

21 世纪高等院校信息与通信工程规划教材
21st Century University Planned Textbooks of Information and Communication Engineering

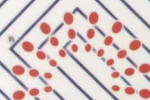
数字电子技术

梁龙学 主编
李峰 邸敬 吴小红 陈占林 编著

Digital Electronic Technique



人民邮电出版社
POSTS & TELECOM PRESS



高校系列

21 世纪高等院校信息与通信工程规划教材
21st Century University Planned Textbooks of Information and Communication Engineering

数字电子技术

梁龙学 主编
李峰 邸敬 吴小红 陈占林 编著

Digital Electronic Technique

人民邮电出版社
北京



图书在版编目(CIP)数据

数字电子技术 / 梁龙学主编 ; 李峰等编著. — 北京 : 人民邮电出版社, 2010.9
21世纪高等院校信息与通信工程规划教材
ISBN 978-7-115-23302-8

I. ①数… II. ①梁… ②李… III. ①数字电路—电子技术—高等学校—教材 IV. ①TN79

中国版本图书馆CIP数据核字(2010)第150191号

内 容 提 要

本书是根据兰州交通大学电信学院电子技术教研室多年来的教学实践经验,并结合新的课程体系和教学内容改革的需要而编写的数字电子技术教材。本书力求简明、实用,每章内容不仅层次清楚,而且语言生动形象,好学易记。全书共8章,内容包括逻辑代数基础、门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲信号的产生和变换、半导体存储器及数—模转换和模—数转换等。为便于对各章内容总体的理解与掌握,章前有提要,章末有小结,并配有数量和难易程度适中的习题。

本书通俗易懂、深入浅出,着重学生分析问题、解决问题能力的培养,便于自学。本书可作为高校(尤其是工科类院校)电子信息、通信工程、计算机应用、自动控制及自动化,以及其他相近专业的本、专科的教学用书,也可作为其他专业师生及电子工程技术人员的参考书。

21 世纪高等院校信息与通信工程规划教材

数字电子技术

-
- ◆ 主 编 梁龙学
编 著 李 峰 邸 敬 吴小红 陈占林
责任编辑 滑 玉
执行编辑 董 楠
 - ◆ 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>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 17.25 2010 年 9 月第 1 版
字数: 420 千字 2010 年 9 月河北第 1 次印刷

ISBN 978-7-115-23302-8

定价: 30.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

本书是为了适应普通高等工科院校电子信息类专业面向 21 世纪教学改革的需要而编写的。

数字电子技术基础为电类专业实践性和实用性均很强的技术基础课程之一，同模拟电子技术基础一道构成了学生知识体系的“硬件”基础，是所有电类专业学生必须重点掌握的专业基础课程。随着“轻专业，重基础”教学改革的进行，本课程在人才培养中的作用就显得更加重要。

本书既适用“先模拟后数字”的教学体系，也适用“模拟数字同时进行”的教学体系。编写本书的基本思想如下。

- 从“电子技术基础课程”的定位出发，既要满足教学内容的需要，又要具有一定的实用性。充分体现了工科院校专业基础教学“保基础、重应用、少而精”的特点。全书内容安排可以满足 80 学时、64 学时和 48 学时等不同学时的教学要求。

- “活”是数字电子技术的灵魂。理解记忆、活学活用，是学好以及今后用好数字电子技术的关键。本书在编写风格上充分体现了这一点。

- 本书在编写时还很注重概念以及由原理电路到实际应用电路（集成电路）的过渡，使学生不但“知其然”，而且“知其所以然”。这样不仅有利于学生对知识的理解和掌握，而且对学生工程应用能力和创新能力的培养亦大有裨益。

- 强调理论与工程实践相结合。本书选编了较多的例题和应用实例。为了加强对课堂知识的理解与掌握，各章都配有一定数量的习题。

- 在内容的叙述和顺序安排上，力求删繁就简，深入浅出，通俗易懂。考虑到不同专业、不同学时教学内容和课堂安排上的差异，有些章节为选学内容，教学时可根据实际需要加以取舍。

本书共 8 章，由梁龙学主编。其中第 1 章、第 7 章由邸敬编写，第 2 章由李峰编写，第 3 章、第 8 章由梁龙学编写，第 4 章、第 6 章由陈占林编写，第 5 章及附录由吴小红编写。在编写过程中得到了兰州理工大学技术工程学院的大力支持，王瑞祥和谢黎明不辞辛劳地审阅了全书，并对原稿提出了许多建设性的意见；任宗义在本书大纲的制订与编写过程中，提出了很多宝贵的修改意见。在此一并表示衷心的感谢。

