

普通高等教育计算机基础课程规划教材

微机原理与接口技术

WEIJI YUANLI YU JIEKOU JISHU

唐翔 主编 刘红玲 徐亚峰 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等教育计算机基础课程规划教材

微机原理与接口技术

唐 翔 主编

刘红玲 徐亚峰 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是根据教育部高等学校计算机基础课程教学指导委员会编制的《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》中有关理工类专业“微机原理与接口技术”课程教学要求组织编写的。全书共分为7章：微型计算机基础及工作原理、微处理器、指令系统和汇编语言程序设计、存储系统、输入/输出技术、微机接口及应用、微机总线及I/O接口标准。

本书内容全面系统、概念清楚、例题丰富、通俗易懂、实用性强。本书适合作为高等学校理工类各专业微机原理与接口技术的教学用书，也可作为微机应用系统设计和开发人员的参考书或培训教材。

图书在版编目（CIP）数据

微机原理与接口技术 / 唐翔主编. —北京：中国铁道出版社，2011.7

（普通高等教育计算机基础课程规划教材）

ISBN 978-7-113-12518-9

I. ①微… II. ①唐… 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（2011）第012698号

书 名：微机原理与接口技术

作 者：唐 翔 主编

策划编辑：吴宏伟 苏 博

责任编辑：杜 鹏 侯 颖

特邀编辑：王 惠

封面设计：付 巍

责任印制：李 佳

读者热线：400-668-0820

封面制作：白 雪

出版发行：中国铁道出版社（北京市宣武区右安门西街8号 邮政编码：100054）

印 刷：三河市华业印装厂

版 次：2011年7月第1版 2011年7月第1次印刷

开 本：787mm×1092mm 1/16 印张：18.25 字数：431千

印 数：3 000册

书 号：ISBN 978-7-113-12518-9

定 价：28.00元

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社计算机图书批销部联系调换。

普通高等教育计算机基础课程规划教材

主任委员：冯博琴

副主任委员：管会生 李凤霞

委 员：（按姓氏笔画排序）

刘红梅 曲建民 何东健

张长海 李俊山 周 苏

唐 翔 高 飞 曹岳辉

计算机基础教学在我国高等教育中已有 30 多年的发展历史,已经成为我国高等教育的重要组成部分,是培养大学生综合素质的重要环节。计算机不仅为解决专业领域问题提供有效的方法和手段,而且提供了一种独特的处理问题的思维方式;计算机及互联网有着极其丰富的信息和知识资源,为学生学习提供了广阔的空间以及良好的学习工具;善于使用互联网和办公软件是良好的交流表达能力和团队合作能力的重要基础;同时,计算机基础教学也为学生创新能力的培养奠定了基础。不难发现,现在几乎所有领域的重大成就无不得益于计算科学的支持,计算科学已经和理论科学、实验科学并列成为推进社会文明进步和科技发展的三大手段。事实上,当今任何一项被称为“高科技”的项目或专业、职业,无一不是与计算机紧密结合的。计算机基础教学应致力于使大学生掌握计算科学的基本理论和方法,为培养复合型创新人才服务。

本届教指委以科学发展观为指导,为促进计算机基础教学不断向科学、规范、成熟的方向发展,于 2009 年 10 月发布了《高等学校计算机基础教学战略研究报告暨计算机基础课程教学基本要求》(以下简称《基本要求》),它充实了“4 个领域×3 个层次”的计算机基础教学的知识结构,提出和构建了计算机基础教学的实验体系,科学地描述各专业大类核心课程的教学基本要求。《基本要求》提出了计算机基础教学应该达到的 4 项“能力结构”要求,即对计算机的认知能力、利用计算机解决问题的能力、基于网络的协同能力、信息社会中的终身学习能力。以此为源头,构建培养这 4 种能力的两大支柱,即计算机基础教学的“知识体系”和“实验体系”。这两大体系中蕴含着计算机基础教学所包含的所有内容,即 148 个知识单元、884 个知识点、119 个实验单元和 529 个技能点。根据教学目标,可以从中选取若干知识单元、知识点、实验单元和技能点,构建所需课程。这项研究基本上厘清了我国高校计算机基础教学的体系、内容和要求,向科学、规范和可操作的方向迈出了一大步。

