

2019-2020学年秋季学期

第四部分-A 组合逻辑电路

○数字电路

○组合逻辑电路

Combinational Logic

在任意时刻，电路的输出仅决定于该时刻的输入，与电路原来的状态无关。

○时序逻辑电路

Sequential Logic

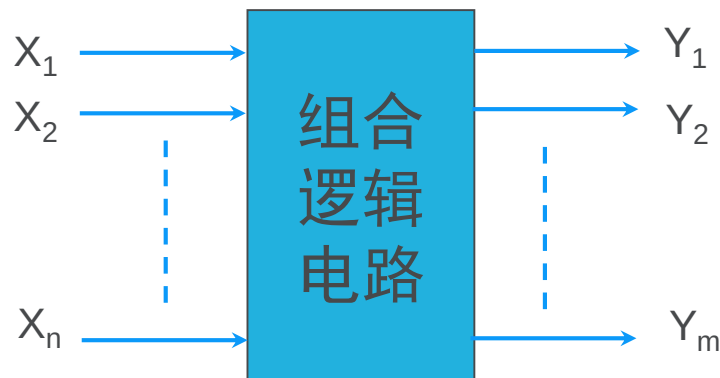
电路的输出不仅受该时刻输入的影响，还受电路原来状态的影响（包含内部存储）。

○组合逻辑电路

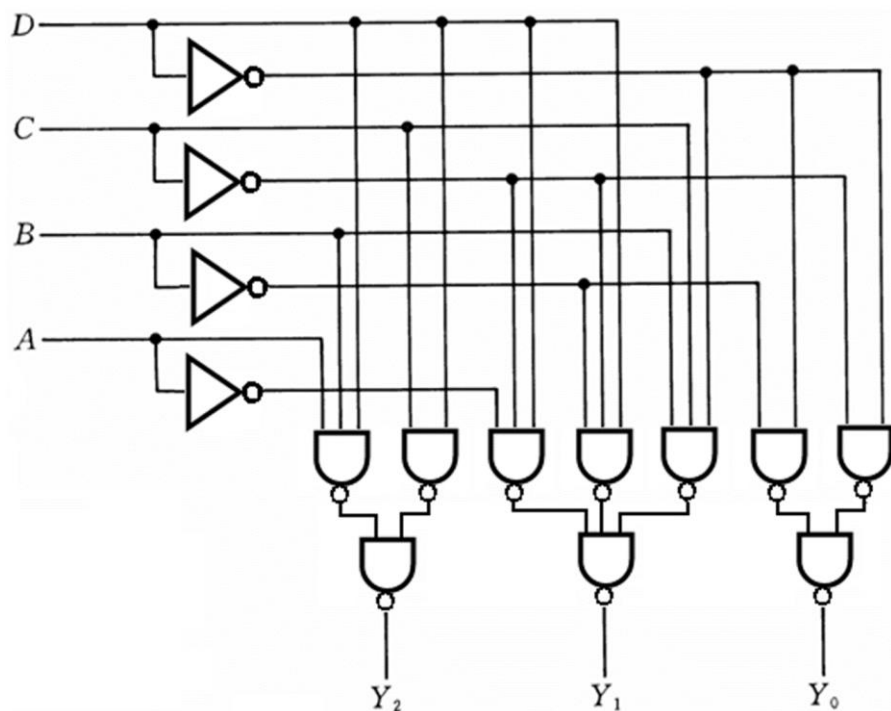
○多输入多输出的逻辑函数

○不包含内部存储

$$Y = F(X)$$



组合逻辑电路的分析



$$Y_0 = \bar{B}\bar{D} + \bar{C}\bar{D}$$

$$Y_1 = \bar{A}\bar{C}D + \bar{B}\bar{C}D + BCD$$

$$Y_2 = ABD + CD$$

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
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

组合逻辑电路的基本设计方法

○逻辑抽象

将用文字描述的问题抽象为逻辑函数：

- 分析因果关系，确定输入和输出变量。
- 对输入和输出变量进行二进制编码。
- 列出真值表。

○写出逻辑表达式

○化简逻辑表达式

○将逻辑表达式转化为电路图

交通灯故障监视—逻辑抽象

○问题描述：

假设一组交通灯，由红黄绿三盏灯组成。正常工作情况下，任何时候有且仅有一盏灯点亮。如果出现没有灯点亮或者多于一盏灯点亮则为故障。当交通灯故障时，一个故障信号开启，提示维修人员检查。请设计该故障信号。

○分析：

- 设定红黄绿为R(red), A(amber), G(green)三个逻辑变量，故障为Z。
- 那么，正常状态包括： $R\bar{A}\bar{G}$ $\bar{R}A\bar{G}$ $\bar{R}\bar{A}G$
- 故障状态包括： RAG $RA\bar{G}$ $\bar{R}AG$ $\bar{R}\bar{A}\bar{G}$

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

交通灯故障监视—逻辑表达式及其化简

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

$$Z = RAG + RA\bar{G} + R\bar{A}G + \bar{R}AG + \bar{R}\bar{A}\bar{G}$$

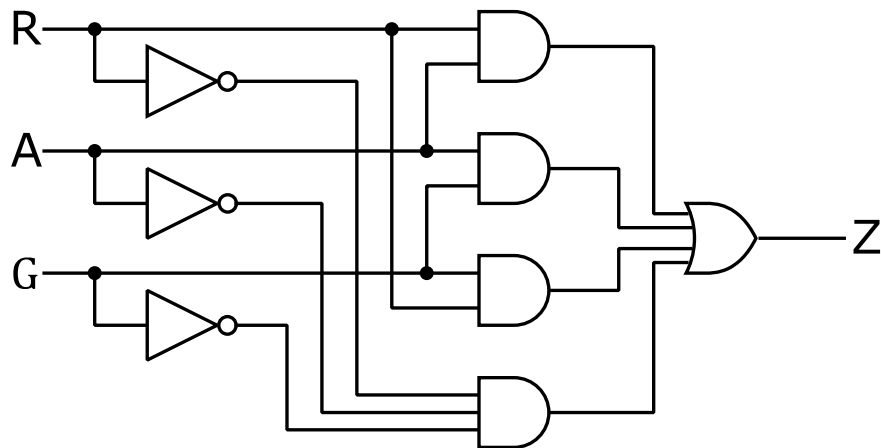
$$Z = RA(G + \bar{G}) + RG(A + \bar{A}) + AG(R + \bar{R}) + \bar{R}\bar{A}\bar{G}$$

$$Z = RA + RG + AG + \bar{R}\bar{A}\bar{G}$$

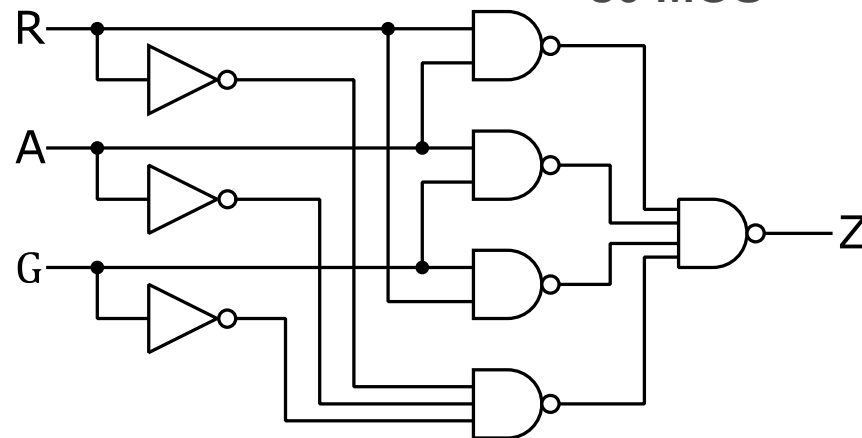
交通灯故障监视—电路图 (最小项)

$$Z = RA + RG + AG + \overline{R}\overline{A}\overline{G}$$

40 MOS

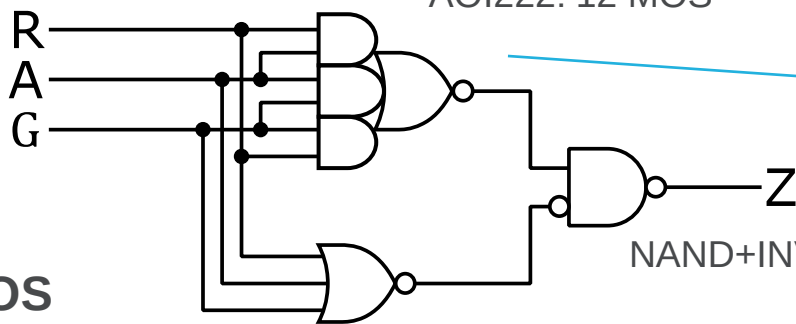


30 MOS



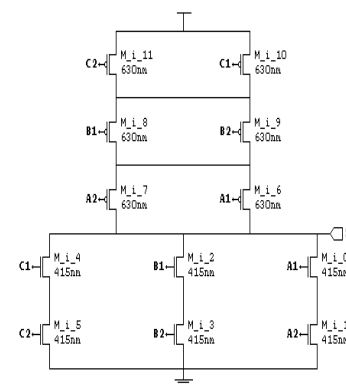
AOI222: 12 MOS

24 MOS



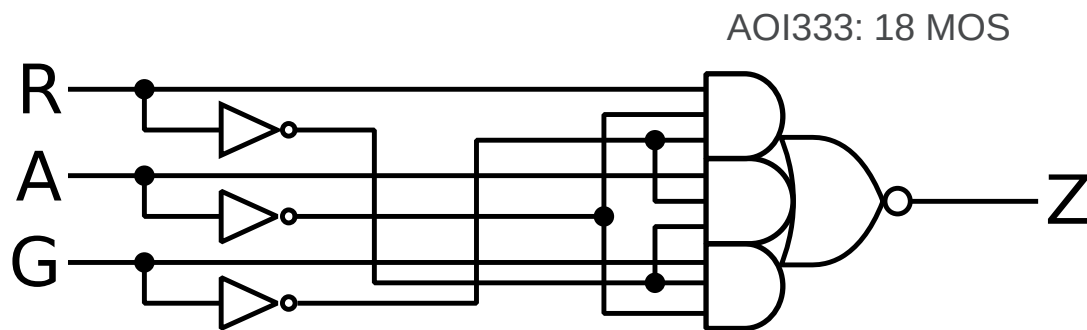
NAND+INV: 6 MOS

NOR3: 6 MOS

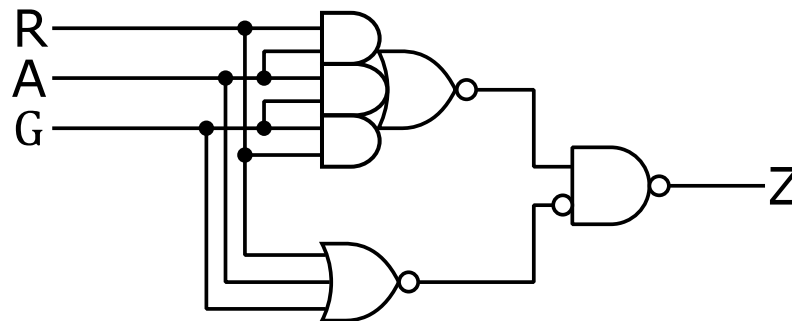


交通灯故障监视—电路图 (最大项)

$$Z = \overline{R\bar{A}\bar{G}} + \bar{R}A\bar{G} + \bar{R}\bar{A}G$$



24 MOS



组合逻辑电路总结

- 组合逻辑电路的定义
 - 无内部（存储）状态的数字逻辑电路
- 组合逻辑电路的分析
 - 真值表
 - 电路拆分
- 组合逻辑电路的设计
 - 逻辑抽象
 - 写出逻辑表达式
 - 化简逻辑表达式
 - 将逻辑表达式转化为电路图

○编码器：信号到二进制编码

- 普通编码器
- 优先编码器 (148)
- 优先译码器 (147)

○译码器：二进制到信号

- 二进制译码器 (138)
- 二—十译码器 (42)
- BCD—七段译码器 (48)

○运算单元：

- 选择器 (153)
- 加法器 (283)
- 比较器 (85)

