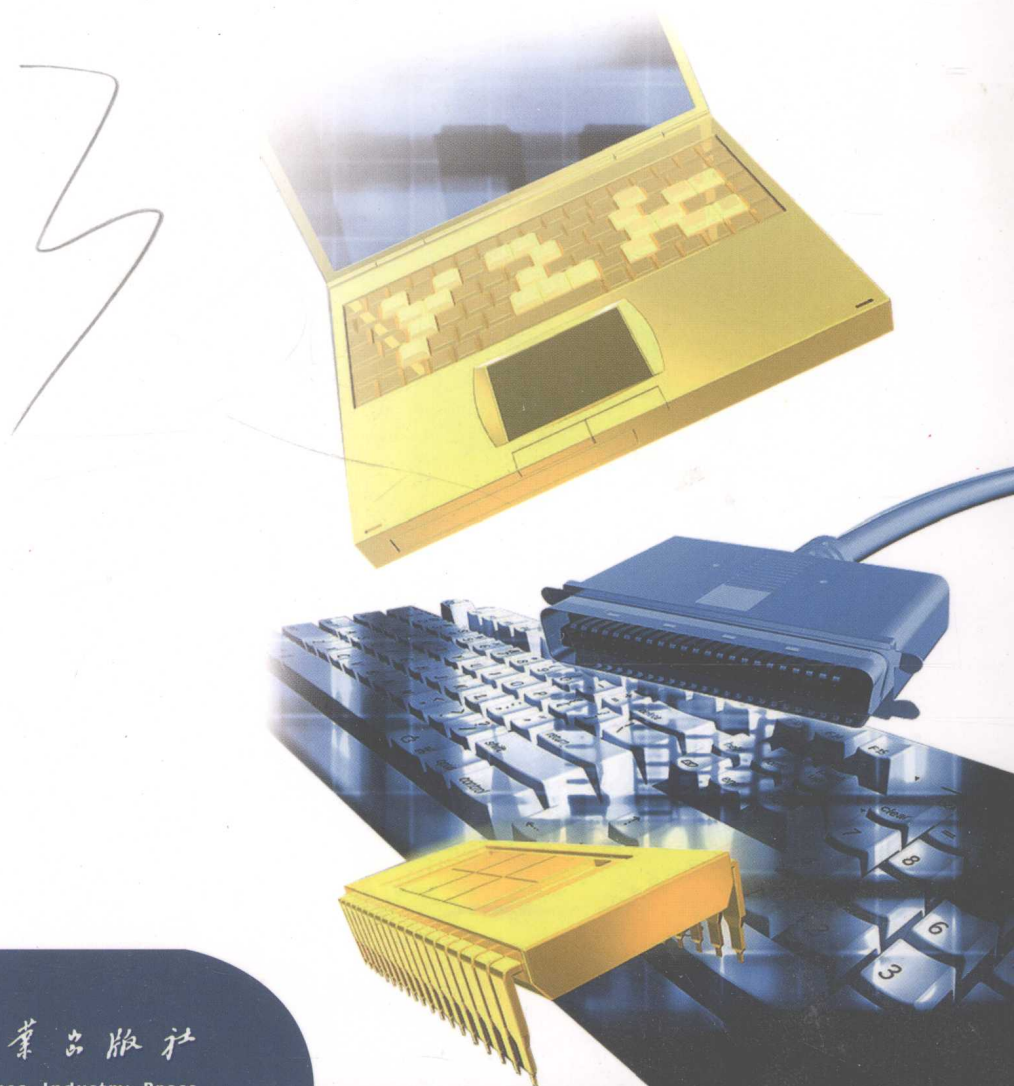


TP36
138



微型计算机 原理与接口技术

王宁宁 张瑾 谭彦彬 陈铨 编著



国防工业出版社
National Defense Industry Press

微型计算机原理与接口技术

王宁宁 张瑾 谭彦彬 陈钺 编著

国防工业出版社

· 北京 ·

内 容 简 介

本书以 Intel 8086 为背景,从微型计算机系统的角度讲解微型计算机的工作原理、指令系统、汇编语言程序设计、存储系统、输入/输出、中断技术及可编程接口芯片,在此基础上讲述高性能微处理器的技术发展以及总线技术,最后介绍常用外围设备的工作原理。

