



单片机原理及应用

基于Keil和Proteus的仿真技术



马继伟 伦翠芬 杨 英◎主编



燕山大学出版社
YANSHAN UNIVERSITY PRESS

单片机原理及应用

——基于 Keil 和 Proteus 的仿真技术

马继伟 伦翠芬 杨 英 主编

 燕山大学出版社

2020 · 秦皇岛

图书在版编目(CIP)数据

单片机原理及应用:基于 Keil 和 Proteus 的仿真技术/马继伟,伦翠芬,杨英主编. —秦皇岛:燕山大学出版社,2020.12

ISBN 978-7-81142-189-7

I. ①单… II. ①马…②伦…③杨… III. ①单片微型计算机 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2020)第 137319 号

单片机原理及应用——基于 Keil 和 Proteus 的仿真技术

马继伟 伦翠芬 杨 英 主编

出版人:陈 玉

责任编辑:孙志强

封面设计:刘韦希

出版发行: 燕山大学出版社
YANSHAN UNIVERSITY PRESS

地 址:河北省秦皇岛市河北大街西段 438 号

邮政编码:066004

电 话:0335-8387555

印 刷:中国标准出版社秦皇岛印刷厂

经 销:全国新华书店

开 本:787 mm×1092 mm 1/16 印 张:20.25 字 数:435 千字

版 次:2020 年 12 月第 1 版 印 次:2020 年 12 月第 1 次印刷

书 号:ISBN 978-7-81142-189-7

定 价:59.00 元

版权所有 侵权必究

如发生印刷、装订质量问题,读者可与出版社联系调换

联系电话:0335-8387718

前 言



STC 系列单片机相比传统的 8051 单片机具有强大的优势,它在我国应用市场占有较大的份额。STC 单片机的在线编程功能以及分系列的资源配置,增加了单片机型号的选择性,应用开发可根据单片机应用系统的功能要求选择合适的单片机,从而降低单片机应用系统的开发难度和开发成本,提高了单片机应用系统的性价比。本教材将单片机技术的教学、单片机技术的发展和工程应用进行了无缝对接。针对 51 单片机基本架构、指令系统、开发环境进行了系统介绍;并且针对单片机作为控制核心,对单片机最小测控系统的开发设计,以及外围的功率接口部分都作了详细介绍。

STC 系列单片机的指令系统和标准的 8051 内核完全兼容,而且 STC 系列单片机已发展了 STC89/90 系列、STC10/11 系列、STC12 系列、STC15 系列、STC8 系列,本教材基于 STC89C51 详细介绍了 80C51 的硬件结构、指令系统与应用编程。

本教材根据作者多年单片机教学、电子大赛指导以及开发应用实践,根据学生的学习规律和教学经验总结整理而成。全书内容力求应用性,降低学习难度,以提高学生的工程设计能力与实践动手能力为目标。全书共 12 章,包括微型计算机基础、80C51 单片机的内核、单片机应用的开发工具、80C51 单片机指令系统、80C51 单片机的程序设计、80C51 单片机的中断系统与定时器、80C51 单片机并行人机接口技术、80C51 单片机的串行通信、单片机 A/D 与 D/A 转换接口、功率接口技术及应用、单片机串行接口总线技术及应用、单片机应用系统设计等。具有以下特点:

(1) 采用“双”语编程教学。在加强单片机的硬件结构和工作原理的理解学习环节利用汇编指令系统深入浅出地加以引导,在程序的功能模块化以及可读性方面辅以 C51 项目案例,突出 C51 编程、调试训练。

(2) 理论联系实际。在指令系统学习之前先介绍了单片机的应用开发工具,详细介绍了单片机开发程序的编辑、编译、调试、下载的完整流程。强调单片机的应用性与实践性,并将这一思想贯穿于案例项目教学。

(3) 采用项目教学。针对这门课程实践性强的特点,引入项目教学,将具体的应用案例作为项目教学包引入课堂和实验教学环节,利用开发软件建立单片机实验环节。在教材的第 6~12 章都引入了各种接口的项目教学包。

