数字逻辑电路

杨文霞 孙青林 编著

南開大學出版社

数字逻辑电路

杨文霞　孙青林　编著

南开大学出版社

天　津

图书在版编目(CIP)数据

数字逻辑电路 / 杨文霞，孙青林编著. —天津：南开大学出版社，2014.4

ISBN 978-7-310-04432-0

Ⅰ. ①数… Ⅱ. ①杨…②孙… Ⅲ. ①数字电路—逻辑电路—高等学校—教材 Ⅳ. ①TN79

中国版本图书馆 CIP 数据核字(2014)第 041692 号

南开大学出版社出版发行

出版人:孙克强

地址:天津市南开区卫津路 94 号　邮政编码:300071

营销部电话:(022)23508339　23500755

营销部传真:(022)23508542　邮购部电话:(022)23502200

*

天津市蓟县宏图印务有限公司印刷

全国各地新华书店经销

*

2014 年 4 月第 1 版　2014 年 4 月第 1 次印刷

260×185 毫米　16 开本　23.625 印张　598 千字

定价:41.00 元

如遇图书印装质量问题,请与本社营销部联系调换,电话:(022)23507125

内容简介

本书是为高等学校电子类、计算机类、自动化类以及其他相近专业编写的教材。本书的特点是在介绍基本理论和基本分析方法的基础上，增加了数字系统设计的内容，初步介绍了EDA和硬件描述语言，强化了可编辑逻辑器件的内容。

主要内容有：逻辑代数基础、门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲波形的产生和整形、半导体存储器和可编程逻辑器件、数字系统设计、EDA及硬件描述语言和数／模、模／数转换器。

本书既注重基本理论的阐述，又重视实际应用能力的培养。既可作为高等学校电子、自动化类以及相近专业的专业基础课教材，也可为从事电子技术工作的人员提供参考。

南开大学出版社网址：http://www.nkup.com.cn

投稿电话及邮箱： 022-23504636 QQ：1760493289
QQ：2046170045(对外合作)
邮购部： 022-23507092
发行部： 022-23508339 Fax：022-23508542

南开教育云：http://www.nkcloud.org

App：南开书店 app

南开教育云由南开大学出版社、国家数字出版基地、天津市多媒体教育技术研究会共同开发，主要包括数字出版、数字书店、数字图书馆、数字课堂及数字虚拟校园等内容平台。数字书店提供图书、电子音像产品的在线销售；虚拟校园提供 360 校园实景；数字课堂提供网络多媒体课程及课件、远程双向互动教室和网络会议系统。在线购书可免费使用学习平台，视频教室等扩展功能。

目 录

第 1 章　逻辑代数基础

内容提要　逻辑代数是分析和设计数字逻辑电路的数学工具。本章首先介绍数字系统中常用的几种数制和码制，然后介绍逻辑代数中的基本公式、常用公式和基本定理以及逻辑函数的表示方法，最后重点介绍逻辑函数的化简方法。

1.1　数字逻辑电路概述

1.1.1　数字信号和数字电路

电子技术中的工作信号分为模拟信号和数字信号两大类。模拟信号是指时间上和数值上都是连续变化的信号，如电视的图像和伴音信号，在生产过程中由传感器检测的由某种物理量（如温度、压力）转化成的电信号等。传输、处理模拟信号的电路称为模拟电路。数字信号是指时间上和数值上都是不连续变化的离散信号，如生产中记录零件个数的计数信号、电子表的秒信号等。它们的变化发生在一系列离散的瞬间，数值的大小和增减总是某一最小单位的整数倍。传输、处理数字信号的电路称为数字逻辑电路，简称数字电路。

1.1.2　数字电路的特点

在数字电路中，数字信号往往表现为突变的电压或电流，因此用 0 和 1 组成的二值量表示数字信号最为简单。在数字电路中常用 0 和 1 分别代表电压的高、低，脉冲的有、无两个离散值。所以数字电路在电路结构、工作状态、研究内容和分析方法等方面都与模拟电路不同，具有如下一些特点：

