普通高等院校电子信息类“十二五”规划教材

数字电子技术

SHUZI DIANZI JISHU

石建平 主编
陶文海 吴敏 张武 副主编

免费索取课件及习题答案
E-mail: yqu@ndip.cn

国防工业出版社
National Defense Industry Press

普通高等院校电子信息类“十一五”规划教材

数字电子技术

石建平 主编

陶文海 吴敏 张武 副主编

国防工业出版社

·北京·

前　言

本书是依据教育部颁发的《数字电路与逻辑设计》和《数字电子技术基础》的基本要求，鉴于二、三类理工院校蓬勃发展这一现实，为适应我国二、三类理工院校的教学条件、师资和学生状况而编写的。

编者近年来在多所二、三类理工院校担任本课程的教学工作，发现目前流行的一些教材都是由一些重点大学的教授针对一类学校的办学条件、师资和学生的状况编写的，内容太多，新器件和新技术太多，要求学生自学或选学的内容太多。这在一定程度上影响了基础内容的介绍，任课老师要进行大量的删选和补充工作，给教学工作带来一些困难，因此，我们联合多所二、三类理工院校担任本课程教学任务的教师参加编写本教材，这对于教材内容的取舍、课程体系结构的调整无疑是有益的，相信所编写出来的教材能更适合多数院校的教学需要。

编写本书的指导思想是着重本课程的基础内容、基本分析与设计方法以及基本的实验技能；在介绍小、中、大规模集成电路时，大幅删除烦琐数学推导与内部结构分析，着重介绍它们的外部特性及其性能；同时简要介绍一些新器件与新技术。其目的在于，使学生掌握本课程的基础内容，为许多后续课程的学习打下良好的基础；同时也适应数字电子技术发展的要求，培养不断吸收与应用新技术的能力。

本教材参考学时数为55~65。其中加“*”的部分为选学内容。为便于教学，章首有内容提要和概述，章末有小结和习题。

本书由安徽师范大学物理与电子信息学院石建平教授主编，并负责全书的策划和统审工作；参加编写的有陶文海（安徽师范大学），吴敏、张武、周琼（安徽农业大学），从宏寿、王彦、郎佳红（安徽工业大学），赵秀华（安徽工程科技学院），周珍良（安徽铜陵学院）。

由于编者水平所限，错误和不妥之处恳请广大读者批评指正。

目 录

第 1 章　数制与代码

一个数通常可以用两种不同的方法来进行表示，一种方法是按“值”表示；另一种方法是按“形”表示。

按“值”表示，是指选择某种进位制来确定某个数的值或大小，这就是数制。

按“值”表示时需要注意解决三个方面的问题：①恰当地选择数字符号——数码及其计数规则；②确定小数点的位置；③正确地表示出数的正、负。

按“形”表示，是指按照一定的编码方法来形象地表示某个数。

按“形”表示时，首先要确定编码规则，然后按此编码规则编出一组代码，最后给每一个代码赋以一定的含义，这就是码制（或代码）。

本章主要介绍各种进位计数制。从人们熟悉的十进制开始，把基本概念扩展到二进制、八进制和十六进制，并进一步导出任意（r）进制的相关规律。在了解各种数制的基础上，进一步介绍各种数制之间的相互转换方法。本章还介绍了二进制数的补码、正负表示法以及补码运算。本章最后介绍数字系统中常用的二进制代码。

1.1　数　　制

一种数制的基本特征是它们的基数，基数是指该数制中用来表示数值的数字符号——数码个数。

数的数制表示方法一般都采用位置计数法。位置计数法是指在一个数中，每一个数码和数码所在的位置共同决定了该数的大小；即数码本身有大小，而每一个数码所在的位置也载有表示该数大小的一个特定的数值，这个数值称为位置的“权”，简称为位“权”。每一个位置的“权”可以用基数的幂的形式来表示。

1.1.1　十进制

十进制（Decimal）是人们经常使用的一种数制。十进制有 0、1、2、3、4、5、6、7、8、9 这十个数码，所以它的基数为 10，位“权”就是…，10^2（百位），10^1（十位），10^0（个位），10^{-1}（十分位），10^{-2}（百分位），…。十进制的计数规则是“逢十进一”，即 $9+1=10_D$，下标 D 表示十进制数。

