

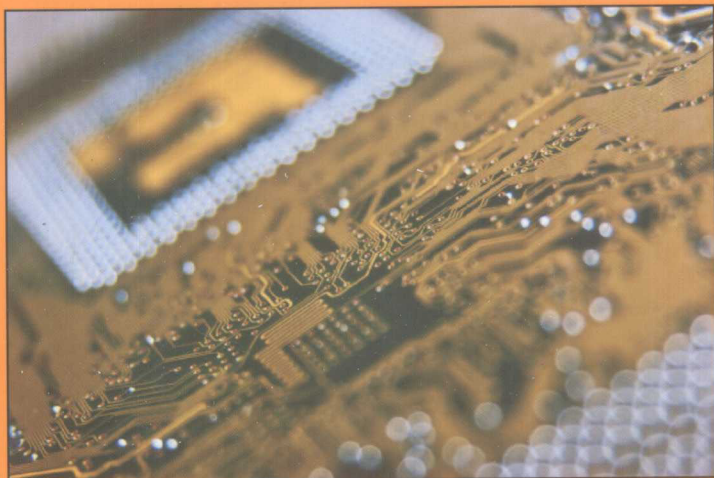


国家示范性高等职业教育电子信息类“十二五”规划教材

# 数字电子技术 与应用

主编 余红梅

SHUZI DIANZI JISHU YU YINGYONG



华中科技大学出版社

<http://www.hustp.com>

国家示范性高等职业教育电子信息类“十二五”规划教材

# 数字电子技术与应用

主 编 余红梅

主 审 彭 节 李 紘

华中科技大学出版社

中国·武汉

## 内 容 提 要

本书从五个典型的学习型工作任务出发,以数字电子技术基础知识为主,主要阐述常用数字电路元器件的特性和功能,常用数字电路的功能、结构、简单原理及单元电路的检修,同时融入常用数字集成电路的识别、检测及应用等。五个典型工作任务的完成,不仅可以使学生掌握有关基础理论知识,提高实际操作技能,还可以培养学生团结协作、规范操作的工作态度。

本书可作为高等职业院校电子与信息、通信技术及其相近专业的教材,也可作为有关工程技术人员学习数字电子技术的参考用书。

## 图书在版编目(CIP)数据

数字电子技术与应用/余红梅主编. —武汉:华中科技大学出版社,2012.6  
ISBN 978-7-5609-8171-0

I. ①数… II. ①余… III. ①数字电路-电子技术-高等职业教育-教材 IV. ①TM79

中国版本图书馆 CIP 数据核字(2012)第 142907 号

## 数字电子技术与应用

余红梅 主编

策划编辑:韩大才 杨 帆

责任编辑:康 序

封面设计:龙文装帧

责任校对:祝 菲

责任监印:张正林

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321915

录 排:武汉楚海文化传播有限公司

印 刷:武汉市籍缘印刷厂

开 本:787mm×1092mm 1/16

印 张:12.25

字 数:309 千字

版 次:2015 年 3 月第 1 版第 1 次印刷

定 价:28.00 元



本书若有印装质量问题,请向出版社营销中心调换  
全国免费服务热线:400-6679-118 竭诚为您服务  
版权所有 侵权必究

# 前 言

为了满足高等职业院校电类专业教学的要求,加快应用型人才培养的步伐,结合高等职业院校的办学特色,编者组织编写了本书。

本书的第一个特点是以行动为导向,以工学结合人才培养模式的改革与实践为基础,遵循“从完成简单工作任务到完成复杂工作任务”的能力形成规律,设计出五个典型学习型工作任务,通过对这五个工作任务的学习,使学生在完成任务的过程中培养职业能力。

本书的第二个特点是为每个任务设计了科学合理的流程,包括任务提出、相关知识、任务实施、评价与总结。其中,在任务提出环节明确了本任务所要达到的知识目标和技能目标,把相关知识细分为基本知识、知识拓展与应用,特别是在知识拓展与应用部分,根据每个任务中所涉及的主要器件或电路,结合军械装备,介绍了对应知识点在相关装备中的典型应用电路。

本书的第三个特点是将知识点与能力点有机结合,注重培养学生的工程应用能力和解决实际问题的能力。本书注重各种数字电路器件的外部特性,对器件的内部结构与电路原理没有进行过多的阐述,通过各种应用实例来使学生熟悉器件在数字电子系统中的应用。

本书的第四个特点是体现了实施过程中的形成性考核机制,成绩的评定贯穿于各项任务的整个实施过程中,每项任务的成绩严格按照评分标准来进行。

本书参考学时为80学时(含部分实践),具体安排如下:任务1共安排14学时,其中理论10学时,实践4学时;任务2共安排10学时,其中理论6学时,实践4学时;任务3共安排18学时,其中理论8学时,实践10学时;任务4共安排22学时,其中理论10学时,实践12学时;任务5共安排16学时,其中理论6学时,实践10学时。使用者可根据具体情况适当增减学时。

本书由武汉军械士官学校余红梅副教授担任主编并负责全书的统稿工作,参加本书编写的还有李紘、张卫清、张涛、陈绍勇、张刚、张勇、任琴、石磊。其中,任务1由张涛和张卫清共同编写,任务2由张刚、石磊、余红梅和张卫清共同编写,任务3由李紘、张勇和任琴共同编写,任务4由陈绍勇、余红梅和张勇共同编写,任务5由余红梅和张卫清共同编写。

本书由武汉军械士官学校彭节副教授、李紘副教授任主审,他们对编写提纲及书稿进行了认真审阅,并提出了许多宝贵意见和建议,在此致以诚挚的谢意!

为了方便教学,本书还配有电子课件等教学资源包,相关教师和学生可以登录“我爱读书网”(www.ibook4us.com)免费注册并下载,或者发邮件至hustpeiit@163.com免费索取。

由于编者水平有限,书中难免存在错误和不妥之处,敬请读者批评指正。

编 者

2014年12月



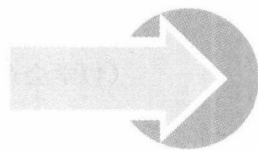
|                                  |       |
|----------------------------------|-------|
| <b>任务 1 裁判器判决电路的分析与制作</b>        | (1)   |
| 1.1 任务提出                         | (2)   |
| 1.2 相关知识                         | (3)   |
| 1.3 任务实施                         | (25)  |
| 1.4 评价与总结                        | (28)  |
| <b>任务 2 简易占空比可调的矩形波发生器的分析与制作</b> | (33)  |
| 2.1 任务提出                         | (34)  |
| 2.2 相关知识                         | (35)  |
| 2.3 任务实施                         | (55)  |
| 2.4 评价与总结                        | (58)  |
| <b>任务 3 简易 4 路抢答器的分析与制作</b>      | (61)  |
| 3.1 任务提出                         | (62)  |
| 3.2 相关知识                         | (63)  |
| 3.3 任务实施                         | (86)  |
| 3.4 评价与总结                        | (90)  |
| <b>任务 4 一位加法计算器的分析与制作</b>        | (95)  |
| 4.1 任务提出                         | (96)  |
| 4.2 相关知识                         | (97)  |
| 4.3 任务实施                         | (123) |
| 4.4 评价与总结                        | (127) |
| <b>任务 5 定时器的分析与制作</b>            | (133) |
| 5.1 任务提出                         | (134) |
| 5.2 相关知识                         | (135) |
| 5.3 任务实施                         | (153) |
| 5.4 评价与总结                        | (157) |
| <b>附录 A 数字频率的设计与制作</b>           | (161) |
| <b>附录 B 常见电气图形符号对照表</b>          | (169) |
| <b>附录 C 虚拟仿真软件 EWB 简介</b>        | (171) |
| <b>附录 D 常用数字集成电路引脚图</b>          | (181) |
| <b>参考文献</b>                      | (189) |



# 任务1

## 裁判器判决电路的 分析与制作

SHUZI  
DIANZIJI SHU  
YUYINGYONG





## 1.1 任务提出



裁判器在人们的日常生活中得到了广泛的应用,如运动会中举重、拳击等有多名裁判的项目,运动员的最终成绩按照少数服从多数的原则评定。虽然不同项目、不同领域的裁判器判决方式各有特色,但其工作原理基本相同。本任务的目标是设计制作一款简单的裁判电路,有一名主裁判和两名副裁判对运动员动作结果进行裁决,当主裁判和至少一名副裁判认为动作正确(开关闭合)时,运动员最终成绩为合格,点亮发光二极管。

任务电路图如图 1.1.1 所示。

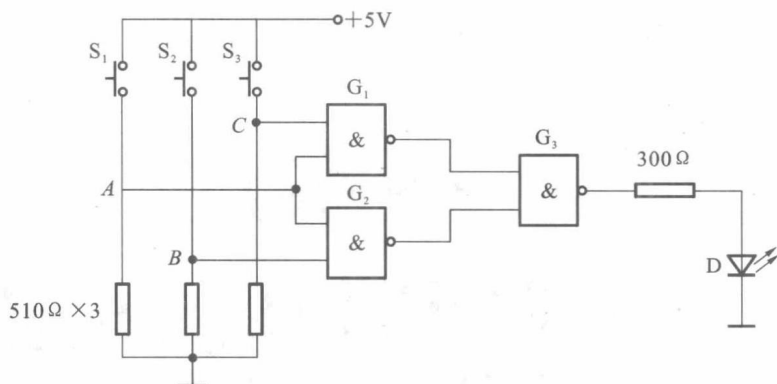


图 1.1.1 裁判器判决电路图

通过本任务相关知识的学习,完成裁判器电路制作,达到以下教学目标。

### 1. 知识目标

- (1) 知道数制的表示方法,学会数制间相互转换的方法。
- (2) 知道 8421BCD 码、余 3 码、格雷码的意义及表示方法。
- (3) 会熟练运用逻辑代数的基本公式、基本定律和三个重要规则。
- (4) 学会逻辑化简的基本方法。
- (5) 学会基本逻辑门电路的工作原理和应用。
- (6) 学会逻辑函数的表示方法间的转换方法。
- (7) 知道 TTL 和 CMOS 两类集成门电路的外部特性。

### 2. 技能目标

- (1) 学会 74LS00、74LS02、74LS04、74LS08 等常用芯片的使用方法。
- (2) 能够按工艺要求独立进行电路装配、测试和调试。
- (3) 能够独立排除装配、调试过程中出现的故障。



## 1.2 相关知识 >>>

### 1.2.1 基本知识

#### 1. 数制和码制

##### 1) 数制

在日常生活中,我们习惯用十进制数,而数字电路中的基本工作信号是二进制的数字信号。因此,在数字系统中进行数字的运算和处理,采用的都是二进制数。有时,为了表示方便,也经常采用十六进制数和八进制数。

##### (1) 十进制数。

十进制数是日常生活和工作中最常用的进位计数制,十进制数的特点如下。

①基数是 10。十进制数采用十个基本数码,即 0、1、2、3、4、5、6、7、8、9,任何一个数都可以用上述十个数码按一定规律排列起来表示。

②计数规律是“逢十进一”。

任意一个十进制数均可以展开为

$$(N)_{10} = \sum k_i \times 10^i$$

其中, $k_i$ 是第  $i$  位的系数,它可以是 0~9 这十个数码中的任意一个。若整数部分的位数是  $n$ ,小数部分的位数为  $m$ ,则  $i$  包涵从  $n-1$  到 0 的所有正整数和从  $-1$  到  $-m$  的所有负整数。例如:

$$(143.75)_{10} = 1 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

上式中的下标 10 表示  $N$  是十进制数。

##### (2) 二进制数。

数字电路中应用最广泛的是二进制数,二进制数具有以下特点。

①基数是 2,采用两个数码 0 和 1。

②计数规律是“逢二进一”。

任何一个二进制数可展开为

$$(N)_2 = \sum k_i \times 2^i$$

例如:

$$(1001.11)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (9.75)_{10}$$

上式中的下标 2 表示  $N$  是二进制数。

##### (3) 十六进制数。

十六进制数具有以下特点。

①十六进制数的基数是 16,采用 16 个数码:0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。

②十六进制数的计数规律是“逢十六进一”。

任何一个十六进制数可展开为





$$(N)_{16} = \sum k_i \times 16^i$$

例如：

$$(80.7F)_{16} = 8 \times 16^1 + 0 \times 16^0 + 7 \times 16^{-1} + 15 \times 16^{-2} = (128.4960937)_{10}$$

式中的下标 16 表示 N 是十六进制数。

(4)八进制数。

八进制数具有以下特点。

- ①八进制数的基数是 8,采用 8 个数码:0、1、2、3、4、5、6、7。
- ②八进制数的计数规律是“逢八进一”。

任何一个八进制数可展开为

$$(N)_8 = \sum k_i \times 8^i$$

例如：

$$(54)_8 = 5 \times 8^1 + 4 \times 8^0 = (44)_{10}$$

式中的下标 8 表示 N 是八进制数。

2)码制

不同的数码不仅可以表示数量的不同大小,而且还能用于表示不同的事物。后一种情况下,这些数码已没有表示数量大小的含义,只是表示不同事物的代号而已。这些数码称为代码。

为便于记忆和处理,在编制代码时要遵循一定的规则,这些规则称为码制。

例如,在用 4 位二进制数表示 1 位十进制数的 0~9 这十个状态时,就有多种不同的码制。通常将这些代码称为二-十进制代码,简称 BCD 码(binary coded decimal)。

表 1.2.1 列出了具有一定规律的常用 BCD 码。

表 1.2.1 常用 BCD 码

| 十进制数 | 8421 码                             | 2421 码                             | 5421 码                             | 余 3 码   |
|------|------------------------------------|------------------------------------|------------------------------------|---------|
| 0    | 0 0 0 0                            | 0 0 0 0                            | 0 0 0 0                            | 0 0 1 1 |
| 1    | 0 0 0 1                            | 0 0 0 1                            | 0 0 0 1                            | 0 1 0 0 |
| 2    | 0 0 1 0                            | 0 0 1 0                            | 0 0 1 0                            | 0 1 0 1 |
| 3    | 0 0 1 1                            | 0 0 1 1                            | 0 0 1 1                            | 0 1 1 0 |
| 4    | 0 1 0 0                            | 0 1 0 0                            | 0 1 0 0                            | 0 1 1 1 |
| 5    | 0 1 0 1                            | 1 0 1 1                            | 1 0 0 0                            | 1 0 0 0 |
| 6    | 0 1 1 0                            | 1 1 0 0                            | 1 0 0 1                            | 1 0 0 1 |
| 7    | 0 1 1 1                            | 1 1 0 1                            | 1 0 1 0                            | 1 0 1 0 |
| 8    | 1 0 0 0                            | 1 1 1 0                            | 1 0 1 1                            | 1 0 1 1 |
| 9    | 1 0 0 1                            | 1 1 1 1                            | 1 1 0 0                            | 1 1 0 0 |
| 位权   | 8 4 2 1<br>$b_3 \ b_2 \ b_1 \ b_0$ | 2 4 2 1<br>$b_3 \ b_2 \ b_1 \ b_0$ | 5 4 2 1<br>$b_3 \ b_2 \ b_1 \ b_0$ | 无权      |

注意,BCD 码用 4 位二进制码表示的只是十进制数的一位。如果是多位十进制数,应先将每一位用 BCD 码表示,然后组合起来。

例如,将十进制数 83 分别用 8421 码、2421 码和余 3 码可表示为



$$(83)_{10} = (10000011)_{8421\text{ BCD}}$$
$$(83)_{10} = (11100011)_{2421\text{ BCD}}$$
$$(83)_{10} = (10110110)_{\text{余3 BCD}}$$

还有一种常用的四位无权码称为格雷码(Gray),其编码如表 1.2.2 所示。这种编码是按照“相邻性”编码的,即相邻两码之间只有一位数字不同。格雷码常用于模拟量的转换中,当模拟量发生微小变化而可能引起数字量发生变化时,格雷码仅改变 1 位,这样与其他码同时改变两位或多位的情况相比更为可靠,可减少出错的可能性。

表 1.2.2 格雷码

| 十进制数 | G <sub>3</sub> G <sub>2</sub> G <sub>1</sub> G <sub>0</sub> |
|------|-------------------------------------------------------------|
| 0    | 0 0 0 0                                                     |
| 1    | 0 0 0 1                                                     |
| 2    | 0 0 1 1                                                     |
| 3    | 0 0 1 0                                                     |
| 4    | 0 1 1 0                                                     |
| 5    | 0 1 1 1                                                     |
| 6    | 0 1 0 1                                                     |
| 7    | 0 1 0 0                                                     |
| 8    | 1 1 0 0                                                     |
| 9    | 1 1 0 1                                                     |
| 10   | 1 1 1 1                                                     |
| 11   | 1 1 1 0                                                     |
| 12   | 1 0 1 0                                                     |
| 13   | 1 0 1 1                                                     |
| 14   | 1 0 0 1                                                     |
| 15   | 1 0 0 0                                                     |

2. 不同进制数之间的相互转换

1) 二进制数和十进制数的相互转换

(1) 二进制数转换为十进制数。

二进制数转换为十进制数称为二-十转换。将二进制数按位权展开,求出其各位加权系数之和,即得到相应的十进制数。

【例 1.2.1】 将二进制数 10011.101 转换成十进制数。

【解】 将每一位二进制数乘以位权,然后相加,可得

$$(10011.101)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$
$$= (19.625)_{10}$$

(2) 十进制数转换为二进制数。

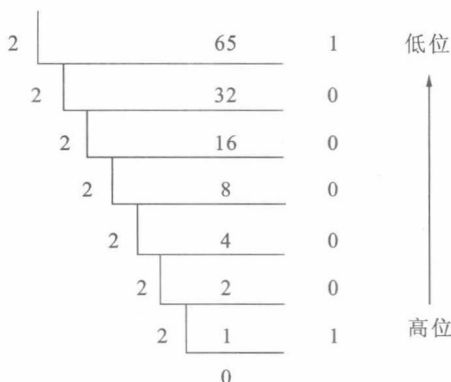
十进制整数转换成二进制整数采用“除 2 取余,逆序排列”法。

十进制小数转换成二进制小数采用“乘 2 取整,顺序排列”法。

因整数部分和小数部分的转换方法不同,所以要分开转换。

【例 1.2.2】 将十进制数(65)<sub>10</sub>转换成二进制数。

【解】 将十进制数(65)<sub>10</sub>转换成二进制数的过程如下。



从上例可看出,“除 2 取余”,就是不断用 2 去整除十进制数直到商为 0;“逆序排列”就是把得到的余数从最低位起逆序排列,即可得到相应的二进制数。所以

$$(65)_{10} = (1000001)_2$$

**【例 1.2.3】** 将十进制数  $(0.562)_{10}$  转换成二进制数。

**【解】** 用“乘 2 取整”法,可按如下步骤转换。

取整

$$\begin{aligned} 0.562 \times 2 &= 1.124 \cdots 1 \\ 0.124 \times 2 &= 0.248 \cdots 0 \\ 0.248 \times 2 &= 0.496 \cdots 0 \\ 0.496 \times 2 &= 0.992 \cdots 0 \\ 0.992 \times 2 &= 1.984 \cdots 1 \end{aligned}$$

如果需要,还可以继续下去。位数越多越精确,若只要五位小数,则

$$(0.562)_{10} = (0.10001)_2$$

显然,十进制数转换成二进制数,可能存在误差。

十进制数转换成其他任意进制数都可以使用基数乘除法。例如,十进制数转换成十六进制数,则整数部分为“除 16 取余,逆序排列”,小数部分为“乘 16 取整,顺序排列”。

## 2) 二进制数和十六进制数的相互转换

二进制数转换为十六进制数的方法为:以小数点为分界,四位一组,不足四位时整数部分高位补零,小数部分低位补零。

十六进制数转换为二进制数的方法:每一位十六进制数用四位二进制数表示。

例如,把二进制数  $(10010.00111)_2$  转换成十六进制数的过程如下:

$$(10010.00111)_2 = (0001\ 0010.0011\ 1000)_2 = (12.38)_{16}$$

例如,将十六进制数  $(3F.25)_{16}$  转换成二进制数的过程如下。

$$(3F.25)_{16} = (0011\ 1111.0010\ 0101)_2 = (111111.00100101)_2$$

## 3. 逻辑变量和逻辑函数

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。后来,由于布尔代数被广泛应用于解决开关电路和数字逻辑电路的分析与设计上,所以也把布尔代数称为逻辑代数。

逻辑代数中用字母表示变量,称为逻辑变量。





如果以逻辑变量作为输入,以运算结果作为输出,那么当输入变量的取值确定之后,输出的取值便随之而定。因此,输出和输入之间是一种函数关系,称为逻辑函数,写为如下形式。

$$F=F(A,B,C,\dots)$$

在逻辑函数中,虽然用字母表示变量,但其取值只有 0 和 1 两种状态,所以我们讨论的都是二值逻辑函数。

#### 4. 逻辑运算

##### 1) 与逻辑

决定某种结果的所有条件都具备时,结果才能发生,这种逻辑关系称为与逻辑关系,简称与逻辑。在图 1.2.1 所示的电路中,只有当开关 A 与开关 B 都闭合时,灯 F 才能亮;只要有一个开关断开,灯就灭。因此,灯亮与开关 A、B 的状态之间是与逻辑关系。与逻辑关系可用下式表示。

$$F=A \cdot B$$

式中的“·”可以省略,因此上式也可写成  $F=AB$  的形式。

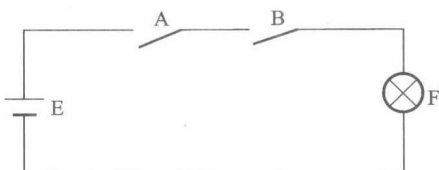


图 1.2.1 与逻辑关系

通常,我们把结果发生或条件具备用逻辑“1”表示,结果不发生或条件不具备用逻辑“0”表示。在此电路中,灯亮用逻辑“1”表示,灯灭用逻辑“0”表示;开关接通用逻辑“1”表示,断开用逻辑“0”表示。因此,本电路中的逻辑关系也可写成下述形式。

| A | B | F   | $A \cdot B = F$ |
|---|---|-----|-----------------|
| 断 | 断 | 灭 → | $0 \cdot 0 = 0$ |
| 通 | 断 | 灭 → | $1 \cdot 0 = 0$ |
| 断 | 通 | 灭 → | $0 \cdot 1 = 0$ |
| 通 | 通 | 亮 → | $1 \cdot 1 = 1$ |

右边的是逻辑与运算,或称为逻辑乘法运算。

把逻辑与运算列成图表,这种图表称为逻辑真值表,或简称为真值表(见表 1.2.3)。

与逻辑运算还可以用图 1.2.2 所示的图形符号表示。

与逻辑运算的规则为:“有 0 出 0,全 1 出 1”。

表 1.2.3 与逻辑真值表

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

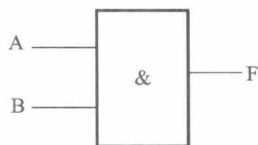


图 1.2.2 与逻辑符号



2)或逻辑

决定某一结果有  $N$  个条件,只要有一个或一个以上的条件具备,结果就发生,这种逻辑关系称为或逻辑关系,简称或逻辑。图 1.2.3 所示的电路中,开关 A 和 B 只要有一个接通,则灯 F 亮。因此,灯亮和开关 A、B 状态之间是或逻辑关系。或逻辑关系可以用下式来表示。

$$F = A + B$$

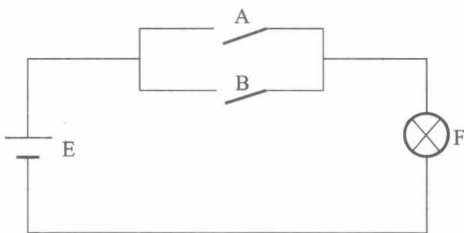


图 1.2.3 或逻辑关系

同样,灯亮、开关接通,我们用逻辑“1”来表示;灯灭、开关断开,我们用逻辑“0”表示。因此本电路中的逻辑关系也可写成下述形式。

| A | B | F | $A + B = F$             |
|---|---|---|-------------------------|
| 断 | 断 | 灭 | $\rightarrow 0 + 0 = 0$ |
| 断 | 通 | 亮 | $\rightarrow 0 + 1 = 1$ |
| 通 | 断 | 亮 | $\rightarrow 1 + 0 = 1$ |
| 通 | 通 | 亮 | $\rightarrow 1 + 1 = 1$ |

右边的是逻辑或运算,或称为逻辑加法运算。

逻辑或真值表如表 1.2.4 所示。

或逻辑运算还可以用图 1.2.4 所示的图形符号表示。

表 1.2.4 或逻辑真值表

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

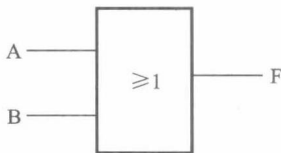


图 1.2.4 或逻辑符号

或逻辑运算的规则为:“有 1 出 1,全 0 出 0”。

3)非逻辑

当结果的条件不具备时,结果发生,这样的逻辑关系称为非逻辑关系,简称非逻辑,或者简称为逻辑非。在图 1.2.5 所示的电路中,当开关 A 断开(条件不具备)时,灯亮(结果发生);当开关 A 闭合时,灯灭。因此,灯亮与开关 A 状态之间为非逻辑关系。非逻辑可以用下式来表示。

$$F = \bar{A}$$

式中的  $\bar{A}$  读作“A 非”或“A 反”。

同样,灯亮、开关接通,我们用逻辑“1”表示;灯灭、开关断开,我们用逻辑“0”表示。因此,本电路中的逻辑关系也可写成下述形式。

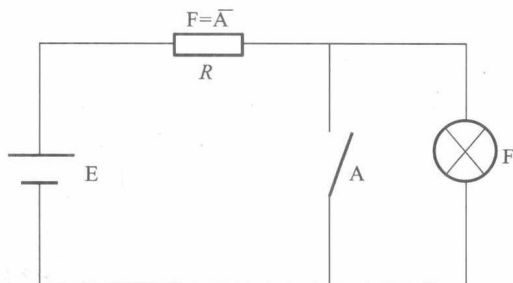


图 1.2.5 非逻辑关系

| A | F | $\bar{A} = F$ |
|---|---|---------------|
| 断 | 亮 | $\bar{0} = 1$ |
| 通 | 灭 | $\bar{1} = 0$ |

右边是逻辑非运算。逻辑非真值表如表 1.2.5 所示。

非逻辑运算还可以用图 1.2.6 所示的图形符号表示。

表 1.2.5 非逻辑真值表

| A | F |
|---|---|
| 0 | 1 |
| 1 | 0 |

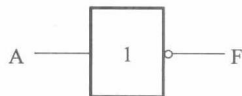


图 1.2.6 非逻辑符号

实际的逻辑问题往往比与、或、非复杂得多,不过它们都可以用与、或、非的组合来实现。最常见的复合逻辑有与非、或非、异或等。

#### 4) 与非逻辑

与非是由与运算和非运算组合而成的,如图 1.2.7 所示。

| A | B | $F = \overline{AB}$ |
|---|---|---------------------|
| 0 | 0 | 1                   |
| 0 | 1 | 1                   |
| 1 | 0 | 1                   |
| 1 | 1 | 0                   |

(a) 逻辑真值表



(b) 逻辑符号

图 1.2.7 与非逻辑运算

#### 5) 或非逻辑

或非是由或运算和非运算组合而成的,如图 1.2.8 所示。

#### 6) 异或逻辑

异或是一种二变量逻辑运算,当两个变量取值相同时,逻辑函数值为 0;当两个变量取值不同时,逻辑函数值为 1。异或的逻辑真值表和相应逻辑门的符号如图 1.2.9 所示。

异或也可以用与、或、非的组合表示。

$$A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$$



| A | B | $L = \overline{A+B}$ |
|---|---|----------------------|
| 0 | 0 | 1                    |
| 0 | 1 | 1                    |
| 1 | 0 | 0                    |
| 1 | 1 | 0                    |

(a) 逻辑真值表

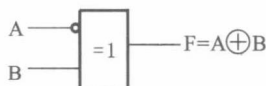


(b) 逻辑符号

图 1.2.8 或非逻辑运算

| A | B | $F = A \oplus B$ |
|---|---|------------------|
| 0 | 0 | 0                |
| 0 | 1 | 1                |
| 1 | 0 | 1                |
| 1 | 1 | 0                |

(a) 逻辑真值表



(b) 逻辑符号

图 1.2.9 异或逻辑运算

### 7) 同或逻辑

同或和异或相反,当两个变量取值相同时,逻辑函数值为 1;当两个变量取值不同时,逻辑函数值为 0。同或的逻辑真值表和相应逻辑门的符号如图 1.2.10 所示。

| A | B | $A \odot B$ |
|---|---|-------------|
| 0 | 0 | 1           |
| 0 | 1 | 0           |
| 1 | 0 | 0           |
| 1 | 1 | 1           |

(a) 逻辑真值表



(b) 逻辑符号

图 1.2.10 同或逻辑运算

同或也可以用与、或、非的组合表示。

$$A \odot B = A \cdot B + \bar{A} \cdot \bar{B}$$

异或与同或互为反运算,即

$$A \odot B = \overline{A \oplus B}$$

## 5. 逻辑函数的表示方法

常用的逻辑函数表示方法有逻辑真值表(简称真值表)、逻辑函数式(也称逻辑式或函数式)和逻辑图。

### 1) 逻辑真值表

将输入变量所有取值对应的输出值找出来,列成表格,即可得真值表。

例如,图 1.2.11 所示的是一个举重裁判电路,根据电路的工作原理不难看出,只有  $A=1$ ,



同时 B、C 至少有一个为 1 时, F 才等于 1, 于是可列出表 1.2.6 所示的真值表。

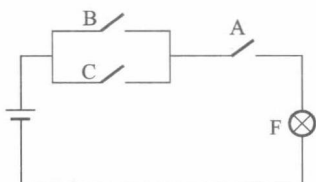


图 1.2.11 举重裁判电路

表 1.2.6 图 1.2.11 电路的真值表

| 输 入 |   |   | 输 出 |
|-----|---|---|-----|
| A   | B | C | F   |
| 0   | 0 | 0 | 0   |
| 0   | 0 | 1 | 0   |
| 0   | 1 | 0 | 0   |
| 0   | 1 | 1 | 0   |
| 1   | 0 | 0 | 0   |
| 1   | 0 | 1 | 1   |
| 1   | 1 | 0 | 1   |
| 1   | 1 | 1 | 1   |

## 2) 逻辑函数式

把输出与输入之间的逻辑关系写成与、或、非等运算的组合式, 就得到了所需的逻辑函数式。

在图 1.2.11 所示的电路中, 根据对电路功能的要求和与、或的逻辑定义, “B 和 C 中至少有一个合上”可以表示为  $(B+C)$ , “同时还要求合上 A”, 则应写作  $A(B+C)$ 。因此得到输出的逻辑函数式为

$$F = A(B+C)$$

## 3) 逻辑图

将逻辑函数中各变量之间的与、或、非等逻辑关系用逻辑符号表示出来, 就可以画出表示函数关系的逻辑图。

为了画出图 1.2.11 所示电路功能的逻辑图, 只要用逻辑运算的图形符号代替  $F = A(B+C)$  中的代数运算符号便可得到图 1.2.12 所示的逻辑图。

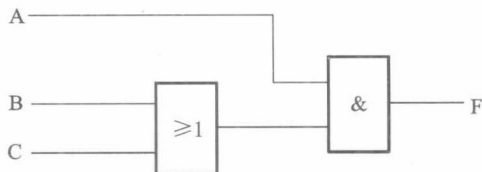


图 1.2.12 举重裁判电路的逻辑图

## 4) 各种表示方法间的互相转换

既然同一个逻辑函数可以用三种不同的方法描述, 那么这三种方法之间必能相互转换。经常用到的转换方式有以下几种。





(1)由真值表写出逻辑函数式。

由真值表写逻辑函数式可按以下步骤进行。

第一步,找出真值表中函数值为“1”的那些输入变量取值的组合。

第二步,每组输入变量取值的组合对应一个乘积项,其中取值为“1”的写入原变量,取值为“0”的写入反变量。

第三步,将这些乘积项相加,即可得到对应的逻辑函数式。

例如,已知一个奇偶判别函数的真值表如表 1.2.7 所示,其对应的逻辑函数式如下。

表 1.2.7 奇偶判别真值表

| A | B | C | F                           |
|---|---|---|-----------------------------|
| 0 | 0 | 0 | 0                           |
| 0 | 0 | 1 | 0                           |
| 0 | 1 | 0 | 0                           |
| 0 | 1 | 1 | $1 \cdots \cdots \bar{A}BC$ |
| 1 | 0 | 0 | 0                           |
| 1 | 0 | 1 | $1 \cdots \cdots A\bar{B}C$ |
| 1 | 1 | 0 | $1 \cdots \cdots AB\bar{C}$ |
| 1 | 1 | 1 | $1 \cdots \cdots ABC$       |

$$F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

(2)由逻辑式列出真值表。

将输入变量取值的所有组合状态逐一代入逻辑式求出函数值,列成表,即可得到真值表。

例如, $F = A + \bar{B}C + \bar{A}B\bar{C}$  对应的真值表如表 1.2.8 所示。

表 1.2.8  $F = A + \bar{B}C + \bar{A}B\bar{C}$  的真值表

| A | B | C | $\bar{B}C$ | $\bar{A}B\bar{C}$ | F |
|---|---|---|------------|-------------------|---|
| 0 | 0 | 0 | 0          | 0                 | 0 |
| 0 | 0 | 1 | 1          | 0                 | 1 |
| 0 | 1 | 0 | 0          | 1                 | 1 |
| 0 | 1 | 1 | 0          | 0                 | 0 |
| 1 | 0 | 0 | 0          | 0                 | 1 |
| 1 | 0 | 1 | 1          | 0                 | 1 |
| 1 | 1 | 0 | 0          | 0                 | 1 |
| 1 | 1 | 1 | 0          | 0                 | 1 |

(3)由逻辑式画出逻辑图。

用逻辑符号代替逻辑式中的运算符号,即可画出逻辑图。

例如, $F = A\bar{B} + \bar{A}B$  对应的逻辑图如图 1.2.13 所示。

(4)由逻辑图写出逻辑式。

从输入端到输出端逐级写出每个逻辑符号对应的逻辑式,就可以得到对应的逻辑函数式了。

例如,图 1.2.14 所示的逻辑图所对应的逻辑式为