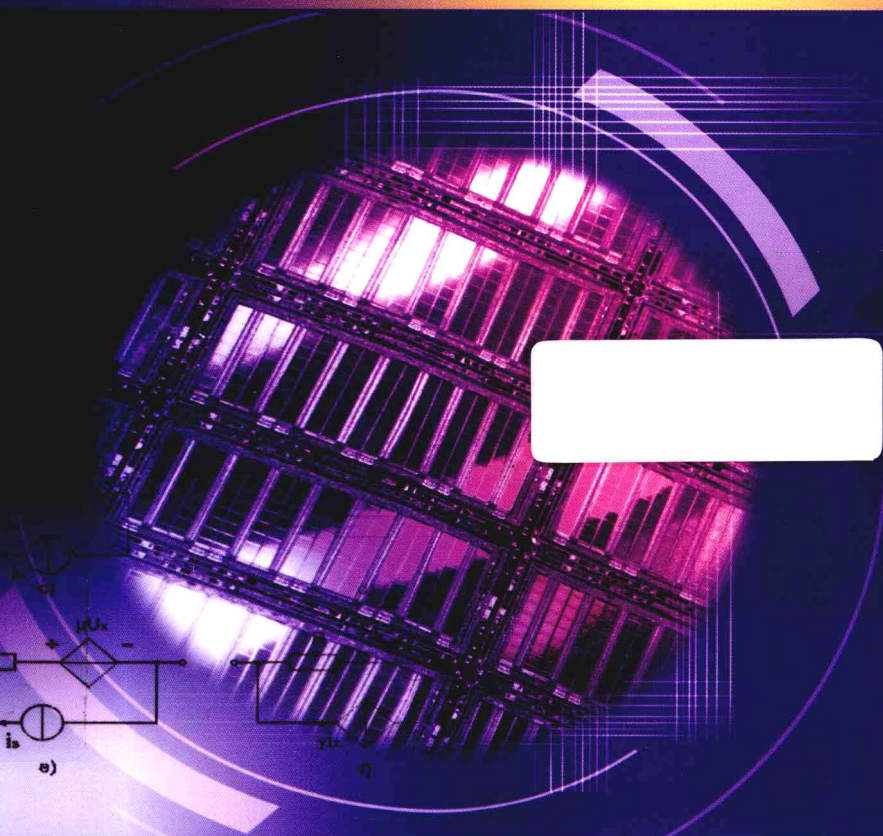




普通高等教育“十二五”规划教材

数字电子技术基础

宋婀娜 主编



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”规划教材

数字电子技术基础

主 编 宋婀娜
副主编 王国新 刘 睿
参 编 訾 鸿 艾延宝 李 娜
主 审 赵金宪



机械工业出版社

全书共分为 8 章, 内容包括数字逻辑基础、逻辑代数、组合逻辑电路、时序逻辑电路、脉冲产生与变换电路、大规模集成电路、数字信号与模拟信号的转换、数字系统设计基础。每章后附有小结和习题。在附录中, 介绍了常用集成门电路、VHDL 硬件设计语言、Quartus II 软件等实用内容。

本书内容精炼, 强调知识的基础性、结构的系统性, 注重实用性。

本书可以作为高等院校自动化、电子信息工程、电气工程、通信工程、测控技术与仪器、计算机等专业的教材, 也可供从事电子技术工作的工程技术人员参考。

图书在版编目 (CIP) 数据

数字电子技术基础/宋婀娜主编. —北京: 机械工业出版社, 2012. 7

普通高等教育“十二五”规划教材

ISBN 978-7-111-37998-0

I. ①数… II. ①宋… 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 (2012) 第 163179 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: 徐 凡 责任编辑: 徐 凡

版式设计: 霍永明 责任校对: 张 媛

封面设计: 张 静 责任印制: 乔 宇

北京机工印刷厂印刷 (三河市南杨庄国丰装订厂装订)

2012 年 8 月第 1 版第 1 次印刷

184mm × 260mm · 16.25 印张 · 396 千字

标准书号: ISBN 978-7-111-37998-0

定价: 30.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066 教材网: <http://www.cmpedu.com>

销售一部: (010) 68326294 机工官网: <http://www.cmpbook.com>

销售二部: (010) 88379649 机工官博: <http://weibo.com/cmp1952>

读者购书热线: (010) 88379203 封面无防伪标均为盗版

前 言

本书是依据教育部教学指导委员会颁布的课程教学基本要求编写的。全书分为8章。第1章介绍数字逻辑基础,第2章介绍分析数字电路的数学工具——逻辑代数,第3章介绍组合逻辑电路及常用的组合逻辑器件,第4章介绍触发器和时序逻辑电路,第5章介绍脉冲产生与变换电路,第6章介绍半导体存储器等大规模集成电路,第7章介绍A/D与D/A转换器,第8章介绍数字系统设计基础。除此之外,在附录中介绍了集成门电路的内部结构,常用数字集成电路的逻辑符号、命名方法,以及VHDL硬件描述语言和Quartus II开发软件。

本书的编写在注重基础知识讲述的同时,还融入了许多相关教材中最新的思想、理论和技术,既有实用性又有先进性,可满足应用型本科学生的培养需求。具有以下特点:

1. 内容精炼,注重实用。删减了对集成门电路内部电路的分析,侧重数字集成电路的逻辑功能和应用,重点介绍数字电路的分析和设计方法,注重读者对实用性的要求。

2. 理论与实际紧密结合。在数字电路的介绍中,采用当前的主流芯片,引入工程实例,解决实际问题,提高学生学习兴趣。

3. 基础与系统并重。强调对基本知识点的覆盖,降低知识点的难度与深度,有利于学生的学习和掌握;同时又强调“数字电子技术”知识的系统性,在书中除对组合逻辑电路和时序逻辑电路的分析和设计等内容进行讲解外,还介绍了数字系统设计的先进方法和手段。

4. 对于数字电子技术中常用的专用名词和专业术语给出了对应的英文解释,为学生学习专业英语打下了一定的基础,同时有助于学生对全英文教材阅读和理解。

宋婀娜任本书的主编,负责全书的整体规划与统稿工作。王国新、刘睿为副主编。参加本书编写工作的还有訾鸿、艾延宝、李娜。其中,第1、4章由宋婀娜、李娜编写,第2、3章由王国新、艾延宝编写,第5、8章由訾鸿编写,第6、7章由刘睿编写,附录部分由王国新、刘睿编写。赵金宪教授任本书的主审,并对本书的编写提出了许多宝贵的意见,在此表示衷心的感谢。

