高 等 学 校 教 材

数字逻辑与数字系统基础

沈建国　雷剑虹　主编

高等教育出版社

内容提要

全书共分9章，前三章论述数字逻辑的基本原理，讨论了组合逻辑电路和时序逻辑电路的工作原理及其设计思想；第四章简述了模数和数模转换的工作原理；第五章介绍了基本脉冲电路工作原理；第六至第九章讨论目前应用广泛的PAL、GAL、FPGA和isp的组成结构、基本的ABEL_HDL硬件描述语言的使用及编程技术；第八、九章介绍数字系统设计的基础及Verilog-HDL语言的设计方法。非电类、师范类物理专业以及电类专业可根据不同要求选择相关章节进行教学。本书也可供相关工程技术人员参考。

图书在版编目(CIP)数据

数字逻辑与数字系统基础/沈建国，雷剑虹主编.
北京：高等教育出版社，2004.1
ISBN 7-04-013048-3

Ⅰ.数… Ⅱ.①沈…②雷… Ⅲ.①数字逻辑-高等学校-教材②数字系统-高等学校-教材 Ⅳ.TP302.2

中国版本图书馆CIP数据核字(2003)第101993号

出版发行	高等教育出版社	购书热线	010-64054588
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	010-82028899		http://www.hep.com.cn

经　　销　新华书店北京发行所
排　　版　高等教育出版社照排中心
印　　刷

开　　本	787×960　1/16	版　　次	年　月第1版
印　　张	26.75	印　　次	年　月第　次印刷
字　　数	500 000	定　　价	33.20元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。
版权所有　侵权必究

郑 重 声 明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：（010）58581897/58581698/58581879/58581877

传　　真：（010）82086060

E - mail：dd@ hep. com. cn 或 chenrong@ hep. com. cn

通信地址：北京市西城区德外大街 4 号
　　　　　高等教育出版社法律事务部

邮　　编：100011

购书请拨打电话：（010）64014089　64054601　64054588

此为试读，需要完整PDF请访问：www.ertongbook.com

前　言

数字电路课程是电子工程、自动化技术、计算机等电类专业和机电一体化等非电类专业的主要技术基础课，同时也是理科、师范类物理专业学生的必修课。

现代科学技术的发展日新月异，电子技术的新理论、新技术、新器件更是层出不穷。大规模和超大规模集成电路（LSI/VLSI）、通用逻辑电路（GAL）、现场可编程逻辑电路（FPGA）及在系统可编程器件（isp）的出现，使逻辑电路的设计逐步从选择通用的标准 74 系列逻辑器件向通用逻辑电路 GAL 或现场可编程逻辑电路 FPGA 转移。数字电路器件的更新换代，一方面使数字系统的设计发生极大的变化，另一方面也给传统的‘数字电路’课程教学体系、教学内容、教学方法和任课老师提出了新的挑战。编写本书的目的是为学生提供数字电路设计、分析的基础知识。

数字逻辑与数字系统基础是一门专业基础课，本书的主要读者是大学二年级的学生，它既要分析经典的组合逻辑与时序逻辑的原理，又要兼顾数字器件的新发展及其相关的软件工具。新的内容，既不是作为陪衬，也不是资料的搬迁，而是以完整的体系和较完备的分析设计实例献给读者，并尽可能的深入浅出，使读者容易接受。编写本教材中，我们力求在论述基本原理时突出重点，精简内容，给新技术和新知识点较多空间。使学生在掌握基础知识的同时，对新技术有较全面的认识和了解。有利于培养学生在较短时间内掌握和使用新技术的能力。

本书共分九章，前三章论述数字逻辑的基本原理，讨论了组合逻辑电路和时序逻辑电路的工作原理及其设计思想；第四章简述了模数和数模转换的工作原理；第五章介绍了基本脉冲电路的工作原理；第六到第九章讨论目前应用广泛的 PAL、GAL 和 isp 的组成结构、工作原理、基本的 ABEL－HDL 硬件描述语言的使用及编程技术。第八、第九章介绍 Verilog－HDL 语言的设计方法及数字系统设计的基础。

本教材的使用，根据不同学校的特点及条件可分二种情况：

1. 非电类、师范类物理系以数字逻辑电路为主，可编程逻辑作一般性介绍，第一章用 14～18 学时，第二章用 10～14 学时，第三章用 20～24 学时，第四和第五章各用 6～8 学时，总学时为 56～72 学时，第六到第九章可按各学校具体情况处理；

