

21世纪大学计算机基础规划教材

微机原理 与汇编语言程序设计

刘永华 王成端 主 编
张秀芝 周建梁 陈子富 副主编

- 较完整地阐述微型计算机的基本概念、基本组成和基本工作原理，侧重对常用接口芯片应用的介绍。
- 兼顾完整性，注重实用性，大量例题的讲解可使学生掌握设计接口电路和编写接口程序的方法。
- 跟踪新技术，保持先进性，使教材能反映计算机技术的最新发展水平，适应社会的需求。
- 内容重点突出，层次分明，对应用性较强的内容进行重点描述，对实际使用较少的内容仅做简单介绍。



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪大学计算机基础规划教材

微机原理与 汇编语言程序设计

刘永华 王成端 主 编

张秀芝 周建梁 陈子富 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书融入作者多年教学、科研和实践的经验与体会,以培养学生应用能力为主线,理论与实践相结合,并注意反映计算机技术的最新发展情况。全书共分13章,主要包括微型计算机技术概述、微机系统与微处理器、微机指令系统、汇编语言、存储器接口技术、并行接口技术及应用、串行通信技术及应用、定时/计数器技术及应用、中断处理技术及应用、DMA技术及应用、模拟接口技术及应用、总线技术以及综合应用等内容。

本书可作为普通高校计算机及电子信息类专业本科教材,也可作为高职高专计算机应用及相关专业教材,还可作为工程技术人员学习的参考书。

图书在版编目(CIP)数据

微机原理与汇编语言程序设计/刘永华,王成端主编.

北京:中国铁道出版社,2006.5

21世纪大学计算机基础规划教材

ISBN 7-113-07124-4

I. 微... II. ①刘... ②王... III. ①微型计算机—理论—高等学校—教材②汇编语言—程序设计—高等学校—教材 IV. ①TP36②TP313

中国版本图书馆CIP数据核字(2006)第056818号

书 名: 微机原理与汇编语言程序设计

作 者: 刘永华 王成端 张秀芝 周建梁 陈子富

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 秦绪好

责任编辑: 苏 茜 崔晓静 王 丹

封面设计: 薛 为

封面制作: 白 雪

责任校对: 李 旸

印 刷: 河北省遵化市胶印厂

开 本: 787×1092 1/16 印张: 20.25 字数: 487千

版 本: 2006年7月第1版 2006年7月第1次印刷

印 数: 1~5 000册

书 号: ISBN 7-113-07124-4/TP·1864

定 价: 26.00元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前言

计算机应用能力是 21 世纪人才不可缺少的基本素质。目前, 微型计算机技术飞速发展, 其应用已涉及各个领域, 掌握微型计算机技术是计算机及电子信息类专业人才的基本要求。然而, 现今微型计算机技术及应用的教材大多内容比较陈旧, 最新的技术内容较少, 既不适合计算机专业本科生使用, 也不适合高职高专学生的教学要求。因此, 为培养技术应用型人才, 根据教育部规划教材编写的指导思想与原则要求, 我们编写了本书。

本书可以帮助学生掌握微型计算机的硬件组成及应用, 学会运用指令系统和汇编语言进行程序设计, 熟悉各种类型的接口技术及应用, 树立起微型计算机体系结构的基本概念。

全书以培养学生应用能力为主线, 理论与实践相结合, 其主要特色如下。

- 较完整地阐述微型计算机的基本概念、基本组成和基本工作原理, 对常用接口芯片的介绍侧重应用。
- 兼顾完整性, 注重实用性。对常用接口芯片加大了应用实例的介绍, 书中运用大量例题使学生在学完后能动手设计接口电路和编写接口程序。
- 跟踪新技术, 保持先进性。在现有教材的基础上, 适当增删内容, 突出重点, 使教材能反映计算机技术的最新发展水平, 适应社会的需求。
- 内容重点突出, 层次分明。书中并未对微机接口技术的所有内容泛泛而谈, 而是针对教学对象的特点进行内容的合理分配。由浅入深, 循序渐进, 对应用性较强的内容进行重点描述, 而实际使用较少的内容做简单介绍, 既提高了应用能力, 又开拓了知识面。例题注意综合性与设计性相结合, 使学生能够学以致用。

