



21 世纪高等院校电气信息类系列教材

Electrical Information •
Science and Technology

微型计算机原理与接口技术

第③版

张荣标 等编著



附赠电子教案

<http://www.cmpedu.com>



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

微型计算机原理与接口技术

第 3 版

张荣标 等编著



机械工业出版社

本书以 Intel 系列微处理器为背景,介绍了微型计算机原理与接口技术。全书以弄懂原理、掌握应用为编写宗旨,在内容安排上注重系统性、逻辑性、先进性与实用性。本书分三个部分:微型计算机原理部分(第 1、2、6 章),汇编语言程序设计部分(第 3、4、5 章),接口与应用部分(第 7、8、9、10 章)。根据 Intel 系列微处理器的向下兼容性,着重讲解了 16 位微型计算机的工作原理、指令系统、8086 汇编语言程序设计以及接口技术。考虑到目前 32 位 CPU 的广泛应用,又重点介绍了其代表芯片 80386 的工作原理,特别是 80386 的存储器管理技术。

为便于读者自学,本书在内容安排方面除附有一定的习题外,还增设了详细的习题例解。

本书可以作为高等院校电气信息类专业教材,也可供从事微型机系统设计和应用的技术人员自学和参考。

本书提供配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ:308596956,电话:010-88379753)。

图书在版编目(CIP)数据

微型计算机原理与接口技术/张荣标等编著. —3 版. 北京:机械工业出版社,2016.9
21 世纪高等院校电气信息类系列教材
ISBN 978-7-111-54454-8

I. ①微… II. ①张… III. ①微型计算机-理论-高等学校-教材②微型计算机-接口技术-高等学校-教材 IV. ①TP36

中国版本图书馆 CIP 数据核字(2016)第 179499 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:时 静 责任校对:张艳霞

责任印制:乔 宇

北京振兴源印务有限公司印刷

2016 年 8 月第 3 版·第 1 次印刷

184mm×260mm·25.25 印张·610 千字

0001-3500 册

标准书号:ISBN 978-7-111-54454-8

定价:59.00 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

服务咨询热线:(010)88379833

机工官网:www.cmpbook.com

读者购书热线:(010)88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面无防伪标均为盗版

金书网:www.golden-book.com

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、生物医学工程等子专业。这些子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,在语言描述上力求准确、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

第3版前言

本教材自出版以来,受到了广大师生的关爱,作为一本高等院校人才培养的教材,需要不断地更新和改进,所以从教和学的角度对原教材进行了必要的修订。这次修订主要目的是增加学生实验环节,以往微型计算机原理只有在接口技术学习后才会进入实验环节,第1到第5章学生一般很少有动手机会,其主要原因是相关的实验设施有限,难以实现边学习、边上机的互动教学方式。近年来,随着微型计算机的普及,学生普遍拥有笔记本电脑或台式机,充分利用这一有利条件,使学生从抽象思维中走出来,通过 DEBUG 软件使学生对微型计算机原理中的寄存器、存储器、标志位、数据、地址等信息有很好的感性认识。关于寻址方式、指令功能、汇编语言及编程等课程内容,目前大部分微型计算机原理教科书都是进行理论性的讲解,本教材旨在通过 DEBUG 软件使学生弄清楚这些课程内容在微型计算机中实现的具体过程,从原来看不见摸不着靠记忆的抽象思维转变成看得见有结果的实际操作,这样可大大提高学生学习微型计算机原理的兴趣和信心。因此,在这次修订的第3版教材中,努力创建教与学过程互动的新思路,增添课堂实验演示内容,实现边学习、边上机的互动教学模式。

对原教材进行了如下修订:第1章增添了工具软件操作的基本知识,为学生能在自己的PC上完成实验打好基础。第2章增添了微处理器寄存器、标志位及存储器的认知环节,采用DEBUG工具软件观察其中数据的现状与修改后的变化。第3章内容是8086CPU主要功能的体现,学生学习起来太抽象不容易理解,通过DEBUG调试软件上机实验,用操作前后的数据变化让学生理解各种操作数寻址方式实现的过程以及各类指令运行后所产生的作用。第4章增添了伪指令语句上机实验、表达式上机实验以及各例程序DEBUG调试操作,使学生学习汇编语言语法不再抽象和死记硬背,在操作中理解汇编语言语法的真正含义。第5章增添了汇编语言编程算法和结构程序实例的DEBUG调试操作,通过程序运行前后的数据变化让学生理解程序运行过程中各个环节的细微变化。第6~10章是8086CPU与外部器件进行信息传输的各种接口知识,在相关的接口实验课程中都有所安排,这次修订主要是顺应器件的市场变化,对部分内容做了必要的更新。

