
数 字 逻 辑

丁 贤 庆

ahhfdxq@163.com

Home work (P38)

☞ 1.3.3 (2) (3)

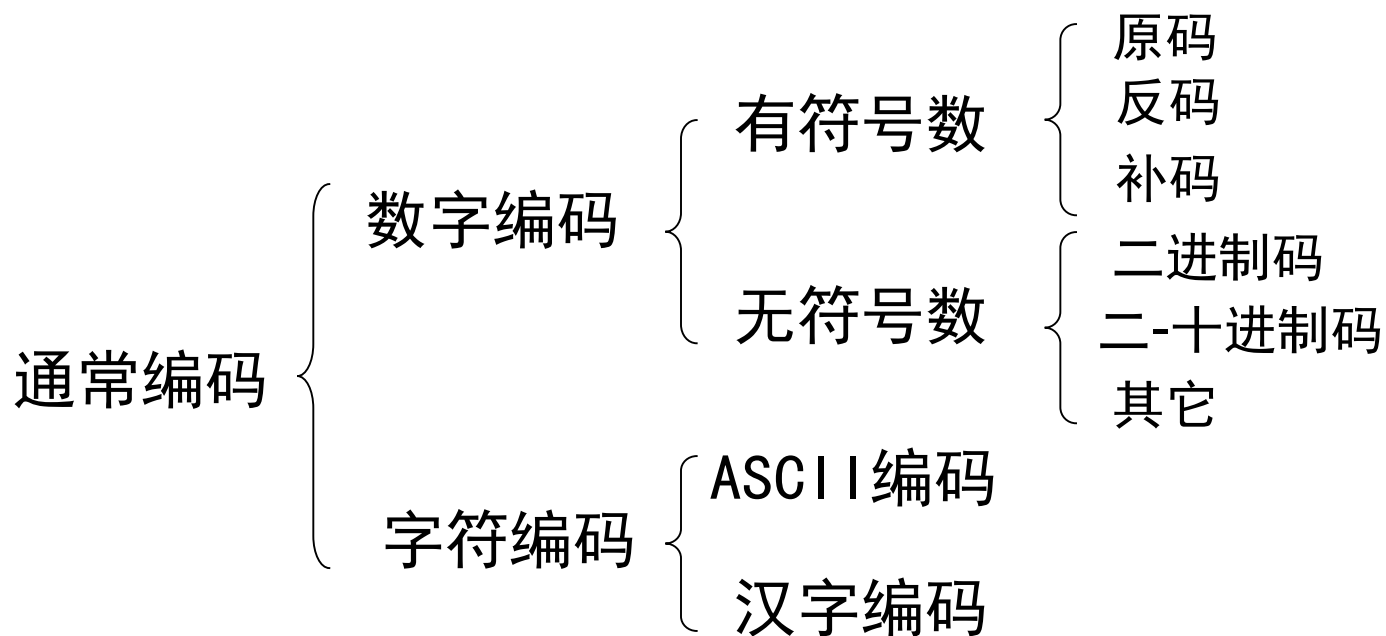
☞ 1.3.4 (2) (3)

☞ 1.4.1 (2) (3)

☞ 1.4.3 (2) (3)

1. 4 二进制编码

给一个信息或符号指定一个具体的二进制码去代表它，这一过程称为二进制编码



1.4.2 格雷码

3 位二进制码与格雷码的对照表

□ 格雷码是一种常见的无权码。

□ 特点是：任何两个相邻代码之间仅有一位取值不同。

□ 其首、尾两个代码之间也仅有一位不同，因此，格雷码又称为**循环码**。

十进制数	二进制码 $b_2b_1b_0$	格雷码 $G_2G_1G_0$
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

1.4.2 格雷码

3 位二进制码与格雷码的对照表

□ 格雷码另一个特点：

最高位的0和1只改变一次；若以最高位的0和1之间的交界为轴，其他位的代码是上下对称的。所以，格雷码又是反射码。

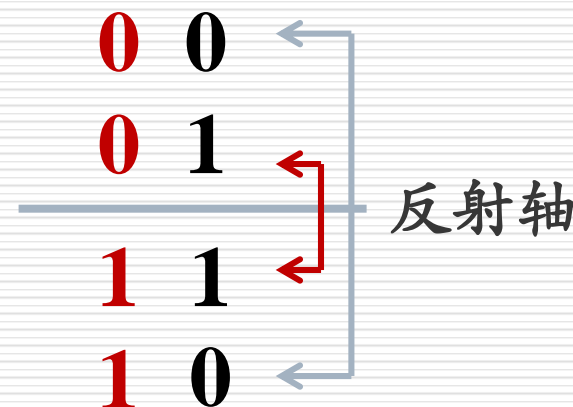
十进制数	二进制码 $b_2b_1b_0$	格雷码 $G_2G_1G_0$
0	000	000
1	001	001
2	010	011
3	011	<u>010</u>
4	100	110
5	101	111
6	110	101
7	111	100

