

普通高等教育规划教材

数字逻辑技术基础

杨学敏 刘继承 主编

王毓银 主审

机械工业出版社

本书将加强工程素质和创新意识的培养作为教材改革的重点，对传统教材进行了尝试性的改革：精简了门电路的工作原理、小规模集成电路等内容；增加了 VHDL，并将其作为基本描述方法贯穿全书；运用数字逻辑特有的描述方法阐述器件的功能、功能扩展及应用，突出逻辑性，避免文字赘述；增强了应用的设计性、综合性和灵活性。

本书既满足原国家教育委员会颁布的课程教学基本要求，又符合当前我国高等教育教学课程体系、内容的改革和压缩学时的实际。书中不仅参考了国内外优秀教材，还融入了编者多年的教学经验。

全书共 6 章，主要内容有：数字逻辑基础，组合逻辑的分析与设计，时序逻辑的分析与设计，常用中大规模集成电路及其应用，存储器和可编程逻辑器件，数字系统设计基础。

本书可作为高等院校电子与通信专业教材，也可作为相关专业的教材和技术人员的参考书。

图书在版编目 (CIP) 数据

数字逻辑技术基础 / 杨学敏, 刘继承主编. —北京: 机械工业出版社, 2004.5

普通高等教育规划教材

ISBN 7-111-14297-7

I. 数... II. ①杨...②刘... 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 (2004) 第 028102 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 王保家 闫晓宇

责任编辑: 卢若薇 版式设计: 霍永明 责任校对: 陈延翔

封面设计: 张 静 责任印制:

印刷厂印刷·新华书店北京发行所发行

2004 年 6 月第 1 版第 1 次印刷

787mm×1092mm 1/16·17 印张·415 千字

定价: 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

普通高等教育应用型人才培养规划教材 编审委员会委员名单

主任：刘国荣 湖南工程学院
副主任：左健民 南京工程学院
陈力华 上海工程技术大学
鲍 泓 北京联合大学
王文斌 机械工业出版社

委员：(按姓氏笔画排序)

刘向东 华北航天工业学院
任淑淳 上海应用技术学院
何一鸣 常州工学院
陈文哲 福建工程学院
陈 峻 扬州大学
苏 群 黑龙江工程学院
娄炳林 湖南工程学院
梁景凯 哈尔滨工业大学(威海)
童幸生 江汉大学

电子与通信类专业分委员会

主任：鲍 泓 北京联合大学

副主任：张立臣 常州工学院

李国洪 华北航天工业学院

委员：(按姓氏笔画排序)

邓 琛 上海工程技术大学

叶树江 黑龙江工程学院

李金平 北京联合大学

沈其聪 总参通信指挥学院

杨学敏 成都理工大学

秘书长：何希才 北京联合大学

序

工程科学技术在推动人类文明的进步中一直起着发动机的作用。随着知识经济时代的到来，科学技术突飞猛进，国际竞争日趋激烈。特别是随着经济全球化发展和我国加入WTO，世界制造业将逐步向我国转移。有人认为，我国将成为世界的“制造中心”。有鉴于此，工程教育的发展也因此面临着新的机遇和挑战。

迄今为止，我国高等工程教育已为经济战线培养了数百万专门人才，为经济的发展作出了巨大的贡献。但据IMD1998年的调查，我国“人才市场上是否有充足的合格工程师”指标排名世界第36位，与我国科技人员总数排名世界第一形成很大的反差。这说明符合企业需要的工程技术人员特别是工程应用型技术人才市场供给不足。在此形势下，国家教育部近年来批准组建了一批以培养工程应用型本科人才为主的高等院校，并于2001、2002年两次举办了“应用型本科人才培养模式研讨会”，对工程应用型本科教育的办学思想和发展定位作了初步探讨。本系列教材就是在这种形势下组织编写的，以适应经济、社会发展对工程教育的新要求，满足高素质、强能力的工程应用型本科人才培养的需要。