这次教材的修订由张荣标教授统稿,其中第6章和第10章由牛雪梅博士参与修订,其余由张荣标修订,教材中的第2~5章的所有程序及DEBUG调试部分已由作者的研究生朱丽虹、黄林奎等同学在微型计算机上验证通过。

由于时间仓促和作者水平有限,存在一些不足之处,敬请读者批评指正。

作 者

第2版前言

本教材自出版以来,受到了广大师生的关爱,作者在此表示衷心的感谢! 作为一本高等院校人才培养的教材,需要不断地更新和改进,所以我们从教和学的角度,对原教材进行了必要的改版。在改版中尽量保留原教材的特点,主要从精简教材的思路出发,对原教材进行了如下修正:

1. 列表说明,便于比较

对一些功能性的说明,采用表格形式列出,达到简洁明了的效果,教师可对其中的内容进行逐一解释,同时便于学生总结与比较,如对芯片引脚的介绍、标志寄存器位的说明等。

2. 归类合并,便于应用

从应用角度出发对接口章节进行归类合并,以便学生对相关知识的联想记忆,如中断系统与 8237A DMA 控制器,可编程并行接口芯片与串行通信技术,计数器/定时器与模拟量转换。

3. 更新内容,顺应发展

随着微处理器的不断更新换代,新增了微处理器最新发展动态。同时新增了模拟量转换内容,即 A-D 与 D-A 转换技术。

4. 提炼例解,强化典型

主要体现在习题例解和练习题的修改。使内容更加典型化,便于学生自学和教师对学生的考核,开拓学生的解题思路和提高教学的可操作性。

本书由张荣标教授统稿,其中第 6 章由陆文昌副教授编写,第 7 章由冯友兵博士编写,第 10 章由李岚副教授编写,其余各章由张荣标教授编写。

作者

第1版前言

微型计算机原理与接口技术是自动化、电气、电子信息以及其他电气信息类专业的一门重要专业基础课。随着微处理器技术的不断发展和用人单位对人才培养的更高要求,迫切需要一批适合新形势需要的相关教材。为此,本书作者参考现有教材,扬长避短,结合多年来一线教学的经验,并征求同行教师以及学生对微型计算机原理教材的要求,从教和学的角度出发,着手编写了本教材。与现有教材相比,本教材有如下特点:

1. 增设题解,便于自学

微型计算机原理与接口技术这门课内容多,课时少,除教师课堂上讲解外,学生必须花一定的时间复习和巩固已学过的知识。本书除编入一定量的习题外,还编入了习题例解,这在很大程度上减轻了该专业基础课的教学压力。

2. 面向实用,夯实基础

本教材侧重基础知识,用模型机讲解 CPU 的工作原理,以 8086 CPU 为背景,系统地讲解了 16 位微型计算机的工作原理。考虑到目前 32 位 CPU 的广泛应用,又重点介绍了其代表芯片 80386 的工作原理,特别是 80386 的存储器管理技术。这样,可以使学生从基本原理出发,把握先进技术。

3. 力求图示,方便理解

本教材尽可能采用图示的方法,让学生有一种感性认识。如介绍微型计算机系统时,采用实物图片,使学生对微机有一种实实在在的感觉,激发出对微型计算机原理学习的兴趣;在讲解指令的寻址过程中,采用示意图的方式,使学生一目了然。

4. 条理清晰,便于领会

本教材中通篇都贯穿了“条理清晰”这一特点,学生比较容易掌握要点。

5. 突出重点,详解难点

从学生实际应用出发,在掌握了必要的基础知识情况下,将重点放在汇编语言编程和接口技术的学习上,这些内容也是学生学习的难点。

全书由张荣标教授统稿,其中第 6 章由陆文昌副教授编写,其余各章由张荣标教授编写。书中的汇编语言程序已由作者的研究生冯友兵、李华、章云峰、陈相朝等同学在计算机上验证通过。本书还得到了赵德安教授、李岚博士的大力支持。同时对参与书稿录入和整理工作的硕士研究生们表示感谢。

