



普通高等院校电子信息类应用型规划教材

微机原理与接口技术

主编 冯涛



WEIJI YUANLI
YU JIEKOU JISHU



北京邮电大学出版社
www.buptpress.com

普通高等院校应用型规划教材

微机原理与接口技术

主 编	冯 涛		
副主编	秦 颖	张永祥	董晓红
	姚 丹	姜 威	王 蕊
	商俊平	张志锋	张会发

北京邮电大学出版社
· 北 京 ·

内 容 简 介

本书是非计算机专业、电类本科“微机原理与接口技术”课程的应用型教材,较系统全面地阐述了 16 位微型计算机的工作原理、体系结构、汇编程序的设计方法以及并行输入/输出、中断、DMA 控制器、可编程定时器/计数器、串行通信等接口技术,并对微机的发展方向与新技术作了介绍。

本书内容丰富、结构合理,附有大量的实例和练习题,既可用做教材,也可供自学和工程人员参考。

图书在版编目(CIP)数据

微机原理与接口技术/冯涛主编. --北京:北京邮电大学出版社,2010.9

ISBN 978-7-5635-2419-8

I. ①微… II. ①冯… III. ①微型计算机—理论—高等学校—教材②微型计算机—接口—高等学校—教材 IV. ①TP36

中国版本图书馆 CIP 数据核字(2010)第 172452 号

书 名: 微机原理与接口技术

主 编: 冯 涛

责任编辑: 刘春棠

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish @ bupt. edu. cn

经 销: 各地新华书店

印 刷: 北京忠信诚胶印厂

开 本: 787 mm×1 092 mm 1/16

印 张: 17

字 数: 419 千字

印 数: 1—1 500 册

版 次: 2010 年 9 月第 1 版 2010 年 9 月第 1 次印刷

ISBN 978-7-5635-2419-8

定 价: 29.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前言

自 20 世纪 70 年代初第 1 代微型计算机问世以后,计算机以惊人的速度发展,其应用已渗透到各行各业的各个领域,深入到科学计算、信息处理、过程控制、仪器仪表、事务管理、计算机辅助设计与制造、家用电器、网络通信服务等方面,极大地改变着人们的工作和生活方式,已成为社会前进的巨大推动力,因此微处理器和微计算机的相关知识已成为现代科技人员和高等院校各专业学生的重点课程。

本书共分 10 章,主要包括微型计算机系统概述、微处理器、8086/8088 的指令系统、汇编语言程序设计、微型计算机总线、存储器、微型计算机和外设间的数据传输、中断系统、微型计算机常用接口技术、微型计算机的发展方向与新技术介绍。学习“微机原理与接口技术”,目的在于加强对微型计算机硬件组成的理解,提高对计算机硬件的应用,甚至硬件开发的能力。

本书是依据培养应用型人才的教学特点,为适应“三个面向”对高校人才的教学需要而编写的。本书深入浅出、通俗易懂,结合当今科技与生产的实际,列举大量图表和例题来帮助读者对知识有较深、较广的理解,培养学习兴趣。每章所附的练习题将有助于学生巩固所学的知识。

本书第 1、2、3、4 章由张永祥编写,第 5、6、7、10 章和附录由秦颖编写,第 8、9 章由冯涛编写。在编写过程中,本书参考了大量的国内外文献资料,特此对这些文献的作者致以谢意。

由于时间仓促,书中难免存在不足之处,恳请读者批评指正。

编 者

目 录

第 1 章 微型计算机系统概述	1
1.1 微型计算机发展概况	1
1.2 计算机中数的表示与编码	4
1.2.1 计算机中常用的数制及其转换	4
1.2.2 带符号数的表示	7
1.2.3 计算机中常用的编码	8
1.3 微型计算机系统	10
1.4 计算机常用的名词术语	13
1.5 微型计算机的主要性能指标	14
练习题	15
第 2 章 微型计算机系统的微处理器	16
2.1 8086/8088 的编程结构	16
2.2 寄存器的结构	20
2.3 8086 的引脚和工作模式	23
2.3.1 最小工作模式和最大工作模式	24
2.3.2 8086/8088 CPU 的引脚信号和功能	24
2.4 8086 系统的配置	28
2.4.1 最小模式系统	29
2.4.2 最大模式系统	32
2.5 8086 存储器组织	34
2.6 8086 的 I/O 端口组织	38
2.7 8086 的总线操作时序	39
2.7.1 时序的基本概念	39
2.7.2 基本时序分析	40
练习题	43
第 3 章 8086/8088 的指令系统	45
3.1 指令的基本格式	45

