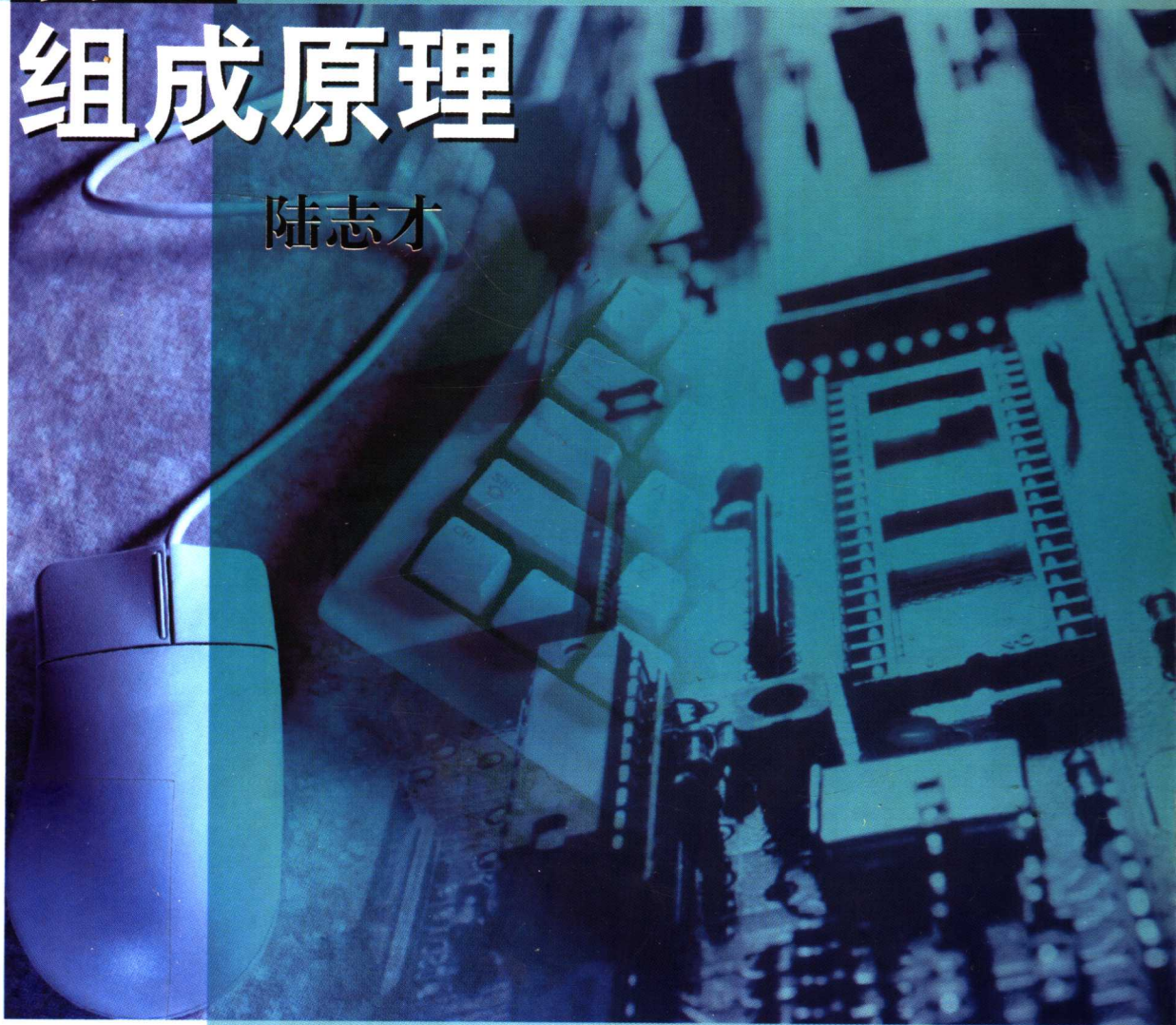
普通高等教育“十五”国家级规划教材

微型计算机组成原理

陆志才



高等教育出版社

普通高等教育“十五”国家级规划教材

微型计算机组成原理

陆志才

高等教育出版社

内容提要

本书是由教育部组织的普通高等教育“十五”国家级规划教材。本书从必备的基础讲起，系统介绍了16位和包括Pentium 4、Xeon在内的现代32位微型计算机的组成原理、体系结构及基本接口技术。其内容涉及：微型计算机组成基础、各种档次的处理器结构、各种类型的半导体存储器、输入/输出接口、可编程接口芯片、传统和现代的控制逻辑、面向现代微型计算机的总线技术以及传统的和现代的乃至下一代微型计算机的体系结构等。

本书具有以下特色：基础部分简要，讲解通俗易懂；内容新，尤其突出现代微型计算机的组成，对地位越来越重要的总线技术和控制芯片组作了重点介绍；大部分程序示例采用高级语言编写；内容的选择、深浅把握及顺序安排经过精心设计。

本书可作为高等院校计算机科学与技术及相关专业的教材，亦可作为从事研发、生产、教学和应用开发的广大科技工作者的自学用书。

图书在版编目(CIP)数据

微型计算机组成原理/陆志才. —北京：高等教育出版社，2003.2

ISBN 7-04-011772-X

Ⅰ.微… Ⅱ.陆… Ⅲ.微型计算机-计算机体系结构 Ⅳ.TP360.3

中国版本图书馆CIP数据核字(2003)第000419号

出版发行	高等教育出版社	购书热线	010-64054588
社　　址	北京市东城区沙滩后街55号	免费咨询	800-810-0598
邮政编码	100009	网　　址	http://www.hep.edu.cn
传　　真	010-64014048		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	北京未来科学技术研究所有限责任公司印刷厂		