1.4.2 格雷码

□ 构造格雷码的方法一：

2 位格雷码

- 一位格雷码有两个代码：0 和 1。
- $(n+1)$ 位格雷码中的前 2^n 个代码是将 n 位格雷码，按顺序排列，最高位补0。
- $(n+1)$ 位格雷码中的后 2^n 个代码是将 n 位格雷码，按倒序排列，最高位补1。



1.4.2 格雷码

□ 格雷码构成方法

- 一位格雷码有两个代码：0 和 1。
- $(n+1)$ 位格雷码中的前 2^n 个代码是将 n 位格雷码，按顺序排列，最高位补0。
- $(n+1)$ 位格雷码中的后 2^n 个代码是将 n 位格雷码，按倒序排列，最高位补1。

3 位格雷码

0	0	0	反射轴
0	0	1	
<hr/>			
0	1	1	反射轴
0	1	0	
<hr/>			
1	1	0	
1	1	1	
1	0	1	
1	0	0	

4 位二进制码与格雷码的对照表

二进制码 $b_3b_2b_1b_0$	格雷码 $G_3G_2G_1G_0$
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

Gray Code

格雷码

Gray Code Constructing

构造格雷码-----异或运算实现

From binary number to Gray code

The width is same, the MSB is same;

From **right to left**, if **a bit in binary number is same as its right bit**, the **gray bit is 0**. If it is different, **the gray bit is 1**.

Examples:

Binary number: 1001 0010
Gray codes: 1101 1011

二进制码 1001 0010
格雷码 1101 1011

Gray Code

Translation between Gray Code and Binary Code

Binary code: $b_{n-1} b_{n-2} b_{n-3} \dots b_1 b_0$

Gray code: $g_{n-1} g_{n-2} g_{n-3} \dots g_1 g_0$

$$g_i = b_{i+1} \oplus b_i \quad g_{n-1} = b_{n-1} \oplus 0 = b_{n-1} \quad (n-1 \text{ 是最高位})$$

5的二进制码是: 0101 \rightarrow 5的格雷码是: 0111

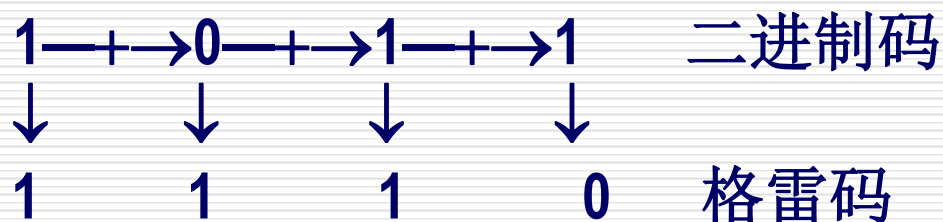
$$\because A = B \oplus C \quad \rightarrow \quad C = A \oplus B \quad \rightarrow \quad B = A \oplus C$$

$$\therefore b_i = b_{i+1} \oplus g_i \quad b_{n-1} = g_{n-1} \oplus 0 = g_{n-1} \quad (n-1 \text{ 是最高位})$$

