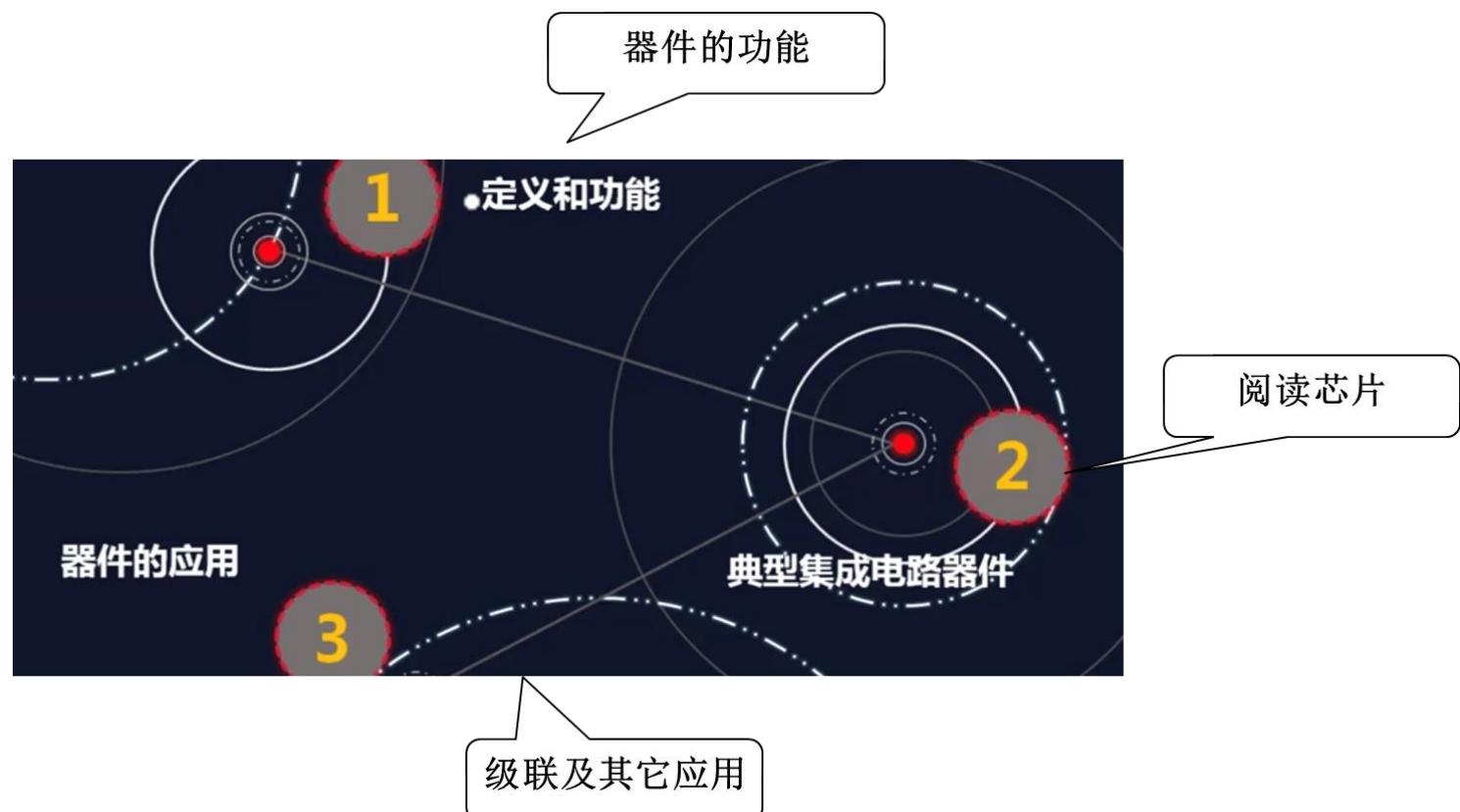


4.4 若干典型的组合逻辑集成电路

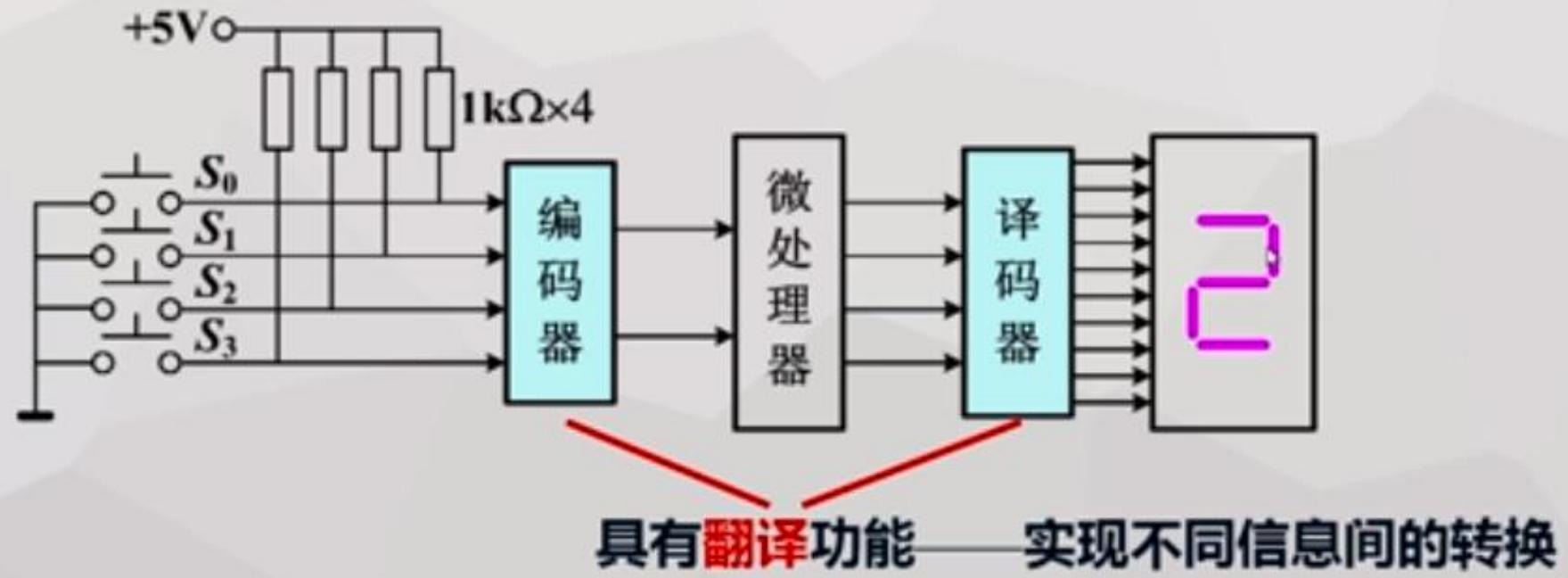


4.4.1 编码器

代码转换电路



- 在用文字和十进制数表示的信息与二进制信息之间，完成不同信息间相互转换的电路。
- 按照信息转换方式的不同，分为编码器 (encoder) 和译码器 (decoder)。



编码：用二进制代码表示具有某种特定含义的信号

译码：把一组代码所表示的特定含义翻译出来。

1、编码器 (Encoder)的定义

编码：赋予二进制代码特定含义的过程称为编码。

如：8421BCD码中，用1000表示数字8

如：ASCII码中，用1000001表示字母A等

编码器：具有编码功能的逻辑电路。

如：BCD编码器，将10个编码输入信号分别编成10个4位码输出。

如：8线-3线编码器，将8个输入的信号分别编成 8个3位二进制数码输出。

2、编码器 (Encoder)的分类

普通编码器：任何时候只允许输入一个有效编码信号，否则输出就会发生混乱。

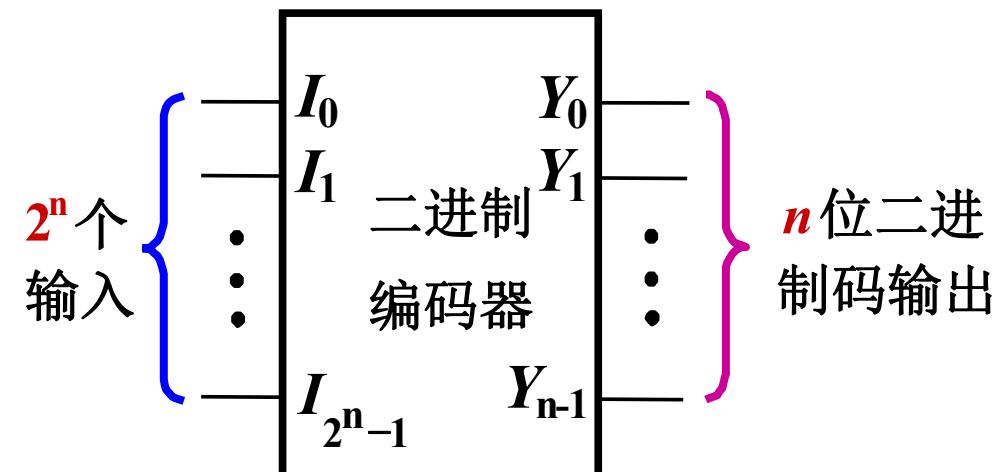
优先编码器：允许同时输入两个以上的有效编码信号。当同时输入几个有效编码信号时，优先编码器能按预先设定的优先级别，只对其中优先权最高的一个进行编码。

3、编码器的工作原理

普通二进制编码器：

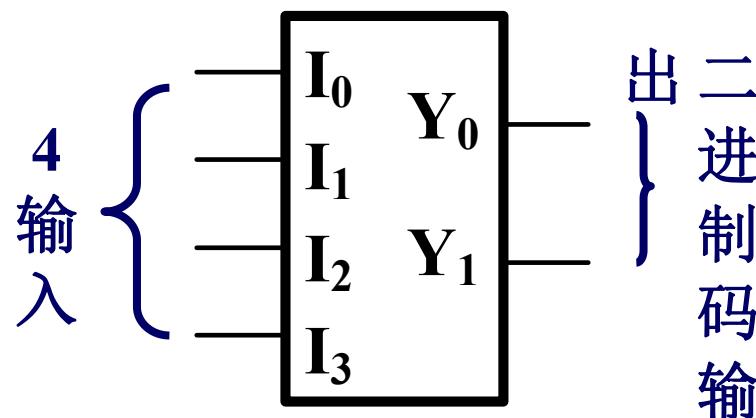
将输入信号表达为输出的二进制代码。

二进制编码器的结构框图



4线-2线普通二进制编码器 (设计)

(a) 逻辑框图



$$Y_1 = \bar{I}_0 \bar{I}_1 I_2 \bar{I}_3 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3$$

$$Y_0 = \bar{I}_0 I_1 \bar{I}_2 \bar{I}_3 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3$$

(b) 逻辑功能表

I_0	I_1	I_2	I_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

编码器的输入为高电平有效。

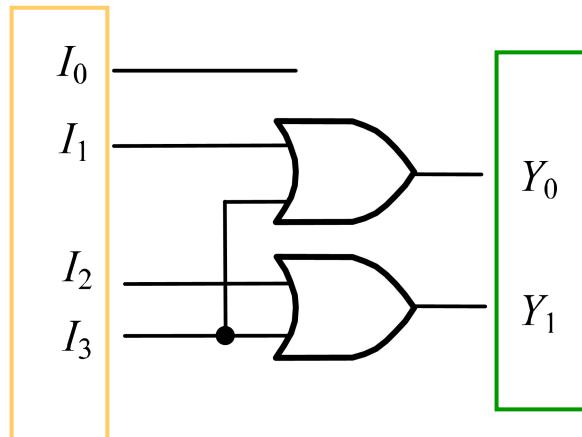
除表中列出4个输入组合外，将输入的其它12种组合对应的输出？

该表达式是否可以再简化？

除表中列出4个输入组合外，将输入的其它12种组合对应的输出？

如果看做无关项，则表达式为：
$$Y_1 = I_2 + I_3$$

$$Y_0 = I_1 + I_3$$



若有2个以上的输入为有效信号？

当 $I_1 = I_2 = 1$ 时， $Y_1 Y_0 = 11$

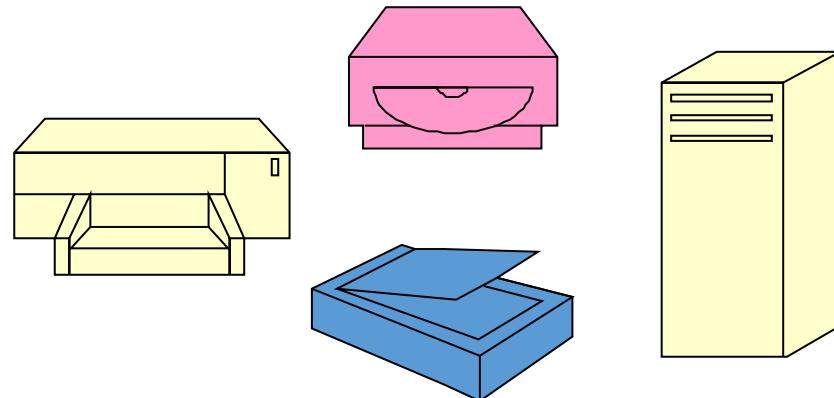
当只有 I_3 为1时， $Y_1 Y_0 = 11$

此时，无法输出有效编码。

结论：普通编码器不能同时输入两个以上的有效编码信号。

4、优先编码器

优先编码器的提出：



实际应用中，经常有两个或更多输入编码信号同时有效。

必须根据轻重缓急，规定好这些外设允许操作的先后次序，即优先级别。

优先编码器：

识别多个编码请求信号的优先级别，并进行相应编码的逻辑部件。

优先编码器 (4—2 线优先编码器) (设计)

输入编码信号高电平有效，输出为二进制代码

输入编码信号优先级从高到低为 $I_3 \sim I_0$

输入为编码信号 $I_3 \sim I_0$ 输出为 $Y_1 Y_0$

(1) 列出功能表

输入				输出	
I_0	I_1	I_2	I_3	Y_1	Y_0
1	0	0	0	0	0
X	1	0	0	0	1
X	X	1	0	1	0
X	X	X	1	1	1

低 高

(2) 写出逻辑表达式

$$Y_1 = \overline{I_2} \overline{I_3} + I_3$$
$$Y_0 = \overline{I_1} \overline{I_2} \overline{I_3} + I_3$$

(3) 画出逻辑电路 (略)

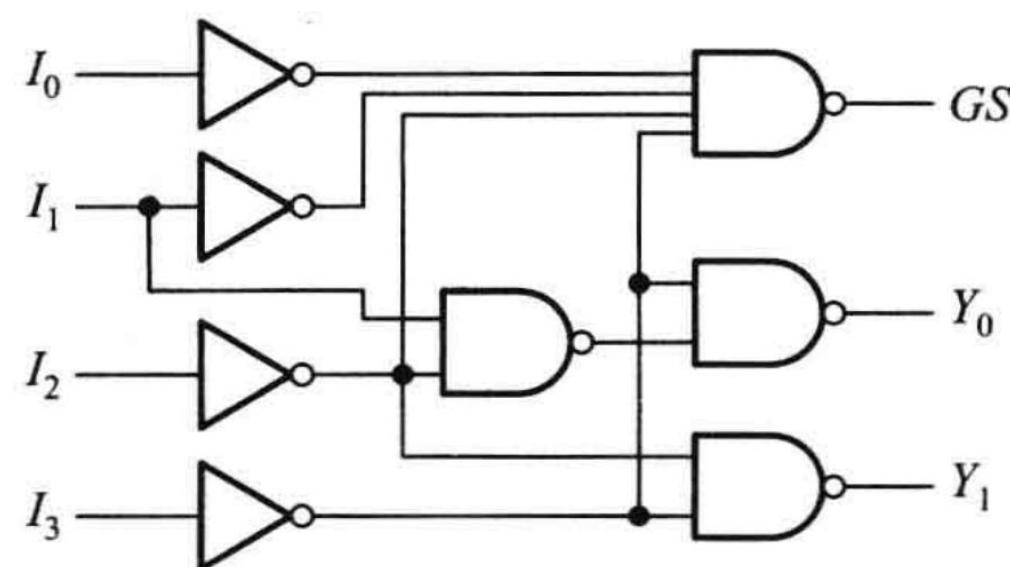
4.4.2 试用与非门设计一4输入的优先编码器,要求输入、输出及工作状态标志均为高电平有效。列出真值表,画出逻辑图。

输入				输出		
I_0	I_1	I_2	I_3	Y_1	Y_0	GS
0	0	0	0	0	0	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1
x	x	x	1	1	1	1

$$Y_1 = I_2 \bar{I}_3 + I_3 = I_2 + I_3 = \overline{\overline{I}_2 \overline{I}_3}$$

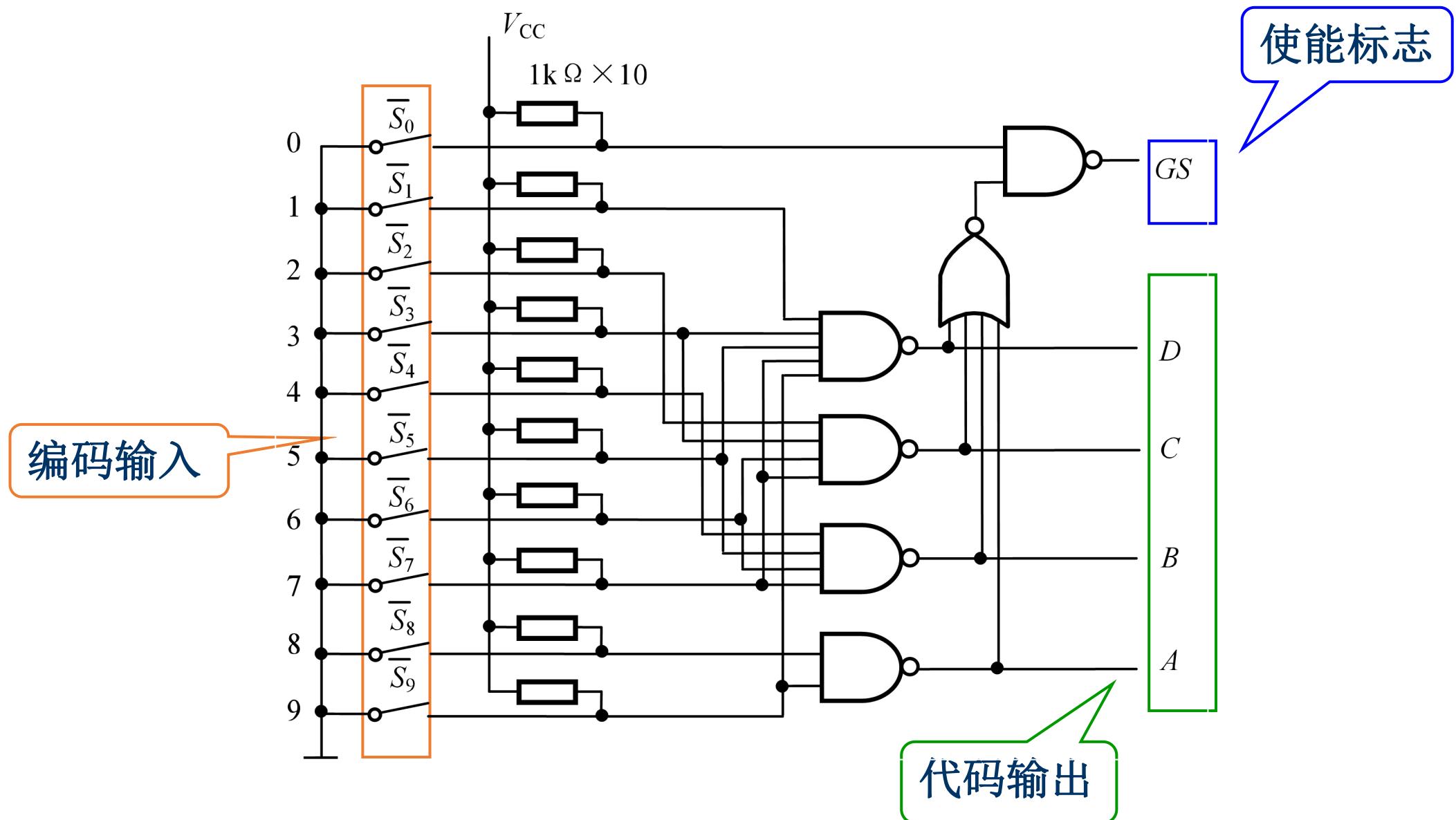
$$Y_0 = I_1 \bar{I}_2 \bar{I}_3 + I_3 = I_1 \bar{I}_2 + I_3 = \overline{\overline{I}_1 \overline{I}_2 \overline{I}_3}$$

$$GS = I_0 + I_1 + I_2 + I_3 = \overline{\overline{I}_0 \overline{I}_1 \overline{I}_2 \overline{I}_3}$$



设计8421BCD编码器

键盘输入逻辑电路由编码器组成，
即用十个按键和门电路组成的8421BCD编码器。



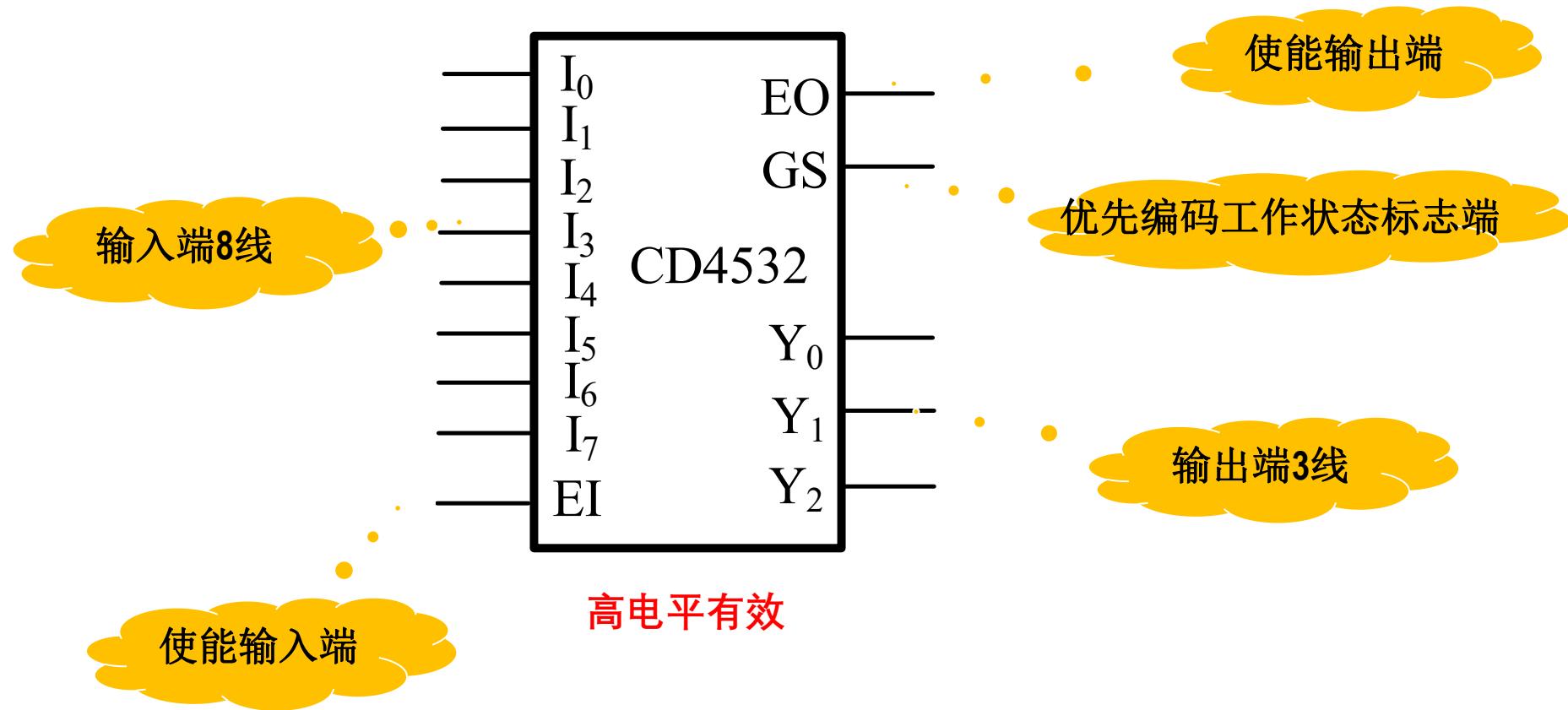
键盘输入8421BCD码编码器功能表

	输入										输出				
\bar{S}_0	\bar{S}_1	\bar{S}_2	\bar{S}_3	\bar{S}_4	\bar{S}_5	\bar{S}_6	\bar{S}_7	\bar{S}_8	\bar{S}_9	A	B	C	D	GS	
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	
1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	
1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	
1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1
1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	
1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	
1	1	1	1	0	1	1	1	1	1	0	0	1	1	1	
1	1	0	1	1	1	1	1	1	1	0	0	1	0	1	
0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1

该编码器为输入低电平有效，输出高电平有效，GS为标志位。

5、典型编码器电路

优先编码器CD4532的示意框图(中规模集成芯片)（16管脚）



常用集成电路优先编码器种类

74x147:

10线-4线优先编码器，输入信号**低电平有效**，输出为BCD码。

74x148:

8线-3线优先编码器，输入信号**低电平有效**，输出为二进制代码。

CD4532:

8线-3线优先编码器，输入信号**高电平有效**，输出为二进制代码。

6、应用---优先编码器CD4532

EI=1时，编码器工作，EI=0时，禁止编码器工作。

思路：先控制端，再输入到输出端

EI=1时，且所有输入端=0时，GS=0，EO=1。

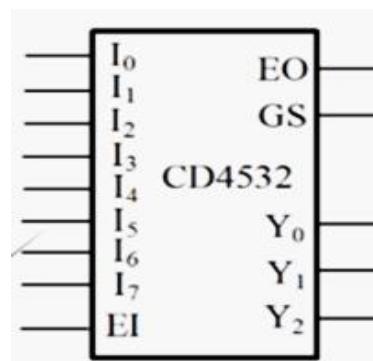
CD4532功能表

EI=1时，且至少有一个输入端=1时，GS=1，EO=0。

输入										输出			
EI	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0	GS	EO
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	0	1	0
1	0	0	1	×	×	×	×	×	1	0	1	1	0
1	0	0	0	1	×	×	×	×	1	0	0	1	0
1	0	0	0	0	1	×	×	×	0	1	1	1	0
1	0	0	0	0	0	1	×	×	0	1	0	1	0
1	0	0	0	0	0	0	1	×	0	0	1	1	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0

