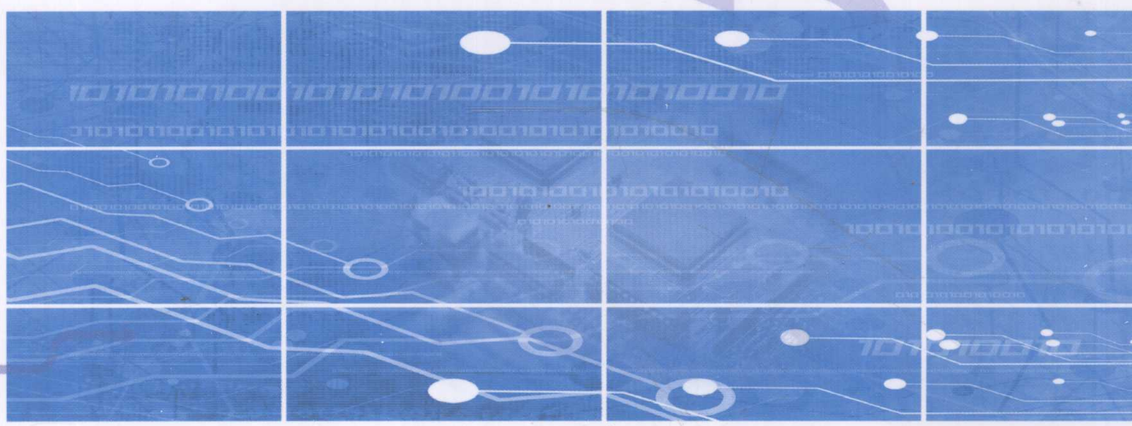
HZ BOOKS
华章教育

计算机应用技术规划教材



现代数字电路基础

黄健文 章鸣嬛 编著
孙德文 主审

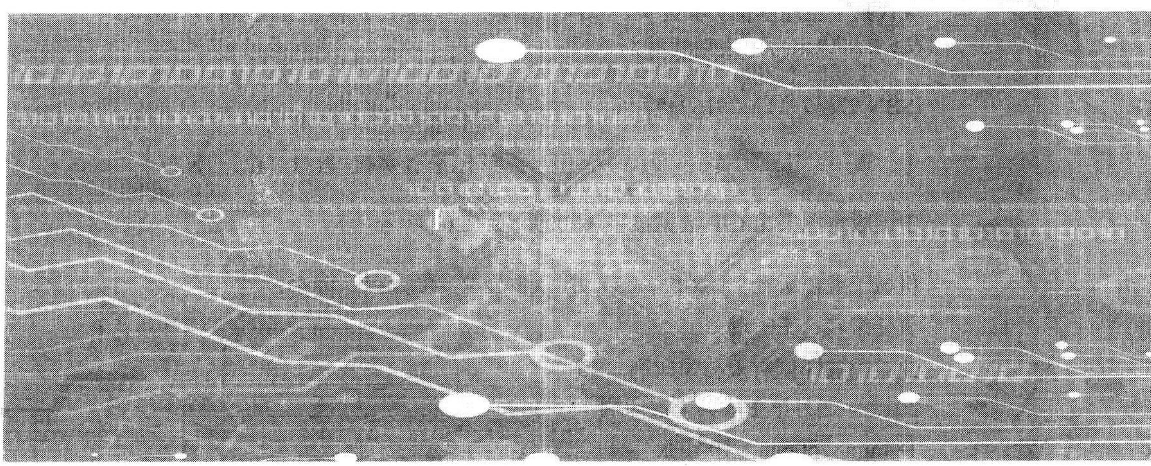
附光盘

机械工业出版社
China Machine Press

计算机应用技术规划教材

现代数字电路基础

黄健文 章鸣嬛 编著
孙德文 主审



机械工业出版社
China Machine Press

本书系统地介绍数字逻辑电路的基本概念、基本理论、基本分析方法；讲述常用数字逻辑部件的功能和应用。主要内容包括：数制和数码的概念、逻辑代数和逻辑门电路、组合逻辑电路、时序逻辑电路、半导体存储器和可编程逻辑器件、集成电路建模语言、计算机部件逻辑功能软件仿真。本教材以 CMOS 电路为基本单元，应用计算机辅助学习软件，帮助读者理解和掌握典型电路的逻辑功能计算机仿真技术。适合作为高等院校计算机相关专业数字电路课程的教材或参考书。

