高校工科非计算机专业

计算机应用知识与应用能力

蒋立源
胡正国 主编 张遵濂 主审

等级考试指南

西北工业大学出版社

高校工科非计算机专业

计算机应用知识与应用能力等级考试指南

蒋立源　胡正国　主编

蒋立源　胡正国
朱宏兴　高宏宾　吴　健　编

张遵濂　主　审

西北工业大学出版社

1994年1月　西安

(陕)新登字 009 号

【内容简介】 本书是为了适应近年来一些省市教委在普通高校非计算机专业中普及计算机知识，提高学生应用计算机的能力，推进非计算机专业学生计算机应用知识与应用能力等级考试制度的需要而编写的。全书共七章。第一章介绍计算机系统硬件的基础知识，包括电子数字计算机的发展简史、计算机的数制和转换、运算方法、计算机的组成及各主要部件的特性和功能等内容。第二至七章是有关计算机软件方面的知识，分别对 PASCAL 程序设计、算法概念、数据结构和编译系统、操作系统、数据库系统的组成及主要功能以及软件开发技术进行了简述。各章均附有一定数量的习题和作业。

本书由西北工业大学计算机系根据陕西省教委《非计算机专业学生计算机应用知识与应用能力等级考试大纲》编写的，可作为大专院校非计算机专业的"计算机应用(偏软)"课的参考教材，供大专学生参加等级考试使用，也可供科技人员和管理人员参考。

高校工科非计算机专业

计算机应用知识与应用能力等级考试指南

蒋立源　胡正国　主编

责任编辑　李　珂

责任校对　享　邑

*

(西安市友谊西路 127 号　邮编 710072)

全国各地新华书店经销

西北工业大学出版社印刷厂印装

ISBN 7-5612-0491-4/TP·62

*

开本 787×1092 毫米 1/16　14.5　印张　345 千字

1994 年 1 月第 1 版　　1994 年 1 月第 1 次印刷

印数：1—6 000 册　　定价：7.80 元

前　　言

计算机科学技术的出现是20世纪人类最伟大的科技成就之一。自80年代以来，计算机产业在我国得到了迅猛的发展，计算机的应用领域不断扩大，已产生了巨大的社会和经济效益。在此种形势下 向广大科技工作者和各种管理人员普及计算机应用方面的知识，提高应用计算机的水平，对促进各行业科技进步，加速各项事业的发展，无疑是一项十分重要的措施。特别是对我国高校为数众多的非计算机专业的在读学生来说，普遍开展计算机应用知识的教学，在拓宽专业面向，优化专业人材的知识和能力结构，以及更好地适应经济建设和社会发展等方面，将更加具有重大的意义。为此，近年来，许多省市的教育主管部门，都相继建立了高校非计算机专业学生计算机应用知识与应用能力等级考试制度。本书正是为了适应这方面教学工作的需要，根据陕西省教委所制订的《普通高校非计算机专业学生计算机应用知识与能力等级考试大纲》编写的。

全书共七章。第一章介绍计算机系统硬件的基础知识，包括电子数字计算机的发展简史、计算机的数制及其转换、运算方法、计算机的组成及各主要部件的特性和功能等内容。第二至第七章是计算机软件方面的知识，分别对PASCAL语言、算法概念、数据结构和编译系统、操作系统、数据库系统的组成及功能以及软件开发技术进行了简述。各章均附有一定数量的习题和作业，旨在使读者通过完成这些习题来加深对课堂教学内容的理解和增强使用计算机的能力。至于配合“等级考试”的练习题目，由于需要很大的篇幅，故不可能在本书中列出，我们拟在今后再专门编写有关的习题集和应试指导书。

作为一本计算机基础知识方面的入门教材，我们在编写过程中力图做到：既注意材料的精选，又使之具有较强的科学性和系统性；既注意讲清计算机系统软、硬件的结构和基本功能，而又不纠缠于它们的实现细节；既注意教学内容阐述的准确性，又尽量使基本概念的解释深入浅出、文字通俗易懂，以便于自学。

本书由西北工业大学计算机科学与工程系蒋立源和胡正国主编。参加编写工作的有：朱宏兴（第一章）、胡正国（第二章）、高宏宾（第三章）、蒋立源（第四、五章和第六章的1～3节）和吴健（第六章的第4节和第七章）等同志。

