九五 ★★★★★

普通高等教育“九五”部级重点教材

GZ 普通高等专科教育机电类规划教材

数字电子技术

哈尔滨理工大学工业技术学院 孙建三 主编
哈尔滨工业大学威海分校 张秀珍
淄博学院 于桂音 副主编

机械工业出版社
China Machine Press

数字电子技术 机械工业出版社

普通高等专科教育机电类规划教材

数 字 电 子 技 术

主　编　孙建三
副主编　张秀珍　于桂音
参　编　王　军　刘卫民　温琪莱
主　审　闫　军

机 械 工 业 出 版 社

本书是根据全国高等工程专科学校电气类专业教学指导委员会制定的“九五”教材规划而编写的，是机械工业部重点教材，与刘仁宇主编的《模拟电子技术》组成配套教材。

全书共分为九章，内容包括数字电路基础、逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲波形的产生与整形、数/模和模/数转换器、存储器及可编程器件、数字电子电路读图等。各章均附有小结、习题与思考题，书末还有部分习题答案。

本书可作为高等工程专科学校电气、电子类和相近专业数字电子技术课程的教材，也可作为职工大学、业余大学、电视大学同类专业的数字电子技术课程教材，还可作为从事电子技术的工程技术人员的参考用书。

图书在版编目（CIP）数据

数字电子技术/孙建三主编．—北京：机械工业出版社，2000.5 重印

普通高等专科教育机电类规划教材

ISBN 7-111-06856-4

Ⅰ．数… Ⅱ．孙… Ⅲ．数字电路-高等学校-教材 Ⅳ．TN711.5

中国版本图书馆 CIP 数据核字（1999）第 69234 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）

责任编辑：韩雪清　版式设计：霍永明　责任校对：张　媛

封面设计：姚　毅　责任印制：路　琳

北京机工印刷厂印刷·新华书店北京发行所发行

2001 年 2 月第 1 版第 3 次印刷

787mm×1092mm 1/16·13.75 印张·331 千字

8 001—12 000 册

定价：18.00 元

前　言

本书是根据全国高等工程专科学校电气类专业教学指导委员会制订的“九五”教材规划而编写的。是机械工业部重点教材，与刘仁宇主编的《模拟电子技术》组成配套教材。

本书的主要内容有：数字电路基础、逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲波形的产生与整形、数模与模数转换器、存储器及可编程器件、数字电子电路读图等。

本书的特点是：在编写过程中始终贯穿以培养能力为主、以应用为目的的原则。编写内容和深度符合国家教委组织制订的高等工程专科学校电子技术基础课程的教学基本要求。为了适应数字电子技术、特别是数字集成器件的飞速发展，在编写思路上，注意了知识的更新，以小规模集成电路引路，逐步向大规模集成电路深入，淡化集成电路内部结构和内部工作原理的讲述，注重器件的外部功能特性和应用，尽可能多地介绍常用器件和最新器件及应用实例，在选择内容上注意了与微型计算机的密切联系，适当增加了与计算机课程衔接的内容，为读者学习微型计算机开扩思路并打下基础。本书的逻辑图形符号及器件的型号均采用了最新国家标准，同时兼顾了国外集成器件逻辑符号的流行画法，并在附录中给出了常用逻辑图表符号对照表，以提高阅图的适应性。书中每章后有一定数量的习题与思考题，与正文内容密切配合，有利于组织教学和学生自学，书末附有部分习题答案。书中打 * 号的章节为选学内容，本教材的参考学时为 50～60 学时。

本书可作为高等工程专科学校电气、电子类和相近专业数字电子技术课程的教材，也可作为职工大学、业余大学、电视大学同类专业的数字电子技术课程教材，还可作为从事电子技术的工程技术人员的参考用书。

参加本书编写的有哈尔滨理工大学工业技术学院孙建三（第八、九章及附录）、王军（第三章）、哈尔滨工业大学威海分校张秀珍（第二、四章）、淄博学院于桂音（第六、七章）、昆明冶金高等专科学校刘卫民（第五章）、重庆钢铁高等专科学校温琪莱（第一章）等。孙建三任主编，负责全书的组织、修改和定稿工作；张秀珍、于桂音任副主编。