由于作者水平有限,存在一些不足之处,恳请读者批评指正。

目 录

出版说明

第3版前言

第2版前言

第1版前言

第1章 微型计算机基础	1
1.1 计算机中的数制与码制	1
1.1.1 计算机中的数制	1
1.1.2 计算机中的码制及补码运算	3
1.1.3 计算机中的小数点问题	7
1.1.4 计算机中信息的编码	8
1.2 微型计算机的组成	10
1.3 计算机的基本工作原理	13
1.3.1 模型计算机	13
1.3.2 程序运行过程	16
1.4 汇编语言上机工具软件	21
1.5 习题例解	24
1.6 练习题	27
第2章 80x86 微处理器	29
2.1 Intel 公司微处理器发展概述	29
2.2 8086 微处理器	31
2.2.1 8086 CPU 内部功能结构	31
2.2.2 8086 CPU 内部流水线管理工作原理	35
2.2.3 8086 CPU 的存储器组织	36
2.2.4 8086 CPU 寄存器与存储器 DEBUG 上机操作	38
2.2.5 8086 CPU 总线周期的概念	40
2.2.6 8086 CPU 的引脚信号及工作模式	40
2.2.7 8086 CPU 的操作时序	45
2.3 80286 微处理器	49
2.3.1 80286 CPU 的主要性能	49
2.3.2 80286 CPU 的功能结构	49
2.3.3 80286 CPU 的寄存器	50
2.3.4 80286 CPU 的存储器寻址	51
2.4 80386 微处理器	52
2.4.1 80386 CPU 的主要性能	52

2.4.2	80386 CPU 的功能结构	53
2.4.3	80386 CPU 的寄存器	54
2.4.4	80386 CPU 的存储器管理	57
2.4.5	80386CPU 寄存器与存储器 DEBUG 上机操作	64
2.5	80486 微处理器	65
2.5.1	80486 CPU 的主要性能	65
2.5.2	80486 CPU 的功能结构	66
2.6	习题例解	67
2.7	练习题	69
第3章	寻址方式与指令系统	71
3.1	数据类型及其存储规则	71
3.1.1	基本数据类型及其存储	71
3.1.2	数字数据类型	72
3.1.3	指针数据类型	72
3.1.4	字符串、位及位串数据类型	72
3.2	计算机指令格式	73
3.2.1	指令的助记符格式	73
3.2.2	80x86 指令编码格式	74
3.3	8086 CPU 的寻址方式	75
3.3.1	操作数的寻址方式	75
3.3.2	指令地址的寻址方式	80
3.3.3	寻址方式的 DEBUG 上机实验	82
3.4	8086 指令系统	83
3.4.1	数据传送类指令	83
3.4.2	算术运算类指令	91
3.4.3	逻辑运算和移位指令	97
3.4.4	串操作指令	100
3.4.5	控制转移类指令	105
3.4.6	处理器控制类指令	110
3.5	80x86 的寻址方式及新增的指令	112
3.5.1	虚地址方式下的寻址方式	112
3.5.2	80286 CPU 新增指令	116
3.5.3	80386/80486 CPU 新增指令	119
3.6	习题例解	122
3.7	练习题	125
第4章	汇编语言语法和 DOS 功能调用	128
4.1	汇编语言中的基本数据	128
4.2	伪指令语句	129
4.2.1	数据定义伪指令语句	129

4.2.2	标识符赋值与解除伪指令语句	130
4.2.3	段定义伪指令语句	132
4.2.4	过程定义伪指令语句	134
4.2.5	程序开始与结束伪指令语句	134
4.2.6	方式定义伪指令语句	136
4.2.7	结构定义伪指令语句	136
4.2.8	分组伪指令语句	139
4.2.9	其他伪指令语句	139
4.2.10	伪指令语句上机实验	141
4.3	汇编语言中的表达式	144
4.3.1	算术运算符	144
4.3.2	逻辑运算符	146
4.3.3	关系运算符	146
4.3.4	分析运算符	146
4.3.5	修改属性运算符	148
4.3.6	汇编语言中的表达式上机实验	149
4.4	指令语句	151
4.5	宏指令语句及其使用	153
4.6	DOS 系统功能调用	158
4.7	习题例解	162
4.8	练习题	165
第5章	汇编语言程序设计	168
5.1	顺序结构程序设计	168
5.2	分支结构程序设计	171
5.2.1	二分支结构	172
5.2.2	多分支结构	174
5.3	循环结构程序设计	177
5.3.1	循环程序的组成与结构形式	177
5.3.2	循环程序的控制方法	178
5.3.3	多重循环程序设计	184
5.4	子程序结构程序设计	187
5.4.1	子程序的定义与调用	187
5.4.2	子程序的参数传送	189
5.4.3	子程序嵌套与递归调用	199
5.5	模块化程序设计	203
5.6	习题例解	206
5.7	练习题	212
第6章	存储器	214
6.1	概述	214

