

21世纪高等学校信息类应用型系列教材

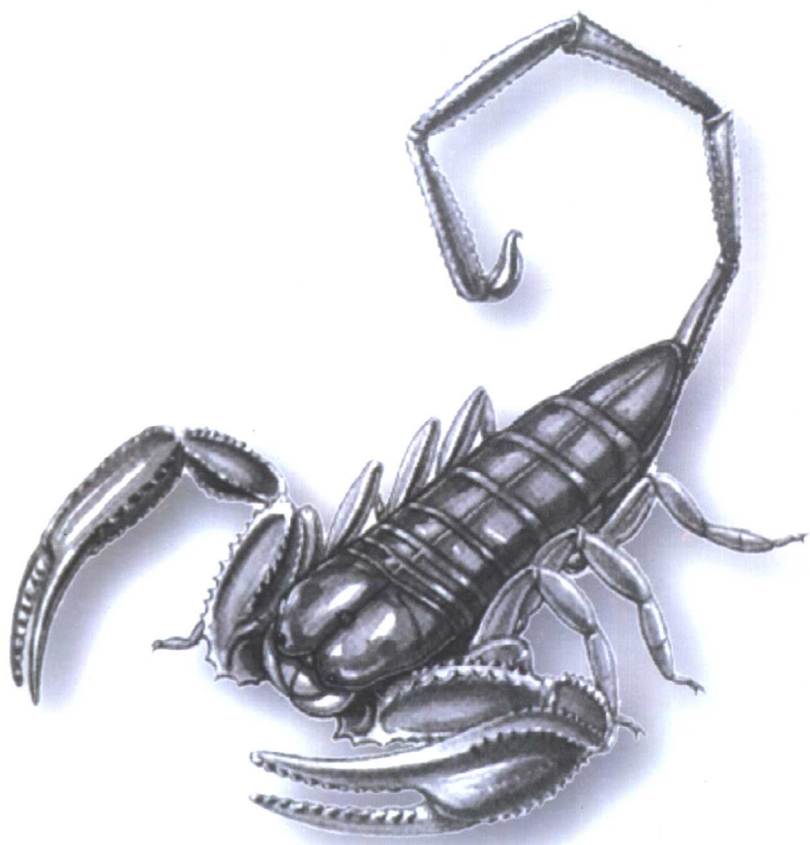


注 / 重 / 实 / 践

面 / 向 / 应 / 用

微型计算机原理 及接口技术

常通义 主编



HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS
E-mail: hustpp@wuhan.cngb.com

华中科技大学出版社

21 世纪高等学校信息类应用型系列教材

微型计算机原理及接口技术

主 编 常通义

副主编 孙立功

编 者 王 磊 程传洛 唐之义

主 审 普杰信

华中科技大学出版社

图书在版编目(CIP)数据

微型计算机原理及接口技术/常通义 主编
武汉：华中科技大学出版社，2003 年 11 月
ISBN 7-5609-3057-3

I. 微…
II. ①常… ②孙…
III. 微型计算机-接口技术
IV. TP364

微型计算机原理及接口技术

常通义 主编

责任编辑：万亚军
责任校对：封春英

封面设计：潘 群
责任监印：熊庆玉

出版发行：华中科技大学出版社 武昌喻家山 邮编：430074 电话：(027)87545012

录 排：华中科技大学惠友科技文印中心
印 刷：湖北省通山县印刷厂

开本：787×1092 1/16 印张：15 字数：318 000
版次：2003 年 11 月第 1 版 印次：2003 年 11 月第 1 次印刷 定价：22.50 元
ISBN 7-5609-3057-3/TP·516

(本书若有印装质量问题，请向出版社发行部调换)

内 容 简 介

本书是为了适应计算机技术发展的形势和非电类专业“微型计算机原理及接口技术”的教学需要而编写的。全书共分九章：计算机基础知识，8086 微处理器，存储器，8086 指令系统，输入/输出和中断，可编程接口芯片，汇编语言程序设计，A/D 与 D/A 转换，总线。各章均附有习题。

