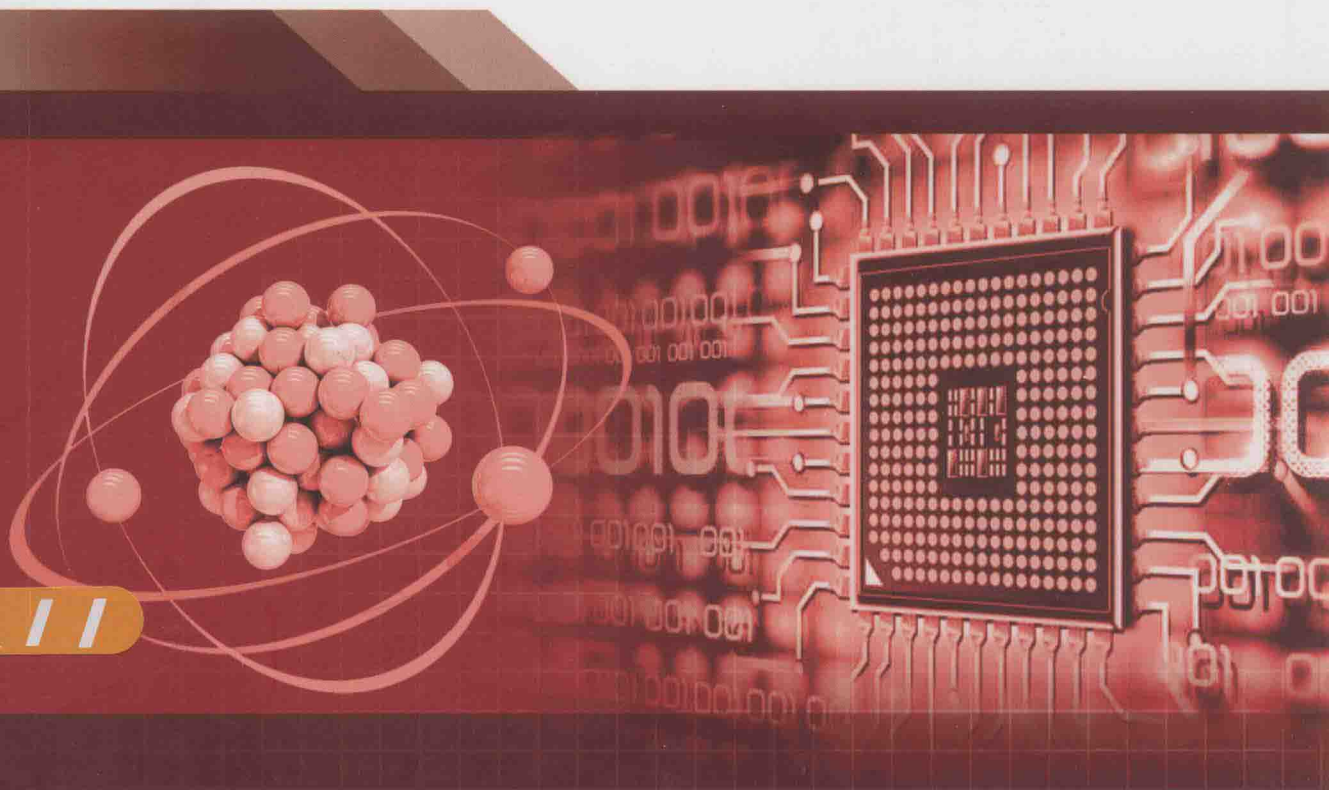


 北京劳动保障职业学院国家骨干校建设资助项目

# 单片机

## 控制技术及应用



李广军 纪娜  主编

 机械工业出版社  
CHINA MACHINE PRESS



北京劳动保障职业学院国家骨干校建设资助项目

# 单片机控制技术的应用

主 编 李广军 纪 娜  
参 编 程晓辉 蒙 娟 宁玉红  
万 捷 孔 杰 赵海静



机械工业出版社

本书以 80C51 单片机为例,讲述了单片机控制技术及应用。内容包括单片机基础知识、结构与原理、指令系统、汇编语言设计、中断系统、定时器/计数器、串行通信、单片机的并行扩展技术及单片机 A-D、D-A 转换接口技术等。

本书可作为高职高专电子、通信、电气、机电专业单片机课程教材,也可供从事单片机应用的工程技术人员参考,非常适合单片机爱好者自学使用。

### 图书在版编目 (CIP) 数据

单片机控制技术及应用/李广军等主编. —北京:机械工业出版社,2013.11  
北京劳动保障职业学院国家骨干校建设资助项目  
ISBN 978-7-111-44921-8

I. ①单… II. ①李… III. ①单片微型计算机-计算机控制-高等职业教育-教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2013) 第 282894 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:罗莉 责任编辑:罗莉

版式设计:常天培 责任校对:陈延翔

封面设计:赵颖喆 责任印制:李洋

北京华正印刷有限公司印刷

2014 年 1 月第 1 版第 1 次印刷

184mm×260mm·12 印张·295 千字

0001—3000 册

标准书号:ISBN 978-7-111-44921-8

定价:39.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066 教材网:<http://www.cmpedu.com>