6.1.1 半导体存储器的分类	214
6.1.2 半导体存储器的性能指标	215
6.1.3 半导体存储器的一般结构及组成	216
6.2 随机存取存储器 RAM	219
6.2.1 静态 RAM	219
6.2.2 动态 RAM	222
6.2.3 RAM 存储容量的扩展方法	223
6.2.4 RAM 存储器与 CPU 的连接	225
6.3 只读存储器 ROM	226
6.3.1 只读存储器的结构	226
6.3.2 只读存储器的分类	227
6.3.3 PROM 基本存储电路	228
6.3.4 典型 EEPROM 芯片	228
6.4 高速缓存存储器 Cache	229
6.4.1 Cache 的基本结构和工作原理	229
6.4.2 Cache 存储器组织	231
6.5 几种新型的半导体存储器	234
6.6 习题例解	236
6.7 练习题	238
第7章 中断系统与 8237A DMA 控制器	240
7.1 中断系统概述	240
7.1.1 中断的概念及其作用	240
7.1.2 中断处理系统	241
7.2 8086 CPU 中断系统	244
7.2.1 8086 CPU 的中断源	244
7.2.2 8086 CPU 的中断响应过程	246
7.2.3 中断向量表	248
7.2.4 中断程序设计	250
7.3 中断控制器 Intel 8259A	251
7.3.1 8259A 的引脚信号及结构	251
7.3.2 8259A 的工作方式	254
7.3.3 8259A 的编程	256
7.3.4 8259A 的应用举例——在 IBM PC/XT 中的应用	260
7.3.5 8086 中断响应总线周期操作	262
7.4 可编程 DMA 控制器 8237A	263
7.4.1 8237A 的特性与结构	263
7.4.2 内部寄存器	265
7.4.3 8237A 的工作时序	270
7.4.4 8237A 的初始化编程与应用举例	271

7.5 习题例解	273
7.6 练习题	277
第8章 输入/输出接口基础与总线	279
8.1 概述	279
8.1.1 外部设备及其信号	279
8.1.2 输入/输出接口的功能	280
8.2 CPU 与端口之间的接口技术	281
8.2.1 最常用的简单输入/输出接口芯片	281
8.2.2 端口的编址方式	283
8.2.3 端口与 CPU 之间的接口	285
8.3 CPU 与端口之间的数据传送方式	287
8.3.1 程序控制传送方式	287
8.3.2 中断技术传送方式	291
8.3.3 DMA 传送方式	292
8.4 总线技术	294
8.4.1 概述	294
8.4.2 PC 总线	296
8.4.3 ISA 总线	297
8.4.4 PCI 总线	299
8.5 习题例解	302
8.6 练习题	306
第9章 可编程并行接口芯片与串行通信技术	308
9.1 可编程并行接口芯片 8255A	308
9.1.1 8255A 的结构	308
9.1.2 方式选择	310
9.1.3 各方式的功能	312
9.1.4 端口 C 的状态字	316
9.1.5 8255A 应用举例	317
9.2 串行通信	318
9.2.1 串行通信的数据传送方向	318
9.2.2 串行通信的异步与同步通信方式	319
9.2.3 波特率及收发端的同步	321
9.2.4 常用串行接口介绍	323
9.3 可编程串行通信接口芯片 8251A	329
9.3.1 8251A 内部结构和外部引脚	330
9.3.2 8251A 编程	334
9.3.3 8251A 的应用	337
9.4 习题例解	340
9.5 练习题	345

