



第4章 常用组合逻辑功能器件

本章将介绍几种常用的中规模集成电路(**MSI**),这些中规模集成电路分别具有特定的**逻辑功能**,称为**功能模块**,用功能模块设计组合逻辑电路,具有许多优点.

中规模集成电路(**MSI Medium Scale Integration**),通常指含逻辑门数为10门~99门(或含元件数100个~999个)。





4.1 自顶向下的模块化设计方法

顶: 指系统功能,即系统总要求,较抽象.

向下: 指根据系统总要求,将系统分解为若干个子系统,再将每个子系统分解为若干个功能模块... ..,直至分成许多各具特定功能的基本模块为止.

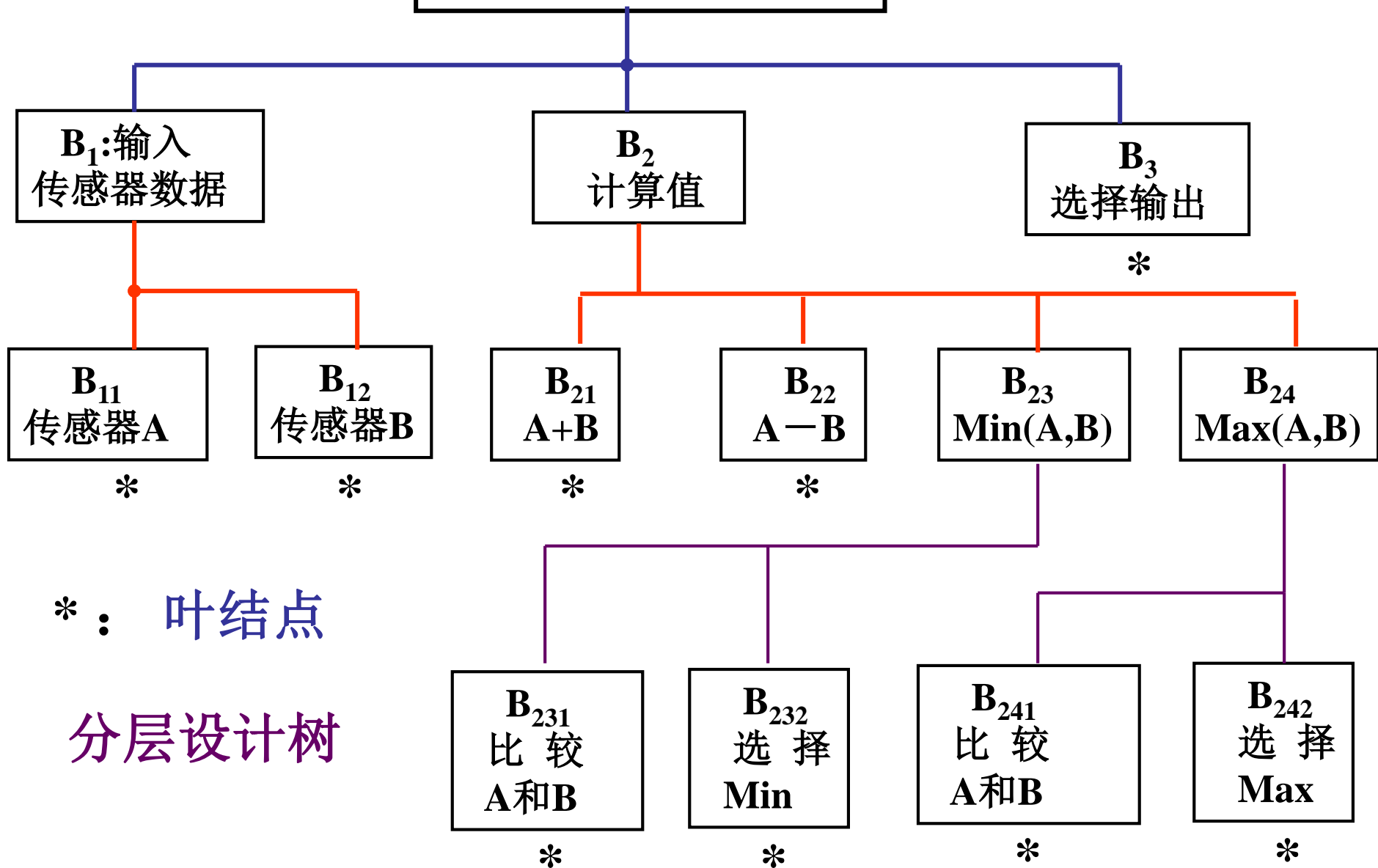
例: 设计一个数据检测系统,功能表如下:

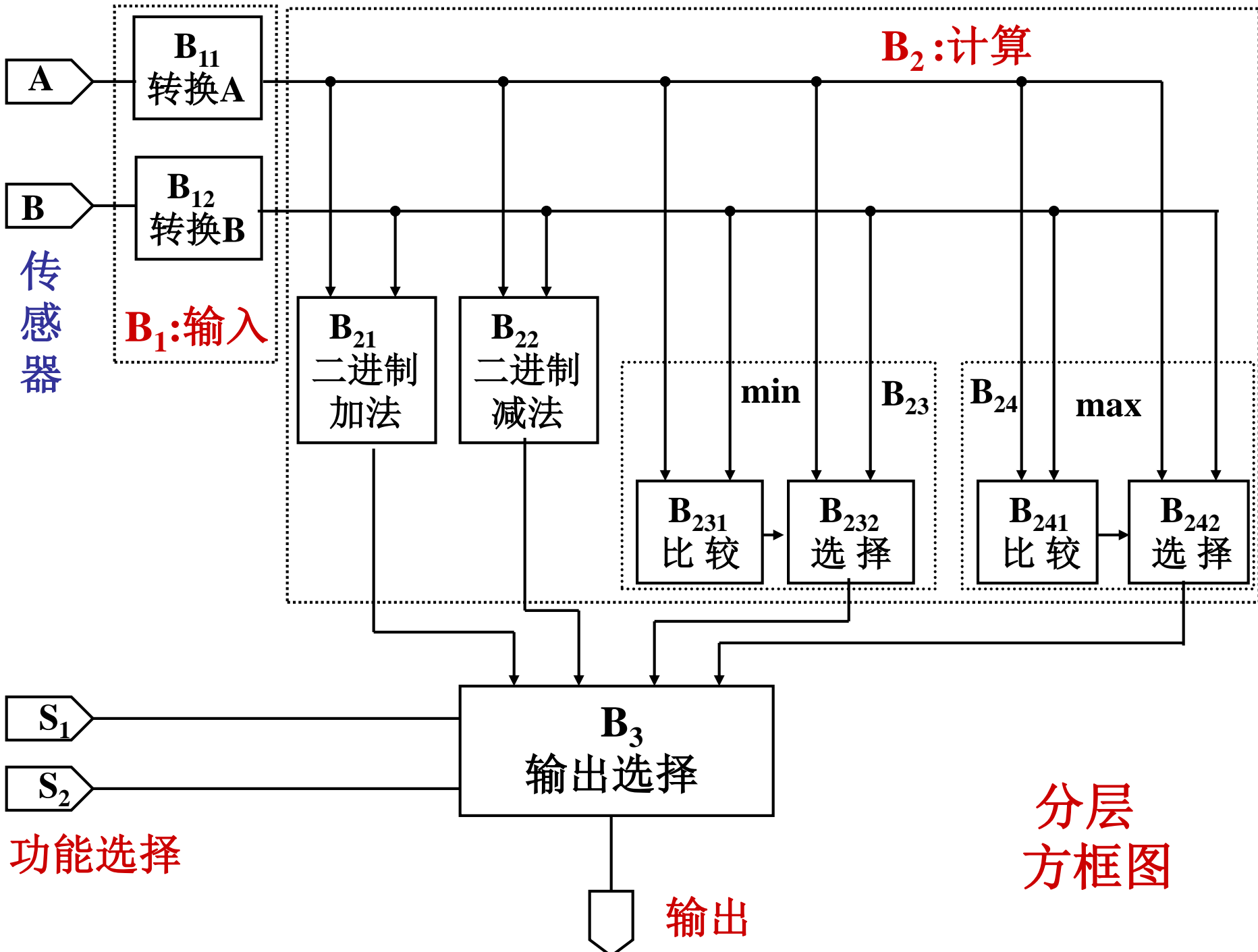
数据**A**、**B**分别来自两个传感器.

S_1	S_2	输出功能
0	0	$A + B$
0	1	$A - B$
1	0	$\text{Min}(A, B)$
1	1	$\text{Max}(A, B)$



B: 数据检测系统 顶层







4.2 编码器

将信息(如数和字符等)转换成符合一定规则的代码.

4.2.1 二进制编码器

用 n 位二进制代码对 $N=2^n$ 个特定信息进行编码的逻辑电路.

输入互相排斥编码器、优先编码器

设计方法: 以例说明





设计一个具有**输入互相排斥**条件的编码器。

输入: X_0 、 X_1 、 X_2 、 X_3

输出: A_1 、 A_0

对应关系:

输入	A_1	A_0
X_0	0	0
X_1	0	1
X_2	1	0
X_3	1	1



X_3	X_2	X_1	X_0	A_1	A_0
0	0	0	0	×	×
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	×	×
0	1	0	0	1	0
0	1	0	1	×	×
0	1	1	0	×	×
0	1	1	1	×	×
1	0	0	0	1	1
1	0	0	1	×	×
1	0	1	0	×	×
1	0	1	1	×	×
1	1	0	0	×	×
1	1	0	1	×	×
1	1	1	0	×	×
1	1	1	1	×	×

$X_3 \backslash X_2 \backslash X_1 X_0$	00	01	11	10
00	×	0	×	0
01	1	×	×	×
11	×	×	×	×
10	1	×	×	×

$$A_1 = X_2 + X_3$$

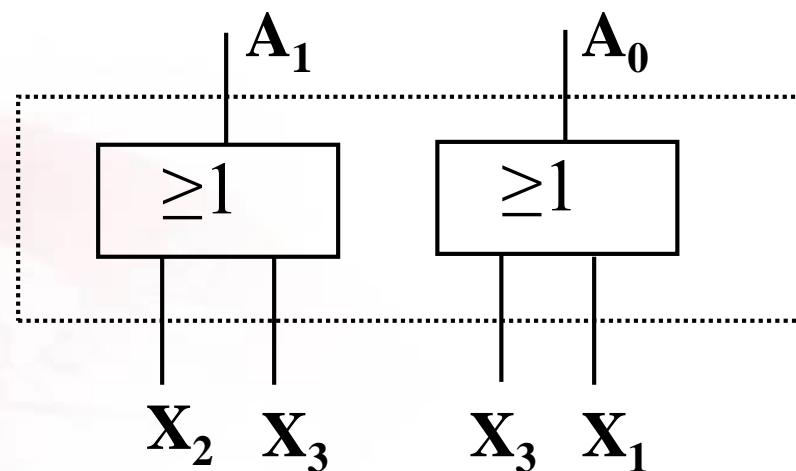
$X_3 \backslash X_2 \backslash X_1 X_0$	00	01	11	10
00	×	0	×	1
01	0	×	×	×
11	×	×	×	×
10	1	×	×	×

$$A_0 = X_1 + X_3$$



4线—2线编码器电路图：

- (1) 编码器在任何时候只允许有一个输入信号有效；
- (2) 电路无 X_0 输入端；
- (3) 电路无输入时,编码器的输出与 X_0 编码等效.





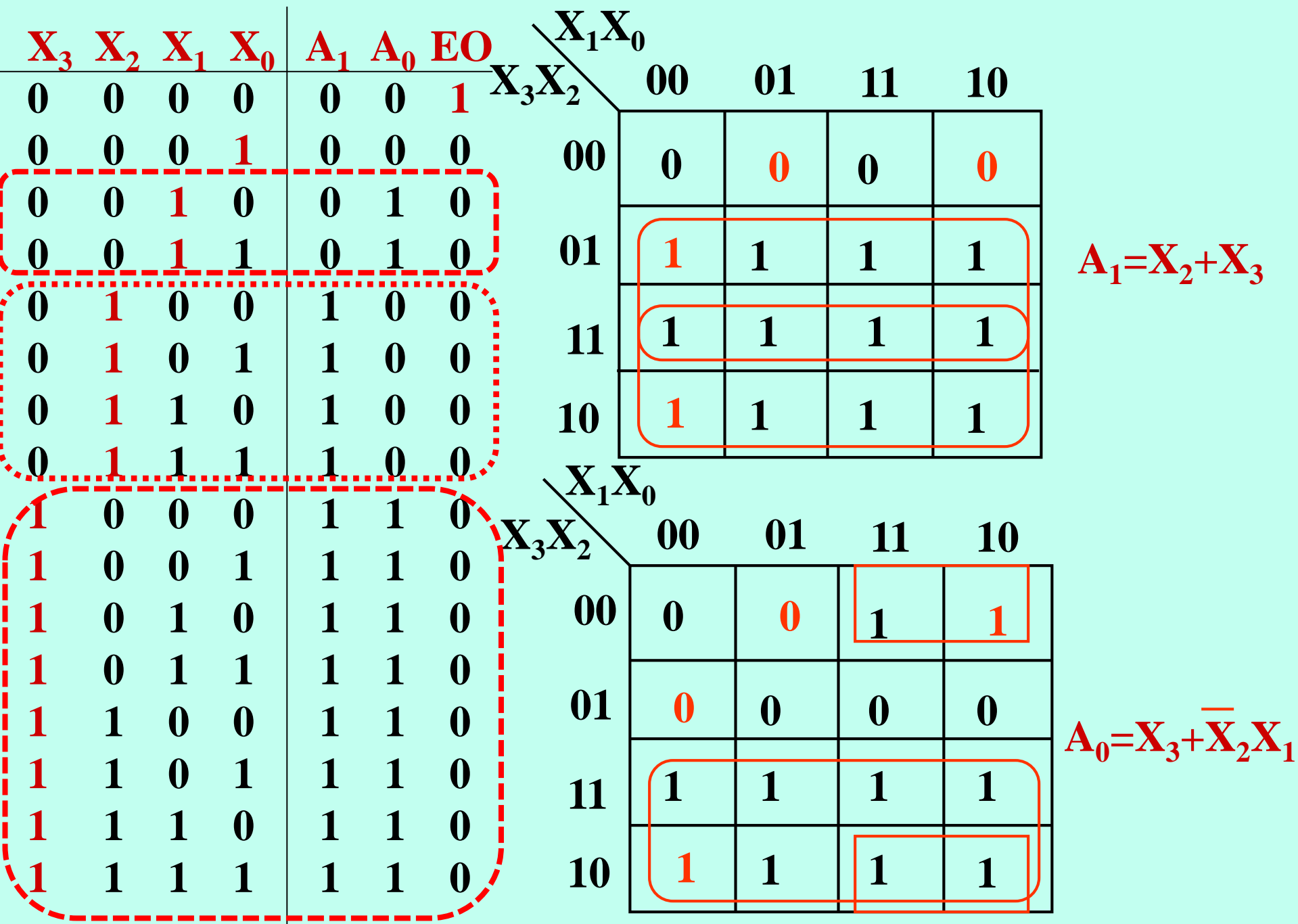
带输出使能(Enable)端的优先编码器:

输出使能端: 用于判别电路是否有信号输入.

优先: 对输入信号按轻重缓急排序,当有多个信号同时输入时,只对优先权高的一个信号进行编码.

下面把上例4线—2线编码器改成带输出使能(Enable)端的优先编码器,假设输入信号优先级的次序为: X_3, X_2, X_1, X_0 .

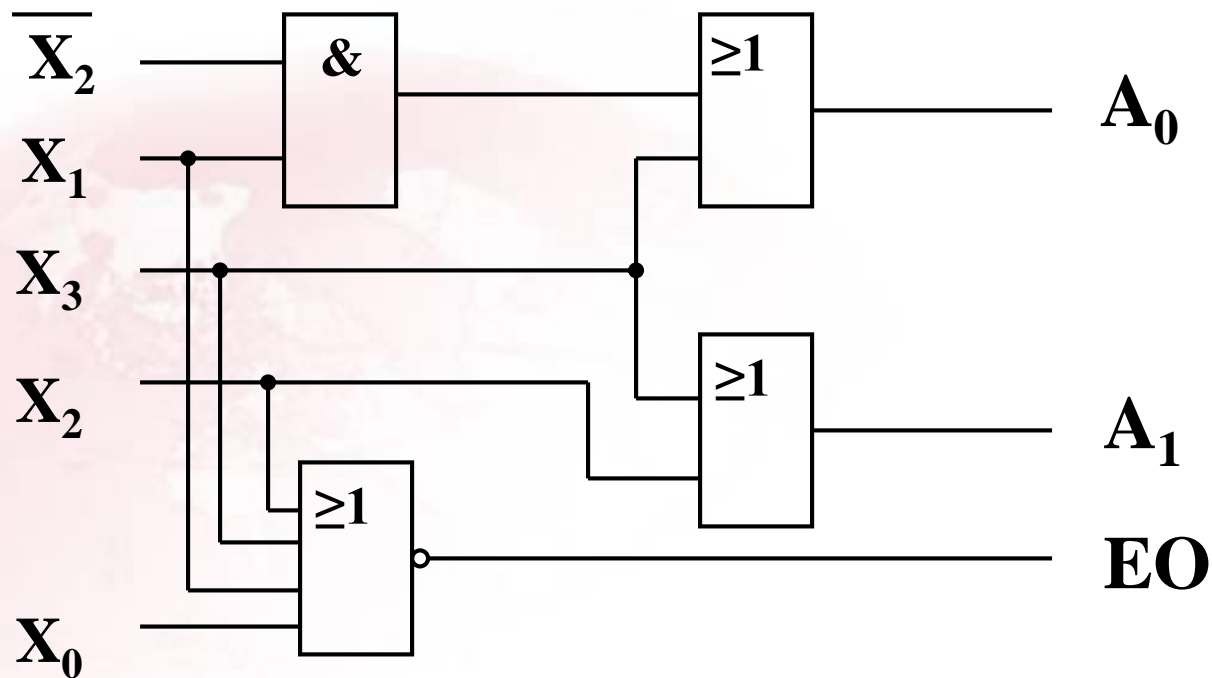






$$EO = \overline{X_3} \overline{X_2} \overline{X_1} \overline{X_0} = \overline{X_3 + X_2 + X_1 + X_0}$$

编码器
电路图





4.2.2 二—十进制编码器

输入: $I_0, I_1, I_2 \dots \dots I_9$, 表示十个要求编码的信号.

输出: BCD码.

电路有十根输入线, 四根输出线, 常称为**10线—4线**编码器



4.2.3 通用编码器集成电路

1. 8线—3线优先编码器74148



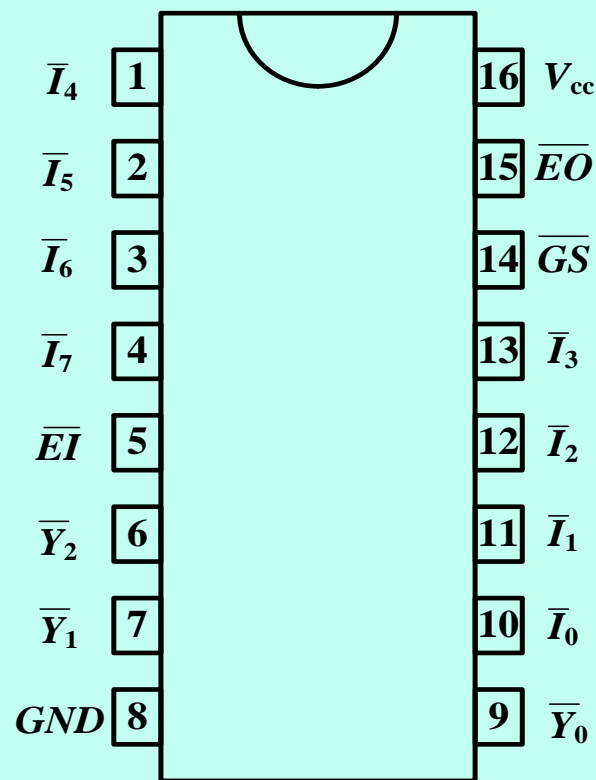
High priority

输入使能

输入低电平有效



逻辑图



引脚图



74148功能表

输入使能

扩展输出

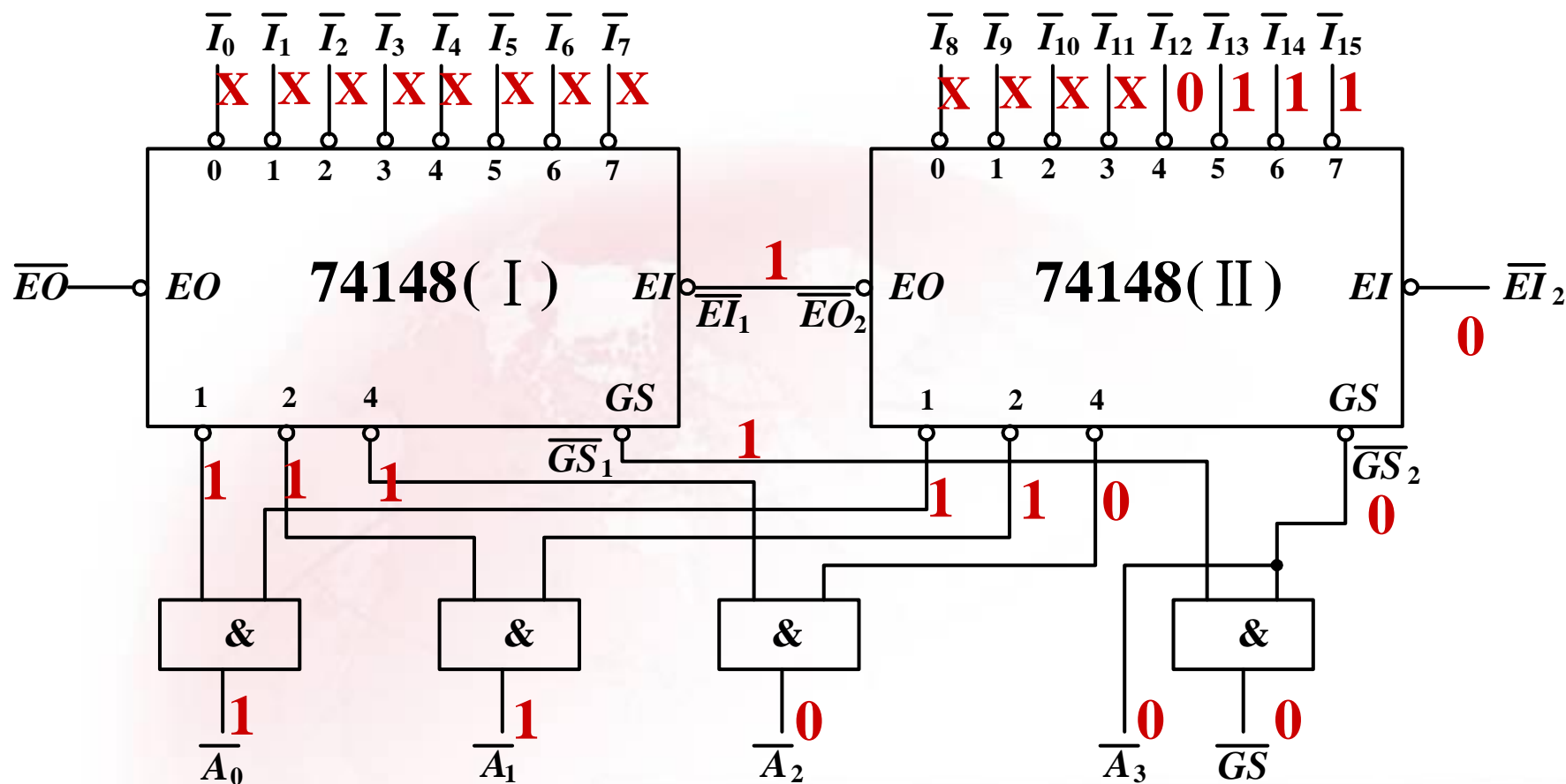
选通输出

输 入									输 出				
\overline{EI}	$\overline{I_0}$	$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	$\overline{I_4}$	$\overline{I_5}$	$\overline{I_6}$	$\overline{I_7}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$	\overline{GS}	\overline{EO}
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	×	×	×	×	×	×	0	1	0	0	1	0	1
0	×	×	×	×	×	0	1	1	0	1	0	0	1
0	×	×	×	0	1	1	1	1	0	1	1	0	1
0	×	×	0	1	1	1	1	1	1	0	0	0	1
0	×	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

