

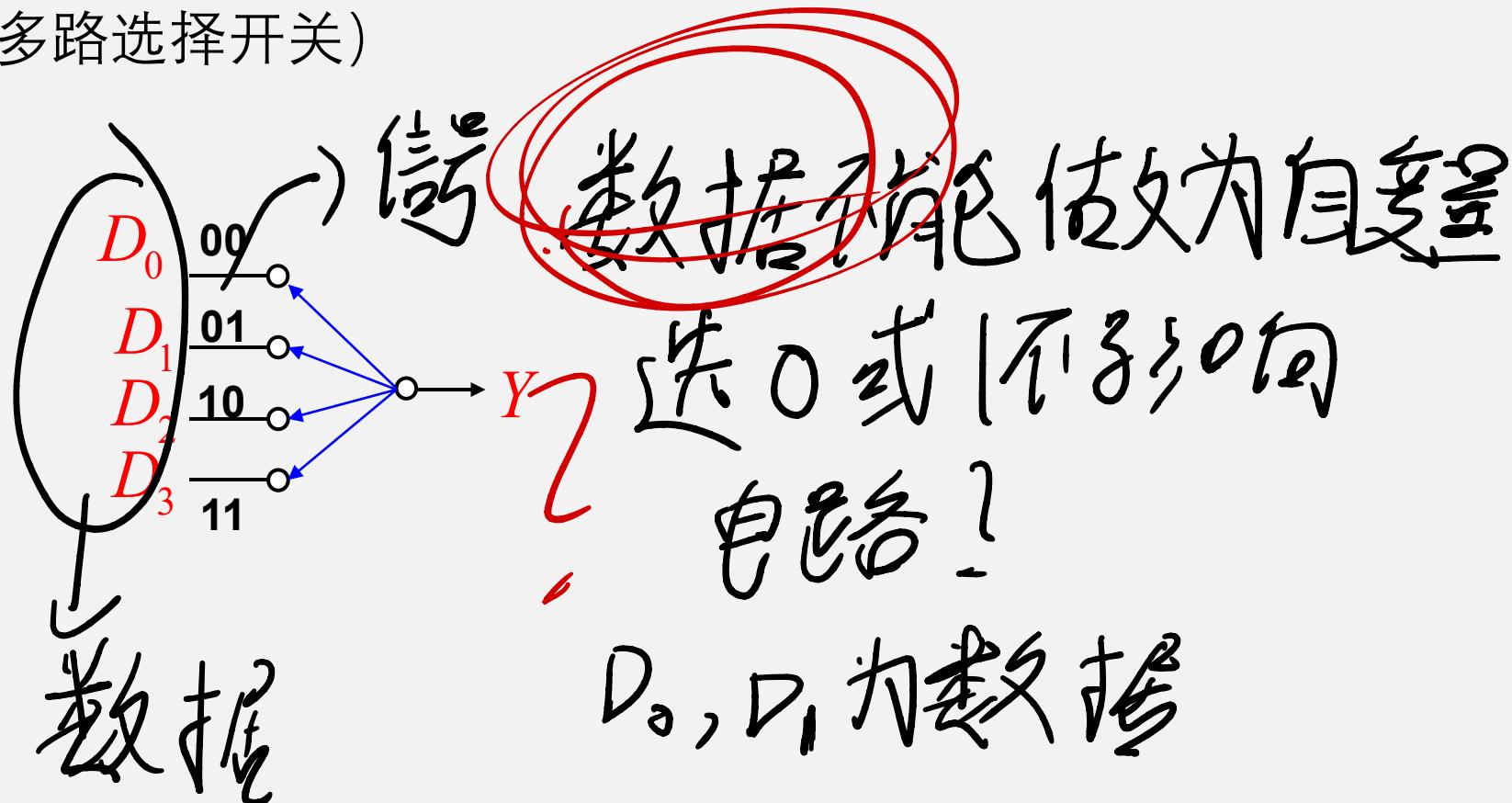


第四章 组合逻辑电路设计(三)

秦磊华 计算机学院

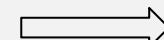
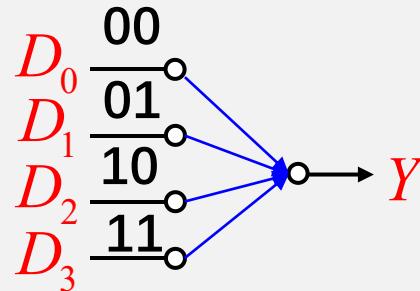
1. 多路选择器的基本功能

从一组输入数据中，选择出某一个数据，完成这种功能的逻辑电路称为数据选择器（或称为多路选择开关）



4.7 多路选择器(Multiplexer)设计

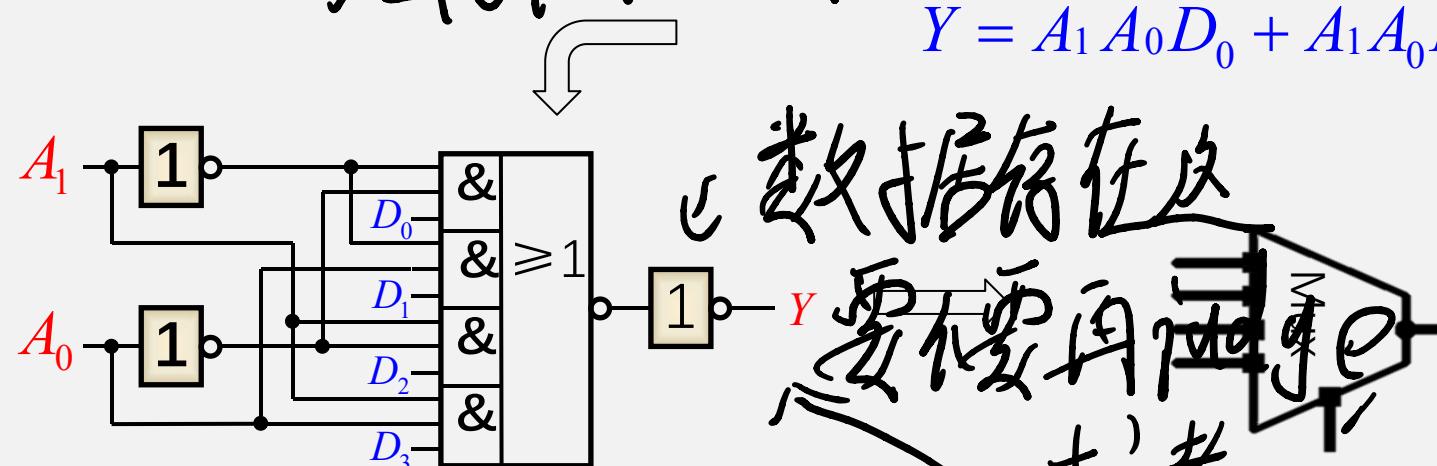
2. 4路数据选择器的设计 (MUX)