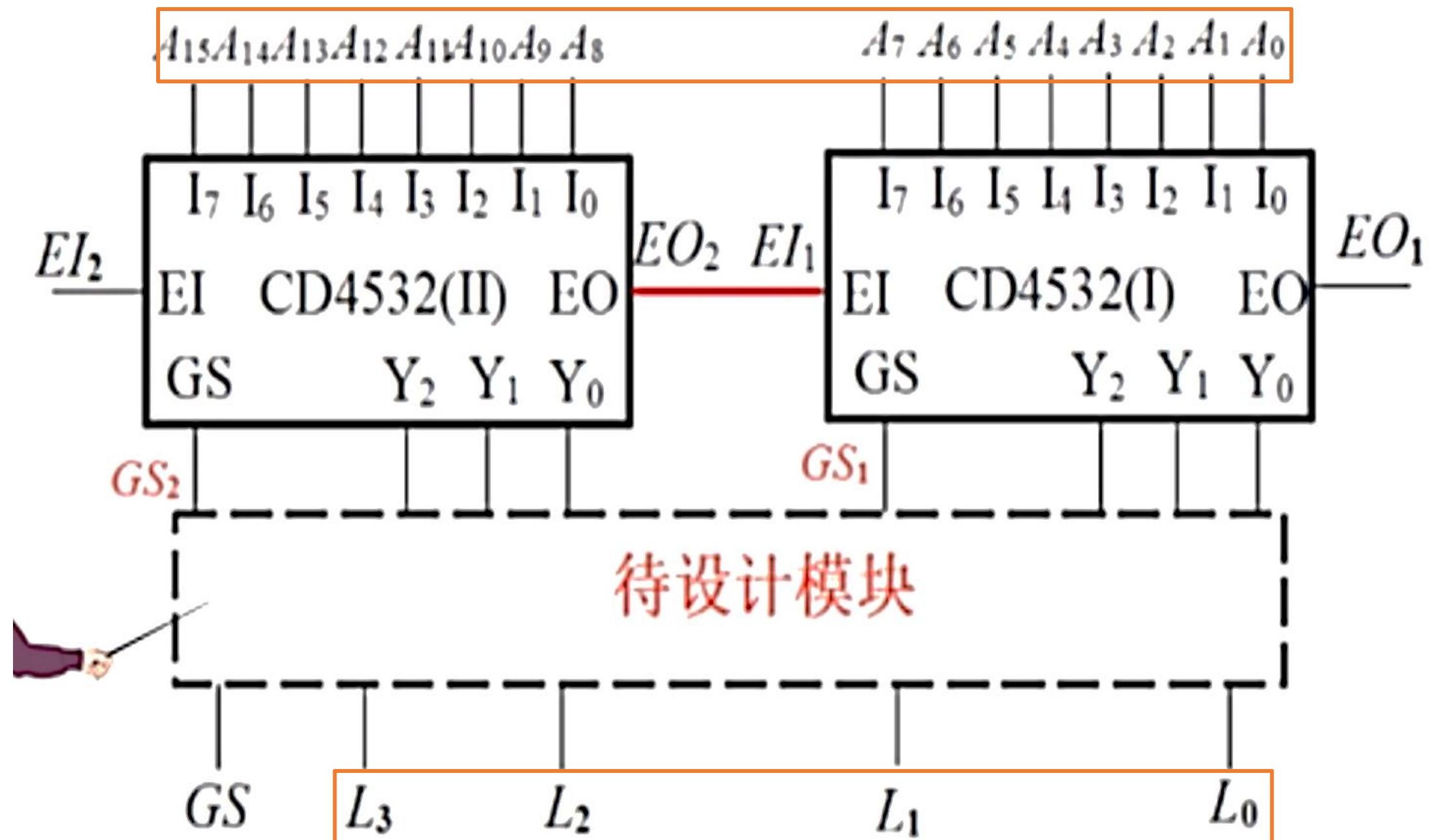
思考：为什么要设计GS、EO输出信号？

在优先编码器芯片中，设置使能端EI、EO的目的，就是为了能够将多个芯片级联起来，构成规模更大的编码器。

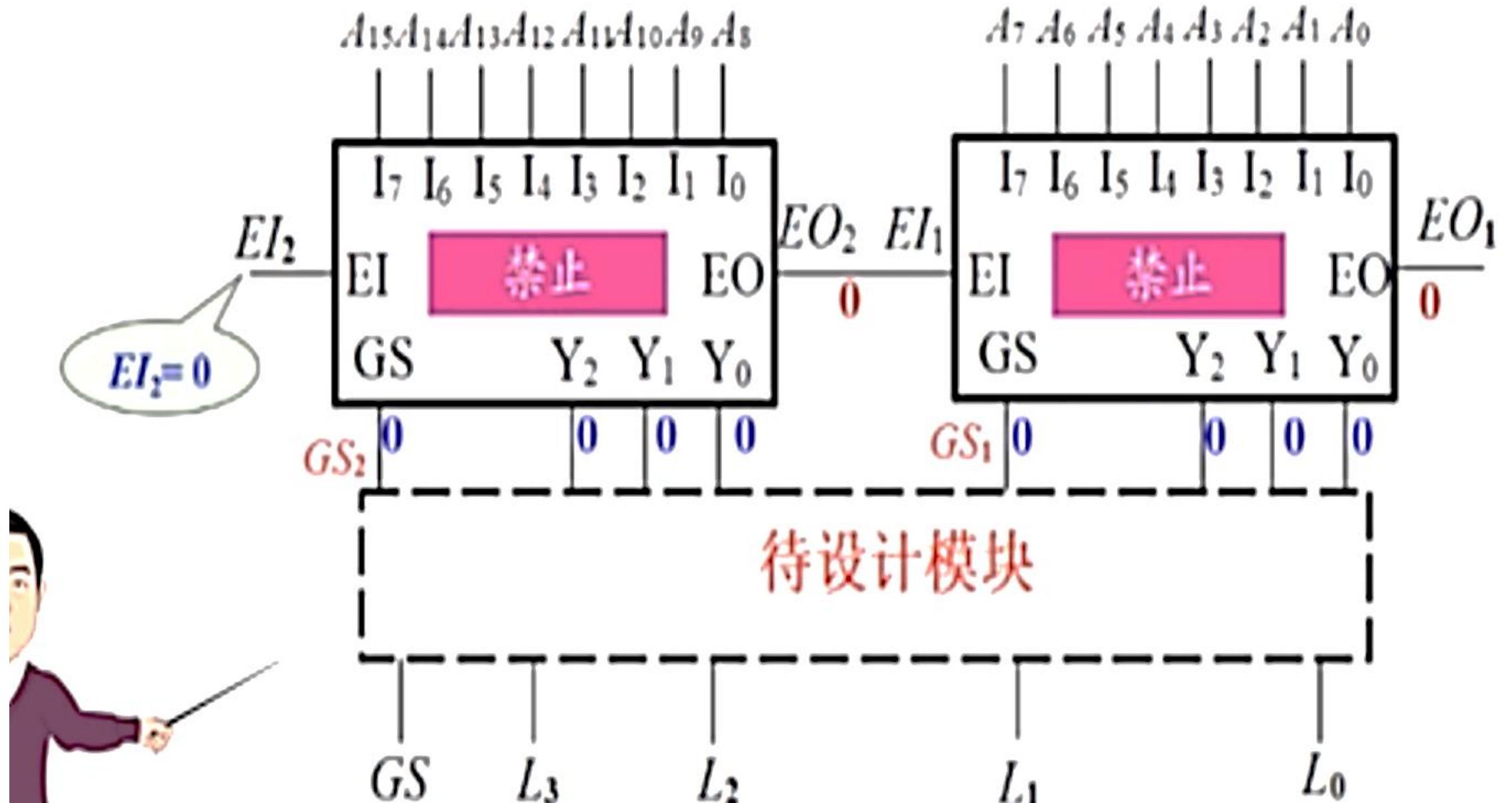


EI	GS	EO	状态
0	0	0	不工作
1	0	1	工作，但无输入
1	1	0	工作，且有输入
1	1	1	不可能出现

例题：用两片8线-3线编码器CD4532组成16线-4线优先编码器。

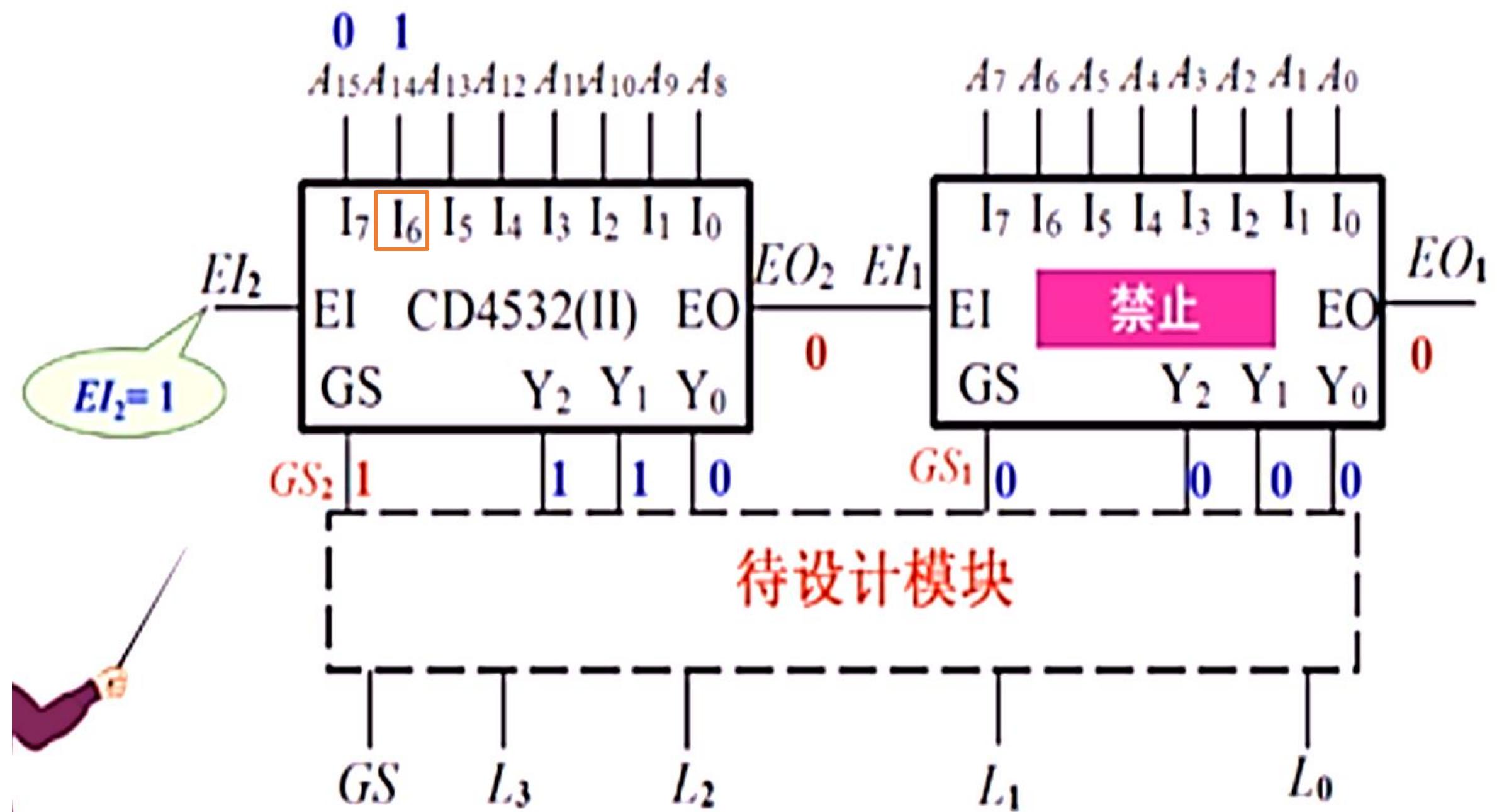


两片8线-3线的CD4532 → 16线-4线优先编码器



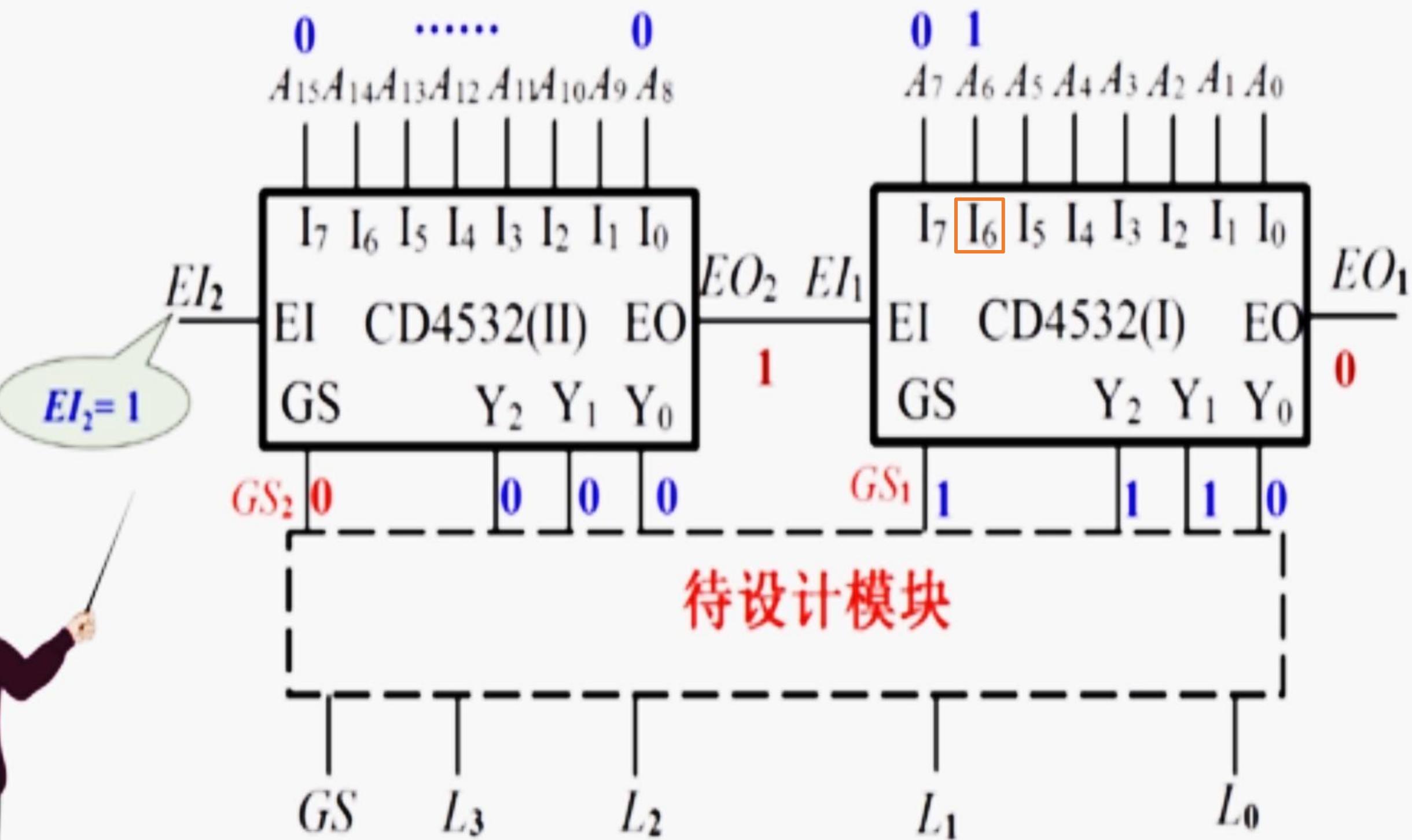
思考：两个芯片的优先级哪个高一些？ $A_0 \sim A_{15}$ 的优先级如何

两片8线-3线的CD4532 → 16线-4线优先编码器

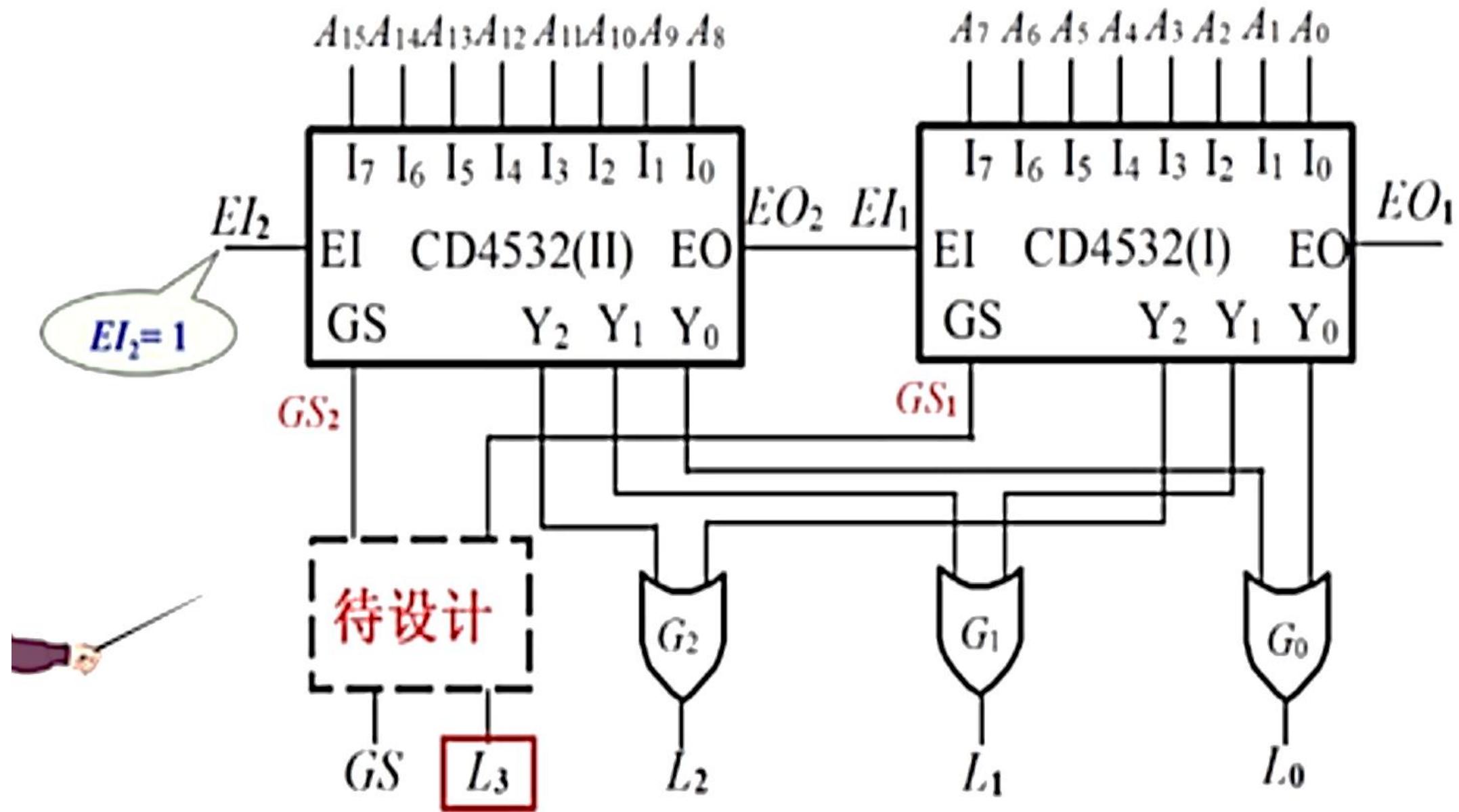


结论：2号芯片的优先级比1号芯片优先级高。

两片8线-3线的CD4532 → 16线-4线优先编码器

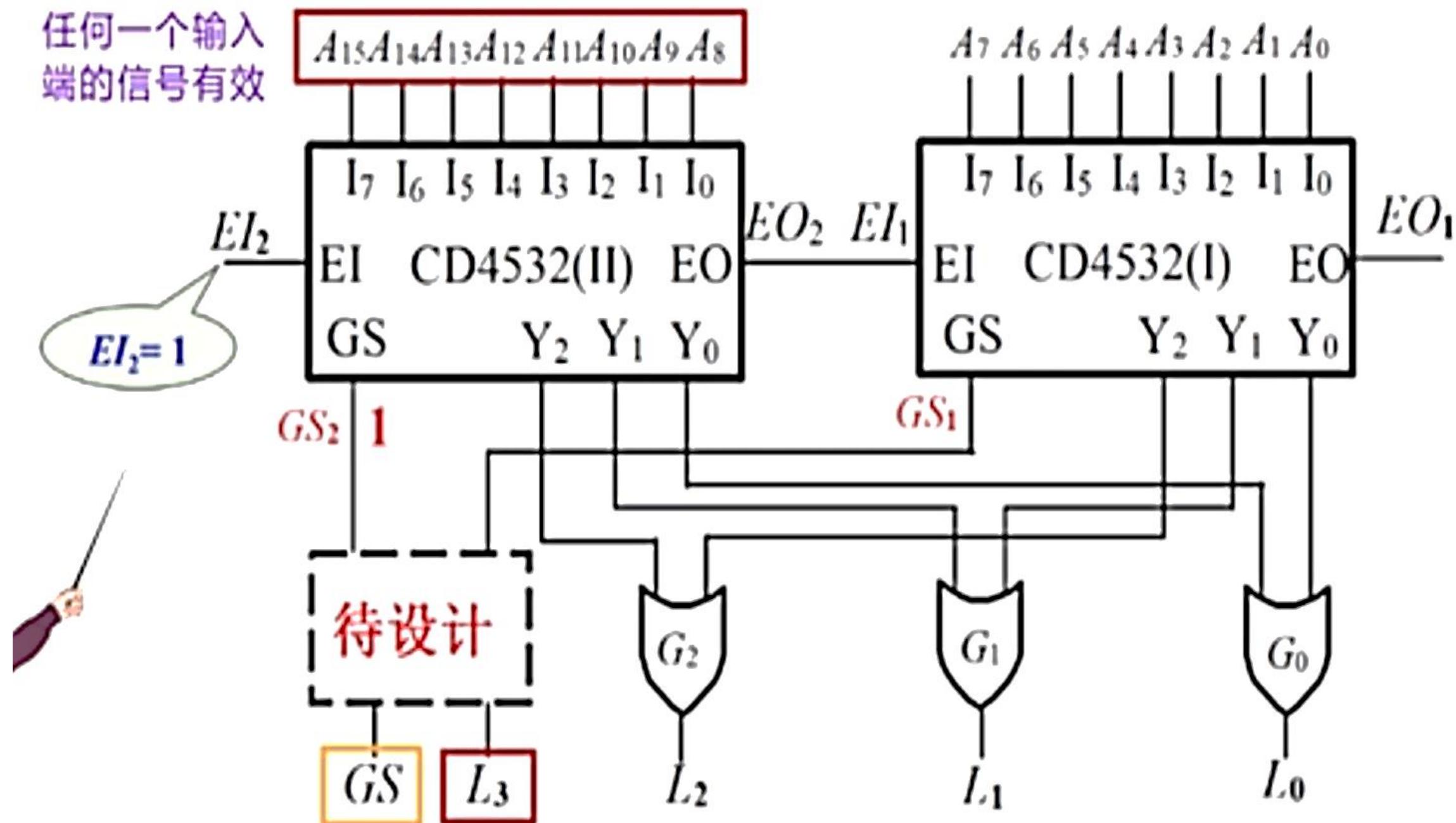


两片8线-3线的CD4532 → 16线-4线优先编码器



最高位：控制端构成
低三位：片输出共用

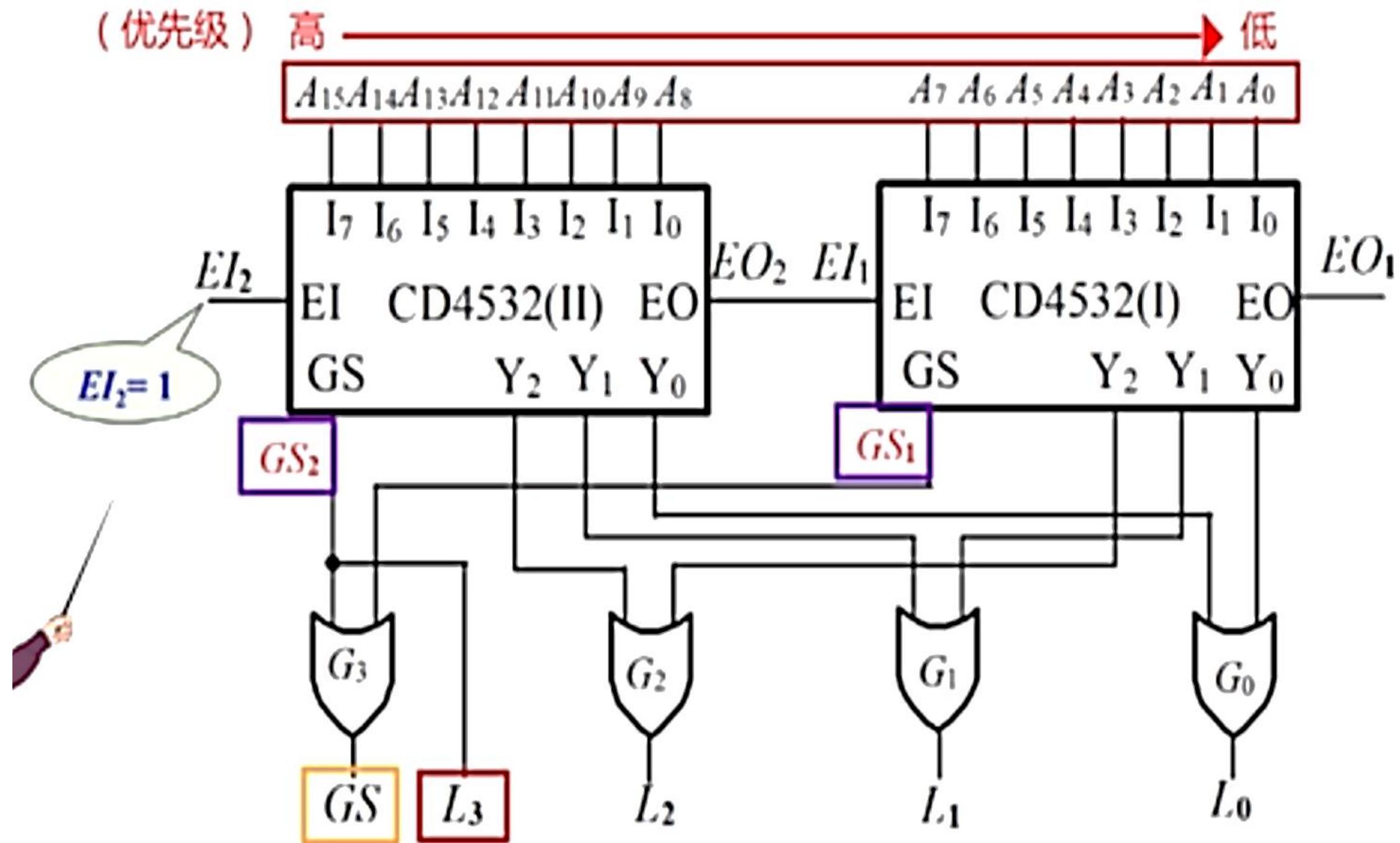
两片8线-3线的CD4532 → 16线-4线优先编码器



最高位：控制端构成
低三位：片输出共用

两片8线-3线的CD4532 → 16线-4线优先编码器

(优先级) 高 → 低

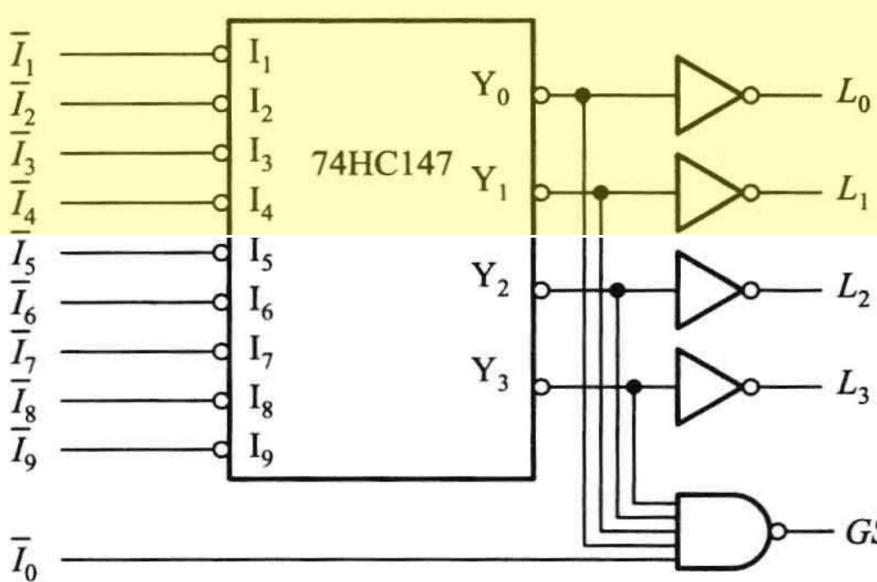


最高位：控制端构成
低三位：片输出共用

4.4.3 优先编码器 74HC147 的功能表如表题 4.4.3 所示, 试用 74HC147 和适当的门构成输入为低有效的 $\bar{I}_0 \sim \bar{I}_9$, 输出为低电平有效的 8421BCD 码, 并具有编码输出标志的编码器。

表题 4.4.3 优先编码器 74HC147 功能表

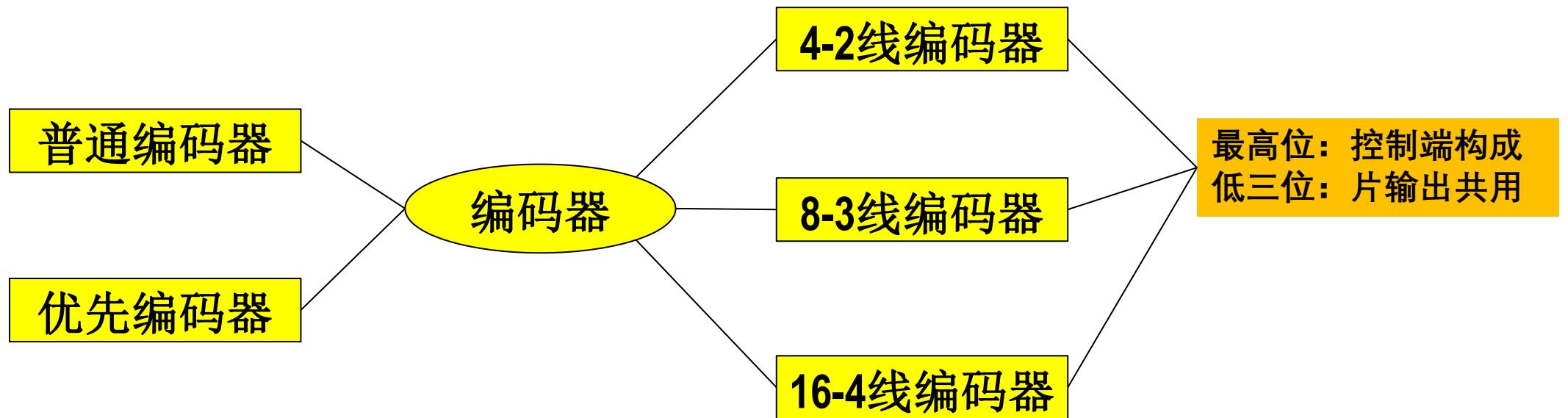
输入										输出			
\bar{I}_1	\bar{I}_2	\bar{I}_3	\bar{I}_4	\bar{I}_5	\bar{I}_6	\bar{I}_7	\bar{I}_8	\bar{I}_9		\bar{Y}_3	\bar{Y}_2	\bar{Y}_1	\bar{Y}_0
1	1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x	x	0	0 1	1 0	1 0	0 1
x	x	x	x	x	x	x	x	0	1	0	1	1	1
x	x	x	x	x	x	x	0	1	1	1	0	0	0
x	x	x	x	x	0	1	1	1	1	1	0	0	1
x	x	x	x	0	1	1	1	1	1	1	0	1	0
x	x	x	0	1	1	1	1	1	1	1	0	1	1
x	x	0	1							1	1	0	0
x	0	1	1							1	1	0	1
0	1	1	1							1	1	1	0



4.4.4 试用74HC147设计键盘编码电路,十个按键分别对应十进制数0~9,编码器的输出为8421BCD码。要求按键9的优先级别最高,并且有工作状态标志,以说明没有按键按下和按键0按下两种情况。

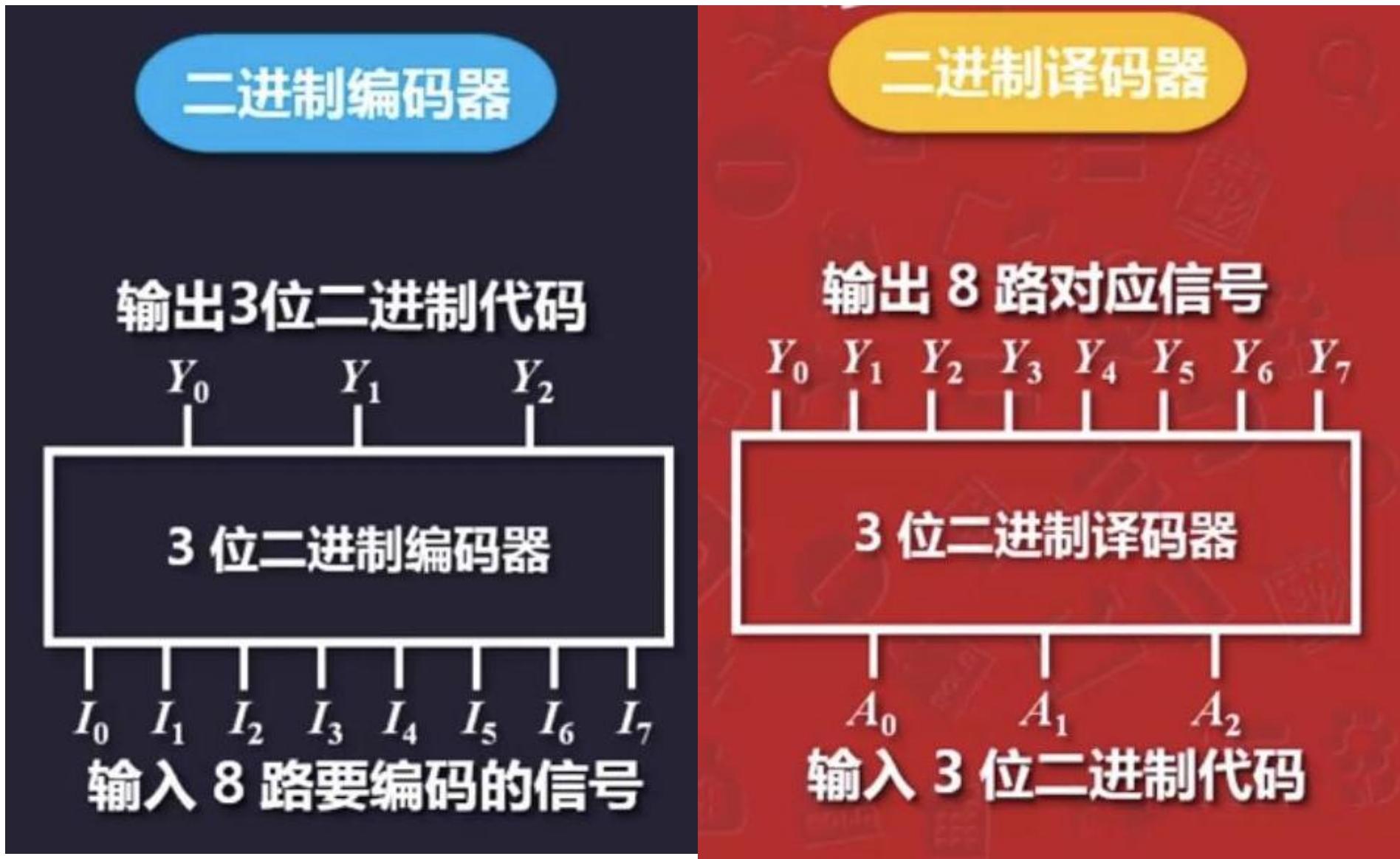
输入										输出				
\bar{S}_0	\bar{S}_1	\bar{S}_2	\bar{S}_3	\bar{S}_4	\bar{S}_5	\bar{S}_6	\bar{S}_7	\bar{S}_8	\bar{S}_9	A	B	C	D	GS
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
x	x	x	x	x	x	x	x	x	0	1	0	0	1	1
x	x	x	x	x	x	x	x	0	1	1	0	0	0	1
x	x	x	x	x	x	x	0	1	1	0	1	1	1	1
x	x	x	x	x	x	0	1	1	1	0	1	1	0	1
x	x	x	x	x	0	1	1	1	1	0	1	0	1	1
x	x	x	x	0	1	1	1	1	1	0	1	0	0	1
x	x	x	0	1	1	1	1	1	1	0	0	1	1	1
x	x	0	1	1	1	1	1	1	1	0	0	1	0	1
x	0	1	1	1	1	1	1	1	1	0	0	0	1	1
0	1	1	1	1	1	1	1	1	1	0	0	0	0	1

小结：



掌握：
• CD4532 和74HC147芯片
• 功能表
• 有效电平

4.4.2 译码器/数据分配器



1. 译码器的定义与分类

译码：译码是编码的逆过程，它能将二进制码翻译成代表某一特定含义的信号。
(即电路的某种状态)