(4) 强化单片机应用的通信和控制功能。第 8 章介绍了单片机的串行通信,

在第 10 章介绍了单片机的功率驱动接口,第 11 章介绍了单片机的常用串行接口,第 12 章介绍了单片机应用系统开发注意事项。这些内容的教学训练将会为单片机测控工程师的培养训练打下坚实基础。

(5) 本书所有示例都有详细说明,并在 Proteus 软件中进行了仿真实验。各章之间既相互关联,又独立成篇。

本教材力求实用性、应用性,降低学习难度,提高学生的工程设计能力与实践动手能力。

目 录



第 1 章 微型计算机基础	1
1.1 数制与编码	1
1.1.1 数制及不同数制之间的转换	1
1.1.2 微型计算机中数的表示方法	4
1.1.3 微型计算机中常用编码	5
1.2 微型计算机的基本组成	7
1.3 指令、程序、编程语言	8
1.4 微型计算机的工作过程	9
1.5 微型计算机的应用形态	11
本章小结	11
习题 1	12
第 2 章 80C51 单片机的内核	13
2.1 单片机概述	13
2.1.1 单片机的概念	13
2.1.2 常见单片机	13
2.2 80C51 单片机资源概述与引脚功能	15
2.2.1 80C51 单片机资源与功能概述	15
2.2.2 80C51 单片机的封装及信号引脚	15
2.3 80C51 单片机的内部结构	17
2.3.1 80C51 单片机的内部结构	17
2.3.2 CPU 结构	18
2.4 80C51 单片机的存储结构	20
2.5 80C51 单片机的并行 I/O 口	24
2.5.1 80C51 单片机并行 I/O 口的结构及操作	24
2.5.2 80C51 单片机并行 I/O 口的使用注意事项	27
2.6 80C51 单片机的时钟与复位	29

■ 单片机原理及应用——基于 Keil 和 Proteus 的仿真技术

2.6.1 时钟电路	29
2.6.2 定时单位	29
2.6.3 复位方式与初始化状态	30
本章小结	31
习题 2	31
第 3 章 单片机应用的开发工具	33
3.1 Keil μ Vision4 集成开发环境	33
3.1.1 Keil μ Vision4 集成开发环境概述	33
3.1.2 Keil C 集成开发环境下的程序编辑、编译与调试	34
3.2 STC 系列单片机的在线编程(ISP)	42
3.2.1 STC 系列单片机在线编程(ISP)典型应用电路	42
3.2.2 STC 系列单片机 PC 端下载软件的使用	43
3.3 Proteus 模拟仿真软件	44
3.3.1 Proteus 模拟仿真软件概述	44
3.3.2 Proteus 操作界面简介	45
3.3.3 仿真电路原理图设计	47
3.3.4 Proteus 仿真调试	49
本章小结	50
习题 3	50
第 4 章 80C51 单片机指令系统	52
4.1 概述	52
4.2 数据传送类指令	57
4.3 算术运算类指令	64
4.4 逻辑运算与循环移位类指令	69
4.5 控制转移类指令	72
4.6 位操作类指令(17 条)	79
4.7 伪指令(8 条)	82
本章小结	85
习题 4	85
第 5 章 80C51 单片机的程序设计	90
5.1 汇编语言程序设计	90

5.1.1	程序编制的方法和技巧	90
5.1.2	程序的模块化设计	91
5.2	基本程序设计结构与程序设计举例	92
5.3	C51 程序设计	98
5.3.1	C51 基础	99
5.3.2	C51 程序设计	105
	本章小结	112
	习题 5	112
第 6 章	80C51 单片机的中断系统与定时器	115
6.1	中断系统概述	115
6.1.1	中断系统的几个概念	115
6.1.2	中断的技术优势	116
6.1.3	中断系统需要解决的问题	116
6.2	80C51 单片机的中断系统	117
6.2.1	中断源与中断向量	117
6.2.2	80C51 单片机的中断响应	120
6.2.3	80C51 中断应用举例	122
6.3	80C51 单片机外部中断的扩展	124
6.4	80C51 单片机的定时器/计数器	126
6.4.1	80C51 单片机的定时器/计数器的结构和工作原理	127
6.4.2	80C51 单片机的定时器/计数器的控制	128
6.4.3	80C51 单片机的定时器/计数器的工作方式	129
6.4.4	80C51 单片机的定时器/计数器的应用举例	136
	本章小结	141
	习题 6	142
第 7 章	80C51 单片机并行人机接口技术	144
7.1	LED 数码管显示接口与应用实例	144
7.1.1	数码管静态显示接口与应用	145
7.1.2	数码管动态显示接口与应用	148
7.2	LCD 显示接口与应用实例	151
7.2.1	LCD 显示器概述	151
7.2.2	字符型 LCD1602 与应用实例	153

