

第四章 组合逻辑电路

4.1 概 述

主要要求:

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

4.3.3 数据选择器

主要要求:

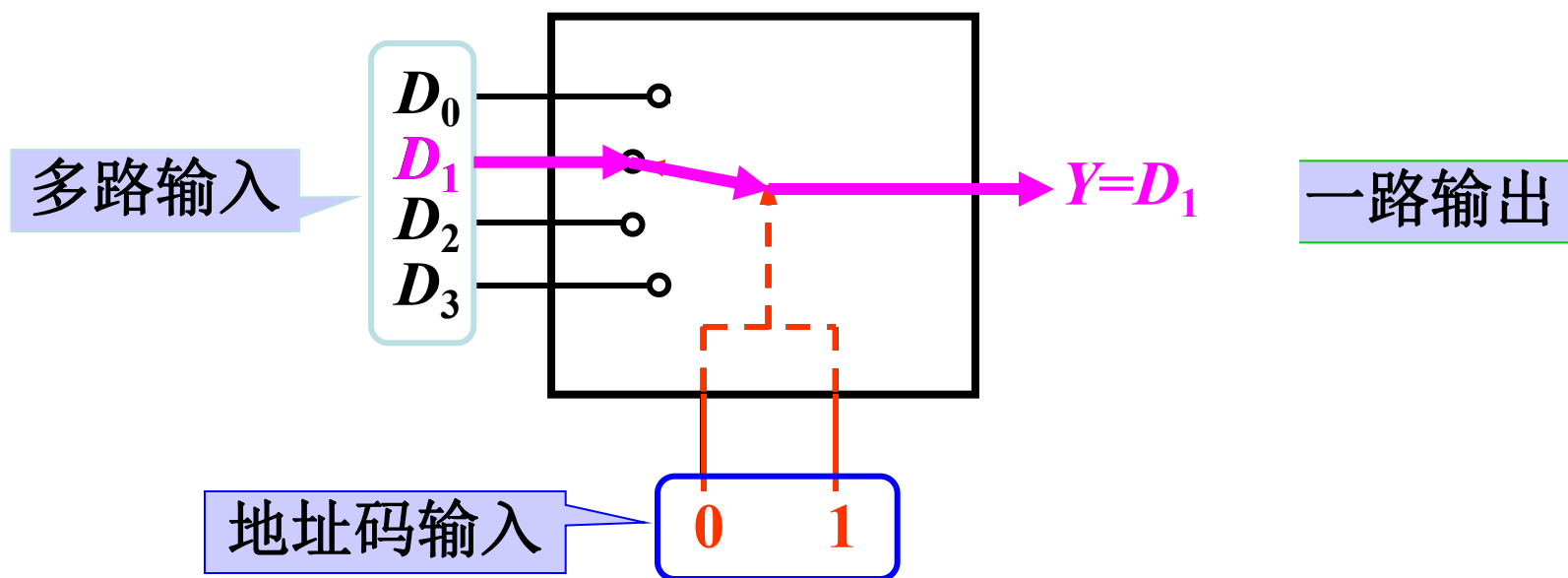
- 理解数据选择器和数据分配器的作用。
- 理解常用数据选择器的逻辑功能及其使用。
- 掌握用数据选择器实现组合逻辑电路的方法。

一、数据选择器和数据分配器的作用

数据选择器：根据地址码的要求，从多路输入信号中选择其中一路输出的电路。

又称多路选择器 (Multiplexer, 简称MUX) 或多路开关。

4 选 1 数据选择器工作示意图

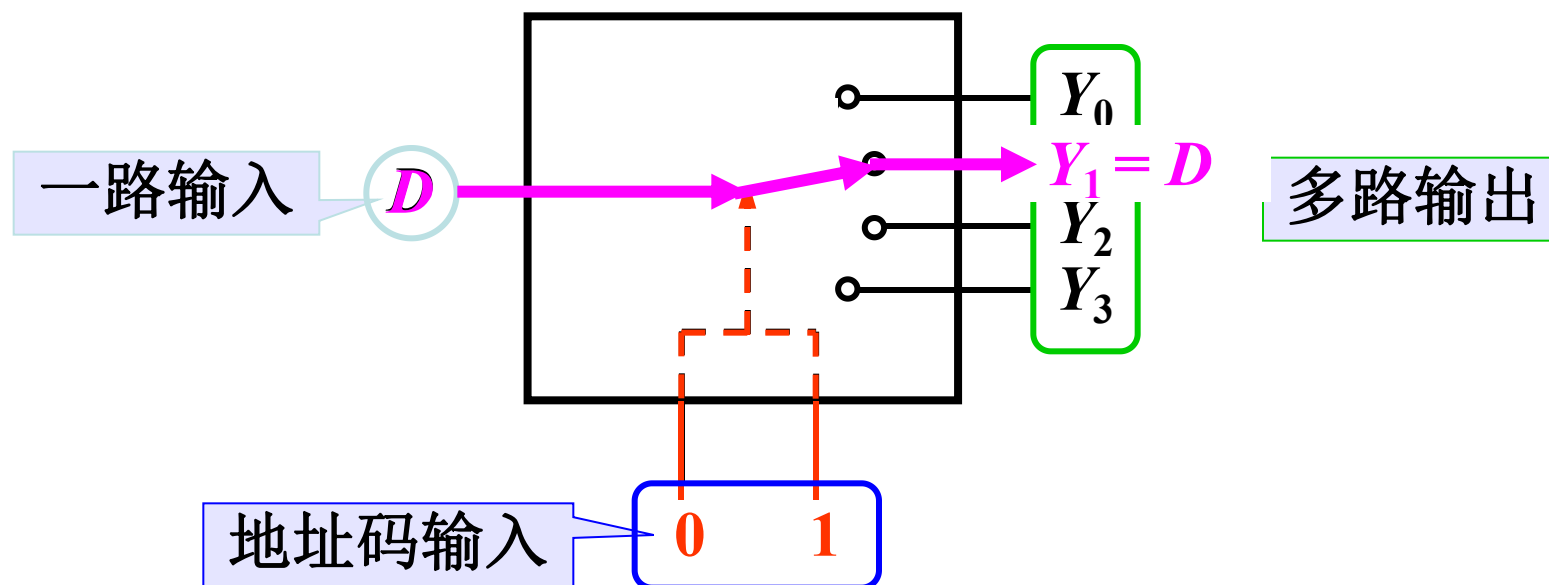


数据选择器的输入信号个数 N 与地址码个数 n 的关系为 $N = 2^n$

数据分配器：根据地址码的要求，将一路数据分配到指定输出通道上去的电路。

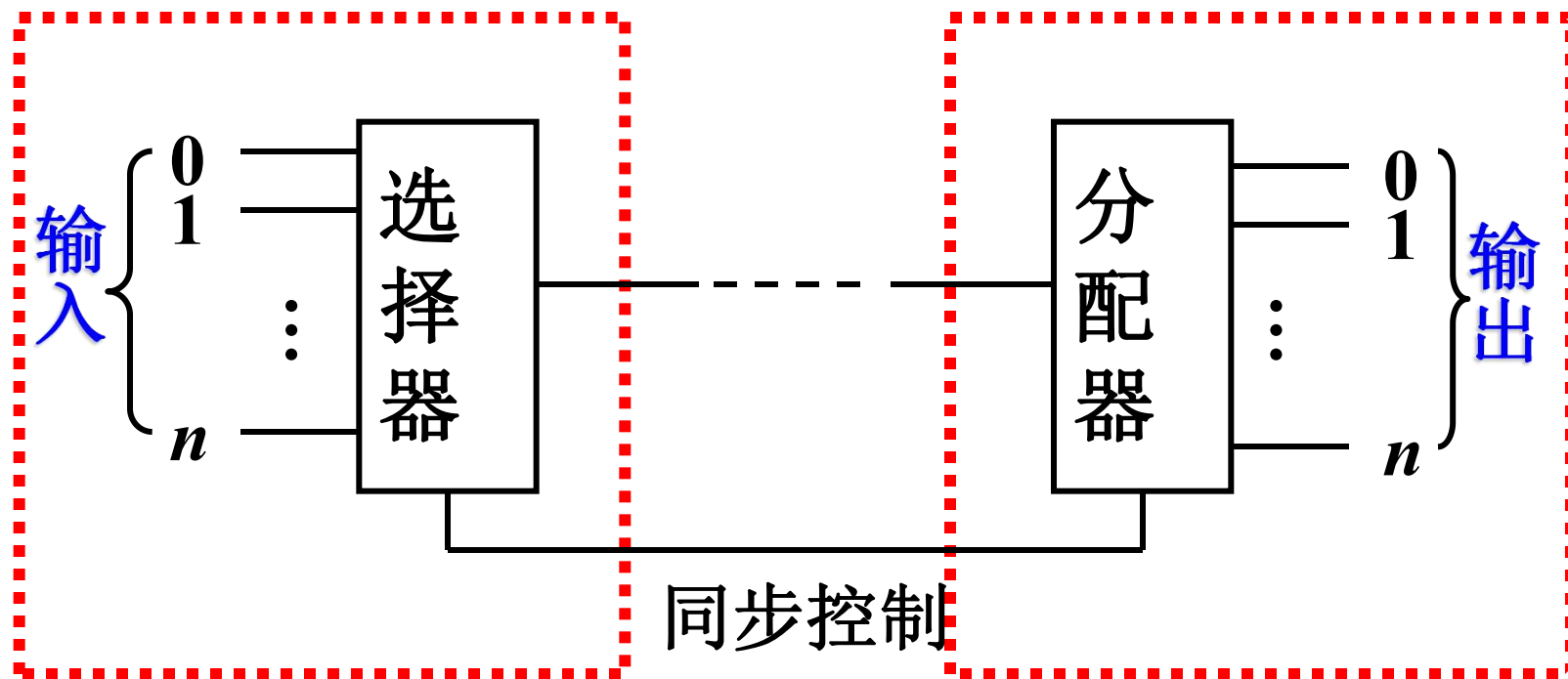
Demultiplexer, 简称DMUX

4路数据分配器工作示意图



发送端，并—串

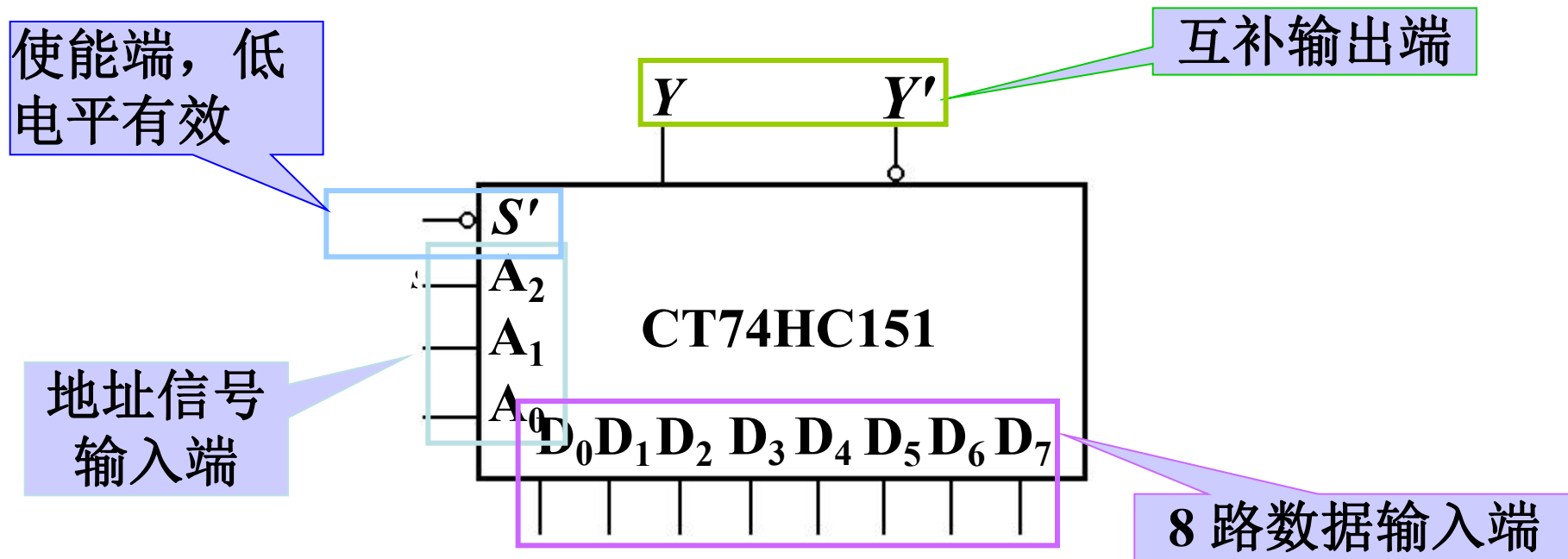
接收端，串—并



串行传输数据示意图

二、数据选择器的逻辑功能及其使用

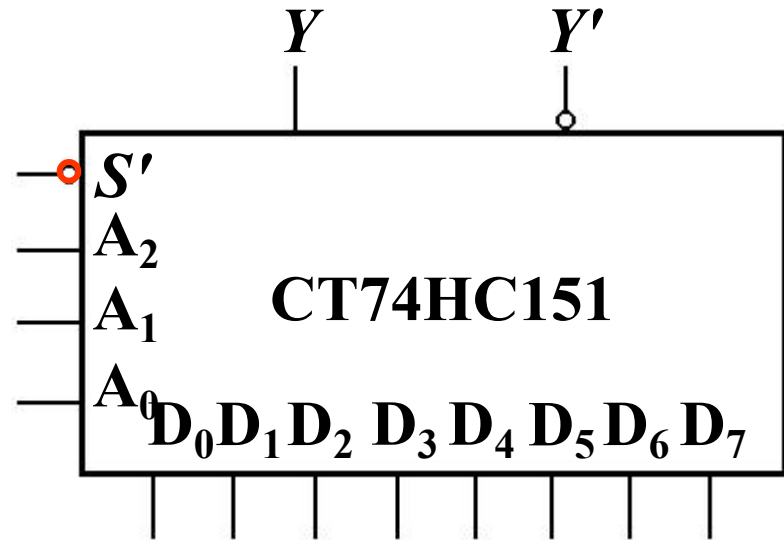
8 选 1 数据选择器 CT74HC151



CT74HC151的逻辑功能示意图

8 选 1 数据选择器 CT74HC151 真值表

输 入				输 出
S'	A_2	A_1	A_0	Y
1	×	×	×	0
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7



