



高职高专“十一五”规划教材

机械电子类

数字电子技术

赵利 郑英兰 主编



冶金工业出版社

www.cnmp.com.cn

高职高专“十一五”规划教材·机械电子类

数字电子技术

主 编 赵 利 郑英兰

副主编 裴咏枝 邓玉良 宋风忠

北 京
冶 金 工 业 出 版 社
2009

内 容 简 介

本书以理论够用、实用为主,注重实践的教学思想而编写。着重介绍数字电路的新理论、新技术以及新器件,对数字电路的常用集成电路作了详细的介绍。本书在编写过程中,力求简明扼要、通俗易懂。书中给出了大量的例题和习题,便于读者自学。

本书内容主要包括:数字逻辑基础、集成逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲信号的产生与整形、数/模和模/数转换器、半导体存储器和可编程逻辑器件等。

本书可作为应用型人才培养的本科以及高职高专院校的电子技术、通信、计算机和电气自动化等专业的教材,也可供从事电子技术工程的人员参考使用。

图书在版编目(CIP)数据

数字电子技术/赵利,郑英兰主编. —北京:冶金工业出版社,
2009.1
ISBN 978-7-5024-4833-2

I. 数… II. ①赵…②郑… III. 数字电路—电子技术 IV. TN79

中国版本图书馆 CIP 数据核字(2009)第 004652 号

出 版 人 曹胜利

地 址 北京北河沿大街嵩祝院北巷 39 号,邮编 100009

电 话 (010)64027926 电子信箱 postmaster@cnmip.com.cn

责任编辑 刘 源

ISBN 978-7-5024-4833-2

北京天正元印务有限公司印刷;冶金工业出版社发行;各地新华书店经销

2009 年 1 月第 1 版,2009 年 1 月第 1 次印刷

787mm×1092mm 1/16; 15 印张; 353 千字; 232 页; 1-3000 册

29.00 元

(本书如有印装质量问题,本社发行部负责退换)

前 言

本书是作者根据国家教育部最新制订的《高职高专教育数字电子技术基础课程教学基本要求》，并在多年教学实践的基础上编写而成的。本书可作为应用型人才培养的本科以及高职高专院校的电子技术、通信、计算机和电气自动化等专业的教材，也可供从事电子技术工程的人员参考使用。

数字电子技术是理工科专业一门重要的专业基础课。本书着重介绍数字电子电路的基本理论以及电路的分析与设计方法。在内容的节选上注重先进性和实用性，突出了基本理论、基本概念和基本分析方法，回避了烦琐的集成电路内部的分析和数学推导。在章节的编排上力求兼顾国内一些通用教材的体系。为便于读者对所讲述内容的理解和消化，在介绍每章内容时尽量讲清思路和分析方法，以启发学生的思维，力图加强基本理论、基本知识和基本技能的培养。

全书共 8 章。第 1 章介绍了分析和设计数字电路的工具——数字逻辑基础。第 2 章介绍了数字电路最基本的单元——集成逻辑门电路，对集成逻辑门电路的原理和电气特性作了较为详细的介绍。第 3 章介绍了组合逻辑电路，重点介绍了几种常见的、典型的组合电路。第 4 章和第 5 章是本书的重点，分别介绍了集成触发器和时序逻辑电路。触发器是时序逻辑电路的基本单元，从触发器的结构、动作特性以及不同逻辑功能的触发器之间的转换等角度对它们作了较为详细的介绍。时序逻辑电路是数字电路中最重要内容，书中对寄存器、计数器以及顺序脉冲发生器等典型电路作了详尽的描述。第 6 章介绍了脉冲信号的产生与整形，主要介绍了 3 种与脉冲电路相关的典型电路：施密特触发器、多谐振荡器和单稳态触发器。第 7 章介绍了数/模和模/数转换器，对 A/D、D/A 转换的多种方法以及相关的集成电路作了介绍。第 8 章介绍了半导体存储器和可编程逻辑器件，重点介绍了只读存储器 ROM 和随机存储器 RAM 的结构、原理和应用。可编程逻辑器件是一种新型器件，本书对 PLA、PAL 和 GAL 等典型 PLD 器件的原理和应用也作了详细的介绍。