2. 对于电类学生，第一章 14 学时，第二章 8 ~ 10 学时，第三章 18 学时，第四章 6 学时，第五章 4 学时，第六章和第七章各 6 学时，第八章 4 学时，第九章 4 学时，总学时为 72 学时。

以上安排不包括实验课时。

参加本书编写的有沈建国教授、雷剑虹讲师、徐力平副教授、金之诚讲师。其中第一、二、三、四、五章主要由沈建国教授执笔，第六、七、八、九章主要由雷剑虹讲师执笔，全部书稿由沈建国教授统稿。

浙江大学电气工程学院王小海教授、祁才君教授审阅全部书稿，提出了不少宝贵意见及建议。

华东师范大学电子系数字系统教研组的老师对本书提出不少建议，孟杰、王祖杰、奚慧婷等作了不少工作，在此表示感谢。

由于作者水平有限，书中会有不少不恰当之处，恳请使用本书的读者提出宝贵意见。

作　者

2003 年 8 月

目　　录

此为试读,需要完整PDF请访问: www.ertongbook.com

第一章　逻辑代数和逻辑门电路

内 容 提 要

本章主要论述数字逻辑关系的基本原理和实现数字逻辑关系的基本门电路及其外特性。

首先简单介绍数字系统中数的表示方法，不同计数制及其相互转换。然后讨论逻辑代数的基本公式及定理，重点讲述应用逻辑代数化简逻辑函数的方法——代数法和几何(卡诺图)法。最后介绍实现数字逻辑关系的具体电路——逻辑门电路及其外特性。

1.1　数制与编码

在日常生活中，常用十进制数表示一个数值的大小，但在计算机及数字系统中通常采用二进制数表示一个值的大小，有时也采用八进制数和十六进制数表示一个数的值，数制是计数进位制的简称。

1.1.1　数制

一、十进制数

在十进制数中有 0、1、2、3、4、5、6、7、8、9 等十个数字符号，表示 0～9 的数值，当一个数大于 9 时，向相邻高位进 1，以 10 表示。即 $9+1=10$，称作为逢十进一。在十进制数中虽然只有 0～9 十个数字符号，但经适当组合可以表现任何有限的数值。例如，某数三佰贰拾五点二可以写成 325.2，也可用多项式表示：

$$\begin{aligned}325.2 &= 3\times10^{(3-1)}+2\times10^{(2-1)}+5\times10^{(1-1)}+2\times10^{-1}\\ &=300+20+5+0.2 \qquad (1.1.1)\end{aligned}$$

在小数点左边第一位乘 10^0，第二位乘 10^1，第三位乘 10^2，而小数点右面第一位乘 10^{-1}。可见，数符在数中所处的位置不同，数符再乘上 10 的 i 次方，就表示该数符在其数值中的大小。例如，在式(1.1.1)中小数点后的 2 其值为 0.2，小数点左面第二位的 2 其值是 20。i 是数符在某数的第 n 位减一的值，10 的 i

此为试读，需要完整PDF请访问：www.ertongbook.com

次方称作为“权”，对于一个十进制数 n 的一般表示如下所示：

$$(N)_{10}=d_{n-1}\times10^{n-1}+d_{n-2}\times10^{n-2}+\cdots+d_1\times10+d_0\times10^0+d_{-1}\times10^{-1}+\cdots+d_{-m}\times10^{-m}$$

$$=\sum_{i=-m}^{n-1}d_i\times10^i \qquad (1.1.2)$$

式中 n 表示整数部分的位数；m 表示小数部分的位数；10^i 表示第 i 位的权；d_i 表示 i 位的数符。通常我们把(1.1.2)式称作为按权展开式或多项式之和。对于一个数，理论上讲可以采用任何一种进位制表示。但是，对于电子电路而言，若选用十进制数，则很难找到一个有十种状态的简单电路与十个数符相对应，所以在数字电路中不用十进制数表示一个数值。

二、二进制数

二进制数只有 0 和 1 两个数字符号，分别代表 0 和 1 两个数值。当数值大于 1 时，则向高位进 1，即 $1+1=10$，称作“逢二进一”。二进制数也可以用多项式的和来表示，例如 $(10\ 111.01)_2$ 可表示为：

$$(10\ 111.01)_2=1\times2^4+0\times2^3+1\times2^2+1\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}$$

