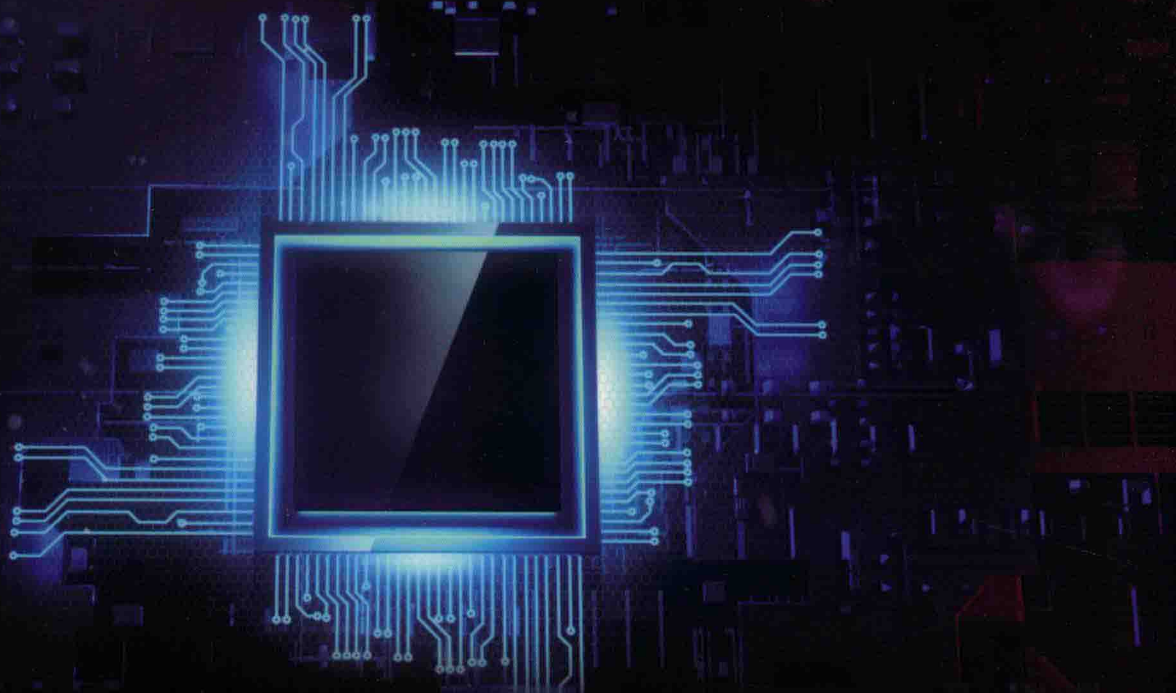

S H U Z I D I A N Z I J I S H U

普通高等教育"十三五"规划教材

# 数字电子技术

主编◎李莎　刘剑滨　杨家林

任课教师可申请 课件PPT 可申请

任课教师可申请 试题 可申请

任课教师可申请 课程标准 教学计划 可申请

华中科技大学出版社
http://www.hustp.com

普通高等教育"十三五"规划教材

# 数字电子技术

主　编　李　莎　刘剑滨　杨家林
副主编　王雪霏　刘　晨　龙艳萍



华中科技大学出版社
http://www.hustp.com
中国·武汉

## 内 容 简 介

本书主要包括：绪论、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、脉冲的产生与整形电路、数/模与模/数转换电路、数字系统设计等内容。

本书既可以作为高等院校自动化、电气工程、电子信息、通信等相关专业的教材，也可以作为高等院校机械类、近机类各专业的教材和相关专业技术人员的参考书。

**图书在版编目(CIP)数据**

数字电子技术/李莎，刘剑滨，杨家林主编. 一武汉：华中科技大学出版社，2019.8
ISBN 978-7-5680-4950-4

Ⅰ.①数… Ⅱ.①李… ②刘… ③杨… 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(2019)第 150786 号

**数字电子技术**
Shuzi Dianzi Jishu

李 莎 刘剑滨 杨家林 主编

策划编辑：张 毅
责任编辑：段亚萍
封面设计：孢 子
责任校对：李 琴
责任监印：朱 玢
出版发行：华中科技大学出版社(中国·武汉) 电话：(027)81321913
武汉市东湖新技术开发区华工科技园 邮编：430223
录 排：华中科技大学惠友文印中心
印 刷：武汉华工鑫宏印务有限公司
开 本：787mm×1092mm 1/16
印 张：16.5
字 数：420 千字
版 次：2019 年 8 月第 1 版第 1 次印刷
定 价：45.00 元

