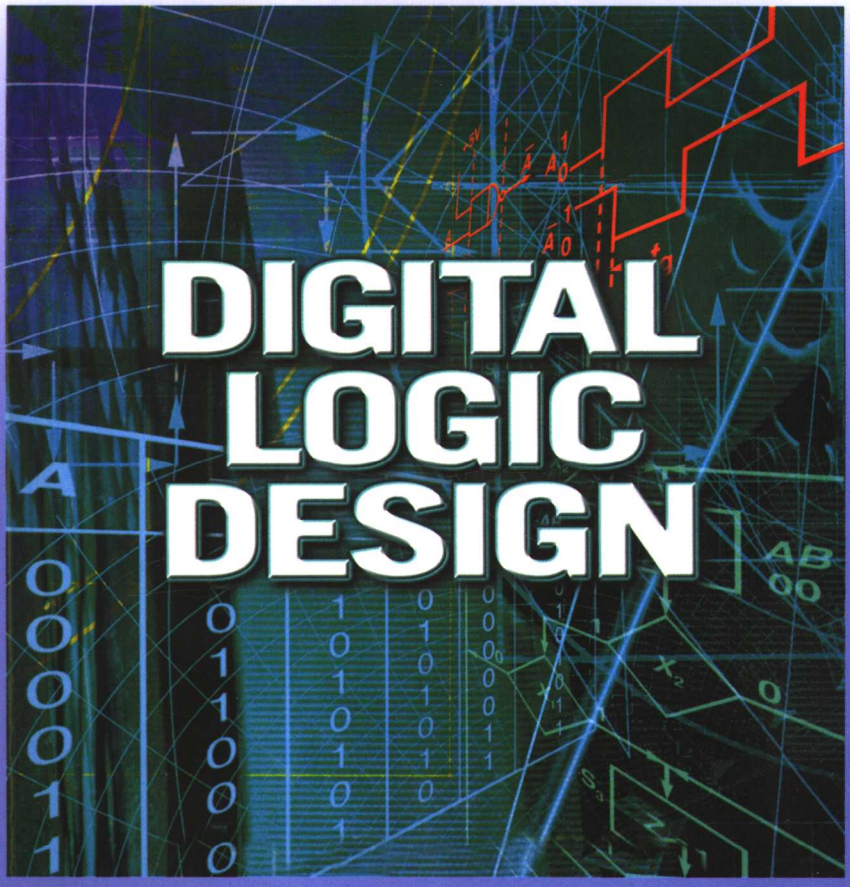
国外著名高等院校
信息科学与技术优秀教材

Newnes

数字逻辑设计(第四版)

