全国机械行业高等职业教育“十二五”规划教材
高等职业教育教学改革精品教材

数字电子技术

SHUZI DIANZI JISHU

刘苏英　主编

机械工业出版社
CHINA MACHINE PRESS

全国机械行业高等职业教育“十二五”规划教材
高等职业教育教学改革精品教材

数字电子技术

主 编 刘苏英
副主编 曹光华 汤德荣
参 编 眭 玲
主 审 武昌俊

机械工业出版社

本书是根据高等职业教育电子信息类理实一体教学的实际需求，结合作者多年来实际教学及实践经验，以培养技能应用型人才为目的而进行编写的。

本书以基本逻辑门功能及静态参数测试、八路抢答器的设计与制作、数字时钟的设计与制作、三位半直流数字电压电流表的设计与制作四个实际项目引入数字电子技术的学习。项目1基本逻辑门功能及静态参数测试主要介绍数字电路的基础知识和基本逻辑门在使用上的相应知识；项目2八路抢答器的设计与制作主要介绍组合逻辑电路的分析和设计、常用组合逻辑电路的应用；项目3数字时钟的设计与制作主要介绍时序逻辑电路的分析和设计、常用时序逻辑电路的应用；项目4三位半直流数字电压电流表的设计与制作主要介绍模-数和数-模转换的原理及应用、综合电子电路的设计与制作方法。该书用四个实际项目概括了所有知识点，每个知识点都有项目与其对应。项目与知识点不像其他教材那样，讲完理论后做个与其对应的实验便罢，而是让读者通过项目来学习理论，实现了理论与实践的紧密结合。

根据高职高专技能型人才培养要求，本书适当把握教材的难度与深度，重视学生应用能力的培养；在必要的知识点以外还增加了拓展知识部分，以便部分同学更深层次地理解知识，拓展知识面。

本书既可作为高职高专电子信息类专业教材，也可作为相关专业的教学用书或技术人员参考书。

本书还配有电子教学参考资料包（教学计划、电子教案、习题答案及项目仿真），凡选用本书作为教材的教师，可登录机械工业出版社教材服务网 www. cmpedu. com 下载或发送电子邮件至cmpgaozhi@ sina. com索取。咨询电话：010-88379375。

图书在版编目（CIP）数据

数字电子技术 / 刘苏英主编. —北京：机械工业出版社，2013. 6
全国机械行业高等职业教育“十二五”规划教材
高等职业教育教学改革精品教材
ISBN 978-7-111-41204-5

Ⅰ. ①数… Ⅱ. ①刘… Ⅲ. ①数字电路－电子技术－高等职业教育－教材 Ⅳ. ①TN79

中国版本图书馆 CIP 数据核字（2013）第 138216 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：边　萌　责任编辑：边　萌　邹云鹏
版式设计：霍永明　责任校对：陈秀丽
封面设计：鞠　杨　责任印制：张　楠
北京京丰印刷厂印刷
2013年8月第1版·第1次印刷
184mm×260mm·10.25印张·250千字
0 001—3 000册
标准书号：ISBN 978-7-111-41204-5
定价：22.00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

根据《国务院关于大力推进职业教育改革与发展的决定》的有关精神,结合高职高专电子信息类专业教学的实际情况,总结近年来教学及实践经验,作者完成了本书的编写工作。

电子技术可分为模拟电子技术和数字电子技术两大类：前者主要分析处理在时间上和幅度上连续变化的模拟信号，对应模拟电路；后者主要分析处理在时间上和数值上离散变化的数字信号，对应数字电路。本教材可以作为模拟电子技术课程的后续课程，也可以和模拟电子技术课程同时学习。

本书的主要内容有

项目1：基本逻辑门功能及静态参数测试，主要介绍数字电路的基础知识、数制与编码、逻辑代数及逻辑函数的化简，基本逻辑门的使用方法及使用注意事项。

项目2：八路抢答器的设计与制作，主要介绍组合逻辑电路的分析和设计，常用组合逻辑电路编码器、译码器、算术运算电路的功能及应用，组合逻辑电路中的竞争与冒险问题，组合逻辑电路的设计、制作与调试方法。

项目3：数字时钟的设计与制作，主要介绍部分组合逻辑电路数据选择器和数据分配器、数据比较器的功能及应用，时序逻辑电路的分析和设计，常用时序逻辑电路集成触发器、寄存器、计数器的功能及应用，脉冲波形的产生与整形电路，时序逻辑电路的设计、制作与调试方法。

项目4：三位半直流数字电压电流表的设计与制作，主要介绍模-数和数-模转换的原理及应用，综合电子电路的设计、制作与调试方法。

另外知识拓展部分主要介绍了晶体管的开关特性、集电极开路门与三态输出门、门电路使用中应注意的问题、数字集成逻辑电路简介、触发器的脉冲工作特性及主要参数、半导体存储器和可编程逻辑器件，以便学生从知识的深度和广度上进行拓展。

项目电路使用的载体有实验箱或实验台、面包板、万能板和PCB四种。

本书在编写过程中以能力培养为主，以应用为目的。内容的深度和广度符合国家高职高专电子技术基础课程的教学基本要求。淡化集成电路内部结构，注重器件的功能和应用，尤其是多个器件间的综合应用。本教材的参考学时为80学时。

