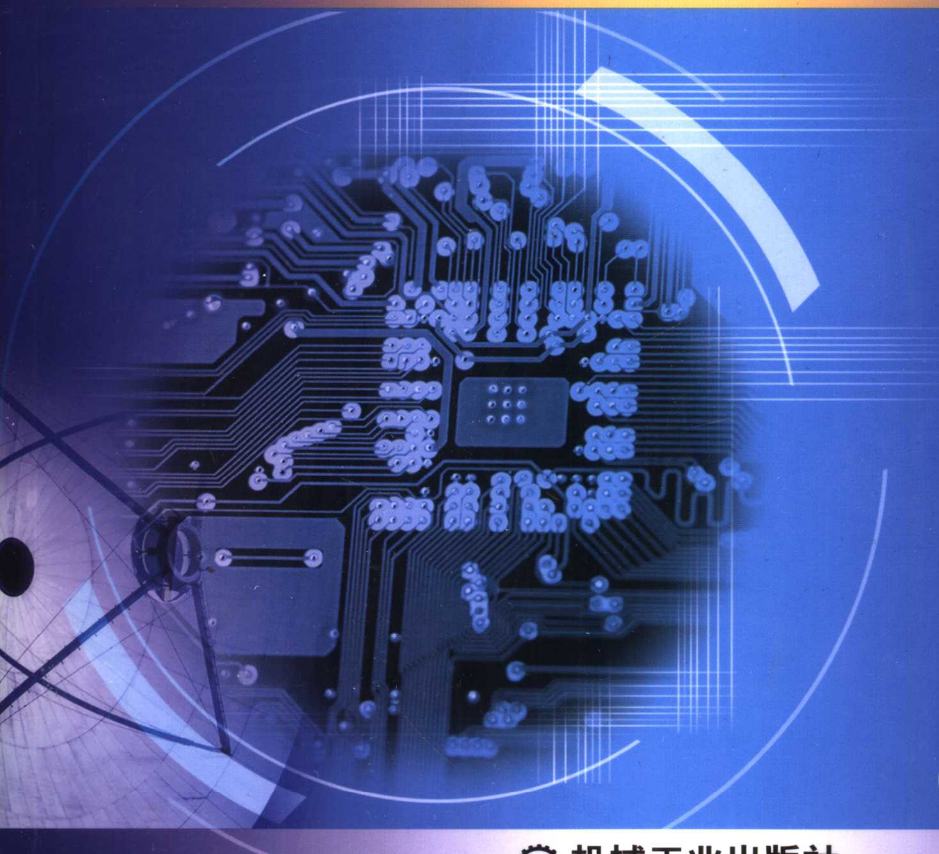
普通高等教育“十一五”电子电气基础课程规划教材

数字逻辑与数字电路

徐晓光　编著



附光盘

机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十一五”电子电气基础课程规划教材

数字逻辑与数字电路

徐晓光 编著
张 元 主审

机 械 工 业 出 版 社

本书是数字逻辑与数字电路课程的教材，共8章，主要内容有数字逻辑基础，逻辑化简，组合逻辑电路，触发器、寄存器和计数器，同步时序逻辑电路，异步时序逻辑电路，半导体存储器与可编程逻辑技术，数字逻辑电路设计实例等。

本书综合了本学科的理论和知识、并充分考虑了与相关课程的联系，汲取了国外最新原版教材的精华，介绍了数字集成电路的最新发展。

本书附录内容有常用脉冲产生电路，ADC 和 DAC 技术，VHDL 语言简介，二进制逻辑元件的图形符号等。

该书内容丰富、全面，知识新颖。既有详实的理论分析，又注重知识的实用性。

书中所附光盘内包含有“卡诺图软件”和“TTL、CMOS 和高速 CMOS 数字集成电路数据库”两个软件。在电脑日益普及的今天，它们为读者学习、应用书中的知识提供了极好的帮助。

本书配有免费电子课件，欢迎选用本书作教材的老师索取，电子邮箱：wbj@ mail. machineinfo. gov. cn。

本书可以作为相关课程的本科教材，也十分适合于工程技术人员使用。

图书在版编目（CIP）数据

数字逻辑与数字电路/徐晓光编著．—北京：机械工业出版社，2007．8
ISBN 978-7-111-21967-5

Ⅰ．数…　Ⅱ．徐…　Ⅲ．①数字逻辑②数字电路：逻辑电路
Ⅳ．TP302．2　TN79

中国版本图书馆 CIP 数据核字（2007）第 113747 号

机械工业出版社（北京市百万庄大街22号　邮政编码100037）
责任编辑：王保家　版式设计：张世琴　责任校对：张　媛
封面设计：张　静　责任印制：杨　曦
北京机工印刷厂印刷（北京双新装订有限公司装订）
2008年1月第1版·第1次印刷
184mm×260mm·16．75印张·410千字
标准书号：ISBN 978-7-111-21967-5
　　　　　ISBN 978-7-89482-267-3（光盘）
定价：30．00元（含1CD）