位权展开式是位置计数法的数学描述形式。一个具有 n 位整数和 m 位小数的任意十进制数 N_D（$N_D=a_{n-1}a_{n-2}\cdots a_1a_0a_{-1}a_{-2}\cdots a_{-m}$是它的简写形式）的位权展开式为

$$\begin{aligned}N_D = {} & a_{n-1}\times 10^{n-1} + a_{n-2}\times 10^{n-2} + \cdots + a_1\times 10^1 + a_0\times 10^0 + \\ & a_{-1}\times 10^{-1} + a_{-2}\times 10^{-2} + \cdots + a_{-m}\times 10^{-m}\end{aligned}$$

即

$$N_D = \sum_{i=-m}^{n-1} a_i \times 10^i,\ a_i \in \{0,\ \ 1, \cdots, 9\} \tag{1-1}$$

【例 1-1】 求十进制数 278.94_D 的位权展开式。

解 $278.94_D = 2\times10^2 + 7\times10^1 + 8\times10^0 + 9\times10^{-1} + 4\times10^{-2}$
$= 200 + 70 + 8 + 0.9 + 0.04$

1.1.2 二进制

数字系统中通常采用的数制是二进制(Binary)。二进制只有 0、1 这两个数码,所以它的基数为 2,位"权"就是$\cdots, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, \cdots$。二进制的计数规则是"逢二进一",即 $1+1=10_B$,下标 B 表示二进制数。

一个具有 n 位整数和 m 位小数的任意二进制数 N_B($N_B = b_{n-1}b_{n-2}\cdots b_1 b_0 b_{-1} b_{-2}\cdots b_{-m}$ 是它的简写形式)的位权展开式为

$$N_B = b_{n-1}\times2^{n-1} + b_{n-2}\times2^{n-2} + \cdots + b_1\times2^1 + b_0\times2^0 + b_{-1}\times2^{-1} + b_{-2}\times2^{-2} + \cdots + b_{-m}\times2^{-m}$$

即

$$N_B = \sum_{i=-m}^{n-1} b_i \times 2^i,\ b_i \in \{0,1\} \tag{1-2}$$

【例 1-2】 求二进制数 10011.1101_B 的位权展开式。

解 $10011.1101_B = 1\times2^4 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2} + 1\times2^{-4}$
$= 16 + 2 + 1 + 0.5 + 0.25 + 0.0625$
$= 19.8125_D$

1.1.3 八进制和 16 进制

由于多位二进制数不便识别和记忆,因此常采用八进制和 16 进制作为二进制的缩写形式。使用八进制或 16 进制可以将对应的二进制数的字长缩短为原来的 1/3 或 1/4,因此八进制和 16 进制是计算机资料中常用的数制。

1. 八进制

八进制(Octal)有 0、1、2、3、4、5、6、7 这八个数码,所以它的基数为 8,位"权"就是$\cdots$, $8^2, 8^1, 8^0, 8^{-1}, 8^{-2}, \cdots$。八进制的计数规则是"逢八进一",即 $7+1=10_O$,下标 O 表示八进制数。

一个具有 n 位整数和 m 位小数的任意八进制数 N_O($N_O = c_{n-1}c_{n-2}\cdots c_1 c_0 c_{-1} c_{-2}\cdots c_{-m}$ 是它的简写形式)的位权展开式为

$$N_O = c_{n-1}\times8^{n-1} + c_{n-2}\times8^{n-2} + \cdots + c_1\times8^1 + c_0\times8^0 + c_{-1}\times8^{-1} + c_{-2}\times8^{-2} + \cdots + c_{-m}\times8^{-m}$$