全书内容共分 13 章。第 1 章主要介绍数据表示与运算、码制和计算机语言; 第 2 章主要介绍微计算机系统与微处理器, 包括微计算机系统组成和工作原理、8086/8088 微处理器以及 Pentium 系列微处理器; 第 3 章主要介绍 8086/8088 指令系统; 第 4 章主要介绍汇编语言语句、伪指令和汇编语言程序的结构; 第 5 章主要介绍 I/O 端口的常用译码方法, SRAM、DRAM、ROM 的存储容量扩展以及与 CPU 的连接; 第 6 章主要介绍并行接口的基本概念、并行接口组成、数据传送方式和 8255A 的内部结构、8255A 的编程方法及应用举例; 第 7 章主要介绍串行通信的基本概念、接口标准、IBM-PC/XT 机异步通信接口技术及应用; 第 8 章主要介绍定时/计数器的基本概念和分类, 以 8253-5 为例, 介绍了其内部结构、各种工作方式和初始化编程方法, 结合实例应用编程; 第 9 章主要介绍中断的基本概念与中断系统的结构, 并以 8259A 中断控制器为例介绍了其内部结构、特性、命令字、中断操作、初始化编程及应用举例等; 第 10 章主要介绍 DMA 的基本概念、8237A 芯片结构、初始化编程和 8237A 的应用举例; 第 11 章主要介绍 D/A 转换原理、D/A 转换器性能指标、D/A 转换芯片及其接口、A/D 转换原理、A/D 转换器性能指标、A/D 转换芯片及其接口、多路模拟开关及采样保持电路等, 并举例说明了数据采集系统的设计方法; 第 12 章主要介绍总线的基本概念和分类, 对常用的系统总线和外部总线也作了系统的介绍; 第 13 章主要介绍微机接口技术应用系统的硬件设计、软件设计与总体设计。

本课程建议教学计划安排 72 学时, 实验与实践教学内容单独安排。

本书由刘永华、王成端主编, 张秀芝、周建梁、陈子富副主编。其中第 1 章~第 3 章由

王成端编写，第4章和第5章由周建梁编写，第6章～第8章由张秀芝编写，第9章由陈子富编写，第10章～第13章由刘永华编写，王宇晓、王丰、李晓波、邢维岩、魏先民等老师参与部分章节的编写及文稿录入、排版和绘图等工作，在此表示感谢。全书由刘永华、王成端负责统稿整理。

山东大学的石冰教授和山东交通学院的沈祥玖教授对全书进行了审阅，并提出了许多宝贵意见，在此表示衷心感谢。

由于编者水平有限，编写时间仓促，书中疏漏之处在所难免，敬请广大读者批评指正。

编 者

2006年5月

目 录

第 1 章 概述	1
1.1 数据表示与运算	1
1.1.1 进位计数制及不同计数制之间的转换	1
1.1.2 二进制数和十六进制数运算	3
1.1.3 数据表示	4
1.1.4 定点数与浮点数	7
1.2 码制	8
1.2.1 带符号数的编码	8
1.2.2 数的定点表示与浮点表示	11
1.2.3 ASCII 码与 BCD 码	13
1.3 计算机语言	14
1.3.1 机器语言	14
1.3.2 汇编语言	14
1.3.3 高级语言	15
习题一	16
第 2 章 微机系统及微处理器	17
2.1 微机系统及其工作原理	17
2.1.1 概述	17
2.1.2 硬件系统结构	19
2.1.3 软件系统的结构	22
2.1.4 微机的工作原理	23
2.2 8086/8088 微处理器	24
2.2.1 8086 CPU 的内部结构	25
2.2.2 8086 CPU 寄存器组织	27
2.2.3 8086 CPU 引脚功能	30
2.2.4 8086/8088 CPU 的存储器组织和 I/O 组织	34
2.2.5 最小模式和最大模式下的基本配置	37
2.2.6 8086 CPU 内部时序	39
2.3 Pentium 系列微处理器	42
2.3.1 Pentium	42
2.3.2 Pentium Pro	43
2.3.3 Pentium MMX	43