本书基于作者多年从事教学及科研工作的实践,循序渐进、深入浅出,力求化解学生所反映的难点问题,使教材内容更符合学生的认知过程,便于教学及自学。

本书是高等院校非计算机专业的本科生教材,可作为成人高等教育的培训教材,也可供广大科技工作者自学参考。

图书在版编目(CIP)数据

微型计算机原理与接口技术 / 王宁宁等编著. —北京:

国防工业出版社, 2009. 6

ISBN 978-7-118-06193-2

I. 微... II. 王... III. ①微型计算机—理论②微型计算机—接口 IV. TP36

中国版本图书馆 CIP 数据核字(2009)第 014803 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

腾飞印务有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 25 字数 578 千字

2009 年 6 月第 1 版第 1 次印刷 印数 1—4000 册 定价 46.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

前 言

《微型计算机原理与接口技术》是理工科机电类各专业本科生的一门必修课程。通过本课程的学习,使学生基本掌握微型计算机的硬件体系结构,熟知计算机硬件的工作原理和操作过程,并培养学生运用所学的知识进行软、硬件开发的初步能力。

本教材是为非计算机专业的学生编写的。考虑到非计算机专业的学生计算机基础知识薄弱,在教材的编写过程中我们力求做到:

(1)处理好教材内容的逻辑顺序、知识的内在顺序以及学生认识能力的发展顺序三者之间的关系,精心设计教材体系、安排教材内容,使内容的进程更符合学生的认知过程,易于理解。

(2)在注重基础知识的同时,保证教材内容的先进性。本教材从 8086 微处理器入手,使学生掌握微机的一般原理并建立微机系统的概念。在此基础上讲解 32 位、64 位等高性能微机的技术发展,介绍微机技术的新成果及发展趋势。

(3)对内容做合理取舍和科学浓缩。对于飞速更新换代的微处理器,本书浓缩各种新技术的主要内容,并淡化了某些内部原理及细节,以利于非计算机专业的学生对本课程的主要内容及重点知识的掌握。

此外,本书将存储器和 32 位、64 位微处理器两章放在第 8 章和第 9 章,一方面是为了使学生尽早进入实验教学环节并开始 I/O 接口电路的硬件实验;另一方面,经过一段时间的学习,使学生对这两章中一些有深度的内容更易于接受。

本书第 1、2、3、4、10 章以及第 7、8、9 章中的部分内容由王宁宁编写,第 5、6、7 章由张瑾编写,第 8 章由陈铖编写,第 9、11 章由谭彦彬编写。全书由王宁宁统稿。

限于编者的水平,编写时间仓促,错误和不足之处在所难免,敬请读者批评指正。

编 者
2009 年 3 月

目 录

第1章 基础知识	1
1.1 概述	1
1.2 计算机中的数与字符	3
1.2.1 二进制数与十六进制数的表示	3
1.2.2 位、字节、字、双字和四字	4
1.2.3 KB、MB、GB 和 TB	4
1.2.4 带符号数的表示法	5
1.2.5 数的定点与浮点表示	9
1.2.6 字符的编码	11
1.3 计算机基本工作原理	13
1.3.1 存储程序工作原理	13
1.3.2 计算机系统的组成	13
1.3.3 冯·诺依曼结构	15
习题	15
第2章 微型计算机基础	16
2.1 微型计算机系统的组成	16
2.1.1 微处理器、微机和微机系统	16
2.1.2 微型计算机的基本结构	16
2.1.3 微处理器	18
2.1.4 存储器	22
2.1.5 输入/输出接口电路	24
2.2 微机的工作过程	25
2.3 8086 微处理器	30
2.3.1 8086 微处理器的结构	30
2.3.2 8086 的引脚信号	40
2.3.3 8086 系统的最小模式和最大模式	45
2.3.4 8086 的存储器组织和 I/O 组织	47

2.3.5 8086 的总线操作.....	51
2.4 微型计算机系统的性能指标	54
习题	55
第3章 指令系统	57
3.1 概述	57
3.2 8086 的操作数寻址方式	59
3.2.1 立即寻址	59
3.2.2 寄存器寻址	60
3.2.3 存储器寻址	60
3.2.4 隐含寻址	66
3.2.5 I/O 端口寻址	66
3.3 8086 的机器指令格式	67
3.4 8086 指令系统	69
3.4.1 数据传送指令	70
3.4.2 算术运算指令	78
3.4.3 逻辑操作指令	91
3.4.4 程序控制指令	94
3.4.5 串操作指令	109
3.4.6 处理器控制指令	115
3.5 80X86 的扩展指令简介	117
3.5.1 指令集的 32 位扩展	117
3.5.2 扩展原有功能	118
3.5.3 80X86 新增指令	119
习题	122
第4章 汇编语言程序设计.....	127
4.1 程序设计语言概述	127
4.2 汇编语言的基本语法	128
4.2.1 汇编语言源程序的结构	128
4.2.2 汇编语言语句的格式	130
4.2.3 数据项及表达式	131
4.3 伪指令	135
4.3.1 数据定义伪指令	135
4.3.2 符号定义伪指令	138

