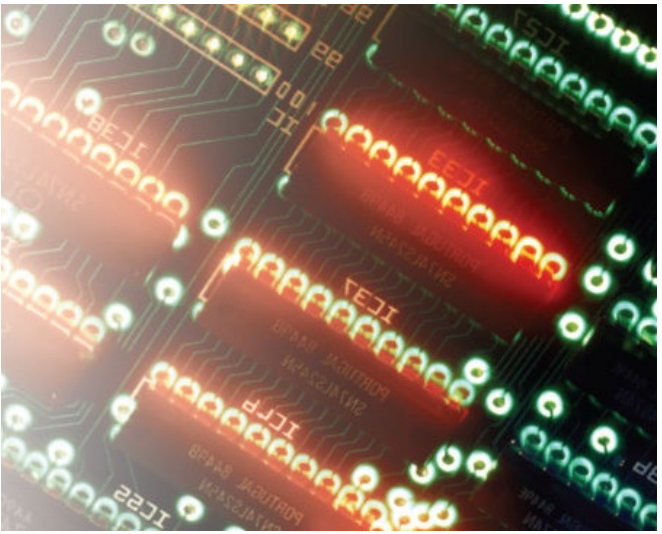
浙江大学
ZHEJIANG UNIVERSITY

组合逻辑1



刘 鹏

浙江大学

信息与电子工程学院

Email: liupeng@zju.edu.cn

复习

- 逻辑化简
  - 卡诺图

本节内容

- 组合电路概念
- 组合电路设计方法
- 组合电路模块：编码器和译码器

基本公式

- 根据与、或、非的定义，得布尔恒等式

序号	公　　式	序号	公　　式
		10	$1' = 0;\ 0' = 1$
1	$0 \cdot A = 0$	11	$1 + A = 1$
2	$1 \cdot A = A$	12	$0 + A = A$
3	$A \cdot A = A$	13	$A + A = A$
4	$A \cdot A' = 0$	14	$A + A' = 1$
5	$A \cdot B = B \cdot A$	15	$A + B = B + A$
6	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	16	$A + (B + C) = (A + B) + C$
7	$A \cdot (B + C) = A \cdot B + A \cdot C$	17	$A + B \cdot C = (A + B)(A + C)$
8	$(A \cdot B)' = A' + B'$	18	$(A + B)' = A'B'$
9	$(A')' = A$		

最小项的编号

最小项	取值	对应	编号
	A B C	十进制数	
$A'B'C'$	0 0 0	0	m_0
$A'B'C$	0 0 1	1	m_1
$A'BC'$	0 1 0	2	m_2
$A'BC$	0 1 1	3	m_3
$AB'C'$	1 0 0	4	m_4
$AB'C$	1 0 1	5	m_5
ABC'	1 1 0	6	m_6
ABC	1 1 1	7	m_7

最大项的编号

最大项	取值	对应	编号
	$A\ B\ C$	十进制数	
$A'+B'+C'$	1 1 1	7	M_7
$A'+B'+C$	1 1 0	6	M_6
$A'+B+C'$	1 0 1	5	M_5
$A'+B+C$	1 0 0	4	M_4
$A+B'+C'$	0 1 1	3	M_3
$A+B'+C$	0 1 0	2	M_2
$A+B+C'$	0 0 1	1	M_1
$A+B+C$	0 0 0	0	M_0

卡诺图化简法

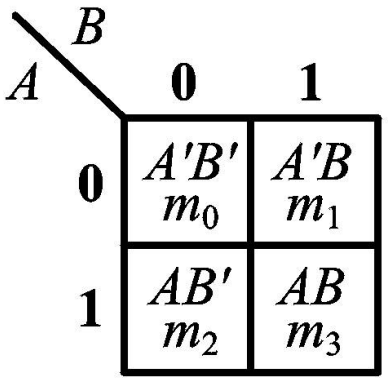
逻辑函数的卡诺图表示法

- 实质：将逻辑函数的最小项之和的以图形的方式表示出来

- 以2^n个小方块分别代表 n 变量的所有最小项，并将它们排列成矩阵，而且使**几何位置相邻**的两个最小项在**逻辑上也是相邻的**（只有一个变量不同），就得到表示n变量全部最小项的卡诺图

表示最小项的卡诺图

- 2变量卡诺图



- 3变量的卡诺图

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

- 4变量的卡诺图

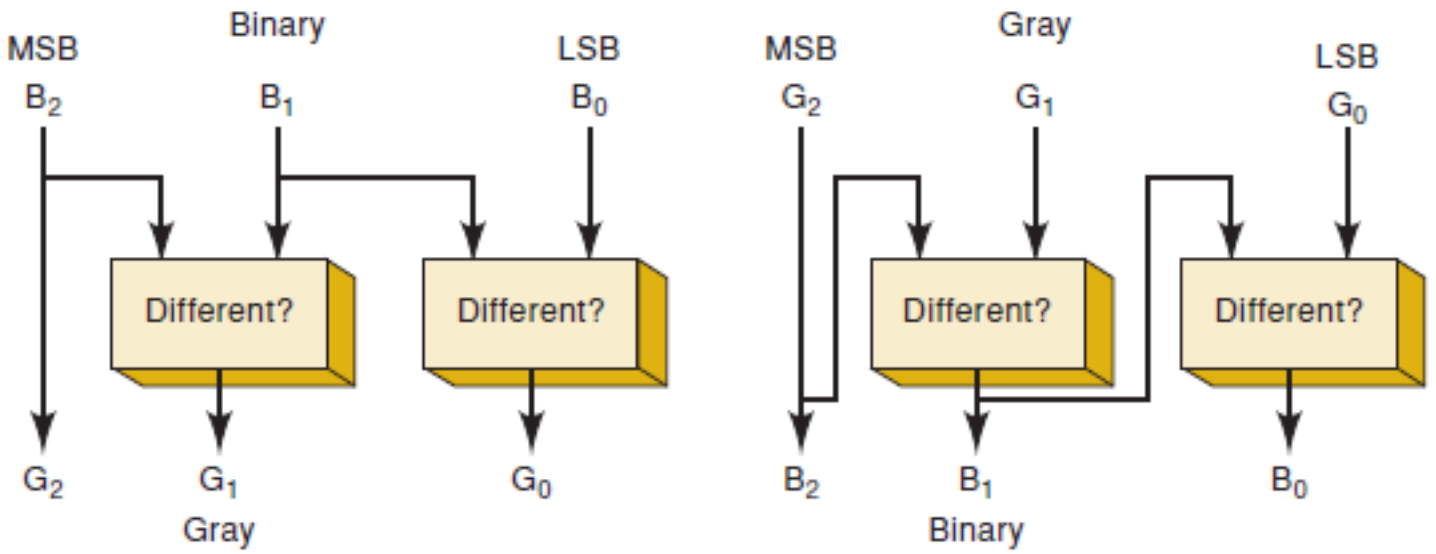
AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

用卡诺图化简函数

- 依据：具有相邻性的最小项可合并，消去不同因子
- 在卡诺图中，最小项的相邻性可以从图形中直观地反映出来
- 合并最小项的原则：
  - 两个相邻最小项可合并为一项，消去一对因子
  - 四个排成矩形的相邻最小项可合并为一项，消去两对因子
  - 八个相邻最小项可合并为一项，消去三对因子

3-bit 二进制码和格雷码（Gray code）

B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0



Decimal	Binary	Hexadecimal	BCD	GRAY
0	0	0	0000	0000
1	1	1	0001	0001
2	10	2	0010	0011
3	11	3	0011	0010
4	100	4	0100	0110
5	101	5	0101	0111
6	110	6	0110	0101
7	111	7	0111	0100
8	1000	8	1000	1100
9	1001	9	1001	1101
10	1010	A	0001 0000	1111
11	1011	B	0001 0001	1110
12	1100	C	0001 0010	1010
13	1101	D	0001 0011	1011
14	1110	E	0001 0100	1001
15	1111	F	0001 0101	1000

组合逻辑的内容

- 组合电路的设计步骤
- 基本组合电路单元
  - 编码器Encoder
  - 译码器Decoder
  - 选择器Multiplexer
  - 比较器Comparator
  - 加法器Adder
  - 乘法器Multiplier（**可选**）
- 电路HDL描述

❑组合逻辑电路的特点

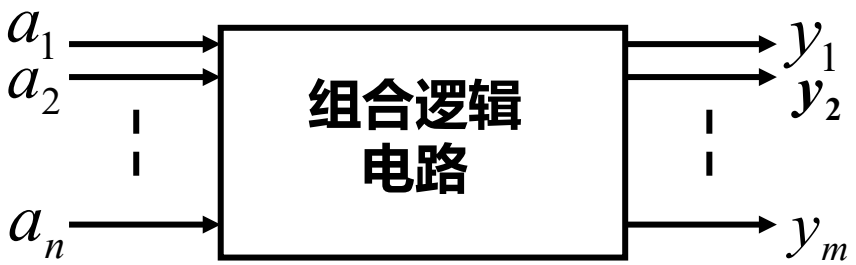
功能

任意时刻的输出仅
取决于该时刻的输入
没有反馈

电路结构

不含存储单元

❑逻辑功能的描述



组合逻辑电路的框图

$$\boldsymbol{Y = F(A)}$$

$$y_1 = f_1(a_1, a_2, \ldots, a_n)$$
$$y_2 = f_2(a_1, a_2, \ldots, a_n)$$

$$y_m = f_m(a_1, a_2, \ldots, a_n)$$

组合逻辑电路的设计方法

一、逻辑抽象

- 分析因果关系，确定输入/输出变量
- 定义逻辑状态的含意（赋值）
- 列出定义输出和输入之间关系的真值表

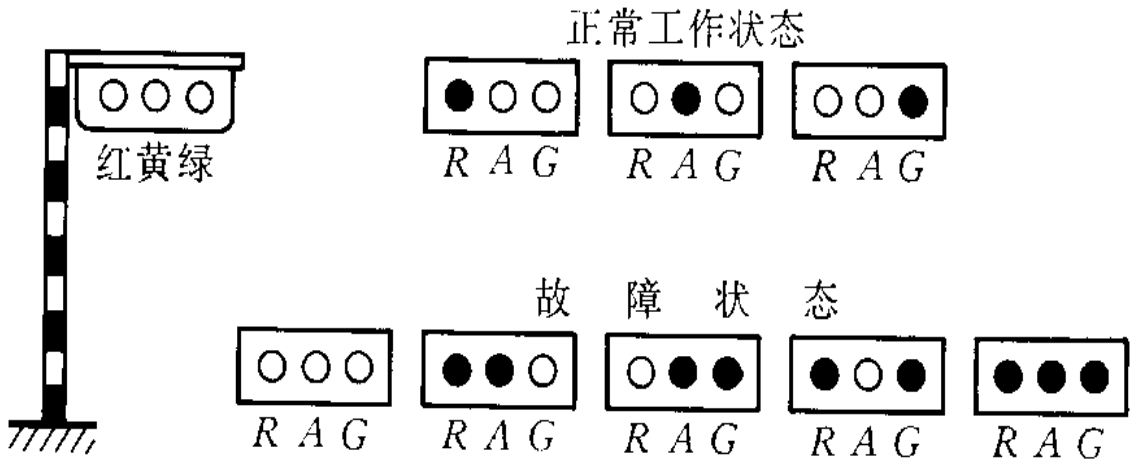
二、写出函数的最简表达式

- 写出每个输出为1的乘积项
- 写出乘积项之和
- 简化逻辑表达式

三、用逻辑门电路或集成电路模块实现表达式

设计举例

- 设计一个监视交通信号灯状态的逻辑电路



设计举例

1. 抽象

- 输入变量:

  红（R）、黄（A）、绿（G）

- 输出变量:

  故障信号（Z）

2. 写出逻辑表达式

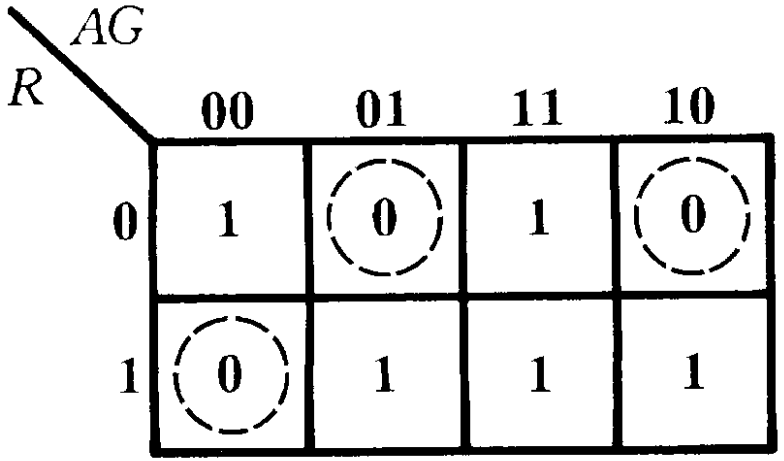
$$Z = R'A'G' + R'AG + RA'G + RAG' + RAG$$

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

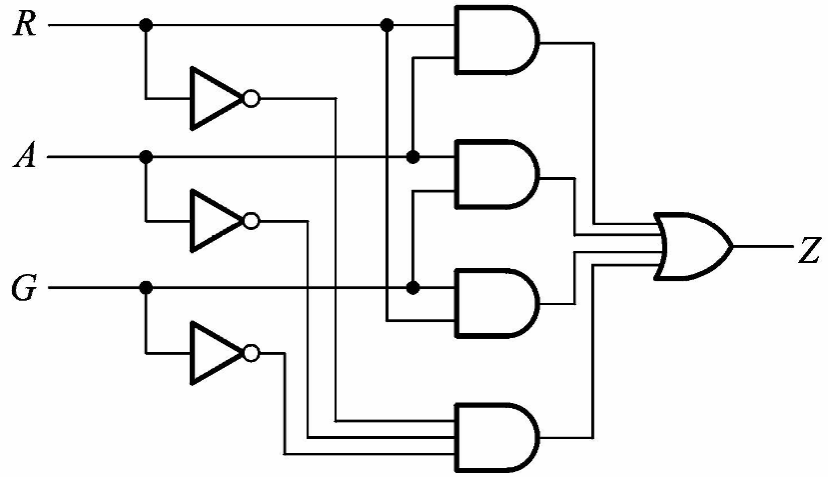
设计举例

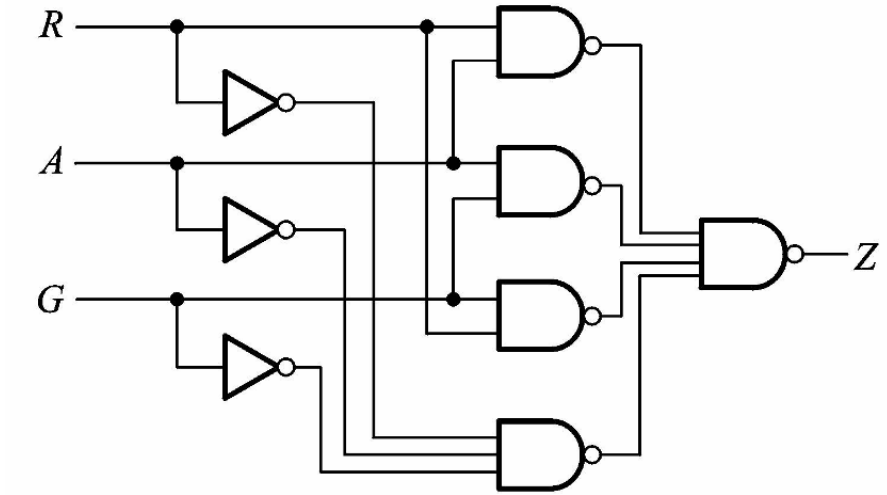
化简

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



3. 画出逻辑图

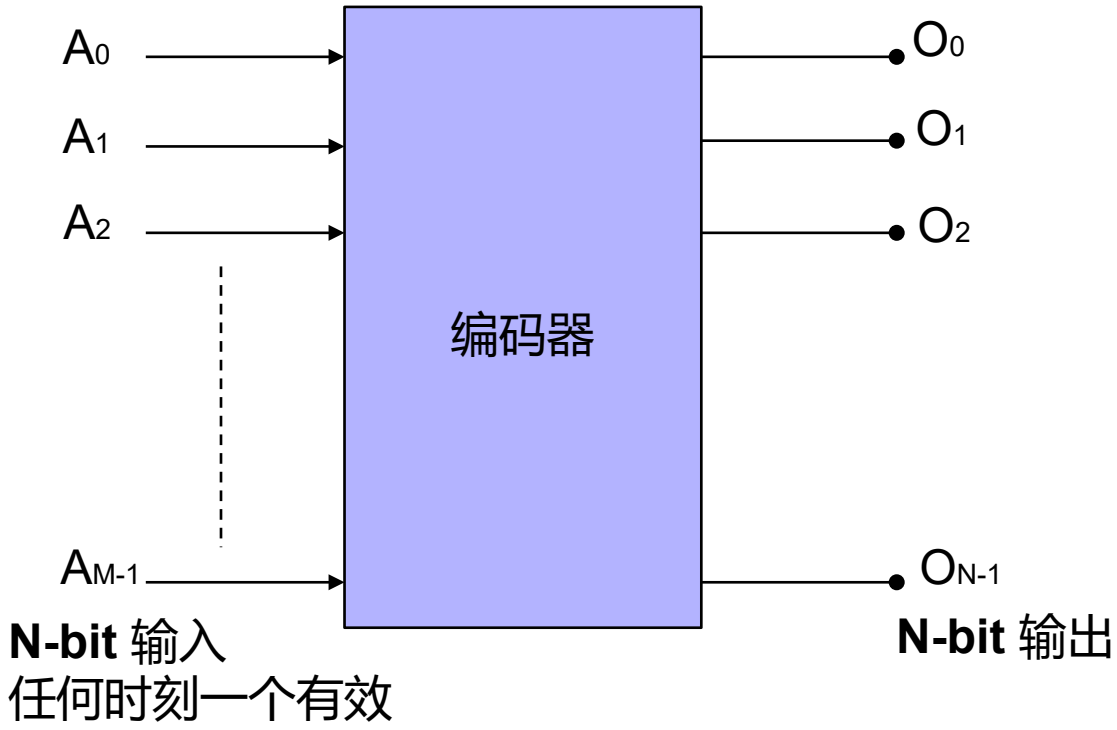




编码器

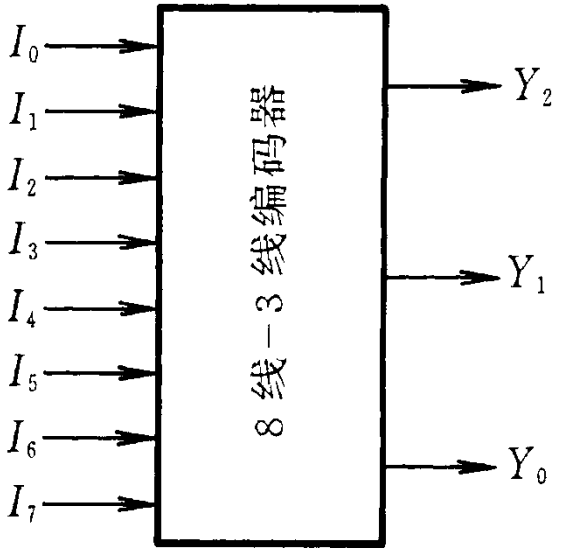
编码：将输入的每个高/低电平信号变成一个对应的二进制代码

- 普通编码器
- 优先编码器



普通编码器

- 特点：任何时刻只允许输入一个编码信号
- 例：3位二进制普通编码器



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

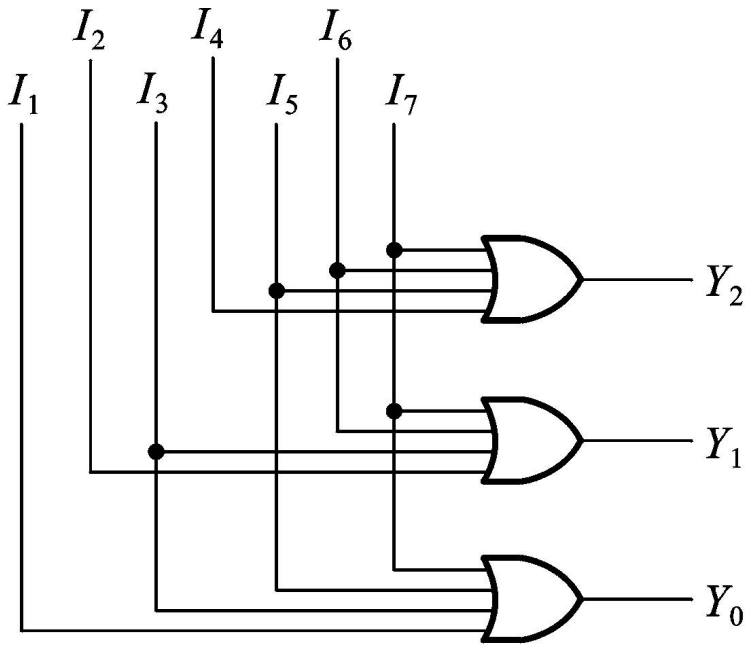
$$Y_2 = I_7' I_6' I_5' I_4 I_3' I_2' I_1' I_0' + I_7' I_6' I_5 I_4' I_3' I_2' I_1' I_0$$
$$+ I_7' I_6 I_5' I_4' I_3' I_2' I_1' I_0' + I_7 I_6' I_5' I_4' I_3' I_2' I_1' I_0$$

利用无关项化简

$$Y_2 = I_4 + I_5 + I_6 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_0 = I_1 + I_3 + I_5 + I_7$$



任何时候只有一个输入时激活的，或有两个输入同时激活，则输入就会产生一个没有定义的组合。对于这个不确定因素，编码器必须建立优先机制，使得只有一个输出被编码

优先编码器

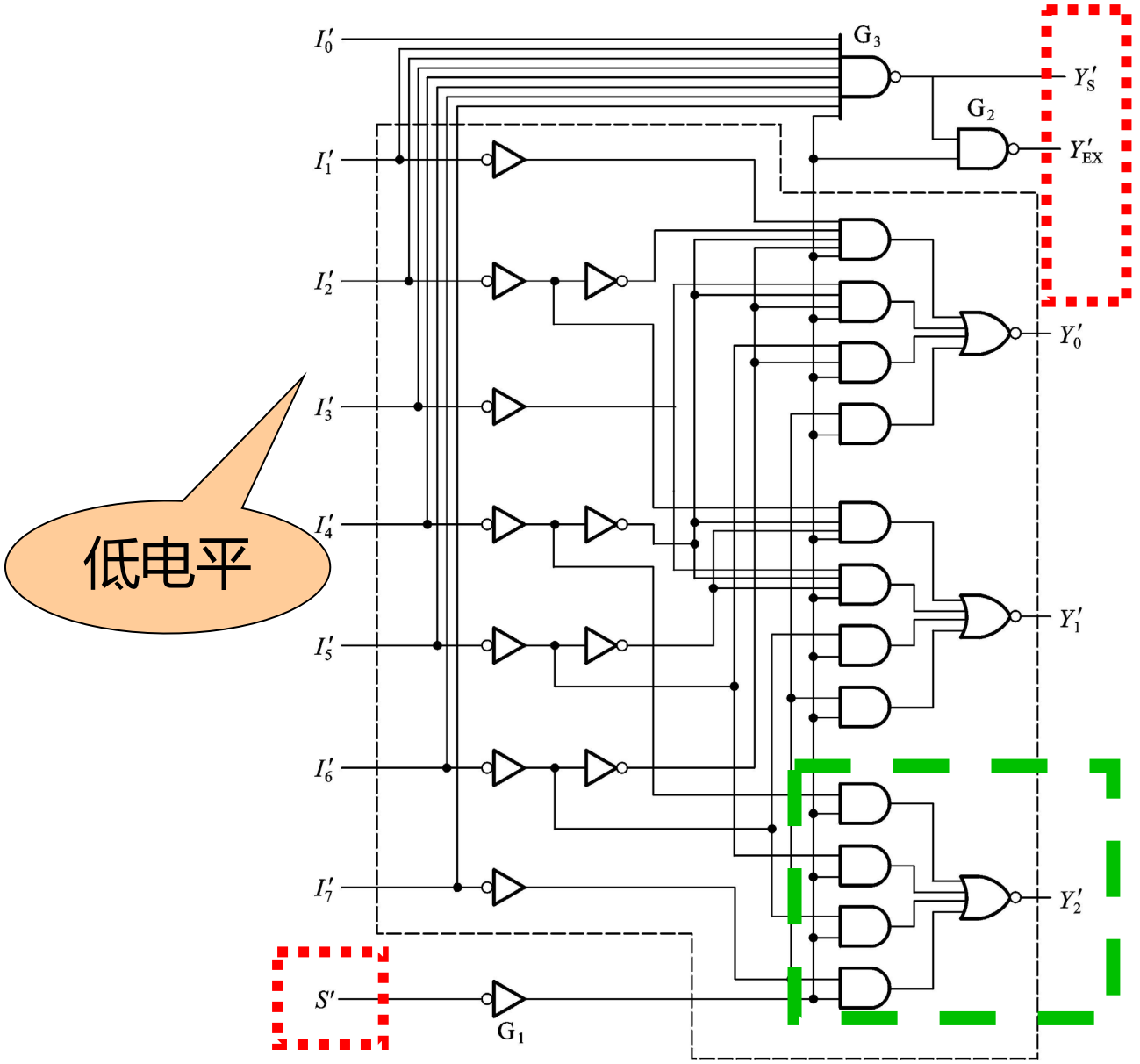
- 特点：允许同时输入两个以上的编码信号，但只对其中优先权最高的一个进行编码
- 例：8线-3线优先编码器
- 设I_7优先权最高...I_0优先权最低

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
X	X	X	X	X	X	X	1	1	1	1
X	X	X	X	X	X	1	0	1	1	0
X	X	X	X	X	1	0	0	1	0	1
X	X	X	X	1	0	0	0	1	0	0
X	X	X	1	0	0	0	0	0	1	1
X	X	1	0	0	0	0	0	0	1	0
X	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0

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

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

$$Y_2 = I_7 + I_6 + I_5 + I_4$$



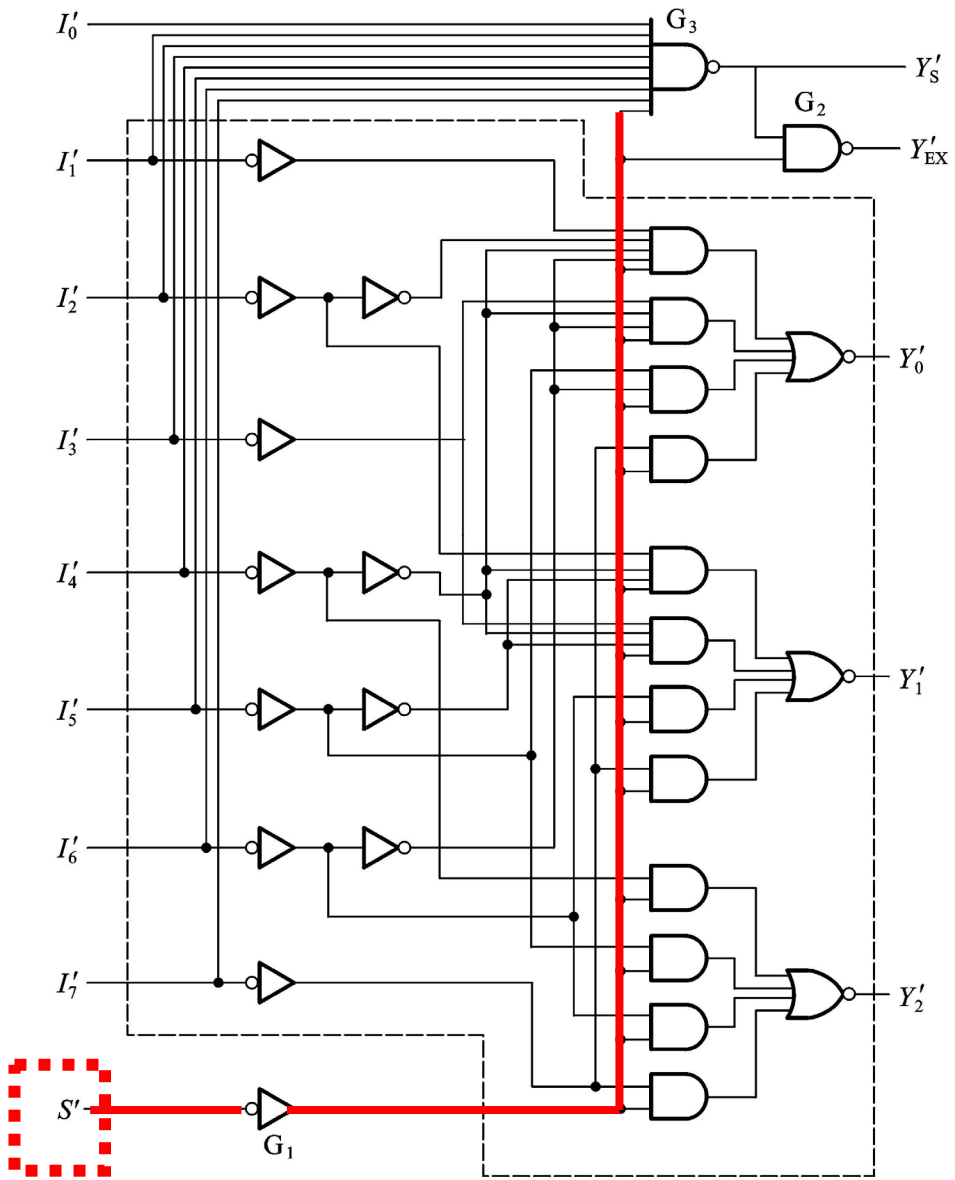
实例：74HC148

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

选通信号

$$Y_2' = [(I_7 + I_6 + I_5 + I_4)S]'$$

$$Y_2' = [(I_7 + I_6 + I_5 + I_4)S]'$$
$$Y_1' = [(I_7 + I_6 + I_5 I_4' I_3' + I_2 I_4' I_5')S]'$$
$$Y_0' = [(I_7 + I_6' I_5 + I_3 I_4' I_6' + I_1 I_2 I_4' I_6')S]'$$



选通信号

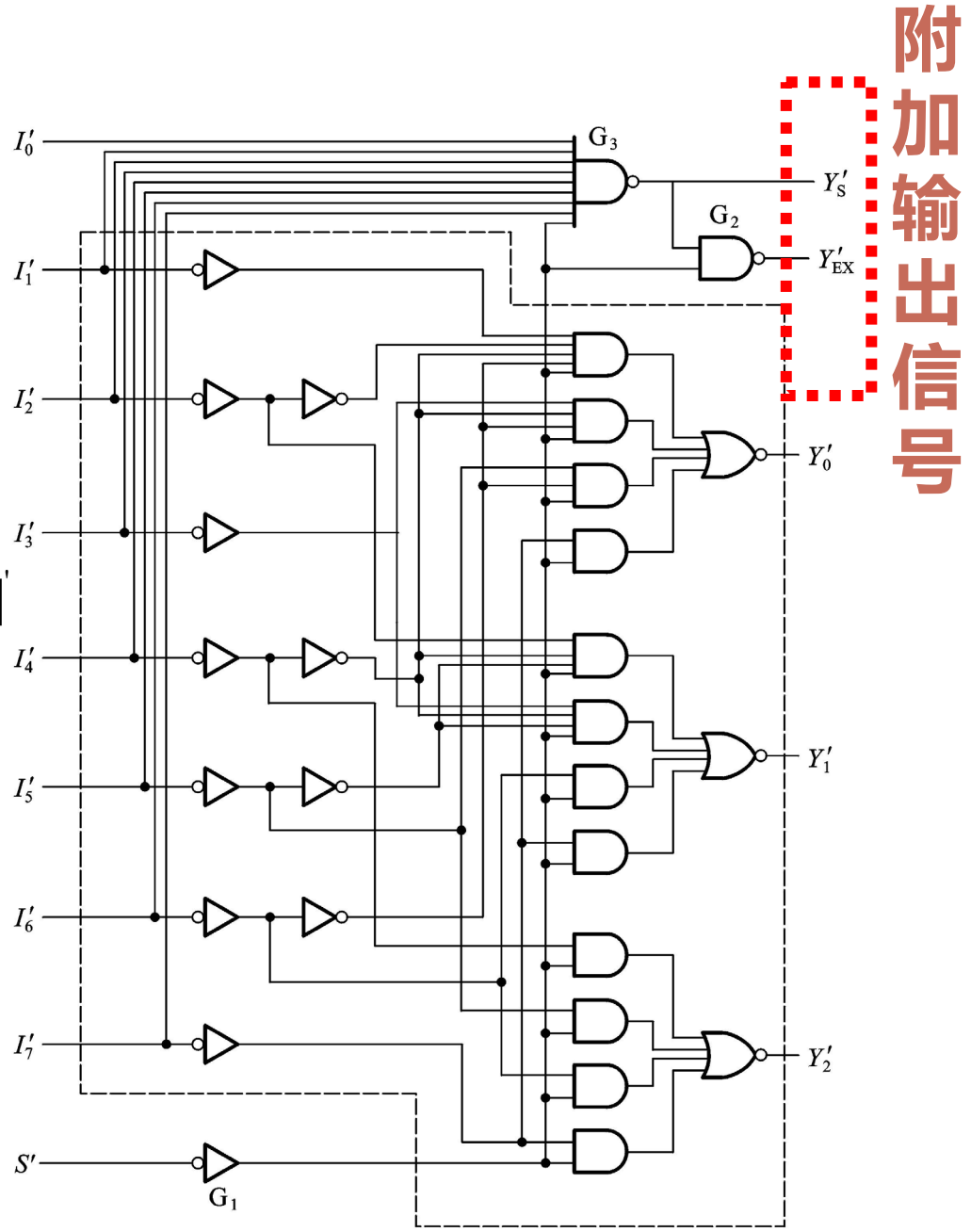
为0时，电路工作无编码输入

$$Y_S' = (I_7'I_6'I_5'I_4'I_3'I_2'I_1'I_0'S)'$$

$$Y_{EX}' = [(I_7'I_6'I_5'I_4'I_3'I_2'I_1'I_0'S)'S]'$$

$$= [(I_7 + I_6 + I_5 + I_4 + I_3 + I_2 + I_1 + I_0)S]'$$

为0时，电路工作有编码输入



附加输出信号

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	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	X	X	X	X	X	X	X	0	0	0	0	1	0
0	X	X	X	X	X	X	0	1	0	0	1	1	0
0	X	X	X	X	X	0	1	1	0	1	0	1	0
0	X	X	X	X	0	1	1	1	0	1	1	1	0
0	X	X	X	0	1	1	1	1	1	0	0	1	0
0	X	X	0	1	1	1	1	1	1	0	1	1	0
0	X	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

附加输出信号的状态及含义

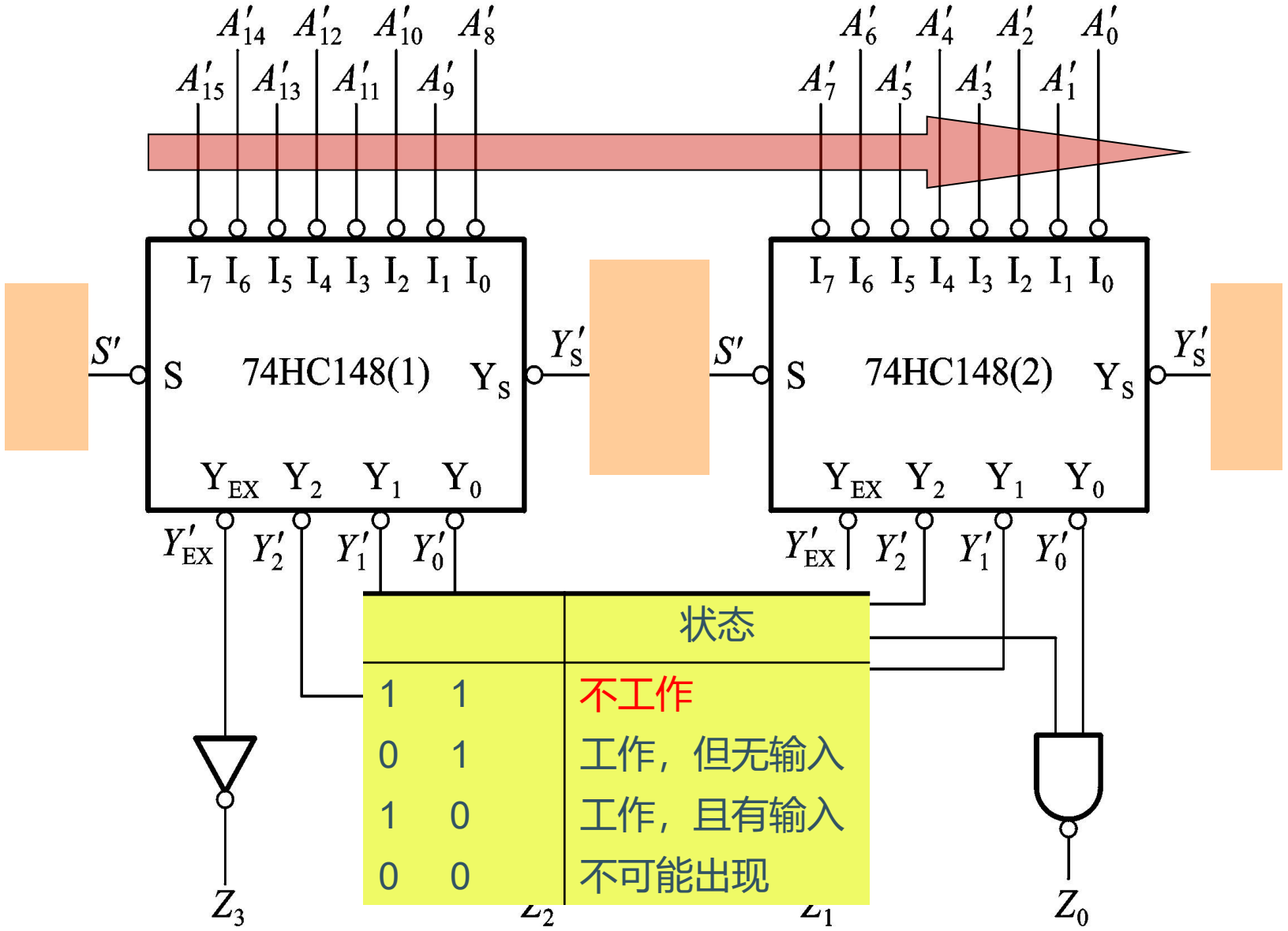
$Y_S^{'}$	$Y_{EX}^{'}$	状态
1	1	不工作
0	1	工作，但无输入
1	0	工作，且有输入
0	0	不可能出现

控制端扩展功能举例

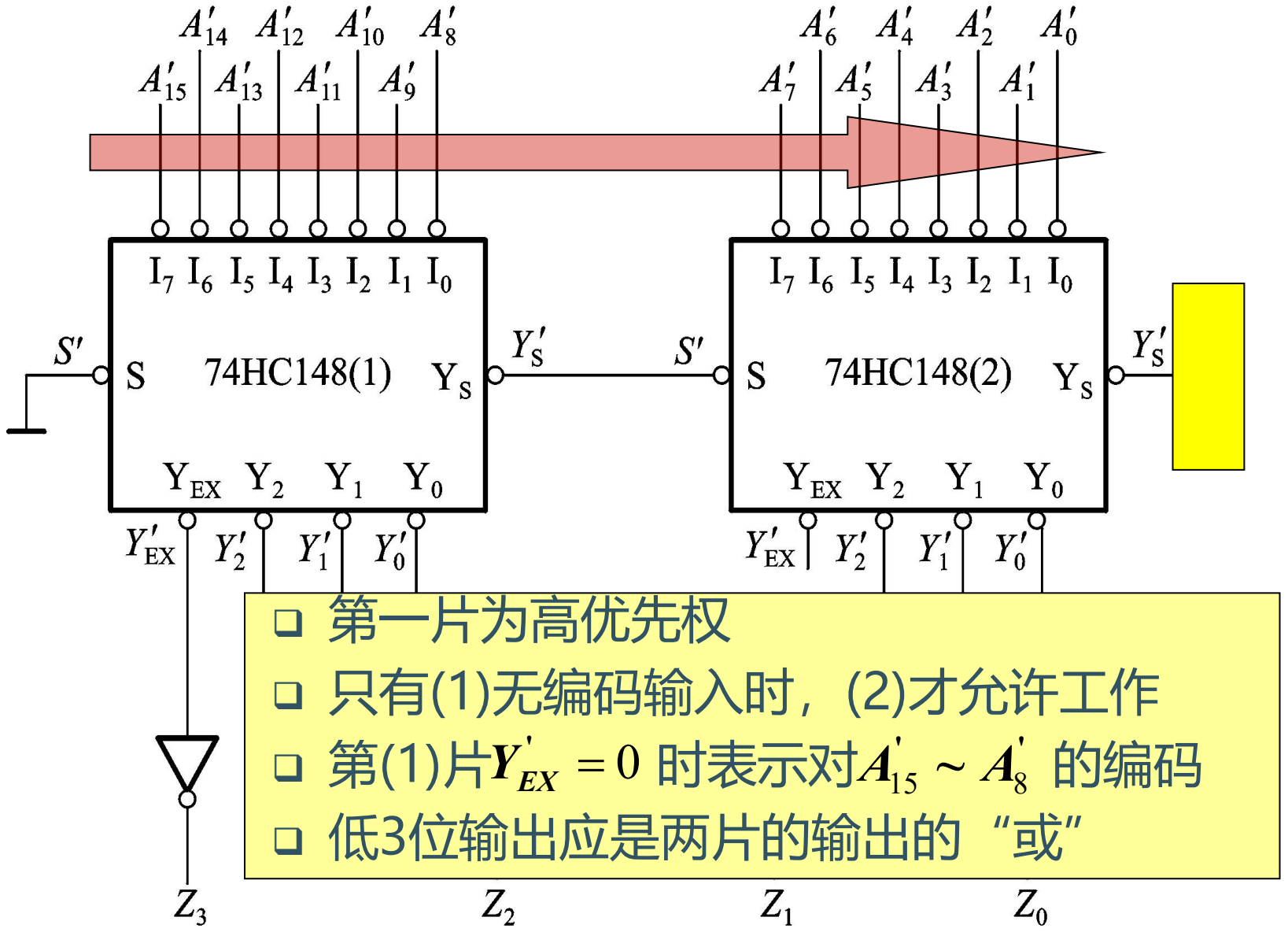
- 例：用两片8线-3线优先编码器

16线-4线优先编码器

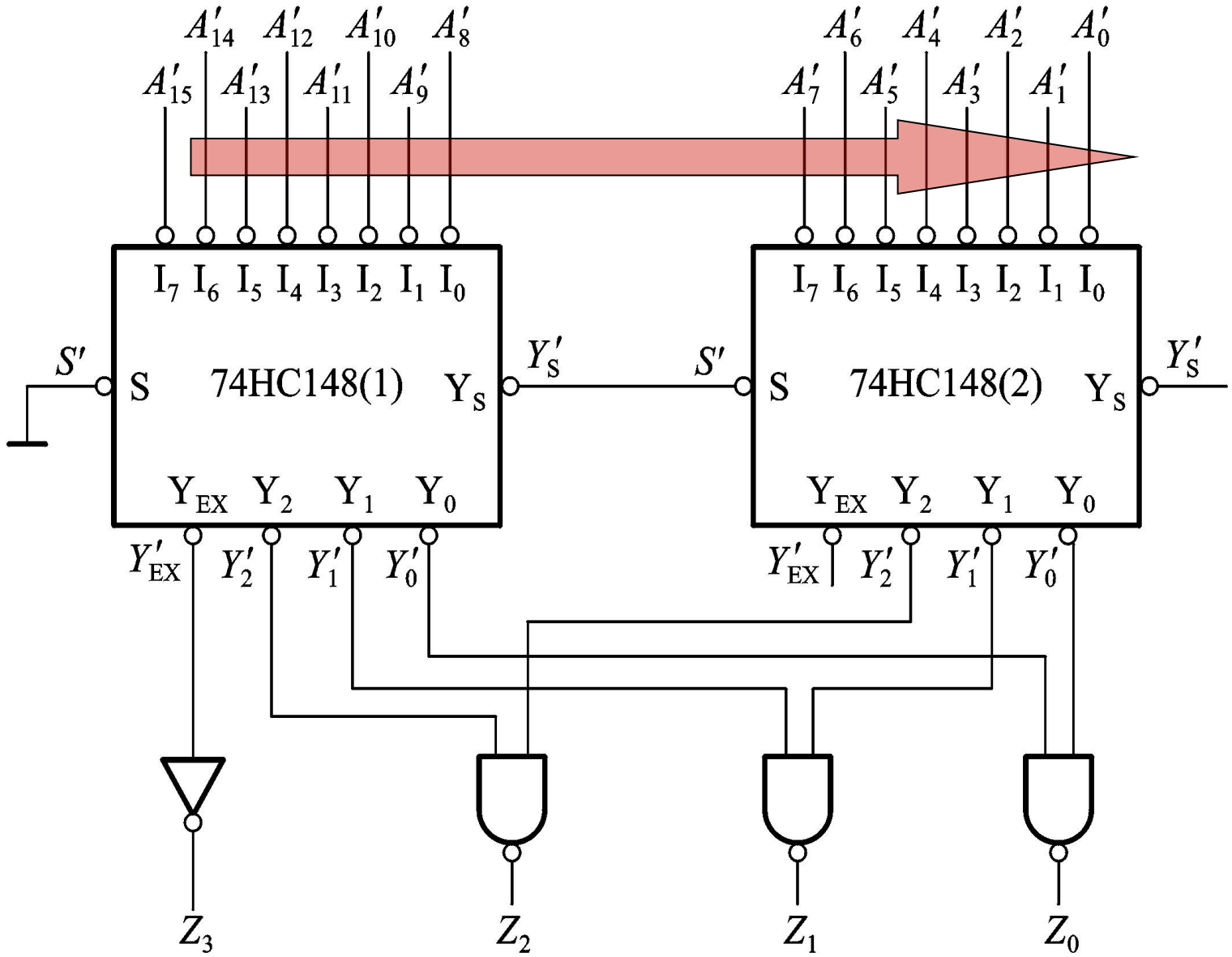
其中，A_{15}' 的优先权最高· · ·



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



- 第一片为高优先权
- 只有(1)无编码输入时，(2)才允许工作
- 第(1)片$Y'_{EX}=0$ 时表示对$A'_{15}\sim A'_8$ 的编码
- 低3位输出应是两片的输出的“或”



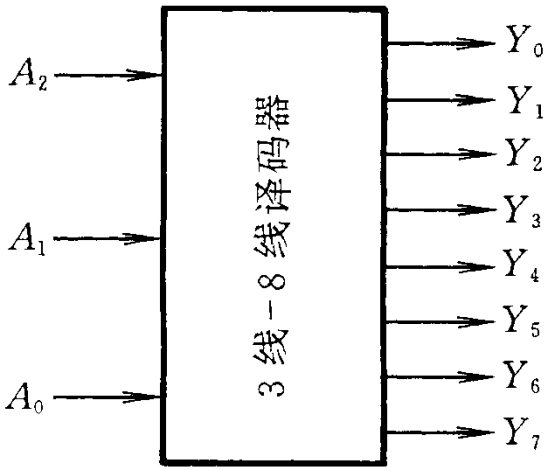
译码器

- 译码：将每个输入的二进制代码译成对应的输出高、低电平信号

- 常用的有：二进制译码器，二-十进制译码器，显示译码器等

一、二进制译码器

例：3线—8线译码器



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

真值表 → 逻辑表达式

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

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

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

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

…

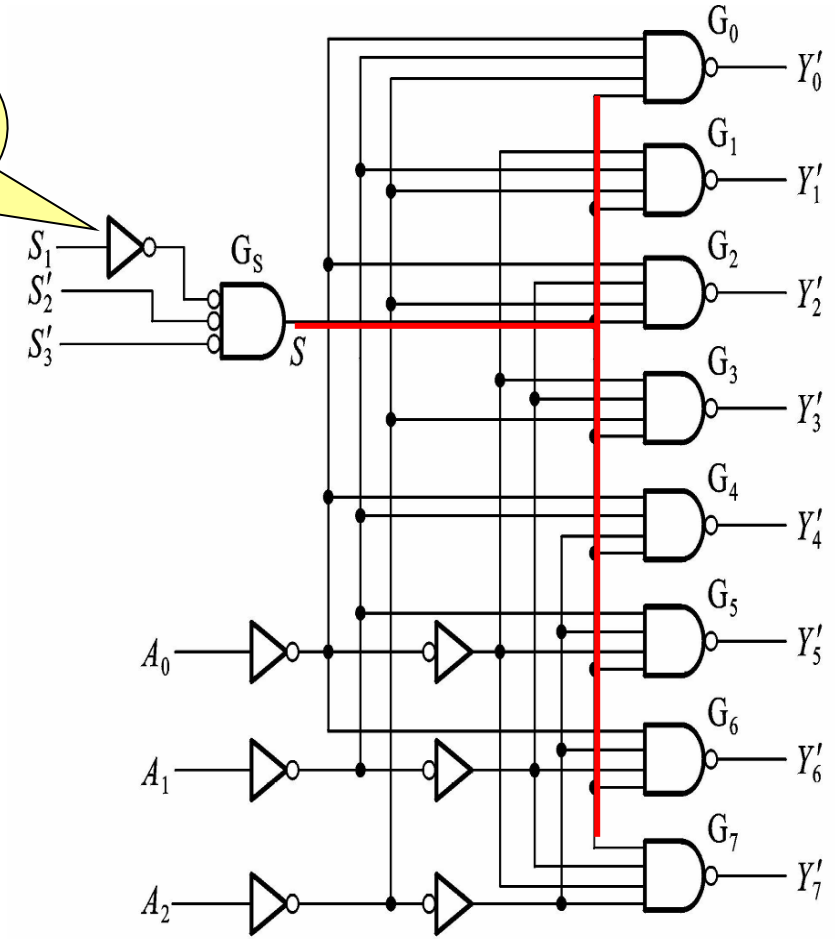
$$Y_7 = A_2 A_1 A_0 = m_7$$

集成译码器实例：74HC138

附加
控制端

$$S = S_3 S_2 S_1$$

$$Y_i' = (S \quad m_i)'$$



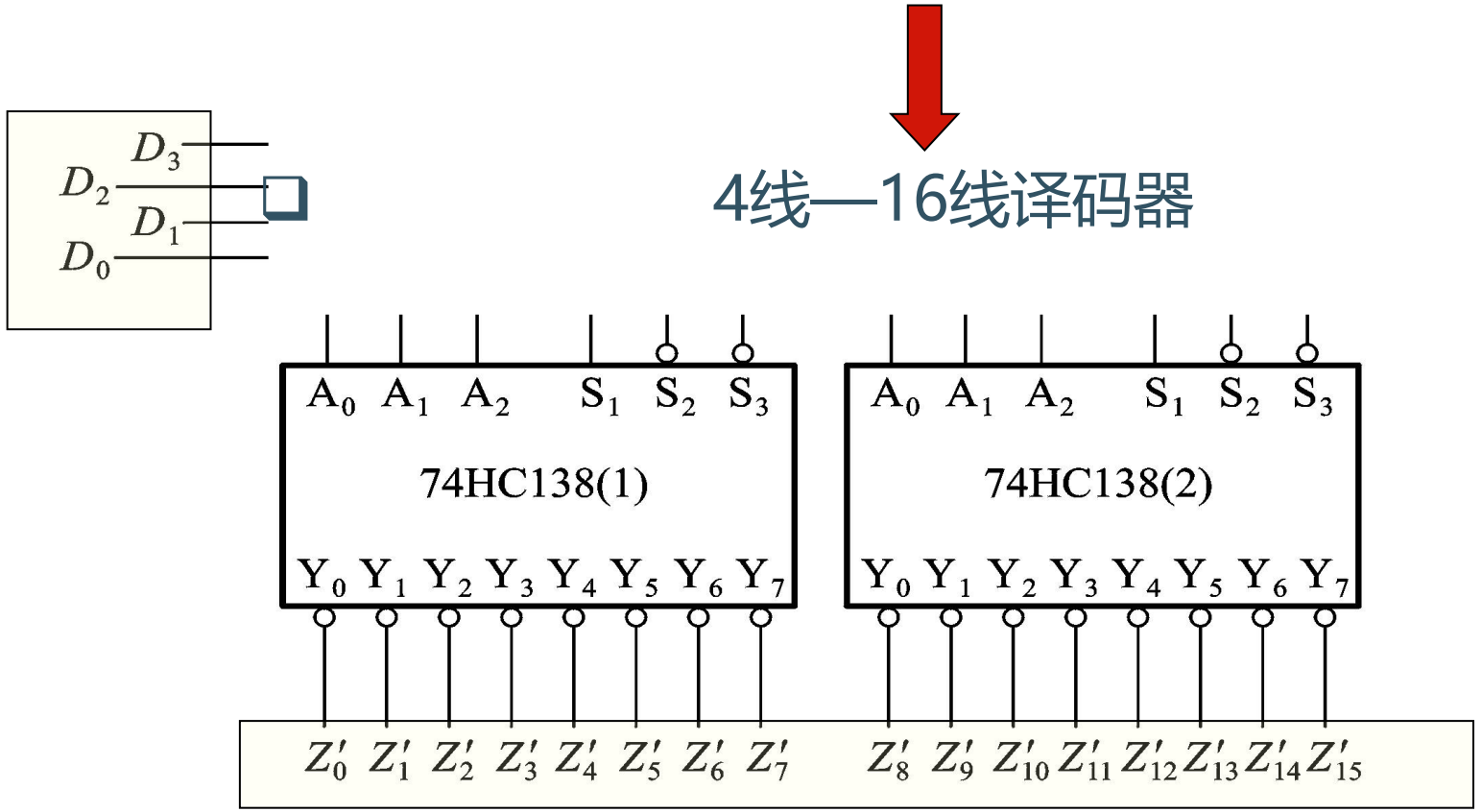
低电平
输出

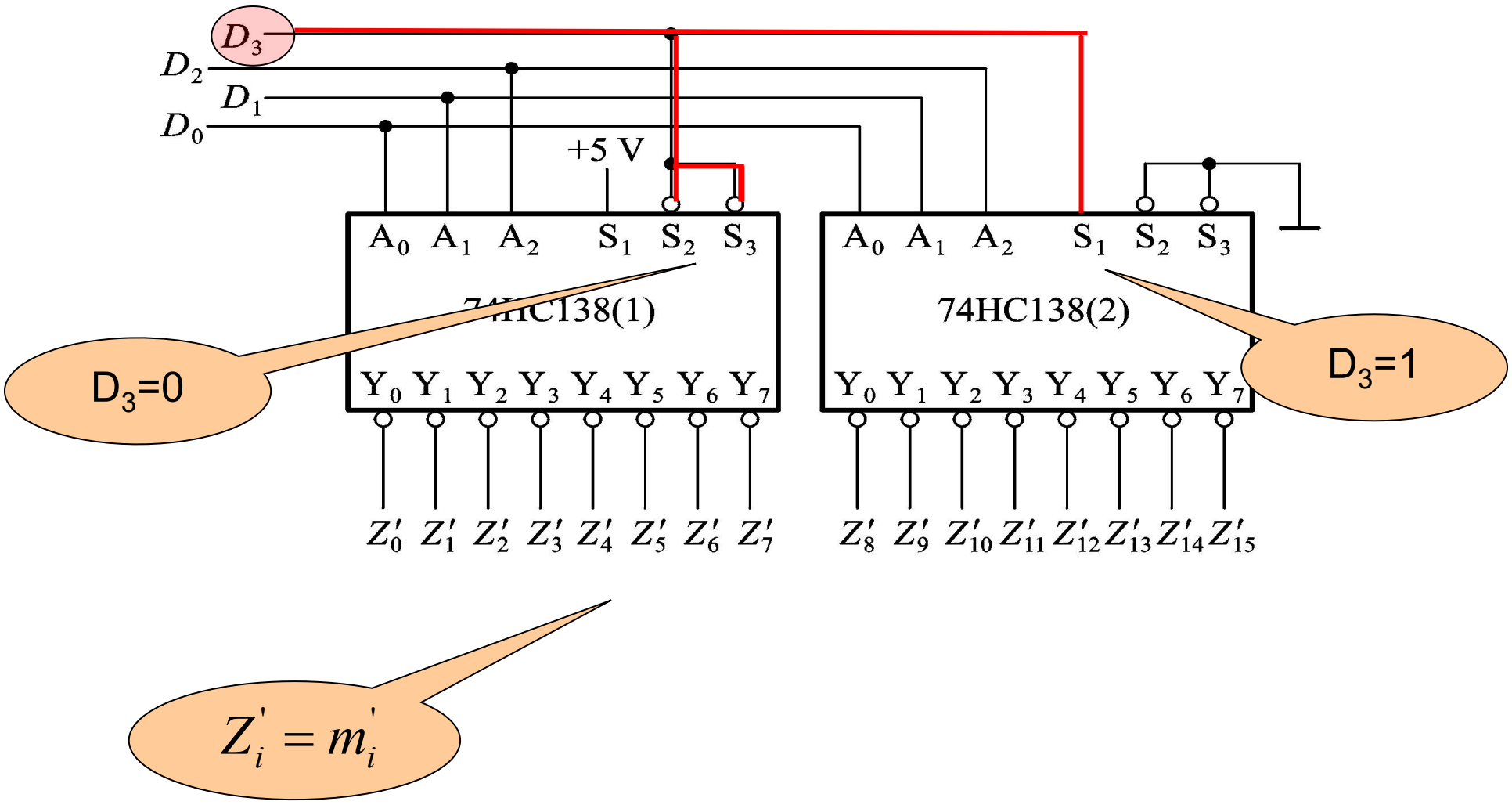
74HC138的功能表

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	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	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

❑利用附加控制端进行扩展

例:用74HC138（3线—8线译码器）



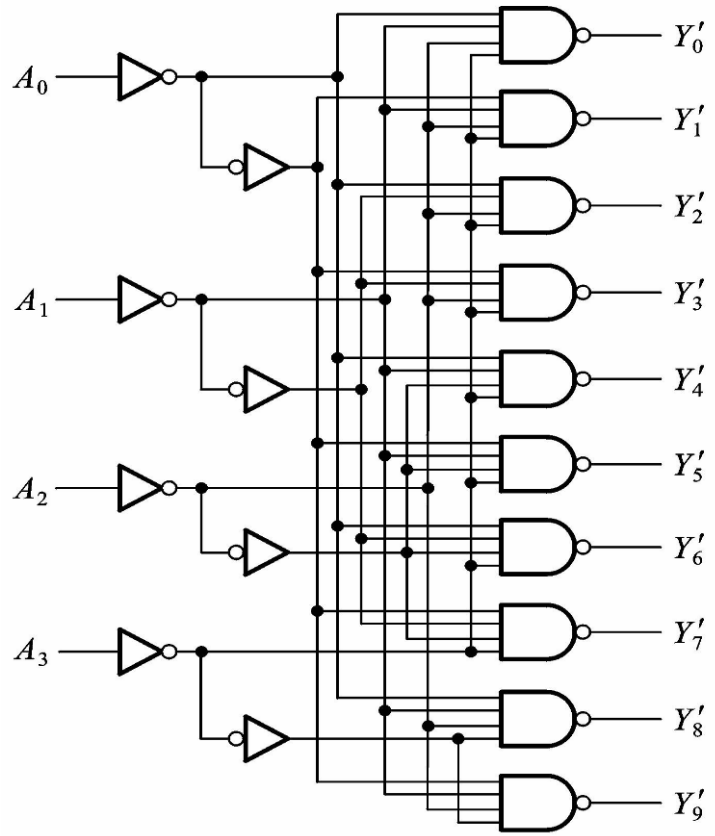

D_3
D_2
D_1
D_0
+5 V
A_0 A_1 A_2 S_1 S_2 S_3
74HC138(1)
A_0 A_1 A_2 S_1 S_2 S_3
74HC138(2)
Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7
Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7
Z'_0 Z'_1 Z'_2 Z'_3 Z'_4 Z'_5 Z'_6 Z'_7
Z'_8 Z'_9 Z'_{10} Z'_{11} Z'_{12} Z'_{13} Z'_{14} Z'_{15}
D_3=0
D_3=1
$Z_i^{'} = m_i^{'}$

二—十进制译码器

- 将输入BCD码的10个代码译成10个高、低电平的输出信号
  BCD码以外的伪码，输出均无低电平信号产生

- 74HC42

$$Y_i' = m_i' \quad (i = 0 \sim 9)$$



用译码器设计组合逻辑电路

1. 基本原理

3位二进制译码器给出3变量的全部最小项

。。。

n位二进制译码器给出n变量的全部最小项

任意逻辑函数

将n位二进制译码输出的最小项组合起来，可获得任何形式的输入变量不大于n的组合函数

$$Y = \sum m_i$$

用译码器设计组合电路例

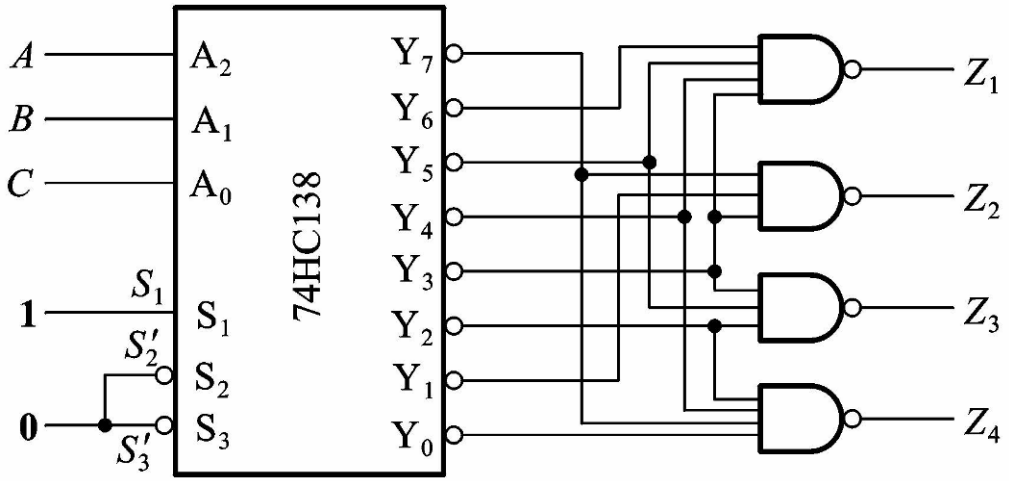
利用74HC138设计一个多输出的组合逻辑电路，输出逻辑函数式为：

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

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

$$Z_3 = A'B + AB'C$$

$$Z_4 = A'BC' + B'C' + ABC$$



$$Z_1 = AC' + A'BC + AB'C = \sum m(3,4,5,6)$$

$$Z_2 = BC + A'B'C = \sum m(1,3,7)$$

$$Z_3 = A'B + AB'C = \sum m(2,3,5)$$

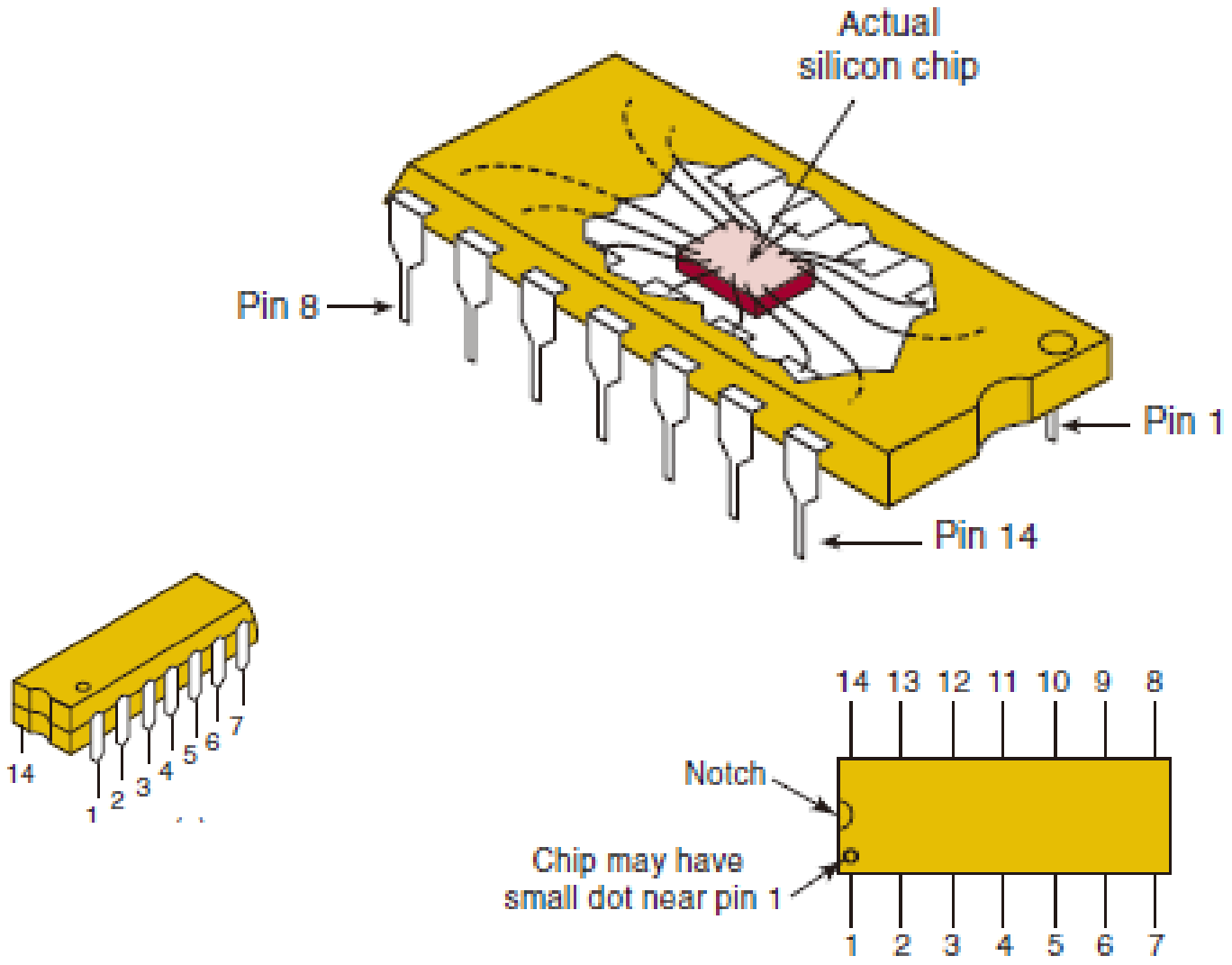
$$Z_4 = A'BC' + B'C' + ABC = \sum m(0,2,4,7)$$

$$Z_1 = \sum m(3,4,5,6) = (m_3' m_4' m_5' m_6')'$$

$$Z_2 = \sum m(1,3,7) = (m_1' m_3' m_7')'$$

$$Z_3 = \sum m(2,3,5) = (m_2' m_3' m_5')'$$

$$Z_4 = \sum m(0,2,4,7) = (m_0' m_2' m_4' m_7')'$$


Actual
silicon chip
Pin 8
Pin 1
Pin 14
14
1
2
3
4
5
6
7
14 13 12 11 10 9 8
Notch
Chip may have
small dot near pin 1
1 2 3 4 5 6 7

组合逻辑

刘 鹏

浙江大学

信息与电子工程学院

Email: liupeng@zju.edu.cn