西北工业大学张遵濂教授对本书进行了仔细审阅，提出了许多宝贵的意见。在编写过程中，我们还得到了西北工业大学出版社许多同志的热情支持和帮助。在此我们一并表示衷心的感谢。

由于我们水平有限，加之编写时间仓促，本书不免有疏漏或不妥之处，恳请读者不吝指正。

编　者

1993年6月于西安

目 录

第一章 计算机硬件基础知识

本章主要介绍计算机发展简史、计算机的数制和码制、逻辑代数以及计算机的组成和计算机网络的基本概念。通过本章的学习，要求能进行十进制、二进制和十六进制数之间的转换，明确补码在计算机中的作用，能对逻辑代数进行简单的运算，了解计算机的基本结构与工作过程。

1.1 计算机发展简史

1.1.1 第一代电子计算机

世界上第一台可以由程序控制的计算机称为电子数字积分器与计算器(Electronic Numerical Integrator And Calculator)，简称ENIAC。它在1945年12月诞生，1946年2月在美国宾夕法尼亚大学正式交付使用，主要供美国军队计算弹道曲线之用。这台计算机的字长只有12位，运算速度为每秒5 000次加法或500次乘法或50次除法运算。它使用了18 800个电子管、70 000个电阻、1 000个电容、6 000个开关。它的体积为$30\times3\times1$ m^3、耗电150 kW，占地面积约为170 m^2，重达30t以上，真是一个庞然大物。但它毕竟是现代电子计算机的始祖，为当今的电子计算机奠定了基础。

1945年，与ENIAC问世的同时，冯·诺依曼在他的报告中提出了程序控制和程序存贮的新概念，而且用这个新概念设计了一台堪称现代计算机原理模型的通用电子计算机EDVAC。

第一代电子计算机是电子管数字计算机，其发展年代大约为1946至1958年。计算机的逻辑元件为电子管，主存贮器采用磁鼓、磁芯，外存贮器已开始使用磁带，主要用机器语言来编制程序，后期才逐步配置了汇编语言。当时主要用在科学计算方面。

1.1.2 第二代电子计算机

第二代是晶体管计算机，其发展年代为1958至1964年。计算机的逻辑元件为晶体管，主存贮器仍用磁芯，外存贮器已开始使用磁盘。在此期间，软件已开始有很大的发展，出现了一些通用的算法语言及编译程序，计算机的应用已发展至各种事务的数据处理，并开始用于工业控制。

1.1.3 第三代电子计算机

第三代是集成电路计算机，其发展年代为1964至1971年。计算机的逻辑元件为小规模和中规模的集成电路(SSI和MSI)，主存贮器仍以磁芯为主，外存贮器已开始使用小型磁盘(软磁盘)或盒式磁带机。小型计算机也随着集成电路规模增大而很快地发展起来。在程序设计技术方面形成了三个独立的系统：操作系统、编译系统和应用程序。

第三代计算机在计算机的发展史中处于很重要的地位。操作系统中的多道程序、分时系统

等概念的提出，结合计算机终端设备的广泛使用，使得用户可以在自己的办公室或家里使用远离自己的计算机。

1.1.4 第四代电子计算机

第四代是大规模集成电路计算机，它是大规模集成电路(LSI)迅猛发展的产物。由于在一块芯片上可集成上千万电子元件，因而可使电子计算机的体积大为缩小，这就导致了微型计算机的问世。因为微型计算机具有体积小、功耗低、重量轻、价格便宜、可靠性高、使用方便等一系列优点，故获得了广泛的应用和快速的发展。自微型计算机于1971年问世以来，大约每隔2～4年就更新换代一次，至今已经历了四个阶段的演变：

第一阶段(1971～1973年)为4位和低档8位微处理器及微型计算机(第一代微机)。美国Intel公司首先研制成功4位的4004微处理器，以它为核心再配以RAM，ROM和I/O接口芯片就构成了MCS－4微型计算机。随后又研制出8位的8008微处理器及MCS—8微型计算机，其特点是指令系统简单、运算功能较差、速度较慢，采用机器语言或简单的汇编语言编程，但价格低廉。

第二阶段(1973～1978年)为中档8位微处理器和微型计算机(第二代微机)。其间又分为两个阶段：1973～1975年为典型的第二代，以美国Intel公司的8080和Motorola公司的MC6800为代表，集成度提高1～2倍，运算速度提高一个数量级；1976～1978年为高档的8位微型计算机阶段，被称为第二代半微型计算机。代表产品是美国Zilog公司的Z80和Intel公司的8085微处理器。集成度和运算速度都比典型的第二代提高了1倍以上。