3.2	8086/8088 的寻址方式	46
3.3	8086/8088 指令系统	51
3.3.1	数据传送类指令	51
3.3.2	算术运算类指令	60
3.3.3	逻辑运算与移位类指令	74
3.3.4	串操作类指令	78
3.3.5	控制转移类指令	82
3.3.6	标志操作和处理器控制类指令	92
	练习题	93
第 4 章	汇编语言程序设计	96
4.1	汇编语言程序格式	98
4.2	常用伪指令	99
4.3	常数、变量及标号	101
4.4	运算符	102
4.5	程序设计方法	105
4.5.1	顺序结构	105
4.5.2	分支结构	107
4.5.3	循环结构	112
4.5.4	子程序结构	119
	练习题	122
第 5 章	微型计算机总线	124
5.1	总线概述	124
5.2	8088 最大模式下总线信号的形成	126
5.3	常用微型计算机总线介绍	127
5.3.1	PC 总线	127
5.3.2	ISA 总线	128
5.3.3	PCI 总线	130
5.3.4	USB 总线	132
5.4	总线仲裁和握手技术	133
5.4.1	总线主设备和总线从设备	133
5.4.2	总线仲裁	134
5.4.3	总线握手	135
	练习题	136
第 6 章	存储器	138
6.1	存储器的分类	138

6.2 存储器的结构	139
6.2.1 RAM	139
6.2.2 ROM	143
6.3 存储器系统设计	145
6.3.1 存储器芯片介绍	145
6.3.2 存储器地址选择	147
6.3.3 存储器扩展	148
6.4 新型存储技术	149
6.4.1 高速缓存器	149
6.4.2 光盘存储器	150
6.4.3 虚拟存储器	151
练习题	152
第7章 微型计算机和外设间的数据传输	153
7.1 概述	153
7.2 CPU 和输入/输出设备间的信号	154
7.2.1 CPU 与 I/O 接口之间的接口信号	154
7.2.2 端口地址与编址方式	154
7.3 CPU 和外设之间的数据传送方式	155
7.3.1 无条件传送方式	155
7.3.2 查询传送方式	156
7.3.3 中断传送方式	159
7.3.4 DMA 传送方式	159
练习题	161
第8章 中断技术	162
8.1 中断概述	162
8.1.1 中断的概念	162
8.1.2 中断源	163
8.1.3 中断的类型	163
8.1.4 中断类型号	164
8.1.5 中断矢量表	164
8.1.6 中断优先级	165
8.1.7 中断嵌套	165
8.2 8086 CPU 的中断处理过程	166
8.2.1 中断请求	166
8.2.2 中断响应	167

8.2.3 中断处理	168
8.2.4 中断返回	168
8.3 可编程中断控制器 8259A	168
8.3.1 8259A 的内部结构与引脚	169
8.3.2 8259A 的中断响应时序	171
8.3.3 8259A 的工作方式	172
8.3.4 8259A 的命令字	173
8.3.5 8259A 的编程	178
8.4 DOS 下中断服务程序的编写	179
练习题.....	180
第 9 章 I/O 接口技术	182
9.1 I/O 接口	182
9.2 并行接口芯片 8255A	183
9.2.1 8255A 概述	183
9.2.2 8255A 的控制字	186
9.2.3 8255A 的工作方式	187
9.2.4 8255A 的编程	190
9.3 定时器/计数器 8253	192
9.3.1 8253 的基本功能、引脚及内部结构	192
9.3.2 8253 的工作方式	193
9.3.3 8253 的控制字与编程	197
9.4 DMA 控制器 8237A	199
9.4.1 8237A 概述	199
9.4.2 8237A 的引脚	200
9.4.3 8237A 的内部结构与寄存器	201
9.4.4 8237A 的软件命令	207
9.4.5 8237A 的工作时序	208
9.4.6 8237A 的初始化	209
9.5 串行通信接口及其应用	210
9.5.1 串行通信数据传送方式	210
9.5.2 串行通信协议	211
9.5.3 可编程串行通信接口芯片 8250	212
练习题.....	221
第 10 章 微型计算机的发展方向及多媒体技术	223
10.1 微型计算机新技术的发展.....	223

