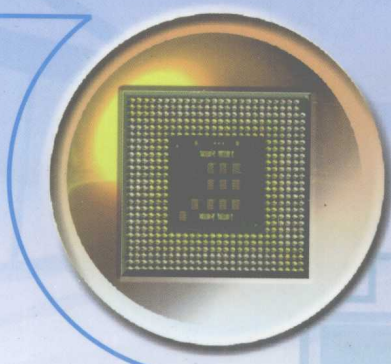


# 微机系统原理

# 与接口技术

主 编 李广军  
副主编 林水生 丁庆生  
          阎 波 何 羚



电子科技大学出版社

<http://www.uestcp.com.cn>

# 微机系统原理与接口技术

主 编

李广军

副主编

林水生 丁庆生 阎 波 何 羚

电子科技大学出版社

## 图书在版编目 (CIP) 数据

微机系统原理与接口技术/李广军主编. —成都: 电子科技大学出版社, 2005.2  
ISBN 7-81094-753-2

I. 微... II. 李... 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.TP36

中国版本图书馆 CIP 数据核字 (2005) 第 008603 号

## 内 容 提 要

本书对以 Intel 微处理器为核心的微机系统, 从计算机系统的简单模型入手, 全面、系统、深入地介绍了微机系统的组成、工作原理、总线标准、接口设计等相关技术及实际应用。本书注重吸取微机发展的最新技术和最新知识, 并结合了作者多年的科研、教学工作经验和体会。

本书共分 9 章, 分别介绍微机系统的基本知识、从 8086 到奔腾 IV 微处理器的内部结构、80X86 指令系统及汇编语言程序设计、存储器组织结构、中断系统、微机接口基础、微机系统总线规范及接口技术、微机接口芯片及应用实例等。本书给出了许多应用实例, 并附有大量习题。本书可作为高等院校理工科电子类专业本、专科学生的“微机原理”、“微机接口技术”等课程的教材, 也可作为工程技术人员的参考用书和考研用书。

## 微机系统原理与接口技术

主 编 李广军

副主编 林水生 丁庆生 阎 波 何 羚

---

出 版 电子科技大学出版社 (成都市建设北路二段四号, 邮编: 610054)  
责任编辑 张 勋 张 琴  
发 行 电子科技大学出版社  
印 刷 成都蜀通印务有限责任公司  
开 本 787×1092 1/16 印张 36 字数 922 千字  
版 次 2005 年 2 月第一版  
印 次 2005 年 6 月第二次印刷  
书 号 ISBN 7-81094-753-2/TP·406  
印 数 3001~6000 册  
定 价 45.00 元

---

# 前 言

为适应计算机和嵌入式系统、微电子和通信等技术领域的迅速发展，作者根据多年从事高校微型计算机系统原理、接口技术的教学实践以及应用开发的实际经验和体会，参照国家教育部和工科计算机课程教学指导委员会关于“微机系统原理与接口技术”教学体系的构想，并考虑到后 PC 技术、嵌入式系统/SOC 设计等相关技术的发展和需求，组织编写了“微机系统原理与接口技术”课程教学的本科生教材。

本书可作为高校理工科非计算机专业的高年级本科生学习有关“微型计算机系统原理和接口技术”、“现代微型计算机系统原理与应用”、“微机总线和接口技术”方面课程的教材。同时，也适合从事“现代微机应用与开发”、“嵌入式系统设计”工程技术人员作为参考用书。

虽然微型计算机和嵌入式计算机系统领域的技术发展令人目不暇接，新型 CPU、芯片组和接口芯片系列不断推出，但其性能和规模的发展仍然遵循了摩尔定律。无论是从计算机系统结构和总线接口技术原理的角度理解，还是从现代的嵌入系统/SOC 等新兴技术的发展趋势来看，各种计算机系统和 CPU 系列处理器在实质性的计算机技术和工作原理方面并无太大差别。

考虑到非计算机专业学生的计划内学时的限制，而计算机系统原理的建立需要一个由浅入深、循序渐进的过程，在指令系统和汇编语言程序设计中仍以 8086 为主，并进一步介绍其他 CPU 性能的改进和功能的扩展。