DIGITAL LOGIC DESIGN

FOURTH EDITION

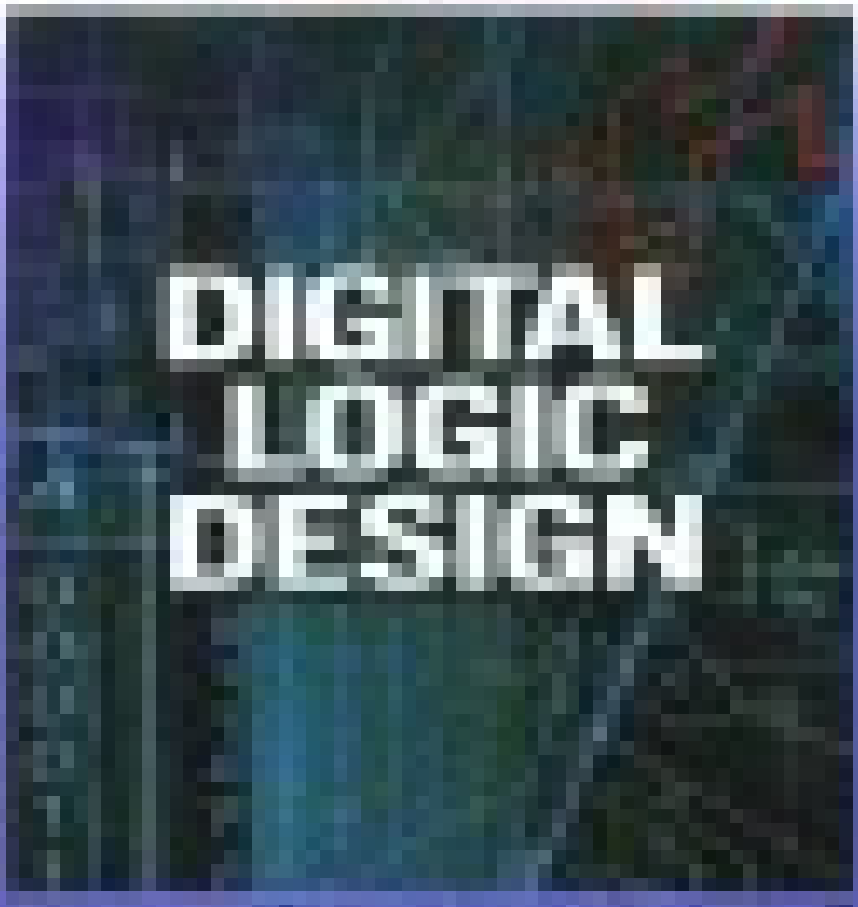


〔美〕Brian Holdsworth　Clive Woods　著
李仁发　肖玲　吴强　译

人民邮电出版社
POSTS & TELECOM PRESS

数字逻辑设计

DIGITAL LOGIC DESIGN



国外著名高等院校信息科学与技术优秀教材

数字逻辑设计(第四版)

DIGITAL LOGIC DESIGN

FOURTH EDITION

〔美〕Brian Holdsworth　Clive Woods　著

李仁发　肖玲　吴强　译

人民邮电出版社

图书在版编目（CIP）数据

数字逻辑设计：第4版/（美）何尔德斯沃斯斯著；李仁发，肖玲，吴强译.
—北京：人民邮电出版社，2006.5
国外著名高等院校信息科学与技术优秀教材
ISBN 7-115-13721-8

Ⅰ.数… Ⅱ.①何… ②李… ③肖… ④吴… Ⅲ.数字电路—逻辑设计—高等学校—教材
Ⅳ.TN79

中国版本图书馆CIP数据核字(2006)第030319号

国外著名高等院校信息科学与技术优秀教材
数字逻辑设计（第四版）

◆ 著 [美]Brian Holdsworth Clive Woods
译 李仁发 肖 玲 吴 强
责任编辑 刘映欣
◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
◆ 开本：787×1092 1/16
印张：24.75
字数：601千字 2006年5月第1版
印数：1-4 000册 2006年5月北京第1次印刷
著作权合同登记 图字：01-2004-6844号

ISBN 7-115-13721-8/TP·4841

定价：45.00元

读者服务热线：(010) 67132705 印装质量热线：(010) 67129223

内容提要

本书系统地介绍了数字逻辑电路设计的基础知识，内容全面，实用性强。全书共分 13 章，系统地阐述了数制和码制、布尔代数、卡诺图和布尔函数化简、组合逻辑设计方法、中规模集成电路的组合逻辑设计、锁存器和触发器、计数器和寄存器、时钟驱动的时序电路、事件驱动的电路、测量与接口、可编程逻辑器件、算术运算电路、故障诊断与测试等内容，并附有逻辑功能符号的介绍。书中每章有一定数量的习题，书后给出了答案。

作者在最新修订的第四版中增加了扩展的二-十进制码、格雷码、触发器应用、轴和线性编码器、存储元件和现场可编程门阵列（FPGA）等内容，新增了介绍数字部件与模拟信号之间接口一章的内容，更新了故障定位等章节内容，使本书覆盖了数字系统设计的最新技术和进展情况。

本书内容由浅入深，既适用于数字逻辑电路与数字系统的基础教学，也可用于深入提高，适合作为高等学校计算机专业“数字逻辑”课程的教材，亦可供从事计算机、自动化及电子学方面生产和科学研究的人员等参考。

序

在新修订的这一版《数字逻辑设计》中，我们保留了第三版中对数字逻辑设计内容的全面介绍，同时借此机会对其中大部分内容进行了修订和更新。与以前的版本一样，该版本覆盖了本科或研究生数字逻辑系统课程所需的全部内容，也可供在该领域工作的毕业生学习参考。为此，我们保留了前版的所有基础知识介绍，不需要读者有很多的背景知识。考虑到近年来硬件的发展趋势，我们对后面的高级章节做了相应的修改。有些章节的最后设置了不少附加问题，其中一些问题没有给出答案，是为了让读者在没有解答提示的情况下锻炼自己的设计能力。