由于编者水平有限，书中难免存在缺点和错误，敬请读者批评指正。

编 者

2010 年 7 月

目 录

第 1 章 逻辑代数基础	1	2.2.2 半导体三极管的开关特性.....	38
1.1 概述	1	2.2.3 MOS 管的开关特性	40
1.1.1 数字量和模拟量	1	2.3 分立元件门电路.....	40
1.1.2 数制和码制	1	2.3.1 二极管与门.....	40
1.2 逻辑代数中的 3 种基本运算	6	2.3.2 二极管或门.....	41
1.3 逻辑代数的基本公式和常用公式	9	2.3.3 三极管非门.....	42
1.3.1 基本公式	9	2.3.4 MOS 管非门	42
1.3.2 若干常用公式.....	10	2.4 TTL 集成门电路	43
1.4 逻辑代数的 3 个重要规则.....	11	2.4.1 TTL 反相器的电路结构和工作原理.....	43
1.5 逻辑函数及其表示方法.....	13	2.4.2 TTL 反相器的静态特性和动态特性.....	43
1.5.1 逻辑函数.....	13	2.4.3 其他类型的 TTL 门电路.....	47
1.5.2 逻辑函数的表示方法.....	13	2.4.4 TTL 电路的改进系列	51
1.6 逻辑函数的公式化简法.....	16	2.5 集成 CMOS 门电路	52
1.6.1 逻辑函数的最简表达式	16	2.5.1 CMOS 反相器的工作原理.....	52
1.6.2 常用的公式化简方法.....	18	2.5.2 CMOS 反相器的静态特性和动态特性.....	52
1.6.3 逻辑函数的两种标准形式.....	21	2.5.3 其他类型的 CMOS 门电路.....	56
1.7 逻辑函数的卡诺图化简法.....	23	2.5.4 CMOS 电路产品系列、主要特点及使用注意事项.....	58
1.7.1 逻辑函数的卡诺图表示法.....	23	第 3 章 组合逻辑电路	65
1.7.2 用卡诺图化简逻辑函数	25	3.1 概述.....	65
1.8 具有约束项的逻辑函数及其化简	28	3.2 组合逻辑电路的分析.....	65
1.8.1 约束项、任意项和逻辑函数式中的无关项.....	28	3.2.1 分析方法及步骤.....	65
1.8.2 无关项在化简逻辑函数中的应用.....	29	3.2.2 分析方法举例.....	66
第 2 章 门电路	36	3.3 组合逻辑电路的设计.....	68
2.1 概述.....	36	3.3.1 设计方法.....	68
2.2 半导体二极管、三极管和 MOS 管的开关特性	37	3.3.2 设计举例.....	69
2.2.1 半导体二极管的开关特性.....	37	3.3.3 逻辑门多余输入端的	

处理	70	特性	128
3.4 常用 MSI 组合逻辑器件	71	4.7.2 同步 RS 触发器的动态特性	129
3.4.1 编码器	71	4.7.3 主从触发器的动态特性	130
3.4.2 译码器	77	4.7.4 维持阻塞触发器的动态特性	131
3.4.3 数据选择器	88	第 5 章 时序逻辑电路	137
3.4.4 加法器	94	5.1 概述	137
3.4.5 数值比较器	99	5.2 时序电路的分析	138
3.5 组合逻辑电路中的竞争—冒险现象	102	5.2.1 同步时序电路的分析	138
3.5.1 产生竞争—冒险的原因	102	* 5.2.2 异步时序电路的分析	143
3.5.2 竞争—冒险的判别方法	103	5.3 时序电路设计	144
3.5.3 冒险现象的消除	104	5.3.1 时序电路设计方法及步骤	145
第 4 章 触发器	110	5.3.2 计数器及其分类	146
4.1 概述	110	5.4 计数器设计	146
4.2 基本触发器	111	5.4.1 基于触发器的计数器设计	146
4.2.1 用或非门组成的基本 RS 触发器	111	5.4.2 基于集成计数器的计数器设计——集成计数器及其应用	151
4.2.2 用与非门组成的基本 RS 触发器	113	5.5 寄存器	173
4.2.3 基本 RS 触发器的主要特点	114	5.5.1 数码寄存器	173
4.3 同步触发器	115	5.5.2 锁存器	174
4.3.1 同步 RS 触发器	115	5.5.3 移位寄存器	175
4.3.2 同步 D 触发器	117	5.5.4 移位寄存器的应用	177
4.4 主从触发器	118	5.6 顺序脉冲发生器	179
4.4.1 主从 RS 触发器	118	* 5.7 时序逻辑电路中的竞争—冒险现象	181
4.4.2 主从 JK 触发器	120	第 6 章 脉冲信号的产生和变换	187
4.4.3 T 触发器和 T' 触发器	122	6.1 多谐振荡器	187
4.5 边沿触发器	123	6.1.1 环形振荡器	187
4.6 触发器逻辑功能表示方法及其相互转换	126	6.1.2 对称式多谐振荡器	190
4.6.1 触发器逻辑功能表示方法	126	6.1.3 石英晶体振荡器	191
4.6.2 触发器逻辑功能表示方法之间的转换	128	6.2 单稳态触发器	192
4.7 触发器的动态特性	128	6.2.1 微分型单稳态触发器	193
4.7.1 基本 RS 触发器的动态特性	128	6.2.2 集成单稳态触发器	195