译码器：具有译码功能的逻辑电路称为译码器。

译码器的分类：

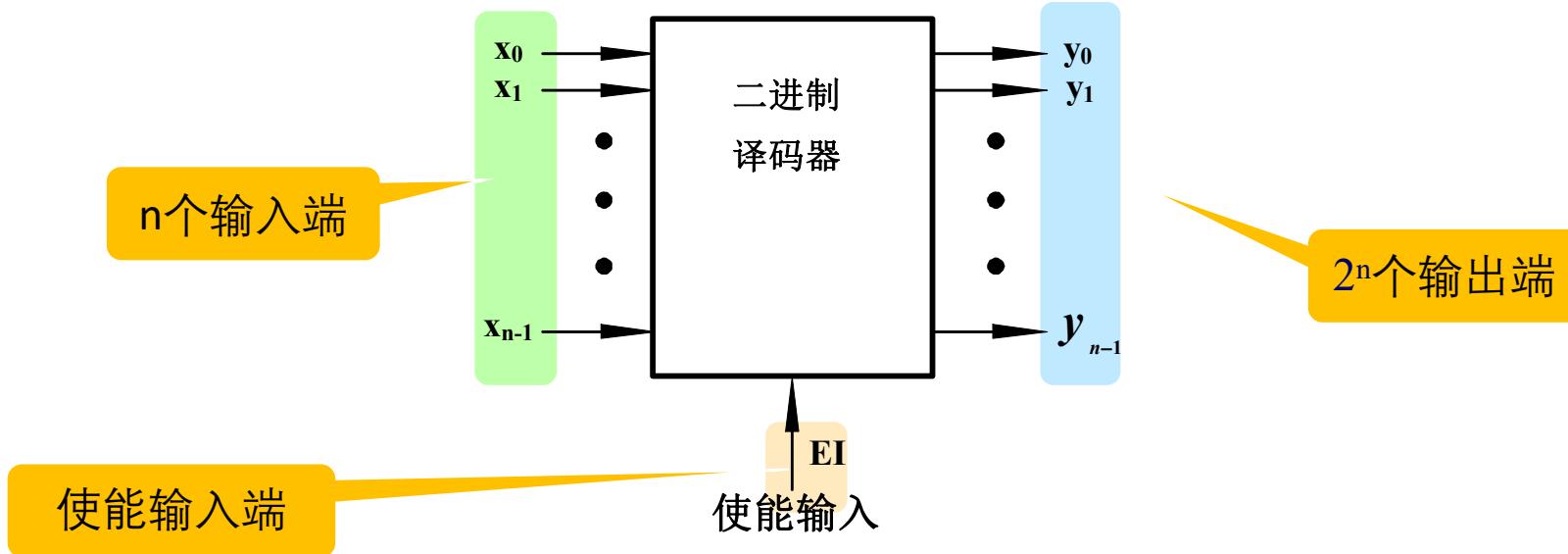
唯一地址译码器 将一系列代码转换成与之一一对应的有效信号。

常见的唯一地址译码器：
 {
 二进制译码器
 二…十进制译码器
 显示译码器

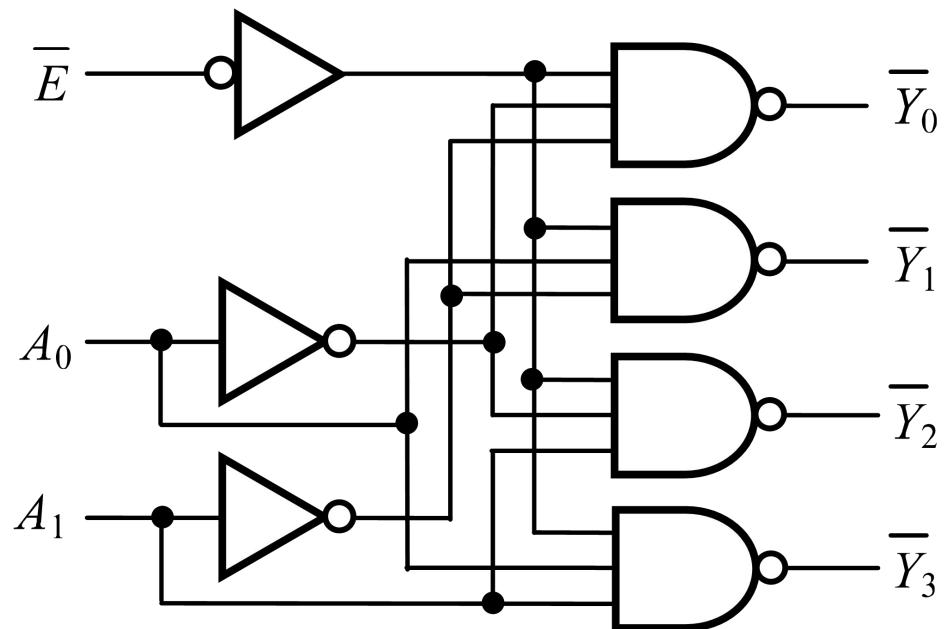
代码变换器 将一种代码转换成另一种代码。

2. 典型二进制译码器电路及应用

常用于计算机中对存储器单元地址的译码



2 线-4线译码器的逻辑电路（分析）



逻辑功能表

输入			输出			
\bar{E}	A_1	A_0	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

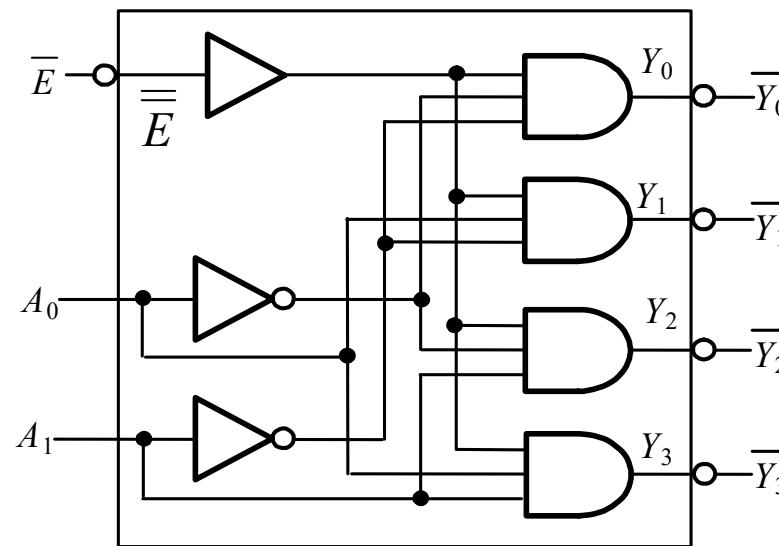
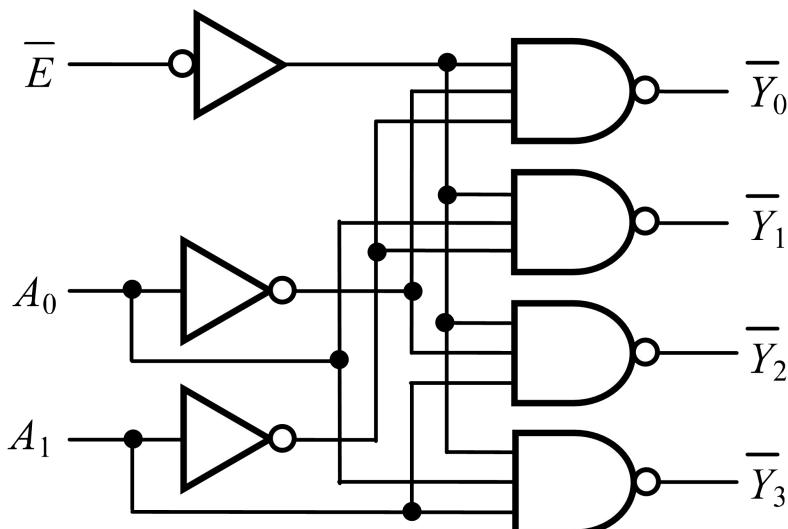
$$\bar{Y}_0 = \overline{\bar{E} A_1 A_0}$$

$$\bar{Y}_1 = \overline{\bar{E} \bar{A}_1 A_0}$$

$$\bar{Y}_2 = \overline{\bar{E} A_1 \bar{A}_0}$$

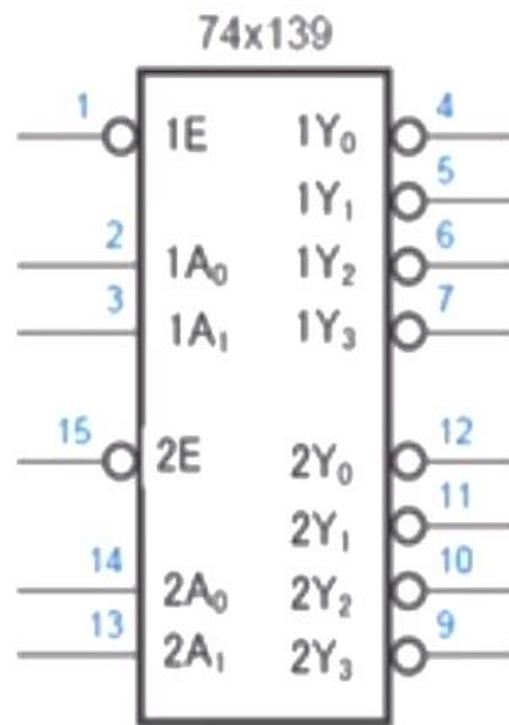
$$\bar{Y}_3 = \overline{\bar{E} \bar{A}_1 \bar{A}_0}$$

2线-4线译码器 (74HC139)



逻辑功能表

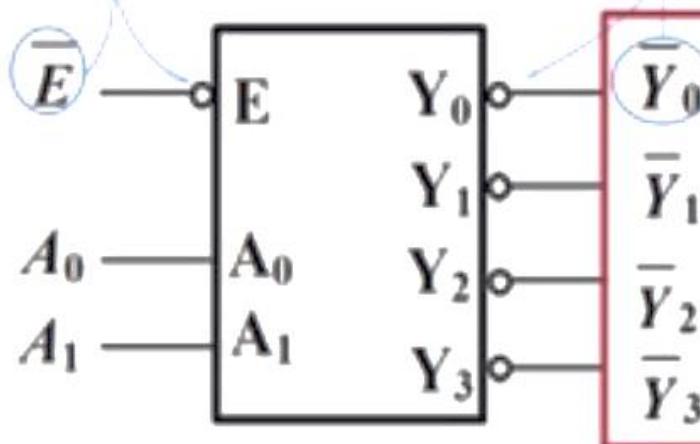
输入			输出			
\bar{E}	A_1	A_0	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



逻辑符号 (a)

低电平是有效状态

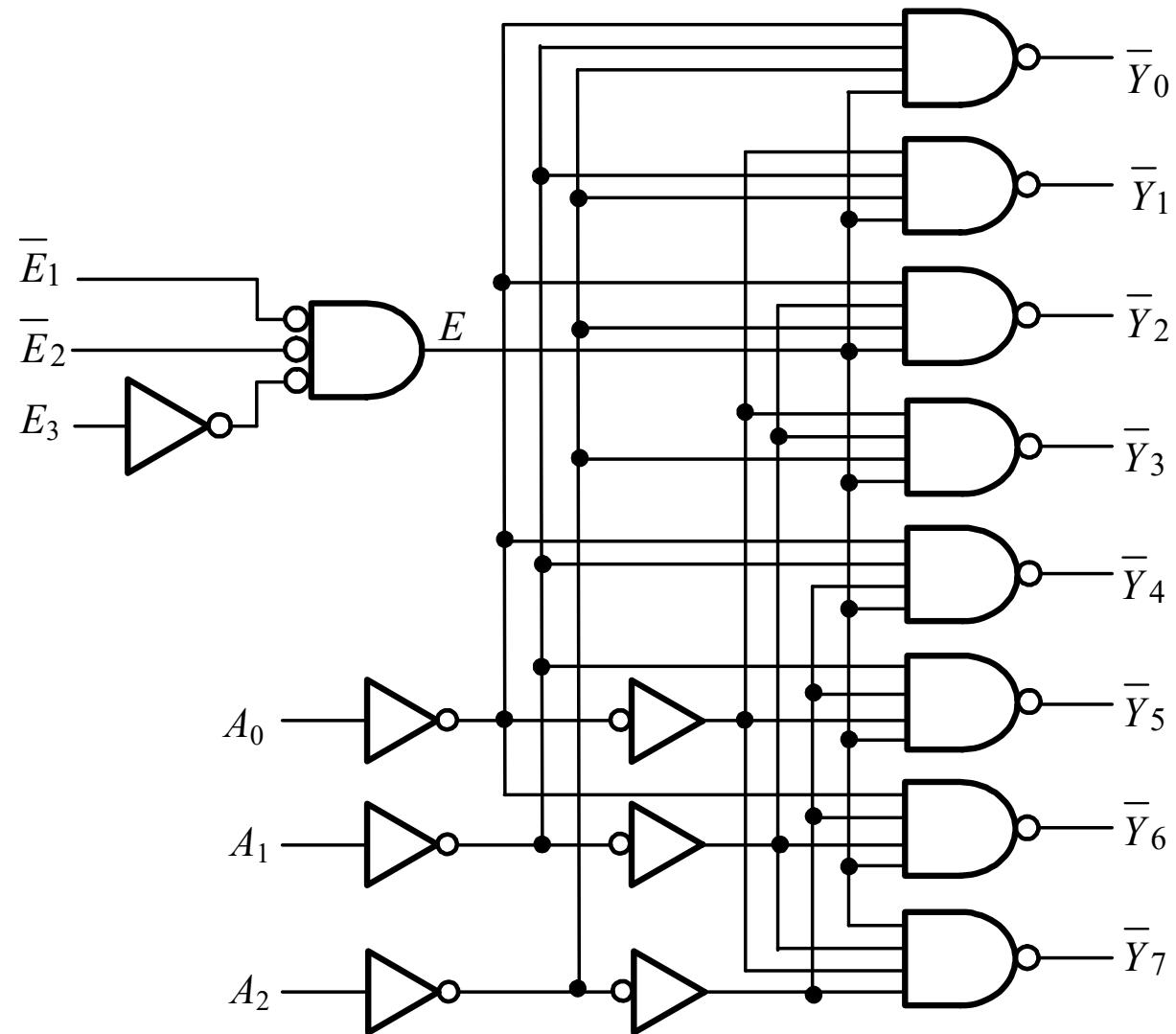
1/2 74x139



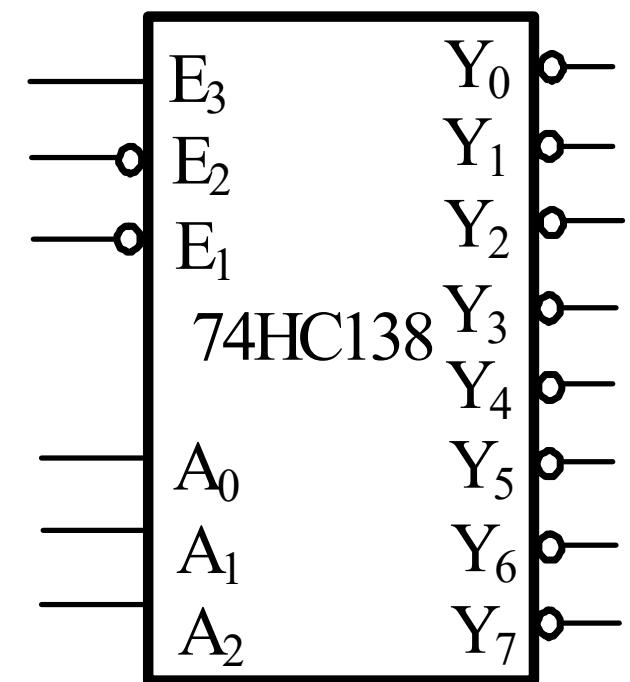
逻辑符号 (b)

- 逻辑符号字母上面的“—”号说明该输入或输出是低电平有效。
- 在推导表达式的过程中，如果低有效的输入或输出变量(如)上面的“—”号参与运算。

3线-8线译码器（74HC138）

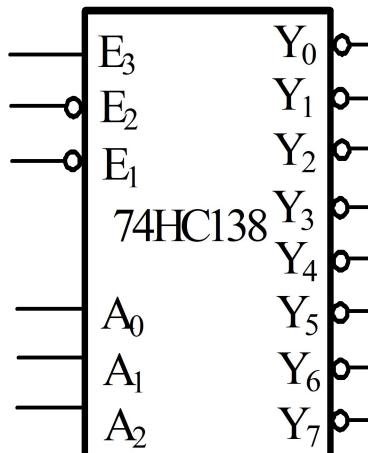


逻辑图



逻辑符号

3线—8线译码器（74HC138）功能表



输入						输出							
E_3	\bar{E}_2	\bar{E}_1	A_2	A_1	A_0	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7
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
0	X	X	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	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

$$\bar{Y}_0 = \overline{\bar{A}_2 \cdot \bar{A}_1 \cdot \bar{A}_0} \quad \bar{Y}_1 = \overline{\bar{A}_2 \cdot \bar{A}_1 \cdot A_0} \quad \bar{Y}_2 = \overline{\bar{A}_2 \cdot A_1 \cdot \bar{A}_0} \quad \bar{Y}_3 = \overline{\bar{A}_2 \cdot A_1 \cdot A_0}$$

$$\bar{Y}_4 = \overline{A_2 \cdot \bar{A}_1 \cdot \bar{A}_0} \quad \bar{Y}_5 = \overline{A_2 \cdot \bar{A}_1 \cdot A_0} \quad \bar{Y}_6 = \overline{A_2 \cdot A_1 \cdot \bar{A}_0} \quad \bar{Y}_7 = \overline{A_2 \cdot A_1 \cdot A_0} \quad \bar{Y}_i = \overline{m_i}$$

74x138芯片的使能信号有效时（当 $E_3=1$ ， $E_2=E_1=0$ 时）
它的输出包含了输入A2、A1、A0组成的所有最小项的非。

3. 用译码器实现逻辑函数

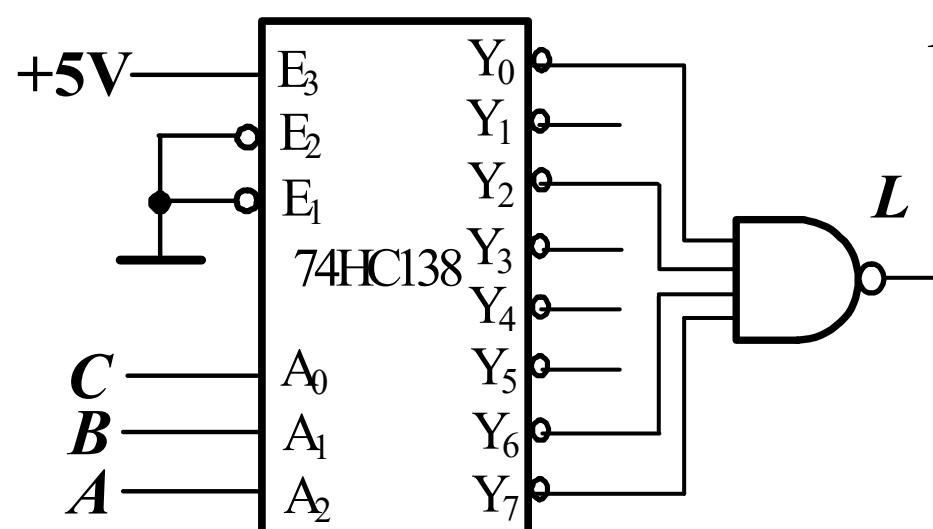
- 74x138芯片的使能信号有效时，它的输出包含了输入A2、A1、A0组成的所有最小项的非。
- 任何一个逻辑函数式都可以变换为最小项之和的形式。

基于这一点用该器件能够方便地实现三变量逻辑函数。

例题：用一片74HC138实现函数 $L = \overline{A}\overline{C} + AB$

首先将任何一个逻辑函数式都可以变换为最小项之和的形式。

其次，74x138芯片的使能信号有效时，它的输出包含了输入A2、A1、A0组成的所有最小项的非。



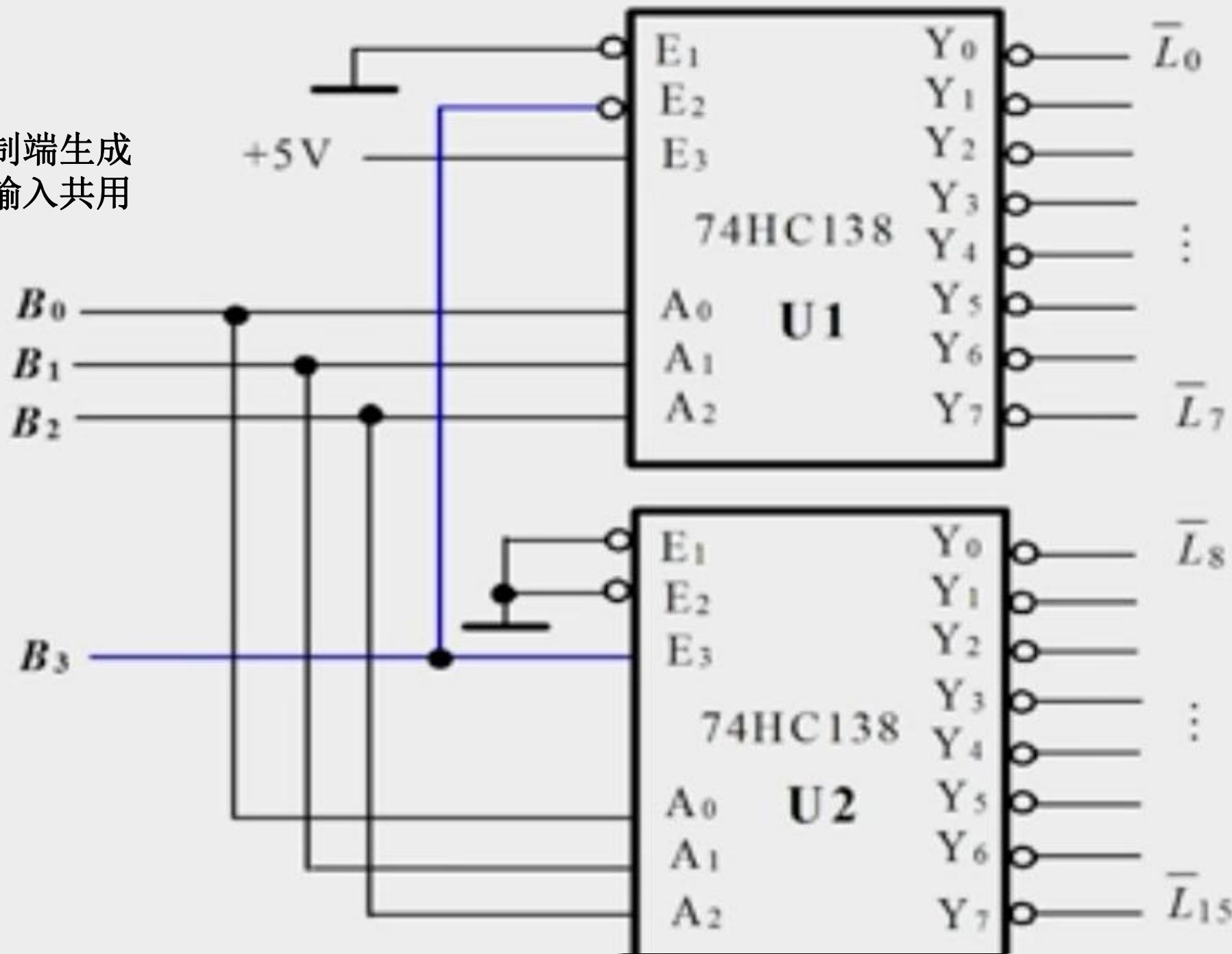
$$\begin{aligned}L &= \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC \\&= \overline{\overline{m}_0 + m_2 + m_6 + m_7} \\&= \overline{\overline{Y}_0 \cdot \overline{Y}_2 \cdot \overline{Y}_6 \cdot \overline{Y}_7}\end{aligned}$$

在译码器的输出端加一个与非门，即可实现给定的组合逻辑函数。

4. 译码器的扩展

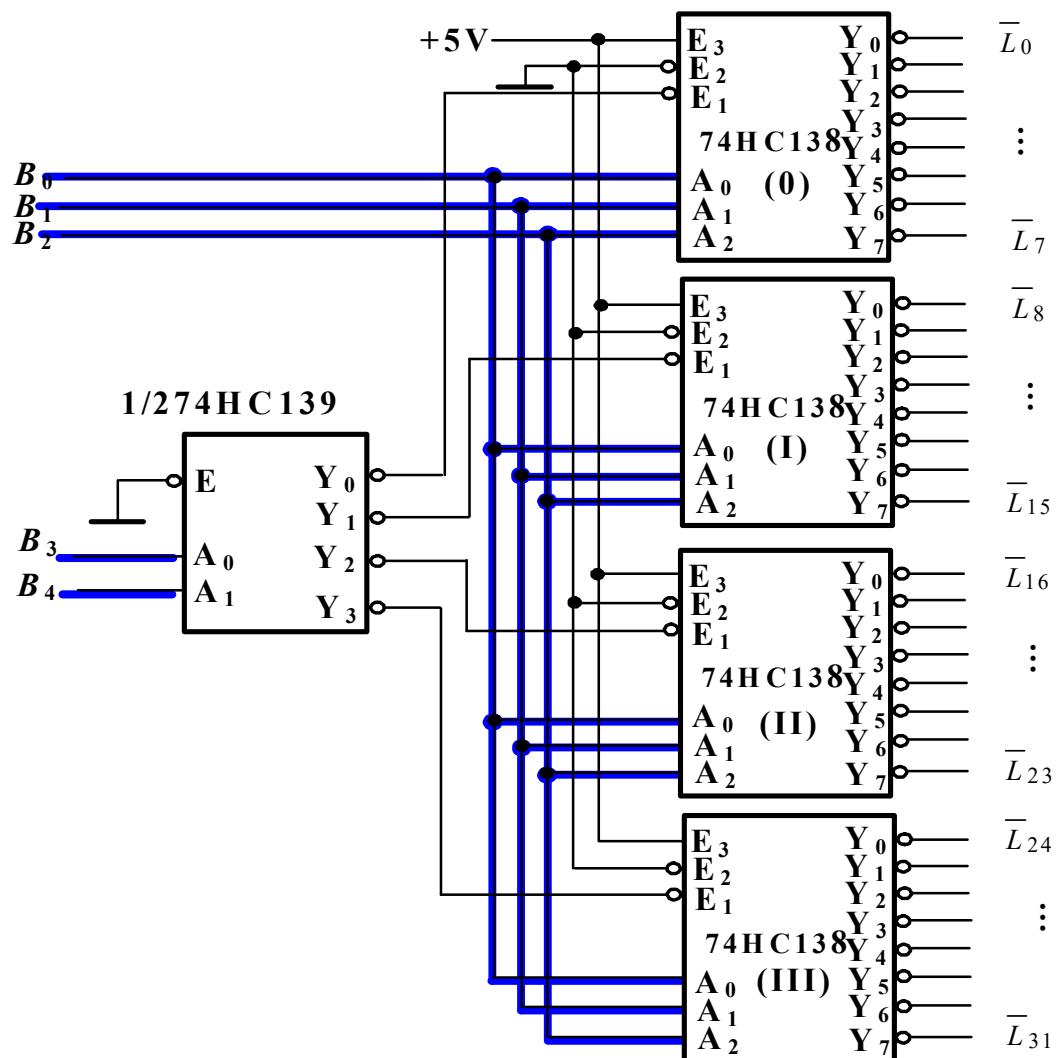
用74X138构成4线-16线译码器

高位端：控制端生成
低位端：片输入共用



用74X139和74X138构成5线-32线译码器

高位端：控制端生成
低位端：片输入共用



$B_4B_3 = 00$ 时，片 (0) 译码

$B_4B_3 = 01$ 时，片 (1) 译码

$B_4B_3 = 10$ 时，片 (2) 译码

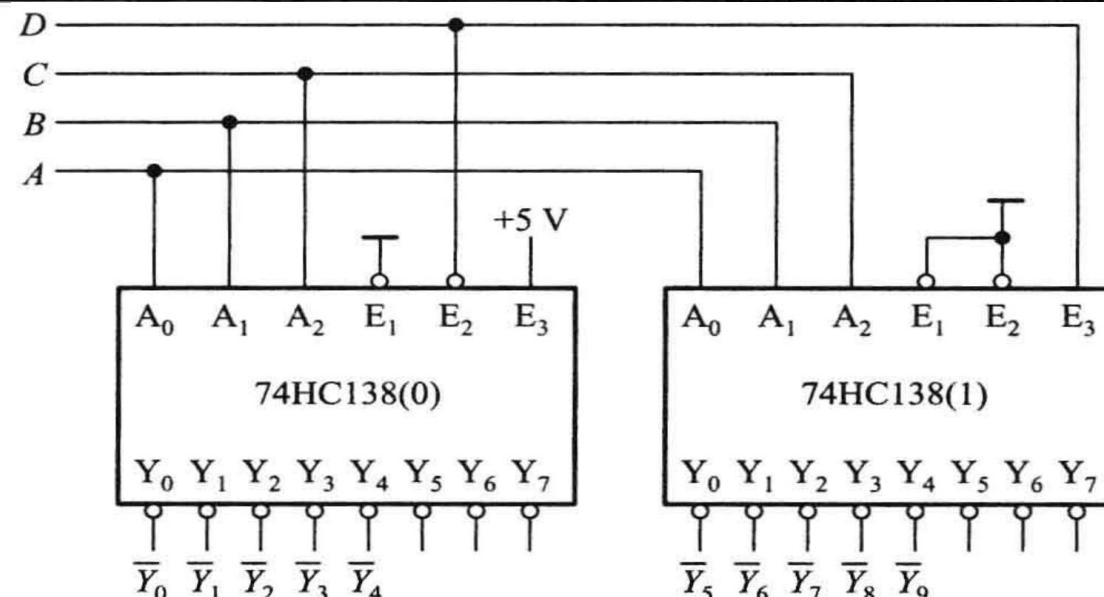
$B_4B_3 = 11$ 时，片 (3) 译码

作业：4.4.6 4.4.7

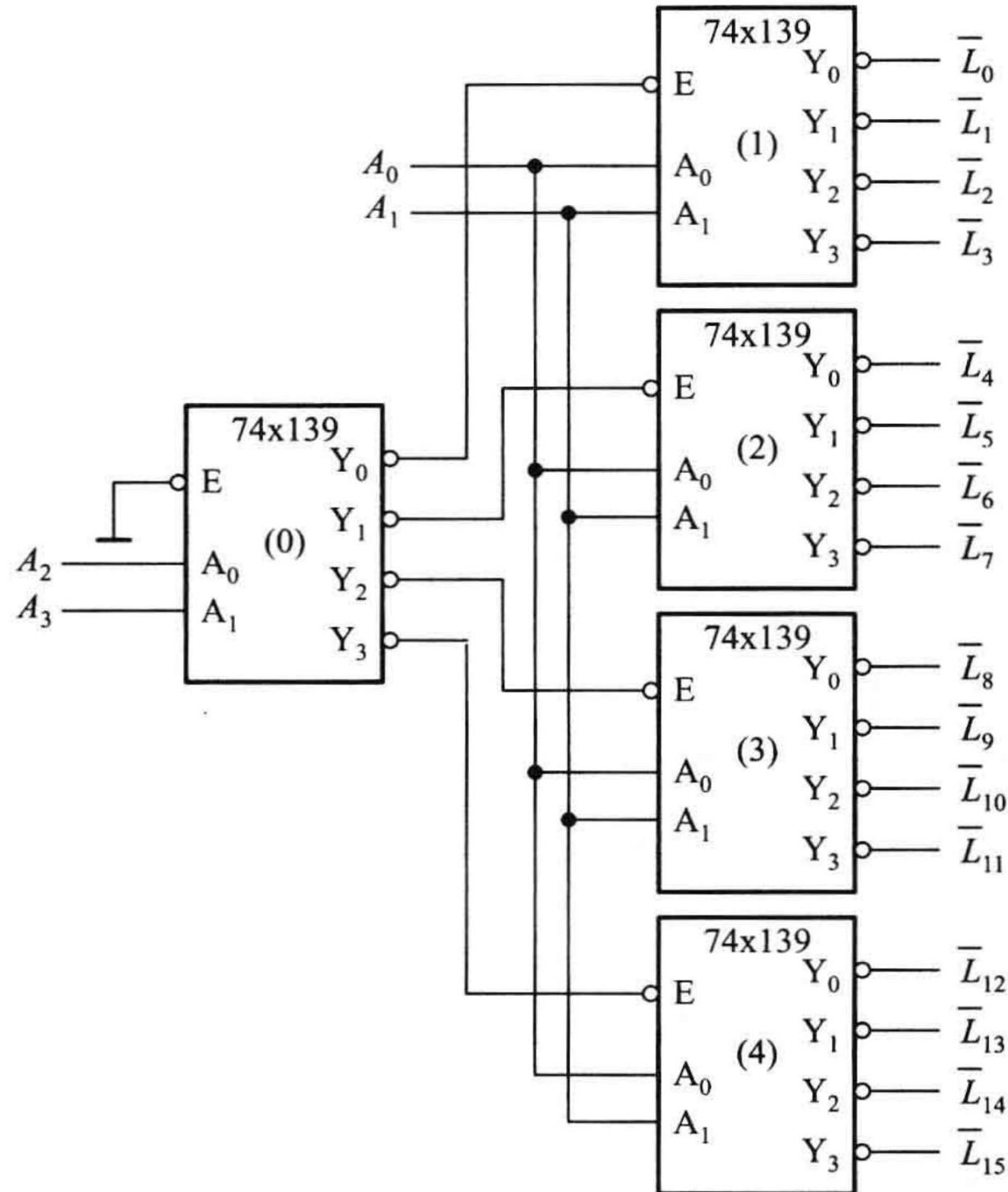
4.4.6 译码器的真值表如表题 4.4.6 所示, 试用 74HC138 实现该译码器。

表题 4.4.6 译码器的真值表

选择输入				译码输出									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	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	1	1	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	1	1	1	1	0	1
1	1	0	0	1	1	1	1	1	1	1	1	1	0