前　　言

《数字逻辑与数字电路》是一本讲述数字电子技术原理和应用知识的教科书。它是作者在参考了大量的相关资料、结合多年的教学科研经验之后，精心编写而成的一本具有鲜明特色的书籍。

本书综合了本学科的各种理论和知识、并充分考虑了本课程与其他相关课程，如单片机原理和应用、通信电子线路等的联系。

本书汲取了国外最新原版教材的精华，介绍了数字集成电路的最新发展成果。

数字逻辑是一门历史悠久的学科，在人类历史上有关数字逻辑的研究很早就已经开始了。近年来随着科学技术的进步，数字电子技术有了飞速的发展。各种新技术、新器件层出不穷。即使是在21世纪的今天，数字逻辑仍在不断进步和发展。近年来，数字逻辑集成电路在高速、低功耗、低电压、带电插拔、小逻辑等许多方面都取得了长足的进步，相关的产品已经投入市场。这些内容在本书中皆有阐述。

鉴于作者对于数字逻辑的研究和认识，在本书内容中编写了一般中文教材没有涉及的许多知识，如对“蕴涵”、“质蕴涵”和“必要质蕴涵”等基本概念进行了详细的介绍；系统性地讲解了数字逻辑集成电路实际应用中各种问题的解决方案等。

IEC逻辑符号（二进制逻辑元件图形符号的中国国家标准采用的即为IEC标准，见附录C）的优点是明显的，它是一种功能强大的符号语言，能够很好地描述逻辑元件的逻辑功能。

二进制逻辑元件的IEC标准能够直接由符号给出逻辑元件的功能信息，从而能够通过元件符号直接把握元件的功能和特性，而不再依赖于元件的真值表和原理图。因此，本书对IEC逻辑符号进行了详细的说明，并将其大量运用到各章节内容中。

对于目前国外仍在大量使用的“特异形逻辑符号”也不应当完全加以排斥，因为这种传统逻辑符号有着它自己的特点，而且这种符号在许多文献上仍十分常见。因此，在本书中对特异形逻辑符号采取了兼收并蓄的态度。

针对目前普通存在的读者学习了数字逻辑电路理论知识后，却不善于解决实际问题的现象，本书编写了“数字逻辑电路设计实例”一章。它对于运用数字逻辑理论进行实际电路的设计，具有很好的参考价值。

另外，本书附带有作者编写的两个软件：“TTL、CMOS和高速CMOS集成电路数据库”和“卡诺图软件2.0”。

“TTL、CMOS和高速CMOS集成电路数据库”软件包含有完整的TTL、CMOS和高速CMOS集成电路的数据资料，同时又具备十分方便的信息查找功能。

由于资料完备，该软件对于读者学习逻辑集成电路的知识、了解数字逻辑集成电路的整体状况，是十分有益的。此外，它也是从事数字逻辑电路设计不可多得的工具。因为一个优秀的电路设计，必须从数量众多、信息完备的相关资料中挑选出合适的集成电路，以最佳方式来满足设计的要求。而该数据库软件正好提供了这样的便利。

“卡诺图软件”是一个用于逻辑化简的计算机辅助分析软件；具有新颖、实用的特点；能够实现逻辑函数的自动化简，它具有多种输入功能，包括真值表、最小项、最大项和表达式输入等；支持POS、SOP、正函数和反函数等各种函数形式。因此，“卡诺图软件”是学习和应用数字逻辑课程知识的一个有力工具。

可以说，上述软件与图书配合，构成了对于课程知识诠释的完美结合。它适应了计算机普及应用的当今时代，赋予了本书与众不同的鲜明特色。

关于HDL语言，本书也作了恰如其分的处理。HDL语言是计算机得到广泛应用后，出现的一种描述数字逻辑电路结构和功能的计算机语言。它十分适用于可编程逻辑器件的开发设计，以及超大规模集成电路的芯片设计。由于上述内容并不属于本书的研究领域，所以作者没有将HDL的详细知识写入到本书的正文，而是在书后的附录中对VHDL进行了简要的介绍。

考虑到一些学校的课时可能不足，书中带*的内容为选修内容。

同时，本书的编写力求做到语言生动、避免写成语言呆板的产品说明书风格。

最后要感谢大家对本书出版的支持和帮助。本书固然是作者大量心血的结晶，但它的诞生也离不开众多前辈的研究成果，离不开各位同事、朋友和家人的关心和帮助。在此我要衷心地感谢担任主审的张元教授和各位领导、朋友们，他们为本书提出了许多宝贵的建议，使我受益非浅。

