



基于Verilog HDL与 Cadence的数字系统 设计技术

解本巨 编著



清华大学出版社

基于Verilog HDL与 Cadence的数字系统 设计技术

解本巨 编著



清华大学出版社
北 京

内 容 简 介

本书是以 EDA 技术设计为出发点,专门针对各大高校信息、自动化、计算机专业在校学生和在公司中初始学习硬件技术的开发人员而编写的数字系统制版技术材料,主要目的是使读者克服学习硬件开发技术的困难,使学习硬件技术像学习软件技术一样简单。本书的技术开发以逻辑代数的运算、定理和化简方法为理论指导,研究原理图设计方法,引入 FPGA 的开发软件 Quartus II 9.1,在其中利用 Verilog HDL 设计实现电路常用芯片的开发,可以在不必了解芯片内部工作原理的基础上,通过程序设计者的硬件行为描述获得芯片及引脚的相关信息。本书选择 Cadence SPB 16.3 作为设计数字系统原理图和电路板的软件,用同一个数字系统实例说明设计原理图和制作电路板的连续过程:原理图→网络表→焊盘、封装、制版→导入网络表→布局→覆铜→布线→后处理→送厂家制版。

本书可以作为嵌入式数字系统开发的基础技术设计指导书,是硬件制版技术快速入门的绝佳教材,为更多的硬件技术设计爱好者提供了广阔的空间。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

基于 Verilog HDL 与 Cadence 的数字系统设计技术/解本巨编著.--北京:清华大学出版社,2013.5
ISBN 978-7-302-31470-7

I. ①基… II. ①解… III. ①数字系统—系统设计—高等学校—教材 IV. ①TP271

中国版本图书馆 CIP 数据核字(2013)第 024093 号

责任编辑:高买花 王冰飞

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:17

字 数:416 千字

版 次:2013 年 5 月第 1 版

印 次:2013 年 5 月第 1 次印刷

印 数:1~3000

定 价:29.50 元

FPGA、ARM 和单片机嵌入式开发在目前 EDA 设计技术中具有广泛的应用。FPGA 是以 Nios II 软核处理器为核心,通过 IP 核的配置实现微处理器对外部设备的直接控制与数据交换;而 ARM 是通过微处理器中的指令控制,通过接口与外部设备交换数据。FPGA 使用内部编程的方法代替 ARM 系统中接口的功能,极大地简化了硬件电路的设计。这种以软件设计代替硬件设计原理研究的方式,是本书嵌入式开发研究的创新点,极大地降低了 EDA 设计开发的难度,也为嵌入式开发打下了一个坚实的基础,扩大了嵌入式开发的适应人群。

Verilog HDL 是一种硬件描述语言,它是以 C++ 语法为基础的,用文本形式描述数字系统的行为实现过程。用它设计的常用组合、时序元件能够用程序再现电路设计过程和元件的工作原理,更重要的是能够展示芯片引脚的外围特性,为嵌入式数字系统的扩展提供有力的信息支持。本书以 Quartus II 9.1 作为 Verilog 程序编辑编译软件,能够自动生成等效的数字元件电路。Quartus II 也是 FPGA 设计开发的专用软件。

嵌入式开发分为印刷电路板制作和系统软件设计两个层面,而印刷电路板制作是嵌入式设计的最大难题,其主要原因是对数字元件设计不够熟练,对元件引脚外围特性在设计中的关联性作用无从入手。用 Verilog 设计芯片正是为了加强对外围特性的了解,从而建立芯片引脚间的逻辑关系,设计出完整的能够实现一定功能的数字系统。电路板制作首先要根据外围特性设计出数字系统的原理图,然后生成网络表,网络表中主要指出原理图中电子元器件对应的元件封装,然后导入网络表,在版图中对封装进行布局、覆铜、布线等操作。本书引进 Cadence 公司的 Allegro SPB 16.3 作为 EDA 开发工具,它提供了良好而且交互的工作接口和强大完善的功能,为当前高速、高密度、多层次的复杂 PCB 设计布局、布线提供了最完美的解决方案。