本书除第 8 章部分地方采用国际通用符号外，其余章节全部采用了国家标准 GB/T4728《电气简图用图形符号》所规定的逻辑符号。

本书由赵利、郑英兰任主编，裴咏枝、邓玉良、宋风忠任副主编，杨健、罗泽林参加编写。

由于编者水平所限，书中如有不足之处敬请读者批评指正，以便修订时改进。如读者在使用本书的过程中有其他意见或建议，恳请向编者(bjzhangxf@126.com)踊跃提出宝贵意见。

编 者

目 录

第 1 章 数字逻辑基础..... 1	习题..... 26
1.1 概述..... 1	第 2 章 集成逻辑门电路..... 29
1.1.1 数字信号与模拟信号..... 1	2.1 概述..... 29
1.1.2 数字电路的特点及应用..... 1	2.2 基本逻辑门电路..... 29
1.2 数制和码制..... 2	2.2.1 二极管的开关特性..... 29
1.2.1 数制..... 2	2.2.2 三极管的开关特性..... 31
1.2.2 不同数制间的转换..... 2	2.2.3 MOS 管的开关特性..... 34
1.2.3 二十进制码..... 5	2.2.4 分立元件门电路..... 35
1.3 逻辑代数中的常用运算..... 6	2.2.5 组合逻辑门电路..... 38
1.3.1 算术运算..... 6	2.3 TTL 集成逻辑门电路..... 39
1.3.2 基本逻辑运算..... 6	2.3.1 TTL 与非门..... 39
1.3.3 复合逻辑运算..... 8	2.3.2 TTL 与非门的电气特性..... 41
1.4 逻辑代数的基本公式及定律..... 10	2.3.3 其他功能的 TTL 门电路..... 45
1.4.1 逻辑代数的公式..... 11	2.3.4 TTL 集成逻辑门电路 系列简介..... 48
1.4.2 逻辑代数的定律..... 11	2.3.5 TTL 集成逻辑门电路的 使用注意事项..... 49
1.5 逻辑函数的表达方法..... 13	2.4 CMOS 集成逻辑门电路..... 50
1.5.1 逻辑函数的建立..... 13	2.4.1 CMOS 反相器..... 51
1.5.2 逻辑函数的表示方法..... 14	2.4.2 其他功能的 CMOS 门电路..... 52
1.5.3 各种表示方法之间的 互相转换..... 15	2.4.3 CMOS 数字集成电路的 系列..... 55
1.6 逻辑函数的代数化简法..... 17	2.4.4 CMOS 集成逻辑门电路的 使用注意事项..... 56
1.6.1 最简的概念..... 17	2.5 TTL 电路和 CMOS 电路的接口..... 57
1.6.2 化简方法..... 17	2.5.1 TTL 电路驱动 CMOS4000 系列电路..... 57
1.7 逻辑函数的卡诺图化简法..... 19	2.5.2 CMOS 门驱动 TTL 门..... 58
1.7.1 化简的基本原理..... 19	
1.7.2 化简的方法及步骤..... 22	
1.7.3 具有无关项的逻辑函数的 化简..... 24	
本章小结..... 25	