由于水平有限,书中难免会有疏漏和不足之处,如果您在阅读本书时发现不足之处或对内容有修改的意见和建议,请与我们联系。在此,诚恳欢迎广大读者批评指正。

编 者

目 录

前言

第1章 数字逻辑基础 1

1.1 数字信号与数字电路 1

1.1.1 数字信号 1

1.1.2 脉冲信号 1

1.1.3 数字电路 2

1.1.4 数字电路的发展与分类 3

1.2 数制与码制 3

1.2.1 数制 3

1.2.2 码制 7

1.2.3 二进制数的原码、反码和补码及其运算 9

本章小结 9

习题 9

第2章 逻辑代数 11

2.1 逻辑代数概述 11

2.2 逻辑运算 11

2.2.1 基本逻辑运算 11

2.2.2 复合逻辑运算 13

2.3 逻辑函数的表示方法 15

2.4 逻辑代数的运算公式和规则 16

2.4.1 逻辑代数的基本定律和恒等式 16

2.4.2 逻辑代数的基本规则 17

2.5 逻辑函数的化简方法 19

2.5.1 化简的意义 19

2.5.2 代数化简法 19

2.5.3 卡诺图化简法 21

本章小结 29

习题 30

第3章 组合逻辑电路 33

3.1 组合逻辑电路概述 33

3.1.1 组合逻辑电路的特点 33

3.1.2 组合逻辑电路的逻辑功能描述 33

3.1.3 组合逻辑电路的类型和研究方法 33

3.2 组合逻辑电路的分析方法 34

3.3 组合逻辑电路的设计方法 36

3.4 若干典型的组合逻辑集成电路 40

3.4.1 编码器 40

3.4.2 译码器 44

3.4.3 数据选择器 52

3.4.4 数值比较器 56

3.4.5 算术运算电路 59

3.5 组合逻辑电路中的竞争与冒险 64

3.5.1 产生竞争与冒险的原因 65

3.5.2 消去竞争与冒险的方法 66

本章小结 67

习题 67

第4章 时序逻辑电路 71

4.1 触发器 71

4.1.1 RS 触发器 71

4.1.2 JK 触发器 76

4.1.3 D 触发器 79

4.1.4 T 触发器 81

4.1.5 触发器之间的转换 81

4.1.6 锁存器 84

4.2 时序逻辑电路概述 85

4.2.1 时序逻辑电路的特点和结构 85

4.2.2 时序逻辑电路的分类 85

4.3 时序逻辑电路的分析 86

4.3.1 时序逻辑电路的分析方法与步骤 86

4.3.2 时序逻辑电路分析举例 87

4.4 时序逻辑电路的设计 93

4.4.1 同步时序逻辑电路的设计 93

4.4.2 同步时序逻辑电路设计举例 94

4.4.3 异步时序逻辑电路的设计 98

4.5 常用时序集成电路 98

4.5.1 时序集成电路的逻辑符号 98

4.5.2 寄存器和移位寄存器 100

4.5.3 计数器 104

4.6 常用 MSI 计数模块的应用 110

4.6.1 获得任意进制计数器 110

4.6.2 计数器模块的其他应用	115	6.1.2 只读存储器	162
4.6.3 移位寄存器型计数器	116	6.2 可编程逻辑器件	166
本章小结	120	6.2.1 简单可编程逻辑器件	167
习题	121	6.2.2 高密度可编程逻辑器件	172
第5章 脉冲产生与变换电路	129	本章小结	176
5.1 集成 555 定时器	129	习题	177
5.1.1 555 定时器的电路结构	129	第7章 数字信号与模拟信号的转换 ...	178
5.1.2 555 定时器的电路功能	130	7.1 D/A 转换器	178
5.2 施密特触发器	131	7.1.1 D/A 转换器的基本原理	178
5.2.1 用门电路构成施密特触发器	131	7.1.2 集成 D/A 转换器及主要技术 参数	179
5.2.2 用 555 定时器构成施密特触 发器	133	7.2 A/D 转换器	184
5.2.3 集成施密特触发器	134	7.2.1 A/D 转换器的基本原理	184
5.2.4 施密特触发器的应用	135	7.2.2 集成 A/D 转换器及主要技术 参数	187
5.3 单稳态触发器	136	本章小结	191
5.3.1 用门电路构成单稳态触发器	137	习题	191
5.3.2 用 555 定时器构成单稳态触 发器	139	第8章 数字系统设计基础	194
5.3.3 用施密特触发器构成单稳态 触发器	140	8.1 数字系统基本概念	194
5.3.4 集成单稳态触发器	141	8.2 数字系统设计的一般过程	194
5.3.5 单稳态触发器的应用	145	8.3 数字系统设计的描述方法	196
5.4 多谐振荡器	146	8.4 用可编程逻辑器件进行数字系统 设计	200
5.4.1 用门电路组成的多谐振荡器	146	本章小结	210
5.4.2 用 555 定时器构成多谐振荡器 ...	148	习题	210
5.4.3 用施密特触发器构成多谐振 荡器	150	附录	212
5.4.4 石英晶体多谐振荡器	151	附录 A 集成逻辑门电路的内部结构 简介	212
5.4.5 多谐振荡器的应用	152	附录 B 常用数字集成电路的逻辑符 号、命名方法及索引	223
本章小结	153	附录 C VHDL 硬件描述语言基础	228
习题	154	附录 D Quartus II 开发软件简介	239
第6章 大规模集成电路	158	参考文献	251
6.1 半导体存储器	158		
6.1.1 随机存取存储器	158		

第 1 章 数字逻辑基础

1.1 数字信号与数字电路

1.1.1 数字信号

数字信号(Digital Signal)是指离散的、不连续的信号形式,如电子秒表的秒信号、生产流水线上的计数信号。数字信号以一定的最小量值为量化单位。

可用一系列断续变化的电压脉冲表示电路中的数字信号,如用恒定的正电压(5V)表示二进制数 1,用恒定的负电压(或 0V)表示二进制数 0,这两个电压值可称为逻辑电平。当用 1 表示高电平、0 表示低电平时,称为正逻辑;当用 0 表示高电平、1 表示低电平时,称为负逻辑。在数字电路与系统的分析与设计中,一般采用的是正逻辑体制。

用 0 和 1 表示两种相反对立的状态,如晶体管的导通和截止、机械开关的开启或闭合、磁性材料的两种不同剩磁状态等,通常称为二值数字逻辑。在数字系统中,这两种状态可用来表示数、字母、符号以及其他类型的信息。

数字信号具有传输可靠、易于存储、抗干扰能力强、稳定性好等优点。

1.1.2 脉冲信号

脉冲信号(Pulse Signal),通常指在短暂的时间间隔内作用于电路的电压或电流信号。将脉冲信号赋以特定的数字含义后,就称为数字信号。常用的脉冲信号波形有矩形波、锥形波、锯齿波等,如图 1-1 所示。脉冲信号一般是周期性变化的,最常见的是矩形波脉冲信号。

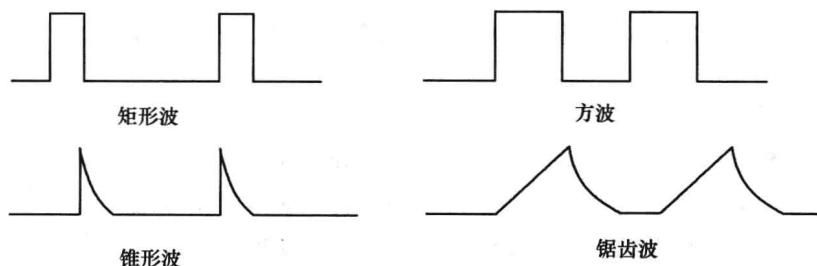


图 1-1 各种脉冲信号波形

1. 描述脉冲的几个名词

1) 上升沿与下降沿: 脉冲波形由低电位跳变到高电位称为脉冲的上升沿(正边沿); 脉冲波形由高电位跳变到低电位称为脉冲的下降沿(负边沿)。

2) 脉冲的前沿与后沿: 脉冲出现的过程称为脉冲的前沿; 脉冲消失的过程称为脉冲的后沿。

3) 正跳变与负跳变: 脉冲波形由低电位跳变到高电位的过程称为脉冲的正跳变; 脉冲波形由高电位跳变到低电位的过程称为脉冲的负跳变。

4) 正脉冲与负脉冲: 如果脉冲出现时的电位比出现前的电位高, 这样的脉冲称为正脉冲。如果脉冲出现时的电位比出现前的电位低, 这样的脉冲称为负脉冲。

2. 矩形脉冲的主要参数

在图 1-2 所示的波形中, 脉冲的上升沿与下降沿都是陡直的, 这样的脉冲称为理想的矩形脉冲。

理想的矩形脉冲可以用以下参数来描述:

- 1) 幅度: 脉冲的底部到顶部之间的变化量称为脉冲的幅度, 用 V_m 表示。
- 2) 宽度: 从脉冲出现到脉冲消失所用的时间称为脉冲的宽度, 用 t_w 表示。
- 3) 周期: 两个相邻脉冲对应点之间的时间间隔称为脉冲的周期, 用 T 表示。
- 4) 占空比: 脉冲宽度占整个周期的百分比, 用 q 表示, $q = \frac{t_w}{T} \times 100\%$ 。

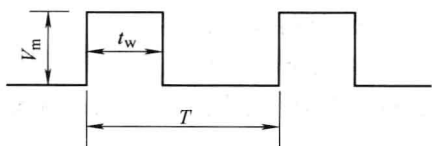


图 1-2 理想的矩形脉冲信号波形及主要参数

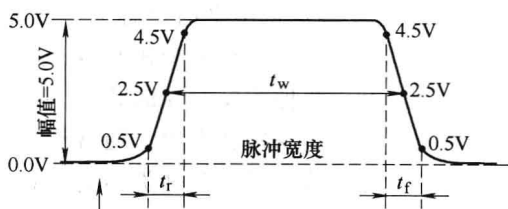


图 1-3 实际矩形脉冲波形及主要参数

实际的矩形脉冲往往与理想的矩形脉冲不同, 即脉冲的前沿与脉冲的后沿都不是陡直的, 如图 1-3 所示。描述实际的矩形脉冲除以上理想脉冲波形的参数外还有以下参数:

- 1) 宽度 t_w : 从脉冲前沿的 $0.5V_m$ 到脉冲后沿的 $0.5V_m$ 两点之间的时间间隔称为脉冲的宽度, 又称为脉冲的持续时间。
- 2) 上升时间 t_r : 指脉冲的上升沿从 $0.1V_m$ 上升到 $0.9V_m$ 所用的时间。
- 3) 下降时间 t_f : 指脉冲的下降沿从 $0.9V_m$ 下降到 $0.1V_m$ 所用的时间。

1.1.3 数字电路

传递和处理数字信号的电路称为数字电路(Digital Circuit), 因其处理的信号为逻辑电平信号, 所以也称为数字逻辑电路。数字电路中通常含有对数字信号进行传送、逻辑运算、控制、计数、寄存、显示以及信号的产生、整形、变换等功能的部件。数字电路的分析重点是电路的输入与输出之间的逻辑关系。

数字电路是组成数字逻辑系统的硬件基础, 其具有严格的时序性和逻辑性, 与模拟电路相比, 数字电路有许多显著的优点:

- 1) 数字电路具有高速、高稳定性与高可靠性。
- 2) 可方便地长期存储和无限复制数字信息。
- 3) 易于设计, 具有可编程性。同时, 数字电路设计技术具有可扩展性。
- 4) 数字集成电路易于标准化大规模生产, 经济性好。

1.1.4 数字电路的发展与分类

数字电路的发展经历了电子管、晶体管、集成电路等发展阶段。1961年集成电路的出现,大大促进了电子技术的发展,尤其是促进了数字电路和微型计算机的飞速发展。

数字集成电路按集成度可分为小规模、中规模、大规模、超大规模和特大规模,见表1-1。集成度是指单个芯片上含有的晶体管数量或等效门的数量。

表 1-1 不同规模的数字集成电路

分 类	集成度(等效门的数量)	典型数字集成电路
小规模(SSI)	≤ 12	门电路、触发器
中规模(MSI)	12 ~ 99	译码器、编码器、计数器
大规模(LSI)	100 ~ 9999	小容量存储器、门阵列
超大规模(VLSI)	9999 ~ 99999	单片微处理器
特大规模(ULSI)	≥ 100000	高密度可编程逻辑器件

图1-4为常见的中小规模数字集成电路实物图,图1-5为常见的大规模、超大规模数字集成电路实物图。

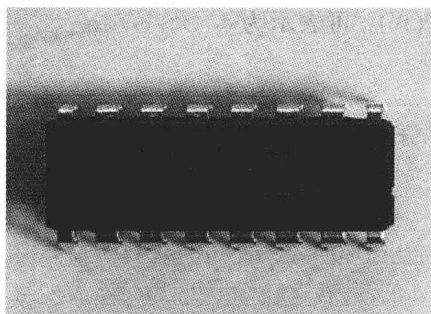


图 1-4 常见的中小规模数字集成电路实物图



图 1-5 超大规模数字集成电路实物图

按照逻辑特性的不同,数字电路可分为两大类:组合逻辑电路和时序逻辑电路。

数字电路广泛应用于计算机、通信、自动控制、消费类电子产品等领域。现代计算机就是最典型的数字电路系统。除此之外,如单片机、DSP、数码产品、移动电话等,无不采用数字电路实现。随着数字电路技术的发展,会有更多的数字电子产品问世,使人们的生活更加丰富多彩。

1.2 数制与码制

1.2.1 数制

数制(Numeral System)也称计数制,是多位数码的中每一位的构成方法以及从高位到低位的进位规则。人们通常采用的数制有十进制、二进制、八进制和十六进制。

学习数制,必须首先掌握数码、基数和位权这三个概念。

数码(Digits): 数制中表示基本数值大小的不同数字符号。例如, 十进制有十个数码: 0、1、2、3、4、5、6、7、8、9; 二进制的数码: 0、1。

基数(Base 或 Radix): 数制所使用数码的个数。例如, 二进制的基数为 2; 十进制的基数为 10。

位权(Weight): 数码在不同位置的倍率值。例如, 十进制数 123, 其中, 数码 1 的位权是 100, 2 的位权是 10, 3 的位权是 1。

数制的特点如下:

1) 采用基数, r 进制的基数就是 r 。

2) 基数确定数码的个数。如十进制的数码个数为 10, 即 0、1、2、3、4、5、6、7、8、9; 二进制的数码个数为 2, 即 0 和 1。

3) 进位规则, 如十进制为逢 10 进 1, 二进制为逢 2 进 1, 即“逢 r 进 1”。

以任何数字 0, 1, 2, $\dots$, $r-1$ 为基数的 r 进制的数 R 可写成下式:

$$R = a_{n-1}r^{n-1} + a_{n-2}r^{n-2} + \dots + a_1r^1 + a_0r^0 + a_{-1}r^{-1} + \dots + a_{-m}r^{-m} = \sum_{i=-m}^{n-1} a_i r^i \quad (1-1)$$

式中, n 为整数部分的位数; m 为小数部分的位数; a_i 为系数; r^i 为数码具有的权重。

1. 十进制 (Decimal Number System)

一个形式如 $d_{n-1}d_{n-2}\dots d_1d_0d_{-1}\dots d_{-m}$ 的十进制数 $(N)_D$ 可表示为

$$\begin{aligned} (N)_D &= (d_{n-1}d_{n-2}\dots d_1d_0d_{-1}\dots d_{-m})_D \\ &= d_{n-1} \times 10^{n-1} + d_{n-2} \times 10^{n-2} + \dots + d_1 \times 10^1 + d_0 \times 10^0 + d_{-1} \times 10^{-1} + \dots + d_{-m} \times 10^{-m} \\ &= \sum_{i=-m}^{n-1} d_i \times 10^i \end{aligned} \quad (1-2)$$

式中, n 为整数部分的位数; m 为小数部分的位数, d_i 为系数, 可以在 0~9 中取值。

例如, $(123.45)_D = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$ 。

由以上讨论可以归纳出:

1) 十进制基数为 10, 即有 0~9 十个数码。

2) 进位规则为“逢 10 进 1”。

3) 不同位置数码具有不同的位权, 即 10^i 。

十进制数表示方法有多种, 如 $(123.45)_D$ 、 $(123.45)_{10}$ 、123.45D 等。

要使一个电子元器件或电路具有严格区分的十个状态来与十进制的十个不同的数码相对应是非常困难的, 因此人们日常习惯使用的十进制难于用电路实现, 因此数字电路中一般不直接使用十进制。

2. 二进制 (Binary Number System)

一个形式如 $b_{n-1}b_{n-2}\dots b_1b_0b_{-1}\dots b_{-m}$ 的二进制数 $(N)_B$ 可表示为

$$\begin{aligned} (N)_B &= (b_{n-1}b_{n-2}\dots b_1b_0b_{-1}\dots b_{-m})_B \\ &= b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \dots + b_1 \times 2^1 + b_0 \times 2^0 + b_{-1} \times 2^{-1} + \dots + b_{-m} \times 2^{-m} \end{aligned}$$

$$= \sum_{i=-m}^{n-1} b_i \times 2^i \quad (1-3)$$

式中, n 为整数部分的位数; m 为小数部分的位数, b_i 为系数, 可以在 0、1 中取值。

例如, $(1011.01)_B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$ 。

与十进制类似, 二进制有如下规律:

- 1) 基数为 2, 即 0、1 两个数码。
- 2) 进位规则为“逢 2 进 1”。
- 3) 不同位置数码具有不同的位权, 即 2^i 。

二进制数可用 $1011.01B$, $(1011.01)_B$, $(1011.01)_2$ 等形式表示。可用具有两种稳定状态电子元器件来模拟 0 和 1, 表示二进制数。这种方法简单方便, 信号的处理、存储和传输都十分可靠, 因此数字信号是用二进制表示的。

除利用二进制数进行算术运算外, 还可进行逻辑运算。

3. 八进制和十六进制

数字电路中直接处理的是二进制数, 当位数增多时, 用二进制表示的数比较难读难记, 于是引入了八进制和十六进制两种数制。

八进制或十六进制用 8 或 16 作为基数, 可用 $(N)_O$ 表示八进制, 用 $(N)_H$ 表示十六进制。八进制的数码是由 0、1、2、…、7 表示; 十六进制的数码是由 0、1、2、…、9、A、B、C、D、E、F 表示。

4. 常用数制的转换

(1) 其他进制转换为十进制

将其他进制按权位展开, 然后各项相加, 就得到相应的十进制数。

例 1-1 将其他进制的数转换为等值的十进制数。

$$(1011.011)_B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = (11.375)_D$$

$$(273.65)_O = 2 \times 8^2 + 7 \times 8^1 + 3 \times 8^0 + 6 \times 8^{-1} + 5 \times 8^{-2} = (187.828125)_D$$

$$(2FA.6C)_H = 2 \times 16^2 + 15 \times 16^1 + 10 \times 16^0 + 6 \times 16^{-1} + 12 \times 16^{-2} = (762.421785)_D$$