其整数部分最高位权是 2^4，小数部分最低位权是 2^{-2}。可见，二进制数同样可以表示任意一个数值。一般表达式如下所示：

$$(N)_B=\sum_{i=-m}^{n-1}b_i\times2^i=b_{n-1}b_{n-2}\cdots b_1b_0\cdot b_{-1}b_{-2}\cdots b_{-m}$$

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

$$+b_{-2}\times2^{-2}+\cdots+b_{-m}\times2^{-m} \qquad (1.1.3)$$

(1.1.3)式中 $-m$ 是小数点后的位数，n 是小数点前的位数。$(N)_B$ 的下标 B(Binary)表示二进制。

由于二进制数只有二个字符，因此它的每 1 位数都可以用具有二个不同稳定状态的电路来实现，例如晶体管电路中的饱和及截止，开关电路的开和关，继电器的接通与断开等，因而二进制数在计算机技术中有广泛的应用。

但是，用二进制数表示一个数值时，位数较长，例如 $(10\ 111.01)_2$ 要用 7 位二进制数，若用十进制数表示该数 $(23.25)_{10}$ 只需 4 位十进制数。可见，使用二进制数时不够方便也不习惯。因而，在计算机技术中常采用十六进制数和八进制数。

十六进制采用十六个数符：0、1、2、3、4、5、6、7、8、9、A(相当于十进制的 10)、B(11)、C(12)、D(13)、E(14)、F(15)，是“逢十六进一”。例如十六进制的 AE4 可用多项式表示如下：

$$(AE4)_H=A\times16^2+E\times16^1+4\times16^0=A\times16^2+E\times16+4 \qquad (1.1.4)$$

(1.1.4)式中下标 H(Hexadecimal)表示十六进制。

同理，八进制数有 0、1、2、3、4、5、6、7 等八个数符，是“逢八进一”，用下标

O(Octal)表示，例如：

(73)$_O$ 表示八进制。

$$(73)_O = 7\times 8^1 + 3\times 8^0 = 7\times 8 + 3$$

三、不同进制之间的转换

由于人们日常习惯使用十进制数，但在计算机和数字系统内部大多使用二进制数或用二进制为基础的其他进位制数。因此，就需要在机器的输入或输出端对不同进制的数之间进行转换。

1. 二进制数转换为十进制数

二进制数转换为十进制数可按下述方法进行：首先计算出(1.1.3)式中各项的积，然后求和，就可得到该二进制数对应的十进制数，我们称作为按权展开求和法。例如：

例 1.1.1 求$(10\ 111.01)_2$对应的十进制数值。

解：
$$\begin{aligned}(10\ 111.01)_2 &= 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2}\\ &= 16 + 4 + 2 + 1 + 0.25\\ &= (23.25)_{10}\end{aligned}$$

2. 十进制数转换成二进制数

十进制数转换成二进制数，分整数部分和小数部分，采取不同的方法加以转换。

整数部分的转换采用‘除二取余法”，即把十进制数除以 2，取出余数 1 或 0 作为相应的二进制数的最低位，把商再除以 2，取余数 1 或 0，作相应的二进制数的第 2 位，依次类推，直到最后商为零为止。这时所得余数为最高位。

例 1.1.2 把$(71)_{10}$转换为二进制数。

解：

		余数
2	71	
2	35	1
2	17	1
2	8	1
2	4	0
2	2	0
2	1	0
	0	1

所以：$(71)_{10} = (1000111)_2$

小数部分的转换采用‘乘 2 取整’法，即把小数部分乘以 2，取其积的整数 1 或 0，作小数部分的最高位，然后取积的小数部分再乘以 2 取其积的整数 1 或 0，作整数部分的第 2 位，依次类推，直到积等于零或所需精度为止。

例 1.1.3 把$(0.375)_{10}$转换为二进制。

解：

$$
\begin{array}{r}
0.375 \\
\times \quad 2 \\
\hline
0.750
\end{array}
\quad \text{整数部分为 0，即小数部分最高位是 0}
$$

$$
\begin{array}{r}
\times \quad 2 \\
\hline
1.50
\end{array}
\quad \text{整数部分为 1，即小数部分第 2 位是 1}
$$

$$
\begin{array}{r}
\times \quad 2 \\
\hline
1.0
\end{array}
\quad \text{整数部分为 1，即小数部分第 3 位是 1}
$$

