



新世纪高职高专
电气自动化技术类课程规划教材

单片机原理及应用

新世纪高职高专教材编审委员会组编

主编 田亚娟 主审 常建刚

大连理工大学出版社

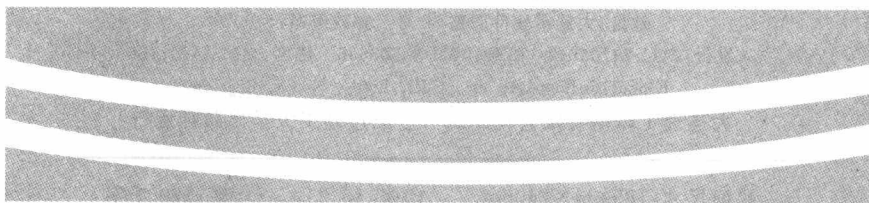


新世纪高职高专电气自动化技术类课程规划教材

单片机原理及应用

新世纪高职高专教材编审委员会组编

主 编 田亚娟 副主编 杨志邦 宁玉珊 朱海洋
主 审 常建刚



DANPIANJI YUANLI JI YINGYONG

大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

图书在版编目(CIP)数据

单片机原理及应用/田亚娟主编. —大连:大连理工大学出版社, 2008. 9

新世纪高职高专电气自动化技术类课程规划教材

ISBN 978-7-5611-4420-6

I. 单… II. 田… III. 单片微型计算机—高等学校: 技术学校—教材 IV. TP368.1

中国版本图书馆 CIP 数据核字(2008)第 129145 号

大连理工大学出版社出版

地址:大连市软件园路 80 号 邮政编码:116023

发行:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail: dutp@dutp.cn URL: <http://www.dutp.cn>

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm 印张:16.25 字数:380 千字

2008 年 9 月第 1 版

2008 年 9 月第 1 次印刷

责任编辑:赵晓燕 吴媛媛

责任校对:冯克柱

封面设计:张莹

ISBN 978-7-5611-4420-6

定 价:28.00 元

总 序

我们已经进入了一个新的充满机遇与挑战的时代,我们已经跨入了 21 世纪的门槛。

20 世纪与 21 世纪之交的中国,高等教育体制正经历着一场缓慢而深刻的革命,我们正在对传统的普通高等教育的培养目标与社会发展的现实需要不相适应的现状作历史性的反思与变革的尝试。

20 世纪最后的几年里,高等职业教育的迅速崛起,是影响高等教育体制变革的一件大事。在短短的几年时间里,普通中专教育、普通高专教育全面转轨,以高等职业教育为主导的各种形式的培养应用型人才的教育发展到与普通高等教育等量齐观的地步,其来势之迅猛,发人深思。

无论是正在缓慢变革着的普通高等教育,还是迅速推进着的培养应用型人才的高职教育,都向我们提出了一个同样的严肃问题:中国的高等教育为谁服务,是为教育发展自身,还是为包括教育在内的大千社会?答案肯定而且惟一,那就是教育也置身其中的现实社会。

由此又引出高等教育的目的问题。既然教育必须服务于社会,它就必须按照不同领域的社会需要来完成自己的教育过程。换言之,教育资源必须按照社会划分的各个专业(行业)领域(岗位群)的需要实施配置,这就是我们长期以来明乎其理而疏于力行的学以致用问题,这就是我们长期以来未能给予足够关注的教育目的问题。

如所周知,整个社会由其发展所需要的不同部门构成,包括公共管理部门如国家机构、基础建设部门如教育研究机构和各种实业部门如工业部门、商业部门,等等。每一个部门又可作更为具体的划分,直至同它所需要的各种专门人才相对应。教育如果不能按照实际需要完成各种专门人才培养的目标,就不能很好地完成社会分工所赋予它的使命,而教育作为社会分工的一种独立存在就应受到质疑(在市场经济条件下尤其如此)。可以断言,按照社会的各种不同需要培养各种直接有用人才,是教育体制变革的终极目的。

