

高等院校电子信息类专业规划教材

数字电子技术基础

SHUZI DIANZI JISHU JICHU

主 编 吴 拓

副主编 易亚军 罗 俊



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

高等院校电子信息类专业规划教材

数字电子技术基础

吴 拓 主编

易亚军 罗 俊 副主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书为适应应用型本科教育的人才培养目标而编写的。主要内容有数字逻辑基础、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲波形的产生与整形、半导体存储器与可编程逻辑器件、数模和模数转换及数字电子电路综合设计。内容简明扼要、深入浅出,重点突出、实用性强,以能力培养为重点,理论分析与应用实例相配合,以提高学生分析问题和解决问题的能力。

本书可作为高等院校电子信息类、自动化类、计算机类及机电一体化类等专业的专业基础课教材,也可供其他从事电子技术工作的工程技术人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

数字电子技术基础/吴拓主编. —北京:电子工业出版社,2016.11

ISBN 978-7-121-29023-7

I. ①数… II. ①吴… 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(2016)第128741号

策划编辑:朱怀永

责任编辑:郝黎明

印 刷:三河市良远印务有限公司

装 订:三河市良远印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×1092 1/16 印张:18 字数:518.4千字

版 次:2016年11月第1版

印 次:2016年11月第1次印刷

定 价:39.80元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010)88254608。

前 言

当今时代是一个数字化时代,数字电子技术作为国家科学技术和国民经济发展的基础,在电子计算机、电子通信、工业控制、交通管制及家用电器等各个领域,有着无处不在、举足轻重的地位。《数字电子技术基础》是电子信息类、机电一体化类、自动化类、计算机类等专业的专业基础课程,也是实践性很强的技术基础课程,随着数字化和信息化技术的飞速发展,该课程对于培养学生的专业技能和创新能力也凸显出更为重要的作用。

本书的编写是为了适应电子信息产业发展的需要,培养应用型本科专业技术人才而编写的。本书具有以下特点。

1. 培养具有较强职业能力的应用型人才的目标明确

遵循教育部对“电子技术基础课程”的基本要求,本着“宽口径、厚基础、重实践,够用为度”的原则,本教材围绕“培养有一定理论基础,有较强实操能力,有足够创新意识的应用型人才”这一目标,整合和优化教学内容。

2. 实施以能力培养为重点的“教-学-做”一体化教学模式

为了达到理论联系实际、增强实践教学环节的目的,本书实施“教-学-做”一体化教学模式,按照四大模块即清晰的教学导航、系统的理论知识、足够的实操训练、必要的应用示例来编写,知识目标和能力目标明确、重点和难点突出,以期达到以能力培养为重点的最佳效果。

3. 加强实操训练以满足岗位就业对专业技能的需要

为了满足许多岗位对就业人员的数字电子技术与技能提出的新要求,本书精心编录了12个数字电子技术实操训练项目,期望通过实操训练项目调动和激发学生自主获取相关知识和技能积极性、主动性。

4. 内容新、精、齐,理论分析与应用实例相配合

编者力求出精品、力求创新,力求内容系统、科学、全面,吸收科技最新成果,对重点内容尤其是与生产实际联系紧密的内容进行细致的推理解析,理论分析与应用实例相配合,以期提高学生分析问题和解决问题的能力。

本书的主要内容有:数字逻辑基础、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲波形的产生与整形、半导体存储器与可编程逻辑器件、数模和模数转换及数字电子电路综合设计,书中配有一定量的思考题与习题并附有部分习题答案。内容简明扼要,深入浅出,思路清晰,重点突出,实用性强。

本书由吴拓教授任主编,易亚军、罗俊任副主编;第1、2章由吴拓编写,第3、4、9章由罗俊编写,第5、6章由易亚军编写,第7、8章由宋宗峰编写、各章的实操训练由杨博编写;全书由吴拓教授进行统稿和定稿。

本书的编写得到了各界同仁的大力支持,在此谨致深深的谢意!

