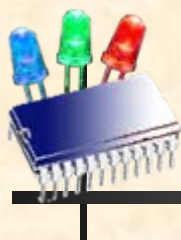


第3章 组合逻辑电路

本章主要介绍组合逻辑电路的特点、组合逻辑电路的分析方法和设计方法，分别讨论了编码器、译码器、数据选择器、数据分配器、加法器等常用的组合逻辑功能器件；简要介绍了组合逻辑电路中的竞争冒险。



- 概述
- 组合逻辑电路的分析方法
- 小规模组合逻辑电路的设计方法
- 常用组合逻辑功能器件
- 组合逻辑电路中的竞争-冒险

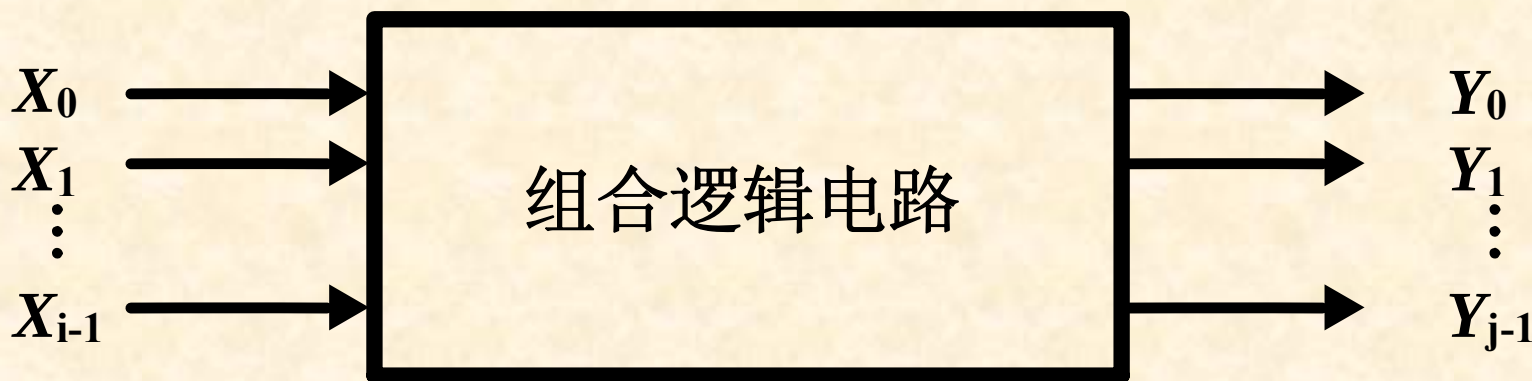




3.1 概述

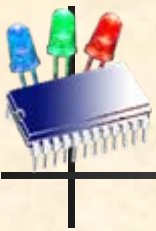
3.1.1 组合逻辑电路的特点

组合逻辑电路有若干个输入和若干个输出。



功能特点:任何时刻的输出仅仅决定于当时的输入信号。

结构特点:不含反馈支路，不包含存储元件。



3.1.2 组合逻辑电路逻辑功能描述方式

描述组合逻辑电路功能的主要方式有以下几种：

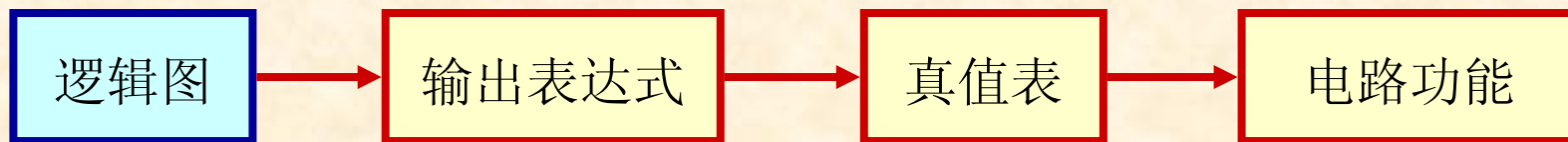
- (1) 表述逻辑功能的逻辑函数表达式；
- (2) 由逻辑符号及相互连线组成的逻辑电路图；
- (3) 描述逻辑功能的真值表或描述逻辑器件的逻辑功能表；
- (4) 反映逻辑电路的输出与输入逻辑关系的波形图；
- (5) 卡诺图不仅可用来化简逻辑函数,同时它也是逻辑函数的一种表现形式。



3.2 组合逻辑电路的分析方法

组合逻辑电路的分析是根据给定的逻辑电路，通过分析找出电路的逻辑功能。

组合逻辑电路的分析过程：根据给定电路，写出输出与输入之间的逻辑表达式，然后把全部输入组合代入表达式，计算得出输出结果，并以真值表的形式表示出来，最后根据真值表说明电路的功能。





例3.2分析电路，并说明电路功能。

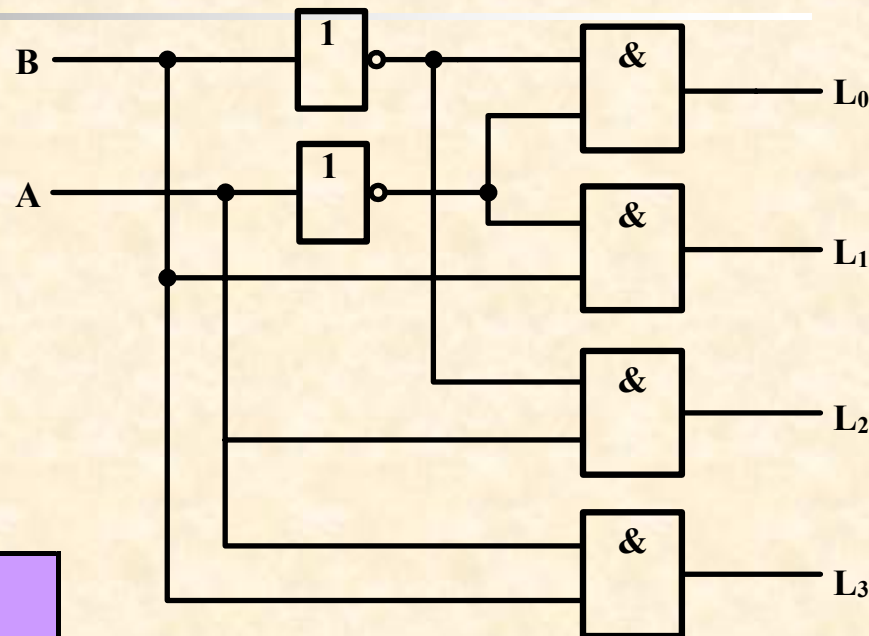
第1步 写出逻辑函数表达式。

$$L_0 = \overline{A} \cdot \overline{B} \quad L_1 = \overline{A}B$$

$$L_2 = A\overline{B} \quad L_3 = AB$$

第2步 列出真值表。

A	B	L ₀	L ₁	L ₂	L 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



第3步 说明电路的功能。

该电路完成的是译码功能，
高电平有效



例3.3 一个双端输入，双端输出的组合逻辑电路如图示，分析该电路的逻辑功能。

第1步 写出函数表达式。

$$Z_1 = AB \quad Z_2 = \bar{A} + Z_1 = \bar{A} + AB \quad Z_3 = \bar{B} + Z_1 = \bar{B} + AB$$

$$S = \overline{Z_2 Z_3} = \overline{(\bar{A} + AB)(\bar{B} + AB)} = A \oplus B$$

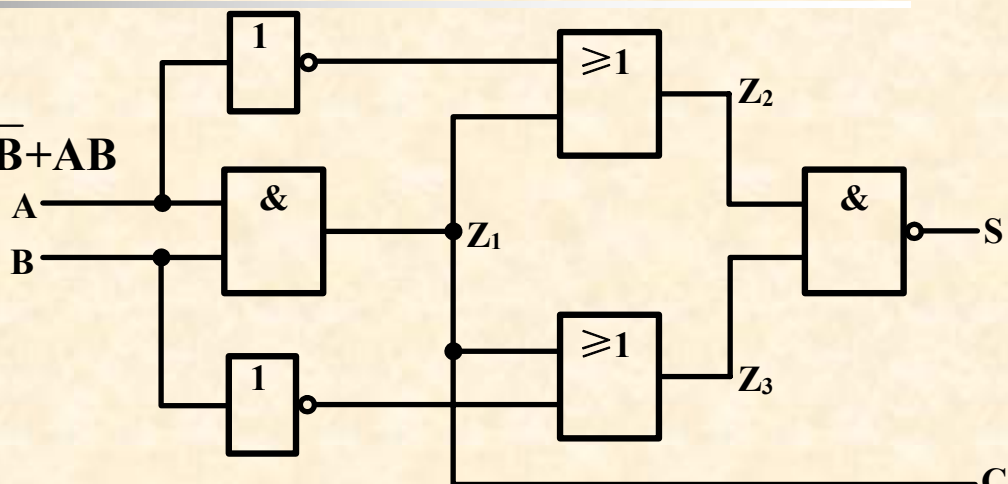
$$C = Z_1 = AB$$

第2步 列出真值表。

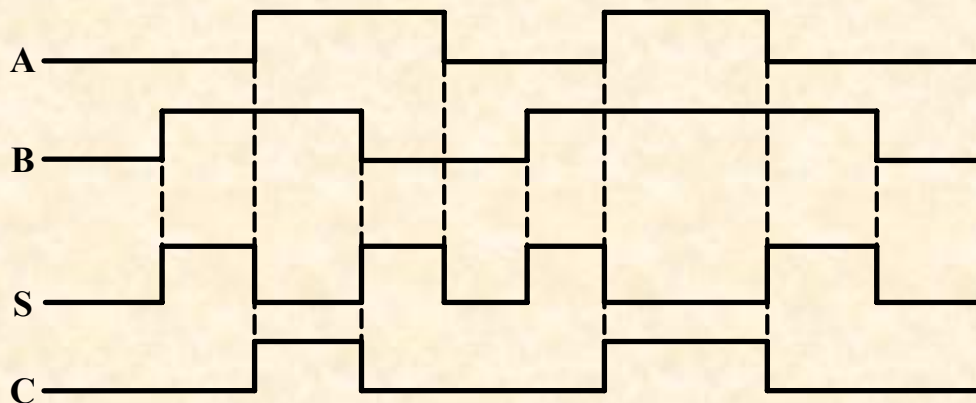
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

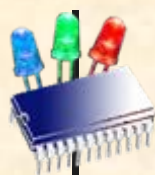
第3步 说明电路的功能。

该电路完成半加功能，A、B是两个加数，S为和数，C为向高位的进位。



第4步 画波形图(可选)





例3.4 一组组合逻辑电路如图示,试分析该电路的逻辑功能。

第1步

写出函数表达式。

$$G_0 = B_0 \oplus B_1$$

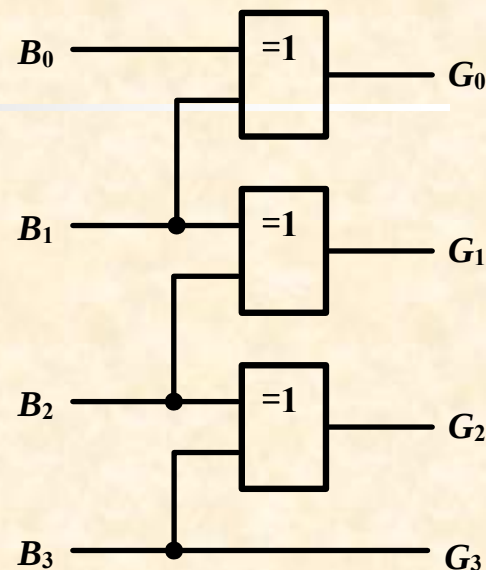
$$G_1 = B_1 \oplus B_2$$

$$G_2 = B_2 \oplus B_3$$

$$G_3 = B_3$$

第2步 列出真值表。

B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀	B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	1
0	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1
0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0
0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	1	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0



第3步 说明电路的功能。

真值表表明, 在4位二进制码输入的情况下, 输出是4位格雷码。因此, 图示电路能够实现4位二进制码到4位格雷码的转换。

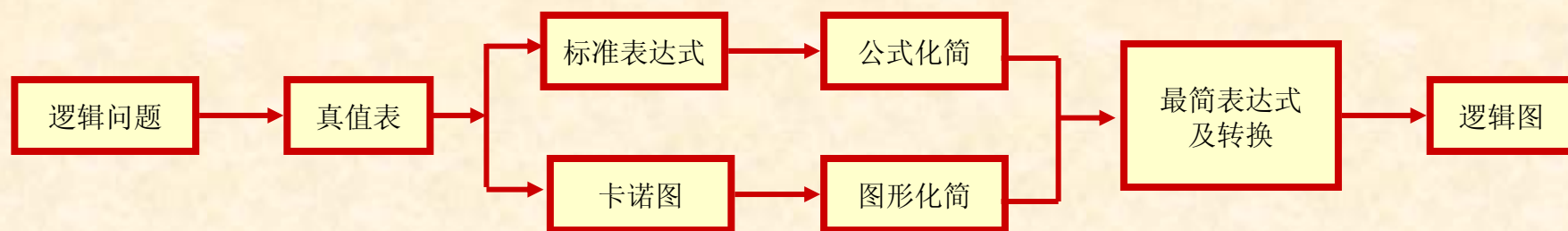




3.3 组合逻辑电路设计的一般方法

组合逻辑电路的设计就是在给定逻辑功能及要求的条件下，通过设计方法，得到满足功能要求的逻辑电路。

设计是分析电路的反过程。



- ①逻辑抽象 ②写逻辑函数表达式 ③函数化简 ④表达式转换 ⑤画逻辑图



例3.5 设计一个三变量表决器，其中A有否决权。

(1) 列出真值表

(2) 导出函数式或卡诺图

A	B	C	L
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

卡诺图
10

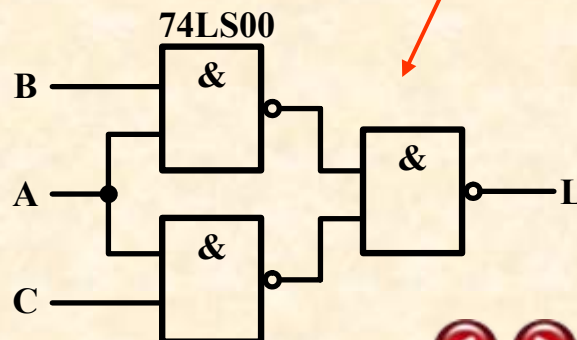
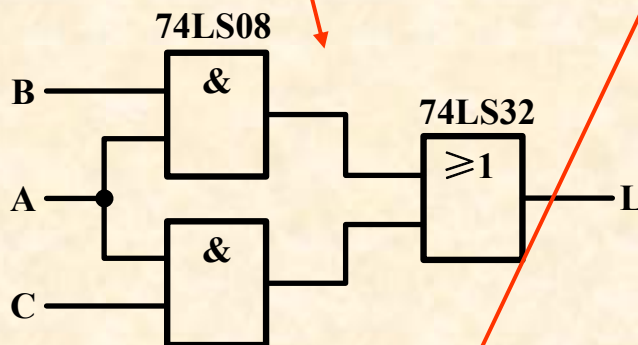
(3) 得到最简式

$$L = AB + AC$$

或

$$L = \overline{\overline{A}\overline{B}} \cdot \overline{\overline{A}\overline{C}}$$

(4) 画逻辑图





例3.6 设计实现四位格雷码到二进制码的电路。

(1) 由4位二进制码到格雷码转换式可得相反的转换式

$$G_3 = B_3$$

$$G_2 = B_2 \oplus B_3$$

$$G_1 = B_1 \oplus B_2$$

$$G_0 = B_0 \oplus B_1$$

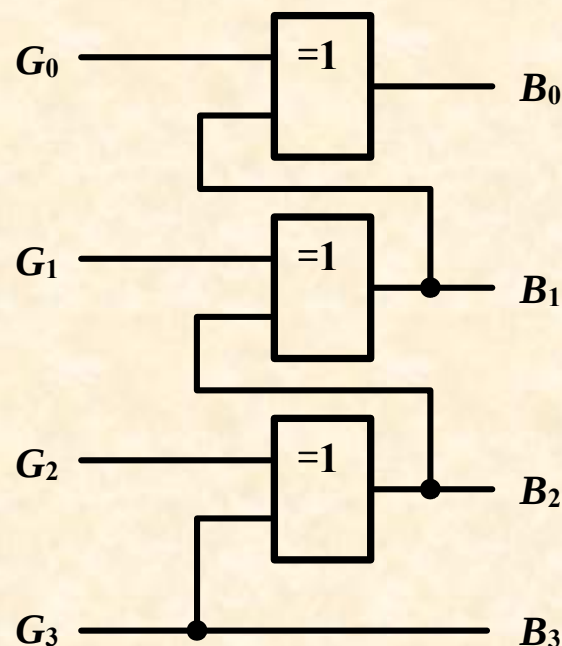
$$B_3 = G_3$$

$$B_2 = G_2 \oplus B_3 = G_2 \oplus G_3$$

$$B_1 = G_1 \oplus B_2 = G_1 \oplus G_2 \oplus G_3$$

$$B_0 = G_0 \oplus B_1 = G_0 \oplus G_1 \oplus G_2 \oplus G_3$$

(2) 画逻辑图



G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
0	0	0	1	0	0	0	1	1	0	0	1	1	1	1	0
0	0	1	0	0	0	1	1	1	0	1	0	1	1	0	0
0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	1
0	1	0	0	0	1	1	1	1	1	0	0	1	0	0	0
0	1	0	1	0	1	1	0	1	1	0	1	1	0	0	1
0	1	1	0	0	1	0	0	1	1	1	0	1	0	1	1
0	1	1	1	0	1	0	1	1	1	1	1	1	0	1	0

卡诺图
11-13





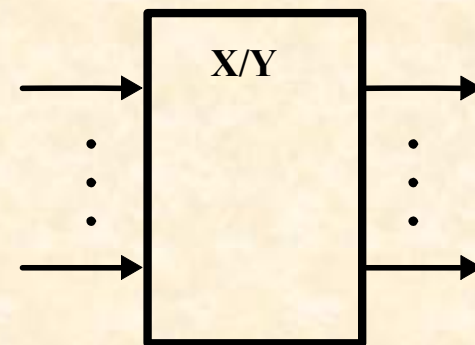
3.4 编码器和译码器

编码器和译码器是常见的组合逻辑基本单元电路，已有标准化的集成电路功能器件可供选用，本节着重介绍编码器和译码器的分类、工作原理及应用。



3.4.1 编码器

用二进制代码表示有关信息的过程称为**编码**。例如，在电子设备中，用二进制码表示字符，称为字符编码；用二进制码表示十进制数，称为二-十进制编码。能完成编码功能的电路称为**编码器**。编码器的通用逻辑符号如图所示，图中的X和Y分别表示输入和输出，在实际电路中可以用适当的符号代替。



- 编码器又分为**普通编码器**、**优先编码器**、**8421BCD编码器**等。在普通编码器中，任何时刻只允许一个输入信号有效，否则输出将发生混乱。在优先编码器中，对每一位输入都设置了优先权，因此允许两位以上的输入信号同时有效，但优先编码器只对优先级较高的输入进行编码，从而保证编码器工作的可靠性。



1. 二进制编码器

N 位输出可以表示 2^N 个输入信号，这种编码器称为 2^N 线— N 线编码器。

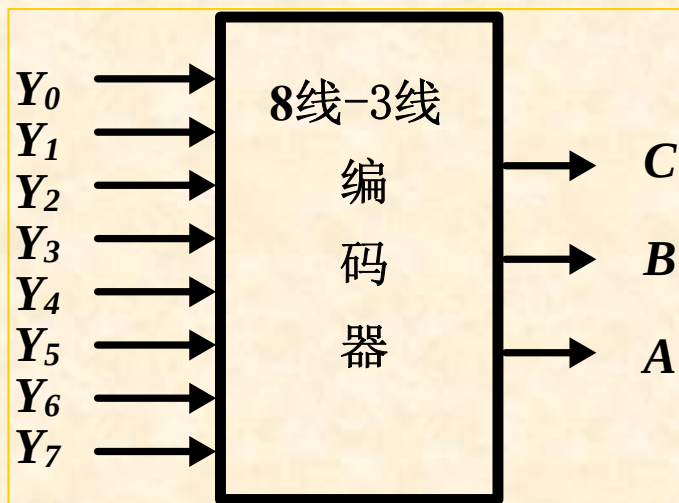
3位二进制编码器的原理框图，它有8个输入端 $Y_7—Y_0$ ，有3个输出端 C 、 B 、 A ，所以称为8线—3线编码器。

