

在数字电路中，常用的组合电路有加法器、编码器、译码器、数据分配器和多路选择器等。下面几节分别介绍这几种典型组合逻辑电路的基本结构、工作原理和使用方法。

2.1 加法器(Adders)



二进制(Binary)

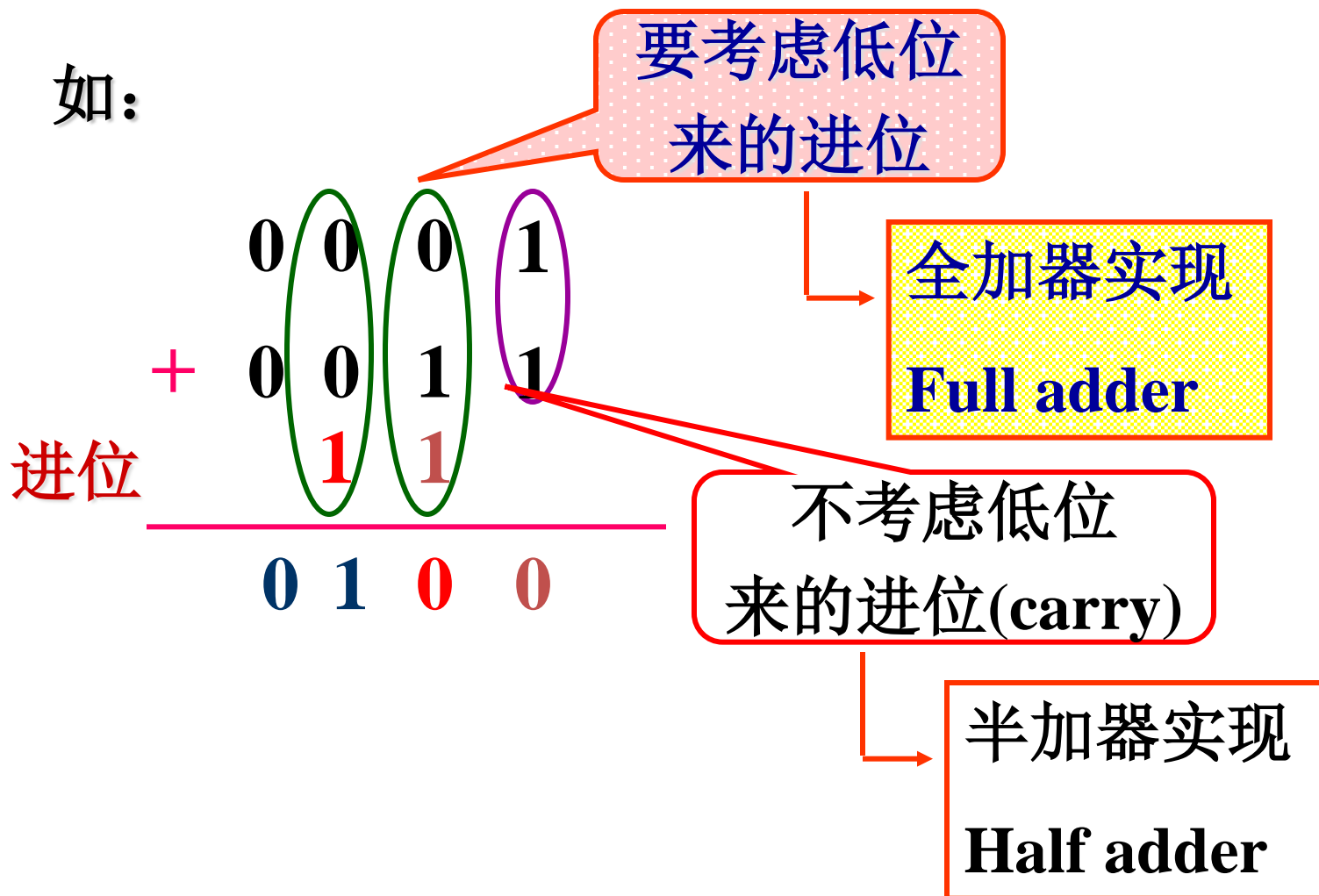
在数字电路中，为了把电路的两个状态 (“1”态和 “0”态)与数码对应起来，采用二进制。

二进制：0，1两个数码，“逢二进一”。

2.1 加法器

加法器：实现(realize)二进制加法运算的电路

如：



2.1.1 半加器(Half adder)

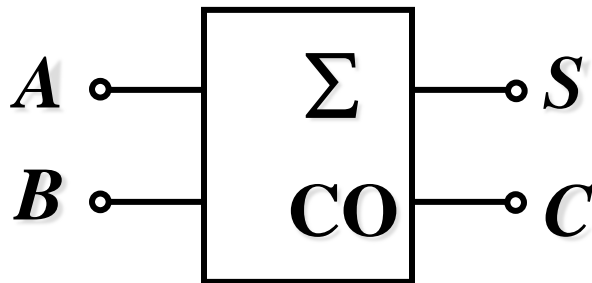
半加：实现两个一位(bit)二进制数相加，不考虑来自低位(low bit)的进位(carry bit)。

半加器：

两个输入 $\begin{cases} A \\ B \end{cases}$ 表示两个同位相加的数

两个输出 $\begin{cases} S & \text{— 表示半加和(half sum bit)} \\ C & \text{— 表示向高位的进位} \end{cases}$