航天工程的先驱、美国加州理工学院的冯·卡门教授有句名言：“科学家研究已有的世界，工程师创造未有的世界。”科学在于探索客观世界中存在的客观规律，所以科学强调分析，强调结论的惟一性。工程是人们综合应用科学（包括自然科学、技术科学和社会科学）理论和技术手段去改造客观世界的实践活动，所以它强调综合，强调方案优缺点的比较并作出论证和判断。这就是科学与工程的主要不同之处。这也就要求我们对工程应用型人才的培养和对科学研究型人才的培养应实施不同的培养方案，采用不同的培养模式，采用具有不同特点的教材。然而，我国目前的工程教育没有注意到这一点，而是：①过分侧重工程科学（分析）方面，轻视了工程实际训练方面，重理论，轻实践，没有足够的工程实践训练，工程教育的“学术化”倾向形成了“课题训练”的偏软现象，导致学生动手能力差。②人才培养模式、规格比较单一，课程结构不合理，知识面过窄，导致知识结构单一，所学知识中有一些内容已陈旧，交叉学科、信息科学的内容知之甚少，人文社会科学知识薄弱，学生创新能力不强。③教材单一，注重工程的科学分析，轻视工程实践能力的培养；注重理论知识的传授，轻视学生个性特别是创新精神的培养；注重教材的系统性和完整性，造成课程方面的相互重复、脱节等现象；缺乏工程应用背景，存在内容陈旧的现象。④老师缺乏工程实践经验，自身缺乏“工程训练”。⑤工程教育在实践中与经济、产业的联系不密切。要使我国工程教育适应经济、社会的发展，培养更多优秀的工程技术人员，我们必须努力改革。

组织编写本套系列教材，目的在于改革传统的高等工程教育教材，建设一套富有特色、有利于应用型人才培养的本科教材，满足工程应用型人才培养的要求。

本套系列教材的建设原则是：

1. 保证基础，确保后劲

科技的发展，要求工程技术人员必须具备终生学习的能力。为此，从内容安排上，保证学生有较厚实的基础，满足本科教学的基本要求，使学生日后具有较强的发展后劲。

2. 突出特色，强化应用

围绕培养目标，以工程应用为背景，通过理论与工程实际相结合，构建工程应用型本科教育系列教材特色。本套系列教材的内容、结构遵循如下9字方针：知识新、结构新、重应用。教材内容的要求概括为：“精”、“新”、“广”、“用”。“精”指在融会贯通教学内容的基础上，挑选出最基本的内容、方法及典型应用；“新”指将本学科前沿的新进展和有关的技术进步新成果、新应用等纳入教学内容，以适应科学技术发展的需要。妥善处理好传统内容的继承与现代内容的引进。用现代的思想、观点和方法重新认识基础内容和引入现代科技的新内容，并将这些内容按新的教学系统重新组织；“广”指在保持本学科基本体系下，处理好与相邻以及交叉学科的关系；“用”指注重理论与实际融会贯通，特别是要注入工程意识，包括经济、质量、环境等诸多因素对工程的影响。

3. 抓住重点，合理配套

工程应用型本科教育系列教材的重点是专业课（专业基础课、专业课）教材的建设，并做好与理论课教材建设同步的实践教材的建设，力争做好与之配套的电子教材的建设。

4. 精选编者，确保质量

遴选一批既具有丰富的工程实践经验，又具有丰富的教学实践经验的教师担任编写任务，以确保教材质量。

我们相信，本套系列教材的出版，对我国工程应用型人才培养质量的提高，必将产生积极作用，会为我国经济建设和社会发展作出一定的贡献。

机械工业出版社颇具魄力和眼光，高瞻远瞩，及时提出并组织编写这套系列教材，他们为编好这套系列教材做了认真细致的工作，并为该套系列教材的出版提供了许多有利的条件，在此深表衷心感谢！

编委会主任 刘国荣教授
湖南工程学院院长

前 言

《数字逻辑技术基础》是为高等学校工科电子与通信专业编写的教材。该书在满足原国家教育委员会颁布的课程教学基本要求的前提下，根据当前我国高等教育教学课程体系、内容的改革和压缩学时的实际，将加强工程素质和创新意识的培养作为教材改革的重点，对传统教材进行了尝试性的改革。

1. 重视基础，与时俱进。在保证课程基本理论系统性和基本内容完整性的原则下，精简了门电路的工作原理、小规模集成电路等传统内容。全书以 CMOS 器件为主，增加了 VHDL，并将其作为数字逻辑的基本描述方法进行介绍，贯穿全书。

2. 突出逻辑性。运用数字逻辑特有的各种描述方法阐述器件的功能、功能扩展及应用，避免文字赘述，并将其作为知识应用体系的重要组成部分。

3. 注重应用。遵循知识只有通过应用才能转变成技能并形成和发展为能力的规律，把教材的重点从器件的工作原理介绍转移到与工程设计密切相关的技术要点及应用举例上；将应用的重点从分析转移到设计上；增加应用的综合性，突出模块化设计思想；增加习题的实践性，以构成完整的知识应用体系。