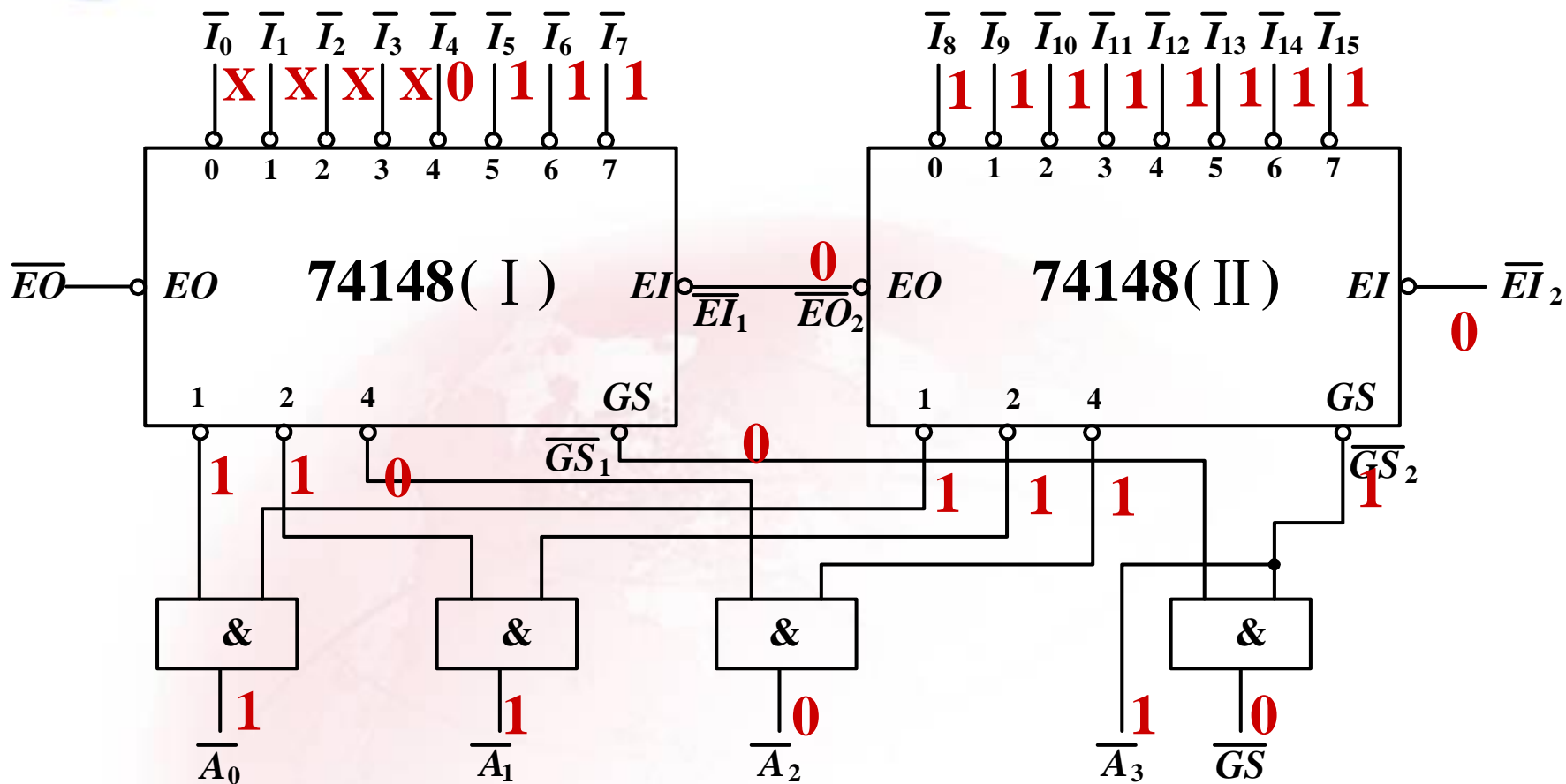
讨论题：如何写出 \overline{GS} 和 \overline{EO} 的表达式？



例：用两片74148构成16线—4线优先编码器。
高位芯片工作情况：



低位芯片工作情况:





思考题:

如何用若干片74148构成一个32线—5线编码器，电路如何设计？

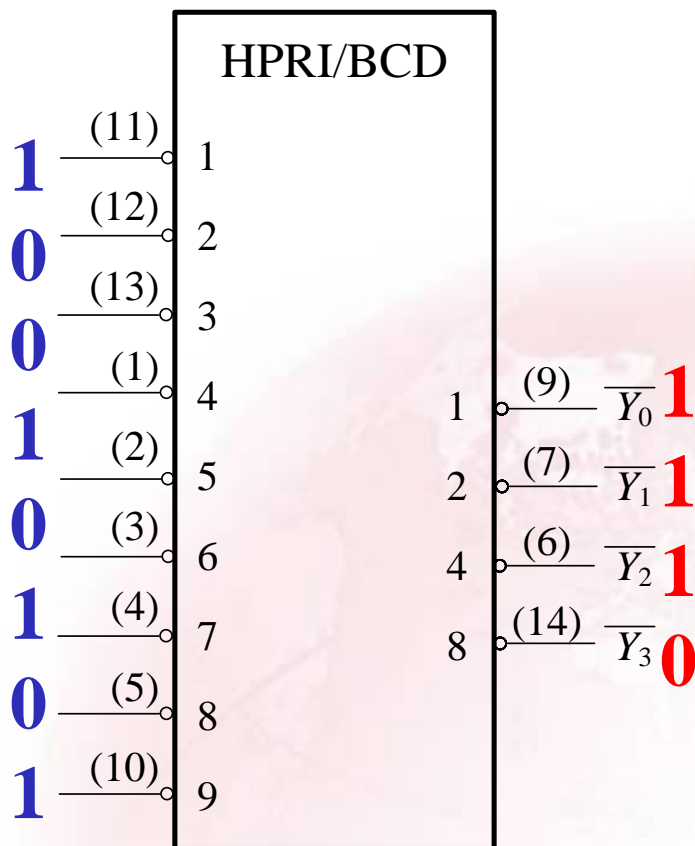
扩展电路设计提示：

- 1) 观察上例编码器低三位输出电路结构，并找出规律；
- 2) 分析高位输出和各 \overline{GS} 之间的关系，将 \overline{GS} 作为输入，高位信号作为输出，设计一输出电路。

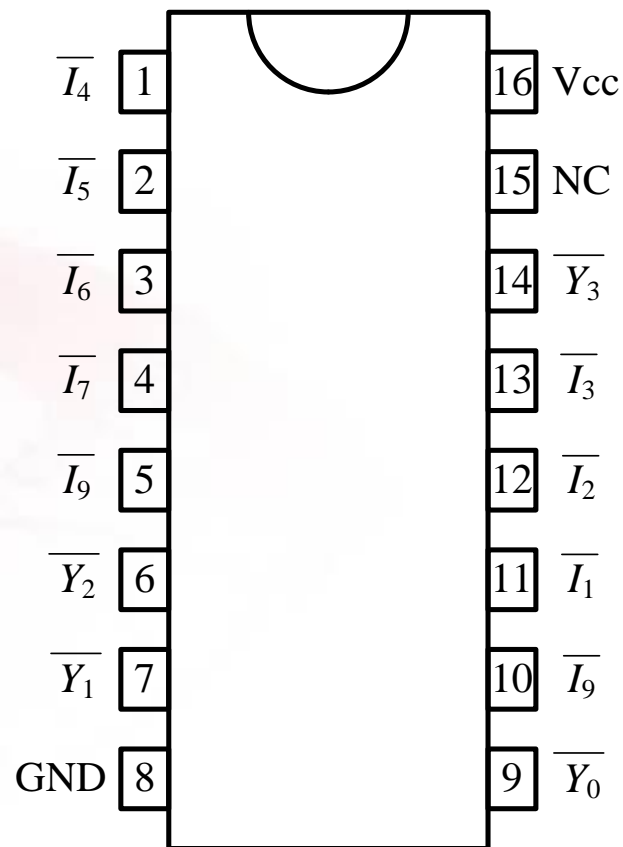




2. 10线—4线优先编码器74147



逻辑图



引脚图





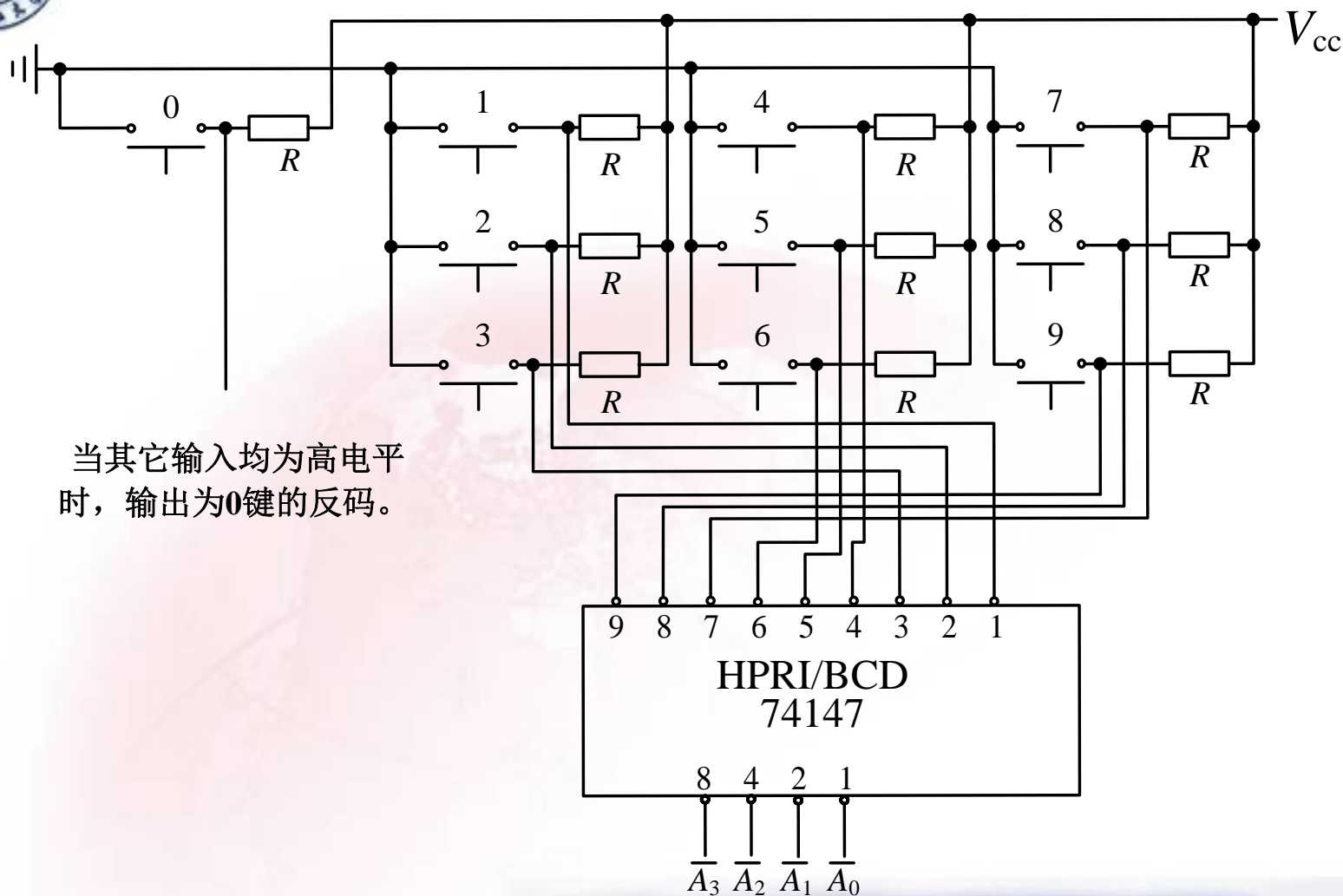
74147功能表

十进制数	输 入									输 出			
	$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	$\overline{I_4}$	$\overline{I_5}$	$\overline{I_6}$	$\overline{I_7}$	$\overline{I_8}$	$\overline{I_9}$	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
0	1	1	1	1	1	1	1	1	1	1	1	1	1
9	×	×	×	×	×	×	×	×	0	0	1	1	0
8	×	×	×	×	×	×	×	0	1	0	1	1	1
7	×	×	×	×	×	×	0	1	1	1	0	0	0
6	×	×	×	×	×	0	1	1	1	1	0	0	1
5	×	×	×	×	0	1	1	1	1	1	0	1	0
4	×	×	×	0	1	1	1	1	1	1	0	1	1
3	×	×	0	1	1	1	1	1	1	1	1	0	0
2	×	0	1	1	1	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	1	0





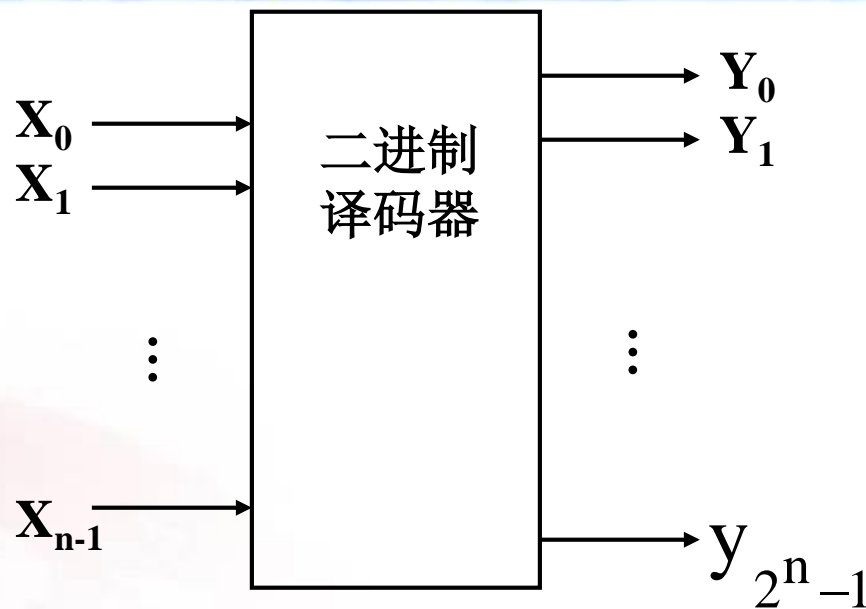
4.2.4 编码器应用举例





4.3 译码器/数据分配器

译码是编码的逆过程，作用是将一组码转换为确定信息。



4.3.1 二进制译码器

输入： 二进制代码，有 n 个；

输出： 2^n 个特定信息。

1. 译码器电路结构

以2线—4线译码器为例说明

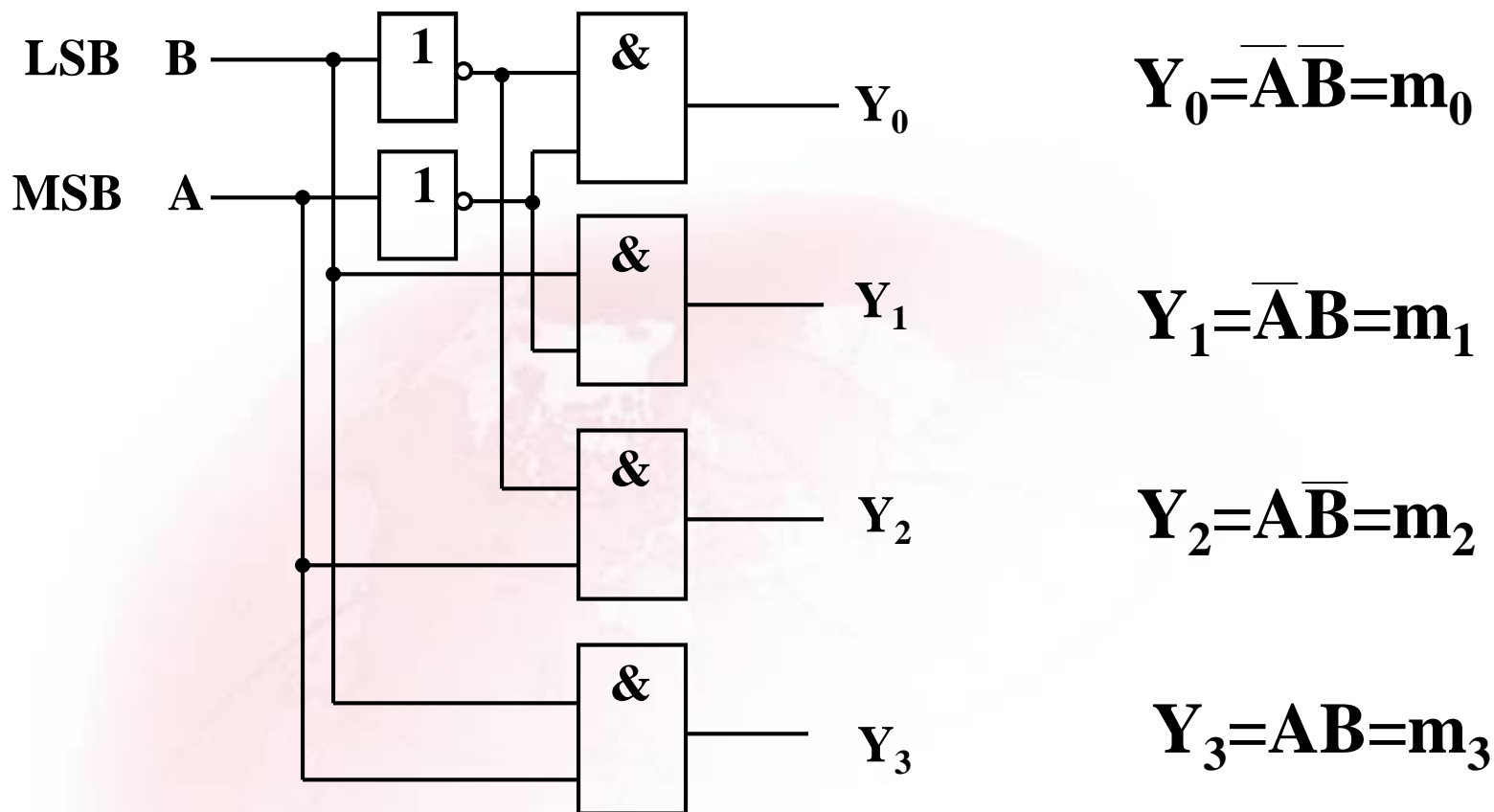
真值表为：

A	B	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1





高电平输出有效的2线-4线译码器电路图



思考：若输出为低电平有效，则输出表达式如何表示？

高电平有效2-4译码器真值表

A	B	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

低电平有效2-4译码器真值表

A	B	\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

由真值表容易得出:

① 高电平输出有效二进制译码器,其输出逻辑表达式为:

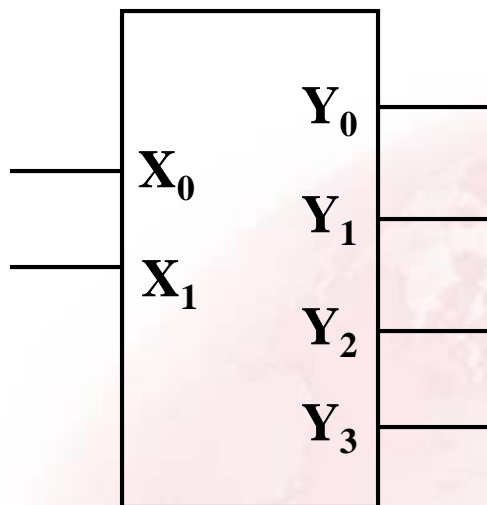
$$Y_i = m_i \quad (m_i \text{ 为输入变量所对应的最小项})$$

② 低电平输出有效二进制译码器,其输出逻辑表达式为:

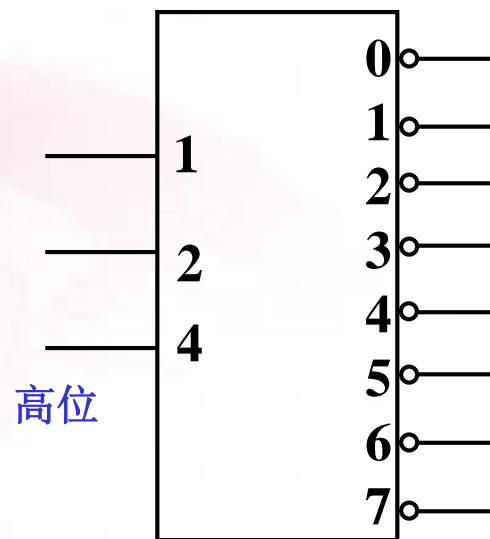
$$\overline{Y}_i = \overline{m}_i \quad (m_i \text{ 为输入变量所对应的最小项})$$



二进制译码器逻辑符号



2线——4线二进制译码器



3线——8线二进制译码器
(输出低有效)





* 用译码器实现组合逻辑函数

原理：二进制译码器能产生输入信号的全部最小项,而所有组合逻辑函数均可写成最小项之和的形式.