7.2.3 常用 12864 液晶操作实例	160
7.3 键盘接口与应用实例	168
7.3.1 独立按键及应用	169
7.3.2 行列矩阵键盘结构及工作方式	175
7.3.3 行列矩阵键盘应用举例	181
本章小结	183
习题 7	184
第 8 章 80C51 单片机的串行通信	185
8.1 串行通信基础	185
8.2 80C51 单片机串行口	188
8.2.1 80C51 串行口的硬件结构	188
8.2.2 80C51 串行口的控制	189
8.2.3 80C51 串行口工作方式	190
8.2.4 串行口的波特率	194
8.3 串行口的应用举例	195
8.3.1 方式 0 的编程和应用	195
8.3.2 方式 1 的编程和应用	197
8.3.3 多机通信	202
8.4 80C51 单片机与 PC 机的通信	205
8.4.1 单片机与 PC 机 RS-232C 串行通信的接口设计	205
8.4.2 单片机与 PC 机 USB 总线通信的接口设计	206
8.4.3 串行口打印在调试程序中的应用	207
本章小结	210
习题 8	210
第 9 章 单片机 A/D 与 D/A 转换接口	212
9.1 单片机测控系统与模拟输入通道	212
9.1.1 单片机测控系统概述	212
9.1.2 模拟输入通道	212
9.2 A/D 转换器的主要性能指标及分类	215
9.2.1 A/D 转换器的主要性能指标	215
9.2.2 A/D 转换器的分类及应用说明	216
9.3 常用 A/D 转换芯片及接口技术	217

9.3.1	ADC0809 及接口	217
9.3.2	AD574 及接口	223
9.4	常用 D/A 转换芯片及接口技术	224
9.4.1	D/A 转换器的主要性能指标	224
9.4.2	DAC0832 D/A 转换器及接口	225
9.4.3	DAC0832 连接方式	226
9.5	A/D 与 D/A 转换芯片的串行接口	230
9.5.1	通过 I ² C 总线的串行接口	230
9.5.2	I ² C 接口 A/D 与 D/A 转换器 PCF8591 接口及其应用	231
	本章小结	237
	习题 9	237
第 10 章	功率接口技术及应用	239
10.1	开关型接口	239
10.1.1	简单开关量输出接口	239
10.1.2	光电耦合接口	240
10.1.3	继电器接口	242
10.2	电压调节接口	245
10.2.1	晶闸管工作原理	245
10.2.2	晶闸管在电压调解中的应用	247
10.2.3	固态继电器	248
10.3	电机控制与应用实例	249
10.3.1	直流电机的控制	249
10.3.2	步进电机的控制	252
10.3.3	步进电机与单片机的接口	254
	本章小结	257
	习题 10	258
第 11 章	单片机串行接口总线技术及应用	259
11.1	单片机串行接口扩展概述	259
11.1.1	单片机需要串行接口扩展的原因	259
11.1.2	单片机串行接口扩展实现方法	259
11.2	单总线数字温度传感器 DS18B20 与应用实例	260
11.2.1	DS18B20 性能特点	260