逻辑符号：



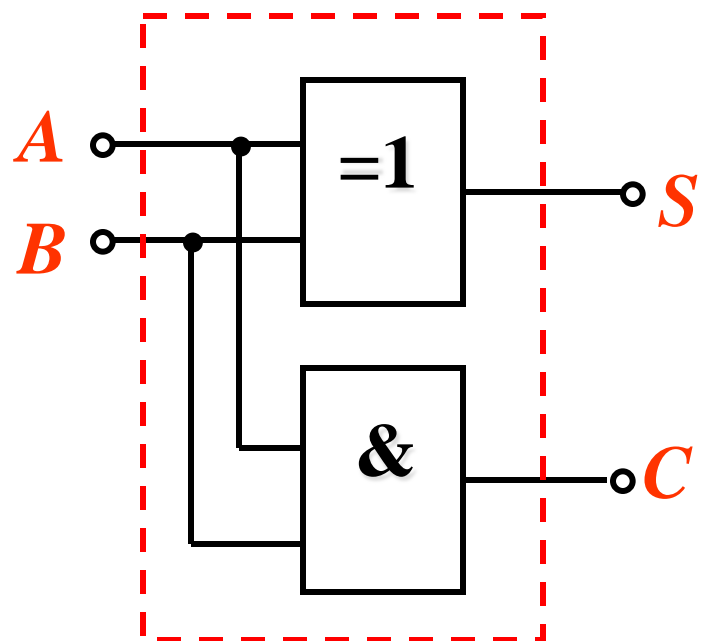
半加器逻辑状态表

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

逻辑表达式

$$S = A\bar{B} + \bar{A}B = A \oplus B$$

$$C = AB$$



逻辑图



(1) 列逻辑状态表

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

(2) 写出逻辑式

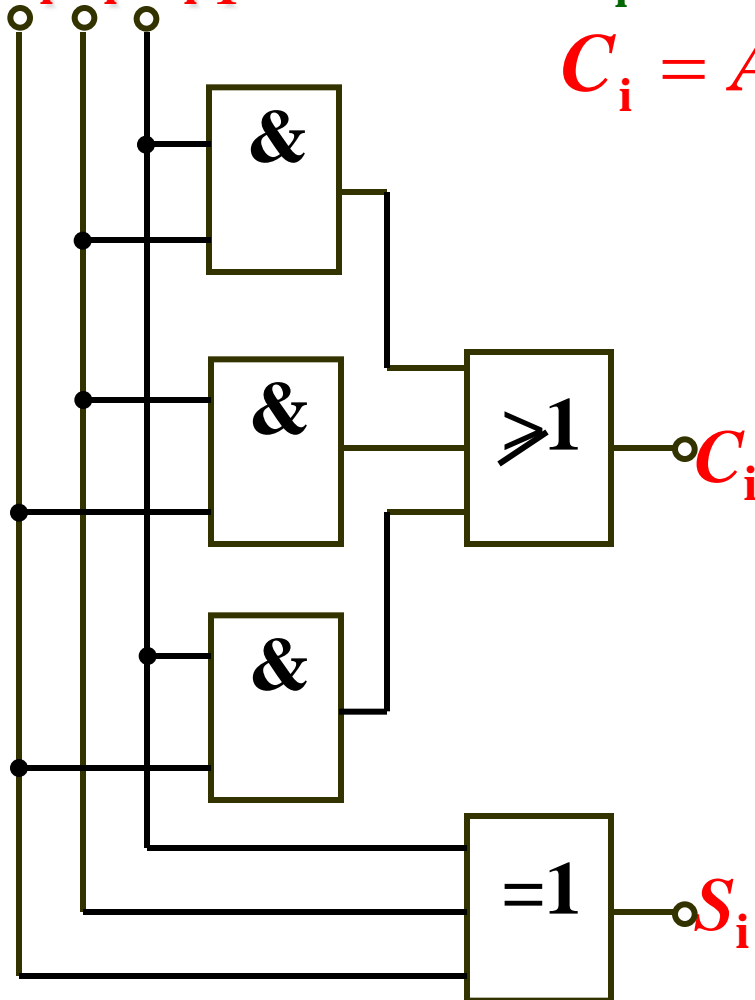
$$\begin{aligned} S_i &= \overline{A_i} \overline{B_i} C_{i-1} + \overline{A_i} B_i \overline{C_{i-1}} + A_i \overline{B_i} \overline{C_{i-1}} + A_i B_i C_{i-1} \\ &= A_i \oplus B_i \oplus C_{i-1} \end{aligned}$$

$$\begin{aligned} C_i &= \overline{A_i} B_i C_{i-1} + A_i \overline{B_i} C_{i-1} + A_i B_i \overline{C_{i-1}} + A_i B_i C_{i-1} \\ &= A_i B_i + B_i C_{i-1} + A_i C_{i-1} \end{aligned}$$

