第2章 组合逻辑

返回目录

2.1 组合逻辑分析

所谓组合逻辑分析，就是根据已知逻辑电路图，找出组合逻辑电路的输入与输出关系，确定在什么样的输入取值组合下对应的输出为 1 。

组合逻辑电路分析的一般过程：

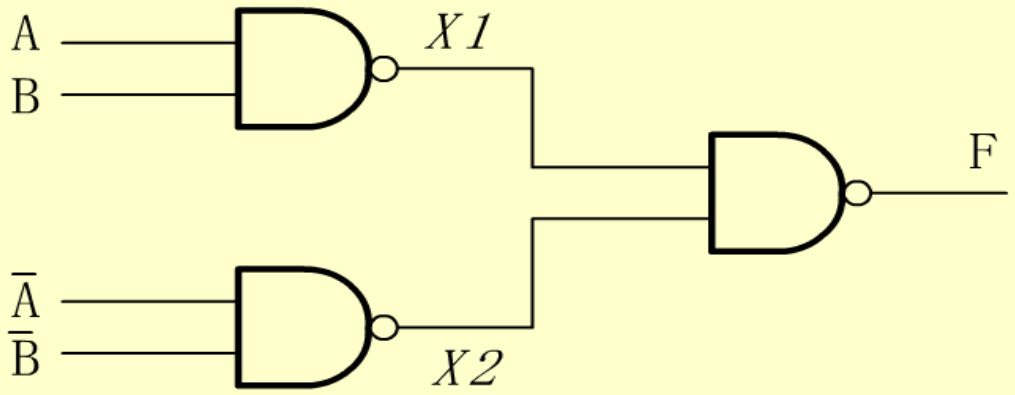
1. 对于较简单的逻辑电路 ----> “逐级电平推导”法
2. 对于较复杂的逻辑电路 ----> “列写布尔表达式”法或“数字波形图法”等
3. 迫不得已的情况下 ----> “列写真值表”法

2.1 组合逻辑分析

2.1.1 逐级电平推导法

先假设输出为逻辑1或0，然后逐级向前推导，直到确定输入的逻辑值。

【例1】分析图中所示的逻辑电路



(a) 原逻辑电路图

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

真值表

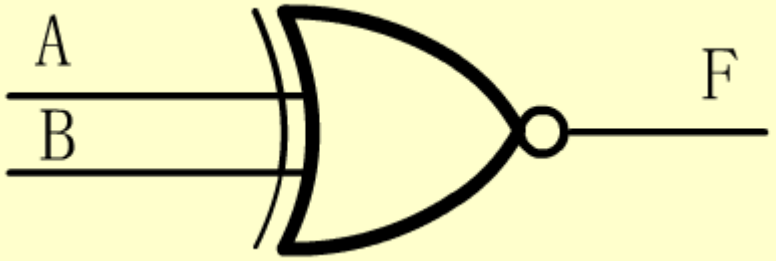
采用逐级电平推导法：

由F=1，知X_1=0或X_2=0

由X_1=0，知A=1，B=1

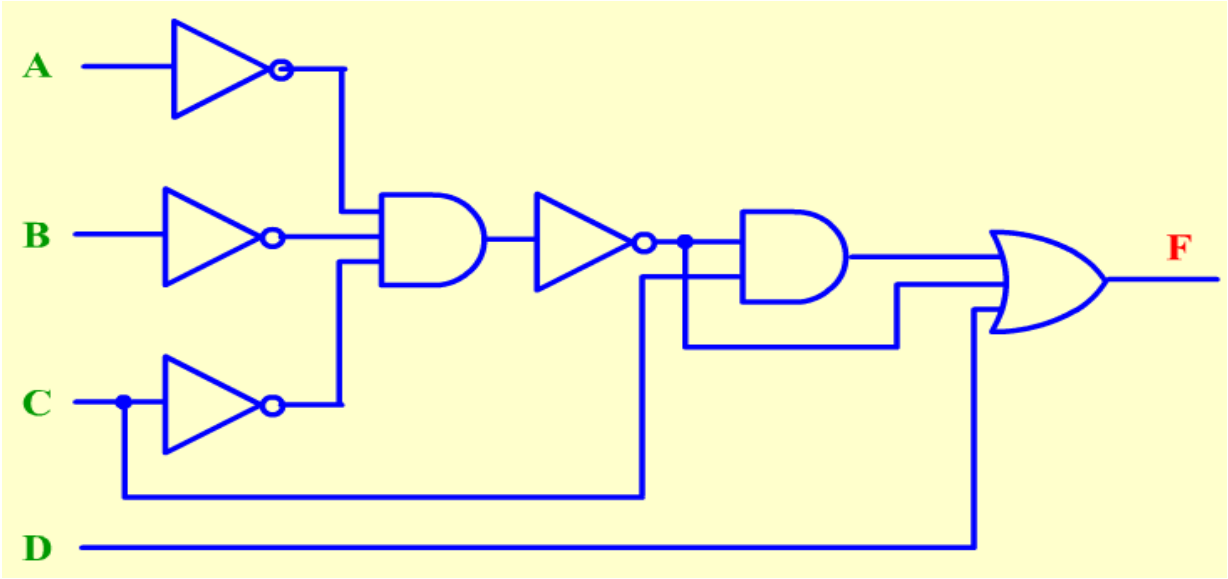
由X_2=0，知$\overline{A}$=1，$\overline{B}$=1

由此可知：当输入量A、 B 都为1或0时，输出F=1。



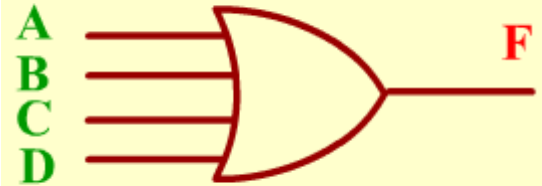
2.1.2 列写布尔表达式法

例2 指出图示电路的逻辑功能



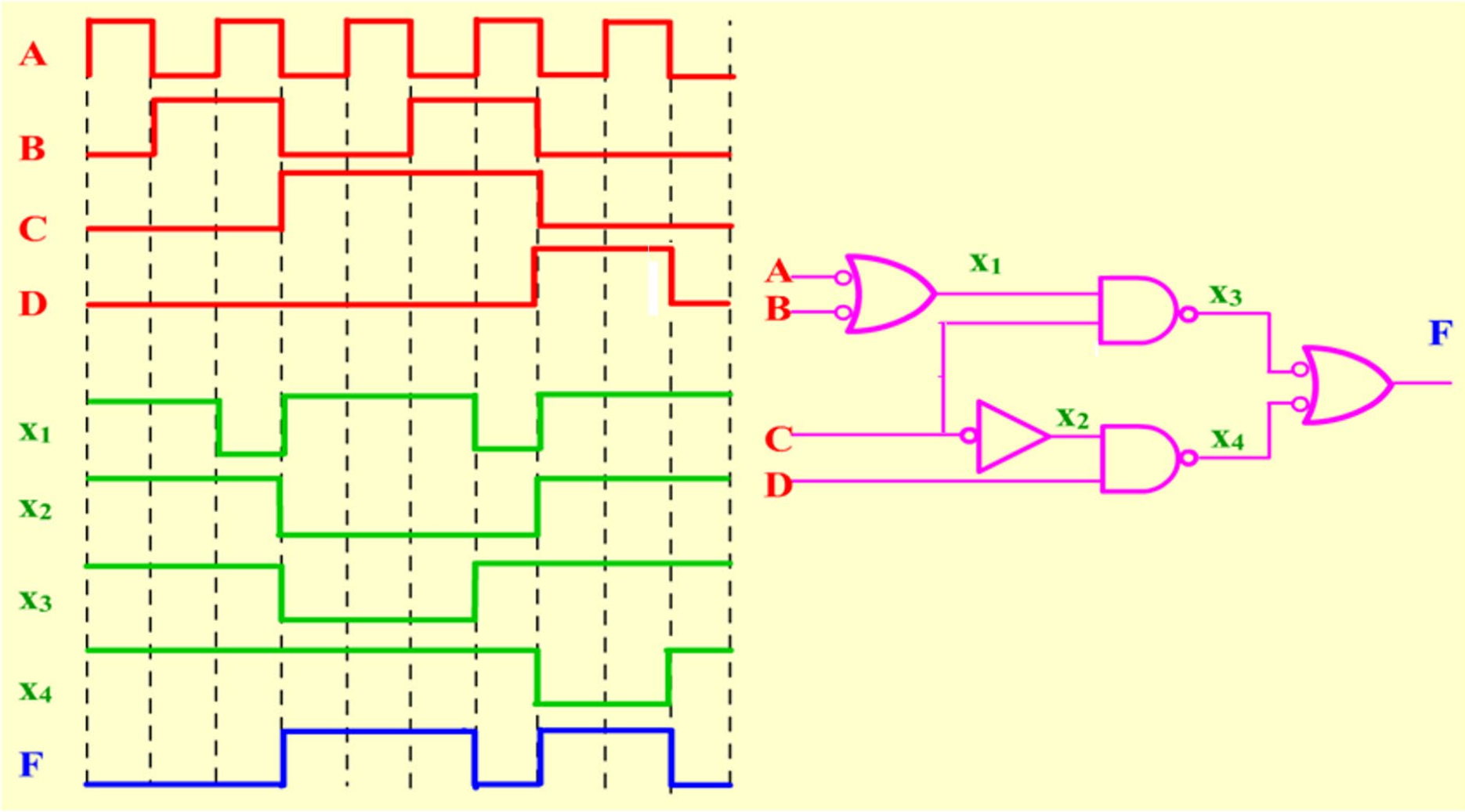
$$F = (\overline{\overline{A}\,\overline{B}\,\overline{C}})C + \overline{\overline{A}\,\overline{B}\,\overline{C}} + D$$

$$\begin{aligned} F &= (\overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}})C + \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} + D \\ &= AC + BC + CC + A + B + C + D \\ &= AC + BC + C + A + B + C + D \\ &= C(A + B + 1) + A + B + D \\ &= A + B + C + D \end{aligned}$$



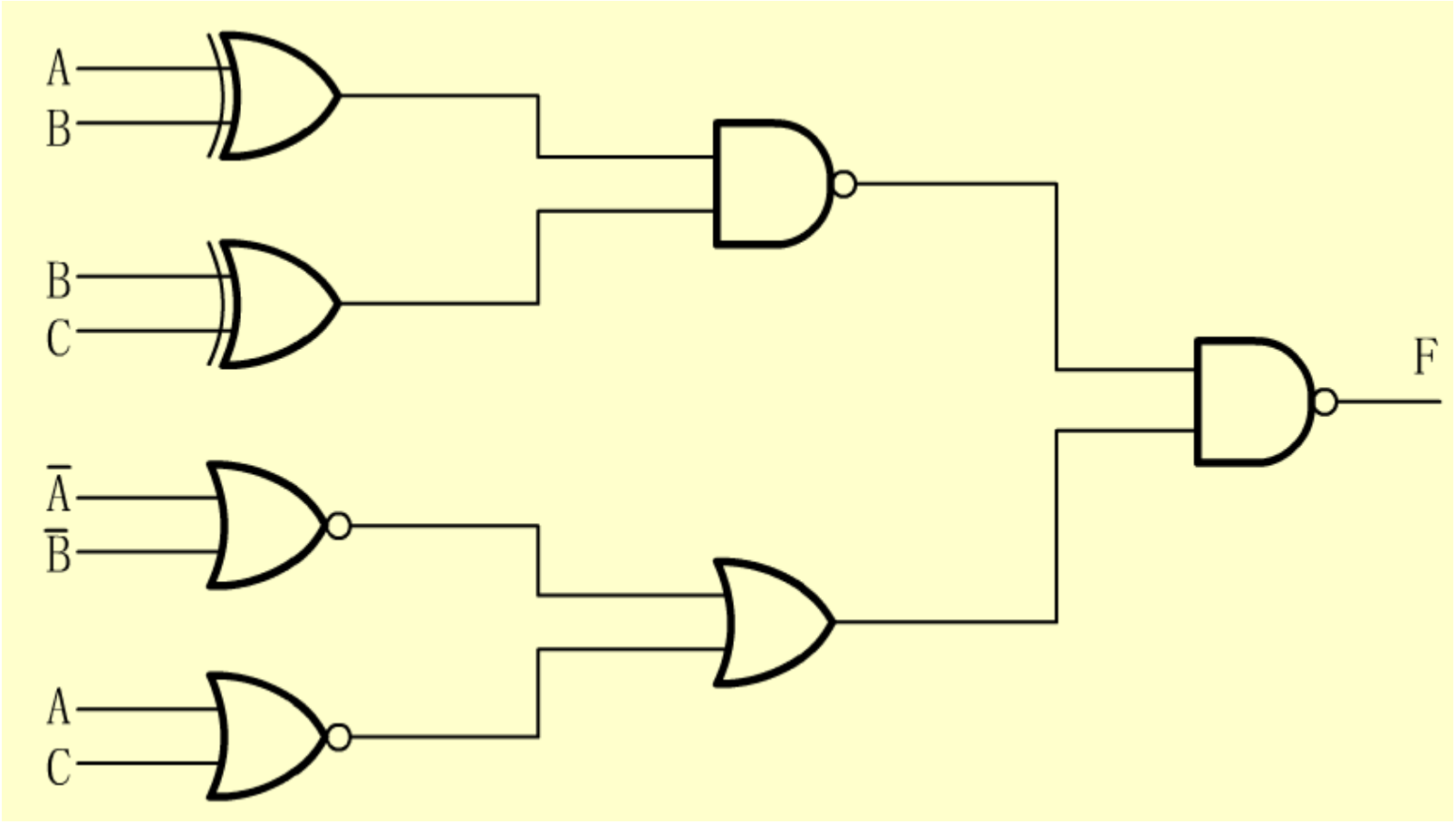
2.1.3 数字波形图分析法

这种方法是对逻辑门的所有输入变量施以输入波形，逐级画出各个门电路的输出波形，乃至画出最后的输出波形。



2.1.4 列写逻辑电路真值表法

【例4】分析图中所示电路的逻辑功能



[解]

该电路比较复杂，一次写出它的输出函数表达式比较困难。故从输入端开始，逐级写出，然后写出总的输出表达式：

$$F = \overline{(A \oplus B) \cdot (B \oplus C)} \cdot \overline{(\overline{A} + \overline{B} + \overline{A + C})}$$

但从这一表达式无法推知该电路的逻辑功能。为此利用布尔代数，对表达式展开并进行化简：

$$\begin{aligned}
F &= (A \oplus B)(B \oplus C) + (\overline{A} + \overline{B})(A + C) \\
&= (A\overline{B} + \overline{A}B)(B\overline{C} + \overline{B}C) + (\overline{A}C + A\overline{B} + \overline{B}C) \\
&= A\overline{B}C + \overline{A}B\overline{C} + \overline{A}C + A\overline{B} + \overline{B}C \\
&= A\overline{B} + \overline{A}(B\overline{C} + C) + \overline{B}C \\
&= A\overline{B} + \overline{A}(B + C) + \overline{B}C \\
&= A\overline{B} + \overline{A}B + \overline{A}C + \overline{B}C \\
&= A \oplus B + \overline{A}C + \overline{B}C
\end{aligned}$$