全书共分为 6 章。第 1 章是数字系统开发的理论基础,通过对逻辑运算、逻辑定理、卡诺图化简、逻辑符号的讲解,引出设计逻辑电路原理图的基本方法;通过逻辑关系分析实现组合逻辑电路、同步时序逻辑电路、异步时序逻辑电路和电路常用芯片的设计。第 2 章主要描述 Verilog 语言的语法及规则、建模和模块调用,然后讲述开发环境 Quartus II 9.1 及在其中实现原理图制作、Verilog 程序建模、模块调用与芯片工作过程仿真。第 3 章主要讲述数字系统设计过程中常用分立元件、IC 芯片、IO 芯片的原理、设计与应用。首先阐述常用分立元件电阻、电容、二极管、三极管的工作原理、元件外形和逻辑符号;接着说明设计电源电路和脉冲电路的方法;然后用 Verilog 建模实现常用的组合逻辑电路、时序逻辑电路芯片的设计。第 4 章主要讲述 Cadence SPB 16.3 的安装与简介以及绘制平面元件的方法,通过对十进制可逆计数器 74LS192 等芯片外围特性的分析,设计出 100 进制可逆计数器的原理图并进行绘制,最后生成元件对应的网络表。第 5 章主要介绍制作 100 进制可逆计数器系统电路板的详细过程。建立焊盘并对网络表中所有元件进行封装,建立电路板后导入网络

表并对元件进行手工布局,为 V_{CC} 和 GND 层进行覆铜,最后实现自动布线和布线优化。第 6 章主要是对设计好的电路板进行后续处理和电路板加工前的准备工作。后续处理包括测试点的添加和修改、重命名元件序号、调整文字面;而电路板加工前的准备工作主要实现可装配检查、丝印层的建立、为印制电路板厂商建立各种制版文件等。

本书是编者在实践过程中,根据数字系统和嵌入式系统设计的经验,为 EDA 设计爱好者编写的有关嵌入式设计技术的基础教程,是 IT 业硬件设计者快速入门的力作,旨在指导学习者像学习软件一样学习硬件技术,从而能够在短期内快速掌握数字系统的设计方法和电路板的制作过程。

本书所有设计过程通过一个综合实例来实现,具有学习的完整性和连续性,每一个设计都通过了编者的设计和验证,但不能完全排除某些小失误,敬请广大读者批评指正。

本书是嵌入式设计入门的实训教材,由刘国柱教授、孟友新教授、江守寰老师、曹玲老师、张会萍同学共同参与完成,感谢他们付出的辛勤劳动,同时也感谢清华大学出版社编辑刘向威为本书所做的工作。

编 者

2013 年 3 月

目 录

第 1 章 数字系统设计基础	1
1.1 逻辑代数	1
1.1.1 逻辑运算	1
1.1.2 逻辑定理与化简	5
1.1.3 卡诺图化简	7
1.2 组合逻辑电路设计	13
1.2.1 组合逻辑电路设计方法	13
1.2.2 3 线-8 线译码器设计	15
1.2.3 8 路数据选择器设计	17
1.2.4 七段显示译码器设计	18
1.3 时序逻辑电路设计	20
1.3.1 时序电路的描述方法	21
1.3.2 触发器	22
1.3.3 同步时序逻辑电路设计方法	25
1.3.4 异步时序逻辑电路设计方法	27
1.3.5 十进制加法计数器设计	27
1.3.6 寄存器设计	32
1.4 基于 Nios II 的 FPGA 技术	35
1.4.1 FPGA 简介和工作原理	36
1.4.2 Nios II 软核处理器	37
第 2 章 硬件描述语言 Verilog HDL 与集成开发环境	39
2.1 Verilog 语言简介	39
2.2 Verilog HDL 语法规则	39
2.2.1 标识符	40
2.2.2 命令语句格式	40
2.2.3 数字值集合	41
2.2.4 变量与数据类型	42
2.2.5 运算符与表达式	44
2.2.6 结构语句	47
2.3 Verilog HDL 建模	50
2.3.1 模块结构	50