① 数字信号常用二进制数来表示。由于数字电路中只有 0 和 1 两个数字，所以在数字运算时采用二进制数制比较方便。电路上可用 0 和 1 表示电子器件的开、关两种对立的状态。

② 数字电路在稳态时，电子器件（如三极管）工作于开关状态，即工作在饱和区和截止区。由于三极管的饱和和截止两种状态的外部表现为电流的有、无，电压的高、低，所以这两种状态可分别对应于 1 和 0 两个数码。而在模拟电路中，电子器件通常工作于放大状态。

③ 数字电路的基本单元电路简单，对元件的精度要求不高，允许有较大的误差。因为数字信号的 0 和 1 没有数量的含义，而只是状态的含义，所以电路工作时只要能可靠地区分 0 和 1 两个状态即可。因此数字电路便于集成化、系列化生产，具有使用方便，可靠性高，价格低廉等优点。

④ 数字电路研究的对象是输入信号和输出信号之间的逻辑关系。而模拟电路研究的对象是电路对输入信号的放大和变换功能。

⑤ 数字电路的分析工具是逻辑代数，表达电路功能主要用功能表、真值表、逻辑表达式和波形图等。而模拟电路采用的分析方法是图解法和微变等效电路。

⑥ 数字电路能对数字信号进行各种逻辑运算和算术运算，所以在各种数控装置、智能仪表以及计算机中得到广泛应用。

由于数字电路具有误差小、抗干扰性强、精度高、数据容易保存等优点，使得数字电路近年来得到长足的发展，数字系统和数字设备已经广泛应用于各个领域。

1.1.3 数字电路的分类

数字电路按其组成的结构不同可分为分立元件电路和集成电路两大类。其中，集成电路按集成度大小又分为小规模集成电路（SSI，集成度为 1～10 门/片）、中规模集成电路（MSI，集成度为 10～100 门/片）、大规模集成电路（LSI，集成度为 100～1000 门/片）和超大规模集成电路（VLSI，集成度大于 1000 门/片）。

按电路所用器件的不同，数字电路可分为双极型和单极型电路。其中，双极型电路又有 DTL、TTL、ECL、IIL、HTL 等多种；单极型电路有 JFET、NMOS、PMOS、CMOS 等四种。

根据电路逻辑功能的不同特点，数字电路又可分为组合逻辑电路和时序逻辑电路两大类。

1.2 数　制

在用数字量表示物理量的数值大小时，对于比较大的数字，一位数显然不够，需要用多位数码。多位数中每一位的构成方式和进位规则称为数制。数制是用计数符号的个数（亦称基数）来命名的。日常生活中最常用的是十进制。而在数字系统中，多采用二进制，有时也采用八进制和十六进制。

一种进位计数制包含两个基本因素：

（1）基数

基数是计数制中所用计数符号（数码）的个数，常用 R 表示。例如在十进制中，包含 0，1，2，…，9 等 10 个计数数码，进位规则是“逢 10 进 1”，所以它的基数是 10。

（2）位权

在一个多位数中，处在不同数位的同一数码代表着不同的数值。每一数位的数值是由该位数码的值乘以处在该位的一个固定常数，这一常数称为位权值，亦称位权或权值。例如在十进制中，个位的位权是 10^0，十位的位权是 10^1，百位的位权是 10^2，依此类推。由此可见，数位的位权是基数的幂。或者说，数码 1 在该位的数值即是该位的权值。

1.2.1 十进制

基数 R=10 的数制就是十进制（decimal）。它是我们最熟悉的一种计数体制。它有 0～9 十个计数符号，按照一定的规律排列起来，表示数值的大小。低位向高位的进位规则是“逢 10 进 1”。每个数位的权值是基数从 0 开始的连续次幂。任意一个十进制数$(N)_{10}$可表示为

$$(N)_{10} = k_{n-1}\times 10^{n-1} + k_{n-2}\times 10^{n-2} + \cdots + k_0\times 10^0 + k_{-1}\times 10^{-1} + \cdots + k_{-m}\times 10^{-m} = \sum_{i=-m}^{n-1} k_i \times 10^i \quad (1.2.1)$$