开　　本	787×1092　1/16	版　　次	2003年2月第1版
印　　张	30.5	印　　次	2003年2月第1次印刷
字　　数	620 000	定　　价	29.00元

前　言

“计算机组成原理”和“微机原理与接口技术”是计算机科学与技术专业及相关电子工程类专业的重要技术基础课，同时直接面向应用，在计算机学科及相关电子工程类学科中起着相当重要的作用。随着计算机技术和集成电路技术的发展，微机早已成为主流机型，因此，在“计算机组成原理”教材中有一定篇幅介绍微机。这样，两门组成原理在内容上存在重复。此外，为了适应计算机软件、网络、多媒体等新技术的发展，在教学计划中需要压缩硬件课程的学时数。本教材正是为了适应上述需要而编写的。

本书除将两门课有机地合并为一门之外，还具有以下特色：

1. 基础部分简要、扎实

从必备的基础讲起，由简单到复杂，循序渐进。在介绍时不涉及过多的细节，不将教材写成某一机器和某些芯片的技术说明书或技术手册，但对于重要的基础还是予以展开，有思路，有分析。

2. 内容新，突出现代微机的组成

在现代微机的硬件组成中总线和控制芯片组的作用越来越重要，并且在微机的应用和开发（包括操作系统等系统软件的开发）中需要这些知识。本书对这两方面做了重点介绍，几乎包括了现代微机中所有总线类型，乃至下一代总线。

3. 大部分程序示例采用高级语言编写

考虑到微机程序的开发，包括开发与硬件相关的程序，大多采用高级语言，本书的大部分程序示例采用 DOS 平台上的 Turbo _ C 编写，对 Windows 环境下的编程也做了简介。

4. 内容选择、深浅把握及顺序安排经过精心设计

编者从事计算机教学与科研多年，早先还参加过计算机的设计、研制，了解计算机的发展，对两门组成原理中哪些是基础，对其相互之间的关系较为清楚。本书内容的选择、深浅把握及顺序安排都经过精心设计，实际上也是多年教学经验的积累。

