
数 字 逻 辑

丁 贤 庆

ahhfdxq@163.com

Home work (P74)

☞ 2.1.3 (1) (3)

☞ 2.2.1 (2) (3)

☞ 2.2.3 (2) (3)

☞ 2.2.7 (2)

几点说明

- 1. 作业，鼓励超额完成任务。例如：除了完成课堂布置的作业外，还利用空闲时间完成课后的其它题目。作业酌情加分。
- 2. 我们会在每次收作业的前2天，将对应作业题的答案公布出来，大家自己用红笔在自己的作业本上进行批改。只要改正了，作业中的错题不扣分。如果作业没有批改的痕迹，反而会扣分。
- 3. 作业扣分的规则是：字迹潦草，画图不规范，表述不清，错题不批改、不更正。

3. 溢出(在设计运算器时, 会遇到运算出错的情况, 例如: 溢出)

在设计运算器时, 通常会设计一个状态寄存器PSW, 通过状态寄存器PSW中的溢出位的值是0还是1, 来推测运算是否出错。

例1.3.8 在4位系统里, 试用4位二进制补码计算5+7。

解: 因为 $(5+7)_{\text{补}} = (5)_{\text{补}} + (7)_{\text{补}}$
 $= 0101 + 0111$
 $= 1100$

$$\begin{array}{rcccc} & 0 & 1 & 0 & 1 \\ + & 0 & 1 & 1 & 1 \\ \hline [1] & 1 & 0 & 0 & \end{array}$$

解决溢出的办法: 进行硬件位扩展。如果硬件不能进行扩展, 就需要在状态寄存器里将溢出状态位标为1。

4. 溢出情况的判别（两加数符号相同，与和的符号不同）

如何判断是否产生溢出？

$$\begin{array}{r} + 4 \\ +) + 3 \\ \hline + 7 \end{array} \quad \begin{array}{r} 0 \ 1 \ 0 \ 0 \\ + 0 \ 0 \ 1 \ 1 \\ \hline [0] 0 \ 1 \ 1 \ 1 \end{array}$$

$$\begin{array}{r} + 2 \\ +) + 6 \\ \hline + 8 \end{array} \quad \begin{array}{r} 0 \ 0 \ 1 \ 0 \\ + 0 \ 1 \ 1 \ 0 \\ \hline [0] 1 \ 0 \ 0 \ 0 \end{array}$$

$$\begin{array}{r} - 5 \\ +) - 3 \\ \hline - 8 \end{array} \quad \begin{array}{r} 1 \ 0 \ 1 \ 1 \\ + 1 \ 1 \ 0 \ 1 \\ \hline [1] 1 \ 0 \ 0 \ 0 \end{array}$$

$$\begin{array}{r} - 3 \\ +) - 6 \\ \hline - 9 \end{array} \quad \begin{array}{r} 1 \ 1 \ 0 \ 1 \\ + 1 \ 0 \ 1 \ 0 \\ \hline [1] 0 \ 1 \ 1 \ 1 \end{array}$$

判断溢出的方法：如果两个加数的符号相同，而和的符号与它们不同，则运算结果是错误的，产生了溢出。

在6位的系统里，采用补码运算式如下右图所示，
请问该补码运算对应的等式是：

- ☐ A $-6+7=1$
- ☒ B $-6-7=-13$
- ☐ C $2+1=3$
- ☐ D $-2-1=-3$

$$\begin{array}{r}
 1\ 1\ 1\ 0\ 1\ 0 \\
 +\ 1\ 1\ 1\ 0\ 0\ 1 \\
 \hline
 [1]\ 1\ 1\ 0\ 0\ 1\ 1
 \end{array}$$

提交

上题的解答：

在6位的系统里，采用补码运算式如下右图所示，
请问该补码运算对应的等式是：

$$\begin{aligned} \text{解： 因为 } (-6-7)_{\text{补}} &= (-6)_{\text{补}} + (-7)_{\text{补}} \\ &= 111010 + 111001 \\ &= [1] \ 110011 \\ &= (-13)_{\text{补}} \end{aligned}$$

