第3章 数制、编码与逻辑代数

本章知识结构

3.1 数制
- 3.1.1 十进制数
- 3.1.2 二进制数
- 3.1.3 十六进制数
- 3.1.4 数制转换

3.2 编码
- 3.2.1 8421BCD 码、2421BCD 码和 5421BCD 码
- 3.2.2 余 3 码
- 3.2.3 格雷码
- 3.2.4 奇偶校验码

3.3 逻辑代数
- 3.3.1 逻辑代数的常量和变量
- 3.3.2 逻辑代数的基本运算规律
- 3.3.3 逻辑表达式的化简
- 3.3.4 逻辑表达式、逻辑电路和真值表相互转换
- 3.3.5 逻辑代数在逻辑电路中的应用

数制就是数的进位制，十进制是平常使用最多的数制，而数字电路系统中常使用二进制。编码是指用二进制数表示各种数字或符号的过程。逻辑代数是分析数字电路的数学工具。在分析和设计数字电路时需要应用逻辑代数。

3.1 数 制

数制就是数的进位制。在日常生活中，经常会接触到 0、7、8、9、168、295 等这样的数字，这些数字就是一种数制——十进制数。另外，数制还有二进制数和十六进制数等。

3.1.1 十进制数

十进制数有以下两个特点。

① 有 10 个不同的数码：0、1、2、3、4、5、6、7、8、9。任意一个十进制数均可以由这 10 个数码组成。

② 遵循“逢十进一”的计数原则。对于任意一个十进制数 N，它都可以表示成

$$N=a_{n-1}\times 10^{n-1}+a_{n-2}\times 10^{n-2}+\cdots+a_1\times 10^1+a_0\times 10^0+a_{-1}\times 10^{-1}+\cdots+a_{-m}\times 10^{-m}$$

其中 m 和 n 为正整数。

这里的 a_{n-1}、a_{n-2}……a_{-m} 称为数码，10 称为基数，10^{n-1}、10^{n-2}……10^{-m} 是各位数码的位权。

例如，根据上面的方法可以将十进制数 3 259.46 表示成 $3\,259.46=3\times10^3+2\times10^2+5\times10^1+9\times10^0+4\times10^{-1}+6\times10^{-2}$。

请写出 8 436.051 的展开式：

8 436.051=____。

3.1.2 二进制数

十进制数是最常见的数制，除此以外，还有二进制数、八进制数、十六进制数等。在数字电路中，二进制数用得最多。

1. 二进制数的特点

二进制数有以下两个特点。

① 有两个数码：0 和 1。任何一个二进制数都可以由这两个数码组成。

② **遵循"逢二进一"的计数原则。**对于任意一个二进制数 N，它都可表示成

$$N=a_{n-1}\times 2^{n-1}+a_{n-2}\times 2^{n-2}+\cdots+a_0\times 2^0+a_{-1}\times 2^{-1}+\cdots+a_{-m}\times 2^{-m}$$

其中 m 和 n 为正整数。

这里的 a_{n-1}、$a_{n-2}\cdots\cdots a_{-m}$ 称为数码，2 称为基数，2^{n-1}、$2^{n-2}\cdots\cdots 2^{-m}$ 是各位数码的位权。

例如，二进制数 11011.01 可表示为 $(11011.01)_2=1\times2^4+1\times2^3+0\times2^2+1\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}$。

请写出 $(1011.101)_2$ 的展开式：

$(1011.101)_2=$______________________________。

2. 二进制的四则运算

（1）加法运算

加法运算法则是"逢二进一"。运算规律如下

0+0=0　0+1=1　1+0=1　1+1=10

当遇到"1+1"时就向相邻高位进 1。

例如，求 $(1011)_2+(1011)_2$，可以用与十进制数相同的竖式计算

$$\begin{array}{r}1011\\ +\ 1011\\ \hline 10110\end{array}$$

即 $(1011)_2+(1011)_2=(10110)_2$。

（2）减法运算

减法运算法则是"借一当二"。运算规律如下

0−0=0　1−0=1　1−1=0　10−1=1

当遇到"0−1"时，需向高位借 1 当"2"用。

例如，求 $(1100)_2-(111)_2$

$$\begin{array}{r}1100\\ -\ \ 111\\ \hline 101\end{array}$$

即 $(1100)_2-(111)_2=(101)_2$。

（3）乘法运算

乘法运算法则是"各数相乘，再作加法运算"。运算规律如下

0×0=0　1×0=0　0×1=0　1×1=1

例如，求 $(1101)_2\times(101)_2$

$$\begin{array}{r} 1101 \\ \times\quad 101 \\ \hline 1101 \\ 1101 \\ \hline 1000001 \end{array}$$