所以：$(0.375)_{10} = (0.011)_2$

必须指出，在‘乘 2 取整’运算中，积的整数部分不再参加运算。

在十进制小数转换到二进制时，往往连乘 2 以后，小数部分不等于 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，只有取一定量的位数，以保证误差在要求的精度范围之内。

例 1.1.4 将十进制数 0.13 转换为二进制数，精确到小数点后 4 位。

解：

$$
\begin{array}{r}
0.13 \\
\times \quad 2 \\
\hline
0.26
\end{array}
\quad \text{整数部分为 0，即小数点后第 1 位是 0}
$$

$$
\begin{array}{r}
\times \quad 2 \\
\hline
0.521
\end{array}
\quad \text{整数部分为 0，即小数点后第 2 位是 0}
$$

$$
\begin{array}{r}
\times \quad 2 \\
\hline
1.04
\end{array}
\quad \text{整数部分为 1，即小数点后第 3 位是 1}
$$

$$
\begin{array}{r}
\times \quad 2 \\
\hline
0.08
\end{array}
\quad \text{整数部分为 0，即小数点后第 4 位是 0}
$$

$$
\begin{array}{r}
\times \quad 2 \\
\hline
\end{array}
$$

可见，十进制数 0.13 连续四次乘以 2 以后，其小数部分是 0.08，小于 0.5，根据‘四舍五入’的原则，第 4 位为 0，所以：$(0.13)_{10} \approx (0.0010)_2$。

例 1.1.5 把 $(0.628)_{10}$ 转换到二进制，精确到小数点后 8 位。

解：

$$
\begin{array}{rl}
0.628 & \\
\times \quad 2 & \\
\hline
1.256 & \text{-------------------- 1 ----- } b_{-1} \\
\times \quad 2 & \\
\hline
0.512 & \text{-------------------- 0 ----- } b_{-2} \\
\times \quad 2 & \\
\hline
1.024 & \text{-------------------- 1 ----- } b_{-3} \\
\times \quad 2 & \\
\hline
0.048 & \text{-------------------- 0 ----- } b_{-4} \\
\times \quad 2 & \\
\hline
0.096 & \text{-------------------- 0 ----- } b_{-5}
\end{array}
$$

$$\begin{array}{r} \times \quad 2 \\ \hline 0.192 \end{array} \text{- - - - - - - - - - - - - - - - - - 0 - - - - -} b_{-6}$$

$$\begin{array}{r} \times \quad 2 \\ \hline 0.384 \end{array} \text{- - - - - - - - - - - - - - - - - - 0 - - - - -} b_{-7}$$

$$\begin{array}{r} \times \quad 2 \\ \hline 0.768 \end{array} \text{- - - - - - - - - - - - - - - - - - 0 - - - - -} b_{-8}$$

对 0.628 连八次乘 2 以后，其小数部分是 0.768，根据“四舍五入”的原则，第 8 位不是 0 而取 1。所以 $(0.628)_{10} = (0.10100001)_2$，其误差 $e < 2^{-9}$。

若一个十进制数既有整数部分又有小数部分，转换时整数部分采用“除 2 取余”法，小数部分采用“乘 2 取整”法，然后把转换的结果合并起来即可。

例 1.1.6　将 $(71.375)_{10}$ 转换为二进制数。

解：$(71.375)_{10} = (71)_{10} + (0.375)_{10} = (1000111)_2 + (0.011)_2 = (1000111.011)_2$

3. 二进制数转换成十六进制数

二进制数转换成十六进制数可按下述方法进行，以小数点为基准分别向左向右按 4 位分组，最后不满 4 位则需加 0，每组以十六进制数符代替即可。

例 1.1.7　将 $(100111.11)_2$ 转换为十六进制数

解：$(100111.11)_2 = 10,0111.1100 = (27.B)_{16}$

二进制数转换成八进制数的方法与二进制数转换为十六进制数的方法相似，它是以 3 位分组，每组以八进制数符代替。

4. 十六进制数转换成二进制数

例 1.1.8　把 $(AE3.3)_{16}$ 转换成二进制数

解：$(AE3.3)_{16} = (101011100011.0011)_2$

例 1.1.9　把十六进制数 $(38A.B)_{16}$ 转换成二进制数

解：$(38A.B)_{16} = (001110001010.1100)_2 = (1110001010.11)_2$

需要注意的是，在对十六进制数每一位用二进制数表示时，整数部分 1 前面的 0 和小数部分后面的 0 可以省略。

现将十进制、二进制、八进制、十六进制的对照表列于表 1.1.1。

表 1.1.1　几种常见的数制对照表

十进制数	二进制数	八进制数	十六进制数
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5

续表

十进制数	二进制数	八进制数	十六进制数
6	0110	6	6
7	0111	7	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
16	10000	20	10