本书由长春大学闫军任主审。他提出了许多宝贵意见和建议，在此表示衷心的感谢。

由于编者水平有限，书中难免会有许多缺点、错误和不妥之处，诚恳希望使用本教材的师生和读者给予批评指正。

编　者

1998 年 5 月

目　录

第一章　数字电路基础

第一节　概　　述

一、数字信号与数字电路

电子电路中的信号可分为两类，一类在时间上和幅度上都是连续的，称为模拟信号，传送和处理模拟信号的电路，称为模拟电路；另一类在时间上和幅度上都是离散的，称为数字信号，例如，计时装置的时基信号、灯光闪烁等信号都属于数字信号，传送和处理数字信号的电路，称为数字电路。

二、数字电路举例

图 1-1 是一个电子秒表的框图，其基本功能是在手动控制下，记忆一段时间并显示出来。基本原理是在手动单脉冲作用下，控制逻辑电路根据需要产生清零、计数、停止三个控制信号，同时使信号发生器产生计时基准脉冲信号。开始计数之前，发清零信号，使计数器清零。然后发计数信号，计时脉冲进入计数器计数并经译码显示电路显示出来。最后发停止脉冲，停止计数，并保持从计时开始到停止这段时间的显示数字。

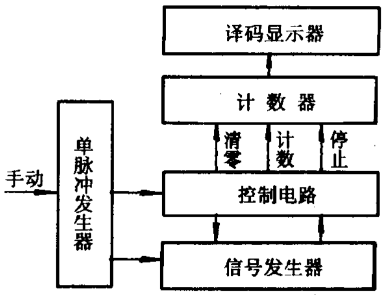

图 1-1　电子秒表原理框图

上述实例包括了脉冲整形、信号产生、逻辑控制、计数、译码显示电路，这些单元电路都是典型的数字电路。

三、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，即工作在截止区和饱和区。

(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。

第二节　数制与码制

一、数制

数制即计数体制。日常生活中最常用的是十进制数，而在数字系统和计算机中主要采用的是二进制数。

(一) 十进制数

十进制数采用 0，1，2，3，…，9 十个不同的数码来表示任何一位数，十进制数的基数是 10，进位规律是“逢十进一”，各数码处在不同数位时，所代表的数值是不同的。例如

$$192.85=1\times10^2+9\times10^1+2\times10^0+8\times10^{-1}+5\times10^{-2}$$

其中，10^2，10^1，10^0，10^{-1}，10^{-2}等分别称为十进制数各数位的权，都是 10 的幂。因此，对于任何一个十进制数，其数值都可表示为

$$
\begin{aligned}
[N]_{10} &= k_{n-1}\times 10^{n-1} + k_{n-2}\times 10^{n-2} + \cdots + k_0\times 10^0 + k_{-1}\times 10^{-1} \\
&\quad + k_{-2}\times 10^{-2} + \cdots + k_{-m}\times 10^{-m} \\
&= \sum_{i=-m}^{n-1} k_i \times 10^i \qquad (1\text{-}1)
\end{aligned}
$$

式中，k_i 为基数 10 的第 i 次幂的系数；m、n 为正整数；$k_i\times10^i$ 称为加权系数。

（二）二进制数

二进制数只有两个数码 0 和 1，基数是 2，进位规律是“逢二进一”，每个数位的权是 2 的幂，同样，二进制数也可以按权展开

$$
\begin{aligned}
[N]_2 &= k_{n-1}\times 2^{n-1} + k_{n-2}\times 2^{n-2} + \cdots + k_0\times 2^0 + k_{-1}\times 2^{-1} \\
&\quad + k_{-2}\times 2^{-2} + \cdots + k_{-m}\times 2^{-m} \\
&= \sum_{i=-m}^{n-1} k_i \times 2^i \qquad (1\text{-}2)
\end{aligned}
$$