销售一部:(010)68326294 机工官网:<http://www.cmpbook.com>

销售二部:(010)88379649 机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010)88379203 封面无防伪标均为盗版

# 前 言

本书从社会需要和高等职业教育的特点出发,确定了教学目标:掌握单片机基本结构和原理、MCS-51 系列单片机指令系统、汇编语言程序设计方法、单片机存储系统、输入/输出接口电路、定时器/计数器、中断系统、串行通信技术及模拟量输入/输出接口技术等。

本书以 80C51 单片机为例,讲述了单片机控制技术及应用。内容包括单片机基础知识、结构与原理、指令系统、汇编语言设计、中断系统、定时器/计数器、串行通信、单片机的并行扩展技术及单片机 A-D、D-A 转换接口技术等。

根据高职高专教育注重培养学生实践动手能力的要求,本书以单片机应用为主线,理论与实践紧密结合,注重对单片机资源应用方法的总结,使初学者更容易理解和掌握;每章包含相应的例题,以便使学生更深入学习理论知识,加强实际操作的练习,并安排了一定的习题与思考题,方便读者练习和提高。

本书可作为高职高专电子、通信、电气、机电专业单片机课程教材,也可供从事单片机应用的工程技术人员参考,非常适合单片机爱好者自学使用。

本书由北京劳动保障职业学院机电工程系教师团队编写。李广军编写第 1 章和第 2 章;纪娜编写第 3 章;程晓辉编写第 4 章;蒙娟编写第 5 章;宁玉红编写第 6 章;万捷编写第 7 章;孔杰、赵海静编写第 8 章。全书由李广军统稿。

由于作者水平所限,书中可能会存在某些错误和不妥之处,恳请广大读者批评指正。

作 者

# 目 录

## 前言

## 第1章 计算机运算基础 ..... 1

- 1.1 计算机中的数制及数制的转换 ..... 1
  - 1.1.1 计算机中的数制 ..... 1
  - 1.1.2 数制间的转换 ..... 2
- 1.2 计算机中数的表示方法 ..... 4
  - 1.2.1 定点机中数的表示方法 ..... 4
  - 1.2.2 浮点机中数的表示方法 ..... 5
  - 1.2.3 二进制数的运算 ..... 6
- 1.3 计算机中数的表示形式 ..... 7
  - 1.3.1 机器数的原码、反码和补码 ..... 7
  - 1.3.2 计算机中二进制数的单位表示 ..... 8
  - 1.3.3 计算机使用二进制数的原因 ..... 9
- 1.4 计算机中使用的编码 ..... 9
- 本章小结 ..... 11
- 习题 ..... 11

## 第2章 80C51 的结构及原理 ..... 12

- 2.1 单片机的概述 ..... 12
  - 2.1.1 单片机的定义 ..... 12
  - 2.1.2 单片机的特点 ..... 12
  - 2.1.3 单片机的发展 ..... 13
  - 2.1.4 单片机的应用 ..... 13
- 2.2 80C51 单片机的逻辑结构 ..... 14
  - 2.2.1 80C51 单片机的组成 ..... 14
  - 2.2.2 中央处理单元 ..... 14
  - 2.2.3 存储器 ..... 15
  - 2.2.4 输入/输出接口 ..... 16
- 2.3 80C51 单片机的引脚及功能 ..... 16
  - 2.3.1 80C51 单片机的引脚封装 ..... 16
  - 2.3.2 80C51 单片机引脚及功能 ..... 17
- 2.4 80C51 单片机的存储器组织 ..... 19
  - 2.4.1 80C51 单片机的程序存储器 ..... 19
  - 2.4.2 80C51 单片机的数据存储器 ..... 20
  - 2.4.3 80C51 单片机的特殊功能寄存器 ..... 21

- 2.5 80C51 单片机的并行 I/O 口 ..... 24
  - 2.5.1 P0、P2 口的结构 ..... 24
  - 2.5.2 P1、P3 口的结构 ..... 26
  - 2.5.3 并行口的负载能力 ..... 27
  - 2.5.4 并行口的应用举例 ..... 28
- 2.6 80C51 单片机的时钟与复位 ..... 29
  - 2.6.1 80C51 单片机的时钟 ..... 29
  - 2.6.2 80C51 单片机的定时单位 ..... 30
  - 2.6.3 80C51 单片机的复位方式与初始化状态 ..... 31
  - 2.6.4 80C51 单片机的复位电路 ..... 32
- 本章小结 ..... 33
- 习题 ..... 34

## 第3章 80C51 的指令系统 ..... 35

- 3.1 指令概述 ..... 35
  - 3.1.1 汇编语言 ..... 35
  - 3.1.2 指令格式 ..... 36
  - 3.1.3 常用符号 ..... 36
- 3.2 寻址方式 ..... 37
  - 3.2.1 立即寻址 ..... 37
  - 3.2.2 直接寻址 ..... 38
  - 3.2.3 寄存器寻址 ..... 38
  - 3.2.4 寄存器间接寻址 ..... 39
  - 3.2.5 变址寻址 ..... 39
  - 3.2.6 相对寻址 ..... 40
  - 3.2.7 位寻址 ..... 40
  - 3.2.8 寻址空间 ..... 41
- 3.3 80C51 的指令系统 ..... 42
  - 3.3.1 数据传送指令 ..... 42
  - 3.3.2 算术运算指令 ..... 48
  - 3.3.3 逻辑运算指令 ..... 52
  - 3.3.4 控制转移指令 ..... 55
  - 3.3.5 位操作指令 ..... 61
- 本章小结 ..... 63
- 习题 ..... 68