華中出版

# 前言 QIANYAN

数字电子技术的发展十分迅速，今天，数字电子产品几乎覆盖了从工农业生产到日常生活的所有领域。“数字电子技术”是各高等院校电子信息、电气工程、自动控制、机电、计算机及其他相关专业的必修基础课程，该课程既具有很强的理论性、系统性，又有很强的工程性、实践性。

本书以数字电子技术的基本知识为主线，系统地介绍了数字电路的基本组成单元，结合若干典型电路讲解了数字电路的基本概念、基本分析方法和基本设计方法。本书既全面、系统地论述了数字电子技术的基础理论知识，又强调了数字电子技术的实用性，还介绍了一些数字电子技术设计的新方法和新技术。全书基础夯实，重视应用，突出创新，注重理论为实践服务，许多章节以实际任务开篇，引起读者好奇心，使读者在完成任务中学习理论知识，拓宽了知识面，既保证必要的基本知识、基本理论，又注重读者能力培养，注重科学性、先进性和实用性。

本书由李莎、刘剑滨、杨家林担任主编，由王雪霏、刘晨、龙艳萍担任副主编，韩冰、张欢、徐永锦参与校稿。全书由李莎负责整理和统稿。

本书在编写过程中参考了相关文献，在此对所引用文献的作者、译者一并表示衷心的感谢！同时，借此机会向为本书的出版给予了大力支持的编者所在院校领导和华中科技大学出版社表示深深的谢意！

由于编者的水平有限，加之数字电子技术也在不断发展，书中难免存在不完善及错误之处，恳请同行及广大读者批评指正！

编　者

**2019 年 6 月**

目录 MULU

# 绪　论

1. 数字电路简介

1）信号的分类

电子电路中的信号可以分为两大类：数字信号和模拟信号。

在自然界的许多物理量中，有一些如温度、压力、声音、质量等都具有一个共同的特点，即它们在时间上是连续变化的，幅值上也是连续取值的。这种在时间上和数值上都连续变化的物理量称为模拟量，表示模拟量的信号称为模拟信号。

与模拟量相对应的另一类物理量称为数字量。这些信号的变化发生在一系列离散的瞬间，其值也是离散的，如电子表的秒信号、生产流水线上记录零件个数的计数信号、人的心率等。这种在时间和数值上具有离散性的物理量称为数字量，表示数字量的信号称为数字信号。

2）电路的分类

处理模拟信号的电子电路称为模拟电路，处理数字信号的电子电路称为数字电路（也称逻辑电路）。具体来讲，模拟电路是分析、处理或产生模拟信号的电路，分析模拟电路常采用等效电路分析法；数字电路是指对数字信号进行传送、逻辑运算、控制、计数、寄存、显示以及脉冲信号的产生与变换等的电路。

3）数字电路的特点

与模拟电路相比，数字电路主要有下列优点：

（1）由于数字电路是以二值数字逻辑为基础的，只有 0 和 1 两个基本数字，易于用电路来实现，比如可用二极管的导通与截止这两个对立的状态来表示数字信号的逻辑 0 和逻辑 1。

（2）由数字电路组成的数字系统工作可靠，精度较高，抗干扰能力强。它可以通过整形很方便地去除叠加于传输信号上的噪声与干扰，还可利用差错控制技术对传输信号进行查错和纠错。

（3）数字电路不仅能完成数值运算，而且能进行逻辑判断和运算，这在控制系统中是不可缺少的。

（4）数字信息便于长期保存，比如可将数字信息存入磁盘、光盘等长期保存。

（5）数字集成电路产品系列多、通用性强、成本低。

由于数字电路具有一系列优点，数字电路在电子设备或电子系统中得到了越来越广泛的应用，计算机、计算器、电视机、音响系统、视频记录设备、光碟、长途电信及卫星系统等，无一不采用了数字系统。

2. 常用芯片

数字电路的常用芯片有三种：标准芯片、可编程逻辑器件和定制芯片。

1）标准芯片

将一些常用逻辑功能电路制造成芯片，它们在功能和规格上均符合公认的标准，称为标准芯片，属于小规模集成电路 SSI。设计者选择能完成某种功能的芯片并确定其连接方式，设计