本书的主要读者对象是非电类专业的本科生，也可作为其他专业的本科生和工程技术人员的参考书以及自学用书。

前 言

电子计算机是信息技术的基础,是 20 世纪最卓越的成就之一。目前,计算机在科学计算、数据处理、过程控制、计算机辅助设计和制造(CAD / CAM)、人工智能方面的应用日益广泛深入,显示了旺盛的生命力和更为广阔的应用前景。本课程以 8086 为主讲机型,介绍微机的基本原理和应用问题。这里的“应用”主要不是指用于计算,而是指用于组成一个检测、控制等实时系统来解决科研生产中有关问题。一个计算机系统由硬件和软件两部分组成。系统中由计算机和其他实际电路元件等组成的组合体称为硬件。系统硬件的工作是由程序控制的,控制程序就是系统的软件。本课程的内容包括计算机的硬件设置和软件编制两部分内容。这些内容是计算机应用的基础。

本书是为了适应对非电专业学生进行 16 位微型机教学的需要,根据教研室多年的“微型计算机原理及应用”教学经验,在试用的基础上再总结提高编写而成的。编写本书的指导思想和本书的特点是:立足应用,注重基本原理、基本概念和基本方法,精选内容、突出重点、叙述扼要、深入浅出。

全书共分九章,第一、二、七章由常通义编写,第三、四章由孙立功编写,第五章由王磊编写,第六章和附录由程传洛编写,第八、九章由唐之义编写。全书由常通义担任主编,孙立功担任副主编,普杰信教授担任主审。田葳、刘珊中、许春媚对书稿进行了认真审阅并提出了许多宝贵的意见和建议,赵海霞、阎保定参加了试用教材的编写工作,在此一并表示衷心感谢。

由于编者水平所限,书中难免有错误和不妥之处,敬请使用本书的师生和读者批评指正。

编 者

2003 年 6 月于河南科技大学

目 录

第一章 计算机基础知识	(1)
第一节 数制	(1)
一、数制的基与权	(1)
二、数制之间的相互转换	(2)
第二节 计算机的逻辑运算和数值运算	(3)
一、逻辑运算	(3)
二、数值运算	(3)
第三节 信息交换码	(7)
一、ASCII 码	(7)
二、奇偶校验	(8)
第四节 微型计算机的组成及其中的信息传输	(9)
一、微型计算机的功能及组成	(9)
二、总线及微型计算机中的数据传输	(9)
第五节 微型计算机的基本逻辑部件	(11)
一、算术逻辑部件 (ALU)	(11)
二、寄存器	(11)
三、计数器	(11)
四、存储器和地址译码器	(13)
五、指令译码器	(15)
第六节 微型计算机的基本工作原理	(16)
一、简化微型计算机的组成	(16)
二、简化微型计算机的总线及其各部分的信息传输	(17)
三、简化微型计算机的指令系统	(18)
四、程序设计	(18)
五、运行程序——微型计算机的工作原理	(19)
六、控制矩阵	(20)
习题一	(21)
第二章 8086 微处理器	(22)
第一节 8086 微处理器的内部结构	(22)
一、执行部件 (EU)	(22)
二、总线接口部件 (BIU) 的组成和工作特点	(24)
三、存储器的分段和物理地址的形成	(25)
四、8086 内存的组织 and CPU 对存储器的访问	(26)

第二节 8086 微处理器 (CPU) 的引脚功能.....	(28)
一、地址线、数据线和状态线.....	(28)
二、控制线.....	(29)
三、其他.....	(30)
四、最大工作模式下重新定义的管脚意义.....	(30)
第三节 8086CPU 的工作模式.....	(31)
一、最小模式系统.....	(31)
二、最大模式系统*.....	(34)
第四节 8086 CPU 的基本总线控制时序.....	(37)
一、8086 最小方式下的总线读周期.....	(37)
二、8086 最小方式下的总线写周期.....	(38)
习题二.....	(39)

第三章 存储器..... (41)

