第四章　组合逻辑电路设计（二）

秦磊华　计算机学院

4.5 logisim的基本使用

例1 设计一个比较两个三位二进制数是否相等的数值比较器。

（两个3位二进制数分别为A = $a_3a_2a_1$，B = $b_3b_2b_1$）

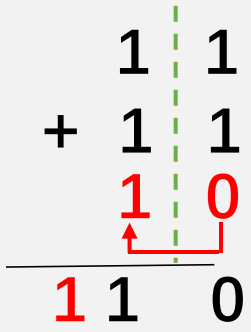
$$F_{=} = (\overline{a_3} \cdot \overline{b_3} + a_3b_3) \cdot (\overline{a_2} \cdot \overline{b_2} + a_2b_2) \cdot (\overline{a_1} \cdot \overline{b_1} + a_1b_1)$$

$$F_{A>B} = A_3\overline{B_3} + (A_3B_3 + \overline{A_3}\,\overline{B_3})(A_2\overline{B_2}) + (A_3B_3 + \overline{A_3}\,\overline{B_3})(A_2B_2 + \overline{A_2}\,\overline{B_2})(A_1\overline{B_1})$$

$$F_{A<B} = \overline{A_3}B_3 + (A_3B_3 + \overline{A_3}\,\overline{B_3})(\overline{A_2}B_2) + (A_3B_3 + \overline{A_3}\,\overline{B_3})(A_2B_2 + \overline{A_2}\,\overline{B_2})(\overline{A_1}B_1)$$

4.6 基本组合逻辑功能部件设计

1. 一位全加器FA(Full-Adder)设计



加数X_i	加数Y_i	低位进位C_i	和S_i	进位C_{i+1}
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

型号	功能	PDT_{MAX}
74LS86	4-2异或	30ns
74LS32	4-2或门	22ns
74LS00	4-2与非门	15ns
74LS04	6-非门	15ns
74LS08	4-2与门	20ns

$$S_i = X_i \oplus Y_i \oplus C_i$$

$$C_{i+1} = X_i Y_i + (X_i + Y_i)C_i$$

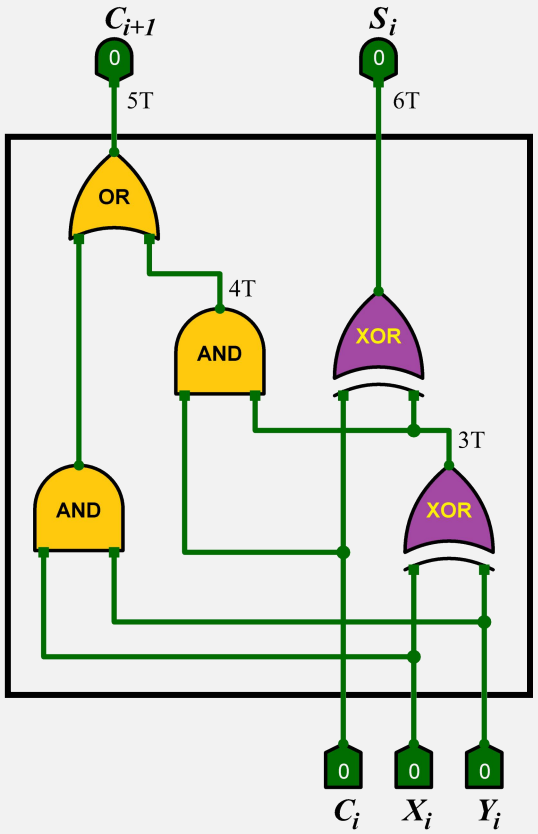
$$C_{i+1} = X_i Y_i + (X_i \oplus Y_i)C_i$$

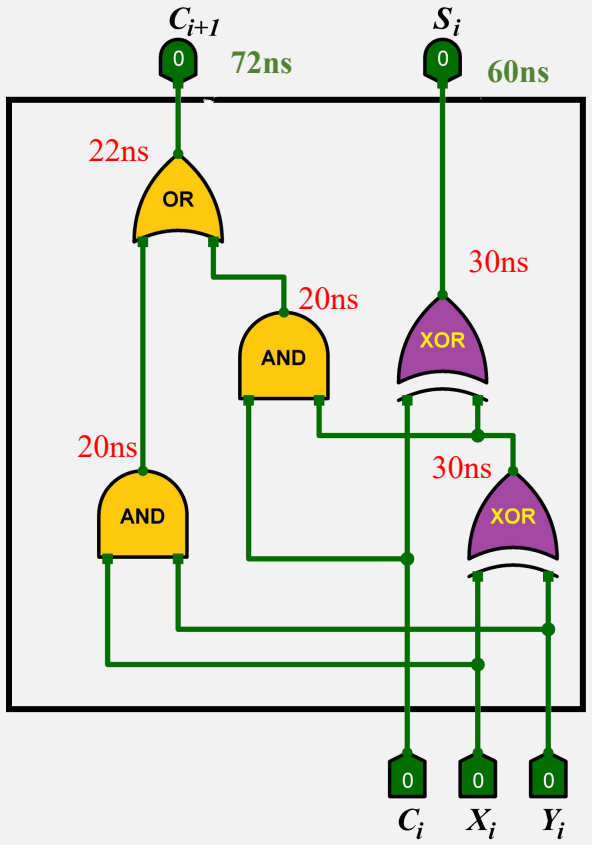
4.6 基本组合逻辑功能部件设计

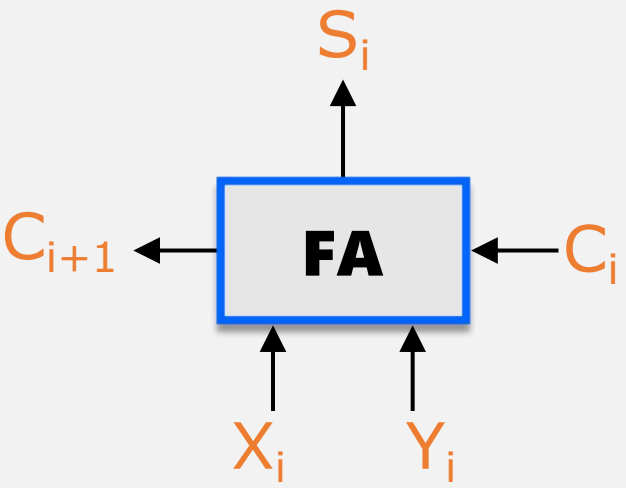
$$S_i = X_i \oplus Y_i \oplus C_i$$

$$C_{i+1} = X_i Y_i + (X_i \oplus Y_i) C_i$$

型号	功能	PDT_{MAX}
74LS86	4-2异或	30ns
74LS32	4-2或门	22ns
74LS00	4-2与非门	15ns
74LS04	6-非门	15ns
74LS08	4-2与门	20ns







4.6 基本组合逻辑功能部件设计

$$S_i = X_i \oplus Y_i \oplus C_i$$

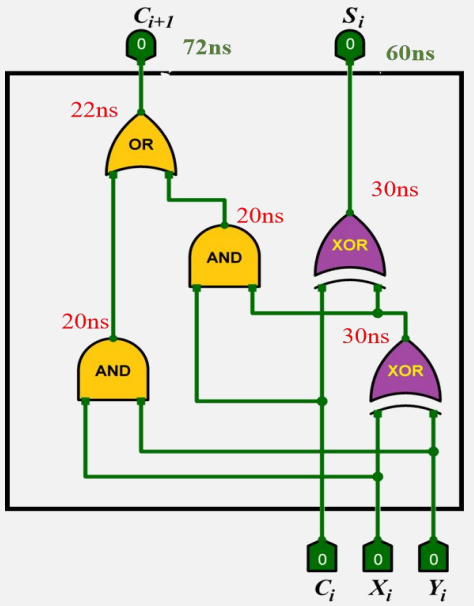
$$C_{i+1} = X_i Y_i + (X_i + Y_i) C_i$$

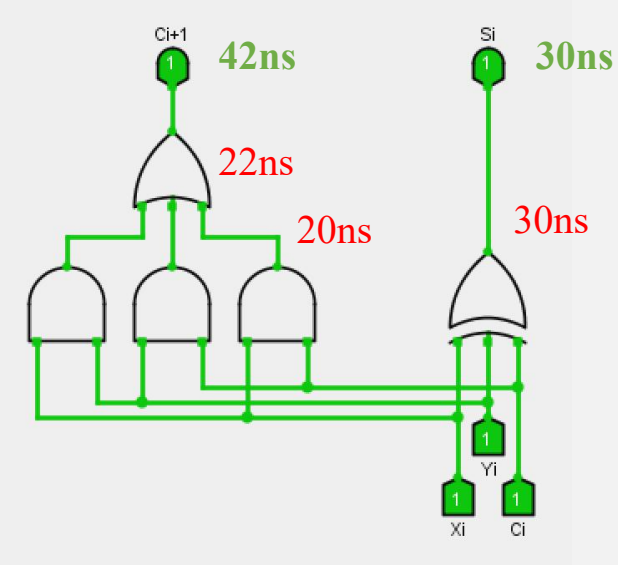
$$= X_i Y_i + X_i C_i + Y_i C_i$$

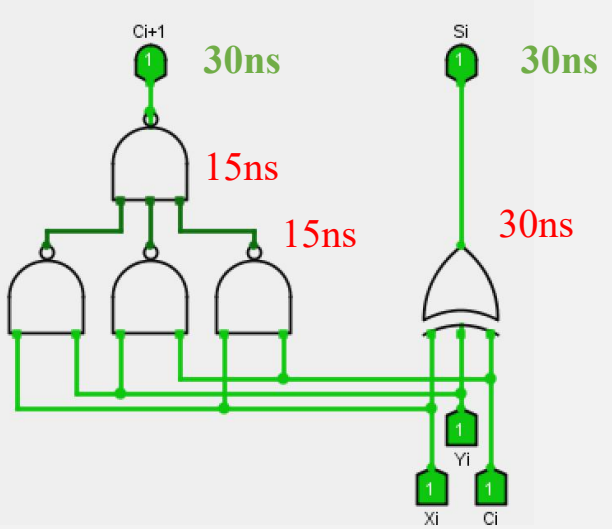
$$= \overline{\overline{X_i Y_i + X_i C_i + Y_i C_i}}$$

$$= \overline{\overline{X_i Y_i}} \cdot \overline{\overline{X_i C_i}} \cdot \overline{\overline{Y_i C_i}}$$

型号	功能	PDT_{MAX}
74LS86	4-2异或	30ns
74LS32	4-2或门	22ns
74LS00	4-2与非门	15ns
74LS04	6-非门	15ns
74LS08	4-2与门	20ns
74136	4输入异或门	30ns

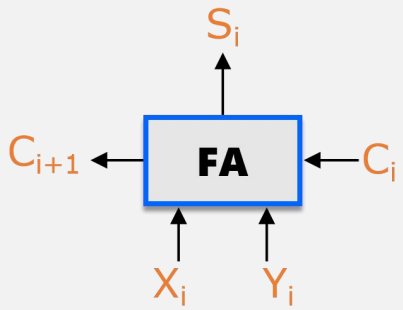




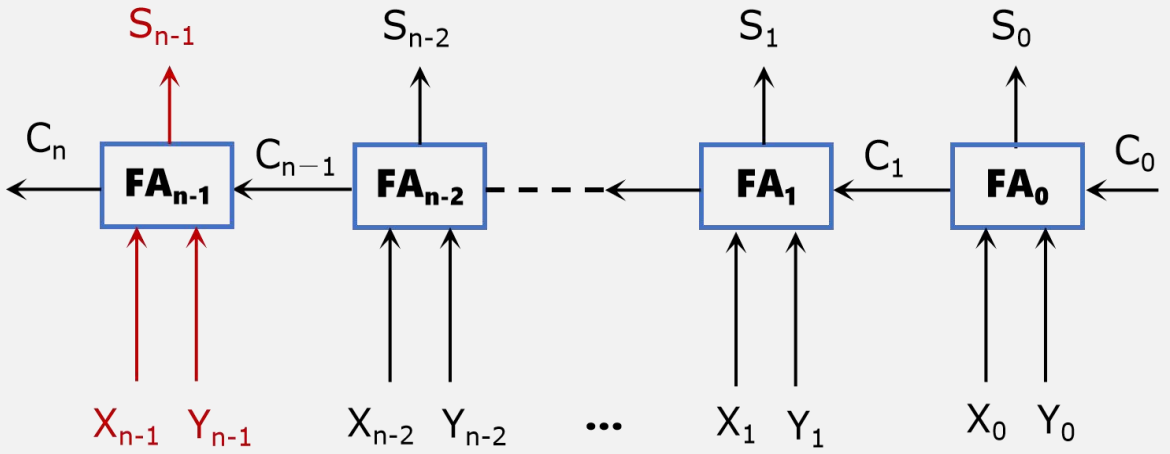


4.6 基本组合逻辑功能部件设计

能用FA构建任意位数的多位加法器吗？




```
    1 0 1 0 0 1
+   1 1 1 0 0 0
  ---------------
  0 1 1 0 0 0
  ---------------
1 1 0 0 0 0 1
```



4.6 基本组合逻辑功能部件设计

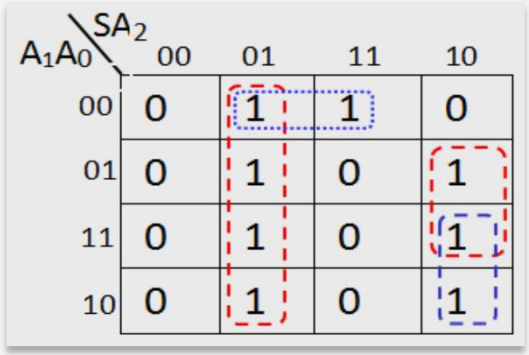
2.设计一个3位求补电路

$x = +1011001 \dashrightarrow [x]_{补} = 01011001$

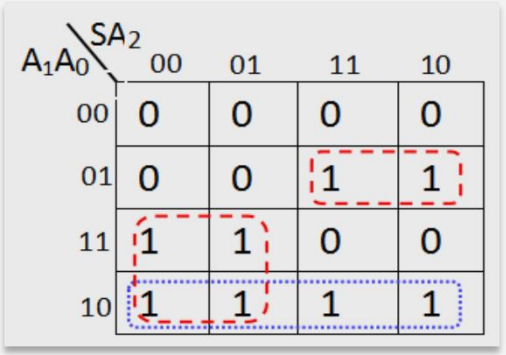
$x = -1011001 \dashrightarrow [x]_{补} = 10100111$

$x = -1011001$

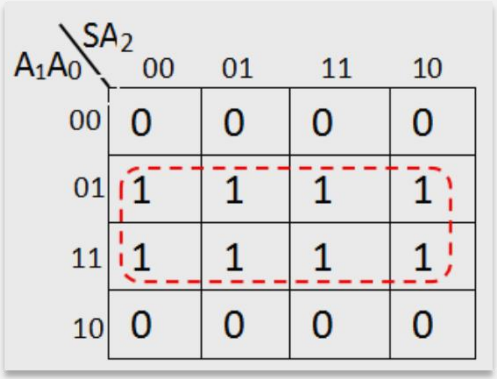
输入端				输出端		
S	A_2	A_1	A_0	F_2	F_1	F_0
1	0	0	0	0	0	0
1	0	0	1	1	1	1
1	0	1	0	1	1	0
1	0	1	1	1	0	1
1	1	0	0	1	0	0
1	1	0	1	0	1	1
1	1	1	0	0	1	0
1	1	1	1	0	0	1
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	1	0	0
0	1	0	1	1	0	1
0	1	1	0	1	1	0
0	1	1	1	1	1	1



F_2



F_1



F_0

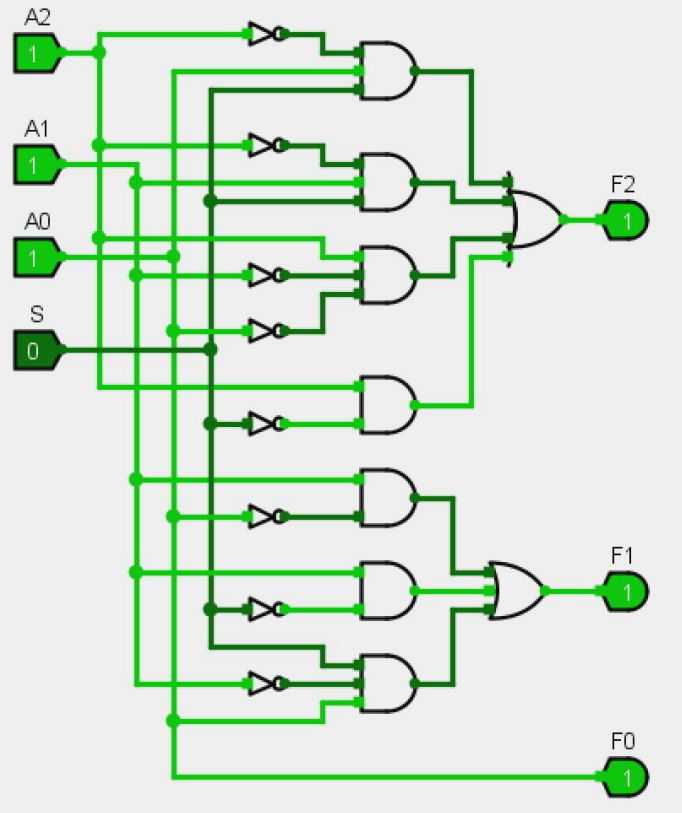
$$F_2 = \overline{S}A_2 + A_2\overline{A_1}\,\overline{A_0} + S\overline{A_2}A_0 + S\overline{A_2}A_1$$

$$F_1 = \overline{A_0}A_1 + A_1\overline{S} + S\overline{A_1}A_0$$

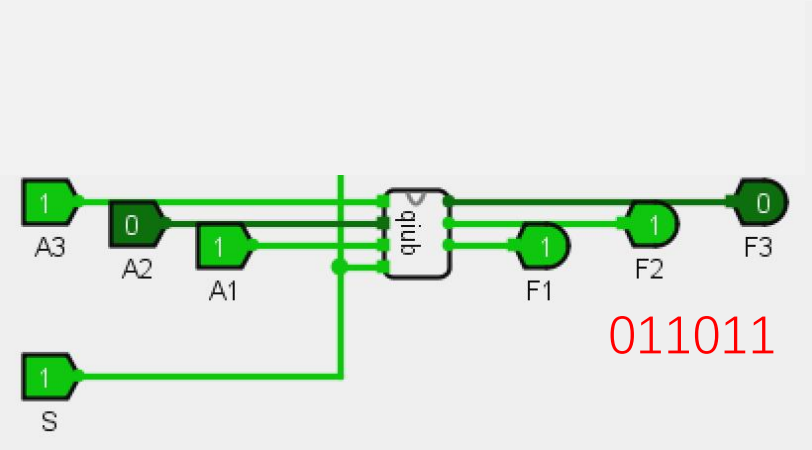
$$F_0 = A_0$$

4.6 基本组合逻辑功能部件设计

2.设计一个3位求补电路



1 101101

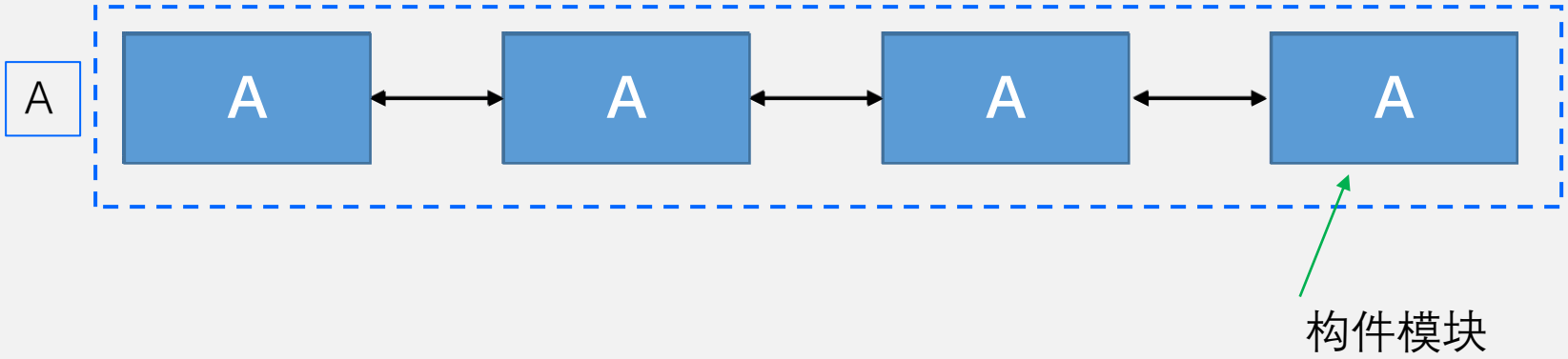


011011

3. 硬件迭代设计

1)基本概念

若一个规模可扩展的逻辑电路可通过重复使用功能相同的子模块设计而成，称这样的硬件设计为硬件的迭代设计。



3. 硬件迭代设计

2)具有迭代性的硬件构件模块设计方法

(1)判定逻辑功能的可分解性

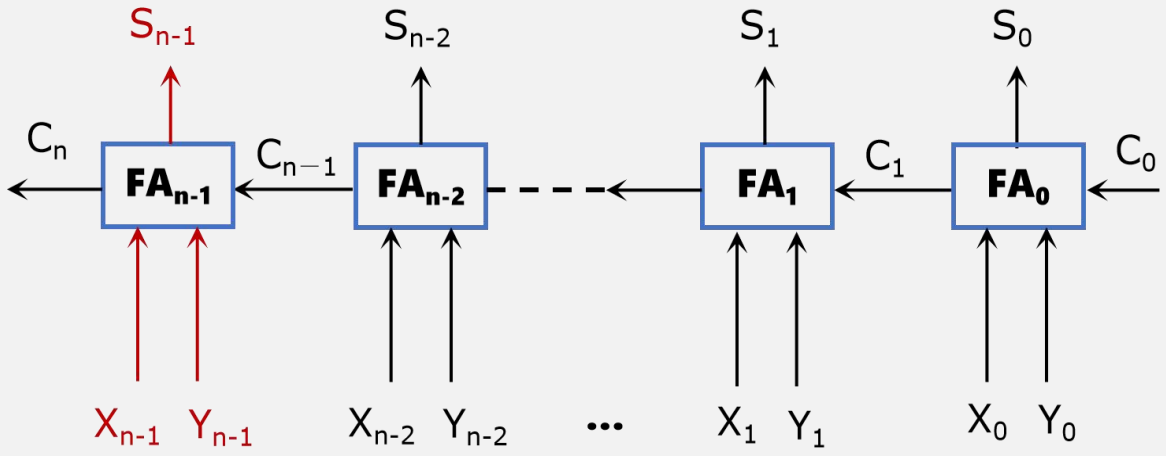
硬件构件模块应是具有完整逻辑功能、可单独使用的电路模块。因此，设计构件模块时，首先要确定对应逻辑功能是否可被分解与重用，确定可被分解后，再从是否便于分解、设计、重用等角度综合确定分解方法、构件模块的规模、分解与重用的策略等

3. 硬件迭代设计

2)具有迭代性的硬件构件模块设计方法

加法功能可被分解、迭代使用方便、可扩展性好!

```
  1 0 1 0 0 1
+ 1 1 1 0 0 0
  0 1 1 0 0 0
-------------
1 1 0 0 0 0 1
```



3. 硬件迭代设计

2)具有迭代性的硬件构件模块设计方法

(2)分析构件模块的可扩展机制

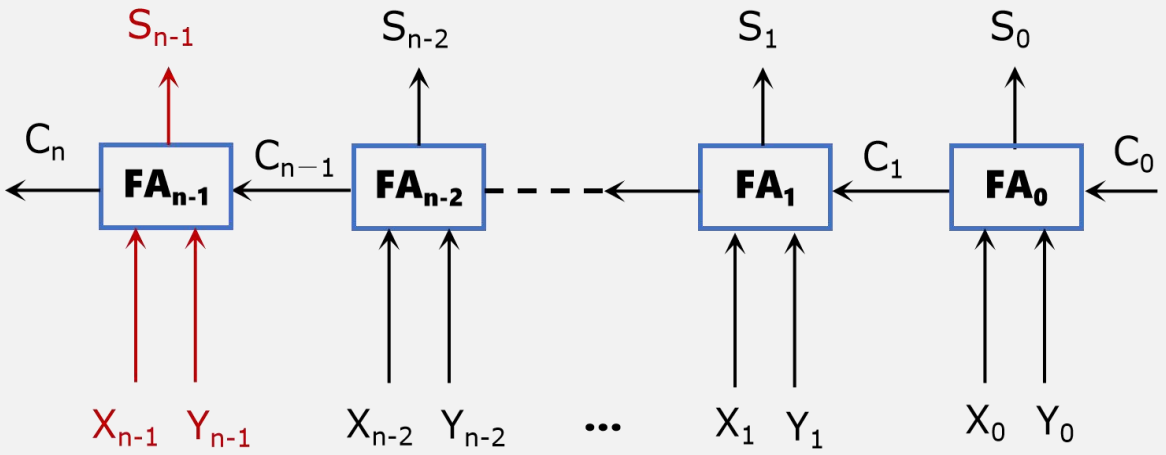
可扩展是构建模块应用于迭代设计的根本属性，是构件模块为支持迭代设计所需的输入、输出及控制信息。不同逻辑功能的构件模块具有不同的可扩展机制。如何找到构建模块的可扩展机制是构件模块设计的重点与难点。

3. 硬件迭代设计

2)具有迭代性的硬件构件模块设计方法

(2)分析构件模块的可扩展机制

```
    1 0 1 0 0 1
+   1 1 1 0 0 0
  0 1 1 0 0 0
1 1 0 0 0 0 1
```



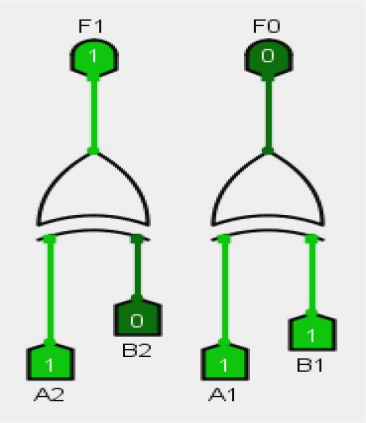
一位全加器的可扩展机制是什么?

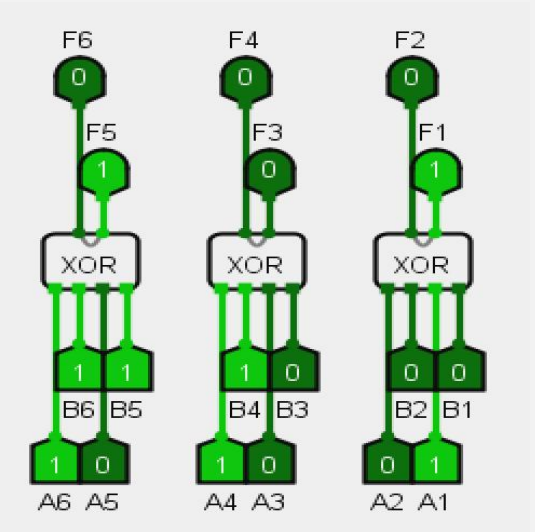
3. 硬件迭代设计

2)具有迭代性的硬件构件模块设计方法

(2)分析构件模块的可扩展机制

```
  1 0 1 0 0 1
⊕ 1 1 1 0 0 0
-------------
  0 1 0 0 0 1
```



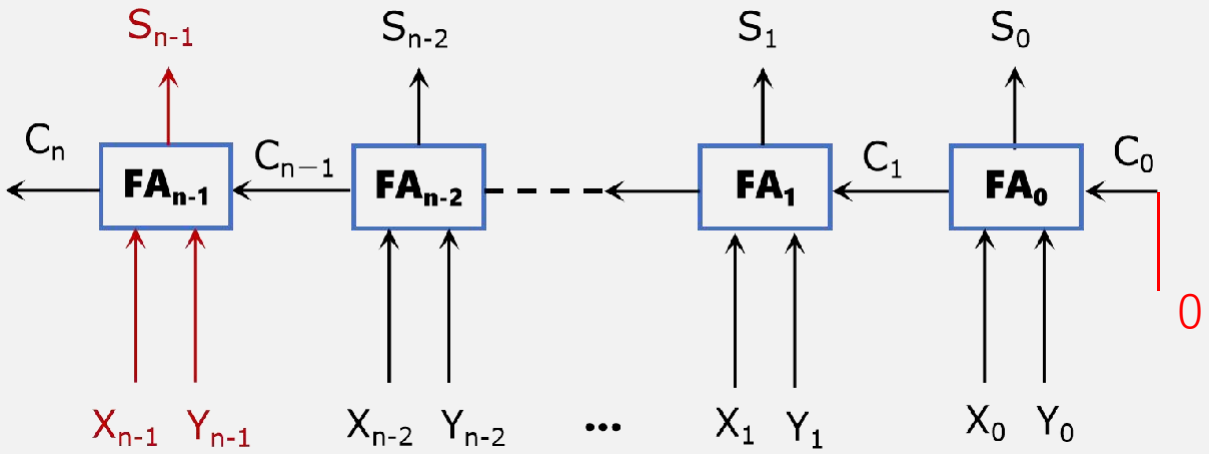


3. 硬件迭代设计

2)具有迭代性的硬件构件模块设计方法

(3)定义构件模块的可扩展接口

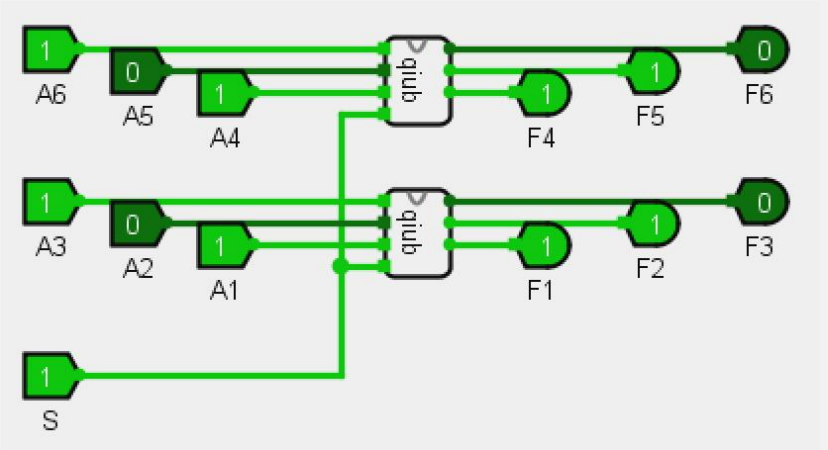
构件模块的可扩展机制通过其接口定义与实现，包括输入、输出和控制等接口。应清晰表达可扩展接口的逻辑功能、输入输出属性、使用方法和使用时应关注的问题。



4. 可迭代的求补电路设计

1)设计求补构件模块(电路)

输入端				输出端		
S	A_2	A_1	A_0	F_2	F_1	F_0
1	0	0	0	0	0	0
1	0	0	1	1	1	1
1	0	1	0	1	1	0
1	0	1	1	1	0	1
1	1	0	0	1	0	0
1	1	0	1	0	1	1
1	1	1	0	0	1	0
1	1	1	1	0	0	1
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	1	0	0
0	1	0	1	1	0	1
0	1	1	0	1	1	0
0	1	1	1	1	1	1



原有设计的问题在哪里?

4. 可迭代的求补电路设计

1)设计求补构件模块(电路)

(1)判定逻辑功能的可分解性 → 求补功能可分解

(2)分析构件模块的可扩展机制 → 从右向左第一次出现 1，注重机制的传递

(3)定义构件模块的可扩展接口 → 输出(C1): 本模块处理的数据中是否首次出现“1或传输从底位模块传输过来的相关信息;

输位(C0): 接收其他模块输出有关出现”1”的信息。

(4)设计构件模块

$$x = -1011001$$

$$[x]_{补} = 10100111$$

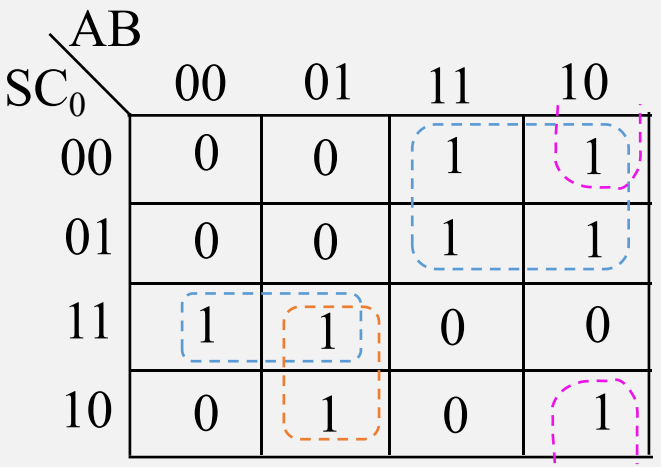
4. 可迭代的求补电路设计

1)设计求补构件模块(电路)

A	B	S	C0	F1	F0	C1
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	1	1	1
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	1	1	1
0	1	1	1	1	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	1	0	1
1	0	1	1	0	1	1
1	1	0	0	1	1	0
1	1	0	1	1	1	0
1	1	1	0	0	1	1
1	1	1	1	0	0	1

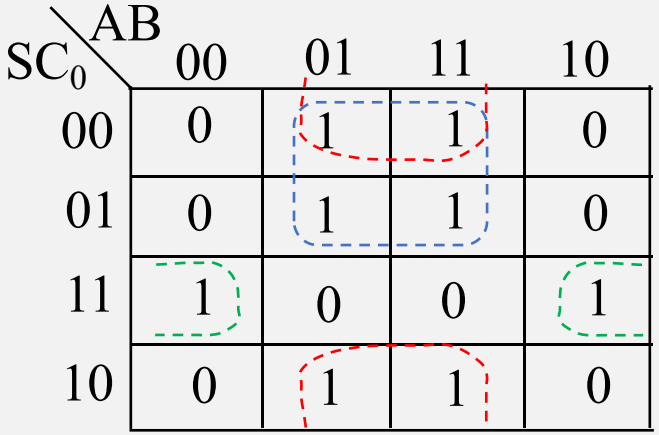
F1=Σm(3,6,7,8,9,10,12,13)

F0=Σm(3,4,5,6,11,12,13,14)

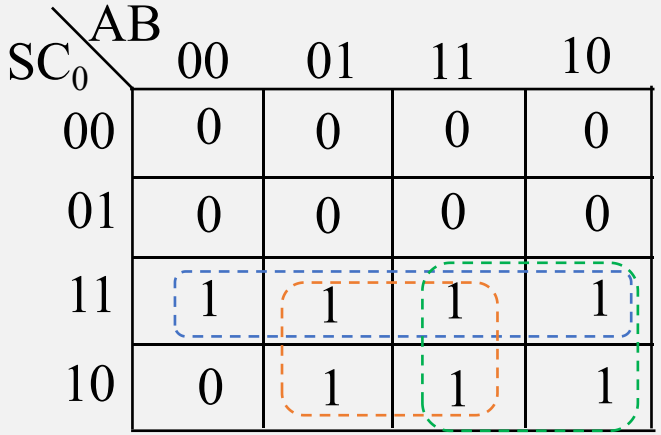


F2= $A\overline{S}$ +$\overline{A}$ SC0 + $\overline{A}$BS + $A\overline{B}\overline{C0}$

C1=Σm(3,6,7,10,11,14,15)



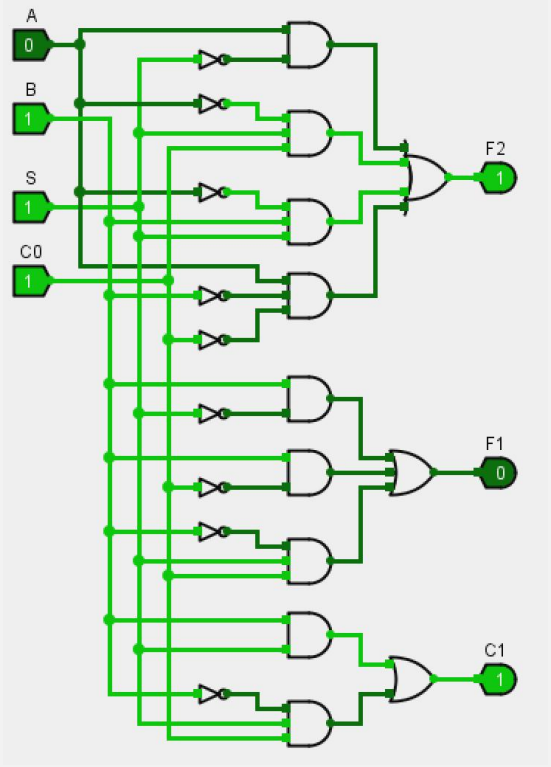
F1= $B\overline{S}$ +$B\overline{C0}$ + $\overline{B}$ SC0



C1= SC0 +BS +AS

4. 可迭代的求补电路设计

1)设计求补构件模块(电路)



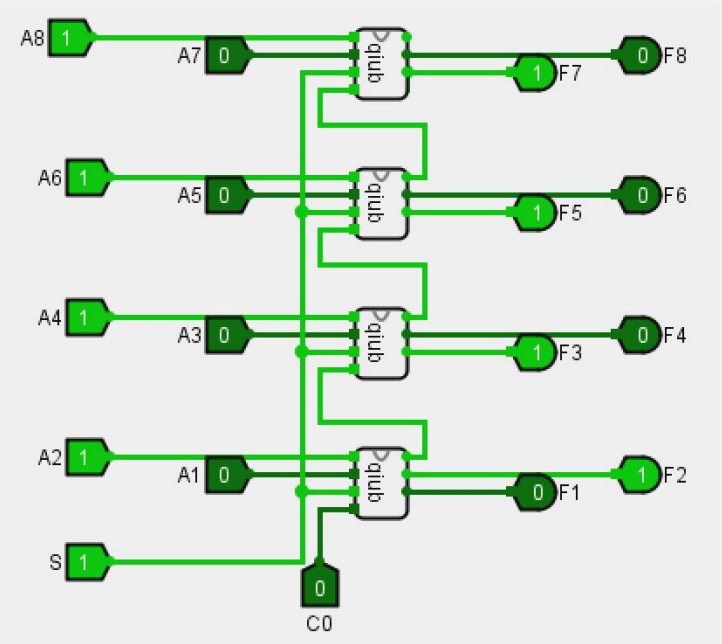
$$F2 = A\overline{S} + \overline{A}\,SC0 + \overline{A}BS + A\overline{B}\overline{C0}$$

$$F1 = B\overline{S} + B\overline{C0} + \overline{B}\,SC0$$

$$C1 = SC0 + BS + AS$$

4. 可迭代的求补电路设计

2)设计规模可扩展求补电路



5.编码器

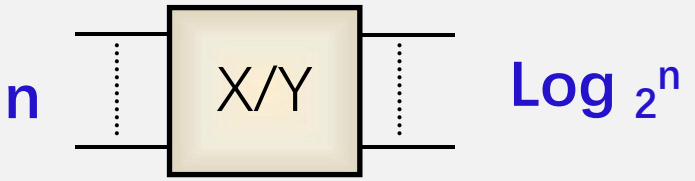
1)编码的基本概念

用 文字、数字、符号等标识特定对象，将数据从一种形式变成另一种形式。

传感器就是一种常见的编码器，如位置传感器、压力传感器等。

2)编码器

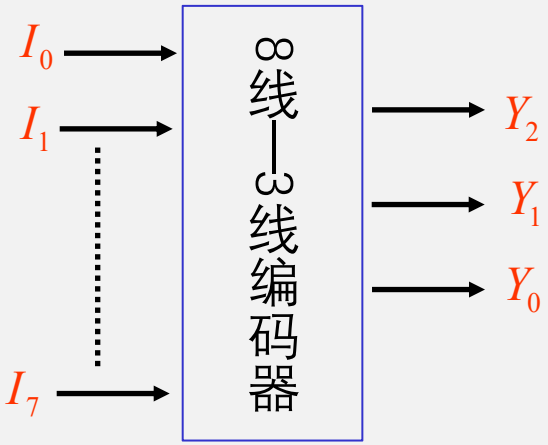
能够完成编码功能的电路叫编码器



3)编码器分类

普通编码器和优先编码器

4)普通编码器



8线－3线编码器：

输入是I_0-$I_7$8个高电平信号，

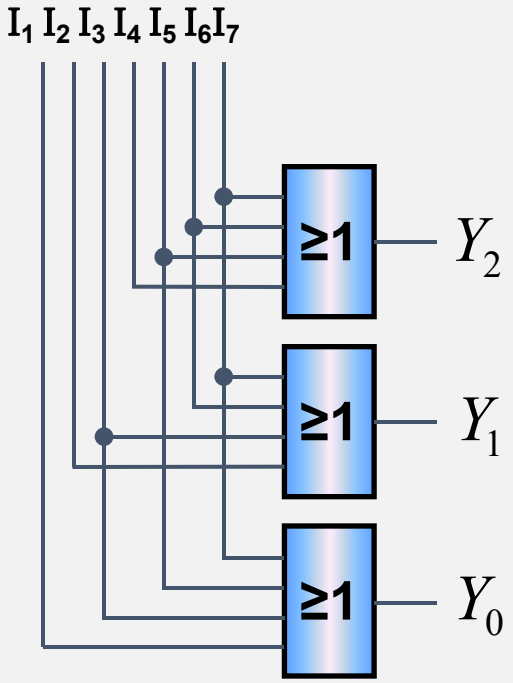
输出是3位二进制代码$Y_2Y_1Y_0$

任何时刻只允许输入一个编码信号，不允许同时输入多个编码信号

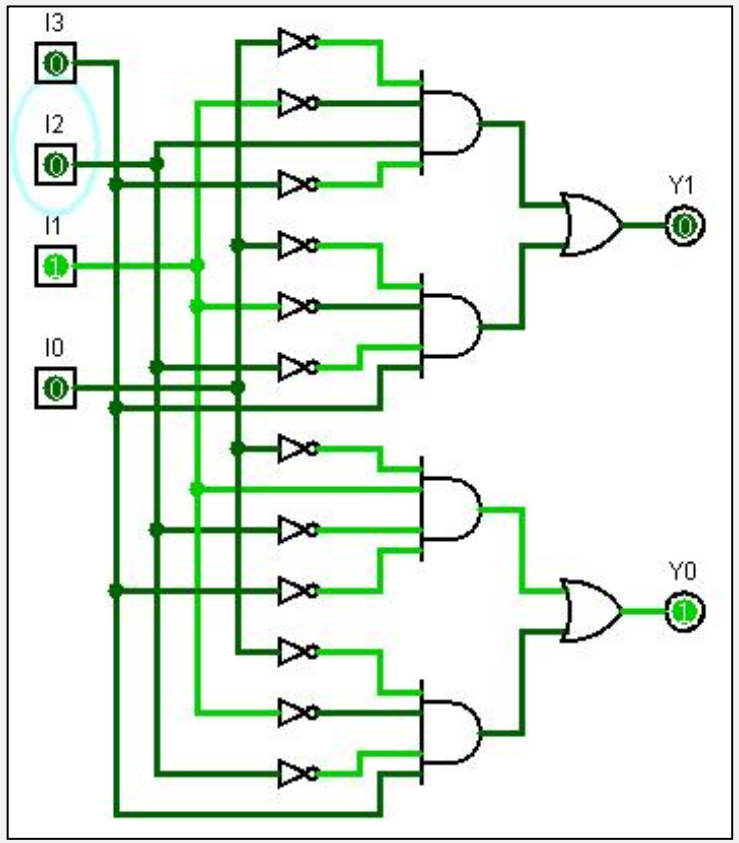
4.6 基本组合逻辑功能部件设计

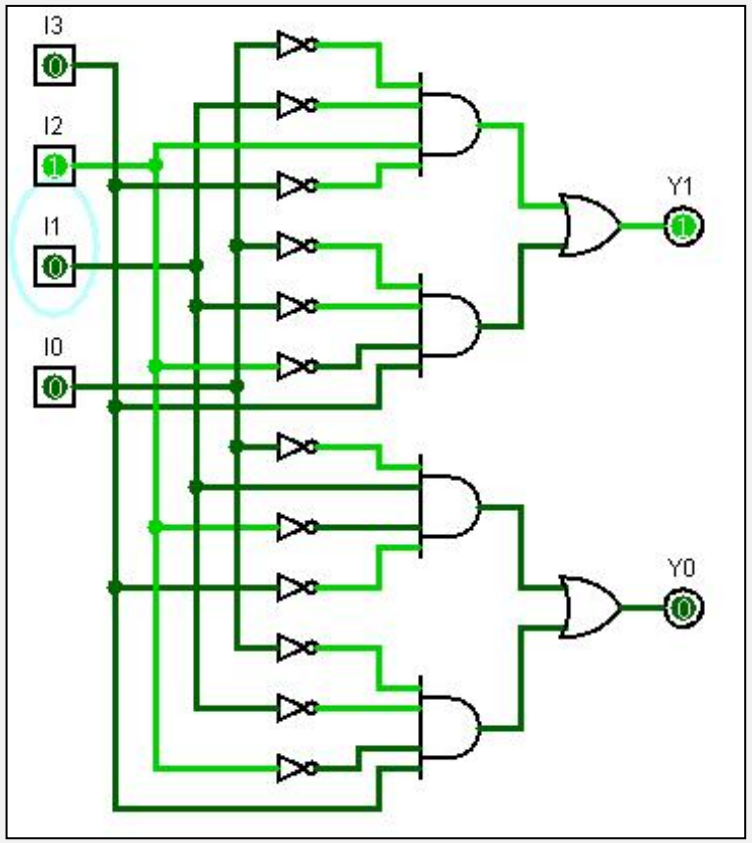
输入								输出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

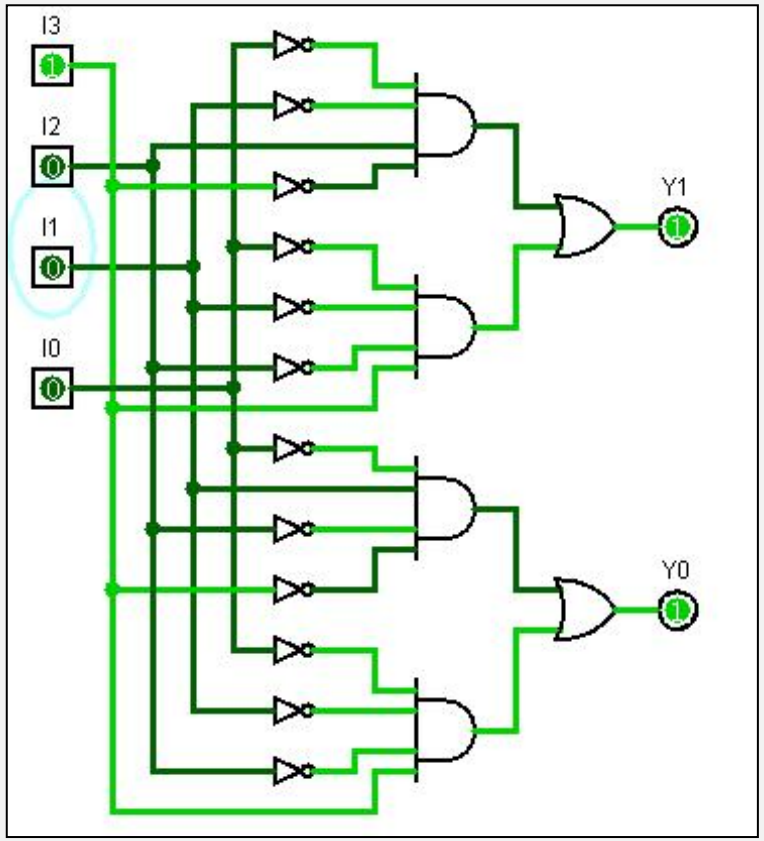
$$Y_2 = I_4 + I_5 + I_6 + I_7$$
$$Y_1 = I_2 + I_3 + I_6 + I_7$$
$$Y_0 = I_1 + I_3 + I_5 + I_7$$



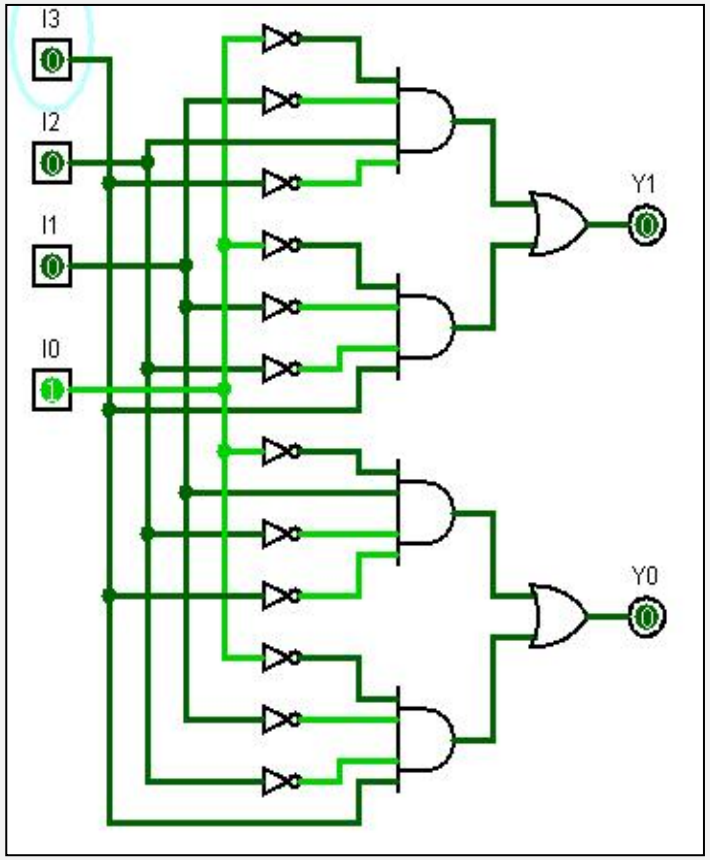
4.6 基本组合逻辑功能部件设计

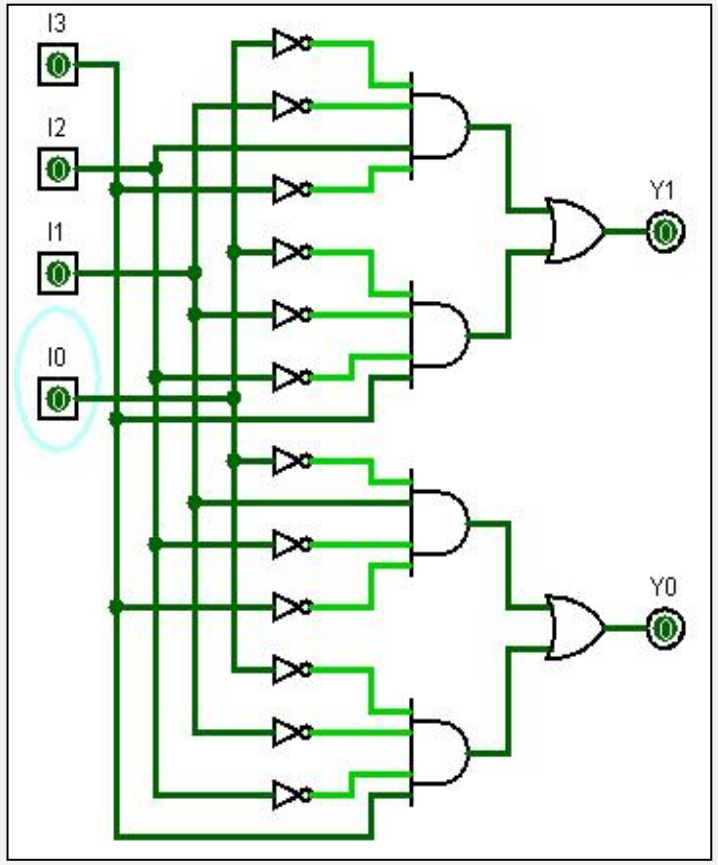


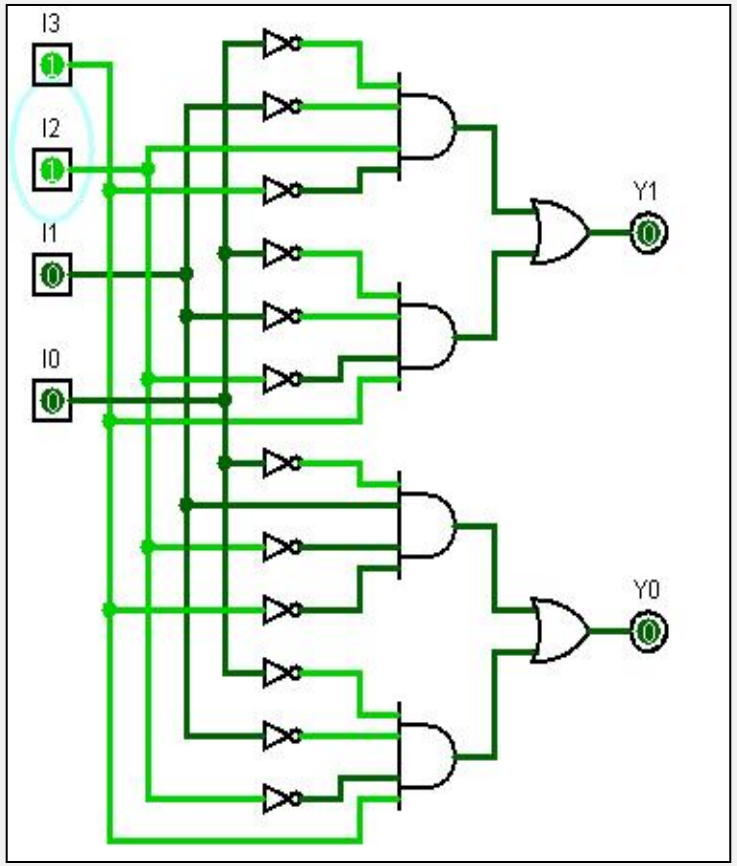




4.6 基本组合逻辑功能部件设计







发现存在的问题! 如何解决?

5)优先编码器

优先编码器允许同时在几个输入端有输入信号，编码器按输入信号排定的优先顺序，只对同时输入的几个信号中优先权最高的输入进行编码。

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

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

$$Y_1 = I_7 + \overline{I_7} \cdot I_6 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot \overline{I_4} \cdot I_3 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot \overline{I_4} \cdot \overline{I_3} \cdot I_2 = I_7 + I_6 + \overline{I_5} \cdot \overline{I_4} \cdot (I_3 + I_2)$$

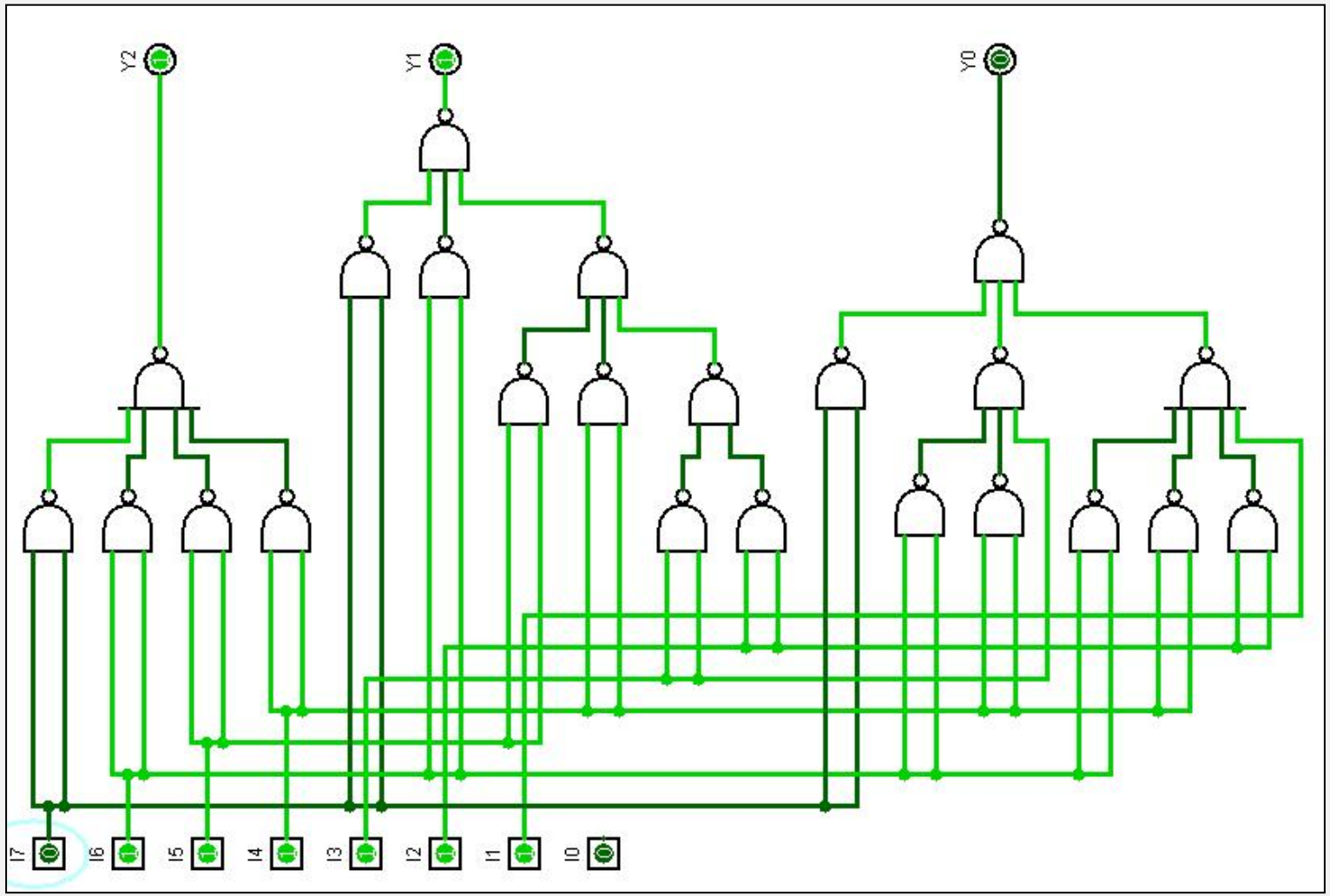
$$Y_0 = I_7 + \overline{I_7} \cdot \overline{I_6} \cdot I_5 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot \overline{I_4} \cdot I_3 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot \overline{I_4} \cdot \overline{I_3} \cdot \overline{I_2} \cdot I_1 = I_7 + \overline{I_6}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

4.6 基本组合逻辑功能部件设计

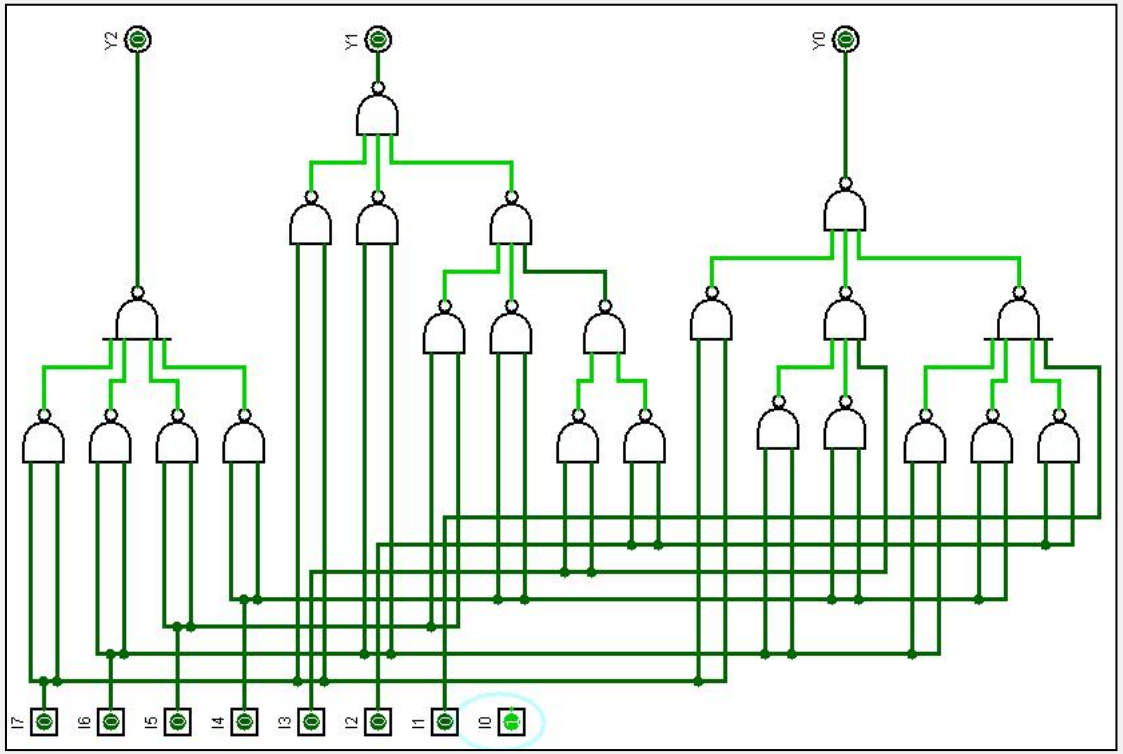
$$Y_2 = I_4 + I_5 + I_6 + I_7$$
$$Y_1 = I_2 + I_3 + I_6 + I_7$$
$$Y_0 = I_1 + I_3 + I_5 + I_7$$

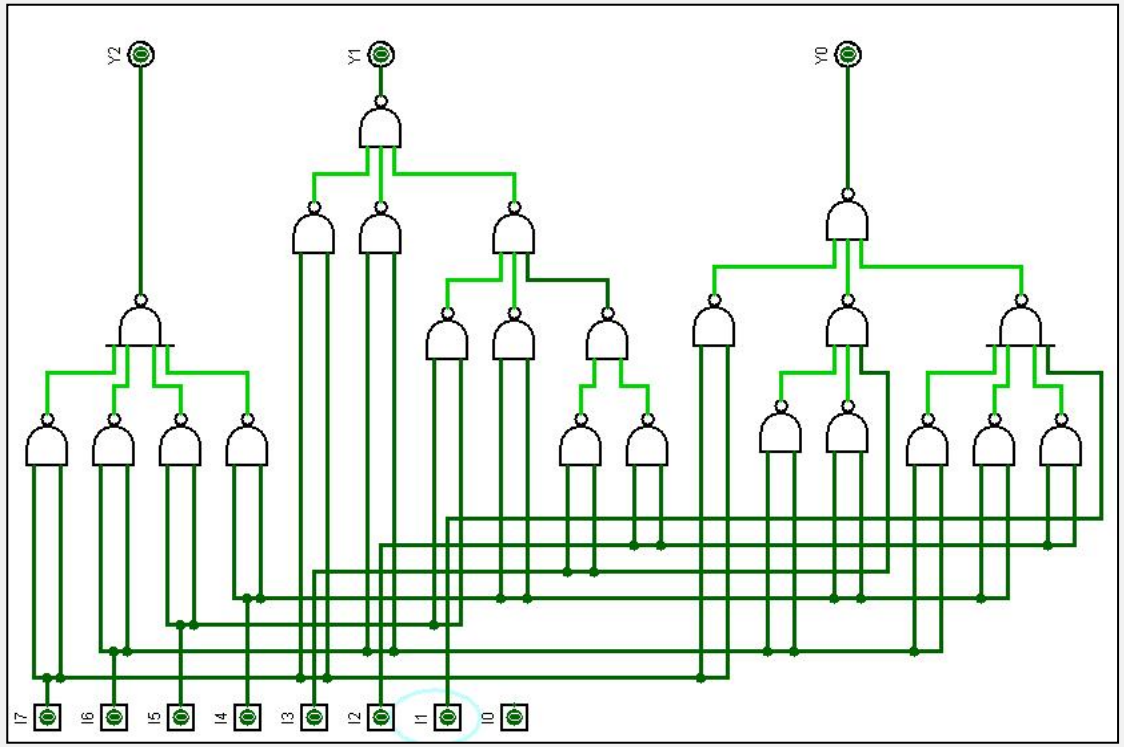
$$Y_2 = I_7 + \overline{I_7} \cdot I_6 + \overline{I_7} \cdot \overline{I_6} \cdot I_5 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot I_4 = I_7 + I_6 + I_5 + I_4$$
$$Y_1 = I_7 + \overline{I_7} \cdot I_6 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot \overline{I_4} \cdot I_3 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot \overline{I_4} \cdot \overline{I_3} \cdot I_2 = I_7 + I_6 + \overline{I_5} \cdot \overline{I_4} \cdot (I_3 + I_2)$$
$$Y_0 = I_7 + \overline{I_7} \cdot \overline{I_6} \cdot I_5 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot \overline{I_4} \cdot I_3 + \overline{I_7} \cdot \overline{I_6} \cdot \overline{I_5} \cdot \overline{I_4} \cdot \overline{I_3} \cdot \overline{I_2} \cdot I_1 = I_7 + \overline{I_6} \cdot I_5 + \overline{I_6} \cdot \overline{I_4} \cdot I_3 + \overline{I_6} \cdot \overline{I_4} \cdot \overline{I_2} \cdot I_1$$

4.6 基本组合逻辑功能部件设计

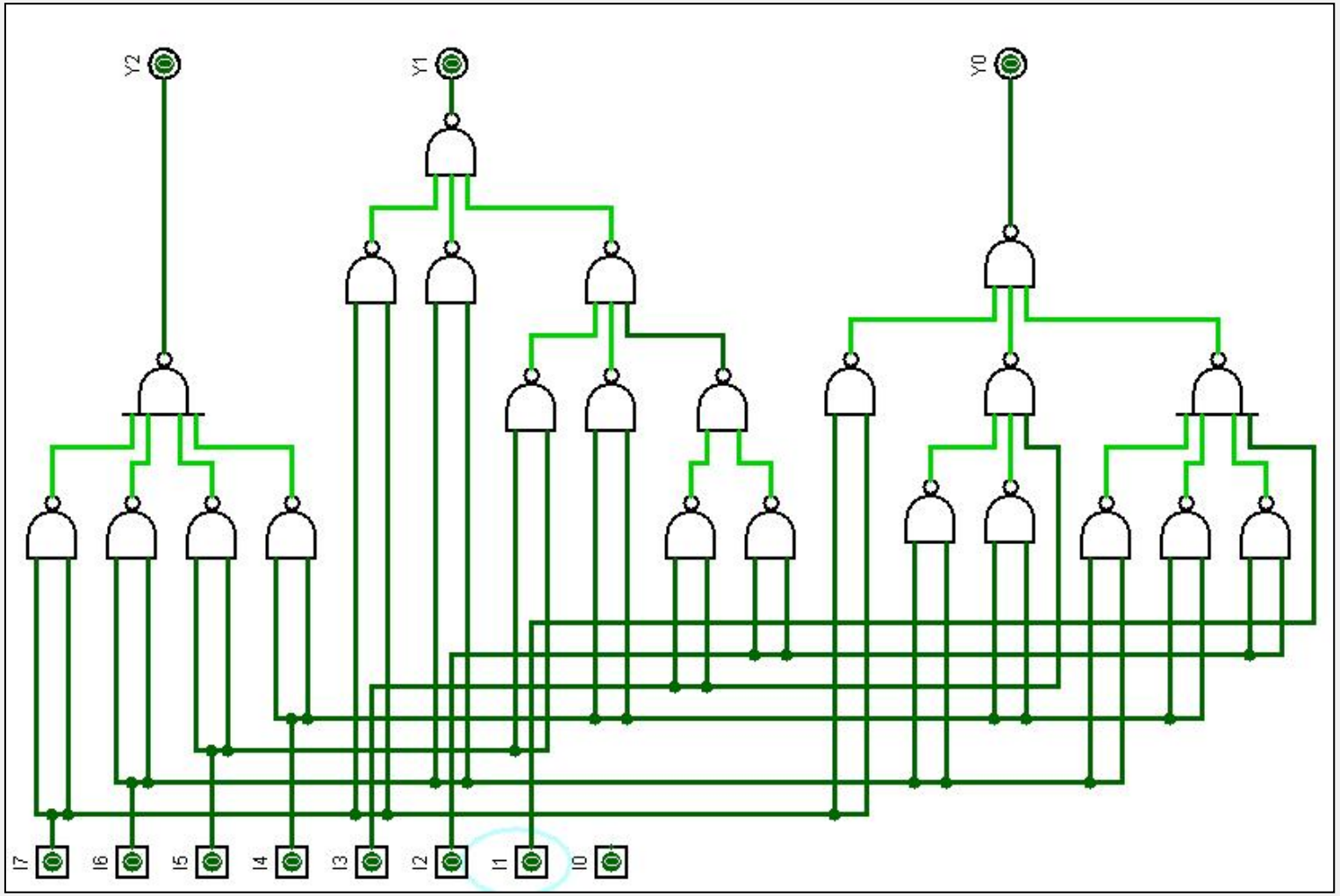


4.6 基本组合逻辑功能部件设计





4.6 基本组合逻辑功能部件设计



6)带使能控制的优先编码器

输入使能端	输入								输出			扩展	使能输出
$\overline{S}$	$\overline{I_7}$	$\overline{I_6}$	$\overline{I_5}$	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	$\overline{I_0}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$	$\overline{Y_{EX}}$	$\overline{Y_s}$
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	×	×	×	×	×	×	×	0	0	0	0	1
0	1	0	×	×	×	×	×	×	0	0	1	0	1
0	1	1	0	×	×	×	×	×	0	1	0	0	1
0	1	1	1	0	×	×	×	×	0	1	1	0	1
0	1	1	1	1	0	×	×	×	1	0	0	0	1
0	1	1	1	1	1	0	×	×	1	0	1	0	1
0	1	1	1	1	1	1	0	×	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

$$\overline{Y}_2 = \overline{(I_4 + I_5 + I_6 + I_7) \cdot ST} \qquad \overline{Y}_1 = \overline{(I_2\overline{I}_4\overline{I}_5 + I_3\overline{I}_4\overline{I}_5 + I_6 + I_7) \cdot ST}$$

$$\overline{Y}_0 = \overline{(I_1\overline{I}_2\overline{I}_4\overline{I}_6 + I_3\overline{I}_4\overline{I}_6 + I_5\overline{I}_6 + I_7) \cdot ST} \qquad \overline{Y_S} = \overline{\overline{I_0} \cdot \overline{I_1} \cdot \overline{I_2} \cdot \overline{I_3} \cdot \overline{I_4} \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7} \cdot S}$$

6)带使能控制的优先编码器

输入使能端	输入								输出			扩展	使能输出
$\overline{S}$	$\overline{I_7}$	$\overline{I_6}$	$\overline{I_5}$	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	$\overline{I_0}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$	$\overline{Y_{EX}}$	$\overline{Y_s}$
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	×	×	×	×	×	×	×	0	0	0	0	1
0	1	0	×	×	×	×	×	×	0	0	1	0	1
0	1	1	0	×	×	×	×	×	0	1	0	0	1
0	1	1	1	0	×	×	×	×	0	1	1	0	1
0	1	1	1	1	0	×	×	×	1	0	0	0	1
0	1	1	1	1	1	0	×	×	1	0	1	0	1
0	1	1	1	1	1	1	0	×	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

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

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

6.译码器

1)基本概念

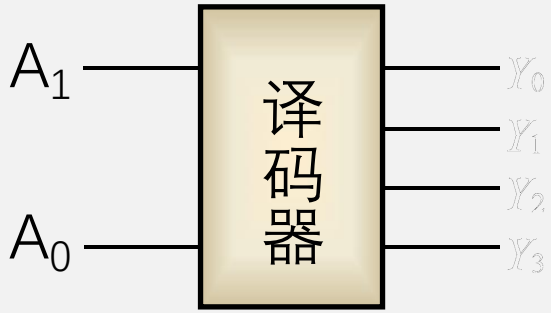
编码器的逆过程，将输入的每个二进制代码翻译成对应的输出高、低电平。

2)译码器分类

- 变量译码器
- 码制变换译码器
- 数字显示译码器

3)变量译码器

变量译码器是表示输入状态的组合逻辑网络，如2:4译码器



对输入的2位二进制数进行译码，具有 $2^2 = 4$ 个输出

4)2:4变量译码器

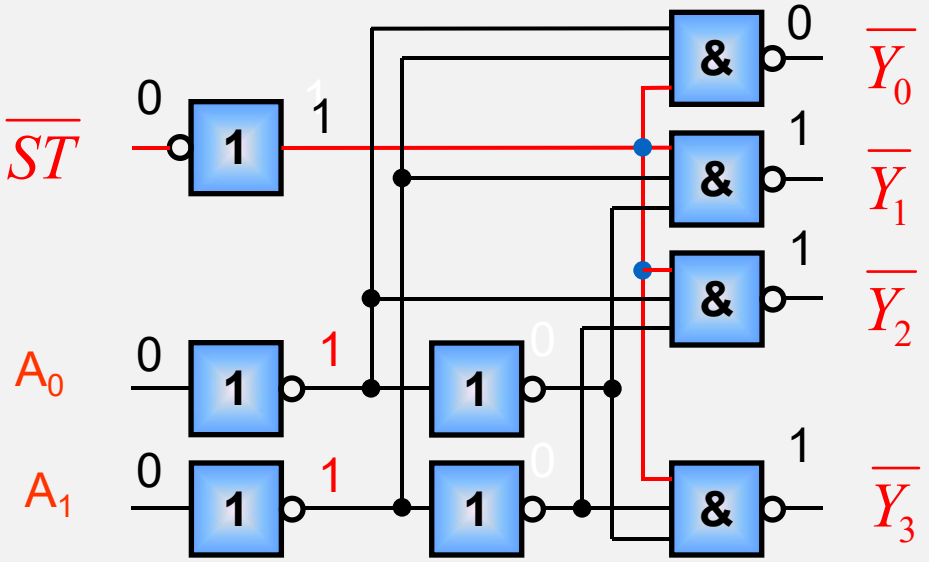
A_1	A_0	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

$$\overline{Y_3} = \overline{A_1 A_0} \qquad \overline{Y_2} = \overline{A_1 \overline{A_0}}$$

$$\overline{Y_1} = \overline{\overline{A_1} A_0} \qquad \overline{Y_0} = \overline{\overline{A_1}\,\overline{A_0}}$$

5)带选通功能的2:4变量译码器

$\overline{ST}$	A_1	A_0	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1



$$\overline{Y_3} = \overline{A_1 A_0 \overline{\overline{ST}}} = \overline{A_1 A_0 ST}$$

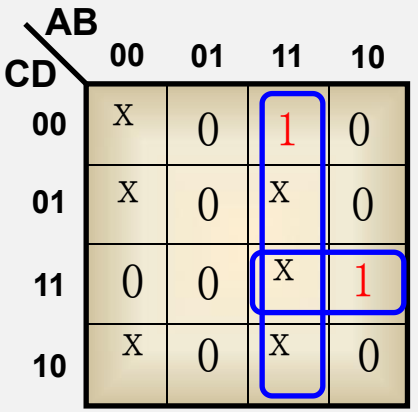
$$\overline{Y_2} = \overline{A_1 \overline{A_0} ST} \qquad \overline{Y_1} = \overline{\overline{A_1} A_0 ST} \qquad \overline{Y_0} = \overline{\overline{A_1}\,\overline{A_0} ST}$$

7.码制变换译码器

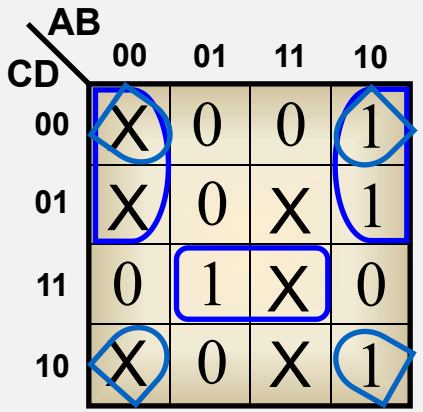
码制变换译码器的功能是将一种码制转换为另一种码制。

例1: 设计一个将余三码转换为8421BCD码的转换电路

A	B	C	D	W	X	Y	Z
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1



W



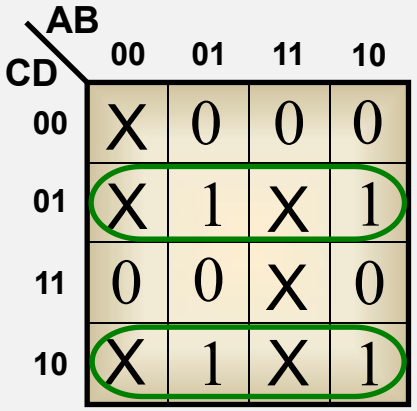
X

$$W = AB + ACD$$

$$X = \overline{B}\,\overline{C} + \overline{B}\,\overline{D} + BCD$$

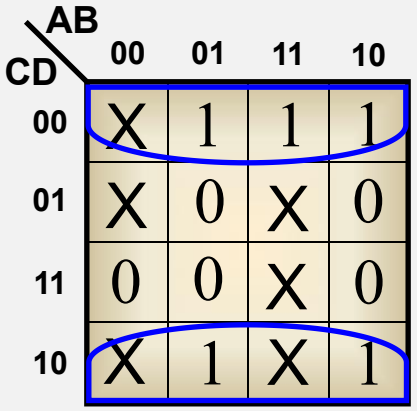
4.6 基本组合逻辑功能部件设计

A	B	C	D	W	X	Y	Z
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1



Y

$$Y = \overline{C}D + C\overline{D}$$



Z

$$Z = \overline{D}$$

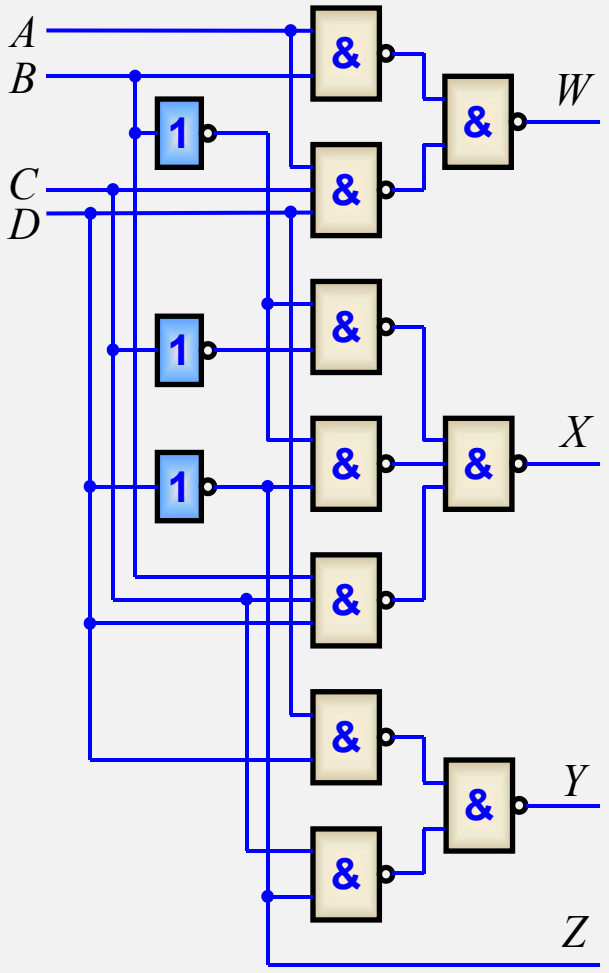
4.6 基本组合逻辑功能部件设计

$$W = \overline{\overline{AB} \cdot \overline{ACD}}$$

$$X = \overline{\overline{\overline{AC}} \cdot \overline{\overline{BD}} \cdot \overline{BCD}}$$

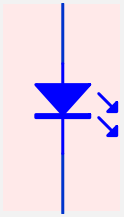
$$Y = \overline{\overline{C\overline{D}} \cdot \overline{C\overline{D}}}$$

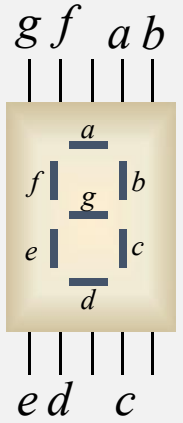
$$Z = \overline{D}$$

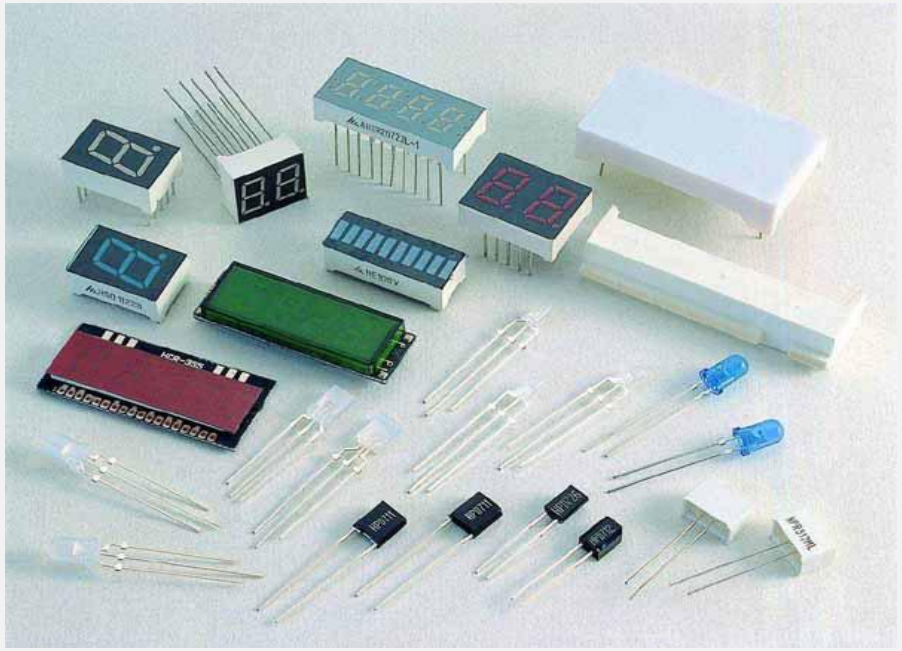


8. 数字显示译码器

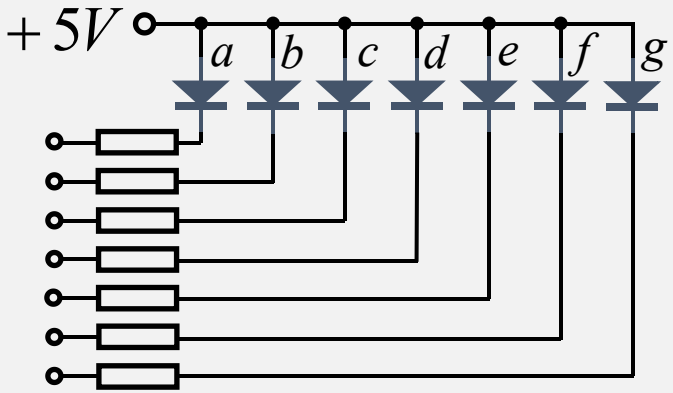
发光二极管可以单独封装，也可以组合封装为LED数码管。



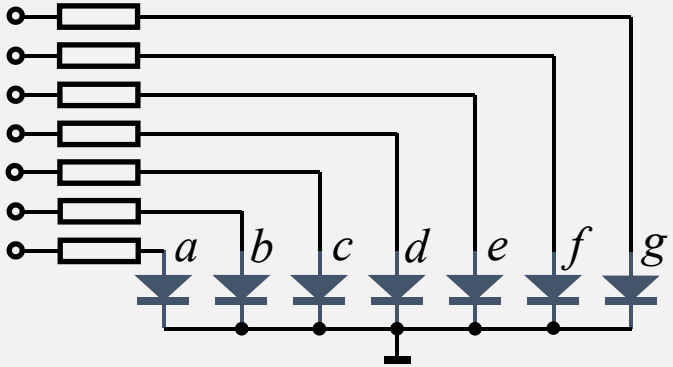




发光二极管按驱动方式又分为共阳极和共阴极接法。



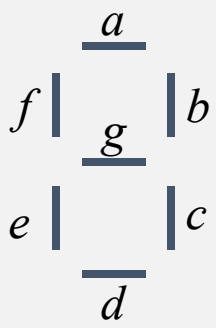
共阳极接法



共阴极接法

4.6 基本组合逻辑功能部件设计

例 设计8421BCD七段显示译码电路。

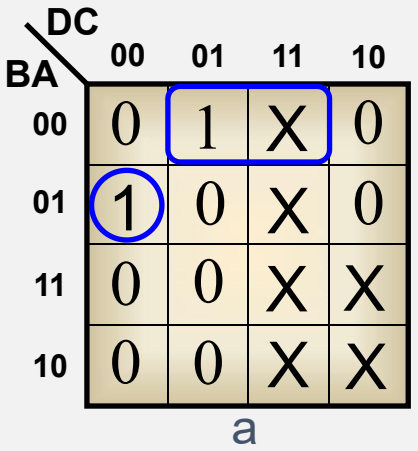


本题采用共阳极设计

D	C	B	A	a	b	c	d	e	f	g	显示
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	1	0	0	9

4.6 基本组合逻辑功能部件设计

D	C	B	A	a	b	c	d	e	f	g	显示
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	1	0	0	9



$$a = \overline{D}\,\overline{C}\,\overline{B}A + C\,\overline{B}\,\overline{A}$$

4.6 基本组合逻辑功能部件设计

BA \ DC	00	01	11	10
00	0	0	X	0
01	0	1	X	0
11	0	0	X	X
10	0	1	X	X

b

$$b = C\overline{B}A + CB\overline{A}$$

BA \ DC	00	01	11	10
00	0	0	X	0
01	0	0	X	0
11	0	0	X	X
10	1	0	X	X

c

$$c = \overline{C}\overline{B}A$$

BA \ DC	00	01	11	10
00	0	1	X	0
01	1	0	X	0
11	0	1	X	X
10	0	0	X	X

d

$$d = \overline{C}\overline{B}A + CBA + \overline{D}CBA$$

BA \ DC	00	01	11	10
00	0	1	X	0
01	1	1	X	1
11	1	1	X	X
10	0	0	X	X

e

$$e = A + \overline{C}\overline{B}A$$

BA \ DC	00	01	11	10
00	0	0	X	0
01	1	0	X	0
11	1	1	X	X
10	1	0	X	X

f

$$f = BA + \overline{C}B\overline{A} + \overline{D}CBA$$

BA \ DC	00	01	11	10
00	1	0	X	0
01	1	0	X	0
11	0	1	X	X
10	0	0	X	X

g

$$g = \overline{D}\overline{C}B + CBA$$

4.6 基本组合逻辑功能部件设计

$$a = \overline{D}\overline{C}\overline{B}A + C\overline{B}\overline{A}$$

$$b = \overline{C}\overline{B}A + C\overline{B}\overline{A}$$

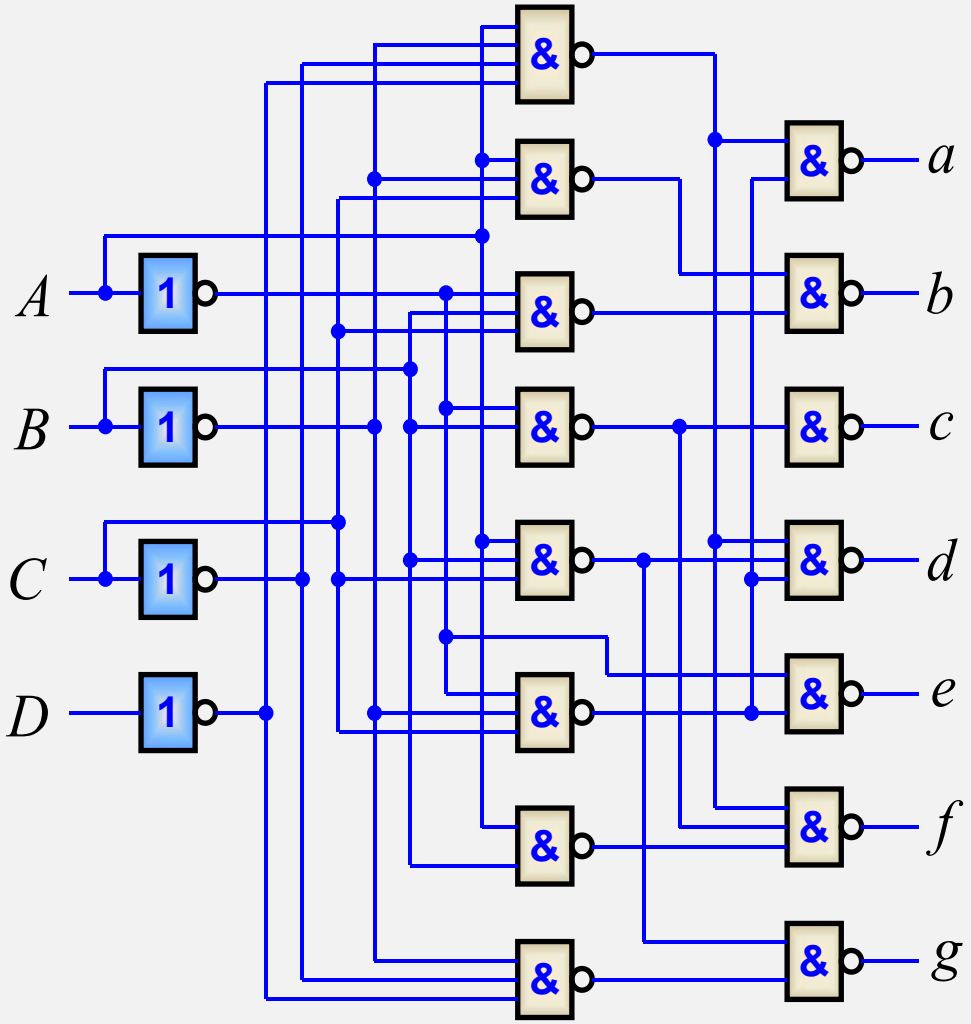
$$c = \overline{C}\overline{B}\overline{A}$$

$$d = \overline{C}\overline{B}\overline{A} + \overline{C}BA + \overline{D}\overline{C}\overline{B}A$$

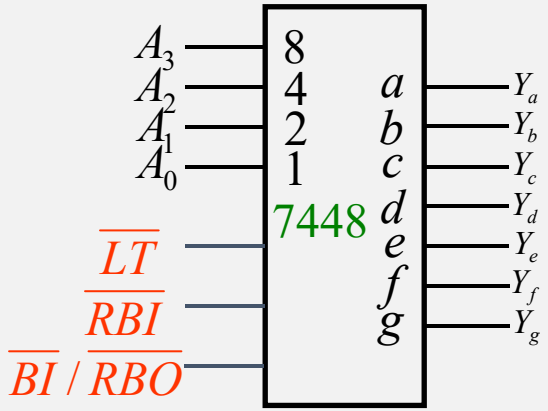
$$e = A + \overline{C}\overline{B}\overline{A}$$

$$f = BA + \overline{C}\overline{B} + \overline{D}\overline{C}\overline{B}A$$

$$g = \overline{D}\overline{C}\overline{B} + CBA$$



数字显示译码器的动态显示



灯测试输入端主要用于检查LED的好坏。

$\overline{LT}=$ { 1 时，正常译码。
0 时，输出 a ~ g 全 “1” 七段全亮。

消隐输入端（与灭0输出端共用）

$\overline{BI}=$ { 0 时，不管输入何种状态，输出全0
1 时，正常译码。

灭0输入端，熄灭无意义的0

$\overline{RBI}=$ { 0 时，灭掉不要显示的0，001→1
1 时，显示0,不灭中间0。101→101

灭0输出端 $\overline{RBO}$ 与(灭0输入端配合使用)

即：灭0输入等于0,灭0输出一定等于0。