随着教育体制变革的进一步深入,高等院校的设置是否会同社会对人才类型的不同需要一一对应,我们姑且不论。但高等教育走应用型人才培养的道路和走研究型(也是一种特殊应用)人才培养的道路,学生们根据自己的偏好各取所需,始终是一个理性运行的社会状态下高等教育正常发展的途径。

高等职业教育的崛起,既是高等教育体制变革的结果,也是高等教育体制变革的一个阶段性表征。它的进一步发展,必将极大地推进中国教育体制变革的进程。作为一种应用型人才培养的教育,它从专科层次起步,进而应用本科教育、应用硕士教育、应用博士教育……当应用型人才培养的渠道贯通之时,也许就是我们迎接中国教育体制变革的成功之日。从这一意义上说,高等职业教育的崛起,正是在为必然会取得最后成功的教育体制变革奠基。

高等职业教育还刚刚开始自己发展道路的探索过程,它要全面达到应用型人才培养的正常理性发展状态,直至可以和现存的(同时也正处在变革分化过程中的)研究型人才培养的教育并驾齐驱,还需要假以时日;还需要政府教育主管部门的大力推进,需要人才需求市场的进一步完善发育,尤其需要高职教学单位及其直接相关部门肯于做长期的坚忍不拔的努力。新世纪高职高专教材编审委员会就是由全国 100 余所高职高专院校和出版单位组成的旨在以推动高职高专教材建设来推进高等职业教育这一变革过程的联盟共同体。

在宏观层面上,这个联盟始终会以推动高职高专教材的特色建设为己任,始终会从高职高专教学单位实际教学需要出发,以其对高职教育发展的前瞻性的总体把握,以其纵览全国高职高专教材市场需求的广阔视野,以其创新的理念与创新的运作模式,通过不断深化的教材建设过程,总结高职高专教学成果,探索高职高专教材建设规律。

在微观层面上,我们将充分依托众多高职高专院校联盟的互补优势和丰裕的人才资源优势,从每一个专业领域、每一种教材入手,突破传统的片面追求理论体系严整性的意识限制,努力凸现高职教育职业能力培养的本质特征,在不断构建特色教材建设体系的过程中,逐步形成自己的品牌优势。

新世纪高职高专教材编审委员会在推进高职高专教材建设事业的过程中,始终得到了各级教育主管部门以及各相关院校相关部门的热忱支持和积极参与,对此我们谨致深深谢意,也希望一切关注、参与高职教育发展的同道朋友,在共同推动高职教育发展、进而推动高等教育体制变革的进程中,和我们携手并肩,共同担负起这一具有开拓性挑战意义的历史重任。

新世纪高职高专教材编审委员会

2001 年 8 月 18 日

前 言

《单片机原理及应用》是新世纪高职高专教材编审委员会组编的电气自动化技术类课程规划教材之一。

本教材在编写的过程中力求突出以下特色:

1. 全书以 MCS-51 系列单片机为主线,介绍了 MCS-51 单片机的结构和工作原理,详细介绍了 MCS-51 单片机的编程语言和程序设计方法,通过典型应用实例进一步介绍了单片机的系统扩展和工程设计方法。

2. 在编写过程中,编者根据多年来不同专业的单片机应用技术教学经验,结合高职高专教育的教学特点,注意内容的实用性,突出应用能力的培养,力争做到语言精练,内容深入浅出、通俗易懂。

3. 本教材通过一些典型实验进一步阐述单片机的应用技巧。每章都附有本章知识要点、本章小结、思考题与习题,便于学生明确学习目标、巩固所学知识。

全书共分为 10 章和 1 个附录,内容主要包括微型计算机的基础知识、MCS-51 单片机的结构和原理、指令系统、汇编语言程序设计、中断系统及定时/计数器、单片机串行通讯及接口、单片机系统扩展、A/D 与 D/A 转换的接口技术、单片机系统的工程设计、基于 C 语言的单片机程序设计和实验指导等。

