
第3章 组合逻辑电路

第一讲 组合逻辑电路概述

第二讲 典型组合逻辑部件设计

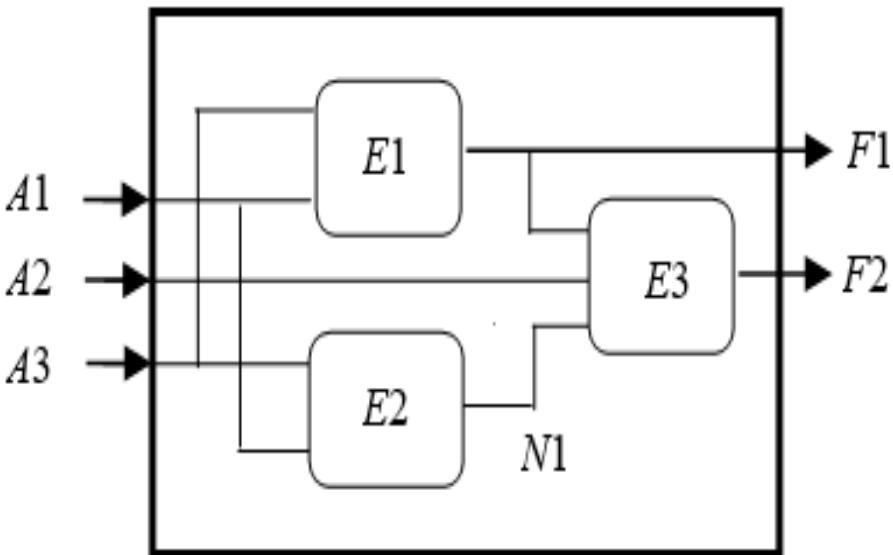
第三讲 组合逻辑电路时序分析

第一讲 组合逻辑电路概述

- ◆ 组合逻辑电路构成规则
- ◆ 逻辑电路图
- ◆ 两级与多级组合逻辑电路
- ◆ 组合逻辑电路设计
- ◆ 无关项、非法值和高阻态

1 组合逻辑电路概述

- ◆ 数字逻辑电路可被看成是带有若干输入端和若干输出端的黑盒子，每个输入端和输出端只有高电平、低电平两种状态，对应1或0。
- ◆ 分为组合 (combinational) 逻辑电路和时序 (sequential) 逻辑电路两种类型。
 - 组合逻辑电路的输出值仅依赖于当前输入值 (本章)
 - 时序逻辑电路的输出值不仅依赖输入值，还与当前状态（现态）有关。电路中存在存储部件或反馈结构 (第4章)

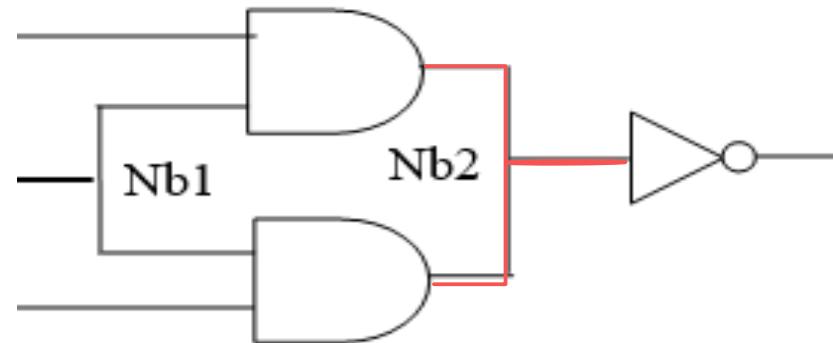


- ◆ 黑盒内部可被看成由若干元件和若干结点互连而成。
 - 元件本身又可以是一个数字逻辑电路
 - 结点可以是输入结点(如 A_1)、内部结点(如 N_1)和输出结点(如 F_1)

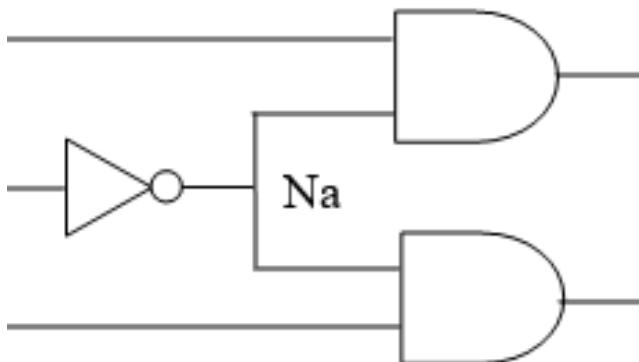
1.1 组合逻辑电路构成规则

- ◆ 最简单的组合逻辑电路是逻辑门电路，用于实现基本逻辑运算
- ◆ 组合逻辑电路构成规则

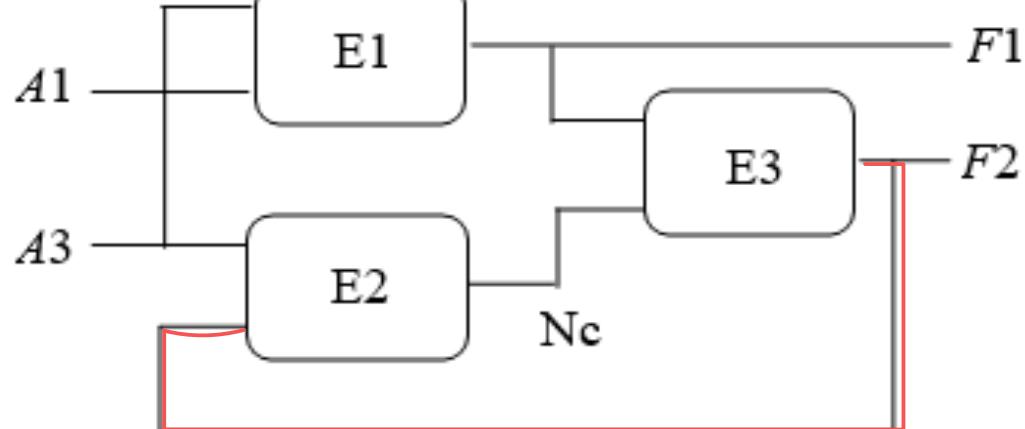
- 每个元件本身是组合逻辑电路
- 输出结点不能互连
- 输出结点不能反馈到输入端



不是组合逻辑电路



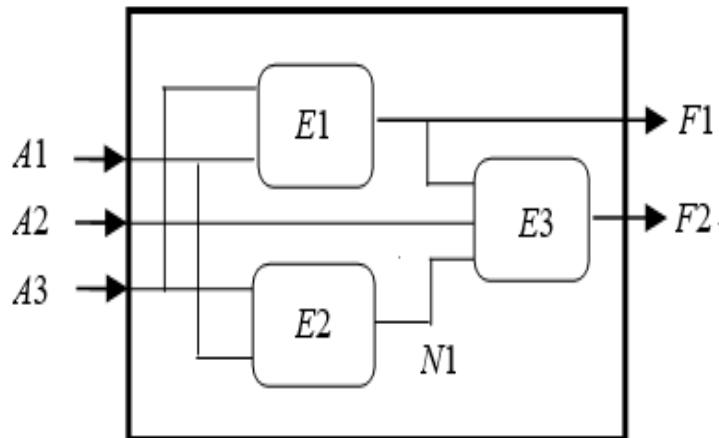
是组合逻辑电路



不是组合逻辑电路

1.2 逻辑电路图

- ◆ 逻辑电路图描述数字电路内部元件的结构及其相互连接关系
- ◆ 每个逻辑电路图对应一个逻辑表达式（多个输入，一个输出）



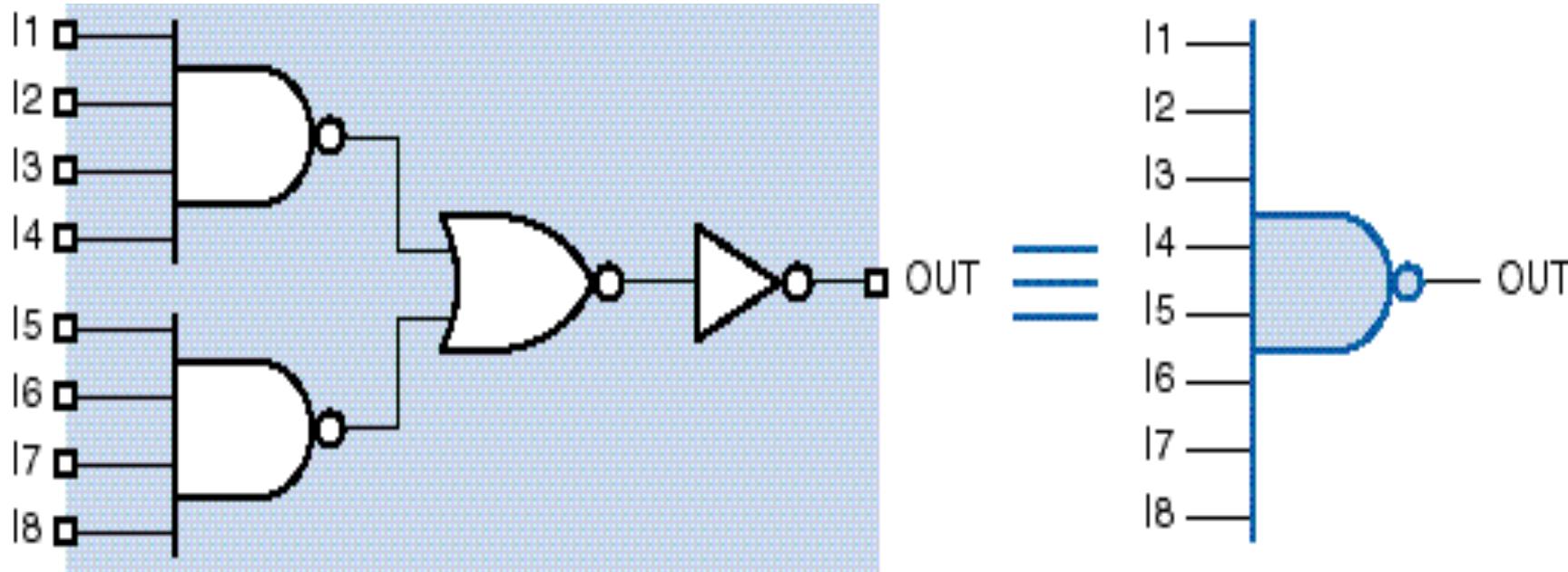
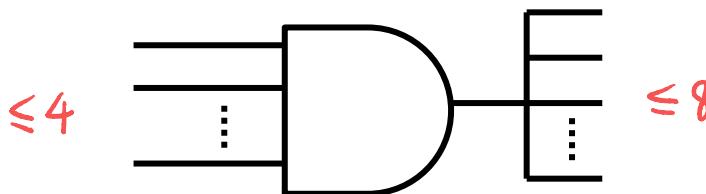
可以完成复杂功能的组合逻辑电路可以有多个输出——

而其中每一个输出端对应一个逻辑表达式（一个逻辑电路图），不同输出端的逻辑表达式一般不同，也可以相同。

- ◆ 一个真值表可能对应多个不同的逻辑表达式，从而对应多个不同的逻辑电路图，因而可以有多个不同的实现方式（功能相同，但电路所用的元器件不同、化简程度不同。。。。）
- ◆ 任何逻辑表达式都可写成与、或、非三种基本运算的逻辑组合
- ◆ 一个逻辑门的输出可作为另一个逻辑门的输入

1.2 逻辑电路图

- ◆ **扇入系数**: 一个逻辑门所允许的输入端的最大数目
- ◆ **扇出系数**: 一个逻辑门输出端信号所能驱动的下一级输入端的最大数目



1.2 逻辑电路图

- ◆画逻辑电路图时，须依据逻辑运算的**优先级**确定逻辑门间的**连接关系**
 - 优先级高的运算对应的逻辑门的输出，是优先级低的运算对应逻辑门的输入
 - 优先级顺序如下：**非 > 与和与非 > 异或和同或 > 或和或非**
- ◆插播一个小回顾：异或、同或的表达式

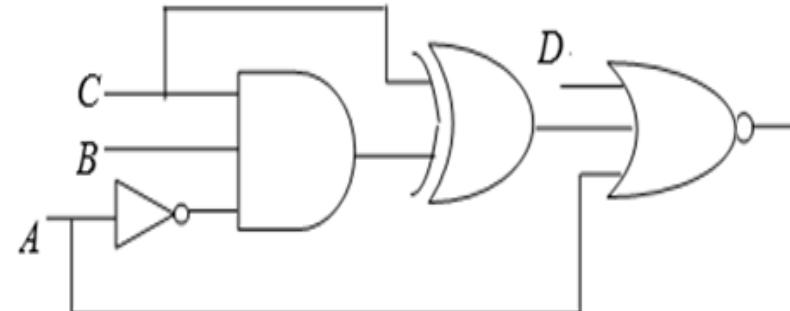
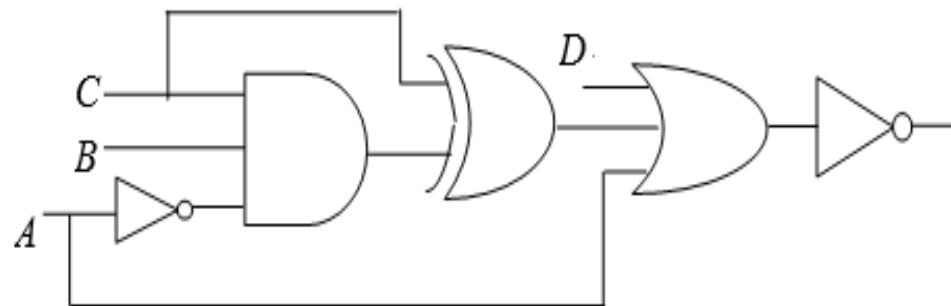
$$X \oplus Y = \overline{X} \cdot Y + X \cdot \overline{Y}$$

$$X \odot Y = \overline{X} \cdot \overline{Y} + X \cdot Y$$

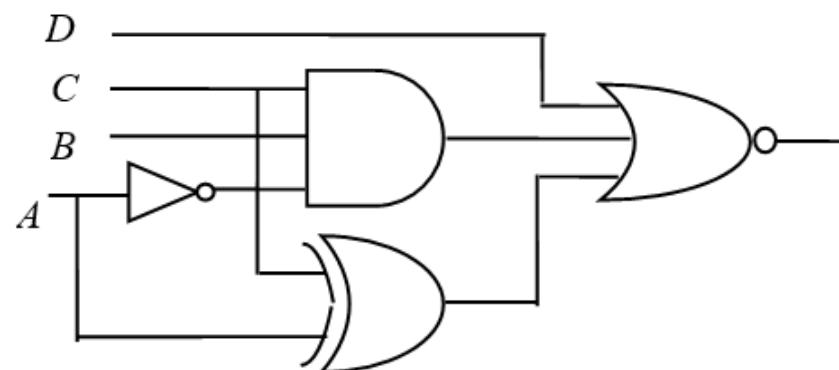
X, Y	异或
0, 0	0
0, 1	1
1, 0	1
1, 1	0

1.2 逻辑电路图

例：画出 $\overline{A \cdot B \cdot C} + C + A + D$ 对应的逻辑电路图



$\overline{A \cdot B \cdot C} + C + A + D$?