2.3.2	时延	53
2.3.3	3 种建模方式	53
2.3.4	模块调用	58
2.4	编辑环境 Quartus II 9.1 与应用	60
2.4.1	Quartus II 9.1 安装与编辑环境介绍	61
2.4.2	原理图绘制	67
2.4.3	使用 Verilog 语言实现电路设计	74
2.5	Quartus II 9.1 实现电路输出仿真	77
第 3 章	数字系统常用元件及实现	84
3.1	常用分立元件及电路	84
3.1.1	常用分立元件	84
3.1.2	电源电路的实现与设计	92
3.1.3	脉冲时序发生电路设计	93
3.2	组合元件的 Verilog 设计	95
3.2.1	数据通路的设计	95
3.2.2	运算电路的设计	98
3.3	时序元件的 Verilog 设计	101
3.3.1	触发器的设计	101
3.3.2	计数器的设计	105
3.3.3	寄存器的设计	107
第 4 章	基于 Cadence PCB 的数字系统原理图设计	110
4.1	Cadence SPB 16.3 安装与简介	110
4.1.1	Cadence SPB 16.3 破解安装步骤	110
4.1.2	Cadence SPB 16.3 简介	112
4.2	创建平面元件	119
4.3	原理图设计	123
4.3.1	绘制原理图	123
4.3.2	原理图后续处理	137
第 5 章	PCB 电路板制作	144
5.1	Allegro 工作环境配置	144
5.1.1	整体绘图参数设置	144
5.1.2	颜色的设置	147
5.1.3	格点参数设置	148
5.1.4	子集(层)选项设置	148
5.1.5	盲孔和埋孔的设置	148
5.1.6	自动保存功能设置	151

5.2	焊盘与 PCB 封装的建立	151
5.2.1	创建焊盘	152
5.2.2	创建元件封装符号	165
5.2.3	100 进制可逆计数器电路焊盘的设计	175
5.2.4	100 进制可逆计数器电路封装设计	177
5.3	电路板建立与设计规则的设置	182
5.3.1	使用电路板向导建立电路板	183
5.3.2	手动建立电路板	185
5.3.3	导入网络表	189
5.4	布局	192
5.4.1	电路板的规划	192
5.4.2	元件的手工摆放	196
5.4.3	元件的快速摆放	199
5.4.4	生成报告文件	200
5.4.5	手工布局 100 进制可逆计数电路板	200
5.4.6	自动布局	202
5.5	覆铜	205
5.5.1	基本概念	205
5.5.2	为平面层建立覆铜区域	207
5.5.3	平面层分割	209
5.5.4	覆铜的编辑操作	213
5.6	布线	213
5.6.1	布线的原则	213
5.6.2	手动布线	214
5.6.3	自动布线	217
5.6.4	扇出布线	226
5.6.5	布线优化	226
第 6 章	电路板加工前的处理工作	229
6.1	PCB 后续处理	229
6.1.1	自动测试点的添加与修改	229
6.1.2	重命名元件序号	232
6.1.3	调整文字面	233
6.2	电路板加工前的准备工作	234
6.2.1	设计的可装配性检查	234
6.2.2	建立丝印层	237
6.2.3	生成报告文件	238
6.2.4	建立和查看底片文件	238
6.2.5	向厂商提供文件	249

附录 A	Cadence 元件库介绍	250
附录 B	DRC 检测常见错误	252
附录 C	74 系列数字集成电路型号功能表	256
附录 D	CMOS 系列数字集成电路型号功能表	260
参考文献		263