即

$$N_O = \sum_{i=-m}^{n-1} c_i \times 8^i,\ c_i \in \{0,1,\cdots,7\} \tag{1-3}$$

【例 1-3】 求八进制数 372.01_O 的位权展开式。

解 $372.01_O = 3\times8^2+7\times8^1+2\times8^0+1\times8^{-2}$

$= 192+56+2+0.015625$

$= 250.015625_D$

2. 16 进制

16 进制（Hexadecimal）有 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F（其中 A ~ F 表示 10 ~ 15）这十六个数码，所以它的基数为 16，位“权”就是…，16^2，16^1，16^0，16^{-1}，16^{-2}，…。16 进制的计数规则是“逢十六进一”，即 $F+1=10_H$，下标 H 表示 16 进制数。

一个有 n 位整数和 m 位小数的任意 1 进制数 N_H（$N_H=d_{n-1}d_{n-2}\cdots d_1d_0d_{-1}d_{-2}\cdots d_{-m}$ 是它的简写形式）的位权展开式为

$$N_H = d_{n-1}\times16^{n-1}+d_{n-2}\times16^{n-2}+\cdots+d_1\times16^1+d_0\times16^0+ d_{-1}\times16^{-1}+d_{-2}\times16^{-2}+\cdots+d_{-m}\times16^{-m}$$

即

$$N_H = \sum_{i=-m}^{n-1} d_i\times16^i,\ d_i\in\{0,\ 1,\cdots,9,A,\cdots,F\} \tag{1-4}$$

【**例 1-4**】 求 16 进制数 $E5D.A3_H$ 的位权展开式。

解 $E5D.A3_H = 14\times16^2+5\times16^1+13\times16^0+10\times16^{-1}+3\times16^{-2}$

$= 3584+80+13+0.625+0.01171875$

$= 3677.63671875_D$

1.1.4 任意进制

任意进制（也称为 r 进制）有 $0,1,2,\cdots,(r-1)$ 这 r 个数码，所以它的基数为 r，位“权”就是…，r^2，r^1，r^0，r^{-1}，r^{-2}，…。r 进制的计数规则是“逢 r 进一”，即 $(r-1)+1=10_r$，下标 r 表示任意进制数。

一个具有 n 位整数和 m 位小数的任意 r 进制数 N_r（$N_r=p_{n-1}p_{n-2}\cdots p_1p_0p_{-1}p_{-2}\cdots p_{-m}$ 是它的简写形式）的位权展开式为

$$N_r = p_{n-1}\times r^{n-1}+p_{n-2}\times r^{n-2}+\cdots+p_1\times r^1+p_0\times r^0+ p_{-1}\times r^{-1}+p_{-2}\times r^{-2}+\cdots+p_{-m}\times r^{-m}$$

即

$$N_r = \sum_{i=-m}^{n-1} p_i\times r^i,\ p_i\in\{0,1,\cdots,(r-1)\} \tag{1-5}$$

【**例 1-5**】 求四进制数 30.12_4 的位权展开式。

解 $30.12_4 = 3\times4^1+1\times4^{-1}+2\times4^{-2}$

$= 12+0.25+0.125$

$= 12.375_D$

也就是说，一个任意进制数的位权展开式，可以由该数中的每一个数码乘以它所在位置的“权”，然后再将这些乘积累加起来得到；而且一个任意进制数的位权展开式的值一定是十进制数。

1.2 数制之间的相互转换

1.2.1 其他进制数转换为十进制数

将二进制数、八进制数、16 进制数以及 r 进制数转换为十进制数的方法很简单，只要按照位置计数法，求出各个数码与所在位置“权”的乘积，然后把各乘积项累加起来，就可得到转换结果。

1.2.2 十进制数转换为其他进制数

任意十进制数都是由整数和小数这两部分组成，对整数和小数部分分别进行转换，再将这两部分的转换结果排列组合在一起即可得到完整的转换结果。

下面分别讨论十进制整数和小数部分的转换原理。

1. 整数部分

对整数部分通常采用基数除法——“除基反序取余法”，直到商为 0 时停止。

对于一个任意的十进制整数 N_D，总是与一个 n 位的 r 进制整数 M_r 一一对应，即 $N_D = M_r = p_{n-1}p_{n-2}\cdots p_1p_0$。而 M_r 的位权展开式为

$$N_D = M_r = p_{n-1} \times r^{n-1} + p_{n-2} \times r^{n-2} + \cdots + p_1 \times r^1 + p_0 \times r^0$$

$$N_D = r \times (p_{n-1} \times r^{n-2} + p_{n-2} \times r^{n-3} + \cdots + p_1 \times r^0) + p_0 = r \times A_0 + p_0$$

