用的表示二进制数的 3 个基本单位从小到大依次为:位、字节和字。

#### 1. 位(bit)

位(bit)是二进制数的最小单位,读作“比特”。在计算机中位仅能存放一位二进制数码 1 或 0,一般可用于表示两种状态,如“开”或“关”,“真”或“假”。

#### 2. 字节(byte)

字节(byte)是由 8 位二进制数构成的,读作“拜特”。字节是计算机最基本的数据单位,也是衡量数据量多少的基本单位,计算机中的数据、代码、指令、地址基本上是以字节为单位进行存储的。

#### 3. 字(word)

字(word)是由两个字节的二进制构成,即 16 位二进制数。将两个字的构成定义为双字。这些定义在计算机中只是用于表示二进制数的大小。在计算机系统中,字用于衡量计算机一次性处理数据的能力,把字定义为一台计算机上所能并行处理的二进制数,字的位数(或长度)称为字长。字长必须是字节的整数倍。如:MCS-51 单片机字长 8 位,AVR16 单片机字长 16 位,ARM 单片机字长 32 位。

### 1.2.3 机器数的表示形式

#### 1. 机器数与真值

由于计算机只认二进制数,而实际表示数据时,有时需要带上正(+)、负号(-)。要把这样的二进制数表示成计算机认识的数,就需要将正、负号进行数值化处理,将正号数值化为一位二进制“0”,将负号数值化为一位二进制“1”。由此转化得到的二进制数称为机器数,也就是计算机认识的数,那么转化前原来带正、负符号的数对应地称为真值。



## 2. 无符号数与有符号数的机器数表示

无符号数的机器数的全部二进制数码只表示数值大小,无正负概念。在计算中常用于表示无正、负概念的数、代码和存储器的地址等。此时的机器数和真值是一样的。

有符号数具有正、负的概念,实际书写时需要在数值前添加正负号,例如:

十进制的+56,转换成二进制数为+0111000

十进制的-56,转换成二进制数为-0111000

这样表示的数称为真值。但是计算机内部是无法在二进制数前直接加上正负号的,所以采用符号数值化方法来实现有符号数的表示方法。将代表正号的“0”或代表负号的“1”放到表示数值大小的数码序列的最前面。以计算机中典型基本数据单位一个字节8位二进制数为例,其机器数由1位符号位和7位数值构成,例如:

十进制的+56,转换成机器数为00111000

十进制的-56,转换成机器数为10111000

由此可见,有符号数的机器数的最高位表示的是符号位,其余位为数值位。

## 3. 小数的表示

在数值计算中,小数的存在是不可避免的,计算机中采用了定点数和浮点数两种机器数表示方法来解决小数点的问题。定点和浮点中的“点”是指小数点。小数点并没有进行数值化处理。计算机中只是做了数据表示格式的处理。定点数中小数点的位置是固定的。按小数点在计算机中表示数时规定的位置,定点数又分为定点小数和定点整数两种表示方法。而浮点数的小数点位置是不固定的。

## 1.2.4 机器数的原码、反码与补码

机器数有原码、反码和补码共3种表示方法。一个数有多种表示方法是简化运算电路和提高运算速度的需要。

### 1. 原码

原码是将二进制数符号数值化以后的表示形式,是机器数的原始表示,是对应于反码和补码的称呼。

### 2. 反码

正数的反码与原码相同。例如:

十进制数+76的原码表示为  $[+76]_{\text{原}} = 01001100$

十进制数+76的反码表示为  $[+76]_{\text{反}} = 01001100$

十进制数+0的反码表示为  $[+0]_{\text{反}} = 00000000$

负数的反码是由原码转换得到的,转换方法如下:符号位不变,数值位按位取反。例如:

十进制数-76的原码表示为  $[-76]_{\text{原}} = 11001100$

十进制数-76的反码表示为  $[-76]_{\text{反}} = 10110011$

十进制数-0的原码表示为  $[-0]_{\text{原}} = 10000000$

十进制数-0的反码表示为  $[-0]_{\text{反}} = 11111111$

### 3. 补码

正数的补码与原码相同。例如:



十进制数+8的原码表示为  $[+8]_{\text{原}} = 00001000$

十进制数+8的反码表示为  $[+8]_{\text{反}} = 00001000$

十进制数+8的补码表示为  $[+8]_{\text{补}} = 00001000$

十进制数+0的补码表示为  $[+0]_{\text{补}} = 00000000$

负数的补码是把反码的最低位加1。例如:

十进制数-8的原码表示为  $[-8]_{\text{原}} = 10001000$

十进制数-8的反码表示为  $[-8]_{\text{反}} = 11110111$

十进制数-8的补码表示为  $[-8]_{\text{补}} = 11111000$

十进制数-0的补码表示为  $[-0]_{\text{补}} = 00000000$

原码、反码和补码都是二进制符号数的表示方法,其共同特点是:最高位为符号位,正数的原码、反码和补码相同。由上面的例子可见-0和+0的原码和反码是不一样的,而两者的补码是一样的,由此在计算机中为了保持一致性,正负数都用补码方式表示。此外,还应注意以下两点。

(1) 负数补码的转换过程是:原码→反码→补码。

(2) 负数的补码再取补就得到原码,以十进制数-85为例进行说明。例如:

$[-85]_{\text{原}} = 11010101$

$[-85]_{\text{反}} = 10101010$

$[-85]_{\text{补}} = 10101011$

$[(-85)_{\text{补}}]_{\text{反}} = 11010100$

$[(-85)_{\text{补}}]_{\text{补}} = 110101001 = [-85]_{\text{原}}$

讲解原码和反码是为了引出补码的概念,因为补码在计算机中被广泛采用。补码运算可以将符号位当成数据位对待,因此可以把有符号数与无符号数统一起来,并将二进制减法运算变为加法运算,从而给符号数的运算提供了方便;同时也有利于简化运算电路,提高运算速度。

### 1.2.5 二进制的运算

计算机只能进行二进制的算术和逻辑运算,其他复杂的运算和操作都是建立在此基础之上的。

#### 1. 二进制算术运算

二进制的算术运算和十进制的算术运算基本相同,其加、减、乘、除运算规则如下。

##### 1) 二进制加法

加法运算规则:  $0+0=0$ ;  $0+1=1$ ;  $1+0=1$ ;  $1+1=0$ (向前进位)。

##### 2) 二进制减法

减法运算规则:  $0-0=0$ ;  $1-0=1$ ;  $1-1=0$ ;  $0-1=1$ (向前借位)。

##### 3) 二进制乘法

乘法运算规则:  $0 \times 0 = 0$ ;  $1 \times 0 = 0$ ;  $0 \times 1 = 0$ ;  $1 \times 1 = 1$ 。

##### 4) 二进制除法

除法运算规则:  $0 \div 1 = 0$ ;  $1 \div 1 = 1$ ;  $1 \div 0$ ;  $0 \div 0$ (无意义)。

## 2. 二进制逻辑运算

逻辑运算是对逻辑数据进行的运算。由于二进制数具有逻辑属性,因此,可以使用二进制数实现逻辑运算。常用的逻辑运算共有4种,即逻辑“或”运算、逻辑“与”运算、逻辑“非”运算和逻辑“异或”运算。逻辑运算的结果数据也是逻辑型的。其运算特点如下:按位进行运算,不同位之间不发生任何联系,而且每一位的结果要么是“0”,要么是“1”。

### 1) 逻辑与

逻辑与运算也称为逻辑乘法运算,运算符号为“ $\times$ ”、“ $\wedge$ ”或“ $\cdot$ ”。假定有逻辑变量 A 和 B,逻辑与运算的结果为 C,则逻辑与运算可表示为:

$$C = A \times B \quad \text{或} \quad C = A \wedge B \quad \text{或} \quad C = A \cdot B$$

对于 A 和 B 的不同状态组合,逻辑与运算如表 1.1 所示。

表 1.1 逻辑与运算

| A | B | $A \wedge B$ |
|---|---|--------------|
| 0 | 0 | 0            |
| 0 | 1 | 0            |
| 1 | 0 | 0            |
| 1 | 1 | 1            |

从表 1.1 中可以看到,只有两个逻辑变量都为“真”时,逻辑与的运算结果才为“真”。能说明逻辑与关系的最典型实例是照明电路上的串联开关,如图 1.3 所示。只有 A 和 B 两个开关都闭合时,电灯 C 才能点亮。

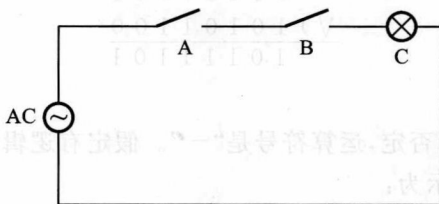


图 1.3 串联开关控制的与关系

逻辑与运算也是二进制数的一种基本运算。例如,二进制数 10100101 和 11001001 的与运算可表示为:

$$\begin{array}{r} 10100101 \\ \wedge) 11001001 \\ \hline 10000001 \end{array}$$

### 2) 逻辑或

逻辑或运算也称为逻辑加法运算,运算符号为“+”或“ $\vee$ ”。假定有逻辑变量 A 和 B,逻辑或运算的结果为 C,则逻辑或运算可表示为:

$$C = A + B \quad \text{或} \quad C = A \vee B$$

对于 A 和 B 的不同状态组合,逻辑或运算如表 1.2 所示。



表 1.2 逻辑或运算

| A | B | $A \vee B$ |
|---|---|------------|
| 0 | 0 | 0          |
| 0 | 1 | 1          |
| 1 | 0 | 1          |
| 1 | 1 | 1          |

