

# 《数字逻辑》

## （第四讲）

厦门大学信息学院软件工程系 曾文华

2023年10月24日

# 课程内容

- 全书共9章：

第1章 基本知识

第2章 逻辑代数基础

第3章 集成门电路与触发器

第4章 组合逻辑电路

第5章 同步时序逻辑电路

第6章 异步时序逻辑电路

第7章 中规模通用集成电路及其应用

第8章 可编程逻辑器件

第9章 综合应用举例



# 第4章 组合逻辑电路

- 4.1 组合逻辑电路分析
- 4.2 组合逻辑电路设计
- 4.3 组合逻辑电路的险象

- 数字系统中的逻辑电路按其是否具有记忆功能分为**组合逻辑电路**和**时序逻辑电路**。
- **组合逻辑电路**是指电路在任何时刻产生的稳定输出值，仅仅取决于该时刻各输入值的组合，而与过去的输入值无关。
- 图4.1：组合逻辑电路的一般结构。
- 组合逻辑电路的**特点**：
  - ① 由逻辑门电路组成，不包含任何记忆元件；
  - ② 信号是单向传输的，不存在任何反馈回路。

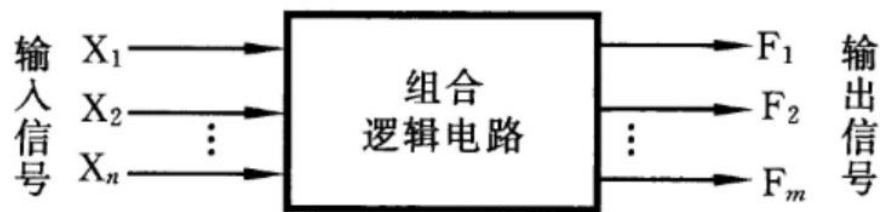


图 4.1 组合逻辑电路的一般结构

# 4.1 组合逻辑电路分析

4.2.1 分析方法概述

4.2.2 分析举例

## • 4.1.1 分析方法概述

- 所谓**逻辑电路分析**是指：对一个给定的逻辑电路，找出其输出与输入之间的逻辑关系。
- 组合逻辑电路分析的一般步骤：
  - 1、根据逻辑电路图写出输出函数表达式；
  - 2、化简输出函数表达式（见第2章的“2.4 逻辑函数化简”）；
  - 3、列出输出函数真值表；
  - 4、功能评述。

## • 4.1.2 分析举例

— **例4.1** 分析图4.2(a)所示组合逻辑电路。

— 解：

• 1、根据逻辑电路图写出输出函数表达式：

- $P_1 = \neg A$
- $P_2 = B + C$
- $P_3 = \neg(B \cdot C)$
- $P_4 = \neg(P_1 \cdot P_2) = \neg(\neg A \cdot (B + C))$
- $P_5 = \neg(A \cdot P_3) = \neg(A \cdot \neg(B \cdot C))$
- $F = \neg(P_4 \cdot P_5) = \neg(\neg(\neg A \cdot (B + C)) \cdot \neg(A \cdot \neg(B \cdot C)))$

• 2、化简输出函数表达式（采用代数化简法）：

$$F = \neg(\neg(\neg A \cdot (B + C)) \cdot \neg(A \cdot \neg(B \cdot C))) = (A \oplus B) + (A \oplus C)$$

• 3、列出输出函数真值表（见表4.1）。

• 4、功能评述

- 根据真值表，可知该电路仅当A、B、C同为0，或者同为1时，输出为0；其他情况，输出均为1。  
该电路通常称为“**不一致电路**”。
- 此外，根据化简后的函数表达式，可以用图4.2(b)实现该电路，并且图4.2(b)要比图4.2(a)简单。

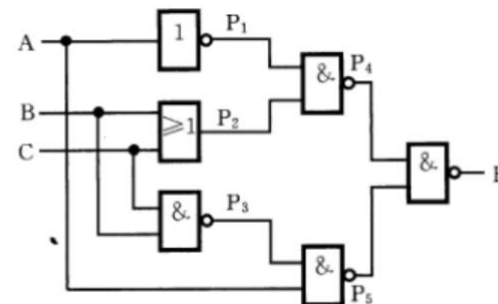


图4.2(a) 逻辑电路

表 4.1 真值表

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

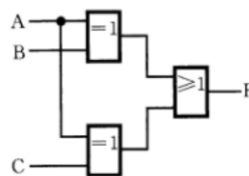


图4.2(b) 逻辑电路

— **例4.2** 分析图4.3(a)所示组合逻辑电路。

— 解：

- 1、根据逻辑电路图写出输出函数表达式：

- $S = \neg((A \cdot B) \cdot A) \cdot \neg((A \cdot B) \cdot A)$

- $C = \neg(A \cdot B)$

- 2、化简输出函数表达式（采用代数化简法）：

- $S = \neg((A \cdot B) \cdot A) \cdot \neg((A \cdot B) \cdot A) = A \cdot B + \neg A \cdot B = A \oplus B$

- $C = \neg(A \cdot B) = A \cdot B$

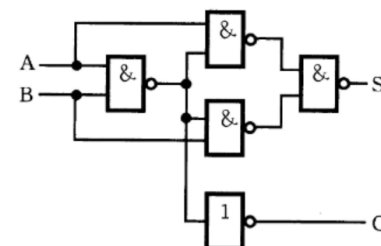


图4.3(a) 逻辑电路

- 3、列出输出函数真值表（见表4.2）。

- 4、功能评述

- 根据真值表，可知该电路为**1位半加器HA**（Half Adder），其中A、B为输入，S为“和”，C为“进位”。半加器的逻辑符号如图4.3(b)所示。

表 4.2 真值表

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

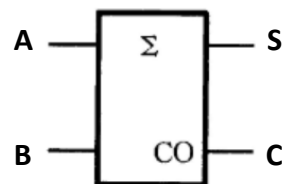


图4.3(b) 半加器的逻辑符号

— **例4.3** 分析图4.4所示组合逻辑电路。已知A、B、C、D为8421码，说明该电路的功能。

— 解：

• 1、根据逻辑电路图写出输出函数表达式：

—  $W = A \oplus B \cdot (C + D)$

—  $X = B \oplus (C + D)$

—  $Y = C \oplus D$

—  $Z = /D$

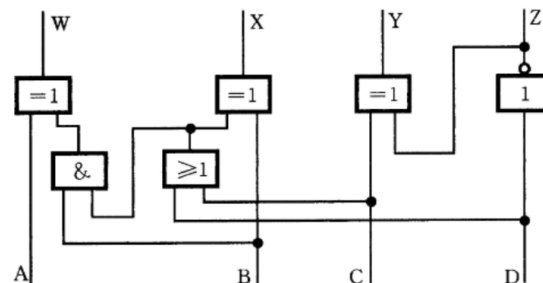


图 4.4 逻辑电路

• 2、化简输出函数表达式：

— 因为W、X、Y、Z已经是最简逻辑式，因此不需要化简。

• 3、列出输出函数真值表（见表4.3）：

— 因为A、B、C、D为8421码，因此输入只允许为0000~1001。

• 4、功能评述

— 根据真值表，可知该电路的输出为1位十进制数的**余3码**（见第1章的表1.3），即该电路是将**8421码**转换为余3码的**代码转换电路**。

表 4.3 真值表

ABCD	WXYZ	ABCD	WXYZ
0000	0011	0101	1000
0001	0100	0110	1001
0010	0101	0111	1010
0011	0110	1000	1011
0100	0111	1001	1100

表 1.3 常用的 3 种 BCD 码

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



# 4.2 组合逻辑电路设计

4.2.1 设计方法概述

4.2.2 设计举例

4.2.3 设计中几个实际问题的处理

## • 4.2.1 设计方法概述

- 根据问题要求完成的逻辑功能，求出在特定条件下实现该功能的逻辑电路，称为**逻辑设计**，也称**逻辑综合**。逻辑设计是逻辑分析的逆过程。
- 组合逻辑电路的**设计过程**如下：
  - 1、建立给定问题的逻辑描述
    - 这是最重要的一步。首先要确定电路的输入和输出，然后建立输出与输入的逻辑关系（逻辑表达式）。建立逻辑表达式的方法有两种：**真值表法**、**分析法**。
  - 2、求出逻辑函数的最简表达式
  - 3、选择逻辑门类型并进行逻辑函数变换
  - 4、画出逻辑电路图

## • 4.2.2 设计举例

— 例4.4 设计一个3变量“多数表决电路”。

— 解：

• 1、建立给定问题的逻辑描述（采用真值表法）

- 输入为A、B、C，输出为F。
- 多数表决电路：当A、B、C中有2个或3个1时，F的值为1；其他情况，F的值为0。
- 因此， $F(A,B,C) = \sum m(3,5,6,7)$ 。

• 2、求出逻辑函数的最简表达式

- 函数F的真值表见表4.4，卡诺图见图4.5(a)。
- 采用卡诺图化简法，得到： $F(A,B,C) = A \cdot B + A \cdot C + B \cdot C$ 。

• 3、选择逻辑门类型并进行逻辑函数变换

- 假设采用“与非门”来实现该电路，需要将F转换为“与非-与非”的表达形式：
- $F(A,B,C) = \neg(\neg(A \cdot B) \cdot \neg(A \cdot C) \cdot \neg(B \cdot C))$ 。

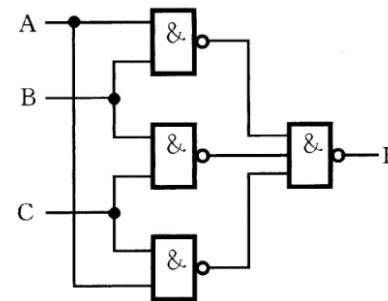
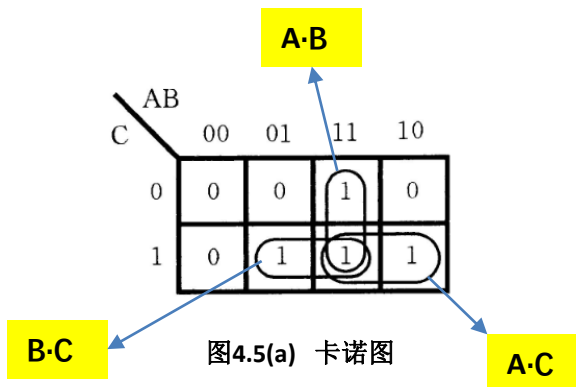
• 4、画出逻辑电路图