即 $(1101)_2 \times (101)_2=(1000001)_2$。

（4）除法运算

除法运算法则是“各数相除，再作减法运算”。运算规律如下

$$\mathbf{0 \div 1=0 \quad 1 \div 1=1}$$

例如，求 $(1111)_2 \div (101)_2$

$$\begin{array}{r} 11 \\ 101\overline{)1111} \\ \underline{101} \\ 101 \\ \underline{101} \\ 0 \end{array}$$

即 $(1111)_2 \div (101)_2=(11)_2$。

3.1.3 十六进制数

十六进制数有以下两个特点。

① 有 16 个数码：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F，这里的 A、B、C、D、E、F 分别代表 10、11、12、13、14、15。

② 遵循“逢十六进一”的计数原则。对于任意一个十六进制数 N，它都可表示成

$$N=a_{n-1} \times 16^{n-1}+a_{n-2} \times 16^{n-2}+\cdots+a_0 \times 16^0+a_{-1} \times 16^{-1}+\cdots+a_{-m} \times 16^{-m}$$

其中 m 和 n 为正整数。

这里的 a_{n-1}、a_{n-2}……a_{-m} 称为数码，16 称为基数，16^{n-1}、16^{n-2}……16^{-m} 是各位数码的位权。

例如，十六进制数可表示为 $(3A6.D)_{16}=3 \times 16^2+10 \times 16^1+6 \times 16^0+13 \times 16^{-1}$。

十六进制常用字母 H 表示，故 $(3A6.D)_{16}$ 也可表示成 3A6.DH。

请写出 $(B65F.6)_{16}$ 的展开式：

$(B65F.6)_{16}$=____。

3.1.4 数制转换

不同数制之间可以相互转换，下面介绍几种数制之间的转换方法。

1. 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：将二进制数各位数码与位权相乘后求和，就能得到十进制数。

例如，$(101.1)_2=1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}=4+0+1+0.5=(5.5)_{10}$

2. 十进制数转换成二进制数

十进制数转换成二进制数的方法是：采用除 2 取余法，即将十进制数依次除 2，并依次记下余数，一直除到商数为 0，最后把全部余数按相反次序排列，就能得到二进制数。

例如，将十进制数 $(29)_{10}$ 转换成二进制数，方法为

2	29	余 1	a_0 ↑	低位
2	14	余 0	a_1	
2	7	余 1	a_2	
2	3	余 1	a_3	
2	1	余 1	a_4	高位
	0			

即 $(29)_{10}=(11101)_2$。

3. 二进制与十六进制的相互转换

（1）二进制数转换成十六进制数

二进制数转换成十六进制数的方法是：从小数点起向左、右按 4 位分组，不足 4 位的，整数部分可在最高位的左边加“0”补齐，小数点部分不足 4 位的，可在最低位右边加“0”补齐，每组以其对应的十六进制数代替，将各个十六进制数依次写出即可。

例如，将二进制数 $(1011000110.111101)_2$ 转换为十六进制数，转换如下

$$
\begin{aligned}
&(1011000110.111101)_2\\
&=(\underline{0010}\ \underline{1100}\ \underline{0110}\ \underline{\ .\ }\ \underline{1111}\ \underline{0100})_2\\
&\quad\ \ \downarrow\qquad\ \downarrow\qquad\ \downarrow\quad\ \downarrow\quad\ \downarrow\qquad\ \downarrow\\
&=(\ 2\qquad C\qquad 6\quad\ .\quad\ F\qquad 4\ )_{16}\\
&=(2C6\,.\,F4)_{16}
\end{aligned}
$$

注：十六进制的 16 位数码为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F，它们分别与二进制数 0000、0001、0010、0011、0100、0101、0110、0111、1000、1001、1010、1011、1100、1101、1110、1111 相对应。