第1章

数字系统设计基础

进行数字系统的开发,需要利用数字逻辑电路的设计技术,绘制出相关的电路原理图,进一步转换为能够提供给厂家的印制电路板。在设计中选择使用的芯片能够在市场上购买到。无论如何,开发过程中,掌握数字逻辑相关的基础知识和设计电路的基本方法是成为一个优秀的 EDA 设计者的先决条件。数字系统严格来说是由一门逻辑数学构成的综合体,逻辑数学这里称为逻辑代数,它的逻辑变量有别于算术运算的数值变量,具有二值逻辑 0、1 的特点,逻辑值与电路中的逻辑电平建立一一对应的关系,逻辑电路在这种关系中得以体现。用 1 表示高电平、0 表示低电平称为正逻辑表示法,反之称为负逻辑表示法,本书采用正逻辑表示。

1.1 逻辑代数

逻辑代数是分析和设计电路的基础,是由英国科学家乔治·布尔(George Boole)于 1849 年创立的,故又称布尔代数,主要描述从一种逻辑(输入)向另一种逻辑(输出)的转换,从而得出设计基本逻辑电路(芯片)的基本方法。逻辑代数中的变量取值只有两个,即逻辑 0 和逻辑 1,也称之为逻辑常量,它们不表示数字的大小,只表示两种对立的逻辑状态。变量通过逻辑关系构成的表达式称为逻辑函数,例如 $Y=F(A,B,C,\dots)$,其中 F 为输出函数, A 、 B 、 C 、 $\dots$ 为输入变量, Y 是输出变量。

1.1.1 逻辑运算

逻辑关系相当于算术运算中的运算关系,逻辑运算符相当于算术运算中的算术运算符。逻辑代数的基本逻辑运算有:逻辑与(乘)、逻辑或(加)、逻辑非,其他逻辑运算由这 3 种运算复合而成。

1. 与运算

与逻辑的定义:仅当决定事件(F)发生的所有条件($A,B,C,\dots$)均满足时,事件(F)才能发生。表达式为:

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

式中小圆点表示书写的与运算符。与运算符可省略,即:

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

1) 与运算规则

$$0 \cdot 0 = 0; 0 \cdot 1 = 1 \cdot 0 = 0; 1 \cdot 1 = 1$$

2) 与运算的逻辑符号

任何逻辑运算要用一定的逻辑符号来表示,逻辑关系才能得到分析和应用,逻辑符号也是开发电路原理图的组成部分。本书采用国际 IEEE 标准通用逻辑符号表示逻辑关系。对于与运算表达式 $F = A \cdot B$,其逻辑符号如图 1-1 所示,其中,AND2 代表与门的逻辑符号名称,即 2 引脚输入的与门;inst 是对与门引用的实例化名称,是可以定义和修改的。

2. 或运算

或逻辑的定义:当决定事件(F)发生的各种条件(A,B,C,...)中,只要有一个或多个条件具备,事件(F)就发生。表达式为:

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

式中加号表示书写用的或运算符。

1) 或运算规则

$$0 + 0 = 0; 0 + 1 = 1 + 0 = 1; 1 + 1 = 1$$

2) 或运算的逻辑符号

对于或运算表达式 $F = A + B$,其逻辑符号如图 1-2 所示,在绘制原理图软件中的逻辑符号名称为 OR2,即 2 引脚输入的或门。

3. 非运算

非逻辑指的是逻辑的否定。当决定事件(F)发生的条件(A)满足时,事件不发生;条件不满足时,事件反而发生。表达式为:

$$F(A) = \bar{A}$$

在绘制电路原理图时,其逻辑符号如图 1-3 所示,其中 NOT 为非门逻辑符号名称。

非运算的运算规则为:

$$\bar{0} = 1 \quad \bar{1} = 0$$



图 1-1 二输入变量的与门国际
准逻辑符号



图 1-2 二输入变量的或门国际
准逻辑符号



图 1-3 非门的国际标
准逻辑符号

4. 常见复合运算

利用与、或、非 3 种基本运算可以构造出具有实际用途的复合运算。常见的复合运算有与非、或非、异或和同或运算等。

1) 与非运算

与非运算是两个或两个以上的变量先进行与运算再把计算结果取非的逻辑过程。根据与非运算的运算规则,只有所有变量都取值为 1,输出的逻辑结果为 0,否则输出的逻辑结果

应该为 1。如果只输入两个变量 A、B,则其表达式为:

$$F(A,B) = \overline{AB}$$

其逻辑符号如图 1-4 所示,其中 NAND2 是二输入与非门的逻辑符号名称。

逻辑运算的逻辑计算结果可以通过真值表来体现。真值表是一种能同时体现输入变量各种取值和输出变量计算结果情况的表格。与非运算的真值表如表 1-1 所示。

表 1-1 与非运算的真值表

A	B	F(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



图 1-4 二输入变量与非门的国际标准逻辑符号

2) 或非运算

或非运算是由或运算和非运算组合在一起的一种运算。逻辑表达式如下:

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

或非运算的真值表如表 1-2 所示。

二输入变量的或非门的国际标准逻辑符号如图 1-5 所示。NOR2 为元件逻辑符号标识名称。

表 1-2 或非运算的真值表

A	B	F(A,B)
0	0	1
0	1	0
1	0	0
1	1	0



图 1-5 二输入变量的或非门的国际标准逻辑符号

3) 异或运算

当两个输入逻辑变量 A、B 不同时,输出 F 为 1;当两个输入逻辑变量 A、B 相同时,输出 F 为 0,这种逻辑运算称为异或运算。其逻辑表达式为:

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

⊕是异或运算符,真值表描述的数据运算情况如表 1-3 所示。

异或门的国际标准逻辑符号如图 1-6 所示,图中 XOR 是该逻辑符号在电路设计中的符号名称。

表 1-3 异或运算的真值表

A	B	F(A,B)
0	0	0
0	1	1
1	0	1
1	1	0



图 1-6 异或门国际标准逻辑符号

4) 同或运算

当两个输入逻辑变量 A、B 相同时,输出 F 为 1; 当两个输入逻辑变量 A、B 不同时,输出 F 为 0,这种逻辑运算称为同或运算。其逻辑表达式为:

$$F(A,B) = AB + \overline{A}\overline{B} = A\odot B$$

⊙ 是同或运算符,真值表描述的数据运算情况如表 1-4 所示。

同或门的国际标准逻辑符号如图 1-7 所示,图中 XNOR 是该逻辑符号在电路设计中的符号名称。

表 1-4 同或运算的真值表

A	B	F(A,B)
0	0	1
0	1	0
1	0	0
1	1	1



图 1-7 同或门国际标准逻辑符号

5. 逻辑函数及其表示方法

1) 逻辑函数

可以看到,逻辑表达式是由逻辑变量和与、或、非 3 种运算符连接起来所构成的式子。在逻辑表达式中,等式右边的字母 A、B、C、D 等称为输入逻辑变量,等式左边的字母 Y 称为输出逻辑变量,字母上面没有非运算符的叫做原变量,有非运算符的叫做反变量。

逻辑函数是指如果对应于输入逻辑变量 A、B、C、…的每一组确定值,输出逻辑变量 Y 就有唯一确定的值,则称 Y 是 A、B、C、…的逻辑函数。记为

$$Y=F(A,B,C,\cdots)$$

与普通代数不同的是,在逻辑代数中,不管是变量还是函数,其取值都只能是 0 或 1,并且这里的 0 和 1 只表示两种不同的状态,没有数量的含义。

2) 逻辑函数的表示方法

一个逻辑函数是一个输入与输出之间的关系表达式,根据情况不同,需要用不同描述法来表示它。逻辑函数的描述方法有真值表、逻辑函数表达式、逻辑图、卡诺图、波形图等。逻辑表达式、真值表描述法在前面已有所讲述,在这里只介绍逻辑图表示法和波形图表示法。

(1) 逻辑图表示法。

在已知逻辑函数基础上,用与、或、非、异或等逻辑运算符号表示逻辑函数中各变量之间的逻辑关系的图形称为逻辑图。已知逻辑表达式:

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

则用与、或、非逻辑运算符号表示出的逻辑图如图 1-8 所示。

首先画出 A 的非符号,输出为 F₁,画出 B 的非符号,输出为 F₂,作为第一级;然后画出 AB 与门符号,设输出为 F₃,同时画出 $\overline{A}\overline{B}$ 的与门符号,设输出为 F₄,F₃、F₄ 作为第二级;最后画出 F₃+F₄ 的或门符号,作为第三级。

(2) 波形图表示法。

用输入端在不同逻辑信号作用下所对应的输出信号的波形图表示电路的逻辑关系。

图 1-9 是图 1-8 所示逻辑图的波形图。

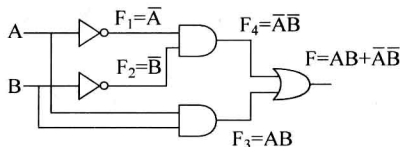


图 1-8 函数的逻辑图表示法

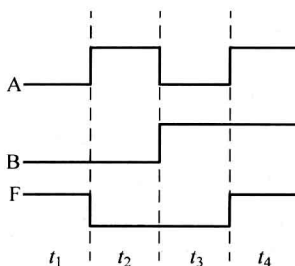


图 1-9 波形图表示法

图 1-9 中 4 个 t 时刻 F 的输出值分别对应 A 、 B 输入的 4 种组合值。

1.1.2 逻辑定理与化简

逻辑代数具有一定的公式、定律和规则,用它们对逻辑表达式进行处理,可以完成对逻辑电路的化简、分析、设计、表达电路意图等。

1. 逻辑代数的基本公式、定律

利用与、或、非 3 种基本逻辑运算和恒等式代入规则可以证明下面公式或定律两边是成立的。

(1) 0-1 律。即数值 0、1 和变量的与、或运算: $A+0=A$; $A+1=1$; $A \cdot 0=0$; $A \cdot 1=A$ 。

(2) 重叠律。即变量自身的与、或运算: $A+A=A$; $A \cdot A=A$ 。

(3) 互补律。即变量与反变量的与、或运算: $A+\bar{A}=1$; $A \cdot \bar{A}=0$ 。

(4) 自补律。 $\bar{\bar{A}}=A$ 。

(5) 结合律。3 个变量两两组合的与、或运算: $(A+B)+C=A+(B+C)$; $ABC=A(BC)$ 。

(6) 交换律。两个变量之间与、或运算位置互换: $A+B=B+A$; $AB=BA$ 。

(7) 分配律。3 个变量与或混合运算重新分配: $A(B+C)=AB+AC$; $A+BC=(A+B)(A+C)$ 。

(8) 反演律。即摩根定理: $\overline{A+B+C+\dots}=\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots$; $\overline{ABC\dots}=\bar{A}+\bar{B}+\bar{C}+\dots$ 。

(9) 吸收率。可由以上定律推导: $A+A \cdot B=A$; $A \cdot (A+B)=A$; $A+\bar{A}B=A+B$ 。

(10) 冗余律。 $AB+\bar{A}C+BC=AB+\bar{A}C$; $(A+B)(\bar{A}+C)(B+C)=(A+B)(\bar{A}+C)$ 。

以上定律的证明可以采用真值表形式,把输入变量的取值组合一一列出,然后求出左、右两边表达式的值,如果都相等则定律成立,例如摩根定律。

【例 1-1】 一些定律可以利用其他定律来证明,试证明: $AB+\bar{A}C+BC=AB+\bar{A}C$ 。

证明: 左式 $= AB+\bar{A}C+BC = AB+\bar{A}C+(A+\bar{A})BC$
 $= AB+\bar{A}C+ABC+\bar{A}BC$
 $= AB(1+C)+\bar{A}C(1+B)$
 $= AB+\bar{A}C$
 $= \text{右式}$

这个定律在以后化简逻辑函数中通过添加或吸收多余项来实现。

2. 逻辑代数的基本规则

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 的位置都用同一个逻辑函数代替,则等式仍然成立,这个规则称为代入规则。