本书的主要内容是：以 80X86 为样机，主要介绍了微型计算机的数据类型、微处理器结构、微型计算机系统组成原理及基本工作原理；半导体存储器及存储器管理技术；寻址方式、指令系统与汇编语言程序设计；中断控制与管理、总线技术、输入/输出接口技术；并简要介绍了微型机系统的基本应用技术。本教材在编写过程中注意了内容安排由浅入深、循序渐进，对基本概念讲述清楚，并有大量实例，具有软硬结合、图文并茂、内容丰富及取材较新的特点，不仅适合作教材，而且可作为工程技术人员的自学用书。

本书在教材内容的选取与组织上掌握以下几点编写原则：

(1) 注意基础及实用性。重点介绍一般计算机的基本系统结构与组成原理。基础内容不针对某种机型，这样能使教学基本内容有相对的稳定性，同时又兼顾系统性；跟踪了微机系统及应用技术的发展趋向，使教材内容不失先进性。

(2) 淡化 CPU 内部的结构组成。从系统设计和编程应用的角度介绍 CPU 的功能、体系结构系统的建立和连接、接口技术与应用，为微机的应用以及硬件的扩展、开发等后续的课程打下一定的基础。

(3) 根据目前国内高校的教学与实验条件、应用的主流机型，选择 80X86 微处理器来具体讲述 CPU 的体系结构、寻址方式、指令系统、存储器、总线及其连接技术、汇编语言程序设计、可编程接口器件及接口技术和微机系统结构。

(4) 总线和接口设计是组成任何微机应用系统和嵌入系统的关键技术。虽然目前微机系统都有功能齐全的多功能外围接口芯片，如 82380 系列等，但从微机应用、嵌入系统、SOC

设计的角度,我们认为仍应以常用的可编程接口电路芯片,如 8255、8253、8250 等为典型的介绍对象,以便于学生能较好掌握软、硬件相结合的可编程接口技术。而众多的多功能外围接口芯片仅是上述基本功能芯片的集成而已。

(5)理工科非计算机专业学科较多,不同专业差别很大,对本课程的要求各异,有不同的教学时数。本书在编写时遵循了“宽编窄用”的内容选取原则和模块化、结构化的内容组织原则,以具有较宽的适应面和灵活的选择余地,利于实施不同对象、不同层次、不同学时数的教学。本教材在内容取材上尽量做到教学内容模块化,有些章节内容可以选学。本书参考学时数为 60~80。

作者认为,本书的核心内容是:微机系统组成原理、总线和接口技术。但不论内容和学时如何取舍,都应把实验放在重要的地位。

本书共 9 章,由李广军主编并编写第 8 章;林水生编写第 1、6、9 章;丁庆生编写第 2、5 章;阎波编写第 4、7 章;何羚编写第 3 章并整理附录。

本书的编写指导思想、编写大纲和内容取舍均由电子科技大学“微机系统原理与接口技术”课程组全体教师多次讨论审定,并得到了电子科技大学教务处、电子科技大学教材委员会及电子科技大学出版社的领导和专家们的热情帮助与支持。

本书是编者在多年从事微机系统原理与接口技术教学和应用开发的基础上,并参考了国内外大量文献资料编写而成,在此,特向有关作者表示感谢。

由于编者的水平有限,加之时间比较仓促,书中一定存在错误或不妥之处,请读者不吝指正。

编 者

2004 年 12 月

## 目 录

第 1 章 概述.....	1
1.1 信息在计算机中的表示.....	1
1.1.1 计算机中的数据存储类型.....	1
1.1.2 机器数与真值.....	2
1.1.3 原码、反码与补码.....	2
1.1.4 定点数与浮点数.....	3
1.1.5 计算机中常用码制.....	6
1.2 计算机发展概况.....	8
1.2.1 计算机的发展.....	8
1.2.2 微型计算机的发展.....	9
1.3 微机系统结构与工作原理.....	11
1.3.1 微机系统结构.....	11
1.3.2 微处理器简化模型.....	14
1.3.3 指令与程序.....	16
1.3.4 程序的执行过程.....	16
1.4 微机硬件.....	18
1.4.1 微机系统的主要性能指标.....	18
1.4.2 微机先进技术.....	19
1.4.3 微机系统的组成.....	21
1.4.4 微机系统举例.....	23
1.5 微机软件.....	27
1.5.1 计算机语言和语言处理程序.....	27
1.5.2 软件的分类型.....	28
1.5.3 操作系统的概念.....	29
1.6 嵌入式系统.....	30
1.6.1 单片机.....	30
1.6.2 嵌入式系统.....	30
习题.....	32
第 2 章 微处理器与系统结构.....	34
2.1 微处理器的性能指标.....	34
2.1.1 字长.....	34

