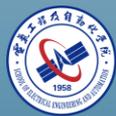




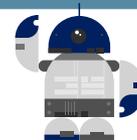
江苏师范大学
JIANGSU NORMAL UNIVERSITY



电气工程及自动化学院
SCHOOL OF ELECTRICAL ENGINEERING AND AUTOMATION

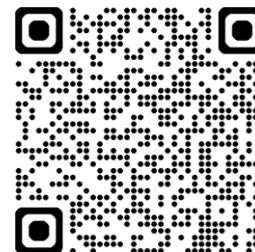


数电
技术



第3章 组合逻辑电路

李灿 | 12#503A | lic@jsnu.edu.cn | <https://sslic.cn>





第3章 组合逻辑电路

- 组合逻辑电路的一般分析方法
- 组合逻辑电路的设计方法
- 组合逻辑电路中的竞争-冒险
- 常用的集成组合逻辑电路



第3章 组合逻辑电路

<p>重点</p>	<ul style="list-style-type: none">✓ 组合逻辑电路的分析方法✓ 组合逻辑电路的设计方法✓ 常用集成组合逻辑电路
<p>难点</p>	<ul style="list-style-type: none">● 竞争与冒险● 逻辑电路的设计



§3.1 组合逻辑电路的一般分析方法



3.1 组合逻辑电路的一般分析方法

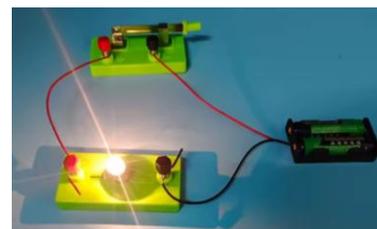
数字 逻辑 电路	组合 逻辑 电路	<p><u>功能</u>：输出只取决于当前时刻的输入</p> <p><u>结构</u>：门电路组成，不包含记忆单元及回路</p>
	时序 逻辑 电路	<p><u>功能</u>：输出取决于当前时刻的输入和原来的状态</p> <p><u>结构</u>：组合电路和记忆单元组成</p>

3.1 组合逻辑电路的一般分析方法



■ 组合逻辑电路的结构特点

- ① 只由门电路组成
- ② 电路的输入与输出无反馈路径
- ③ 电路中不包含记忆单元
- ④ 电路的输出与原来的状态无关





3.1 组合逻辑电路的一般分析方法

■ 组合逻辑电路的分析

◇ 任务

- 逻辑电路图 → 电路逻辑功能

◇ 目的

- 了解电路功能
- 改进电路设计

◇ 分析步骤

- ① 根据逻辑电路图，写出输出逻辑函数表达式
- ② 化简逻辑表达式
- ③ 根据化简后的逻辑表达式，列真值表
- ④ 由真值表或表达式分析电路功能



3.1 组合逻辑电路的一般分析方法

例3-1

例 分析如图所示电路的逻辑功能。

解 (1) 写出逻辑表达式

$$L = \overline{\overline{AB} \cdot A \cdot \overline{AB} \cdot B}$$

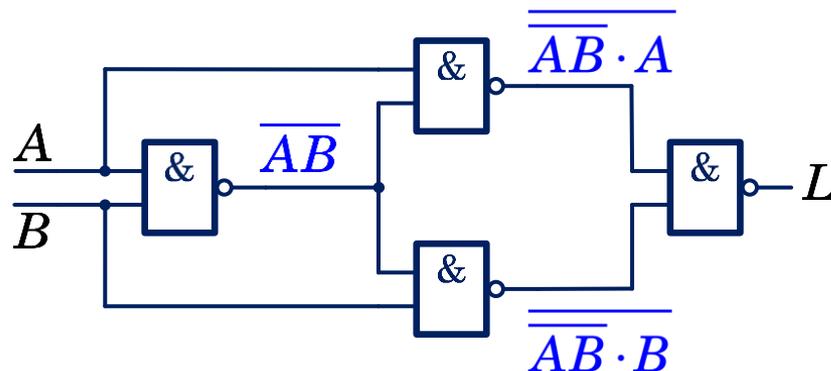
(2) 化简

$$\begin{aligned} L &= \overline{AB} \cdot A + \overline{AB} \cdot B \\ &= (\overline{A} + \overline{B})A + (\overline{A} + \overline{B})B \\ &= A\overline{B} + \overline{A}B \end{aligned}$$

(3) 列真值表 **»»**

(4) 确定电路的逻辑功能 **异或关系**

当两个输入信号相同时，输出**0**
当输入信号不同时，输出**1**



A	B	L
0	0	0
0	1	1
1	0	1
1	1	0

3.1 组合逻辑电路的一般分析方法

例3-2

例 分析如图所示电路的逻辑功能。

解 (1) 写出逻辑表达式

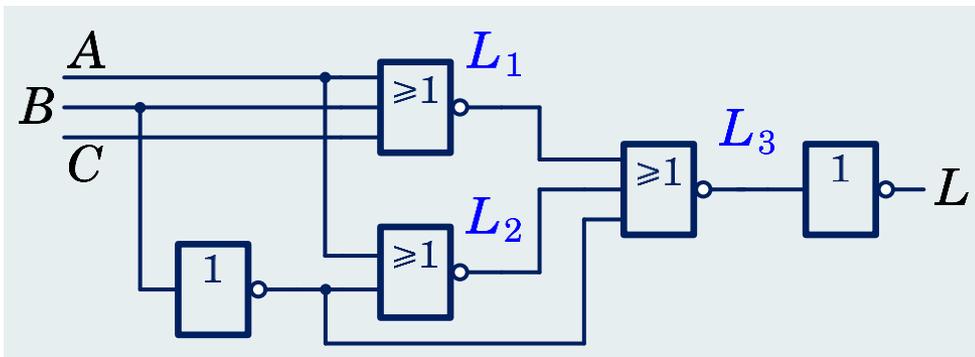
(2) 化简

$$\begin{aligned}
 L &= L_1 + L_2 + \overline{B} \\
 &= \overline{A} \overline{B} \overline{C} + \overline{A} B + \overline{B} \\
 &= \overline{A} + \overline{B} \\
 &= \overline{AB}
 \end{aligned}$$

(3) 列真值表 **»»»**

(4) 确定电路的逻辑功能 **与非关系**

电路的输出 **L** 只与输入 **A、B** 有关，与输入 **C** 无关



$$\begin{aligned}
 L_1 &= \overline{A} + B + C \\
 L_2 &= \overline{A} + \overline{B} \\
 L_3 &= \overline{L_1} + L_2 + \overline{B} \\
 L &= \overline{L_3}
 \end{aligned}$$

A	B	C	L
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



§3.2 组合逻辑电路设计方法



3.2 组合逻辑电路设计方法

■ 组合逻辑电路的设计

◇ 任务

- 逻辑功能要求 → 设计电路

◇ 分析步骤

- ① 根据所给逻辑要求，确定输入、输出变量（逻辑抽象）
- ② 根据所给逻辑功能要求，列真值表
- ③ 根据真值表，写逻辑表达式
- ④ 化简和变换逻辑表达式
- ⑤ 根据逻辑表达式，画逻辑电路图



3.2 组合逻辑电路设计方法

例3-3

例 设计三人表决电路。每人一个按键，如果同意则按下，不同意则不按。结果用指示灯表示，多数同意时指示灯亮，否则不亮。

解 (1) 逻辑抽象

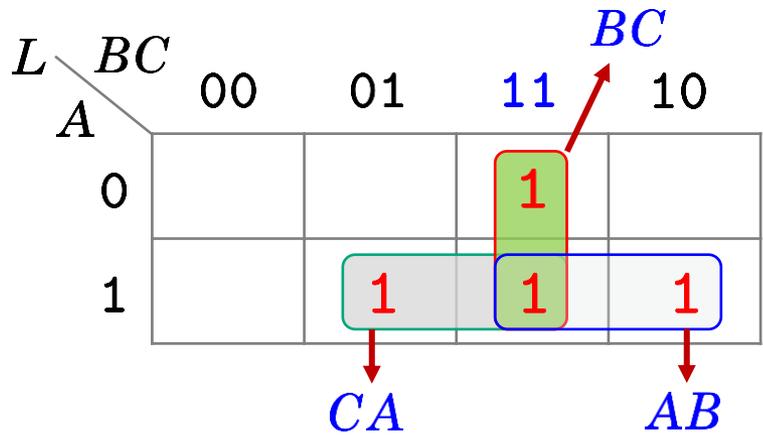
三个按键A、B、C按下时为“1”，不按时为“0”
输出量为L，多数赞成时是“1”亮，否则是“0”灭

(2) 列真值表

(3) 写逻辑表达式 $L = \sum m(3,5,6,7)$

(4) 化简 $\ggg L = AB + BC + CA$

A	B	C	L
0	0	0	0
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



3.2 组合逻辑电路设计方法

例3-3

例 设计三人表决电路。每人一个按键，如果同意则按下，不同意则不按。结果用指示灯表示，多数同意时指示灯亮，否则不亮。

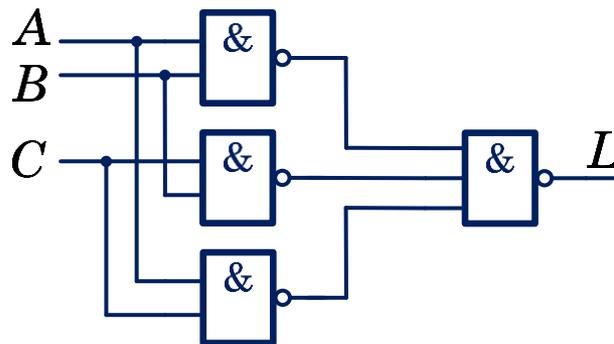
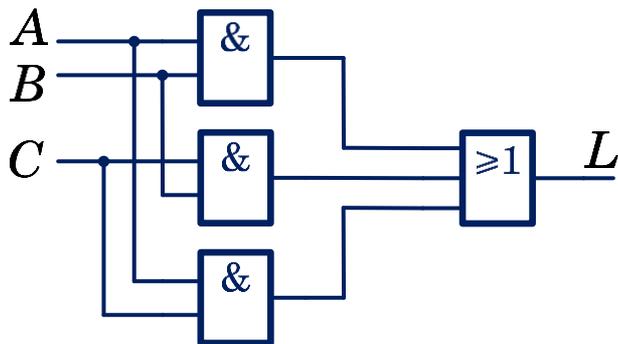
解 (5) 根据逻辑表达式，画逻辑电路图

$$L = AB + BC + CA$$

$$\ggg L = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{CA}}$$

✧ 用“与门”和“或门”实现

✧ 用“与非门”实现





§3.3 组合逻辑电路中的“竞争—冒险”



3.3 组合逻辑电路中的“竞争—冒险”

一、什么是竞争—冒险？

◇ 竞争

- **原因**：信号经不同路径到达某一点时，所用的时间不同，或者各个门电路自身的延时不同，导致各信号到达输出级的时间不同
- **定义**：在组合电路中，门电路的两个输入信号同时向相反的逻辑电平跳变时，如果有时间差，则这个时间差称为竞争
- **危害**：时延对数字系统是有害的，它会降低系统的工作速度，还会产生竞争冒险现象

◇ 冒险

- **定义**：由于竞争使得电路产生了暂时错误输出
- **分类**：
 - ① 逻辑冒险（一个变量的变化）
 - ② 功能冒险（多个变量的变化）

3.3 组合逻辑电路中的“竞争—冒险”

一. 什么是竞争—冒险？

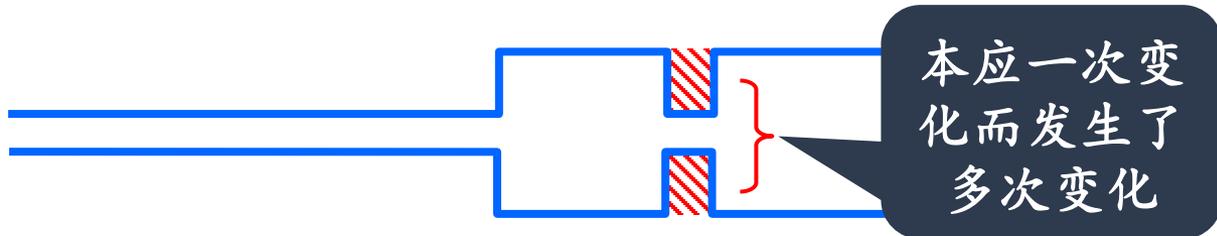
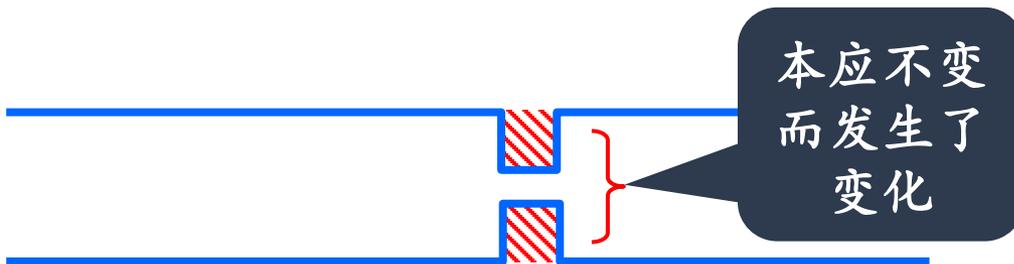
- ◇ 逻辑冒险：
 - ① 静态险象（本应不变而发生了变化）
 - ② 动态险象（本应一次变化而发生了多次变化）

静态0型

静态1型

动态0型

动态1型

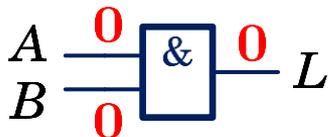


- ◇ 输出错误：
 - ① 0型险象（产生低电平错误）
 - ② 1型险象（产生高电平错误）

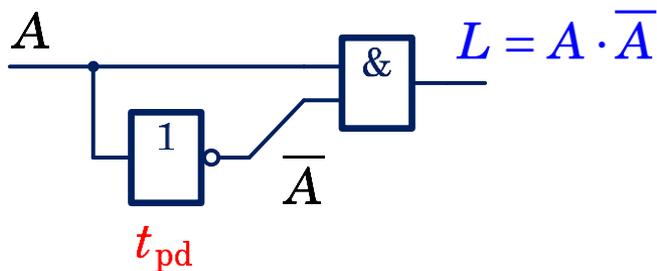
3.3 组合逻辑电路中的“竞争—冒险”

二. 竞争—冒险现象产生的原因？

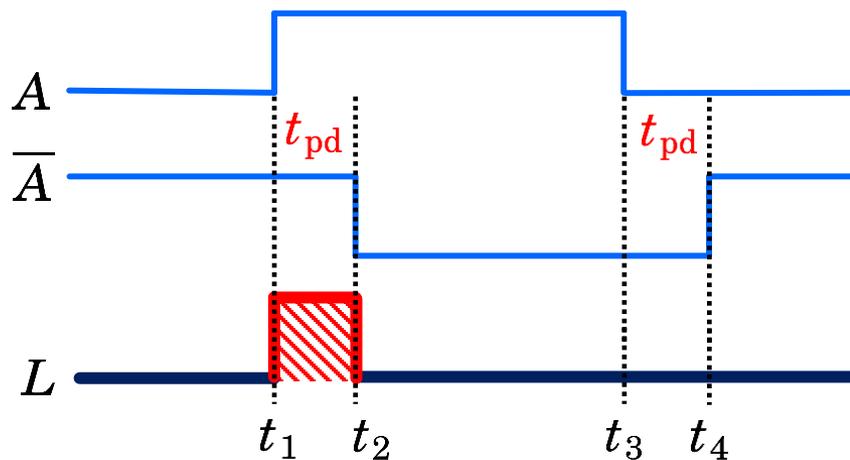
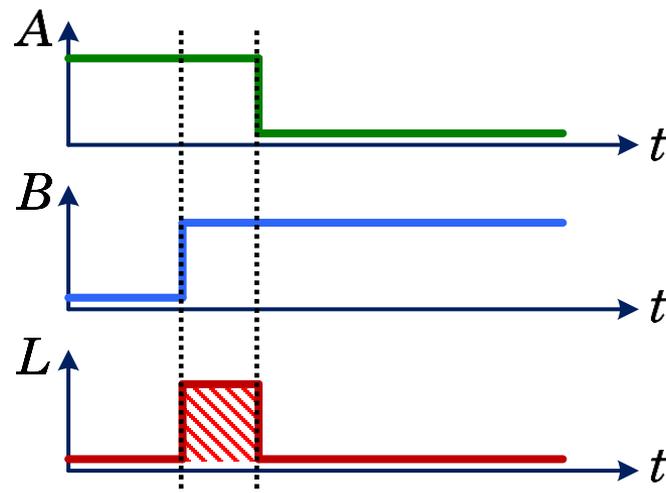
✧ 信号路径不同



