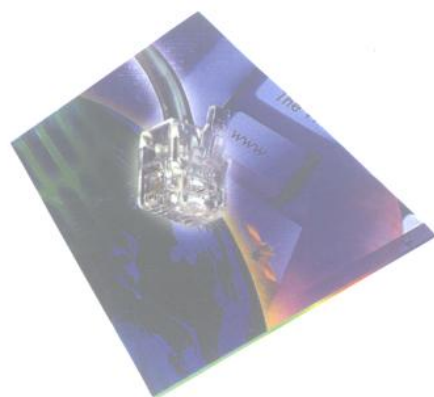


高等院校选用教材系列



从 8086 到 Pentium III 微型计算机及接口技术



马维华 奚抗生 编著
易仲芳 毛建国



科学出版社
Science Press

从 8086 到 Pentium III

微型计算机及接口技术

马维华 奚抗生
易仲芳 毛建国 编著

科学出版社

内 容 简 介

本书以 Intel 处理器为核心的微型计算机为背景,全面、系统、深入地介绍了微型计算机的基本组成、工作原理和实际应用,注重吸取微机发展的最新技术和最新知识,并将其融于全书之中。

全书共分 10 章,分别介绍微型计算机的基本知识、从 8086 到 Pentium III 循序渐进地介绍了微处理器的内部结构及特点、指令系统、汇编语言程序设计、存储器、输入输出基础、中断系统、通信与接口技术、微机总线及其接口标准和微型计算机系统板等。每章后面都有大量思考题与习题。本书可作为高等院校各个专业本、专科教材,也可作为工程技术人员的参考书。

图书在版编目(CIP)数据

从 8086 到 Pentium III 微型计算机及接口技术/马维华等编著.北京:科学出版社,2000.2

ISBN 7-03-007843-8

I. 从… II. 马… III. ①微型计算机-基本知识②微型计算机-接口-技术 IV. TP38

中国版本图书馆 CIP 数据核字(1999)第 71280 号

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

新 蕾 印 刷 厂 印 刷

新华书店北京发行所发行 各地新华书店经售

*

2000 年 2 月第 一 版 开本:787×1092 1/16

2000 年 2 月第一次印刷 印张:27 1/4

印数:1—6 000 字数:627 000

定价:36.00 元

(如有印装质量问题,我社负责调换〈北燕〉)

前 言

微型计算机原理及应用是工科学生学习和掌握微型计算机硬件知识和汇编语言程序设计的重要课程,课程的任务是使学生能从理论和实践上掌握微机的基本组成、工作原理、接口技术及硬件连接,建立微机系统的整机概念,使学生初步具有微机系统软、硬件开发的能力。

然而,微型计算机的新技术、新机型、新应用层出不穷,日新月异,要达到上述课程任务所提出的要求,就要既符合教学体系的连贯性要求,又要把握微型计算机发展的脉搏。由于微机发展一日千里,以致于微机原理的课本远远滞后于微型计算机的发展,为克服这一局限性,我们在南京航空航天大学原自编教材的基础上,结合多年教学实践,大量汲取最新资料,编写成本书。

本书共分 10 章,第 1 章概述微型计算机的基础知识,第 2 章介绍微机的核心部件——微处理器,第 3 章为微处理器的指令系统,第 4 章介绍汇编语言程序设计,第 5 章介绍存储器,第 6 章介绍输入输出基础,第 7 章为微机的中断系统,第 8 章介绍通信与接口技术,第 9 章为微机总线及其接口标准,第 10 章介绍微机的系统板。

为适应当前微机发展的新形势,本书在内容上除了力求讲清基本原理外,还尽可能地反映微机最新技术。例如微处理器一章从 8086/8088 到 Pentium/Pentium Pro/MMX 与 Pentium II 以及 Pentium III,详细介绍了 Intel 公司的系列微处理器;指令系统一章,除了介绍 8086/8088 基本指令集外,还对 80286、80386、80486 到 Pentium 等新增指令作了必要的介绍;存储器一章在传统半导体存储器的基础上,对微机内存层次结构、内存管理、CMOS RAM、ROM BIOS、Shadow 等作了必要介绍;中断系统一章不限于 8086 中断系统,充分考虑当前处理器保护方式下的中断与异常;接口技术部分,介绍串并行通信的概念,并行接口、串行接口、定时器/计数器接口、A/D、D/A 转换接口、打印机接口、显示器接口、软硬盘接口和调制解调器等;总线一章全面介绍系列扩展总线,从 8 位的 XT、16 位的 ISA,到 32 位的 EISA、VESA,可扩充为 64 位的 PCI 及以 Pentium II 以后发展起来的 AGP;最后从微机硬件系统角度介绍系统的系统板,并对微型计算机的硬件新技术作了必要介绍。