4.4.7 2线-4线译码器74x139的输入为高电平有效,使能输入及输出均为低电平有效。试用74x139构成4线-16线译码器。



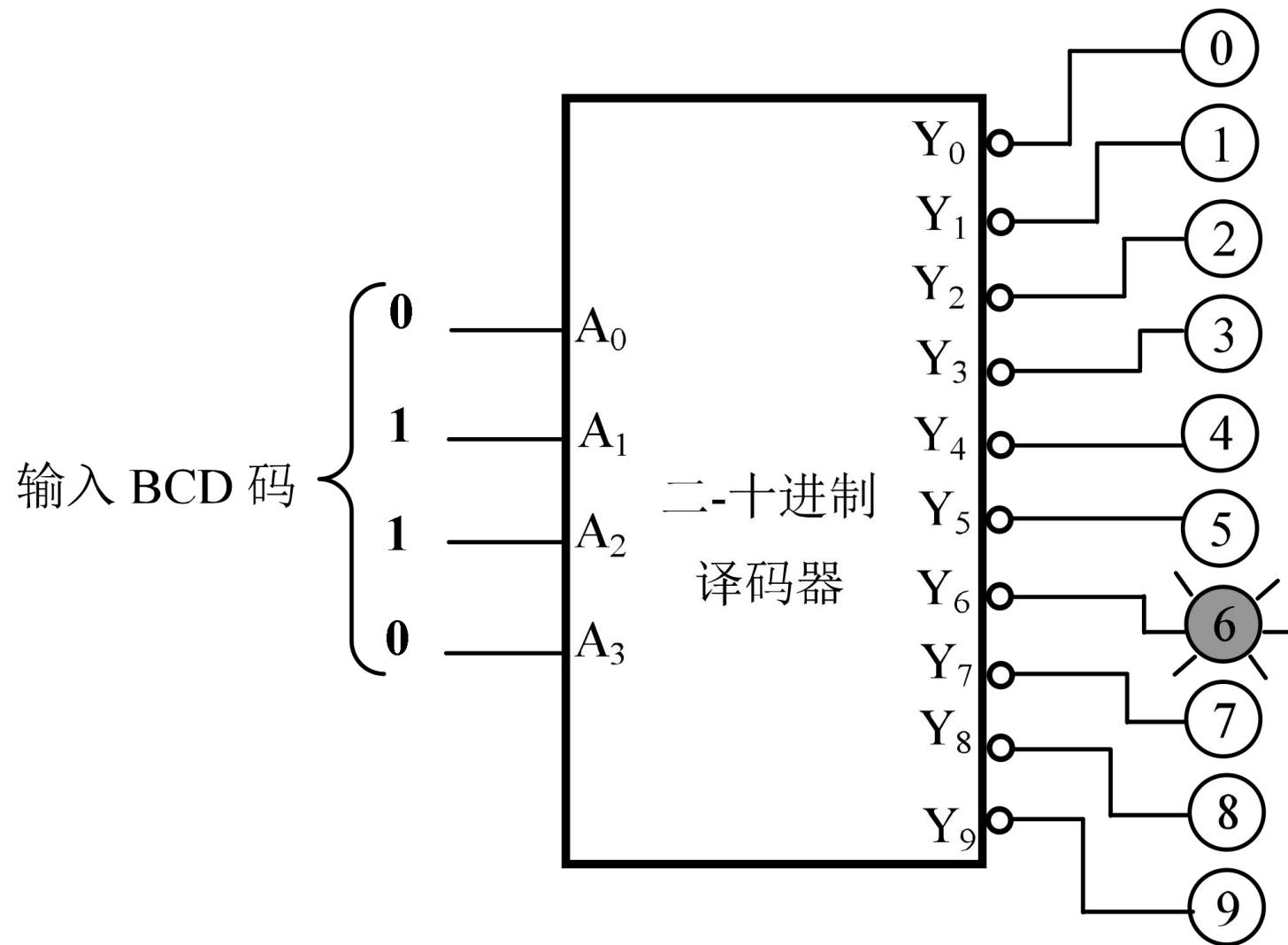
5. 二-十进制译码器

用四位二进制数0000-1001表示对应的0-9的十进制数。

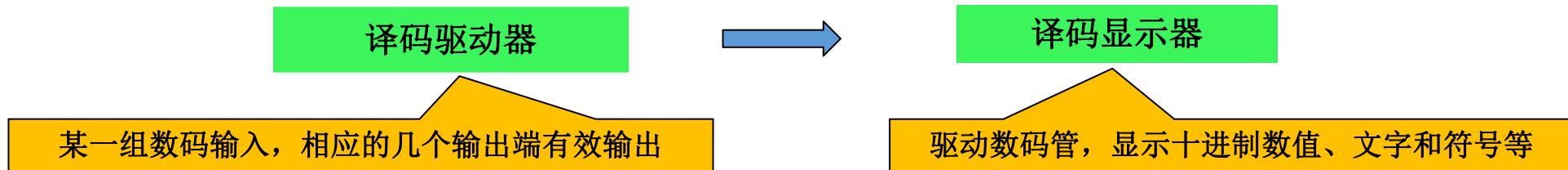
表 4.4.8 二-十进制译码器真值表

二-十进制译码器应用

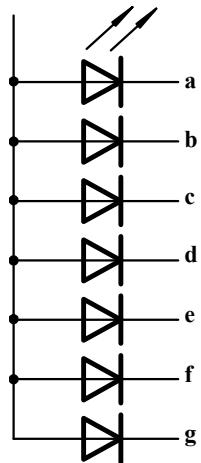
功能：将8421BCD码译成为10个状态输出。



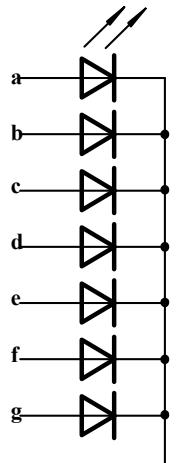
6. 显示译码器----七段显示译码器



最常用的显示器有：半导体发光二极管和液晶显示器。



共阳极显示器



共阴极显示器

阳级连在一起接高电平

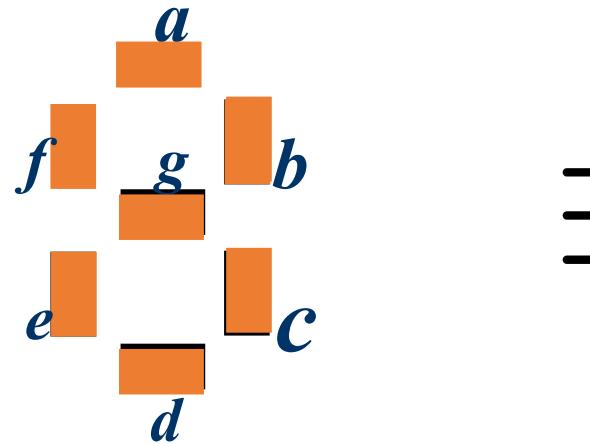
阴级连在一起接低电平

功能：将输入的4位BCD码翻译成“七段码”送给数码管，以便数码管能够显示各个BCD码所对应的十进制数字。

一类译码器的输出是高电平有效，
用来驱动共阴极显示器。

显示
译码器

另一类译码器输出的是低电平有效，
用来驱动共阳极显示器。



显示器分段布局图

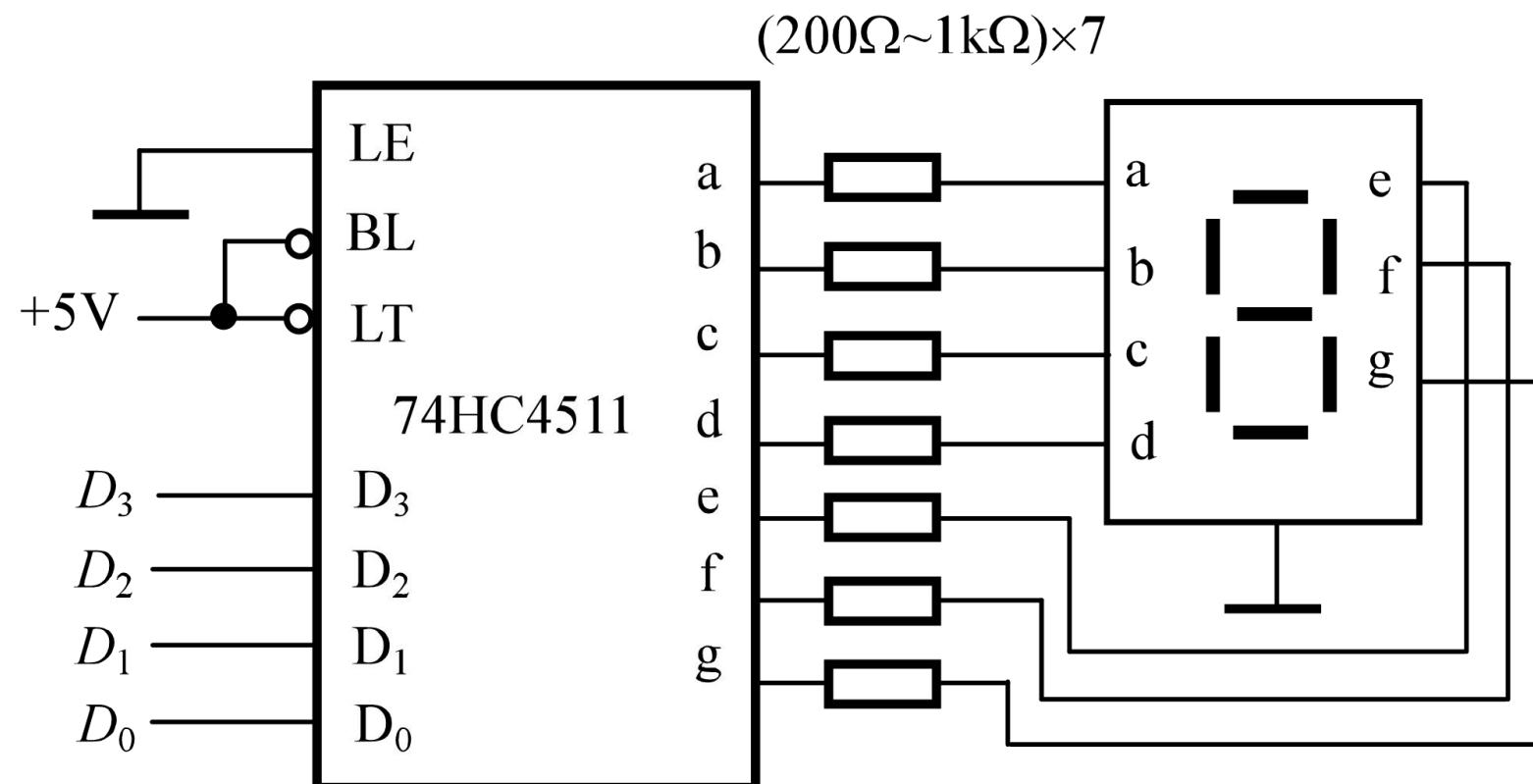
七段数码显示器显示（0~9）的情况：

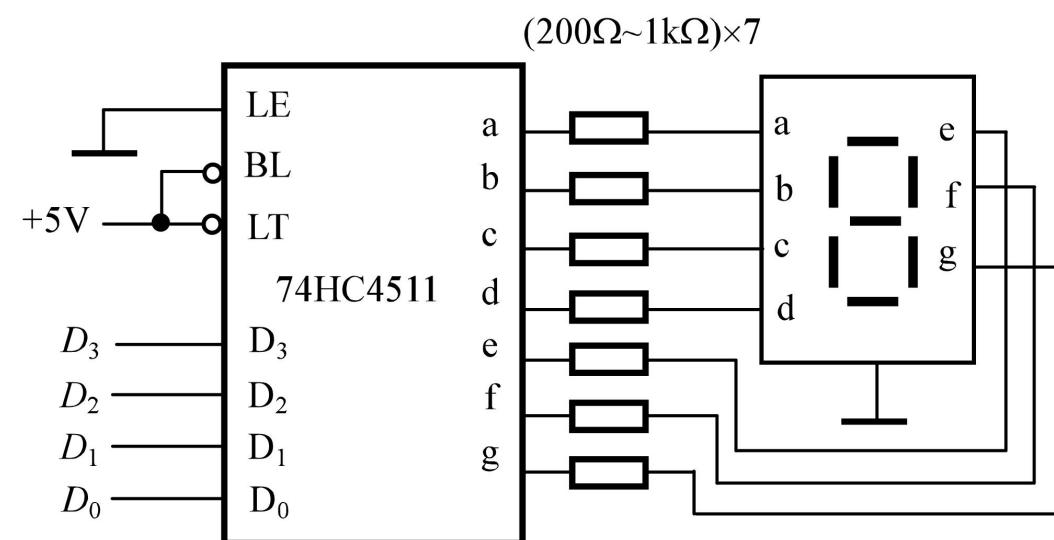
$D_3\ D_2\ D_1\ D_0$	a	b	c	d	e	f	g
0 0 0 0	1	1	1	1	1	1	0
0 0 0 1	0	1	1	0	0	0	0
0 0 1 0	1	1	0	1	1	0	1
0 0 1 1	1	1	1	1	0	0	1
0 1 0 0	0	1	1	0	0	1	1
0 1 0 1	1	0	1	1	0	1	1
0 1 1 0	0	0	1	1	1	1	1
0 1 1 1	1	1	1	0	0	0	0
1 0 0 0	1	1	1	1	1	1	1
1 0 0 1	1	1	1	0	0	1	1

常用的集成七段显示译码器

-----CMOS七段显示译码器**74HC4511**

显示译码器与显示器的连接方式



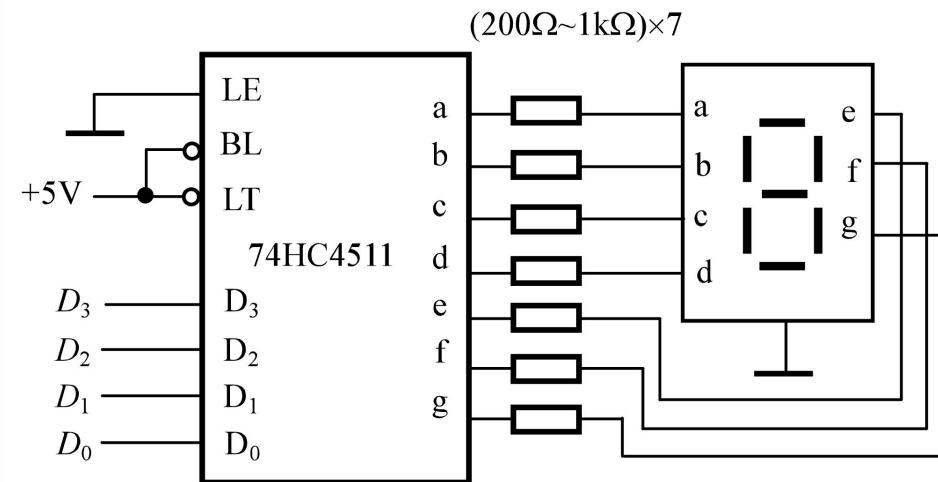


CMOS七段显示译码器74HC4511功能表

锁存端 灭灯端 灯测试端

十进制 或功能	输入						输出							字形	
	LE	BL	LT	D ₃	D ₂	D ₁	D ₀	a	b	c	d	e	f	g	
0	0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
3	0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
7	0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
8	0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
9	0	1	1	1	0	0	1	1	1	1	1	0	1	1	9

共阴



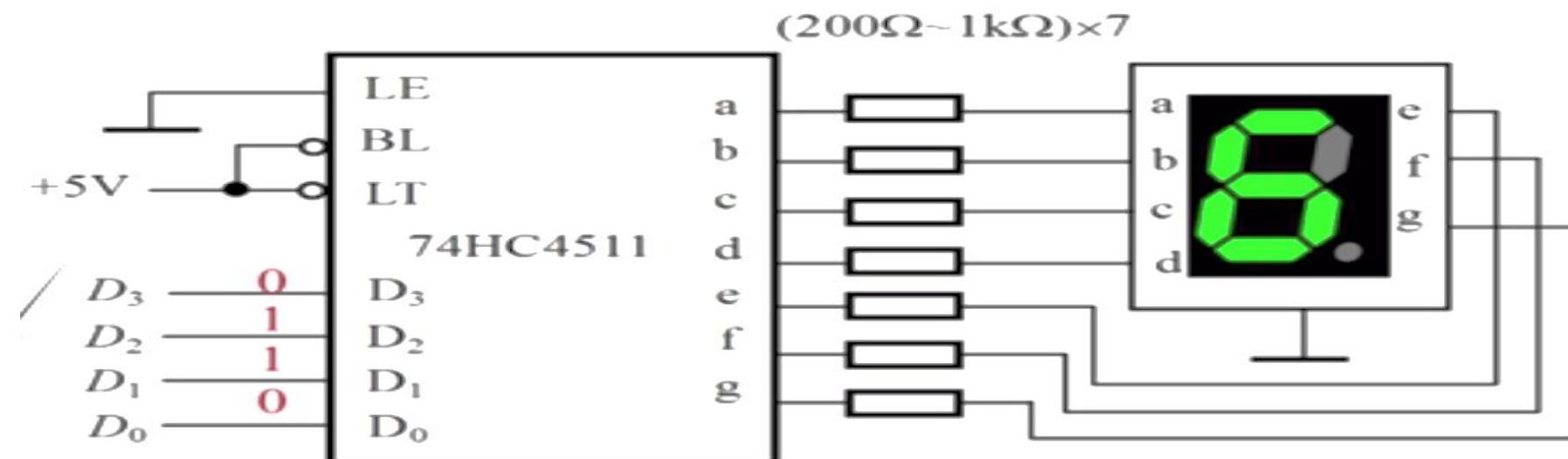
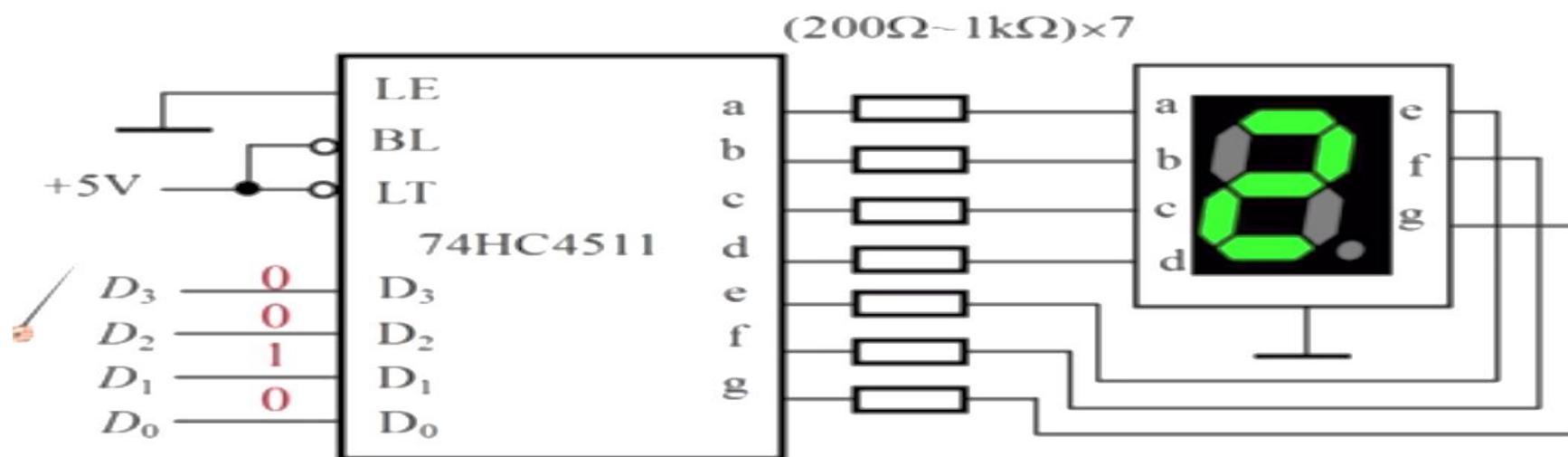
CMOS七段显示译码器74HC4511功能表(续)

锁存端 灭灯端 灯测试端

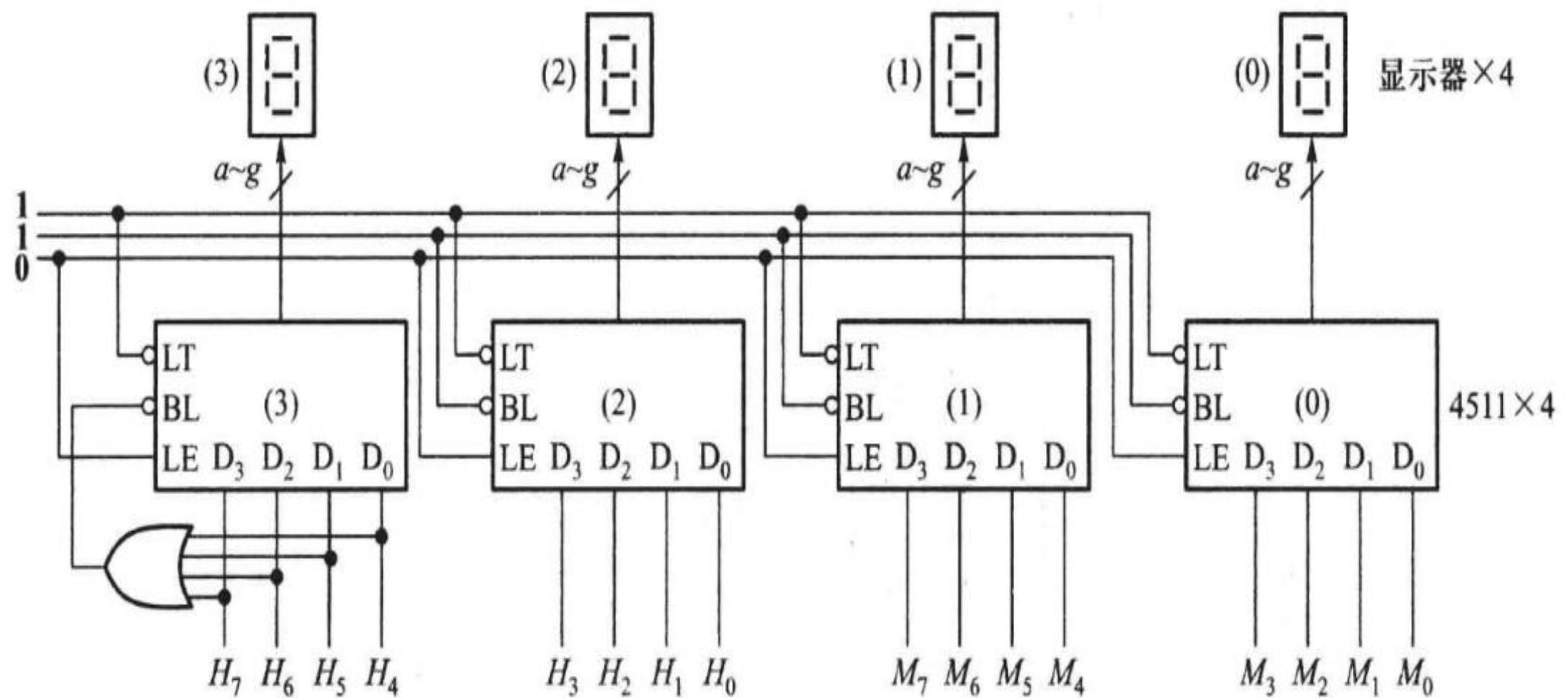
显示电路：

1. 静态显示电路：一个译码器与一个显示器相连接。
2. 动态显示电路：一个译码器与多个显示器相连接，再增加显示控制电路。

静态显示电路：



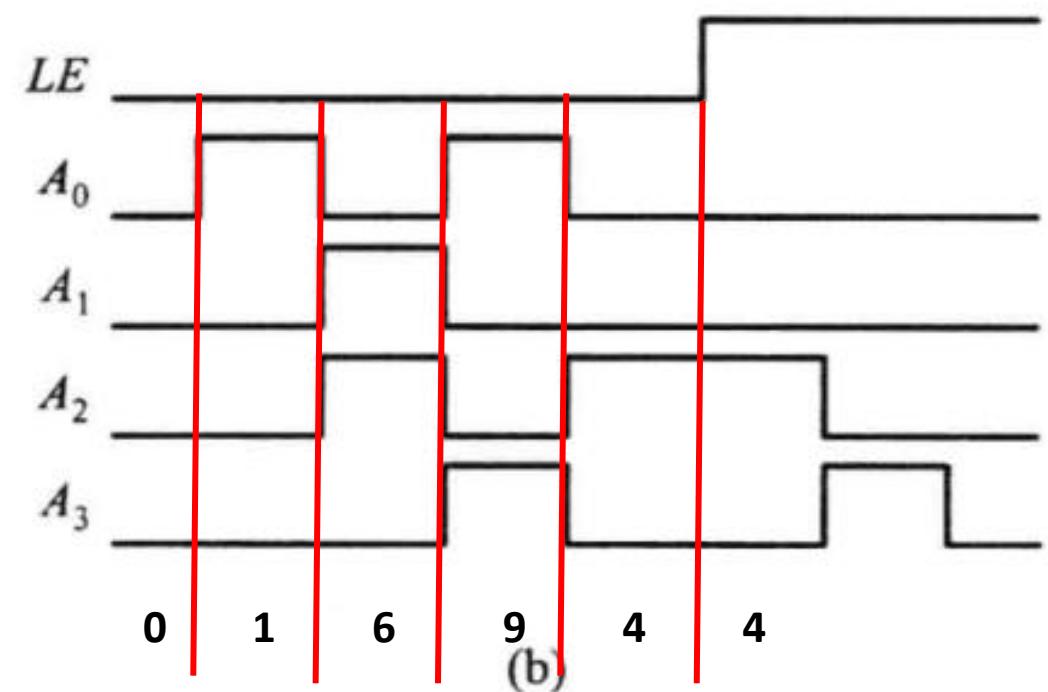
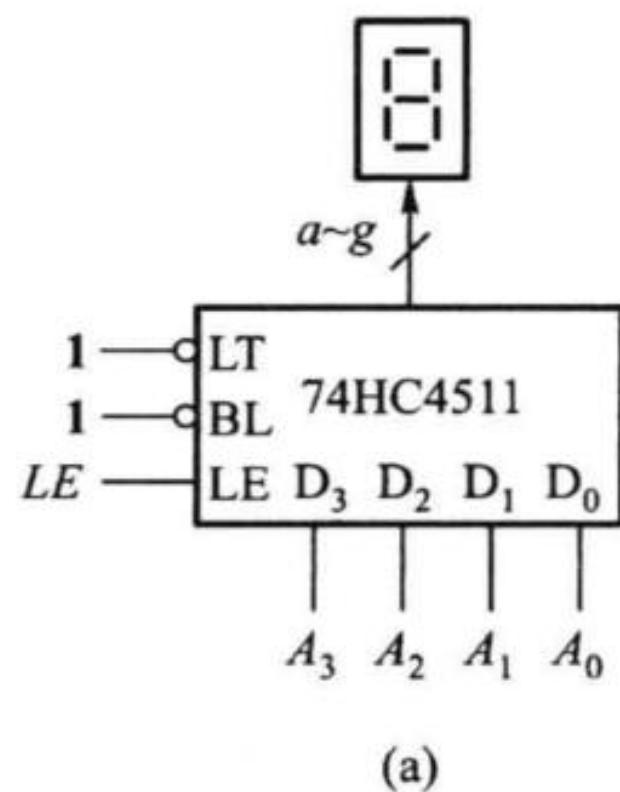
4.4.13 由 74HC4511 构成 24 小时及分钟的译码电路如图题 4.4.13 所示, 试分析小时高位是否具有零熄灭功能。



图题 4.4.13

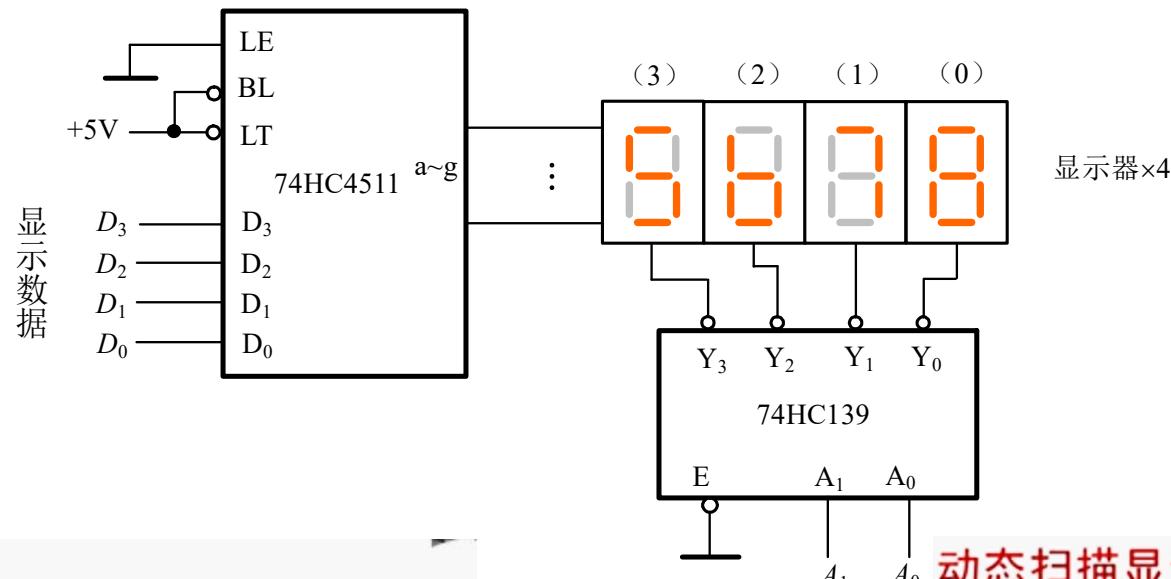
具有灭灯功能

4.4.14 七段显示译码电路如图题 4.4.14(a) 所示, 对应图题 4.4.14(b) 所示输入波形, 试确定显示器显示的字符序列是什么?

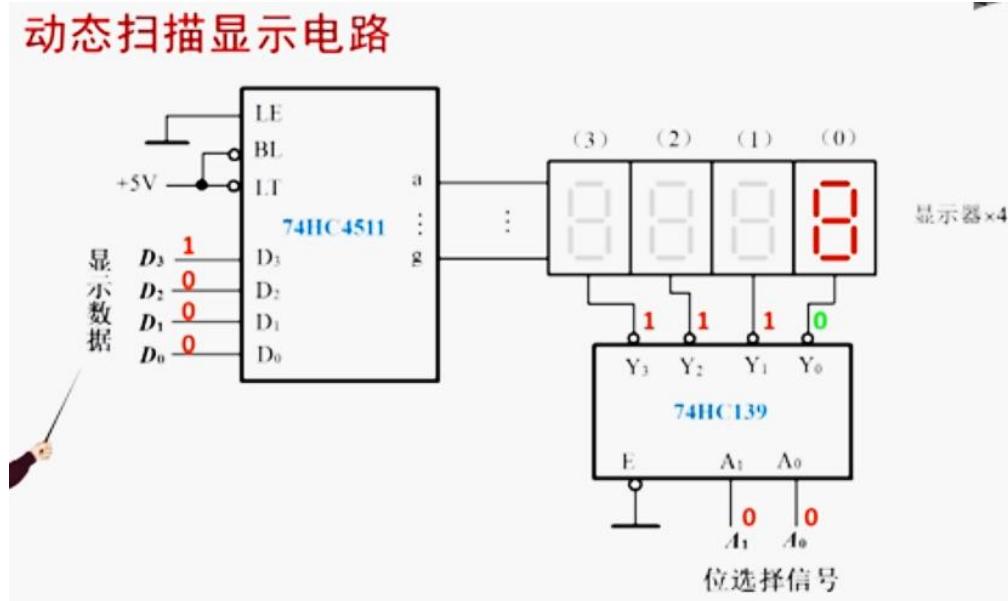


动态显示电路：一个译码器与多个显示器相连接，再增加显示控制电路

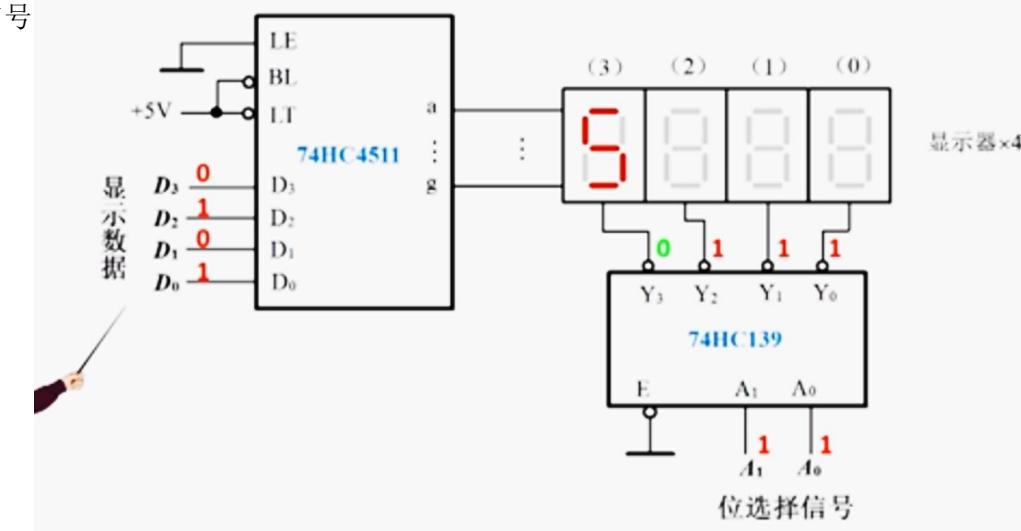
例4.4.5 由译码器、显示译码及4个七段显示器构成的4位动态显示电路如图所示，试分析工作原理。



动态扫描显示电路



A₁ A₀ 动态扫描显示电路



位选择信号A1、A0控制 $\overline{Y_3} \sim \overline{Y_0}$ 依次产生低电平，使4个显示器轮流显示。

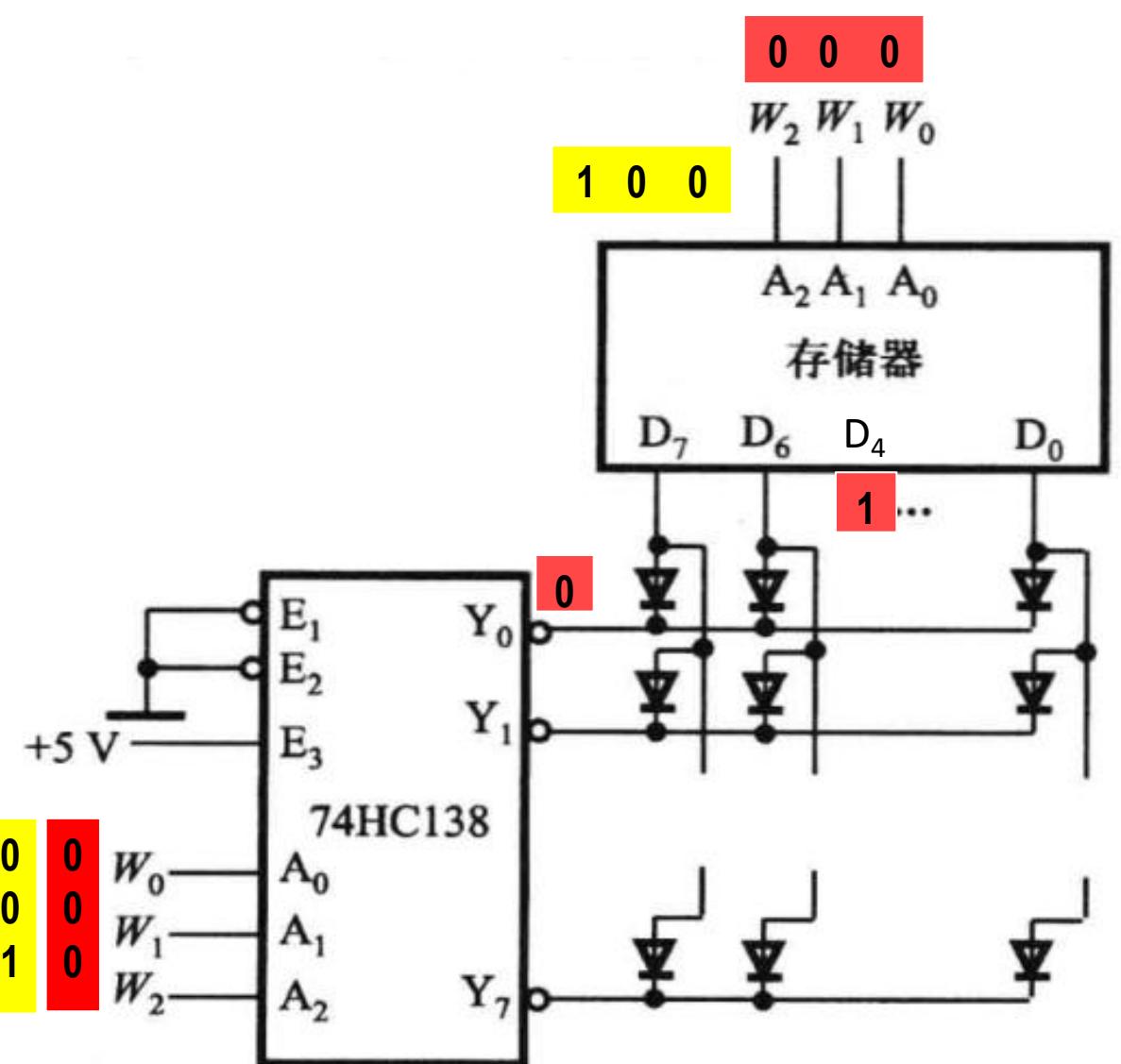
要显示的数据组依次送到D3、D2、D1、D0 分别在4个显示器上显示。

利用人的视觉暂留时间，可以看到稳定的数字。

显示器频率 $25\text{Hz} < f_C < 100\text{Hz}$

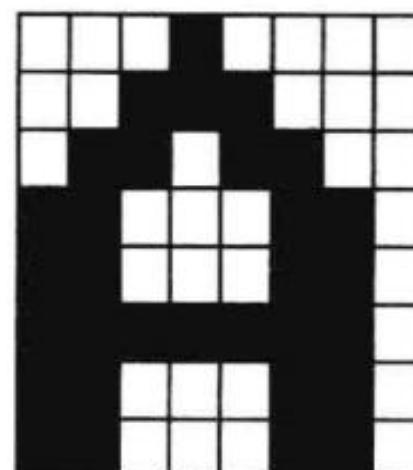
4.4.13 4.4.14 4.4.15

4.4.15 图题 4.4.15 所示为 8×8 个 LED 阵列显示示意图。3 线—8 线译码器控制逐行扫描,从上到下每次显示一行。存储阵列共有 8×8 个存储单元,每个单元存放 1 位显示的数据,需要显示的点存 1,否则存 0。地址线 W_2, W_1, W_0 从 000 到 111 变化时,每次将一组 8 个数据送到输出端,控制发光二极管,需要发光的二极管接 1,否则接 0。如要显示的字型如图题 4.4.15(b) 所示,试写出存储器存放的数据。若人的视觉暂留时间为 0.05 秒,在满足 LED 阵列图像稳定不闪烁的情况下,试计算地址变换的最低频率。