2.5.3 TTL 和 CMOS 电路带负载时的接口问题	58	3.7.5 集成加法器的应用	93
本章小结	59	3.8 组合逻辑电路中的竞争冒险	95
习题	60	3.8.1 产生竞争冒险的原因	95
第 3 章 组合逻辑电路	64	3.8.2 冒险现象的识别	96
3.1 概述	64	3.8.3 冒险现象的消除方法	96
3.2 组合逻辑电路的分析和设计方法	65	本章小结	97
3.2.1 组合逻辑电路的分析方法	65	习题	97
3.2.2 组合逻辑电路的设计方法	66	第 4 章 集成触发器	101
3.3 编码器	70	4.1 概述	101
3.3.1 二进制编码器	70	4.2 基本 RS 触发器	101
3.3.2 二十进制编码器	71	4.2.1 由与非门组成的基本 RS 触发器	101
3.3.3 优先编码器	72	4.2.2 由或非门组成的基本 RS 触发器	102
3.3.4 编码器的应用	73	4.2.3 触发器功能的描述	103
3.4 译码器	74	4.2.4 基本 RS 触发器的动作特点	105
3.4.1 二进制译码器	75	4.2.5 集成 RS 触发器	105
3.4.2 二十进制译码器	76	4.3 同步触发器	106
3.4.3 数字显示译码器	77	4.3.1 时钟同步 RS 触发器	106
3.4.4 译码器的应用	80	4.3.2 时钟同步 JK 触发器	109
3.5 数据选择器	82	4.4 主从触发器	112
3.5.1 数据选择器的基本概念及工作原理	82	4.4.1 主从 RS 触发器	112
3.5.2 集成数据选择器	84	4.4.2 主从 JK 触发器	113
3.5.3 数据选择器的应用	85	4.4.3 主从触发器的动作特点	114
3.6 数值比较器	87	4.5 边沿触发器	114
3.6.1 1 位数值比较器	87	4.5.1 边沿 JK 触发器	115
3.6.2 多位数值比较器	87	4.5.2 维持阻塞 D 触发器	117
3.6.3 集成数值比较器	88	4.6 CMOS 边沿触发器	118
3.7 加法器	89	4.6.1 CMOS 边沿 D 触发器	119
3.7.1 半加器	89	4.6.2 CMOS 边沿 JK 触发器	120
3.7.2 全加器	90	4.7 其他类型触发器以及不同类型触发器之间的转换	120
3.7.3 多位数加法器	91		
3.7.4 快速进位集成 4 位加法器	92		

4.7.1	T 触发器	121	6.1.1	脉冲信号	170
4.7.2	T'触发器	121	6.1.2	主要参数	170
4.7.3	不同类型触发器之间的 转换	121	6.1.3	脉冲波形的产生	171
4.8	触发器的应用举例	123	6.2	集成 555 定时器及其应用举例	171
4.8.1	由触发器构成单脉冲发 生器	123	6.2.1	集成 555 定时器的组成和 工作原理	171
4.8.2	分频电路	123	6.2.2	集成 555 定时器应用 举例	173
4.8.3	多路控制开关电路	124	6.3	单稳态触发器	174
4.8.4	4 路抢答器	125	6.3.1	CMOS 门构成的单稳态 触发器	174
本章小结		126	6.3.2	555 定时器构成的单稳态 触发器	176
习题		127	6.3.3	集成单稳态触发器的应用	177
第 5 章	时序逻辑电路	132	6.4	多谐振荡器	180
5.1	概述	132	6.4.1	由门电路构成的多谐 振荡器	180
5.2	时序逻辑电路的分析	133	6.4.2	555 定时器构成的多谐 振荡器	183
5.2.1	时序逻辑电路的基本 分析步骤	133	6.4.3	多谐振荡器的应用	184
5.2.2	分析举例	134	6.5	施密特触发器	185
5.3	寄存器和移位寄存器	138	6.5.1	用门电路构成的施密特 触发器	185
5.3.1	数码寄存器	138	6.5.2	由 555 定时器构成的施密特 触发器	186
5.3.2	移位寄存器	139	6.5.3	集成施密特触发器及其 应用	188
5.3.3	寄存器的应用	141	本章小结		189
5.4	计数器	143	习题		189
5.4.1	二进制计数器	144	第 7 章	数/模和模/数转换器	192
5.4.2	十进制计数器	149	7.1	概述	192
5.4.3	集成计数器	152	7.2	数/模转换器	192
5.4.4	利用计数器的级联获得大 容量 N 进制计数器	162	7.2.1	权电阻网络 D/A 转换器	192
本章小结		164			
习题		165			
第 6 章	脉冲信号的产生与整形	170			
6.1	概述	170			

7.2.2	R-2R T 型电阻网络 D/A 转换器	194
7.2.3	R-2R 倒 T 型电阻网络 D/A 转换器	195
7.2.4	D/A 转换器的主要技术 指标	197
7.2.5	集成 D/A 转换器	198
7.3	模/数转换器	199
7.3.1	A/D 转换的一般步骤	200
7.3.2	并联比较型 A/D 转换器	201
7.3.3	逐次逼近型 A/D 转换器	202
7.3.4	双积分型 A/D 转换器	205
7.3.5	A/D 转换器的主要技术 指标	208
	本章小结	208
	习题	209