式中，k_i 为基数 2 的第 i 次幂的系数；m，n 为正整数。

（三）八进制与十六进制

用二进制表示数时，数码串很长，书写和查错都很不方便，因此常用八进制和十六进制。

八进制数有 0，1，2，…，7 八个数码，基数是 8，进位规律是“逢八进一”，每个数位的权是 8 的幂。

十六进制数有 0，1，2，…，9，A，B，C，D，E，F 十六个数码，基数是 16，进位规律是“逢十六进一”，每个数位的权的是 16 的幂。

八进制数和十六进制数都可以按权展开。

二、二进制与其它进制的相互转换

（一）二进制数转换成十进制数

二进制数转换成十进制数的方法是按权展开，求加权系数之和。

例 1-1 将二进制数 $[1101010]_2$ 转换成十进制数。

解

$$
\begin{aligned}
[1101010]_2 &= 1\times2^6+1\times2^5+0\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0 \\
&= 2^6+2^5+2^3+2^1 \\
&= 64+32+8+2 \\
&= [106]_{10}
\end{aligned}
$$

（二）十进制数转换成二进制数

十进制数转换成任意进制数都可用基数乘除法。

十进制整数转换成二进制数可采用“除 2 取余，逆序排列”法，其步骤如下：

(1) 将给定的十进制数除以 2，余数便是二进制数的最低位。

(2) 将上一步的商再除以 2，余数便是二进制数的次低位。

(3) 重复步骤 (2)，直至商等于 0 为止。各次除得的余数，逆序排列，即可得到相应的二进制数。

例 1-2 将十进制数 $[53]_{10}$转换成二进制数。

解

$$\begin{array}{r|l l l}
2 & 53 & \cdots\cdots 1 & \uparrow\ \text{低位} \\
2 & 26 & \cdots\cdots 0 & \\
2 & 13 & \cdots\cdots 1 & \\
2 & 6 & \cdots\cdots 0 & \\
2 & 3 & \cdots\cdots 1 & \\
2 & 1 & \cdots\cdots 1 & \\
 & 0 & & \text{高位}
\end{array}$$

最后得 $(53)_{10}=(110101)_2$

十进制小数可以用基数乘法转换成二进制小数，即所谓“乘 2 取整，顺序排列法”。下面通过一个例子说明具体作法。

例 1-3 将 $(0.872)_{10}$转换成二进制数$\left(误差\ e<\frac{1}{2^4}\right)$。

解

$$\begin{array}{r}
0.872 \\
\times 2 \\ \hline
\boxed{1}.744 \\
\times 2 \\ \hline
\boxed{1}.488 \\
\times 2 \\ \hline
\boxed{0}.976 \\
\times 2 \\ \hline
\boxed{1}.952
\end{array}\ \downarrow$$

所以得 $(0.872)_{10}=(1101)_2$ 转换到第四位则误差小于$\frac{1}{2^4}$

（三）二进制数转换成八进制数和十六进制数

例 1-4 $[1100111]_2=[001,\ 100,\ 111]_2=[147]_8$

$[10100111]_2=[1010,\ 0111]_2=[A7]_{16}$

三、二进制码

在数字系统的人机对话时，需要把十进制数值、不同的文字、符号用二进制数码来表示。建立这种与十进制数值、文字、符号一一对应的代码称为编码，常用的编码包括二-十进制码、格雷码以及字符代码等。

（一）二-十进制码

用二进制代码来表示一个给定的十进制数，称为二-十进制编码，简称 BCD 码（Binary Coded Decimal）。0 和 1 组成的四位二进制数有 $2^4=16$ 种组合方式，可任选其中 10 种来表示十进制数的 0～9 这十个数码，因此，编码的方案很多。表 1-1 给出了几种常用的二-十进制编码。

因为 4 位二进制数代码共有 16 个不同的组合，用它对 0～9 十个十进制数编码，总有 6 个不用的状态，叫它无关状态，或叫伪码，例如 8421 码中的 1010～1111 为六个伪码。

