



免费提供电子教案



21 世纪 高 等 院 校 电 气 信 息 类 系 列 教 材

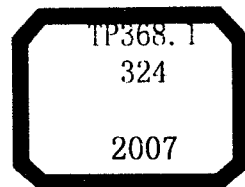
单片机原理及应用

李念强 王玉泰 张鲁 张羽 编著



机械工业出版社
CHINA MACHINE PRESS





21 世纪高等院校电气信息类系列教材

单片机原理及应用

李念强 王玉泰 张鲁 张羽 编著

机械工业出版社

本书共 11 章,前 7 章讲解单片机硬件结构、功能特点、指令系统与程序设计、中断系统、通信接口等基本原理、基本概念;后 4 章阐述单片机应用系统中的抗干扰设计、单片机系统外部功能扩展,并结合应用举例,详细介绍了系统开发与应用,使读者真正了解与掌握单片机应用系统从设计、调试到工程实现的全过程。同时还介绍了单片机仿真系统 Proteus-IsIs。本书叙述清晰、通俗易懂、重点突出、理论与应用紧密结合,且配有初始化程序,实用性强。

本书可作为普通高校计算机及应用、工业自动化、自动化仪表、电子信息及其相关专业师生的教材和教学参考书,也可作为广大工程技术人员及单片机开发人员的参考资料。

图书在版编目(CIP)数据

单片机原理及应用/李念强等编著. —北京:机械工业出版社,2007.3
(21 世纪高等院校电气信息类系列教材)
ISBN 978-7-111-21108-2

I. 单... II. 李... 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(2007)第 031805 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:时 静
责任编辑:时 静 李馨馨
责任印制:杨 曦

北京市朝阳区展望印刷厂印刷

2007 年 4 月第 1 版·第 1 次印刷
184mm×260mm·17.75 印张·434 字
0001—5000 册
标准书号:ISBN 978-7-111-21108-2
定价:25.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
销售服务热线电话:(010)68326294
购书热线电话:(010)88379639 88379641 88379643
编辑热线电话:(010)88379739
封面无防伪标均为盗版

出版说明

随着科学技术的不断进步，整个国家自动化水平和信息化水平的长足发展，社会对电气信息类人才的需求日益迫切、要求也更加严格。在教育部颁布的“普通高等学校本科专业目录”中，电气信息类（Electrical and Information Science and Technology）包括电气工程及其自动化、自动化、电子信息工程、通信工程、计算机科学与技术、电子科学与技术、生物医学工程等专业。这些专业的人才培养对社会需求、经济发展都有着非常重要的意义。

在电气信息类专业及学科迅速发展的同时，也给高等教育工作带来了许多新课题和新任务。在此情况下，只有将新知识、新技术、新领域逐渐融合到教学、实践环节中去，才能培养出优秀的科技人才。为了配合高等院校教学的需要，机械工业出版社组织了这套“21世纪高等院校电气信息类系列教材”。

本套教材是在对电气信息类专业教育情况和教材情况调研与分析的基础上组织编写的，期间，与高等院校相关课程的主讲教师进行了广泛的交流和探讨，旨在构建体系完善、内容全面新颖、适合教学的专业教材。

本套教材涵盖多层面专业课程，定位准确，注重理论与实践、教学与教辅的结合，在语言描述上办求准确、清晰，适合各高等院校电气信息类专业学生使用。

机械工业出版社

前 言

目前,单片机作为微型计算机的一个重要分支,发展迅速,应用广泛。尤其是 AT89X51 系列单片机,在我国已经得到了广泛的应用。

本书以近年来被广泛使用的单片机 AT89S51 为例,从原理、接口及实践应用三个方面进行了系统介绍。在内容的选择及编写方面,力求实用和典型。书中的应用设计实例,大多来自单片机的应用实践,且经过检验。在本书的第 10 章,介绍了单片机设计的一般过程与方法,并在作者所做过的单片机应用系统中,选出两个典型应用实例进行了详细分析。