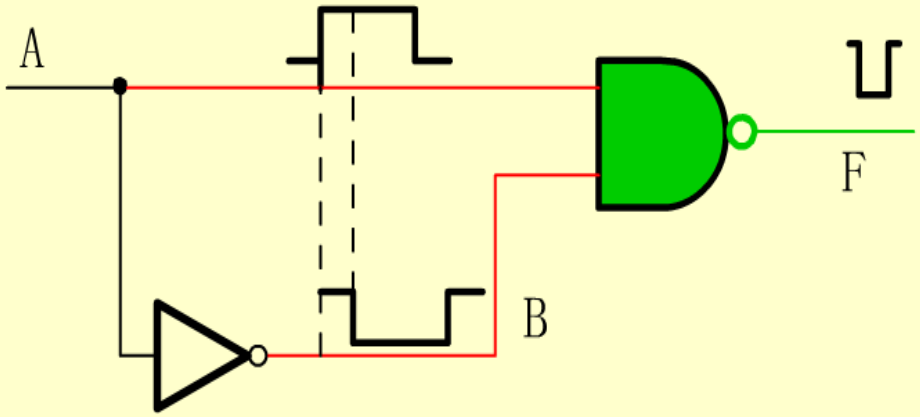
由此看出，原始逻辑电路不是最简的，读者可以根据简化后的表达式自行画出逻辑图进行比较。现根据简化表达式列出逻辑真值表如下：由真值表看出：当ABC组合为001、010、011、100、101时，输出F为1。

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

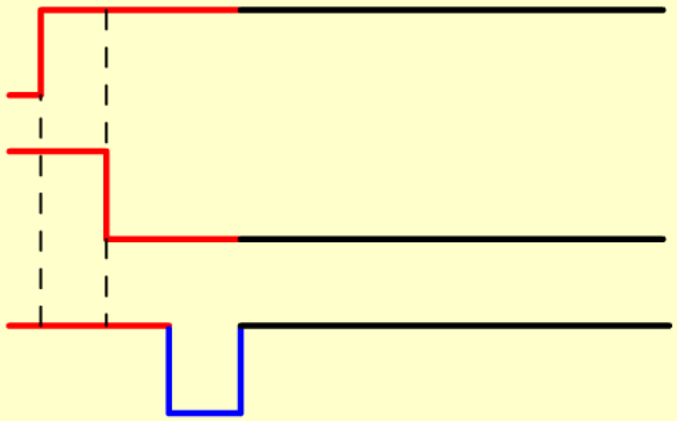
2.1.5 组合逻辑中的竞争冒险

前面讨论组合逻辑电路时，都是假定输入和输出信号已处于稳定状态下来分析的。下面讨论信号在状态转换过程中，有些电路出现的一种现象———竞争冒险：

在组合电路中，当逻辑门有两个互补输入信号同时向相反状态变化时，输出端可能产生过渡干扰脉冲的现象称为竞争冒险。



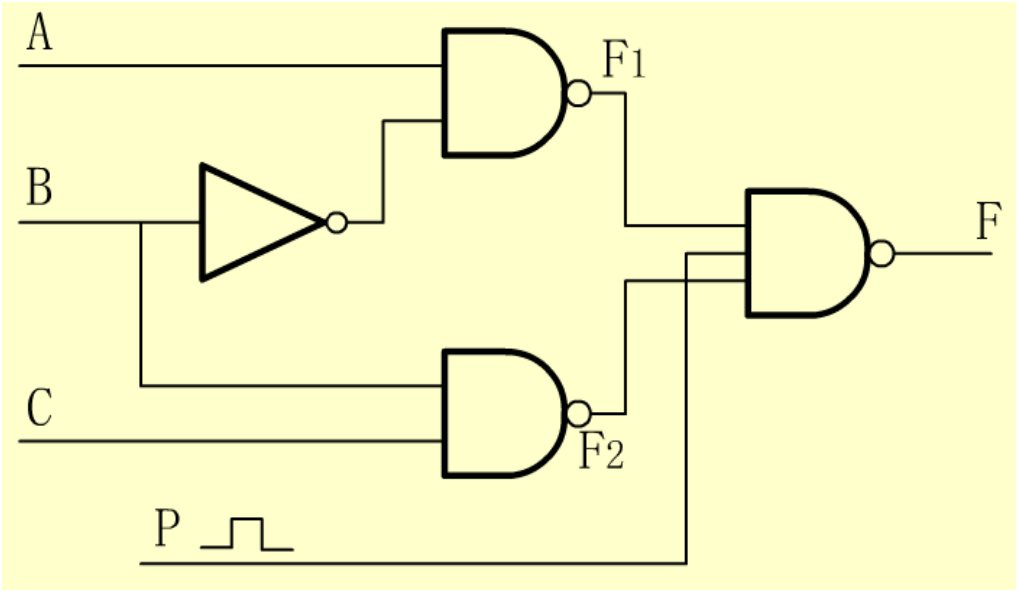
（a）组合逻辑电路

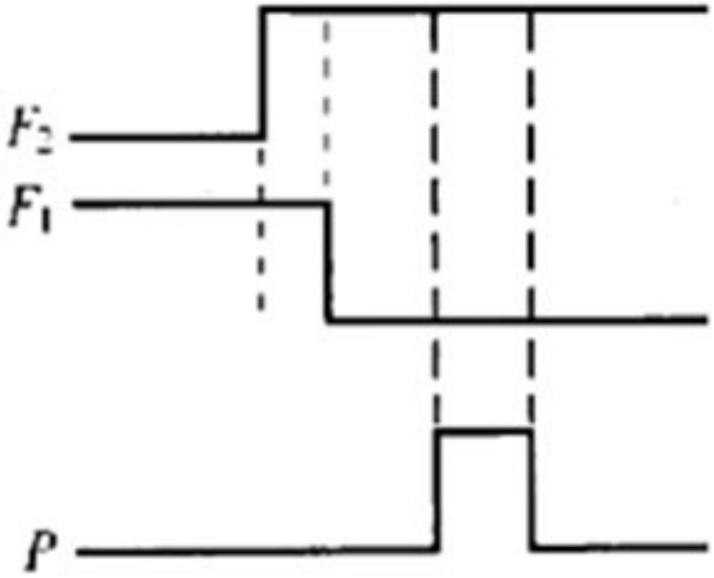


(b)A由0变为1时的波形

通常采用以下两种方法：

1. 加选通脉冲





当A=C=1，$F=\overline{B}+B$，存在竞争冒险。

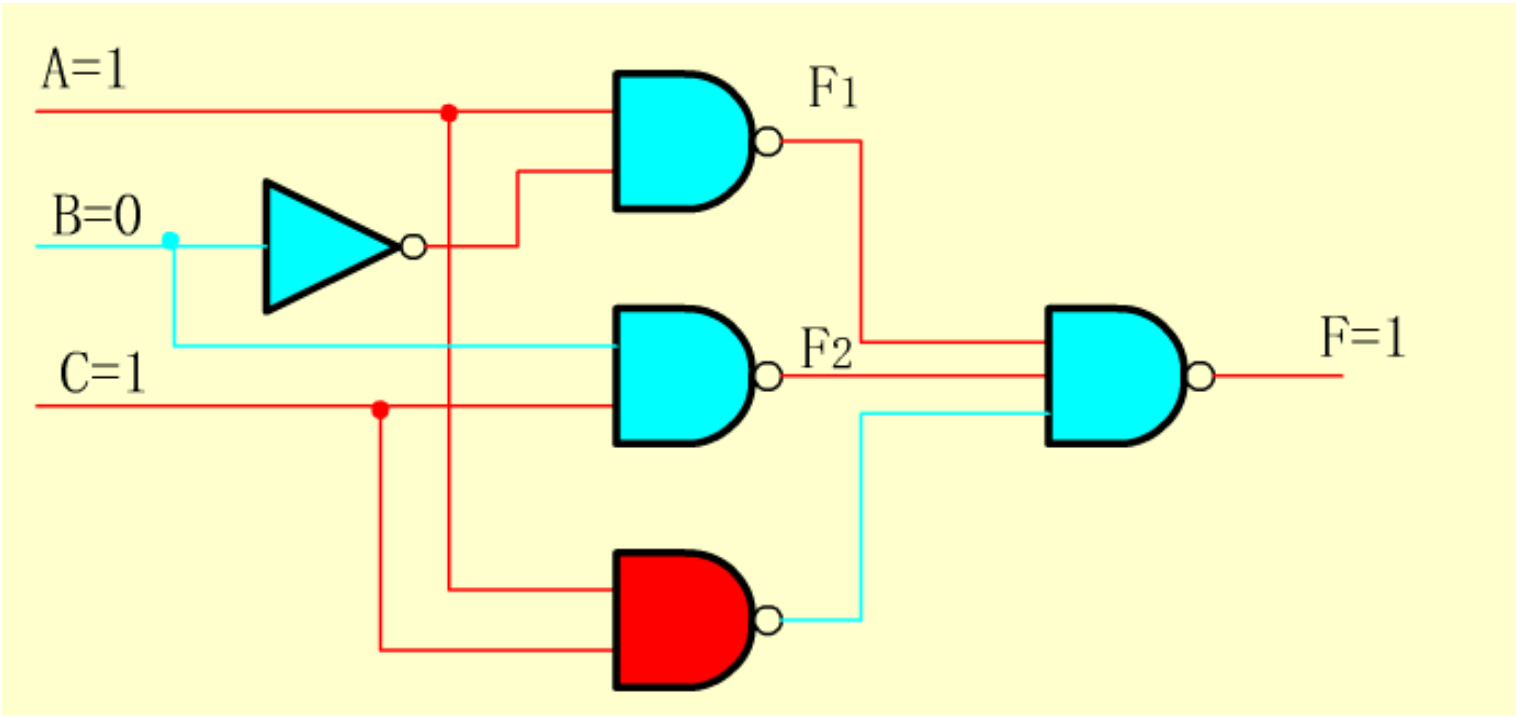
解决：在接收了输入信号并且电路达到了新的稳态之后，才加入选通脉冲。

2. 修改逻辑设计

上例中，我们可以把表达式变换一下，根据常用布尔公式可知：

$$F=A\overline{B}+BC=A\overline{B}+BC+AC$$

上式增加了AC项以后，函数关系不变，但当A=C=1时，输出F恒为1，不再产生干扰脉冲。所以，把电路按上式修改，即可消除竞争冒。



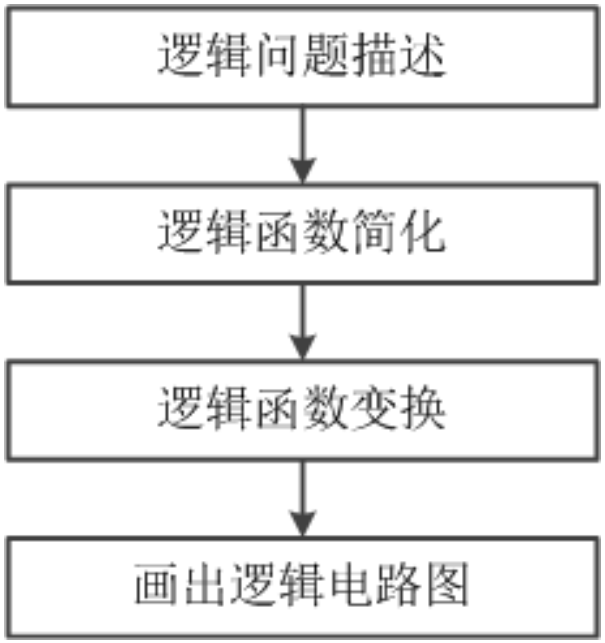
2.2 组合逻辑设计

2.2.1 组合逻辑设计步骤

- *什么是组合逻辑设计?*

组合逻辑设计是组合逻辑分析的逆过程，即最终画出满足功能要求的组合逻辑电路图。

组合逻辑设计的步骤 ?

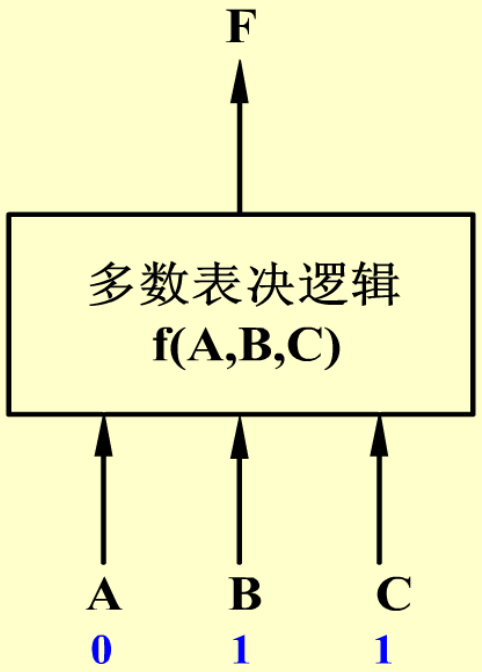


2.2.2 逻辑问题的描述

所谓逻辑问题的描述，就是将文字描述的设计要求抽象为一个逻辑表达式。

通常的方法是：先建立输入输出逻辑变量的真值表，再由真值表写出逻辑表达式。有些情况下，可由设计要求直接建立逻辑表达式。

【例5】设计一个多数表决电路，以判断A、B、C三人中是否多数赞同。



m_i	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$$F = \Sigma(3,5,6,7) = AB\overline{C} + ABC + \overline{A}BC + A\overline{B}C$$

【例6】$X=x_1x_2$, $Y=y_1y_2$是两个正整数，写出X>Y的逻辑表达式

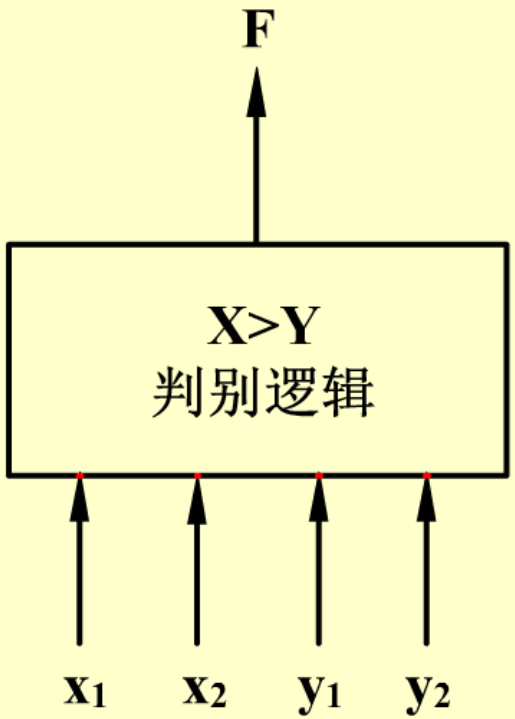


表 2.3 X > Y 的化简真值表

x_1	x_2	y_1	y_2	F
1	×	0	×	1
0	1	0	0	1
1	1	1	0	1

$$F=x_1x_2y_1\overline{y}_2 + x_1\overline{y}_1 + \overline{x}_1x_2\overline{y}_1\overline{y}_2$$