本书系统介绍了 16 位和包括 Pentium 4、Xeon 在内的现代 32 位微机的组成原理、体系结构和基本接口技术。全书共 12 章。第一章集中介绍属于传统计算机组成原理的一些必备基础知识（其他相关原理分散在后续章节）。第二章简要介绍 8086 微处理器、总线周期及系统总线的形成。第三章讨论了各种类型的半导体存储器、简单存储器子系统的设计以及高速缓冲存储器（Cache）。第四章对主机与外设之间的数据传送方式、输入/输出接口的基本结构及简单输入/输出接口的设计做了阐述。第五章讨论了中断系统，其中包括现代微机中

的中断控制逻辑。第六章用较大篇幅介绍了可编程接口芯片。第七章介绍现代微处理器的基础——80386,涉及80386本身的结构、总线周期、重要的内部机制以及系统的构成。第八章先对高档微处理器中采用的新技术做集中说明,然后分别列举Intel公司x86系列每一种微处理器的主要特点。第九章在对总线技术做了概述后介绍了现代微机所采用的内部总线:ISA、PCI和AGP。第十章讨论了现代微机最常用的几种设备总线:SCSI、IDE、USB,并对FireWire总线做了简介。第十一章在介绍控制逻辑和控制芯片组的基础上重点讨论了各个阶段(包括下一代)微机的系统结构。到第十一章结束,读者对现代微机的硬件组成可建立起一个完整的印象。第十二章从软硬结合的角度介绍了微机中的串行通信。

考虑到不少院校在原理课之前已安排了汇编语言课和C语言课,本书对这两种语言未做专门介绍。要求读者具有阅读程序和编写简单程序或程序段的能力。此外,要求读者具有最基本的数字电路基础。

本教材建议讲授72学时,目录中标注"*"的内容可安排学生自学。本课程的实践教学内容单独安排。

微机的组成相当复杂,即使是对一个局部,也往往难以给出完整的电路图。因此本书中绝大多数插图为框图。由于框图只是组成的粗略表示,所以建议读者在分析这些图时最好不要延用传统的电路分析方法——由输入导出输出。也就是说,要逐步学会从不同角度(确切地说,从不同高度)观察微机的组成,重点放在部件之间的联系,有时还要把时间关系考虑进去。

在本书中,逻辑符号采用国家标准表示。附录中给出了本书涉及到的国家标准与国外流行表示的对照关系。

本书的编写得到高等教育出版社的大力支持。南开大学计算机科学与技术系几届研究生何凯、王德谦、胡科森、商艳莉为本书的编写做了大量基础研究和实践工作,商艳莉硕士还为本书的部分初稿提出宝贵建议。天津市软件与理论重点学科负责人南开大学教授杨恩鲁博士为本书的编写提供了资助。在此一并表示衷心感谢。

本书编写过程中参考了国内外大量优秀教材和文献资料,并参考了www.intel.com、www.ansi.org、www.usb.org、www.pcisig.com、www.atmel.com、www.adaptec.com等网站的资料,在此,向有关作者一并致谢。

由于编者水平有限,时间仓促,书中难免有错误和不当之处,敬请读者和同行专家批评指正。编者的电子信箱是:luzhc@nankai.edu.cn。

本书的多媒体教学光盘正在制作,请需要的教师与编者联系。

编者

2003年1月于南开大学

目　录

* 对应的内容可安排学生自学。

第一章　微型计算机组成基础

电子计算机的产生(1946年)和发展是20世纪人类最伟大的成就之一。在短短的50多年中,电子计算机的组成经历了以电子管、晶体管、中小规模集成电路以及大规模和超大规模集成电路为主要标志的四代的变化。与此同时,计算机软件技术也发生了巨大的变化。随着计算机技术和大规模集成电路技术的发展,微型计算机应运而生(20世纪70年代初期),并得到长足发展,尤其从20世纪90年代中期开始,更呈现突飞猛进之势。现代微型计算机的功能已远远超过过去的大型计算机。事实上,微型计算机的组成也经历了以微处理器位数为主要标志的四代变化,即从4位和低档8位机到中档和高档8位机,再到16位机,然后到32位机的发展。下一个目标是64位机。