7的格雷码是: 0100 \rightarrow 7的二进制码是: 0111

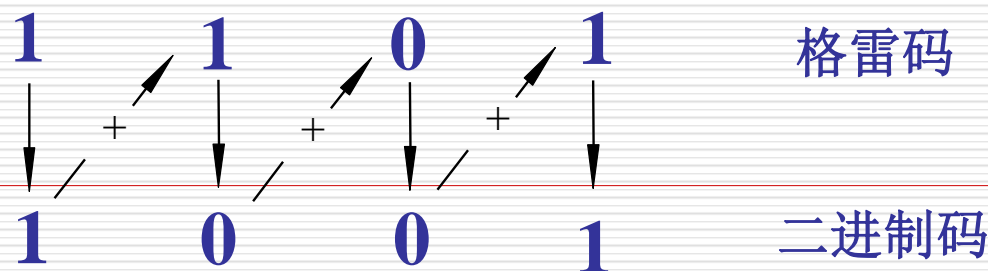
□ 二进制码到格雷码的转换

- (1) 格雷码的最高位（最左边）与二进制码的最高位相同。
- (2) 从左到右，逐一将二进制码相邻的两位相加（舍去进位），作为格雷码的下一位。



□ 格雷码到二进制码的转换

- (1) 二进制码的最高位（最左边）与格雷码的最高位相同。
- (2) 将产生的每一位二进制码，与下一位相邻的格雷码相加（舍去进位），作为二进制码的下一位。



奇偶校验码

信息位 + 1位校验位。

具有检错能力，能发现奇数个代码位同时出错的情况。

构成：信息位（可以是任一种二进制代码）及一位校验位。

校验位数码的编码方式：

“奇校验”时，使校验位和信息位所组成的每组代码中含有奇数个1；

“偶校验”时，使校验位和信息位所组成的每组代码中含有偶数个1。

二进制码1001 0010



校验位

$$(463.5)_{10} = (\quad ? \quad)_{8421\text{BCD}}$$

- ☐ A 10001100011.0101
- ☒ B 010001100011.0101
- ☐ C 10001100011.1000
- ☐ D 010001100011.1010

提交

上一题的解答：

$$(463.5)_{10} = (\quad ? \quad)_{8421\text{BCD}}$$

$$(463.5)_{10} = \left[\begin{array}{c} \boxed{0}100 \\ \hline 4 \end{array} \quad \begin{array}{c} 0110 \\ \hline 6 \end{array} \quad \begin{array}{c} 0011 \\ \hline 3 \end{array} . \quad \begin{array}{c} 0101 \\ \hline 5 \end{array} \right]_{8421\text{BCD}}$$

$$(863.2)_{10} = \left[\begin{array}{c} 1111 \\ \hline 8 \end{array} \quad \begin{array}{c} 1100 \\ \hline 6 \end{array} \quad \begin{array}{c} 0011 \\ \hline 3 \end{array} . \quad \begin{array}{c} 001\boxed{0} \\ \hline 2 \end{array} \right]_{2421\text{BCD}}$$

不能省略！

不能省略！

ASCII 码

ASCII码为美国信息交换标准代码，由七位二进制代码（ $b_7b_6b_5b_4b_3b_2b_1$ ）组成。

- ◆ 它共有128个代码，可以表示大、小写英文字母、十进制数、标点符号、运算符号、控制符号等，普遍用于计算机的键盘指令输入和数据等，如表所示。

ASCII码

表 ASCII 码表

$b_4b_3b_2b_1$	$b_7b_6b_5$							
	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

键盘上的“0”按键对应于30H，“A”按键对应于41H，
“a”按键对应于61H。

ASCII码表（续）

$b_4b_3b_2b_1$	$b_7b_6b_5$							
	000	001	010	011	100	101	110	111
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

键盘上的“回车”按键对应于0CH,就是十进制数13。

在七位ASCII码前补加一位“1”，则构成我国汉字编码使用的扩展ASCII码($1b_7b_6b_5b_4b_3b_2b_1$)。用两个扩展ASCII码编码一个汉字，最多可编码 $128 \times 128 = 16384$ 个汉字或字符。

1.5 二值逻辑变量与基本逻辑运算

◆ 逻辑运算： 当0和1表示逻辑状态时，两个二进制数码按照某种特定的因果关系进行的运算。

逻辑运算使用的数学工具是逻辑代数。

◆ 逻辑代数与普通代数:与普通代数不同,逻辑代数中的变量只有0和1两个可取值，它们分别用来表示完全两个对立的逻辑状态。

◆ 在逻辑代数中，有与、或、非三种基本的逻辑运算。

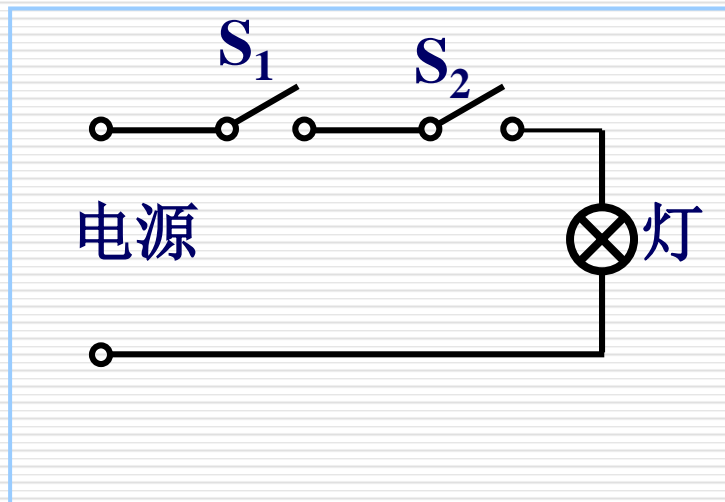
◆ 逻辑运算的描述方式： 逻辑代数表达式、真值表、逻辑图、卡诺图、波形图和硬件描述语言（HDL）等。