第 8 章	半导体存储器和可编程 逻辑器件	210
8.1	半导体存储器	210
8.1.1	概述	210
8.1.2	只读存储器	210
8.1.3	随机存取存储器	215
8.2	可编程逻辑器件	220
8.2.1	概述	220
8.2.2	可编程逻辑器件的基本 结构	220
8.2.3	可编程阵列逻辑 PAL	224
8.2.4	通用阵列逻辑 GAL	225
	本章小结	230
	习题	231
	参考文献	232

第 1 章 数字逻辑基础

内容提要

本章介绍了逻辑代数的基本概念，常用的数制和码制，逻辑代数的基本公式和主要定律，以及逻辑函数的表示方法及各种化简方法。

1.1 概 述

1.1.1 数字信号与模拟信号

自然界中的物理量种类繁多，性质各异，分类方法也很多，如按照物理量变化规律的特点分类，可以分为两大类。

一类是模拟量，它的变化在时间上和数值(幅度)上都是连续的，如电压量和温度值等。表示模拟量的信号叫模拟信号，工作在模拟信号下的电路称作模拟电路。

另一类就是数字量，它的变化在时间上和数值上都是离散的，或说其变化是发生在一系列离散的瞬间，如产品的数目和运动员的号码等。表示数字量的信号叫数字信号，工作在数字信号下的电路称作数字电路。

1.1.2 数字电路的特点及应用

相比于模拟电路，数字电路具有以下特点：

(1) 在工艺上更容易做出高密度集成电路。在数字电路中，其基本单元电路的结构简单，电路参数可以有较大的离散性，这样较容易通过半导体制造工艺将数目庞大的基本单元电路做在一块硅片上，构成高密度的集成电路，便于集成、系列化生产，成本低廉，具有较高的性价比。

(2) 工作准确可靠、精度高、抗干扰能力强。数字电路采用的是二进值代码，工作时只需要判断高低电平、信号的有无，电路实现简单，同时也提高了工作的可靠性。数字电路信号比较强，噪声干扰不容易侵入，且数字抗干扰技术比较容易实现，故而数字电路的抗干扰能力较强。

(3) 数字信息保存方便、保存期长、保密性好。随着计算机技术的发展，各种数字存储器件和设备种类较多，存储容量大，性能稳定，为保存数字信息提供了有力的保障。数字信息的加密处理方便、可靠，不容易丢失和被窃。

(4) 数字电路产品系列多，品种齐全，通用性和兼容性都比较好，使用方便。

数字电路不仅能完成数值运算，还可以进行逻辑运算与判断，在控制系统中是不可或缺的。数字电路在电子计算机、电机、通信设备、自动控制、雷达、家用电器、日常电子小产品以及汽车电子等领域得到了广泛的应用。

1.2 数制和码制

1.2.1 数制

数制就是计数的方法，具体地说，就是把多位数码中每一位的构成方法和进位规则称为数制。在数字电路中，按照具体需要确定采用何种计数方法。

任意 N 进制数均可表示为 $D = \sum k_i \times N^i$ ，其中 k 为第 i 位的系数， N 为基数， N^i 为位权。进位规则为“逢 N 进 1”，使用时，在数字的右下角标注 N 以表示相应的进制。常见的数制有十进制、二进制、八进制和十六进制等。

1.2.1.1 十进制

十进制是我们最熟悉的计数进位制，也是日常生活和工作中应用最广泛的进制。用 0、1、2、3、4、5、6、7、8、9 十个数码代表一位十进制数的十个不同状态，基数是 10，进位规则为“逢十进一”。

例如，十进制数 386 可写为： $(386)_{10} = 3 \times 10^2 + 8 \times 10^1 + 6 \times 10^0$ 。

1.2.1.2 二进制

二进制数的每位只有 0 和 1 两个数码，基数为 2，进位规则为“逢二进一”。二进制是数字电路中最基本的数制。

例如，二进制数 1001 可写为： $(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ 。

1.2.1.3 八进制

八进制数的每位有 0、1、2、3、4、5、6、7 八个数码，基数是 8，进位规则是“逢八进一”。

例如，八进制数 624 可写为： $(624)_8 = 6 \times 8^2 + 2 \times 8^1 + 4 \times 8^0$ 。

1.2.1.4 十六进制

十六进制数的每位有 0、1、2、3、4、5、6、7、8、9 以及 A(10)、B(11)、C(12)、D(13)、E(14)和 F(15)十六个数码，基数是 16，进位规则为“逢十六进一”。十六进制是目前计算机中广泛采用的数制，计算机程序大多采用十六进制符号书写。

