

数字电路与微处理机

SHUZI DIANLU YU
WEICHULIJI



[美] H·陶布 著

印丕勤 陆嘉宝 刘 舒 译

印丕勤 校

中国建筑工业出版社

数字电路与微处理机

〔美〕 H·陶布 著

印丕勤 陆嘉宝 刘 舒 译

印 丕 勤 校

中国建筑工业出版社

本书系统地讲述数字电路和逻辑设计的基本原理,以8080型机为实例讲述微处理机和基于微处理机的系统,把数字电路与微处理机作了有机的联系。主要内容包
括:逻辑代数,逻辑函数,基本组合电路,触发器,寄存器和计数器,运算,存储器,时序电路,控制器,计算机,微处理机,输入输出操作等。

本书可作大专院校有关专业《数字电路》课程的教材或教学参考书,可供有关工程技术人员阅读参考。

责任编辑:刘家屿

Herbert Taub
DIGITAL CIRCUITS AND MICROPROCESSORS
Mc Graw-Hill Book Company 1982

• • •
数字电路与微处理机

印丕勤 陆嘉宝 刘 舒 译

印丕勤 校

中国铁道出版社出版 (北京西郊百万庄)

新华书店北京发行所发行 各地新华书店经售

河北省固安县印刷厂印刷

开本: 787×1092毫米 1/16,印张: 25³/₄ 字数: 624 千字

1987年 12 月第一版 1987年 12 月第一次印刷

印数: 1—6,440 册 定价5.95 元

ISBN7—112—00244—3/TP·1

统一书号: 15040·5291

译者序

本书是美国一些大学（如加州大学柏克莱分校等）电机和计算机科学系所采用的教材，类似于我国大学的电机和计算机科学系的《数字电路》教材。该课程是该系有关领域的一门核心课程。

本书有如下几个特点：

1. 从讲述通常的数字电路开始，过渡到讲述微处理机。使数字电路与微处理机作了有机的联系，避免了脱节。

2. 以8080微处理机作为实例简要地讲述微处理机，既使学生接触了实际的微处理机，又能为学生进一步深入学习其它较复杂的微处理机和微计算机打下良好的基础。

3. 尽可能避免讲述对一般工程技术人员不必详细了解的电子电路。

4. 某些方面的讲述较为详尽，分析较为透彻。

因此，将此书译出供有关专业的师生及工程技术人员参考。对所发现的原书的错误及不妥之处已作了改正，不另加注。

本书的前言、第4章至第8章（包括习题）和附录由上海建筑材料工业学院印丕勤翻译，第1章至第3章（包括习题）由上海建筑材料工业学院陆嘉宝翻译，第9章到第11章（包括习题）由北方交通大学刘舒翻译。全书译稿由印丕勤审校。

由于译者水平有限，错误和缺点在所难免，望读者批评指正。

译者

一九八六年七月

前 言

本书是一本入门教材,适用于一学期的课程。它包括数字系统和逻辑设计的全部基本原理,对微处理机和基于微处理机的系统也作了初步介绍。由于微处理机有着日益增长的重要地位,所以应尽早地在一门工程或者计算机科学课程中介绍这些具有多方面用途的微处理机。

第1章,讲述逻辑变量和布尔代数,描述和分析逻辑门及逻辑连接词,主要为了使真值表能够系统化,所以介绍了二进制数制。第2章,讲述逻辑函数的标准形式和卡诺图。第3章,讲述包括译码器、编码器、代码转换器、多路转换器和多路分配器的基本组合逻辑电路,要强调的是,可以买到所有这些元件的集成电路芯片,因此,简单介绍集成电路的族。解释集成电路芯片上控制端的表征惯例,这个讨论导致研究在某些应用中正在普及的混合逻辑。列举了一些目前表示组合元件和其它元件的较新的逻辑符号的例子。第4章,详细地讲述基本的存储元件触发器,对锁存器和触发器作了仔细的区分。讲述为了在同步系统中能正确地工作的触发器所需要的特性,也讲述用触发器组成的存储寄存器、移位寄存器和计数器。要强调的是,也可买到本章中所述元件的集成电路芯片,本章举了这些集成电路芯片的例子。第5章,讲述算术运算,主要介绍加法运算,阐明和分析先行进位原理。第6章,讲述存储器,包括静态和动态RAM、ROM、PLA、串行存储器和大容量存储器,也分析了读存储器和写存储器的定时要求。第7章,介绍同步型和基本型时序系统的分析和设计,描述了流程图、状态图和状态表的概念,也分析了消除冗余状态的方法。在第4章里,对移位寄存器和计数器等时序电路作了一些讨论,而第7章则是正式地、有条理地、有系统地讲述时序电路。