书中引用了Texas Instruments、National Semiconductor和Xilinx等许多外国电子公司的技术资料，在此本人向这些公司表示诚挚的感谢。

本书的编写还参考了大量的其他参考文献和资料，在此向相关书籍和文章的作者表示衷心的谢意。

本书配有免费电子课件，欢迎选用本书作教材的老师索取，电子邮箱：wbj@mail.machineinfo.gov.cn。

由于本人水平有限，书中难免会存在许多错误和不足之处，恳请大家原谅并不吝赐教。

作者

于河南工业大学

联系方式：Email：xuxiaoguang01@163.com

目　录

第 1 章　数字逻辑基础

1.1　概述

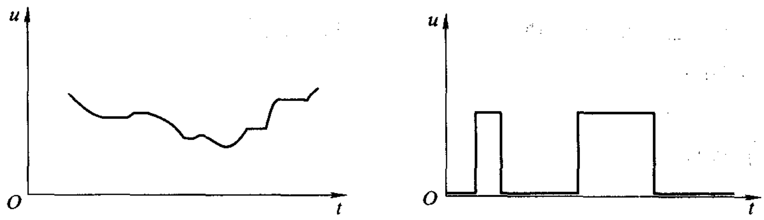
电子电路可以分为**模拟电路**与**数字电路**两大类。模拟电路所处理的信号为**模拟信号**，数字电路所处理的信号是**数字信号**。自然界存在的物理量，如声音、电流、电压等均为模拟的。模拟信号的特点是其数值随时间变化时呈现**连续**性。数字信号是一种人为加工产生的信号，它的信号大小随时间变化时呈现出**离散**特性。

模拟信号在任一时刻的数值大小可以是任意数值，而数字信号在任一时刻的数值大小只能取两个数值——高电平与低电平。

数字电子电路构成的电子装置，具有精度高，抗干扰能力强等许多优点。数字电路还是数字电子计算机的构成基础。

早期的电子系统都是模拟电子系统，随着科学技术的发展，数字电子系统得到了越来越广泛的应用。如通信系统、测量仪器仪表、自动控制系统等都已经从传统的模拟系统发展成为数字电子系统。

图 1-1 是模拟信号和数字信号的波形图。



模拟信号　　　　数字信号

图 1-1　模拟信号和数字信号的波形图

图 1-2 是模拟交流电压表和数字交流电压表的框图，从中能够比较出模拟电路和数字电路两种方案各自的特点。

20 世纪 60 年代小规模的数字集成电路诞生，其后中规模、大规模和超大规模数字集成电路相继问世。目前能够方便地改变芯片逻辑功能的**可编程逻辑器件**代表了数字逻辑集成电路的最高水平。

近年来微型计算机和嵌入式电子系统在生产实践和科学实验中得到了广泛的应用。虽然这些电子系统的核心为大规模和超大规模数字集成电路，但一个功能完整的电子系统，还离不开中小规模数字集成电路，如 TTL、CMOS 和高速 CMOS 系列等集成电路的配合。

数字电路又可以分为**组合逻辑电路**与**时序逻辑电路**两大类。因此数字电路又被称为**数字逻辑电路**。

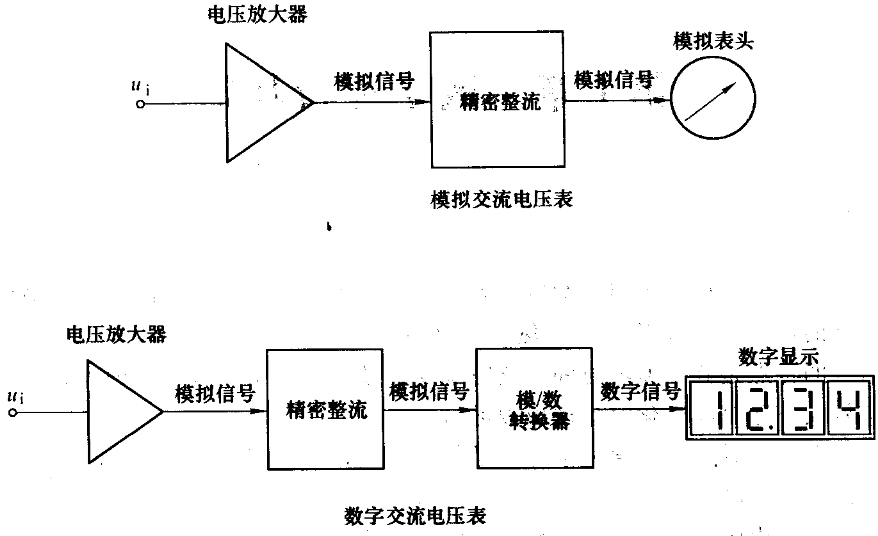


图 1-2　模拟交流电压表与数字交流电压表框图

数字逻辑是一门研究数字逻辑设计问题的理论课程。数字逻辑既是一门历史悠久的学科，又是一门生机勃勃的学科。特别是电子计算机的广泛应用，极大地推动了数字逻辑分析与设计技术的发展。