10.1.1 CPU 的发展	223
10.1.2 存储器的发展.....	225
10.1.3 外设及接口的发展.....	226
10.2 多媒体技术.....	227
10.2.1 多媒体计算机的组成.....	227
10.2.2 多媒体设备及接口.....	230
附录 1 8086、8088 指令系统一览表	241
附录 2 指令表	253
附录 3 中断向量地址一览表	258
参考文献.....	260

第 1 章 微型计算机系统概述

1946 年,世界上第一台电子数字计算机在美国问世。作为 20 世纪的先进技术成果之一,计算机最初只是作为一种现代化的计算工具。在 60 多年的发展历程中,计算机从采用电子管、晶体管、中小规模集成电路、大规模集成电路到超大规模集成电路。在 70 年代初,大规模集成电路技术的发展和微型计算机(简称微机)的出现为计算机的广泛应用开拓了极其广阔的前景,展示了它在科学发展技术领域中日益显要的地位。计算机特别是微型计算机的科学技术水平、生产规模和应用已成为衡量一个国家现代化水平的重要标志。计算机已经远不只是一种计算工具,它已渗透到国民经济和生活的各个领域,像火车头一样带动着各行各业向现代化迈进。微型计算机的发展和应用已引起各界的关注,并已转化成巨大的推动社会前进的生产力。

本章主要介绍微型计算机的基础知识,包括微型计算机的发展概况、计算机中数的表示与编码、微型计算机的组成及工作原理等。要求掌握信息和数据在微型计算机中的表示与运算,了解微型计算机的发展历程及整机的工作流程。

1.1 微型计算机发展概况

第 1 代电子计算机 ENIAC(Electronic Numerical Integrator And Calculator)使用了 18 800 个电子管,重 30 t,占地 150 m²,耗电 150 kW,每秒完成 5 000 次加法运算。第 2 代电子计算机在 1958 年推出,用晶体管代替了电子管,大大降低了计算机的成本和体积,运算速度成百倍地提高。1965 年,以中小规模集成电路为主体的计算机问世,计算机的体积进一步缩小,配上各类操作系统,计算机的性能极大提高。1970 年,大规模集成电路(LSI)研制成功,计算机也发展到第 4 代,微型计算机正是第 4 代计算机的典型代表。1971 年,在美国硅谷第 1 台微型计算机诞生了,从而开创了微型计算机的新时代。

微型计算机与大型机、中型机、小型机在系统结构和工作原理上相比没有本质区别,但由于它采用了 LSI 器件,集成度高,因此具有独特的优点:体积小,重量轻,可靠性高,结构配置灵活,价格低廉,从而发展迅猛。微型计算机自 1971 年问世以来,几乎每隔两三年就推出一代新的微处理器(Central Processing Unit,CPU)。CPU 是微型计算机的核心,其性能的优劣将直接影响整机的性能。下面介绍一下 CPU 的发展情况。

1. 第 1 代 CPU(1971 年开始)

4 位和 8 位 CPU(数据总线分别为 4 位和 8 位)的典型产品如下。

1971 年 Intel 4004

1972 年 Intel 8008

Intel 4004 芯片采用 PMOS 工艺,集成度为每片 2 300 只晶体管,时钟频率小于 1 MHz,平均指令执行时间为 $10\sim 15\ \mu\text{s}$,采用机器语言编程。这种微处理器为内嵌式处理器,又称为灵巧型处理器(Smart),主要用在控制设备中,如现金计数器、交通灯控制等。

2. 第 2 代 CPU(1974 年开始)

8 位 CPU(数据总线 8 位)的典型产品如下。

1974 年 Intel 8080 1974 年 Motorola MC6800

1975 年 Zilog Z80 1976 年 Intel 8085

8080 采用 NMOS 工艺,集成度为每片 4 500 只晶体管,时钟频率为 2 MHz,平均指令执行时间为 $1\sim 2\ \mu\text{s}$,能寻址 64 KB 内存空间。用它构成的 CPU 在结构上已具有计算机的体系结构,有中断和 DMA 等功能,指令系统较为完善,软件上也配备了汇编语言、BASIC 和 FORTRAN 语言,使用单用户操作系统。

3. 第 3 代 CPU(1978 年开始)