图书在版编目（CIP）数据

现代数字电路基础 / 黄建文，章鸣嬛编著. —北京：机械工业出版社，2010.2
（计算机应用技术规划教材）

ISBN 978-7-111-29169-5

Ⅰ. 现…　Ⅱ. ①黄…　②章…　Ⅲ. 数字电路-计算机-教材　Ⅳ. TN79

中国版本图书馆 CIP 数据核字（2009）第 220283 号

机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码　100037）
责任编辑：王　璐
北京瑞德印刷有限公司印刷
2010 年 2 月第 1 版第 1 次印刷
184mm×260mm • 10 印张
标准书号：ISBN 978-7-111-29169-5
　　　　　ISBN 978-7-89451-323-6（光盘）

定价：23.00 元（附光盘）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

序 言

It gives me a great pleasure to preface this new book written by my colleague and friend Professor Jianwen Huang, and co-written with her collegues Prof. Dewen Sun, lecturer Minghuan Zhang (from Shanghai Jiaotong University and Shanda University).This book is intended as an introductory digital circuits book for students in Computer Science, Computer Software Engineering, and Information. The philosophy of the authors is to illustrate the theory through a large number of design examples and problems, in order to give students both fundaments but also an intuitive feeling for how digital circuits operate.

The book is focused on CMOS based digital circuits, and has been written for students without specific previous knowledge of electronics. With the use of a companion software DSCH for logic design and simulation, the students will be able to illustrate both the basic circuits, either combinationnal and sequential, but also the fundamental blocks used in microprocessor circuits. Using DSCH, students can also simulate the behavior of their own circuits by modifying existing examples or exploring new design techniques, on their own computers.

Many books on digital logic design have restricted their contents to combinational and sequential circuits, with emphasis on logic theory and a lack of information about how this theory applies to build modern microprocessors. This book also introduces a very simple microprocessor, illustrated by some interesting examples, including schematic diagrams that may be executed, simulated and modified by the students, using the companion tool DSCH.

Throughout the book, basic concepts related to programmable logic devices (PLDs) are also introduced, the authors put an emphasis on HDL-based design, a very efficient design method used to design complex logic circuits. A set of examples and experimental works are introduced and explained.

I wish to congratulate the authors for having written this book, and try to innovate teaching methods in the unlimited field of logic circuits. The book gives a comprehensive overview of modern logic design fundaments, it will be necessary to students in Computer and Information fields, giving them the in-depth understanding of logic design concepts and also the computer aided instruction learning ways.

Professor Etienne SICARD
Author of Dsch and Microwind tools
INSA Toulouse, France
Etienne.sicard@insa-toulouse.fr

能为本书写序言我感到十分荣幸。本书由我的朋友兼同事黄建文教授与孙德文、章鸣嬛老师编著。这是一本适合于计算机科学、软件工程及信息技术等专业学生使用的数字电路入门教材。作者的出发点是通过大量设计实例和问题解答来阐明数字电路理论。期望学生们在得到基础知识的同时，对电路的运行也有直观的了解。

本书集中讲述以 CMOS 为基础的数字电路，读者无需事先具有专业的电子学知识。利用与教材配套的逻辑设计和仿真软件 DSCH，帮助读者理解组合电路、时序电路及微处理器电路基础模块。应用 DSCH，学生们可以在自己的计算机上修改已提供的电路实例或探索新的设计技术，并对其性能进行仿真。

很多数字逻辑电路教材，将它们的内容限制在组合电路和时序电路的理论分析，缺少这些基础电路在构成现代微处理器方面的讲解。本书通过一些有趣的例子介绍了一个十分简单的微处理器电路，该电路可以在 DSCH 环境下运行、仿真过程也可以被学生修改。

本书对于可编程逻辑器件（PLD）的基本概念也有所介绍。作者强调基于 HDL、硬件描述语言的设计技术，它们对于复杂逻辑电路的设计是十分有效的。一些例子和实验工作将在教材中得到解释。

祝贺作者完成了这本书。当今数字电路领域的发展是永无休止的，他们尝试着对于该领域的教学方法有所改进和创新。本教材对近代逻辑设计基础作了全面概述，这些知识对于计算机和信息领域的学生是必要的，帮助他们较为深入地明白逻辑设计的概念及利用计算机进行辅助学习的方法。

