高等学校
工科电子类 规划教材

数字逻辑

毛法尧 欧阳星明 任宏萍

华中科技大学出版社

数 字 逻 辑

毛法尧　欧阳星明　任宏萍

华中科技大学出版社
（华中理工大学出版社）

图书在版编目(CIP)数据

数字逻辑/毛法尧 等编
武汉:华中科技大学出版社, 1996年9月
ISBN 7-5609-1374-1

Ⅰ. 数…
Ⅱ. ①毛… ②欧阳… ③任…
Ⅲ. 数字-逻辑-教材
Ⅳ. TP33

数字逻辑

毛法尧 欧阳星明 任宏萍

责任编辑:黄以铭

*

华中科技大学出版社出版发行
(武昌喻家山 邮编:430074)
华中科技大学出版社照排室照排
华中科技大学印刷厂印刷

*

开本:787×1092 1/16 印张:19.25 字数:462 000
1996年9月第1版 2004年9月第19次印刷
ISBN 7-5609-1374-1/TP·190
定价:21.80元

内 容 简 介

本教材系统地介绍数字系统逻辑设计的基本理论和方法，着重阐述数字逻辑电路的一般原理，并反映数字逻辑电路的新发展。

本教材适用于高等学校计算机及应用、计算机软件、自动控制、通信等专业，也可供有关工程技术人员参考。

出版说明

根据国务院关于高等学校教材工作的规定，我部承担了全国高等学校和中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978～1990年，已编审、出版了三个轮次教材，及时供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻国家教委《高等教育“八五”期间教材建设规划纲要》的精神，“以全面提高教材质量水平为中心，保证重点教材，保持教材相对稳定，适当扩大教材品种，逐步完善教材配套”，作为“八五”期间工科电子类专业教材建设工作的指导思想，组织我部所属的八个高等学校教材编审委员会和四个中等专业学校专业教学指导委员会，在总结前三轮教材工作的基础上，根据教育形势的发展和教学改革的需要，制订了1991～1995年的“八五”(第四轮)教材编审出版规划。列入规划的，以主要专业主干课程教材及其辅助教材为主的教材约300余种。这批教材的评选推荐和编审工作，由各编委会或教学指导委员会组织进行。

这批教材的书稿，其一是从通过教学实践、师生反应较好的讲义中经院校推荐，由编审委员会(小组)评选择优产生出来的，其二是在认真遴选主编人的条件下进行约编的，其三是经过质量调查在前几轮组织编写出版的教材中修编的。广大编审者、各编审委员会(小组)、教学指导委员会和有关出版社，为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还可能有缺点和不足之处，希望使用教材的单位、广大教师和同学积极提出批评和建议，共同为不断提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

前　言

本教材系按电子工业部的工科电子类专业教材1991～1995年编审出版规划，由工科电子类教材编审委员会计算机教材编审小组征稿并推荐出版。责任编委为谢希仁。

本教材由华中理工大学毛法尧，欧阳星明，任宏萍编。

数字逻辑属计算机科学与工程(类)本科重要的专业基础课，课程的参考学时为70～80学时。本课程的目的是使学生了解和熟悉从对数字系统提出要求开始，一直到用集成电路实现所需逻辑功能为止的整个过程。熟练掌握数字系统逻辑设计的理论和方法，对于从事计算机研制、开发和应用的科技工作者来说是十分必要的。

全教材共分九章。第一章数制与码制，介绍数字系统中常用的数制及转换、码制和编码。第二章逻辑代数基础，介绍逻辑代数、逻辑函数及函数化简。第三章组合逻辑电路，着重介绍组合逻辑电路的分析和设计方法。第四章同步时序逻辑电路，介绍同步时序逻辑电路的一般分析和设计方法。第五章异步时序逻辑电路，讨论脉冲异步时序逻辑电路和电平异步时序逻辑电路的分析和设计。第六章采用中、大规模集成电路的逻辑设计，介绍用中、大规模集成电路进行逻辑设计的特点和一般方法。第七章数字系统设计，仅对数字系统逻辑设计的方法和步骤进行探讨。第八章计算机辅助逻辑设计，简单介绍计算机辅助逻辑设计这一先进设计技术。第九章逻辑器件，介绍目前广泛使用的几种器件的结构和特点。

本教材由毛法尧编写第一、七、八、九章，欧阳星明编写第二、三、六章，任宏萍编写第四、五章，毛法尧统编全稿。在编写过程中得到了兄弟院校同行的热忱关怀，并提出了许多宝贵意见，在这里表示诚挚的感谢。由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编　者

1995.10于华中理工大学

目　　录

第 1 章　数制与码制

计算技术的发展促进了科学技术和生产的飞跃发展。现在，计算机已广泛地应用于科学计算、数据处理和生产过程控制等领域中。它是数字系统中最常见的、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。这些离散元素可以是十进制数字、某种字母、各种算符及标点符号等。各种离散元素按不同方式排列可以表示大量的信息。例如，字母 C，O，M，P，U，T，E 和 R 可形成单词 COMPUTER，数字 1984 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，数字系统的信号只有有或无两个离散量，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十分可靠地反映两个离散量，且在工程上比较容易实现，再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，有的则是将连续过程有意加以量化后而得到的。例如，一份学生成绩单就是天然的离散过程，上面有学生的姓名、课程名称、成绩等。又如，科学工作者在科学研究工作中常常要观察连续的过程，但仅将一些特殊的数值记录下来列成表格，这样就将连续的信息量化了，所以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数的形式出现，人们熟悉的十进制数不能被机器直接接受，因此，当人机通信时，则需要将十进制数转换成二进制数，以便机器接受。机器运算结束时，将二进制数再转换成十进制数。为此，必须讨论代码特征和运算，及各种数制的转换。