1. 与运算

(1) 与逻辑:只有当决定某一事件的条件全部具备时,这一事件才会发生。这种因果关系称为与逻辑关系。

与逻辑举例

电路状态表



开关 S_1	开关 S_2	灯
断	断	灭
断	合	灭
合	断	灭
合	合	亮

1. 与运算

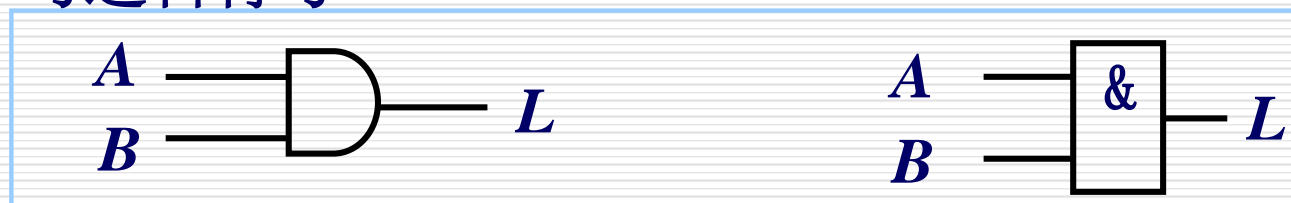
与逻辑举例状态表

开关S ₁	开关S ₂	灯	开关断开记为: 0 开关合上记为: 1
断	断	灭	
断	合	灭	
合	断	灭	
合	合	亮	

逻辑真值表

A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

与逻辑符号



美国标准

国际标准

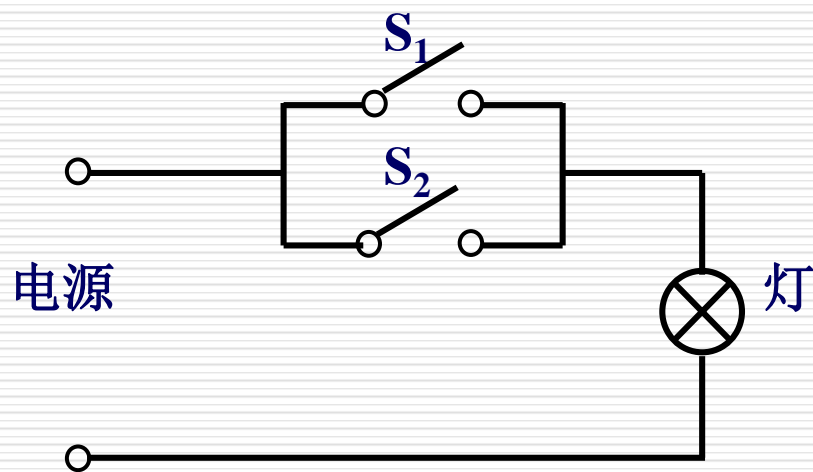
逻辑表达式

与逻辑: $L = A \cdot B = AB$

2、或运算

只要在决定某一事件的各种条件中，**有一个或几个条件具备**时，这一事件就会发生。这种因果关系称为或逻辑关系。

或逻辑举例



电路状态表

开关 S_1	开关 S_2	灯
断	断	灭
断	合	亮
合	断	亮
合	合	亮

2、或运算

或逻辑举例状态表

开关 S_1	开关 S_2	灯
断	断	灭
断	合	灭
合	断	灭
合	合	亮

逻辑真值表

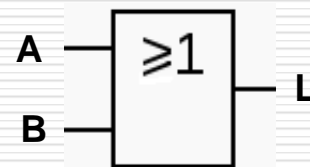
