

# 第四章 组合逻辑电路

## 4.1 概 述

### 主要要求:

- 掌握组合逻辑电路和时序逻辑电路的概念。
- 了解组合逻辑电路的特点与描述方法。

# 一、组合逻辑电路的概念

● 数字电路根据逻辑功能特点的不同分为

组合逻辑电路

指任何时刻的输出仅取决于该时刻输入信号的组合，而与电路原有的状态无关的电路。

时序逻辑电路

指任何时刻的输出不仅取决于该时刻输入信号的组合，而且与电路原有的状态有关的电路。

## 二、组合逻辑电路的特点与描述方法

- 组合逻辑电路的逻辑功能特点：

没有存储和记忆功能。

- 组合电路的组成特点：

由门电路构成，不含记忆单元，只存在从输入到输出的通路，没有反馈回路。

- 组合电路的描述方法主要有逻辑表达式、真值表、卡诺图和逻辑图等。

## 4.2 组合逻辑电路的分析和设计

### 一、组合逻辑电路的基本分析方法

分析思路：

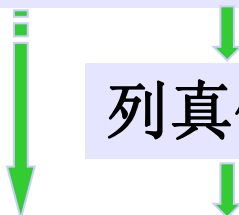
根据给定逻辑电路，找出输出输入间的逻辑关系，从而确定电路的逻辑功能。

基本步骤：

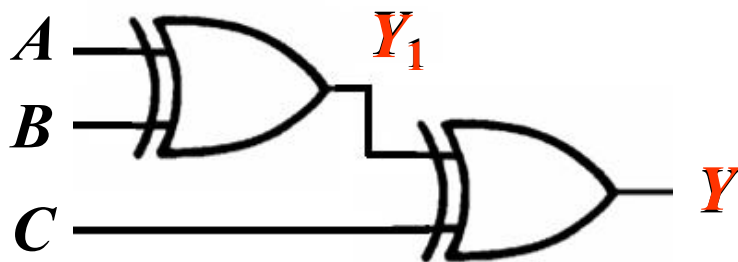
根据给定逻辑图**写出输出逻辑式**，并进行必要的化简

列真值表

分析逻辑功能



★ [例] 分析下图所示逻辑电路的功能。



解: (1) 写出输出逻辑函数式

$$Y_1 = A \oplus B$$

$$Y = A \oplus B \oplus C$$

$$= (A \oplus B)C' + (A \oplus B)' \cdot C$$

$$= A'B'C + A'BC' + AB'C' + ABC \rightarrow$$

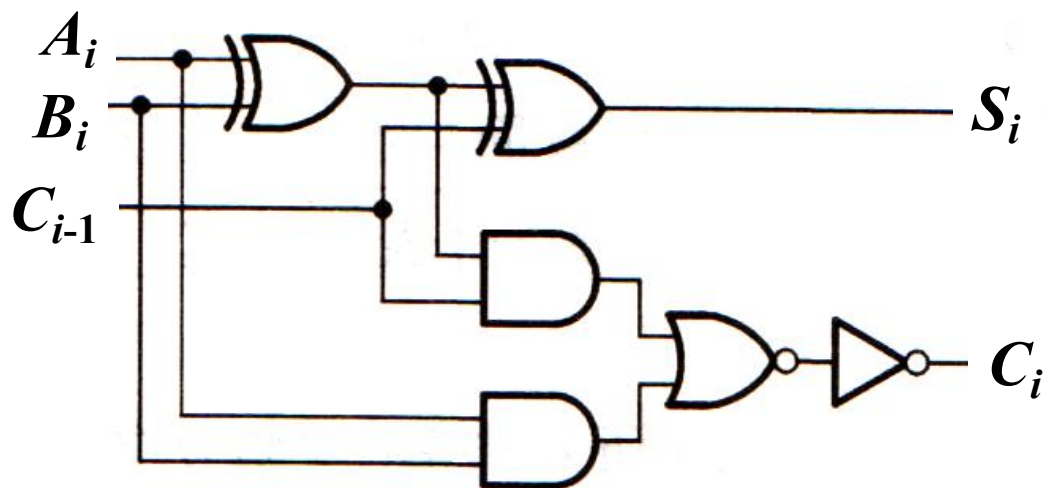
(3) 分析逻辑功能

$A$ 、 $B$ 、 $C$  三个输入变量中, 有奇数个 1 时, 输出为 1, 否则输出为 0。因此, 图示电路为三位判奇电路, 又称奇校验电路。

(2) 列逻辑函数真值表

| 输 入 |     |     | 输 出 |
|-----|-----|-----|-----|
| $A$ | $B$ | $C$ | $Y$ |
| 0   | 0   | 0   | 0   |
| 0   | 0   | 1   | 1   |
| 0   | 1   | 0   | 1   |
| 0   | 1   | 1   | 0   |
| 1   | 0   | 0   | 1   |
| 1   | 0   | 1   | 0   |
| 1   | 1   | 0   | 0   |
| 1   | 1   | 1   | 1   |