第8章,讲述控制器,目的是为了讲述微处理机。微处理机是由若干存储寄存器和工作寄存器、ALU及控制器组成的。控制器似乎具有神秘的功能,它精确地按照确切的顺序、在确切的时刻正确无误地做确切的工作,完成一个任务后准确地做下一个任务。事实上,控制器只不过是一种专用的时序电路,并没有什么难以理解的概念。但对于初学者来说,对控制器没有清晰的概念仍然不可避免地是一种忧虑的根源。讲述控制器,如果没有一个专门的、具体的范例是很难使学生接受控制器的一般概念的,因此,本章以下列方式编写,希望能使初学者树立起掌握控制器的信心。在本章的一开始,介绍的所有可能的数字操作是相当少的并且是极简单的,所有数字操作的执行是响应于逻辑门的启动,即逻辑门的置“1”,以使读者感到简明易懂。接着,介绍一个需要用控制器的很简单的数字系统的结构。详细地设计了一个控制器,先用第7章介绍的最少状态数设计法设计一个控制器,接着用移位寄存器式控制器替代那个原先的控制器。移位寄存器式控制器使用较多的硬件,但是有一个很大的优点,即它的工作细节是一目了然的,为了更复杂的工作所需要的修改几乎凭检验就能完成。最后,详细地设计了一个具有简单思维能力(四条指令)的“计算机”的控制器。这个设计,毫不含糊地,清晰地说明了怎样改进控制器的性

能,以适应一条指令的要求。在本章,学生也遇到了**程序计数器、存储器地址寄存器和指令寄存器**的概念。读者能够简明地了解表征微处理机的全部结构,了解存储指令和存储所存程序计算用的数据的存储器的典型内容。

第9章也是为了讲述微处理机而写的。本章介绍一台简单(16条指令)计算机的结构,使读者对在更复杂的系统中会遇到的指令型式有初步的了解。介绍了**转移和子程序调用指令**,用汇编语言写了几个简明的程序。本章也介绍了用**微程序设计**的控制的问题,举了些简单的例子。

某些作者,虚构一台假想的微处理机作为例子讲述微处理机。这种处理方法有一个致命的缺点,即学生失去了接触实际使用的微处理机的机会,而该假想的微处理机实际上几乎与实际使用的微处理机一样复杂。另一些作者,在描述和解释微处理机时罗列了若干实际使用的微处理机,这种处理方法,也常常使学生产生模糊的概念。广泛使用的第三种方法,也是本教材所采用的方法,是集中介绍一台简单的、实际使用的微处理机。这种处理方法,是通过分析一台专门的、实际使用的微处理机,使学生能很好地熟悉该台微处理机,为更好地理解其它的微处理机打下基础。本教材是选用众所周知的、并被高度重视的8080微处理机来讲述微处理机的。纵然8085微处理机改进了8080微处理机,但本教材仍讲述8080微处理机,因为它较简单,宜作为初步介绍的微处理机。第10章讲述8080微处理机的结构、指令和程序设计。第11章讲述8080微处理机的输入输出操作。

本教材包括了某些在一学期中不能讲完的内容。从作者的不成熟的观点看来,本教材可以安排在一个整学期加上第二个学期的大约五分之一时间内讲完。因而,在第二个学期的其余时间应当采用一本新教材,以深入一步地讲述微处理机和微计算机。然而,如省略某些在初次学习时不是基本内容的章节,在一学期中讲完本教材也是完全可行的。可省略的章节是:1.17, 1.25, 1.26, 2.12, 5.10至5.12, 6.10至6.17, 7.6至7.9, 7.11至7.19, 8.12和8.13。