4.3.3	段定义伪指令	139
4.3.4	过程定义伪指令	142
4.3.5	结束伪指令	143
4.3.6	宏指令	146
4.4	DOS 功能调用	147
4.5	汇编语言程序设计的基本方法	153
4.5.1	顺序程序	154
4.5.2	分支程序	156
4.5.3	循环程序	159
4.5.4	子程序设计	163
4.5.5	汇编语言程序举例	168
习题		174
第5章	输入与输出	178
5.1	输入/输出接口	178
5.1.1	外设接口电路的功能	178
5.1.2	CPU 与 I/O 之间的信号	179
5.1.3	接口电路的 I/O 端口及编址方式	179
5.2	数据传送的控制方式	181
5.2.1	程序控制传送方式	181
5.2.2	中断传送方式	184
5.2.3	DMA 方式	184
5.2.4	I/O 处理机方式	186
习题		187
第6章	中断技术	188
6.1	中断概述	188
6.1.1	中断基本概念	188
6.1.2	中断及响应过程	188
6.1.3	中断源的识别与判优	190
6.2	8086 的中断系统	192
6.2.1	8086 中断分类	192
6.2.2	中断向量表	194
6.2.3	8086 CPU 的中断响应过程	195
6.2.4	8086 系统的中断程序设计要注意的问题	196

6.3	可编程中断控制器 8259A	198
6.3.1	8259A 内部组成及工作原理	198
6.3.2	8259A 的工作方式	200
6.3.3	8259A 的编程	204
	习题	210
第 7 章	可编程接口芯片及应用	212
7.1	可编程并行通信接口芯片 8255A	212
7.1.1	并行通信与并行接口	212
7.1.2	可编程并行通信接口芯片 8255A	213
7.2	可编程串行通信接口芯片 8251A	222
7.2.1	串行通信与串行接口	222
7.2.2	可编程串行通信接口芯片 8251A	227
7.3	可编程计数/定时控制器 8253	233
7.3.1	8253 的结构和工作原理	234
7.3.2	8253 的控制字	236
7.3.3	8253 的工作方式	236
7.3.4	8253 的编程及应用举例	241
7.4	可编程 DMA 控制器 8237A	245
7.4.1	8237A 的结构	246
7.4.2	8237A 的编程	252
7.5	模数、数模转换接口	255
7.5.1	概述	255
7.5.2	典型的 A/D 转换器芯片及其与 CPU 的连接	257
7.5.3	典型的 D/A 转换器芯片及其与系统的连接	260
	习题	264
第 8 章	半导体存储器及其接口	267
8.1	概述	267
8.1.1	存储系统的层次结构	267
8.1.2	局部性原理	269
8.2	半导体存储器	269
8.2.1	半导体存储器的基本原理	269
8.2.2	典型存储芯片介绍	274
8.2.3	半导体存储器与 CPU 的连接	277

8.3	高速缓冲存储器	287
8.3.1	Cache 的组织方式	287
8.3.2	高速缓存的读写与替换	289
8.4	存储器管理	290
8.4.1	虚拟存储器概念	290
8.4.2	分段机制	291
8.4.3	分页机制	293
8.4.4	保护方式	294
	习题	295
第9章	32 位/64 位微处理器及新技术	297
9.1	微处理器发展概况及采用的主要技术	297
9.1.1	8086 之后微处理器的发展概况	297
9.1.2	Cache 技术	299
9.1.3	流水线和超标量流水线技术	300
9.1.4	动态执行技术	300
9.1.5	MMX 技术	302
9.1.6	SSE 技术	303
9.1.7	精简指令集计算机技术	304
9.1.8	精确并行指令计算技术	305
9.1.9	超线程技术和对称多处理技术	305
9.2	典型的微处理器简介	306
9.2.1	微处理器的结构	306
9.2.2	Pentium 微处理器	308
9.2.3	Pentium 4 微处理器	321
9.2.4	双核与多核微处理器	322
9.2.5	64 位微处理器	325
	习题	326
第10章	总线技术	327
10.1	总线的基本概念	327
10.1.1	单总线结构和分层总线结构	327
10.1.2	微型计算机总线的组成	329
10.1.3	微型计算机总线的分类	330
10.1.4	标准总线规范和标准总线性能指标	331