★ [例] 分析下图电路的逻辑功能。



解：(1) 写出输出逻辑函数式

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

$$\begin{aligned} C_i &= (A_i \oplus B_i)C_{i-1} + A_i B_i \\ &= A_i B_i' C_{i-1} + A_i' B_i C_{i-1} + A_i B_i \end{aligned}$$

(3) 分析逻辑功能

将两个一位二进制数  $A_i$ 、 $B_i$  与低位来的进位  $C_{i-1}$  相加， $S_i$  为本位和， $C_i$  为向高位产生的进位。这种功能的电路称为全加器。

(2) 列真值表

| 输 入   |       |           | 输 出   |       |
|-------|-------|-----------|-------|-------|
| $A_i$ | $B_i$ | $C_{i-1}$ | $S_i$ | $C_i$ |
| 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     |

## 二、组合逻辑电路的基本设计方法

设计思路：

分析给定逻辑要求，设计出能实现该功能的组合逻辑电路。

基本步骤：

分析设计要求并列出现值表→求最简输出逻辑式→画逻辑图。

首先分析给定问题，弄清楚输入变量和输出变量是哪些，并规定它们的符号与逻辑取值（即规定它们何时取值0，何时取值1）。然后分析输出变量和输入变量间的逻辑关系，列出真值表。

根据真值表用代数法或卡诺图法求最简与或式，然后根据题中对门电路类型的要求，将最简与或式变换为与门类型对应的最简式。

## (一) 单输出组合逻辑电路设计举例

★ [例] 设计一个 **A、B、C 三人表决电路**。当表决某个提案时，**多数人同意，则提案通过，但 A 具有否决权。用与非门实现。**

解： (1) 分析设计要求，列出真值表

设  $A$ 、 $B$ 、 $C$  同意提案时取值为 1，不同意时取值为 0； $Y$  表示表决结果，提案通过则取值为 1，否则取值为 0。可得真值表如右。

(2) 化简输出函数，并求最简与非式

| $A \backslash BC$ |    |    |    |    |
|-------------------|----|----|----|----|
|                   | 00 | 01 | 11 | 10 |
| 0                 | 0  | 0  | 0  | 0  |
| 1                 | 0  | 1  | 1  | 1  |

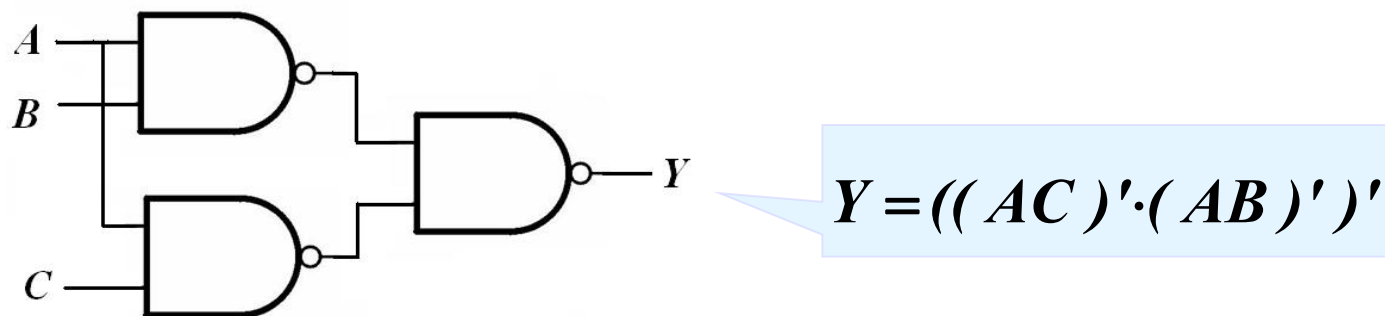
$$Y = AC + AB$$

$$= ((AC + AB)')' = ((AC)' \cdot (AB)')'$$

| 输 入 |     |     | 输出  |
|-----|-----|-----|-----|
| $A$ | $B$ | $C$ | $Y$ |
| 0   | 0   | 0   | 0   |
| 0   | 0   | 1   | 0   |
| 0   | 1   | 0   | 0   |
| 0   | 1   | 1   | 0   |
| 1   | 0   | 0   | 0   |
| 1   | 0   | 1   | 1   |
| 1   | 1   | 0   | 1   |
| 1   | 1   | 1   | 1   |