有关测量与接口一章的内容几乎是全新的，有关可编程逻辑器件及故障诊断和测试章节的内容被扩充了。一方面，这是由于技术的进步和设计者可用器件范围的扩大；另一方面，是为了强调故障查找的方法对于专业工程师来说，不再是深奥的、不能应用于实践的、不重要的，而是值得我们认真学习和运用的。

> 在这种方框中显示的内容通常在本书的后面不会被使用到，也不如主要内容重要，有时只是对主要内容的总结。这些内容可能比主要内容更难一些，可能在某种方式上是非同寻常的或晦涩难懂的；一般而言，这些内容给出的结果没有详细的论证求解过程，更多的是作为一种挑战留给感兴趣的读者去完成。第一次阅读本书的读者，或不是为了完整浏览本书的读者，可以跳过这些内容。

在书中我们使用了“旧”的 IEEE 逻辑符号，而不是“新”的 BS3939 符号。这是由于感觉到工程界对此有态度上的转变，现在 IEEE 符号和 BS 符号都被推荐使用了。现代的计算机辅助设计（CAD）系统可以很容易地打印出“旧”的符号，抵消了当初引进“新”符号的主要优势。然而，由于读懂“新”符号也是一种有用技能，我们在附录中给出了“新”符号系统一览表。

本书各章习题均附有答案，有需要答案的教师可发邮件至 Liuyingxin@ptpress.com.cn 免费索取。

译者序

“数字逻辑”课程是高等院校计算机科学与技术、信息与通信工程、电子信息与技术、控制科学与工程学科等各个专业的必修基础课程之一，在本科教学课程体系结构中占据着核心地位。建设好“数字逻辑”课程对于提高电子信息类专业本科教学水平，培养面向新世纪的高素质计算机、电子、通信、自动控制等专业人才十分重要。

课程建设的一个重要环节是教材建设。教育部在《关于进一步加强高等学校本科教学工作的若干意见》文件中指出，“对发展迅速和应用性强的课程，要不断更新教材内容，积极开发新教材，并使高质量的新版教材成为教材选用的主体”。因此，湖南大学计算机与通信学院在人民邮电出版社的大力支持下，组织翻译了《Digital Logic Design，Fourth Edition》这一国外经典教材，期望与国内同仁共享国外先进成果与经验。

本书是两位享有盛誉的学者 Brian Holdsworth 和 Clive Woods 创造性劳动的结晶，是备受推崇的教材，是针对培养从事 VLSI 电路、印刷电路板、多芯片模块及计算机电路设计的工程师和科研人员开设数字逻辑电路课程而编写的，本书向读者全面而清晰地揭示了数字设计理论与实践方面的最新观点。作者特别强调从系统的角度解决问题的方法，并配有很多设计实例，同时将计算机辅助设计方法贯穿全书，使其能应用到相关的概念和设计原理上。全书内容丰富，覆盖面广，系统地阐述了数制和码制、布尔代数、组合逻辑设计、时序逻辑设计、算术运算电路设计等内容，并附有逻辑功能符号的介绍。在最近修订的第四版中，作者还更新并补充了测量与接口、可编程逻辑器件、故障诊断和测试等章节的内容，以反映硬件技术的发展趋势和应用设计的需要。本书内容有浅有深，既适用于数字逻辑电路与数字系统的基础教学，也可作深入提高之用。我们相信，书中所展现的作者渊博的电路设计知识和丰富的数学经验对从事数字逻辑设计的教学和科研人员将大有裨益。

本书由湖南大学计算机与通信学院院长李仁发教授组织翻译，肖玲、吴强、王家琴、潘怿、杨莉、张惜珍、魏峰、高利、禹亮等参与了各章及附录的翻译。由于水平有限，翻译中的不当之处在所难免，恳请读者不吝批评指正。

湖南大学计算机与通信学院院长

李仁发

2006 年 3 月

目 录

第1章 数字系统与编码

1.1 简　介

一个数字逻辑系统除了其固有的逻辑能力之外，也应该具有良好的数字计算能力，因此它必须能够执行加、减、乘、除4种基本数学运算。人们一般采用十进制数字系统进行数学计算，但是，数字机器是采用二进制的，其数字运算是使用二进制数字系统来实现的。