16 位 CPU 的典型产品如下。

1978 年 Intel 8086 1979 年 Zilog Z8000

1979 年 Motorola 68000 1982 年 Intel 80286 和 Motorola 68010

8086 采用 HMOS 工艺,集成度达每片 2.9 万只晶体管,时钟频率有 5 MHz、8 MHz、10 MHz,平均指令执行时间为 $0.5\ \mu\text{s}$ 。该 CPU 具有丰富的指令系统,采用多级中断、多重寻址方式,有段寄存器结构,配有磁盘操作系统、数据库管理系统和多种高级语言,数据总线为 16 位,地址总线为 20 位,可寻址 1 MB 内存空间,投放市场后迅速占领了市场,促进了个人计算机的应用和推广。

4. 第 4 代 CPU(1983 开始)

32 位 CPU 的典型产品如下。

1983 年 Zilog Z80000 1984 年 Motorola 68020

1985 年 Intel 80386 1989 年 Intel 80486 和 Motorola 68040

80386 采用 CHMOS 工艺和 132 条引脚的针筒阵列封装,集成度达每片 27.5 万只晶体管,指令执行速度达到 10 MIPS。其工作方式除 80286 的实模式和保护模式外,还增加了虚拟 8086 模式。在实模式下,能运行 8086 指令,而运行速度却比 80286 快 3 倍。而 80486 集成度达每片 100 万只晶体管。与 80386 不同的是,80486 将不同功能的芯片电路集成到一个芯片上。在 80486 芯片上,除有 80386 外,还集成了 80387 浮点运算器(FPU)、82385 高速暂存控制器和 8 KB 的高速缓冲器(Cache)。这样,80486 就在 80386 的基础上更加高速化。当时钟频率为 25 MHz 时,指令执行速度达 15 MIPS;而时钟频率为 33 MHz 时,指令执行速度达 15 MIPS。

5. 第 5 代 CPU(1993 年开始)

奔腾(Pentium)产品时代的典型产品如下。

1993 年 Pentium(也称为 Intel 80586) 1995 年 Pentium Pro

1997 年 Pentium II 1999 年 Pentium III

2000 年 Pentium 4

Pentium 微处理器的面市称得上是微型计算机发展史上的里程碑。Pentium 微处理器不仅是对前代产品 80486 的改进,而且从设计思想上把提高微处理器内部指令的并行性和高效率作为指导。它把芯片上的 Cache 加倍为 16 KB,并分为两个:一个 8 KB 作为指令缓冲器(L1),另一个 8 KB 作为数据缓冲存储器(L2);数据总线宽度由 32 位增加到 64 位;采用双整数处理器技术,允许每个时钟周期同时执行两条指令。这种有两个独立的整数器的技术又称为超标量(Superscalar)技术。为了摆脱 80486 时代处理器名称的混乱,世界最大的 CPU 制造商 Intel 公司把这类新一代的产品命名为 Pentium(奔腾),并进行了注册。

Pentium Pro(高能奔腾)的集成度为每片 550 万个晶体管,时钟频率为 150 MHz,运行速度达到 400 MIPS,是一种比 80586 更快的第 2 代奔腾产品。其具有更优化的内部体系结构,整数处理器增加为 3 个,浮点运算速度也加快,这样内部可以同时执行 3 条指令;片内除原有的第 1 级 16 KB 高速缓冲存储器 L1 外,还增加了一个 256 KB 的第 2 级高速缓冲存储器 L2;采用双重独立总线和动态执行技术,地址总线又增加了 4 条(共 36 条),能寻址 64 GB 存储空间。

1996 年 Intel 公司将多媒体扩展技术(MultiMedia Extension, MMX)应用到 Pentium 芯片上,推出 Pentium MMX 处理器。其外部引脚与 80586 兼容,但在指令系统中增加了 57 条多媒体指令,用于音频、视频、图形图像数据处理,使多媒体、通信处理能力得到了很大的提高。

Pentium II 集成度为每片 750 万只晶体管,主频分为 233 MHz、266 MHz、300 MHz、333 MHz、350 MHz、400 MHz、450 MHz。其主要特性是:双重独立总线结构(二级高速缓存总线及处理器到主内存的系统总线分别独立);内置 MMX 技术;片内高速缓存 L1 从 16 KB 增加到 32 KB,外部高速缓存 L2 的容量为 512 KB,并以 CPU 主频的一半速度运行;动态执行使在给定时间内能处理更多的数据,提高了 CPU 的工作效率;采用单边接触插盒(Single Edge Contact Cartridge, SECC)和 slot1 接口标准。