### (3) 根据输出逻辑式画逻辑图



## (二) 多输出组合逻辑电路设计举例

★ [例] 试设计半加器电路。

解：(1) 分析设计要求，  
列真值表。

| 输 入   |       | 输 出   |       |
|-------|-------|-------|-------|
| $A_i$ | $B_i$ | $S_i$ | $C_i$ |
| 0     | 0     | 0     | 0     |
| 0     | 1     | 1     | 0     |
| 1     | 0     | 1     | 0     |
| 1     | 1     | 0     | 1     |

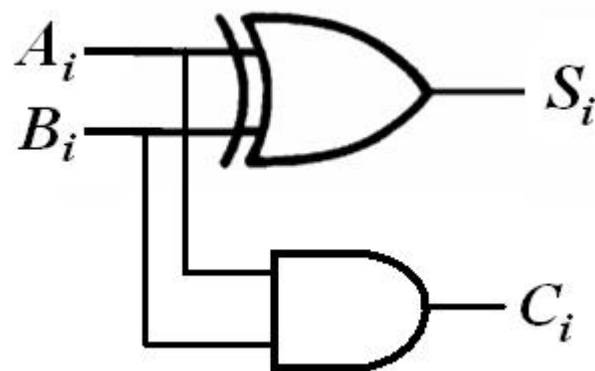
将两个 1 位二进制数相加，而不考虑低位进位的运算电路，称为半加器。

(3) 画逻辑图

(2) 求最简输出函数式

$$S_i = A_i \oplus B_i$$

$$C_i = A_i B_i$$



## 4.3 若干常用的组合逻辑电路

### 4.3.1 编码器

主要要求：

- 理解编码的概念。
- 理解常用编码器的类型、逻辑功能和使用方法。

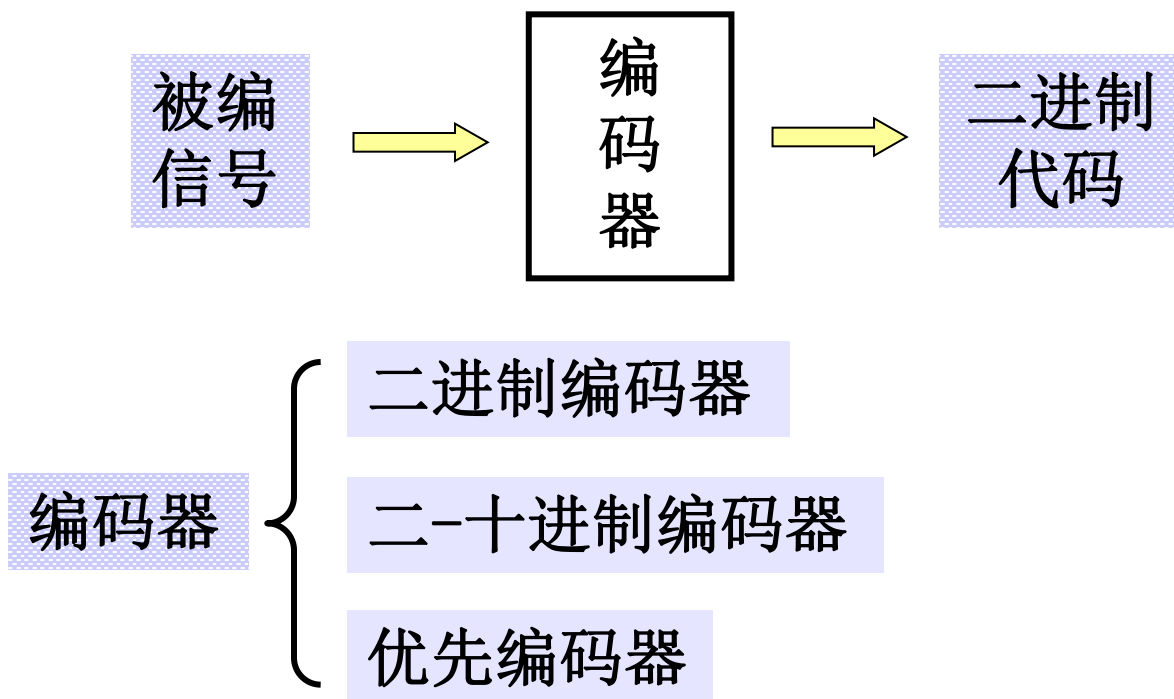
# 一、编码器的概念与类型

**编码**

将具有特定含义的信息编成相应二进制代码的过程。

**编码器(即Encoder)**

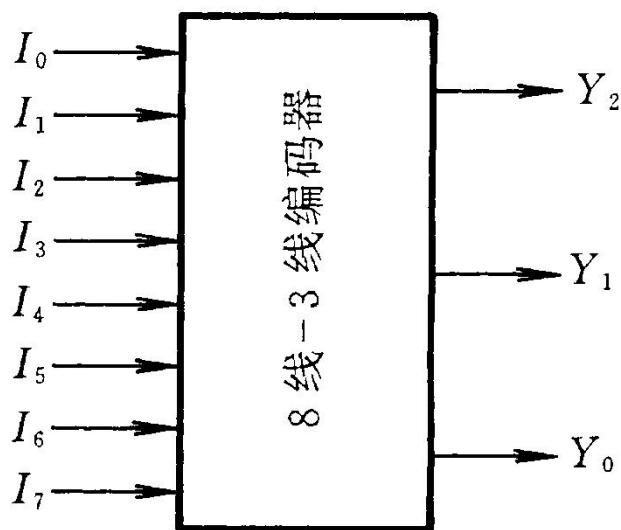
实现编码功能的电路



## 一、普通编码器

**特点：**任何时刻只允许输入一个编码信号。

**例：**三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     |

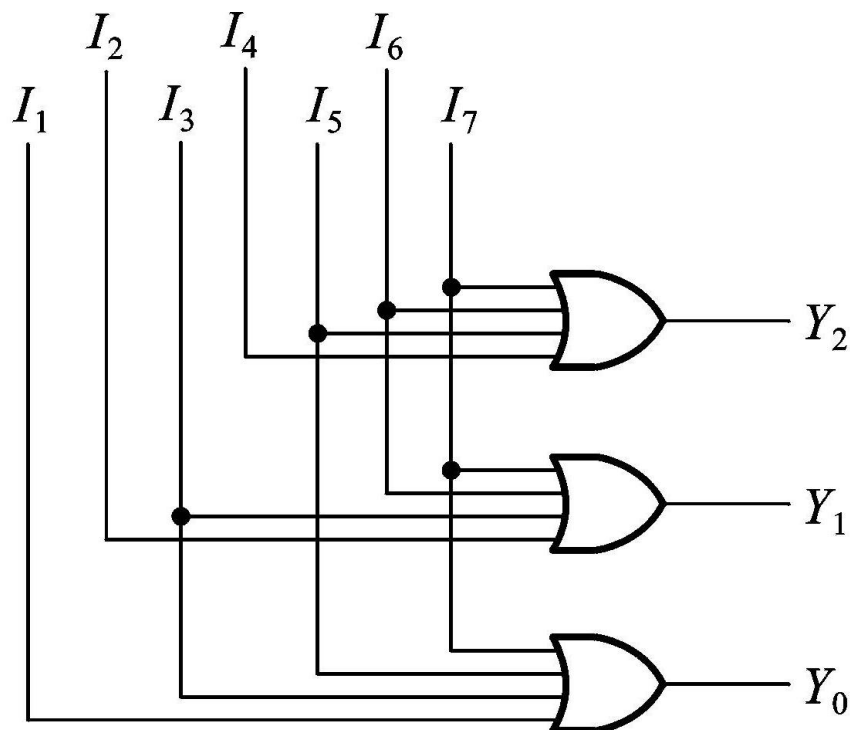
$$Y_2 = I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0' + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0' + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0' + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0'$$

利用无关项化简，得：

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

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

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



## 二、优先编码器

允许同时输入数个编码信号，并只对其中优先权最高的信号进行编码输出的电路。

普通编码器在任何时刻只允许一个输入端请求编码，否则输出发生混乱。

| 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     |

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   |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $I_0$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ | $Y_2$ | $Y_1$ | $Y_0$ |
| ×     | ×     | ×     | ×     | ×     | ×     | ×     | 1     | 1     | 1     | 1     |
| ×     | ×     | ×     | ×     | ×     | ×     | 1     | 0     | 1     | 1     | 0     |
| ×     | ×     | ×     | ×     | ×     | 1     | 0     | 0     | 1     | 0     | 1     |
| ×     | ×     | ×     | ×     | 1     | 0     | 0     | 0     | 1     | 0     | 0     |
| ×     | ×     | ×     | 1     | 0     | 0     | 0     | 0     | 0     | 1     | 1     |
| ×     | ×     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     |
| ×     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     |
| 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