从表 1.2 中可以看到,在两个逻辑变量中只要有一个为“真”,则逻辑或运算的结果即为“真”。能说明逻辑或关系的例子很多,例如照明电路上的并联开关,如图 1.4 所示。

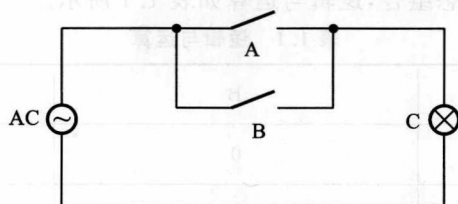


图 1.4 具有逻辑或关系的并联开关

A 和 B 两个开关中只要有一个闭合或两个都闭合,就能使电灯 C 点亮。这种开关并联的关系就是典型的逻辑或关系。

在计算机的数据处理应用中,有时需要使用二进制数的逻辑或运算来实现,因此逻辑或运算也是二进制数的一种基本运算。例如,二进制数 00111001 和 10101100 的或运算可表示为:

$$\begin{array}{r} 00111001 \\ \vee) 10101100 \\ \hline 10111101 \end{array}$$

### 3) 逻辑非

逻辑非运算也称为逻辑否定,运算符号是“ $\neg$ ”。假定有逻辑变量 A,它的逻辑非运算结果为 C,则逻辑非运算可表示为:

$$C = \bar{A}$$

逻辑非运算就是求反运算,因此,逻辑非运算只有两种情况,如表 1.3 所示。

表 1.3 逻辑非运算

| A | $\bar{A}$ |
|---|-----------|
| 0 | 1         |
| 1 | 0         |

### 4) 逻辑异或

逻辑异或运算符号为“ $\oplus$ ”。假定有逻辑变量 A 和 B,其逻辑异或运算结果为 C,则逻辑异或运算可表示为:

$$C = A \oplus B$$

对于 A 和 B 的不同状态组合,逻辑异或运算如表 1.4 所示。



表 1.4 逻辑异或运算

| A | B | $A \oplus B$ |
|---|---|--------------|
| 0 | 0 | 0            |
| 0 | 1 | 1            |
| 1 | 0 | 1            |
| 1 | 1 | 0            |

从表 1.4 中可以看到,逻辑异或运算的特点是,当两个变量的逻辑状态相同时,结果为“假”;当两个变量的逻辑状态不同时,结果为“真”。例如,二进制数 01010011 和 11001010 的“异或”运算可表示为:

$$\begin{array}{r}
 01010011 \\
 \oplus 11001010 \\
 \hline
 10011001
 \end{array}$$

### 1.3 程序设计常用进制数

在使用中,计算机使用的数是二进制数,而在编程中常用十六进制来表示数,人们在平时使用中,数的表示采用的是十进制。因此,需要知道这三种进制之间的转换关系。计算机中使用二进制数,但二进制却给使用者带来许多不便,例如,位数太多,不便书写和阅读等。所以程序设计人员在程序中表示数据时很少直接使用二进制,而是大多使用人们熟悉且与二进制转换方便的其他进制数,常用的包括十进制数和十六进制数等。

使用这些进制数只是为了人们读写方便、直观,没有其他意义。这些数据输入计算机后,一般借助编译软件将它们转换为二进制数。

#### 1.3.1 十进制数与十六进制数

##### 1. 十进制数(decimal)

十进制数是我们最熟悉的数制,是一种最典型的进位计数制。它具有 0~9 共 10 个数码,其基数为 10,逢 10 进位,各数位的权为 10 的幂。一般在十进制数后面加后缀“D”或“d”,以便与其他进制数进行区别。

##### 2. 十六进制数(hexadecimal)

十六进制数有 16 个数码,即 0~9、A、B、C、D、E、F,逢 16 进位,各数位的权为 16 的幂。

在程序中使用不同进制数要注意它们之间的区别,具体做法是:在二进制数后面加标志字符 B(binary),例如,二进制数 10101100,应写为 10101100B;在十六进制数后面加标志字符 H(hex),例如,3AFH 或 0CAH,如果十六进制数以字母开头,应在前面加一个 0,以表明是十六进制数而不是字符组合;而十进制数后面什么也不用加,因为编译软件默认无后缀标识的数按十进制数处理,当然编译软件的默认进制数也是可以重新设定的。