出具有所需功能的电路。

20 世纪 80 年代以前，通常选用标准芯片设计逻辑电路。然而，随着集成电路技术的飞速发展，功能很少的标准芯片逐步被可编程逻辑器件代替。

2）可编程逻辑器件

可编程逻辑器件（programmable logic device，PLD）是一种大规模集成电路，通过对器件内部的编程来改变器件的逻辑功能。在使用前其内部是“空的”，采用一定的方式对其编程，可将其配置成具有特定逻辑功能的器件。可编程逻辑器件的逻辑功能可反复修改。设计者最初设计的是产品的原型，在此后的硬件测试运行过程中，可以发现问题，并通过可编程逻辑器件再编程，进行设计修正，或在原设计中加入新的功能对产品进行升级。可编程逻辑器件可以用来实现典型标准芯片无法实现的复杂大型逻辑电路。可编程逻辑器件也称为半定制芯片，可编程逻辑器件的出现使得产品设计变得容易。

3）定制芯片

定制芯片出厂时功能已经确定，是专为特殊应用所生产的，有时也称为专用集成电路芯片（application specific integrated circuit，ASIC）。定制芯片最大的优点在于可以针对特定任务做最优化设计，因此常常能够达到更高的性能，而且定制芯片中有可能比其他类型芯片集成更多的逻辑电路。虽然这种芯片的生产成本很高，但是如果批量大，将成本分摊至每个芯片，则每个芯片的成本降低。此外，ASIC 可以将多个芯片集成到一个芯片上，使产品体积缩小，从而降低成本。

**3. 课程内容与学习建议**

1）课程内容

本课程由脉冲和数字两大部分构成。脉冲部分主要介绍脉冲信号的概念以及脉冲信号的产生与整形等内容。数字部分主要包括组合逻辑电路和时序逻辑电路。组合逻辑电路介绍基本的逻辑运算及运算相应的门电路、逻辑函数等，同时介绍组合逻辑电路的分析与设计方法，重点介绍中规模集成器件的应用。时序逻辑电路的基本单元是触发器，这部分介绍常用触发器的工作原理、逻辑功能及描述方法，重点介绍典型时序电路（如计数器、寄存器、序列发生器等）的工作原理、分析与设计方法，尤其是集成器件的应用。

2）学习建议

（1）熟练掌握数字电路的分析、设计的基本方法与步骤，在实际应用中，除了关注电路的逻辑功能，还要考虑电路的性能要求，如功耗、响应速度和抗干扰能力等。

（2）数字电路以集成电路为主，学习中应注重掌握其外部特性，了解电路内部结构是为了更好地掌握数字电路的逻辑功能。

（3）数字电子技术课程是实践性很强的课程，加强实践性训练是掌握这门课程必不可少的条件，建议课后配套实验实训，本书最后一章提供了数字系统案例供学习借鉴。

# 第 1 章　数字逻辑基础

数字电路的基本单元是逻辑门电路，分析工具是逻辑代数，在功能上则着重强调电路输入与输出间的因果关系。数字电路抗干扰能力强、精度高、便于集成，因而在自动控制系统、测量设备、电子计算机等领域获得了日益广泛的应用。数字电路的研究对象是输出与输入之间的逻辑关系，可以用逻辑代数来描述。本章将首先介绍数字电路的一些基本概念及数字电路中常用的数制与码制，然后介绍逻辑运算及其基本定律，最后阐述逻辑函数的描述方法与化简。

## 1.1 数　制

数制就是计数进位制，它规定了数码处于不同位置所代表的数值。在日常生活中，人们常采用十进制，但在数字电路中通常采用二进制，有时也采用十六进制或八进制。

### 1.1.1　常用数制

数字电路中经常遇到计数问题，鉴于电路的开关特性，在数字系统中多采用二进制，有时用八进制和十六进制，而人们最熟悉的却是十进制。它们之间可以互相转换。

基数是指数制中所使用的数码的个数，也称为底数，反映进位规则。例如 $R$ 进制数的基数为 $R$，计数规则为"逢 $R$ 进一""借一作 $R$"。

对于任意数制 $(S)_R$，其数学描述均可表示为

$$(S)_R = \sum_{i=-m}^{n-1} a_i \times R^i \tag{1.1.1}$$

式中，$S$ 表示某个 $R$ 进制数，分别由 $R$ 个符号组合而成；$i$ 表示 $S$ 的位权；$n$、$m$ 分别表示 $S$ 整数和小数的位数；$a_i$ 表示 $S$ 第 $i$ 位的数码，且必定是上述 $R$ 个符号中的某一个。它的计数原则是逢 $R$ 进一，借一当 $R$。