本书由安徽机电职业技术学院刘苏英任主编，编写项目3并负责全书的组织、修改和定稿工作。汤德荣任副主编，编写项目2及项目3中3.1和3.2节，曹光华任副主编，编写项目4及附录，眭玲编写项目1。本书由武昌俊任主审。另外浙江天煌科技实业有限公司也为本书的编写提供了一些资料，在此表示衷心的感谢。

由于编者水平有限，书中难免会有不妥和错误之处，恳请使用本教材的师生和读者给予批评指正。

编　者

目　　录

前言
项目 1　基本逻辑门功能及静态参数测试……1
【项目描述及目标】……1
【相关知识】……1
1.1　概述……1
1.2　数制与编码……2
1.3　逻辑代数……6
1.4　逻辑函数的化简……10
【项目分析】……16
【项目实施】……18
【项目评价与习题】……21
【拓展知识 1】　二极管、晶体管的开关特性……23
【拓展知识 2】　TTL 集电极开路门与三态输出门的应用……25
【拓展知识 3】　门电路使用中应注意的问题……27
项目 2　八路抢答器的设计与制作……30
【项目描述及目标】……30
【相关知识】……31
2.1　组合逻辑电路……31
2.2　编码器……33
2.3　译码器……38
2.4　算术运算电路……43
2.5　组合逻辑电路中的竞争与冒险……46
【项目分析】……47
【项目实施】……49
【项目评价与习题】……50
【拓展知识】　数字集成逻辑电路简介……52
项目 3　数字时钟的设计与制作……56
【项目描述及目标】……56
【相关知识】……56
3.1　数据选择器和数据分配器……56
3.2　数值比较器……61
3.3　集成触发器……62
3.4　时序逻辑电路……76
3.5　寄存器……85
3.6　计数器……91
3.7　脉冲波形的产生与整形……102
【项目分析】……115
【项目实施】……117
【项目评价与习题】……120
【拓展知识】　触发器的脉冲工作特性及主要参数……124
项目 4　三位半直流数字电压电流表的设计与制作……127
【项目描述及目标】……127
【相关知识】……127
4.1　数-模和模-数转换……127
4.2　集成 D-A、A-D 转换器的应用……136
【项目分析】……140
【项目实施】……144
【项目评价与习题】……148
【拓展知识】　半导体存储器和可编程逻辑器件……149
附录　常用数字集成电路汇编……153
参考文献……158

项目1　基本逻辑门功能及静态参数测试

【项目描述及目标】

本项目主要是验证基本逻辑门的功能和静态参数，讲述数字电路基础知识，包括数字电路概述、数制与编码、逻辑代数、逻辑函数的化简及基本逻辑门的测试等内容。要求学生熟练掌握这些基础知识及基本逻辑门的使用；掌握逻辑门的分类、命名及使用注意事项等。通过本项目的学习，学生应达到以下目标：

序　号	类　别	目　　标
1	知识点	（1）掌握数字电路基础知识 （2）掌握逻辑门的分类、命名方法
2	技能点	（1）学会基本逻辑门的使用及注意事项 （2）已知数字集成电路的型号，可以查询其功能及使用方法 （3）已知要设计电路的功能要求，可以查询对应的集成电路型号并根据实际要求选择最适合的集成电路型号 （4）能够测试集成电路的好坏 （5）掌握集成电路的接线方法 （6）掌握数字电路的连接、调试及故障排除方法
3	职业素养	（1）培养学生的沟通能力及团队协作精神 （2）培养学生的细心、耐心 （3）培养学生良好的职业道德 （4）培养学生的质量、成本、安全和环保意识

【相关知识】

1.1　概述

电子电路中的信号可分为两类，一类是在时间上和幅度上都连续的信号，称为模拟信号，如图1-1a所示，传送和处理模拟信号的电路，称为模拟电路；另一类是在时间上和幅度上都离散的信号，称为数字信号，如图1-1b所示，传送和处理数字信号的电路，称为数字电路。

数字电路的特点：

（1）数字电路的工作信号是离散的数字信号。

（2）数字电路中，半导体器件均工作在开关状态，即工作在饱和区和截止区。

（3）数字电路研究的主要问题是输入、输出之间的逻辑关系。

（4）数字电路的主要分析工具是逻辑代数。

本项目主要介绍数字电路的基础知识：数制与编码、逻辑函数的表示方法及其化简等。通过本项目的学习，大家应该能够熟练地掌握图 1-2 中所示的各种转换。

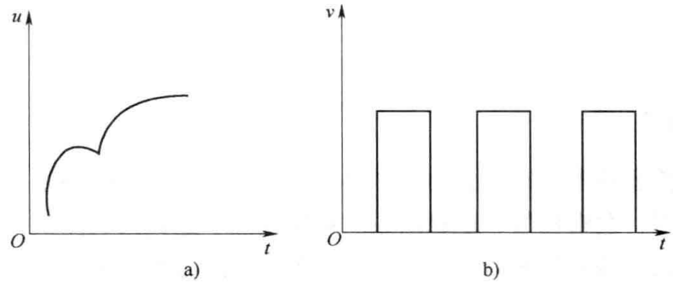


图 1-1　模拟信号和数字信号波形图

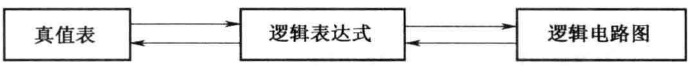


图 1-2　逻辑函数各表示形式间的转换关系

1.2　数制与编码

1.2.1　数制

数制是计数进位制的简称。在日常生活和生产中，人们习惯用十进制数，而在数字电路和计算机中，只能识别“0”和“1”构成的数码，所以经常采用的是二进制数和十六进制数，有些地方还用到八进制数。