第三阶段(1978～1981年)是16位微处理器和微型计算机，又称为第一代超大规模集成电路(VLSI)的微处理器。代表产品是Intel公司的8086，Zilog公司的Z8000和Motorola公司的M68000。这些16位微型计算机都具有丰富的指令系统，并配有强有力的软件系统，各种电路功能大为增强。

第四阶段(1981年以后)为32位微处理器和微型计算机(第四代微机)。代表产品是Intel公司的80386，BELL研究所的MAC—32，NS公司的NS16032。

当前微处理器与微型计算机正朝着以下几个方向发展：发展高性能的16位和32位微处理器；发展带有软件固化的微型计算机；发展多微处理器系统和多个微型计算机所组成的微型机局域网络；发展功能更强的通用和专用接口芯片。

1.1.5 第五代电子计算机

尽管大规模、超大规模集成电路迅猛发展，各种各样的高性能的计算机不断出现，但它们还都属于第四代计算机。对下一代计算机主要特征的展望还没有一个完全、肯定的结论，但是比较普遍的看法是第五代计算机应以智能计算机为代表。各国都正在大力开展智能计算机方面的研究工作，预计21世纪的初期，将会取得光辉夺目的成果。

目前电子计算机的应用范围已非常广泛，而且应用领域还在不断迅速扩大。概括说来，其典型的应用有以下几个方面：科学计算、数据处理、自动控制和计算机辅助设计等，它正在渗透到各个部门，并深入到家庭日常生活之中。

1.2 计算机的数制和码制

1.2.1 二进制、十六进制和十进制数之间的转换

1.2.1.1 十进制数(Decimal)

人们最常用的数制是十进制,它由0～9等十个符号来表示数值,这十个符号就是数字。十进制是采用位置记数法的数制,即每个数字的位置决定了它的值或权。同样的数字在不同的位置代表的值或权是不一样的。如

$$555.5=5\times10^2+5\times10^1+5\times10^0+5\times10^{-1}$$

任何一个十进制数N都可表示为

$$(N)_{10}=D_{n-1}\times10^{n-1}+D_{n-2}\times10^{n-2}+\cdots+D_1\times10^1+D_0\times10^0+D_{-1}\times10^{-1}+\cdots+D_{-m}\times10^{-m}$$

$$=\sum_{i=n-1}^{-m}D_i\times10^i$$

其中,i表示数位,D_i表示第i位数码,可以是0～9等十个数码中的任一个,由数N来确定。m,n为正整数,分别表示小数和整数的位数。10为计数制的底数(基数)。

可见,一个十进制数的值,可以用按权展开的多项式来表示,每一个数位都对应一个基值(权)。一个十进制数的权,小数点左面的是十的正次幂,小数点右面的是十的负次幂。

位置记数法(带权记数法)的数制有以下几个主要特点:

(1) 数字的个数等于基数,最大的数字比基数小1。

(2) 每个数字都要乘以基数的幂次,而该幂次是由每个数所在的位置决定的。

(3) 低位向高位的进位是"逢基数进一"。

1.2.1.2 二进制数(Binary)

在计算机中,广泛使用二进制计数法。二进制数与十进制数类似,也有三个主要特点:

(1) 有两个不同的数字符号,即0和1;

(2) 逢二进一;

(3) 在数的表示中,每个数字都要乘以基数2的幂次,而此幂次是由该数字所在位置决定的。如

$$(101.1)_2=1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}$$

$$=(4+0+1+0.5)_{10}=(5.5)_{10}$$

二进制数有以下两个主要性质:

(1) 二进制数的小数点向右移一位,即整个数向左移一位,数值就增大1倍。反之,小数点向左移一位,即数向右移一位,数值就缩小一半,如

$$(101.1)_2=(5.5)_{10}$$

$$(1011.0)_2=(11)_{10}$$

$$(10.11)_2=(2.75)_{10}$$

(2) 对于二进制整数,若最低位是1,则此数为奇数,如0011;若最低位是0,则此数为偶数,如0110。

由于二进制数只含有两个数字0和1,因此,一个数的每一数位就可以方便地用具有两个

不同稳定状态的元件来表示。例如，氖灯的亮与灭，开关的闭合与断开，晶体管的截止与导通等。同时，由于每一位可由两个稳定状态的元件来表示，因此可以节约存贮设备。例如，要表示0～999这1 000个数，十进制须用三位数，共需3×10=30个稳定状态的设备量。而用二进制数表示时，则须用10位($2^{10}=1024$)，只需10×2=20个稳定状态的设备量。可见，用二进制表示数所需要的设备量比十进制的要少。