4. 激发思维。注意解题方法的多样性、独特性。一法多用，如卡诺图、降维卡诺图、次态卡诺图、复合卡诺图；一题多解，如在组合逻辑和时序逻辑设计举例中，提供了四五种实现方案；增强教材的启发性，充分展现思维的灵活性、发散性，以培养读者的工程素质和创新意识。

本书融入了编者多年的教学经验，并参考国内外优秀教材编写而成。其内容既反映了当前数字逻辑技术的发展，符合当前社会对人才的需求，又符合学生的接受能力，易教、易学。本教材按 64~72 学时编写，不包括实践学时。书中标有“*”的内容为选学内容。

参加本书正文部分编写的有，杨学敏（第一、二、三、六章）、刘继承（第四、五章）、张兆莉绘制了第四、五章的部分电路图。参加本书自我检测题及答案、思考题及习题编写的有王传新（第四、五章）、曾维（第一章）、刘瑛（第二章）、杜正聪（第三章）。

北京联合大学信息学院首席教授王毓银科学严谨地审阅了全部书稿，提出了许多宝贵意见，对此我们深表感谢。由于编者的能力和水平有限，教材中一定还存在不少缺点和不足，殷切希望读者批评指正。

编 者

2004 年 2 月

目 录

序	二、译码器及其应用	55
前言	第三节 数据选择器	63
第一章 数字逻辑基础	一、数据选择器的逻辑功能	63
第一节 数制与码制	二、数据选择器的应用	66
一、进位计数制	第四节 算术运算电路	68
二、进位计数制的相互转换	一、全加器	68
三、二进制编码	二、超前进位产生器	71
第二节 逻辑运算	三、算术/逻辑单元	71
一、基本逻辑运算	第五节 数值比较器	74
二、复合逻辑运算	一、数值比较器的逻辑功能	74
三、正逻辑与负逻辑	二、数值比较器的位数扩展	75
第三节 逻辑函数及其表示方法	第六节 奇偶校验器	77
一、逻辑函数	一、奇偶校验器的逻辑功能	77
二、逻辑函数的表示方法	二、奇偶校验器位数的扩展	78
三、各种表示方法间的相互转换	第七节 组合逻辑电路的设计	79
第四节 逻辑函数的化简	一、组合逻辑电路的设计方法	79
一、逻辑函数的最简形式	二、组合逻辑电路设计举例	80
二、逻辑代数和公式化简法	小结	84
三、卡诺图化简法	自我检测题	84
四、具有无关项逻辑函数的化简	思考题和习题	85
* 五、计算机化简法	第三章 时序逻辑的分析与设计	90
第五节 数字集成电路	第一节 集成触发器	90
一、数字集成电路的分类	一、触发器的结构和工作原理	90
二、数字集成电路的主要电气指标	二、触发器的逻辑功能	100
三、门电路的电压传输特性	三、集成触发器的时间参数	102
四、集成电路的输出结构	第二节 时序逻辑电路的分析	102
小结	一、时序逻辑电路的基本概念	102
自我检测题	二、时序逻辑电路的分析方法	103
思考题和习题	三、异步信号的处理	107
第二章 组合逻辑的分析与设计	第三节 计数器	108
第一节 组合逻辑电路的分析	一、计数器的逻辑功能	108
一、组合逻辑电路的基本概念	二、任意进制计数器的构成方法	113
二、组合逻辑电路的分析方法	三、计数器的应用	118
三、组合逻辑电路中的竞争冒险	第四节 寄存器及移位寄存器	119
第二节 编码器及译码器	一、寄存器及其应用	119
一、编码器	二、移位寄存器及其应用	121