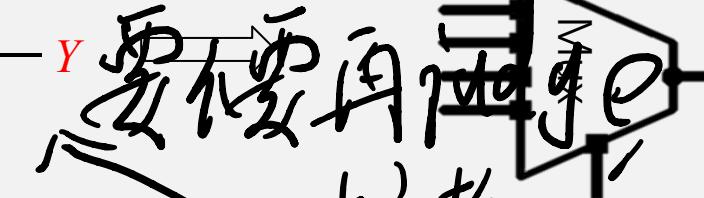
D	A ₁	A ₀	Y
D ₀	0	0	D ₀
D ₁	0	1	D ₁
D ₂	1	0	D ₂
D ₃	1	1	D ₃

多出引脚接地，否则有干扰。

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



数据存在此

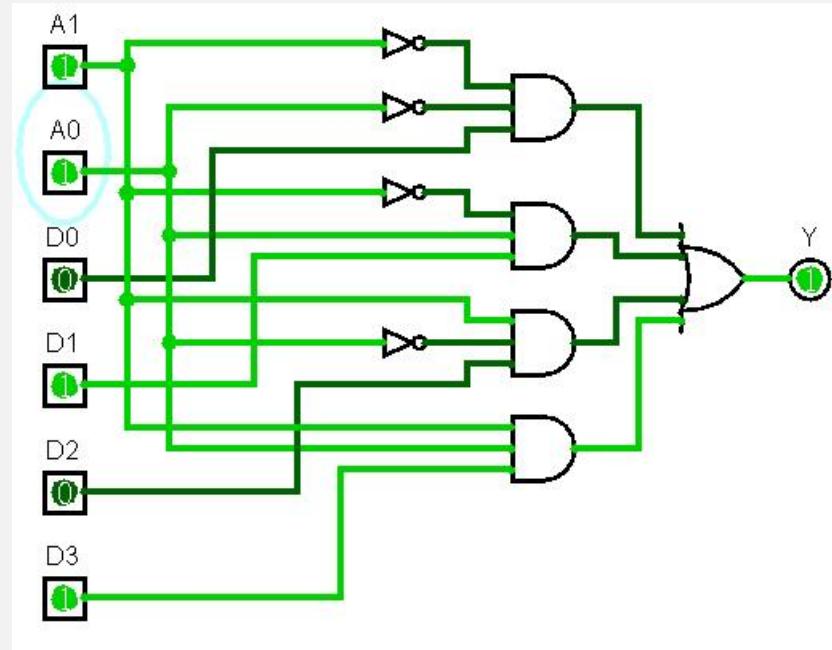


要使用两个
或者

问题如果4路
4个data,不想
输入data从哪里

4.7 多路选择器(Multiplexer)设计

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



4.7 多路选择器(Multiplexer)设计

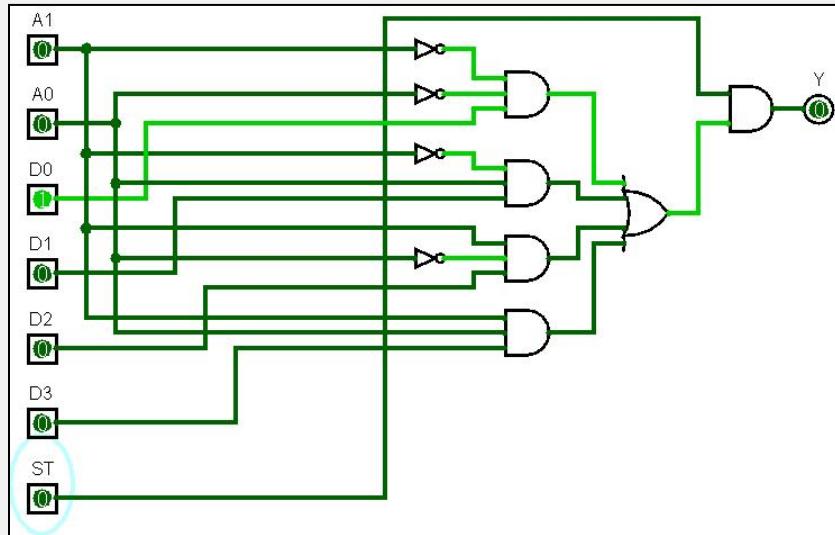
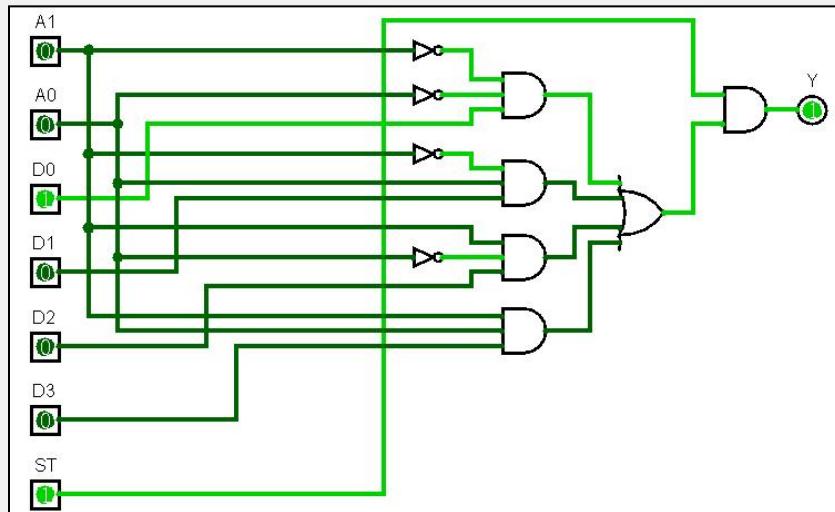
\overline{ST}_1	A_1	A_0	Y_1
1	X	X	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3

$$Y = \overline{ST}_1(\overline{A_1}\overline{A_0}D_0 + \overline{A_1}A_0D_1 + A_1\overline{A_0}D_2 + A_1A_0D_3)$$