(2) 将十进制转换成其他进制

应将十进制数的整数部分和小数部分分开转换。

整数部分的转换步骤如下(基数除法):

- 1) 把要转换的数除以新的进制的基数, 把余数作为新进制的最低位(LSB)。
- 2) 把上一次得到的商再除以新的进制基数, 把余数作为新进制的次低位。
- 3) 继续上一步, 直到最后的商为零, 这时的余数就是新进制的最高位(MSB)。

小数部分的转换步骤如下(基数乘法):

1) 把要转换数的小数部分乘以新进制的基数, 把得到的整数部分作为新进制小数部分的最高位(MSB)。

2) 把上一步得到的小数部分再乘以新进制的基数, 把整数部分作为新进制小数部分的次高位。

3) 继续上一步, 直到小数部分变成零或者达到预定的要求。

例 1-2 将 $(13.123)_D$ 转换为等值的二进制数。

整数部分为 13, 转换过程如下:

对应的二进制数整数为 $(1101)_B$ ，有

$$\begin{array}{r}
 \text{余数} \\
 \begin{array}{r}
 13 \\
 2 \overline{) 6} \\
 2 \overline{) 3} \\
 2 \overline{) 1} \\
 0
 \end{array}
 \begin{array}{l}
 1 \text{ MSB} \\
 0 \\
 1 \\
 1 \text{ LSB}
 \end{array}
 \end{array}$$

小数部分为 0.125，转换过程如下：

对应的二进制数小数为 $(0.001)_B$ ，有

$$\begin{array}{r}
 \text{积的整数} \\
 \begin{array}{r}
 0.125 \times 2 = 0.25 \\
 0.25 \times 2 = 0.5 \\
 0.5 \times 2 = 1.00
 \end{array}
 \begin{array}{l}
 0 \text{ LSB} \\
 0 \\
 1 \text{ MSB}
 \end{array}
 \end{array}$$

$$(13.125)_D = (1101.001)_B$$

例 1-3 将 $(1735.1875)_D$ 转换为等值的八进制数。

整数部分为 1735，转换过程如下：

对应的八进制数整数为 $(3307)_O$ ，有

$$\begin{array}{r}
 \text{余数} \\
 \begin{array}{r}
 1735 \\
 8 \overline{) 216} \\
 8 \overline{) 27} \\
 8 \overline{) 3} \\
 0
 \end{array}
 \begin{array}{l}
 7 \\
 0 \\
 3 \\
 3
 \end{array}
 \end{array}$$

整数部分为 0.1875，转换过程如下：

对应的八进制数整数为 $(0.14)_O$ ，有

$$\begin{array}{r}
 \text{积的整数} \\
 \begin{array}{r}
 0.1875 \times 8 = 1.50 \\
 0.5 \times 8 = 4.00
 \end{array}
 \begin{array}{l}
 0 \\
 4
 \end{array}
 \end{array}$$

$$(1735.1875)_D = (3307.04)_O$$

(3) 二进制数与八进制数、十六进制数的相互转换

二进制数转换为八进制数、十六进制数：它们之间满足 $2^3 = 8$ 和 $2^4 = 16$ 的关系，因此把要转换的二进制数从低位到高位每 3 位或 4 位一组，整数部分高位不足时在前面添“0”，小数部分不足时在后面添“0”，然后把每组二进制数转换成八进制数或十六进制数即可。

八进制数、十六进制数转换为二进制数：把上面的过程反过来即可。

例 1-4 将下列二进制数转换为等值的八进制数。

$$(1101101011.0111101)_B = (001,101,101,011.011,110,100)_B = (1553.364)_O$$

$$(1101101011.0111101)_B = (0011,0110,1011,0111.1010)_B = (36B.7A)_H$$

例 1-5 将下列数转换为等值的二进制数。

$$(274.356)_O = (100111100.011101110)_B$$

$$(5A.3C6)_H = (1011010.00111100011)_B$$

1.2.2 码制

数码除了可以单纯地表示数以外，还可表示不同事物和状态。用数码表示符号、文字、逻辑关系等信息的过程叫做编码，在编码过程中遵循的规则称为码制 (Code System)，这些组合的数码称为代码。在日常生活中人们用得最多的是十进制代码，如邮政编码、电话号码等，此时代码已经没有数值大小的概念。代码可分为数字型和字符型，数字型用来表示数字的大小，字符型用来表示不同的符号或事物。

在数字电路中是用 0 和 1 二进制数的组合来表示不同的符号、状态的，即二进制代码。 n 位二进制代码可表示 2^n 个信号，如果需要编码的信号有 N 项，则所需的二进制代码位数 n 应满足

$$2^{n-1} < N \leq 2^n$$

下面介绍几种常用的代码。

1. 二—十进制码

用 4 位二进制码表示 1 位十进制数，称为二—十进制码 (Binary Coded Decimal)，或 BCD 码。根据编码规则的不同，BCD 码还可分为有权码和无权码，常见的几种 BCD 码见表 1-2。