（2）十六进制数转换成二进制数

十六进制数转换成二进制数的方法是：从左到右将待转换的十六进制数中的每个数依次用 4 位二进制数表示。

例如，将十六进制数（13AB.6D）$_{16}$转换成二进制数

$$
\begin{array}{ccccccc}
(1 & 3 & A & B & . & 6 & D)_{16} \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
=(\underline{0001} & \underline{0011} & \underline{1010} & \underline{1011} & \underline{.} & \underline{0110} & \underline{1101})_2
\end{array}
$$

$$=(0001001110101011.01101101)_2$$

3.2 编　码

数字电路只能处理二进制形式的信息，而实际上经常会遇到其他形式的信息，如十进制数字、字母和文字等，这些信息数字电路是无法直接处理的，必须要将其先处理成二进制数。**用二进制数表示各种数字或符号的过程称为编码。编码是由编码电路来完成的。**

编码电路的种类很多，在本节主要介绍二-十进制编码。**利用 4 位二进制数组合表示十进制 10 个数的编码，称为二-十进制编码，简称 BCD 码。**根据编码方式不同，可分为 8421BCD 码、2421BCD 码、5421BCD 码、余 3 码、格雷码和奇偶校验码。

3.2.1　8421BCD 码、2421BCD 码和 5421BCD 码

1. 8421BCD 码

8421BCD 码是一种有权码，它的 4 位二进制从高到低的位权依次为 $2^3=8$、$2^2=4$、$2^1=2$、$2^0=1$。

8421BCD 码转换成十进制数举例说明如下

$$(0110)_{8421BCD}=0\times2^3+1\times2^2+1\times2^1+0\times2^0=0\times8+1\times4+1\times2+0\times1=(6)_{10}$$

$$(011001011000.01000010)_{8421BCD}=(0110\ 0101\ 1000\ .0100\ 0010)_{8421BCD}=(6\ 5\ 8\ .4\ 2)_{10}=(658.42)_{10}$$

十进制数转换成 8421BCD 码举例说明如下

$$(7)_{10}=(0111)_{8421BCD}$$

$$(901.73)_{10}=(1001\ 0000\ 0001.0111\ 0011)_{8421BCD}=(100100000001.01110011)_{8421BCD}$$

2. 2421BCD 码和 5421BCD 码

2421BCD 码、5421BCD 码和 8421BCD 码相似，它们都是有权码。**2421BCD 码的 4 位二进制从高到低的位权依次为 2、4、2、1。5421BCD 码的 4 位二进制从高到低的位权依次为 5、4、2、1。**它们与十进制数的相互转换与 8421BCD 码相同。

8421BCD 码、2421BCD 码、5421BCD 码、余 3 码与十进制数的对应关系见表 3-1。

表 3-1 常见 BCD 码与十进制数对照表

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

2421BCD 码与十进制数的相互转换举例说明如下

$$(1010)_{2421BCD}=1\times2+0\times4+1\times2+0\times1=2+0+2+0=(4)_{10}$$

$$(702.54)_{10}=(1101\ 0000\ 0010.1011\ 0100)_{2421BCD}$$

5421BCD 码与十进制数的相互转换举例说明如下

$$(1010)_{5421BCD}=1\times5+0\times4+1\times2+0\times1=5+0+2+0=(7)_{10}$$

$$(702.54)_{10}=(1010\ 0000\ 0010.1000\ 0100)_{5421BCD}$$

3.2.2 余 3 码

余 3 码是由 8421BCD 码加上 3（0011）得来的，它是一种无权码。余 3 码与十进制数的相互转换举例说明如下

$$(0111)_{余3码}=(0111-0011)_{8421BCD}=(0100)_{8421BCD}=(4)_{10}$$

$$(6)_{10}=(0110)_{8421BCD}=(0110+0011)_{余3码}=(1001)_{余3码}$$

$$(7.5)_{10}=(0111.0101)_{8421BCD}=(1010.1000)_{余3码}$$

3.2.3 格雷码

两个相邻代码之间仅有 1 位数码不同的无权码称为格雷码。十进制数与格雷码的对应

关系见表 3-2。

表 3-2　十进制数与格雷码对照表

十进制数	格雷码	十进制数	格雷码
0	0000	9	1101
1	0001	10	1111
2	0011	11	1110
3	0010	12	1010
4	0110	13	1011
5	0111	14	1001
6	0101	15	1000
7	0100	权	无权
8	1100		

从表 3-2 中可以看出，相邻的两个格雷码之间仅有 1 位数码不同，如 5 的格雷码是 0111，它与 4 的格雷码 0110 仅最后 1 位不同，与 6 的格雷码 0101 仅倒数第 2 位不同。其他的编码方法表示的数码在递增或递减时，往往多位发生变化，3 的 8421BCD 码 0011 与 4 的 8421BCD 码 0100 同时有 3 位发生变化，这样在数字电路处理中很容易出错，而格雷码在递增或递减时，仅有 1 位发生变化，这样不容易出错，所以格雷码常用于高分辨率的设备中。