例如，十六进制数 1A28 可写为： $(1A28)_{16} = 1 \times 16^3 + 10 \times 16^2 + 2 \times 16^1 + 8 \times 16^0$ 。

用上述同样的方法也可以表示小数，唯一的区别是小数点右边的各位数码要乘以基数的负的幂次。

例如，二进制数 10.0011 表示为： $(10.0011)_2 = 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$ 。

十进制数 2.39 可写为： $(2.39)_{10} = 2 \times 10^0 + 3 \times 10^{-1} + 9 \times 10^{-2}$ 。

1.2.2 不同数制间的转换

二进制数在计算机中广泛使用，但它相比于十进制数和十六进制数等，存在位数太多

和不便读写等缺点,也就是说,各种数制在不同的应用场合各有千秋,因此我们需要掌握不同数制间的转换,以方便工作。不同数制间的转换方法类似,都是按照 $D = \sum k_i \times N^i$ 展开,然后求其各位加权系数之和,即可得到相应的进制数。

1.2.2.1 二进制数转换为十进制数

将二进制数转换为等值的十进制数称为二-十进制转换。其转换方法是将被二进制数按 $D = \sum k_i \times 2^i$ 展开,然后求其各位加权系数之和,就可以得到相应的十进制数。

例如,将二进制数 1101、10.1011 转换成相对应的十进制数分别为:

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (13)_{10}$$

$$(10.1011)_2 = 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} = (2.6875)_{10}$$

1.2.2.2 十六进制数转换为十进制数

将十六进制数转换为等值的十进制数称为十六-十进制转换。其转换方法是将被十六进制数按 $D = \sum k_i \times 16^i$ 展开,然后求其各位加权系数之和,就可以得到相应的十进制数。

例如,将十六进制数 2A08 转换成相对应的十进制数为:

$$(2A08)_{16} = 2 \times 16^3 + 10 \times 16^2 + 0 \times 16^1 + 8 \times 16^0 = (10760)_{10}$$

1.2.2.3 十进制数转换为二进制数

将十进制数转换为等值的二进制数称为十-二进制转换。其转换方法是采用“除2取余”法。下面通过具体的例子说明十-二进制转换步骤。

【例 1-1】 将十进制数 41 转换成二进制数。

解: 第一步,把给定的十进制数 41 除以 2,得商为 20,余数为 1,取出余数 1 作为二进制数最低位数的数码 k_0 。

第二步,将前步所得商 20 再除以 2,得商为 10,余数为 0,取出余数 0 作为二进制数次低位数的数码 k_1 。

第三步,重复第二步的工作,直到商为 0 为止,最后一次得到的余数即为二进制数最高位数的数码 k_n 。

具体过程如下:

2	41		余数为1,即 $k_0=1$
2	20		余数为0,即 $k_1=0$
2	10		余数为0,即 $k_2=0$
2	5		余数为1,即 $k_3=1$
2	2		余数为0,即 $k_4=0$
2	1		余数为1,即 $k_5=1$
	0		

可得转换结果为: $(41)_{10} = (101001)_2$ 。

1.2.2.4 二进制数与八进制数之间的相互转换

将二进制数转换为等值的八进制数称为二-八进制转换。因为 3 位二进制数恰好有 8 个状态，所以二-八转换可以采用“3 位 1 组”法。即从低位到高位依次将每 3 位二进制数划分为 1 组，高位不足 3 位的前面加以 0 补足 3 位，然后将每 1 组用其等值的八进制数表示，就得到相应的八进制数。