本书特别注重实用性,在介绍具体内容时,尽量列举实例,有些程序段和接口电路可直接用于实际系统中;在叙述上力求深入浅出,通俗易懂。

本书内容全面、新颖、实用,除可作为微型计算机原理和应用、微型计算机系统及应用、微型计算机技术与接口、微型计算机与接口通信等课程的本、专科教材外,也可作为从事微机硬软件开发人员、工程技术人员以及需要了解微型计算机技术者的参考书。对于参加全国计算机等级考试(三级偏硬,3A)的读者也有较大帮助。作为本、专科教材使用时,通过适当选择,既适合计算机专业的学生使用,也适用于非计算机专业。如果学时有限,可以根据教学要求选择一定章节学习,有些章节可以自学。为了让学生巩固学习的知

识,每一章后面都有大量思考题和习题。

本书第 1 章由易仲芳编写,第 3、4 章以及 7.3 节由奚抗生编写并录入,第 2、5、6、7、9 和 10 章由马维华编写并录入,第 8 章由毛建国编写,全书由马维华统稿。陈鸿茂老师提出了许多宝贵意见,高航、李遥、严青华、张有成等在使用原教材中也提出了宝贵意见和建议。另外,在编写本书过程中得到徐涛、谭白磊等的大力支持和帮助,在此一并表示感谢。

由于计算机技术飞速发展,加之作者水平有限和时间仓促,书中难免有疏漏和错误之处,恳请同行专家及读者提出批评意见。

编 者

1999 年 6 月

目 录

前言

第 1 章 概述	(1)
1.1 微型计算机的发展概况	(1)
1.2 计算机中信息的表示	(2)
1.2.1 机器数和真值	(3)
1.2.2 原码、反码和补码	(4)
1.2.3 补码的运算	(7)
1.2.4 定点数与浮点数	(8)
1.2.5 微型计算机中常用码制	(11)
1.2.6 计算机中图、声、像的表示	(14)
1.3 微型计算机的硬件结构和基本工作原理	(14)
1.3.1 微型计算机的基本结构	(14)
1.3.2 中央处理器(CPU)	(15)
1.3.3 存储器	(17)
1.3.4 输入输出接口及外部设备	(19)
1.3.5 总线	(21)
1.4 微型计算机的软件系统	(21)
1.4.1 软件的分类	(21)
1.4.2 操作系统的概念	(22)
1.4.3 计算机语言和语言处理程序	(23)
1.4.4 支持软件	(25)
1.5 微型计算机系统及性能指标	(26)
1.5.1 计算机系统的组成	(26)
1.5.2 以 Intel 处理器为核心的微型计算机系统	(26)
1.5.3 微型计算机的主要性能指标	(28)
思考与习题	(29)
第 2 章 微处理器	(31)
2.1 8086/8088 微处理器	(32)
2.1.1 8086/8088 的内部结构	(32)
2.1.2 8086/8088 的引脚信号与工作模式	(37)
2.1.3 8086/8088 的时序	(42)
2.1.4 8086/8088 的存储器及 I/O 组织	(45)
2.2 80286 微处理器	(46)
2.2.1 80286 的主要特点	(46)
2.2.2 80286 的功能结构	(47)
2.2.3 80286 的编程结构	(48)
2.2.4 80286 保护方式下的存储器寻址	(49)