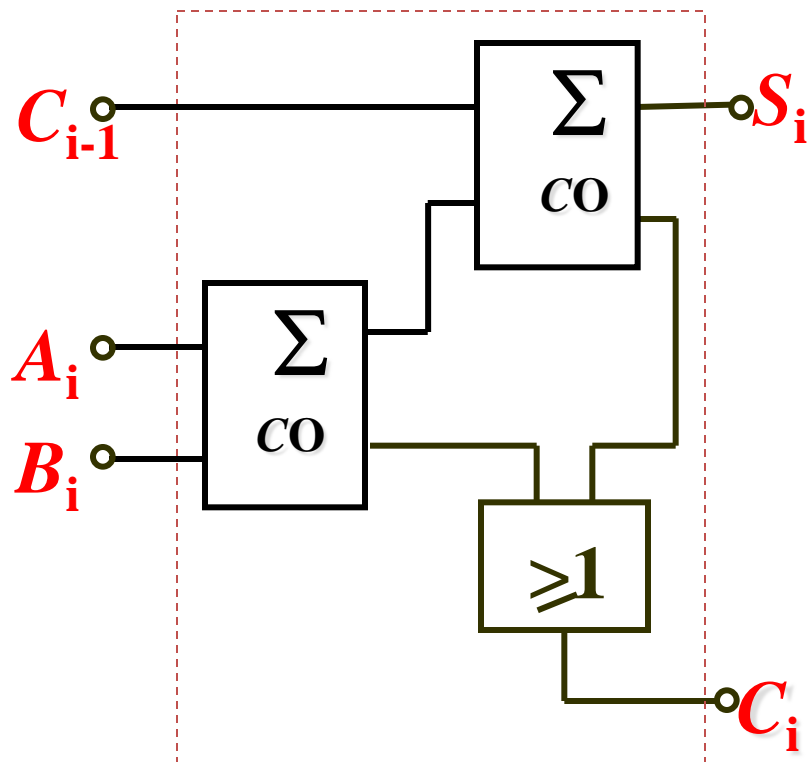
$A_i B_i C_{i-1}$

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

$$C_i = A_i B_i + B_i C_{i-1} + A_i C_{i-1}$$



逻辑图

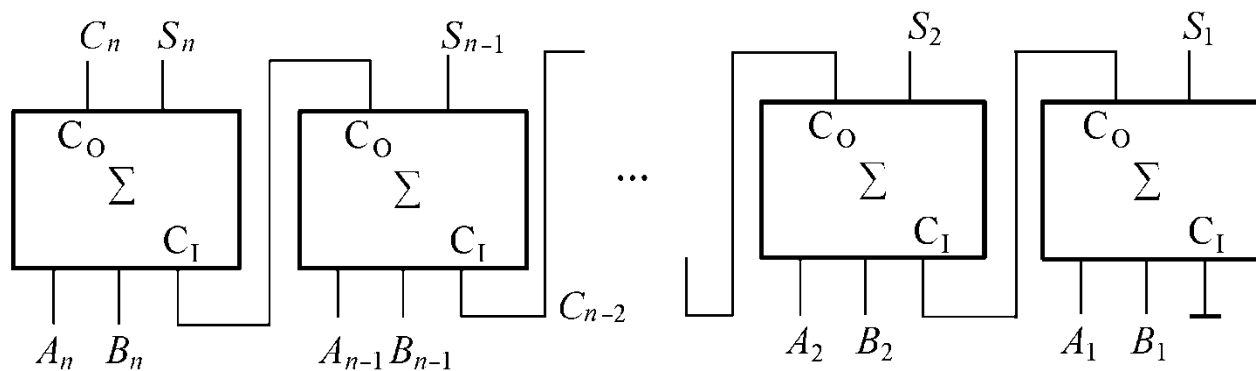


半加器构成的全加器

N位全加器

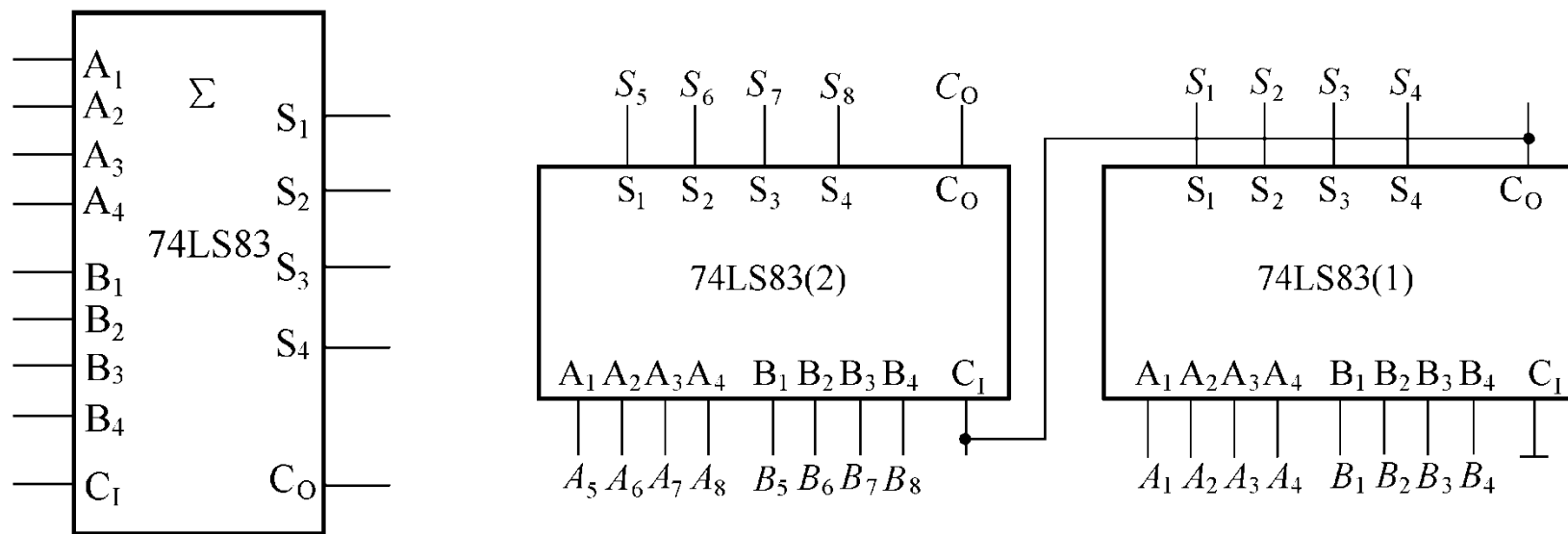
被加数 A	A_n	A_{n-1}	A_{n-2}	\cdot	\cdot	\cdot	A_2	A_1
加数 B	B_n	B_{n-1}	B_{n-2}	\cdot	\cdot	\cdot	B_2	B_1
低位向相邻高位的进位	C_{n-1}	C_{n-2}	C_{n-3}	\cdot	\cdot	\cdot	C_1	
进位数 C	C_n	C_{n-1}	C_{n-2}	\cdot	\cdot	\cdot	C_2	C_1
和数 S	S_n	S_{n-1}	S_{n-2}	\cdot	\cdot	\cdot	S_2	S_1