2.3.4 Pentium II	44
2.3.5 Pentium III	44
2.3.6 Pentium 4	45
习题二	45
第 3 章 微机指令系统	47
3.1 寻址方式	47
3.1.1 操作数类型	47
3.1.2 寻址方式	47
3.2 指令系统	55
3.2.1 数据传送指令	55
3.2.2 算术运算指令	60
3.2.3 逻辑运算指令	68
3.2.4 移位指令	69
3.2.5 转移指令	72
3.2.6 字符串操作指令	76
3.2.7 处理器控制指令	79
3.2.8 输入/输出指令	80
3.2.9 中断指令	81
习题三	82
第 4 章 汇编语言	85
4.1 汇编语言语句	85
4.1.1 语句的类别与结构	85
4.1.2 指令语句的操作数	87
4.1.3 指令语句中的运算符和操作符	88
4.2 伪指令	92
4.2.1 数据定义与符号定义伪指令	92
4.2.2 段定义伪指令	94
4.2.3 模块定义与通信伪指令	95
4.2.4 过程定义伪指令	96
4.2.5 其他伪指令	96
4.3 汇编语言程序的结构	97
4.3.1 汇编语言程序的结构	97
4.3.2 程序正常返回 DOS 的方法	99
4.4 高级汇编语言技术	99
4.4.1 条件汇编	99
4.4.2 宏汇编	101
4.4.3 结构	102

4.4.4 记录	103
习题四	103
第 5 章 存储器接口技术	105
5.1 I/O 端口的寻址方式	105
5.1.1 I/O 端口	105
5.1.2 端口地址编址方式	105
5.1.3 端口访问指令	106
5.1.4 I/O 端口地址分配和选用	108
5.2 I/O 端口地址译码	109
5.2.1 I/O 端口地址译码方法	109
5.2.2 固定式端口地址译码	110
5.2.3 开关式可选端口地址译码	112
5.3 半导体存储器接口	113
5.3.1 半导体存储器接口的基本技术	113
5.3.2 静态 RAM 与 CPU 的连接	116
5.3.3 动态 RAM 与 CPU 的连接	118
5.3.4 ROM 存储器与 CPU 的连接	120
习题五	123
第 6 章 并行接口技术及应用	124
6.1 并行接口概述	124
6.1.1 并行接口概念	124
6.1.2 握手联络信号	125
6.2 简单并行接口	126
6.2.1 并行输入	126
6.2.2 并行输出	128
6.2.3 双向式输入输出	129
6.2.4 带有联络信号的输入输出	130
6.2.5 中断式输入	131
6.3 8255A 可编程并行接口芯片	132
6.3.1 8255A 引脚介绍	132
6.3.2 内部结构	133
6.3.3 工作方式控制字	134
6.3.4 3 种工作方式	135
6.3.5 8255A 编程	139
6.4 8255A 应用举例	141
6.4.1 并行打印机接口设计	141
6.4.2 双机并行通信接口设计	144
习题六	147