**1. 十进制**

十进制(decimal)是指以 10 为基数的计数体制。十进制可用 0～9 十个数码来表示。它的数学描述如下

$$(S)_{10} = \sum_{i=-m}^{n-1} a_i \times 10^i \tag{1.1.2}$$

例如，一个十进制数 135.45 可表示为

$$(135.45)_{10} = 1\times10^2 + 3\times10^1 + 5\times10^0 + 4\times10^{-1} + 5\times10^{-2}$$

**2. 二进制**

二进制(binary)用 0 和 1 两个数码表示。超过 1 的数必须用多位数表示，其数学描述为

$$(S)_2 = \sum_{i=-m}^{n-1} a_i \times 2^i \tag{1.1.3}$$

例如，$(1011.101)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

**3. 八进制**

八进制(octal)用 0～7 八个数码表示，其数学描述为

$$(S)_8 = \sum_{i=-m}^{n-1} a_i \times 8^i \tag{1.1.4}$$

例如，$(135.45)_8 = 1\times8^2 + 3\times8^1 + 5\times8^0 + 4\times8^{-1} + 5\times8^{-2}$

**4. 十六进制**

十六进制(hexadecimal)用数字 0～9 和字母 A、B、C、D、E、F 共十六个符号表示，其数学描述为

$$(S)_{16} = \sum_{i=-m}^{n-1} a_i \times 16^i \tag{1.1.5}$$

例如，$(5AB.CD)_{16} = 5\times16^2 + 10\times16^1 + 11\times16^0 + 12\times16^{-1} + 13\times16^{-2}$

在数字电路中，为了区分不同数制所表示的数，可以采用括号加注下标的形式，也可以在数的后面加后缀，如二进制加后缀 B，八进制加后缀 Q(一般不用 O，以免被人误以为 0)，十进制加后缀 D(常将后缀 D 省略)，十六进制加后缀 H。例如，

$$(135.45)_{10} = 135.45\text{D} = 135.45$$

$$(1011.101)_2 = 1011.101\text{B}$$

$$(135.45)_8 = 135.45\text{Q}$$

$$(5\text{AB.CD})_{16} = 5\text{AB.CDH}$$

### 1.1.2 数制间的转换

通常数制的转换，主要体现在两方面，一方面是任意进制数与十进制数的转换；另一方面是二进制数、八进制数和十六进制数三者之间的转换。

**1. 任意进制数转换为十进制数**

将任意进制数转换为十进制数的方法为：按权相加，就可以得到所对应的十进制数。

例如，二进制数 101.01 转换成十进制数可以表示为

$$(101.01)_2 = 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} = (5.25)_{10}$$

八进制数 167 转换成十进制数可以表示为

$$(167)_8 = 1\times8^2 + 6\times8^1 + 7\times8^0 = (119)_{10}$$

十六进制数 1C4.68 转换成十进制数可以表示为

$$(1\text{C}4.68)_{16} = 1\times16^2 + 12\times16^1 + 4\times16^0 + 6\times16^{-1} + 8\times16^{-2} = (452.40625)_{10}$$

**2. 十进制数转换为二进制数**

把十进制数转换为 $R$ 进制数的方法为：整数部分除 $R$ 取余数，小数部分乘 $R$ 取整数。

首先讨论整数部分的转换。

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(1.1.3)展开可得到

$$\begin{aligned}(S)_{10} &= k_n \times 2^n + k_{n-1} \times 2^{n-1} + \cdots + k_1 \times 2^1 + k_0 \times 2^0 \\ &= (k_n \times 2^{n-1} + k_{n-1} \times 2^{n-2} + \cdots + k_1) \times 2 + k_0\end{aligned} \tag{1.1.6}$$

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

同理，将得到的商再除以 2，得到的商为 $k_n \times 2^{n-2} + k_{n-1} \times 2^{n-3} + \cdots + k_2$，余数为 $k_1$，依此类推，反复将每次得到的商除以 2，取其余数，就可以求得二进制整数的每一位。

其次讨论小数部分的转换。

若 $(S)_{10}$ 是一个十进制的小数，对应的二进制小数为 $(0.k_{-1}k_{-2}\cdots k_{-m})_2$，则按式(1.1.3)展开可得到

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

$$2(S)_{10} = k_{-1} + (k_{-2} \times 2^{-1} + k_{-3} \times 2^{-2} + \cdots + k_{-m} \times 2^{-m+1}) \tag{1.1.7}$$