第一节 概述.....	(41)
一、存储器的基本概念.....	(41)
二、存储器的性能指标.....	(41)
三、存储器的分类.....	(42)
第二节 半导体存储器的工作原理.....	(42)
一、半导体存储器的分类.....	(42)
二、半导体存储器的基本组成.....	(44)
三、半导体存储器的读/写操作.....	(46)
第三节 RAM 的结构及其常用芯片.....	(46)
一、RAM 的结构.....	(47)
二、常用 RAM 芯片.....	(48)
第四节 ROM 的结构及其常用芯片.....	(49)
一、ROM 的结构.....	(49)
二、常用 ROM 芯片.....	(50)
第五节 存储器与 CPU 的连接.....	(52)
一、只读存储器与 8086 CPU 的连接.....	(52)
二、静态 RAM 与 8086 CPU 的连接.....	(53)
三、EPROM、静态 RAM 与 8086 CPU 连接实例.....	(54)
第六节 存储体系.....	(56)
一、外存储器及其种类.....	(56)
二、存储体系的形成与发展.....	(57)
习题三.....	(58)

第四章 8086 指令系统..... (59)

第一节 汇编语言指令的格式与寻址方式.....	(59)
-------------------------	------

一、8086 汇编语言指令的格式	(59)
二、8086 的寻址方式	(60)
第二节 数据传输类指令	(62)
一、通用数据传输指令	(64)
二、目标地址传输指令	(66)
三、标志位传输指令	(66)
四、输入/输出数据传输指令	(67)
第三节 算术运算类指令	(67)
一、加法指令	(67)
二、减法指令	(69)
三、乘法指令	(70)
四、除法指令	(71)
五、调整指令	(71)
第四节 逻辑操作类指令	(72)
一、逻辑运算指令	(72)
二、移位指令	(74)
第五节 程序转移类指令	(76)
一、无条件转移指令	(76)
二、条件转移指令	(78)
三、循环控制指令	(79)
四、中断控制指令	(79)
第六节 字符串操作指令	(80)
一、字符串操作指令的特点	(80)
二、字符串操作指令	(81)
第七节 处理器控制类指令	(84)
一、标志位操作指令	(84)
二、外同步指令	(85)
习题四	(85)
第五章 输入/输出和中断	(87)
第一节 输入和输出	(87)
一、输入/输出概念	(87)
二、I/O 端口的寻址方式	(87)
三、8086 的 I/O 指令	(88)
四、I/O 数据传输的控制方式	(89)
第二节 中断	(92)
一、中断的基本概念	(92)
二、中断的基本类型	(92)
三、中断请求的提出和传输	(93)

四、中断优先级	(94)
五、中断响应	(95)
六、中断处理	(96)
第三节 8086 的中断系统	(97)
一、外部中断	(97)
二、内部中断	(98)
三、中断向量表	(99)
四、中断过程	(99)
第四节 可编程中断控制器 8259A	(100)
一、功能与结构	(100)
二、编程概述	(103)
习题五	(110)
第六章 可编程接口芯片	(111)
第一节 可编程并行接口芯片 8255A	(111)
一、8255A 的结构与功能	(111)
二、8255A 的工作方式和初始化编程	(113)
三、8255A 应用举例	(118)
第二节 串行通信接口芯片 8251A	(121)
一、串行通信概述	(121)
二、可编程通信接口 8251A	(124)
第三节 计数定时控制器 8253	(132)
一、基本结构和功能	(132)
二、8253 的工作方式	(134)
三、8253 计数/定时器应用编程	(138)
四、8253 计数/定时器应用举例	(138)
习题六	(140)
第七章 汇编语言程序设计的基本方法和技巧	(142)
第一节 概述	(142)
第二节 伪指令和宏指令	(143)
一、伪指令语句	(143)
二、宏指令语句	(147)
第三节 8086 汇编语言程序设计的基本语法	(149)
一、源程序语句的组成部分	(149)
二、标号和名字	(149)
三、助记符和定义符	(149)
四、操作数和参数	(150)
五、注释	(153)

