普通高等教育“十一五”国家级规划教材

数字逻辑（第2版）

毛法尧　编著

高等教育出版社
Higher Education Press

普通高等教育“十一五”国家级规划教材

数字逻辑

（第2版）

毛法尧　编著

高等教育出版社

内容提要

本书第一版是“教育部面向21世纪教学内容和课程体系改革”的研究成果，被列入教育部面向21世纪课程教材、“九五”国家级重点教材。第二版被列为“十一五”国家级重点教材。

本书着重介绍数字系统逻辑设计的基本理论和方法，同时对数字技术的新成果和新方法作了适当介绍。

本书系统地阐述了以下内容：数制与编码，逻辑代数基础，组合逻辑电路和时序逻辑电路的分析与设计，数字系统设计，自动逻辑综合，逻辑模拟与测试，硬件描述语言 Verilog HDL，以及逻辑器件。

本书可作为计算机类、电子类、自动化类相关专业的教材，也可作为有关专业工程技术人员的参考书。

图书在版编目(CIP)数据

数字逻辑/毛法尧编著. —2版. —北京：高等教育出版社，2008.3

ISBN 978-7-04-023220-2

Ⅰ.数… Ⅱ.毛… Ⅲ.数字逻辑-高等学校-教材 Ⅳ.TP302.2

中国版本图书馆CIP数据核字(2008)第015416号

策划编辑 倪文慧 **责任编辑** 张海波 **封面设计** 于文燕 **责任绘图** 尹 莉

版式设计 张 岚 **责任校对** 姜国萍 **责任印制** 韩 刚

出版发行	高等教育出版社	**购书热线**	010-58581118
社 址	北京市西城区德外大街4号	**免费咨询**	800-810-0598
邮政编码	100011	**网 址**	http://www.hep.edu.cn
总 机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经 销	蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印 刷	北京中科印刷有限公司	**畅想教育**	http://www.widedu.com
		版 次	2000年6月第1版
开 本	787×1092 1/16		2008年3月第2版
印 张	23.5	**印 次**	2008年3月第1次印刷
字 数	530 000	**定 价**	29.30元

物料号 23220-00

再版前言

数字逻辑是计算机类、电子类、自动化类大学本科学生重要的专业基础课，是电子计算机的基础理论之一。

随着现代科学技术的发展，大规模和超大规模集成电路日新月异，促使数字系统设计发生了极大的变化，也对数字逻辑课程传统的教学体系、内容和方法提出了新的挑战。

数字逻辑是一门专业基础课，它既要分析经典的组合逻辑和时序逻辑的原理，又要兼顾数字器件的新发展及由此产生的有关数字系统设计的新技术和新方法。

在编写本教材时，我们力求在注重基础知识，介绍基本理论和方法同时，尽可能反映近些年来数字技术的新发展和新应用。使读者在掌握基础知识的同时，对新技术和新方法有较全面的认识和了解。

全书共十一章。第一章和第二章介绍数制、码制、逻辑代数和函数化简方法；第三～五章讨论组合逻辑电路和时序逻辑电路的分析和设计；第六章和第七章结合中、大和超大规模集成电路的应用讨论组合逻辑、时序逻辑和数字系统设计的方法；第八～十章探讨自动逻辑综合、逻辑模拟和测试，以及硬件描述语言等先进设计技术；第十一章介绍广泛使用的几种器件的结构和特点。

为便于读者学习和掌握，使认识过程符合客观规律，本书贯穿两条主线，一条是器件—电路—系统，另一条是基本理论—分析方法—设计方法。

本书着眼于培养读者分析问题和解决问题的能力以及逻辑思维的能力。对每一个逻辑问题的讨论，力求思路清晰，深入浅出，避免就事论事，以期达到举一反三的目的。

最后，对为本教材的编写提供支持的领导和同事表示诚挚的谢意。感谢高等教育出版社的编辑同志对教材出版的热情帮助及付出的辛勤劳动，感谢漆文琰、严璟同志在资料整理、图表制作、文字加工等方面所做的工作。

限于编者水平，书中疏漏和不足之处在所难免，殷切希望读者批评指正。

编　者

2007 年 10 月

于华中科技大学

第一版前言

数字逻辑是计算机科学与技术(类)大学本科学生重要的专业基础课,是电子计算机的基础理论之一。

本书对数字逻辑的核心内容作了全面系统的阐述。在阐释数字逻辑电路基本概念和原理的基础上,着重介绍数字系统逻辑设计的基本理论和方法,集中讨论逻辑设计的重点问题,突出分析方法和设计方法。