A	B	L
0	0	0
0	1	1
1	0	1
1	1	1



或逻辑符号



美国标准



国际标准

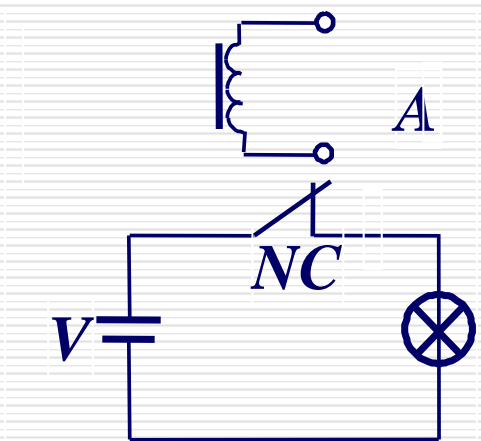
逻辑表达式

或逻辑: $L = A + B$

3. 非运算

事件发生的**条件具备**时，事件不会发生；事件发生的条件不具备时，事件发生。这种因果关系称为非逻辑关系。

非逻辑举例



非逻辑举例状态表

A	灯
不通电	亮
通电	灭

3. 非运算

非逻辑举例状态表

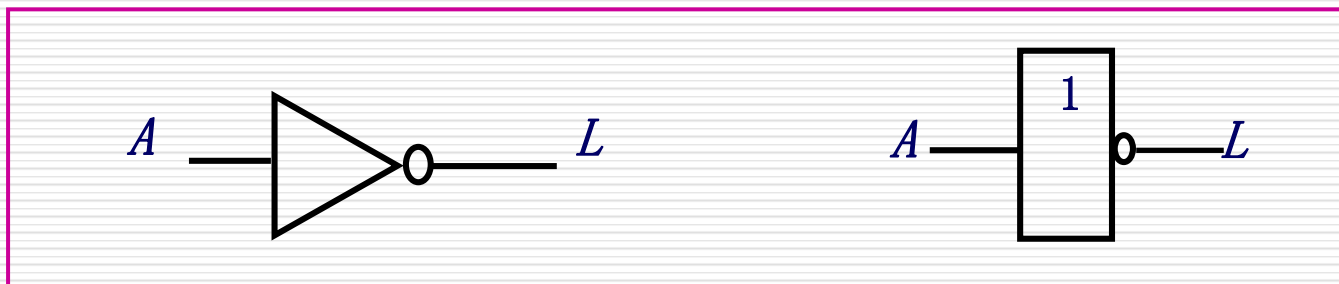
A	灯
不通电	亮
通电	灭



非逻辑真值表

A	L
0	1
1	0

非逻辑符号



逻辑表达式

$$L = \overline{A}$$

对于一个非零的 n 位二进制数，以下说法错误的是（ ）

- ☐ A 如果该数是负数，则它的反码等于 $(n \text{ 个 } 1)_2$ ，减去该数
- ☐ B 如果该数是负数，则它的补码等于 $(n \text{ 个 } 1)_2$ ，减去该数，加1
- ☒ C 如果该数是负数，则它的补码等于 $(1 \text{ 后面 } n \text{ 个 } 0)_2$ ，减去该数。减1
- ☐ D 如果该数是负数，则它的补码等于 2^n ，减去该数