2.1.2	指令数 .....	35
2.1.3	运算速度 .....	35
2.1.4	访存空间 .....	35
2.1.5	虚拟存储空间 .....	35
2.1.6	多处理器系统 .....	36
2.1.7	指令作业方式 .....	36
2.1.8	微处理器芯片的制造工艺 .....	36
2.2	8086 微处理器结构 .....	36
2.2.1	8086 微处理器的内部结构 .....	36
2.2.2	8086 微处理器的寄存器结构 .....	40
2.2.3	8086 微处理器的引脚特性 .....	43
2.2.4	8086 微处理器的总线特性 .....	46
2.2.5	8086 微处理器与 8088 微处理器的比较 .....	47
2.2.6	8086/8088 微处理器的工作模式 .....	48
2.3	8086 的总线操作及时序 .....	52
2.3.1	时钟信号发生器芯片 8284A 介绍 .....	52
2.3.2	总线周期 .....	52
2.3.3	总线操作时序 .....	54
2.4	8086 的存储器与 I/O 组织结构 .....	58
2.4.1	存储器的标准结构 .....	58
2.4.2	8086 存储器的奇偶库结构 .....	59
2.4.3	8086 存储器的分段结构 .....	60
2.4.4	物理地址和逻辑地址 .....	62
2.4.5	8086 存储器的堆栈结构 .....	62
2.4.6	存储器的存储区分配 .....	63
2.4.7	存储器与 I/O 组织 .....	64
2.5	80X86 微处理器的结构 .....	65
2.5.1	80186/80188 微处理器 .....	65
2.5.2	80286 微处理器 .....	65
2.5.3	80386 微处理器 .....	68
2.5.4	80486 微处理器 .....	75
2.5.5	Pentium 和 Pentium Pro 微处理器 .....	76
2.5.6	Pentium II、Pentium III 及 Pentium IV 微处理器 .....	76
	习题 .....	78
<b>第 3 章</b>	<b>80X86 指令系统 .....</b>	<b>81</b>
3.1	8086 指令格式与编码 .....	81
3.2	8086 寻址方式 .....	82

3.2.1	非存储器寻址方式 .....	83
3.2.2	存储器寻址方式 .....	84
3.2.3	段寄存器的使用规则 .....	87
3.3	8086 指令系统 .....	89
3.3.1	数据传送指令 .....	89
3.3.2	算术运算指令 .....	96
3.3.3	逻辑运算和移位指令 .....	107
3.3.4	串操作指令及重复前缀 .....	114
3.3.5	控制转移指令 .....	119
3.3.6	输入/输出指令 .....	134
3.3.7	处理器专用指令 .....	135
3.4	80X86 增强和扩充的指令 .....	137
3.4.1	80286 增强和扩充的指令 .....	137
3.4.2	80386 增强和扩充的指令 .....	141
3.4.3	80486 扩充的指令 .....	142
	习题 .....	143
<b>第 4 章</b>	<b>宏汇编语言程序设计 .....</b>	<b>148</b>
4.1	汇编语言程序设计的特点 .....	148
4.2	8086 宏汇编源程序的组成 .....	149
4.2.1	段 .....	149
4.2.2	语句 .....	150
4.3	8086 宏汇编语言的基本语法 .....	152
4.3.1	字符集 .....	152
4.3.2	常量 .....	152
4.3.3	保留字 .....	153
4.3.4	标识符 .....	153
4.3.5	表达式 .....	154
4.4	8086 宏汇编语言常用伪指令 .....	157
4.4.1	数据定义伪指令 .....	158
4.4.2	程序结构伪指令 .....	161
4.4.3	模块链接伪指令 .....	165
4.4.4	列表控制伪指令 .....	165
4.4.5	与宏有关的伪指令 .....	166
4.4.6	条件汇编伪指令 .....	170
4.4.7	结构性数据定义伪指令 .....	171
4.5	8086 宏汇编语言程序设计技术 .....	173
4.5.1	汇编源程序的基本结构形式 .....	173