将 n 个一位全加器串接起来



4位集成加法器及其扩展

全加器的功能扩展：集成全加器最多为4位，如果要构成8位或者更多位的加法器就必须由多片4位全加器串接(Series)而成。



2.2 编码器(Encoder)



把二进制码按一定规律编排，使每组代码具有一特定的含义，称为编码。具有编码功能的逻辑电路称为编码器。

n 位二进制代码有 2^n 种组合，可以表示 2^n 个信息。

要表示 N 个信息所需的二进制代码应满足

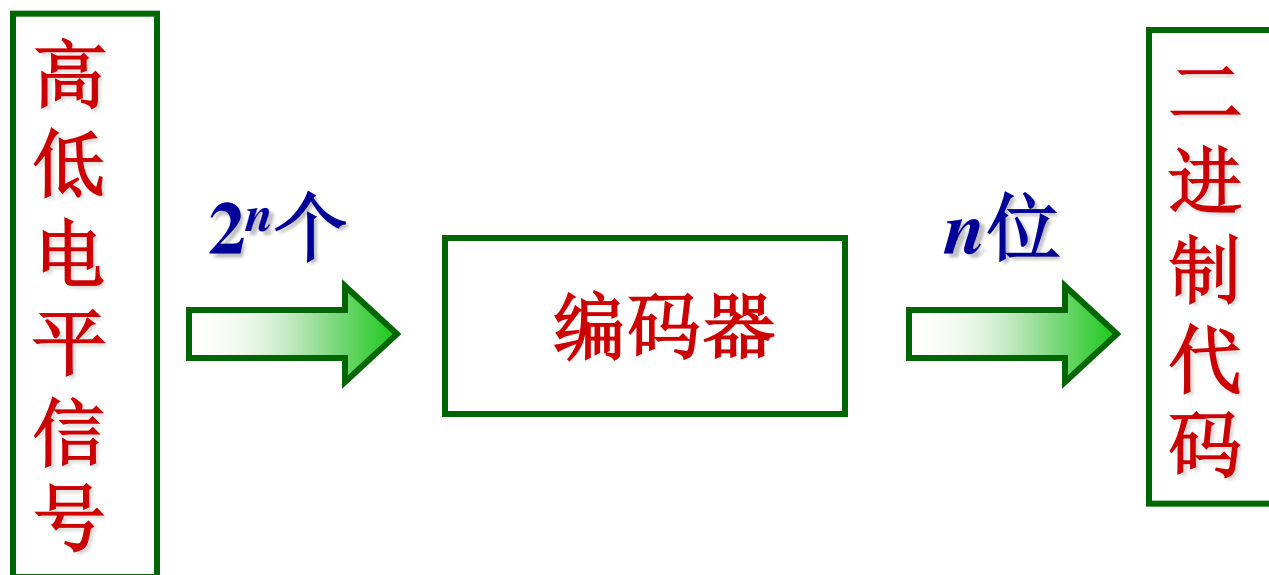
$$2^n \geq N$$



2.2.1 二进制编码器



将输入信号编成二进制代码的电路。



例：设计一个编码器，满足以下要求：

- (1) 将 I_0 、 I_1 、... I_7 8个信号编成二进制代码。
- (2) 编码器每次只能对一个信号进行编码，不允许两个或两个以上的信号同时有效。
- (3) 设输入信号高电平有效。

(1) 分析要求：

输入有8个信号，即 $N=8$ ，根据 $2^n \geq N$ 的关系，即 $n=3$ ，即输出为三位二进制代码。

(2) 列编码表:



输入	输出		
	Y_2	Y_1	Y_0
I_0	0	0	0
I_1	0	0	1
I_2	0	1	0
I_3	0	1	1
I_4	1	0	0
I_5	1	0	1
I_6	1	1	0
I_7	1	1	1