现在，**计算机辅助分析**在数字逻辑设计中的应用日益成熟；用于数字逻辑分析的计算机软件，已成为数字逻辑设计的有力工具。

本章将介绍数字逻辑的基本概念，如二进制数、编码、逻辑关系、逻辑门和逻辑代数等，为后面章节的学习打下基础。

1.2　数制与编码

1.2.1　十进制数

十进制是通常使用的数制（计数体系）。所谓十进制就是用 0、1、2、3、4、5、6、7、8、9 这 10 个符号排列起来，以表示所有的数值。

当数值大于 9 时，就采用进位的方法，进位的法则是：逢十进一。

任何一个形式为 $q_{n-1}q_{n-2}q_{n-3}\cdots q_1q_0.q_{-1}q_{-2}\cdots q_{-m}$ 的十进制数的大小 N，可以表示为

$$N = q_{n-1}\times 10^{n-1} + q_{n-2}\times 10^{n-2} + \cdots q_1\times 10^1 + q_0\times 10^0 + q_{-1}\times 10^{-1} + q_{-2}\times 10^{-2} + \cdots q_{-m}\times 10^{-m} \quad (1\text{-}1)$$

例如，$(2005.67)_{10} = 2\times 10^3 + 0\times 10^2 + 0\times 10^1 + 5\times 10^0 + 6\times 10^{-1} + 7\times 10^{-2}$，式中，$10^k$ 称为十进制数各位的“权”；10 为十进制计数体系的“基数”。

在多位数中，每一个相同的数码位于不同的位置时，代表着不同的数值。这是因为不同的位置具有不同的权的缘故。

十进制数常用字母 D（Decimal）进行标识，例如（1961）$_D$、（2005）$_D$ 等。

1.2.2 二进制数

为了用电学物理量来表示十进制，就需要找一种有 10 个状态的物理量，这是比较困难的。而有两个状态的物理量却非常容易找到。因此在数字电路中，理所当然地要采用二进制计数体系。在数字电路中采用高电平与低电平来分别表示两个不同的数码。

所谓二进制就是用两个符号 0 和 1，来表示所有的数值。当数大于 1 时，也采用进位的方法；但是进位的法则为逢二进一。

任何一个二进制数 $q_{n-1}q_{n-2}q_{n-3}\cdots q_1q_0.q_{-1}q_{-2}\cdots q_{-m}$的大小 N，可以表示为

$$\begin{aligned}N &= q_{n-1}\times 2^{n-1}+q_{n-2}\times 2^{n-2}+\cdots q_1\times 2^1+q_0\times 2^0+q_{-1}\times 2^{-1}\\&\quad+q_{-2}\times 2^{-2}+\cdots q_{-m}\times 2^{-m}\end{aligned} \tag{1-2}$$

例如，$(1001.101)_2=1\times 2^3+0\times 2^2+0\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}=(9.625)_{10}$，式中，$2^k$ 称为二进制数各位的“权”。二进制数的“基数”为 2。

应当注意：二进制数的各位数码 q_k 只能取 0 或 1。

二进制数常用字母 B（Binary）进行标识，例如（10101）$_B$、（1001）$_B$ 等。

二进制数的最高位数码，常用 **MSB**（Most Significant Bit）进行标识。

二进制数的最低位数码，常用 **LSB**（Least Significant Bit）进行标识。

在一个二进制数中，其 MSB 是衡量该二进制数数值大小最为重要的数码；而 LSB 是对该二进制数数值大小影响最小的数码。

1. 二进制数的算术运算

（1）二进制数的加法运算　一位二进制数的加法运算规则如下所示：

$$0+0=0,\ 0+1=1+0=1,\ 1+1=10$$

多位二进制的加法运算可以仿照大家熟悉的十进制数运算方法进行，不过要记住“逢二进一”。

[例 1-1]　计算（10101）$_B$ +（11001）$_B$。

解：

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

所以，（10101）$_B$ +（11001）$_B$ =（101110）$_B$。

（2）二进制数的乘法运算　一位二进制数的乘法运算规则如下所示：

$$0\times 0=0,\ 0\times 1=1\times 0=0,\ 1\times 1=1$$

多位二进制的乘法运算，与十进制数的乘法运算类似。它由乘数的每一位与被乘数相乘，产生乘数每一位的部分乘积，每个部分乘积都向前一个部分乘积的左方移动一位，然后将这些部分乘积求和而得。