1.2 逻辑电路图

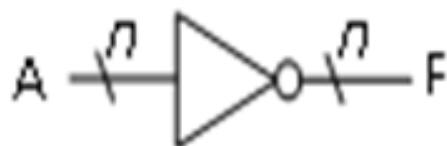
- n位逻辑运算在输入端和输出端标注位数即可



$$F = A \cdot B$$



$$F = A + B$$



$$F = \bar{A}$$



$$F = A \oplus B$$

1.3 两级和多级组合逻辑电路

- ◆ 信号通过逻辑门时在时间上存在延迟。从输入信号改变开始，到输出信号发生改变所用的时间称为**门延迟 (gate delay)** 门的传播延迟
- ◆ 任何逻辑表达式都可以转换成与-或表达式和或-与表达式，因此，任何组合逻辑电路都可以是一个**两级电路**
- ◆ 与-或表达式对应电路：第一级是若干个与门，第二级是一个或门

1.3 两级和多级组合逻辑电路

- ◆ 信号通过逻辑门时在时间上存在延迟。从输入信号改变开始，到输出信号发生改变所用的时间称为**门延迟 (gate delay)**
- ◆ 任何逻辑表达式都可以转换成**与-或表达式**和**或-与表达式**，因此，任何组合逻辑电路都可以是一个**两级电路**
- ◆ **与-或表达式对应电路：第一级是若干个与门，第二级是一个或门**

例： $\overline{A} \cdot B \cdot C \oplus C + A + D$ 可转换为 $\overline{A} \cdot B \cdot \overline{D} + \overline{A} \cdot \overline{C} \cdot \overline{D}$

转换前：最长路径从A输入端到输出经过了非门、与门、异或门、或门和非门；**转换后：**最长路径只经过非门、与门和或门

设与、或门延迟时间都是2ns，异或门延迟为3ns，不考虑非门延迟，则转换前传输时间为7ns，转换后为4ns

1.3 两级和多级组合逻辑电路

- ◆ 信号通过逻辑门时在时间上存在延迟。从输入信号改变开始，到输出信号发生改变所用的时间称为**门延迟 (gate delay)**
- ◆ 任何逻辑表达式都可以转换成**与-或表达式**和**或-与表达式**，因此，任何组合逻辑电路都可以是一个**两级电路**
- ◆ **与-或表达式对应电路：**第一级是若干个与门，第二级是一个或门

例： $\overline{A} \cdot B \cdot C \oplus C + A + D$ 可转换为 $\overline{A} \cdot B \cdot \overline{D} + \overline{A} \cdot \overline{C} \cdot \overline{D}$

转换前：最长路径从A输入端到输出经过了非门、与门、异或门、或门和非门；**转换后：**最

设与、或门延迟时间为7ns，
转换前传输时间为7ns × 4 = 28ns

- ◆ **两级组合逻辑电路好处：**使用两级组合电

如 $F = A \oplus B \oplus C = A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + A \cdot B \cdot C$

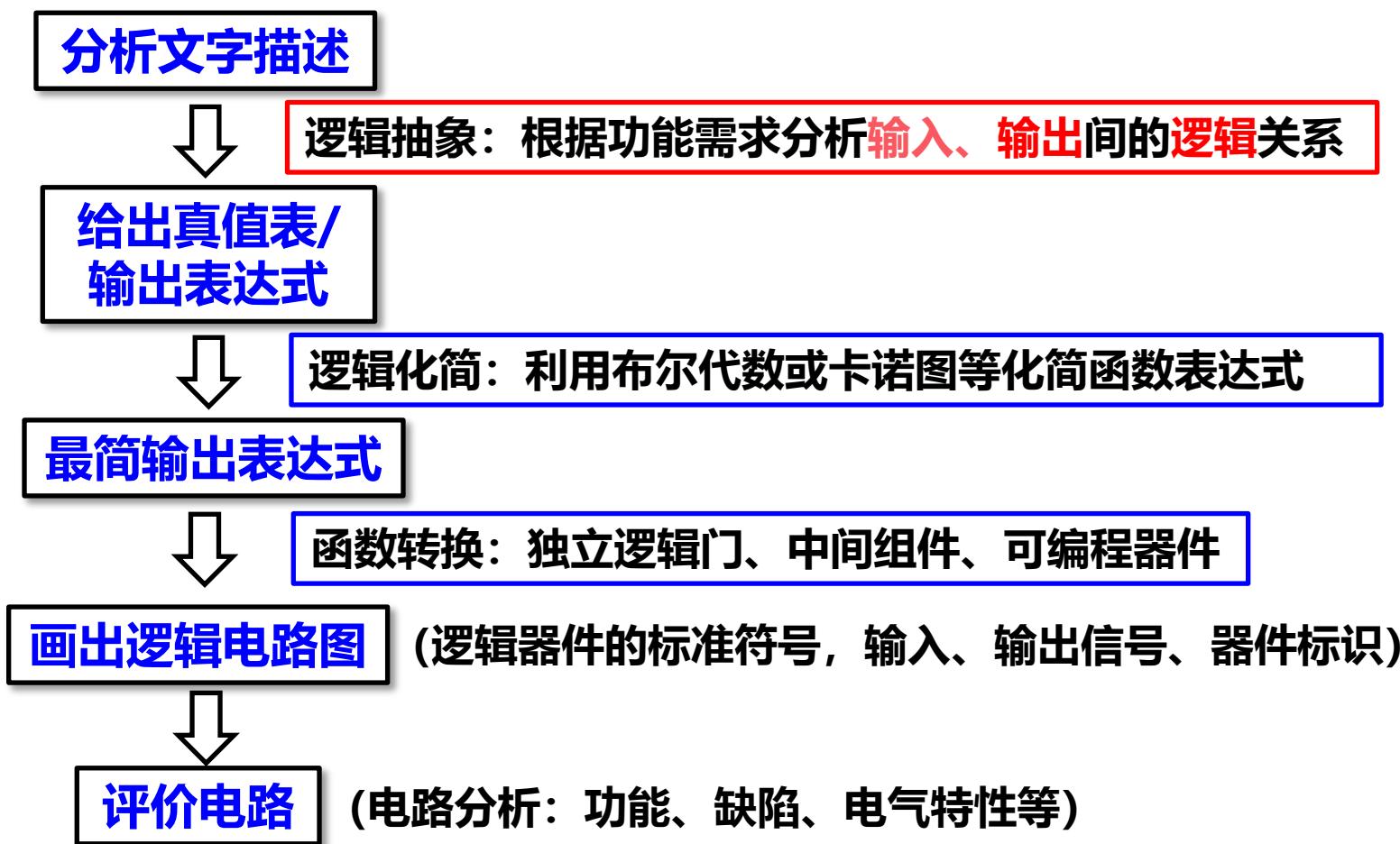
128个8输入端与门和一个128输入端或门！7个2输入端异或门

- ◆ 采用两级还是多级需要在**速度和成本**之间进行权衡

最终，任意两级电路还都可以转换为等价的、用与非门（或非门）构成的电路，则转换后的速度更快；

1.4 组合逻辑电路设计

从文字描述到逻辑电路或系统设计的整个过程如下：

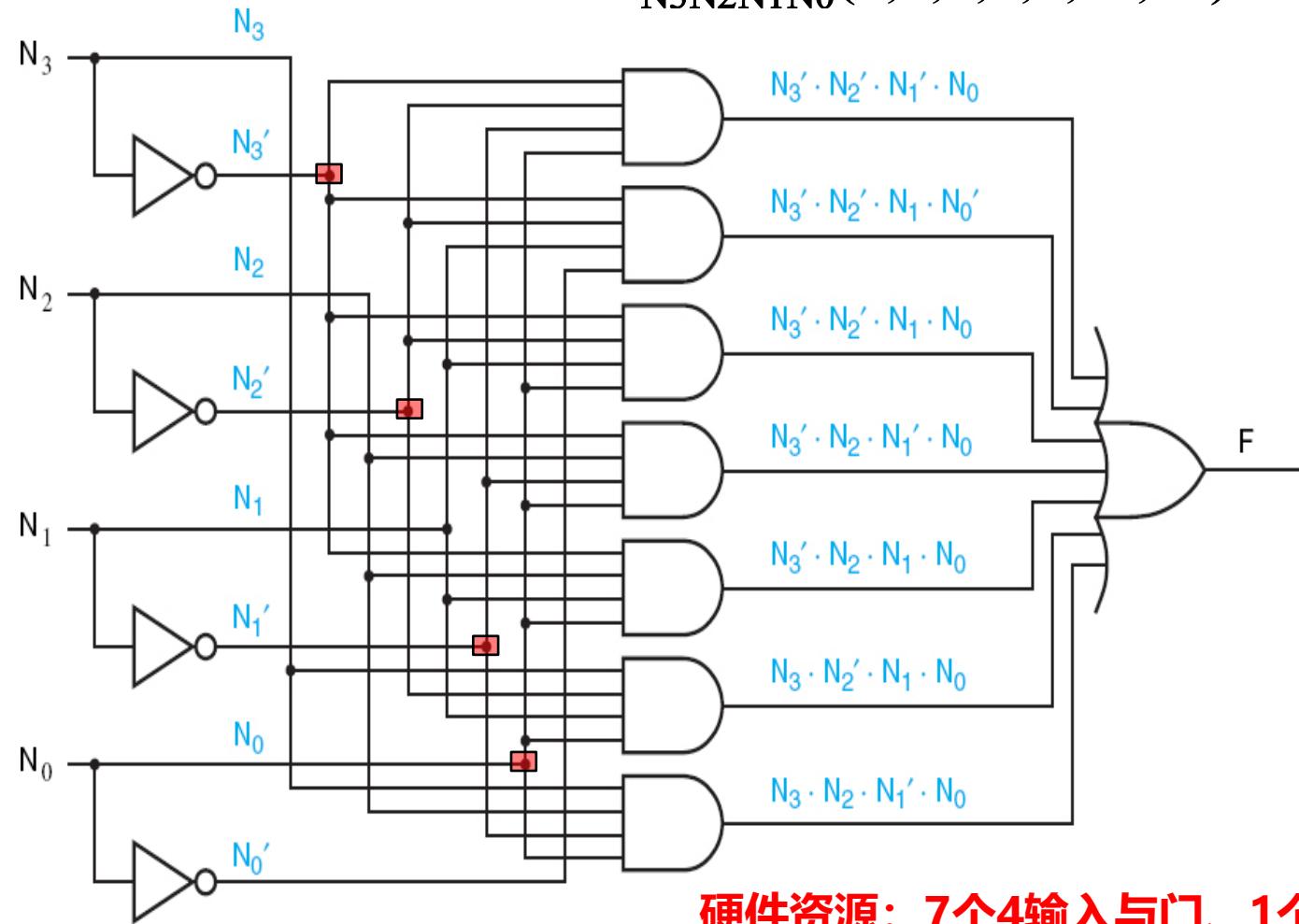


1.4 组合逻辑电路设计（例子1）

例1：素数检测器的设计

- 4位输入, $N_3 N_2 N_1 N_0$

写出最小项表达式 $F = \sum_{N_3 N_2 N_1 N_0} (1, 2, 3, 5, 7, 11, 13)$



列出真值表

row	N_3	N_2	N_1	N_0	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

硬件资源：7个4输入与门、1个7输入或门

1.4 组合逻辑电路设计 (例子1)

利用布尔代数化简, 以减少逻辑门数和输入端数

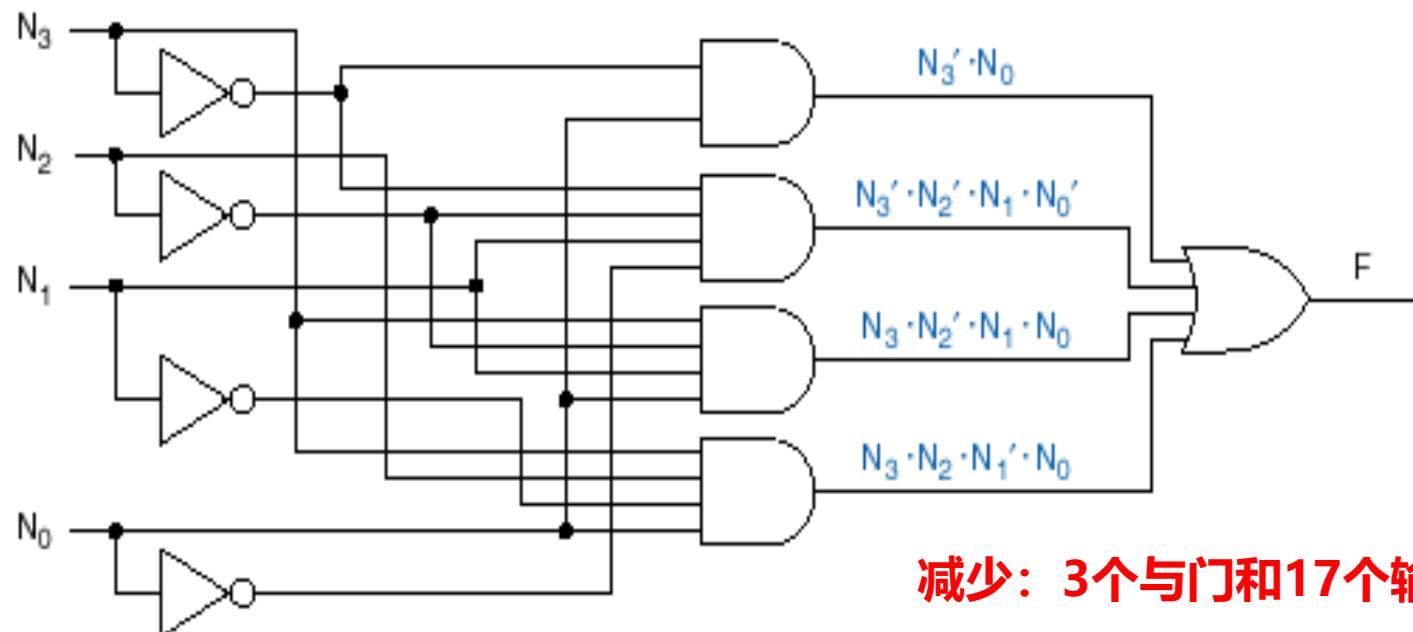
$$X \cdot Y + X \cdot Y' = X$$

$$F = \Sigma_{N_3 N_2 N_1 N_0} (1, 2, 3, 5, 7, 11, 13)$$

$$= N_3' \cdot N_2' \underline{N_1' N_0} + N_3' \cdot N_2' \cdot \underline{N_1 N_0} + N_3' \cdot N_2 \cdot \underline{N_1' N_0} + N_3' \cdot N_2 \cdot \underline{N_1 N_0} + \dots$$

$$= N_3' \underline{N_2' N_0} + N_3' \cdot \underline{N_2 N_0} + \dots$$

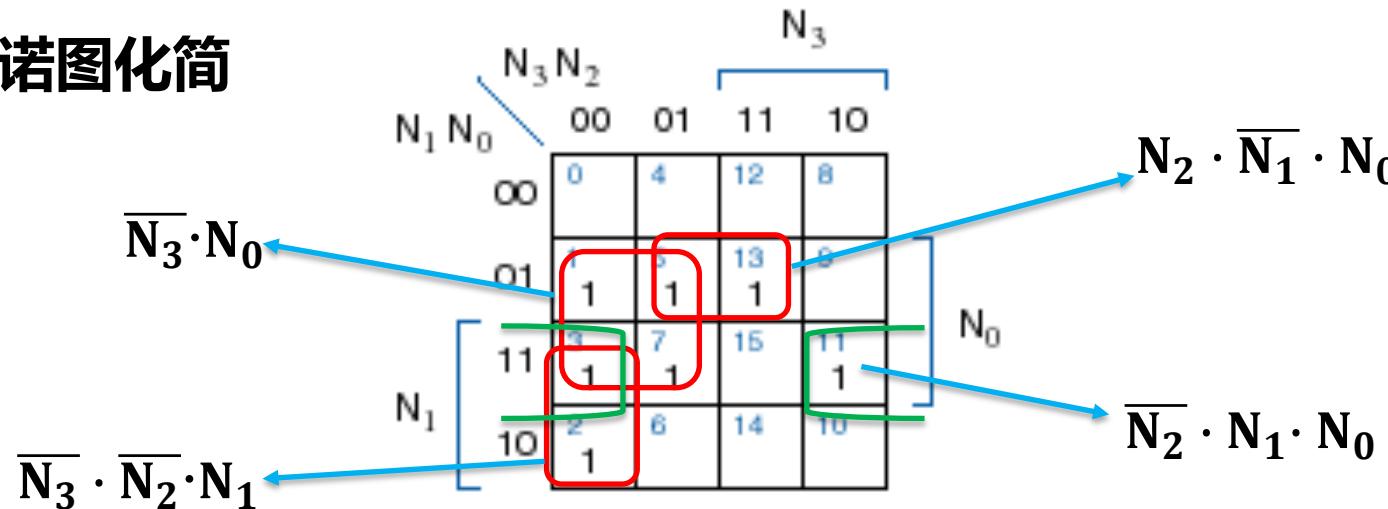
$$= N_3' \cdot N_0 + N_3' \cdot N_2' \cdot N_1 \cdot N_0' + N_3 \cdot N_2' \cdot N_1 \cdot N_0 + N_3 \cdot N_2 \cdot N_1' \cdot N_0$$