由于编者水平有限,书中难免存在疏漏之处,恳请广大读者不吝指正。

编 者

2016年3月于广州

目 录

第 1 章 数字逻辑基础	1	2.3.1 TTL 反相器电路结构及原理	41
1.1 数字信息与数字电路	1	2.3.2 TTL 反相器的电压传输特性	
1.1.1 数字信息	1	和抗干扰能力	44
1.1.2 数字电路	2	2.3.3 TTL 反相器的静态输入特性、输出	
1.2 数制与码制	4	特性和负载能力	46
1.2.1 几种常用的数制	4	2.3.4 TTL 反相器的动态特性	51
1.2.2 不同数制间的转换	6	2.3.5 TTL 门电路的其他类型	53
1.2.3 几种常用的码制	8	2.3.6 TTL 集成门系列简介	56
1.3 逻辑代数	10	2.4 CMOS 集成门电路	58
1.3.1 逻辑代数中 3 种基本运算	10	2.4.1 CMOS 反相器的电路结构和工作	
1.3.2 复合逻辑运算	12	原理	61
1.3.3 逻辑代数的基本公式	14	2.4.2 CMOS 反相器的特性及参数	62
1.3.4 逻辑代数的基本定理	14	2.4.3 CMOS 门电路的其他类型	64
1.3.5 逻辑代数的应用实例	15	2.4.4 CMOS 集成门系列简介	66
1.4 逻辑函数及其表示方法	16	2.5 可以“线与”的集成门电路	68
1.4.1 逻辑函数的定义	16	2.5.1 集电极开路门 (OC 门)	69
1.4.2 逻辑函数的表示方法	16	2.5.2 三态输出门 (TS 门)	70
1.4.3 各种表示方法间的相互转换	17	2.5.3 漏极开路门 (OD 门)	72
1.5 逻辑函数的化简	18	2.6 集成门电路的应用及其应注意	
1.5.1 逻辑函数的最简形式	18	的问题	72
1.5.2 公式化简法	19	2.6.1 集成门电路的应用	72
1.5.3 卡诺图化简法	20	2.6.2 集成门电路使用应注意的问题	74
思考与练习题	28	2.6.3 TTL 电路与 CMOS 电路之间	
第 2 章 逻辑门电路	31	的接口问题	76
2.1 二极管、三极管和场效应管		2.7 实操训练	78
的开关特性	31	实操训练一: “与非门”的功能测试	
2.1.1 二极管的开关特性	31	及应用	78
2.1.2 三极管的开关特性	33	思考与练习题	83
2.1.3 场效应管的开关特性	36	第 3 章 组合逻辑电路	86
2.2 分立元件门电路	38	3.1 组合逻辑电路概述	86
2.2.1 二极管门电路	38	3.2 组合逻辑电路的分析与设计	87
2.2.2 三极管反相器	39	3.2.1 组合逻辑电路的分析	87
2.2.3 与非门电路	40	3.2.2 组合逻辑电路的设计	88
2.3 TTL 集成门电路	41	3.3 常用组合逻辑电路	89

3.3.1 编码器	89	4.7.2 触发器的功能转换	123
3.3.2 译码器	91	4.8 实操训练	124
3.3.3 数据选择器	93	实操训练五: 触发器的功能测试	
3.3.4 加法器	94	及应用	124
3.3.5 数值比较器	96	思考与练习题	127
3.4 用中规模集成电路设计组合逻辑		第 5 章 时序逻辑电路	130
电路	96	5.1 时序逻辑电路的基本概念	130
3.4.1 用译码器设计组合逻辑电路 ..	96	5.1.1 时序逻辑电路的分类	130
3.4.2 用数据选择器设计组合逻辑		5.1.2 时序逻辑电路的基本结构和	
电路	97	描述方法	131
3.4.3 用加法器设计组合逻辑电路 ..	98	5.2 时序电路的分析方法	131
3.5 组合逻辑电路的竞争-冒险现象 ..	99	5.2.1 同步时序电路的分析方法	131
3.5.1 竞争-冒险的概念及其产生		5.2.2 异步时序电路的分析方法	135
原因	99	5.3 寄存器和移位寄存器	137
3.5.2 消除竞争-冒险的方法	100	5.3.1 寄存器和移位寄存器结构组成	
3.6 实操训练	101	及工作原理	137
实操训练二: SSI 组合逻辑电路	101	5.3.2 集成(移位)寄存器及其	
实操训练三: MSI 组合逻辑		应用	141
电路(一)	102	5.4 计数器	143
实操训练四: MSI 组合逻辑		5.4.1 同步计数器结构组成及原理 ..	144
电路(二)	104	5.4.2 异步计数器结构组成及原理 ..	149
思考与练习题	106	5.4.3 集成计数器及其应用	153
第 4 章 触发器	108	5.5 时序电路的设计方法	162
4.1 概述	108	5.6 应用实例	167
4.2 基本 RS 触发器	109	5.7 实操训练	170
4.2.1 由与非门构成的基本 RS		实操训练六: SSI 时序逻辑电路	170
触发器	109	实操训练七: MSI 时序逻辑电路——	
4.2.2 由或非门构成的基本 RS		计数器	172
触发器	111	实操训练八: MSI 时序逻辑电路——	
4.3 同步触发器	112	移位寄存器	174
4.3.1 同步 RS 触发器	112	思考与练习题	175
4.3.2 同步 D 触发器	114	第 6 章 脉冲波形的产生与整形	180
4.4 主从触发器(脉冲触发)	115	6.1 概述	180
4.4.1 主从 RS 触发器	115	6.2 多谐振荡器	180
4.4.2 主从 JK 触发器	116	6.2.1 非对称式多谐振荡器	181
4.5 边沿触发器(边沿触发)	118	6.2.2 对称式多谐振荡器	182
4.6 触发器的逻辑功能及描述方法	119	6.2.3 石英晶体多谐振荡器	183
4.7 集成触发器	122	6.3 施密特触发器	184
4.7.1 常用集成触发器	122	6.3.1 施密特触发器的功能	184