第五节 时序逻辑电路的设计	130	概念	196
一、用触发器实现同步时序逻辑电路 ...	130	一、可编程逻辑器件简介	196
二、用中规模集成电路实现同步时序逻辑	133	二、可编程逻辑器件的分类	197
小结	135	第四节 复杂可编程逻辑器件 (CPLD)	199
自我检测题	136	一、CPLD 的结构与工作原理	199
思考题和习题	139	二、CPLD 的编程举例	205
第四章 常用中大规模集成电路及其应用	145	第五节 现场可编程门阵列 (FPGA)	206
第一节 555 定时器及其应用	145	一、FPGA 的结构与工作原理	206
一、555 定时器的功能	145	二、FPGA 的配置举例	212
二、555 定时器的典型应用	146	小结	216
第二节 集成单稳态触发器	151	自我检测题	217
一、非重复触发单稳态触发器	151	思考题和习题	217
二、可重复触发单稳态触发器	153	* 第六章 数字系统设计基础	222
三、单稳态触发器的典型应用	154	第一节 数字系统的设计方法	222
第三节 数-模及模-数转换	157	一、数字系统的组成	222
一、数-模转换器 (DAC)	157	二、层次化结构设计	222
二、模-数转换器 (ADC)	165	三、自顶向下的设计方法	224
小结	177	四、VHDL 的特点	228
自我检测题	177	第二节 VHDL 程序基础	229
思考题和习题	178	一、程序的基本结构	229
第五章 存储器和可编程逻辑器件	184	二、数据类型与运算符	232
第一节 随机存取存储器 (RAM)	184	三、基本描述语句	234
一、RAM 的基本结构	184	自我检测题答案	238
二、存储器的容量扩充	188	附录	245
第二节 只读存储器 (ROM)	189	附录 A ASCII (美国标准信息交换码)	245
一、ROM 的基本结构	190	附录 B IEEE 标准程序包	246
二、集成 EPROM 简介	192	参考文献	259
第三节 可编程逻辑器件的基本			

第一章 数字逻辑基础

本章介绍数字逻辑的基础知识。首先扼要介绍二、十进制数间的相互转换、十进制数的二进制编码、补码。然后讲述分析和设计数字逻辑的基本方法——逻辑函数的表示方法及化简方法。最后介绍数字集成电路的正确使用，特别是 TTL 及 CMOS 集成电路的特点。

第一节 数制与码制

数字逻辑就其名称而言与数字有密切的关系，常用的计算机就是主要由数字逻辑电路组成的设备。计算机不但能对数值进行运算，而且还能对声音、图像、文字、符号进行处理，对过程进行控制。因此，数字逻辑中的“数字”不仅表示数值大小，而且还可以表示其他任意信息。本节仅对数值的表示方法作简要说明。

一、进位计数制

(一) 十进制计数制

为了表示数值的大小，人类祖先在长期生产劳动实践中学会了用十个指头计数，从而产生了十进制数。任意一个十进制数可以表示为

$$\begin{aligned}(D)_{10} &= K_{n-1}10^{n-1} + K_{n-2}10^{n-2} + \dots + K_010^0 + K_{-1}10^{-1} + \dots + K_{-m}10^{-m} \\ &= \sum_{i=-m}^{n-1} K_i 10^i\end{aligned}\quad (1-1)$$

式中，10 为计数制的基数，表示有 0~9 十个不同的数字符号，且逢十进位； K_i 为第 i 位的系数，可以是 0~9 十个数字符号中任意一个； 10^i 为第 i 位的权； n 、 m 为正整数， n 表示整数部分位数， m 表示小数部分位数。

例如： $(2003.9)_{10} = 2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 3 \times 10^0 + 9 \times 10^{-1}$

若以 R 取代式 (1-1) 中的 10，即可得到 R 进制数的普遍形式

$$(D)_R = \sum_{i=-m}^{n-1} K_i R^i\quad (1-2)$$

式中， R 为基数，表示有 R 个不同的数字符号，且逢 R 进位。

(二) 二进制计数制

在数字系统中广泛采用二进制计数制。这是因为二进制数的每一位只取 0 或 1，可以用具有两个不同稳定状态的电子元器件（如晶体管的饱和与截止）来表示，并且数据的存储和传送可用简单、可靠的方式进行。任意二进制数可以表示为

$$(D)_2 = \sum_{i=-m}^{n-1} K_i 2^i\quad (1-3)$$

式中 2 为基数，表示有两个数符，且逢二进位。例如

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

(三) 十六进计数制

采用二进计数制,对数字系统来说,处理、存储、传输极为方便,然而书写很不方便。因此常采用十六进计数制进行书写或打印。十六进计数制有 16 个数符,如表 1-1 所示,且逢十六进位。

表 1-1 十六进制数符

十进制数	二进制数	十六进制数	十进制数	二进制数	十六进制数
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

任意十六进制数可以表示为

$$(D)_{16} = \sum_{i=-m}^{n-1} k_i 16^i \quad (1-4)$$

例如: $(3AE.6)_{16} = 3 \times 16^2 + 10 \times 16^1 + 14 \times 16^0 + 6 \times 16^{-1}$