11.2.2 DS18B20 温度传感器程序设计	261
11.3 基于 DS1302 万年历的应用实例	270
11.3.1 DS1302 基本知识	270
11.3.2 DS1302 的控制命令字	271
11.3.3 DS1302 的寄存器	271
11.3.4 DS1302 数据的传送	273
11.3.5 DS1302 驱动程序源代码	273
11.4 I ² C 串行总线原理	277
11.4.1 I ² C 总线概述	277
11.4.2 E ² PROM AT24C02 与单片机的通信实例	280
本章小结	285
习题 11	286
第 12 章 单片机应用系统设计	287
12.1 单片机应用系统的开发流程	287
12.1.1 单片机应用系统的设计原则	287
12.1.2 单片机应用系统的开发流程	287
12.2 硬件设计要求	292
12.3 软件设计要求	293
12.4 可靠性设计要求	294
本章小结	296
习题 12	296
附 录	297
附录 1 ASCII 表	297
附录 2 80C51 单片机汇编指令系统表	298
附录 3 C 语言运算符和结合性	303
附录 4 Proteus 提供的所有元件分类及子类	304
附录 5 STC-ISP 下载编程软件及功能工具简介	310
参考文献	314

第 1 章 微型计算机基础

1.1 数制与编码

数制与编码是微型计算机的基本数字逻辑基础,是学习微型计算机的必备知识。数制与编码的知识在数字逻辑或信息技术基础课程中已经学过,为了巩固这部分知识,在微机原理部分能够灵活熟练应用,在这里再简要理一理。

1.1.1 数制及不同数制之间的转换

所谓数制,即计数的方法,通常采用进位计数制。在微型计算机的学习与应用中,主要用到十进制、二进制、十六进制三种计数方法。十进制是日常采用的数制;二进制数是计算机工作的基础,计算机只能使用二进制数,但是为了读写记忆微型计算机的地址和程序代码、运算数字,一般都采用十六进制。

1. 常用进位计数制及表示方法

二进制、十进制、十六进制的计数规则与表示方法如表 1.1 所示。

表 1.1 二进制、十进制、十六进制的计数规则与表示方法

进位制	计数规则	基数	各位的权	数码	权值展开式	表示法	
						后缀字符	下标
二进制	逢二进一	2	2^i	0,1	$(b_{n-1} \cdots b_1 b_0 \cdot b_{-m})_2 = \sum_{i=-m}^{n-1} b_i \times 2^i$	B	$()_2$
十进制	逢十进一	10	10^i	0,1,⋯, 9	$(d_{n-1} \cdots d_1 d_0 \cdot d_{-m})_{10} = \sum_{i=-m}^{n-1} d_i \times 10^i$	D	$()_{10}$
						通常省略	
十六进制	逢十六进一	16	16^i	0,1,⋯, 9,A,⋯, F	$(h_{n-1} \cdots h_1 h_0 \cdot h_{-m})_{16} = \sum_{i=-m}^{n-1} h_i \times 16^i$	H	$()_{16}$

2. 数制之间的转换

任意进制之间相互转换,整数部分和小数部分的转换规则不同。各进制的相互转换关系如图 1.1 所示。

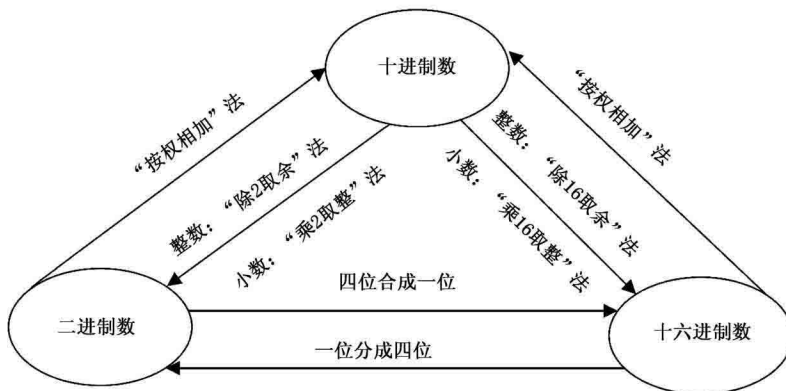
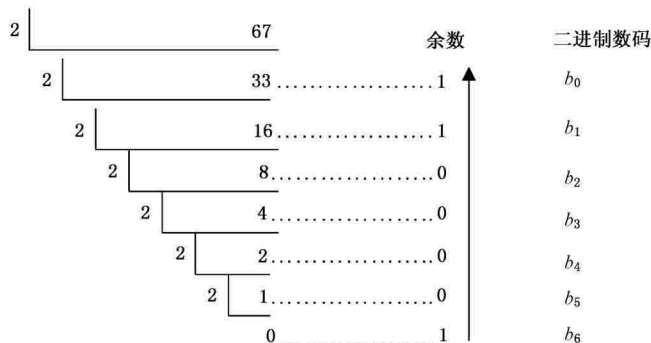