4.5.2	汇编源程序的设计步骤 .....	176
4.5.3	汇编源程序的模块化设计方法 .....	178
4.5.4	实用程序设计例程 .....	189
	习题 .....	216
<b>第 5 章</b>	<b>存储器 .....</b>	<b>223</b>
5.1	存储器的分类及性能指标 .....	223
5.1.1	计算机存储器的发展 .....	223
5.1.2	内存储器与外存储器 .....	223
5.1.3	存储器的分类 .....	224
5.1.4	半导体存储器的性能指标 .....	226
5.1.5	半导体存储器的结构特点 .....	227
5.2	半导体存储器 .....	229
5.2.1	随机存取存储器 .....	229
5.2.2	只读存储器 .....	236
5.2.3	常用存储器件 .....	240
5.3	内存储器系统的设计 .....	250
5.3.1	存储器芯片的选择 .....	250
5.3.2	存储器系统的地址分配 .....	251
5.3.3	存储器芯片与微处理器的三总线连接 .....	252
5.3.4	存储器芯片的片选控制方法 .....	252
5.3.5	线选法的应用 .....	253
5.3.6	部分译码法的应用 .....	255
5.3.7	全译码法的应用 .....	256
5.3.8	内存储器系统设计总结 .....	260
5.4	微机存储器的层次结构及管理 .....	260
5.4.1	高速缓存-主存体系结构 .....	261
5.4.2	三级存储器层次结构 .....	261
5.4.3	存储器管理 .....	262
	习题 .....	262
<b>第 6 章</b>	<b>微机接口基础 .....</b>	<b>264</b>
6.1	输入/输出接口概述 .....	264
6.1.1	为什么需要接口 .....	264
6.1.2	接口电路中的信息 .....	265
6.1.3	接口电路的基本功能和结构 .....	266
6.2	I/O 端口的编址方法 .....	268
6.2.1	存储器映像编址方法 .....	268
6.2.2	独立 I/O 编址方法 .....	269

6.3 I/O 接口的端口地址译码.....	270
6.3.1 I/O 端口的地址分配.....	270
6.3.2 I/O 端口的地址译码.....	271
6.4 I/O 接口的数据传输控制方式.....	276
6.4.1 直接程序控制方式.....	277
6.4.2 中断控制方式.....	282
6.4.3 直接存储器存取控制.....	283
6.5 基本输入/输出接口.....	285
6.5.1 键盘/鼠标接口.....	285
6.5.2 数码管显示接口.....	293
6.5.3 并行打印机接口.....	296
习题.....	305
<b>第 7 章 中断</b> .....	<b>307</b>
7.1 中断概述.....	307
7.1.1 中断处理方式的特点.....	307
7.1.2 中断系统.....	308
7.1.3 中断处理过程.....	309
7.1.4 程序中断与子程序调用的区别.....	309
7.2 8086 的中断系统.....	310
7.2.1 8086 系统的中断源.....	310
7.2.2 8086 系统中断索引的实现.....	312
7.2.3 8086 系统中断处理的全过程.....	313
7.2.4 8086 系统中有关 INTR 中断的处理.....	316
7.3 80X86 系列高档微处理器的中断系统.....	322
7.3.1 80486 系统的中断源.....	322
7.3.2 80486 系统的中断全过程.....	323
7.3.3 80486 系统对中断/异常的管理.....	324
7.4 80X86 系列微机的硬中断控制逻辑.....	326
7.4.1 IBM PC/XT 的硬中断控制系统.....	326
7.4.2 高档微机的硬中断控制系统.....	327
7.4.3 PC 机 INTR 中断的应用.....	328
习题.....	329
<b>第 8 章 系统总线与接口技术</b> .....	<b>331</b>
8.1 总线的概念和分类.....	331
8.1.1 总线的概念.....	331
8.1.2 总线的分类.....	331
8.1.3 并行总线.....	332