因为十进制系统有10个数字，这就要求一个十状态机来表示这10个数字，每一个十进制数字用一个状态来表示。在电子世界中很难找到一个可用的十状态机，而像工作在开关模式上的晶体管的二状态机却可以找到，因此二进制数字系统对于数字工程师具有重要的意义。除了二进制系统之外，许多其他的系统，比如十六进制系统就和逻辑编程设备联合被使用，因此数字工程师必须熟悉各种各样不同的数字系统。

在数字机器上执行数学运算的过程和人们采用纸和笔计算的方法不必完全相同。例如，减法的运算过程是用加法来实现的，其中包括了补运算的使用。

再者，很明显常常需要一台数字机应以十进制的表示方式输出。由于机器通常是采用纯二进制方式进行计算，因而需要一种由二进制数表示十进制数的方法，它需要一个二—十进制系统。必须设计这样一种方法，以使单纯的二进制计算可以被转换成二—十进制数来实现，这就找到了一个有效的连接外部十进制世界的接口，从而输出或者读取一个十进制数字。

在数字系统的使用中，信息编码是要考虑的基本因素。需要对使用到的符号进行编码，包括十进制的数字、字母表中的字母和经常使用到的各种各样的符号，比如“=”、“?”等。我们把二—十进制作为十进制编码的一种编码方式。加权码就是许多例子中的一个。除了加权码之外还有很多编码的方式可用，比如XS3码，使用者可以自由选择适合的编码方式。在用户选择之前应考虑它的特性。实际最广泛使用的是8-4-2-1加权码，它被称作自然的二—十进制（naturally binary coded decimal）。

本章的目的就是要描述经常使用的各种数字系统，并且改进在一台机器上执行4种基本运算操作的方法。此外，将描述一些经常使用的编码方式。

1.2 数字系统

人们最熟悉的数字系统是十进制系统。一个十进制数比如$(473.85)_{10}$被表示成如

下形式：

$$(N)_{10} = 4\times10^2 + 7\times10^1 + 3\times10^0 + 8\times10^{-1} + 5\times10^{-2}$$

数字 $(N)_{10}$ 是由一系列相乘的 $(10)_{10}$ 的项组成。数字权值的大小根据位置不同而不同。数字 $(10)_{10}$ 被称作数字系统的基数。它的大小等于系统中所含的数字的数目。例如，在十进制系统中有 10 个数字，包括 0～9，而二进制数字系统的基数是 2，并且只有 2 个数字 0 和 1。

在任何数字系统中，十进制数 $(N)_{10}$ 的量值都可以用下面的等式来表示：

$$(N)_{10} = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \cdots a_0b^0 + a_{-1}b - 1\cdots + a_{-m}b^{-m}$$

其中 n 是整数部分数字的个数，m 是小数部分数字的个数。b 是该系统的基数，a 是系统中以 b 为基数的数字。用这样的等式可以把二进制数 $(101.11)_2$ 表示如下：

$$\begin{aligned}(N)_{10} &= 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2}\\ &= 4.0 + 0.0 + 1.0 + 0.5 + 0.25\\ &= (5.75)_{10}\end{aligned}$$

其他两个比较重要的数字系统是八进制和十六进制系统。八进制系统包括 0～7 共 8 个数字。一个典型的八进制数是 $(27.2)_8$，它的十进制表示如下：

$$\begin{aligned}(N)_{10} &= 2\times8^1 + 7\times8^0 + 2\times8^{-1}\\ &= 16.0 + 7.0 + 0.25\\ &= (23.25)_{10}\end{aligned}$$

在十六进制系统中有 16 个数字，由于只有 10 个数字可用，通常使用字母表中的前 6 个字母 A 到 F 表示其他的那 6 个数字，它们对应的十进制数值如下：

$(A)_{16} = (10)_{10}$　　$(B)_{16} = (11)_{10}$　　$(C)_{16} = (12)_{10}$

$(D)_{16} = (13)_{10}$　　$(E)_{16} = (14)_{10}$　　$(F)_{16} = (15)_{10}$

一个典型的十六进制数 $(A2.C)_{16}$ 表示成十进制数如下：

$$\begin{aligned}(N)_{10} &= A\times16^1 + 2\times16^0 + C\times16^{-1}\\ &= 160.0 + 2.0 + 0.75\\ &= (162.75)_{10}\end{aligned}$$