$$\begin{array}{r} 111010 \\ + 111001 \\ \hline [1]110011 \end{array}$$

$(-6)_{\text{补}}$
 $(-7)_{\text{补}}$
 $(-13)_{\text{补}}$

此处结果的符号，与两个加数的符号位都相同，同为**1**，
所以**不存在溢出出错**的情况，也就是**正常的补码运算**。

4. 几种常用复合逻辑运算

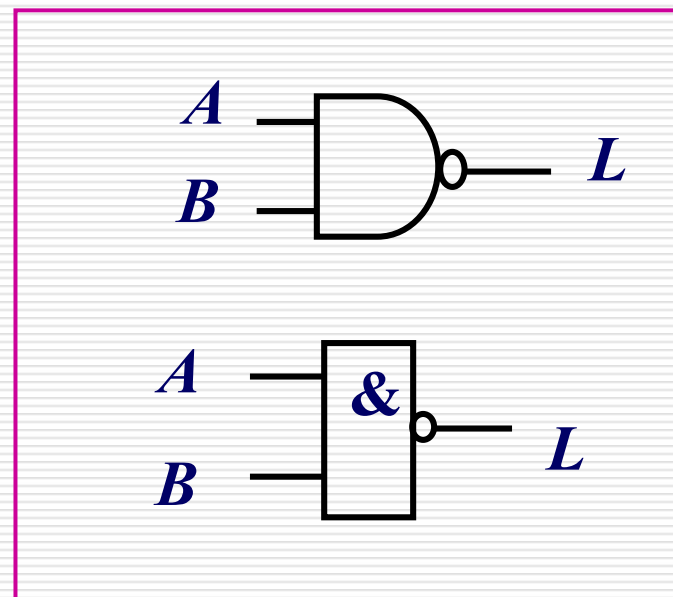
1)与非运算

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



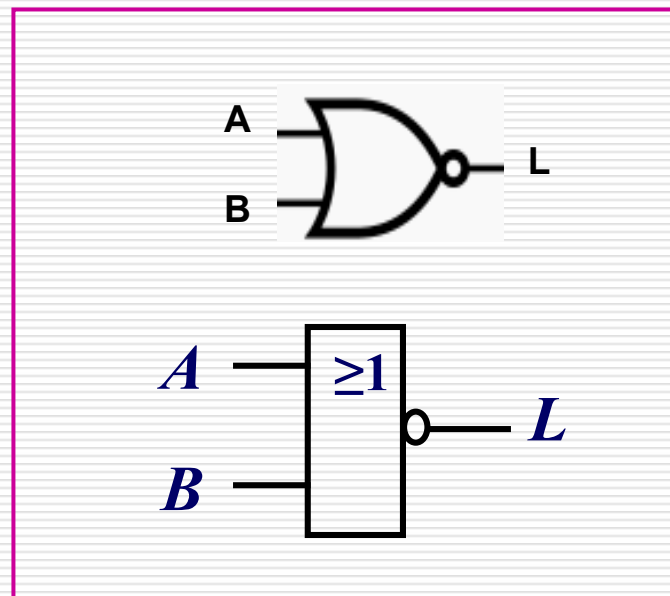
$$L = \overline{A \cdot B}$$

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

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

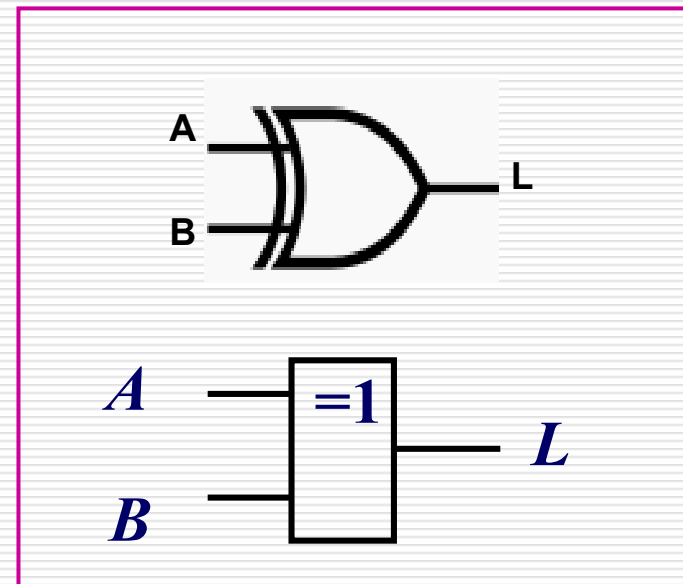
3) 异或逻辑

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

4) 同或运算

若两个输入变量的值相同，输出为1，否则为0。

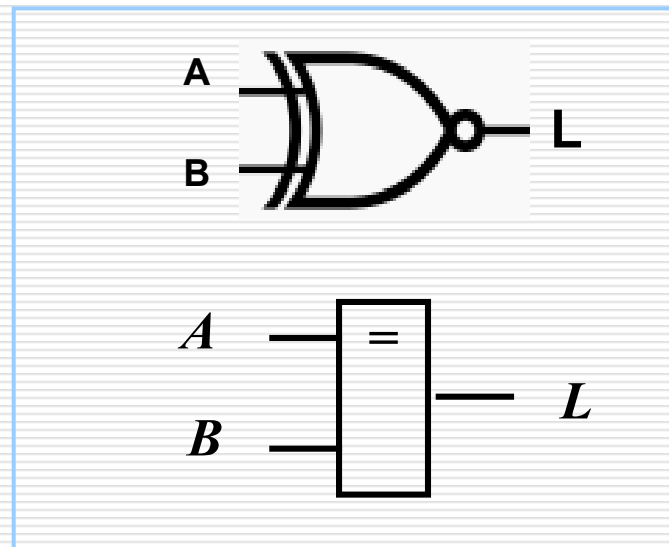
同或逻辑真值表

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

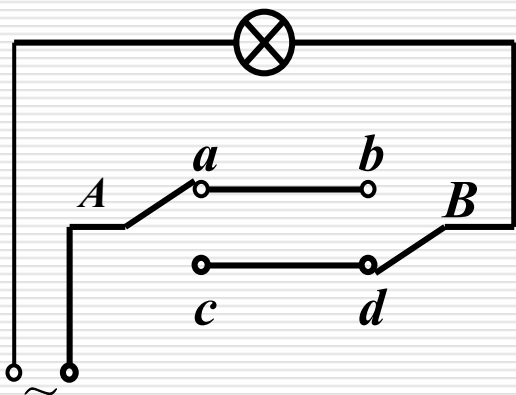


1.6 逻辑函数的建立及其表示方法

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 $L=1$

$$L = A B$$

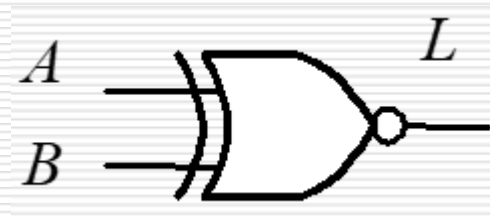
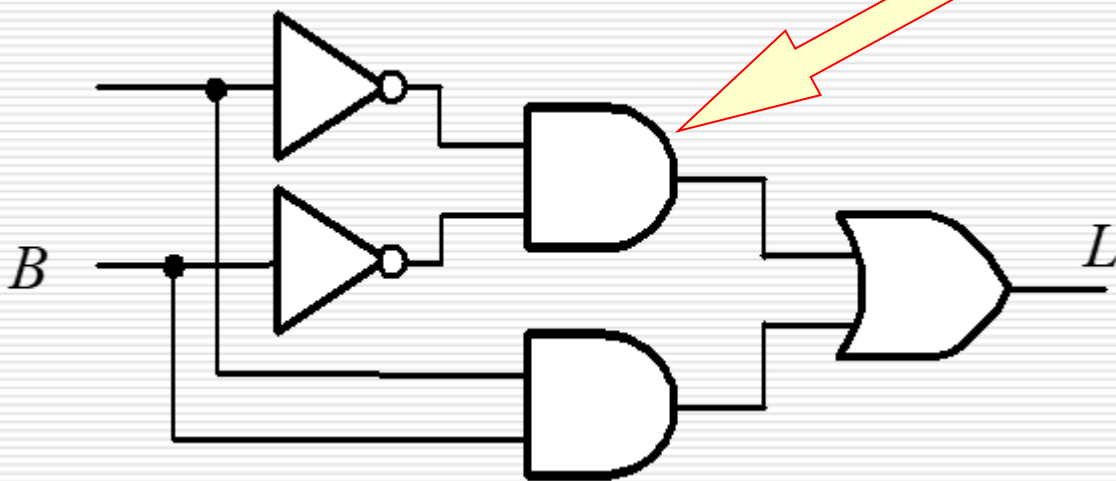
可以将 $A=1, B=1$ 代入上式验证 $L=1$

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 $L = AB + \overline{AB}$ ，试画出其逻辑图