此外，二进制数的运算比较简单，如

加法运算：

$$0+0=0,$$
$$0+1=1+0=1,$$
$$1+1=0,\text{进位}1$$

减法运算：

$$0-0=0,$$
$$1-1=0,$$
$$1-0=1,$$
$$0-1=1,\text{有借位}$$

正因为二进制数只有两个数符，因此可以使用逻辑运算，这为计算机的设计提供了方便。

在日常生活中，经常遇到人们习以为常的数是十进制数，但计算机内又宜于按二进制进行运算，因此，在使用电子计算机时，就必须把十进制的原始数据换算成计算机所能接受的二进制数。计算机在运算结束后，又应把二进制数转换为人所习惯的十进制数。这两个换算的过程可以全由计算机自行完成。

1.2.1.3　十六进制数(Hexadecimal)

使用二进制数且当数值较大时，书写与阅读都很不方便且容易出错，记忆又困难。因此，通常用十六进制来作为二进制数的缩写。在微型计算机中，目前通用的字长为8位或16位，这恰巧可用2位或4位十六进制数来表示。因此，十六进制应用十分普遍，它已经成为微处理机产业的标准。

十六进制数也有三个主要特点：

(1) 有16个不同的数字符号，即0～9，A，B，C，D，E，F(其中，A，B，C，D，E，F分别代表十进制的10，11，12，13，14和15)；

(2) 逢十六进一；

(3) 在数的表示中，每个数字都要乘以基数16的幂次，而此幂次由该数字所在位置决定。如

$$(5AF.8)_{16}=5\times16^2+10\times16^1+15\times16^0+8\times16^{-1}$$
$$=(1280+160+15+0.5)=(1455.5)_{10}$$

1.2.1.4　进位数制之间的转换

1. 十进制整数转换为二进制数。十进制整数转换为二进制数采用“除2取余法”，即逐次用2去除要转换的十进制数，直至商为0，每次所得的余数即为二进制数码，最先得到的为整数的最低有效位K_0，最后得到的为整数的最高有效位K_{n-1}。如将25转换为二进制数：

2	25	
2	12	余数 $1=K_0$
2	6	余数 $0=K_1$
2	3	余数 $0=K_2$
2	1	余数 $1=K_3$
	0	余数 $1=K_4$

所以 $$(25)_{10}=(K_4K_3K_2K_1K_0)_2=(11001)_2$$

2. 十进制小数转换为二进制数。十进制小数转换为二进制数采用"乘 2 取整法"，即逐次用 2 去乘要转换的十进制小数，将每次所得的整数(0 和 1)，依次记作 K_{-1}，$K_{-2}\cdots$。若乘积的小数部分最后能为 0，则最后一次乘积的整数部分记作 K_{-m}，即为

$$(0.K_{-1}K_{-2}\cdots K_{-m})_2$$

但十进制小数并不是都能用有限位的二进制数精确地表示，因此不断用 2 去乘十进制小数时，就不一定都能使尾数部分等于 0，过程可能会无限制地进行下去。这时只要根据精度要求，转换到一定的位数为止。如将 0.625 转换为二进制数：

$0.625\times2=1.250$　　整数 $1=K_{-1}$

$0.25\times2=0.5$　　整数 $0=K_{-2}$

$0.5\times2=1.0$　　整数 $1=K_{-3}$

所以 $$(0.625)_{10}=(0.K_{-1}K_{-2}K_{-3})_2=(0.101)_2$$

对于一个具有整数和小数部分的十进制数，在转换为二进制数时，只要把它分为整数和小数两部分，然后再把它们分别转换为二进制数，最后用小数点把这两部分连起来即可。

3. 十六进制与二进制数之间的转换。因为 $2^4=16$，所以一位十六进制数相当于四位二进制数，它们是完全对应的。因此，十六进制与二进制数之间的转换十分方便。

(1) 十六进制转换为二进制数。由于每位十六进制数可以用四位二进制数来表示，故把十六进制数的每一位均用四位二进制数展开，便得到了相应的二进制数。例如，把十六进制数 $(8AC)_{16}$ 转换为二进制数可表示如下：