3.2.4　奇偶校验码

二进制数据在传递、存储过程中，可能会发生错误，即有时“1”变成“0”或“0”变成“1”。为了检查二进制数有无错误，可以采用奇偶校验码。

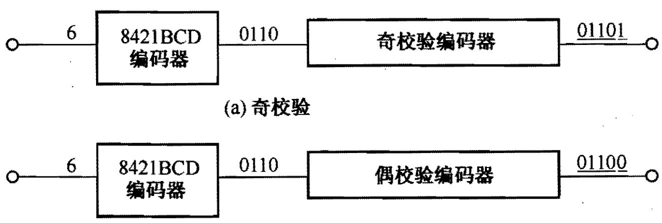
奇偶校验码由信息位和校验位组成。信息位就是数据本身，可以是位数不受限的任意二进制数；校验位是根据信息位中的“1”或“0”的个数加在信息位后面的 1 位二进制数。

奇偶校验码可分为奇校验码和偶校验码两种。校验位产生的规则是：对于奇校验，若信息位中有奇数个“1”，则校验位为“0”，若信息位中有偶数个“1”，则校验位为“1”；对于偶校验，若信息位中有偶数个“1”，则校验位为“0”，若信息位中有奇数个“1”，则校验位为“1”。

下面以图 3-1 来说明奇偶校验码的形成过程。

图 3-1（a）所示为奇校验编码，十进制数 6 先经 8421BCD 编码器转换成 0110，再送

到奇校验编码器，因为 0110 中 1 的个数是偶数，为保证整个奇偶校验码“1”的个数为奇数，校验位应为“1”，编码输出的数据为 01101。



(a) 奇校验

(b) 偶校验

图 3-1　奇偶校验码

图 3-1（b）所示为偶校验编码，十进制数 6 先经 8421BCD 编码器转换成 0110，再送到偶校验编码器，因为 0110 中 1 的个数是偶数，所以校验位为“0”，编码输出的数据为 01100。

在传递奇偶校验码数据时，如果数据中的某位发生了错误，如奇校验码 01101 在传递时变成了 01001，这样信息位“1”的个数为奇数，按奇校验规则校验位应为“0”，但校验位为“1”，这样信息位与校验位不相符，说明该数据出错。

奇偶校验编码只能发现 1 位数出错，不能发现 2 位以上（偶数位）数字出错，不过 2 位数字同时出错的可能性很小。另外，奇偶校验编码不能发现是数据中的哪 1 位出错。目前有一种汉明校验码，它既能发现错误又能查出错误数的位置，这种编码是在奇偶校验码的基础上改进的，如果有兴趣，读者可以查阅有关资料。

奇偶校验码虽然有一些缺陷，但它编码简单、实现容易，在要求不是很高的数字电路系统中仍被广泛采用。

3.3　逻辑代数

逻辑代数又称开关代数，是 19 世纪一位英国数学家布尔创立的，因而又称布尔代数。**逻辑代数是按一定逻辑规律进行运算的代数，它是研究数字电路的数学工具，为分析和设计数字电路提供了理论基础。**

3.3.1　逻辑代数的常量和变量

常量是指不变化的量，如 2、15 等都是常量；**变量是指会发生变化的量**，如 A 既可以代表 8，也可以代表 17，这里的 A 就是变量，它可以根据需要取不同的值，变量常用字母

表示。

逻辑代数有以下两个特点。

① 逻辑代数的常量有两个："1"和"0"；而变量只能有两个值："1"和"0"。

② 逻辑代数中的"1"和"0"不是表示数量大小，而是表示两种对立的逻辑状态（如真或假，高或低，开或关等）。

3.3.2 逻辑代数的基本运算规律

普通的代数在运算时有一定的规律，逻辑代数在运算是也有一定的规律，主要有基本运算定律和常用的恒等式。

1. 逻辑代数的基本运算定律

逻辑代数的基本运算定律见表 3-3。

表 3–3 逻辑代数的基本运算定律

自等律	$A+0=A \qquad A\cdot 1=A$
0-1 律	$A+1=1 \qquad A\cdot 0=0$
重叠律	$A+A=A \qquad A\cdot A=A$
互补律	$A+\overline{A}=A \qquad A\cdot \overline{A}=0$
吸收律	$A+AB=A \qquad A(A+B)=A$
非非律	$\overline{\overline{A}}=A$
交换律	$A+B=B+A \qquad AB=BA$
结合律	$(A+B)+C=A+(B+C) \qquad (AB)C=A(BC)$
分配律	$A(B+C)=AB+AC \qquad A+BC=(A+B)(A+C)$
反演律（摩根定理）	$\overline{AB}=\overline{A}+\overline{B} \qquad \overline{A+B}=\overline{A}\cdot\overline{B}$