图 1.1 各进制的相互转换关系图

(1) 二进制、十六进制转换为十进制。将二进制、十六进制数按权值展开式展开求和，即为十进制数。

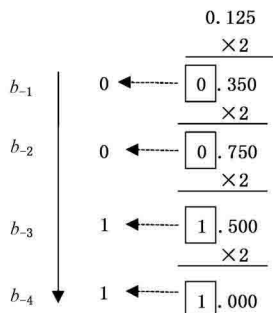
(2) 十进制转换成二进制。将十进制数分成整数与小数部分，转换方法不同。

① 整数部分——除 2 取余，倒序排列，如下所示：



所以 $(67)_{10} = (1000011)_2$ 。

② 小数部分——乘 2 取整，正序排列，如下所示：



所以 $(0.125)_{10} = (0.1100)_2$ 。

将上述部分合起来，则 $(67.125)_{10} = (1000011.1100)_2$ 。

(3) 二进制与十六进制互转

① 二进制转换为十六进制。以小数点为界,往左、往右4位二进制数为一组,每4位二进制数用1位十六进制表示,往左高位不够用0补齐,往右低位不够用0补齐,例如:

$$(1010011.011111)_2 = (0101\ 0011.0111\ 1100)_2 = (53.7C)_{16}$$

② 十六进制转换为二进制。每位十六进制数用4位二进制数表示,再将整数部分最高位的0去掉,小数部分最低位的0去掉,例如:

$$(5E18.26)_{16} = (0101\ 1110\ 0001\ 1000.0010\ 0110)_2 = (101111000011000.0010011)_2$$

常用的数制转换工具很多,我们最常用的就是PC机附件中的计算器(科学型),可实现各数之间的相互转换。打开计算器工具界面“查看”菜单栏中选择“程序员”。

转换方法:先选择被转换数制的类型,输入转换数字,再选择目标数制转换类型,此时输出显示的就是转换后的数字。如125转换为十六进制、二进制,先选择数制类型为十进制,在输入框中输入数字125,然后再选择数制类型十六进制,此时显示框中的数字即为转换后的十六进制数字7D,再选择数制类型为二进制,此时显示框中的数字即为转换后的二进制数字1111101。

3. 二进制的算术运算规则

(1) 加法运算规则

$$0+0=0,0+1=1,1+1=0(\text{有进位})$$

(2) 减法运算规则

$$0-0=0,1-0=1,1-1=0,0-1=1(\text{有借位})$$

(3) 乘法运算规则

$$0\times 0=0,1\times 0=0,1\times 1=1$$

(4) 除法运算规则

$$0\div 0(\text{无意义}),0\div 1=0,1\div 1=1,1\div 0(\text{无意义})$$

4. 二进制的逻辑运算规则

微型计算机应用经常用到的逻辑运算包括:与、或、异或、非,这些逻辑包括字节运算和位运算,在单片机检测控制系统中经常用到位逻辑运算,两个二进制数的位与位之间进行相应的逻辑运算,没有进位和借位的情况。

(1) 逻辑“与”运算规则

逻辑“与”运算也称为逻辑乘法运算,运算符“ \times ”“ \wedge ”或“ \cdot ”。

$$0\times 0=0,0\times 1=0,1\times 0=0,1\times 1=1$$

(2) 逻辑“或”运算规则

逻辑“或”运算也称为逻辑加法运算,运算符“ $+$ ”“ \vee ”。

$$0+0=0,1+0=1,0+1=1,1+1=1$$

(3) 逻辑“异或”运算规则

逻辑“异或”运算也称为“半加”运算,两个二进制数异或的结果与加法相同,只是没有进位,运算符“ \oplus ”。简单记忆该逻辑就是相同为 0,相异为 1。

$$0 \oplus 0 = 0, 1 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 1 = 0$$