6.3.2	由 CMOS 门构成的施密特触发器	184
6.3.3	集成施密特触发器	186
6.3.4	施密特触发器的应用	187
6.4	单稳态触发器	189
6.4.1	用门电路组成的单稳态触发器	189
6.4.2	集成单稳态触发器	191
6.4.3	单稳态触发器的应用	193
6.5	555 定时器及其应用	193
6.5.1	555 定时器的电路组成与功能	193
6.5.2	用 555 定时器构成施密特触发器	195
6.5.3	用 555 定时器构成多谐振荡器	197
6.5.4	用 555 定时器构成单稳态触发器	199
6.6	应用实例	201
6.7	实操训练	202
	实操训练九: 脉冲波形的变换与产生	202
	实操训练十: 555 集成定时器	204
	思考与练习题	206
第 7 章	半导体存储器与可编程逻辑器件	211
7.1	概述	211
7.2	随机存储器 RAM	212
7.2.1	RAM 存储单元	212
7.2.2	RAM 的结构	214
7.2.3	RAM 的扩展	215
7.3	只读存储器 ROM	216
7.3.1	固定 ROM	216
7.3.2	可编程只读存储器 PROM	218
7.3.3	现代常用 ROM	219
7.4	可编程逻辑器件	219
7.4.1	PLD 基本原理	220
7.4.2	PLD 分类	223
7.5	高密度可编程逻辑器件	224

7.5.1	复杂可编程逻辑器件 CPLD	224
7.5.2	现场可编程门阵列 FPGA	224
7.5.3	基于芯片的设计方法	225
	思考与练习题	226

第 8 章	数模和模数转换	227
8.1	概述	227
8.2	数模转换器 (DAC)	228
8.2.1	DAC 的基本原理	228
8.2.2	倒 T 形电阻网络 DAC	228
8.2.3	权电阻网络 DAC	230
8.2.4	数模转换输出极性的扩展	231
8.2.5	DAC 的主要技术参数	232
8.2.6	集成 DAC	233
8.3	模数转换器 (ADC)	233
8.3.1	并联比较型 ADC	234
8.3.2	逐次比较型 ADC	236
8.3.3	双积分型 ADC	238
8.3.4	ADC 的主要技术参数	240
8.3.5	集成 ADC	241
8.4	取样-保持电路	242
8.5	应用实例	243
8.6	实操训练	244
	实操训练十一: 数/模转换器及应用	244
	实操训练十二: 模/数转换器及应用	246
	思考与练习题	248

第 9 章	数字电子电路综合设计	250
9.1	数字电子电路的设计方法	250
9.1.1	自顶向下设计方法	250
9.1.2	试凑设计法	251
9.1.3	数字电子电路系统分析及设计的一般步骤	252
9.2	数字电子电路的设计举例	252
9.2.1	十字路口交通管理器	252
9.2.2	数字频率计的设计与制作	261
9.3	课程设计参考题目	265

附录 A	部分习题答案	267
	参考文献	280

第1章 数字逻辑基础

教学导航

知识目标

1. 了解数字电路、数字信息的特点。
2. 了解数制和码制，掌握各种数制间的转换。
3. 掌握常见逻辑运算；掌握运用真值表、逻辑函数式、逻辑图和卡诺图等逻辑函数的表示方法。

能力目标

1. 通过本章内容的学习，能够熟练进行数制转换。
2. 通过本章内容的学习，能够熟练运用公式法和卡诺图法化简逻辑函数。

学习重点

1. 逻辑函数不同表示方法之间的转换；
2. 公式法和卡诺图法化简逻辑函数。

学习难点

逻辑函数的化简。

1.1 数字信息与数字电路

1.1.1 数字信息

物理信息有电、声、光、磁等多种形式。在物理信息处理中，应用较多的是电信息。研究电信息的产生与处理的技术就是电子技术。电子电路中的信息可分为两大类，即模拟信息和数字信息。

模拟信息是一种在时间上、数值上均连续变化的电信息，在一定范围内可以取任意实数值，如图 1-1(a)所示。这样的信息可以是由温度、声音、压力、流量、速度等转换出的电信息。传输和处理模拟信息的电路称为模拟电路，模拟电路主要研究的是如何不失真地放大模拟信息。

数字信息是在时间上、数值上均断续变化的离散的电信息，如图 1-1(b)所示。通常用高电平和低电平表示数字信息的两种状态，或用 1 和 0 表示。通常用 1 表示高电平，用 0 表示低电平，称为正逻辑。本书均采用正逻辑；也可以用 0 表示高电平，用 1 表示低电平，称为负逻辑。数字信息是由 0 和 1 以不同的组合形式构成的，每一种形式代表一种含义。

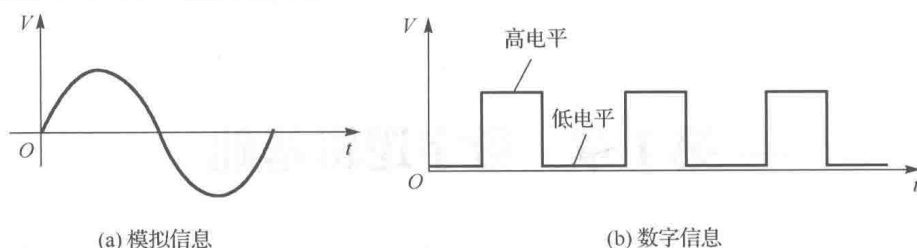


图 1-1 数字信息和模拟信息

图 1-2 所示为工业流水生产线上记录工件个数的计数系统输出信息。在流水线的一侧放置一个光源，另一侧放置一个接收装置。当工件通过光源时，光源被遮挡，接收装置接收不到光源信息；没有工件通过时，接收装置则接收到光源信息。接收装置把光源信息转换成电信号，其输出信息如图 1-2 所示。输出信号为高电平时表示没有工件通过，输出信号为低电平时表示有工件通过。若准确记录输出信号低电平的个数就能记录工件的个数。由此可见，电路只要能够准确区分高、低电平即可。高、低电平并不是某一个电压值，而是指一个电压范围，如 $2\sim 4\text{V}$ 为高电平， $0\sim 0.8\text{V}$ 为低电平。

