



第8章 逻辑代数基础

§ 8.1 数字电路概述

§ 8.2 逻辑运算与逻辑函数

§ 8.3 集成逻辑门电路

§ 8.4 逻辑函数的化简

§ 8.5 具有约束项的逻辑函数

§ 8.6 逻辑函数不同表达形式之间的相互转换



§ 8.1 数字电路概述

模拟电路：处理和传输模拟信号的电路。

模拟信号：

时间上连续：任意时刻有一个相对的值。

数值上连续：可以是在一定范围内的任意值。

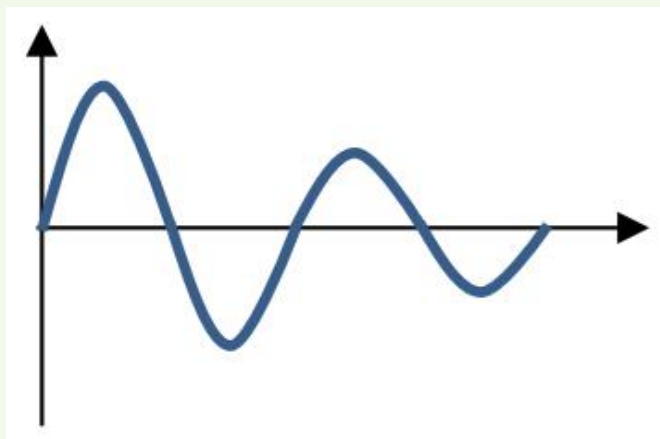
例如：电压、电流、温度、声音等。

优点：用精确的值表示事物。

缺点：很难度量；

容易受噪声的干扰；

难以保存。



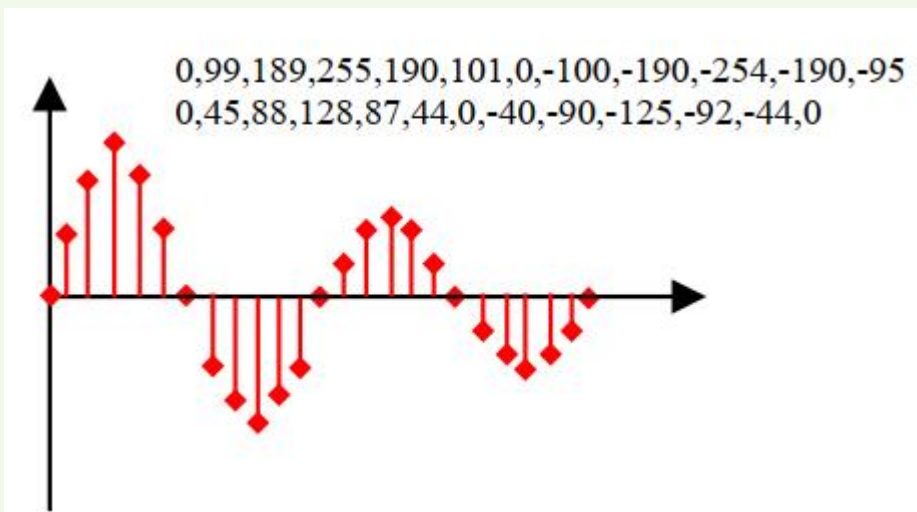
数字电路：处理和传输数字信号的电路。

数字信号：

时间上离散：只在某些时刻有定义。

数值上离散：只能是某一个最小单位的整数倍。

如原子的个数、汽车的数量等。



模拟信号可以用数字的形式来表示

即把模拟信号经过取样、量化、编码，就可转化成二进制的数字信息（由0、1数字串所构成的数字流）。



◆数字电路与模拟电路的比较

区别点		模拟电路	数字电路
工作信号		时间、数值都连续的 模拟信号	时间、数值都离散的 数字信号
基本构成单元 为半导体元件	工作状态	线性放大状态	开关状态
	外部电路条件	必须使器件工作在 线性放大区	必须使器件满足 开关工作的条件
研究的主要问题		研究电路对输入信号的 放大和变换功能	研究电路输出和输入之间的 逻辑关系
基本单元电路		放大器	逻辑门和触发器



◆ 数字电路的优点

现代社会是一个
数字化的时代！

抗干扰能力强，精度高；

数字信号易保存和读取；

基本电路结构简单，适合集成和系列化生产。

通用性好，可采用标准化的逻辑部件构成；

数字电路在日常生活、自动控制、测量仪器、
通信等领域得到广泛应用！



➤ 数字电视

有线电视传输信号

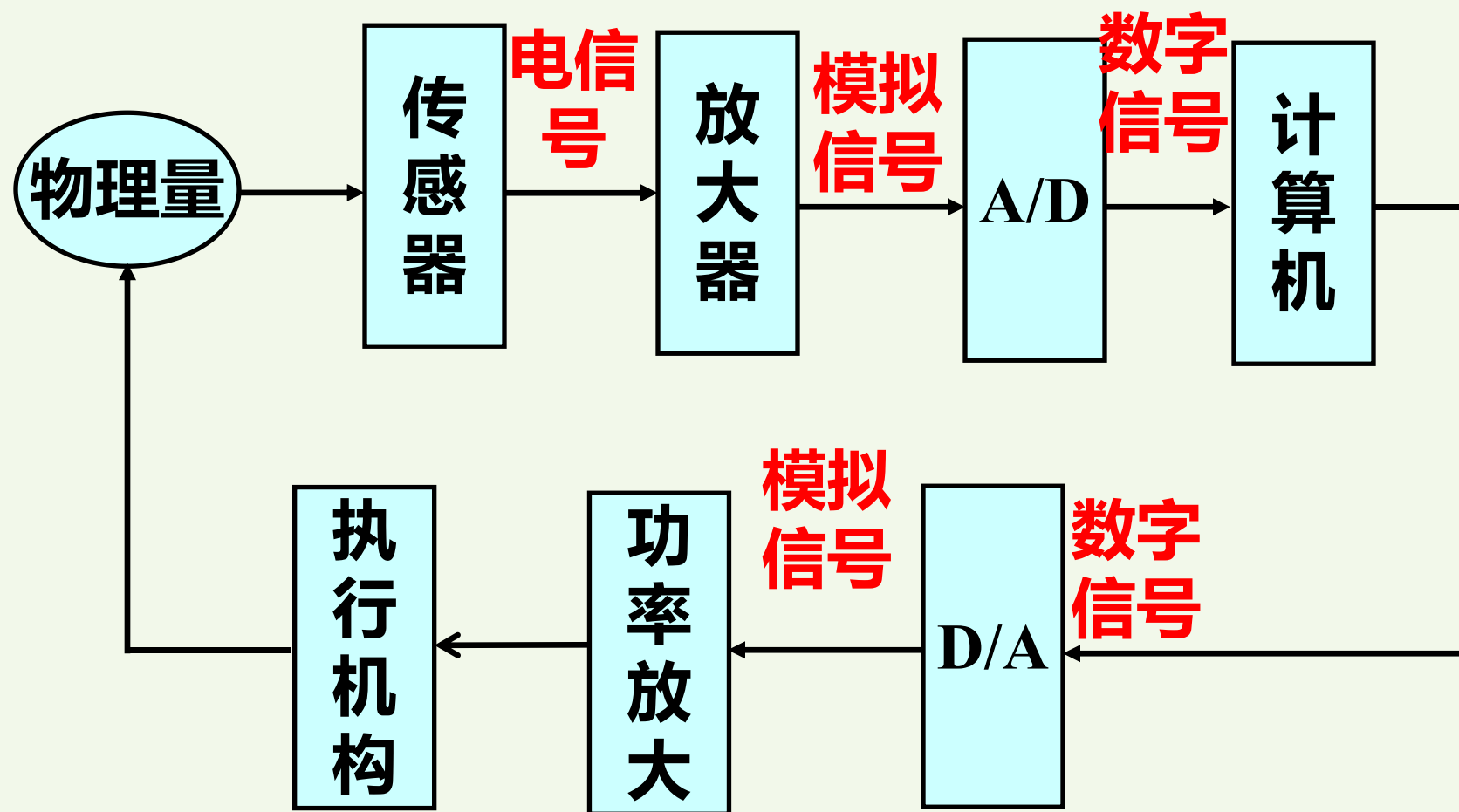
{ 过去：模拟信号
目前：数字信号



将传统的模拟电视信息经过采样、量化、编码，转化成二进制的数字信息，然后进行处理、传输、存储和记录，通过**机顶盒**接收、解码、转换成AV信号再输入到电视机播放。



典型的数字测量控制系统框图





数字电路主要学习内容

- 逻辑代数基本知识
- 组合逻辑电路
- 时序逻辑电路
- 模数转换电路



8.1.2 数制与编码

数制：多位数码中每一位的构成方法以及由低位到高位进位的规则。

数制	基数	数码	计数规则
十进制	10	0~9	逢十进一
二进制	2	0、1	逢二进一
八进制	8	0~7	逢八进一
十六进制	16	0~9、 ABCDEF	逢十六进一

常用的数制有：
十进制、二进制、八进制、十六进制等。

目前在微型计算机系统中普遍采用8位、16位、32位二进制并行运算，而它们分别可以用2位、4位、8位十六进制数表示，因为用**十六进制数**书写程序十分方便，成为当前的**主流程序书写模式**。



数制转换

1. 各种进制转换为十进制

规则：按“权”展开法

任一 N 进制数 D 的十进制展开式为： $(D)_N = \sum k_i N^i$

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13D$$

$$(347.5)_8 = 3 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 5 \times 8^{-1} = 231.625D$$

$$(2C.5)_{16} = 2 \times 16^1 + 12 \times 16^0 + 5 \times 16^{-1} = 44.3125D$$



2.十进制转换为其他进制

规则:

整数部分: 除基取余, 逆序排列 (除基逆取余)

小数部分: 乘基取整, 顺序排列 (乘基顺取整)

例如: $(35.78)_{10} = (100011.1100011)_2$

解: 整数部分

$35 \div 2 = 17$	余数1
$17 \div 2 = 8$	余数1
$8 \div 2 = 4$	余数0
$4 \div 2 = 2$	余数0
$2 \div 2 = 1$	余数0
$1 \div 2 = 0$	余数1



小数部分

$0.78 \times 2 = 1.56$	整数1
$0.56 \times 2 = 1.12$	整数1
$0.12 \times 2 = 0.24$	整数0
$0.24 \times 2 = 0.48$	整数0
$0.48 \times 2 = 0.96$	整数0
$0.96 \times 2 = 1.82$	整数1
$0.82 \times 2 = 1.64$	整数1





3. 二进制、八进制、十六进制之间的相互转换

规则：以小数点为基准，分别向左、向右分组

每三位二进制数对应一位八进制数

每四位二进制数对应一位十六进制数

$$\text{例：} (10011101.11)_2 = (\underline{010} \ \underline{011} \ \underline{101} . \underline{110})_2 = (235.6)_8$$

$$(10011101.11)_2 = (\underline{1001} \ \underline{1101} . \underline{1100})_2 = (9D.C)_{16}$$

$$(4FD.A)_{16} = (\underline{0100} \ \underline{1111} \ \underline{1101} . \underline{1010})_2$$



八进制 $\xleftrightarrow{\text{二进制}}$ 十六进制

$$(532.7)_8 = (101\ 011\ 010.111)_2$$

$$= (\underline{0001}\ \underline{0101}\ \underline{1010}.\ \underline{1110})_2$$

$$= (15A.E)_{16}$$

不同进制数的对照举例



十进制 (D)	二进制 (B)	八进制 (O)	十六进制 (H)
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F



编码：用二进制数表示文字、符号等信息的过程

□最常用的编码为**二进制类编码**与**ASCII码**

ASCII码是一种用7位二进制数码表示数字、字母或符号的代码，是计算机通用的标准代码。

在二进制类编码中，以**二-十进制码(BCD码)**使用最多，即用**四位二进制代码**来表示**0~9十个字符**，如8421BCD码、2421BCD码、5421BCD码、余三码、格雷码等。

注：BCD是Binary-Coded Decimal的缩写



几种常见的BCD代码

8421码+3

循环码

编码种类	8421码	2421码	5421码	余3码	格雷码
十进制数	8421码	2421码	5421码	余3码	格雷码
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1	0 0 1 1
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 0	0 0 1 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1	0 1 1 0
5	0 1 0 1	1 0 1 1	1 0 0 0	1 0 0 0	0 1 1 1
6	0 1 1 0	1 1 0 0	1 0 0 1	1 0 0 1	0 1 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0	1 0 1 0	0 1 0 0
8	1 0 0 0	1 1 1 0	1 0 1 1	1 0 1 1	1 1 0 0
9	1 0 0 1	1 1 1 1	1 1 0 0	1 1 0 0	1 1 0 1
权	8 4 2 1	2 4 2 1	5 4 2 1	无权码	无权码

将十进制数2574用841BCD码表示



□循环码:

- 相邻两个代码只有一位状态不同（相邻性）；
- 首尾两个代码也仅有一位不同（循环性）；
- 以中间为对称的每两个代码仅第一位状态不同；（反射性）

1位循环码

$\left\{ \begin{array}{c} 0 \\ 1 \end{array} \right.$

2位循环码

$\left\{ \begin{array}{cc} 0 & 0 \\ 0 & 1 \\ \hline 1 & 1 \\ 1 & 0 \end{array} \right.$

3位循环码

$\left\{ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ \hline 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{array} \right.$



课堂练习

第一节1, 2, 3, 4, 5, 6





§8.2 逻辑运算与逻辑函数

8.2.1 逻辑代数的三种基本运算

◆**逻辑代数**：研究和揭示逻辑运算关系的一门学科，它是逻辑电路分析和设计的数学基础。

由英国科学家**乔治·布尔**所创立，所以逻辑代数又称为布尔代数。

◆**逻辑代数和普通代数的比较**：

相同之处：都是研究函数与变量之间的关系，

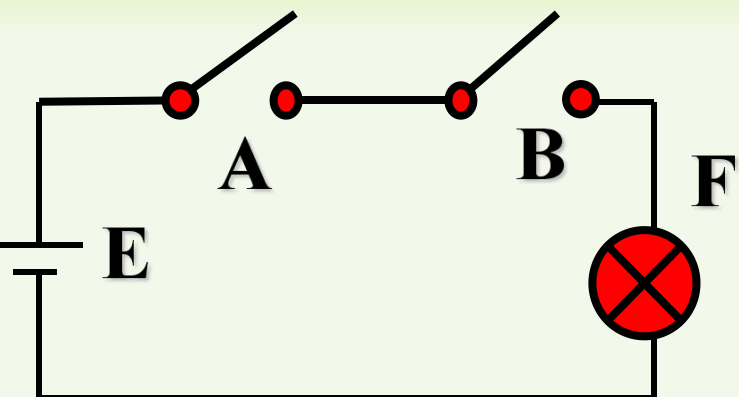
且都是用字母（ A 、 B 、 C 、 F 等）表示**变量**；

不同之处：如下表所示：

区别	基本运算种类	取值范围
普通代数	加、减、乘、除	$-\infty \sim +\infty$
逻辑代数	与、或、非	0或1 ，表示两种不同的 逻辑状态

◆ “与” 逻辑

只有当决定某一事件的所有条件都满足时，该事件才发生。



开关 A、B { 打开 “0”
 闭合 “1”

灯F { 灭 “0”
 亮 “1”