DSCH 及 MICROWIND 软件作者：（法）西加教授
INSA Toulouse, France
Etienne.sicard@insa-toulouse.fr

前　言

微电子技术和计算机技术是当今发展最快的技术领域，它们的应用已经无所不在。本书根据“教育部高等学校计算机科学与技术教学指导委员会（2006-2010）” 所提出的专业方向意见，定位于计算机科学与技术本科。本书特点之一是试图在数字电路教程中淡化一些传统的电路组成和分析，强调 CMOS 电路应用。CMOS 数字集成电路由于具有低功耗、较强的抗干扰能力及电路结构简单等特点，使它们成为 21 世纪数字集成电路的主角。

本书的另一个特点是充分利用计算机辅助学习软件，各章的学习重点和难点均通过计算机仿真，帮助读者较快理解典型电路的逻辑功能，引导读者建立起逻辑变量间动态时序关系的概念，对于学习后续课程（例如，计算机组成原理、微机技术和计算机系统结构等）是十分必要的。本书各章均附有思考-实践题和习题，鼓励读者在计算机上进行数字电路逻辑功能仿真或验证设计结果的正确性。

本书第一部分（第 1～5 章）介绍数制与编码、逻辑代数基础、CMOS 电路基础、组合逻辑电路和时序逻辑电路的分析与设计。这些知识是理解数字电路（系统）的基础，也是进一步学习大规模集成电路的基础。书中典型电路和逻辑功能模块的构成，大多以 CMOS 反相器和传输门为底层器件，与传统的 TTL 电路相比，其结构简单，便于理解，具有更好的外特性，适合于大规模集成。

本书第二部分（第 6～8 章）介绍半导体存储器和可编程逻辑器件、硬件描述语言 VHDL、数字电路应用。将前面介绍的数字电路传统分析方法（真值表、状态表、状态表图等），过渡到应用高级语言的建模和仿真分析方法。帮助具有数字电路和计算机语言基础的读者通过学习 VHDL 描述模型，了解大规模集成电路设计软件化的发展趋势。第 8 章以一个十分简单的 4 位微处理器模型的分析和部分功能仿真，作为学习数字电路的总结，也是学习后续计算机课程的基础。

本书附录所介绍的 DSCH 及 MICROWIND 软件，是在 PC 机平台上运行的 CMOS 电路设计和仿真软件，其教育版可以从有关网站免费下载。本书的配套光盘提供了例题库及中外学生的习作供读者学习参考。

本书适合软件工程或信息技术方向的本科生使用，是学习数字电路基础的教材或教学参考书，也可供有关工程技术人员参考。

在本书的编写过程中，非常感谢法国应用科学学院[㊀]E.SICARD 教授提供的教学软件支持及 CMOS 集成电路设计文档示例，并热情地给本书写了序言。同时，上海交通大学软件学院和上海杉达学院计算机学院在本书的编写过程中给予了大力支持，在此一并表示感谢！
由于作者水平有限，书中难免存在错误与疏漏之处，恳请读者批评、指正。

编者

2009 年 10 月

㊀ 法国应用科学学院的全称是 Institut National des Sciences Appliquées，简称 INSA-Toulouse。——作者注

目　录

第一部分

数字电路（系统）基础及计算机功能仿真

第 1 章　数制与编码

学习要点

（1）熟悉十进制数、二进制数、八进制数、十六进制数以及不同数制间的转换。
（2）熟悉带符号机器数的表示方法——原码、反码、补码。
（3）熟悉带符号机器数补码运算。
（4）了解十进制数的 8421BCD 编码。

数制（Number System）是一种表示数值的方法，常用的计数制是进位计数制。我们最熟悉的十进制计数制就是一种进位计数制，但它在数字系统中实现起来较为困难。在数字电路中最常使用的是二进制计数制，它是目前计算机硬件唯一能够识别的数制。但二进制书写起来太长，因此又引入了八进制与十六进制。本章首先学习数制及其相互转换，然后了解计算机中机器数的表示，最后介绍编码，了解应用二进制代码表示其他类型信息的方法。