2.2.5 引脚信号与总线周期	(53)
2.3 80386 微处理器	(54)
2.3.1 80386 概况	(54)
2.3.2 80386 的内部结构	(55)
2.3.3 80386 的三种工作方式	(56)
2.3.4 80386 寄存器	(58)
2.3.5 80386 的存储器地址方式	(63)
2.3.6 80386 引脚信号与总线周期	(67)
2.4 CISC 与 RISC	(69)
2.4.1 CISC	(70)
2.4.2 RISC	(70)
2.5 80486 微处理器	(72)
2.5.1 80486 概况	(72)
2.5.2 80486 的内部结构及工作原理	(73)
2.5.3 80486 的引脚信号	(75)
2.5.4 80486 的内部寄存器	(76)
2.5.5 80486 的指令流水线与总线周期	(78)
2.6 Pentium 微处理器	(78)
2.6.1 Pentium 微处理器概况	(78)
2.6.2 Pentium 微处理器的内部结构及工作原理	(80)
2.7 Pentium Pro 微处理器	(83)
2.8 MMX 及 MMX Pentium 微处理器	(84)
2.8.1 MMX 与多能奔腾处理器	(84)
2.8.2 MMX 的特点	(84)
2.9 Pentium II 和 Pentium III 微处理器	(85)
2.9.1 Pentium II 微处理器	(85)
2.9.2 Pentium III 微处理器	(88)
2.10 新一代 64 位微处理器 Merced	(90)
思考与习题	(90)
第 3 章 80X86 指令系统	(95)
3.1 指令格式与编码	(95)
3.2 寻址方式	(96)
3.2.1 立即寻址	(96)
3.2.2 寄存器寻址	(97)
3.2.3 存储器寻址	(97)
3.3 8086/8088 指令系统	(99)
3.3.1 数据传送指令	(99)
3.3.2 算术运算指令	(103)
3.3.3 逻辑指令	(109)
3.3.4 串处理指令	(111)
3.3.5 控制转移指令	(113)
3.3.6 处理器控制指令	(121)
3.4 80286 增强和扩充的指令	(122)

3.5 80386 增强和扩充的指令	(124)
3.6 80486 和 Pentium 增强和扩充的指令	(127)
思考与习题	(128)
第4章 汇编语言程序设计	(130)
4.1 概述	(130)
4.1.1 汇编语言的基本概念	(130)
4.1.2 80X86 宏汇编语言的基本语法	(131)
4.1.3 汇编程序和汇编处理过程	(133)
4.1.4 汇编语言的功能和特点	(137)
4.2 80X86 宏汇编语言的数据和表达式	(137)
4.2.1 常量	(138)
4.2.2 变量	(138)
4.2.3 标号	(139)
4.2.4 表达式和运算符	(139)
4.3 80X86 宏汇编语言的伪指令语句	(144)
4.3.1 基本伪指令语句	(144)
4.3.2 高级伪指令语句	(154)
4.3.3 伪指令的增强与扩充	(161)
4.4 汇编语言程序设计方法	(166)
4.4.1 汇编语言程序设计的基本方法	(166)
4.4.2 顺序程序设计	(169)
4.4.3 分支程序设计	(170)
4.4.4 循环程序设计	(174)
4.4.5 子程序设计	(181)
思考与习题	(189)
第5章 微型计算机中的存储器	(193)
5.1 存储器概述	(193)
5.2 半导体存储器分类及性能指标	(193)
5.2.1 半导体存储器分类	(193)
5.2.2 半导体存储器的技术指标	(194)
5.3 随机存取存储器	(195)
5.3.1 静态随机存取存储器(SRAM)	(195)
5.3.2 动态随机存取存储器(DRAM)	(199)
5.3.3 集成随机存取存储器(IRAM)	(201)
5.3.4 高速 RAM	(201)
5.4 只读存储器	(204)
5.4.1 掩膜型只读存储器(MROM)	(204)
5.4.2 可编程只读存储器(PROM)	(205)
5.4.3 可擦除可编程只读存储器(EPROM)	(205)
5.4.4 电可擦除可编程只读存储器(EEPROM)	(207)
5.4.5 闪速存储器(Flash Memory)	(207)
5.5 存储器的扩展	(208)
5.5.1 位扩展	(208)