第四节 汇编语言程序设计的基本方法和技巧	(153)
一、程序设计时要考虑的问题	(153)
二、汇编语言程序设计的步骤	(154)
三、汇编语言程序设计举例	(155)
第五节 分支程序和循环程序的设计	(157)
一、分支程序	(157)
二、循环程序	(159)
第六节 子程序的设计	(164)
一、子程序设计应注意的问题	(164)
二、子程序的调用和返回	(165)
三、子程序设计举例	(165)
第七节 IBM PC DOS 系统的功能调用	(168)
一、DOS 功能调用的方法及应注意的问题	(168)
二、常用 DOS 功能调用	(169)
三、DOS 功能调用应用举例	(171)
习题七	(175)
第八章 模/数 (A/D) 与数/模 (D/A) 转换	(177)
第一节 概述	(177)
第二节 数/模 (D/A) 转换器	(177)
一、数/模转换器的组成及工作原理	(177)
二、数/模转换的性能指标	(179)
三、D/A 转换器 DAC0832	(179)
四、DAC0832 应用举例	(182)
第三节 A/D 转换器	(183)
一、A/D 转换器的组成及工作原理	(183)
二、A/D 转换器的性能指标	(185)
三、A/D 转换器 ADC0809	(185)
四、12 位 A/D 转换器 AD574	(189)
习题八	(191)
第九章 总线	(193)
第一节 总线概述	(193)
一、总线和总线标准	(193)
二、总线的种类	(193)
三、信息在总线上的传输方式	(194)
四、总线仲裁	(194)
五、总线通信协议	(195)
第二节 PC 总线	(195)

- 一、PC 总线插槽及引脚信号..... (196)
 - 二、PC 总线信号说明..... (196)
 - 三、总线的负载能力..... (198)
- 第三节 几种系统总线的结构及功能..... (199)
 - 一、ISA 总线..... (199)
 - 二、EISA 总线..... (203)
 - 三、PCI 总线..... (205)
- 第四节 外部总线标准..... (208)
 - 一、IEEE-488 通用标准总线..... (208)
 - 二、RS-232C 总线..... (209)
- 习题九..... (212)
- 附录 指令系统*..... (213)
- 参考文献..... (227)

第一章 计算机基础知识

计算机的基本功能是进行数值运算、逻辑运算和数据处理。计算机是一个复杂的数字逻辑系统，由若干基本部件组成。本章首先介绍计算机进行数值运算时采用的数制和实现基本运算的电路，其次介绍总线的概念与信息传输原理，接着介绍组成计算机的基本逻辑部件，最后通过一个计算机模型介绍各逻辑部件之间的联系和基本工作原理。本章是本课程的基础。

第一节 数 制

数制是人们利用符号来计数的科学方法。日常生活和通常的科学计算中均采用十进制来计数，在计算机中则使用二、八、十六进制和二-十进制来计数。

一、数制的基与权

某种数制所使用的计数符号（数码）的个数叫做这种数制的基数，简称基；表示基本单位的数符“1”在某个数位上所表示的数值叫该数位的权。下面介绍上述各数制的基与权。

1. 十进制

十进制用 0、1、2、3、4、5、6、7、8、9 十个符号计数，它的基是 10。计数单位“1”在不同数位代表不同的数值，如图 1-1 所示。“1”在不同数位上所表示的数值有所不同，但都是 10 的整数指数幂，即各数位的位权都是 10 的整数指数幂，如个位的位权是 10^0 (1)，十位的位权是 10^1 (10)，十分位的位权是 10^{-1} (0.1) 等。

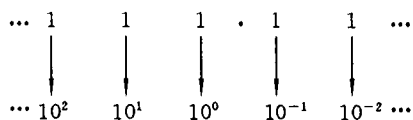


图 1-1 十进制的位权

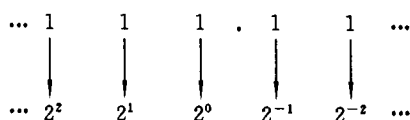


图 1-2 二进制的位权

2. 二进制

二进制采用“0”和“1”两个字符计数，其基为 2。与十进制相似，它的各数位位权也都是基（2）的整数指数幂。

如图 1-2 所示，小数点向左第一位的权是 2^0 (1)，第二位的权是 2^1 (2) ……小数点向右各位的权依次为 $2^{-1}(\frac{1}{2})$, $2^{-2}(\frac{1}{4})$ 等。

3. 八进制

它采用 0、1、2、3、4、5、6、7 八个字符计数，其基是 8，各位的权由小数点向左依次为 8^0 、 8^1 、 8^2 ……由小数点向右依次为 8^{-1} 、 8^{-2} ……

4. 十六进制

它采用 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个数符计数，其

基为 16。各位的位权从小数点往左依次为 16^1 、 16^2 、 16^3 ……小数点向右依次为 16^{-1} 、 16^{-2} ……

5. 二-十进制

这是一种用四位二进制数码 (BCD 码) 来表示十进制数字的计数制。它的本质是十进制,只是十个数符分别采用“0000”、“0001”、“0010”、“0011”、“0100”、“0101”、“0110”、“0111”、“1000”和“1001”十个四位二进制数表示。

