

数字逻辑设计

高翠芸

School of Computer Science

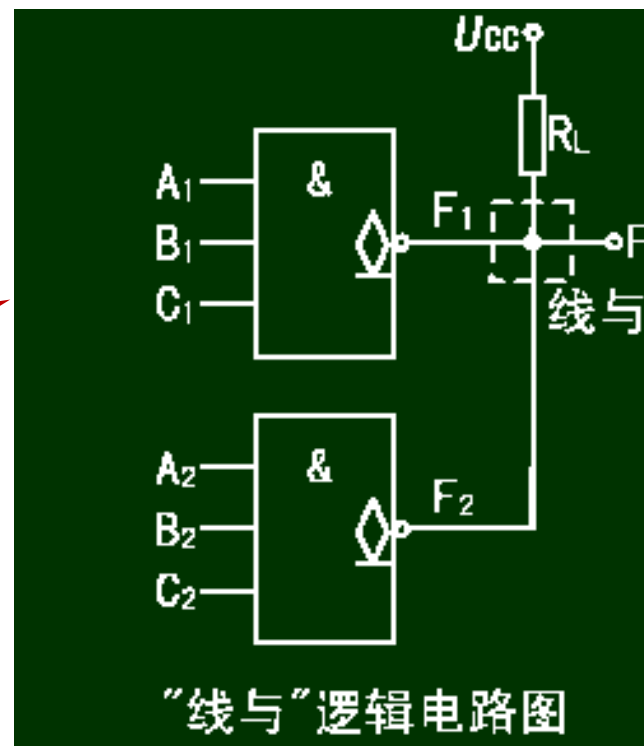
gaocuiyun@hit.edu.cn

线与和线或

- **线与**，接上拉电阻至**电源**。 $(\sim A) \& (\sim B) = \sim (A + B)$
- **线或**，接下拉电阻至**地**。 $(\sim A) + (\sim B) = \sim (AB)$
- 当看到一些芯片的 OC 输出端连在一起，而有一个**上拉电阻**时，这就是**线或/线与**了，但有时上拉电阻做在芯片的输入端内。

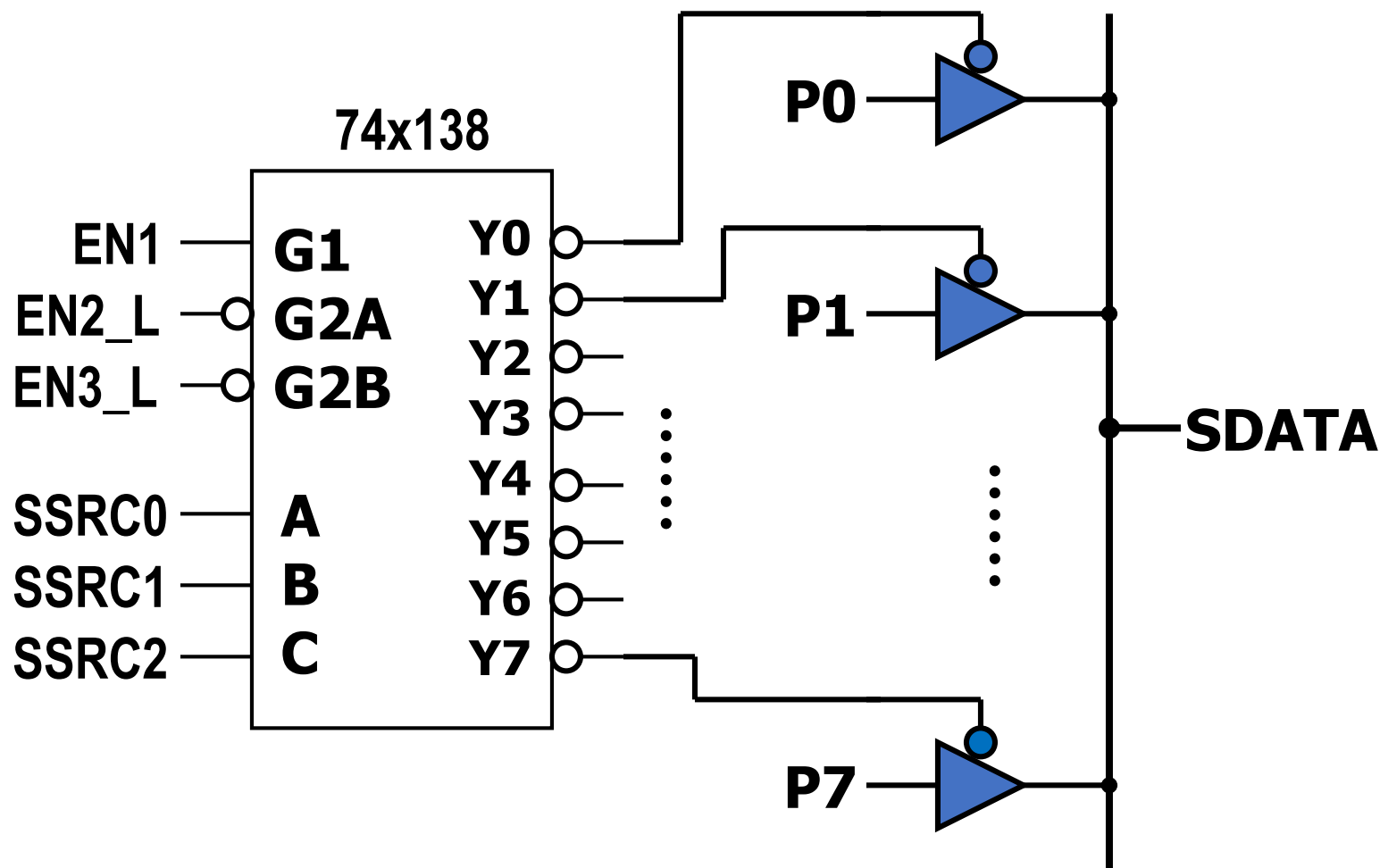
$$F = F_1 \cdot F_2 = A_1 B_1 C_1 \cdot \overline{A_2 B_2 C_2}$$

不使用OC门，
需要2个与非门、
1个与门

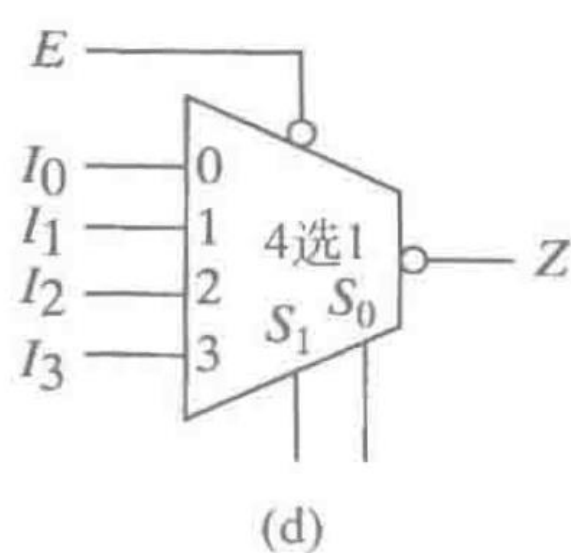
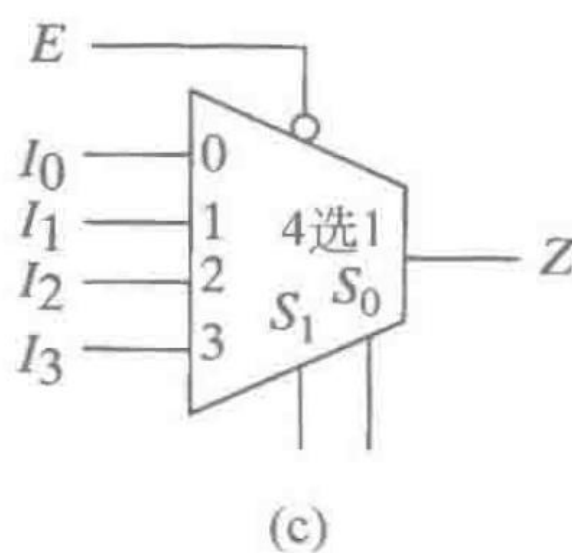
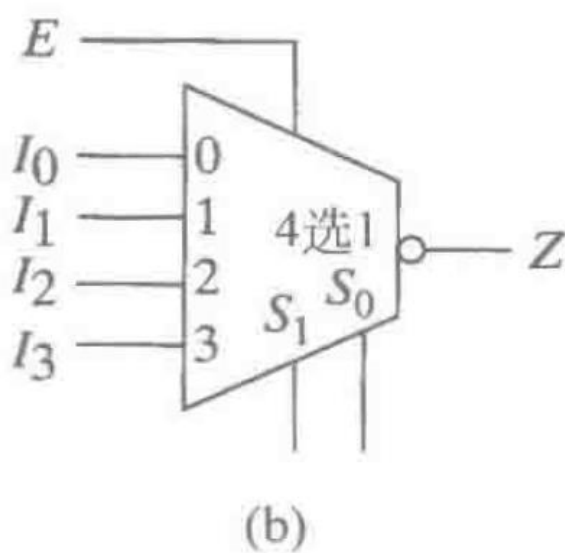
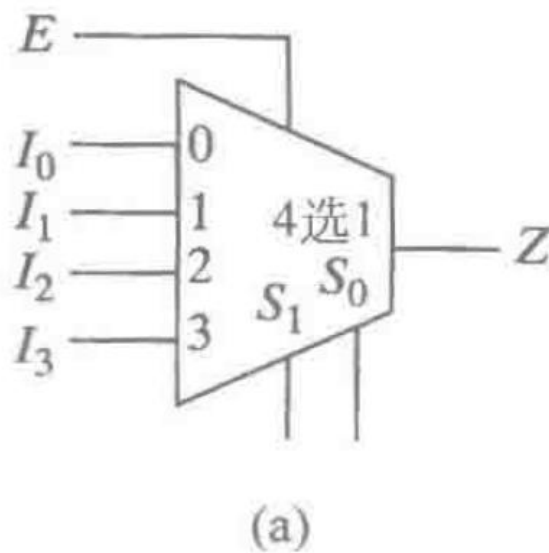


使能端

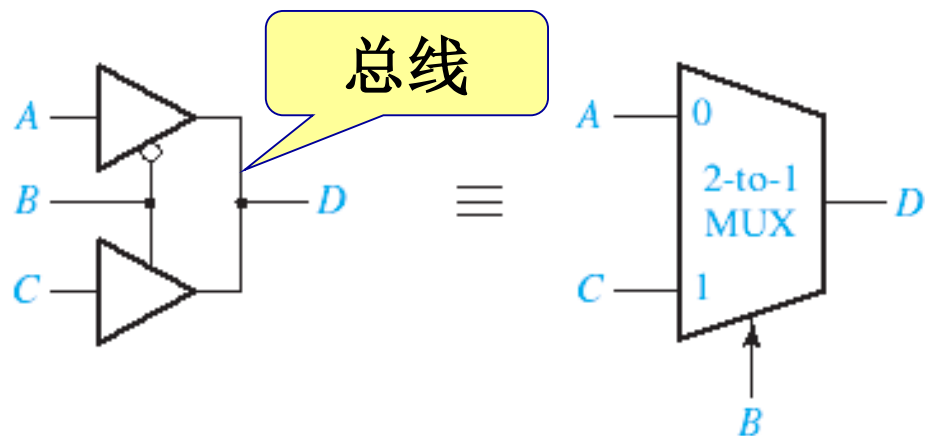
使能端是**控制信号输入端**，又叫**使能输入端**（enable），它是芯片的一个输入引脚，或者电路的一个输入端口，只有该引脚激活，芯片才能工作，若符号上面有一横/带有非号，则表示**低电平有效**；否则为**高电平有效**。



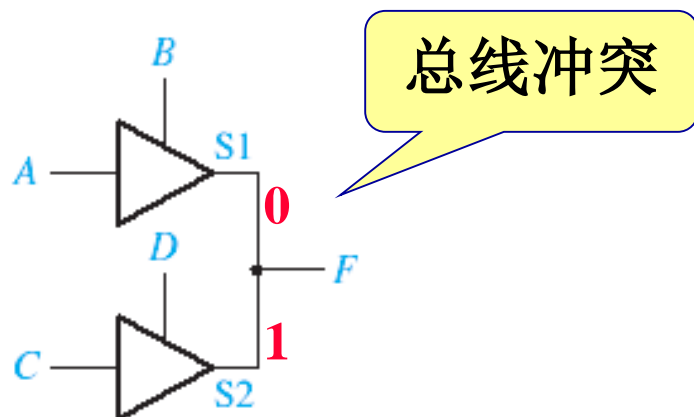
高有效和低有效



三态门中的总线冲突



$$D = B'A + BC$$

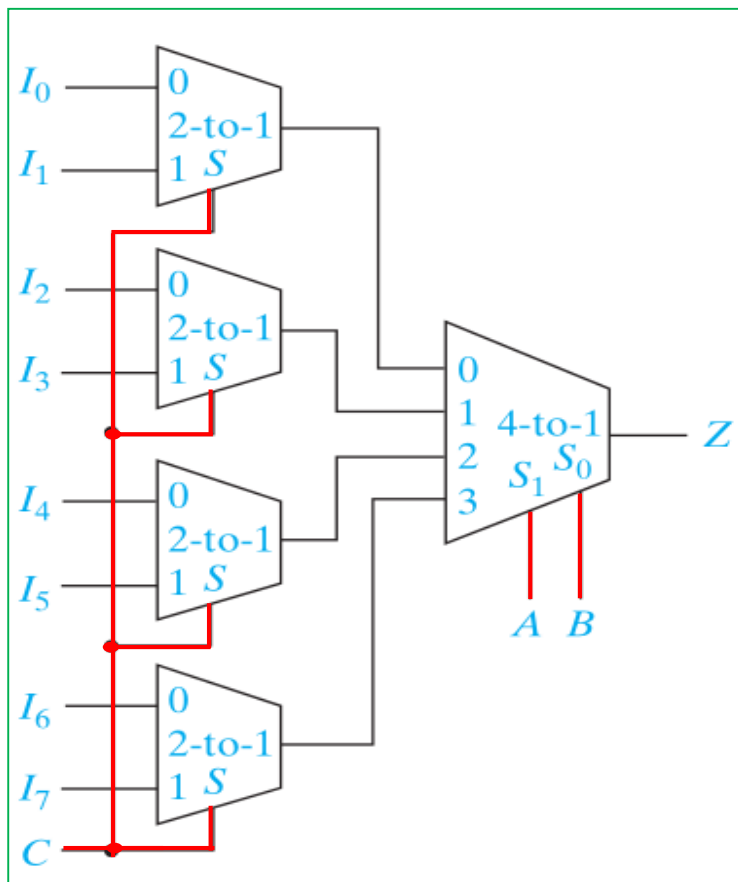


	S_2			
S_1	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

数据选择器

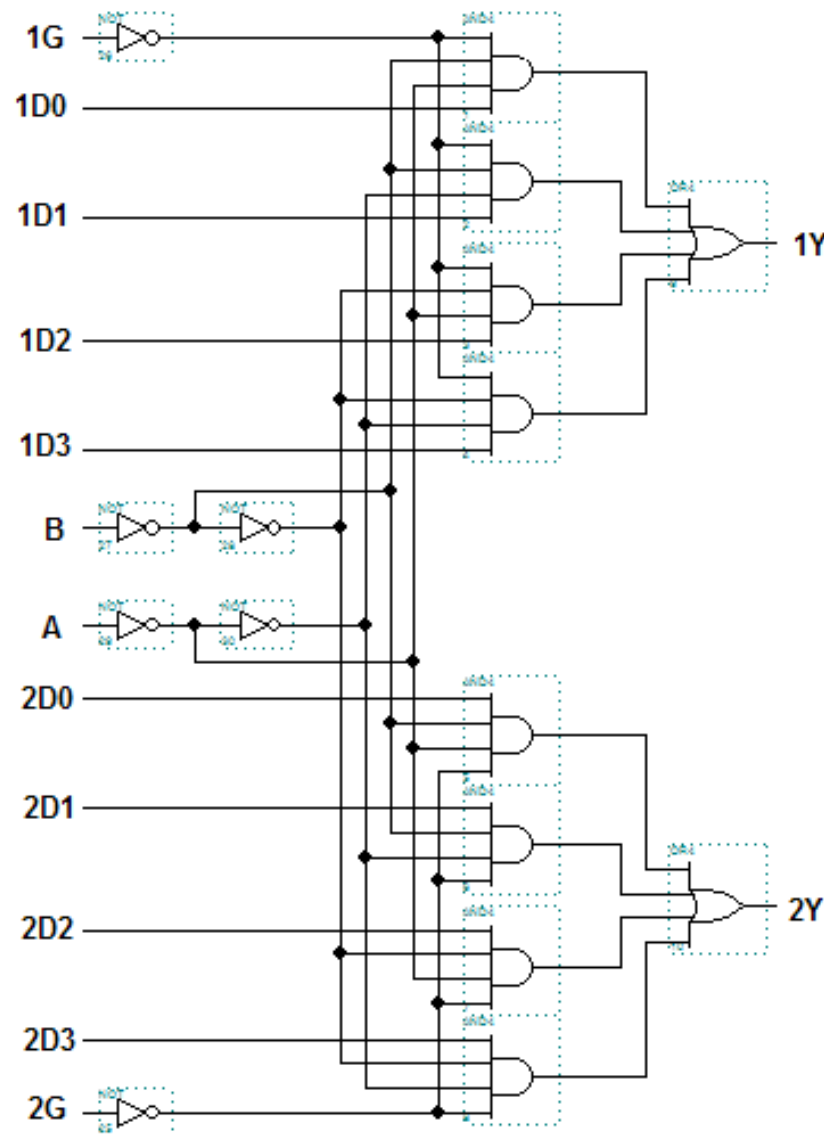
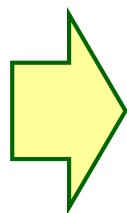
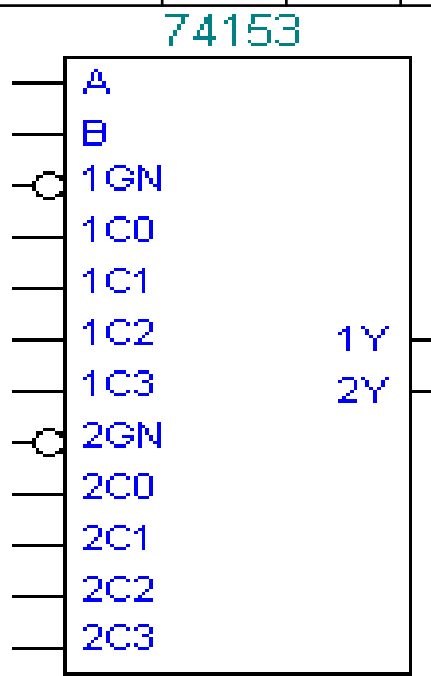
用数据选择器级联实现8选1。

- 能否用两个2选1MUX实现3选1？
- 能否用两个4选1和一个2选1实现8选1？
- 能否用8选1MUX实现一个4选1MUX？



双4选1典型器件74LS153

1Gn	2Gn	A	B	1Y	2Y
1	1	×	×	0	0
0	0	0	0	1C ₀	2C ₀
0	0	0	1	1C ₁	2C ₁
0	0	1	0	1C ₂	2C ₂
0	0	1	1	1C ₃	2C ₃



Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

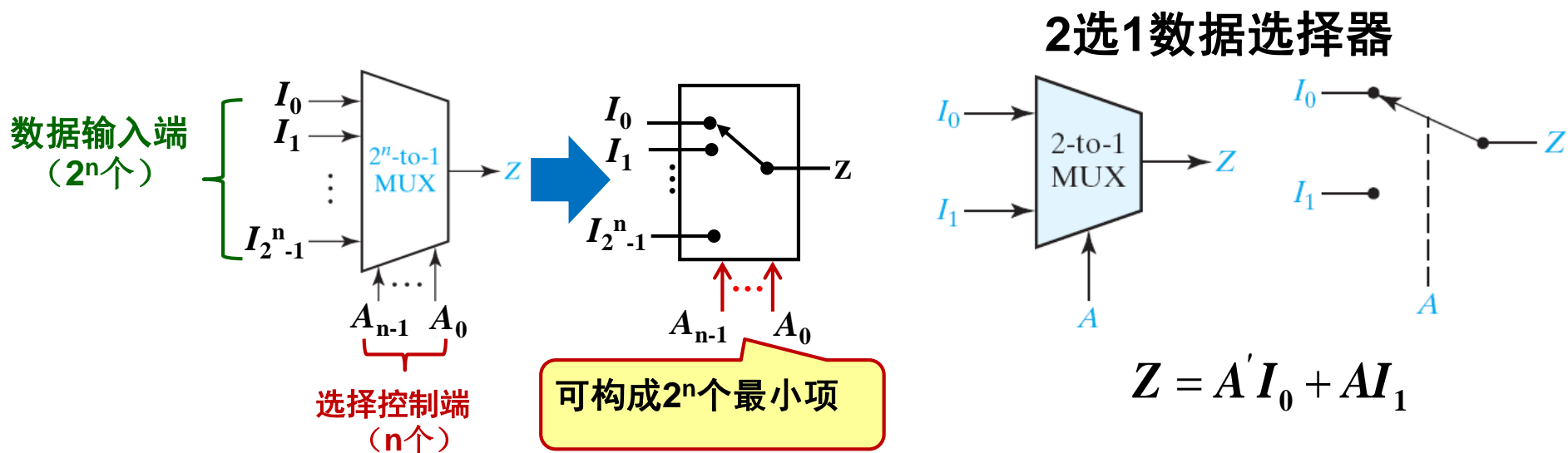
Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

集成电路的分类

分类	单芯片内集成的逻辑门数量	集成内容	器件封装	需要掌握的内容
小规模 (SSI)				
中规模 (MSI)				
大规模 (LSI)				
超大规模 (VLSI)				

数据选择器/多路开关



$$Z = \sum_{k=0}^{2^n-1} m_k I_k$$

控制端最小项 m_k 的序号 K , 指向了第 K 路数据输入端 I_k 。

m_k —— n 个控制变量的最小项

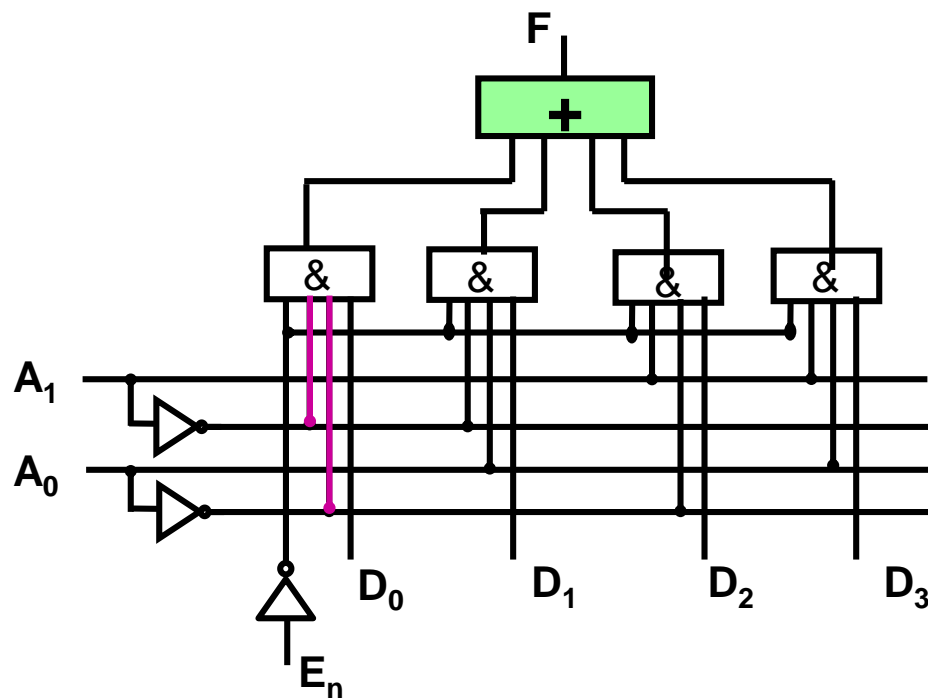
I_k —— 第 k 路数据输入

数据选择器的功能:

- ① 从多路输入中选择一个送往输出端 (2^n 选1);
- ② 选择哪一路输入送到输出端由控制信号决定;

用途: 实现多通道的数据传送;

4选1数据选择器

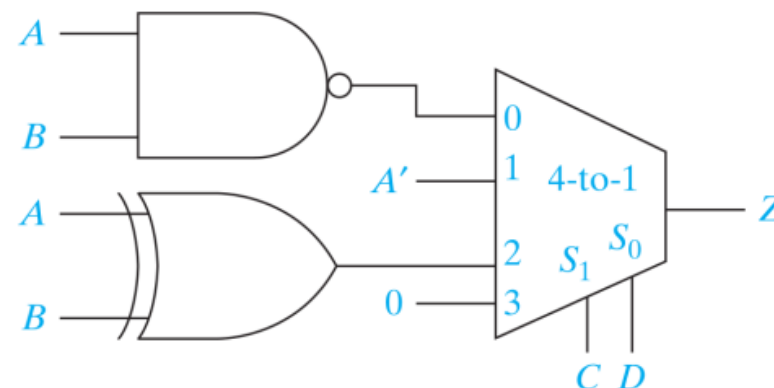


$$F = \bar{E}_n (D_0 \bar{A}_1 \bar{A}_0 + D_1 \bar{A}_1 A_0 + D_2 A_1 \bar{A}_0 + D_3 A_1 A_0)$$

E_n	A_1	A_0	F
1	×	×	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3

功能表

■ 典型应用——实现常规逻辑函数



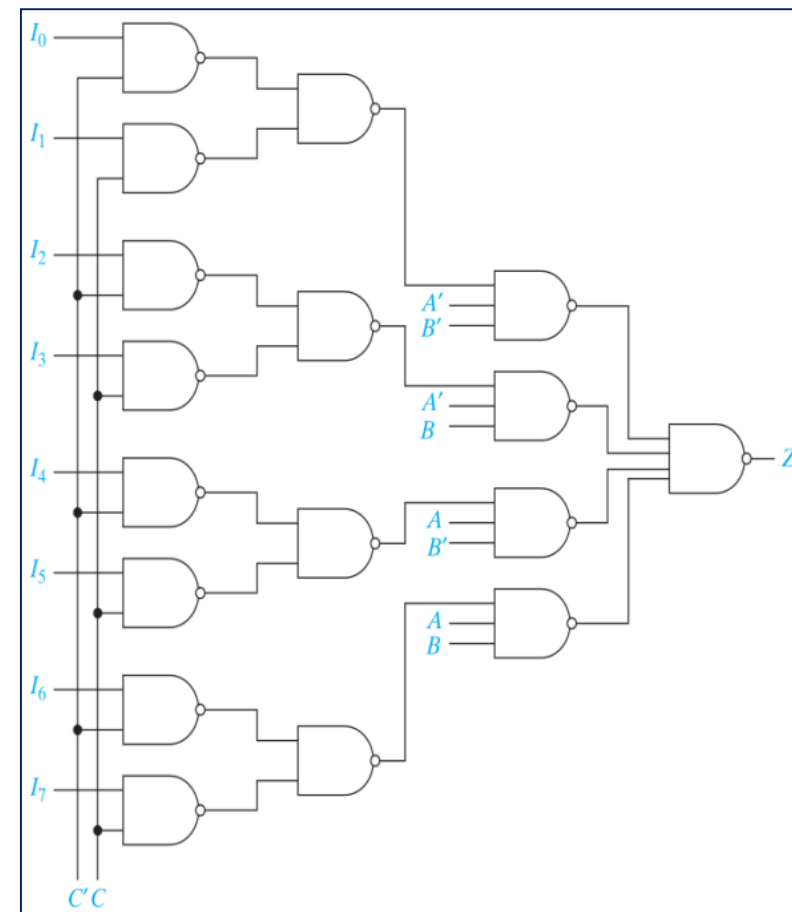
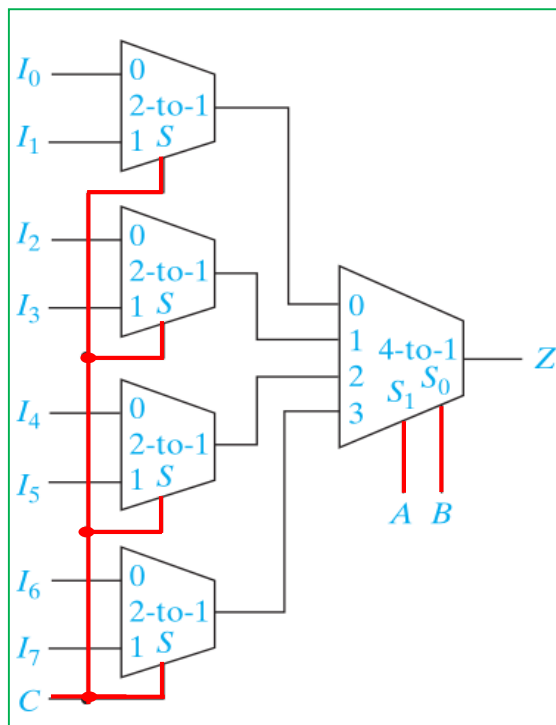
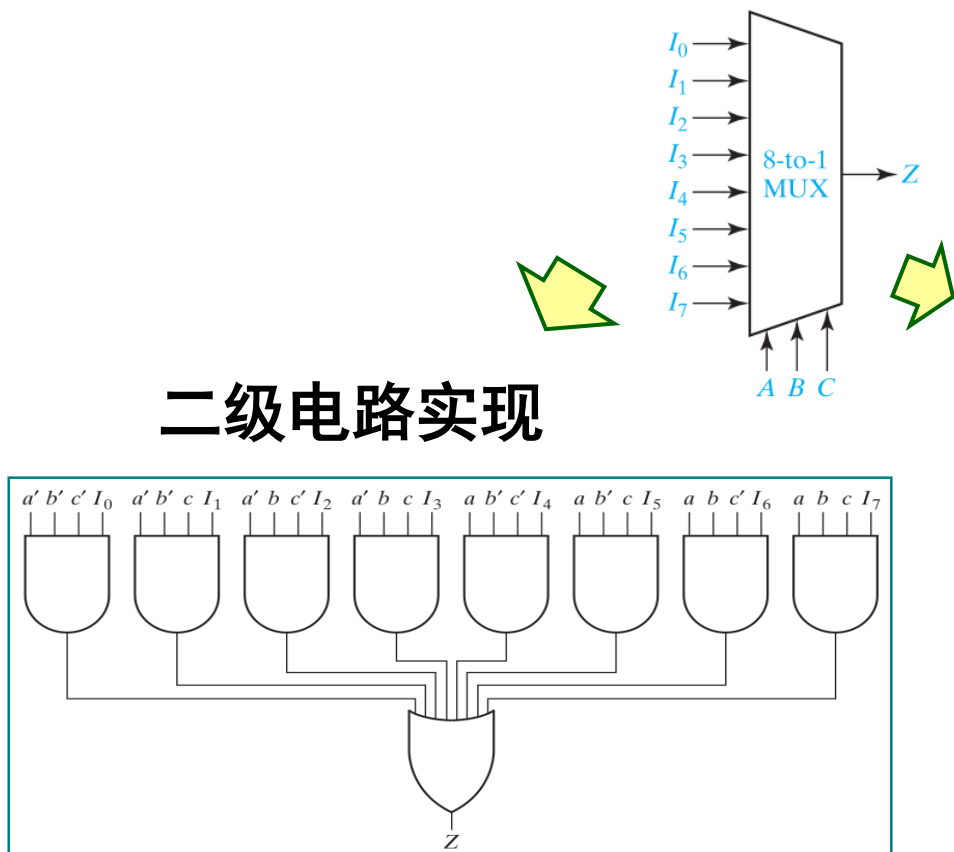
$$\begin{aligned} Z &= \bar{C}\bar{D}(\bar{A} + \bar{B}) + \bar{C}D\bar{A} + C\bar{D}(A\bar{B} + \bar{A}B) + CD(0) \\ &= \bar{A}\bar{C} + A\bar{B}\bar{D} + \bar{A}\bar{B}DB \odot C \end{aligned}$$

8选1数据选择器

单一逻辑门实现

数据选择器级联实现

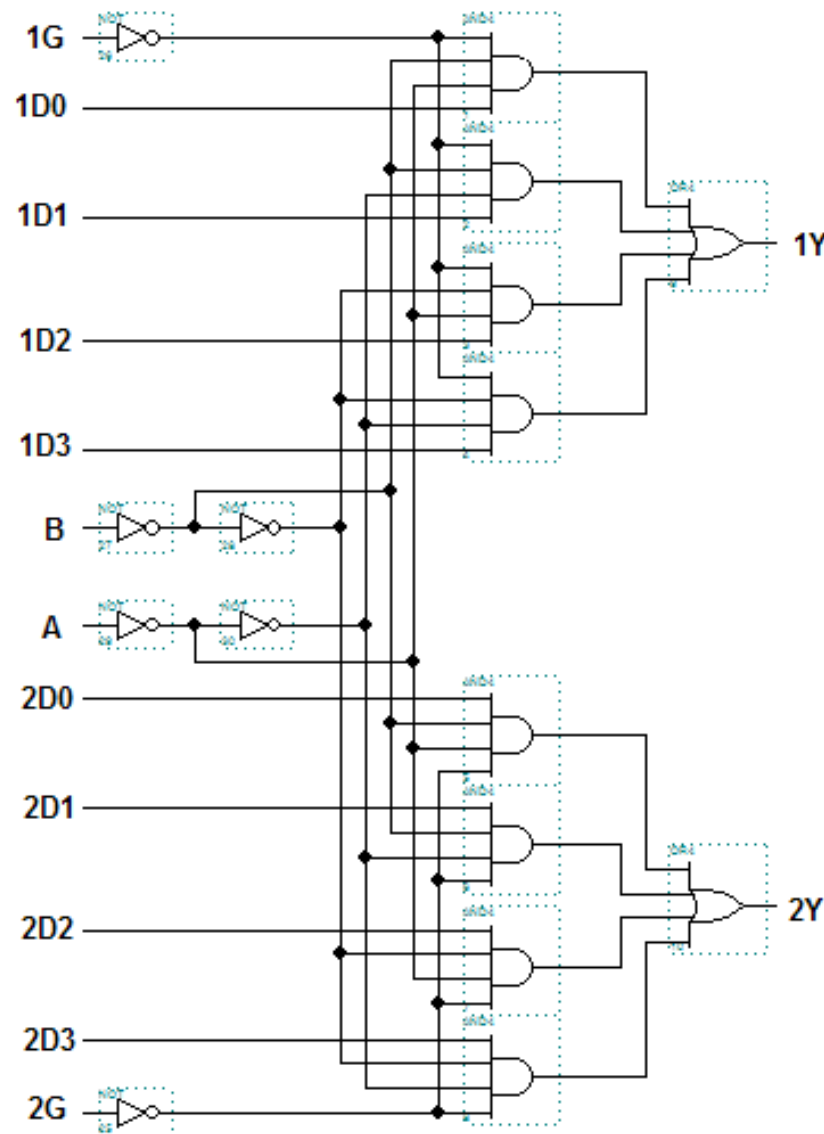
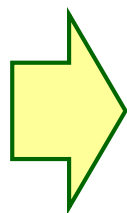
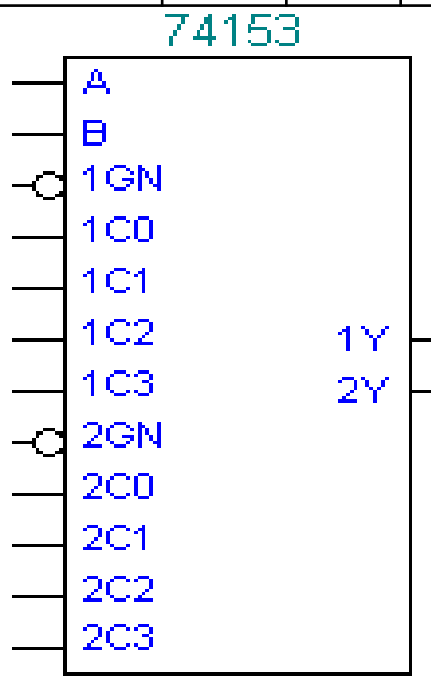
二级电路实现



$$Z = A'B'C'I_0 + A'B'CI_1 + A'BC'I_2 + A'BCI_3 + AB'C'I_4 + AB'CI_5 + ABC'I_6 + ABCI_7$$

双4选1典型器件74LS153

1Gn	2Gn	A	B	1Y	2Y
1	1	×	×	0	0
0	0	0	0	1C ₀	2C ₀
0	0	0	1	1C ₁	2C ₁
0	0	1	0	1C ₂	2C ₂
0	0	1	1	1C ₃	2C ₃



Unit 7 组合逻辑元件

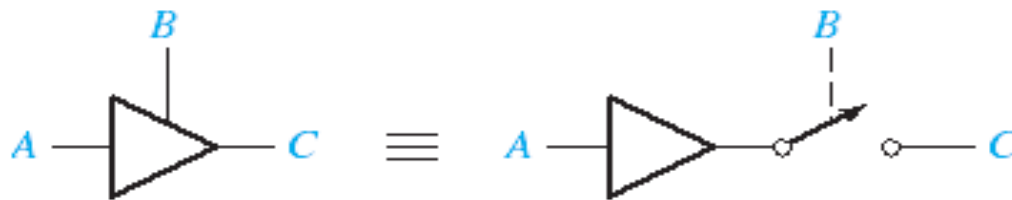
- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

三态门(Three-State Buffers)

三态——

- 0
- 1
- Z: 高阻态

- 包括三态恒等门、三态非门、三态与非门等，**缓冲器**（驱动门）。
- 用途之一：可用来增强输出驱动能力



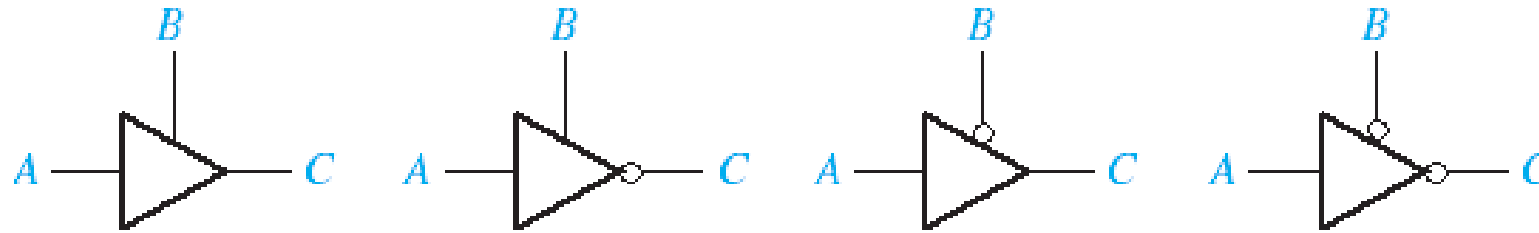
三态门（恒等）

B: 使能端，高电平有效

真值表

B	A	C
0	0	Z
0	1	Z
1	0	0
1	1	1

三态门 (Three-State Buffers)



B	A	C
0	0	Z
0	1	Z
1	0	0
1	1	1

(a)

B	A	C
0	0	Z
0	1	Z
1	0	1
1	1	0

(b)

B	A	C
0	0	0
0	1	1
1	0	Z
1	1	Z

(c)

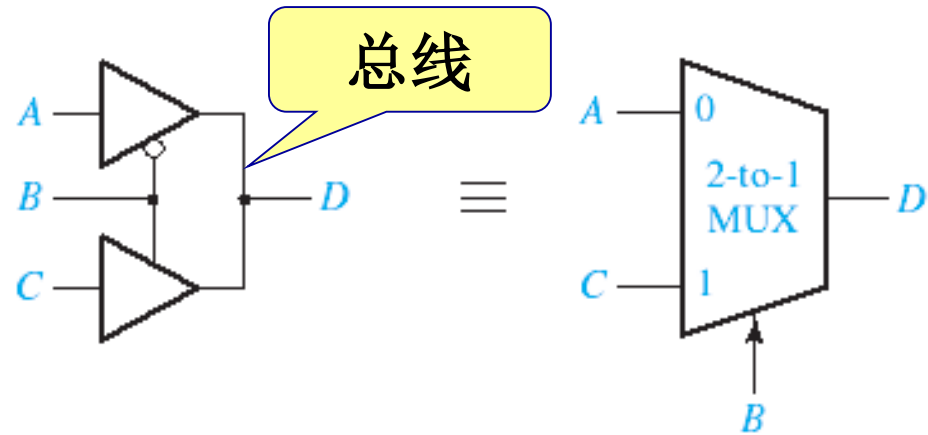
B	A	C
0	0	1
0	1	0
1	0	Z
1	1	Z

(d)

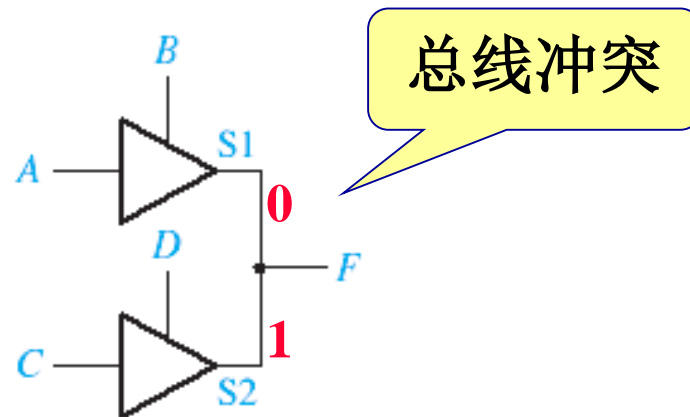
高阻态：电阻很大，相当于开路

高阻态相当于该门同与它连接的电路处于断开的状态。（实际电路中不可能去断开它）

三态门

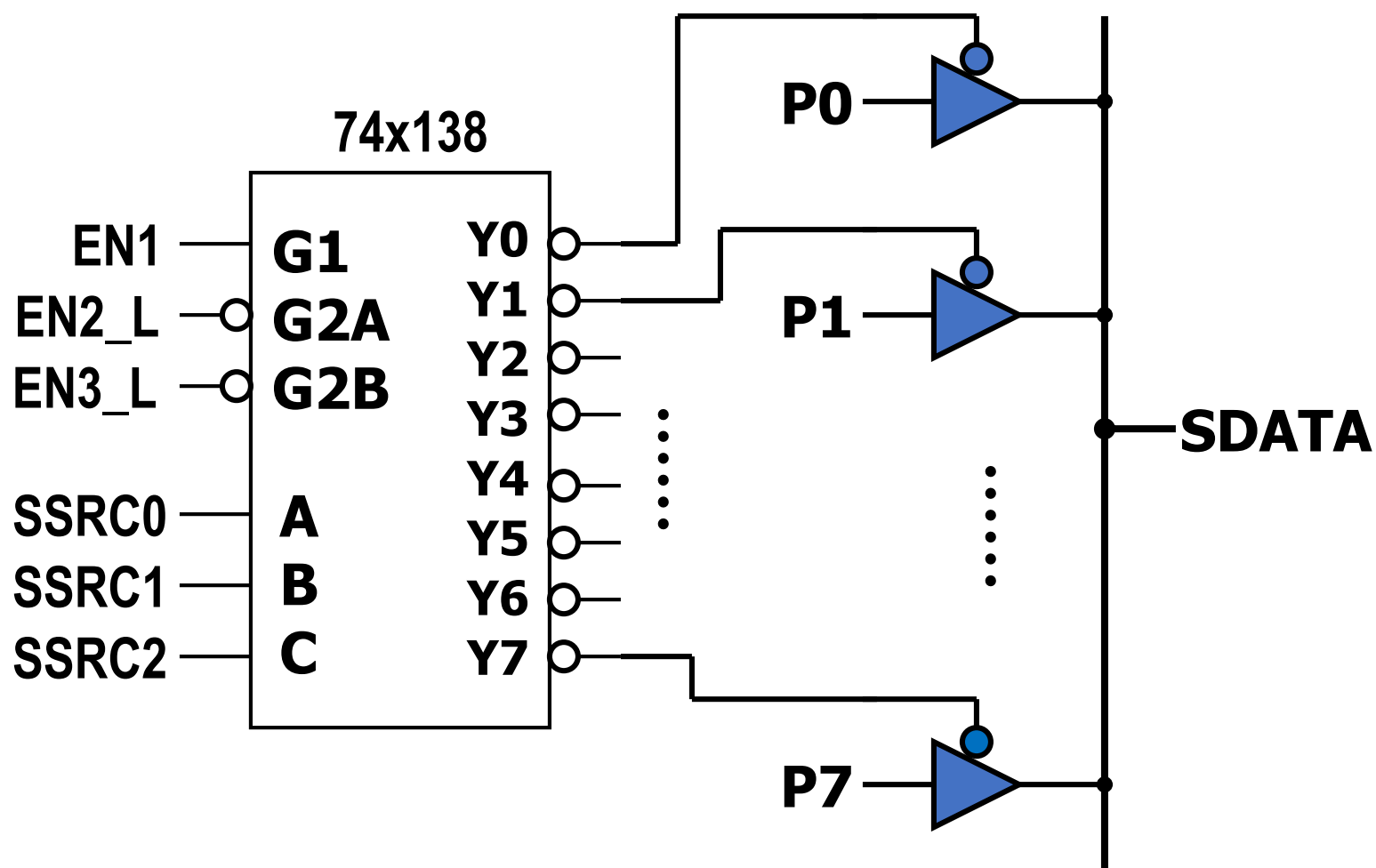


$$D = B'A + BC$$



	S_2			
S_1	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

三态门

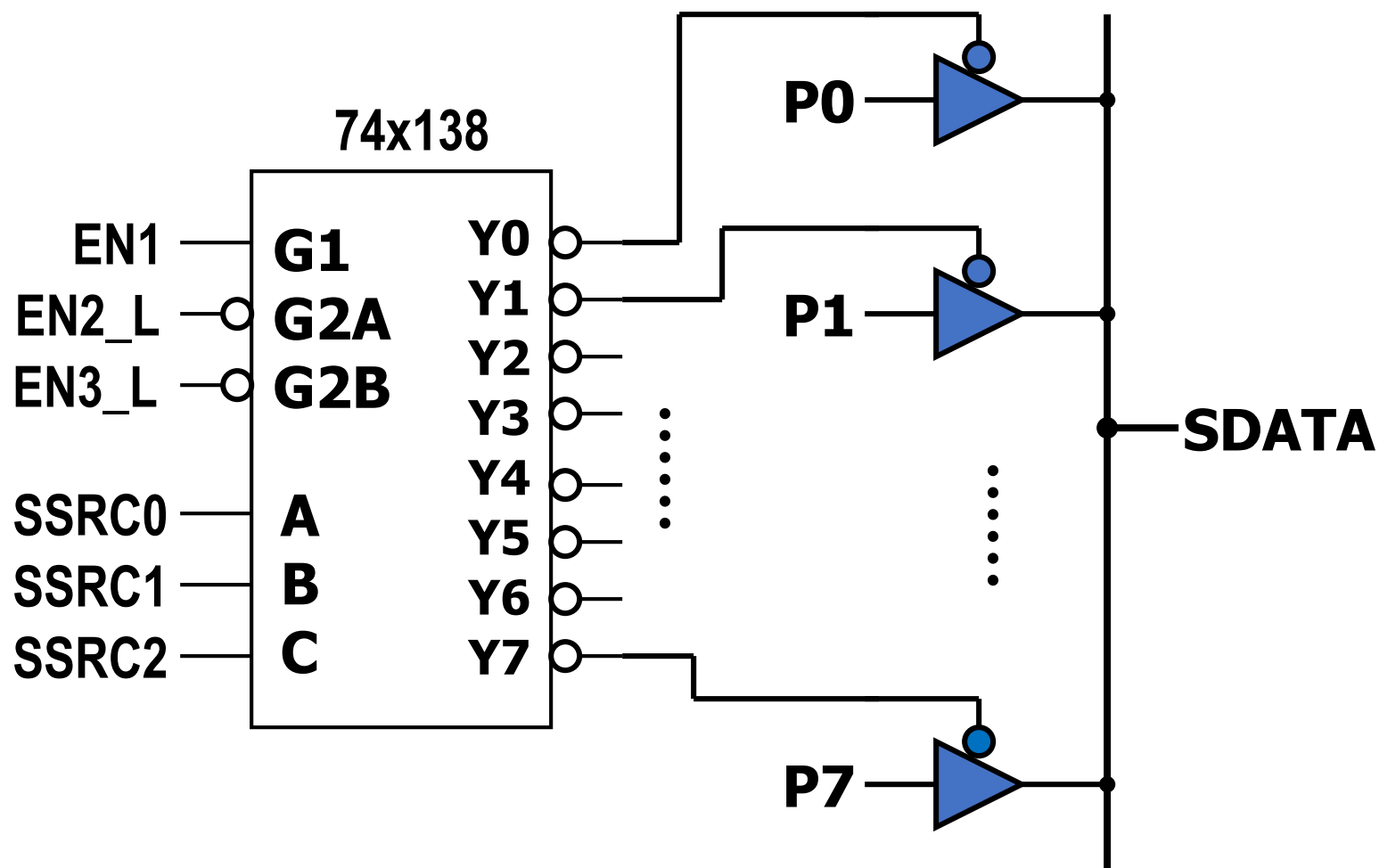


三态器件允许多个信号源共享单个“总线”（同线），但线上每次仅一个器件“谈话”

假如不是全部EN线有效，则没有一个三态缓冲器能被使能，此时SDATA上的逻辑值是“未定义”，悬空信号的实际电压值依赖于电路细节。

图 8个信号源共享1根三态总线/同线

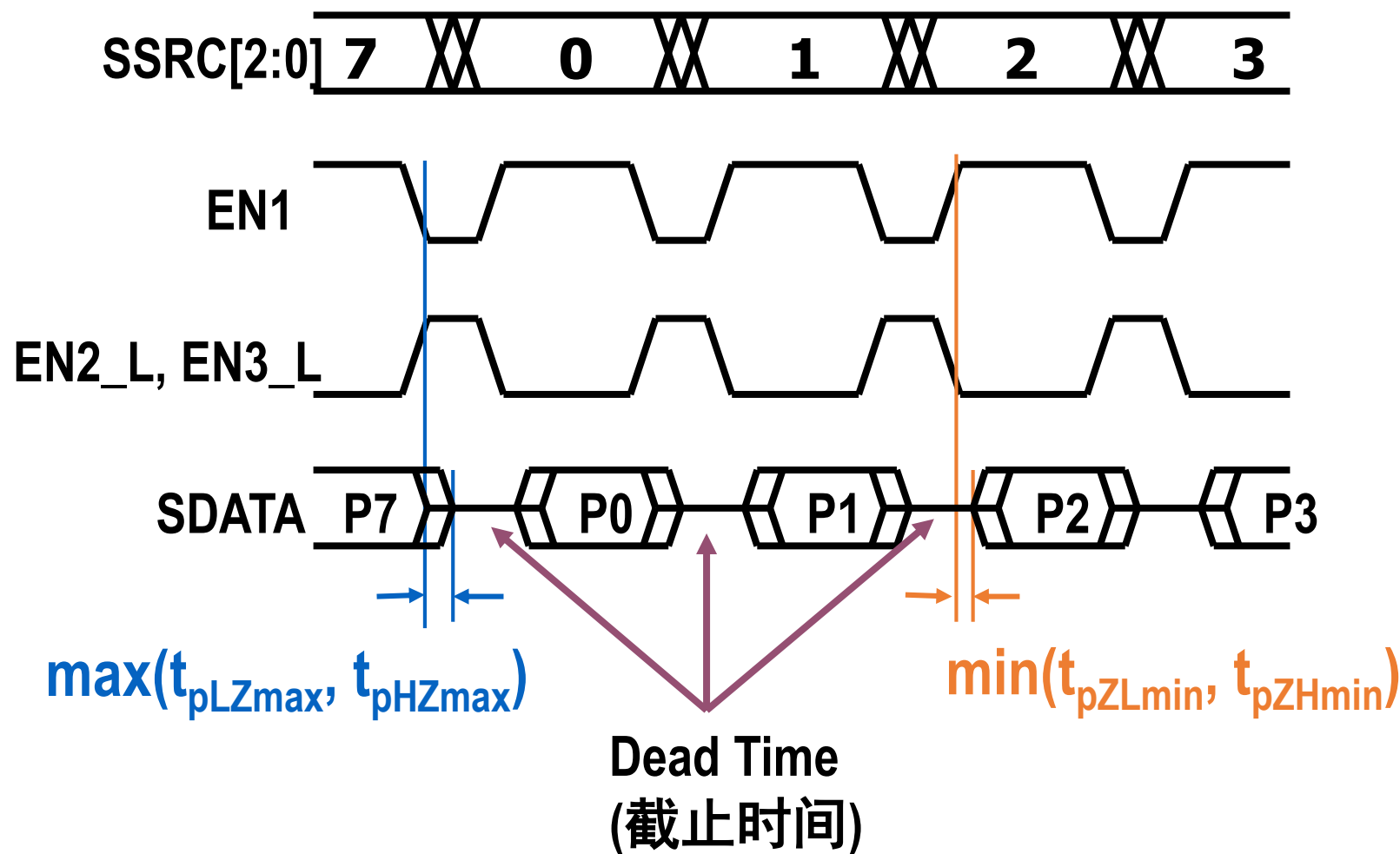
三态门



对典型的三态器件，**进入高阻态比离开高阻态快**。也会使得系统中产生冲突（fighting）

图 8个信号源共享1根三态总线

三态门

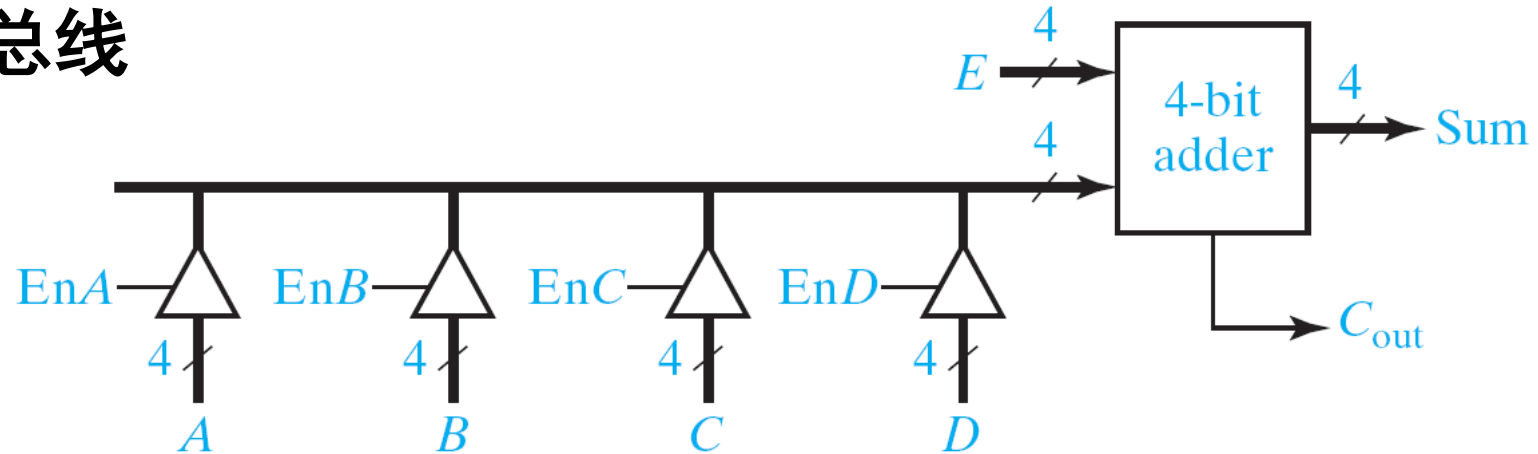


使用三态器件唯一真正安全的方法是**设计逻辑控制**，以保证同线上有一段**截止时间**（dead time），在此期间不应有任何器件驱动同线。

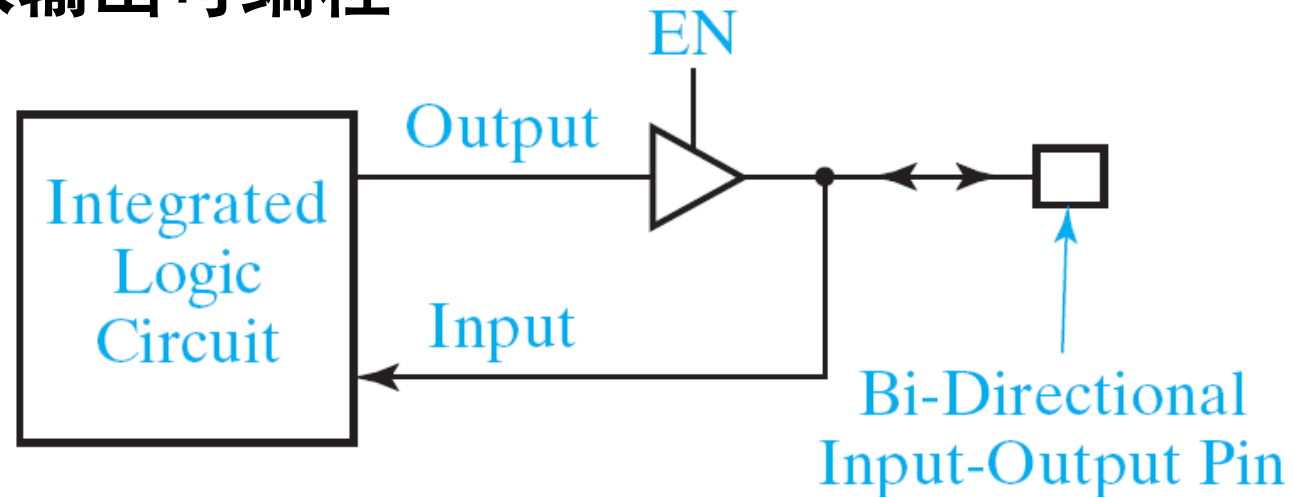
图 三态同线定时图

三态门应用

■ 三态总线

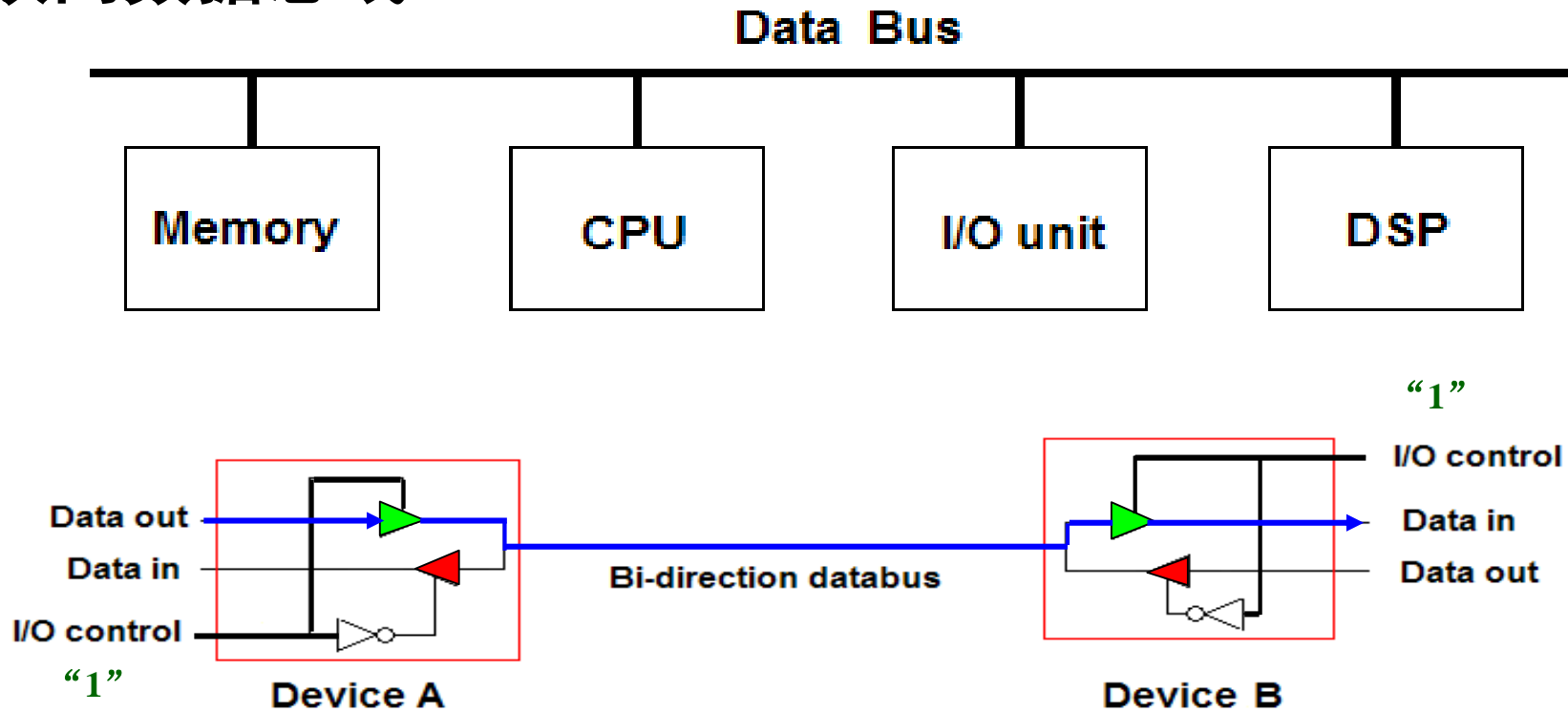


■ 管脚输入输出可编程



三态门应用——续

■ 双向数据总线



三态门应用——续

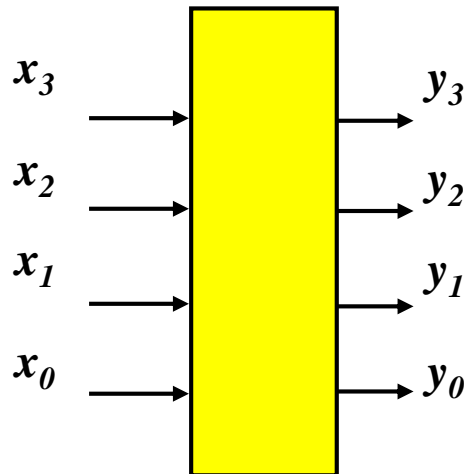
内存里的一个存储单元

- 读写控制线处于**低电位**时，可以写入；
- 读写控制线处于**高电位**时，可以读出
- 但是不读不写，就要用**高阻态**

三态门的应用

- $X=X_3X_2X_1X_0$ 为8421BCD码，设计一个MOD 5选择电路，要求选择那些能被5整除的数输出。

①真值表（F为控制信号）



控制信号：判断条件



三态门使能

X_3	X_2	X_1	X_0	F	X_3	X_2	X_1	X_0	F
0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	1	0	1	0	×
0	0	1	1	0	1	0	1	1	×
0	1	0	0	0	1	1	0	0	×
0	1	0	1	1	1	1	0	1	×
0	1	1	0	0	1	1	1	0	×
0	1	1	1	0	1	1	1	1	×

三态门的应用——续

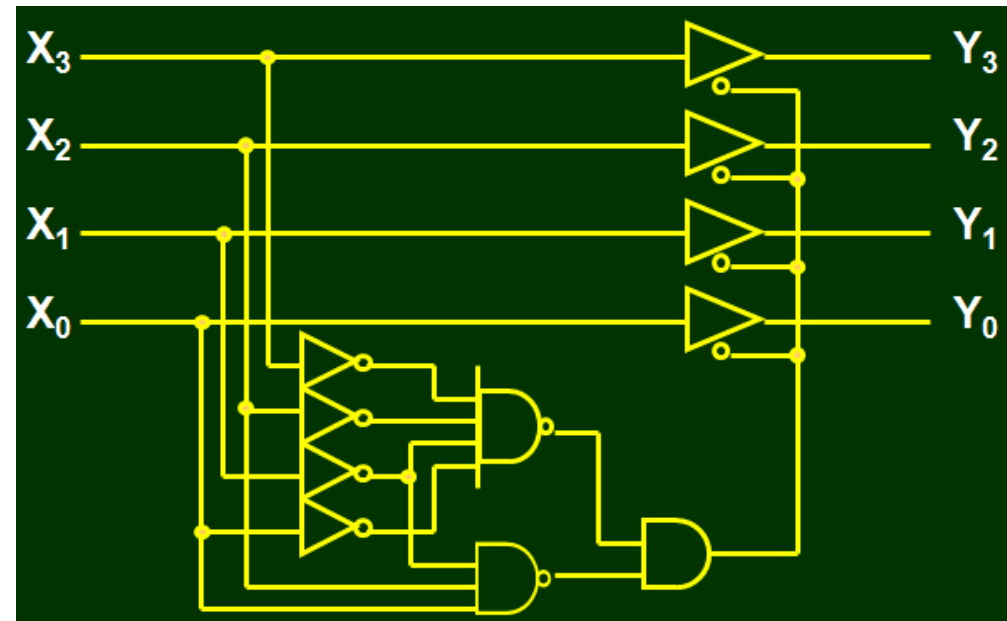
② 化简

x_1x_0					
x_3x_2		00	01	11	10
	00	1	0	0	0
	01	0	1	0	0
	11	×	×	×	×
	10	0	0	×	×

$$\begin{aligned} F &= \overline{\overline{X_2 \bar{X}_1 X_0} + \bar{X}_3 \bar{X}_2 \bar{X}_1 \bar{X}_0} \\ &= (\overline{X_2 \bar{X}_1 X_0}) (\overline{\bar{X}_3 \bar{X}_2 \bar{X}_1 \bar{X}_0}) \end{aligned}$$

$$\bar{F} = (\overline{X_2 \bar{X}_1 X_0}) (\overline{\bar{X}_3 \bar{X}_2 \bar{X}_1 \bar{X}_0})$$

③ 逻辑图

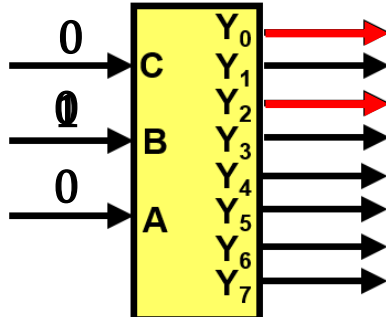


Unit 7 组合逻辑元件

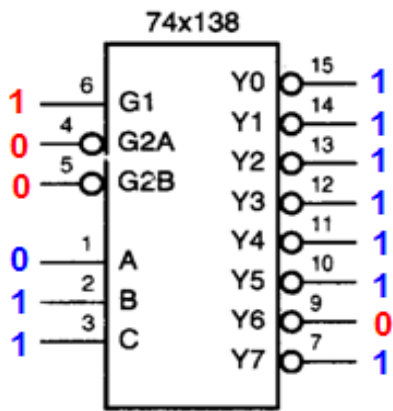
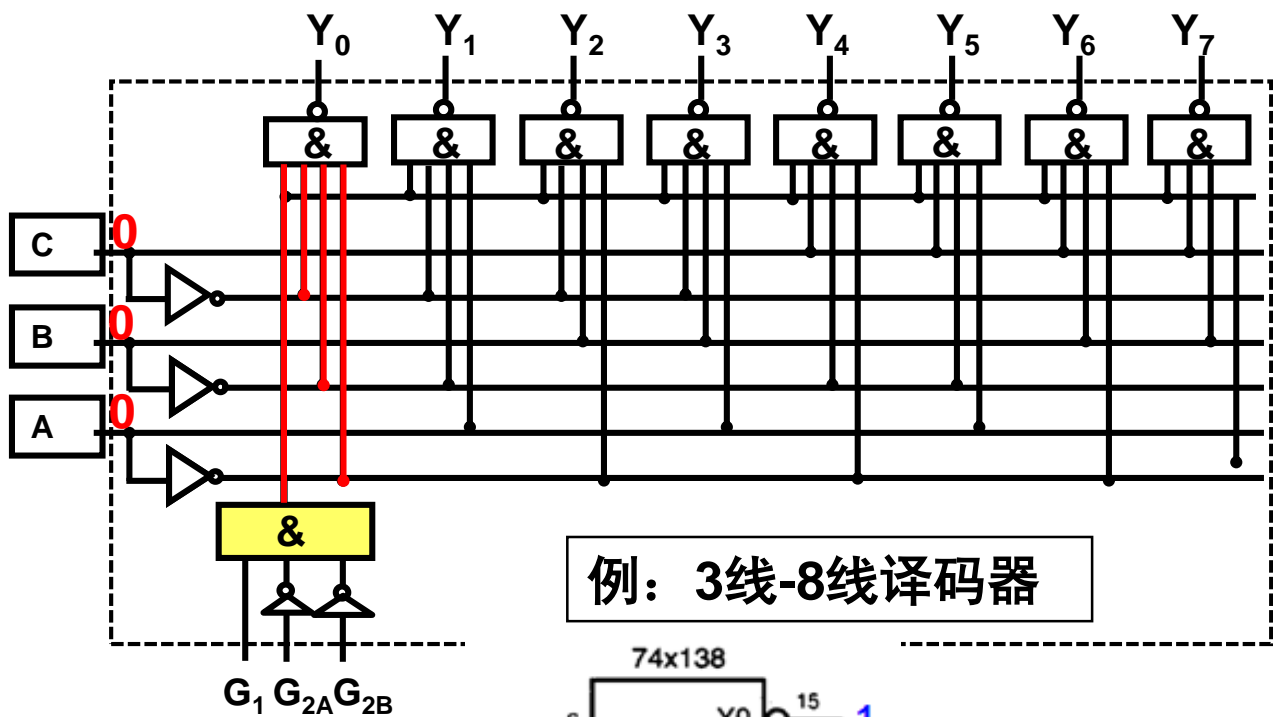
- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

译码器及分类

- ◆ 特点：多输入、多输出的组合逻辑电路
- ◆ 功能：将一种编码转换为另一种编码

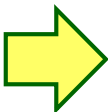
分类	特点	译码演示
二进制译码器		 (3线-8线译码器)
代码转换译码器		
显示译码器		

二进制译码器举例——3线-8线译码器



典型芯片: 74LS138

使能端			输入			译码输出							
G ₁	G _{2A}	G _{2B}	C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
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	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	1	0	1	1
1	0	0	1	0	1	1	1	1	1	1	1	0	1
1	0	0	1	1	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1	1	1	1	0

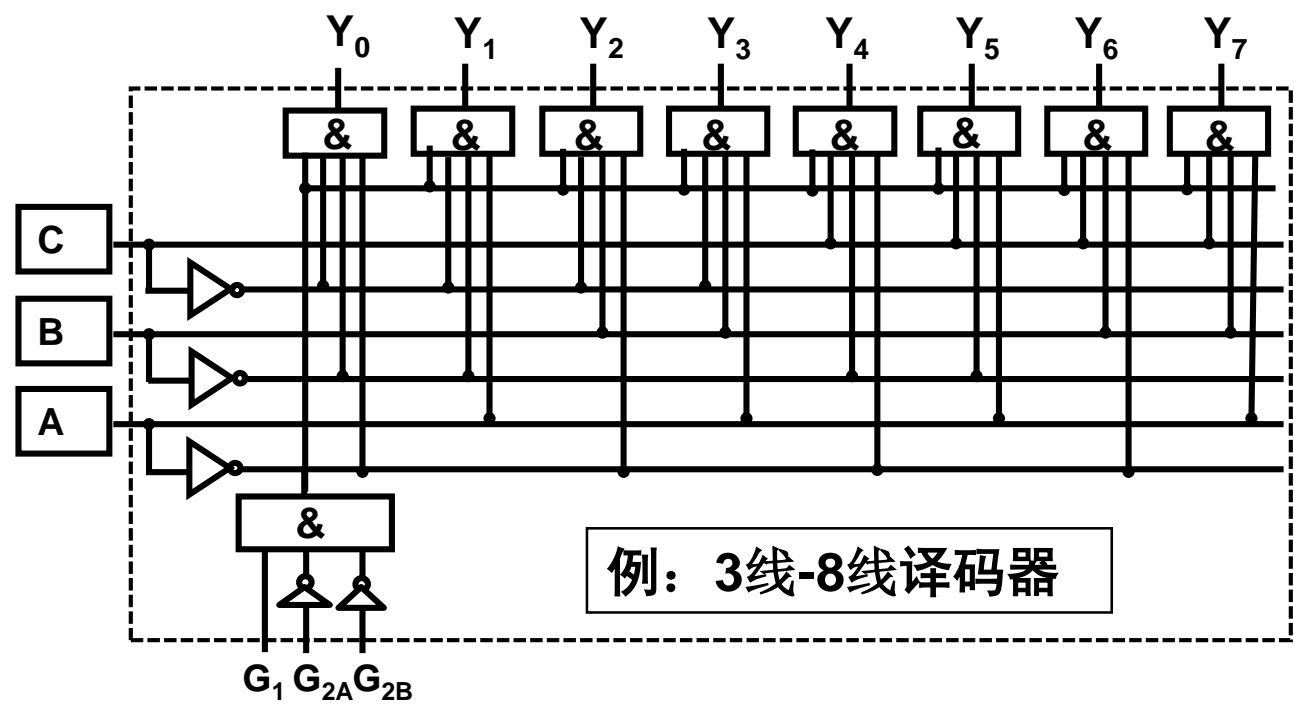


译码器输出：低电平有效

$$y_i = \bar{m}_i = M_i$$

3线-8线译码器

译码器输出：高电平有效 $\Rightarrow y_i = m_i$



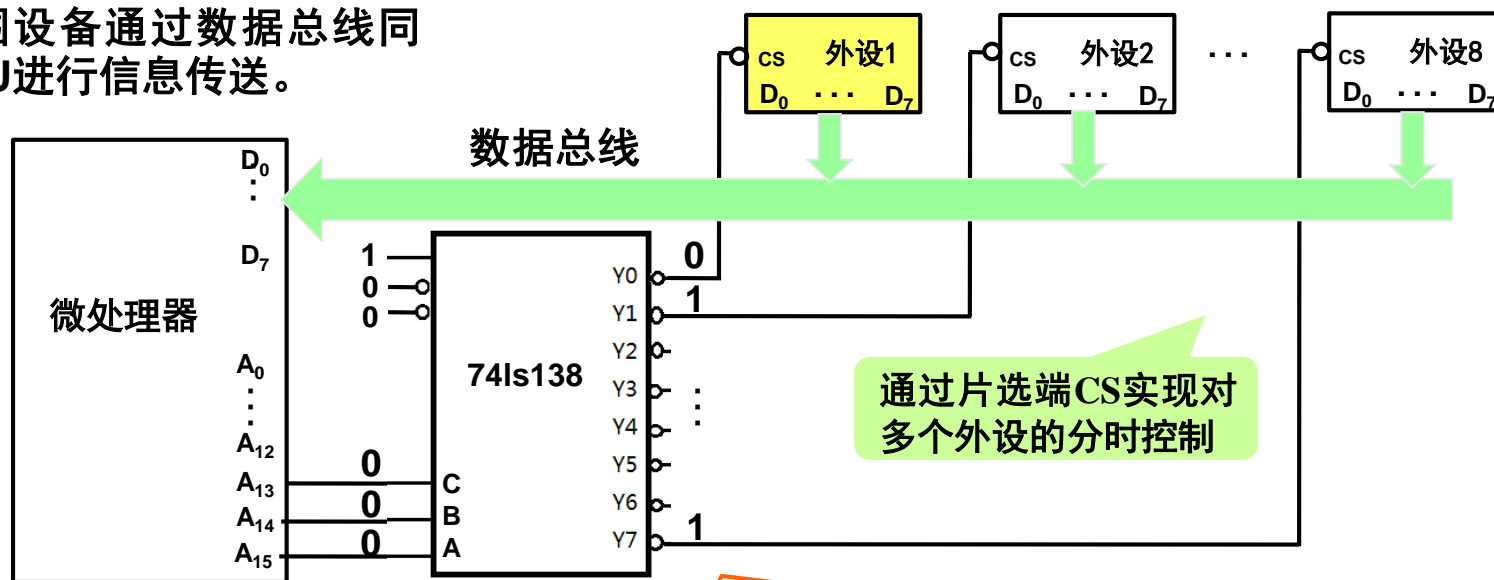
使能端			输入			译码输出							
G ₁	G _{2A}	G _{2B}	C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	X	X	X	X	X	0	0	0	0	0	0	0	0
X	1	X	X	X	X	0	0	0	0	0	0	0	0
X	X	1	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	1	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	0	1	0
1	0	0	1	1	1	0	0	0	0	0	0	0	1

二进制译码器的典型应用——地址译码

■ 微处理器的地址译码

*假设D0—D7连接到各个外设的低8位地址线。

外围设备通过数据总线同CPU进行信息传送。



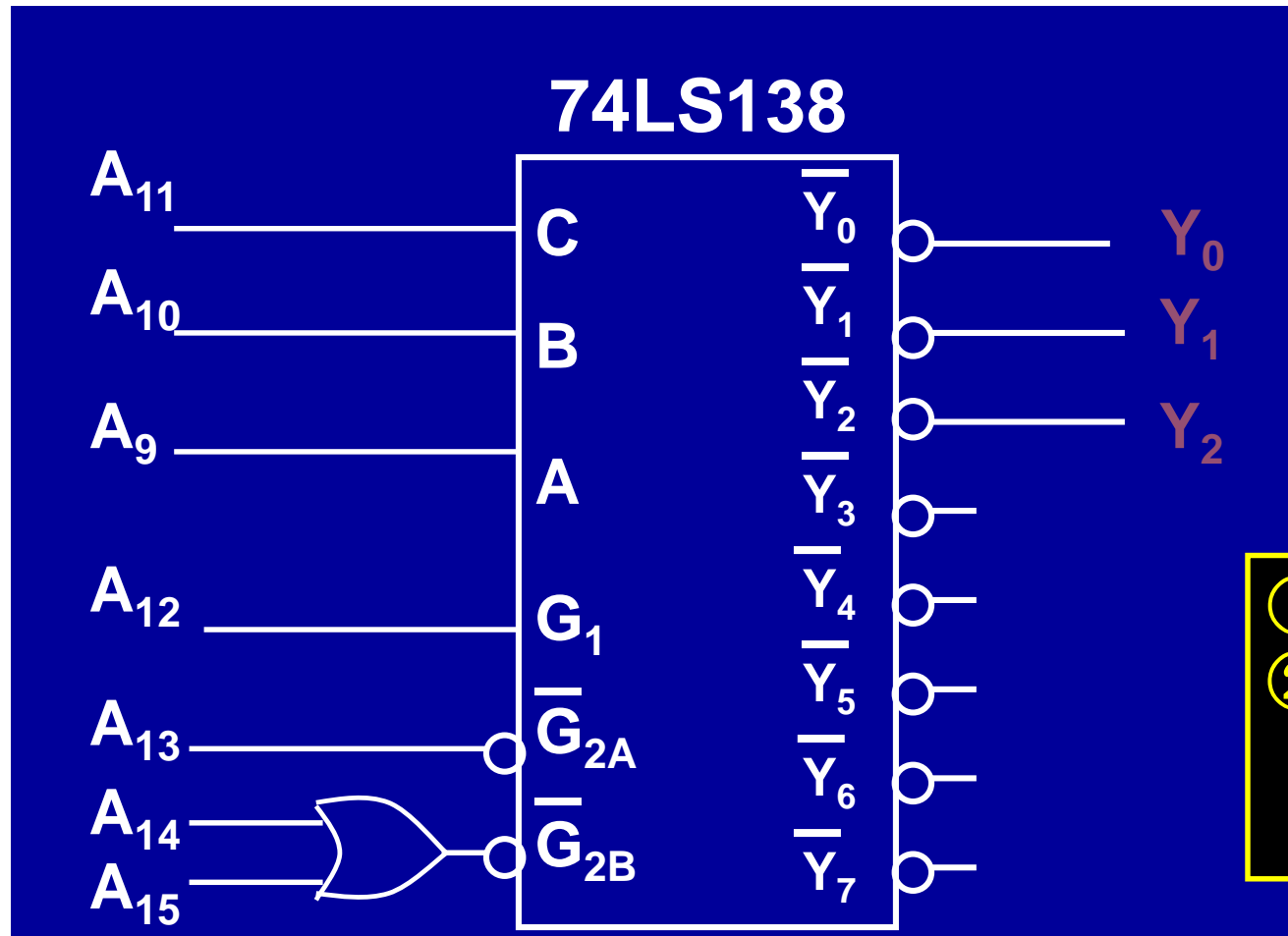
任意时刻 **N** 个外设中只有一个能被选中，其余外设的数据端均为高阻态。

通过片选端CS实现对多个外设的分时控制

利用二进制译码器 **N** 中选一的特点。

地址译码

- 图示电路的整个地址译码范围？各个外设的地址译码范围？

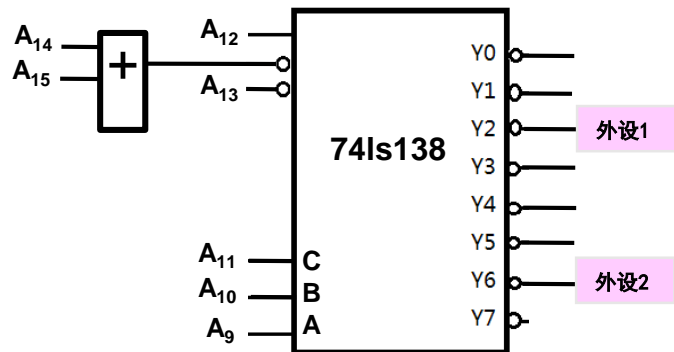


- ① 1000H~1FFFH
- ② Y_0 : 1000H~11FFH
 Y_1 : 1200H~13FFH
 Y_2 : 1400H~15FFH

1 整个地址译码范围为： [填空1]H— [填空2]H

2 外设1的地址译码范围为： [填空3]H— [填空4]H

3 外设2的地址译码范围为： [填空5]H— [填空6]H



作答

地址译码例题

■ 地址译码

图示电路的整个地址译码范围？

各个外设的地址译码范围？

整个译码器的地址译码范围：

A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	1	0	0	0	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

最小取值 1000H

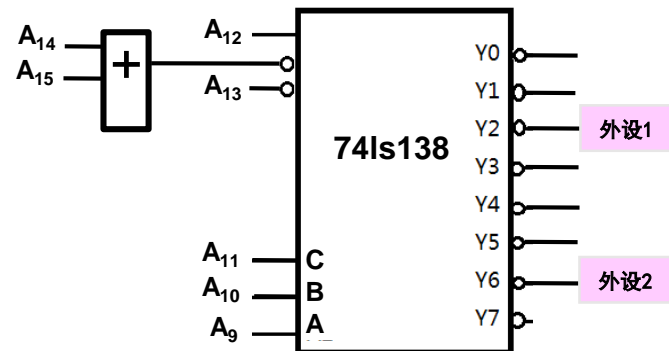
最大取值 1FFFH

外设1的地址译码范围：

A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	1	0	1	0	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1

最小取值 1400H

最大取值 15FFH



二进制译码器的典型应用——译码器级联

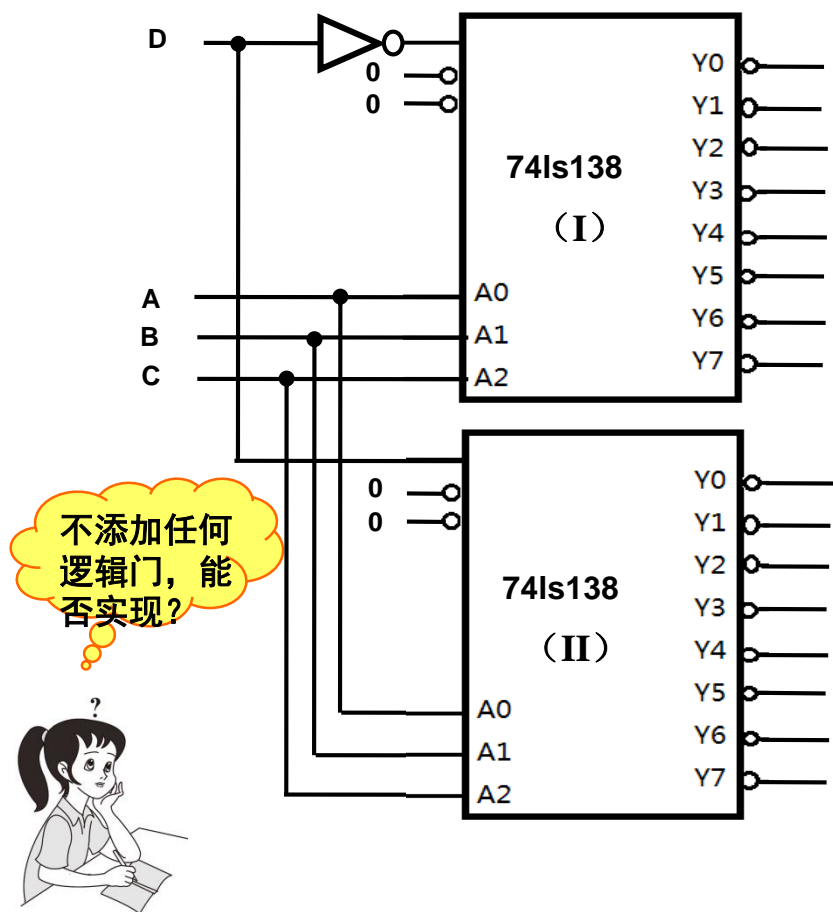
• 3线-8线译码器扩展为4线-16线译码

(I)

输入				译码输出							
D	C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0

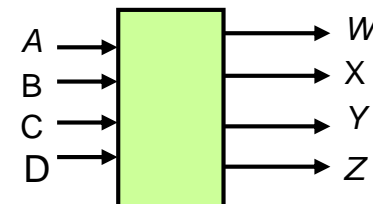
(II)

输入				译码输出							
D	C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
1	0	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	0	1	1	1	1	1	1
1	0	1	0	1	1	0	1	1	1	1	1
1	0	1	1	1	1	1	0	1	1	1	1
1	1	0	0	1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	0



代码转换译码器

- 例：设计一个译码器，
将输入的4位二进制数转换为典型格雷码



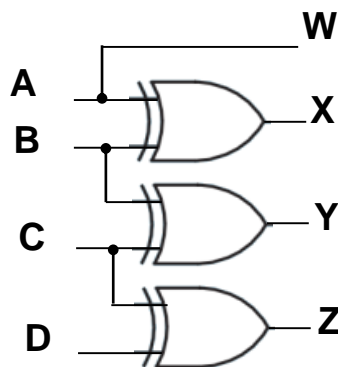
AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

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

$$\begin{cases} W = A \\ X = A \oplus B \\ Y = B \oplus C \\ Z = C \oplus D \end{cases}$$

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

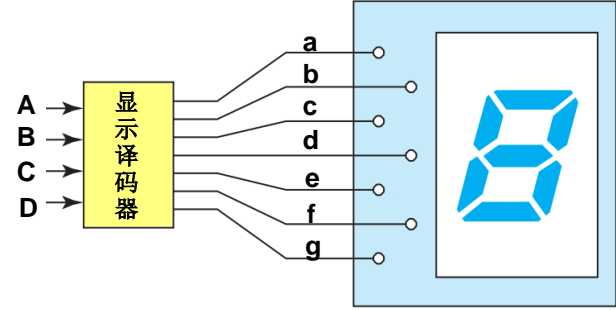
AB \ CD	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1



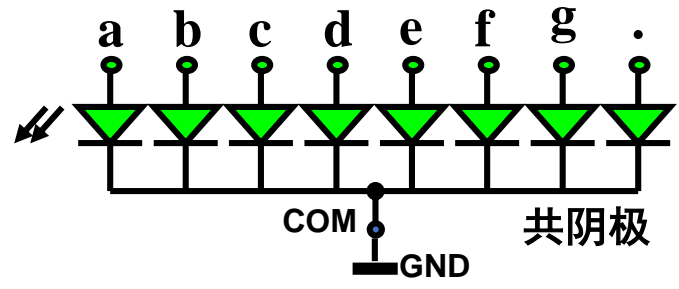
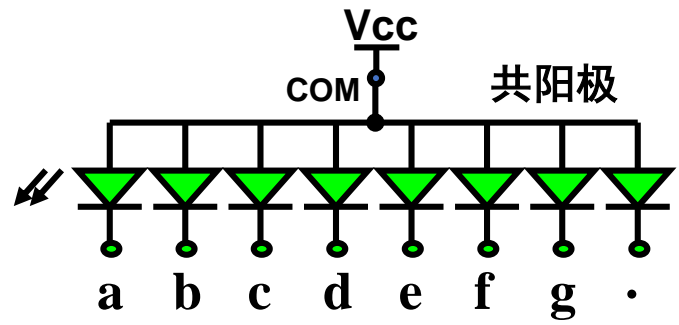
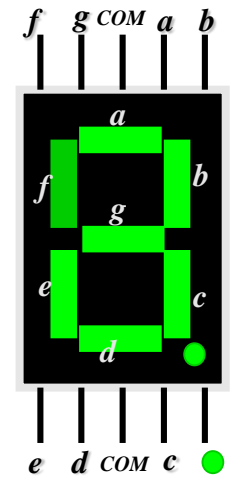
ABCD	WXYZ	ABCD	WXYZ
0000	0000	1000	1100
0001	0001	1001	1101
0010	0011	1010	1111
0011	0010	1011	1110
0100	0110	1100	1010
0101	0111	1101	1011
0110	0101	1110	1001
0111	0100	1111	1000

七段显示译码器

显示译码器：与显示器件（如数码管）配合，将输入代码转换为十进制码或特定编码，并在显示器件上显示相应的字形



七段数码管



8421BCD码驱动的共阴极七段数码管显示译码器功能表

输入				译码输出						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

a

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

$$a = A + C + BD + BD$$

b

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

$$b = B + CD + CD$$

c

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

$$c = B + C + D$$

d

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

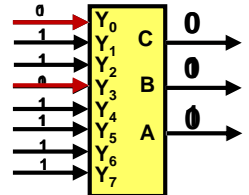
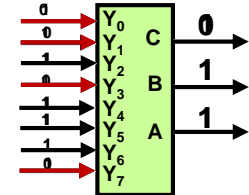
$$d = A + CD + BC + BD + BCD$$

Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

编码器

- ◆ 特点：多输入、多输出的组合逻辑电路
- ◆ 功能：将二进制码按照一定规律编排，使其具有特定含义，与译码器互逆。

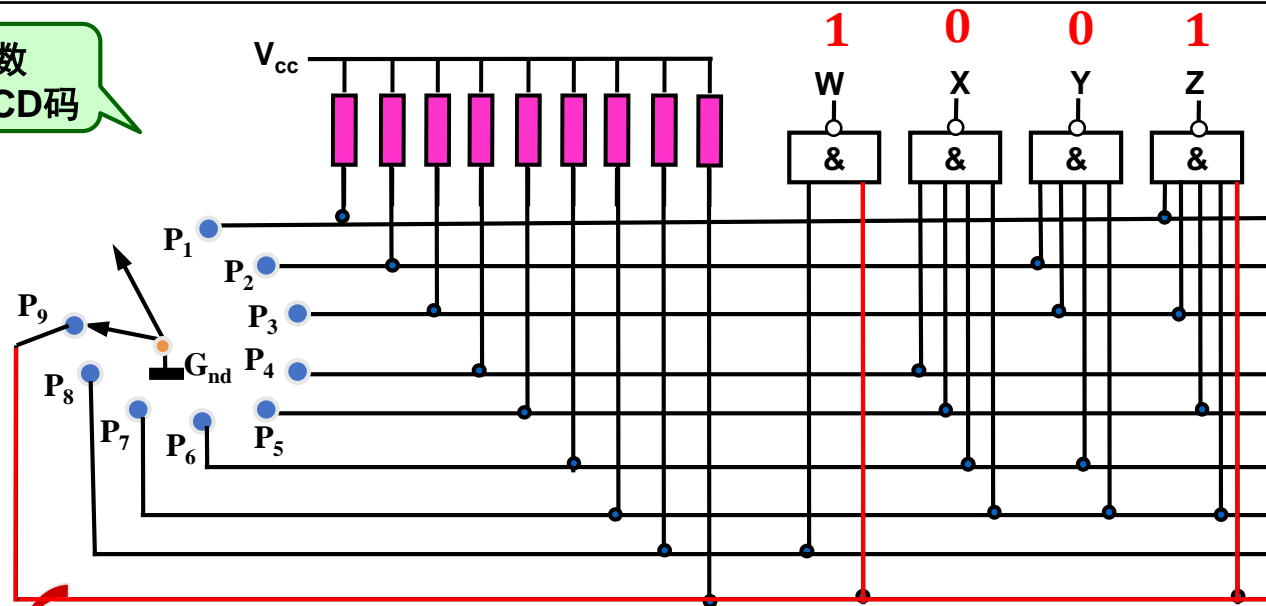
常用编码器	特点	编码演示
普通编码器 (二进制编码器)	输入： N 位，任何时刻 N 根输入线中只能有一个输入有效， N ($N=2^n$) 中取一。 输出： n 位二进制码	 (8 线-3 线编码器)
优先编码器	允许同时输入两个以上的有效编码输入信号，优先编码器能按照预先设定的优先级别，只对其中优先级最高的输入进行编码。	 (8 线-3 线优先编码器)

二进制编码器例子——键盘编码器

输入：十进制数
输出：8421BCD码

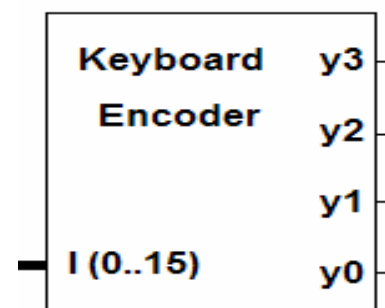
键盘编码器功能表

$P_9 \dots P_1$	按键	WXYZ
11111111	0	0000
11111110	1	0001
11111101	2	0010
111111011	3	0011
111110111	4	0100
111101111	5	0101
111011111	6	0110
110111111	7	0111
101111111	8	1000
011111111	9	1001



扩展

7	8	9	/
4	5	6	*
1	2	3	-
0	.	+	=



0	I0	0000	8	I8	1000
1	I1	0001	9	I9	1001
2	I2	0010	/	I10	1010
3	I3	0011	*	I11	1011
4	I4	0100	-	I12	1100
5	I5	0101	+	I13	1101
6	I6	0110	.	I14	1110
7	I7	0111	=	I15	1111

$$W=(P_8 \cdot P_9)'$$

$$Y=(P_2 \cdot P_3 \cdot P_6 \cdot P_7)'$$

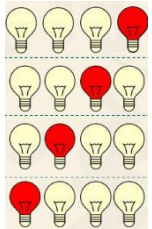
$$X=(P_4 \cdot P_5 \cdot P_6 \cdot P_7)'$$

$$Z=(P_1 \cdot P_3 \cdot P_5 \cdot P_7 \cdot P_9)'$$

二进制编码器——例：4线-2线编码器

4:2编码器

抢答器输出



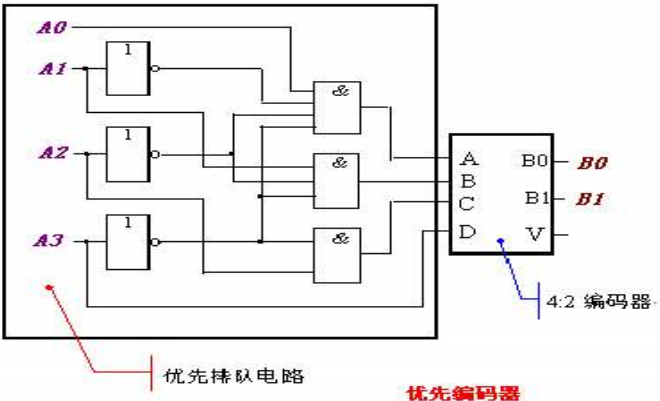
A ₃	A ₂	A ₁	A ₀	B ₁	B ₀
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

计算机配有四个外部设备：声卡(A0)，硬盘驱动器(A1)，鼠标(A2)，网卡(A3)，B₀、B₁为编码输出。



某一时刻只允许输入一个编码信号，如A₁ (A₁=1) 向 CPU 请求传送数据，CPU 根据接收的编码 B₁B₀ = 01，启动硬盘驱动器，开始传送数据。

普通编码器缺点：无法避免错误输入（同时输入多路有效信号），容易造成混乱。



4:2优先编码器

A ₃	A ₂	A ₁	A ₀	B ₁	B ₀
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

$$A = A_0 \overline{A_1} \overline{A_2} \overline{A_3}$$

$$B = A_1 \overline{A_2} \overline{A_3}$$

$$C = A_2 \overline{A_3}$$

$$D = A_3$$

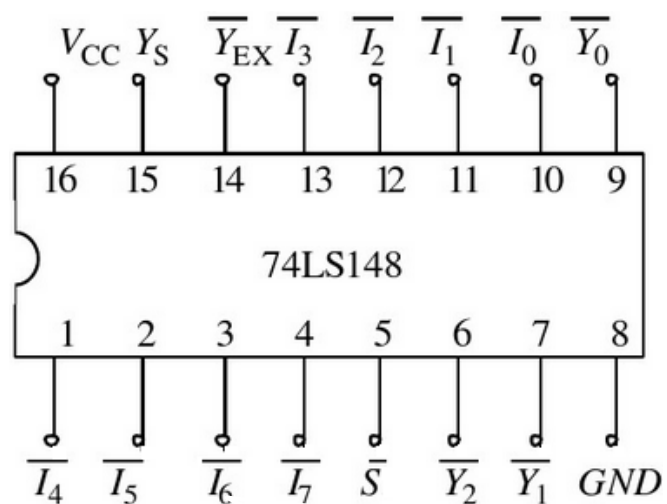
二进制编码器：

- 可以对 2^n 个输入对象编码
- 只需 n 个输出端（每个对象获得一个 n 位编码）
- 编码具有唯一性

优先编码器：

- 允许同时输入多路有效信号
- 按照预先设定的优先级，只对其中优先级最高的输入进行编码。

编码器典型芯片74LS148



输入和输出均为**低电平有效**。

输入使能

8线-3线优先编码器

多级级联

输出使能

输 入									输 出				
\overline{S}	\overline{I}_7	\overline{I}_6	\overline{I}_5	\overline{I}_4	\overline{I}_3	\overline{I}_2	\overline{I}_1	\overline{I}_0	\overline{Y}_2	\overline{Y}_1	\overline{Y}_0	\overline{Y}_{EX}	Y_S
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	×	×	×	×	×	×	×	0	0	0	0	1
0	1	0	×	×	×	×	×	×	0	0	1	0	1
0	1	1	0	×	×	×	×	×	0	1	0	0	1
0	1	1	1	0	×	×	×	×	0	1	1	0	1
0	1	1	1	1	0	×	×	×	1	0	0	0	1
0	1	1	1	1	1	0	×	×	1	0	1	0	1
0	1	1	1	1	1	1	0	×	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

标志位：

0：编码输出；1：非编码输出

编码器与译码器的实际应用



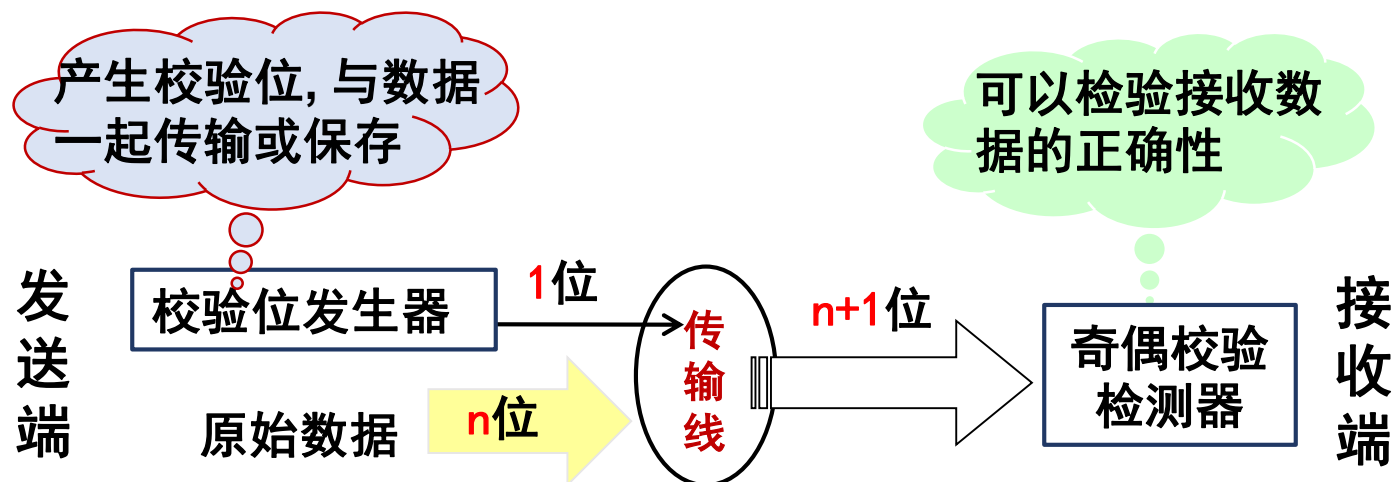
Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

奇偶校验器

- 用来检查数据传输和存取过程中是否产生错误的组合逻辑电路。
(就是检测数据中包含“1”的个数是奇数还是偶数)
- 广泛用于计算机的内存储器以及磁盘等外部设备中

{ 奇偶校验发生器：可产生奇偶校验位，与数据一起传输或保存
奇偶校验检测器：可以检验所接受数据的正确性



被校验的原始数据和1位校验位组成 $n+1$ 位校验码。



校验码: $n+1$ 位

偶校验位逻辑值的表达式:

$$P_E = A_3 \oplus A_2 \oplus A_1 \oplus A_0$$

奇校验位逻辑值的表达式:

$$P_O = A_3 \oplus A_2 \oplus A_1 \oplus A_0$$

偶校验位逻辑值电路是在奇校验位逻辑值电路输出端加非门实现

4位二进制数校验码真值表

$A_3 A_2 A_1 A_0$	P_E	P_O
0 0 0 0	0	1
0 0 0 1	1	0
0 0 1 0	1	0
0 0 1 1	0	1
0 1 0 0	1	0
0 1 0 1	0	1
0 1 1 0	0	1
0 1 1 1	1	0
1 0 0 0	1	0
1 0 0 1	0	1
1 0 1 0	0	1
1 0 1 1	1	0
1 1 0 0	0	1
1 1 0 1	1	0
1 1 1 0	1	0
1 1 1 1	0	1