将 N_D 除以 r 后所得商为 A_0，余数为 p_0，此为该 r 进制整数最低一位的数码，即

$$A_0 = r \times (p_{n-1} \times r^{n-3} + p_{n-2} \times r^{n-4} + \cdots + p_2 \times r^0) + p_1 = r \times A_1 + p_1$$

再将 A_0 除以 r，商为 A_1，余数为 p_1，此为该 r 进制整数倒数第二位的数码。

连续地将商除以 r，就可依次得到数码 $p_2, p_3, \cdots$，直到商为 0 时停止，得到该 r 进制整数的最高位数码 p_{n-1}。

2. 小数部分

对小数部分通常采用基数乘法——“乘基顺序取整法”，直到小数部分为 0 或达到所需要的转换精度时停止。

对于一个任意的十进制小数 N_D，总是与一个 m 位的任意 r 进制小数 M_r 一一对应，即 $N_D = M_r = 0.p_{-1}p_{-2}\cdots p_{-m}$。而 M_r 的位权展开式为

$$N_D = M_r = p_{-1} \times r^{-1} + p_{-2} \times r^{-2} + \cdots + p_{-m} \times r^{-m}$$

$$N_D \times r = p_{-1} + (p_{-2} \times r^{-1} + p_{-3} \times r^{-2} + \cdots + p_{-m} \times r^{-m+1}) = p_{-1} + A_{-1}$$

将 N_D 乘以 r，其中 p_{-1} 作为整数溢出，此为该 r 进制小数最高一位的数码，括号中的数 A_{-1} 仍为小数，即

$$A_{-1} \times r = p_{-2} + (p_{-3} \times r^{-1} + p_{-4} \times r^{-2} + \cdots + p_{-m} \times r^{-m+2}) = p_{-2} + A_{-2}$$

再将 A_{-1} 乘以 r，其中 p_{-2} 作为整数溢出，此为该 r 进制小数第二位的数码；括号中的数 A_{-2} 仍为小数；

连续地将小数部分乘以 r，就可依次得到数码 $p_{-3}, p_{-4}, \cdots$，直到小数部分为 0 时停止，得到该 r 进制小数的最低一位数码 p_{-m}，或达到需要的转换精度时停止。

【例 1－6】 $41.375_D = 101001.011_B$

解

整数部分

除数	被除数或商	余数	
2	41	$\boxed{1}$	↑
2	20	$\boxed{0}$	↑ 反
2	10	$\boxed{0}$	↑
2	5	$\boxed{1}$	↑
2	2	$\boxed{0}$	↑ 序
2	1	$\boxed{1}$	↑
	0		

小数部分

	0.375
	× 2
顺↓	$\boxed{0}.75$
	× 2
↓	$\boxed{1}.50$
	× 2
序↓	$\boxed{1}.00$

【例 1－7】 $153.25_D = 231.2_O$

解

整数部分

除数	被除数或商	余数	
8	153	$\boxed{1}$	↑ 反
8	19	$\boxed{3}$	↑
8	2	$\boxed{2}$	↑ 序
	0		

小数部分

	0.25
	× 8
顺序↓	$\boxed{2}.00$

1.2.3 二进制数和八进制数之间的相互转换

由于 $2^3 = 8$，所以每一位八进制数与 3 位二进制数一一对应，见表 1－1。

表 1－1 八进制与二进制对照表

八进制	二进制	八进制	二进制
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

1. 二进制数转换为八进制数

从小数点开始，向左（对整数部分）、向右（对小数部分）将每 3 位二进制数组成一组，最高位和最低位不足 3 位的补 0（缺几位就补几个 0）；再将每 3 位二进制数用 1 位对应的八进制数进行替换就可以得到转换结果。

【例 1－8】 将 11100110101.0100001111_B 转换为八进制数。

解 11100110101.0100001111_B

$= \underline{011}\ \underline{100}\ \underline{110}\ \underline{101}.\underline{010}\ \underline{000}\ \underline{111}\ \underline{100}_B$