[例 1-2]　计算（10101）$_B$ ×（11001）$_B$。

解：

$$\begin{array}{r}
10101 \\
\times\ 11001 \\
\hline
10101 \\
00000 \\
00000 \\
10101 \\
10101 \\
\hline
1000001101
\end{array}$$

所以，$(10101)_B \times (11001)_B = (1000001101)_B$。

（3）二进制数的减法运算　减法运算是加法运算的逆运算，当某一位二进制数不够减时，应当向高一位借位，借1当2。

［例1-3］　计算 $(101010)_B - (11001)_B$。

解：

$$\begin{array}{r}
101010 \\
-\quad 11001 \\
\hline
10001
\end{array}$$

所以，$(101010)_B - (11001)_B = (10001)_B$。

（4）二进制数的除法运算　除法运算是乘法运算的逆运算，其运算过程与十进制数的除法运算类似，使用了减法和移位技术。

［例1-4］　计算 $(1111101)_B \div (11001)_B$。

解：

$$\begin{array}{r@{\,}r}
 & 0000101 \\
\cline{2-2}
11001 \big) & 1111101 \\
 & 11001 \\
\cline{2-2}
 & 11001 \\
 & 11001 \\
\cline{2-2}
 & 0
\end{array}$$

所以，$(1111101)_B \div (11001)_B = (101)_B$。

1.2.3　二进制数与十进制数的相互转换

1. 二进制数转换成十进制数　二进制数转换成十进制数可按式（1-2）进行

$$(q_{n-1}q_{n-2}q_{n-3}\cdots q_1q_0.\ q_{-1}q_{-2}\cdots q_{-m})_B$$
$$=(q_{n-1}\times 2^{n-1}+q_{n-2}\times 2^{n-2}+\cdots q_1\times 2^1+q_0\times 2^0+q_{-1}\times 2^{-1}+q_{-2}\times 2^{-2}+\cdots q_{-m}\times 2^{-m})_D$$

［例1-5］　将二进制数1011100110转换成相应的十进制数。

解：

$$(1011100110)_2 = 1\times 2^9 + 1\times 2^7 + 1\times 2^6 + 1\times 2^5 + 1\times 2^2 + 1\times 2^1$$
$$= 512 + 128 + 64 + 32 + 4 + 2 = (742)_{10}$$

2. 十进制数转换成二进制数　十进制数转换成二进制数有许多方法，其中之一是“除 2 取余法”和“乘 2 取整法”。

“除 2 取余法”用于将十进制数的整数部分转换成二进制数。

“除 2 取余法”的算法流程如图 1-3 所示。

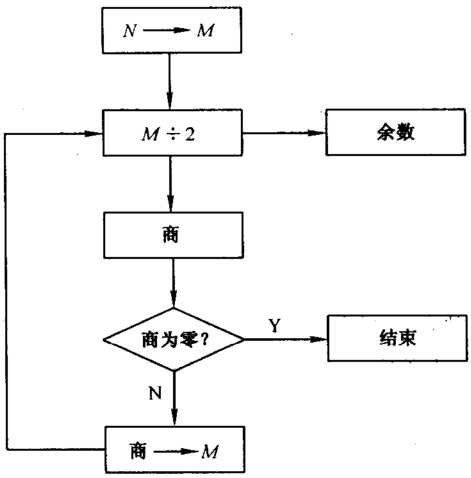


图 1-3　除 2 取余法算法流程

在图 1-3 中，N 为被转换的十进制数整数部分，M 为运算所需的中间变量。产生的第一个余数为相应二进制数的最低位 LSB，将运算所得的各个余数排列起来，即为转换后的二进制数整数。

[**例 1-6**]　将十进制数 2005 转换成二进制数。

解：

2|2005…1 ——→ LSB
2|1002…0
2|501…1
2|250…0
2|125…1
2|62…0
2|31…1
2|15…1
2|7…1
2|3…1
2|1…1 ——→ MSB
0

所以，$(2005)_D = (11111010101)_B$。

“乘 2 取整法”用于将十进制数的小数部分转换成二进制数。

"乘 2 取整法"的算法流程如图 1-4 所示。

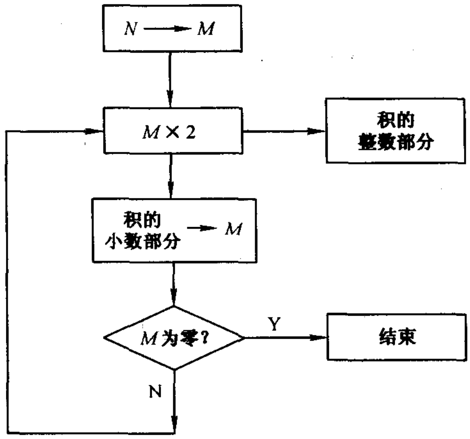


图 1-4 乘 2 取整法算法流程

在图 1-4 中，N 为被转换的十进制数小数部分，M 为运算所需的中间变量。产生的第一个整数为相应二进制小数的最高位 MSB，将运算所得的各个整数排列起来，即为转换后的二进制数小数。