8AC

1000　1010　1100

$$(8AC)_{16}=(100010101100)_2$$

又如 $$(0.E5B)_{16}=(0.111001011011)_2$$

(2) 二进制转换为十六进制数。二进制整数转换为十六进制时，可以从最低位开始，每四位分为一组，不够四位的以 0 补足为四位，然后把每四位二进制数用相应的十六进制数表示即可。如二进制整数 $(10111011000)_2$ 可转换如下：

0101　1101　1000

5　D　8

所以 $$(10111011000)_2=(5D8)_{16}$$

二进制小数转换为十六进制时，从小数点右面第一位开始，每四位分为一组，最后不足四位的以0补足为四位，然后把每一组二进制数用相应的十六进制表示即可。如二进制小数 $(0.110111111)_2$ 可转换如下：

$$(0.110111111)_2=(0.110111111000)_2=(0.DF8)_{16}$$

1.2.2 二进制数表示

1.2.2.1 数的定点与浮点表示

一个十进制数 123.45 可以表示为

$$123.45=10^3\times 0.12345$$

同样，对二进制数，如 1010.01 可表示为

$$1010.01=2^{100}\times 0.101001$$

任意一个二进制数 N 可用通式表示为

$$N=2^j\times S$$

其中，j 是二进制整数的位数，称其为数 N 的阶码；S 是二进制小数，称其为数 N 的尾数。尾数 S 表示数 N 的全部有效数字，而阶码 j 指明了小数点的位置。

对任何一个数，若阶码 j 总是固定不变的，则把这种表示法称为数的定点表示，这样的数称为定点数，采用这种表示法的机器叫做定点计算机。如果阶码 j 可以取不同的值，则把这种表示称为数的浮点表示，这样的数称为浮点数，采用这种表示法的机器叫做浮点计算机。

通常，定点计算机的阶码取 j=0，故该定点数只能是小数。定点数的表示格式为

符号	.	数　　值

小数点的位置在尾数部分最高位之前，符号位之后，即小数点在符号位与尾数部分最高位之间。

按上述约定，在定点计算机中参与运算的数的绝对值必须小于 1。通常，在实际需要计算的数中，不一定都是小数，可以是整数，也可以既有整数又有小数部分。为了在定点计算机中用这些数进行运算，可以乘上一个比例因子，使其变成小数。例如，257.98 乘上一个 10^{-3}的比例因子，即成为 0.25798。当计算机输出后，再乘上比例因子 10^3，就恢复为原来的值。

比例因子要选择恰当，不能选得太小，否则计算机运算结果仍可能超过 1，即计算机出现了“溢出”，这将迫使计算机停机或进行“溢出处理”。若比例因子选得过大，小数点后好几位都是 0，虽然不会溢出，但计算结果的精度将受到很大影响。