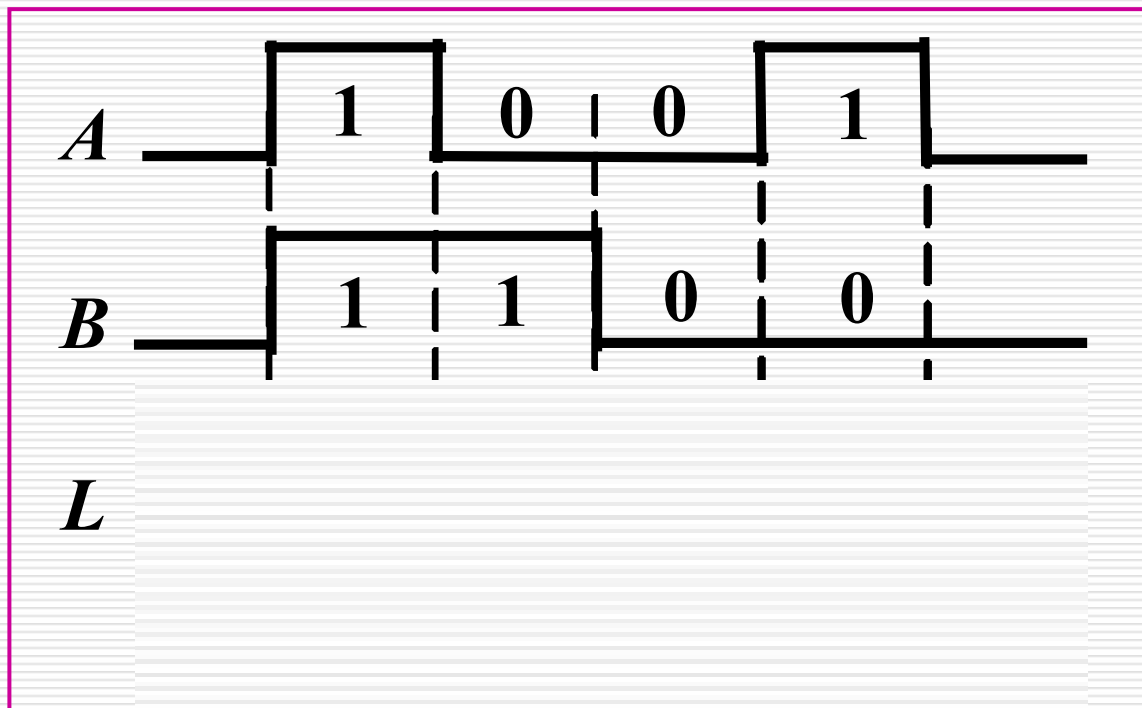


4. 波形图表示方法

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

A	B	L
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

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

(2)化简逻辑表达式（第2章介绍）

上式不需要简化

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

(3)根据与或逻辑表达式画逻辑图

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

用与、或、非符号代替相应的逻辑符号，注意运算次序。

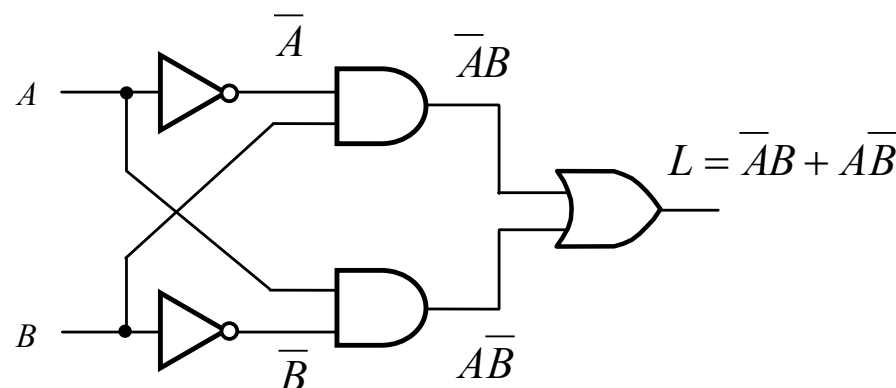
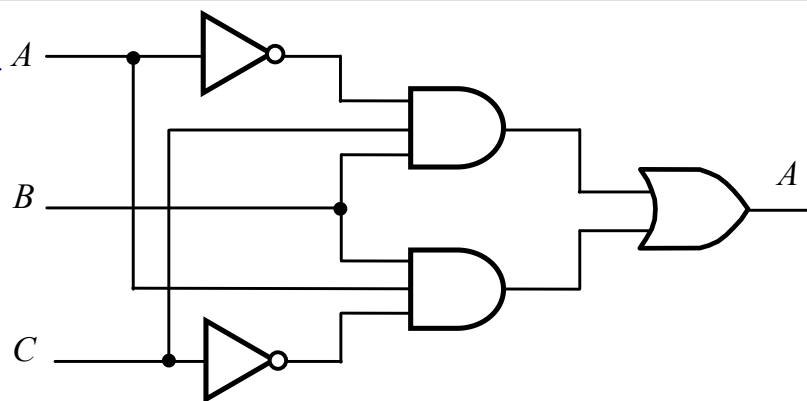
2. 逻辑图到真值表的转换

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

第二章

逻辑代数与硬件描述语言基础

2.1 逻辑代数的基本定理和规则

2.1.1 逻辑代数的基本定律和恒等式

2.1.2 逻辑代数的基本规则

2.1.1 逻辑代数的基本定律和恒等式

1、基本公式

$$0、1律: A + 0 = A \quad A + 1 = 1 \quad A \cdot 1 = A \quad A \cdot 0 = 0$$

$$互补律: A + \overline{A} = 1 \quad \overline{A} \cdot A = 0$$

$$交换律: A + B = B + A \quad A \cdot B = B \cdot A$$

$$结合律: A + B + C = (A + B) + C \quad A \cdot B \cdot C = (A \cdot B) \cdot C$$

$$分配律: A (B + C) = AB + AC \quad A + BC = (A + B)(A + C)$$

重叠律:

$$A + A = A$$

$$A \cdot A = A$$

反演律(摩根定理): $\overline{A + B} = \overline{A} \cdot \overline{B}$ $\overline{AB} = \overline{A} + \overline{B}$

吸收律

$$A + A \cdot B = A$$

$$A \cdot (A + B) = A$$

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

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

其它常用恒等式

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

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

2、基本公式的证明（真值表证明法）

例 证明： $A + \bar{A} \cdot B = A + B$