中国铁道出版社热心于计算机教育,在计算机基础教学方面办了许多实事,在高校师生中赢得了良好口碑。在《基本要求》发布之后,我们组织国内一批知名教授和有实力的作者,按照《基本要求》编写了本丛书,以推动《基本要求》的贯彻,提高高校计算机基础教学质量。

本丛书定位于应用型本科,内容充分体现应用性,兼顾基础性;强调学生的动手能力培养,避免过多的理论内容;教材尽量采用案例驱动。丛书按照计算机基础教学六门核心课程组织,有的课程或因平台不同,或因教材编写风格、定位等不同,会有一门课程多本教材的情况,这是为了给老师提供更多的选择,以使其找到更合适自己的优秀教材。

我们希望本丛书的出版,能对推动我国高校的计算机基础教学改革尽到一份力量。书中难免存在不足之处,恳望读者不吝指正。谢谢大家。

冯博琴

2010.10.8

冯博琴,西安交通大学教授,博士生导师,现任教育部 2006—2010 年高校计算机基础课程教学指导委员会副主任委员,全国计算机基础教育研究会副会长,陕西省计算机教育研究会理事长。

“微机原理与接口技术”是大学计算机基础教学中的核心课程，主要介绍微型计算机硬件系统组成、工作原理及接口电路设计方法等方面的知识，培养学生微机应用系统的初步开发能力，为今后从事计算机控制和计算机信息处理等相关领域的研究打下基础。本教材是根据教育部高等学校计算机基础课程教学指导委员会编制的《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》中有关理工类专业“微机原理与接口技术”课程的教学要求编写的。依据要求中的知识单元顺序，以 16 位微型计算机为基础，同时追踪 32 位与 64 位主流系列高性能微机的技术发展方向，并以常用的 PC 系列微型计算机为主线，详细介绍微型计算机技术的基础内容，适当分析、介绍微型计算机技术的新发展。内容覆盖了要求中的各个知识点，实例丰富、应用性强。

全书共分为 7 章，主要内容如下：

第 1 章主要介绍了微型计算机系统的基础知识，包括计算机中数据的表示、组成微型计算机的基本逻辑电路、指令与程序的执行过程及微型计算机的基本工作原理；同时简要说明了微型计算机的发展过程及性能指标，让学习者先建立一个完整的微型计算机系统的概念。

第 2 章详细介绍了 8086/8088 微处理器的内部结构、引脚定义、工作模式以及总线时序，并从实际应用出发，讲述了 Pentium 微处理器的特点、程序设计模型和工作模式；简要说明了嵌入式系统和嵌入式微处理器的概念及应用。

第 3 章讲述了 80x86 的寻址方式、8086/8088 的指令系统以及汇编语言程序设计的过程和方法，包括 DOS 下的汇编语言程序框架、开发方法、伪指令、模块化程序设计方法和设计实例；同时简要介绍了汇编语言和 C/C++ 语言混合编程的基础知识。

第 4 章简要介绍了微型计算机的存储系统，包括存储器的概念、分类、存储器体系结构等；重点讲述了主存储器的组织、与 CPU 的接口技术，同时介绍了微型计算机系统的高速缓冲存储器和辅助存储器。

第 5 章介绍了微型计算机输入/输出系统的组成、特点、接口功能、设计方法，以及微处理器与外设间数据传送的控制方式；详细阐述了微型计算机的中断系统、实模式下的中断处理过程、中断程序设计方法，以及中断控制器 8259A 的功能、结构和编程应用等方面的知识。