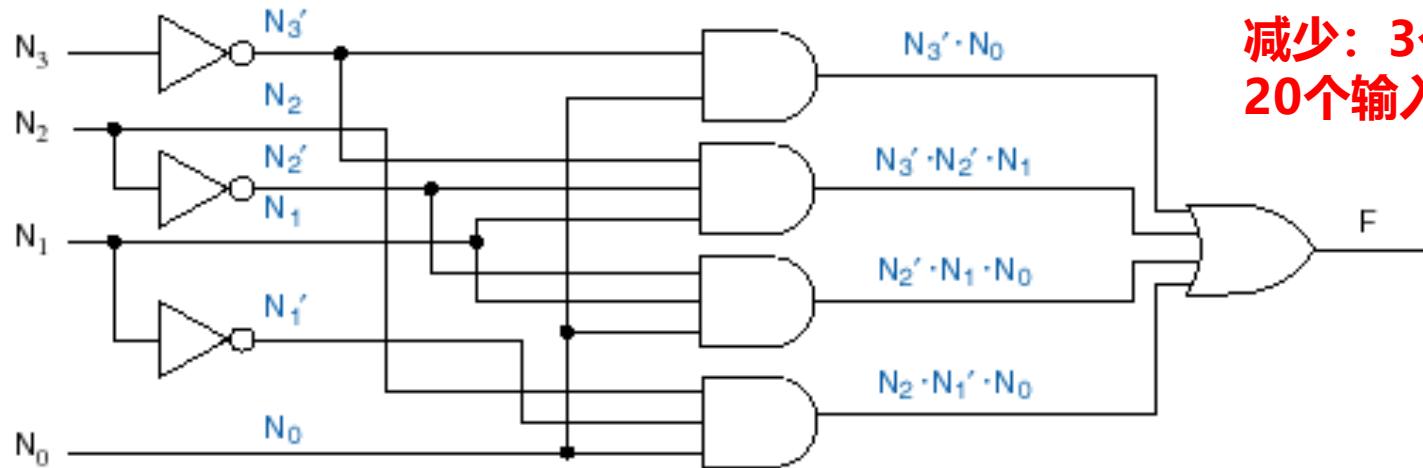
减少: 3个与门和17个输入端

1.4 组合逻辑电路设计（例子1）

按卡诺图化简



$$F = N_3' \cdot N_0 + N_3' \cdot N_2 \cdot N_1 + N_2' \cdot N_1 \cdot N_0 + N_2 \cdot N_1' \cdot N_0$$

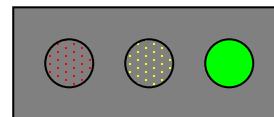
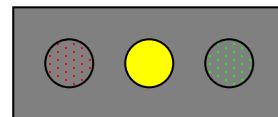
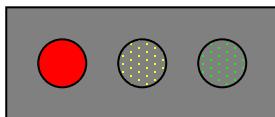


1.4 组合逻辑电路设计（例子2）

例2：设计一个监视交通信号灯工作状态的逻辑电路

1、进行逻辑抽象

正常工作状态



输入变量：红R 黄Y 绿G 三盏灯的状态

灯亮为1，不亮为0

输出变量：故障信号F

正常工作为0，发生故障为1

真值表				
故障状态	R	Y	G	F
A grey rectangular panel with three circular lights, all showing a dotted pattern.	0	0	0	1
A grey rectangular panel with three circular lights. The first light from the left is red and has a solid red dot pattern. The second and third lights are off, showing a dotted pattern.	0	0	1	0
A grey rectangular panel with three circular lights. The second light from the left is yellow and has a solid yellow dot pattern. The first and third lights are off, showing a dotted pattern.	0	1	0	0
A grey rectangular panel with three circular lights. The third light from the left is green and has a solid green dot pattern. The first and second lights are off, showing a dotted pattern.	0	1	1	1
A grey rectangular panel with three circular lights. The first light from the left is red and has a solid red dot pattern. The third light is off, showing a dotted pattern. The second light is yellow and has a solid yellow dot pattern.	1	0	0	0
A grey rectangular panel with three circular lights. The first light is off, showing a dotted pattern. The second light is yellow and has a solid yellow dot pattern. The third light is green and has a solid green dot pattern.	1	0	1	1
A grey rectangular panel with three circular lights, all showing a solid color pattern (red, yellow, green).	1	1	0	1
A grey rectangular panel with three circular lights, all showing a solid color pattern (red, yellow, green).	1	1	1	1