✧ 器件延时不同



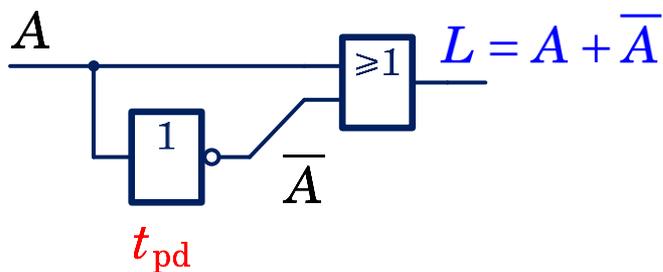
当 $t_{pd} \neq 0$ 时
在 t_1-t_2 内，输出尖峰脉冲



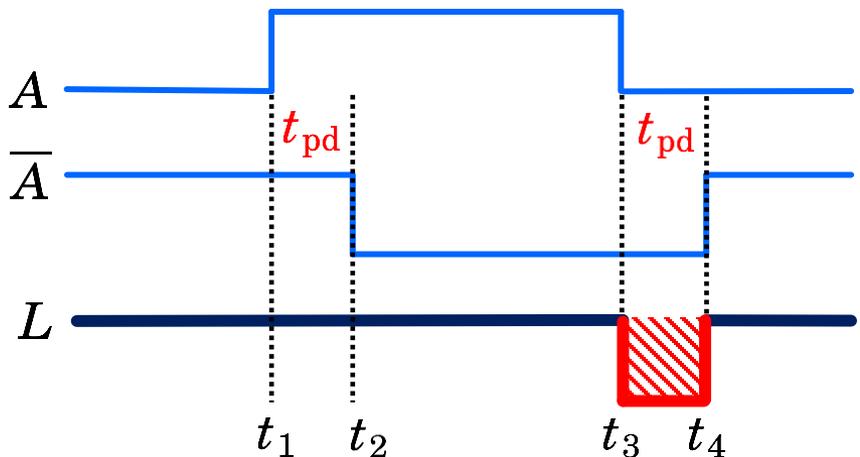
3.3 组合逻辑电路中的“竞争—冒险”

二. 竞争—冒险现象产生的原因？

✧ 器件延时不同



当 $t_{pd} \neq 0$ 时
在 t_3-t_4 内，输出尖峰脉冲



✧ 通常情况下

- ① $A \cdot \bar{A}$ 产生 1 型险象
- ② $A + \bar{A}$ 产生 0 型险象



3.3 组合逻辑电路中的“竞争—冒险”

三. 如何判断竞争—冒险现象？

✧ 代数法

- 检查是否存在互补变量
- 检查是否出现互补变量的“与”“或”形式，即
 - ① 表达式在一定条件下成为 $\bar{X} + X$ 或者 $\bar{X} \cdot X$ 的形式
 - ② 此时可能~~可能~~出现竞争—冒险现象



3.3 组合逻辑电路中的“竞争—冒险”

例-补

例 判断下面函数能否产生逻辑竟冒险现象？

$$L = \overline{A}\overline{C} + \overline{A}B + AC$$

解 变量 A 和 C 具备竞争的条件，分别进行检查

检查 A

$$BC = 00 \Rightarrow L = \overline{A}$$

$$BC = 01 \Rightarrow L = A$$

$$BC = 10 \Rightarrow L = \overline{A}$$

$$BC = 11 \Rightarrow L = A + \overline{A}$$

检查 B

$$AB = 00 \Rightarrow L = \overline{C}$$

$$AB = 01 \Rightarrow L = 1$$

$$AB = 10 \Rightarrow L = C$$

$$AB = 11 \Rightarrow L = C$$

当 $B = C = 1$ 时， A 的变化可能使电路产生险象
无论 A 和 B 为何值， C 的变化时不会产生险象

3.3 组合逻辑电路中的“竞争—冒险”

三. 如何判断竞争—冒险现象？

✧ 卡诺图法

- 使用条件：逻辑函数为“与或”式
- 使用方法：观察是否存在“相切”的卡诺圈，若存在则可能产生冒险

例 判断下面函数能否产生逻辑竟冒险现象？

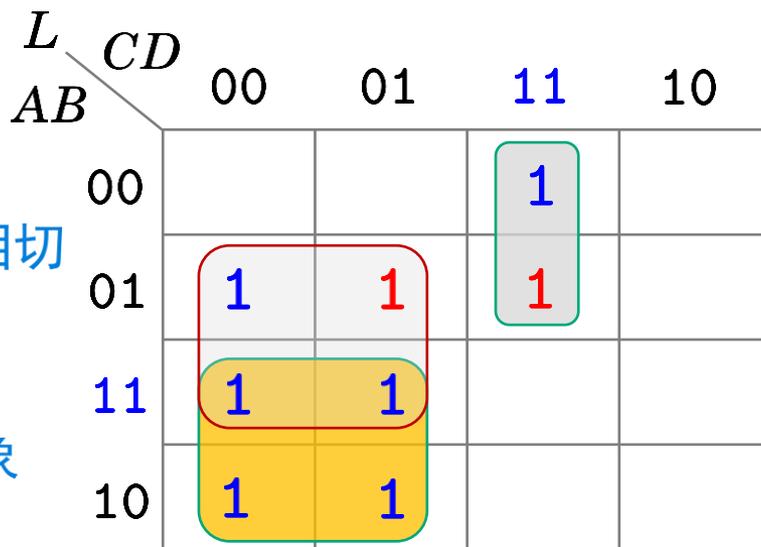
$$L = B\bar{C} + A\bar{C} + \bar{A}CD$$

解 卡诺图如右

相邻最小项 $\bar{A}B\bar{C}D$ 与 $\bar{A}BCD$ 卡诺圈相切

当 $B = D = 1, A = 0$ 时, $L = \bar{C} + C$

电路可能由于 C 的变化而产生冒险现象





3.3 组合逻辑电路中的“竞争—冒险”

四. 如何消除竞争—冒险现象？

① 修改设计，增加冗余项

- 原理：表达式中“加”多余的“与项”或者“乘”多余的“或项”，使原函数不再出现互补变量的“与”“或”

• 两种方式：

① 利用定理： $L = AC + B\bar{C} = AC + B\bar{C} + AB$

当 $A = B = 1$ 时，由 $L = C + \bar{C}$ （冒险）变成了 $L = 1$ （无冒险）

② 在卡诺图中增加卡诺圈，消除“相切”



3.3 组合逻辑电路中的“竞争—冒险”

例-补

例 请用增加冗余项（增加卡诺圈）的方法消除下列逻辑表达式中的险象。

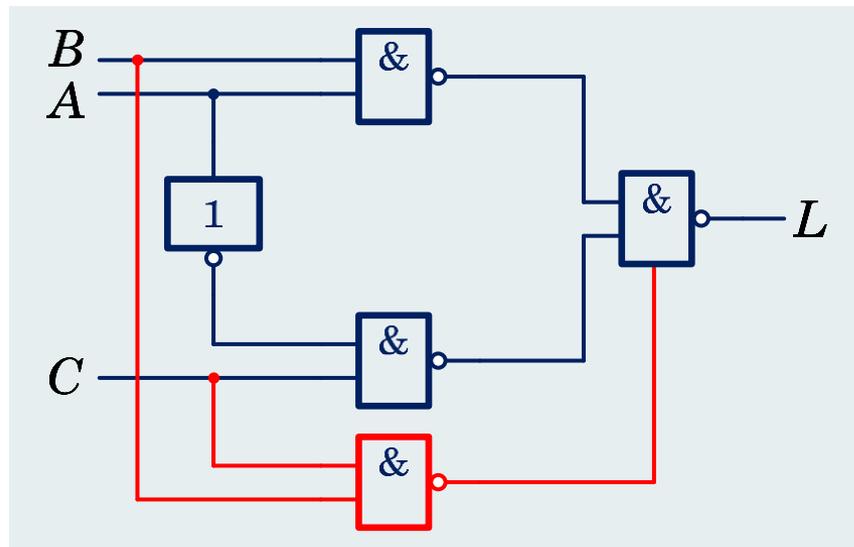
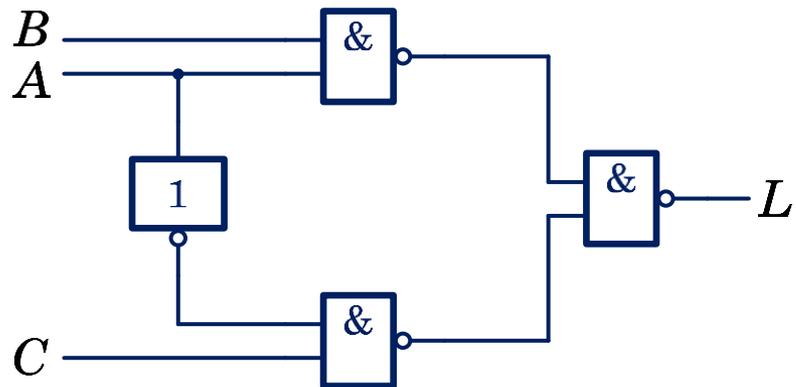
解 列电路逻辑函数表达式

$$L = \overline{\overline{AB}} \cdot \overline{\overline{AC}} = AB + \overline{AC}$$

根据定理，增加冗余项 BC

$$\begin{aligned} L &= AB + \overline{AC} + BC \\ &= \overline{\overline{AB}} \cdot \overline{\overline{AC}} \cdot \overline{\overline{BC}} \end{aligned}$$

用附加与非门改进后的电路



3.3 组合逻辑电路中的“竞争—冒险”

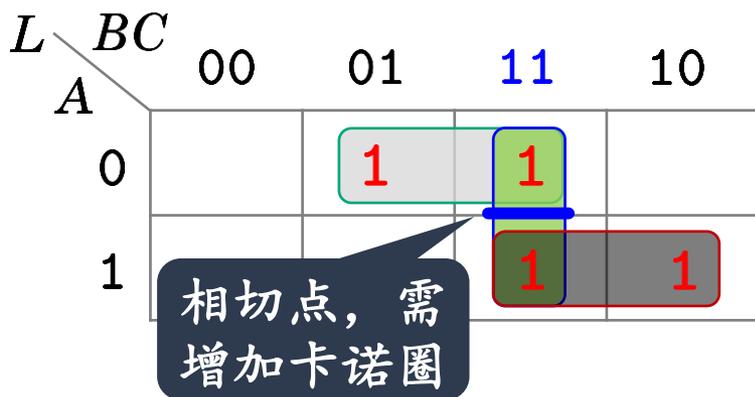
例-补

例 请用增加冗余项的方法消除图示电路中的险象。

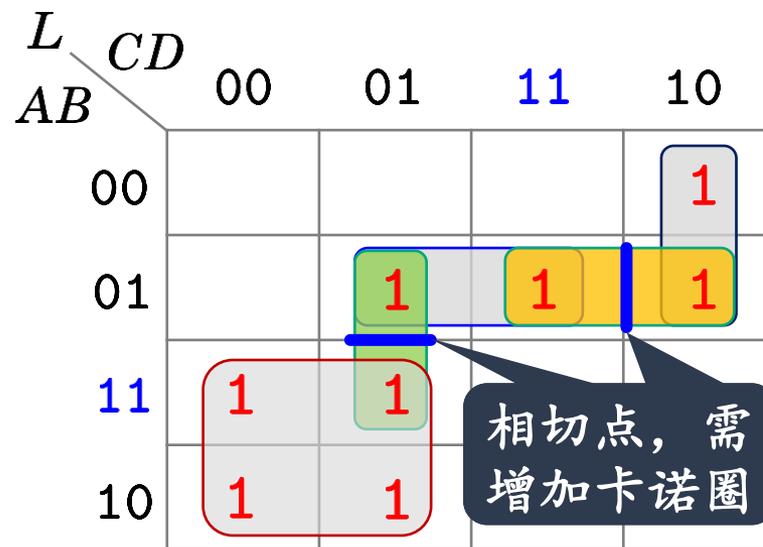
$$L = AB + \bar{A}C$$

$$L = A\bar{C} + \bar{A}BD + \bar{A}C\bar{D}$$

解 在卡诺图中增加卡诺圈，消除“相切”



$$L = AB + \bar{A}C + BC$$



$$L = A\bar{C} + \bar{A}BD + B\bar{C}D + \bar{A}C\bar{D} + \bar{A}BC$$

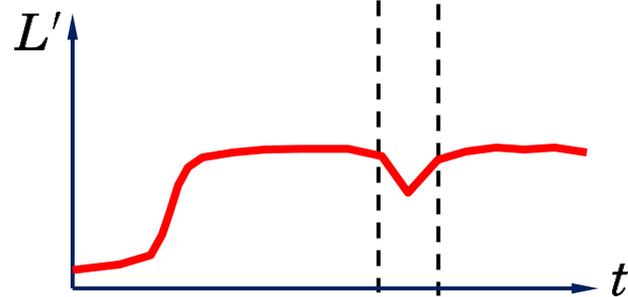
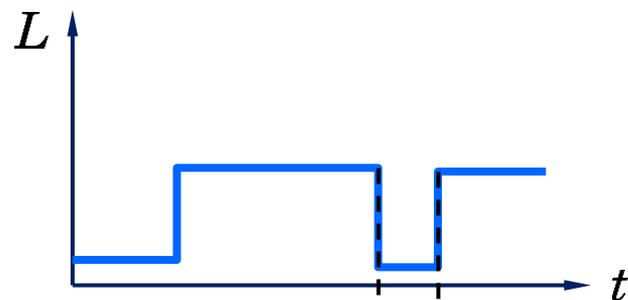
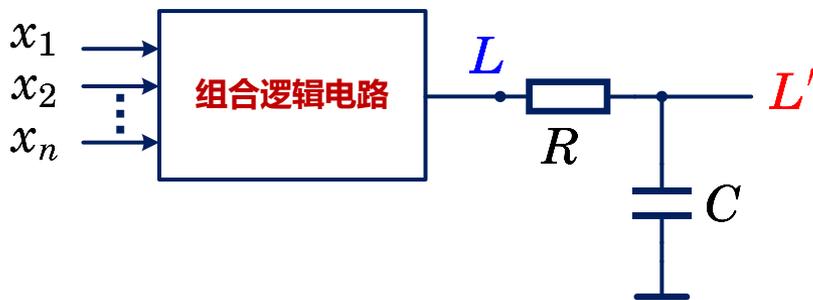


3.3 组合逻辑电路中的“竞争—冒险”

四. 如何消除竞争—冒险现象？

② 增加惯性延时环节

- 实现方法：在电路输出端连接一个惯性延时环节，通常用RC滤波器



注意要合适的时间常数 $\tau = RC$ ，要求 τ 足够大，以便“削平”尖脉冲；但又不能太大，防止正常输出发生畸变

3.3 组合逻辑电路中的“竞争—冒险”

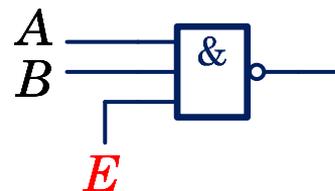
四. 如何消除竞争—冒险现象？

③ 加选通脉冲

- 实现方法:

- ① 先使 $E = 0$ ，关闭与非门

- ② 等 A 、 B 信号都来到后，让 $E = 1$ ，得到可靠的 $L = AB$



④ 其他方法

- 计算机仿真
- 实验验证

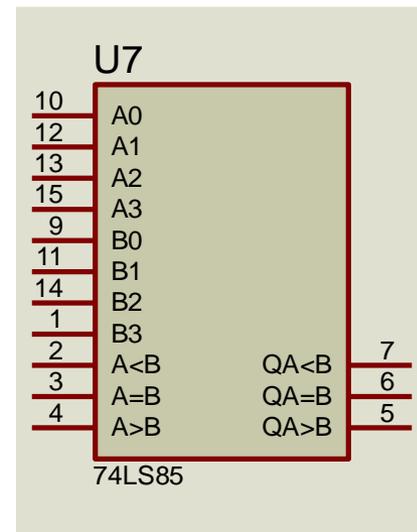
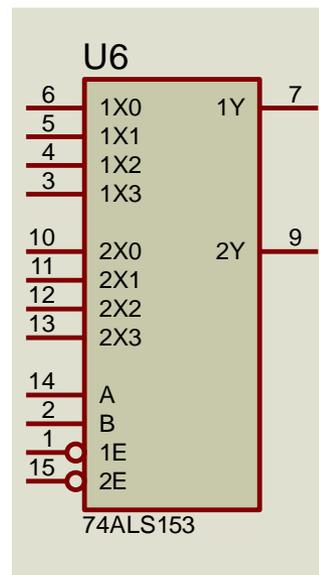
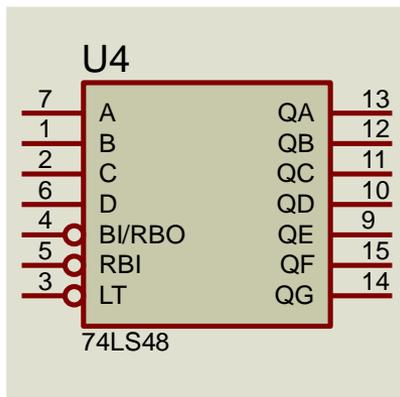
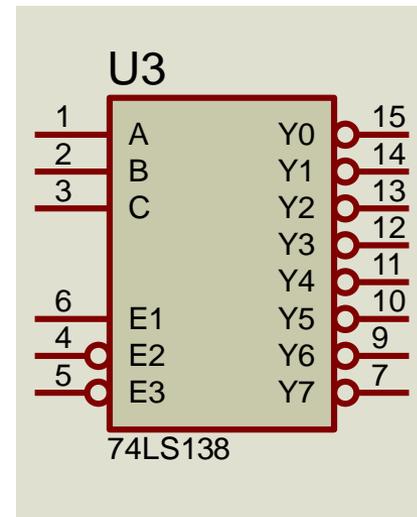
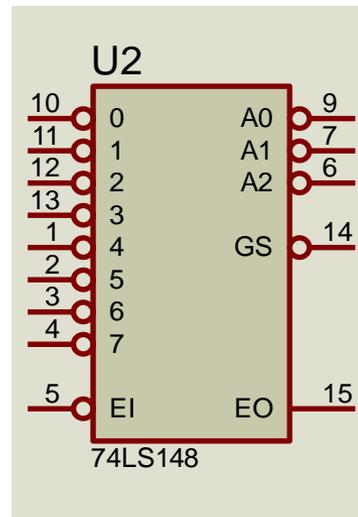
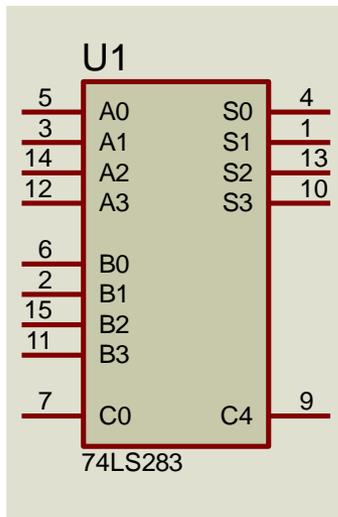


§3.4 常用的集成组合逻辑电路

3.4 常用的集成组合逻辑电路

■ 中等规模集成电路

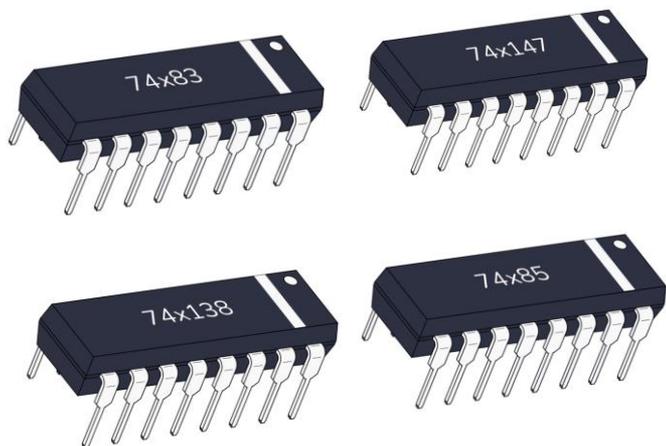
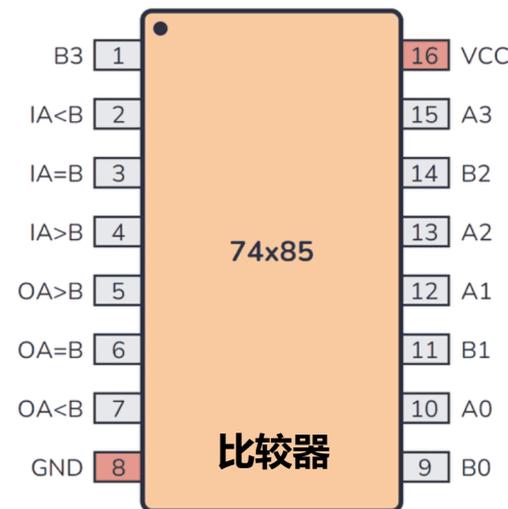
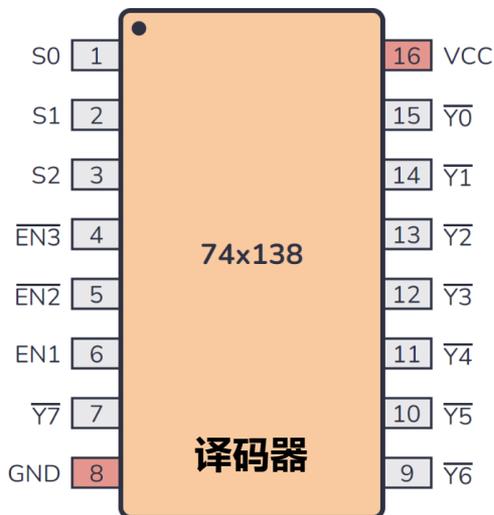
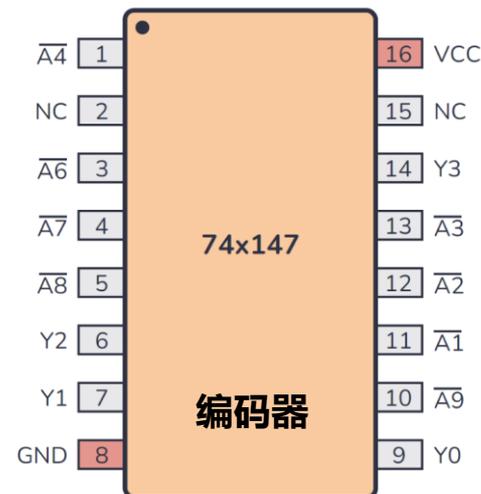
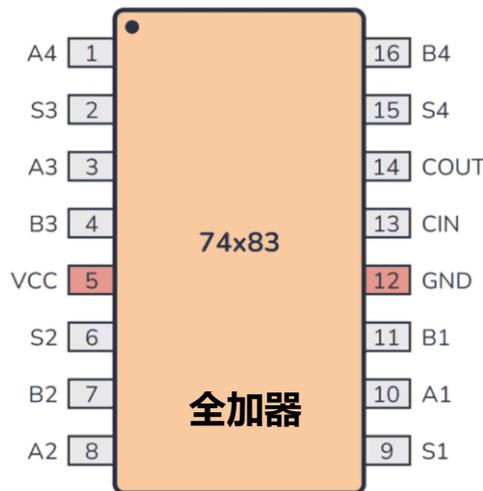
- ① 加法器
- ② 编码器
- ③ 译码器
- ④ 数据选择器
- ⑤ 数值比较器



3.4 常用的集成组合逻辑电路

■ 中等规模集成电路

- ① 加法器
- ② 编码器
- ③ 译码器
- ④ 数据选择器
- ⑤ 数值比较器





3.4 常用的集成组合逻辑电路

一. 加法器

◇ 实现二进制加法运算的电路

- 是构成计算机中算术运算电路的基本单元

◇ **1位加法器**

- 1位半加法器
- 1位全加法器

◇ **多位加法器**

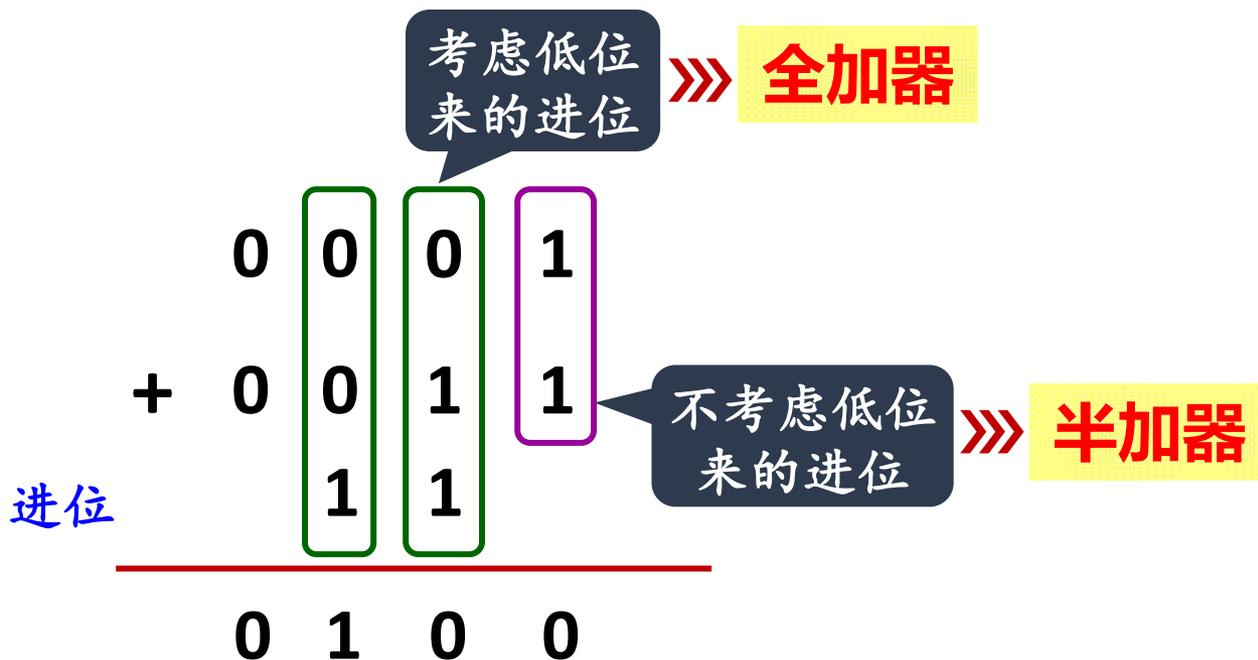
- 串行加法器
- 并行加法器



3.4 常用的集成组合逻辑电路

一. 加法器：1位加法器

◇ 举例





3.4 常用的集成组合逻辑电路

一. 加法器：1位全加器设计



◇ 逻辑抽象

	A_i			
+	B_i	C_{i-1}		
	C_i	S_i		

A_i — 加数

B_i — 被加数

C_{i-1} — 低位的进位

S_i — 本位和

C_i — 进位

◇ 真值表

A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

3.4 常用的集成组合逻辑电路

一. 加法器：1位全加器设计

✧ 逻辑表达式及其变换

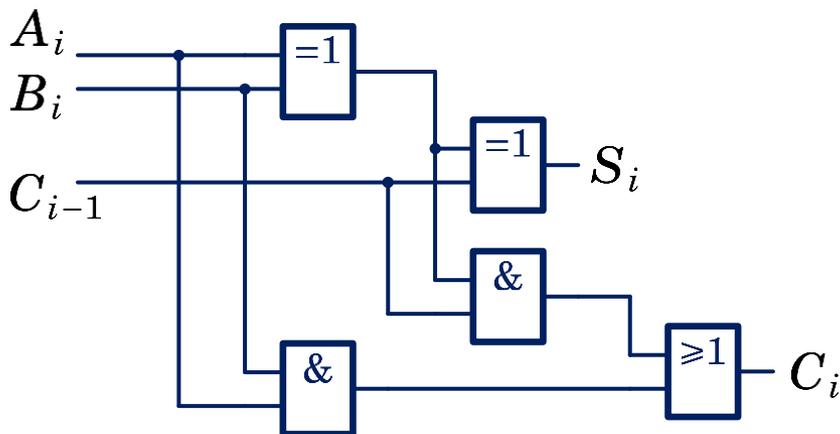
$$S_i = (A_i \bar{B}_i + \bar{A}_i B_i) \bar{C}_{i-1} + (\bar{A}_i \bar{B}_i + A_i B_i) C_{i-1}$$

$$= A_i \oplus B_i \oplus C_{i-1}$$

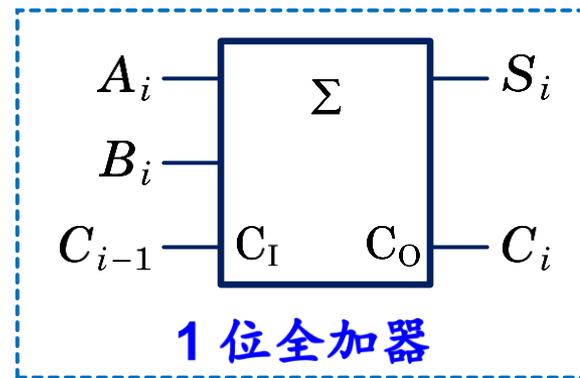
$$C_i = (\bar{A}_i B_i + A_i \bar{B}_i) C_{i-1} + A_i B_i$$

$$= (A_i \oplus B_i) C_{i-1} + A_i B_i$$

✧ 逻辑电路图



A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



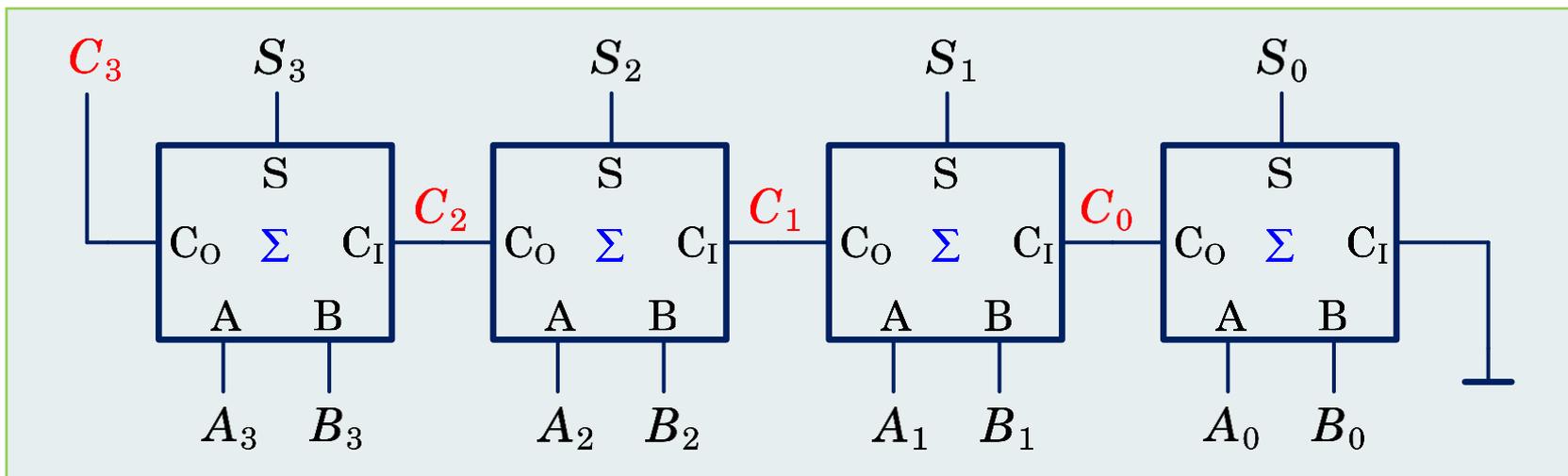
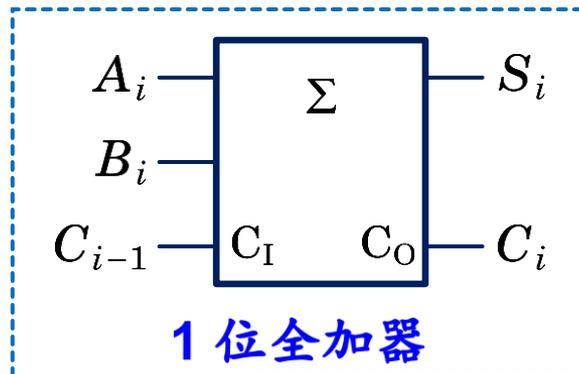
3.4 常用的集成组合逻辑电路