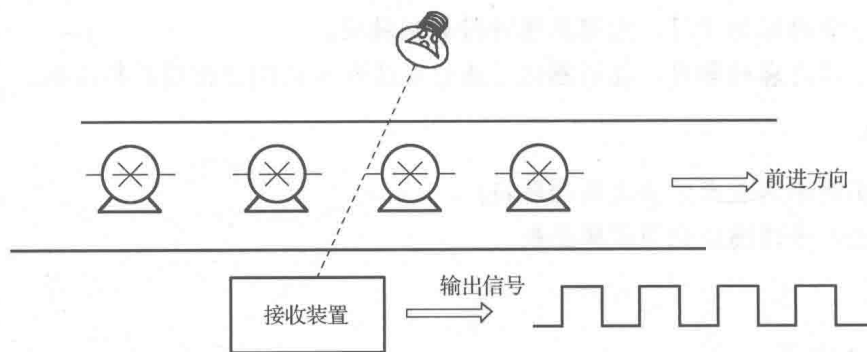


图 1-2 工业流水线计数系统

1.1.2 数字电路

传输和处理数字信息的电路称为数字电路。数字电路的输入、输出都是如图 1-2 中所示的数字信息。

1. 数字电路的特点

数字电路可以实现数字信息的存储、处理和传输。由于数字信息一般只有 0 和 1 两种状态，因此数字电路具有如下特点。

(1) 数字电路在稳态时，电子元器件工作在开关状态。数字电路和模拟电路一样，都是由二极管、三极管等元器件构成的。数字电路高、低电平的转换犹如一个开关。数字电路中，采用二极管的导通与截止或三极管的饱和与截止来实现开关的通与断。这两种对立的状态，分别用“1”和“0”表示。

(2) 数字电路的基本单元电路只有 0 和 1 两个状态。数字电路基本单元电路只要能可靠地区分高、低电平即可。高、低电平都有一个允许的变化范围，只有当干扰信息相当强烈，超出了允许的高、低电平范围，才有可能改变元件的工作状态，所以数字电路抗干扰能力强、

可靠性高。数字电路无论多么复杂,都是由几种最基本的单元电路组成,而且对元件精度要求不高,允许有较大误差,因此数字电路的单元电路结构简单、便于集成、成本低廉、使用方便。

(3) 数字电路的分析和设计应用的主要工具是逻辑代数。数字电路研究的是输入和输出的逻辑关系,包括逻辑分析和逻辑设计,所以数字电路也称为逻辑电路。分析和设计数字电路以逻辑代数为工具,利用真值表、逻辑表达式、卡诺图和波形图等来表示电路的逻辑功能。

(4) 数字电路可以大规模集成,运行速度快、功耗低、可编程。随着半导体制造工艺的不断发展,数字器件的集成度越来越高,加之可编程逻辑器件的应用,提高了使用的灵活性,并大大缩短了数字电路的研发周期。

(5) 数字电路不仅能完成数值运算,还能进行逻辑推理和逻辑判断。因此可以用来制造数控装置、智能仪表、数字通信设备及电子计算机等现代化的高科技产品。

(6) 数字电路的数字信息需要进行转换。在自然界中大多数信号都是模拟信息,当这些信息需要用数字电路进行处理时,就要进行模拟信息到数字信息的转换。由于数字电路输出的也是数字信息,数字信息不能直接回到自然界中,因此数字电路输出的数字信息需要转换成模拟信息,才能重新被利用。数字信息和模拟信息之间的转换可以通过数模(D/I)转换器和模数转换器(I/D)来实现。

2. 数字电路的应用

数字电路由于具有集成度高、运行速度快、功耗低、抗干扰能力强、可靠性高、便于长期存储、可编程、使用灵活等优点,因而在电子计算机、通信设备、自动控制、雷达、电机、家用电器、电子小产品、智能仪表、数控设备及汽车电子等领域得到了广泛的应用。数字电路的应用和发展极大地改变了人们生产、生活方式。

3. 数字电路的分类

数字电路经历了电子管、半导体分立器件到集成电路的发展历程,发展极其迅速,目前广泛使用的是数字集成电路。数字集成电路是把数字电路的基本单元逻辑电路集成到一块半导体芯片上。数字电路的种类繁多,大致可以分为以下几类:

(1) 按电路的组成分类。分为分立元件和集成电路两类。分立元件现已基本上被集成电路取代。数字集成电路按集成度的大小,可分为小规模集成电路(Small Scale Integration, SSI)、中规模集成电路(Medium Scale Integration, MSI)、大规模集成电路(Large Scale Integration, LSI)和超大规模集成电路(Very Large Scale Integration, VLSI)等。集成度是指在一张硅片上集成逻辑门或器件数量的多少。

(2) 按电路使用的器件分类。按使用的器件不同,数字电路可分为双极型电路和单极型电路。双极型电路是由二极管、三极管双极型器件构成的电路,包括TTL等多种。单极型电路是由半导体场效应管单极型器件构成的电路,包括CMOS、NMOS、PMOS等类型。