例：试用3线–8线译码器和逻辑门实现下列函数

$$\begin{aligned} F(Q,X,P) &= \sum m(0, 1, 4, 6, 7) \\ &= \prod M(2, 3, 5) \end{aligned}$$

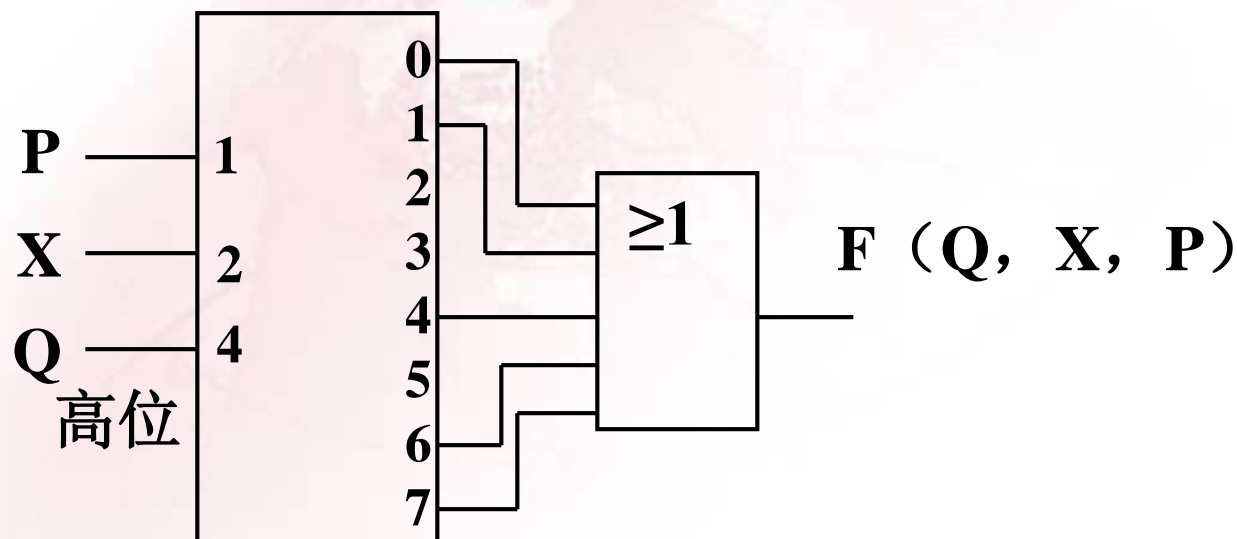




解题的几种方法：

(1) 利用高电平输出有效的译码器和或门。

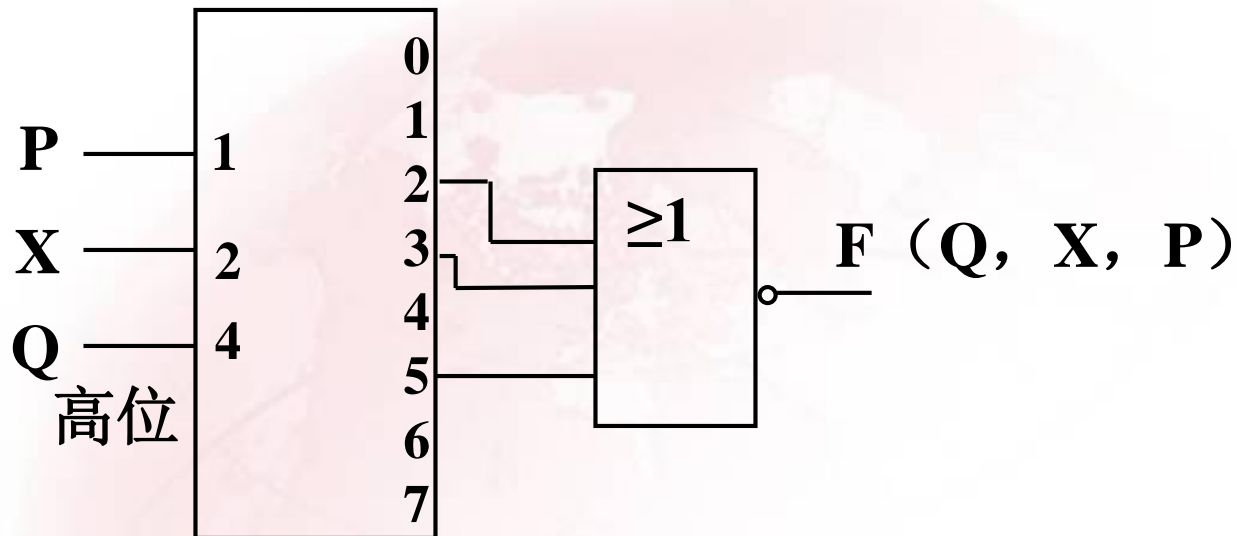
$$F(Q,X,P)=m_0+m_1+m_4+m_6+m_7$$





(2) 利用高电平输出有效的译码器和或非门。

$$F(Q, X, P) = m_0 + m_1 + m_4 + m_6 + m_7 = \overline{m_2 + m_3 + m_5}$$

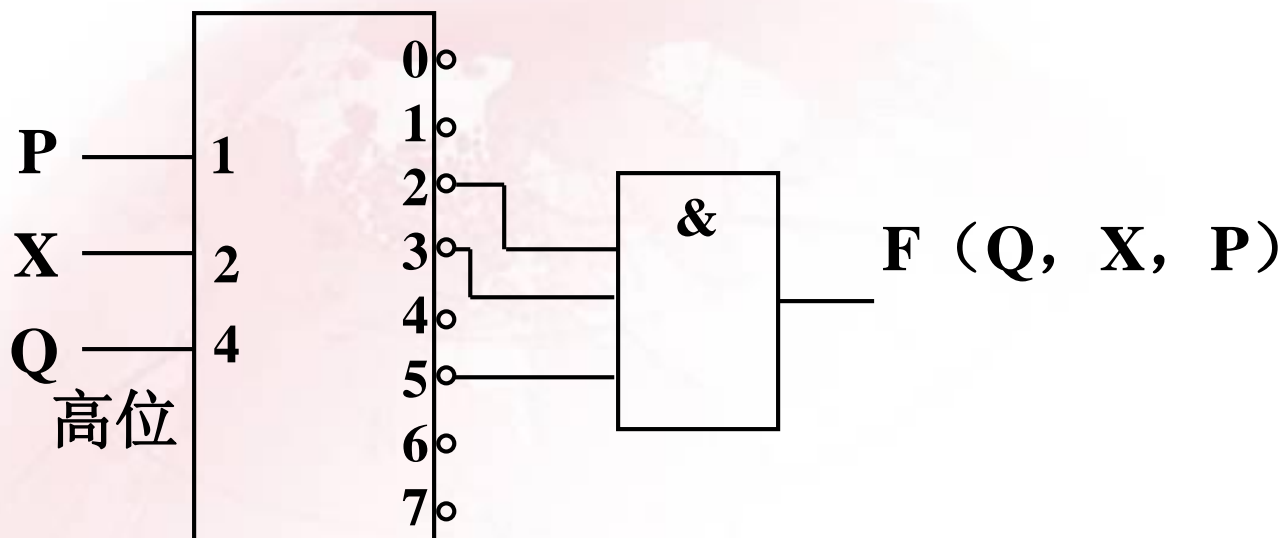




(3) 利用低电平输出有效的译码器和与门。

$$F(Q, X, P) = m_0 + m_1 + m_4 + m_6 + m_7 = M_2 M_3 M_5$$

$$= \overline{m}_2 \overline{m}_3 \overline{m}_5$$

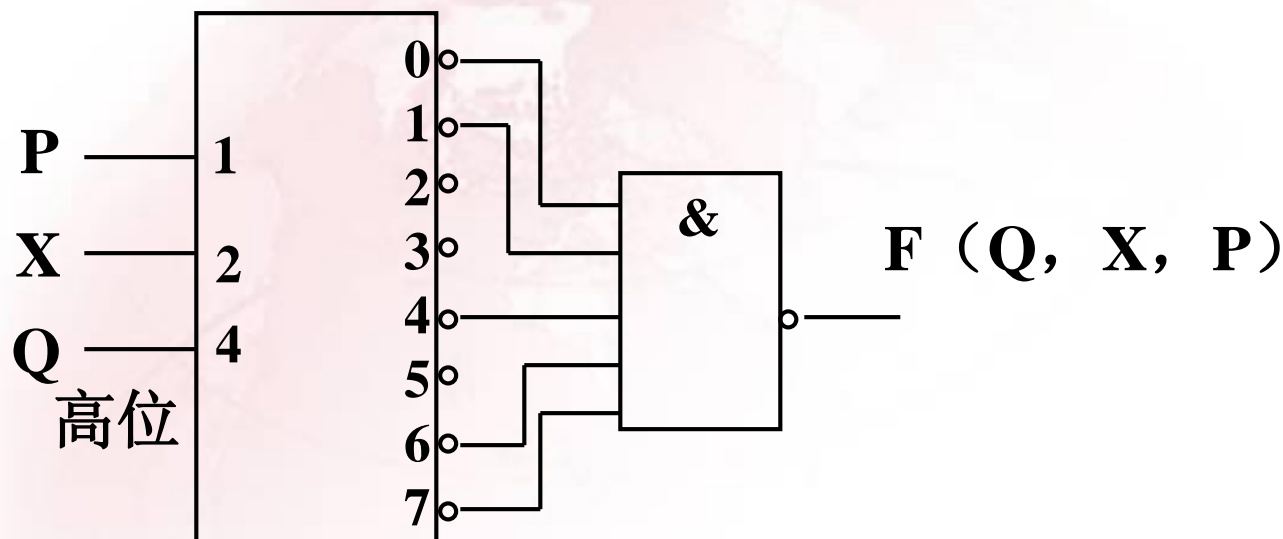




(4) 利用低电平输出有效的译码器和与非门。

$$F(Q, X, P) = m_0 + m_1 + m_4 + m_6 + m_7$$

$$= \overline{m_0 m_1 m_4 m_6 m_7}$$





2. 译码器的使能控制输入端

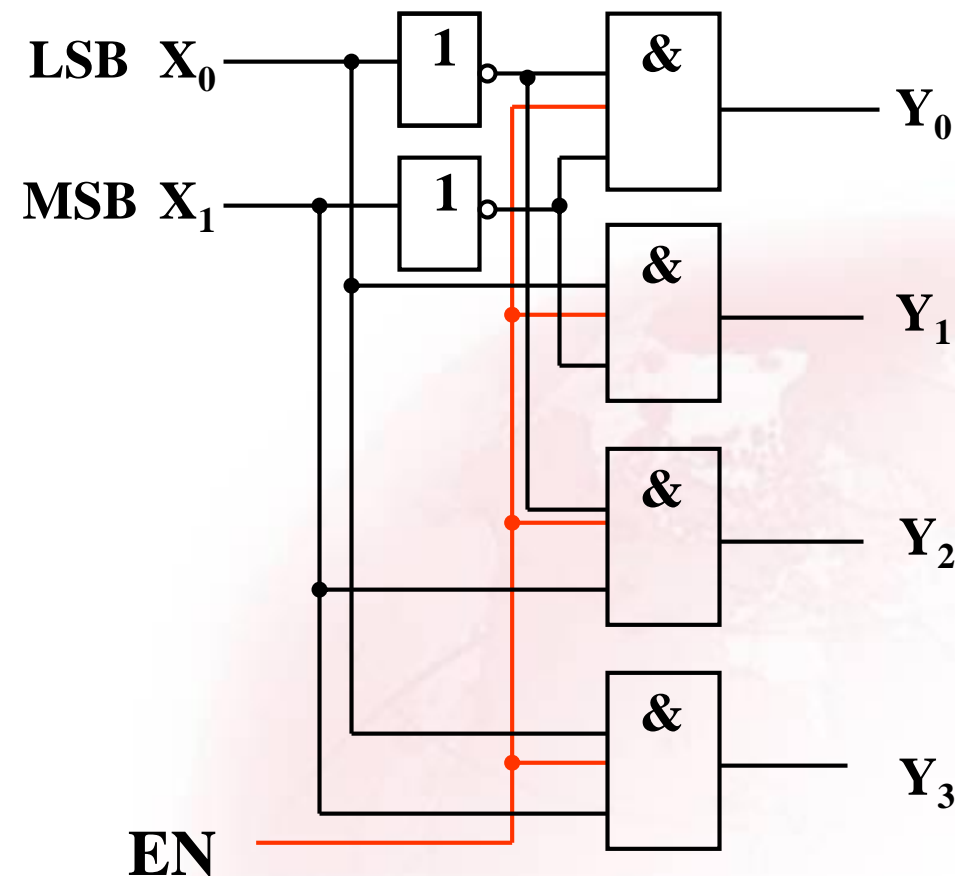
- 1) 利用使能输入控制端，既能使电路正常工作，也能使电路处于禁止工作状态；
- 2) 利用使能输入控制端，能够实现译码器容量扩展。



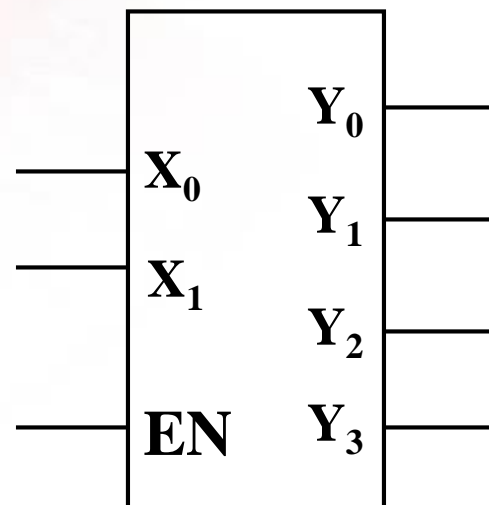


EN为使能控制输入端，
EN=0，输出均为0；
EN=1，输出译码信号。

电路满足： $Y_i = m_i EN$



逻辑图



逻辑符号



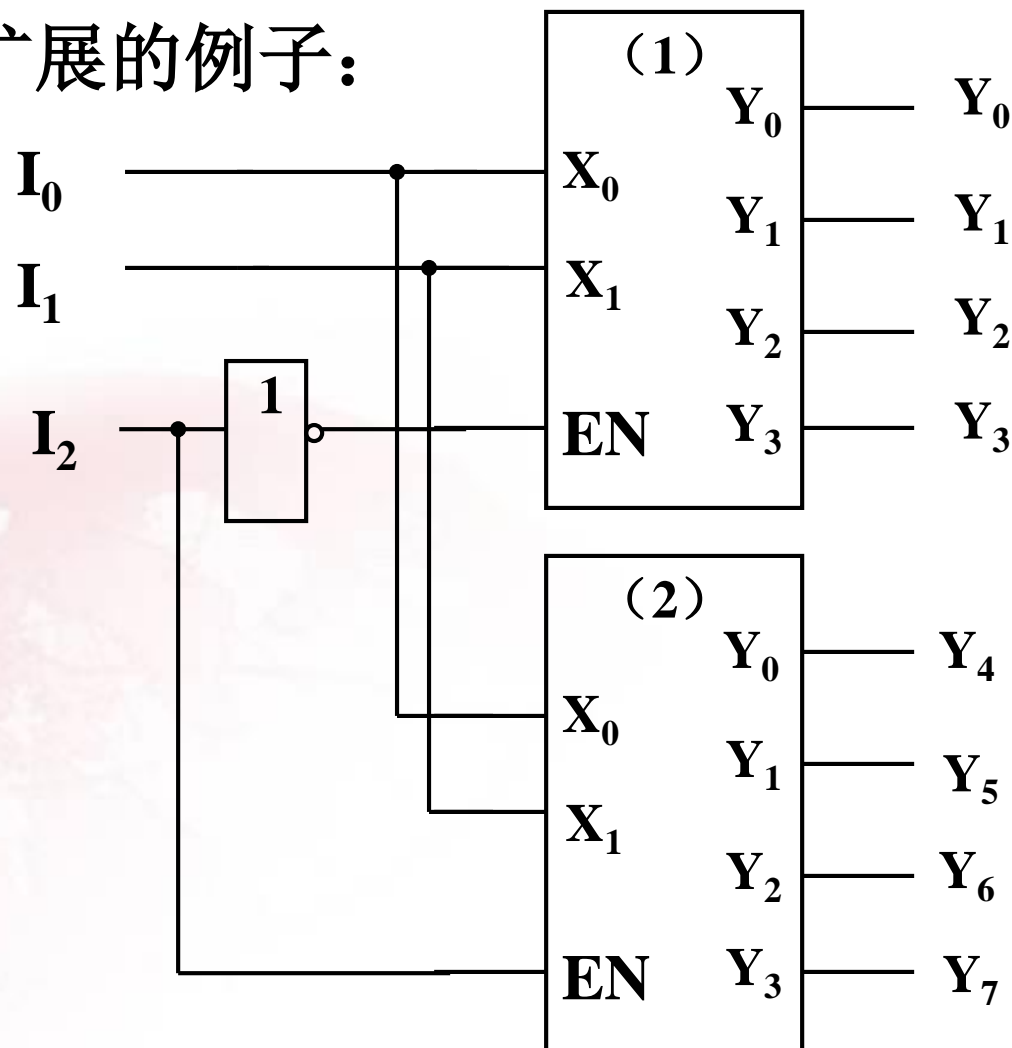


利用使能端实现扩展的例子：

由两片2线—4线译码器
组成3线—8线译码器

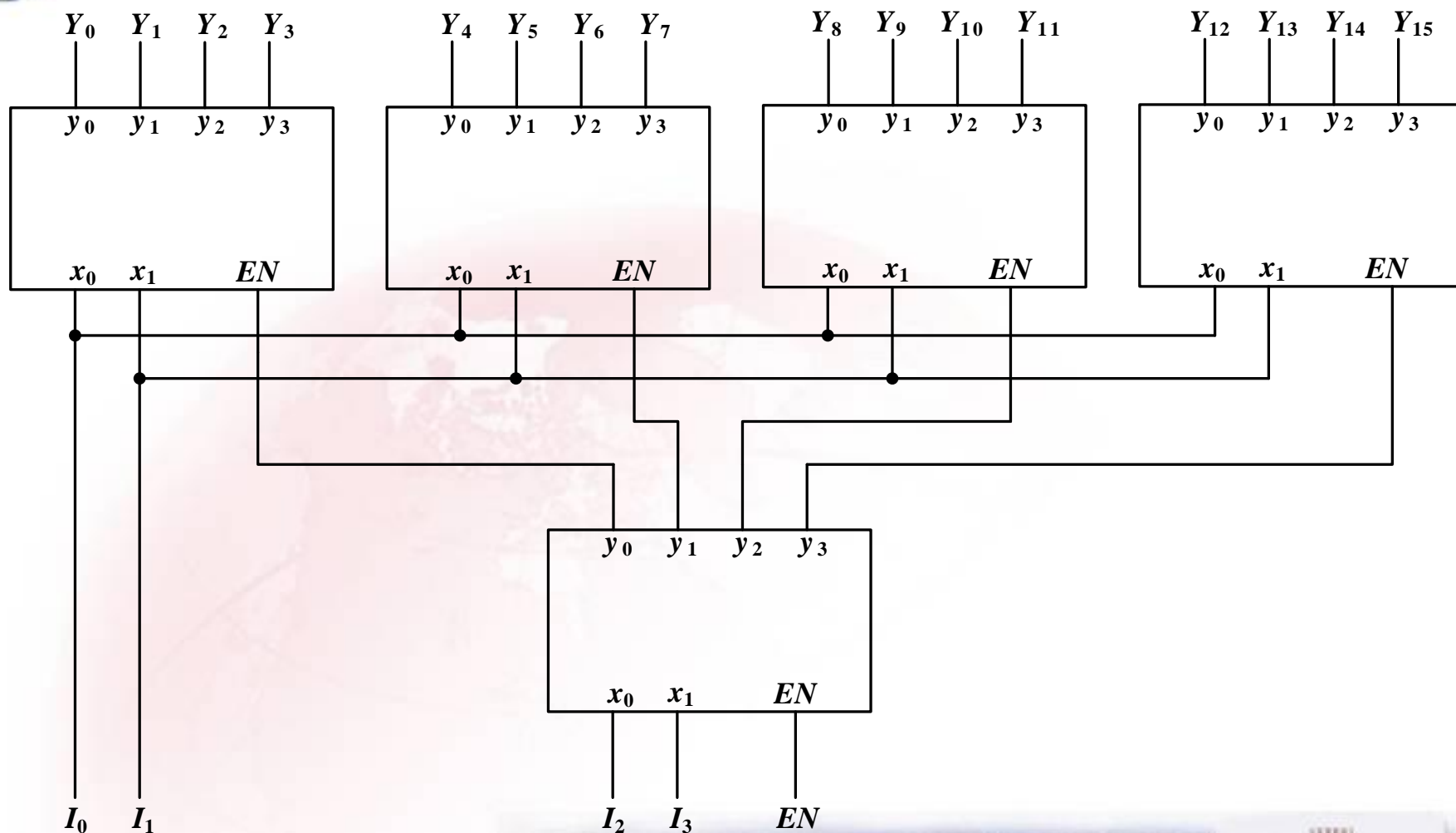
当 $I_2=0$ 时，(1)片工作，
(2)片禁止。

当 $I_2=1$ 时，(1)片禁止，
(2)片工作。





思考： 将2线—4线译码器扩展成4线--16线译码器



4.3.2 二—十进制译码器 (常称4线—10线译码器)

输入：BCD码.

输出：十个高、低电平.

真值表

A_3	A_2	A_1	A_0	\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3	\overline{Y}_4	\overline{Y}_5	\overline{Y}_6	\overline{Y}_7	\overline{Y}_8	\overline{Y}_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
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	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	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