5.5.2 字扩展	(209)
5.5.3 字位全扩展	(210)
5.6 微机内存层次结构	(212)
5.6.1 内存层次结构	(212)
5.6.2 主存储器	(212)
5.6.3 高速缓冲存储器	(215)
5.6.4 虚拟内存	(218)
5.7 微机系统中的其他存储部件	(219)
5.7.1 CMOS RAM	(219)
5.7.2 ROM BIOS	(221)
5.7.3 Shadow RAM	(225)
5.8 微型计算机系统的内存管理	(226)
5.8.1 微机系统中的内存分类	(226)
5.8.2 扩展内存管理	(228)
5.8.3 扩充内存管理	(229)
5.8.4 微机常用操作系统的内存管理	(231)
思考与习题	(232)
第 6 章 输入输出基础	(236)
6.1 概述	(236)
6.1.1 输入/输出与输入/输出接口	(236)
6.1.2 I/O 接口的功能	(236)
6.1.3 CPU 与 I/O 设备之间的接口信息	(237)
6.1.4 I/O 端口的编址方法	(238)
6.2 输入输出控制方式	(239)
6.2.1 直接程序控制方式	(239)
6.2.2 中断控制方式	(241)
6.2.3 DMA 控制方式	(241)
6.3 I/O 接口的基本结构及特点	(243)
6.3.1 I/O 接口的基本结构	(243)
6.3.2 I/O 接口的特点	(244)
6.4 I/O 接口的读写技术	(245)
6.4.1 I/O 端口的地址分配	(245)
6.4.2 I/O 端口的译码	(246)
6.4.3 端口的读写控制	(249)
6.5 DMA 控制器	(252)
6.5.1 DMA 控制器 8237 结构及引脚	(252)
6.5.2 8237 内部寄存器功能及格式	(255)
6.5.3 DMA 控制器的编程及应用	(258)
6.5.4 DMA 读写时序	(262)
思考与习题	(264)
第 7 章 微型计算机的中断系统	(267)
7.1 中断的基本概念	(267)
7.1.1 中断与异常	(267)

7.1.2 中断分类	(268)
7.1.3 实地址方式与保护地址方式下的中断	(271)
7.1.4 外部中断源的管理	(274)
7.2 可编程中断控制器 8259A	(276)
7.2.1 8259A 的内部结构及工作原理	(276)
7.2.2 8259A 的引脚信号	(279)
7.2.3 8259A 的工作方式	(280)
7.2.4 8259A 的编程方法	(282)
7.2.5 高级中断控制器	(289)
7.3 中断调用及中断程序设计	(290)
7.3.1 BIOS 中断调用	(290)
7.3.2 DOS 系统功能调用	(291)
7.3.3 中断程序设计	(292)
思考与习题	(298)
第 8 章 微机通信与接口技术	(301)
8.1 并行通信与串行通信	(301)
8.1.1 并行通信与并行接口	(301)
8.1.2 串行通信与串行接口	(302)
8.1.3 串行通信方式及异步通信协议	(304)
8.1.4 串行异步通信标准接口	(307)
8.2 可编程串行通信接口芯片 16550 与 8250	(312)
8.2.1 16550 及 8250 的内部结构及引脚	(313)
8.2.2 串行接口的编程及应用	(319)
8.3 可编程并行通信接口芯片 8255	(325)
8.3.1 8255 的内部结构及引脚信号	(325)
8.3.2 8255 的工作方式	(326)
8.3.3 8255 的编程	(330)
8.3.4 键盘和显示器接口	(334)
8.4 可编程定时/计数器芯片 8253 与 8254	(340)
8.4.1 8253 和 8254 的内部结构及引脚信号	(340)
8.4.2 8253 和 8254 的工作方式	(342)
8.4.3 8253 和 8254 的编程方法	(346)
8.4.4 8253 应用	(348)
8.5 A/D 与 D/A 转换接口	(350)
8.5.1 D/A 转换器	(350)
8.5.2 A/D 转换器	(360)
8.6 打印机接口	(368)
8.6.1 打印机适配器组成	(368)
8.6.2 打印机接口的应用	(371)
8.7 显示适配器接口	(372)
8.7.1 显示适配器接口概述	(372)
8.7.2 显示适配器的 I/O 寄存器	(375)
8.7.3 对显示适配器的编程	(376)