(3) 按逻辑功能分类。按逻辑功能不同,数字电路可分为组合逻辑电路和时序逻辑电路。组合逻辑电路是指输出只与输入状态的组合有关,而与先前的状态无关的电路,具有无记忆性。时序逻辑电路是指输出不仅与输入状态的组合有关,而且与电路原状态有关的电路,具有记忆性。

1.2 数制与码制

数字信息通常都是用数码的形式给出的。不同的数码可以用来表示数量大小不同的事物，这时仅用一位数码往往不够，因而需要遵循一定的规则组成多位数码使用。

1.2.1 几种常用的数制

数制是人们用以计数的方法。在日常生活中，人们习惯使用十进制数，而在数字电路中常使用二进制数、八进制数或十六进制数。在进位计数制中，包含基数（也称为模）和位权两个基本因素。

基数是指计数制中用到的数字符号个数。例如，十进制数由 0~9 共 10 个不同的数字符号组成，所以它的基数是 10。一般来说，在 N 位计数制中，包含 0, 1, 2, 3, ..., $(N-1)$ 个数字符号，进位规律是“逢 N 进 1”，即每一位计满 N 就向高位进 1，简称 N 进制。

计数时，一般采用位置计数法。位权就是指在一个进位计数制表示的数中，处于不同位置的数字代表不同的数值。某一位的数值是由这一位数字的值乘以处于该位的一个固定常数。不同位上的固定常数称为位权值，简称权值或权。不同位有不同的权值，如十进制数个位的权值是 10^0 ，十位的权值是 10^1 ，百位的权值是 10^2 。

1. 十进制 (Decimal)

十进制是以 10 为基数的计数体制。它的基数是 10，其进位规律为“逢 10 进 1”或“借 1 当 10”。任意一个十进制数都可以按权展开，写成以 10 为底的幂的和形式，例如

$$2135.76 = 2 \times 10^3 + 1 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 + 7 \times 10^{-1} + 6 \times 10^{-2}$$

式中， 10^3 、 10^2 、 10^1 、 10^0 、 10^{-1} 、 10^{-2} 分别为千位、百位、十位、个位、小数点右第一位、小数点右第二位的权值，也就是相应位所代表的实际数值。由此可见位数越高，权值越大，相邻高位权值是相邻低位权值的 10 倍。任意十进制数可表示为

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

式中， k_i 为第 i 位的系数，它可以是 0~9 这 10 个数字符号中的任意一个。若整数部分的位数是 n ，小数部分的位数是 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，则可得到任意进制 (N 进制) 数的普遍形式

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

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

2. 二进制 (Binary)

十进制是人们最习惯的计数方法，但却很难用电路来实现，因此在计数电路中一般不直接使用十进制。目前在数字电路中使用最广泛的是二进制。它只由两个数字符号 0 和 1 组成。它同十进制数一样，自左到右由高位到低位排列。计数规律为“逢 2 进 1”或“借 1 当 2”。

任何一个二进制数都可以展开为 $(D)_2 = \sum k_i \times 2^i$ 。同十进制数一样，每个数字处于不同

位代表不同的数值。例如，二进制数 1101.01 所代表的十进制数是

$$(1101.01)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (13.25)_{10}$$

式中， 2^3 、 2^2 、 2^1 、 2^0 、 2^{-1} 、 2^{-2} 分别为相应位的权值，相邻高位权值是相邻低位权值的 2 倍。

上式中分别使用下标 2 和 10 表示括号里的数是二进制数和十进制数。有时也用 B 和 D 代替 2 和 10 这两个下标。

(1) 二进制的优点。二进制相对十进制来说有如下优点。

① 二进制数只有两个数字符号 0 和 1，因此很容易用电路元件的开关状态来表示。例如，二极管的导通与截止、三极管的截止与饱和、电平的高与低等，都可以将其中一个状态规定为 0，另一个状态规定为 1，以表示二进制数。这种表示简单方便，所用元件数目少，存储和传送也十分可靠。而要使一个电路或电子器件具有严格区分的 10 个状态来对应于十进制数中的 10 个不同的数字符号是相当困难的，因此在计数电路中一般不直接使用十进制。

② 二进制的基本运算规则同十进制运算规则相似，但要简单得多。例如，两个一位十进制数相乘，其规律要用“九九乘法表”才能表示，而两个一位二进制数相乘，其乘法规律只有四种组合，即 $0 \times 0 = 0$ 、 $0 \times 1 = 1 \times 0 = 0$ 、 $1 \times 1 = 1$ 。运算规则简单，必然使运算电路和控制电路简化，进而设备也可以很简单。

由于上述优点，目前在数字系统和计算机中几乎全部采用二进制数。

(2) 二进制的缺点。相对十进制来说，二进制也有它的缺点。

① 日常生活中二进制使用较少，使用时通常要进行二-十进制的转换。使用数字系统运算时，通常要先将人们熟悉的十进制原始数据转换成二进制数，运算结束后再将其转换成人们常用的十进制数，才能方便人们进行逻辑思维、判断和记忆。

② 表示同样大小的一个数，二进制数要比十进制数位数多得多。例如，两位的十进制数 91 变为二进制数为 1011011，需要 7 位。为了减少位数，又方便表示，也常采用八进制和十六进制。