1. 8421 码、2421 码、5421 码

这几种代码的共同特点是，每一组代码中的每一位的权是固定不变的，称为恒权代码。其加权系数之和，即是所对应的十进制数。例如

$$[1001]_{8421BCD}=[9]_{10},\ [1001]_{5421BCD}=[6]_{10}$$

表 1-1 常用的 BCD 码

十进制数	8421 码	2421 码	5421 码	余 3 码	余 3 循环码
0	0000	0000	0000	0011	0010
1	0001	0001	0001	0100	0110
2	0010	0010	0010	0101	0111
3	0011	0011	0011	0110	0101
4	0100	0100	0100	0111	0100
5	0101	1011	1000	1000	1100
6	0110	1100	1001	1001	1101
7	0111	1101	1010	1010	1111
8	1000	1110	1011	1011	1110
9	1001	1111	1100	1100	1010

2. 余 3 码

余 3 码所表示的十进制数比所对应的自然二进制码所代表的十进制数多“3”，余 3 码中的每一位没有固定的权，称为变权代码。余 3 码中，0 和 9 的代码，1 和 8 的代码，…，等都互为反码，是一种对 9 的自补代码。

3. 余 3 循环码

余 3 循环码也是一种变权代码，它从循环码（见表 1-2）的第四个状态开始取十个状态代表十进制数。

表 1-2 循环码编码表

十进制数	循环码				十进制数	循环码			
	G_3	G_2	G_1	G_0		G_3	G_2	G_1	G_0
0	0	0	0	0	8	1	1	0	0
1	0	0	0	1	9	1	1	0	1
2	0	0	1	1	10	1	1	1	1
3	0	0	1	0	11	1	1	1	0
4	0	1	1	0	12	1	0	1	0
5	0	1	1	1	13	1	0	1	1
6	0	1	0	1	14	1	0	0	1
7	0	1	0	0	15	1	0	0	0

（二）循环码

循环码也称格雷（Gray）码，其编码表如表 1-2 所示。循环码的特点是任意两个相邻数所对应的代码之间仅有一位不同。

*（三）字符编码

常用的字母和字符编码有 ASCII 码和 ISO 码。ASCII 码是美国标准信息交换码的简称，其编码表如表 1-3。这是一组八位二进制代码，用 $b_6 \sim b_0$ 七位表示 $2^7=128$ 个不同的信息，第八位 b_7 作为奇偶校验位。

表 1-3 ASCII 编码

字符 $b_6b_5b_4$ / $b_3b_2b_1b_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	\	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	—	=	M	]	m	}
1110	SO	RS	·	>	N	∧	n	~
1111	SI	US	/	?	O	—	o	DEL

第三节 逻辑代数

逻辑代数（布尔代数[1]）是一种描述事物逻辑关系的数学方法，是研究逻辑电路的数学工具。

一、逻辑变量

事物的发展变化都有一定的因果关系。例如图 1-2 所示的指示灯控制电路。我们用字母 Y 表示指示灯，用 A、B 表示两个开关。指示灯 Y 的亮、灭两种状态取决于开关 A、B 的通断两种状态。我们将 A、B 称为输入变量，将 Y 称为输出变量。

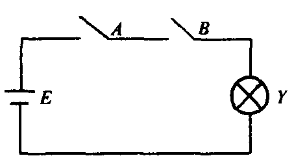

图 1-2 指示灯控制电路

可见，逻辑变量和普通代数中的变量一样，都是用字母表示的。但是，逻辑变量描述的是事物对立的逻辑状态（如上例中开关的通、

[1] 布尔代数是英国数学家 G. Boole 提出的。

断，灯的亮、灭)。采用的是仅有两个数值的变量。在逻辑代数中，我们通常用逻辑的 0 和 1 来表示事物的两种状态，所以逻辑变量与普通代数变量不同的是它的取值只有 0 和 1 两种可能，是一种二值量。

二、三种基本逻辑运算

逻辑代数中有与、或、非三种基本逻辑运算。下面结合具体实例分别进行讨论。

(一) 与运算

由图 1-2 给出的指示灯控制电路可知，如果有一个开关不接通或两个均不接通，指示灯不亮。只有当两个开关全部接通时，指示灯才会亮。指示灯的亮灭与开关的通断间存在一种“与”逻辑关系，即只有决定事物结果（灯亮）的几个条件（开关 A、B 接通）全部具备时，结果才会发生。表 1-4 是这种灯控电路的与逻辑关系表。

