数字逻辑

第三章 组合逻辑电路分析与设计

北京理工大学 计算机学院

张磊

leizhang@bit.edu.cn

本章内容

二. 组合逻辑功能模块

- 1. 组合功能模块
- 2. 基本逻辑函数
- 3. 译码和译码器
- 4. 基于译码器的组合电路
- 5. 编码和编码器
- 6. 选择和复用器
- 7. 基于复用器的组合电路

4. 编码和编码器

- 译码
  - 输入n位，输出m ($n \le m \le 2^n$) 位
  - 例子：输入二进制码，在输出中将对应位置1
    - 010→00000100
- 编码
  - 输入最大m ($n \le m \le 2^n$) 位，输出n位
  - 例子：输入中某位为1，输出中编码出位置
    - 00000100→ 010
- 译码和编码互逆

4. 编码和编码器

- 编码
  - 输入最大m ($n \le m \le 2^n$) 位，输出n位
  - 例子：输入中某位为1，输出中编码出相应位
    - 00000100→ 010
- 编码器
  - 实现编码功能的电路
  - m-n 编码器
  - 例子:
    - 输入：只有1位是1的输入，如 00000100
    - 输出：1的位置的编码，如010

- 例子：十进制-BCD编码器
  - 输入: 10位代表从0到9，(D_0, ..., D_9)
  - 输出: 4位BCD码
  - 函数: 若输入位 D_i 是1, 则输出(A_3, A_2, A_1, A_0)是i的BCD码
- 如何得到电路?
  - 真值表→卡诺图优化→优化
  - 但是10个输入?

- 思考：A_j 什么时候是1？
  - i的二进制中A_j位是1
  - 输入D_i 是布尔方程A_j 的一项

十进制符号	**BCD码**	十进制符号	**BCD码**
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

- 思考： A_j 什么时候是1?
  - i的二进制中A_j位是1
  - 输入D_i 是布尔方程A_j 的一项
- 布尔方程:
  - $A_3 = D_8 + D_9$
  - $A_2 = D_4 + D_5 + D_6 + D_7$
  - $A_1 = D_2 + D_3 + D_6 + D_7$
  - $A_0 = D_1 + D_3 + D_5 + D_7 + D_9$

$$A_0 = D_1 + D_3 + D_5 + D_7$$
$$A_1 = ?$$
$$A_2 = ?$$

□例子：八-二进制编码器

- **输入:** 八进制数字（D_0，…，D_7）
- **输出:** 二进制

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

- 如果输入中不止一位为1
  - 如01001000
  - 则编码器输出：多个位编码的或
  - 不能正常工作

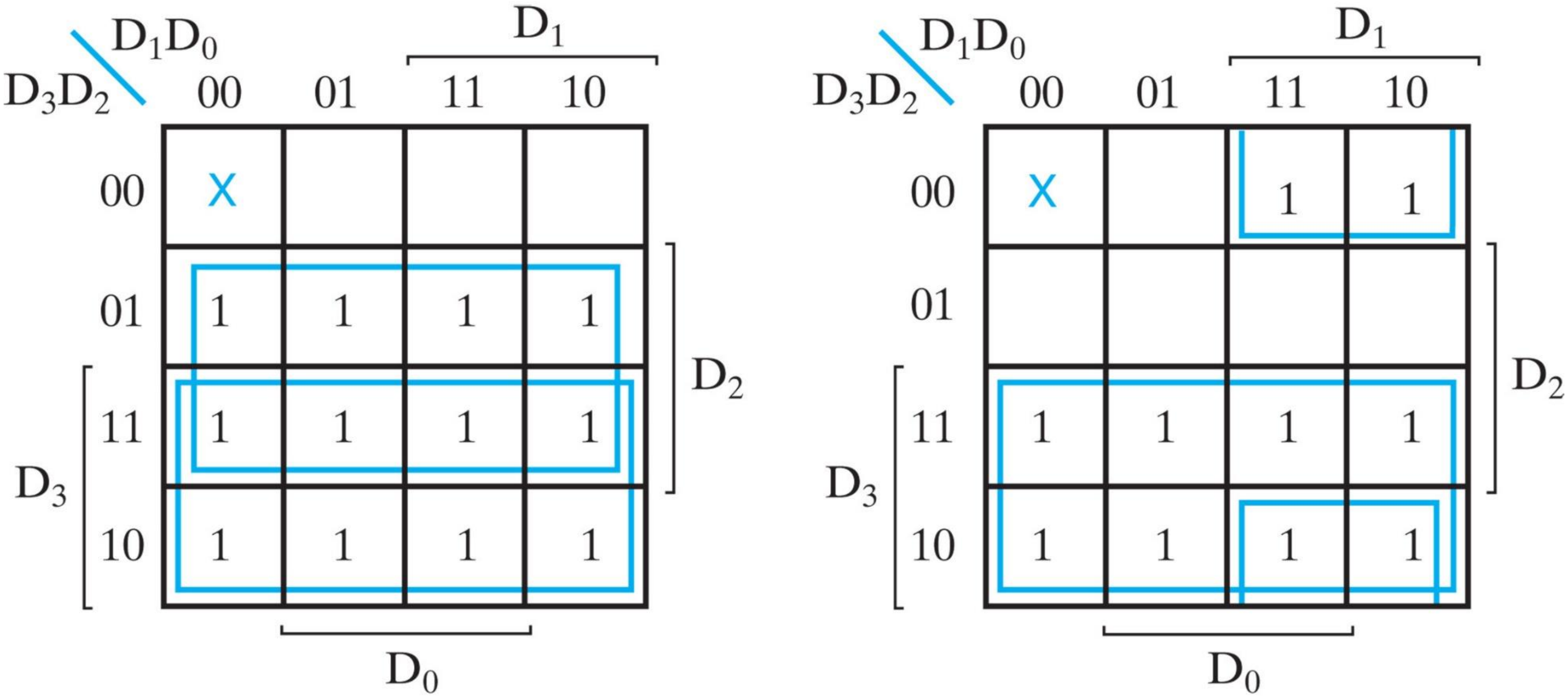
- 如果输入中不止一位为1
  - 如01001000
  - 则编码器输出：多个位编码的或
  - 不能正常工作
- 怎么办?
- 思想：选择最重要的输入位编码
  - 能接受所有的输入组合
  - 能产生有意义的输出
  - 优先编码器

- 例子：4输入优先编码器
  - 输入：(D_3, D_2, D_1, D_0)
  - 输出：A_1, A_0 和 V，V表示是否有1出现

Inputs				Outputs		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

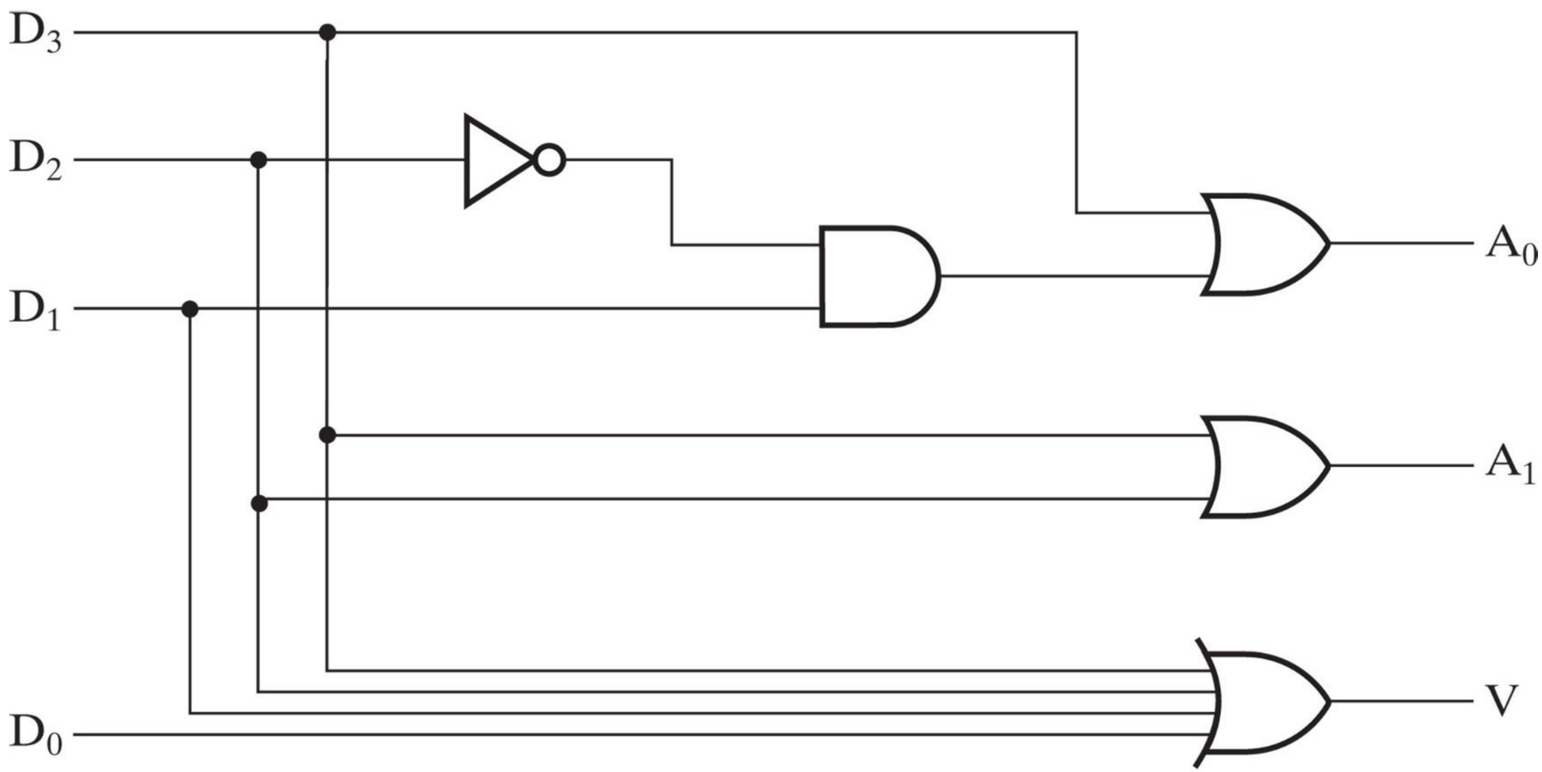
- X表示0或1，表条目对应乘积项而不是最小项

- 例子：4输入优先编码器
  - 优化及电路设计



$A_1 = D_2 + D_3$　　　　$A_0 = D_3 + D_1\overline{D}_2$

- 例子：4输入优先编码器
  - 优化及电路



- 例子：5输入优先编码器
  - 输入： (D_4, D_3, D_2, D_1, D_0)
  - 输出： A_2, A_1, A_0 和 V，V表示是否有1出现

No. of Min-terms/Row	Inputs					Outputs			
	D4	D3	D2	D1	D0	A2	A1	A0	V
1	0	0	0	0	0	X	X	X	0
1	0	0	0	0	1	0	0	0	1
2	0	0	0	1	X	0	0	1	1
4	0	0	1	X	X	0	1	0	1
8	0	1	X	X	X	0	1	1	1
16	1	X	X	X	X	1	0	0	1

- X表示0或1，表条目对应乘积项而不是最小项

- 如何得到电路?
  - 真值表→卡诺图优化→优化
  - 但是5个输入?
- 可以直接从表中读出方程，并进行优化 .
- 观察法

No. of Min-terms/Row	Inputs					Outputs			
	D4	D3	D2	D1	D0	A2	A1	A0	V
1	0	0	0	0	0	X	X	X	0
1	0	0	0	0	1	0	0	0	1
2	0	0	0	1	X	0	0	1	1
4	0	0	1	X	X	0	1	0	1
8	0	1	X	X	X	0	1	1	1
16	1	X	X	X	X	1	0	0	1

$$A_2 = D_4$$

$$A_1 = \overline{D}_4 D_3 + \overline{D}_4 \overline{D}_3 D_2 = \overline{D}_4 F_1, \ F_1 = (D_3 + D_2)$$

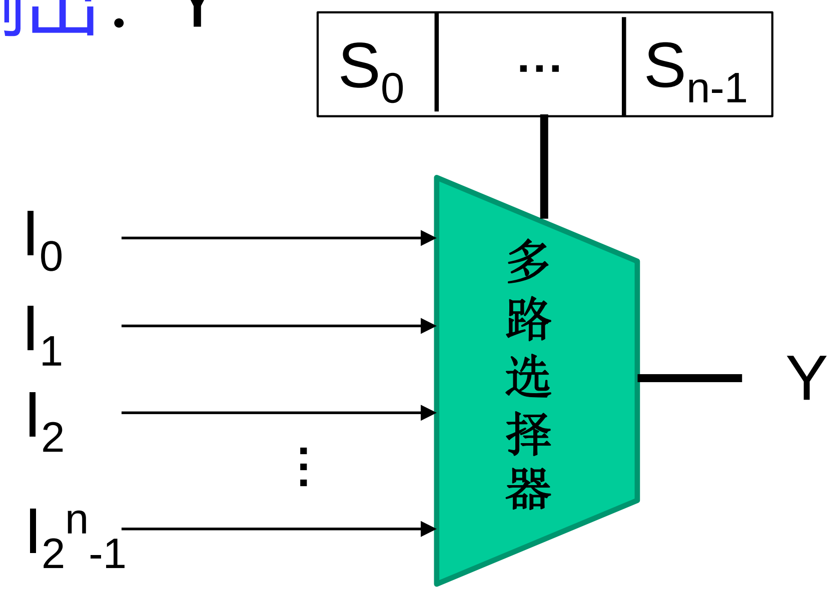
$$A_0 = \overline{D}_4 D_3 + \overline{D}_4 \overline{D}_3 \overline{D}_2 D_1 = \overline{D}_4 (D_3 + \overline{D}_2 D1)$$

$$V = D_4 + F_1 + D_1 + D_0$$

6. 多路复用器

- 选择
  - 计算机系统的关键功能模块
- 执行选择的电路：
  - 输入
    - 一组待选择的数据
    - 一组用来进行选择的选择信号
  - 一个输出
- 执行选择的逻辑电路被称为多路复用器

- 多路复用器： 从输入选择信息并输出
  - 输入
    - 待选择数据： 最多 2^n 个输入 (I_{2^n-1}, … I_0)
    - 选择信号： n个，(S_{n-1}, … S_0)
  - 1个输出：Y

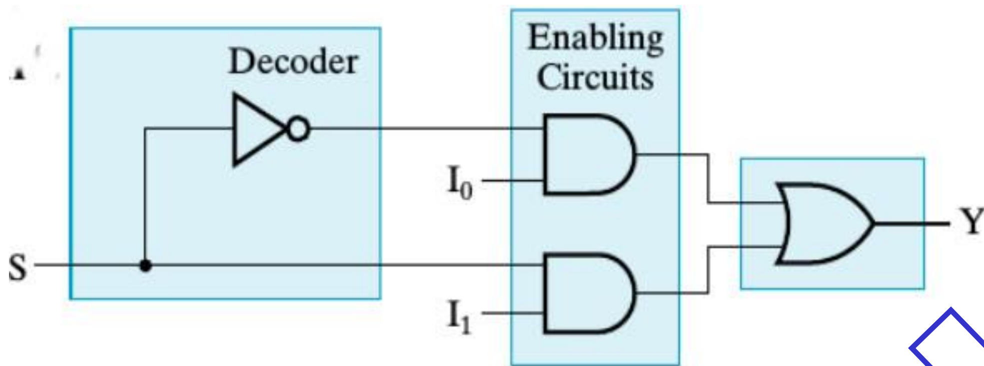


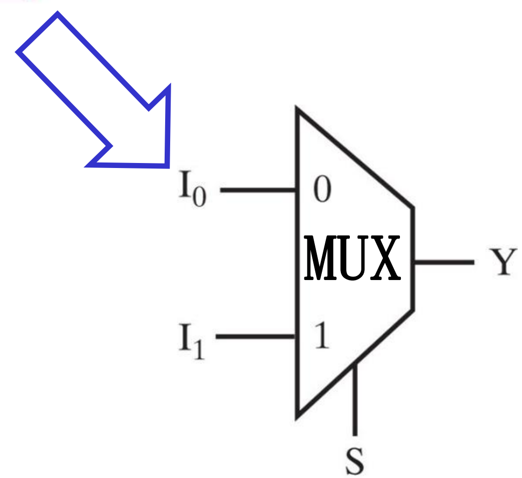
- 例子：2-1多路复用器
- 输入：
  - 待选择数据：2^1个， I_0, I_1
  - 选择信号：n = 1个 ， S_0
    - $S_0 = 0$ 选择输入 I_0
    - $S_0 = 1$ 选择 I_1
- 输出：Y
- 方程:

$$Y = \overline{S} I_0 + S I_1$$

- 电路:

□ 例子：2-1多路复用器





- □ 1-2 译码器
- □ 2个 使能（2输入与门）
- □ 2-输入或门

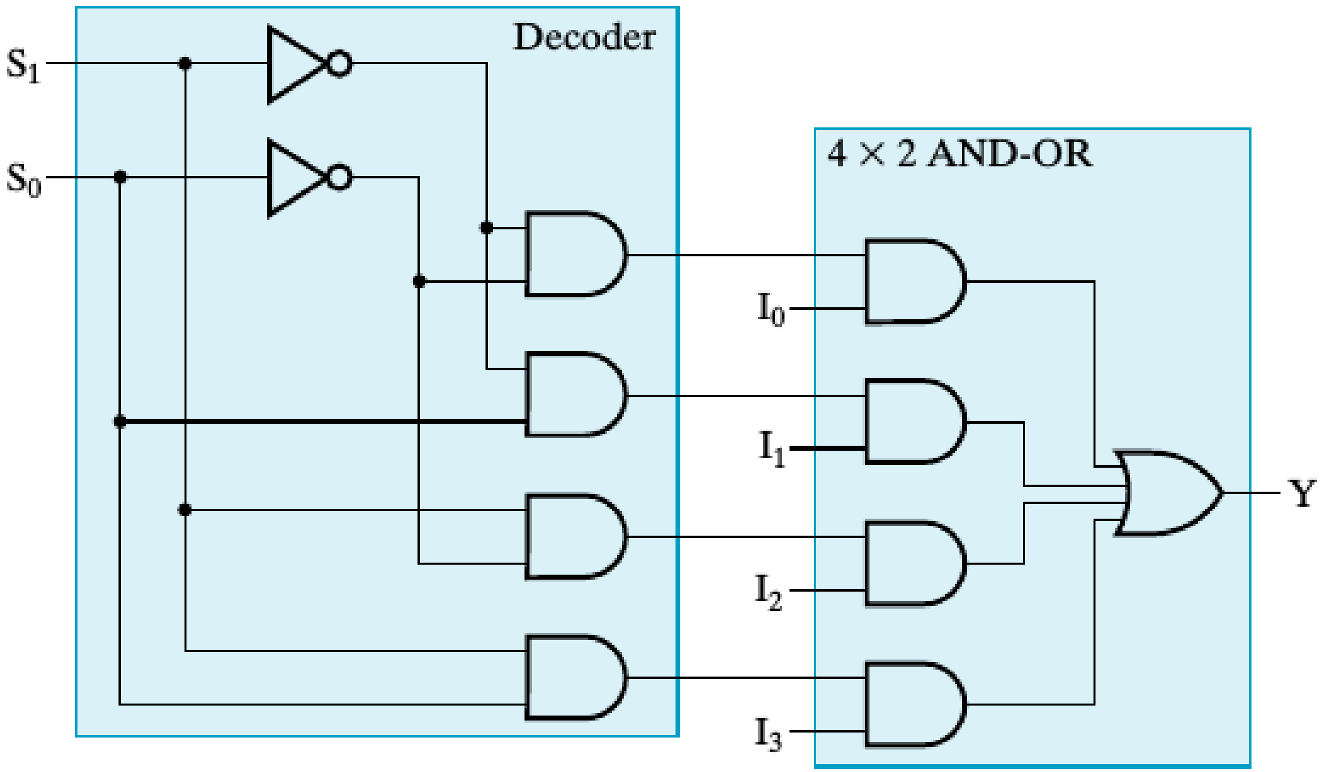
- 2^n-1 多路复用器
  - n-2^n 译码器
  - 2^n 使能（2输入与门）
  - 2^n 输入或门
- 后面两个看作$2^n \times 2$与或门
  - 2 表示与门输入数量
  - 2^n 表示与门的数量
  - 1个2^n 输入的或门

- 2^2-1 多路复用器
  - 2-2^2 译码器
  - $2^2 \times 2$ 与-或门

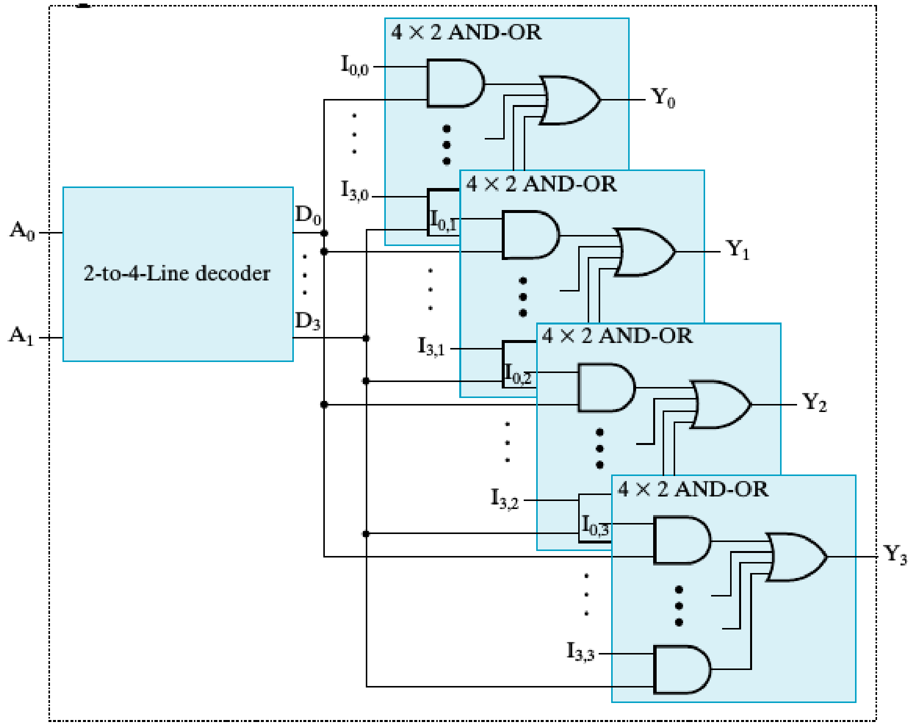
$$Y = \overline{S}_1\overline{S}_0 I_0 + \overline{S}_1\overline{S}_0 I_1 + S_1\overline{S}_0 I_2 + S_1 S_0 I_3$$

X	Y	Z	Product Term	Symbol	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\overline{X}\,\overline{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\overline{X}Y\overline{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\overline{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\overline{Y}\,\overline{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\overline{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\overline{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

- 2^2-1 多路复用器
  - 2-2^2 译码器
  - $2^2 \times 2$ 与-或门



- 位宽展开：4-1 四位多路复用器
  - 选择位向量而不是单个位
  - 平行的使用四个$2^n \times 2$ 与-或



7. 基于复用器的组合电路

- 实现m个函数，包含n个变量
- 方法1：m位宽 2^n-1 多路复用器
- 方法2：m位宽2^{n-1}-1多路复用器

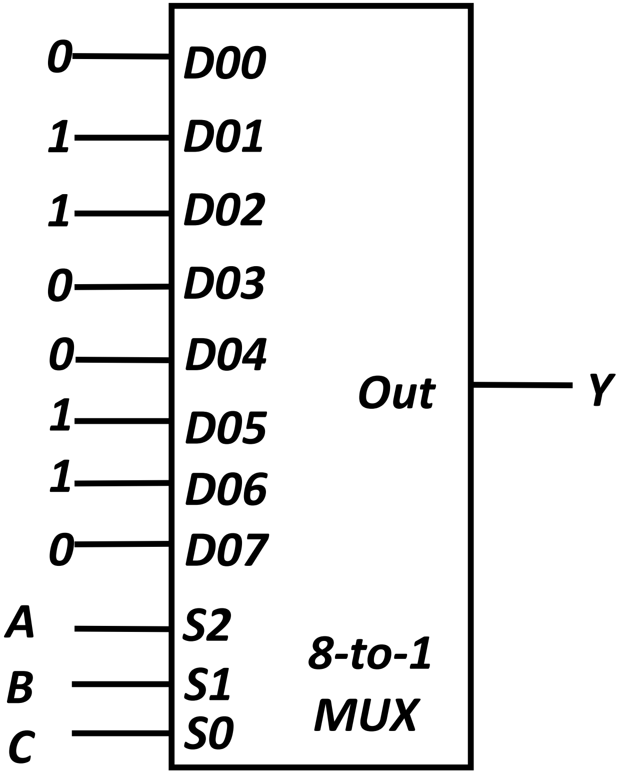
- 方法1： m位宽 2^n-1 多路复用器
  - 得到函数的真值表
  - 根据真值表
    - 将函数输入$S_{n-1}, \dots, S_0$作为选择信号
    - 真值表中的值作为多路复用器的待选择数据
    - 将多路复用器的输出标识成函数输出

- 例子：格雷码到二进制码 转换
- 真值表如图所示
  - x=C
  - y和z比较复杂

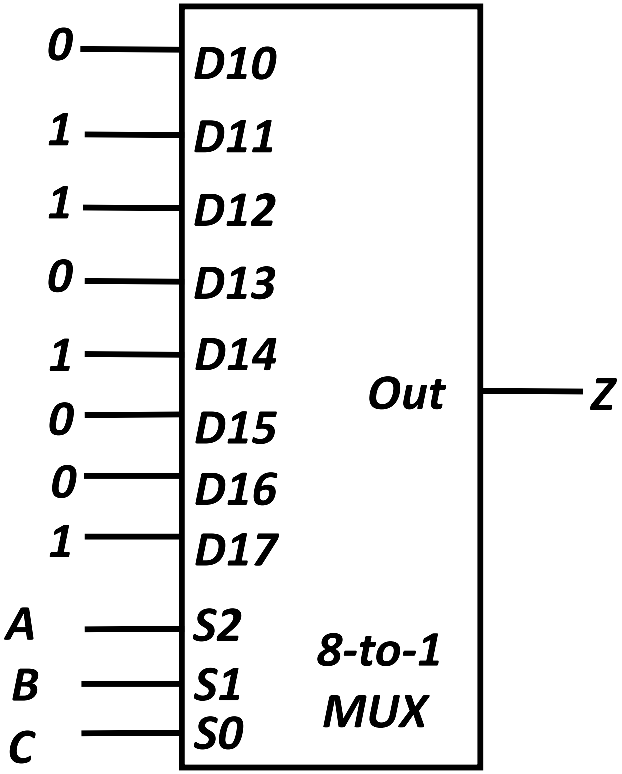
Gray A B C	Binary x y z
0 0 0	0 0 0
1 0 0	0 0 1
1 1 0	0 1 0
0 1 0	0 1 1
0 1 1	1 0 0
1 1 1	1 0 1
1 0 1	1 1 0
0 0 1	1 1 1

- 重新排列使得输入按计数顺序
  - y和z 可以通过一个双位8-1多路复用器实现
    - 将A, B, C连接到选择信号
    - 将y和z连接到两个输出
    - 将他们各自的真值连接到待选择数据

Gray A B C	Binary x y z
0 0 0	0 0 0
0 0 1	1 1 1
0 1 0	0 1 1
0 1 1	1 0 0
1 0 0	0 0 1
1 0 1	1 1 0
1 1 0	0 1 0
1 1 1	1 0 1



Gray A B C	Binary x y z
0 0 0	0 0 0
0 0 1	1 1 1
0 1 0	0 1 1
0 1 1	1 0 0
1 0 0	0 0 1
1 0 1	1 1 0
1 1 0	0 1 0
1 1 1	1 0 1



- 方法2：m位宽 2^{n-1}-1 多路复用器
  - 得到函数的真值表
    - 基于n-1个变量值，将真值表中的行配对
      - n-1个变量一致
    - 设剩下的变量为X
    - 每一配对中，将输出表达成(0, 1, X, $\overline{x}$)
  - 根据真值表
    - 将n-1个变量作为选择信号
    - (0, 1, X, $\overline{x}$)作为待选择数据
    - 将多路复用器的输出标识成函数输出

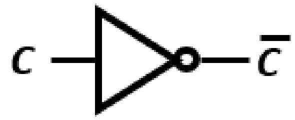
- 例子：格雷码到二进制码 转换
- 真值表如图所示
  - x=C
  - y和z比较复杂

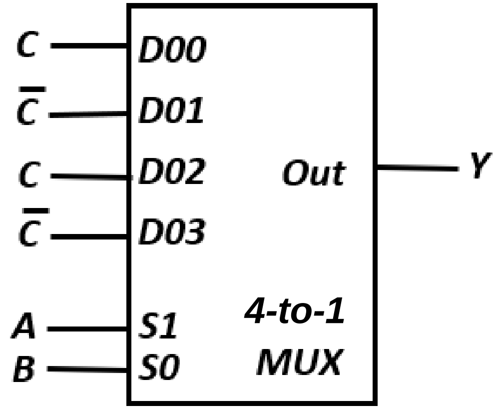
Gray A B C	Binary x y z
0 0 0	0 0 0
1 0 0	0 0 1
1 1 0	0 1 0
0 1 0	0 1 1
0 1 1	1 0 0
1 1 1	1 0 1
1 0 1	1 1 0
0 0 1	1 1 1

- 重排真值表，使得输入按照计数升序。

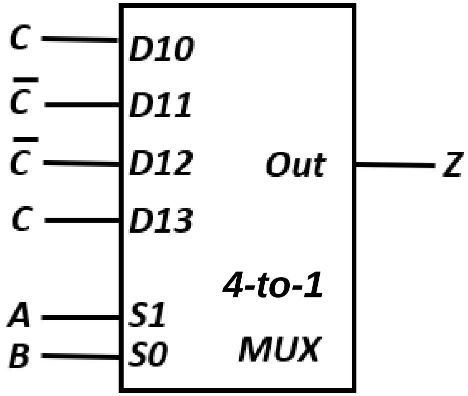
Gray A B C	Binary x y z	Rudimentary Functions of C for y	Rudimentary Functions of C for z
0 0 0	0 0 0	$F = C$	$F = C$
0 0 1	1 1 1		
0 1 0	0 1 1	$F = \overline{C}$	$F = \overline{C}$
0 1 1	1 0 0		
1 0 0	0 0 1	$F = C$	$F = \overline{C}$
1 0 1	1 1 0		
1 1 0	0 1 0	$F = \overline{C}$	$F = C$
1 1 1	1 0 1		

Gray A B C	Binary x y z	Rudimentary Functions of C for y	Rudimentary Functions of C for z
0 0 0	0 0 0	F = C	F = C
0 0 1	1 1 1		
0 1 0	0 1 1	F = $\overline{C}$	F = $\overline{C}$
0 1 1	1 0 0		
1 0 0	0 0 1	F = C	F = $\overline{C}$
1 0 1	1 1 0		
1 1 0	0 1 0	F = $\overline{C}$	F = C
1 1 1	1 0 1		





Gray A B C	Binary x y z	Rudimentary Functions of C for y	Rudimentary Functions of C for z
0 0 0	0 0 0	$F = C$	$F = C$
0 0 1	1 1 1		
0 1 0	0 1 1	$F = \overline{C}$	$F = \overline{C}$
0 1 1	1 0 0		
1 0 0	0 0 1	$F = C$	$F = \overline{C}$
1 0 1	1 1 0		
1 1 0	0 1 0	$F = \overline{C}$	$F = C$
1 1 1	1 0 1		



- 方法1VS方法2
  - 方法1简单，方法2复杂
  - 方法2 的门成本约是方法1的一半

香农展开式定理

$$f(x_1,x_2,\ldots,x_k)=x_k\cdot f(x_1,x_2,\ldots,x_{k-1},1)+\overline{x}_k\cdot f(x_1,x_2,\ldots,x_{k-1},0)$$

- 3-42 使用2个8-1多路复用器来构造一个15-1多路复用器。两个多路复用器应该相互连接，这样用于产生选择码0000至1110上的附加逻辑就最少。
- 提示:
- 两个多路复用器：1号，2号
  - 1号多路复用器可以选择8个数据：0000-0111
  - 2号多路复用器
    - 选择7个数据：1000-1110
    - 1号多路复用器的输出→1111
  - 将最高位作为复用器选择信号