- 见图4.5(b)。

序号	ABC	几个1
$m_0$	000	0
$m_1$	001	1
$m_2$	010	1
$m_3$	011	2
$m_4$	100	1
$m_5$	101	2
$m_6$	110	2
$m_7$	111	3

表 4.4 真值表

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



— **例4.5** 设计一个比较两个3位二进制数是否相等的**数值比较器**。

— 解：

• 1、建立给定问题的逻辑描述（采用分析法）

- 输入为 $A=a_3a_2a_1$ 、 $B=b_3b_2b_1$ ，共6位；输出为F，1位二进制数。
- A与B相等，必须是： $a_3=b_3$ ， $a_2=b_2$ ， $a_1=b_1$ 。
- 因此， $F = (a_3 \cdot b_3 + a_3 \cdot /b_3) \cdot (a_2 \cdot b_2 + a_2 \cdot /b_2) \cdot (a_1 \cdot b_1 + a_1 \cdot /b_1)$ 。

• 2、求出逻辑函数的最简表达式

- 函数F已经是最简表达式，不需要再简化。

• 3、选择逻辑门类型并进行逻辑函数变换

- 假设采用“**异或门**”和“**或非门**”来实现该电路，将F转换为如下的形式：
- $F = /(a_3 \oplus b_3) \cdot /(a_2 \oplus b_2) \cdot /(a_1 \oplus b_1) = /((a_3 \oplus b_3) + (a_2 \oplus b_2) + (a_1 \oplus b_1))$ 。

• 4、画出逻辑电路图

- 见图4.6。

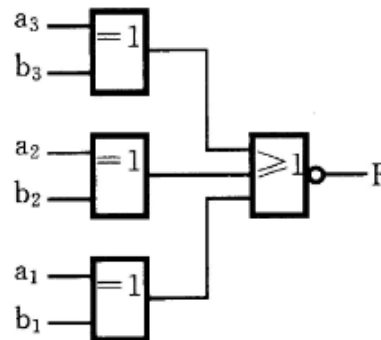


图 4.6 逻辑电路

– 例4.6 设计一个乘法器，用于产生两个2位二进制数相乘的积。

– 解：

• 方法一：采用真值表法

– (1) 建立给定问题的逻辑描述

» 2位二进制数的最大值为3，相乘后，最大值为9。

» 因此，输入为 $A=A_1A_0$ 、 $B=B_1B_0$ ，共4位；输出为 $M=M_3M_2M_1M_0$ ，共4位。

» 列出输出与输入的真值表，见表4.5。

» 根据真值表，可以得到：

- $M_3 = \sum m(15)$
- $M_2 = \sum m(10, 11, 14)$
- $M_1 = \sum m(6, 7, 9, 11, 13, 14)$
- $M_0 = \sum m(5, 7, 13, 15)$

表 4.5 真值表

$A_1$	$A_0$	$B_1$	$B_0$	$M_3$	$M_2$	$M_1$	$M_0$	$A_1$	$A_0$	$B_1$	$B_0$	$M_3$	$M_2$	$M_1$	$M_0$
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0
0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1	1	1	0	1	0	0	1	1
0	1	1	0	0	0	1	0	1	1	1	0	0	1	1	0
0	1	1	1	0	0	1	1	1	1	1	1	1	0	0	1

– (2) 求出逻辑函数的最简表达式

» 采用卡诺图化简法，得到输出函数的最简“与-或”表达式：

- $M_3 = A_1 \cdot A_0 \cdot B_1 \cdot B_0$
- $M_2 = A_1 \cdot A_0 \cdot B_1 + A_1 \cdot B_1 \cdot B_0$
- $M_1 = A_1 \cdot B_1 \cdot B_0 + A_1 \cdot A_0 \cdot B_0 + A_1 \cdot A_0 \cdot B_1 + A_0 \cdot B_1 \cdot B_0$
- $M_0 = A_0 \cdot B_0$

– (3) 选择逻辑门类型并画出逻辑电路图

» 假设采用“与门”和“或门”来实现该电路，具体见图4.7。

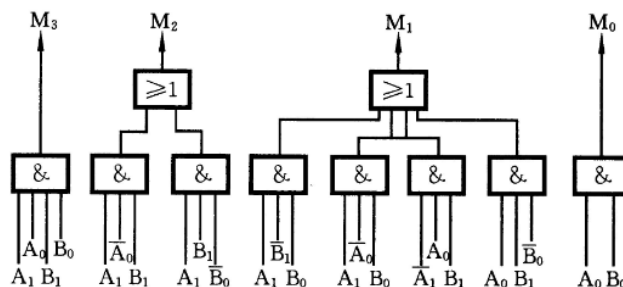


图 4.7 方法 I 的逻辑电路

- 方法二：采用**分析法**

- (1) 建立给定问题的逻辑描述

- 2位二进制数乘法运算方法**如右所示**，其中：

- $C_1 = A_1 \cdot B_0 \cdot A_0 \cdot B_1$
      - $C_2 = C_1 \cdot A_1 \cdot B_1 = A_1 \cdot B_0 \cdot A_0 \cdot B_1 \cdot A_1 \cdot B_1 = A_1 \cdot B_0 \cdot A_0 \cdot B_1$
      - $M_0 = A_0 \cdot B_0$
      - $M_1 = A_1 \cdot B_0 \oplus A_0 \cdot B_1$
      - $M_2 = C_1 \oplus A_1 \cdot B_1 = A_1 \cdot B_0 \cdot A_0 \cdot B_1 \oplus A_1 \cdot B_1$
      - $M_3 = C_2 = A_1 \cdot B_0 \cdot A_0 \cdot B_1$

$$\begin{array}{r}
 \begin{array}{cc} & A_1 & A_0 \\ \times (\text{乘}) & B_1 & B_0 \\ \hline & C_2 & C_1 & A_1 \times B_0 & A_0 \times B_0 \\ + (\text{加}) & & A_1 \times B_1 & A_0 \times B_1 \\ \hline M_3 & M_2 & M_1 & M_0
 \end{array}
 \end{array}$$

- (2) 选择逻辑门类型并画出逻辑电路图

- 假设采用“**异或门**”和“**与门**”来实现该电路，具体见图4.8。

- 比较图4.7和图4.8，图4.7使用了10个门电路，图4.8只使用了7个门电路，因此图4.8的电路更简单。

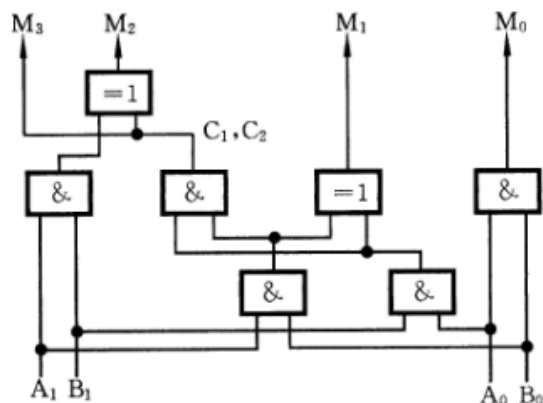


图 4.8 方法 II 的逻辑电路

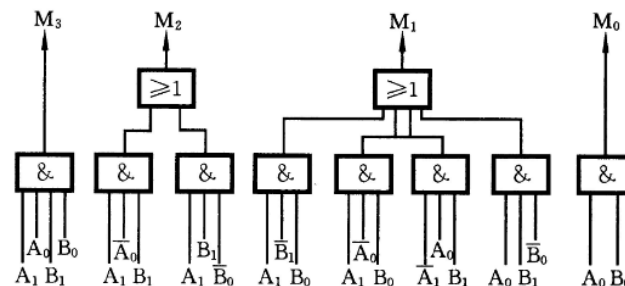


图 4.7 方法 I 的逻辑电路

## • 4.2.3 设计中几个实际问题的处理

### – 1、包含无关条件的组合逻辑电路设计

- 前面3个例子（例4.4、例4.5、例4.6）的组合逻辑电路设计中，输入变量的任何一种组合，都有确定的输出值。
- 即对于一个具有 $n$ 个输入变量的组合逻辑电路，假设输入变量的 $m$ 种取值组合对应的输出值为1，则 $2^n - m$ 种取值组合对应的输出值为0。因此，输出可以用 $m$ 个最小项之和表示。
- 但是实际问题中，有一些输入变量的取值组合是不会出现的；或者有一些输入变量的取值组合，其对应的输出值并不关心。
- 通常将这类问题称为包含无关条件的逻辑问题，与这些输入取值组合对应的最小项称为无关最小项（无关项、任意项），描述这类问题的逻辑函数称为包含无关条件的逻辑函数。
- 例4.7 设计一个组合逻辑电路，用于判别以余3码表示的1位十进制数是否为合数（0~9中，4、6、8、9为合数；与合数对应的是质数：2、3、5、7）。

• 解：

- 输入为A、B、C、D，4位二进制数；输出为F，1位二进制数。输入与输出的关系见表4.6（真值表）：
  - 当输入对应的余3码其对应的十进制数为4、6、8、9时，输出F=1。
  - 当输入对应的余3码其对应的十进制数为0、1、2、3、5、7时，输出F=0。
  - 当输入为不允许出现的值时（0000、0001、0010、1101、1110、1111，即表1.3最右列中没有出现的6种情况），输出F=d（d表示可以是0，也可以是1）。
- 因此有： $F = \sum m(7,9,11,12) + \sum d(0,1,2,13,14,15)$ 。函数F对应的卡诺图如图4.9(a)所示。如果只对卡诺图中的“1小方块”画卡诺圈（图4.9(a)），则得到F的简化函数： $F = A \cdot B \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D$ 。
- 如果将卡诺图中的“d小方块”也加入卡诺圈（图4.9(b)），则得到F的简化函数： $F = A \cdot B + A \cdot D + B \cdot C \cdot D$ 。显然，这个式子更简单。
- 采用“与非门”实现F，需要先将F转换为“与非-与非”表达式： $F = \overline{\overline{A \cdot B + A \cdot D + B \cdot C \cdot D}} = \overline{\overline{A \cdot B} \cdot \overline{A \cdot D} \cdot \overline{B \cdot C \cdot D}}$ 。对应的逻辑电路如图4.10所示。