第 7 章 串行通信技术及应用	148
7.1 串行通信的基本概念	148
7.1.1 串行通信的特点	148
7.1.2 数据通信方式	148
7.1.3 串行通信方式	149
7.1.4 信息的校验方式	153
7.1.5 传输速率与传送距离	153
7.1.6 信号的调制与解调	154
7.1.7 串行接口的基本结构和基本功能	155
7.2 EIA-RS-232C 串行接口标准	157
7.2.1 电气特性	158
7.2.2 接口信号功能	159
7.2.3 信号线的连接	161
7.3 IBM-PC/XT 机异步通信接口	163
7.3.1 异步通信适配器的组成	163
7.3.2 INS 8250 的结构和外部特性	164
7.3.3 INS 8250 的内部寄存器及其编程方法	165
7.3.4 INS 8250 的编程	171
习题七	179
第 8 章 定时/计数器技术及应用	181
8.1 定时/计数器的基本概念	181
8.2 定时/计数器的分类	181
8.2.1 软件定时/计数器	181
8.2.2 硬件定时/计数器	181
8.2.3 可编程定时/计数器	182
8.3 可编程定时器/计数器 8253-5	182
8.3.1 8253-5 的主要特性	182
8.3.2 8253-5 的引脚与功能结构	182
8.3.3 8253-5 的内部结构	183
8.3.4 8253-5 的方式控制字	184
8.3.5 8253-5 的 6 种工作方式	186
8.3.6 8253-5 编程	190
8.4 8253-5 应用举例	192
习题八	195
第 9 章 中断处理技术及应用	196
9.1 中断的基本概念	196

9.1.1	中断的定义	196
9.1.2	中断的处理过程	197
9.1.3	中断源、中断识别及其优先级	198
9.1.4	中断向量	199
9.2	8086/8088 的中断系统	200
9.2.1	8086/8088 的中断系统结构	200
9.2.2	内部中断	203
9.2.3	外部中断	205
9.3	8259A 中断控制器	208
9.3.1	8259A 的外部特性和内部结构	208
9.3.2	8259A 的控制字及中断操作功能	212
9.4	8259A 的应用举例	219
9.4.1	8259A 在 PC/XT 及 PC/AT 系统中的初始化编程	219
9.4.2	8259A 应用举例	221
	习题九	222
第 10 章	DMA 技术及应用	223
10.1	DMA 技术概述	223
10.1.1	基本概念	223
10.1.2	DMA 控制器	224
10.2	DMA 控制器 8237A	225
10.2.1	主要特性	225
10.2.2	8237A 引脚及结构	226
10.2.3	8237A 的软件命令	234
10.2.4	8237A 的工作时序	234
10.3	8237A 初始化编程	236
10.4	8237A 应用实例	237
10.4.1	初始化	237
10.4.2	应用实例	238
	习题十	240
第 11 章	模拟接口技术及应用	241
11.1	D/A 转换器	241
11.1.1	D/A 转换器工作原理	241
11.1.2	D/A 转换器性能指标	242
11.1.3	D/A 转换芯片	243
11.1.4	D/A 转换器的接口	246
11.2	A/D 转换器	250
11.2.1	A/D 转换器的工作原理	250

11.2.2	A/D 转换器的主要性能指标	252
11.2.3	A/D 转换芯片	253
11.2.4	A/D 转换器的接口	256
11.3	多路模拟开关及采样保持电路	259
11.3.1	多路模拟开关	259
11.3.2	采样保持电路	260
11.4	数据采集系统设计	262
11.4.1	数据采集系统的构成	262
11.4.2	应用实例	263
11.4.3	数据采集接口设计应注意的问题	264
习题十一		265
第 12 章 总线技术		266
12.1	概述	266
12.1.1	总线的类别	266
12.1.2	总线的优点	266
12.2	系统总线	267
12.2.1	IBM-PC 总线	267
12.2.2	ISA 总线	269
12.2.3	EISA 总线	271
12.2.4	PCI 总线	272
12.2.5	STD 总线	274
12.2.6	AGP 总线	276
12.3	外部总线	278
12.3.1	IEEE-488 总线	278
12.3.2	SCSI 总线	281
12.3.3	IDE 总线	283
12.3.4	USB 总线	283
12.3.5	Fire Wire 串行总线 (IEEE 1394)	288
习题十二		291
第 13 章 综合应用		292
13.1	微机接口设计的基本方法	292
13.1.1	微机接口信号分析	292
13.1.2	微机接口信号转换	292
13.1.3	微机接口驱动程序的分析与设计	292
13.1.4	微机接口设计与分析时应注意的问题	293
13.2	硬件系统的设计	293
13.3	软件系统的设计	294