8.1.4	串行总线 .....	338
8.2	ISA 总线 .....	348
8.2.1	ISA 总线的主要特点 .....	349
8.2.2	ISA 总线信号 .....	349
8.2.3	ISA 总线的电源规格和机械尺寸 .....	352
8.2.4	ISA 总线的典型操作时序 .....	353
8.3	USB 接口 .....	356
8.3.1	概述 .....	357
8.3.2	USB 的物理接口和电气特性 .....	358
8.3.3	USB 系统组成 .....	358
8.3.4	USB 通信流 .....	360
8.3.5	USB 的传输方式 .....	361
8.3.6	USB 交换的包格式 .....	362
8.3.7	USB 2.0 与 USB OTG .....	364
8.3.8	典型 USB 接口器件及应用 .....	366
8.4	PCI 总线 .....	373
8.4.1	PCI 总线的特点 .....	373
8.4.2	PCI 总线信号线定义 .....	376
8.4.3	总线命令 .....	380
8.4.4	PCI 总线协议 .....	386
8.4.5	总线上的数据传输过程 .....	390
8.4.6	PCI 配置寄存器 .....	394
8.4.7	PCI 总线的其他功能 .....	396
8.4.8	PCI 总线的电气规范 .....	399
8.4.9	PCI 总线接口设计 .....	400
	习题 .....	405
第 9 章	微机接口芯片及应用 .....	408
9.1	可编程并行接口 8255 .....	408
9.1.1	8255 的引脚和内部结构 .....	409
9.1.2	8255 的控制字 .....	412
9.1.3	8255 的工作方式 .....	414
9.1.4	8255 应用 .....	425
9.2	可编程串行接口 8250/16550 .....	430
9.2.1	8250/16550 的引脚和内部结构 .....	430
9.2.2	8250/16550 初始化及应用编程 .....	436
9.2.3	串行通信的 BIOS 功能调用 .....	439
9.3	可编程定时/计数器 .....	440

9.3.1	8254/8253 的引脚和内部结构	441
9.3.2	8253/8254 的控制字	443
9.3.3	8254/8253 的工作方式	445
9.3.4	8254 与 8253 的比较	451
9.3.5	8254/8253 的应用	452
9.4	可编程中断控制器 8259	459
9.4.1	8259 的引脚和内部结构	459
9.4.2	8259 的控制命令字	463
9.4.3	8259 的工作方式	468
9.5	可编程 DMA 控制器	474
9.5.1	8237 的引脚信号及内部结构	474
9.5.2	8237 的控制字与寄存器	478
9.5.3	8237 编程及应用	482
9.6	模/数和数/模接口	485
9.6.1	A/D 接口芯片	485
9.6.2	D/A 接口芯片	493
9.6.3	A/D、D/A 的综合应用	499
9.7	多功能高集成 I/O 控制器 82801	506
9.7.1	82801 结构及引脚信号	507
9.7.2	82801 的 I/O 空间	511
	习题	514
附录 1	ASCII 码表	517
附录 2	8086 指令系统一览	519
附录 3	8086 指令对标志位的影响	526
附录 4	8086 宏汇编常用伪指令	527
附录 5	汇编语言源程序上机调试过程	529
附录 6	8086 宏汇编出错信息说明	536
附录 7	DEBUG 常用命令	541
附录 8	PC 机系统中断类型码及中断源	543
附录 9	DOS 功能调用 (INT 21H)	545
附录 10	BIOS 功能调用	554
附录 11	常用数字器件引脚及功能	560
	参考文献	563

# 第 1 章 概 述

本章从计算机的初级模型结构入手，介绍了微机系统的一些基础知识，包括计算机系统中信息的表示方式，微机系统的组成、结构与工作原理，以及嵌入式系统的基本概念。

## 1.1 信息在计算机中的表示

计算机不仅能进行数值计算，而且还能处理文字、声音、图形、图像等各种信息，这些信息统称为数据。具有数值大小和正负特性的数据称为数值数据；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为非数值数据。两者在计算机内部都是以不同形式编码的二进制数来表示和存储的。

### 1.1.1 计算机中的数据存储类型

计算机系统处理的数据对象可分为位 b (bit)、字节 B (Byte)、字 W (Word)、双字 DW (DoubleWord)、四字 QW (QuadWord) 等。

位是计算机系统所能表示的最基本、最小的数据单元，也就是二进制数中的一个数值位。一个位只有两种取值情况：即“0”或“1”。位所对应的是硬件电路中单根导线上“低”电平或“高”电平的状态。