二、数制之间的相互转换

二、十、十六进制可分别用字母 B、D、H 表示,如二进制数 1101 可写作 1101B,十进制数 43 可写作 43D(D 一般可省略),十六进制数 A2 可写作 A2H。

1. 将十进制数转换为二进制数

【例 1-1】将十进制整数 13 转换为二进制数。

解 将十进制整数转换为二进制数采用除 2 取余法,即将 13 除以 2 得商取出余数,再将商除以 2 得商取出余数,如此辗转相除直至商 0 取余为止,如图 1-3 所示。

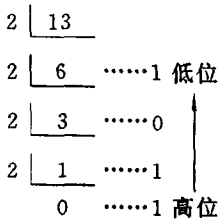


图 1-3 除 2 取余法

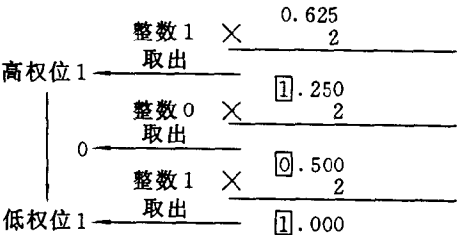


图 1-4 乘 2 取整法

然后将最后一个余数作高位,第一个余数为最低位依次排列即得所求的二进制数。由此可得:

$$13D=1101B$$

【例 1-2】将十进制小数 0.625 转化为二进制数。

解 将十进制小数转换为二进制小数采用乘 2 取整法,即将纯小数乘 2 取出积的整数,再将积的小数部分乘 2,取出积的整数……依次乘 2 取整直至积的小数部分为 0 (或得到适当的二进制的小数位数为) 为止。最后将最先取得的整数作高位,依次将所取乘积整数部分排列到小数点后就可以了,如图 1-4 所示。由此可得:

$$0.625D=0.101B$$

【例 1-3】将 13.625D 转换为二进制数。

解 将既有整数部分又有小数部分的十进制数转换成二进制数时,先用除 2 取余法转换整数部分,再用乘 2 取整法转换小数部分,最后将两部分拼合起来。这样可得:

$$13.625D=1101.101B$$

2. 将二进制数转换为十进制数

采用将二进制数按权展开成多项式和的形式再求和的方法,即将各位数字乘以所在位的权再相加的方法进行转换。

【例 1-4】将 101011B 转换为十进制数。

解 $101011B = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 = 43D$

3. 二进制数与十六进制数间的相互转换

以小数点为基准向左向右每四位划为一节，最后不足四位的补 0 凑成四位，然后写出各节二进制数值的十六进制码，即可将二进制数转换为十六进制数。如：

$$0011\ 1100.1110\ B = 3C.EH$$

反之，将一个十六进制数的各位数字用四位二进制数来表示就可把十六进制数转换为二进制数。如：

$$2A.5H = 0010\ 1010.0101\ B$$

第二节 计算机的逻辑运算和数值运算

一、逻辑运算

计算机可以对二进制数进行“与”、“或”、“非”、“异或”等逻辑运算。逻辑运算是按位进行的。

设 $A=10100111$ ， $B=11000001$ ，则有如下逻辑运算。

1. “与”运算

A 和 B 的“与”运算如图 1-5 所示，即：

$$Y = A \cdot B = 10000001$$

2. “或”运算

A 和 B 的“或”运算如图 1-6 所示，即：

$$Y = A + B = 11100111$$

3. “非”运算

$$Y = \overline{A} = 01011000$$

$$F = \overline{B} = 00111110$$

4. “异或”运算

A 和 B 的“异或”运算如图 1-7 所示，即：

$$Y = A \oplus B = 01100110$$

$$\begin{array}{r} A: 10100111 \\ \wedge) B: 11000001 \\ \hline 10000001 \end{array}$$

图 1-5 “与”运算

$$\begin{array}{r} A: 10100111 \\ \vee) B: 11000001 \\ \hline 11100111 \end{array}$$