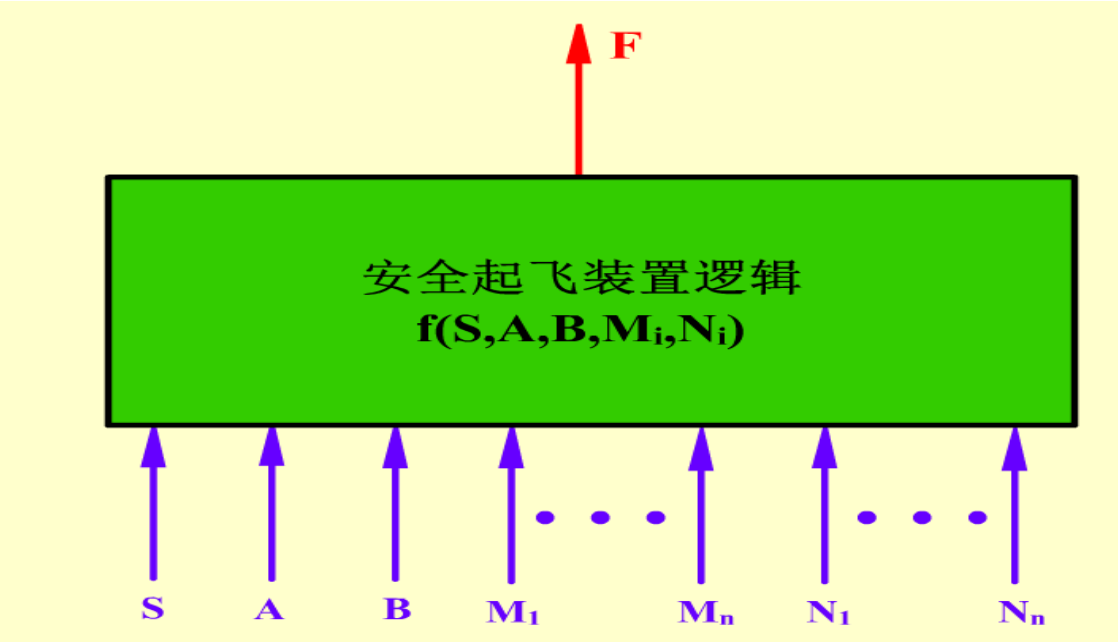
【例7】 某民航客机的安全起飞装置在同时满足下列条件时，发出允许滑跑信号：

①发动机开关接通；

②飞行员入座，且座位保险带已扣上；

③乘客入座，且座位保险带已扣上，或座位上无乘客。

试写出允许发出滑跑信号的逻辑表达式。



$$F=f(S,A,B,M_i,N_i)=S\cdot A\cdot B(M_1N_1+\overline{M}_1)\cdot(M_2N_2+\overline{M}_2)\cdots(M_nN_n+\overline{M}_n)$$

$$=SAB(N_1+\overline{M}_1)\cdot(N_2+\overline{M}_2)\cdots(N_n+\overline{M}_n)$$

2.2.3 利用任意项的逻辑设计

在逻辑表达式中加入任意项（无关项），可使表达式变得更简单。

【例9】用与非门设计一个判别电路，判别8421码的十进制的值≥5

设输入变量为A、B、C、D，输出变量为F， 当 ABCD>=0101时， F=1；当ABCD<0101时，F=0。A、B、C、D的取值不可能出现 1010 ~1111，故约束方程为：

$\sum \Phi(10, 11, 12, 13, 14, 15)=0$

AB \ CD	00	01	11	10
00				
01		5	7	6
11	Φ	Φ	Φ	Φ
10	8	9	Φ	Φ

真值表

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	Φ
1	0	1	1	Φ
1	1	0	0	Φ
1	1	0	1	Φ
1	1	1	0	Φ
1	1	1	1	Φ

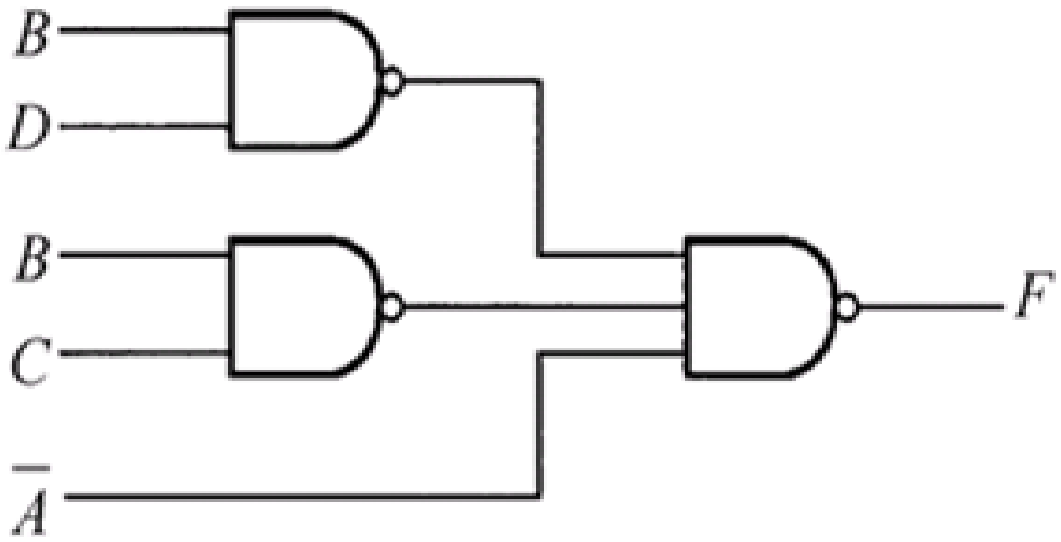
由真值表列出的F的逻辑表达式：

$$F=\sum(5,6,7,8,9)+\sum\Phi(10,11,12,13,14,15)$$

式中$\sum\Phi$部分是任意项，可根据化简的需要引入其中的若干项，使逻辑表达式为最简。利用卡诺图化简，可得化简结果如下：

$$F=BD+BC+A$$

表达式要求用与非门实现，电路图如下：



2.3 组合逻辑电路的等价变换

2.3.1 狄摩根定理的应用

与非门、非或门等价性验证

非与门、或非门等价性验证

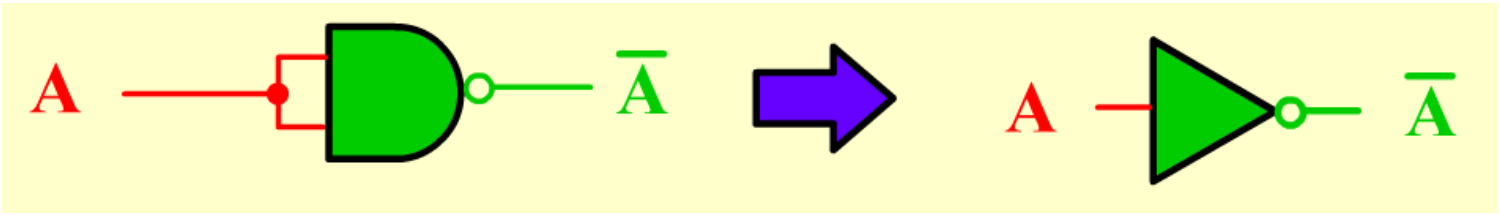
与非门　　非或门

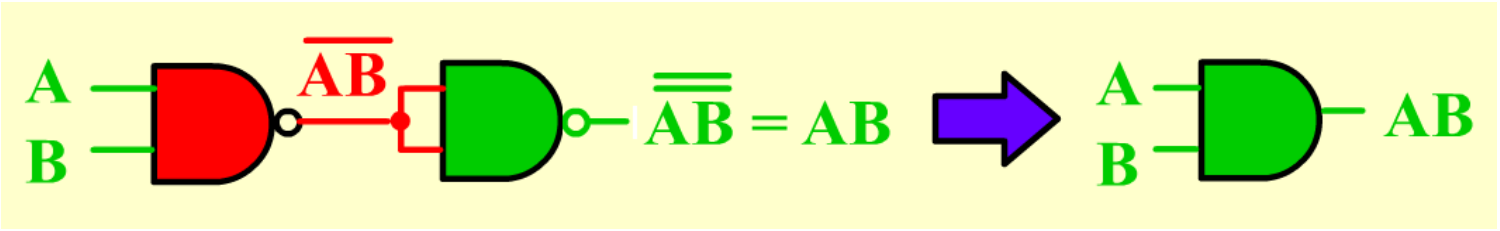
输入		输出	
A	B	$\overline{AB}$	$\overline{A}+\overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

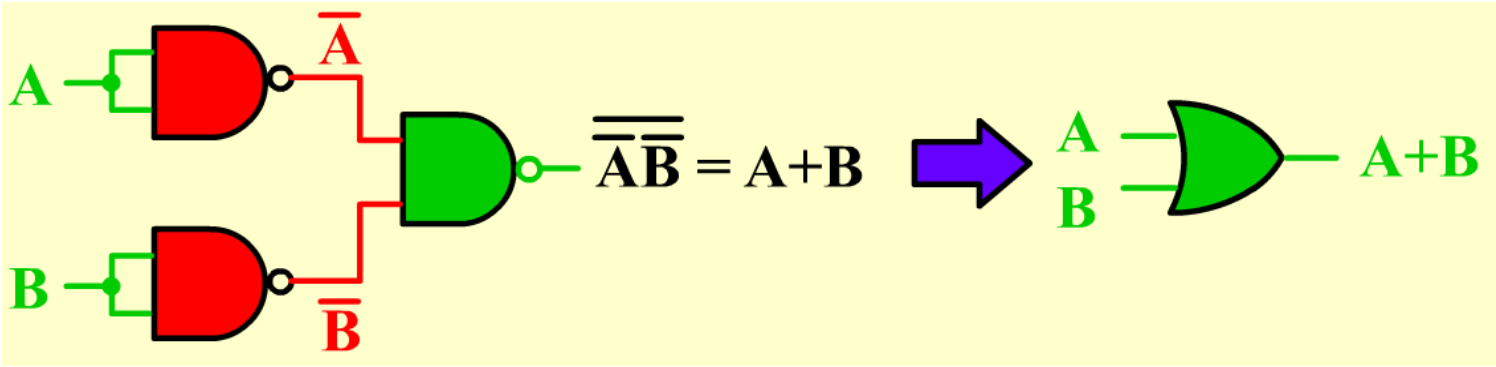
或非门　　非与门

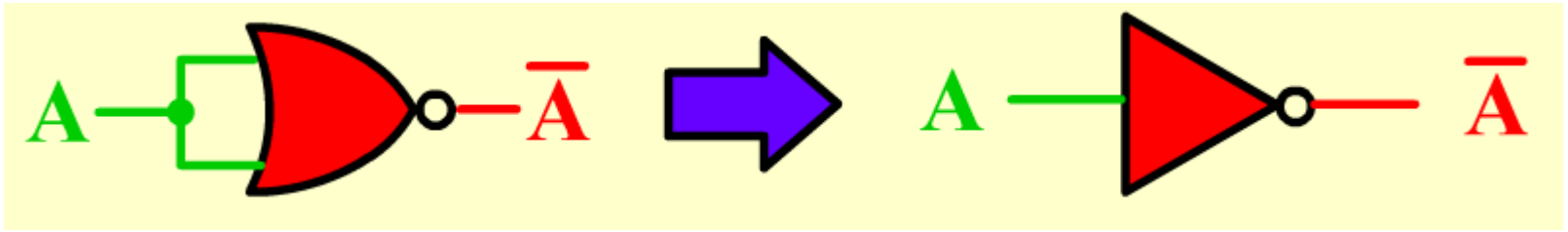
输入		输出	
A	B	$\overline{A+B}$	$\overline{\overline{A}\,\overline{B}}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

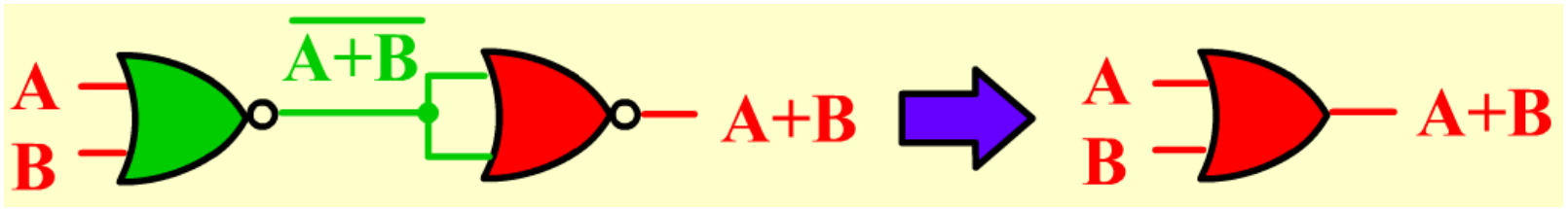
2.3.2 与非门、或非门作为通用元件

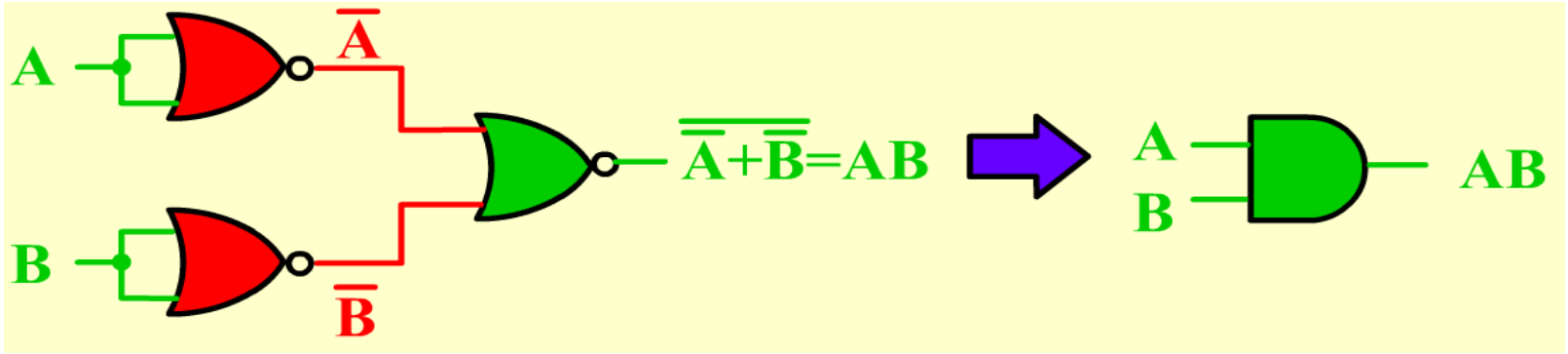








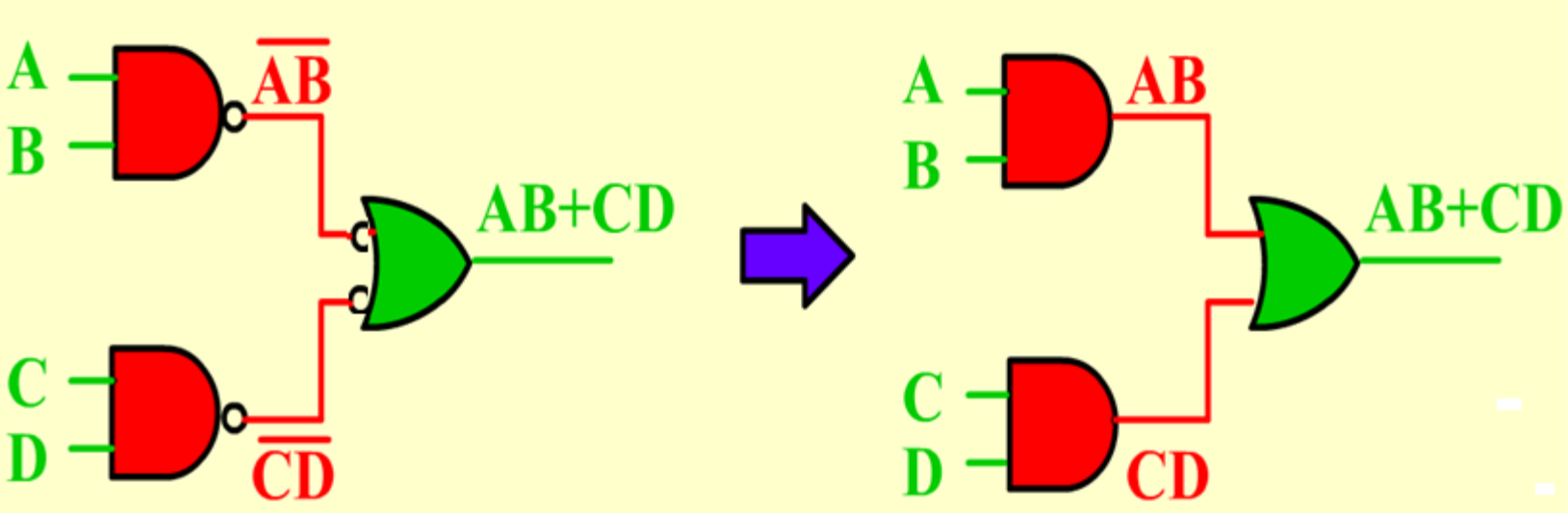




[思考]

思考题： 你能利用与非门 / 非或门实现 $F=\overline{A}+B$ ？

2.3.3 利用与非门 / 非或门进行等价变换



2.3.4 逻辑函数的“与或非”门实现

将最简“与或”表达式变换为“与或非”表达式的方法有两种：

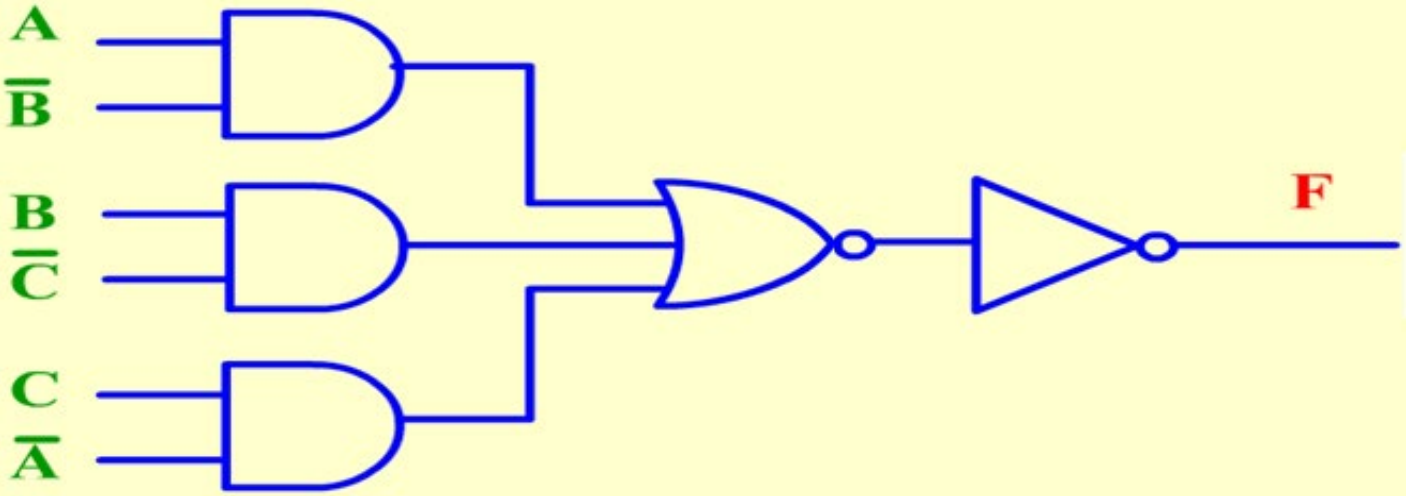
一是对F两次求反； 二是对$\overline{F}$一次求反。

如何让信号传输经过门的级数最少？

例12 用与或非门实现函数 $F=A\overline{B}+B\overline{C}+C\overline{A}$

（1）F两次求反，可得

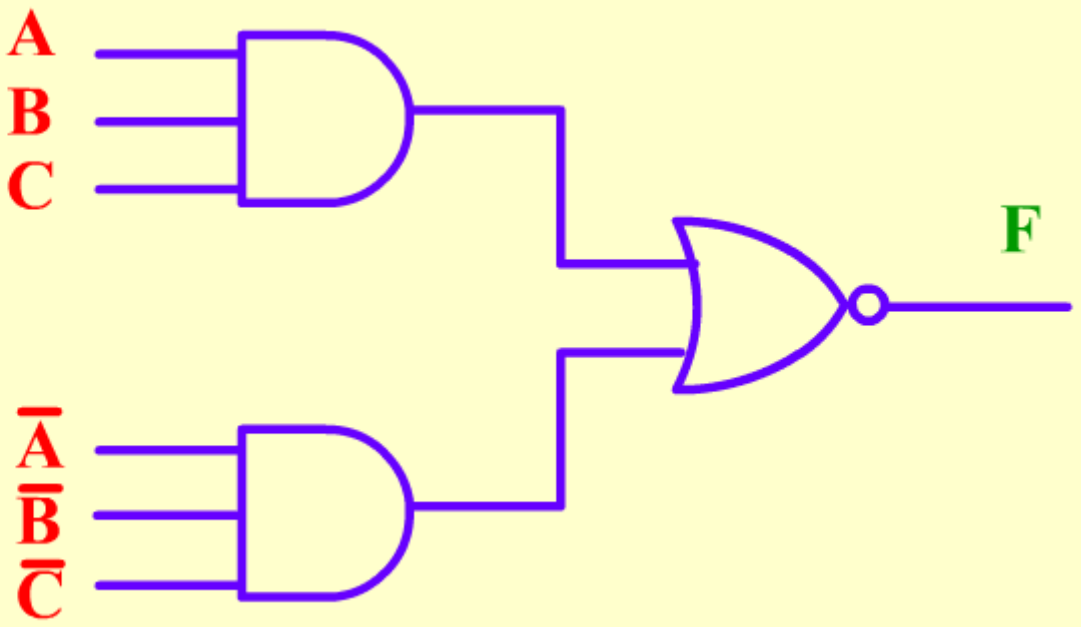
$$F=\overline{A\overline{B}+B\overline{C}+C\overline{A}}$$



(2) $\overline{F}$一次求反，可得

$$\overline{F}=\overline{A\,\overline{B}+B\,\overline{C}+C\,\overline{A}}=\overline{A}\,\overline{B}\,\overline{C}+A\,B\,C$$

$$F=\overline{\overline{F}}=\overline{\overline{A}\,\overline{B}\,\overline{C}+A\,B\,C}$$



比较可知，第二种方法所得之结果速度快，信号传输只经过两级门。

2.4 数据选择器与分配器

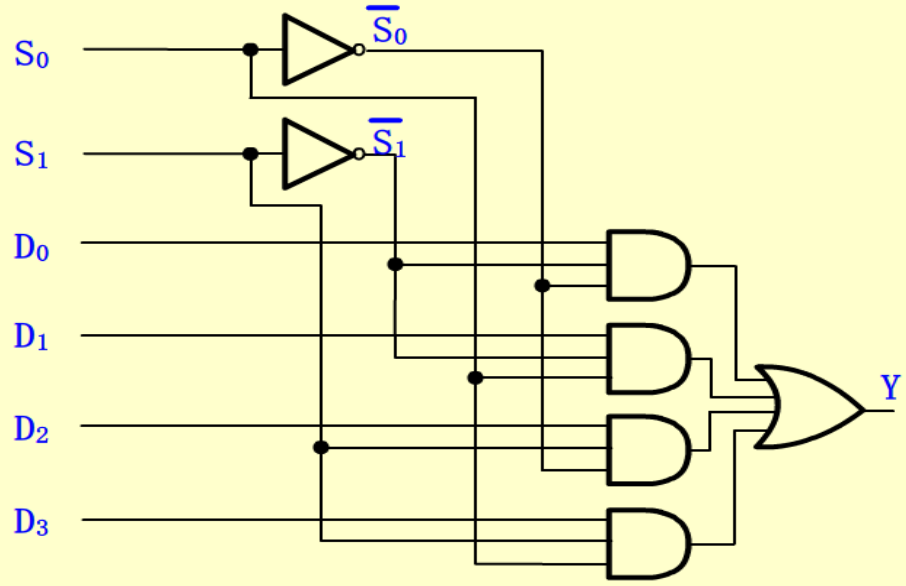
2.4.1 数据选择器

● 什么是数据选择器MUX

数据选择器又称：多路转换器　多路开关

定义：多路输入、单路输出的组合逻辑构件。

用途：　TDM时分复用



4-1线数据选择器功能表

选择输入		数据输入				输出
S_1	S_0	D_0	D_1	D_2	D_3	Y
X	X	X	X	X	X	0
0	0	D_0	X	X	X	D_0
0	1	X	D_1	X	X	D_1
1	0	X	X	D_2	X	D_2
1	1	X	X	X	D_3	D_3

● 四选一MUX的逻辑表达式：

$$F = D_0\overline{S_1}\,\overline{S_2} + D_1\overline{S_1}S_2 + D_2S_1\overline{S_2} + D_3S_1S_2$$

$$= \sum_{i=0}^{3} D_i M_i$$

其中Di（i=0, 1，2，3）是四路数据输入。Mi是两个地址输入（S_1, S_0）的4个最小项。

推广之，八选一MUX的逻辑表达式：

$$F = \sum_{i=0}^{7} D_i M_i$$

其中Di（i=0, 1...7）是八路数据输入。Mi是3个地址输入（S_2, S_1, S_0）的8个最小项

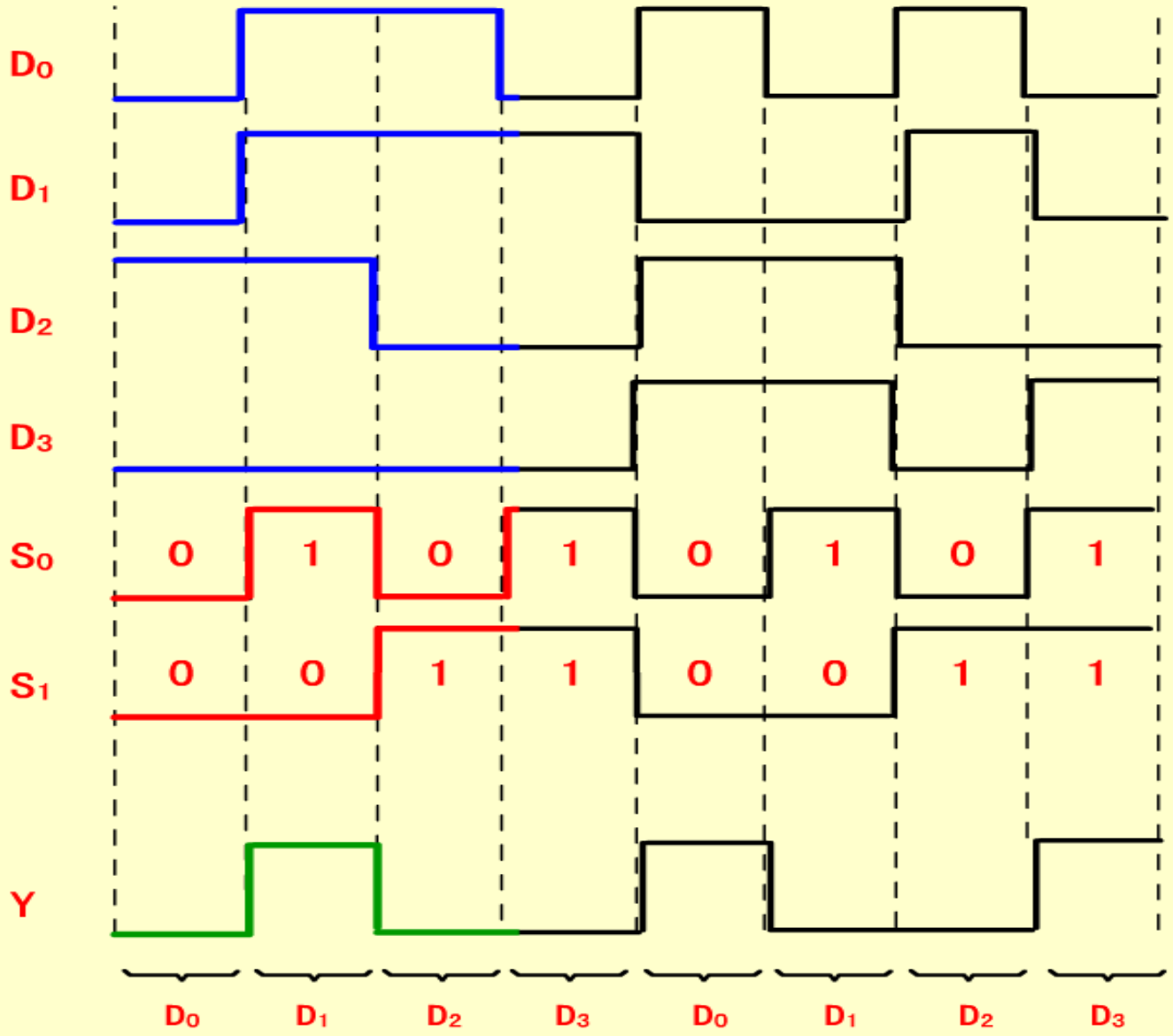


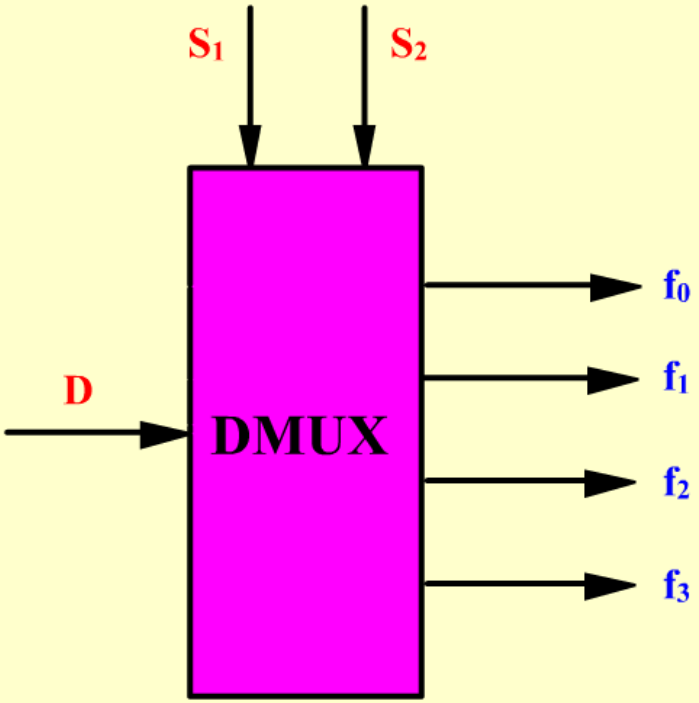
图2.20 四选一MUX的输入输出波形图

2.4.2 数据分配器

数据分配器DMUX， 与MUX相反：

定义:单路输入、多路输出的组合逻辑构件。

用途： 数据交换



输 入		输 出			
S_1	S_2	f_0	f_1	f_2	f_3
0	0	D	1	1	1
0	1	1	D	1	1
1	0	1	1	D	1
1	1	1	1	1	D

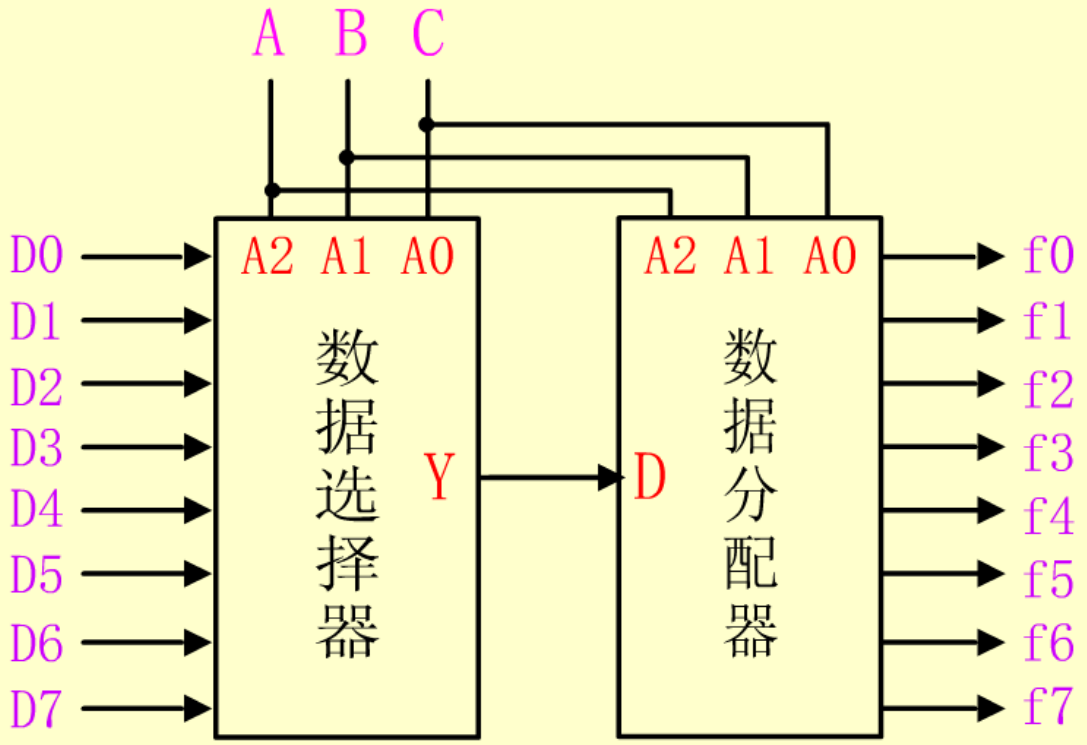
1线—4线数据分配器

【例2】 利用DMUX和MUX设计一个实现8路数据传输的逻辑电路。

[解]

使用一个8选1的MUX ，再用一个1：8线的DMUX，并将它们的地址输入端$S_2S_1S_0$连在一起，使$S_2S_1S_0$上的控制信号依次由000--001--010--011--100--101--110--111定时变化，则可以分时实现8路数据传输。

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



2.5 译码器和编码器

2.5.1 译码器

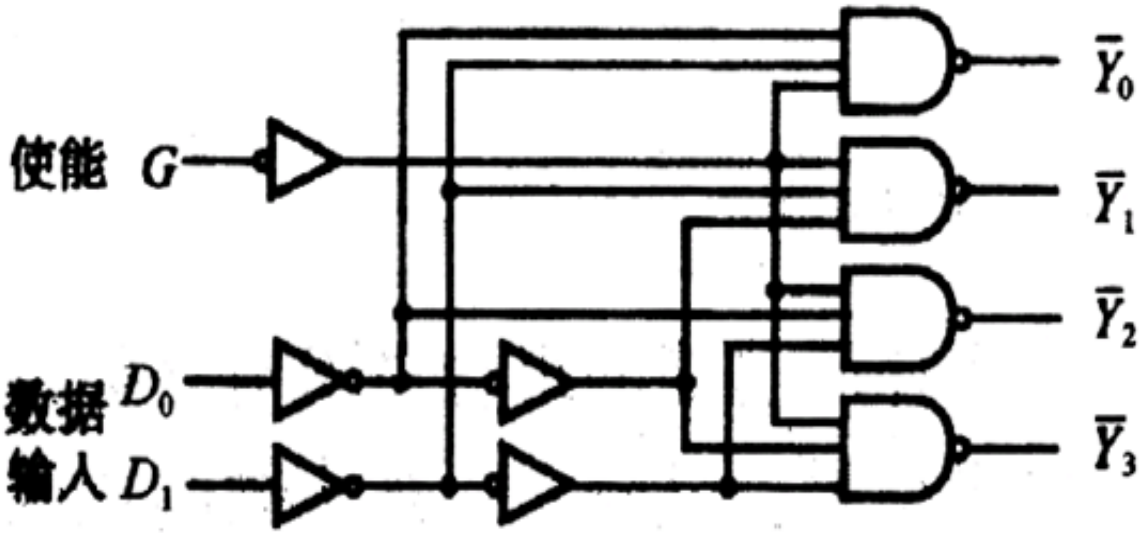
实现译码功能的组合逻辑电路称为译码器.

输入是一组二进制代码 → 输出是一组高低电平信号,对于每输入一组不同的代码：只有一个输出呈现有效状态

应用：如地址译码器

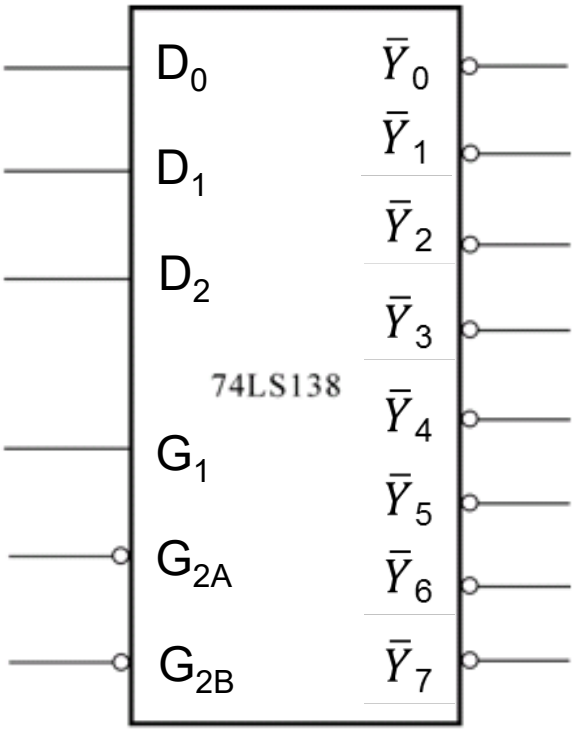
1. 3线:8线译码器和2线：4线译码器

2线：4线译码器：



1. 3 线:8 线译码器和2线：4线译码器

3线：8线译码器可对 8 线中的某一线进行译码.

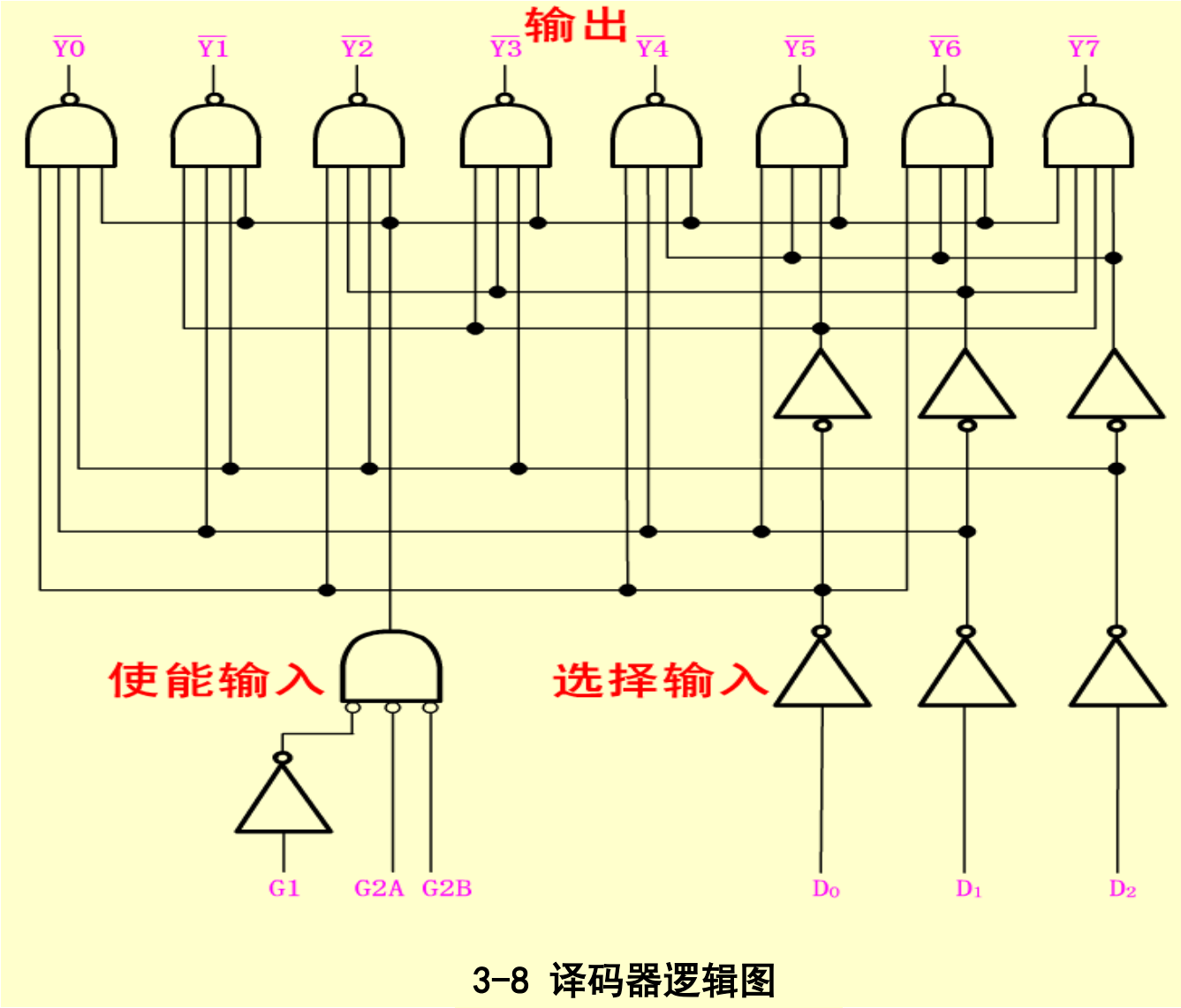


74LS138功能表

使能输入		选择输入			输出							
G1	G2	D_2	D_1	D_0	$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$	$\overline{Y4}$	$\overline{Y5}$	$\overline{Y6}$	$\overline{Y7}$
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

1. 3 线:8 线译码器和2线：4线译码器

3线：8线译码器可对 8 线中的某一线进行译码.

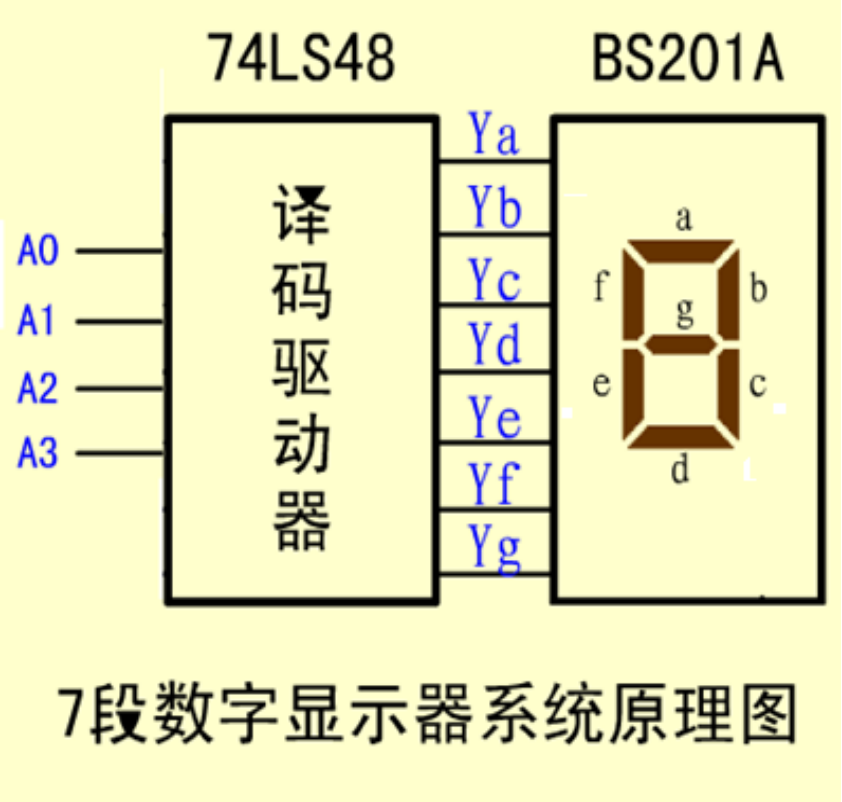


3-8 译码器逻辑图

2. 七段数字译码显示系统

在数字系统中，人们常常采用简易数字显示电路将测量或运算结果用数码直接显示出来，以便于监视系统工作情况。

七段荧光数码管的显示系统由译码/驱动器 74LS48和共阴极荧光数码管BS201A组成



7段数字显示器系统原理图

74LS48逻辑功能表

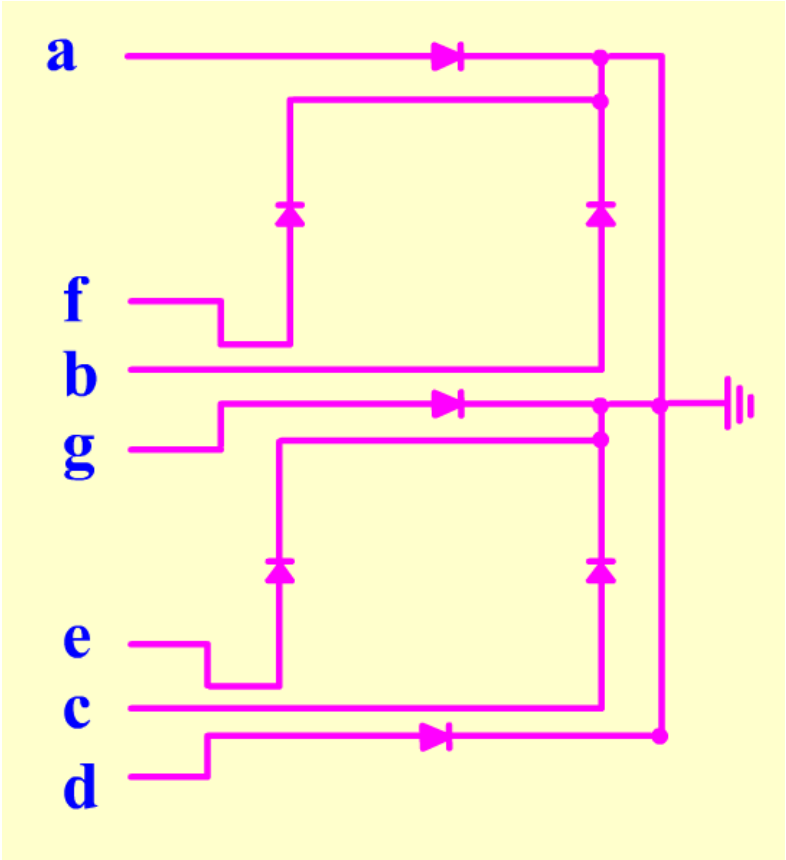
输入				输出							显示字符
A3	A2	A1	A0	Ya	Yb	Yc	Yd	Ye	Yf	Yg	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9

七段荧光数码管是分段式半导体显示器件，7个发光二极管组成七个发光段。

发光二极管显示电路有两种连接方式：

一种是七个发光二极管共用一个阳极，称为共阳极电路。

另一种是7个发光二极管共用一个阴极，称为共阴极电路。



2.5.2 编码器

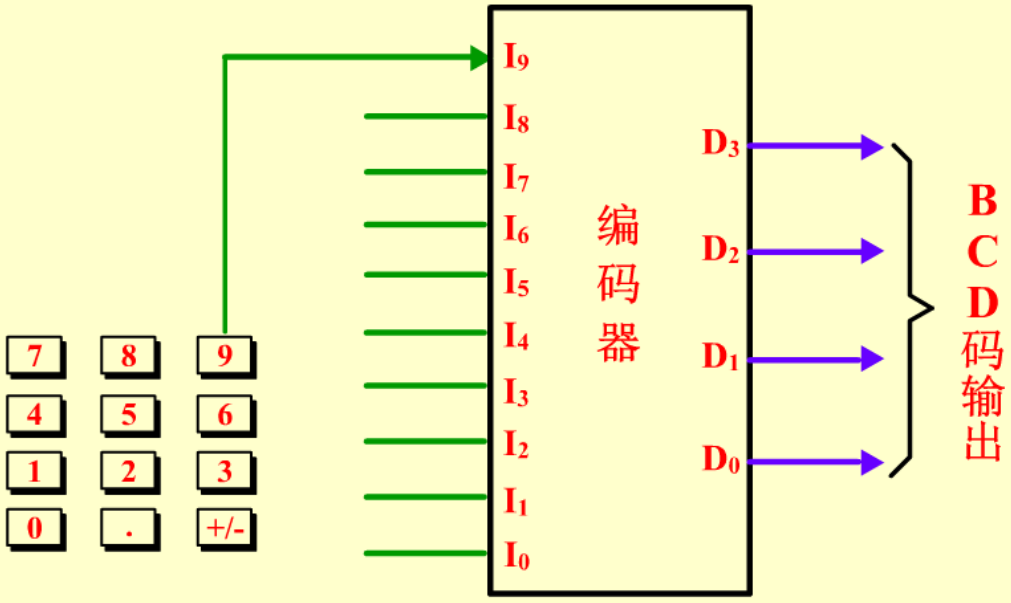
对所处理的信息或数据赋予“一组”二进制代码，称为编码。

与译码器相反

1. 普通编码器

在任意一时刻所有输入线中只允许一个输入线上有信号。

为某个输入引脚产生一个码，如为每个按键生成一个BCD码。



(a)结构框图

表2.9 普通编码器真值表

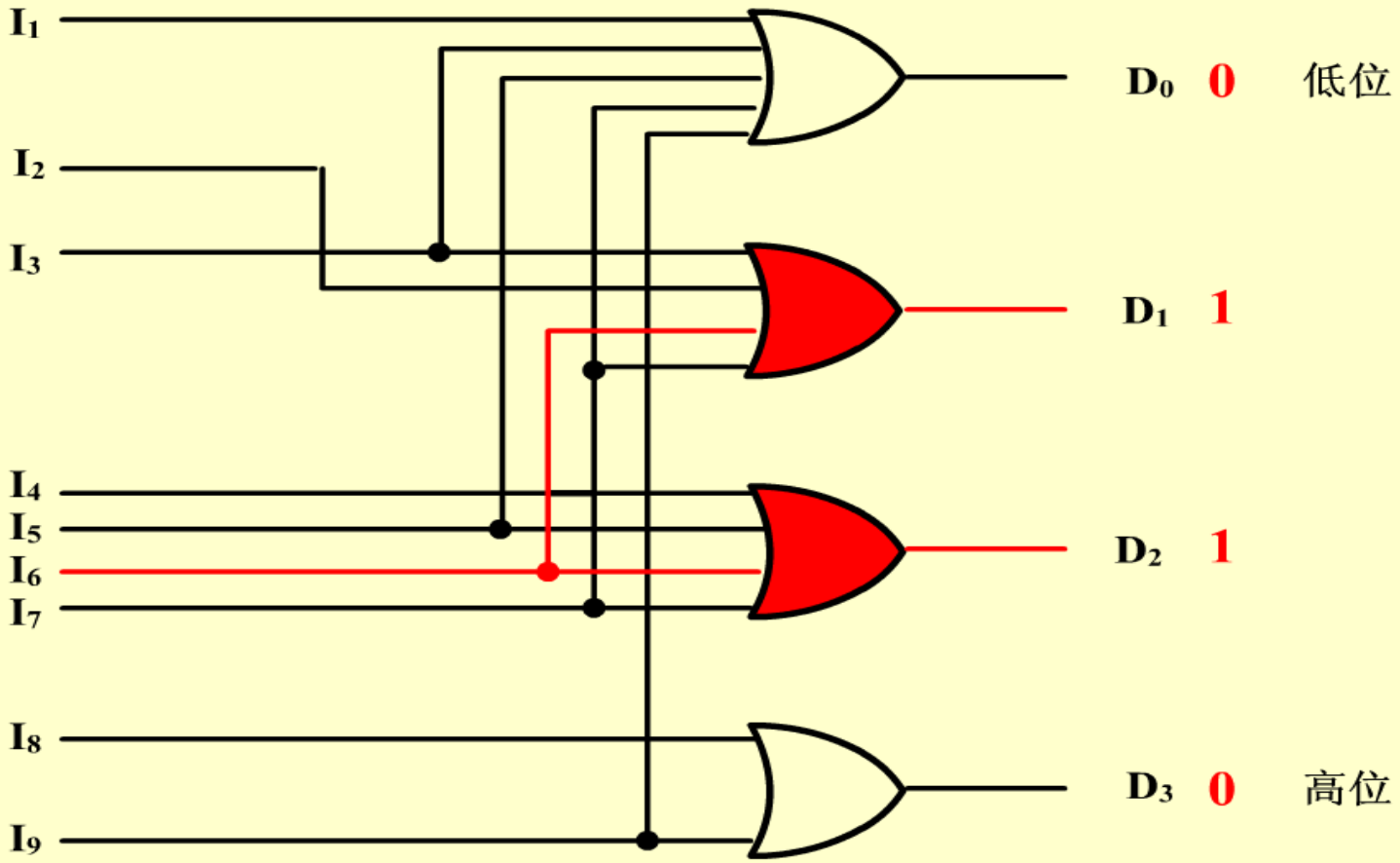
输入	输出 BCD 码			
十进制数字信号	D_3	D_2	D_1	D_0
I_0	0	0	0	0
I_1	0	0	0	1
I_2	0	0	1	0
I_3	0	0	1	1
I_4	0	1	0	0
I_5	0	1	0	1
I_6	0	1	1	0
I_7	0	1	1	1
I_8	1	0	0	0
I_9	1	0	0	1

$D_3=I_8+I_9$

$D_2=I_4+I_5+I_6+I_7$

$D_1=I_2+I_3+I_6+I_7$

$D_0=I_1+I_3+I_5+I_7+I_9$



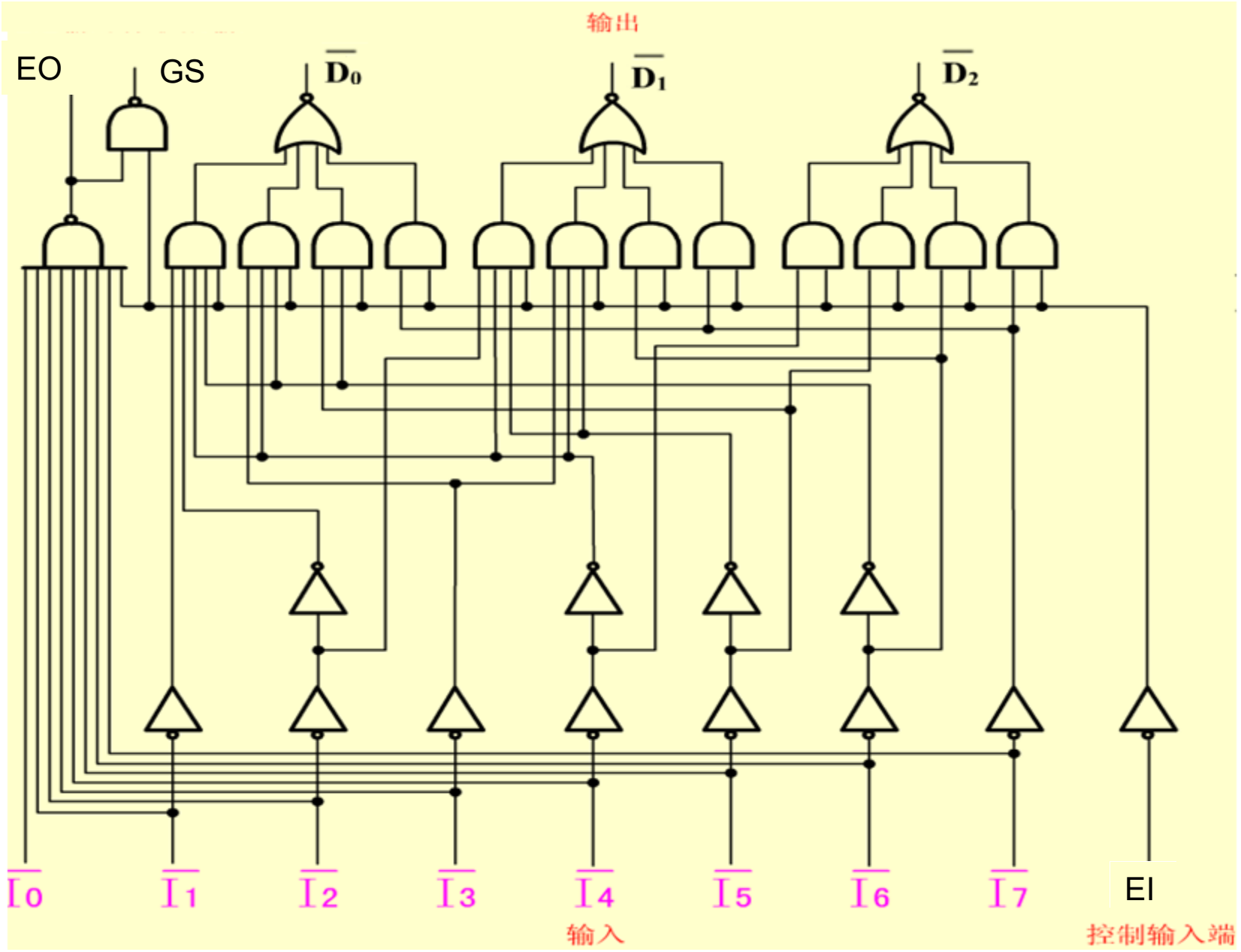
(b)逻辑电路图

2. 优先编码器

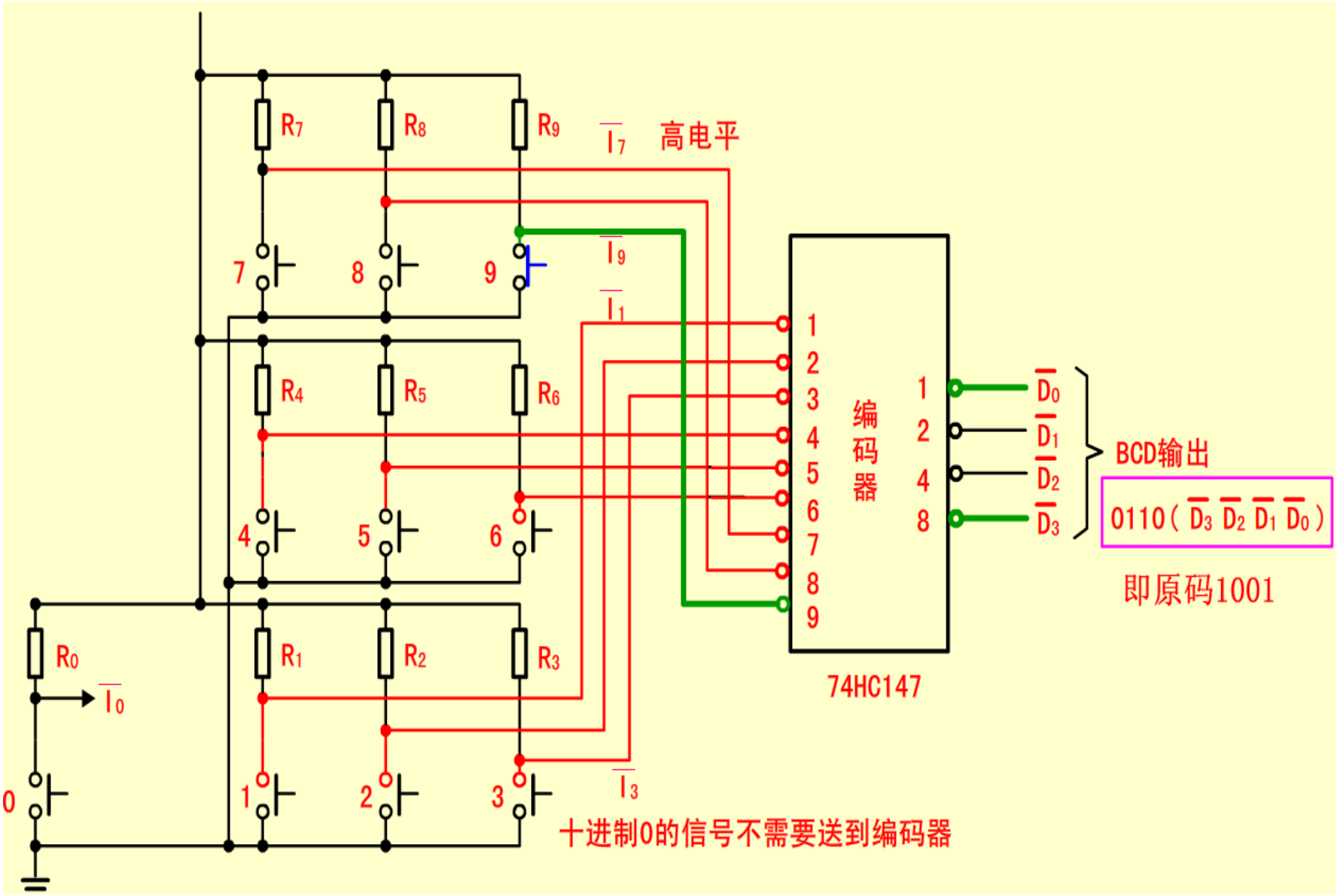
不同于普通编码器： 它允许多个输入线上同时有信号。

如何解决混乱？

EI	$\overline{I_0}$	$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	$\overline{I_4}$	$\overline{I_5}$	$\overline{I_6}$	$\overline{I_7}$	$\overline{D_2}$	$\overline{D_1}$	$\overline{D_0}$	GS	EO
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	×	×	×	×	×	×	0	1	0	0	1	0	1
0	×	×	×	×	×	0	1	1	0	1	0	0	1
0	×	×	×	×	0	1	1	1	0	1	1	0	1
0	×	×	×	0	1	1	1	1	1	0	0	0	1
0	×	×	0	1	1	1	1	1	1	0	1	0	1
0	×	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1


输出
EO
GS
$\overline{D_0}$
$\overline{D_1}$
$\overline{D_2}$
$\overline{I_0}$
$\overline{I_1}$
$\overline{I_2}$
$\overline{I_3}$
$\overline{I_4}$
$\overline{I_5}$
$\overline{I_6}$
$\overline{I_7}$
EI
输入
控制输入端

【例15】设计十进制数字键盘的 优先编码逻辑。 （课堂练习）

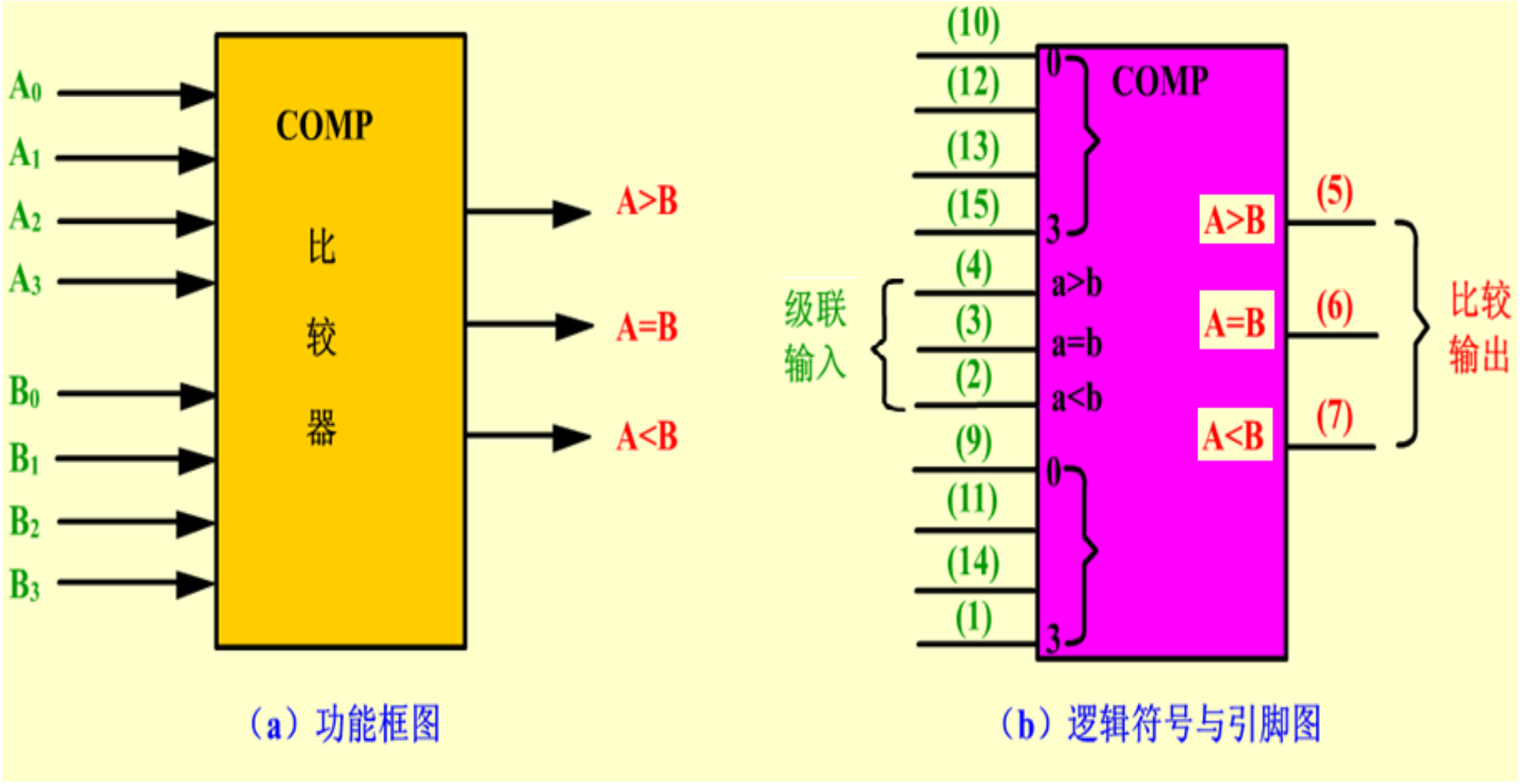


2.6 数据比较器和加法器

2.6.1 数据比较器

1. 定义

用来完成两组二进制数码大小比较的逻辑电路，称为数据比较器。



表格 2.11 4位比较器74HC85功能表

比较输入				级联输入			输出		
a_3b_3	a_2b_2	a_1b_1	a_0b_0	a>b	a<b	a=b	A>B	A<B	A=B
$a_3>b_3$	X	X	X	X	X	X	1	0	0
$a_3<b_3$	X	X	X	X	X	X	0	1	0
$a_3=b_3$	$a_2>b2$	X	X	X	X	X	1	0	0
$a_3=b_3$	$a_2<b2$	X	X	X	X	X	0	1	0
$a_3=b_3$	$a_2=b2$	$a_1>b_1$	X	X	X	X	1	0	0
$a_3=b_3$	$a_2=b2$	$a_1<b_1$	X	X	X	X	0	1	0
$a_3=b_3$	$a_2=b2$	$a_1=b_1$	$a_0>b_0$	X	X	X	1	0	0
$a_3=b_3$	$a_2=b2$	$a_1=b_1$	$a_0<b_0$	X	X	X	0	1	0
$a_3=b_3$	$a_2=b2$	$a_1=b_1$	$a_0=b_0$	1	0	0	1	0	0
$a_3=b_3$	$a_2=b2$	$a_1=b_1$	$a_0=b_0$	0	1	0	0	1	0
$a_3=b_3$	$a_2=b2$	$a_1=b_1$	$a_0=b_0$	0	0	1	0	0	1

【例16】使用74HC85比较器组成 8 位比较器。

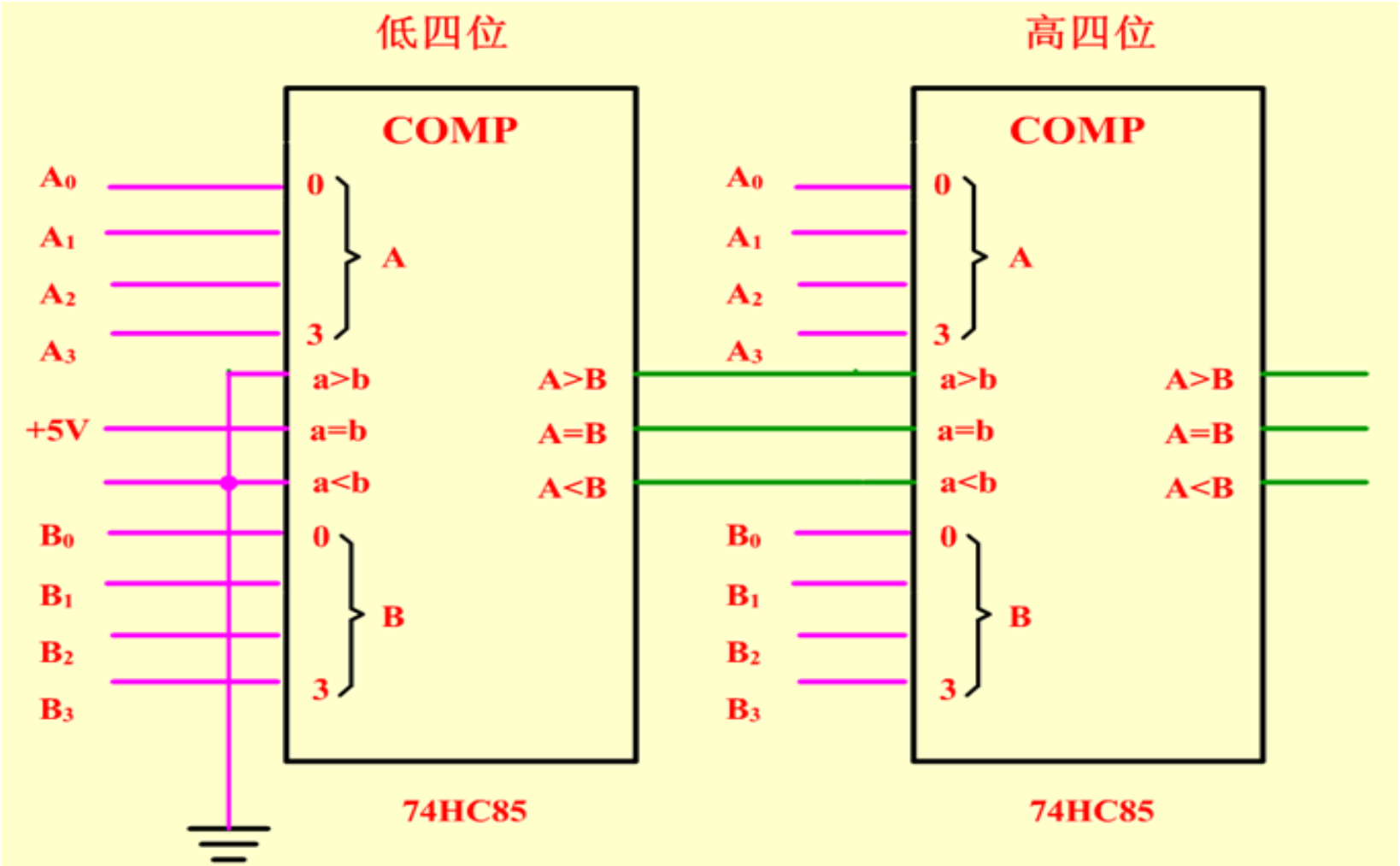


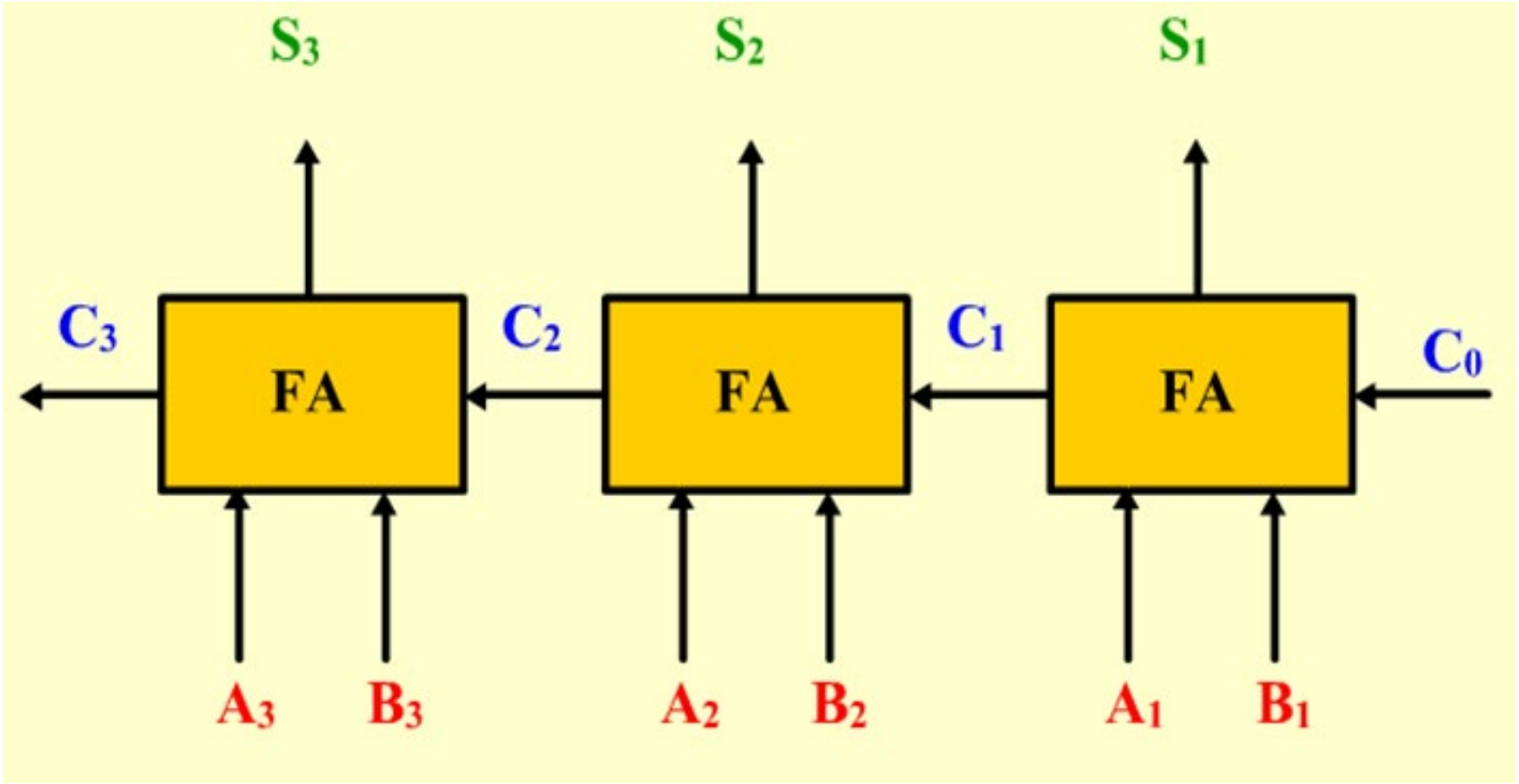
图2.30 使用2片74HC85组成8位比较器

2.6.2 加法器

1. 串行加法器

由多个全加器（FA）串行连接而成。

串行进位方式，进位信号逐位向上传递，延迟大



2.6.2 加法器

1. 串行加法器

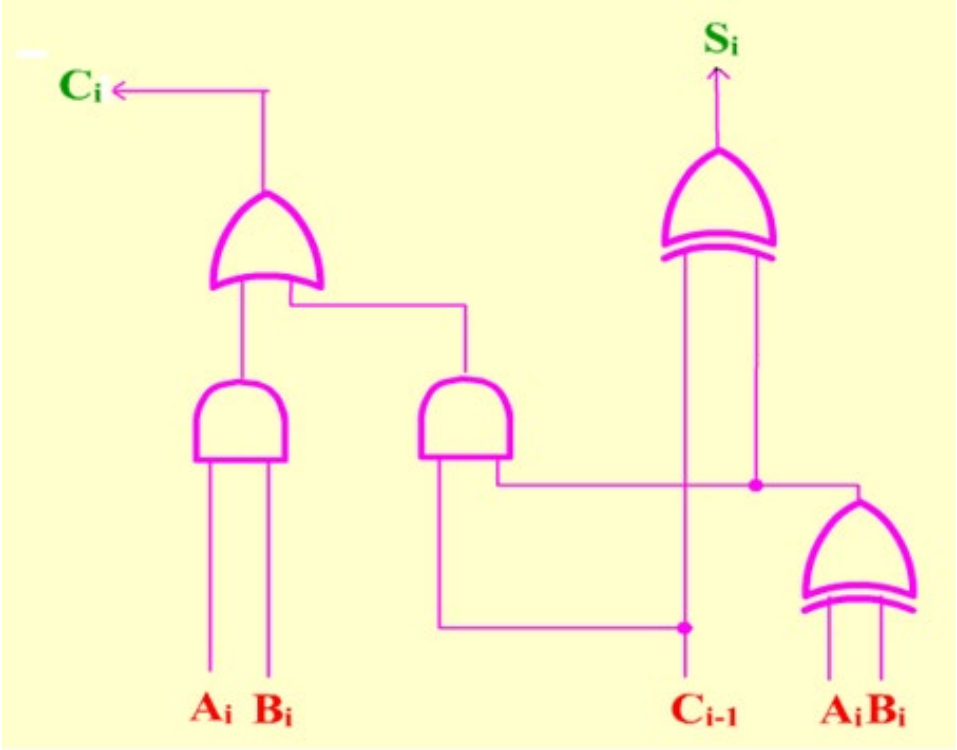
输入			输出	
A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

全加器真值表

Si和Ci的逻辑表达式：

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

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}$$



2. 并行加法器

4 位超前进位并行加法器74LS283

设 $A=A_4A_3A_2A_1$　　$B=B_4B_3B_2B_1$

$S_1=A_1 \oplus B_1 \oplus C_0$　　$C_1=A_1B_1+(A_1 \oplus B_1)C_0$

$S_2=A_2 \oplus B_2 \oplus C_1$　　$C_2=A_2B_2+(A_2 \oplus B_2)C_1$

$S_3=A_3 \oplus B_3 \oplus C_2$　　$C_3=A_3B_3+(A_3 \oplus B_3)C_2$

$S_4=A_4 \oplus B_4 \oplus C_3$　　$C_4=A_4B_4+(A_4 \oplus B_4)C_3$

令　$G_i=A_iB_i$　$P_i=A_i \oplus B_i$　则　$C_i=G_i+P_iC_{i-1}$

采用递推方法，得：

$C_1=G_1+P_1C_0$

$C_2=G_2+P_2C_1=G_2+P_2G_1+P_2P_1C_0$

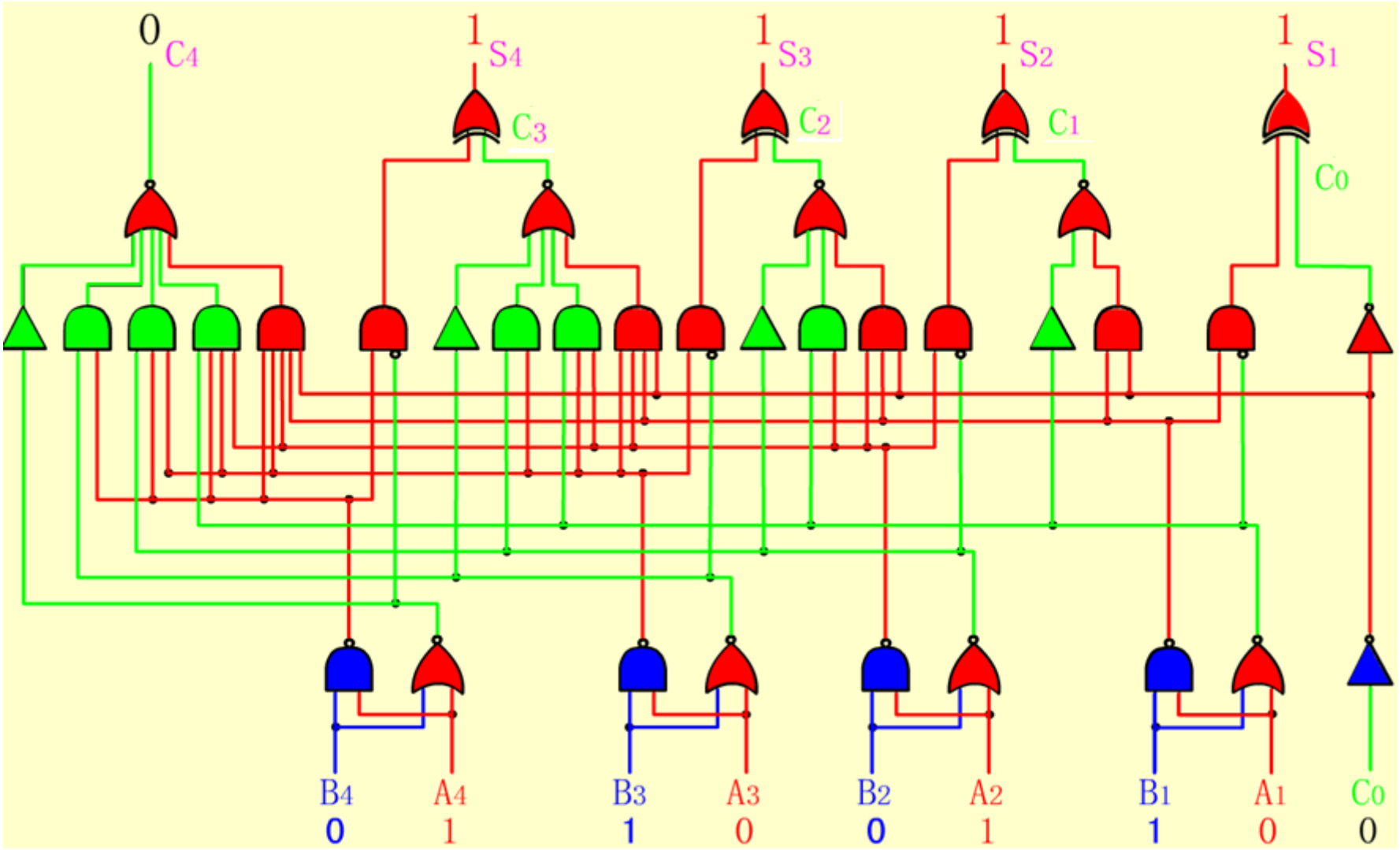
$C_3=G_3+P_3C_2=G_3+P_3G_2+P_3P_2P_1G_1+P_3P_2P_1C_0$

$C_4=G_4+P_4C_3=G_4+P_4G_3+P_4P_3G_2+P_4P_3P_2G_1+P_4P_3P_2P_1C_0$

C_4的表达式说明，最低位的进位符号C_0可以超前传送到最高位C_4上，从而使加法器的运算速度大大加快了。

2. 并行加法器

4 位超前进位并行加法器74LS283



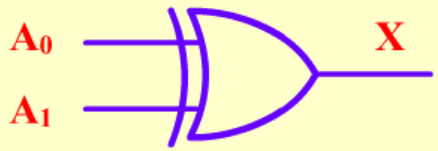
2.7 奇偶校验器

2.7.1 奇偶校验的基本原理

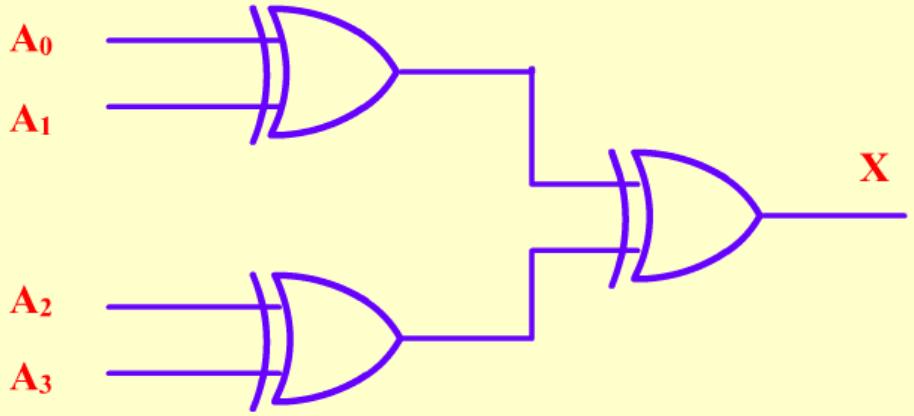
什么是奇偶校验器?

利用奇(偶)校验方法进行检错的组合逻辑电路称为奇偶校验器。

偶数个1，它的和总是0；奇数个1，它的和总是1。

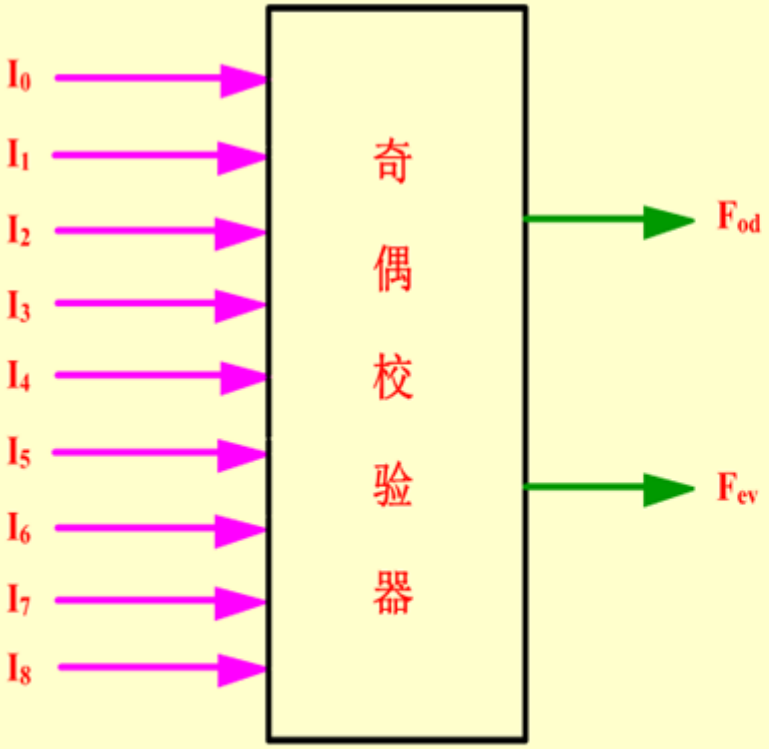


(a)2比特求和



(b)4比特求和

2.7.2 具有奇偶校验的数据传输

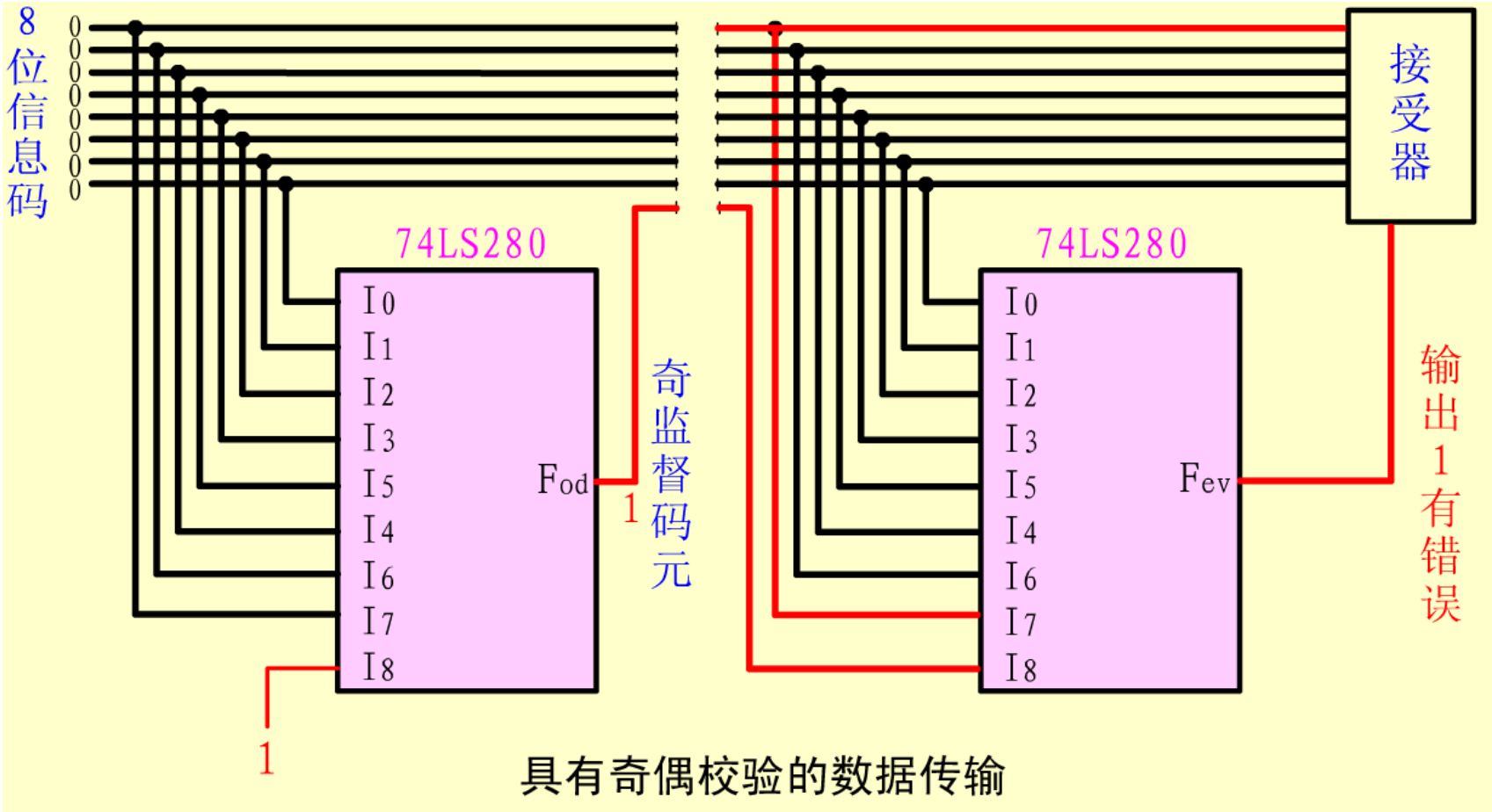


输　入	输 出	
I0-I8 中1的个数	Fod	Fev
偶数(0, 2, 4, 6, 8)	0	1
奇数(1, 3, 5, 7, 9)	1	0

图2.34 74LS280逻辑框图及功能表

2.7.2 具有奇偶校验的数据传输

图2.35中所示为一个具有奇校验器的数据传输系统，采用2片74LS280，74LS280逻辑框图如下：



具有奇偶校验的数据传输

2.7.2 具有奇偶校验的数据传输

在发送端：使用一个74LS280产生监督位F_{od}信号，I_8 =1，F_{od}则指示8位数据1的个数为偶数：

$$F_{od} = (I_0 \oplus I_1 \oplus I_2 \oplus I_3 \oplus I_4 \oplus I_5 \oplus I_6 \oplus I_7) \oplus I_8$$

因此F_{od}取值使 9 位码组（8位数据加I_8）中奇数个1

在接收端： 9 位码组作为第二个74LS280输入，取F_{ev}指示正确性（ 9 位码组保持奇数个1）

$$F_{ev} = \overline{F_{od}} = \overline{(I_0 \oplus I_1 \oplus I_2 \oplus I_3 \oplus I_4 \oplus I_5 \oplus I_6 \oplus I_7) \oplus I_8}$$

若F_{ev} =0，则传输正确