一、加法器：多位串行加法器

✧ 构成方式

- 将n位全加器串联，低位全加器的进位输出连接到相邻的高位全加器的进位输入

$$A_3A_2A_1A_0 + B_3B_2B_1B_0 = C_3S_3S_2S_1S_0$$



- ✧ **缺点**：进位信号是由低位向高位逐级传递的，速度不高

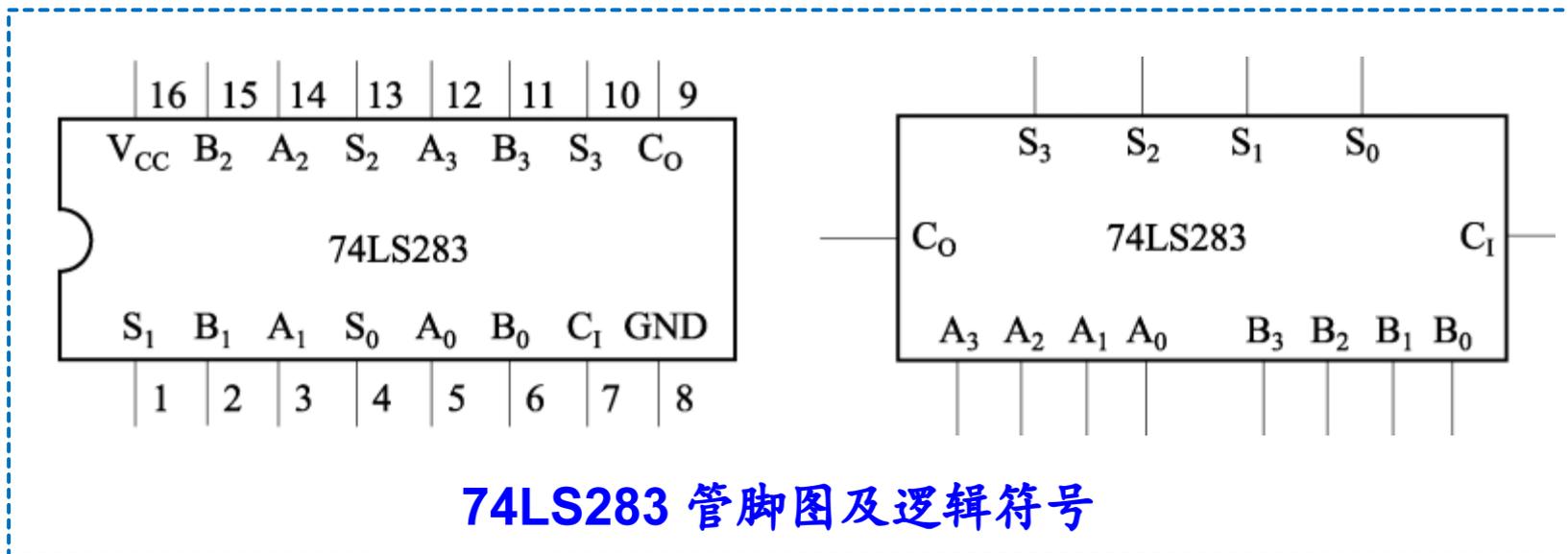
3.4 常用的集成组合逻辑电路

一、加法器：多位并行加法器（超前进位加法器）

✧ 进位并行产生，取代逐级传递

$$A_3A_2A_1A_0 + B_3B_2B_1B_0 + C_{-1} = C_3S_3S_2S_1S_0$$

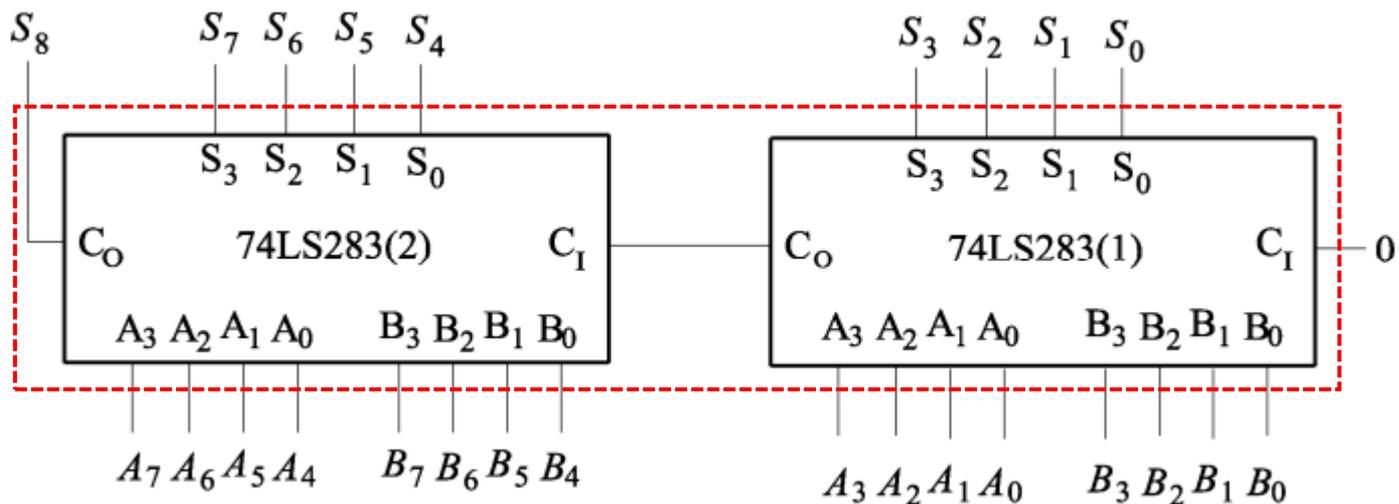
- 常用4位超前进位加法器有74LS283、CD4008等



3.4 常用的集成组合逻辑电路

一、加法器：74LS283的应用

① 加法器级联实现多位二进制数加法运算



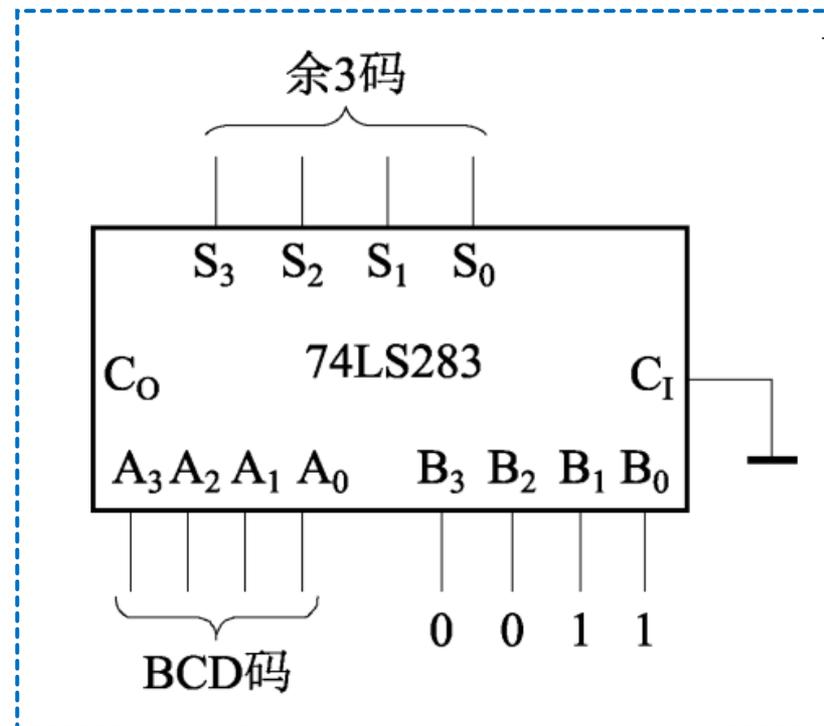
2 片 74LS283 组成的 8 位二进制数加法电路

3.4 常用的集成组合逻辑电路

一. 加法器：74LS283的应用

② 将8421BCD码转换成余3码

余3码 = 8421BCD + 0011

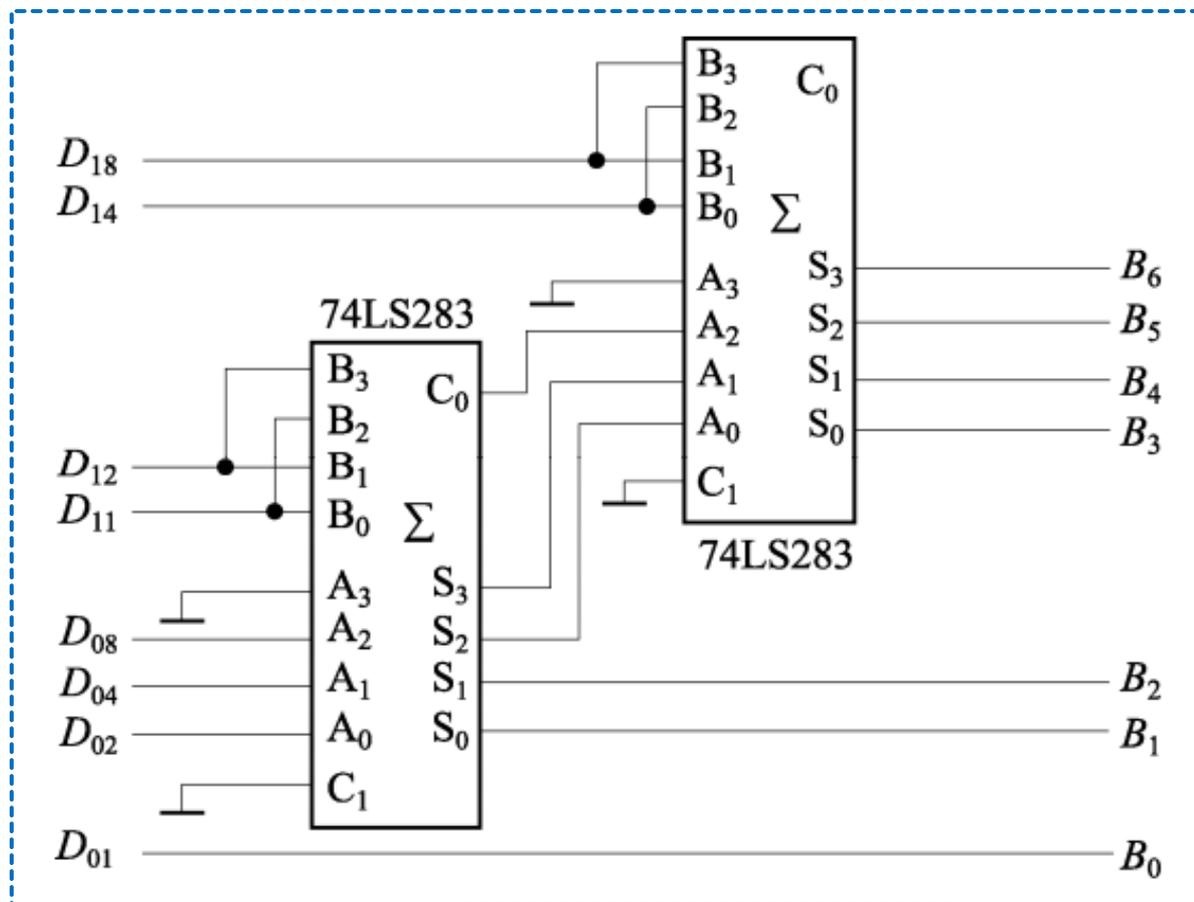


3.4 常用的集成组合逻辑电路

一. 加法器：74LS283的应用

③ 将两位8421BCD码转换成二进制码

- 码制变换可以看做七位二进制数的加法运算





3.4 常用的集成组合逻辑电路

二. 编码器

◇ 实现编码功能的逻辑电路

- **编码**：用符号或数字表示特定对象的过程，一般用二进制代码表示

◇ 编码器类型

- **普通编码器**：任何时候只允许一个编码输入信号有效
- **优先编码器**：允许同时输入两个以上的有效编码信号

n 位二进制代码有 2^n 种组合，可以表示 2^n 个信息



3.4 常用的集成组合逻辑电路

二. 编码器：普通编码器（设计一个8线/3线普通编码器）



◇ 逻辑抽象

- ▣ 8个输入：设为 $A_0 \sim A_7$ ，且高电平有效
- ▣ 3位二进制代码输出：设为 $L_2、L_1、L_0$
- ▣ 约束关系：仅允许一个输入信号有效

◇ 列真值表

A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	L_2	L_1	L_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$L_2 = A_4 + A_5 + A_6 + A_7$$

$$L_1 = A_2 + A_3 + A_6 + A_7$$

$$L_0 = A_1 + A_3 + A_5 + A_7$$

根据输出的3为二进制，可直接确定输入端子哪个为高电平（0除外）

3.4 常用的集成组合逻辑电路

二. 编码器：普通编码器（设计一个8线/3线普通编码器）

✧ 列逻辑表达式

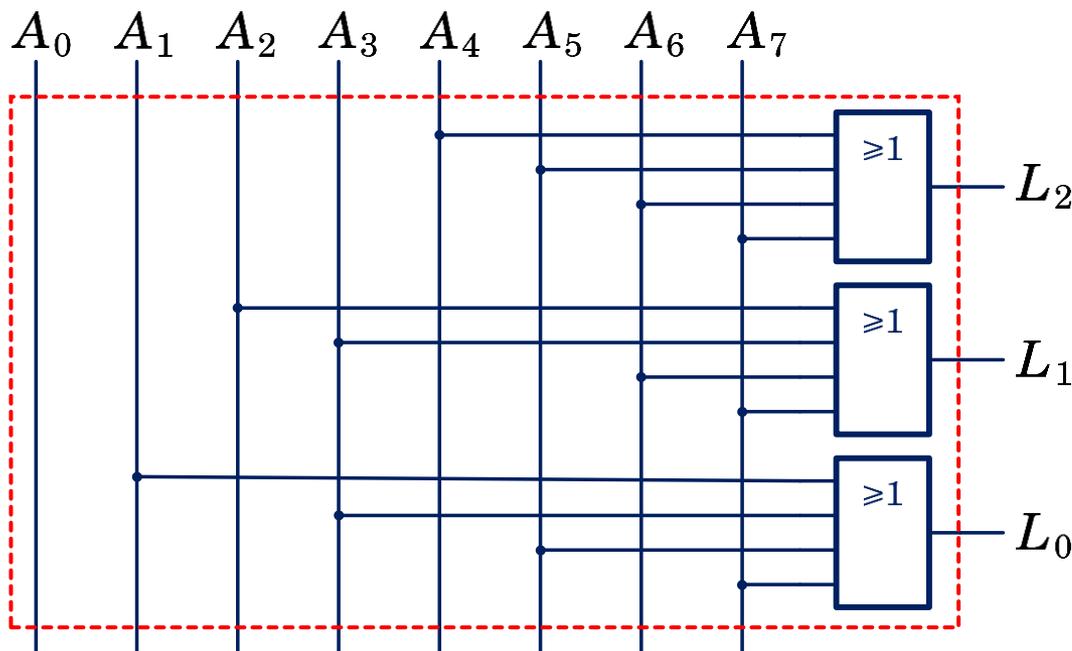
$$L_2 = A_4 + A_5 + A_6 + A_7$$

$$L_1 = A_2 + A_3 + A_6 + A_7$$

$$L_0 = A_1 + A_3 + A_5 + A_7$$

✧ 画逻辑电路图

当所有的输入都为0时，无法区分是哪个输入信号的编码





3.4 常用的集成组合逻辑电路

二. 编码器：优先编码器

✧ 含义

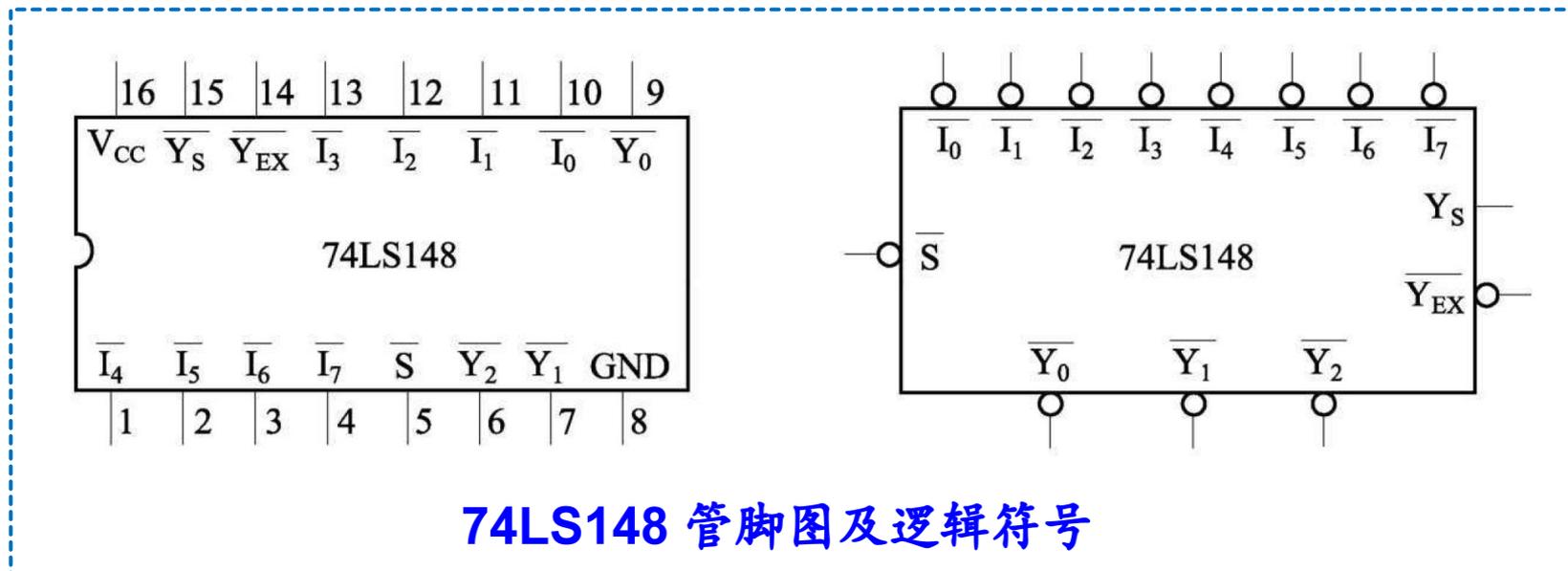
- 识别多个编码请求信号的优先级别，并进行相应编码的逻辑部件

✧ 优点

- 无论有多少个编码信号有效，只对优先级别最高的那个进行编码

3.4 常用的集成组合逻辑电路

二. 编码器：集成3位二进制优先编码器（74LS148）



- 🔗 8 位输入 $\bar{I}_0 \sim \bar{I}_7$ ，低电平有效，高位优先
- 🔗 3 位反码输出 $\bar{Y}_0 \sim \bar{Y}_2$
- 🔗 1 个输入使能端 \bar{S} ：使能端有效（低电平）时，允许编码，否则禁止
- 🔗 2 个输出扩展端 Y_S （选通输出）、 Y_{EX} （低电平表示已编码）