6.2.3 单稳态触发器的应用	198
6.3 施密特触发器	200
6.3.1 门电路组成的施密特触发器	201
6.3.2 集成施密特触发器	203
6.3.3 施密特触发器的应用 ...	204
6.4 555 定时器及其应用	206
6.4.1 555 定时器的电路结构与功能	206
6.4.2 555 定时器的应用	207
第 7 章 半导体存储器	214
7.1 概述	214
7.2 只读存储器	215
7.2.1 ROM 的结构及工作原理	215
7.2.2 ROM 的不同类型	218
7.3 随机存储器	219
7.3.1 静态随机存储器	220
* 7.3.2 动态随机存储器	223
7.4 存储器容量的扩展	223
7.4.1 位扩展方式	223
7.4.2 字扩展方式	224
7.5 用存储器实现组合逻辑函数	225

第 8 章 数—模转换和模—数转换	233
8.1 概述	233
8.2 D/A 转换器	234
8.2.1 D/A 转换器的基本原理	234
8.2.2 常用 D/A 转换电路	235
8.2.3 DAC 的主要技术指标	238
8.2.4 常用集成 DAC 及应用举例	240
8.3 A/D 转换器	243
8.3.1 A/D 转换器的基本原理	243
8.3.2 常用 A/D 转换电路	247
8.3.3 ADC 的主要技术指标	256
8.3.4 常用集成 ADC 及应用举例	257
* 8.4 数模接口电路的应用举例	259
8.4.1 程控增益放大器	259
8.4.2 数据采集与控制系统 ...	260
附录 数字电子技术综合应用举例 ...	266
参考文献	268

逻辑代数是分析和设计数字电路的基本数学工具，它的基本和常用运算也是数字电路要实现的重要操作。本章首先介绍了数制与码制，其次介绍了逻辑代数的基本公式、常用公式和重要规则，然后讲述逻辑函数及其表示方法，最后介绍了如何应用公式法和卡诺图法化简逻辑函数。

1.1 概述

1.1.1 数字量和模拟量

在观察自然界中形形色色的物理量时不难发现，尽管它们的性质各异，但就其变化规律和特点而言，不外乎两大类。

其中一类物理量的变化在时间上和数量上都是离散的。也就是说，它们的变化在时间上是不连续的，总是发生在一系列离散的瞬间。这一类物理量叫做数字量，用数字量表示的信号叫做数字信号，把工作在这种数字信号下的电子电路叫做数字电路。

例如，一件事的是与非，一个开关的闭合与断开，一盏电灯的亮与灭等，均只有两种状态，可用 0 和 1 来分别表示。在这样的二值逻辑中，变量的取值只有 0 和 1 两种，没有第三种可能。显然这一类的物理量无论在时间上还是在数量上都是不连续的，故为数字量。

另一类物理量的变化在时间上及数值上均是连续的。这一类物理量叫做模拟量，用模拟量表示的信号叫做模拟信号，把工作在这种模拟信号下的电子电路称为模拟电路。

例如，热敏传感器在工作时输出的电压信号就属于模拟信号，因为在任何情况下被测温度都不可能发生跳变，所以测得的电压信号无论在时间上还是在数量上都是连续的。而且，这个电压信号在连续变化过程中的任何一个取值都有具体的物理意义，即表示某一时刻被测物体的温度。

1.1.2 数制和码制

1. 数制

用数字量表示物理量的大小时，仅用一位数码往往不够用，因此经常需要用进位计数的方法由多位数码表示。人们把多位数码中每一位的构成方法以及从低位到高位进位的规则称为数制。

在数字电路中经常使用的计数进制除了十进制以外，还有二进制和十六进制。

(1) 十进制

十进制是人们在日常生活和工作中最常使用的进位计数制，可谓是人类共同的“数学语言”。在十进制数中，每一位有 0~9 十个数码，所以计数的基数是 10。超过 9 的数必须用多位数表示，其中低位和相邻高位之间的关系是“逢十进一”，故称为十进制。例如

$$256.75 = 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

所以任意一个十进制数 D 均可展开为

$$D = \sum k_i \times 10^i \quad (1-1)$$

其中 k_i 是第 i 位的系数，它可以是 0~9 这 10 个数码中的任何一个。若整数部分的位数是 n ，小数部分的位数为 m ，则 i 包含从 $n-1$ 到 0 的所有正整数和从 -1 到 $-m$ 的所有负整数。

若以 N 取代式 (1-1) 中的 10，即可得到任意进制 (N 进制) 数展开式的普遍形式

$$D = \sum k_i N^i \quad (1-2)$$

式中 i 的取值与式 (1-1) 的规定相同。 N 称为计数的基数， k_i 为第 i 位的系数， N^i 称为第 i 位的权。

(2) 二进制

目前在数字电路中广泛应用的是二进制计数方法，可谓是机器（或数字电路）的“数学语言”。在二进制数中，每一位仅有 0 和 1 两个可能的数码，所以计数基数为 2。低位和相邻高位间的进位关系是“逢二进一”，故称为二进制。