1.3 数字系统之间的转换

以任意基数表示的一个数字都可以分成两部分，小数点左边的整数部分（a）和小数点右边的小数部分（b）。在转换为其他数字系统的过程中，这两个部分的转换过程是不相同的。

在基数为 b 的系统中，整数部分 $(N_I)_{10}$ 表示成如下十进制值：

$$(N_I)_{10} = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \cdots a_1b^1 + a_0b^0$$

等式两边分别除以基数 b 得到：

$$\frac{(N_I)_{10}}{b} = a_{n-1}b^{n-2} + a_{n-2}b^{n-3} + a_{n-3}b^{n-4} + \cdots a_1b^0 + \frac{a_0}{b}$$

第一次除以基数的结果得到了余数 a_0，把这个最小的数字舍弃，接着重复上面的除法可以产生余数 $a_1, a_2 \cdots a_{n-1}$。下面的例子用这种连续除以基数的方法分别把十进制数 $(100)_{10}$ 转换为二进制、八进制和十六进制：

2	100	0	↑	8	100	4	↑	16	100	4	↑
2	50	0	|	8	12	4	|	16	6	6	|
2	25	1	|	8	1	1	|		0		
2	12	0	|		0						
2	6	0	|								
2	3	1	|								
2	1	1	|								
	0										

$(100)_{10}=(1100100)_2$　$=(144)_8$　$=(64)_{16}$

在基数为 b 的系统中，小数部分 $(N_F)_{10}$ 表示成如下十进制值：

$$(N_F)_{10} = a_{-1}b^{-1} + a_{-2}b^{-2} + \cdots a_{-m}b^{-m}$$

如果等式两边分别乘以基数，那么

$$b(N_F)_{10} = a_{-1} + a_{-2}b^{-1} + \cdots a_{-m}b^{-(m-1)}$$

第一次相乘提出了系数 a_{-1}，随后的乘法将分别提出系数 a_{-2}，$a_{-3}\cdots a_{-m}$。

为了说明这个过程，请看下面把十进制数 $(0.265)_{10}$ 分别转换成其对应的二进制、八进制和十六进制的例子：

.265×2	.265×8	.265×16
0.530×2	2.120×8	4.240×16
1.060×2	0.960×8	3.840×16
0.120×2	7.680×8	D.440×16
0.240×2	5.440×8	7.040×16
0.480	3.520	0.640
$(0.265)_{10}=(0.0100)_2$	$(0.20753)_8$	$(0.43D70)_{16}$

十进制数 $(0.265)_{10}$ 分别被表示成 5 位的二进制、八进制和十六进制数。除了这些从十进制到二进制、八进制和十六进制的换算外，把八进制和十六进制转换成二进制也是可行的，反之亦然。

每一个八进制数字都可以用一个三位的二进制数来表示，如图 1.1 (a) 所示。将一个二进制数字串转换成八进制的时候，要从最低位开始，每 3 个数字分成一组。每组 3 个二进制数所表示的八进制数，对应与如下的辅助转换表格。

$$(110\ 001\ 011\ 100)_2$$
$$= (6\quad 1\quad 3\quad 4)_8$$

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

(a) 八进制/二进制转换表

十进制	二进制	十进制	二进制
0	000	8	1000
1	001	9	1001
2	010	A	1010
3	011	B	1011
4	100	C	1100
5	101	D	1101
6	110	E	1110
7	111	F	1111

(b) 十六进制/二进制转换表

图 1.1

如果二进制数有小数部分的话，首先要找到八进制的小数部分，从二进制的小数点开始，向右每 3 位分成一组。其每组 3 位数所对应的八进制值可以在转换表格中找到。例如：

$$(100\ 001\ 010\ 100\ .010)_2$$
$$= (4\quad 1\quad 2\quad 4\quad .2)_8$$

八进制数同样可以转换成二进制数，方法就是按照对应的表格，把每一个八进制数替换

成一个3位的二进制数。例如：

$$(4\ \ 3\ \ 2\ \ .7)_8$$
$$= (100\ 011\ 010\ .111)_2$$

相似地，每一个十六进制数可以用一个4位的二进制数表示，如图1.1（b）所示。将一个二进制数转换成十六进制数的时候，从小数点开始，向左和向右，分别把每4个数字分成组。然后按照下面的十六进制转换表格把每组的4个二进制数转换成十六进制数。例如：