将八进制数转换为等值的二进制数称为八-二进制转换。其转换方法和二-八进制转换相反，只要将八进制数的每 1 位分别用 3 位二进制数表示即可。

【例 1-2】 将二进制数 1101010011 转换成对应的八进制数。

解：第一步，将给定的二进制数 1101010011 从低位到高位依次把每 3 位划分为 1 组，即：

001, 101, 010, 011

第二步，将第一步所得各组数用其对应的八进制数表示，结果就是二进制数转换成的八进制数，即：

二进制数	001,	101,	010,	011
	↓	↓	↓	↓
八进制数	1,	5,	2,	3

可得转换结果为： $(1101010011)_2 = (1523)_8$ 。

【例 1-3】 将八进制数 7125 转换成对应的二进制数。

解：把给定的八进制数 7125 的每 1 位用 3 位二进制数表示，即：

八进制数	7,	1,	2,	5
	↓	↓	↓	↓
二进制数	111,	001,	010,	101

可得转换结果为： $(7125)_8 = (111001010101)_2$ 。

1.2.2.5 二进制数与十六进制数之间的相互转换

将二进制数转换为等值的十六进制数称为二-十六进制转换。二进制数和十六进制数的相互转换在计算机中应用非常广泛。其转换方法同二-八进制转换类似。唯一的区别是变“3 位 1 组”为“4 位 1 组”。

【例 1-4】 将二进制数 1101011011 转换成对应的十六进制数。

解：第一步，将给定的二进制数 1101011011 从低位到高位依次把每 4 位二进制数划分为 1 组，即：

0011, 0101, 1011

第二步，将第一步所得各组数用对应的十六进制数表示，结果就是二进制数转换成的

十六进制数，即：

二进制数	0011, 0101, 1011
↓	↓ ↓
十六进制数	3, 5, B

可得转换结果为： $(1101011011)_2 = (35B)_{16}$ 。

【例 1-5】 将十六进制数 2C95 转换成对应的二进制数。

解：把给定的十六进制数 2C95 的每 1 位改用 4 位二进制数表示，即：

十六进制数	2, C, 9, 5
↓	↓ ↓ ↓ ↓
二进制数	0010, 1100, 1001, 0101

可得转换结果为： $(2C95)_{16} = (10110010010101)_2$ 。

1.2.3 二-十进制码

表示不同事物的代号称为代码，编制代码的规则称为码制。常用的是二-十进制码，取其英文字头，通常被称作 BCD 码，即用 4 位二进制数来表示一位十进制数。在二-十进制编码中，一般分为有权码和无权码。4 位二进制数码有多种不同的组合方式，因此选取方式不同，可得到不同的编码方式。其中，广泛使用的是 8421BCD 码，它是一种有权码，其位权从高位到低位依次为： $8(2^3)$ 、 $4(2^2)$ 、 $2(2^1)$ 、 $1(2^0)$ 。无权码中常见的是余 3 码、余 3 循环码和格雷码等。当十进制数用余 3 码表示时，要比 8421 码在二进制数值上多 3，故称余 3 码，它可由 8421 码加 0011 得到。几种常见的 BCD 编码见表 1-1。

表 1-1 几种常见的 BCD 编码

十进制数	有权码				无权码		
	8421 码	5421 码	5211 码	2421 码	余 3 码	余 3 循环码	格雷码
0	0000	0000	0000	0000	0011	0010	0000
1	0001	0001	0001	0001	0100	0110	0001
2	0010	0010	0100	0010	0101	0111	0011
3	0011	0011	0101	0011	0110	0101	0010
4	0100	0100	0111	0100	0111	0100	0110
5	0101	1000	1000	1011	1000	1100	0111
6	0110	1001	1001	1100	1001	1101	0101
7	0111	1010	1100	1101	1010	1111	0100
8	1000	1011	1101	1110	1011	1110	1100
9	1001	1100	1111	1111	1100	1010	1000

例如，将十进制数 935 转换为 8421BCD 码可写为： $(935)_{10} = (100100110101)_{8421BCD}$ 。

1.3 逻辑代数中的常用运算

1.3.1 算术运算

数字逻辑电路中,二进制数码 0 和 1 可以表示数量的大小。这个时候,它就可以进行数值运算,即算术运算。其运算规则与十进制数的运算规则类似,只要注意“逢二进一”即可。

例如,对两个二进制数 1010 和 0011 进行加法和减法运算。

加法运算为

$$\begin{array}{r} 1010 \\ + 0011 \\ \hline 1101 \end{array}$$

减法运算为

$$\begin{array}{r} 1010 \\ - 0011 \\ \hline 0111 \end{array}$$

$$\text{即 } (1010)_2 + (0011)_2 = (1101)_2, \quad (1010)_2 - (0011)_2 = (0111)_2。$$

