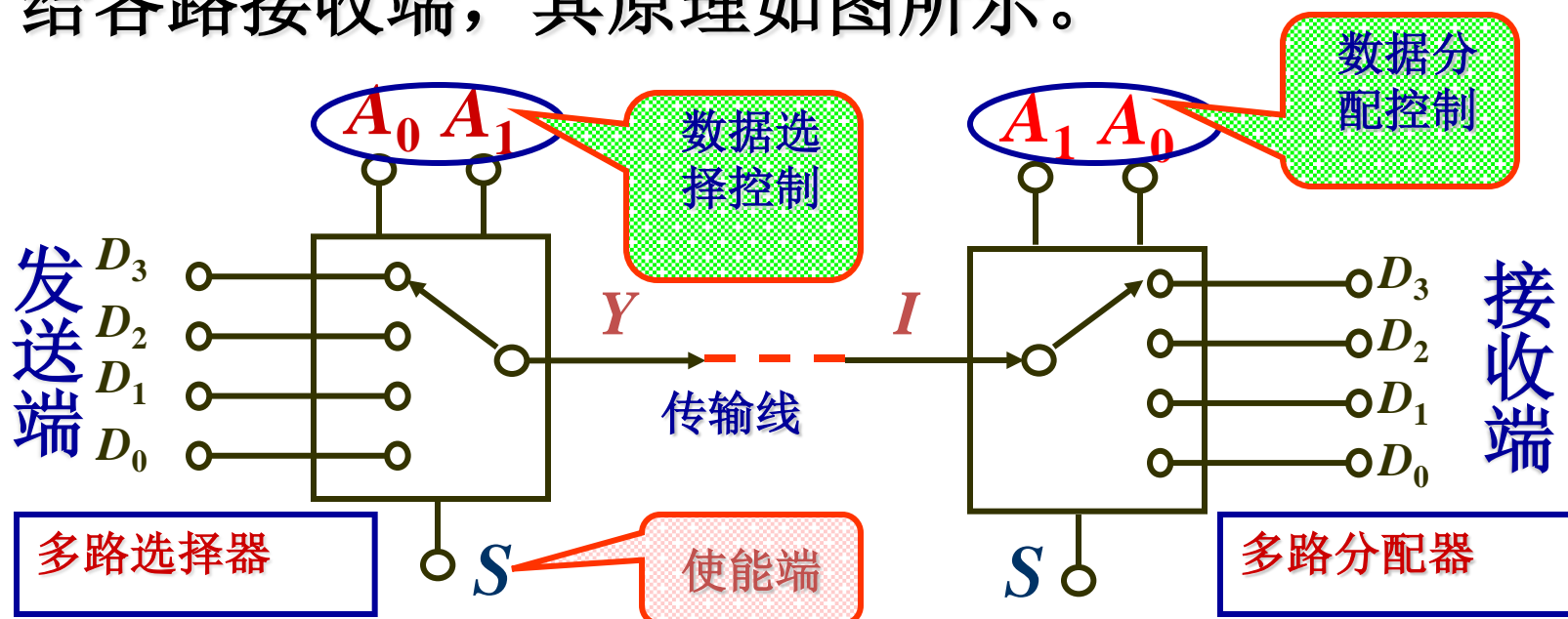


## 2.4 数据分配器和数据选择器

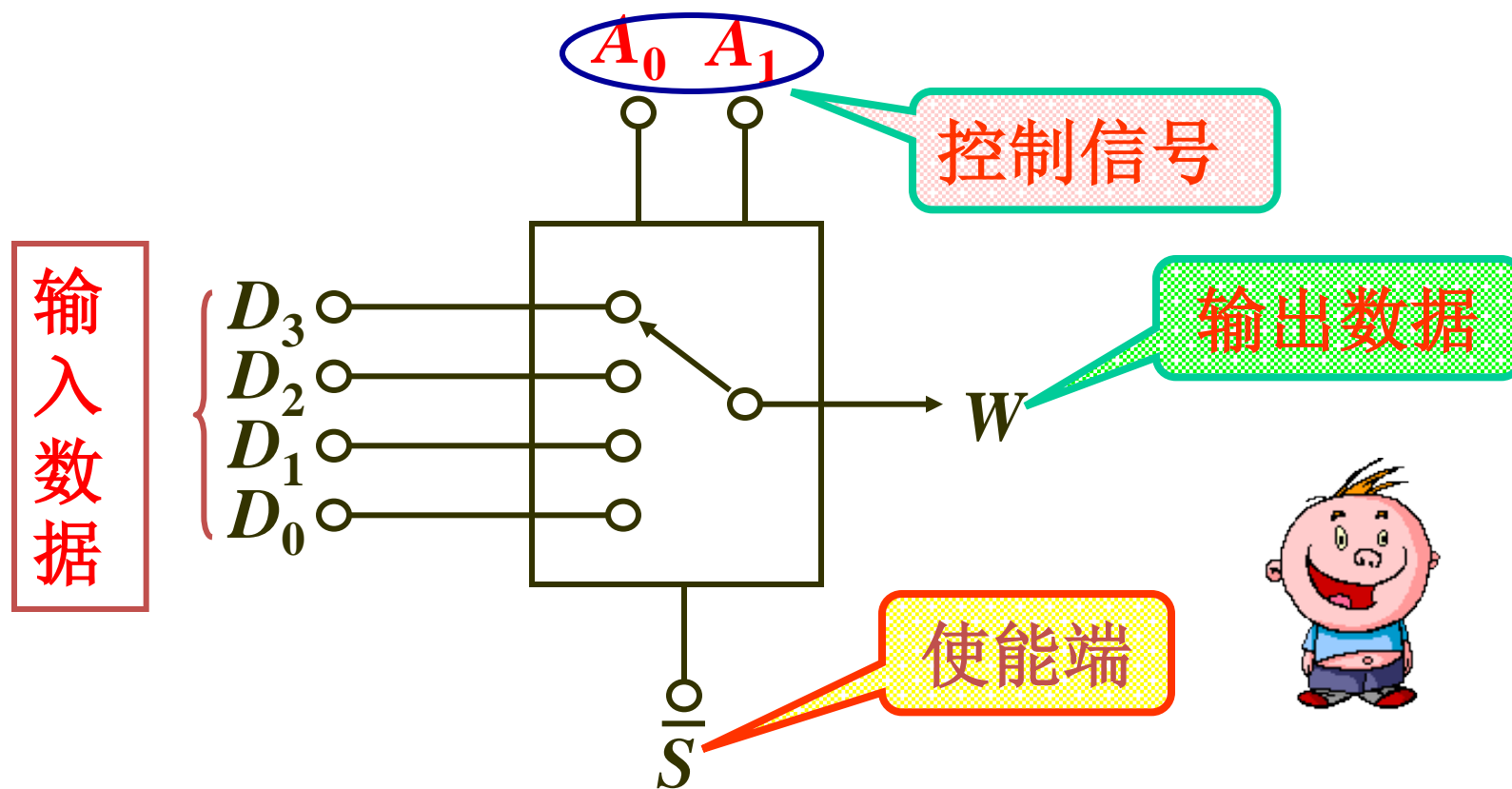
在数字电路中，当需要进行远距离多路数字传输时，为了减少传输线的数目，发送端常通过一条公共传输线，用多路选择器分时发送数据到接收端，接收端利用多路分配器分时将数据分配给各路接收端，其原理如图所示。