[**例 1-7**] 将十进制数 0.8125 转换成二进制数。

解:

0.8125	0.625	0.25	0.5
× 2	× 2	× 2	× 2
1.625	1.25	0.5	1.0
⋮	⋮	⋮	⋮
1	1	0	1
MSB			LSB

所以，$(0.8125)_D = (0.1101)_B$。

1.2.4 十六进制数

所谓十六进制就是用 16 个符号 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 来表示所有的数值。十六进制数进位的法则为逢十六进一。

任何一个十六进制数 $q_{n-1}q_{n-2}q_{n-3}\cdots q_1q_0.q_{-1}q_{-2}\cdots q_{-m}$的大小 N，可以表示为

$$N = q_{n-1} \times 16^{n-1} + q_{n-2} \times 16^{n-2} + \cdots q_1 \times 16^1 + q_0 \times 16^0 + q_{-1} \times 16^{-1} + q_{-2} \times 16^{-2} + \cdots q_{-m} \times 16^{-m} \tag{1-3}$$

十六进制数的"基数"为 16。

应当注意：十六进制数的各位数码 q_k 可以取 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 之一。其中 A、B、C、D、E 和 F 分别代表十进制的 10、11、12、13、14 和

15。

十六进制数常用字母 H(Hexadecimal)进行标识，例如$(389)_H$、$(6AB2)_H$ 和$(3FFE)_H$ 等。通常十六进制数也可以用下面形式表示，如 389H，6AB2H 等。

十六进制的用途，主要是用于表示二进制数。因为十六进制数与二进制数的相互转换十分方便，而且一个二进制数用相应的十六进制数表示时，十分简短紧凑，所以在数字电子系统中人们经常使用十六进制数。

1. 二进制数转换成十六进制数 分析可知，每 4 位二进制数正好对应 1 位十六进制数。因此，将二进制数转换成十六进制数的方法十分简单：把二进制数从其 LSB 到 MSB 分组，每 4 位一组。然后将每一组二进制数转换成为一位相应的十六进制数即可。例如

$$(1011100110)_B = (10,\ 1110,\ 0110)_B = (2E6)_H = 2E6H$$

2. 十六进制数转换成二进制数 将十六进制数转换成二进制数的方法也很容易：将每 1 位十六进制数用 4 位相应的二进制数表示、代入即可。例如

$$234FH = (0010,\ 0011,\ 0100,\ 1111)_B = (10001101001111)_B$$

1.2.5 二进制码

将若干个 0 和 1 按一定的规律编排在一起，编成不同的代码，并且赋予每个代码以固定的含义，这种过程称做**编码**。

常用的二进制编码有 BCD 码、ASCII 码等。

1. BCD 码 十进制数是人类最熟悉、使用最广泛的数，在数字电路中也经常需要表示和处理十进制数。为此就要用二进制数来表示十进制数，这实际上是对十进制数的编码，即所谓 **BCD 码**。

BCD (Binary Coded Decimal) 意为“由二进制数编码的十进制数码”，也即 BCD 码是用二进制码代表的十进制数码。

因为十进制数共有 10 个基本符号，所以 BCD 码由 4 位二进制数组成。4 位二进制数有 16 个不同的值。故可以从 16 个 4 位二进制数中选取 10 个来对十进制数进行编码。而当选取的方法不同时，就可以得到不同的 BCD 码，如 8421 码、2421 码、5421 码、余 3 码和格雷码等，如表 1-1 所示。

表 1-1 不同的 BCD 码

十进制数	8421 码	2421 码	5421 码	余 3 码	格雷码
0	0000	0000	0000	0011	0000
1	0001	0001	0001	0100	0001
2	0010	0010	0010	0101	0011
3	0011	0011	0011	0110	0010
4	0100	0100	0100	0111	0110
5	0101	1011	1000	1000	0111
6	0110	1100	1001	1001	0101
7	0111	1101	1010	1010	0100
8	1000	1110	1011	1011	1100
9	1001	1111	1100	1100	1101

（1）8421 码（自然 BCD 码）　如表 1-1 所示，8421 码是从 16 个 4 位二进制数中选取前 10 个数码的编码方式。将 8421 码转换成相应的十进制数的方法，与将普通二进制数转换成为相应十进制数的方法相同。而之所以称之为 8421 码，是因为它是一种**有权码**（恒权码）。其权值分别为 8、4、2、1。