CT74HC151 输出函数表达式

输 入				输 出
S'	A_2	A_1	A_0	Y
1	×	×	×	0
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7

$$\begin{aligned}
 Y &= A_2'A_1'A_0'D_0 + A_2'A_1'A_0D_1 + \\
 &\quad A_2'A_1A_0'D_2 + A_2'A_1A_0D_3 + \\
 &\quad A_2A_1'A_0'D_4 + A_2A_1'A_0D_5 + \\
 &\quad A_2A_1A_0'D_6 + A_2A_1A_0D_7 \\
 &= m_0D_0 + m_1D_1 + m_2D_2 + m_3D_3 + \\
 &\quad m_4D_4 + m_5D_5 + m_6D_6 + m_7D_7
 \end{aligned}$$

其输出端能提供地址
输入变量的全部最小项。

能实现函数
发生器吗？
怎样实现？

三、用数据选择器实现组合逻辑函数

$$Y = m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 +$$

$$m_4 D_4 + m_5 D_5 + m_6 D_6 + m_7 D_7$$

$$Y = \sum_{i=0}^{2^k-1} m_i D_i$$

设： K 为选择器的选择输入端数， N 为逻辑函数的变量数

1、 $N=K$

2、 $N < K$

3、 $N > K$

1、 $N=K$

★ [例] 试用数据选择器实现函数 $Y = AB + AC + BC$ 。

代数法求解 选用 CT74HC151

解: (1) 写出逻辑函数的最小项表达式

$$Y = AB + AC + BC = A'BC + AB'C + ABC' + ABC$$

(2) 写出数据选择器的输出表达式

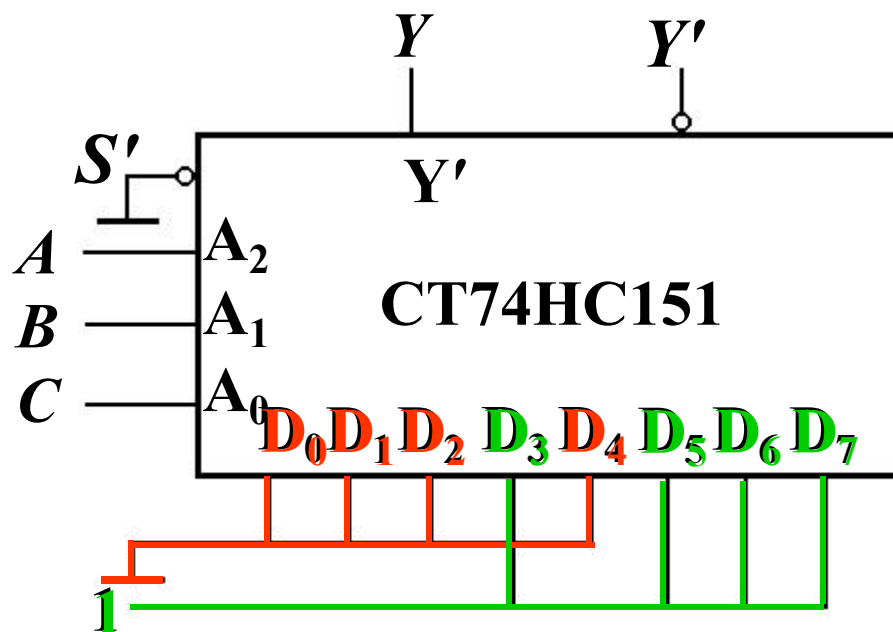
$$Y' = A'B'C'D_0 + A'B'CD_1 + A'BC'D_2 + A'BCD_3 \\ + AB'C'D_4 + AB'CD_5 + ABC'D_6 + ABCD_7$$

(3) 比较 Y 和 Y' 两式中最小项的对应关系

$$\text{令 } A = A_2, B = A_1, C = A_0$$

$$\text{为使 } Y = Y', \text{ 应令 } \begin{cases} D_0 = D_1 = D_2 = D_4 = 0 \\ D_3 = D_5 = D_6 = D_7 = 1 \end{cases}$$

(4) 画连线图



卡诺图法求解

[例] 试用数据选择器实现函数 $Y = AB + AC + BC$ 。

解: (1) 选择数据选择器 选用 CT74HC151

(2) 画出 Y 和数据选择器输出 Y' 的卡诺图

Y
的
卡
诺
图

BC					
		00	01	11	10
A	0	0	0	1	0
	1	0	1	1	1

Y'
的
卡
诺
图

$A_2 \backslash A_1 A_0$		$A_1 A_0$			
		00	01	11	10
A_2	0	D_0	D_1	D_3	D_2
	1	D_4	D_5	D_7	D_6

(3) 比较逻辑函数 Y' 和 Y 的卡诺图

设 $Y = Y'$ 、 $A = A_2$ 、 $B = A_1$ 、 $C = A_0$

对比两张卡诺图后得 $\begin{cases} D_0 = D_1 = D_2 = D_4 = 0 \\ D_3 = D_5 = D_6 = D_7 = 1 \end{cases}$

(4) 画连线图 与代数法所得图相同

用数据选择器实现逻辑函数, 用8—1MUX实现Y(A,B,C), 真值表已知

对比

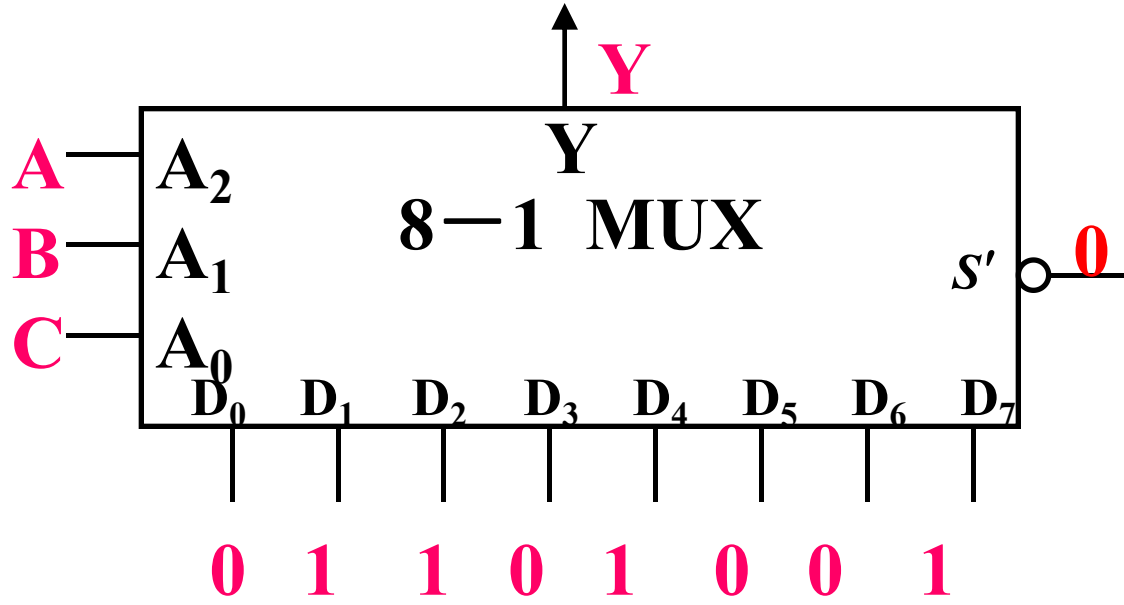
Y函数真值表

8-1MUX功能表

ABC	Y
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

A ₂ A ₁ A ₀	Y
000	D ₀
001	D ₁
010	D ₂
011	D ₃
100	D ₄
101	D ₅
110	D ₆
111	D ₇

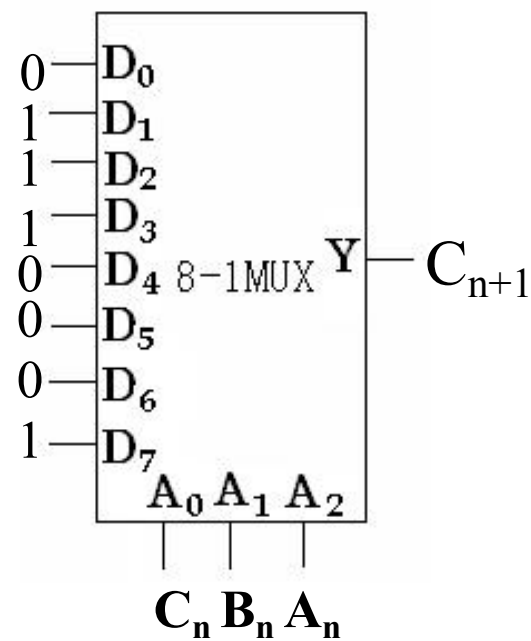
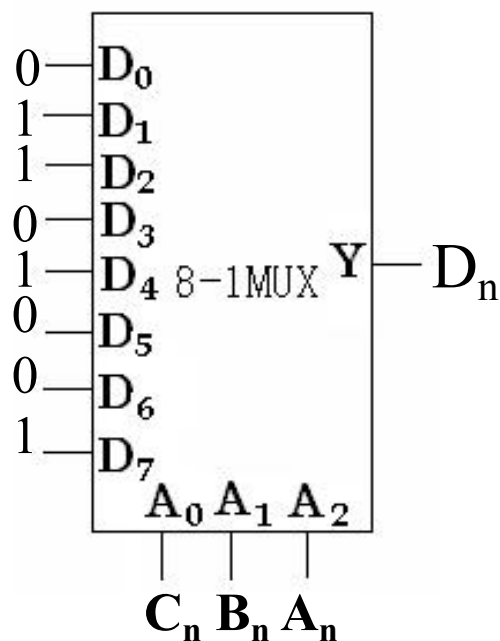
- 1) 列8-1MUX功能表
- 2) 对比Y函数真值表和8-1MUX功能表
- 3) 连接对应输入输出信号
- 4) 给控制信号正确的值



例：用8-1MUX实现一位全减器

全减器真值表

输入			输出	
A_n	B_n	C_n	D_n	C_{n+1}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



2、 $N < K$

例 用8—1MUX实现 $Y = X_1 + X_0$

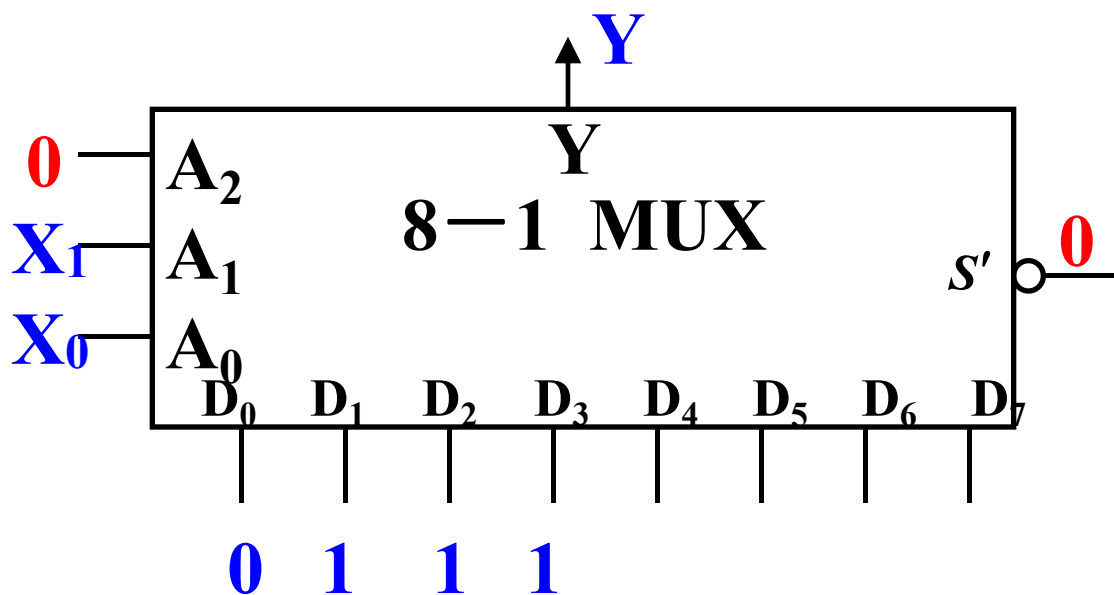
- 1) 列函数Y的真值表
- 2) 对比Y函数真值表和8—1MUX功能表
- 3) 连接对应输入输出信号
- 4) 给控制信号S及多余信号A2赋正确的值

Y函数真值表

$X_1 X_0$	Y
0 0	0
0 1	1
1 0	1
1 1	1

8-1MUX功能表

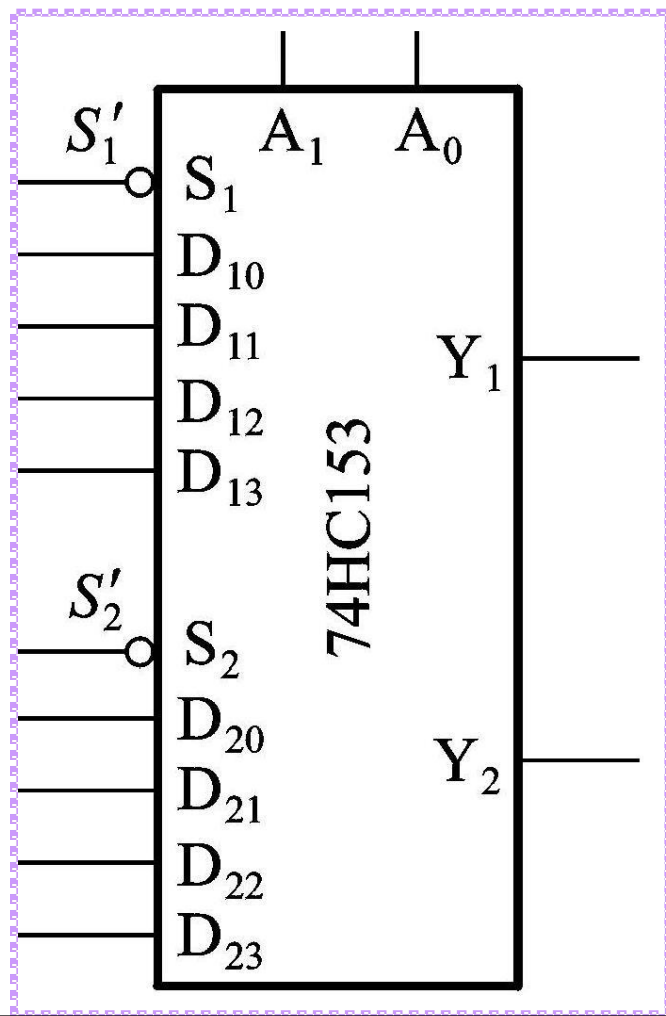
$A_2 A_1 A_0$	Y
0 0 0	D_0
0 0 1	D_1
0 1 0	D_2
0 1 1	D_3
1 0 0	D_4
1 0 1	D_5
1 1 0	D_6
1 1 1	D_7