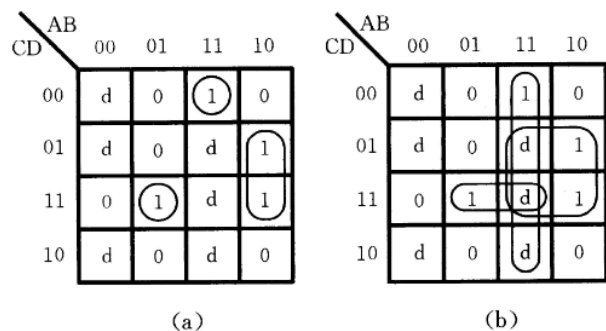


图 4.9 卡诺图

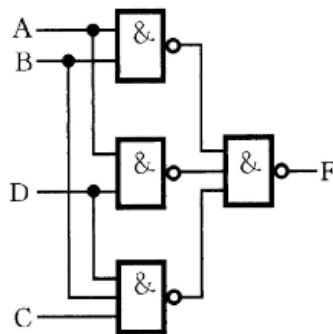


图 4.10 逻辑电路

表 4.6 真值表

A	B	C	D	F	A	B	C	D	F
0	0	0	0	d	1	0	0	0	0
0	0	0	1	d	1	0	0	1	1
0	0	1	0	d	1	0	1	0	0
0	0	1	1	0	1	0	1	1	1
0	1	0	0	0	1	1	0	0	1
0	1	0	1	0	1	1	0	1	d
0	1	1	0	0	1	1	1	0	d
0	1	1	1	1	1	1	1	1	d

表 1.3 常用的 3 种 BCD 码

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

## – 2、多输出函数的组合逻辑电路设计

- 设计**多输出函数**（多个输出）的组合逻辑电路时，因为各输出函数之间往往存在相互联系，因此应该将它们作为一个整体考虑，而不应该将其截然分开。
- 关键是在函数化简时，要**找出各输出函数的共用项**，以便在逻辑电路中实现**对逻辑门的共享**，从而使电路**整体结构最简**。
- 例4.8** 假定某组合逻辑电路结构框图如图4.11所示，试用最少的“与非门”实现该电路的功能。
- 解：
  - 首先画出 $F_1$ 和 $F_2$ 的卡诺图，如图4.12(a)和图4.12(b)所示。如果分别对 $F_1$ 和 $F_2$ 进行简化，并采用“与非门”实现，则有（图4.12(c)）：

$$\gg F_1 = \overline{A} \cdot C + B \cdot C = \overline{((\overline{A} \cdot C + B \cdot C))} = \overline{((\overline{A} \cdot C) \cdot (B \cdot C))}$$

$$\gg F_2 = A \cdot B + B \cdot \overline{C} = \overline{((A \cdot B + B \cdot \overline{C}))} = \overline{((A \cdot B) \cdot (B \cdot \overline{C}))}$$

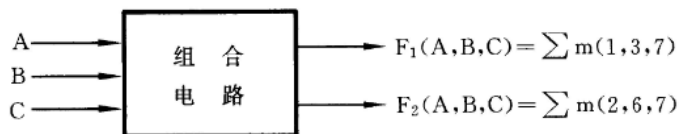
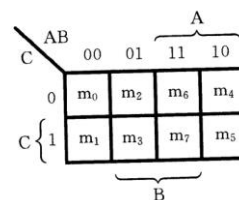
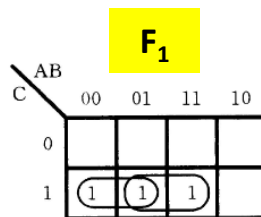
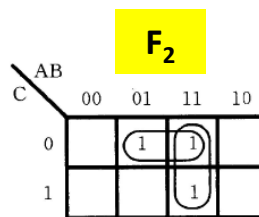


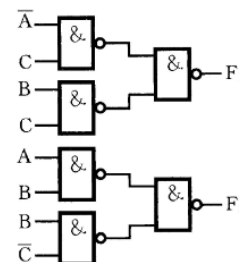
图 4.11 组合逻辑电路结构框图



(a)



(b)



(c)

图 4.12 卡诺图及逻辑电路之一

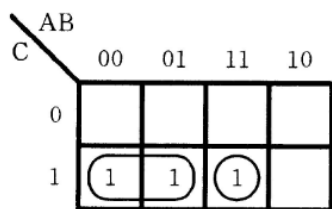


- 考虑到两个输出函数应充分“共享”，则可以采用图4.13(a)和图4.13(b)的卡诺图化简方式，并采用“与非门”实现，则有（图4.13(c)）：

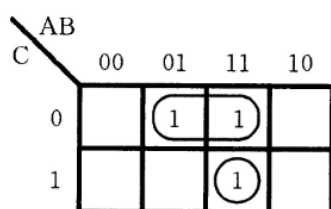
- $$F_1 = \bar{A} \cdot C + A \cdot B \cdot C = \overline{\overline{(\bar{A} \cdot C + A \cdot B \cdot C)}} = \overline{\overline{(\bar{A} \cdot C)} \cdot \overline{(A \cdot B \cdot C)}}$$

- $$F_2 = A \cdot B \cdot C + B \cdot \bar{C} = \overline{\overline{(A \cdot B \cdot C + B \cdot \bar{C})}} = \overline{\overline{(A \cdot B \cdot C)} \cdot \overline{(B \cdot \bar{C})}}$$

- 可见，图4.13(c)的电路（5个与非门）要比图4.12(c)的电路（6个与非门）简单一些。

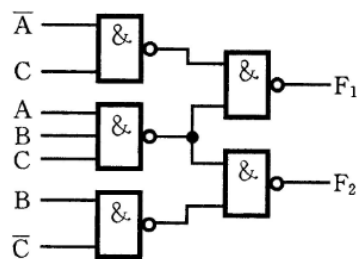


(a)



(b)

图 4.13 卡诺图及逻辑电路之二



(c)

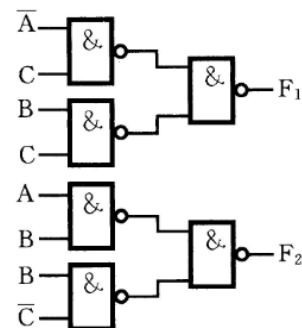


图4.12(c) 逻辑电路

• **例4.9** 设计一个全加器。

• 解：

- 全加器（FA: Full Adder）的输入有3个： $A_i$ 、 $B_i$ 、 $C_{i-1}$ ；输出有2个： $S_i$ 、 $C_i$ 。其中， $C_{i-1}$ 为来自低位的进位， $C_i$ 为向高位的进位。全加器的真值表见表4.7。
- 根据真值表，可以得到全加器的最小项表达式，以及卡诺图（图4.14）：
  - $S_i = \sum m(1,2,4,7)$
  - $C_i = \sum m(3,5,6,7)$
- 根据卡诺图，可以得到简化后的逻辑表达式：
  - $S_i = /A_i \cdot /B_i \cdot C_{i-1} + /A_i \cdot B_i \cdot /C_{i-1} + A_i \cdot /B_i \cdot /C_{i-1} + A_i \cdot B_i \cdot C_{i-1}$
  - $C_i = A_i \cdot B_i + A_i \cdot C_{i-1} + B_i \cdot C_{i-1}$
- 如果采用“异或门”和“与非门”实现，需要将逻辑表达式进行变换，得到图4.15(a)所示的逻辑电路：
  - $S_i = /A_i \cdot /B_i \cdot C_{i-1} + /A_i \cdot B_i \cdot /C_{i-1} + A_i \cdot /B_i \cdot /C_{i-1} + A_i \cdot B_i \cdot C_{i-1} = A_i \oplus B_i \oplus C_{i-1}$
  - $C_i = A_i \cdot B_i + A_i \cdot C_{i-1} + B_i \cdot C_{i-1} = /(/(A_i \cdot B_i) \cdot /(A_i \cdot C_{i-1}) \cdot /(B_i \cdot C_{i-1}))$

表 4.7 全加器真值表

$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

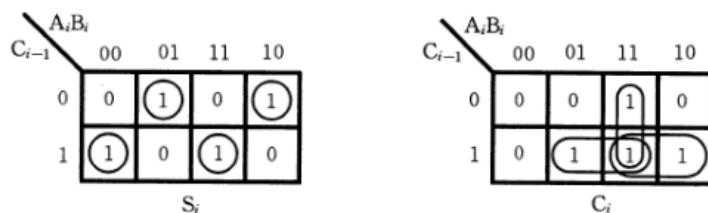


图 4.14 函数  $S_i$  和  $C_i$  的卡诺图

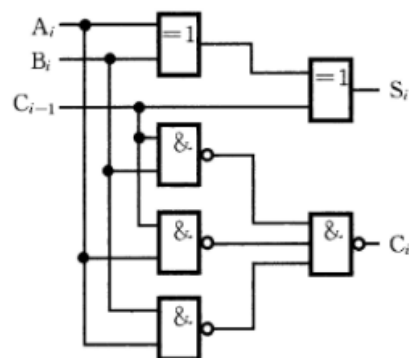


图4.15(a) 逻辑电路

- 图4.15(a)虽然对单个函数是最简的，但是在整体上不是最简的。
- 可以对 $C_i$ 进行如下的变换，得到图4.15(b)所示的逻辑电路：
  - $C_i = /A_i \cdot /B_i \cdot C_{i-1} + /A_i \cdot B_i \cdot /C_{i-1} + A_i \cdot /B_i \cdot /C_{i-1} + A_i \cdot B_i \cdot C_{i-1} = /(A_i \oplus B_i) \cdot C_{i-1} \cdot /(A_i \cdot B_i)$
- 显然，图4.15(b)（3个与非门+2个异或门）比图4.15(a)（4个与非门+2个异或门）要简单一些。
- 并且实现图4.15(a)需要三种芯片（2输入异或门、2输入与非门、3输入与非门），实现图4.15(b)只需要两种芯片（2输入异或门、2输入与非门）。
- 图4.16给出了采用7486（4个2输入异或门）和7400（4个2输入与非门）实现图4.15(b)全加器的引脚连接图。