若要证明以上各个定律是否正确，可将各变量的取值代入相应的式子中，再计算等号左右的值是否相等。例如证明自等律 A+0=A，可先设 A=1，会有 1+0=1，再假设 A=0，就有 0+0=0，结果都符合 A+0=A，所以 A+0=A 是正确的。

2. 常用的恒等式

在进行逻辑代数运算时，可运用前面介绍的各种定律，另外，逻辑代数中还有一些常见的恒等式，在某些情况下应用这些等式可以使逻辑代数运算更为简单快捷。下面介绍几种最常用的恒等式。

（1）$AB+A\overline{B}=A$

该恒等式证明如下

$$
\begin{aligned}
&AB+A\overline{B}\\
&=A(B+\overline{B})\\
&=A
\end{aligned}
$$

此等式又称为合并律。

（2）$A+\overline{A}B=A+B$

该恒等式证明如下

$$
\begin{aligned}
A+\overline{A}B&=A(1+B)+\overline{A}B\\
&=A+AB+\overline{A}B=A+B(A+\overline{A})\\
&=A+B
\end{aligned}
$$

以上等式说明，在一个与或表达式中，如果一项（A）的非是另一项（$\overline{A}B$）的因子，则此因子（$\overline{A}$）是多余的，故它是另一种形式的吸收律。

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

该恒等式证明如下

$$
\begin{aligned}
AB+\overline{A}C+BC&=AB+\overline{A}C+BC(A+\overline{A})\\
&=AB+\overline{A}C+ABC+\overline{A}BC\\
&=AB(1+C)+\overline{A}C(1+B)\\
&=AB+\overline{A}C
\end{aligned}
$$

此等式有一个推论

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

以上等式说明，在一个与或表达式中，如果两项分别包含 A 和 $\overline{A}$，而其余的因子（B、C）为第 3 项的因子，则第 3 项是多余的。此等式又称为添加律。

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

该恒等式证明如下

$$
\begin{aligned}
\overline{A\overline{B}+\overline{A}B}&=\overline{A\overline{B}}\cdot\overline{\overline{A}B}=(\overline{A}+\overline{\overline{B}})\cdot(\overline{\overline{A}}+\overline{B})=(\overline{A}+B)\cdot(A+\overline{B})\\
&=\overline{A}A+\overline{A}\,\overline{B}+BA+B\overline{B}\\
&=0+\overline{AB}+AB+0\\
&=AB+\overline{A}\,\overline{B}
\end{aligned}
$$

此等式还有一个推论

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

上面等式说明，由两项组成的与或表达式中（如 $A\overline{B}+\overline{A}\,\overline{B}$），如果两项中的部分因子互

补（$A\bar{B}$、$\bar{A}\bar{B}$两项中A与$\bar{A}$互补），那么将这两项其余部分（$\bar{B}$）各自取反，就得到这个函数的反函数（$AB+\bar{A}B$）。

3.3.3 逻辑表达式的化简

1. 逻辑表达式化简的意义

利用逻辑表达式可以分析数字电路，逻辑表达式又是设计数字电路的依据。但同一逻辑关系往往可以有几种不同的表达式，有的表达式简单些，有的则较复杂，如下面两个表达式

$$Y=A+\bar{A}B$$
$$Y=A+B$$

上面两个表达式的逻辑关系是完全一样的，可以明显看出第2个表达式要比第1个简单。逻辑表达式越简单，与之对应的电路也就越简单。**逻辑表达式化简就是将比较复杂的表达式转化成最简单的表达式。**那么什么才是最简表达式呢？**所谓最简表达式就是：式子中的乘积项最少；在满足乘积项最少的条件下，每个乘积项中的变量个数最少。**