第 6 章讲述了微型计算机系统的简单接口芯片、可编程接口芯片以及常用的人机交互设备及接口方法，包括并行接口芯片 8255A、串行接口芯片 16550、定时/计数芯片 8253/8254 以及模拟接口芯片，详细介绍了这些芯片的内部结构、与微处理器的接口以及应用编程。

第 7 章介绍了微型计算机中使用的系统总线标准以及常用的 I/O 接口标准，包括 ISA、PCI

和最新的 PCI-E 总线标准。接口标准则重点讲述了现代微型计算机必备的 USB 接口标准以及高性能串行总线标准 IEEE 1394。

本书由唐翔主编，刘红玲、徐亚峰为副主编。第 4、5、6 章由唐翔编写，第 3、7 章由刘红玲编写，第 1、2 章由徐亚峰编写，全书由唐翔统稿、定稿。在本书的编写过程中，中国铁道出版社的编辑给予了很大的支持，在此表示由衷的感谢！

由于作者水平有限及时间仓促，书中的疏漏和不当之处在所难免，敬请广大读者和同仁批评指正。

编 者

2011 年 4 月

第 1 章 微型计算机基础及工作原理	1
1.1 计算机中数据的表示	2
1.1.1 数值型数据的表示	2
1.1.2 定点数和浮点数	5
1.1.3 非数值数据的表示	7
1.2 二进制数的运算	10
1.2.1 二进制数的算术运算	10
1.2.2 二进制数的逻辑运算	12
1.3 微型计算机的基本组成电路	13
1.3.1 基本逻辑门	13
1.3.2 触发器	14
1.3.3 寄存器	16
1.3.4 译码器	17
1.3.5 三态输出电路	18
1.3.6 加法电路与算术逻辑运算单元	18
1.3.7 存储器	19
1.4 微型计算机的工作原理	20
1.4.1 计算机的发展	20
1.4.2 计算机的分类	21
1.4.3 冯·诺依曼计算机体系结构	23
1.4.4 计算机系统的构成	23
1.4.5 微型计算机的硬件基本结构	24
1.4.6 微型计算机系统的性能指标	25
思考与练习	26
第 2 章 微处理器	27
2.1 微处理器概述	28
2.1.1 微处理器的基本概念	28
2.1.2 微处理器的基本结构与功能	28
2.1.3 微处理器的发展过程	29
2.2 8086/8088 CPU 结构	30
2.2.1 执行部件 (EU)	31
2.2.2 总线接口部件 (BIU)	32
2.2.3 内部寄存器及其功能	33
2.2.4 存储器组织	35

2.3	8086/8088 CPU 的外部结构和工作模式	36
2.3.1	8086/8088 CPU 的主要引脚及其功能	36
2.3.2	8086/8088 CPU 在最小模式和最大模式下的典型配置	39
2.4	8086/8088 微处理器的基本时序	41
2.4.1	指令周期、总线周期及时钟周期	41
2.4.2	最小模式下的典型时序	41
2.4.3	最大模式下的典型时序	43
2.5	Pentium 系列微处理技术概述	44
2.5.1	Pentium 微处理器的内部结构	44
2.5.2	Pentium 微处理器的特点	45
2.5.3	Pentium 微处理器的程序设计模型	46
2.5.4	Pentium 微处理器的工作模式	49
2.6	嵌入式系统和嵌入式处理器概述	51
	思考与练习	53
第 3 章	指令系统和汇编语言程序设计	54
3.1	指令与指令系统的基本概念	55
3.1.1	指令和指令系统	55
3.1.2	指令的基本格式	55
3.2	寻址方式	56
3.2.1	指令的寻址方式	56
3.2.2	操作数的寻址方式	56
3.3	8086/8088 的指令系统	58
3.3.1	数据传送类指令	58
3.3.2	算术运算类指令	61
3.3.3	逻辑运算和移位循环类指令	64
3.3.4	串操作类指令	66
3.3.5	程序控制类指令	68
3.3.6	处理器控制类指令	72
3.4	汇编语言程序的开发过程	73
3.4.1	汇编语言源程序和汇编程序	73
3.4.2	汇编语言源程序的结构	74
3.5	汇编语言基本语法	75
3.5.1	汇编语言语句的种类及其格式	75
3.5.2	汇编语言的数据	76
3.5.3	汇编语言的伪指令语句	82
3.5.4	DOS 和 BIOS 中断调用	85
3.6	汇编语言程序设计基本方法	87
3.6.1	顺序程序设计	87