13.3.1	微机软件结构.....	294
13.3.2	微机软件接口.....	294
13.4	微机硬件中断的开发与设计.....	295
13.4.1	微机硬件中断开发概述.....	295
13.4.2	中断处理程序设计.....	296
13.5	综合应用举例.....	297
13.5.1	硬件设计.....	297
13.5.2	软件设计.....	298
13.5.3	程序框图.....	299
13.5.4	参考程序.....	303
习题十三		310
参考文献		311

第 1 章 概述

本章主要讲解了微机原理及应用的基本概念。通过本章学习，读者应该掌握以下内容。

- 进位计数制及相互转换
- 码制、定点数与浮点数
- 计算机语言

1.1 数据表示与运算

1.1.1 进位计数制及不同计数制之间的转换

1. 二进制数、八进制数和十六进制数

(1) 二进制数

日常生活中一般采用十进制进行计数，但计算机只能识别 0、1 代码，也就是说计算机采用二进制进行计数。二进制数只有 0、1 两个数码，其基数为 2，遵循逢二进一的原则，它的第 k 位权以 2^k 表示。

二进制数 $a_n a_{n-1} \cdots a_0 b_{-1} b_{-2} \cdots b_{-m}$ 的值是：

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_0 \times 2^0 + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-m} \times 2^{-m}$$

其中 a_i, b_j 为 0、1 两个数码中的一个。二进制数的描述是在其尾部加注字母 B，例如：

$$10100101B = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 128 + 32 + 4 + 1 = 165$$

N 位二进制数可以表示 2^N 个数。例如 3 位二进制数可以表示 8 个数，如表 1-1 所示。

表 1-1 3 位二进制与十进制数对应表

二进制数	000	001	010	011	100	101	110	111
相应的十进制数	0	1	2	3	4	5	6	7

4 位二进制数则表示十进制的 0~15 共 16 个数，如表 1-2 所示。

表 1-2 4 位二进制与十进制数的对应表

二进制数	0000	0001	0010	0011	0100	0101	0110	0111
相应的十进制数	0	1	2	3	4	5	6	7
二进制数	1000	1001	1010	1011	1100	1101	1110	1111
相应的十进制数	8	9	10	11	12	13	14	15

从上表可以看出，二进制数表示的个数越多位数越长，不便于人们阅读、书写和记忆，因此人们经常使用八进制数或十六进制数来表示二进制数。

(2) 八进制数

八进制数有 0、1、2、3、4、5、6、7 共 8 个数码，其基数为 8，遵循逢八进一的原则，

它的第 k 位权以 8^k 表示。八进制数的描述是在其尾部加注字母 O 或 Q。

八进制数 $a_n a_{n-1} \cdots a_0 b_{-1} b_{-2} \cdots b_{-m}$ 的值是：

$$a_n \times 8^n + a_{n-1} \times 8^{n-1} + \cdots + a_0 \times 8^0 + b_{-1} \times 8^{-1} + b_{-2} \times 8^{-2} + \cdots + b_{-m} \times 8^{-m}$$

例如：534Q = $5 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 = 5 \times 64 + 3 \times 8 + 4 \times 1 = 348$

(3) 十六进制数

十六进制数有 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F 共 16 个数码，其中 A, B, C, D, E, F 表示 10~15 共 6 个数码，其基数为 16，遵循逢十六进一的原则，它的第 k 位权以 16^k 表示。十六进制数的描述是在其尾部加注字母 H。

十六进制数 $a_n a_{n-1} \cdots a_0 b_{-1} b_{-2} \cdots b_{-m}$ 的值是：

$$a_n \times 16^n + a_{n-1} \times 16^{n-1} + \cdots + a_0 \times 16^0 + b_{-1} \times 16^{-1} + b_{-2} \times 16^{-2} + \cdots + b_{-m} \times 16^{-m}$$