面对功能强大、结构复杂的现代微型计算机,要学习其组成原理,应从基础开始,循序渐进地学习,因此,本章将介绍一些必备的基础知识,其中大部分属于传统计算机组成原理课程的内容,具体包括:计算机中数据信息的表示、初级计算机的组成和运行原理、微型计算机的基本结构、微型计算机的基本数据类型、堆栈。

1.1　计算机中数据信息的表示

1.1.1　计算机中数的进位制

学习计算机,首先要弄清计算机中数的进位制。关于这一点,可用一句话来概括:当使用汇编语言或高级语言编程时一般采用十进制表示,有时出于某种需要也采用十六进制或二进制表示,而在计算机内部,数据的表示、存储及运算均采用二进制。

1. 二进制

1) 十进制回顾

可以用三点来描述十进制:一是有十个数码(0、1～9);二是计数时逢十进一;三是采用位置表示法。所谓位置表示法,是指同一个数码在数中不同位置所表示的值不同。例如,数535.5中,数码5出现在百位,代表500;出现在个位,代表5;出现在小数点后第一位,代表0.5。一个数码所表示的值等于它乘以该位的权。所谓"权"是指某一位单位数字所表示的

值。对十进制而言，从小数点向左，各位的权依次是 $1(10^0)$、$10(10^1)$、$100(10^2)$、$1\ 000(10^3)$…；从小数点向右，各位的权依次是 $0.1(10^{-1})$、$0.01(10^{-2})$、$0.001(10^{-3})$…。高一位的权是低一位的 10 倍。

2）什么是二进制

和十进制相类似，也可以用三点来描述二进制：一是有两个数码（0 和 1）；二是计数时逢二进一；三是采用位置表示法。从小数点向左，各位的权依次是 2^0、2^1、2^2、2^3…；从小数点向右，各位的权依次是 2^{-1}、2^{-2}、2^{-3}…。高一位的权是低一位的 2 倍。

3）计算机中为什么采用二进制

计算机中采用二进制是基于以下三点：

（1）容易表示

从原理上讲，只要有两种稳定状态、并能进行状态转换的电路都可以用来表示二进制数码，用其中一种状态表示 0，用另一种状态表示 1。而表示十进制数码的单元电路必须有十种稳定状态，并能方便进行状态转换，时至今日还没有这样的电路。

（2）运算简单

以一位数的加法为例，对十进制需 55 条规则，而对二进制仅需 3 条规则（0 + 0 = 0；0 + 1 = 1 + 0 = 1；1 + 1 = 0，向高一位进位 1）。

（3）便于使用一种数学工具——逻辑代数进行运算

在计算机中，算术运算是通过逻辑电路来实现的。当然，逻辑运算也通过逻辑电路来实现。逻辑电路是和逻辑表达式相对应的。设计逻辑电路时，往往是先写出逻辑表达式，再进行电路实现。借助于逻辑代数，可将逻辑表达式变换成最简形式，从而使实现表达式的电路最简单。

4）二进制数与十进制数的相互转换

（1）二进制数转换成十进制数

将每一位的数码乘以该位所对应的权，再求所得乘积的和。这种方法叫按权展开求和。例如：

$$1011.101_{(2)} = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} = 11.625_{(10)}.$$

由于二进制数的每一位只有 0 和 1 两种可能，而任何数乘 1 得本身，乘 0 得 0，所以在进行转换时，可直接写出为 1 的位对应的权，再求和。上例可写成：

$$1011.101_{(2)} = 2^3 + 2^1 + 2^0 + 2^{-1} + 2^{-3} = 11.625_{(10)}.$$

（2）十进制数转换成二进制数

整数部分和小数部分需分别进行，然后再将转换结果写到一起。

对整数部分，所用的方法是除以 2 取余，例如：

2|11 余数
2|5 1……转换结果的小数点向左第一位
2|2 1……转换结果的小数点向左第二位
2|1 0……转换结果的小数点向左第三位
0 1……转换结果的小数点向左第四位，对该数为转换结果的最高位