3.6.2	分支程序设计	89
3.6.3	循环程序设计	92
3.6.4	子程序设计	96
3.7	汇编语言和 C/C++ 语言混合编程	102
3.7.1	混合编程的基本规则	102
3.7.2	C/C++ 语言中内嵌的汇编指令	102
3.7.3	独立的汇编目标码	104
	思考与练习	106
第 4 章	存储系统	111
4.1	存储器概述	112
4.1.1	存储器的分类	112
4.1.2	存储系统体系结构	114
4.1.3	存储器的主要性能指标	116
4.2	半导体存储器	116
4.2.1	半导体存储器的特点	117
4.2.2	半导体存储器的基本结构	117
4.2.3	典型芯片	118
4.3	微机系统中的主存储器组成	121
4.3.1	存储器的扩展技术	121
4.3.2	译码电路的设计	123
4.3.3	存储器芯片与 CPU 的连接	126
4.3.4	微机的主存储器组成	131
4.3.5	DRAM 内存条简介	133
4.4	高速缓冲存储器	136
4.4.1	“cache-主存”存储体系	136
4.4.2	主存与 cache 的地址映射方式	138
4.4.3	cache 的替换策略	140
4.4.4	cache 的更新策略	140
4.5	辅助存储器	141
4.5.1	硬盘存储器	141
4.5.2	光盘存储器	144
4.5.3	新型辅助存储器	147
	思考与练习	150
第 5 章	输入/输出技术	151
5.1	输入/输出概述	152
5.1.1	I/O 系统的组成和特点	152
5.1.2	I/O 端口的编址方式	154
5.1.3	I/O 地址译码	155

5.1.4 接口的分类及特点	157
5.1.5 接口设计方法	158
5.2 基本输入/输出方法	160
5.2.1 程序控制方式	160
5.2.2 中断控制方式	161
5.2.3 DMA 方式	161
5.3 中断技术	162
5.3.1 中断及中断响应的一般过程	162
5.3.2 中断控制器 8259A	164
5.3.3 微机的中断系统	176
5.3.4 中断控制程序设计	179
思考与练习	181
第 6 章 微机接口及应用	183
6.1 数字接口电路	184
6.1.1 简单接口芯片的应用	184
6.1.2 可编程并行 I/O 接口芯片	184
6.1.3 可编程串行 I/O 接口芯片	194
6.1.4 可编程定时器/计数器芯片	207
6.2 模拟接口电路	220
6.2.1 D/A 转换器及其应用	220
6.2.2 A/D 转换器及其应用	228
6.2.3 工业闭环控制系统结构框架	232
6.3 人机交互设备及接口	233
6.3.1 键盘及其接口	233
6.3.2 鼠标及其接口	237
6.3.3 显示器及其接口	239
6.3.4 打印机及其接口	244
思考与练习	248
第 7 章 微机总线及 I/O 接口标准	250
7.1 总线概述	251
7.1.1 总线的基本概念	251
7.1.2 总线的分类	252
7.1.3 总线的基本结构	252
7.1.4 总线的主要技术指标	256
7.2 系统总线标准	257
7.2.1 ISA 总线	257
7.2.2 EISA 总线	259
7.2.3 PCI 局部总线	260