根据式 (1-2)，任何一个二进制数均可展开为

$$D = \sum k_i 2^i \quad (1-3)$$

并可计算出它所表示的十进制数的大小。例如

$$\begin{aligned} (101.11)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= (5.75)_{10} \end{aligned}$$

上式中分别使用下脚注的 2 和 10 表示括号里的数是二进制和十进制数。有时也用 B (Binary) 和 D (Decimal) 代替 2 和 10 这两个脚注。

(3) 十六进制

二进制的最大优点就是容易用电路来识别及实现，但不便于书写及表示，于是出现了二进制的缩写形式（八进制、十六进制等），其中最常用的为十六进制。

十六进制数的每一位有十六个不同的数码，分别用 0~9、A (10)、B (11)、C (12)、D (13)、E (14)、F (15) 表示。因此，任意一个十六制数均可展开为

$$D = \sum k_i 16^i \quad (1-4)$$

并由此式计算出它所表示的十进制数值。例如

$$\begin{aligned} (2A.7F)_{16} &= 2 \times 16^1 + 10 \times 16^0 + 7 \times 16^{-1} + 15 \times 16^{-2} \\ &= (42.4960937)_{10} \end{aligned}$$

式中的下脚注 16 表示括号里的数是十六进制，有时也用 H (Hexadecimal) 代替这个脚注。

由于目前在微型计算机中普遍采用 8 位、16 位和 32 位二进制并行运算，而 8 位、16 位和 32 位的二进制数可以用 2 位、4 位和 8 位的十六进制数表示，因而用十六进制符号书写程序十分简便。

2. 数制转换

前边已经讲过，二进制数、十进制数分别为机器及人类的“数学语言”，二者之间就不可避免地需要进行相互转换。

(1) 二—十转换

把二进制数转换为等值的十进制数称为二—十转换。转换时只要将二进制数按权展开，然后把所有各项的数值按十进制相加，就可以得到等值的十进制数了。例如

$$\begin{aligned}(1011.01)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= (11.25)_{10}\end{aligned}$$

(2) 十—二转换

十—二转换就是把十进制数转换成等值的二进制数。可分以下 3 种情况讨论。

① 纯整数的转换——除 2 取余倒排法。假定十进制整数为 $(S)_{10}$ ，等值的二进制数为 $(k_n k_{n-1} \cdots k_0)_2$ ，则依式 (1-3) 可知

$$\begin{aligned}(S)_{10} &= k_n 2^n + k_{n-1} 2^{n-1} + \cdots + k_1 2^1 + k_0 2^0 \\ &= 2(k_n 2^{n-1} + k_{n-1} 2^{n-2} + \cdots + k_1) + k_0\end{aligned}\quad (1-5)$$

上式表明，若将 $(S)_{10}$ 除以 2，则得到的商为 $k_n 2^{n-1} + k_{n-1} 2^{n-2} + \cdots + k_1$ ，而余数即 k_0 。

同理，可将式 (1-5) 除以 2 得到的商写成

$$k_n 2^{n-1} + k_{n-1} 2^{n-2} + \cdots + k_1 = 2(k_n 2^{n-2} + k_{n-1} 2^{n-3} + \cdots + k_2) + k_1 \quad (1-6)$$

由式 (1-6) 不难看出，若将 $(S)_{10}$ 除以 2 所得的商再次除以 2，则所得余数即 k_1 。

依次类推，反复将每次得到的商再除以 2，就可求得二进制数的每一位了。

注意：在写取转换结果（即取余）时的顺序为先下后上。

例如，将 $(173)_{10}$ 化为二进制数可如下进行

2	173	 余数 = 1 = k_0
2	86	 余数 = 0 = k_1
2	43	 余数 = 1 = k_2
2	21	 余数 = 1 = k_3
2	10	 余数 = 0 = k_4
2	5	 余数 = 1 = k_5
2	2	 余数 = 0 = k_6
2	1	 余数 = 1 = k_7
	0	

故 $(173)_{10} = (10101101)_2$ 。

② 纯小数的转换——乘 2 取整顺排法。若 $(S)_{10}$ 是一个十进制的小数，对应的二进制小数为 $(0.k_{-1} k_{-2} \cdots k_{-m})_2$ ，则据式 (1-3) 可知

$$(S)_{10} = k_{-1} 2^{-1} + k_{-2} 2^{-2} + \cdots + k_{-m} 2^{-m}$$

将上式两边同乘以 2 得到

$$2(S)_{10} = k_{-1} + (k_{-2} 2^{-1} + k_{-3} 2^{-2} + \cdots + k_{-m} 2^{-m+1}) \quad (1-7)$$