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



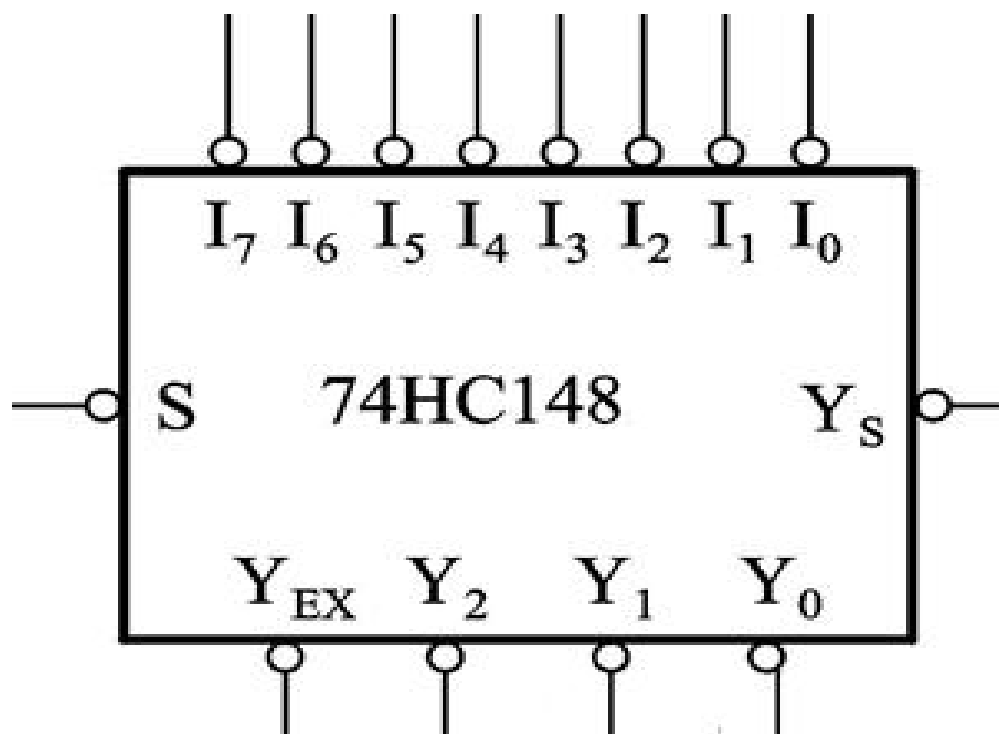
$$Y_2 = I_7 + I_6 + I_5 + I_4$$

$A + A'B = A + B$



## 编码器

### 8线—3线优先编码器74HC148



### 优先编码

优先编码器允许几个输入端**同时**加上信号，电路只对其中**优先级最高的信号**进行编码。

## 管脚定义：

$\bar{I}_0 \sim \bar{I}_7$  : 输入, 低电平有效, 优先级别依次为  $\bar{I}_7 \sim \bar{I}_0$ 。

$\bar{Y}_2 \sim \bar{Y}_0$  : 编码输出端。

$\bar{S}$  : 使能输入端;  $\bar{S} = 0$  时, 编码,  
 $\bar{S} = 1$  时, 禁止编码。

$\bar{Y}_s$  : 使能输出端,  $\bar{Y}_s = 0$ , 表示芯片已经开始工作,  
但因所有的输入都无效,  
输出固定为111, 允许其他芯片工作。

$\bar{Y}_{EX}$  : 扩展输出端, 编码状态下 ( $\bar{S}=0$ ), 若有输入信号,  
 $\bar{Y}_{EX}=0$ 。表示芯片已经开始工作, 且有优先码输出。

## 74HC148的功能表

[illegible]