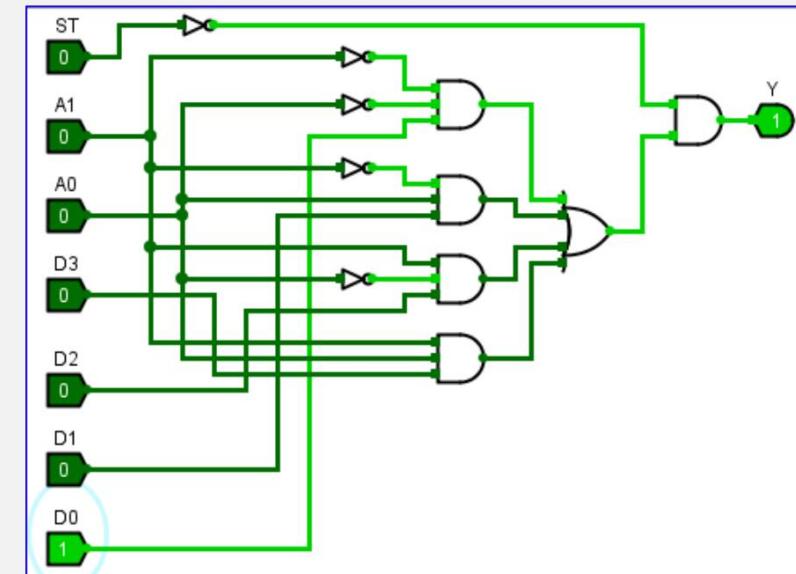
Enable: when not 0, output is the selected input

4.7 多路选择器(Multiplexer)设计



使能信号为正

\overline{ST}_1	A_1	A_0	Y_1
1	X	X	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3



使能信号为负

4.7 多路选择器(Multiplexer)设计

3. 带使能和可扩展功能的8路数据选择器设计

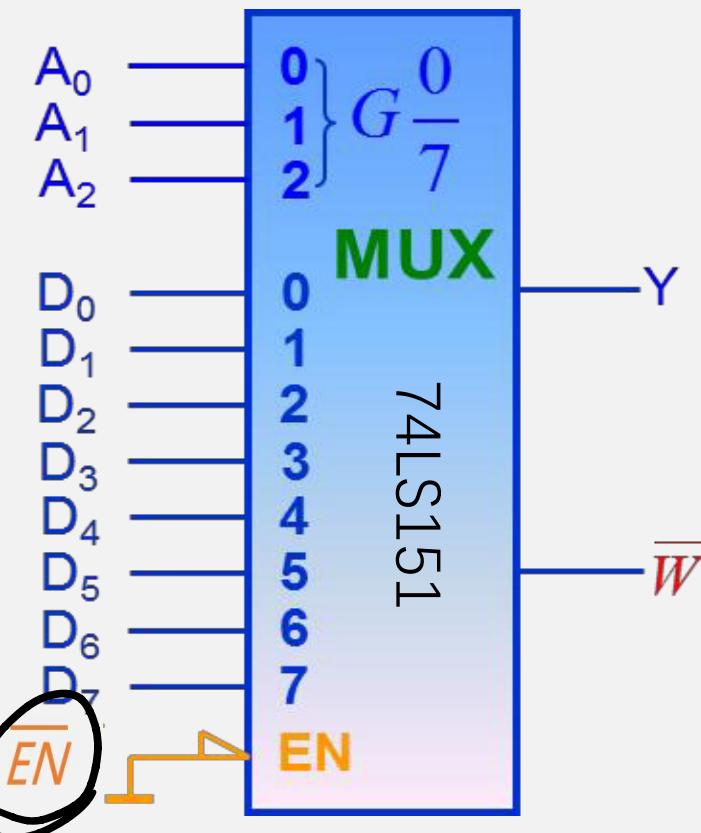
\overline{EN}	A_2	1	0			—
1	X	X	X	0	1	
0	0	0	0	D_0	\overline{D}_0	
0	0	0	1	D_1	\overline{D}_1	
0	0	1	0	D_2	\overline{D}_2	
0	0	1	1	D_3	\overline{D}_3	
0	1	0	0	D_4	\overline{D}_4	
0	1	1	1	D_5	\overline{D}_5	
0	1	0	0	D_6	\overline{D}_6	
0	1	1	1	D_7	\overline{D}_7	

\overline{EN} : 选通端, 低有效。

Y, \overline{W} : 互补输出端。

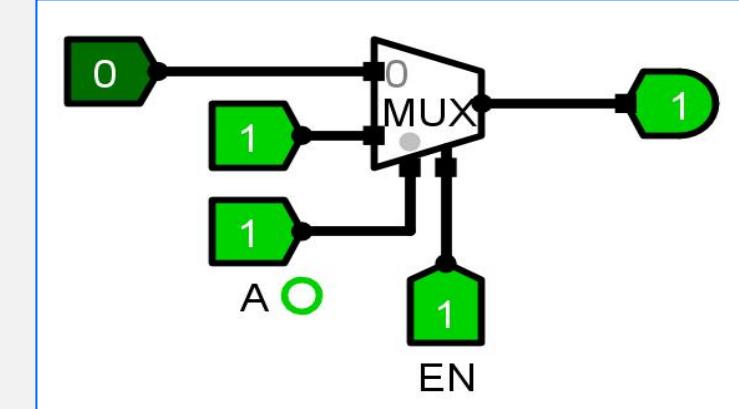
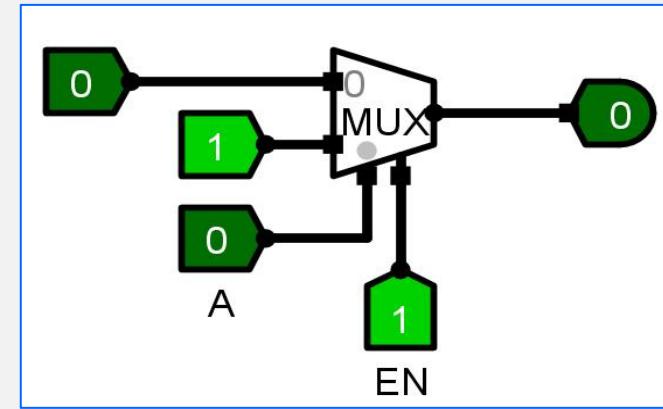
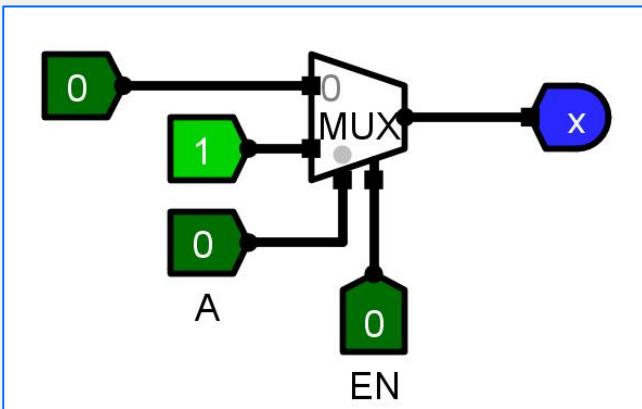
可参照4路选择器写出Y逻辑表达式