Pentium III 采用 0.25 μ s 工艺制造,内部集成 950 万只晶体管。此外,它还具有以下特点:主频为 450 MHz,系统总线频率为 100 MHz,双重独立总线,一级缓存 L1 为 32 KB (16 KB 指令缓存和 16 KB 数据缓存),二级缓存 L2 为 512 KB,以 CPU 核心速度的一半运行;采用 SECC2 封装形式,增加了能够增强音频、视频和 3D 图形效果的数据流单数据多扩展(Streaming SIMD Extensions, SSE)指令集,共 70 条新指令;能同时处理 4 个单精度浮点变量,提供全新的“处理器分离模式”。

Pentium 4 主频为 1.5~3.6 GHz,有 144 条 SSE2 新指令集,虽然 SSE2 指令集解码之后将会被这部分 8 通道的缓存寄存起来,并且这部分缓存同时也负责预测处理通道中的数据。这样做的目的是减少长数据通道带来的坏处。Pentium 4 有着很独特的 L2 缓存,CPU 与 L2 之间有 256 bit 的内部传输通道,同时每个时钟周期都能实现从 L2 缓存中交换数据,也就是说这个数据带宽的峰值是现有 CPU 和 L2 缓存之间的数据带宽当中最高的。

以 Intel 微处理器为代表,表 1-1 给出了微处理器性能的演进过程。

表 1-1 Intel 微处理器性能的演进

性能 芯片	地址总线	数据总线	一级缓存	二级缓存	工作频率	集成度
8080	16	8			2 MHz	4 500 只/片
8088	20	8			5 MHz	29 000 只/片
8086	20	16			5 MHz, 8 MHz, 10 MHz	29 000 只/片
80286	24	16			12 MHz, 20 MHz, 25 MHz	13.4 万只/片
80386SX	24	16			16 MHz, 25 MHz, 33 MHz	27.5 万只/片
80386DX	32	32			16 MHz, 33 MHz, 40 MHz	27.5 万只/片
80486DX	32	32	8 KB		25~100 MHz	120 万只/片
Pentium	32	64	16 KB		66~200 MHz	310 万只/片
Pentium MMX	32(36)	64	16 KB		200~300 MHz	450 万只/片
Pentium Pro	36	64	16 KB	256 KB	150~200 MHz	550 万只/片
Pentium II	36	64	32 KB	512 KB	233~450 MHz	750 万只/片
Pentium II Xeon	36	64	32 KB	512 KB	350~450 MHz	750 万只/片
Pentium III	36	64	32 KB	512 KB	450~1.4 GHz	950 万只/片
Pentium 4	36	64	32 KB	1 MB	1.3~2.8 GHz	4 200 万只/片

随着科学技术的发展,未来计算机的发展趋势有如下几个方面。

- ① 朝着微型计算机和巨型计算机两极方向发展。
- ② 开发和研究的热点是多媒体计算机。
- ③ 未来计算机发展的总趋势是智能化计算机。

1.2 计算机中数的表示与编码

计算机中的数是以器件的物理状态来表示的。一个具有两种不同状态,并且能够相互转换的器件即可用来表示一位二进制数。因为二进制数表示法简单、可靠,所以计算机中采用二进制数字系统。计算机处理的各种信息,无论其表现形式是文本、符号、图形,还是声音、图像,都必须以二进制数的形式来表示。本节将介绍计算机中常用的数制及其转换、带符号数的表示以及计算机中常用的编码。要求掌握常用的数制及其转换方法、带符号数的表示及补码运算、计算机中常用的编码。

1.2.1 计算机中常用的数制及其转换

人们最常用的是十进制数,但为了存储和计算方便,计算机采用二进制数字系统,只包含 0 和 1 两个数。为了书写和阅读方便,引入了八进制数和十六进制数。下面介绍这几种数制及其转换。

1. 几种数制的表示

书写时,使用后缀表明数的进制。

(1) 二进制(Binary)

后缀:B。

元素:0,1。

例如:1010.1011B。

(2) 八进制(Octal)