图 1-6 “或”运算

$$\begin{array}{r} A: 10100111 \\ \oplus B: 11000001 \\ \hline 01100110 \end{array}$$

图 1-7 “异或”运算

二、数值运算

计算机进行数值运算时采用二进制。

1. 加法运算

二进制加法运算具有逢 2 进 1 的特点。两个一位二进制数加法运算的法则是： $0+0=0$ ， $0+1=1$ ， $1+0=1$ ， $1+1=10$ 。

(1) 半加器。

设 A、B 为两个一位二进制数，如用 S 表示它们相加的本位和，C 表示进位值，则可得 A、B 求和运算的真值表（见表 1-1）。由表 1-1 可得：

$$S = \overline{A}B + A\overline{B}, C = A \cdot B$$

表 1-1 半加器真值表

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

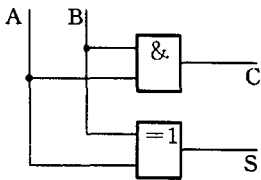


图 1-8 半加器

根据以上逻辑式可画出如图 1-8 所示的电路，这就是半加器。

(2) 全加器。

设 A、B 分别为 n 位二进制数， $A=A_{n-1}\cdots A_i\cdots A_0$ ， $B=B_{n-1}\cdots B_i\cdots B_0$ 。对 A、B 进行求和运算时，要从最低位开始按位相加，除最低位外，其余各位按位相加时必须考虑其相邻低位相加后向本位的进位，然后得出本位和和向高位的进位。例如，对第 i 位求和时，其低位相加向本位的进位为 C_i ，用 S_i 表示本位和， C_{i+1} 表示本位相加向高位的进位，可得出如表 1-2 所示的真值表。根据真值表可得 S_i 和 C_{i+1} 与 A_i 、 B_i 、 C_i 之间的逻辑关系式： $S_i=A_i \oplus B_i \oplus C_i$ ， $C_{i+1}=B_i \cdot C_i + A_i \cdot C_i + A_i \cdot B_i$ ，反映这种逻辑关系的电路如图 1-9 所示，这就是全加器。

表 1-2 全加器真值表

A_i	B_i	C_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

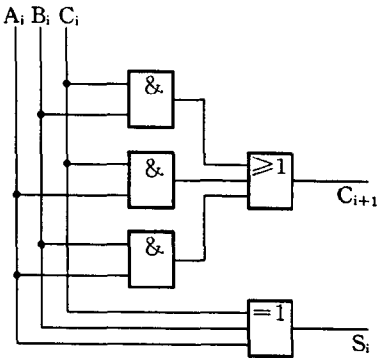


图 1-9 全加器

半加器和全加器的电路符号如图 1-10 所示。

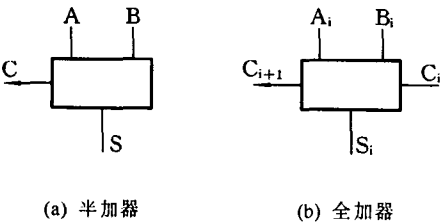


图 1-10 半加器和全加器的电路符号

图 1-11 所示的为一个由全加器和半加器组成的四位二进制加法电路。应该注意，若该加法电路的加数、被加数、和数都是四位，则向四权位的进位将自动丢失。

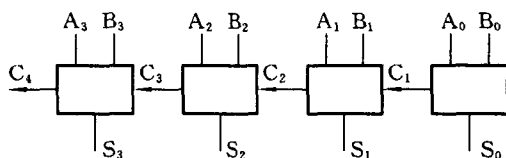


图 1-11 四位二进制加法电路

2. 减法运算——补码加法

数有三种表示形式：原码、补码、反码。如果数用补码来表示，则可将减法运算用加法来实现，即若用补码表示数，用加法器既可进行加法运算又可进行减法运算。为了简化运算电路，计算机中的二进制数都用补码来表示。

(1) 补码的概念。

一个计数系统（器）的状态个数叫它的模，如四位二进制计数器有 16 种状态，它的模就是 16 (2^4)，记作 mod16。它能表示的数的绝对值不能大于模。当计数器要表示的数的绝对值大于模时，大于模的整数倍部分将被丢失。