表 1-2 常见的几种 BCD 码

十进制 \ 码制	8421BCD 码	2421BCD 码	5421BCD 码	4221BCD 码	余三码
0	0000	0000	0000	0000	0011
1	0001	0001	0001	0001	0100
2	0010	0010	0010	0010	0101
3	0011	0011	0011	0011	0110
4	0100	0100	0100	0110	0111
5	0101	1011	1000	0111	1000
6	0110	1100	1001	1010	1001
7	0111	1101	1010	1011	1010
8	1000	1110	1011	1110	1011
9	1001	1111	1100	1111	1100
位权	8421	2421	5421	4221	无权码

(1) 有权码

每个代码中的 1 都代表一个固定的十进制数值，称为这一位的权值，把每位的代表的十进制数值加起来，得到的结果就是它所代表的十进制数。例如在 8421BCD 码中，从左到右的每一位的权值分别为 8、4、2、1，因此称为 8421BCD 码，其他 2421 码、4221 码、5421 码等都是如此得到的。它们都是有码码 (Weighted Code)。

(2) 无权码

在无权码的编码方式中，每个代码中的 1 不代表固定的数值，但也有一定的编码规则。例如余三码是由 8421BCD 码加 3 得到的。余三码 (Excess-3 Code) 中的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 的码组之间互为反码，可简化 BCD 码的减法运算。

2. 格雷码

多位二进制数在形成和传输的过程中, 由于各位的变化速度不同而可能产生错误, 为此出现了多种可靠性编码, 格雷码(Gray Code)就是其中的一种。4 位的格雷码见表 1-3。格雷码有如下特点:

- 1) 任意相邻代码中的数码只有一位不同, 其余各位均相同, 属于无权码。
- 2) 不同位数的格雷码首尾循环, 因此也称为循环码。
- 3) 任何一个十进制数都有与其对应的格雷码, 而不是由低位格雷码拼凑而来的。例如, $(17)_D$ 的格雷码是 $(11001)_G$ 而不是 (00010100) , 这样在数码变化时可大大减少错码的可能性。

自然二进制码与格雷码之间的转换规则如下:

已知自然二进制码为

$$B_{n-1}B_{n-2}\cdots B_2B_1B_0$$

对应格雷码为

$$G_{n-1}G_{n-2}\cdots G_2G_1G_0$$

则保留最高位 $G_{n-1} = B_{n-1}$, 其余各位为

$$G_i = B_{i+1} \oplus B_i, \quad i = 0, 1, 2, \cdots, n-2。$$

反之, 将格雷码转换为自然二进制码的转换规则与之类似:

保留最高位 $B_{n-1} = G_{n-1}$, 其余各位为

$$B_{i-1} = G_{i-1} \oplus B_i, \quad i = 0, 1, 2, \cdots, n-1$$

其中, “ $\oplus$ ”为异或逻辑运算, 参与运算的两数相同结果为 0, 不同结果为 1。

表 1-3 4 位格雷码

十进制数	二进制数	格雷码	十进制数	二进制数	格雷码
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

3. ASCII 码

ASCII(American Standard Code for Information Interchange)码, 美国信息交换标准码, 是目前国际上广泛采用的一种字符码。ASCII 码用 7 位二进制代码表示 128 个不同的字符和符号, 其中包括 96 个图形字符(大小写英文字母 52 个, 数字符 10 个, 专用图形符号 34 个), 32 个控制字符。

ASCII 码是目前大部分计算机与外部设备交换信息所采用的字符编码。例如, 键盘将按键的字符用 ASCII 码送入计算机, 计算机将处理好的数据用 ASCII 码传送到显示器或打印机等外部设备等。

1.2.3 二进制数的原码、反码和补码及其运算

1. 原码、反码和补码

1) 原码：二进制数可分为有符号数和无符号数，原码、反码和补码都是针对有符号数定义的。有符号数用最高位表示符号，“0”表示正，“1”表示负。

原码就是这个数的二进制本身形式。

2) 反码：正数的反码为原码，负数的反码为除了符号位外各位取反(将0变为1, 1变为0)。

3) 补码：补码是原码按指定规则经过变换后构成的一种二进制码。补码的最高位为符号位，正数为“0”，负数为“1”。

正数的补码与原码相同；负数的补码是将原码(除符号位外)逐位求反，然后在最低位加1得到。

补码也称为二进制数的基数的补码或称为2的补码；反码也称为二进制数的降基数的补码或1的补码。无论是补码还是反码，按定义再求补或求反一次，将还原为原码。

2. 算术运算

1) 反码运算：两数反码之和等于两数之和的反码，即

$$[N_1]_{\text{反}} + [N_2]_{\text{反}} = [N_1 + N_2]_{\text{反}}$$

二进制数的符号位参加运算，当符号位有进位时，需循环进位，即把符号位进位加到和的最低位。

2) 补码运算：补码的运算与反码的运算相似，两数补码之和等于两数之和的补码，即

$$[N_1]_{\text{补}} + [N_2]_{\text{补}} = [N_1 + N_2]_{\text{补}}$$

符号位参加运算，但不需要循环进位，如有进位，自动丢弃。由于补码运算无循环进位，比反码运算简单，因而应用更广泛。但补码的运算应在其相应位数表示的数值范围内进行，否则将可能产生错误的计算结果。