列出等式、右边的函数值的真值表

A	B	\bar{A}	$\bar{A} \cdot B$	$A + \bar{A}B$	$A + B$
0	0				
0	1				
1	0				
1	1				

例：试化简下列逻辑函数 $L = (A + B)(\bar{A} + B)$

$$L = A\bar{A} + AB + B\bar{A} + BB \text{ (分配律)}$$

$$= 0 + AB + B\bar{A} + B \quad (A \cdot \bar{A} = 0, A \cdot A = A)$$

$$= AB + B\bar{A} + B \quad (A + 0 = A)$$

$$= B(A + \bar{A} + 1) \quad [AB + AC = A(B + C)]$$

$$= B \cdot 1 = B \quad (A + 1 = 1, A \cdot 1 = A)$$

2.1.2 逻辑代数的基本规则

1. 代入规则 : 在包含变量 A 逻辑等式中, 如果用另一个函数式代入式中所有 A 的位置, 则等式仍然成立。这一规则称为代入规则。

例: $B(A + C) = BA + BC$,

用 $A + D$ 代替 A , 得

$$B[(A + D) + C] = B(A + D) + BC = BA + BD + BC$$

代入规则可以扩展所有基本公式或定律的应用范围

2. 反演规则:

对于任意一个逻辑表达式 L ，若将其中所有的与 (\cdot) 换成或 $(+)$ ，或 $(+)$ 换成与 (\cdot) ；原变量换为反变量，反变量换为原变量；将1换成0，0换成1；则得到的结果就是原函数的反函数 \bar{L} 。

- Swapping $+$ and \cdot (与或交换)
- Complementing all variables (变量取反)
- 遵循原来的运算优先 (Priority) 次序
- 不属于单个变量上的反号应保留不变，即多个变量上的反号保留不变。

例2.1.1 试求 $L = \bar{A}\bar{B} + CD + 0$ 的非函数

解：按照反演规则，得

$$\bar{L} = (A + B) \cdot (\bar{C} + \bar{D}) \cdot 1 = (A + B)(\bar{C} + \bar{D})$$

➤ Complement Rules (反演规则)

Example 1: Write the Complement function for each of the following logic functions. (写出下面函数的反函数)

$$F1 = A \cdot (B + C) + C \cdot D$$

$$\overline{F1} = ?$$

$$= (\bar{A} + \bar{B} \cdot \bar{C}) \cdot (\bar{C} + \bar{D})$$

$$F2 = \overline{A \cdot B} + C \cdot D \cdot \bar{E}$$

$$\overline{F2} = ?$$

$$= (\bar{A} + \bar{B}) \cdot (\bar{C} + \bar{D} + E)$$

$$= A \cdot B \cdot (\bar{C} + \bar{D} + E)$$

3. 对偶规则:

对于任何逻辑函数式，若将其中的与（ \cdot ）换成或（ $+$ ），或（ $+$ ）换成与（ \cdot ）；并将1换成0，0换成1；那么，所得的新的函数式就是 L 的对偶式，记作 L' 。

•对偶规则

$\cdot \Leftrightarrow +$ $0 \Leftrightarrow 1$ 所有变量不变，所有运算符变化

变换时不能破坏原来的运算顺序（优先级）。也就是说，为了保持运算的先后顺序不变，需要增加括号。

例：逻辑函数 $L = (A + \bar{B})(A + C)$ 的对偶式为

$$L' = \bar{A}\bar{B} + AC$$

•对偶原理 (Principle of Duality)

若两逻辑式相等，则它们的对偶式也相等。

4. 香农展开定理:

任何一个逻辑函数都可以重新表示为

$$f(x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) = \overline{x_i} \cdot f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \\ + x_i \cdot f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

例:利用香农展开定理将3变量函数变为2变量函数, 将变量A分解出来。 $L(A,B,C) = AB+BC+AC$

$$\begin{aligned} L(A,B,C) &= \overline{A} \cdot L(0,B,C) + A \cdot L(1,B,C) \\ &= \overline{A} \cdot (0 \cdot B + BC + 0 \cdot B) + A \cdot (1 \cdot B + BC + 1 \cdot C) \\ &= \overline{A} \cdot (BC) + A \cdot (B + C) \\ &= \overline{A} \cdot L_0 + A \cdot L_1 \end{aligned}$$

式中, $L_0 = BC$, $L_1 = B + C$

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

可以减少函数自变量的数目, 降低函数的复杂度。

2.2 逻辑函数表达式的形式

2.2.1 逻辑函数表达式的形式

2.2.2 最小项与最小项表达式

2.2.3 最大项与最大项表达式

2.2 逻辑函数表达式的形式

2.2.1 逻辑函数表达式的基本形式

1、与-或表达式

若干与项进行或逻辑运算构成的表达式。由与运算符和或运算符连接起来。

$$L = A \cdot C + \bar{C} \cdot D$$

2、或-与表达式

若干或项进行与逻辑运算构成的表达式。由或运算符和与运算符连接起来。

$$L = (A + C) \cdot (B + \bar{C}) \cdot D$$

通常表达式为混合形式

$$L = A \cdot (B \cdot C + \bar{B} \cdot \bar{C}) + A \cdot (B \cdot \bar{C} + \bar{B} \cdot C)$$

经过变换可转换为上述两种基本形式

2.2.2 最小项与最小项表达式

1. 最小项的定义和性质

n 个变量 X_1, X_2, \dots, X_n 的最小项是 n 个因子的乘积，每个变量都以它的原变量或非变量的形式在乘积项中出现，且仅出现一次。一般 n 个变量的最小项应有 2^n 个。

例如， A 、 B 、 C 三个逻辑变量的最小项有（ $2^3=$ ）8个，即

$$\overline{A}\overline{B}\overline{C}、\overline{A}\overline{B}C、\overline{A}B\overline{C}、\overline{A}BC、A\overline{B}\overline{C}、A\overline{B}C、AB\overline{C}、ABC$$

$$\overline{A}B、\overline{A}BC\overline{A}、A(B+C) \text{ 等则不是最小项。}$$

2、最小项的性质 三个变量的所有最小项的真值表

A	B	C	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC
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

- 对于任意一个最小项，只有一组变量取值使得它的值为1；
- 任意两个最小项的乘积为0；
- 全体最小项之和为1。

m : minterm

3、最小项的编号

三个变量的所有最小项的真值表

			m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
A	B	C	$\overline{A}\overline{B}\overline{C}$	$\overline{A}B\overline{C}$	$A\overline{B}\overline{C}$	ABC	$\overline{A}\overline{B}C$	$\overline{A}BC$	$AB\overline{C}$	ABC
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

最小项的表示：通常用 m_i 表示最小项， m 表示minterm，下标 i 为最小项号。反变量 $\rightarrow 0$ 原变量 $\rightarrow 1$ ，就形成下标 i 的值。

2. 最小项表达式

由若干最小项相或构成的表达式，也称为标准与-或式。

- 为“与或”逻辑表达式；
- 在“与或”式中的每个乘积项都是最小项。

例1 将 $L(A,B,C) = AB + \bar{A}C$ 化成最小项表达式

$$\begin{aligned} L(A,B,C) &= AB(C + \bar{C}) + \bar{A}(B + \bar{B})C \\ &= ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}C \\ &= m_7 + m_6 + m_3 + m_1 \\ &= \sum m(7, 6, 3, 1) \end{aligned}$$