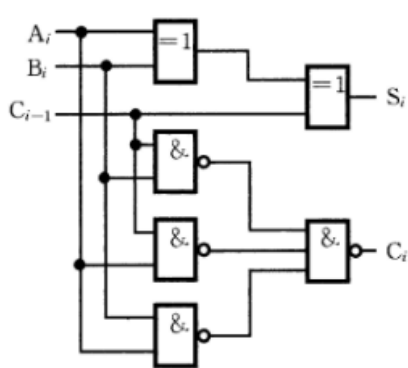


图4.15(a) 逻辑电路

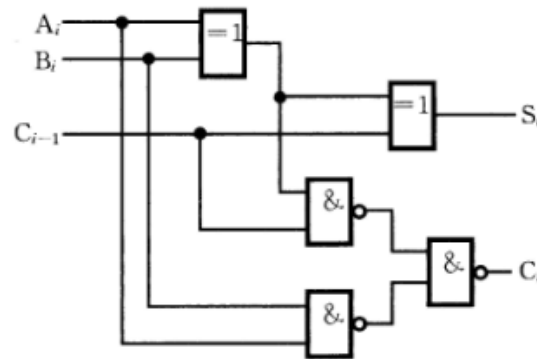


图4.15(b) 逻辑电路

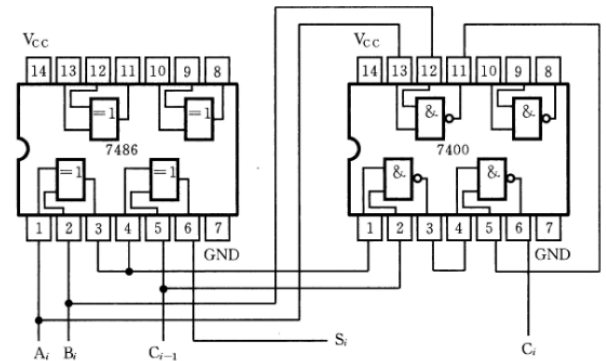


图 4.16 全加器的芯片引脚连接图

### – 3、无反变量提供的组合逻辑电路设计

- 有一些输出函数的逻辑表达式中含有**输入变量的反变量**，如果采用**非门**来实现反变量，则会增加电路的复杂性。

- 可以通过逻辑公式变换的方式，在**不需要非门**的情况下，实现这种逻辑电路。

- **例4.10** 输入变量中无反变量时，用“与非门”实现逻辑函数： $F = \neg A \cdot B + B \cdot \neg C + A \cdot \neg B \cdot C + A \cdot C \cdot \neg D$ 。

- 解：

- 函数F已经是最简表达式，当采用“与非门”和“非门”实现时，直接按照函数F的表达式画出的电路如图4.17(a)所示。

- 如果将函数F进行如下的变换，可以不需要“非门”，电路见图4.17(b)：

$$\gg F = \neg A \cdot B + B \cdot \neg C + A \cdot \neg B \cdot C + A \cdot C \cdot \neg D = B \cdot \neg(A \cdot C) + A \cdot C \cdot \neg(B \cdot D) = \neg(\neg(B \cdot \neg(A \cdot C)) + \neg(A \cdot C \cdot \neg(B \cdot D)))$$

- 可见，图4.17(b)的电路（5个与非门）要比图4.17(a)的电路（5个与非门+4个非门）简单很多。

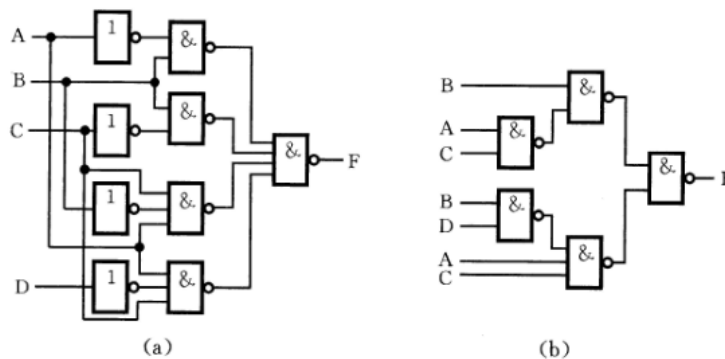


图 4.17 逻辑电路

- **例4.11** 在输入无反变量提供时，用最少的逻辑门实现逻辑函数： $F = \neg A \cdot B + B \cdot \neg C + A \cdot \neg B \cdot C$ 。

• 解：

- 函数F已经是最简表达式，当采用“与非门”和“非门”实现时，直接按照函数F的表达式画出的电路如图4.18(a)所示。
- 如果将函数F进行如下的变换，可以不需要“非门”，并且使电路简化，见图4.18(b):
  - $F = \neg A \cdot B + B \cdot \neg C + A \cdot \neg B \cdot C = (A \cdot C) \oplus B$
- 可见，图4.18(b)的电路（1个与门+1个异或门）要比图4.18(a)的电路（4个与非门+3个非门）简单很多。

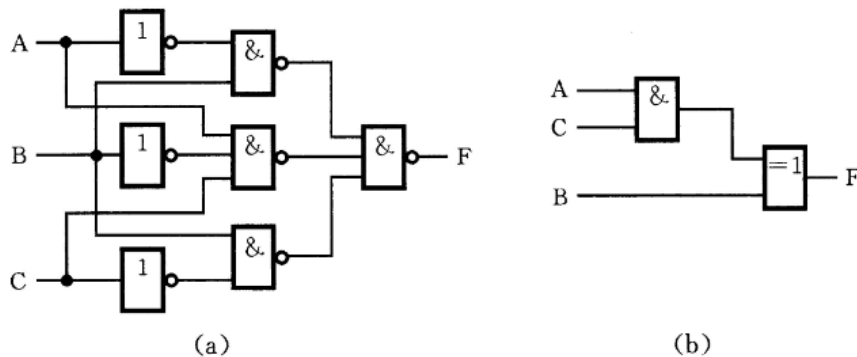


图 4.18 逻辑电路

- **例4.12** 设计一个组合逻辑电路，用来判断献血者与受血者**血型是否相容**。血型相容规则如表4.8所示。

解：

- 输入变量：献血者的血型（4种情况，用2位二进制数表示，WX）、受血者的血型（4种情况，用2位二进制数表示，YZ），血型编码如表4.9所示。输出变量：血型是否相容（1位二进制数，F）。

- 根据表4.8和表4.9，可以得到F的逻辑公式：

$$F = \neg W \cdot \neg X \cdot (\neg Y \cdot \neg Z + Y \cdot \neg Z) + \neg W \cdot X \cdot (\neg Y \cdot \neg Z + Y \cdot \neg Z) + W \cdot \neg X \cdot Y \cdot \neg Z + W \cdot X \cdot (\neg Y \cdot \neg Z + \neg Y \cdot Z + Y \cdot \neg Z + Y \cdot Z) \\ = \neg W \cdot \neg X \cdot \neg Z + Y \cdot \neg Z + W \cdot X + X \cdot Y \cdot Z$$

- 如果直接用“与非门”和“非门”实现上述逻辑公式，见图4.19(a)。

- 如果改变血型的编码，见表4.10，则可以得到下面的逻辑公式：

$$F = \neg W \cdot \neg X \cdot (\neg Y \cdot \neg Z + \neg Y \cdot Z + Y \cdot \neg Z + Y \cdot Z) + \neg W \cdot X \cdot (\neg Y \cdot \neg Z + Y \cdot \neg Z) + W \cdot \neg X \cdot (Y \cdot \neg Z + Y \cdot Z) + W \cdot X \cdot Y \cdot Z = (\neg W + Y) \cdot (\neg X + Z)$$

- 如果用“或非门”和“非门”实现上述逻辑公式，见图4.19(b)。

- 可以，图4.19(b)的电路（3个或非门+2个非门）要比图4.19(a)的电路（5个与非门+4个非门）简单很多。

表 4.8 血型相容规则表

受血 献血	A	B	AB	O
A	✓		✓	
B		✓	✓	
AB			✓	
O	✓	✓	✓	✓

表 4.9 血型编码(1)

血 型	献 W X	受 Y Z
A	0 0	0 0
B	0 1	0 1
AB	1 0	1 0
O	1 1	1 1

表 4.10 血型编码(2)

血 型	献 W X	受 Y Z
O	0 0	0 0
A	0 1	0 1
B	1 0	1 0
AB	1 1	1 1

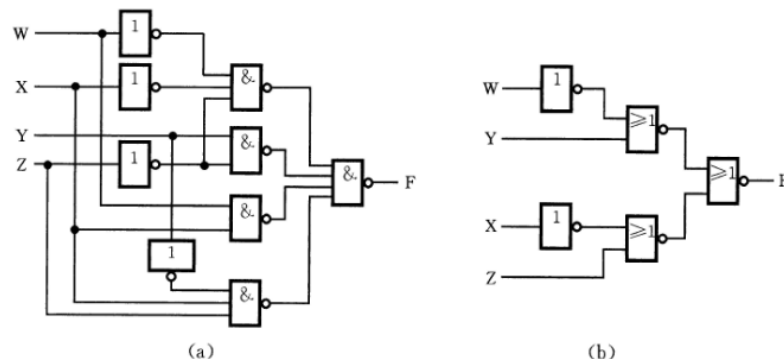


图 4.19 逻辑电路

# 4.3 组合逻辑电路的险象

4.3.1 险象的产生

4.3.2 险象的判断

4.3.3 险象的消除

## • 4.3.1 险象的产生

- 在实际逻辑电路中，信号经过同一电路中的不同路径所产生的时延通常是不一样的，即输入信号经过不同路径到达输出端的时间有先后，这种现象称为**竞争现象**。
- 由于电路中存在竞争现象，使得输入信号的变化可能引起输出信号出现非预期的错误输出，这种现象称为**险象**。
- 通常将不产生错误输出的竞争称为**非临界竞争**，而导致错误输出的竞争称为**临界竞争**。