式中，k_i 是第 i 位的系数，它的取值为 0～9 十个数码中的任意一个。10 是基数，10^i 为第 i 位的权值，n 为整数的位数，m 为小数的位数。由此可见，任何一个十进制数都可以写成按权值展开的形式。如十进制数$(123.45)_{10}$按权值展开为

$$(123.45)_{10} = 1\times 10^2 + 2\times 10^1 + 3\times 10^0 + 4\times 10^{-1} + 5\times 10^{-2}$$

十进制数的表示方法可以推广到任意进制数。任意进制数的一般表示式为

$$(N)_R=\sum_{i=-m}^{n-1}k_i\times R^i \tag{1.2.2}$$

式中，R 为基数，R^i 为第 i 位的权值，k_i 是第 i 位的系数，它的取值为 0～R－1 中的任一个。

1.2.2　二进制

二进制（binary）就是以 2 为基数的计数体制。它的特点是只有 0、1 两个计数符号，进位规则是“逢 2 进 1”。任意一个二进制数$(N)_2$可表示为

$$(N)_2=k_{n-1}\times 2^{n-1}+k_{n-2}\times 2^{n-2}+\cdots+k_0\times 2^0+k_{-1}\times 2^{-1}+\cdots+k_{-m}\times 2^{-m}=\sum_{i=-m}^{n-1}k_i\times 2^i \tag{1.2.3}$$

式中符号的意义同十进制，只是系数k_i的取值仅为 0 和 1 中的一个。例如二进制数$(1101.101)_2$可展开为

$$(1101.101)_2=1\times 2^3+1\times 2^2+0\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}$$

1.2.3　八进制和十六进制

用二进制表示一个较大的数时，位数太多，书写和阅读均不方便。因此在计算机中还常常使用八进制（octadic）和十六进制（hexadecimal）。对于同一个数表示起来位数少，并且它们和二进制之间的转换也很方便。

八进制是基数为 8 的计数体制，用计数符号为 0，1，2，3，4，5，6，7。进位规则是“逢 8 进 1”。一个八进制数可表示为

$$(N)_8=\sum_{i=-m}^{n-1}k_i\times 8^i \tag{1.2.4}$$

系数k_i的取值为数码 0～7 中的一个。八进制数$(123.45)_8$按权值展开为

$$(123.45)_8=1\times 8^2+2\times 8^1+3\times 8^0+4\times 8^{-1}+5\times 8^{-2}$$

十六进制就是以 16 为基数的计数体制，用 0～9 和 A、B、C、D、E、F 十六个符号表示，进位规则是“逢 16 进 1”。一个十六进制数的一般表示为

$$(N)_{16}=\sum_{i=-m}^{n-1}k_i\times 16^i \tag{1.2.5}$$

十六进制数$(3A.4B)_{16}$按权值展开为

$$(3A.4B)_{16}=3\times 16^1+10\times 16^0+4\times 16^{-1}+11\times 16^{-2}$$

为了区别不同数值，需要加下标加以标注。有时下标可用不同数制的英文字头表示。如$(N)_B$、$(N)_O$、$(N)_D$和$(N)_H$分别表示二进制、八进制、十进制和十六进制数。

1.2.4　各种数制之间的转换

在实际应用中，常常需要进行各种数制之间的相互转换。数制之间的转换可归为两类：十进制数和非十进制数之间的转换；2^n 进制数之间的转换。分别介绍如下。

1. 十进制数和非十进制数之间的转换

1）非十进制数转换成十进制数

由二进制、八进制、十六进制数的表示式可知，只要将它们按位权展开，求出各位数值之和，即可得到对应的十进制数。

【例 1.2.1】 将二进制数 $(1011.01)_2$ 转换为十进制数。

解　将$(1011.01)_2$按位权展开如下：

$$\begin{aligned}(1011.01)_2 &= 1\times2^3+0\times2^2+1\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}\\ &= 8+2+1+0.25=11.25\end{aligned}$$