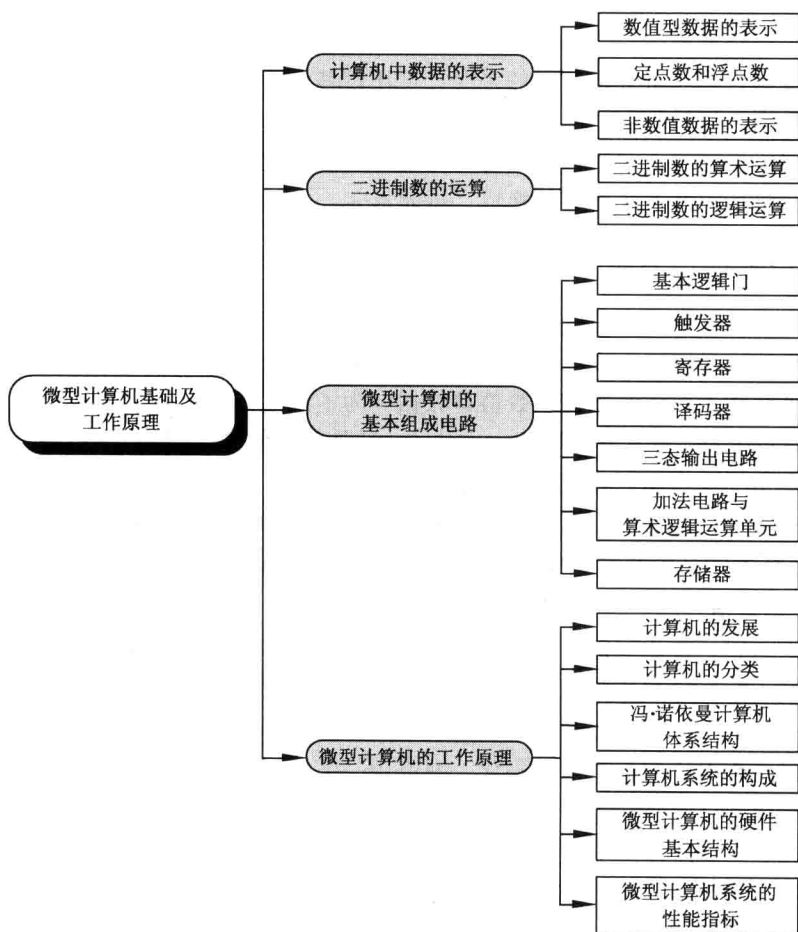
7.2.4 PCI-E 局部总线.....	261
7.3 外设总线.....	262
7.3.1 外设总线（I/O 接口）概述.....	262
7.3.2 常用外设总线（I/O 接口）标准.....	263
思考与练习.....	271
附录 A DEBUG 主要命令.....	272
参考文献.....	276

第1章 微型计算机基础及工作原理

引言

微型计算机是以微处理器为核心的计算机的统称，从第一代微处理器 4004 问世以来，微型计算机已有了突飞猛进的发展，不管是性能还是应用领域，其发展速度都远远超过了大型机、中型机和小型机。本章简单介绍微型计算机的基础知识、基本组成电路及微型计算机的工作原理。

本章结构图



学习目标

通过对本章内容的学习，应该能够：

- 了解：计算机的发展、分类以及微型计算机系统的性能指标。
- 理解：计算机体系结构、计算机系统构成以及微型计算机的硬件基本结构。
- 应用：掌握微型计算机系统中数值数据、非数值数据的表示；掌握定点数和浮点数的使用；掌握微型计算机的基本组成电路，并能够在实践中灵活运用。
- 分析：通过本章学习，学会分析微型计算机系统的硬件组成电路，能较深入地理解各组成部分的工作原理。

1.1 计算机中数据的表示

通过计算机来解决实际问题，需要把一系列信息输入到计算机中，并且让计算机能够识别和处理，这就要求提取描述客观实际的特征信息，并且用计算机能识别的形式把它表示出来，也就是我们所说的数据。计算机通过对数据的加工和处理，最终解决实际问题。