3. 八进制 (Octal) 和十六进制 (Hexadecimal)

由于二进制数比十进制数位数多，不便于书写和记忆，因此经常用八进制数和十六进制数来表示二进制数。

(1) 八进制

八进制数是以 8 为基数的计数体制，有 0、1、2、3、4、5、6、7 共 8 个数字符号，计数规律为“逢 8 进 1”或“借 1 当 8”。每一个数字处在不同位代表不同的数值，如， $(106)_8 = 1 \times 8^2 + 0 \times 8^1 + 6 \times 8^0 = (70)_{10}$ ，该八进制数可表示为

$$(D)_8 = \sum k_i \times 8^i \quad (1-3)$$

式中， k_i 为基数 8 的 i 次幂的系数，它可以是 0~7 这 8 个数字中的任意一个。

(2) 十六进制

十六进制数有 0~9、A、B、C、D、E、F 共 16 个数字符号，其中 A~F 分别相当于十进制数中的 10~15；计数规律为“逢 16 进 1”或“借 1 当 16”。同样，每一个数字处在不同位代表不同的数值。例如， $(3BE)_{16} = 3 \times 16^2 + 11 \times 16^1 + 14 \times 16^0 = (958)_{10}$ ，该十六进制数可表示为

$$(D)_{16} = \sum k_i \times 16^i$$

(1-4)

式中， k_i 为基数 16 的 i 次幂的系数，它可以是 0~F 这 16 个数字中的任意一个。

目前微型计算机中普遍采用 8 位、16 位、32 位和 64 位二进制数并行运算，而 8 位、16 位、32 位和 64 位二进制数可以用 1 位、2 位、4 位和 8 位十六进制数表示，因而用十六进制符号编写程序很方便。

表 1-1 所示为十进制数 0~15 与等值二进制数、八进制数和十六进制数的对照表。

表 1-1 不同进制数的对照表

十进制数	二进制数	八进制数	十六进制数
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

1.2.2 不同数制间的转换

1. 二进制数与十进制数之间的相互转换

(1) 二进制数转换成十进制数

二进制数转换成十进制数比较简单，只要把二进制数按权值展开，将各项的数值按十进制相加，就可得到等值十进制数。例如

$$(1001.01)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (9.25)_{10}$$

(2) 十进制数转换成二进制数

十进制数转换成二进制、八进制和十六进制数的可以分两步：

① 整数部分采用“除基取余”法，即“除以基数，取其余数作为系数，由低到高，直至商为 0 为止”。

② 小数部分采用“乘基取整”法，即“乘以基数，取其整数作为系数，由高到低，直至纯小数部分为 0 或到一定精度为止”

十进制数转换成二进制数，按上述原则分两步为：

① 整数的转换。任何十进制数的整数部分可用辗转除以 2 取余法转换成二进制数，其原理如下。

若某一个十进制数 N 可转换为 m 位二进制数, 即

$$(N)_{10} = (K_{m-1}K_{m-2}\cdots K_1K_0)_2$$

将二进制数按权值展开, 其多项式表示为

$$\begin{aligned}(N)_{10} &= (K_{m-1}K_{m-2}\cdots K_1K_0)_2 = (K_{m-1}\times 2^{m-1} + K_{m-2}\times 2^{m-2} + \cdots + K_1\times 2^1 + K_0\times 2^0)_{10} \\ &= [2\times(K_{m-1}\times 2^{m-2} + K_{m-2}\times 2^{m-3} + \cdots + K_1) + K_0]_{10}\end{aligned}$$

用除以 2, 则 $(N)_{10} \div 2 = K_{m-1}\times 2^{m-2} + K_{m-2}\times 2^{m-3} + \cdots + K_2\times 2 + K_1$ 余数为 K_0

商再除以 2 得 $(K_{m-1}\times 2^{m-2} + K_{m-2}\times 2^{m-3} + \cdots + K_2\times 2 + K_1) \div 2$

$$= K_{m-1}\times 2^{m-3} + K_{m-2}\times 2^{m-4} + \cdots + K_2 \quad \text{.....余数为 } K_1$$

依次不断用前次的商再除以 2, 直至 $K_{m-1} \div 2$ 余数为 K_{m-1}

可见, 每次除以 2 所得的余数就是十进制数 N 对应的二进制数的系数 $K_{m-1}K_{m-2}\cdots K_1K_0$ 。

例如, 将十进制数 17 转换为二进制, 则

	余数	
$2 \overline{)17}$	1	(K_0)
$2 \overline{)8}$	0	(K_1)
$2 \overline{)4}$	0	(K_2)
$2 \overline{)2}$	0	(K_3)
1	1	(K_4)

读取顺序

解得 $(17)_{10} = (10001)_2 = (K_4 K_3 K_2 K_1 K_0)_2$ 。

② 小数部分的转换。小数部分可用“乘 2 取整”的方法, 求得相应的二进制数。

若 N 是一个十进制小数, 对应的二进制小数多项式表示为

$$(N)_{10} = K_{-1}\times 2^{-1} + K_{-2}\times 2^{-2} + \cdots + K_{-m}\times 2^{-m}$$