在浮点计算机中，浮点数的表示由四部分组成：阶符、阶码、数符（尾数的符号）、数码（尾数）。其格式为

阶符	阶码	数符	.	数　　码

若定点计算机的字长为 9 位，其中 8 个二进制数码位，一个符号位，则该计算机所能表示的数值范围（绝对值）是

$$0.00000001\sim 0.11111111$$

或

$$2^{-8}\sim 1-2^{-8}$$

一般地，若计算机的字长是 n 位，其中一位符号位，(n－1)位数码位，其所能表示的数值范围（绝对值）是

$$2^{-(n-1)}\sim 1-2^{-(n-1)}$$

对于浮点计算机，总是规定尾数部分的最高位是 1，即

$$\frac{1}{2}\leqslant S<1$$

若浮点计算机的字长为13位，阶码为三位，阶符为一位，数码位为八位，数符为一位，则所能表示数的范围是

最小值：11110.10000000

最大值：01110.11111111

或

$$2^{-7}\times2^{-1}\sim2^{7}\times(1-2^{-8})$$

若阶码位是m位，数码位是n位，另外阶码和数码各有一位符号位，则所能表示数的范围为

$$2^{-(2^m-1)}\times2^{-1}\sim2^{(2^m-1)}\times(1-2^{-n})$$

应当注意，浮点数的正负号是由尾数的正负号决定的，而阶码的正负号只决定小数点的位置，即决定浮点数的绝对值大小。

因为浮点计算机中数的小数点是浮动的，所以一般的数就可以不用比例因子了。

浮点计算机通用性强，但结构复杂；定点计算机结构简单，但计算结果的精度受影响，且使用不太方便。

通常，微型计算机均为定点计算机(有的也可进行浮点运算)，而且又是整数，即小数点固定在数的末尾。因此，以下的讨论仅限于定点整数。

1.2.2.2 数的符号表示法

通常，约定一个数的最高位为符号位，故若字长为8位，则D_7为符号位，$D_6\sim D_0$为数字位。符号位用0表示正，用1表示负。如

$$x=(01010101)_2=+85$$

$$x=(11010101)_2=-85$$

这样，数的符号在机器中也就数码化了。把连同一个符号位在一起的一个数称为机器数；而把原来的实际数本身叫做机器数的真值。如定点机中的两个机器数分别为

01010101 和 11010101

则它们的真值分别为

+1010101 和 -1010101

1.2.2.3 原码、反码和补码

1. 原码。如上所述，正数的符号位用0表示，负数的符号位用1表示。这种表示法称为原码。

设 $x=x_1x_2\cdots x_{n-1}$

$$[x]_原=\begin{cases}0\quad x_1x_2\cdots x_{n-1} & 当\ x\geqslant0\\1\quad x_1x_2\cdots x_{n-1} & 当\ x\leqslant0\end{cases}$$

例如

若 $x=+85$， 则$[x]_原=01010101$

若 $x=-85$， 则$[x]_原=11010101$

8位二进制的原码有以下几个特点：

(1)“0”有两种表示法：

$$[+0]_原=00000000\qquad[-0]_原=10000000$$

(2) 8 位二进制的原码所能表示的数值范围为 +127～-127。

(3) 一个数的原码，数的数码部分不变，而仅仅由 0 和 1 分别来表示数的正负。

原码表示简单易懂，而且与真值转换方便。但是当两个异号数相加或两个同号数相减，就要做减法，致使机器结构复杂化或增加机器的运算时间。为了把减法运算转换为加法运算就引进了数的反码和补码表示。

2. 反码

$$[x]_{反}=\begin{cases}0 & x_1x_2\cdots x_{n-1} \quad 当\ x\geqslant 0\\ 1 & \overline{x}_1\overline{x}_2\cdots\overline{x}_{n-1} \quad 当\ x\leqslant 0\end{cases}$$

其中，

$$\overline{x}_i=\begin{cases}0 & 当\ x_i=1\\ 1 & 当\ x_i=0\end{cases}$$

正数的反码表示与原码相同，符号位为 0，其余位为数码位；而负数的反码表示，符号位为 1，其余位为数码位按位取反。例如

若 $x=+85$， 则 $[x]_{反}=01010101$

若 $x=-85$， 则 $[x]_{反}=10101010$

8 位二进制的反码有以下几个特点：

(1) "0"有两种表示法：

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

(2) 8 位二进制的反码所能表示的数值范围为 +127～-127。

(3) 当符号位为 0(正数)时，后面的七位为数码部分；而当符号位为 1(负数)时，必须把后七位按位取反，才表示它的二进制值。

3. 补码

$$[x]_{补}=\begin{cases}0 & x_1x_2\cdots x_{n-1} \quad 当\ x\geqslant 0\\ 1 & \overline{x}_1\overline{x}_2\cdots\overline{x}_{n-1}+1 \quad 当\ x<0\end{cases}$$

正数的补码表示与原码相同，符号位为 0，其余位为数码位；而负数的补码表示，符号位为 1，其余位为数码位按位取反后再加 1。例如

若 $x=+85$， 则 $[x]_{补}=01010101$

若 $x=-85$， 则 $[x]_{补}=10101010+1$

$=10101011$

8 位二进制补码有以下几个特点：

(1) 数 0 的补码只有一种形式：

$$[+0]_{补}=[-0]_{补}=00000000$$

(2) 8 位二进制的补码所能表示的数值范围为 +127～-128。

(3) 当符号位为 0(正数)时，后面的七位为数码部分；而当符号位为 1(负数)时，其余七位必须取反后再加 1 才是它的二进制值。

4. 补码运算。采用补码以后，可以把减法运算转化为求补相加运算，从而使电路结构简单，运算速度加快。

(1) 加法运算。由于 $[x+y]_{补}=[x]_{补}+[y]_{补}$，所以只要将 x 的补码和 y 的补码相加，而不必去考虑它们的正负号，就能得到两数和的补码。

设 $x_1=+0001010$ $\qquad x_2=-0001010$