存储器存放的数据功能表:

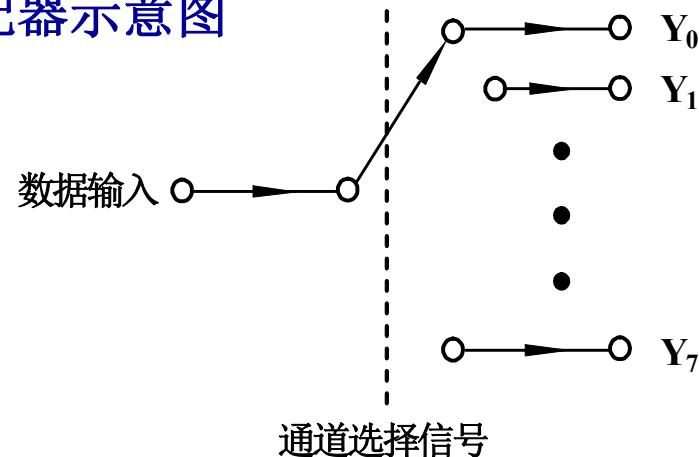
W_2	W_1	W_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	1	0	0	0	0
0	0	1	0	0	1	1	1	0	0	0
0	1	0	0	1	1	0	1	1	0	0
0	1	1	1	1	0	0	0	1	1	0
1	0	0	1	1	0	0	0	1	1	0
1	0	1	1	1	1	1	1	1	1	0
1	1	0	1	1	0	0	0	1	1	0
1	1	1	1	1	0	0	0	1	1	0



7. 数据分配器

用途：一个PC机，与多个外部设备的连接

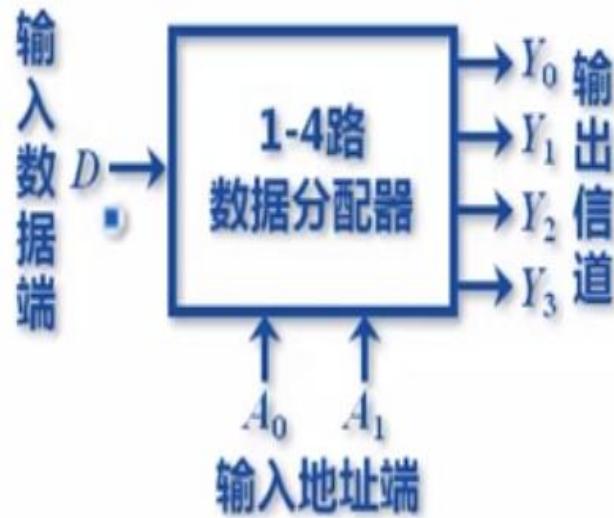
数据分配器示意图



数据分配器：相当于多输出的单刀多掷开关，是将公共数据线上的数据按需要送到不同的通道上。



1-4 路数据分配器



真值表

数据	地址		输出			
	A_1	A_0	Y_0	Y_1	Y_2	Y_3
D	0	0	D	0	0	0
D	0	1	0	D	0	0
D	1	0	0	0	D	0
D	1	1	0	0	0	D

$$Y_0 = \bar{A}_1 \bar{A}_0 D \quad Y_1 = \bar{A}_1 A_0 D$$

$$Y_2 = A_1 \bar{A}_0 D \quad Y_3 = A_1 A_0 D$$

数据	地址		输出			
	A_1	A_0	Y_0	Y_1	Y_2	Y_3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

2-4 线译码器

$$Y_0 = \bar{A}_1 \bar{A}_0 \quad Y_1 = \bar{A}_1 A_0$$

$$Y_2 = A_1 \bar{A}_0 \quad Y_3 = A_1 A_0$$

数据分配器可以用带有使能端的二进制译码器实现

1-8 路数据分配器

先寻址

3 位输入地址
8 路输出信道

再传输

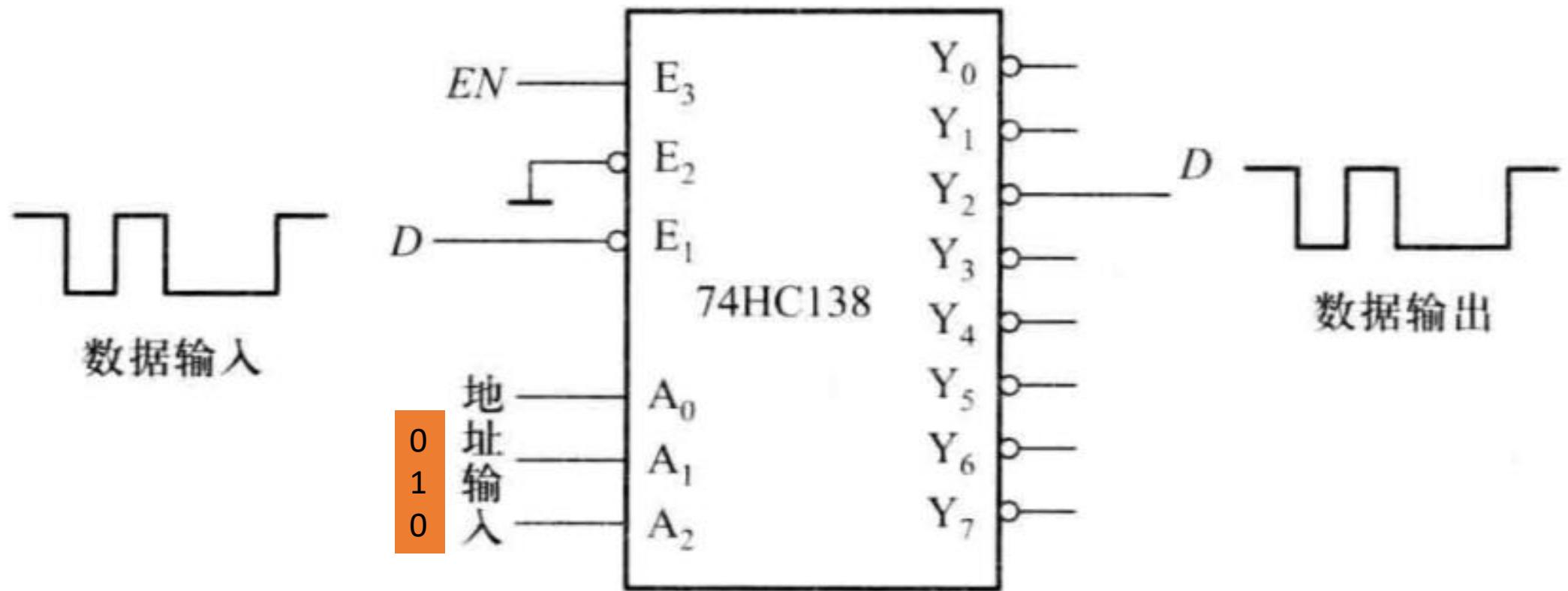


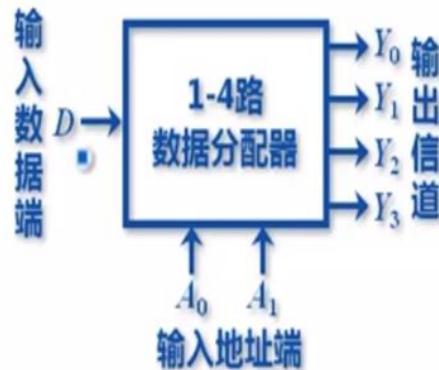
图 4.4.19 用 3 线-8 线译码器作为数据分配器

数据输入端：由使能控制端完成
寻址端：由地址输入端完成

3线-8线译码器可以把1个数据信号分配到8个不同的通道上去。

例：试用门电路设计一个具有低电平使能控制的1线-4线数据分配器，使能信号无效时，电路所有的输出为高阻态。当通道选择信号将1路输入信号连接到其中1路输出端时，其他输出端为高阻态。

1-4 路数据分配器



1. 列真值表

输出端有3种状态（0、1、z），输出级是4个三态门组成。其控制信号由 \bar{E} 、 S_1 、 S_0 共同作用产生。

输入			输出			
\bar{E}	S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	z	z	z	In
0	0	1	z	z	In	z
0	1	0	z	In	z	z
0	1	1	In	z	z	z
1	x	x	z	z	z	z

2. 写出4个三态门控制端的逻辑表达式

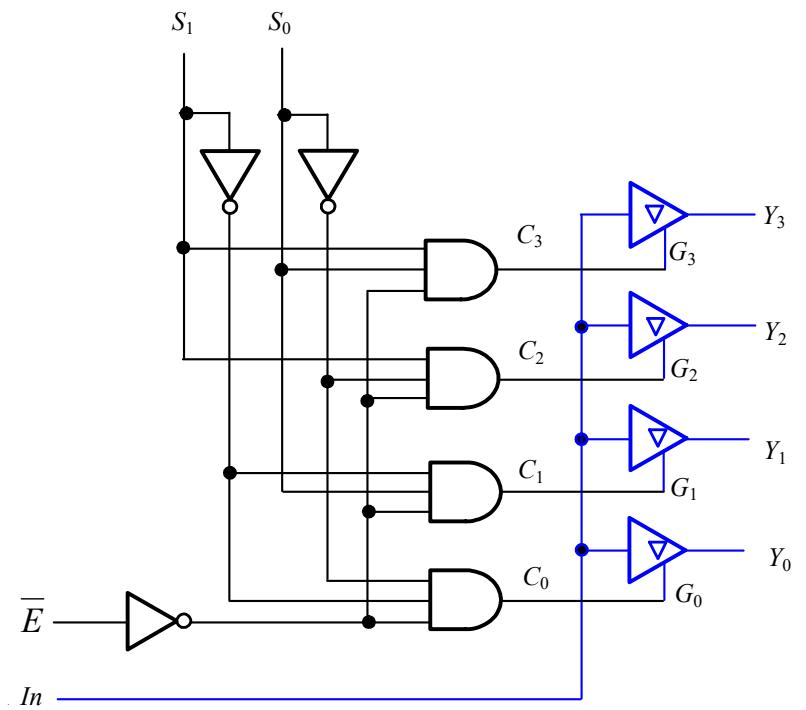
$$C_0 = \overline{\overline{E}} \cdot \overline{S_1} \cdot \overline{S_0}$$

$$C_1 = \overline{\overline{E}} \cdot \overline{S_1} \cdot S_0$$

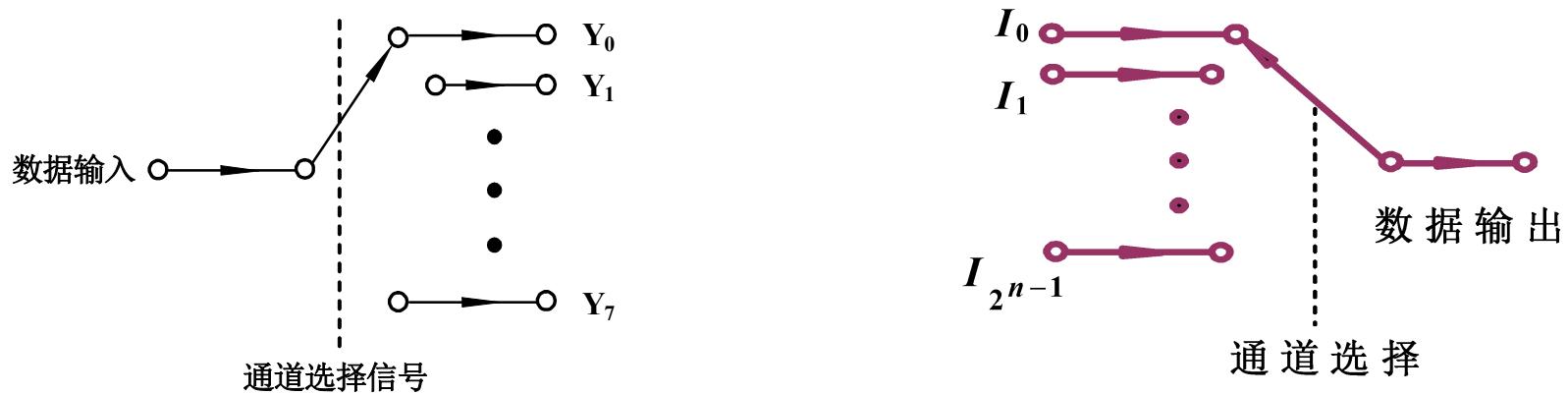
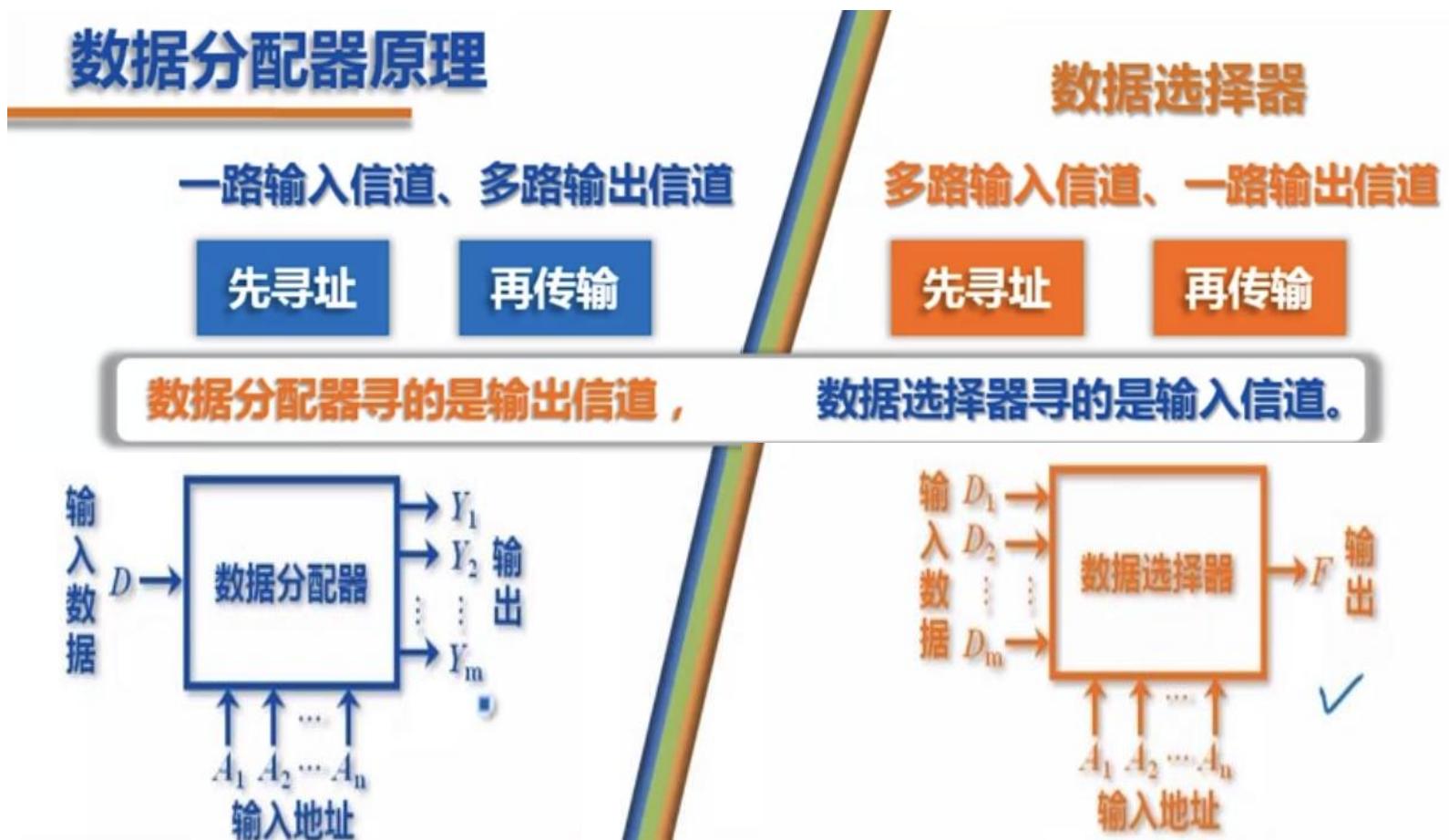
$$C_2 = \overline{\overline{E}} \cdot S_1 \cdot \overline{S_0}$$

$$C_3 = \overline{\overline{E}} \cdot S_1 \cdot S_0$$

3. 画逻辑电路



4.4.3 数据选择器



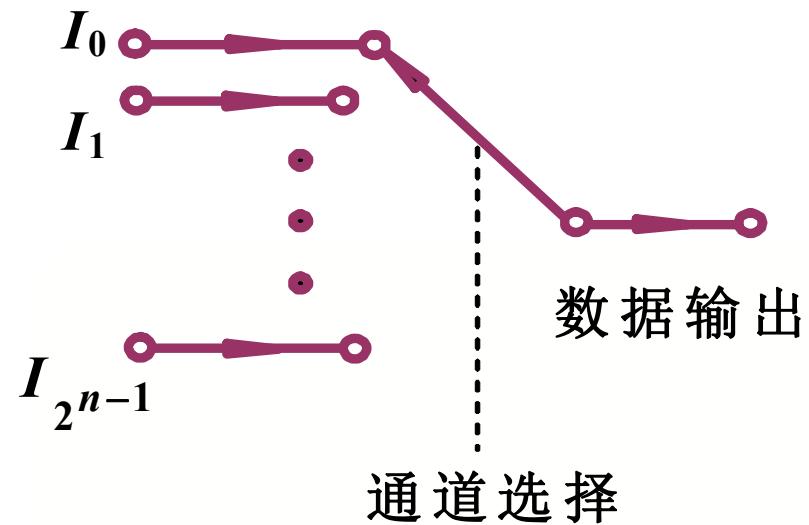
1、数据选择器的定义与功能

数据选择器：能实现数据选择功能的逻辑电路。

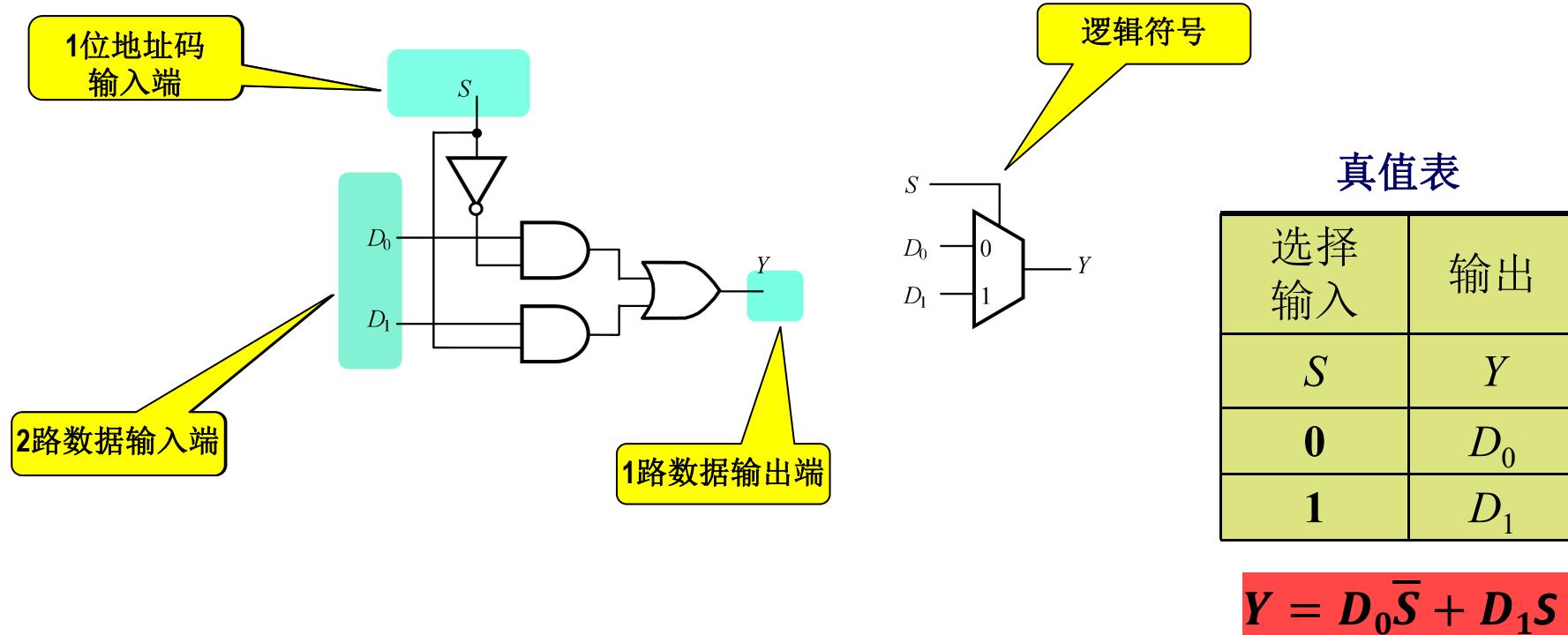
它的作用相当于多个输入的单刀多掷开关，又称“多路开关”。

数据选择的功能：

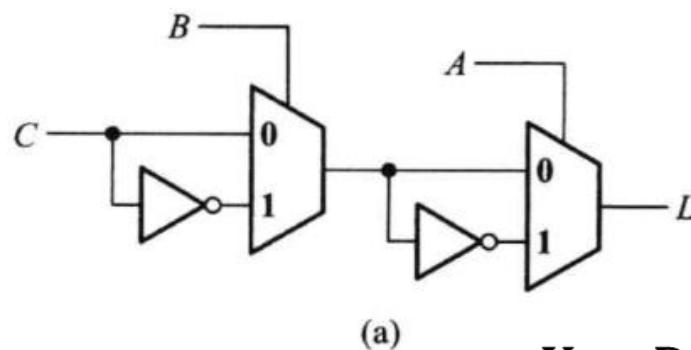
在通道选择信号的作用下，将多个通道的数据分时传送到公共的数据通道上去的。



2选1数据选择器



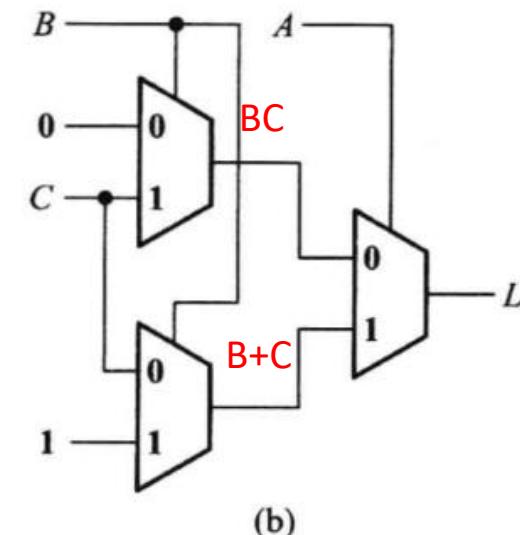
4.4.20 电路如图，试写出输出端的逻辑函数式



(a)

$$Y = D_0 \bar{S} + D_1 S$$

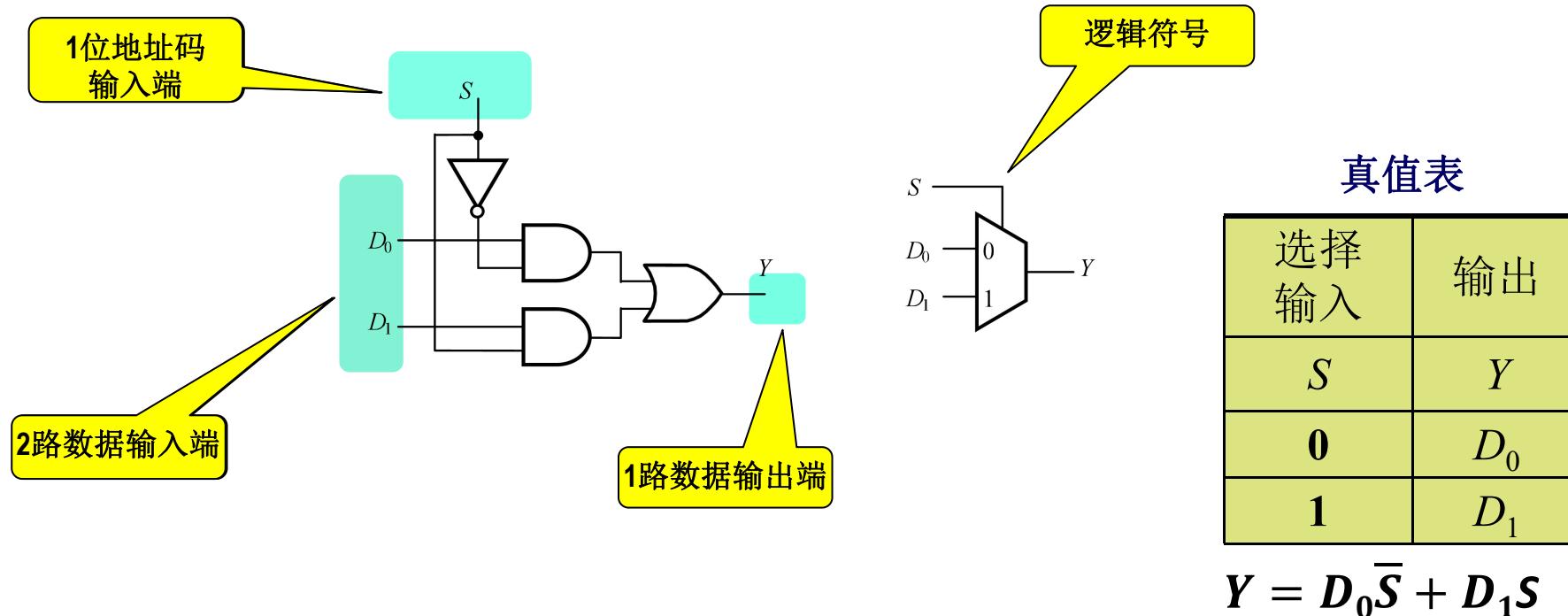
$$L = A \oplus B \oplus C$$



(b)

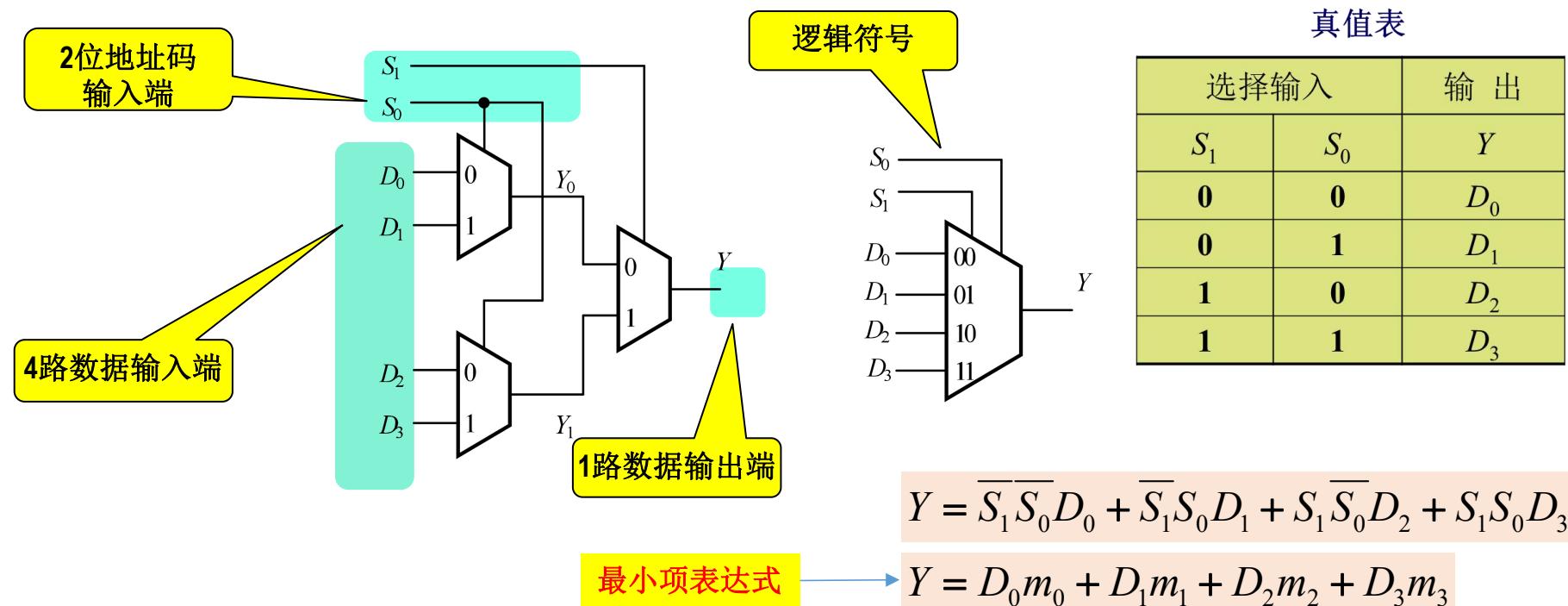
$$\begin{aligned} L &= \overline{ABC} + (B + C) A \\ &= BC + AB + AC \end{aligned}$$

2选1数据选择器



4选1数据选择器

逻辑电路 ---由3个2选1数据选择器构成4选1数据选择器。



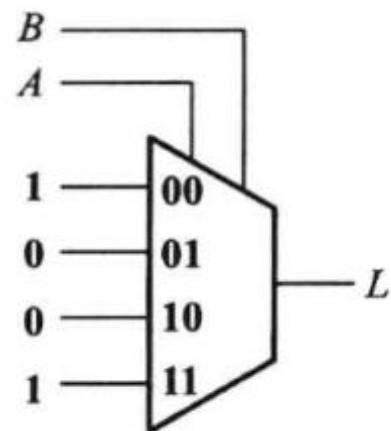
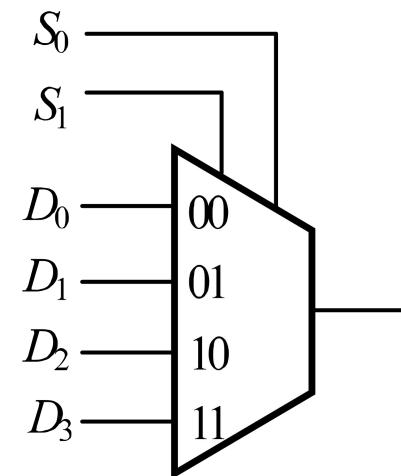
2. 数据选择器实现逻辑函数

4.4.22 试用 4 选 1 数据选择器产生逻辑函数：

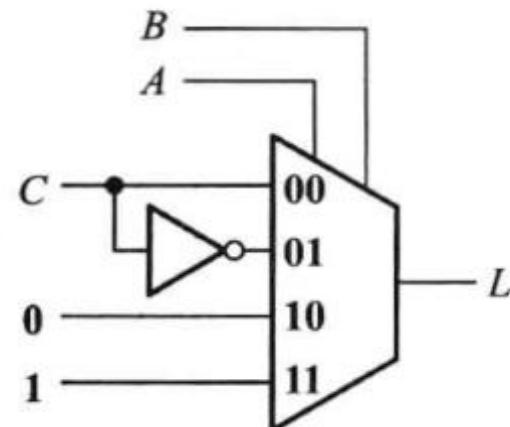
$$(1) L(A, B) = \overline{A} \overline{B} + AB$$

$$(2) L(A, B, C) = \sum m(1, 2, 6, 7)。$$

$$Y = \overline{S_1} \overline{S_0} D_0 + \overline{S_1} S_0 D_1 + S_1 \overline{S_0} D_2 + S_1 S_0 D_3$$



(a)



(b)

总结：

利用数据选择器实现函数的一般步骤：（变量数=地址端数）

a、将函数变换为最小项表达式

b、地址信号 S_1 、 S_0 作为函数的输入变量

c、逻辑表达式中有 m_i , 则相应 $D_i = 1$, 其他的数据输入端均为 0。

当变量数 $A, B, C >$ 地址端数

将多的变量接入数据输入端

将函数变换为最小项表达式

例题：试用数据选择器实现下列逻辑函数

① 用4选1数据选择器实现

$$L_0 = \overline{A}\overline{B} + A\overline{B}$$

用4选1数据选择器实现

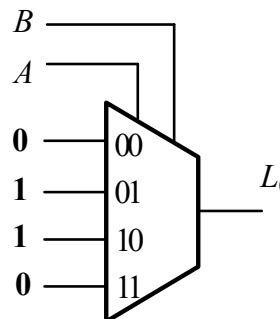
$$L_1 = AB + AC + BC$$

② 用2选1数据选择器和必要的逻辑门实现 $L_1 = AB + A\overline{C} + BC$

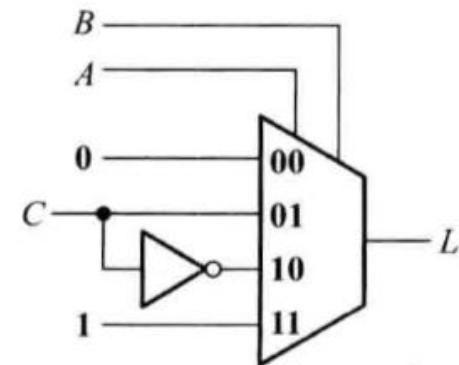
$$Y = \overline{S_1}\overline{S_0}D_0 + \overline{S_1}S_0D_1 + S_1\overline{S_0}D_2 + S_1S_0D_3$$

① 当 $S_1 = A, S_2 = B,$

$$D_0 = D_3 = 0, D_1 = D_2 = 1$$



$$\begin{aligned} L_1 &= AB + A\overline{C} + BC \\ &= A\overline{B}\overline{C} + \overline{ABC} + AB \end{aligned}$$