对于在软件设计中经常遇到的各种常用程序的设计及硬件接口设计中的各种常用和典型的设计方案,都作了详细的介绍。许多功能子程序和典型的接口方案均可以直接使用。

本书在写作上,力求文字精炼、通俗易懂。没有把过多的篇幅放在介绍芯片的工作原理上,而是从应用出发,重点介绍其使用方法。突出理论联系实际,强调工程性和实用性,并力求结构合理、深入浅出,便于理解与自学。

全书共分 11 章。前 7 章从应用设计方面介绍了 AT89S51 单片机的硬件结构、工作原理及指令系统;第 8 章介绍单片机的典型接口设计;第 9 章介绍单片机系统的抗干扰设计;第 10 章借助于应用实例介绍了单片机应用系统设计的一般方法。第 11 章介绍了单片机仿真系统的环境和使用方法。

本书第 2、4、5、6、9、11 章由济南大学李念强编写;第 1、3 章由济南大学张鲁编写;第 10 章由王玉泰编写;第 7、8 章由山东电力高等专科学校张羽编写。全书由李念强统稿。张梦麟硕士绘制了本书的大部分图表并对程序进行了调试,在本书编写过程中曾参考和引用了国外许多专家与学者的论文和专著,在此一并致谢。

由于作者水平所限,书中难免存在缺点和错误,欢迎广大读者批评指正。

编 者

2007 年 1 月于济南

目 录

出版说明

前言

第 1 章 数制与编码	1
1.1 进位计数制及各计数制间的转换	1
1.2 二进制数的运算	2
1.2.1 二进制数的算术运算	2
1.2.2 二进制数的逻辑运算	2
1.3 带符号数的表示方法——原码、反码、补码	3
1.4 定点数与浮点数	3
1.4.1 定点表示法	4
1.4.2 浮点表示法	4
1.5 BCD 码和 ASCII 码	5
1.5.1 BCD 码(Binary Coded Decimal)	5
1.5.2 BCD 码运算及十进制调整	6
1.5.3 ASCII 码	6
1.6 习题	7
第 2 章 AT89S 系列单片微机系统结构	8
2.1 单片微型计算机发展概述	8
2.1.1 单片机的应用	9
2.1.2 MCS-51 和 AT89 系列单片机	10
2.2 AT89S 系列单片微机的硬件结构	12
2.2.1 AT89S51 单片机的硬件结构特点	12
2.2.2 AT89S51 单片机的引脚功能	14
2.3 AT89S51 单片机的中央处理器	15
2.3.1 算术/逻辑运算部件 ALU	15
2.3.2 专用寄存器	16
2.3.3 振荡器、时钟电路及时序	17
2.4 并行 I/O 口结构	19
2.5 AT89S51 单片机的存储器结构	21
2.5.1 程序存储器	22
2.5.2 内部数据存储器	23
2.5.3 特殊功能寄存器(SFR)	23
2.5.4 位地址空间	26
2.5.5 外部数据存储器	27
2.6 AT89S5X 单片微机的布尔处理器	27
2.7 AT89S5X 单片微机的工作方式	28
2.7.1 复位方式	28
2.7.2 程序执行方式	30