所以，$(1011.01)_2=(11.25)_{10}$。

【例 1.2.2】 将十六进制数$(A2.8C)_{16}$转换为十进制数。

解

$$\begin{aligned}(A2.8C)_{16} &= 10\times16^1+2\times16^0+8\times16^{-1}+12\times16^{-2}\\ &= 160+2+\frac{1}{2}+\frac{3}{64}=(162\frac{35}{64})_{10}\\ &\approx (162.547)_{10}\end{aligned}$$

2）十进制数转换成非十进制数

十进制数转换成非十进制数时，要将其整数部分和小数部分分别转换，再把结果合并成目的数制。

（1）整数部分的转换——除基（R）取余法

整数部分转换采用除基取余法。即用目的数制的基数去除十进制数，第一次所得的余数为目的数的最低位，把得到的商再除以 R，所得余数为目的数的次低位，依此类推，直至商为 0 时，所得余数为目的数的最高位。这种方法称为“除基取余法”。

【例 1.2.3】 将十进制数$(29)_{10}$转换为二进制数、八进制数和十六进制数。

解　因为是十进制整数，按照除基取余法，转换为二进制数应逐次除以 2 取余数：

			余数	
2	29		余数	
2	14	………	1	(k_0)
2	7	………	0	(k_1)
2	3	………	1	(k_2)
2	1	………	1	(k_3)
	0	………	1	(k_4)

所以，$(29)_{10}=(11101)_2$。

同一方法可把十进制数$(29)_{10}$转换为八进制数和十六进制数。

			余数	
8	29		余数	
8	3	………	5	(k_0)
	0	………	3	(k_1)

			余数	
16	29		余数	
16	1	………	13	(k_0)
	0	………	1	(k_1)

所以，$(29)_{10}=(35)_8$，$(29)_{10}=(1D)_{16}$。

（2）小数部分的转换——乘基（R）取整法

小数部分的转换采用乘基取整法。即用目的数制的基数 R 乘以十进制数，第一次相乘，

其结果的整数部分为目的数的最高位（k_{-1}），把结果的小数部分再乘以 R，所得结果的整数部分为目的数的次高位，依此类推，直至小数部分为 0 或者达到要求的精度为止。所以这种方法称为“乘基取整法”。

【例 1.2.4】将十进制数 $(0.625)_{10}$ 转换为二进制数、八进制数和十六进制数。

解　按乘基取整的方法，有：

$0.625\times2=1.25$　$1\cdots k_{-1}$　　$0.625\times8=5.00$　　$0.625\times16=10.00$

$0.25\times2=0.50$　$0\cdots k_{-2}$

$0.5\times2=1.00$　$1\cdots k_{-3}$

所以，$(0.625)_{10}=(0.101)_2=(0.5)_8=(0.A)_{16}$。

如果转换中小数部分一直不为 0，则根据精度要求，确定位数。转换成二进制数时，末位采用“0 舍 1 入”的原则。转换成其他进制数时，方法类似。

【例 1.2.5】将十进制数 $(3.42)_{10}$ 转换为二进制数（保留 4 位小数）。

解　对十进制数的整数和小数部分分别进行转换，然后合并结果即可。

整数部分

$$\begin{array}{r|l} 2 & 3 \\ \hline 2 & 1 \\ \hline & 0 \end{array}$$

余数

………1　(k_0)

………1　(k_1)

小数部分

$0.42\times2=0.84\ldots\ldots0$　(k_{-1})

$0.84\times2=1.68\ldots\ldots1$　(k_{-2})

$0.68\times2=1.36\ldots\ldots1$　(k_{-3})

$0.36\times2=0.72\ldots\ldots0$　(k_{-4})

$0.72\times2=1.44\ldots\ldots1$　(k_{-5})

所以，$(3.42)_{10}=(11.0111)_2$。

如果十进制数的位数较多，作除法非常烦琐，可以先把十进制数写成二进制数的按权展开式，然后写出二进制形式。这样可以简化转换过程。如将 $(1214)_{10}$ 转换为二进制数：

$$\begin{aligned}(1214)_{10}&=2^{10}+2^7+2^5+2^4+2^3+2^2+2^1\\&=(10010111110)_2\end{aligned}$$