10.1.5 常见总线接口标准分类	332
10.2 常用微型计算机总线	332
10.2.1 PC/XT 总线	333
10.2.2 ISA 工业标准总线	335
10.2.3 PCI 总线	338
10.2.4 USB 总线	346
习题	352
第 11 章 常用外围设备	353
11.1 概述	353
11.1.1 外围设备的分类	353
11.1.2 外围设备的作用	354
11.2 输入设备	354
11.2.1 键盘	354
11.2.2 鼠标	357
11.2.3 扫描仪	359
11.2.4 数码相机	361
11.3 输出设备	362
11.3.1 显示设备	362
11.3.2 打印设备	364
11.4 外部存储器	366
11.4.1 硬盘存储器	366
11.4.2 移动硬盘	368
11.4.3 U 盘	368
11.4.4 光盘存储器	368
附录	370
附录 A 8086/8088 指令表	370
附录 B 8086/8088 中断向量地址表	378
附录 C 常用的 DOS 软中断及 DOS 系统功能调用简表	380
附录 D BIOS 功能调用	385
附录 E DEBUG 命令表	388
参考文献	390

第1章 基础知识

1.1 概 述

1946年,以 ENIAC(Electronic Numerical Integrator and Calculator)命名的世界上第一台电子计算机诞生于美国宾夕法尼亚大学。这台本是为着军事目的而研制的计算机使用了18800只电子管,耗电150kW,重30t,占地167m²,造价50万美元。每秒可进行5000次加法运算或者400次乘法运算。ENIAC将人类带入了一个崭新的计算机时代,并因此被誉为信息时代的启明星。它的历史意义远远超过了其实用价值。

现在的程序员们恐怕很难想象,ENIAC的编程是通过重新连接它的电路进行的。像早期的电话接线总机那样,ENIAC带有布满插座的插座板,以便通过调整插座来改变插座板上的电路连接,完成对ENIAC的编程。这显然是一件极其耗时的事情。ENIAC的另一个问题是电子管的寿命短,最长只有3000h,且工作电压高,电能消耗大,散发出很多热量,机器的运行常因某个管子被烧坏而死机,故障又难以迅速排除,这是电子管计算机亟待解决的问题。

1947年年末,美国电话电报公司的贝尔实验室成功研制出了晶体管。晶体管被看作20世纪最伟大的发明之一。与电子管相比,它的寿命延长了1000倍,能耗却只是电子管的几十分之一,并且体积大大缩小,耐振动、抗冲击、可靠性提高了100倍。晶体管的发明荣获了1956年的诺贝尔物理奖。1958年,体积大为缩小、能耗明显降低、可靠性增强、速度提高的晶体管计算机取代了电子管计算机。人们根据器件的发展对电子计算机划代,于是电子管计算机被称为第一代计算机,晶体管计算机则为第二代计算机。

晶体管的出现拉开了集成电路(Integrated Circuit, IC)出台的序幕。1958年,得克萨斯仪器公司成功地把20多个包括晶体管、电阻和电容的元器件以及它们之间的连线做在了一个4mm²的硅片上,开发出了第一块集成电路,实现了电路的微型化,为人类的微电子革命吹响了进军号。这项发明于2000年获得了诺贝尔物理学奖。20世纪60年代中期,芯片的集成度(指一定尺寸的芯片上能做出多少个晶体管)已经提高到几百甚至上千只晶体管。器件的发展推动着计算机的更新换代,1965年,第三代计算机——集成电路计算机登台亮相。第三代计算机的体积更小、速度更快、可靠性更高。

20世纪70年代起,集成电路进入飞速发展时期。从20世纪70年代到80年代,大规模、超大规模集成电路时代相继而至。电子计算机发展到以大规模、超大规模集成电路为主体的第四代。大规模集成电路的出现为计算机的微型化创造了条件,微型计算机应运而生,这是微电子技术引发计算机技术产生巨大变革的结果。