- 例如，图4.20(a)的逻辑公式为： $F = \neg(\neg(A \cdot B) \cdot \neg(A \cdot C)) = A \cdot B + A \cdot C$ 。书上有误：图4.20(a)中的“与门”应该是“与非门”。
- 图4.20(a)中，当 $B=C=1$ 时， $d=\neg A$ ， $e=\neg(A \cdot B)=\neg(A \cdot 1)=\neg A$ ， $g=\neg(A \cdot C)=\neg A$ ， $F = A + \neg A = 1$ ；即此时无论A是0还是1，输出F都应该保持1不变。
- 但是，由于门电路有延时（假设延时为 $t_{pd}$ ），该电路的输出波形如图4.20(b)所示，产生了“0”型险象。

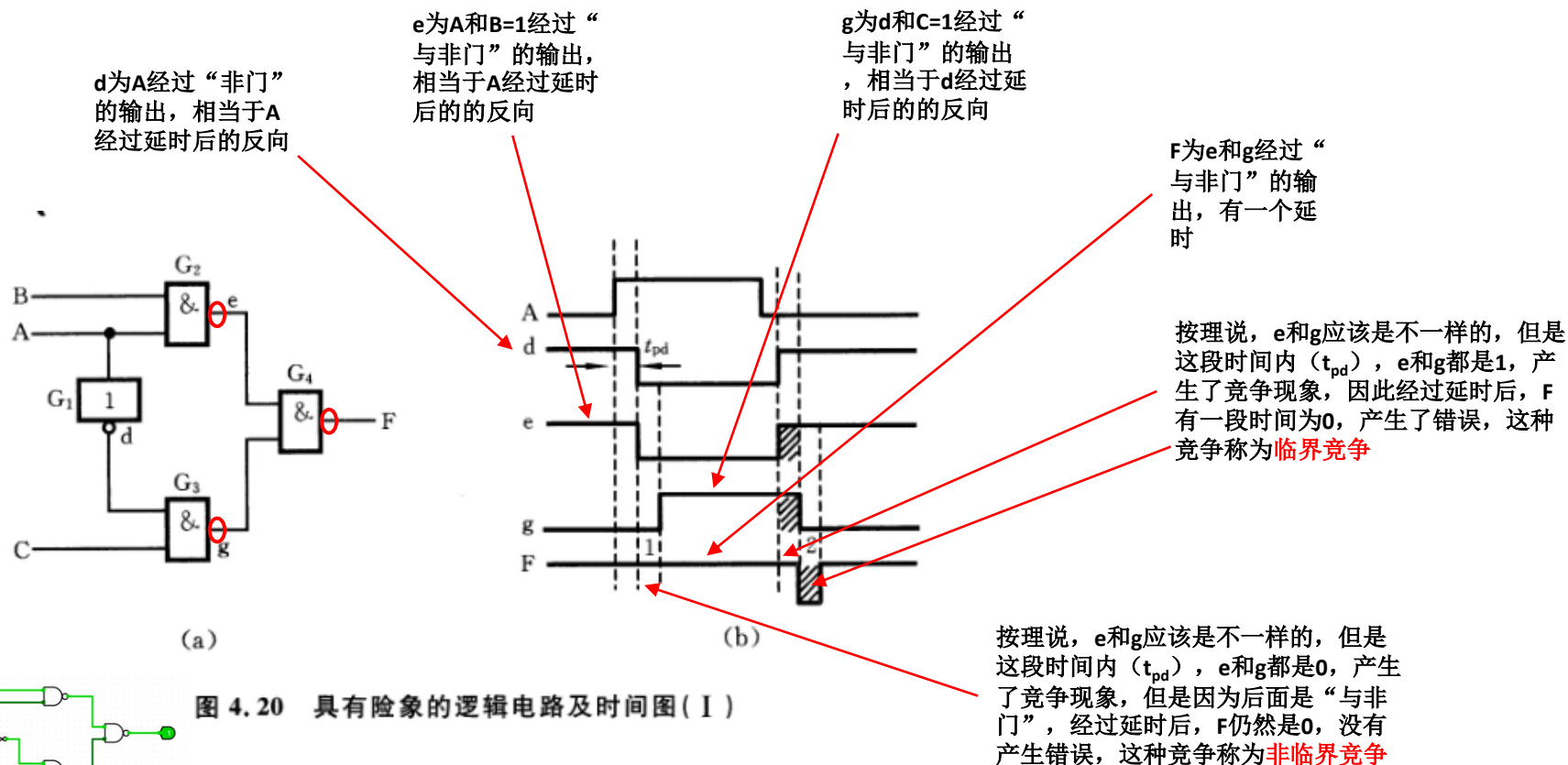


图 4.20 具有险象的逻辑电路及时间图 ( I )



- 再例如，图4.21(a)的逻辑公式为： $F = \neg(\neg(A+B) + \neg(A+C)) = (A+B) \cdot \neg(A+C)$ 。书上有误：图4.21(a)中的“或门”和“与门”都应该是“或非门”。
- 图4.21(a)中，当 $B=C=0$ 时， $d=\neg A$ ， $e=A+B=A+0=A$ ， $g=\neg(A+C)=\neg(A+0)=\neg A$ 。 $F = A \cdot \neg A = 0$ ；即此时无论A是0还是1，输出F都应该保持0不变。
- 但是，由于门电路有延时（假设延时为 $t_{pd}$ ），该电路的输出波形如图4.21(b)所示，产生了“1”型险象。

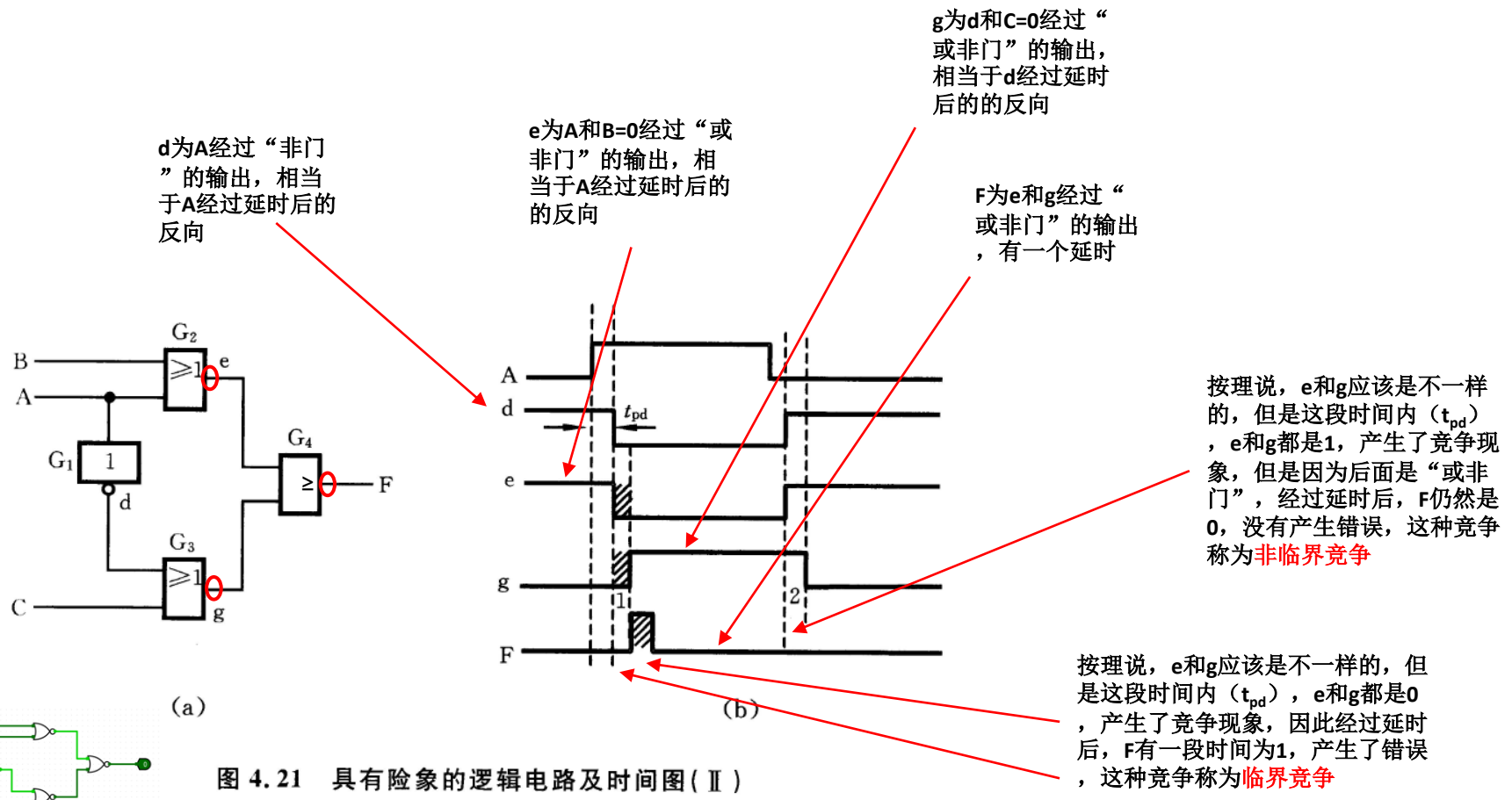


图 4.21 具有险象的逻辑电路及时间图 ( II )

## • 4.3.2 险象的判断

– 险象的判断方法有两种：代数法、卡诺图法。

### – 1、代数法

• 如果逻辑函数表达式中有某个变量的原变量（ $x$ ）和反变量（ $/x$ ），则通过消去逻辑函数表达式中的其他变量，再看逻辑函数表达式是否会变为“ $x+/x$ ”或者“ $x\cdot/x$ ”的形式，若是，则说明该逻辑函数电路可能会产生险象。