例2 将 $L(A, B, C) = \overline{(AB + \overline{A}\overline{B} + \overline{C})\overline{AB}}$ 化成最小项表达式

a. 去掉非号

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

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

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

b. 去括号

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

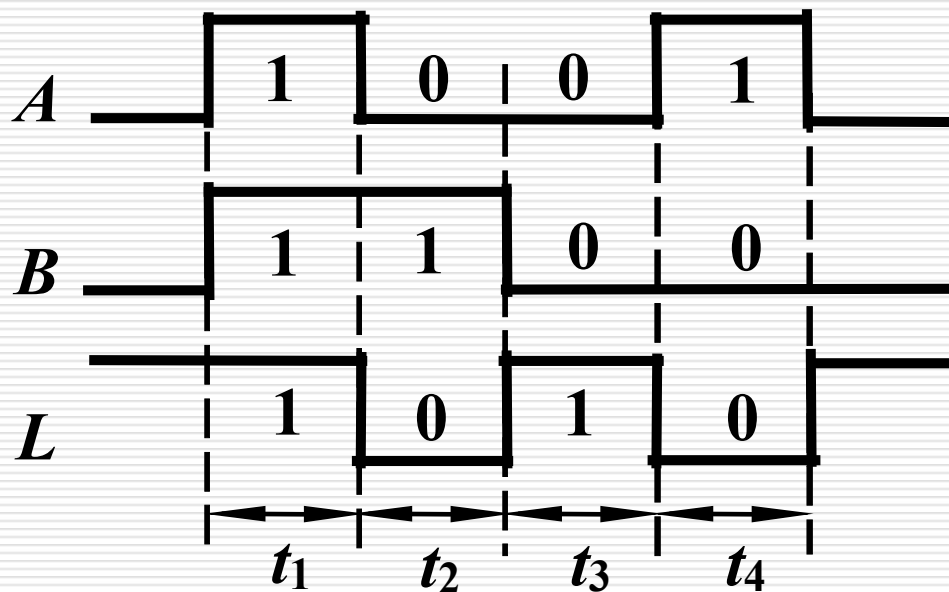
$$= \overline{A}BC + A\overline{B}C + AB(C + \overline{C})$$

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

$$= m_3 + m_5 + m_7 + m_6 = \sum m(3, 5, 6, 7)$$

已知一个逻辑电路的波形图如下图所示，请问该图形对应的逻辑关系式正确的为（ ）

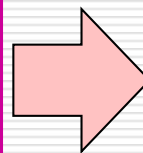
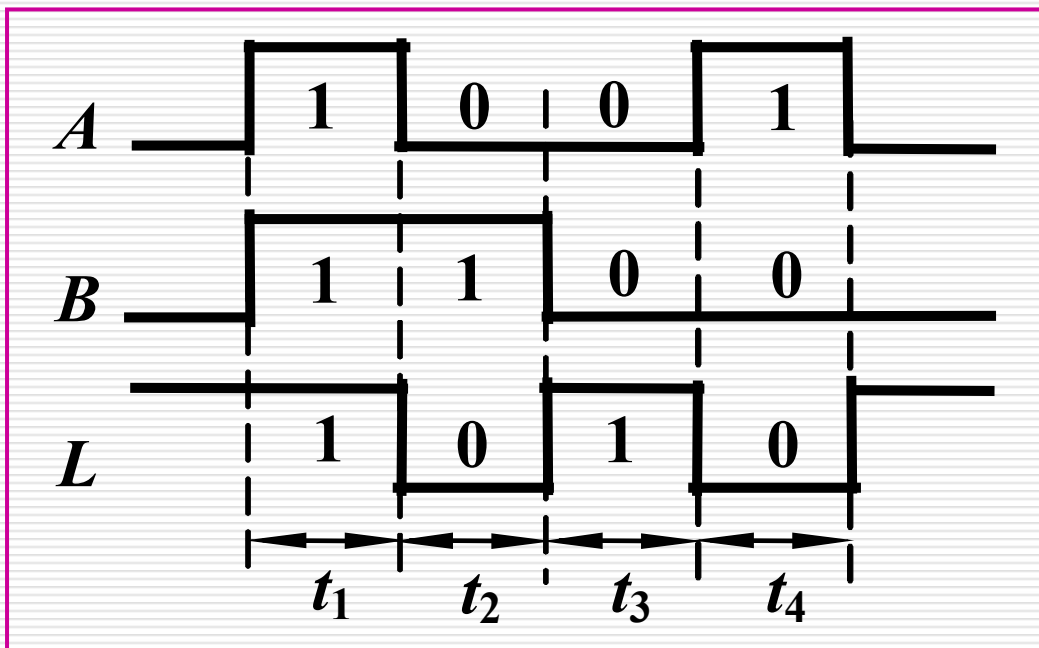
- ☐ A $L = \overline{A \cdot B}$
- ☐ B $L = A \cdot B$
- ☐ C $L = A \oplus B$
- ☒ D $L = \overline{A} \overline{B} + AB$



提交

上页题目的解法

根据波形图，写出真值表。根据真值表，写出表达式。



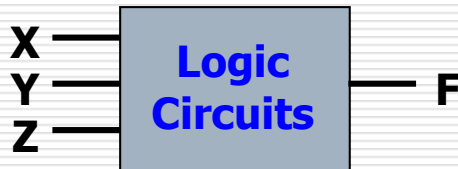
真值表

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

Switching Algebra

逻辑代数

➤ Standard Representations of Logic Functions



How to obtain the logic expression from True Table?