本教材提供了大量的课外作业,还备有题解。

纽约城市学院电气工程系主任M·贾维特教授阅读了本书的大部分手稿,并且作了很多有价值的建议,我在此表示感谢。J·L·陶布先生在我准备手稿的过程中提供了大量的很有效的帮助,我在此表示最真诚的感谢。J·鲁宾夫人熟练地打印了手稿,也在此表示感谢。

H·陶布

目 录

第 1 章 逻辑变量的代数	1
1.1 变量和函数	1
1.2 逻辑变量	1
1.3 逻辑变量的值	2
1.4 单个逻辑变量的函数	3
1.5 两个逻辑变量的函数	3
1.6 “或”函数	5
1.7 逻辑系统的实现	5
1.8 逻辑变量的电压表示法	6
1.9 逻辑“非”	7
1.10 0、1表示法	8
1.11 二进制数制	9
1.12 十进制数与二进制数之间的转换	10
1.13 八进制和十六进制数制	11
1.14 二进制数和逻辑变量	12
1.15 布尔代数的定理	13
1.16 狄·摩根定理	15
1.17 维恩图	16
1.18 二变量函数	19
1.19 “异”	19
1.20 “与非”和“或非”函数	20
1.21 隐含函数	22
1.22 各种运算之间的关系	22
1.23 运算的充分性	23
1.24 “与非”运算的充分性和“或非”运算的充分性	23
1.25 应用布尔代数定理的实例	24
1.26 补充的实例	26
1.27 逻辑图	28
1.28 数字码	29
1.29 术语	30
1.30 数据代码	30

第 2 章 逻辑函数	32
2.1 逻辑函数的标准形式: 标准的积之和	32
2.2 标准的和之积	33
2.3 最小项和最大项的编号	35
2.4 用最小项和最大项说明函数	35
2.5 最小项、最大项和真值表之间的关系	36
2.6 二级门装置	37
2.7 用一种类型门的装置	39
2.8 卡诺图	43
2.9 用卡诺图简化逻辑函数	46
2.10 其它的逻辑相邻	47
2.11 卡诺图上较大的组合	49
2.12 五变量和六变量卡诺图	51
2.13 卡诺图的使用法	53
2.14 不是用最小项表示的函数的作图法	58
2.15 不完全确定的函数	60
第 3 章 基本组合电路	61
3.1 概述	61
3.2 逻辑电路的族	62
3.3 TTL族	63
3.4 CMOS族	66
3.5 ECL族	66
3.6 封装	67
3.7 逻辑操作开关	67
3.8 逻辑操作开关“或”门及线“与”接法	68
3.9 图腾柱输出	71
3.10 三态输出	72
3.11 IC门的实例	73
3.12 控制端符号: 混合逻辑	75
3.13 控制信号的符号	78
3.14 适用于门电路装置的混合逻辑	79
3.15 译码器	81
3.16 编码器	84
3.17 代码转换器	87
3.18 多路转换器	89
3.19 关于集电极开路和三态输出的多路转换	92
3.20 分成多路处理	95

第 4 章 触发器、寄存器和计数器	96
4.1 概述	96
4.2 由“或非”门组成的锁存器	97
4.3 由“与非”门组成的锁存器	99
4.4 无抖动开关	101
4.5 门控锁存器	102
4.6 定时	103
4.7 锁存器作为存储元件应用的局限性	105
4.8 主从触发器	107
4.9 触发器的定时图	109
4.10 两相定时	110
4.11 JK触发器	111
4.12 主从触发器捕获1的特性	113
4.13 边沿触发的JK触发器	114
4.14 D型触发器	116
4.15 建立、保持和传播延迟时间	119
4.16 寄存器到寄存器传送	120
4.17 移位寄存器	122
4.18 移位寄存器的其它特点和用途	124
4.19 计数器	127
4.20 环形计数器	128
4.21 尾变(约翰逊)计数器	129
4.22 其它的同步计数器	130
4.23 同步计数器的速度比较	132
4.24 任意模的同步计数器	133
4.25 同步可逆计数器	137
4.26 封锁	138
4.27 纹波计数器	138
4.28 集成电路计数器芯片	141
第 5 章 运算	144
5.1 符号数的表示法	144
5.2 符号数的二进制补码表示法	145
5.3 符号数的二进制反码表示法	147
5.4 两个二进制数的加法	149
5.5 串行加法器	150
5.6 并行加法	153
5.7 简单的加减法计算器	153
5.8 减法器	155