1.3.2 基本逻辑运算

自然界中的许多现象都存在着对立的双方,如电位的高、低,灯泡的亮、灭等,它们都可以采用逻辑变量来表示。在数字逻辑电路中,二进制数码 0 和 1 不仅可以表示数量的大小,而且可以表示两种对立的逻辑状态,这种逻辑关系称为二值逻辑,此时进行的运算称为逻辑运算。

逻辑运算和算术运算是完全不同的两种运算,二值逻辑之间需要按照某种指定的因果关系进行所谓的逻辑运算。

数字电路中,基本的逻辑关系有与逻辑、或逻辑和非逻辑。通常用输入信号来表示“条件”,用输出信号来表示“结果”,输入和输出之间的因果关系即为逻辑关系。

逻辑代数中最基本的 3 种逻辑运算是与运算(简称与)、或运算(简称或)和非运算(简称非)。实现它们的电路分别称作与门、或门和非门。其他运算均是在这 3 种基本运算基础上演变而来的。逻辑电路中,如果各输入变量(常用 A 、 B 、 C ……表示)的取值确定后,则输出变量(常用 Y 表示)的值也就被唯一确定了。习惯上称 Y 是 A 、 B 、 C ……的逻辑函数。下面通过一些简单的例子来解释与、或和非 3 种基本运算的含义。

1.3.2.1 与逻辑运算

如图 1-1 所示电路,灯 Y 亮(结果发生)的要求是开关 A 、 B (条件)必须都闭合,那么灯与开关的关系就是与逻辑关系。如开关断开和灯灭用 0 表示,开关闭合和灯亮用 1 表示,则这种逻辑关系见表 1-2,这种图表叫作逻辑真值表,或简称为真值表。

这个例子表明,当决定某一事件的所有条件都同时具备时,事件的结果才会发生,这种因果关系就称为与逻辑,用“.”表示。由表 1-2 可知,在输入 A 、 B 中,任一为 0 时, Y 为 0;只有当 A 、 B 都为 1 时,

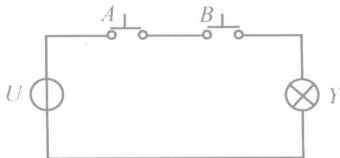


图 1-1 与逻辑的电路图

Y 才为1。 Y 与 A 、 B 间的关系可用 $Y = A \cdot B$ 或 $Y = AB$ 表示。对于多变量的与逻辑关系可以表示成 $Y = A \cdot B \cdot C \cdots$ 或 $Y = ABC \cdots$ 。实现与运算的电路为与门,其逻辑符号如图1-2所示。

表 1-2 与逻辑真值表

输入		输出
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



图 1-2 与门逻辑符号

1.3.2.2 或逻辑运算

如图1-3所示电路,当开关 A 、 B (条件)中有任何一个闭合时,灯 Y 就会亮(结果发生),那么灯与开关的关系就是或逻辑关系。如变量 A 、 B 、 Y 给出和与逻辑同样的赋值时,则这种逻辑关系见表1-3。

这个例子表明,当决定某一事件的所有条件中,只要有一个条件具备了,事件结果就会发生,这种因果关系称为或逻辑,用“+”表示。由表1-3可知,在输入 A 、 B 中,任一或全部为1时, Y 为1;只有当 A 、 B 都为0时, Y 才为0。

Y 与 A 、 B 间的关系可用 $Y = A + B$ 表示。对于多变量的或逻辑关系可以表示成 $Y = A + B + C + \cdots$ 。实现或运算的电路为或门,其逻辑符号如图1-4所示。

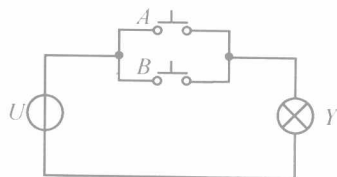


图 1-3 或逻辑的电路图

表 1-3 或逻辑真值表

输入		输出
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

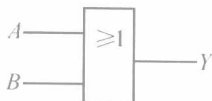


图 1-4 或门逻辑符号

1.3.2.3 非逻辑运算

如图1-5所示电路,当开关 A (条件)动作时,灯 Y 不亮(结果不发生),当开关 A (条件)不动作时,灯 Y 就会亮(结果发生),那么灯与开关的关系就是非逻辑关系。如变量 A 、 Y 给出和与逻辑同样的赋值时,则这种逻辑关系见表1-4。