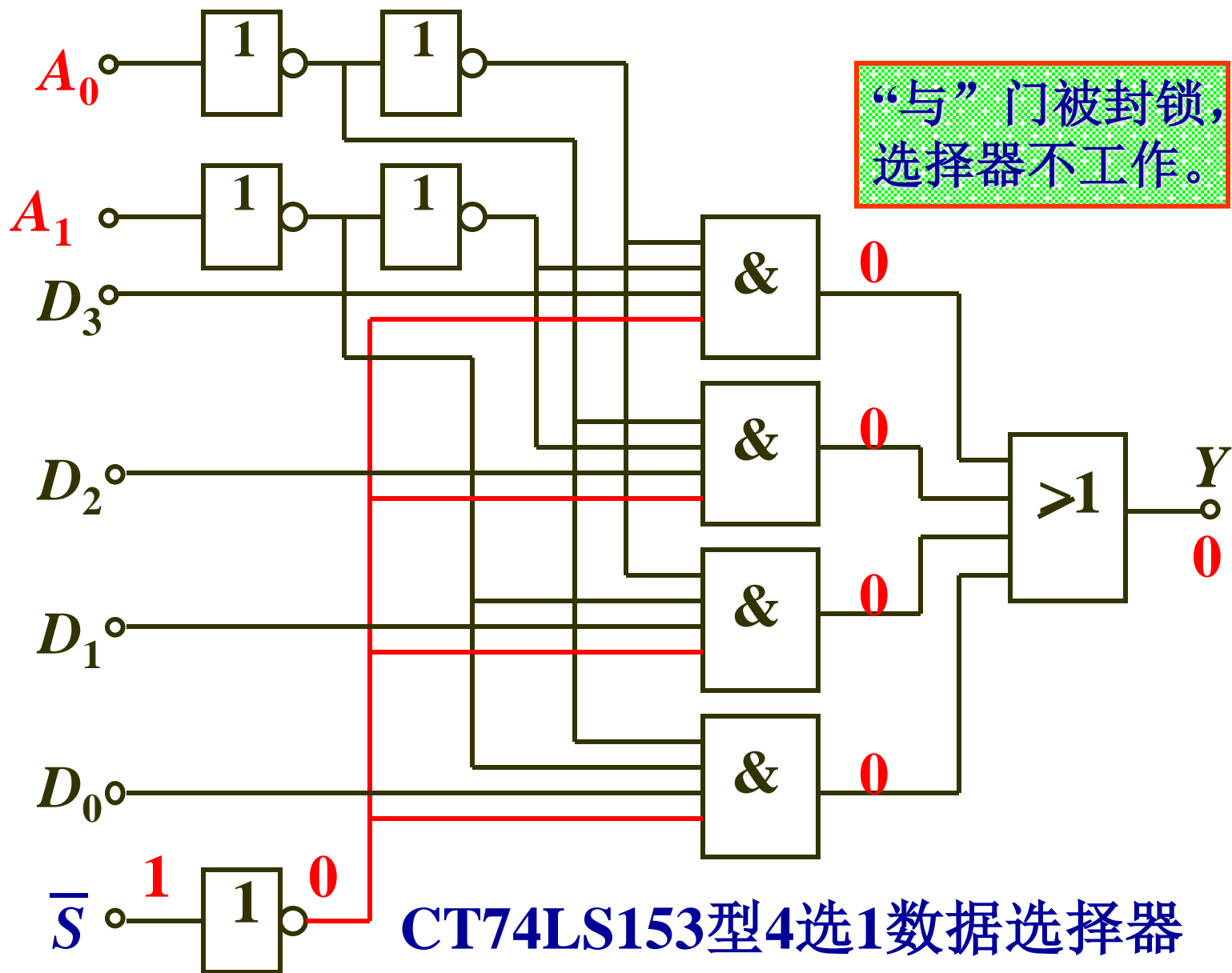


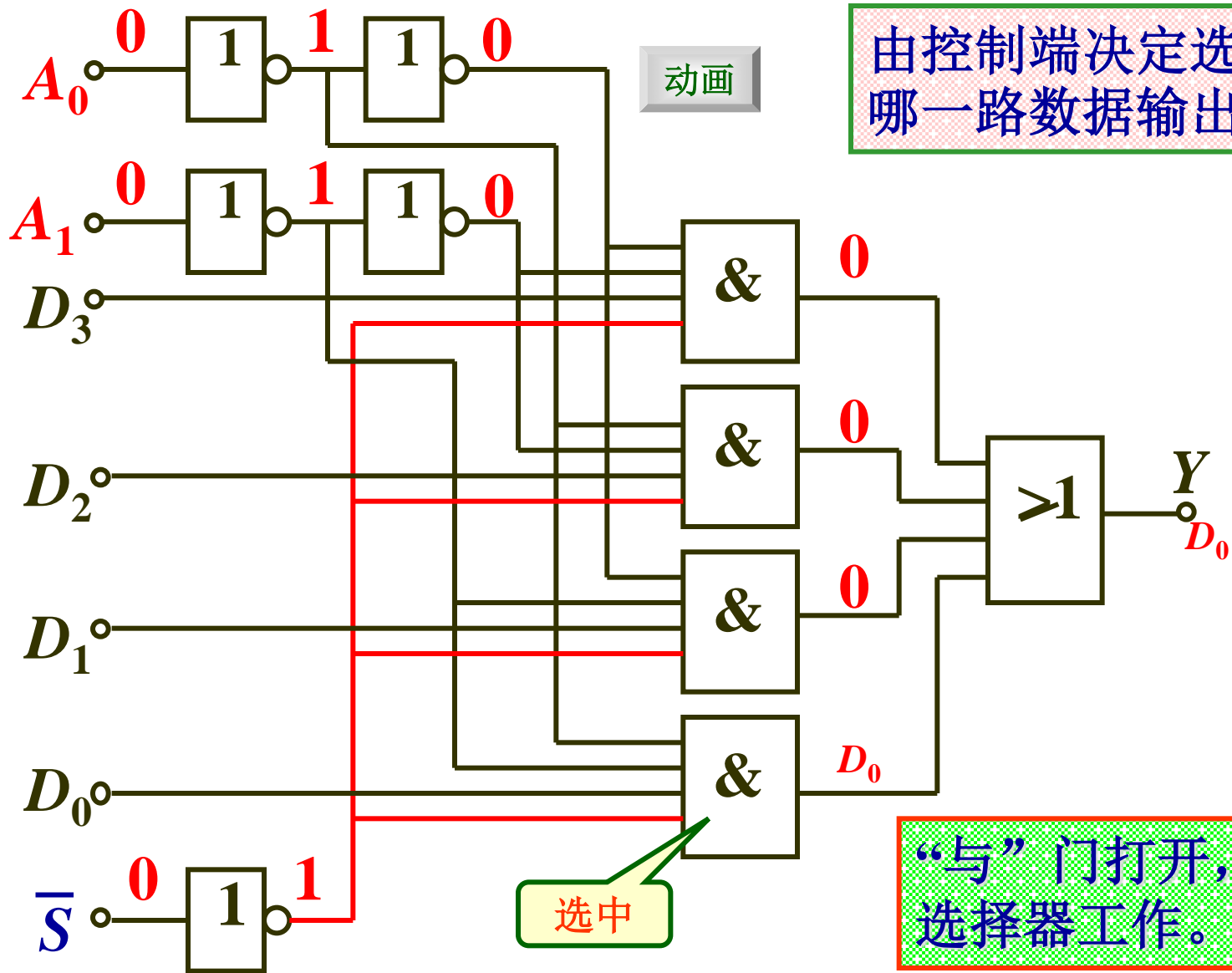
## 2.4.1 数据选择器(data selector, multiplexer)