2. 逻辑表达式化简的方法

根据逻辑表达式可以设计数字逻辑电路，为了使设计出来的电路最简单，需要将逻辑表达式转化为最简表达式，这就要求对逻辑表达式进行化简。逻辑表达式化简的方法主要有公式法和卡诺图法，下面仅介绍公式法。

公式法是根据逻辑代数基本定律公式和恒等式，将逻辑表达式转换为最简式。利用公式法化简逻辑表达式的常用方法有并项法、吸收法、消去法和配项法。

（1）并项法

它是利用公式$AB+A\bar{B}=A(B+\bar{B})=A$，将两个乘积项合并成一项，合并时消去互补变量。例如：

① $A(BC+B\bar{C})+A(\bar{B}C+\bar{B}\bar{C})$

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

$=AB(C+\bar{C})+A\bar{B}(C+\bar{C})$（利用公式$A+\bar{A}=1$）

$=AB+A\bar{B}$

$=A(B+\bar{B})$（利用公式$A+\bar{A}=1$）

$=A$

② $A\bar{B}C+\overline{A\bar{B}}C$

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

$=C$

（2）吸收法

它是利用公式 $A+AB=A(1+B)=A$，消去多余项。例如：

① $A\overline{B}+A\overline{B}CD(E+F)$

$=A\overline{B}[1+CD(E+F)]$（利用 $1+A=1$）

$=A\overline{B}$

② $\overline{C}+\overline{A}B\overline{C}D$

$=\overline{C}(1+\overline{A}BD)$（利用 $1+A=1$）

$=\overline{C}$

（3）消去法

它是利用 $A+\overline{A}B=A+B$，消去多余项。例如：

① $AB+\overline{A}C+\overline{B}C$

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

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

$=AB+C$

② $A\overline{B}+\overline{A}B+ABCD+\overline{A}\ \overline{B}CD$

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

$=(A\overline{B}+\overline{A}B)+CD(\overline{A\overline{B}+\overline{A}B})$（利用 $A+\overline{A}B=A+B$）

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

（4）配项法

有些表达式不能直接利用公式化简，这时往往可以用 $A=A(B+\overline{B})=AB+A\overline{B}$ 的方式将部分乘积项变为两项，或利用 $AB+\overline{A}C=AB+\overline{A}C+BC$ 增加一个项，再利用公式进行化简。例如：

① $A\overline{B}+B\overline{C}+\overline{B}C+\overline{A}B$

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

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

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

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

② $A\overline{B}+B\overline{C}+\overline{B}C+\overline{A}B$

$=A\overline{B}+B\overline{C}+\overline{B}C+\overline{A}B+\overline{A}C$（利用恒等式 $AB+\overline{A}C=AB+\overline{A}C+BC$ 增加一项 $\overline{A}C$）

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

$=A\overline{B}+\overline{A}C+B\overline{C}+\overline{A}B$（利用恒等式 $AB+\overline{A}C+BC=AB+\overline{A}C$ 消去一项 $\overline{B}C$）

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

$=A\overline{B}+\overline{A}C+B\overline{C}$（利用恒等式 $AB+\overline{A}C+BC=AB+\overline{A}C$ 消去一项 $\overline{A}B$）

3.3.4 逻辑表达式、逻辑电路和真值表相互转换

任何一个逻辑电路，它的输入与输出关系都可以用逻辑表达式表示出来，反之，任何一个逻辑表达式总可以设计出一个逻辑电路来对它进行运算；逻辑表达式可以用真值表直观显示各种输入及对应的输出情况，而根据真值表也可以写出逻辑表达式。总之，逻辑表达式、逻辑电路和真值表之间是可以相互转换的，了解它们的互相转换对设计和分析数字电路非常重要。