功能表

A	B	F
开	开	灭
开	合	灭
合	开	灭
合	合	亮

真值表

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

列写真值表时应注意：

❖ n 个输入变量，共有 2^n 种输入状态组合

❖ 最好按二进制数由小到大的顺序列写



◆ “与” 逻辑

□ 逻辑表达式: $F = A \bullet B = AB$

“与” 运算又称逻辑 “乘”

运算规律:

$$0 \bullet 0 = 0 \qquad 0 \bullet 1 = 0$$

$$1 \bullet 0 = 0 \qquad 1 \bullet 1 = 1$$

有0出0, 全1出1

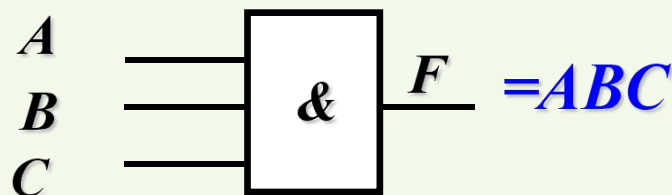
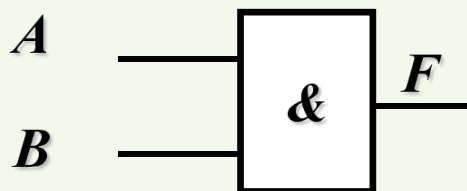
A B	F
0 0	0
0 1	0
1 0	0
1 1	1



◆ “与” 逻辑

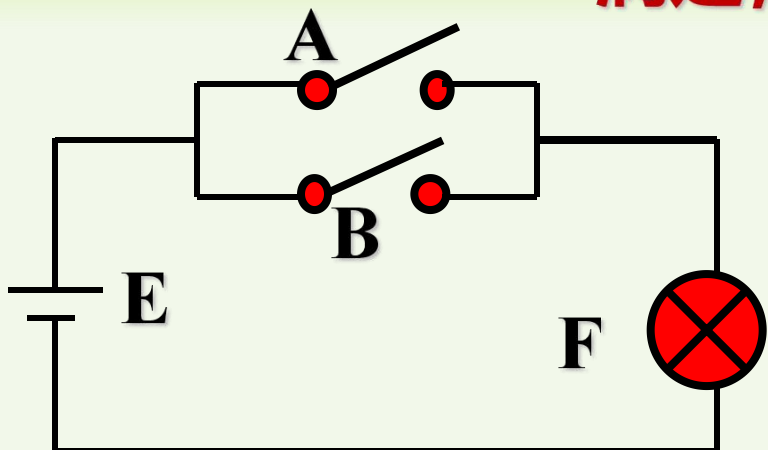
在逻辑电路中，把能实现“与”运算的基本单元电路称为“与”门。

□ “与” 门的逻辑符号：



◆ “或” 逻辑

决定某一事件的所有条件中只要有一个满足，事件就会发生。



开关 { 打开 “0”
 闭合 “1”

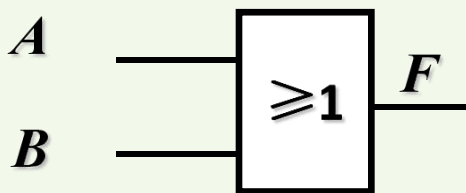
灯 { 灭 “0”
 亮 “1”

□ 真值表

<i>A</i>	<i>B</i>	<i>F</i>	
0	0	0	$0 + 0 = 0$
0	1	1	$0 + 1 = 1$
1	0	1	$1 + 0 = 1$
1	1	1	$1 + 1 = 1$

□ 逻辑表达式: $F=A+B$
“或” 运算又称逻辑 “加”

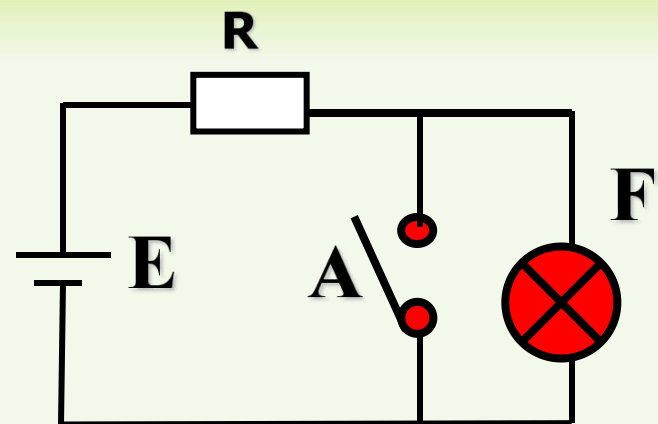
□ 逻辑符号为:



有1出1，全0出0

◆ “非” 逻辑

事件的结果总是与事件的条件相反。



开关 { 打开 “0”
 闭合 “1”

灯 { 灭 “0”
 亮 “1”

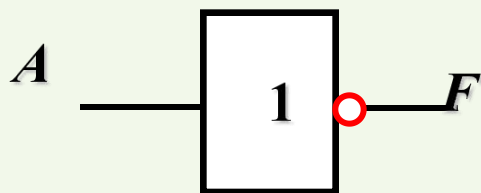
□ 真值表

A	F
0	1
1	0

□ 逻辑表达式: $F = \bar{A}$

“非” 运算的运算规则: $\bar{0} = 1$ $\bar{1} = 0$

□ 逻辑符号为:



注意:

“非” 门只有一个输入端



8.2.2 常用的复合逻辑运算

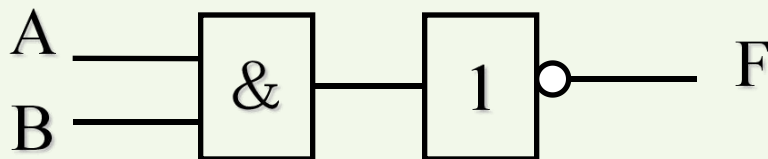
在逻辑代数中，除了“与”、“或”、“非”三种基本逻辑运算外，经常用到的还有一些由这三种基本运算构成的复合运算：

- “与非” 运算
- “或非” 运算
- “与或非” 运算
- “异或” 运算
- “同或” 运算



◆ “与非” 逻辑

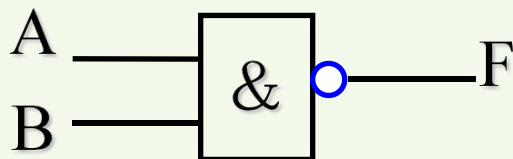
□ 逻辑结构:



□ 逻辑表达式:

$$F = \overline{AB}$$

□ 逻辑符号:



常用芯片: 74LS00, 74LS20

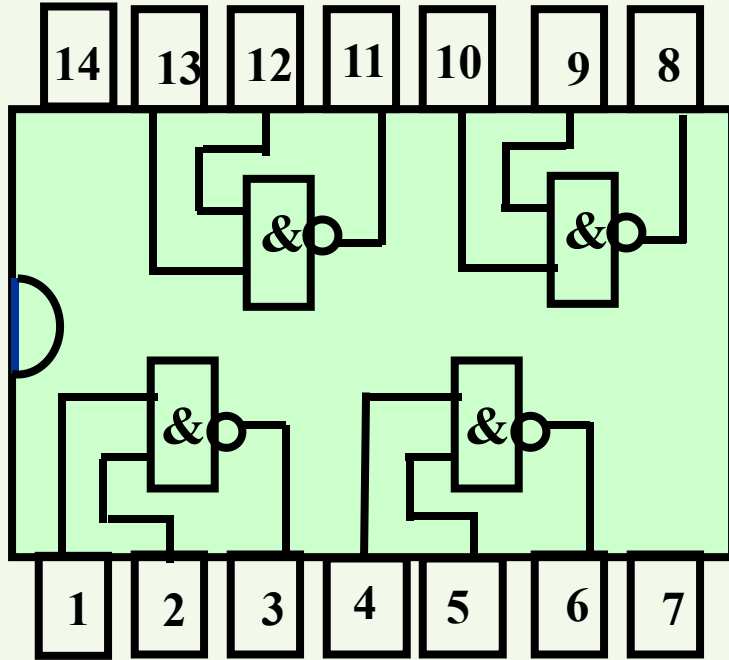
□ 真值表:

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

有0出1, 全1出0

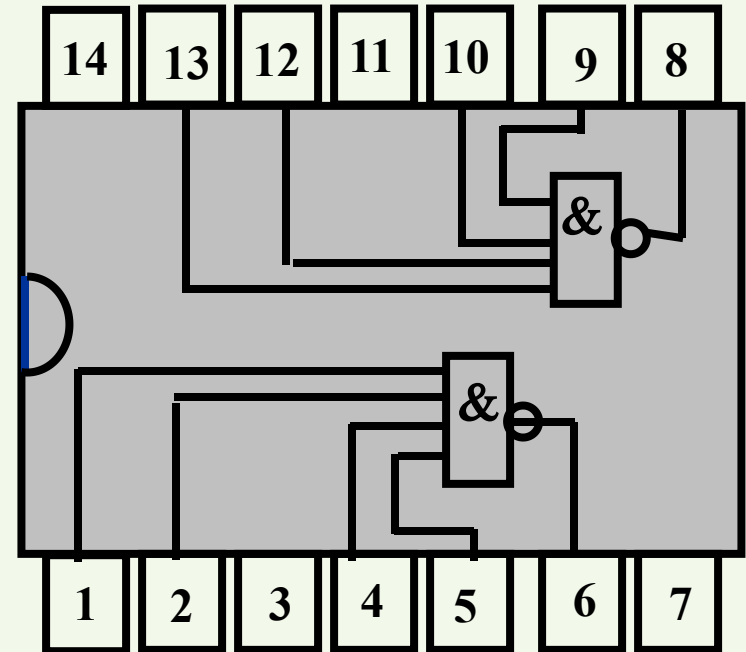


74LS00（四2输入与非门）



“四”表示有4个与非门，“2”表每个门有两个输入端。7脚接地，14脚接+5V电源

74LS20（双4输入与非门）

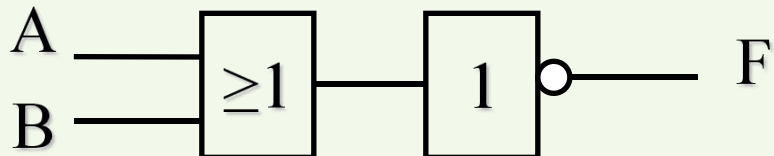


7脚接地，14脚接+5V电源



◆ “或非” 逻辑

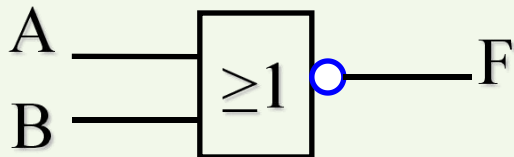
□ 逻辑结构:



□ 逻辑表达式:

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

□ 逻辑符号:



□ 真值表:

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

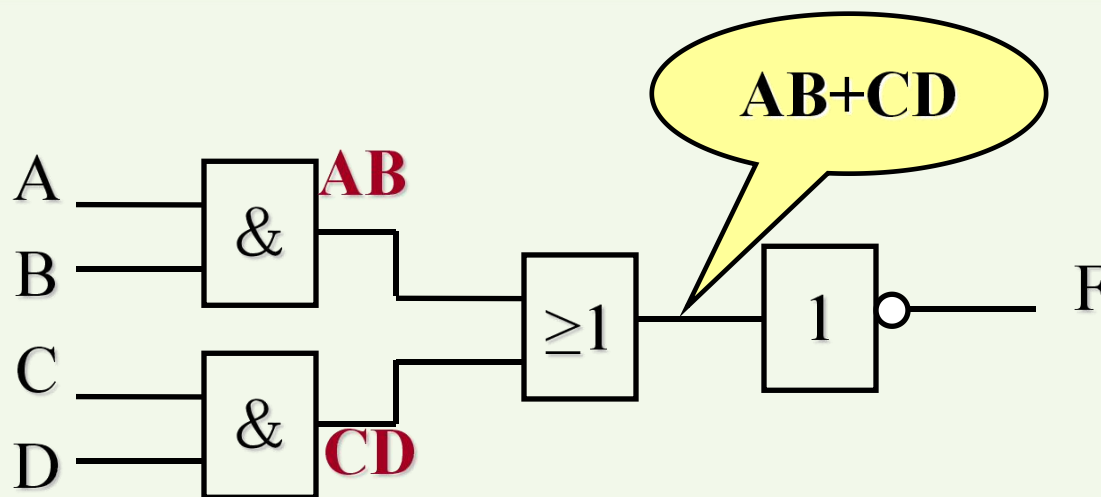
有1出0，全0出1

常用芯片: 74LS02, 74LS27



◆ “与或非” 逻辑

□ 逻辑结构:

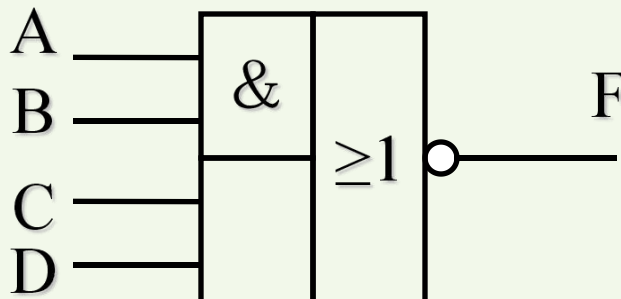


□ 逻辑表达式:

$$F = \overline{AB + CD}$$

常用芯片:
74LS54, 74LS64

□ 逻辑符号:

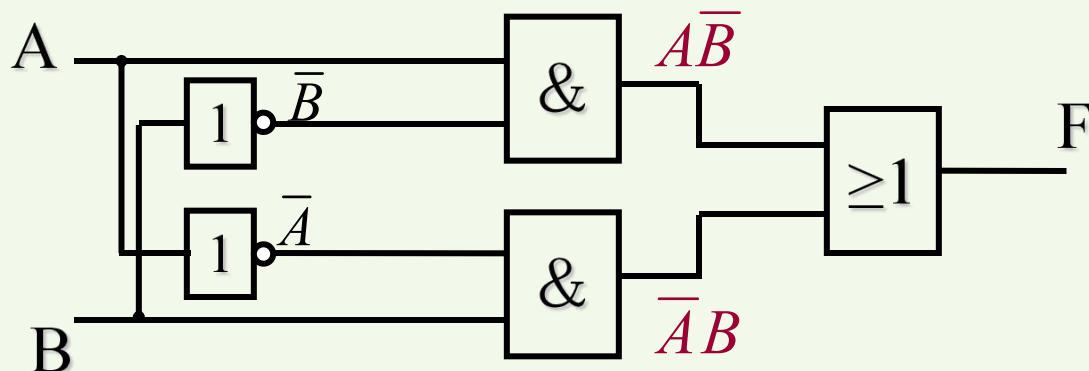




◆ “异或” 逻辑

□ 真值表:

□ 逻辑结构:



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

两个变量的异或:

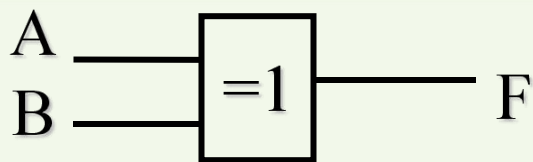
相异出**1**, 相同出**0**

□ 逻辑表达式:

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



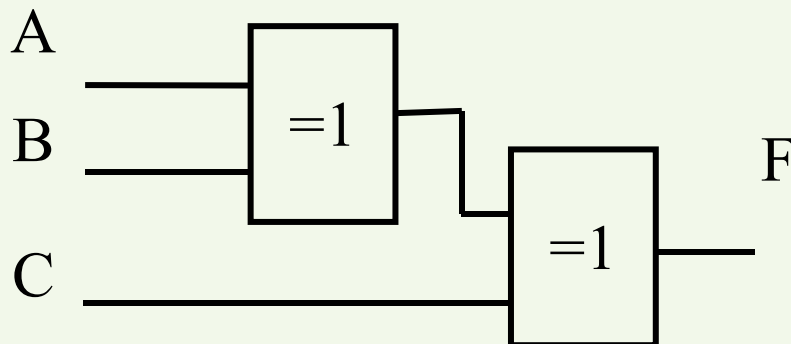
□ 异或门的逻辑符号：



常用芯片： 74LS86

注意： “异或” 门只有两个输入端

□ 如何实现 $F = A \oplus B \oplus C$



◆ “同或” 逻辑

□ 真值表:

A B	F_1	F
0 0	0	1
0 1	1	0
1 0	1	0
1 1	0	1

$$F_1 = A \oplus B$$

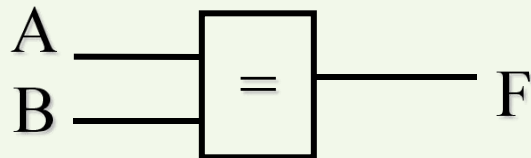
两个变量的同或:

相同出**1**, 相异出**0**

□ 逻辑表达式:

$$\begin{aligned} F &= \overline{A \oplus B} \\ &= A \odot B \\ &= \overline{A} \overline{B} + AB \end{aligned}$$

□ 逻辑符号:



$$\overline{A \oplus B} = A \odot B$$

$$\overline{A \odot B} = A \oplus B$$

等式

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

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

□注意: $A \oplus B \oplus C \neq \overline{A \odot B \odot C}$

A	B	C	$A \oplus B \oplus C$	$A \odot B \odot C$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$A \oplus B \oplus C = A \odot B \odot C$$

根据“同或”和“异或”的运算规律，还可得到以下运算公式：

$$(1) A \oplus A = 0; \quad A \oplus \bar{A} = 1; \quad A \oplus 1 = \bar{A}; \quad A \oplus 0 = A$$

$$(2) A \odot A = 1; \quad A \odot \bar{A} = 0; \quad A \odot 1 = A; \quad A \odot 0 = \bar{A}$$

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

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

$$A \oplus \bar{A} = \bar{A} \cdot \bar{A} + A \cdot \bar{\bar{A}} = \bar{A} + A = 1$$

$$A \odot A = A \cdot A + \bar{A} \cdot \bar{A} = A + \bar{A} = 1$$

$$A \oplus 0 = \bar{A} \cdot 0 + A \cdot \bar{0} = A$$

$$A \odot 0 = A \cdot 0 + \bar{A} \cdot \bar{0} = \bar{A}$$

$$(A \oplus B) \oplus C = (A \oplus C) \oplus B$$

$$(A \oplus B) \oplus \bar{A} = (A \oplus \bar{A}) \oplus B = 1 \oplus B = \bar{B}$$



8.2.3 逻辑函数的表示方法

逻辑函数的 表示方法

真值表

逻辑表达式

逻辑图

波形图

卡诺图

◆真值表



将 n 个输入变量的 2^n 个取值组合与其对应的输出函数值一一列出来的一个表格。

例如：设计一个三人（A、B、C）表决使用的逻辑电路，当多数人赞成（输入为1）时表决结果F有效，输出为1，否则F为0。

输入有 $2^3=8$ 个不同的取值组合，把8种输入组合下对应的输出值列成表格，可得到真值表。

<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
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



◆逻辑表达式

逻辑表达式的形式有多种，其中“与或”表达式是最基本的表达形式，由“与或”表达式可以转换成其他各种形式。

(1)标准与或式

在真值表中列出所有的取值组合——乘积项（与项），将这些乘积项中使函数为1的项（最小项）加起来，即得到标准与或式。

将所有最小项相加

标准与或式

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

为了表达方便，通常可以将最小项用 m_i 来表示

$$F(A,B,C) = m_3 + m_5 + m_6 + m_7$$

乘积项

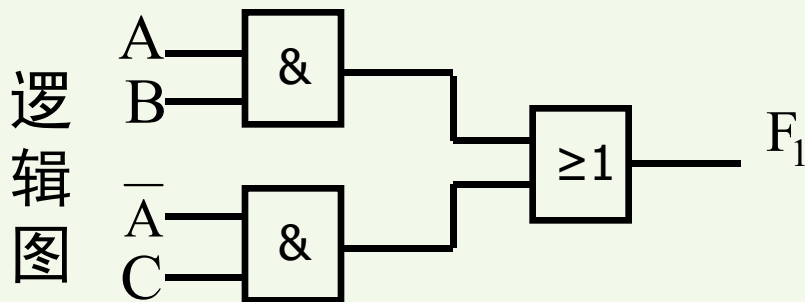
		A	B	C	F
m_0	$\overline{A}\overline{B}\overline{C}$	0	0	0	0
m_1	$\overline{A}\overline{B}C$	0	0	1	0
m_2	$\overline{A}B\overline{C}$	0	1	0	0
m_3	$\overline{A}BC$	0	1	1	1
m_4	$A\overline{B}\overline{C}$	1	0	0	0
m_5	$A\overline{B}C$	1	0	1	1
m_6	$AB\overline{C}$	1	1	0	1
m_7	ABC	1	1	1	1

注意：变量以A为最高位、C为最低位

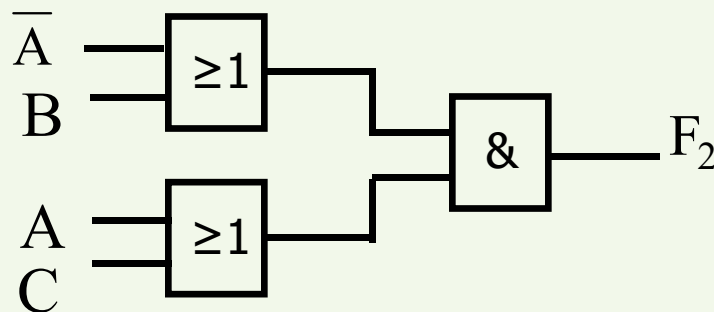


(3)其他形式的逻辑表达式

$$F_1 = AB + \bar{A}C \quad \text{“与或”式}$$

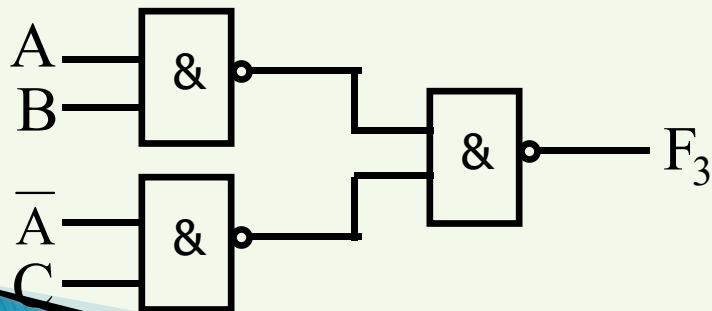


$$F_2 = (\bar{A} + B)(A + C) \quad \text{“或与”式}$$



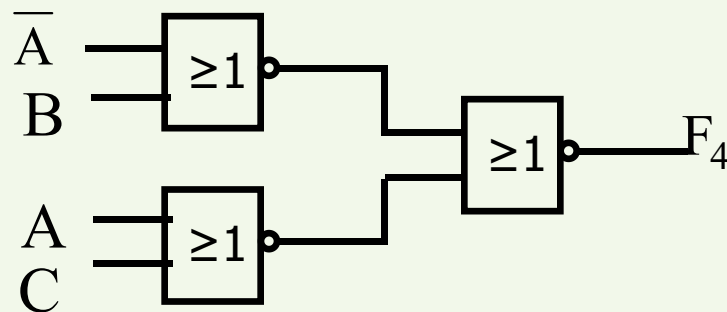
$$F_3 = \overline{\overline{AB} \overline{AC}}$$

“与非—与非”式



$$F_4 = \overline{\overline{\bar{A} + B} \overline{A + C}}$$

“或非—或非”式





由“与或”式求
“与非-与非”式

$$\begin{aligned} F_1 &= AB + \overline{A}C \\ &= \overline{\overline{AB + \overline{A}C}} \\ &= \overline{AB} \overline{AC} \\ &= F_3 \end{aligned}$$

摩根定律

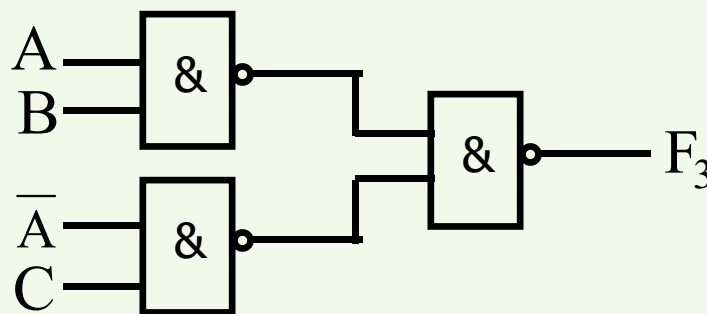
摩根定律：

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

◆ 逻辑图

逻辑图是用逻辑符号表示逻辑关系的图形表示法。

$$F_3 = \overline{\overline{AB} \overline{AC}}$$



逻辑图

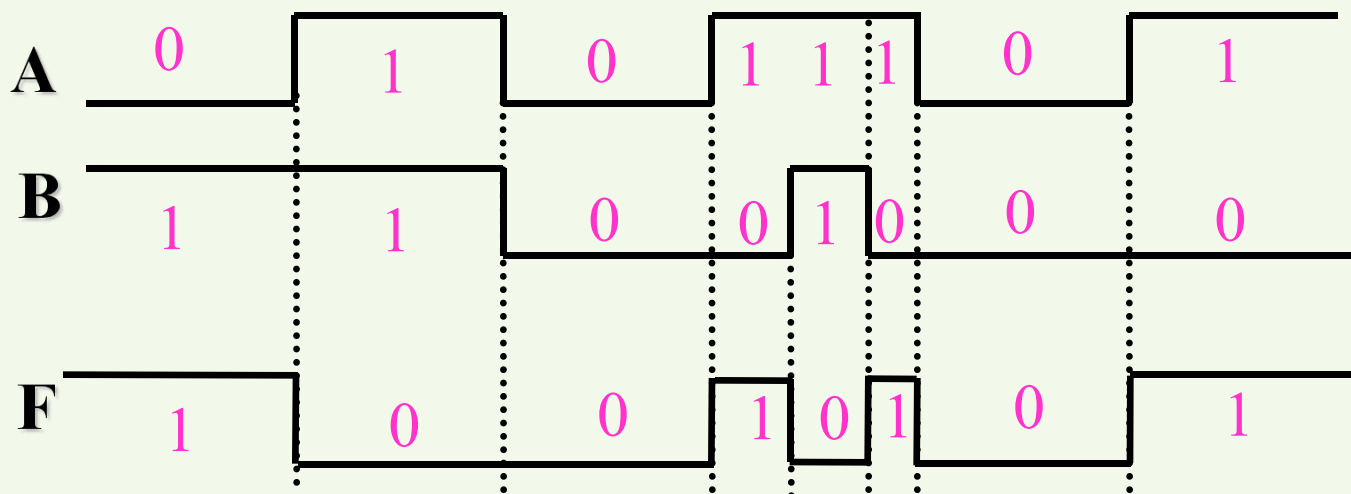
由于逻辑图形符号都有对应的集成电路器件，所以能很方便的将逻辑图实现为具体的硬件电路。



□ 波形图:

--- 也叫时序图，用变量随时间变化的波形来反映**输出与输入之间一一对应关系**的一种图形表示法

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





课程小结

1、数字电路概述

- 模拟信号和数字信号
- 数字电路的特点
- 数制与码制
 - 数制：二进制、八进制、十进制、十六进制
 - 码制：二-十进制编码（BCD码）

2、逻辑运算与逻辑函数

- 逻辑代数的基本概念
- 三种基本逻辑运算及其门电路
 - 与逻辑，或逻辑，非逻辑
- 常用复合逻辑运算
 - 与非，或非，与或非，异或，同或
- 逻辑函数的表示方法
 - 真值表，逻辑表达式，逻辑图，波形图



课堂练习
第二节1, 4, 5, 6,
7, 8, 10





8.3 集成逻辑门电路

实现逻辑关系的单元电路称为门电路。

◆ 分立元件构成的逻辑门电路

✧ 工作原理

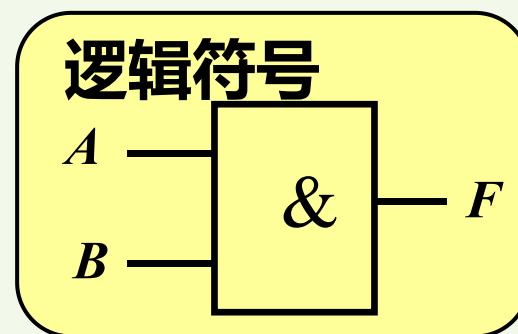
✧ 二极管构成的“与”门电路

F 与 A 、 B 间的逻辑关系可表示为:

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

$F = A \cdot B$

U_A (V)	U_B (V)	U_F (V)
0	0	0.7
0	5	0.7
5	0	0.7
5	5	5





◆ **集成逻辑门** - - 把晶体管、电阻及电路连线等制作在一块半导体的基片上，并封装在一个壳体內的逻辑门电路。

➤ **集成逻辑门电路的分类**

按**基本组成元件**可分为：

TTL (*Transistor-Transistor Logic*):

以**双极型三极管**作为开关器件

CMOS (*Complementary Metal-oxide-Semiconductor*):