如果用二值量 0 和 1 来表示逻辑状态，设开关断开和灯不亮用 0 表示，而开关接通和灯亮用 1 表示，则可得到表 1-5。这种用逻辑变量的真正取值反映逻辑关系的表格称为逻辑真值表。

若用逻辑表达式来描述与逻辑，则可写为

$$Y=A\cdot B \tag{1-3}$$

式中小圆点“·”表示 A、B 的与逻辑关系，称作与运算、逻辑乘。小圆点可省略。曾有用符号 ∧、∩、& 表示与运算的。$A\cdot B$ 读作 A 与 B。

与逻辑关系也可用逻辑符号表示。图 1-3 为与逻辑符号。

表 1-4　与逻辑关系表

开关 A	开关 B	灯 Y
断	断	灭
断	通	灭
通	断	灭
通	通	亮

表 1-5　与逻辑真值表

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

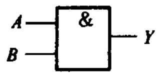

图 1-3　与逻辑符号

(二) 或运算

图 1-4a 表示一简单或逻辑电路。只要开关 A、B 其中一个接通或两个都接通，则灯亮，而当开关 A、B 均不通时，则灯不亮。其或逻辑关系表和真值表如表 1-6，表 1-7。

表 1-6　或逻辑关系表

开关 A	开关 B	灯 Y
断	断	灭
断	通	亮
通	断	亮
通	通	亮

表 1-7　或逻辑真值表

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

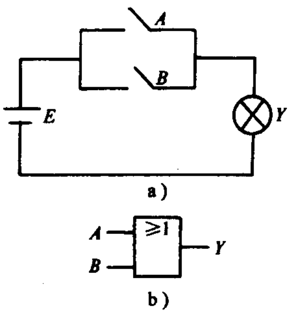

图 1-4　或逻辑关系和符号

a) 或逻辑关系举例

b) 或逻辑符号

由此，可总结出或逻辑关系：在决定事物结果的几个条件中，只要一个或一个以上条件得到满足时，结果就会发生；否则，结果不会发生。

逻辑变量间的逻辑或关系，也称或运算、逻辑加运算，并用符号“＋”表示。曾有用符号∨、∪表示或运算的。A、B 和 Y 的或逻辑关系表达式为

$$Y=A+B \tag{1-4}$$

$A+B$ 读作 A 或 B。图 1-4b 是或逻辑符号。

（三）非运算

由图 1-5 所示电路可知，当开关 A 接通时，指示灯 Y 不亮；而当开关 A 关断时，指示灯亮。其逻辑关系表和真值表如表 1-8、表 1-9 所示。由此可总结出第三种逻辑关系：非逻辑关系，即当条件满足时，结果不会发生；而条件不满足时，结果才会发生。

在逻辑代数中，逻辑非称为非运算，也称作求反运算，通常在变量上方加一短横线表示非运算。所以逻辑表达式为

$$Y=\overline{A} \tag{1-5}$$

$\overline{A}$ 读作 A 非。逻辑非的符号如图 1-5b 所示。图中小圆圈表示非运算。

表 1-8　非逻辑关系表

开关 A	灯 Y
断	亮
通	灭

表 1-9　非逻辑真值表

A	Y
0	1
1	0

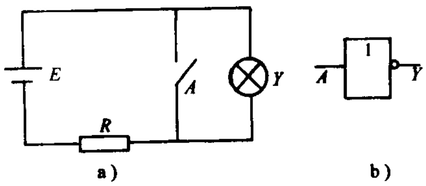

图 1-5　非逻辑关系与符号

a）非逻辑关系　b）非逻辑符号

三、逻辑函数、逻辑函数的表示方法及相互转换

从上面所讨论的几种逻辑关系中可以看到，当输入变量的取值确定之后，输出变量的取值也就被确定了，因而输入与输出之间是一种函数关系，我们将这种函数关系称之为逻辑函数。实际的逻辑问题，通常都是由三种基本逻辑运算组合而成的。逻辑函数的一般表达式可以写作

$$Y=F\ (A,\ B,\ C,\ D,\ \cdots) \tag{1-6}$$

任何一个具体事物的因果关系都可以用一个逻辑函数来描述。例如这样一个实际逻辑问题：有 A、B 两台电话机公用一条电话线，因此对外通话时两台话机不能同时使用，设电话线用逻辑变量 Y 表示，则话机是否使用与电话线是否通话就存在一种逻辑关系。