8.8 软硬盘接口	(380)
8.8.1 软盘控制器接口	(380)
8.8.2 硬盘适配器接口	(382)
8.9 调制解调器	(384)
8.9.1 信号的调制与解调	(384)
8.9.2 调制解调器	(386)
8.10 简单实用的双机通信	(387)
思考与习题	(389)
第 9 章 微机总线及其接口标准	(393)
9.1 总线概述	(393)
9.1.1 总线的分类	(393)
9.1.2 总线的主要参数	(394)
9.2 总线层次及信号类型	(394)
9.2.1 总线系统的层次	(394)
9.2.2 总线系统的信号类型	(395)
9.3 ISA 总线	(395)
9.3.1 ISA 总线特性及接口信号	(395)
9.3.2 基于 ISA 总线接口的设计要点	(398)
9.4 EISA 总线	(399)
9.5 VESA 总线	(400)
9.5.1 局部总线	(400)
9.5.2 VESA 总线	(400)
9.6 PCI 总线	(400)
9.6.1 PCI 总线的特点及接口信号	(401)
9.6.2 PCI 总线的应用	(404)
9.7 AGP	(404)
思考与习题	(405)
第 10 章 微型计算机系统板	(407)
10.1 系统板概述	(407)
10.1.1 系统板的基本组成	(407)
10.1.2 微型计算机系统板的种类	(408)
10.2 PC/XT 和 286 系统板	(408)
10.2.1 PC/XT 系统板	(408)
10.2.2 286 系统板	(409)
10.3 386 和 486 系统板	(410)
10.3.1 386 系统板	(410)
10.3.2 486 系统板	(410)
10.4 Pentium 和 Pentium II/III 系统板	(411)
10.4.1 奔腾系统板	(411)
10.4.2 Pentium II/III 系统板	(412)
10.4.3 ATX 系统板	(413)
10.5 系统板重要元器件及接口功能介绍	(414)
10.5.1 主板上的主要芯片	(415)

10.5.2 主板上的重要插(槽)座	(416)
10.6 微机硬件新技术	(418)
10.6.1 关于存储器的一系列新技术	(418)
10.6.2 关于主板芯片组的新技术	(419)
10.6.3 关于硬盘的新技术	(419)
10.6.4 其他新技术	(420)
思考与习题	(422)
参考文献	(423)

第 1 章 概 述

1.1 微型计算机的发展概况

自从 70 年代初期以来,随着微电子技术的飞速发展,大规模和超大规模集成电路芯片不断涌现,以微处理器为核心的微型计算机在计算机的发展洪流中取得了重大突破,它对计算机的发展、应用和普及带来了极其深刻的影响。因此,有人说微处理器及微型计算机的出现和崛起是计算机的第二次革命,这确实有一定道理。

下面,首先说明几个基本概念,然后进一步介绍微处理器、微型计算机以及微型计算机系统的发展概况。

微处理器(Microprocessor,简称 MP 或 μP)通常是指在一块大规模或超大规模集成电路芯片上,把冯·诺依曼计算机体系结构中的运算器和控制器集成进去。这样,它虽然不是通常所指的微型计算机,但却是微型计算机的核心部件(即中央处理器 CPU)。近几年来,随着微电子和超大规模集成技术的迅速发展,在它的内部不仅包括运算器和控制器等基本部件,而且已经把数学协处理器、高速缓冲存储器以及多种接口和控制部件,甚至把多媒体部件也集成到一块芯片内。因此,其功能之强大,是完全可以想象的。

微型计算机(Microcomputer,简称 MC 或 μC)是以微处理器为核心,配上用大规模集成电路制作的存储器、输入输出接口电路以及系统总线等部件的“裸机”,它包含冯·诺依曼计算机体系结构中的五个部件,通常把它称为微型计算机,或简称微型机。特别要指出的是,为了进一步微型化,在微型计算机的发展过程中,还出现了单片计算机和单板计算机,其中前者是将微型计算机的所有部件全部集成在一块芯片上,而后者则是将微型计算机的各个部件安装在一块印制电路板上,从而使微型计算机更适合于小型化的应用场合。

微型计算机系统(Microcomputer System,简称 MCS 或 μCS)是以微型计算机为核心,配置相应的外部设备和系统软件及应用软件,从而使其具有独立的数据处理和运算能力的设备,通常把它称为微型计算机系统。也就是说,微型计算机系统是微型计算机硬件、软件以及外部设备的集合,使裸机变成一台完整的、可供用户直接使用的计算或控制设备。

通常,根据微处理器的字长和功能,可以将微型计算机的发展划分为以下几个阶段:

第一阶段(1971~1973 年)是 4 位或 8 位低档微处理器和微型计算机时代,通常称之为第一代,其典型产品是 Intel 4004 和 Intel 8008 微处理器以及分别由它们组成的 MCS-4 和 MCS-8 微型计算机,它们的基本特点是采用 PMOS 工艺,集成度低(1200~2000 晶体管/片),系统结构和指令系统均比较简单,主要采用机器语言或简单的汇编语言,基本指令的执行时间约 10~20 μs ,用于家用电器和简单的控制场合。