3.4 常用的集成组合逻辑电路

二. 编码器：集成3位二进制优先编码器（74LS148）

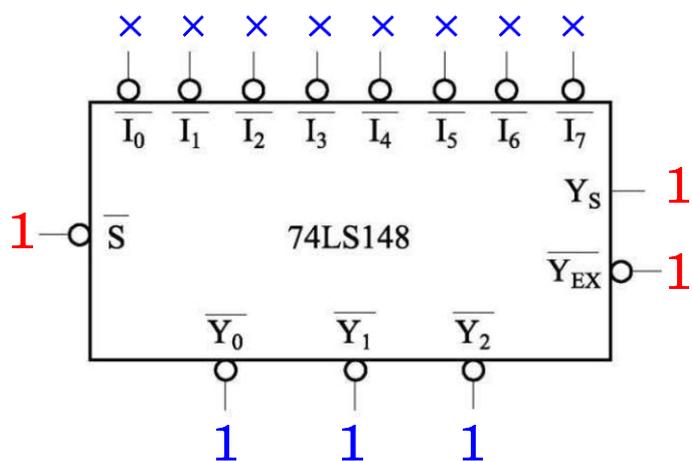
✧ 逻辑功能表

\overline{S}	\overline{I}_0	\overline{I}_1	\overline{I}_2	\overline{I}_3	\overline{I}_4	\overline{I}_5	\overline{I}_6	\overline{I}_7	\overline{Y}_2	\overline{Y}_1	\overline{Y}_0	\overline{Y}_{EX}	Y_S
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	×	×	×	×	×	×	×	0	0	0	0	0	1
0	×	×	×	×	×	×	0	1	0	0	1	0	1
0	×	×	×	×	0	1	1	1	0	1	1	0	1
0	×	×	×	0	1	1	1	1	1	0	0	0	1
0	×	×	0	1	1	1	1	1	1	0	1	0	1
0	×	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

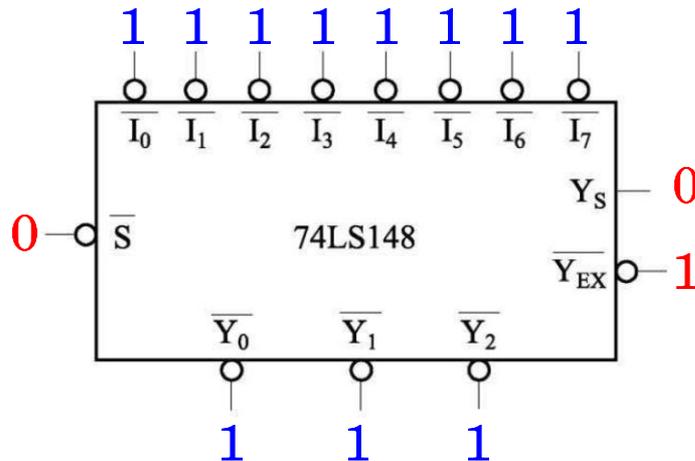
3.4 常用的集成组合逻辑电路

二. 编码器：集成3位二进制优先编码器（74LS148）

◇ 三种工作状态（均输出111）



编码器不工作

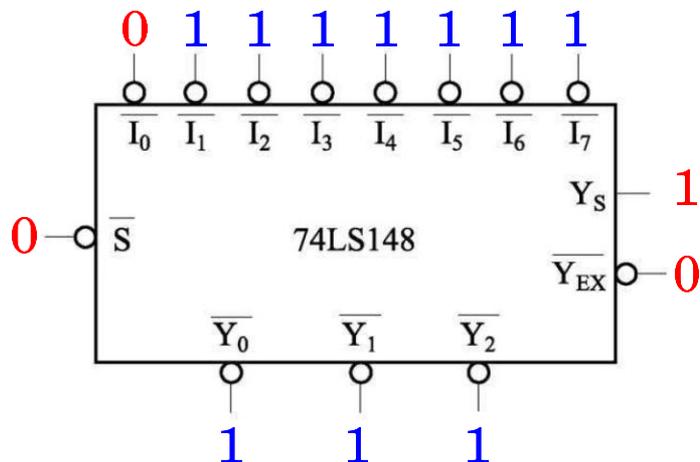


编码器工作，但没有申请编码的信号

3.4 常用的集成组合逻辑电路

二. 编码器：集成3位二进制优先编码器（74LS148）

◇ 三种工作状态（均输出111）



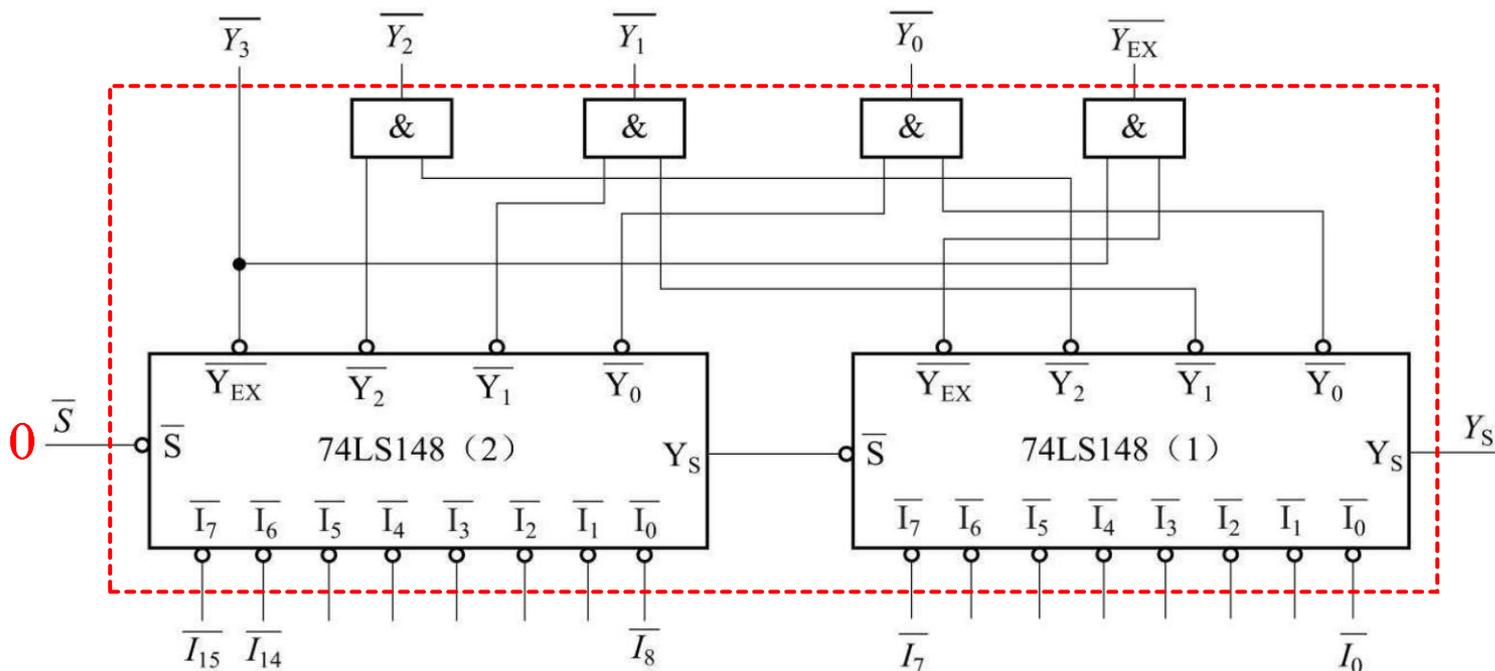
编码器工作，并且只有优先级最低的信号申请编码

3.4 常用的集成组合逻辑电路

二. 编码器：74LS148的应用

① 多片74LS148级联，实现更多信号编码

- 举例：编码16个信号（16线/4线优先编码器）



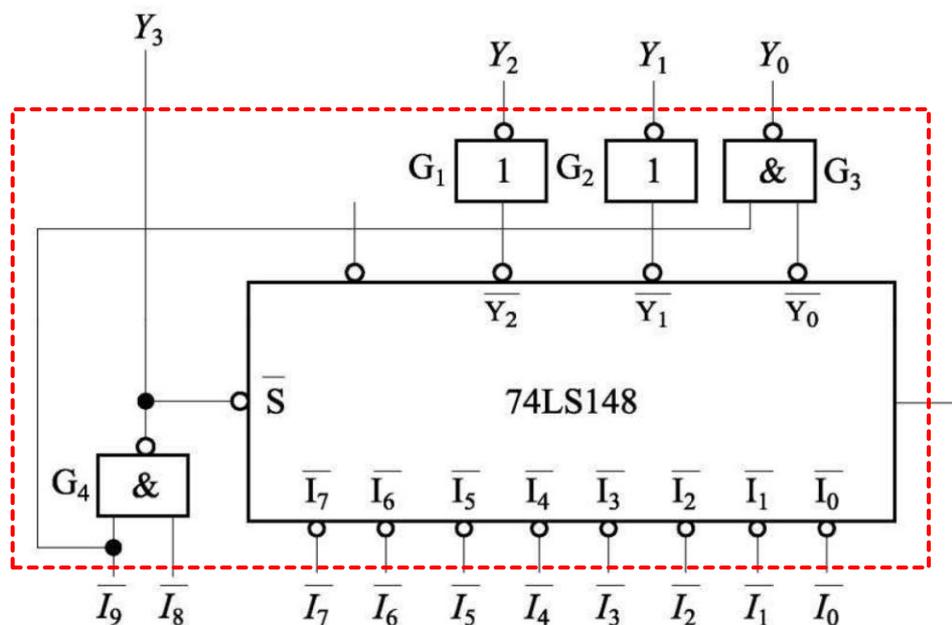
- ① 片2无有效输入时，片1正常工作，编码17-10
- ② 片2有有效输入时，片1禁止工作，编码115-18

3.4 常用的集成组合逻辑电路

二. 编码器：74LS148的应用

② 利用74LS148实现8421BCD编码

- 0~9 (0000~1001)





3.4 常用的集成组合逻辑电路

三. 译码器

- ✧ 将具有特定含义的**二进制码转换成对应的输出信号**的电路
 - **译码**：编码的逆过程
- ✧ 译码器类型
 - **变量译码器**：每一组输入变量，都对应一个输出线有信号表示
 - **码制变换译码器**：将一种码制转换为另一种码制
 - **显示译码器**：将数码所代表的“数字”显示出来

3.4 常用的集成组合逻辑电路

三. 译码器：变量译码器（设计一个2线/4线译码器）



◇ 逻辑抽象

- ▣ 2个输入：A、B，高电平有效
- ▣ 4个输出： \bar{Y}_3 、 \bar{Y}_2 、 \bar{Y}_1 、 \bar{Y}_0 ，低电平有效
- ▣ 1个使能端： \bar{EI}

◇ 列真值表

输入			输出			
\bar{EI}	A	B	\bar{Y}_3	\bar{Y}_2	\bar{Y}_1	\bar{Y}_0
1	×	×	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

◇ 列逻辑表达式

$$\bar{Y}_3 = \bar{\bar{\bar{EI}}} \cdot AB$$

$$\bar{Y}_2 = \bar{\bar{\bar{EI}}} \cdot A\bar{B}$$

$$\bar{Y}_1 = \bar{\bar{\bar{EI}}} \cdot \bar{A}B$$

$$\bar{Y}_0 = \bar{\bar{\bar{EI}}} \cdot \bar{A}\bar{B}$$

3.4 常用的集成组合逻辑电路

三. 译码器：变量译码器（设计一个2线/4线译码器）

✧ 列逻辑表达式

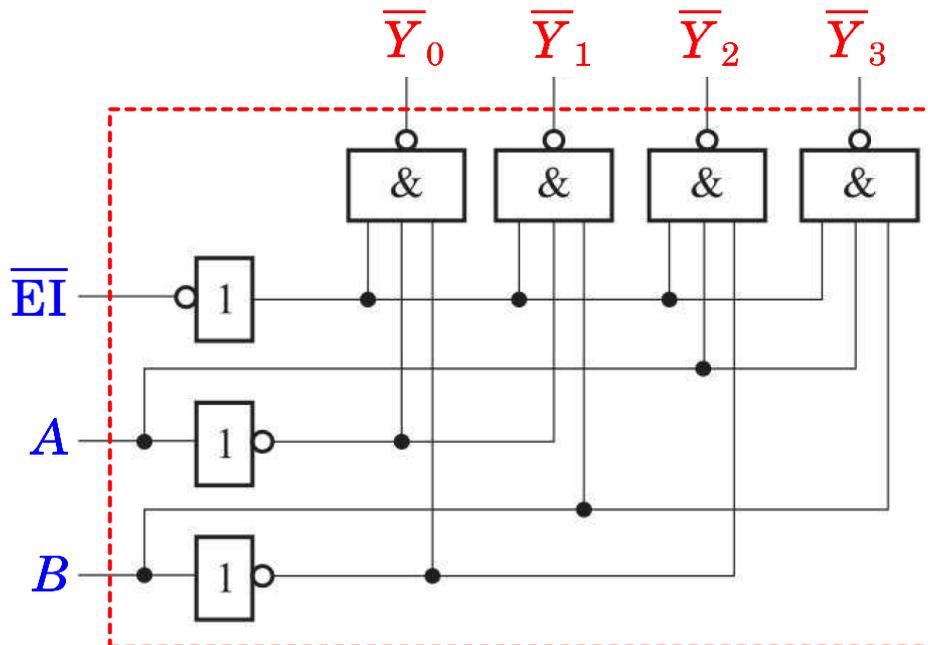
$$\bar{Y}_3 = \overline{\overline{\overline{EI}} \cdot AB}$$

$$\bar{Y}_2 = \overline{\overline{\overline{EI}} \cdot A\bar{B}}$$

$$\bar{Y}_1 = \overline{\overline{\overline{EI}} \cdot \bar{A}B}$$

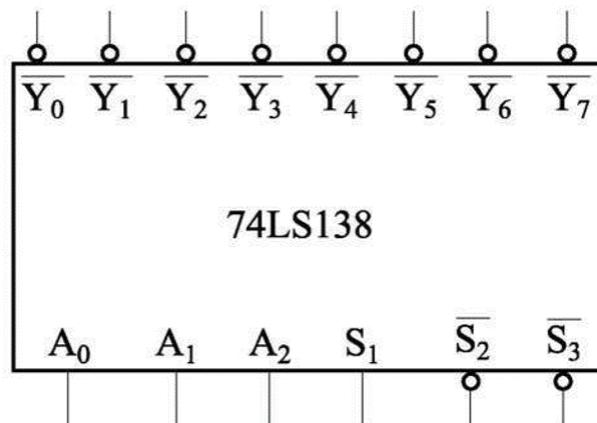
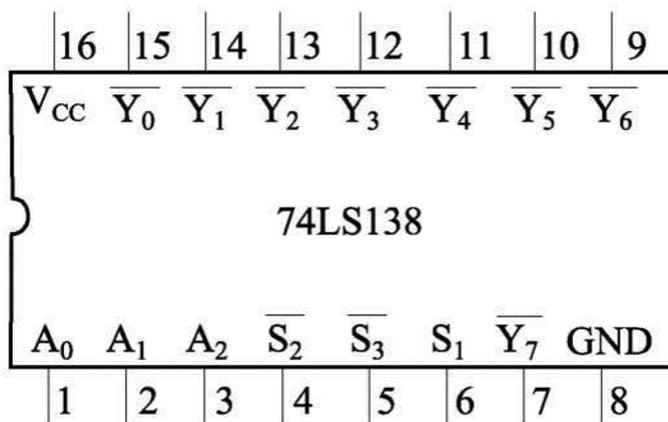
$$\bar{Y}_0 = \overline{\overline{\overline{EI}} \cdot \bar{A}\bar{B}}$$