式(1.1.7)表明，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 又可以得到

$$2(k_{-2} \times 2^{-1} + k_{-3} \times 2^{-2} + \cdots + k_{-m} \times 2^{-m+1}) = k_{-2} + (k_{-3} \times 2^{-1} + \cdots + k_{-m} \times 2^{-m+2}) \tag{1.1.8}$$

乘积的整数部分即 $k_{-2}$。依此类推，反复将每次乘积的小数部分乘以 2，取其整数部分，就可以求得二进制小数的每一位。

例如，将 $(26.625)_{10}$ 转换为二进制数步骤如下：

(1) 整数部分的转换：

$$\begin{array}{r|l l}
2 & 26 & \\
2 & 13 & 余0 \\
2 & 6 & 余1 \\
2 & 3 & 余0 \\
2 & 1 & 余1 \\
 & 0 & 余1
\end{array} \quad \uparrow 读取方向$$

故 $(26)_{10} = (11010)_2$。

(2) 小数部分的转换：

$$\begin{array}{r l}
 & 0.\ 6\ 2\ 5 \\
\times & 2 \\
\hline
 & 1.\ 2\ 5\ 0 \quad 整数\ 1(k_{-1}) \\
 & 0.\ 2\ 5\ 0 \\
\times & 2 \\
\hline
 & 0.\ 5\ 0\ 0 \quad 整数\ 0(k_{-2}) \\
 & 0.\ 5\ 0\ 0 \\
\times & 2 \\
\hline
 & 1.\ 0\ 0\ 0 \quad 整数\ 1(k_{-3})
\end{array} \quad \downarrow 读取方向$$

故 $$(0.625)_{10} = (0.101)_2$$

所以有 $$(26.625)_{10} = (11010.101)_2$$

**3. 二进制数和十六进制数的相互转换**

因为 4 位二进制数恰好有 16 个状态，所以 4 位二进制数与 1 位十六进制数有直接对应关系，即 4 位二进制数可直接写为 1 位十六进制数，1 位十六进制数也可直接写为 4 位二进制数。

将十六进制数转换为二进制数的方法是：将十六进制数的每一位用等值的 4 位二进制数

代替。例如，将$(1100010.11001)_2$转换为十六进制数时可以得到

$$\frac{0110}{6}\ \frac{0010.}{2.}\ \frac{1100}{C}\ \frac{1000}{8}$$

故
$$(1100010.11001)_2=(62.C8)_{16}$$

同理，十六进制数转换为二进制数时，只需要将十六进制数的每一位用等值的4位二进制数代替就可以了。例如，$(B3.7)_{16}=(10110011.0111)_2$。

**4. 二进制数与八进制数之间的转换**

因为3位二进制数有8个状态，所以3位二进制数与1位八进制数有直接对应关系，即3位二进制数可直接写为1位八进制数，1位八进制数也可直接写为3位二进制数。

将二进制数转换为八进制数的方法是：将二进制数整数部分自右至左每3位分为一组，最后不足3位时左边用0补足；小数部分自左至右每3位分为一组，最后不足3位时在右边用0补足。将八进制数转换为二进制数时，只需将八进制数的每一位用等值的3位二进制数代替就行了。

例如，将$(1100010.10111)_2$转换为八进制数可以得到

$$\frac{001}{1}\ \frac{100}{4}\ \frac{010.}{2.}\ \frac{101}{5}\ \frac{110}{6}$$

故
$$(1100010.10111)_2=(142.56)_8$$

同理，八进制数转二进制数时，只需将八进制数的每一位用等值的3位二进制数替代。

**【思考】** 在十-二转换中，整数部分的转换方法和小数部分的转换方法有何不同?

### 1.1.3 算术运算

在数字电路中，1位二进制数码的0和1不仅可以表示数量的大小，而且可以表示两种不同的逻辑状态。当两个二进制数码表示两个数量大小时，它们之间可以进行数值运算，这种运算称为算术运算。二进制数的算术运算可以分为无符号二进制数和有符号二进制数的算术运算。

**1. 无符号二进制数的算术运算**

二进制数的加、减、乘、除4种运算规则与十进制数类似，唯一的区别在于进位和借位的规则不同。

1) 二进制数加法

无符号二进制数加法规则是“逢二进一”，即$0+0=0$，$0+1=1$，$1+1=10$。

例如，计算两个二进制数1001与0101的和：

$$\begin{array}{r} 1\ 0\ 0\ 1 \\ +\ 0\ 1\ 0\ 1 \\ \hline 1\ 1\ 1\ 0 \end{array}$$