第二阶段(1974~1978 年)是 8 位中高档微处理器和微型计算机时代,通常称之为第二代,其典型产品是 Intel 公司的 8080/8085、Motorola 公司的 MC6800 和 Zilog 公司的 Z80 等微处理器,以及各种 8 位单片计算机,如 Intel 公司的 8048、Motorola 公司的

MC6801 和 Zilog 公司的 Z8 等,它们的特点是采用 NMOS 工艺,集成度提高 4 倍左右(5000~9000 晶体管/片),运算速度提高 10~15 倍(基本指令的执行时间约 $1\sim 2\mu\text{s}$),指令系统比较完善,已具有典型的计算机体系结构以及中断、DMA 等控制功能,软件除采用汇编语言外,还配有 BASIC、FORTRAN、PL/I 等高级语言及其相应的解释程序和编译程序,并在后期配上了操作系统,如 CP/M 就是当时世界上流行的一种微机操作系统。

第三阶段(1978~1984 年)是 16 位微处理器和微型计算机时代,通常称之为第三代,其典型产品是 Intel 公司的 8086/8088、80286、Motorola 公司的 MC68000 和 Zilog 公司的 Z8000 等微处理器。它们的特点是采用 HMOS 工艺,集成度(20 000~70 000 晶体管/片)和运算速度(基本指令的执行时间约 $0.5\mu\text{s}$)比第二代产品提高了一个数量级,指令系统更加丰富、完善,采用多级中断系统、多种寻址方式、段式存储器结构、硬件乘除部件,并配置了强有力的软件系统。这一时期的微机有 IBM PC/XT(8088)、IBM PC/AT(80286)及其兼容机等。

第四阶段(1985~1991 年)是 32 位微处理器和微型计算机时代,通常称之为第四代,其典型产品是 Intel 公司的 80386/80486、Motorola 公司的 MC68030/68040 等微处理器,以及相应的 IBM PC 兼容机,如 386、486 等,它们的特点是采用 HMOS 或 CMOS 工艺,集成度高达 100 万晶体管/片以上,具有 32 位数据和 32 条地址总线,工作主频达 25MHz 以上,基本指令的工作速度达到或超过 25MIPS(Million of Instructions Per Second)。微型计算机的功能已经达到甚至超过超级小型机,完全可以胜任多用户、多任务的作业。

第五阶段(1992 年以后)是奔腾系列微处理器和奔腾系列微型计算机时代,通常称之为第五代,其典型产品是 Intel 公司的 Pentium,内部采用超标量指令流水线结构,并具有相互独立的指令和数据高速缓存,工作频率愈来愈高,基本指令的工作速度越来越高。随着 MMX(Multi-Media Extended)和 Pentium II/Pentium III 微处理器的出现,使微机的发展在网络化、多媒体化和智能化等方面跨上了更高的台阶。而且这一阶段的以 Pentium 命名的处理器不断改进,主要有 Pentium、Pentium Pro、MMX Pentium 以及 Pentium II 和 Pentium III 等。

应该指出,Pentium、Pentium Pro 以及具有 MMX 技术的 Pentium 和 Pentium II/Pentium III,尽管它们的内部与工作寄存器直接相连的是 32 位数据宽度,但外部数据总线却是 64 位。

下一阶段的处理器将是 64 位时代,如 Intel 的 Merced 处理器。

下面将介绍计算机中信息的表示、微型计算机的硬件结构、软件系统以及微型计算机系统的基本概念,从而为学习后续章节打下基础。

1.2 计算机中信息的表示

电子数字计算机是一个自动化信息处理机,能处理文字、数据,还能处理声音、图形和图像等各种信息,它们统称数据。具有数值大小和正负特征的数据,称数值数据;而文字、图形、声像之类的并无数值大小和正负特征的信息称非数值数据。两者在计算机内部都是以二进制数即用 0、1 以不同形式编码表示与存储的。

1.2.1 机器数和真值

在计算机中,只能表示0和1两种数码,为了表示正数和负数,专门选择一位来表示数的符号,即选择最高位作为符号位,当该位为“0”时,表示正数;当该位为1时,表示负数。

这就是说,数的符号在计算机中也数码化了。我们把一个数在机器(计算机)中的表示形式称为机器数,而把原来实际数本身的值叫真值(机器数真值)。真值可以用二进制数表示,也可用十进制数表示。两个机器数 N_1 和 N_2 如图 1.1 所示。

数位	D ₁₅	D ₁₄	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
N ₁	0	1	0	1	0	1	1	0	0	1	0	1	1	0	1	0
N ₂	1	1	0	1	0	1	1	0	0	1	0	1	1	0	1	0
说明	符号位	数 值 位														