1. 逻辑表达式与逻辑电路的相互转换

（1）根据逻辑电路写出逻辑表达式

根据逻辑电路写出逻辑表达式比较简单，下面以图3-2所示的逻辑电路来说明。

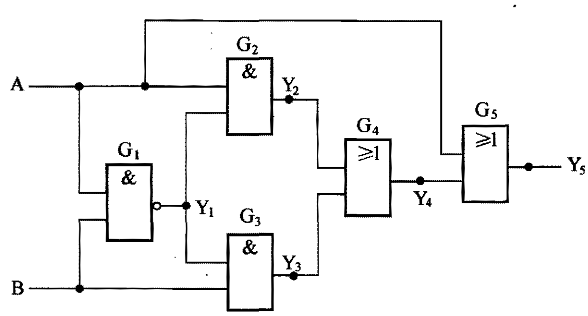


图3-2 根据逻辑电路写出逻辑表达式例图

根据逻辑电路写出逻辑表达式过程一般分为以下两步。

① 从前往后依次写出逻辑电路中各门电路的逻辑表达式。

门电路 G_1：$Y_1=\overline{AB}$；

门电路 G_2：$Y_2=AY_1$；

门电路 G_3：$Y_3=Y_1B$；

门电路 G_4：$Y_4=Y_2+Y_3$；

门电路 G_5：$Y_5=A+Y_4$。

② 依次将前一个门电路的表达式代入后一个门电路的表达式中，最终就能得到整个逻辑电路的表达式。

将 $Y_1=\overline{AB}$ 代入 $Y_2=AY_1$ 中，得到 $Y_2=A\overline{AB}$；

将 $Y_1=\overline{AB}$ 代入 $Y_3=Y_1B$ 中，得到 $Y_3=\overline{AB}B$；

将 $Y_2=A\overline{AB}$ 和 $Y_3=\overline{AB}B$ 代入到 $Y_4=Y_2+Y_3$ 中，得到 $Y_4=A\overline{AB}+\overline{AB}B$；

将 $Y_4=A\overline{AB}+\overline{AB}B$ 代入 $Y_5=A+Y_4$ 中，得到 $Y_5=A+(A\overline{AB}+\overline{AB}B)$。

最终得到的 $Y_5=A+(A\overline{AB}+\overline{AB}B)$就是图 3-2 所示逻辑电路的逻辑表达式。

（2）根据逻辑表达式画出逻辑电路

由逻辑表达式画出逻辑电路的过程与逻辑表达式的运算过程相似。下面以画逻辑表达式 $Y=(A+B)\overline{AB}$ 的逻辑电路为例来说明。

$Y=(A+B)\overline{AB}$ 的运算顺序：先将 A 和 B 进行或运算 (A+B)，同时将 A 和 B 进行与非运算（$\overline{AB}$）；然后将 A、B 或运算的结果和 A、B 与非运算的结果进行与运算，就可以得出表达式的最终结果。

$Y=(A+B)\overline{AB}$ 的逻辑电路图的绘制过程如下：先画出 A、B 的或门电路（完成 A+B 运算），再在垂直并列的位置画出 A、B 的与非门电路（完成 $\overline{AB}$ 运算），然后以这两个门电路输出端作为后级的输入端在后面画一个与门电路（完成$(A+B)\overline{AB}$ 运算），这样画出的电路就是 $Y=(A+B)\overline{AB}$ 的逻辑电路图。

$Y=(A+B)\overline{AB}$ 的逻辑电路图如图 3-3 所示。

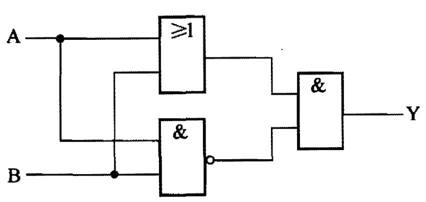


图 3-3　根据逻辑表达式画出逻辑电路例图

2. 逻辑表达式与真值表的相互转换

（1）根据逻辑表达式列出真值表

真值表是描述数字电路输入、输出逻辑关系的表格，依据真值表可以很直观地看出输入与输出之间的逻辑关系。下面以列逻辑表达式 $Y=\bar{A}B+A\bar{B}$ 的真值表为例来说明，具体过程如下：

① 首先画 1 个 2 行多列表格，第 2 行行距较大，列数与逻辑表达式的变量个数一致，$Y=\bar{A}B+A\bar{B}$ 中的变量数有 3 个，即 A、B、Y，所以列出 1 个 2 行 3 列的表格，见表 3-4。