本章小结

本章介绍了数字信号的定义和特征，数字电路的分类与优点，以及在数字电路和数字系统中使用二进制数来表示数据、符号、信息等；讲解了数制与码制，常用的二进制、八进制、十六进制及它们之间的转换，以及常用的BCD码、格雷码、ASCII码等。

习 题

1-1 完成下列转换：

(1) $(1010110.01101)_B = (\quad)_O = (\quad)_H$

(2) $(36.125)_D = (\quad)_B = (\quad)_H$

(3) $(465.43)_O = (\quad)_B = (\quad)_H = (\quad)_D$

(4) $(8F.FF)_H = (\quad)_B = (\quad)_O = (\quad)_D$

(5) $(459)_D = (\quad)_{8421\text{BCD码}} = (\quad)_{\text{余三码}}$

(6) $(36.09)_D = (\quad)_{8421\text{BCD码}} = (\quad)_{\text{余三码}}$

1-2 写出下列二进制数的原码、反码和补码：

- (1) $(+1011)_B$ 的原码为(), 反码为(), 补码为()
 (2) $(+00110)_B$ 的原码为(), 反码为(), 补码为()
 (3) $(-1101)_B$ 的原码为(), 反码为(), 补码为()
 (4) $(-00110)_B$ 的原码为(), 反码为(), 补码为()

1-3 选择题

- (1) 下列各组数中, 是六进制数的是()。
 A. 14752 B. 62593 C. 53452 D. 37418
- (2) 十进制数 62 对应的十六进制数是()。
 A. 3EH B. 36H C. 38H D. 3DH
- (3) 下列数中与十进制数 163D 不相等的是()。
 A. A3H B. 10100011B C. $(000101100011)_{8421BCD\text{码}}$ D. 1001000110
- (4) 已知二进制数 11001010, 其对应的十进制数为()。
 A. 202 B. 192 C. 106 D. 92
- (5) 十进制数 78 所对应的二进制数和十六进制数分别为()。
 A. 1100001B, 61H B. 1001110B, 4EH C. 1100001B, C2H D. 1001110B, 9CH
- (6) 与八进制数 $(166)_8$ 等值的十六进制数和十进制数分别为()。
 A. 76H, 118D B. 76H, 142D C. E6H, 230D D. 74H, 116D
- (7) 与二进制数 $(1100110111.001)_2$ 等值的十六进制数为()。
 A. 337.2H B. 637.1H C. 1467.1H D. C37.4H
- (8) 下列数中最大的是()。
 A. 100101110B B. 12FH C. 301D D. $(10010111)_{8421BCD\text{码}}$
- (9) 用 0、1 两个符号对 100 个信息进行编码, 则至少需要()。
 A. 8 位 B. 7 位 C. 9 位 D. 6 位
- (10) 相邻两组编码只有 1 位不同的编码是()。
 A. 2421BCD 码 B. 8421BCD 码 C. 余三码 D. 格雷码

第2章 逻辑代数

2.1 逻辑代数概述

在数字逻辑电路中,用1位二进制数码的0和1表示一个事物的两种不同逻辑状态。这种只有两种对立逻辑状态的逻辑关系称为二值逻辑。所谓“逻辑”,在这里是指事物间的因果关系。当两个二进制数码表示不同的逻辑状态时,它们之间按照指定的某种因果关系进行的运算称为逻辑运算(Boolean Operation)。逻辑运算实现的是输出与输入之间的逻辑关系,即结果与条件(或原因)的关系。描述这种关系的函数称为逻辑函数。

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。1938年,克劳德·香农(Claude E. Shannon)把布尔代数应用到继电器开关电路的设计,因此将布尔代数又称为开关代数。随着数字技术的发展,布尔代数在二值逻辑电路中得到了广泛应用,成为数字逻辑电路分析和设计的基础,所以,布尔代数也称为逻辑代数。下面将会看到,虽然有些逻辑代数的运算公式在形式上和普通代数的运算公式相同,但是两者所包含的物理意义有本质的不同。逻辑代数中的变量称为逻辑变量。逻辑运算表示的是逻辑变量以及常量之间逻辑状态的推理运算,而不是数量之间的运算。

虽然在二值逻辑中,每个变量的取值只有0和1两种可能,只能表示两种不同的逻辑状态,但是人们可以用变量的不同组合来表示事物的多种逻辑状态,处理复杂的逻辑问题。

2.2 逻辑运算

2.2.1 基本逻辑运算

在逻辑代数中,基本逻辑运算(Basic Logic Operation)有与、或、非三种,下面分别讨论这三种基本运算。

1. 与运算

有一个事件,当决定该事件的所有条件全部具备之后,该事件才会发生,这样的因果关系称为“与”逻辑关系,也称为逻辑乘,或者称为与运算(AND)。例如,图2-1a所示为一个简单的与逻辑电路模型,只有当两个开关同时闭合时,指示灯才亮。只要有一个开关断开或者两个开关均断开时,则指示灯都不亮。因此它们之间满足与逻辑关系。

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 L 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表示的与逻辑关系的图表,这种输入逻辑变量所有取值的组合与其所对应的输出逻辑函数值构成的表格,称为真值表(TruthTable)。与逻辑的真值表见表2-1。