图 1.1 机器数示例

N_1 对应的真值为 +101011001011010B 即 +22106;

N_2 对应的真值为 -101011001011010B 即 -22106。

机器数有如下特点:

1. 机器数的正负号数值化。
2. 机器数所能表示数的范围受到机器(计算机)字长的限制。

那么什么是计算机字长呢?我们先来解释几个计算机中常用的术语。

(1)位(Bit):位是计算机所能表示的最小数据单位,它只能有二种状态“0”和“1”。要想表示更大的数,就得把更多的位组合起来作为一个整体,每增加一位,所能表示的数就增大一倍。

(2)字节(Byte):一个8位二进制数称为一个字节,是数据处理的基本单位,即以字节为单位存储和解释信息。微型计算机存储器是由一个个存储单元构成的,每个存储单元的大小就是一个字节。所以存储器均以字节单位编址,存储容量大小也以字节数来度量。常使用的度量单位有 KB、MB、GB 和 TB,其大小分别为:

$$1\text{KB} = 2^{10} = 1024 \text{ Bytes(字节)}$$

$$1\text{MB} = 2^{10} \text{KB} = 2^{20} \text{字节} = 1048576 \text{ 字节}$$

$$1\text{GB} = 2^{10} \text{MB} = 2^{30} \text{字节} = 1073741824 \text{ 字节}$$

$$1\text{TB} = 2^{10} \text{GB} = 2^{40} \text{字节} = 1099511627776 \text{ 字节}$$

(3)字(Word):计算机处理数据时,CPU 通过内部数据总线一次存取、加工和传送的数据长度称为字,通常习惯地将一个16位二进制数(2个字节)称为一个字,四个字节称为双字。

(4)字长(Word Length):指字的二进制数的位数。8位微处理器的字长为8位,每个字由一个字节构成;而在16位微处理器中,每个字由2个字节构成;在32位微处理器中,每个字由4个字节构成;在64位微处理器中,每个字由8个字节构成。由于字长是计算

机一次所能处理的实际位数的长度,所以字长是衡量计算机性能的一个重要标志。字长越长,精度越高。

3. 小数点不能直接标出,需要按一定方式约定小数点的位置。

1.2.2 原码、反码和补码

既然一个数的数值和符号全都是数码,那么当对这种机器数进行运算时,符号位怎样处理?能不能也同数值位一道参加运算呢?为妥善解决这些问题,引出了机器数的三种不同的编码形式,即原码、反码和补码。分别记作: $[X]_{\text{原}}$ 、 $[X]_{\text{反}}$ 和 $[X]_{\text{补}}$ 。

一、原码

设 $X = X_1, X_2, \dots, X_{n-1}$, 其中 X_i 为一位二进制数, $i = 1, 2, \dots, (n-1)$ 。

则

$$[X]_{\text{原}} = \begin{cases} 0 & X_1 & X_2 & \cdots & X_{n-1} & \text{当 } X \geq 0 \\ 1 & X_1 & X_2 & \cdots & X_{n-1} & \text{当 } X \leq 0 \end{cases}$$

即一个数的原码,就是数值部分不变,而用 0 和 1 分别来表示数的符号“+”和“-”的机器数。在下面的表述中用尾部 B 表示二进制数, H 表示十进制数。

【例】 $X_1 = 67 = +1000011\text{B}$ $[X_1]_{\text{原}} = 01000011\text{B}$
 $X_2 = -67 = -1000011\text{B}$ $[X_2]_{\text{原}} = 11000011\text{B}$

在原码表示法中,根据定义,数 0 的原码有两种不同形式(设字长为 8 位):

$$[+0]_{\text{原}} = 00000000\text{B}, \quad [-0]_{\text{原}} = 10000000\text{B}$$

原码表示简单易懂,而且与真值的转换方便。但原码表示的数不便于计算机运算,因为在两原码数运算时,首先要判断它们的符号,然后再决定用加法还是用减法,因而使机器的结构相应地复杂化或增加机器的运算时间。为解决上述弊病,引入反码和补码表示法。