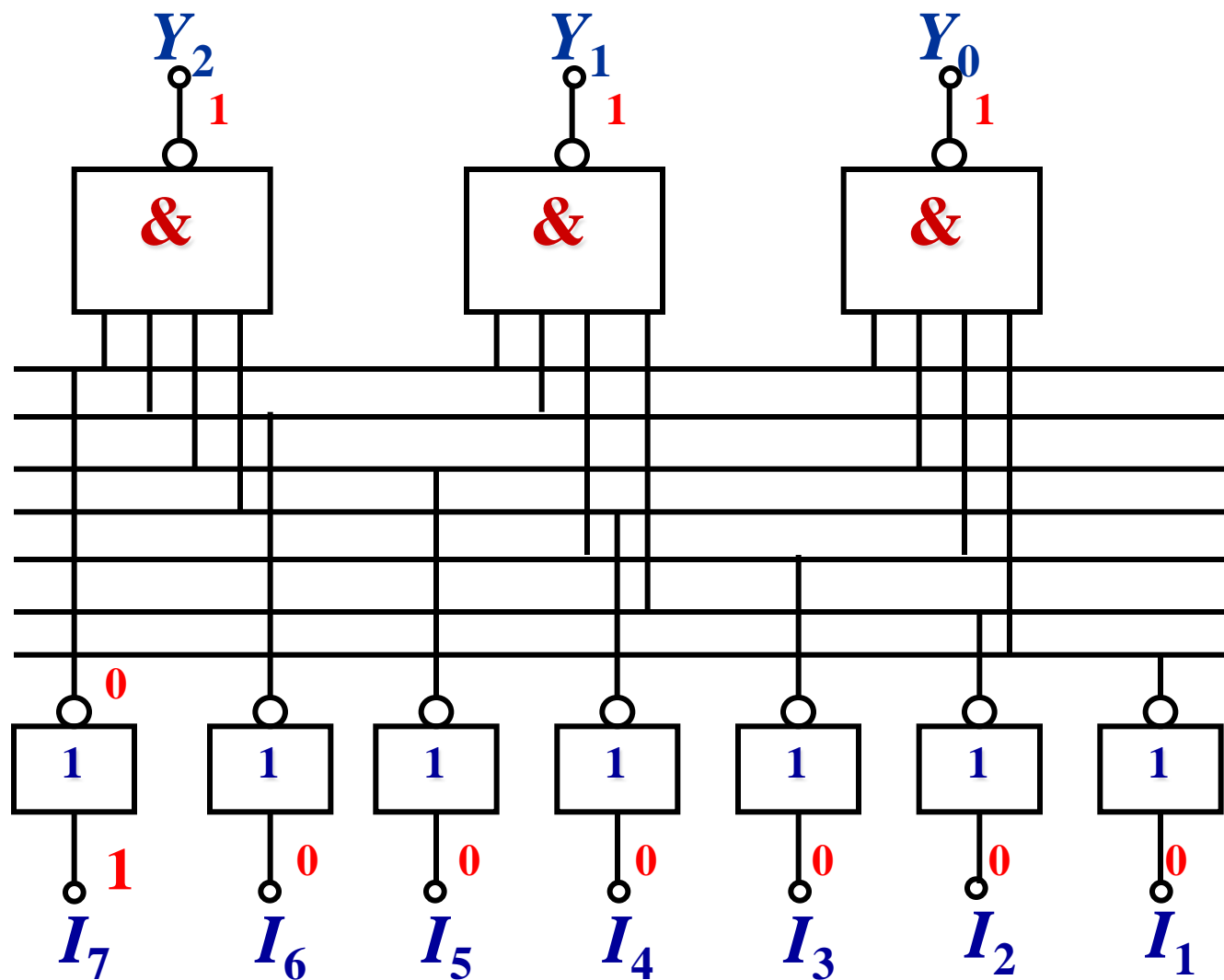
(3) 写出逻辑式并转换成“与非”式

$$\begin{aligned} Y_2 &= I_4 + I_5 + I_6 + I_7 = \overline{\overline{I_4 + I_5 + I_6 + I_7}} \\ &= \overline{\overline{I_4} \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7}} \end{aligned}$$

$$\begin{aligned} Y_1 &= I_2 + I_3 + I_6 + I_7 = \overline{\overline{I_2 + I_3 + I_6 + I_7}} \\ &= \overline{\overline{I_2} \cdot \overline{I_3} \cdot \overline{I_6} \cdot \overline{I_7}} \end{aligned}$$

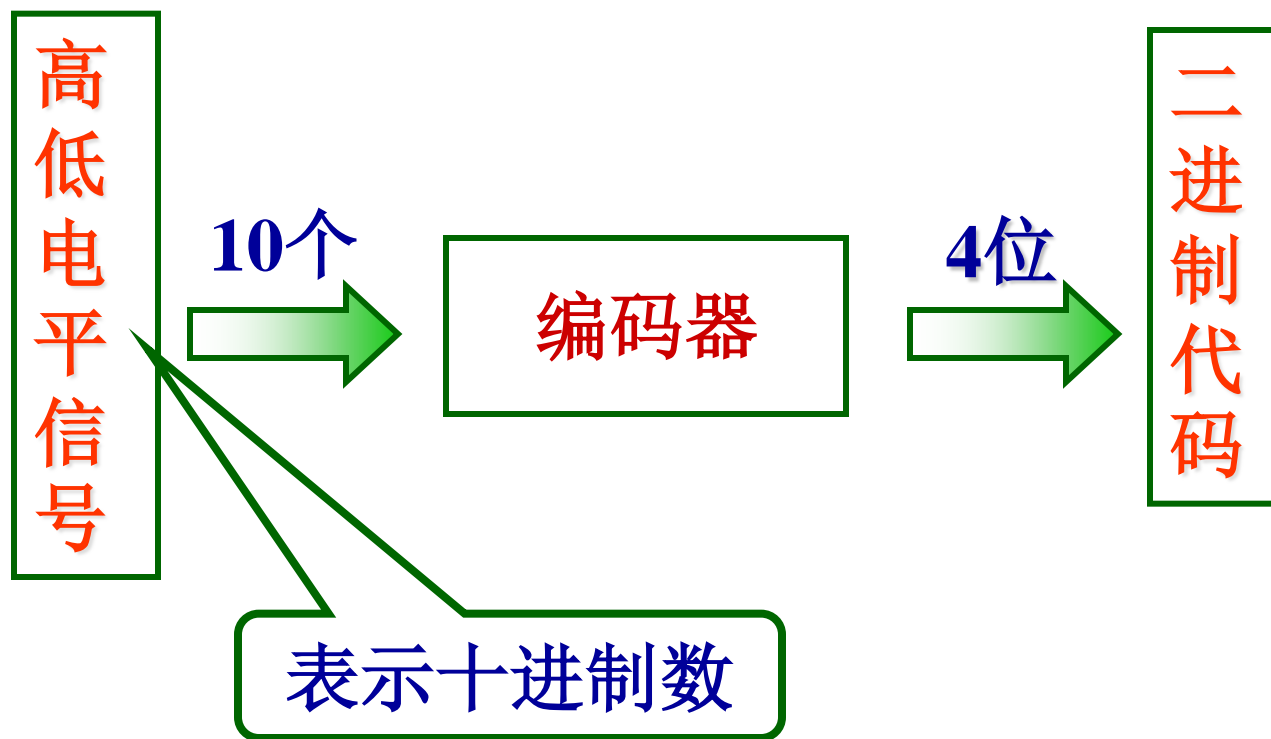
$$\begin{aligned} Y_0 &= I_1 + I_3 + I_5 + I_7 = \overline{\overline{I_1 + I_3 + I_5 + I_7}} \\ &= \overline{\overline{I_1} \cdot \overline{I_3} \cdot \overline{I_5} \cdot \overline{I_7}} \end{aligned}$$

(4) 画出逻辑图



2.2.2 二 - 十进制编码器

将十进制数 0~9 编成二进制代码的电路



列编码表:

四位二进制代码
可以表示十六种
不同的状态，其
中任何十种状态
都可以表示0~9
十个数码，最常
用的是8421码。

8421BCD码编码表

输 入	输 出			
	Y_3	Y_2	Y_1	Y_0
0 (I_0)	0	0	0	0
1 (I_1)	0	0	0	1
2 (I_2)	0	0	1	0
3 (I_3)	0	0	1	1
4 (I_4)	0	1	0	0
5 (I_5)	0	1	0	1
6 (I_6)	0	1	1	0
7 (I_7)	0	1	1	1
8 (I_8)	1	0	0	0
9 (I_9)	1	0	0	1

写出逻辑式并化成“或非”门和“与非”门

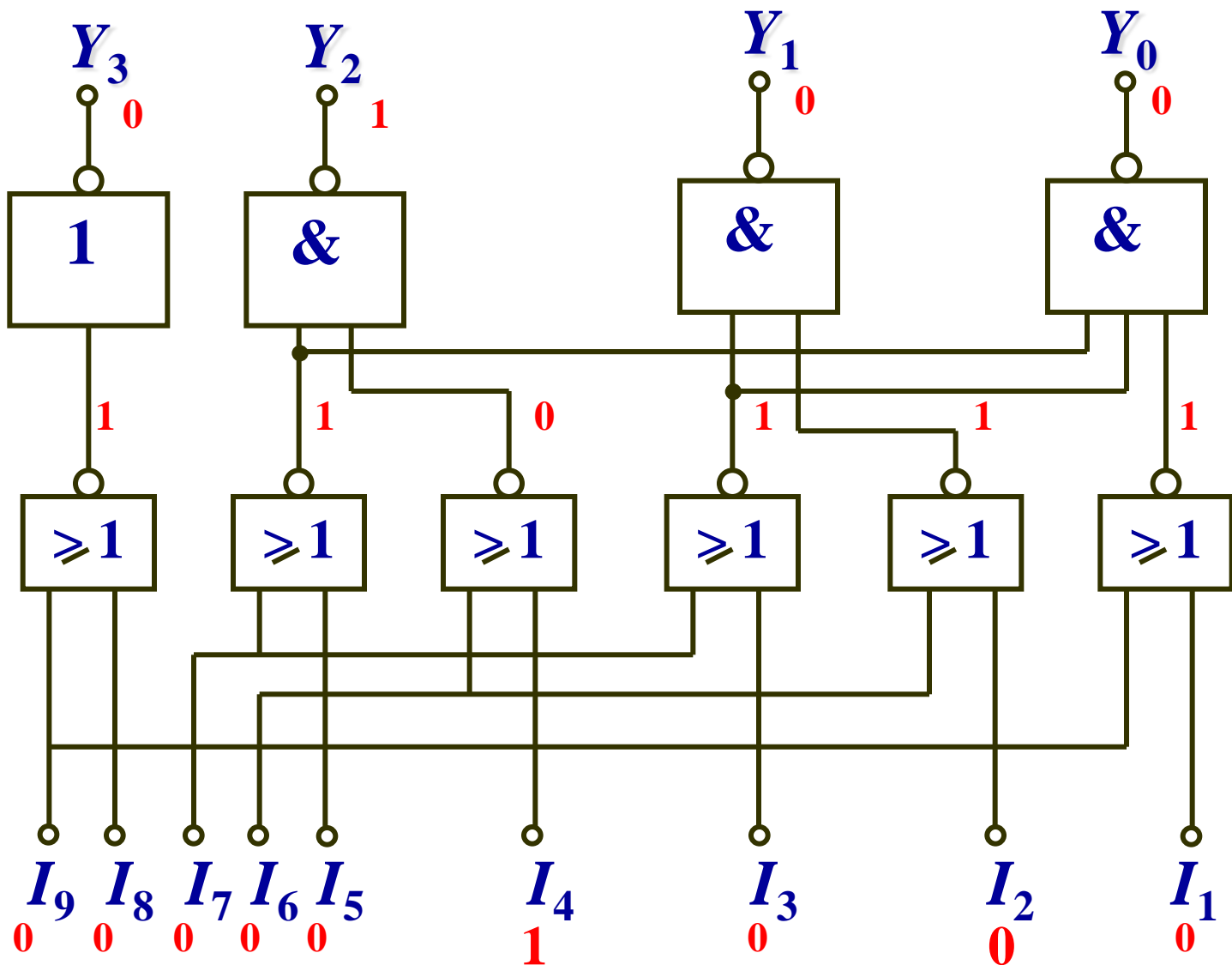
$$Y_3 = \overline{\overline{I_8 + I_9}}$$

$$Y_2 = \overline{\overline{I_4 + I_5 + I_6 + I_7}} = \overline{\overline{I_4 + I_6}} \cdot \overline{\overline{I_5 + I_7}}$$

$$Y_1 = \overline{\overline{I_2 + I_3 + I_6 + I_7}} = \overline{\overline{I_2 + I_6}} \cdot \overline{\overline{I_3 + I_7}}$$

$$\begin{aligned} Y_0 &= \overline{\overline{I_1 + I_3 + I_5 + I_7 + I_9}} \\ &= \overline{\overline{I_1 + I_9}} \cdot \overline{\overline{I_3 + I_7}} \cdot \overline{\overline{I_5 + I_7}} \end{aligned}$$

画出逻辑图



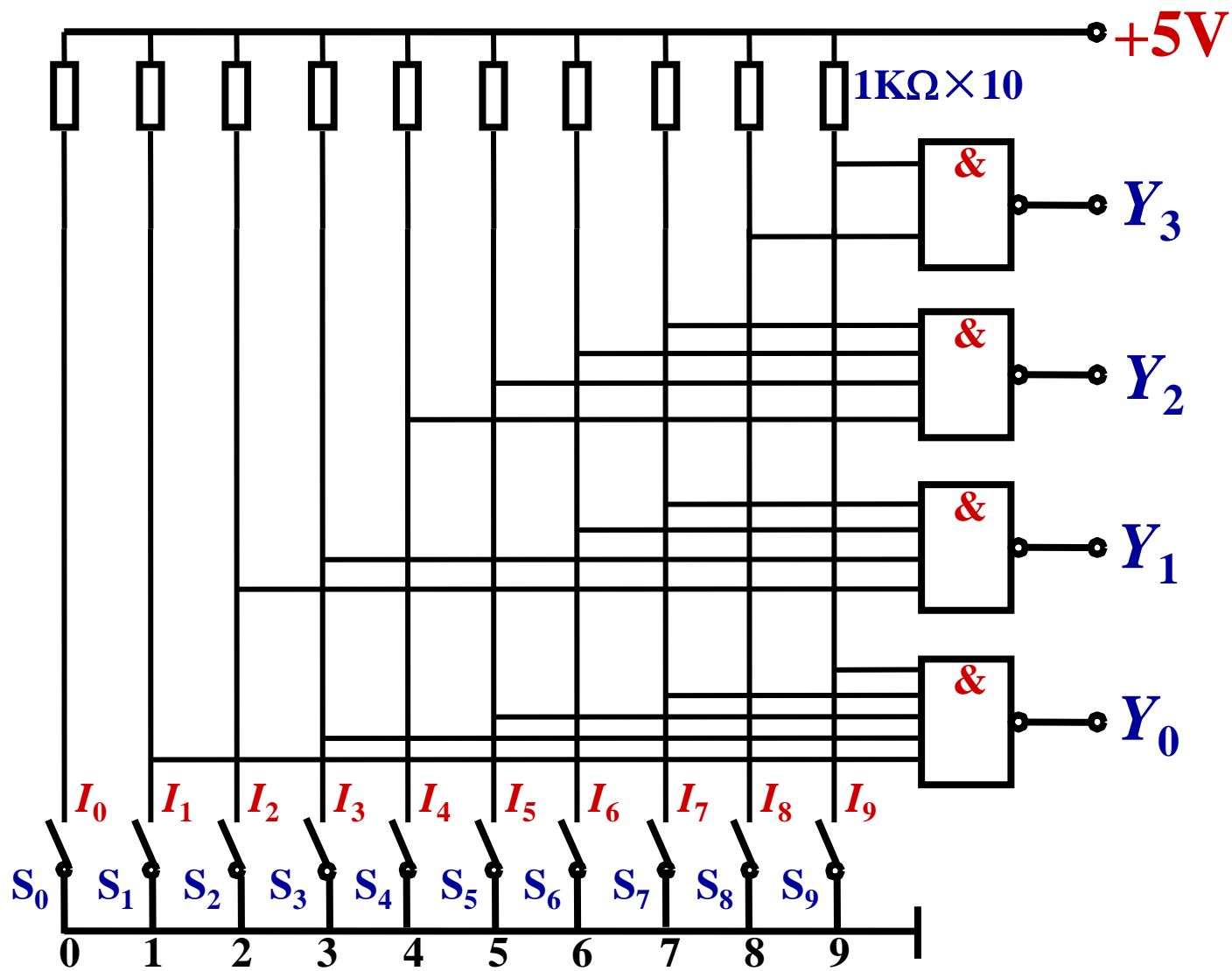
法二：

$$Y_3 = \overline{\overline{I_8 + I_9}} = \overline{\overline{I_8} \cdot \overline{I_9}}$$

$$Y_2 = \overline{\overline{I_4 + I_5 + I_6 + I_7}} = \overline{\overline{I_4} \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7}}$$

$$Y_1 = \overline{\overline{I_2 + I_3 + I_6 + I_7}} = \overline{\overline{I_2} \cdot \overline{I_3} \cdot \overline{I_6} \cdot \overline{I_7}}$$

$$\begin{aligned} Y_0 &= \overline{\overline{I_1 + I_3 + I_5 + I_7 + I_9}} \\ &= \overline{\overline{I_1} \cdot \overline{I_3} \cdot \overline{I_5} \cdot \overline{I_7} \cdot \overline{I_9}} \end{aligned}$$



十键8421码编码器的逻辑图

2.2.3 优先编码器



当有**两个或两个以上**的信号同时输入编码电路，电路只能对其中一个优先级别高的信号进行编码。

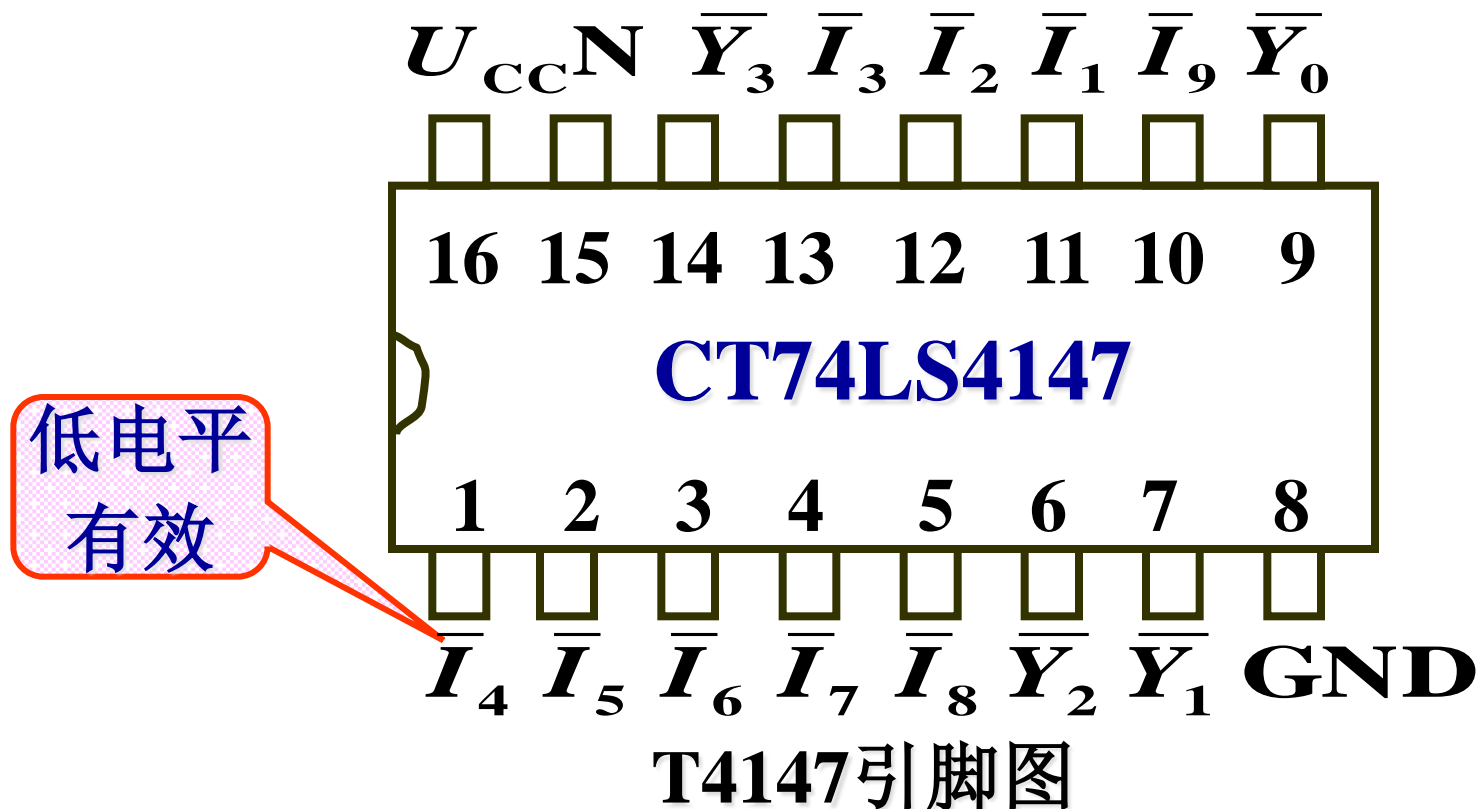
即允许几个信号同时有效，但电路只对其中优先级别高的信号进行编码，而对其它优先级别低的信号不予理睬。

常用的优先编码器有8线—3线(74LS148、CT54LS148等)，10线—4线8421BCD优先编码器(74LS147、CT54LS147、CC40147等)。

CT74LS4147 编码器功能表

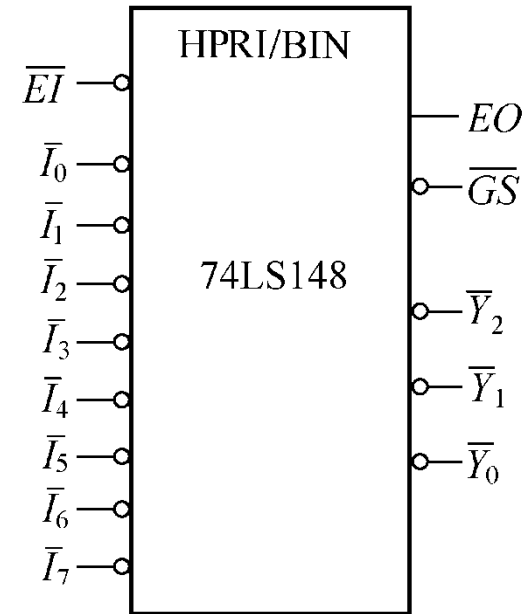
输 入 (低电平有效)									输 出(8421反码)			
$\overline{I_9}$	$\overline{I_8}$	$\overline{I_7}$	$\overline{I_6}$	$\overline{I_5}$	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
1	1	1	1	1	1	1	1	1	1	1	1	1
0	×	×	×	×	×	×	×	×	0	1	1	0
1	0	×	×	×	×	×	×	×	0	1	1	1
1	1	0	×	×	×	×	×	×	1	0	0	0
1	1	1	0	×	×	×	×	×	1	0	0	1
1	1	1	1	0	×	×	×	×	1	0	1	0
1	1	1	1	1	0	×	×	×	1	0	1	1
1	1	1	1	1	1	0	×	×	1	1	0	0
1	1	1	1	1	1	1	0	×	1	1	0	1
1	1	1	1	1	1	1	1	0	1	1	1	0

例: CT74LS147集成优先编码器(10线-4线)



74LS148器件管脚(pin)描述

\overline{EI} 为使能输入端，低电平有效；
 $\overline{I_0} \sim \overline{I_7}$ 是输入信号，低电平有效；
 $\overline{Y_2} \sim \overline{Y_0}$ 是输出信号，采用反码输出；
为了扩展器件功能还增设有：
输出使能端 EO ，编码选择端 \overline{GS} 。



(b)

74LS148真值表

[illegible]

编码器的功能扩展

采用2片74LS148将8线—3线优先编码器扩展为16线—4线优先编码器。

其中：

$\overline{I}_{15} \sim \overline{I}_0$ 为输入信号，

$\overline{Y}_3 \sim \overline{Y}_0$ 为输出信号，

\overline{S} 为扩展编码器使能输入端。