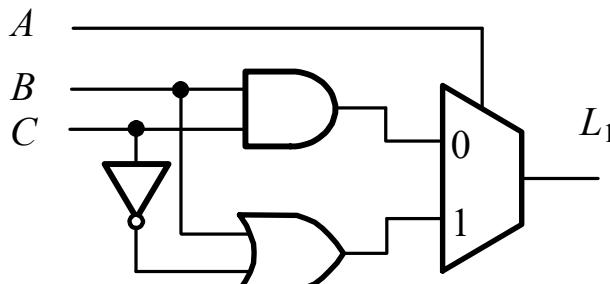


② 2选1数据选择器只有1个选通端接输入A，表达式有3个变量。

因此数据端需要输入2个变量。考察真值表B、C与 L_1 的关系。

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

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



(c)

输入			输出	
A	B	C	L_1	
0	0	0	0	$L_1 = BC$
0	0	1	0	
0	1	0	0	
0	1	1	1	
1	0	0	1	$L_1 = B + \overline{C}$
1	0	1	0	
1	1	0	1	
1	1	1	1	

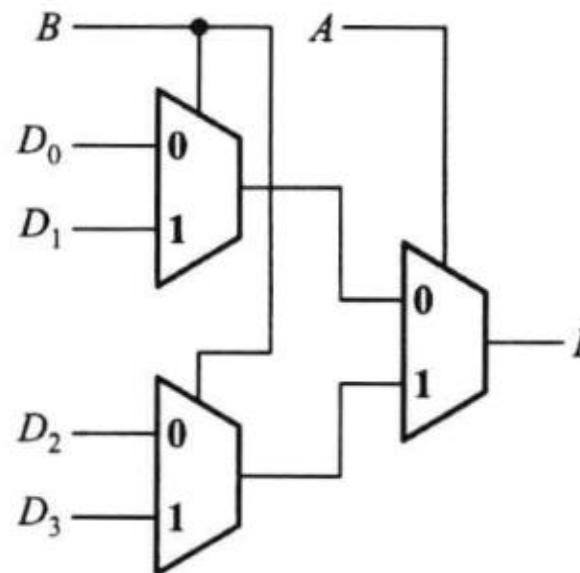
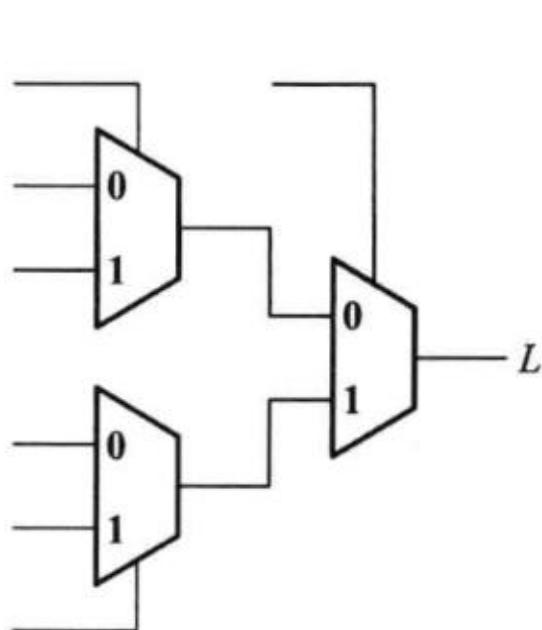
4.4.21 由 2 选 1 数据选择器构成的电路如图题 4.4.21 所示, 其数据输入端和选择输入端可接 0,1 或输入变量, 试用该电路实现下列逻辑函数。

$$(1) L(A, B, C) = AB + C$$

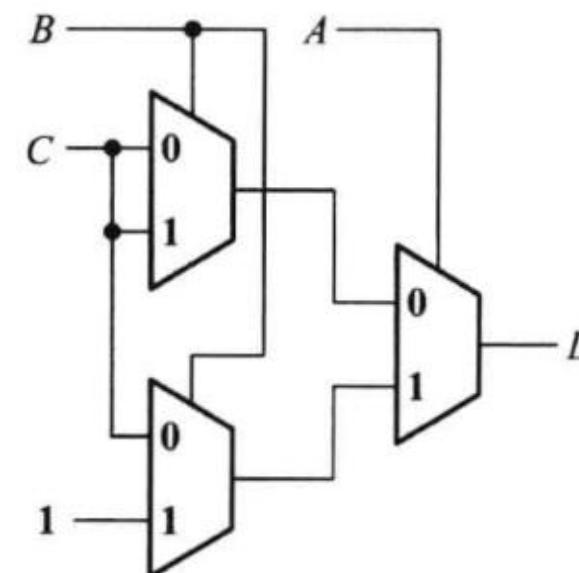
$$(2) L(A, B, C) = AC + BC$$

解: 用图题 4.4.21 所示电路实现逻辑函数的方法有很多种, 这里只介绍一种方法。

(1) 将变量 B 、 A 分别接入选择输入端, 如图题解 4.4.21(a) 所示。其输出端与数据输入端的关系如表题解 4.4.21(a) 所示。根据逻辑函数式列出的真值表如表题解 4.4.21(b) 所示。对比表题解 4.4.21(a) 和(b) 可以得出 $D_0 = D_1 = D_2 = C$, $D_3 = 1$, 电路如图题解 4.4.21(b) 所示。



(a)



(b)

$$\begin{aligned} & (D_0\bar{B} + D_1B)\bar{A} + (D_2\bar{B} + D_3B)A \\ & = \bar{A}\bar{B}D_0 + \bar{A}BD_1 + A\bar{B}D_2 + AB\bar{D}_3 \end{aligned}$$

$$\begin{aligned} L &= AB + C \\ &= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC \end{aligned}$$

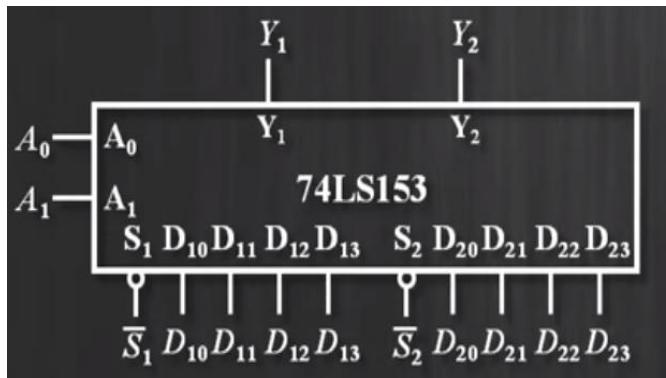
$$D_0 = D_1 = D_2 = C \quad D_3 = 1$$

$$\begin{aligned} L &= AC + BC \\ &= \bar{A}\bar{B}C + ABC + \bar{A}BC + ABC \\ &= \bar{A}BC + \bar{A}\bar{B}C + ABC \\ \therefore D_0 &= 0 \quad D_1 = D_2 = D_3 = C \end{aligned}$$

3. 集成数据选择器

4选1数据选择器----74x153

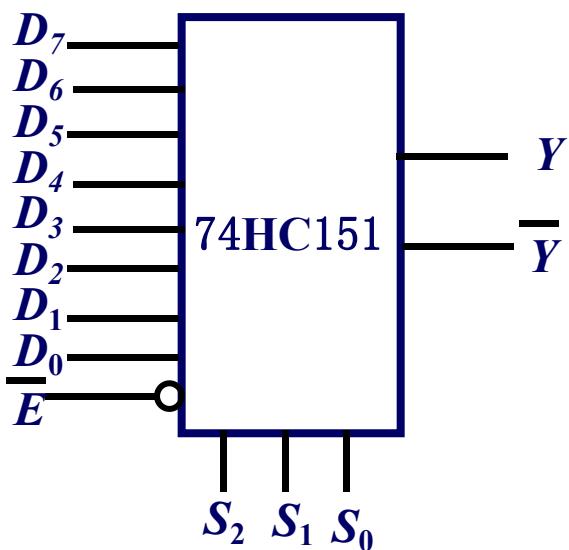
4+2+1



选通	地址	数据	输出
\bar{S}	$A_1 \ A_0$	D	Y
1	$\times \ \times$	\times	0
0	0 0	$D_0 \sim D_3$	D_0
0	0 1	$D_0 \sim D_3$	D_1
0	1 0	$D_0 \sim D_3$	D_2
0	1 1	$D_0 \sim D_3$	D_3

8选1数据选择器74HC151

8+3+1



74HC151逻辑符号

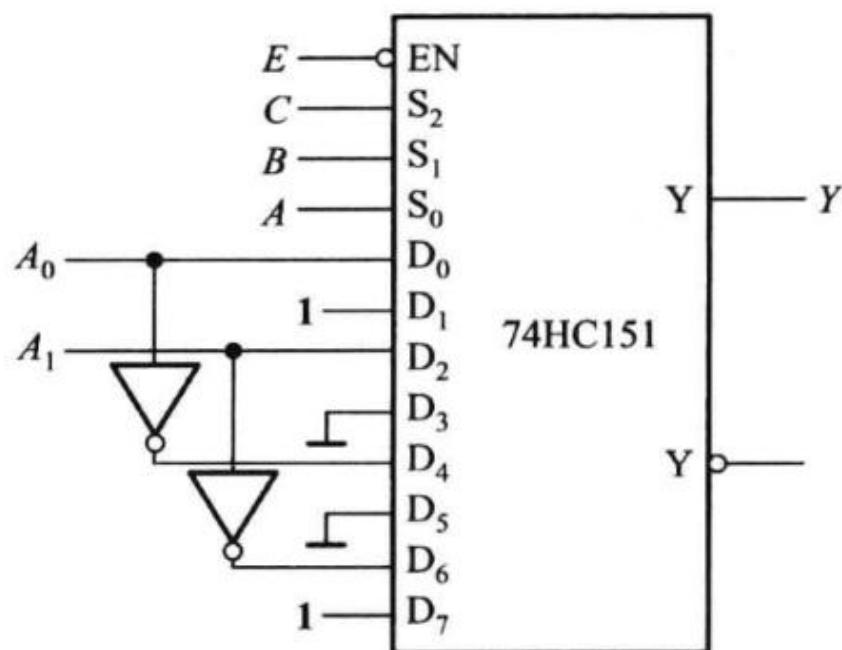
选通	地址	数据	输出
\bar{S}	$A_1 \ A_1 \ A_0$	D	$Y \ \bar{Y}$
1	$\times \ \times \ \times$	\times	0 1
0	0 0 0	$D_0 \sim D_7$	$D_0 \ \bar{D}_0$
0	0 0 1	$D_0 \sim D_7$	$D_1 \ \bar{D}_1$
0	0 1 0	$D_0 \sim D_7$	$D_2 \ \bar{D}_2$
0	0 1 1	$D_0 \sim D_7$	$D_3 \ \bar{D}_3$
0	1 0 0	$D_0 \sim D_7$	$D_4 \ \bar{D}_4$
0	1 0 1	$D_0 \sim D_7$	$D_5 \ \bar{D}_5$
0	1 1 0	$D_0 \sim D_7$	$D_6 \ \bar{D}_6$
0	1 1 1	$D_0 \sim D_7$	$D_7 \ \bar{D}_7$

$$Y = D_0m_0 + D_1m_1 + D_2m_2 + D_3m_3 \dots\dots$$

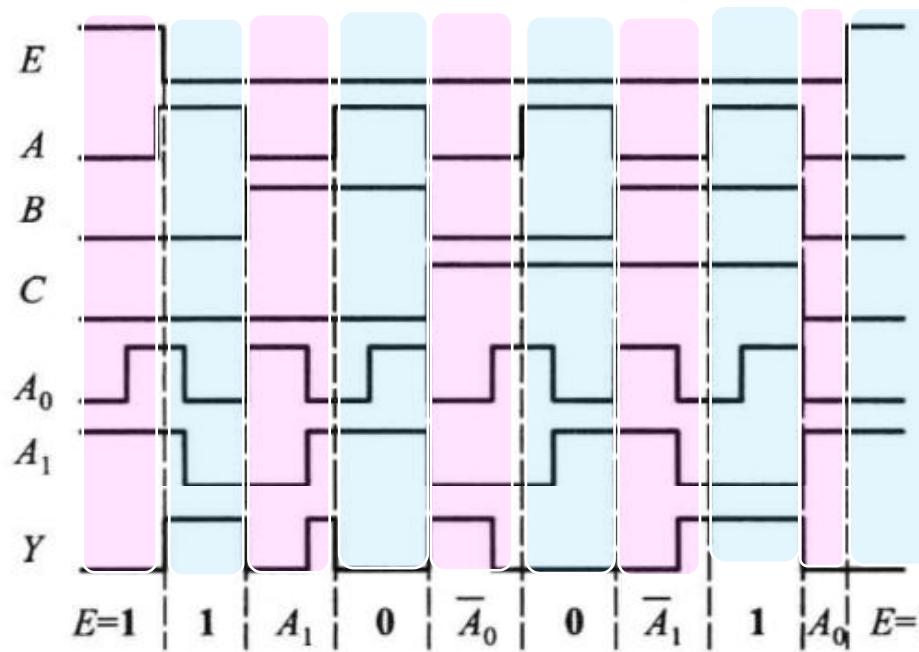
↑
最小项表达式

4.4.26 74HC151 的连接方式和各输入端的输入波形

如图题 4.4.26 所示,画出输出端 Y 的波形。



选通 \bar{S}	地址			数据 D	输出	
	A_1	A_1	A_0		Y	\bar{Y}
1	X	X	X	X	0	1
0	0	0	0	$D_0 \sim D_7$	D_0	\bar{D}_0
0	0	0	1	$D_0 \sim D_7$	D_1	\bar{D}_1
0	0	1	0	$D_0 \sim D_7$	D_2	\bar{D}_2
0	0	1	1	$D_0 \sim D_7$	D_3	\bar{D}_3
0	1	0	0	$D_0 \sim D_7$	D_4	\bar{D}_4
0	1	0	1	$D_0 \sim D_7$	D_5	\bar{D}_5
0	1	1	0	$D_0 \sim D_7$	D_6	\bar{D}_6
0	1	1	1	$D_0 \sim D_7$	D_7	\bar{D}_7



$$Y = D_0 m_0 + D_1 m_1 + D_2 m_2 + D_3 m_3 \dots$$

↑
最小项表达式

4.4.27 应用 74HC151 实现如下逻辑函数

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

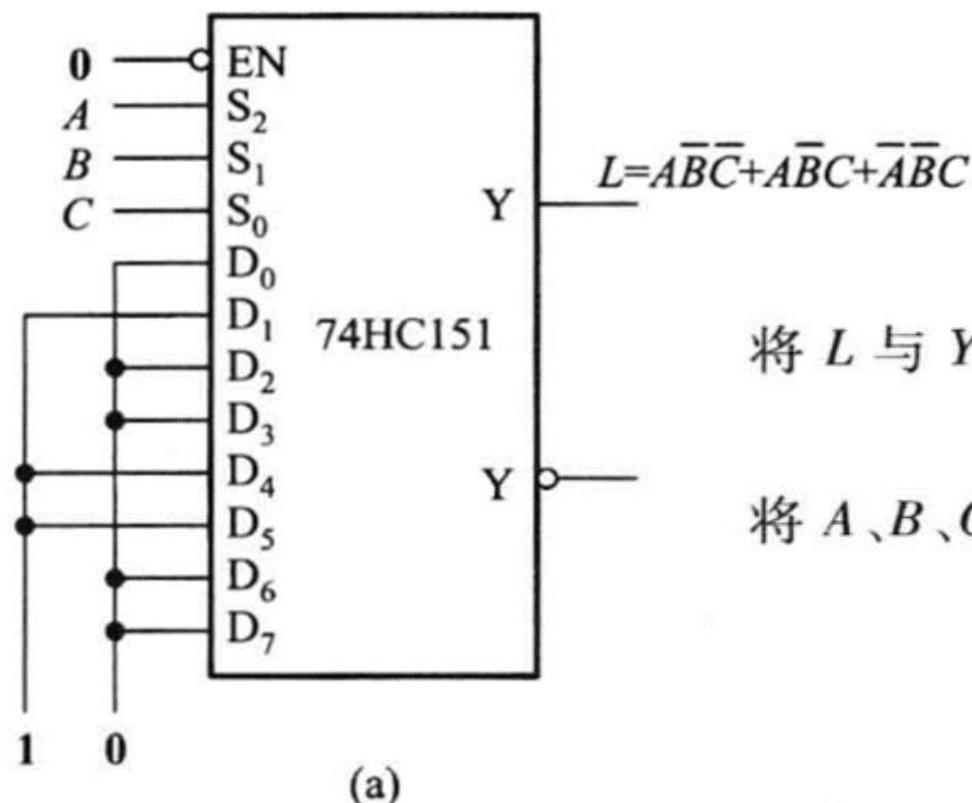
$$(2) L = (A \odot B) \odot C$$

(1) 将逻辑函数 $L = A \bar{B} \bar{C} + A \bar{B} C + \bar{A} \bar{B} C$ 写成如下形式

$$L = m_4 + m_5 + m_1$$

数据选择器集成电路芯片 74HC151 的标准表达式

$$\begin{aligned} Y &= \bar{S}_2 \bar{S}_1 \bar{S}_0 D_0 + \bar{S}_2 \bar{S}_1 S_0 D_1 + \bar{S}_2 S_1 \bar{S}_0 D_2 + \bar{S}_2 S_1 S_0 D_3 + S_2 \bar{S}_1 \bar{S}_0 D_4 \\ &\quad + S_2 \bar{S}_1 S_0 D_5 + S_2 S_1 \bar{S}_0 D_6 + S_2 S_1 S_0 D_7 \\ &= m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 + m_4 D_4 + m_5 D_5 + m_6 D_6 + m_7 D_7 \end{aligned}$$



将 L 与 Y 比较可得

$$D_0 = D_2 = D_3 = D_6 = D_7 = 0 \quad D_1 = D_4 = D_5 = 1$$

将 A 、 B 、 C 分别与地址输入端 S_2 、 S_1 、 S_0 连接, 即可得到电路, 如图

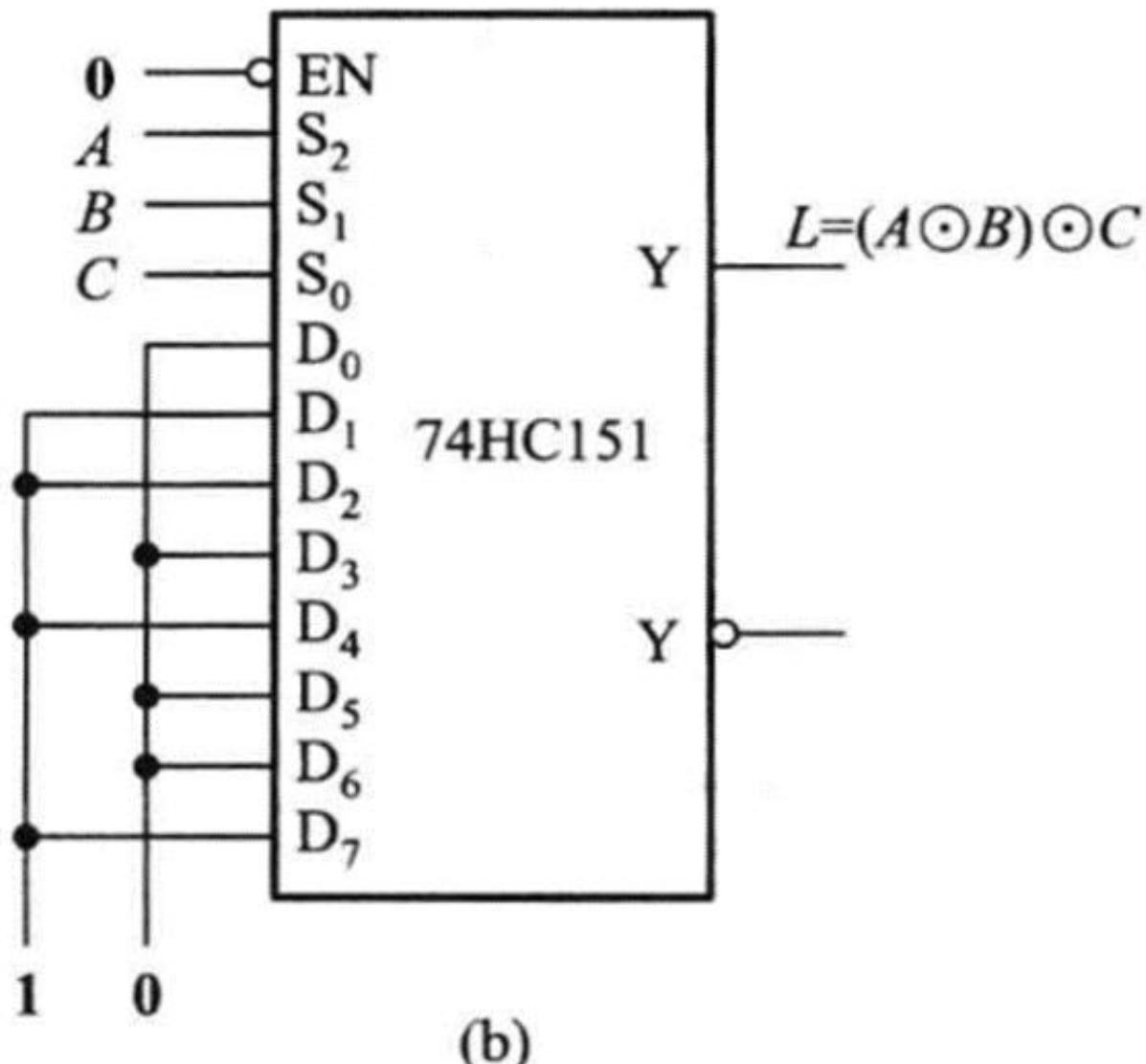
(2) 将逻辑函数表达式展开成最小项形式

$$\begin{aligned}Y &= A \odot B \odot C = (\bar{A}\bar{B} + AB) \odot C = \overline{\overline{\bar{A}\bar{B}} + ABC} + (\bar{A}\bar{B} + AB)C = (\bar{A}B + A\bar{B})\bar{C} + \bar{A}\bar{B}C + ABC \\&= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC = m_1 + m_2 + m_4 + m_7\end{aligned}$$

可得

$$D_0 = D_3 = D_5 = D_6 = 0 \quad D_1 = D_2 = D_4 = D_7 = 1$$

同理, 将 A 、 B 、 C 分别与地址输入端 S_2 、 S_1 、 S_0 连接, 即可得到电路, 如图题解 4.4.27(b) 所示。

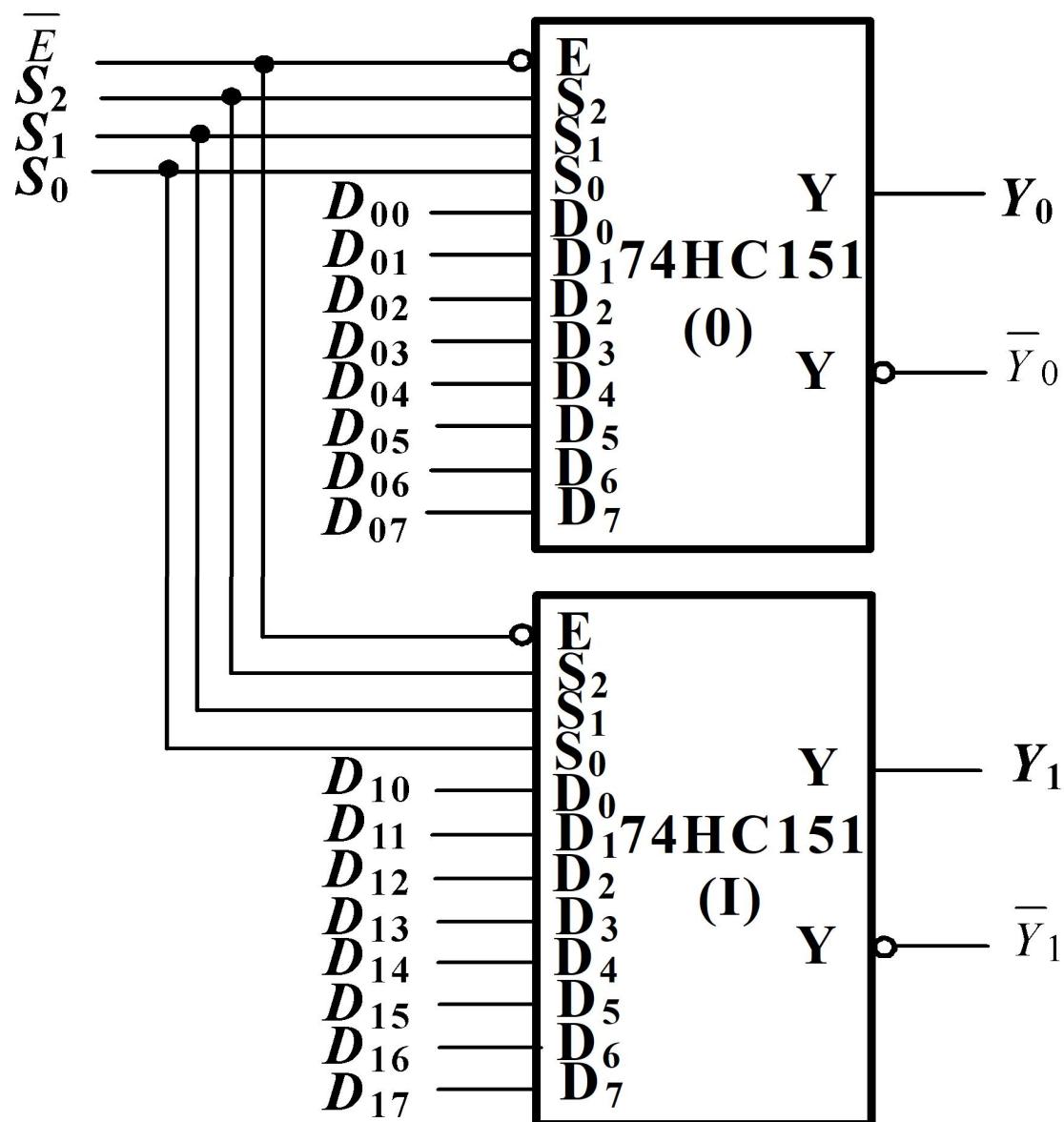


4. 数据选择器的扩展应用

(1) 位的扩展

用两片74x 151组成**二位**八选一的数据选择器

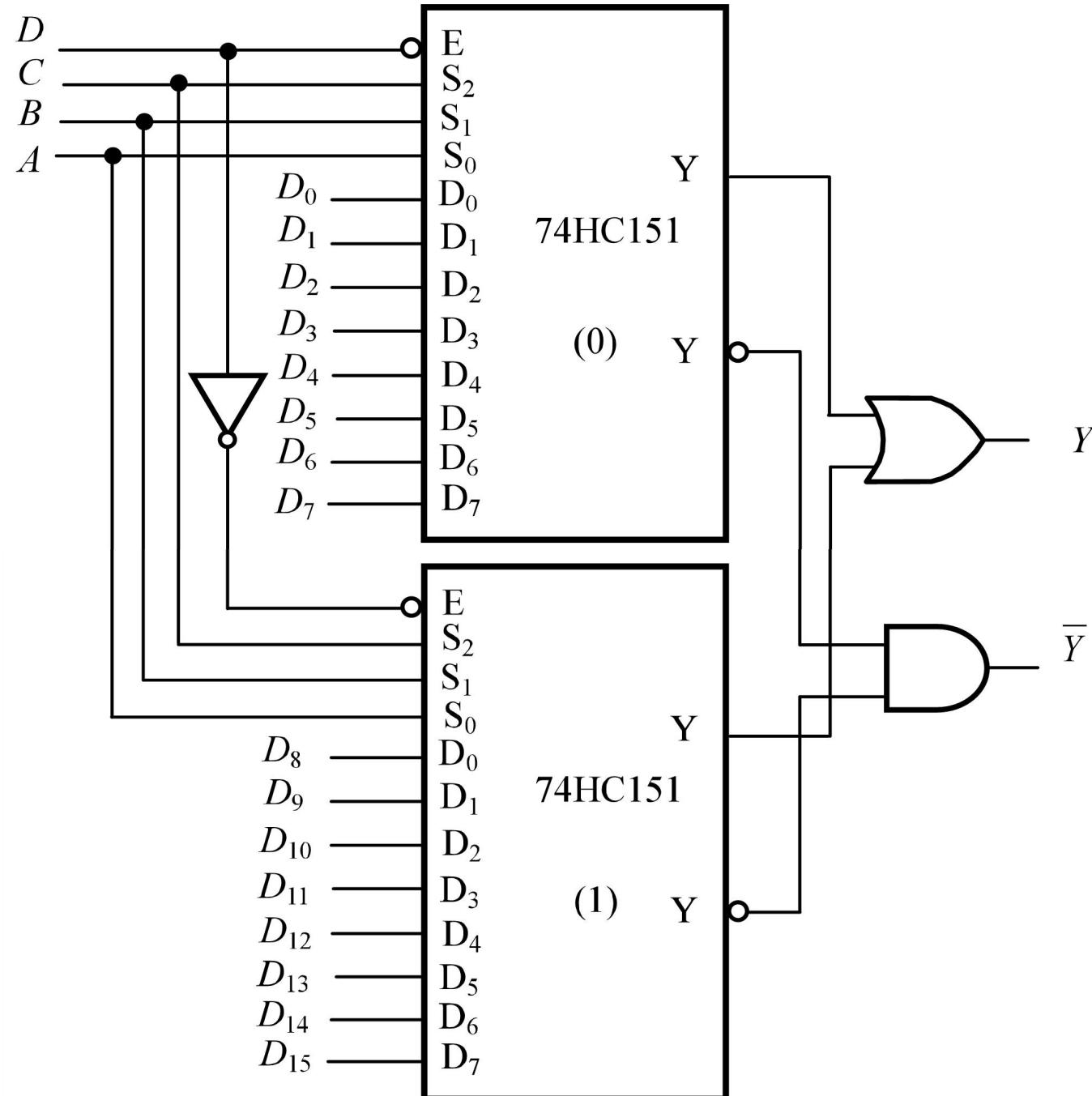
地址共用
使能控制端：决定片选



(2) 字的扩展

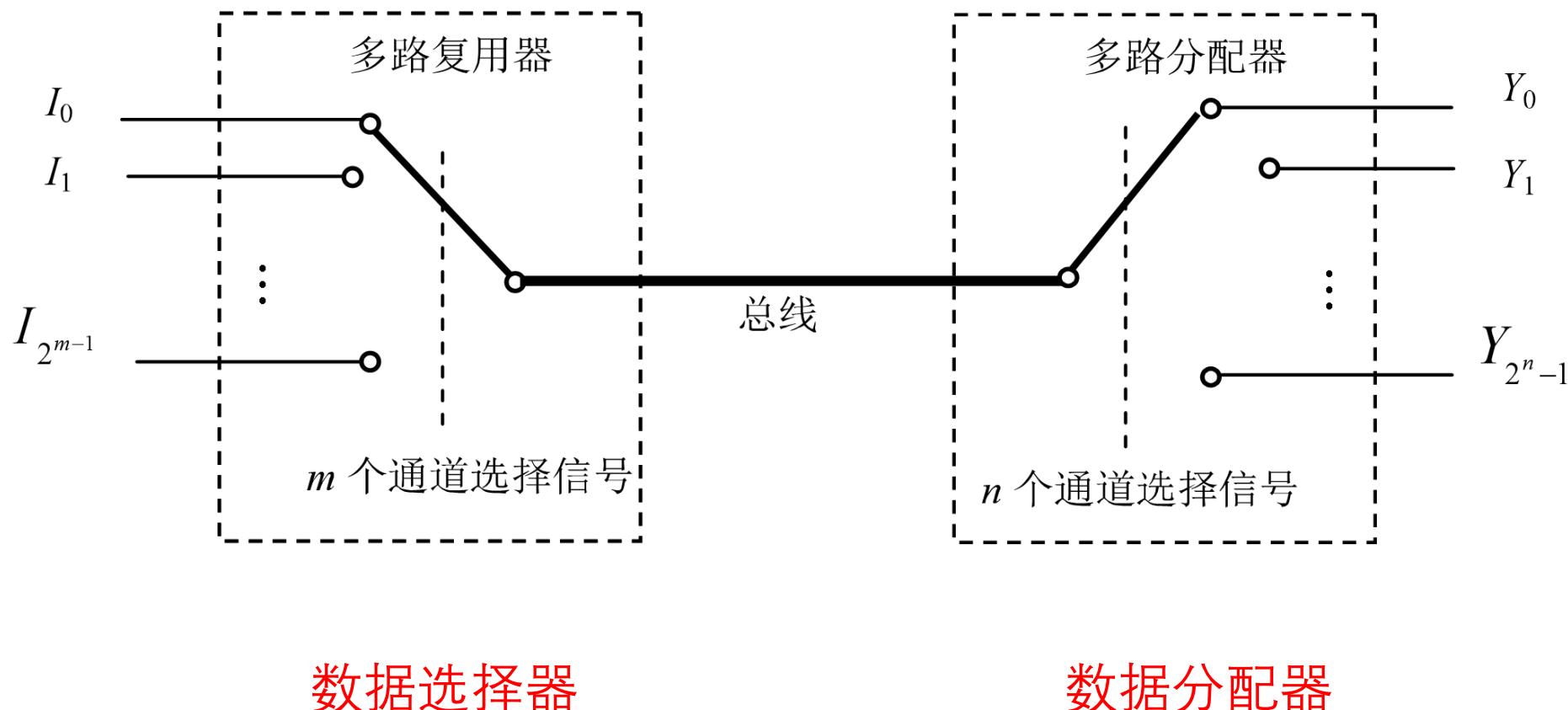
将两片74X151连接成一个16选1的数据选择器，

低位：输入地址共用
高位：控制端反向共接

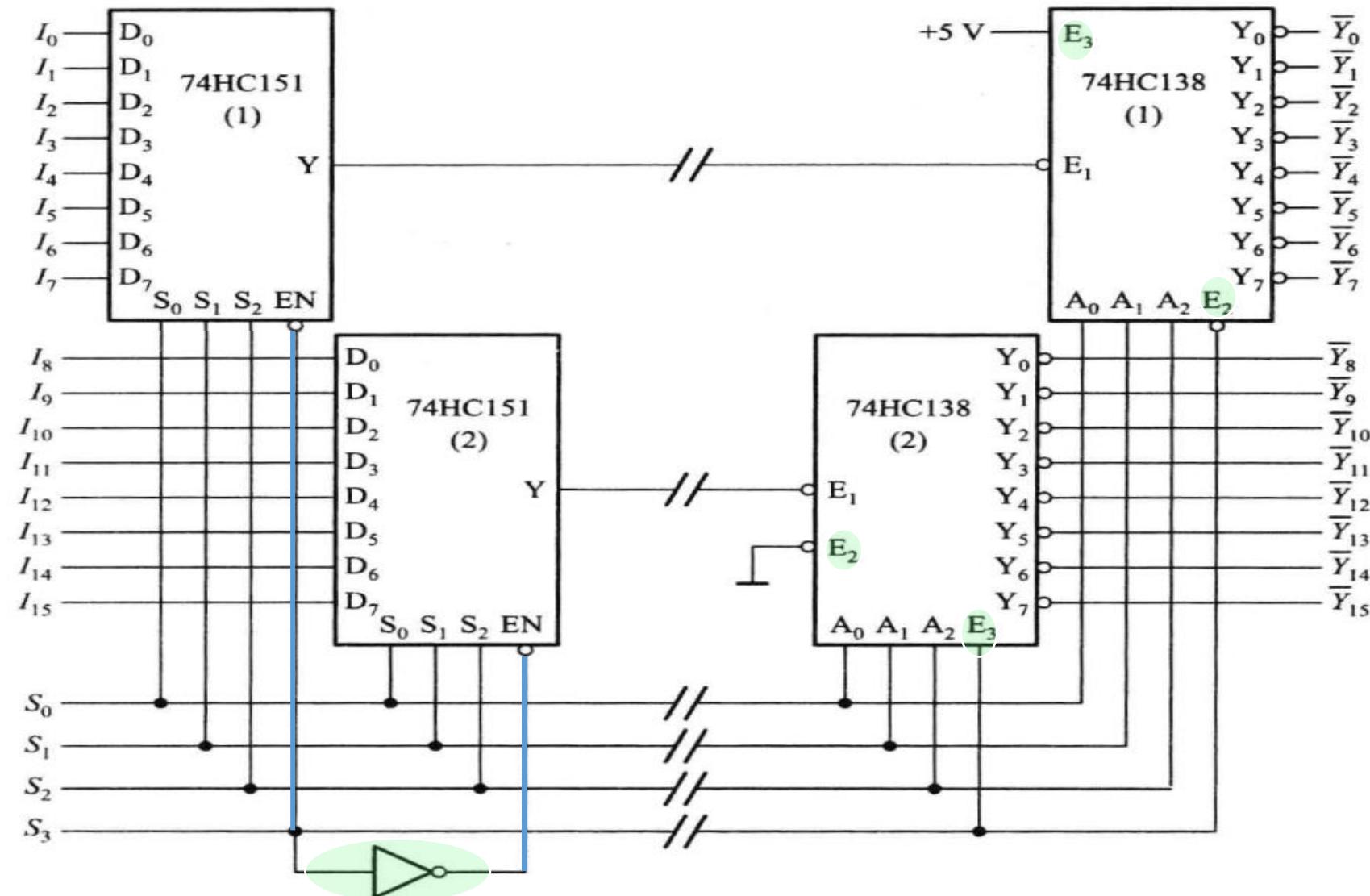


5. 数据选择器、数据分配器与总线的连接

这种信息传输的基本原理在通信系统、计算机网络系统、以及计算机内部各功能部件之间的信息转送等等都有广泛的应用。



4.4.28 应用数据选择器 74HC151 和 3 线-8 线译码器 74HC138 设计一个数据传输电路，其功能是在 4 位通道选择信号的控制下，能将 16 个输入数据中的任何一个传送到 16 个输出端中相对应的一个输出端，其示意图如图题 4.4.28 所示。