微电子技术的发展,为数字系统提供了大量廉价、种类繁多、功能各异的逻辑器件。然而,逻辑设计仍是集成电路研制和计算机设计不可缺少的一环。对于从事计算机研制、开发和应用的科学工作者来说,熟悉和掌握数字系统逻辑设计的理论和方法是十分重要的。

全书共十章:第一章和第二章介绍数制、码制、逻辑代数和函数化简;第三章至第五章讨论组合逻辑电路和时序逻辑电路的分析与设计;第六章和第七章结合中、大规模集成电路的应用论述组合逻辑、时序逻辑和数字系统设计的方法,第八章和第九章探讨自动逻辑综合、逻辑模拟与测试等先进设计技术;第十章介绍数字系统中广泛使用的几种器件的结构和特点。在本书编写过程中,力求既强调基础理论,又注重实际应用;既体现系统性,又突出重点;既有一定深度,又注意深入浅出。

限于编者水平,书中疏漏和不足之处在所难免,殷切希望读者批评指正。

编　者

1999年4月

于华中理工大学

目　录

1 第一章 数制与编码

计算技术的进步促进了科学技术和生产的飞跃发展。现在，计算机已广泛应用于科学与工程计算、数据和信息处理、过程和实时控制、计算机辅助设计和辅助制造以及人工智能等领域。计算机是数字系统中最常见、最有代表性的一种设备。

数字系统的特点是它所处理的信息都是离散元素，这些元素可以是各种数字、字母、算符及符号等。离散元素按不同方式排列可以表示各种信息，例如，字母 C，O，M，P，U，T，E 和 R 可组成单词 COMPUTER，数字 1999 表示一个确定的数，等等。

在数字系统中，信息的离散元素是以称为信号的物理量来表示的，电压和电流就是最常用的电信号。通常，数字系统的信号为“有”或“无”两个离散量，因此，称为二进制信号。由于只有导通和截止两种工作状态的电子器件能可靠地反映两个离散量，并且在工程上比较容易实现，加上人类的逻辑思维方式也倾向于二值，所以，数字系统常采用二进制信号。

信号的离散量有的是自然形成的，有的则是对连续过程加以量化后而得到的。例如，一份学生成绩单就是由离散量组成的，包括学生的学号、姓名、课程名称、成绩等。又如，科学技术工作者在进行科学研究工作时，常常要观察连续的过程，但仅将一些特定的数值记录下来并列成表，这样就将连续的信息离散并且量化了，表格中的每一个数字都已成为离散量。

数字系统处理的是离散元素，而这些离散元素通常以二进制形式出现，人们熟悉的十进制数不能被机器直接接收。因此，当人机通信时，需将十进制数转换成二进制数，以便机器接收。机器运算结束时，再将二进制数转换成十进制数。

本章主要讨论数字系统中数的表示方法。

1.1 进位计数制

1.1.1 十进制数的表示

用一组统一的符号和规则表示数的方法，称为进位计数制，简称数制。

原则上说，一个数可以用任何一种进位计数制来表示和运算。但不同数制其运算方法及难

易程度各不相同。选择什么样的进位计数制来表示数，对数字系统的性能影响很大。

在日常生活中，人们通常采用十进制数来计数，每位数可用 10 个数码之一来表示，即 0，1，2，3，4，5，6，7，8，9。十进制的基数为 10，基数表示进位计数制所具有的数字符号的个数，十进制具有的数字符号个数为 10。

十进制数的计算规则是由低位向高位进位时“逢十进一”，也就是说，每位累计不能超过 9，计满 10 就应向高位进 1。

当人们看到一个十进制数，如 632.45 时，就会立刻想到：这个数的最左位（即第一位）为百位（6 代表 600），第二位为十位（3 代表 30），第三位为个位（2 代表 2），小数点右边第一位为十分位（4 代表 4/10），第二位为百分位（5 代表 5/100）。这里百、十、个、十分之一和百分之一都是 10 的幂。在一个采用进位计数制表示的数中，不同数位上的固定常数称为“权”。十进制数 632.45 从左至右各位的权分别是：10^2，10^1，10^0，10^{-1}，10^{-2}。这样，632.45 按权展开的形式如下

$$632.45=6\times10^2+3\times10^1+2\times10^0+4\times10^{-1}+5\times10^{-2}$$

其中，等式左边的表示方法称为位置计数表示法，等式右边则是其按权展开表示法。

一般说来，对于任意一个十进制数 N，可用位置计数法表示如下

$$(N)_{10}=(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_{10}$$

也可用按权展开表示法表示如下

$$\begin{aligned}(N)_{10}&=a_{n-1}\times10^{n-1}+a_{n-2}\times10^{n-2}+\cdots+a_1\times10^1+a_0\times10^0+\\&\quad a_{-1}\times10^{-1}+a_{-2}\times10^{-2}+\cdots+a_{-m}\times10^{-m}\\&=\sum_{i=-m}^{n-1}a_i\times10^i\end{aligned}$$