所以
$$1001+0101=1110$$

无符号二进制数的加法运算是算术运算的基础，数字系统中的各种运算都将通过它来进行。

2) 二进制数减法

无符号二进制数减法法则是“借一作二”，即$0-0=0$，$1-1=0$，$1-0=1$，$0-1=1$。

其中，0 减 1 不够减，所以向高位借 1。

例如，计算两个二进制数 1001 与 0101 的差：

$$\begin{array}{r} 1\ 0\ 0\ 1 \\ -\ 0\ 1\ 0\ 1 \\ \hline 0\ 1\ 0\ 0 \end{array}$$

所以 $1001-0101=0100$

由于无符号二进制数中无法表示负数，所以要求被减数一定要大于减数。

3）二进制数乘法和除法

乘法运算是由左移被乘数和加法运算组成，而除法运算是由右移被除数和减法运算组成。例如，两个二进制数 1001 和 0101 的乘法运算和除法运算为：

$$\begin{array}{r} 1\ 0\ 0\ 1 \\ \times\ \ 0\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0\ \ \\ 1\ 0\ 0\ 1\ \ \ \ \\ 0\ 0\ 0\ 0\ \ \ \ \ \ \\ \hline 1\ 0\ 1\ 1\ 0\ 1 \end{array} \qquad \begin{array}{r} 1.\ 1\ 1\ \cdots \\ 1\ 0\ 1\overline{)1\ 0\ 0\ 1\ \ \ \ \ \ } \\ \underline{1\ 0\ 1\ \ \ \ \ \ \ \ } \\ 1\ 0\ 0\ 0\ \ \ \ \\ \underline{1\ 0\ 1\ \ \ \ \ } \\ 1\ 1\ 0\ \ \\ \underline{1\ 0\ 1\ \ } \\ 1\ \ \end{array}$$

所以 $1001\times0101=101101$， $1001\div0101=1.11\cdots$

**2. 有符号二进制数的算术运算**

1）有符号二进制数的补码

二进制数的负数需要用有符号的二进制数表示，在定点运算的情况下，二进制数的最高位表示符号位，用 0 表示正数，用 1 表示负数，其余部分为数值位。其表示形式有原码、反码和补码 3 种。

原码：最高位为符号位，数值位为绝对值对应的二进制数。例如 $(+12)_{10}=(01100)_2$，$(-12)_{10}=(11100)_2$。其中，二进制数的最左边的位即最高位代表符号，其余 4 位表示数值。

反码：正数的反码与原码相同，负数的反码是符号位不变，数值位为原码各位取反。例如 $(+12)_{反}=(01100)_2$，$(-12)_{反}=(10011)_2$。

补码：正数的补码与原码相同，负数的补码是符号位不变，数值位在反码的数值位最低位加 1。例如 $(+12)_{补}=(01100)_2$，$(-12)_{补}=(10100)_2$。

在数字系统中，常常将负数用补码表示，以便将减法运算变为加法运算。

2）有符号二进制数的减法运算

采用补码的形式，可以很方便地进行有符号二进制数的减法运算。减法运算的原理是减去一个正数，相当于加上一个负数，即 $A-B=A+(-B)$，对 $-B$ 求补码，然后进行加法运算。

例如，用 4 位二进制补码计算 5－3 的过程如下：

$$\begin{aligned}(5-3)_{补}&=(5)_{补}+(-3)_{补}\\&=0101+1101=0010\end{aligned} \qquad \begin{array}{r} 0\ 1\ 0\ 1 \\ +\ \ \ 1\ 1\ 0\ 1 \\ \hline [1]\ 0\ 0\ 1\ 0 \end{array}$$

自动丢弃

两个二进制补码相加时，超过位数的进位在计算中自动丢弃，所以$(5-3)_{补}=(0010)_2$。

3）溢出

$n$位有符号的二进制数的原码、反码和补码的数值范围分别为：

原码

$$-(2^{n-1}-1)\sim+(2^{n-1}-1)$$

反码

$$-(2^{n-1}-1)\sim+(2^{n-1}-1)$$

补码

$$-2^{n-1}\sim+(2^{n-1}-1)$$

当计算结果超过此数值范围时，就会产生溢出。

例如，用4位二进制补码计算5+7，得到

$$(5+7)_{补}=(5)_{补}+(7)_{补}=0101+0111=1100$$

计算结果1100表示－4，而实际正确的结果应该是12，错误产生的原因在于4位二进制补码表示的范围是－8～＋7，而本例中的结果12超出了4位二进制补码表示的范围，因而产生了溢出。解决溢出的办法是进行位扩展，即用更多位的二进制补码来表示，就不会产生溢出了。