<b>第4章 汇编语言程序设计</b> .....	71	5.5 定时器/计数器的四种工作方式及其应用 .....	108
4.1 程序设计概述 .....	71	5.5.1 定时器/计数器的初始化 .....	108
4.1.1 程序设计语言 .....	71	5.5.2 方式0 及应用实例 .....	109
4.1.2 汇编语言源程序的编辑与汇编 .....	72	5.5.3 方式1 及应用实例 .....	111
4.1.3 汇编语言程序的基本结构 .....	75	5.5.4 方式2 及应用实例 .....	112
4.1.4 程序设计方法和技巧 .....	75	5.5.5 方式3 .....	115
4.2 顺序程序设计 .....	77	本章小结 .....	116
4.3 循环程序设计 .....	78	习题 .....	116
4.3.1 循环程序设计方法 .....	78	<b>第6章 80C51 单片机的串行通信</b> .....	118
4.3.2 循环程序设计实例 .....	79	6.1 串行通信概述 .....	118
4.4 分支程序设计 .....	84	6.1.1 异步通信和同步通信 .....	118
4.4.1 分支程序设计方法 .....	84	6.1.2 串行通信的数据传送方式 .....	120
4.4.2 分支程序设计实例 .....	85	6.2 80C51 串行通信 .....	121
4.5 子程序设计 .....	88	6.2.1 80C51 单片机串行口结构 .....	121
4.5.1 子程序设计方法 .....	88	6.2.2 80C51 单片机串行口控制机制 .....	122
4.5.2 子程序设计实例 .....	89	6.3 80C51 单片机串行口的工作方式 .....	124
4.6 查表程序设计 .....	90	6.3.1 串行工作方式0 .....	124
4.6.1 查表程序设计方法 .....	90	6.3.2 串行工作方式1 .....	125
4.6.2 查表程序设计实例 .....	90	6.3.3 串行工作方式2 和3 .....	125
本章小结 .....	91	6.4 串行通信数据传输速率 .....	126
习题 .....	91	6.4.1 传输速率的表示方法 .....	126
<b>第5章 80C51 的中断系统及定时器/计数器</b> .....	93	6.4.2 80C51 单片机波特率的设置 .....	126
5.1 80C51 单片机的中断系统 .....	93	6.5 80C51 单片机串行通信的应用 .....	127
5.1.1 80C51 单片机中断的概念 .....	93	6.5.1 单片机串行口扩展并行输入输出应用 .....	127
5.1.2 80C51 单片机的中断源及中断向量 .....	94	6.5.2 单片机与单片机之间的通信 .....	129
5.2 80C51 单片机中断系统的结构及控制 .....	95	本章小结 .....	132
5.2.1 80C51 单片机中断系统的结构 .....	95	习题 .....	133
5.2.2 80C51 单片机中断系统的控制 .....	95	<b>第7章 并行扩展技术</b> .....	134
5.2.3 80C51 单片机中断系统的优先级控制 .....	98	7.1 扩展概述 .....	134
5.3 80C51 单片机中断处理过程 .....	99	7.1.1 单片机并行扩展总线 .....	135
5.3.1 中断响应的条件 .....	99	7.1.2 并行扩展系统的 I/O 编址和芯片选取 .....	137
5.3.2 中断响应的过程 .....	100	7.2 存储器扩展技术 .....	138
5.3.3 中断服务程序 .....	101	7.2.1 存储器的类型 .....	138
5.4 80C51 单片机的定时器/计数器 .....	105	7.2.2 程序存储器扩展 .....	140
5.4.1 定时器/计数器的原理 .....	105	7.2.3 数据存储器扩展 .....	143
5.4.2 定时器/计数器的应用 .....	106	7.3 单片机并行 I/O 接口扩展 .....	145
5.4.3 定时器/计数器的控制 .....	106	7.3.1 单片机并行 I/O 接口扩展基础知识 .....	145

7.3.2 8255 可编程并行接口扩展 .....	146	8.1 模拟量输入接口 (A-D 转换) .....	170
7.3.3 8155 可编程并行接口扩展 .....	150	8.1.1 A-D 转换器概述 .....	170
7.4 键盘接口技术 .....	154	8.1.2 8 位 A-D 转换器芯片及与 80C51 单片机的接口 .....	171
7.4.1 键盘的结构 .....	155	8.1.3 高于 8 位 A-D 转换器芯片及与 80C51 单片机的接口 .....	176
7.4.2 键码和键盘的扫描 .....	157	8.2 模拟量输出接口 (D-A 转换) .....	176
7.4.3 用 8255 实现键盘接口 .....	160	8.2.1 D-A 转换器概述 .....	176
7.5 LED 显示接口技术 .....	162	8.2.2 8 位 D-A 转换器芯片及与 80C51 单片机的接口 .....	177
7.5.1 LED 显示器概述 .....	162	8.2.3 高于 8 位 D-A 转换器芯片及与 80C51 单片机的接口 .....	183
7.5.2 LED 显示器接口 .....	164	本章小结 .....	185
本章小结 .....	168	习题 .....	185
习题 .....	169		
<b>第 8 章 单片机 A-D 及 D-A 转换接口 .....</b>	<b>170</b>		