本章主要讨论数字系统中数的表示方法。首先讨论不同的进位计数制及其相互转换；然后讨论二进制数在计算机中的表示方法，包括数的符号、数值、小数点的表示和数的运算。另外，还将介绍计算机中常用的几种代码。

1.1　进位计数制

1.1.1　十进制数的表示

在日常生活中，人们通常采用十进制数来计数，每位数可用下列十个数码之一来表示，即 0，1，2，3，4，5，6，7，8，9。十进制的基数为 10，基数表示进位制所具有的数字符号个数，十进制具有的数字符号个数为 10。

十进制数的计算规律是由低位向高位进位“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)，小数点右面第一位为十分位(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 的次幂，它取决于系数所在的位置，称之为“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：

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

等式左边的表示方法称之为位置记数法，等式右边则是其按权展开式。

一般说来，对于任意一个十进制数 N，可用位置记数法表示如下：

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

也可用按权展开式表示如下：

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

式中，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

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

用按权展开式表示为

$$\begin{aligned}(N)_2 &= a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}+\cdots+a_1\times2^1+a_0\times2^0+a_{-1}\times2^{-1}+a_{-2}\times2^{-2} \\ &\quad +\cdots+a_{-m}\times2^{-m} \\ &= \sum_{i=-m}^{n-1}a_i\times2^i\end{aligned}$$

式中，a_i 表示各个数字符号为数码 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。

1.1.3 任意进制数的表示

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。因此，常采用八进制或十六进制。

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”，相邻两位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，所以为了便于区分，通常在数字的右下角标注 8 或 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”，相邻两位高位的权值是低位权值的 16 倍。例如，数$(54AF.8B)_{16}$就是一个十六进制数。通常，在数字的右下角标注 16 或 H。

与二进制数一样，任意一个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式的形式表示。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 的表示方法也有两种形式，即

位置记数法：　　$(N)_r=(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_r$

按权展开式：

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

式中，a_i 表示各个数字符号为 $0\sim r-1$ 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”。

不同数制的各种数码见表 1.1，该表列出了当 r 为 10,2,8,16 时，各种进位计数制中开头的 16 个自然数。

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

1.1.4 二进制的特点

对于一个数，原则上说，可以用任何一种进位计数制来计数和运算。但不同数制，其运算方法及难易程度各不相同。因此，选择什么样的进位计数制来表示数，对数字系统的成本和性能影响很大。在数字系统中，常用二进制来表示数字和进行运算。这是由于它具有如下特点：

(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 1 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。其运算规则是：

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

$0\times0=0$　　$0\times1=0$

$1\times0=0$　　$1\times1=1$

除法规则

$0\div1=0$　　$1\div1=1$

例 1.1　进行 1101+1011 运算。

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，且向相邻高位进 1。

例 1.2　进行 11101−10011 运算。

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，就要采用"借一当二"法则向相邻高位借 1，也就是从那一位减去 1。

例 1.3　进行 1101×1011 运算。

解

$$\begin{array}{r} 1101 \\ \times)\ 1011 \\ \hline 1101 \\ 1101 \\ 0000 \\ 1101 \\ \hline 10001111 \end{array}$$

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要按“逢二进一”的原则。

例 1.4 进行 $10010001 \div 1011$ 运算。

解

```
                 1101…      商
       1011 √10010001
              1011
              1110
              1011
                1101
                1011
                  10…      余数
```

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)二进制数的数码 0 和 1，可与逻辑代数中逻辑变量的“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，在逻辑运算中可以使用逻辑代数这一数学工具。

二进制数的缺点是书写长，不便记忆和阅读。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。

1.2 数制转换

在计算机和其它数字系统中，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它进位制数。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。这里就存在一个不同数制的相互转换问题。

1.2.1 二进制数和十进制数的转换

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

$$\begin{aligned}(11010.101)_2 &= 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 \\ &\quad + 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} \\ &= 16 + 8 + 2 + 0.5 + 0.125 \\ &= (26.626)_{10}\end{aligned}$$

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。将一个十进制数写成

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

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

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

一、整数转换

十进制数的整数部分采用"除 2 取余"法进行转换，即把十进制整数除以 2，取出余数 1 或 0 作为相应二进制数的最低位，把得到的商再除以 2，再取余数 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，要将十进制整数 58 转换为二进制整数，就要把它写成如下形式：

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

只要求出等式中的各个系数 $a_{n-1},a_{n-2},\cdots,a_1,a_0$，便得到二进制数。

将上式两边除以 2，得

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

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，则取 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

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

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，$(58)_{10}=(111010)_2$。

二、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。