（2）余 3 码　从表 1-1 可见，余 3 码可由 8421 码与 3 的二进制数，即 0011 相加而得到。故称为余 3 码。分析可知，余 3 码是一种**无权码**。因为余 3 码相应的十进制数无法用固定的权值计算出来。

（3）格雷码　格雷码也是一种常用的无权码。它的特点是相邻的两个码字之间，仅有一位相异。这样更加易于对数据的纠错。

2. ASCII 码　ASCII（American Standard Code for Information Interchange）码是一种数字系统中常用的、表示字符的二进制编码。ASCII 码用 7 位二进制数来表示 PC 键盘上的所有按键信息，以及打印机等设备的控制信息。

常用字符的 7 位 ASCII 码值如表 1-2 所示。

表 1-2　常用字符的 7 位 ASCII 码值（用十六进制数表示）

字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII
NUL	00	4	34	M	4D	f	66
BEL	07	5	35	N	4E	g	67
LF	0A	6	36	O	4F	h	68
FF	0C	7	37	P	50	i	69
CR	0D	8	38	Q	51	j	6A
SP	20	9	39	R	52	k	6B
!	21	:	3A	S	53	l	6C
"	22	;	3B	T	54	m	6D
#	23	<	3C	U	55	n	6E
$	24	=	3D	V	56	o	6F
%	25	>	3E	W	57	p	70
&	26	?	3F	X	58	q	71
'	27	@	40	Y	59	r	72
(	28	A	41	Z	5A	s	73
)	29	B	42	[	5B	t	74
*	2A	C	43	\	5C	u	75
+	2B	D	44	]	5D	v	76
,	2C	E	45	∧	5E	w	77
–	2D	F	46	—	5F	x	78
.	2E	G	47	`	60	y	79
/	2F	H	48	a	61	z	7A
0	30	I	49	b	62	{	7B
1	31	J	4A	c	63	|	7C
2	32	K	4B	d	64	}	7D
3	33	L	4C	e	65	~	7E

注：NUL 表示空，BEL 表示报警，LF 表示换行，FF 为走纸控制，CR 为回车，SP 为 Space 键。

3. 二进制数补码 二进制数的补码表示采用的是**正负符号 + 绝对值**表示法。此时补码的最高位为符号位，正数的最高位为 0，负数的最高位为 1。

（1）二进制数的反码 将二进制数的各位数码取反，就可以得到该二进制数的**反码**。所谓取反，就是将 1 变成 0，将 0 变成 1。应当注意：求一个二进制数的反码时，必须按照一定的字长进行，如 8 位字长或 16 位字长等。

例如 8 位字长时，$(11001010)_B$ 的反码为 $(00110101)_B$。

（2）二进制数的补码

1）正数的补码：为其原来的数码。

2）负数的补码：将该负数对应的二进制正数的反码加 1 后，就可以得到该二进制负数的**补码**。

例如，求 -117 的补码时，因为 $(117)_D = (01110101)_B$，$(01110101)_B$ 的反码为 $(10001010)_B$，加 1 后得到 $(-117)_D$ 的补码为 $[-117]_补 = (10001011)_B = 8BH$。

（3）利用二进制数的补码实现减法运算 利用补码能够将二进制数的减法运算转换成加法运算。有公式如下：

$$[X-Y]_补 = [X]_补 + [-Y]_补 \tag{1-4}$$

例如 $[42-11]_补 = [42]_补 + [-11]_补 = (00101010)_B + (11110101)_B = (100011111)_B$（最高位舍掉）。所以，$42-11 = [(00011111)_B]_补 = 31$。

又如，$[11-42]_补 = [11]_补 + [-42]_补 = (00001011)_B + (11010110)_B = (11100001)_B$，即 $[11-42] = -[(11100001)_B]_补$。所以，$11-42=-31$。

1.3 逻辑关系与逻辑门电路

1.3.1 基本逻辑关系

自然界中许多事物之间存在着一定的逻辑关系。其中“与”、“或”和“非”是 3 种基本的逻辑关系。

1. 逻辑“与”关系 逻辑“与”关系是指事物之间的这样一种逻辑关系：设有 3 个事件，A、B 和 C。事件 C 的发生与否，是与 A、B 的发生与否有关的。其关系为：当 A、B 同时发生时，C 才会发生，就称事件 C 的发生与否和事件 A、B 之间存在着“与”逻辑关系。记为 $C = A \cdot B$。