这个例子表明，当决定某一事件的条件具备了，事件结果就不发生，而当决定事件的条件不具备时，结果则一定发生，这种因果关系就称为非逻辑，用变量上边的“ $\bar{}$ ”表示。由表 1-4 可知，当 A 为 0 时， Y 为 1； A 为 1 时， Y 为 0。 Y 与 A 间的关系可用 $Y = \bar{A}$ 表示。实现非运算的电路为非门，其逻辑符号如图 1-6 所示。

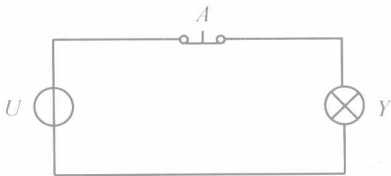


图 1-5 非逻辑的电路图

表 1-4 非逻辑真值表

输入		输出
A		Y
0		1
1		0

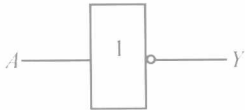


图 1-6 非门逻辑符号

1.3.3 复合逻辑运算

1.3.3.1 与非逻辑运算

与非逻辑运算是与运算和非运算的复合运算，即先进行与运算，而后再进行非运算。设输入变量为 A 、 B ，输出变量为 Y 时，它的逻辑表达式如下：

$$Y = \overline{AB} \tag{1-1}$$

与非运算的真值表见表 1-5。由该表可看出：输入变量 A 、 B 中只要有 0 时，输出 Y 便为 1；只有当输入 A 、 B 都为 1 时，输出 Y 才为 0。实现与非运算的电路为与非门，其逻辑符号如图 1-7 所示。

表 1-5 与非逻辑真值表

输入		输出
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

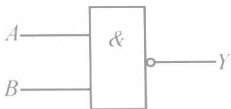


图 1-7 与非门逻辑符号

1.3.3.2 或非逻辑运算

或非逻辑运算是或运算和非运算的复合运算,即先进行或运算,而后再进行非运算。设输入变量为 A 、 B , 输出变量为 Y 时, 它的逻辑表达式如下:

$$Y = \overline{A + B} \quad (1-2)$$

或非运算的真值表见表 1-6。由该表可看出: 输入变量 A 、 B 中只要有 1 时, 输出 Y 便为 0; 只有当输入 A 、 B 都为 0 时, 输出 Y 才为 1。实现或非运算的电路为或非门, 其逻辑符号如图 1-8 所示。

表 1-6 或非逻辑真值表

输入		输出
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

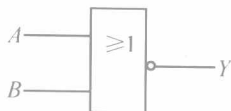


图 1-8 或非门逻辑符号

1.3.3.3 与或非逻辑运算

与或非逻辑运算是与运算和或非运算组成的复合运算,即先进行 A 、 B 和 C 、 D 的与运算,而后再进行或非运算。设输入变量为 A 、 B 、 C 、 D , 输出量为 Y 时, 它的逻辑表达式如下:

$$Y = \overline{AB + CD} \quad (1-3)$$

由该式可看出: 输入变量 A 、 B 和 C 、 D 中同时有 0 时, 输出 Y 为 1; 只有 A 、 B 或 C 、 D 或 A 、 B 、 C 、 D 同时为 1 时, 输出 Y 才为 0。实现与或非运算的电路为与或非门, 其逻辑符号如图 1-9 所示。

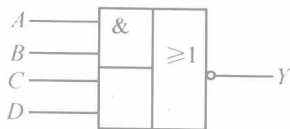


图 1-9 与或非门逻辑符号

1.3.3.4 异或逻辑运算

异或运算是只有两个输入变量的逻辑运算。当输入变量 A 、 B 取值不同时, 输出 Y 为 1; 当输入变量 A 、 B 取值相同时, 输出 Y 为 0。异或运算主要用于判断两个输入变量取值是否不同。异或运算的真值表见表 1-7, 其逻辑表达式如下: