新世纪闯关丛书

●考研 ●课考

脉冲与数字电路考点分析及效果测试

丛书编委会 编

西北工业大学出版社
哈尔滨工程大学出版社

【内容简介】 本书是根据21世纪大学生考研及课程学习辅导的需要而编写的。主要内容包括了数制与编码、逻辑函数、中小规模电路分析及设计、大规模集成电路、可编程逻辑器件等。

全书旨在配合《脉冲与数字电路》在大学期间的同步学习,从纲目要求、考点指南、基本题解析、全真题解答、目标测试等方面给出了较为详尽的辅导。在编写风格上,力求简明扼要、通俗易懂;在内容安排上,力求全面细致、仿真性强。通过本书的学习,读者一定会从中汲取丰富的知识和学习的技巧,起到事半功倍的效果。

本书既可作为报考硕士研究生人员的学习辅导书,也可作为相关专业学生在校课程学习和复习的指导书,还可作为相关技术人员和大学教师的参考资料。

图书在版编目(CIP)数据

脉冲与数字电路考点分析及效果测试/ 丛书编委会编.—哈尔滨:哈尔滨工程大学出版社,2003.1

ISBN 7-81073-391-5

Ⅰ.脉… Ⅱ.丛… Ⅲ.脉冲电路-研究生-入学考试-自学参考资料 Ⅳ.TN78

中国版本图书馆CIP数据核字(2002)第102520号

出版发行:西北工业大学出版社,哈尔滨工程大学出版社
通信地址:西安市友谊西路127号 邮编:710072 电话:(029)8493844
网 址:http:// www.nwpup.com
印 刷 者:陕西天元印务有限责任公司
开 本:787 mm×1 092 mm 1/ 16
印 张:11.25
字 数:261千字
版 次:2003年3月第1版 2003年3月第1次印刷
印 数:1～4 000册
定 价:13.00元

新世纪闯关丛书
编审委员会

新世纪闯关丛书
编著委员会

策　　划　张近乐
主　　编　王兴亮
副 主 编　许　杰　李　彦　林家薇　赵雪岩
编写人员　《模拟电子线路考点分析及效果测试》
许　杰　王维忠　李　云　石雨荷　曹闹昌
《脉冲与数字电路考点分析及效果测试》
李　云　石雨荷　许　杰　王维忠　曹闹昌
《电路分析基础考点分析及效果测试》
王国红　李　彦　柳革命
《信号与系统考点分析及效果测试》
李　彦　柳革命　王国红
《通信系统原理考点分析及效果测试》
林家薇　杜思深　张德纯　王兴亮
《微机原理与应用考点分析及效果测试》
赵雪岩　刘　明　秦　莲　姚　群　程绍智　耿　磊

序

跨入 21 世纪的中国，对高层次人才的需求更加迫切，越来越多的应考青年已跻身于考研大军之中。然而，能够针对应考人员需求的学习资料却非常匮乏，为了使理工科电类专业的考生有更加丰富的复习资料，又能使考生在很短的时间内熟练掌握相关的学习内容，达到事半功倍的效果，我们特地编写了这套考研与课程考试相结合的辅导丛书。

本丛书首批共 6 本，即《模拟电子线路考点分析及效果测试》、《脉冲与数字电路考点分析及效果测试》、《电路分析基础考点分析及效果测试》、《信号与系统考点分析及效果测试》、《通信系统原理考点分析及效果测试》及《微机原理与应用考点分析及效果测试》。其中《模拟电子线路考点分析及效果测试》和《脉冲与数字电路考点分析及效果测试》以模拟电子线路和脉冲数字电路为基本内容，侧重于基本概念、线路分析与综合设计；《电路分析基础考点分析及效果测试》和《信号与系统考点分析及效果测试》则从电路和系统的角度入手，注重电路与系统的理论分析和应用；《通信系统原理考点分析及效果测试》从通信系统模型入手，注重基本概念、基本原理及通信技术的性能分析和应用；《微机原理与应用考点分析及效果测试》则以 8086 为主线，注重基本概念、基本原理以及微机的基本应用。丛书的风格一致，各章中均有纲目要求、考点指南、基本题解答、全真题解析及目标测试等，力求使学习者在学习中抓住主线，从各个方面深入掌握各章内容，达到预期的目的。

丛书的特点是简明扼要、层次分明、内容广泛、分

析透彻、针对性强，能够起到典型引路的作用，编著者深信，通过本丛书的学习，读者一定会从中受益。

丛书既可用作高等院校相关专业和学生报考硕士研究生复习辅导书，也可用作相关专业在校课程学习和复习指导书，还可作为通信技术人员和大学有关教师的参考资料。

本丛书由西北工业大学出版社社长张近乐策划，王兴亮教授任主编，许杰、李彦、林家薇、赵雪岩任副主编。张近乐、王兴亮统编全书。

衷心感谢全体作者为本丛书的编写所付出的艰辛劳动；感谢西北工业大学出版社社长张近乐为丛书精心策划使编写水平得以提升并顺利出版；感谢哈尔滨工程大学出版社为本丛书的出版所付出的努力。

编　委　会

2002 年秋于空军工程大学

前　　言

《脉冲与数字电路考点分析及效果测试》侧重于基本概念、基本原理、典型电路的构成、常用电路的分析设计方法和一些常用器件的应用,目的在于提高学生分析问题和解决问题的能力。全书共有 4 章,主要内容有数制与编码、逻辑函数;中小规模电路分析及设计;大规模集成电路;可编程逻辑器件及应用。每章分本章纲目、考点指南、基本题解答、全真题解析、目标测试 5 部分。本章纲目给出了该章的大纲及要求,使学习者有的放矢,使学习内容更加具体化,重点更加突出;考点指南全面系统地归纳和总结课程内容;基本题解答给出了大学课程中最基本的例题和习题的解题过程和分析方法,有助于启发和提高学生的思维能力和解题能力;全真题解析主要选用近年来北京航空航天大学、哈尔滨工业大学、北京理工大学、上海交通大学、北京邮电大学、西安交通大学、西安电子科技大学、电子科技大学、国防科技大学、西北工业大学、空军工程大学等院校的研究生入学试题,并对试题做了适当的修改,为此在全真题中不再一一表明;目标测试可用来检测学生对内容的掌握程度,书后附有目标测试答案,以供学生参考。

本书融入了编著者多年从事该课程教学的经验和体会,其特点是内容广泛、分析透彻细致、指导性强。可作为报考硕士研究生人员的学习辅导书,也可作为相关专业学生及自考生在校课程学习和复习指导书,还可作为有关技术人员和大学教师的参考资料。

本书由许杰组织编写。其中，第 1，2 章由李云、

许杰编写;第 3,4 章由石雨荷、王维忠编写。曹闸昌参与了以上部分内容的编写以及习题的解答工作。张清宇、葛玉鹊副教授对全书的编写给予了具体的指导,在此表示感谢。

由于编著者的水平所限,书中的不足之处在所难免,恳请广大读者和专家予以批评指正。

编 著 者

2002 年 12 月

目　　录

数制与编码、逻辑函数

1 .1　本章纲目

1 .1 .1　数制与编码

1 .教学要求

掌握常用数制及其转换方法,掌握二 — 十进制代码的特点。

2 .内容提要

数字信号与数字电路:十、二、八、十六进制数及其转换方法;二 — 十进制代码及其特点;算术运算与逻辑运算。

3 .重点难点

重点:二 — 十进制代码;二、十、十六进制数的转换方法。

难点:二、十进制数的转换方法。

1 .1 2　逻辑函数及其简化

1 .教学要求

1) 熟练掌握三、四变量逻辑函数的卡诺图化简法。

2) 掌握基本逻辑及其运算;掌握逻辑代数的常用公式和基本规则。

3) 了解逻辑函数的代数化简法和逻辑函数的表示方法。

2 .内容提要

基本逻辑及其运算;复合逻辑及其运算;真值表与逻辑函数;逻辑代数的常用公式及基本规则;逻辑函数的标准形式及函数简化。

3 .重点难点

重点:逻辑函数的卡诺图化简法、常用公式和基本规则。

难点:五变量逻辑函数的卡诺图化简法。

1.2 考点指南

1.2.1 数制与编码

1.数制

数制是计数进位制的简称。在我们日常生活中常使用的是十进制数,而在数字电路中采用的是二进制数。当二进制位数较多时,书写起来很麻烦,特别是在写错了以后不易查找错误,为此书写常采用八进制数和十六进制数。表 1-1 列出了十进制数、二进制数、八进制数以及十六进制数的表示形式。

表 1-1

	十进制数	二进制数	八进制数	十六进制数
数符	0,1,…9	0,1	0,1,…7	0,1,…,9,A,B,C,D,E,F 其中,A ~ F 依次表示十进制数 10,11,12,13,14,15
基数	10	2	8	16
位权值	10^i	2^i	8^i	16^i
按权展开式	$(N)_{10}=\sum_{i=-m}^{n-1} a_i 10^i$	$(N)_2=\sum_{i=-m}^{n-1} a_i 2^i$	$(N)_8=\sum_{i=-m}^{n-1} a_i 8^i$	$(N)_{16}=\sum_{i=-m}^{n-1} a_i 16^i$
	n,m 均为正整数,n 表示整数部分数位,m 表示小数部分数位			

2.不同进制数相互转换

1) 将 R 进制数转换为十进制数

方法:将 R 进制数按位权展开,再相加。

例 1 将二进制数 $(11010.101)_2$ 转换成十进制数。

$$(11010.101)_2 = 1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3} =$$

$$16+8+2+0.5+0.125=(26.625)_{10}$$

2) 将十进制数转换成 R 进制数

方法:将整数部分和小数部分分别进行转换,然后再将它们合并起来。

$$\begin{cases}\text{整数部分:除 } R \text{ 取余数法}\\ \text{小数部分:乘 } R \text{ 取整数法}\end{cases}$$

十进制数整数转换成 R 进制数整数,采用逐次除以基数 R 取余数的方法。

① 将给定的十进制整数除以 R,余数作为 R 进制数的最低位。

② 把第一次的商再除以 R,余数作为次低位。

③ 重复 ② 的步骤,记下余数,直至最后的商为 0,最后的余数即为 R 进制的最高位。

十进制数纯小数转换成 R 进制小数,采取逐次乘以 R,取乘积的整数部分方法。第一次乘 R 的积的整数部分为 R 进制小数部分最高位。将第一次积的小数部分再乘以 R 所得积的整数部分为 R 进制的小数次高位,依次进行下去,直至最后乘积为 0。若最后乘积不会出现 0,要求

达到一定的精度为止。

精确到 0.1%(千分之一)　　取 10 位　　因为$\frac{1}{2^{10}} = \frac{1}{1024}$

精确到 1%(百分之一)　　取 7 位　　因为$\frac{1}{2^{7}} = \frac{1}{128}$

精确到 10%(十分之一)　　取 4 位　　因为$\frac{1}{2^{4}} = \frac{1}{16}$

3) 基数 R 为 2^k 各进制之间的相互转换

① 二进制 → 八进制、十六进制

由于八进制的基数 $8 = 2^3$，十六进制的基数 $16 = 2^4$，因此，将二进制数转换成八进制数和十六进制数相当方便。其转换规则是：

从小数点起向左右两边按 3 位(或 4 位)分组，不满 3 位(或 4 位)的，加 0 补足，每组以其对应的八进制(或十六进制)数码代替，即 3 位合 1 位(或 4 位合 1 位)，顺序排列即为变换后的等值八进制(或十六进制)数。

例 2　$(110101.001000111)_2 = (\qquad)_8 = (\qquad)_{16}$

解　先从小数点起向两边每 3 位合 1 位，不足 3 位的加 0 补足，则得相应的八进制数

$$(\underbrace{110}_{6}\ \underbrace{101}_{5}.\underbrace{001}_{1}\ \underbrace{000}_{0}\ \underbrace{111}_{7})_2 = (65.107)_8$$

从小数点起向两边每 4 位合 1 位，不足 4 位的加 0 补足，则可得相应的十六进制数

$$(\underbrace{^{00}11}_{3}\ \underbrace{0101}_{5}.\underbrace{0010}_{2}\ \underbrace{0011}_{3}\ \underbrace{1^{000}}_{8})_2 = (35.238)_{16}$$

② 八进制、十六进制 → 二进制

方法：从小数点起，对于八进制数，1 位拉 3 位；对于十六进制数，1 位拉 4 位。

例 3

$$(\underset{011}{3}\ \underset{111}{7}.\underset{010}{2})_8 = (11111.01)_2$$

$$(\underset{0001}{1}\ \underset{1010}{A}.\underset{1111}{F})_8 = (11010.1111)_2$$

3.二 — 十进制代码(BCD 代码)

凡用二进制码来表示 1 位十进制数的代码称为二 — 十进制代码，即 BCD 代码。

由于十进制数共有 0，1，…，9 十个数码，因此，至少需要 4 位二进制码来表示 1 位十进制数。4 位二进制码共有 $2^4 = 16$ 种码组，在这 16 种码组中可任选 10 种来表示十个十进制数码。选择方法不同就构成了各种编码。常用的 BCD 码有 8421BCD 码，余 3BCD 码，余 3 格雷 BCD 码等。

1.2.2　逻辑函数

1.基本逻辑函数

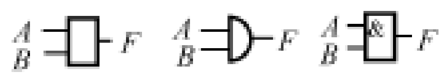
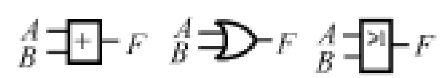
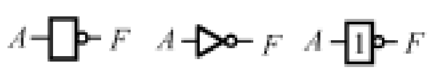
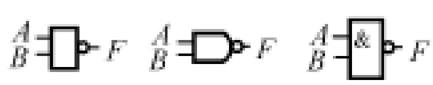
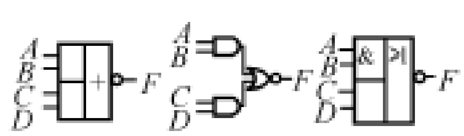
所谓“逻辑”，是指事物的前因(输入)与后果(输出)之间所遵循的规律。

所有的逻辑变量和逻辑函数的取值只能是 0 和 1 这两种状态，这两种状态是互为对立的。逻辑代数中的公式、规则、定理均要用二值逻辑的因果关系来理解。

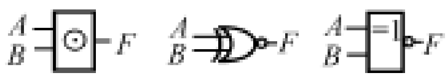
最基本的逻辑关系有 3 种：“与逻辑”、“或逻辑”、“非逻辑”，它们所对应的 3 种基本运算是

"逻辑乘""逻辑加""逻辑非"。利用这 3 种基本运算，可构成一些复合逻辑，即与非、或非，与或非，异或，同或等。实现这些逻辑运算的电路统称为门电路。其逻辑函数，均可通过逻辑函数表达式，逻辑符号，真值表及基本运算规则来描述，如表 1-2 所示。

表 1-2　几种常用的逻辑

<table>
<tr><th rowspan="2">逻辑</th><th colspan="4">描述方法</th></tr>
<tr><th>逻辑函数表达式</th><th>逻辑符号</th><th>真值表</th><th>基本运算规则</th></tr>
<tr><td>与逻辑</td><td>$F = A \cdot B$</td><td></td><td><table><tr><th>A</th><th>B</th><th>$F = AB$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>$A \cdot 1 = A$
$A \cdot 0 = 0$
$A \cdot A = A$
$A \cdot \overline{A} = 0$</td></tr>
<tr><td>或逻辑</td><td>$F = A + B$</td><td></td><td><table><tr><th>A</th><th>B</th><th>$F = A + B$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>$A + 1 = 1$
$A + 0 = A$
$A + A = A$
$A + \overline{A} = 1$</td></tr>
<tr><td>非逻辑</td><td>$F = \overline{A}$</td><td></td><td><table><tr><th>A</th><th>$F = \overline{A}$</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td><td>$\overline{\overline{A}} = A$</td></tr>
<tr><td>与非逻辑</td><td>$F = \overline{A \cdot B}$</td><td></td><td><table><tr><th>A</th><th>B</th><th>$F = \overline{A \cdot B}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td>$\overline{A \cdot 1} = \overline{A}$
$\overline{A \cdot 0} = 1$
$\overline{A \cdot B} = \overline{A} + \overline{B}$</td></tr>
<tr><td>或非逻辑</td><td>$Y = \overline{A + B}$</td><td></td><td><table><tr><th>A</th><th>B</th><th>$F = \overline{A + B}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td>$\overline{A + 0} = \overline{A}$
$\overline{A + 1} = 0$
$\overline{A + B} = \overline{A} + \overline{B}$</td></tr>
<tr><td>与或非逻辑</td><td>$Y = \overline{AB + CD}$</td><td></td><td><table><tr><th>A</th><th>B</th><th>C</th><th>D</th><th>$F = \overline{AB + CD}$</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table></td><td>$\overline{AB + CD} = \overline{AB} \cdot \overline{CD}$</td></tr>
</table>

续表

<table>
<tr><th rowspan="2">逻辑</th><th colspan="4">描述方法</th></tr>
<tr><th>逻辑函数表达式</th><th>逻辑符号</th><th>真值表</th><th>基本运算规则</th></tr>
<tr><td>异或逻辑</td><td>$F = A \quad B = \overline{A}B + A\overline{B}$</td><td></td><td>A B \| F = A B
0 0 \| 0
0 1 \| 1
1 0 \| 1
1 1 \| 0</td><td>$A \quad A = 0$
$A \quad \overline{A} = 1$
$A \quad 0 = A$
$A \quad 1 = \overline{A}$
$\overline{A \quad B} = A \odot B$
$A \quad \overline{B} = \overline{A \quad B}$
$A \quad B \quad C =$
$A \odot B \odot C$</td></tr>
<tr><td>同或逻辑</td><td>$F = A \odot B = \overline{AB} + AB$</td><td></td><td>A B \| Y = A⊙B
0 0 \| 1
0 1 \| 0
1 0 \| 0
1 1 \| 1</td><td>$\overline{A \odot B} = A \quad B$</td></tr>
</table>

2 .基本定律、规则与定理

逻辑代数的基本定律、规则与定理是化简逻辑函数的基础，是逻辑运算的重要工具。

1) 逻辑代数的基本定律

如表 1－3 所示。

表 1－3　逻辑代数基本定律

序号	基本定律		对偶式
1	0 - 1 律	$1 \cdot A = A$	$0 + A = A$
		$0 \cdot A = 0$	$1 + A = 1$
2	互补律	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
3	交换律	$A + B = B + A$	$AB = BA$
4	结合律	$A + (B + C) = (A + B) + C$	$A(BC) = (AB) \cdot C$
5	分配律	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
6	重叠律	$A + A = A$	$A \cdot A = A$
7	非非律	$\overline{\overline{A}} = A$	
8	反演律（狄·摩根定律）	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

2)3 个规则

(1) 代入规则

任何一个含有变量 A 的等式，如果将所有出现 A 的位置都代之以一个逻辑函数式，则等式仍成立。

(2) **对偶规则**

对于任何一个逻辑函数式 F,**若将其中的"·"换成"+","+"换成"·","1"换成"0","0"换成"1",则得到的一个新函数式** F^* **称为原函数式** F **的对偶函数式。**

(3) **反演规则**

对于任何一个逻辑函数式 F,**若将其中的"·"换成"+","+"换成"·","1"换成"0","0"换成"1",并将原变量换成反变量,反变量换成原变量,则得到的一个新的函数式**$\overline{F}$ **称为原函数式** F **的反函数。**

3) **基本定理**

(1) **吸收定理**

$$AB + A\overline{B} = A$$

$$A + AB = A$$

$$A + \overline{A}B = A + B$$

(2) **多余项定理**

$$AB + \overline{A}C + BC = AB + \overline{A}C$$

推论: $AB + \overline{A}C + BCDE\cdots = AB + \overline{A}C$

3 .**逻辑函数的标准形式**

1) **最小项表达式(标准与或式)**

在与或式中,每一个乘积项都包含了全部输入变量,每个输入变量或以原变量形式或以反变量形式在乘积项中出现,并且仅仅出现一次,这个逻辑式叫做最小项表达式或叫做标准与或式。

例如,对于三变量函数

$F_{(A,B,C)} = ABC + A\overline{B}C + \overline{A}B\overline{c}$ **这样的与或式叫做最小项表达式。**

$F_{(A,B,C)} = ABC + A\overline{B}C + \overline{A}\,\overline{c}$ **这个逻辑式不是最小项表达式,而是一般的与或式。**

为了便于叙述及使用方便,可以对每个变量的取值组合用一个最小项记号 m_i **来表示。对于** n **个变量必定有** 2^n **个最小项,最小项中以原变量形式出现记为** 1,**以反变量形式出现记为** 0,**则按** 1 **和** 0 **的顺序组成的二进制数对应的十进制数就是** i **值。**

例如: A, B, C 3 **个变量可组成** $2^3 = 8$ **个最小项。**$m_0 = \overline{A}\,\overline{B}\,\overline{c}$, $m_1 = \overline{A}\,\overline{B}C, \cdots, m_7 = ABC$。

对于前面的最小项表达式可表示为

$$F_{(A,B,C)} = ABC + A\overline{B}C + \overline{A}B\overline{c} = m_7 + m_5 + m_2$$

2) **最大项表达式(标准或与式)**

在或与式中,每个或项包含了全部变量,每个变量或以原变量形式或以反变量形式出现,并且仅仅出现一次,这个逻辑式叫做最大项表达式或叫做标准或与式。

最大项用 M_i **表示**, i **表示最大项的序号,与最小项不同的是,在确定** i **值时,原变量取值为** 0,**而反变量取值为** 1。

例如,最大项表达式

$$F_{(A,B,C)} = (A + B + C)(A + \overline{B} + C)(\overline{A} + B + C) = M_0 \cdot M_2 \cdot M_4$$

需要说明的是,在说最小项或最大项时,首先应明确是几变量函数,例如 AB **在两变量中是最小项,而在三变量中就属于一般项了。**

标准式具有惟一性,任何逻辑函数的标准式只有一个。同一个逻辑函数式可以表示成繁简不同、逻辑运算不同的多种形式,以便采用不同的器件实现其功能。

4 .卡诺图法化简逻辑函数

1) 掌握卡诺图的作图方法。

2) 熟悉卡诺图的性质。

3) 掌握利用卡诺图合并最小项的方法。

圈图方法如下:

(1) 被圈的方格数必须是 2^i 个。

(2) 每个小方格可以多次被圈,但每个圈内至少有一个小方格是初次被圈,否则会出现多余项。

(3) 所圈的圈要尽量的大,这样所包含的与项的个数少。

(4) 总圈数要少,这样与项少。

(5) 求最简与或式。

在卡诺图上通过圈 1 格合并化简。即在所圈的圈中,去掉不同的变量,保留相同的变量,且见 1 写原变量,见 0 写反变量,每个圈构成一个与项,将所有的与项相或,可得出原函数的最简与或式。

(6) 求最简或与式。

在卡诺图上通过圈 0 格合并化简。即在所圈的圈中,去掉不同的变量,保留相同的变量,且见 0 写原变量,见 1 写反变量,每个圈构成一个或项,将所有的或项相与,可得出原函数的最简或与式。

5 .具有无关最小项(任意项) 的逻辑函数

1) 任何一个 n变量的逻辑函数,总能用 m个最小项之和的形式来表示,这 m个最小项是使该函数式逻辑值为 1 的输入变量的取值。若剩下的 $2^n - m$ 个最小项使函数式的逻辑值为 0,这就表明此函数式与其 2^n 个最小项都有关,这一函数称为完全描述的逻辑函数。