1. 十进制

十进制数是以 10 为“基数”的计数体制，有 0 ~ 9 十个数码。数的组成自左向右由高位到低位排列，计数时，“逢十进一，借一当十”。数码在不同的位置代表的数值不同，称之为“位权”或简称为“权”，位数越高，权值越重。

例如，十进制数 54. 214 可表示为

$$(54.214)_D = 5\times10^1 + 4\times10^0 + 2\times10^{-1} + 1\times10^{-2} + 4\times10^{-3}$$

那么，任意一个十进制整数 $(N)_D$ 都可以按权展开为

$$\begin{aligned}(N)_D &= K_{n-1}\times10^{n-1} + K_{n-2}\times10^{n-2} + \cdots + K_1\times10^1 + K_0\times10^0 \\ &= \sum_{i=0}^{n-1} K_i\times10^i\end{aligned}$$

式中，下标 D 表示十进制数；K_i 代表第 i 位的系数，可取 0 ~ 9 这 10 个数码中的任一个；10^i 为第 i 位的权；n 为原数的位数。

2. 二进制

二进制数是以 2 为基数的计数体制。它只有 0 和 1 两个数码，采用“逢二进一，借一当二”的计数规律。一个二进制数 $(N)_B$ 可以按权展开为

$$(N)_B = K_{n-1}\times 2^{n-1} + K_{n-2}\times 2^{n-2} + \cdots + K_1\times 2^1 + K_0\times 2^0 = \sum_{i=0}^{n-1} K_i\times 2^i$$

式中，下标 B 表示二进制数；K_i 表示第 i 位的系数，只能取 0 或 1；2^i 为第 i 位的权；n 为原数位数。

例如，四位二进制数 1011，可以表示为

$$(1011)_B = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0$$

二进制数的运算规则为

加法 $0+0=0$　　$0+1=1+0=1$　　$1+1=10$

乘法 $0\times 0=0$　　$0\times 1=1\times 0=0$　　$1\times 1=1$

二进制数比较简单，只有 0 和 1 两个数码，而且运算规则也很简单，所以二进制数在数字电路中获得了广泛应用。但是二进制数也有缺点：用二进制表示一个数时，位数多，读写不方便，而且也难记忆。

3. 八进制

八进制数是以 8 为基数的计数体制，它用 0，1，2，…，7 这 8 个数码表示，采用“逢八进一，借一当八”的计数规律。三位二进制码可用一位八进制码表示。一个八进制数 $(N)_O$ 可按权展开为

$$(N)_O = K_{n-1}\times 8^{n-1} + K_{n-2}\times 8^{n-2} + \cdots + K_1\times 8^1 + K_0\times 8^0 = \sum_{i=0}^{n-1} K_i\times 8^i$$

式中，下标“O”表示八进制数；K_i 表示第 i 位的系数，可取 0 ~ 7 这 8 个数；8^i 为第 i 位的权；n 为原数总位数。

例如，一个三位八进制数 625，可以表示为

$$(625)_O = 6\times 8^2 + 2\times 8^1 + 5\times 8^0$$

4. 十六进制

十六进制数是以 16 为基数的计数体制，它用 0，1，2，…，9，A，B，C，D，E，F 这 16 个数码表示，采用“逢十六进一，借一当十六”的计数规律。四位二进制码可用一位十六进制码表示。一个十六进制数 $(N)_H$ 可以按权展开为

$$(N)_H = K_{n-1}\times 16^{n-1} + K_{n-2}\times 16^{n-2} + \cdots + K_1\times 16^1 + K_0\times 16^0 = \sum_{i=0}^{n-1} K_i\times 16^i$$

例如，一个四位十六进制数 4A8C，可以表示为

$$(4A8C)_H = 4\times 16^3 + 10\times 16^2 + 8\times 16^1 + 12\times 16^0$$

1.2.2 数制转换

1. 二进制、八进制、十六进制数转换为十进制数

将一个二进制、八进制或十六进制数转换成十进制数，只要写出该进制数的按权展开式，然后按十进制数的计数规律相加，就可以得到所求的十进制数。

【例 1-1】 将二进制数 $(1101)_B$ 转换成十进制数。

解： $(1101)_B = 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 = (13)_D$

【例 1-2】 将八进制数 $(156)_O$ 转换成十进制数。

解： $(156)_O = 1\times8^2+5\times8^1+6\times8^0=(110)_D$

【例 1-3】 将十六进制数 $(5D4)_H$ 转换成十进制数。

解： $(5D4)_H=5\times16^2+13\times16^1+4\times16^0=(1492)_D$

2. 十进制正整数转换为二进制、八进制、十六进制数

对于整数部分，可分别采用“除 2 取余法”、“除 8 取余法”、“除 16 取余法”，求得转换后二、八、十六进制数整数部分的各位数码。转换时注意，第一个余数为转换数码的最低位数码，最后一个余数为转换数码的最高位数码，直到商为 0 为止。