微型计算机一经问世,便以前所未有的势头迅猛发展并不断地推陈出新。

微型计算机的发展是以微处理器的发展为特征的。微处理器是微型计算机的核心部

件,它是一个主要由运算器的和控制器组成的大规模集成电路芯片。世界上第一块微处理器于1971年问世,这就是Intel公司的Marcian Hoff研制成功的Intel 4004。令人始料不及的是,这块当初只是为高级袖珍计算机而设计的、主要用于算术运算的微处理器竟然引发了一场计算机革命,这场革命使计算机从科学殿堂走向大众。

1980年,一直称雄大型计算机领域的IBM公司将视线投向微型计算机。IBM以“制造一种人们真正想拥有的机器,真正属于个人的电脑”为宗旨,于1981年8月12日推出了命名为“Personal Computer”的个人计算机,该机以Intel 8088作为IBM PC的CPU。此后,它的缩写“PC”便成为个人计算机的代名词。这种面向个人的计算机果然很受欢迎,于是IBM投入更多的人力和财力增强它的功能,促进它的普及。同时IBM还采取了一项重大举措:公开PC机除BIOS之外的全部技术资料,使别的软件公司能够开发基于IBM PC的应用程序。这一开放的做法深得人心,它一方面使得IBM的个人计算机最大范围地得到软件人员开发的各种应用软件的支持,迅速扩大了其应用范围;另一方面又使IBM的个人计算机成为一个标准,使其他微型计算机都不得不向它看齐。IBM PC的推出不仅极大地推动了微型计算机的发展,而且标志着微型计算机时代的真正到来。

由于IBM公司选择了Intel公司的芯片作为其个人计算机的微处理器,伴随着个人计算机时代的到来,在生产微处理器方面,Intel已经成为一面旗帜。可以说,Intel的发展基本上反映了微处理器的发展历程。

从20世纪70年代至今,微型计算机已经推出了7代产品。与计算机的划代不同,微型计算机的划代主要依据微处理器的功能、字长和集成度。表1-1列出了各代微处理器的简单情况。有关微处理器的更多内容见第9章。

表1-1 微处理器发展简表

特点 分代	典型微处理器芯片	字长 /位	集成度/ (晶体管/片)	时钟频率 /MHz	运行速度/ MIPS
第一代 1971年—1972年	Intel 4004/8008	4/8	1千~2千	<1	0.05
第二代 1973年—1977年	Intel 8080, Z80, M6800	8	5千~1万	2~4	0.1~1
第三代 1978年—1983年	Intel 8086/80286, M68000, Z8000	16	2万~13万	5~16	1~2.7
第四代 1984年—1992年	Intel 80386/80486, M68030/68040	32	27万~100多万	16~66	6~40
第五代 1993年—1994年	Pentium	32	310万	60~200	100~200
第六代 1995年—2001年	Pentium Pro, Pentium II/III/4, K6, K7, Power PC	64	550万~5000万	133~2400	>300
第七代 2002年以后	Itanium, K8, Power 5	64		800~	>3000

注:MIPS(Millions of Instructions Per Second)是指每秒执行的指令数,单位为百万条

微型计算机除了像电子计算机那样,能够自动、高速、精确地进行信息处理之外,还有

着它自己特殊的优点:

(1)体积小,功耗低。

由于采用了超大规模集成电路技术,数千万只晶体管集成在一个几十平方毫米的硅片上,从而使过去耗电几十甚至上百千瓦的庞然大物如今只需若干芯片就能装配起来,功耗仅为几瓦。这一优点对于航天、军事和自动控制等领域显然具有特别的意义。

(2)可靠性高,使用环境要求低。

超大规模集成电路技术大大减少了元器件的数目及连线,可靠性明显提高。加之微型计算机体积小,便于对特殊部件采用密封保护措施,因此一般环境下都可以使用。而其他类型的计算机往往对环境要求苛刻,有些甚至需要对空气净化才能满足要求。

(3)系统设计灵活。

由于微计算机中的各功能部件已成为标准化、系统化产品,用户可依需要和经济能力方便地构成规模不同的系统,极大地方便了想使用微计算机的用户。

(4)价格低廉。

微型计算机正是由于这一突出的优点才走入千家万户,获得最广泛的应用。1964年,当身家50亿美元的计算机IBM360系列开发出来时,不要说个人或家庭,就是一般的院校或机构,谁敢想象拥有这样一台天价的计算机呢?可是今天,价格越来越便宜的个人计算机快步走向大众,进入家庭,使计算机再也不是只有少数专业人员才能接触的科学界贵族了。计算机性能价格比的提高速度是迄今为止任何其他领域的产品难以比拟的。