输出低电平有效

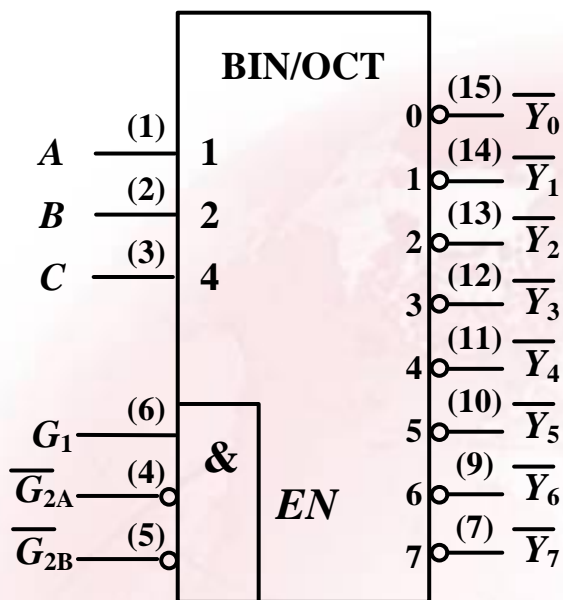
伪码

拒伪码

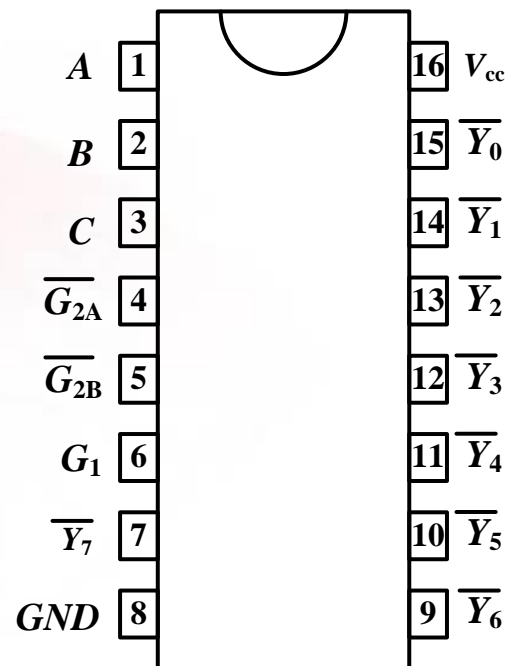


4.3.3 通用译码器集成电路

(1) 74138 带使能端3线—8线译码器



逻辑图

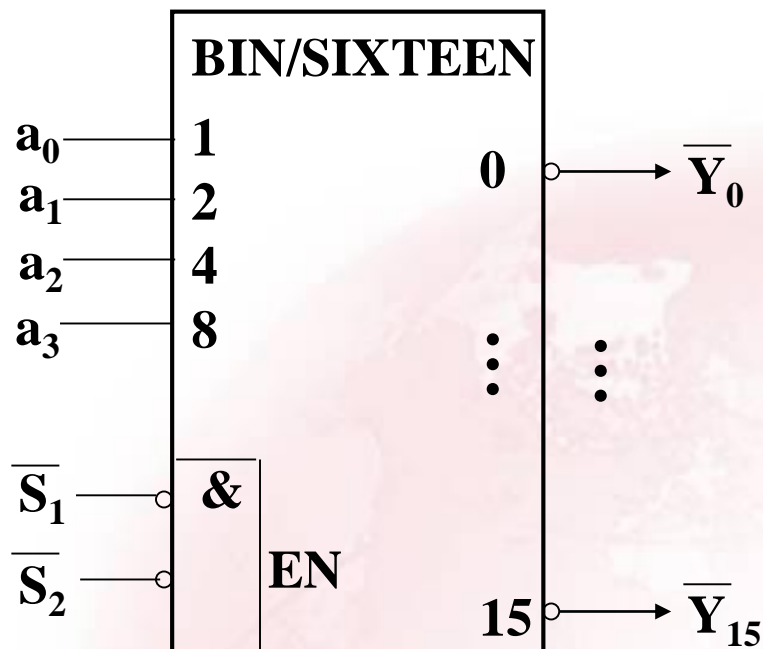


引脚图



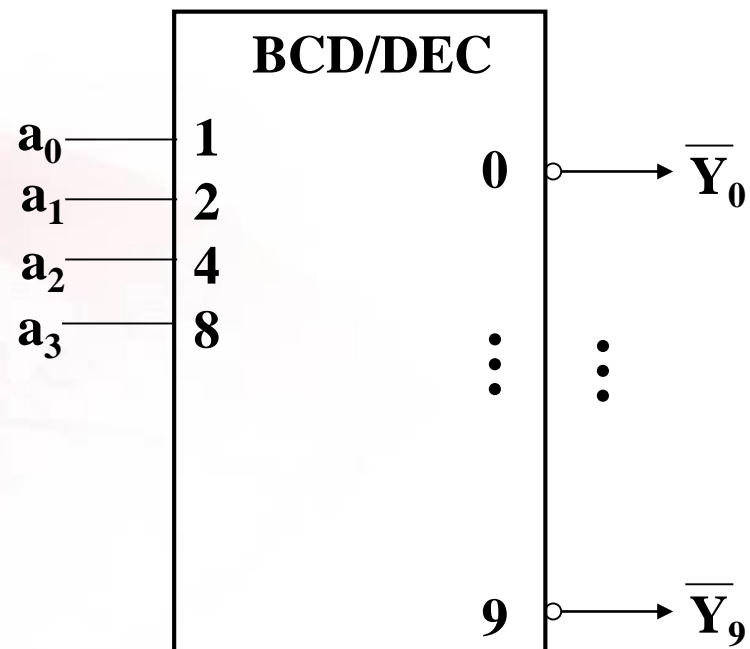


(2) 74154



4线—16线译码器

(3) 7442



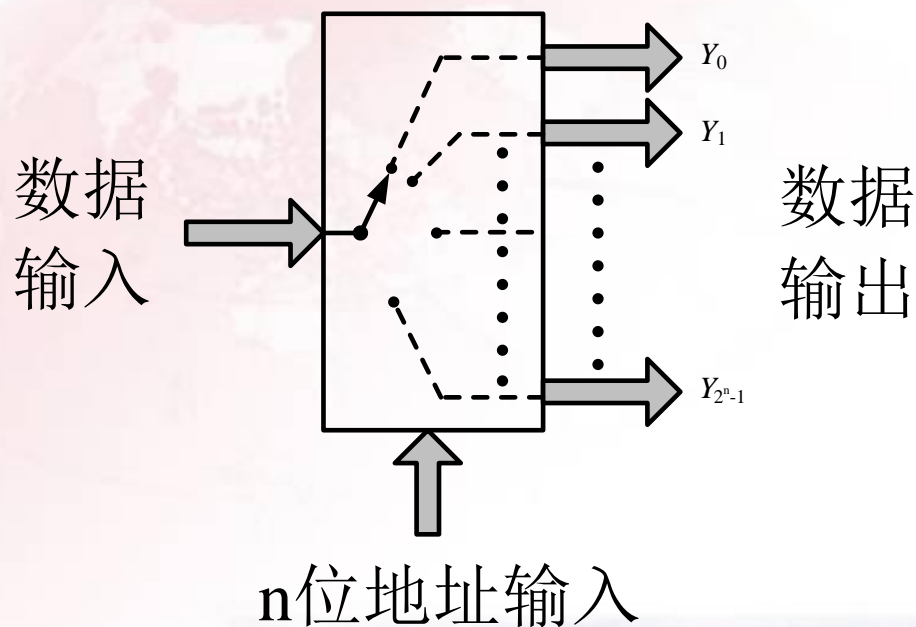
4线—10线译码器





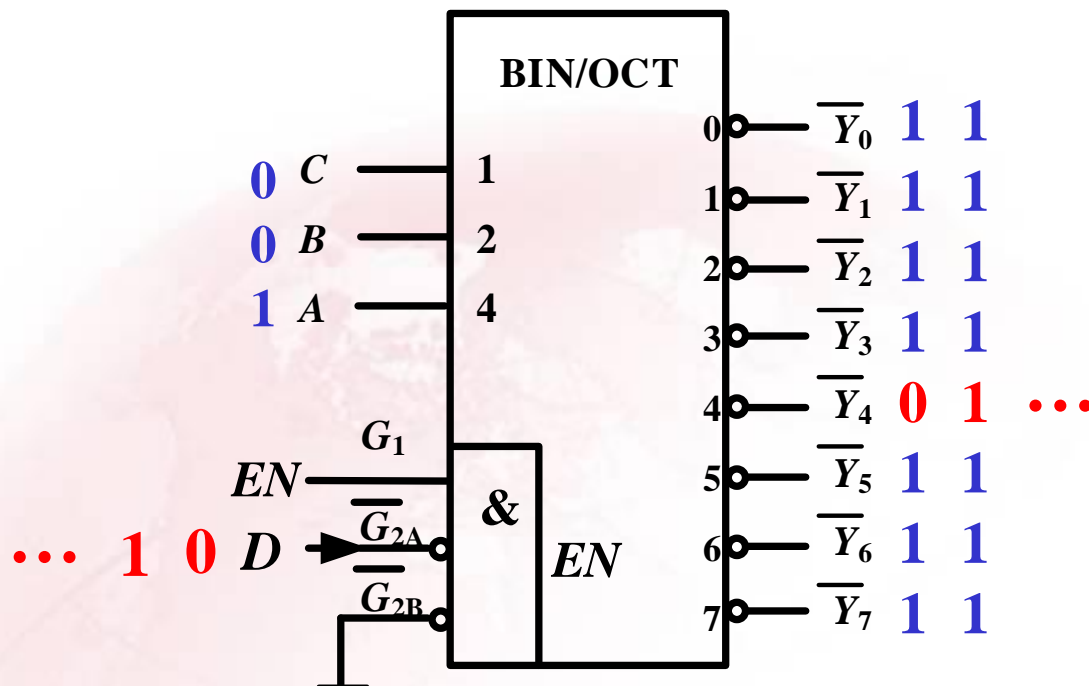
4.3.4 数据分配器

数据分配是将一个数据源输入的数据根据需要送到不同的输出端上去，实现数据分配功能的逻辑电路称为数据分配器。分配器又叫多路复用器。





数据分配器一般用带使能控制端的二进制译码器实现。



用74138作为数据分配器



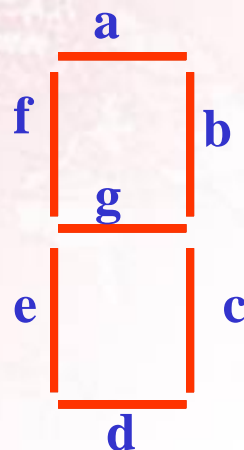


4.3.5显示译码器

用于驱动数码显示器，使其显示有用的 字符或图形

1. 半导体数码管（Light Emitting Diode简称LED）

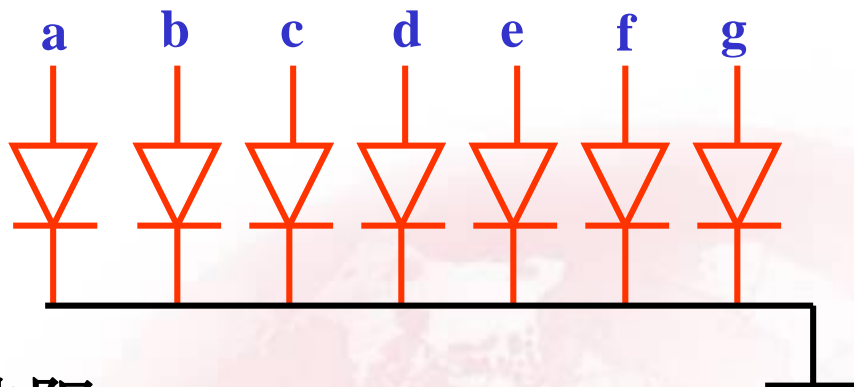
七段数码管
显示器





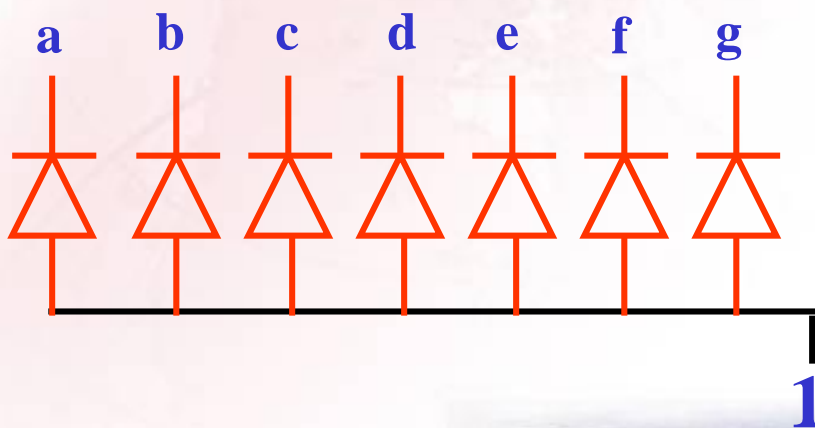
七段数码管的两种连接方法：

① 共阴

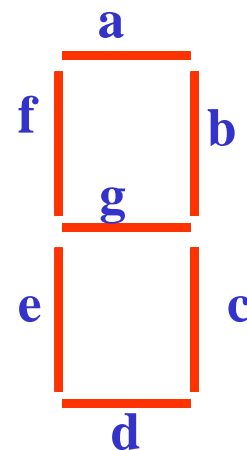


阳极加高电平字段亮。

② 共阳



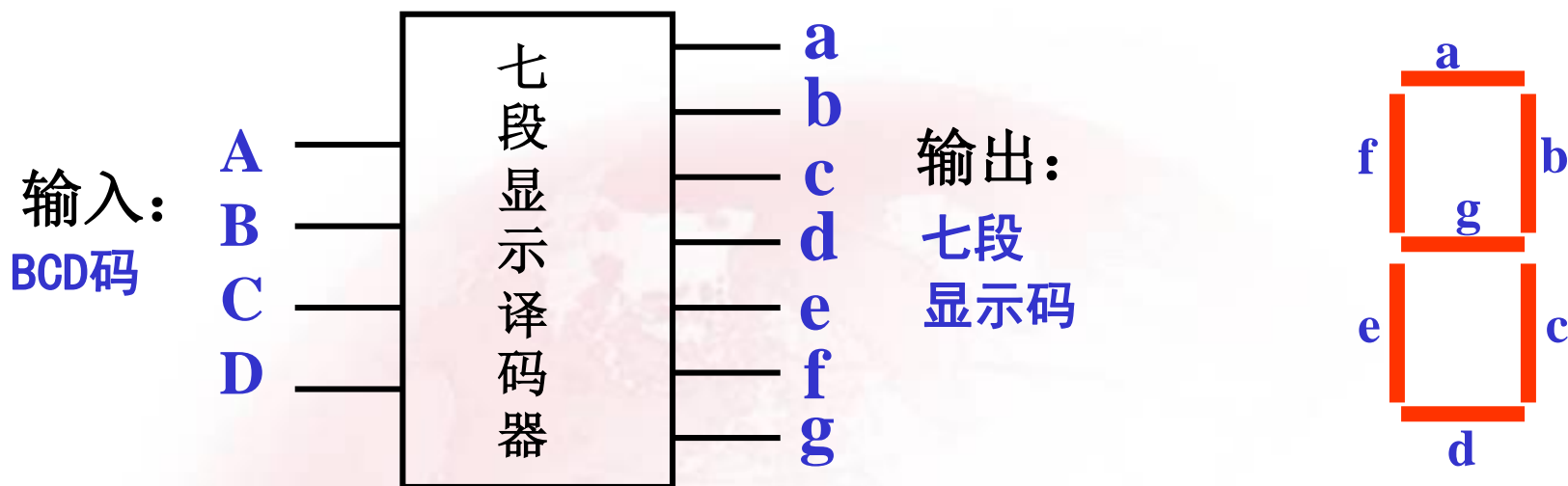
阴极加低电平字段亮。





显示译码器设计

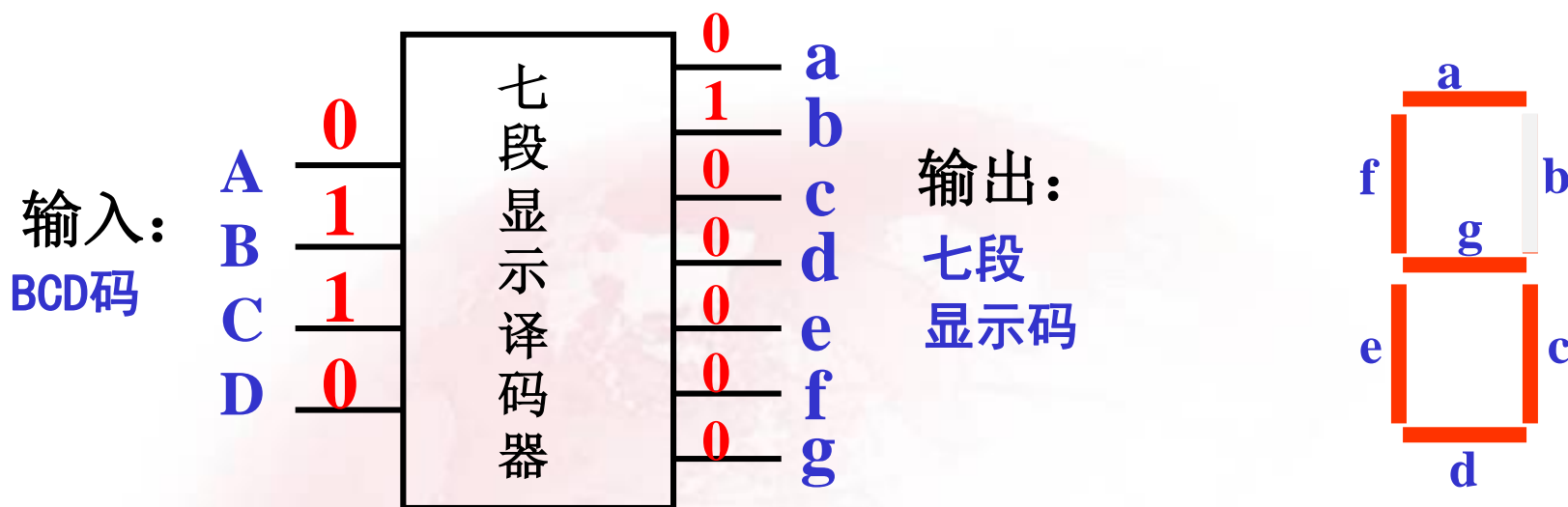
功能：将表示数字的**BCD**码转换成**七段显示码**。





显示译码器设计

功能：将表示数字的**BCD**码转换成**七段显示码**。



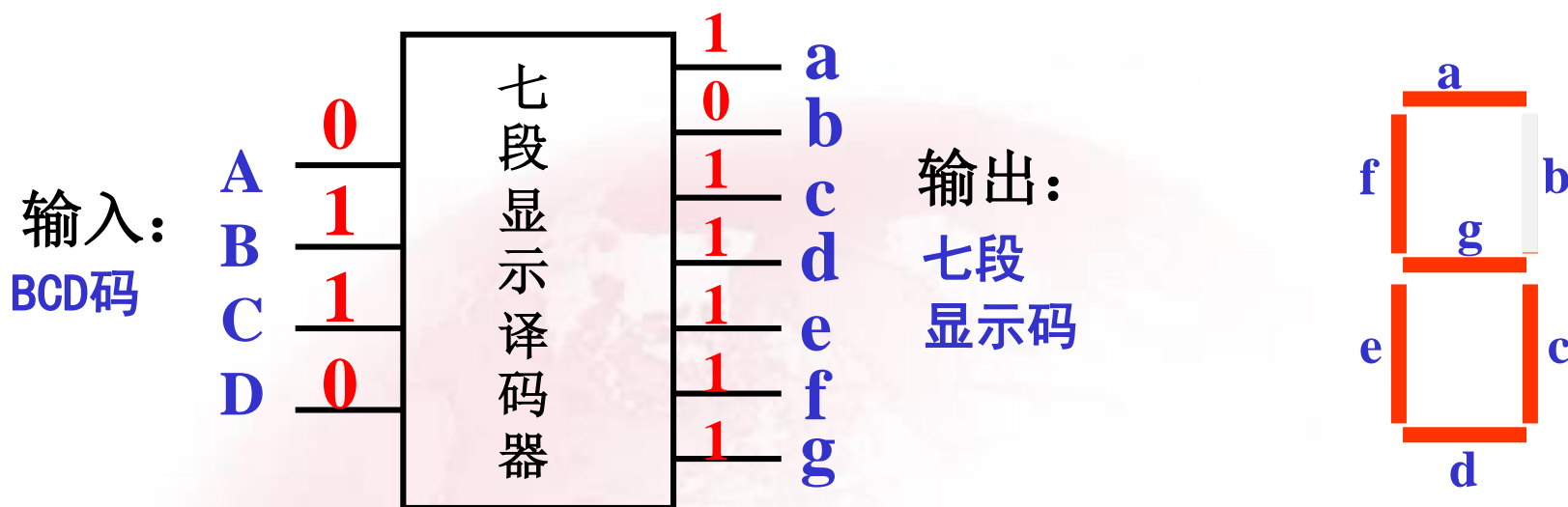
驱动**共阳**显示器：输出**低电平**为有效电平





显示译码器设计

功能：将表示数字的**BCD**码转换成**七段显示码**。



驱动**共阳**显示器：输出**低电平**为有效电平

驱动**共阴**显示器：输出**高电平**为有效电平





BCD码 — 七段显示译码器（驱动共阳显示器）真值表

十进制	输 入				输 出							字形
	<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	
0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0	1	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0	2
3	0	0	1	1	0	0	0	0	1	1	0	3
4	0	1	0	0	1	0	0	1	1	0	0	4
5	0	1	0	1	0	1	0	0	1	0	0	5
6	0	1	1	0	0	1	0	0	0	0	0	6
7	0	1	1	1	0	0	0	1	1	1	1	7
8	1	0	0	0	0	0	0	0	0	0	0	8
9	1	0	0	1	0	0	0	0	1	0	0	9

数字逻辑电路教学课程

化简后表达式:

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

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

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

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

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

$$f = AB + \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}$$

$$g = ABC + \overline{B}\overline{C}\overline{D}$$

BA \ DC	00	01	11	10
00		1		
01	1			
11	x	x	x	x
10			x	x

(a)

BA \ DC	00	01	11	10
00				
01		1		1
11	x	x	x	x
10			x	x

(b)

BA \ DC	00	01	11	10
00				1
01				
11	x	x	x	x
10			x	x

(c)

BA \ DC	00	01	11	10
00		1		
01	1		1	
11	x	x	x	x
10			x	x

(d)

BA \ DC	00	01	11	10
00		1	1	
01	1	1	1	
11	x	x	x	x
10		1	x	x

(e)

BA \ DC	00	01	11	10
00		1	1	1
01			1	
11	x	x	x	x
10			x	x

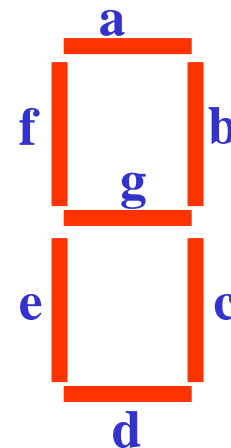
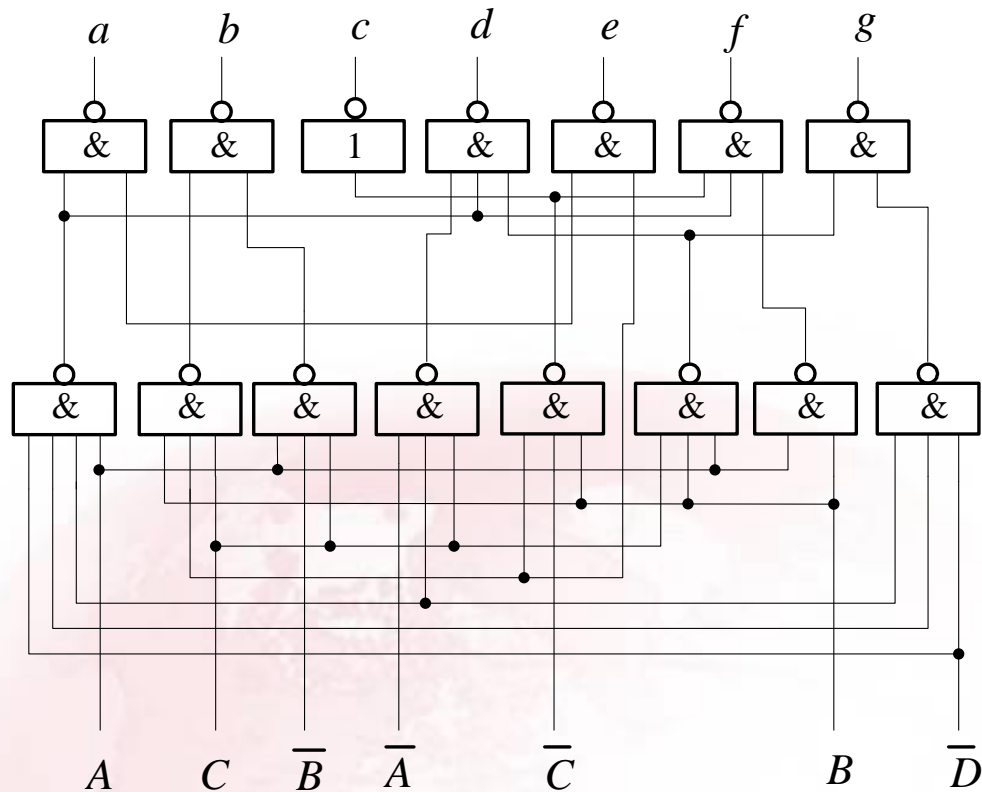
(f)

BA \ DC	00	01	11	10
00	1	1		
01			1	
11	x	x	x	x
10			x	x

(g)

* 多输出逻辑函数化简: 整体最简





七段显示译码器逻辑图

思考题:

根据上面设计, 判断当输入DCBA为1010时, LED显示什么字形?