奇偶校验器一般由异或门构成

异或门真值表

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

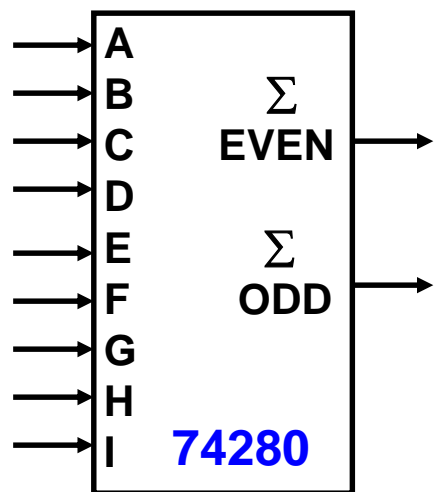
异或门特性

- 两个输入中有奇数个“1”，输出为1；有偶数个“1”，输出为0。
- 扩展：n个1位二进制数中有奇数个“1”，输出为1；有偶数个“1”，输出为0。

奇偶校验器

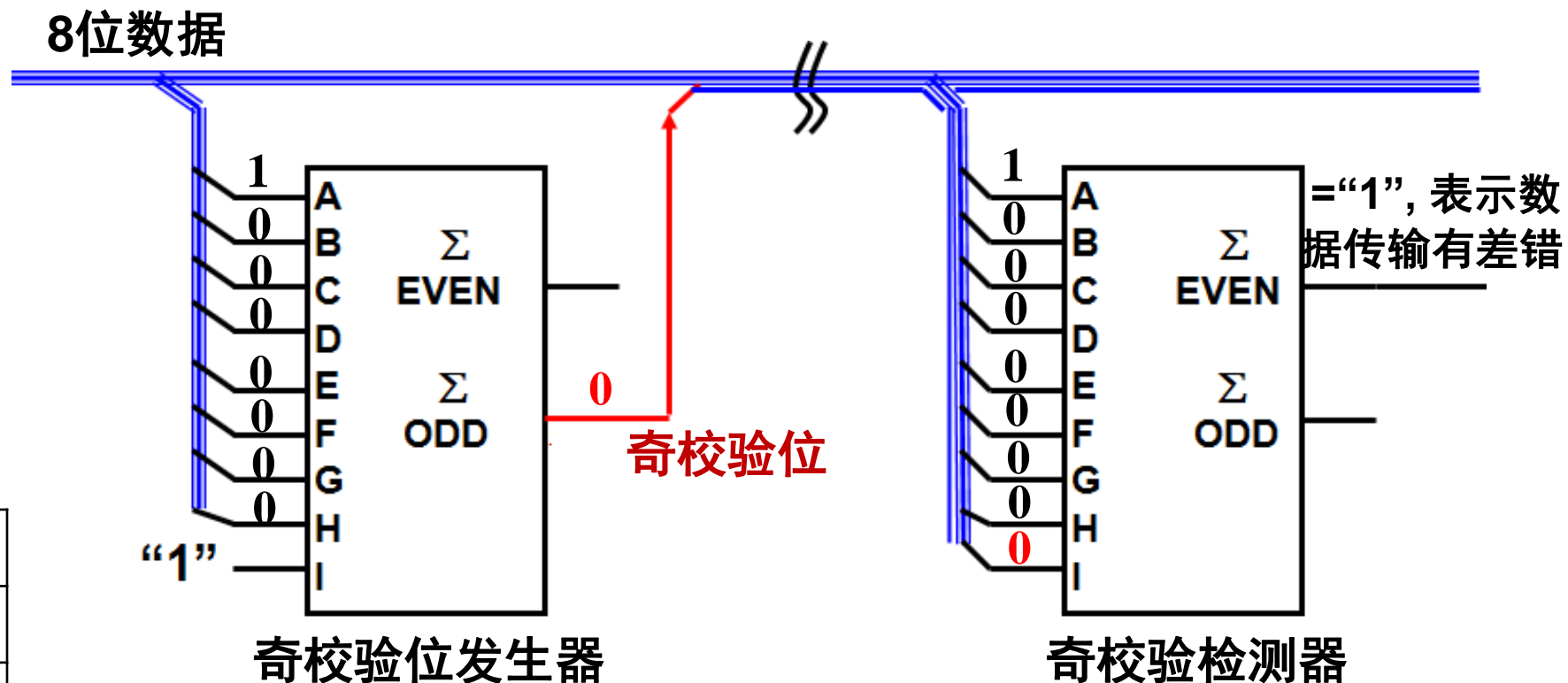
奇偶校验器/产生器：
74xx180、74xx280

例) 用9位奇偶校验器74LS280设计一个8位二
进制码的**奇校验位**发生器和检测器。



74XX280功能表

A~I	EVEN	ODD
偶数个“1”	1	0
奇数个“1”	0	1



奇偶校验器

奇偶校验实际应用意义

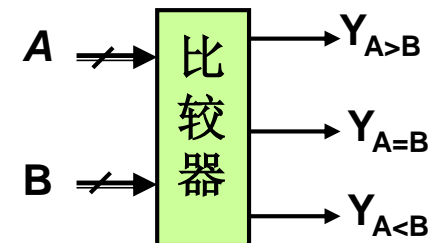
- ① 能够检测传送出错，但不能确定错误位置，不能纠错；
- ② 数据在存储或传送过程中，发生一位错误的可能性占96%以上；
- ③ 电路简单，容易实现，且有实际应用意义。

Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

数值比较器

- 计算机中对数据的基本处理方法
 - 加、减、乘、除
 - 比较运算
- 数值比较器：一种关系运算电路
 - 能对2个 n 位二进制数 A 和 B 进行比较的多输入、多输出的组合逻辑电路
 - 比较结果： $Y_{A>B}$ 、 $Y_{A<B}$ 、 $Y_{A=B}$



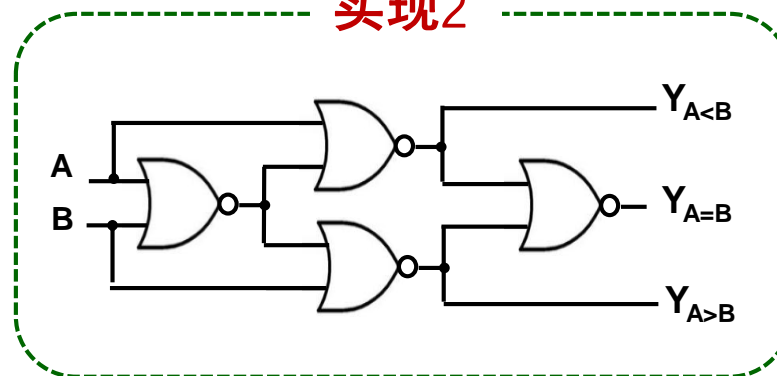
一位数值比较器

真值表

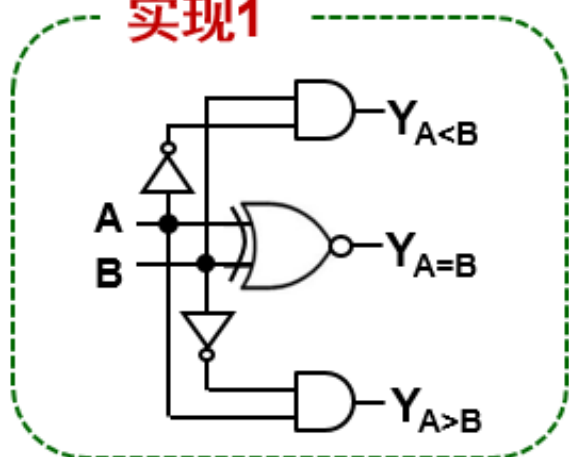
A	B	$Y_{A=B}$	$Y_{A>B}$	$Y_{A<B}$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

$$\begin{cases} Y_{A=B} = A \odot B \\ Y_{A>B} = A\bar{B} \\ Y_{A<B} = \bar{A}B \end{cases}$$

实现2



实现1



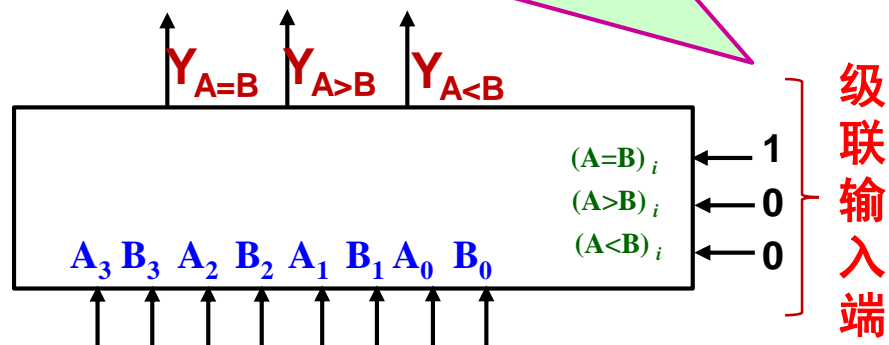
$$\begin{cases} Y_{A=B} = \bar{A}\bar{B} + AB = (A + \bar{B})(\bar{A} + B) = \overline{(A + \bar{A} + \bar{B})(B + \bar{B} + A)} \\ Y_{A>B} = A\bar{B} = \overline{\overline{A\bar{B}}} = \overline{\overline{A} + B} = A + \overline{(\bar{A} + B)} \\ Y_{A<B} = \bar{A}B = \overline{\overline{\bar{A}B}} = \overline{A + \bar{B}} = \bar{A} + \overline{(\bar{A} + B)} \end{cases}$$

多位数值比较器

- 自高而低逐位比较，只有在高位相等时，才需要比较低位。

接低位芯片的比较结果，用于芯片扩展。

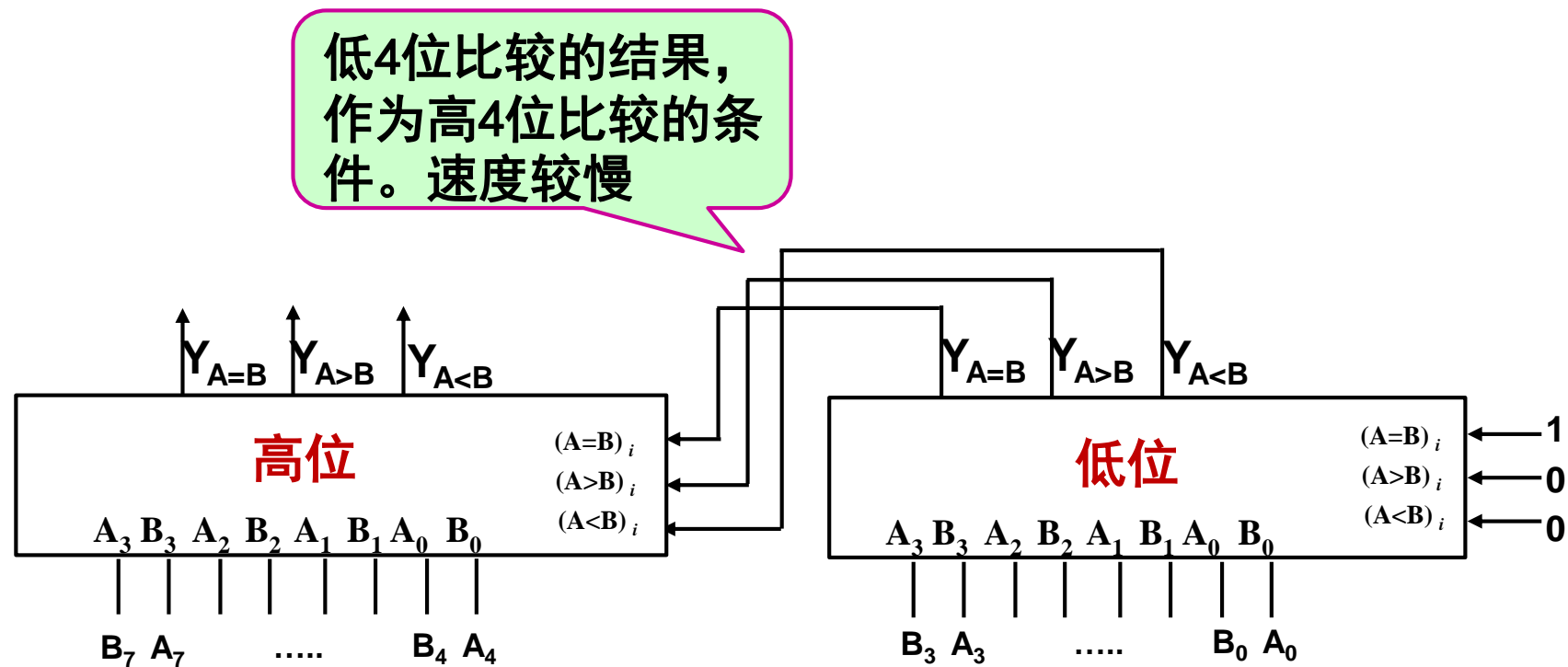
比较2个4位二进制数的大小时，3个输入端 $(A=B)_i$ 、 $(A>B)_i$ 、 $(A<B)_i$ 应接100，当 $A_3A_2A_1A_0 = B_3B_2B_1B_0$ 比较器的输出 $Y_{A=B}Y_{A>B}Y_{A<B} = 100$



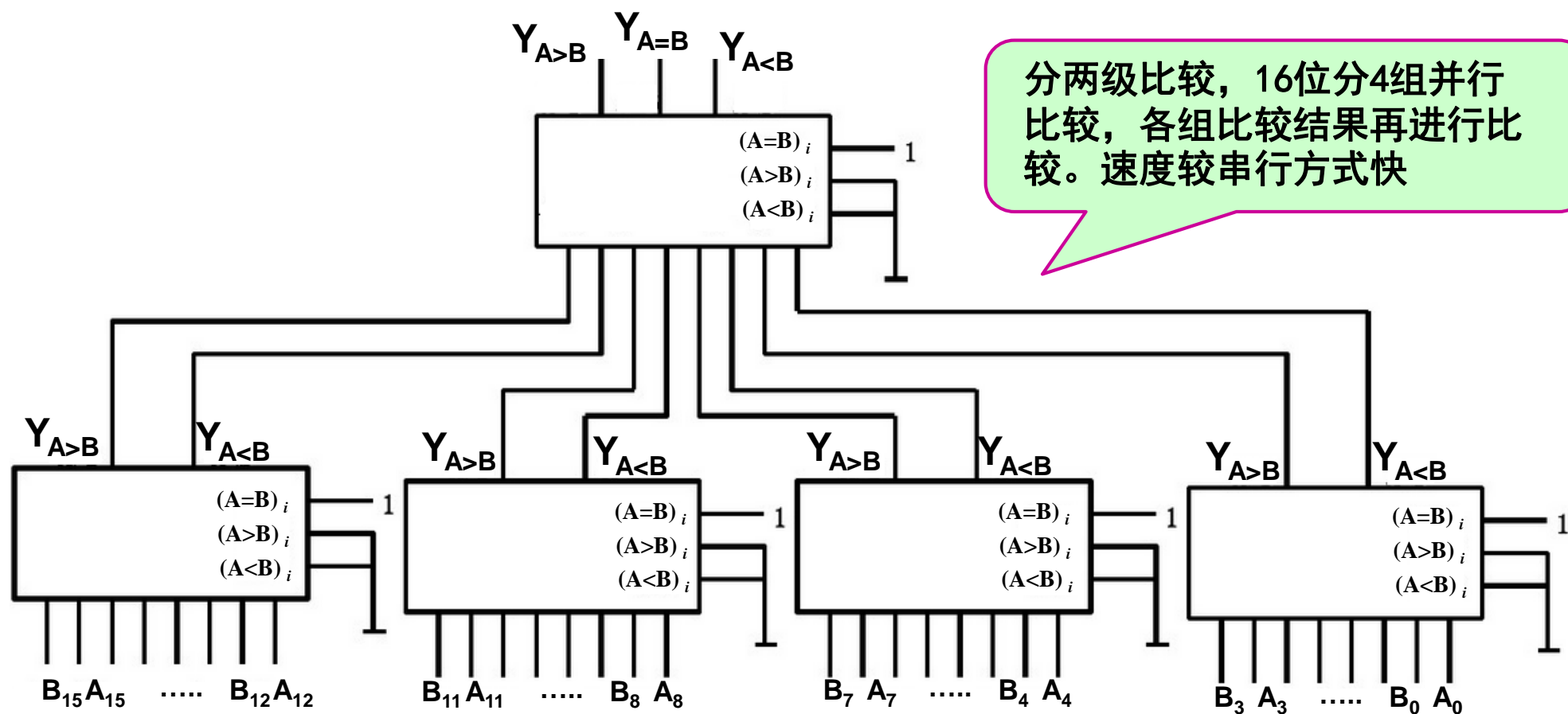
当 $A_3A_2A_1A_0 = B_3B_2B_1B_0$ ，比较器的输出复现3个输入端 $(A=B)_i$ 、 $(A>B)_i$ 、 $(A<B)_i$ 的状态。

比较输入				级联输入			输出		
A_3 B_3	A_2 B_2	A_1 B_1	B_0 A_0	$(A>B)_i$	$(A<B)_i$	$(A=B)_i$	$Y_{A>B}$	$Y_{A<B}$	$Y_{A=B}$
$A_3 > B_3$	X	X	X	X	X	X	1	0	0
$A_3 < B_3$	X	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	1	0	0
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	0	0	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	1	0	1	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	1	1	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	0	1	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	1	1	1	1

数值比较器的级联——①串行方式



数值比较器的级联——②并行方式



Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路