



# 第五章 数字集成电路基本模块

---

## 5.1 组合逻辑电路



# 组合逻辑电路

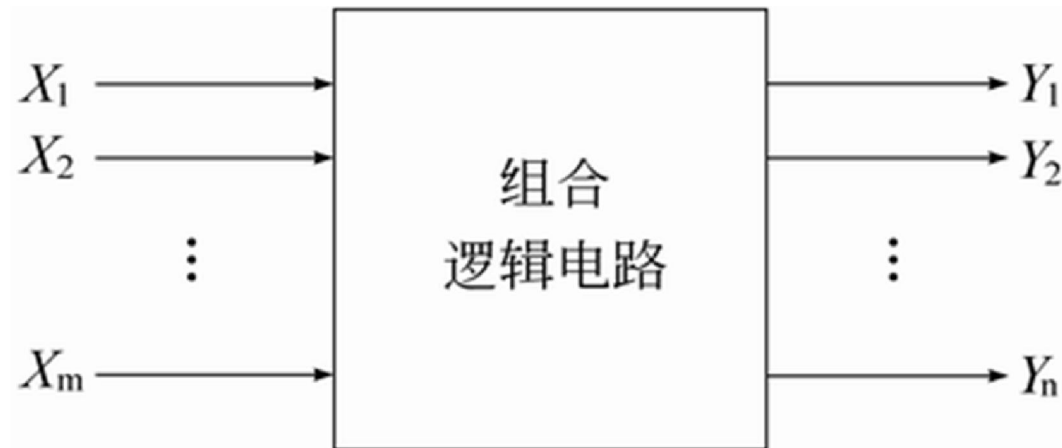
---

- 组合逻辑电路基础
- 多路器和逆多路器
- 编码器和译码器

# 组合逻辑电路

- 对于组合逻辑电路，若电路有 $m$ 个输入 $x_1, x_2, \dots, x_m$ ，产生 $n$ 个输出信号 $y_1, y_2, \dots, y_n$ ，
- 则输出与输入之间的关系可以表示为：

$$Y = F(X)$$



# 组合逻辑电路

■ 组合逻辑电路单元设计的基本过程是：

1) 列出真值表

2) 逻辑表达式

3) 适当的结构形式

4) 逻辑图和电路图

5) 每个器件的参数

6) 验证功能和性能



# 组合逻辑电路

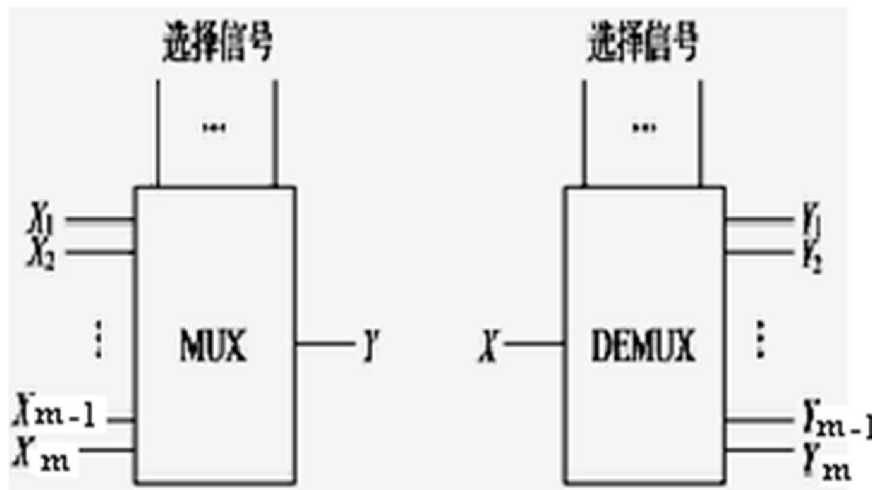
---

- 组合逻辑电路基础
- 多路器和逆多路器
- 编码器和译码器

## 多路器和逆多路器

**多路器 (MUX)**：通过控制信号从多个数据来源中选择一个传送出去。

- **逆多路器 (DEMUX)**：根据控制信号把一个数据送到多个输出端中的某一个。



- ◆ 控制信号的位数应满足： $2^n = m$

## 四选一多路器

### 真值表

s1	s0	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3

$$Y = \overline{s_1}\overline{s_0}D_0 + \overline{s_1}s_0D_1 + s_1\overline{s_0}D_2 + s_1s_0D_3$$

直接的实现方式：用一个与或非门加一个输出反相器

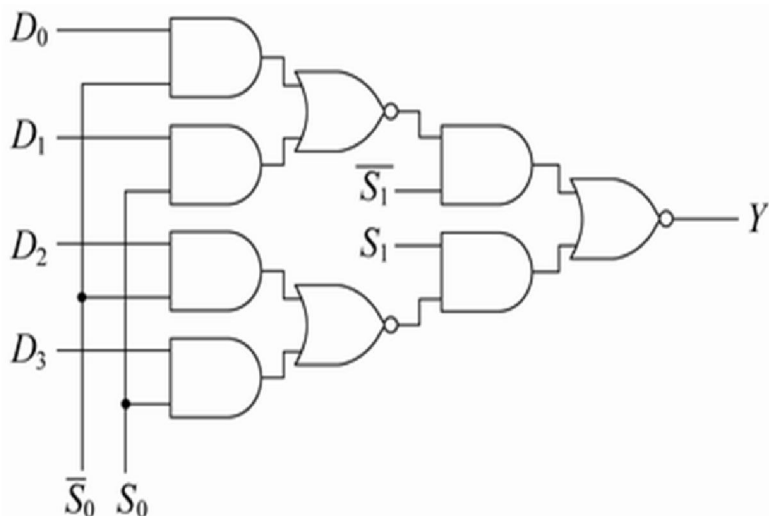
**优点：**用反相器作输出级有较好的输出驱动能力

**缺点：**第一级的与或非门扇入系数太大，将严重影响电路性能

**考虑：**总的管子数目，采用对称设计的面积？

## 四选一多路器：互补CMOS结构

- 另辟蹊径：做变换



$$\begin{aligned} Y &= \bar{s}_1 \bar{s}_0 D_0 + \bar{s}_1 s_0 D_1 + s_1 \bar{s}_0 D_2 + s_1 s_0 D_3 \\ &= \bar{s}_1 (\bar{s}_0 D_0 + s_0 D_1) + s_1 (\bar{s}_0 D_2 + s_0 D_3) \\ &= \left[ (\bar{s}_0 D_0 + s_0 D_1) + s_1 \right] \cdot \left[ (\bar{s}_0 D_2 + s_0 D_3) + \bar{s}_1 \right] \\ &= \overline{(\bar{s}_0 D_0 + s_0 D_1) \cdot s_1} + \overline{(s_0 D_2 + s_0 D_3) \cdot \bar{s}_1} \end{aligned}$$

**优点：**逻辑门简化，有利于减小面积。用3个相同逻辑门，便于版图设计

**缺点：**用与或非门做输出级，输出驱动能力差。



# 四位CMOS二选一多路器

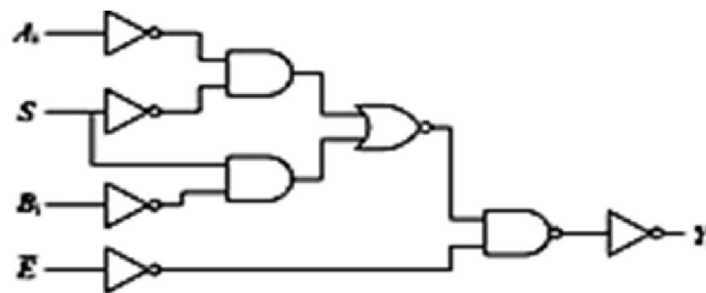
$\overline{E}$	S	Y3~Y0
1	0	0
1	1	0
0	0	A3~A0
0	1	B3~B0

$$Y_i = \overline{\overline{E}} \left( \overline{S} A_i + S B_i \right)$$

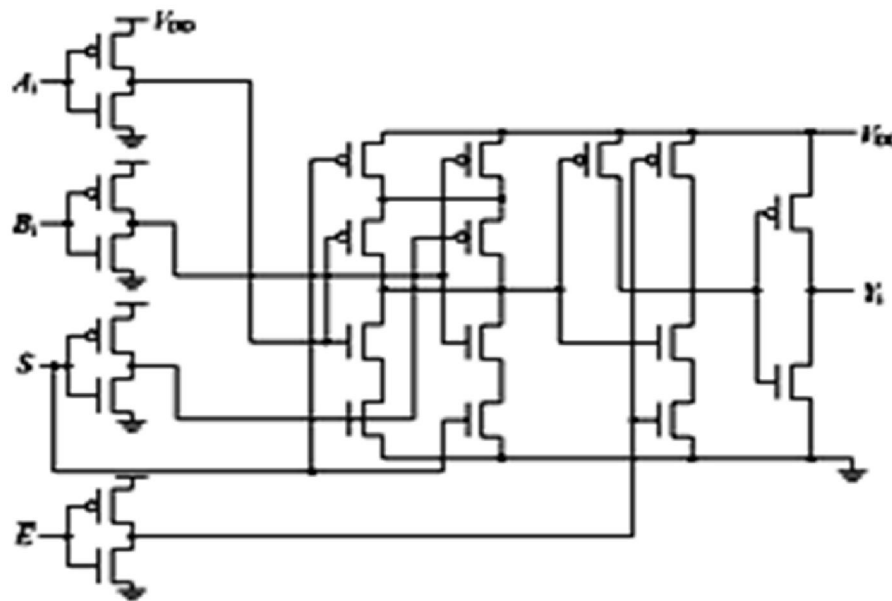
改进：加输入、输出  
缓冲器。

最终实现方案：

$$Y_i = \overline{\overline{\overline{\overline{E}}}} \cdot (\overline{\overline{\overline{\overline{S}}}} A_i + \overline{\overline{\overline{\overline{S}}}} B_i).$$



(a)



(b)

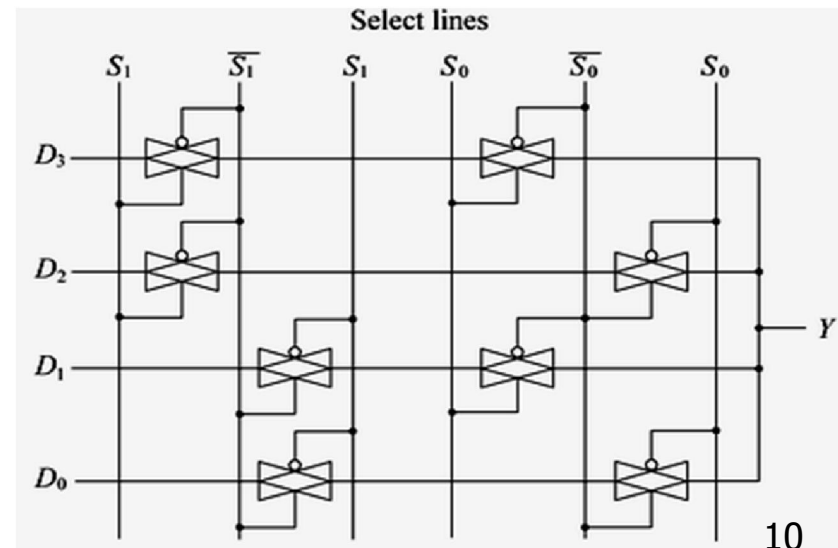
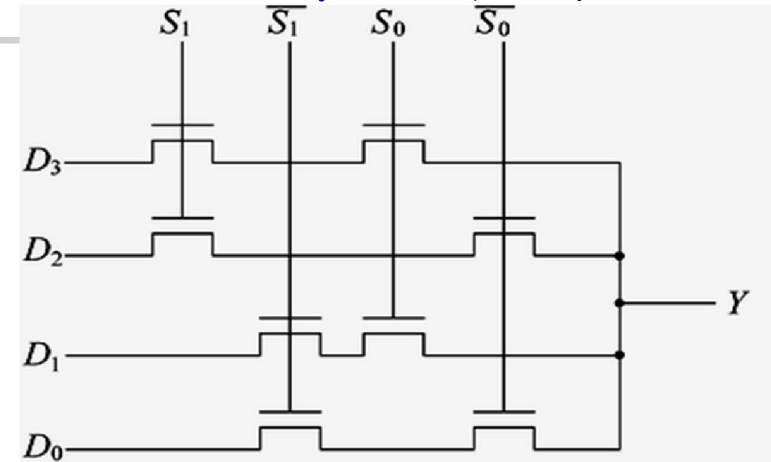
## 四选一多路器和逆多路器：传输门结构

$$Y = \overline{s_1} \overline{s_0} D_0 + \overline{s_1} s_0 D_1 + s_1 \overline{s_0} D_2 + s_1 s_0 D_3$$

- ◆ 优点：电路简化，版图规整
- ◆ 缺点：高电平有损失，输出驱动能力差。

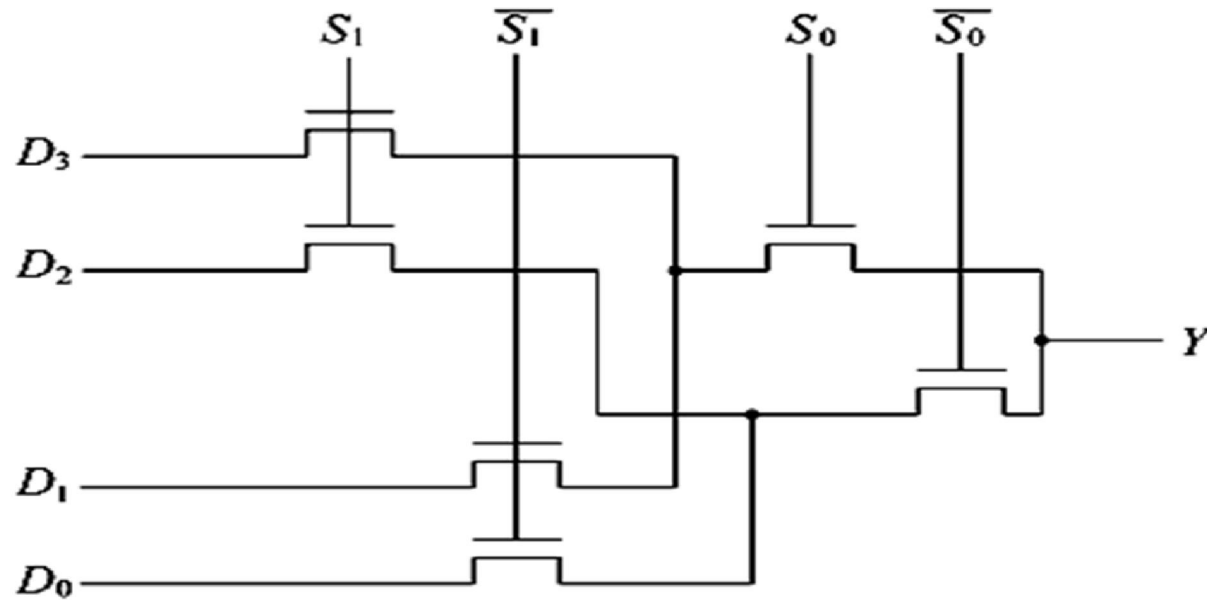
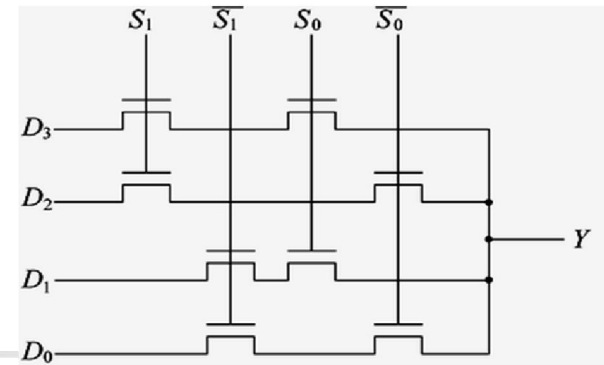
用CMOS代替NMOS，避免高电平损失；

将上述电路的I/O颠倒使用，就是逆多路器。



## 四选一多路器：传输门结构

- NMOS传输门进一步优化
- 利用两级二选一实现





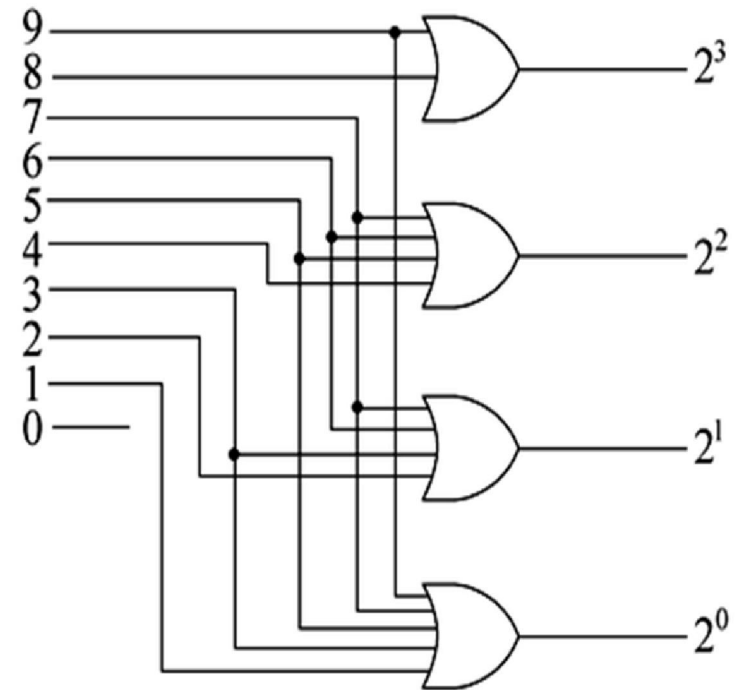
# 组合逻辑电路

---

- 组合逻辑电路基础
- 多路器和逆多路器
- 编码器和译码器

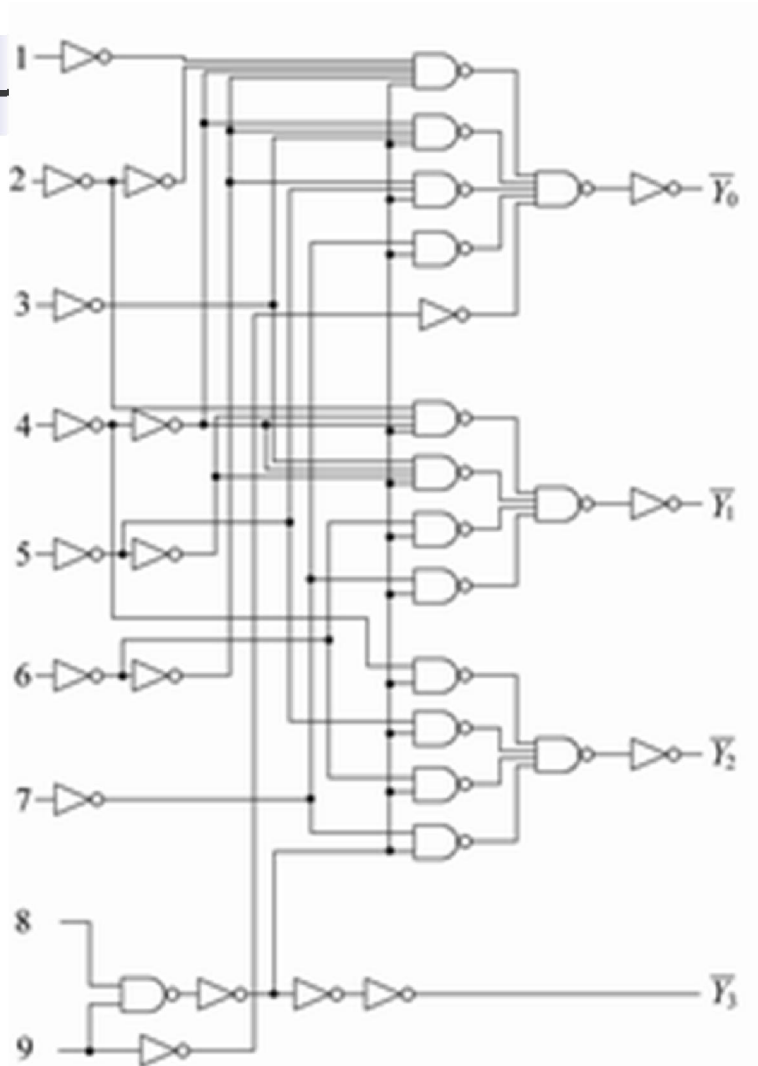
## 编码器 (encoder)

- 编码器把一组  $m$  个输入信号用一组  $n$  位 ( $2^n \geq m$ ) 二进制代码表示, 使它们之间一一对应。
- 产生BCD码的编码器



10-4编码器

# 10-4编码器的一种逻辑结构



注意：

集成电路中输入、  
输出都经过反相器  
作缓冲器



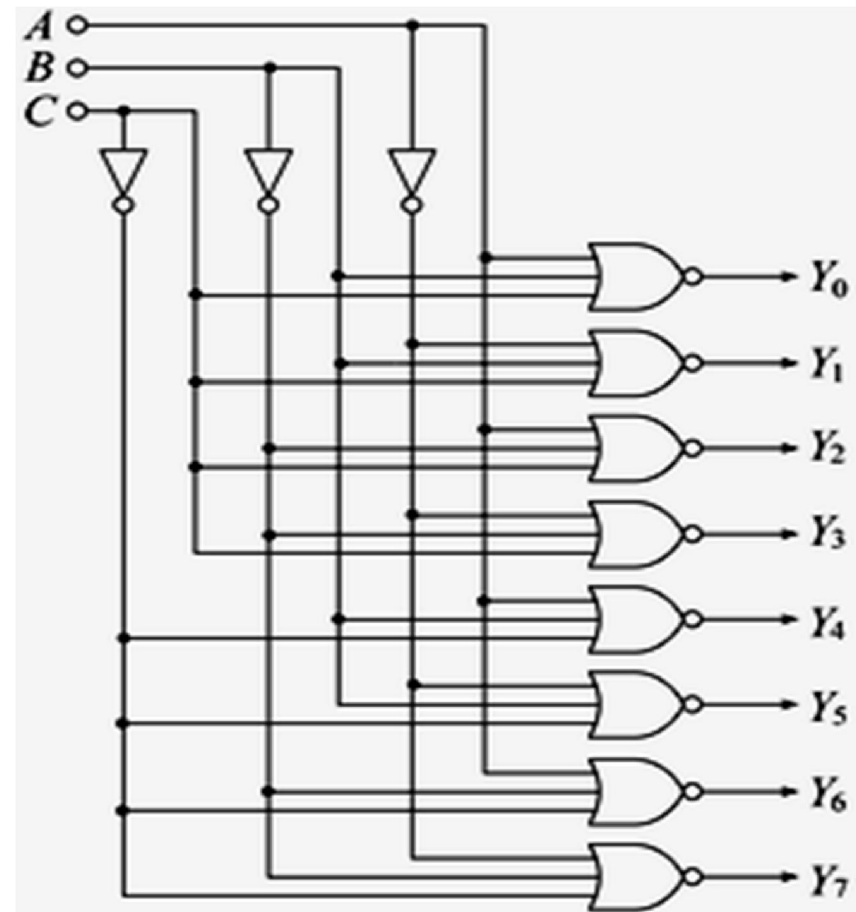
## 译码器 (decoder)

---

- 译码器：解读输入的二进制代码。根据输入代码的值在一组输出中相应的一个输出线上产生输出信号。
- 在数字系统中常用的译码器：
  - 二进制变量译码器
  - 码制变换译码器
  - 显示译码器

## 二进制译码器

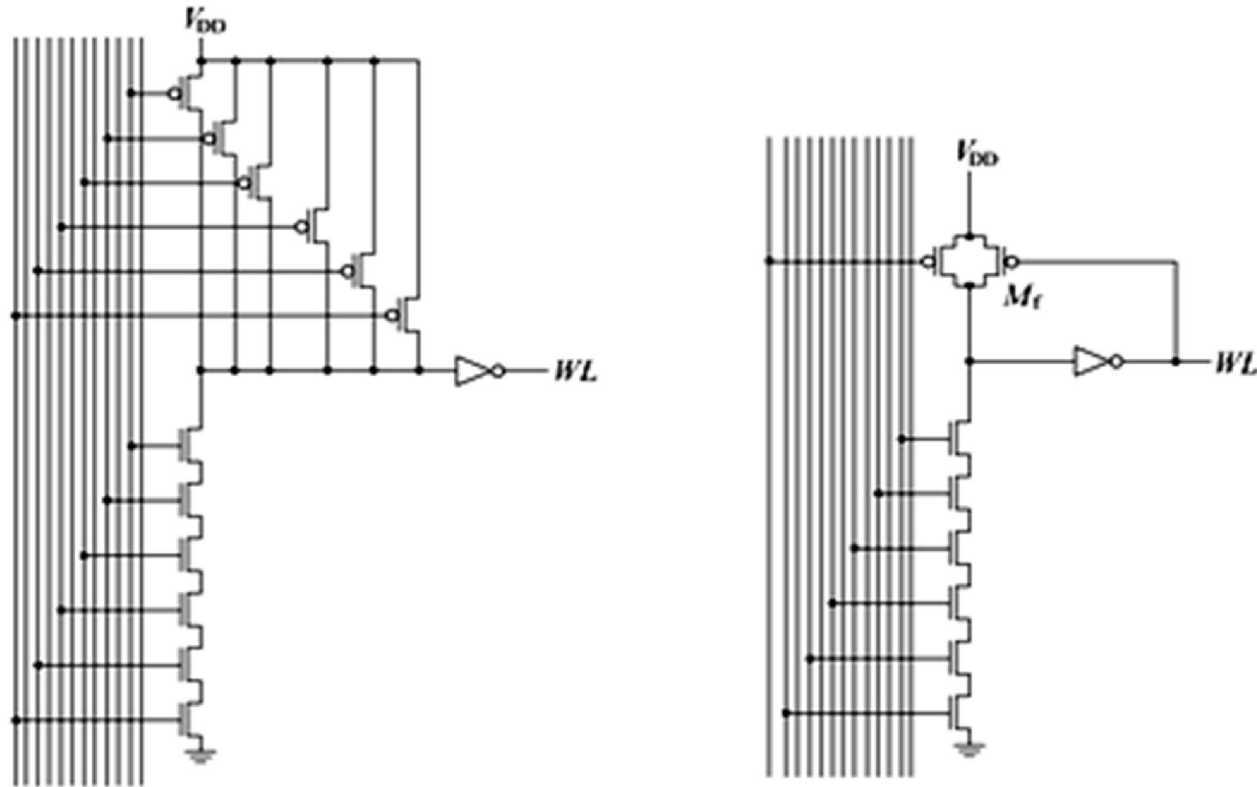
- 当输入一个 $n$ 位二进制变量时，在 $m$ 个输出线中**只有一个**是高电平或是低电平。
- 每个输出分别对应输入变量的一个**最小项**。
- 用一系列与非门或者一系列或非门实现。





## 存储器中的地址译码器

- 采用类NMOS电路或动态电路的形式。



增加一根时钟信号线 $\phi$ ,每个与非门少4个PMOS管

# 码制变换译码器

例：4位二进制代码翻译成格雷码

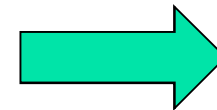
二进制码				格雷码			
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

## ■例：4位二进制代码翻译成格雷码

根据真值表得到如下逻辑表达式

$$\begin{aligned} G_0 &= \bar{B}_3\bar{B}_2\bar{B}_1B_0 + \bar{B}_3\bar{B}_2B_1\bar{B}_0 + \bar{B}_3B_2\bar{B}_1B_0 + \bar{B}_3B_2B_1\bar{B}_0 & G_1 &= \bar{B}_3\bar{B}_2\bar{B}_1\bar{B}_0 + \bar{B}_3\bar{B}_2\bar{B}_1B_0 + \bar{B}_3\bar{B}_2B_1\bar{B}_0 + \bar{B}_3\bar{B}_2B_1B_0 \\ &+ B_3\bar{B}_2\bar{B}_1B_0 + B_3\bar{B}_2B_1\bar{B}_0 + B_3B_2\bar{B}_1B_0 + B_3B_2B_1\bar{B}_0, & &+ B_3\bar{B}_2\bar{B}_1\bar{B}_0 + B_3\bar{B}_2\bar{B}_1B_0 + B_3\bar{B}_2B_1\bar{B}_0 + B_3\bar{B}_2B_1B_0, \\ G_2 &= \bar{B}_3B_2\bar{B}_1\bar{B}_0 + \bar{B}_3B_2\bar{B}_1B_0 + \bar{B}_3B_2B_1\bar{B}_0 + \bar{B}_3B_2B_1B_0 & G_3 &= B_3\bar{B}_2\bar{B}_1\bar{B}_0 + B_3\bar{B}_2\bar{B}_1B_0 + B_3\bar{B}_2B_1\bar{B}_0 + B_3\bar{B}_2B_1B_0 \\ &+ B_3\bar{B}_2\bar{B}_1\bar{B}_0 + B_3\bar{B}_2\bar{B}_1B_0 + B_3\bar{B}_2B_1\bar{B}_0 + B_3\bar{B}_2B_1B_0, & &+ B_3B_2\bar{B}_1\bar{B}_0 + B_3B_2\bar{B}_1B_0 + B_3B_2B_1\bar{B}_0 + B_3B_2B_1B_0 \end{aligned}$$

每个输出都是一系列最小项之和

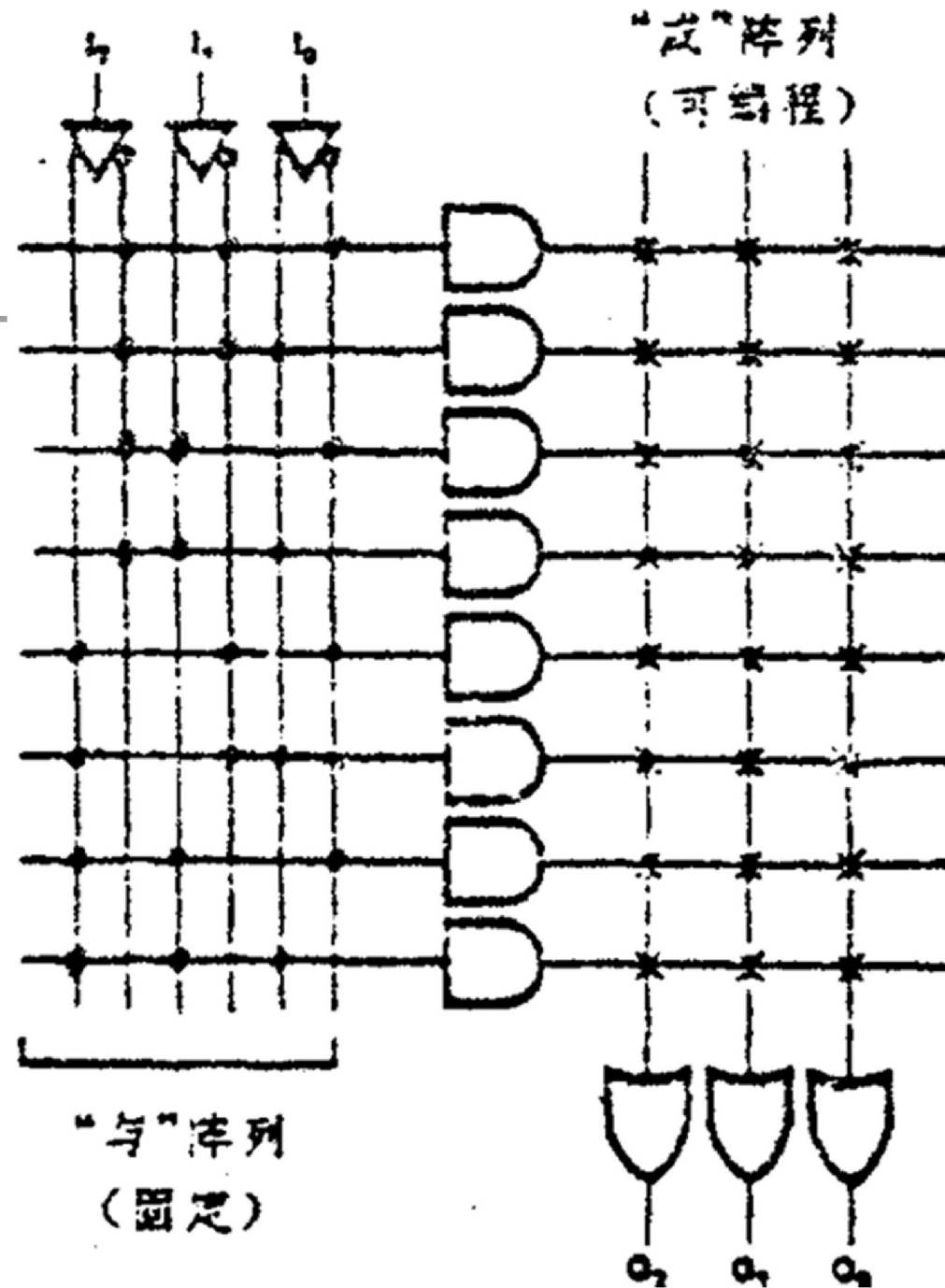


用4—16的二进制译码器产生全部最小项，再组合输出

ROM实现：产生全部最小项，即符合完全译码的规律



# ROM实现组合逻辑



# ROM实现组合

## 逻辑

$$G_0 = \overline{B_3} \overline{B_2} \overline{B_1} B_0 + \overline{B_3} \overline{B_2} B_1 \overline{B_0} + \overline{B_3} B_2 \overline{B_1} B_0 + \overline{B_3} B_2 B_1 \overline{B_0} \\ + B_3 \overline{B_2} \overline{B_1} B_0 + B_3 \overline{B_2} B_1 \overline{B_0} + B_3 B_2 \overline{B_1} B_0 + B_3 B_2 B_1 \overline{B_0}$$

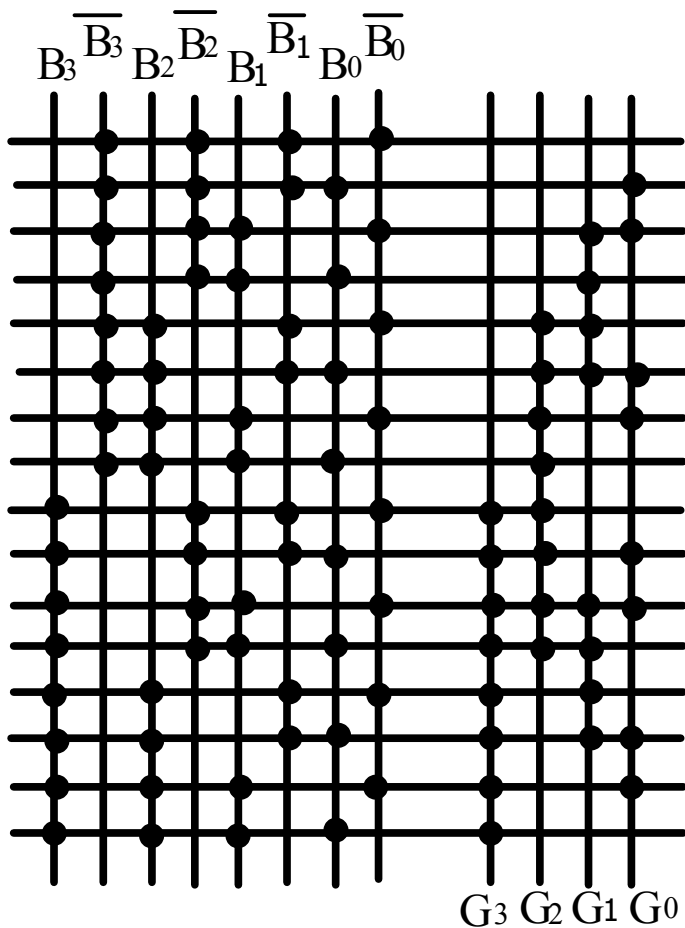
$$G_2 = \overline{B_3} B_2 \overline{B_1} \overline{B_0} + \overline{B_3} B_2 \overline{B_1} B_0 + \overline{B_3} B_2 B_1 \overline{B_0} + \overline{B_3} B_2 B_1 B_0 \\ + B_3 \overline{B_2} \overline{B_1} \overline{B_0} + B_3 \overline{B_2} \overline{B_1} B_0 + B_3 \overline{B_2} B_1 \overline{B_0} + B_3 \overline{B_2} B_1 B_0$$

$$G_1 = \overline{B_3} \overline{B_2} \overline{B_1} \overline{B_0} + \overline{B_3} \overline{B_2} \overline{B_1} B_0 + \overline{B_3} \overline{B_2} \overline{B_1} B_0 + \overline{B_3} \overline{B_2} B_1 \overline{B_0} \\ + B_3 \overline{B_2} \overline{B_1} \overline{B_0} + B_3 \overline{B_2} \overline{B_1} B_0 + B_3 \overline{B_2} B_1 \overline{B_0} + B_3 \overline{B_2} B_1 B_0$$

$$G_3 = B_3 \overline{B_2} \overline{B_1} \overline{B_0} + B_3 \overline{B_2} \overline{B_1} B_0 + B_3 \overline{B_2} B_1 \overline{B_0} + B_3 \overline{B_2} B_1 B_0 \\ + B_3 B_2 \overline{B_1} \overline{B_0} + B_3 B_2 \overline{B_1} B_0 + B_3 B_2 B_1 \overline{B_0} + B_3 B_2 B_1 B_0$$

$$G_3 = B_3, \quad G_2 = \overline{B_3} B_2 + B_3 \overline{B_2}$$

$$G_1 = \overline{B_2} B_1 + B_2 \overline{B_1}, \quad G_0 = \overline{B_1} B_0 + B_1 \overline{B_0}$$



ROM结构