（二）逻辑函数的表示方法及相互转换

表示逻辑函数的方法有逻辑真值表（简称真值表）、逻辑函数表达式（也称表达式）、逻辑图、工作波形图和卡诺图。

这里结合上一个具体实例，介绍前四种表示方法以及它们之间的相互转换方法。用卡诺图表示逻辑函数的方法将在后面专门介绍。

1. 真值表

在话机通话的实例中，我们设话机使用、电话线通话为逻辑 1，话机不使用、电话线不通话为逻辑 0，很容易列出输入变量 A、B 的不同取值与函数 Y 的值的对应关系，即逻辑真值表

（表 1-10）。可见，逻辑真值表是将输入变量的各种可能取值和相应的函数值排列在一起组成的表格。一个确定的逻辑函数只有一个逻辑真值表。

逻辑真值表能够直观、明了地反映变量取值和函数值的对应关系，但变量多时列写比较繁琐。

在列写真值表时，输入变量的取值组合按照二进制递增（或递减）的顺序排列较好，因为这样做既不易遗漏，也不会重复。

表 1-10 通话实例真值表

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

2. 逻辑函数表达式

由真值表 1-10 可知，在 A、B 状态的四种组合中只有第二种（$A=0$，$B=1$）和第三种（$A=1$，$B=0$）两种状态组合才能使函数值 Y 为 1。A、B 之间相与，而这种状态组合之间相或，不论变量 A、B 或函数 Y，凡取 1 值的用原变量表示，取 0 值的用反变量表示，则可写出 Y 的函数表达式

$$Y=\overline{A}B+A\overline{B} \tag{1-7}$$

由此可见，逻辑函数式是一种用与、或、非等逻辑运算组合起来的表达式。用它表示逻辑函数、形式简洁，书写方便，便于推演、变换，但是它不能直接表示出变量取值与函数间的对应关系，而且同一逻辑函数可以有不同的逻辑函数表达式。