二、进位计数制的相互转换

人们习惯的是十进制数,数字系统采用的是二进制数,书写时采用十六进制数,因此,必然产生各种进位计数制的相互转换问题。

(一) 二进制数与十六进制数的转换

由于 $2^4 = 16$,所以一位十六进数所能表示的数值恰好相当于四位二进制数所能表示的数值,如表 1-1 所示。因此它们之间的转换极为方便。

二进制数转换成十六进制数时,整数部分自右向左每四位二进制数构成一组,最后一组不足四位时,左边用 0 补足;小数部分则自左向右每四位一组,最后不足四位时右边用 0 补足;然后用一位十六进制数符取代。例如

$$\begin{aligned} & (1011110.1011001)_2 \\ &= (0101 \ 1110 \cdot 1011 \ 0010)_2 \\ &= (5 \quad E \cdot B \quad 2)_{16} \end{aligned}$$

十六进制数转换为二进制数时,将一位十六进制数符用对应的四位二进制数取代,依次排列,然后去掉整数部分最高位的 0 和小数部分最低位的 0。例如

$$\begin{aligned} & (3 \quad A \quad E \cdot 6)_{16} \\ &= (0011 \ 1010 \ 1110 \cdot 0110)_2 \\ &= (1110101110.011)_2 \end{aligned}$$

(二) 二进制数与十进制数间的转换

1. 二进制数转换成十进制数

将二进制数按位权展开相加即可转换成对应的十进制数。二进制数各位的位权见表 1-2。例如

$$\begin{aligned}
 (11001.0101)_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} \\
 &\quad + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\
 &= 2^4 + 2^3 + 2^0 + 2^{-2} + 2^{-4} \\
 &= 16 + 8 + 1 + \frac{1}{4} + \frac{1}{16} \\
 &= (25.31)_{10}
 \end{aligned}$$

表 1-2 二进制数的位权

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2 ⁿ	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768

小数部分转换时应注意精度。上例最低位为 1/16，转换为十进制数只需到百分位。当 m 较大时，按位权展开相加进行转换很繁琐，可以采用先将小数部分乘以 2^m 变为整数进行转换后再除以 2^m。如上例小数部分

$$\begin{aligned}
 (.0101)_2 \times (10000)_2 &= (101)_2 & 2^4 &= (10000)_2 \\
 (101)_2 &= (5)_{10} & 5 \div 2^4 &= 5 \div 16 \approx 0.31
 \end{aligned}$$

由此例看出：任意二进制数乘以 2^m，等于该数小数点向右移 m 位，数位左移 m 位；除以 2^m，等于该数小数点向左移 m 位，数位右移 m 位。

2. 十进制数转换为二进制数

需将十进制数的整数部分和小数部分分别进行转换。

(1) 整数部分

1) 方法一：除基数取余，先低后高，直到商为零。可采用十六进制数作为中间过渡。例如将 (725)₁₀ 转换为二进制数。

16	725	商	余数	
16	45	←	5	最低有效位
16	2		13	
	0		2	最高有效位

转换结果：(725)₁₀ = (2D5)₁₆ = (1011010101)₂

2) 方法二：按位权相减。

725	
- 512 2 ⁹
213	
- 128 2 ⁷
85	
- 64 2 ⁶
21	
- 16 2 ⁴
5	

(5)₁₀ = (0101)₂, 则 2⁹、2⁷、2⁶、2⁴、2²、2⁰ 位取值 1, 其他位取值 0, 转换结果与方法一相同。

(2) 小数部分 乘基数取整, 先高后低, 直到所要求的精度为止。例如将 (0.787)₁₀ 转换为二进制数。十进制数的精度为千分之一, 由表 1-2 得知应取 m=10。采用十六进制数作为中间过渡, 需要 3 位。

$$\begin{array}{r}
 0.787 \qquad \qquad \qquad \text{整数部分} \\
 \times \quad 16 \\
 \hline
 12.592 \quad \dots\dots\dots 12 \quad \dots\dots\dots \text{最高位} \\
 0.592 \\
 \times \quad 16 \\
 \hline
 9.472 \quad \dots\dots\dots 9 \\
 0.472 \\
 \times \quad 16 \\
 \hline
 7.552 \quad \dots\dots\dots 7 \quad \dots\dots\dots \text{最低位}
 \end{array}$$

(0.C97)₁₆ = (0.110010010111)₂, 按所要求的精度, 只需要 10 位, 对第 11 位 0 舍 1 入

$$(0.110010010111)_2 \approx (0.1100100110)_2$$

转换结果: (.787)₁₀ = (.1100100110)₂

则 (725.787)₁₀ = (1011010101.1100100110)₂

三、二进制编码

数码不仅用于表示数值大小, 还可以用来表示许多事物。例如生活中地址的邮政编码、火车列次、飞机航班、学生学号等, 这些数码的内涵因人们约定的意义不同而异, 已没有数值大小的含义。生活中人们采用十进制编码, 而在数字系统中采用的是二进制编码。