提交

4. 几种常用复合逻辑运算

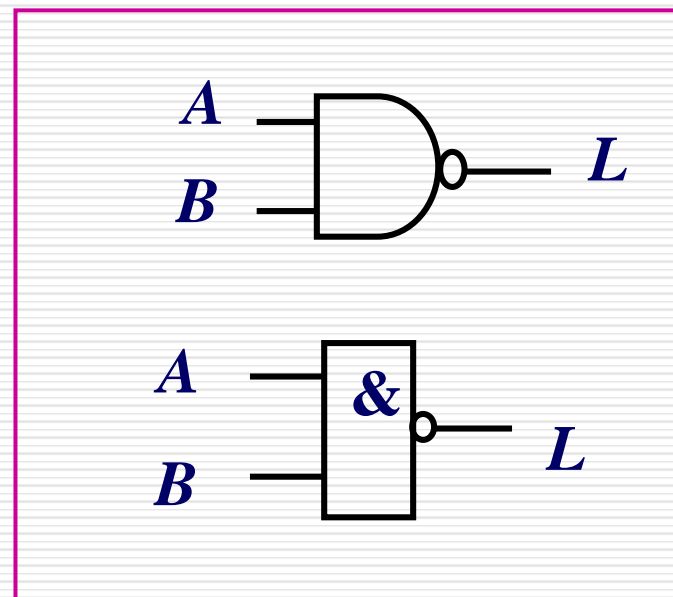
1) 与非运算

两输入变量与非
逻辑真值表

A	B	L
0	0	1
0	1	1
1	0	1
1	1	0

与非逻辑表达式

与非逻辑符号



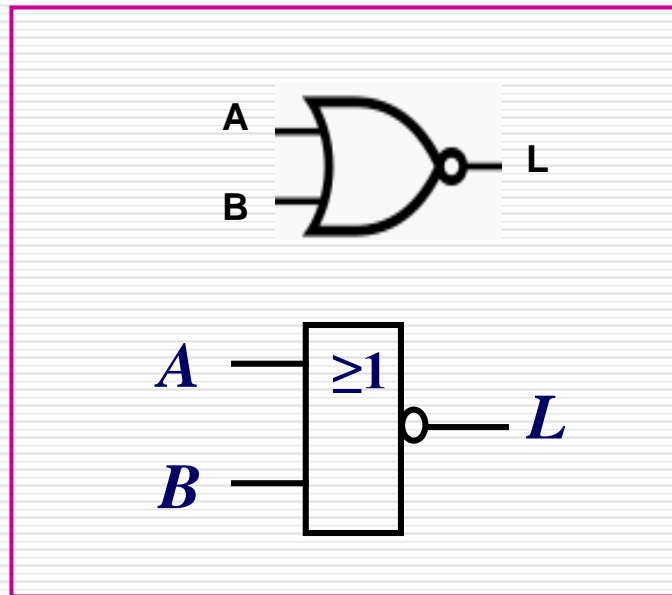
$$L = \overline{A \cdot B}$$

2)或非运算

或非逻辑符号

两输入变量或非
逻辑真值表

A	B	L
0	0	1
0	1	0
1	0	0
1	1	0



或非逻辑表达式

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

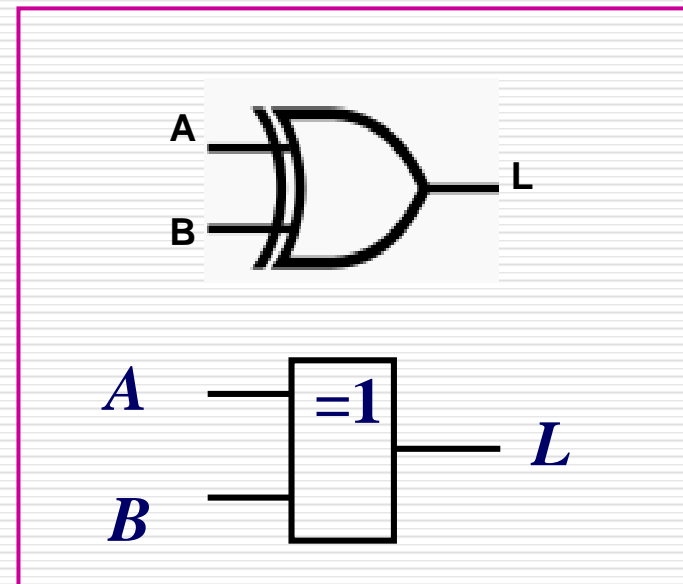
3) 异或逻辑

若两个输入变量的值相异，输出为1，否则为0。

异或逻辑真值表

A	B	L
0	0	0
0	1	1
1	0	1
1	1	0

异或逻辑符号



异或逻辑表达式 $L = A \oplus B = \bar{A}B + A\bar{B}$

4) 同或运算

若两个输入变量的值相同，输出为1，否则为0。

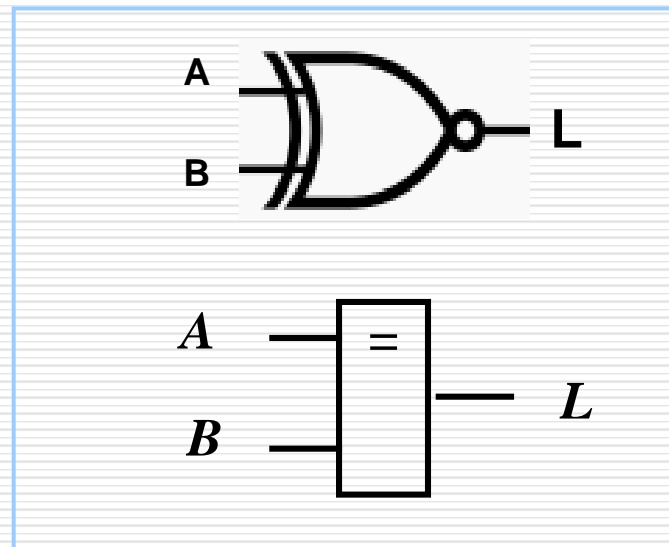
同或逻辑真值表

A	B	L
0	0	1
0	1	0
1	0	0
1	1	1

同或逻辑表达式

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