编码器：普通编码器

○编码

○用数字或符号来表示某一对象或信号的过程称为编码

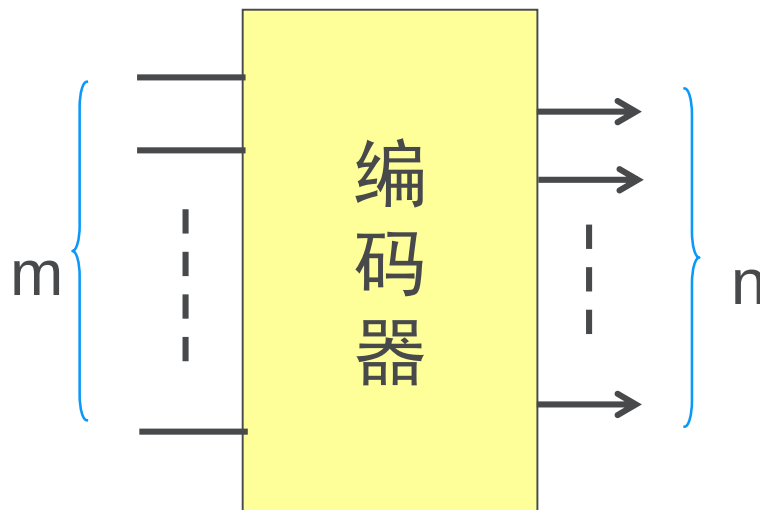
○二进制编码器

○将十进制的十个数0、1、2...9编成二进制编码

○n位二进制代码可以表示 2^n 个信号

○m是输入量，它是数、字符、算符的状态信号，即高、低电平。

○ $m \leq 2^n$



编码器：普通编码器—8线-3线编码器

3位二进制（8线—3线）编码器真值表

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

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

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

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

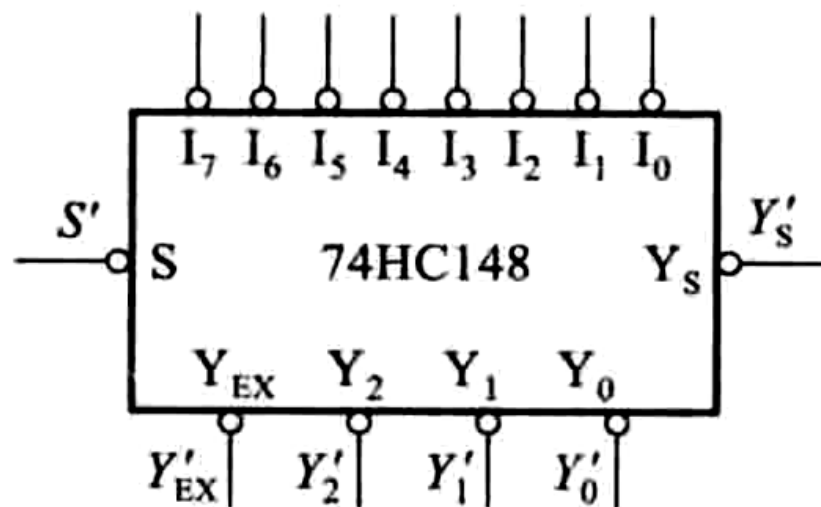
任何时刻只允许输入一个编码信号，否则输出将发生混乱。

编码器：优先编码器（148）

允许多个输入信号同时有效，但仅按其中优先级别最高的有效输入信号编码。

○74HC148 (T4148)：8线二进制编码

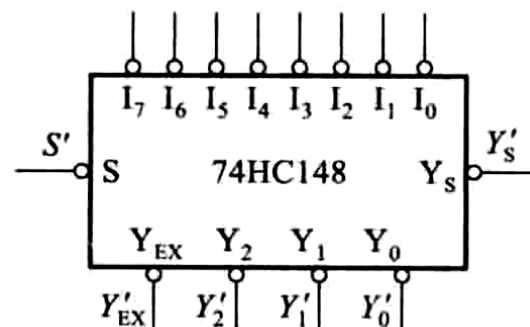
- $I'_7 \sim I'_0$ ：信号输入端，低有效， I'_7 为最高位。
- $Y'_2 \sim Y'_0$ ：信号输出端，低有效， Y'_2 为最高位。
- S' ：选通输入端，低有效。
- Y'_S ：选通输出端，低有效。
- Y'_{EX} ：优先标志输出端，低有效。



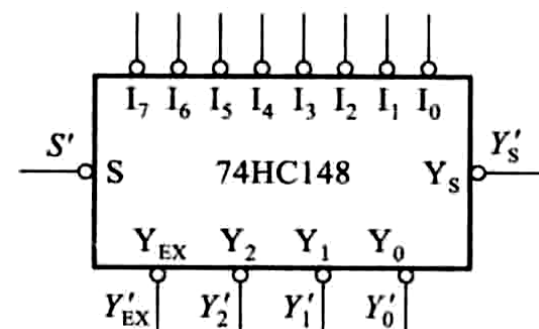
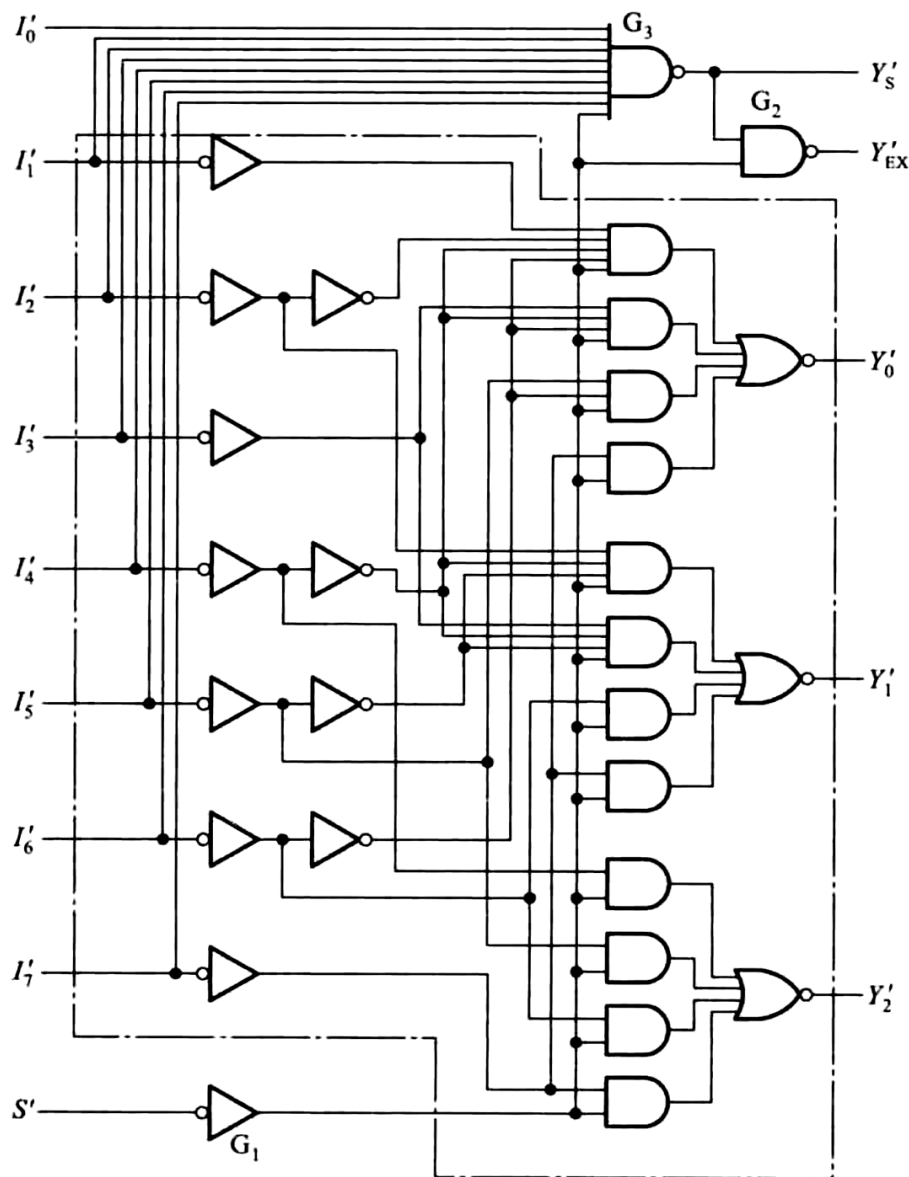
编码器：优先编码器（148）—真值表

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

- $I_7 \sim I_0$ ：信号输入端，低有效， I_7 为最高位。
- $Y_2 \sim Y_0$ ：信号输出端， Y_2 为最高位。
- S ：选通输入端。
- Y_S ：选通输出端。
- Y_{EX} ：优先标志输出端。



编码器：优先编码器（148）—电路图



$$Y_0' = S' + \overline{I_7'} + \overline{I_5'} I_6' + \overline{I_3'} I_4' I_6' + \overline{I_1'} I_2' I_4' I_6'$$

$$Y_1' = S' + \overline{I_7'} + \overline{I_6'} + \overline{I_3'} I_4' I_5' + \overline{I_2'} I_4' I_5'$$

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

$$Y_S' = S' + \overline{I_7'} I_6' I_5' I_4' I_3' I_2' I_1' I_0'}$$

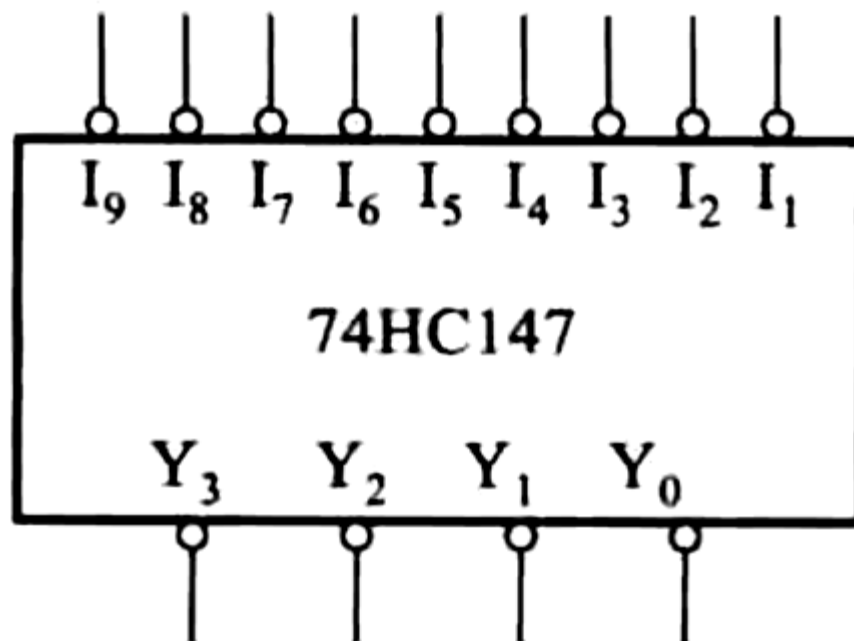
$$Y_{EX}' = S' + \overline{I_7'} \overline{I_6'} \overline{I_5'} \overline{I_4'} \overline{I_3'} \overline{I_2'} \overline{I_1'} \overline{I_0'}$$

编码器：优先编码器（147）

○74HC147: 10线二进制编码

○ $I'_9 \sim I'_1$: 信号输入端，低有效， I'_9 为最高位。

○ $Y'_3 \sim Y'_0$: 信号输出端， Y'_3 为最高位。



编码器：优先编码器（147）—真值表

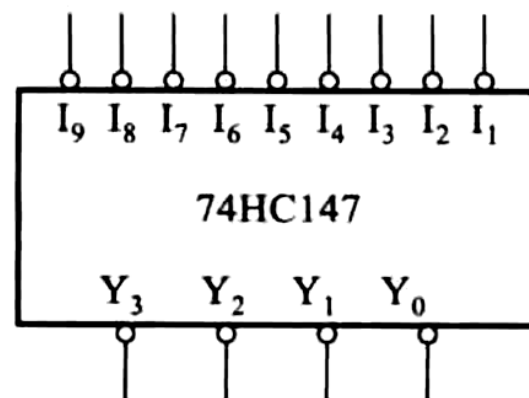
输入									输出			
I'_1	I'_2	I'_3	I'_4	I'_5	I'_6	I'_7	I'_8	I'_9	Y'_3	Y'_2	Y'_1	Y'_0
1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x	0	0	1	1	0
x	x	x	x	x	x	x	0	1	0	1	1	1
x	x	x	x	x	x	0	1	1	1	0	0	0
x	x	x	x	x	0	1	1	1	1	0	0	1
x	x	x	x	0	1	1	1	1	1	0	1	0
x	x	x	0	1	1	1	1	1	1	0	1	1
x	x	0	1	1	1	1	1	1	1	1	0	0
x	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0

$$Y'_0 = \overline{I'_9} + \overline{I'_7}I'_8I'_9 + \overline{I'_5}I'_6I'_8I'_9 + \overline{I'_3}I'_4I'_6I'_8I'_9 + \overline{I'_1}I'_2I'_4I'_6I'_8I'_9$$

$$Y'_1 = \overline{I'_7}I'_8I'_9 + \overline{I'_6}I'_8I'_9 + \overline{I'_3}I'_4I'_5I'_8I'_9 + \overline{I'_2}I'_4I'_5I'_8I'_9$$

$$Y'_2 = \overline{I'_7}I'_8I'_9 + \overline{I'_6}I'_8I'_9 + \overline{I'_5}I'_8I'_9 + \overline{I'_4}I'_8I'_9$$

$$Y'_3 = \overline{I'_9} + \overline{I'_8}$$



问题： I'_0 在哪里？

编码器：优先编码器（147）——真值表

输入									输出			
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
1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x	0	0	1	1	0
x	x	x	x	x	x	x	0	1	0	1	1	1
x	x	x	x	x	x	0	1	1	1	0	0	0
x	x	x	x	x	0	1	1	1	1	0	0	1
x	x	x	x	0	1	1	1	1	1	0	1	0
x	x	x	0	1	1	1	1	1	1	0	1	1
x	x	0	1	1	1	1	1	1	1	1	0	0
x	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