✧ 画逻辑电路图



3.4 常用的集成组合逻辑电路

三. 译码器：集成变量译码器74LS138（3线/8线译码器）



74LS138 管脚图及逻辑符号

- 3 个输入端 $A_2 \sim A_0$
- 8 个输出 $\bar{Y}_0 \sim \bar{Y}_7$ ，低电平有效
- 3 个输入使能端 S_1 、 \bar{S}_2 、 \bar{S}_3 ：需同时有效才有译码信号输出

3.4 常用的集成组合逻辑电路

三. 译码器：集成变量译码器74LS138（3线/8线译码器）

◇ 逻辑功能表

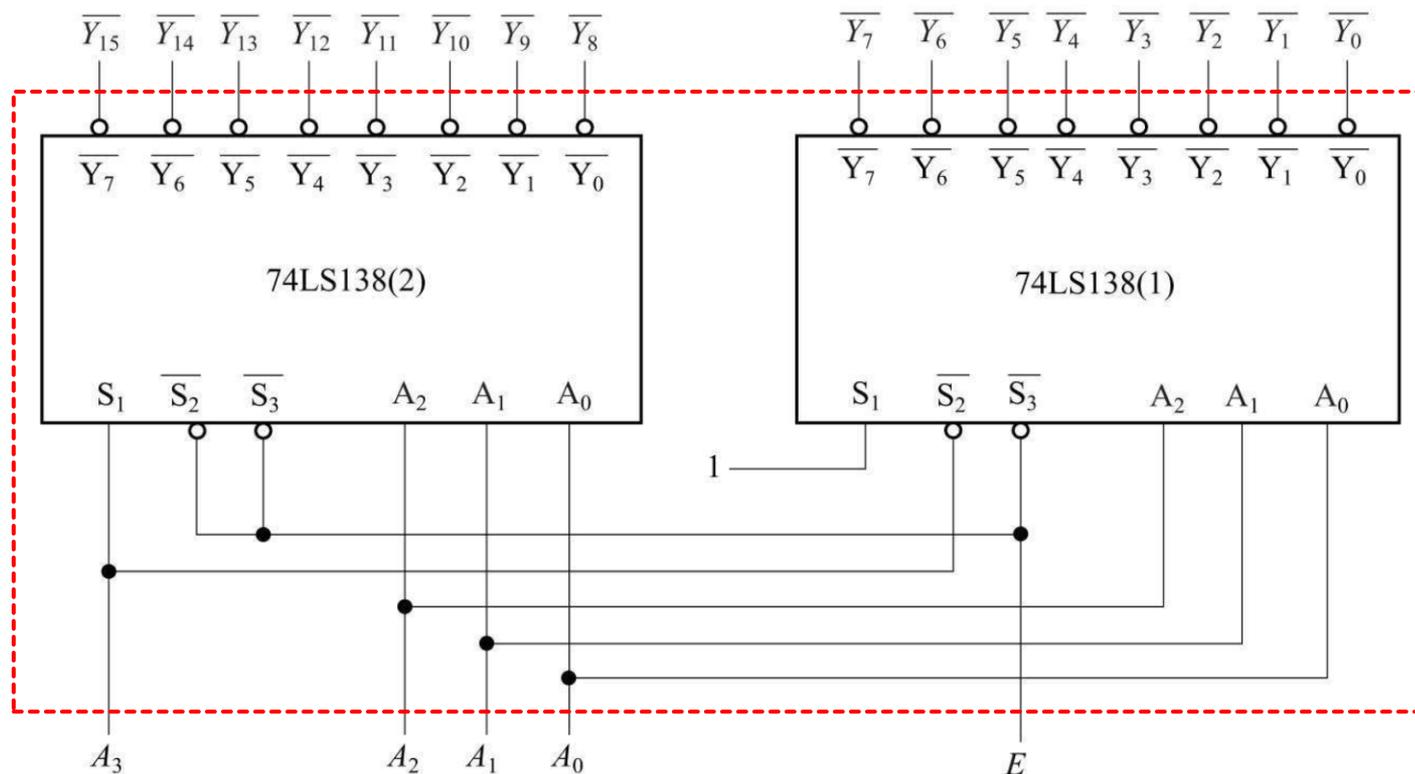
输入						输出							
S_1	\bar{S}_2	\bar{S}_3	A_2	A_1	A_0	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7
×	1	×	×	×	×	1	1	1	1	1	1	1	1
×	×	1	×	×	×	1	1	1	1	1	1	1	1
0	×	×	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

3.4 常用的集成组合逻辑电路

三. 译码器：74LS138的应用

① 译码器的扩展

- 用138扩展为4线/16线译码器





3.4 常用的集成组合逻辑电路

三. 译码器：74LS138的应用

② 实现组合逻辑电路

- 译码器的每个输出端分别与一个最小项相对应，因此辅以适当的门电路，便可实现组合逻辑函数

n 线 / 2^n 线的译码器，可产生不多于 n 个变量的任意逻辑函数

✧ 实现步骤

- ① 把原函数化为最小项之和形式
- ② 根据函数的变量数 n ，确定用 n 线 / 2^n 线译码器
- ③ 译码器输出 0 有效时，输出端应附加与非门

✧ 注意事项

- ① 函数变量的权位应与所用译码器输入代码的权位相对应
- ② 控制端的条件要满足芯片正常工作的状态



3.4 常用的集成组合逻辑电路

例3-4

例 请用译码器和门电路实现逻辑函数 $L = AB + BC + AC$ 。

解 (1) 将逻辑函数转换成最小项表达式

$$L = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

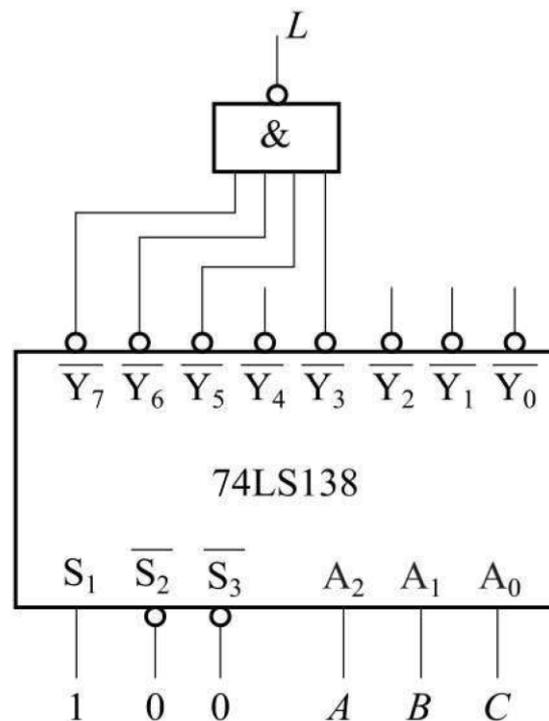
$$= m_3 + m_5 + m_6 + m_7$$

(2) 可以选用 74LS138 译码器实现

(3) 转换成“与非—与非”形式

$$L = \overline{\bar{m}_3 \cdot \bar{m}_5 \cdot \bar{m}_6 \cdot \bar{m}_7}$$

(4) 绘出电路图

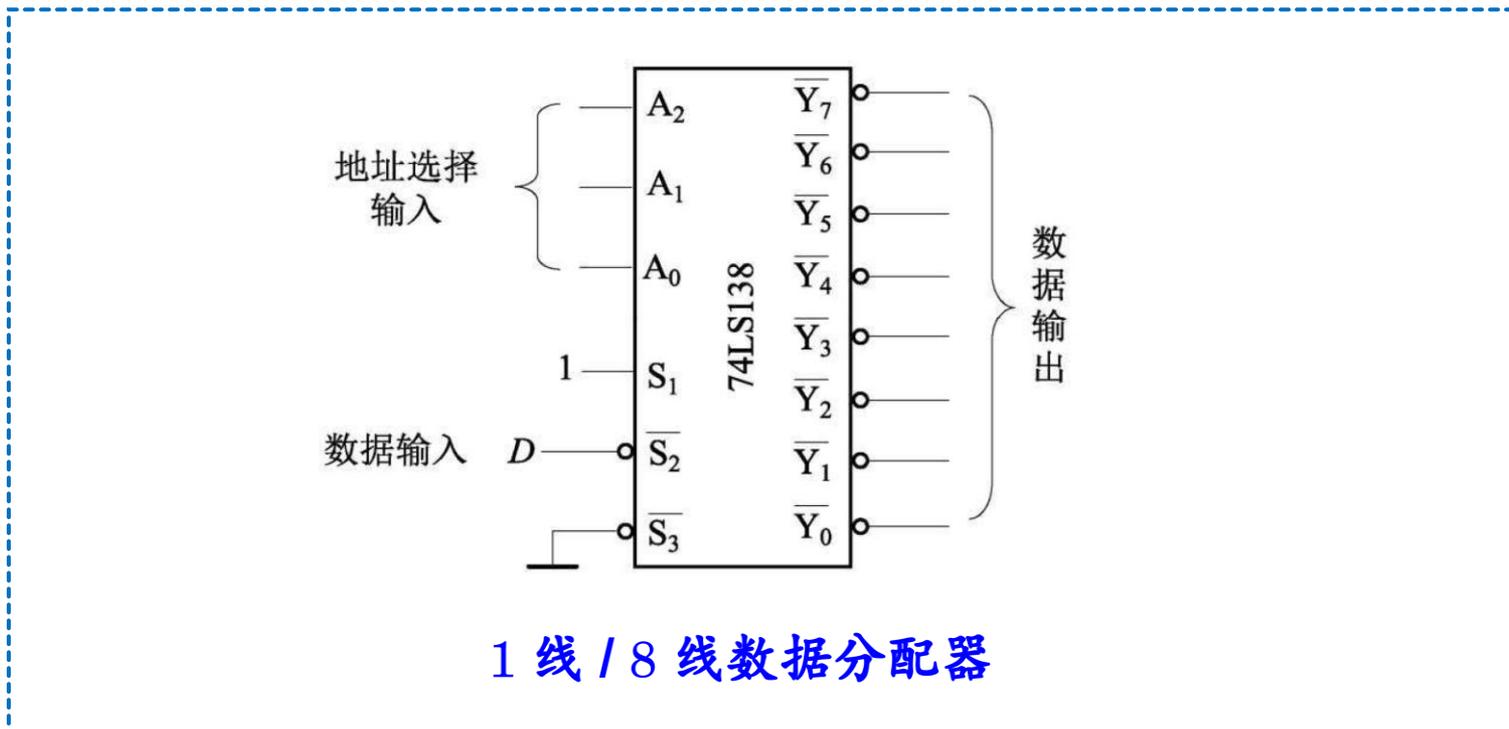


3.4 常用的集成组合逻辑电路

三. 译码器：74LS138的应用

③ 构成数据分配器

- 将一路输入数据，根据地址选择码分配给多路数据输出中的某一路

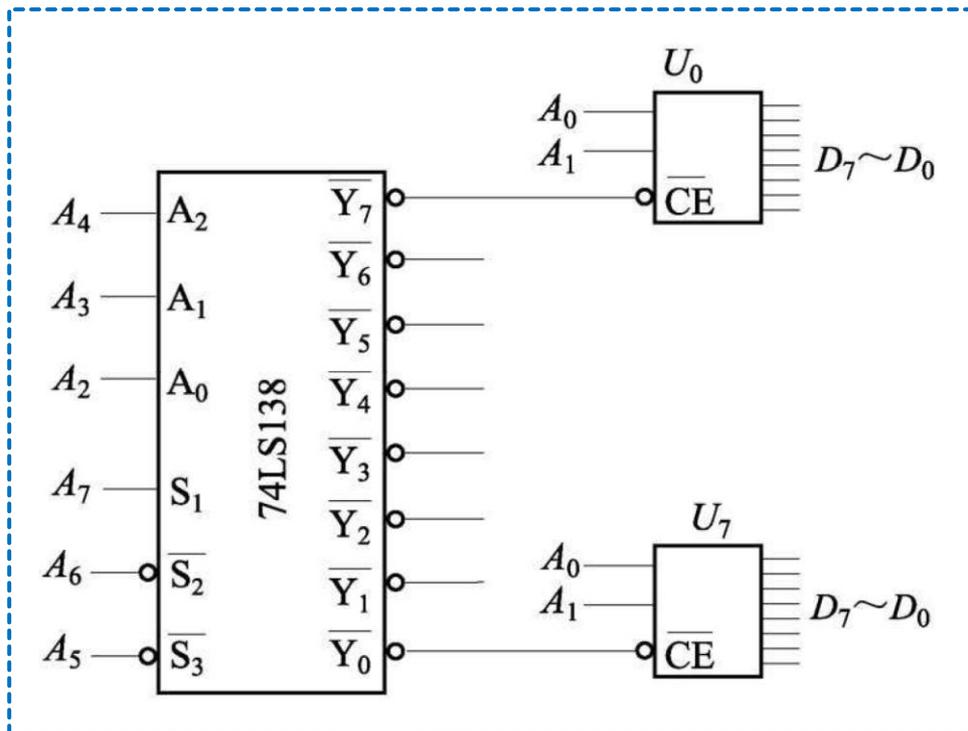


3.4 常用的集成组合逻辑电路

三. 译码器：74LS138的应用

④ 做地址分配

- 实现不同器件的选通



- 当 $A_7 \sim A_2 = 100000$ 时，使能端有效，地址位为 000，所以选通 U_0 器件，其对应的数据即可进行传递
- 由于 $A_0 \sim A_1$ 的值可以任取，所以 U_0 对应的寻址范围为 80H ~ 83H



3.4 常用的集成组合逻辑电路

三. 译码器：码制变换译码器（设计一个4线/10线译码器）

二—十进制译码器真值表													
输入				输出									
A	B	C	D	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈	Y ₉
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