其中，a_i 表示各个数字符号，为 0～9 这 10 个数码中的任意一个；n 为整数部分的位数；m 为小数部分的位数。

通常，对十进制数的表示，可以在数字的右下角标注 10 或 D。

1.1.2 二进制数的表示

数字系统中使用的进位计数制并不限于十进制，当进位基数为 2 时，称为二进制。在二进制中，只有 0 和 1 两个数码。二进制的计数规则是由低位向高位“逢二进一”，即每位计满 2 就向高位进 1，例如，$(1101)_2$ 就是一个二进制数。不同数位的数码表示的值不同，各位的权值是以 2 为底的连续整数幂，从右向左递增。

对于任意一个二进制数 N，用位置计数法表示为

$$(N)_2=(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_2$$

用按权展开法表示为

$$(N)_2 = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \cdots + a_1 \times 2^1 + a_0 \times 2^0 + a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i \times 2^i$$

其中，a_i 表示各个数字符号，为 0 或 1；n 为整数部分的位数；m 为小数部分的位数。

通常，对二进制数的表示，可以在数字右下角标注 2 或 B。

在数字系统中，常用二进制来表示数字和进行运算。因为它具有如下特点：

(1) 二进制数只有 0 和 1 两个数码，任何具有两个不同稳定状态的元件都可用来表示一位二进制数。例如，晶体管的导通和截止，脉冲信号的“有”和“无”等。

(2) 二进制运算规则简单。其运算规则如下：

加法规则：

0+0=0　　0+1=1

1+0=1　　1+1=0(同时向相邻高位进 1)

减法规则：

0−0=0　　0−1=1(同时向相邻高位借 1)

1−0=1　　1−1=0

乘法规则：

0×0=0　　0×1=0

1×0=0　　1×1=1

除法规则：

0÷1=0　　1÷1=1

下面举例说明。

例 1.1　计算1101+1011。

解

$$\begin{array}{r} 1101 \\ +)\ 1011 \\ \hline 11000 \end{array}$$

两个二进制数的加法运算和十进制数的加法运算相似，但采用“逢二进一”的法则，每位数累计到 2 时，本位就记为 0，同时向相邻高位进 1。

例 1.2　计算11101−10011。

解

$$\begin{array}{r} 11101 \\ -)\ 10011 \\ \hline 1010 \end{array}$$

二进制减法运算从低位起按位进行，在遇到 0 减 1 时，采用“借一当二”法则向相邻高位借 1，也就是本位为 2，同时相邻高位减 1。

例 1.3 计算1101×1001。

解

$$
\begin{array}{r}
1\ 1\ 0\ 1 \\
\times)\ 1\ 0\ 0\ 1 \\
\hline
1\ 1\ 0\ 1 \\
0\ 0\ 0\ 0 \\
0\ 0\ 0\ 0 \\
1\ 1\ 0\ 1 \\
\hline
1\ 1\ 1\ 0\ 1\ 0\ 1
\end{array}
$$

二进制数的乘法运算和十进制数的乘法运算相似，所不同的是对部分积进行累加时遵循“逢二进一”的原则。

例 1.4 计算10010001÷1011。

解

$$
\begin{array}{r}
1\ 1\ 0\ 1 \cdots\cdots \text{商} \\
1\ 0\ 1\ 1\overline{)1\ 0\ 0\ 1\ 0\ 0\ 0\ 1} \\
\underline{1\ 0\ 1\ 1} \\
1\ 1\ 1\ 0 \\
\underline{1\ 0\ 1\ 1} \\
1\ 1\ 0\ 1 \\
\underline{1\ 0\ 1\ 1} \\
1\ 0 \cdots\cdots \text{余数}
\end{array}
$$

二进制数的除法运算同十进制数的除法运算类似，但采用二进制数的运算规则。

(3) 二进制数只有两个状态，数字的传输和处理不容易出错，可靠性高。

(4) 二进制数的数码 0 和 1，可与逻辑代数中逻辑变量的值“假”和“真”对应起来。也就是说，可用一个逻辑变量来表示一个二进制数码。这样，在逻辑运算中就可以使用逻辑代数这一数学工具。