对于小数部分，可分别采用“乘 2 取整法”、“乘 8 取整法”、“乘 16 取整法”，求得转换后二、八、十六进制数小数部分的各位数码。转换时注意，将原十进制数小数部分乘以要转换的计数体制的基数，如 2、8、16，取其积的整数部分作为转换后小数部分的数码，剩余的纯小数部分再乘以基数。将先得到的整数作为转换后数码小数部分的高位，将后得到的整数作为低位，直到其纯小数部分为 0 或到达一定的精度为止。

【例 1-4】 将十进制数 $(35.25)_D$ 转换为二进制数。

解： 整数部分采用“除 2 取余法”

2	35		
2	17	余 $1\cdots K_0=1$	低位 ↑
2	8	余 $1\cdots K_1=1$	
2	4	余 $0\cdots K_2=0$	
2	2	余 $0\cdots K_3=0$	
2	1	余 $0\cdots K_4=0$	
	0	余 $1\cdots K_5=1$	高位

最后商为 0，于是得

$$(35)_D=(K_5K_4K_3K_2K_1K_0)_B=(100011)_B$$

小数部分采用“乘 2 取整法”

$0.25\times2=0.5$	整数	$0\cdots K_{-1}=0$	高位 ↓
$0.5\times2=1.0$	整数	$1\cdots K_{-2}=1$	低位

得 $(0.25)_D=(0.01)_B$

所以，$(35.25)_D=(100011.01)_B$

【例 1-5】 将 $(139)_D$ 转换成八进制数。

解： 采用“除 8 取余法”

8	139		
8	17	余 $3\cdots K_0=3$	低位 ↑
8	2	余 $1\cdots K_1=1$	
	0	余 $2\cdots K_2=2$	高位

最后商为 0，于是得

$$(139)_D = (K_2K_1K_0)_O = (213)_O$$

【例 1-6】 将 $(139)_D$ 转换成十六进制数。

解： 采用"除 16 取余法"

$$\begin{array}{r|cll}
16 & 139 & & \\ \cline{2-2}
16 & 8 & \text{余 } 11 \cdots K_0 = \mathrm{B} & \text{低位} \\ \cline{2-2}
 & 0 & \text{余 } 8 \cdots K_1 = 8 & \text{高位}
\end{array}$$

最后商为 0，于是得

$$(139)_D = (K_1K_0)_H = (8B)_H$$

3. 八进制数、十六进制数与二进制数的相互转换

因为 $2^3 = 8$，所以对三位二进制数来讲，从 000 ~ 111 共有 8 种组合状态，可以分别将这 8 种状态用来表示八进制数码 0，1，2，…，7。这样，每一位八进制数正好相当于三位二进制数。反过来，每三位二进制数就相当于一位八进制数。

同理，$2^4 = 16$，四位二进制数共有 16 种组合状态，可以分别用来表示十六进制的 16 个数码。这样，每一位十六进制数正好相当于四位二进制数。反过来，每四位二进制数就相当于一位十六进制数。

【例 1-7】 将八进制数 $(625)_O$ 转换为二进制数。

解： $(625)_O = (110010101)_B$

【例 1-8】 将二进制数 $(110100111)_B$ 转换为十六进制数。

解： $(110100111)_B = (1A7)_H$

当要求将八进制数和十六进制数互相转换时，可通过先转换为二进制数来完成。

1.2.3 编码

在二进制数字系统中，每一位数只有 0 或 1 两个数码，只限于表达两个不同的信号。用若干位二进制数码表示的数字、文字符号以及其他不同的事物，称为代码。赋予每个代码以固定含义的过程，就称为编码。常用的编码有二 - 十进制码、格雷码等。

1. 二-十进制码

所谓二-十进制编码，就是用四位二进制代码来表示一位十进制数码，简称 BCD 码。由于四位二进制码有 0000，0001，…，1111 等 16 种不同的组合状态，故可以选择其中任意 10 个状态以代表十进制中 0 ~ 9 这十个数码，其余 6 种组合是无效的。因此，按选取方式的不同，可以得到不同的二-十进制编码，最常用的是 8421 码。

8421 码是选用四位二进制码的前 10 个代码 0000 ~ 1001 来表示十进制的十个数码。它是一种有权码。四位二进制编码中由高位到低位的权依次是 8，4，2，1，故称为 8421 码。在 8421 码这类有权码中，如果将其二进制码乘以其对应的权后求和，就是该编码所表示的十进制数。8421 码是最基本的和最常用的一种编码，因此必须熟记。其他编码如 2421 码、5421 码等，大家可以在用到时进行查询。

2. 格雷码

格雷码也是常用的一种编码，它的特点是两个相邻的码只有一位不同。这种编码可靠性高，出错机会少。

1.3 逻辑代数

1.3.1 基本概念与基本逻辑运算

1. 逻辑变量与逻辑函数

逻辑代数是按一定逻辑规律进行运算的代数，它和普通代数一样有自变量和因变量。虽然自变量都可用字母 A，B，C，…来表示，但是只有两种取值，即 0 和 1。这里的 0 和 1 不代表数量的大小，而是表示两种对立的逻辑状态。例如，用“1”和“0”表示事物的“真”与“假”，电位的“高”与“低”，脉冲的“有”与“无”，开关的“闭合”与“断开”等。这种仅有两个取值的自变量具有二值性，称为逻辑变量。