enable



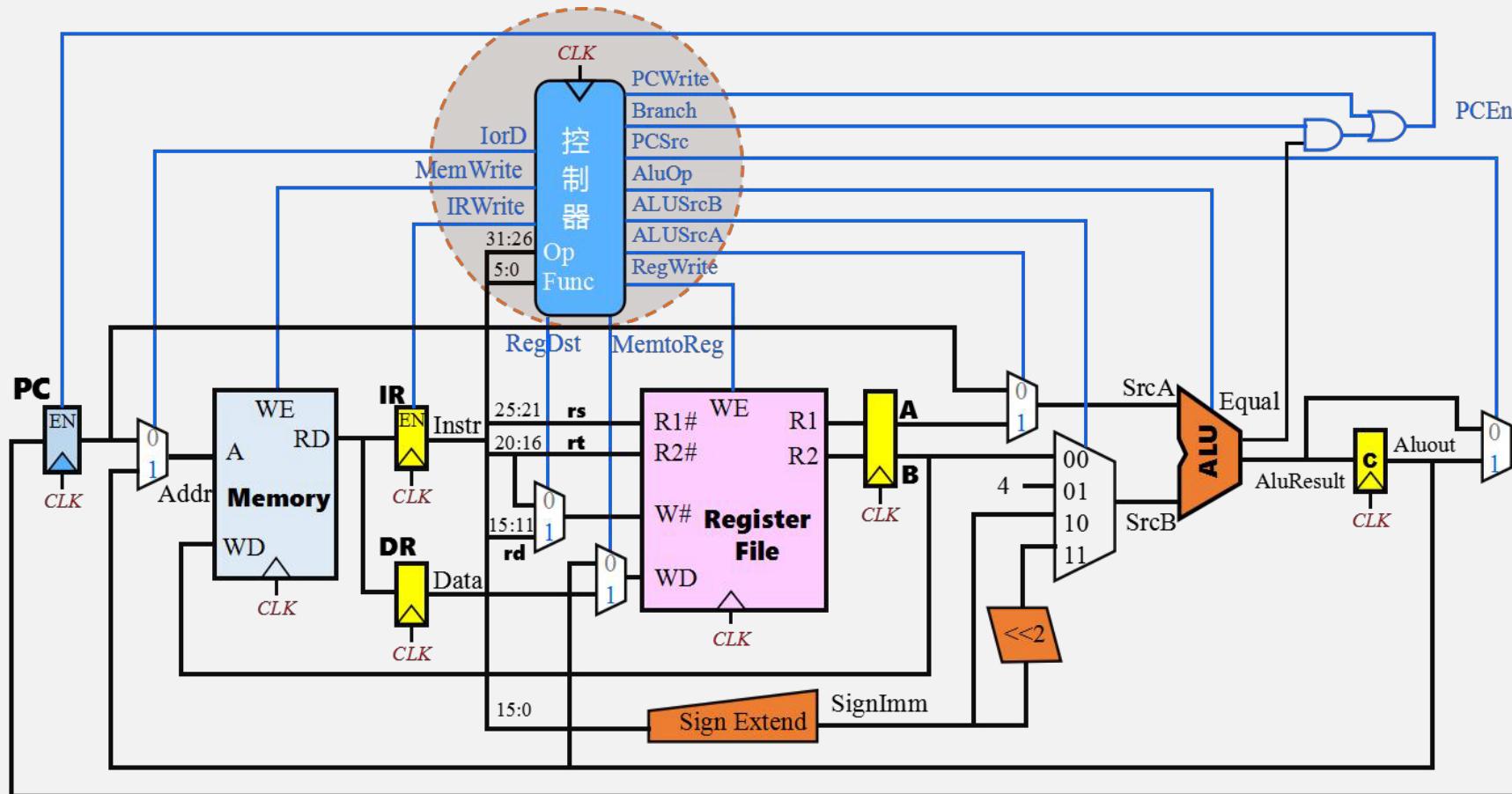
$$Y = (\overline{A}_2 \overline{A}_1 \overline{A}_0 D_0 + \overline{A}_2 \overline{A}_1 A_0 D_1 + \overline{A}_2 A_1 \overline{A}_0 D_2 + \overline{A}_2 A_1 A_0 D_3 + A_2 \overline{A}_1 \overline{A}_0 D_4 + A_2 \overline{A}_1 A_0 D_5 + A_2 A_1 \overline{A}_0 D_6 + A_2 A_1 A_0 D_7) \overline{EN}$$

4.7 多路选择器(Multiplexer)设计



4.7 多路选择器(Multiplexer)设计

4.数据选择器的应用

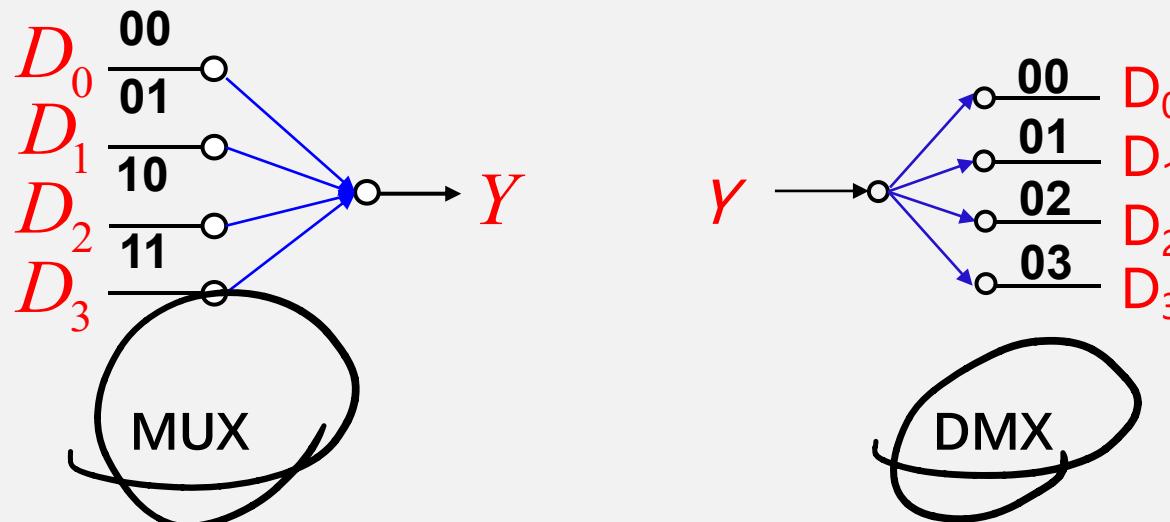


同一目标有多个数据来源时，在其入口处需使用多路选择器

4.8 多路分配器(解复用器 Demultiplexer)

1. 多路分配器的基本功能

将1个输入数据，根据需要传送到m个输出端的任何一个输出端的电路，称为数据分配器、多路分配器或解复用器，其逻辑功能正好与多路选择器相反。



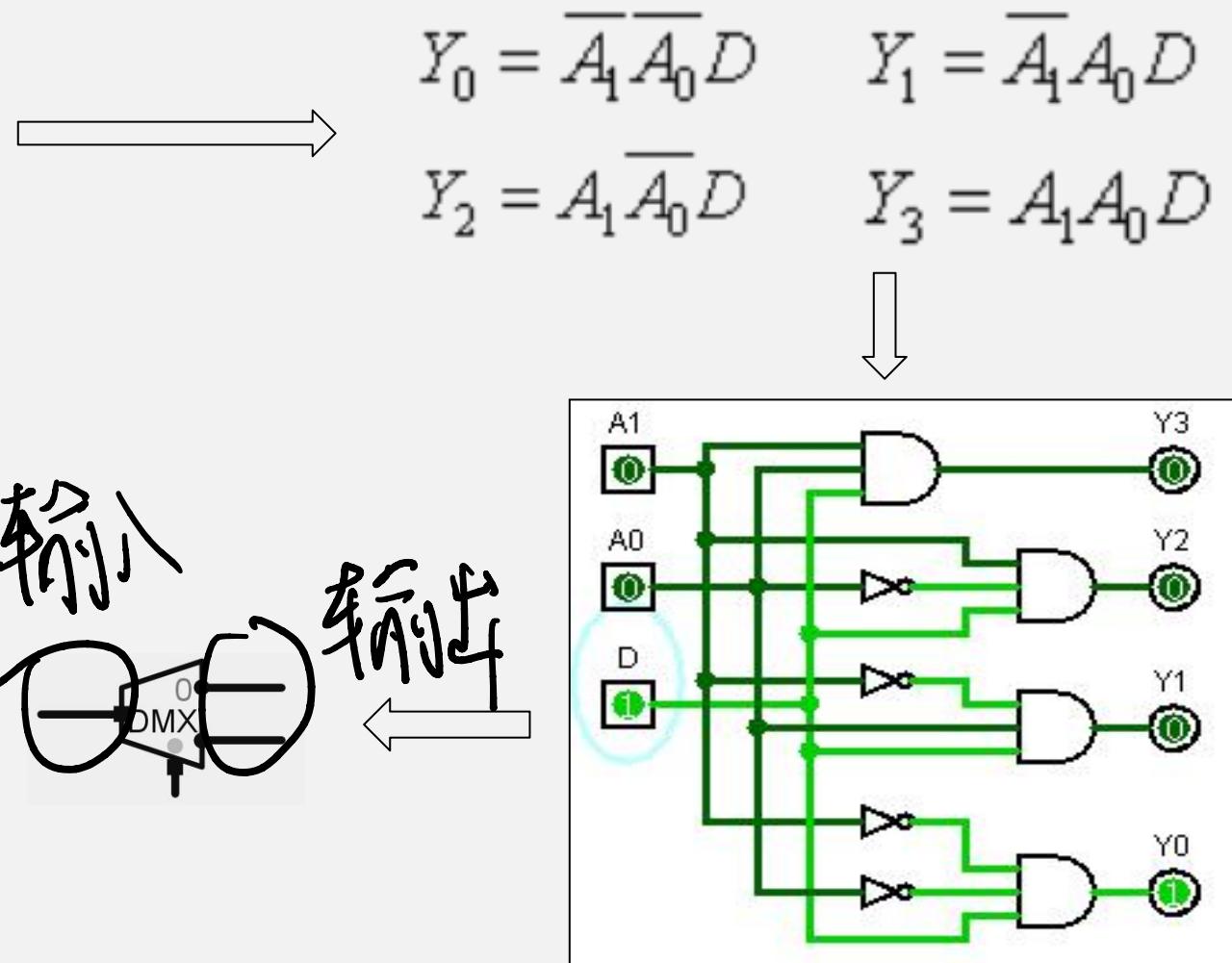
4.8 多路分配器(解复用器 Demultiplexer)

2. 多路分配器的设计

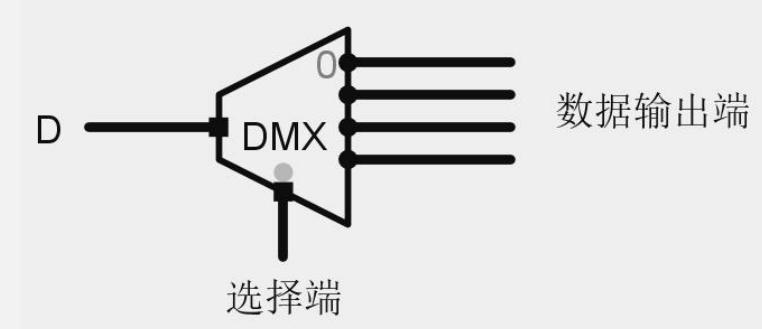
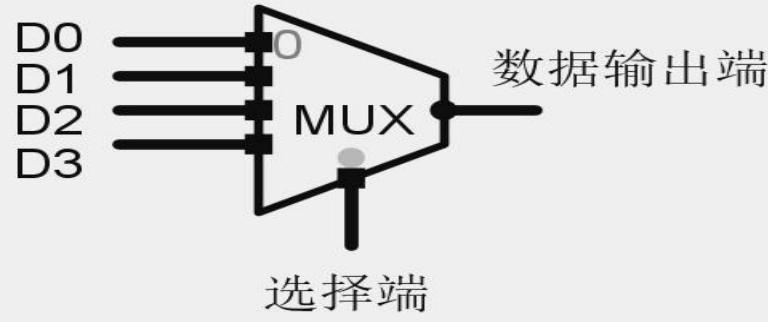
A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

A_1	A_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

多路选择器真值表



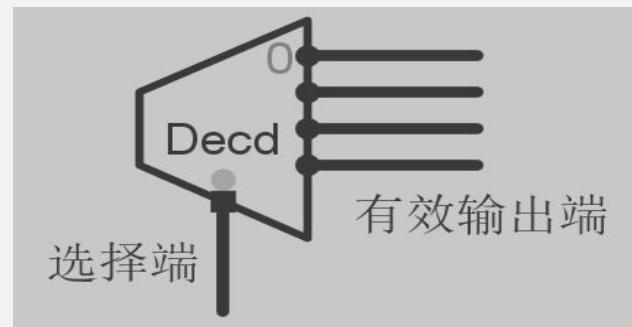
4.9 多路选择器、多路分配器、译码器比较



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

D 可以取
不同值
从而产生不同值

$$Y_0 = \overline{A_1} \overline{A_0} D \quad Y_1 = \overline{A_1} A_0 D \quad Y_2 = A_1 \overline{A_0} D \quad Y_3 = A_1 A_0 D$$



$$Y_0 = \overline{A_1} \overline{A_0} \quad Y_1 = \overline{A_1} A_0 \quad Y_2 = A_1 \overline{A_0} \quad Y_3 = A_1 A_0$$

1. 利用变量译码器实现组合逻辑函数

A ₁	A ₀	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$Y_3 = A_1 A_0 \quad Y_2 = A_1 \bar{A}_0$$

$$Y_1 = \bar{A}_1 A_0 \quad Y_0 = \bar{A}_1 \bar{A}_0$$

一个n变量输入的变量译码器，其输出包含了n个输入变量的全部最小项。用n变量译码器加输出门就能实现任何形式的输入变量不大于n的组合逻辑函数。

4.10 基于基本组合逻辑功能部件的组合逻辑设计

例1 用译码器实现一组多输出函数

$$F_1 = A\bar{B} + \bar{B}C + AC$$

$$F_2 = \bar{A}\bar{B} + B\bar{C} + ABC$$

$$F_3 = \bar{A}C + BC + A\bar{C}$$

三输入八输出 \rightarrow 逻辑门组合

解：三输入变量的多输出函数，用3-8译码器实现

将多输出函数写成最小项之和形式,再配合适当的逻辑门即可。

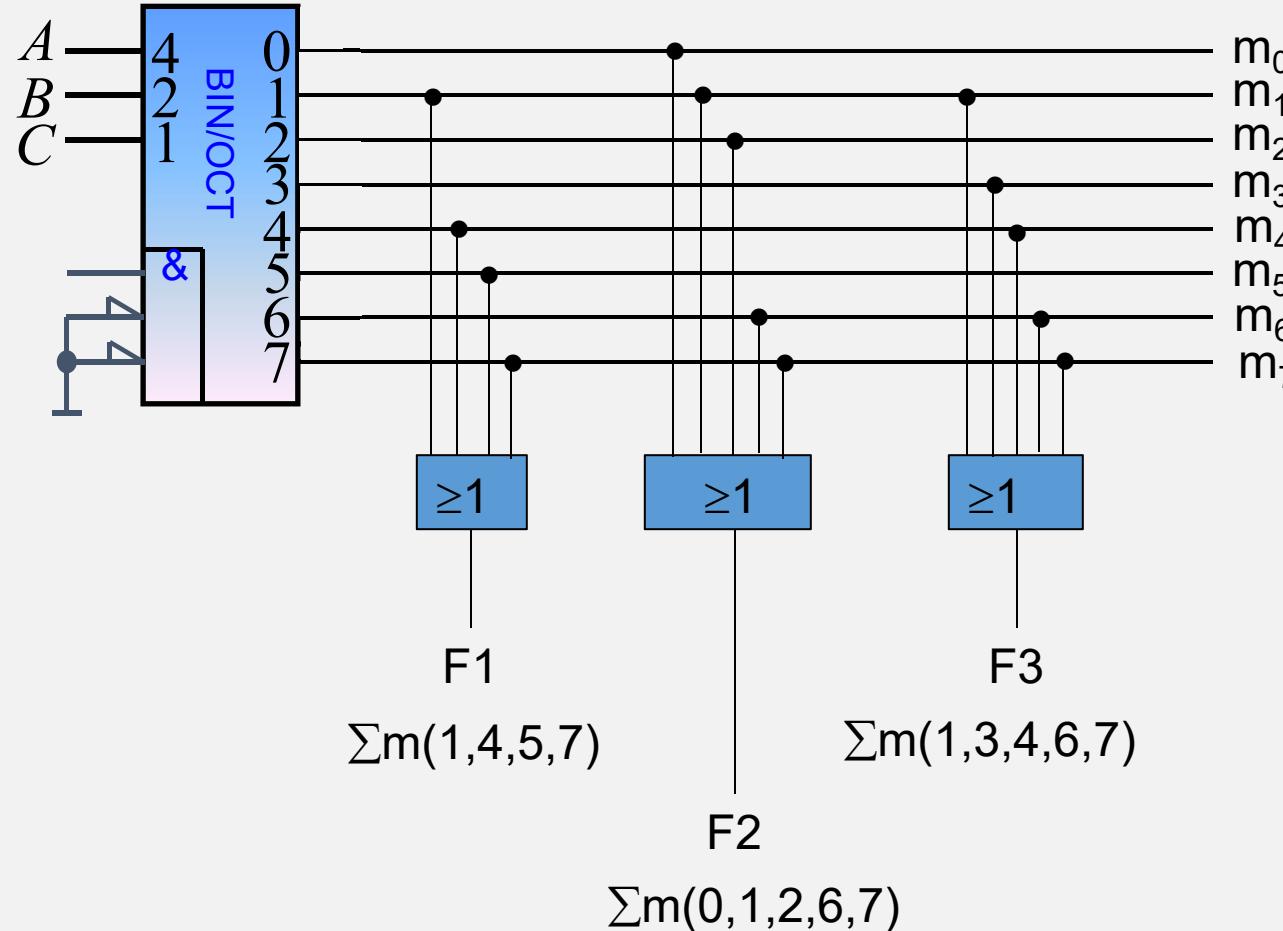
$$F_1 = A\bar{B} + \bar{B}C + AC = \sum m(1, 4, 5, 7)$$

$$F_2 = \bar{A}\bar{B} + B\bar{C} + ABC = \sum m(0, 1, 2, 6, 7)$$

$$F_3 = \bar{A}C + BC + A\bar{C} = \sum m(1, 3, 4, 6, 7)$$

$$\begin{aligned} & A\bar{B}(C + \bar{C}) \\ & A\bar{B}(+ A\bar{B}\bar{C}) \\ & \Phi_2 \Phi_1 \end{aligned}$$

4.10 基于基本组合逻辑功能部件的组合逻辑设计



4.10 基于基本组合逻辑功能部件的组合逻辑设计

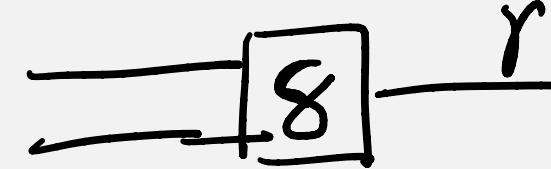
若译码器是以反变量形式输出, 即输出的是 \bar{m}_i , 则:

$$F_1 = A\bar{B} + \bar{B}C + AC = m_1 + m_4 + m_5 + m_7$$

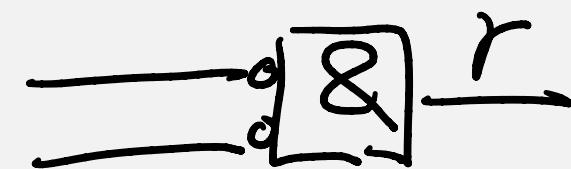
$$= \overline{\overline{m}_1 + m_4 + m_5 + m_7}$$

$$= \overline{\overline{Y}_1 \cdot \overline{Y}_4 \cdot \overline{Y}_5 \cdot \overline{Y}_{\bar{m}_i}}$$

一般有



但有:



$$F_2 = \overline{A\bar{B}} + B\bar{C} + ABC = \sum m(0, 1, 2, 6, 7)$$

$$= \overline{\overline{Y}_0 \cdot \overline{Y}_1 \cdot \overline{Y}_2 \cdot \overline{Y}_6 \cdot \overline{Y}_7}$$

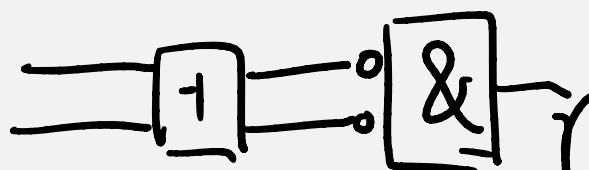
$$= \overline{\overline{m}_0 \cdot m_1 \cdot m_2 \cdot m_6 \cdot m_7}$$

我们要在前面加一个

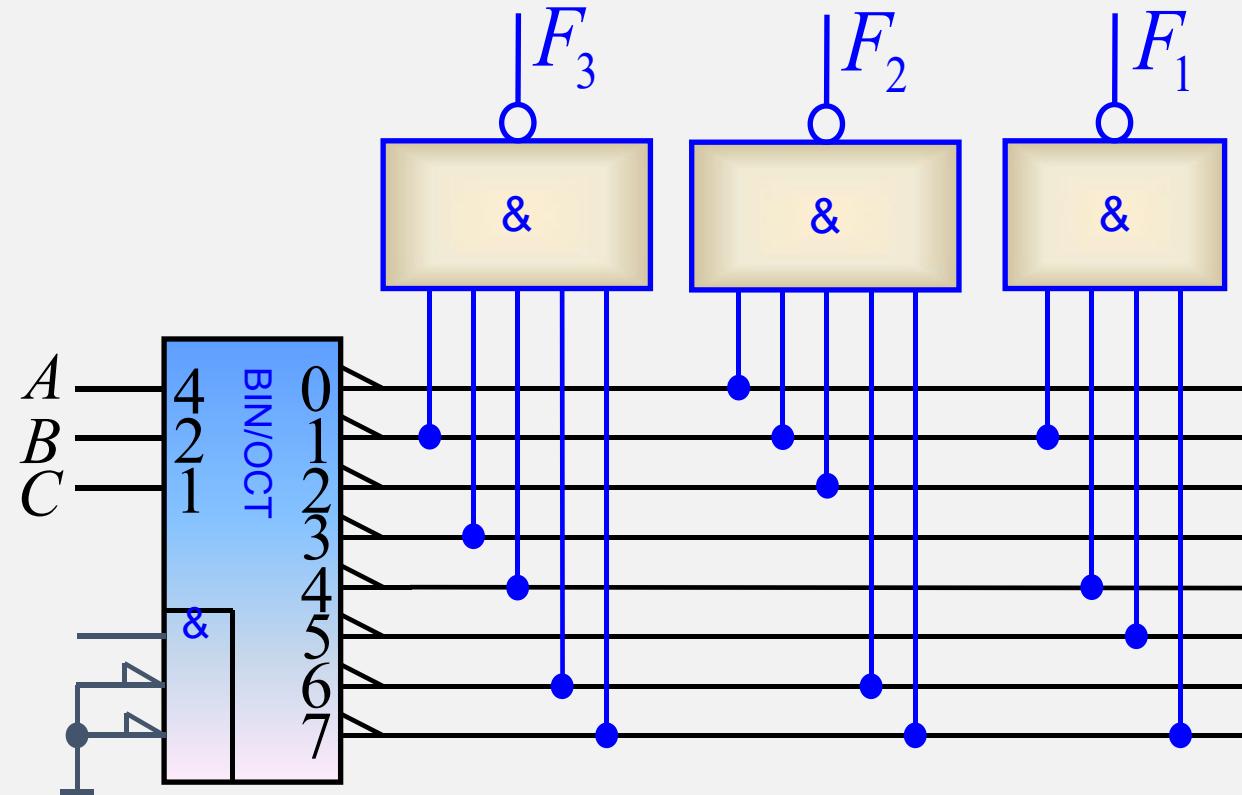
$$F_3 = \overline{AC} + BC + A\bar{C} = \sum m(1, 3, 4, 6, 7)$$

$$= \overline{\overline{Y}_1 \cdot \overline{Y}_3 \cdot \overline{Y}_4 \cdot \overline{Y}_6 \cdot \overline{Y}_7}$$

取反器, 使行
输入逻辑一致



4.10 基于基本组合逻辑功能部件的组合逻辑设计



4.10 基于基本组合逻辑功能部件的组合逻辑设计

例2：用2-4译码器和适当的逻辑门实现逻辑函数

$$F_1 = A\bar{B} + \bar{B}C + AC$$

$$F_2 = \bar{A}\bar{B} + B\bar{C} + ABC$$

$$F_3 = \bar{A}C + BC + A\bar{C}$$

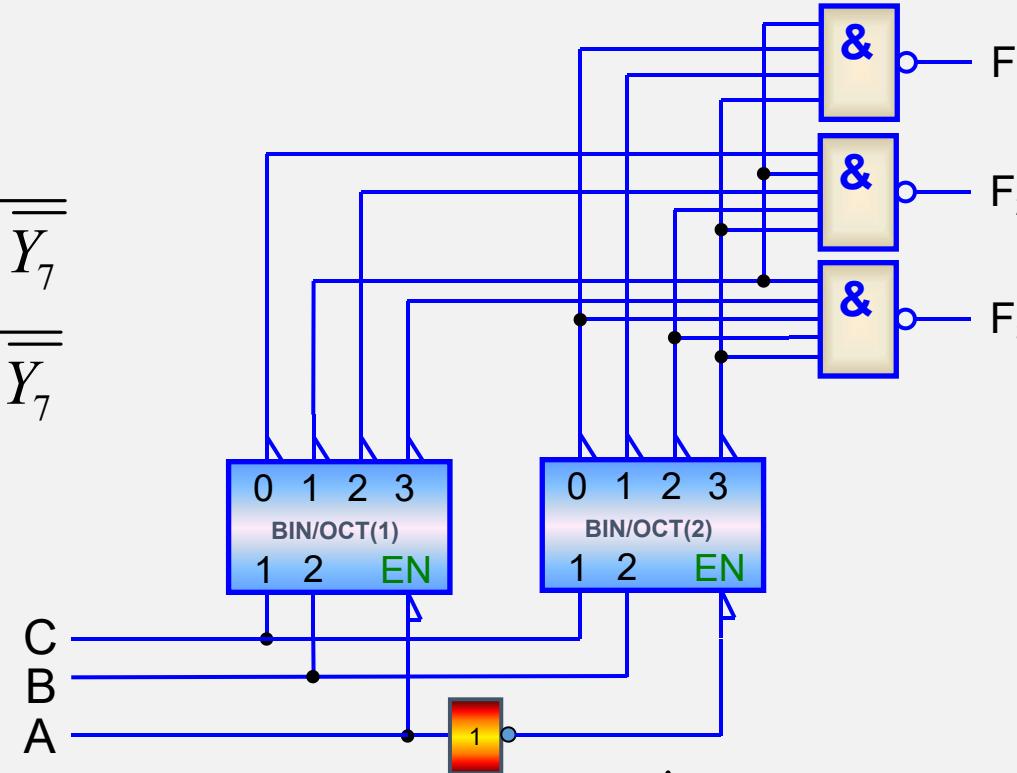
$$F_1 = A\bar{B} + \bar{B}C + AC = \overline{\overline{m}_1 \cdot \overline{m}_4 \cdot \overline{m}_5 \cdot \overline{m}_7} = \overline{\overline{Y}_1 \cdot \overline{Y}_4 \cdot \overline{Y}_5 \cdot \overline{Y}_7}$$

$$F_2 = \bar{A}\bar{B} + B\bar{C} + ABC = \overline{\overline{m}_0 \cdot \overline{m}_1 \cdot \overline{m}_2 \cdot \overline{m}_6 \cdot \overline{m}_7} = \overline{\overline{Y}_0 \cdot \overline{Y}_1 \cdot \overline{Y}_2 \cdot \overline{Y}_6 \cdot \overline{Y}_7}$$

$$F_3 = \bar{A}C + BC + A\bar{C} = \overline{\overline{m}_1 \cdot \overline{m}_3 \cdot \overline{m}_4 \cdot \overline{m}_6 \cdot \overline{m}_7} = \overline{\overline{Y}_1 \cdot \overline{Y}_3 \cdot \overline{Y}_4 \cdot \overline{Y}_6 \cdot \overline{Y}_7}$$

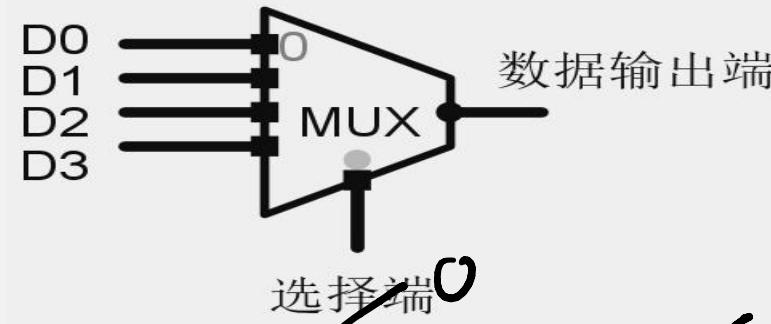
4.10 基于基本组合逻辑功能部件的组合逻辑设计

$$F_1 = \overline{\overline{Y_1} \cdot \overline{Y_4} \cdot \overline{Y_5} \cdot \overline{Y_7}}$$
$$F_2 = \overline{\overline{Y_0} \cdot \overline{Y_1} \cdot \overline{Y_2} \cdot \overline{Y_6} \cdot \overline{Y_7}}$$
$$F_3 = \overline{\overline{Y_1} \cdot \overline{Y_3} \cdot \overline{Y_4} \cdot \overline{Y_6} \cdot \overline{Y_7}}$$



A 作 enable ?

4.10 基于基本组合逻辑功能部件的组合逻辑设计



选择端 0 1 2 3 4 5 6 7

$$Y = \overline{A_1} \overline{A_0} D_0 + \overline{A_1} A_0 D_1 + A_1 \overline{A_0} D_2 + A_1 A_0 D_3$$
$$F_1 = A \overline{B} + \overline{B} C + A C = m_1 + m_4 + m_5 + m_7$$
$$F_2 = \overline{A} \overline{B} + B \overline{C} + A B C = \sum m(0, 1, 2, 6, 7)$$
$$F_3 = \overline{A} C + B C + A \overline{C} = \sum m(1, 3, 4, 6, 7)$$

0-7中的4个组
因为D是可2进制

一组取
2个不行的

本节内容完