式 (1-7) 说明，将小数 $(S)_{10}$ 乘以 2 所得乘积的整数部分即 k_{-1} 。

同理, 将乘积的小数部分再乘以 2 又可得到

$$2(k_{-2}2^{-1} + k_{-3}2^{-2} + \cdots + k_{-m}2^{-m+1}) = k_{-2} + (k_{-3}2^{-1} + \cdots + k_{-m}2^{-m+2}) \quad (1-8)$$

亦即乘积的整数部分就是 k_{-2} 。

依次类推, 将每次乘 2 后所得乘积的小数部分再乘以 2, 便可求出二进制小数的每一位。

例如, 将 $(0.8125)_{10}$ 化为二进制小数可如下进行

$$\begin{array}{rcl}
 & 0.8125 & \\
 \times & 2 & \\
 \hline
 & 1.6250 & \cdots \cdots \text{整数部分}=1=k_{-1} \\
 & 0.6250 & \\
 \times & 2 & \\
 \hline
 & 1.2500 & \cdots \cdots \text{整数部分}=1=k_{-2} \\
 & 0.2500 & \\
 \times & 2 & \\
 \hline
 & 0.5000 & \cdots \cdots \text{整数部分}=0=k_{-3} \\
 & 0.5000 & \\
 \times & 2 & \\
 \hline
 & 1.0000 & \cdots \cdots \text{整数部分}=1=k_{-4}
 \end{array}$$

故 $(0.8125)_{10} = (0.1101)_2$ 。

需要注意的是, 取整的顺序为自上而下, 与纯整数的转换写取顺序正好相反。

③ 既有整数又有小数的转换——分别进行转换。如

$$(173.8125)_{10} = (10101101.1101)_2$$

首先将整数部分 (173) 和小数部分 (0.8125) 分别转换为对应的二进制数, 再把两个转换结果合写在一起即可。

(3) 二一十六转换

把二进制数转换成等值的十六进制数称为二一十六转换。

由于 4 位二进制数恰好有 16 个状态, 而把这 4 位二进制数看作一个整体时, 它的进位输出又正好是逢十六进一, 所以只要从低位到高位将每 4 位二进制数分为一组并代之以等值的十六进制数, 即可得到对应的十六进制数。

例如, 将 $(01001111.10110011)_2$ 化为十六进制数时可得

$$\begin{array}{cccc}
 (0100, 1111.1011, 0011)_2 & & & \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 = (4 & F & B & 3)_{16}
 \end{array}$$

简言之, 就是以小数点为基准, 整数部分向左、小数部分向右每 4 位为一组, 不足部分补零, 再转化为对应的十六进制数即可。

(4) 十六一二转换

十六一二转换是指把十六进制数转换成等值的二进制数。转换时只需将十六进制数的每一位用等值的 4 位二进制数代替就行了。

例如, 将 $(8FA.C6)_{16}$ 化为二进制数时得到

$$\begin{array}{ccccc}
 (8 & F & A & C & 6)_{16} \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 = (1000 & 1111 & 1010 & 1100 & 0110)_2
 \end{array}$$

(5) 十六进制数与十进制数的转换

在将十六进制数转换为十进制数时，可根据式 (1-4) 将各位按权展开后相加求得。在将十进制数转换为十六进制数时，可以先转换成二进制数，然后再将得到的二进制数转换为等值的十六进制数即可。

3. 码制

不同的数码不仅可以表示数量的大小，而且还能用来表示不同的事物。在后一种情况下，这些数码不再表示数值的大小，只是表示不同事物的代码。

例如在举行长跑比赛时，为便于识别运动员，通常给每个运动员编一个号码。显然，这些号码仅仅代表不同的运动员，已失去了数量大小的含意。

为便于记忆和处理，在编制代码时总要遵循一定的规则，这些规则就叫码制。如用 4 位二进制数来表示 1 位十进制数的编码方法称为二—十进制代码，简称 BCD (Binary Coded Decimal) 代码。BCD 代码是一种十分重要的码制，在数字电路中应用广泛。

由于有多种实现方法，故 BCD 码种类很多。表 1-1 中列出了几种常见的 BCD 代码，它们的编码规则各不相同。

表 1-1 几种常见的 BCD 代码					
十进制数 \ 编码种类	8421 码	余 3 码	2421 码	5421 码	余 3 循环码
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0	0 0 1 0
1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 1	0 1 1 0
2	0 0 1 0	0 1 0 1	0 0 1 0	0 0 1 0	0 1 1 1
3	0 0 1 1	0 1 1 0	0 0 1 1	0 0 1 1	0 1 0 1
4	0 1 0 0	0 1 1 1	0 1 0 0	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 0 0	1 0 1 1	1 0 0 0	1 1 0 0
6	0 1 1 0	1 0 0 1	1 1 0 0	1 0 0 1	1 1 0 1
7	0 1 1 1	1 0 1 0	1 1 0 1	1 0 1 0	1 1 1 1
8	1 0 0 0	1 0 1 1	1 1 1 0	1 0 1 1	1 1 1 0
9	1 0 0 1	1 1 0 0	1 1 1 1	1 1 0 0	1 0 1 0
权	8 4 2 1		2 4 2 1	5 4 2 1	