由**NMOS**和**PMOS**互补组合而成（一种场效应管）

两者性能比较：

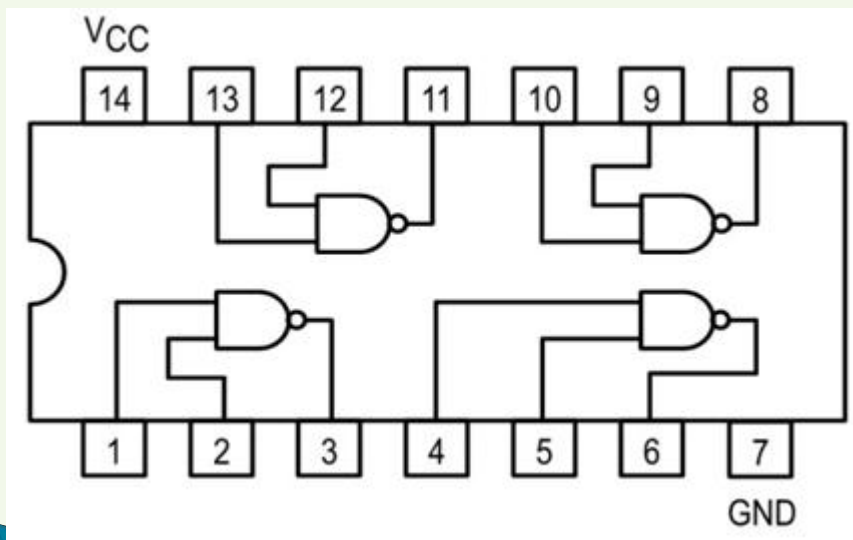
TTL: 电路速度快，功耗较大

CMOS: 电路速度慢，功耗很低

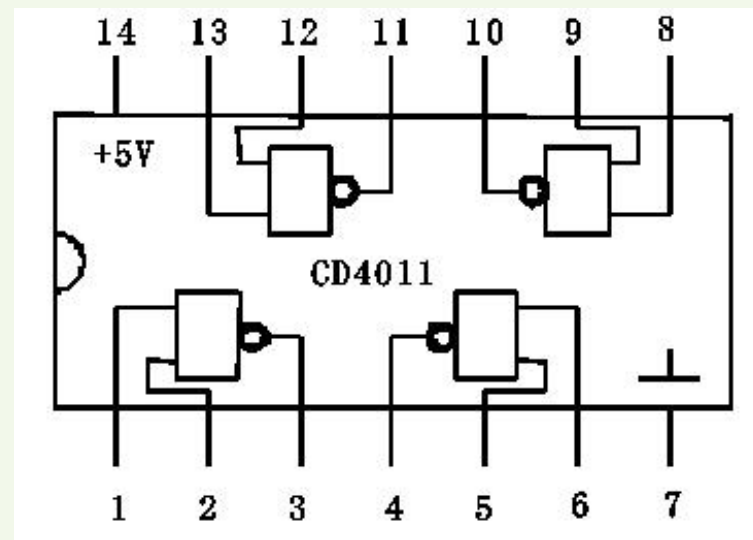
➤ TTL集成逻辑门电路

- TTL产品系列:** 74XX, 74HXX, 74SXX, **74LSXX**

不同: 平均传输延迟时间和平均功耗有差异。
相同: 其他参数和外引线彼此相容, 结构特点相同, 电气参数基本相同。



74LS00引脚图



4011引脚图



• TTL门电路的极限参数:

----用以保证芯片能够安全的工作

一般取5V

名称	符号	最大变化范围	单位
电源电压	V_{CC}	4.5~5.5	V
输入电压	V_{IN}	-0.5~5.5	V
输入电流	I_I	-3.0~+5.0	mA
环境温度	T_A	-55~+125	$^{\circ}\text{C}$



• 输入、输出电平:

保证门与门之间逻辑关系的正确性。

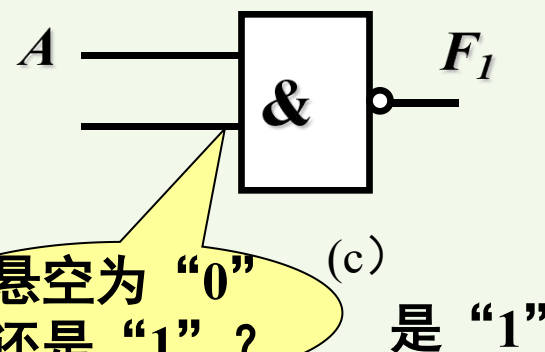
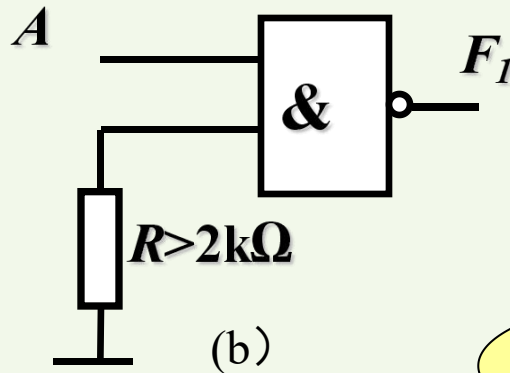
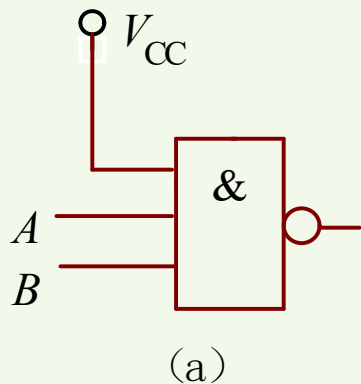
名称	符号	额定值(V)
最大输入低电平	$V_{IL(\max)}$	<0.8
最小输入高电平	$V_{IH(\min)}$	>2.0
最大输出低电平	$V_{OL(\max)}$	<0.4
最小输出高电平	$V_{OH(\min)}$	>2.4

注意: 在逻辑电路中, “高”、“低” 电平是一个离散的概念。

典型值: $\begin{cases} V_H = 3.6V \\ V_L = 0.2V \end{cases}$

与非门及与门多余输入端的处理

(1) 多余输入端接高电平。

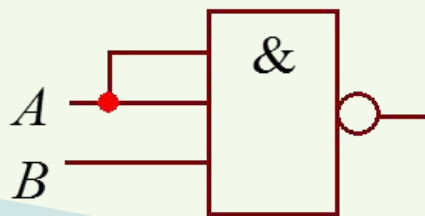


悬空为“0”
还是“1”？

是“1”

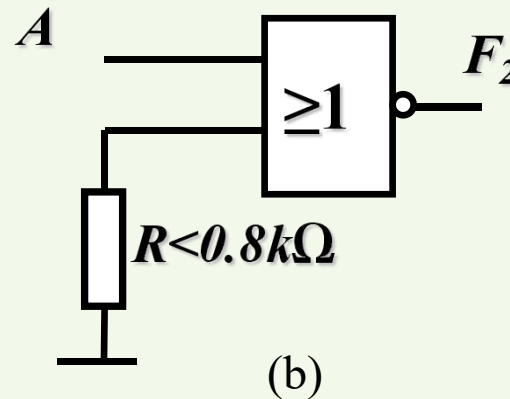
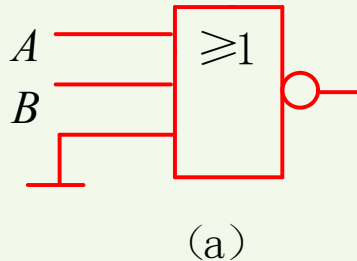
TTL的输入端是有电流的，外接电阻增大时，输入端变为高电平，此时的电阻值称为**开门电阻**。

(2) 将多余输入端和已使用的输入端并联使用。



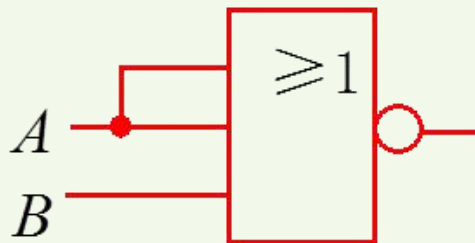
或非门及或门多余输入端的处理

(1) 多余输入端接**低**电平。



0.8kΩ称为**关门电阻**

(2) 将多余输入端和已使用的输入端**并联**使用。





➤ CMOS集成门电路的特点和注意事项

特点: 1) 功耗低

2) 工作电源电压范围宽

3) 抗干扰能力强

4) 带负载能力强

5) 输出电压幅度大

使用注意事项:

1) 多余的输入端不能悬空

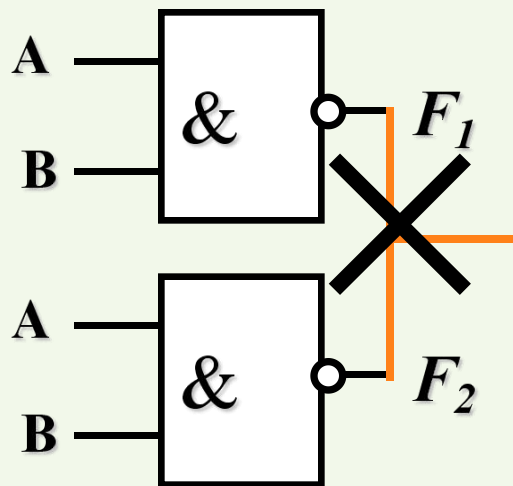
2) 注意输入电路的过流保护

3) 电源电压极性不能反接，以防止输出短路。



8.3.2 特殊TTL门电路

◆ 集电极开路“与非”门（OC门）

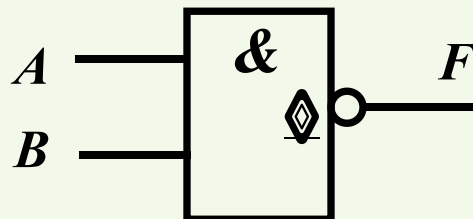


(1) 电流过大可能损坏
(2) 输出逻辑混乱
 $F=?$

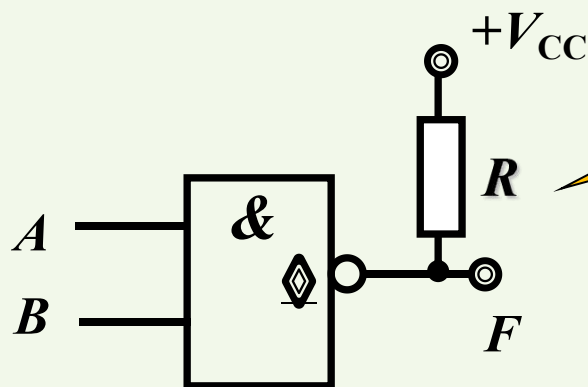
“与非”门的输出端不可以直接联接在一起！

但是集电极开路“与非”门的输出端可以直接联接在一起！

(1) OC “与非” 门的逻辑符号



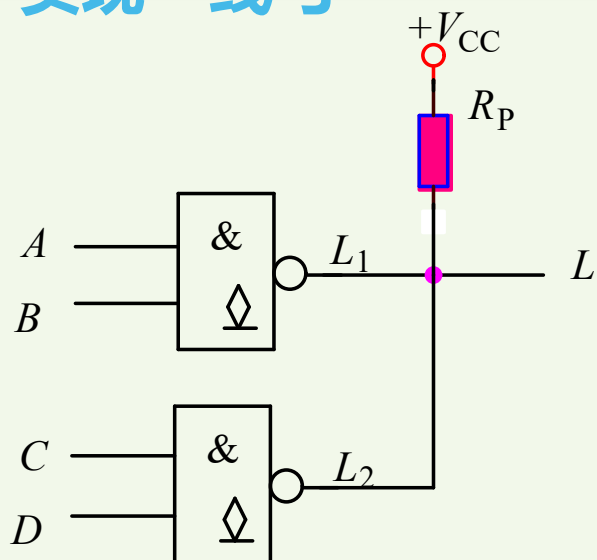
OC门在使用时一定要外接电阻



注意:根据输出高低电平的数值选择阻值

(2) OCL门的应用:

(a) 实现“线与”

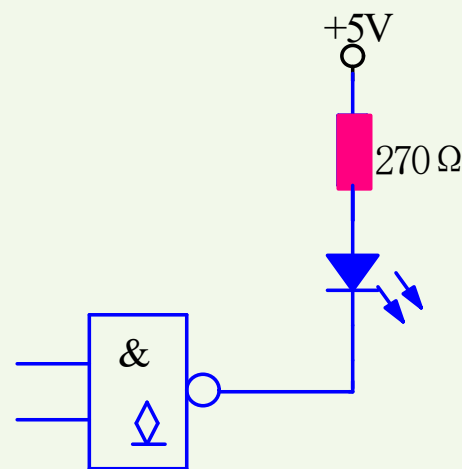


逻辑关系为:

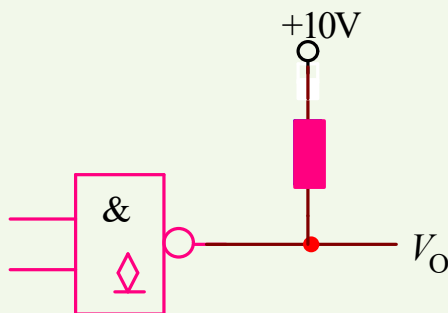
$$L = L_1 \cdot L_2 = \overline{AB} \cdot \overline{CD}$$