② 将所有的变量符号写入第 1 行的表格中。

③ 将输入变量的各种可能值写入第 2 行表格内，并根据逻辑表达式写出相应的输出变量值，见表 3-5。

表 3-5 即为逻辑表达式 $Y=\overline{A}B+A\overline{B}$ 的真值表。

（2）根据真值表写出逻辑表达式

根据真值表写逻辑表达式的过程如下：

表 3-4　2 行 3 列表

表 3-5　$Y=\overline{A}B+A\overline{B}$ 真值表

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

① 从真值表上找出输出为 1 的各行，再把这些行的输入变量写成乘积的形式，如果变量值为 0，要在变量上加非。

② 把以上各行的乘积项相加。

下面以表 3-6 为例来说明由真值表写逻辑表达式的过程。

表 3-6　由真值表写出逻辑表达式例表

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

首先在真值表中找到输出变量值为 1 的各行，表中共有 3 行输出变量为 1，将这些行的输入变量写成乘积形式：$\overline{A}BC$、$A\overline{B}\,\overline{C}$、$ABC$，然后将这 3 个乘积项相加，得到表达式 $Y=A\overline{B}\,\overline{C}+\overline{A}BC+ABC$。此表达式就是真值表 3-6 的逻辑表达式。

3.3.5　逻辑代数在逻辑电路中的应用

逻辑代数对分析和设计逻辑电路有很重要的作用，特别是在设计逻辑电路时，逻辑表达式化简的应用可以使设计出来的逻辑电路简单化。

例如，根据逻辑表达式 Y=AB+AC 设计出它的逻辑电路。

方法一：并列画出 A、B 的与门电路和 A、C 的与门电路，再以这两个与门电路的输出端作为后级电路的输入端在后面画一个或门电路，画出的逻辑电路如图 3-4 所示。

方法二：观察到 Y=AB+AC 不是最简式，先对它化简，得到 Y=A(B+C)，再画出它的逻辑电路，如图 3-5 所示。

从上面的情况可以得出这样的结论：**当需要根据逻辑表达式设计逻辑电路时，首先观**

察其是否为最简表达式，如果不是，要将它化简成最简表达式，再依据最简式设计出逻辑电路。

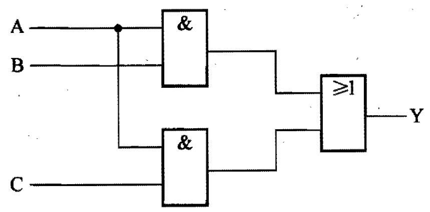


图 3-4　Y=AB+AC 逻辑电路

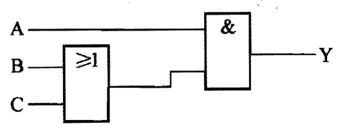


图 3-5　Y=A(B+C)逻辑电路

习题 3

一、填空题

1. 二进制数有________和________两个数码，计数时遵循_______________的原则。对于任意一个二进制数 N，它都可表示成 N=________。

2. 十六进制数有 16 个数码，分别是________，计数时遵循_______________的原则。对于任意一个十六进制数 N，它都可表示成 N=________。

3. 二进制数 110.01 转换成十进制数为________；十进制数 23 转换成二进制数为________；二进制数 1011110.101 转换成十六进制数为________；十六进制数 A6.3C 转换成二进制数为________。

4. 编码是指_______________的过程。编码是由_______________来完成的。利用________位二进制数组合表示________的编码，称为二-十进制编码，简称________码。

5. 对于十进制数 8，用 8421BCD 码表示应为________，用 2421BCD 码表示应为________，用 5421BCD 码表示应为________；用余 3 码表示应为________。

6. 两个相邻代码之间仅有________位数码不同的无权码称为格雷码，格雷码在递增或递减时，有________位发生变化，这样不容易出错，所以格雷码常用于________分辨率的设备中。

7. 奇偶校验码由________位和________位组成。奇偶校验码可分为________码和________码两种。1011 的奇校验码为________（5 位），偶校验码为________；1001 的奇校验码为________，偶校验码为________。

8. 常量是指_______________的量，变量是指_______________的量，C、Y 是_______________量，1、86 是________量，变量常用________表示，逻辑代数中的变量有________个值，分别是________。

9. 最简表达式的特点是：式子中的________最少，在满足______________的条件下，每个乘积项中的________个数最少。

10. 逻辑代数的基本运算定律主要有：①A+0=________，A · 1=________；②A+1=________，A · 0=________；③A+A=________，A · A=________；④$A+\overline{A}$=________，$A\cdot\overline{A}$=________；⑤A+AB=________，A(A+B)=________；⑥$\overline{\overline{A}}$=________；⑦A+B=________，AB=________；⑧(A+B)+C=________，(AB)C=________；⑨A(B+C)=________，A+BC=________；⑩$\overline{AB}$=________，$\overline{A+B}$=________。

二、分析题

先对 $Y=AB+\overline{A}C+\overline{B}C$ 进行化简，再根据化简后的表达式画出逻辑电路图。