本教材由田亚娟任主编,杨志邦、宁玉珊、朱海洋任副主编,另外,黄春平、欧阳明星也参加了本教材的编写。具体编写分工如下:田亚娟编写第 1、6、9 章和附录;宁玉珊编写第 3 章;朱海洋编写第 2、8 章;黄春平编写第 4 章和附录的部分内

容;杨志邦编写第5章;欧阳明星编写第7、10章。田亚娟对本教材的编写思路与大纲进行了总体策划,指导全书的编写,负责统稿和定稿。秦皇岛职业技术学院的常建刚老师和安徽水利水电职业技术学院的何强老师审阅了全书,并提出了许多宝贵的意见和建议,在此表示衷心的感谢!

由于编者水平有限,书中难免存在错误或不妥之处,热忱欢迎读者对本教材提出批评与建议。

所有意见和建议请发往:gzjckfb@163.com

欢迎访问我们的网站:<http://www.dutpgz.cn>

联系电话:0411-84707492 84706104

编者
2008年9月



第 1 章 绪论	1
1.1 数制与编码	1
1.2 计算机概述	8
1.3 单片机的发展过程与应用领域	10
本章小结	15
思考题与习题	15
第 2 章 MCS—51 单片机的结构和原理	16
2.1 MCS—51 单片机结构	16
2.2 MCS—51 单片机的存储器组织	20
2.3 MCS—51 单片机的并行端口结构与操作	26
2.4 MCS—51 单片机的引脚及其功能	29
2.5 MCS—51 单片机的时序和工作方式	30
本章小结	34
思考题与习题	35
第 3 章 MCS—51 单片机的指令系统	36
3.1 指令系统概述	36
3.2 MCS—51 单片机的寻址方式	38
3.3 数据传送类指令	42
3.4 算术运算类指令	45
3.5 逻辑运算类指令	49
3.6 控制转移类指令	52
本章小结	58
思考题与习题	58
第 4 章 MCS—51 单片机的程序设计	61
4.1 汇编语言程序设计基础	61
4.2 顺序程序设计	65
4.3 分支程序设计	67
4.4 循环程序设计	69
4.5 子程序设计	75

4.6 算术运算和数码转制程序设计·····	78
本章小结·····	85
思考题与习题·····	85
第5章 中断系统及定时/计数器·····	87
5.1 MCS—51 单片机的中断系统·····	87
5.2 中断系统的编程和应用举例·····	95
5.3 MCS—51 单片机的定时/计数器·····	100
5.4 定时/计数器应用举例·····	106
本章小结·····	110
思考题与习题·····	110
第6章 单片机串行通信及接口·····	112
6.1 串行通信基础·····	112
6.2 MCS—51 单片机的串行接口·····	114
6.3 串行口应用举例·····	119
本章小结·····	129
思考题与习题·····	129
第7章 单片机系统扩展·····	131
7.1 概述·····	131
7.2 存储器扩展·····	133
7.3 简单 I/O 口扩展·····	138
7.4 可编程 I/O 接口芯片 8155·····	141
7.5 MCS—51 单片机的键盘接口·····	146
7.6 MCS—51 单片机的显示接口·····	150
本章小结·····	161
思考题与习题·····	161
第8章 A/D 和 D/A 转换的接口技术·····	162
8.1 概述·····	162
8.2 D/A 转换器及其与单片机的接口·····	164
8.3 A/D 转换器及其与单片机的接口·····	169
本章小结·····	177
思考题与习题·····	177
第9章 单片机系统的工程设计·····	179
9.1 MCS—51 单片机应用系统设计概述·····	179
9.2 单片机控制的温度采集系统·····	183
9.3 单片机控制的纸机转速系统·····	189