同或逻辑逻辑符号

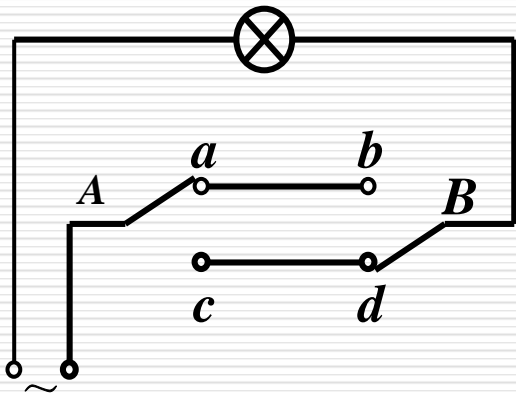


1.6 逻辑函数的建立及其表示方法

1.6.1 逻辑函数几种表示方法

1. 真值表表示

楼道灯开关示意图



确定变量、函数，并赋值

开关： 变量 A、B

灯 ： 函数 L

A、B: 向上—1 向下--0

L : 亮---1; 灭---0

逻辑抽象，列出真值表
开关状态表

开关 A	开关 B	灯
下	下	亮
下	上	灭
上	下	灭
上	上	亮

逻辑真值表

A	B	L
0	0	1
0	1	0
1	0	0
1	1	1

2、逻辑函数表达式表示。

逻辑表达式是用与、或、非等运算组合起来，表示逻辑函数与逻辑变量之间关系的逻辑代数式。

例：已知某逻辑函数的真值表，试写出对应的逻辑函数表达式。

逻辑真值表

A	B	L
0	0	1
0	1	0
1	0	0
1	1	1

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

可以将 $A=0, B=0$ 代入上式验证 $L=1$

$$L = A B$$

可以将 $A=1, B=1$ 代入上式验证 $L=1$

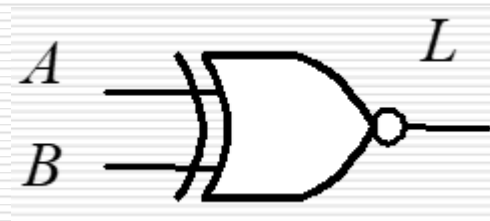
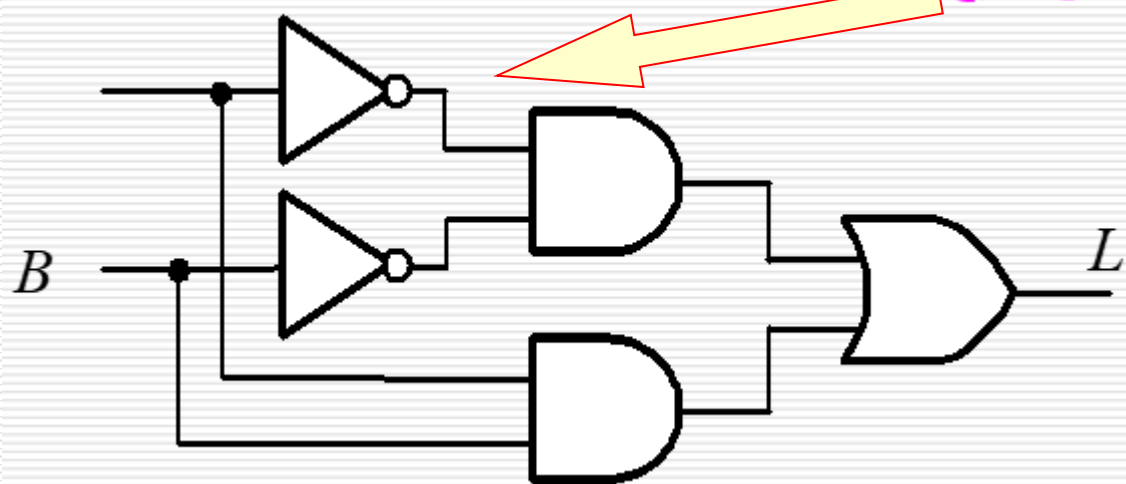
$$L = \overline{A} \overline{B} + A B$$