所谓二进制编码, 即是用若干位特定的二进制数来表示数据或信息的过程。为了便于记忆和处理, 编码时总要遵循一定的规则, 称这些规则为码制; 称编码的结果 (若干位特定的二进制数) 为代码。

(一) 二—十进制 (BCD) 码

BCD 为 Binary Coded Decimal 的缩写, 即用二进制代码形式来表示十进制数符。十进制数有 10 个数符, 需用 4 位二进制代码表示。4 位二进制代码有 16 种组合, 而表示十进制数符只需要 10 种组合, 因此用 4 位二进制代码表示十进制数符有

$$A_{16}^{10} = \frac{16!}{(16-10)!} \approx 2.9 \times 10^{10}$$

种编码方案。表 1-3 列出了几种常用的 BCD 码。

表 1-3 常用的 BCD 码

十进制数	8421 码	2421 码	5211 码	余 3 码	格雷码
0	0000	0000	0000	0011	0000
1	0001	0001	0001	0100	0001
2	0010	0010	0011	0101	0011

(续)

十进制数	8421 码	2421 码	5211 码	余 3 码	格雷码
3	0011	0011	0101	0110	0010
4	0100	0100	0111	0111	0110
5	0101	1011	1000	1000	1110
6	0110	1100	1010	1001	1010
7	0111	1101	1100	1010	1000
8	1000	1110	1110	1011	1100
9	1001	1111	1111	1100	0100

8421 码的各位有固定的权，从左到右的权分别为 8、4、2、1，称为有权码。它与字符 0~9 的 ASCII 码（见附录 A）的低 4 位码相同，有利于简化输入输出过程中 BCD 码与 ASCII 码的相互转换操作。

2421 码、5211 码和余 3 码的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 相加为 1111，即具有互补性，这种特性在数字系统中是很有用处的。2421 码为有权码，从左到右的权分别为 2、4、2、1；5211 码和余 3 码为无权码。如果将余 3 码和 8421 码看成 4 位二进制数，则余 3 码是在 8421 码数值的基础上加 $(0011)_2 = (3)_{10}$ 而成，故将此代码称为余 3 码。用余 3 码进行十进制数相加时，能正确产生进位信号，并给减法运算带来方便。

格雷码又称循环码，为无权码。其任意两个相邻代码间只有一位取值不同，当数值递增或递减时，用格雷码不会发生竞争冒险现象（详见第二章第一节），从而提高了电路的工作速度、抗干扰能力和可靠性。

若将 $(725.787)_{10}$ 表示成表 1-3 所列的几种常用的 BCD 码，则

$$\begin{aligned}
 (725.787)_{10} &= (0111\ 0010\ 0101.0111\ 1000\ 0111)_{8421\text{码}} \\
 &= (1101\ 0010\ 1011.1101\ 1110\ 1101)_{2421\text{码}} \\
 &= (1100\ 0011\ 1000.1100\ 1110\ 1100)_{5211\text{码}} \\
 &= (1010\ 0101\ 1000.1010\ 1011\ 1010)_{\text{余}3\text{码}} \\
 &= (1000\ 0011\ 1110.1000\ 1100\ 1000)_{\text{格雷码}}
 \end{aligned}$$

由此看出： $(725)_{10}$ 的 BCD 码与转变为二进制数 $(1011010101.1100100110)_2$ 截然不同。前者为其“代码”，可表示成多种形式；而后者为其“值的大小”，结果是惟一的。

(二) 原码及补码

在数字系统中，数值的正、负号也用 0 和 1 表示。符号位为最高位，0 表示正号，1 表示负号，以后各位表示数值。

原码：最高位为符号位，以后各位为数值大小。

$$\begin{array}{ccc}
 (+89)_{10\text{原}} = \boxed{0} 1\ 0\ 1\ 1\ 0\ 0\ 1 & & (-89)_{10\text{原}} = \boxed{1} 1\ 0\ 1\ 1\ 0\ 0\ 1 \\
 \text{符号位} & & \text{符号位}
 \end{array}$$

补码：最高位为符号位；正数，以后各位为数值大小，与原码相同；负数，以后各位为其原码逐位取反，最低位加 1。

$$\begin{array}{ccc}
 (+89)_{10\text{补}} = \boxed{0} 1\ 0\ 1\ 1\ 0\ 0\ 1 & & (-89)_{10\text{补}} = \boxed{1} 0\ 1\ 0\ 0\ 1\ 1\ 1
 \end{array}$$