第 10 章 计数器/定时器与模拟量转换	347
10.1 可编程计数器/定时器	347
10.1.1 可编程计数器/定时器的基本工作原理	347
10.1.2 8253 的内部结构及引脚	349
10.1.3 8253 的控制字	350
10.1.4 8253 的工作方式	352
10.1.5 8253 的应用举例	356
10.2 数 - 模转换	357
10.2.1 概述	357
10.2.2 D - A 转换芯片	358
10.2.3 D - A 转换器的应用	360
10.3 模 - 数转换	361
10.3.1 概述	362
10.3.2 典型的 A - D 转换芯片	363
10.3.3 AD574 转换器应用实例	366
10.4 习题例解	368
10.5 练习题	370
附录	372
附录 A 8086 指令表	372
附录 B 伪操作指令表	382
附录 C DOS 功能调用表(INT 21H)	385
附录 D DEBUG 命令表	389
参考文献	391

第1章 微型计算机基础

随着微处理器制造技术的不断发展,计算机的结构越来越复杂,功能越来越强大,性能越来越优越,计算机原理所涉及的内容也就越来越多,但是计算机基本原理没有改变,只要对计算机的基础知识有充分的了解,就能从容地面对计算机日新月异的变化。

本章详细讲解了计算机中数(大小、符号、小数点)及字符信息的表达问题;描述了微处理器、微型计算机和微型计算机系统三者之间的关系;通过模型计算机阐述了计算机的基本工作原理;系统地概述了汇编语言上机工具软件,为微型计算机原理的学习奠定基础。

1.1 计算机中的数制与码制

数是客观事物的量在人们头脑中的反映。一个“量”相同的数可以用不同的计数制度来表示,这就形成了不同的数制。表达一个数的大小和正负的不同方法叫作码制。

1.1.1 计算机中的数制

1. 数的位置表示法

数制是人们按某种进位规则进行计数的科学方法。位置表示法是表示数的常用方法。在数的位置表示法中,基数取值不同便可得到不同进位制的表达式。设待表示的数为 N , 则

$$N = \sum_{i=-m}^{n-1} a_i X^i \quad (1-1)$$

式中, X 为基数; a_i 为系数 ($0 < a_i < X - 1$); m 为小数位数; n 为整数位数。

在计算机中常用的数制有二进制、八进制、十六进制和十进制,相应的 X 取值为 2、8、16、10。在计数或加法运算过程中,它们分别是逢二进一、逢八进一、逢十六进一和逢十进一。在减法运算过程中,它们分别是借一当二、借一当八、借一当十六和借一当十。在数的位置表示法中,它们的后缀分别是 B、Q、H 和 D(或省略不写)。

下面用例题来说明用二进制、八进制、十六进制表示数的结果(十进制数)。

【例 1-1】 (1) 二进制数

$$10011.11\text{B} = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 19.75$$

(2) 八进制数

$$7345.6\text{Q} = 7 \times 8^3 + 3 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 + 6 \times 8^{-1} = 3813.75$$

(3) 十六进制

$$4\text{AC6H} = 4 \times 16^3 + 10 \times 16^2 + 12 \times 16^1 + 6 \times 16^0 = 19142$$

人们习惯使用的是十进制,而计算机中采用的基本数制是二进制(八进制和十六进制作为二进制的一种便于表示的编写形式)。这是为什么呢?采用二进制数,不仅因为它只有 0 和 1 两个系数,还因为 0 和 1 用电路实现起来很方便。即数字电路中通常的两种稳态:逻辑器件的饱和与截止状态。相应地形成低电位和高电位,以此代表两个数码:0 和 1。计算机通常

用高电位代表1,低电位代表0。采用这样精心设计的电路系统,既简单又快捷。

2. 数制之间的转换

(1) 任意进制数转换为十进制数

对二进制、八进制和十六进制以及任意进制数转换为十进制数可采用表达式(1-1)展开求和实现,详见例1-1。

(2) 二进制、八进制和十六进制数之间转换

一位八进制数相当于三位二进制数;一位十六进制数相当于四位二进制数。它们之间的转换十分方便。

【例1-2】 二进制转换成八进制和十六进制数。

$$\begin{aligned} 1101100101100011\text{B} &= 154543\text{Q} \\ &= \text{D963H} \end{aligned}$$

(3) 十进制数转换为二进制数

当十进制数转换为二进制数时,需要将整数部分和小数部分分开。整数常采用“除2取余法”,而小数则采用“乘2取整法”。需提及的是十进制小数并不是都能用有限的二进制小数精确地表示,此时要根据精度的要求来确定被转换的二进制位数。