# 第 1 章 计算机运算基础

## 【学习目的】

1. 了解计算机中的数制。
2. 掌握数制之间的转换。
3. 熟悉计算机中数的表示方法及数的表示形式。

电子计算机是一种能对信息加工处理的机器，具有记忆、判断和运算的能力。

## 1.1 计算机中的数制及数制的转换

十进制是人们生活中普遍使用的计数制，但计算机都是以二进制形式进行算术运算和逻辑运算操作的，微型计算机也不例外。因此，对于用户在键盘上输入的十进制数字和符号命令，微型计算机必须先把它们转换成二进制形式进行识别、运算和处理，然后再把二进制形式的运算结果转换为人们容易识别的十进制数字和符号，并在显示器上显示出来。

上述过程都是由计算机自动完成的，在微型计算机中除了用到二进制和十进制的数制外，经常会用到八进制和十六进制的计数制，为了使读者弄清计算机中数制转换的原理，先对计算机中常用的数制和数制的转换进行讨论。

### 1.1.1 计算机中的数制

所谓数制是指数的制式，是人们利用符号计数的一种科学方法。数制有很多种，微型计算机中常用的数制有十进制、二进制、八进制和十六进制等。

#### 1. 十进制 (Decimal)

十进制是人们生活中普遍使用的数制，它用 0、1、2、…、9 这十个数来描述。十进制数的主要特点如下：

① 它有 0~9 十个不同的数，这是构成所有十进制数的基本符号。

② 它是逢 10 进位的。十进制数在计数过程中，当它的某位计满 10 时就要向它邻近的高位进 1。

#### 2. 二进制 (Binary)

二进制是在计算机系统中使用的数制，它用 0、1 这两个数来描述。二进制数的主要特点如下：

① 任何二进制数都是由 0、1 这两个数组成。

② 二进制数的基数为 2，它遵循逢 2 进 1 的进位计数原则。

#### 3. 十六进制 (Hexadecimal)

十六进制是计算机指令代码和数据以及软件工具中经常使用的数制，它用 0、1、…、9 和 A、B、…、F 这十六个数和字母来描述。十六进制数的主要特点如下：



- ① 任何一个十六进制数都是由 0、1、…、9 和 A、B、…、F 这十六个数和字母构成。
- ② 十六进制数的基数为 16，它遵循逢 16 进 1 的进位计数原则。
- 为方便起见，现将部分十进制、二进制和十六进制数的对照表列于表 1-1。

表 1-1 部分十进制、二进制和十六进制数的对照表

整 数			小 数		
十 进 制	二 进 制	十 六 进 制	十 进 制	二 进 制	十 六 进 制
0	0000	0	0	0	0
1	0001	1	0.5	0.1	0.8
2	0010	2	0.25	0.01	0.4
3	0011	3	0.125	0.001	0.2
4	0100	4	0.0625	0.0001	0.1
5	0101	5	0.03125	0.00001	0.08
6	0110	6	0.015625	0.000001	0.04
7	0111	7			
8	1000	8			
9	1001	9			
10	1010	A			
11	1011	B			
12	1100	C			
13	1101	D			
14	1110	E			
15	1111	F			
16	10000	10			

### 1.1.2 数制间的转换

在计算机中都是以二进制数进行算术运算和逻辑运算操作的，而人们习惯使用十进制数，计算机会自动对不同数制的数进行转换。下面学习不同数制的数是如何转换的。

#### 1. 二进制数和十进制数间的转换

##### (1) 二进制数转换成十进制数

二进制数转换成十进制数只要把欲转换数按权展开后相加即可。例如：

$$11010.01\text{B} = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^{-2} = 26.25$$

##### (2) 十进制数转换成二进制数

本转换过程是上述转换过程的逆过程，但十进制整数和小数转换成二进制整数和小数的方法是不相同的，现分别进行介绍。

##### 1) 十进制整数转换成二进制整数

常用的是“除 2 取余法”。用 2 连续去除要转换的十进制数，直到商小于 2 为止，然后把各次余数按最后得到的为最高位、最早得到的为最低位，依次排列起来所得到的数便是所求的二进制数。

<b>【例 1-1】</b> 100D = _____ B	余数
2 $\overline{) 100}$	0 (最低位)
2 $\overline{) 50}$	0
2 $\overline{) 25}$	1
2 $\overline{) 12}$	0
2 $\overline{) 6}$	0
2 $\overline{) 3}$	1
2 $\overline{) 1}$	1 (最高位)
0	