5.9	快速加法器	156
5.10	先行进位加法器	156
5.11	应用于组的先行进位	160
5.12	附加的先行进位单元的用途	161
5.13	算术逻辑运算部件	163
5.14	BCD加法	167
5.15	乘法和除法	169
第 6 章	存储器	170
6.1	随机存取存储器	170
6.2	半导体RAM的结构	171
6.3	并联的存储器芯片	172
6.4	一维和二维的存储器内部结构	175
6.5	只读存储器	175
6.6	ROM 编码器的结构	178
6.7	可编程序只读存储器和可擦可编程只读存储器	179
6.8	存储器的易失性	180
6.9	存储器的开关时间	180
6.10	可编程序逻辑阵列	184
6.11	动态 RAM	185
6.12	动态 RAM的数据读出	186
6.13	动态存储器的特点	188
6.14	串行存储器	189
6.15	电荷耦合器件, 串行存储器	192
6.16	存储栈	195
6.17	大容量存储器	196
6.18	符号含义	199
第 7 章	时序电路	202
7.1	状态	202
7.2	计数器作为时序系统的例子	203
7.3	模 4 可逆计数器	207
7.4	时序检测器	210
7.5	穆尔电路和米利电路	215
7.6	冗余状态的消除	219
7.7	用划分法消除冗余状态	221
7.8	实例	223
7.9	状态分配	227
7.10	两种可供选择的设计方法	227

7.11	基本型时序电路	229
7.12	分析实例	231
7.13	设计实例	232
7.14	竞态	234
7.15	临界状态的消除	235
7.16	实例	237
7.17	冗余状态的消除	238
7.18	消除冗余状态的另一个例子	240
7.19	冒险和异步电路	243
第 8 章	控制器	245
8.1	寄存器传送	245
8.2	其它操作	246
8.3	响应多个命令的寄存器	249
8.4	简单控制器	252
8.5	控制器的设计	255
8.6	移位寄存器式控制器	258
8.7	控制器的条件响应	260
8.8	减法的序列	264
8.9	简单计算机	265
8.10	计算机的运行	269
8.11	计算机控制器的设计图	271
8.12	中断	272
8.13	信号交换	275
第 9 章	计算机	279
9.1	一种改进结构	279
9.2	指令	283
9.3	指令小结	289
9.4	加法和减法	289
9.5	JMP和ISZ 指令的用法	292
9.6	乘法程序	293
9.7	说明子程序调用的程序	294
9.8	微程序设计	297
9.9	微程序转移	299
9.10	条件转移	300
9.11	流水传送	301
9.12	微程序控制的控制器	302
9.13	控制用 ROM 的内容	304

9.14 寻址方式	305
9.15 堆栈	306
第 10 章 微处理机	309
10.1 概述	309
10.2 程序设计员的微处理机结构	310
10.3 8080的单、双和三字节指令	313
10.4 数据传送指令	314
10.5 直接涉及累加器的指令	315
10.6 算术指令	317
10.7 几个实例	320
10.8 加1和减1指令	322
10.9 逻辑指令	322
10.10 比较指令	323
10.11 循环移位指令	324
10.12 取反和置“1”指令	325
10.13 实例	325
10.14 转移指令	326
10.15 调用和返回指令	327
10.16 推入和弹出指令	330
10.17 乘法实例	333
10.18 6800型微处理机	335
10.19 6800型微处理机的寻址方式	336
10.20 6800型微处理机的条件码寄存器	337
第 11 章 输入输出操作	339
11.1 8080的I/O控制信号的产生	339
11.2 隔开的及存储器映射的输入输出	342
11.3 IN和OUT指令的用法	345
11.4 未编码键盘	347
11.5 外围设备的控制	350
11.6 定时循环	350
11.7 中断	352
11.8 允许中断和禁止中断	354
11.9 轮询中断	355
11.10 其它的 I/O 通信	358
附 录	
附录A ASCII码(十六进制,最高有效二进制数位为零)	362