4.4.4 数值比较器

数值比较器：对两个1位数字进行比较（A、B），
以判断其大小的逻辑电路。

两个数比较，一个结果，三种可能（ $A>B, A<B, A=B$ ）

1. 1位二进制数比较器

输入：两个一位二进制数 A, B 。

输出： $F_{A>B}=1$ ，表示 A 大于 B

$F_{A<B}=1$ ，表示 A 小于 B

$F_{A=B}=1$ ，表示 A 等于 B

1位二进制数值比较器

(1) 1位数值比较器真值表

输入		输出		
A	B	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

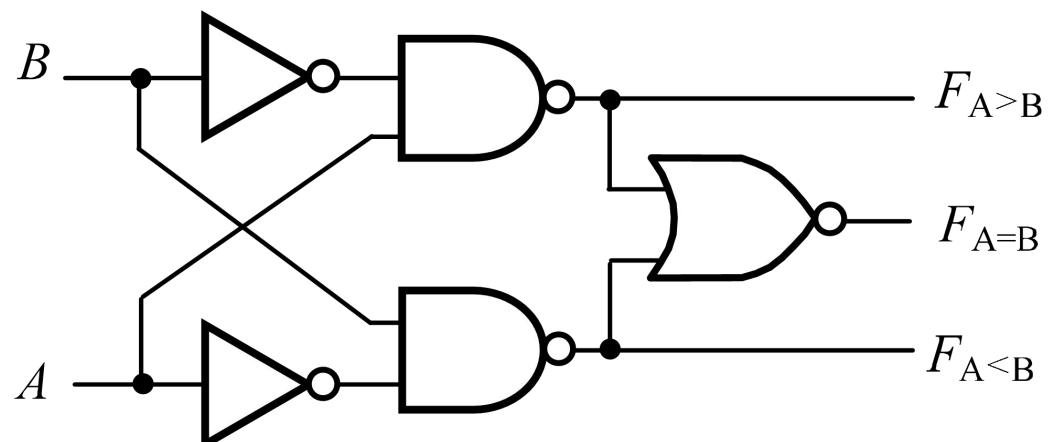
(2) 逻辑函数表达式

$$F_{A>B} = A \bar{B}$$

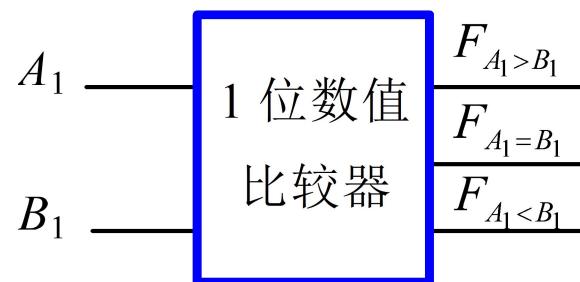
$$F_{A<B} = \bar{A} B$$

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

(3) 逻辑电路图



(4) 逻辑符号



2. 2位二进制数值比较器： 比较两个2位二进制数的大小的电路

输入：两个2位二进制数 $A=A_1A_0$ 、 $B=B_1B_0$

- (1) 当高位 (A_1 、 B_1) 不相等时，无需比较低位 (A_0 、 B_0)，高位比较的结果就是两个数的比较结果。
- (2) 当高位相等时，两数的比较结果由低位比较的结果决定。

2位数值比较器真值表

输入				输出		
A_1	B_1	A_0	B_0	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_1 > B_1$		\times		1	0	0
$A_1 < B_1$		\times		0	1	0
$A_1 = B_1$	$A_0 > B_0$			1	0	0
$A_1 = B_1$	$A_0 < B_0$			0	1	0
$A_1 = B_1$	$A_0 = B_0$			0	0	1

$$F_{A>B} = A_1 \bar{B}_1 + (\bar{A}_1 \bar{B}_1 + A_1 B_1) A_0 \bar{B}_0$$

$$= F_{A_1 > B_1} + F_{A_1 = B_1} \cdot F_{A_0 > B_0}$$

$$F_{A<B} = \bar{A}_1 B_1 + (\bar{A}_1 \bar{B}_1 + A_1 B_1) \bar{A}_0 B_0$$

$$= F_{A_1 < B_1} + F_{A_1 = B_1} \cdot F_{A_0 < B_0}$$

$$F_{A=B} = F_{A_1 = B_1} \cdot F_{A_0 = B_0}$$

能否用1位数值比较器设计两位数值比较器?

总结: 用一位数值比较器设计多位数值比较器的原则

(1) 2位数值比较器真值表

输入		输出				
A_1	B_1	A_0	B_0	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_1 > B_1$	\times			1	0	0
$A_1 < B_1$	\times			0	1	0
$A_1 = B_1$	$A_0 > B_0$			1	0	0
$A_1 = B_1$	$A_0 < B_0$			0	1	0
$A_1 = B_1$	$A_0 = B_0$			0	0	1

(2) 逻辑函数表达式:

$$F_{A>B} = \bar{A}_1 \bar{B}_1 + (\bar{A}_1 \bar{B}_1 + A_1 \bar{B}_1) A_0 \bar{B}_0$$

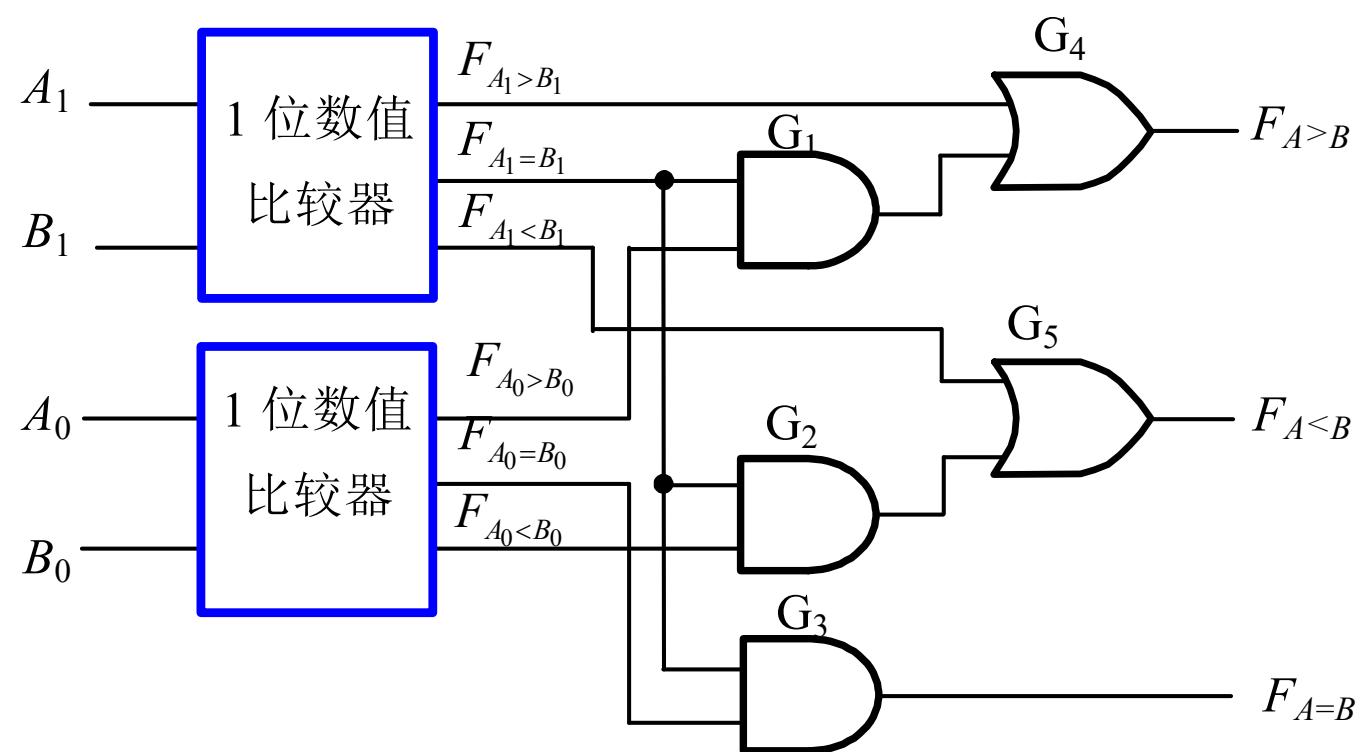
$$= F_{A_1 > B_1} + F_{A_1 = B_1} \cdot F_{A_0 > B_0}$$

$$F_{A<B} = \bar{A}_1 B_1 + (\bar{A}_1 B_1 + A_1 B_1) \bar{A}_0 B_0$$

$$= F_{A_1 < B_1} + F_{A_1 = B_1} \cdot F_{A_0 < B_0}$$

$$F_{A=B} = F_{A_1 = B_1} \cdot F_{A_0 = B_0}$$

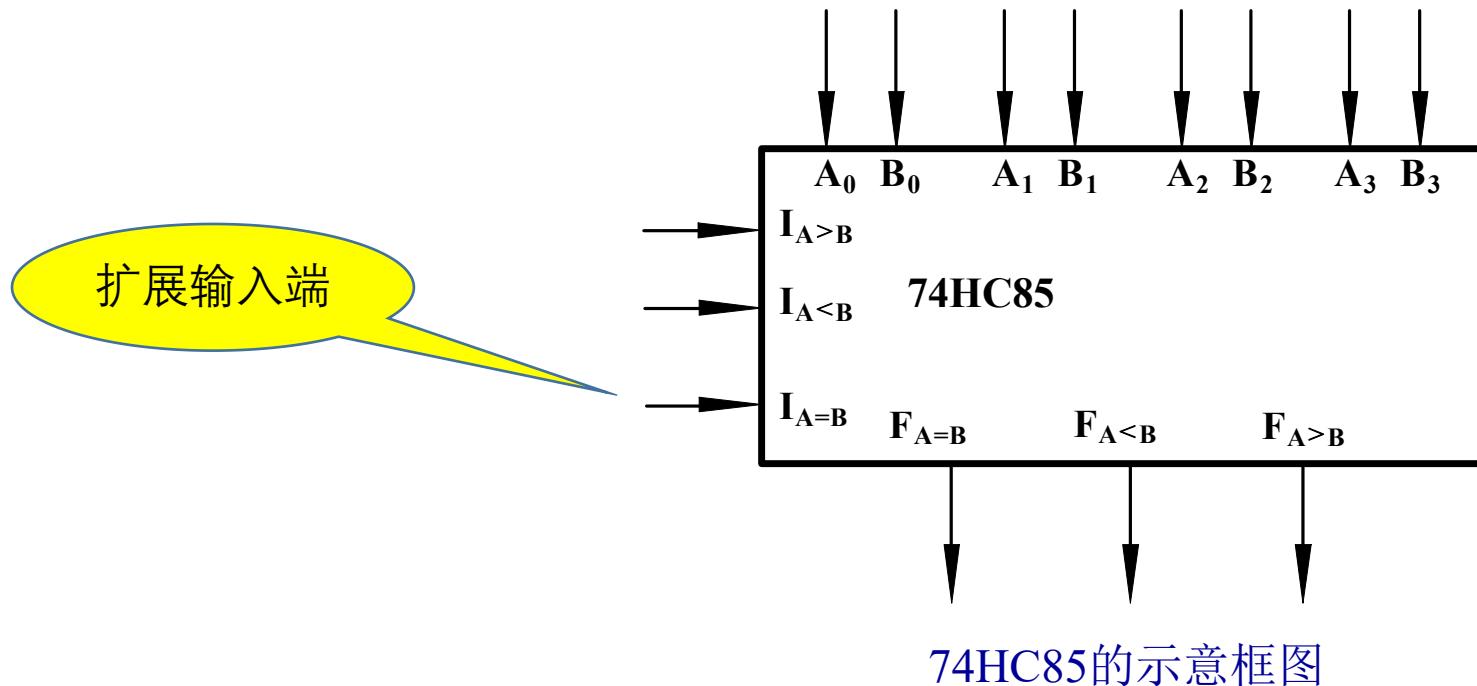
(3) 逻辑电路图:



3. 集成数值比较器

(1) 集成数值比较器74HC85的功能

74HC85是四位数值比较器，其工作原理和两位数值比较器相同。



学会阅读功能表：

1. 读什么-----逻辑功能+控制方式或扩展方式
2. 怎么读-----根据输入控制端或扩展端将功能表分块
-----从输入端找原因，从输出端找结果。

74HC85数值比较器功能表

输入							输出						
A_3	B_3	A_2	B_2	A_1	B_1	A_0	B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_3 > B_3$		\times		\times		\times		\times	\times	\times	1	0	0
$A_3 < B_3$		\times		\times		\times		\times	\times	\times	0	1	0
$A_3 = B_3$		$A_2 > B_2$		\times		\times		\times	\times	\times	1	0	0
$A_3 = B_3$		$A_2 < B_2$		\times		\times		\times	\times	\times	0	1	0
$A_3 = B_3$		$A_2 = B_2$		$A_1 > B_1$		\times		\times	\times	\times	1	0	0
$A_3 = B_3$		$A_2 = B_2$		$A_1 < B_1$		\times		\times	\times	\times	0	1	0
$A_3 = B_3$		$A_2 = B_2$		$A_1 = B_1$		$A_0 > B_0$		\times	\times	\times	1	0	0
$A_3 = B_3$		$A_2 = B_2$		$A_1 = B_1$		$A_0 < B_0$		\times	\times	\times	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
$A_3 = B_3$		$A_2 = B_2$		$A_1 = B_1$		$A_0 = B_0$		\times	\times	1	0	0	1

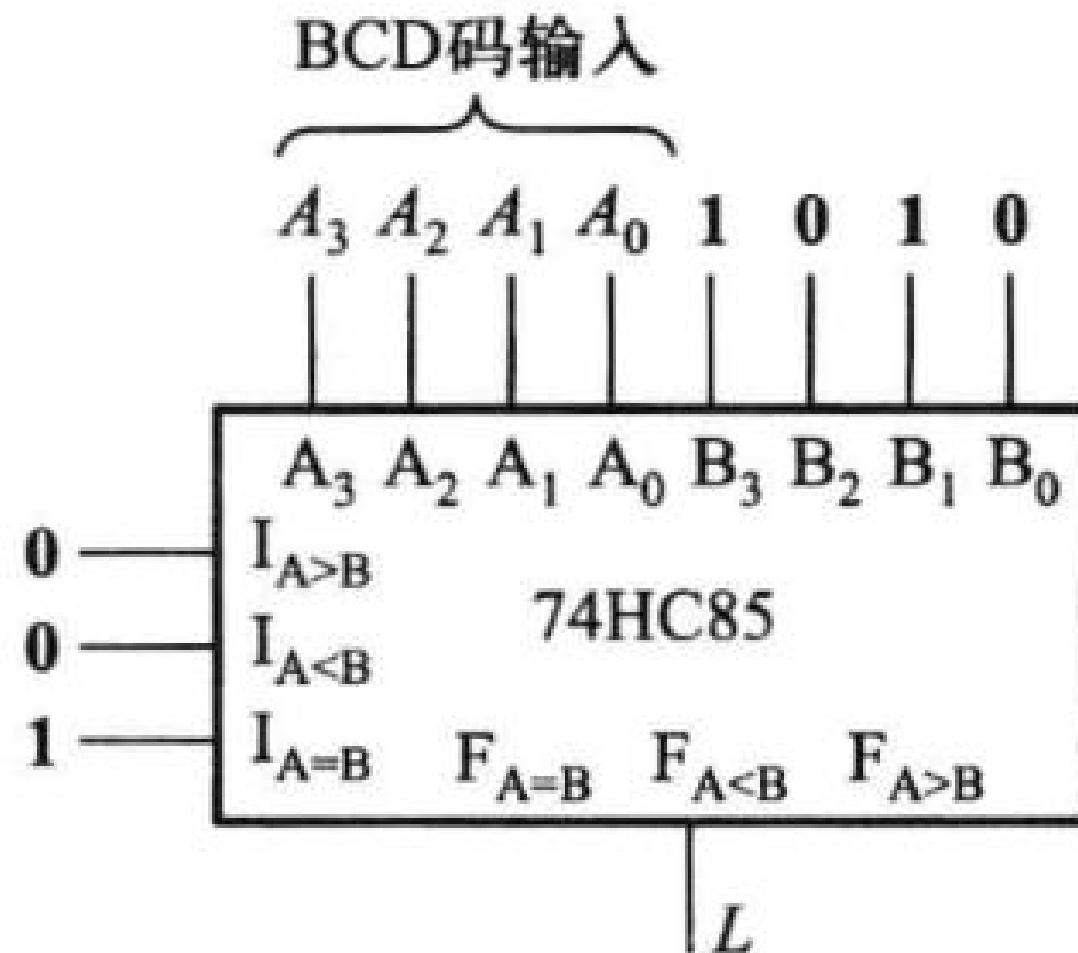
74HC85数值比较器表达式

$$\begin{aligned}
 F_{A>B} = & F_{A_3 > B_3} + F_{A_3 = B_3} F_{A_2 > B_2} + F_{A_3 = B_3} F_{A_2 = B_2} F_{A_1 > B_1} + F_{A_3 = B_3} F_{A_2 = B_2} F_{A_1 = B_1} F_{A_0 > B_0} \\
 & + F_{A_3 = B_3} F_{A_2 = B_2} F_{A_1 = B_1} F_{A_0 = B_0} I_{A>B}
 \end{aligned}$$

$$\begin{aligned}
 F_{A<B} = & F_{A_3 < B_3} + F_{A_3 = B_3} F_{A_2 < B_2} + F_{A_3 = B_3} F_{A_2 = B_2} F_{A_1 < B_1} + F_{A_3 = B_3} F_{A_2 = B_2} F_{A_1 = B_1} F_{A_0 < B_0} \\
 & + F_{A_3 = B_3} F_{A_2 = B_2} F_{A_1 = B_1} F_{A_0 = B_0} I_{A<B}
 \end{aligned}$$

$$F_{A=B} = F_{A_3 = B_3} F_{A_2 = B_2} F_{A_1 = B_1} F_{A_0 = B_0} I_{A=B}$$

4.4.32 用数值比较器74HC85设计一个8421BCD码有效性测试电路。
即当输入8421BCD码时，输出为1，否则为0.



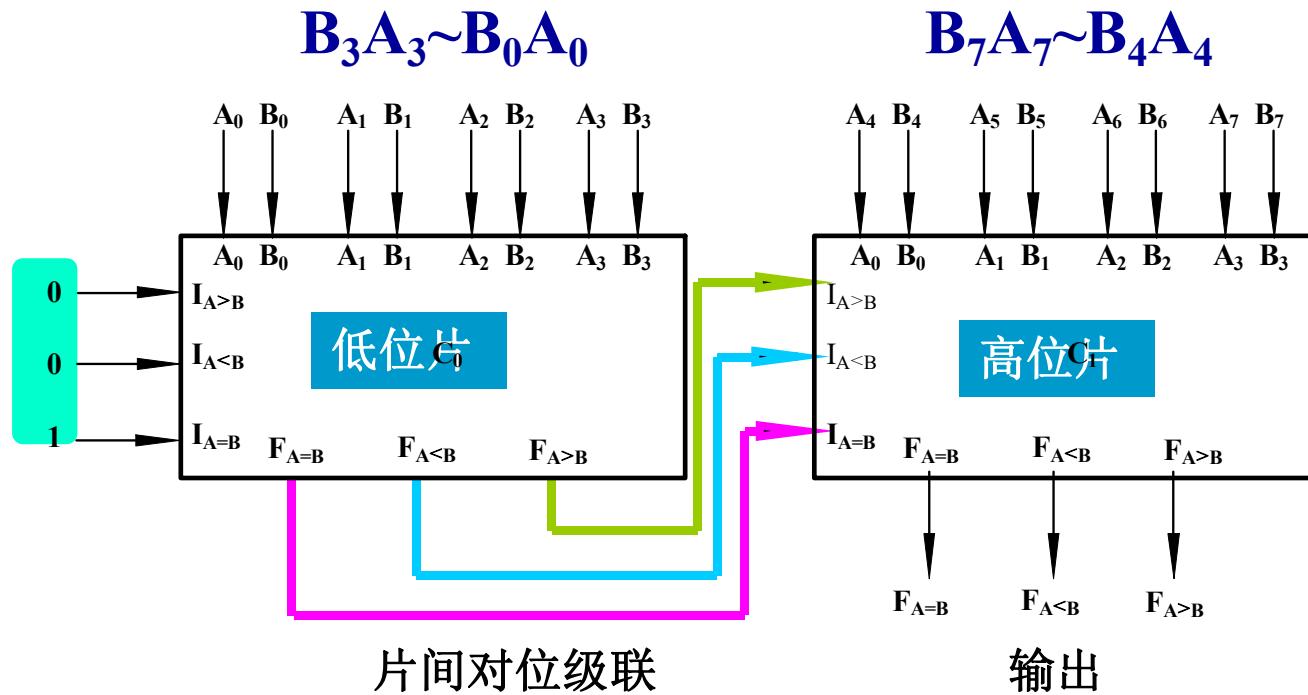
图题解 4.4.32

4. 集成数值比较器的位数扩展

用两片74HC85组成8位数值比较器（串联扩展方式）。

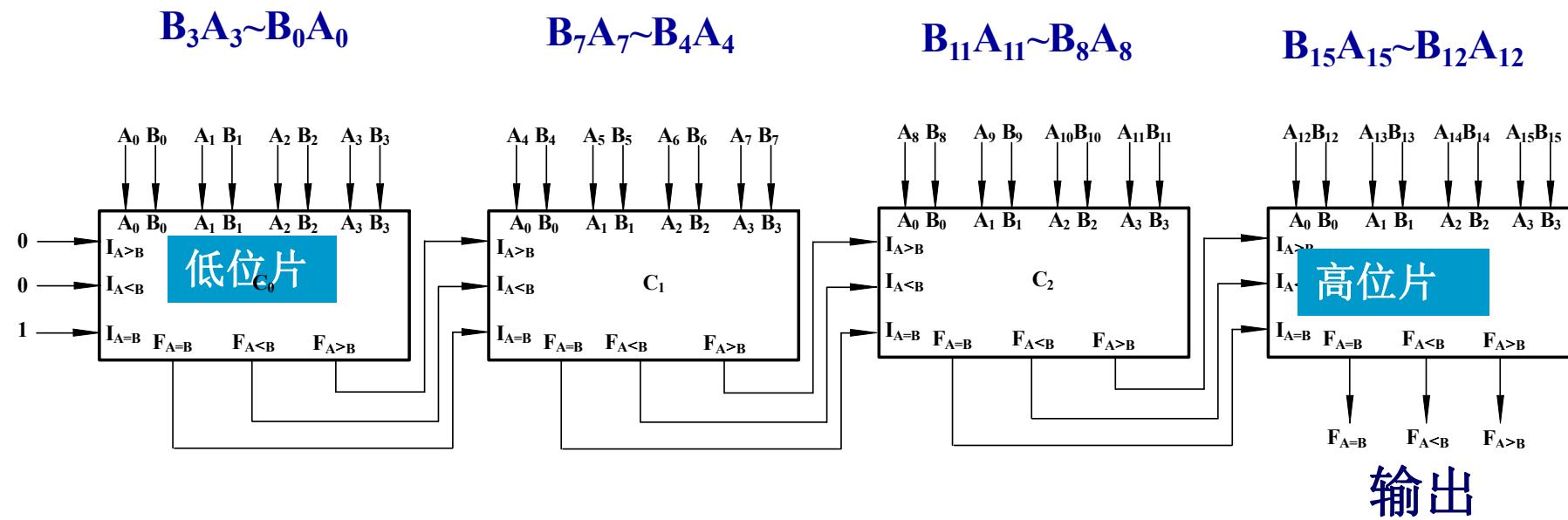
输入: $A = A_7A_6A_5A_4A_3A_2A_1A_0$ $B = B_7B_6B_5B_4B_3B_2B_1B_0$

输出: $F_{A>B}$ $F_{A<B}$ $F_{A=B}$



采用串联扩展方式数值比较器

用四片74HC85组成16位数值比较器（串联扩展方式）。

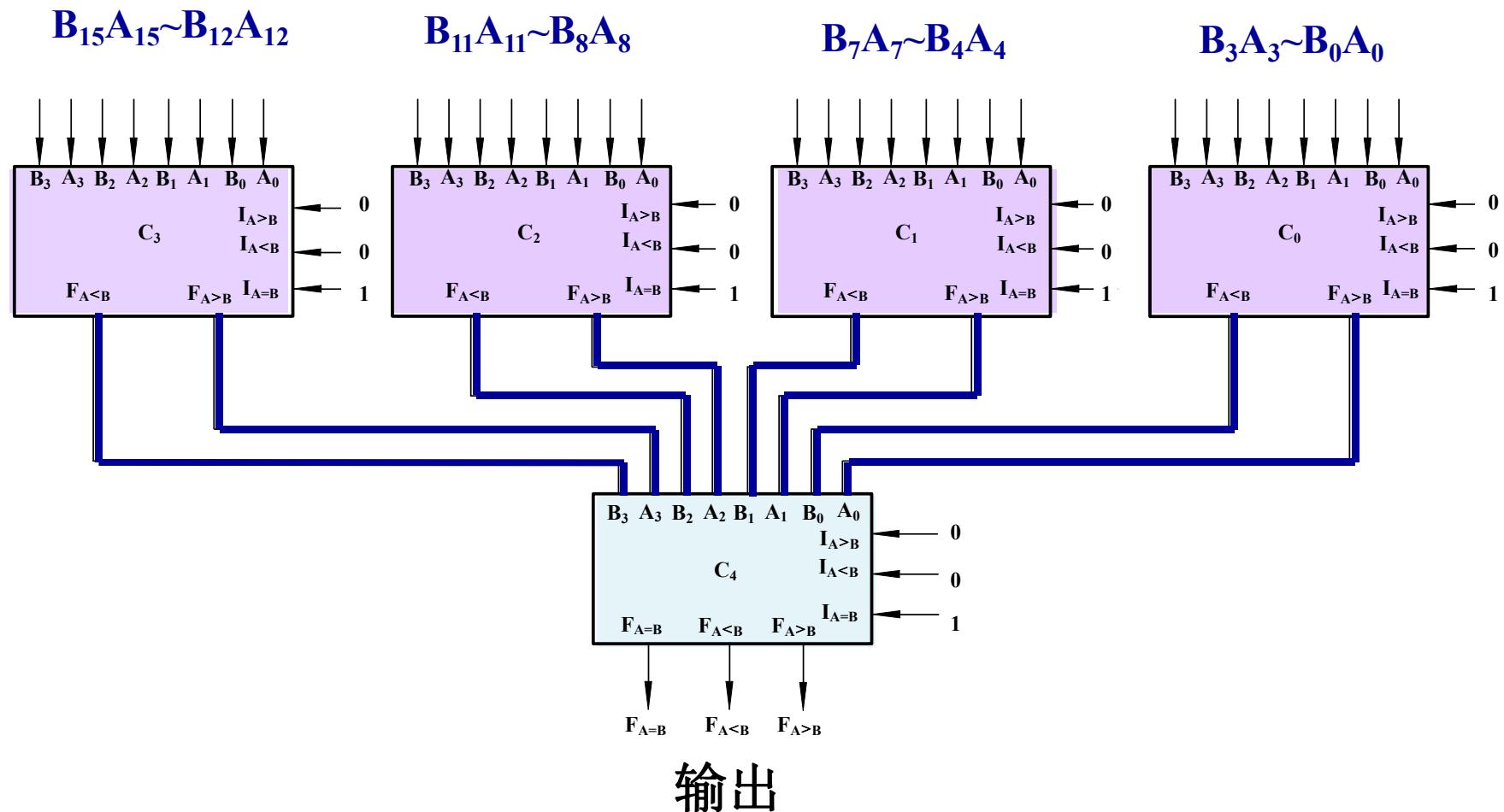


问题：如果每一片延迟时间为10ns， 16位串行比较器延迟时间？

40ns

采用并联扩展方式数值比较器

用五片74HC85组成16位数值比较器的并联扩展方式。



问题：如果每一片延迟时间为10ns， 16位并行比较器延迟时间？

20ns

4.4.5 算术运算电路

1. 半加器和全加器

加法器：半加器和全加器两种。

半加器：两个1位二进制数相加时，不考虑低位来的进位的加法

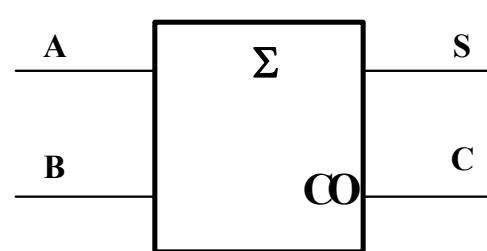
全加器：在两个1位二进制数相加时，考虑低位进位的加法

(1) 1位半加器 (Half Adder)

• 半加器的真值表

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

• 逻辑符号

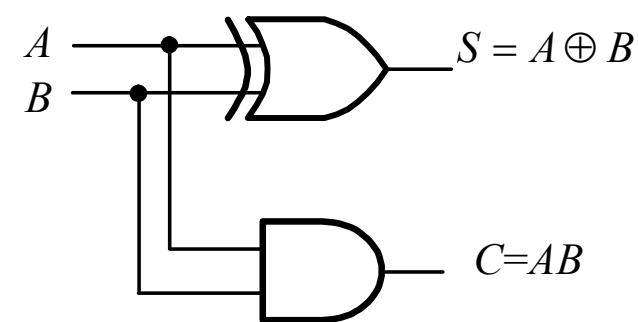


• 逻辑表达式

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

$$C = AB$$

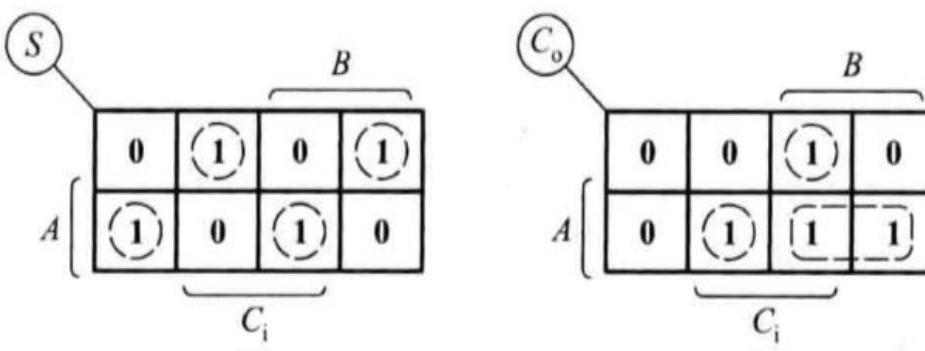
• 逻辑图



(2) 全加器 (Full Adder)

- 全加器真值表

A	B	C _i	S	C _o
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



- 逻辑表达式

$$S = \overline{A}\overline{B}C_i + \overline{A}B\overline{C}_i + A\overline{B}\overline{C}_i + ABC_i$$