1) 十进制整数转换为二进制整数。转换方法是除2取余,直到商等于零为止,逆序排列余数即可。对数值比较大的十进制数进行转换时,可采用先将十进制整数转换为十六进制整数,然后再将十六进制整数转换为二进制整数。十进制整数转换为十六进制整数的方法是除16取余,直到商等于零为止,逆序排列余数。

【例1-3】 将十进制数19,3910分别转换为相对应的二进制数。

解

2	19			
2	9	商为9, 余数为 1	↑	低位
2	4	商为4, 余数为 1		
2	2	商为2, 余数为 0		
2	1	商为1, 余数为 0		
2	0	商为0, 余数为 1		高位
16	3910			
16	244	商为244, 余数为 6	↑	低位
16	15	商为 15, 余数为 4		
16	0	商为 0, 余数为 15(F)		高位

转换结果分别为 $19\text{D} = 10\ 011\text{B}$; $3910\text{D} = \text{F46H} = 111101000110\text{B}$ 。

2) 十进制小数转换为二进制小数。转换方法是将小数部分乘2取整,直到乘积的小数部分等于零为止(若永不为零则根据精度要求截取一定的位数),顺序排列每次乘积的整数部分即可。

【例1-4】 将十进制数19.8125转换为二进制数。

解 整数部分可由例1-3的结果得

$$19\text{D} = 10011\text{B}$$

小数部分 0.8125D 的转换过程:

0.8125D × 2 = 1.625	得小数部分为 0.625	整数部分为 1	↓ 高位 低位
0.625D × 2 = 1.25	得小数部分为 0.25	整数部分为 1	
0.25D × 2 = 0.5	得小数部分为 0.5	整数部分为 0	
0.5D × 2 = 1.0	得小数部分为 0	整数部分为 1	

转换结果为 19.8125D = 10011.1101B。

1.1.2 计算机中的码制及补码运算

一个数除了有量的大小之分外还有正负的区别。为了处理数的符号问题,在计算机中引进了码制的概念。通常用二进制数的最高位来表示数的符号位。常用的码制有原码、补码、反码及偏移码。

1. 原码

用二进制数的最高位表示数的符号,通常规定以 0 表示正数,1 表示负数,其余各位表示数值本身,则称该二进制数为原码表示法。

设机器字长为 n ,数 X 的原码为 $[X]_{\text{原}}$,则原码的定义如下。

$$[X]_{\text{原}} = \begin{cases} X & 0 \leq X \leq 2^{n-1} - 1 \\ 2^{n-1} + |X| & -(2^{n-1} - 1) \leq X \leq 0 \end{cases} \quad (1-2)$$

【例 1-5】 设机器字长为 $n=8$ 时,试求 +0、+6、+127、-0、-6、-127 的原码。

$$\begin{aligned} \text{解 } [+0]_{\text{原}} &= 00000000 & [-0]_{\text{原}} &= 10000000 \\ [+6]_{\text{原}} &= 00000110 & [-6]_{\text{原}} &= 10000110 \\ [+127]_{\text{原}} &= 01111111 & [-127]_{\text{原}} &= 11111111 \end{aligned}$$

由此可见,原码是把符号数值化了的数,在计算机中称为机器数。对正数来说,原码与相应的二进制数完全相同;对负数来说,二进制数的最高位一定是“1”,其余各位是该数的绝对值。零的原码表示有正零和负零之分。原码表示法最大优点是简单直观,但不便于加减运算。原码数的运算完全类同于正负数的笔算。例如,两个数相减,先比较两个数绝对值的大小,然后绝对值大的数减去绝对值小的数,最后在结果前面加上原来绝对值较大的数的符号。因而,处理过程非常烦琐,要求计算机的结构也极为复杂。

2. 反码

设机器字长为 n ,数 X 的反码为 $[X]_{\text{反}}$,则反码的定义如下。

$$[X]_{\text{反}} = \begin{cases} X & 0 \leq X \leq 2^{n-1} - 1 \\ 2^n - 1 + X & -(2^{n-1} - 1) \leq X \leq 0 \end{cases} \quad (1-3)$$