附录B	364
附录C 8080指令表.....	365
练习题.....	368

第 1 章 逻辑变量的代数

1.1 变量和函数

我们都熟悉变量和变量的函数的概念。变量的定义域，即变量 x 可以取值的范围，可以用无数种方法确定。例如， x 可以是从小负无穷大到正无穷大的所有实数；或者 x 可以被限制在 -17 到 -4 的区间内；或者 x 可以被限制在 1 到 10 范围内的正整数，等等。

函数是根据（自）变量 x 来确定另一个（因）变量 y 的规则， y 对于 x 的依赖关系可记为 $y=f(x)$ 。例如，假设我们想用 x 平方乘 5 再加 3 这个规则来确定 y ，那么，应当用式 $y=5x^2+3$ 表示 y 和 x 之间的函数关系。在这个简单的例子中，应用乘法和加法的算术运算确定 y 值。然而，当 x 的允许值的数目有限时，可以仅用一张表来表示函数，在表中对每一个 x 值有其对应的 y 值。当 x 的允许值的数目少时，使用这种表来表示函数是完全可行的，也是最方便的。例如，考虑上面的 $y=5x^2+3$ 这个例子，我们限定 x 为整数值 $x=0、1、2$ 和 3 。那么， y 和 x 之间的函数关系能够用表格形式来确定，如图1.1-1所示。

简单地引伸这些基本思想，显然，变量，不管是自变量还是因变量没有必要一定是数值。例如，假设自变量 x 的定义域为十字路口交通管理色灯的颜色，因变量 y 表示向十字路口行驶时要求驾车者的行为。那么， y 和 x 之间的函数关系如图1.1-2所示。 x 能够假设的值用陈述句“交通管理色灯为绿色”，或者“交通管理色灯为黄色”，或者“交通管理色灯为红色”表示。类似地， y 能够假设的值是“驾车者应当继续行驶汽车”，等等。

x	$y=f(x)$
0	3
1	8
2	23
3	48

图 1.1-1 数值函数

x	$y=f(x)$
绿色	继续
黄色	减慢
红色	停止

图 1.1-2 函数关系

1.2 逻辑变量

逻辑变量是具有三个特殊性质的变量：

1. 逻辑变量可以假设为仅有的两个可能值中的一个值或另一个值。
2. 这些值是用陈述句表示的，如同上述的交通管理色灯里的例子那样。
3. 用陈述句表示的这两个可能值必须是根据人的思维即逻辑而得到的，它们是不相

容的。

虽然,如所指出的那样,变量不需要有数值的含意,但也没有理由排除变量是数值的情况。因此,变量 x 可以有两个且仅有两个可能的、不相容的、用陈述句如“ x 值是7”和“ x 值是13”表示的值。逻辑变量的其它性质,将在下面的讨论中出现,在讨论中,仍用交通管理色灯这个例子。

假设,我们要求交通管理色灯只能是红色或者绿色,排除灯是黄色的可能性,也排除灯的颜色正在变化时,即既不显示绿色也不显示红色的时间间隔。那么,在这种情况下,图1.1-2中的 x 是逻辑变量。它的取值或者用陈述句“灯为绿色”表示,我们可以将它表示为 $x = \text{绿色}$;或者用陈述句“灯为红色”表示,我们可以将它表示为 $x = \text{红色}$ 。如果想要表示 $x = \text{红色}$,我们或者就用这种方法表示,或者用 $x = \text{非绿色}$ 来表示。在较简单的表示法中,“非”是用在那个值上面画一横杠的方法表示的。那么, $x = \text{非绿色}$ 就可写为 $x = \overline{\text{绿色}}$ 。最后,有 $x = \overline{\text{绿色}} = \text{红色}$ 。