将上式两边同时乘以 2, 得

$$2\times(N)_{10} = K_{-1} + K_{-2}\times 2^{-1} + K_{-3}\times 2^{-2} \cdots + K_{-m}\times 2^{-m+1} = K_{-1} + N_1$$

$$(N_1 = K_{-2}\times 2^{-1} + K_{-3}\times 2^{-2} \cdots + K_{-m}\times 2^{-m+1})$$

可知上式的整数部分为 K_{-1} , 将其小数部分 N_1 再乘以 2, 得

$$2N_1 = K_{-2} + K_{-3}\times 2^{-3} \cdots + K_{-m}\times 2^{-m+2} = K_{-2} + N_2$$

$$(N_2 = K_{-3} + K_{-4}\times 2^{-3} \cdots + K_{-m}\times 2^{-m+2})$$

上式右边的整数部分为 K_{-2} 。

重复上述乘法计算, 即可依次求得 K_{-1} 、 K_{-2} 、 $\cdots$ 、 K_{-m} 。例如, 将 $(0.875)_{10}$ 转换为二进制数, 则

$$\begin{aligned}0.875\times 2 &= 1.750 & \text{整数部分} &= 1 = K_{-1} \\ 0.750\times 2 &= 1.500 & \text{整数部分} &= 1 = K_{-2} \\ 0.500\times 2 &= 1.000 & \text{整数部分} &= 1 = K_{-3}\end{aligned}$$

因此, $(0.875)_{10} = (0.111)_2$ 。

如遇小数部分乘 2 始终不能取整, 则可按精度要求处理。如将 $(0.715)_{10}$ 转换为二进制数, 精确到小数点后第 5 位, 则为 $(0.715)_{10} = (0.10110)_2$ 。

2. 八进制数、十六进制数与二进制数之间的相互转换

八进制数和十六进制数易于转换为二进制数。由于 $8 = 2^3$ 、 $16 = 2^4$ ，所以八进制数中任何一个数码均可用 3 位二进制数来表示；十六进制数中任何一个数码均可用 4 位二进制数来表示。例如

$$(15.5)_8 = (001\ 101.101)_2$$
$$(B2.C)_{16} = (1011\ 0010.1100)_2$$

同样，二进制数也易于转换成八进制数或十六进制数，只要将二进制数的整数部分从低位向高位每 3 位或 4 位分成一组，最高一组不足 3 位或 4 位时在高位用 0 补足；小数部分从高位向低位每 3 位或 4 位分成一组，最后一组不足 3 位或 4 位时在低位补 0，然后把 3 位或 4 位二进制数用相应的八进制数或十六进制数表示。例如，将 $(101\ 111.011)_2$ 转换为八进制数或十六进制数，则

$$(101\ 111.011)_2 = (57.3)_8$$
$$(101\ 111.011)_2 = (0010\ 1111.0110)_2 = (2F.6)_{16}$$

3. 八进制数、十六进制数与十进制数之间的相互转换

将八进制数或十六进制数转换成十进制数，可将各位按权值展开后相加求得。在将十进制数转换为八进制数或十六进制数时，可先转换为二进制数，然后再将得到的二进制数转换为八进制数或十六进制数。例如，

$$(73.875)_{10} = (1001001.111)_2 = (111.7)_8 = (49.E)_{16}$$

1.2.3 几种常用的码制

将一定位数的数码按一定的规则排列起来表示特定对象，称为代码或编码；将形成这种代码所遵循的规则称为码制。在数字系统中，常用一定位数的二进制数码来表示数字、符号和汉字等。因此也可以说，码制即指用二进制数码表示数字或符号的编码方法。下面介绍几种常用的码制。

1. 二-十进制代码（Binary-Coded Decimal 码，BCD 码）

BCD 码是用 4 位二进制数表示 1 位十进制数的编码方法。4 位二进制代码共有 16 个（0000~1111），选取其中 10 个代码与十进制数 0~9 相对应。因此，用 4 位二进制数表示 1 位十进制数时，可以有很多种编码方式。编码方式一般分为有权码和无权码两种。有权码是指二进制数中的每一位都对对应固定的权值，把每一位代表的权值加起来，所得的结果就是所表示的十进制数，常称为自然码。无权码是指二进制数中的每一位无固定的权值，但都有一定的规律，它必须遵循一定的规则。例如，余 3 码比 8421BCD 码多余码（0011），格雷码是相邻两个数之间仅有一位不同。如表 1-2 所示为几种常用的 BCD 码与十进制数的对应关系。

表 1-2 几种常用的 BCD 码与十进制数的对应关系

十进制数	有权码				无权码	
	8421BCD 码	2421 (A) 码	2421 (B) 码	5421 码	余 3 码	格雷码
0	0000	0000	0000	0000	0011	0000
1	0001	0001	0001	0001	0100	0001
2	0010	0010	0010	0010	0101	0011
3	0011	0011	0011	0011	0110	0010
4	0100	0100	0100	0100	0111	0110

续表

十进制数	有权码				无权码	
	8421BCD 码	2421 (A) 码	2421 (B) 码	5421 码	余 3 码	格雷码
5	0101	0101	1011	1000	1000	0111
6	0110	0110	1100	1001	1001	0101
7	0111	0111	1101	1010	1010	0100
8	1000	1110	1110	1011	1011	1100
9	1001	1111	1111	1100	1100	1000
权	8421	2421	2421	5421	无权	无权