后缀:O,为防止与数字0混淆,常后缀Q。

元素:0,1,2,3,4,5,6,7。

例如:17.6Q。

(3) 十进制(Decimal)

后缀:D或省略。

元素:0,1,2,3,4,5,6,7,8,9。

例如:4 659.37。

(4) 十六进制(Hexadecimal)

后缀:H。

元素:0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F。

例如:8D.AH。

总之, R 进制数的特点如下。

① 用 R 个符号表示数。这些符号称为数码。数码的个数称为基数, R 进制的基数是 R 。

② 在一个数中,每个数码表示的值不仅取决于数码本身,还取决于它所处的位置,每一位有各自的权。

③ 采用逢 R 进一的进位规则。

④ 小数点右移一位相当于乘以 R ,左移一位相当于除以 R 。

2. 进制计数之间的转换

(1) R 进制数转换为十进制数

其基本方法是“按权展开,并求和”。

例 1-1 将1101.101B、127.04Q、1DF.4H转换为十进制数。

解: $1101.101B = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 13.625$

$127.04Q = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 0 \times 8^{-1} + 4 \times 8^{-2} = 87.0625$

$1DF.4H = 1 \times 16^2 + 13 \times 16^1 + 15 \times 16^0 + 4 \times 16^{-1} = 479.25$

(2) 十进制数转换为 R 进制数

其基本方法是:对整数和小数部分分别进行转换。

① 整数部分的转换:“除 R 取余,倒写”,即十进制数多次除以转换进制基数,每次取一个余数,直到商为0,最后将余数从低位排向高位。

例 1-2 将十进制数38分别转换成二进制数、八进制数、十六进制数。

$$\begin{array}{r}
 \text{解: } 2 \overline{) 38} \\
 \underline{2 \overline{) 19}} \quad 0 \uparrow \\
 \underline{2 \overline{) 9}} \quad 1 \uparrow \\
 \underline{2 \overline{) 4}} \quad 1 \uparrow \\
 \underline{2 \overline{) 2}} \quad 0 \uparrow \\
 \underline{2 \overline{) 1}} \quad 0 \uparrow \\
 \quad \quad 0 \quad 1
 \end{array}
 \quad
 \begin{array}{r}
 8 \overline{) 38} \\
 \underline{8 \overline{) 4}} \quad 6 \uparrow \\
 \quad \quad 0 \quad 4 \uparrow
 \end{array}
 \quad
 \begin{array}{r}
 16 \overline{) 38} \\
 \underline{16 \overline{) 2}} \quad 6 \uparrow \\
 \quad \quad 0 \quad 2 \uparrow
 \end{array}$$

所以

$$38 = 100110B \quad 38 = 46Q \quad 38 = 26H$$

② 小数部分的转换：“乘 R 取整，顺写”，即用转换进制的基数乘以小数部分，每乘一次取积的整数，直至小数部分为 0 或达到转换精度要求的位数，最后将整数从最高位排到最低位。

例 1-3 将 0.6875 转换成二进制数、八进制数、十六进制数。

$$\begin{array}{r}
 \text{解: } 0.6875 \\
 \times 2 \\
 \hline 1.375 \quad 1 \downarrow \\
 \times 2 \\
 \hline 0.750 \quad 0 \downarrow \\
 \times 2 \\
 \hline 1.500 \quad 1 \downarrow \\
 \times 2 \\
 \hline 1.000 \quad 1 \downarrow
 \end{array}
 \quad
 \begin{array}{r}
 0.6875 \\
 \times 8 \\
 \hline 5.5000 \quad 5 \downarrow \\
 \times 8 \\
 \hline 4.0000 \quad 4 \downarrow
 \end{array}
 \quad
 \begin{array}{r}
 0.6875 \\
 \times 16 \\
 \hline 11.0000
 \end{array}$$

所以

$$0.6875 = 0.1011B \quad 0.6875 = 0.54Q \quad 0.6875 = 0.BH$$

(3) 二进制数与八进制数、十六进制数之间的相互转换

因为 $8 = 2^3$, $16 = 2^4$, 所以二进制数与八进制数、十六进制数之间的转换相对简单。

$$\begin{array}{c}
 \xleftarrow{\text{三合一}} \quad \xrightarrow{\text{四合一}} \\
 Q \quad \quad \quad B \quad \quad \quad H \\
 \xrightarrow{\text{一分三}} \quad \xleftarrow{\text{一分四}}
 \end{array}$$