现在,从航天工业、军工生产、自动化控制、石油勘探、数字通信到家用电器、时尚娱乐、儿童玩具,微型计算机已遍及国民经济的各个部门,进入人们生活的方方面面,并且被越来越多的家庭和个人所拥有,这种更大范围的使用和研究又反作用于微计算机的开发和发展。微型计算机在改变着世界诸多事物进程的同时,更使我们切实感受着它对我们生活的深刻影响。

微型计算机的价值又远非它自身的发展所能衡量。因为它可以改变其他经济部门的生产过程,提高生产率,开拓新领域,增加新品种。因此,它是推动经济和科技高速发展的有力工具。推广使用微型计算机的程度已成为衡量科学技术和工业生产现代化的重要标志之一。

1.2 计算机中的数与字符

1.2.1 二进制数与十六进制数的表示

数在机器中是以器件的物理状态表示的。计算机中用电平的“高”、“低”或电流的“有”、“无”分别表示“1”、“0”,所以计算机中采用的是二进制数字系统。二进制数是计算机唯一能够识别的。因此,任何信息——数据、符号、文字、指令(程序)在计算机中都是以二进制表示的。

尽管计算机内部采用二进制数,但二进制数的书写和阅读却显得冗长不便,于是人们通常使用与二进制有着简单转换关系的十六进制表示计算机中的数。为了区别不同的数制,在每个数的后面用一个字母加以注明。用字母标注不同进制数的方法是:

二进制数后面跟字母B(Binary),如1101B。

十进制数后面跟字母 D (Decimal) 或不加字母,如 64D 或 64。
十六进制数后面跟字母 H (Hexdecimal) ,如 BOA3H。

1.2.2 位、字节、字、双字和四字

计算机中最小的数据单位是位,以 bit(Binary Digit)表示,它是指一位的二进制数。
一个 8 位的二进制数称为一个字节,用字母 B(Byte)表示。字节是计算机中表示数据的独立单位。在微型计算机中,数据的存放都以字节表示,数据的处理也常以字节为单位进行。
一个 16 位的二进制数称为一个字,用字母 W(Word)表示。它由两个连续的字节组成,因而一个字又可分为高字节和低字节两部分。
一个 32 位的二进制数称为一个双字,用字母 DW(Double - Word)表示。它由两个连续的字(四个连续的字节)构成,因而可分为高字和低字两部分。
一个 64 位的二进制数称为一个四字,用字母 QW(Quad - Word)表示。它由两个连续的双字构成。

考虑到数的表示范围与所需的二进制数的位数之间的关系,计算机中编号的顺序通常从“0”开始。因此,一个字节的 8 个二进制位从低到高的编号是从 bit0 到 bit7。字、双字和四字同理,如图 1-1 所示。

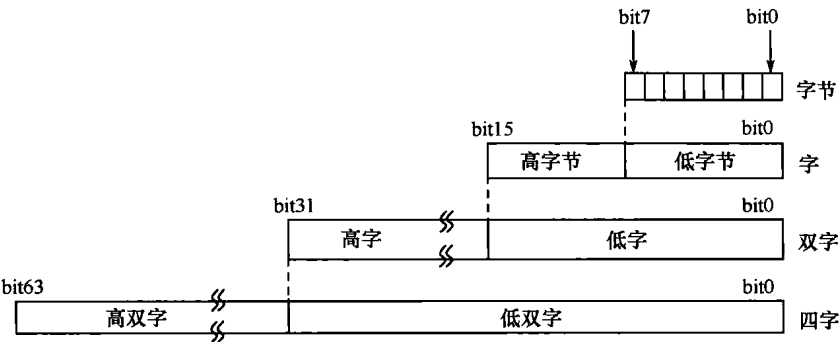


图 1-1 字节、字、双字和四字

1.2.3 KB、MB、GB 和 TB

计算机可以存放并处理大量的信息。为了方便地描述很大的量,常使用下列术语简称:KB(千字节)、MB(兆字节)、GB(吉字节)和 TB(太字节)。这些术语所代表的数值如表 1-2 所列。