9.4 步进电机控制应用实例	194
本章小结	197
思考题与习题	197
第 10 章 基于 C 语言的单片机程序设计	199
10.1 概 述	199
10.2 C51 语言语法要素	201
10.3 C51 存储器类型与存储器模式	210
10.4 C51 程序设计实例	215
本章小结	224
思考题与习题	225
附录	226
参考文献	250

第 1 章

绪 论

本章知识要点

本章主要讲解计算机的数制、编码和计算机中数的表示方法,然后介绍计算机的产生和发展、计算机的基本结构、单片机的基本组成、单片机的应用及 MCS-51 系列单片机。包含的知识点如下:

1. 二进制数、十进制数和十六进制数之间的换算关系。
2. 二进制数原码、反码和补码的表示方法;BCD 码和 ASCII 码的基本概念。
3. 计算机的产生、发展及基本结构。
4. 单片机的发展过程及其应用领域。

1.1 数制与编码

计算机是用数字电路进行数据处理的,数字电路作数据处理是基于二进制数的进位计数制,而人们习惯于十进制数。因此,学习计算机首先要掌握计算机中用到的计数方法、数的表示方法和运算方法。

1.1.1 数制及其转换

习惯用的十进制数,是 10 个不同的数字 0,1,2…9 依据“逢十进一”的规则进行计数和运算的。

十进制数中,各位数字表示的值与数字本身和数字所在的位置有关,数字所处的位置不同,代表的数的大小也就不同。从右起第一位是个位,第二位是十位,第三位是百位……依次类推。个、十、百、千等在数字上叫做“权”,十进制数的权是以 10 为底的幂。某一位上的数字与该位的权的乘积就表示了该位数值的大小。任何一个十进制数都可表示为

$$\begin{aligned} D &= D_n D_{n-1} D_{n-2} \cdots D_1 D_0 D_{-1} \cdots D_{-m} \\ &= D_n \times 10^n + D_{n-1} \times 10^{n-1} + D_{n-2} \times 10^{n-2} + \cdots + D_1 \times 10^1 \\ &\quad + D_0 \times 10^0 + D_{-1} \times 10^{-1} + \cdots + D_{-m} \times 10^{-m} \\ &= \sum_{i=-m}^n (D_i \times 10^i) \end{aligned}$$

式中, D_i 是第 i 位的数字;10 为基数; n 、 m 为正整数。

计算机中要用到二进制数和十六进制数,除在运算中采用“逢二进一”和“逢十六进一”的规则外,其他与十进制数基本相同。

1. 二进制数及其转换

(1) 二进制数的表示

二进制数有两个不同的数字 0、1,基数是 2,依据“逢二进一”的规则。

例如,二进制数 11.01,为了区别于其他进制的数,加后缀“B”写成 11.01B 或 $(11.01)_2$ 。

二进制数 11.01B 按权展开成多项式如下

$$(11.01)_2 = 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (3.25)_{10}$$

十进制数的后缀为“D”(通常省略)或写成 $(3.25)_{10}$ 。

(2) 二进制数和十进制数的转换

① 二进制数转换成十进制数

方法:乘权求和法。

【例 1-1】 将 $(101.01)_2$ 转换为十进制数。

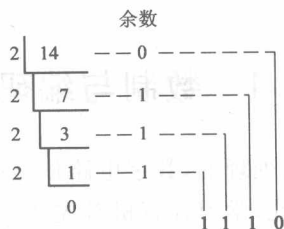
解: $(101.01)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (5.25)_{10}$

② 十进制整数转换成二进制数

方法:除 2 取余逆排法。即将十进制整数不断除以 2,并取余数,直到商为 0,所得余数从后往前读,即为等值的二进制数。

【例 1-2】 将 $(14)_{10}$ 转换成二进制数。

解:



可得转换结果: $(14)_{10} = (1110)_2$

③ 十进制小数转换成二进制小数

方法:乘 2 取整顺排法。即将十进制小数不断乘以 2,第一次乘以 2 所得整数为转换进制的最高小数系数,直到积为 0 或达到要求精度为止。

【例 1-3】 将 $(0.625)_{10}$ 转换成二进制小数。

解: 小数转换后仍为小数,转换过程如下:

$$\begin{array}{r} 0.625 \\ \times 2 \\ \hline 1.250 \\ \downarrow \\ 1 \end{array}$$

$$\begin{array}{r} 0.25 \\ \times 2 \\ \hline 0.50 \\ \downarrow \\ 0 \end{array}$$

$$\begin{array}{r} 0.50 \\ \times 2 \\ \hline 1.00 \\ \downarrow \\ 1 \end{array}$$

可得转换结果: $(0.625)_{10} = (0.101)_2$

2. 十六进制数及其转换

(1) 十六进制数的表示

十六进制数由数字 0~9 和字母符号 A~F 组成,分别表示十进制数的 0~15,基数为 16,依据“逢十六进一”的规则计数和运算。

例如,十六进制数 3AF.5E,其后缀为“H”写成 3AF.5EH 或 $(3AF.5E)_{16}$ 。十六进制数 3AF.5EH 按权展开成多项式如下:

$$(3AF.5E)_{16} = 3 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 + 5 \times 16^{-1} + 14 \times 16^{-2} = (943.367187)_{10}$$

(2) 十六进制数的转换

① 十六进制数转换成二进制数

方法:一拉四法。即将每 1 位十六进制数用相应的 4 位二进制数替代。

【例 1-4】 将 $(2BF.3C)_{16}$ 转换成二进制数。

解:转换过程如下:

$$\begin{array}{ccccccccc} & 2 & & B & & F & & 3 & & C \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 0010 & 1011 & 1111 & 0011 & 1100 \end{array}$$

可得转换结果: $(2BF.3C)_{16} = (001010111111.00111100)_2$

② 二进制数转换成十六进制数

方法:四合一法。以小数点为界,分别向左右每 4 位一组由相应的 1 位十六进制数替代,最后一组不足 4 位时,应用“0”补齐 4 位。

【例 1-5】 将 $(101111101110.101001)_2$ 转换成十六进制数。

解:转换过程如下:

$$\begin{array}{ccccccccc} \underline{(0)101} & & \underline{1111} & & \underline{0110} & & \underline{1010} & & \underline{01(00)} \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 5 & & F & & 6 & & A & & 4 \end{array}$$

可得转换结果: $(101111101110.101001)_2 = (5F6.A4)_{16}$

③ 十六进制数与十进制数之间的转换

十六进制数与十进制数之间的转换可以采用同二进制数与十进制数之间转换类似的方法,即十六进制数转换成十进制数采用“乘权求和法”;十进制数转换成十六进制数,整数部分采用“除 16 取余逆排法”;小数部分采用“乘 16 取整顺排法”,小数部分通常有误差,一般转换到所要求的精度即可。

由于 16 进制数的基数 16 比较大,做乘除法计算时较繁琐,所以一般十六进制数与十进制数之间的转换是通过二进制数间接进行的。

1.1.2 机器数与真值

前面介绍的二进制数没有提及符号,称为无符号数。带有正、负符号的数在数学中用“+”和“-”表示,但是,计算机不能,因为计算机中所有信息都是用二进制数表示的,符号也不能例外。在计算机中规定用最高位作为符号位,用“0”表示“+”;用“1”表示“-”。于是数的符号在计算机中被数码化了,即从表示形式上看符号位与数值位毫无区别。我们把机器中以编码形式表示的数称为机器数,而把原来一般书写形式表示的数称为真值。

例如,一个 8 位机器数与真值的关系可表示如下:

真值 $N_1 = +1011011B; N_2 = -1011011B$

机器数 $N_1 = 01011011B; N_2 = 11011011B$

若一个数的所有数位均为数值位,则该数为无符号数;若一个数的最高位为符号位而其他数位为数值位,则该数为有符号数。由此可见,对于同一个单元,存放的无符号数和有符号数所能表示的数值范围是不同的。以 8 位的存储单元为例,当它存放无符号数时,数的范围为 $0 \sim 255$;当它存放有符号数时,数的范围为 $-127 \sim 127$ 。