答案: 100D = 1100100B

## 2) 十进制小数转换成二进制小数

通常采用“乘2取整法”。用2连续去乘要转换的十进制小数，直到所得积的小数部分为0或满足所需精度为止，然后把各次整数按最先得到的为最高位、最后得到的为最低位，依次排列起来所对应的数便是所求的二进制小数，现结合实例加以介绍。

<b>【例 1-2】</b> 0.625D = _____ B	
乘2取整	整数部分
0.625	
× $\overline{) 2}$	
1.250	1
0.25	
× $\overline{) 2}$	
0.50	0
× $\overline{) 2}$	
1.0	1

答案: 0.625D = 0.101B

## 2. 二进制数与十六进制数的转换

### 1) 二进制数转换为十六进制数

采用四位二进制数合成为一位十六进制数的方法，以小数点为界分成左侧整数部分和右侧小数部分；整数部分从小数点开始，向左每4位二进制数一组，不足4位在数的前面补0；小数部分从小数点开始，向右每4位二进制数一组，不足4位在数的后面补0，然后每组用十六进制数码表示，并按序相连即可。

**【例 1-3】** 把 111010.011110B 转换为十六进制数。

0011 1010. 0111 1000 = 3A.78H  
 3 A 7 8

### 2) 十六进制数转换为二进制数

将十六进制数的每位分别用4位二进制数码表示，然后它们按序连在一起即为对应的二进制数。

**【例 1-4】** 把 2BD4H 和 20.5H 转换为二进制数

2BD4H = 0010 1011 1101 0100B

20.5H = 0010 0000.0101B

### 3. 十六进制数与十进制数的转换

#### (1) 十六进制数转换成十进制数

十六进制数转换成十进制数的方法和二进制数转换成十进制数的方法类似，将十六进制数按权展开后求和即得到十进制数。

**【例 1-5】** 将十六进制数 3DF2H 转换成十进制数。

$3DF2H = 3 \times 16^3 + 13 \times 16^2 + 15 \times 16^1 + 2 \times 16^0 = 15858$

#### (2) 十进制数转换成十六进制数

① 十进制整数转换成十六进制整数与十进制整数转换成二进制整数类似，十进制整数转换成十六进制整数可以采用“除 16 取余法”。用 16 连续去除要转换的十进制整数，直到商数小于 16 为止，然后把各次余数按逆得到顺序排列起来所得的数，便是所求的十六进制数。

② 十进制小数转换成十六进制小数的方法类似十进制小数转换成二进制小数，常采用“乘 16 取整法”。把欲转换的十进制小数连续乘以 16，直到所得乘积的小数部分为 0 或达到所需精度为止，然后把各次整数按相同的得到顺序排列起来所得的数，便是所求的十六进制小数。

## 1.2 计算机中数的表示方法

在讨论计算机如何对有符号数或无符号数进行运算和处理之前，先弄清计算机中数的表示方法是十分必要的。在计算机中，小数和整数都是以二进制形式表示的，但对小数点通常有定点和浮点两种表示方法。小数点采用定点表示法的称为定点机，采用浮点表示法的叫做浮点机。

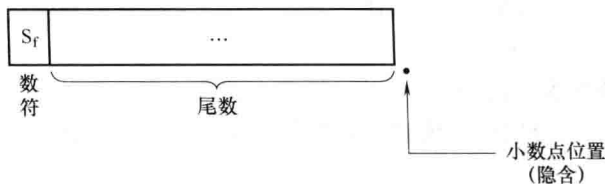
### 1.2.1 定点机中数的表示方法

在定点计算机中，二进制数的小数点位置是固定不变的，小数点位置可以固定在数值位之前，也可以固定在数值位之后。前者称为定点小数计算机，后者叫做定点整数计算机。

在理论和习惯上，小数点固定在中间位置比较合适，因为它所能表示的数既有整数又有小数部分，会给数在数制间替换带来麻烦，故这种方法通常并不为计算机设计师们所采用。

#### 1. 定点整数表示法

在采用定点整数表示法的计算机中，小数点位置被固定在数值位之后。因此，这种计算机在实际运算前应先把参加运算的数（二进制形式）按适当比例替换成纯整数，并在运算后把结果操作数按同一比例还原后输出。设  $N$  为某一定点二进制整数，其表示形式为



其中,  $S_f$  为数符,  $S_f=0$  表示  $N$  为正数,  $S_f=1$  表示  $N$  为负数。

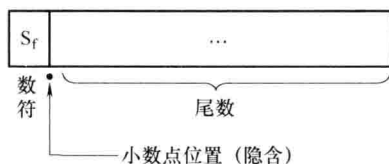
数的表示形式在大多数计算机中都是采用定点整数法, MCS-51 单片机也是一种定点整数计算机。因此, MCS-51 单片机只能对二进制整数进行直接运算和处理, 它在遇到二进制小数时, 必须把该小数按比例扩大成二进制整数后进行处理, 并在处理完后再按同样比例缩小后进行输出。

定点整数表示法的优点是, 运算规则简单, 它所能表示的数的范围没有相同位数的浮点法大。例如, 一个 16 位的二进制定点整数  $N$ , 若它的  $S_f$  占 1 位、尾数占 15 位, 则它所能表示的原码数的范围为

$$|N| \leq 11 \cdots 11 = 100 \cdots 00 - 1 = 2^{15} - 1, \text{ 近似形式为 } -2^{15} \leq N \leq 2^{15}$$