表 1-2 术语代表的数值

术 语	表示的数值	简 称
1KB(kilobyte)	2^{10} 字节 = 1024 字节	1 千字节
1MB(megabyte)	2^{20} 字节 = 1048576 字节	1 兆字节
1GB(gigabyte)	2^{30} 字节 = 1073741842 字节	1 千兆字节
1TB(terabyte)	2^{40} 字节 = 1099511627776 字节	1 兆兆字节
注意:1KB≠1000 字节,1MB≠1000000 字节。习惯上把 2^{16} 称为 64K,是因为 $2^{16}=2^6 \times 2^{10}=64 \times 1024$,所以 64K 的确切值为 $64 \times 1024=65536$		

1.2.4 带符号数的表示法

数的符号在计算机中也是以二进制数表示的,称为符号位。通常将符号位放在一个数的最高位,并用“0”表示正,“1”表示负。符号位也数值化了的数叫做机器数。一个数被表示为机器数之后,它原来的值被称为机器数的真值。

一个带符号数在机器中可以用三种方法表示,即原码、反码和补码。下面以8位机器数为例来说明这三种码制的表示方法。

1. 原码

正数的符号位为0,负数的符号位为1,数值部分与真值相同,这种表示方法称为原码表示法。

如: $[+100]_{\text{原}} = 01100100$
 $[-100]_{\text{原}} = 11100100$

↑ 数值位
符号位

“0”在原码中有两种表示: $[+0]_{\text{原}} = 00000000$
 $[-0]_{\text{原}} = 10000000$

2. 反码

正数的反码表示与原码相同。负数的反码是它相应正数的按位取反。如:

$[+100]_{\text{反}} = 01100100$
 $[-100]_{\text{反}} = 10011011$
 $[+63]_{\text{反}} = 00111111$
 $[-63]_{\text{反}} = 11000000$

“0”在反码中也有两种表示: $[+0]_{\text{反}} = 00000000$
 $[-0]_{\text{反}} = 11111111$

3. 补码

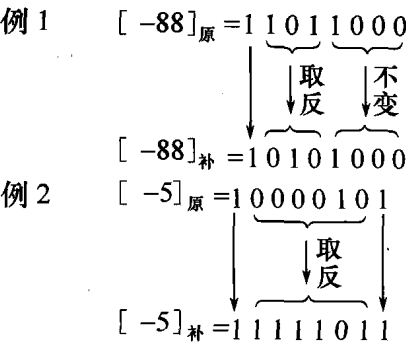
正数的补码表示与原码相同。负数的补码由该数的反码加1形成。试比较:

$[+100]_{\text{原}} = 01100100$
 $[-100]_{\text{反}} = 10011011$
 $[-100]_{\text{补}} = 10011100$
 $[+63]_{\text{原}} = 00111111$
 $[-63]_{\text{反}} = 11000000$
 $[-63]_{\text{补}} = 11000001$
 $[+0]_{\text{原}} = 00000000$
 $[-0]_{\text{反}} = 11111111$
 $[-0]_{\text{补}} = 00000000$

可见, $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000$, 所以“0”在补码中只有一种表示。

计算机只能通过将原码“变反加1”的方法得到一个负数的补码。而人在求负数的补码时则可以用更简单的办法:将原码“两头不变,中间取反”。具体地说就是:先写出一个

负数的原码,让这个原码左端的符号位“1”不变;右边从最低位起,向左找到第一个“1”,这部分的数值不变;这两个“1”之间的全部数值按位取反,即得到这个负数的补码。



由于“1”变反后再加 1 仍是“1”。“10”变反为“01”,再加 1 又回到“10”。无论“1”之后跟几个“0”,变反之后再加 1 将还原为原来的数。因此,“两头不变,中间取反”的道理显而易见。

表 1-3 是 8 位机器数的对照表。

表 1-3 8 位机器数对照表

真 值	原 码	反 码	补 码
+127	0111 1111	0111 1111	0111 1111
+126	0111 1110	0111 1110	0111 1110
+125	0111 1101	0111 1101	0111 1101
+124	0111 1100	0111 1100	0111 1100
⋮	⋮	⋮	⋮
+2	0000 0010	0000 0010	0000 0010
+1	0000 0001	0000 0001	0000 0001
+0	0000 0000	0000 0000	0000 0000
-0	1000 0000	1111 1111	0000 0000
-1	1000 0001	1111 1110	1111 1111
-2	1000 0010	1111 1101	1111 1110
-3	1000 0011	1111 1100	1111 1101
⋮	⋮	⋮	⋮
-124	1111 1100	1000 0011	1000 0100
-125	1111 1101	1000 0010	1000 0011
-126	1111 1110	1000 0001	1000 0010
-127	1111 1111	1000 0000	1000 0001
-128			1000 0000