4. 通用七段显示译码器集成电路

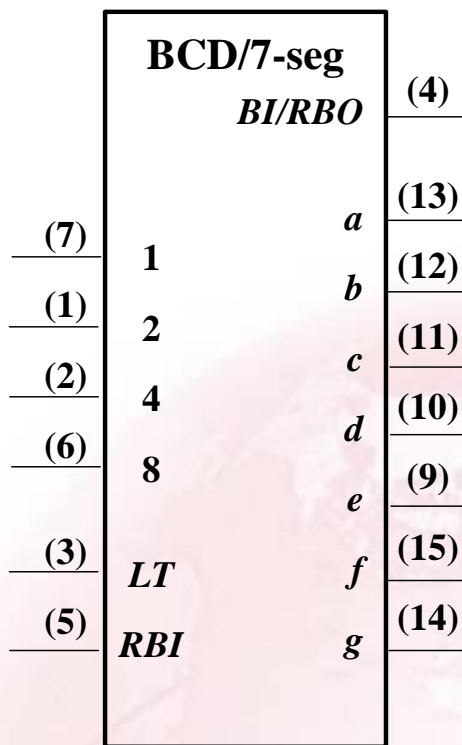
常用的七段显示译码器集成电路有**7446**、**7447**、**7448**、**7449**和**4511**等。下面重点介绍七段显示译码器**7448**。

七段显示译码器**7448**输出高电平有效，用以驱动共阴极显示器。

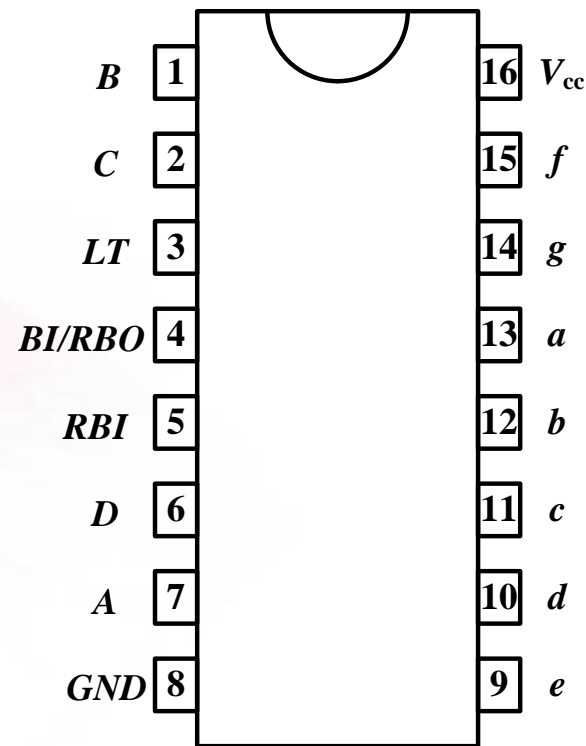




BCD
码输入



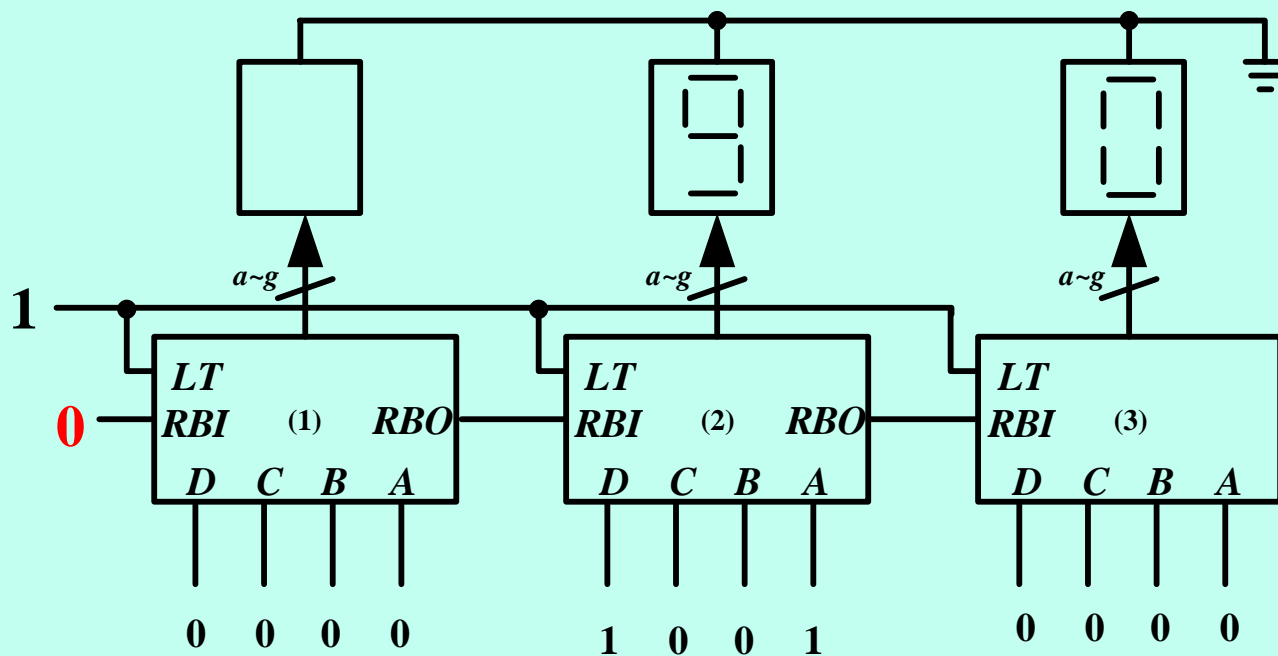
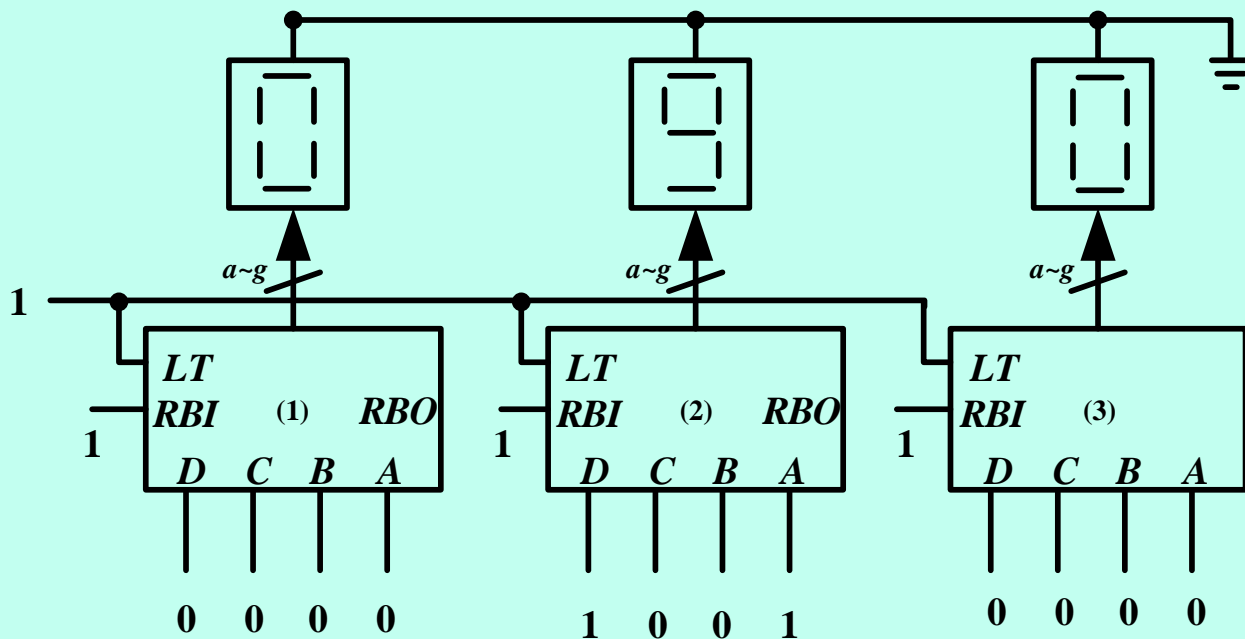
逻辑图



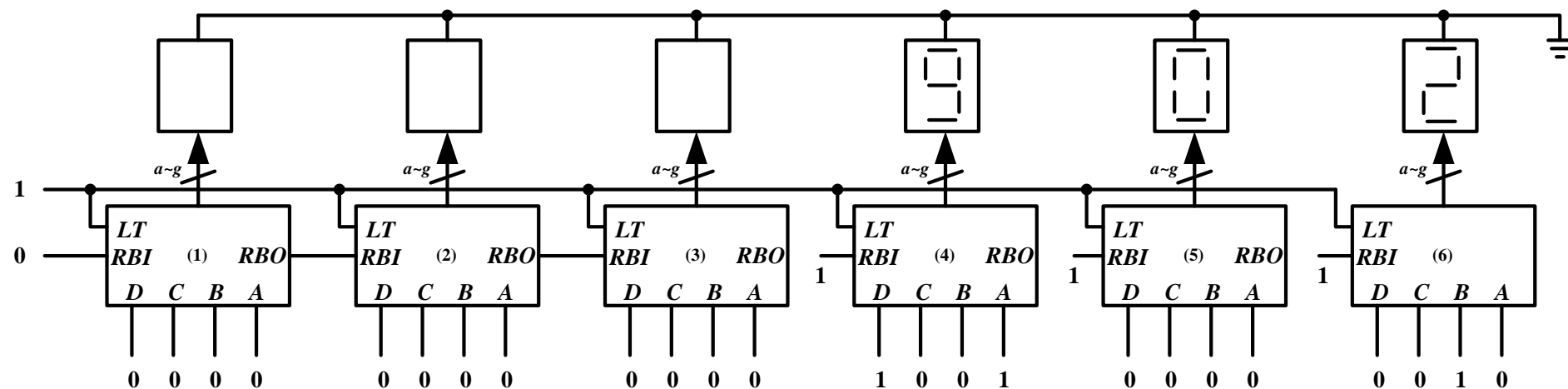
引脚图



多位静态显示电路



7448实现多位显示



由于第1片的 RBI 为0，而 $DCBA=0000$ ，所以满足灭零条件， $RBO=0$ 。第2、3片也满足灭零条件。

第4、5、6片驱动正常显示。

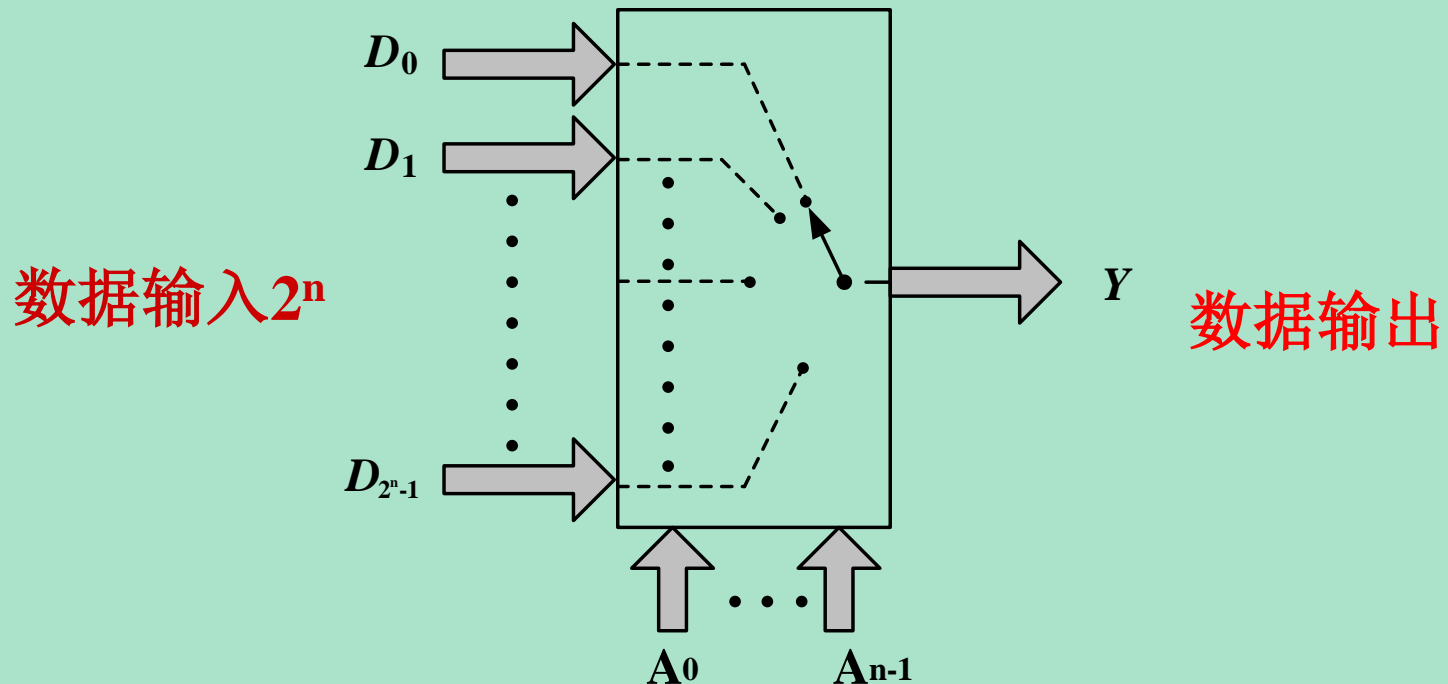
思考题：如第1片输入 $DCBA$ 不等于0000，2、3两片灭零条件吗？

4.4 数据选择器

功能: 从多路输入数据中选择其中的一路送至输出端.

数据选择器简称**MUX**, 数据输入端数称为**通道数**.

数据选择器功能示意图:



选择信号 (地址码) 输入 n



4.4.1 数据选择器的电路结构

以四选一数据选择器为例讨论

功能表

地址码

$A_1 A_0$		Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

输出端

数据
输入端

输出函数表达式:

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

$$Y = \sum_{i=0}^3 m_i D_i$$

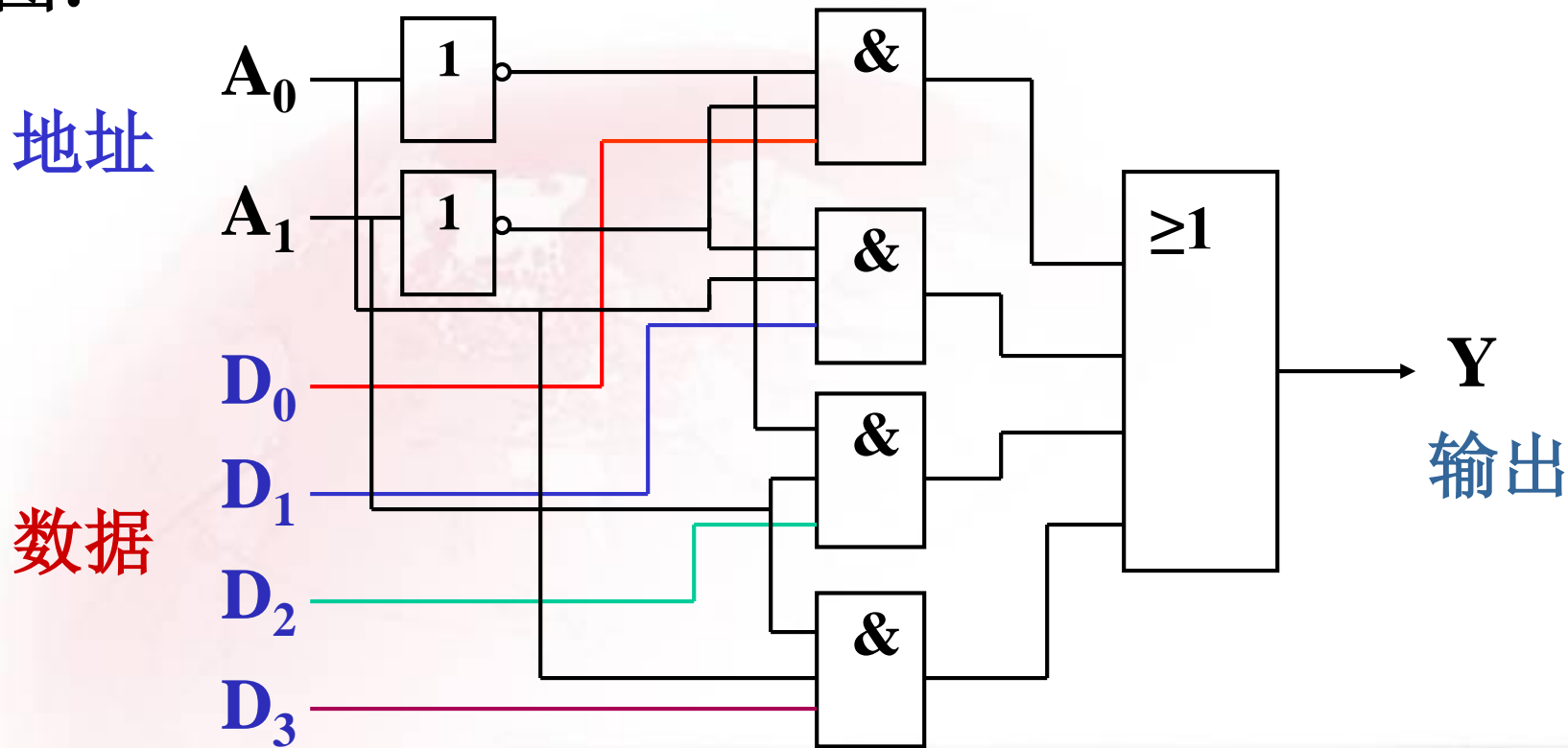


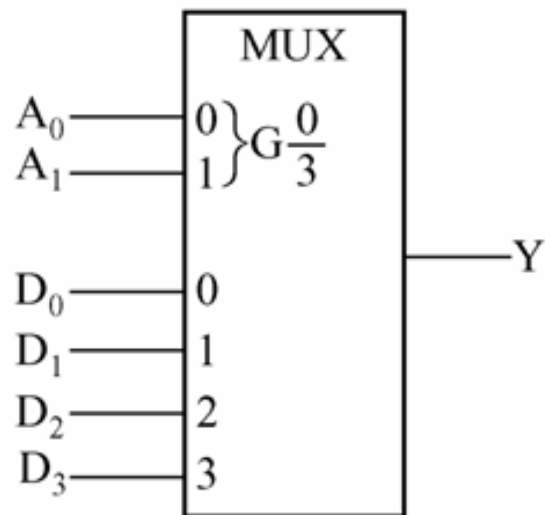


表达式:

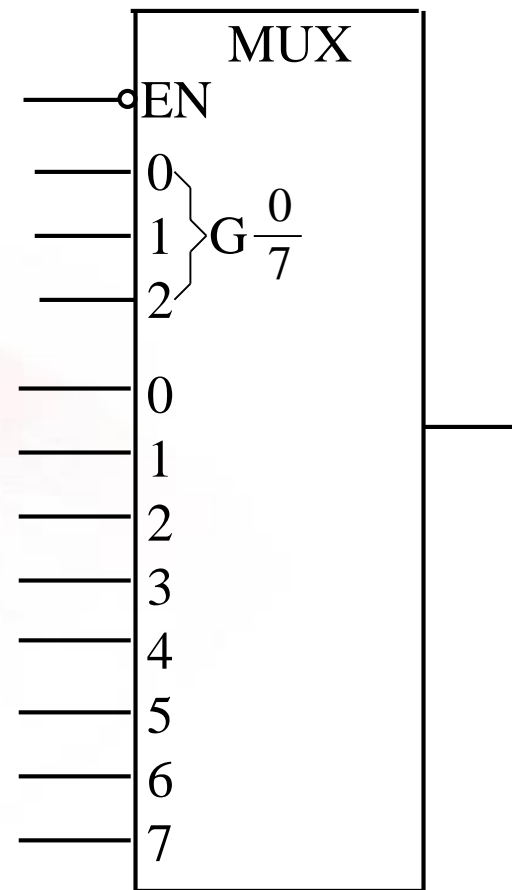
$$Y = (\bar{A}_1 \bar{A}_0) D_0 + (\bar{A}_1 A_0) D_1 + (A_1 \bar{A}_0) D_2 + (A_1 A_0) D_3$$

电路图:





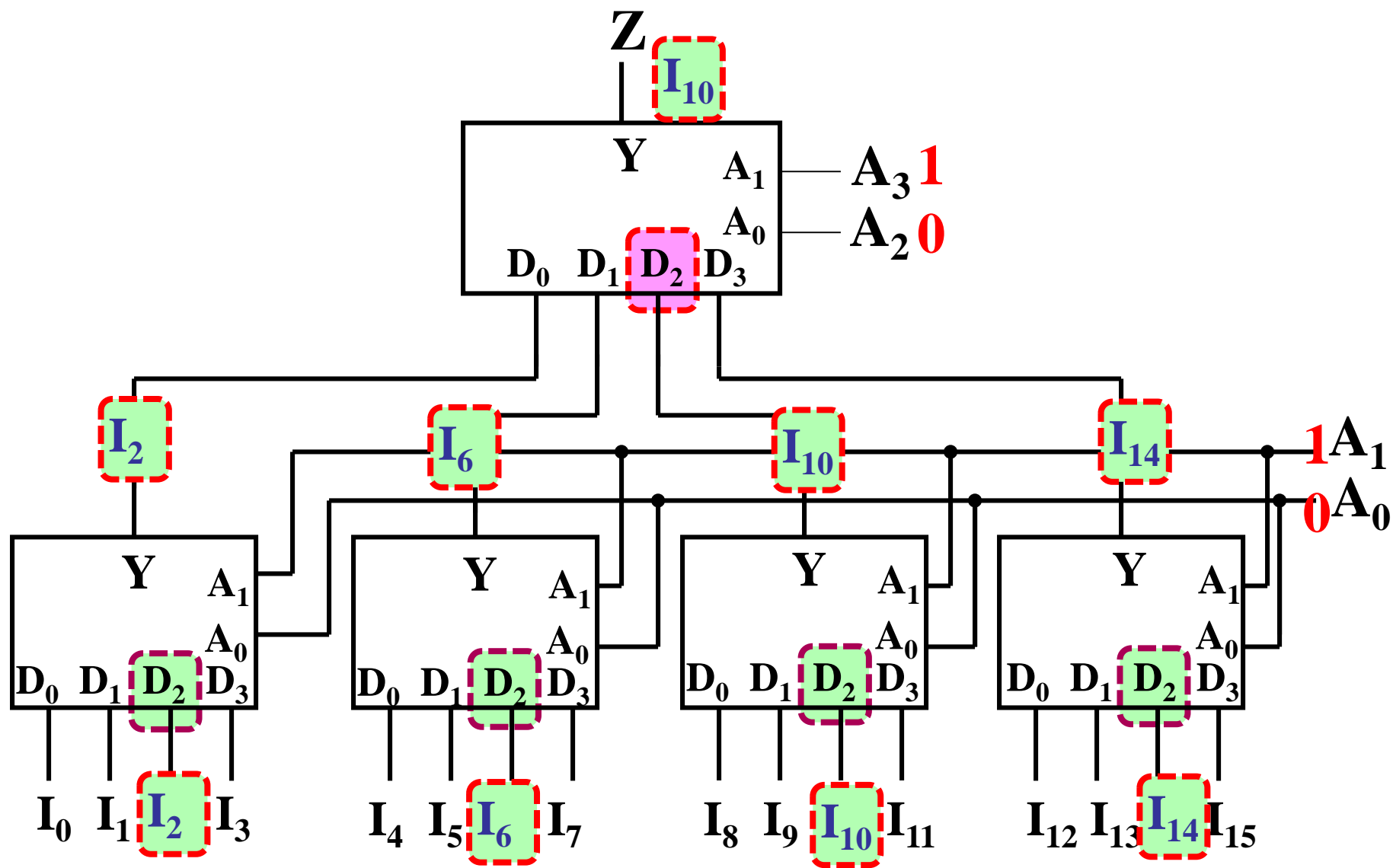
4选1数据选择器的逻辑符号



8选1数据选择器的逻辑符号



数据选择器通道扩展：由四选一数据选择器组成十六选一数据选择器的例子



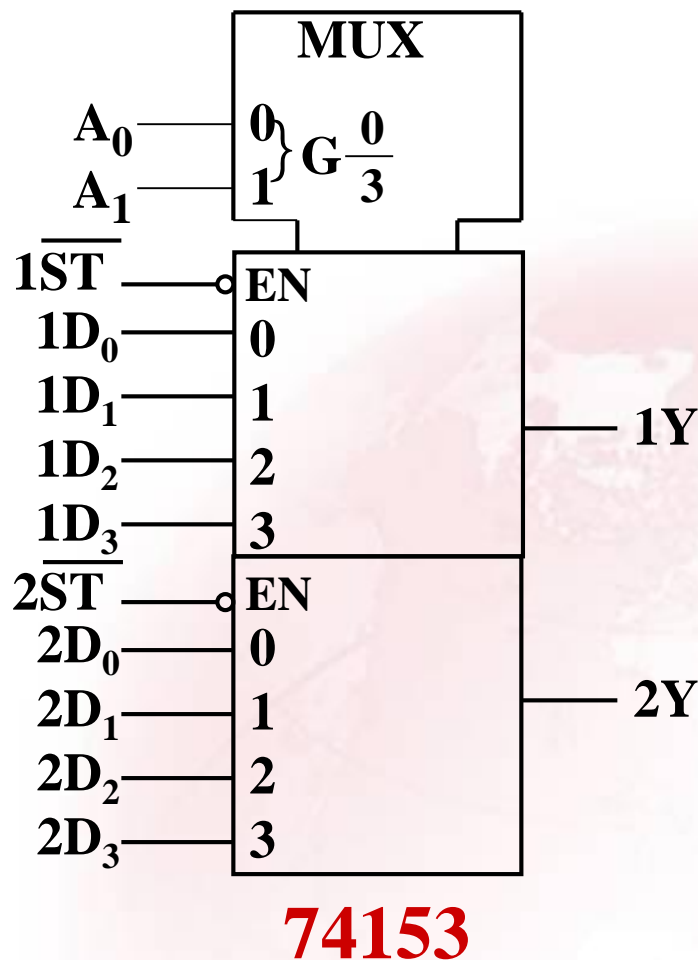


4.4.2 通用数据选择器集成电路

常用MUX集成电路

输入数	TTL	CMOS(数字)	CMOS(模拟)	ECL
16	74150	4515	4067	
2×8	74451		4096	
8	74151	4512	4051	10164
4×4	74453			
2×4	74153	4539	4052	10174
8×2	74604			
4×2	74157	4519		10159