如果逻辑变量 A，B，C，…的取值确定了，逻辑函数 Y 的值也就被唯一地确定了，那么称 Y 是 A，B，C，…的逻辑函数，写作 $Y=F$（A，B，C，…）。

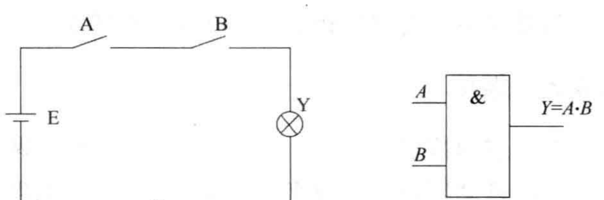
2. 基本逻辑运算

所谓逻辑，是指“条件”与“结果”的关系。在数字电路中，用输入信号反映“条件”，用输出信号反映“结果”，从而输入和输出之间就存在一定的因果关系，称之为逻辑关系。在逻辑代数中，有与逻辑、或逻辑、非逻辑三种基本逻辑关系，相应的基本逻辑运算为与、或、非，对应的门电路有与门、或门、非门。

（1）与运算　“与运算”又称“与逻辑”或“逻辑乘”，实现“与运算”的逻辑电路称为“与门”。

图 1-3a 所示的开关电路中，只有当开关 A 和 B 都闭合时，灯 Y 才亮；A 和 B 中只要有一个断开，灯就灭。

如果以开关闭合作为条件，灯亮作为结果，图 1-3a 所示电路可以表示这样一种因果关系：“只有当决定一件事情（灯亮）的所有条件（开关 A、B）都具备（都闭合）时，这件事情才能实现”。这种逻辑关系称为“与逻辑”，记为



a) 与逻辑电路　　b) 与运算的逻辑符号

图 1-3　与逻辑关系

$$Y=A\cdot B$$

式中的“·”表示“与运算”或“逻辑乘”，与普通代数中的乘号一样，它可省略不写，也可省略不读。与运算的逻辑符号如图 1-3b 所示。

与逻辑的运算规则为

$$0\cdot 0=0,\ 0\cdot 1=0,\ 1\cdot 0=0,\ 1\cdot 1=1$$

与逻辑还可以用真值表来表示。所谓真值表，就是将逻辑变量各种可能取值的组合及其相应逻辑函数值列成的表格，如表 1-1 所示。根据与门的逻辑功能，还可画出其波形图，如

图 1-4 所示。

表 1-1　与门真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

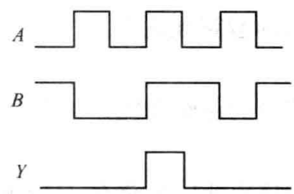


图 1-4　与逻辑波形图

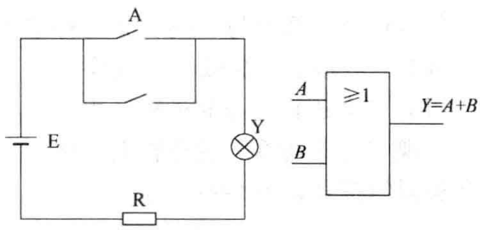
常用的集成与门电路有 74LS08、CC4081 等。

(2) 或运算（逻辑加）　当决定一件事情的所有条件中只要有一条具备，这件事情就能实现时，这种因果关系称为或逻辑，也称为或运算或逻辑加。实现“或运算”的逻辑电路称为“或门”。

图 1-5a 所示的开关电路中，开关 A 和 B 中只要有一个闭合，灯 Y 就会亮，则 Y 与 A 和 B 的关系属于或逻辑。其逻辑表达式为

$$Y = A + B$$

式中的“+”为表示“或运算”或“逻辑加”，其逻辑符号如图 1-5b 所示。



a）或逻辑电路　　b）或逻辑符号

图 1-5　或逻辑关系

或逻辑的运算规则为

$0+0=0$，$0+1=1$，$1+0=1$，$1+1=1$

根据或门的逻辑功能，可得或门的真值表如表 1-2 所示，波形图如图 1-6 所示。

表 1-2　或门真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

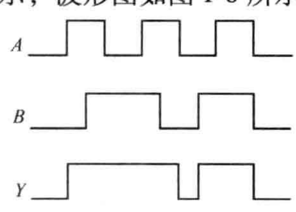


图 1-6　或逻辑波形图

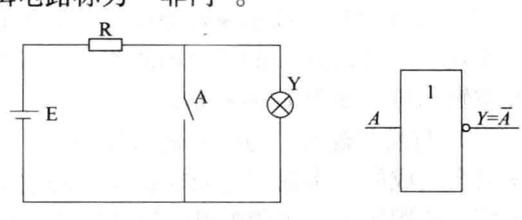
常用的集成或门电路有 74LS32、CC4071 等。

(3) 非运算（非逻辑、逻辑反）　条件的具备与事情的实现刚好相反，这种因果关系称为非逻辑，也称逻辑反。实现“非运算”的逻辑电路称为“非门”。

图 1-7a 所示的开关电路中，当开关 A 闭合时，灯 Y 不会亮；当开关 A 断开时，灯 Y 就会亮。此时 Y 与 A 的关系属于非逻辑，其逻辑表达式为

$$Y = \overline{A}$$

式中，字母 A 上方的横线表示“非逻辑”，读作“非”，即“$\overline{A}$”读作“A 非”。非



a) 非逻辑电路　　b) 非逻辑符号

图 1-7　非逻辑关系图

逻辑的逻辑符号如图 1-7b 所示。

非逻辑的运算规则为

$$\overline{0}=1,\ \overline{1}=0$$

根据非门的逻辑功能，可得非门真值表如表 1-3 所示，还可画出其波形图，如图 1-8 所示。

常用的集成非门电路有 74LS04、74LS06 等。

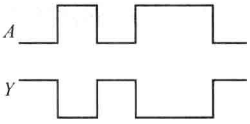


图 1-8　非逻辑波形图

表 1-3　非门真值表

A	Y
0	1
1	0

3. 复合逻辑运算

复合逻辑关系由与、或、非三种基本逻辑关系组合而成。常见的复合逻辑关系有：与非、或非、与或非、异或及同或等。

（1）与非逻辑　与非逻辑是由与、非两种基本逻辑关系按照先与后非的顺序复合而成的。实现“与非运算”的逻辑电路称为“与非门”。以两输入端与非逻辑为例，其逻辑表达式和逻辑门符号如图 1-9a 所示。