几点说明:

① 对于 8 位的二进制数,原码和反码可以表示的数值范围是 $-127 \sim +127$,即 $-(2^7-1) \sim +(2^7-1)$;补码表示的数值范围是 $-128 \sim +127$,即 $-2^7 \sim +(2^7-1)$ 。对于 n 位的二进制数,去掉一位符号位,数值部分为 $n-1$ 位,则原码和反码能表示的数值

范围是 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$;补码表示的数值范围是 $-2^{n-1} \sim +(2^{n-1}-1)$ 。

② $[[X]_{\text{补}}]_{\text{补}} = [X]_{\text{原}}$ 。所以对补码再求补又回到原码。

若已知有一补码表示的带符号数为 11111100B(FCH),按照“两头不变,中间取反”的原则可得这个数的原码为 10000100B,于是可知该数的真值为 -4。

③ 比较表 1-3 可见,负数的绝对值越大,其对应的反码和补码中高位的“0”越多,“1”越少;负数的绝对值越小,其对应的反码和补码中高位的“0”越少,“1”越多。这与我们习惯的看法相悖。由于反码和补码都是在对原码“变反”的基础上形成的,这一现象并不奇怪。了解了这一点,今后在遇到补码时,应当对其真值有一个正确的估计。

引进补码一方面使得符号位能和数值部分一起参与运算从而简化了运算规则;另一方面还使得减法运算可以转化为加法运算,进一步简化计算机中的运算电路和处理过程。下面来说明如何利用补码把减法转换为加法。

先以一个钟表对时为例。

若现在的标准时间是 6 点整,但一只钟却指在 10 点,要将它校准到 6 点整有两种拨法:逆时针拨 4h 或顺时针拨 8h,两种拨法分别记为

逆时针拨: $10 - 4 = 6$

顺时针拨: $10 + 8 = 12 + 6 = 6 \quad (\text{mod } 12)$

Mod 12 表示以 12 为模的运算。模(modulo)是一个计数系统的最大限量,到了这个限量,系统将回到“0”重新计数,模自然丢失。所以对于模 12 的计数系统,10 加 8 与 10 减 4 等价,或者说 $(10 - 4)$ 可以转换为 $(10 + 8)$,即

$10 - 4 = 10 + 8 \quad (\text{mod } 12)$

这个式子的意思是:等式两边同除以 12,它们的余数相同。或者说, $(10 - 4)$ 与 $(10 + 8)$ 对模 12 同余,因此上式又称为同余式。同时我们称 (-4) 与 $(+8)$ 对模 12 互为补数,并表示为

$-4 = 8 \quad (\text{mod } 12)$

同理, (-5) 与 $(+7)$, (-6) 与 $(+6)$, (-9) 与 $(+3)$ 等等,对模 12 也都互为补数,即对模 12 同余。不难看出,一个负数的补数等于模加上该负数。如

$[-4]_{\text{补}} = 12 + (-4) \quad (\text{mod } 12)$

模运算中引进了补数的概念后,减法转化为加法的情况就可以推广到一般模运算中:在某一个模确定的数字系统中,数 A 减去绝对值小于模的另一数 B,总可以用 A 加上“B 的负数与模之和”(即补数)来代替。上面校准时间的例子用这句话表述时可记为

$10 - 4 = 10 + (-4 + 12) \quad (\text{mod } 12)$

用二进制表示的 2^n ,是一个有 n 个“0”的 $n+1$ 位数。计算机的字长为 n 位时, 2^n 就只能以 n 个“0”表示,最左边的数字“1”丢失,所以 n 位二进制数的模为 2^n ,也称模是 2。

为了与补数相区别,我们把符号位也数值化了的补数称为补码。

一般来说,对于 n 位的二进制数,设最高位为符号位,则整数 X 的补码定义为

$[X]_{\text{补}} = 2^n + X \quad -2^{n-1} \leq X < 2^{n-1} \quad (\text{mod } 2)$

由定义知,当 X 为正数时, $[X]_{\text{补}} = X$;当 X 为负数时,按照定义求 X 的补码就要做减法。但从前面我们已经知道,计算机对一个二进制的负数求补码时,是通过把该数对应的正数连同符号位按位取反再加 1 得到的,即借助原码和反码求得补码。这就避免了求补