① 八进制数转换为二进制数：1 位八进制数用 3 位二进制数表示。

② 十六进制数转换为二进制数：1 位十六进制数用 4 位二进制数表示。

③ 二进制数转换为八进制数：从小数点开始，分别向左、右两边把 3 位二进制数划为一组，最左和最右一组不足 3 位用 0 补充，然后每组用一个八进制数代替。

④ 二进制数转换为十六进制数：与八进制数类似，但是 4 位分为一组。

例 1-4 将 101011.1011B 转换为八进制数和十六进制数。

解：101011.1011B 三合一为八进制数，四合一为十六进制数，不足补 0。

$$\frac{101011}{5 \quad 3} \cdot \frac{101100}{5 \quad 4} B = 53.54Q$$

$$\frac{00101011}{2 \quad B} \cdot \frac{1011}{B} B = 2B.BH$$

例 1-5 将 576.3Q、3D、A9H 转换为二进制数。

解：576.3Q 一分三、3D、A9H 一分四为二进制数。

$$\frac{5 \quad 7 \quad 6 \quad 3}{101111110011} Q = 101111110.011B$$

$$\frac{3 \quad D \quad A \quad 9}{0011110110101001} H = 111101.10101001B$$

1.2.2 带符号数的表示

计算机中的数用二进制表示,数的符号也用二进制表示,一般用最高有效位来表示数的符号,0表示正数,1表示负数。将一个数与符号用数值化表示,这样的数称为机器数。机器数的字长由计算机的字长(即数据线的数量)决定,计算机的字长也决定了机器数的表示范围。

8位字长(即8条数据线)可以表示 $2^8=256$ 个数,对无符号数,取值范围为00000000B~11111111B,即00H~FFH(0~255);对有符号数,由于最高位表示符号位,故取值范围为10000000B~01111111B,即80H~7FH(-128~+127)。

16位字长(即16条数据线)可以表示 $2^{16}=65\,536$ 个数,无符号数的取值范围为0000H~FFFFH(0~65\,535),有符号数的取值范围为8000H~7FFFH(-32\,768~+32\,767)。

n 位字长(即 n 条数据线)可以表示 2^n 个数。

常用的机器数的表示方法有原码、反码、补码,而机器数所代表的实际数值称为真值。

例 1-6 写出下列机器数的真值。

$$[X_1]_{\text{机}}=01000101\text{B} \quad [X_2]_{\text{机}}=11000101\text{B}$$

解:真值: $X_1=+1000101\text{B}=+69$

$$X_2=-1000101\text{B}=-69$$

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为原码。

2. 反码

正数的反码与其原码相同;负数的反码最高位不变,即“1”表示负数,其余位为数值位按位取反。

例 1-7 若计算机的字长为8位,写出+4、-4、+0、-0、+127、-127的原码、反码。

$$\begin{aligned} \text{解:} [+4]_{\text{原}} &= 00000100\text{B} & [-4]_{\text{原}} &= 10000100\text{B} \\ [+4]_{\text{反}} &= [+4]_{\text{原}} = 00000100\text{B} & [-4]_{\text{反}} &= 11111011\text{B} \\ [+0]_{\text{原}} &= 00000000\text{B} & [-0]_{\text{原}} &= 10000000\text{B} \\ [+0]_{\text{反}} &= [+0]_{\text{原}} = 00000000\text{B} & [-0]_{\text{反}} &= 11111111\text{B} \\ [+127]_{\text{原}} &= 01111111\text{B} & [-127]_{\text{原}} &= 11111111\text{B} \\ [+127]_{\text{反}} &= [+127]_{\text{原}} = 01111111\text{B} & [-127]_{\text{反}} &= 10000000\text{B} \end{aligned}$$

由上面的例子可以看出,+0和-0的原码不同,即“0”占用两个码,真值0有两种不同的表示形式。因此,对于8位机,原码只能表示 $2^8-1=255$ 个数,表示整数范围为-127~+127。同理,反码也只能表示255个数。