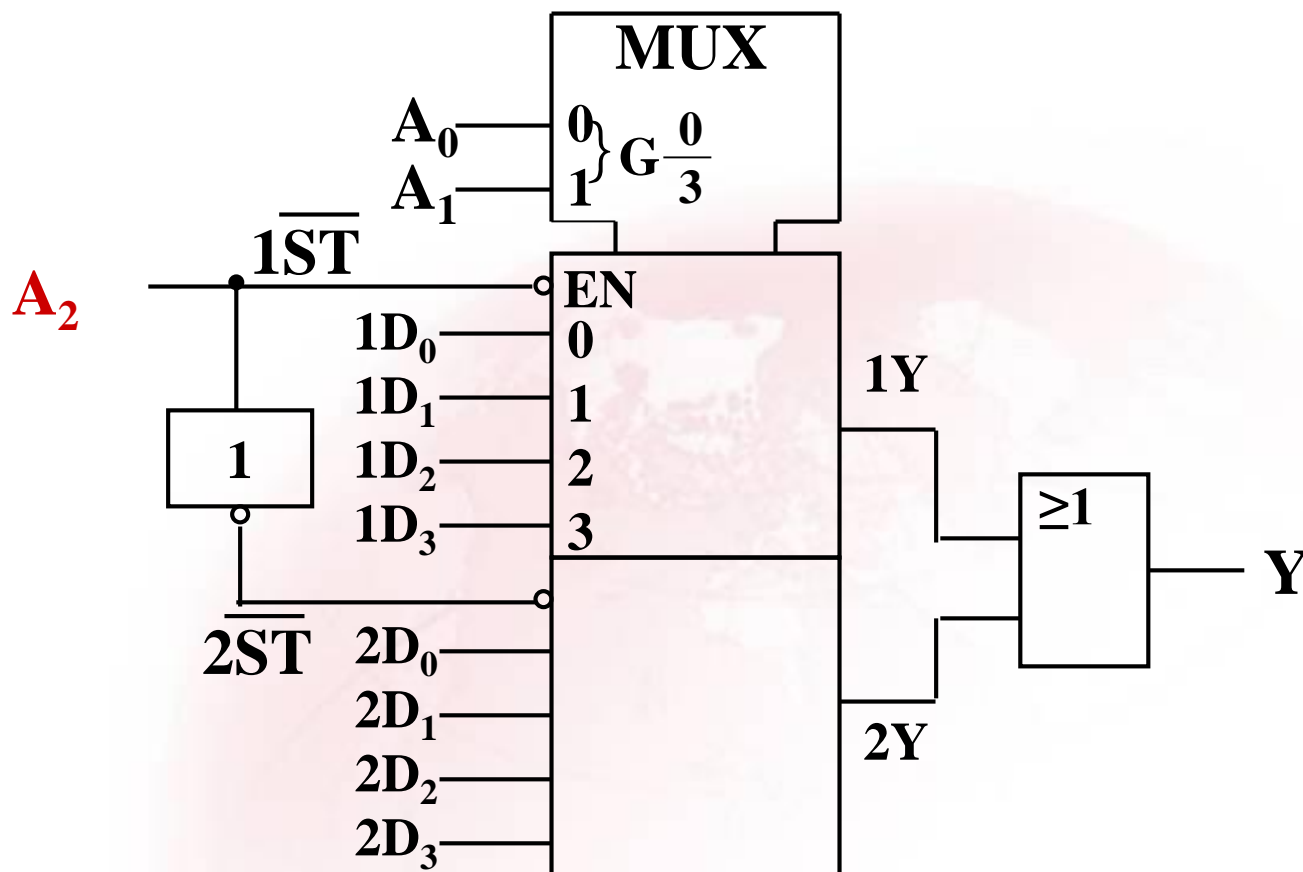


$$Y = (\overline{A_1}\overline{A_0})D_0 + (\overline{A_1}A_0)D_1 + (A_1\overline{A_0})D_2 + (A_1A_0)D_3)ST$$

内部结构由与、或、非等门组成。



利用选通控制端实现通道扩展的例子:



$A_2=0$ 时, 由
 A_1A_0 选择 $1D_i$

$A_2=1$ 时, 由
 A_1A_0 选择 $2D_i$





4.4.3 数据选择器应用举例

1. 用数据选择器实现组合逻辑函数

基本思想:

由数据选择器的一般表达式 $Y = \sum m_i D_i$

可知,利用地址变量产生所有最小项,通过数据输入信号 D_i 的不同取值,来选取组成逻辑函数的所需最小项.

例 试用八选一数据选择器74151实现逻辑函数

$$F(A, B, C) = \sum m(0, 2, 3, 5)$$





解：待实现的函数为： $F(A, B, C) = \sum m(0, 2, 3, 5)$
 $= \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C$

74151的输出表达式为： $Y = \sum_{i=0}^7 m_i D_i$

$$Y = (\overline{A}_2\overline{A}_1\overline{A}_0\mathbf{D_0} + \overline{A}_2\overline{A}_1A_0\mathbf{D_1} + \overline{A}_2A_1\overline{A}_0\mathbf{D_2} + \overline{A}_2A_1A_0\mathbf{D_3} \\ + A_2\overline{A}_1\overline{A}_0\mathbf{D_4} + A_2\overline{A}_1A_0\mathbf{D_5} + A_2A_1\overline{A}_0\mathbf{D_6} + A_2A_1A_0\mathbf{D_7})ST$$

比较两式：

令： $\overline{ST}=0$ $\mathbf{A_2=A ; A_1=B ; A_0=C}$

$\mathbf{D_0=D_2=D_3=D_5=1}$ $\mathbf{D_1=D_4=D_6=D_7=0}$

$\mathbf{Y=F}$





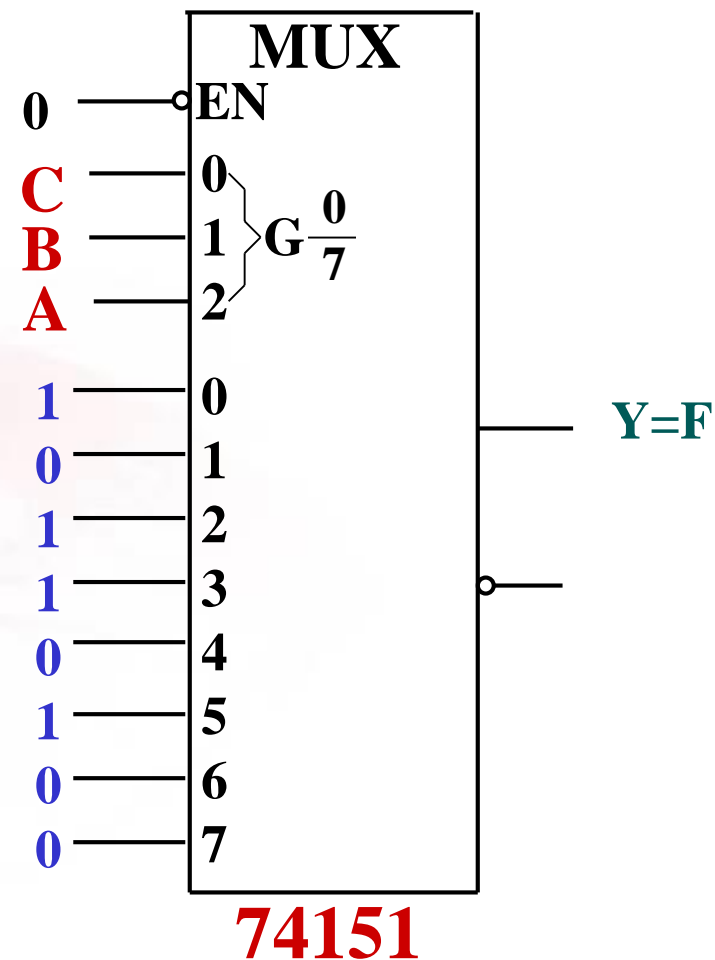
$$\overline{ST}=0$$

$$A_2=A ; A_1=B ; A_0=C$$

$$D_0=D_2=D_3=D_5=1$$

$$D_1=D_4=D_6=D_7=0$$

$$Y=F$$



例：试用4选1MUX实现逻辑函数

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

解：当MUX被选通时，其输出逻辑表达式为：

$$Y = (\overline{A_1}\overline{A_0})\mathbf{D_0} + (\overline{A_1}A_0)\mathbf{D_1} + (A_1\overline{A_0})\mathbf{D_2} + (A_1A_0)\mathbf{D_3}$$

将函数F写成： $F = \overline{A}\overline{B} \cdot \mathbf{1} + \overline{A}\overline{B} \cdot \mathbf{0} + A\overline{B} \cdot \overline{\mathbf{C}} + AB \cdot \mathbf{C}$

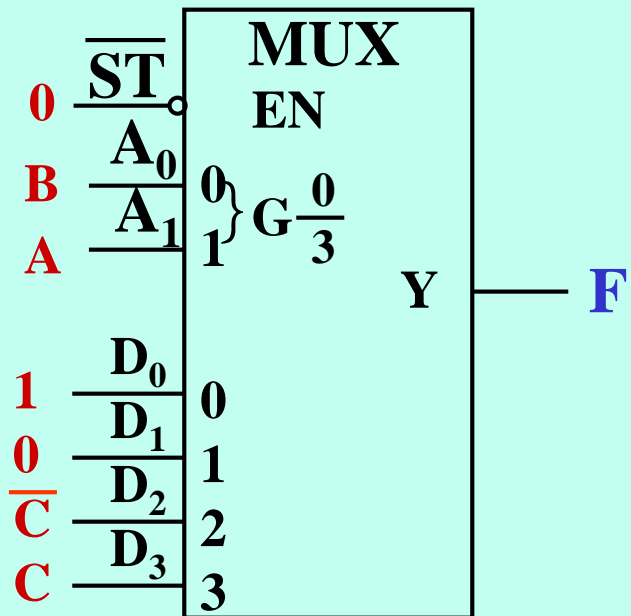
比较两式，令

$$A_1 = A; \quad A_0 = B;$$

$$\mathbf{D_0 = 1}, \quad \mathbf{D_1 = 0}, \quad \mathbf{D_2 = \overline{C}}, \quad \mathbf{D_3 = C}$$

则 $\mathbf{Y = F}$

注：该题的解法不唯一。



例：用四选一数据选择器实现逻辑函数：

$$F(A,B,C,D)=\Sigma m(1,2,4,9,10,11,12,14,15)$$

解： 令数据选择器的地址 $A_1A_0=AB$

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

$$\bar{A}\bar{B}(\bar{C}D + C\bar{D}) = \bar{A}_1\bar{A}_0D_0$$

$$\bar{A}B(\bar{C}\bar{D}) = \bar{A}_1A_0D_1$$

$$AB(C + \bar{D}) = A_1A_0D_3$$

$$A\bar{B}(C + D) = A_1\bar{A}_0D_2$$

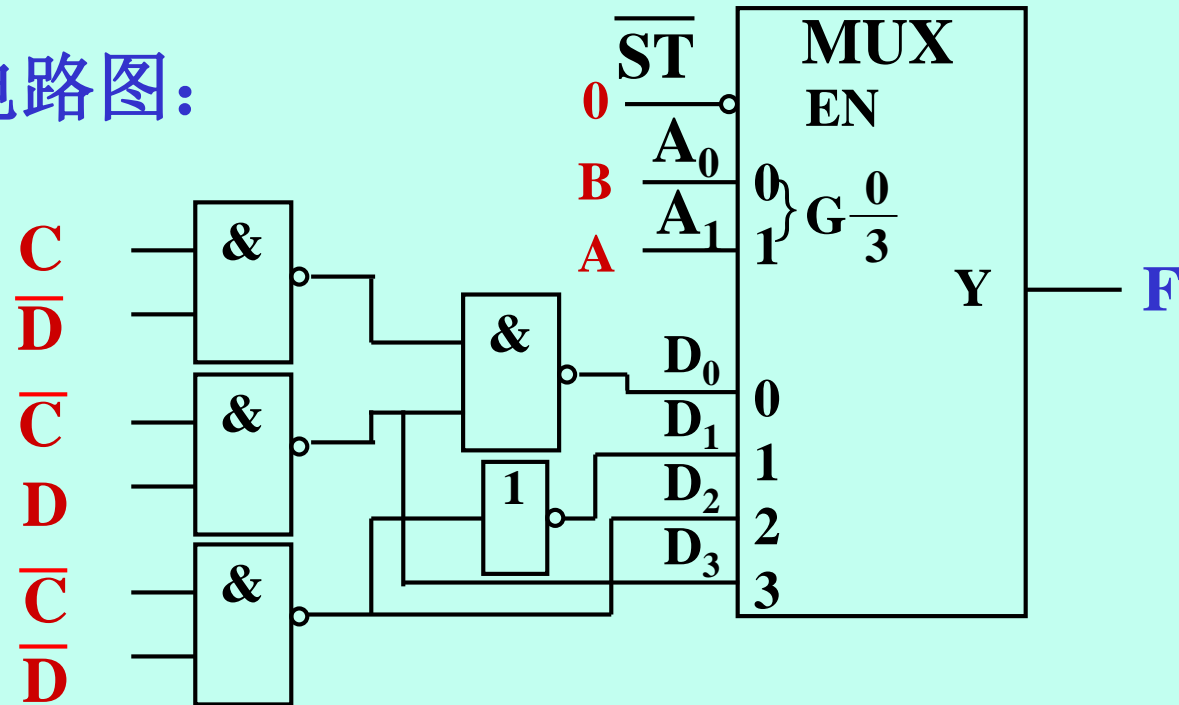
$$D_0 = \bar{C}D + C\bar{D} = \overline{\bar{C}D \cdot C\bar{D}}$$

$$D_1 = \bar{C}\bar{D} = \overline{\overline{\bar{C}\bar{D}}}$$

$$D_2 = C + D = \overline{\bar{C}\bar{D}}$$

$$D_3 = C + \bar{D} = \overline{\bar{C}D}$$

电路图：



注：上面采用A、B作为地址变量。实际上，地址变量的选取是任意的，选不同的变量为地址变量时，**数据输入端**的信号也要随之变化。

例：用四选一数据选择器实现逻辑函数：

$$F(A,B,C,D)=\sum m(1,2,4,9,10,11,12,14,15)$$

解： 令数据选择器的地址 $A_1A_0=CD$

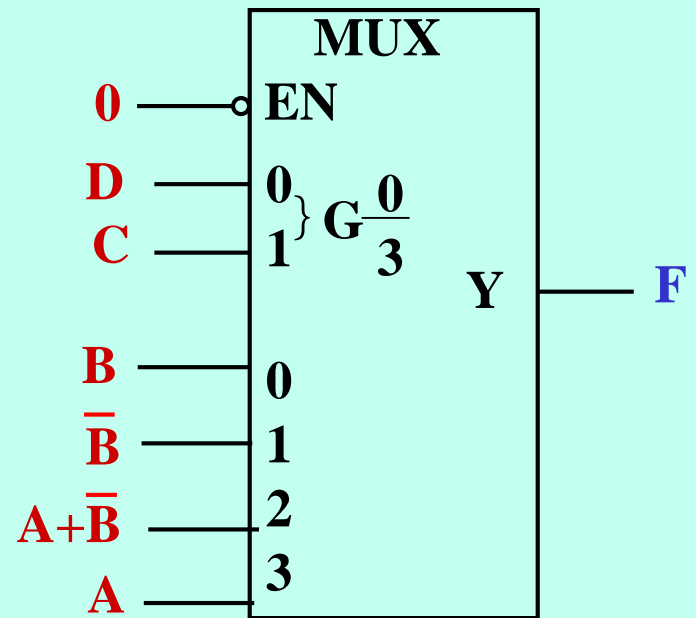
		CD			
		00	01	11	10
AB	00		1		1
	01	1			
	11	1		1	1
	10		1	1	1

$$D_0=B$$

$$D_2=A+\bar{B}$$

$$D_1=\bar{B}$$

$$D_3=A$$





4.4.3 数据选择器应用举例

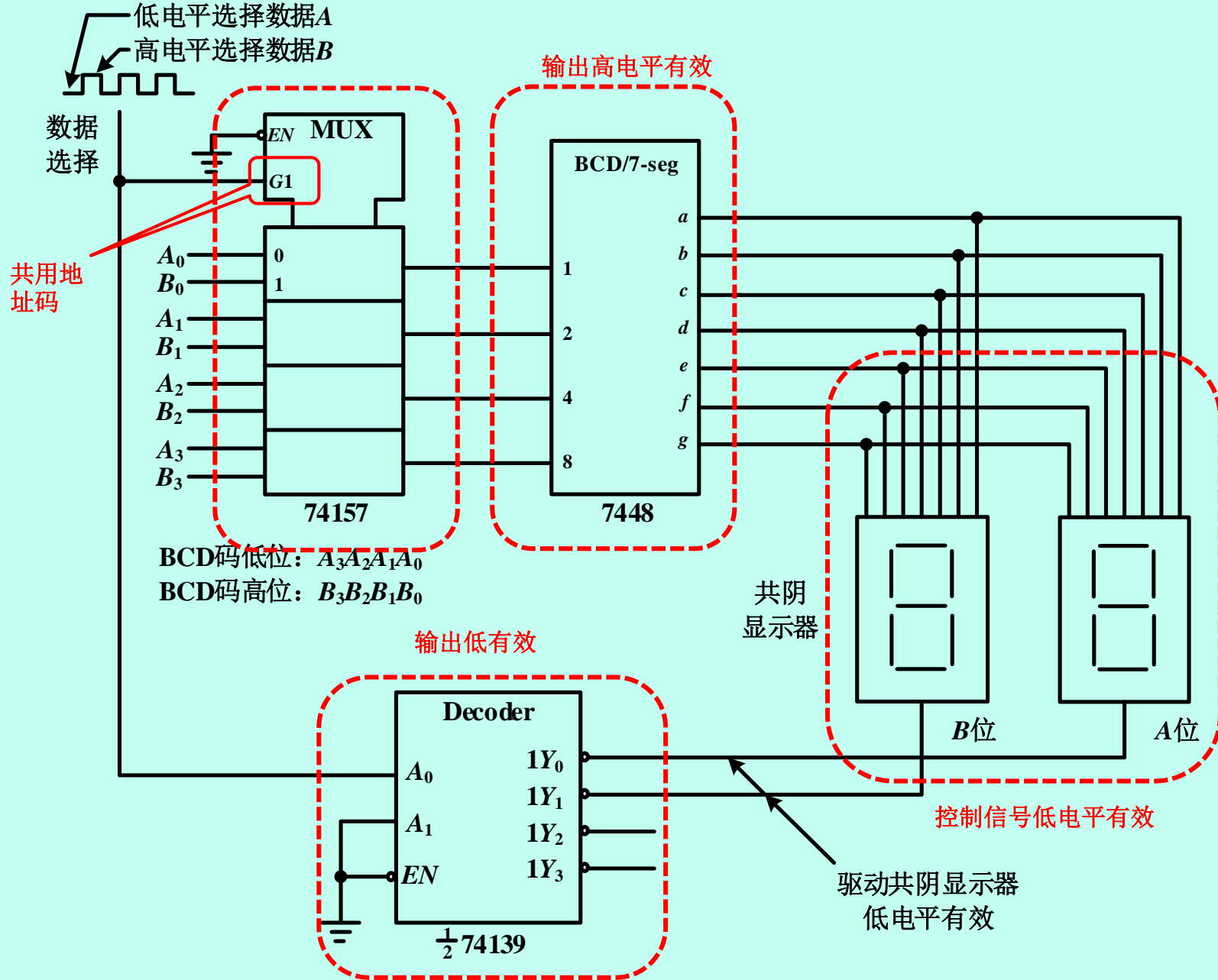
2. 动态显示电路

用于驱动七段数码管的显示电路分为两种，一种称为静态显示，另一种称为动态显示。

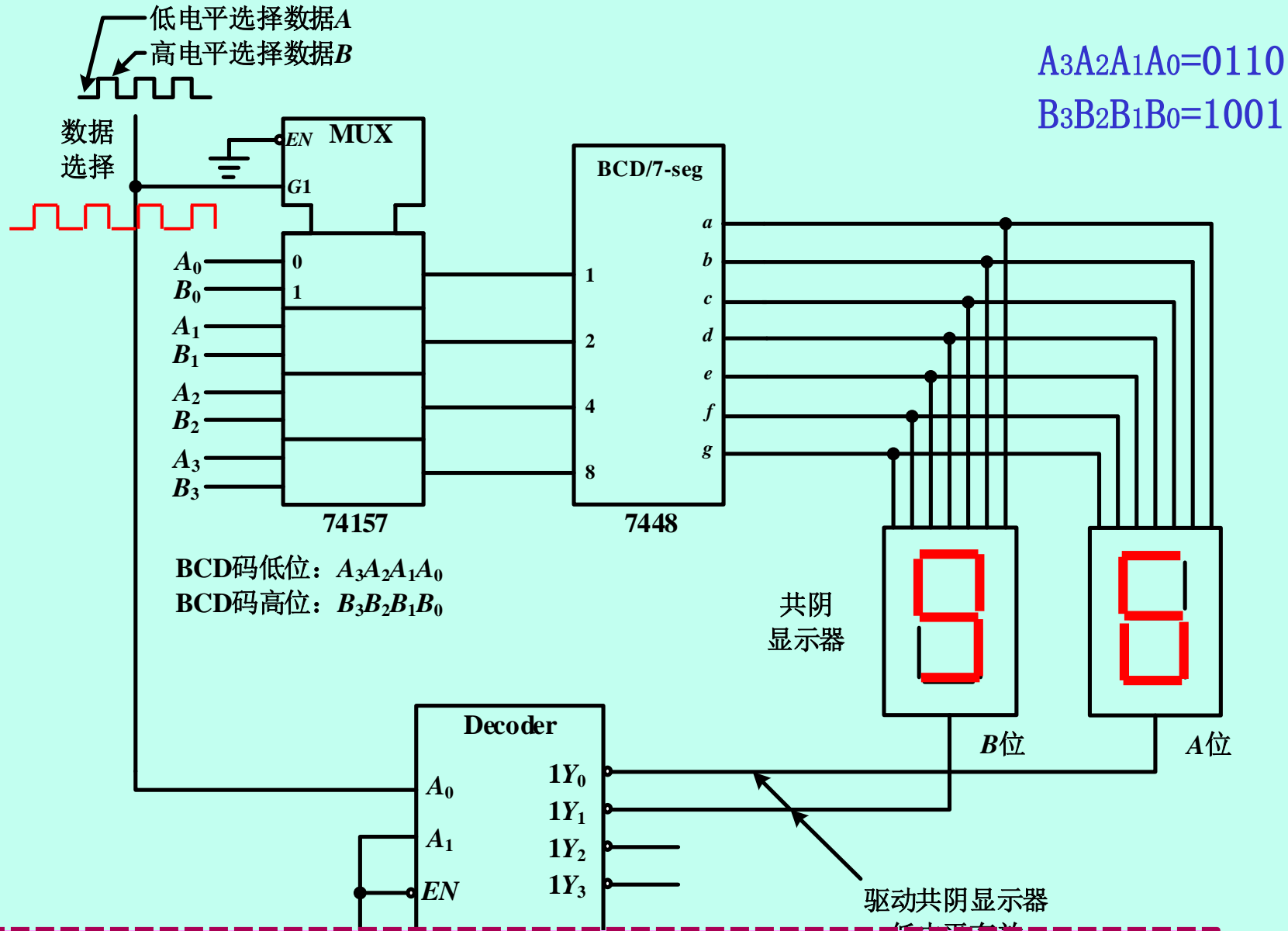
静态显示：每一个七段数码管显示器由单独的显示译码器驱动。

动态显示：利用数据选择器的分时复用功能，由一个显示译码器驱动任意多个七段数码管显示器。





动态显示电路原理图



动态扫描频率太低：数据闪烁（单管扫描不低于30Hz）

动态扫描频率太高：数据传输不稳定

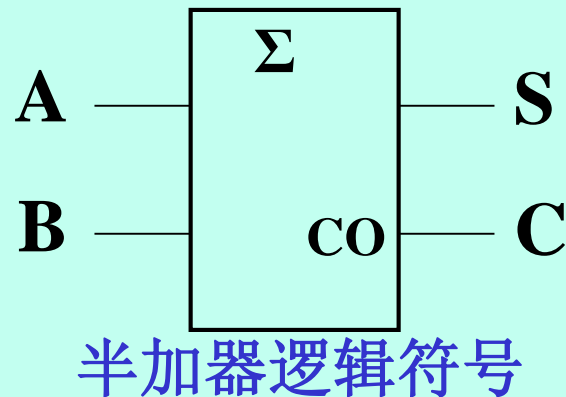
4.5 算术运算电路

算术运算电路的核心为**加法器**。

4.5.1 基本加法器

1. 半加器(HA)

仅考虑两个一位二进制数相加，而不考虑低位的进位，称为**半加**。



设: **A**、**B**为两个**加数**，**S**为**本位**的**和**，**C**为**本位**向高位的**进位**。

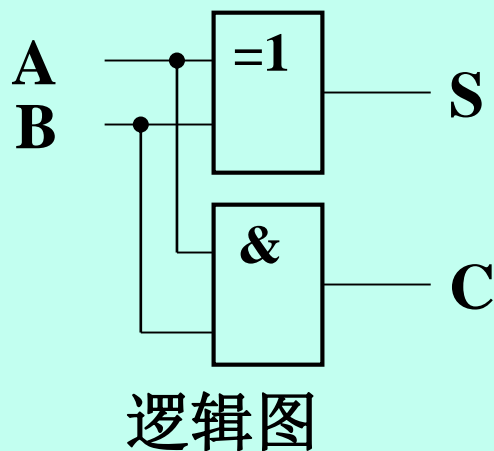
真值表

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

逻辑方程

$$S = A \oplus B$$

$$C = AB$$





2. 全加器

在多位数相加时,除考虑本位的两个加数外,还须考虑低位向本位的进位.