(c) 用做驱动器

如图是用来驱动发光二极管的电路。



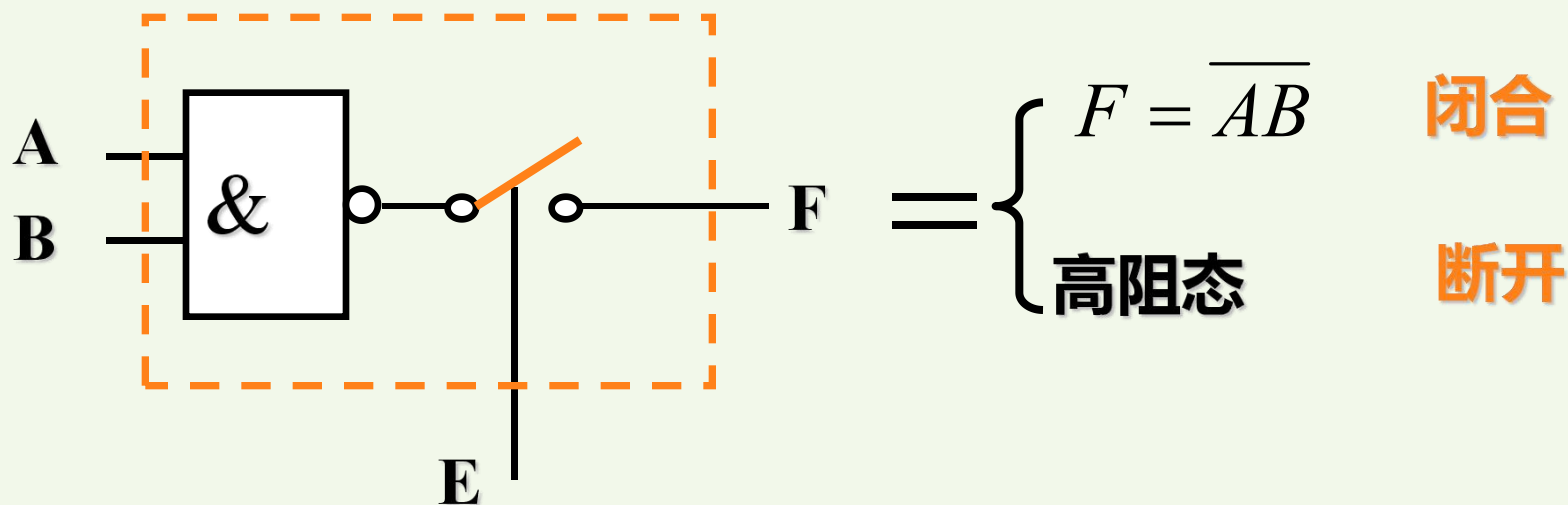
(b) 实现电平转换



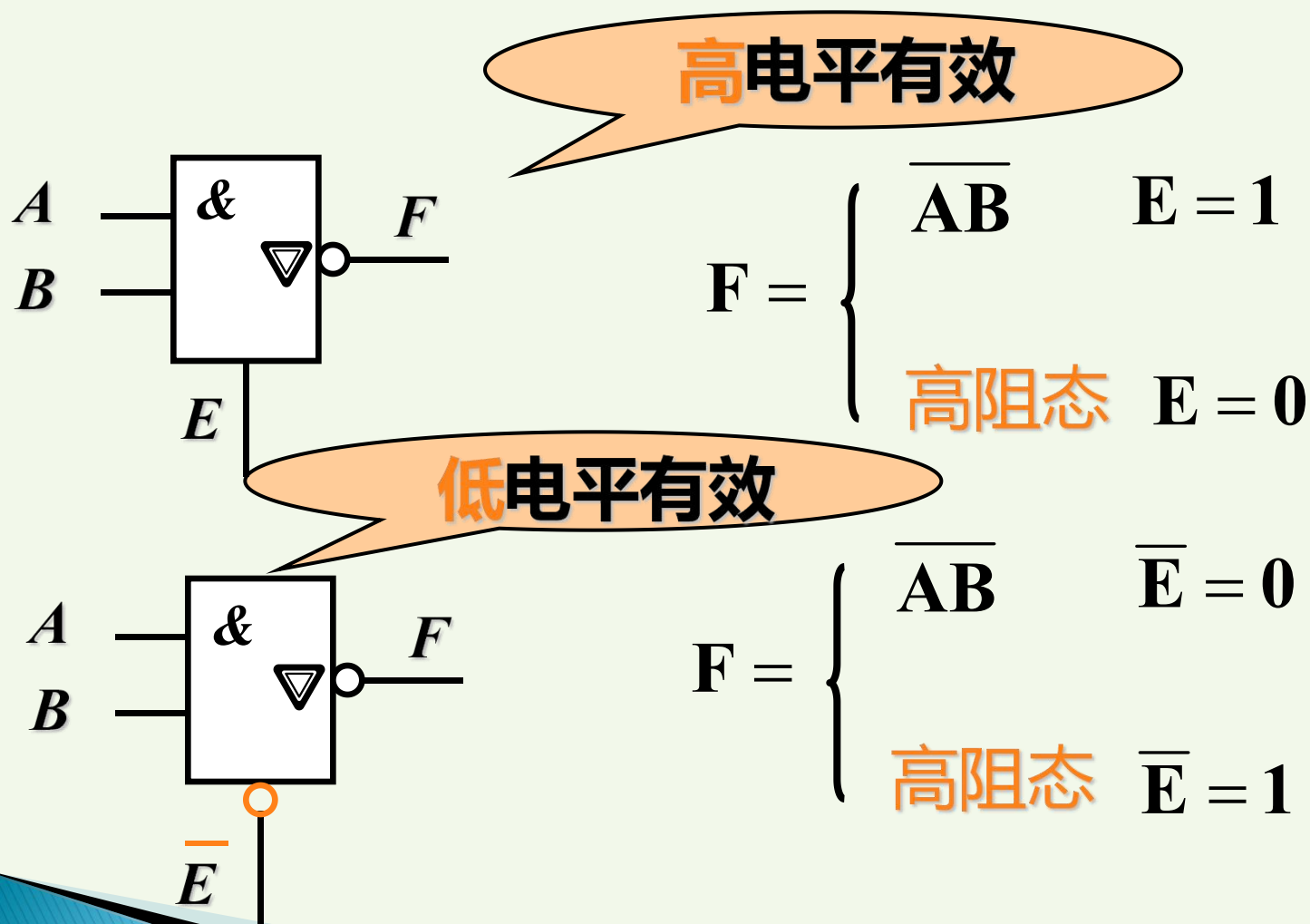
外接合适的电阻，可使
输出高电平变为10V。

◆ 三态门

---输出不但有 “0”， “1”二个状态；
还具有 “高阻态”的**第三状态**.



(1) 三态门的逻辑符号



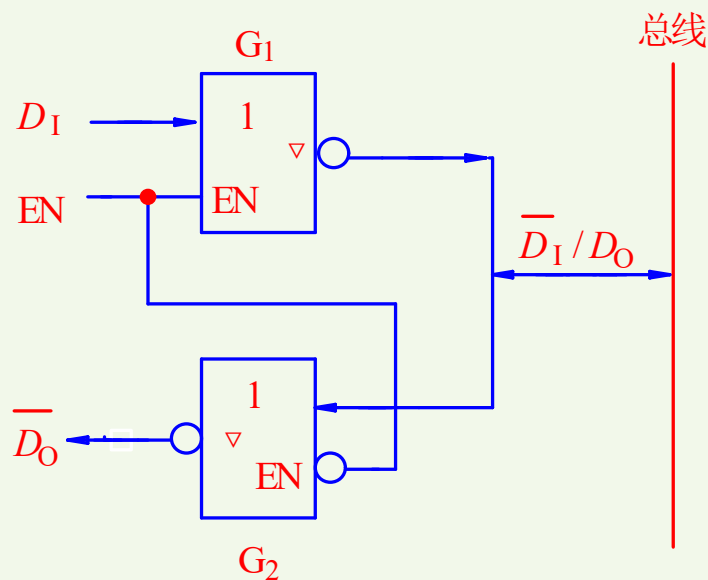
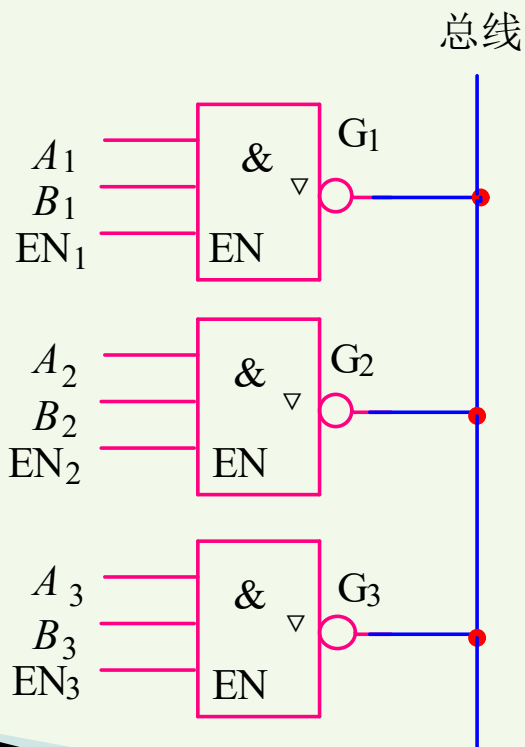


(2) 三态门的应用:

三态门在计算机总线结构中有着广泛的应用。

(a) 组成单向总线，实现信号的分时单向传送。

(b) 组成双向总线，实现信号的分时双向传送。



控制E使各端口数据分时单向传输



课堂练习

第三节1, 3, 5





8.4.1 逻辑代数的基本公式

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot \bar{A} = 0$$

$$A \cdot A = \bar{\bar{A}} = A$$

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

$$(AB)C = A(BC)$$

$$A \cdot (B + C) = AB + AC$$

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

$$A + 1 = 1$$

$$A + 0 = A$$

$$A + \bar{A} = 1$$

$$A + A = A$$

$$A + B = B + A$$

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

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

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

0-1律

互补律

重叠律

交换律

结合律

还原律

分配律

反演律



反演律也称摩根定律

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

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

- 用真值表证明摩根定律

A	B	\overline{A}	\overline{B}	\overline{AB}	$\overline{A} + \overline{B}$	$\overline{A + B}$	$\overline{A} \overline{B}$
0	0	1	1	1	1	1	1
0	1	1	0	1	1	0	0
1	0	0	1	1	1	0	0
1	1	0	0	0	0	0	0

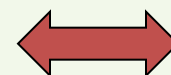
$$\overline{ABC} = \overline{A} + \overline{B} + \overline{C}$$

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



“与”形式

转换



“或”形式



常用逻辑公式

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

$$A + AB = A$$

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

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

$$\begin{aligned} & \because AB + \bar{A}C + BC \\ &= AB + \bar{A}C + BC(A + \bar{A}) \\ &= AB + \bar{A}C + ABC + \bar{A}BC \\ &= AB + \bar{A}C \end{aligned}$$

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

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

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

$$= A + B$$

对偶式

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

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

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

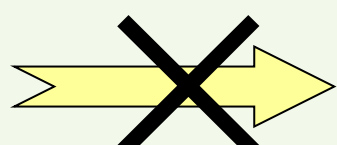
冗余律

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

讨论



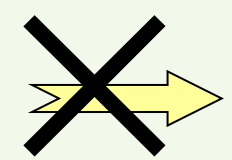
$$AB + \bar{A}C + BC = AB + \bar{A}C$$

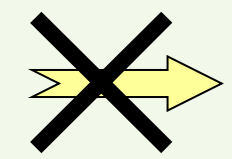
 $BC = 0$?

◆注意:

逻辑代数中不存在减法运算和除法运算!

逻辑等式左右两边相同的因子不能消去。

例如 $A \cdot A = A$  $A=1$

$A + A = A$  $A=0$



逻辑公式 三大规则

代入规则

对偶规则

反演规则



◆代入规则

在任何一个逻辑等式中，如果将等式两边所有出现的某个变量都代之以一个逻辑函数，则等式仍然成立。

例： $\overline{A \bullet B} = \overline{A} + \overline{B}$

$A = CD$

$\longrightarrow \overline{CD \bullet B} = \overline{CD} + \overline{B}$

$\longrightarrow \overline{BCD} = \overline{B} + \overline{C} + \overline{D}$

例： $A + \overline{A}B = A + B \longrightarrow \begin{cases} \overline{A} + AB = \overline{A} + B \\ A + \overline{A}BC = A + BC \end{cases}$



◆对偶规则

若 $F = G$ 则 F' (F 的对偶式) = G' (G 的对偶式)

$$F \left\{ \begin{array}{l} 1 \rightarrow 0 \\ 0 \rightarrow 1 \\ "\bullet" \rightarrow "+" \\ "+" \rightarrow "\bullet" \end{array} \right. \xrightarrow{\text{F的对偶式}} F' \quad G \left\{ \begin{array}{l} 1 \rightarrow 0 \\ 0 \rightarrow 1 \\ "\bullet" \rightarrow "+" \\ "+" \rightarrow "\bullet" \end{array} \right. \xrightarrow{\text{G的对偶式}} G'$$

例如:

$$\overset{F}{A \cdot (B + C)} = \overset{G}{AB + AC}$$

则

$$\overset{F'}{A + BC} = \overset{G'}{(A + B)(A + C)}$$

逻辑代数的很多公式都是对偶式



$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot \bar{A} = 0$$

$$A \cdot A = A$$

$$\overline{\overline{A}} = A$$

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

$$(AB)C = A(BC)$$

$$A \cdot (B + C) = AB + AC$$

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



$$A + 1 = 1$$

$$A + 0 = A$$

$$A + \bar{A} = 1$$

$$A + A = A$$

$$A + B = B + A$$

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

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

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

◆ 反演规则

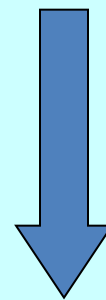
已知逻辑函数表达式 F

$$F \left\{ \begin{array}{l} 1 \rightarrow 0 \\ 0 \rightarrow 1 \\ "\bullet" \rightarrow "+" \\ "+" \rightarrow "\bullet" \end{array} \right. \Rightarrow \bar{F}$$

原变量 \rightarrow 反变量
反变量 \rightarrow 原变量

例:

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



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



应用反演规则时应**注意**:

- ❖ “先括号、然后乘、最后加”的运算顺序
- ❖ 不属于单个变量上的非号应保持不变

例: $F = \overline{A\overline{B} + C(A + \overline{D})}$



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

8.4.2 逻辑函数的公式化简法



◆公式化简法

在“与或”表达式的基础上，利用逻辑函数的公式消去表达式中多余的乘积项和每个乘积项中多余的因子，从而得出逻辑函数的**最简“与或”式**。

常用的方法有：

- 并项法
- 吸收法
- 消项法
- 消因子法
- 配项法



●并项法:

利用 $A + \overline{A} = 1$, 将两项合并成一项

例1: $F_1 = \underline{A\overline{B}\overline{C}} + \underline{AB\overline{C}} + \underline{ABC} + \underline{A\overline{B}C}$

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

● 吸收法:

含有多重“非”运算时，可应用摩根定理将其展开

根据 $A + AB = A$ 可将 AB 项消去

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

例2: $F_2 = A + \overline{\overline{A}CD}(B + \overline{E}D) + CD$

↓

$$= A + (\underline{A + CD})(B + \overline{E}D) + CD$$

$$= (A + CD) + (\cancel{A + CD})(B + \overline{E}D)$$

$$A + AB = A$$

$$= A + CD$$



● 消项法:

根据 $AB + \overline{A}C + BC = AB + \overline{A}C$ 可将 BC 项消去。

例3: $F_3 = ABC + \underline{\overline{A}D} + \overline{C}D + BD$

摩根定律

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

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

$$= ABC + \overline{A}CD + \cancel{BD}$$

展开为“与或”式

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

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

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



● 消因子法:

利用 $A + \bar{A}B = A + B$ 消去多余因子

例4: $F_4 = AB + \bar{A}C + \bar{B}C$
 $= AB + C(\bar{A} + \bar{B})$
 $= AB + \overline{ABC}$
 $= AB + C$

摩根
定理

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



● 配项法:

利用 $A + A = A$ 重复写入某一项

或根据 $AB + \bar{A}C = AB + \bar{A}C + BC$ 增加BC项

例5:
$$F_5 = \underline{ABC} + \bar{A}BC + A\bar{B}C$$
$$= ABC + \bar{A}BC + A\bar{B}C + ABC$$
$$= BC + AC$$



例6:

$$F_6 = \overline{A}\overline{B} + \underline{AC} + \overline{ADE} + \underline{\overline{CD}}$$

$$= \overline{A}\overline{B} + \underline{\overline{ADE}} + \underline{AC} + \overline{CD} + \underline{AD}$$

$$= \overline{A}\overline{B} + \underline{AC} + \overline{CD} + \cancel{\underline{AD}} + E$$

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

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



例7: $F_7 = \underline{AB} + \bar{A}C + \bar{B}C$

配项法

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

合并
项法

$$A + \bar{A} = 1$$

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

吸收法

$$A + AB = A$$

$$= AB + C$$

➤ **另解:** $F_7 = AB + \bar{A}C + \bar{B}C$

摩根
定律

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

$$= AB + \underline{\bar{A}BC}$$

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

$$= AB + C$$



结 论

- 用公式化简法需要记的公式多，技巧性强。
- 用公式化简法时往往需要**经验**，而且**难以验证**所得表达式是否为最简式。



逻辑函数的卡诺图化简法

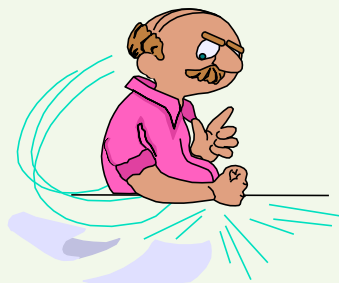
问题的引出：

公式化简法的不足：

公式太多、技巧性太强、很难判断化简结果是否为最简

?