例如：2ACH = $2 \times 16^2 + 10 \times 16^1 + 12 \times 16^0 = 2 \times 256 + 10 \times 16 + 12 \times 1 = 684$

表 1-3 列出了几种常用的进位制的基数和数码。

表 1-3 几种常用的进位制的基数和数码

进位计数制	基 数	数 码
十六进制数	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
十进制数	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
八进制数	8	0, 1, 2, 3, 4, 5, 6, 7
二进制数	2	0, 1

2. 不同数制之间的转换

(1) 非十进制数转换为十进制数

各位非十进制数码乘以与其对应的权然后再求和即为该数对应的十进制数。例如：

$$1011100.1011\text{B} = 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} = 92.6875\text{D}$$

$$\text{A031H} = 10 \times 163 + 3 \times 161 + 1 \times 160 = 41009$$

$$1001\text{Q} = 1 \times 83 + 1 \times 80 = 513$$

(2) 十进制数转换为非十进制数

十进制数转换为非十进制数一般整数部分采用除基数取余法，小数部分采用乘基数取整法。除基数取余法的具体方法是把待转换的十进制数的整数部分不断除以要转换为的非十进制的基数，并记下余数，直到商为 0 时为止；乘基数取整法的具体方法是把待转换的十进制数的小数部分不断乘以要转换为的非十进制基数，逐次记下乘积整数部分的值，直到小数部分为 0 时为止。

现以十进制数转换为二进制数为例进行说明。十进制数转换为二进制数将采用除 2 取余法和乘 2 取整法：把待转换的十进制数的整数部分不断除以 2，并记下余数，直到商为 0 时为止；对小数部分的转换是把待转换的十进制数的小数部分不断乘以 2，并逐次记下乘积整数部分的值，直到小数部分为 0 时为止。

【例 1.1】将 137.8125D 转换为二进制数。

整数部分 137D，按除 2 取余法转换，则：

$137/2=68$	$(a_0=1)$
$68/2=34$	$(a_1=0)$
$34/2=17$	$(a_2=0)$
$17/2=8$	$(a_3=1)$
$8/2=4$	$(a_4=0)$
$4/2=2$	$(a_5=0)$
$2/2=1$	$(a_6=0)$
$1/2=0$	$(a_7=1)$

故 $137D=10001001B$ 。

小数部分为 $0.8125D$ ，按乘 2 取整法转换，则

$0.8125 \times 2 = 1.625$	$(b_1=1)$
$0.625 \times 2 = 1.25$	$(b_2=1)$
$0.25 \times 2 = 0.5$	$(b_3=0)$
$0.5 \times 2 = 1.0$	$(b_4=1)$

故 $0.8125D=0.1101B$ 。

所以， $137.8125D=10001001.1101B$ 。

十进制数转换为十六进制数和八进制数的方法与十进制数转换为二进制数的方法类似。

(3) 十六进制数与二进制数之间的转换

因为 $16=2^4$ ，所以一个十六进制数中的每一位都可以用 4 位二进制数表示，形成相应的二进制数。

【例 1.2】	C	B	9	A
	1100	1011	1001	1010

即 $CB9AH=1100101110011010B$

反之，只要把二进制数从低位到高位每 4 位组成一组，再用十六进制数来表示就可以将二进制数转换成十六进制数了。

【例 1.3】	0111	0101	1011	1111
	7	5	B	F

所以， $0111010110111111B=75BFH$ 。

1.1.2 二进制数和十六进制数运算

1. 二进制数的运算

加法运算规则：

$0+0=0$
$0+1=1$
$1+0=1$
$1+1=10$ (1 为进位)