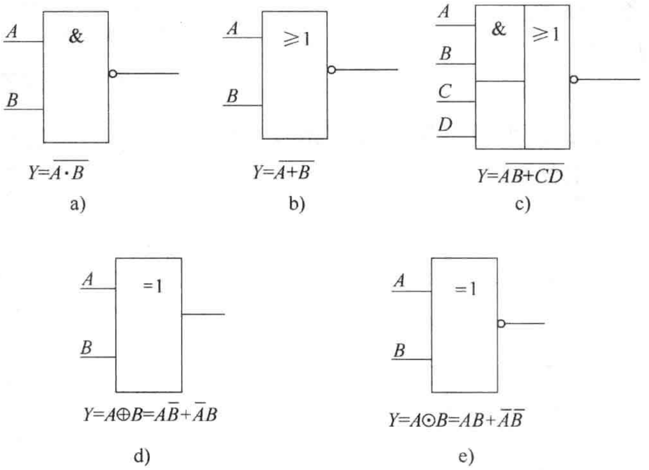


图 1-9　常用复合逻辑关系图

（2）或非逻辑　或非逻辑是由或、非两种基本逻辑关系按照先或后非的顺序复合而成的。实现“或非运算”的逻辑电路称为“或非门”。以两输入端或非逻辑为例，其逻辑表达式和逻辑门符号如图 1-9b 所示。

（3）与或非逻辑　与或非逻辑是由与、或、非三种基本逻辑关系按照先与再或后非的顺序复合而成的。实现“与或非运算”的逻辑电路称为“与或非门”。以四输入端与或非逻辑为例，其逻辑表达式和逻辑门符号如图 1-9c 所示。

（4）异或逻辑　以两输入端异或逻辑为例，当两输入信号相异（不同）时，输出为“1”；当两输入信号相同时，输出为“0”。实现“异或运算”的逻辑电路称为“异或门”。

其逻辑表达式和逻辑门符号如图 1-9d 所示。

（5）同或逻辑　以两输入端同或逻辑为例，当两输入信号相同时，输出为“1”；当两输入信号相异（不同）时，输出为“0”。实现“同或运算”的逻辑电路称为“同或门”。其逻辑表达式和逻辑门符号如图 1-9e 所示。

4. 逻辑函数的表示方法及相互转换

逻辑函数的表示方法有真值表、逻辑函数表达式、逻辑图、工作波形图和卡诺图。这五种方法之间的相互转换在后面会逐步介绍。

1.3.2　逻辑代数的基本定律和规则

1. 逻辑代数的基本定律

（1）交换律　$A \cdot B = B \cdot A, A + B = B + A$。

（2）结合律　$A(BC) = (AB)C, A + (B + C) = (A + B) + C$。

（3）分配律　$A(B + C) = AB + AC, A + BC = (A + B)(A + C)$。

（4）0－1 律　$1 \cdot A = A, 0 \cdot A = 0, 1 + A = 1, 0 + A = A$。

（5）互补律　$A \cdot \overline{A} = 0, A + \overline{A} = 1$。

（6）重叠律　$A \cdot A = A, A + A = A$。

（7）反演律（德·摩根定律）　$\overline{A \cdot B} = \overline{A} + \overline{B}, \overline{A + B} = \overline{A} \cdot \overline{B}$。

（8）还原律　$\overline{\overline{A}} = A$。

（9）附加律　$AB + \overline{A}C + BC = AB + \overline{A}C$，

$(A + B) \cdot (\overline{A} + C) \cdot (B + C) = (A + B) \cdot (\overline{A} + C)$，

$(A + B) \cdot (\overline{A} + B) = B$，

$AB + \overline{A}B = B$。

2. 基本规则

逻辑代数中有以下三个比较重要的基本规则。

（1）代入规则　在任何一个含有变量 X（假设某变量）的等式中，如果将等式两边所有出现变量 X 的位置都代之以一个逻辑函数 Y，则此等式仍然成立。利用代入规则可扩大公式的应用范围。

例如，在 $A + BC = (A + B)(A + C)$ 中，用 $Y = B + D$ 来取代等式中的变量 A，则有

等式左边　$A + BC = (B + D) + BC = B + D$

等式右边　$(A + B)(A + C) = (B + D + B)(B + D + C) = (B + D)(B + D + C) = B + D$

可见等式仍然成立。

（2）反演规则　对逻辑函数 Y 求其反函数的过程叫反演。

将一个逻辑函数 Y 中的运算符号由“·”变“+”、“+”变“·”，由“0”变“1”、“1”变“0”，由原变量变反变量、反变量变原变量，那么所得到的新函数即为原函数 Y 的反函数，这个规则就是反演规则。