• **例4.13** 已知描述某组合逻辑电路的逻辑函数为 $F=/A\cdot/C+/A\cdot B+A\cdot C$ ，判断该逻辑电路是否可能产生险象。

• 解：

– 因为逻辑函数中有 $A$ 和 $/A$ 、 $C$ 和 $/C$ ，因此，变量 $A$ 、 $C$ 均具备产生险象的条件。

– 先考察 $A$ ，可知， $A$ 的变化可能使电路产生险象：

- »  $BC=00$      $F=/A$
- »  $BC=01$      $F=A$
- »  $BC=10$      $F=/A$
- »  $BC=11$      $F=/A+A$

– 再考察 $C$ ，可知， $C$ 的变化不会导致电路产生险象：

- »  $AB=00$      $F=/C$
- »  $AB=01$      $F=/C+1$
- »  $AB=10$      $F=C$
- »  $AB=11$      $F=C$

- **例4.14** 试判断逻辑函数 $F=(A+B) \cdot (\neg A+C) \cdot (\neg B+C)$ 描述的电路是否可能产生险象。
- 解：
  - 因为逻辑函数中有 $A$ 和 $\neg A$ 、 $B$ 和 $\neg B$ ，因此，变量 $A$ 、 $B$ 均具备产生险象的条件。
  - 先考察 $A$ ，可知， $A$ 的变化可能使电路产生险象：
    - $BC=00$   $F=A \cdot \neg A$
    - $BC=01$   $F=A$
    - $BC=10$   $F=0$
    - $BC=11$   $F=1$
  - 再考察 $B$ ，可知， $B$ 的变化也可能使电路产生险象：
    - $AC=00$   $F=B \cdot \neg B$
    - $AC=01$   $F=B$
    - $AC=10$   $F=0$
    - $AC=11$   $F=1$

## – 2、卡诺图法

- 当逻辑函数采用“与-或”表达式时，如果卡诺图中的两个卡诺圈存在“**相切**”关系，则该电路可能产生险象。

- 例4.15** 已知描述某组合逻辑电路的逻辑函数为 $F = /A \cdot D + /A \cdot C + A \cdot B \cdot /C$ ，判断该逻辑电路是否可能产生险象。

- 解：

– 首先画出函数 $F$ 的卡诺图，并画出卡诺圈，如图4.22所示。

– 图中红色的卡诺圈和蓝色的卡诺圈“**相切**”，因此，该电路可能产生险象。验证：当 $B=D=1$ 、 $C=0$ 时， $F=A+/A$ 。

– 红色的卡诺圈和绿色的卡诺圈“相交”，蓝色的卡诺圈和绿色的卡诺圈既不“相交”也不“相切”，这两种情况都不会产生险象。

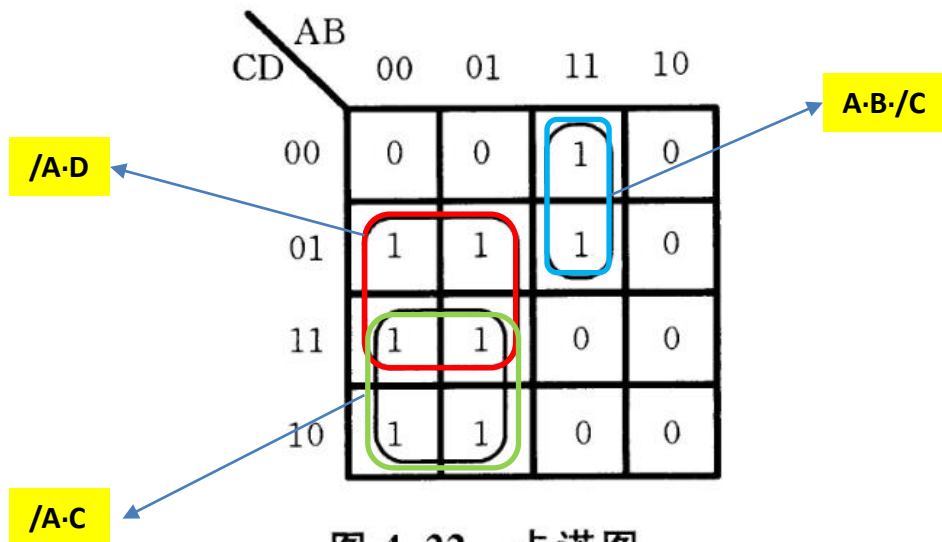
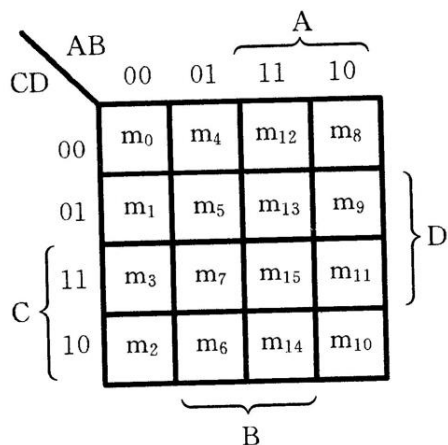


图 4.22 卡诺图

## • 4.3.3 险象的消除

– 消除险象的方法有3种：增加冗余项、增加惯性延迟环节、选通法。

### – 1、用增加冗余项的方法消除险象

- 增加冗余项的方法：在逻辑函数表达式中，“或”上多余的与项，或者“与”上多余的或项，使逻辑函数不可能在某种条件下变成“ $X+X$ ”或“ $X \cdot X$ ”的形式，从而消除可能产生的险象。

- 例4.16 用增加冗余项的方法消除图4.20(a)所示电路中可能产生的险象。

- 解：

- 图4.20(a)对应的逻辑公式为： $F = \overline{(\overline{A \cdot B}) \cdot \overline{(A \cdot C)}} = A \cdot B + \overline{A} \cdot C$ 。当 $B=C=1$ 时， $F = A + \overline{A} = 1$ ，产生了“0”型险象。

- 因为， $B=C=1$ 时， $F = A + \overline{A} = 1$ ；可知 $B \cdot C$ 为逻辑公式 $F$ 的冗余项。即： $F = A \cdot B + \overline{A} \cdot C + B \cdot C$ 。增加冗余项后的电路如图4.23所示，该电路不再产生险象。

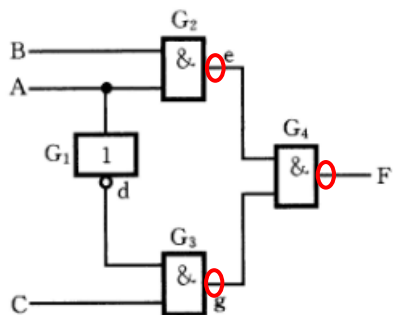


图4.20(a) 具有险象的逻辑电路

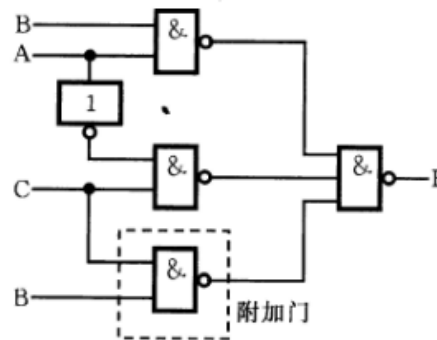
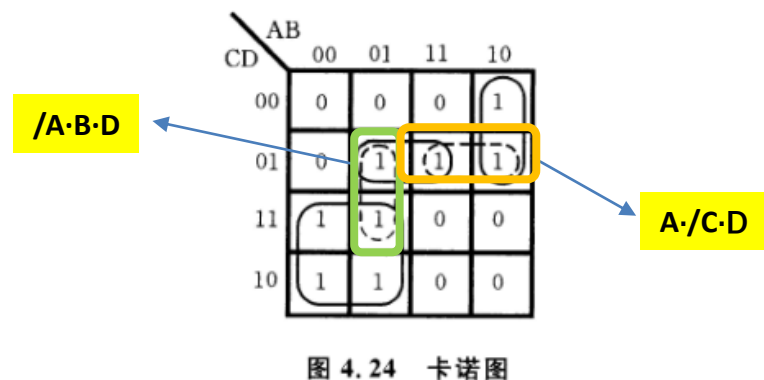
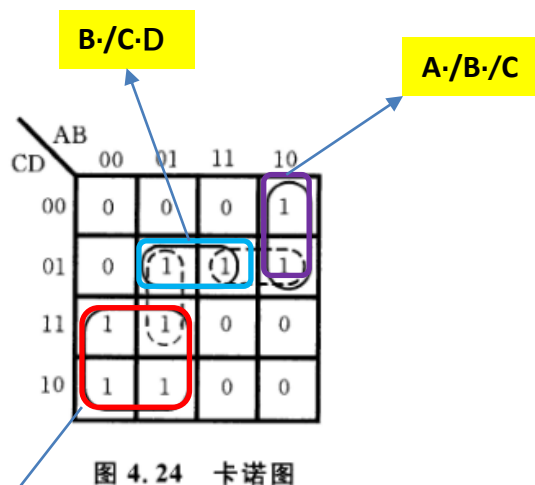


图 4.23 增加冗余项后的逻辑电路

- **例4.17** 已知描述某组合逻辑电路的函数表达式为 $F = \bar{A} \cdot C + B \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot \bar{C}$ ，用增加冗余项的方法消除该电路中可能产生的险象。