# 实例：74HC148 (8—3线优先编码器)

$$Y'_S = (I'_0 I'_1 I'_2 I'_3 I'_4 I'_5 I'_6 I'_7 S')'$$

电路工作但无编码输入

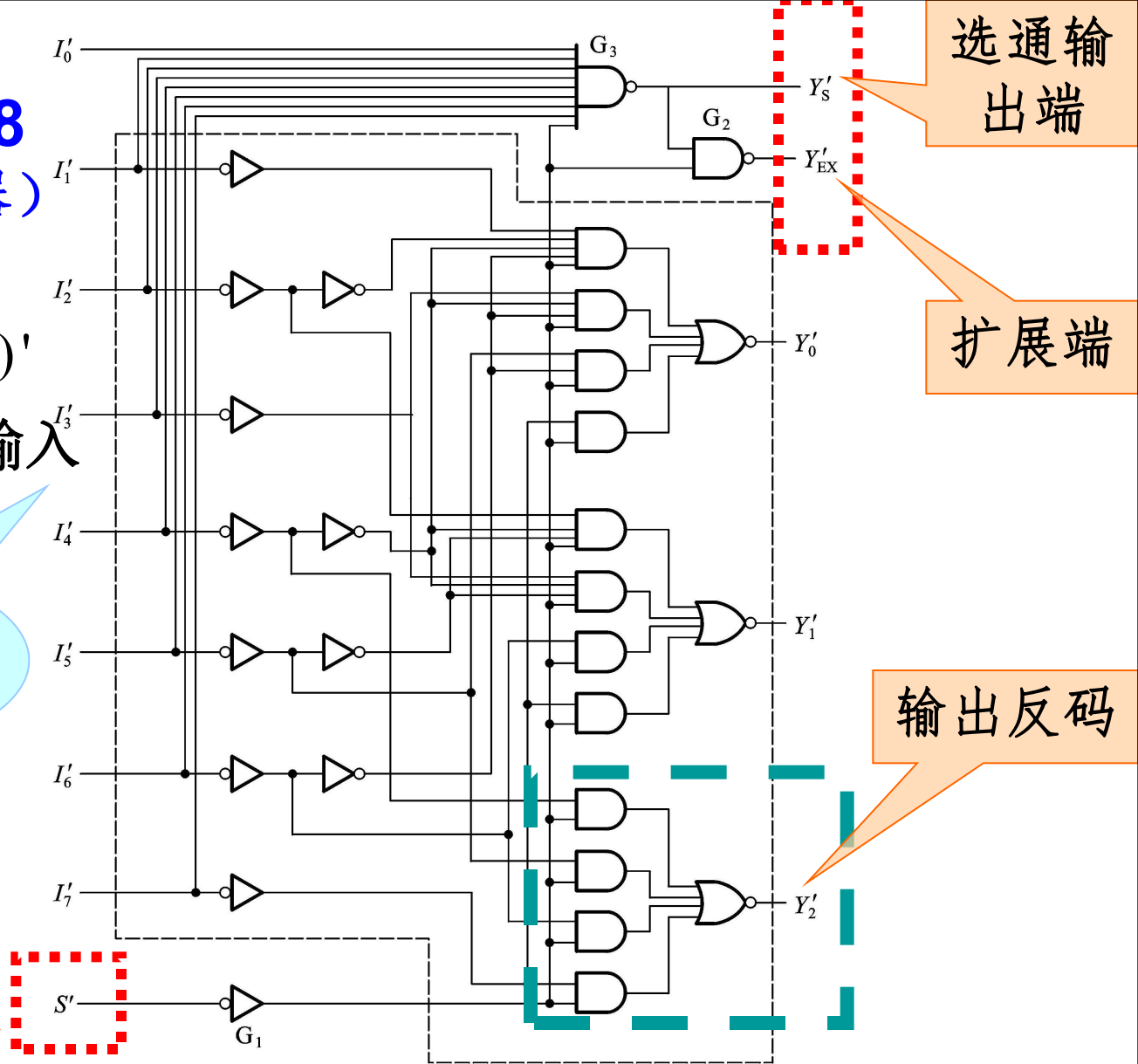
低电平输入有效

选通输入端  
(低电平有效)