国际标准化组织对数据的定义为：数据是对事实、概念或指令的一种特殊表达形式，这种特殊的表达形式可以用人工的方式或自动化的装置进行通信、翻译转换或者进行加工处理。根据数据的性质，在计算机中可以把数据分为数值型数据和非数值型数据。数值型数据是指具有特定值的一类数据，可用来表示数量的多少，可比较其大小。对于通用计算机来说，非数值型数据包括字符数据、逻辑数据、图像、声音和视频数据等，这些数据没有大小的概念。计算机只能识别和处理以二进制形式表达的数据，而实际应用系统中的数据形式多种多样，在计算机处理时都需要通过一定的方法将其转换成二进制形式。

1.1.1 数值型数据的表示

数值型数据是指具有特定值的一类数据。由于计算机只能识别“0”和“1”，因此日常生活中使用的十进制数也必须转换成二进制或者用二进制编码来表示。为了平时交流和书写方便，也经常使用八进制和十六进制来表示数值型数据。为了避免产生误解，在给出一串数值的同时必须指明这个数的进制。通常有两种表示方法：一种是在数值的后面加上一个表示进制的后缀，如B表示二进制数，Q表示八进制数，H表示十六进制数，D表示十进制数；另一种方式是在该串数据外加上括号，然后用下标的方式指明进制。例如：1010B、(1010)₂都表示二进制数1010，1010Q、(1010)₈都表示八进制数1010，1010H、(1010)₁₆都表示十六进制数1010。

1. 机器数与真值

根据用途的不同，数值型数据又可分为有符号数和无符号数，有符号数即正、负数。在日常生活中用“+”、“-”号加绝对值来表示数值的大小，用这种形式表示的数据在计算机技术中称为“真值”。对于有符号数据，需要把数据的符号（正号或负号）、小数点（数值型数据不可能都是整数）以及数据的值都进行数码化，即都用“0”和“1”来表示。而无符号数据除了没有符号的表示问题外，其余和有符号数据的表示方法一样。

对于有符号数，一般约定二进制数的最高位为符号位，并用“0”表示正号，用“1”表示负号。这种数值连同符号在内都数码化的数据表示称为机器数，它是计算机中使用的数据表示形式。

为了能正确地地区别出真值和各种机器数，用X表示真值， $[X]_{原}$ 表示原码， $[X]_{反}$ 表示反码， $[X]_{补}$ 表示补码。

2. 原码

原码是一种最简单、最直接的机器数表示法，其最高位为符号位，“0”表示正数，“1”表示负数，其余各位表示真值的绝对值。原码的定义为

若真值 $X \geq +0$ ，则 $[X]_{\text{原}} = X$ 。

若真值 $X \leq -0$ ，则 $[X]_{\text{原}} = 2^{n-1} - X$ ，其中 n 为机器字长。

例如，当机器字长为 8 时：

$$\begin{array}{ll} [+1]_{\text{原}} = 00000001 & [+99]_{\text{原}} = 01100011 \\ [-1]_{\text{原}} = 10000001 & [-99]_{\text{原}} = 11100011 \end{array}$$

原码表示法的优点是简单直观，转换容易，易于理解，缺点是加减运算规则复杂，并且 0 的表示不唯一。

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

3. 反码

反码表示法中，正数的反码最高位为符号位“0”，其余各位表示数的绝对值，也等于真值本身，即正数的反码与原码相同；负数的反码最高位为符号位“1”，其余各位为真值的数值部分按位取反，即等于其原码的数值位各位取反。反码定义为

若真值 $X \geq +0$ ，则 $[X]_{\text{反}} = X$ 。

若真值 $X \leq -0$ ，则 $[X]_{\text{反}} = 2^{n-1} + X$ ，其中 n 为机器字长。

例如，当机器字长为 8 时：

$$\begin{array}{ll} [+1]_{\text{反}} = 00000001 & [+99]_{\text{反}} = 01100011 \\ [-1]_{\text{反}} = 11111110 & [-99]_{\text{反}} = 10011100 \end{array}$$