1.4 组合逻辑电路设计（例子2）

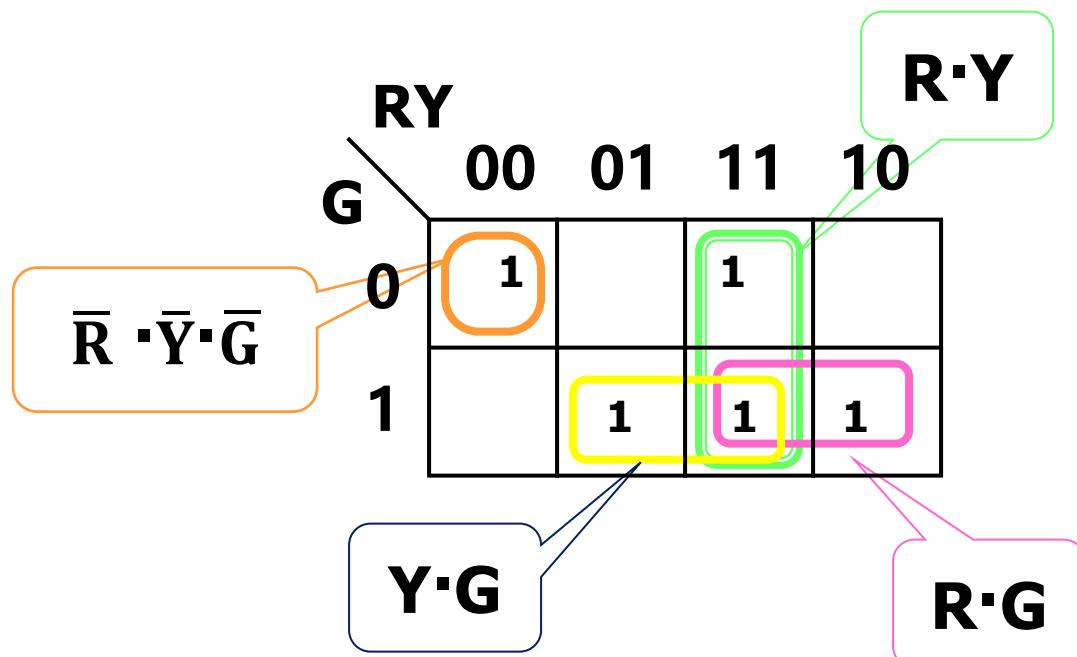
逻辑抽象结果

真 值 表

R	Y	G	F
0	0	0	1
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

2、逻辑化简

写出逻辑函数式并化简

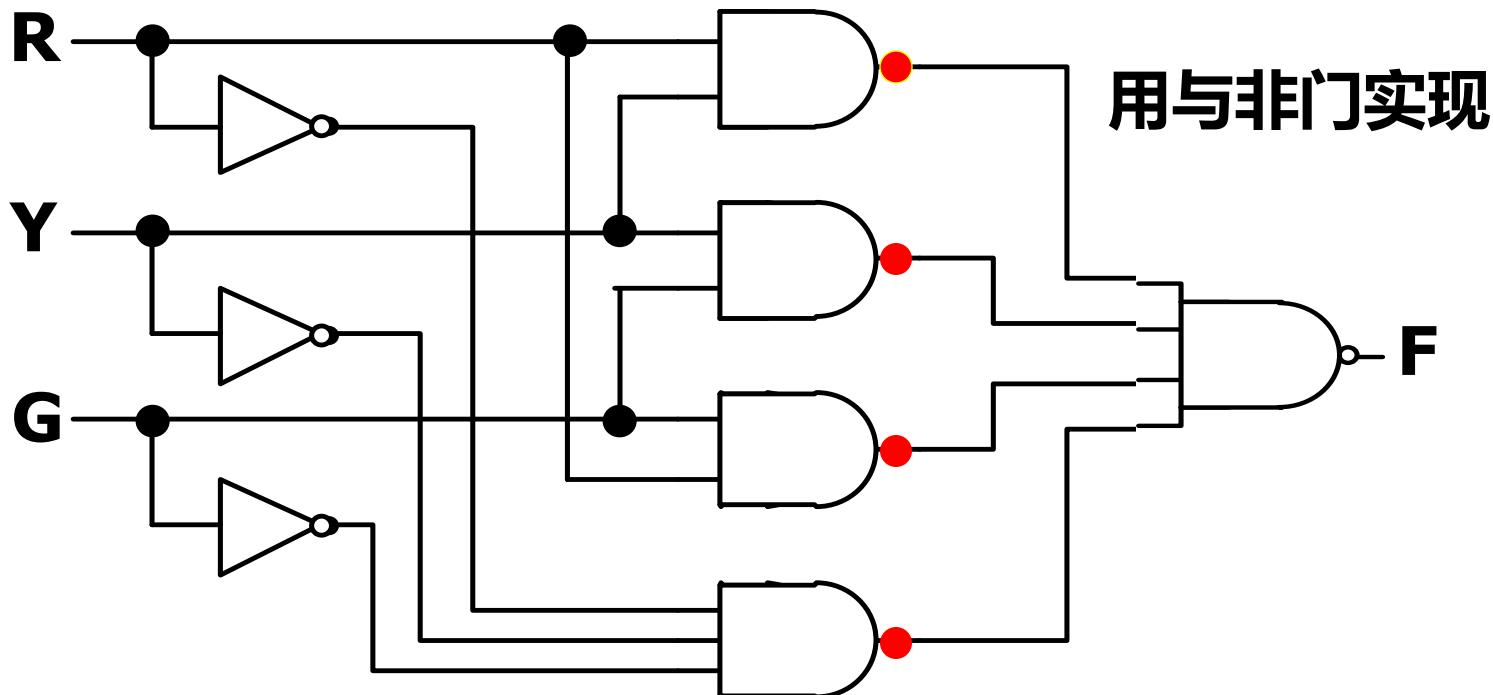


$$F = \bar{R} \cdot \bar{Y} \cdot \bar{G} + R \cdot Y + R \cdot G + Y \cdot G$$

1.4 组合逻辑电路设计（例子2）

3、电路设计

$$F = \bar{R} \cdot \bar{Y} \cdot \bar{G} + R \cdot Y + R \cdot G + Y \cdot G$$



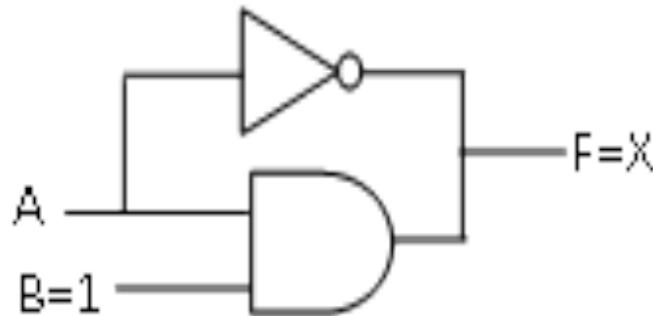
以上：正常工作为0，发生故障为1

如果：正常工作为1，发生故障为0？

1.5 无关项、非法值和高阻态

◆ 非法值

- 信号值不能被有效识别为高电平或低电平，处于不确定状态。
- 例如：下图（设计错误的电路）中的信号X



不管A是0还是1，F结点都会同时被高电平和低电平驱动

△ ◆ 无关项

- 某些输入组合对应的输出值可以是任意值，某些输入组合不可能出现。这些输入组合对应的输出值在化简时可标识为d，表示可以取值0或1，具体数值根据化简的需要而定

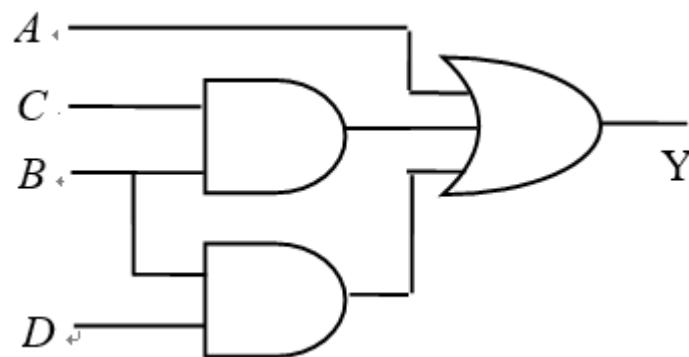
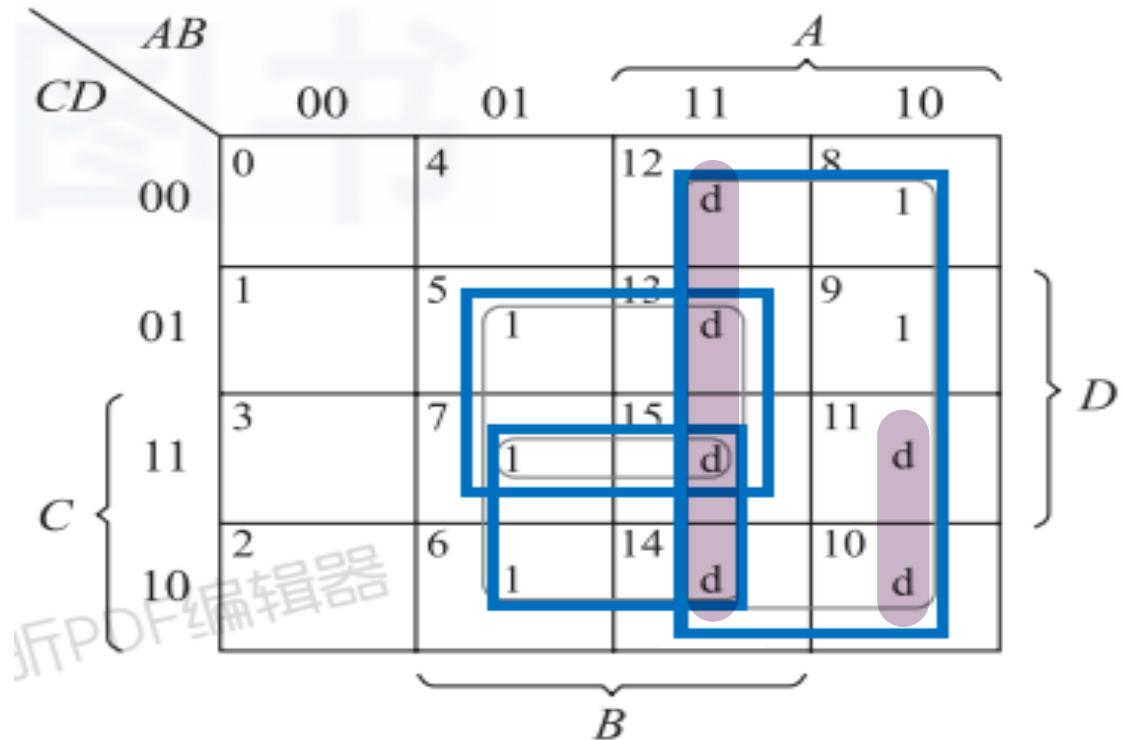
- 可简化电路从而降低成本，但也更易受干扰

例如：8421 自然BCD码输入时，大于1001的编码为无关项

1.5 无关项的使用

例：设计一个检测电路，当NBCD（8421）码数值大于等于5时，输出为1

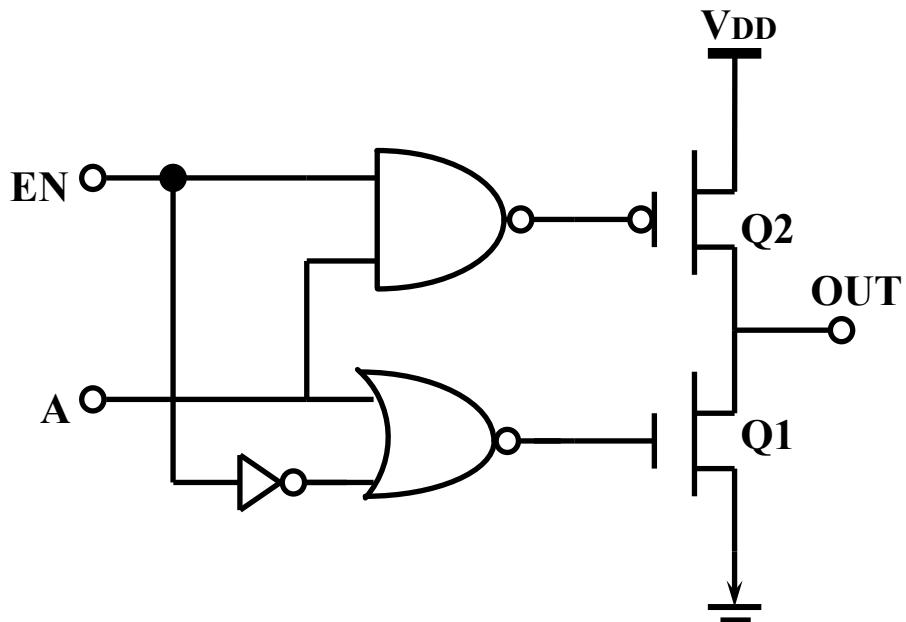
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	d
1	0	1	1	d
1	1	0	0	d
1	1	0	1	d
1	1	1	0	d
1	1	1	1	d



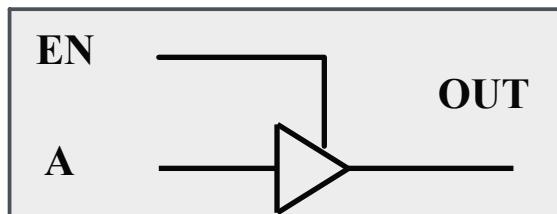
$$Y = A + B \bullet C + B \bullet D$$

1.5 高阻态、三态门

- ◆ **高阻态Hi-Z**: 输出处于非正常逻辑态的第三种电气态，好像和电路断开一样（也可视为电阻超高，无电流通过，对下一级电路无影响）
- ◆ **三态门 (three-state gate)** 是一种重要的总线接口电路，也称三态缓冲器，其输出既可以是通常的逻辑值1或0，又可以是高阻态
- ◆ 三态门有一个额外的输出**使能控制端EN**

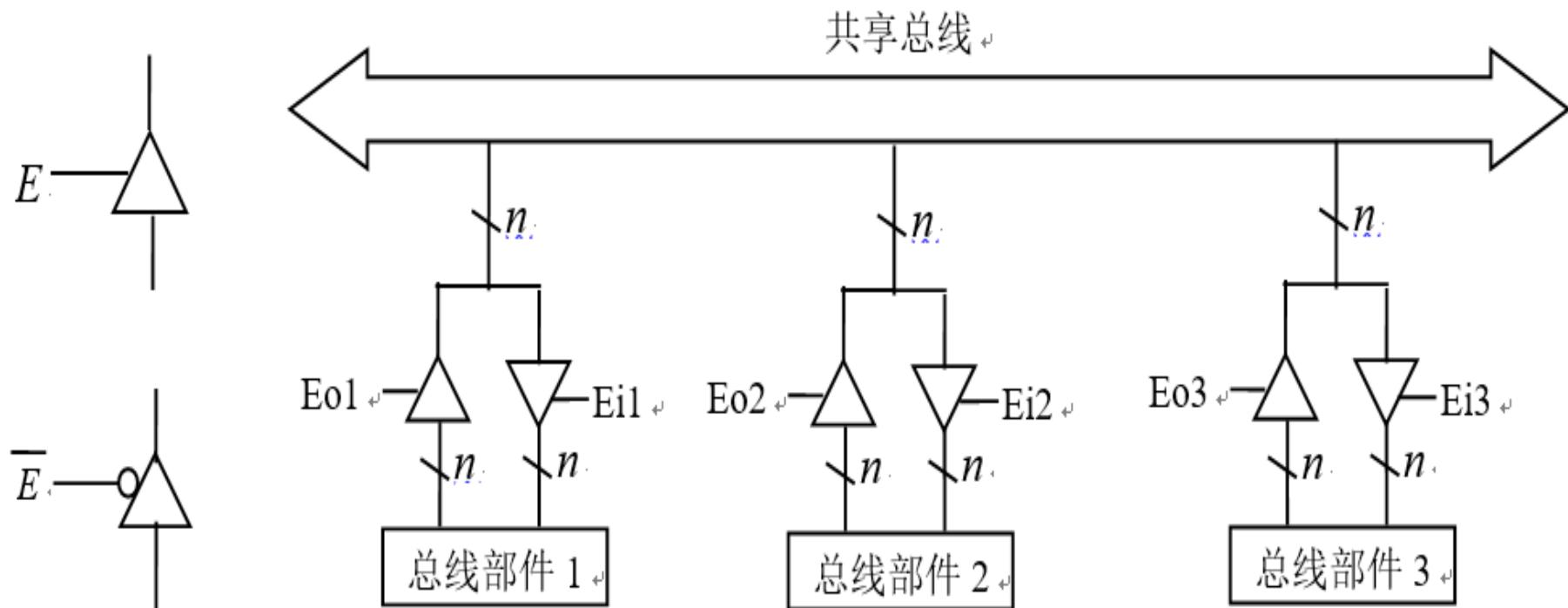


EN	A	Q1	Q2	OUT
L	L	off	off	Hi-Z
L	H	off	off	Hi-Z
H	L	on	off	L
H	H	off	on	H



1.5 三态门的使用 (例)

三态门用途：可用于连接总线，多个三态输出连在一起等



第3章 组合逻辑电路

第一讲 组合逻辑电路概述

第二讲 典型组合逻辑部件设计

第三讲 组合逻辑电路时序分析

第二讲 典型组合逻辑部件

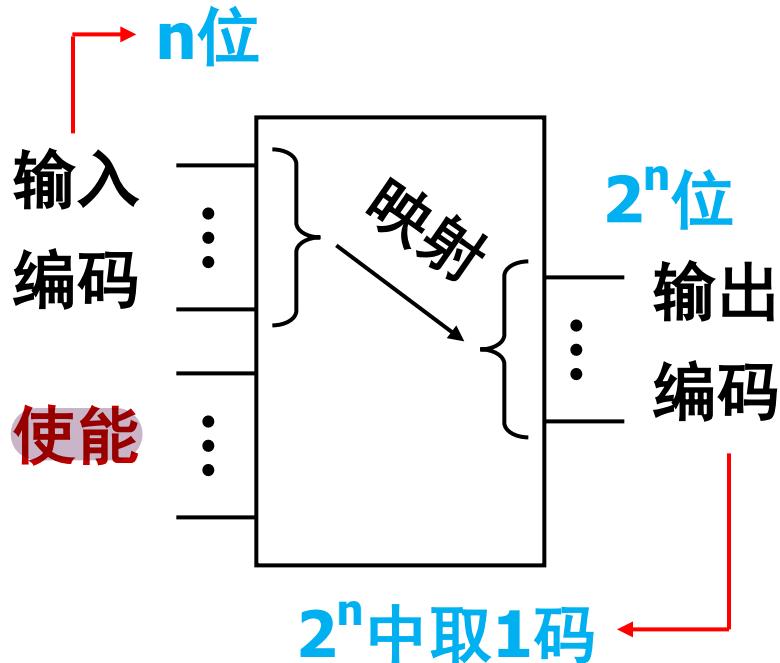
- ◆译码器和编码器
- ◆多路选择器和多路分配器
- ◆半加器和全加器

2 典型组合逻辑部件

- ◆ 大型的复杂数字系统通常采用**层次化、模块化方式**构建
- ◆ 数字系统由**基本组合逻辑元件**和**时序逻辑元件**相互连接而构建
- ◆ 组合逻辑元件的基本功能有**译码、选择、比较、运算、缓存并传送等**

2.1 (a) 译码器

- ◆ 译码器 (decoder) : 一种多输入、多输出的组合电路。
 - 电路功能反映输入编码与输出编码之间的映射关系
 - 最简单的译码器输入和输出关系是：若输入的二进制编码值是 x ，则第 x 条输出线为 1，其余输出全为 0
 - 可以通过使能端 EN 来控制



- **$n-2^n$ 译码器**
 - 输入： n 位二进制编码
 - 输出： 2^n 中取 1 码
- 例如：
 - 2-4 译码器
 - 3-8 译码器
 - 4-16 译码器

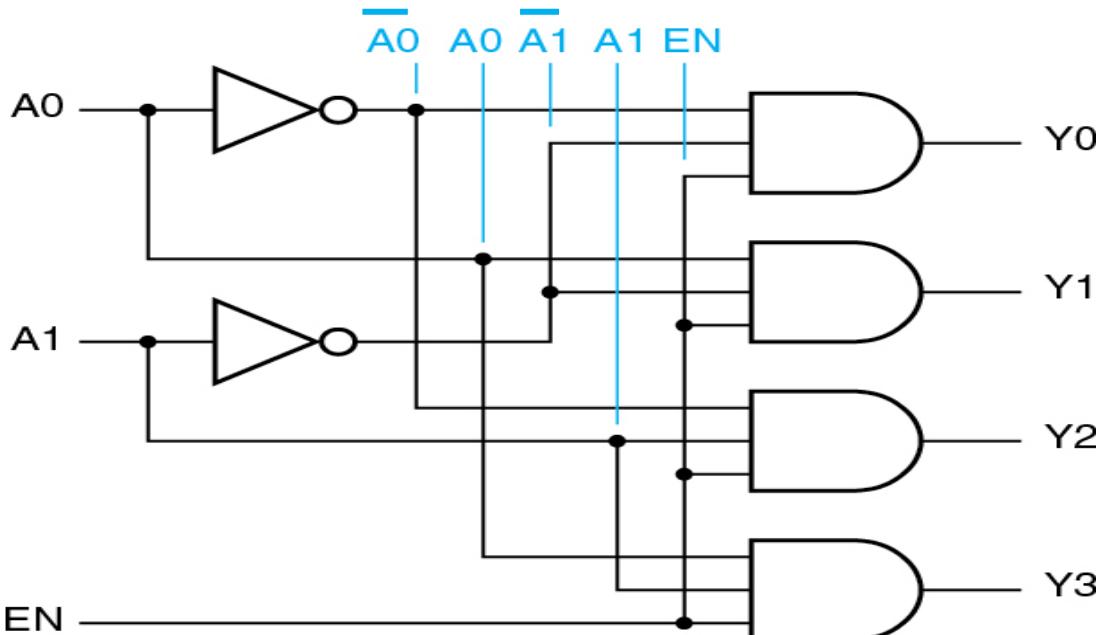
2.1 (a) 2-4译码器

x表示该输入任意，即任意值

例1：2-4译码器芯片74X139

- ◆ 输出端高电平有效，表示选中对应输入信号的最小项
- ◆ 通过使能端EN (Enable) 禁止或实现相应功能
- ◆ EN=0时，输出为全0

Inputs			Outputs			
EN	A1	A0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



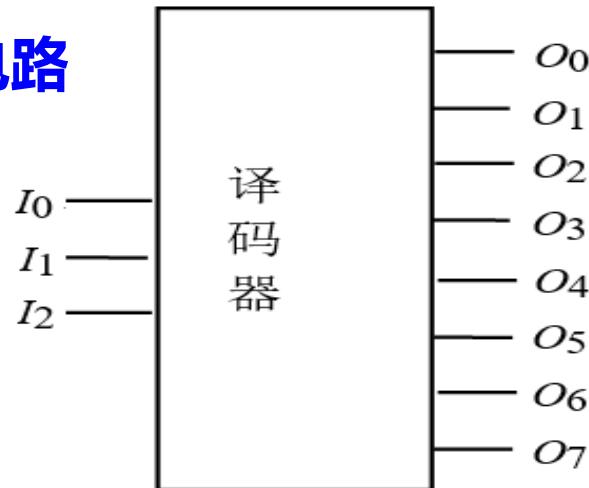
其实上面是四张真值表。
下面是四个逻辑电路图合体

$$\text{如, } Y_2 = EN \cdot A1 \cdot \bar{A0} = EN \cdot m_2$$

2.1 (a) 3-8译码器

例2：3-8译码器电路

不带EN端



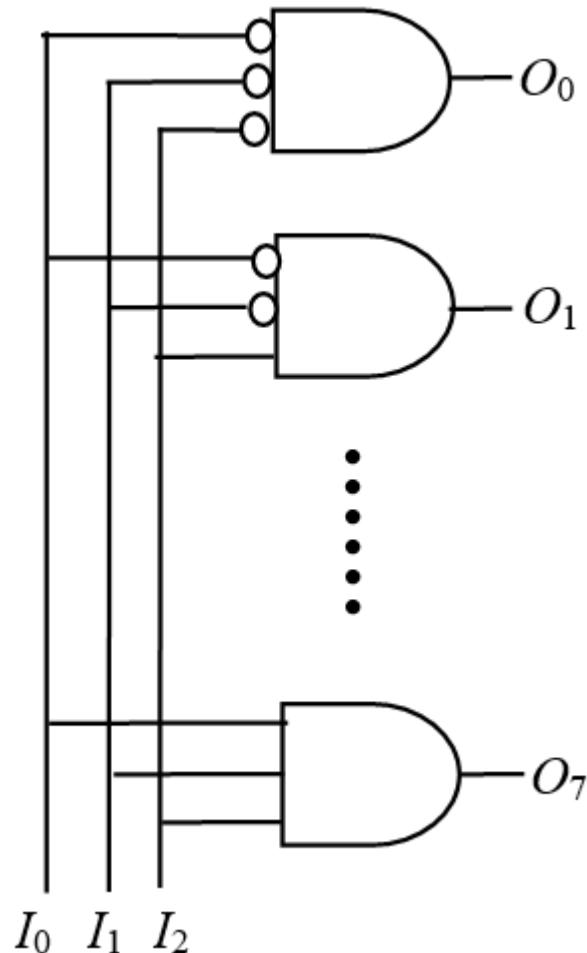
I_0	I_1	I_2	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

输入从高到低位： I_0, I_1, I_2
谁是最高位，谁是最低位，不能弄错

输出：

O_0 有效，代表输入为0...