1.1.3 原码、反码、补码

在计算机中,机器数可以用三种方法表示:原码、反码和补码。

1. 原码

用原码表示带符号二进制数时,将最高位规定为符号位,其余位为数值位,符号位为“0”表示该数是正数;符号位为“1”表示该数是负数。

【例 1-6】 设 $x_1 = +1011011B, x_2 = -1011011B$,请分别写出它们在 8 位微型机中的原码形式。

解: $x_1 = +1011011B$ $[x_1]_{\text{原}} = 01011011B$

$x_2 = -1011011B$ $[x_2]_{\text{原}} = 11011011B$

用原码表示时,“0”这个数非常特别,它有“+0”和“-0”之分,因此其原码有两种表示形式: $[+0]_{\text{原}} = 00000000B; [-0]_{\text{原}} = 10000000B$

原码表示方法简单,容易读,但运算时必须把符号与数值分开。如加法运算,先要判断两个数的符号,相同可直接相加;符号不同则不能相加,而要用绝对值大的数减去绝对值小的数,其差值为两数和,而符号取自绝对值大的。由此可见,原码计算很不方便,计算机在运算中一般不采用原码,从而引出了反码和补码。

2. 反码

反码也是有符号二进制数的一种表示方法。正数的反码同原码;负数的反码,符号位为 1,数值位是原码的数值位按位取反。

【例 1-7】 设 $x_1 = +1011011B, x_2 = -1011011B$,请分别写出它们在 8 位微型机中的反码形式。

解: $x_1 = +1011011B$ $[x_1]_{\text{反}} = 01011011B$

$x_2 = -1011011B$ $[x_2]_{\text{反}} = 10100100B$

数“0”在反码中也可以写成两种形式: $[+0]_{\text{反}} = 00000000B; [-0]_{\text{反}} = 11111111B$

3. 补码

补码是计算机中有符号数的常用表示方法。正数的补码同原码,负数的补码是反码加 1。

【例 1-8】 设 $x_1 = +1011011B, x_2 = -1011011B$,请写出它们在 8 位微型机中的补码形式。

解: $x_1 = +1011011B$ $[x_1]_{\text{补}} = 01011011B$

$x_2 = -1011011B$ $[x_2]_{\text{补}} = 10100101B$

数“0”的补码表示形式是惟一的,即: $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000B$

有符号的 8 位二进制数对应的原码、反码、补码的表示及相应的十进制数对照见表 1-1。

表 1-1 数的几种表示方法对照表

十进制数	二进制数	原码	反码	补码
+0	+0000000	00000000	00000000	00000000
+1	+0000001	00000001	00000001	00000001
+2	+0000010	00000010	00000010	00000010
...	...	...	...	...
+126	+1111110	01111110	01111110	01111110
+127	+1111111	01111111	01111111	01111111
-0	-0000000	10000000	11111111	00000000
-1	-0000001	10000001	11111110	11111111
-2	-0000010	10000010	11111101	11111110
...	...	...	...	...
-126	-1111110	11111110	10000001	10000010
-127	-1111111	11111111	10000000	10000001
-128	-1000000	无法表示	无法表示	10000000

综上所述可知原码、反码和补码有如下特点：

(1) 三种编码的最高位为符号位，用“0”表示正，用“1”表示负。

(2) 对于正数，三种编码的表示方法是一样的。

(3) 对于负数，三种编码的符号位均为 1，数值部分不同。原码的数值部分与原数相同；反码的数值部分为原数按位取反；补码的数值部分为反码加 1。

(4) 8 位二进制数的原码、反码和补码所能表示的数值范围是不完全相同的。原码和反码为 $-127 \sim +127$ ，“0”有两种表示；补码为 $-128 \sim +127$ ，0 只有一种表示。