8421 码是 BCD 代码中最常用的一种。在这种编码方式中每一位二值代码的 1 都代表一个固定数值，把每一位的 1 代表的十进制数加起来，得到的结果就是它所代表的十进制数码。由于代码中从左到右每一位的 1 分别表示 8、4、2、1，所以把这种代码叫做 8421 码。每一位的 1 代表的十进制数称为这一位的权。8421 码中每一位的权是固定不变的，它属于恒权代码。

余 3 码的编码规则与 8421 码不同，如果把每一个余 3 码看作 4 位二进制数，则它的数值要比它所表示的十进制数码多 3，故而将这种代码叫做余 3 码。

余 3 码不是恒权代码。如果试图把每个代码视为二进制数，并使它等效的十进制数与所表示的代码相等，那么代码中每一位的 1 所代表的十进制数在各个代码中是不能固定的。

2421 码是一种恒权代码。从左到右每一位的 1 分别表示 2、4、2、1，即它们的权分别为 2、4、2、1。

5421 码是另一种恒权代码。从左到右每一位的权分别为 5、4、2、1。

需要指出的是，2421 码和 5421 码的编码方案都不是唯一的，表 1-1 只列出了一种编码方案。如 5421BCD 码中的数码 5，既可以用 1000 表示，也可以用 0101 表示。

余 3 循环码（又称格雷码）是一种变权码，每一位的 1 在不同代码中并不代表固定的数值。它的主要特点是相邻的两个代码之间仅有一位的状态不同。

1.2 逻辑代数中的 3 种基本运算

1849 年，英国数学家乔治·布尔（George Boole）首先提出了描述客观事物逻辑关系的数学方法——布尔代数。后来，由于布尔代数被广泛地应用于解决开关电路和数字逻辑电路的分析与设计上，所以也把布尔代数叫做开关代数或逻辑代数。本章所讲的逻辑代数就是布尔代数在二值逻辑电路中的应用。

逻辑代数中常用大写字母表示变量，这种变量称为逻辑变量。在二值逻辑中，每个逻辑变量的取值只有 0 和 1 两种可能。这里的 0 和 1 已不再表示数量的大小，只代表两种不同的逻辑状态。如开关的闭合与断开、灯的亮与灭、电平的高与低等。

逻辑代数的基本运算有与、或、非 3 种。为便于理解它们的含意，可用以下电路模型来比拟。

图 1-1 中给出了 3 个指示灯的控制电路。在图 1-1 (a) 电路中，只有当两个开关同时闭合时，指示灯才会亮；在图 1-1 (b) 电路中，只要有任何一个开关闭合，指示灯就亮；而在图 1-1 (c) 电路中，开关断开时灯亮，开关闭合时灯反而不亮。

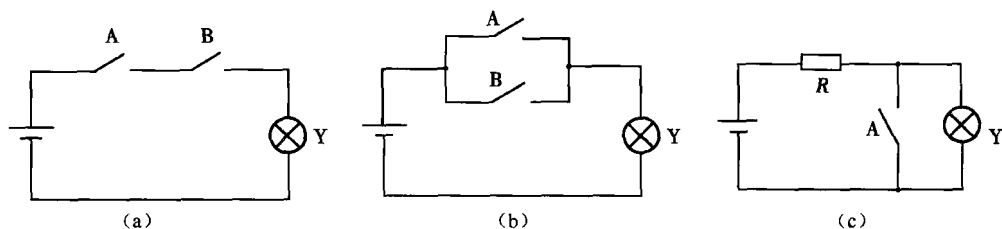


图 1-1 用于说明与、或、非定义的电路

如果把开关闭合作为条件（或导致事物结果的原因），把灯亮作为结果，那么图 1-1 中的 3 个电路代表了 3 种不同的因果关系。

图 1-1 (a) 的例子表明，只有决定事物结果的条件全部具备时，结果才发生。这种因果关系叫做逻辑与，或者叫逻辑乘。可简记为：条件全具备，结果才发生。

图 1-1 (b) 的例子表明，在决定事物结果的诸条件中只要有任何一个条件满足，结果就会发生。这种因果关系叫做逻辑或，也叫做逻辑加。可记为：条件具备，结果就发生。

图 1-1 (c) 的例子表明，条件具备了，结果反而不会发生；条件不具备，结果却发生。这种因果关系叫做逻辑非，也叫做逻辑求反。可记为：条件具备，结果不发生。

若以 A、B 表示开关的状态，并以 1 表示开关闭合，以 0 表示开关断开；以 Y 表示指示灯的状态，并以 1 表示灯亮，以 0 表示不亮，则可以列出以 0、1 表示的与、或、非逻辑关系的图表，如表 1-2、表 1-3 和表 1-4 所示。这种图表叫做逻辑真值表，简称为真值表。