(4) 逻辑“非”运算规则

逻辑“非”运算即取反逻辑,运算符“ \neg ”。

1.1.2 微型计算机中数的表示方法

1. 机器数与真值

计算机中使用的二进制数称为机器数。数学中的数有正、负之分,计算机如何表示数的正负呢?在计算机中数据存放在存储单元中,而每个存储单元是由若干个二进制位组成的,其中每一数位或是 0 或是 1,这样,用一位二进制数刚好可以对应数的“+”号和“-”号。在计算机中规定用“0”表示“+”,用“1”表示“-”。用来表示符号的数位被称为“符号位”(通常为最高数位),这样数的正负就被数码化了。

设有两个数 x_1, x_2 分别为 +75 与 -75,其真值分别为:

$$x_1 = +1001011B, x_2 = -1001011B$$

它们在计算机里的机器数分别表示为(最高位为符号位,字长是 8 位):

$$x_1 = \underline{0}1001011B, x_2 = \underline{1}1001011B$$

以编码形式表示的数称为机器数,把原来一般书写形式表示的数称为真值。

若一个数的所有数位均为数值位,则该数为无符号数,这一类数据是逻辑数,没有正负的概念。例如,存储器地址就是一串无符号的二进制数,表示字符的 ASCII 码也是一组无符号的二进制数。

若一个数的最高位为符号位,其他数位为数值位,则该数为有符号数。思考一下对于 8 位的存储单元,存放无符号数的数值范围是多少,存放有符号数的数值范围又是多少。对于字长为 n 的存储单元,存放无符号数的数值范围是 $0 \sim 2^n - 1$;存放有符号数的数值范围是 $-2^{n-1} \sim +2^{n-1} - 1$ 。

2. 原码、反码、补码

机器数有原码、反码、补码 3 种表示方法。

原码是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,即最高数位表示符号,其余数位表示数值,则称为原码表示法。

反码是为求取补码而定义的,即正数的反码与原码相同。负数的反码由原码转换得到,转换方法:符号位不变,数值位按位取反。

补码的引进。以指针表的时钟校准问题为例说明补码的概念。时钟的计量范围是 $0 \sim 11$,模 = 12(“模”是指一个计量系统的计数范围)。

假如现在是上午7点整,而表却指向了11点,为了校准时间,可以采用倒拨和顺拨两种方法:倒拨就是逆时针减少4个小时(倒拨视为减法,相当于 $11-4=7$),时针指向7;也可以顺时针拨8个小时,时针同样指向7,把顺拨视为加法,相当于 $11+8=12$ (自动丢失)+ $7=7$,这自动丢失的12就叫作模(mod),上述加法就称为“按模12的加法”,用数学式表示为: $11+8=12+7=7(\text{mod}12)$ 。

因时钟转一圈会自动丢失12,故 $11-4$ 与 $11+8$ 是等价的,称8与 -4 对模12互补,8是 -4 对模12的补码。引进补码概念后,就可以将原来的减法 $11-4=7$ 转换为加法 $11+8(-4\text{的补码})=12(\text{自动丢失})+7=7(\text{mod}12)$ 。

补码的定义。通过上面的例子不难理解计算机中负数的补码表示法。表示 n 位的计算机计量范围是 $0\sim 2^n-1$,模 $=2^n$ 。“模”实质上是计量器产生“溢出”的量,它的值在计量器上表示不出来,计量器上只能表示出模的余数。任何有模的计量器,均可化减法为加法运算。若 x 为正数,则 $[x]_{\text{补}}=2^n+x$;若 x 为负数,则 $[x]_{\text{补}}=2^n-|x|$,即负数 x 的补码等于模 2^n 加上其真值或减去其真值的绝对值。

求补码的方法。因为正数的补码与原码反码一致,所以只有负数的补码需要求解。常用负数求补码的方法有三种。

- ① 根据定义求补码,即 $[x]_{\text{补}}=2^n+x$ 或 $[x]_{\text{补}}=2^n-|x|$ 。
- ② 根据反码求补码,(最常用的方法) $[x]_{\text{补}}=[x]_{\text{反}}+1$ 。
- ③ 根据原码求补码,从原码的最低位开始到第一个为1的位之间(包括此位)的各位均不变,此后各位取反,但符号位保持不变。