1.1　二进制数、十六进制数和八进制数

当用数字量表示物理量大小时，仅用一位数码往往不够，必须用进位计数的方法组成多位数码，即利用一串数码表示一个数。在多位数码中，必须掌握每一位对应的权值（Weight），以及低位向高位的进位规则。

常用的数制包括十进制、二进制、八进制和十六进制。

首先以十进制为例说明。十进制由 10 个基本数码 0、1、2、…、9 组成，基数为 10；相邻位数的进位规则是“逢十进一”。可在末尾加上 D 或下标 10 以区别于其他数制。

例 1.1　展开十进制数。

$$234.709_{10}=2\times10^2+3\times10^1+4\times10^0+7\times10^{-1}+0\times10^{-2}+9\times10^{-3}$$

由上例可得到十进制数的展开规律为：

$$D=\sum_{i=m-1}^{-n}K_i10^i$$

式中：K_i 为第 i 位的系数，可以取 0～9 中的任意一个，对应的权值为 10^i。

假设该十进制的整数部分为 m 位，小数部分为 n 位，则 i 的取值范围是从（$m-1$）～0 的所有正整数，及从（-1）～n 的所有负整数。

在例 1.1 中，十进制数 234.709，m=3，i 取值（2～0）；n=3，i 取值（-1～-3）。

若对 N 进制数（N 表示任意进制）进行展开，其普遍形式为：

$$D=\sum_{i=m-1}^{-n}K_iN^i \tag{1-1}$$

式 1-1 中，K_i 为第 i 位的系数，对应的权值为 N^i。i 取值范围与上述十进制的规律相

同，详见后续 1.1.1、1.1.2、1.1.3 节。

1.1.1 二进制数

二进制由两个基本数码 0、1 组成，基数为 2。相邻位数的进位规则是“逢二进一”。通常在末尾加上字母 B 或用下标 2 来区别于其他数制。

在式 1-1 中，取 N=2，得到二进制数的展开的普遍形式：

$$D=\sum_{i=m-1}^{-n} K_i 2^i \tag{1-2}$$

式 1-2 中 K_i 为第 i 位的系数，可以取 0、1 中的任意一个数；第 i 位对应权值 2^i，i 取值范围与十进制的规律相同。

二进制数 10110011，应表示成 10110011_2 或 10110011B。

例 1.2 展开二进制数。

$$\begin{aligned}11001.001_2&=1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+0\times2^{-2}+1\times2^{-3}\\&=1\times2^4+1\times2^3+1\times2^0+1\times2^{-3}\end{aligned}$$

计算机中的数据均采用二进制数表示，这是因为二进制数具有以下优点：

1）二进制数只有两个数码 0、1，易于表示，可用电路所具有的两个不同的稳定状态来表示。例如，电路中的开关是闭合还是断开：若闭合表示“1”，断开表示“0”；电路中有无电流：若有电流表示“1”，无电流表示“0”；晶体管是导通还是截止：若导通表示“1”，截止表示“0”等。

2）二进制数运算简单，大大简化了计算机中运算部件的结构。两个 1 位二进制数的运算规则如下：

加法规则：0+0=0、0+1=1、1+0=1、1+1=10（1 为向高位的进位）

乘法规则：$0\times0=0$、$0\times1=0$、$1\times0=0$、$1\times1=1$

减法规则：0−0=0、0−1=1（向高位借 1）、1−0=1、1−1=0

1.1.2 十六进制数

由于二进制数的基数较小，与数值相等的十进制数相比，它的位数较多，书写和阅读均不方便，为此引入十六进制与八进制数。十六进制由 16 个基本数码组成：0～9、A（10）、B（11）、C（12）、D（13）、E（14）、F（15），基数为 16；第 i 位对应权值 16^i。相邻位数的进位规则是“逢十六进一”。通常在末尾加上字母 H 或用下标 16 以与其他数制加以区别。由于十六进制的每位数码正好对应 4 位二进制数，所以十六进制数可以较简洁地表达位数较多的二进制数值（见 1.2.3 节）。

例 1.3 展开十六进制数。

$$A8.CB4_{16}=A\times16^1+8\times16^0+C\times16^{-1}+B\times16^{-2}+4\times16^{-3}$$

$$1E8.F21_{16}=1\times16^2+E\times16^1+8\times16^0+F\times16^{-1}+2\times16^{-2}+1\times16^{-3}$$