2) 在某些实际应用中,一个 n变量的逻辑函数,并不是与它的 2^n 个最小项都有关,而是只与其中的一部分有关,与另一部分则无关系。这一部分无关的最小项并不决定函数的取值,故称为无关最小项(无关项),记为 d 或 ×。这类逻辑函数称为包含有无关项的逻辑函数,或称为具有约束条件的逻辑函数,也称为不完全描述的逻辑函数。

3) 无关项的出现通常有两种情况: 一是在某些实际问题中,加在逻辑电路上的输入变量的某些取值不可能或不允许出现,这种对于输入变量取值所加的限制称为约束,所对应的最小项称为约束项,它们构成了逻辑函数的约束条件; 二是输入变量的某些取值的出现,不会影响逻辑函数的有效取值,即不影响逻辑功能的实现。通常把这些输入变量的取值所对应的最小项称为任意项。

逻辑函数的约束项及任意项统称为无关最小项,可将它们随意地加到逻辑函数表达式中。在利用卡诺图化简逻辑函数时,可以把它们用符号 × 填入图中,并可随意地视为 1 格或 0 格参与化简,而使函数式化简为最简的形式,并不会影响该逻辑函数的实际功能。

1 .3 基本题解答

例 1 - 1 $(12.34)_{10} = (\qquad)_2 = (\qquad)_8 = (\qquad)_{16}$

分析:此题是将一个十进制数转换成二进制数,八进制数,十六进制数。而将十进制数转换

成任意 R 进制($R \in 2, 8, 16$) 的方法是将整数部分和小数部分分别进行转换,整数部分除 R 取余数,最先得到的是最低位,小数部分乘 R 取整数,最先得到的是最高位,然后再将转换后的整数部分和小数部分合并起来。

解

		余数	
2	12		
2	6	0	LSB
2	3	0	
2	1	1	
	0	1	MSB

	整数	
0.34×2		
$[0.]68$	0	MSB
$\times 2$		
$[1.]36$	1	
$\times 2$		
$[0.]72$	0	
$\times 2$		
$[1.]44$	1	
$\times 2$		
$[0.]88$	0	
$\times 2$		
$[1.]76$	1	
$\times 2$		
$[1.]52$	1	
$\times 2$		
$[1.]04$	1	LSB

所以 $(12.34)_{10} = (1100.01010111)_2$

同理:用 12.34 分别除以 8 或 16,乘以 8 或 16 可得八进制数和十六进制数。

所以 $(12.34)_{10} = (1100.010101110)_2 = (14.256)_8 = (C.57)_{16}$

例 1-2 $(3FCA)_{16} = (\quad\quad)_{10}$

分析:本题是把十六进制数转换成十进制数。其方法是按权展开再相加,十六进制数中的 A,B,C,D,E,F 分别用 10,11,12,13,14,15 代替。

解 $(3FCA)_{16} = 3 \times 16^3 + 15 \times 16^2 + 12 \times 16^1 + 10 \times 16^0 = (16330)_{10}$

例 1-3 $(73.26)_{10} = (\quad\quad)_{8421BCD}$

分析:此题是将十进制数转换成 8421BCD 码。

8421BCD 码是用 4 位二进制数表示 1 位十进制数,用 0000 ~ 1001 这十个数码分别表示十进制数 0 ~ 9。

解 $(73.26)_{10} = (01110011.00100110)_{8421BCD}$

例 1-4 $(31.67)_{10} = (\quad\quad)_{余3BCD}$

分析:余 3BCD 码 0011 ~ 1100 这十个数码分别表示十进制数 0 ~ 9,所以把十进制数转换成余 3BCD 的简便方法是在每位十进制数上加 3,再用相应的 4 位二进制数表示,例如 31.67 每位加 3:

$$3 + 3 = 6 \rightarrow 0110 \quad 1 + 3 = 4 \rightarrow 0100 \quad 6 + 3 = 9 \rightarrow 1001 \quad 7 + 3 = 10 \rightarrow 1010$$

解 $(31.67)_{10} = (01100100.10011010)_{余3BCD}$

例 1-5 $(1011000)_2 = (\quad\quad)_{8421BCD} = (\quad\quad)_{余3BCD}$