## 2. 定点小数表示法

在采用定点小数表示法的计算机中, 小数点的位置被固定在数值位之前。因此, 这种计算机在实际运算前应首先把参加运算的二进制整数按适当比例替换成纯小数, 并在运算结束后把结果操作数 (纯小数) 按同样比例逆替换后输出。设  $N$  为定点小数, 其表示形式为



其中,  $S_f$  为数符,  $S_f=0$  表示  $N$  为正数,  $S_f=1$  表示  $N$  为负数。

定点小数表示法的优点是, 运算规则简单, 但它所能表示的数的范围较小。例如, 一个 16 位的二进制小数  $N$ , 若它的  $S_f$  占 1 位、尾数占 15 位, 则它所能表示的原码数范围为

$$|N| \leq 0.11 \cdots 11 = 1 - 0.00 \cdots 01 = 1 - 2^{-15}$$

$$\text{即 } -(1 - 2^{-15}) \leq N \leq (1 - 2^{-15})$$

### 1.2.2 浮点机中数的表示方法

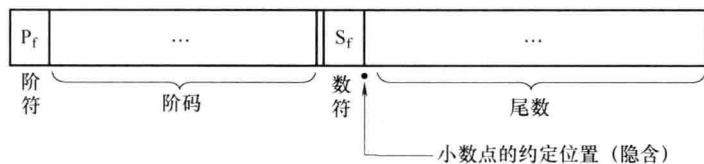
在采用浮点表示的二进制数中, 小数点位置是浮动的、不固定的。

通常, 任意一个二进制数都可以写为

$$N = 2^P \times S$$

式中,  $S$  为二进制数  $N$  的尾数, 代表了  $N$  的实际有效值;  $P$  为  $N$  的阶码, 可以决定小数点的具体位置。例如:  $N = 101.11\text{B} = 2^3 \times 0.10111\text{B}$ 。

因此, 任何一个浮点数  $N$  都由阶码和尾数两部分组成。阶码部分包括阶符和阶码, 尾数部分由数符和尾数组成, 其形式为



其中,  $P_f$  为阶符,  $P_f=0$  表示阶码为正,  $P_f=1$  表示阶码为负;  $S_f$  为数符,  $S_f=0$  表示该数为正数,  $S_f=1$  表示该数为负数; 小数点的约定位置在尾数之前, 实际位置是浮动的, 由阶码

决定。浮点法的优点是，数的表示范围大。浮点表示法的缺点是，运算规则复杂，通常要对阶码和尾数分别运算。

### 1.2.3 二进制数的运算

二进制数的运算分为二进制整数运算和二进制小数运算两种类型，但运算法则完全相同。由于大部分计算机中数的表示方法均采用定点整数表示法，故这里仅介绍二进制整数运算，二进制小数运算与它相同。

二进制数的运算比较简单，包括算术运算和逻辑运算。算术运算包括加、减、乘、除运算；逻辑运算包括逻辑乘、逻辑加、逻辑非和逻辑异或等。

#### 1. 二进制数的算数运算

##### (1) 加法运算

运算规则为

$$0+0=0; 0+1=1+0=1; 1+1=10 \text{ (向高位进位)}$$

##### (2) 减法运算

运算规则

$$0-0=0; 1-0=1; 1-1=0; 0-1=1 \text{ (向高位借1)}$$

##### (3) 乘法运算

运算规则

$$0 \times 0 = 0; 0 \times 1 = 1 \times 0 = 0; 1 \times 1 = 1$$

##### (4) 除法运算

除法是乘法的逆运算。与十进制类似，二进制除法也是从被除数最高位开始，查找出够减除数的位数，并在其最高位处上商 1 和完成它对除数的减法运算，然后把被除数的下一位移到余数位置上。若余数不够减除数，则上商 0，并把被除数的再下一位移到余数位置上；若余数够减除数，则上商 1 并进行余数减除数。这样反复进行，直到全部被除数的各位都下移到余数位置上为止。

#### 2. 逻辑运算

计算机处理数据时常常要用到逻辑运算。逻辑运算由专门的逻辑电路完成。下面介绍几种常用的逻辑运算。

##### (1) 逻辑乘运算

逻辑乘又称逻辑与，常用  $\wedge$  算符表示。逻辑乘运算法则为

$$0 \wedge 0 = 0; 1 \wedge 0 = 0 \wedge 1 = 0; 1 \wedge 1 = 1$$

##### (2) 逻辑加运算

逻辑加又称逻辑或，常用算符  $\vee$  表示。逻辑加的运算规则为

$$0 \vee 0 = 0; 1 \vee 0 = 0 \vee 1 = 1; 1 \vee 1 = 1$$

##### (3) 逻辑非运算

逻辑非运算又称逻辑取反，常采用 “ $\bar{\quad}$ ” 运算符表示。逻辑非的运算规则为

$$\bar{0} = 1; \bar{1} = 0$$

##### (4) 逻辑异或

逻辑异或又称为半加，是不考虑进位的加法，常采用算符表示。逻辑异或的运算规则为

$$0 \oplus 0 = 0; 1 \oplus 1 = 0; 1 \oplus 0 = 0 \oplus 1 = 1$$

### 1.3 计算机中数的表示形式

在现代微型机中，其内部运算器通常由一个补码加法器、n 位寄存器/计数器组和移位控制电路等组成，但它恰能进行各种算术运算和逻辑操作。这就是说，补码加法器既能做加法又能将减法运算变为加法来做，从而大大简化了运算器内部的电路设计。这应归功于人们长期以来对计算机中码制的研究。