O_7 有效，代表输入为7



2.1 (a) 74X138芯片 (了解即可)

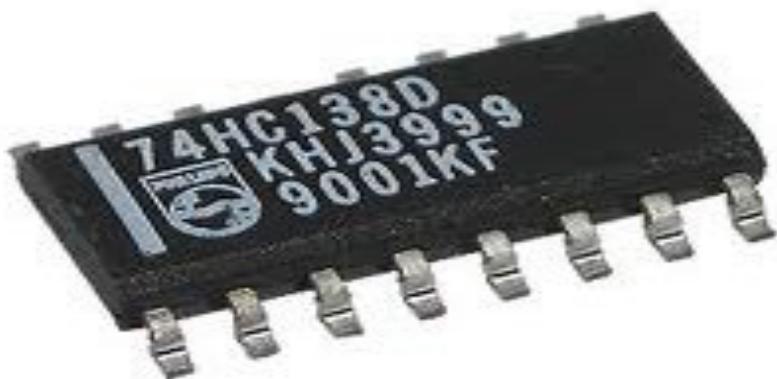
例2：3-8译码器芯片 74X138

◆输出端**低电平有效** (加O, 变量名中加_L), 对应输入信号的最大项

◆有3个使能控制端

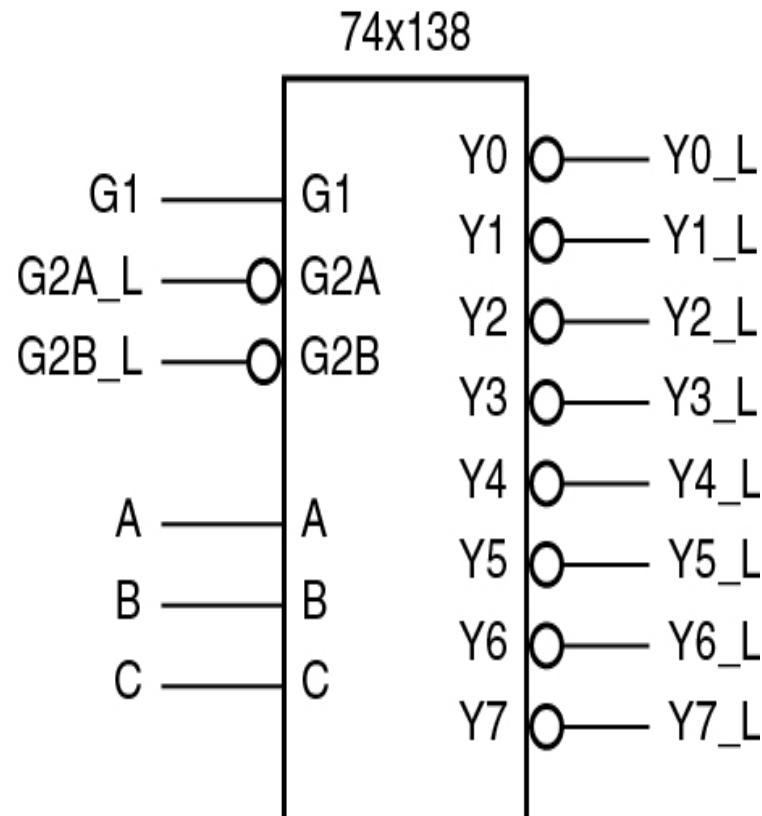
G1、G2A_L、G2B_L

多片扩展为4-16等译码器时可用



74 系列是德州仪器公司生产的中小规模集成电路芯片

低电平有效：一般来说是考虑到电路功耗、抗干扰和稳定性等。



(b)

2.1 (a) 74X138芯片 (了解即可)

74X138的功能表

Inputs						Outputs							
G1	G2A_L	G2B_L	C	B	A	Y7_L	Y6_L	Y5_L	Y4_L	Y3_L	Y2_L	Y1_L	Y0_L
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	0	1	1	1	1	1	1	0	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

EN也是低电平 (0) 时有效

M₆

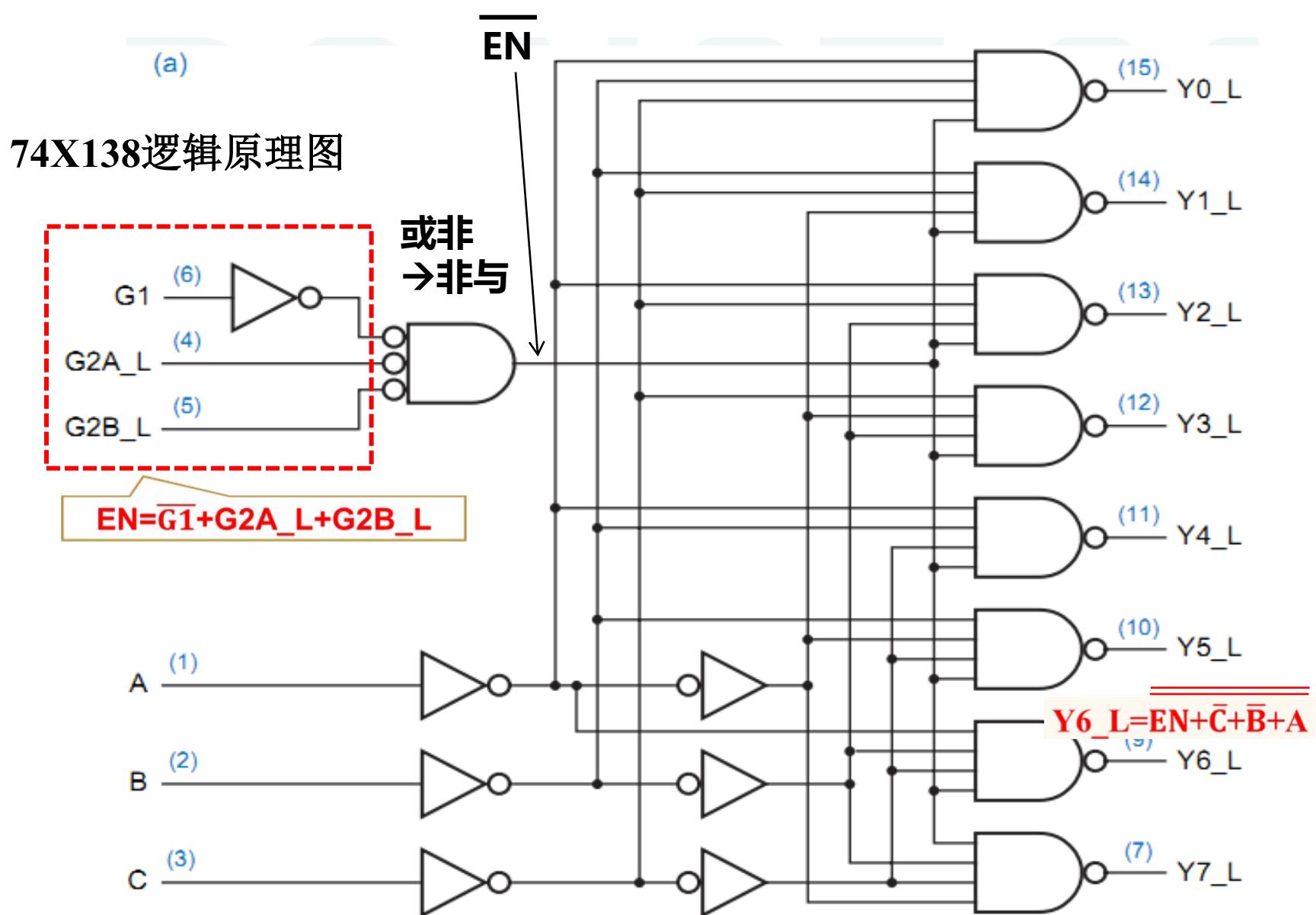
$$EN = \overline{G1} + G2A_L + G2B_L$$

$$\text{例: } Y6_L = EN + \bar{C} + \bar{B} + A$$

2.1 (a) 74X138芯片 (了解即可)

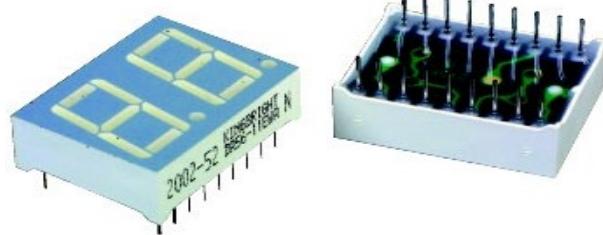
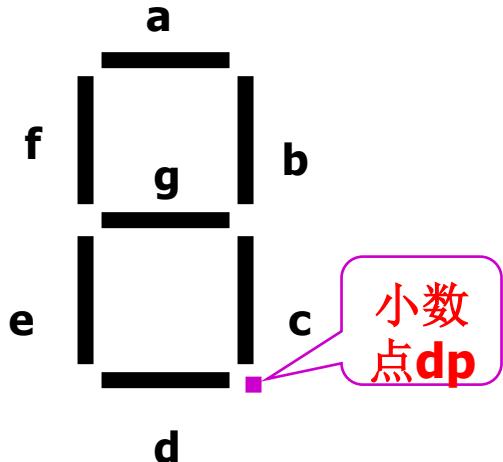
使能控制端

输入端

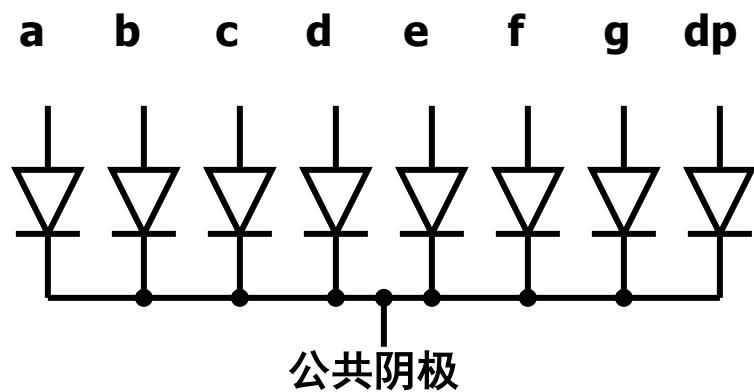
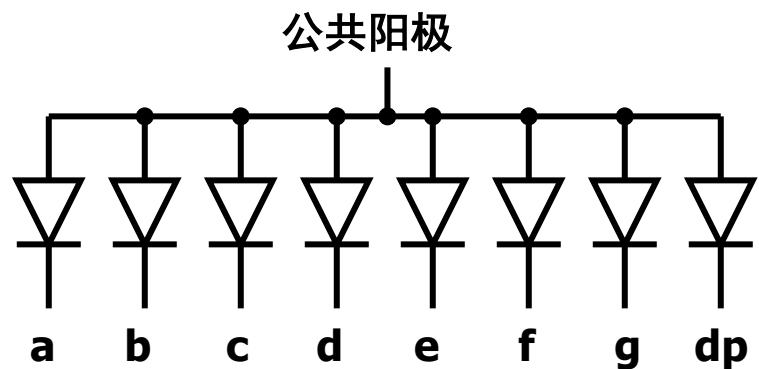


2.1 (a) 译码器的应用

例3：七段显示译码器（数码管）



每段是一个LED（发光二极管），
通过控制其亮和灭，可得到数字
或字母等符号的形状

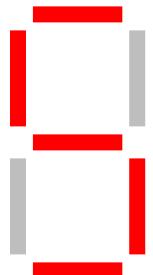
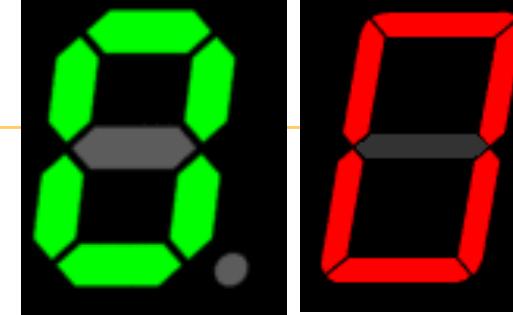


共阳极，输入为低电平二极管导通；
共阴极，输入为高电平二极管导通。

2.1 (a) 译码器的应用

- ◆ 输入信号：4位二进制编码
- ◆ 输出：七段码（的驱动信号） $a \sim g$ （不包括小数点）

假设共阴极，即 1-亮；0-灭



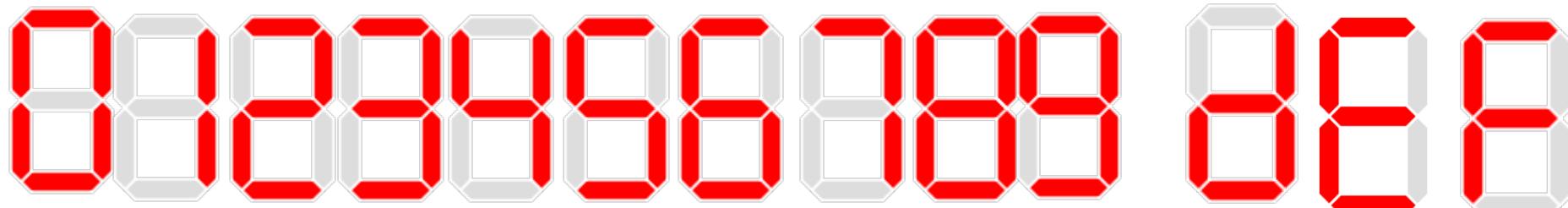
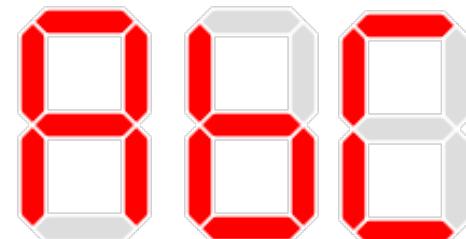
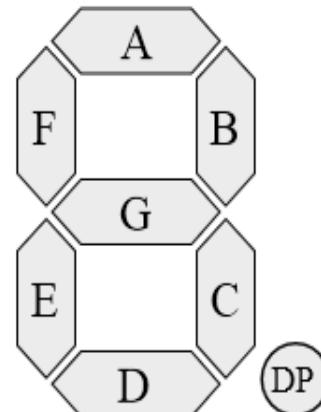
1011011



1111110



0110011



2.1 (a) 译码器的应用

◆ 七段数字显示译码器真值表

——a-g这七个信号各自对应的七张真值表合并得到右表

从高到低位：
A3A2A1A0

	A3 A2 A1 A0	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	1	0	1	1	1	1	1
7	0 1 1 1	1	1	1	0	0	0	0
8	1 0 0 0	1	1	1	1	1	1	1
9	1 0 0 1	1	1	1	0	0	1	1
A	1 0 1 0	1	1	0	1	1	1	1
b	1 0 1 1	0	0	1	1	1	1	1
C	1 1 0 0	1	0	0	1	1	1	0
d	1 1 0 1	0	1	1	1	1	0	1
E	1 1 1 0	1	0	0	1	1	1	1
F	1 1 1 1	1	0	0	0	1	1	1

2.1 (a) 译码器的应用

◆ 根据显示需要，考虑是否使用A~F输入信号

以下是输出信号a的卡诺图

A~F输入信号作为无关项
(最终输入不应该出现A-F)

		A1A0	00	01	11	10
		A3A2	00	01	11	10
A3A2	A1A0	00	1	0	1	1
		01	0	1	1	1
		11	d	d	d	d
		10	1	1	d	d

$$a = A3 + A1 + A2A0 + \overline{A2} \overline{A0}$$

$$a = A3 + A1 + A2 \odot A0$$

	A3	A2	A1	A0	a
A	1	0	1	0	1
b	1	0	1	1	0
C	1	1	0	0	1
d	1	1	0	1	0
E	1	1	1	0	1
F	1	1	1	1	1

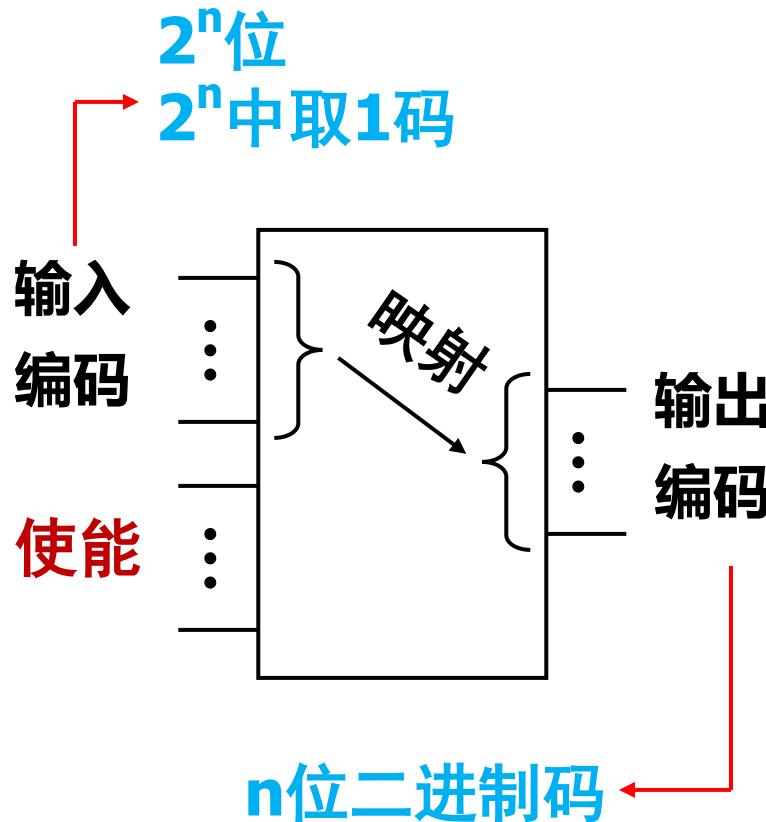
A~F输入信号作为有效项

		A1A0	00	01	11	10
		A3A2	00	01	11	10
A3A2	A1A0	00	1	0	1	1
		01	0	1	1	1
		11	d	d	d	d
		10	1	1	d	d

$$\begin{aligned} a &= A3 \cdot \overline{A2} \cdot \overline{A1} + \overline{A3} \cdot A2 \cdot A0 \\ &+ A3 \cdot \overline{A0} + A1 \cdot \overline{A0} + \overline{A2} \cdot \overline{A0} \\ &+ A2 \cdot A1 + \overline{A3} \cdot A1 \end{aligned}$$

2.1 (b) 编码器

- ◆ 编码器encoder：译码器的反函数电路
即输出是输入信号的二进制编码



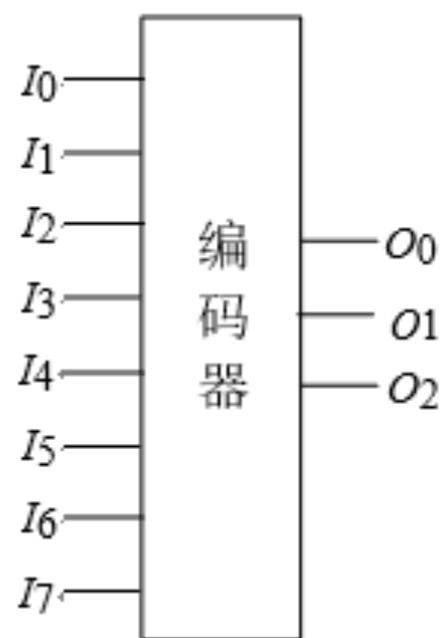
- 最常见是 2^n -n编码器，也称为二进制编码器。
 - 2ⁿ个输入端
 - n个输出端
- 分类：
 - 互斥(唯一输入)编码器
 - 优先级编码器

2.1 (b) 互斥编码器

◆ 3 位二进制编码器 (8-3 编码器)

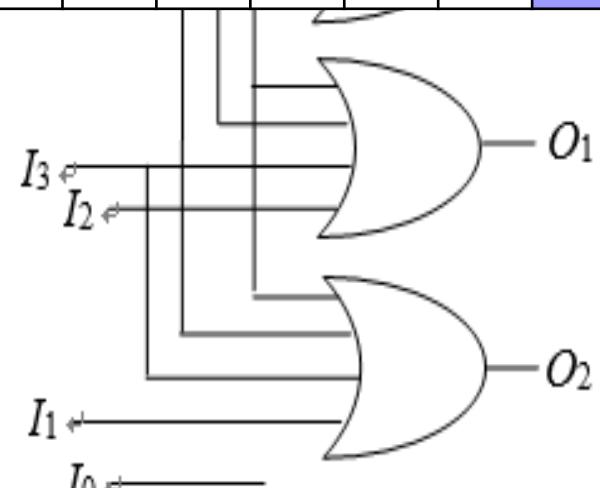
只能有1个为1，输入 $I_0 \sim I_7$ 是一组互斥变量，每次
其余都为0。输出为3位的二进制编码。

$I_0, I_1 \dots I_7$ 分别代表输入值为0~7
编码结果从高到低位: O_0, O_1, O_2



	O_0	O_1	O_2
I_0	0	0	0
I_1	0	0	1
I_2	0	1	0
I_3	0	1	1
I_4	1	0	0
I_5	1	0	1
I_6	1	1	0
I_7	1	1	1

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	O_0
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	1



a) 编码器符号

b) 编码器真值表
(三个真值表的合体)

c) 编码器电路图
(三个电路图的合体)

2.1 (b) 优先权编码器

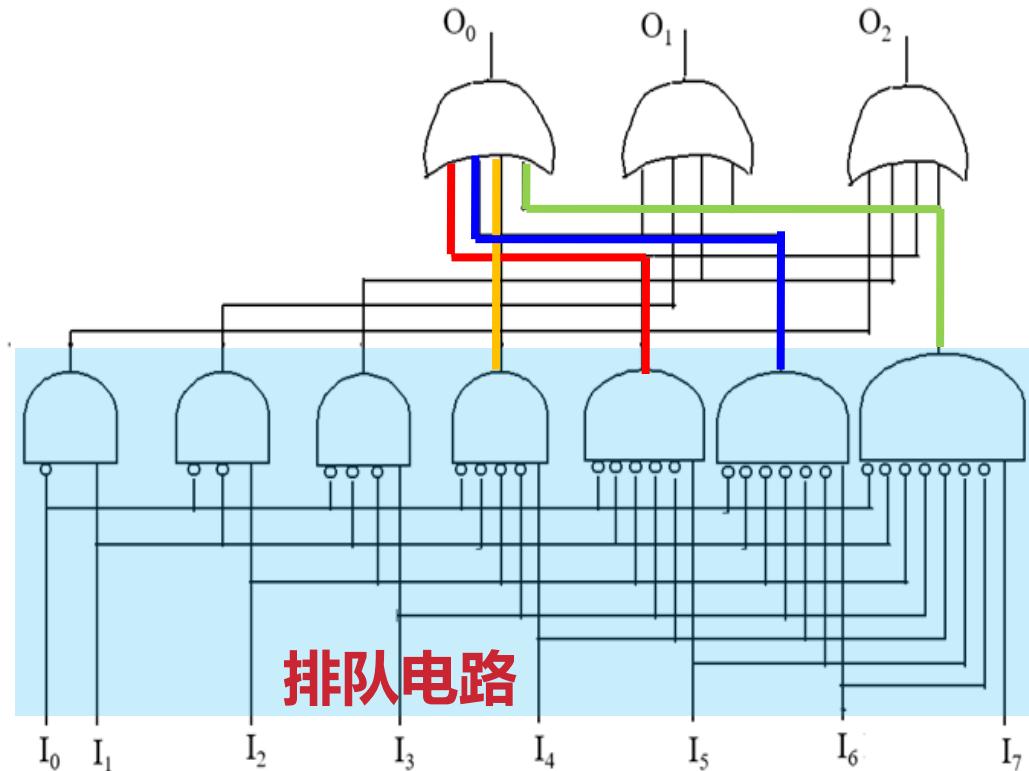
◆ 3 位优先权编码器

编码结果从高到低位: O₀, O₁, O₂

- 多个输入可同时为1，但只对优先级最高的输入进行编码输出
- 假定优先级顺序为 I₀ > I₁ > I₂ > I₃ > I₄ > I₅ > I₆ > I₇，则：

I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	O ₀	O ₁	O ₂
1	x	x	x	x	x	x	x	0	0	0
0	1	x	x	x	x	x	x	0	0	1
0	0	1	x	x	x	x	x	0	1	0
0	0	0	1	x	x	x	x	0	1	1
0	0	0	0	1	x	x	x	1	0	0
0	0	0	0	0	1	x	x	1	0	1
0	0	0	0	0	0	1	x	1	1	0
0	0	0	0	0	0	0	1	1	1	1

优先权编码器
的功能描述

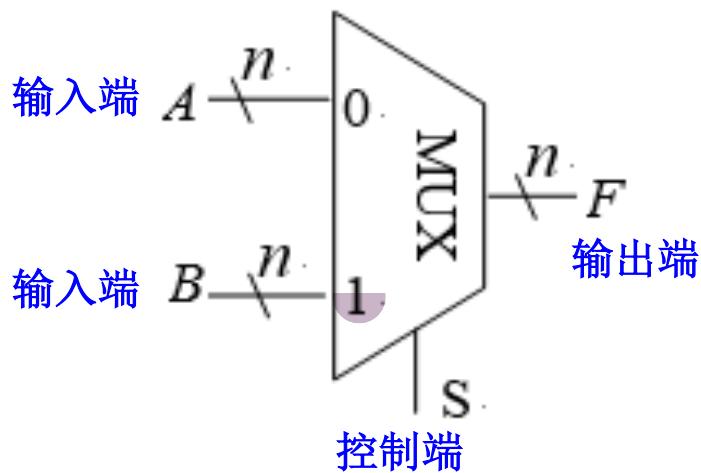


优先权编码器逻辑电路图

2.2 (a) 多路选择器

九选一

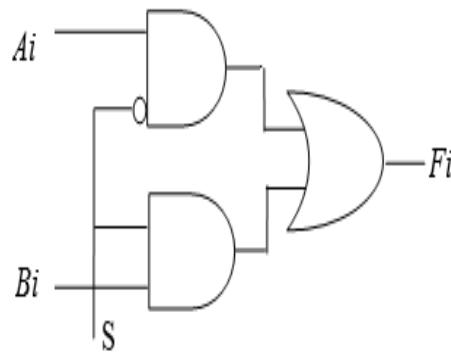
- ◆ 2-路选择器有两个输入端和一个输出端，有一个控制端，用于控制选择哪一路输出
- ◆ 实际计算机中，2-路选择器的每个输入、输出端通常都有n位



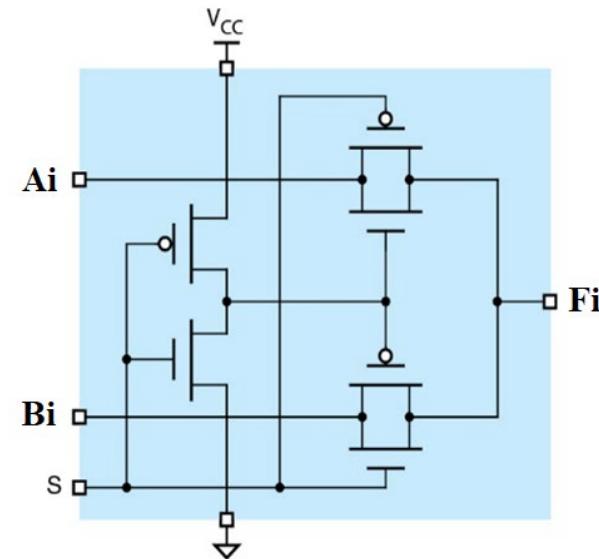
2-路选择器符号

$S=1$ 时 $F=B$

$S=0$ 时 $F=A$



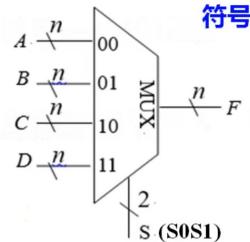
基本门电路实现
一位2-路选择器



传输门实现
2-路选择器

2.2 (a) 多

◆ 4-路选择器

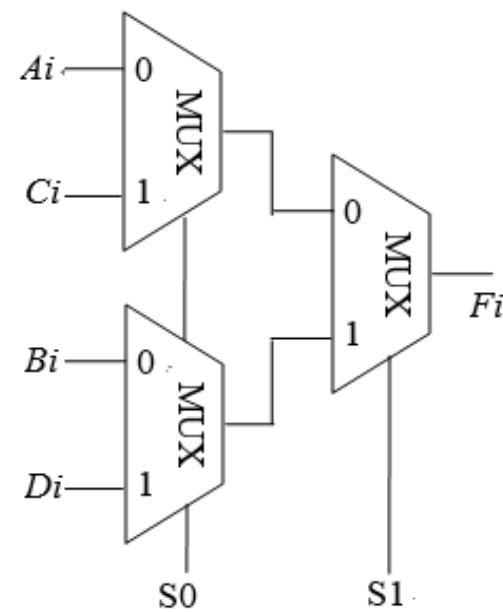
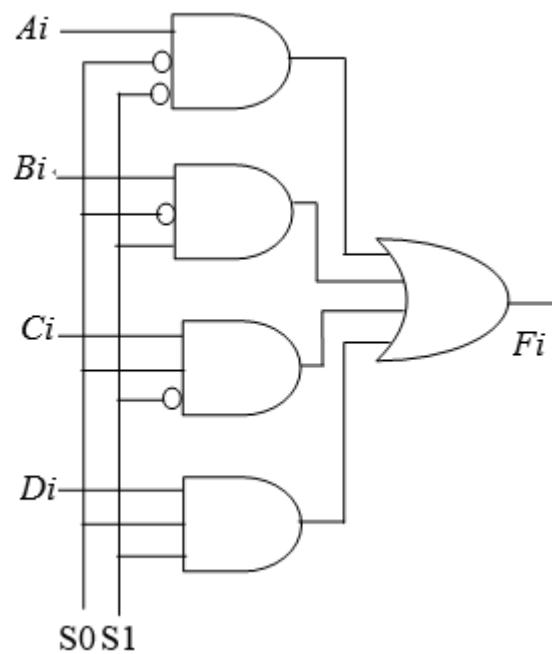


S0	S1	A	B	C	D	F
0	0	1	x	x	x	1
0	0	0	x	x	x	0
0	1	x	1	x	x	1
0	1	x	0	x	x	0
1	0	x	x	1	x	1
1	0	x	x	0	x	0
1	1	x	x	x	1	1
1	1	x	x	x	0	0

真值表

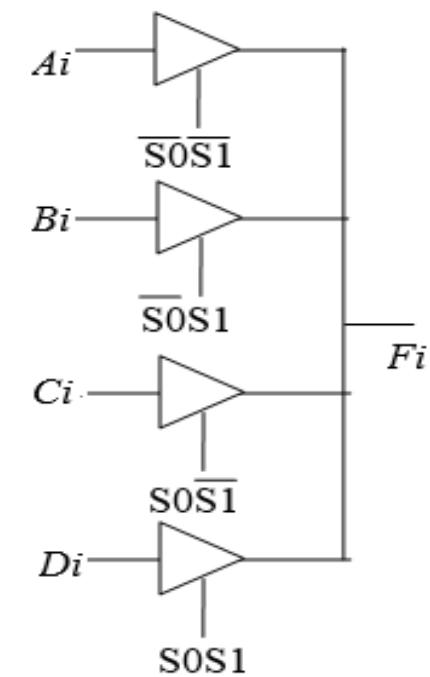
S0	S1	F
0	0	A
0	1	B
1	0	C
1	1	D

◆ 一位4-路选择器的实



两级门电路

多层次级联

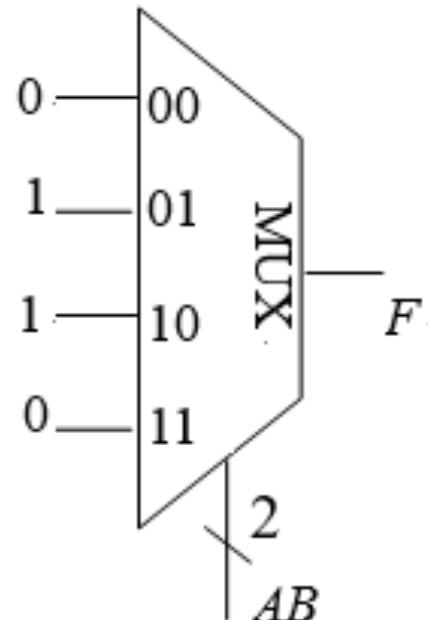


三态门电路

2.2 (a) 多路选择器的应用例子

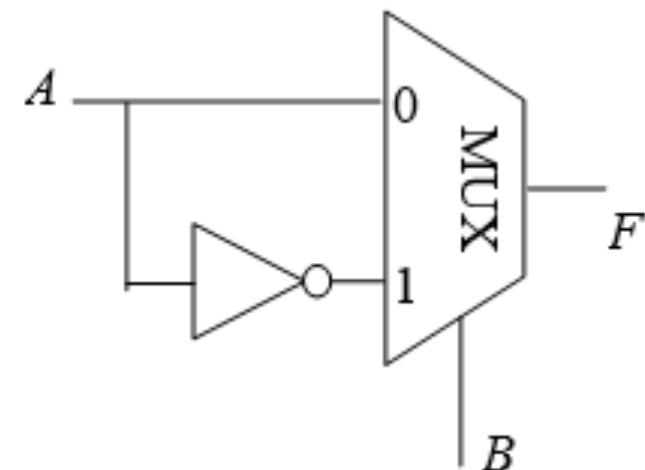
- ◆ 可以基于多路选择器实现组合逻辑电路的功能
- ◆ 例：基于多路选择器实现某组合逻辑电路的功能（可用如下真值表描述）

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



真值表

用一个4-路选择器实现

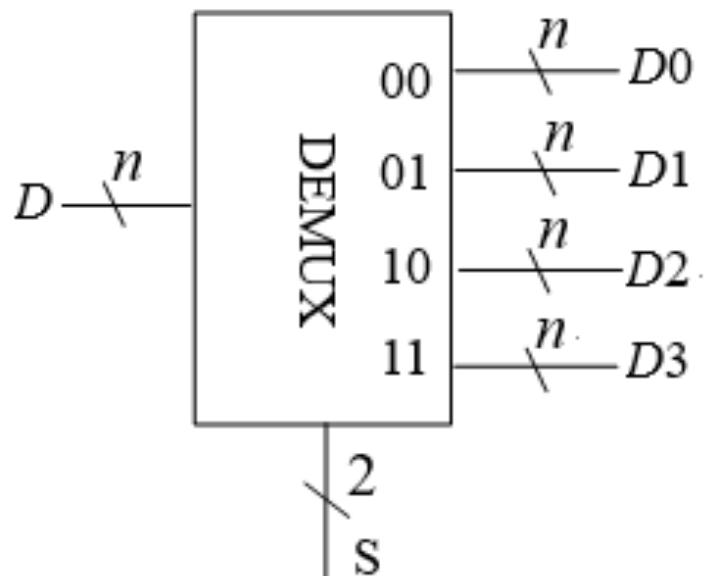


用一个2-路选择器
和一个非门实现

2.2 (b) 多路分配器

- ◆ **多路分配器 (demultiplexer)** : 把唯一的输入信号发送到多个输出端中的一个。从哪一个输出端送出输入信号，取决于控制端。简写为DMUX或DEMUX
- ◆ 4-路分配器的符号和真值表

四路分配器的符号



四路分配器真值表

S0	S1	D0	D1	D2	D3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

四路分配器 ($n=1$) 的电路图?

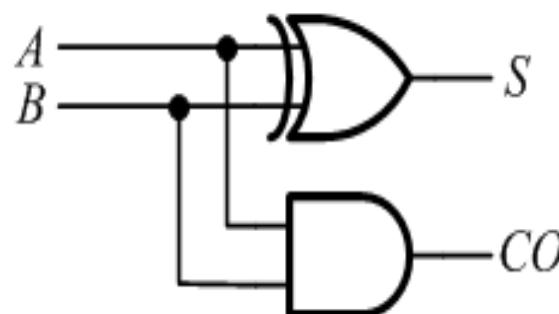
2.3 (a) 半加器

◆ 半加器 (Half Adder, 简称HA) : 仅考虑加数和被加数, 不考虑低位来的进位

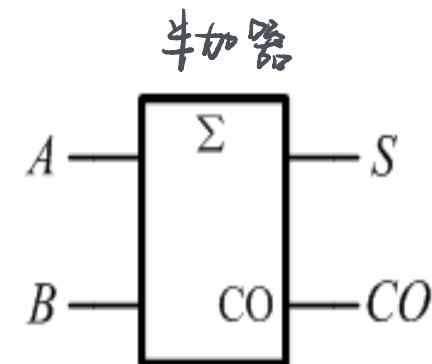
输入		输出	
被加数	加数	和数 <small>sum</small>	进位 <small>Cout</small>
A	B	S	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

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

$$CO = A \cdot B$$



电路图



逻辑符号

2.3 (b) 全加器

◆全加器 (Full Adder, 简称FA)

输入为加数、被加数和低位进位Cin，输出为和F、进位Cout

真值表

A	B	Cin	F	Cout
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

$$F = \overline{A} \cdot \overline{B} \cdot \text{Cin} + \overline{A} \cdot B \cdot \text{Cin} + A \cdot \overline{B} \cdot \text{Cin} + A \cdot B \cdot \text{Cin}$$
$$\text{Cout} = \overline{A} \cdot \overline{B} \cdot \text{Cin} + A \cdot \overline{B} \cdot \text{Cin} + A \cdot B \cdot \overline{\text{Cin}} + A \cdot B \cdot \text{Cin}$$

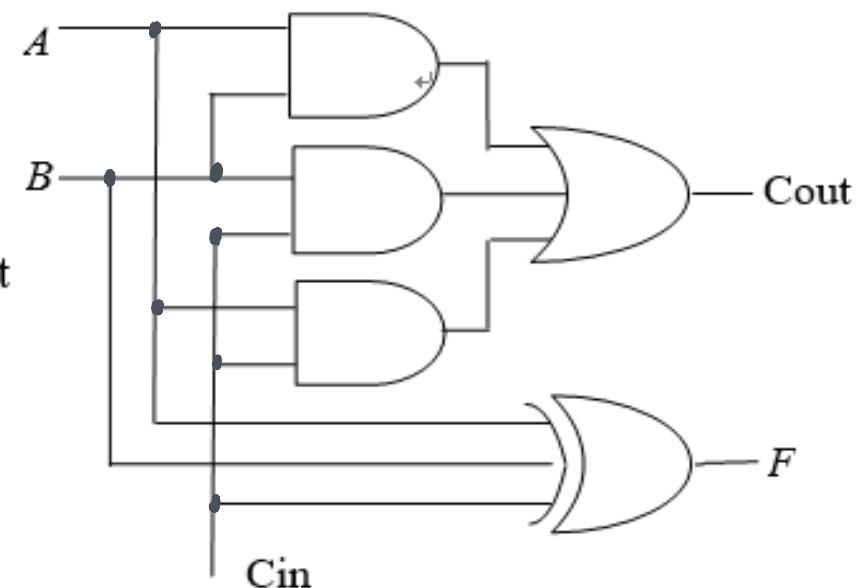
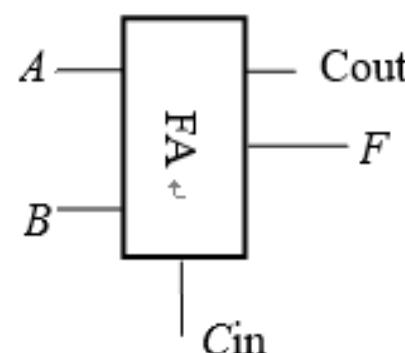
化简后：

$$F = A \oplus B \oplus \text{Cin}$$

$$\text{Cout} = A \cdot B + A \cdot \text{Cin} + B \cdot \text{Cin}$$

全加器逻辑电路图

逻辑符号



第三讲 组合逻辑部件时序分析

- ◆ 传输延迟和最小延迟
- ◆ 竞争冒险

3 组合逻辑电路时序分析

- ◆ 信号通过连线和电路元件时会有一定时间的延迟 (Delay)
- ◆ 电路的延迟取决于电路内部的设计及外部特性，影响因素包括但不限于：
 - 连线的长短、元件的数量
 - 电路制造工艺、工作电压
 - 环境噪声、温度等外在条件
 - 高低电平的转换过渡时间

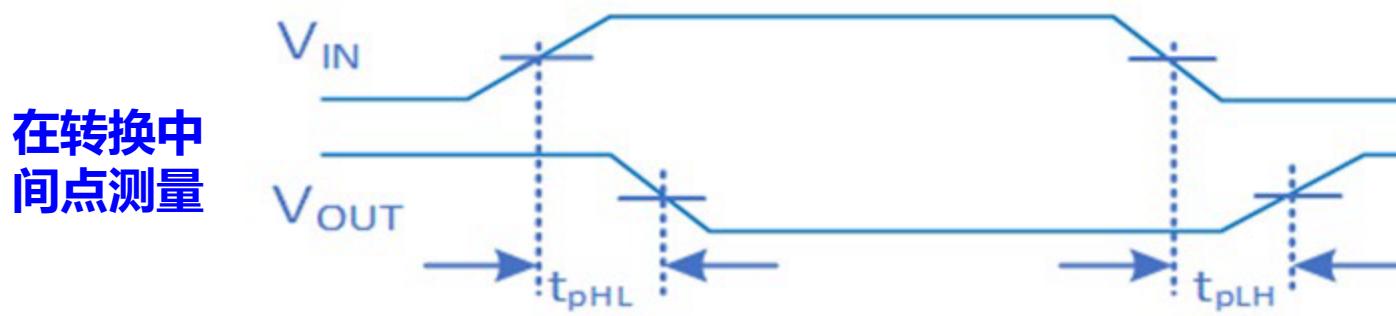
因此，任何组合逻辑电路从输入信号的改变，到随之引起的输出信号的改变，都有一定时间的延迟

* 3.1 t_{pHl} 和 t_{pLH}

◆通常用时序图反映电路的延迟

- 下降沿延迟 t_{pHl} : 输入变化引起相应输出从高到低变化的时间
- 上升沿延迟 t_{pLH} : 输入变化引起相应输出从低到高变化的时间

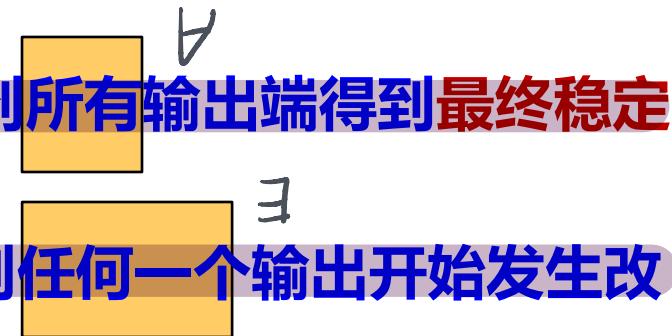
◆通常取信号转换时间中间点来测量延迟时间 (以反相器电路为例)