4. 补码的加减运算

在微型计算机中，原码表示的数易于人们识别，但运算时符号位往往需要单独处理。补码虽不易于识别，但运算方便。所以计算机在进行运算时采用补码形式。下面介绍计算机内部的运算过程及其结果分析。

(1) 补码的加法运算

在进行补码的加法运算时，不管两个数是正数还是负数，按补码的和等于和的补码进行。补码的加法运算的通式为

$$[x_1 + x_2]_{\text{补}} = [x_1]_{\text{补}} + [x_2]_{\text{补}}$$

上式要求 x_1 、 x_2 和 $(x_1 + x_2)$ 三个数必须都在 $-2^{n-1} \sim +2^{n-1}$ 范围内，否则机器便会产生溢出错误。在运算过程中，符号位和数值位一起参加运算，符号位的进位位略去不计。

【例 1-9】已知 $x_1 = +0011101\text{B} = +29$ ， $x_2 = -0000110\text{B} = -6$ ，求 $x_1 + x_2$ 。

解： $[x_1]_{\text{补}} = 00011101\text{B}$ ， $[x_2]_{\text{补}} = 11111010\text{B}$ ，则 $[x_1]_{\text{补}} + [x_2]_{\text{补}}$ 为

$$\begin{array}{r} 00011101 \\ + 11111010 \\ \hline 10001011 \end{array}$$

自然丢失

即 $[x_1 + x_2]_{\text{补}} = [x_1]_{\text{补}} + [x_2]_{\text{补}} = 00010111\text{B}$

真值为： $+0010111\text{B} = +23$ ，可见，符号在参加运算后结果是正确的。

(2) 补码的减法运算

在进行补码的减法运算时，可将两数之差的补码转化为两数补码之和计算。补码减法

的运算通式为

$$[x_1 - x_2]_{\text{补}} = [x_1]_{\text{补}} + [-x_2]_{\text{补}}$$

【例 1-10】 已知 $x_1 = +0001000\text{B} = +8$, $x_2 = +0001111\text{B}$, 求 $x_1 - x_2$ 。

解: $[x_1]_{\text{补}} = 00001000\text{B}$, $[-x_2]_{\text{补}} = 11110001\text{B}$, 则 $[x_1]_{\text{补}} + [-x_2]_{\text{补}}$ 为

$$\begin{array}{r} 00001000 \\ + 11110001 \\ \hline 11111001 \end{array}$$

即 $[x_1 - x_2]_{\text{补}} = [x_1]_{\text{补}} + [-x_2]_{\text{补}} = 11111001\text{B}$

真值为: $-0000111\text{B} = -7$

补码加、减法运算的特点如下:

① 求出参加运算的两个数的补码。

② 用补码相加进行加法运算, 用减数变补相加完成减法运算。

③ 运算时符号位应当作为数的一部分参与运算, 符号位有进位则丢弃。

④ 运算结果也为补码, 要得到真值还须转换。若结果的符号位为“0”, 表示正数, 它就是结果的原码形式; 若结果符号位为“1”, 表示结果为负数的补码, 负数补码求原码时, 符号位不变, 数值位按位取反后加“1”。

⑤ 补码运算时若结果超出了其表示范围, 就会产生“溢出”而出错。

需要说明的是, 计算机不会区别有符号数和无符号数, 只是按规定的法则进行计算。

【例 1-11】 求两个 8 位二进制数 10000010B 和 00011011B 的和。

解:

$$\begin{array}{r} 10000010 \\ + 00011011 \\ \hline 10011101 \end{array}$$

作为无符号数: $130 + 27 = 157$, 而作为有符号数则是: $(-126) + (+27) = -99$ 。

例 1-11 说明, 采用补码计算后, 不管作为无符号数还是有符号数, 运算结果都是正确的。因此, 计算机用补码运算时, 有符号数和无符号数是兼容的, 只是操作者要知道自己做的是有符号数计算还是无符号数计算。