1.1.3 八进制数

八进制数由 8 个基本数码 0～7 组成，基数为 8；第 i 位对应权值 8^i。相邻位数的进位规则是“逢八进一”。通常在末尾加上字母 O 或用下标 8 以与其他数制加以区别。由于八进制的每位数码正好对应 3 位二进制数，所以八进制数也能较简洁地表达位数较多的二进制数值

（见 1.2.3 节）。

例 1.4 展开八进制数。

$$21.7_8=2\times8^1+1\times8^0+7\times8^{-1}$$

1.2 不同数制间的转换

1.2.1 二进制数、八进制数及十六进制数转换成十进制数

1. 二进制数转换成十进制数

将二进制转换成等值的十进制数时，可将二进制数按照式 1-2 展开，再将所有项的数值按照十进制的进位规则相加即可。

例 1.5 将二进制数 11001.001_2 转换成十进制数。

$$\begin{aligned}11001.001_2&=1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+0\times2^{-2}+1\times2^{-3}\\&=16+8+0+0+1+0+0+0.125=25.125_{10}\end{aligned}$$

2. 八进制数转换成十进制数

将八进制转换成等值的十进制数时，可将八进制数按照式 1-1，取 N=8 展开，再将所有项的数值按照十进制的进位规则相加即可。

例 1.6 将八进制数 21.7_8 转换成十进制数。

$$21.7_8=2\times8^1+1\times8^0+7\times8^{-1}=16+1+0.875=17.875_{10}$$

3. 十六进制数转换成十进制数

将十六进制数转换成等值的十进制数时，可将十六进制数按照式（1-1）（取 N=16）展开，再将所有项的数值按照十进制数的进位规则相加即可。

例 1.7 将十六进制数 $2A8.F_{16}$ 转换成十进制数。

$$\begin{aligned}2A8.F_{16}&=2\times16^2+A\times16^1+8\times16^0+F\times16^{-1}=2\times16^2+10\times16^1+8\times16^0+15\times16^{-1}\\&=512+160+8+0.9375=680.9375_{10}\end{aligned}$$

1.2.2 十进制数转换成二进制数

将十进制数转换成等值的二进制数时，要分成整数部分和小数部分分别进行转换，再将整数部分和小数部分合并在一起。

1. 十进制数的整数部分转换成二进制数

十进制数的整数部分的转换采用基数除法。即用二进制的基数 2 除十进制整数，第一次所得余数为二进制数的整数最低位；将得到的商再除以基数 2，所得的余数为二进制整数的次低位，以此类推，直到商为 0，此时所得的余数为二进制整数的最高位。

例 1.8 将 20_{10} 转换成二进制数。

$$\begin{array}{r|l l}
2 & 20 & \\
2 & 10 & \cdots\cdots 0 \\
2 & 5 & \cdots\cdots 0 \\
2 & 2 & \cdots\cdots 1 \\
2 & 1 & \cdots\cdots 0 \\
 & 0 & \cdots\cdots 1
\end{array}\uparrow$$

可得 $20_{10}=10100_2$

对于 10 位以下的二进制数，推荐使用“试凑”的方法。

例 1.9 将 513_{10} 转换成二进制数。

在利用“试凑”的方法时，首先判断要转换的数大致在哪个范围，513_{10} 在 512_{10}（$=2^9$）与 1024_{10}（$=2^{10}$）之间。而 512_{10} 转换成二进制数应是 $10\ 0000\ 0000_2$，因此 $513_{10}=10\ 0000\ 0001_2$。

2．十进制数的小数部分转换成二进制数

十进制数的小数部分的转换采用基数乘法。即用小数乘以二进制的基数 2。第一次所得结果的整数部分为二进制数的小数最高位；将剩下的小数部分再乘以 2，所得结果的整数部分为二进制数的小数次高位，以此类推，直到小数部分为 0 或者达到要求的精度为止。

例 1.10 将 0.625_{10} 转换成二进制数。

$$
\begin{array}{rrl}
 & 0.625 & \\
\times & 2 & \\
\hline
 & 1.25 & \cdots\cdots 1 \\
 & 0.25 & \\
\times & 2 & \\
\hline
 & 0.5 & \cdots\cdots 0 \\
\times & 2 & \\
\hline
 & 1.0 & \cdots\cdots 1
\end{array}\Bigg\downarrow
$$