3.4 常用的集成组合逻辑电路

三. 译码器：码制变换译码器（设计一个4线/10线译码器）

✧ 根据对伪码的不同处理可得两种不同的电路

- 部分译码：

把伪码作无关项处理，当干扰等原因导致伪码时，电路输出可能出错

- 完全译码：拒收伪码

✧ 列逻辑表达式（**部分译码设计**）

$$Y_0 = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$

$$Y_5 = B \cdot \bar{C} \cdot D$$

$$Y_1 = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$$

$$Y_6 = B \cdot C \cdot \bar{D}$$

$$Y_2 = \bar{B} \cdot C \cdot \bar{D}$$

$$Y_7 = B \cdot C \cdot D$$

$$Y_3 = \bar{B} \cdot C \cdot D$$

$$Y_8 = A \cdot \bar{D}$$

$$Y_4 = B \cdot \bar{C} \cdot \bar{D}$$

$$Y_9 = A \cdot D$$

L AB	CD			
	00	01	11	10
00	Y ₀	Y ₁	Y ₃	Y ₂
01	Y ₄	Y ₅	Y ₇	Y ₆
11	×	×	×	×
10	Y ₈	Y ₉	×	×

📖 当因干扰原因出现伪码时，例如 $ABCD = 1111$ 时， $Y_7 = BCD = 1$ ， $Y_9 = AD = 1$ ，出现错误



3.4 常用的集成组合逻辑电路

三. 译码器：码制变换译码器（设计一个4线/10线译码器）

✧ 列逻辑表达式（**完全译码设计**）

L AB	CD	00	01	11	10
	00	Y ₀	Y ₁	Y ₃	Y ₂
01	Y ₄	Y ₅	Y ₇	Y ₆	
11	×	×	×	×	
10	Y ₈	Y ₉	×	×	

$$Y_0 = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$

$$Y_5 = \bar{A} \cdot B \cdot \bar{C} \cdot D$$

$$Y_1 = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$$

$$Y_6 = \bar{A} \cdot B \cdot C \cdot \bar{D}$$

$$Y_2 = \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}$$

$$Y_7 = \bar{A} \cdot B \cdot C \cdot D$$

$$Y_3 = \bar{A} \cdot \bar{B} \cdot C \cdot D$$

$$Y_8 = A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$

$$Y_4 = \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}$$

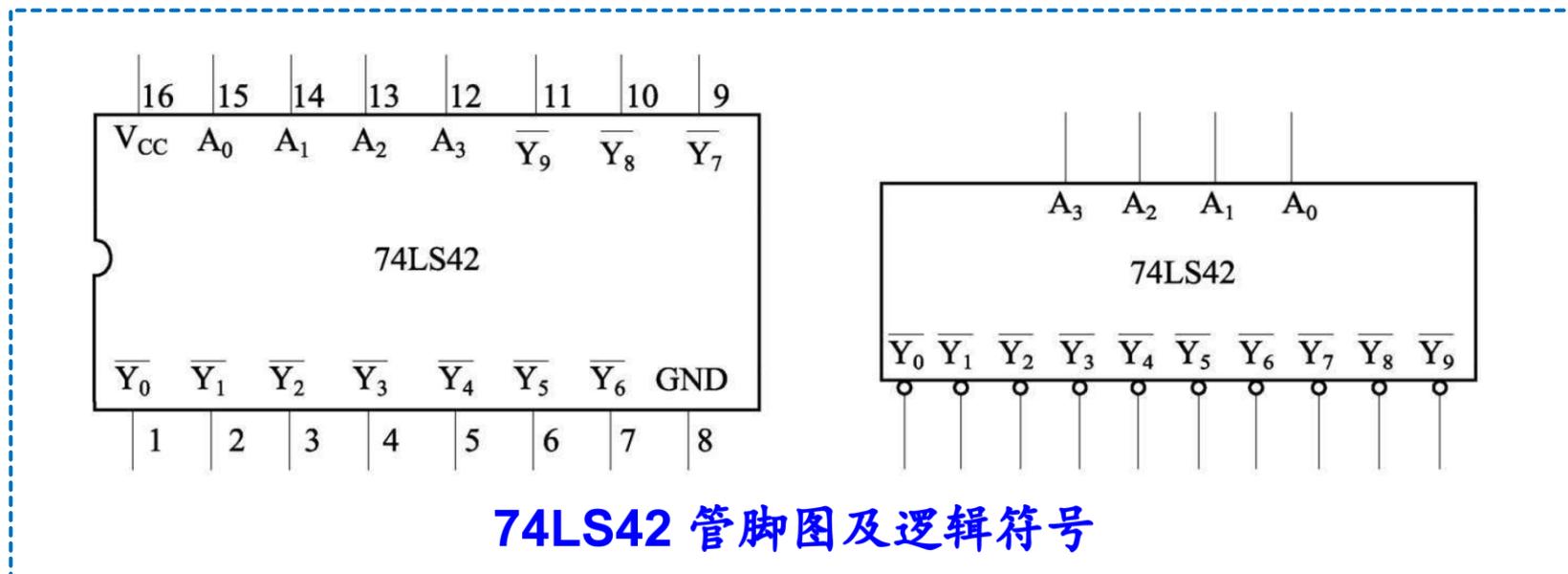
$$Y_9 = A \cdot \bar{B} \cdot \bar{C} \cdot D$$

- ① 部分译码电路简单，但可能出错
- ② 完全译码是最小项输出，伪输入时，不会有伪输出
- ③ 集成电路一般采用完全译码方式



3.4 常用的集成组合逻辑电路

三. 译码器：集成二-十进制译码器74LS42



- ✎ 4 个输入： $A_3 \sim A_0$
- ✎ 10 个输出： $\bar{Y}_0 \sim \bar{Y}_9$ ，低电平有效
- ✎ 无选通输入，全译码电路



3.4 常用的集成组合逻辑电路

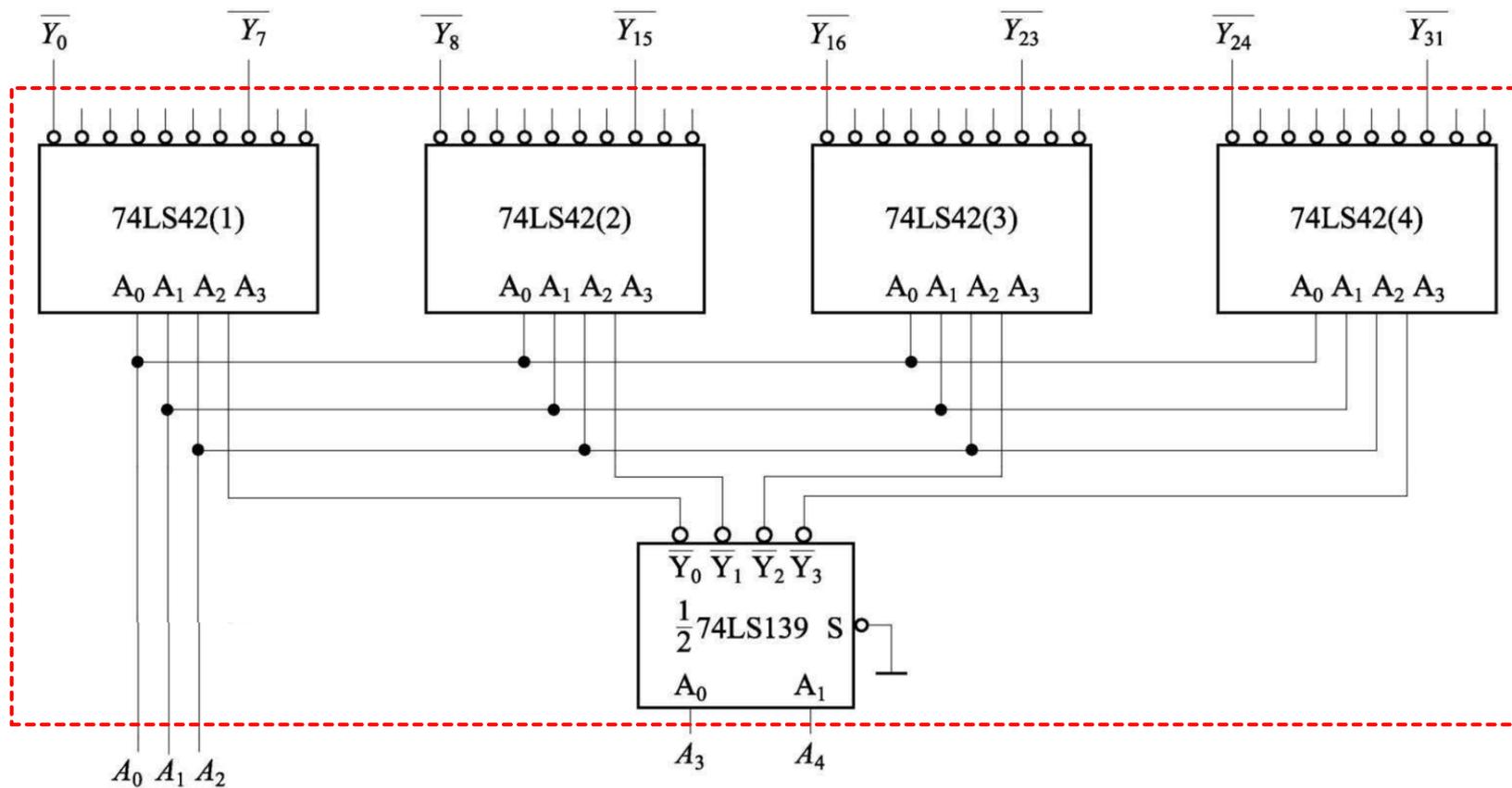
三. 译码器：集成二-十进制译码器74LS42

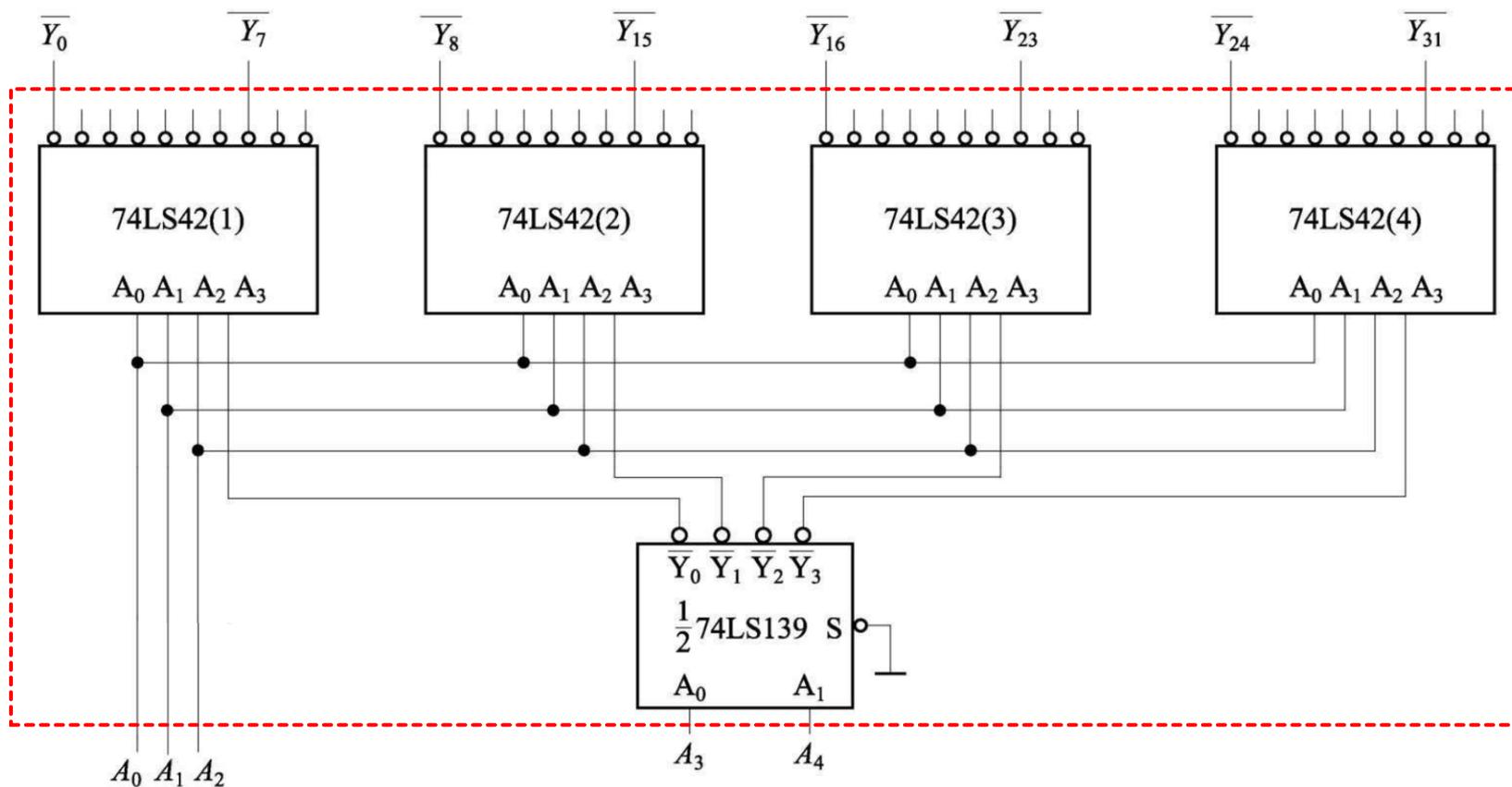
输入 BCD 码				输出									
A_3	A_2	A_1	A_0	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7	\bar{Y}_8	\bar{Y}_9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1010 ~ 1111				1	1	1	1	1	1	1	1	1	1

3.4 常用的集成组合逻辑电路

三. 译码器：74LS42的应用

◇ 用1片74LS139和4片74LS42构成一片5线/32线译码器





- ① 当 $A_4A_3 = 00$ 时，片 1 工作；
当 $A_2A_1A_0 = 000 \sim 111$ 时， $\bar{Y}_7 \sim \bar{Y}_0$ 有低电平输出，其余输出均为高电平
- ② 当 $A_4A_3 = 01$ 时，片 2 工作；
当 $A_2A_1A_0 = 000 \sim 111$ 时， $\bar{Y}_8 \sim \bar{Y}_{15}$ 有低电平输出，其余输出均为高电平
- ③ 当 $A_4A_3 = 10$ 和 $A_4A_3 = 11$ 时，有类似结论

3.4 常用的集成组合逻辑电路

三. 译码器：显示译码器

✧ 半导体显示器（LED）

- 磷砷化镓PN结，外加正向电压发出不同波长的光（红/黄/绿等颜色）

✧ 液晶显示器（LCD）

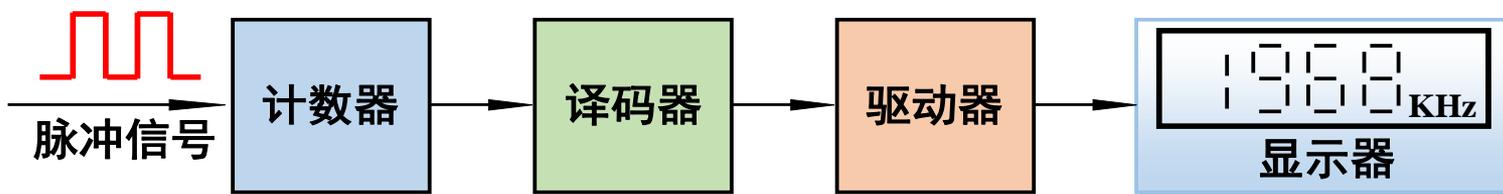
- 既有液体流动性又有晶体光学特性之有机化合物，是通过电场作用和入射光照射改变液晶排列形状、透明度而制成的显示器件



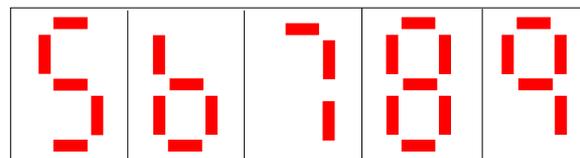
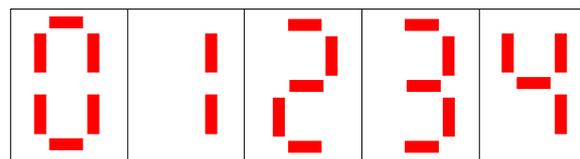
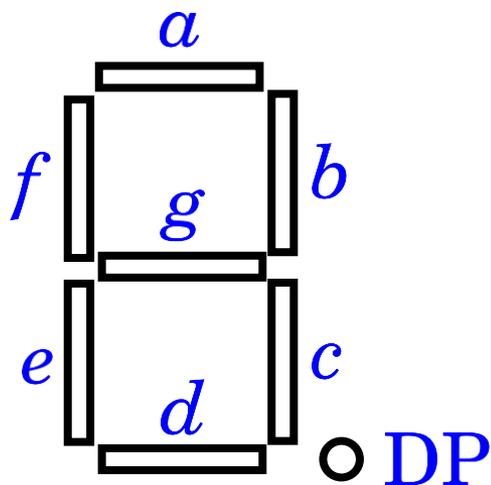
3.4 常用的集成组合逻辑电路