$= 3465.2074_O$

2. 八进制数转换为二进制数

只要将每一位八进制数用对应的 3 位二进制数进行替换就可以得到转换结果。

【例 1-9】 将 3657.0124_O 转换为二进制数。

解 $3657.0124_O = \underline{011}\ \ \underline{110}\ \ \underline{101}\ \ \underline{111}.\underline{000}\ \ \underline{001}\ \ \underline{010}\ \ \underline{100}_B$

$= 11110101111.0000010101_B$

1.2.4 二进制数和 16 进制数之间的相互转换

由于 $2^4 = 16$,所以每一位 16 进制数与 4 位二进制数一一对应,见表 1-2。

表 1-2 16 进制与二进制对照表

16 进 制	二 进 制	16 进 制	二 进 制
0	0000	8	1000
1	0001	9	1001
2	0010	*A*	1010
3	0011	*B*	1011
4	0100	*C*	1100
5	0101	*D*	1101
6	0110	*E*	1110
7	0111	*F*	1111

1. 二进制数转换为 16 进制数

从小数点开始,向左(对整数部分)、向右(对小数部分)将每 4 位二进制数组成一组,最高位和最低位不足 4 位的补 0(缺几位就补几个 0);再将每 4 位二进制数用 1 位对应的 16 进制数进行替换就可以得到转换结果。

【例 1-10】 将 11100110101.0100001111_B 转化为 16 进制数。

解 $11100110101.0100001111_B = \underline{0111}\ \ \underline{0011}\ \ \underline{0101}.\underline{0100}\ \ \underline{0011}\ \ \underline{1100}_B$

$= 735.43C_H$

2. 16 进制数转换为二进制数

只要将每一位 16 进制数用对应的 4 位二进制数进行替换就可以得到转换结果。

【例 1-11】 将 $6E4B.F1D8_H$ 转化为二进制数。

解 $6E4B.F1D8_H = \underline{0110}\ \ \underline{1110}\ \ \underline{0100}\ \ \underline{1011}.\underline{1111}\ \ \underline{0001}\ \ \underline{1101}\ \ \underline{1000}_B$