1.1.3 其他进制数的表示

二进制数运算规则简单，便于电路实现，它是数字系统中广泛采用的一种数制。但用二进制表示一个数时，所用的位数比用十进制数表示的位数多，人们读写很不方便，容易出错，不便记忆。因此，人们常采用八进制数和十六进制数，这两种数制不但容易书写和阅读，便于记忆，而且具有二进制数的特点，十分容易将它们转换成二进制数。

八进制数的基数是 8，采用的数码是 0，1，2，3，4，5，6，7。计数规则是从低位向高位“逢八进一”，对于相邻两位来说，高位的权值是低位权值的 8 倍。例如，数$(47.6)_8$就表示一个八进制数。由于八进制的数码和十进制前 8 个数码相同，为了便于区分，通常在八进制数字的右下角标注 8 或数字后边接 O。

十六进制数的基数为16，采用的数码是0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F。其中，A,B,C,D,E,F分别代表十进制数字10,11,12,13,14,15。十六进制的计数规则是从低位向高位“逢十六进一”，对于相邻两位来说，高位的权值是低位权值的16倍。例如，$(54AF.8B)_{16}$就是一个十六进制数。通常，在十六进制数字的右下角标注16或数字后边接H。

与二进制数一样，八进制数和十六进制数均可用位置计数法的形式和按权展开法的形式表示。

一般说来，对于任意的数N，都能表示成以r为基数的r进制数。数N的位置计数法为

$$(N)_r=(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_r$$

而相应的按权展开表示法为

$$\begin{aligned}(N)_r=&a_{n-1}\times r^{n-1}+a_{n-2}\times r^{n-2}+\cdots+a_1\times r^1+a_0\times r^0+\\&a_{-1}\times r^{-1}+a_{-2}\times r^{-2}+\cdots+a_{-m}\times r^{-m}\\=&\sum_{i=-m}^{n-1}a_ir^i\end{aligned}$$

其中，a_i表示各个数字符号，为0～$r-1$数码中任意一个；r为进位制的基数；n为整数部分的位数；m为小数部分的位数。

r进制的计数规则是从低位向高位“逢r进一”。

常用的不同数制的各种数码如表1.1所示，该表列出了当r为10,2,8,16时各种进位计数制中前16个数。

表1.1 不同进位计数制的各种数码

十进制数 ($r=10$)	二进制数 ($r=2$)	八进制数 ($r=8$)	十六进制数 ($r=16$)
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1.2 数制转换

1.2.1 二进制数与十进制数的转换

在计算机和其他数字系统中，普遍使用二进制数，采用二进制数的数字系统只能处理二进制数或用二进制代码形式表示的其他进位制数。而人们习惯于使用十进制数，所以，在信息处理中首先必须把十进制数转换成计算机能加工和处理的二进制数进行运算，然后再将二进制数的计算结果转换成人们习惯的十进制数。

二进制数转换成十进制数是很方便的，只要将二进制数写成按权展开式，并将式中各乘积项的积算出来，然后各项相加，即可得到与该二进制数相对应的十进制数。例如

$$\begin{aligned}(11010.101)_2 &= 1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0+\\ &\quad 1\times2^{-1}+0\times2^{-2}+1\times2^{-3}\\ &= 16+8+2+0.5+0.125\\ &= (26.625)_{10}\end{aligned}$$

十进制数转换成二进制数时，需将待转换的数分成整数部分和小数部分，并分别加以转换。一个十进制数可写成

$$(N)_{10}=\langle\text{整数部分}\rangle_{10}.\langle\text{小数部分}\rangle_{10}$$

转换时，首先将$\langle$整数部分$\rangle_{10}$转换成$\langle$整数部分$\rangle_2$，然后再将$\langle$小数部分$\rangle_{10}$转换成$\langle$小数部分$\rangle_2$。待整数部分和小数部分确定后，就可写成

$$(N)_2=\langle\text{整数部分}\rangle_2.\langle\text{小数部分}\rangle_2$$

1. 整数转换

十进制数的整数部分采用“除 2 取余”法进行转换，即用十进制数除以 2，取余数 1 或 0 作为相应二进制数的最低位，把得到的商再除以 2，取余数 1 或 0 作为二进制数的次低位，依次类推，继续上述过程，直至商为 0，所得余数为最高位。

例如，为将十进制整数 58 转换为二进制整数，就要把它写成如下形式

$$\begin{aligned}(58)_{10} &= (a_{n-1}a_{n-2}\cdots a_1a_0)_2\\ &= a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}+\cdots+a_1\times2^1+a_0\times2^0\\ &= 2(a_{n-1}\times2^{n-2}+a_{n-2}\times2^{n-3}+\cdots+a_1)+a_0\end{aligned}$$