于是：$11_{(10)} = 1011_{(2)}$。

对小数部分，所用的方法是乘 2 取整。例如将十进制小数 0.8125 转换为二进制：

0.8125→.625→.25→.5→.0 （每次乘积的小数部分）
↓×2 ↓×2 ↓×2 ↓×2
1 1 0 1 （每次乘积的整数部分）

第一次乘 2 的整数部分为转换结果小数点后第一位，第二次乘 2 得第二位，然后依此类推。所以，$0.8125_{(10)} = 0.1101_{(2)}$。

需要指出，有时无论怎样做下去，小数部分都不能为 0，这表明这个十进制小数不能准确地用二进制表示，这时只能近似表示(0 舍 1 入)。例如，按照上面的方法可得

$$0.8_{(10)} = 0.110011001100\cdots_{(2)}.$$

取到小数点后第三位，为 $0.110_{(2)}$；取到小数点后第四位，为 $0.1101_{(2)}$。

2. 十六进制

1）什么是十六进制

也可以用三点来描述：一是有 16 个数码，借用十进制数码 0 到 9，再借用英文字母 A、B、C、D、E、F 分别表示 10、11、12、13、14、15；二是计数时逢十六进一；三是采用位置表示法，但是，高一位的权是低一位的 16 倍。

2）十六进制数与十进制数的相互转换

（1）十六进制数转换成十进制数

按权展开，求和，例如：

$$21F_{(16)} = 2 \times 16^2 + 1 \times 16^1 + 15 \times 16^0 = 543_{(10)}.$$

（2）十进制数转换成十六进制数

与十进制数转换成二进制数相类似，整数部分和小数部分需分别进行，然后再将转换结果写到一起。对整数部分，所用的方法是除以 16 取余；对小数部分，所用的方法是乘以 16 取整。

3）十六进制数与二进制数的相互转换

(1) 十六进制数转换成二进制数

每一位十六进制数码用 4 位等值的二进制数表示即可。例如：

$$21F_{(16)} = 001000011111_{(2)}.$$

(2) 二进制数转换成十六进制数

以小数点为界，分别向左、向右，以四位为一组分组，每一组用一个等值的十六进制数码表示。注意，一组不够 4 位时，用 0 补足 4 位，特别对小数点后最右边一组必须补 0。例如：

$$100011.011\ \boxed{0}_{(2)} = 23.6_{(16)}(\boxed{0}\text{内的 0 是后补的}).$$

可以看出，十六进制数与二进制数之间的转换相当方便。因此，常用十六进制作为二进制数的缩写。例如，在汇编语言源程序和高级语言源程序中，常用十六进制表示内存地址或输入/输出设备的 I/O 端口地址。

在进行十进制数、十六进制数及二进制数相互转换时，经常要用到 $0 \sim 15_{(10)}$ 范围内三种进位制数的转换，其对应关系见表 1.1。表中的对应关系需要牢记。

表 1.1　$0 \sim 15_{(10)}$ 范围内三种进位制数的对应关系

十进制	十六进制	二进制	十进制	十六进制	二进制
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

3. 二进制编码的十进制数(BCD 码)

计算机内部用二进制，而人们习惯用十进制。为了方便二者之间的转换，引入了 BCD (Binary Coded Decimal)码，用二进制来为十进制数编码，即每一位十进制数码用 4 位二进制数表示。表示的方法较多，最常用的是每一位十进制数码用与其值相等的 4 位二进制数来表示。这种 BCD 码称为 8421 码。例如，十进制数 205 的 8421 码为 001000000101。