1.1.2 编码

前面讨论的数都没有考虑符号，一般认为是正数，对于带负号的数在机器中怎样表示，是本节所讨论的内容。

一个带符号的数由二部分组成，一部分表示该数的符号，另一部分表示该数的值。符号有正和负两种，它可以用 1 位二进制数表示。通常用 0 表示正数，用 1 表示负数，放在数值位的最高位前面。对于一个 n 位二进制数，若第一位是符号位，则余下的 $n-1$ 位是该数值的绝对值。一般直接用正号‘ + ’和负号‘ − ”表示符号的二进制数，如 +1101 和 −1101，称作符号数的真值，不能直接用于计算机和数字系统中。符号数值化以后，称作机器数，可以在计算机和数字系统中使用。

例如，二进制正数 1111 在机器中表示如图 1.1.1(a)所示，二进制负数 −1101在机器中表示如图 1.1.1(b)所示。

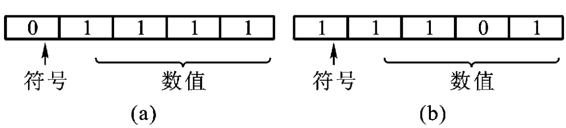

图 1.1.1　二进制数在机器中的表示

在数字系统中，表示机器数的方法很多，通常用的有原码，补码，反码三种。

一、原码

用原码表示正数和负数时，第一位是符号位，0 表示正数，1 表示负数，其余

部分是数值部分。例如已知二个数

$$X_1 = +1001 \qquad X_2 = -1001$$

则 x_1 和 x_2 的原码表示如下：

$$[X_1]_{原} = 01001 \qquad [X_2]_{原} = 11001$$

根据原码表示规则,当真数 N 是整数时。原码一般表示为：

$$[N]_{原} = \begin{cases} N & 0 \leqslant N < 2^n \\ 2^n - N & -2^n < N \leqslant 0 \end{cases} \tag{1.1.5}$$

(1.1.5)式中 n 是真数 N 的位数。

如果真数 N 是小数时。则原码的表达式为：

$$[N]_{原} = \begin{cases} N & 0 \leqslant N < 1 \\ 1 - N & -1 < N \leqslant 0 \end{cases} \tag{1.1.6}$$

从原码的表示式可见：当 N 为正数时$[N]_{原}$和 N 的区别在于增加一位用 0 表示的符号位,且在数的左边,对该数无影响,所以$[N]_{原}$就是 N 本身；当 N 为负数时,$[N]_{原}$和 N 的区别在于增加一位符号位 1；当 $N=0$ 时,有两种不同表示形式：正零和负零。

二、反码

用反码表示正数和负数时,符号位部分与原码表示法相同,数值部分与它的符号位有关：对于正数,反码的数值与原码相同；对于负数,反码数值是原码数值取反组成。例如：

$$X_1 = +1001 \qquad X_2 = -1001$$

$$[X_1]_{原} = 01001 \qquad [X_2]_{原} = 11001$$

$$[X_1]_{反} = 01001 \qquad [X_2]_{反} = 10110$$

根据上述规则,当真数 N 是整数时,反码一般表达式为

$$[N]_{反} = \begin{cases} N & 0 \leqslant N < 2^n \\ 2^{n+1-1} + N & -2^n < N \leqslant 0 \end{cases} \tag{1.1.7}$$

(1.1.7)式中 n 是真数 N 的位数

当真数 N 是小数时,则反码的一般表达式为：

$$[N]_{反} = \begin{cases} N & 0 \leqslant N < 1 \\ (2 - 2^{-m}) + N & -1 \leqslant N < 0 \end{cases} \tag{1.1.8}$$

(1.1.8)式中 m 是 N 小数的位数。例如

$$X_1 = 0.0101 \qquad X_2 = -0.0101$$

$$[X_1]_{反} = 0.0101 \qquad [X_2]_{反} = 1.1010$$

对于 $N=0$,反码也有两种表示形式,即正零和负零。

三、补码

用补码表示正数和负数时,符号位部分与原码表示法相同,0 表示正数,1 表

此为试读，需要完整PDF请访问：www.ertongbook.com