机器数是指数的符号和值均采用二进制的表示形式。因此，机器数在定点和浮点机中的表示形式各不相同。为了方便起见，这里的机器数均指在定点整数机中的表示形式。即最高位是符号位（0 表示正数，1 表示负数），其余位为数值位，小数点约定在数值位之后。在计算机中，机器数有原码、反码、补码、变形原码、变形反码、变形补码和移码等多种形式。

#### 1.3.1 机器数的原码、反码和补码

原码、反码和补码是机器数的三种基本形式，它和机器数的真值不同。机器数的真值定义为采用 + 和 - 表示的二进制数符号，并非真正的机器数。例如，+76 的机器数真值为 +1001100B，原码形式为 01001100B（最高位的 0 表示正数）；-76 的真值为 -1001100B，原码为 11001100B（最高位的 1 表示负数）。

##### 1. 原码

原码表示法是机器数的一种简单的表示法。这种表示法数的最高位为符号位（用 0 或 1 来表示），其余位为数值位，符号位的 0 表示该数为正数，符号位为 1 表示它是负数。通常，一个数的原码可以先把该数用方括号括起来，并在方括号右下角加个“原”字来标记。设有一数为 x，则原码表示可记作  $[x]_{\text{原}}$ 。

**【例 1-6】** 设  $X1 = +1100B$ ， $X2 = -1100B$ ，请分别写出它们在 8 位微机中的原码。

**解：** 其原码记为

$$[X1]_{\text{原}} = [00001100]_{\text{原}}$$

$$[X2]_{\text{原}} = [10001100]_{\text{原}}$$

原码表示数的范围与二进制位数有关。当用 8 位二进制来表示小数原码时，其表示范围为

最大值为 0.1111111，其真值约为 (0.99) D

最小值为 1.1111111，其真值约为 (-0.99) D

当用 8 位二进制来表示整数原码时，其表示范围为

最大值为 01111111，其真值为 (127) D

最小值为 11111111，其真值为 (-127) D

在原码表示法中，对 0 有两种表示形式：

$$[+0]_{\text{原}} = 00000000$$

$$[-0]_{\text{原}} = 10000000$$

## 2. 反码

在微型计算机中，二进制数的反码可由原码得到，有正数的反码和负数的反码之分：正数的反码和原码相同；负数反码的符号位和负数原码的符号位相同，数值位是它的数值位的按位取反。反码的标记方法和原码类似，只要在被括数方括号的右下角添加一个“反”字即可。设有一数为  $x$ ，则反码表示可记作  $[x]_{\text{反}}$ 。

**【例 1-7】** 设  $X1 = +1010110$ ， $X2 = -1001010$ ，请分别写出它们在 8 位微机的原码和反码。

**解：**由于正数的反码就是其原码；负数的反码是符号位不变，数值位是它的数值位的按位取反，所以有

$$[X1]_{\text{原}} = 01010110$$

$$[X1]_{\text{反}} = [X1]_{\text{原}} = 01010110$$

$$[X2]_{\text{原}} = 11001010$$

$$[X2]_{\text{反}} = 10110101$$

反码通常作为求补过程的中间形式，即在一个负数的反码的末位上加 1，就得到了该负数的补码。

## 3. 补码

补码的概念是在计算机中经常会遇到的，二进制数的补码可由反码得到，如果是正数，则该机器数的补码与原码一样；如果是负数，则该机器数的补码是对它的原码（除符号位外）各位取反，并在末位加 1 而得到的。补码的标记方法和原码、反码类似，只要在被括数方括号的右下角添加一个“补”字即可。设有一数  $X$ ，则  $X$  的补码表示记作  $[X]_{\text{补}}$ 。