可得 $0.625_{10}=0.101_2$

综合例 1.9 和例 1.10，可得 $20.625_{10}=10100.101_2$

1.2.3　二进制数、八进制数及十六进制数的相互转换

二进制、八进制及十六进制的转换关系，如表 1-1 所示。

表 1-1　二进制、八进制及十六进制的转换关系

二进制	八进制	十六进制	二进制	八进制	十六进制
0000	0	0	1000	10	8
0001	1	1	1001	11	9
0010	2	2	1010	12	A
0011	3	3	1011	13	B
0100	4	4	1100	14	C
0101	5	5	1101	15	D
0110	6	6	1110	16	E
0111	7	7	1111	17	F

在表 1-1 中，4 位二进制数对应 1 位十六进制数，3 位二进制数对应 1 位八进制数。

1．二进制数转换成八进制数

由于 3 位二进制数正好能够表示 1 位八进制数，因此当二进制数转换成八进制数时，只要将二进制数的整数部分从右到左每 3 位分成一组，最后不足 3 位左边用 0 补足；小数部分从左到右每 3 位分成一组，最后不足 3 位右边用 0 补足，再根据表 1-1 将每 3 位二进制数对

应的八进制数码写出即可。

例 1.11 将 10101111011.0110101_2 转换为八进制数。

$$10101111011.0110101_2=010\ 101\ 111\ 011.011\ 010\ 100_2=2573.324_8$$

2．二进制数转换成十六进制数

由于 4 位二进制数正好能够表示 1 位十六进制数，因此当二进制数转换成十六进制数时，只要将二进制数的整数部分从右到左每 4 位分成一组，最后不足 4 位左边用 0 补足；小数部分从左到右每 4 位分成一组，最后不足 4 位右边用 0 补足，再将每 4 位二进制数对应的十六进制数码写出即可。

例 1.12 将 10101111011.0110101_2 转换成十六进制数。

$$10101111011.0110101_2=0101\ 0111\ 1011.0110\ 1010_2=57B.6A_{16}$$

3．八进制数转换成二进制数

将八进制数转换成二进制数，只需将每位八进制数转换成相应的 3 位二进制数。

例 1.13 将 324.76_8 转换成二进制数。

$$324.76_8=011\ 010\ 100.111\ 110_2$$

4．十六进制数转换成二进制数

将十六进制数转换成二进制数，只要将每位十六进制数转换成相应的 4 位二进制数。

例 1.14 将 $AC3.8E_{16}$ 转换成二进制数。

$$AC3.8E_{16}=1010\ 1100\ 0011.1000\ 1110_2$$

1.3 有符号二进制数

计算机中的二进制数分为无符号数（默认为正数）和有符号数。有符号数由符号位和数值位两部分组成。符号位置于数值的最高有效位（MSB）之前，用“0”表示正数，用“1”表示负数。