在反码的表示中，0 的表示也不唯一。

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

4. 补码

使用原码和反码表示法，在进行数据运算时，符号位必须单独处理，而且数值 0 的表示方法不唯一，因此不适合计算机处理算术运算时使用。使用补码表示法可以避免这些问题，符号与数值部分可以一起参与运算，而且可以将减法运算转换为加法运算，从而有效简化了计算机的运算器设计。

理解补码需要引入“模”的概念。模是一个计量器的最大容量，当计数到达最大值的时候会溢出，又从最小值开始计数。好比把这些数排列成一个首尾相接的圆，当到达最大数（尾部）时再加上 1，就又回到了最小数（首部）。既然是一个圆，从其中的任何一个位置到达某个位置就有两种方式，一种是顺时针，另一种是逆时针，这两种方式所走过的路径和刚好是一个圆，所以称为互补。例如，一个时钟的时针停在 3 点钟的位置，要拨回到 1 点钟位置，有两种方法，一种是顺时针拨 10h，另一种是逆时针拨 2h。如果把顺时针定义成正数，逆时针定义成负数，则对于时钟而言（模为 12）+10 和 -2 互为补数。即：

$$(3+10) \bmod 12 = (3-2) \bmod 12$$

这样就可以把减法运算 $3-2$ 转换成加法运算 $3+10$ 来计算。

对于一个 n 位的二进制计数器，其模为 2^n ，补码定义为

若真值 $X \geq +0$ ，则 $[X]_{\text{补}} = X$ 。

若真值 $X \leq -0$ ，则 $[X]_{\text{补}} = 2^n + X$ ，其中 n 为机器字长。

补码的符号位表示方法与原码相同，其数值部分的表示与数的正负有关：对于正数，数值部分与真值形式相同；对于负数，将真值的数值部分按位取反，且在最低位上加 1。

用补码表示数值时，8 位二进制数补码所表示的真值范围为-128~127；并且 0 的表示也统一了，即 $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000$ 。

例如：

$$[+1]_{\text{补}} = 00000001$$

$$[+99]_{\text{补}} = 01100011$$

$$[-1]_{\text{补}} = 11111111$$

$$[-99]_{\text{补}} = 10011101$$

采用补码系统的计算机需要将真值或原码形式表示的数据转换为补码形式，以便于运算器对其进行运算。通常，从原码形式入手来求补码。

当 X 为正数时， $[X]_{\text{补}} = [X]_{\text{原}} = X$ 。

当 X 为负数时，其 $[X]_{\text{补}}$ 等于把 $[X]_{\text{原}}$ 除去符号位外的各位求反后，再在末位加“1”。

反之，当 X 为负数时，已知 $[X]_{\text{补}}$ ，也可通过对其除符号位外的各位求反加“1”求得 $[X]_{\text{原}}$ 。

当 X 为负数时，由 $[X]_{\text{原}}$ 转换为 $[X]_{\text{补}}$ 的另一种更有效的方法是：自低位向高位，尾数的第一个“1”及其右部的“0”保持不变，左部的各位取反，符号位保持不变。

例如：

$$[X]_{\text{原}} = 1. \quad 111001 \quad 1000$$

$$[X]_{\text{补}} = 1. \quad 000110 \quad 1000$$

不变 取反 不变

这种方法避免了加 1 运算，是实际求补线路逻辑实现的依据。

也可以直接由真值 X 转换为 $[X]_{\text{补}}$ ，其方法更简单：数值位自低位向高位，尾数的第一个“1”及其右部的“0”保持不变，左部的各位取反，负号用“1”表示即可。