• 解：

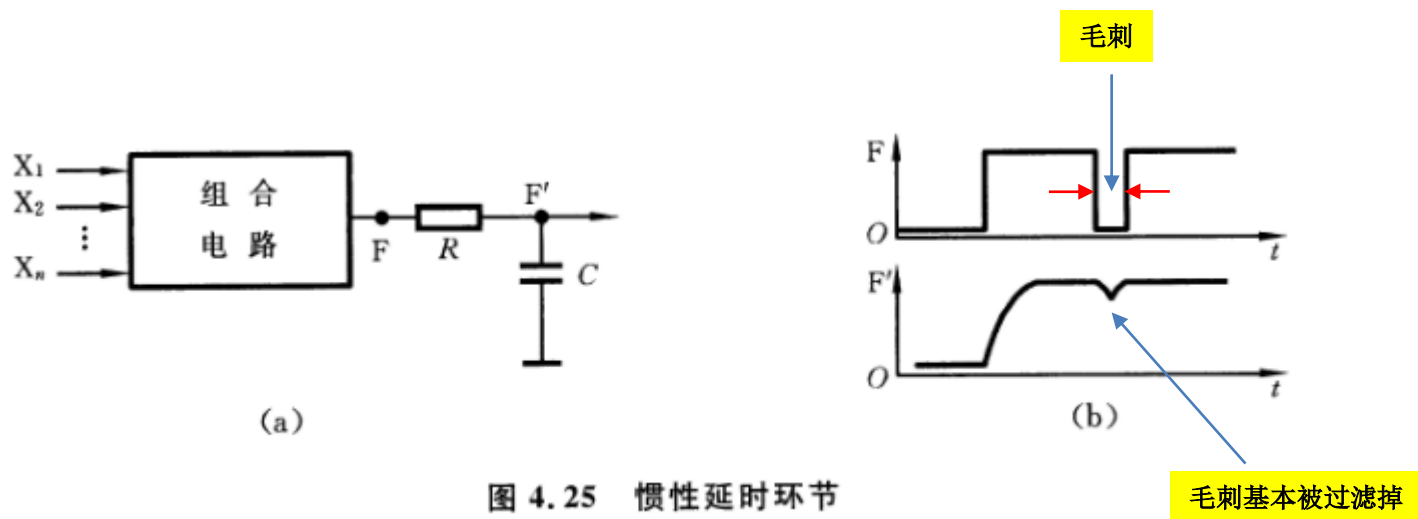
- 首先画出函数的卡诺图和卡诺圈，如图4.24所示。可见，红色卡诺圈与蓝色卡诺圈“相切”，蓝色卡诺圈与紫色卡诺圈“相切”。因此，该电路存在险象。
- 可以采用增加两个（绿色、黄色）冗余的卡诺圈（**冗余项**）来消除险象（消除卡诺圈“相切”的现象），即增加 $\bar{A} \cdot B \cdot D$ 和 $A \cdot \bar{C} \cdot D$ 冗余项。
- 增加冗余项后的函数表达式为： $F = \bar{A} \cdot C + B \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot D + A \cdot \bar{C} \cdot D$ 。



$\bar{A} \cdot C$

## — 2、增加惯性延时环节

- 通常采用**RC**（Resistor-Capacitance，电阻-电容）**电路**作为惯性延时环节，如图4.25(a)所示。
- 图4.25(b)为输出**毛刺**（“0”型险象）经过RC电路（惯性延时环节）后的效果。可见，输出毛刺基本被过滤掉，从而消除了险象。
- RC电路的**时间常数**（ $\tau=RC$ ）应适度选择。一般要求 $\tau$ 大于毛刺（尖脉冲）的宽度。当然 $\tau$ 也不能太大，否则会使正常的输出信号产生不允许的畸变。



### – 3、选通法

- 增加冗余项的方法和增加惯性延时环节的方法，都需要**增加电路的器件**。
- 选通法利用选通脉冲的作用，从时间上加以控制，以避免险象脉冲。选通法**不需要增加任何器件**。
- 例如，对于图4.20(a)具有险象的电路， $B=C=1$ 时，将产生险象（“0”型险象）。
- 为了避开险象，可以采用**选通脉冲**对该电路的输出门加以控制，见图4.26。
- 当发生险象（尖脉冲）时，选通脉冲为低电平，从而使尖脉冲无法通过 $G_4$ 与非门；没有险象时，选通脉冲为高电平，电路送出稳定输出信号。
- **选通法**是在时间上让信号有选择地通过电路。

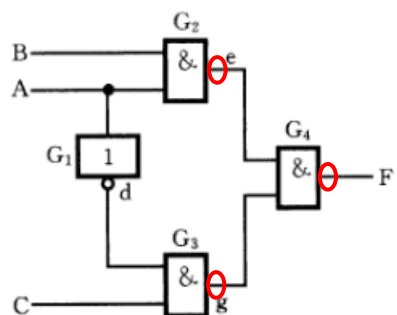


图4.20(a) 具有险象的逻辑电路

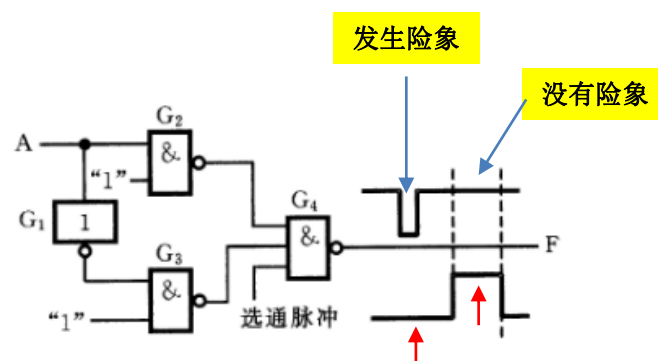


图 4.26 用选通法避开险象原理图



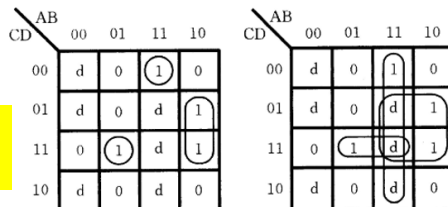
# 本章小结

- **组合逻辑电路**是指电路在任何时刻产生的稳定输出值，仅仅取决于该时刻各输入值的组合，而与过去的输入值无关。
- 组合逻辑电路的**特点**:
  - ① 由逻辑门电路组成，不包含任何记忆元件；
  - ② 信号是单向传输的，不存在任何反馈回路。
- 组合逻辑电路**分析**的一般**步骤**:
  - ① 根据逻辑电路图写出输出函数表达式；
  - ② 化简输出函数表达式（见第2章的“2.4 逻辑函数化简”）；
  - ③ 列出输出函数真值表；
  - ④ 功能评述。
- 组合逻辑电路**设计**（逻辑**综合**）的一般**步骤**:
  - ① 建立给定问题的逻辑描述（建立逻辑表达式的方法有两种：**真值表法**、**分析法**）；
  - ② 求出逻辑函数的最简表达式；
  - ③ 选择逻辑门类型并进行逻辑函数变换；
  - ④ 画出逻辑电路图。

## • 设计中几个实际问题的处理:

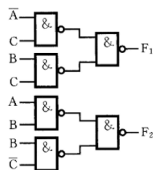
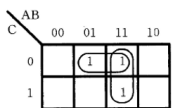
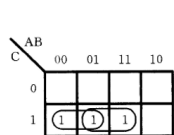
- 1、**包含无关条件**的组合逻辑电路设计: 实际问题中, 有一些输入变量的取值组合是不会出现的; 或者有一些输入变量的取值组合, 其对应的输出值并不关心。通常将这类问题称为包含无关条件的逻辑问题, 与这些输入取值组合对应的最小项称为无关最小项 (**无关项**、任意项, d), 描述这类问题的逻辑函数称为包含无关条件的逻辑函数。

没有利用无关项:  $F = A \cdot B \cdot D + A \cdot B \cdot C / D + / A \cdot B \cdot C \cdot D$

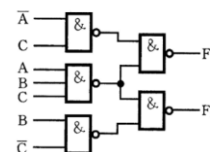
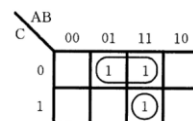
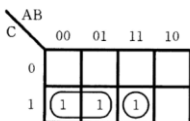


利用无关项: 在卡诺图法化简时, 将无关项画到卡诺圈中, 这样可以使逻辑公式更简单;  $F = A \cdot B + A \cdot D + B \cdot C \cdot D$

- 2、**多输出函数**的组合逻辑电路设计: 设计多输出函数 (多个输出) 的组合逻辑电路时, 因为各输出函数之间往往存在相互联系, 因此应该将它们作为一个整体考虑, 而不应该将其截然分开。关键是在函数化简时, 要**找出各输出函数的共用项**, 以便在逻辑电路中实现对逻辑门的共享, 从而使电路整体结构最简。



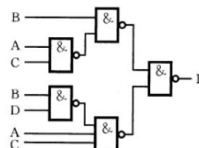
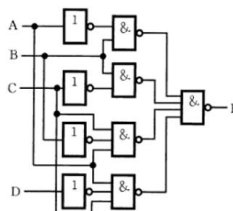
分开设计



整体设计: 找出共用项, 这样使电路更简单

- 3、**无反变量提供**的组合逻辑电路设计: 有一些输出函数的逻辑表达式中含有输入变量的反变量, 如果采用非门来实现反变量, 则会增加电路的复杂性。可以通过逻辑公式变换的方式, 在**不需要非门的情况下**, 实现这种逻辑电路。

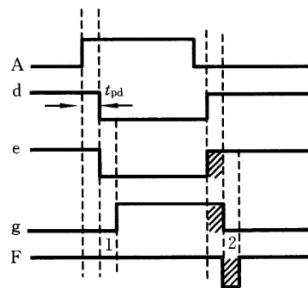
有非门的电路:  $F = /A \cdot B + B \cdot /C + A \cdot /B \cdot C + A \cdot C \cdot /D$



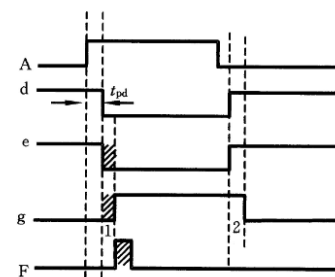
通过逻辑公式变换, **不需要非门**的电路;  $F = /(/(B \cdot /(A \cdot C)) + /(A \cdot C \cdot /(B \cdot D)))$

## • 组合逻辑电路的险象：

- 在实际逻辑电路中，信号经过同一电路中的不同路径所产生的时延通常是不一样的，即输入信号经过不同路径到达输出端的时间有先后，这种现象称为**竞争现象**。
- 由于电路中存在竞争现象，使得输入信号的变化可能引起输出信号出现非预期的错误输出，这个现象称为**险象**。险象包括“0”型险象和“1”型险象。
- 通常将不产生错误输出的竞争称为**非临界竞争**，而导致错误输出的竞争称为**临界竞争**。



“0”型险象



“1”型险象

## • 险象的判断方法：

- **1、代数法：**如果逻辑函数表达式中有某个变量的原变量（ $x$ ）和反变量（ $\bar{x}$ ），则通过消去逻辑函数表达式中的其他变量，再看逻辑函数表达式是否会变为“ $x+\bar{x}$ ”或者“ $x\cdot\bar{x}$ ”的形式，若是，则说明该逻辑函数电路可能会产生险象。
- **2、卡诺图法：**当逻辑函数采用“与-或”表达式时，如果卡诺图中的两个卡诺圈存在“**相切**”关系，则该电路可能产生险象。