Canonical Sum (标准和) :

A sum of the minterms corresponding to truth-table rows (input combinations) for which the function produces a **1** output.

$$F = \bar{X} \cdot \bar{Y} \cdot \bar{Z} + \bar{X} \cdot Y \cdot Z + X \cdot \bar{Y} \cdot Z + X \cdot Y \cdot \bar{Z} + X \cdot Y \cdot Z$$

$$= \sum_{X,Y,Z} (0, 3, 4, 6, 7)$$

True Table

Row	X	Y	Z	F	Minterm
0	0	0	0	1	$\bar{X} \cdot \bar{Y} \cdot \bar{Z}$
1	0	0	1	0	$\bar{X} \cdot \bar{Y} \cdot Z$
2	0	1	0	0	$\bar{X} \cdot Y \cdot \bar{Z}$
3	0	1	1	1	$\bar{X} \cdot Y \cdot Z$
4	1	0	0	1	$X \cdot \bar{Y} \cdot \bar{Z}$
5	1	0	1	0	$X \cdot \bar{Y} \cdot Z$
6	1	1	0	1	$X \cdot Y \cdot \bar{Z}$
7	1	1	1	1	$X \cdot Y \cdot Z$

1 → 原变量 0 → 反变量

2.2.2 最大项与最大项表达式

1. 最大项的定义和性质

n 个变量 X_1, X_2, \dots, X_n 的最大项是 n 个因子或运算，每个变量都以它的原变量或非变量的形式在或项中出现，且仅出现一次。一般 n 个变量的最大项应有 2^n 个。

例如， A 、 B 、 C 三个逻辑变量的最大项有（ $2^3=$ ）8个，即

$$\begin{aligned} &(\bar{A} + \bar{B} + \bar{C}), (\bar{A} + \bar{B} + C), (\bar{A} + B + \bar{C}), (\bar{A} + B + C), \\ &(A + \bar{B} + \bar{C}), (A + \bar{B} + C), (A + B + \bar{C}), (A + B + C) \end{aligned}$$

1. 最大项的定义和性质

最大项的表示：通常用 M_j 表示最大项， M 表示maxterm, 下标 j 为最大项号。原变量 $\rightarrow 0$ ，反变量 $\rightarrow 1$ ，就形成下标 j 的值。

最大项的性质：

- 对于任意一个最大项，只有一组变量取值使得它的值为0；
- The sum of any two different maxterms is 1
(任意两个不同最大项的和为1)
- Product of all maxterms is 0
(全体最大项之积为0)

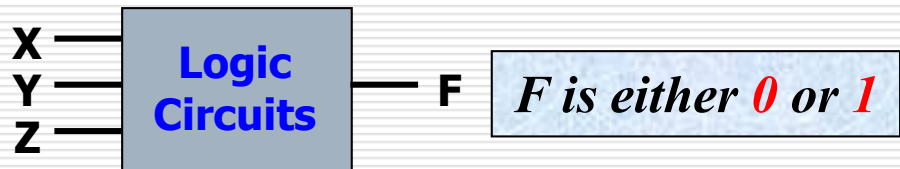
2. 最小项和最大项的关系

两者之间为互补关系： $m_i = \overline{M_i}$ ，或者 $M_i = \overline{m_i}$

Switching Algebra

逻辑代数

➤ Standard Representations of Logic Functions



A **Maxterm** can be defined as a **sum term** that is **0** in exactly one row (行).



Why?

Easy to get the product-of-sums

0 → 原变量 1 → 反变量

How to define the Maxterms (最大项)?

Row	X	Y	Z	Maxterm
0	0	0	0	$X + Y + Z$
1	0	0	1	$X + Y + \bar{Z}$
2	0	1	0	$X + \bar{Y} + Z$
3	0	1	1	$X + \bar{Y} + \bar{Z}$
4	1	0	0	$\bar{X} + Y + Z$
5	1	0	1	$\bar{X} + Y + \bar{Z}$
6	1	1	0	$\bar{X} + \bar{Y} + Z$
7	1	1	1	$\bar{X} + \bar{Y} + \bar{Z}$

0 → 原变量 1 → 反变量

例：逻辑电路的真值表如右，写出最小项和最大项表达式。

$L = \overline{A}BC$ 将 $A=0, B=1, C=1$ 代入左式, $\Rightarrow L=1$
将 $L=1$ 的各个最小项相加, 只要其中的一项为1, 则必有 $L=1$ 成立。

所以: L 等于所有能使 $L=1$ 的最小项的加运算。

最小项表达式:

将 $L=1$ 的各个最小项相加

$$\begin{aligned} L(A, B, C) &= m_3 + m_5 + m_6 \\ &= \sum m(3, 5, 6) \\ &= \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} \end{aligned}$$

<i>A</i>	<i>B</i>	<i>C</i>	<i>L</i>	
0	0	0	0	M_0
0	0	1	0	M_1
0	1	0	0	M_2
0	1	1	1 $\rightarrow m_3$	
1	0	0	0	M_4
1	0	1	1 $\rightarrow m_5$	
1	1	0	1 $\rightarrow m_6$	
1	1	1	0	M_7

根据真值表中 $L=1$ 的情况, 可以写出最小项的表达式。

例：逻辑电路的真值表如右，写出最小项和最大项表达式。

$L=A+B+C$ 将 $A=0, B=0, C=0$ 代入左式, $\Rightarrow L=0$

将 $L=0$ 的各个最大项相乘，只要其中的一项为0，则必有 $L=0$ 成立。

所以：L等于所有能使 $L=0$ 的最大项的乘积。

最大项表达式：

将 $L=0$ 的各个最大项相乘

$$L(A, B, C) = M_0 \cdot M_1 \cdot M_2 \cdot M_4 \cdot M_7$$

$$= \prod M(0, 1, 2, 4, 7)$$

$$= (A + B + C) \cdot (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (\bar{A} + B + C) \cdot (\bar{A} + \bar{B} + \bar{C})$$