二、反码

设 $X = X_1, X_2, \dots, X_{n-1}$, 其中 X_i 为一位二进制数, $i = 1, 2, \dots, (n-1)$, 则

$$[X]_{\text{反}} = \begin{cases} 0 & X_1 & X_2 & \cdots & X_{n-1} & \text{当 } X \geq 0 \\ 1 & \overline{X_1} & \overline{X_2} & \cdots & \overline{X_{n-1}} & \text{当 } X \leq 0 \end{cases}$$

式中, $\overline{X_i}$ 表示对 X_i 取反。

对于正数,反码与原码相同,符号位为 0,其余位为数值位本身;对于负数,反码的符号位为 1,其余位为数值位按位取反(即将 1 变为 0, 0 变为 1)。

例如: $X_1 = 83 = +1010011\text{B}$, $[X_1]_{\text{反}} = 01010011\text{B}$
 $X_2 = -83 = -1010011\text{B}$, $[X_2]_{\text{反}} = 10101100\text{B}$

在反码表示法中,根据定义,数 0 的反码有两种不同形式(设字长为 8 位):

$$[+0]_{\text{反}} = 00000000\text{B}, \quad [-0]_{\text{反}} = 11111111\text{B}$$

三、补码

引入补码的概念是在于将加、减运算简化为单纯的相加运算。

1. 同余的概念和补码

设有两个数 $a=17$, $b=27$, 若用 10 去除 a 和 b , 则它们的余数均为 7, 称 17 和 27 在以 10 为模时是同余的, 并记作

$$17 = 27 \pmod{10}$$

或者说 17 和 27 在以 10 为模时是相等的。此处的模就是一个计量系统所能表示的最大量程(或一个计量单位称为模或模数)。

由同余概念, 不难得出

$$a + M = a \pmod{M}$$

$$a + 2M = a \pmod{M}$$

因此当 a 为负数时, 如 $a = -4$, 在以 10 为模时, 有

$$-4 + 10 = -4 \pmod{10}$$

即 $6 = -4 \pmod{10}$

上式说明, 在以 10 为模时, -4 与 $+6$ 是相等的。我们称 $+6$ 为 -4 的补码, 或者说 $+6$ 与 -4 对模 10 来说互为补数。有了补码概念, 就可将减法转化为加法(加补码)来进行。

【例】 $7 - 4 = 7 + 6 \pmod{10}$

该式说明, 在以 10 为模时, 7 减 4 可以通过 7 加 -4 的补码 6 来进行, 而所得的结果是相同的(只要在加补码 6 时, 将所产生的进位舍弃即可, 这正是以 10 为模的意思)。

2. 补码的求法

设 $X = X_1, X_2, \dots, X_{n-1}$, 其中 X_i 为一位二进制数, $i = 1, 2, \dots, (n-1)$ 。

$$[X]_{\text{补}} = \begin{cases} 0 & X_1 & X_2 & \cdots & X_{n-1} & \text{当 } X \geq 0 \\ 1 & \overline{X_1} & \overline{X_2} & \cdots & \overline{X_{n-1}} + 1 & \text{当 } X \leq 0 \end{cases}$$

$$[X]_{\text{补}} = \begin{cases} X & \text{当 } 0 \leq X < 2^n - 1 \\ 2^n + X & \text{当 } -2^n - 1 \leq X < 0 \pmod{2^n} \end{cases}$$

也就是说, 正数的补码等于原码, 负数的补码等于该数反码加 1。这里限定 X 为整数。

(1) 正数补码

$$[X]_{\text{原}} = [X]_{\text{反}} = [X]_{\text{补}}$$

【例】 $[+127]_{\text{原}} = [+127]_{\text{反}} = [+127]_{\text{补}} = 01111111\text{B}$ (假设字长为 8 位)

(2) 负数的补码

设 $X = -25 = -0011001\text{B}$, $[X]_{\text{原}} = 10011001\text{B}$

$$[X]_{\text{反}} = 11100110\text{B}$$

$$[X]_{\text{补}} = [X]_{\text{反}} + 1 = 11100110 + 1 = 11100111\text{B}$$

即先求负数的反码, 然后在反码最低一位上加 1, 就得该负数的补码。

还可以用更简便方法求得负数 X 的补码, 方法是: 符号位为 1, 将原数值中最右边一个 1 及其后面的 0 保持不变, 而最右一个 1 以左的各位按位取反。

$$X_1 = -0011000\text{B} \quad X_2 = -10000000\text{B}$$

$$[X_1]_{\text{补}} = 11101000\text{B} \quad [X_2]_{\text{补}} = 11000000\text{B}$$