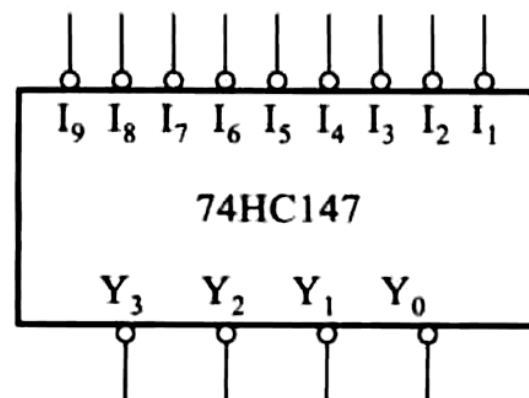
$$Y'_0 = \overline{I'_9} + \overline{I'_7}I'_8I'_9 + \overline{I'_5}I'_6I'_8I'_9 + \overline{I'_3}I'_4I'_6I'_8I'_9 + \overline{I'_1}I'_2I'_4I'_6I'_8I'_9$$

$$Y'_1 = \overline{I'_7}I'_8I'_9 + \overline{I'_6}I'_8I'_9 + \overline{I'_3}I'_4I'_5I'_8I'_9 + \overline{I'_2}I'_4I'_5I'_8I'_9$$

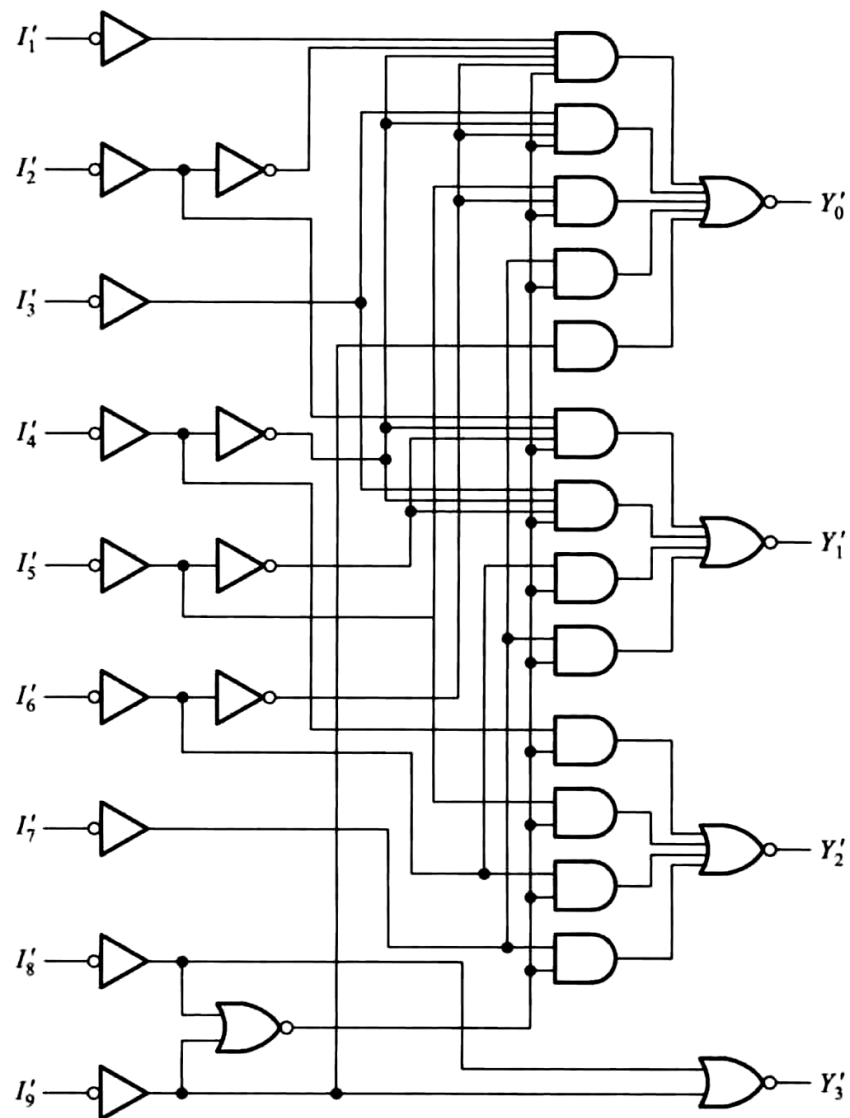
$$Y'_2 = \overline{I'_7}I'_8I'_9 + \overline{I'_6}I'_8I'_9 + \overline{I'_5}I'_8I'_9 + \overline{I'_4}I'_8I'_9$$

$$Y'_3 = \overline{I'_9} + \overline{I'_8}$$

问题： I'_0 在哪里？



编码器：优先编码器（147）— 电路图

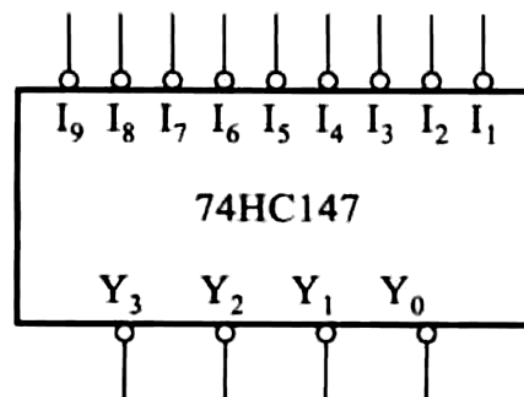


$$Y'_0 = \overline{I'_9} + \overline{I'_7}I'_8I'_9 + \overline{I'_5}I'_6I'_8I'_9 + \overline{I'_3}I'_4I'_6I'_8I'_9 + \overline{I'_1}I'_2I'_4I'_6I'_8I'_9$$

$$Y'_1 = \overline{I'_7}I'_8I'_9 + \overline{I'_6}I'_8I'_9 + \overline{I'_3}I'_4I'_5I'_8I'_9 + \overline{I'_2}I'_4I'_5I'_8I'_9$$

$$Y'_2 = \overline{I'_7}I'_8I'_9 + \overline{I'_6}I'_8I'_9 + \overline{I'_5}I'_8I'_9 + \overline{I'_4}I'_8I'_9$$

$$Y'_3 = \overline{I'_9} + \overline{I'_8}$$



编码器总结

○普通编码器

- 将多路信号变成二进制信号
- 只有当多路信号为独热编码 (one-hot) 时候, 才能正确工作

○优先编码器 (148)

- 解决了独热码的问题
- 优先输出最大的二进制编码
- 8线到3位二进制编码
- 通过使能端解决二进制0编码歧义的问题

○优先译码器 (147)

- 10线到4位二进制编码
- 没有编码歧义问题 ($2^4 > 10$)

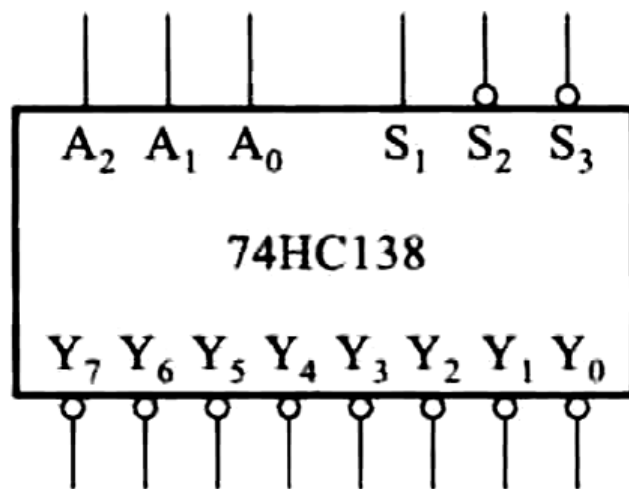
○信号的低有效

- 历史原因
- 注意符号和使用

译码器：二进制译码器 (138)

○74HC138: 二进制译码器

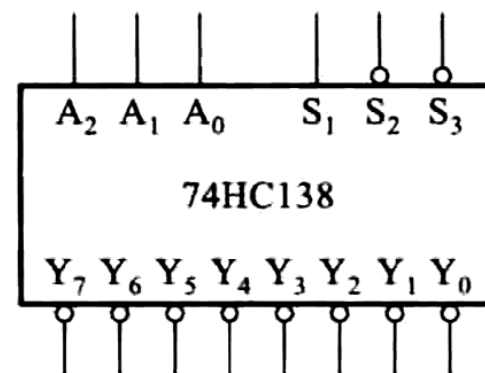
- $A_2 \sim A_0$: 信号输入端, 高有效, A_2 为最高位。
- S_1, S_2', S_3' : 三个使能信号, 当 $1 = S_1 \cdot \overline{S_2'} \cdot \overline{S_3'}$ 时有效。
- $Y_7' \sim Y_0'$: 信号输出端, 低有效, Y_7' 为最高位。
- 把3比特的 A 翻译为对应的输出信号 $\overline{Y_A}$ 。
- 和74HC148的功能互逆。



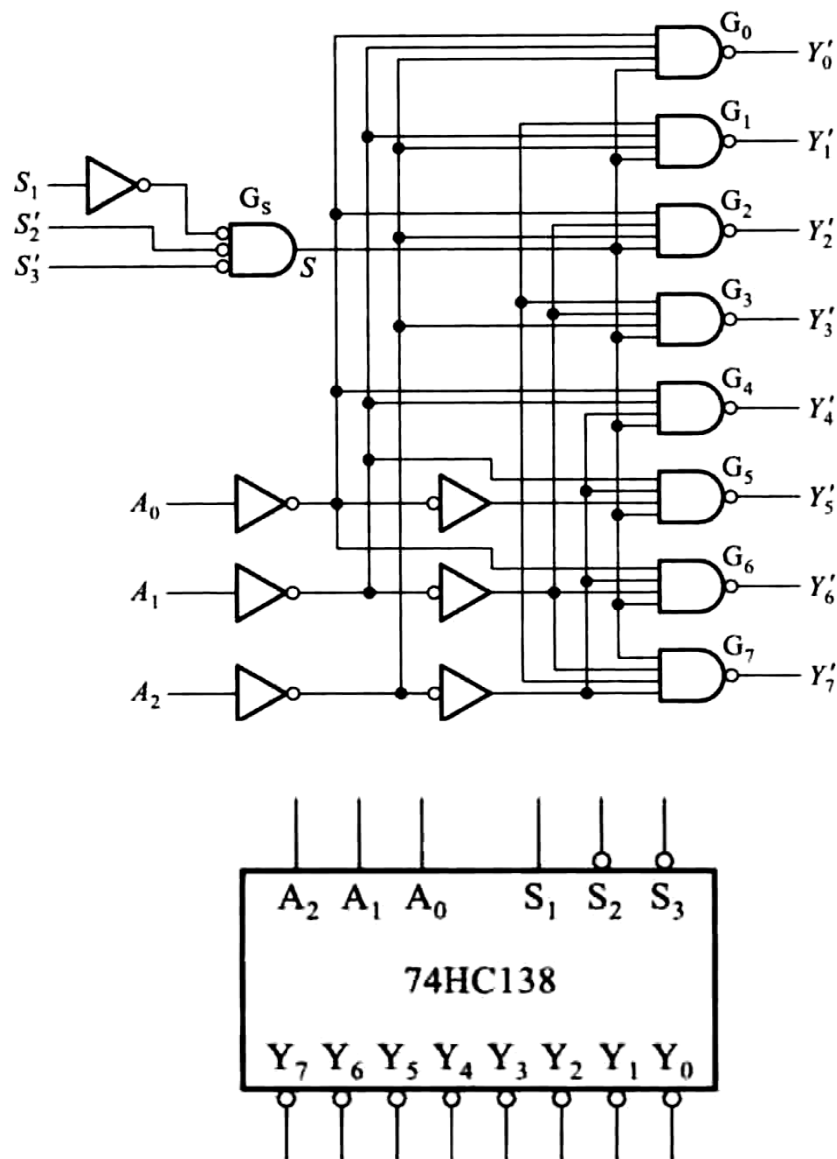
译码器：二进制译码器 (138) —真值表

输入					输出							
S_1	$S'_2 + S'_3$	A_2	A_1	A_0	Y'_0	Y'_1	Y'_2	Y'_3	Y'_4	Y'_5	Y'_6	Y'_7
0	×	×	×	×	1	1	1	1	1	1	1	1
×	1	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

- $A_2 \sim A_0$: 信号输入端，低有效， A_2 为最高位。
- S_1, S'_2, S'_3 : 三个使能信号，当 $1 = S_1 \cdot \overline{S'_2} \cdot \overline{S'_3}$ 时有效。
- $Y'_0 \sim Y'_7$: 信号输出端， Y'_0 为最高位。
- 把3比特的 A 翻译为对应的输出信号 $\overline{Y_A}$ 。
- 和74HC148的功能互逆。



译码器：二进制译码器 (138) — 电路图



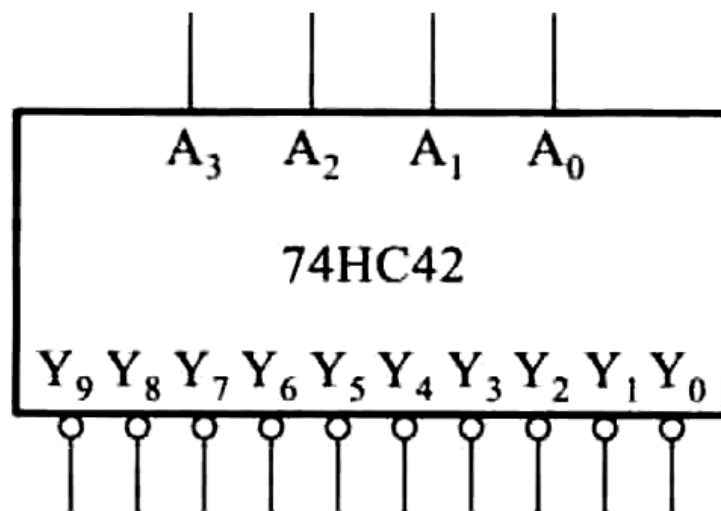
当 $S = S_1 \cdot \overline{S_2} \cdot \overline{S_3} = 1$ 时:

$$\begin{aligned} Y'_0 &= \overline{A_2 \cdot A_1 \cdot A_0} = \overline{m_0} \\ Y'_1 &= \overline{A_2 \cdot A_1 \cdot A_0} = \overline{m_1} \\ Y'_2 &= \overline{A_2 \cdot A_1 \cdot A_0} = \overline{m_2} \\ Y'_3 &= \overline{A_2 \cdot A_1 \cdot A_0} = \overline{m_3} \\ Y'_4 &= \overline{A_2 \cdot A_1 \cdot A_0} = \overline{m_4} \\ Y'_5 &= \overline{A_2 \cdot A_1 \cdot A_0} = \overline{m_5} \\ Y'_6 &= \overline{A_2 \cdot A_1 \cdot A_0} = \overline{m_6} \\ Y'_7 &= \overline{A_2 \cdot A_1 \cdot A_0} = \overline{m_7} \end{aligned}$$

译码器：二—十译码器 (42)

○74HC42: 二—十译码器

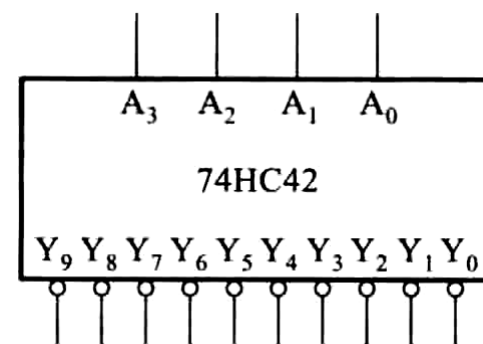
- $A_3 \sim A_0$: 信号输入端, 高有效, A_3 为最高位。
- $Y'_9 \sim Y'_0$: 信号输出端, 低有效, Y'_9 为最高位。
- 把 $A \in [0, 9]$ 翻译为对应的输出信号 $\overline{Y}_A$ 。
- 和74HC147的功能互逆。



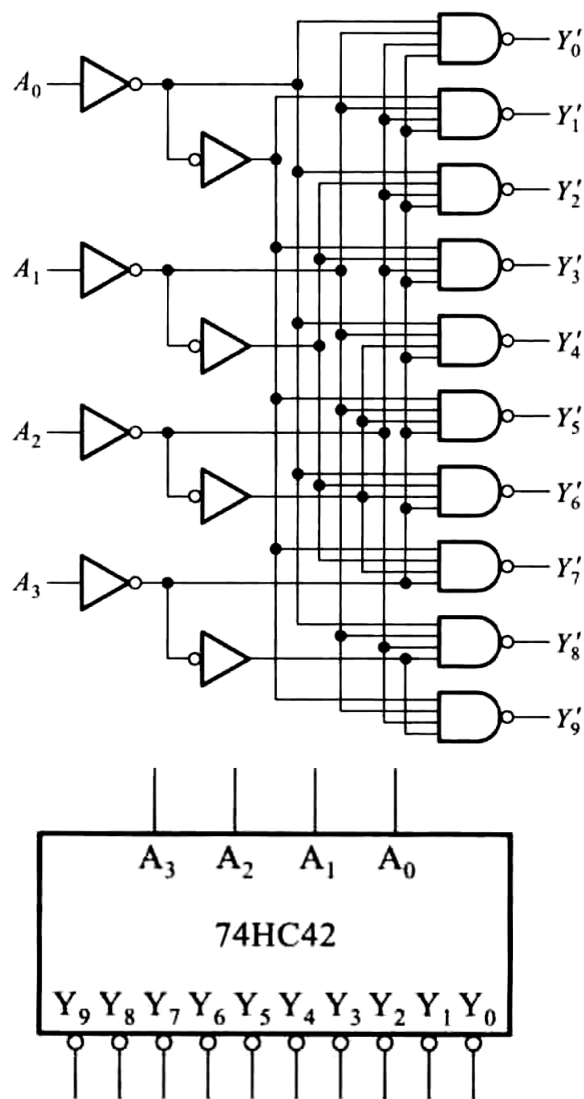
译码器：二—十译码器 (42) —真值表

序号	输入				输出									
	A_3	A_2	A_1	A_0	Y'_0	Y'_1	Y'_2	Y'_3	Y'_4	Y'_5	Y'_6	Y'_7	Y'_8	Y'_9
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0

- $A_3 \sim A_0$: 信号输入端，高有效， A_3 为最高位。
- $Y'_9 \sim Y'_0$: 信号输出端，低有效， Y'_9 为最高位。
- 把 $A \in [0, 9]$ 翻译为对应的输出信号 $\overline{Y}_A$ 。
- 和74HC147的功能互逆。



译码器：二—十译码器 (42) — 电路图



$$\begin{aligned}
 Y'_0 &= \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0} = \overline{m_0} \\
 Y'_1 &= \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot A_0 = \overline{m_1} \\
 Y'_2 &= \overline{A_3} \cdot \overline{A_2} \cdot A_1 \cdot \overline{A_0} = \overline{m_2} \\
 Y'_3 &= \overline{A_3} \cdot \overline{A_2} \cdot A_1 \cdot A_0 = \overline{m_3} \\
 Y'_4 &= \overline{A_3} \cdot A_2 \cdot \overline{A_1} \cdot \overline{A_0} = \overline{m_4} \\
 Y'_5 &= \overline{A_3} \cdot A_2 \cdot \overline{A_1} \cdot A_0 = \overline{m_5} \\
 Y'_6 &= \overline{A_3} \cdot A_2 \cdot A_1 \cdot \overline{A_0} = \overline{m_6} \\
 Y'_7 &= \overline{A_3} \cdot A_2 \cdot A_1 \cdot A_0 = \overline{m_7} \\
 Y'_8 &= A_3 \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0} = \overline{m_8} \\
 Y'_9 &= A_3 \cdot \overline{A_2} \cdot \overline{A_1} \cdot A_0 = \overline{m_9}
 \end{aligned}$$

当输入为 $\{m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}\}$ 时，输出为全高，为无效码。

译码器：BCD—7段LED译码器 (48)

7448: BCD—7段LED译码器

○ $A_3 \sim A_0$: 信号输入端, 高有效, A_3 为最高位。

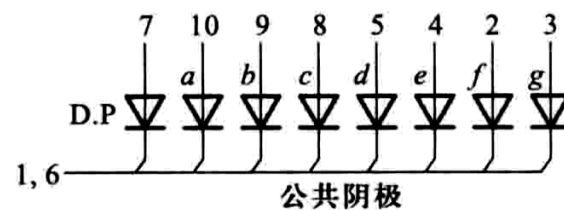
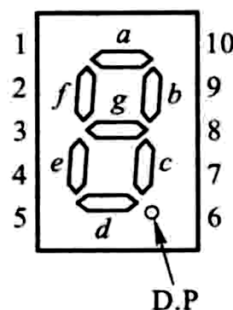
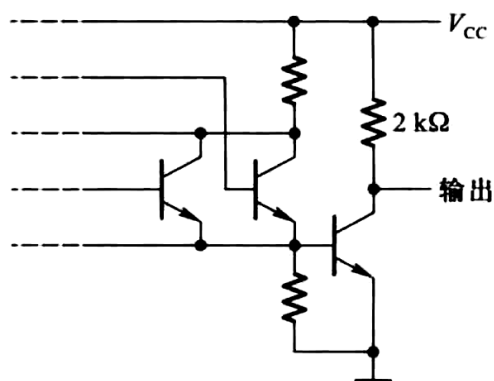
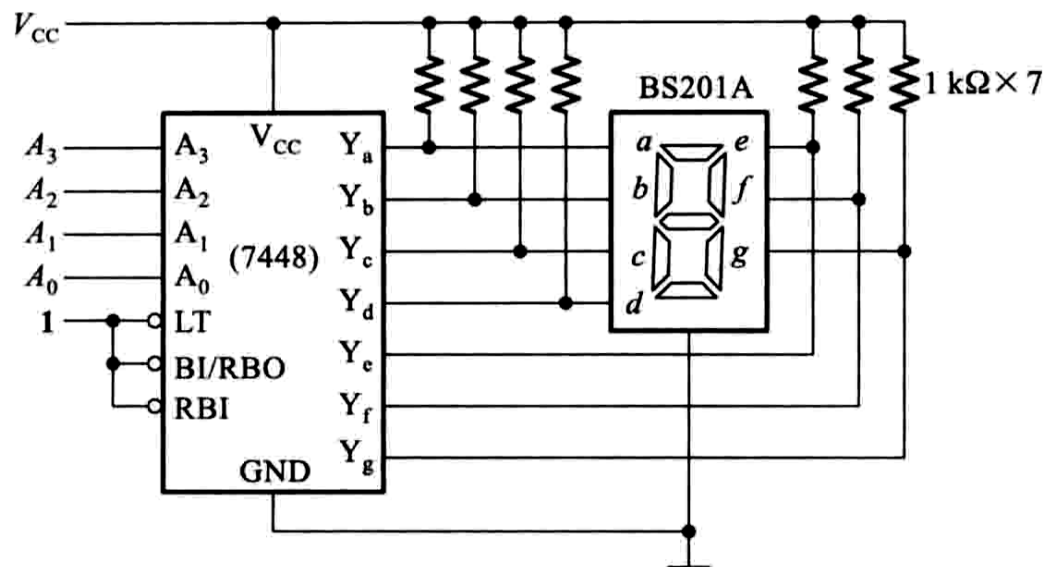
○ $Y_a \sim Y_g$: 信号输出端, 驱动7段LED。

○ LT, RBI, RBO 控制端口。

BS201A: 7段LED显示

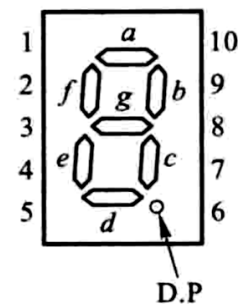
○ 对应48的 $Y_a \sim Y_g$ 和小数点D.P

○ 公共阴极。

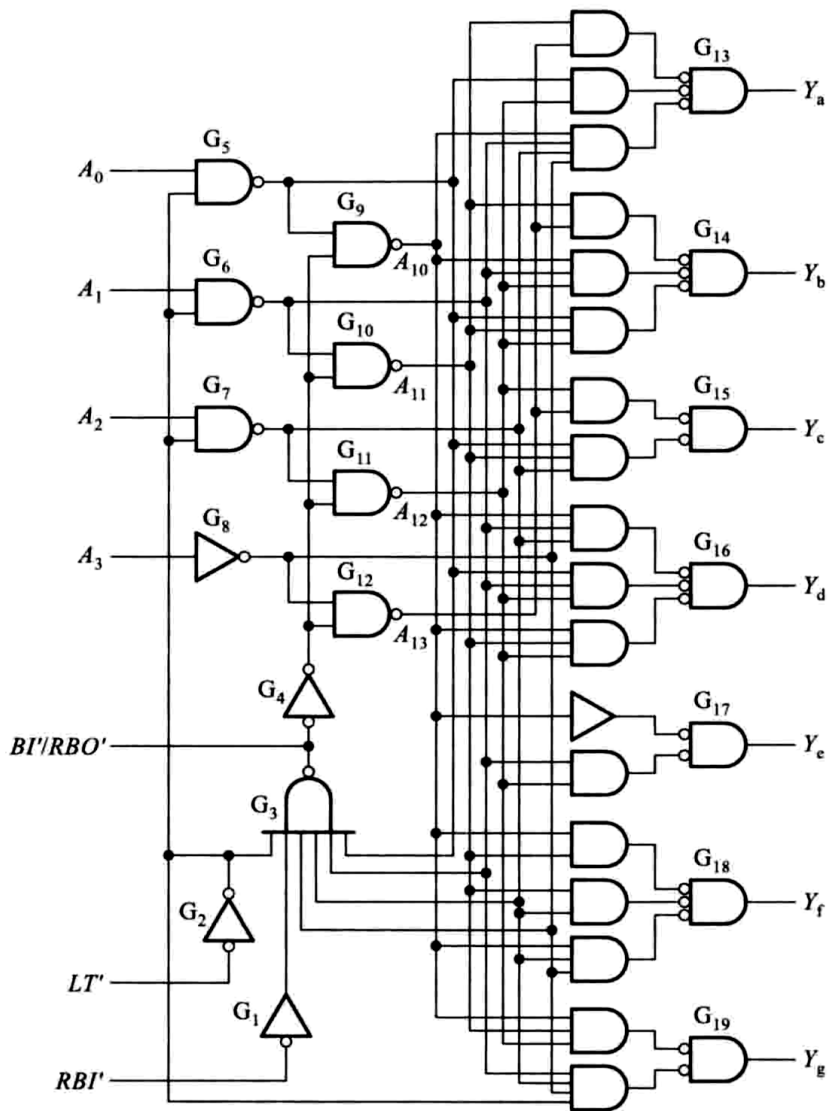


译码器：BCD—7段LED译码器（48）—真值表

输入					输出							字形
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	0
1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	0	1	0	1	1	0	1	1	0	1	2
3	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	0	0	1	1	1	1	1	6
7	0	1	1	1	1	1	1	0	0	0	0	7
8	1	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	1	1	1	0	0	1	1	9

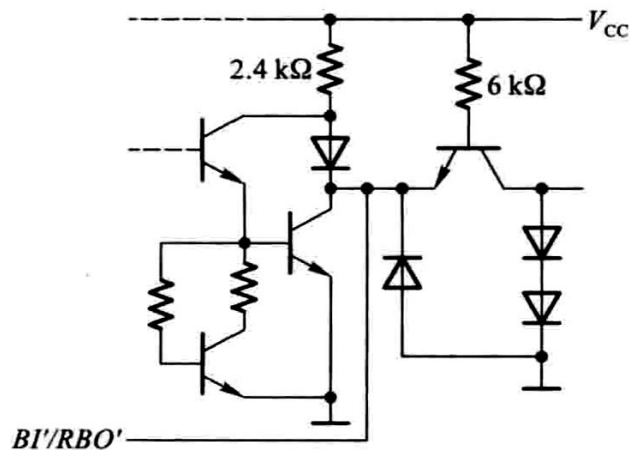


译码器：BCD—7段LED译码器（48）—电路图



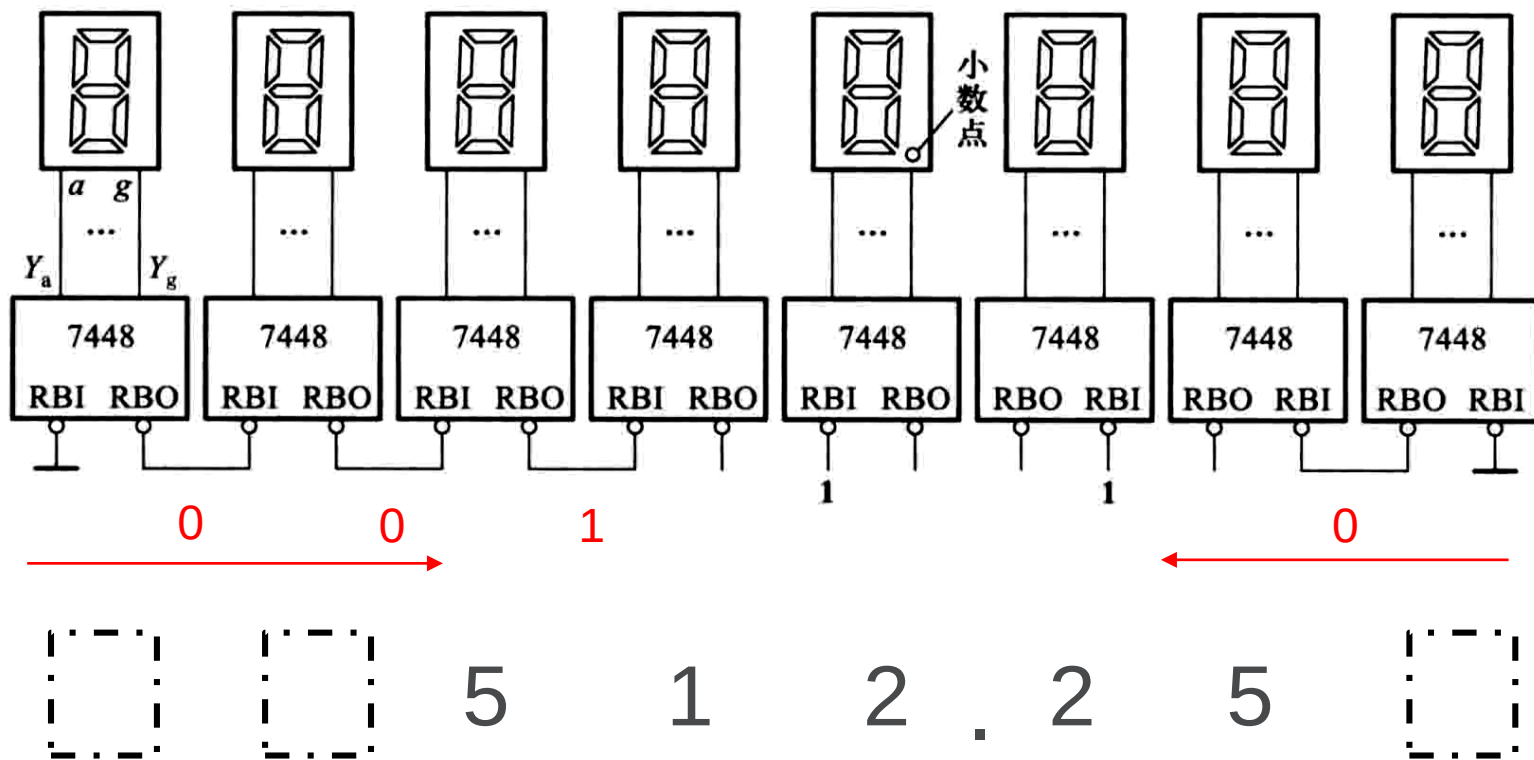
- 当 $LT' \cdot BI' = 1$ 时, $Y_a \sim Y_g$ 的输出按真值表输出。
- 电路使用最大项标准式构建。
- 当 $BI' = 0$ 时, $Y_a \sim Y_g$ 输出全低, LED熄灭。
- 当 $\overline{LT'} = 1$ 时, $Y_a \sim Y_g$ 输出全高, 为LED测试。
- 当 BI' 没有有源驱动时:

$$\begin{aligned} RBO' &= \overline{LT'} + RBI' + A_0 + A_1 + A_2 + A_3 \\ &= \overline{LT' \cdot RBI' \cdot \overline{A_0} \cdot \overline{A_1} \cdot \overline{A_2} \cdot \overline{A_3}} \end{aligned}$$



制

- 当 RBO' 没有有源驱动时: $RBO' = \overline{LT'} + RBI' + A_0 + A_1 + A_2 + A_3$



译码器总结

○二进制码器

- 138：将3比特二进制信号变成8路信号

- 42：将4比特二进制BCD信号变成10路信号

○7段LED译码器（48）

- 将二进制BCD信号翻译为对应LED控制信号

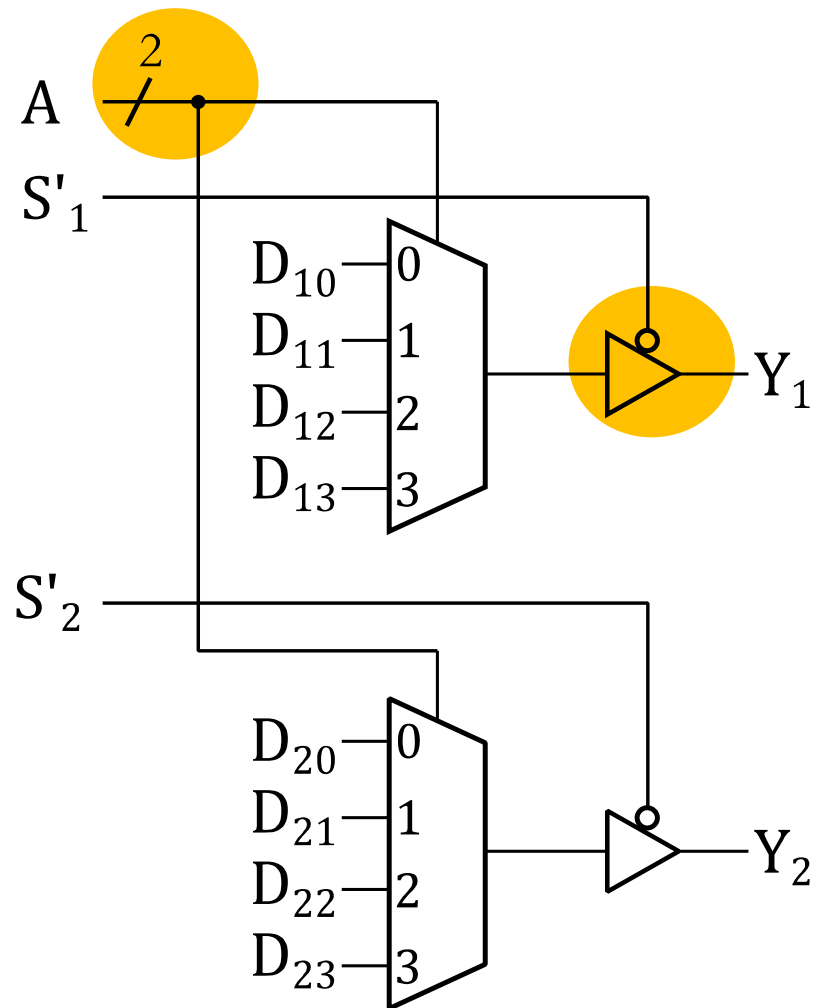
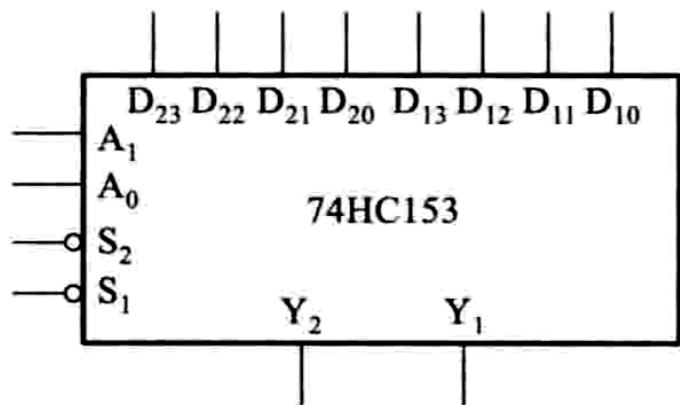
○译码器：

- 将二进制信号翻译为其他信号的模块都可称为译码器

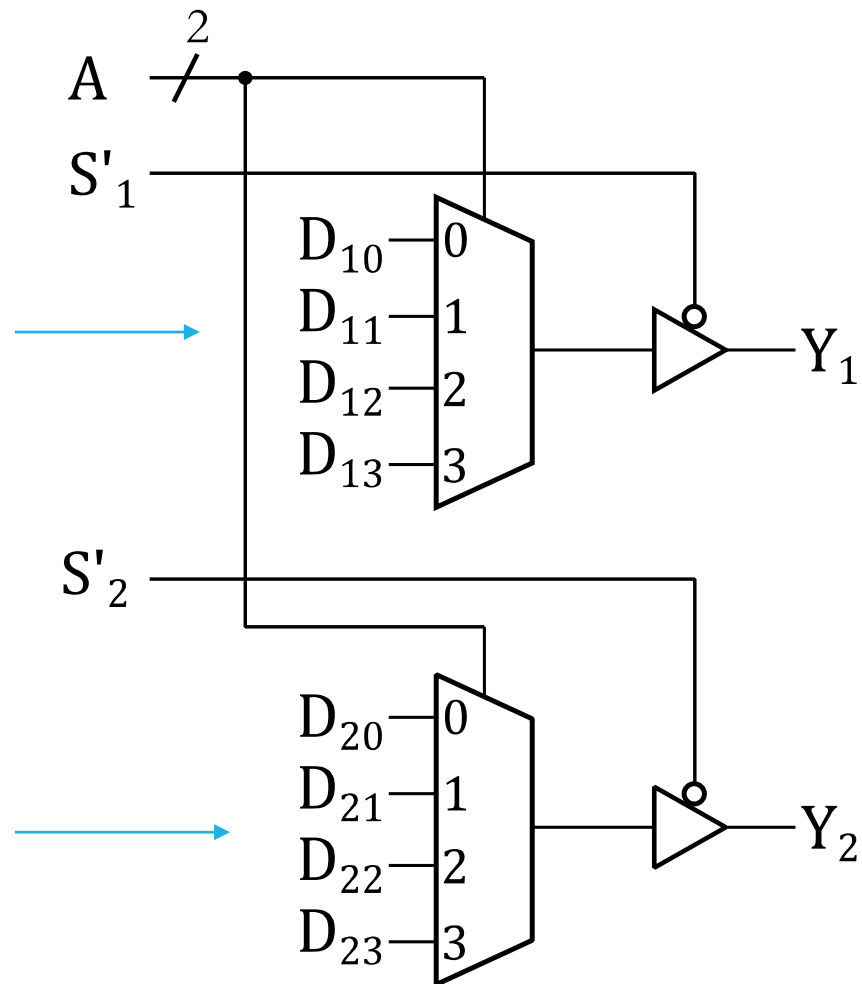
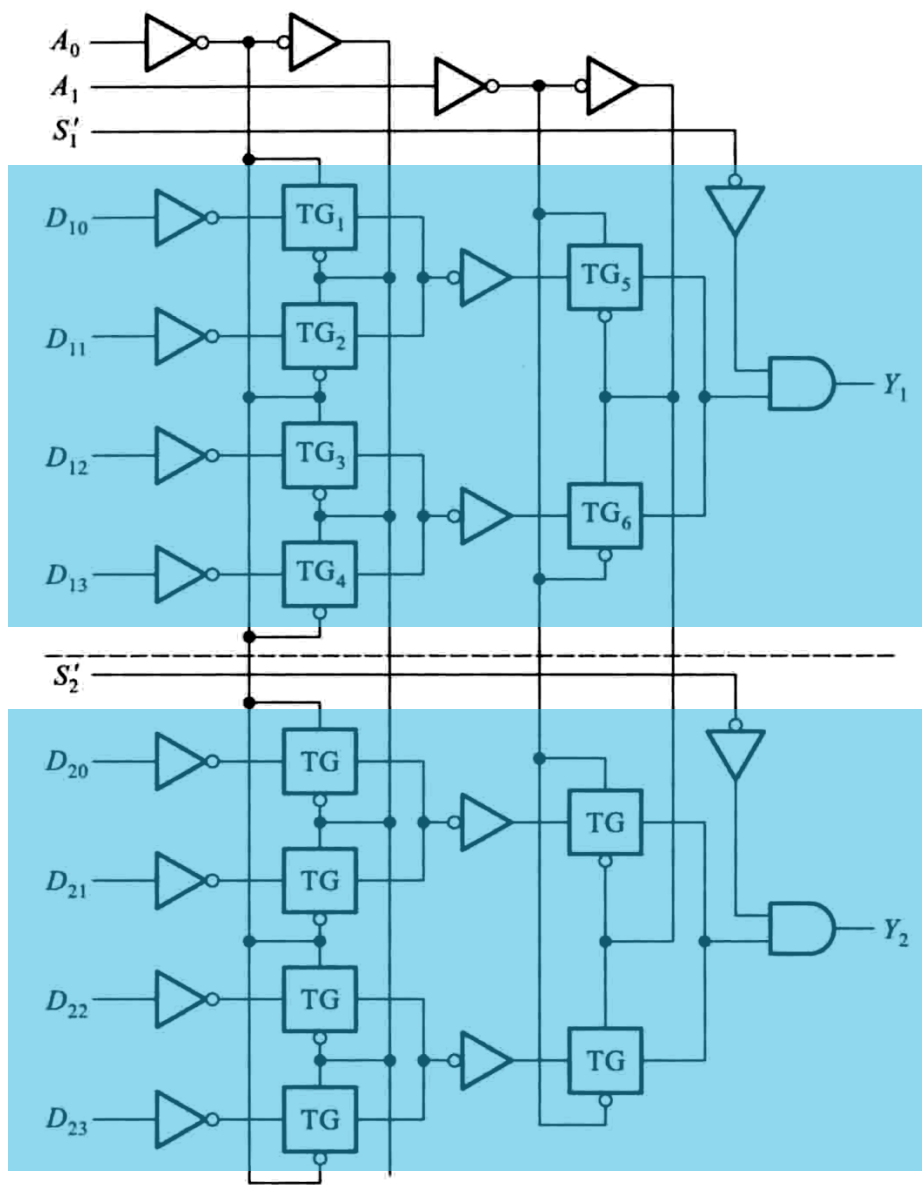
双4选1数据选择器 (153)

74HC153: 双4选1数据选择器

- A: 地址选择, 2bit
- D_1 和 D_2 : 两个数据通道, 4bit
- S'_1 和 S'_2 : 两个数据通道的输出使能, 低有效



双4选1数据选择器 (153) — 电路实现



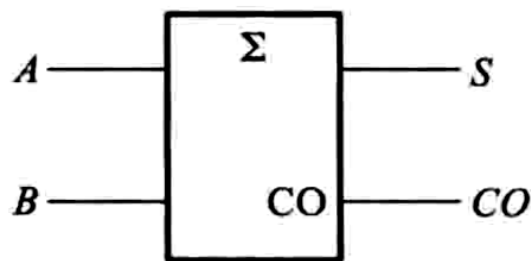
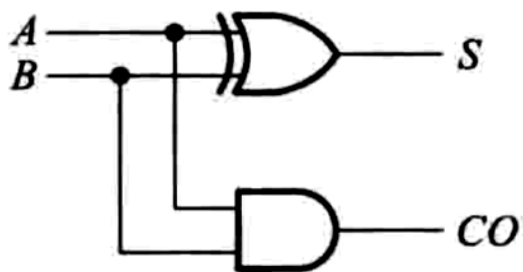
加法器：1位半加器

○1位半加器

- 将两个bit相加
- 计算结果 (S) 和进位 (CO)
- 不考虑低位进位

$$S = A \oplus B$$

$$C_o = A \cdot B$$



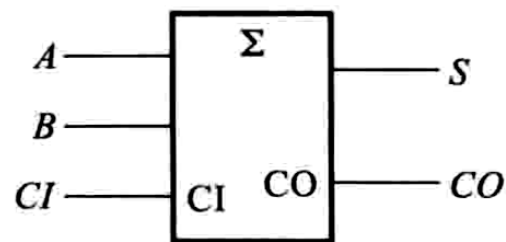
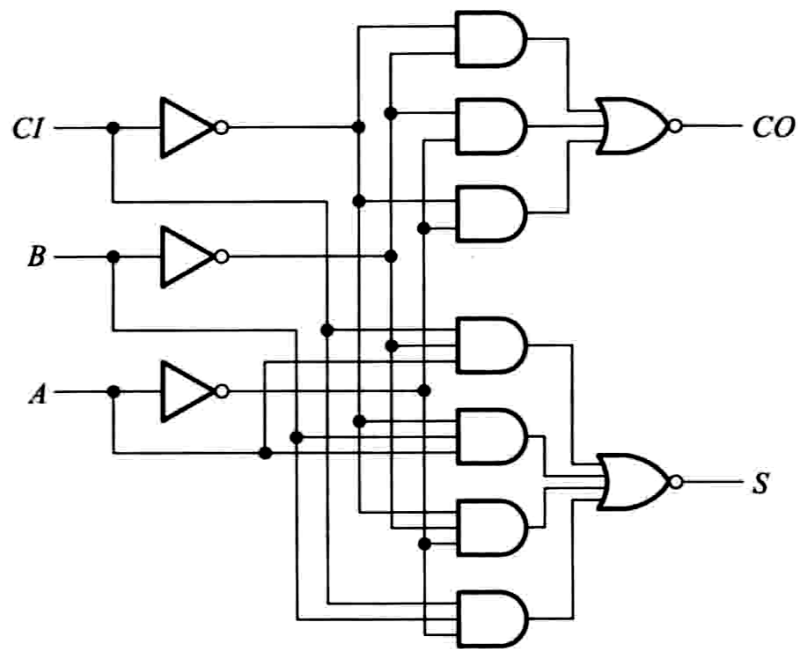
加法器：1位全加器

○1位全加器

○考虑低位进位

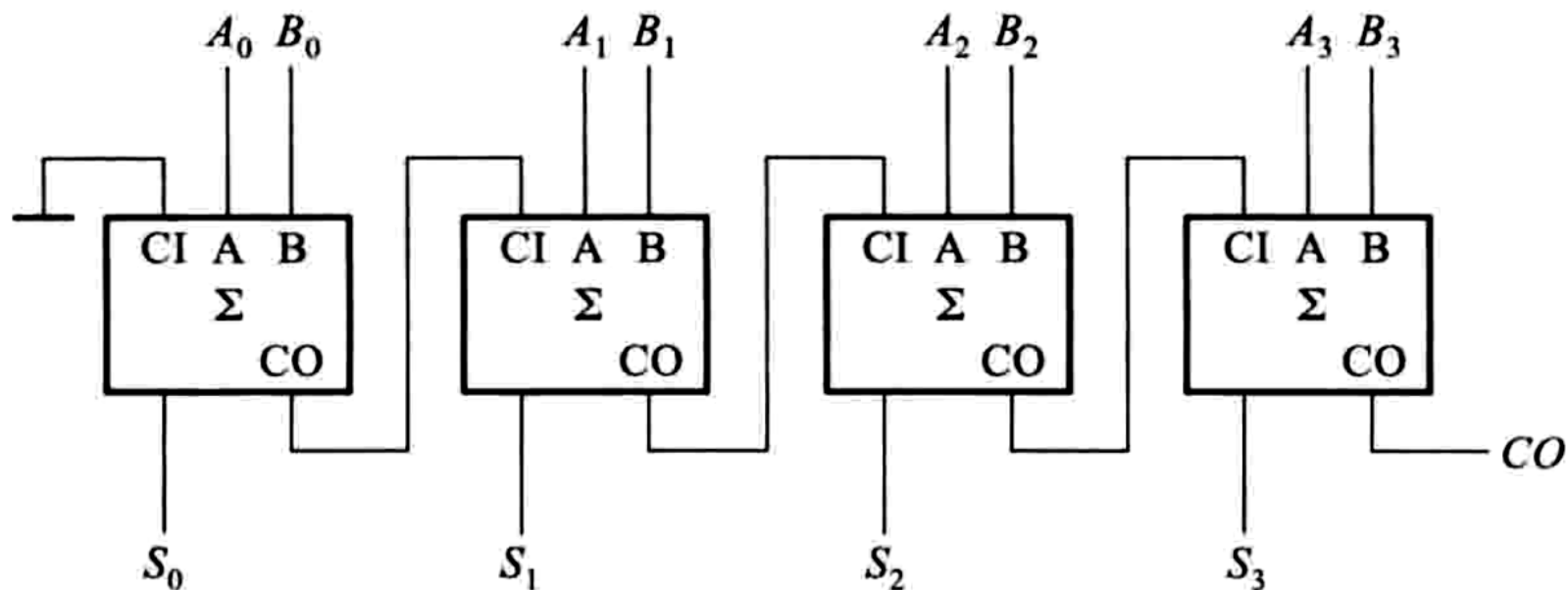
$$S = A \oplus B \oplus C_I$$

$$C_O = A \cdot B + B \cdot C_I + A \cdot C_I$$



加法器：串行多位加法器

○ 串行4位加法器



问题：输出延迟和计算宽度成正比，能否算得更快？

加法器：超前进位（并行）多位加法器

○将串行逻辑展开，得到直接表达式

$$S_i = A_i \oplus B_i \oplus CI_i$$

$$CO_i = A_i B_i + (A_i + B_i) CI_i$$

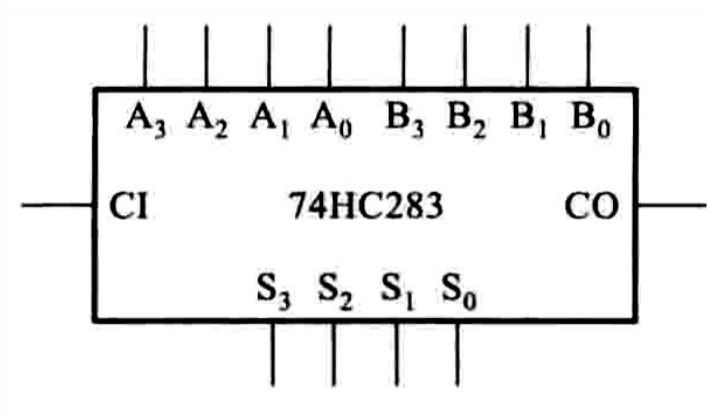
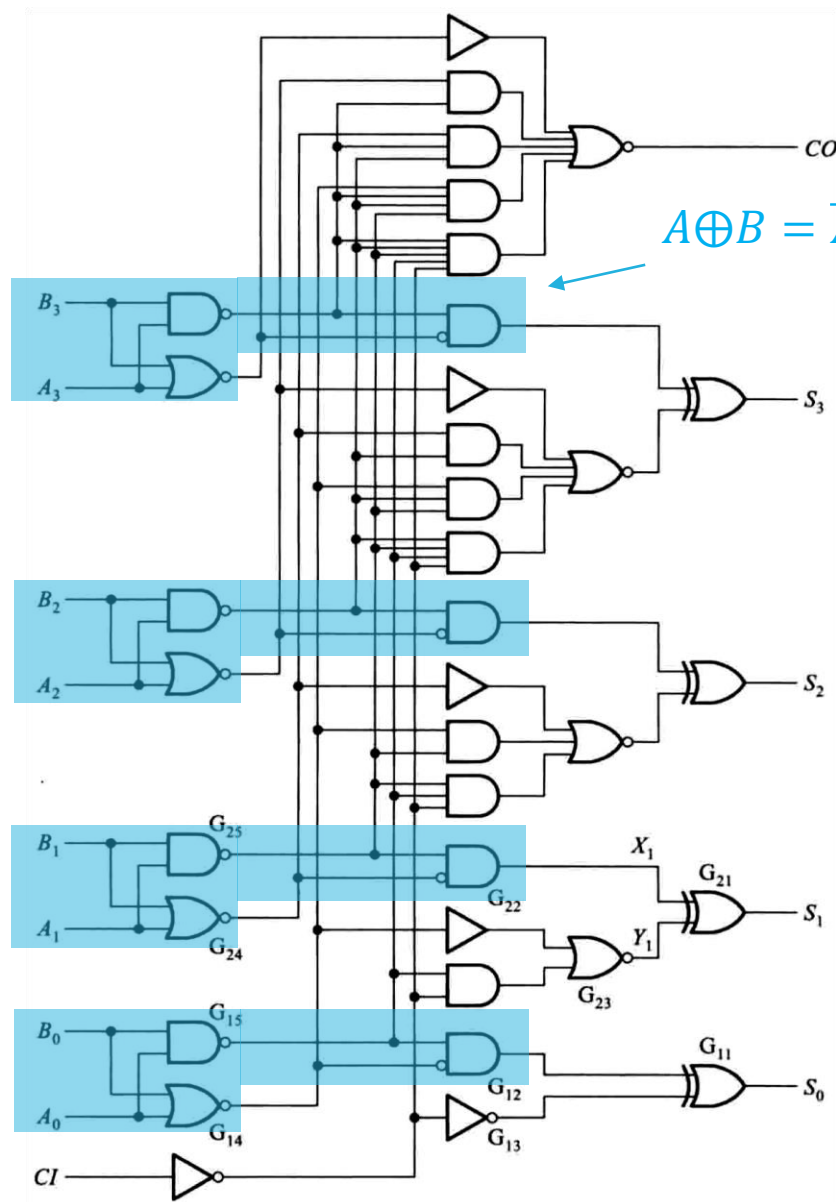
$$CI_i = CO_{i-1}$$

$$CO_i = A_i B_i + (A_i + B_i)(A_{i-1} B_{i-1} + (A_{i-1} + B_{i-1}) CI_{i-1})$$

$$CO_i = A_i B_i + (A_i + B_i) A_{i-1} B_{i-1} + (A_i + B_i)(A_{i-1} + B_{i-1}) CI_{i-1}$$

$$CO_i = A_i B_i + (A_i + B_i) A_{i-1} B_{i-1} + (A_i + B_i)(A_{i-1} + B_{i-1}) A_{i-2} B_{i-2} \\ + (A_i + B_i)(A_{i-1} + B_{i-1})(A_{i-2} + B_{i-2}) CI_{i-2}$$

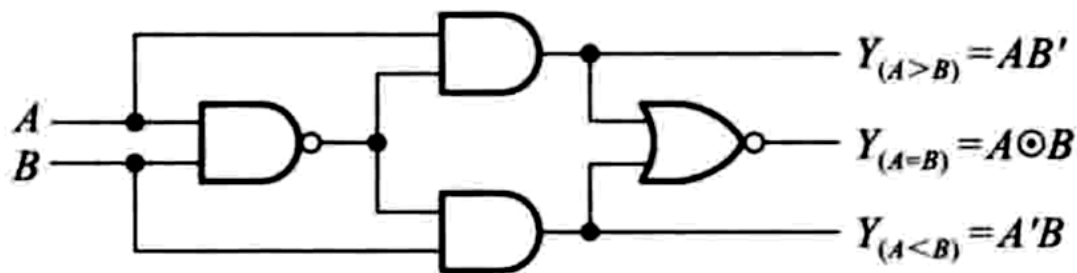
加法器：超前进位4位加法器74HC283



$$\{CO, S\} = \{0, A\} + \{0, B\} + \{0, 0, 0, 0, CI\}$$

比较器

○1位比较器



○多位比较器

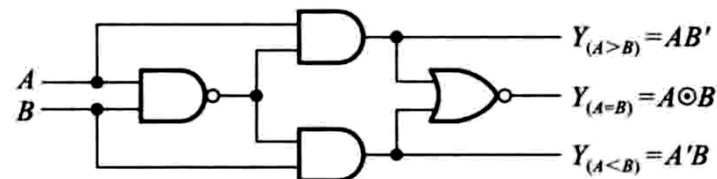
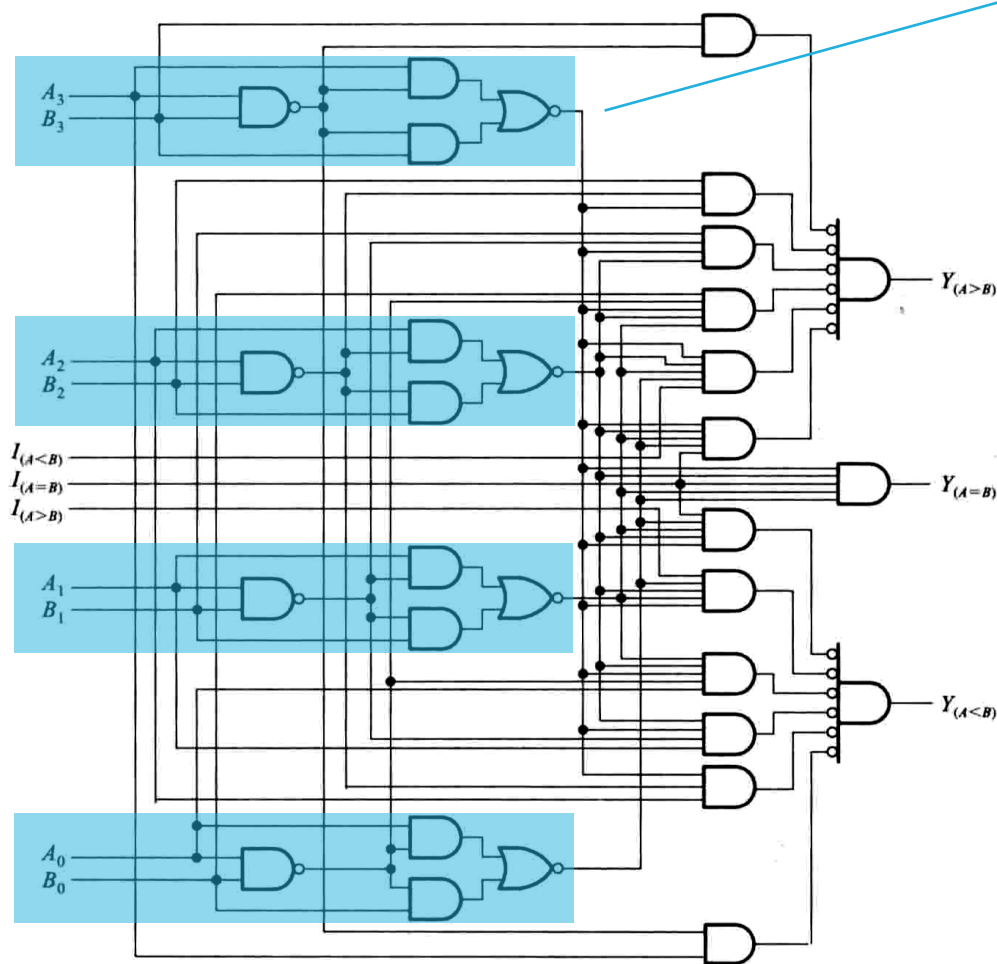
从高位开始比较

$$Y_{A>B} = A_n \overline{B_n} + (A_n \odot B_n)(\overline{A_{n-1} \overline{B_{n-1}}} + (A_{n-1} \odot B_{n-1}) I_{A>B})$$

$$Y_{A=B} = (A_n \odot B_n)(A_{n-1} \odot B_{n-1}) I_{A=B}$$

$$Y_{A<B} = \overline{A_n} B_n + (A_n \odot B_n)(\overline{A_{n-1} B_{n-1}} + (A_{n-1} \odot B_{n-1}) I_{A<B})$$

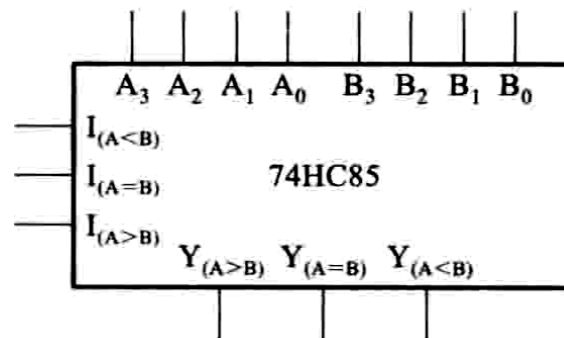
4位数值比较器 (74HC85)



$$Y_{A>B} = A_n \overline{B_n} + (A_n \odot B_n)(A_{n-1} \overline{B_{n-1}} + (A_{n-1} \odot B_{n-1})I_{A>B})$$

$$Y_{A=B} = (A_n \odot B_n)(A_{n-1} \odot B_{n-1})I_{A=B}$$

$$Y_{A<B} = \overline{A_n} B_n + (A_n \odot B_n)(\overline{A_{n-1}} B_{n-1} + (A_{n-1} \odot B_{n-1})I_{A<B})$$



运算单元总结

○选择器

- 利用选择端在多路数字信号中选择一路
- 74HC53: 双4选1选择器

○加法器

- 半加器: 不考虑输入进位
- 全加器: 考虑输入进位
 - 串行和超前进位 (并行)
- 74HC283: 4比特超前进位全加器

○比较器

- 比较两个逻辑信号
- 74HC85: 4位数值比较器

层次化和模块化：模块扩展

○模块扩展

- 利用已有的模块，实现更复杂的电路逻辑
- 自底向上的设计思路

○简单例子：

假设我们只有双输入与非门，如何利用双输入与非门实现：

$$Y = A + BC + \bar{B}D$$

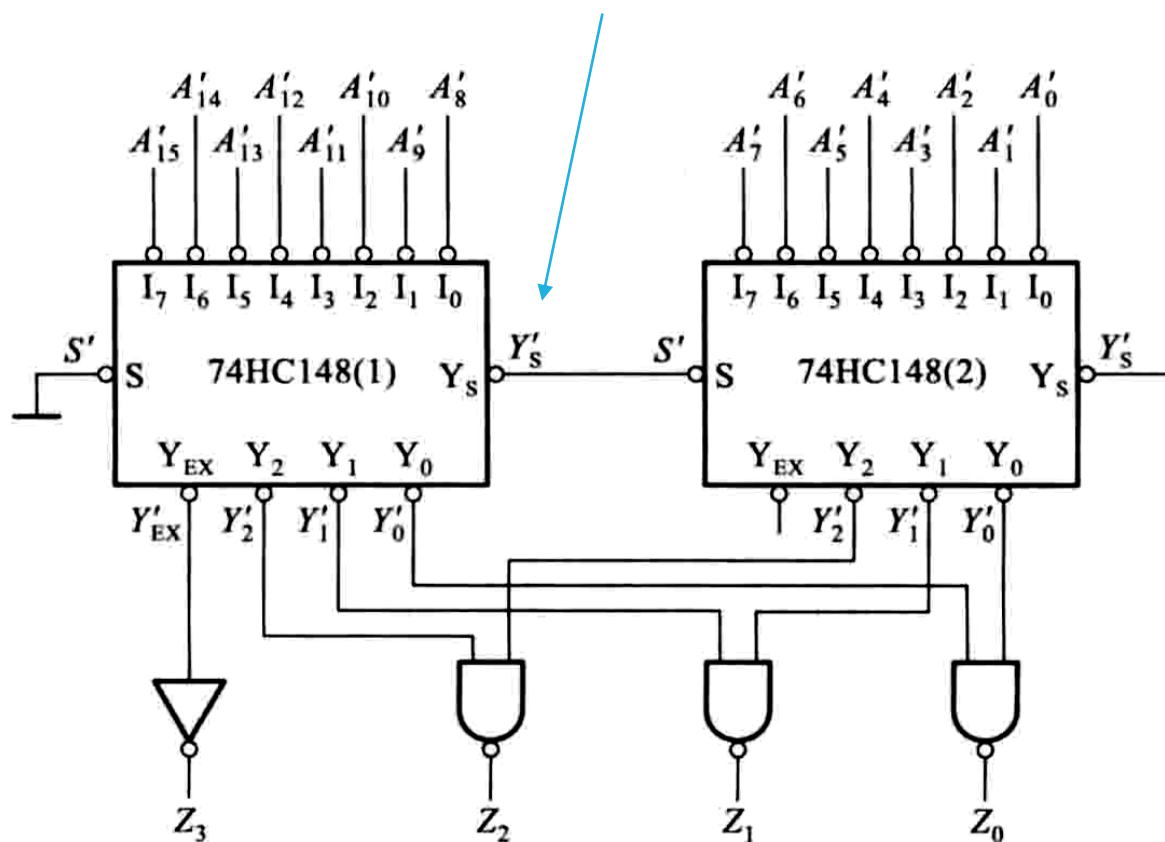
利用第二章的知识，我们可以将上面的逻辑表达式改写为：

$$Y = \overline{\overline{A \cdot 1 \cdot \overline{BC}} \cdot 1 \cdot \overline{B \cdot 1 \cdot D}}$$

这个逻辑表达式就只需要二输入与非门就可实现。

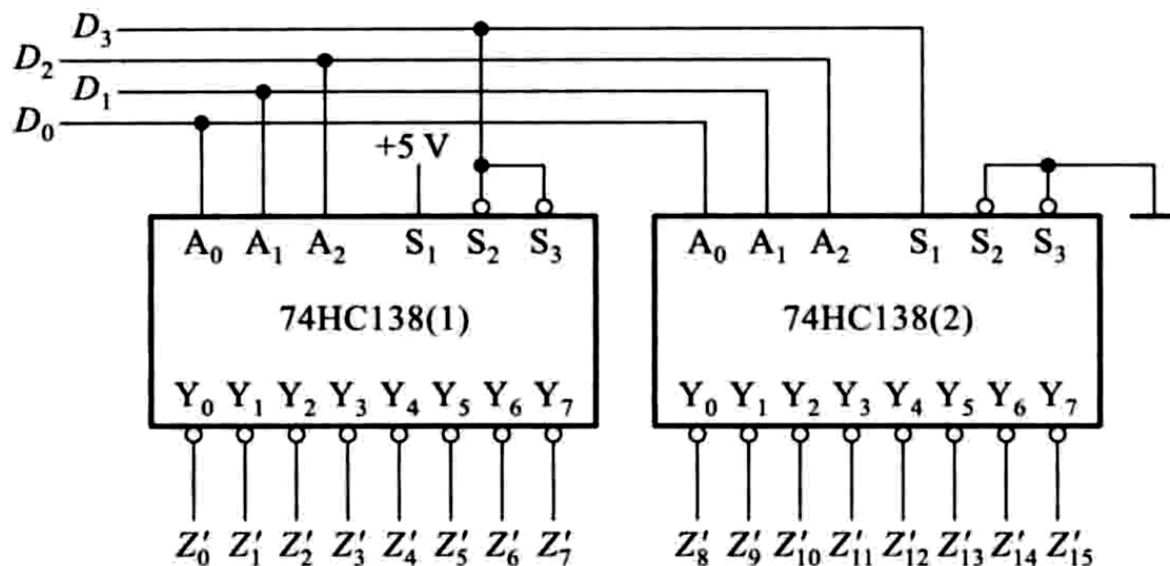
模块扩展例一：扩展编码器

- 利用两个8线—3线编码器实现一个16线—4线编码器
当第一片没有输入时，利用 Y'_S 选通第二片。



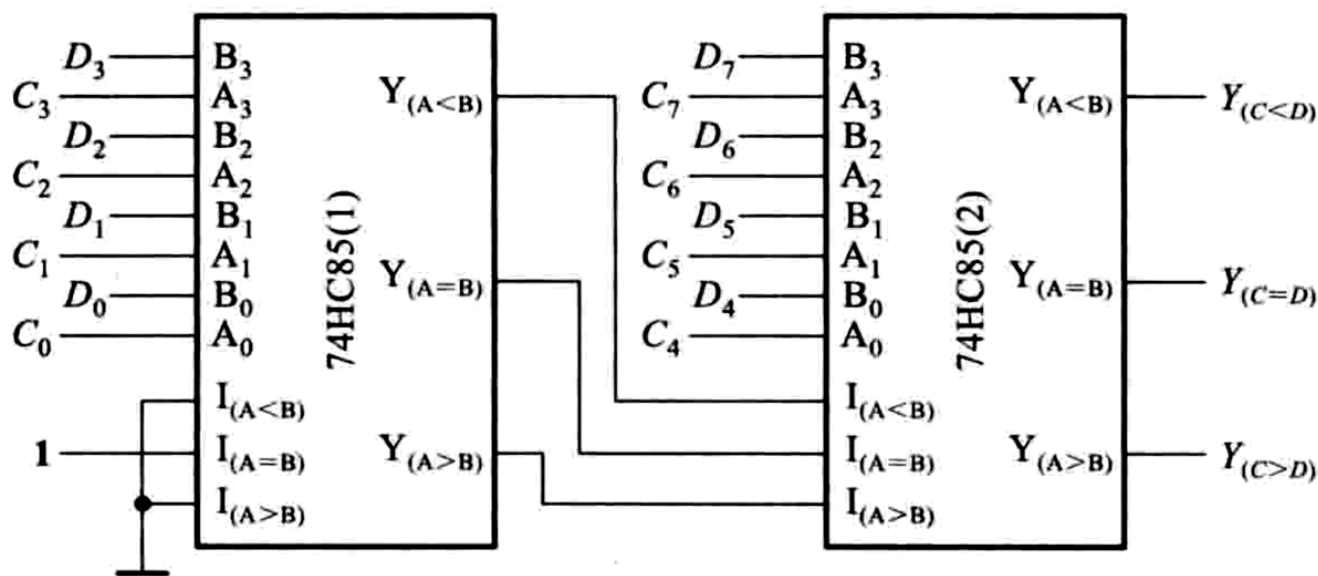
模块扩展例二：扩展译码器

- 利用两个3线—8线译码器实现一个4线—16线译码器
当高位 D_3 为低时选中片一，高时选中片二。



模块扩展例三：扩展比较器

- 利用两个4位数值比较器实现一个8位数值比较器
低4位的比较结果直接输给高位比较器。



模块扩展之功能变换，实例一：选择器

○利用选择器实现交通灯监视

假设一组交通灯，由红黄绿三盏灯组成。正常工作情况下，任何时候有且仅有一盏灯点亮。如果出现没有灯点亮或者多于一盏灯点亮则为故障。当交通灯故障时，一个故障信号开启，提示维修人员检查。请设计该故障信号。

$$Z = RA + RG + AG + \overline{R}\overline{A}\overline{G}$$

使用ite范式 (顺序: A,G)

$$Z = A(R + G) + \overline{A}(RG + \overline{R}\overline{G})$$

$$Z = A(G(1) + \overline{G}(R)) + \overline{A}(G(R) + \overline{G}(\overline{R}))$$

$$Z = AG(1) + \overline{A}G(R) + A\overline{G}(\overline{R}) + \overline{A}\overline{G}(\overline{R})$$

令 $A = \{A, G\}$:

$$Z = A_3(1) + A_2(R) + A_1(\overline{R}) + A_0(\overline{R})$$

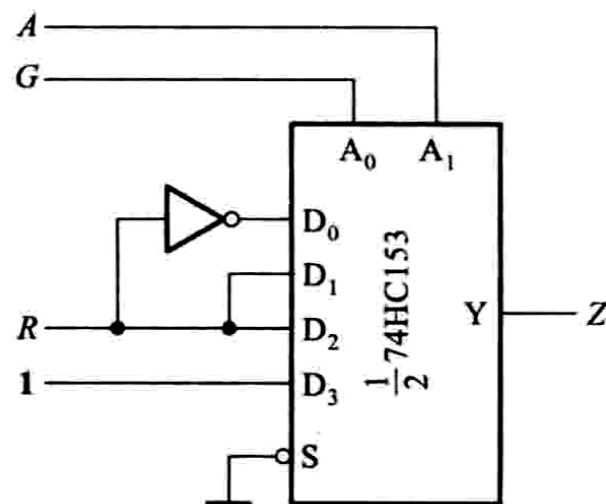
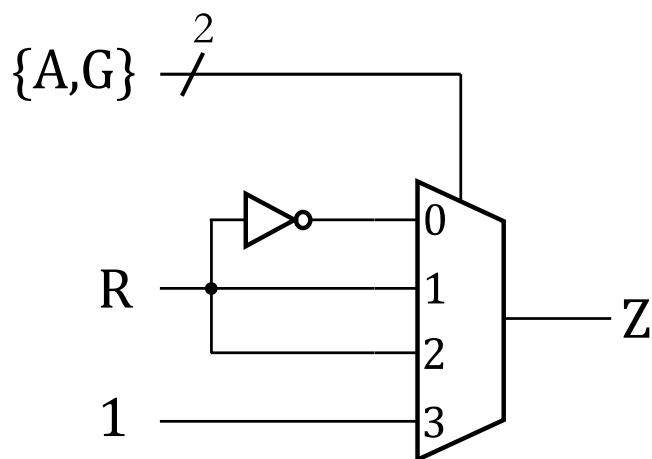
模块扩展之功能变换，实例一：选择器

$$Z = RA + RG + AG + \bar{R}\bar{A}\bar{G}$$

$$Z = AG(1) + \bar{A}G(R) + A\bar{G}(R) + \bar{A}\bar{G}(\bar{R})$$

令 $A = \{A, G\}$:

$$Z = A_3(1) + A_2(R) + A_1(R) + A_0(\bar{R})$$



通过将逻辑表达式中的一部分逻辑变量当作选择器的地址信号，选择器可以被用来实现任意表达式。

自学教材中182页，例4.5.5，使用8选1选择器实现任意3逻辑变量表达式。

总结：考试范围

○组合逻辑电路分析

- 从问题描述归纳总结出逻辑表达式
- 分析电路实现并归纳出逻辑表达式
- 对逻辑表达式经行各种化简和推导

○常用的组合逻辑模块

- 编码器和译码器
- 选择器、加法器、比较器
- 理解常用的组合逻辑模块
 - 不需要记具体实现，会分析即可
 - 本课程不要求记住具体型号（其他院系要求可能不一样）

任何问题?

第四章习题：

2, 3, 4, 9, 10, 12, 19, 25, 29。

○用四路选一选择器设计四变量的多数表决器：

当输入变量 A, B, C, D 有3个或3个以上为1时输出为1，其他情况输出为0。