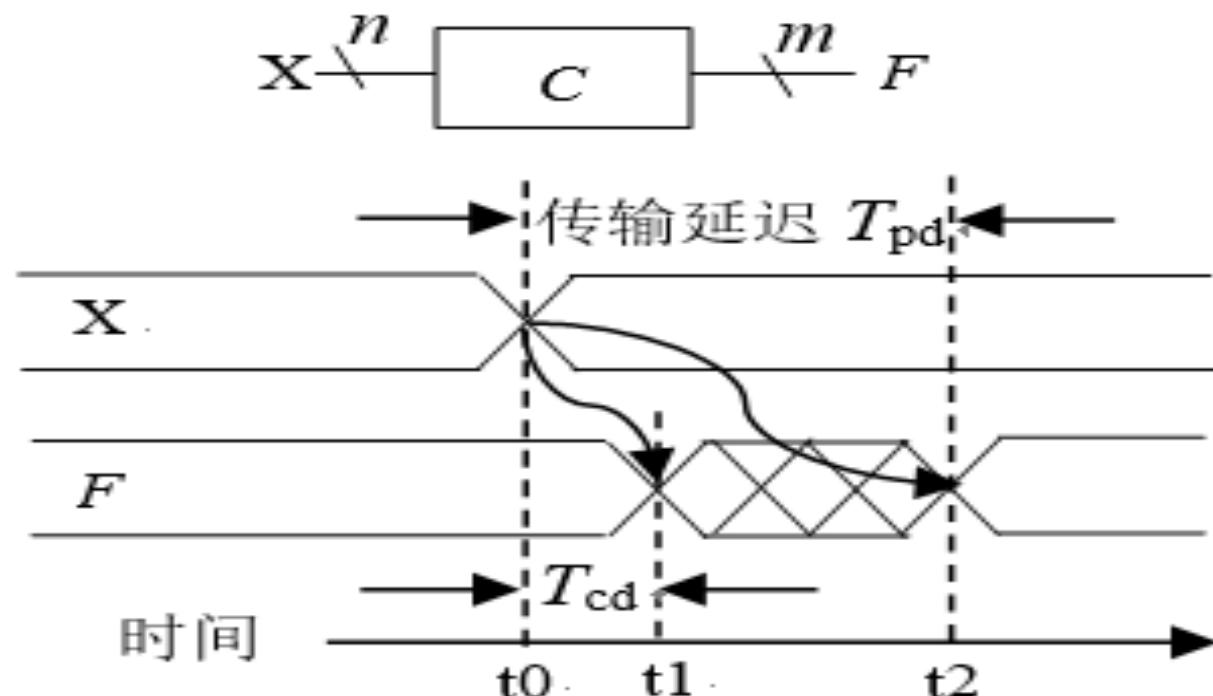
3.1 传输延迟和最小延迟

- 组合逻辑电路的时序特征主要包括**传输延迟 (propagation delay)** 和**最小延迟 (contamination delay)**

- **传输延迟 T_{pd}** : 从输入端的变化开始到所有输出端得到最终稳定的信号所需的最长时间
- **最小延迟 T_{cd}** : 从输入端的变化开始到任何一个输出开始发生改变所需的最短时间

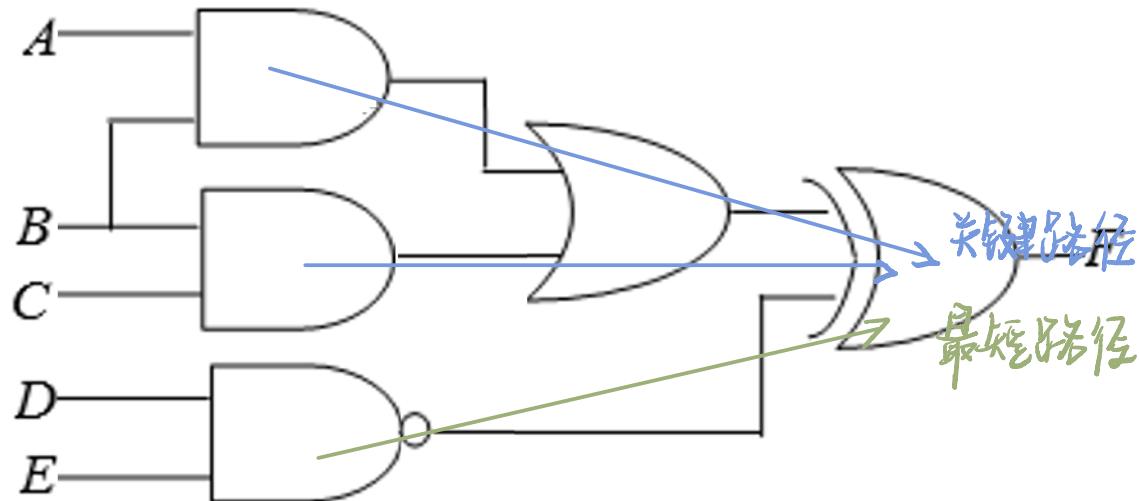


组合逻辑电路C
的传输延迟 T_{pd}
和最小延迟 T_{cd}



3.1 关键路径和最短路径

- ◆ **关键路径：**一个组合逻辑电路在输入和输出之间经过的最长路径
- ◆ 传输延迟就是**关键路径**上所有元件的**传输延迟之和**
- ◆ **最小延迟**就是**最短路径**上所有元件的**最小延迟之和**

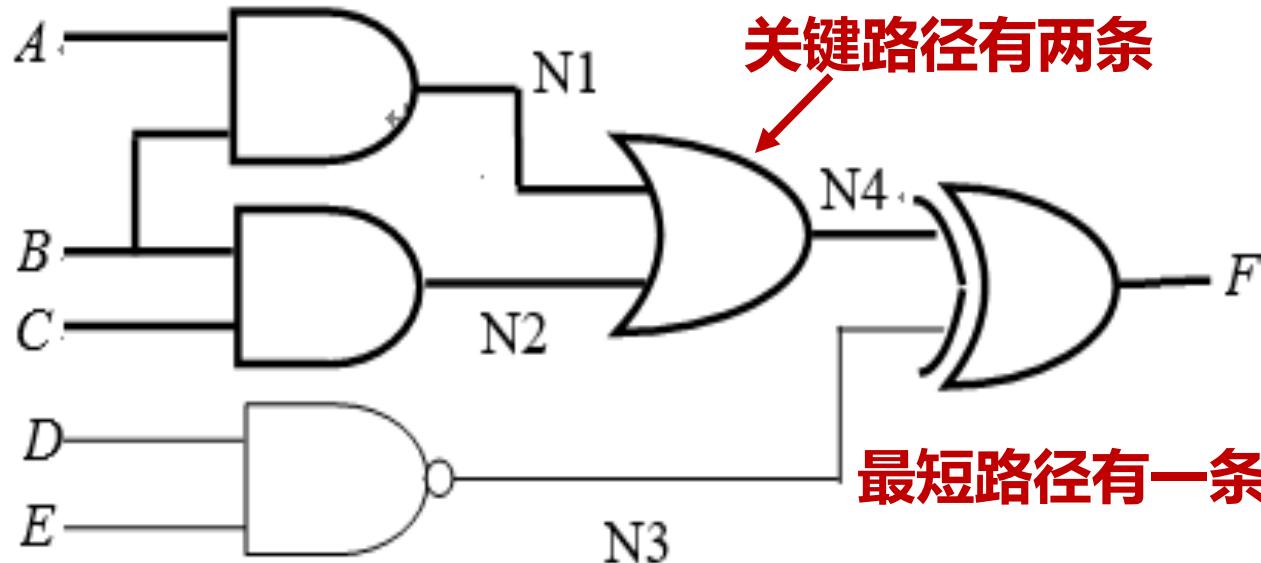


关键路径由哪些结点和逻辑门组成？最短路径呢？
传输延迟和最小延迟怎么计算？

3.1 关键路径和最短路径 (例)

- ◆ 例：假定所有逻辑门电路的传输延迟和最小延迟分别为90ps和60ps，计算下图中组合逻辑电路的传输延迟和最小延迟。

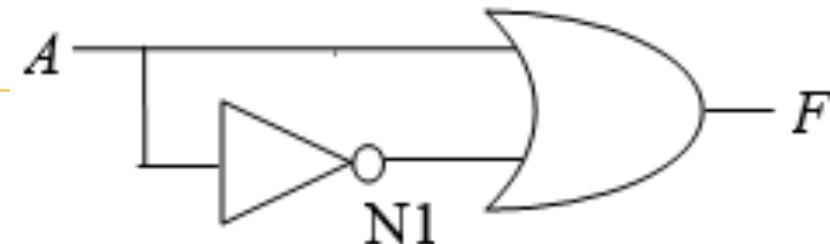
注意：实际电路中器件的延迟总会在一定范围内波动，所以最简单的门电路也可以有最大和最小延迟。



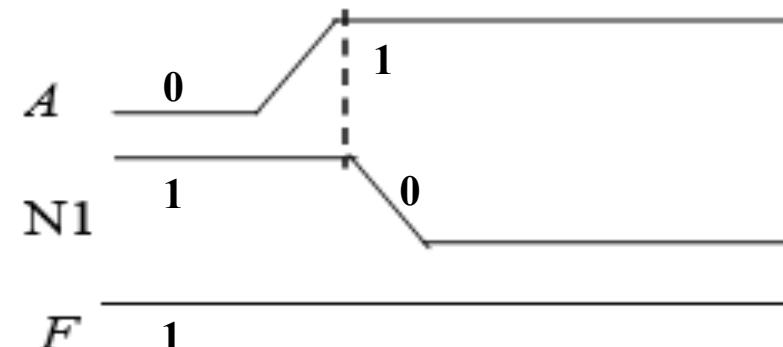
传输延迟 T_{pd} 为3级门传播延迟之和，即 $90\text{ps} \times 3 = 270\text{ps}$
最小延迟 T_{cd} 为2级门最小延迟之和，即 $60\text{ps} \times 2 = 120\text{ps}$

3.2 竞争和冒险

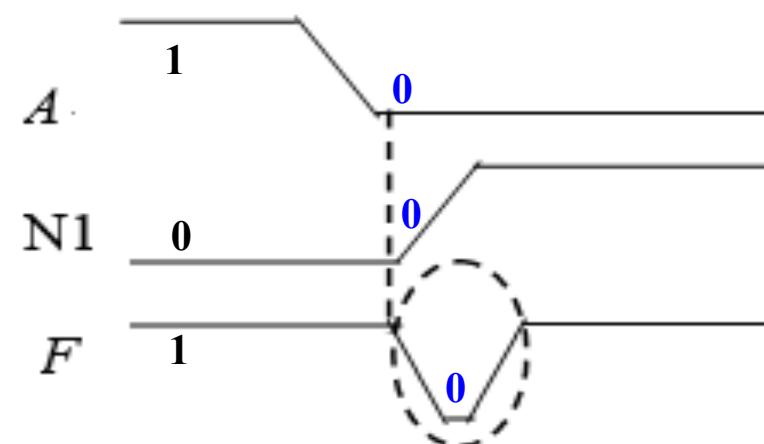
- 某个输入信号经过两条或两条以上的不同路径作用到输出端，由于各路径延迟不同，因而该输入信号对输出端会发生先后不同的影响，该现象称为**竞争 (race)**
- 由于竞争的存在，在输入信号变化的瞬间，输出端可能会出现不正确的尖峰信号，这种信号称为**毛刺 (glitch)**
- 出现毛刺的电路称为存在**冒险 (hazard)**



可能发生冒险的电路



未发生毛刺

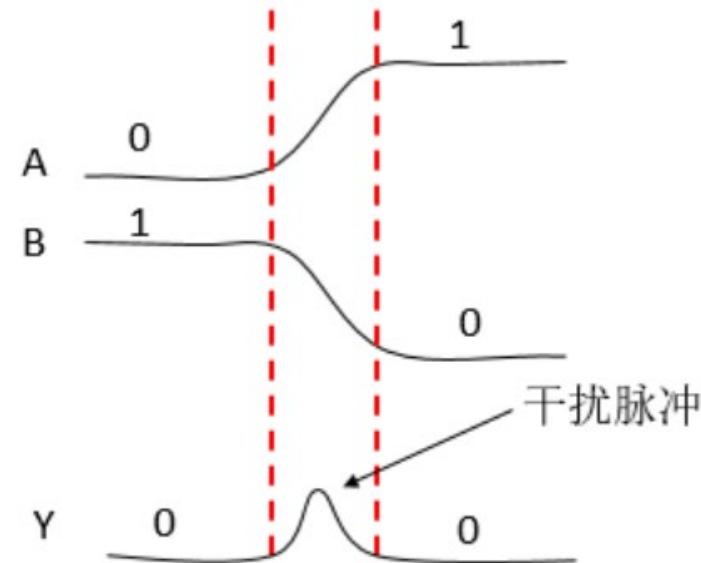


出现毛刺

3.2 竞争和冒险

◆ 实际硬件电路中，只要一个门电路的多个输入端延时不同，就有可能产生竞争与冒险

◆ 典型情况：当一个逻辑门的两个输入端的信号同时向相反方向变化时，从变化开始，到达稳定状态所需的时间不同，产生竞争



与门电路也可能产生毛刺

(1) 在逻辑表达式中，保持一个变量X固定不动，将剩余其他变量用0或1代替，如果最后能化简成 $X+X\sim$ 或者 $X \cdot X\sim$

(2) 卡诺图化简后，有相邻的1没有被包含在同一个卡诺圈里

——可判定此逻辑存在竞争与冒险

A	B	C	00	01	11	10
0	0	0	0	0	1	0
1	1	1	1	1	1	0

$$Y = AB' + BC$$

3.2 竞争和冒险

- 可通过低通滤波或增加冗余项来修改逻辑设计等方式避免毛刺

		B	C	00	01	11	10
		A	0	0	0	1	0
		1	1	1	1	1	0

$Y = AB' + BC + AC$

- 毛刺并不是对所有的输入都有危害，如触发器的D输入端，只要数据不出现在时钟的上长升沿，并满足数据的建立保持时间，就不会对系统造成危害（第四章有相关概念）
- 而当毛刺信号成为系统的控制信号，触发器的清零信号，预置信号，时钟信号，或锁存器的输入信号时就会产生逻辑错误。（后续有相关概念）

第3章总结

- ◆ 数字逻辑电路由若干元件（可以是一个电路）和若干结点互连而成
- ◆ 组合逻辑电路的输出值仅依赖于当前输入值
- ◆ 组合逻辑电路可以是两级电路或多级电路，两级电路的传输时间短，但占用集成电路物理空间更多，需进行时空权衡
- ◆ 组合逻辑电路设计：功能分析-列表-化简-逻辑表达式-画图-评价
- ◆ 无关项指输出取值可任意的项，真值表中用d表示，可用于化简
- ◆ 非法值指同时被高、低电平驱动的输出结点的值。
- ◆ 高阻态是三态门输出结点的一种非正常逻辑态，相当于“断开”
- ◆ 典型组合逻辑部件：译码/编码器、多路选择/分配器、半加/全加器
- ◆ 传输延迟：关键路径上所有元件的传输延迟之和
- ◆ 最小延迟：最短路径上所有元件的最小延迟之和
- ◆ 竞争、冒险、毛刺：不同路径延迟作用在同一输出端而引起

作业：习题3、4、6、7、9、11。提交截止日期：10月12日24:00