A	B	C	L	
0	0	0	0	M_0
0	0	1	0	M_1
0	1	0	0	M_2
0	1	1	1 $\rightarrow m_3$	
1	0	0	0	M_4
1	0	1	1 $\rightarrow m_5$	
1	1	0	1 $\rightarrow m_6$	
1	1	1	0	M_7

根据真值表中 $L=0$ 的情况，可以写出最大项的表达式。

➤ Standard Representations of Logic Functions

Relationship of Minterm and Maxterm

$$\overline{m_i} = M_i$$

$$\overline{M_i} = m_i$$

Row	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

$$F = \sum_{A,B,C} (3, 5, 6)$$

$$F = \prod_{A,B,C} (0, 1, 2, 4, 7)$$

标号互缺

根据F=1的情况写出:

$$F = m_3 + m_5 + m_6$$

根据 $\overline{F} = 1$ 的情况写出:

$$\overline{F} = m_0 + m_1 + m_2 + m_4 + m_7$$

上式两边同时取反, 得:

$$F = M_0 * M_1 * M_2 * M_4 * M_7$$

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

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

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

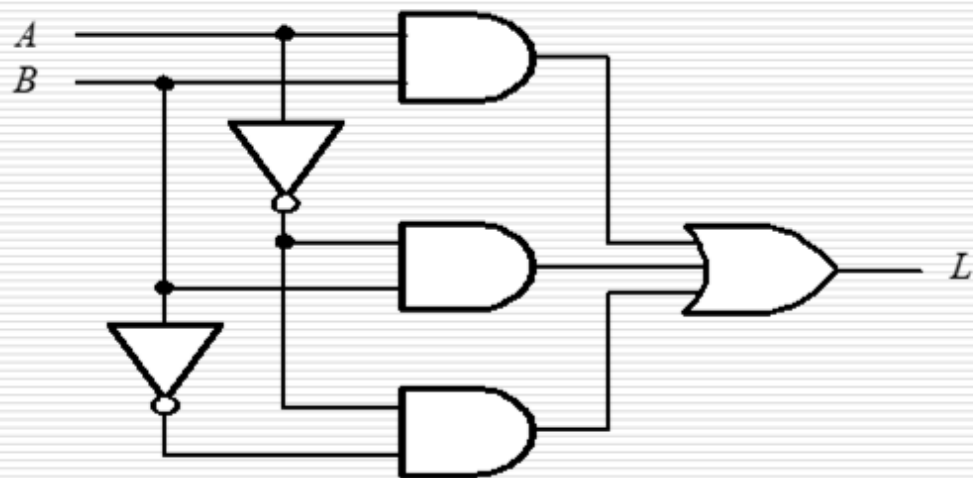
2.3 逻辑函数的代数化简法

2.3.1 逻辑函数的最简形式

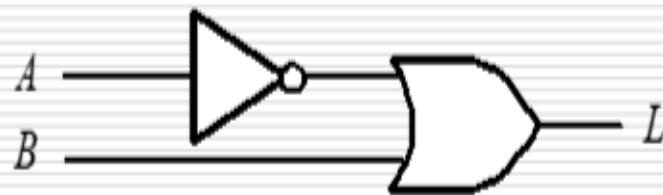
2.3.2 逻辑函数的代数化简法

2.3 逻辑函数的代数法化简

化简的目的：降低电路实现的成本，以较少的门实现电路。



(a) 标准积之和的电路



(b) 成本最低的电路

图 (a) 和图 (b) 的电路逻辑功能相同，但图 (b) 电路简单可靠性高，成本低。

2.3.1 逻辑函数的最简形式

逻辑函数有不同形式，如与-或表达式、与非-与非表达式、或-与表达式、或非-或非表达式以及与-或-非表达式等。

将其中包含的与项数最少，且每个与项中变量数最少的与-或表达式称为**最简与-或表达式**。

$$L = AC + \bar{C}D$$

“与-或” 表达式

$$= \overline{\overline{A} \overline{C}} \cdot \overline{\overline{C} \overline{D}}$$

“与非-与非” 表达式

$$= (A + \bar{C})(C + D)$$

“或-与” 表达式

$$= \overline{(A + \bar{C}) + (C + D)}$$

“或非-或非” 表达式

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

“与-或-非” 表达式

2.3.2 逻辑函数的代数化简法

1、逻辑函数的化简

化简的主要方法：

1. 公式法（代数法）
2. 图解法（卡诺图法）

代数化简法：

运用逻辑代数的基本定律和恒等式进行化简的方法。

并项法： $A + \bar{A} = 1$

$$L = \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} = \bar{A}\bar{B}(C + \bar{C}) = \bar{A}\bar{B}$$

吸收法: $A + AB = A$

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

消去法: $A + \bar{A}B = A + B$

$$L = AB + \underline{\bar{A}C} + \underline{\bar{B}C} = AB + (\bar{A} + \bar{B})C$$

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

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

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

配项法: $A + \bar{A} = 1$

$$L = AB + \bar{A}\bar{C} + \underline{B\bar{C}} = AB + \bar{A}\bar{C} + (\underline{A + \bar{A}})B\bar{C}$$

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

$$= (\underline{AB + AB\bar{C}}) + (\underline{\bar{A}\bar{C} + \bar{A}\bar{C}B})$$

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

2、逻辑函数形式的变化

通常在一片集成电路芯片中只有一种门电路，为了减少门电路的种类，需要对逻辑函数表达式进行变换。

例：已知

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

(1) 求最简的与-或式，并画出相应的逻辑图；

(2) 画出仅用与非门实现的电路。

解：

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

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

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

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

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

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