**【例 1-8】** 设  $X1 = +1010B$ ， $X2 = -01010B$ ，试分别写出它们在 8 位微机中的原码、反码和补码形式。

**解：**由原码、反码和补码的定义得

$$[X1]_{\text{原}} = 00001010B \quad [X2]_{\text{原}} = 10001010B$$

$$[X1]_{\text{反}} = 00001010B \quad [X2]_{\text{反}} = 11110101B$$

$$[X1]_{\text{补}} = 00001010B \quad [X2]_{\text{补}} = 11110110B$$

补码的优点是可以将减法运算转换为加法运算，其符号位可以连同数值位一起运算。这样非常有利于计算机的实现。

**【例 1-9】**  $45H$ ， $-55H$ ，用补码运算的方法求两数之和。

$$\text{解：} [45H]_{\text{补}} - [-55H]_{\text{补}} = [-10H]_{\text{补}}$$

### 1.3.2 计算机中二进制数的单位表示

在计算机中使用的二进制数共有 3 个单位，从小到大依次为位、字节和字。

#### 1. 位 (bit)

这里所说的位是指二进制数的位。位是数的最小单位，bit 是位的英文名称，读作“比特”。

在计算机中位仅有 0 和 1 两个数值，表示两种状态。

#### 2. 字节 (Byte)

8 位二进制数称为一个字节，其英文名称是 Byte，在使用时常用大写字母 B 表示。字节

是最基本的数据单位，计算机中的数据、代码、指令、地址多以字节为单位。

### 3. 字 (Word)

字是一台计算机上所能并行处理的二进制数，字的位数（或长度）称之为字长。字长必须是字节的整数倍。例如，MCS-51 单片机字长为 8 位，MCS-96 单片机字长为 16 位，还有 32 位、64 位字长的计算机。

## 1.3.3 计算机使用二进制数的原因

为什么在计算机中要使用二进制数呢？其原因主要有以下几点：

### 1. 易于实现

在计算机中，数是用不同的物理状态来表示的。因为二进制数只有两个数字 0 和 1，用两种物理状态就可以表示出来。而两种相反的物理状态在技术上极易实现。例如，开关的接通与断开，晶体管的导通与截止，电平的高低，脉冲的有无等。

对于这样两种截然相反的物理状态，不但易于实现，而且状态稳定可靠。而两种以上的物理状态，不但难以实现，而且稳定性也差。

### 2. 运算简单

因为二进制数只有两个数字，所以对二进制数的运算比人们熟悉的十进制数的运算要简单得多，而运算简单将有利于简化计算机的电路结构。

### 3. 具有逻辑属性

由于二进制数的 0 和 1 正好与逻辑值的“假 (F)”和“真 (T)”相对应，因此可以使用二进制数实现逻辑运算，从而使逻辑代数运算成为可能。

### 4. 可靠性高

由于二进制数用两种截然相反的物理状态表示，十分稳定。因此二进制数的处理、存储和传送都最为可靠。

## 1.4 计算机中使用的编码

计算机中除了使用数以外，还使用编码。可以把编码分为两类：一类是数的编码，另一类是文字符号的编码。因为是在计算机中使用，所以编码必须是二进制数。

### 1. 二-十进制编码

在计算机中最常用的是用二进制数给十进制数编码，即通常所说的二-十进制编码。若要给一位十进制数编码，则须用 4 位二进制数。在二-十进制编码中最常用的是 BCD 码。

BCD 码共有 10 个编码，即二进制数 0000 ~ 1001，分别对应十进制数 0 ~ 9。例如，十进制数 3 的 BCD 码是 0011；9 的 BCD 码是 1001；39 的 BCD 码是把 3 和 9 的 BCD 码连在一起，即 00111001，正好为 1 个字节。BCD 码的特点是，4 位之内为二进制关系，每 4 位之间为十进制关系。

定义 BCD 码是为了便于在计算机中使用人们最熟悉的十进制数，特别是在输入与输出操作中。例如，从键盘输入的十进制数到计算机中就变为 BCD 码形式，当然这需要有相应的转换程序。有了十进制数的输入和输出，在计算机中就会存在十进制数的存储和计算。但十进制数计算存在调整问题，即所谓的十进制调整，以解决 BCD 码运算时因进位和借位产



生的偏差。

## 2. ASCII 码

文字符号代码用于在计算机中表示西文字符、汉字以及各种符号，最常用的文字符号代码是 ASCII 码和汉字国标码。这里只介绍 ASCII 码。

ASCII 代表的是“美国信息交换标准代码”。它原是美国的字符代码标准，于 1968 年发表，由于使用广泛，早已被国际标准化组织确定为国际标准，成为计算机领域中最重要代码。

ASCII 码表见表 1-2。

表 1-2 ASCII 码表

$b_6b_5b_4$ $b_3b_2b_1b_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	.	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	.	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	
1100	FF	FS	,	<	L	\	l	l
1101	CR	GS	-	=	M	]	m	
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

ASCII 码中字符和功能符号共计 128 个：其中字符 94 个，包括十进制数字 10 个，英文小写字母 26 个，英文大写字母 26 个，标点符号及专用符号 32 个，功能符 34 个（字符区首尾两个符号 SP 和 DEL 一般归入功能符）。由于  $2^7 = 128$ ，因此 128 个字符和功能符使用 7 位二进制数就可以进行编码，此编码即为 ASCII 码。

ASCII 码表是一个 16 行 × 8 列的矩阵，其中行为编码中的后 4 位二进制数 ( $b_3b_2b_1b_0$ )，列为编码中的前 3 位二进制数 ( $b_6b_5b_4$ )，合在一起为 7 位二进制编码。例如，字符 A 的编码为 1000001。

为了方便，常用十进制数或十六进制数来表示 ASCII 码。例如，字符 A 的 ASCII 码用十进制数表示为 65，用十六进制数表示为 41H。

7 位 ASCII 码结构是基本 ASCII 码，由于在计算机中常用字节（8 位）来表示数据。因此，为凑成一个字节，应在 ASCII 码的最高位补 1 个 0。