那有没有更简单、更直观的化简方法呢？



卡诺图化简法！



1. 卡诺图的结构及其特点

➤ 什么是卡诺图？

卡诺图——由美国工程师卡诺首先提出的一种
用来描述逻辑函数的特殊方格图。

每一个小方格代表逻辑函数的一个最小项。

n 个变量 \longrightarrow 2^n 个最小项 \longrightarrow 2^n 个方格



➤ 卡诺图的结构

二变量的卡诺图

A \ B	0	1
0	$\overline{A}\overline{B}$	$\overline{A}B$
1	$A\overline{B}$	AB

每一个小方格 代表一个最小项

按最小项的十进制取值进行编号

注意：A为高位，B为低位

A \ B	0	1
0	m_0	m_1
1	m_2	m_3



三变量的卡诺图

变量的取值次序
按照**循环码**排列

BC \ A	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

卡诺图结构特点：

几何上**相邻**的两个小方格所代表的
最小项只有一个**变量不同**。

1位循环码

$\left\{ \begin{array}{l} 0 \\ 1 \end{array} \right.$

2位循环码

$\left\{ \begin{array}{ll} 0 & 0 \\ 0 & 1 \\ \hline 1 & 1 \\ 1 & 0 \end{array} \right.$

3位循环码

$\left\{ \begin{array}{lll} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ \hline 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{array} \right.$

四变量的卡诺图

按最小项的十进制
取值进行编号

AB \ CD		CD			
		00	01	11	10
00	00	$\overline{A}\overline{B}\overline{C}\overline{D}$	$\overline{A}\overline{B}\overline{C}D$	$\overline{A}\overline{B}CD$	$\overline{A}\overline{B}C\overline{D}$
	01	$\overline{A}\overline{B}C\overline{D}$	$\overline{A}\overline{B}CD$		$\overline{A}B\overline{C}\overline{D}$
11	11	$A\overline{B}\overline{C}\overline{D}$			
	10	$A\overline{B}C\overline{D}$			

AB \ CD		CD			
		00	01	11	10
00	00	m_0	m_1	m_3	m_2
	01	m_4	m_5	m_7	m_6
11	11	m_{12}	m_{13}	m_{15}	m_{14}
	10	m_8	m_9	m_{11}	m_{10}



小结

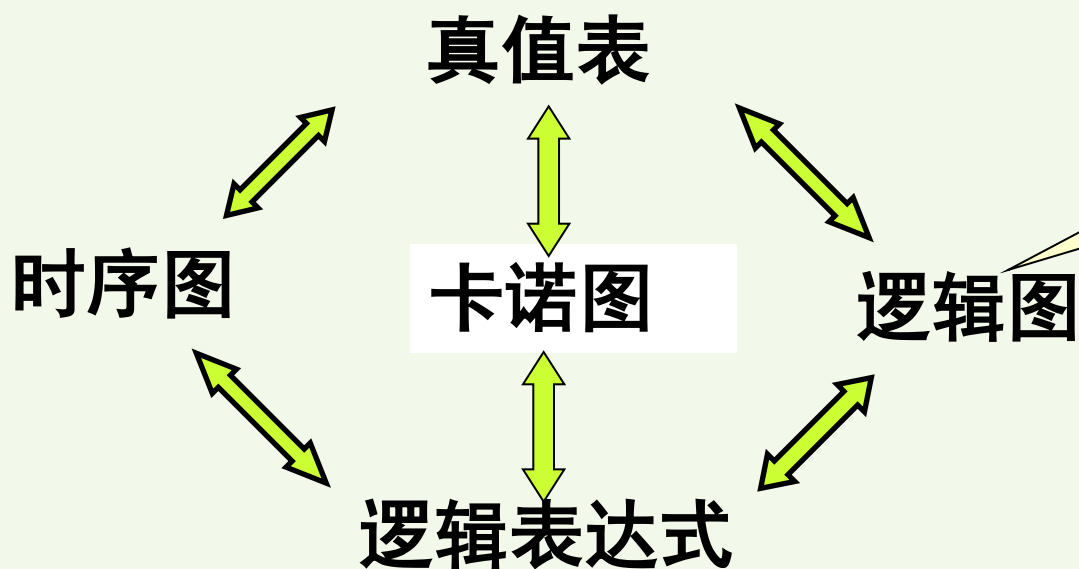
- (1) 卡诺图实质上是逻辑函数的**图形表示**；
- (2) n 个变量的卡诺图一共有 2^n 个小方格；
- (3) 几何上**相邻**的两个小方格所代表的最小项
只有一个**变量不同**。

卡诺图的优点：用几何位置的**相邻性**，形象地表达了构成函数的各个最小项在逻辑上的相邻性。



2. 如何用卡诺图表示逻辑函数？

逻辑函数的五种表达形式：



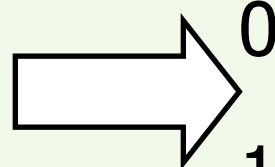
各种表达形式
可以相互转换



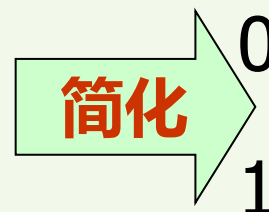
➤ 若已知函数的真值表：

卡诺图和真值表
实质上是相同的！

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



A \ B	0	1
0	0	0
1	0	1

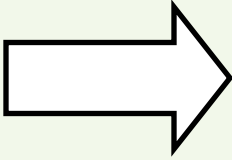


A \ B	0	1
0		
1		1

将在真值表中取值为“1”的最小项所对应的方格填“1”
取值为“0”的最小项所对应的方格填“0”

三变量的逻辑函数：

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



		BC			
		00	01	11	10
A	0		1		1
	1	1		1	1

另一形式

		C	
		0	1
AB	00		1
	01	1	
	11	1	1
	10	1	



➤ 若已知函数的标准“与或”式：

$$F(A, B, C, D) = \sum_m (1, 3, 5, 11, 14)$$

AB \ CD	00	01	11	10
00		1	1	
01		1		
11				1
10			1	

函数式中取值为“1”的最小项所对应的方格填“1”。



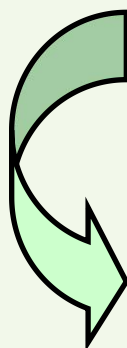
真值表
标准“与或”式 } \longrightarrow 卡诺图



非标准形式函数表达式 \longrightarrow 卡诺图?

利用 $A + \overline{A} = 1$

标准“与或”式





例：试用卡诺图表示逻辑函数：

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

解： $F = A\bar{B} + B\bar{C} + C\bar{A}$

		BC			
		00	01	11	10
A	0		1	1	1
	1	1	1		1

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

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

$$= \sum_m (1, 2, 3, 4, 5, 6)$$



非标准形式函数表达式 \Rightarrow 卡诺图?

最佳方法

“与或”式

方法：根据“与或”式中每个“与”项取值为1时变量的组合来填写相应的小方块。

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

$$A = 1, B = 0, C = \times$$

$$B = 1, C = 0, A = \times$$

$$A = 0, C = 1, B = \times$$

$$F = 1$$

		BC			
		00	01	11	10
A	0		1	1	1
	1	1	1		1



例：试用卡诺图表示逻辑函数：

$$F = A\bar{B}CD + \bar{B}\bar{C}D + AB\bar{D} + BCD\bar{D} + \bar{A}B\bar{C}$$

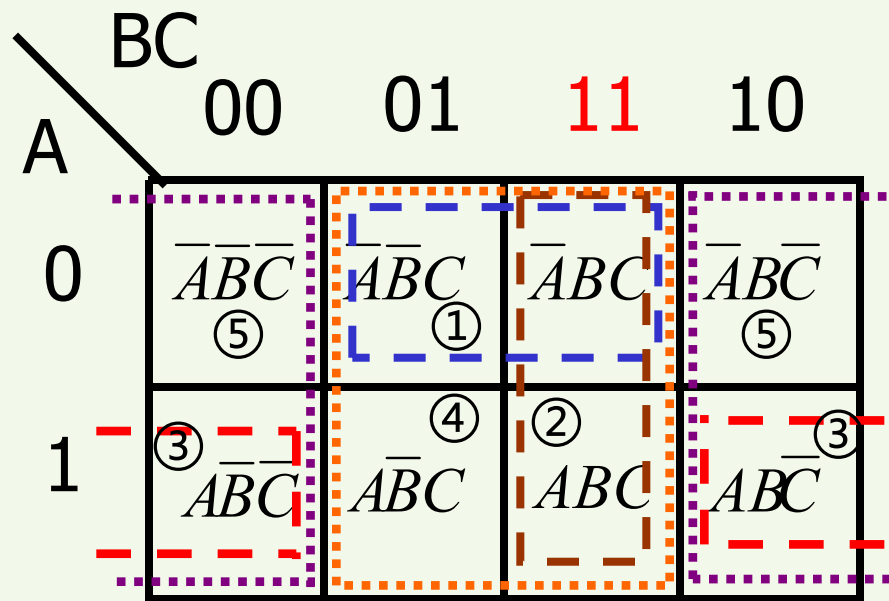
解：

AB \ CD	00	01	11	10
00		1		
01	1	1		1
11	1			1
10		1	1	



3. 利用卡诺图化简逻辑函数的基础:

在卡诺图中，几何上**相邻**的两项仅一个变量不同。因此相邻项可以消去因子进行合并。



总结：消去变化的量，
保留不变的量。

两项合并消去
一个变量

$$\left\{ \begin{aligned} F_1 &= \bar{A}\bar{B}C + \bar{A}BC = \bar{A}C \\ F_2 &= \bar{A}BC + A\bar{B}C = C \\ F_3 &= AC \end{aligned} \right.$$

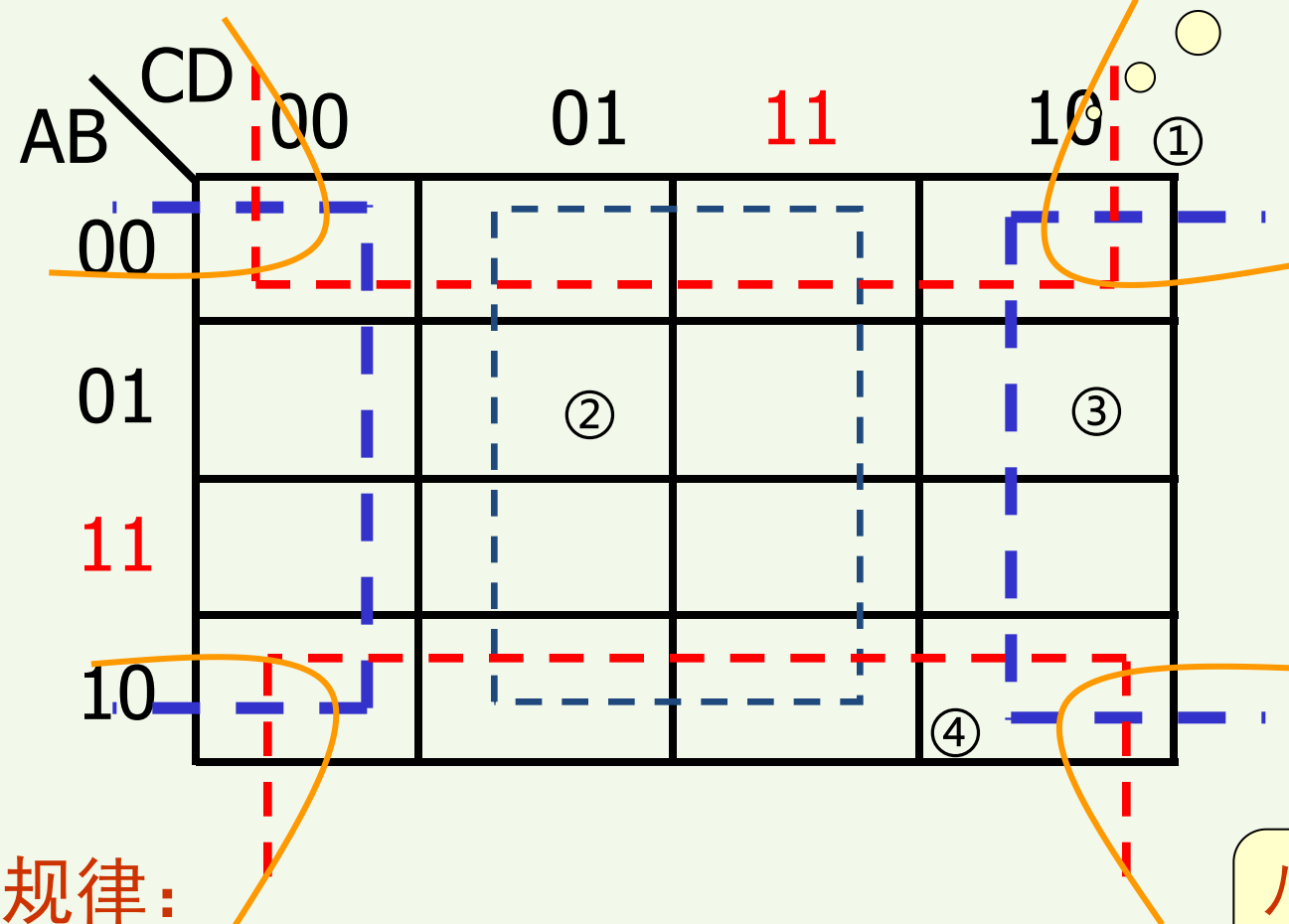
四项合并消
去二个变量

$$\left\{ \begin{aligned} F_4 &= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC \\ &= C \\ F_5 &= m_0 + m_2 + m_4 + m_6 \\ &= \bar{C} \end{aligned} \right.$$



可不可以六项、十项合并？

卡诺图中的“边”与“角”也是相邻的！



$F_1 = \overline{B}\overline{D}$

$F_2 = D$

$F_3 = \overline{D}$

$F_4 = \overline{B}$

规律：
 2^k 个相邻项合并消去k个变量

八项合并
消去三个变量



➤卡诺图化简逻辑函数的步骤

- (1) 画出该逻辑函数的卡诺图；
- (2) 按照“最少、最大”的原则（即圈的最少，圈内的最小项个数尽可能多）圈起所有取值为“1”的相邻项；
- (3) 对每一个矩形圈写出合并结果，再将各圈的结果相或即为所求的最简“与或”式。

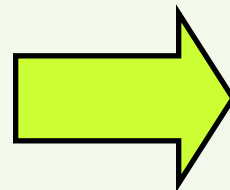


例：用卡诺图将函数F化为最简“与或”式。

$$F(A, B, C) = \sum_m (0, 1, 2, 4, 6)$$

解：

		BC			
		00	01	11	10
A	0	1	1		1
	1	1			1



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



例2：用卡诺图将函数F化为最简“与或”式。

$$F = A\bar{B} + B\bar{C}\bar{D} + ABD + \bar{A}B\bar{C}D + \bar{A}\bar{B}D$$

解：

AB \ CD	00	01	11	10
00		1	1	
01	1	1		
11	1	1	1	
10	1	1	1	1

$$F = A\bar{B} + AD + B\bar{C} + \bar{B}D$$



4. 卡诺图化简应注意的问题一

(1) 圈的个数要最少

CD \ AB	00	01	11	10
00	1	1		1
01		1	1	1
11			1	1
10				

CD \ AB	00	01	11	10
00	1	1		1
01		1	1	1
11			1	1
10				

这样圈
可以吗?





卡诺图化简应注意的问题二

(2) 圈要最大；且必须包含为 2^k 个“1”， $k=0,1,2,\dots$

AB \ CD				
	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11				
10		0	1	1

AB \ CD				
	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11				
10			1	1

这样圈
可以吗？

no



必须是 2^k 个
“1” 合并

AB \ CD				
	00	01	11	10
00	1	1	1	
01	1	1	1	
11				
10				

AB \ CD				
	00	01	11	10
00	1	1	1	
01	1	1	1	
11				
10				

这样圈
可以吗?