利用反演规则，可较容易地求出一个逻辑函数的反函数，但要注意两点：

1）变换过程中要保持原式中的运算顺序：“先括号，然后乘，最后加”。

2）不是单个变量上的“非”号应保持不变。

例如，$Y = (\overline{A} + \overline{B\,\overline{C}D})\overline{E} + 0$，根据反演规则可求出 $\overline{Y} = [A(\overline{\overline{B} + C + \overline{D}}) + E] \cdot 1$

（3）对偶规则　如果将任何一个逻辑函数 Y 中的符号由“·”变“+”、“+”变“·”，由“0”变“1”、“1”变“0”，所有的变量保持不变，这样所得到的新的函数式就是原逻辑函数 Y 的对偶式，记作 Y'。例如，$Y=A\overline{B}+\overline{A}B$，则 $Y'=(A+\overline{B})(\overline{A}+B)$

由原式求对偶式时，要注意保持原式中的运算顺序。

可以证明，如果两个逻辑函数 Y 和 F 相等，那么它们的对偶式 Y' 和 F' 也一定相等。因此，运用对偶规则有时可以使函数的化简、等式的证明大大简化。例如，$A(B+C)=AB+AC$，求这一公式两边的对偶式，则有分配律 $A+BC=(A+B)(A+C)$ 成立。

1.4　逻辑函数的化简

1.4.1　代数法化简

1. 化简的意义和最简的概念

（1）化简的意义　对于同一个逻辑函数，如果表达式不同，则实现它的逻辑电路的复杂程度和所用逻辑元件也不同。例如，逻辑函数

$$Z=A\overline{B}C+AB\overline{C}+ABC=A\overline{B}C+AB=A(B+\overline{B}C)=A(B+C)=AB+AC \quad ①$$
$$=\overline{\overline{AB}\cdot\overline{AC}} \quad ②$$

式①对应电路图如图 1-10a 所示，可用一块两输入与门 74LS08 和一块两输入或门 74LS32 实现。式②对应电路图如图 1-10b 所示，用一块两输入与非门 74LS00 即可实现。

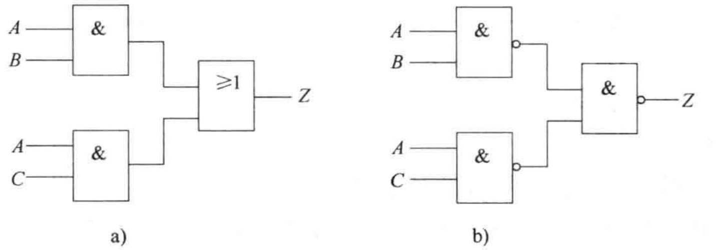


图 1-10　逻辑电路图

从上例可以看出，对逻辑函数做适当的化简，可以使电路的复杂程度降低或使实现电路的成本降低。

（2）最简的概念　一个给定的逻辑函数，其真值表是唯一的，但其表达式可以有多种不同的形式。例如，逻辑函数 $Y=AB+\overline{A}C$ 就可以用如下五种基本形式表示：

$$\begin{aligned} Y &= AB+\overline{A}C && \text{与或表达式} \\ &= (A+C)(\overline{A}+B) && \text{或与表达式} \\ &= \overline{\overline{AB}\cdot\overline{\overline{A}C}} && \text{与非—与非表达式} \\ &= \overline{\overline{A+C}+\overline{\overline{A}+B}} && \text{或非—或非表达式} \\ &= \overline{\overline{A}\,\overline{C}+A\overline{B}} && \text{与或非表达式} \end{aligned}$$

根据上述五种表达式画出的逻辑图如图 1-11 所示，可以看出，不同的化简结果实现起来需要的芯片及芯片个数也不同。

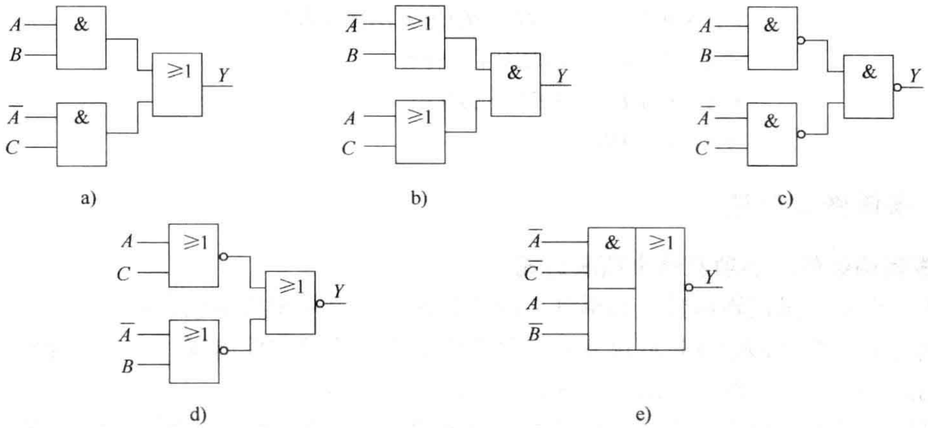


图 1-11　$Y=AB+\overline{A}C$ 的五种逻辑图

对于不同类型的表达式,最简的标准也不一样。最常见的表达式是“与或式”,由它可以比较容易地转换成其他类型的表达式,所以在此主要介绍“与或式”的化简。最简“与或式”的标准是:①乘积项的个数最少;②每一个乘积项中变量的个数最少。