例:

1 1 0 1	加数
1 1 1 1	加数
+ 1 1 1 1	低位向高位的进位
1 1 1 0 0	和

实际参加一位数相加,必须有三个量,它们是:

本位加数 A_i 、 B_i ; 低位向本位的进位 C_{i-1}

一位全加器的输出结果为:

本位和 S_i ; 本位向高位的进位 C_i



全加器电路设计:

由两个半加器实现一个全加器

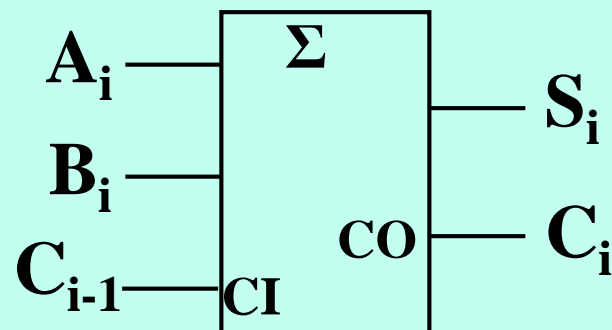
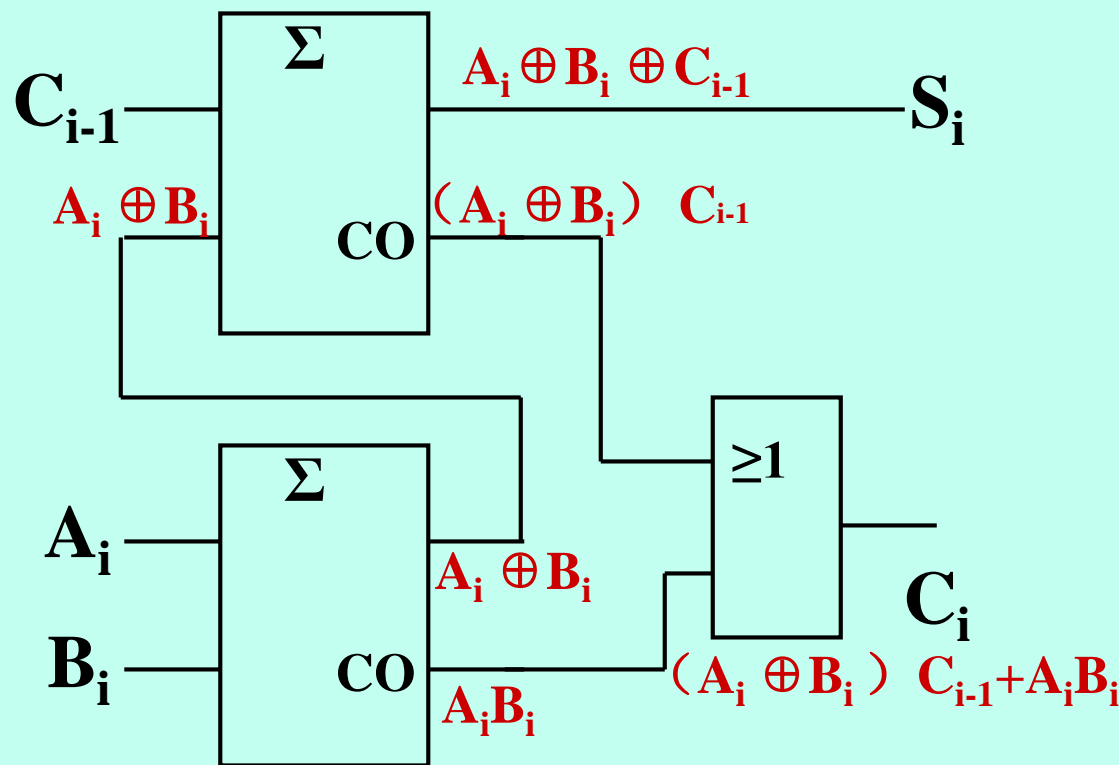
A_i	B_i	C_{i-1}	C_i	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S_i = (\bar{A}_i \bar{B}_i + A_i B_i) C_{i-1} + (\bar{A}_i B_i + A_i \bar{B}_i) \bar{C}_{i-1}$$

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

$$C_i = (\bar{A}_i B_i + A_i \bar{B}_i) C_{i-1} + A_i B_i$$

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

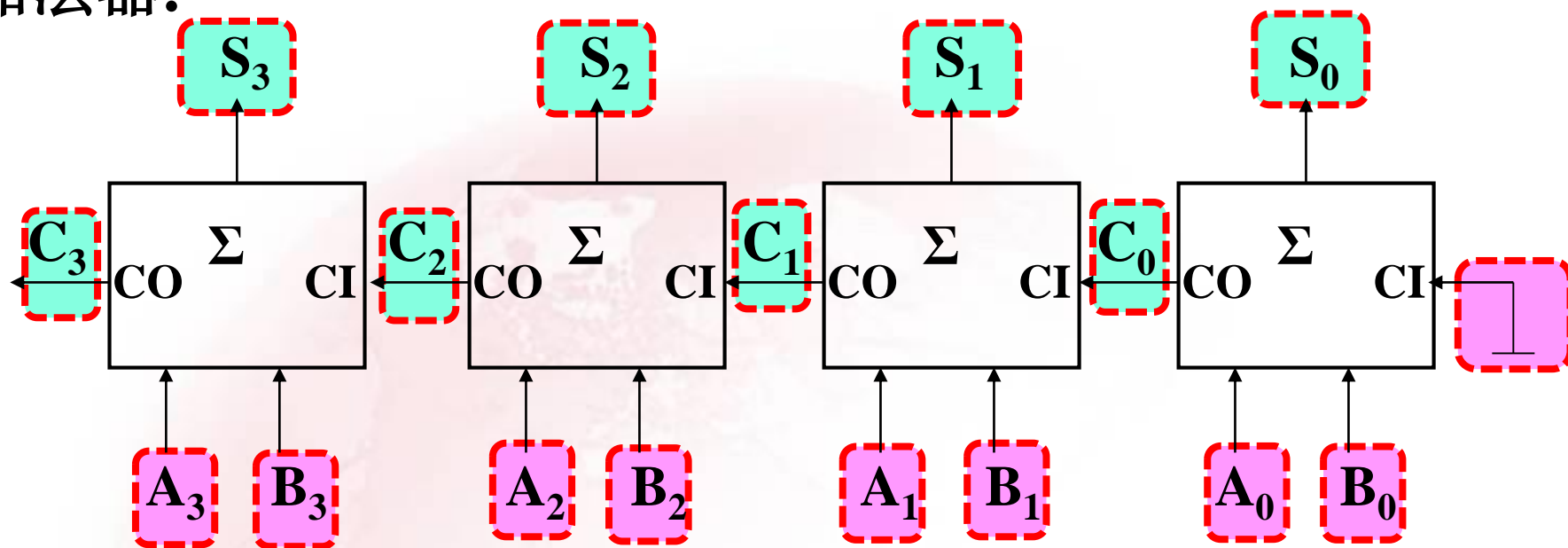


全加器逻辑符号



3. 串行进位加法器

当有多位数相加时,可模仿**笔算**,用**全加器**构成串行进位加法器.



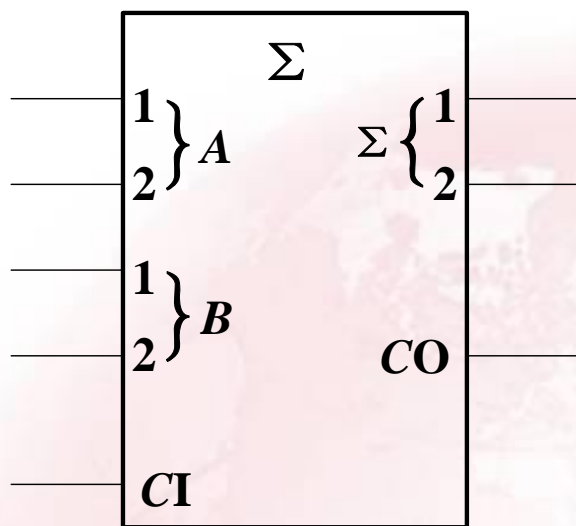
四位串行进位加法器

串行进位加法器特点: **结构简单**; **运算速度慢**.

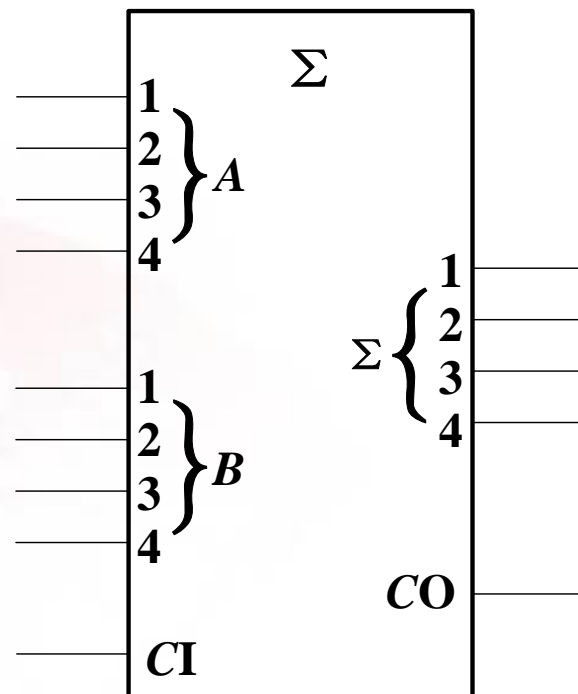




4.5.3 通用加法器集成电路



2位并行加法器7482



4位并行加法器7483



3. 加法器的应用举例

(1) 将8421BCD码转换为余3BCD码的代码转换电路。

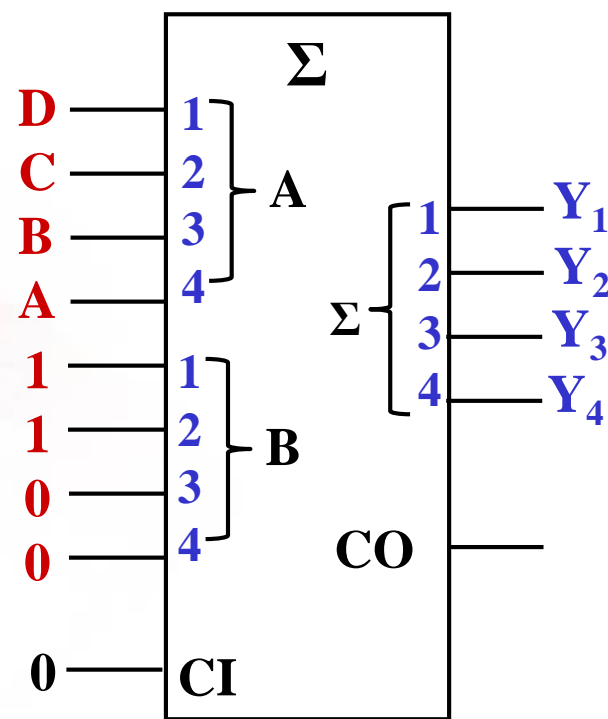
解：设输入8421BCD码为
ABCD（A为高位）

输出余3码为

$Y_4Y_3Y_2Y_1$ （ Y_4 为高位）

问题：如何将余3BCD码转换为8421BCD码。

在二进制补码系统中，减法功能由加“减数”的补码实现。



7483





(2) 四位二进制加/减器

两个运算数分别为:

$$P = P_4P_3P_2P_1 \quad Q = Q_4Q_3Q_2Q_1$$

控制信号为: S

S	功能
0	$(P)_2 + (Q)_2$
1	$(P)_2 - (Q)_2$

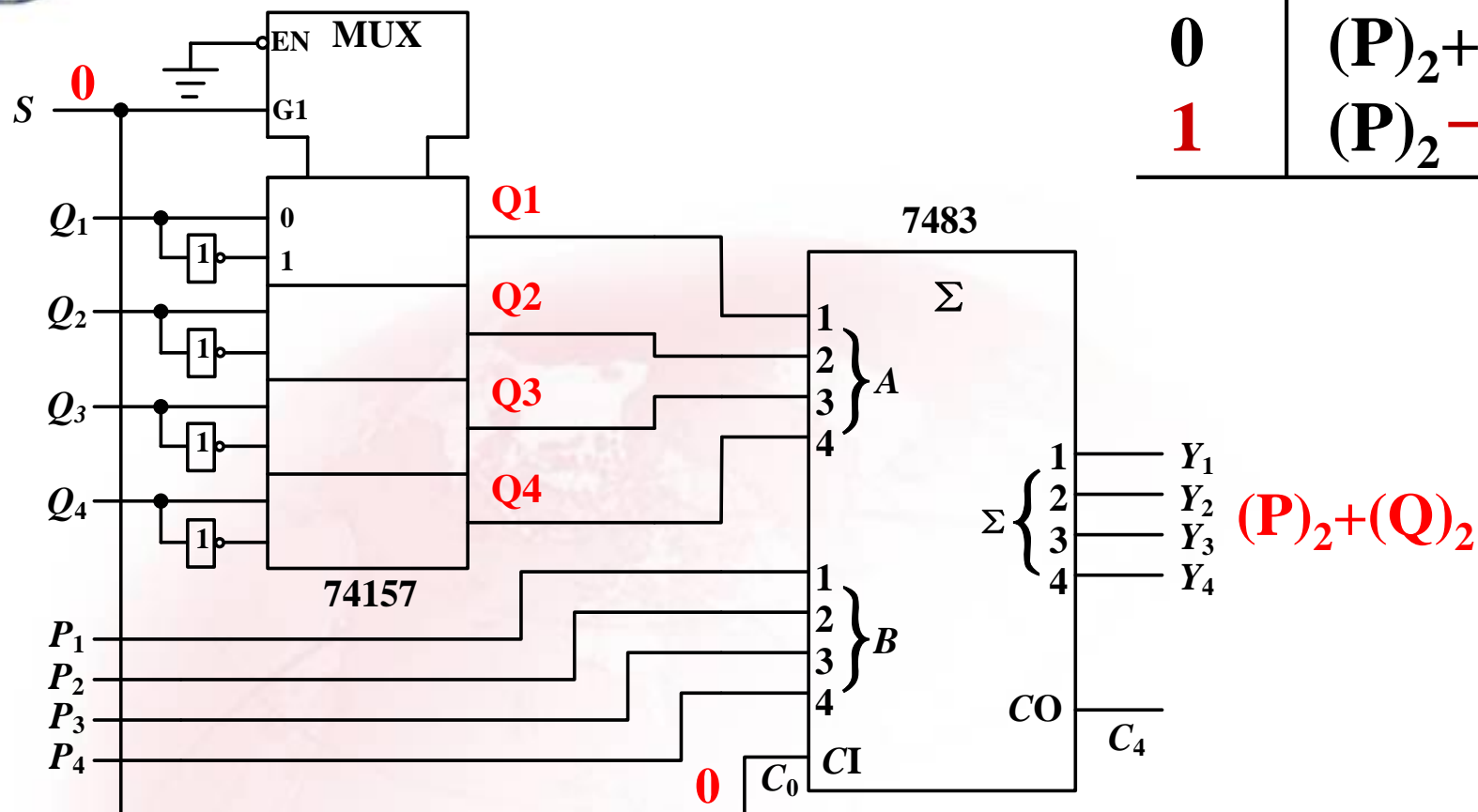
用 4×2 选1数据选择器74157和4位全加器7483, 构成4位二进制加/减器。





数字逻辑电路教学课程

S	功能
0	$(P)_2 + (Q)_2$
1	$(P)_2 - (Q)_2$

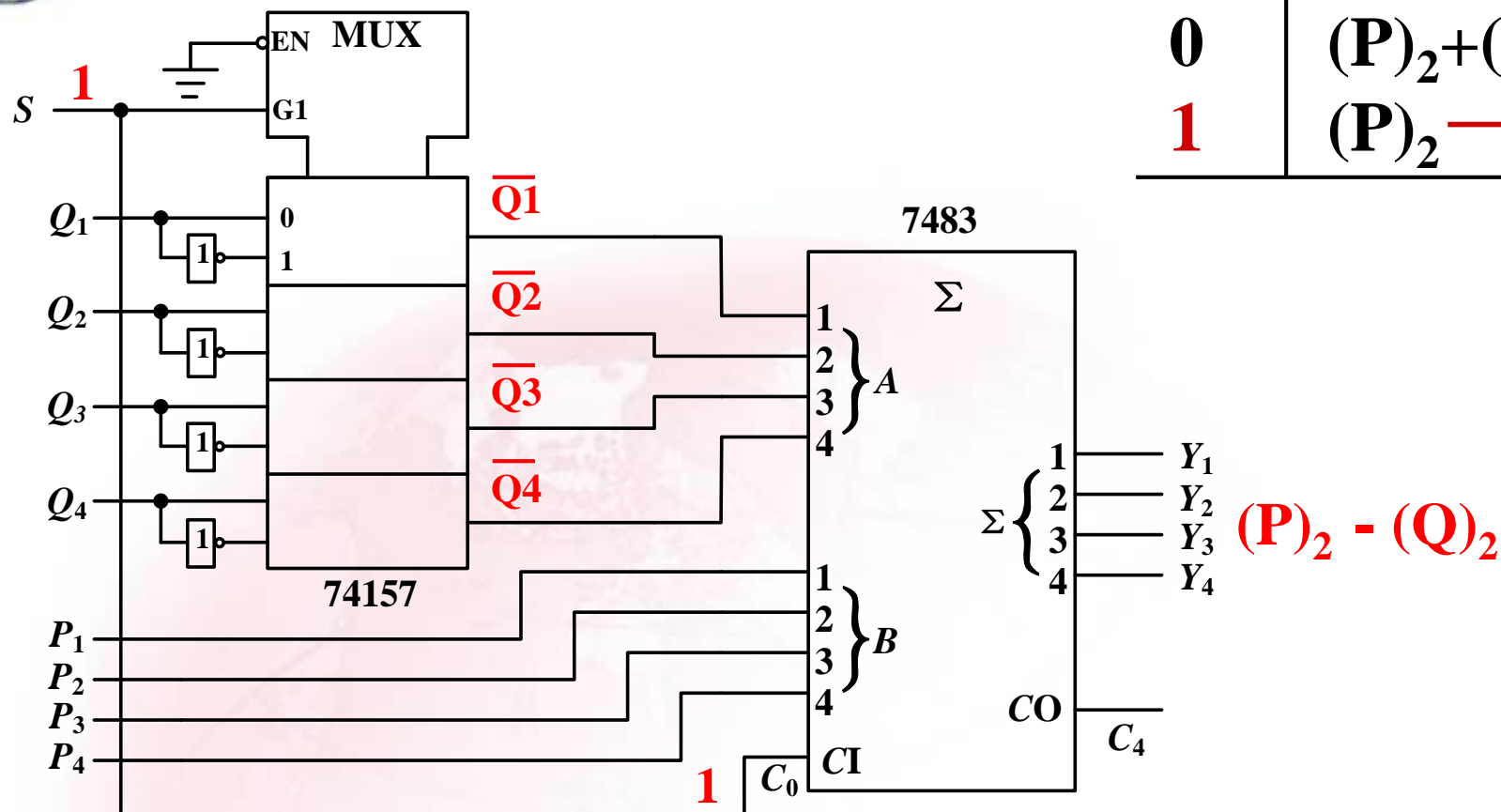


二进制加/减器





S	功能
0	$(P)_2 + (Q)_2$
1	$(P)_2 - (Q)_2$



二进制加/减器



2. 利用**7483**(四位二进制加法器)构成**8421BCD**码加法器.

BCD码加法器示例:

十进制加法: $2+3=5$

二进制加法: $0010+0011=0101$

8421BCD码加法: $0010+0011=0101$

十进制加法: $7+5=12$

二进制加法: $0111+0101=1100$

8421BCD码加法: $0111+0101=0001\ 0010$
 $=10010$

十进制加法: $8+9=17$

二进制加法: $1000+1001=10001$

8421BCD码加法: $1000+1001=0001\ 0111$
 $=10111$

二进制数和8421BCD码对照表

二进制数和: C_4S ($S = S_4S_3S_2S_1$)
 8421码和: K_4B ($B = B_8B_4B_2B_1$)

十进制数	二进制数(和)					8421BCD码(和)				
	C_4	S_4	S_3	S_2	S_1	K_4	B_8	B_4	B_2	B_1
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

$$K_4 = C_4 = 0$$

$$B = S$$

$$K_4 = \overline{C_4} = 1$$

$$B = S + 0110 \text{ 有溢出}$$

$$K_4 = C_4 = 1$$

$$B = S + 0110 \text{ 无溢出}$$

总结上表,可得:

① $K_4 = 1$ 时,需进行加6 (**0110**) 校正;

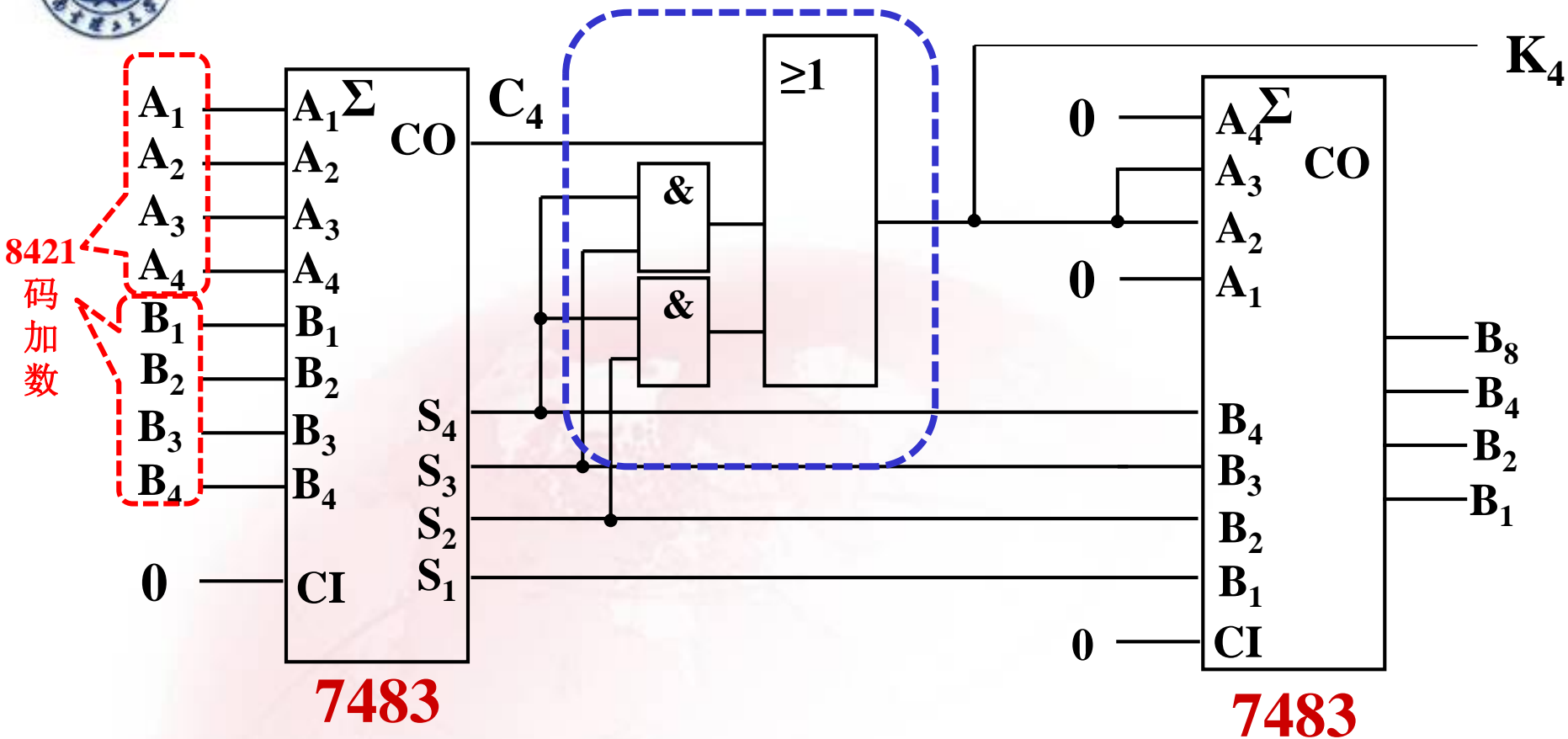
② $K_4 = 1$ 有三种情况:

a. $C_4 = 1$ (对应十进制数**16,17,18,19**)

b. $S_4 = S_3 = 1$ (对应十进制数**12,13,14,15**)

c. $S_4 = S_2 = 1$ (对应十进制数**10,11,14,15**)

$$\text{所以: } K_4 = C_4 + S_4S_3 + S_4S_2$$



8421码加法器





4.6 数值比较器

数值比较器用来判断两个二进制数的**大小或相等**。

4.6.1 一位数值比较器

真 值 表

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

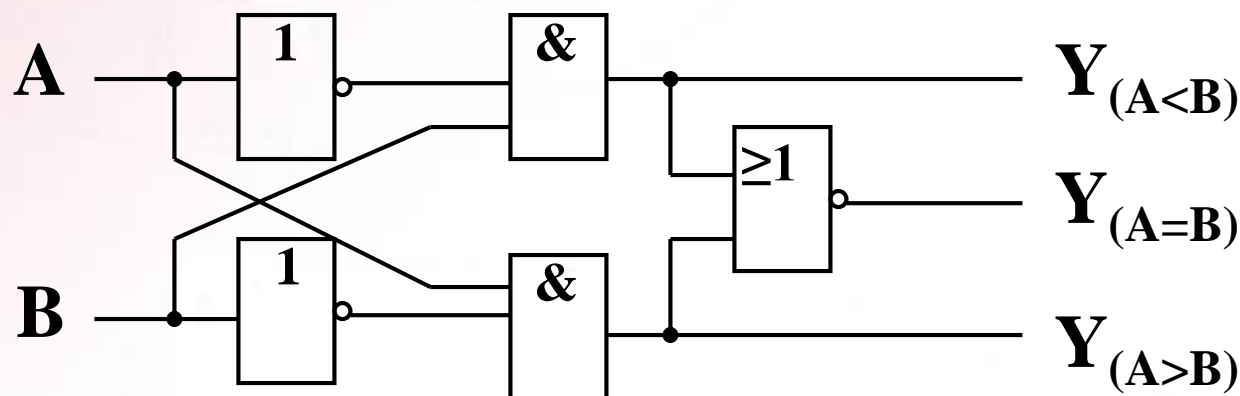
表达式:

$$Y_{(A>B)} = A\bar{B}$$

$$Y_{(A<B)} = \bar{A}B$$

$$Y_{(A=B)} = A \odot B$$

逻辑图





4.6.2 多位数值比较器

比较两个多位数,应首先从高位开始,逐位比较。

例如: $A=A_3A_2A_1A_0$ $B=B_3B_2B_1B_0$

$$Y_{(A>B)} =$$

$$Y_{(A<B)} = \bar{A}_3B_3 + (A_3 \odot B_3) \bar{A}_2B_2 + (A_3 \odot B_3) (A_2 \odot B_2) \bar{A}_1B_1 + (A_3 \odot B_3) (A_2 \odot B_2) (A_1 \odot B_1) \bar{A}_0B_0$$

$$Y_{(A=B)} = (A_3 \odot B_3) (A_2 \odot B_2) (A_1 \odot B_1) (A_0 \odot B_0)$$

四位数值比较器逻辑表达式

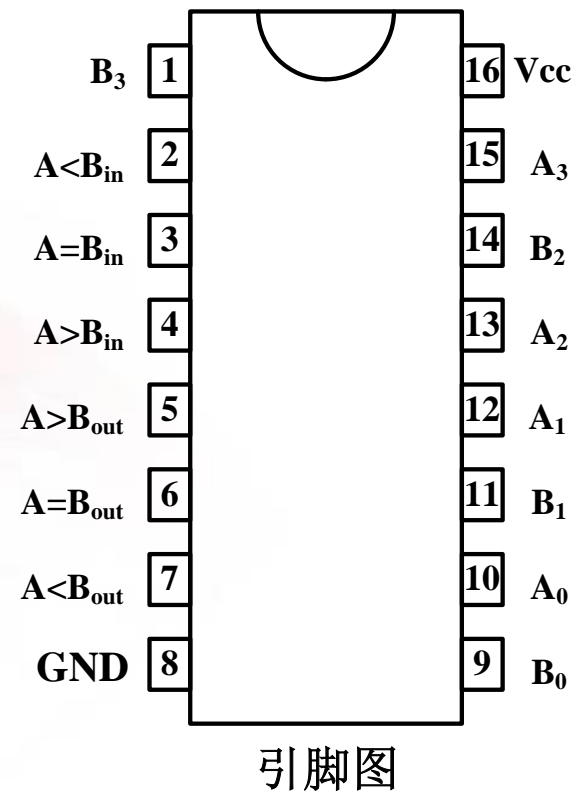
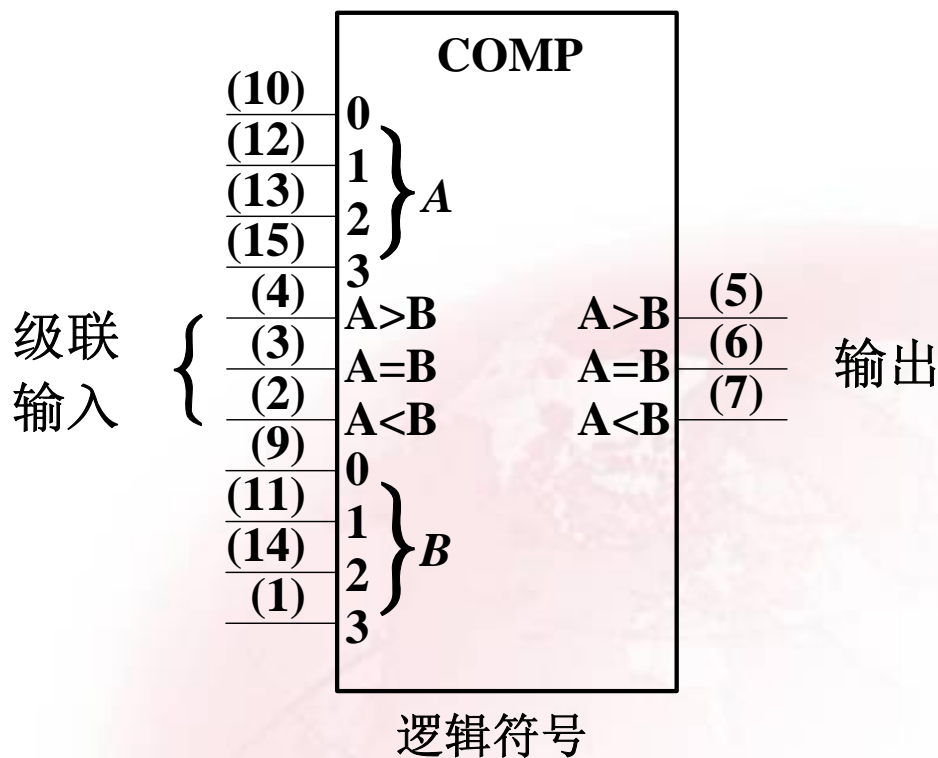


4.6.3 通用数值比较器集成电路

通用数值比较器集成电路有多个品种，属CMOS电路的4位数值比较器的有74HC85(对应的TTL电路为74LS85)、CC14585等。

74HC85为带级联输入的4位数值比较器。

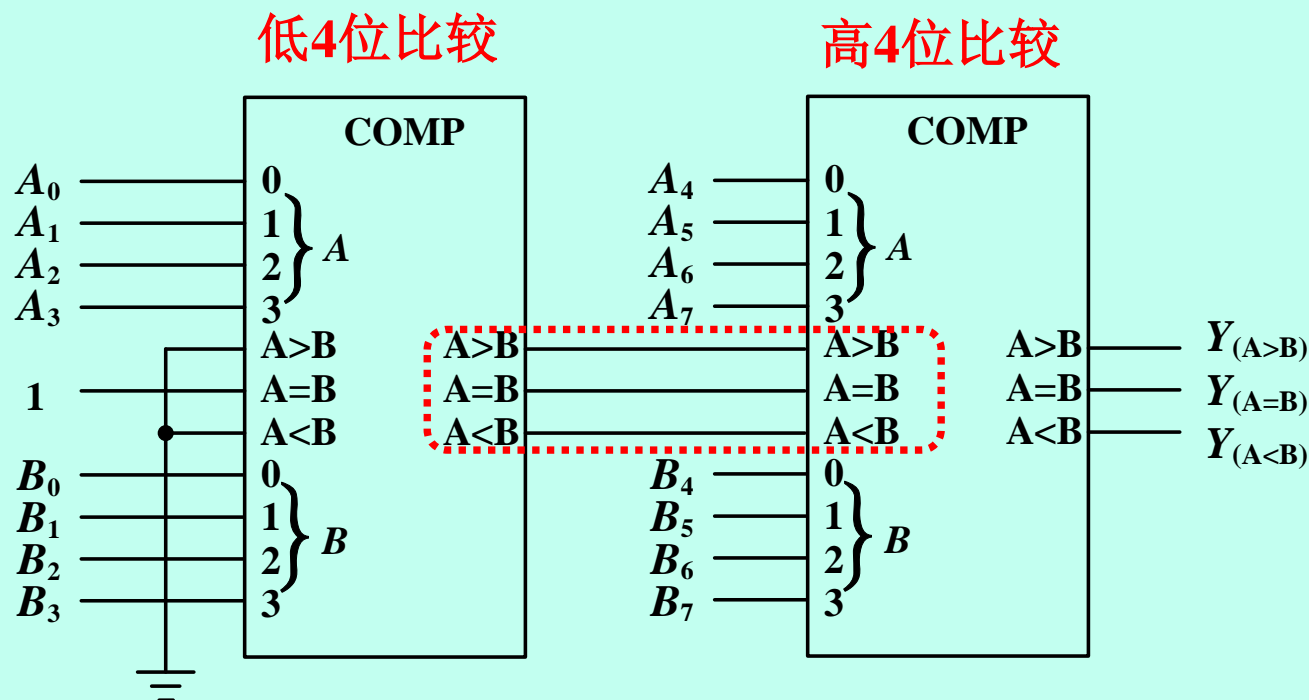




74HC85功能表

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

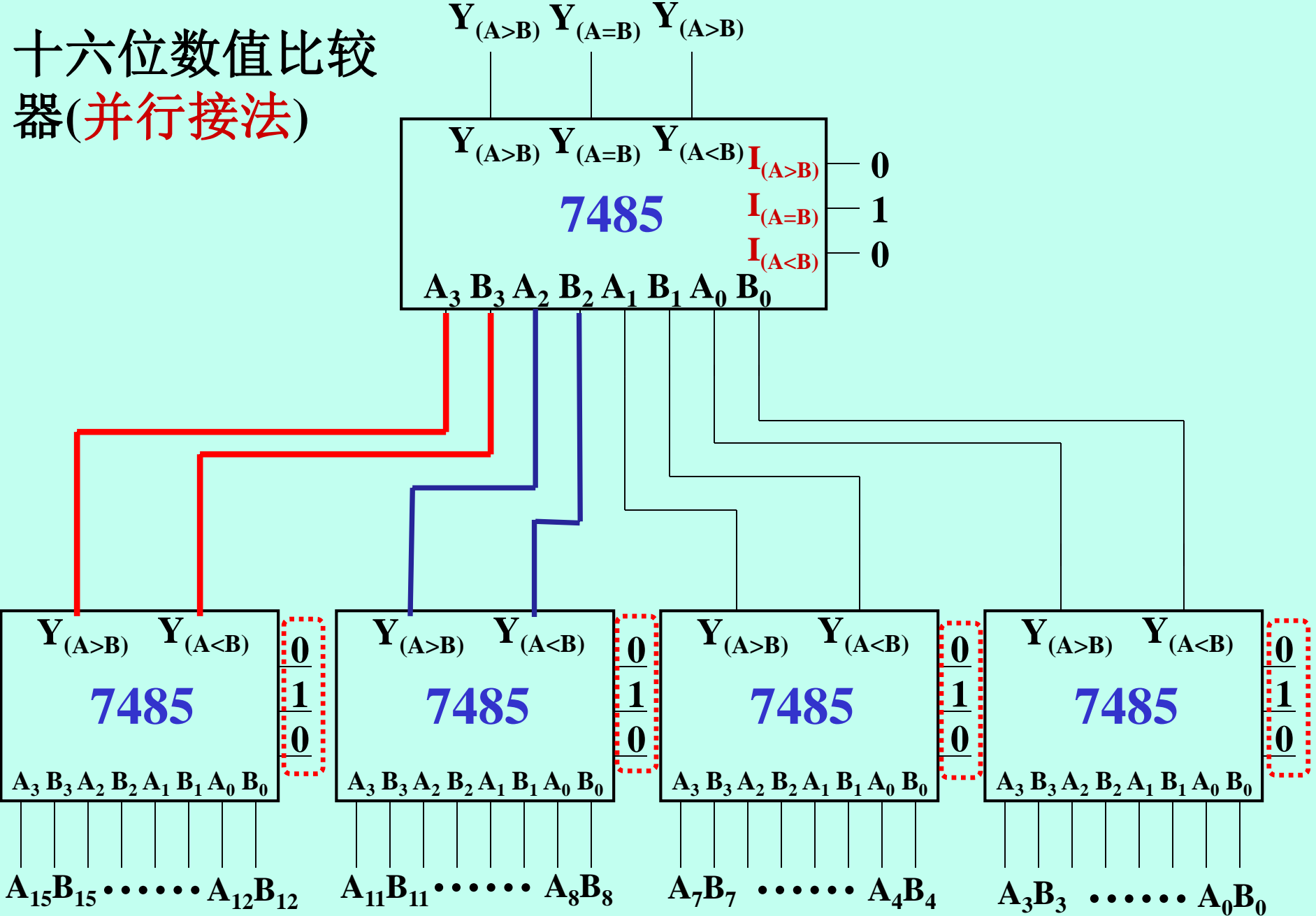
比较器的扩展:



8位数值比较器（串行接法）

- 1、如果高4位可以比较出结果（A大于B，A小于B），则无需看低4位；
- 2、如果高4位相等（A等于B），则需要看低4位比较结果（高位比较器的级联输入端）；
- 3、如果低4位可以比较出结果（A大于B，A小于B），则无需看低位比较器的级联输入端；
- 4、只有当高4位和低4位均相等时，才看低位比较器的级联输入端；

十六位数值比较器(并行接法)





串行接法和并行接法性能比较:

串行接法**电路简单**,但速度**慢**;并行接法**电路复杂**,速度**快**.



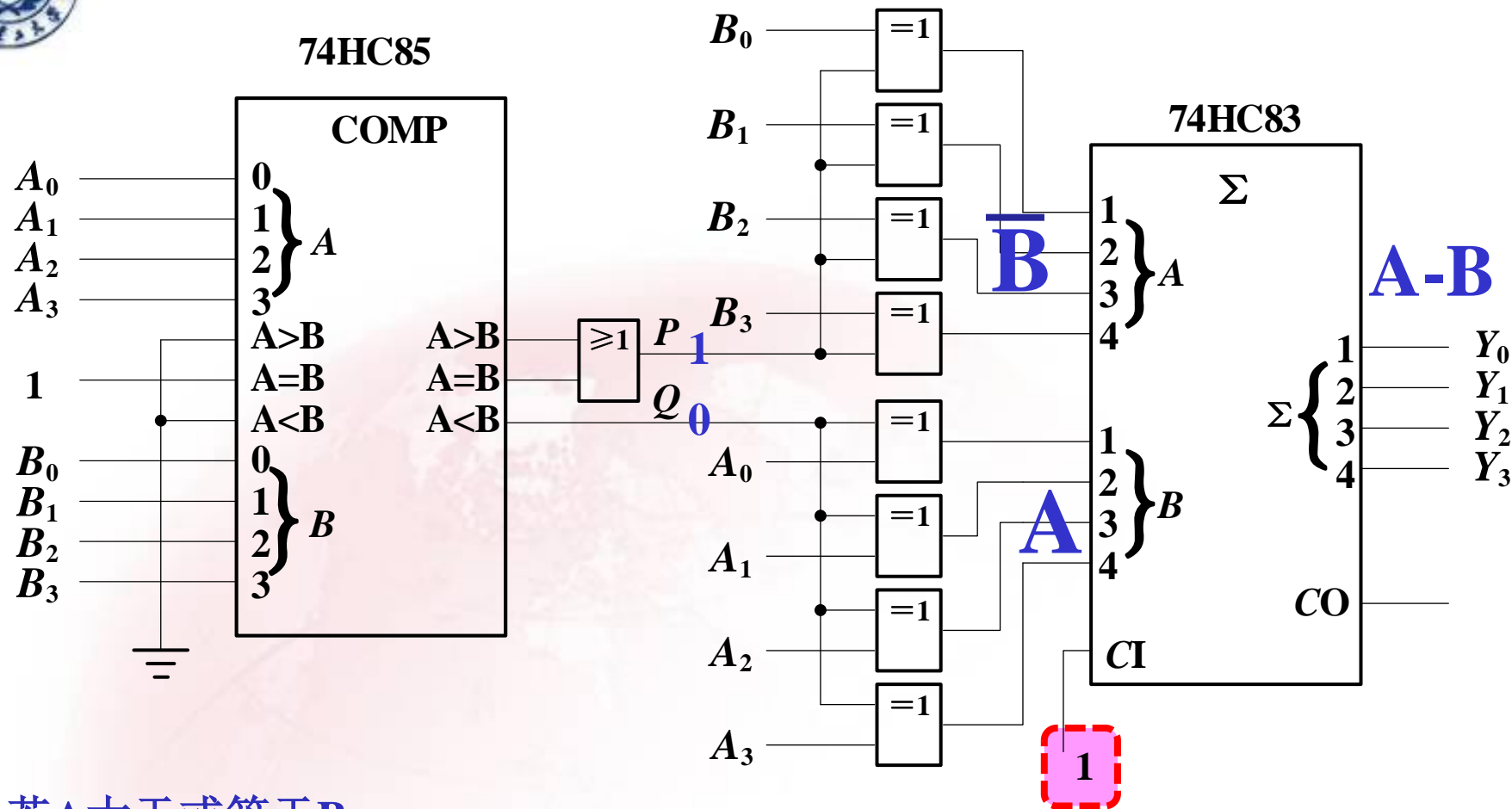


4.6.4 数值比较器应用举例

例：设计一个求两数之差绝对值电路。

设计思路：先将两数比较，对小的数求补，将得到的补码与另一数相加，得到结果。

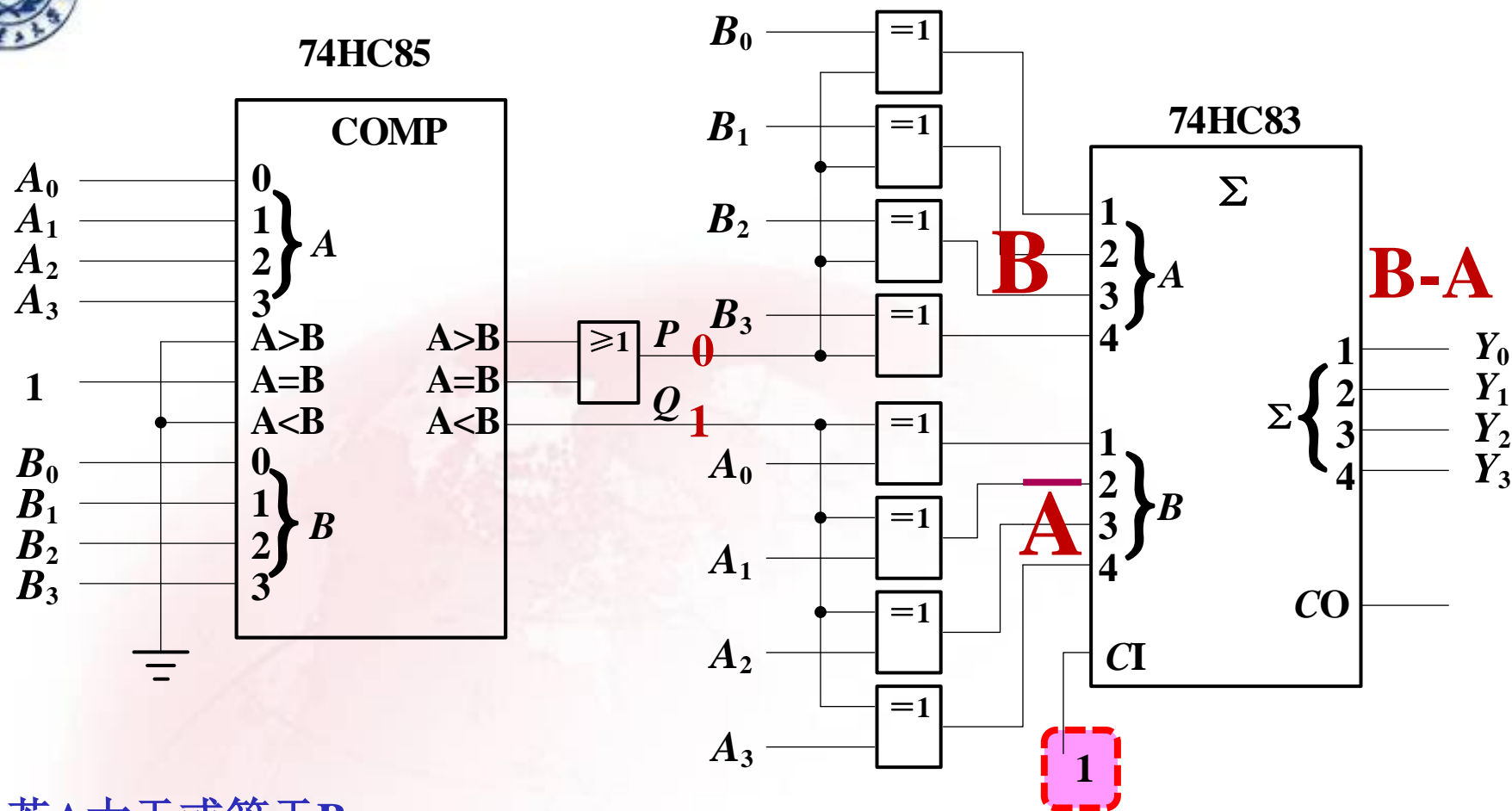




1、若A大于或等于B

求两数之差绝对值电路





1、若A大于或等于B

2、若A小于B

求两数之差绝对值电路