对于普通编码器来说，在任何时刻输入 $Y_7—Y_0$ 中只允许一个信号为有效电平。

编码器有高电平有效和低电平有效两种。

8线-3线编码器编码表

输入	C	B	A
Y_0	0	0	0
Y_1	0	0	1
Y_2	0	1	0
Y_3	0	1	1
Y_4	1	0	0
Y_5	1	0	1
Y_6	1	1	0
Y_7	1	1	1



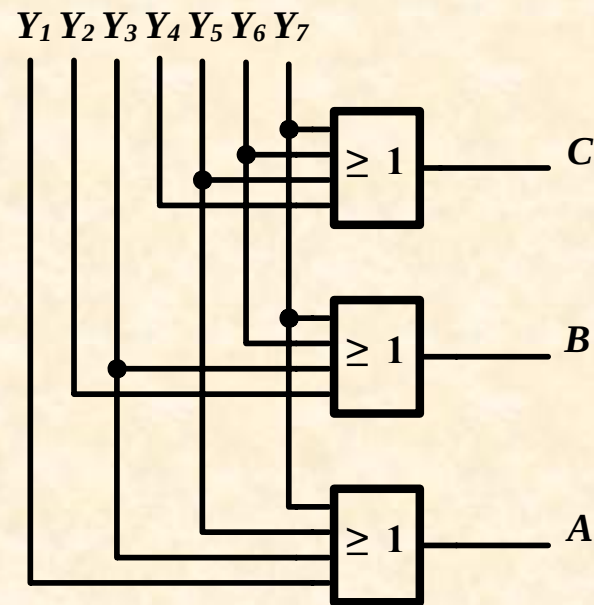
8线—3线编码器的框图





编码器的函数表达式和逻辑图

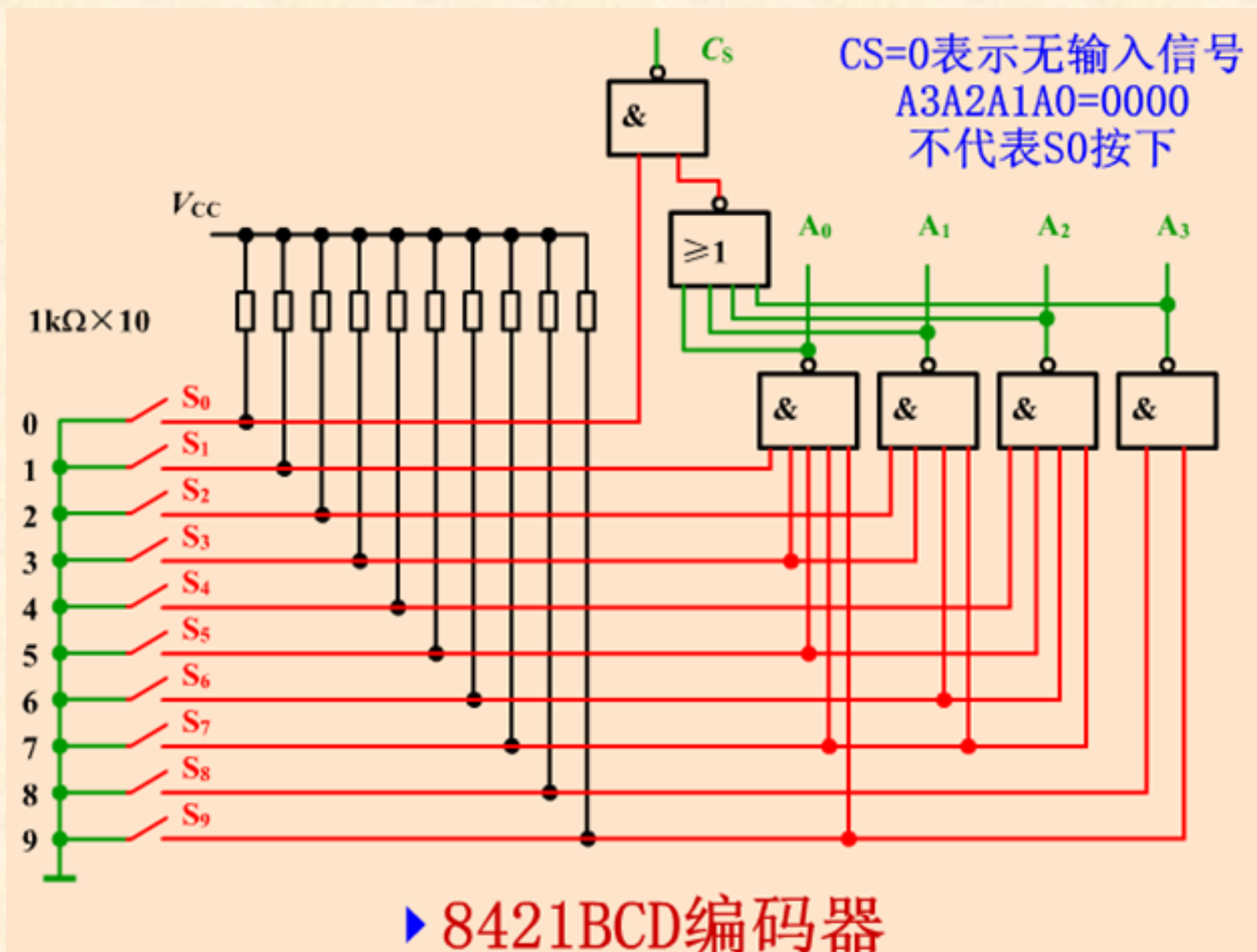
$$\left. \begin{aligned} C &= Y_4 + Y_5 + Y_6 + Y_7 \\ B &= Y_2 + Y_3 + Y_6 + Y_7 \\ A &= Y_1 + Y_3 + Y_5 + Y_7 \end{aligned} \right\}$$

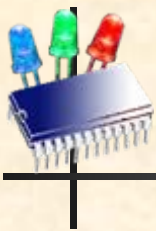


8线—3线普通编码器的逻辑图。

电路简单、方便，无法区分 Y_0 输入和无输入,也无法处理多个输入同时提出编码请求的情况。

将表示十进制数0~9的10个信号编成8421BCD码的电路。





二-十进制编码器功能表

自然数	输 入										输 出				
N	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0	A3	A2	A1	A0	CS
X	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1
2	1	1	1	1	1	1	1	0	1	1	0	0	1	0	1
3	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1
4	1	1	1	1	1	0	1	1	1	1	0	1	0	0	1
5	1	1	1	1	0	1	1	1	1	1	0	1	0	1	1
6	1	1	1	0	1	1	1	1	1	1	0	1	1	0	1
7	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1
8	1	0	1	1	1	1	1	1	1	1	1	0	0	0	1
9	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1





逻辑函数表达式

由功能表可得

$$A_3 = \overline{S_8} + \overline{S_9} = \overline{S_8 \cdot S_9}$$

$$A_2 = \overline{S_4} + \overline{S_5} + \overline{S_6} + \overline{S_7} = \overline{S_4 \cdot S_5 \cdot S_6 \cdot S_7}$$

$$A_1 = \overline{S_2} + \overline{S_3} + \overline{S_6} + \overline{S_7} = \overline{S_2 \cdot S_3 \cdot S_6 \cdot S_7}$$

$$A_0 = \overline{S_1} + \overline{S_3} + \overline{S_5} + \overline{S_7} + \overline{S_9} = \overline{S_1 \cdot S_3 \cdot S_5 \cdot S_7 \cdot S_9}$$

$$\begin{aligned} C_s &= \overline{S_0} + \overline{S_1} + \overline{S_2} + \overline{S_3} + \overline{S_4} + \overline{S_5} + \overline{S_6} + \overline{S_7} + \overline{S_8} + \overline{S_9} \\ &= \overline{S_0} + (\overline{S_8} + \overline{S_9}) + (\overline{S_4} + \overline{S_5} + \overline{S_6} + \overline{S_7}) + (\overline{S_2} + \overline{S_3} + \overline{S_6} + \overline{S_7}) + (\overline{S_1} + \overline{S_3} + \overline{S_5} + \overline{S_7} + \overline{S_9}) \\ &= \overline{S_0} + A_3 + A_2 + A_1 + A_0 = \overline{S_0(A_3 + A_2 + A_1 + A_0)} \end{aligned}$$

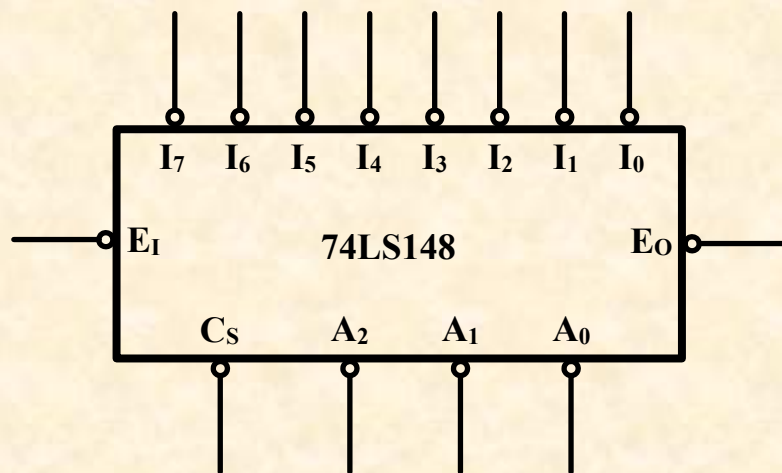
电路简单、方便，解决了S0输入和无输入的区别,但无法处理多个输入同时提出编码请求的情况。



3. 优先编码器

优先编码器电路中，允许两个以上的编码信号同时输入。当N个输入信号同时出现时，只对其中优先权最高的一个输入信号进行编码。这样就可以避免编码混乱的现象发生。

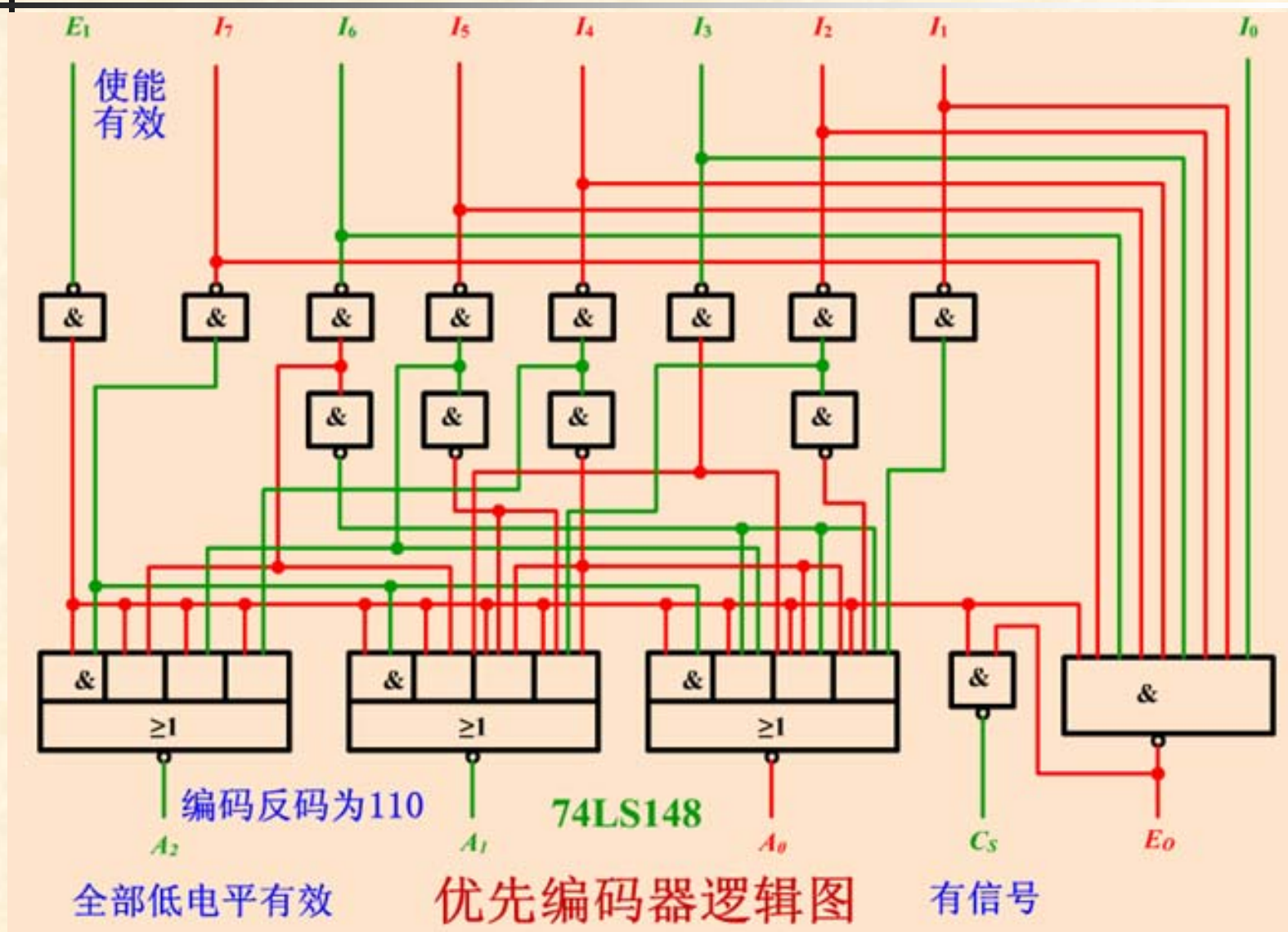
$I_0 \sim I_7$ 为输入端， $A_0 \sim A_2$ 为编码输出端， C_S 为工作状态标志表示有信号， E_0 为输出使能端表示无信号， C_S 、 E_0 主要用于级联和扩展。 E_1 为输入使能端。框图内所有变量均为正逻辑，框图外输入端的小圆圈表示输入信号低电平有效，输出端的小圆圈表示反码输出，可理解为逻辑非。



74LS148的逻辑符号



74LS148优先编码器的逻辑图



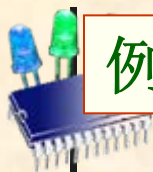


优先编码器74LS148的功能表

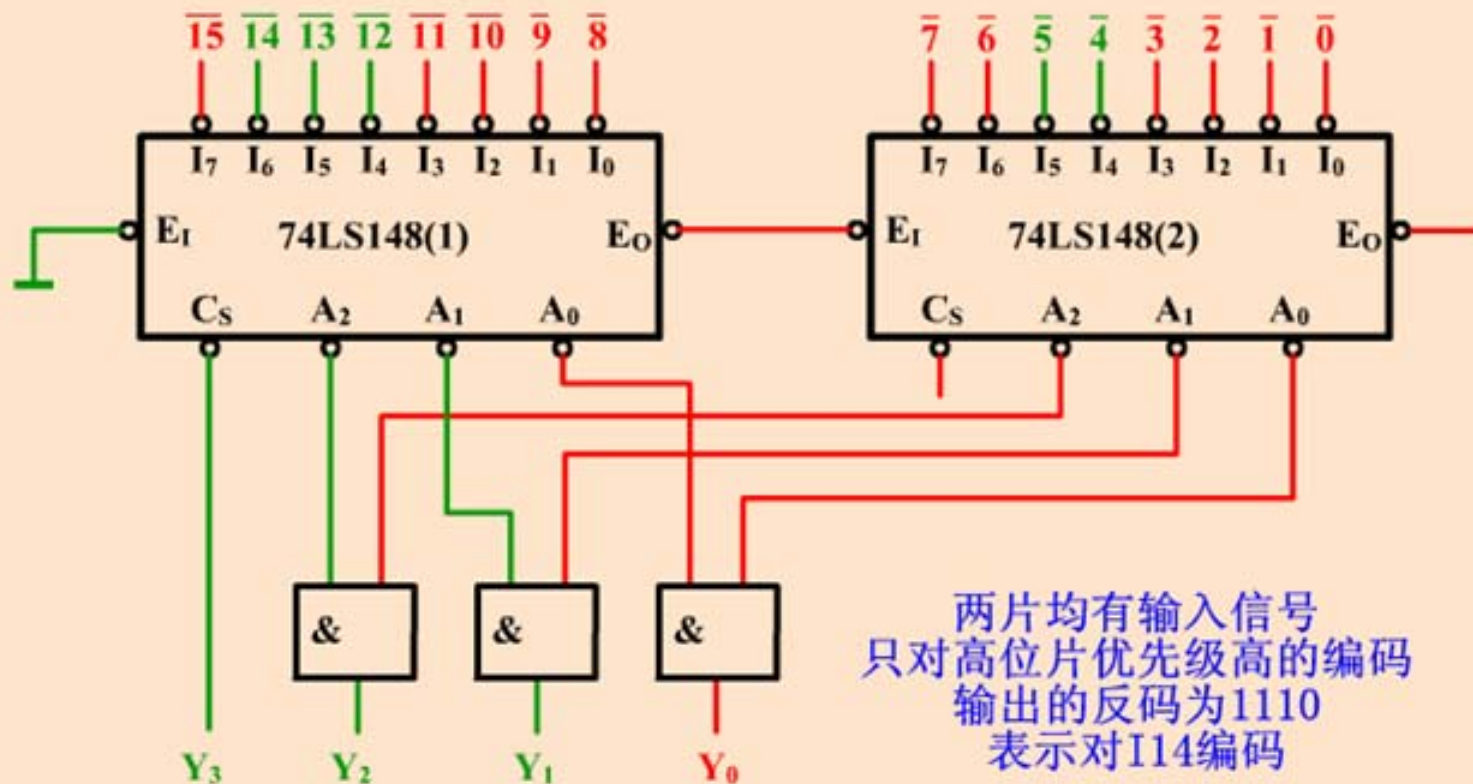
从74LS148的功能表可看出， E_1 为片选控制端，当它为有效电平0时，电路处于正常工作状态，几个输入信号中 I_7 的优先权最高， I_0 的优先权最低。 C_s 是“有编码输入”信号，当它有效时，表示电路处于工作状态且有输入编码信号。由于没有输入编码信号时 E_0 是“无编码输入”信号。

NO	输 入									输 出				
	E_1	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	A_2	A_1	A_0	C_s	E_0
1	1	x	x	x	x	x	x	x	x	1	1	1	1	1
2	0	1	1	1	1	1	1	1	1	1	1	1	1	0
3	0	0	x	x	x	x	x	x	x	0	0	0	0	1
4	0	1	0	x	x	x	x	x	x	0	0	1	0	1
5	0	1	1	0	x	x	x	x	x	0	1	0	0	1
6	0	1	1	1	0	x	x	x	x	0	1	1	0	1
7	0	1	1	1	1	0	x	x	x	1	0	0	0	1
8	0	1	1	1	1	1	0	x	x	1	0	1	0	1
9	0	1	1	1	1	1	1	0	x	1	1	0	0	1
10	0	1	1	1	1	1	1	1	0	1	1	1	0	1





例3.5 试用两片74LS148接成16-4线优先编码器。



► 8-3线优先编码器扩展



3.4.2 译码器

译码是编码的逆过程。它的功能是将具有特定含义的代码进行辨识。具有译码功能的逻辑电路称为译码器。也可以说译码器的实质是实现不同编码之间的转换，把一种编码用逻辑电路翻译成另一种编码。

译码器可以分为二进制译码器、二-十进制译码器、码制转换译码器和显示译码器。



1. 二进制译码器

当译码器电路不是用于代码转换，而是用于表示输入二进制数的状态，译码器的输出每次只有一个输出端为有效电平，其余输出端皆为无效电平时，称为惟一地址译码器，也称为基本译码器。例如，在计算机和其他数字系统中，每输入一个二进制代码，则会输出与此对应的一个存储地址，从而选中对应的存储单元。

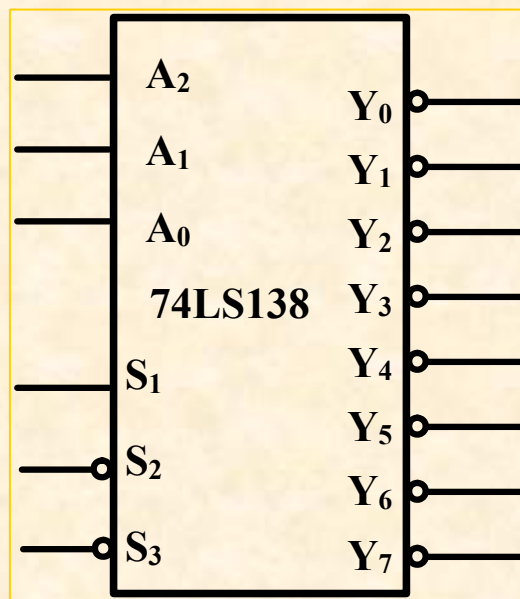
基本译码器有 n 个输入端(即 n 位二进制码)， 2^n 个输出线。

常见的中规模集成译码器有双2线-4线译码器74139、3线-8线译码器74138、4线-16线译码器74154等。



3线-8线译码器

74LS138有三个输入端A2、A1、A0，八个输出端Y0~Y7，故称为3线-8线译码器。三个使能控制端S3、S2、S1。只有控制输入端处于有效状态时，输入与输出之间才有相应的逻辑关系。





74LS138的功能表

由表可看出，当 $S_1=1$ 、 $(S_2+S_3)=0$ 时，译码器处于工作状态，此时，输出表达式为

控制输入		译码输入			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
x	1	x	x	x	1	1	1	1	1	1	1	1
0	x	x	x	x	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

$$Y_0 = \overline{A_2} \overline{A_1} \overline{A_0} = \overline{m_0}$$

$$Y_1 = \overline{A_2} \overline{A_1} A_0 = \overline{m_1}$$

$$Y_2 = \overline{A_2} A_1 \overline{A_0} = \overline{m_2}$$

$$Y_3 = \overline{A_2} A_1 A_0 = \overline{m_3}$$

$$Y_4 = A_2 \overline{A_1} \overline{A_0} = \overline{m_4}$$

$$Y_5 = A_2 \overline{A_1} A_0 = \overline{m_5}$$

$$Y_6 = A_2 A_1 \overline{A_0} = \overline{m_6}$$

$$Y_7 = A_2 A_1 A_0 = \overline{m_7}$$

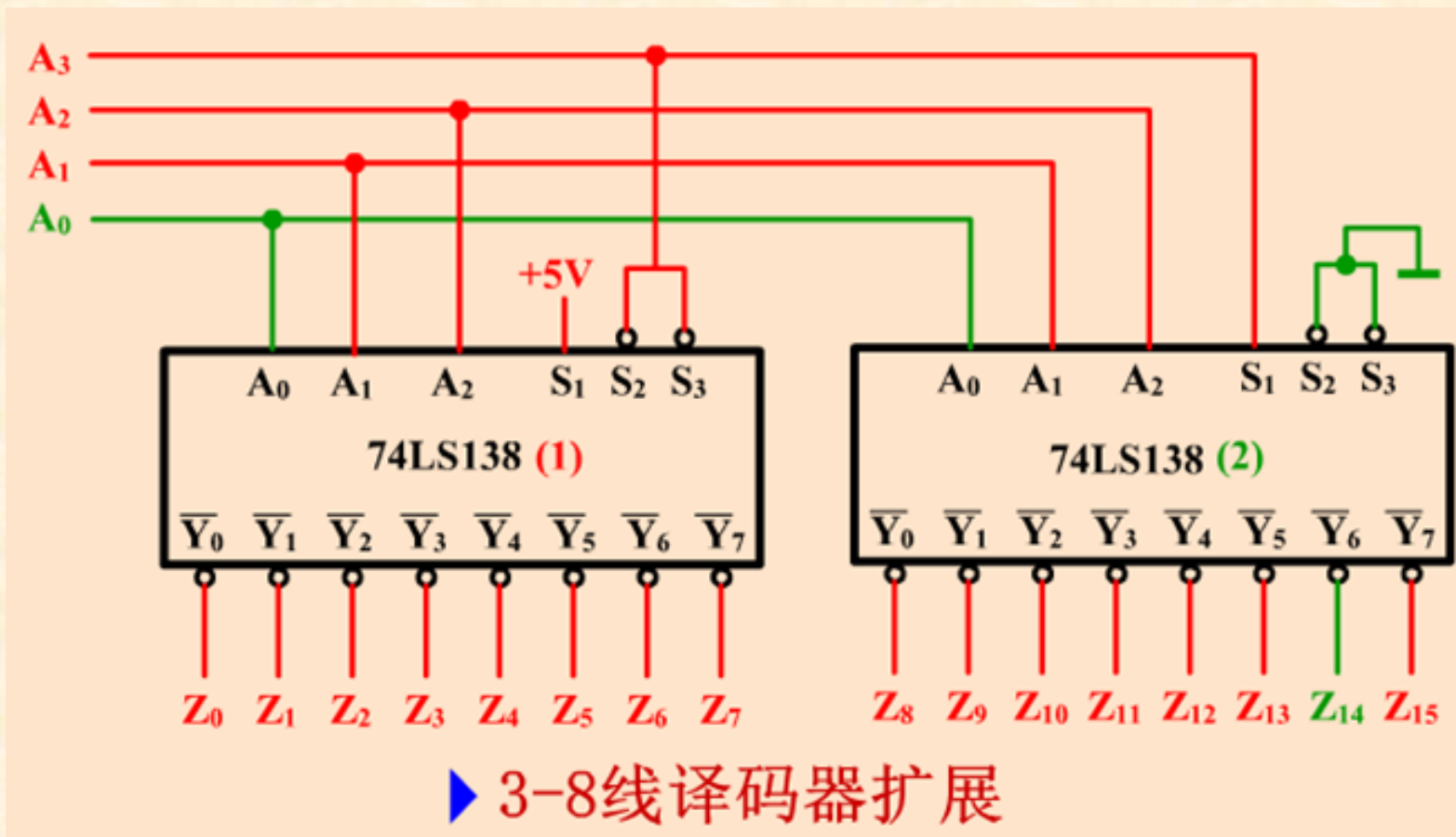
合理地应用使能端 S_1 、 S_2 和 S_3 可以实现74LS138译码器的扩展。将 A 看作变量，低电平有效的 Y_i 是最小项的反。



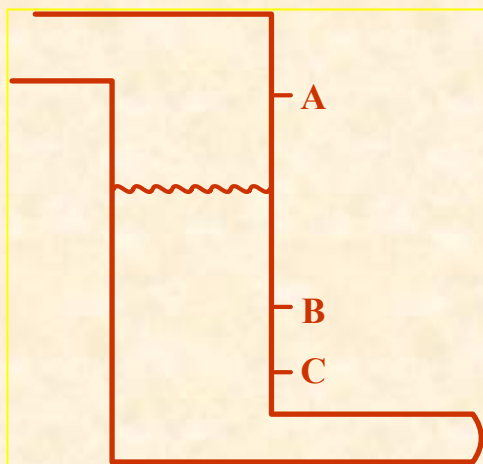


2. 译码器应用举例

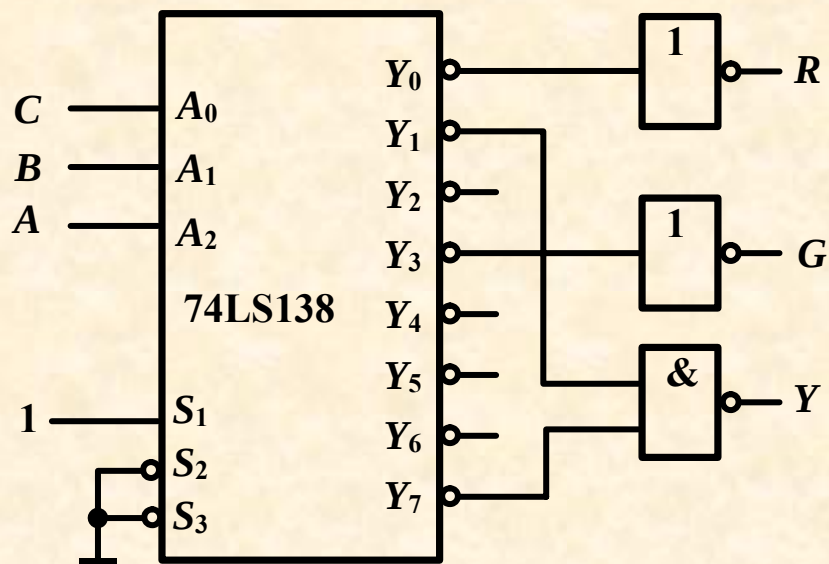
合理地应用使能端 **S3**、**S2**、**S1** 实现两片 3 线-8 线译码器扩展为 4 线-16 线译码器。



例3.8 图为一个加热水容器的示意图。图中**A**、**B**、**C**为水位传感器。当水面在**AB**之间时，为正常状态，绿灯**G**亮；当水面在**BC**之间或者在**A**以上时，为异常状态，黄灯**Y**亮；当水面在**C**以下时，为危险状态，红灯**R**亮。现要求用中规模集成译码器设计一个按上述要求控制三种灯亮或暗的逻辑电路。



输 入			输 出		
A	B	C	G	Y	R
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	x	x	x
0	1	1	1	0	0
1	0	0	x	x	x
1	0	1	x	x	x
1	1	0	x	x	x
1	1	1	0	1	0



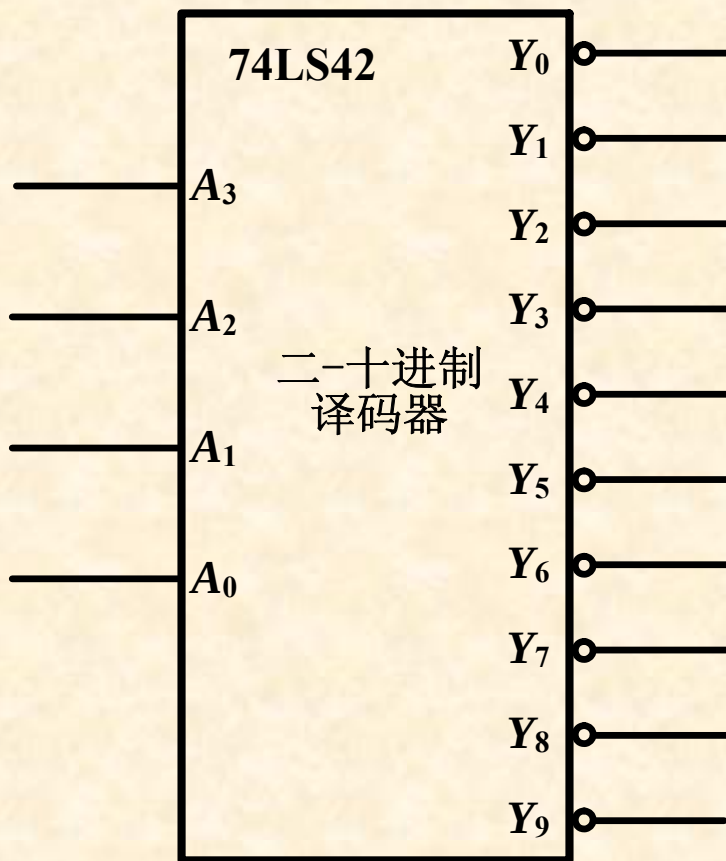
$$G = \overline{A}BC = \overline{Y}_3$$

基本译码器实际上是一个最小项发生器，利用译码器和门电路可以构成各种多变量逻辑函数发生器，产生各种逻辑函数。



3.二-十进制译码器

二-十进制译码器也称**BCD**译码器，它的逻辑功能是将输入的一位**BCD**码译成十个高低电平输出信号。





二-十进制译码器功能表

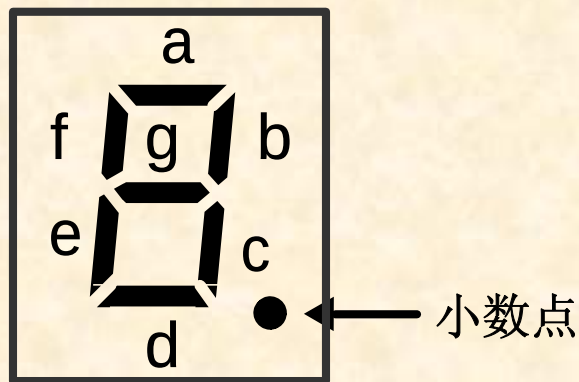
序号	输 入				输 出							
	A ₃	A ₂	A ₁	A ₀	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1
5	0	1	0	1	1	1	1	1	0	1	1	1
6	0	1	1	0	1	1	1	1	1	0	1	1
7	0	1	1	1	1	1	1	1	1	1	0	1
8	1	0	0	0	1	1	1	1	1	1	1	0
9	1	0	0	1	1	1	1	1	1	1	1	0





4.显示译码器

在一些数字系统中，不仅需要译码，而且需要把译码的结果显示出来。用显示译码器驱动显示器件，就可以达到数据显示的目的，目前广泛使用的显示器件是7段数码显示器。7段数码显示器由a—g等7段可发光的线段拼合而成，控制各段的亮或灭，就可以显示不同的字符或数字。7段数码显示器有半导体数码显示器和液晶显示器两种。

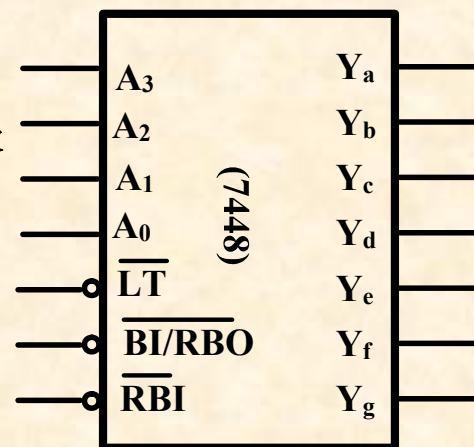
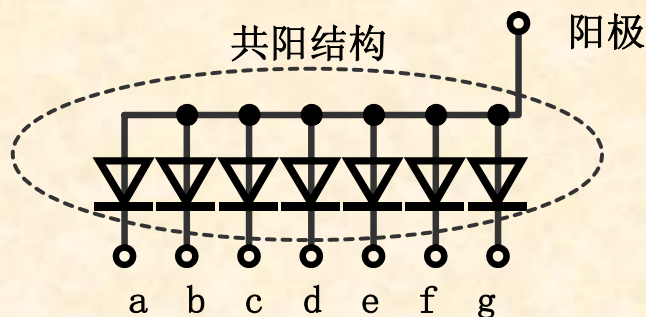
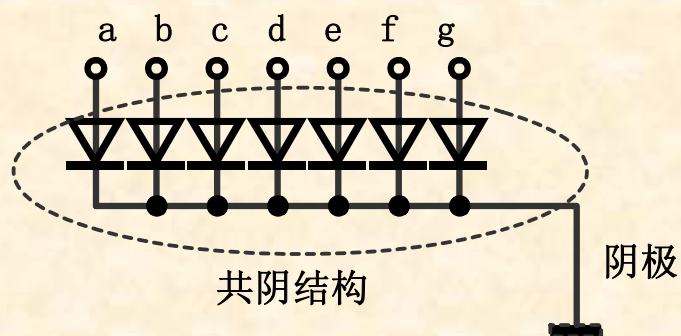


7段数码显示器



两种数码显示器

半导体7段数码管BS201A的每个段都是一个发光二极管LED(Light Emitting Diode)。当LED加上正向电压时，可以发出橙红色的光。有的数码管在右下角还增设了一个小数点，形成8段显示。构成数码管的7只LED的阴极是连接在一起的，属于共阴结构。如果把7只LED的阳极连接在一起，则属于共阳结构。



另一种7段数码显示器是液晶显示器LCD(Light Crystal Display)。液晶显示器中的液态晶体材料是一种有机化合物，在常温下既有液体特性，又有晶体特性。利用液晶在电场作用下产生光的散射或偏光作用原理，便可实现数字显示。一般对LCD的驱动采用正负对称的交流信号。





7448的功能表

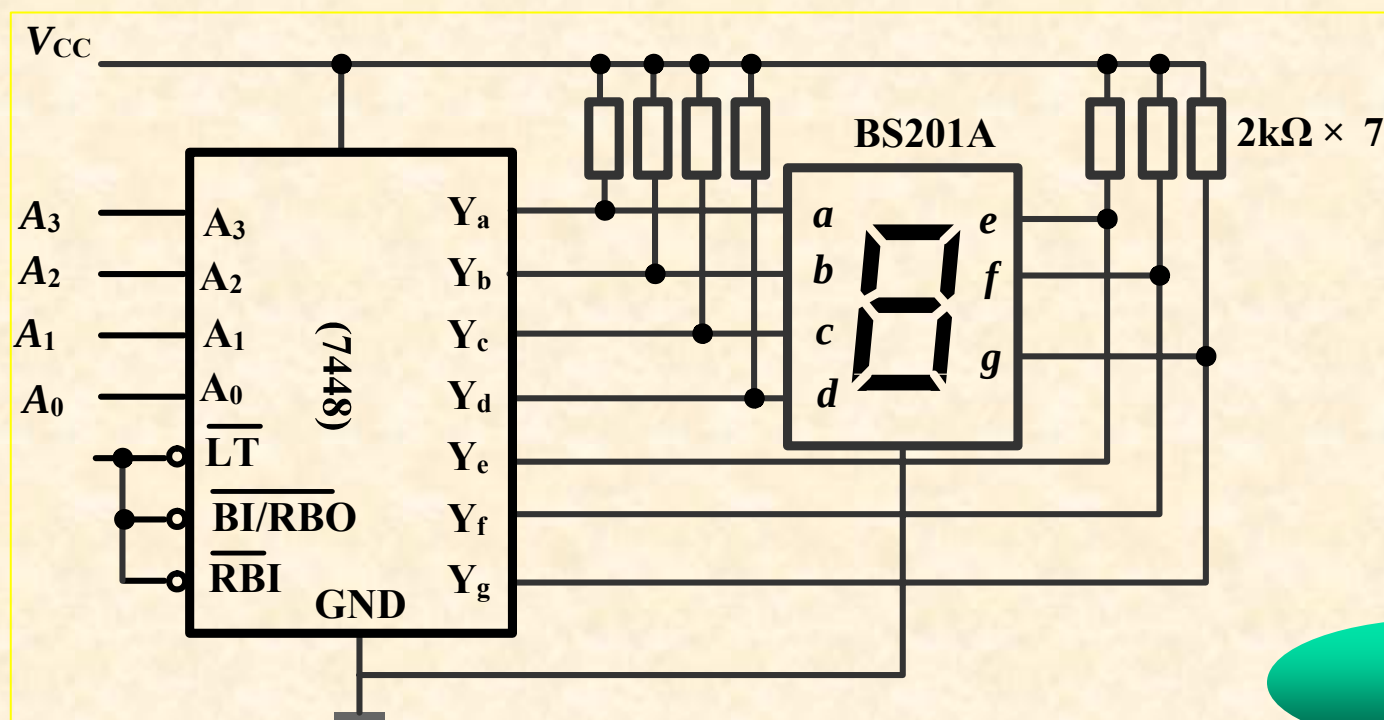
7448 除了完成译码驱动的功能外,还附加了灯测试输入 $\overline{LT}$ 、消隐输入 $\overline{BI}$ 、灭零输入 $\overline{RBI}$ 和灭零输出 $\overline{RBO}$ 等控制信号。

输 入							输 出								字形
数字	$\overline{LT}$	$\overline{RBI}$	A_3	A_2	A_1	A_0	$\overline{BI} / \overline{RBO}$	Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g	
0	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0
1	1	X	0	0	0	1	1	0	1	1	0	0	0	0	1
2	1	X	0	0	1	0	1	1	1	0	1	1	0	1	2
3	1	X	0	0	1	1	1	1	1	1	1	0	0	1	3
4	1	X	0	1	0	0	1	0	1	1	0	0	1	1	4
5	1	X	0	1	0	1	1	1	0	1	1	0	1	1	5
6	1	X	0	1	1	0	1	0	0	1	1	1	1	1	6
7	1	X	0	1	1	1	1	1	1	1	0	0	0	0	7
8	1	X	1	0	0	0	1	1	1	1	1	1	1	1	8
9	1	X	1	0	0	1	1	1	1	1	0	0	1	1	9
消 隐	X	X	X	X	X	X	0	0	0	0	0	0	0	0	8
脉冲消隐	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
试 灯	0	X	X	X	X	X	1	1	1	1	1	1	1	1	



7448实用电路

用7448直接驱动共阴结构的半导体数码管的电路连接如图示。由于7448的输出是集电极开路门结构，需要外加 $2k\Omega$ 的上拉电阻



EXE
链接

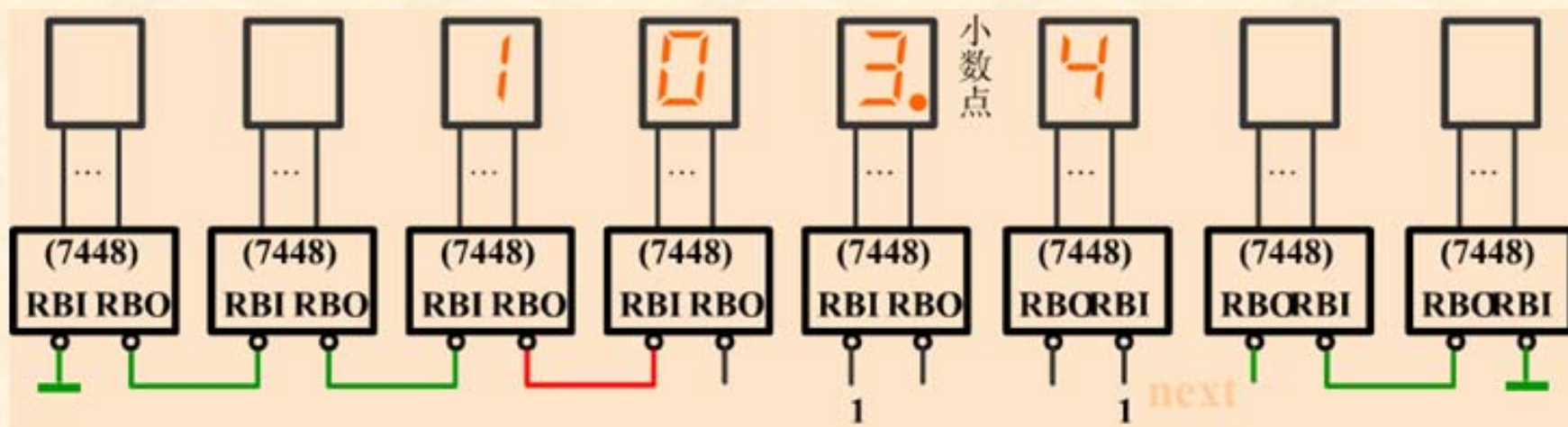
用7448驱动BS201A的连接图





具有灭零控制的实用电路

用7448实现8位十进制数码显示系统的连接如图示，系统具有灭零控制。在图中，只需要把整数部分的高位与低位的相连，小数部分的低位与高位的相连，就可以把前、后多余的0灭掉。在这种连接方式下，整数部分只有高位是0，而且被熄灭的情况下，低位才有灭零输入信号。同理，小数部分只有低位是0，而且被熄灭的情况下，高位才有灭零输入信号。



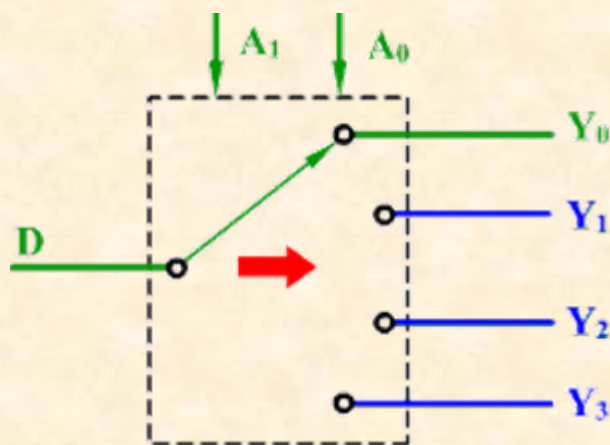
有灭零控制的8位数码显示系统



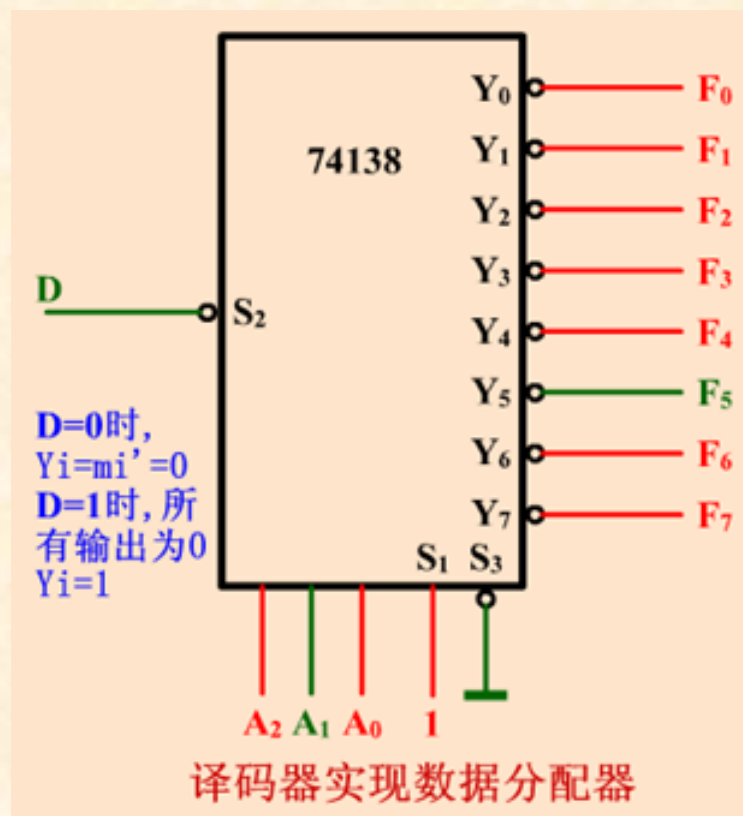
3.5 数据分配器与数据选择器

1. 数据分配器

把一条通道上的数据分配到不同的数据通道上，完成这一功能的电路称为数据分配器(也称多路数据分配器，多路数据调节器)



数据分配器

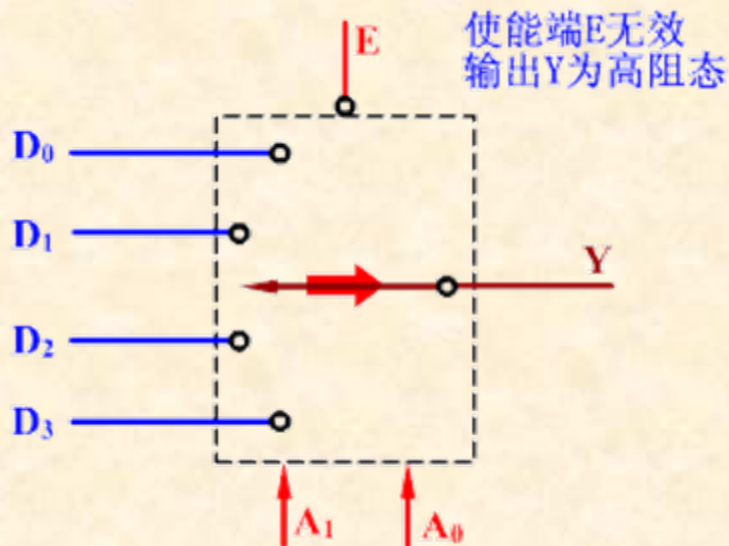


译码器实现数据分配器



3.5.2 数据选择器

在地址信号控制下,从多路数据选出其中的一路数据作为输出的过程叫做数据选择,具有数据选择功能的电路称为数据选择器。常用的有双4选1(74LS153、74LS253), 8选1(74LS152, 74LS153)和16选1(74LS150, 74850, 74851)等数据选择器产品。



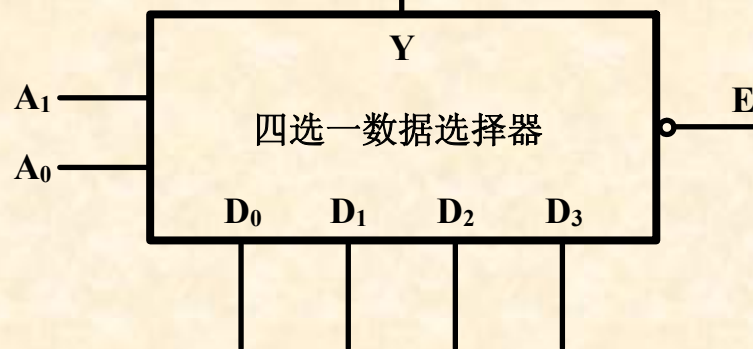
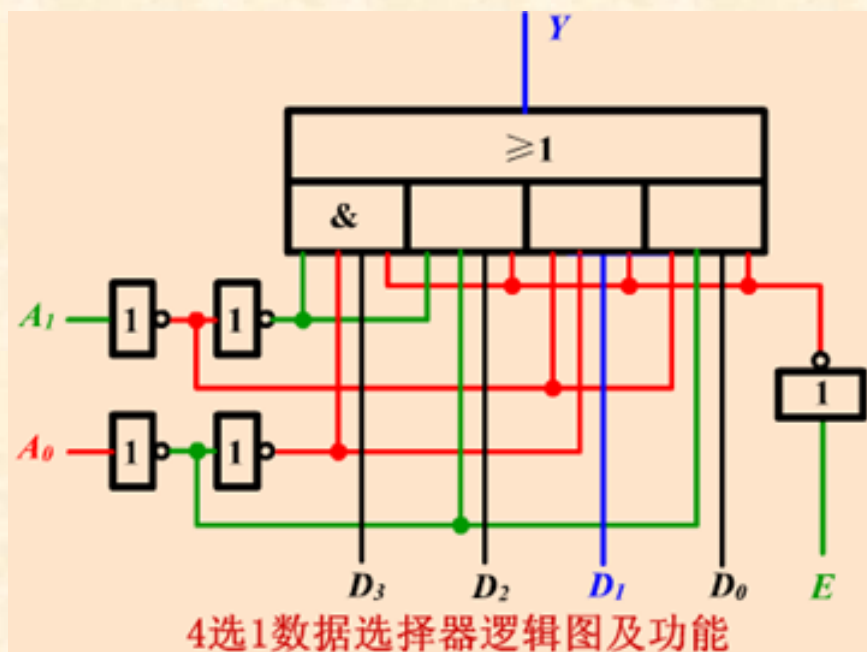
4选1数据选择器等效图



1.数据选择器的功能描述

4选1数据选择器的逻辑图和逻辑符号如图示。图中， A_1A_0 是选择控制信号(也称为地址信号)， D_3-D_0 是数据输入端， Y 是数据输出端， E 是使能控制端。由逻辑图可以得到其输出表达式为

$$Y = \overline{E}(\overline{A_1}\overline{A_0}D_0 + \overline{A_1}A_0D_1 + A_1\overline{A_0}D_2 + A_1A_0D_3) = \overline{E}\sum_{i=0}^3 m_i D_i$$



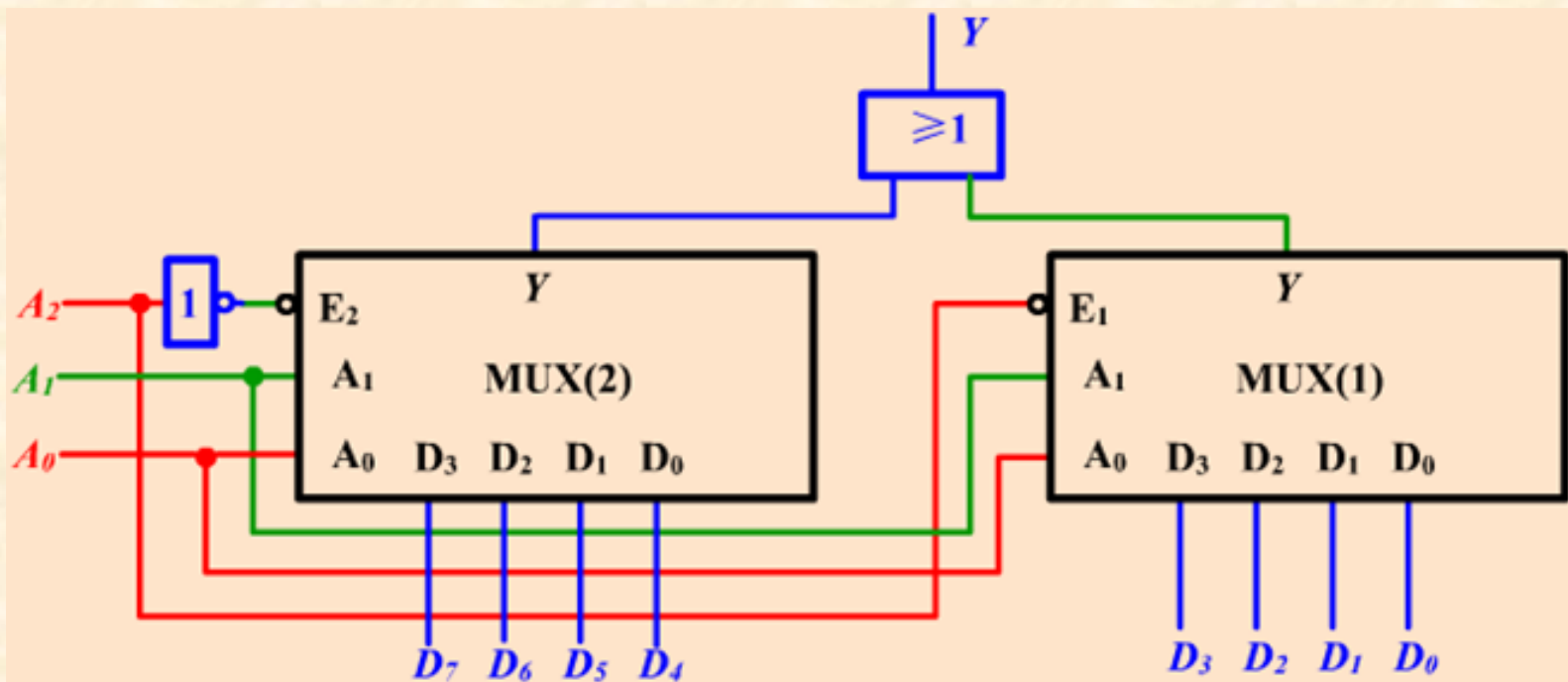
2ⁿ选一数据选择器输出表达式为

$$Y = \overline{E}\sum_{i=0}^{2^n-1} m_i D_i$$



2. 数据选择器功能的扩展

如需要选择的数据通道较多时，可以选用八选一或十六选一数据选择器，也可以把几个数据选择器级联的方法来扩展输入端。



▶ 两片4选1数据选择器扩展为8选1数据选择器



3.数据选择器的应用

数据选择器的应用很广泛，它不仅可以实现有选择的传递数据，

还可以作为逻辑函数发生器，产生逻辑函数，也可以将并行数据转换为串行数据进行传输。



例3.9 用八选一数据选择器实现逻辑函数

$$L = \overline{A}BC + AB\overline{C}$$

解:配项为最小项表达式

$$L(A,B,C) = \sum m(0,2,4,5,6,7)$$

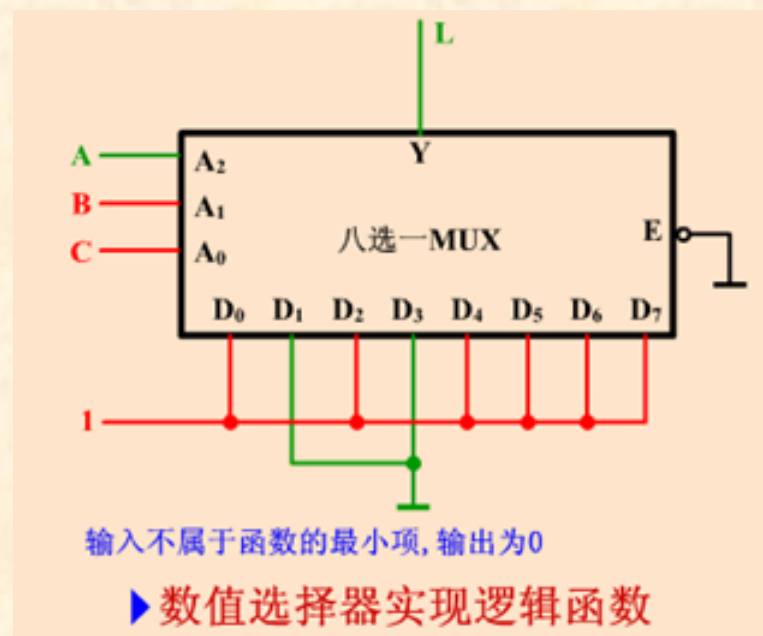
与8-1MUX表达式对照

$$Y(A_2, A_1, A_0) = \sum_{i=0}^7 m_i D_i$$

要使两式相等,必须使

$$D_0, D_2, D_4, D_5, D_6, D_7 = 1$$

$$D_1, D_3 = 0$$



当函数变量数大于控制端数时就要将多出的变量分配到 D_i ,
即 D_i 是多余变量的函数



例3.10

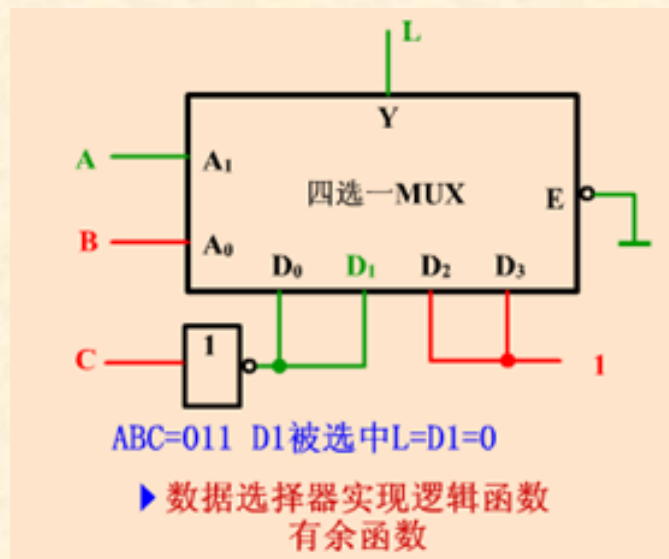
试用四选一数据选择器来实现例3.9中给出的逻辑函数： $L = A\bar{B}C + AB + \bar{C}$

解：将函数配项成最小项表达式并整理成与4选1MUX表达式

$$Y = D_0 \bar{A}_1 \bar{A}_0 + D_1 \bar{A}_1 A_0 + D_2 A_1 \bar{A}_0 + D_3 A_1 A_0$$

相应的形式

$$\begin{aligned} L &= \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} B \bar{C} + A \bar{B} \cdot \bar{C} + A \bar{B} C + A B \bar{C} + A B C \\ &= \bar{C} \cdot \bar{A} \bar{B} + \bar{C} \cdot \bar{A} B + (\bar{C} + C) \cdot A \bar{B} + (\bar{C} + C) \cdot A B \end{aligned}$$





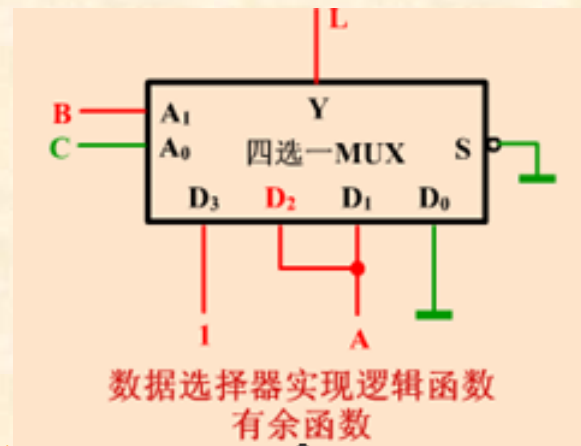
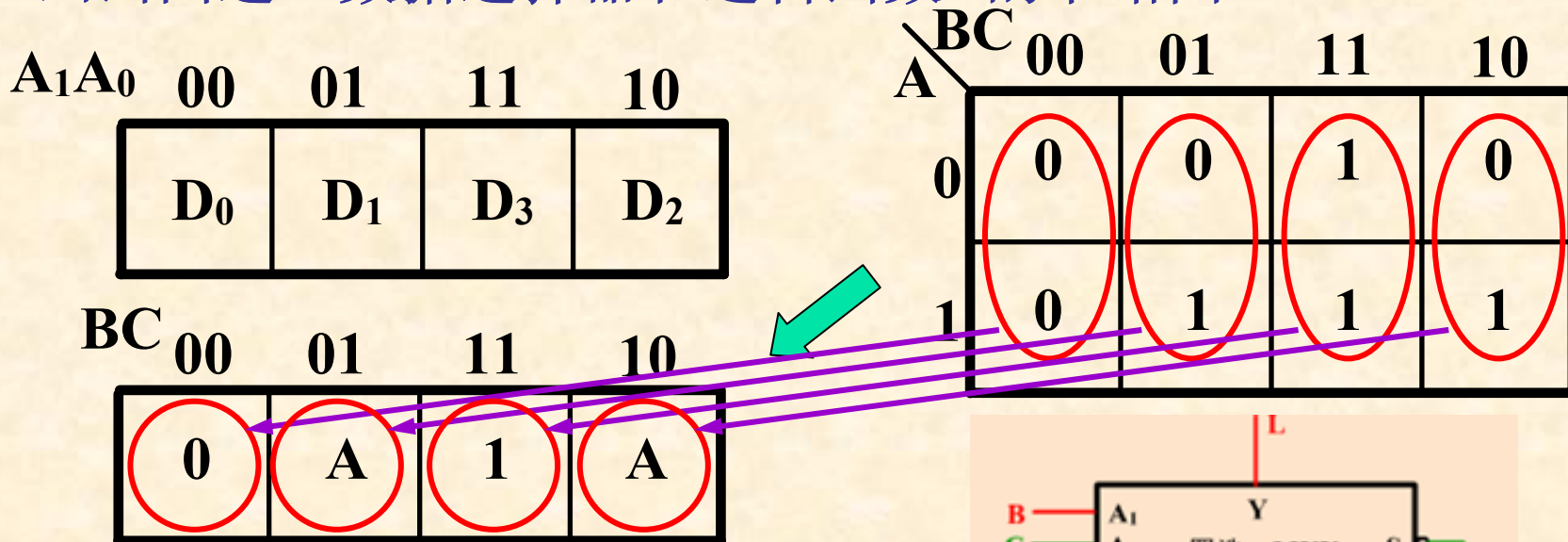
例3.11

试用四选一数据选择器实现三变量多数表决器

解：三变量多数表决器的逻辑表达式为：

$$L = \overline{A}BC + A\overline{B}C + ABC\overline{C} + ABC$$

画出四选一数据选择器和逻辑函数L的卡诺图

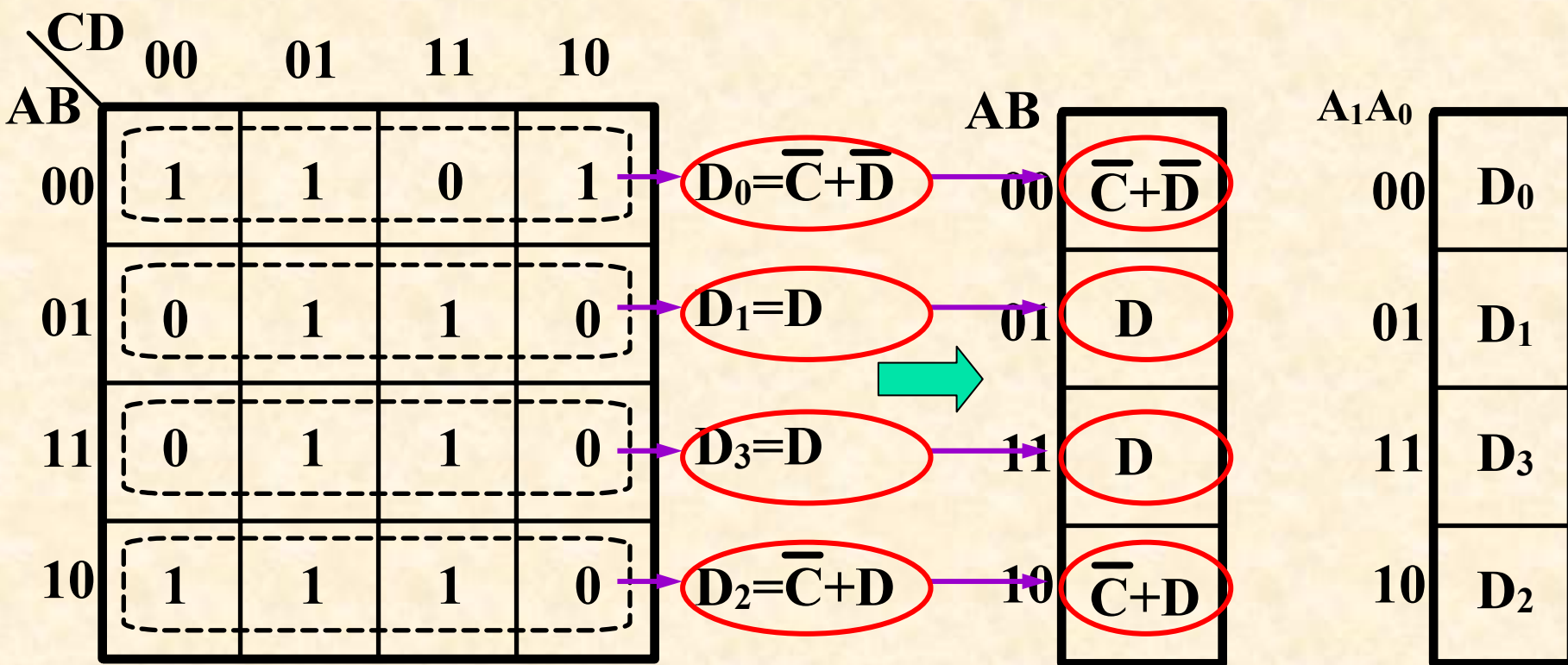




例3.12

试用四选一数据选择器实现如下函数
$$L = \sum m(0, 1, 2, 5, 7, 8, 9, 11, 13, 15)$$

解:画出四选一数据选择器和逻辑函数L的卡诺图



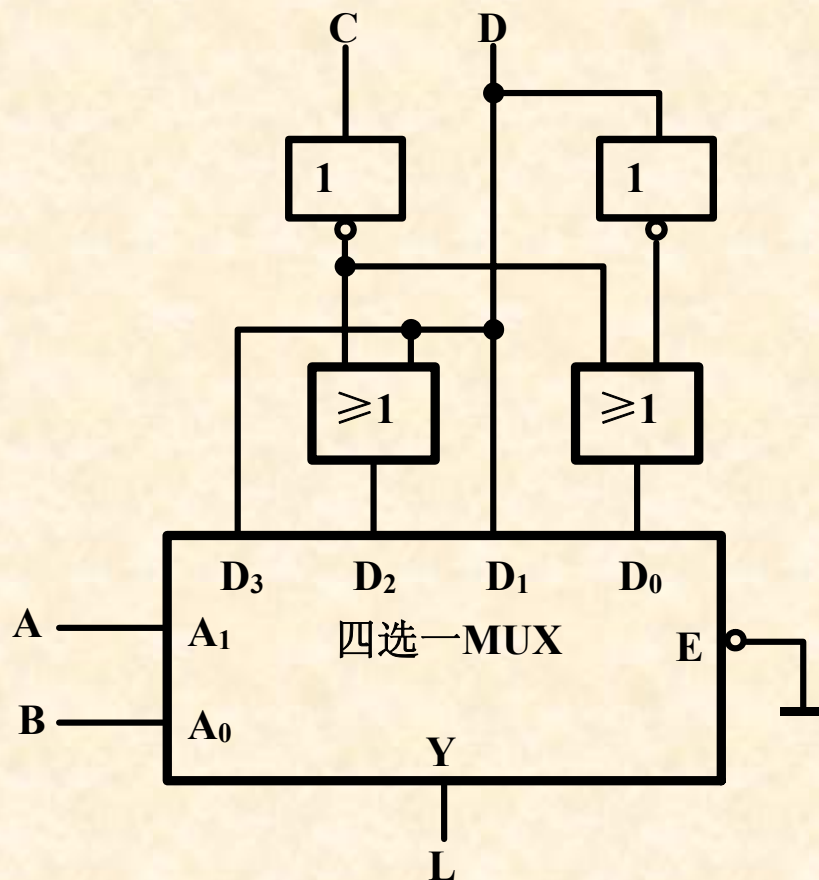


电路连接

对照两个卡诺图画出电路连接

A_1A_0	
00	D_0
01	D_1
11	D_3
10	D_2

AB	
00	$\bar{C} + \bar{D}$
01	D
11	D
10	$\bar{C} + D$





3.6 算术运算电路

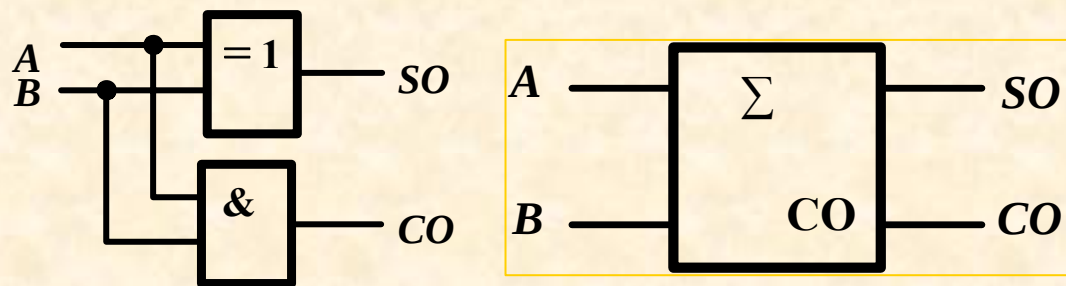
算术运算电路是能够完成二进制数运算的器件，**半加器**和**全加器**是算术运算电路的基本单元电路。



3.6.1 加法器

1. 半加器和全加器

在半加器的电路结构和逻辑符号图中，A、B是两个1位二进制加数的输入端，SO是两个数相加后的和，CO是向高位的进位。写出电路输出的逻辑表达式



A	B	SO	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

半加器电路结构及符号

$$SO = A \oplus B = \overline{A}B + A\overline{B}$$

$$CO = AB$$

根据输出表达式推导出电路的真值表。真值表说明，电路能完成两个1位二进制数的加法运算。这种不考虑来自低位的进位的加法运算，称为半加，能实现半加运算的电路称为半加器。



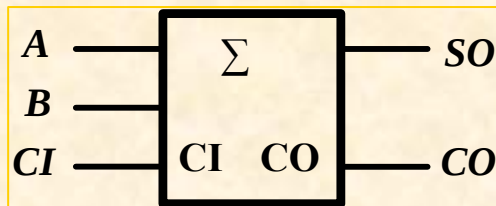
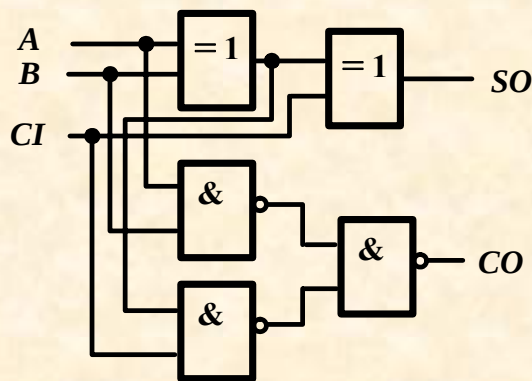


2) 全加器

全加器的电路结构和逻辑符号图中，比半加器多一个输入端 C_I 是低位来的进位。写出电路输出端的逻辑表达式为

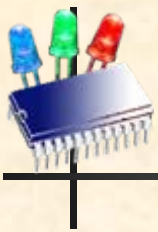
$$SO = A \oplus B \oplus CI = \overline{A}\overline{B} \cdot \overline{CI} + \overline{A}B\overline{CI} + \overline{A} \cdot BCI + ABCI$$

$$CO = \overline{\overline{AB} \cdot \overline{(A \oplus B)CI}} = AB + BCI + ACI$$



A	B	CI	SO	CO
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

导出电路的真值表。真值表说明，电路能完成两个1位二进制数以及低位来的进位的加法运算。这种考虑来自低位来的进位的加法运算，称为全加，能实现全加运算的电路称为全加器。



2. 多位二进制加法器

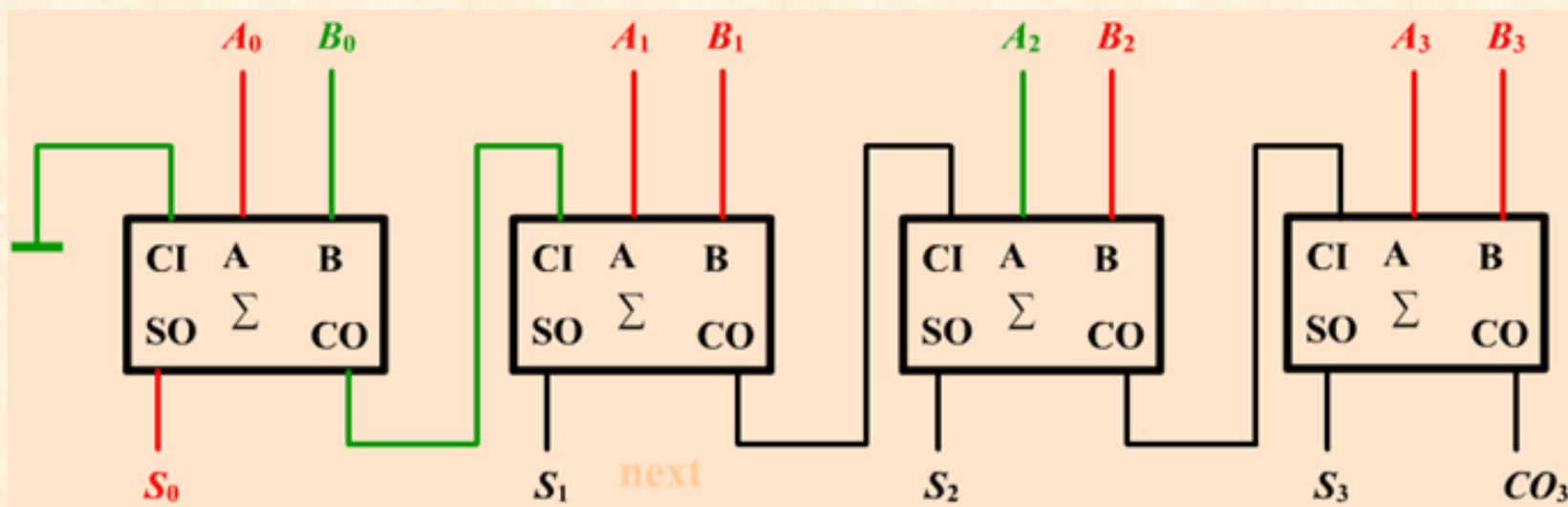
1位全加器可以实现1位二进制数的加法运算，
把N位全加器组合起来，可以实现N位二进制
数的加法器。

在构成多位加法器电路时，按照进位方式的不
同，又有串行进位加法器和超前进位加法器
两种类型。



1) 串行进位加法器电路

加法从低位开始，每一位的相加结果都必须等到低一位的进位产生以后才能建立起来，把这种结构的电路叫做串行进位加法计数器。

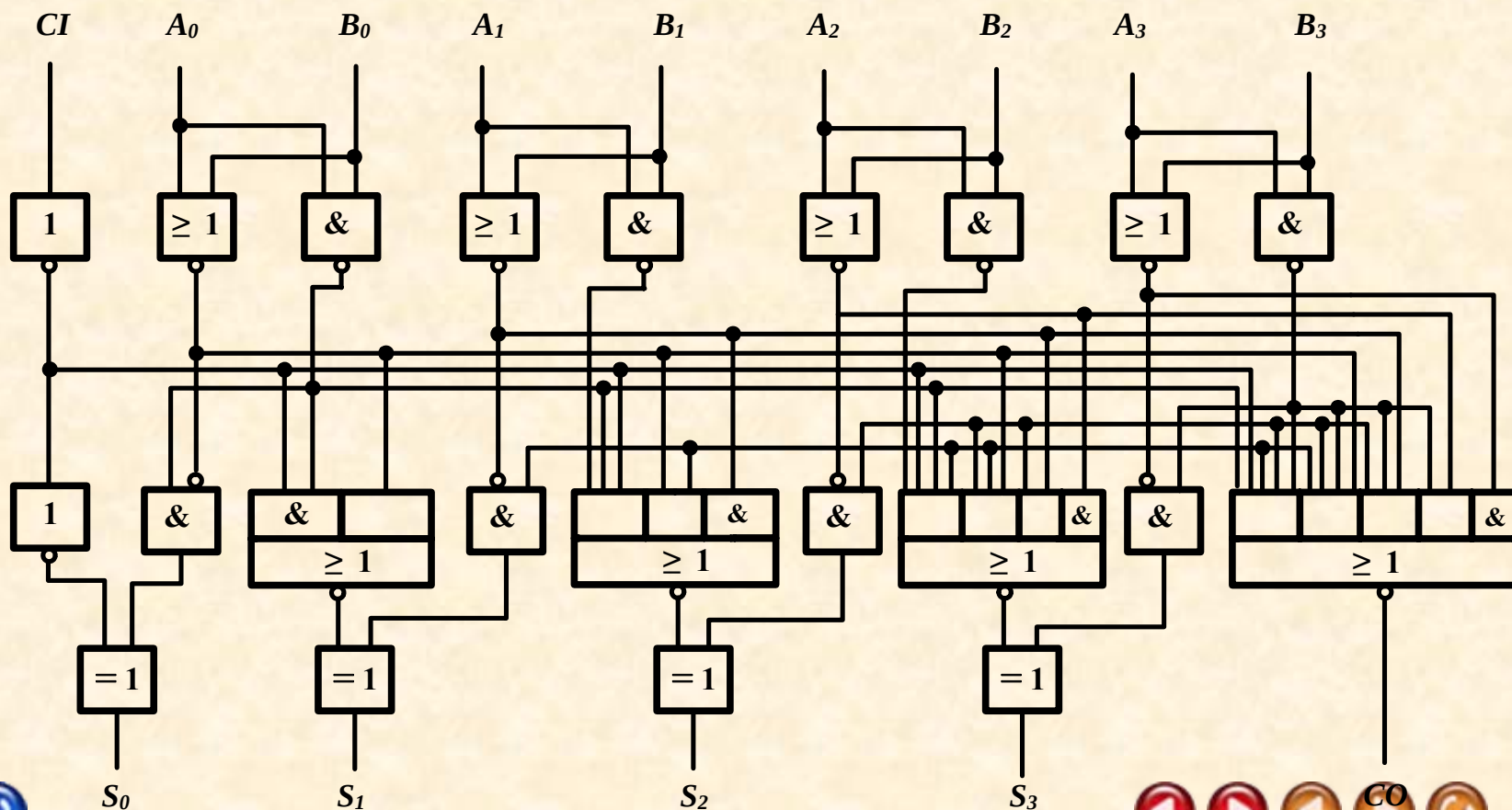


串行进位加法器的优点是电路比较简单，缺点是运算速度不高。为了提高运算速度，可以采用超前进位加法器。



2)超前进位并行加法器

在超前进位加法器电路中，每一位的进位只由加数决定，而与低位的进位无关。这样，多位加法器的加法运算就可同时进行，而不必等到低位的进位到来后才进行，因此提高了运算速度。





例:设计一个2位超前进位加法器

列出真值表

卡诺图
16-18

写出最简与或式

A ₀	B ₀	CO ₀	A ₁	B ₁	CO ₁	S ₀	S ₁
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	0
0	0	0	1	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	1	0	1	1
0	1	0	1	0	0	1	1
0	1	0	1	1	1	1	0
1	0	0	0	0	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	0	0	1	1
1	0	0	1	1	1	1	0
1	1	1	0	0	0	0	1
1	1	1	0	1	1	0	0
1	1	1	1	0	1	0	0
1	1	1	1	1	1	0	1

$$CO_1 = A_1 B_1 + A_0 B_0 B_1 + A_0 B_0 A_1$$

$$S_0 = \overline{A_0} B_0 + A_0 \overline{B_0}$$

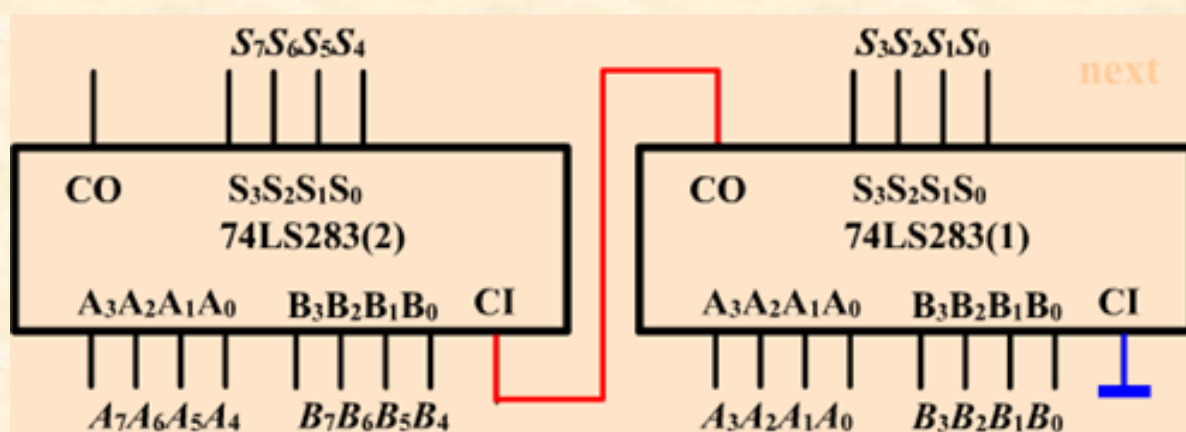
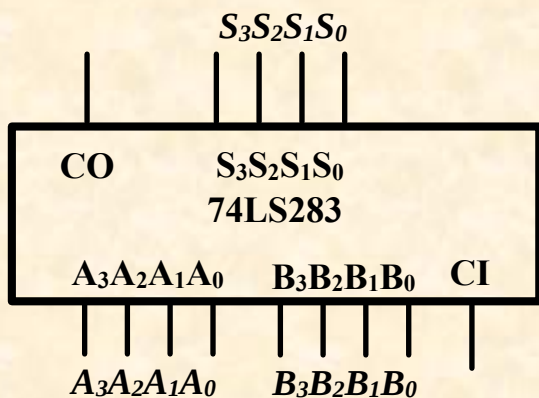
$$\begin{aligned} S_1 = & \overline{A_0} \overline{A_1} B_1 + \overline{B_0} \overline{A_1} B_1 \\ & + \overline{A_0} B_0 A_1 \overline{B_1} + A_0 B_0 \overline{A_1} \overline{B_1} \\ & + A_0 B_0 A_1 B_1 + A_0 \overline{B_0} A_1 \overline{B_1} \end{aligned}$$

画电路图略



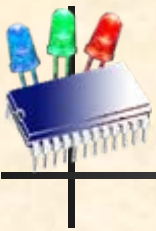
扩展功能

把两片74LS283级联起来，扩展成8位加法器电路如图示



74LS283的逻辑号

用74LS283构成的8位加法器



3.6.2 二进制减法运算

在数字系统中，为了简化系统结构，通常不另外设计减法器，而是将减法运算变为加法运算来处理，使运算器既能够实现加法运算，又可以实现减法运算。

二进制数减法运算用补码的加法运算替代。



1.原码表示法

正负数的表示:如果在二进制数的最高位前增加一位符号位,就可以区分正二进制数和负二进制数。设正数的符号位为0,负数的符号位为1,其余各位表示数的绝对值。

例如:两个数+3和-3分别用二进制数A、B表示,即: $A=+0011$, $B=-0011$, 则可写为:

$$[+0011]_{\text{原}} = \overset{\text{符号位}}{0} \overset{\text{绝对值}}{0011}$$

$$[-0011]_{\text{原}} = 10011$$



2.补码表示法

正数的补码与原码相同,负数求补码方法如下

补码与原码的关系 $N_{\text{补}} = 2^n - N_{\text{原}}$ (n为N的位数)

例如: $(-13)_D = (-1101)_B$

绝对值补码为 $[1101]_{\text{补}} = 2^4 - 1101 = 0011$

加上符号位后得 $[-1101]_{\text{补}} = 10011$

求补过程

$$\begin{array}{rcccccc} & 1 & 1 & 1 & 0 & 1 & (N_{\text{原}}) \\ & 1 & 0 & 0 & 1 & 0 & (\text{对 } N_{\text{原}} \text{ 求反}) \\ + & & & & & 1 & (+1) \\ \hline \end{array}$$

$$1 \ 0 \ 0 \ 1 \ 1 \quad (N_{\text{补}})$$

负数保持符号位不变, 绝对值求反加1。



3.用补码完成减法运算

A、B两个数，减法运算转换为加法运算

$$A - B = A + (-B)_{\text{补码}}$$

例3.13 求9-3和3-9的值

解：先求9-3的值：

二进制补码加法

$$\begin{array}{r} \quad \quad \quad \textcolor{red}{0}\textcolor{blue}{1001} \quad \quad \quad [+9]_{\text{补}} \\ + \quad \quad \quad \textcolor{red}{1}\textcolor{blue}{1101} \quad \quad \quad [-3]_{\text{补}} \\ \hline \textcolor{green}{1}\textcolor{red}{0}\textcolor{blue}{0110} \quad \quad \quad [+6]_{\text{补}} = [+6]_{\text{原}} \\ \textcolor{red}{\uparrow} \quad \quad \quad \textcolor{green}{溢出舍去} \end{array}$$

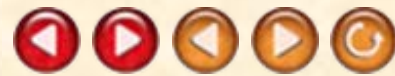


求3-9的值:

$$\begin{array}{rcl} & 00011 & [+3]_{\text{补}} \\ + & 10111 & [-9]_{\text{补}} \\ \hline & 11010 & [-6]_{\text{补}} \\ & 10110 & [-6]_{\text{原}} \quad \text{求反加1} \end{array}$$

补码加法注意:

符号位参与运算;结果有溢出舍去。运算结果为正数,补码即为原码。运算结果为负数再次求补得到原码。





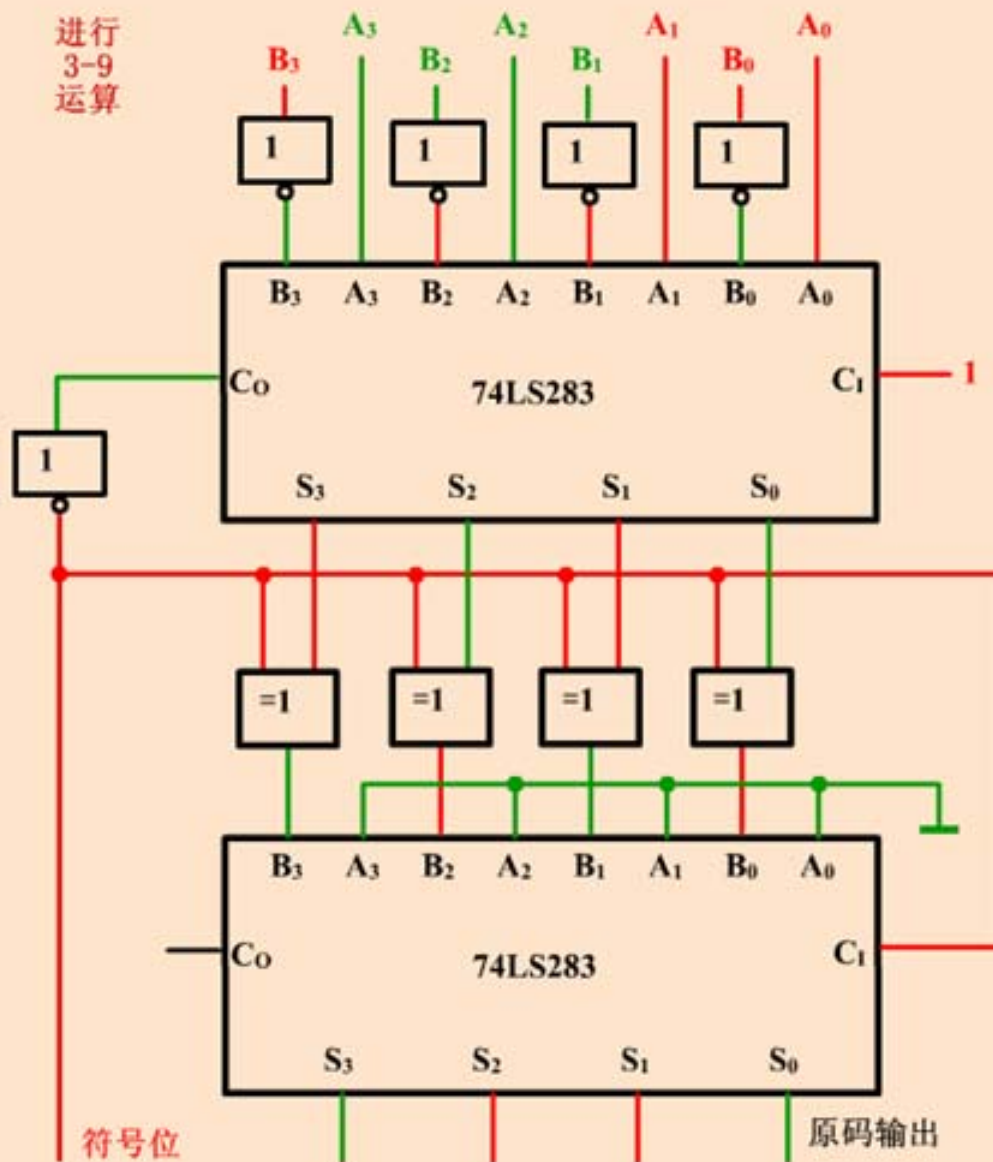
4. 原码输出二进制减法电路

按照补码运算规则设计的原码输出二进制减法电路，如图3.40所示。下面以 $A-B \geq 0$ 和 $A-B < 0$ 两种情况用实例说明其工作过程。



进行
3-9
运算

电路



4位二进制减法器



3.6.3 加法器应用举例

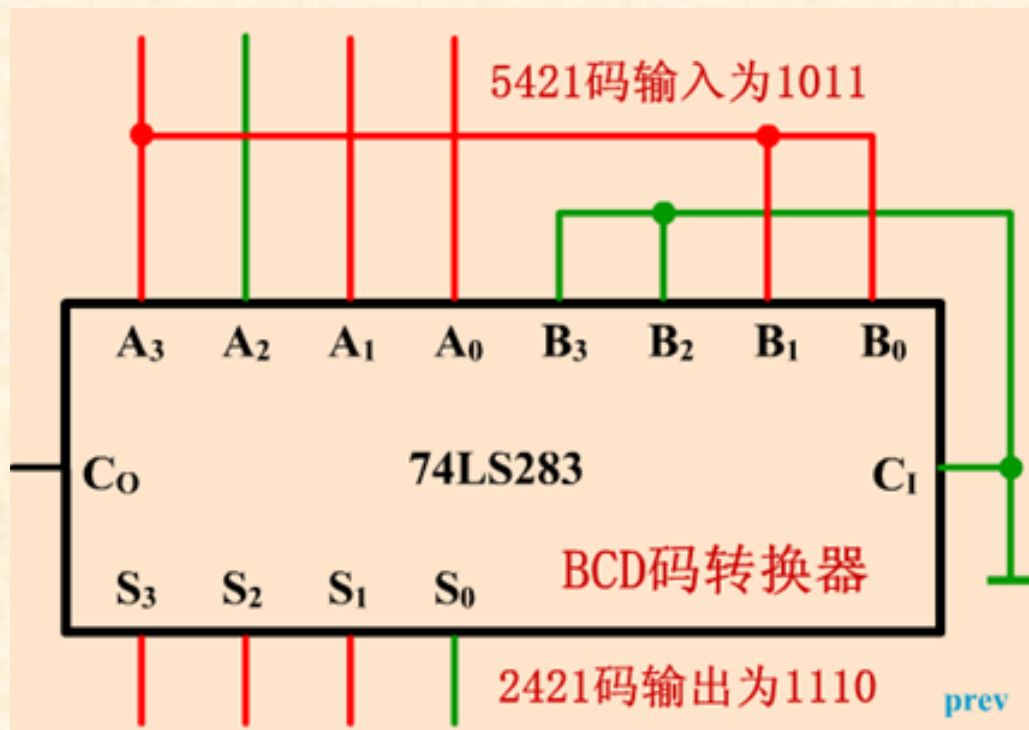
例3.15 利用加法器74LS283将5421BCD码转换为2421BCD码。

解：首先列出两种编码表

十进制数	5421BCD码	2421BCD码
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	1000	1011
6	1001	1100
7	1010	1101
8	1011	1110
9	1100	1111

4以上多3

5以下相同





例3.16 使用四位加法器构成一位8421BCD码的加法电路。

解：BCD码是用4位二进制数表示1位十进制数，BCD码之间是十进制，即逢十进一。

而四位二进制加法器是按四位二进制数进行运算，即逢十六进一。

当和数大于9时，即 $S[3..0] > 1001$ 时，8421BCD码产生进位，而此时十六进制则不一定产生进位，因此需对二进制和数进行修正，即加上6(0110)，让其产生一个进位。

当 $S[3..0] \leq 1001$ (即和数小于等于9)时，则不需要修正或者说加上0(即0000)。



完成设计

需要修正的条件

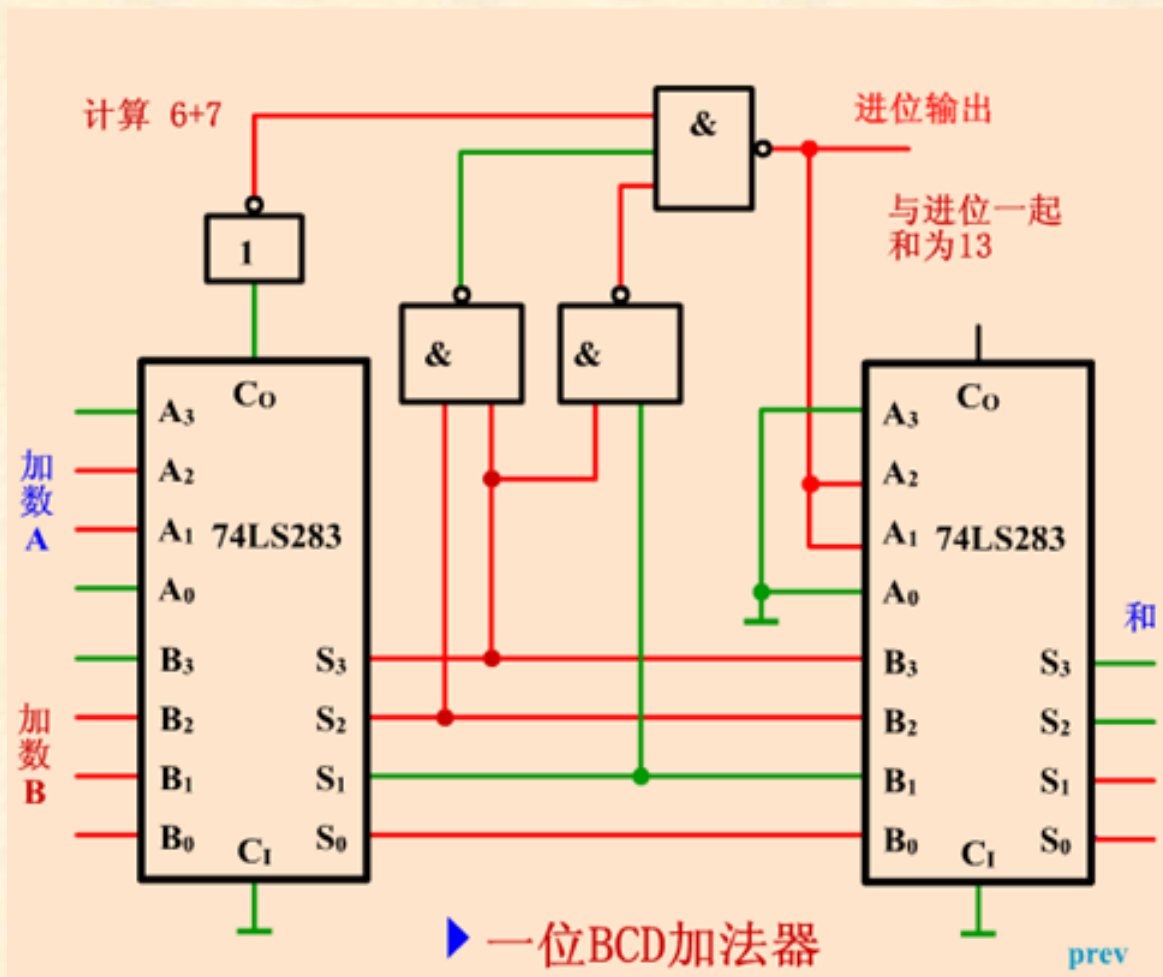
和为16到18的修正

$$L = C_{01} + S_3 S_2 + S_3 S_1 = \overline{\overline{C_{01}}} \cdot \overline{\overline{S_3 S_2}} \cdot \overline{\overline{S_3 S_1}}$$

和为10到15的修正

卡诺图
19

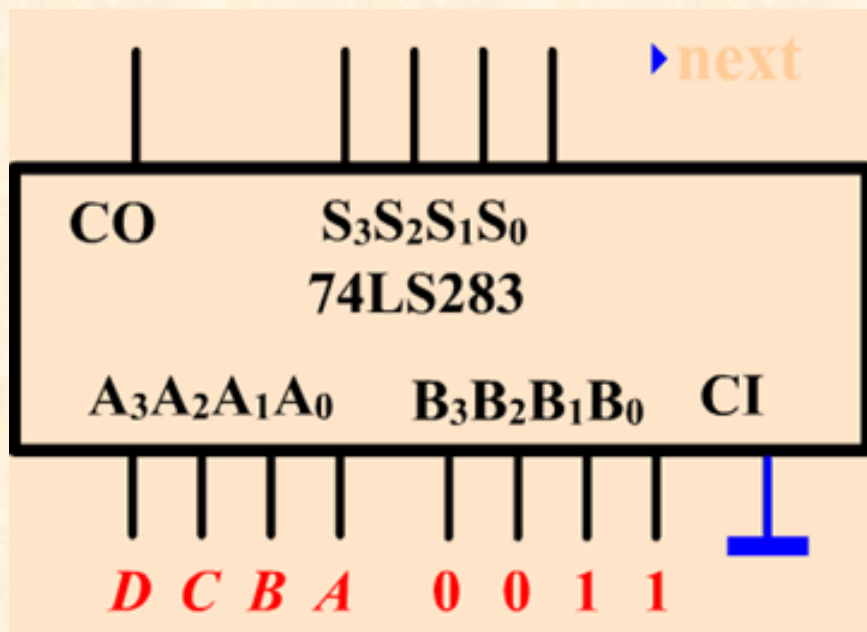
S1S0					
S3S2	00	01	11	10	函数式
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	1	1	1	1	
	0	0	1	1	





实现其它功能

利用74LS283加法运算功能，还可以实现某些具有特定功能的逻辑电路。例如，用74LS283实现由8421码到余3码的代码转换电路，如图示。在图中，DCBA是8421码输入端， $Y_3Y_2Y_1Y_0$ 是余3码输出端。它们之间满足 $Y_3Y_2Y_1Y_0 = DCBA + 0011$ 的关系。



用74LS283实现的代码转换器



3.6.4数值比较器

数值比较器是一种运算电路，它可以对两个二进制数或二-十进制编码的数进行比较，得出大于、小于和相等的结果。



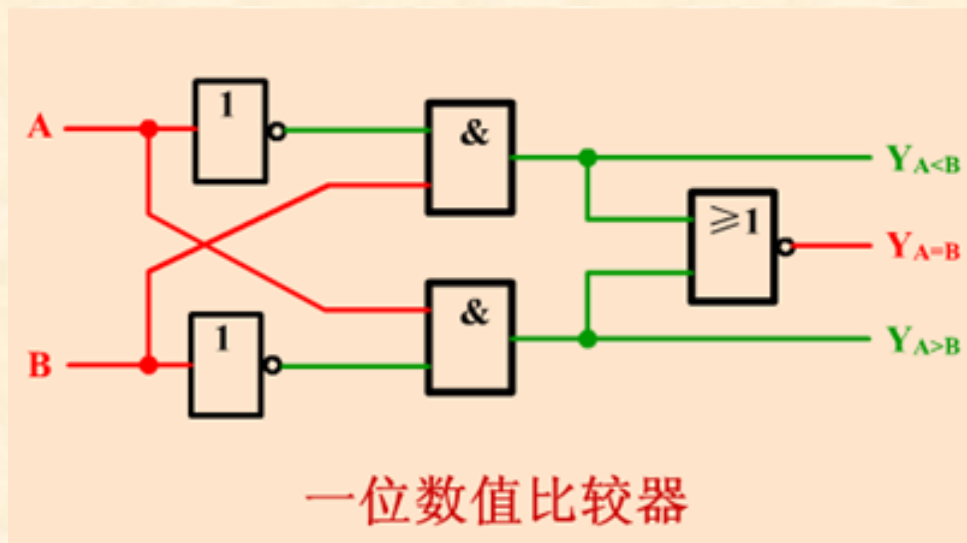
1. 一位数值比较器

1位数值比较器可以对两个1位二进制数A和B进行比较，比较结果分别由 $Y_{A>B}$ (大于)、 $Y_{A<B}$ (小于)和 $Y_{A=B}$ (等于)给出。根据逻辑图得到电路的输出表达式为

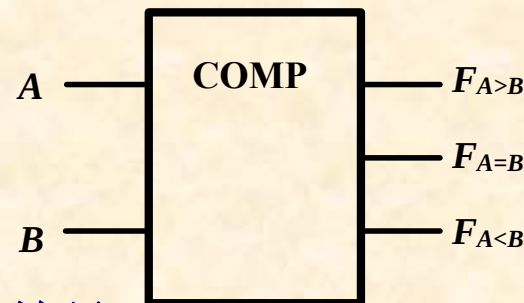
$$\left. \begin{aligned} Y_{A>B} &= A\bar{B} \\ Y_{A=B} &= \overline{A\bar{B} + B\bar{A}} = \overline{A \cdot \bar{B}} + \overline{B \cdot \bar{A}} \\ Y_{A<B} &= \bar{A}B \end{aligned} \right\}$$

1位数值比较器真值表

A	B	$Y_{A>B}$	$Y_{A=B}$	$Y_{A<B}$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



逻辑图



逻辑符号





2. 两位数值比较器

两位数值比较器功能表

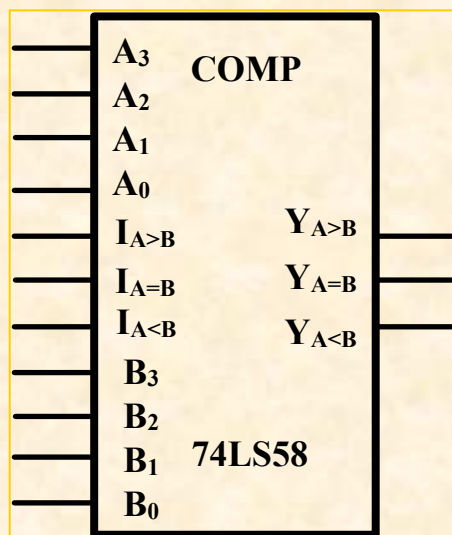
A1 B1	A0 B0	YA>B	YA<B	YA=B
A1>B1	X X	1	0	0
A1<B1	X X	0	1	0
A1=B1	A0>B0	1	0	0
A1=B1	A0<B0	0	1	0
A1=B1	A0=B0	0	0	1

两个多位数比较大小时，必须自高而低的逐位进行比较，若高位的数不等，则可确定数的大小；当高位数相等时，需要比较低位来确定数的大小。



3. 集成数据比较器

常用的4位集成数值比较器有74LS58、MC14585和CC14585等，可以对两个4位二进制数进行比较。7485的功能由表列出。其中， $A_3 \sim A_0$ 是一个4位二进制数A的输入端， $B_3 \sim B_0$ 是另一个4位二进制数B的输入端。 $Y_{A>B}$ 、 $Y_{A=B}$ 和 $Y_{A<B}$ 是输出端。 $I_{A>B}$ 、 $I_{A=B}$ 和 $I_{A<B}$ 是级联输入端，用于芯片的扩展。



输入					输出		
A3 B3	A2 B2	A1 B1	A0 B0	$I_{A>B}$ $I_{A=B}$ $I_{A<B}$	$Y_{A>B}$	$Y_{A=B}$	$Y_{A<B}$
$A_3 > B_3$	X X	X X	X X	X X X	1	0	0
$A_3 < B_3$	X X	X X	X X	X X X	0	0	1
$A_3 = B_3$	$A_2 > B_2$	X X	X X	X X X	1	0	0
$A_3 = B_3$	$A_2 < B_2$	X X	X X	X X X	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X X	X X X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X X	X X X	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X X X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X X X	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	a b c	a	b	c





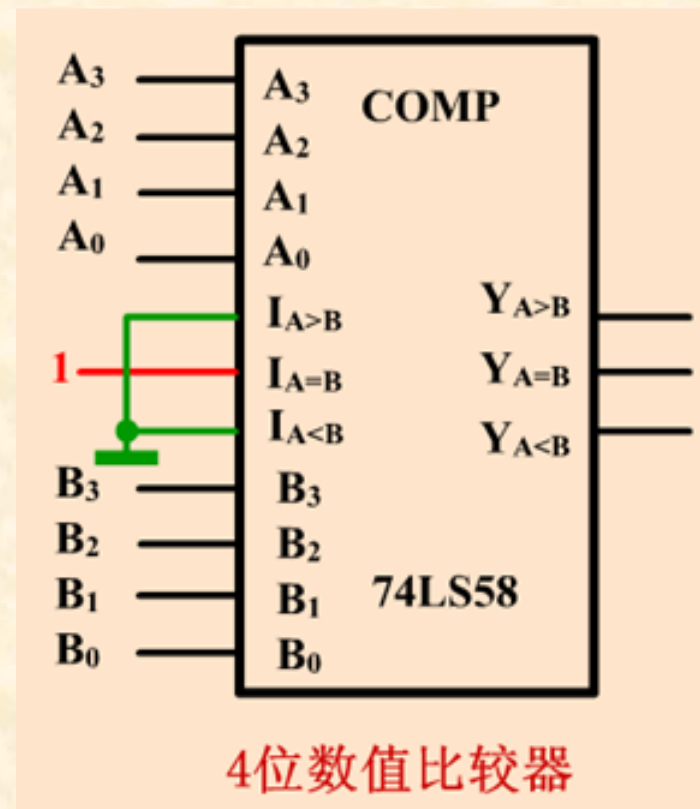
74LS58的正确使用

用一片74LS58可以实现两个4位二进制数的比较，由功能表可知，

当两个4位二进制数相等时，输出 $Y_{A>B}$ 、 $Y_{A=B}$ 和 $Y_{A<B}$ 与级联输入 $I_{A>B}$ 、 $I_{A=B}$ 和 $I_{A<B}$ 相等。

因此，把 $I_{A>B}$ 和 $I_{A<B}$ 接低电平0，把 $I_{A=B}$ 接高电平1。

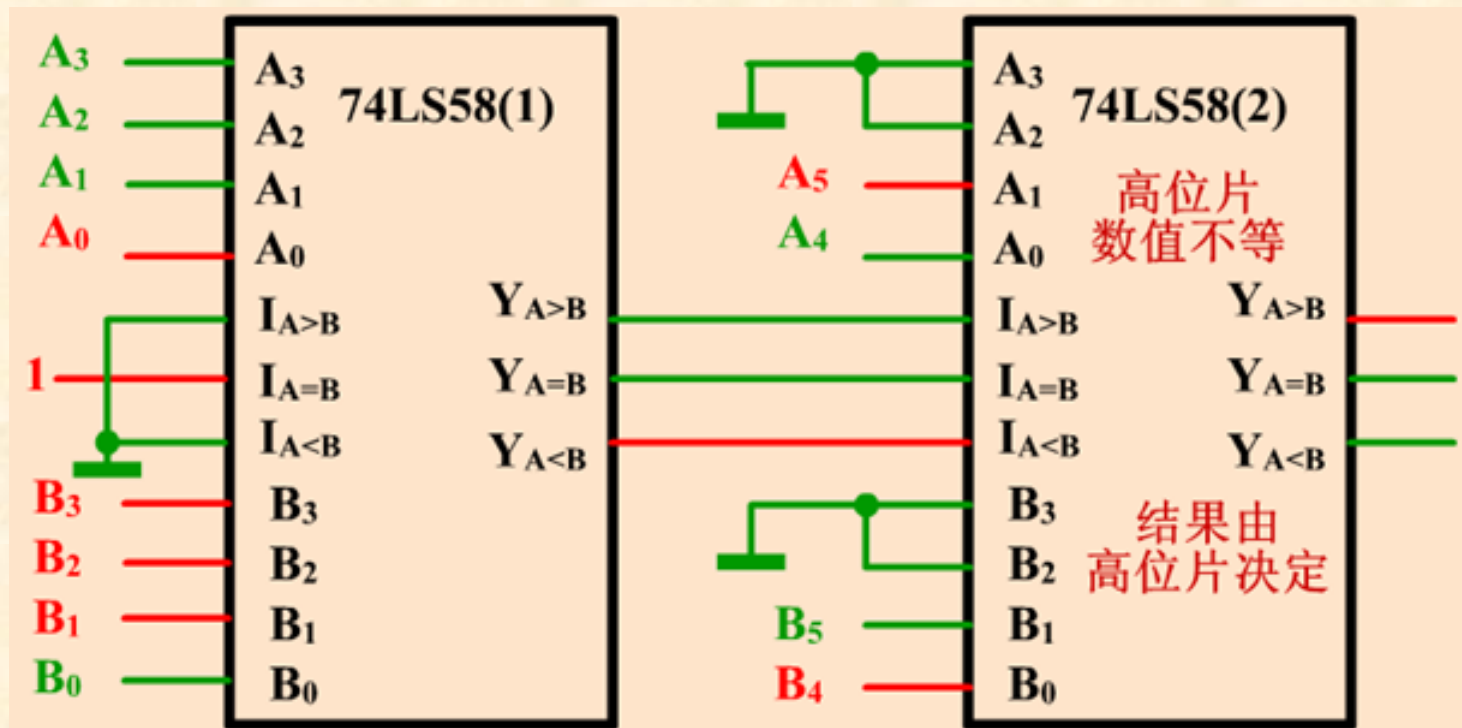
该连接方法同样适用于级联的低四位片





例3.17 用两片74LS58比较两个6位二进制数

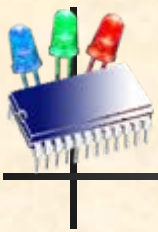
合理使用级联输入端 $I_{A>B}$ 、 $I_{A<B}$ 和 $I_{A=B}$ ，可以实现多片74LS58的扩展。



数值比较器4位扩展为6位

prev





3.7 组合逻辑电路应用举例

组合逻辑电路是各种数字系统的重要组成部分，常用组合逻辑功能器件是构成复杂数字系统的基本单元电路，学习掌握常用组合逻辑功能器件是为了更好地把它应用于数字系统的分析与设计。本节通过几个举例，试图说明组合逻辑电路应用的普遍性与灵活性。



3.7 组合逻辑电路应用举例

组合逻辑电路是各种数字系统的重要组成部分，常用组合逻辑功能器件是构成复杂数字系统的基本单元电路，学习掌握常用组合逻辑功能器件是为了更好地把它应用于数字系统的分析与设计。本节通过几个举例，试图说明组合逻辑电路应用的普遍性与灵活性。



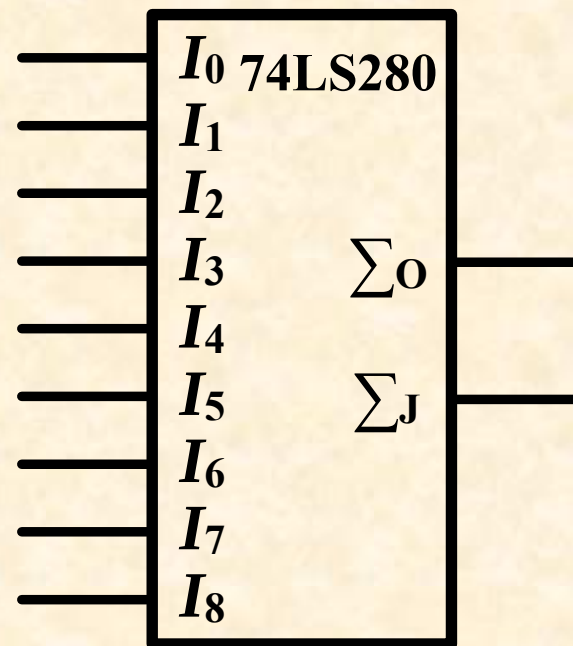
3.7.1 奇偶校验器在数据传输中的应用

当数据代码在系统中从一点传送到另外一点，或者数据代码由一个系统传送到另外一个系统时，可能由于器件的故障或者电子噪声，使某一位代码由1变成0，或者由0变成1，即数据在传输过程中发生错误。尽管出现这种错误的概率很小，但有必要对出现的错误进行检测。许多系统都使用一个奇偶校验位作为位错误检测的手段。一个奇偶校验位附加到多位数码组中，使得这组数码中1的个数总保持偶数或者奇数，这种方法叫做奇偶校验。一个偶校验位使得1的总数保持偶数，而奇校验位使得1的总数为奇数(参见表1.4)。对于一个特定的系统，只能选择奇校验或者偶校验一种校验方式。在数码组中附加的奇偶校验位可以在原数码组的开头或者结尾。



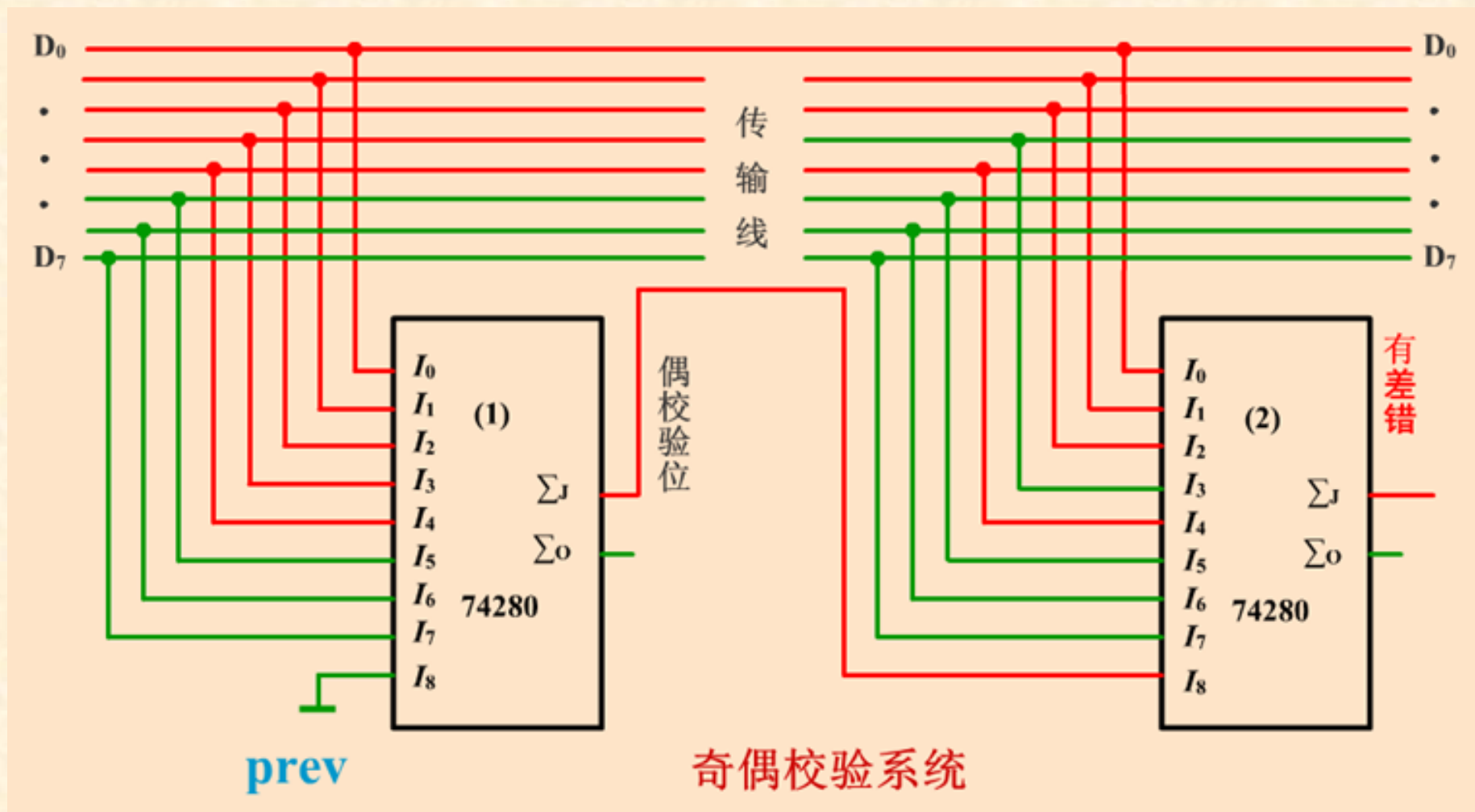
9位奇偶发生器/校验器

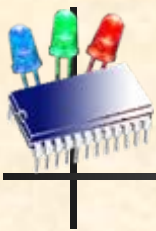
奇偶校验器还具有奇偶产生的功能，通常把它称为奇偶发生器 / 校验器。74LS280是9位奇偶发生器 / 校验器，其逻辑符号如图。74LS280可以对一个9位代码(8位数据位和一位校验位)进行奇校验或者偶校验。或者对一个9位代码产生一个校验位。当输入 $I_0 \sim I_8$ 中有偶数个1时， Σ_o 偶数输出端为高电平；当输入 $I_0 \sim I_8$ 中有奇数个1时， Σ_j 奇数输出端为高电平。 Σ_o 和 Σ_j 输出总是相反。





奇偶发生器/校验器的应用





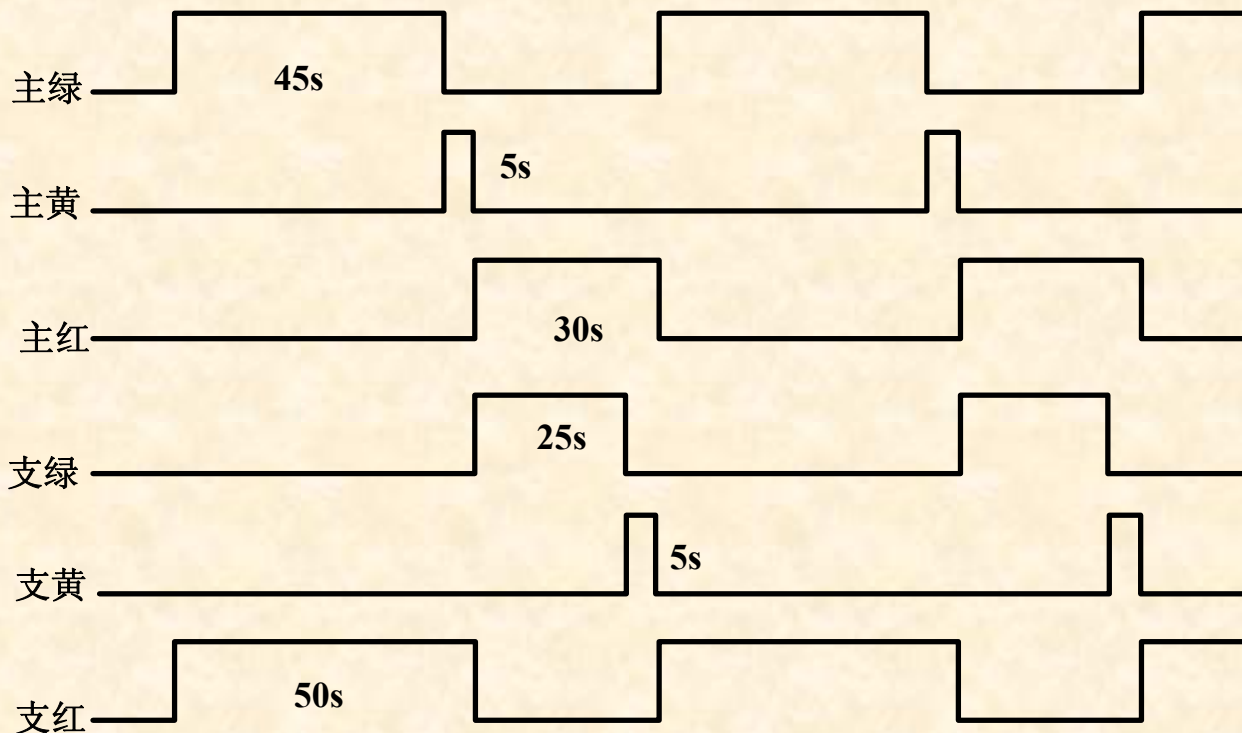
3.7.2 简易交通信号灯控制电路

一个完整的交通信号灯控制系统单凭组合逻辑电路是无法实现的，但其中的部分功能可以利用组合逻辑电路设计。此处仅考虑组合逻辑电路部分，完整的电路将在学习了时序逻辑电路之后讨论。



波形图

某一平交路口交通信号灯定周期控制的灯时波形图。分析各种灯时的要求，可见最小分辨时间为5秒，一个周期为80秒。如果以5秒为单位，可以把一个周期区分为16种状态组合，用S3S2S1S0表示，S3S2S1S0分别取值0000~1111。





真值表

$S_3S_2S_1S_0$	MG	MY	MR	BG	BY	BR	$S_3S_2S_1S_0$	MG	MY	MR	BG	BY	BR
0000	1	0	0	0	0	1	1000	1	0	0	0	0	1
0001	1	0	0	0	0	1	1001	0	1	0	0	0	1
0010	1	0	0	0	0	1	1010	0	0	1	1	0	0
0011	1	0	0	0	0	1	1011	0	0	1	1	0	0
0100	1	0	0	0	0	1	1100	0	0	1	1	0	0
0101	1	0	0	0	0	1	1101	0	0	1	1	0	0
0110	1	0	0	0	0	1	1110	0	0	1	1	0	0
0111	1	0	0	0	0	1	1111	0	0	1	0	1	0

卡诺图
56-59





表达式

依据表并化简可得各个输出函数的表达式如下

$$MG = \overline{S_3} + \overline{S_2} \overline{S_1} \overline{S_0}$$

$$MY = S_3 \overline{S_2} \overline{S_1} \overline{S_0}$$

$$MR = BG + BY$$

$$BG = S_3 S_2 \overline{S_1} + S_3 \overline{S_2} S_1 + S_3 S_1 \overline{S_0}$$

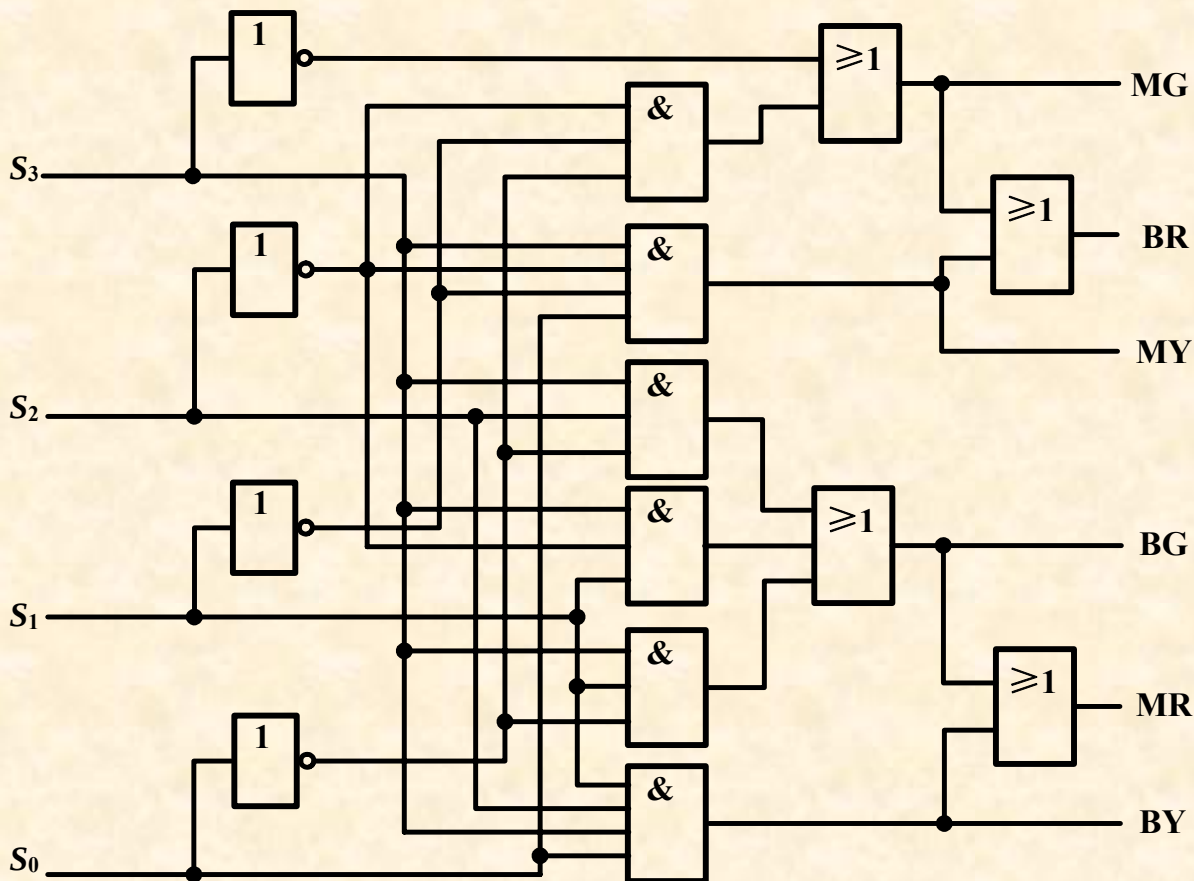
$$BY = S_3 S_2 S_1 S_0$$

$$BR = MG + MY$$



逻辑图

由输出函数表达式作逻辑电路图如图所示





1. 竞争

实际信号从输入到输出的过程中，所通过的路径不同，其经过的门的级数有可能不同；门电路的种类不同，则其平均延迟时间也不尽相同，

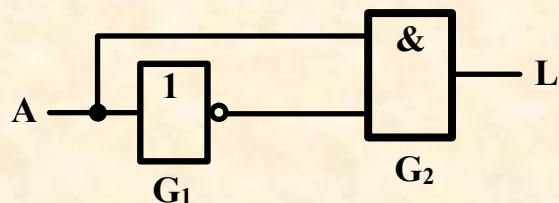
一个输入信号经过不同的路径到达同一点的时间不同。这种现象称为竞争。



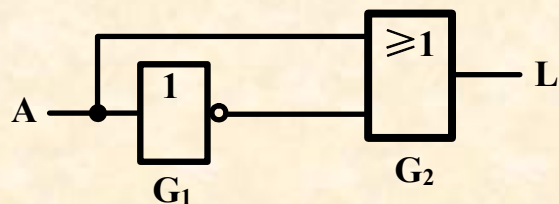


2. 冒险

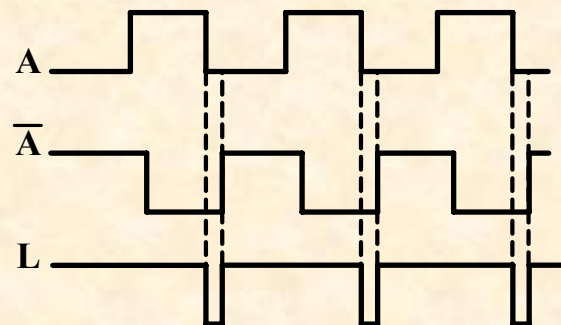
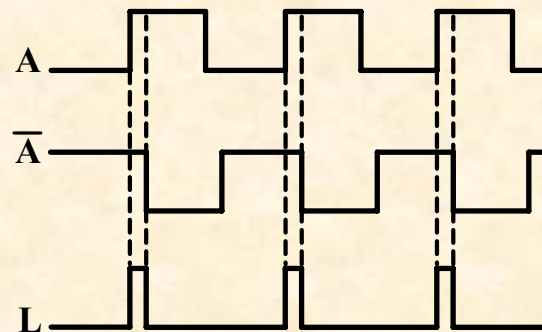
竞争现象有可能导致逻辑电路瞬时输出出现错误信号，这一现象称为冒险(或称为险象)。



$$L = A \cdot \bar{A} = 0$$



$$L = A + \bar{A} = 1$$



注意:并不是所有的竞争都能产生冒险，有一些竞争是不会产生冒险的。



3.8.2 冒险现象的判别

当逻辑函数在一定输入取值下, 可以化为 $L=A + \bar{A}$ 或 $L=A \cdot \bar{A}$ 的形式时, A 的变化将会引起冒险.

几种常用的判别冒险现象的方法

1. 代数法

首先找出具有竞争能力的变量, 然后逐次改变其他变量, 判断是否存在冒险现象。

例3.18判断函数 $L=A\bar{C}+\bar{A}B+\bar{A}C$ 是否存在冒险现象.

解:由逻辑函数式可以看出A和C都具有竞争能力.

根据分析可知,在 $B=1, C=0$ 时, $L=A+\bar{A}$, A可以引起冒险. 而C虽具有竞争能力, A取任何值时C和 $\bar{C}$ 不会同时出现,即C不会引起冒险现象.

例3.19 判断 $L=(\bar{A} + \bar{C}) \cdot (A+B) \cdot (B+C)$ 是否存在冒险现象.

解:由逻辑函数式可以看出A和C都具有竞争能力.

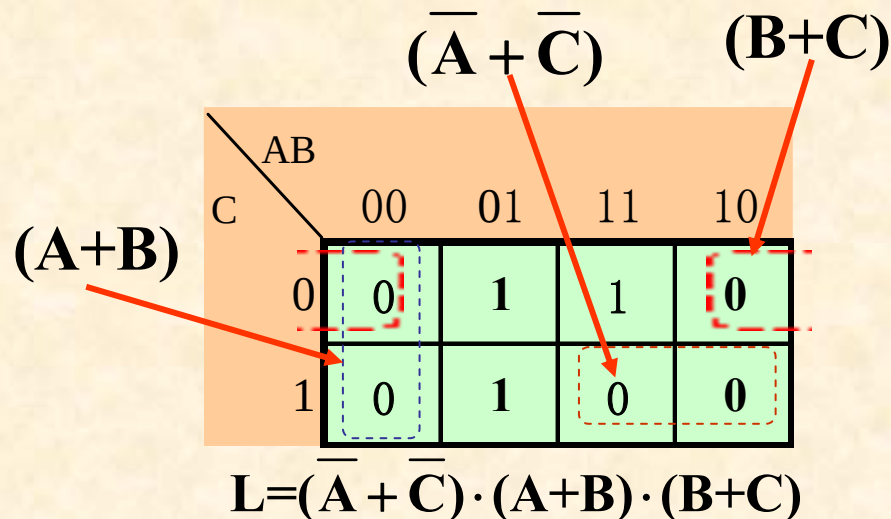
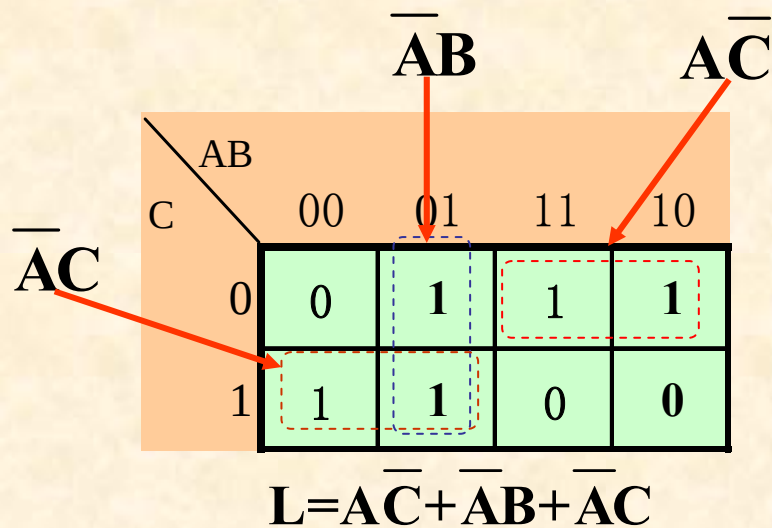
在 $B=0, C=1$ 时, $L=\bar{A} \cdot A$, A可以引起冒险现象.

在 $B=0, A=1$ 时, $L=\bar{C} \cdot C$, C可以引起冒险现象.



2. 卡诺图法

在逻辑化简中，卡诺图法具有简单直观的特点。用卡诺图也可以简单、直观地判断组合逻辑电路的冒险。在卡诺图中，如果两个乘积项(即卡诺图中的两个“圈”)出现相邻而不相交的现象时，则存在冒险。



存在冒险的卡诺图





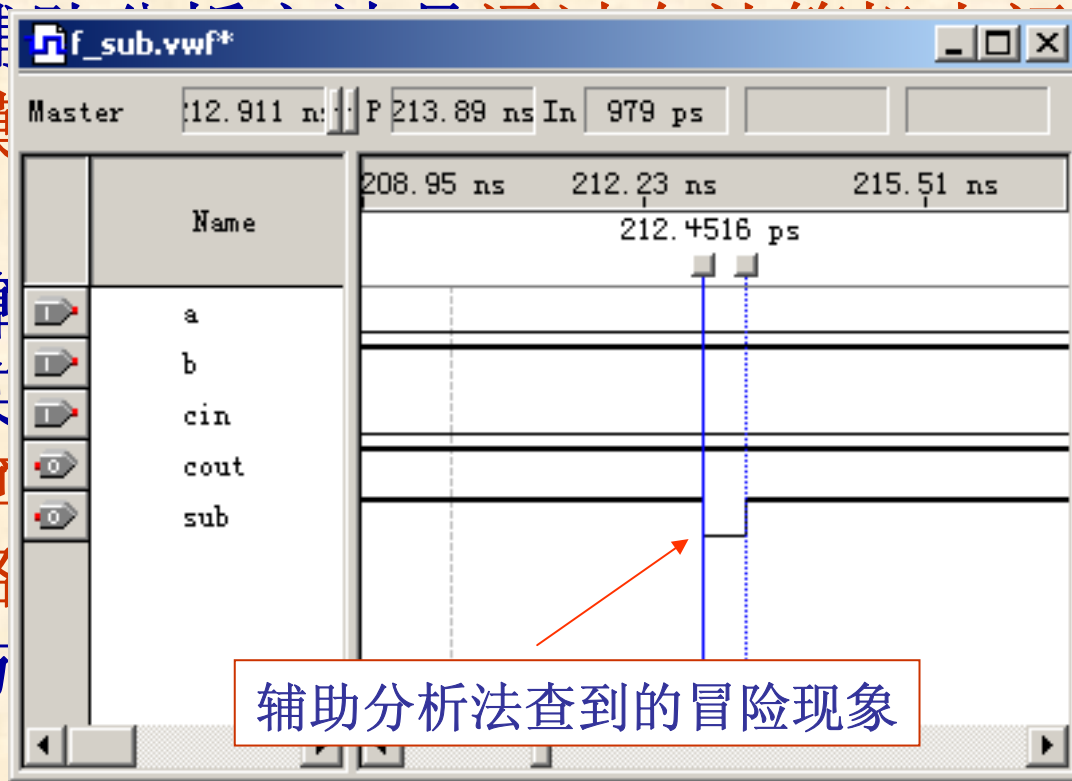
3. 计算机辅助分析法

实际的逻辑电路输入变量通常会比较多，并且有可能多个输入变量同时发生变化。在这种情况下，很难利用上述的判别方法找出所有的冒险现象。

计算机辅助分析法通过计算机辅助设计软件进行数字电路的模拟。

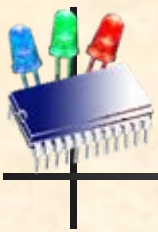
目前，由于计算而且某用计算机字电路验的方

型参数，。所以，实际的数需要用实象。



辅助分析法查到的冒险现象





4.实验法

是检验电路是否存在冒险现象的最有效、最可靠的方法。

实验法是利用实验手段检查冒险的方法，即在逻辑电路的输入端，加入信号所有可能的组合状态，用逻辑分析仪或示波器，捕捉输出端可能产生的冒险现象。实验法检查的结果是最终的结果。



3.8.3 消除冒险现象的方法

当逻辑电路存在着冒险现象时，会对电路的正常工作造成威胁。因此，必须设法消除。

常采用的消除冒险现象的方法有以下几种：
修改逻辑设计、选通法和滤波法几种。



1.修改逻辑设计

(1)增加多余项法

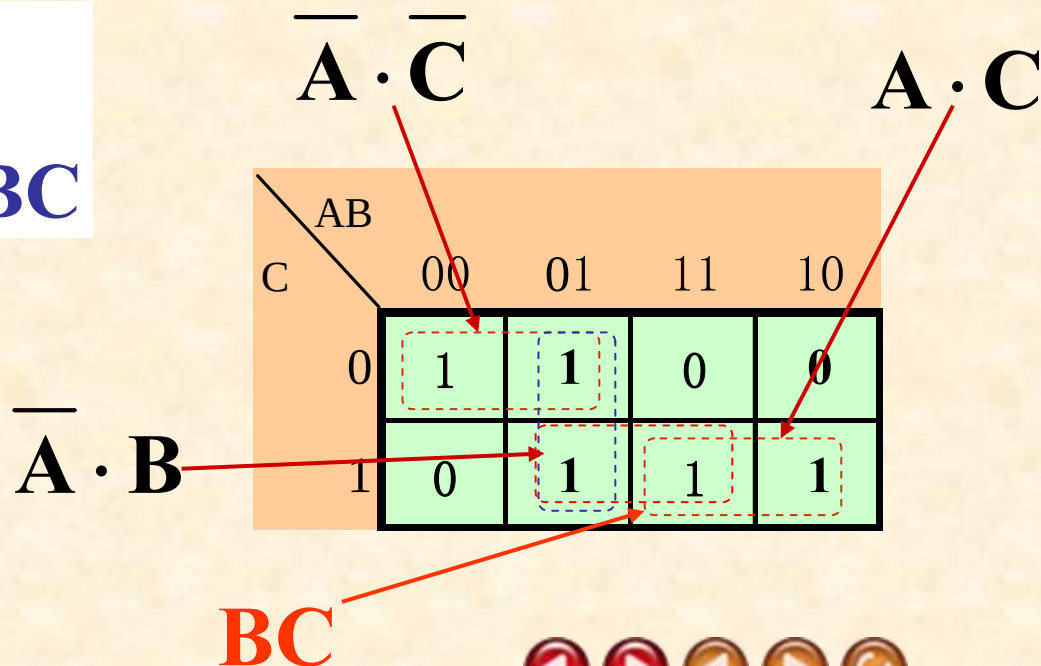
在逻辑表达式中添加多余项，消除冒险现象。

消除函数 $L = A \cdot C + \overline{A} \cdot B + \overline{A} \cdot \overline{C}$ 中存在的冒险

增加多余项的函数为

$$L = A \cdot C + \overline{A} \cdot B + \overline{A} \cdot \overline{C} + BC$$

用多余项将相邻而不相交的乘积项连在一起





(2)消掉互补变量法

对逻辑表达式进行逻辑变换，消掉互补变量。

例 3.20 消除 $L=(\bar{A}+\bar{C}) \cdot (A+B) \cdot (B+C)$ 中的冒险现象。

解：对逻辑表达式进行变换：

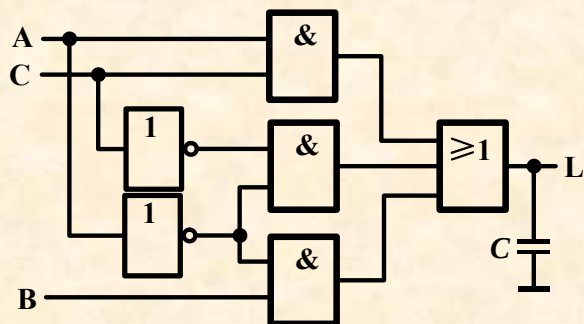
$$L=(\bar{A}+\bar{C}) \cdot (A+B) \cdot (B+C)=\bar{A}B+AB\bar{C}+B\bar{C}+\bar{A}BC=\bar{A}B+B\bar{C}$$

由 $L=\bar{A}B+B\bar{C}$ 组成逻辑电路，不会出现冒险现象。

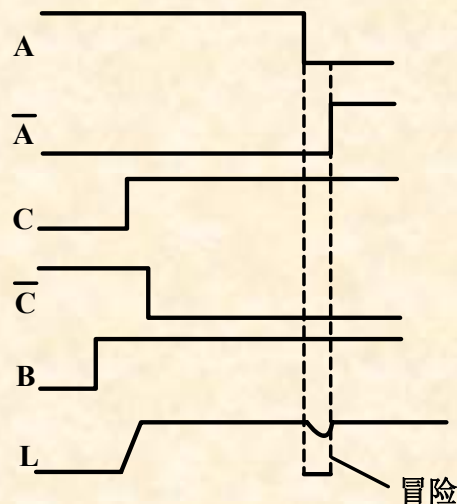


2.滤波法

由于冒险是一种很窄的尖峰脉冲，所以只要在输出端接一个很小的滤波电容 C_o (如图示)，在电路输出端加低通滤波器，就可以削掉尖脉冲信号。



用滤波法消除冒险

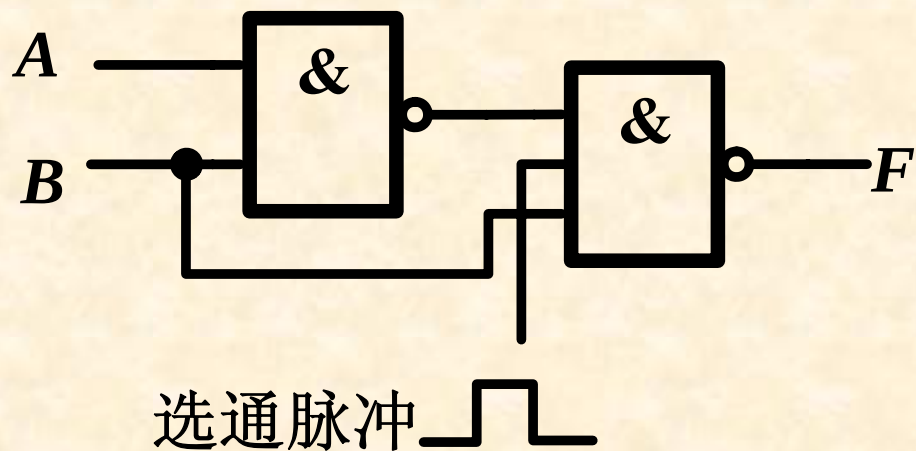


滤波法简单易行，而且不必考虑冒险产生的内部原因，也不需要改变内部电路，只要有冒险，都可以消除。但是在输出端接滤波电容后，会使输出波形上升时间和下降时间增加，使波形变坏。



3.选通法

冒险发生在输入信号发生变化的瞬间，选通法就是在电路的输出门的输入增加选通控制信号，如图示。没有选通脉冲时，电路没有输出；当选通脉冲到来时，输出才是有效信号。选通脉冲是等到输入信号稳定后才出现，这样可以避免冒险。



用选通法消除冒险



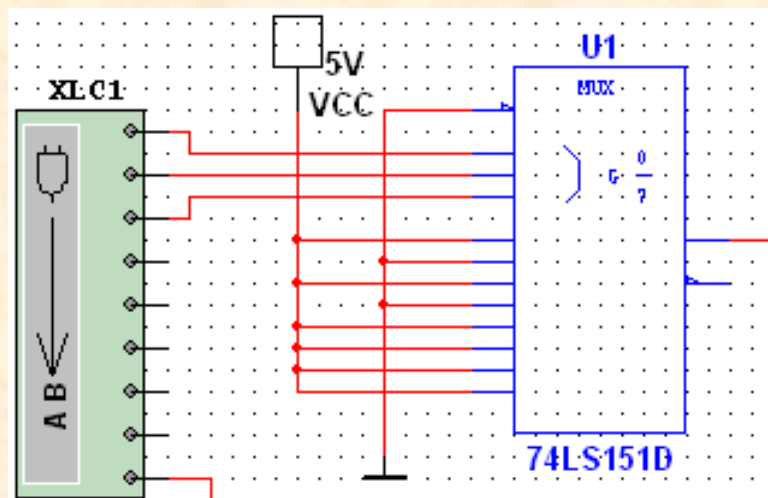
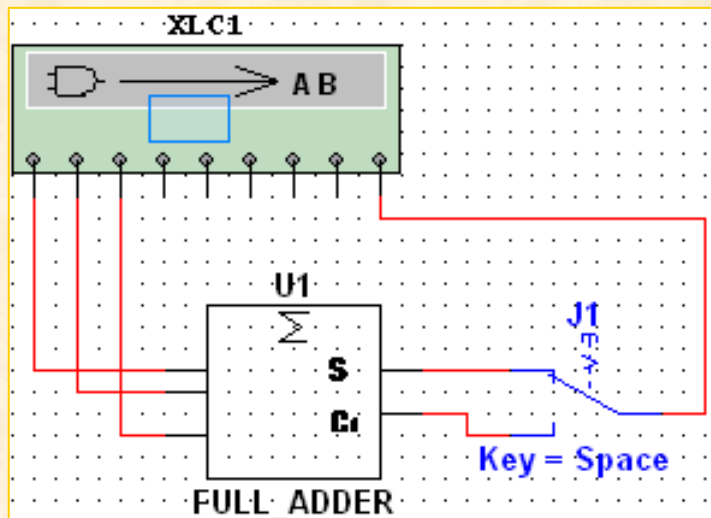
用Multisim的逻辑转换器得到组合电路的真值表和函数式





例2 对集成电路的分析

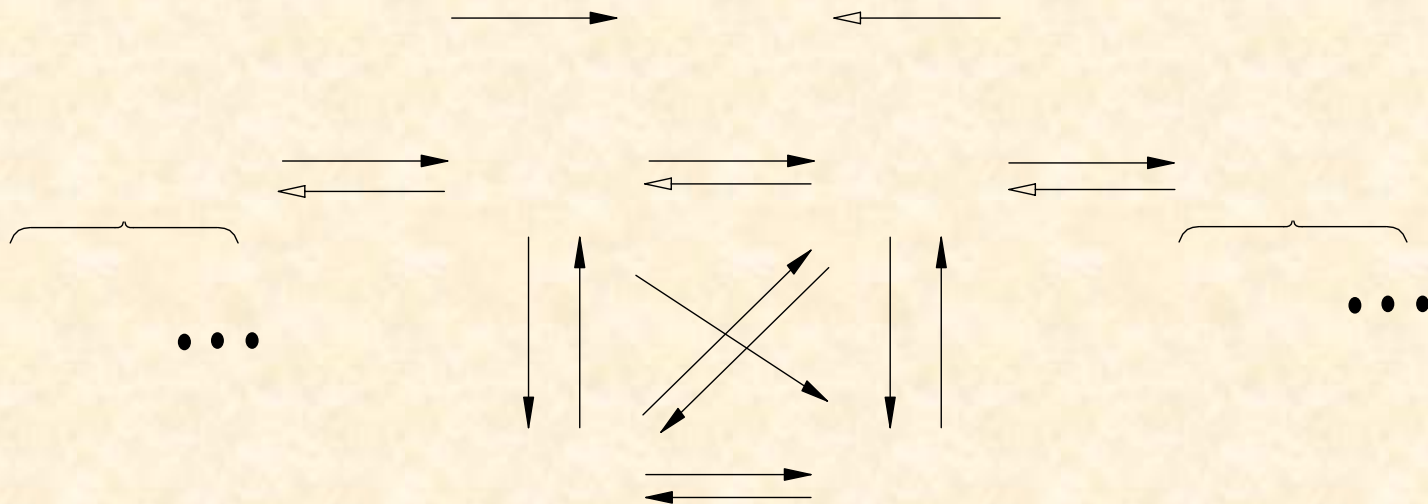
对全加器及改接过的数据选择器的分析





本章小结

- (1) 组合逻辑电路在逻辑功能上的特点是：任意时刻的输出仅仅取决于该时刻的各输入变量的状态组合，而与电路过去的状态无关。它在电路结构上的特点是：只包含门电路，没有存储(记忆)单元。
- (2) 分析组合电路的目的是确定已知电路的逻辑功能。利用逻辑门设计组合逻辑电路，常以电路简单、所用器件个数以及种类最少为设计原则。组合逻辑电路分析与设计的方法可归纳如图所示。





本章小结(续)

- (3)使用中规模集成器件，可以大大简化设计。大多数中规模集成电路都设置了控制端，这些控制端既可以作为输出信号的选通端，用于控制电路的工作状态，又可以用来扩展电路的功能。因此，控制端的灵活应用是中规模组合电路设计中的关键。
- (4)竞争—冒险是组合逻辑电路工作状态转换过程中经常会出现的一种现象。如果负载是一些对尖峰脉冲敏感的电路，则必须采取措施消除由于冒险而产生的尖峰脉冲。如果负载电路对尖峰脉冲不敏感(例如光电显示器)，就不必考虑冒险问题。