2. 2^n 进制数之间的转换

（1）二进制数与八进制数之间的转换

由于八进制数的基数 $8=2^3$，所以 3 位二进制数构成 1 位八进制数。若要将二进制数转换成八进制数，只要将二进制数的整数部分自右向左每 3 位一组，最后一组不足 3 位时以 0 补足；小数部分自左向右每 3 位一组，最后一组不足 3 位时以 0 补齐。再把每组对应的八进制数写出即可。

【例 1.2.6】将二进制数 $(1100110.0101)_2$ 转换为八进制数。

解　按以上方法有

$$\begin{array}{ccccc} \underline{001} & \underline{100} & \underline{110} & . & \underline{010}\ \underline{100} \\ 1 & 4 & 6 & . & 2\quad 4 \end{array}$$

所以，$(1100110.0101)_2=(146.24)_8$。

反之，若要将八进制数转换成二进制数，只要将每 1 位八进制数写成 3 位二进制数，按顺序排列起来即可。

【**例 1.2.7**】将八进制数 $(45.36)_8$ 转换为二进制数。

解

$$\begin{array}{cccc} 4 & 5 & . & 3 \quad 6 \\ \downarrow & \downarrow & & \downarrow \quad \downarrow \\ 100 & 101 & . & 011 \quad 110 \end{array}$$

所以，$(45.36)_8 = (100101.01111)_2$。

（2）二进制数与十六进制数之间的转换

由于十六进制数的基数$16 = 2^4$，所以每 4 位二进制数对应 1 位十六进制数。按照二进制数转八进制数相同的步骤，只要将二进制数每 4 位分组，即可实现二进制数与十六进制数之间的相互转换。

【**例 1.2.8**】将二进制数 $(1011110.110101)_2$ 转换为十六进制数。

解

$$\begin{array}{cccc} \underline{0101} & \underline{1110}. & \underline{1101} & \underline{0100} \\ 5 & \mathrm{E}. & \mathrm{D} & 4 \end{array}$$

所以，$(1011110.110101)_2 = (5E.D4)_{16}$。

（3）八进制数和十六进制数之间转换

八进制数和十六进制数之间相互转换通常采用间接转换法：先将八进制数或十六进制数转换为二进制数，再将二进制数转换为相应的目的进制数。

【**例 1.2.9**】将八进制数 $(36.47)_8$ 转换为十六进制数。

解 $(36.47)_8 = (011110.100111)_2 = (1E.9C)_{16}$

从上面的例子可以看出，直接由十进制数转换为八进制和十六进制比较烦琐，计算容易出错。一般先化成二进制数，然后再化为八进制或十六进制数。如果十进制数位数较多，作除法也同样很麻烦，这时可先把十进制数写成二进制数的按权展开式，然后写出二进制形式。这样可以简化转换过程。如将 $(1214)_{10}$ 转换为八进制和十六进制数：

$$\begin{aligned}(1214)_{10} &= 2^{10} + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 \\ &= (10010111110)_2 \\ &= (2276)_8 = (4BE)_{16}\end{aligned}$$

1.3 码制与编码

在各种数制中，数码是用来表示数值大小的。实际上数码不仅可以表示数量的大小，也可以用来表示不同的事物，如学生的学号、教室的编号等。在这些情况下，数码已没有了数量大小的含义，只是用来表示不同事物的代号。这些数码称为代码。另外我们知道，计算机可以存储和处理很多不同的信息，而实际上机器只能识别二进制数，因而这些信息都是用各种不同的数码来表示的。用数码表示信息的方法很多，建立信息和数码对应关系的过程称为编码。为便于记忆和查找方便，编码时要遵循一定的规则，这些不同的编码规则称为码制。同样，在计算机和其他数字系统体中，0,1,2,…,9 这 10 个十进制数在用二进制数码表示时，就有不同的编码规则。通常将这些代码称为二—十进制代码（Binary Coded Decimal，简称 BCD）。下面介绍几种常用的编码。