表 1-2 与逻辑运算的真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

表 1-3 或逻辑运算的真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

表 1-4 非逻辑运算的真值表

A	Y
0	1
1	0

在逻辑代数中，把与、或、非看作是逻辑变量 A、B 间的 3 种最基本的逻辑运算，并以“·”表示与运算，以“+”表示或运算，以变量上边加“—”表示非运算。因此，A 和 B 进行与逻辑运算时可写成

$$Y = A \cdot B \tag{1-9}$$

A 和 B 进行或逻辑运算时可写成

$$Y = A + B \tag{1-10}$$

对 A 进行非逻辑运算时可写成

$$Y = \overline{A} \tag{1-11}$$

同时，把实现与逻辑运算的单元电路叫做与门，把实现或逻辑运算的单元电路叫做或门，把实现非逻辑运算的单元电路叫做非门（也叫做反相器）。

与、或、非逻辑运算还可以用图 1-2 所示的图形符号来表示。这些图形符号也用于表示相应的门电路。图中上边一行是目前国家标准规定的符号，下边一行是常见于国外一些书刊和资料上的符号。

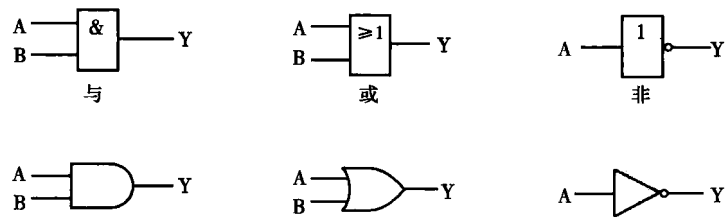


图 1-2 与、或、非的图形符号

实际的逻辑问题往往比与、或、非复杂得多，不过它们都可以用与、或、非的组合来实现。最常见的复合逻辑运算有与非、或非、与或非、异或、同或等。表 1-5～表 1-9 给出了这些复合逻辑运算的真值表。图 1-3 是它们的图形逻辑符号和运算符号。

由表 1-5 可见，将 A、B 先进行与运算，然后将结果求反，最后得到的即 A、B 的与非运算结果。因此，可以把与非运算看作是与运算和非运算的组合。图 1-3 中图形符号上的小圆圈表示非运算。

在与或非逻辑中，A、B 之间以及 C、D 之间都是与的关系，只要 A、B 或 C、D 任何一组同时为 1，输出 Y 就是 0。只有当每一组输入都不全是 1 时，输出 Y 才是 1。

表 1-5 与非逻辑的真值表

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

表 1-6 或非逻辑的真值表

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

表 1-7 与或非逻辑的真值表

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

表 1-8 异或逻辑的真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

表 1-9 同或逻辑的真值表

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

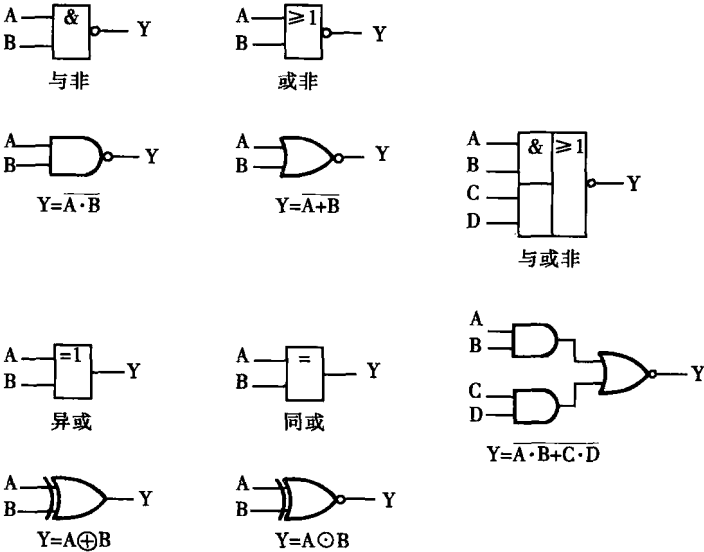


图 1-3 复合逻辑的图形符号和运算符号

异或是这样一种逻辑关系：当 A、B 不同时，输出 Y 为 1；而 A、B 相同时，输出 Y 为 0。异或也可以用与、或、非的组合表示。

$$A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B \quad (1-12)$$

同或和异或正相反，当 A、B 相同时，Y 等于 1；A、B 不同时，Y 等于 0。同或也可以写成与、或、非的组合形式。

$$A \odot B = A \cdot B + \bar{A} \cdot \bar{B} \quad (1-13)$$

而且，由表 1-8 和表 1-9 可见，异或和同或互为反运算，即

$$A \oplus B = \overline{A \odot B}; A \odot B = \overline{A \oplus B} \quad (1-14)$$