例 1.15 有符号二进制数的表示。

设 X 表示“真值”，$X_1=+85_{10}=+1010101_2$，$X_2=-85_{10}=-1010101_2$。

真值在机器中的表现形式是：

$[X_1]=\mathbf{0}\ 1010101_2$（最高位 **0** 为符号位）

$[X_2]=\mathbf{1}\ 1010101_2$（最高位 **1** 为符号位）

这里的 $\mathbf{0}\ 1010101_2$ 和 $\mathbf{1}\ 1010101_2$ 称为“机器数原码”，简称原码。

1.3.1 原码、反码、补码

带符号位的机器数主要有 3 种表示法，分别是原码、反码和补码表示法。

1．原码表示法

机器数$[X]$的原码$[X]_原$表示法如例 1.15 所述，用二进制数“0”或“1”分别表示真值中的符号“+”或“−”。

例 1.16 已知 $X_1=-18_{10}$，$X_2=-0.25_{10}$，设机器数字长 n=8，求 X_1、X_2 的 8 位二进制原码表示形式。

真值 $X_1=-18_{10}=-0010010_2$，$X_2=-0.25_{10}=-0.0100000_2$

机器数原码 $[X_1]_原=\mathbf{1}\ 0010010_2$

$[X_2]_{原}=\mathbf{1}.0100000_2$

（注意，根据本例要求，原码含符号位和数值位共计 8 位。）

例 1.17 将“0”进行 8 位原码表示。

$X_1=+0000000 \quad [X_1]_{原}=\mathbf{0}\ 0000000$

$X_2=-0000000 \quad [X_2]_{原}=\mathbf{1}\ 0000000$

可以看出，0 的原码表示有两种，即有重码。

2．反码表示法

（1）正数的反码表示

符号位为 0，除符号位之外，数值位表示形式与原码相同。

（2）负数的反码表示

符号位为 1，除符号位之外，将数值位按位取反。

例 1.18 已知 $X_1 = -18_{10}$，$X_2= -0.25_{10}$，设机器数字长 n=8，求 X_1、X_2 的反码表示形式。

$X_1= -18_{10}= -0010010_2$，$[X_1]_{原}=\mathbf{1}\ 0010010_2$

$X_2= -0.25_{10}= -0.0100000_2$，$[X_2]_{原}=\mathbf{1}.0100000_2$

$[X_1]_{反}=\mathbf{1}\ 1101101_2$

$[X_2]_{反}=\mathbf{1}.1011111_2$

“0”的反码表示。

$X_1=+0000000，\quad [X_1]_{反}=\mathbf{0}\ 0000000$

$X_2= -0000000，\quad [X_2]_{反}=\mathbf{1}\ 1111111$

可以看出，0 的反码表示也有两种，即也有重码。

3．补码表示法

以时钟对时为例，初步建立补码的概念。

假设当前时间为 7 点整，但时钟指示时间为 9 点整。可以通过两种方法进行校正：

1）将时针反时钟方向拨 2 小时（9–7=2）；或

2）将时针顺时钟方向拨 10 小时（12–2=10）。

这两种方法都能够对准到 7 点整。在这一命题中，对于基数 12，减 2 与加 10 是等价的，于是，称（–2）是（+10）的补码。可表示为：（–2）= +10（mod 12）。

在计算机运算器系统中，往往要求得到原码对于基数 2 的补码（Two's Complement），以便将减法转为加法运算。

对于有符号二进制数求补码的规则如下：

1）符号位为 0 表示正数，其补码表示形式与原码相同。

2）符号位为 1 表示负数，求补码时，除符号位之外，将原码数值位按位取反后再加 1。

例 1.19 已知 X_1=+1010011，X_2= –1010011，字长 n=8。求 X 的补码表示形式。

$[X_1]_{补}=\mathbf{0}\ 1010011$

$[X_2]_{补}=\mathbf{1}\ 0101101$

例 1.20 将“0”用补码表示。

$X_1=+0000000 \quad [X_1]_{补}=\mathbf{0}\ 0000000$

$X_2= -0000000 \quad [X_2]_{原}=\mathbf{1}\ 0000000 \quad [X_2]_{补}=\mathbf{1}\ 1111111+1=\mathbf{0}\ 0000000$

可以看出，当限定字长时，补码表示的+0 与–0 有唯一的表示形式。

1.3.2 带符号位二进制数补码运算

1. 二进制补码加法

补码表示的两个带符号位的二进制数相加时，将两个数（包括符号位）相加，其结果得到和的补码。即：

$$[A+B]_{补}=[A]_{补}+[B]_{补}$$

例 1.21 $X_1=7_{10}$，$X_2=8_{10}$，利用补码运算求二进制数 X_1+X_2，设字长为 8 位。

$$X_1=7_{10}=+0000111_2，X_2=8_{10}=+0001000_2$$

$$[X_1+X_2]_{补}=[X_1]_{补}+[X_2]_{补}=\mathbf{0}\ 0000111_2+\mathbf{0}\ 0001000_2=\mathbf{0}\ 0001111_2$$

$$\begin{array}{rr} & \mathbf{0}\,0000111 \\ +) & \mathbf{0}\,0001000 \\ \hline & \mathbf{0}\,0001111 \end{array}$$