$$Y'_{EX} = ((I'_0 I'_1 I'_2 I'_3 I'_4 I'_5 I'_6 I'_7 S')' S')'$$

$$= ((I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7) \cdot S)'$$

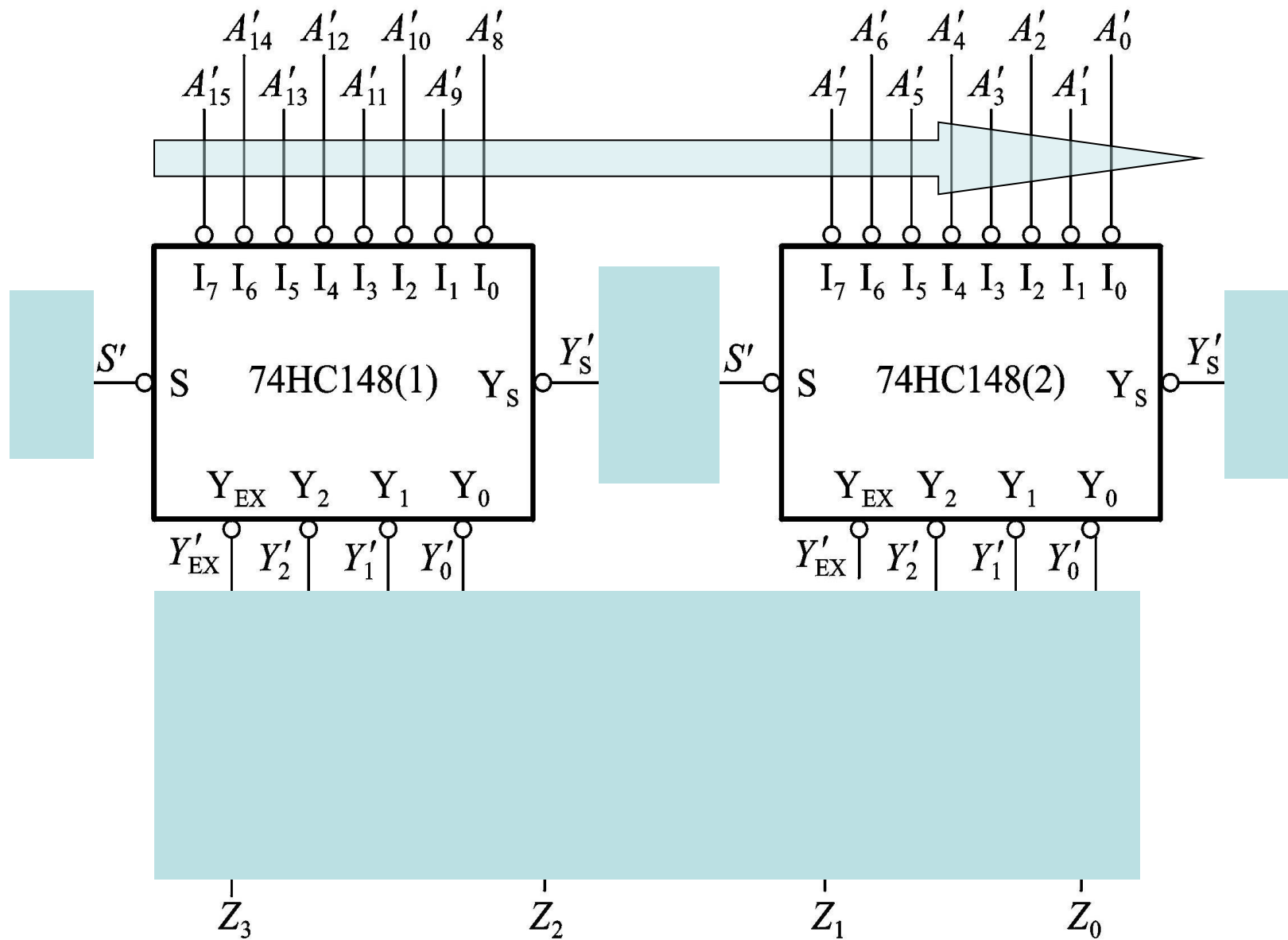
电路工作有编码输入

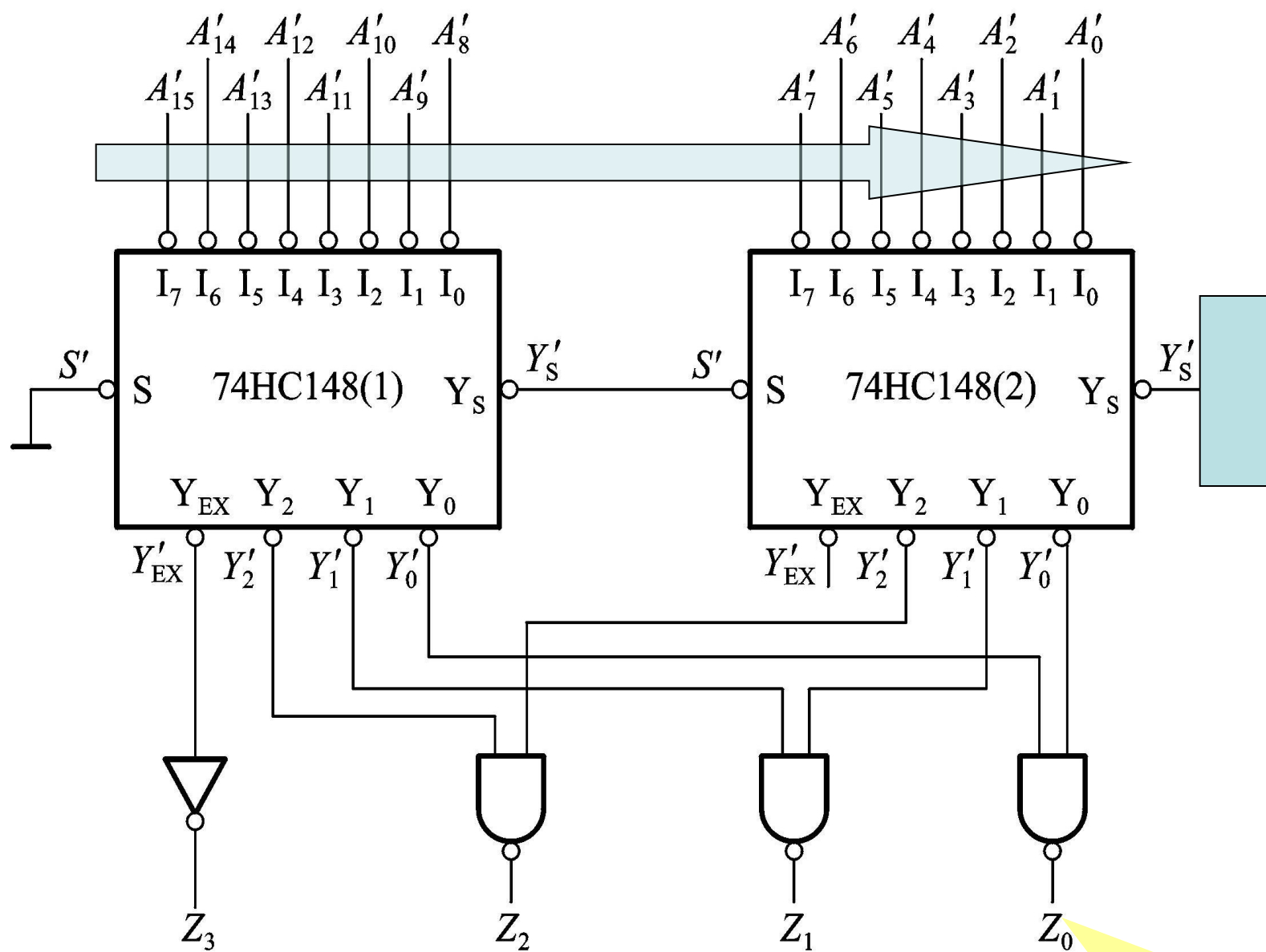


## 控制端扩展功能举例：

例： 用两片**8-3**线优先编码器构成  
**16- 4**优先编码器

其中， $A_{15}'$  的优先权最高。





输出是原码

## 4.3.2 译码器

### 主要要求：

- 理解译码的概念。
- 掌握二进制译码器 **74HC138** 的逻辑功能和使用方法。
- 理解其他常用译码器的逻辑功能和使用方法。
- 掌握**用二进制译码器实现组合逻辑电路**的方法。



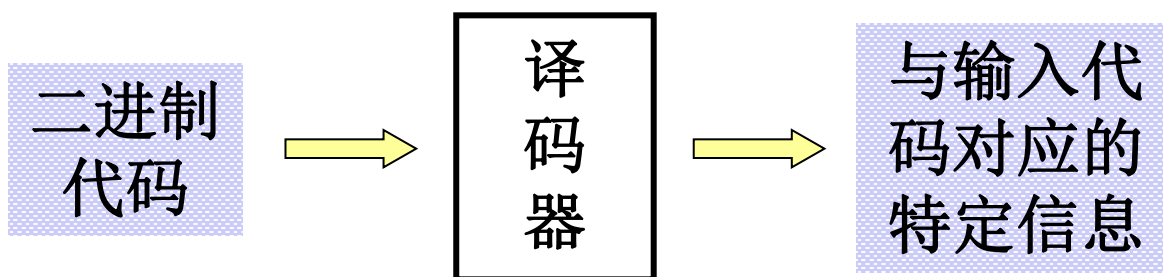
# 一、译码的概念与类型

译码是编码的逆过程。

将表示特定意义信息的  
二进制代码翻译出来。

译码器(即 Decoder)

实现译码功能的电路



译码器

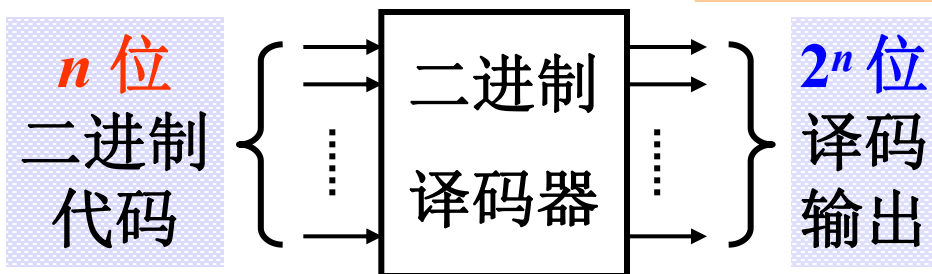
二进制译码器

二 - 十进制译码器

数码显示译码器

## 二、二进制译码器

将输入二进制代码译成相应输出信号的电路。



| 译码输入  |       | 译 码 输 出 |       |       |       |
|-------|-------|---------|-------|-------|-------|
| $A_1$ | $A_0$ | $Y_0$   | $Y_1$ | $Y_2$ | $Y_3$ |
| 0     | 0     | 1       | 0     | 0     | 0     |
| 0     | 1     | 0       | 1     | 0     | 0     |
| 1     | 0     | 0       | 0     | 1     | 0     |
| 1     | 1     | 0       | 0     | 0     | 1     |

译码输出高电平有效

| 译码输入  |       | 译 码 输 出 |        |        |        |
|-------|-------|---------|--------|--------|--------|
| $A_1$ | $A_0$ | $Y'_0$  | $Y'_1$ | $Y'_2$ | $Y'_3$ |
| 0     | 0     | 0       | 1      | 1      | 1      |
| 0     | 1     | 1       | 0      | 1      | 1      |
| 1     | 0     | 1       | 1      | 0      | 1      |
| 1     | 1     | 1       | 1      | 1      | 0      |

译码输出低电平有效

怎样写出输出端表达式