为简化书写，允许将 $A \cdot B$ 简写成 AB，略去逻辑相乘的运算符“ $\cdot$ ”。

1.3 逻辑代数的基本公式和常用公式

1.3.1 基本公式

表 1-10 给出了逻辑代数的基本公式。这些公式也叫布尔恒等式。

式 (1)、式 (2)、式 (11) 和式 (12) 给出了变量与常量间的运算规则。

式 (3) 和式 (13) 是同一变量的运算规律，也叫重叠律。

式 (4) 和式 (14) 表示变量与它的反变量之间的运算规律，也称互补律。

式 (5) 和式 (15) 为交换律，式 (6) 和式 (16) 为结合律，式 (7) 和式 (17) 为分配律。

式 (8) 和式 (18) 是著名的德·摩根 (De · Morgan) 定理，亦称反演律。在逻辑函数的化简和变换中经常要用到这一对公式。

表 1-10 逻辑代数的基本公式

序号	公 式	序号	公 式
		10	$\bar{\bar{1}}=0; \bar{\bar{0}}=1$
1	$0 \cdot A=0$	11	$1+A=1$
2	$1 \cdot A=A$	12	$0+A=A$
3	$A \cdot A=A$	13	$A+A=A$
4	$A \cdot \bar{A}=0$	14	$A+\bar{A}=1$
5	$A \cdot B=B \cdot A$	15	$A+B=B+A$
6	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	16	$A+(B+C) = (A+B) +C$
7	$A \cdot (B+C) = A \cdot B+A \cdot C$	17	$A+B \cdot C= (A+B) \cdot (A+C)$
8	$\overline{A \cdot B}=\bar{A}+\bar{B}$	18	$\overline{A+B}=\bar{A} \cdot \bar{B}$
9	$\bar{\bar{A}}=A$		

式 (9) 表明，一个变量经过两次求反运算之后还原为其本身，所以该式又称还原律。

式 (10) 是对 0 和 1 求反运算的规则，它说明 0 和 1 互为求反的结果。

这些公式的正确性可以用列真值表的方法加以验证。如果等式成立，那么将任何一组变量的取值代入公式两边所得的结果应该相等。因此，等式两边所对应的真值表也必然相同。

【例 1-1】 用真值表证明表 1-10 中式 (17) 的正确性。

解：已知表 1-10 中的式 (17) 为

$$A+B \cdot C = (A+B) \cdot (A+C)$$

将 A、B、C 所有可能的取值组合逐一代入上式的两边，算出相应的结果，即得到表 1-11 的真值表。可见，等式两边对应的真值表相同，故等式成立。

表 1-11			式 (17) 的真值表				
A	B	C	$B \cdot C$	$A+B \cdot C$	$A+B$	$A+C$	$(A+B) \cdot (A+C)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

1.3.2 若干常用公式

表 1-12 中列出了几个常用公式。这些公式是利用基本公式导出的。直接运用这些导出公式可以给化简逻辑函数的工作带来很大方便。

现将表 1-12 中的各式证明如下。

1. 式 (21) $A+A \cdot B=A$

证： $A+A \cdot B=A \cdot (1+B)=A \cdot 1=A$

上式说明在两个乘积项相加时，若其中一项以另一项为因子，则该项是多余的，可以删去。

2. 式 (22) $A+\bar{A} \cdot B=A+B$

证： $A+\bar{A} \cdot B=(A+\bar{A}) \cdot (A+B)=1 \cdot (A+B)=A+B$

这一结果表明，两个乘积项相加时，如果一项取反后是另一项的因子，则该因子是多余的，可以消去。

3. 式 (23) $A \cdot B+A \cdot \bar{B}=A$

证： $A \cdot B+A \cdot \bar{B}=A \cdot (B+\bar{B})=A \cdot 1=A$

这个公式的含意是当两个乘积项相加时，若它们分别包含 B 和 $\bar{B}$ 两个因子而其他因子相同，则两项定能合并，且可将 B 和 $\bar{B}$ 两个因子消去。

4. 式 (24) $A \cdot (A+B)=A$

证： $A \cdot (A+B)=A \cdot A+A \cdot B=A+A \cdot B$

表 1-12 若干常用公式	
序 号	公 式
21	$A+A \cdot B=A$
22	$A+\bar{A} \cdot B=A+B$
23	$A \cdot B+A \cdot \bar{B}=A$
24	$A \cdot (A+B)=A$
25	$A \cdot B+\bar{A} \cdot C+B \cdot C=A \cdot B+\bar{A} \cdot C$ $A \cdot B+\bar{A} \cdot C+BCD=A \cdot B+\bar{A} \cdot C$
26	$A \cdot \bar{A} \cdot \bar{B}=A \cdot \bar{B}; \bar{A} \cdot \bar{A} \bar{B}=\bar{A}$