1.3 逻辑变量的值

在任意一种形式的变量的一般情况中,比如说,采取数值形式的变量,这些变量可以表示任何物理量。因此, x 和 y 可以表示温度、压力、距离、速度或者时间,等等。从数学观点研究变量之间的函数关系时,我们对变量表示什么物理量并不感兴趣。这样,当 $x=1$ 时,从式 $y=5x^2+3$,我们得到 $y=8$,而与 x 和 y 可能代表什么物理量完全没有关系。这些变量能够取用的值,在两种情况中从这些值都是数这个意义上讲都是相同的。

以同样的方式,让我们给逻辑变量的两个可能值取两个名字,为的是我们能够考虑一个变量,而与这个变量可能表示什么无关。任何两个容易区别的名字将是合适的,但是使名字转达不相容性的意图也将是有用的。由于这个原因,这些名字假定为“热和冷”、“里和外”、“高和低”,等等;另一组可能的名字是使用值“真”和“假”,我们将在后面对这组名字作进一步的解释。因而,一个逻辑变量,比如说 A ,或者有值 $A = \text{真}$ (缩写为 $A = T$),或者有值 $A = \text{假}$ (缩写为 $A = F$)。如果事实确实是 $A = \text{真}$ ($A = T$),那么我们完全可以把 $A = \text{真}$ ($A = T$)写成 $A = \overline{\text{假}}$ ($A = \overline{F}$)。

现在,让我们回到交通管理色灯这个例子。(因为涉及逻辑变量,所以我们将按照更通常的习惯分别用 A 和 Z 而不用 x 和 y 表示自变量和因变量。)交通管理色灯的颜色和驾车者固有的响应之间的函数关系如图1.3-1(a)所示。假设对于变量 A 我们任意地把值 $A = T$ 指定给陈述句“灯为红色”。那么, $A = F$ 自然表示陈述句“灯为绿色”。类似地,让我们任意地把 $Z = T$ 和陈述句“驾车者应当继续行驶汽车”联系起来,那么图1.3-1(b)同样也表示灯的颜色和驾车者的行为之间的函数关系。如果使 A 和 Z 表示不同的颜色和驾车者的行为,那么图1.3-1(b)中T和F项的图表就会不同,但是,灯的颜色和驾车者的行为之间的函数关系当然是不会改变的。

像图1.3-1(b)那样,具有项值T和F的表

A	$Z = f(A)$	A	$Z = f(A)$
绿色	继续	F	T
红色	停止	T	F

图 1.3-1 (a) 中的函数关系变为
(b) 中的真值表

称为真值表。

1.4 单个逻辑变量的函数

单个逻辑变量的所有可能的函数 $Z=f(A)$ 如图1.4-1的四个真值表所示。为了保证不遗漏一种可能的函数,我们按照下面的方法着手进行,在 A 列仅列出它的两个可能值 F 和 T 。现在,在 Z 列中有两个必须要填入项值的位置,在每个位置内有两种可能的项值,因此,可能的不同的 Z 列数是 $2 \times 2 = 4$ 。给出了这四种函数后,我们可以确信没有更多的函数了。在图1.4-1(a)中,因为在每行里 Z 下面的项值与 A 下面的项值都是相同的,所以写为

A	$Z = f(A)$	A	Z	A	Z	A	Z
F	F	F	T	F	F	F	T
T	T	T	F	T	F	T	T
(a)		(b)		(c)		(d)	

图 1.4-1 单个变量的四种函数

$Z=A$ 。在图1.4-1(b)中, $Z=\bar{A}$ 。在图1.4-1(c)中, $Z=F$ 。在图1.4-1(d)中, $Z=T$ 。读者可以决定采取这种看法,即图1.4-1(c)和图1.4-1(d)实际上完全不表示函数,这是因为在一种情况里 Z 是假,在另一种情况里 Z 是真,完全与 A 的逻辑值无关的缘故。