BCD 码和下面要介绍的十进制数码(字符'0' ~ '9')的 ASCII 码有简单的转换关系。借助于输入设备，可将一个十进制数用 ASCII 码的形式送入机器中，在计算机内部通过事先编

好的程序;先将该数的 ASCII 码转换成 BCD 码,再将 BCD 码转换成二进制数,这样计算机就可以对其进行运算或其他处理了。当要把内部的一个二进制数输出时,也是通过事先编好的程序,将该数先转换成 BCD 码,再转换成 ASCII 码,通过输出设备显示或打印出相应的字符。

1.1.2 字符编码

字符是指字母、数码、运算符号、标点符号等,当然汉字也属于字符。使用计算机,要涉及到字符。由于计算机只能识别 0 和 1 两种数码,所以字符也应采用二进制编码。目前常用的是 ASCII(American Standard Code for Information Interchange)码。

1. 基本 ASCII 码

用 7 位二进制数来给字符编码。7 位二进制数共有 $2^7 = 128$ 种不同组合,每一种组合可代表一种字符,即作为一种字符的编码。例如:

0110000 (48)* 字符'0';　　1000001 (65) 字符'A';
0110001 (49) 字符'1';　　1000010 (66) 字符'B';
0110010 (50): 字符'2'。

2. 扩充 ASCII 码

用 8 位二进制数来给字符编码。即在基本 ASCII 码前面增加一个二进制位,共 $2^8 = 256$ 种组合,可给 256 种字符编码。前 128 种,最高位为 0,仍用于表示基本 ASCII 字符。如 01000001 (65)仍表示字符'A'。后 128 种,最高位为 1,用于表示 128 种特殊符号,如制表符 └、┘、├、┬ 等。

3. 汉字编码

汉字编码涉及类型较多,这里仅介绍其中几种。

(1) 国标码

国标码的全称是国家标准化信息用汉字编码。国标汉字共 6 763 个。分为两级,一级汉字为常用汉字,共 3 755 个;二级汉字为非常用汉字,共 3 008 个。每个汉字对应 4 位十六进制数。如"大"的国标码为 $3\ 473_{(16)}$,写成二进制为 00110100 01110011。

(2) 输入码

输入码是指将汉字输入到计算机中所用的编码,有几十种之多,且还在不断研究新的输

* 括号中标注的是二进制数所对应的十进制数。

入编码。目前常用的有十几种，如汉语拼音、五笔字型、自然码、区位码等。中文 Windows 环境下的智能 ABC 输入法属拼音输入法，初学者使用起来很方便。区位码又称国标区位码，是国标码的一种变型。它将国标汉字分成 94 个区，每个区又分成 94 个位置，区码、位码分别用两位十进制数表示，在计算机内部用这两位十进制数的 BCD 码表示。如“大”在 20 区、83 位，其区位码为 2083，在机内表示为 00100000 10000011。

(3) 汉字内码

汉字内码是计算机系统内部处理、存储汉字所使用的统一代码。内码可由国标码变换而来，即将国标码的每个字节的最高位置 1，其他位均不变，即可得到内码。例如，已知“大”的国标码为 $3473_{(16)}$，写成二进制为 00110100 01110011，则“大”的内码为 10110100 11110011，写成十六进制为 B4F3。

(4) 字型点阵码

字型点阵码是显示或打印汉字时所用的编码。点阵中每一个位置对应一个二进制位：该位为 1，对应的位置有点；为 0，则对应的位置为空白。每 8 个二进制位组成一个点阵码字节。点阵的规模决定了点阵码的字节数。例如，采用 16 × 16 点阵，一个汉字的字型点阵码为 (16 × 16) ÷ 8 = 32 个字节。

1.1.3　带符号数的表示

上面提及的二进制数没有涉及符号问题，而实际上数有正、负之分。下面就此进行讨论。为叙述方便，先引进两个名词：机器数和真值。将一个数在机器中的表示形式，即编码称为机器数，将数本身称为真值。常用的机器数有三种：原码、补码和反码。