一个 n 位二进制无符号数（即数的绝对值用二进制数表示，数的正、负分别用“+”、“-”表示的数，如+1010，-1001 等）的计数系统的模是 $\text{mod } 2^n$ ，任意数 x 都可以用补码表示。无符号数补码的定义是：

$$[x]_{\text{补}} = \begin{cases} x, & 0 \leq x < 2^n; \\ \text{模} + x, & -2^n \leq x < 0. \end{cases}$$

如 mod16，则 +0110B 的补码是 0110B，-0110B 的补码是 1010B。

再如 +1100B 的补码为 1100B，-1100B 的补码为 0100B。

【例 1-5】求：1100-0110。

解 1100-0110=1100+(-0110)

$$\begin{aligned} [1100-0110]_{\text{补}} &= [1100]_{\text{补}} + [-0110]_{\text{补}} \\ &= 1100 + 1010 = 0110 \end{aligned}$$

总之，正数的补码就是它本身，负数的补码是将其各位按位求反再加“1”。减法运算的结果（补码）和被减数的补码与减数对应的负数的补码相加运算的结果（补码）相同。即：若 $x > 0$ ， $y > 0$ ，则 $z = x - y$ 的运算可用下式进行：

$$[z]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}}$$

(2) 机器数。

计算机中为了表示数的正负，让一个 n 位二进制数的最高位表示符号，叫符号位（符号位为“0”表示正，符号位为“1”表示负），其余各位叫数值位。有符号位的二进制数（简称有符号数）叫机器数。

① 机器数的原码。若机器数的数值位表示数的绝对值，则称为机器数的原码，简称原码。

例如：+8D 的八位二进制原码是 00001000B；

-8D 的八位二进制原码是 10001000B。

② 机器数的补码。若机器数的数值位是数的补码，则称之为机器数的补码，简称补码。

例如：+8D 的八位二进制补码是 00001000B；

-8D 的八位二进制补码是 11111000B。

(3) 补码的求法（已知原码求补码）。

已知机器数的原码要求补码时，原则上根据定义进行。

例如：八位机器数原码 01010111B 的补码是 01010111B。

八位机器数原码是 11010111B，它的补码是什么？

因该机器数原码的符号位是“1”，所以是一个负数，根据定义：

$$\begin{aligned}\text{负数的补码 } [x]_{\text{补}} &= \text{模} + x \\ &= 2^7 - |x| = 10000000 - 1010111 \\ &= (11111111 - 1010111) + 1 \\ &= 0101000 + 1 = 0101001\end{aligned}$$

故原码为 11010111B 的机器数的补码是 0101001B。由此可得出一般结论：将负机器数的原码符号位保持不变，数值位按位求反再加 1 就可得到该负数的补码。

(4) 已知补码求原码。

下面举例说明如何在已知补码的情况下求原码。

【例 1-6】已知八位二进制补码是 01011011B，求其原码。

解 上述二进制补码符号位是“0”，说明是一个正数。因正数的补码就是它本身，所以补码 01011011B 的原码是 01011011B。

【例 1-7】已知八位二进制补码 11011011B，求其原码。

解 设这个负数为 x，则根据补码的定义有：

$$\begin{aligned}[x]_{\text{补}} &= 2^7 + x = 2^7 - |x| \\ |x| &= 2^7 - [x]_{\text{补}} = 2^7 - 1011011 \\ &= (11111111 - 1011011) + 1 \\ &= 0100100 + 1 \\ &= 0100101\end{aligned}$$

所以 x 的原码是：

$$[x]_{\text{原}} = 10100101\text{B}$$

由此也可得出如下一般结论：使负数补码的符号位保持不变，将其数值位按位求反再加 1 就得到该负数的原码。

【例 1-8】求下列八位二进制补码所表示的数值：

- (1) 00000000B; (2) 01111111B;
(3) 11111111B; (4) 10000000B。

解

$$\begin{aligned}(1) [00000000\text{B}]_{\text{补}} &= +0000000\text{B} = 0\text{D} \\ (2) [01111111\text{B}]_{\text{补}} &= +1111111\text{B} = +127\text{D} \\ (3) [11111111\text{B}]_{\text{补}} &= -00000001\text{B} = -1\text{D} \\ (4) [10000000\text{B}]_{\text{补}} &= -10000000\text{B} = -128\text{D}\end{aligned}$$