乘法运算规则：

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

2. 十六进制数的运算

(1) 十六进制数的加法

十六进制数的运算按照逢十六进一的规则进行，即当两个一位数之和 S 小于 16 时，与十进制数同样处理，当两个一位数之和 S 大于等于 16 时，则应该用 $S-16$ 或进位 1 来取代 S 。

$$\begin{array}{r} \text{【例 1.4】} \quad 15C3H \\ + 3D45H \\ \hline 5308H \end{array}$$

(2) 十六进制数的减法

与十进制数类似，被减数够减时可以直接相减，不够减时服从向高位借 1 为 16 的规则。

$$\begin{array}{r} \text{【例 1.5】} \quad 3DA6H \\ - 0FC35H \\ \hline 2DE8H \end{array}$$

1.1.3 数据表示

计算机内的数据有多种形式，其中最主要的是数值数据和字符数据。

1. 数值数据的表示

数值数据可以用不同的码制来表示，常用的有原码、补码和反码。由于进行加减运算时补码表示法的符号位可以参加运算，而且不影响运算结果的正确性，所以多数计算机的有符号整数都采用补码表示法。本章将只重点介绍补码表示法。

(1) 数的补码表示

① 正数的表示

正数采用符号—绝对值表示，即数的最高有效位为 0，表示符号为正，数的其余部分则表示数的绝对值。

例如，假设计算机字长为 8 位，则

$$[+0]_{\text{补}} = 00000000B$$

$$[+1]_{\text{补}} = 00000001B$$

$$[+100]_{\text{补}} = 01100100B$$

② 负数的表示

用补码表示法来表示负数时，可以采用“求反加 1”的方法来完成：先写出与该负数相对应的正数的补码表示（用符号—绝对值法），然后将其按位求反（即 0 变为 1，1 变为 0），最后在末位（最低位）加 1，就可以得到该负数的补码。

【例 1.6】 计算机字长为 8 位，写出 -27 的补码表示。

+27D 可表示为： 00011011

按位求反为： 11100100

末位加 1 后为： 11100101

用十六进制数表示为 E5H。

所以， $[-27]_{\text{补}} = E5H$ 。

【例 1.7】计算机字长为 16 位，写出 -32768 的补码表示。

32768D 可表示为： 1000000000000000

按位求反为： 0111111111111111

末位加 1 后为： 1000000000000000

用十六进制数表示为： 8000

所以， $[-32768]_{\text{补}} = 8000\text{H}$ 。

③ 数的表示范围

• 有符号数的表示范围

一般说来， n 位二进制补码表示的数的范围是：

$$-2^{n-1} \leq N \leq 2^{n-1} - 1$$

例如，8 位二进制数可以表示 $2^8=256$ 个数。在补码表示法中 0 只有一种表示，即 00000000；对于 10000000 这个数，在补码表示法中被定义为 -128。这样，8 位补码能表示的范围为 -128~127。

$n=16$ 时的数的表示范围是：

$$-32768 \leq N \leq +32767$$

• 无符号整数的表示范围

在进行无符号数处理时，把最高有效位作为数值处理。因此，16 位无符号数的表示范围是 $0 \leq N \leq 65535$ ，8 位无符号数的表示范围是 $0 \leq N \leq 255$ 。

(2) 补码的运算

① 求补运算

$$([X]_{\text{补}})_{\text{求补}} = [-X]_{\text{补}}$$

【例 1.8】 $[117]_{\text{补}} = 0075\text{H}$

$$[-117]_{\text{补}} = \text{FF}8\text{BH}$$

对 $[-117]_{\text{补}}$ 作求补运算过程如下。

$[-117]_{\text{补}}$ 为： 111111110001011

按位求反后得： 000000001110101

末位加 1 后得： 000000001110101

所以， $[+117]_{\text{补}} = 0075\text{H}$ 。

② 补码的加、减法运算

补码的加法规则是：

$$[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$$

补码的减法规则是：

$$[X-Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$$

其中，只要对 $[Y]_{\text{补}}$ 求补就可得到 $[-Y]_{\text{补}}$ 。