1. 原码

1) 通俗定义

将数的符号数码化，即用一个二进制位表示符号：对正数，该位取 0，对负数，该位取 1。而数值部分保持数的原有形式(有时需要在高位部分添几个 0)。这样所得结果为该数的原码表示。

例，$X = +1001010$，$Y = -1001010$，$Z = -1110(= -0001110)$。当原码为 8 位时，X、Y 和 Z 的原码分别是：

$$[X]_{原} = 01001010\ ;$$

$$[Y]_{原} = 11001010\ ;$$

$$[Z]_{原} = 10001110\ .$$

其中最高位为符号位。

2）正规定义

$$[X]_{原} = \begin{cases} X, & 0 \leqslant X < 2^{n-1}, \\ 2^{n-1} - X, & -2^{n-1} < X \leqslant 0. \end{cases}$$

其中，X 为真值，n 为原码的位数。这个定义实际是将真值的范围给出来了，当 $n=8$ 时，$-2^7 < X < 2^7$，因而，其数值部分写成二进制形式，最多为 7 位。从该定义可看出，X 为正数时，其原码还是数本身，第 8 位（符号位）补 0；X 为负数时，$-X$ 等于去掉负号，但要加上 2^7，即第 8 位为 $1(2^7 = 10000000_{(2)})$。因此，这个定义和上面的通俗定义是一致的。

3）原码表示的特点

原码表示有三个主要特点：一是直观，与真值转换很方便；二是进行乘、除运算方便；三是加、减运算比较麻烦。第一点是显然的。说原码表示进行乘、除运算方便是因为其数值部分保持了数据的原有形式，对数值部分进行乘或除就可得到积或商的数值部分，而积或商的符号位可由两个数原码的符号位进行逻辑运算而得到。说原码表示进行加、减运算比较麻烦，以加法为例，两个数相加需先判别符号位，若其不同，实际要做减法，这时需再判断绝对值的大小，用绝对值大的数减绝对值小的数，最后还要决定结果的符号位。

2. 补码

1）补码的引进和定义

据统计，在所有的运算中，加、减运算要占到 80% 以上，因此，能否方便地进行正、负数加、减运算，直接关系到计算机的运行效率。

如日常生活中的一个例子——指针式钟表。现在时针指向 11 点钟，要使其指向 6 点钟，有两种方法，一是正拨 7 个格，二是反拨 5 个格。如果把钟表看成一个计算器，正拨看成加运算，反拨看成减运算，那么，在钟表上有：$11-5=11+7$，即 $11+(-5)=11+7$。之所以这样，是因为 $11+7=12+6$，而在钟表上，12 相当于 0，超过 12 时，12 就丢失了。这种运算称为按模运算。钟表的模为 12。所谓“模”，是指一个系统的量程，或者说一个系统所能表示的最大的数（确切地说，为最大数时加 1）。按模运算是指运算结果超过模时，模丢失。当模为整数时，按模运算也可理解成除以模求余数的过程。常用符号“mod”表示按模运算。

在计算机中，一个具体数据类型的位数是确定的。例如，字节型数据为 8 位，当每一位都为 1 时，再加 1，最高位将产生进位。如果不采取措施，这个进位将被丢失，丢失的量为 $2^8=256$，这就是 8 位数据的模。从上面的例子已看到，按模运算，可以使正数加负数转化成正数加正数（$11-5=11+7$），一个负数可以等价于一个正数（-5 等价于 $+7$）。看一下计算机中情况。例如，要将 +0001111(15) 和 −0001100(−12) 相加，实际是要做减法，但不这样做，而是先将 −0001100 与模 100000000(256) 相加，得 11110100(−12+256=244)，再拿原被加数 0001111 和它(11110100)相加，得 00000011(15+244=256+3=3)，最高位的进位，即模丢失。