3. 逻辑图表示方法

◆ 用与、或、非等逻辑符号表示逻辑函数中各变量之间的逻辑关系所得到的图形称为**逻辑图**。

◆ 将逻辑函数式中所有的与、或、非运算符号用相应的逻辑符号代替，并按照逻辑运算的先后次序将这些逻辑符号连接起来，就得到图电路所对应的逻辑图

例：已知某逻辑函数表达式为 $L = \overline{A} \overline{B} + A B$ ，试画出其逻辑图

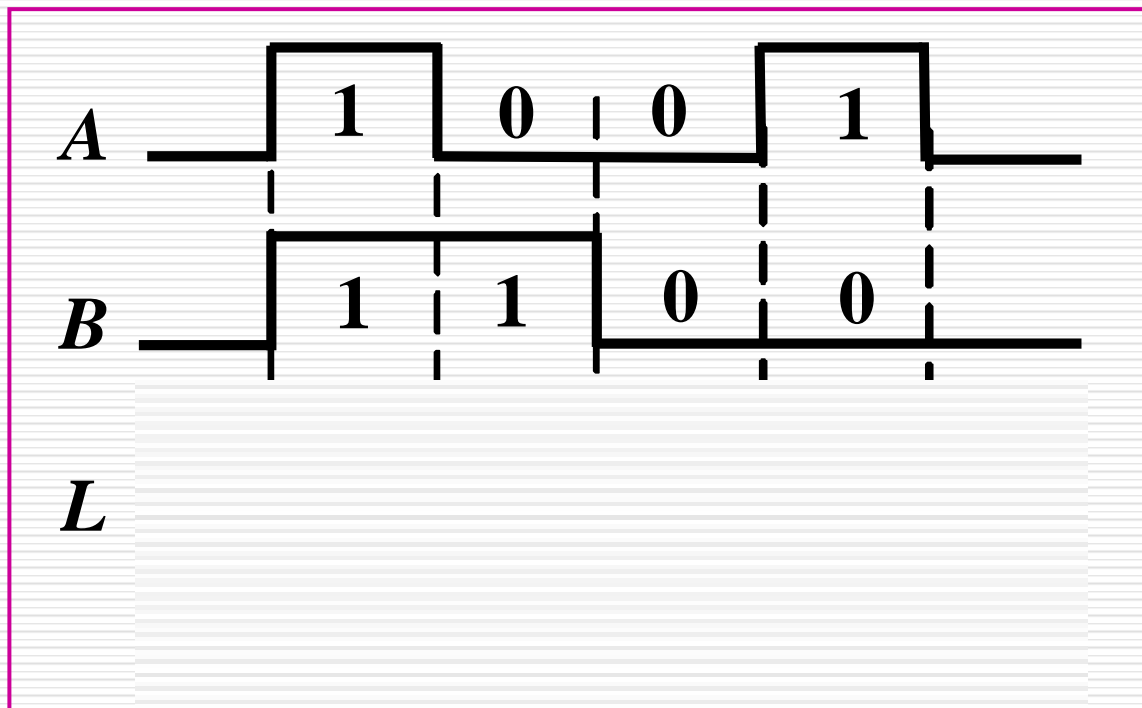


4. 波形图表示方法

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。

真值表

<i>A</i>	<i>B</i>	<i>L</i>
0	0	1
0	1	0
1	0	0
1	1	1



1.6.2 逻辑函数表示方法之间的转换

逻辑函数的真值表、逻辑函数表达式、逻辑图、波形图、卡诺图及HDL描述之间可以相互转换。这里介绍两种转换。

1.真值表到逻辑图的转换

真值表如右表。

转换步骤：

(1)根据真值表写出逻辑表达式

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

(2)化简逻辑表达式（第2章介绍）

上式不需要简化

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

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

<i>A</i>	<i>B</i>	<i>C</i>	<i>L</i>
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

(3)根据与或逻辑表达式画逻辑图

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

用与、或、非符号代替相应的逻辑符号，注意运算次序。

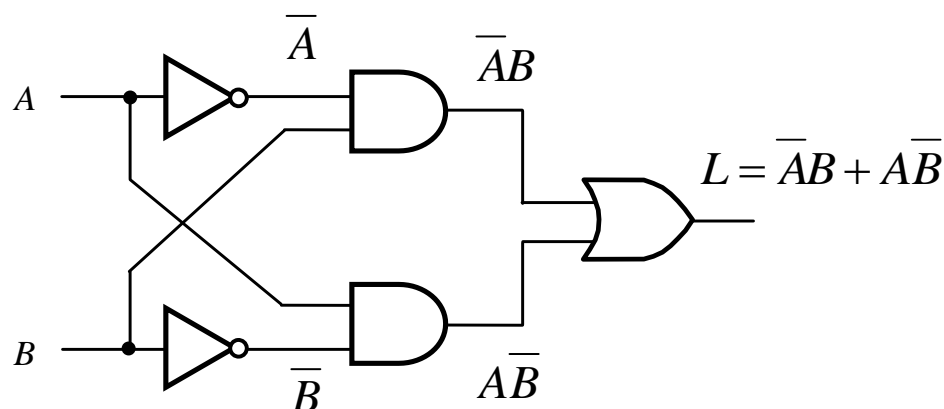
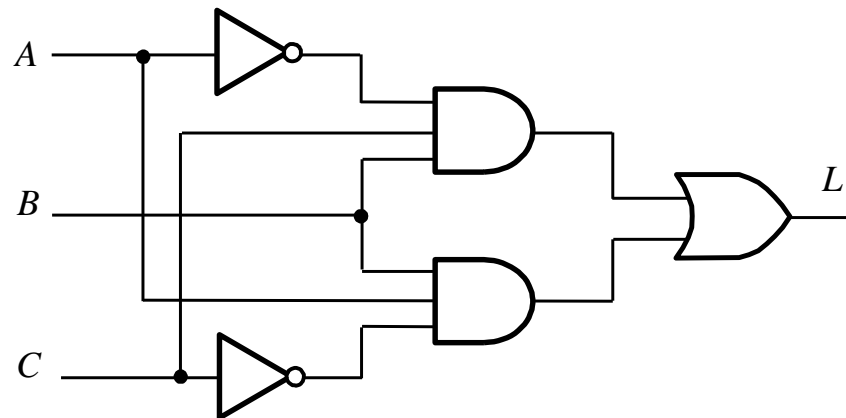
2. 逻辑图到真值表的转换

转换步骤：

(1)根据逻辑图逐级写出表达式

(2)化简变换求最简与或式

(3)将输入变量的所有取值逐一代入表达式得真值表



A	B	L
0	0	0
0	1	1
1	0	1
1	1	0