例如：

$$X = -0.1010001010$$

$$[X]_{\text{补}} = 1.0101110110$$

当采用补码表示法时，可以把减法运算转换为加法运算：

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

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

【例 1-1】在字长为 8 位的情况下，计算 $X=50-18$ 。

$$[X]_{\text{补}} = [50-18]_{\text{补}} = [50]_{\text{补}} + [-18]_{\text{补}}$$

$$[50]_{\text{补}} = 00110010, [-18]_{\text{补}} = 11101110$$

$$00110010$$

$$+ 11101110$$

$$\hline 10010000$$

因为字长为 8 位，所以最高位的“1”丢失，得结果 $[X]_{\text{补}} = 00100000$ ，转换成十进制，真值为 32。

注 意

(1) 运算的范围在所能表示的数值范围内；

(2) $[[X]_{\text{补}}]_{\text{补}} = X$ 。

5. 三种机器数的比较

三种机器数既有共同点，又有各自不同的性质，主要区别有以下几点：

(1) 对于正数，它们都等于真值本身，而对于负数各有不同的表示。

(2) 最高位都表示符号位，补码和反码的符号位可作为数值位的一部分看待，和数值位一起参与运算；但原码的符号位不允许和数值位同等看待，必须分开处理。

(3) 对于真值 0，原码和反码各有两种不同的表示形式，而补码只有唯一的一种表示形式。

(4) 原码、反码表示的正、负数范围相对零来说是对称的；但补码负数表示范围较正数表示范围宽，能多表示一个最负的数（绝对值最大的负数），其值等于 -2^n （纯整数）或 -1 （纯小数）。

表 1-1 所示为真值与 3 种机器数间的对照。表中设字长为 4（含 1 位符号位）。

表 1-1 真值与 3 种机器数间的对照

真 值 X		[X] _原 、[X] _反 、[X] _补	真 值 X		[X] _原	[X] _反	[X] _补
十进制	二进制		十进制	二进制			
+0	+000	0000	-0	-000	1000	1111	0000
+1	+001	0001	-1	-001	1001	1110	1111
+2	+010	0010	-2	-010	1010	1101	1110
+3	+011	0011	-3	-011	1011	1100	1101
+4	+100	0100	-4	-100	1100	1011	1100
+5	+101	0101	-5	-101	1101	1010	1011
+6	+110	0110	-6	-110	1110	1001	1010
+7	+111	0111	-7	-111	1111	1000	1001
+8	—	—	-8	-1000	—	—	1000

在表 1-1 中，要特别注意 1000 这个代码，当其为原码时，对应的真值是-0；当其为补码时，对应的真值是-8；当其为反码时，对应的真值是-7。

1.1.2 定点数和浮点数

数值数据更多情况下既有整数部分，也有小数部分，并且整数部分和小数部分的位数是不固定的，所以在计算机中表示时需要指出小数点的位置。根据计算机表示数据时其小数点位置是否固定，可分为定点表示和浮点表示两种方法。

1. 定点数

定点表示时，所有数据的小数点位置固定不变，在计算机中没有表示小数点的位，小数点的位置是默认的，一般固定在有效数位的最前面或者末尾，这样就有两类定点数，即定点整数和定点小数。

1) 定点整数

小数点的位置约定在最低有效数位之后，其所能表示的是纯整数。如果用 $n+1$ 位来表示，则定点整数的格式如图 1-1 所示。

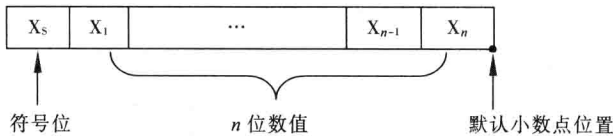


图 1-1 定点整数格式

从图 1-1 的表示格式可以看出， $n+1$ 位定点整数用补码来表示，数值范围为 $-2^n \sim 2^n-1$ 。如果机器字长为 8 位，则可表示的定点整数的范围为 $-128 \sim 127$ 。