【例 1-2】 已知等式 $AB = \bar{A} + \bar{B}$,用函数 $Y = AC$ 代替等式中的 A ,根据代入规则,等式仍然成立,即有: $(AC)B = \overline{AC} + \bar{B} = \bar{A} + \bar{B} + \bar{C}$ 。

代入规则可以扩展所有基本定律的应用范围,即把某变量变换为多个其他变量的表示并代入定律中,定律仍然成立。

2) 反演规则

对于任何一个逻辑表达式 Y ,如果将表达式中的所有“ $\cdot$ ”换成“ $+$ ”、“ $+$ ”换成“ $\cdot$ ”、0 换成 1、1 换成 0、原变量换成反变量、反变量换成原变量,那么所得到的表达式就是函数 Y 的反函数 $\bar{Y}$ (或称补函数)。这个规则称为反演规则。

【例 1-3】 如对于下列表达式:

$$Y = A\bar{B} + C\bar{D}E$$

采用反演规则,则 Y 的反变量为: $\bar{Y} = (\bar{A} + B)(\bar{C} + D + \bar{E})$ 。

这里提供了一种求输出反变量的好方法。

3) 对偶规则

对于任何一个逻辑式 Y ,若将其中所有的“ $+$ ”换成“ $\cdot$ ”、“ $\cdot$ ”换成“ $+$ ”、1 换成 0、0 换成 1,则得到的表达式称为 Y 的对偶式,记做 Y' 。

【例 1-4】 如果 $Y = A + AB$,则其对偶式为 $Y' = A \cdot (A + B)$ 。

若两逻辑式相等,则它们的对偶式也相等。这就是对偶规则。

【例 1-5】 对于下面恒等式:

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

则其对偶式为 $(A+B)(\bar{A}+C)(B+C) = (A+B)(\bar{A}+C)$ 。

3. 逻辑代数的公式化简法

数字逻辑电路在设计过程中需要得出逻辑表达式,并转换为逻辑图或逻辑电路原理图。电路对应的逻辑表达式不一定是简表达式,但仍需要对逻辑表达式进行化简,利用化简后的逻辑表达式构成的逻辑电路,使用的元件成本最低,提高了数字系统的可靠性。

最简表达式如与、或、非、与非、或非、与或非、或与、非、异或等。

运用基本公式和常用公式来化简逻辑函数的方法,主要有以下几种:

1) 并项法

利用公式 $A + \bar{A} = 1$,将两项合并为一项,并消去一个变量。

【例 1-6】 利用分配率,化简 Y 表示的表达式:

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

2) 吸收法

(1) 利用公式 $A+AB=A$, 消去多余的项。

【例 1-7】化简 Y 表示的表达式:

$$Y = \bar{A}B + \bar{A}BCD(E+F) = \bar{A}B$$

【例 1-8】利用摩根定律和吸收法, 化简 Y 表示的表达式:

$$\begin{aligned} Y &= A + \overline{B+CD} + \overline{ADB} = A + BCD + AD + B \\ &= (A+AD) + (B+BCD) = A+B \end{aligned}$$

(2) 利用公式 $A+\bar{A}B=A+B$, 消去多余的变量。

【例 1-9】化简 Y 表示的表达式:

$$\begin{aligned} Y &= AB + \bar{A}C + \bar{B}C \\ &= AB + (\bar{A} + \bar{B})C \\ &= AB + \bar{A}\bar{B}C \\ &= AB + C \end{aligned}$$

3) 配项法

(1) 利用公式 $A=A(B+\bar{B})$, 为某一项配上其所缺的变量, 以使用其他方法进行化简。

【例 1-10】化简 Y 表示的表达式:

$$\begin{aligned} Y &= \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{B}C + \bar{A}B \\ &= \bar{A}\bar{B} + \bar{B}\bar{C} + (A+\bar{A})\bar{B}C + \bar{A}B(C+\bar{C}) \\ &= \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}BC + \bar{A}B\bar{C} \\ &= \bar{A}\bar{B}(1+C) + \bar{B}\bar{C}(1+\bar{A}) + \bar{A}C(\bar{B}+B) \\ &= \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}C \end{aligned}$$