8 个位（严格地说是 8 个二进制位）作为一个整体就构成 1 个字节。计算机中的存储容量就是用字节数来度量的，度量单位有 KB、MB、GB 和 TB，其换算关系为：

$$1\text{KB}=2^{10}\text{B}=1\ 024\text{B}$$

$$1\text{MB}=2^{10}\text{KB}=2^{20}\text{B}=1\ 048\ 576\text{B}$$

$$1\text{GB}=2^{10}\text{MB}=2^{30}\text{B}=1\ 073\ 741\ 824\text{B}$$

$$1\text{TB}=2^{10}\text{GB}=2^{40}\text{B}=1\ 099\ 511\ 627\ 776\text{B}$$

2 个字节联合成整体就是字，因此 1 个字由 16 个二进制位构成；双字就是 2 个字一起作为数据整体的数，共 32 位；四字则是由 4 个字一起构成的数，共 64 位，如图 1-1 所示。

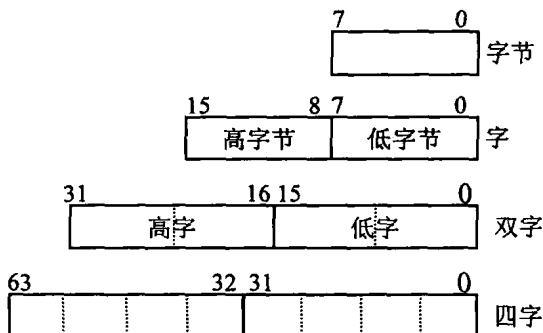


图 1-1 计算机系统中的数据储存类型

在计算机系统中，数据的传输、存储通常以  $2^n$  个字节为单位，例如 1 个字节、2 个字节或 4 个字节等。计算机中存储的多字节数据一定占用连续的存储器单元，各种类型的数据可以交叉存储，如字节之后存储的是双字数据，然后存储字数据等。计算机中存储的数据根据其具体应用不同，可能是数据，也可能是地址信息或状态信息等。

### 1.1.2 机器数与真值

计算机中只能表示“0”和“1”两种符号，因此为了表示正数和负数，可专门用一位来表示数的符号，如使用数据的最高位作为符号位，当其为“0”时，表示正数；而当其为“1”时，表示负数。这就是说，数的符号在计算机中也被数码化了。

我们把用“+”和“-”表示符号的带符号数称为真值，而用“0”和“1”表示符号的带符号二进制（Binary）数称为机器数。机器数除了可以写成二进制的形式外，通常也写成十六进制（Hex）的形式。如真值为+22 106 的数所对应的机器数可写为 0101 0110 0101 1010B 或 565AH；真值为-22 106 的数所对应的机器数则可写为 1101 0110 0101 1010B 或 0D65AH。

为了区分标识符与数值，当十六进制数的最高位为 A~F 的数值符号时，在该数值的最前面再加一个 0，表示它是数值而非标识符，但该数值在计算机中传输和存储时，添加的 0 既不需要传输，也不占用存储器，或者说这个 0 实际上是不存在的，只是数值书写的标识而已。

### 1.1.3 原码、反码与补码

机器数的符号数码化后，带符号数在计算机中的传输、存储就没有任何问题了。但在进行运算时，又应该如何处理呢？符号位需要区别对待还是与数值位一起进行处理呢？为解决这些问题，引入了机器数的 3 种不同编码形式，即原码、反码和补码，分别用  $[X]_{\text{原}}$ 、 $[X]_{\text{反}}$ 、 $[X]_{\text{补}}$  表示。

#### 1. 原码

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

$$[X]_{\text{原}} = \begin{cases} 0X_1X_2\cdots, & X \geq 0 \\ 1X_1X_2\cdots, & X \leq 0 \end{cases}$$

即一个数的原码，就是数值部分不变而用“0”和“1”分别表示数的符号“+”和“-”的机器数。

在原码表示法中，根据定义，数 0 的原码有两种不同的形式，字长为 8 位时的两种表示分别为  $[+0]_{\text{原}}=0000\ 0000\text{B}$ ， $[-0]_{\text{原}}=1000\ 0000\text{B}$ 。

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的运算，因为两个原码数进行运算时，首先要判断它们的符号，然后再决定是用加法还是用减法，因而使计算机结构复杂化，运行时间延长。为解决上述弊病，引入了反码和补码表示法。

#### 2. 反码

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

$$[X]_{\text{反}} = \begin{cases} 0X_1X_2\cdots X_{n-1}, & X \geq 0 \\ 1\bar{X}_1\bar{X}_2\cdots \bar{X}_{n-1}, & X \leq 0 \end{cases}$$