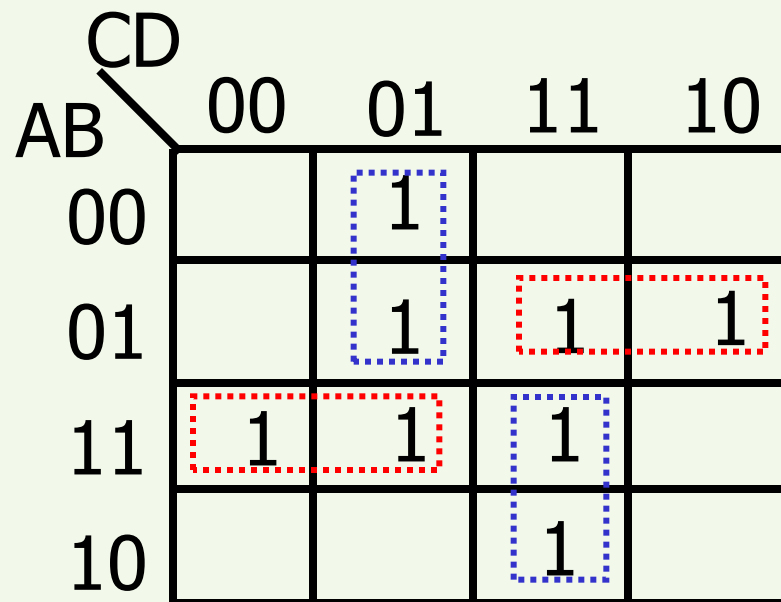
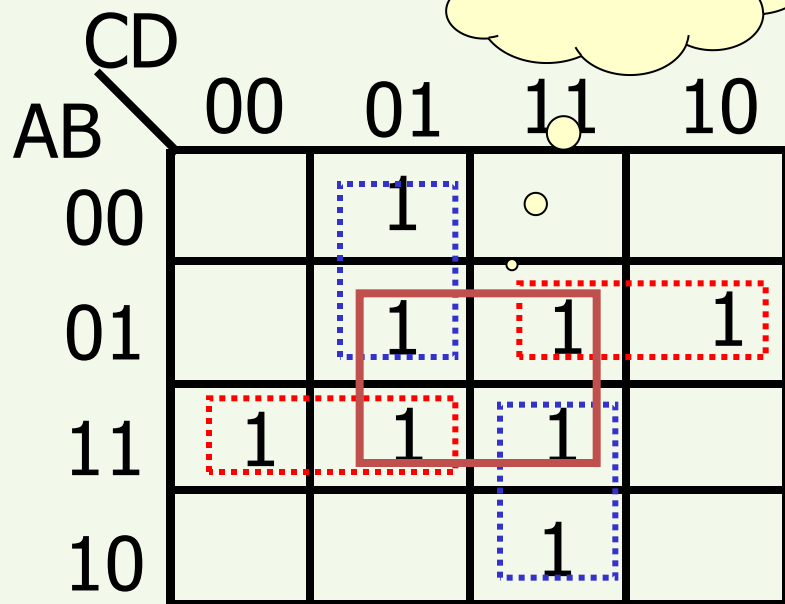
no



➤ 卡诺图化简应注意的问题三

(3) **不能遗漏**（即每个“1”至少被圈过1次），

但圈不能**重复**（即每个圈至少应包含一个新的“1”）



➤卡诺图化简应注意的问题四



(4) 同一个逻辑函数可以画出不同的圈，但是圈的个数必须相同，繁简程度也相同。

AB \ CD				
	00	01	11	10
00		1	1	
01			1	1
11				1
10				

$$F = \overline{A}\overline{B}D + \overline{A}BC + BCD$$

AB \ CD				
	00	01	11	10
00		1	1	
01			1	1
11				1
10				

$$F = \overline{A}\overline{B}D + \overline{A}CD + BCD$$

结论：同一逻辑函数式的化简结果可能不唯一。



课堂练习
第四节1, 2, 5, 6,
7, 9





课程小结

1、逻辑函数的化简

- 逻辑代数的公式与定理
 - 基本公式
 - 若干常用的逻辑公式
- 逻辑函数的公式化简法
- 逻辑函数的卡诺图化简法
 - 卡诺图的构成与特点
 - 卡诺图表示逻辑函数的方法
 - 利用卡诺图化简逻辑函数



➤ 约束项的概念

在逻辑函数中不可能出现或不允许出现的输入变量组合。

例：对8421BCD码而言，有效输入组合为0000~1001，
其余六种输入组合1010~1111不会出现，即为约束项

电梯运行状态指示电路

有A、B、C三个开关分别控制电梯的上行、下行与停止，设取“1”时有效。F表示电梯的运行状态，取“1”时表示电梯在运行中，取“0”时表示电梯停止。

真值表

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

约束项



◆ 具有约束项的逻辑函数的表示方法：

真值表：

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

逻辑表达式：

$$\left\{ \begin{array}{l} F = \overline{A}B\overline{C} + A\overline{B}\overline{C} \\ \text{约束条件:} \\ \overline{A}\overline{B}\overline{C} + \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = 0 \end{array} \right.$$

卡诺图：

		BC			
		00	01	11	10
A	0	×		×	1
	1	1	×	×	×



➤ 约束项在卡诺图化简中的应用

方法：对具有约束项的逻辑函数进行化简时，可利用约束项，即**视需要**把一些约束项当作“1”，另一些当作“0”。

以得到的圈“**最大**”，
而且圈数目“**最少**”
为原则

A \ BC	00	01	11	10
0	x		x	1
1	1	x	x	x

圈内的约束项被当作“1”

未被圈起的约束项
被当作“0”

⇒
$$\left\{ \begin{array}{l} F = \bar{C} \\ \text{约束条件:} \\ \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + ABC = 0 \end{array} \right.$$



例：化简具有约束项的逻辑函数：

$$\begin{cases} F = C\bar{D}(A \oplus B) + \bar{A}B\bar{C} + \bar{A}\bar{C}D \\ \text{约束条件: } AB + CD + A\bar{C}\bar{D} = 0 \end{cases}$$

解：(1) 将F化为“与或”式，即

$$\begin{aligned} F &= C\bar{D}(\bar{A}B + A\bar{B}) + \bar{A}B\bar{C} + \bar{A}\bar{C}D \\ &= \bar{A}BC\bar{D} + A\bar{B}C\bar{D} + \bar{A}B\bar{C} + \bar{A}\bar{C}D \end{aligned}$$

(2) 由卡诺图化简

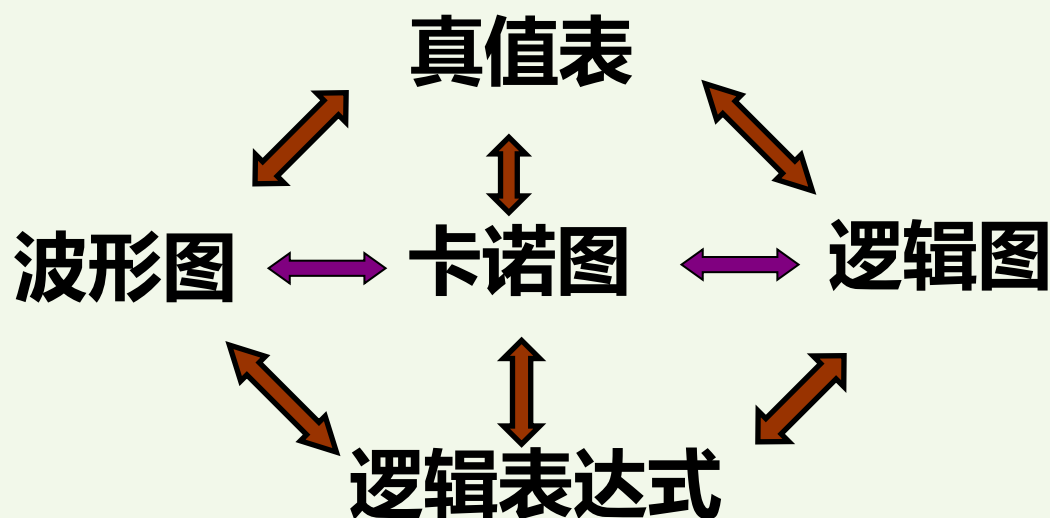


AB \ CD		CD			
		00	01	11	10
AB	00		1	×	
	01	1	1	×	1
	11	×	×	×	×
	10	×		×	1

$$\begin{cases} F = B + \bar{A}D + AC \\ \text{约束条件: } AB + CD + A\bar{C}\bar{D} = 0 \end{cases}$$



8.6 逻辑函数不同表达形式之间的相互转换



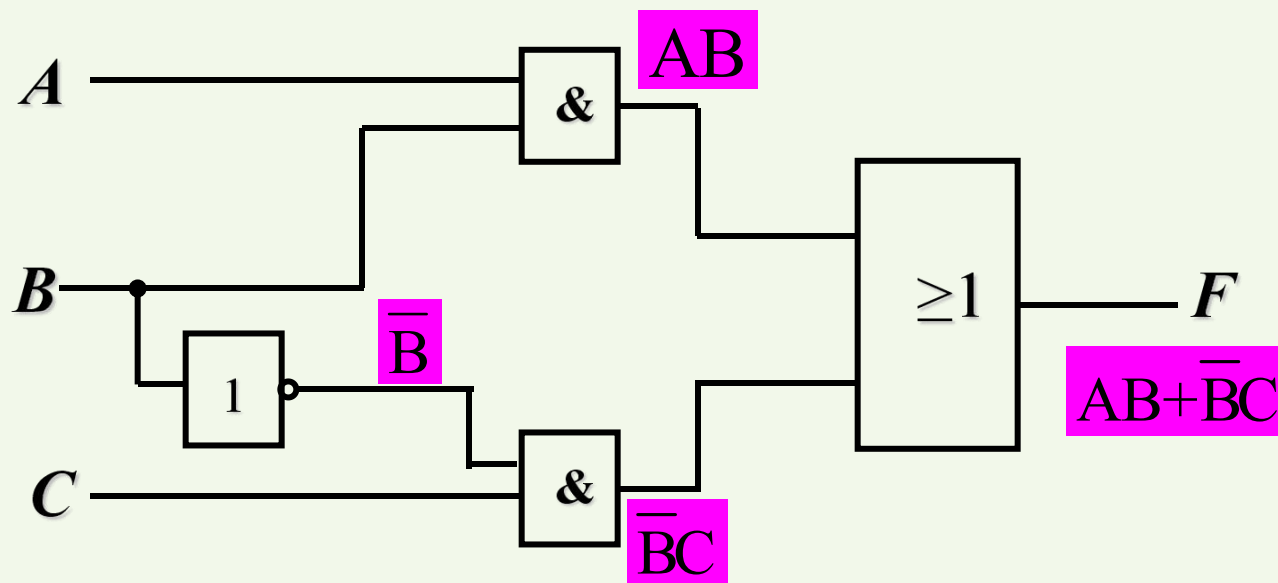
❖ 同一个逻辑函数的各种表达形式之间可以相互转换



◆ 逻辑表达式 \longrightarrow 逻辑图

将式中所有的与、或、非运算用逻辑符号代替，并根据运算优先顺序把这些符号连接起来

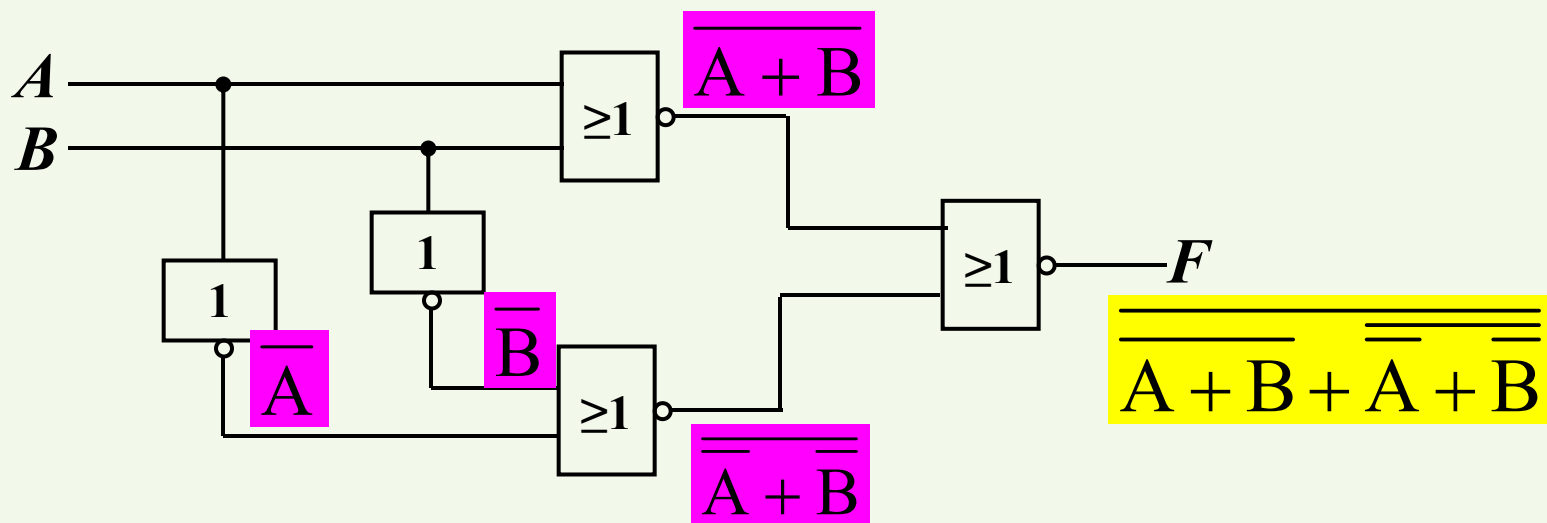
例：已知 $F = AB + \overline{B}C$ ，画出 F 的逻辑电路图





◆ 逻辑图 → 逻辑表达式

从输入端到输出端逐级写出各个逻辑符号对应的逻辑式。



$$F = \overline{A + B + \bar{A} + \bar{B}} = (A + B)(\bar{A} + \bar{B}) = \bar{A}B + A\bar{B}$$

\downarrow

$$\overline{A + B} = \bar{A}\bar{B} \qquad = A \oplus B$$



◆ 逻辑表达式 真值表

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

$$\left. \begin{array}{l} A=1 \\ B=0, C=1 \end{array} \right\} \Rightarrow F=1$$

A	B	C	F
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



◆ 真值表 逻辑表达式

	A	B	F
$\overline{A}\overline{B}$	0	0	0
$\overline{A}B$	0	1	1
$A\overline{B}$	1	0	1
AB	1	1	1

