

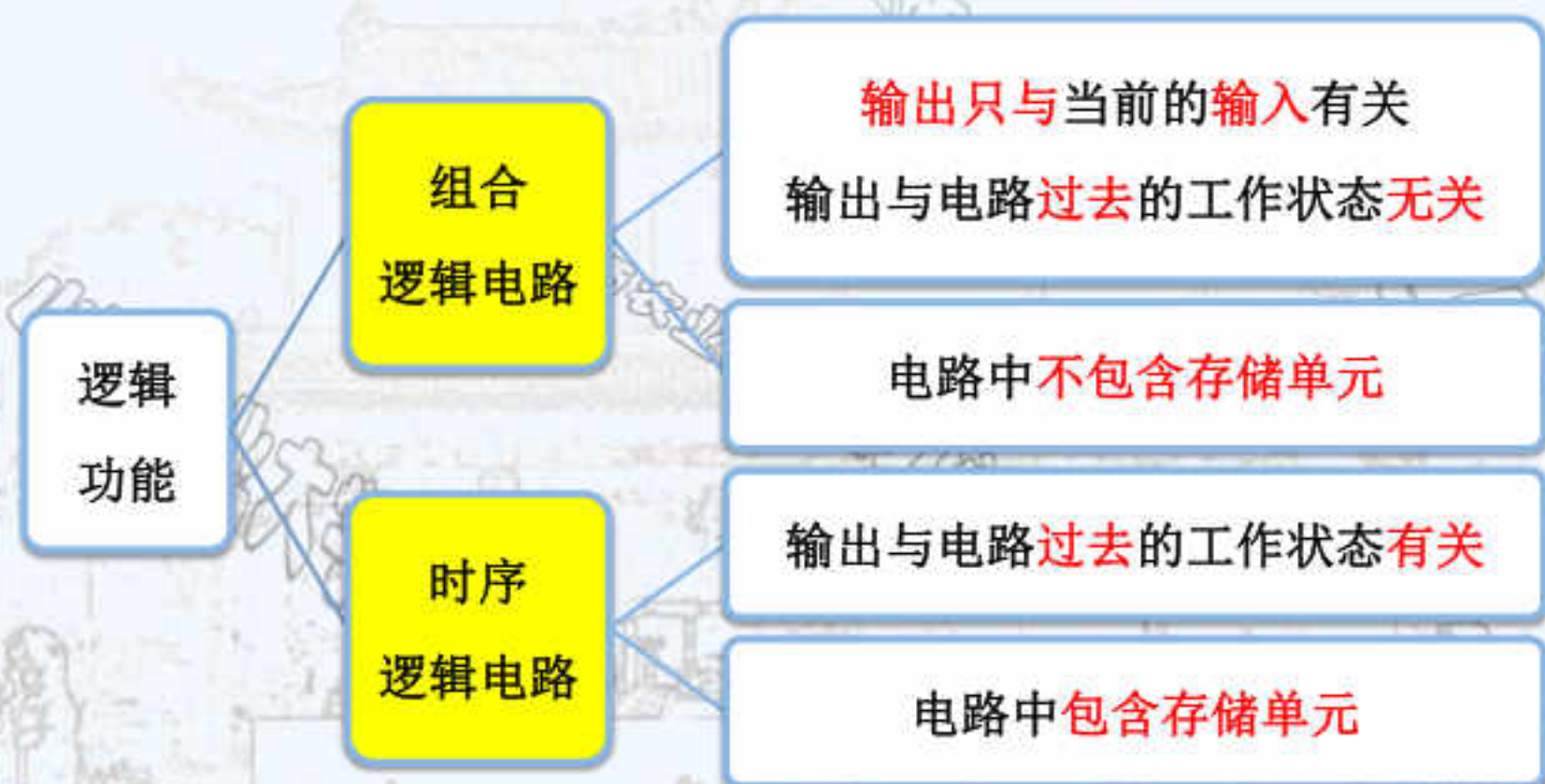
第四章 组合逻辑电路（一）

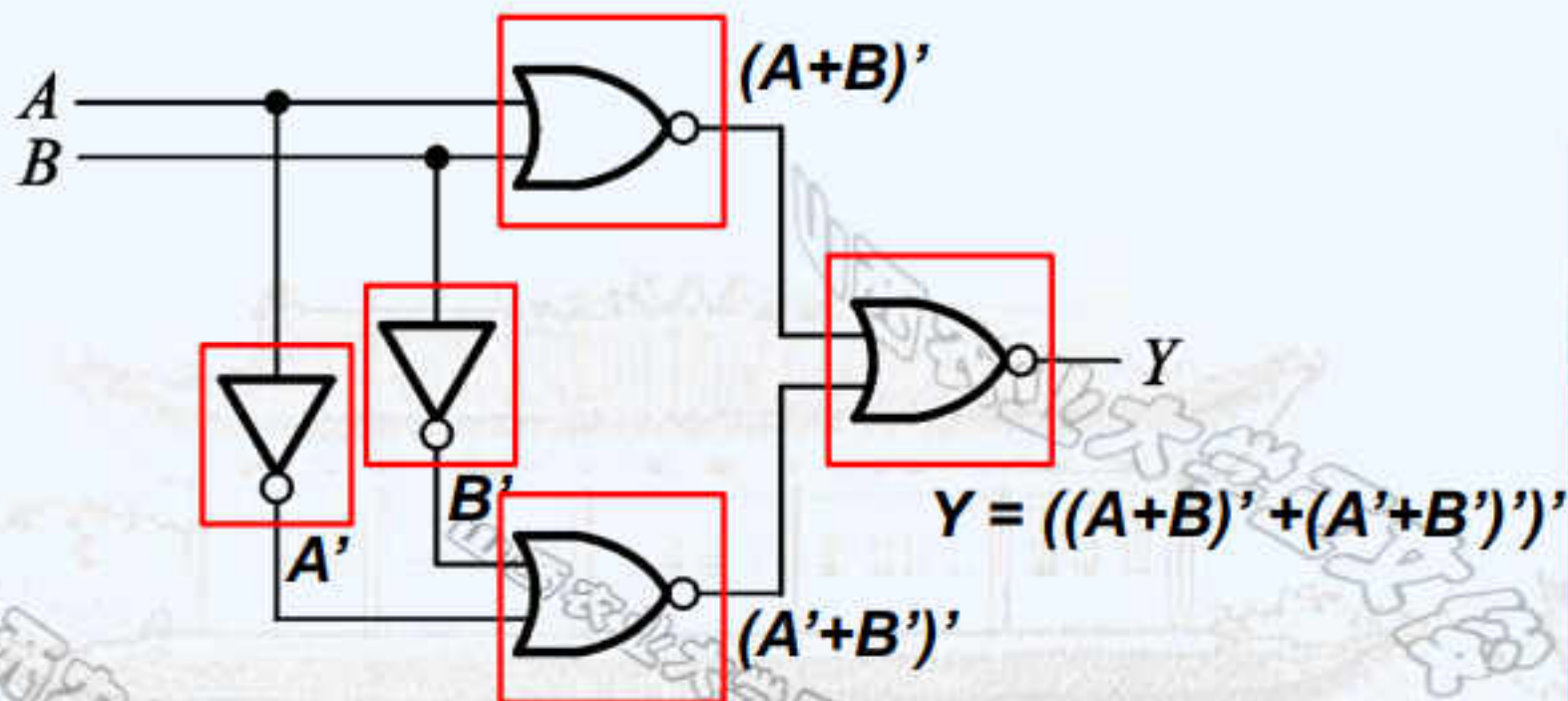
——王文俊

山西农业大学

一、组合逻辑电路的概述

• 1、逻辑电路的分类





$$Y = ((A+B)' + (A'+B')')'$$

$$Y = (A+B)(A'+B')$$

$$Y = AB' + A'B$$

任意时刻，输出 Y 只与当前的输入 A 和 B 有关

• 2、逻辑功能的描述

对于任何一个**多输入**、**多输出**的组合逻辑电路，都可以用下图表示。



逻辑函数

逻辑电路的动态参数:

传输延迟时间 t_{pd}

• 3、集成电路的分类

小规模集成电路

- 各种类型的门电路

中规模集成电路

- 编码器、译码器、数据选择器、加法器、数值比较器等

大规模集成电路

- 可编程逻辑器件PLD

二、组合逻辑电路的分析方法

组合逻辑电路的分析，目的是得到电路的**逻辑功能**。

逻辑函数 推导

- 从**输入端到输出端逐级**写出逻辑函数式，得到表示输出与输入关系的**逻辑函数式**。

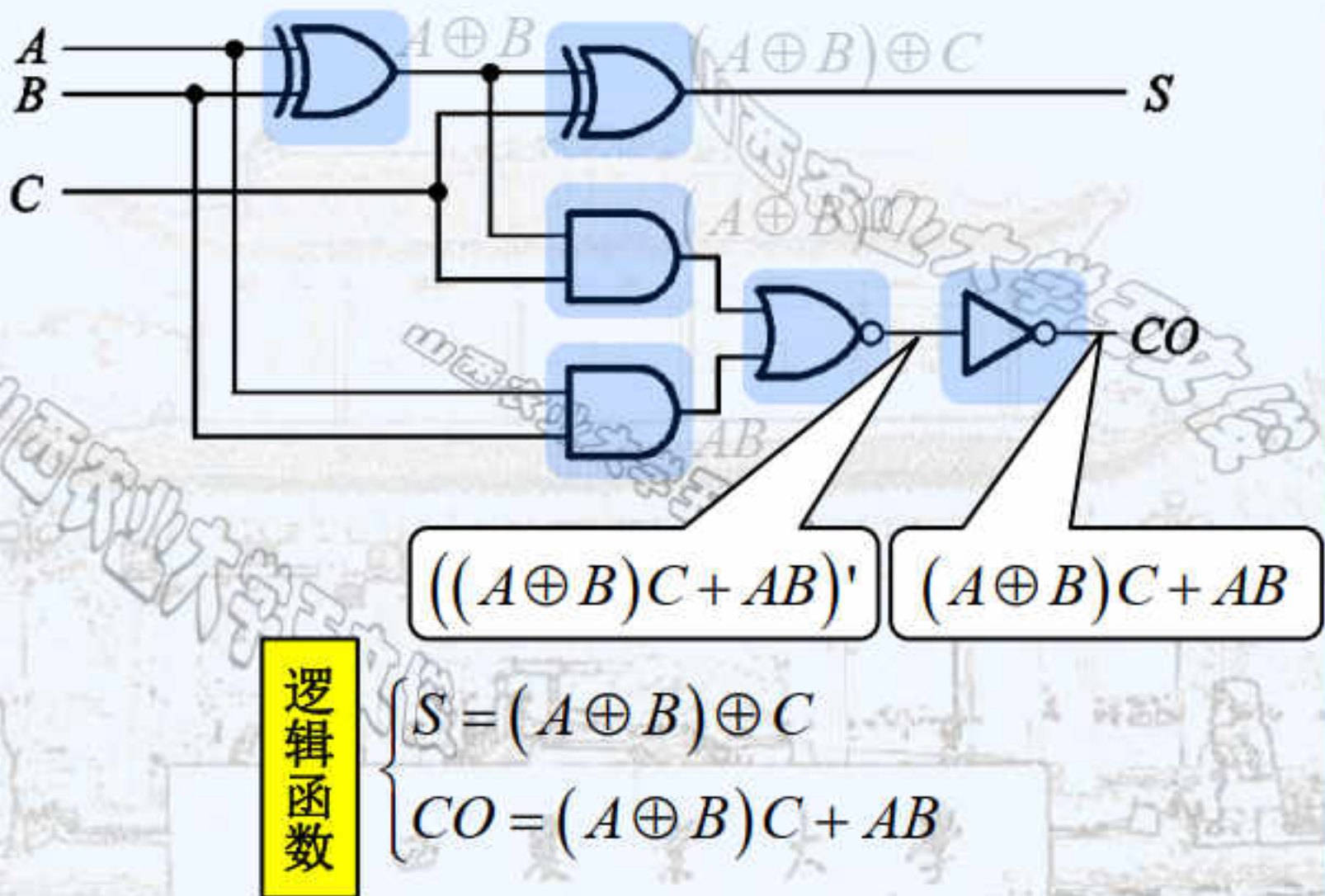
逻辑函数 化简

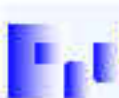
- 用公式或卡诺图化简法将得到的逻辑函数式进行**化简**，使逻辑关系简单明了。

逻辑函数 转换

- 为使电路的逻辑功能更加直观，还可将逻辑函数式转换为**真值表**的形式。

- 例1: 分析下图逻辑电路的逻辑功能



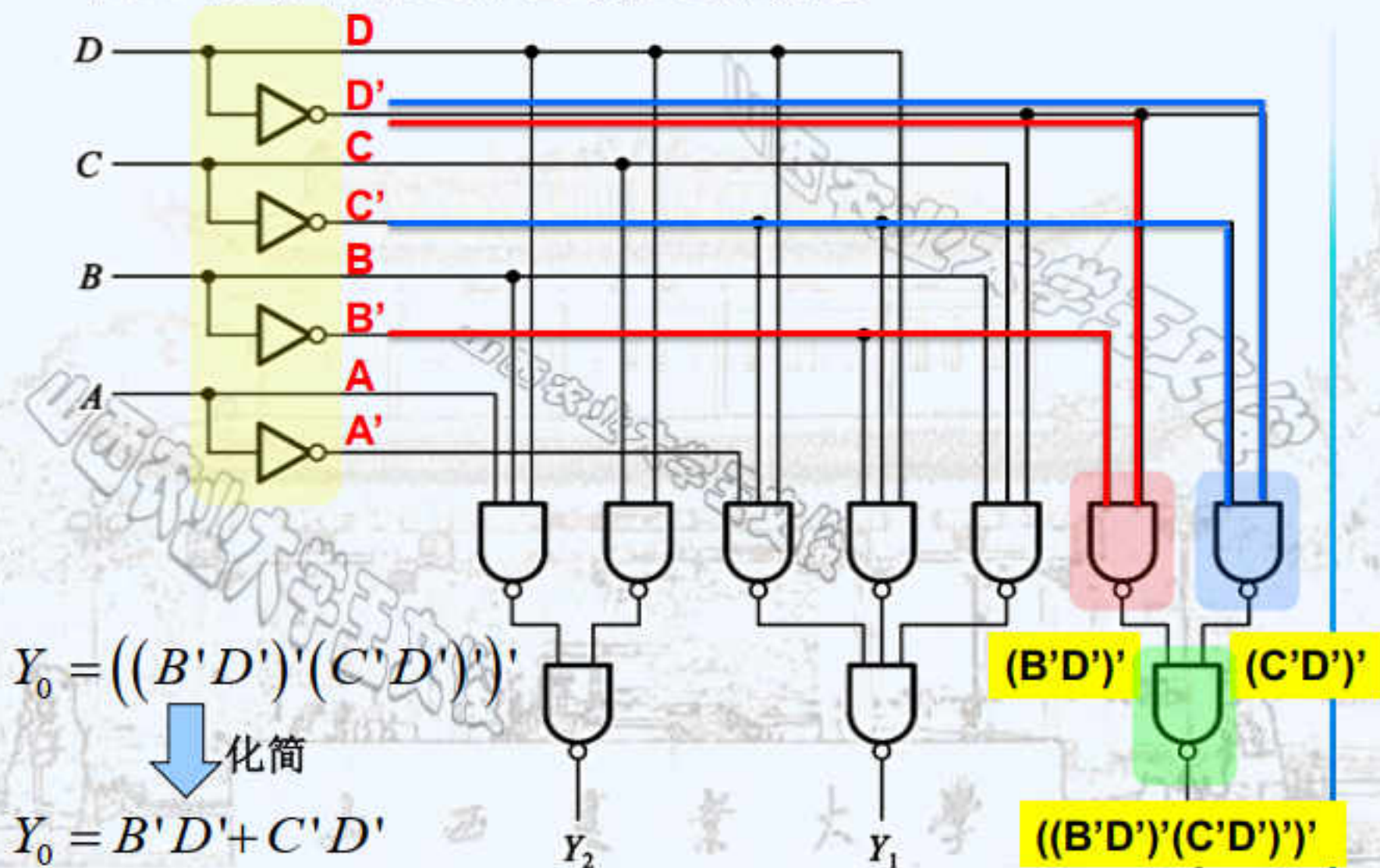


$$S = (A \oplus B) \oplus C$$

$$CO = (A \oplus B)C + AB$$

A	B	C	$A \oplus B$	$S = A \oplus B \oplus C$	$(A \oplus B)C$	AB	CO
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	1	0	0	0
0	1	1	1	0	1	0	1
1	0	0	1	1	0	0	0
1	0	1	1	0	1	0	1
1	1	0	0	0	0	1	1
1	1	1	0	1	0	1	1

• 例2：分析下图逻辑电路的逻辑功能



逻辑函数

$$Y_2 = ((DC)'(DBA)')' = DC + DBA$$

$$Y_1 = ((D'CB)'(DC'B')'(DC'A')')' = D'CB + DC'B' + DC'A'$$

$$Y_0 = ((D'C')'(D'B'))' = D'C' + D'B'$$

输 入				输 出		
D	C	B	A	Y_2	Y_1	Y_0
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0

输 入				输 出		
D	C	B	A	Y_2	Y_1	Y_0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

$DCBA$ 小于等于 5, $Y_0=1$; 6~10 之间, $Y_1=1$; 大于等于 11, $Y_2=1$

三、组合逻辑电路的基本设计方法

• 1、进行逻辑抽象

确定输入变量
和输出变量

- 引起事件的原因定义输入变量
- 事件的结果定义为输出变量

定义逻辑状态
的含义

- 对输入、输出变量进行编码
- 明确0、1所代表的具体含义

列出真值表

- 根据给定的因果关系，列出真值表

• 2、写出逻辑函数式

写出逻辑函数式

- 将真值表转换为逻辑函数式

• 3、逻辑函数的化简或变换

逻辑函数的化简

- 最简：乘积项最少、乘积项因子最少

逻辑函数的变换

- 根据门电路类型，对函数式进行变换

• 4、画出逻辑电路图

画出逻辑电路图

- 画出用门电路组成的逻辑电路图

• 5、设计验证与工艺设计

设计验证

- 验证是否符合设计要求

工艺设计

- 将逻辑电路实现为具体电路装置，所需要的一系列工艺设计工作

- 组合逻辑电路设计的流程

逻辑抽象

定变量

明含义

列表格

逻辑函数式

写函数

函数的
化简或变换

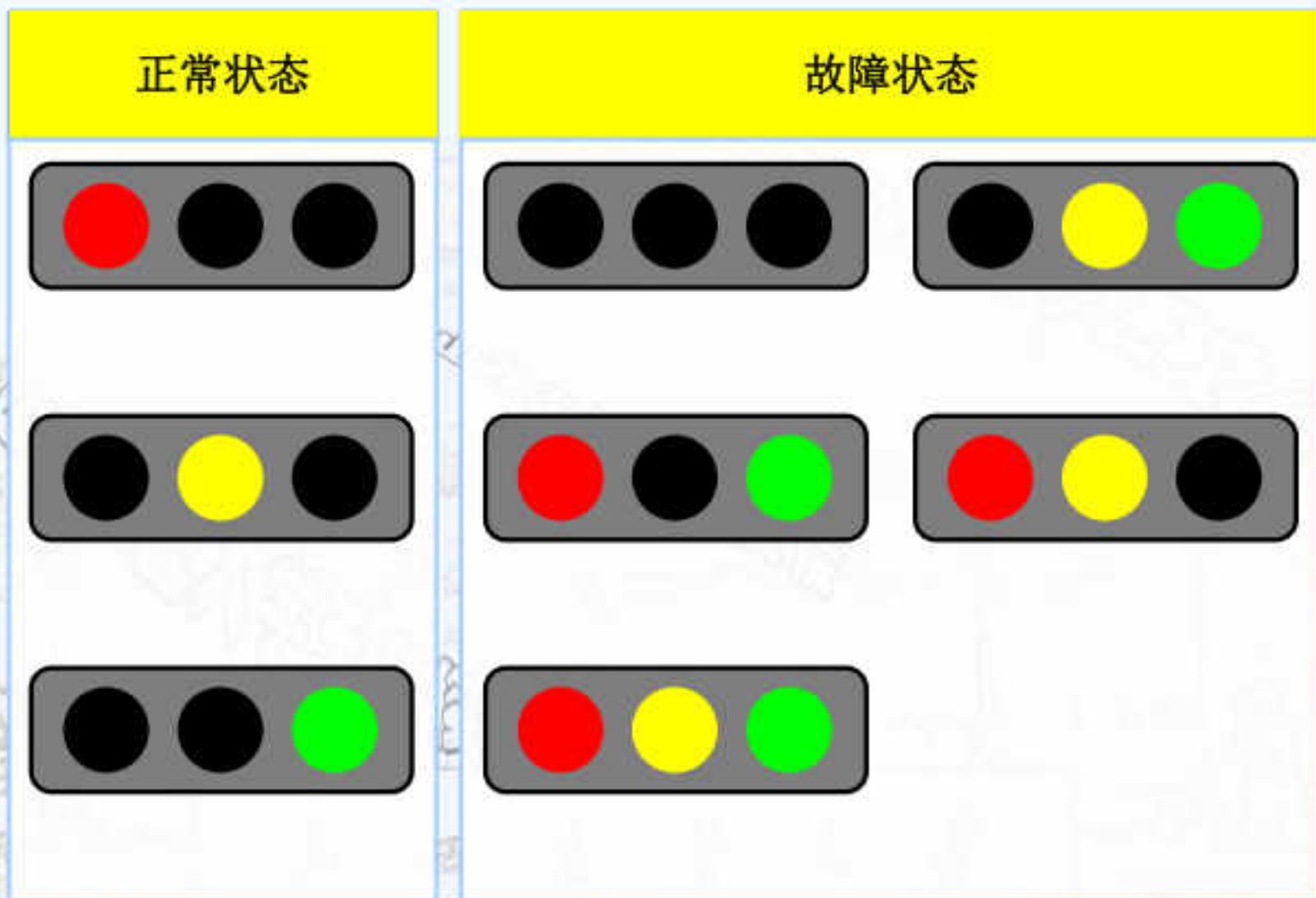
做化简

(选器件)
做变换

逻辑电路图

画电路

- 例3：设计一个监视交通信号灯状态的逻辑电路



定变量

- 输入变量：红灯 R 、黄灯 A 、绿灯 G
- 输出变量：电路工作状态 Z

明含义

- 输入变量 RAG ：灯亮为 1；灯不亮为 0
- 输出变量 Z ：正常为 0，故障为 1



正常状态 $Z = 0$



1 0 0



0 1 0



0 0 1

故障状态 $Z = 1$



0 0 0



0 1 1



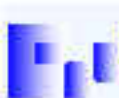
1 0 1



1 1 0



1 1 1



列表格

输入变量			输出
R	A	G	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

写函数

$$Z = R'A'G' + R'AG + RA'G + RAG' + RAG$$

$$R'A'G'$$

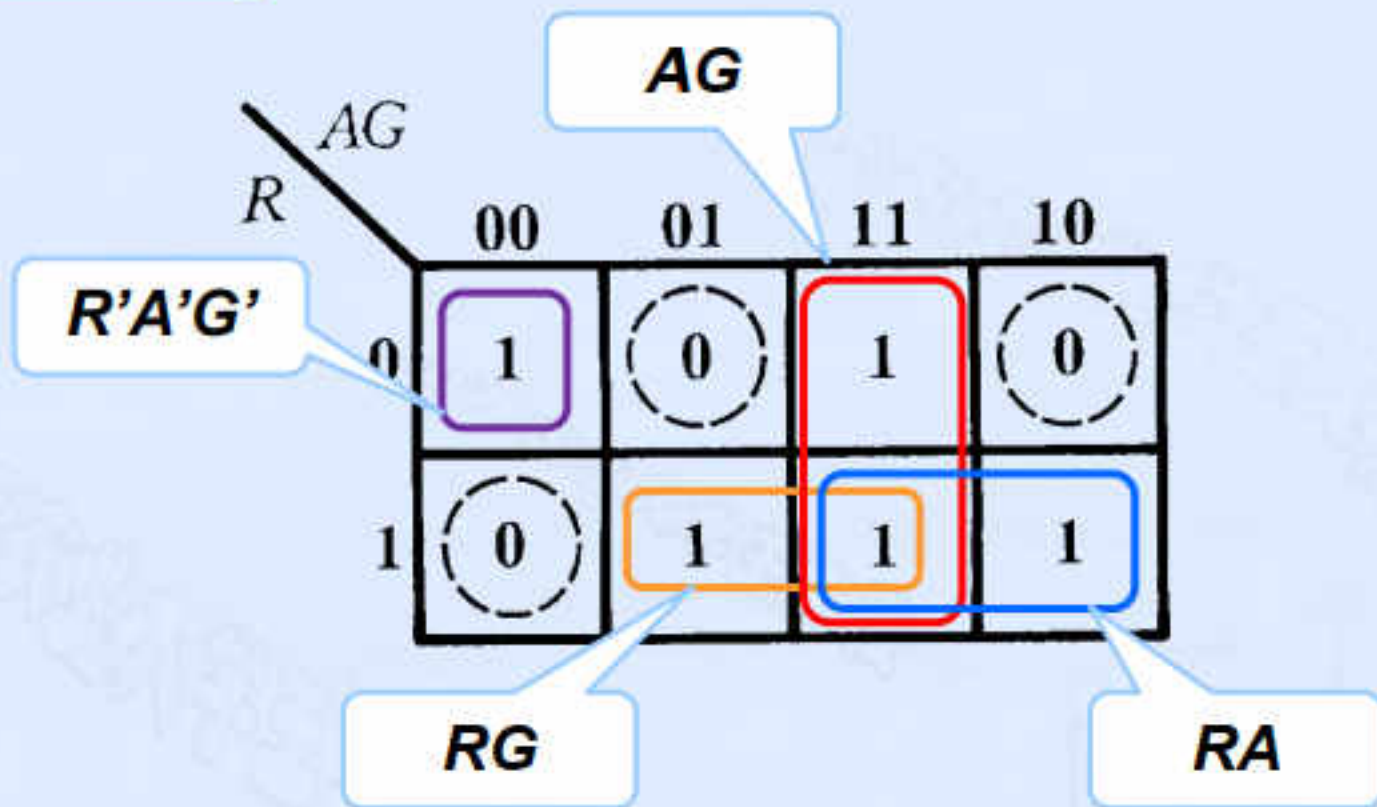
$$R'AG$$

$$RA'G$$

$$RAG'$$

$$RAG$$

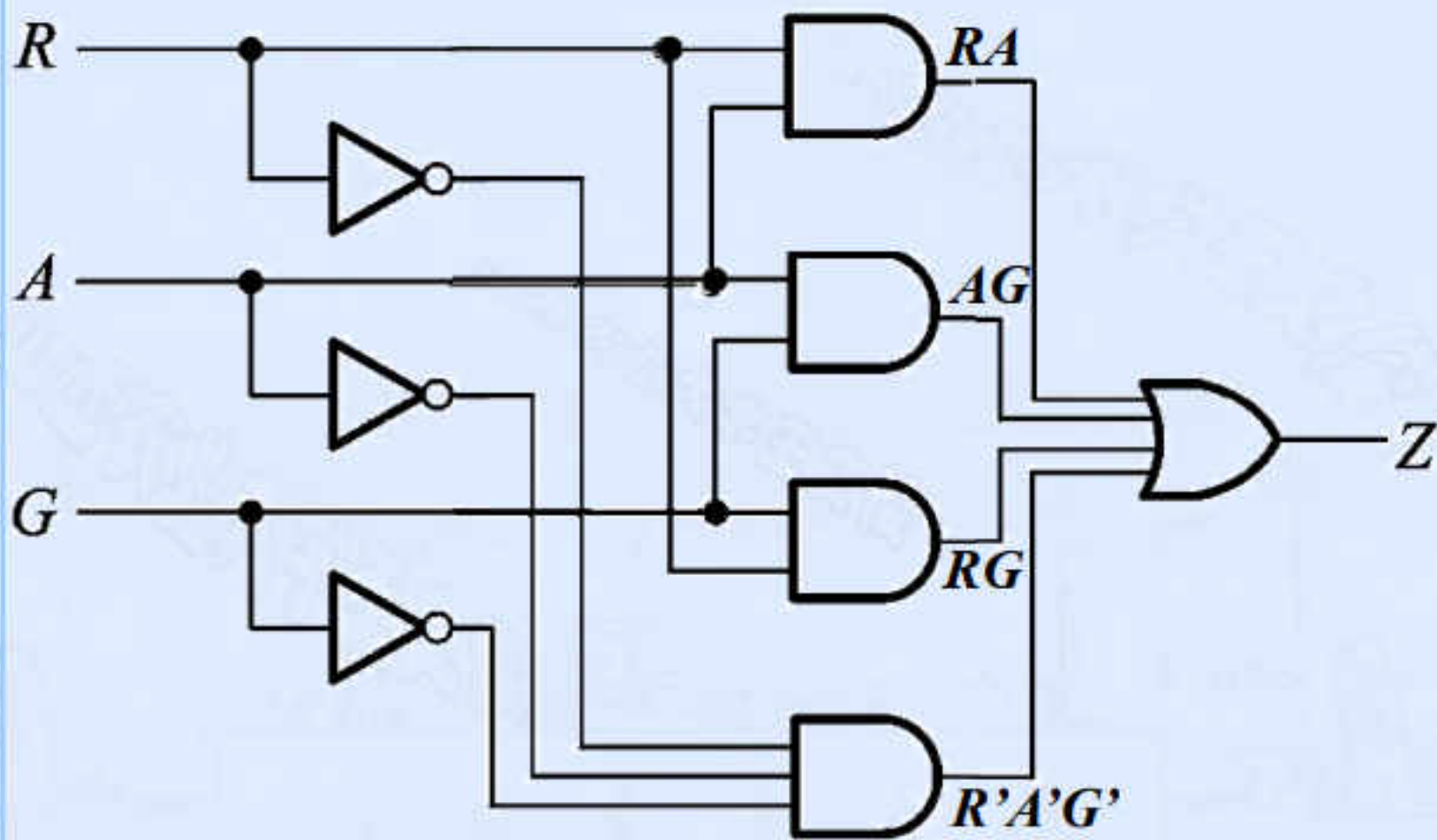
做化简



$$Z = RA + AG + RG + R'A'G'$$

画电路

$$Z = RA + AG + RG + R'A'G'$$



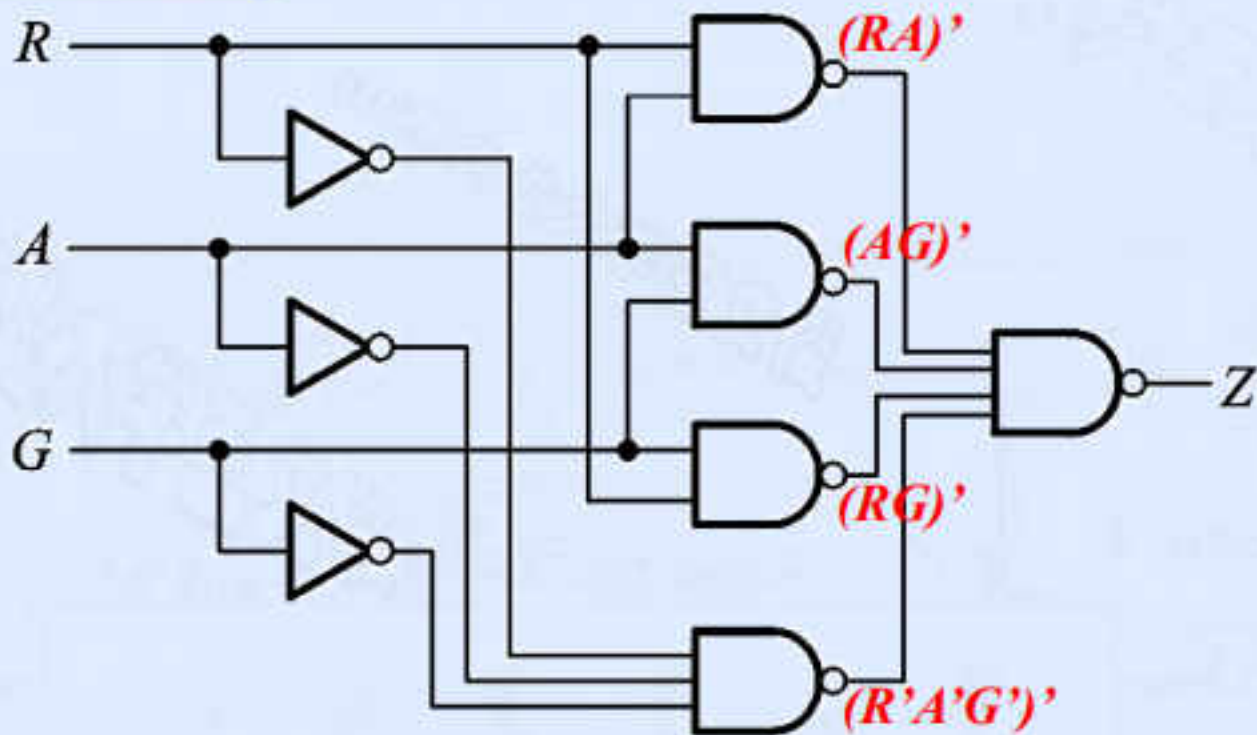
做变换1

与非 - 与非
形式

$$Z = R' A' G' + RA + RG + AG$$

$$= ((R' A' G' + RA + RG + AG)')'$$

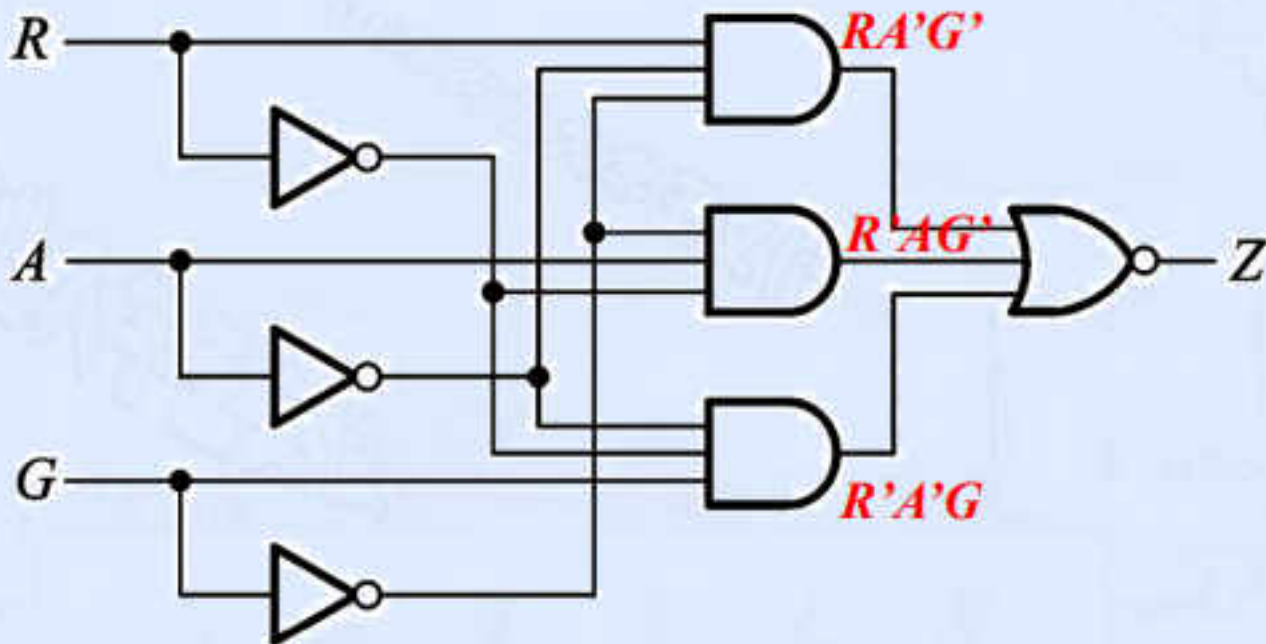
$$= ((R' A' G')' (RA)' (RG)' (AG)')'$$



做变换2

与或非
形式

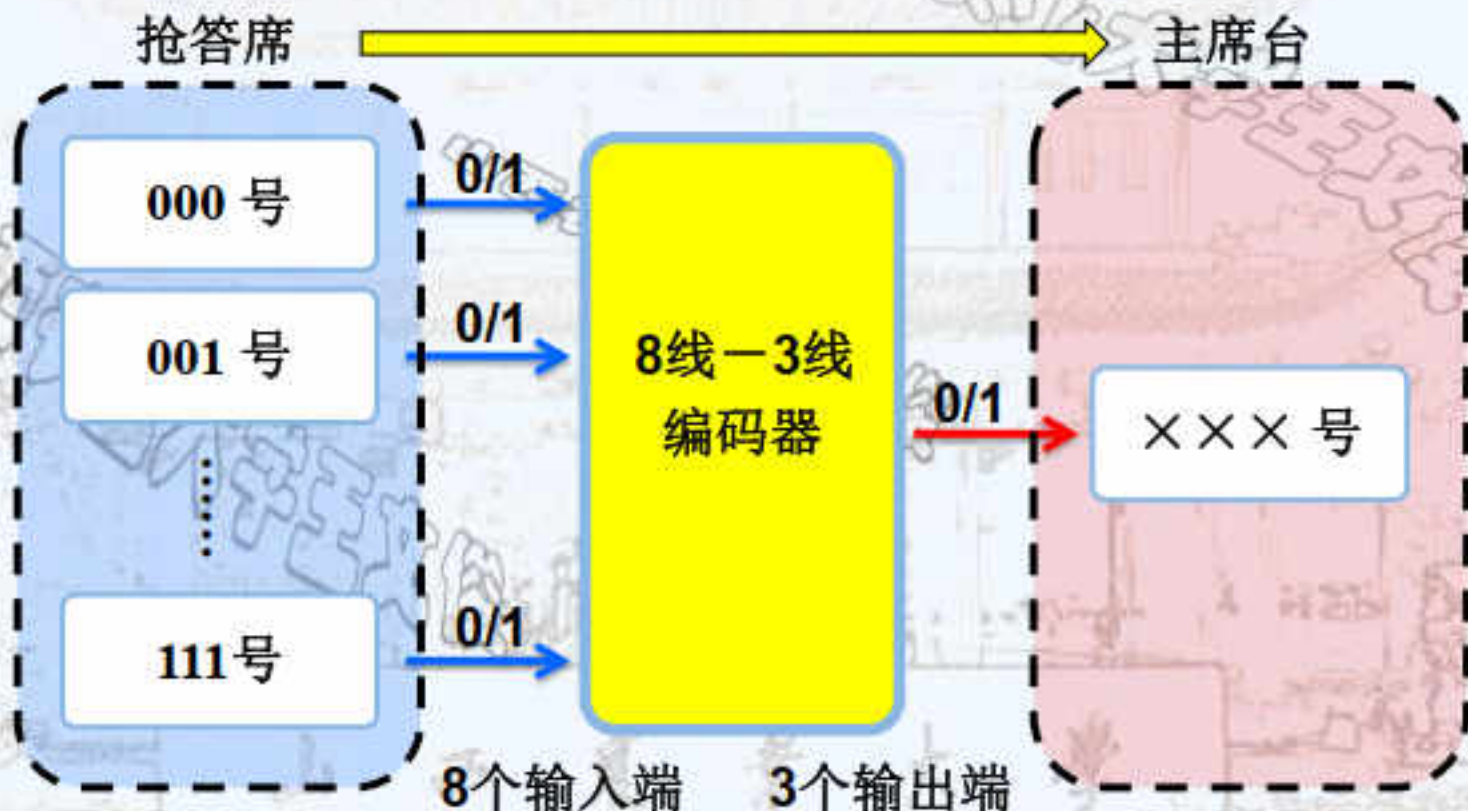
$$Z = R' A' G' + RA + RG + AG$$
$$= (RA'G' + R'AG' + R'A'G)'$$



四、普通编码器

• 1、引言

将输入的每一个高、低电平信号，编码成一个对应的二进制代码。



• 2、普通编码器的设计

定变量

- 输入变量: $I_0(000)$ 、 $I_1(001)$ 、.....、 $I_7(111)$
- 输出变量: 三位二进制数 $Y_2 Y_1 Y_0$

明含义

- 输入变量: 有输入为 1, 无输入为 0
- 输出变量: 1 为有输出, 0 为无输出

列表格

输 入								输 出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



$$\bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7$$



$$\bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7$$



$$\bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7$$



$$\bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7$$

写函数

$$Y_0 = \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7$$

$$Y_1 = \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7$$

$$Y_2 = \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7$$

做化简

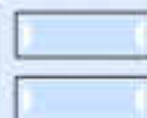
普通编码器：任何时刻只允许有一个编码信号输入。

$$Y_0 = \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 I_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 I_5 \bar{I}_6 \bar{I}_7 + \bar{I}_0 \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 I_5 I_6 \bar{I}_7$$

化简前

 $I_1 = 1$
其他全 = 0

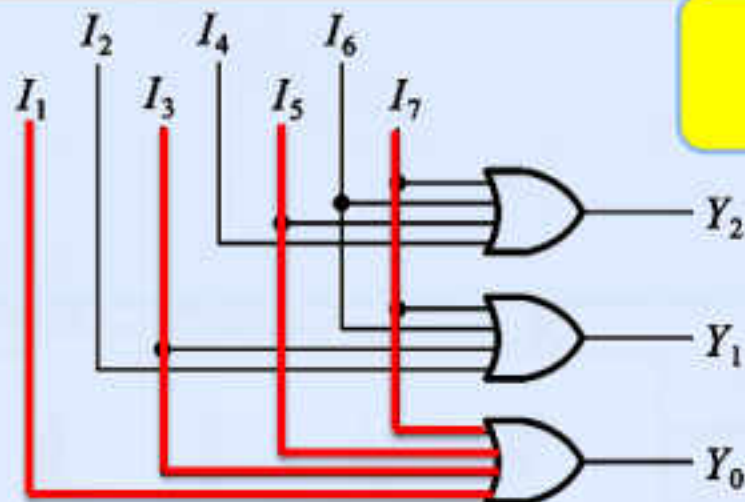

限制项

 $I_1 = 1$
其他至少有一个 = 1


化简后

 $I_1 = 1$

画电路



五、优先编码器

- 1、普通编码器与优先编码器

普通编码器

- 任何时刻只允许输入一个编码信号

优先编码器

- 允许同时输入两个以上的编码信号
- 只对其中优先权最高的一个进行编码

山西农业大学

• 2、优先编码器的设计

定变量

- 输入变量: I_0' (000号)、 I_1' (001号)、.....、 I_7' (111号)
- 输出变量: Y_2' Y_1' Y_0'

明含义

- 输入变量: 有输入为 0, 无输入为 1
- 输出变量: 0 为有输出, 1 为无输出

列表格

 I_7' 优先权最高... I_0' 优先权最低

只对优先权最高的一个进行编码

输 入								输 出		
I_0'	I_1'	I_2'	I_3'	I_4'	I_5'	I_6'	I_7'	Y_2'	Y_1'	Y_0'
×	×	×	×	×	×	×	0	0	0	0
×	×	×	×	×	×	0	1	0	0	1
×	×	×	×	×	0	1	1	0	1	0
×	×	×	×	0	1	1	1	0	1	1
×	×	×	0	1	1	1	1	1	0	0
×	×	0	1	1	1	1	1	1	0	1
×	0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1

 I_7 $I_5 I_6 I_7$ $I_3 I_4 I_5 I_6 I_7$ $I_1 I_2 I_3 I_4 I_5 I_6 I_7$

写函数

$$Y_0' = (I_7' + I_5' I_6' I_7' + I_3' I_4' I_5' I_6' I_7' + I_1' I_2' I_3' I_4' I_5' I_6' I_7')'$$

$$Y_1' = (I_7' + I_6' I_7' + I_3' I_4' I_5' I_6' I_7' + I_2' I_3' I_4' I_5' I_6' I_7')'$$

$$Y_2' = (I_7' + I_6' I_7' + I_5' I_6' I_7' + I_4' I_5' I_6' I_7')'$$

做化简

$$Y_0 = \frac{I_7}{A} + \frac{I_5 I_6' I_7'}{B A'} + \frac{I_3 I_4' I_5' I_6' I_7'}{B A'} + \frac{I_1 I_2' I_3' I_4' I_5' I_6' I_7'}{B A'}$$

$$A + A'B = A + B$$

$$Y_0 = I_7 + I_5 I_6' + I_3 I_4' I_5' I_6' + I_1 I_2' I_3' I_4' I_5' I_6'$$

$$Y_0 = I_7 + \left(\frac{I_5}{A} + \frac{I_3 I_4' I_5'}{B A'} + \frac{I_1 I_2' I_3' I_4' I_5'}{B A'} \right) I_6'$$

$$Y_0 = I_7 + (I_5 + I_3 I_4' + I_1 I_2' I_3' I_4') I_6'$$

$$Y_0 = I_7 + I_5 I_6' + I_3 I_4' I_6' + I_1 I_2' I_3' I_4' I_6'$$

$$Y_0 = I_7 + I_5 I_6' + \left(\frac{I_3}{A} + \frac{I_1 I_2' I_3'}{B A'} \right) I_4' I_6'$$

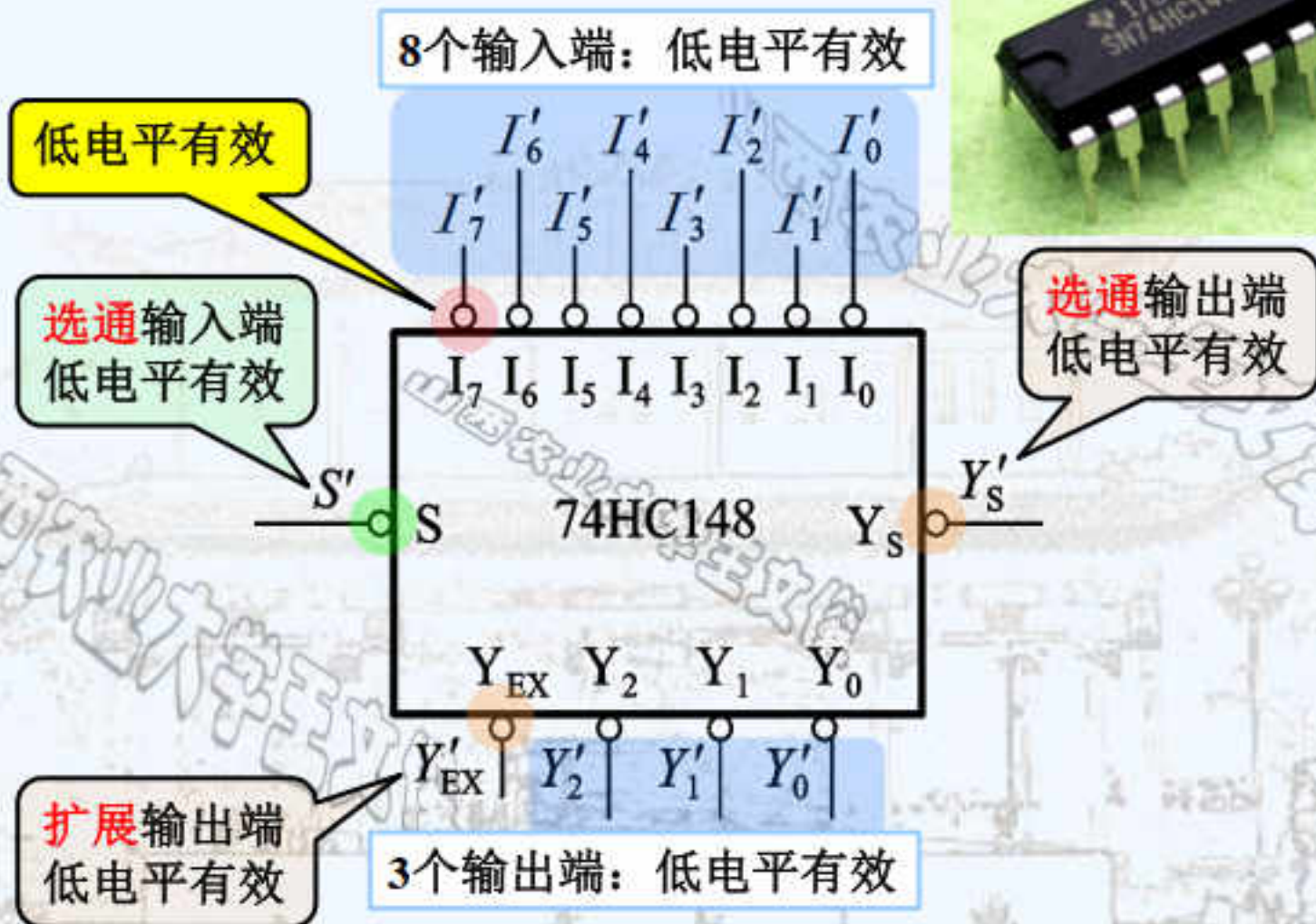
$$Y_0 = I_7 + I_5 I_6' + (I_3 + I_1 I_2') I_4' I_6'$$

$$Y_0 = I_7 + I_5 I_6' + I_3 I_4' I_6' + I_1 I_2' I_4' I_6'$$

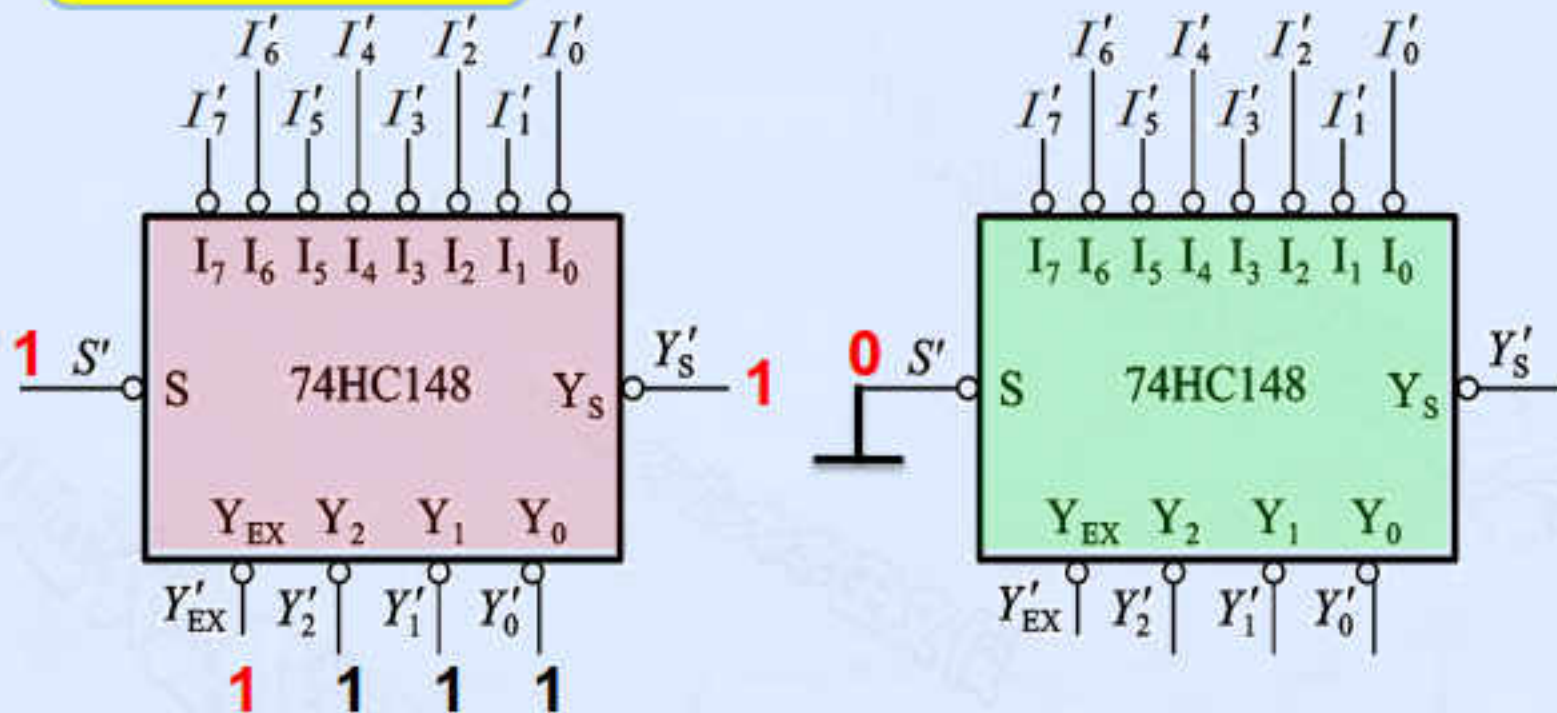
化简得：

$$\begin{cases} Y_2' = (I_4 + I_5 + I_6 + I_7)' \\ Y_1' = (I_2 I_4' I_5' + I_3 I_4' I_5' + I_6 + I_7)' \\ Y_0' = (I_1 I_2' I_4' I_6' + I_3 I_4' I_6' + I_5 I_6' + I_7)' \end{cases}$$

3、优先编码器实例 74HC148



74HC148逻辑框图

选通输入端 S'  $S' = 1$

不工作

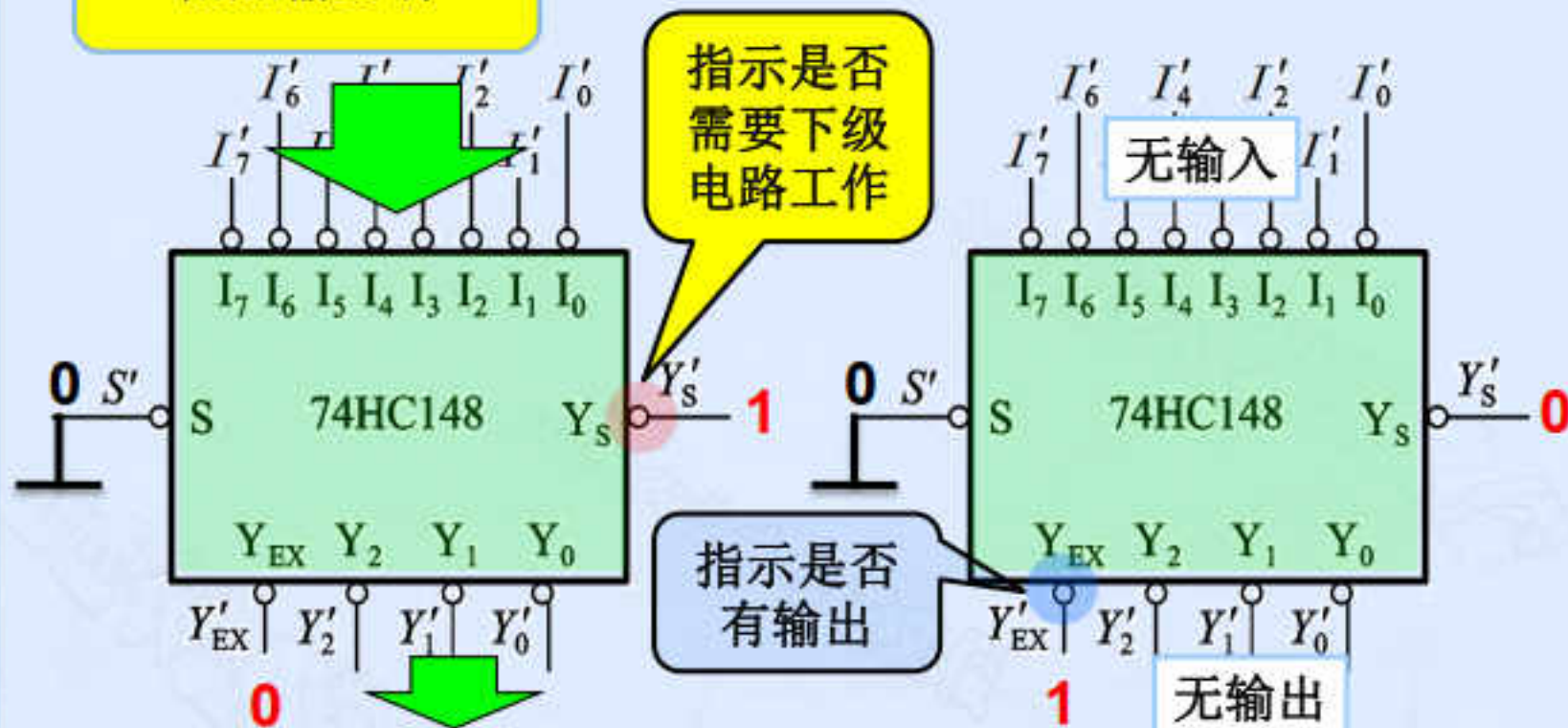
所有输出锁定为高电平

 $S' = 0$

正常工作

输入有效

附加输出端



有低电平输入信号

$Y_S' = 1$ (禁止下级电路工作)
 $Y_{EX}' = 0$ (有输出)

无低电平输入信号

$Y_S' = 0$ (允许下级电路工作)
 $Y_{EX}' = 1$ (无输出)

74HC148功能表

输 入									输 出				
S'	I_0'	I_1'	I_2'	I_3'	I_4'	I_5'	I_6'	I_7'	Y_2'	Y_1'	Y_0'	Y_S'	Y_{EX}'
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	×	×	×	×	×	×	×	0	0	0	0	1	0
0	×	×	×	×	×	×	0	1	0	0	1	1	0
0	×	×	×	×	×	0	1	1	0	1	0	1	0
0	×	×	×	×	0	1	1	1	0	1	1	1	0
0	×	×	×	0	1	1	1	1	1	0	0	1	0
0	×	×	0	1	1	1	1	1	1	0	1	1	0
0	×	0	1	1	1	1	1	1	1	1	0	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0

不工作
输出均为高电平

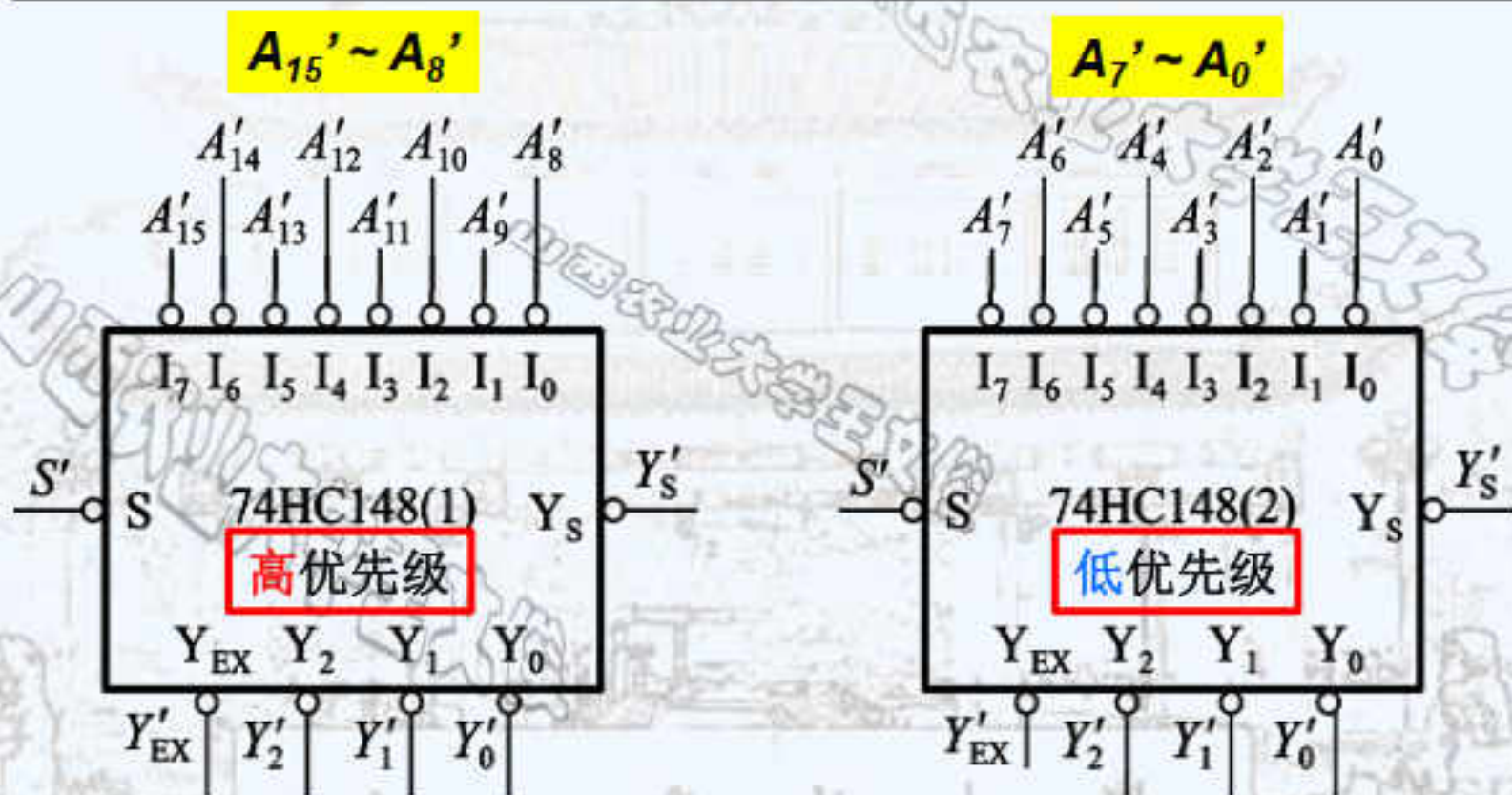
工作
无信号输入

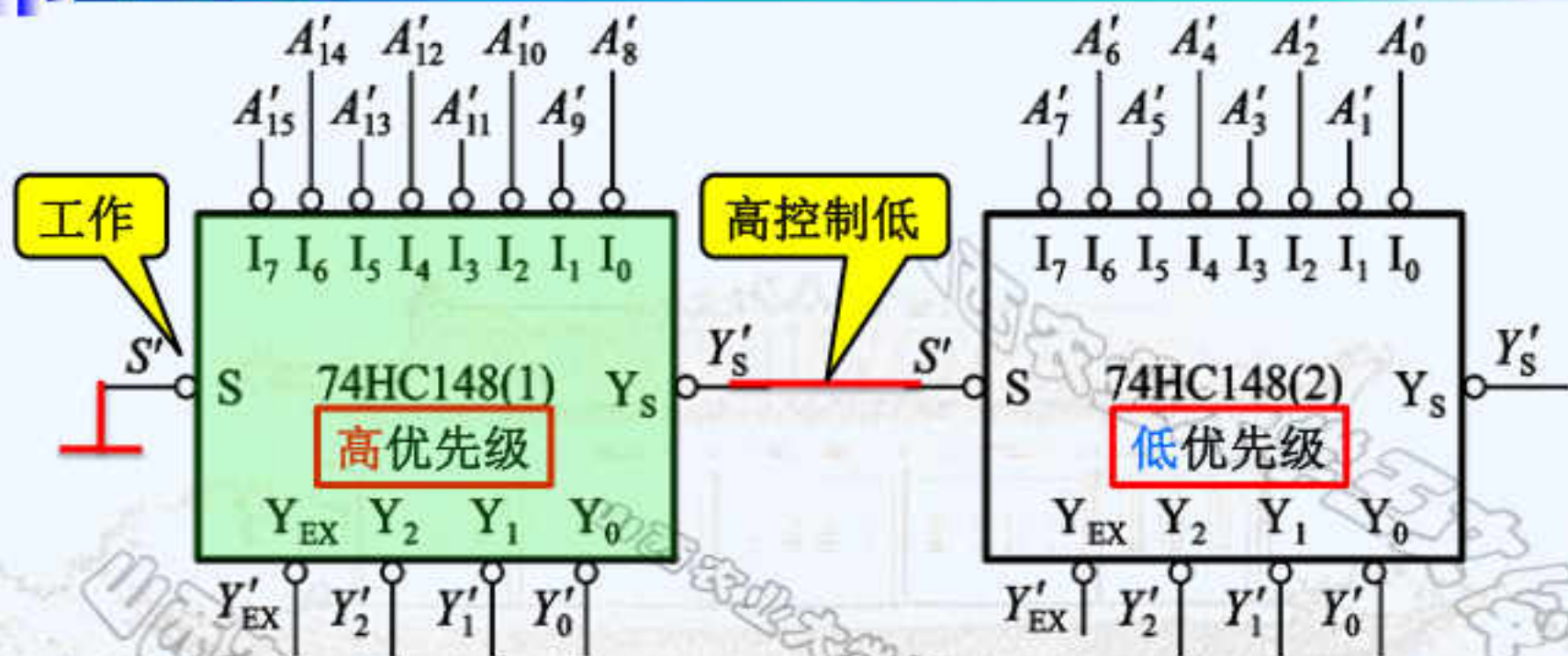
工作
有信号输入

三个111的输出
通过 Y_S' 、 Y_{EX}' 区分

- 例3：用两片8线-3线优先编码器74HC148接成16线-4线优先编码器

将 $A_0' \sim A_{15}'$ 等16个低电平输入编码为 0000 ~ 1111 的4位二进制代码。





高位有输入（反码输出： $0\times\times\times$ ）

输出端	1号	2号	输出
Y_{EX}'	0	1	0
Y_2'	Y_2'	1	Y_2'
Y_1'	Y_1'	1	Y_1'
Y_0'	Y_0'	1	Y_0'

1号扩展输出

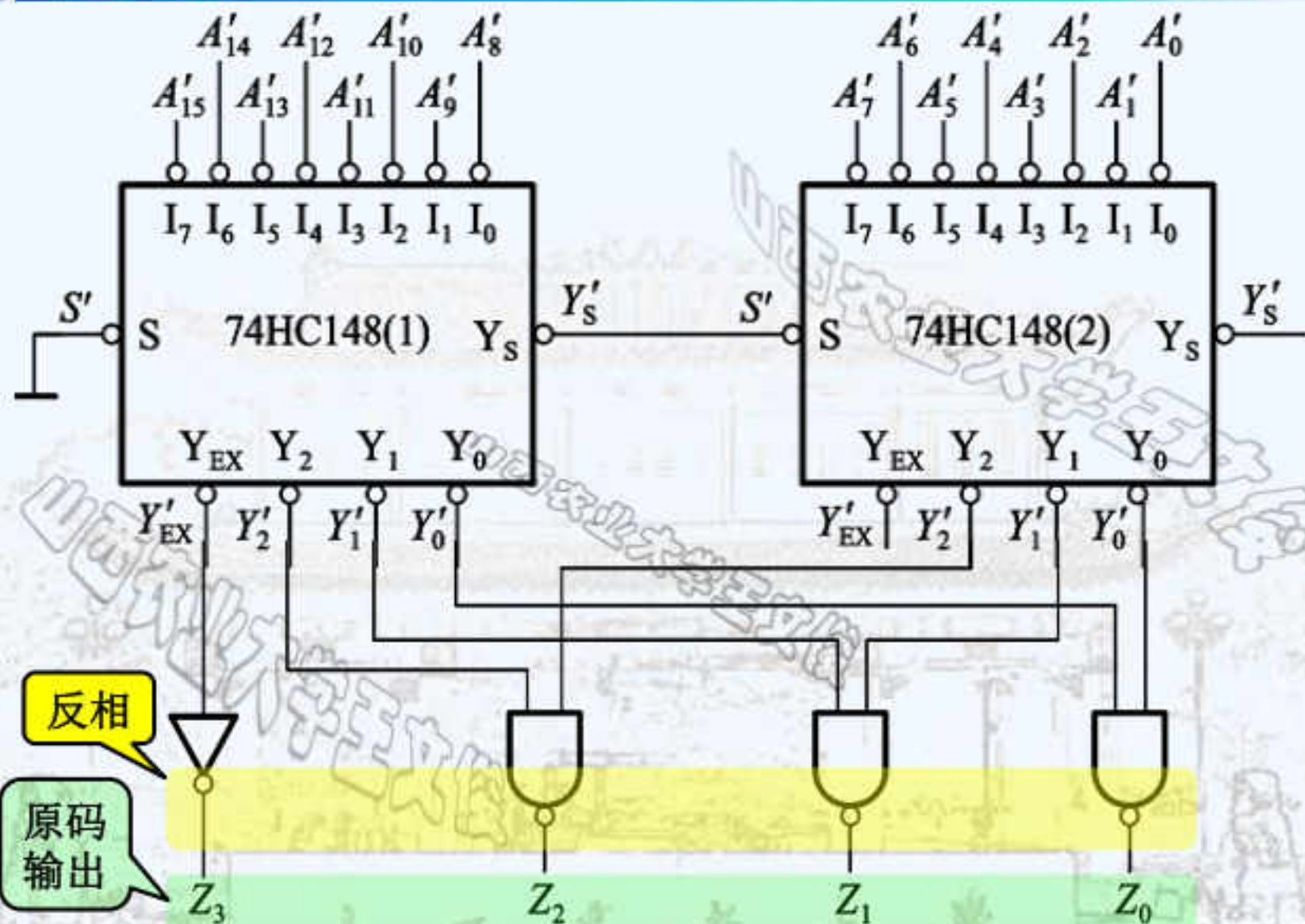
两个输出的与

低位有输入（反码输出： $1\times\times\times$ ）

输出端	1号	2号	输出
Y_{EX}'	1	0	1
Y_2'	1	Y_2'	Y_2'
Y_1'	1	Y_1'	Y_1'
Y_0'	1	Y_0'	Y_0'

1号扩展输出

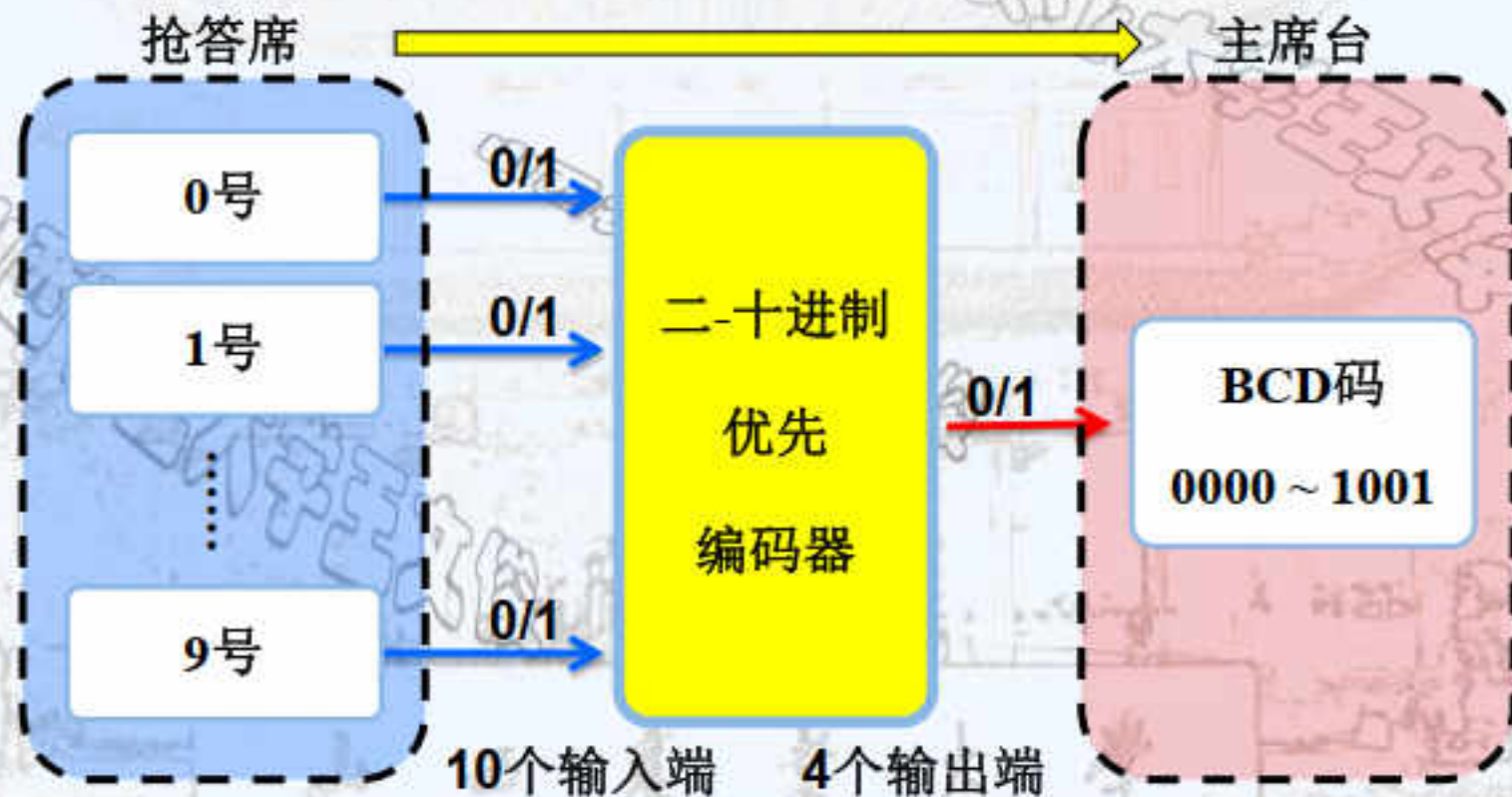
两个输出的与



六、二十进制优先编码器

• 1、引言

将输入的每一个高、低电平信号，编码成对应的十进制**BCD码**。



• 2、优先编码器的设计

定变量

- 输入变量: I_0' (0号)、 I_1' (1号)、.....、 I_9' (9号)
- 输出变量: Y_3' Y_2' Y_1' Y_0' (BCD码为反码形式)

明含义

- 输入变量: 有输入为 0, 无输入为 1
- 输出变量: 0 为有输出, 1 为无输出

列表格

 I_9' 优先权最高... I_0' 优先权最低

只对优先权最高的一个进行编码

输 入										输 出			
I_0'	I_1'	I_2'	I_3'	I_4'	I_5'	I_6'	I_7'	I_8'	I_9'	Y_3'	Y_2'	Y_1'	Y_0'
×	×	×	×	×	×	×	×	×	0	0	1	1	0
×	×	×	×	×	×	×	×	0	1	0	1	1	1
×	×	×	×	×	×	×	0	1	1	1	0	0	0
×	×	×	×	×	×	0	1	1	1	1	0	0	1
×	×	×	×	×	0	1	1	1	1	1	0	1	0
×	×	×	×	0	1	1	1	1	1	1	0	1	1
×	×	×	0	1	1	1	1	1	1	1	1	0	0
×	×	0	1	1	1	1	1	1	1	1	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

BCD码: 1001

 $I_8 I_9'$ $I_6 I_7' I_8' I_9'$ $I_4 I_5' I_6' I_7' I_8' I_9'$ $I_2 I_3' I_4' I_5' I_6' I_7' I_8' I_9'$ $I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'$

$$Y_0' = I_8 I_9' + I_6 I_7' I_8' I_9' + I_4 I_5' I_6' I_7' I_8' I_9' + I_2 I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'$$

写函数

$$Y_0' = I_8' I_9' + I_6' I_7' I_8' I_9' + I_4' I_5' I_6' I_7' I_8' I_9' \\ + I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'$$

$$Y_1' = I_9' + I_8' I_9' + I_5' I_6' I_7' I_8' I_9' + I_4' I_5' I_6' I_7' I_8' I_9' \\ + I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'$$

$$Y_2' = I_9' + I_8' I_9' + I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9' \\ + I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'$$

$$Y_3' = I_7' I_8' I_9' + I_6' I_7' I_8' I_9' + I_5' I_6' I_7' I_8' I_9' + I_4' I_5' I_6' I_7' I_8' I_9' + I_3' I_4' I_5' I_6' I_7' I_8' I_9' \\ + I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'$$

做化简

$$Y_3' = I_7' I_8' I_9' + I_6' I_7' I_8' I_9' + I_5' I_6' I_7' I_8' I_9' + I_4' I_5' I_6' I_7' I_8' I_9' + I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9' + \underline{I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'} + \underline{I_1' I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'}$$

利用 $A + A' = 1$ 合并，消去 I_1 与 I_1'

$$Y_3' = I_7' I_8' I_9' + I_6' I_7' I_8' I_9' + I_5' I_6' I_7' I_8' I_9' + I_4' I_5' I_6' I_7' I_8' I_9' + I_3' I_4' I_5' I_6' I_7' I_8' I_9' + I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9' + \underline{I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'} + \underline{I_2' I_3' I_4' I_5' I_6' I_7' I_8' I_9'}$$

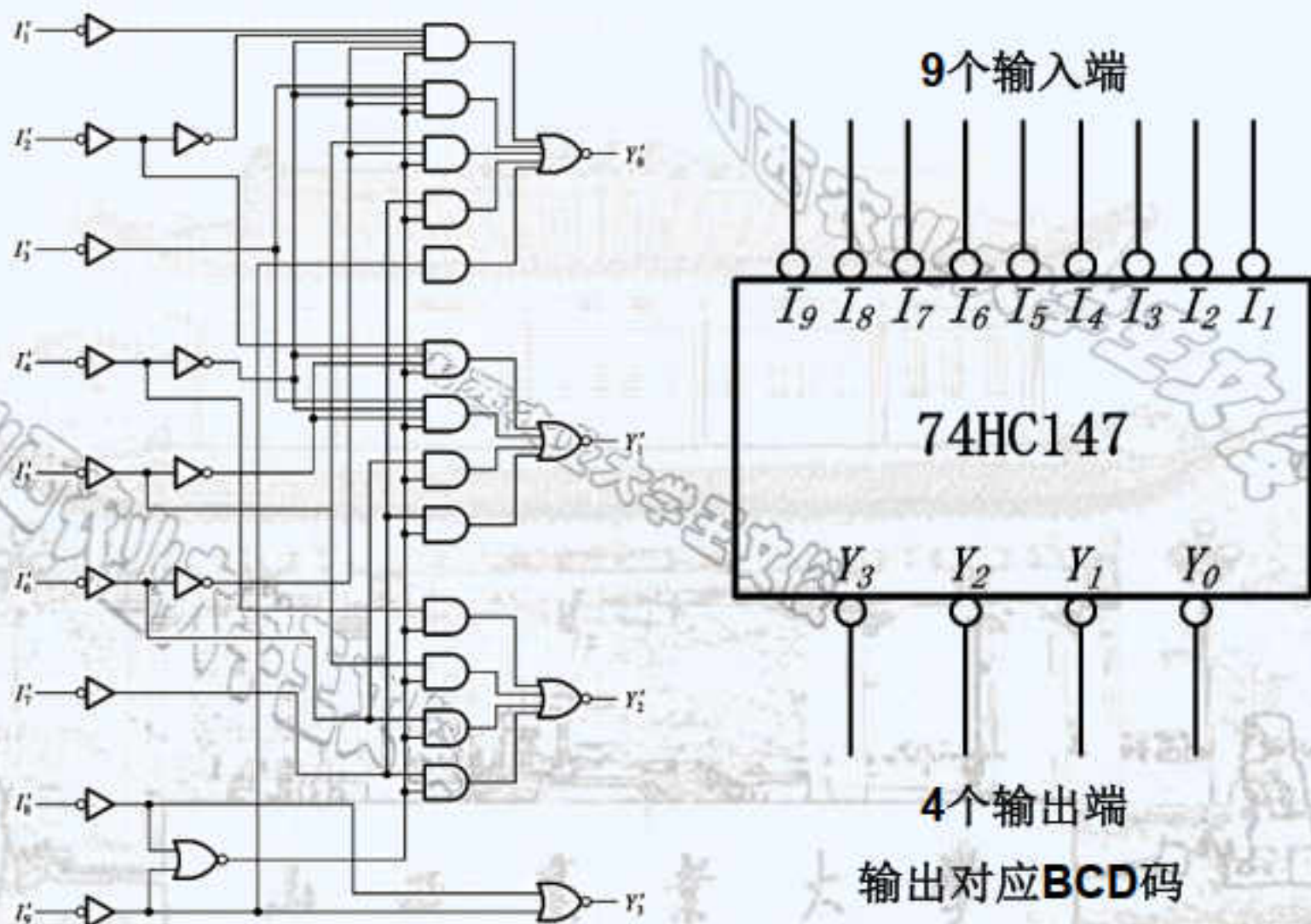
利用 $A + A' = 1$ 合并，消去 I_2 与 I_2'

依次消去 I_3 、 I_4 、 I_5 、 I_6 、 I_7

$$Y_3' = I_8' I_9' = (I_8 + I_9)'$$

$$\begin{cases} Y_2' = (I_7' I_8' I_9' + I_6' I_8' I_9' + I_5' I_8' I_9' + I_4' I_8' I_9')' \\ Y_1' = (I_7' I_8' I_9' + I_6' I_8' I_9' + I_3' I_4' I_5' I_8' I_9' + I_2' I_4' I_5' I_8' I_9')' \\ Y_0' = (I_9 + I_7' I_8' I_9' + I_5' I_6' I_8' I_9' + I_3' I_4' I_6' I_8' I_9' + I_1' I_2' I_4' I_6' I_8' I_9')' \end{cases}$$

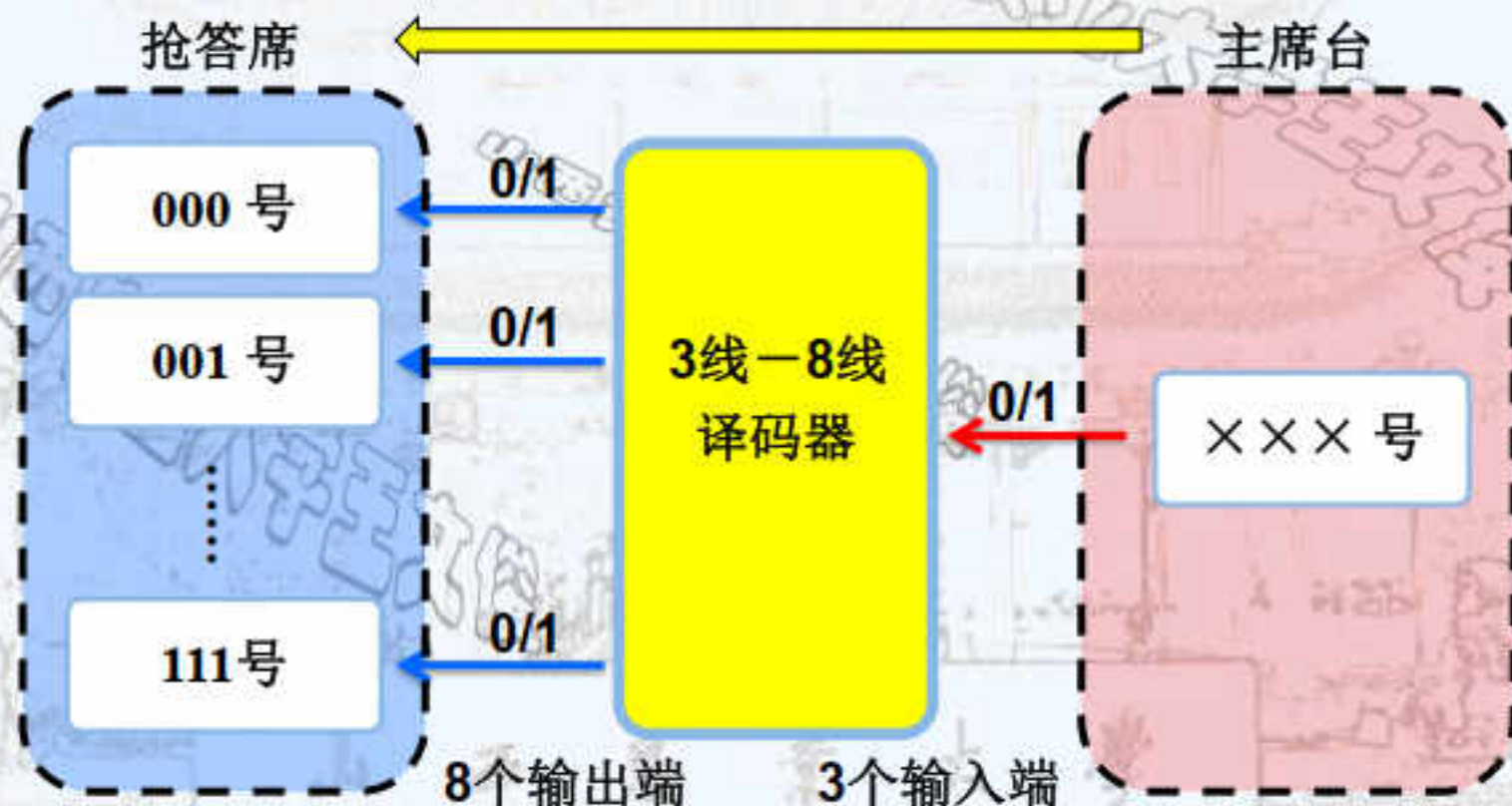
• 3、二进制优先编码器实例74HC147



七、二进制译码器

• 1、引言

将每个输入的**二进制代码**译成对应的输出高、低电平信号。



• 2、二进制译码器的设计

定变量

- 输入变量：三位二进制数 $A_2 A_1 A_0$
- 输出变量： $Y_0(000)$ 、 $Y_1(001)$ 、、 $Y_7(111)$

明含义

- 输入变量：有输入为 1，无输入为 0
- 输出变量：1 为有输出，0 为无输出

列表格

输 入			输 出							
A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

写函数

输出为 $A_2 A_1 A_0$
的全部最小项

$$Y_0 = A_2' A_1' A_0' = m_0$$

$$Y_1 = A_2' A_1' A_0 = m_1$$

$$Y_2 = A_2' A_1 A_0' = m_2$$

$$Y_3 = A_2' A_1 A_0 = m_3$$

$$Y_4 = A_2 A_1' A_0' = m_4$$

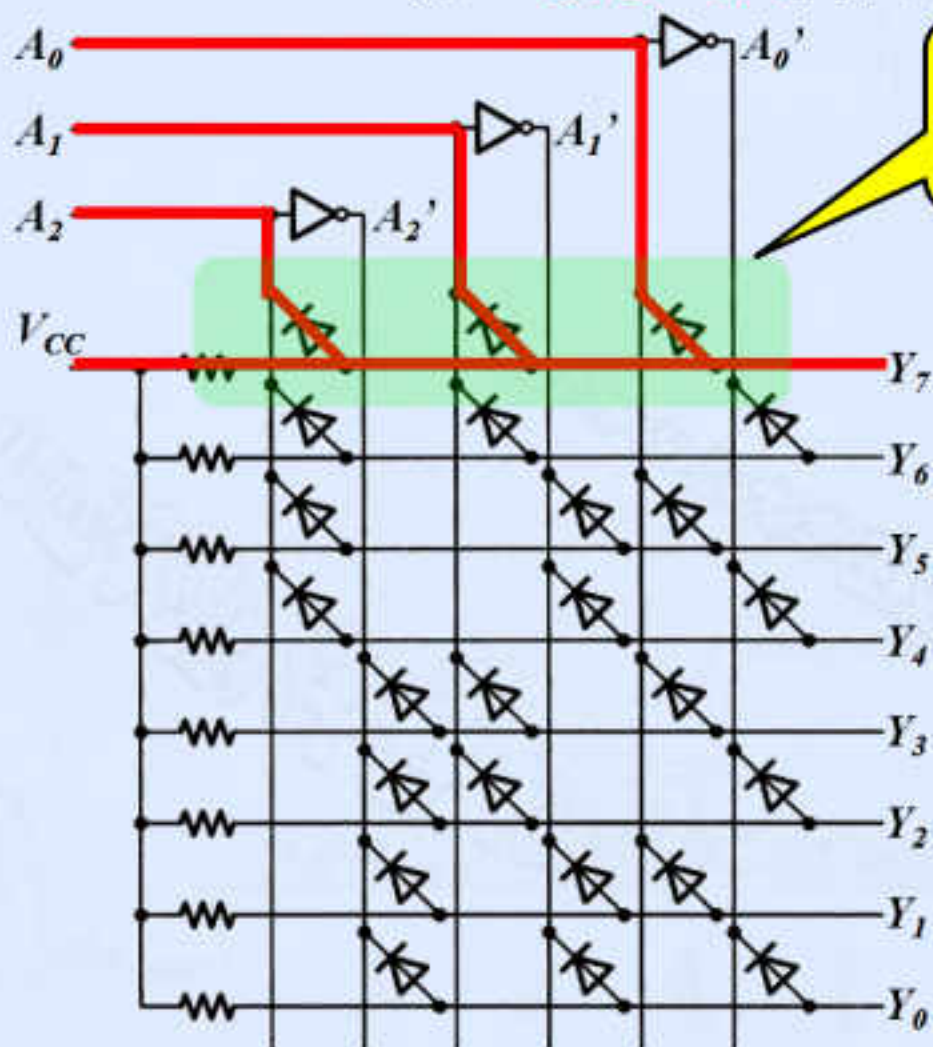
$$Y_5 = A_2 A_1' A_0 = m_5$$

$$Y_6 = A_2 A_1 A_0' = m_6$$

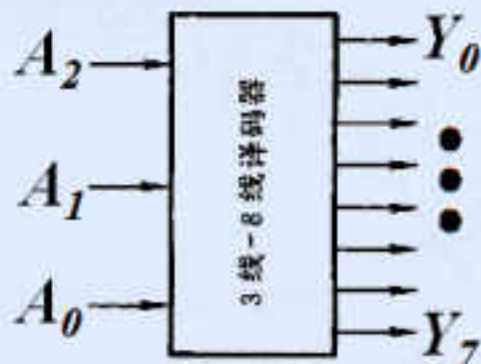
$$Y_7 = A_2 A_1 A_0 = m_7$$

画电路

用二极管与门阵列组成的3线—8线译码器



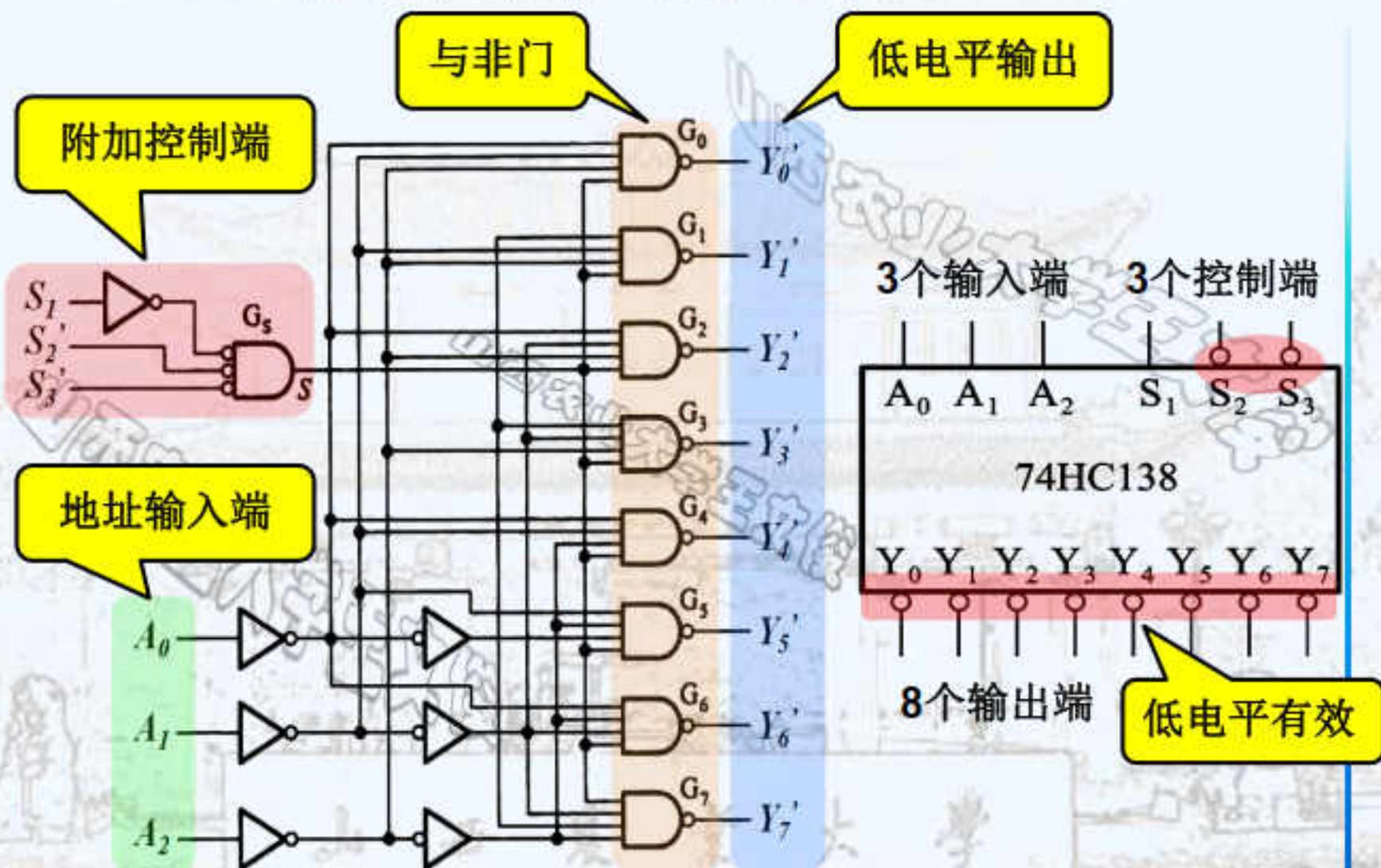
$A_2 A_1 A_0$ 同时为高电平时，
输出高电平



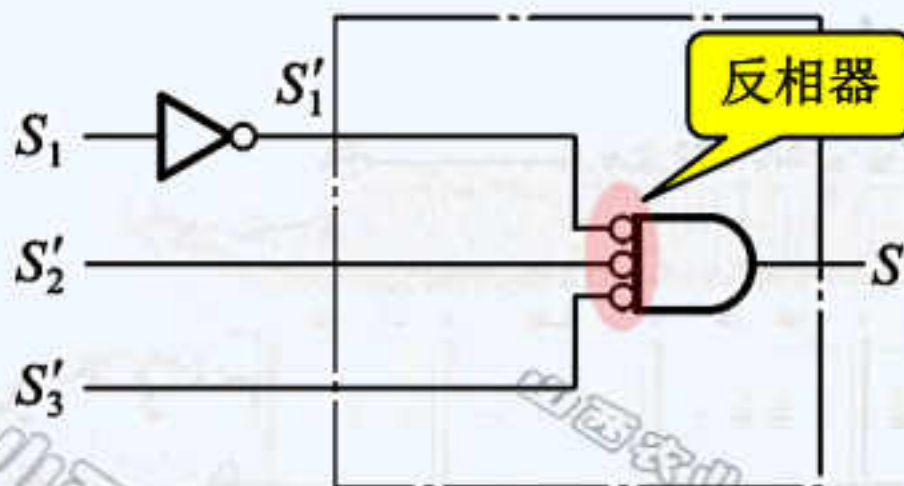
二极管门电路的缺点

- 输入电阻低
- 输出电阻高
- 高低电平偏移

3、CMOS门电路组成的二进制译码器74HC138

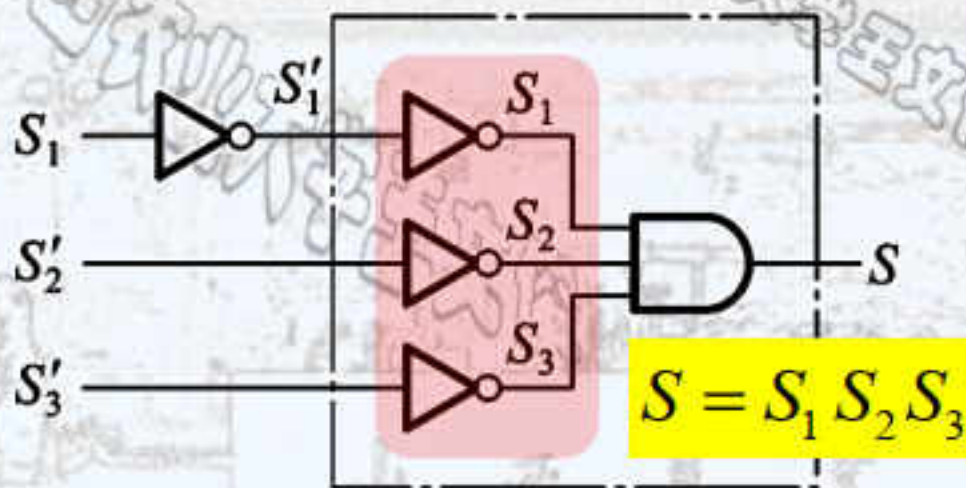


附加控制端

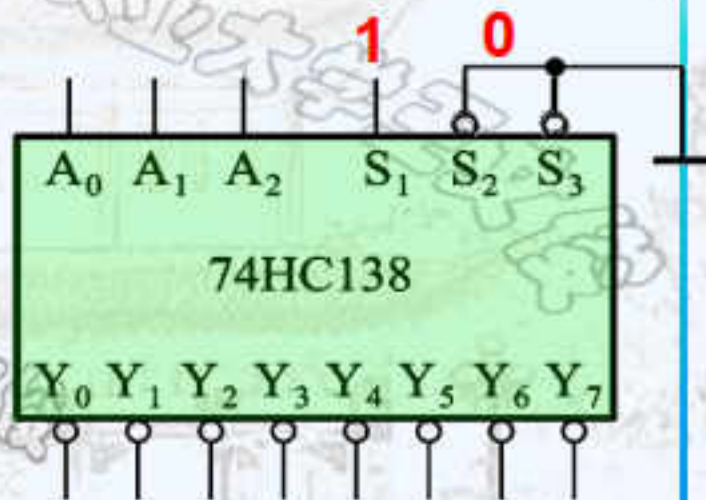


$$S_1 = 1, S_2' = 0, S_3' = 0$$

$S = 1$, 正常工作
正常译码功能



$$S = S_1 S_2 S_3$$



其他控制输入组合

$S = 0$, 禁止工作
输出锁定高电平

74HC138功能表

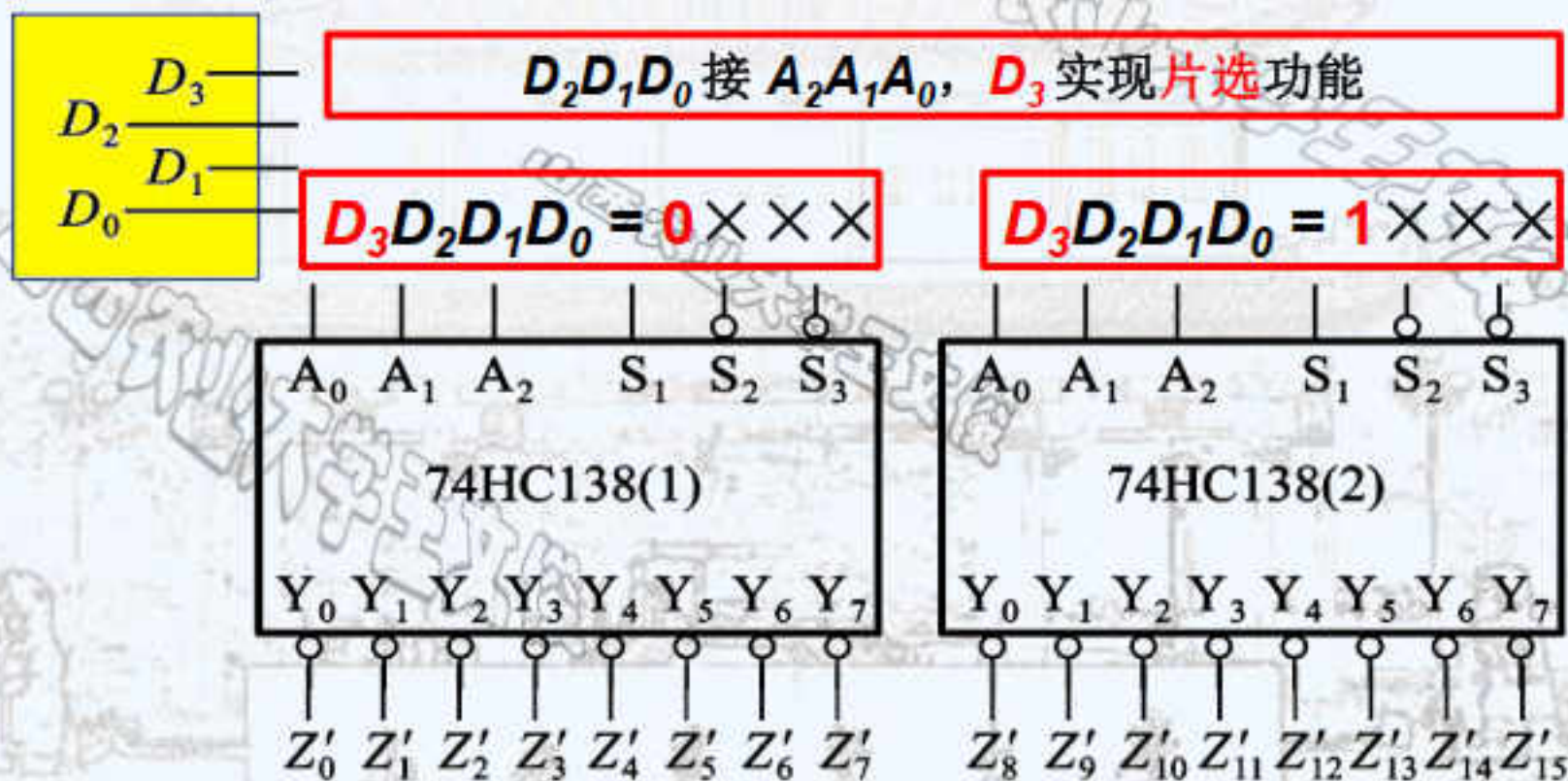
输 入					输 出							
S_1	$S_2'+S_3'$	A_2	A_1	A_0	Y_7'	Y_6'	Y_5'	Y_4'	Y_3'	Y_2'	Y_1'	Y_0'
0	×	×	×	×	1	1	1	1	1	1	1	1
×	1	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

不工作
输出均为
高电平

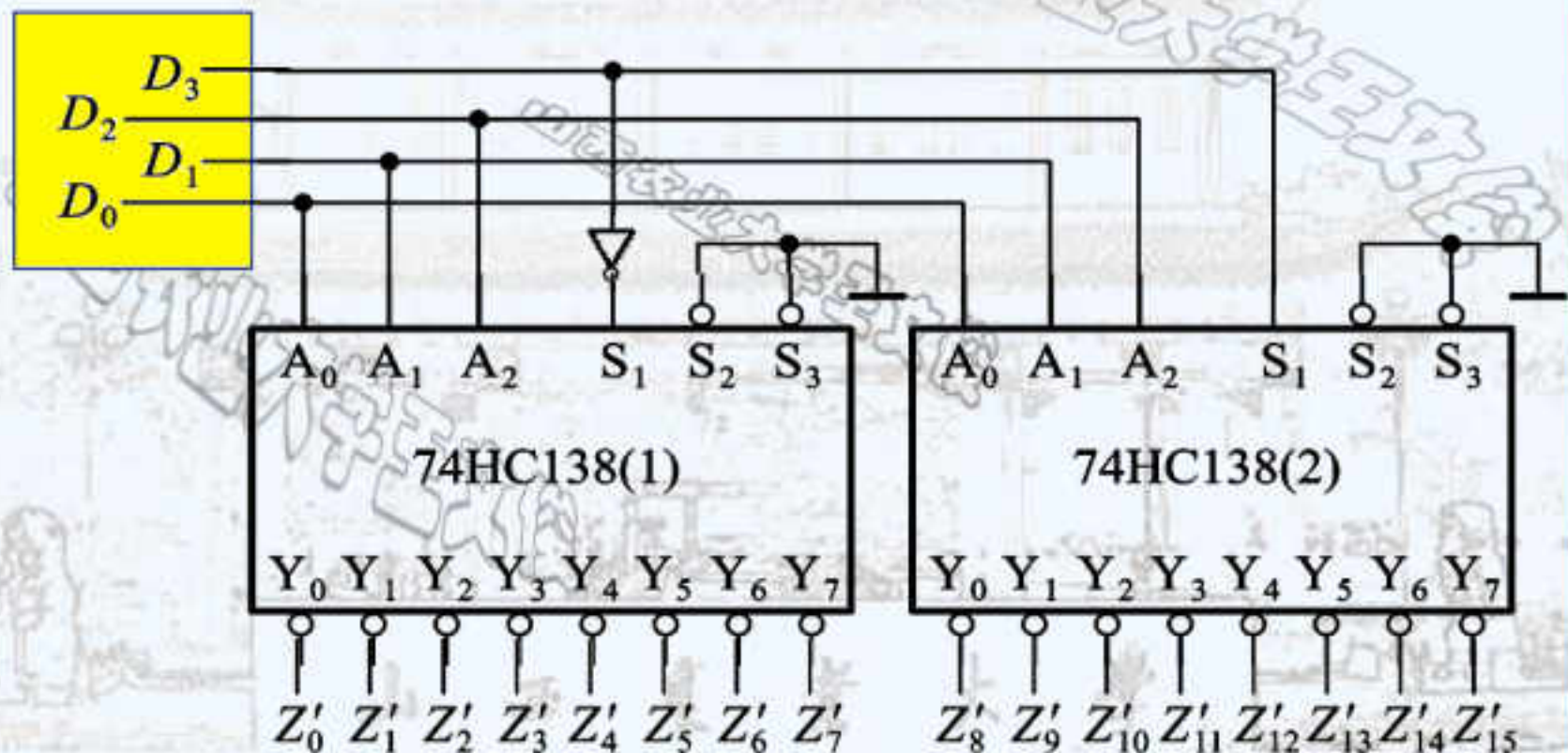
工作
正常译码
功能

- 例4：用3线-8线译码器74HC138组成4线-16线译码器。

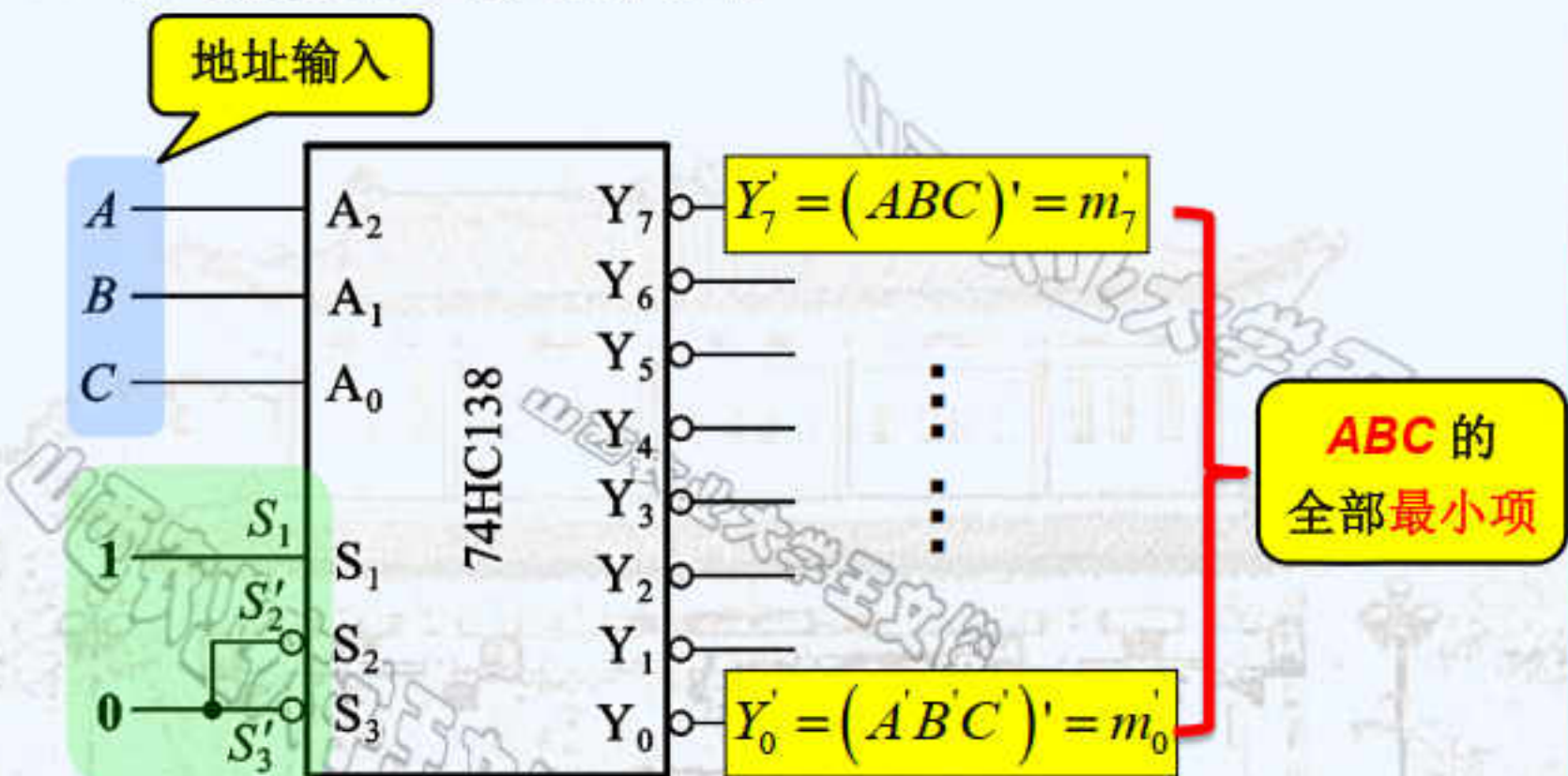
将4位二进制代码 $D_3D_2D_1D_0$ 译成16个独立的低电平信号 $Z'_0 \sim Z'_{15}$



D_3 的值	工作状态	控制输入 S_1	S_1 与 D_3 的关系
0	1号工作	1	D_3'
	2号禁止	0	D_3
1	1号截止	0	D_3'
	2号工作	1	D_3

1号 $S_1 = D_3'$ 2号 $S_1 = D_3$ 

4、译码器设计组合逻辑电路



正常工作

输出为 ABC 三个变量的全部最小项的译码输出，称为**最小项译码器**。

逻辑抽象

- 定变量、明含义、列表格

逻辑函数式

- 写函数

选器件

- 译码器地址输入端个数 $M \geq$ 逻辑函数变量个数 n

函数的化简或变换

- 逻辑函数变换为最小项之和的形式

逻辑电路图

- 画电路，将输出端对应的最小项相加

- 例5: 利用74HC138设计一个多输出的组合逻辑电路

$$\text{输出逻辑函数} \begin{cases} Z_1 = AC' + A'BC + AB'C \\ Z_2 = BC + A'B'C \end{cases}$$

$$Z_1 = AC' + A'BC + AB'C$$

$$Z_2 = BC + A'B'C$$

$$Z_1 = A(B + B')C' + A'BC + AB'C$$

$$Z_2 = (A + A')BC + A'B'C$$

$$Z_1 = \frac{A'BC}{011} + \frac{AB'C'}{100} + \frac{AB'C}{101} + \frac{ABC'}{110}$$

$$Z_2 = \frac{A'B'C}{001} + \frac{A'BC}{011} + \frac{ABC}{111}$$

$$Z_1 = m_3 + m_4 + m_5 + m_6$$

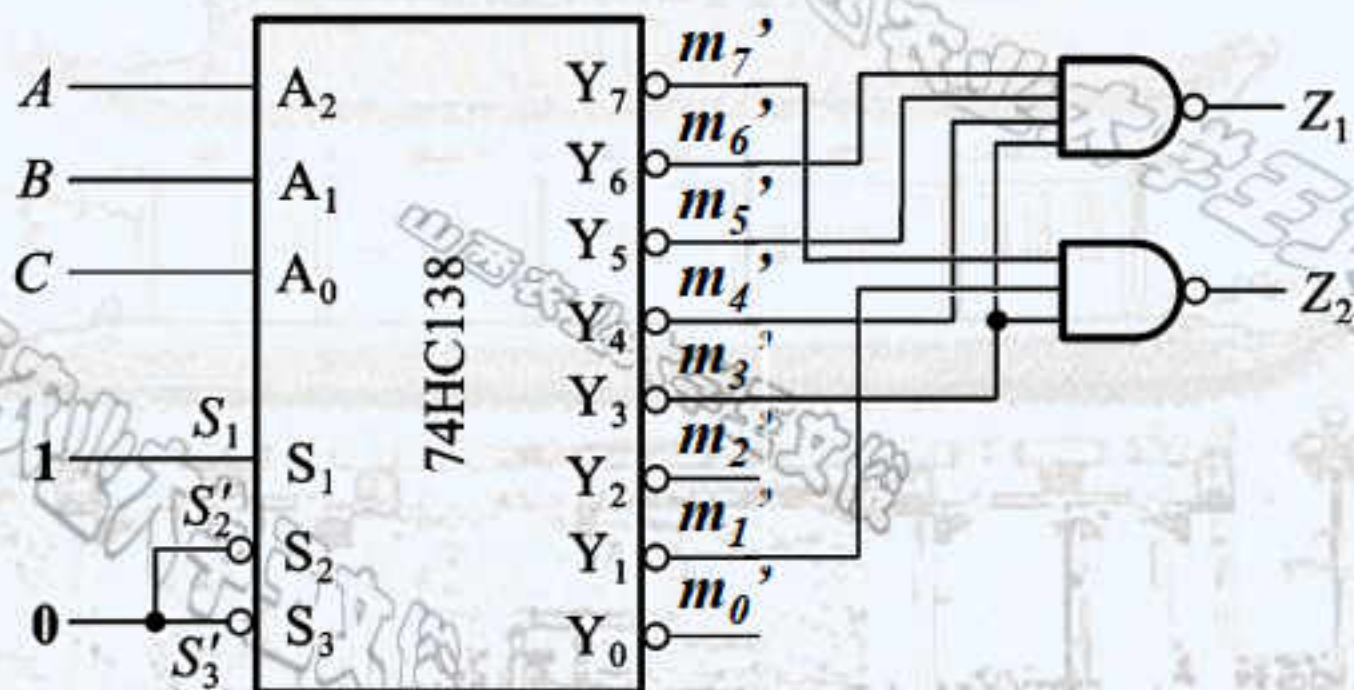
$$Z_2 = m_1 + m_3 + m_7$$

$$Z_1 = (m_3' m_4' m_5' m_6')'$$

$$Z_2 = (m_1' m_3' m_7')'$$

$$Z_1 = (m_3' m_4' m_5' m_6')'$$

$$Z_2 = (m_1' m_3' m_7')'$$



- 例6：设计一个数值比较电路
 - 比较两个二进制数 $A(a_1a_0)$ 和 $B(b_1b_0)$
 - 要求分别给出 $A - B \geq 2$ 、 $B - A \geq 2$ 和 $|A - B| < 2$ 的输出信号

定变量

- 输入变量： a_1 、 a_0 、 b_1 、 b_0
- 输出变量： $Z_1(A-B \geq 2)$ 、 $Z_2(B-A \geq 2)$ 、 $Z_3(|A-B| < 2)$

明含义

- 输入变量：有输入为 1，无输入为 0
- 输出变量：1 为有输出，0 为无输出

列表格

输 入				输 出			输 入				输 出		
a_1	a_0	b_1	b_0	Z_1	Z_2	Z_3	a_1	a_0	b_1	b_0	Z_1	Z_2	Z_3
0	0	0	0	0	0	1	1	0	0	0	1	0	0
0	0	0	1	0	0	1	1	0	0	1	0	0	1
0	0	1	0	0	1	0	1	0	1	0	0	0	1
0	0	1	1	0	1	0	1	0	1	1	0	0	1
0	1	0	0	0	0	1	1	1	0	0	1	0	0
0	1	0	1	0	0	1	1	1	0	1	1	0	0
0	1	1	0	0	0	1	1	1	1	0	0	0	1
0	1	1	1	0	1	0	1	1	1	1	0	0	1

做变换
写函数

$$Z_1 = m_8 + m_{12} + m_{13} = (m'_8 m'_{12} m'_{13})'$$

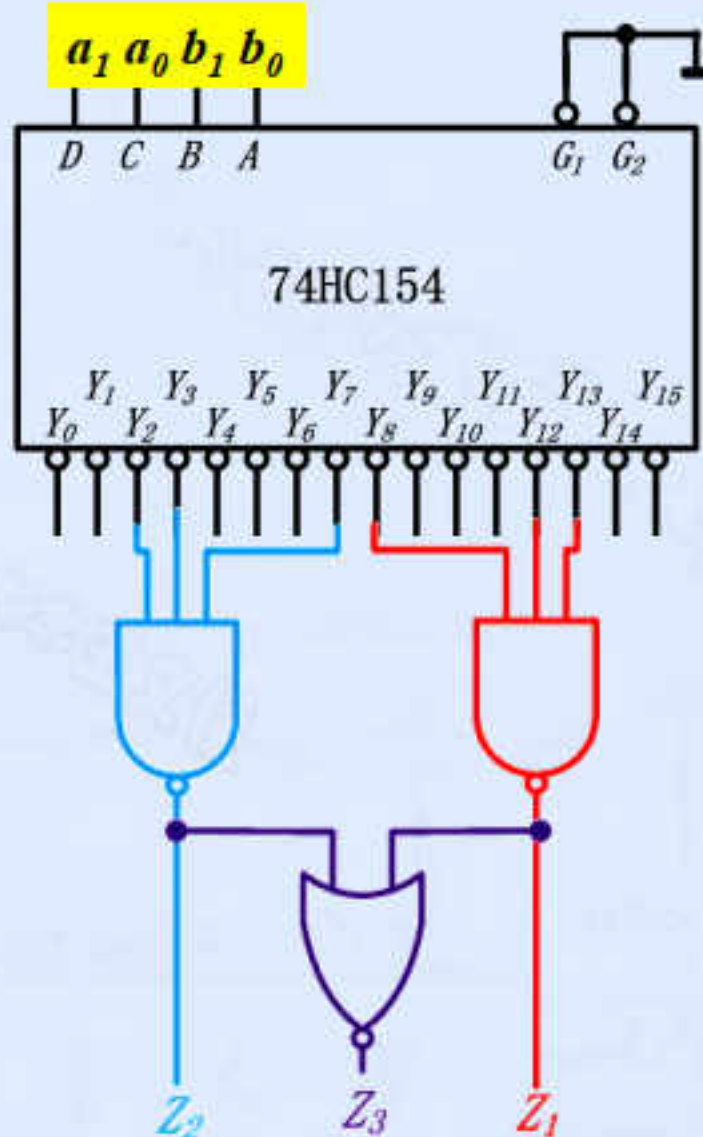
$$Z_3 = (Z_1 + Z_2)'$$

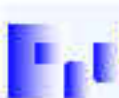
$$Z_2 = m_2 + m_3 + m_7 = (m'_2 m'_3 m'_7)'$$

选器件、画电路

4线-16线译码器

$$\begin{cases} Z_1 = (m_8' m_{12}' m_{13}')' \\ Z_2 = (m_2' m_3' m_7')' \\ Z_3 = (Z_1 + Z_2)' \end{cases}$$

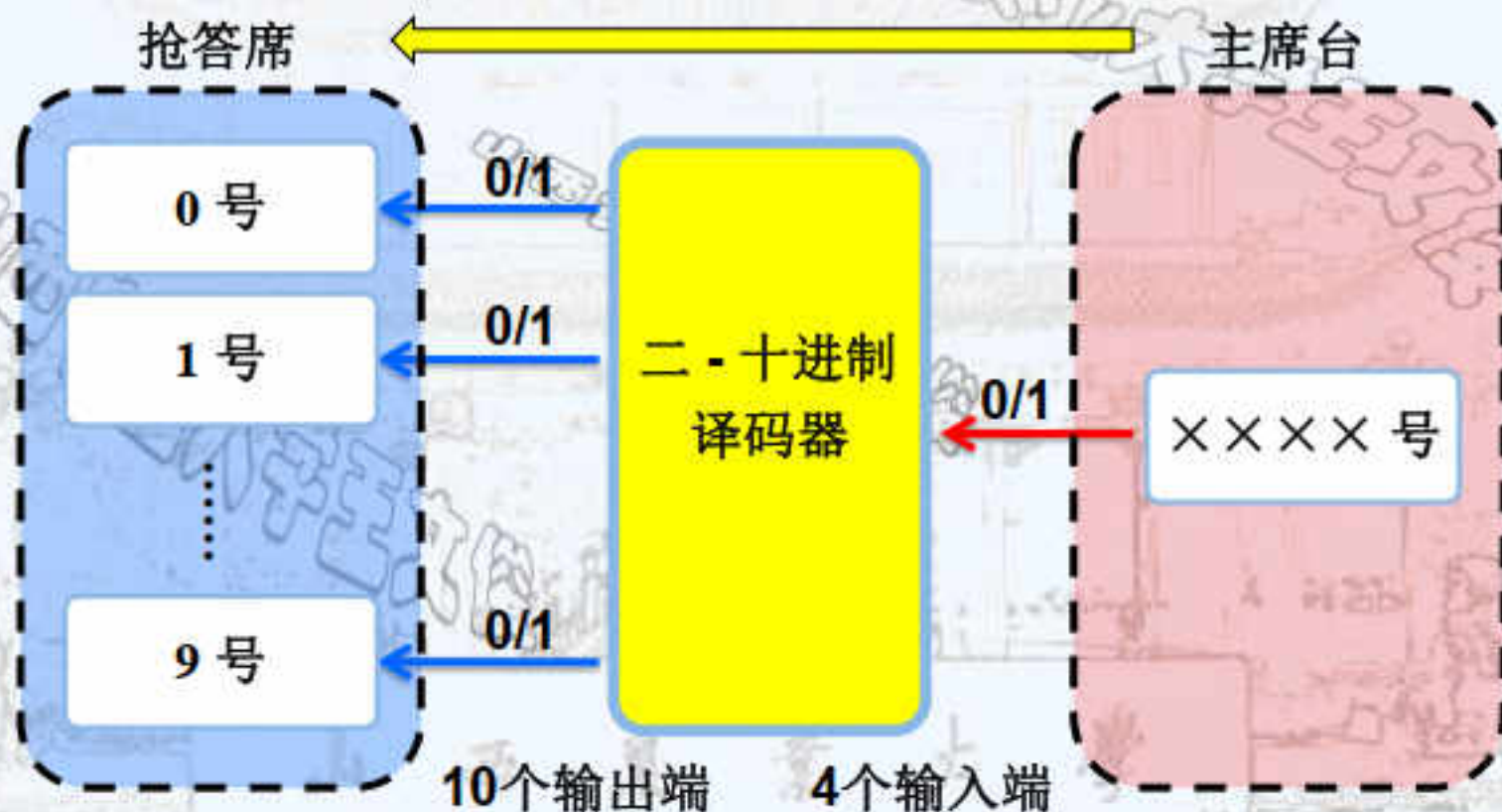




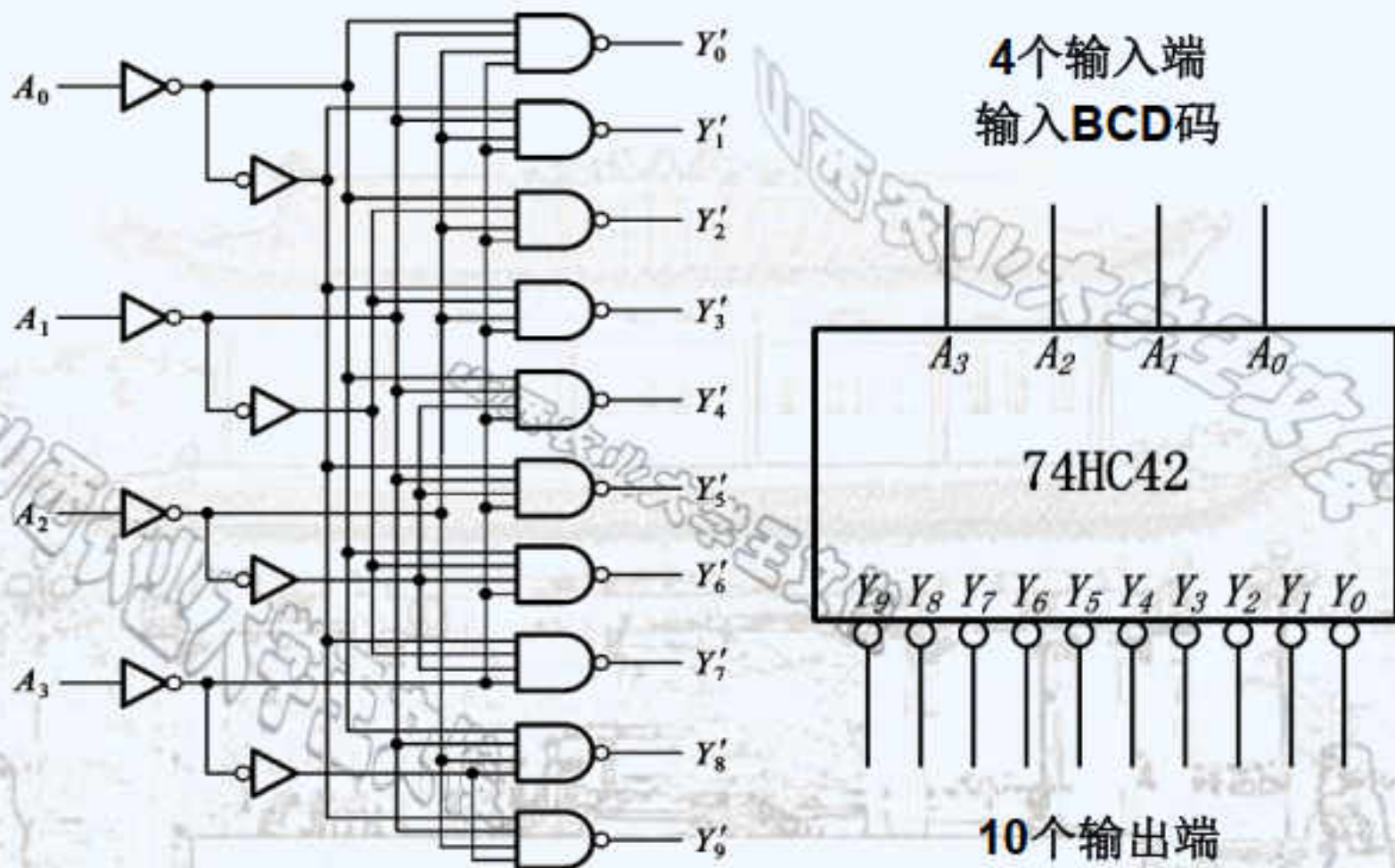
八、二-十进制译码器

• 1、引言

将输入的**BCD码**的10个代码译成10个高低电平输出信号。



• 2、二进制译码器实例74HC42



拒绝伪码：输入BCD码以外伪码（1010 ~ 1111），输出**均为高电平**。

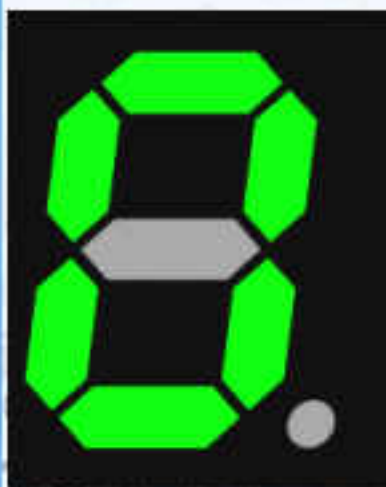
九、显示译码器

• 1、七段字符显示器

用于以**十进制数码**的形式，直观的显示数字系统内的运行数据。

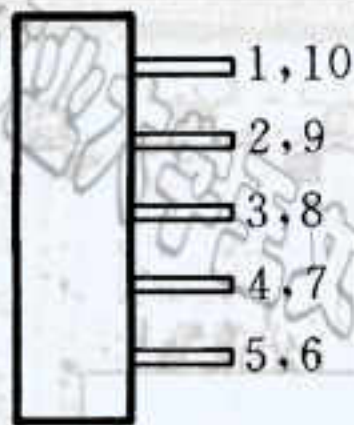
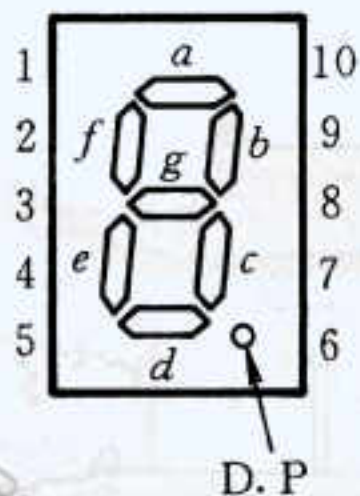


半导体（LED）数码管

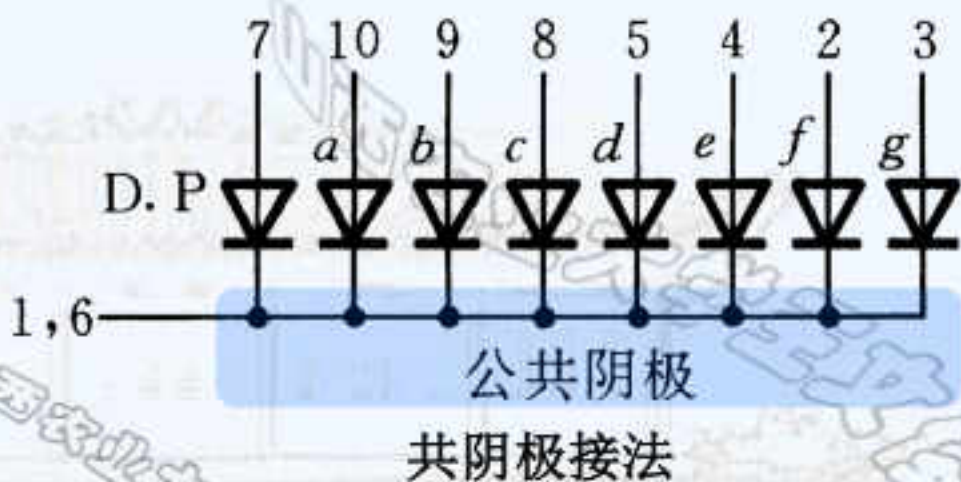


液晶显示器（LCD）

2、半导体数码管BS201A

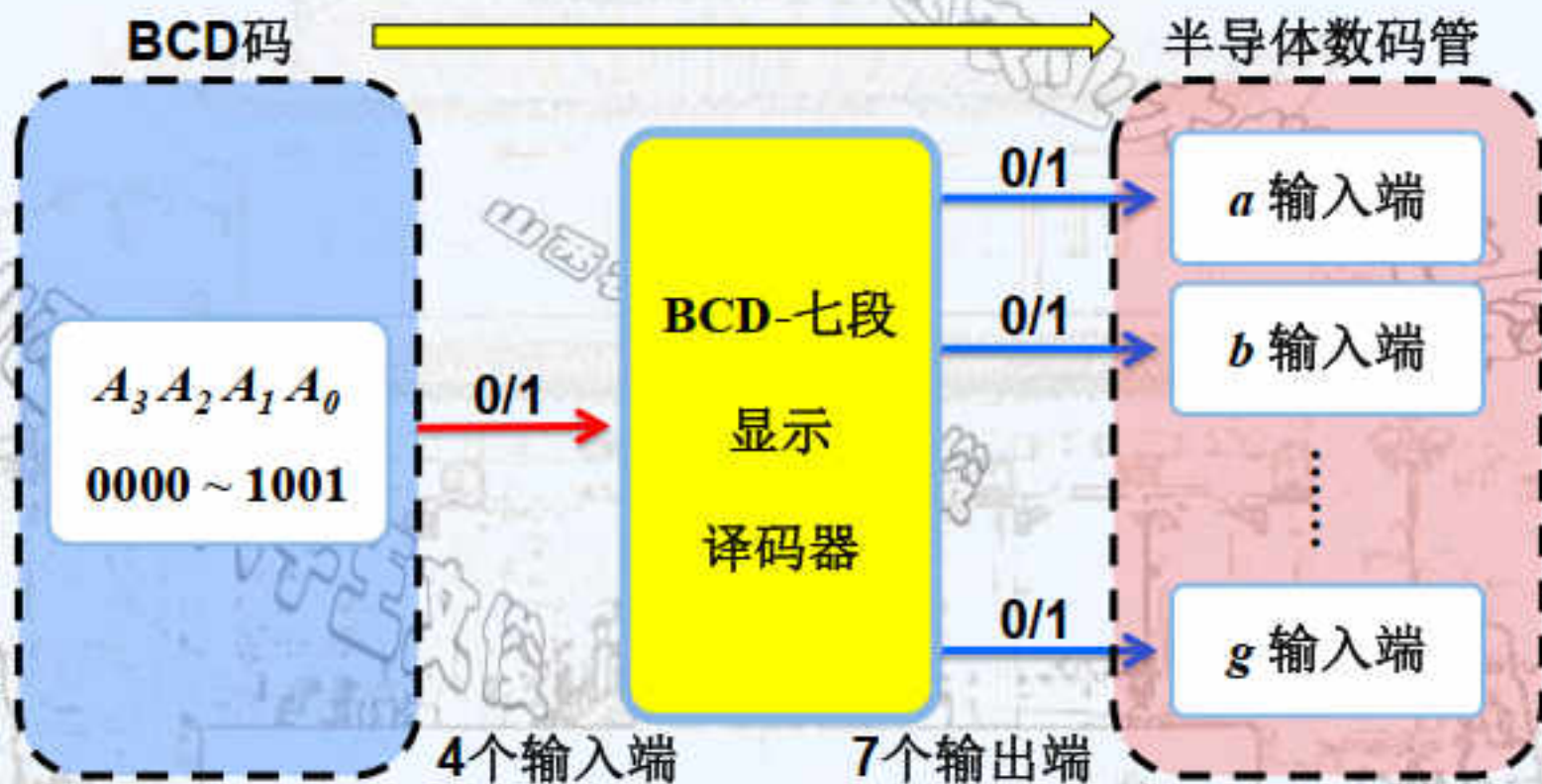


BS201A 外形图



3、BCD-七段显示译码器的设计

显示译码器作用：将**BCD代码**译成数码管所需**驱动信号**。



定变量

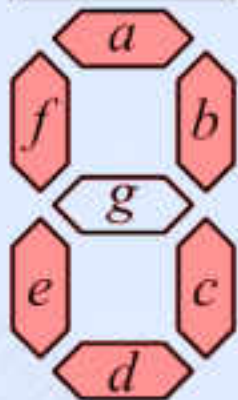
- 输入变量：四位二进制数 $A_3 A_2 A_1 A_0$
- 输出变量： $Y_a(a)$ 、 $Y_b(b)$ 、、 $Y_g(g)$

明含义

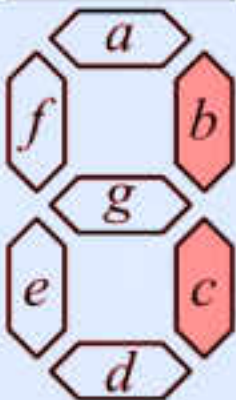
- 输入变量：有输入为 1，无输入为 0
- 输出变量：1 为有输出，0 为无输出

列表格

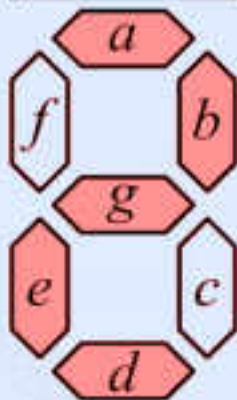
0 0 0 0



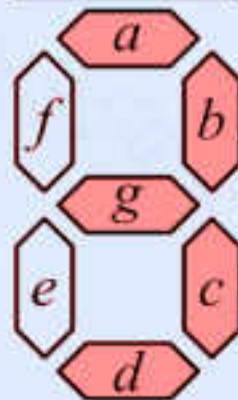
0 0 0 1



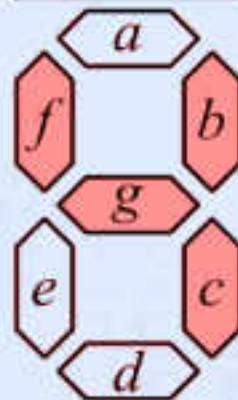
0 0 1 0



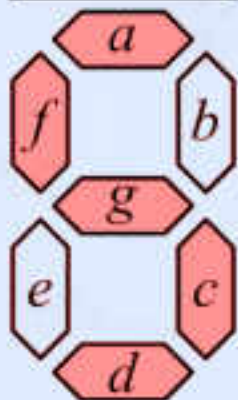
0 0 1 1



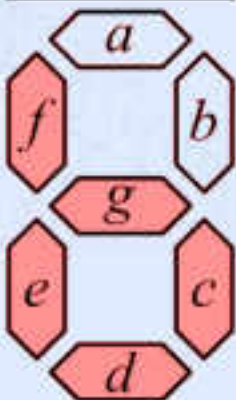
0 1 0 0



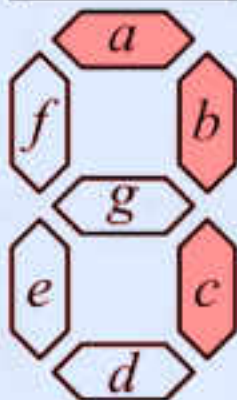
0 1 0 1



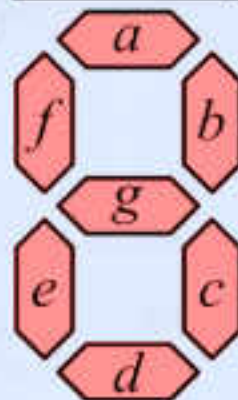
0 1 1 0



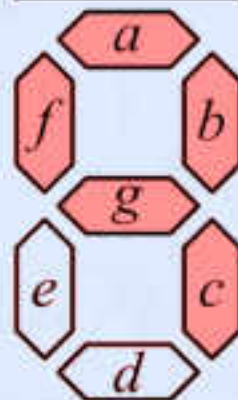
0 1 1 1



1 0 0 0



1 0 0 1



输 入					输 出						
数字	A_3	A_2	A_1	A_0	Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	0	0	0	1	1	0	1
11	1	0	1	1	0	0	1	1	0	0	1
12	1	1	0	0	0	1	0	0	0	1	1
13	1	1	0	1	1	0	0	1	0	1	1
14	1	1	1	0	0	0	0	1	1	1	1
15	1	1	1	1	0	0	0	0	0	0	0

正常输入

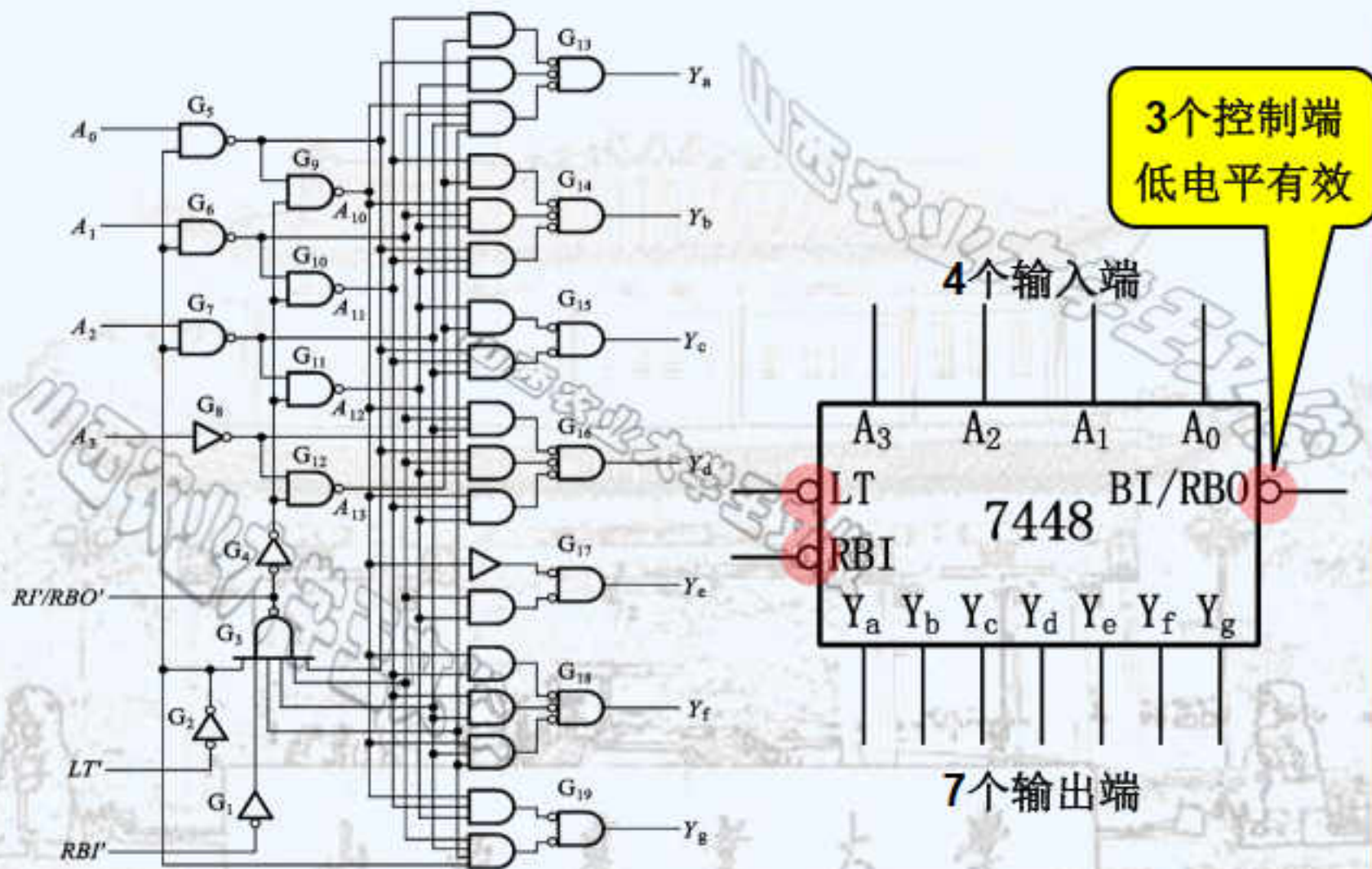
伪码输入

写函数

卡诺图上采用“**合并0后求反**”的方法，获得 $Y_a \sim Y_g$ 的**与或非**形式

$$\left\{ \begin{array}{l} Y_a = (A_3' A_2' A_1' A_0 + A_3 A_1 + A_2 A_0)' \\ Y_b = (A_3 A_1 + A_2 A_1 A_0' + A_2 A_1' A_0)' \\ Y_c = (A_3 A_2 + A_2' A_1 A_0)' \\ Y_d = (A_2 A_1 A_0 + A_2 A_1' A_0' + A_2' A_1' A_0)' \\ Y_e = (A_2 A_1' + A_0)' \\ Y_f = (A_3' A_2' A_0 + A_2' A_1 + A_1 A_0)' \\ Y_g = (A_3' A_2' A_1' + A_2 A_1 A_0)' \end{array} \right.$$

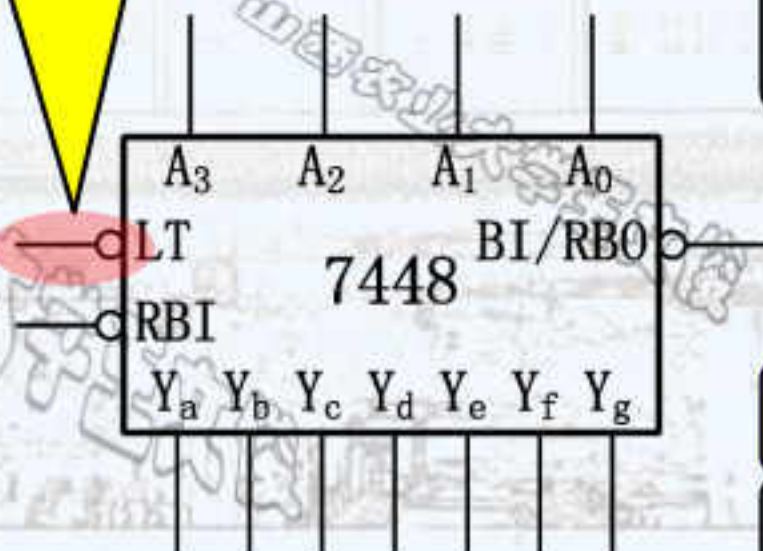
4、BCD-七段显示译码器实例7448



① 灯测试输入 LT' (**L**amp **T**est)

用于检查数码管各段能否**正常发光**。

灯测试输入 LT'
低电平有效



$LT' = 0$

$Y_a \sim Y_g$ 全为高电平
七段数码管**同时点亮**

$LT' = 1$

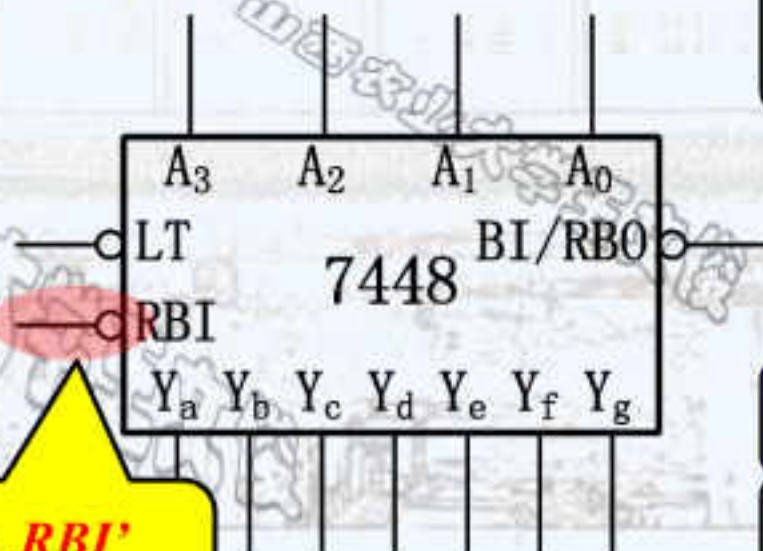
不起作用

② 灭零输入 RBI' (*Ripple Blanking Input*)

当数码管显示0时，用于把不希望显示的 **0** 熄灭。

$RBI' = 0$

当数码管显示为0时，
该0熄灭。



灭零输入 RBI'
低电平有效

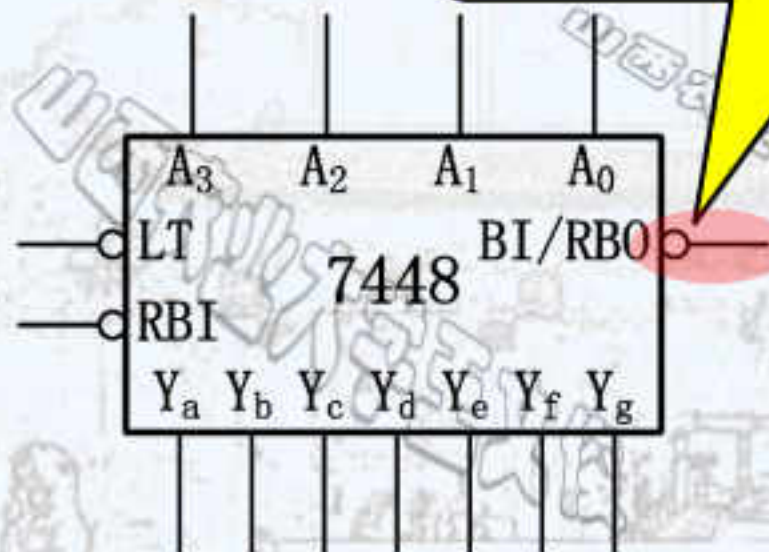
$RBI' = 1$

不起作用

③ 灭零输出 RBO' (**R**ipple **B**lanking **O**utput)

当数码管显示 0，且灭零输入 $RBI = 0$ 时，用于指示是否灭零成功

灭零输出 RBO'
低电平有效



$RBO' = 0$

灭零成功

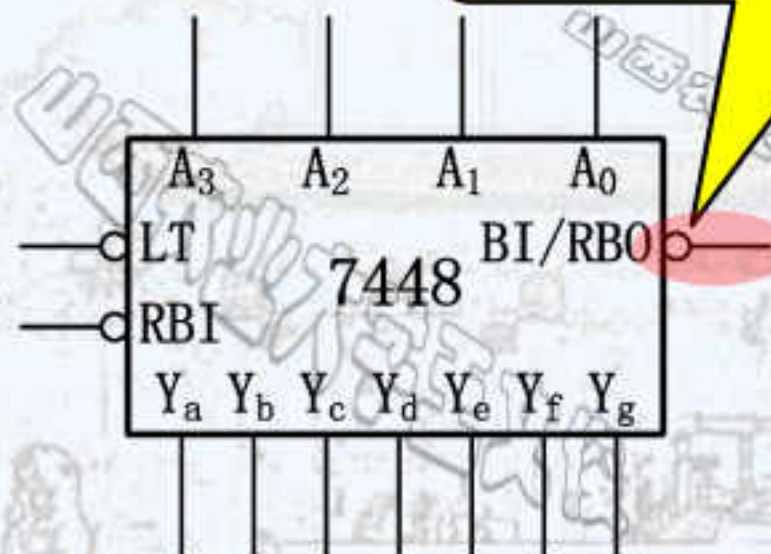
$RBO' = 1$

灭零失败

④ 灭灯输入 BI' (**B**lanking **I**nput)

用于**熄灭**数码管。

灭灯输入 BI'
低电平有效



$BI' = 0$

无论何种输入状态，
数码管均**熄灭**。

$BI' = 1$

不起作用

• 5、7448驱动BS201A