式中,  $\bar{X}_i$  表示对  $X_i$  取反。

对于正数, 反码与原码相同, 即符号位为 0, 其余位为数值位本身; 对于负数, 符号位为 1, 数值位按位求反 (原来为 1 的位变为 0, 而原来为 0 的位变为 1)。

在反码表示法中, 根据定义, 数 0 的反码也有两种不同形式, 字长为 8 位时 0 的两种反码表示分别是  $[+0]_{\text{反}}=0000\ 0000\text{B}$ ,  $[-0]_{\text{反}}=1111\ 1111\text{B}$ 。

反码表示相对复杂, 但用反码运算时, 则要简单一些, 不需要判断数值的正负, 也不需要比较数值的大小。但运算结果的最高进位需要返回到最低位进行修正, 才能得到正确的结果, 从而影响了运算速度。

### 3. 补码

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

$$[X]_{\text{补}} = \begin{cases} 0X_1X_2\cdots X_{n-1}, & X \geq 0 \\ 1\bar{X}_1\bar{X}_2\cdots\bar{X}_{n-1}+1, & X < 0 \end{cases}$$

式中,  $\bar{X}_i$  表示对  $X_i$  取反。

或

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

对于正数, 补码与原码相同, 即符号位为 0, 其余位为数值位本身; 对于负数, 补码为反码再加 1。也可直接由原码求补码, 方法是: 符号位为 1, 原码数值中的最右边一个 1 及其右边的所有 0 都保持不变, 其余各位按位求反。

在补码表示法中, 0 的补码是惟一的。补码的运算比较简单, 而且可以将减法运算变换为加法运算。因此在计算机系统中, 只需要设计加法器电路, 就可完成加法和减法运算, 并通过循环加法实现乘法、循环减法实现除法, 再进一步实现任意复杂程度的运算。不过为了使运算速度得到更快提高, 现代计算机系统中直接设计了硬件乘法器。

由于补码具有运算过程简单、效率高、硬件电路结构简单和便于实现各种运算等优点, 计算机系统中的带符号数通常都采用补码表示法。

数的补码进行运算时, 满足:

$$X = [[X]_{\text{补}}]_{\text{补}}$$

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

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

在计算机系统中, 两个带符号数运算的结果若超出了允许的表示范围, 就会发生溢出, 即意味着结果是错误的。如何判断溢出呢? 进行加法运算时, 如果次高位和最高位都不产生进位或都产生进位, 结果不溢出; 而如果有且只有一处产生进位, 则结果溢出。进行减法运算时, 如果次高位和最高位都不产生借位或都产生借位, 结果不溢出; 而如果有且只有一处产生借位, 则结果溢出。

#### 1.1.4 定点数与浮点数

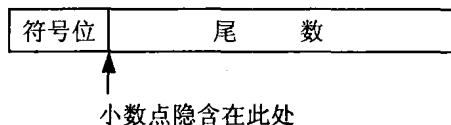
在计算机中, 不仅要处理整数运算, 而且也要处理小数运算。与带符号数的符号位表示不同, 小数点在计算机中并没有表示, 也不需要存储, 而是约定, 即大家统一认可小数点所

处的位置。因此，小数点的位置处理很重要。如果小数点的位置在数的表示中是固定的，则称为定点表示法。另外一种小数的表示方法为浮点表示法。

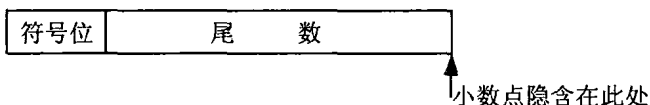
### 1. 定点表示法

定点表示法要求小数点的位置是固定的，但具体在什么位置并无限制。在计算机中，有两种定点数是最常用的（但在具体应用时，还是需要事先约定好）：

一种是定点纯小数，把小数点固定在最高数值位左边，在小数点左边留有一位符号位，而且小数点本身不占位，其格式为：



另一种是定点纯整数，把小数点固定在最低数值位右边，最高位为符号位，而且小数点本身也不占位，其格式为：



【例 1.1】下面的 8 位二进制补码数表示的真值是多少？



【例 1.2】下面的 8 位二进制补码数表示的真值是多少？



从上面的例子可以看出，定点纯整数和定点纯小数在格式上毫无差别，但真值却大不一样，这是因为它们有小数点，而小数点的位置不同但却是隐含的。还需要说明的是：相同位数、不同编码的定点数所表示的数的范围是不同的，如表 1-1 所示。