(2) 利用公式 $A+A=A$, 为某项配上其所能合并的项。

【例 1-11】化简 Y 表示的表达式:

$$\begin{aligned} Y &= ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC \\ &= (ABC + AB\bar{C}) + (ABC + A\bar{B}C) + (ABC + \bar{A}BC) \\ &= AB + AC + BC \end{aligned}$$

4) 消去冗余项法

利用冗余律 $AB+\bar{A}C+BC=AB+\bar{A}C$, 将冗余项 BC 消去。

【例 1-12】化简 Y 表示的表达式:

$$\begin{aligned} Y &= \bar{A}\bar{B} + AC + ADE + \bar{C}D \\ &= \bar{A}\bar{B} + (AC + \bar{C}D + ADE) \\ &= \bar{A}\bar{B} + AC + \bar{C}D \end{aligned}$$

1.1.3 卡诺图化简

利用公式化简可以使逻辑函数变为较为简单的形式, 但得到的逻辑表达式是否为最简则很难判断, 使用卡诺图化简法可以很好地解决这个问题。

1. 最小项的定义及其性质

1) 最小项定义

如果一个函数的某个与运算项包含了函数的全部变量, 其中每个变量都以原变量或反

变量的形式出现,且仅出现一次,则这个乘积项称为该函数的一个标准积项,通常称为最小项。

3 个变量 A、B、C 可组成 8 个最小项。

(1) $\overline{A}\overline{B}\overline{C}$: 把 $ABC=000$ 带入最小项,输出为 1,表示此项为逻辑函数有效项;反之,如果已知某最小项取值为 $ABC=000$,其为逻辑函数中有效项,则可在逻辑函数中用 $\overline{A}\overline{B}\overline{C}$ 表示。

(2) $\overline{A}\overline{B}C$: 表示当 $ABC=001$ 时,此项为逻辑函数有效项,输出为 1。

(3) $\overline{A}B\overline{C}$: 表示当 $ABC=010$ 时,此项为逻辑函数有效项,输出为 1。

(4) $\overline{A}BC$: 表示当 $ABC=011$ 时,此项为逻辑函数有效项,输出为 1。

(5) $A\overline{B}\overline{C}$: 表示当 $ABC=100$ 时,此项为逻辑函数有效项,输出为 1。

(6) $A\overline{B}C$: 表示当 $ABC=101$ 时,此项为逻辑函数有效项,输出为 1。

(7) $AB\overline{C}$: 表示当 $ABC=110$ 时,此项为逻辑函数有效项,输出为 1。

(8) ABC : 表示当 $ABC=111$ 时,此项为逻辑函数有效项,输出为 1。

一般 n 个变量的最小项应有 2^n 个。

2) 最小项的表示方法

通常用符号 m_i 来表示最小项。下标 i 的确定:把最小项中的原变量记为 1,反变量记为 0,当变量顺序确定后,可以按顺序排列成一个二进制数,则与这个二进制数相对应的十进制数就是这个最小项的下标 i 。

3 个变量 A、B、C 的 8 个最小项可以分别表示为:

$$m_0 = \overline{A}\overline{B}\overline{C}, \quad m_1 = \overline{A}\overline{B}C, \quad m_2 = \overline{A}B\overline{C}, \quad m_3 = \overline{A}BC$$

$$m_4 = A\overline{B}\overline{C}, \quad m_5 = A\overline{B}C, \quad m_6 = AB\overline{C}, \quad m_7 = ABC$$

3) 最小项的性质

为了分析最小项的性质,下面列出 3 个变量所有最小项的真值表,如表 1-5 所示。

表 1-5 三变量最小项真值表

A	B	C	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

观察该表,可得出最小项的下列性质:

(1) 任意一个最小项,只有一组变量取值使其值为 1。

(2) 对于输入变量的任意一组取值,任意两个不同的最小项的乘积必为 0。

(3) 对于输入变量的任意一组取值,全部最小项的和必为 1。