【思考】 二进制正、负数的原码、反码和补码三者之间是什么关系？

## 1.2 码 制

不同的数码不仅可以表示数量的不同大小，而且能用来表示不同的事物。在后一种情况下，这些数码已没有数量大小的含义，只是表示不同事物的代号而已，因此这些数码称为代码。为了便于记忆和处理，在编制代码时总要遵循一定的规则，这些规则就叫作码制。

### 1.2.1 二-十进制(BCD)编码

用于表示十进制数的二进制代码称为二-十进制(binary coded decimal)编码，简称为BCD码。它具有二进制数的形式以满足数字系统的要求，又具有十进制数的特点(只有10种数码状态有效)。因为4位二进制数有16种状态，而十进制数只需要10种，从16种状态中选择10种，就有多种组合，这样就有多种编码，表1.2.1中列出了几种常见的BCD码。

表1.2.1 几种常见的BCD码

| 十进制数 | 8421码 | 2421码 | 5421码 | 余3码 |
|---|---|---|---|---|
| 0 | 0000 | 0000 | 0000 | 0011 |
| 1 | 0001 | 0001 | 0001 | 0100 |
| 2 | 0010 | 0010 | 0010 | 0101 |
| 3 | 0011 | 0011 | 0011 | 0110 |
| 4 | 0100 | 0100 | 0100 | 0111 |
| 5 | 0101 | 1011 | 1000 | 1000 |
| 6 | 0110 | 1100 | 1001 | 1001 |

续表

| 十进制数 | 8421码 | 2421码 | 5421码 | 余3码 |
|---|---|---|---|---|
| 7 | 0111 | 1101 | 1010 | 1010 |
| 8 | 1000 | 1110 | 1011 | 1011 |
| 9 | 1001 | 1111 | 1100 | 1100 |
| 权 | 8421 | 2421 | 5421 | 无 |

8421码是BCD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,即从左到右每位的权值分别为8、4、2和1,按权相加即可得该码所表示的十进制数,得到的结果就是它所代表的十进制数码。

2421码也是有权码,对应的权值由高到低分别为2、4、2、1。

5421码也是有权码,它各位的权由高到低分别为5、4、2、1。

余3码的特点是在8421码的基础上加3。

为了能发现和校正错误,提高设备的抗干扰能力,就需采用可靠性代码。格雷码是常见的可靠性代码。典型格雷码构成规则为:最低位以0110为循环节;次低第二位以00111100为循环节;次低第三位以0000111111110000为循环节,依次类推可得表1.2.2。

表1.2.2 格雷码与二进制码关系对照表

| 十进制数 | 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 |

## 1.2.2 字符、数字代码

在计算机的应用过程中,如操作系统命令、各种程序设计语言以及计算机运算和处理信息

的输入输出，经常用到某些字母、数字或各种符号。但在计算机内，任何信息都是用代码表示的，因此这些符号也必须要有自己的编码。

用若干位二进制符号表示数字、英文字母、命令以及特殊符号叫作字符编码，常用的字符编码是美国标准信息交换码（American standard code for information interchange，简称ASCII码，见表1.2.3），它由7位二进制符号组成，它共有128个代码，可以表示大小写英文字母、十进制数、标点符号、运算符号、控制符号等。

ASCII码是目前大部分计算机与外部设备交换信息的字符编码。例如，键盘将按键的字符用ASCII码表示送入计算机，而计算机对处理好的数据也是用ASCII码传送到显示器或打印机。