$= 110111001001011.1111000111011_B$

1.2.5 八进制数和 16 进制数之间的相互转换

八进制数和 16 进制数之间的相互转换必须以二进制数为桥梁,即

八进制数 ⇔ 二进制数 ⇔16 进制数

1.3 二进制数的正负表示法

二进制数的正负表示法有原码表示法、补码表示法和反码表示法三种形式。首先介绍二进制数的补码概念。

1.3.1 二进制数的补码

二进制数的补码有两种形式：一种形式称为基数的补码，即 2 的补码；另一种形式是基数减一的补码，即 1 的补码。

1. 2 的补码

习惯上把 2 的补码简称为补码。如果以 N_B 表示一个具有 n 位整数（小数位不限）的任意二进制数，用$\overline{N_2}$表示其补码，那么有

$$\overline{N_2} = 2^n - N_B \tag{1-6}$$

由式(1-6)可知，二进制数的补码是由参考数 2^n（n 是该二进制数整数位的位数）减去这个数本身得到的。

【例 1-12】 求解 11010_B、0.10101_B 和 11010.10101_B 的补码。

解 11010_B 的补码为 $2^5 - 11010_B = 00110_B$；

0.10101_B 的补码为 $2^1 - 0.10101_B = 1.01011_B$；

11010.10101_B 的补码为 $2^5 - 11010.10101_B = 00101.01011_B$。

2. 1 的补码

习惯上把 1 的补码简称为反码。如果以 N_B 表示一个具有 n 位整数、m 位小数的任意二进制数，用$\overline{N_1}$表示其反码，那么有

$$\overline{N_1} = (2^n - 2^{-m}) - N_B \tag{1-7}$$

由式(1-7)可知，二进制数的反码是由参考数($2^n - 2^{-m}$)（n、m 分别是该二进制数整数位和小数位的位数）减去这个数本身得到的。

【例 1-13】 求解 11010_B、0.10101_B 和 11010.10101_B 的反码。

解 11010_B 的反码为$(2^5 - 2^{-0}) - 11010_B = 00101_B$；

0.10101_B 的反码为$(2^1 - 2^{-5}) - 0.10101_B = 1.01010_B$；

11010.10101_B 的反码为：$(2^5 - 2^{-5}) - 11010.10101_B = 00101.01010_B$。

对例 1-12 和例 1-13 进行分析可得到：

(1) 求二进制数补码的一种方法是，将该二进制数最低一位的 1 及其右边的数码保持不变，左边的数码逐位求反即可。将二进制数补码的最低一位减 1，就可得到该二进制数的反码。

(2) 求二进制数反码的另一种方法是，可将该二进制数的每一位数码直接求反，即数码 1 变成 0、0 变成 1，就可得到它的反码。在二进制数反码的最低一位加 1，就可得到二进制数的补码。

(3) 如果将二进制数的补码再求补一次，或将二进制数的反码再求反一次，就都将还原为原来的二进制数。

1.3.2 二进制数的正负表示法

通常在一个二进制数最高位的左边加上符号位来表示该二进制数的正负（符号位和

最高位之间用逗号分隔,可省略)。通常用“0”表示正号,用“1”表示负号。

在数字系统中,二进制数的正负表示法有原码表示法、补码表示法和反码表示法三种形式。

1. 原码表示法

原码表示法,就是将“0”或“1”加到该二进制数绝对值最高位左端的符号位,便可用来表示正二进制数或负二进制数。

$+45_D$ 的原码表示为 $0,101101_B$; -45_D 的原码表示为 $1,101101_B$。

$+0_D$ 的原码表示为 00000000_B;无 -0_D 的原码表示。

八位二进制数的原码表示范围为 $01111111_B \sim 11111111_B$,对应 $+127_D \sim -127_D$,可以表示 $2^8-1=255$ 个数。

2. 补码表示法

正二进制数的补码表示等同于原码表示,负二进制数的补码表示为符号“1”加上该数绝对值的补码。

$+45_D$ 的补码表示为 $0,101101_B$; -45_D 的补码表示为 $1,010011_B$。

$+0_D$ 的补码表示为 00000000_B; -128_D 的补码表示为 10000000_B。

8 位二进制数的补码表示范围 $01111111_B \sim 10000000_B$ 对应 $+127_D \sim -128_D$,可以表示 $2^8=256$ 个数。

3. 反码表示法

正二进制数的反码表示等同于原码表示;负二进制数的反码表示为符号“1”加上该数绝对值的反码。

$+45_D$ 的反码表示为 $0,101101_B$; -45_D 的反码表示为 $1,010010_B$。

$+0_D$ 的反码表示为 00000000_B; -0_D 的反码表示为 11111111_B。

8 位二进制数的反码表示范围 $01111111_B \sim 10000000_B$,对应 $+127_D \sim -127_D$,可以表示 $2^8-1=255$ 个数。

1.3.3 二进制数的补码运算

在数字电路中,用原码运算求两个正数 M 和 N 的差值($M-N$)时,首先要对减数和被减数进行比较;然后由大数减去小数;最后决定差值的符号。完成这个运算,电路复杂,速度慢。如用补码实现减法运算,可以把减法运算转变为加法运算,即($M-N$)转变为($M+(-N)$),也就是说,被减数 M 加上减数($-N$)的补码,这样就把原码的减法运算转变成了补码的加法运算。

【例 1-14】求 13_D-10_D。

解 $13_D-10_D=13_D+(-10_D)$

13_D 的补码表示为 $0\quad 0\cdots01101_B$;(-10_D)的补码表示为 $1\quad 1\cdots10110_B$。

$$\begin{array}{r} 0\quad 0\cdots01101_B \\ +\quad 1\quad 1\cdots10110_B \\ \hline [1]\ 0\quad 0\cdots00011_B \end{array}$$

最高位的进位[1]溢出,即 $0\quad 0\cdots01101_B+1\quad 1\cdots10110_B=0\quad 0\cdots00011_B=+3_D$,得