得$[X_1+X_2]_{补}=[X_1+X_2]_{原}=\mathbf{0}\ 0001111_2$，

真值 $X_1+X_2=+0001111_2=+15_{10}$。

例 1.22 $X_1=6_{10}$，$X_2=-9_{10}$，利用补码运算求二进制数 X_1+X_2，设字长为 8 位。

$$X_1=6_{10}=+0000110_2，X_2=-9_{10}=-0001001_2$$

$$[X_1+X_2]_{补}=[X_1]_{补}+[X_2]_{补}=\mathbf{0}\ 0000110_2+\mathbf{1}\ 1110111_2=\mathbf{1}\ 1111101_2$$

$$\begin{array}{rr} & \mathbf{0}\,0000110 \\ +) & \mathbf{1}1110111 \\ \hline & \mathbf{1}1111101 \end{array}$$

得$[X_1+X_2]_{补}=\mathbf{1}\ 1111101_2$，$[X_1+X_2]_{原}=\mathbf{1}\ 0000011_2$

真值 $X_1+X_2=-0000011_2=-3_{10}$。

2. 二进制补码减法

补码表示的两个带符号位的二进制数相减时，利用关系：

$$[A-B]_{补}=[A+(-B)]_{补}=[A]_{补}+[-B]_{补}$$

将补码减法转换成补码加法进行运算。

例 1.23 $X_1=-10_{10}$，$X_2=9_{10}$，利用补码运算求二进制数 X_1-X_2。

$$X_1=-10_{10}=-0001010_2，X_2=9_{10}=0001001_2$$

$$[X_1-X_2]_{补}=[X_1+(-X_2)]_{补}=[X_1]_{补}+[-X_2]_{补}=\mathbf{1}\ 1110110_2+\mathbf{1}\ 1110111_2=\mathbf{1}\ 1101101_2$$

$$\begin{array}{rr} & \mathbf{1}1110110 \\ +) & \mathbf{1}1110111 \\ \hline & \mathbf{1}1101101 \end{array}$$

得$[X_1-X_2]_{原}=\mathbf{1}\ 0010011_2$，$X_1-X_2=-19_{10}$。

1.4 二进制编码

在数字系统中，各种类型的信息都要转换成一定位数的二进制码才能进行处理。用二进制数表示特定信息的过程称为编码，编码之后的二进制数称为二进制代码。

1.4.1　二—十进制编码

二—十进制编码，或称十进制数的二进制编码（Binary Coded Decimal，BCD）。常见的 BCD 码有 8421BCD 码、2421BCD 码及余 3 码等。这里只介绍 8421BCD 码。

8421BCD 码是一种有权的 BCD 码，用 4 位 8421BCD 码表示 1 位十进制数，各位的权值依次是 8、4、2 和 1，即 2^3、2^2、2^1 和 2^0（与普通 4 位二进制数的权值相同），0000～1001 分别对应 0～9 十个数。注意，8421BCD 码中不允许 1010～1111 六种情况出现。表 1-2 为十进制数与 8421BCD 码的对应关系。

表 1-2　8421BCD 码

十进制数	8421BCD 码	十进制数	8421BCD 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

例 1.24　将十进制数 78_{10}、30.25_{10} 分别用 8421BCD 码表示。

$$78_{10}=0111\ 1000_{BCD}$$

$$30.25_{10}=0011\ 0000.\ 0010\ 0101_{BCD}$$

提示：十进制数用 BCD 码表示或十进制数转换为等值二进制数，是两个不同的概念。

例如，若将例 1.25 中两个十进制数值数转换为二进制数，结果是：

$$78_{10}=1001110_2;\ 30.25_{10}=11110.01_2$$

1.4.2　字符编码

英文 ASCII 码和汉字码属于字符编码。字符是计算机中使用最多的信息形式之一，是人机通信与交互的重要媒介。在计算机中，要为每个字符指定一个确定的编码，作为识别这些字符的依据。

1. ASCII 码

使用最多的字符编码是 ASCII 编码（美国国家信息交换标准字符码），如表 1-3 所示。

从表 1-3 中可以看出，ASCII 码包括 10 个十进制数字（数字 0～9，对应 30H～39H）、26 个英文字母的大小写形式：A～Z（41H～5AH），a～z（61H～7AH）。以及一定数量的专用符号，共 128 个字符。因此二进制编码需要 7 位，从高到低排列次序为 $b_6b_5b_4b_3b_2b_1b_0$。

表 1-3　ASCII 字符编码表

$b_3b_2b_1b_0$ \ $b_6b_5b_4$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	、	p
0001	SOH	DC1	!	1	A	Q	a	q