8421 码是最基本、最常用的 BCD 码，属于有权码。8421 码选用 0000~1001 这 10 种组合来代表十进制数的 0~9，各位二进制数的权分别为 2^3 、 2^2 、 2^1 、 2^0 （即 8、4、2、1），故称为 8421 码。由于它保存了二进制位权的特点，因此将二进制数各自乘以其权值后相加，即得到所代表的十进制数。因此它与十进制数之间的转换是一种直接按位转换，即一组 4 位二进制数代表 1 位十进制数。

2421 码也是一种有权码，从高位到低位每位的权分别为 2、4、2、1。分为 2421 (A) 码和 2421 (B) 码。2421 (A) 码和 2421 (B) 码的编码方式不完全相同，2421 (B) 码具有互补性，0 与 9、1 与 8、2 与 7、3 与 6、4 与 5 之间的关系是自身按位取反，即这五对代码互为反码。

5421 码也是一种有权码，从高位到低位，每位的权分别为 5、4、2、1。

余 3 码是一种无权码，十进制数用余 3 码表示要比 8421 码在二进制数值上多 3，称为余 3 码。如果将两个余 3 码相加，所得的和将比十进制数和所对应的二进制数多 6。因此在用余 3 码做十进制加法运算时，若两数之和为 10，恰好等于二进制数的 16，于是便从高位自动产生进位信号。

2. 格雷码

格雷码也称循环码，是一种无权码。其特点是任意两组相邻代码之间只有一位不同。典型的格雷码如表 1-3 所示。表中 4 位自然二进制代码的相邻两组代码之间可能有 2 位、3 位，甚至 4 位不同。例如，0111 和 1000 代码中的 4 位都不同，也就是当代码由 0111 变到 1000 时，4 位代码都将发生变化。在实际数字系统中，这 4 位代码不可能同时发生变化，总会有先后之分，从而可能导致系统产生错误响应。而这两组代码对应的格雷码是 0100 和 1100，两者仅有 1 位发生变化。因此，采用格雷码会明显减小数字系统出错的概率。

表 1-3 自然二进制码与格雷码的对比

编 码 顺 序	自然二进制码	格 雷 码
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

1.3 逻辑代数

逻辑代数是 19 世纪中期英国数学家乔治·布尔（George Boole）创立的一门研究客观事物逻辑关系的代数学，也称为布尔代数。逻辑变量及其函数只有 1 和 0 两种取值。这里 1 和 0 往往不表示数值的大小，而表示完全对立的两个方面。1 表示条件具备或事情发生；0 表示条件不具备或事情不发生，反之亦然。

随着数字技术的发展，布尔代数成为研究数字逻辑电路必不可少的工具。在电气控制线路中常用的两态元件，只有“通”和“断”两种状态，是一种二值逻辑电路。将逻辑代数即布尔代数应用于二值逻辑电路中可以简化电路。

1.3.1 逻辑代数中 3 种基本运算

逻辑代数中只有 3 种基本运算：“与”运算（AND）、“或”运算（OR）、“非”运算（NOT）。

1. 逻辑“与”运算

图 1-3 所示的电路中，当两个开关均闭合时，指示灯才会亮。如果把开关闭合作为条件（导致事物结果的原因），把灯亮作为结果，那么图 1-3 表明当决定某一事件的全部条件都满足时，此事件才会而且一定会发生，这种关系被称为“与”逻辑关系，或称为逻辑相乘。

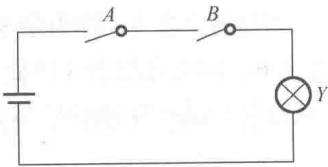


图 1-3 “与”逻辑实例（触点串联）

若以 A 、 B 表示开关的状态，并以 1 表示开关闭合，0 表示开关断开；以 Y 表示指示灯的状态，并以 1 表示灯亮，则 Y 与 A 、 B 的逻辑关系可以用表 1-4 表示，左侧为逻辑电路功能表，右侧为逻辑电路真值表。将输入、输出变量所有相互对应的逻辑值（状态）列在一个表格内，这种表称为真值表。在真值表中，输入变量按照二进制数序列顺序由上而下排列，输出变量是实际逻辑事件含义（因果关系）的逻辑值。真值表能够清楚地表示事物之间的因果关系。

表 1-4 “与”逻辑功能与真值表

A 开关		B 开关	灯 Y	A	B	Y
断开	断开		灭	0	0	0
断开	闭合		灭	0	1	0
闭合	断开		灭	1	0	0
闭合	闭合		亮	1	1	1

实现“与”逻辑的电路称为与门，其逻辑符号如图 1-4 所示。图 1-4(a)给出了被 IEEE（电气与电子工程师协会）和 IEC（国际电工协会）认定的与门图形符号，此符号是目前国外教材和 EDA 软件中普遍采用的国际符号，本书采用这种符号。图 1-4(b)为国标符号。

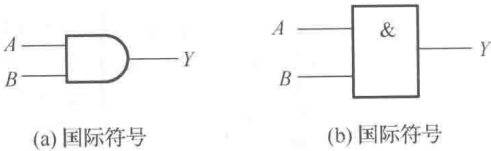


图 1-4 与门逻辑符号

在函数式中，用“ $\cdot$ ”表示与运算， A 和 B 进行与逻辑运算时可以写成 $Y=A \cdot B$ 。“ $\cdot$ ”