1.1.4 BCD 码和字符的 ASCII 码

计算机只能识别“0”和“1”两个符号, 而计算机处理的信息却有多种形式, 例如数字、标点符号、运算符号、各种命令、文字和图形等。要表示这么多的信息并识别它们, 必须对这些信息进行编码。计算机中根据信息对象不同, 编码的方式也不同。常见的码制有 BCD 码和 ASCII 码等。

1. BCD 码(十进制数的二进制编码)

BCD 码是一种具有十进制权的二进制编码。BCD 码的种类较多, 常用的有 8421 码、2421 码、余 3 码和格雷码等。现以常用的 8421 码进行讨论。

(1) 8421 码的定义

8421 码因组成它的 4 位二进制数码的权为 8、4、2、1 而得名。用 4 位二进制数表示 1 位十进制数, 称为十进制数的二进制代码表示法。在此种表示法中, 十组 4 位二进制数分别代表了 0~9 中的 10 个数字符号。BCD(8421)码与十进制数、二进制数的对应关系见表 1-2。

表 1-2 BCD(8421)码与十进制数、二进制数的对应关系

十进制数	二进制数	8421 码	十进制数	二进制数	8421 码
0	0000	0000	8	1000	1000
1	0001	0001	9	1001	1001
2	0010	0010	10	1010	0001 0000
3	0011	0011	11	1011	0001 0001
4	0100	0100	12	1100	0001 0010
5	0101	0101	13	1101	0001 0011
6	0110	0110	14	1110	0001 0100
7	0111	0111	15	1111	0001 0101

(2)二—十进制调整

计算机在进行 BCD 码运算时,是按照二进制进位的,并不是“逢十进一”,因此运算结果可能会出错。计算机在进行 BCD 码加减运算时,必须对二进制加法的结果进行修正,使两个相邻的 BCD 码之间真正能够做到逢十进位。修正的原则是:

- ①进行 BCD 加法运算时:若和的低 4 位大于 9 或低 4 位向高 4 位发生进位,则低 4 位加 6 修正;若高 4 位大于 9 或高 4 位最高位发生进位,则高 4 位加 6 修正。
- ②进行 BCD 减法运算时:若低 4 位大于 9 或低 4 位向高 4 位有借位,则低 4 位减 6 修正;若高 4 位大于 9 或高 4 位最高位有借位,则高 4 位减 6 修正。

上面的修正是由机器内部的十进制调整电路根据专门的十进制调整指令自动完成的。

【例 1-12】 用 BCD 码完成的运算。

解:

0101 0100

+

0100 1000

1001 1100

+

0110

1010 0010

+

0110

0001 0000 0010

加 6 调整

高 4 位加 6 调整

因为低 4 位之和为 1100,大于 9 应做加 6 调整,调整后高 4 位为 1010,又大于 9,还应做一次加 6 调整,因此最后结果为 0001 0000 0010,即为 102,调整后结果正确。

2. ASCII 码(字符编码)

现代微型计算机不仅要处理数字信息,而且还需要处理大量字母和符号信息。这就需要对这些数字、字母和符号进行二进制编码,以供微型计算机识别、存储和处理。这些数字、字母和符号统称为字符,因此上述字母和符号的二进制编码又称为字符的编码。国际上通用的是美国标准信息交换码(American Standard Coded for Information Interchange),简称 ASCII 码。主要用于打印机、绘图机等外设与计算机之间传递信息。

ASCII 码是一种 8 位代码,最高位一般用于奇偶校验,用 7 位代码来对 128 个字符编码,其中 32 个是控制字符,96 个是图形字符,如附录 C 所示。附录 C 是 7 位 ASCII 码字符表,表中最高位未列出,一般表示时都以 0 来代替,而暂不考虑其奇偶检验功能。左边和上边为相应字符的 ASCII 码,其中上边为高三位二进制码,左边为低 4 位二进制码。如数字 0~9 的 ASCII 码表示为十六进制数的 30H~39H,字母 A~Z 的 ASCII 码为 41H~5AH。