3. 逻辑图

将逻辑函数表达式中各变量间的与、或、非等运算关系用相应的逻辑符号表示出来，就是函数的逻辑图。

式（1-7）的逻辑关系，可用图 1-6 的逻辑图来表示。

逻辑图与数字电路的器件有明显的对应关系，便于制作实际电路。但它不能直接进行逻辑推演和变换。

4. 波形图

反映输入和输出波形变化规律的图形，称为波形图，也称时序图。

图 1-7 是给定 A、B 波形后所画出的上述函数 Y 的波形图。

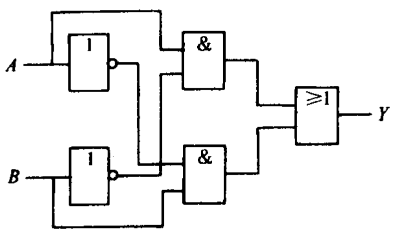

图 1-6 式（1-7）的逻辑图

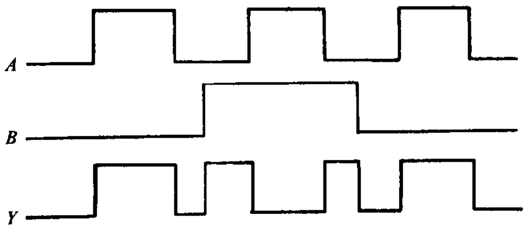

图 1-7 式（1-7）的波形图

波形图能直接反映变量与时间的关系和函数值变化的规律。与实际电路中的电压波形相对应。

5. 各种表示方法间的相互转换

同一逻辑函数可以用几种不同的方式来表示，这几种表示方法之间必然可以互相转换。下面举例说明转换方法。

例 1-4 某逻辑函数的真值表如表 1-11 所示，试将它转换成逻辑表达式，并画出逻辑图。

解 （1）由真值表转换成逻辑表达式。将真值表中使 $Y=1$ 的各乘积项进行逻辑加，可得

$$Y=\overline{A}\,\overline{B}\,\overline{C}+A\,B\,C$$

（2）根据函数表达式，可画出图 1-8 所示的逻辑图。

表 1-11　某逻辑函数真值表

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

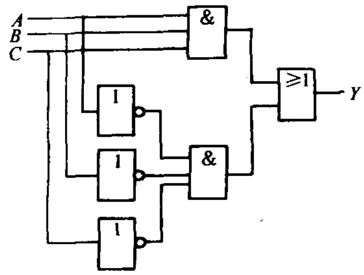

图 1-8　例 1-4 的逻辑图

例 1-5　已知函数的逻辑表达式为 $Y=A+\overline{B}C$，求它对应的真值表，已知输入波形，画出输出波形图。

解　将输入变量 A、B、C 的各种取值逐一代入表中计算，即可得到函数 Y 的真值表，如表 1-12 所示。

表 1-12　例 1-5 的真值表

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

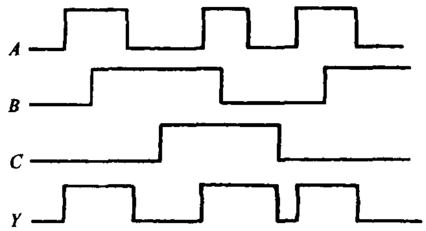

图 1-9　例 1-5 的波形图

根据输入 A、B、C 的波形画出输出 Y 的波形如图 1-9 所示。

例 1-6　已知函数 Y 的逻辑图如图 1-10，写出函数 Y 的逻辑表达式。

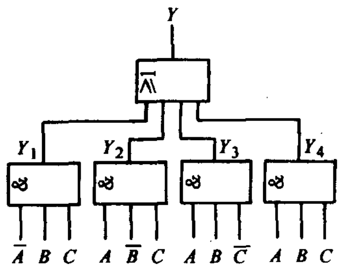

图 1-10　例 1-6 的逻辑图

解　逐级写出输出端函数表达式如下：

$Y_1=\overline{A}BC$

$Y_2=A\overline{B}C$

$Y_3=AB\overline{C}$

$Y_4=ABC$

最后得到函数 Y 的表达式

$Y=Y_1+Y_2+Y_3+Y_4=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC$

第四节　逻辑代数的基本定律和规则

一、基本定律

逻辑代数中有 10 个基本定律，反映了逻辑运算的基本规律。它是化简逻辑函数、分析和

设计逻辑电路的基础，必须熟悉和掌握。

规律 1　0-1 律：$A\cdot 1=A$，$A+0=A$

$A\cdot 0=0$，$A\cdot 1=A$

规律 2　交换律：$AB=BA, A+B=B+A$

规律 3　结合律：$ABC=A(BC)=(AB)C$

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

规律 4　分配律：$A(B+C)=AB+AC$

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

规律 5　重叠律：$AA=A, A+A=A$

规律 6　互补律：$A\overline{A}=0, A+\overline{A}=1$

规律 7　吸收律：$A+AB=A, A(A+B)=A$

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

规律 8　还原律：$\overline{\overline{A}}=A$

规律 9　反演律(摩根律)：$\overline{AB}=\overline{A}+\overline{B}, \overline{A+B}=\overline{A}\ \overline{B}$

规律 10　隐含律：$AB+\overline{A}C+BC=AB+\overline{A}C$

以上定律都可用真值表证明。下面以反演律为例进行证明。

对应 AB 的各种不同组合，如下表求出 $\overline{AB}$ 和 $\overline{A}+\overline{B}$ 的函数值，如果两者真值表相同，则两个函数相等。

AB	$\overline{AB}$	$\overline{A}+\overline{B}$
00	1	1
01	1	1
10	1	1
11	0	0

由真值表可知，有 $\overline{AB}=\overline{A}+\overline{B}$

下面再用公式法证明隐含律。

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

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

$=AB(1+C)+\overline{A}C(1+B)$

$=AB+\overline{A}C$

二、逻辑代数运算的基本规则

(一) 代入规则

在任何一个逻辑等式中，若将等式两边所出现的同一变量以另一函数式代替，等式仍然成立，这一规则称为代入规则。因为任何逻辑函数式和被替代的变量一样，只有 0 和 1 两种取值，所以代入规则是正确的。利用代入规则可以扩展公式和证明恒等式。