3. 补码

正数的补码同原码,负数的补码为其反码(含符号位)加1。

例 1-8 若计算机的字长为8位,写出+4、-4、+0、-0、+127、-127、-128的补码。

$$\begin{aligned} \text{解:} [+4]_{\text{补}} &= [+4]_{\text{原}} = 00000100\text{B} = 04\text{H} \\ [-4]_{\text{补}} &= [-4]_{\text{反}} + 1 = 11111011 + 1 = 11111100\text{B} = \text{FCH} \end{aligned}$$

$$[+0]_{\text{补}} = [+0]_{\text{原}} = 00000000\text{B} = 00\text{H}$$

$$[-0]_{\text{补}} = [-0]_{\text{反}} + 1 = 11111111 + 1 = 00000000\text{B} = 00\text{H}$$

$$[+127]_{\text{补}} = [+127]_{\text{原}} = 01111111\text{B} = 7\text{FH}$$

$$[-127]_{\text{补}} = [-127]_{\text{反}} + 1 = 10000000 + 1 = 10000001\text{B} = 81\text{H}$$

$$[-128]_{\text{补}} = 10000000 = 80\text{H} (\text{对于 8 位机, } -128 \text{ 没有原码和反码})$$

从上面的例子可以看出, $+0$ 和 -0 的补码相同, 即“0”只占用 1 个码。对于 8 位机, -128 虽没有原码和反码, 但有补码, 所以补码的整数表示范围为 $-128 \sim +127$; 对于 16 位机, 补码的整数表示范围为 $-32768 \sim +32767$; 若字长为 n , 则补码的整数表示范围为 $-2^{n-1} \sim +(2^{n-1}-1)$, 恰好是 2^n 个数。所以计算机中常用补码表示数。

补码的运算规则如下。

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

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

例 1-9 $X=28, Y=-73$, 求 $[X+Y]_{\text{补}}, [X]_{\text{补}}+[Y]_{\text{补}}, [X-Y]_{\text{补}}, [X]_{\text{补}}-[Y]_{\text{补}}$ 。

解: $[X+Y]_{\text{补}} = [-45]_{\text{补}} = 11010011\text{B}$

$$[X]_{\text{补}} + [Y]_{\text{补}} = [28]_{\text{补}} + [-73]_{\text{补}} = 00011100\text{B} + 10110111\text{B} = 11010011\text{B}$$

$$[X-Y]_{\text{补}} = [101]_{\text{补}} = 01100101\text{B}$$

$$[X]_{\text{补}} - [Y]_{\text{补}} = [28]_{\text{补}} + [73]_{\text{补}} = 00011100\text{B} + 01001001\text{B} = 01100101\text{B}$$

上面的例子说明采用补码来表示数, 在计算机的加、减法运算中, 不必判断数的正负, 只要符号位参加运算就能自动得到正确的结果。

4. 移码

所谓移码是将真值在数轴上往正方向平移了 2^{n-1} 。对于 8 位数, 平移了 $2^7=128$ 。移码也被称为余码、增码或偏进二进码。移码中, 符号的表示方法与原码、反码和补码相反, 即符号位为“1”时表示正数, 为“0”时表示负数; 其他位与补码相同。所以求一个数的移码时只需将其二进制补码的符号位取反即可。

例 1-10 计算机的字长为 8 位, 写出 $+4$ 、 -4 的补码和移码。

$$\text{解: } [+4]_{\text{补}} = 00000100\text{B} \quad [+4]_{\text{移}} = 10000100\text{B}$$

$$[-4]_{\text{补}} = 11111100\text{B} \quad [-4]_{\text{移}} = 01111100\text{B}$$

1.2.3 计算机中常用的编码

1. BCD 码

BCD(Binary-Coded Decimal)码又称为二—十进制编码或 8421 码, 其方法是用 4 位二进制数表示 1 位十进制数, 自左至右每一位对应的位权是 8、4、2、1。十进制的 0~9 分别用 BCD 码的 0000~1001 表示, 而不是整个十进制数转换成二进制形式。BCD 码有压缩和非压缩两种形式。压缩 BCD 码(Packed BCD)用 1 个字节表示 2 位 BCD 码; 非压缩 BCD 码(Unpacked BCD)用 1 个字节的低 4 位来表示 1 位 BCD 码, 高 4 位为 0(无意义)。

例 1-11 写出 13、1 622 的压缩和非压缩 BCD 码。

解: 十进制	压缩 BCD 码	非压缩 BCD 码
13	0001 0011	00000001 00000011
1 622	00010110 00100010	00000001 00000110 00000010 00000010