$$(1011\ 1010\ 0011\ .0010)_2$$
$$= (B\ \ A\ \ 3\ \ .2)_{16}$$

在其逆转换中，每一个十六进制数将按照转换表，被相应的4个二进制数替代。例如：

$$(4\ \ F\ \ C\ \ 2)_{16}$$
$$= (0100\ 1111\ 1100\ 0010)_2$$

1.4　二进制加减法

一、加法

在图1.2中给出了两个一位二进制数的加法运算规则，下面有两个正的4位二进制数执行加法的例子用到了上述规则：

		2^3	2^2	2^1	2^0	
被加数		1	0	1	1	11
加数		0	1	1	1	+7
和	1	0	0	1	0	18
进位	1	1	1	1		

被加数	加数	和	进位
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

图1.2　两个二进制数字的加法规则

每个位置上的一对数字的权值都在其中表示出来了。可以看到从2^0到2^3列均产生了进位。在设计加法电路时，进位（carry ripple）应该被作为重要的因素加以考虑。

当最高位是1的两个n位二进制数相加的时候，产生的和是一个（$n+1$）位的数。产生的增加位被称作算术上溢（arithmetic overflou）。在用纸和笔计算的过程中增加的数字并没有问题，而在数字机中，在进行加法之前，被加数和加数被分别存放在两个不同的寄存器中，而它们计算出的和将被存放在其中一个寄存器中，在这种情况下，寄存器就必须预留一位，用来为计算结果存储上溢。

二、减法

在图1.3中总结了两个4位正数的减法运算规则，其中减数小于被减数，例如：

	2^3	2^2	2^1	2^0	
被减数	1	1	0	0	12
减数	0	0	1	1	−3
差	1	0	0	1	+9
借位		1	1		

被减数	减数	差	借位
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

图1.3　两个二进制数字的减法规则

如果减数大于被减数，将产生算术下溢，会在最高位产生借位，导致差为负数。而其中的借位可以被用于数字机的一个 1 位寄存器，它工作则表示差为负数。算术下溢的过程见下面的例子：

	2^3	2^2	2^1	2^0	
被减数	0	0	1	1	3
减数	1	1	0	0	−12
差	0	1	1	1	−9
借位	1	1			

在数字机中，减法被广泛的使用于大量的计算过程中，用于比较两个二进制数的大小。如果发生算术下溢，产生的借位表示减数大于被减数，否则，减数将等于或小于被减数。

1.5　带符号的算术

前面的章节中都是针对正数的运算，也只有在这种情况下，当减数大于被减数时，答案才是负数。所以在机器中，如何区别正数和负数成了一个十分重要的问题。有符号的数应运而生。采用在最高位（MSB）之前增添附加位的办法来区分正负数，负数的附加位是 1，正数的附加位是 0，因此有：

$(-23)_{10} = (1,0010111)_2$　　$(+23)_{10} = (0,0010111)_2$

$(-0)_{10} = (1,0000000)_2$　　$(+0)_{10} = (0,0000000)_2$

这被称为符号量级表示法。8 位符号整数可以表示的范围是−127～+127，其中 0 有两种表示方法。

因为可以执行符号量级表示法运算的逻辑电路设计比较复杂，所以很少采用。在实际应用中，大量的机器运算是用补码算术（complement arithmetic）执行的。

1.6　补 码 算 术

这是一种既实用又简单的技术，它可以使数字机在执行有符号数运算时，所需的硬件设备最简化。在实际应用中，采用补码运算，可以将减法的计算过程转化为加法运算。

在任何的数字系统中都有两种情况可以使用补码。在二进制系统中，它们分别是：(a) 2 的补码或称基数补码，以及 (b) 1 的补码或称降基补码。在十进制系统中，它们分别是：(a) 10 的补码或称基数补码，以及 (b) 9 的补码或者称为降基补码。在二进制系统中，采用 1 的补码增加了硬件设备执行的难度，不值得采用，所以在二进制有符号运算中一直应用 2 的补码表示法。

1.7　二进制数的补码表示

二进制数的基数补码由下面的等式定义：