因为乘积项的个数最少,所以对应的逻辑电路所用的与门个数就最少;因为乘积项中变量的个数最少,所以对应逻辑电路所用的与门输入端个数就最少。如果逻辑表达式是最简的,则实现它的逻辑电路也是最简的,但使用的芯片数量却不一定最少。

2. 代数化简法

代数化简法也称公式化简法,就是利用逻辑代数的基本定律和基本规则来简化逻辑函数的一种方法。

(1) 并项法　利用公式 $AB+A\overline{B}=A$,将两项合并成一项,消去一个变量。例如:

$$A(BC+\overline{B}\,\overline{C})+A(B\overline{C}+\overline{B}C)=A(\overline{B\oplus C})+A(B\oplus C)=A$$

(2) 吸收法　利用公式 $AB+A=A$,消去多余的乘积项。例如:

$$A\overline{B}+A\overline{B}CD(E+\overline{F})=A\overline{B}$$

(3) 消去法　利用公式 $A+\overline{A}B=A+B$ 和 $AB+\overline{A}C+BC=AB+\overline{A}C$ 消去多余的因子和多余项。例如:

$$AB+\overline{A}C+\overline{B}C=AB+(\overline{A}+\overline{B})C=AB+\overline{AB}C=AB+C$$

(4) 配项法　利用 $A=A(B+\overline{B})$,对不能直接应用公式化简的乘积项配上 $B+\overline{B}$进行化简。例如:

$$\begin{aligned}Y&=A\overline{B}+B\overline{C}+\overline{B}C+\overline{A}B\\&=A\overline{B}+B\overline{C}+(A+\overline{A})\overline{B}C+\overline{A}B(C+\overline{C})\\&=A\overline{B}+B\overline{C}+A\overline{B}C+\overline{A}\,\overline{B}C+\overline{A}BC+\overline{A}B\overline{C}\\&=(A\overline{B}+A\overline{B}C)+(B\overline{C}+\overline{A}B\overline{C})+(\overline{A}\,\overline{B}C+\overline{A}BC)\\&=A\overline{B}+B\overline{C}+\overline{A}C\end{aligned}$$

逻辑函数化简的途径并不是唯一的,上述四种方法可以任意选用或综合运用。

【例 1-9】　求函数 $Z=AD+A\overline{D}+AB+\overline{A}C+BD+ACEF+\overline{B}EF+DEFG$ 的最简与或式。

解:　$Z=AD+A\overline{D}+AB+\overline{A}C+BD+ACEF+\overline{B}EF+DEFG$

$$
\begin{aligned}
&= A + AB + \overline{A}C + BD + ACEF + \overline{B}EF + DEFG \\
&= A + \overline{A}C + BD + \overline{B}EF + DEFG \\
&= A + C + BD + \overline{B}EF + DEFG \\
&= A + C + BD + \overline{B}EF
\end{aligned}
$$

1.4.2 卡诺图化简法

1. 逻辑函数的最小项及最小项表达式

对于一个 n 变量逻辑函数，如果其与或表达式的每个乘积项都包含 n 个因子，而这 n 个因子分别为 n 个变量的原变量或反变量，每个变量在乘积项中仅出现一次，这样的乘积项称为函数的最小项，这样的与或式也称为函数的最小项表达式。

由逻辑函数的真值表可直接写出其最小项表达式，即将真值表中所有使函数值为 1 的各组变量，以乘积项之和的形式写出来；在乘积项中，取值为 1 的变量写成原变量形式，取值为 0 的变量则写成反变量形式。

例如：$Z=\overline{A}\,\overline{B}C+\overline{A}B\,\overline{C}+AB\,\overline{C}+ABC$，其真值表如表 1-4 所示。

表 1-4 真值表

A	B	C	Z	A	B	C	Z
0	0	0	0	1	0	0	0
0	0	1	1	1	0	1	0
0	1	0	1	1	1	0	1
0	1	1	0	1	1	1	1

Z 也可表示为

$$Z=\overline{A}\,\overline{B}C+\overline{A}B\,\overline{C}+AB\,\overline{C}+ABC=m_{1}+m_{2}+m_{6}+m_{7}=\sum m\ (1,2,6,7)$$

由真值表也可推出 Z 的反函数 $\overline{Z}=\sum m\ (0,3,4,5)$。

从真值表中还可以看出：

一个 n 变量函数，最小项的数目为 2^n 个，其中所有使函数值为 1 的各最小项之和为函数本身，所有使函数值为 0 的各最小项之和为该函数的反函数。

为了方便，最小项常以代号的形式表示为 m_i，m 代表最小项，下标 i 为最小项的编号。i 是 n 变量取值组合排成二进制数码所对应的十进制数。

例如：三变量逻辑函数的最小项有 $2^3=8$ 个，将输入变量取值为 1 的代以原变量，取值为 0 的代以反变量，则得相应最小项。表 1-5 所示是三变量逻辑函数的最小项。

表 1-5 三变量函数的最小项

A	B	C	最小项	简记符号	输入组合对应的十进制数
0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$	m_0	0
0	0	1	$\overline{A}\,\overline{B}C$	m_1	1
0	1	0	$\overline{A}B\,\overline{C}$	m_2	2
0	1	1	$\overline{A}BC$	m_3	3