2.7.3 低功耗工作方式	30
2.8 AT89S5X 单片机的在线编程 ISP	31
2.8.1 编程算法	32
2.8.2 数据查询	33
2.8.3 串行编程命令	33
2.9 习题	33
第 3 章 AT89 系列单片机的指令系统	35
3.1 单片机系统的寻址方式	35
3.1.1 指令概述	35
3.1.2 寻址方式	36
3.2 单片机指令系统的分类	39
3.2.1 数据传送类指令	39
3.2.2 算术运算类指令	43
3.2.3 逻辑运算及移位类指令	47
3.2.4 控制转移类指令	50
3.2.5 位操作(布尔操作)指令	54
3.3 习题	57
第 4 章 AT89S5X 单片机汇编语言程序设计基础	60
4.1 程序设计概述	60
4.1.1 汇编语言程序设计的步骤与方法	60
4.1.2 常用伪指令	60
4.2 程序设计基础与举例	61
4.2.1 简单结构程序	61
4.2.2 分支结构程序	62
4.2.3 循环结构程序	65
4.2.4 子程序结构程序	68
4.2.5 代码转换程序	70
4.2.6 算术运算程序	72
4.2.7 查表程序设计	76
4.2.8 散转(多分支)程序设计	78
4.3 习题	80
第 5 章 AT89 单片机的中断系统	82
5.1 AT89 单片机的中断及中断请求源	82
5.1.1 中断简介	82
5.1.2 AT89S51 中断源	82
5.2 AT89S51 单片机的中断优先级	84
5.3 AT89S51 单片机的中断响应过程及中断矢量地址	85
5.3.1 AT89S51 单片机的中断响应过程	85
5.3.2 中断矢量地址	86
5.3.3 中断的编程及应用	87
5.4 习题	89

第 6 章 AT89 系列单片机的定时器/计数器	90
6.1 AT89S51 单片机的定时器/计数器的基本原理	90
6.2 定时器/计数器 T0、T1 的内部结构	90
6.3 AT89S51 单片机的定时器/计数器工作方式	92
6.4 定时器/计数器 T2	94
6.4.1 定时器/计数器 T2 中的特殊功能寄存器	94
6.4.2 定时器/计数器 T2 的功能选择	96
6.4.3 定时器/计数器 T2 的工作方式	96
6.5 AT89S51 单片机的定时器/计数器编程和应用	99
6.5.1 定时器/计数器溢出率的计算	99
6.5.2 定时器/计数器的编程	100
6.5.3 定时器/计数器的应用举例	100
6.6 习题	107
第 7 章 AT89 系列单片机的串行接口	108
7.1 串行通信概述	108
7.1.1 串行通信传送方式	108
7.1.2 数据传送方向	109
7.2 AT89S51 单片机的串行口结构	110
7.2.1 UART 串行口的结构	110
7.2.2 串行口的控制	111
7.3 AT89S51 单片机的串行口工作方式	112
7.3.1 方式 0	112
7.3.2 方式 1	113
7.3.3 方式 2 和 3	114
7.4 AT89S51 单片机的多处理器通信	115
7.5 AT89S51 单片机的波特率制定方法	115
7.6 AT89S51 单片机串行口的编程和应用	117
7.6.1 方式 0 的编程和应用	117
7.6.2 方式 1 的编程和应用	118
7.6.3 方式 2 的编程和应用	122
7.6.4 方式 3 的编程和应用	122
7.7 习题	124
第 8 章 AT89S 系列单片机系统的扩展	125
8.1 AT89S 系列单片机系统扩展概述	125
8.1.1 最小应用系统	125
8.1.2 系统扩展的内容与方法	126
8.1.3 常用的扩展器件简介	127
8.2 存储器的扩展	128
8.2.1 存储器扩展概述	128
8.2.2 程序存储器的扩展	129
8.2.3 数据存储器的扩展	131

8.3 单片机 I/O 口的扩展	133
8.3.1 并行 I/O 口的扩展	133
8.3.2 串行 I/O 口的扩展技术	145
8.4 串行标准接口的扩展	157
8.4.1 AT89S51 配置 RS-232C 标准串行接口	157
8.4.2 RS-422A 标准接口	160
8.4.3 RS-485 标准接口	161
8.5 A/D 和 D/A 转换器的扩展	162
8.5.1 A/D 转换器的扩展	162
8.5.2 D/A 转换器的扩展	170
8.6 键盘扩展电路	176
8.6.1 单片机应用系统中的键输入	176
8.6.2 独立式按键	178
8.6.3 行列式键盘	181
8.7 显示扩展电路	187
8.7.1 LED 显示及显示器接口	187
8.7.2 LED 显示器与显示方式	188
8.7.3 硬件译码显示器接口	190
8.8 专用显示与键盘管理芯片	192
8.8.1 ZLG7289 串行接口 LED 数码管及键盘管理器件	192
8.8.2 ET6202 串行接口 LED 数码管及键盘管理器件	202
8.9 外部中断扩展电路	210
8.10 习题	212
第 9 章 单片机应用系统中的软件抗干扰设计	213
9.1 干扰的分类及来源	213
9.1.1 差模干扰	213
9.1.2 共模干扰	213
9.2 常用的抗干扰器件	214
9.2.1 隔离变压器	214
9.2.2 光耦合器	215
9.2.3 隔离放大器	216
9.2.4 滤波器	217
9.2.5 DC-DC 转换器	217
9.3 硬件抗干扰	218
9.3.1 差模干扰的抑制方法	218
9.3.2 共模干扰的抑制方法	219
9.4 软件抗干扰技术	219
9.4.1 数字滤波	219
9.4.2 单片机执行过程中的软件抗干扰措施	227
9.5 习题	231
第 10 章 AT89S 单片机系统的应用设计与分析	233

10.1 单片机应用系统的设计	233
10.1.1 单片机应用系统的开发过程	233
10.1.2 单片机应用系统设计的基本原则与方法	234
10.1.3 单片机应用系统的一般结构	236
10.1.4 单片机应用系统的调试	236
10.2 单片机应用开发实例	238
10.2.1 智能 RLC 数字电桥的设计	238
10.2.2 遥控调速电动机的设计	242
10.3 习题	244
第 11 章 单片机仿真软件 Proteus ISIS 介绍	245
11.1 Proteus 6 Professional 界面简介	246
11.2 操作简介	249
11.3 实例——利用 AT89C51 单片机和 DS1302 时钟芯片实现数字钟	249
附录	258
附录 A ASCII 码表	258
附录 B 51 单片机指令表	260
附录 C 常用芯片引脚排列图	264
参考文献	271

第1章 数制与编码

数制是人们对事物数量计数的一种统计规律。在日常生活中最常用的是十进制,但在计算机中,由于其电气元件最易实现的是两种稳定状态:器件的“开”与“关”,电平的“高”与“低”。因此,采用二进制数的“0”和“1”可以很方便地表示计算机内的数据运算与存储。

1.1 进位计数制及各计数制间的转换

二进制数有运算简单、便于物理实现、节省设备等优点,所以目前在计算机中数都是采用二进制数表示。但是二进制数书写起来太长,且不便阅读和记忆。目前大部分微型机是8位、16位或32位的,都是4的整数倍,而4位二进制数即是1位十六进制数,所以微型机广泛采用十六进制数来缩写二进制数。十六进制数用0~9、A~F共16个数码表示十进制数0~15。1个8位的二进制数用2位十六进制数表示,1个16位的二进制数用4位十六进制数表示等。这样书写方便,且便于阅读和记忆。为避免混淆,不同进制数常用数字后加字母作为标注。其中字母B(Binary)表示二进制数;字母Q(Octal的缩写为字母O,为区别数字0故写成Q)表示八进制数;字母D(Decimal)或不加字母表示十进制数;字母H(Hexadecimal)表示十六进制数。

然而,人们最熟悉、最常用的是十进制数。为此,需要熟练地掌握十进制数、二进制数和十六进制数间的相互转换。

1. 各种进制数转换成十进制数

各种进制数转换成十进制数的方法是:将各进制数先按权展成多项式,再利用十进制运算法则求和,即可得到该数对应的十进制数。

【例1-1】 将二进制数1001.101转换为十进制数。

$$\begin{aligned}\text{解: } 1001.101\text{B} &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 8 + 1 + 0.5 + 0.125 = 9.625\end{aligned}$$

2. 十进制数转换为二、八、十六进制数

任一十进制数转换成M进制数,先将整数部分与小数部分分为两部分,并分别进行转换,然后再用小数点将这两部分连接起来。

1) 整数部分转换。十进制整数转换成M进制数可用除M取余法,即用M不断地除待转换的十进制数,直到商等于0为止。将所得的各次余数,依倒序排列,即可得到所转换的M进制整数。

2) 小数部分转换。十进制小数转换成M进制数可用乘M取整法,即用M不断地乘待转换的十进制数,直到积等于0为止。将所得的各次整数,依顺序排列,即可得到所转换的M进制小数。

【例1-2】 将十进制数168.686转换为十六进制数。

解: 分析:十进制数168.686既有整数又有小数,所以要分为两步转换。整数部分168,整数部分采用除16取余法:

$$16 \overline{) 168}$$

$$16 \overline{) 10} \quad \text{余数 } 8, \quad K_0 = 8$$

$$0 \quad \text{余数 } 10, \quad K_1 = A$$

$$168 = A8H$$

小数部分 0.686, 小数部分采用乘 16 取整法:

$$0.686 \times 16 = 10.976 \quad K_{-1} = A$$

$$0.976 \times 16 = 15.616 \quad K_{-2} = F$$

$$0.616 \times 16 = 9.856 \quad K_{-3} = 9$$

$$0.856 \times 16 = 13.696 \quad K_{-4} = D$$

$$0.696 \times 16 = 11.136 \quad K_{-5} = B$$

$$0.686 \approx 0.AF9DBH$$

把整数和小数部分结合起来即为最后结果 $168.686 \approx A8.AF9DBH$

1.2 二进制数的运算

1.2.1 二进制数的算术运算

二进制数不仅物理上容易实现,而且算术运算也比较简单,其加、减法遵循“逢 2 进 1”、“借 1 当 2”的原则。

1. 二进制加法

1 位二进制数的加法规则为:

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10(\text{有进位})$$

2. 二进制减法

1 位二进制数减法规则为:

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

3. 二进制乘法

1 位二进制乘法规则为:

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

4. 二进制除法

二进制除法的运算过程类似于十进制除法的运算过程。

1.2.2 二进制数的逻辑运算

1. “与”运算(AND)

“与”运算又称逻辑乘,运算符为 $\cdot$ 或 $\wedge$ 。“与”运算的规则如下:

$$0 \cdot 0 = 0 \quad 0 \cdot 1 = 1 \cdot 0 = 0 \quad 1 \cdot 1 = 1$$

2. “或”运算(OR)

“或”运算又称逻辑加,运算符为 $+$ 或 $\vee$ 。“或”运算的规则如下:

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

3. “非”运算(NOT)

“非”运算又称逻辑非,如变量 A 的“非”运算记作 $\bar{A}$ 。“非”运算的规则如下:

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

4. “异或”运算(XOR)

“异或”运算的运算符为 $\oplus$,其运算规则如下:

$$0\oplus 0=0 \quad 0\oplus 1=1\oplus 0=1 \quad 1\oplus 1=0$$

1.3 带符号数的表示方法——原码、反码、补码

在前面讨论的二进制数运算均为无符号数运算,但实际的数值是带有符号的,既可能是正数,也可能是负数,前者符号用“+”号表示,后者符号用“-”号表示,运算的结果也可能是正数,也可能是负数。于是在计算机中就存在着如何表示正、负数的问题。

由于计算机只能识别 0 和 1,因此,在计算机中通常把一个二进制数的最高位作为符号位,以表示数值的正与负(若用 8 位表示一个数,则 D7 位为符号位;若用 16 位表示一个数,则 D15 位为符号位),并用 0 表示“+”;用 1 表示“-”。

下面介绍一下原码、补码与反码:

1. 原码

正数的符号位用 0 表示,负数的符号位用 1 表示,数值部分用真值的绝对值来表示的二进制数称之为原码。

2. 反码

一个正数的反码,等于该数的原码;一个负数的反码,等于该负数的原码符号位不变(即为 1),数值位按位求反(即 0 变 1,1 变 0)。

3. 补码

正数的补码与原码相同;负数的补码为该负数的原码符号位不变,数值位逐位求反并在末位加 1。

【例 1-3】 假设 $X_1 = +83$, $X_2 = -76$, 当用 8 位二进制数表示一个数时,求 X_1 、 X_2 的原码、反码及补码。

解: $[X_1]_{\text{原}} = [X_1]_{\text{反}} = [X_1]_{\text{补}} = 01010011\text{B}$

$$[X_2]_{\text{原}} = 11001100\text{B}$$

$$[X_2]_{\text{反}} = 10110011\text{B}$$

$$[X_2]_{\text{补}} = [X]_{\text{反}} + 1 = 10110100\text{B}$$

综上所述可归纳为:

正数的原码、反码、补码就是该数本身;

负数的原码其符号位为 1,数值位不变;

负数的反码其符号位为 1,数值位逐位求反;

负数的补码其符号位为 1,数值位逐位求反并在末位加 1。

1.4 定点数与浮点数

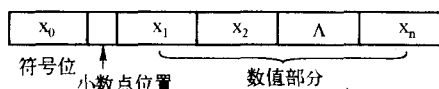
计算机处理的数值数据多数带有小数,小数点在计算机中通常有两种表示方法,一种是约

定所有数值数据的小数点隐含在某一个固定位置上,称为定点表示法,简称定点数;另一种是小数点位置可以浮动,称为浮点表示法,简称浮点数。

1.4.1 定点表示法

所谓定点格式,即约定机器中所有数据的小数点位置是固定不变的。在计算机中通常采用两种简单的约定:将小数点的位置固定在数据的最高位之前,或者是固定在最低位之后。一般常称前者为定点小数,后者为定点整数。

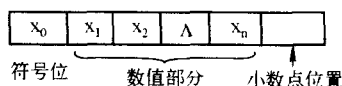
定点小数是纯小数,约定的小数点位置在符号位之后、有效数值部分最高位之前。若数据 x 的形式为 $x = x_0.x_1x_2\ldots x_n$ (其中 x_0 为符号位, $x_1 \sim x_n$ 是数值的有效部分,也称为尾数, x_1 为最高有效位),则在计算机中的表示形式为:



一般说来,如果最末位 $x_n = 1$,前面各位都为 0,则数的绝对值最小,即 $|x|_{\min} = 2^{-n}$ 。如果各位均为 1,则数的绝对值最大,即 $|x|_{\max} = 1 - 2^{-n}$ 。所以定点小数的表示范围是:

$$2^{-n} \leq |x| \leq 1 - 2^{-n}$$

定点整数是纯整数,约定的小数点位置在有效数值部分最低位之后。若数据 x 的形式为 $x = x_0x_1x_2\ldots x_n$ (其中 x_0 为符号位, $x_1 \sim x_n$ 是尾数, x_n 为最低有效位),则在计算机中的表示形式为:



定点整数的表示范围是:

$$1 \leq |x| \leq 2^n - 1$$

当数据小于定点数能表示的最小值时,计算机将它们作 0 处理,称为下溢;大于定点数能表示的最大值时,计算机将无法表示,称为上溢。上溢和下溢统称为溢出。

计算机采用定点数表示时,对于既有整数又有小数的原始数据,需要设定一个比例因子,数据按其缩小成定点小数或扩大成定点整数再参加运算。运算结果,根据比例因子,还原成实际数值。若比例因子选择不当,往往会使运算结果产生溢出或降低数据的有效精度。

用定点数进行运算处理的计算机被称为定点机。

1.4.2 浮点表示法

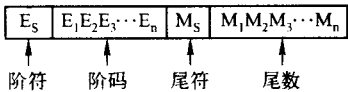
与科学计数法相似,任意一个 J 进制数 N ,总可以写成

$$N = J^E \times M$$

式中 M 称为数 N 的尾数(Mantissa),是一个纯小数; E 为数 N 的阶码(Exponent),是一个整数, J 称为比例因子 J^E 的底数。这种表示方法相当于数的小数点位置随比例因子的不同而在一定范围内可以自由浮动,所以称为浮点表示法。

底数是事先约定好的(常取 2),在计算机中不出现。在机器中表示一个浮点数时,一是要给出尾数,用定点小数形式表示。尾数部分给出有效数字的位数,因而决定了浮点数的表示精

度。二是要给出阶码,用整数形式表示,阶码指明小数点在数据中的位置,因而决定了浮点数的表示范围。浮点数也要有符号位。因此一个机器浮点数应当由阶码和尾数及其符号位组成:



其中 E_s 表示阶码的符号,占一位, $E_1\sim E_n$ 为阶码值,占 n 位,尾符是数 N 的符号,也要占一位。当底数取 2 时,二进制数 N 的小数点每右移一位,阶码减小 1,相应尾数右移一位;反之,小数点每左移一位,阶码加 1,相应尾数左移一位。

若不对浮点数的表示作出明确规定,同一个浮点数的表示就不是惟一的。例如 11.01 也可以表示成 0.01101×2^3 , 0.1101×2^2 等。为了提高数据的表示精度,当尾数的值不为 0 时,其绝对值应大于等于 0.5,即尾数域的最高有效位应为 1,否则要以修改阶码同时左右移小数点的方法,使其变成这一要求的表示形式,这称为浮点数的规格化表示。

当一个浮点数的尾数为 0 时,不论其阶码为何值,或者当阶码的值遇到比它能表示的最小值还小时,不管其尾数为何值,计算机都把该浮点数看成 0 值,称为机器零。

浮点数所表示的范围比定点数大。假设机器中的数由 8 位二进制数表示(包括符号位):在定点机中这 8 位全部用来表示有效数字(包括符号);在浮点机中若阶符、阶码占 3 位,尾符、尾数占 5 位,在此情况下,若只考虑正数值,定点机小数表示的数的范围是 0.0000000 到 0.1111111,相当于十进制数的 0 到 $127/128$,而浮点机所能表示的数的范围则是 $2^{-11}\times 0.0001$ 到 $2^{11}\times 0.1111$,相当于十进制数的 $1/128$ 到 7.5。显然,都用 8 位,浮点机能表示的数的范围比定点机大得多。

尽管浮点表示能扩大数据的表示范围,但浮点机在运算过程中,仍会出现溢出现象。

1.5 BCD 码和 ASCII 码

1.5.1 BCD 码(Binary Coded Decimal)

二进制数以其物理易实现和运算简单的优点在计算机中得到了广泛应用,但人们日常习惯最熟悉的还是十进制。为了既满足人们的习惯,又能让计算机接受,便引入了 BCD 码。它用二进制数码按照不同规律编码来表示十进制数,这样的十进制数的二进制编码,既具有二进制的形式,又具有十进制的特点,便于传递处理。

1 位十进制数有 0~9 共 10 个不同数码,需要由 4 位二进制数来表示。4 位二进制数有 16 种组合,取其 10 种组合分别代表 10 个十进制数码。最常用的方法是 8421BCD 码,其中 8、4、2、1 分别为 4 位二进制数的位权值。表 1-1 给出了十进制数和 8421BCD 码的对应关系。

表 1-1 8421BCD 码

十 进 制 数	8421BCD 码	十 进 制 数	8421BCD 码
0	0000	4	0100
1	0001	5	0101
2	0010	6	0110
3	0011	7	0111

(续)

十进制数	8421BCD码	十进制数	8421BCD码
8	1000	12	0001 0010
9	1001	13	0001 0011
10	0001 0000	14	0001 0100
11	0001 0001	15	0001 0101

1.5.2 BCD 码运算及十进制调整

若想让计算机直接用十进制的规律进行运算,则将数据用 BCD 码来存储和运算即可。

但是,8421BCD 码可表示数的范围为 0000~1111(即十进制的 0~15),而十进制数为 0000~1001(即 0~9)。所以,在运算时,必须注意以下两点:

① 当两个 BCD 数相加结果大于 1001(即大于十进制数 9)时,为使其符合十进制运算和进位规律,需对 BCD 码的二进制运算结果加 0110(加 6)调整。

例如: $4 + 8 = (0100)_{\text{BCD}} + (1000)_{\text{BCD}} = (1100)_{\text{BCD}} > 1001$,

调整后,其结果为: $(1100)_{\text{BCD}} + (0110)_{\text{BCD}} = (00010010)_{\text{BCD}} = 12$ 。

② 当两个 BCD 数相加结果在本位上并不大于 1001,但有低位进位发生,使得两个 BCD 数与进位一起相加,其结果大于 1001,这时也要作加 0110(加 6)调整。

【例 1-4】用 BCD 数完成 $54 + 48$ 的运算。

解: $54 = (01010100)_{\text{BCD}}$, $48 = (01001000)_{\text{BCD}}$

$54 + 48 = (01010100)_{\text{BCD}} + (01001000)_{\text{BCD}} = (000100000010)_{\text{BCD}} = 102$, 计算过程如图 1-1 所示。

$$\begin{array}{r}
 01001000 \\
 + \quad 01010100 \\
 \hline
 10011100 \quad (\text{低4位大于9}) \\
 + \quad \quad 0110 \quad (\text{低4位加6调整}) \\
 \hline
 10100010 \quad (\text{低4位有进位}) \\
 + \quad \quad 0110 \quad (\text{高4位加6调整}) \\
 \hline
 000100000010 \\
 \text{则} (000100000010)_{\text{BCD}} = 102
 \end{array}$$

图 1-1 BCD 码加法

1.5.3 ASC II 码

在计算机的应用过程中,如操作系统命令、各种程序设计语言以及计算机运算和处理信息的输入输出,经常用到某些字母、数字或各种符号,如:英文字母的大、小写;0~9 数字符;+、-、*、/运算符;<、>、=关系运算符等。但在计算机内,任何信息都是用代码表示的,因此,这些符号也必须要有自己的编码。

ASCII 码采用 7 位二进制数对字符进行编码,它包括 10 个十进制数 0~9;大写和小写英文字母各 26 个;32 个通用控制符号;34 个专用符号,共 128 个字符。其中数字 0~9 的 ASCII 编码分别为 30H~39H,英文大写字母 A~Z 的 ASCII 编码从 41H 开始依次编至 5AH。ASCII 编码从 20H~7EH 均为可打印字符,而 00H~1FH 为通用控制符,它们不能被打印出来,只起

控制或标志的作用,如 0DH 表示回车(CR),0AH 表示换行控制(LF),04H(EOT)为传送结束标志。详细的表格见附录 I。

1.6 习题

1. 十进制和二进制数各有什么特点? 请举例加以说明。
2. 十六进制数有什么特点? 学习十六进制数的目的是什么?
3. 什么是定点数,浮点数? 两者有什么优缺点?
4. 把下列十进制数转换为二进制数和十六进制数:

① 135 ② 0.625 ③ 47.6875 ④ 111.111

5. 把下列二进制转换为十进制和十六进制数:

① 11010110B ② 0.1011B ③ 1011.1011B

6. 先把各数变成二进制形式,然后完成加法和减法运算,写在前面的数为被加数或者被减数:

① 97H 和 0FH ② A6H 和 33H ③ F3 和 F4H ④ B6H 和 EDH

7. 写出下列十进制数在 8 位二进制数(含符号位)机中的原码、反码和补码:

① $X = +38$ ② $X = +76$ ③ $X = -54$ ④ $X = -115$

8. 写出下列各数的 BCD 码:

① 47 ② 59 ③ 1996 ④ 1997.6

9. 用十六进制形式写出下列字符的 ASCII 码:

① AB8 ② STUDENT ③ COMPUTER ④ GOOD