补码系统的最大优点是可以在加法或减法处理中,不需因为数字的正负而使用不同的计算方式。只要一种加法电路就可以处理各种有符号数加法,而且减法可以用一个数加上另一个数的补码来表示,因此只要有加法电路及补码电路即可完成各种有符号数的加法及减法运算,在电路设计上相当方便。另外,可以让符号位作为数值直接参加运算,而最后仍然可以得到正确的结果符。

补码系统的0只有一个表示方式,因此在判断数字是否为0时,只要比较一次即可。

特别指出,在计算机中凡是带符号的数一律用补码表示且符号位参加运算,其运算结果也是用补码表示,若结果的符号位为“0”,则表示结果为正数,此时可以认为它是以原码形式表示的;若结果的符号位为“1”,则表示结果为负数,它是以补码形式表示的,如果需要原码来表示该结果,还需要对结果求补,即 $[[x]_{\text{补}}]_{\text{补}}=[x]_{\text{原}}$ 。

1.1.3 微型计算机中常用编码

微型计算机不但要处理数值计算问题,也要处理大量非数值计算问题,因此除了数值需要二进制数表示之外,不论是十进制数还是英文字母、汉字以及某些专用符号和控制命令都必须编成二进制代码,这样它们才能被计算机识别、接收、存储、传送、处理。

1. 十进制数的编码(BCD 码)

在计算机中,十进制数除了转换成二进制数外,还可用二进制数对其进行编码:用 4 位二进制数表示 1 位十进制数,使它既具有二进制数的形式又具有十进制数的特点,即通常所说的二-十进制编码。这种编码又称为 BCD 码(Binary-Coded Decimal),它有 8421 码、5421 码、2421 码以及余 3 码等几种编码形式,其中 8421 码最常用。8421 码与十进制数的对应关系如表 2.1 所示,每位二进制数位都有固定的“权”,各数位的权从左到右分别为 2^3 、 2^2 、 2^1 、 2^0 ,即 8、4、2、1,这与自然二进制数的权完全相同,故 8421 码又称为自然权 BCD 码。其中 1010~1111 这 6 个代码,是不允许出现的,属于非法 8421BCD 码。

表 1.2 8421BCD 编码表

十进制数	8421BCD 码	十进制数	8421BCD 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

BCD 码低位与高位之间是“逢十进一”,而 4 位二进制数(十六进制数)是“逢十六进一”,用二进制加法器进行 BCD 码运算时,如果 BCD 码运算的低、高位的和都在 0~9 之间,则其加法运算规则与二进制加法完全一样;如果相加后某位(BCD 码位,低 4 位或高 4 位)的和大于 9 或有进位,则此位应进行“加 6 调整”。通常在微型计算机中都设置有 BCD 码调整电路,机器执行一条十进制调整指令,机器就会自动根据刚才的二进制加法结果进行修正。BCD 码向高位借位是“借一当十”,而 4 位二进制数(1 位十六进制数)是“借一当十六”,因此在进行 BCD 码减法运算时,如果某位(BCD 码位)有借位时,必须在该位进行“减 6 调整”。

2. 字符编码

微型计算机需要对非数值进行处理(如指令、数据的输入、文字的输入及处理等),必须对字母、文字及某些专用符号进行编码。微型计算机中的字符编码多采用美国信息交换标准代码,如表 1.3 所示。共 128 个字符,其中 94 个是图形字符,可在字符印刷或显示设备上打印出来,包括数字符号 10 个,英文大小写字母 52 个,其他字符 32 个。还有 34 个控制字符,包括传输字符、格式控制字符、设备控制字符、信息分隔符和其他控制字符,这类字符不打印、不显示,但其编码可以进行存储,在信息交换中起控制作用。其中数字 0~9 对应的 ASCII 码是 30H~39H,英文大写字母 A~Z 对应的 ASCII 码是 41H~5AH,小写字母 a~z 对应的 ASCII 码是 61H~7AH,这些规律对今后码制转换的编程非常有用。