从多路数据中选择其中所需要的一路数据输出。

例：四选一数据选择器







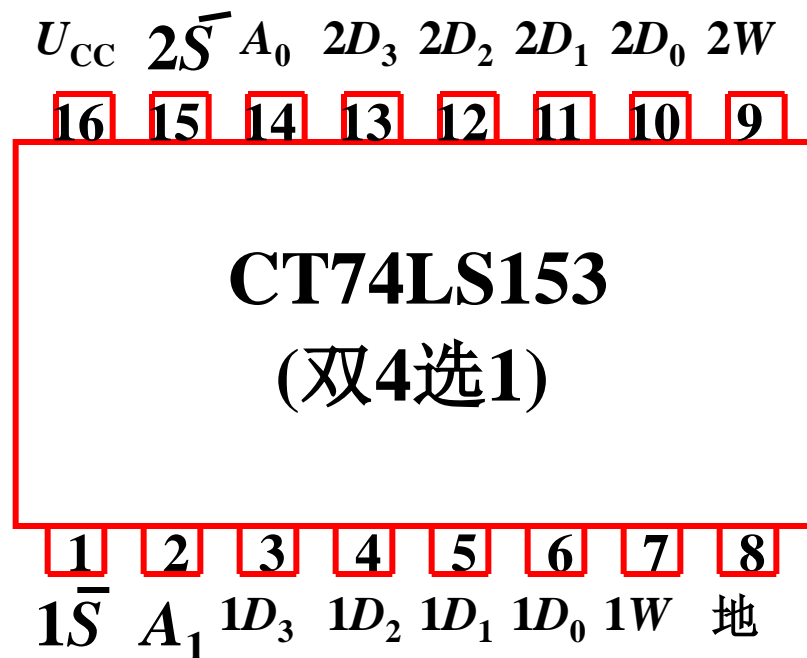
CT74LS153型4选1数据选择器

由逻辑图写出逻辑表达式

$$Y = D_0 \overline{A_1} \overline{A_0} S + D_1 \overline{A_1} A_0 S + D_2 A_1 \overline{A_0} S + D_3 A_1 A_0 S$$

## CT74LS153功能表

使能	选 通		输出
$\overline{S}$	$A_1$	$A_0$	$Y$
1	×	×	0
0	0	0	$D_0$
0	0	1	$D_1$
0	1	0	$D_2$
0	1	1	$D_3$



$\overline{S}_1 = 1$ 时,  $Y = 0$ , 禁止选择;  $\overline{S}_1 = 0$ 时, 正常工作。

# 74LS151型8选1数据选择器

## CT74LS151功能表

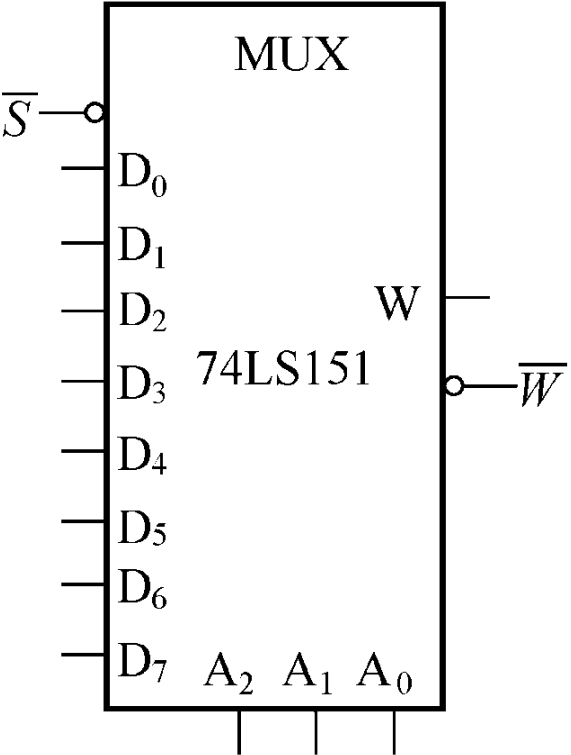
选通	选 择			输出
$\overline{S}$	$A_2$	$A_1$	$A_0$	$W$
1	×	×	×	0
0	0	0	0	$D_0$
0	0	0	1	$D_1$
0	0	1	0	$D_2$
0	0	1	1	$D_3$
0	1	0	0	$D_4$
0	1	0	1	$D_5$
0	1	1	0	$D_6$
0	1	1	1	$D_7$

$\overline{S}$  为使能端

$A_2 \sim A_0$  为地址输入端

$D_0 \sim D_7$  为数据通道输入端

$W$  和  $\overline{W}$  为互补输出端，分别输出原码和反码。



在使能端  $\bar{S}$  有效的情况下，可将输出  $W$  与地址输入  $A_2 \sim A_0$  和数据输入  $D_0 \sim D_7$  之间的关系用一个函数式来描述为：

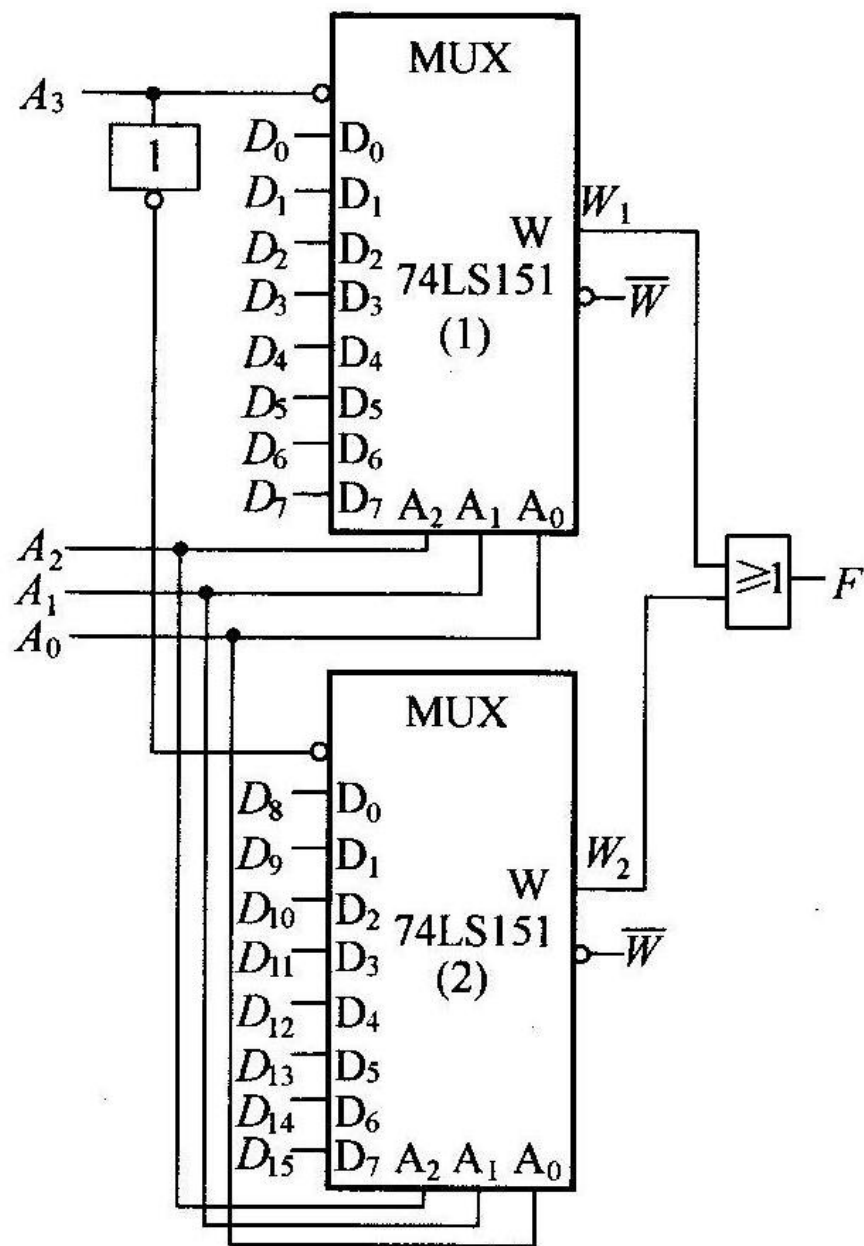
$$W = \bar{A}_2 \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_2 \bar{A}_1 A_0 D_1 + \cdots + A_2 A_1 A_0 D_7 = \sum_{i=0}^7 m_i D_i$$

其中  $m_i$  为由  $A_2$ 、 $A_1$ 、 $A_0$  所构成的最小项

当选择器输入地址为  $n$  位时

$$W = \sum_{i=0}^{2^n-1} m_i D_i$$

数据选择器的扩展：  
用两片74LS151组成  
16选一数据选择器。





例:

用74LS151型8选1数据选择器实现逻辑函数式

$$Y=AB+BC+CA$$

解: 将逻辑函数式用最小项表示

$$\begin{aligned} Y &= \overline{A}B(C + \overline{C}) + BC(A + \overline{A}) + CA(B + \overline{B}) \\ &= \overline{A}BC + A\overline{B}C + ABC\overline{C} + ABC \end{aligned}$$

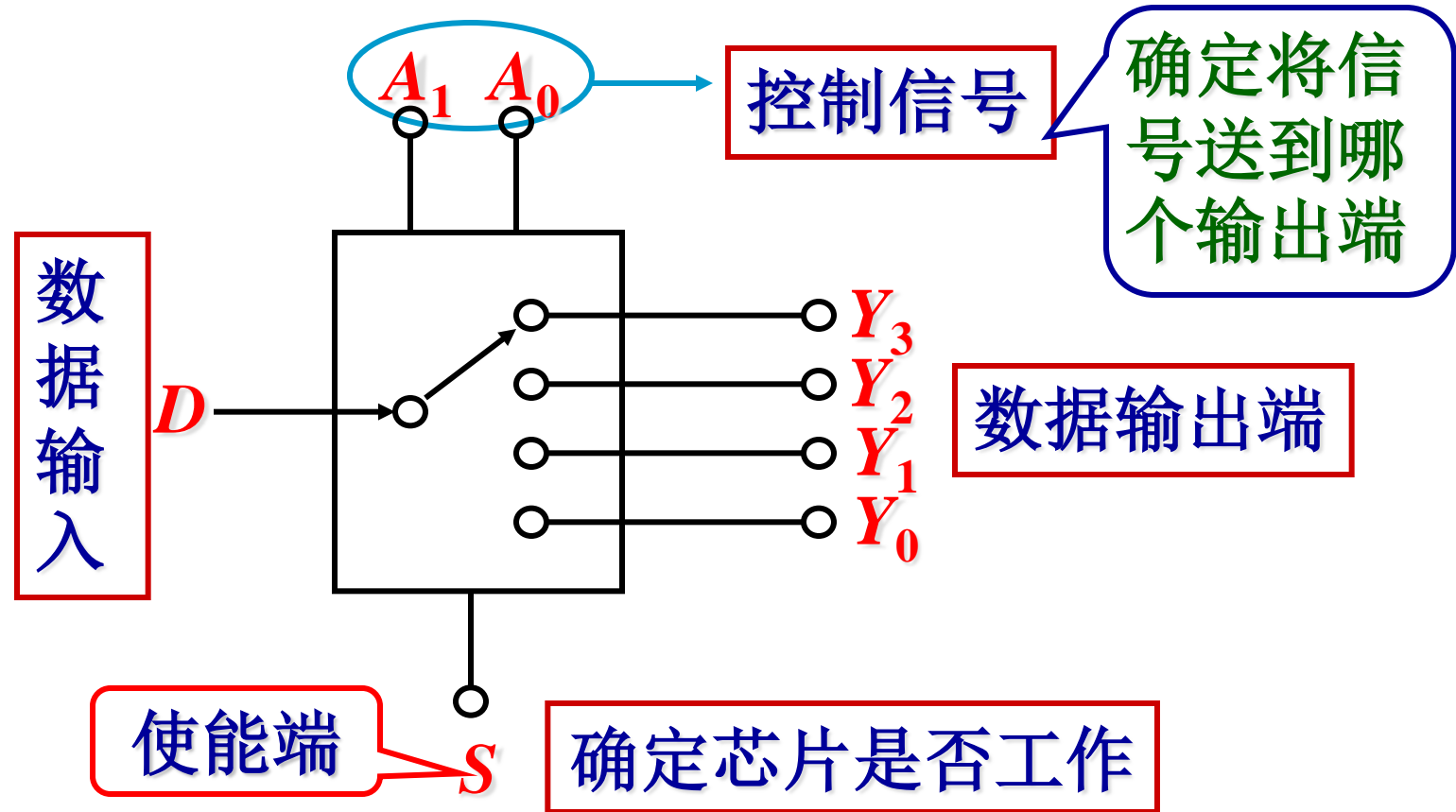
将输入变量A、B、C分别对应地接到数据选择器的选择端A<sub>2</sub>、A<sub>1</sub>、A<sub>0</sub>。由状态表可知,将数据输入端D<sub>3</sub>、D<sub>5</sub>、D<sub>6</sub>、D<sub>7</sub>接“1”,其余输入端接“0”,即可实现输出Y。



## 2.4.2 数据分配器data distributor, demultiplexer



将一个数据**分时**分送到多个输出端输出。



# 数据分配器的功能表

使能	控 制		输 出			
$\overline{S}$	$A_1$	$A_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
1	×	×	0	0	0	0
0	0	0	0	0	0	$D$
0	0	1	0	0	$D$	0
0	1	0	0	$D$	0	0
0	1	1	$D$	0	0	0

### 3 利用中规模集成芯片设计组合逻辑电路

- 如果需要实现的逻辑函数表达式与某种中规模集成器件的逻辑函数表达式形式上完全一致，则使用这种器件最方便。
- 如果需要实现的逻辑函数其变量数比中规模集成器件的输入变量少，则只需将中规模集成器件的多余输入端作适当的处理(固定为1或固定为0)。
- 如果需要实现的逻辑函数其变量数比中规模集成器件的输入变量多，则可通过将中规模集成器件进行扩展的方法来实现。

## 3.1利用译码器来实现组合逻辑函数

- 利用译码器实现一组多输出函数

$$\begin{cases} F_1 = \overline{A}B + \overline{B}C + AC \\ F_2 = A\overline{C} + BC + \overline{A}C \\ F_3 = AB + \overline{A}BC + B\overline{C} \end{cases}$$

解：第一步 选取相应器件

这是一组3变量的多输出函数，因此可选用3—8线译码器。译码输出  $\overline{Y}_i = \overline{m}_i$  ( $i=0,1, \dots, 7$ ),

其中  $m_i$  为由  $A_2$ 、 $A_1$ 、 $A_0$  所构成的最小项。

## 3.1利用译码器来实现组合逻辑函数

第二步 将输出函数写成最小项标准型，并进行相应变换。

$$\begin{aligned} F_1 &= \overline{A}B(C + \overline{C}) + (A + \overline{A})\overline{B}C + A(B + \overline{B})C = m_1 + m_2 + m_3 + m_5 + m_7 \\ &= \overline{\overline{m_1} \cdot \overline{m_2} \cdot \overline{m_3} \cdot \overline{m_5} \cdot \overline{m_7}} \end{aligned}$$

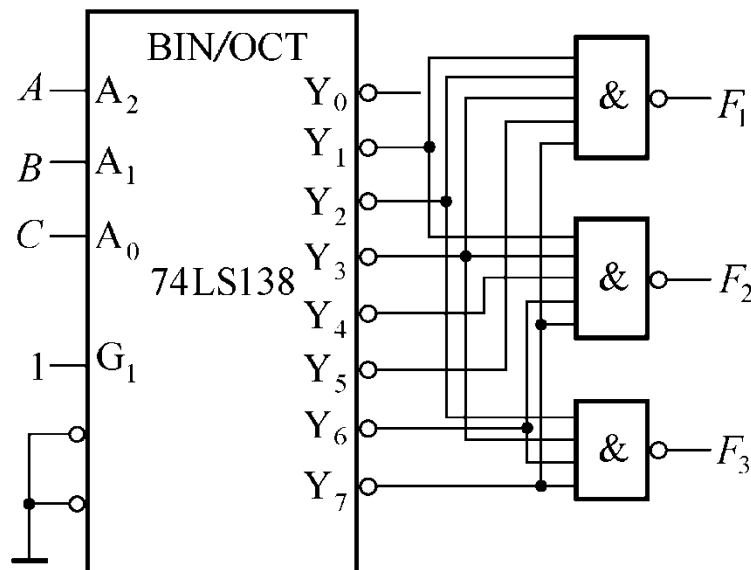
$$\begin{aligned} F_2 &= A(B + \overline{B})\overline{C} + (A + \overline{A})BC + \overline{A}(B + \overline{B})C = m_1 + m_3 + m_4 + m_6 + m_7 \\ &= \overline{\overline{m_1} \cdot \overline{m_3} \cdot \overline{m_4} \cdot \overline{m_6} \cdot \overline{m_7}} \end{aligned}$$

$$\begin{aligned} F_3 &= AB(C + \overline{C}) + \overline{A}BC + (A + \overline{A})\overline{B}\overline{C} = m_2 + m_3 + m_6 + m_7 \\ &= \overline{\overline{m_2} \cdot \overline{m_3} \cdot \overline{m_6} \cdot \overline{m_7}} \end{aligned}$$

## 3.1利用译码器来实现组合逻辑函数

### 第三步 函数对比实现

将输入变量 $A$ 、 $B$ 、 $C$ 加到译码器的地址输入端 $A_2$ 、 $A_1$ 、 $A_0$ ，利用译码器的输出附加与非门，就可以实现逻辑函数 $F_1$ 、 $F_2$ 、 $F_3$ 。



## 3.2利用数据选择器来实现组合逻辑函数

存在下列三种情况：

1、实现函数变量数小于数据选择器地址输入变量数

处理方法：将所需实现函数的变量接至数据选择器的低位地址输入端，而其高位地址接固定“0”电平。

2、函数变量数等于数据选择器地址输入变量数。

处理方法：将所需实现函数变量接至数据选择器的地址输入端，将1或0接至数据选择器相应数据输入端。

3、函数变量数( $m$ )大于数据选择器地址数( $n$ )

处理方法2个：其一是将地址变量数为 $n$ 的数据选择器扩展为地址变量数为 $m$ 的数据选择器，然后再按照第二种情况的处理方法来实现逻辑函数。其扩展方法参见数据选择器的功能扩展；



## 3.2利用数据选择器来实现组合逻辑函数

例子：

试用8选1数据选择器来实现下列逻辑函数：

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

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

$$F = \sum m(0, 1, 5, 6, 7, 9, 10, 14, 15)$$

## 应用举例

### 交通信号灯故障检测电路

交通信号灯在正常情况下，**红灯**( $R$ )亮——停车，**黄灯**( $Y$ )亮——准备，**绿灯**( $G$ )亮——通行。正常时，只有一个灯亮。如果灯全不亮或全亮或两个灯同时亮，都是故障。

解：输入信号三个，输出信号一个

灯亮 — “1” 表示，灯灭 — “0” 表示，

故障 — “1” 表示，正常 — “0” 表示，

(1) 列逻辑状态表

(2) 写出逻辑表达式

$$F = \overline{R} \overline{Y} \overline{G} + \overline{R} Y G + R \overline{Y} G \\ + R Y \overline{G} + R Y G$$

(3) 化简可得:

$$F = \overline{\overline{\overline{R} \overline{Y} \overline{G}}} + YG + RG + RY$$

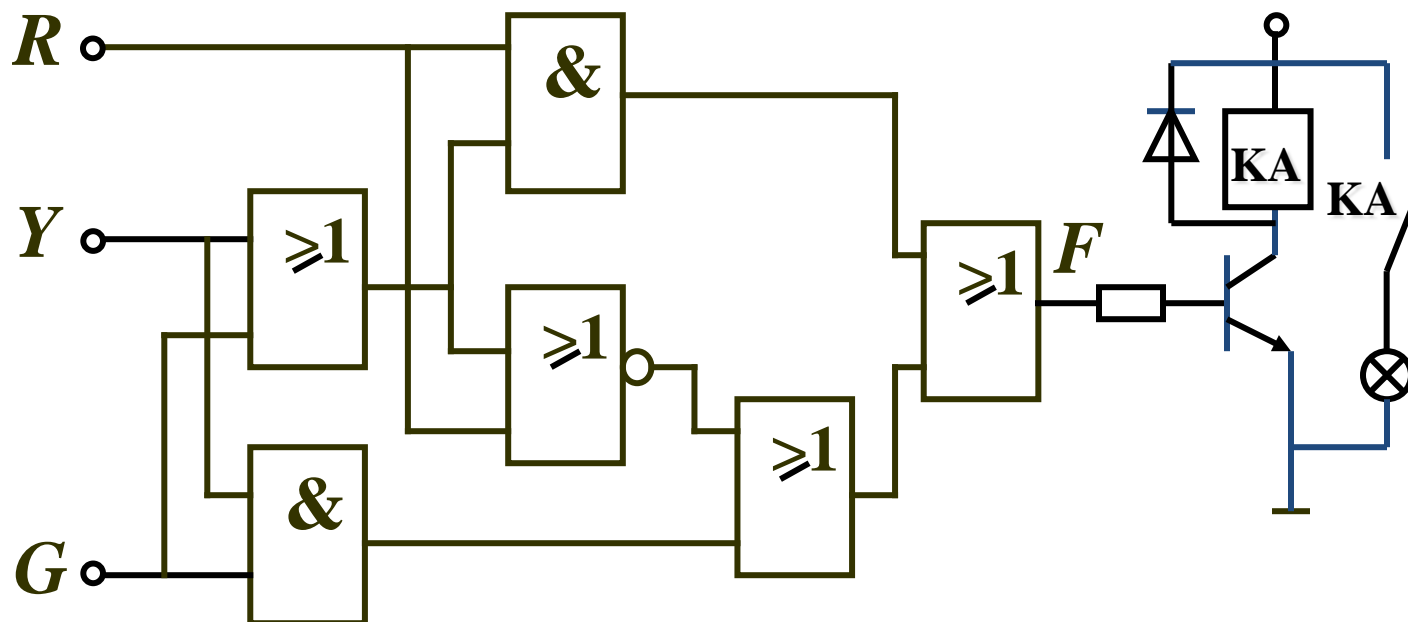
为减少所用门数,将上式变换为:

$$F = \overline{\overline{\overline{R} \overline{Y} \overline{G}}} + R(Y + G) + RG \\ = \overline{R + Y + G} + R(Y + G) + YG$$

<i>R</i>	<i>Y</i>	<i>G</i>	<i>F</i>
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



#### (4) 画逻辑图



发生故障时， $F=1$ ，晶体管导通，继电器KA通电，其触点闭合，故障指示灯亮。