➤ 每组输入变量对应一个乘积项（最小项），其中取值为**1**的写入**原变量**，为**0**的写入**反变量**；

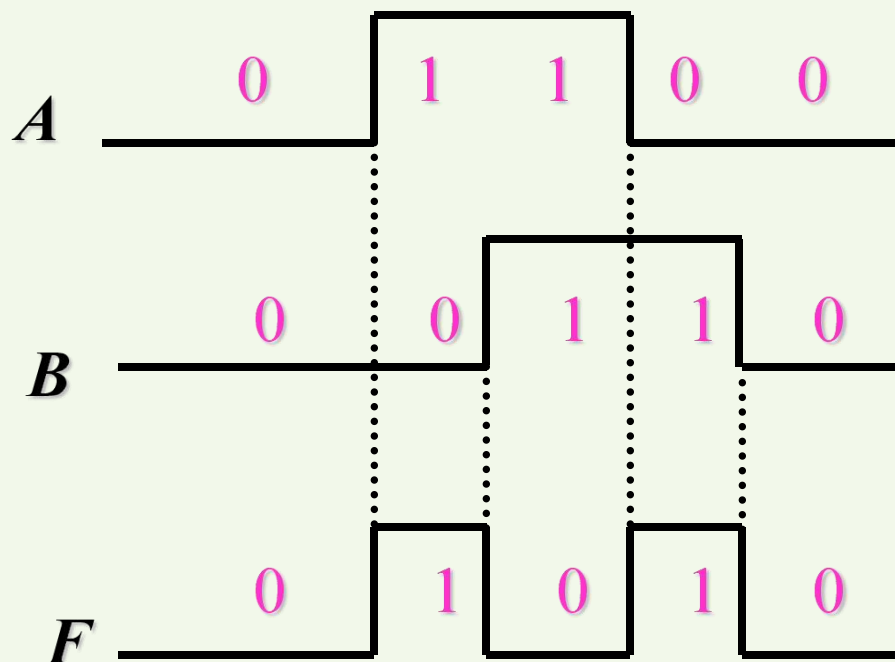
➤ 找出真值表中使逻辑函数 **$F=1$** 的那些输入变量组合对应的最小项；

➤ 将这些最小项**相加**，即得 **F** 的表达式

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



◆ 时序图 → 逻辑图



❖ 由时序图列出真值表

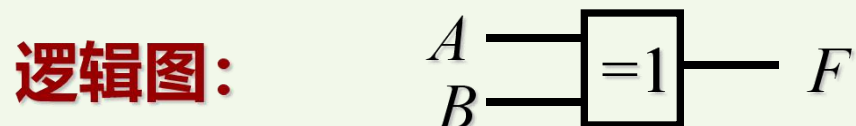
❖ 写出逻辑函数式

❖ 画逻辑图

真值表:

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

逻辑表达式: $F = \overline{A}B + A\overline{B} = A \oplus B$



例1：分别用“与非”门和“或非”门实现逻辑函数

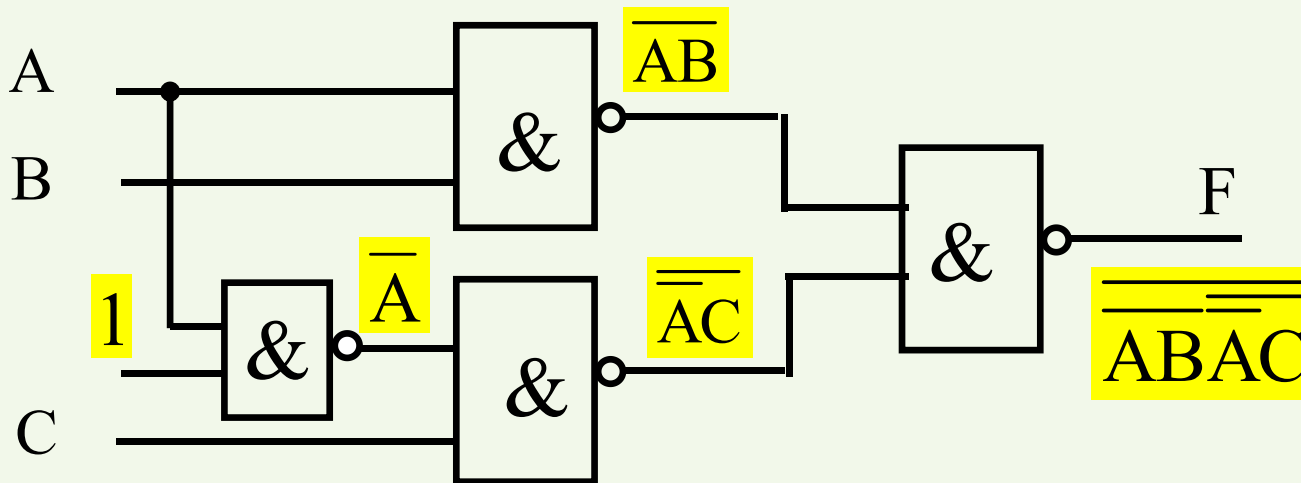
$$F = AB + \overline{A}C$$

解：（1）用“与非”门实现逻辑函数

由“与或”式写出“与非-与非”式

摩根定律

$$F = AB + \overline{A}C = \overline{\overline{AB + \overline{A}C}} = \overline{\overline{AB} \cdot \overline{\overline{A}C}} = \overline{\overline{AB} \cdot AC}$$



(2) 用“或非”门实现逻辑函数 $F = AB + \bar{A}C$

由“与或”式求
“或与”式

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

反演规则

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

冗余率

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

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

反演规则

由“或与”式求
“或非-或非”式

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

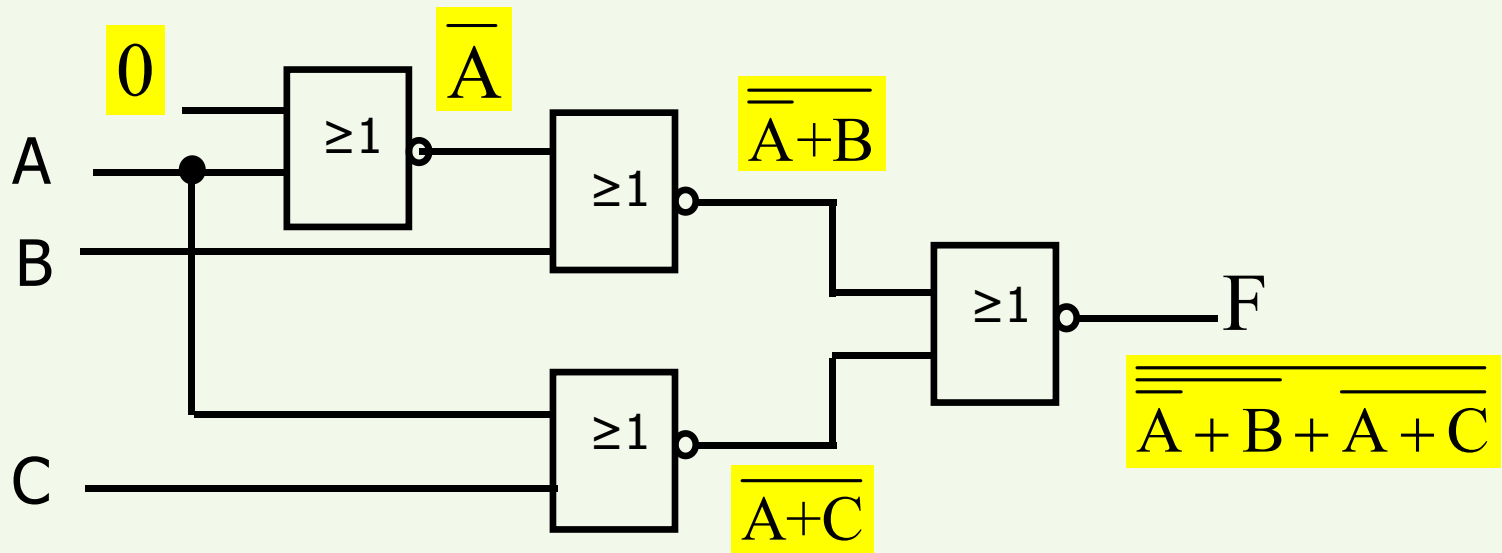
$$= \overline{\overline{(\bar{A} + B)(A + C)}}$$

摩根定律

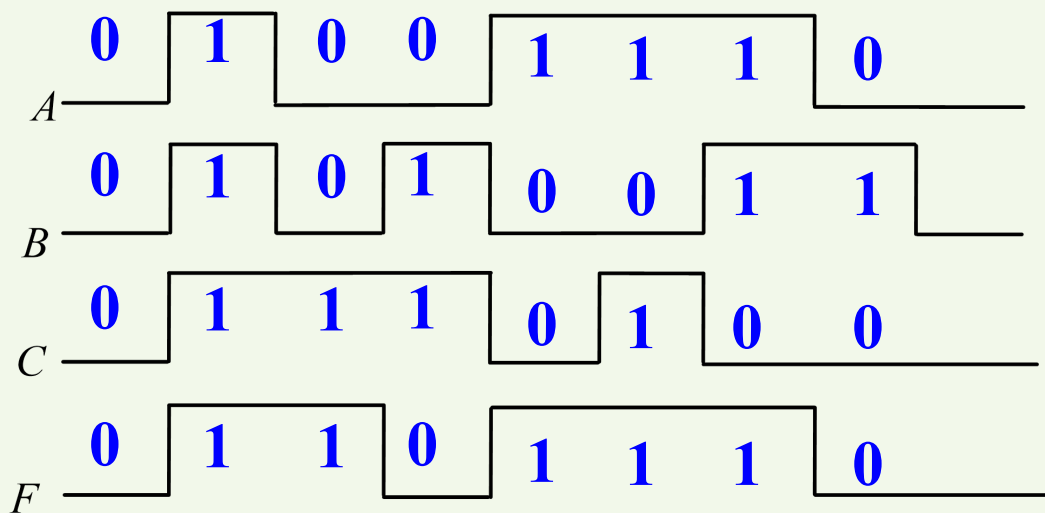
$$= \overline{\bar{A} + B} \overline{A + C}$$



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



例2：逻辑电路输入A、B、C波形与输出F波形如图所示，试分别列出真值表、写出函数式，并用“与非”门实现。



A	B	C	F
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

解：（1）列真值表
（2）写表达式

标准与或式

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

$$F(A, B, C) = m_1 + m_4 + m_5 + m_6 + m_7$$



(3) 用“与非”门实现

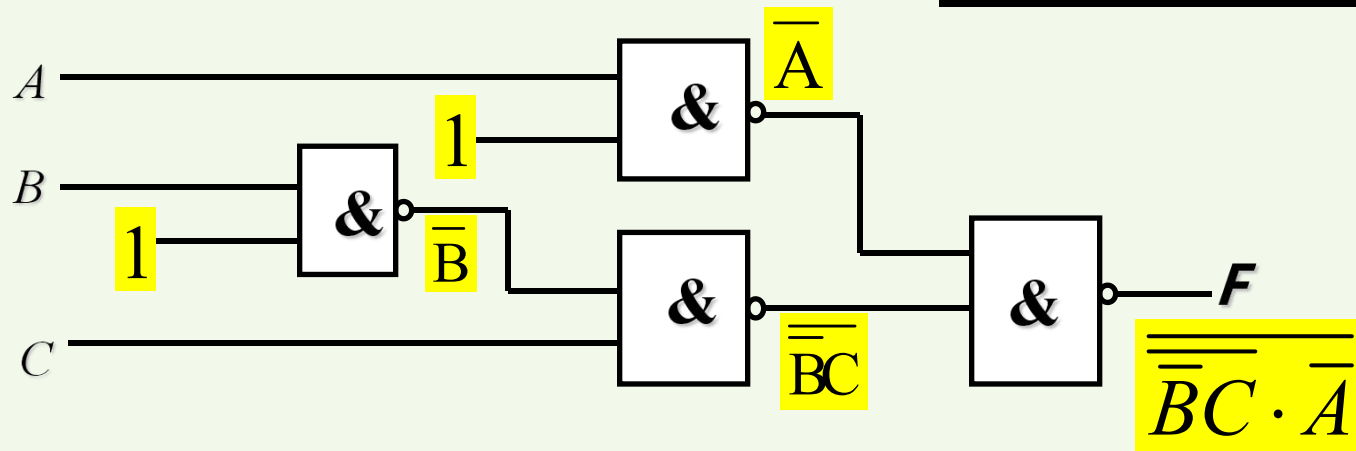
$$F = \overline{A}\overline{B}C + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC$$

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

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

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

BC		00	01	11	10
A	0		1		
	1	1	1	1	1



例3: (1) 写出图(a)所示电路的逻辑表达式, 化简成最简“与或”式; (2) 已知A、B、C波形如图(b)所示, 试画出函数F的波形。

解: (1) 写表达式

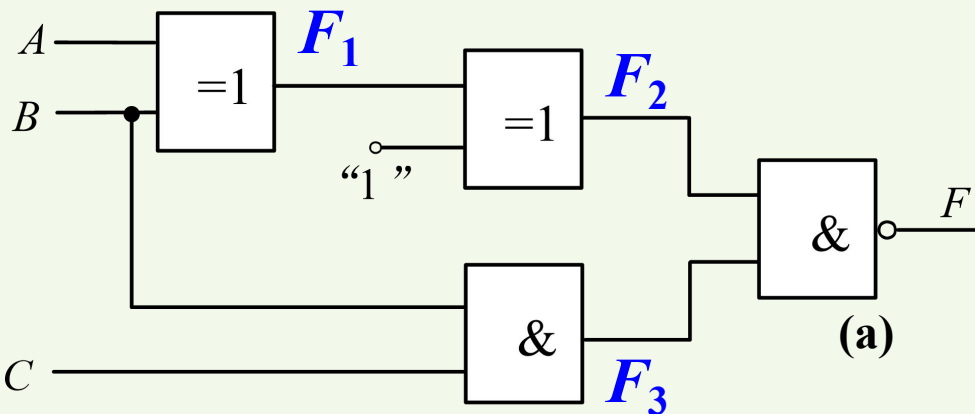
$$F_1 = A \oplus B$$

$$F_2 = F_1 \oplus 1 = \overline{F_1} = \overline{A \oplus B}$$

$$F_3 = BC$$

$$F = \overline{F_2 F_3} = \overline{\overline{A \oplus B} \bullet BC} = A \oplus B + \overline{BC}$$

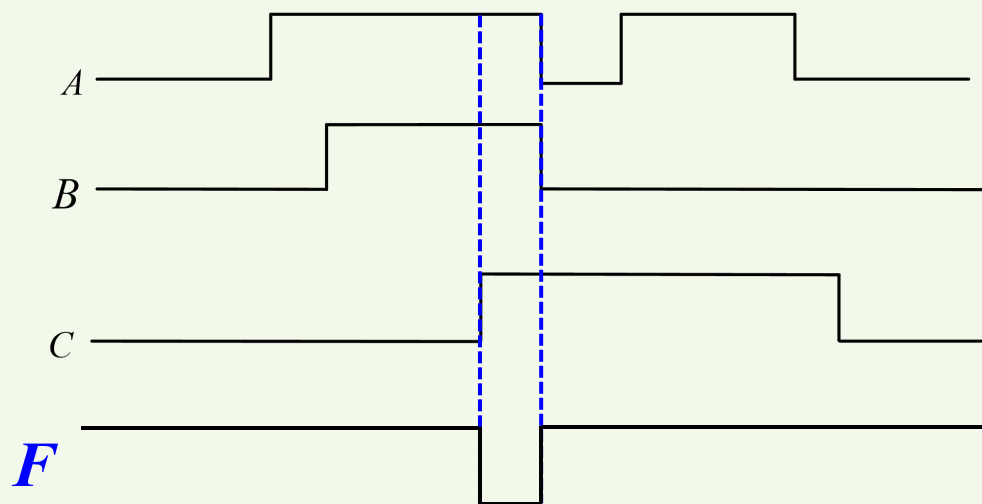
$$= \overline{A}B + A\overline{B} + \overline{B} + \overline{C} = \overline{A} + \overline{B} + \overline{C}$$





(2) 画波形

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



(b)



课堂练习
第五节 2, 4
第六节 2, 3, 4





课程小结

1、具有约束项的逻辑函数

- 约束项的概念
- 具有约束项逻辑函数的化简

2、逻辑函数不同表达式形式之间的相互转换