只要求出等式中的各个系数 $a_{n-1}, a_{n-2}, \cdots, a_1, a_0$，便得到二进制整数。将上式两边除以 2，得

$$(29)_{10}=a_{n-1}\times2^{n-2}+a_{n-2}\times2^{n-3}+\cdots+a_1+\frac{a_0}{2}$$

两数相等，整数部分和小数部分必须对应相等，等式左边余数为 0，则 a_0 为 0。因而得

$$(29)_{10}=2(a_{n-1}\times 2^{n-3}+a_{n-2}\times 2^{n-4}+\cdots+a_2)+a_1$$

将等式两边再除以 2，得

$$(14+\frac{1}{2})_{10}=a_{n-1}\times 2^{n-3}+a_{n-2}\times 2^{n-4}+\cdots+a_2+\frac{a_1}{2}$$

比较等式两边，等式左边余数为 1，则取 a_1 为 1。

依次类推，可得系数 $a_2,a_3,\cdots,a_{n-2},a_{n-1}$。

根据上面讨论的方法，可用下列形式很方便地将十进制整数转换成二进制整数

2⌊58

2⌊29　　余数 0(a_0)最低位

2⌊14　　余数 1(a_1)

2⌊ 7　　余数 0(a_2)

2⌊ 3　　余数 1(a_3)

2⌊ 1　　余数 1(a_4)

0　　余数 1(a_5)最高位

因此，$(58)_{10}=(111010)_2$。

2. 纯小数转换

十进制数的小数部分采用“乘 2 取整”法进行转换，即先将十进制小数部分乘以 2，取其整数 1 或 0 作为二进制小数的最高位；然后将乘积的小数部分再乘以 2，并再取整数作为次高位。重复上述过程，直到小数部分为 0 或达到所要求的精度。

例如，将十进制小数 0.625 转换为二进制小数，需把它写成如下形式

$$\begin{aligned}(0.625)_{10}&=(0.a_{-1}a_{-2}\cdots a_{-m})_2\\&=a_{-1}\times 2^{-1}+a_{-2}\times 2^{-2}+\cdots+a_{-m}\times 2^{-m}\\&=\frac{a_{-1}}{2}+\frac{1}{2}(a_{-2}\times 2^{-1}+\cdots+a_{-m}\times 2^{-m+1})\end{aligned}$$

只要求出各系数 $a_{-1},a_{-2},\cdots,a_{-m}$，便得到二进制小数。

将上式两边乘以 2，得

$$(1.25)_{10}=a_{-1}+(a_{-2}\times 2^{-1}+\cdots+a_{-m}\times 2^{-m+1})$$

根据两个数相等，其整数部分和小数部分必须分别相等的道理，a_{-1} 等于左边的整数，则 a_{-1} 为 1。

等式右边括号内的数仍为小数，因而

$$(0.25)_{10}=\frac{a_{-2}}{2}+\frac{1}{2}(a_{-3}\times 2^{-1}+\cdots+a_{-m}\times 2^{-m+2})$$

再将等式两边乘以 2，得

$$(0.5)_{10}=a_{-2}+(a_{-3}\times 2^{-1}+\cdots+a_{-m}\times 2^{-m+2})$$

比较等式两边的整数，故取 a_{-2} 为 0。如此反复乘以 2，直到小数部分等于 0，即可求得系数 a_{-1}，a_{-2}，…，a_{-m}。

根据上面讨论的方法，可用下列形式很方便地将十进制小数转换成二进制小数

```
    0.625
×)      2
 [1].250     整数 1(a-1)最高小数位
×)      2
 [0].500     整数 0(a-2)
×)      2
 [1].000     整数 1(a-3)最低小数位
```

因此，$(0.625)_{10}=(0.101)_2$。

必须指出：运算时，式中的整数不参加连乘。

在十进制的小数部分转换中，有时连续乘以 2 不一定能使小数部分等于 0，这说明该十进制小数不能用有限位二进制小数表示。这时，只要取足够多的位数，使其误差达到所要求的精度就可以了。下面举例说明。

例 1.5 将十进制数0.18转换成二进制数，精确到小数点后 4 位。

解

```
    0.18
×)      2
 [0].36      整数 0(a-1)
×)      2
 [0].72      整数 0(a-2)
×)      2
 [1].44      整数 1(a-3)
×)      2
 [0].88      整数 0(a-4)
×)      2
 [1].76      整数 1(a-5)
```

十进制数 0.18 连续 4 次乘以 2 后，其小数部分等于 0.88，仍不为 0。由于要求精确到小数点后 4 位，因此将 0.88 再乘以 2，小数点后第 5 位为 1，得