1.5 两个逻辑变量的函数

现在,考虑两个逻辑变量 A 和 B 的函数 $Z=f(A, B)$ 。为了形成这些函数,我们可以从如图1.5-1那样的真值表着手。图中为 A 和 B 的逻辑值的每一种可能的组合提供了一行的位置。由于有两个变量,并且每个变量有两个值,所以四种组合是可能的。于是,为了导出函数,我们仅需在 Z 列中填入项值。因为要填入四项,而每项有两种可能的项值,所以在 Z 下可能的不同的总列数为 $2 \times 2 \times 2 \times 2 = 16$,相应地有两变量的16种可能的函数。正如我们将会看到的,如同单变量函数的情况,我们想采取这种看法,即这些“函数”中的某些函数实际上完全不是函数。我们将最后考虑所有可能的函数。目前,考虑几种特别有意义的函数。

A	B	$Z = f(A, B)$
F	F	
F	T	
T	F	
T	T	

图 1.5-1 未完成的两变量真值表

“与”函数

已经指出,可以用真值表定义逻辑函数。用图1.5-2的真值表定义的函数 $Z=f(A, B)$ 称为“与”函数, Z 对于 A 和 B 的这种依赖关系,写为

$$Z = A \text{ “与” } B \quad (1.5-1)$$

促成这个术语的因素在于,当且仅当 A 与 B 都是真(T)时, $Z = T$ (真),这个事实是能够由真值表证明的。“与”函数的另一种符号是

$$Z = A \cdot B \quad (1.5-2)$$

或者更简单地表示为

$$Z = AB \quad (1.5-3)$$

式(1.5-2)和式(1.5-3)使人联想到 Z 是 A 和 B 这两个因子“乘法”的结果。当然, A 和 B 不是数,而且并不意味着通常算术意义的乘法。然而,正如我们将会知道的,用该符号表示以转达乘法的联想是深思熟虑的,函数 A “与” B 常称为 A 和 B 的逻辑积。

“与”函数的第一个性质为它是可交换的,即如果改变 A 和 B 的次序(交换 A 和 B),那么函数 Z 不变,因此

$$Z = AB = BA \quad (1.5-4)$$

从图1.5-1的真值表可以直接看出这种情况。如果我们将该表的 B 列排在最左边,而不是 A 列排在最左边,那么 Z 列的各项值是不会变化的。

“与”函数的第二个性质为它是可结合的。假设有三个逻辑变量 A 、 B 和 C ,我们先形成逻辑积 AB 。因为这个积本身就是一个逻辑变量,所以我们可以将它与 C 形成逻辑积,得到 $(AB)C$ 。反之,假设我们先形成 BC ,然后形成 $A(BC)$ 。我们将要验证,无论怎样结合这些变量是不会产生任何差别的,因此

$$Z = (AB)C = A(BC) \quad (1.5-5)$$

“与”函数是可结合的这个事实几乎是十分明显的。然而,我们还要通过正式的证明,以便有机会指出用以证明有关逻辑变量的任何定理的基本方法。这个方法是由两个步骤组成的,先作真值表,然后仅仅辨认出是否有相同列,即在每行中都有相同的 T 和 F 项的列,如果有这些列,那么这些列所代表的所有函数都是相同的函数。

A	B	$Z = A \text{ “与” } B$
F	F	F
F	T	F
T	F	F
T	T	T

1	2	3	4	5	6	7
A	B	C	AB	$(AB)C$	BC	$A(BC)$
F	F	F	F	F	F	F
F	F	T	F	F	F	F
F	T	F	F	F	F	F
F	T	T	F	F	T	F
T	F	F	F	F	F	F
T	F	T	F	F	F	F
T	T	F	T	F	F	F
T	T	T	T	T	T	T

图 1.5-2 定义“与”函数的真值表

图 1.5-3 验证“与”函数的结合性

【例 1.5-1】 验证“与”函数是可结合的,即证明式(1.5-5)是正确的。

【解】 现在,我们涉及三个变量 A 、 B 和 C ,因为每个变量允许有两个值,所以三个变量有 $2 \times 2 \times 2 = 8$ 种可能的组合。因此,三个变量的真值表必须有8行,如图1.5-3所示。在这个表的 A 、 B 和 C 列中,列出了三个变量的所有可能的组合(稍后我们将指出,用什么方法可以使我们容易地作这样一个表,以保证既无重复而又无遗漏任一种组合)。

在第四列里,用图1.5-2的真值表给出的逻辑积的定义,一行一行地形成逻辑积 AB 。