负数补码所表示数值的大小为符号位后逐位取反最低位加 1。

$$[10100111]_{\text{补}} = (-1011001)_2 = (-89)_{10}$$

采用补码可将减法运算转化为加法运算（详见第二章第四节）从而简化电路结构。

第二节 逻辑运算

当二进制数码表示数值大小时，它们之间进行的运算为算术运算。其运算规则与十进制数的运算规则基本相同，只是相邻两位之间是逢二进位，而不是逢十进位。

当二进制数码表示不同的逻辑状态时，则按指定的因果关系进行逻辑运算。逻辑运算与算术运算有本质的不同，它给出的是输出与输入间的因果关系，而不是数值运算结果的大小。逻辑运算所遵循的规则为逻辑代数。逻辑代数中也用字母表示逻辑变量，但每个逻辑变量的取值只有 0 和 1。这里的 0 和 1 不再表示数量的大小，而只表示两种不同的逻辑状态。

一、基本逻辑运算

最基本的逻辑运算有与、或、非三种。为了便于理解它们的含义，先看一个简单的例子。

图 1-1 给出了三个指示灯的控制电路。在图 a 中，只有当两个开关同时闭合时，指示灯才亮；在图 b 中，只要有任何一个开关闭合，指示灯就亮；而在图 c 中，开关断开时指示灯亮，开关闭合时指示灯反而不亮。如果把开关闭合作为条件（逻辑变量），把灯亮作为结果（逻辑函数），那么这三个电路表示了三种不同的因果（逻辑）关系。

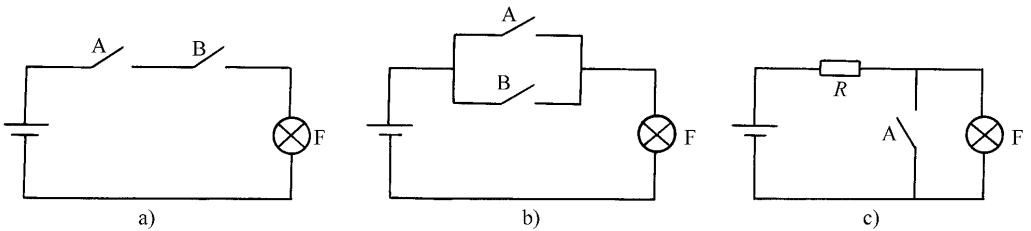


图 1-1 用于说明与、或、非关系的电路

a) 说明与关系的电路 b) 说明或关系的电路 c) 说明非关系的电路

1. 与运算

图 1-1a 表明，只有决定事物结果的全部条件同时具备时，结果才发生。这种因果关系称为与运算。

如果以 1 表示开关闭合或灯亮，以 0 表示开关断开或灯灭，可以列出输入变量（A、B）的各种取值组合与输出变量（F）的对应关系——真值表，如表 1-4 所示。

由其真值表看出，当输入变量（A、B）的取值确定后，F 的值就被唯一地确定了，称 F 是输入变量（A、B）的逻辑函数，函数表达式如表 1-4 所示。与的运算符是“·”，书写时可省略。实现逻辑与的电路称为与门，其逻辑图如表 1-4 所示，图中的“&”表示与。与门可以有多个输入变量， $F = ABCD\dots$ 。

由其真值表看出，与运算的规则为

$$0 \cdot 0 = 0 \quad 0 \cdot 1 = 0 \quad 1 \cdot 0 = 0 \quad 1 \cdot 1 = 1 \quad F \text{ 为 } 1 \text{ 的条件是 } A、B \text{ 同时为 } 1。$$

由此可以推出一般形式

$$A \cdot 0 = 0 \quad A \cdot 1 = A \quad A \cdot A = A$$

2. 或运算

图 1-1b 表明，在决定事物结果的诸条件中只要有一个满足，结果就会发生。这种因果关系称为或运算。表 1-4 中给出了或运算的真值表和函数表达式，或的运算符是“+”。

实现逻辑或的电路称为或门，其逻辑图如表 1-4 所示，图中的“ ≥ 1 ”表示或。或门可以有多个输入变量， $F = A + B + C + D + \dots$ 。

由其真值表看出，或运算的规则为

$$0 + 0 = 0 \quad 0 + 1 = 1 \quad 1 + 0 = 1 \quad 1 + 1 = 1 \quad F \text{ 为 } 1 \text{ 的条件是 } A \text{ 或 } B \text{ 任一为 } 1。$$