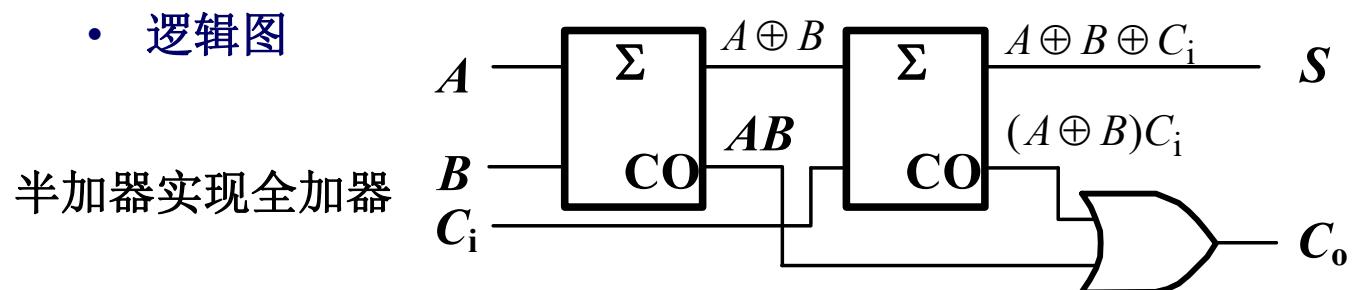
$$= \underline{A} \oplus B \oplus C_i$$

$$C_o = AB + A\overline{B}C_i + \overline{A}BC_i$$

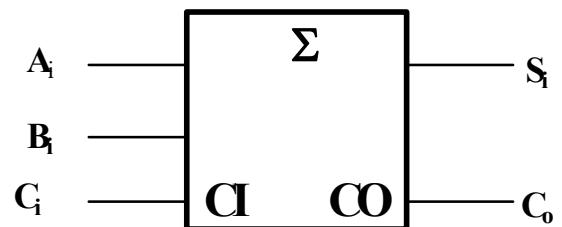
$$= \underline{AB} + (\underline{A} \oplus \underline{B})C_i$$

最小项之和

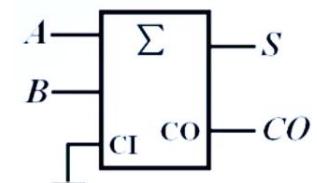
- 逻辑图



- 逻辑符号



• 全加器实现半加器



加法器的应用----三位二进制代码奇偶校验器

全加器真值表

A	B	C	S	C
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

ABC 有奇数个1时， S 为1；

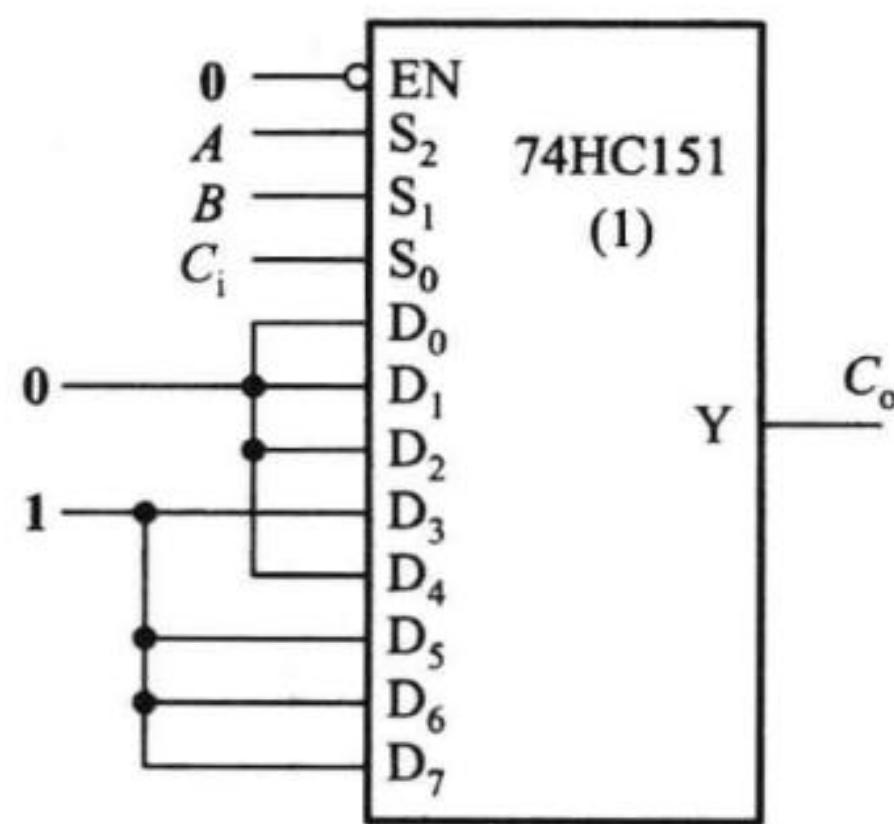
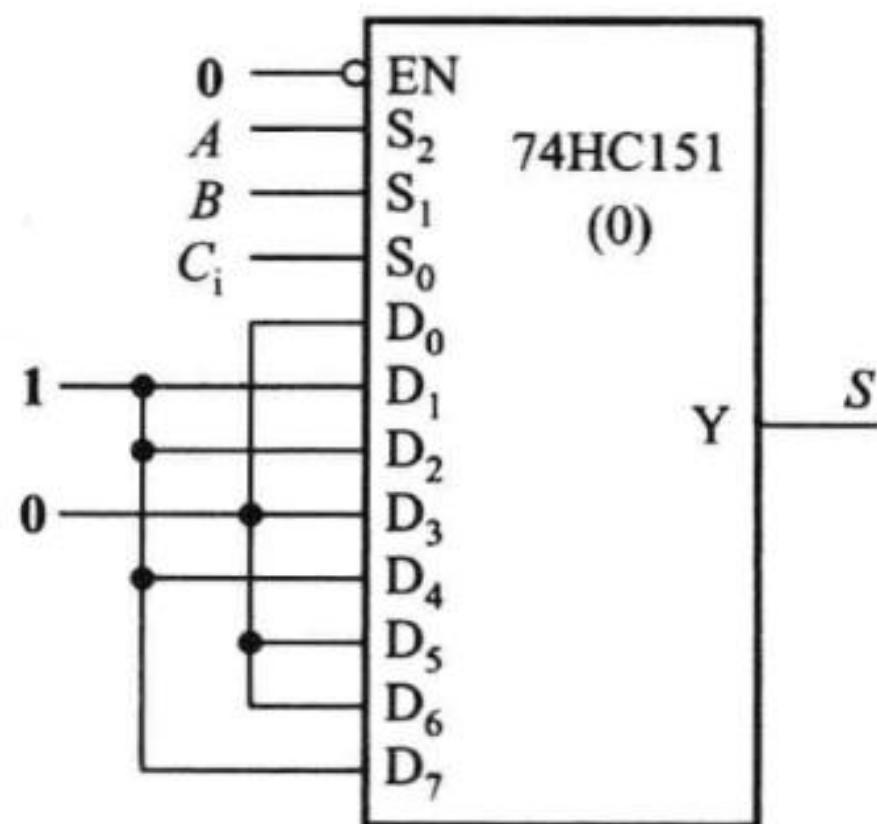
ABC 有偶数个1或全为0时， S 为0。

思考：用全加器组成八位二进制代码奇偶校验器， 电路应如何连接？

4.4.35 试用 8 选 1 数据选择器 74HC151, 实现 1 位二进制全加器。

$$S = \overline{A} \overline{B} C_i + \overline{A} B \overline{C}_i + A \overline{B} \overline{C}_i + ABC_i = m_1 + m_2 + m_4 + m_7$$

$$C_o = \overline{A} BC_i + A \overline{B} C_i + AB \overline{C}_i + ABC_i = m_3 + m_5 + m_6 + m_7$$



全加器, 片(0)实现 S 表达式, 其中

$$D_1 = D_2 = D_4 = D_7 = 1$$

$$D_0 = D_3 = D_5 = D_6 = 0$$

片(1)实现 C_o 表达式, 其中

$$D_3 = D_5 = D_6 = D_7 = 1$$

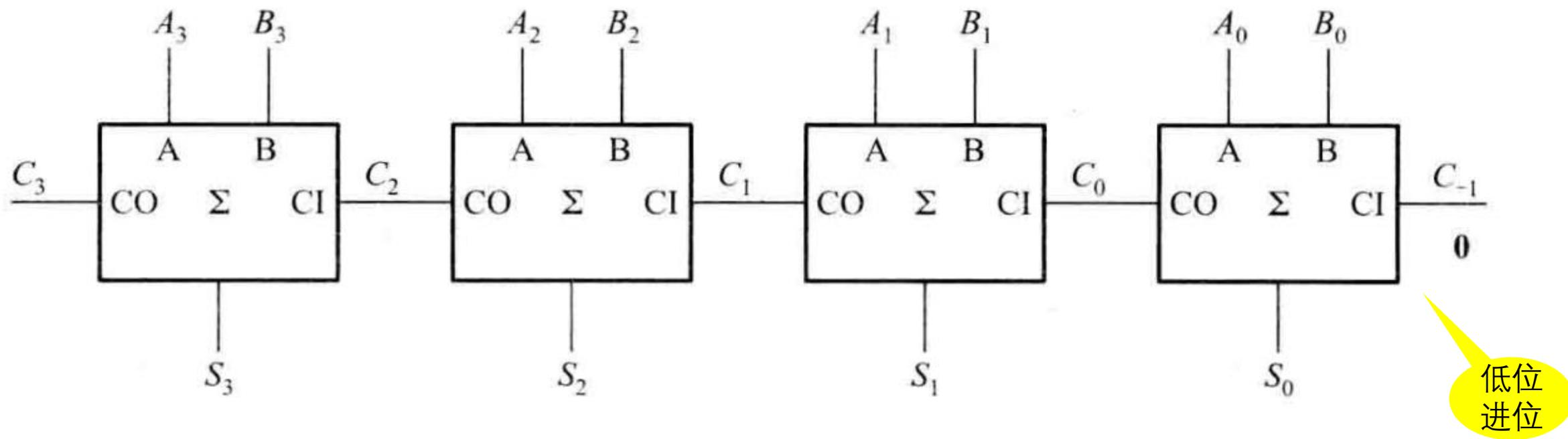
$$D_0 = D_1 = D_2 = D_4 = 0$$

2. 多位数加法器

- 如何用1位全加器实现两个四位二进制数相加？

$$\bullet A_3 \ A_2 \ A_1 \ A_0 + B_3 \ B_2 \ B_1 \ B_0 = ?$$

(1) 串行进位加法器



低位的进位信号送给邻近高位作为输入信号，采用串行进位加法器运算速度不高。

串行进位：任何一位的加法运算必须在相邻你们的运算完成之后才能进行

缺点：运算速度慢

(2) 超前进位加法器

提高运算速度的基本思想：

设计进位信号产生电路，在输入每位的加数和被加数时，同时获得该位全加的进位信号，而无需等待最低位的进位信号。

$$\begin{array}{ccccccc} & \text{3位} & \text{2位} & \text{1位} & \text{0位} & & \\ & A_3 & A_2 & A_1 & A_0 & & \\ + & B_3 & B_2 & B_1 & B_0 & & \\ \hline C_3 & S_3 & S_2 & S_1 & S_0 & & \end{array}$$

$$S_i = \underline{A_i \oplus B_i} \oplus C_{i-1}$$

$$C_i = \underline{A_i B_i} + \underline{(A_i \oplus B_i) C_{i-1}}$$

假设： $G_i = A_i B_i$

$$P_i = A_i \oplus B_i$$

$$\text{则: } S_i = P_i \oplus C_{i-1}$$

$$C_i = G_i + P_i C_{i-1}$$

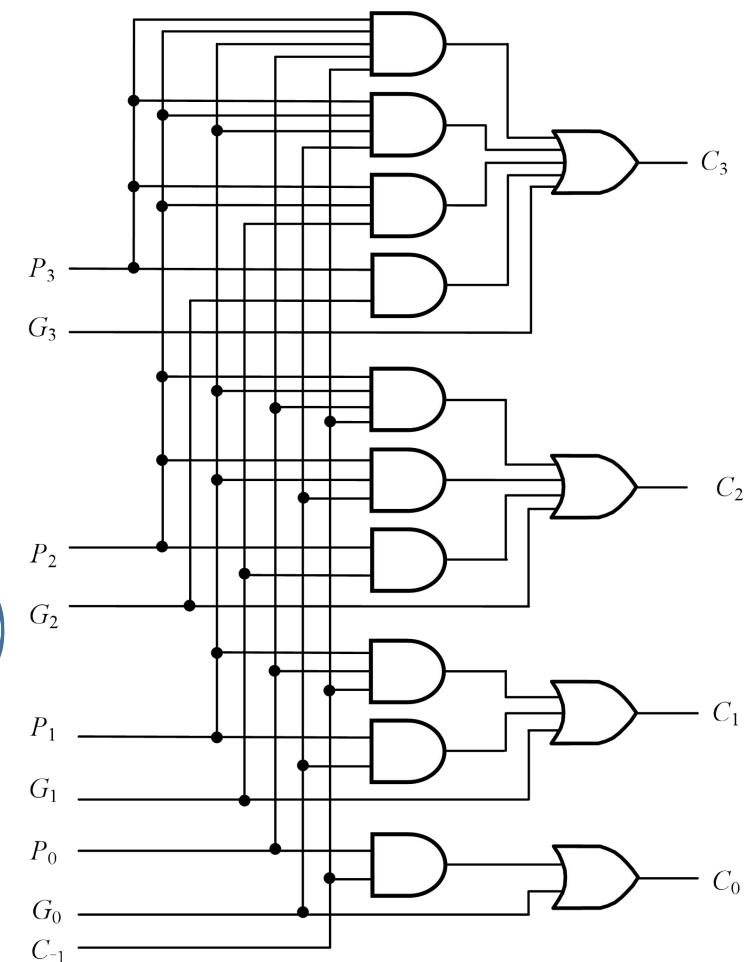
4位全加器进位信号的产生：

$$\text{假设: } G_i = A_i B_i \quad P_i = A_i \oplus B_i$$

$$\text{则: } S_i = P_i \oplus C_{i-1} \quad C_i = G_i + P_i C_{i-1}$$

超前进位产生电路

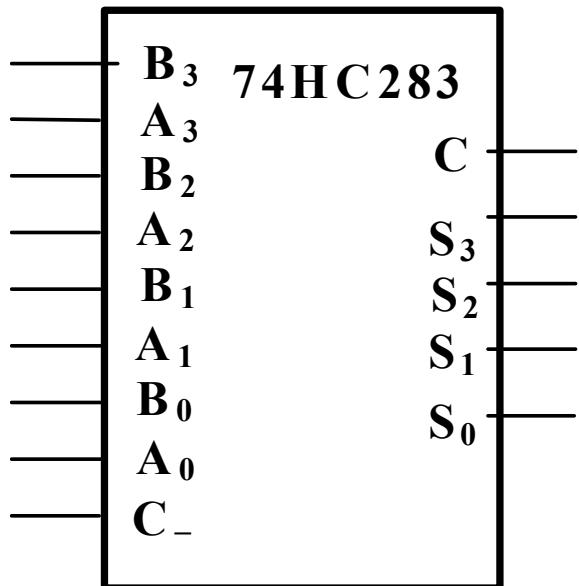
$$\begin{aligned}
 C_0 &= G_0 + P_0 C_{-1} \\
 C_1 &= G_1 + P_1 C_0 \\
 C_1 &= G_1 + P_1 G_0 + P_1 P_0 C_{-1} \\
 C_2 &= G_2 + P_2 C_1 \\
 C_2 &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1} \\
 C_3 &= G_3 + P_3 C_2 = G_3 + P_3 (G_2 + P_2 C_1) = G_3 + P_3 G_2 + P_3 P_2 C_1 \\
 &= G_3 + P_3 G_2 + P_3 P_2 (G_1 + P_1 C_0) \\
 C_3 &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 (G_0 + P_0 C_{-1})
 \end{aligned}$$



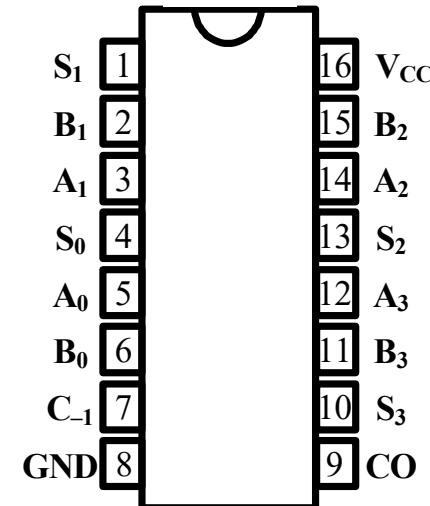
所有的进位信号只由被加数、加数和C-1决定，而与其它低位的进位无关。
提高了速度，但位数增加时，进位电路复杂度增加。

超前进位集成4位加法器74LS283

74HC283逻辑框图



74HC283引脚图



例. 用74X283构成将8421BCD码转换为余3码的码制转换电路。

8421码

0000

+0011

0001

+0011

0010

+0011

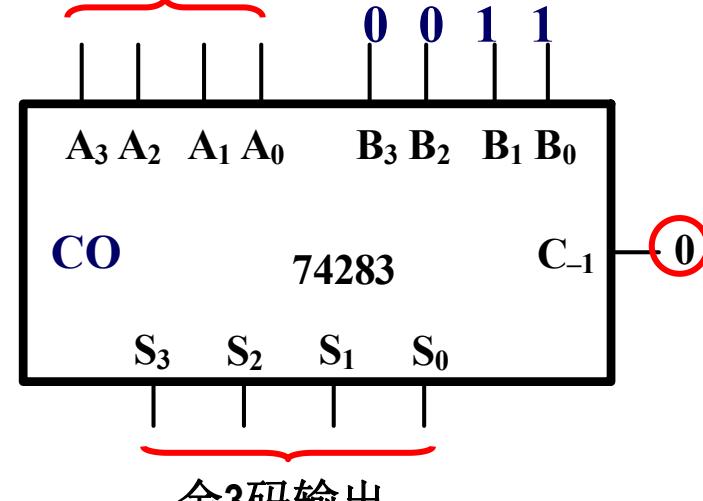
余3码

0011

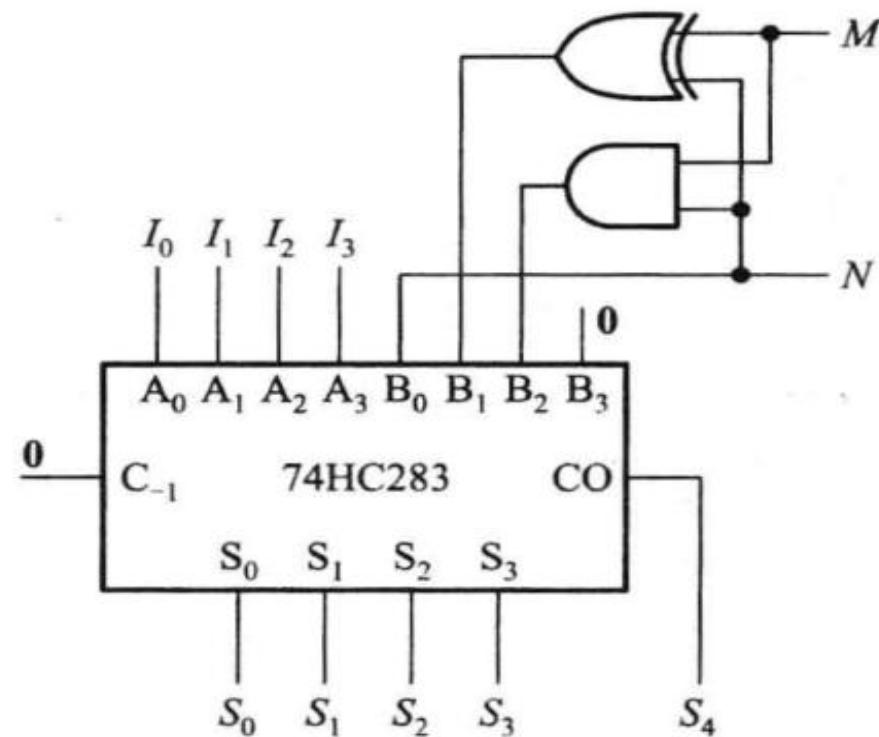
0100

0101

8421码输入



4.4.37 由4位数加法器74HC283构成的逻辑电路如图题4.4.37所示, M 和 N 为控制端,试分析该电路的功能。



表题解 4.4.37

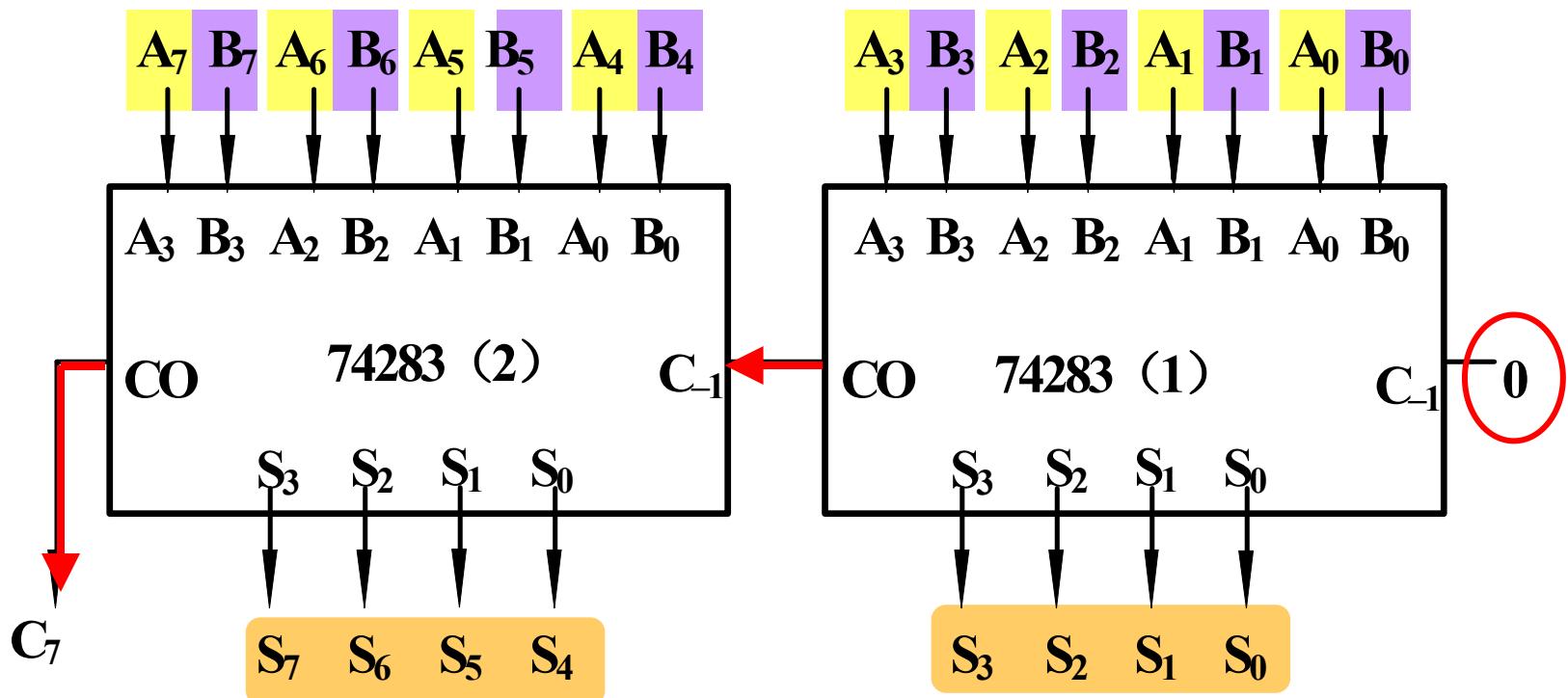
M	N	B_3	B_2	B_1	B_0	S
0	0	0	0	0	0	$I+0$
0	1	0	0	1	0	$I+2$
1	0	0	0	1	1	$I+3$
1	1	0	1	0	1	$I+5$

MN控制不同的加法功能

加法器串行进位扩展连接方式:

例:用两片74LS283构成一个8位二进制数加法器。

74LS283: 4位超前进位加法器



在片内是超前进位，而片与片之间是串行进位。

加法器并行进位扩展连接方式：

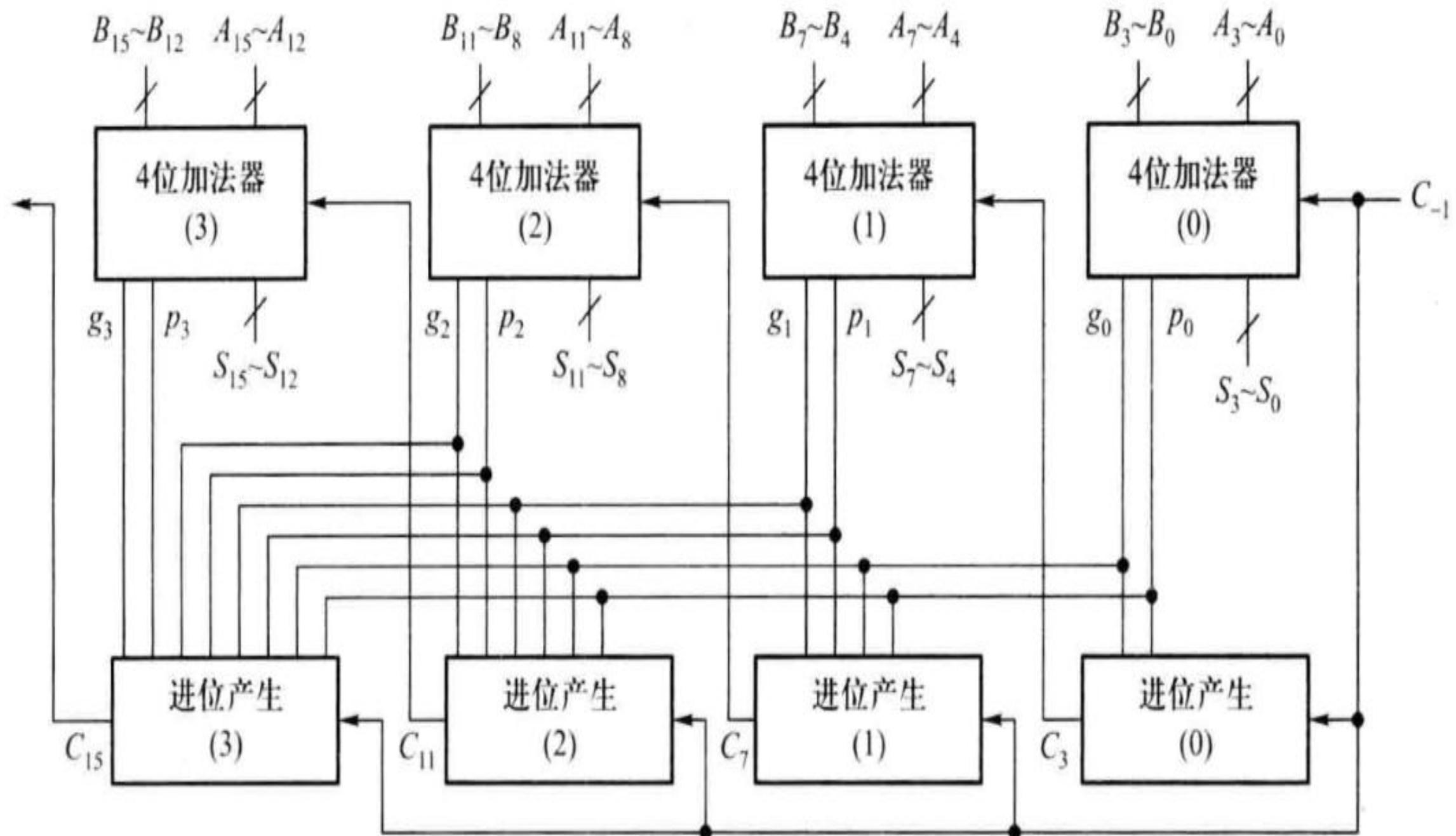


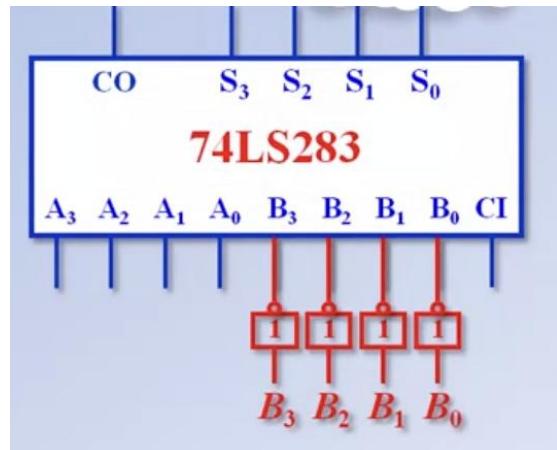
图 4.4.42 加法器并行进位扩展连接方式

3. 减法运算

在实际应用中，通常是将减法运算变为加法运算来处理，即采用加补码的方法完成减法运算

$$A - B = A + B_{\text{反}} + 1 - 2^N$$

适合 $A - B \geq 0$ 的情况。



1) $A - B \geq 0$ 的情况。

$$A = 0101, B = 0010$$

$$\begin{array}{r} \boxed{0} & 1 & 0 & 1 & A \\ \boxed{1} & 1 & 0 & 1 & B_{\text{反}} \\ + & & & & \\ \hline & & & 1 & \end{array}$$

舍弃

当 $A - B \geq 0$ 时，舍弃的进位为 1，所得结果就是差的原码，不需再求反补。

2) $A - B < 0$ 的情况。

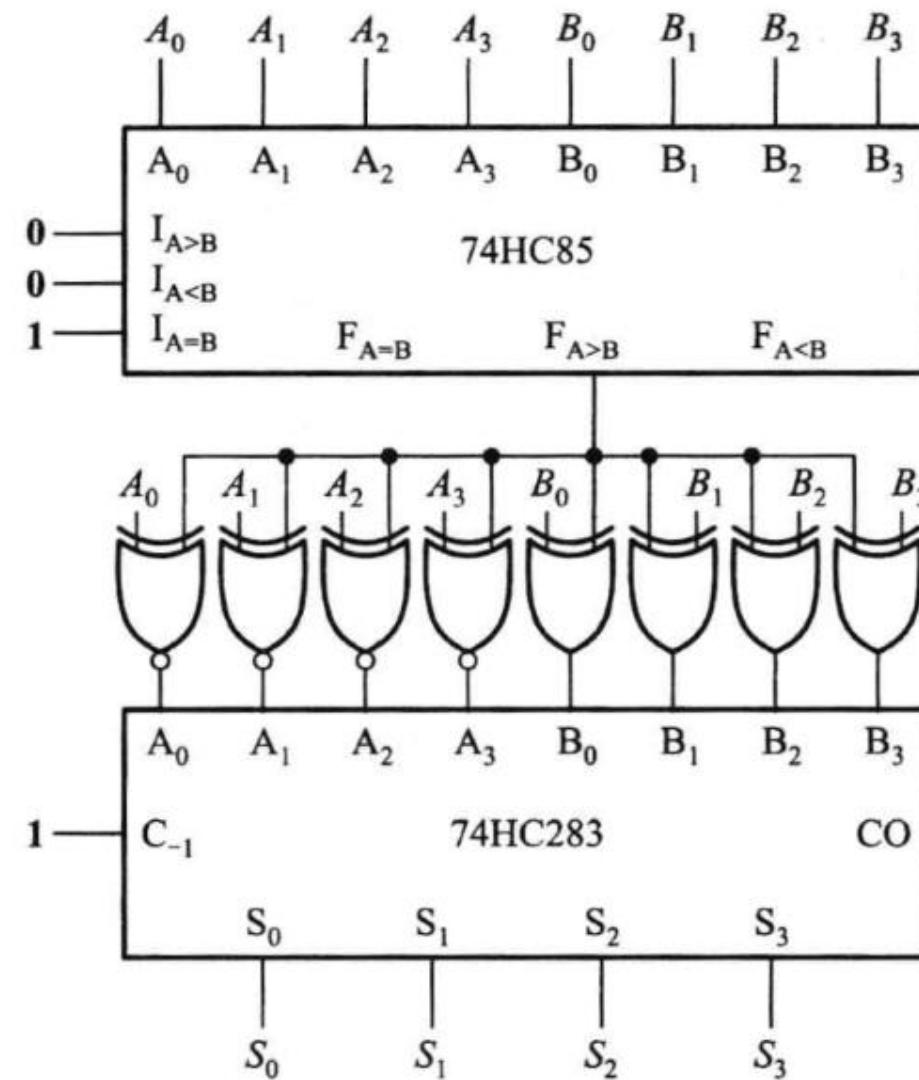
$$A = 0010, B = 0101$$

$$\begin{array}{r} \boxed{0} & 0 & 1 & 0 & A \\ \boxed{1} & 0 & 1 & 0 & B_{\text{反}} \\ + & & & & \\ \hline & & & 1 & \end{array}$$

舍弃

当 $A - B < 0$ 时，舍弃的进位为 0，所得结果是补码，要得到原码需再求补。

4.4.38 逻辑电路如图，试分析电路功能。



比较器与加法器组合题

当 $A > B$ 时, $F_{A>B} = 1$, 则加法器的输入为 $A \odot 1 = A$ 、 $B \oplus 1 = \bar{B}$, 其输出为 $S = A + \bar{B} + 1 = A - B$ 。
当 $A < B$ 时, $F_{A>B} = 0$, 则加法器的输入为 $A \odot 0 = \bar{A}$ 、 $B \oplus 0 = B$, 其输出为 $S = \bar{A} + B + 1 = B - A$ 。
该电路实现减法功能。

作业：

总结并列出所学的中规模集成电路芯片名称、符号及功能表

编码器4532, 147, 148

译码器139, 138, 4511

数值选择器151, 153

数值比较器85

加法器283

作业：

4.4.1 ---4.4.4

4.4.6 ---4.4.9 4.4.13 ---4.4.15

4.4.16 4.4.20---4.4.22 4.4.26---4.4.29

4.4.31 4.4.32

4.4.35 4.4.37 4.4.38