【例 1-6】 设机器字长为 $n=8$ 时,试求 +0、+6、+127、-0、-6、-127 的反码。

$$\begin{aligned} \text{解 } [+0]_{\text{反}} &= 00000000 & [-0]_{\text{反}} &= 11111111 \\ [+6]_{\text{反}} &= 00000110 & [-6]_{\text{反}} &= 11111001 \\ [+127]_{\text{反}} &= 01111111 & [-127]_{\text{反}} &= 10000000 \end{aligned}$$

可以看出,正数的反码与相应的原码完全相同,负数的反码只需把相应的绝对值按位求反即可。用反码来表示负数现已较少采用。

3. 补码

原码和反码都不能方便运算,是否存在另一种数的表示法能方便运算呢? 回答是肯定的,

那就是补码表示法。在数的原码和反码表示法中,参加运算的数的符号是不能参加运算的,而在数的补码表示法中,参加运算的数的符号与数一样,也可以参加运算,并且使减法运算变成加法运算,省去一套减法电路。因此,采用补码运算使计算机的结构大为简化。

补码为什么具有这种功能,为了说明补码的概念,可从时钟校准谈起。若现在是北京时间1点整,而时钟快了两小时,时针指在3点上,要将时钟校准有两种方法:一种是把时钟倒拨两小时,相当于作减法运算;另一种是顺时针拨10小时,相当于作加法运算,则钟面上两种方法得到的结果是一样的。

$$3 + 10 = 1 \quad (\text{时针经过 12 点时自动丢失一个数 12})$$

$$3 - 2 = 3 + (-2) = 1$$

这里把减法运算变成了加法运算,其中10与-2到底有什么关系,自动丢失的一个数12又是什么,这是学习补码概念的关键。数学上把12这个数叫作“模”,10是(-2)对模12的补码。这样,在模12的条件下,负数就可以转化为正数,而正负数相加也就可以转化为正数间的相加。

再考虑计算机运算的特点,计算机中的部件都有固定的位数,假定位数为 n ,则计算机中最大的计数值(包括符号位)为 $2^n - 1$,当大于等于 2^n 时同样会自动丢失一个数 2^n 。因此,计算机中的负数可以表示成以 2^n 为模的补码。这样可以得到计算机中二进制补码的定义。

设机器字长为 n ,数 X 的补码为 $[X]_{\text{补}}$,则补码的定义如下。

$$[X]_{\text{补}} = \begin{cases} X & 0 \leq X \leq 2^{n-1} - 1 \\ 2^n + X & -2^{n-1} \leq X \leq 0 \end{cases} \quad (1-4)$$

【例1-7】 设机器字长为 $n=8$ 时,试求+0、+6、+127、-0、-6、-127的补码。

$$\begin{aligned} \text{解} \quad [+0]_{\text{补}} &= 00000000 & [-0]_{\text{补}} &= 00000000 \\ [+6]_{\text{补}} &= 00000110 & [-6]_{\text{补}} &= 11111010 \\ [+127]_{\text{补}} &= 01111111 & [-127]_{\text{补}} &= 10000001 \end{aligned}$$

可以看出,正数的补码与相应的原码完全相同,负数的补码只需把相应的绝对值按位求反并在末位加1即可。如果从负数的原码来求补码,其规则是原码的符号位不变,其余各位按位求反并在末位加1。一般情况下,只要按上述规则求取补码即可,不需要用补码的定义来求。

4. 偏移码

偏移码主要用于模/数转换过程中,若被转换数需参加运算,则仍要转换为补码。

设机器字长为 n ,数 X 的偏移码为 $[X]_{\text{移}}$,则偏移码的定义如下。

$$[X]_{\text{移}} = 2^{n-1} + X \quad (1-5)$$

【例1-8】 设机器字长为 $n=8$ 时,试求-128、0、+127的偏移码。

$$\begin{aligned} \text{解} \quad [-128]_{\text{移}} &= 00000000 \\ [0]_{\text{移}} &= 10000000 \\ [+127]_{\text{移}} &= 11111111 \end{aligned}$$

可以看出,偏移码是把相应的补码在数轴上向右平移了 2^{n-1} ,从而弥补了补码不直观的缺点,用偏移码表示数的大小可以说是一目了然。注意,它仅是在数轴上平移了 2^{n-1} 个单位,使最小的负数变为0,最大的正数变为最大数111...11。但是,这里的数 2^{n-1} 与补码中模的概念