3、 $N > K$ 有两种方法：扩展法和降维法。

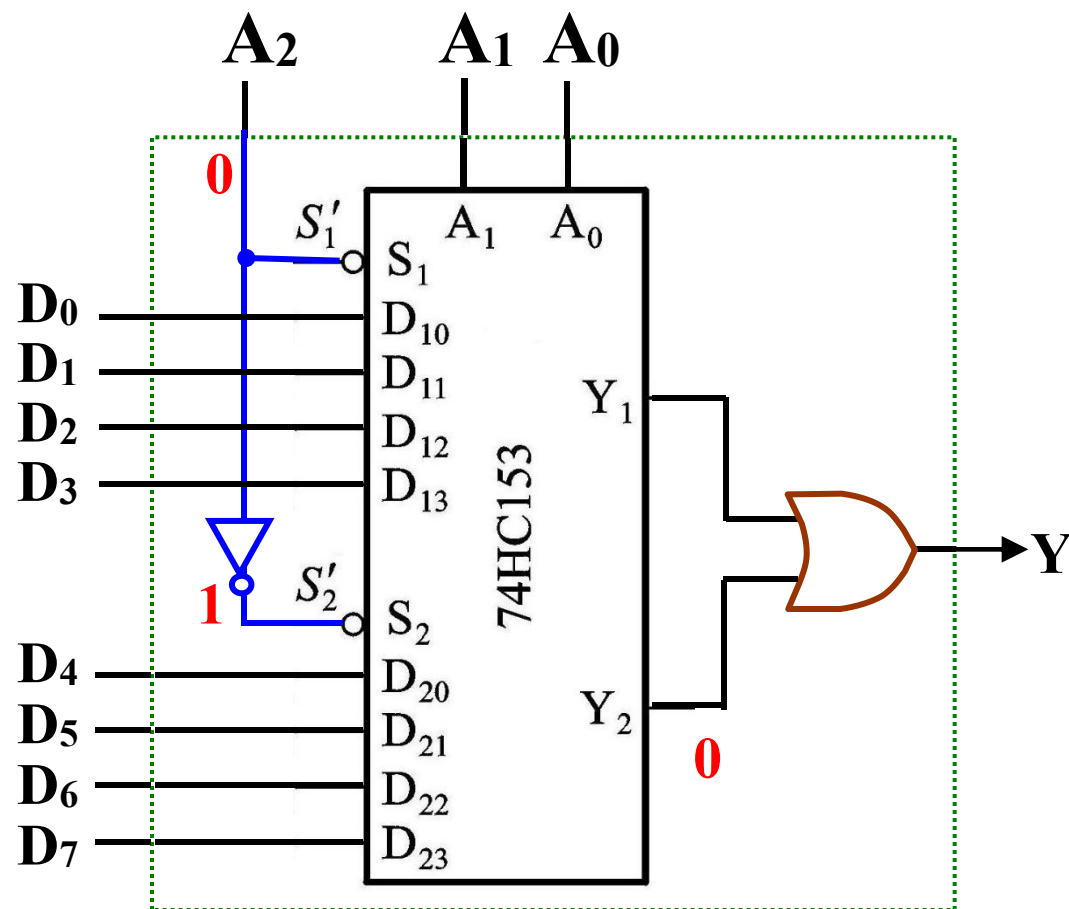
(1) 扩展法

实用芯片 **74HC153**, 双4-1选通器(4-1MUX)



- ◆ 公共的地址输入端(A_1A_0)
- ◆ 独立的数据输入和输出端

例 用两个“4选1”接成一个“8选1”



8-1MUX 功能表

$A_2 A_1 A_0$	Y
0	
0 0 0	D_0
0 0 1	D_1
0 1 0	D_2
0 1 1	D_3
1 0 0	D_4
1 0 1	D_5
1 1 0	D_6
1 1 1	D_7

$$Y = (\overline{A_2} \overline{A_1} \overline{A_0}) D_0 + (\overline{A_2} \overline{A_1} A_0) D_1 + (\overline{A_2} A_1 \overline{A_0}) D_2 + (\overline{A_2} A_1 A_0) D_3 \\ + (A_2 \overline{A_1} \overline{A_0}) D_4 + (A_2 \overline{A_1} A_0) D_5 + (A_2 A_1 \overline{A_0}) D_6 + (A_2 A_1 A_0) D_7$$

(2) 降维法（引入变量卡诺图）

一个逻辑函数卡诺图的变量数称为卡诺图的维数。如果把某些变量也作为卡诺图小方格内的值，则会减少卡诺图的维数，这种卡诺图称为降维卡诺图。

A	BC			
	00	01	11	10
0	1	0	0	0
1	1	1	1	0



A	B	
	0	1
0	C'	0
1	1	C

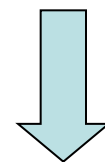
三维降两维

CD		00	01	11	10
AB	00	0	0	0	0
	01	0	0	1	1
	11	1	0	1	0
	10	0	1	1	1



四维降三维

BC		00	01	11	10
A	0				
	1				



三维降二维

B		0	1
A	0		
	1		

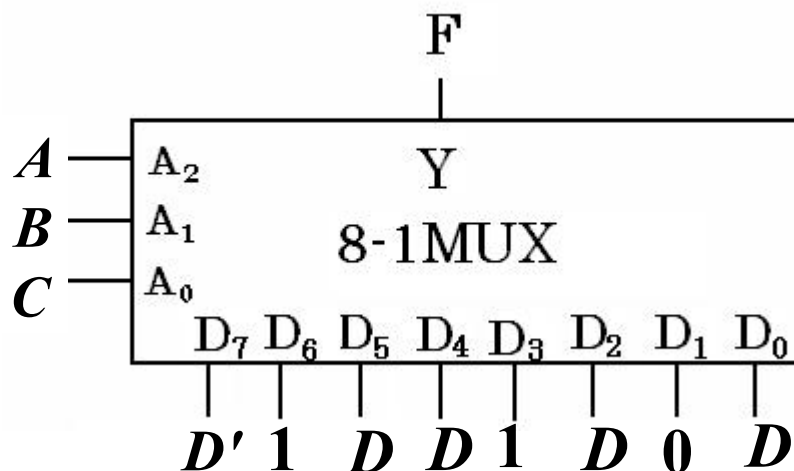
例：用一个8-1MUX实现

$$F(A,B,C,D) = \sum m(1,5,6,7,9,11,12,13,14)$$

$\begin{array}{c} CD \\ \backslash AB \end{array}$	00	01	11	10
00	0	1	0	0
01	0	1	1	1
11	1	1	0	1
10	0	1	1	0

$\begin{array}{c} BC \\ \backslash A \end{array}$	00	01	11	10
0	D	0	1	D
1	D	D	D'	1

如果用4-1MUX实现呢？



用一片8-1 MUX 实现 $Y(A,B,C,D) = \sum m(1,5,6,7,9,11,12,13,14)$,
 A2A1A0已经连接好, 要求填写D0~D7的输入端

1) Y函数K图

\overline{CD} AB		00	01	11	10
		00	01	11	10
00	00	0	1	0	0
01	01	0	1	1	1
11	11	1	1	0	1
10	10	0	1	1	0

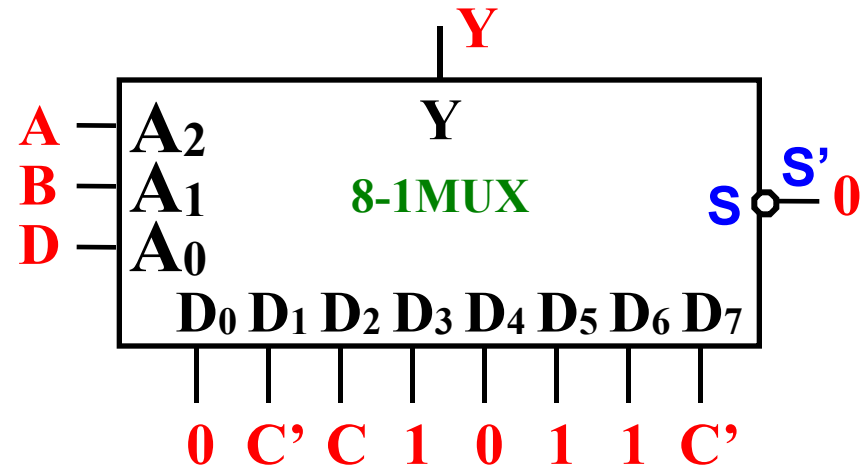
Y

2) Y函数 K图降维

$\overline{A_0}$ $A_2 A_1$ AB		0	1
		00	01
00	00	0	C'
01	01	C	1
11	11	1	C'
10	10	0	1

Y

3) 对应8-1MUX的K图, 连接信号线

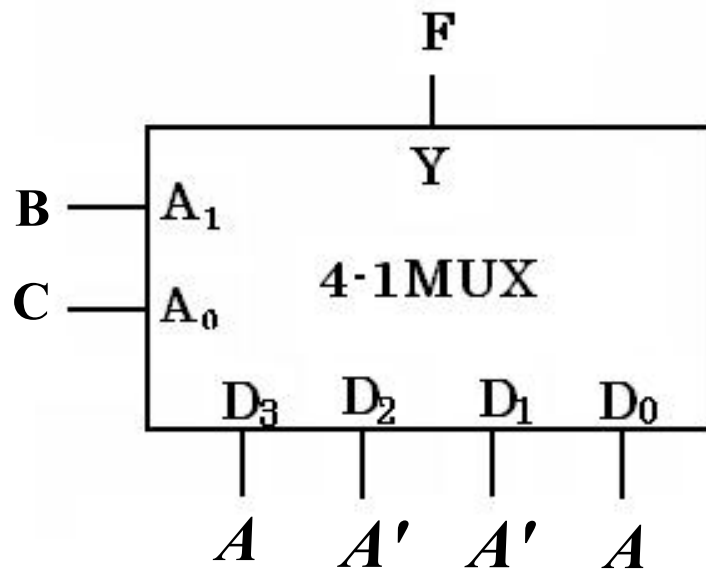
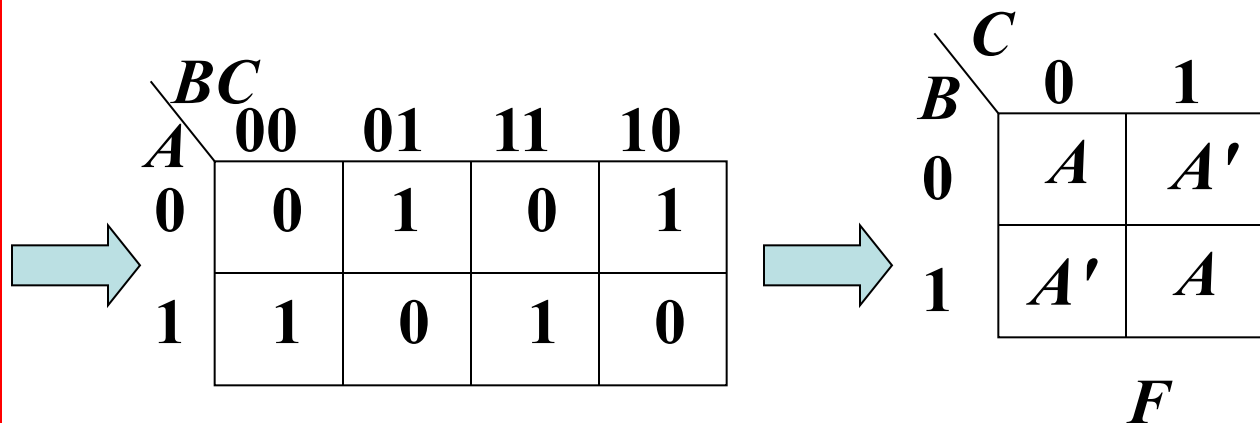


$A \rightarrow A_2$	$0 \rightarrow D_0$
$B \rightarrow A_1$	$C' \rightarrow D_1$
$D \rightarrow A_0$	$C \rightarrow D_2$
$Y \rightarrow Y$	$1 \rightarrow D_3$
	$0 \rightarrow D_4$
	$1 \rightarrow D_5$
	$1 \rightarrow D_6$
	$C' \rightarrow D_7$

例：用4-1MUX实现 $F(A,B,C)$ ，其真值表已知。

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

利用卡诺图降维法



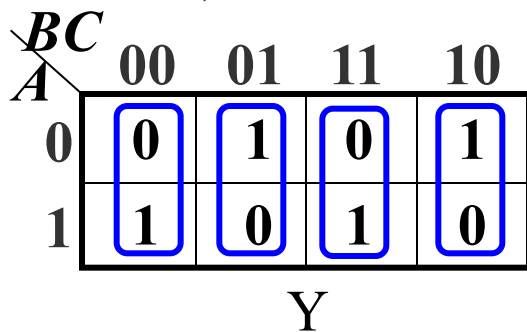
用一片4-1 MUX 实现函数Y(A,B,C)，真值表已知

Y函数
真值表

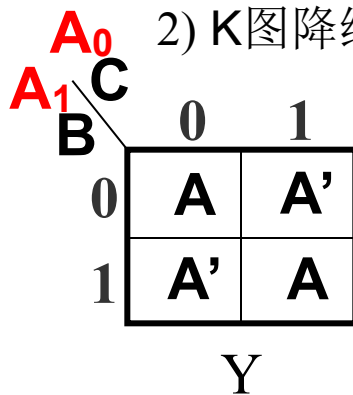
ABC	Y
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

方法I

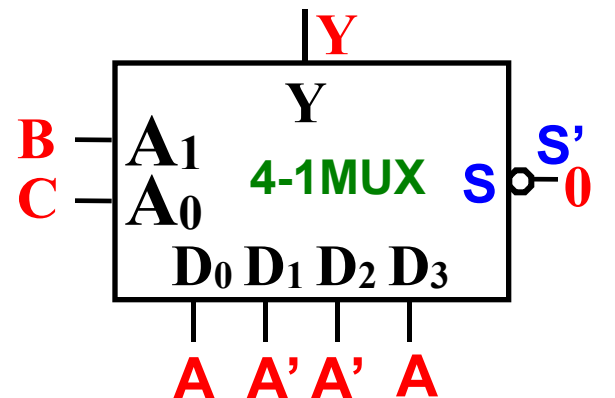
1) Y函数K图



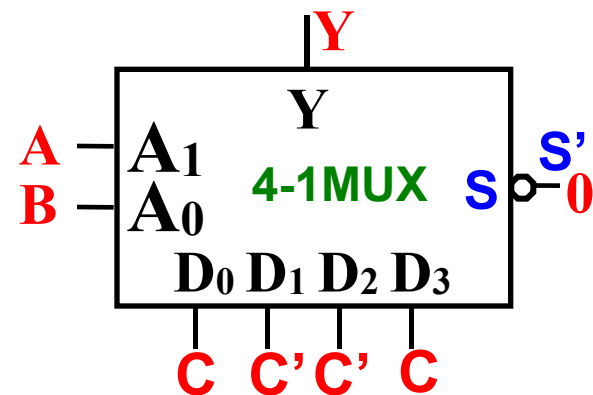
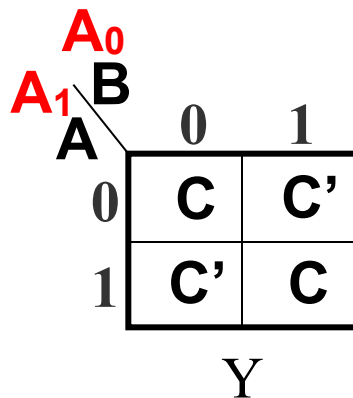
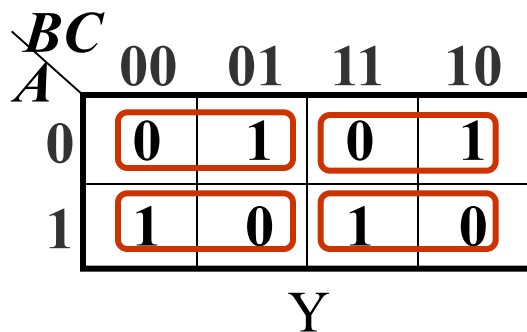
2) K图降维



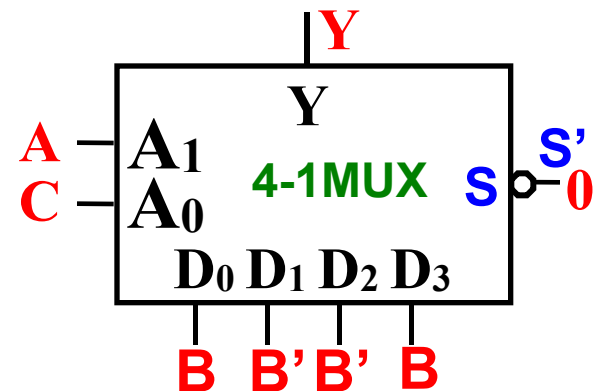
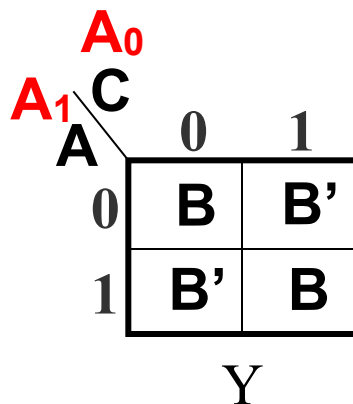
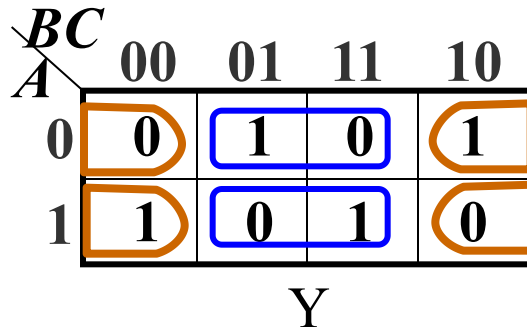
3) 对应连接信号线



方法II

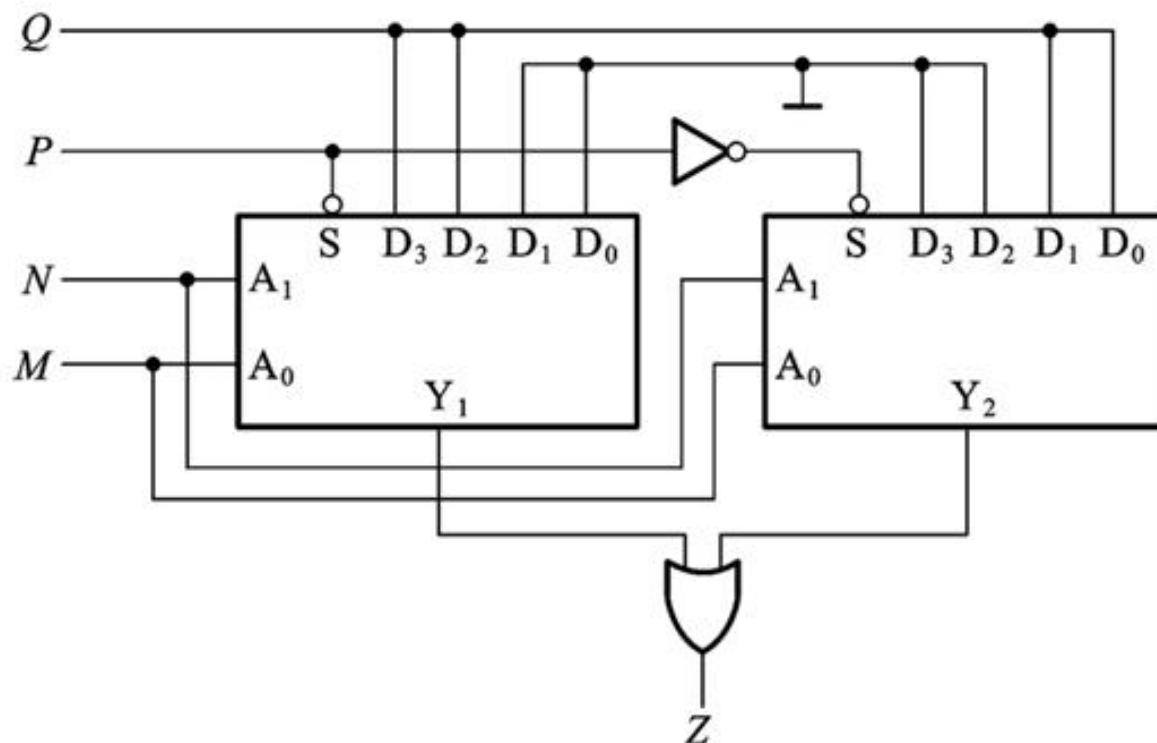


方法III

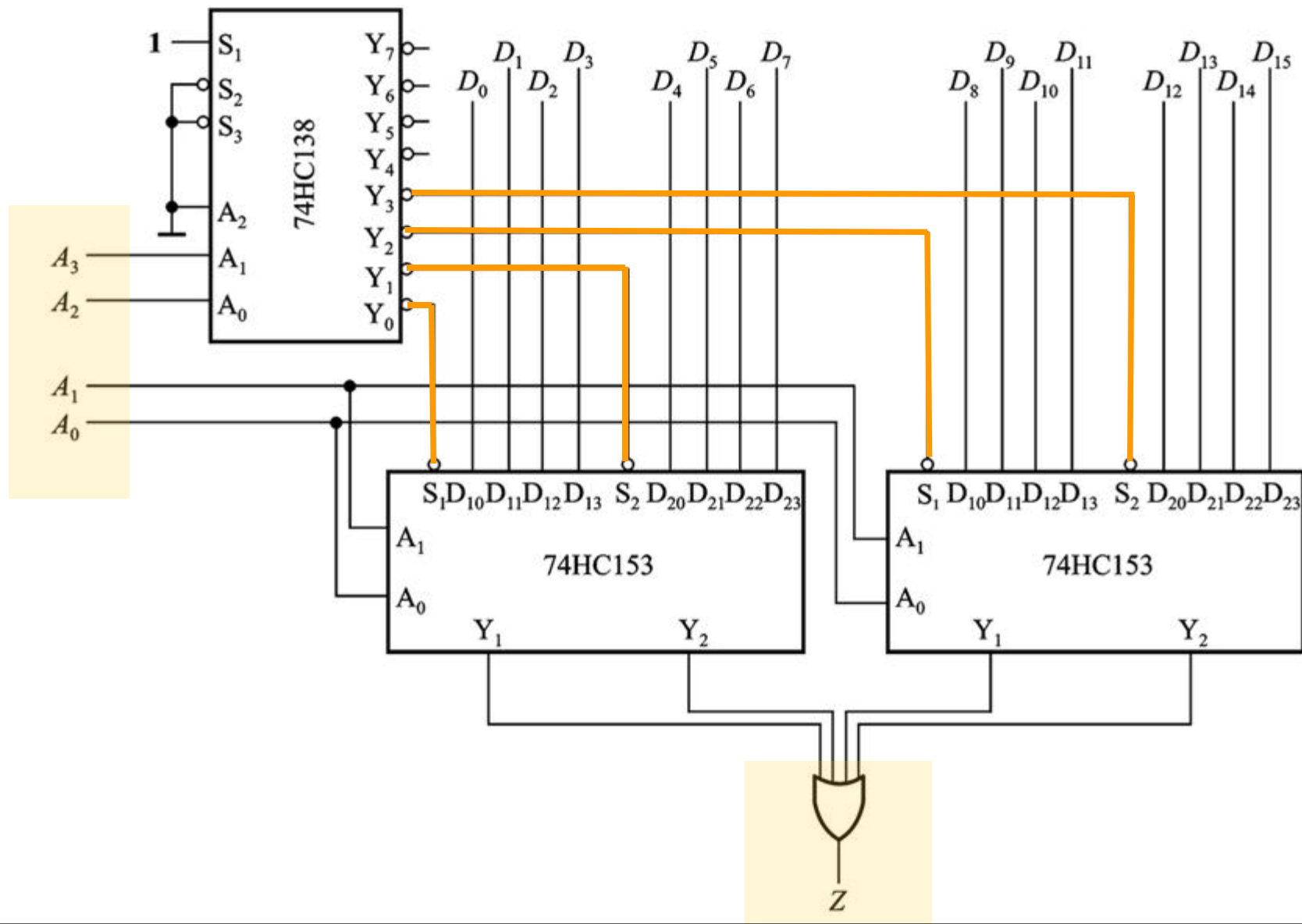


讨论： 图示是用两个4选1数据选择器组成的逻辑电路，
试写出输出端Z与输入M、N、P、Q之间的逻辑
函数式。已知数据选择器的逻辑函数式为：

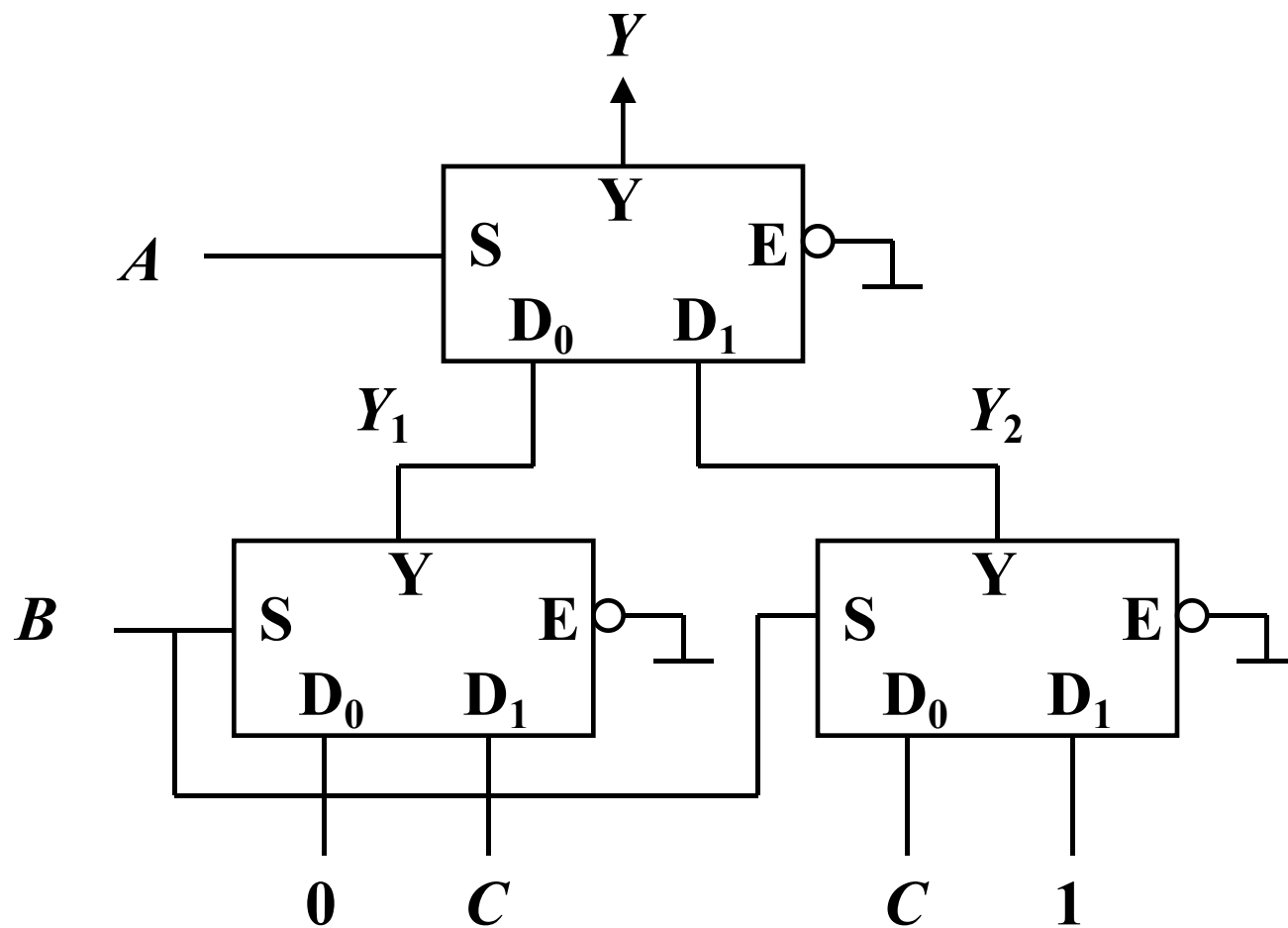
$$Y = (D_0 A_1' A_0' + D_1 A_1' A_0 + D_2 A_1 A_0' + D_3 A_1 A_0)$$



讨论 试用两片双4选1数据选择器74HC153和3线—8线译码器74HC138接成16选1的数据选择器。



讨论：图示电路是由三个**2—1MUX**组成的电路，试分析其逻辑功能。



4.3.4 加法器

主要要求:

- 掌握加法器的逻辑功能及应用。

一、半加器和全加器

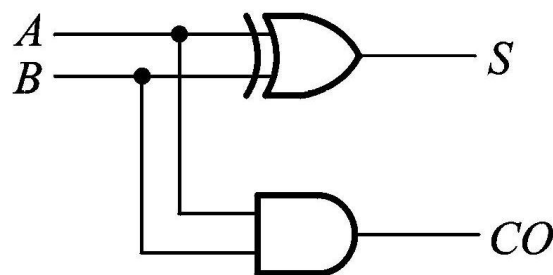
● 半加器

Half Adder, 简称 HA。它只将两个 1 位二进制数相加，而不考虑低位来的进位。

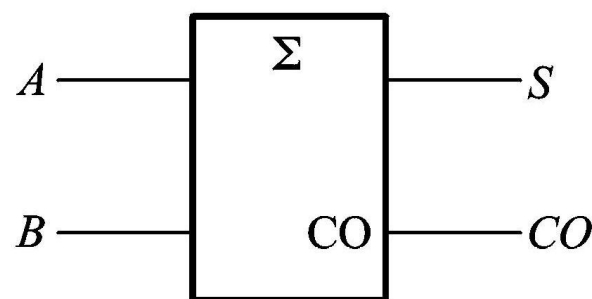
输 入		输 出	
A	B	S	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = AB' + A'B = A \oplus B$$

$$CO = AB$$



(a)



(b)

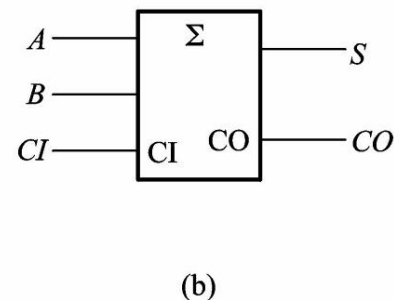
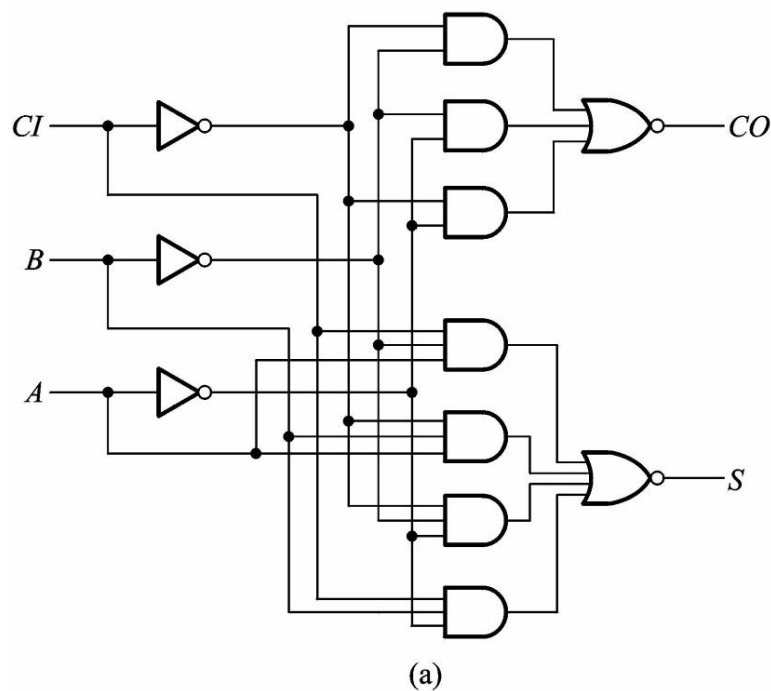
● 全加器

Full Adder, 简称FA。能将本位的两个二进制数和相邻低位来的进位数进行相加。

输 入			输 出	
<i>A</i>	<i>B</i>	<i>CI</i>	<i>S</i>	<i>CO</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

$$S = (A'B'CI' + A'B \cdot CI + AB'CI + ABCI')$$

$$CO = (A'B' + B'CI' + A'CI')$$



二、多位加法器

串行进位加法器

其低位进位输出端依次连至相邻高位的进位输入端，最低位进位输入端接地。因此，高位数的相加必须等到低位运算完成后才能进行，这种进位方式称为串行进位。运算速度较慢。

超前进位加法器

其进位数直接由加数、被加数和最低位进位数形成。各位运算并行进行。运算速度快。



1. 四位串行进位加法器

$A_3 A_2 A_1 A_0$

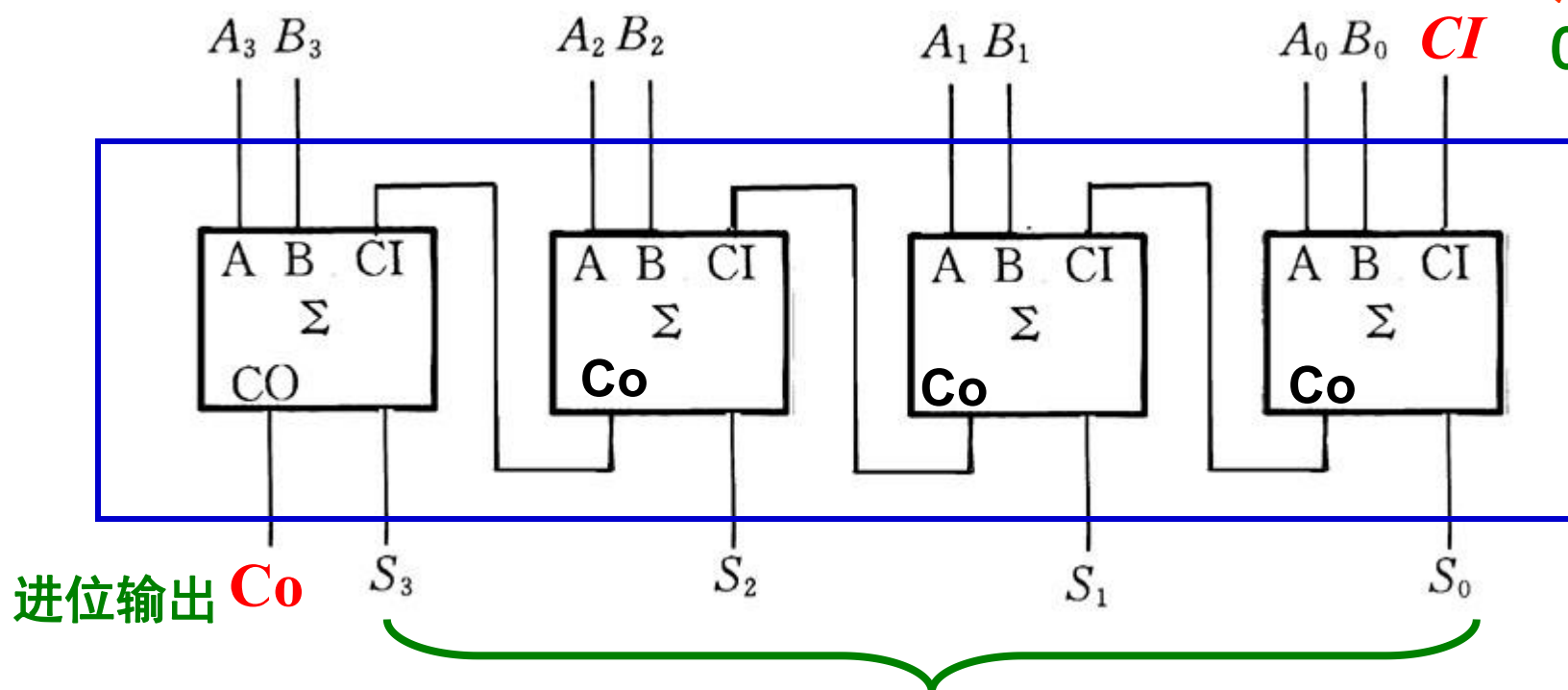
$B_3 B_2 B_1 B_0$

$C_o S_3 S_2 S_1 S_0$

构成：把 n 位全加器串联起来，低位全加器的进位输出连接到相邻的高位全加器的进位输入。

注意：

$CI_0=0$



优点：简单
缺点：慢

和

4位串行进位加法器

2. 超前进位加法器

直接形成进位

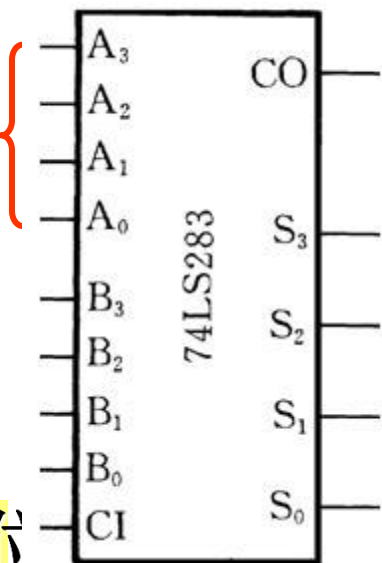
加数

进位输出

加数

和

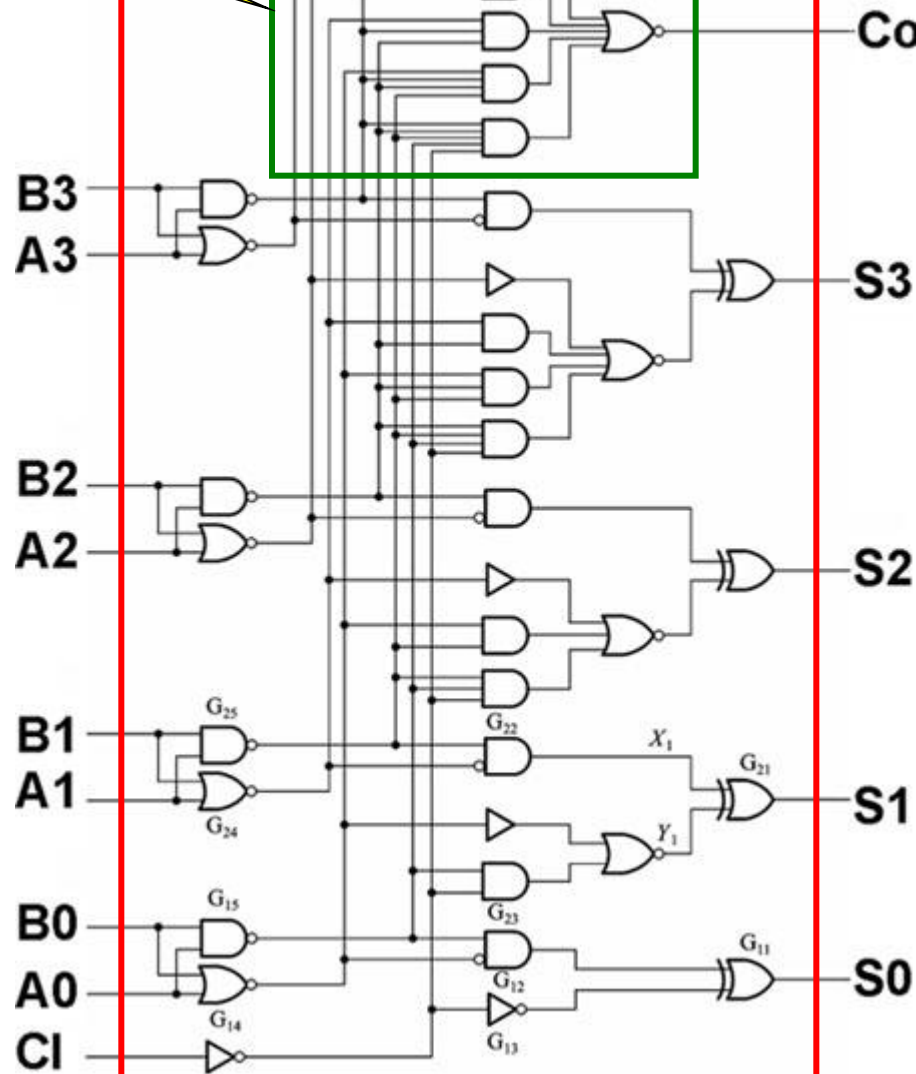
低位进位



4位超前进位加法器
74LS283

优点：快，每一位的**和**及最后的**进位**基本同时产生。

缺点：电路复杂



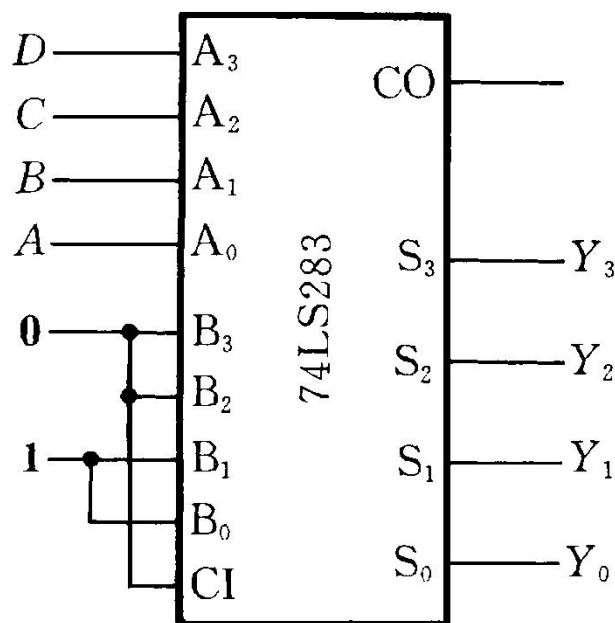
74LS283



用加法器设计组合逻辑电路

例：将BCD8421码转换为余3码。

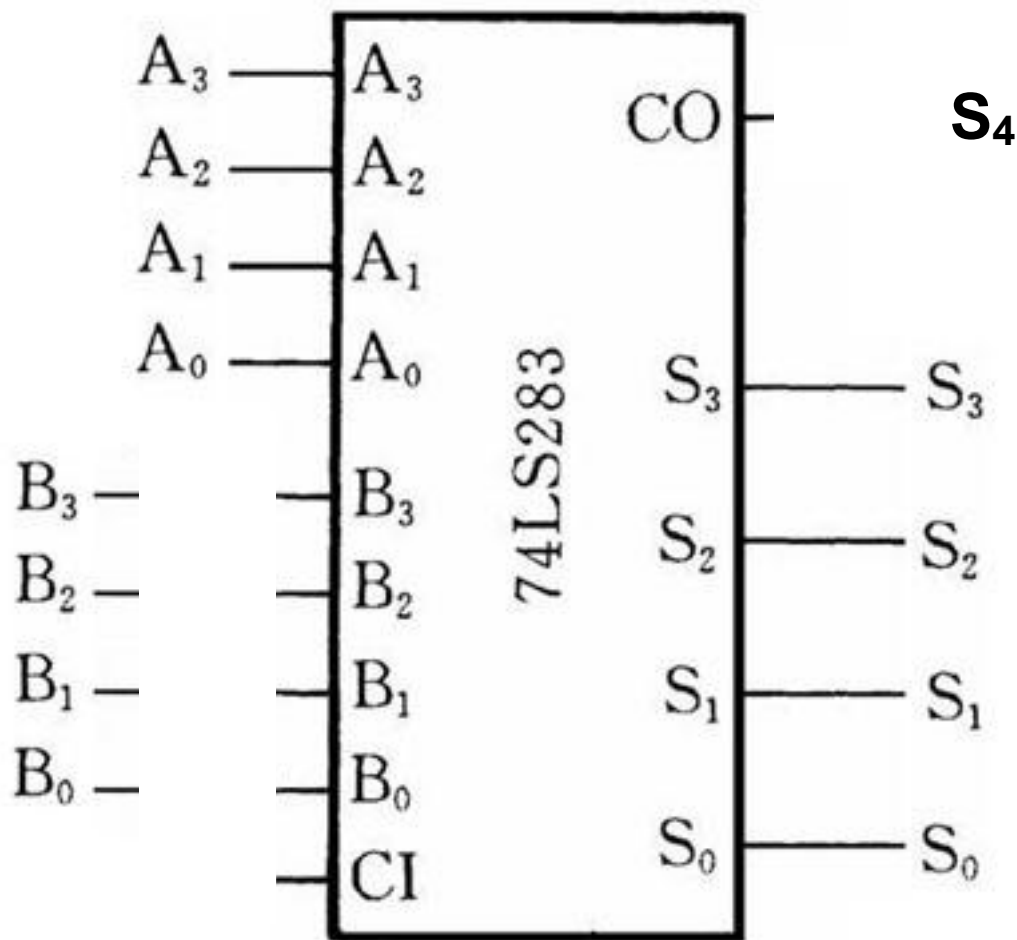
$$Y_3Y_2Y_1Y_0 = DCBA + 0011$$



输 入				输 出			
D	C	B	A	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

例：用74LS283设计四位减法器。

$$A - B = C \longrightarrow A + (-B) = C \longrightarrow A_{\text{补}} + (-B)_{\text{补}} = C_{\text{补}}$$



5位

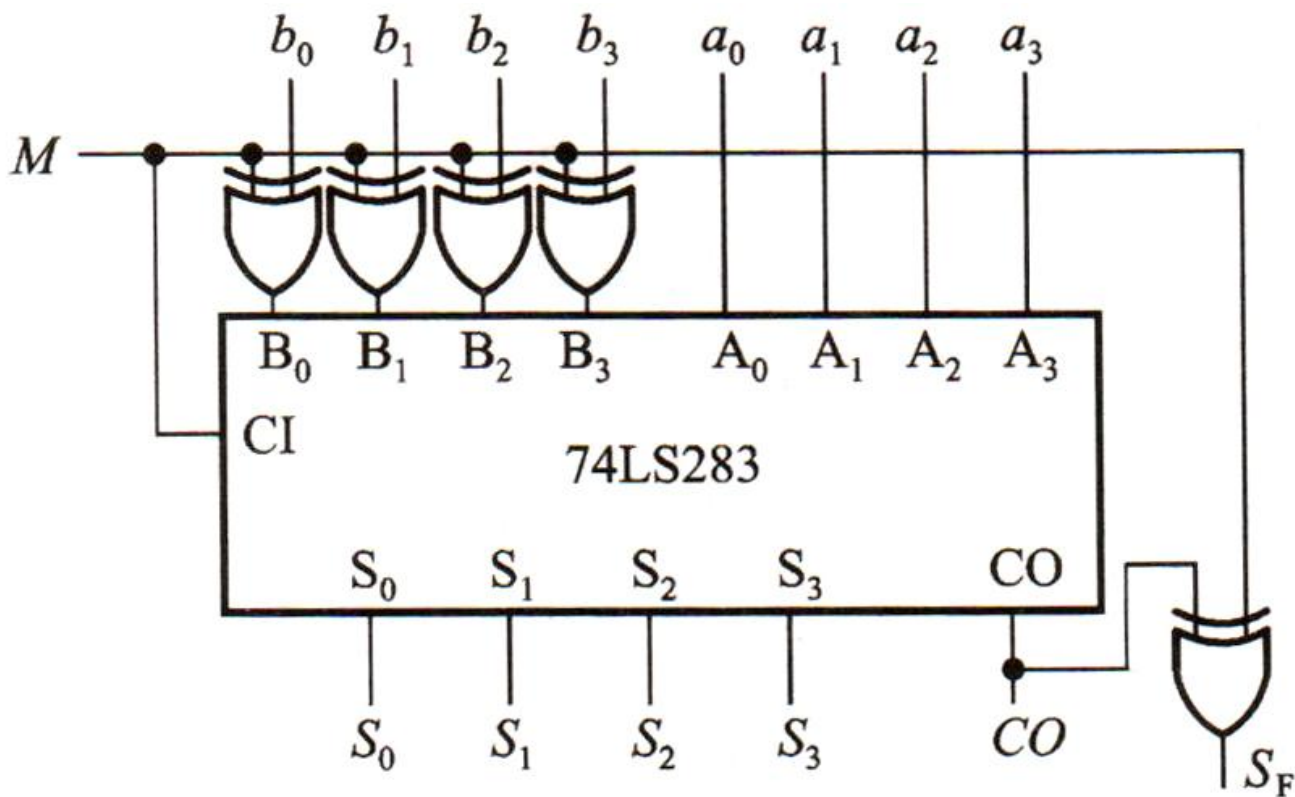
$$\begin{array}{r}
 14 \xrightarrow{\text{补}} 0,1110 \\
 - 11 \xrightarrow{\text{补}} 1,0101 \\
 \hline
 3 \xrightarrow{\text{补}} (1)0,0011 \\
 \hline
 \end{array}$$

$S_4 S_3 S_2 S_1 S_0$

$$\begin{array}{r}
 2 \xrightarrow{\text{补}} 0,0010 \\
 - 11 \xrightarrow{\text{补}} 1,0101 \\
 \hline
 - 9 \xrightarrow{\text{补}} (1)1,0111 \\
 \hline
 \end{array}$$

$S_4 S_3 S_2 S_1 S_0$

讨论：分析电路的逻辑功能。



$M=0$ 时， $B \oplus 0 = B$ ，电路执行 $A+B$ 运算；当 $M=1$ 时， $B \oplus 1 = B'$ ，电路执行 $A-B = A+B'+1$ 运算。

逻辑功能： 二进制并行加法/减法器

4.3.5 数值比较器

主要要求：

- 理解加法器的逻辑功能及应用。
- 了解数值比较器的作用。

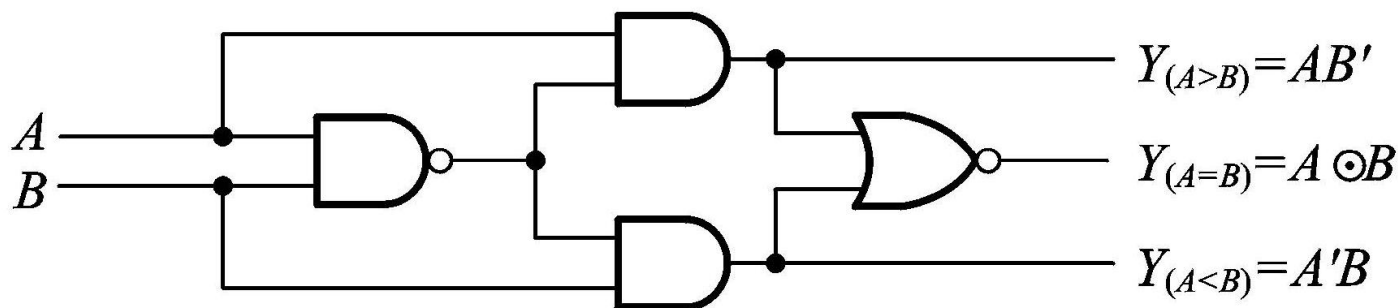
数值比较器

Digital Comparator, 又称数字比较器。用于比较两个数的大小。

(一) 1 位数值比较器

输 入		输 出		
A	B	$Y_{(A>B)}$	$Y_{(A<B)}$	$Y_{(A=B)}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

$$\begin{cases} Y(A > B) = AB' \\ Y(A < B) = A'B \\ Y(A = B) = A'B' + AB = (A \oplus B)' \end{cases}$$



(二) 多位数值比较器

比较原理：从最高位开始逐步向低位进行比较。

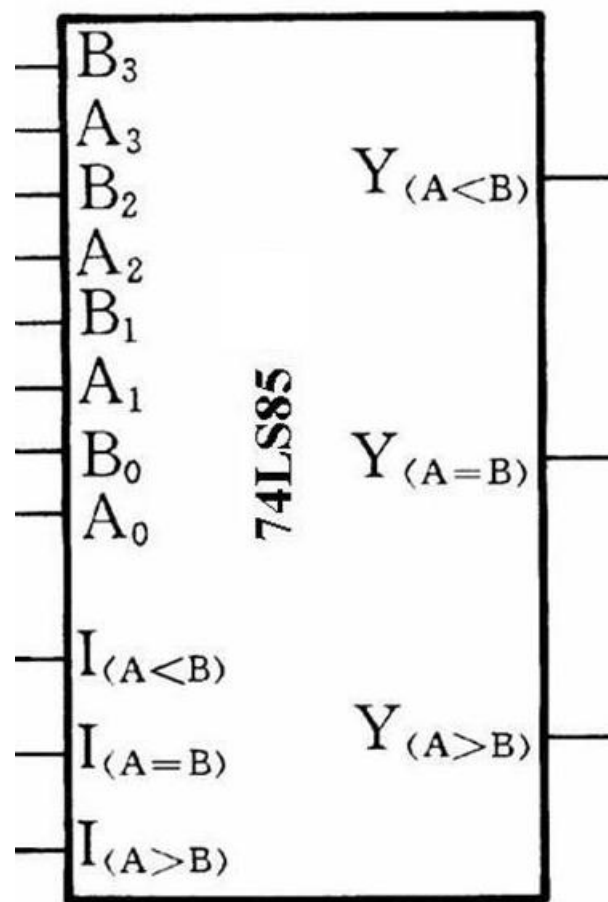
★ 例如 比较 $A = A_3A_2A_1A_0$ 和 $B = B_3B_2B_1B_0$ 的大小：

若 $A_3 > B_3$ ，则 $A > B$ ；若 $A_3 < B_3$ ，则 $A < B$ ；若 $A_3 = B_3$ ，则需比较次高位。

若次高位 $A_2 > B_2$ ，则 $A > B$ ；若 $A_2 < B_2$ ，则 $A < B$ ；若 $A_2 = B_2$ ，则再去比较更低位。

依次类推，直至最低位比较结束。

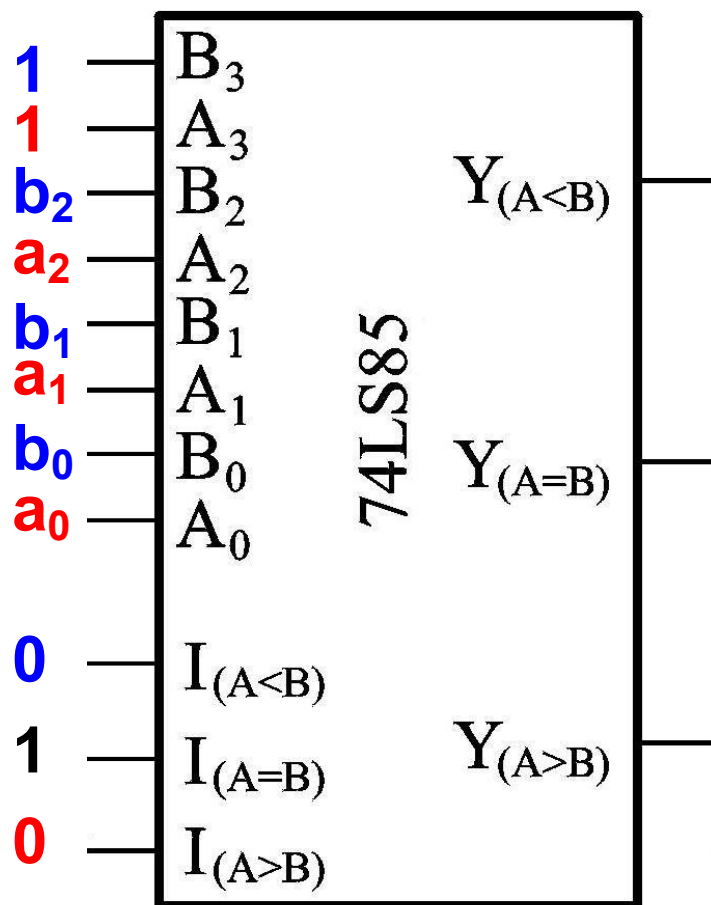
4位数值比较器74LS85的逻辑图



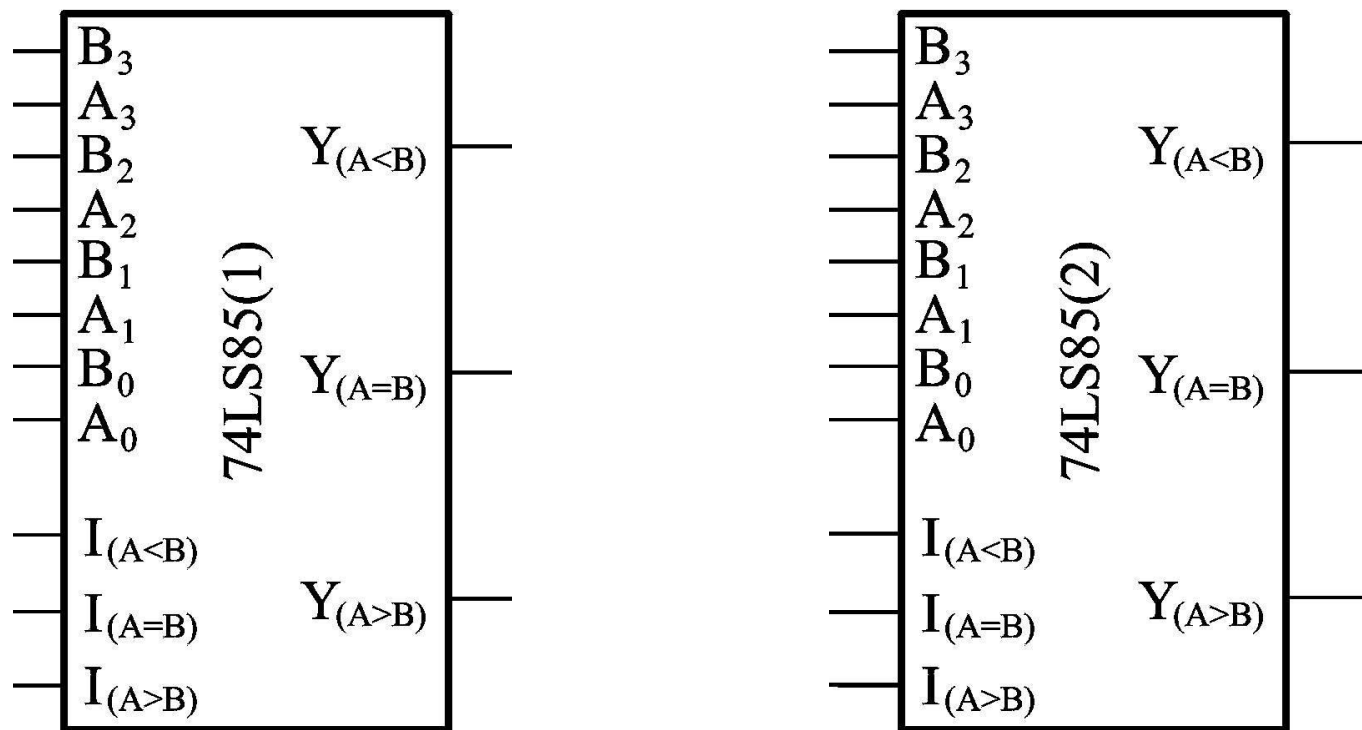
$I_{(A<B)}$, $I_{(A=B)}$ 和 $I_{(A>B)}$
为附加端，用于扩展

$\left. \begin{array}{l} I_{(A<B)} \\ I_{(A=B)} \\ I_{(A>B)} \end{array} \right\}$ 来自低位的比较结果

例：用1片74LS85实现3位数值比较器

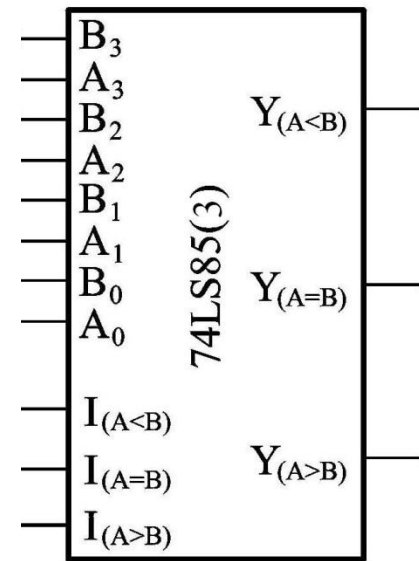
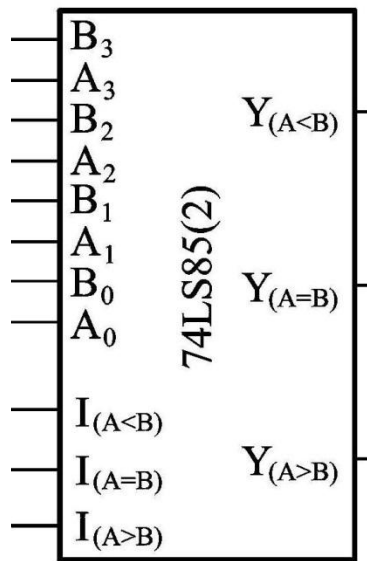
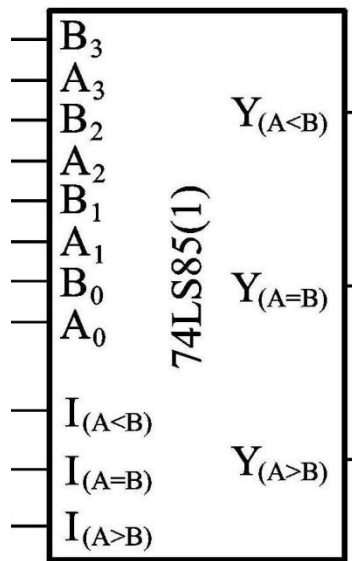


例：用2片74LS85构成8位数值比较器

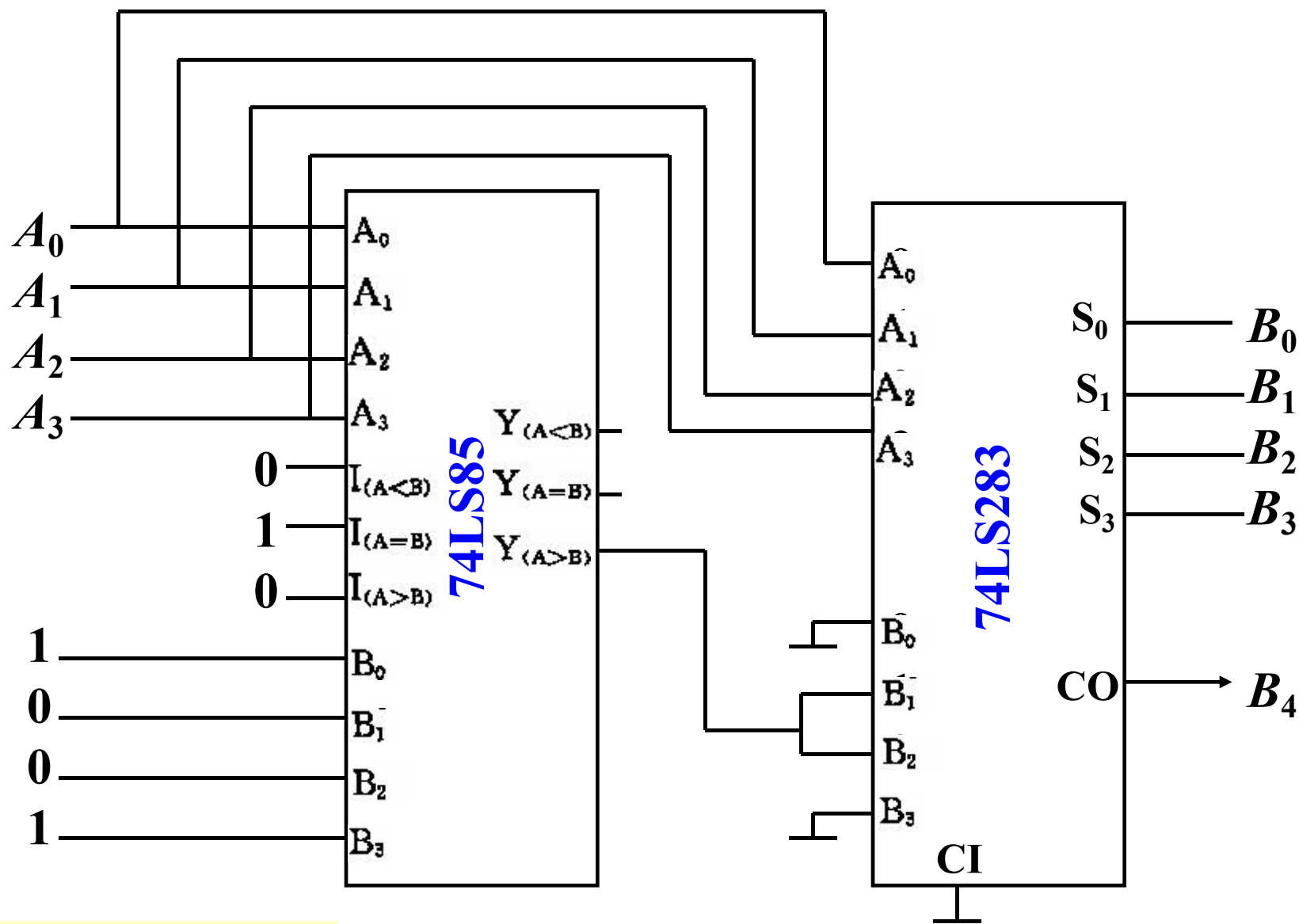


怎样用74LS85组成一个10位数值比较器。

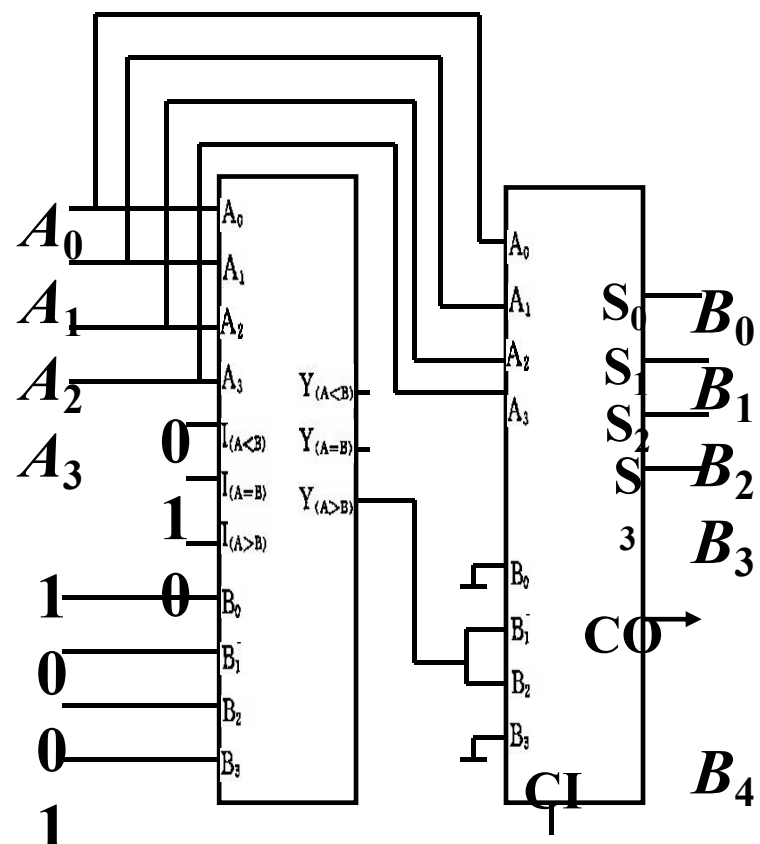
用3片74LS85构成10位数值比较器



讨论：分析下列电路的逻辑功能。



逻辑功能：自然码—BCD8421码转换器



两数之和	二进制结果					期望的BCD结果				
	C ₀₁	S ₁₃	S ₁₂	S ₁₁	S ₁₀	C ₀	S ₃	S ₂	S ₁	S ₀
0	0	0000				0	0000			
1	0	0001				0	0001			
2	0	0010				0	0010			
3	0	0011				0	0011			
4	0	0100				0	0100			
5	0	0101				0	0101			
6	0	0110				0	0110			
7	0	0111				0	0111			
8	0	1000				0	1000			
9	0	1001				0	1001			
10	0	1010				1	0000			
11	0	1011				1	0001			
12	0	1100				1	0010			
13	0	1101				1	0011			
14	0	1110				1	0100			
15	0	1111				1	0101			
16	1	0000				1	0110			
17	1	0001				1	0111			
18	1	0010				1	1000			

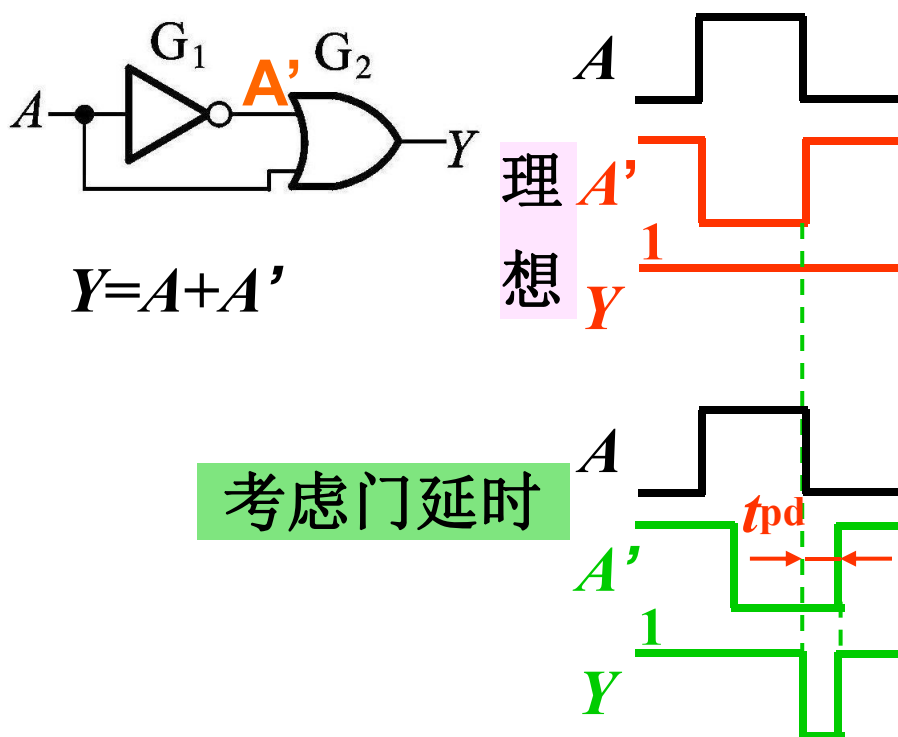
4.4 组合逻辑电路中的竞争冒险

主要要求:

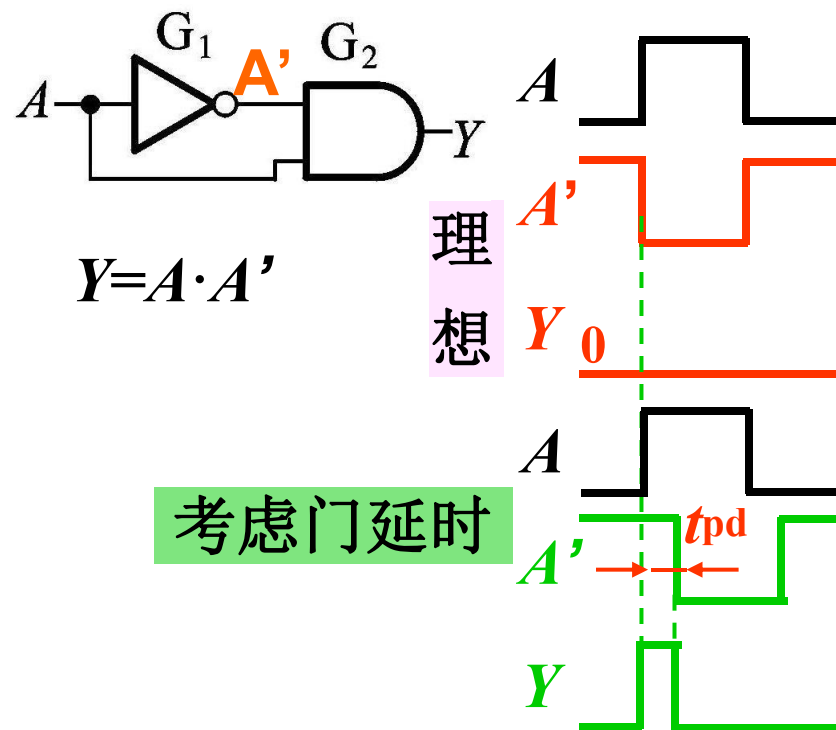
- 了解竞争冒险现象及其产生的原因和消除措施。

4.4.1 竞争—冒险现象及成因

✓ 负尖峰脉冲冒险举例



✓ 正尖峰脉冲冒险举例



可见，在组合逻辑电路中，当一个门电路(如 G_2)输入两个向相反方向变化的信号时，则在输出端可能会产生尖峰干扰脉冲。

4.4.1 竞争—冒险现象及成因

理想状态 { 连线 and 门无延时
输入信号的变化无过渡过程

1. 什么是“竞争”

两个输入“同时向着相反的逻辑电平跳变”，称存在“竞争”。

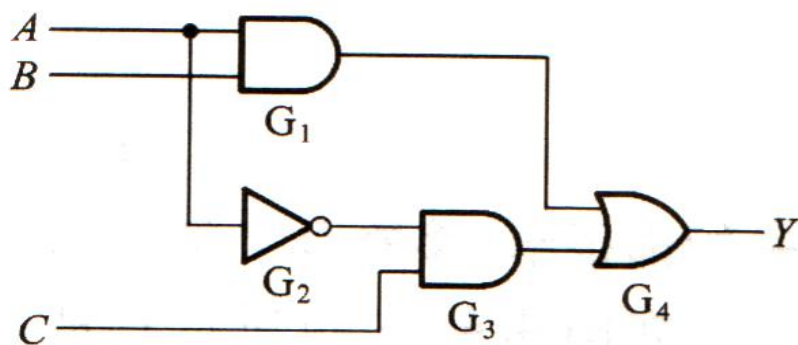
2. 因“竞争”而可能在输出产生尖峰脉冲干扰的现象，称为“竞争-冒险”。

竞争与冒险的判断

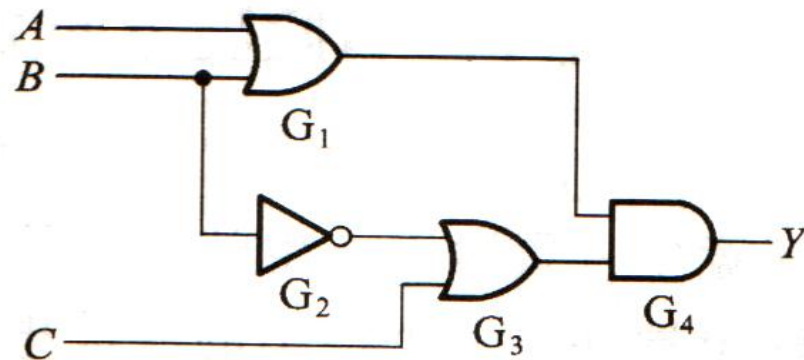
代数法:

当输出端函数在一定条件下可以化简成 $F = X + X'$ 或 $F = X \cdot X'$ 的形式, X 的变化可能会引起冒险。

例：试判断图中电路是否存在竞争—冒险。已知任何瞬间输入变量只可能有一个改变状态。



(a)



(b)

解：图（a）电路 $Y = AB + A'C$ 当 $B=C=1$ 时, $Y = A + A'$

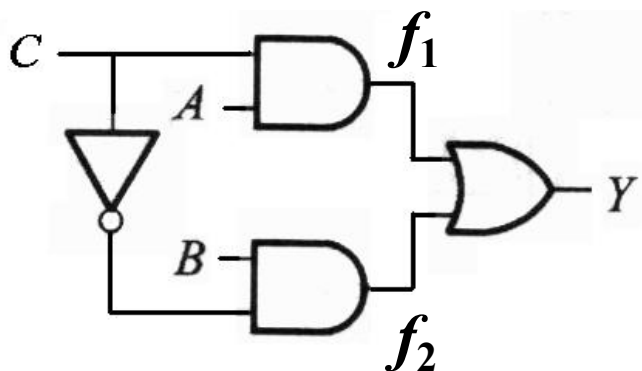
故电路中存在竞争—冒险。

图（b）电路 $Y = (A + B) \cdot (B' + C)$ 当 $A=C=0$ 时, $Y = B \cdot B'$

故电路中也存在竞争—冒险。

卡诺图法:

如果两个卡诺圈相切，而相切处又未被其他卡诺圈包围，则可能发生冒险。当ABC从111变到110时，Y从一个圈跳到另一个圈，函数值可按1—0—1变化，而出现毛刺。



$$Y = f_1 + f_2 = AC + BC'$$

A \ BC				
	00	01	11	10
0	0	0	0	1
1	0	1	1	1

消除的
办法：
增加冗
余项

这两种判断方法局限性都太大。

目前可以通过计算机辅助分析的方法检查出电路是否存在竞争—冒险现象。

另外可以用实验来检查电路是否存在竞争—冒险现象。

练习：

试分析在实现逻辑函数 $F(A,B,C,D) = A'B'D + AB'C' + BCD$

的电路中，是否存在竞争—冒险？在什么时刻出现？如何消除？

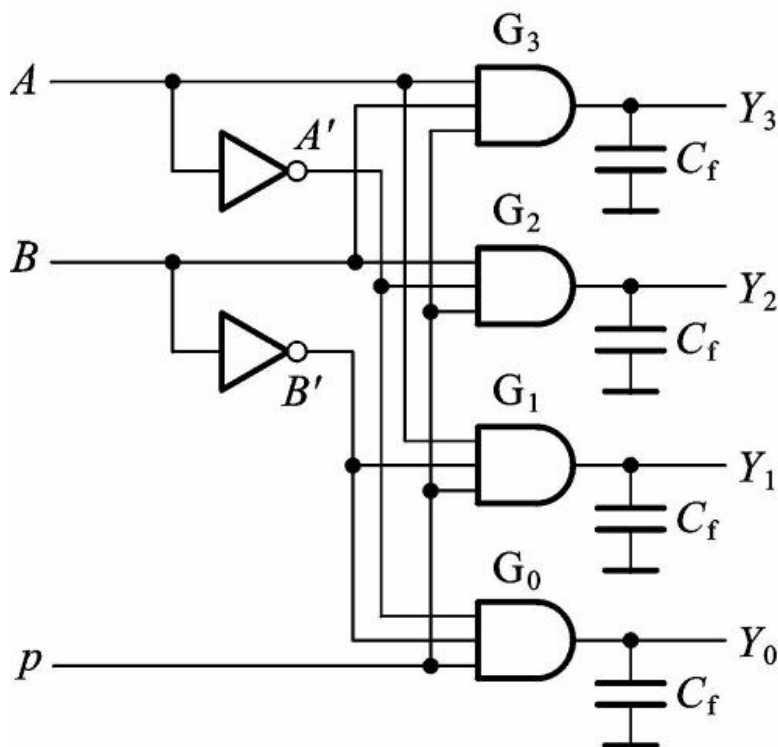
CD		AB			
		00	01	11	10
00	0	1	1	0	
01	0	0	1	0	
11	0	0	1	0	
10	1	1	0	0	

存在竞争冒险，两处。

消除冒险的方法：

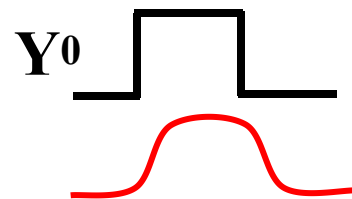
1. 接入滤波电容

由于尖峰干扰脉冲的宽度很窄，在可能产生尖峰干扰脉冲的门电路输出端与地之间接入一个容量为几十皮法的电容就可吸收掉尖峰干扰脉冲。

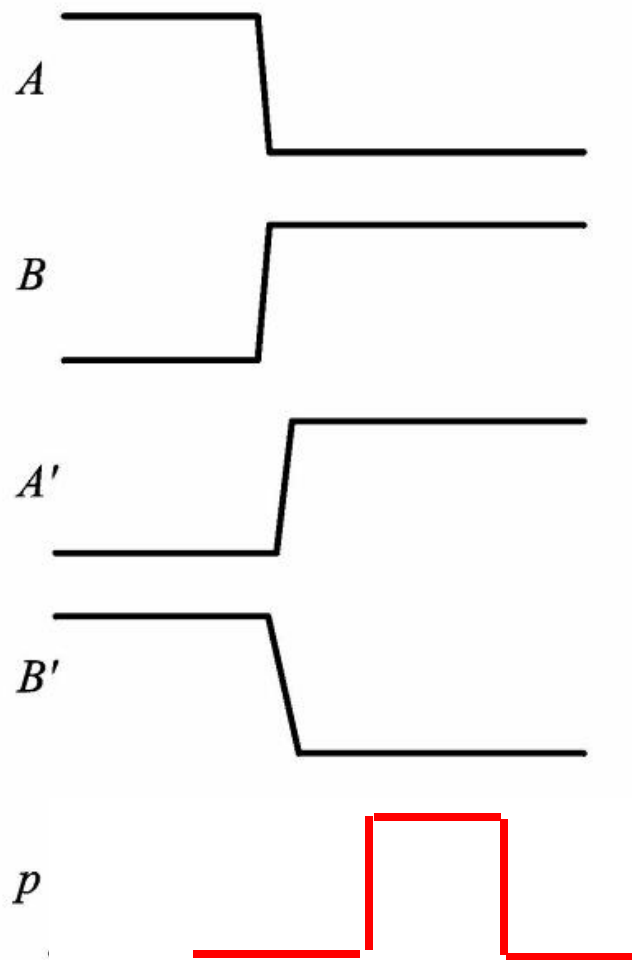
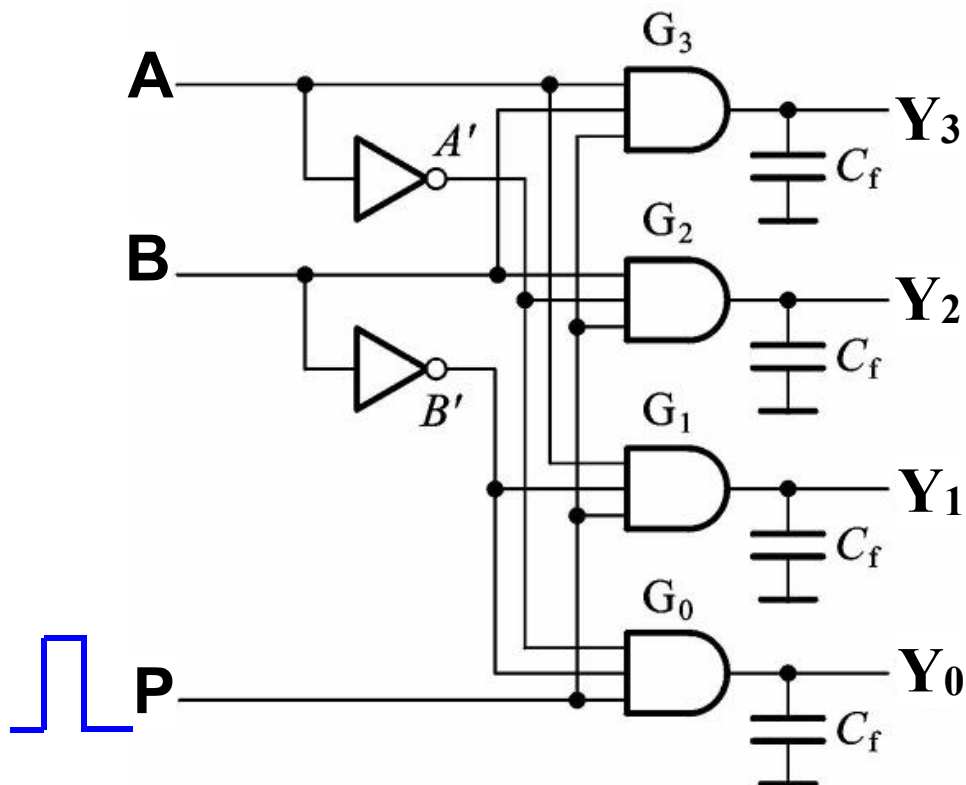


优点：简单易行

缺点：输出电压波形上升时间和下降时间增加，波形变坏



2. 引入选通脉冲



优点：较简单，无需增加元件

缺点：需要产生同步选通脉冲，宽度和时间要求严格

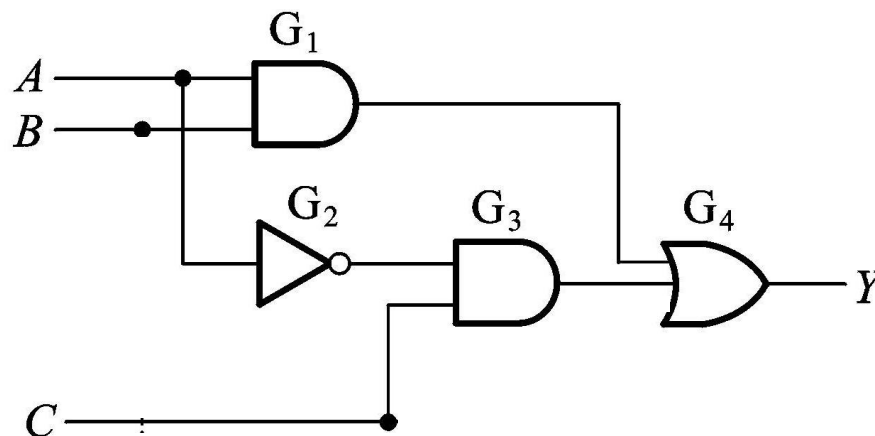
3. 修改逻辑设计, 增加冗余项

$$Y = AB + A'C$$

在 $B = C = 1$ 的条件下, $Y = A + A' \Rightarrow$ 稳态下 $Y = 1$

当 A 改变状态时存在竞争冒险

$$Y = AB + A'C + BC$$



优点: 效果较好

缺点: 增加元件, 只能消除由延时不一致造成的冒险