由此可以推出一般形式

$$A + 0 = A \quad A + 1 = 1 \quad A + A = A$$

3. 非运算

图 1-1c 表明，条件具备时，结果不发生；而条件不具备时，结果一定发生。这种因果关系称为非运算。表 1-4 给出了其真值表和函数表达式，非的运算符是逻辑变量上方加“-”，如 \bar{A} 读作“A 非”。

实现逻辑非的电路称为非门，其逻辑图如表 1-4 所示，图中的小圆圈“。”表示非；“1”表示只有一个输入变量。

由其真值表看出，非运算的规则为

$$\bar{0} = 1 \quad \bar{1} = 0 \quad F \text{ 为 } 1 \text{ 的条件是 } A \text{ 为 } 0, A \text{ 为 } 1 \text{ 时 } F \text{ 为 } 0。$$

由此可以推出一般形式

$$A \cdot \bar{A} = 0 \quad A + \bar{A} = 1 \quad \bar{\bar{A}} = A$$

二、复合逻辑运算

实际的逻辑问题往往比与、或、非复杂得多，需要用与、或、非的组合来实现。最常见的复合逻辑运算有与非、或非、异或、同或、与或非等。

1. 与非、或非运算

由表 1-4 的真值表看出：与非运算是先与后非；或非运算是先或后非。实现与非、或非运算的电路称为与非门、或非门，其函数表达式、逻辑图如表 1-4 所示。其基本运算规则可以用列真值表的方法进行验证，如果等式成立，那么将任何一组变量的取值代入等号两边所得的结果应该相等。如验证 $\overline{AB} = \bar{A} + \bar{B}$ 的真值表如表 1-5 所示。

表 1-4 基本的逻辑运算

表示方法 逻辑运算	逻辑图		表达式	函数式	真值表			基本运算 规则
	我国标准	国外常用		VHDL	A	B	F	
与			$F = A \cdot B = AB$ $F \leq A \text{ AND } B$		0	0	0	$A \cdot A = A$ $A \cdot \bar{A} = 0$ $A \cdot 1 = A$ $A \cdot 0 = 0$

(续)

表示方法 逻辑运算	逻辑图		表达式	函数式 VHDL	真值表			基本运算 规则
	我国标准	国外常用			A	B	F	
或			$F = A + B$ $F \leq A \text{ OR } B$		0 0 0 1 1 0 1 1	0 1 1 1	$A + A = A$ $A + \bar{A} = 1$ $A + 1 = 1$ $A + 0 = A$	
非			$F = \bar{A}$ $F \leq \text{NOT } A$		0 1	1 0	$\bar{\bar{A}} = A$	
与非			$F = \overline{AB}$ $F \leq A \text{ NAND } B$		0 0 0 1 1 0 1 1	1 1 1 0	$\overline{AB} = \bar{A} + \bar{B}$	
或非			$F = \overline{A + B}$ $F \leq A \text{ NOR } B$		0 0 0 1 1 0 1 1	1 0 0 0	$\overline{A + B} = \bar{A} \bar{B}$	
异或			$F = A \oplus B$ $F \leq A \text{ XOR } B$		0 0 0 1 1 0 1 1	0 1 1 0	$A \oplus 0 = A$ $A \oplus 1 = \bar{A}$ $A \oplus A = 0$ $A \oplus \bar{A} = 1$	
同或			$F = A \odot B$ $F \leq A \text{ XNOR } B$		0 0 0 1 1 0 1 1	1 0 0 1	$A \odot 0 = \bar{A}$ $A \odot 1 = A$ $A \odot A = 1$ $A \odot \bar{A} = 0$	
与或非			$F = \overline{AB + CD}$ $F \leq \text{NOT}((A \text{ AND } B) \text{ OR } (C \text{ AND } D))$		略			

2. 异或、同或运算

(1) 异或运算 由表 1-4 的真值表看出，异或运算的逻辑关系是：只有逻辑变量 A 和 B 的取值不同时，逻辑函数的输出才为 1。由表 1-4 中的函数表达式看出，异或的运算符是“ \oplus ”。异或也可以用与、或、非的组合表示，即

$$A \oplus B = \bar{A}B + A\bar{B}$$

表 1-5 $\overline{AB} = \bar{A} + \bar{B}$ 的真值表

A	B	\overline{AB}	\bar{A}	\bar{B}	$\bar{A} + \bar{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

实现异或运算的电路称为异或门，其逻辑图如表 1-4 所示，图中的“=1”表示异或。

由真值表看出，异或运算的规则为 www.ertongbook.com