例 1-7　证明公式 $\overline{ACD}=\overline{A}+\overline{C}+\overline{D}$。

证　利用摩根定律 $\overline{AB}=\overline{A}+\overline{B}$，用函数式 $Y=CD$ 代替等式两边的变量 B 得

左边　$\overline{AB}=\overline{AY}=\overline{ACD}$

右边　$\overline{A}+\overline{B}=\overline{A}+\overline{CD}=\overline{A}+\overline{C}+\overline{D}$

所以　$\overline{ACD}=\overline{A}+\overline{C}+\overline{D}$

(二) 反演规则

对于任意一个逻辑函数式 Y，若将式中所有的“·”换成“+”，“+”换成“·”；0 换成 1，1 换成 0；原变量换成反变量，反变量换成原变量，那么所得到的新的逻辑函数表达式就是原逻辑函数 Y 的反函数 $\overline{Y}$。这个规则称为反演规则。

利用反演规则，可以很方便地求得一个逻辑函数的反函数。

使用反演规则时，应注意运算符号的先后顺序：先括号，后与，最后或。另外还要注意，不是一个变量上的公共非号要保持不变。

例 1-8 已知 $Y=A\ B+\overline{C}\ D$,求 $\overline{Y}$。

解 利用反演规则可得 $\overline{Y}=(\overline{A}+\overline{B})\cdot(C+D)$

例 1-9 求函数 $Y=(AB+C)\cdot\overline{CD}$的反函数 $\overline{Y}$。

解 $Y=(\overline{A}+\overline{B})\overline{C}+\overline{\overline{C}+\overline{D}}$

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

上式不能写成 $\overline{Y}=\overline{A}+\overline{B}\ \overline{C}+\overline{C}+\overline{D}$

（三）对偶规则

对于任何一个逻辑函数 Y,若把式中的“·”换成“+”,“+”换成“·”;0 换成 1,1 换成 0;所得到的新的逻辑表达式就是原函数 Y 的对偶式 Y'。

如果两个逻辑函数式相等,则它们的对偶式也相等。这就是对偶规则。

前面介绍的基本定律中,规律 1 中的前两式,规律 2～7、9 中的每对式子都互为对偶式。利用对偶规则可证明恒等式。

例 1-10 证明等式 $A\overline{B}+\overline{A}B=(\overline{A}+\overline{B})(A+B)$。

解 首先求左边的对偶式 $Y'=(A+\overline{B})\cdot(\overline{A}+B)=AB+\overline{A}\ \overline{B}$

而右边的对偶式 $Y'=\overline{A}\ \overline{B}+AB$

可见,等式两边的对偶式相等,故原式相等。

同样,在运用对偶规则时,也要注意运算的先后顺序。

第五节　逻辑函数的代数化简法

在逻辑电路设计中，对逻辑函数化简具有十分重要的意义，最简的与或表达式应遵循乘积项最少，每个乘积项的变量数最少的化简原则。

一个逻辑函数可以有多种不同的表达式，例如：

$Y=AC+\overline{C}D$	与—或表达式
$=\overline{\overline{AC}\cdot\overline{\overline{C}D}}$	与非—与非表达式
$=\overline{\overline{A}C+\overline{C}\ \overline{D}}$	与—或—非表达式
$=(A+\overline{C})\cdot(C+D)$	或—与表达式
$=\overline{\overline{A+\overline{C}}+\overline{C+D}}$	或非—或非表达式

因为从逻辑函数的真值表直接得到的是一个与或表达式，同时与或表达式也比较容易地转换成其它形式的表达式，因此，我们主要讨论与或式的化简方法。

代数化简法就是用逻辑代数的基本定律和公式对逻辑函数进行化简,常用以下几种方法。

一、并项法

利用公式 $AB+A\overline{B}=A$，将两项合并成一项，并消去一个变量。如

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

二、吸收法

利用公式 $A+AB=A$ 消去多余的项。如