**表1.2.3 美国标准信息交换码（ASCII码）**

| $b_3b_2b_1b_0$ | $b_6b_5b_4$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 0000 | NUL | DLE | SP | 0 | @ | P | ` | p |
| 0001 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 0010 | STX | DC2 | " | 2 | B | R | b | r |
| 0011 | ETX | DC3 | # | 3 | C | S | c | s |
| 0100 | EOT | DC4 | $ | 4 | D | T | d | t |
| 0101 | ENQ | NAK | % | 5 | E | U | e | u |
| 0110 | ACK | SYN | & | 6 | F | V | f | v |
| 0111 | BEL | ETB | ' | 7 | G | W | g | w |
| 1000 | BS | CAN | ( | 8 | H | X | h | x |
| 1001 | HT | EM | ) | 9 | I | Y | i | y |
| 1010 | LF | SUB | * | : | J | Z | j | z |
| 1011 | VT | ESC | + | ; | K | [ | k | { |
| 1100 | FF | FS | , | < | L | \ | l | \| |
| 1101 | CR | GS | - | = | M | ] | m | } |
| 1110 | SO | RS | . | > | N | ^ | n | ~ |
| 1111 | SI | US | / | ? | O | _ | o | DEL |

**【思考】** 8421码、2421码、5421码、余3码在编码规则上各有何特点？

## 1.3 逻辑运算

事物往往存在两种对立的状态，如电灯的亮与暗、开关的通与断、电平的高与低等。在逻辑代数中，为了描述事物两种对立的逻辑状态，采用的是仅有两个取值的变量。这种变量称为逻辑变量。逻辑变量与普通代数变量一样，都用字母表示。但是，它和普通代数变量有着本质的区别，逻辑变量的取值只有两种，即逻辑0和逻辑1，0和1称为逻辑常量，它们并不表示数

量的大小,而是表示两种对立的逻辑状态。

### 1.3.1　三种基本逻辑运算

逻辑代数的基本运算有与、或、非三种。下面结合指示灯控制电路的实例分别讨论。

**1. 与运算**

图 1.3.1 给出了指示灯的两开关串联控制电路。由图可知,只有开关 $A$ 与开关 $B$ 全部闭合,指示灯 $Y$ 才会亮,否则指示灯不亮。

由此得到这样的逻辑关系:只有决定事物结果的若干条件全部满足时,结果才会发生。这种条件和结果的关系称为逻辑与。

在逻辑代数中,把逻辑变量之间的逻辑与关系称为与运算,也叫逻辑乘,并用符号“·”表示“与”。因此,输入量 $A$、$B$ 与输出量 $Y$ 的与逻辑关系可写成

$$Y = A \cdot B \tag{1.3.1}$$

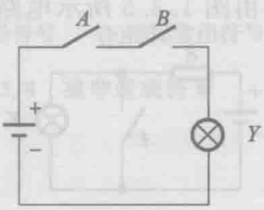


**图 1.3.1　串联电路**

“·”在表达式中常被省略。

在逻辑代数中,逻辑关系除了可以用逻辑函数表达式表示外,还可以用真值表和逻辑符号表示。这里开关闭合和灯亮用 1 表示,开关断开和灯不亮用 0 表示,则可得到表 1.3.1。这种用逻辑变量的真正取值反映逻辑关系的表格称为逻辑真值表,简称真值表。

为了方便数字逻辑电路的分析与设计,各种逻辑运算还可用逻辑符号表示,与逻辑的逻辑符号如图 1.3.2 所示。

**表 1.3.1　与逻辑真值表**

| $A$ | $B$ | $Y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |


(a)国标图形符号　(b)国际常用符号

**图 1.3.2　与逻辑符号**

**2. 或运算**

图 1.3.3 给出了指示灯的两开关并联控制电路。由图可知,开关 $A$ 或开关 $B$ 只要有一个闭合,指示灯 $Y$ 就亮,否则指示灯不亮。

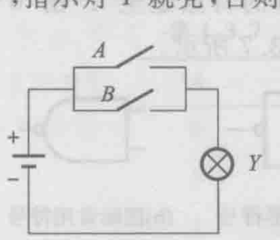


**图 1.3.3　并联开关电路**

开关 $A$、$B$ 和指示灯 $Y$ 的逻辑关系表明:在决定事物结果的若干条件中,只要满足一个或一个以上条件时,结果就会发生。这种因果关系称为逻辑或,也叫或逻辑关系。

在逻辑代数中,把逻辑变量之间的或逻辑关系称为或运算,也叫逻辑加,并用符号“+”表示“或”。因此输入量 $A$、$B$ 与输出量 $Y$ 的或逻辑关系可写成

$$Y = A + B \tag{1.3.2}$$

按照前述假设,用二值逻辑变量可以列出或逻辑的真值表,即开关断开和灯不亮均用“0”表示,而开关闭合和灯亮均用“1”表示,或逻辑真值表如表 1.3.2 所示。或逻辑关系也可以用逻辑符号表示,图 1.3.4 所示为或逻辑符号。