表 1-1 n 位定点数的范围

码制	n 位定点纯整数	n 位定点纯小数	8 位整数	8 位纯小数
$[X]_{\text{原}}$	$-2^{n-1}-1 \sim 2^{n-1}-1$	$-(1-2^{-(n-1)}) \sim 1-2^{-(n-1)}$	-127~127	-0.992 187 5~0.992 187 5
$[X]_{\text{反}}$	$-2^{n-1}-1 \sim 2^{n-1}-1$	$-(1-2^{-(n-1)}) \sim 1-2^{-(n-1)}$	-127~127	-0.992 187 5~0.992 187 5
$[X]_{\text{补}}$	$-2^{n-1} \sim 2^{n-1}-1$	$-1 \sim 1-2^{-(n-1)}$	-128~127	-1~0.992 187 5

### 2. 浮点表示法

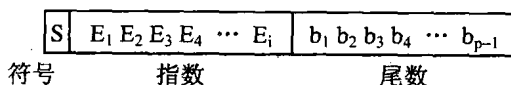
为了在位数有限的前提下，尽量扩大位的表示范围，同时又保持位的有效精度，计算机往往用浮点数表示数值。浮点数是用指数和尾数来表示的，由于指数可以选择不同的编码方式，尾数的格式和小数点的位置也可以有不同的约定，因此浮点数的表示方法不是惟一的。计算机大多采用 IEEE 规定的浮点数表示方法，即：



$$(-1)^S 2^E (b_0 \nabla b_1 b_2 b_3 \cdots b_{p-1})$$

式中,  $(-1)^S$  是该数的符号位,  $S=0$  表示正数,  $S=1$  表示负数。E 为指数, 是一个带偏移量的整数, 表示成无符号的整数形式。 $(b_0 \nabla b_1 b_2 b_3 \cdots b_{p-1})$  是尾数, 共有 p 位二进制,  $\nabla$  表示隐含的小数点位置。

在表示成规格化的形式时, 通常  $b_0=1$ , 并与小数点一起被隐含了, 二进制编码的规格化形式为:



在奔腾微机系统中, 浮点数有 3 种类型: 单精度浮点数、双精度浮点数及扩充精度浮点数, 如表 1-2 所示。

表 1-2 微机系统的 3 种浮点数类型

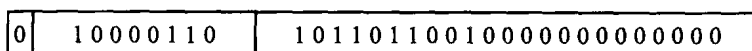
浮点数中的参数	单精度浮点数	双精度浮点数	扩充精度浮点数
浮点数长度 (位数)	32	64	80
尾数长度 p (位数)	24	53	65
符号位 S (位数)	1	1	1
指数长度 E (位数)	8	11	15
最小指数	-126	-1 022	-16 382
最大指数	+127	+1 023	+16 383
指数的偏移量值	+127	+1 023	+16 383

【例 1.3】将十进制数 219.125 表示成单精度浮点数。

分析: 十进制数 219.125 的二进制形式为 1101 1011.001B。

二进制数化成规格化形式:  $1101 1011.001B=1.1011 0110 01 \times 2^7$ 。

指数为 7, 所以  $E=7+127=134=1000 0110B$ 。则可得到二进制表示的规格化的浮点数形式:



符号                  指数                  尾数 (共 23 位), 包括隐含的  $b_0$  共 24 位

【例 1.4】已知单精度浮点数为 1100 1000 0101 0110 0010 0000 0000 0000, 求其对应的真值。

由给定的浮点数格式可知  $S=1$ , 表示是负数,  $E=1001 0000B=144$ , 所以指数为  $E=144-127=17$ , 尾数为后 23 位, 去掉后面无效的 0, 该数的真值为:

$$-1.1010 1100 01 \times 2^{17} = -11 0101 1000 1000 0000B = -35880H = -219264$$

需要说明的是: 在以前的微机系统中, 把一个浮点数 N 表示成 2 的 P 次幂与绝对值小于 1 的数 S 相乘的形式:

$$N=2^P S$$

式中, S 是 N 的尾数, 为数值的有效数字部分, 通常用带符号的定点小数表示, 一般用原码表示。P 是指数, 即阶码, 通常用带符号整数的补码表示。阶码 P 的大小规定了数的范围, 尾数 S 的长度规定了数的有效数字的位数 (精度)。二进制编码的形式为: