
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

1. 组合功能模块

□ 组合功能模块

- 在电路设计中经常使用的公共模块
- 每个功能模块对应一个组合电路实现

□ 按集成度高低的不同

- 小规模集成电路 (SSI): 10-100 个晶体管
- 中规模集成电路 (MSI): 100-1000
- 大规模集成电路 (LSI): 1000-100000
- 超大规模集成电路 (VLSI): 100000以上

□ 芯片(集成电路)工艺 (nm)

- 晶体管栅极的宽度, 也称栅长
- 栅长越短
 - 同尺寸的硅片可集成更多晶体管
 - 频率越高, 功耗更低

□ 目前水平: 5nm, 迈向3nm

□芯片产业链

- 制造设备: 光刻机、蚀刻机和薄膜沉积
- 设计
- 生产
- 封测

□ 制造设备

- 蚀刻机：中微半导体 7nm
- 光薄膜沉积：北方华创 28nm
- 光刻机：
 - 上海微电子 90nm, 荷兰ASML(阿斯麦) 5nm
 - 差距最大



□ 芯片设计

- 华为海思, 5nm, ARM架构需要授权

□ 芯片生产

- 台积电: 3nm
- 中芯国际: 14nm, 有望2020年四季度量产7nm

□ 芯片封测

- 技术含量相对较低
- 长电科技、华天科技、通富微电
- 世界第一梯队

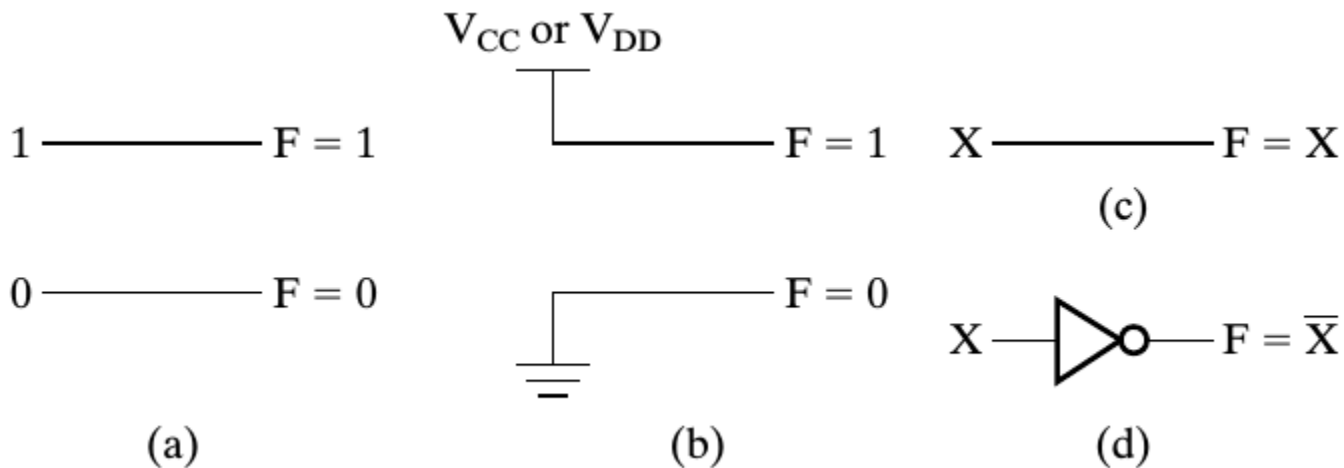
2. 基本逻辑函数

- 单变量函数
- 多位函数
- 使能函数

□ 单变量函数

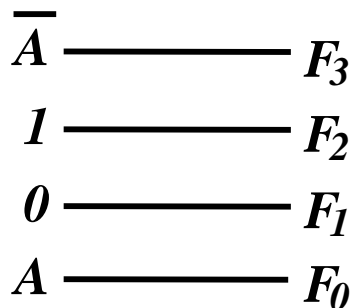
- 一个变量X的函数
- 可以在输入处用作功能块

X	F = 0	F = X	F = \bar{X}	F = 1
0	0	0	1	1
1	0	1	0	1

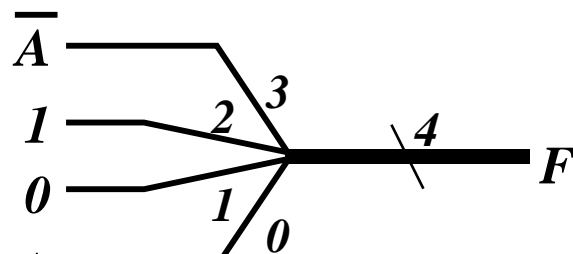


□ 多位函数

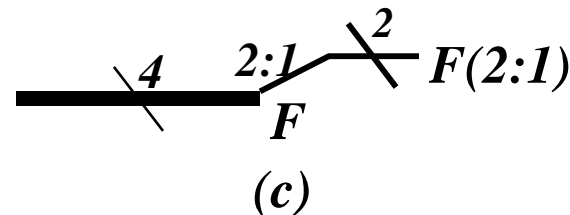
- 1位函数的向量
- 粗线代表总线，其是一个向量信号, 如图(b)
- 可以从总线中分割出一个位子集, 如 (c, d)



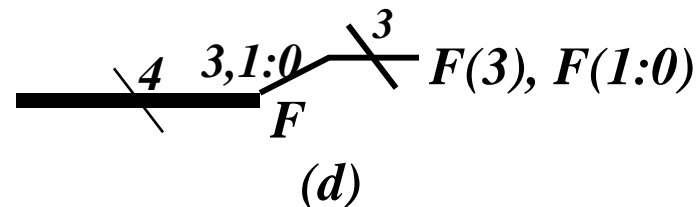
(a)



(b)



(c)

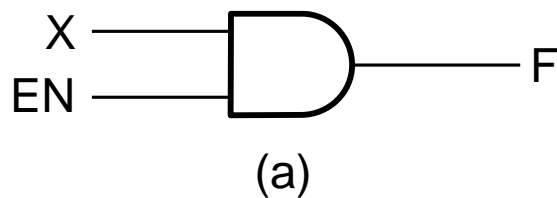


(d)

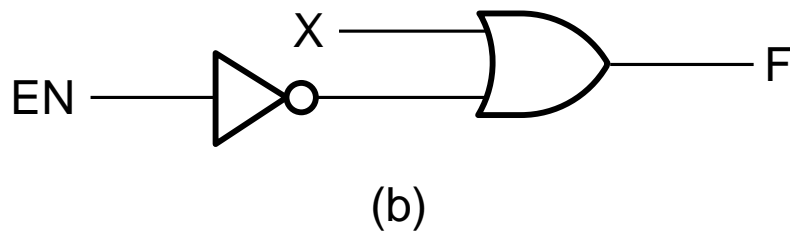
□ 使能函数

- 是否允许信号从输入传到输出
- 引入使能信号 EN
 - $EN=1$ 允许信号传输
 - $EN=0$ 阻止信号传输
 - 输出用固定值替代，可能是0或者1

□ 固定值是 0



□ 固定值是 1



3. 译码和译码器

□ 译码

- 输入 n 位, 输出 m ($n \leq m \leq 2^n$) 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 ($n \leq m \leq 2^n$) 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

□译码

- 输入 n 位, 输出 m ($n \leq m \leq 2^n$) 位
- 例子: 输入二进制码, 在输出中将对应位置1
 - 010 → 00000100

□译码器

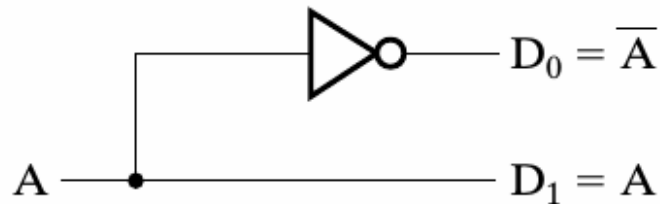
- 实现译码功能的电路
- n - m 译码器
- 例子:
 - 输入: 1的位置的编码, 如010
 - 输出: 只有1位是1的输出, 如 00000100

□ 如何设计一个1-2 译码器？

A	D₀	D₁
0	1	0
1	0	1

$$D_0 = \overline{A}$$

$$D_1 = A$$



□ 如何设计一个2-4译码器？

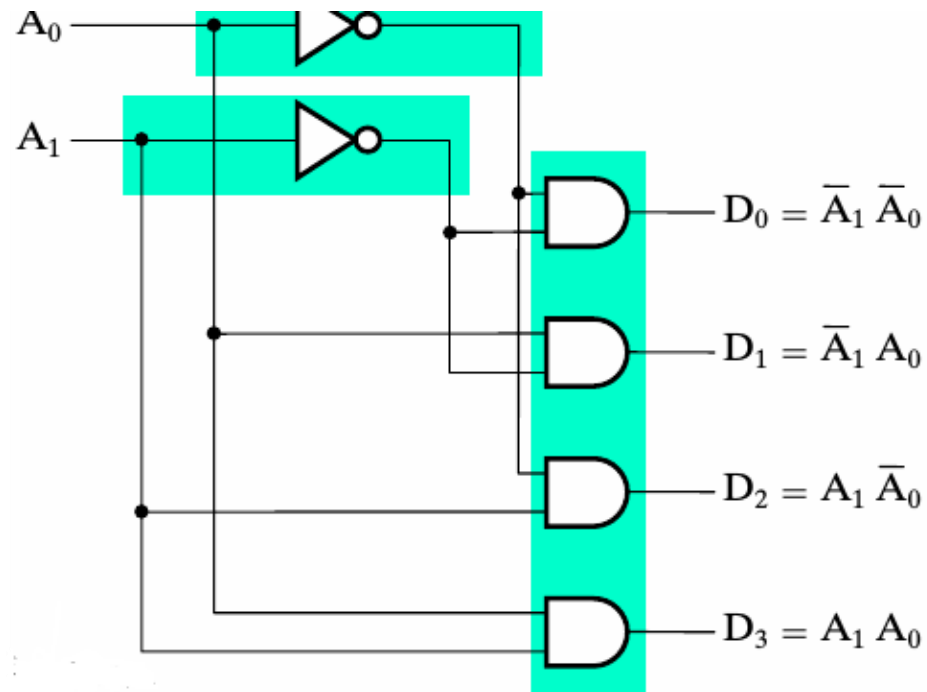
A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$D_0 = \bar{A}_1 \bar{A}_0$$

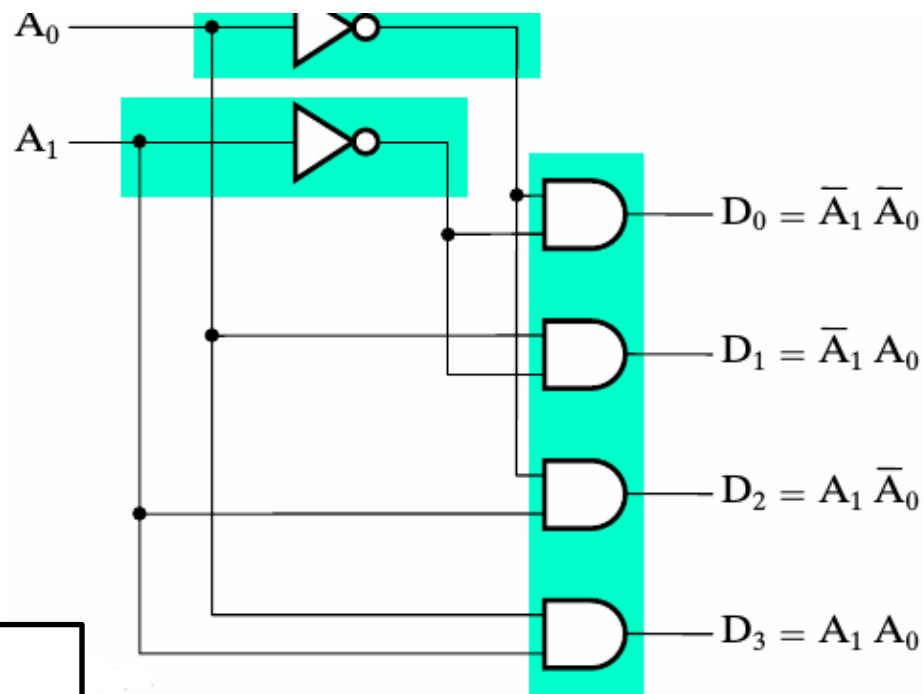
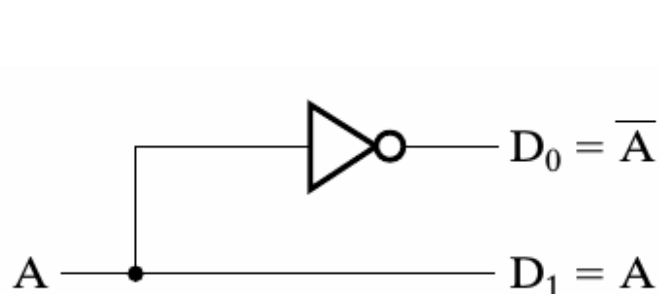
$$D_1 = \bar{A}_1 A_0$$

$$D_2 = A_1 \bar{A}_0$$

$$D_3 = A_1 A_0$$



□ 1-2译码器和2-4译码器有什么联系？



2-4译码器构成：

- 2个1-2译码器
- 4个与门

□ $n-2^n$ 译码器展开:

- 需要 2^n 个与门
- 每个输出与门被两个译码器驱动
 - 这两个译码器输入相等或相差1
- 将这两个译码器按照同样过程展开
- 直至到1-2译码器

□ 上述过程可经修改应用到输出 $\neq 2^n$ 的译码器

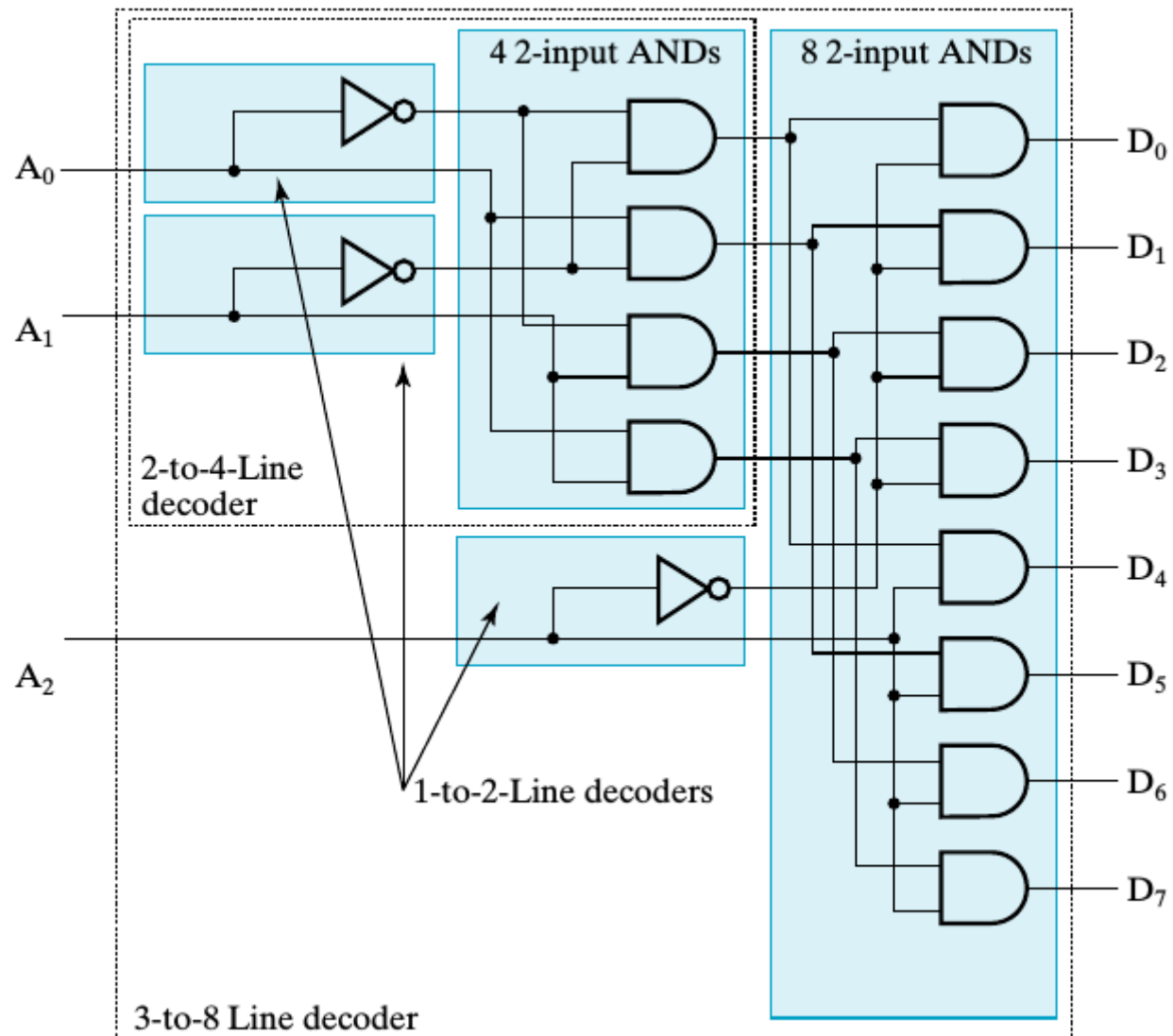
分级思想+与门组合

□ 例子: 3-8 译码器

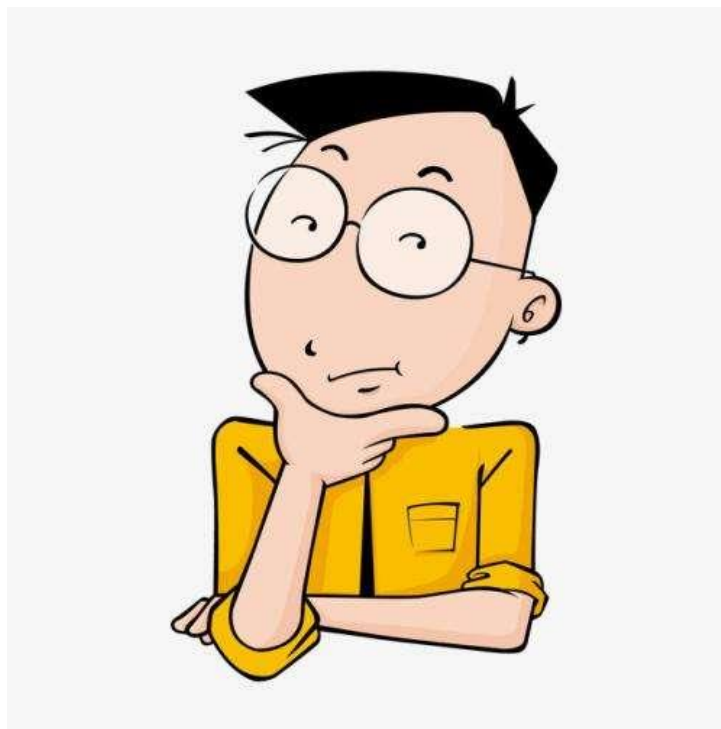
- 需要8个输出与门
- 每个输出与门被两个译码器驱动
 - 最相近的两个译码器
 - 2-4 译码器
 - 1-2 译码器

□ 2-4译码器

- 需要4个输出与门
- 每个输出与门被两个译码器驱动
 - 最相近的两个译码器
 - 2个 1-2 译码器



□ 如何构建一个7-128 译码器?



□ 7-128 译码器

- 需要128个输出与门
- 每个输出与门被两个译码器驱动
 - 最相近的两个译码器
 - 4-16 译码器
 - 3-8 译码器

□ 4-16译码器

- 需要16个输出与门
- 每个输出与门被两个译码器驱动
 - 最相近的两个译码器
 - 2个 2-4 译码器

□ 带有使能的译码器

- 电路输出增加使能信号-EN

□ 真值表

- 注意X可以表示0和1

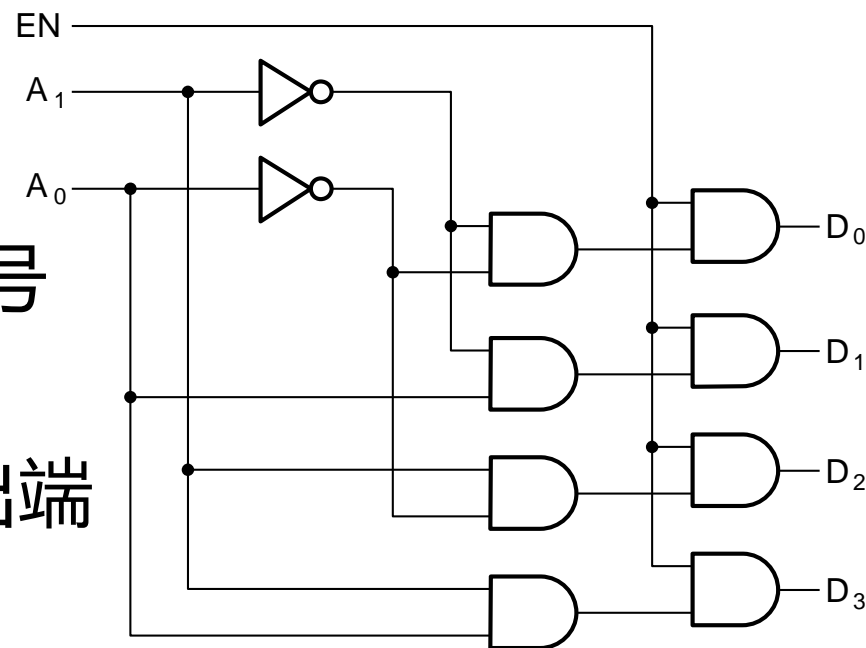
EN	A ₁	A ₀	D ₀	D ₁	D ₂	D ₃
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

(a)

□ 也被称为1-4 多路分配器

- EN为输入数据
- A₁A₀ 为输出端选择信号
 - 可以看作使能信号
- 将数据输出到选择输出端

□ 同一电路，两个视角



(b)

4. 基于译码器的组合电路

□ 实现1个函数，其中有 n 个变量

➤ 最小项之和的表达式，即标准型

➤ 一个 $n-2^n$ 译码器，译码器输出对应最小项

➤ 1个或门，将最小项或起来

□ 方法1:

➤ 得到函数的真值表

➤ 如果1在真值表中，就连接译码器输出和或门

□ 方法2:

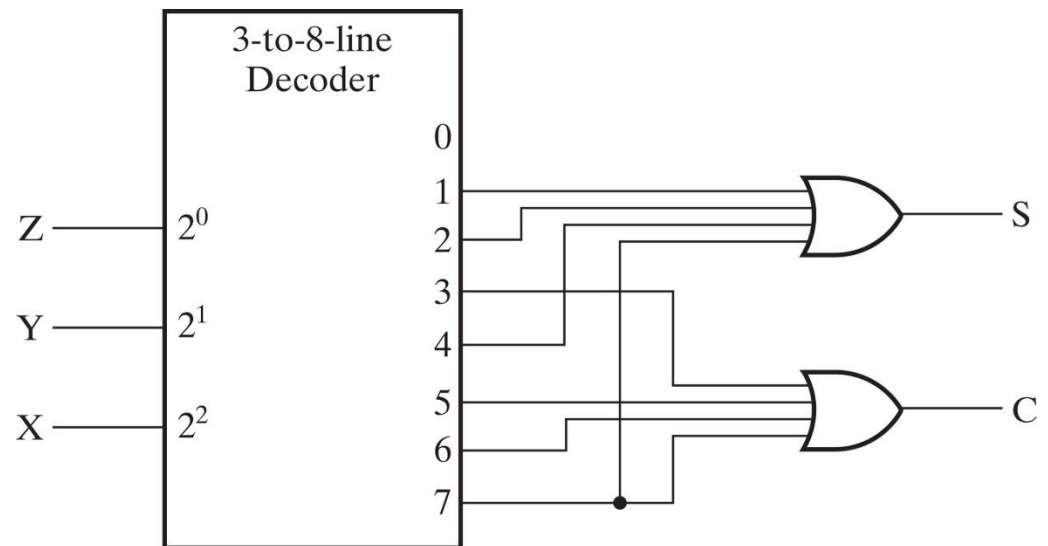
➤ 得到输出函数的最小项

➤ 将最小项用或门连接起来

4. 基于译码器的组合电路

■ 1位二进制加法器

X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$$S(X, Y, Z) = \sum m(1, 2, 4, 7)$$

$$C(X, Y, Z) = \sum m(3, 5, 6, 7)$$