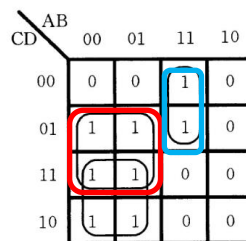


图 4.22 卡诺图

- 消除险象的方法有3种：

- 1、用**增加冗余项**的方法消除险象：在逻辑函数表达式中，“或”上多余的与项，或者“与”上多余的或项，使逻辑函数不可能在某种条件下变成“ $X+/X$ ”或“ $X\cdot/X$ ”的形式，从而消除可能产生的险象。

存在险象的逻辑函数： $F = /A\cdot C + B\cdot /C\cdot D + A\cdot /B\cdot /C$

冗余项： $/A\cdot B\cdot D$ 、 $A\cdot /C\cdot D$

增加冗余项后的逻辑函数： $F = /A\cdot C + B\cdot /C\cdot D + A\cdot /B\cdot /C + /A\cdot B\cdot D + A\cdot /C\cdot D$

- 2、**增加惯性延时环节**：通常采用RC电路作为惯性延时环节，RC电路的时间常数（ $\tau = RC$ ）要求大于毛刺（尖脉冲）的宽度。当然 $\tau$ 也不能太大，否则会使正常的输出信号产生不允许的畸变。

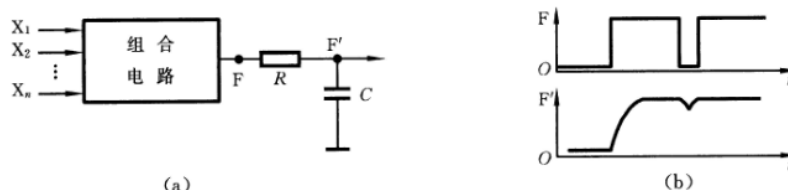


图 4.25 惯性延时环节

- 3、**选通法**：利用选通脉冲的作用，从时间上加以控制，以避免险象脉冲。选通法不需要增加任何器件。

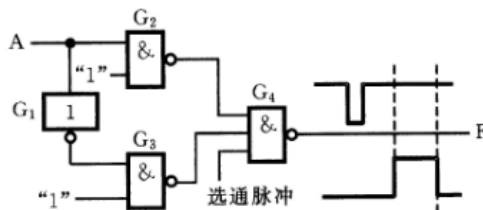


图 4.26 用选通法避开险象原理图

# 习题 (P114)

- 4.1
- 4.3
- 4.4
- 4.5
- 4.8
- 4.9
- 4.10
- 4.12

# 习题 (P114)

4.1 分析图 4.27 所示的组合逻辑电路,说明电路功能,并画出其简化逻辑电路图。

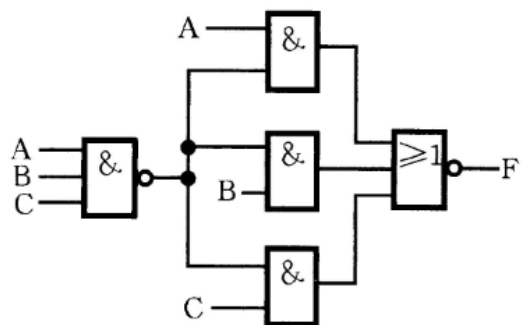


图 4.27 组合逻辑电路

# 习题 (P114)

4.2 分析图 4.28 所示的组合逻辑电路:(1) 指出在哪些输入取值下,输出 F 的值为 1;  
(2) 改用异或门实现该电路的逻辑功能。

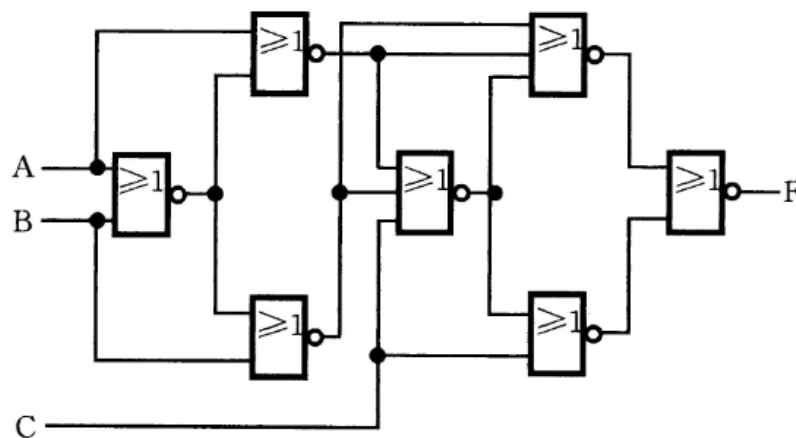


图 4.28 组合逻辑电路

# 习题 (P114)

4.3 分析图 4.29 所示组合逻辑电路,列出真值表,说明该电路的逻辑功能。

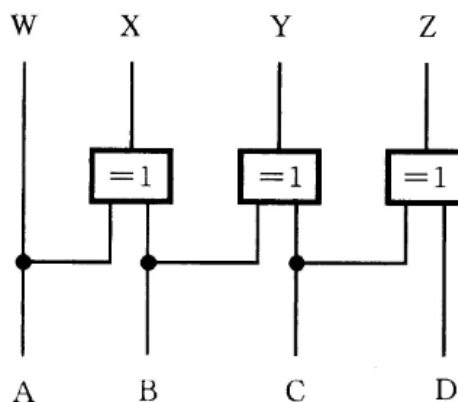


图 4.29 组合逻辑电路



# 习题 (P114)



4.4 设计一个组合逻辑电路,该电路输入端接收两个 2 位二进制数  $A=A_2A_1, B=B_2B_1$ 。当  $A>B$  时,输出  $Z=1$ ,否则  $Z=0$ 。

# 习题 (P114)



4.5 设计一个代码转换电路,将 1 位十进制数的余 3 码转换成 2421 码。

# 习题 (P114)

4.6 假定  $X=AB$  代表一个 2 位二进制数, 试设计满足如下要求的逻辑电路( $Y$  也用二进制数表示):

(1)  $Y=X^2$

(2)  $Y=X^3$

# 习题 (P114)

4.7 用与非门设计一个组合逻辑电路,该电路输入为 1 位十进制数的 2421 码,当输入的数字为素数时,输出  $F$  为 1,否则  $F$  为 0。

# 习题 (P114)



4.8 设计一个“四舍五入”电路。该电路输入为 1 位十进制数的 8421 码,当其值大于或等于 5 时,输出  $F$  的值为 1,否则  $F$  的值为 0。

# 习题 (P114)



4.9 设计一个检测电路,检测 4 位二进制码中 1 的个数是否为偶数。若为偶数个 1,则输出为 1,否则输出为 0。



# 习题 (P114)

4.10 设计一个加/减法器,该电路在  $M$  控制下进行加、减运算。当  $M=0$  时,实现全加器功能;当  $M=1$  时,实现全减器功能。

# 习题 (P114)

4.11 在输入不提供反变量的情况下,用与非门组成实现下列函数的最简电路。

$$(1) F = A\bar{B} + \bar{A}C + B\bar{C} \quad (2) F = A\bar{B}\bar{C} + BC\bar{D} + A\bar{C}\bar{D} + \bar{B}CD$$



# 习题 (P114)



4.12 下列函数描述的电路是否可能产生险象？在什么情况下产生险象？若产生险象，试用增加冗余项的方法消除。

$$(1) F_1 = AB + A\bar{C} + \bar{C}D \quad (2) F_2 = AB + \bar{A}CD + BC \quad (3) F_3 = (A + \bar{B})(\bar{A} + \bar{C})$$

# 作业样例

- 1.1 冯·诺依曼结构计算机的基本思想是什么？按此思想设计的计算机硬件系统应由哪些部件组成？它们各有何作用？

• 答：

- 数学家冯·诺依曼提出了计算机制造的三个基本原则，即采用二进制逻辑、程序存储执行（存储程序和程序控制）以及计算机由五个部分组成（运算器、控制器、存储器、输入设备、输出设备），这套理论被称为冯·诺依曼体系结构。
- 计算机硬件系统应由运算器、控制器、存储器、输入设备、输出设备等组成。
- 运算器是一种用于信息加工处理的部件，它对数据进行算术运算和逻辑运算。运算器通常由算术逻辑单元（ALU, Arithmetic and Logic Unit）和一系列寄存器组成。通常将运算器一次运算能处理的二进制位数称为机器字长。现代计算机具有多个寄存器，称为寄存器组。
- 控制器是整个计算机的指挥中心，它可使计算机各部件协调工作。计算机中有两股信息在流动，一股是控制流信息，另一股是数据流信息。控制流信息的发源地是控制器，控制器产生控制流信息的依据来自3个方面：指令寄存器、状态寄存器和时序电路。
- 存储器的主要功能是存放程序和数据，目前计算机的主存储器都是半导体存储器。
- 输入设备就是将信息输入计算机的外部设备，它将人们熟悉的信息形式转换成计算机能接收并识别的信息形式。
- 输出设备就是将计算机运算结果转换成人们和其他设备能接收和识别的信息形式的设备，如字符、文字、图形、图像、声音等。

# 关于作业提交

- **1周内**必须提交（上传到学院的FTP服务器上），否则认为是迟交作业；如果期末仍然没有提交，则认为是未提交作业。
  - 作业完成情况成绩=第1次作业提交情况\*第1次作业评分+第2次作业提交情况\*第2次作业评分+.....+第N次作业提交情况\*第N次作业评分。
  - 作业评分：A（好）、B（中）、C（差）三挡。
  - 作业提交情况：按时提交（1.0）、迟交（0.5）、未提交（0.0）。
- 请采用电子版的格式（**PPT文档**）上传到FTP服务器上，文件名取“学号+姓名+第X次作业.pptx”。
  - 例如：11920222202406+刘济华+第4次作业.pptx
- 下次上课时（2023年11月7日）会**随机抽取2位同学**到讲台上汇报作业。
- 第4次作业提交的截止日期为：**2023年11月6日晚上24点。**

**Thanks**