还可以用记号 0 和 1 来标记事件的两种状态。一般用 1 表示事件的发生，用 0 表示事件的不发生。因此对于逻辑“与”关系就有

$$0 \cdot 0 = 0 \tag{1-5}$$

$$0 \cdot 1 = 0 \tag{1-6}$$

$$1 \cdot 0 = 0 \tag{1-7}$$

$$1 \cdot 1 = 1 \tag{1-8}$$

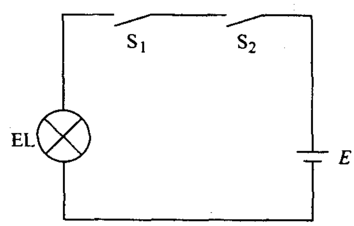


图 1-5 “与”逻辑关系

图 1-5 所示的“与”逻辑关系的电路中，开关 S_1、S_2 为串联连接，灯泡 EL 的发光与否就和 S_1、S_2 的闭合与否之间为逻辑“与”的关系。只有当 S_1、S_2 同时闭合时，

EL 才会发光。否则无论是 S_1、S_2 中哪一个单独闭合，或者两者全都断开，EL 都是不会亮的。

2. 逻辑“或”关系 逻辑“或”关系是指事物之间的这样一种逻辑关系：设有 3 个事件，A、B 和 C。事件 C 的发生与否，是与 A、B 的发生与否有关的。其关系为：当 A、B 中任何一个事件发生，都会导致 C 的发生，就称事件 C 的发生与否和 A、B 之间存在逻辑“或”的关系。记为 $C = A + B$。

对于逻辑“或”关系有

$$0 + 0 = 0 \tag{1-9}$$

$$0 + 1 = 1 \tag{1-10}$$

$$1 + 0 = 1 \tag{1-11}$$

$$1 + 1 = 1 \tag{1-12}$$

应当注意，这里的“+”号和数值 0、1 都是表示逻辑关系的记号，不能与普通的算术运算相混淆。

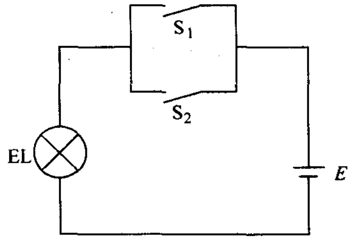


图 1-6 “或”逻辑关系

图 1-6 所示的“或”逻辑关系电路中，开关 S_1、S_2 为并联连接，灯泡 EL 的发光与否就和 S_1、S_2 的闭合与否之间为逻辑“或”的关系。无论是 S_1、S_2 中哪一个单独闭合，或者两者全都闭合，EL 都会发光。只有当 S_1、S_2 全都断开时，EL 才会不亮。

3. 逻辑非关系 逻辑“非”关系是指事物之间的这样一种逻辑关系：设有两个事件，A 和 C。事件 C 的发生与否，是与 A 的发生与否有关的。其关系为：当 A 事件发生时，C 一定不会发生；而当 A 事件不发生时，C 一定会发生，就称事件 C 的发生与否和 A 之间存在逻辑“非”的关系。记为 $C = \overline{A}$。

对于逻辑“非”关系有

$$\overline{0} = 1 \tag{1-13}$$

$$\overline{1} = 0 \tag{1-14}$$

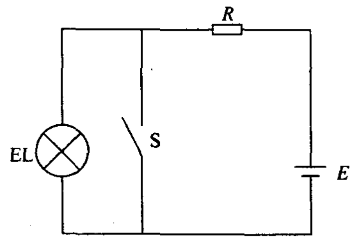


图 1-7 “非”逻辑关系

图 1-7 所示的“非”逻辑关系电路中，开关 S 与灯泡 EL 并联；灯泡 EL 的发光与否就和 S 的闭合与否之间为逻辑“非”的关系。当 S 闭合时，EL 被短路，不会发光；当 S 断开时，电流正常流过 EL，灯泡发光。

4. 复合的逻辑关系 将基本的“与”、“或”和“非”等逻辑关系复合起来，就能够得到各种更为复杂的逻辑关系。如“与非”、“或非”、“异或”和“同或”等。

在图 1-8 所示“与非”逻辑关系电路中，灯泡 EL 的发光与否，与开关 S_1、S_2 的闭合与否之间的逻辑关系为“与非”关系：只有 S_1、S_2 全都闭合时，EL 才会不发光；除此之外的其他情况下，EL 都将发光。

“与非”逻辑关系，表示为 $C = \overline{A \cdot B}$。

在图 1-9 所示“或非”逻辑关系电路中，灯泡 EL 的发光与否，与开关 S_1、S_2 的闭合与否之间的逻辑关系为“或非”关系：只要 S_1、S_2 中有一个闭合，EL 就会

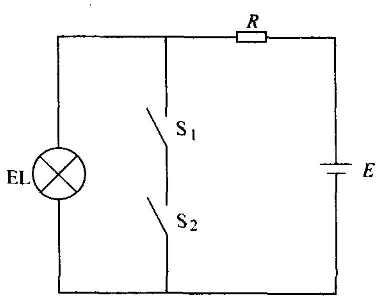


图 1-8 “与非”逻辑关系