三. 译码器：显示译码器

◇ 数码显示器工作原理



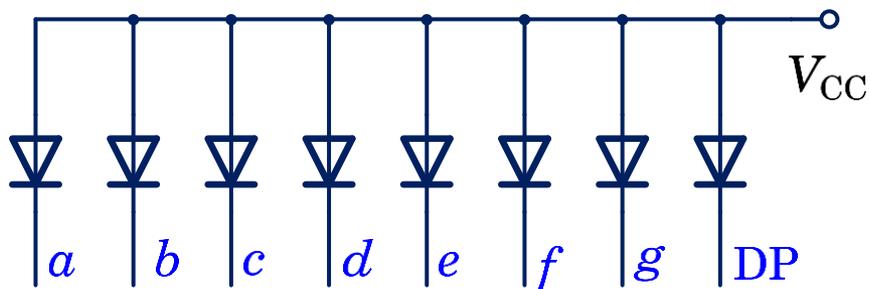
- 由七个发光二极管组成的数码显示器，可以显示十进制数



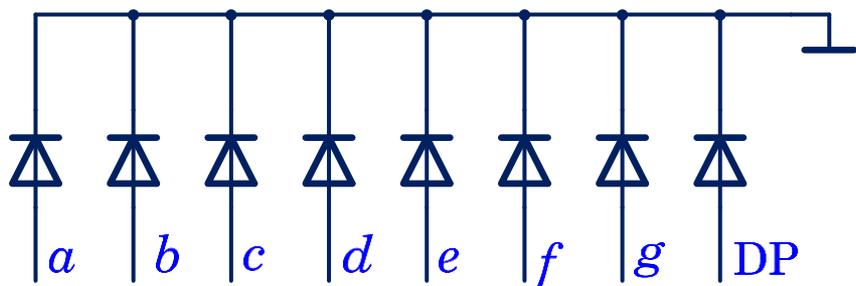
3.4 常用的集成组合逻辑电路

三. 译码器：显示译码器

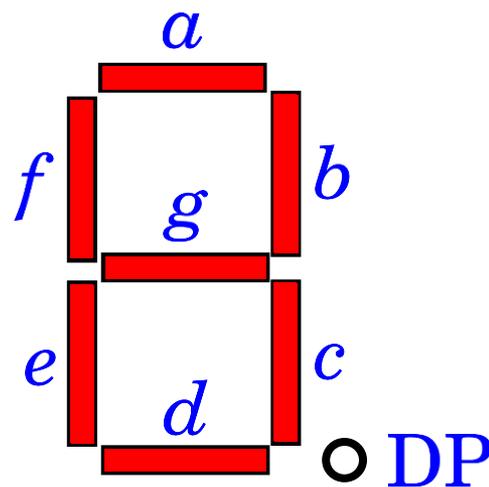
◇ 数码显示器工作原理（等效电路）



共阳极，需 0 驱动



共阴极，需 1 驱动

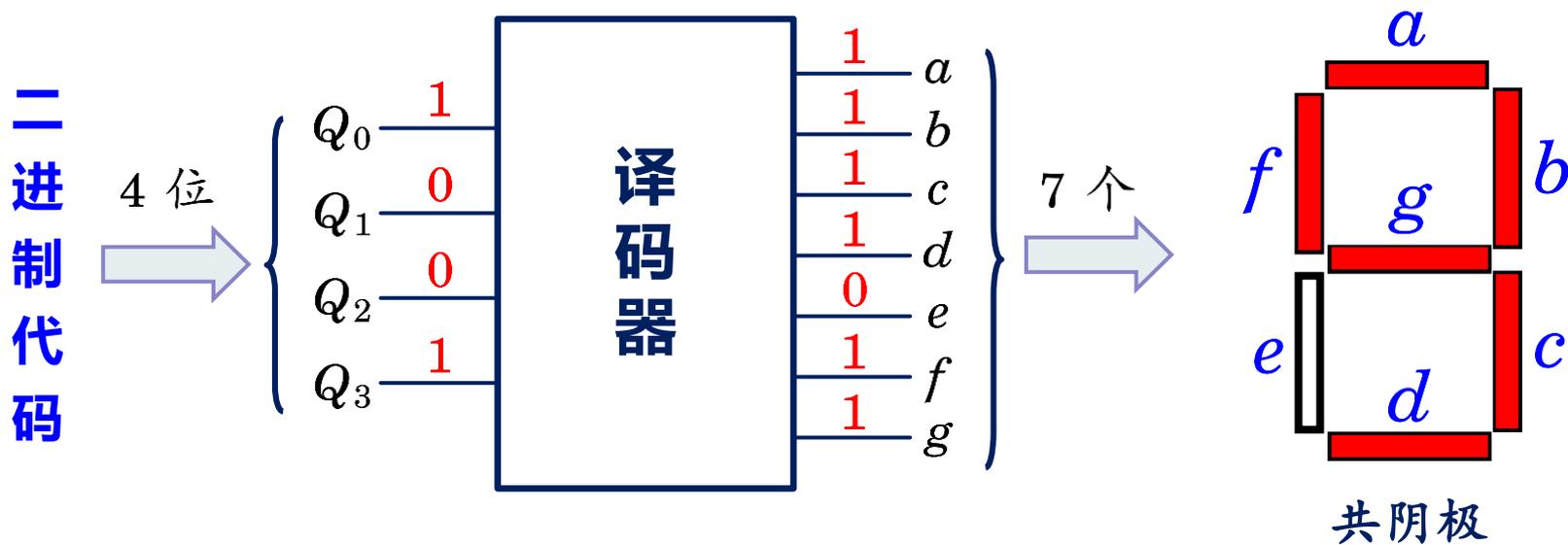


<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	DP
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	0
1	1	0	1	1	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮



3.4 常用的集成组合逻辑电路

三. 译码器：显示译码器



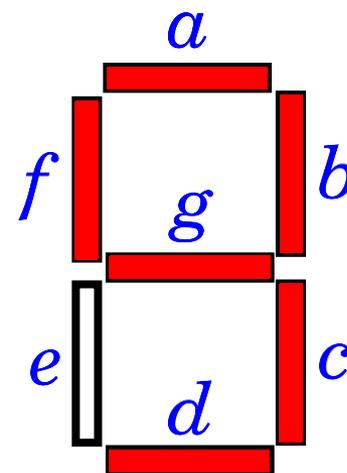


3.4 常用的集成组合逻辑电路

三. 译码器：显示译码器

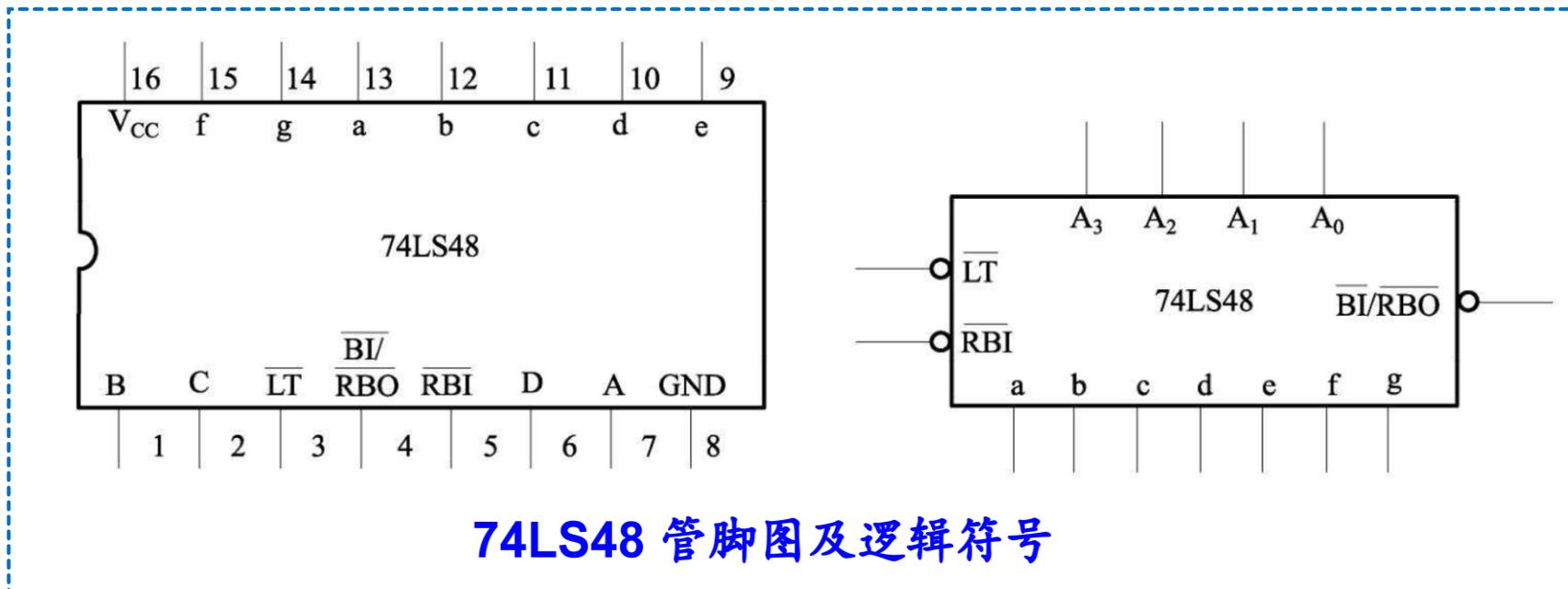
✧ BCD码七段显示译码器状态表

输入				输出						显示数码	
Q_3	Q_2	Q_1	Q_0	a	b	c	d	e	f		g
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9



3.4 常用的集成组合逻辑电路

三. 译码器：集成显示译码器/驱动器74LS48



4 个输入： $A_3 \sim A_0$

7 个输出： a 、 b 、 c 、 d 、 e 、 f 、 g ，高电平有效

3 个特殊功能端：灯测试输入端 \overline{LT} 消隐输入 $\overline{BI/RBO}$ 灭零输入 \overline{RBI}

74LS48 功能表

十进制 或功能	输 入						$\overline{BI}/\overline{RBO}$	输 出							字形
	\overline{LT}	\overline{RBI}	A_3	A_2	A_1	A_0		Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g	
0	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0
1	1	×	0	0	0	1	1	0	1	1	0	0	0	0	1
2	1	×	0	0	1	0	1	1	1	0	1	1	0	1	2
14	1	×	1	1	1	0	1	0	0	0	1	1	1	1	
15	1	×	1	1	1	1	1	0	0	0	0	0	0	0	不显示
消隐	×	×	×	×	×	×	0	0	0	0	0	0	0	0	不显示
脉冲消隐	1	0	0	0	0	0	0	0	0	0	0	0	0	0	不显示
灯测试	0	×	×	×	×	×	1	1	1	1	1	1	1	1	8

① 正常译码： $\overline{LT} = 1$ ， $\overline{BI}/\overline{RBO} = 1$ ，对输入十进制数的二进制码（0000~1001）译码

② 试灯：当 $\overline{LT} = 0$ 时，无论输入怎样， $a \sim g$ 输出全 1，数码管七段全亮

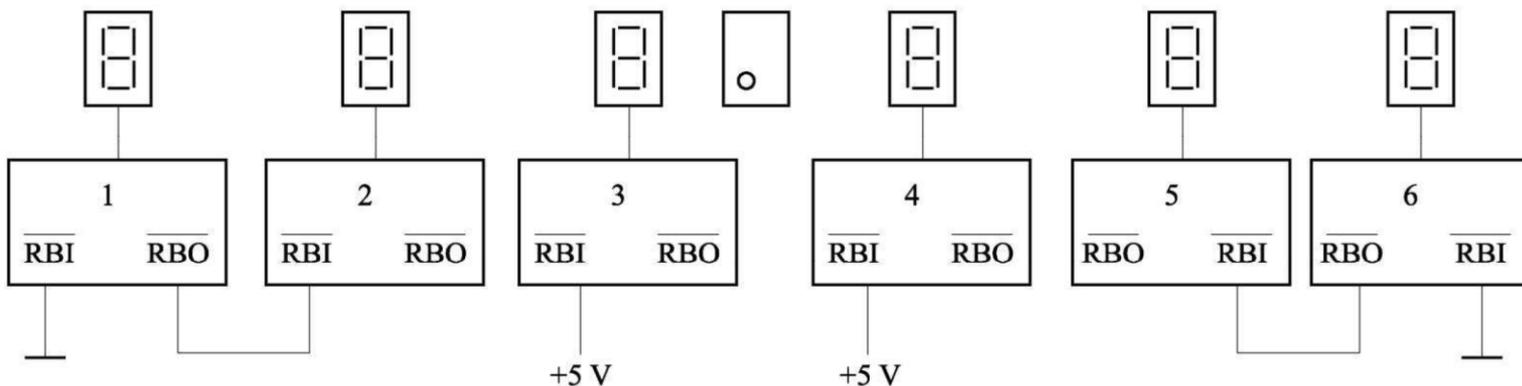
③ 灭零：当输入 $\overline{RBI} = 0$ ，输入为 00000 时，译码器的 $a \sim g$ 输出全 0，数码管全灭



3.4 常用的集成组合逻辑电路

例3-5

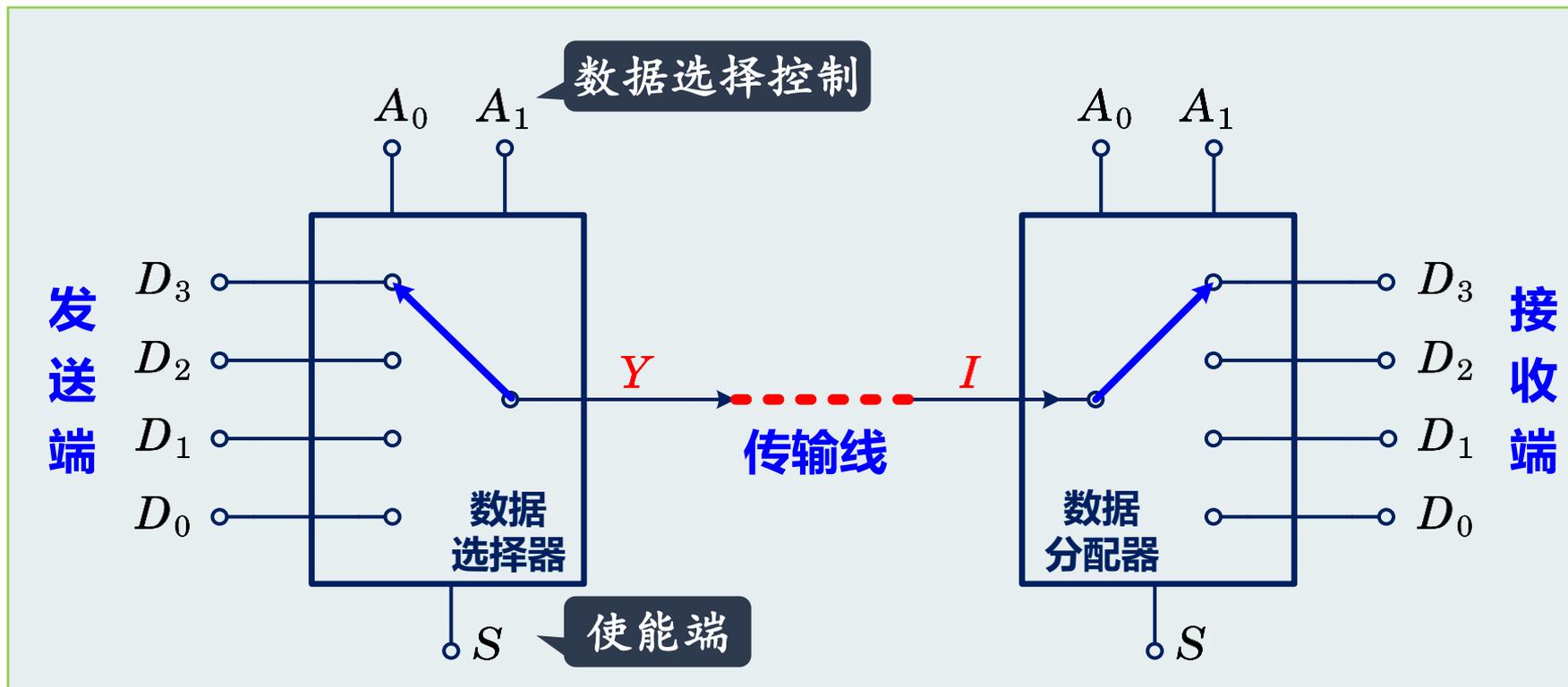
例 在多位十进制数码显示时，整数前和小数后的 0 是无意义的，称为“无效 0”。试分析图示电路的工作原理。



3.4 常用的集成组合逻辑电路

四. 数据选择器

✧ 根据地址选择码从多路输入数据中选择一路，送到输出端



✧ **应用**: 数字电路涉及到远距离通信时，并 → 串（分时） → 并

3.4 常用的集成组合逻辑电路

四. 数据选择器：4选1数据选择器原理

◇ 逻辑抽象

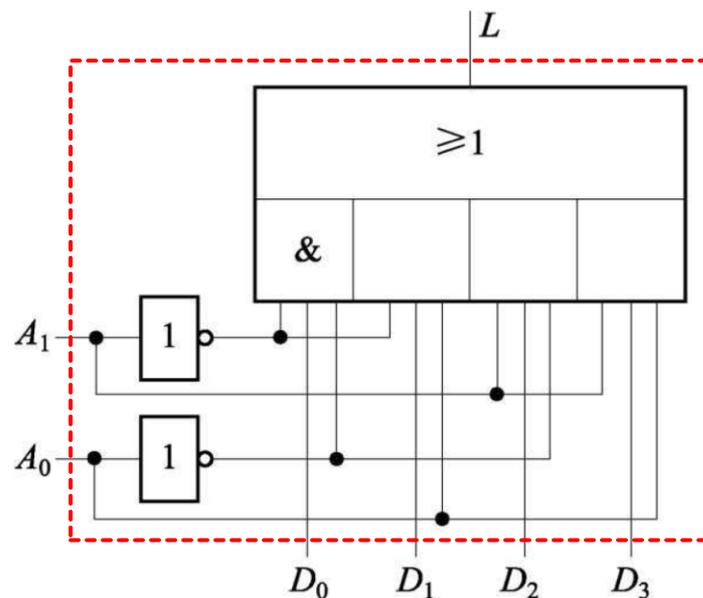
输入：2 个地址位 A_1 、 A_0 ；4 个数据位 D_3 、 D_2 、 D_1 、 D_0
 输出：1 个数据位 L

◇ 列真值表

输入地址		输入数据				输出
A_1	A_0	D_0	D_1	D_2	D_3	L
0	0	a_0	×	×	×	a_0
0	1	×	a_1	×	×	a_1
1	0	×	×	a_2	×	a_2
1	1	×	×	×	a_3	a_3

◇ 列逻辑表达式

$$\begin{aligned}
 L &= \bar{A}_1\bar{A}_0 \cdot D_0 + \bar{A}_1A_0 \cdot D_1 + A_1\bar{A}_0 \cdot D_2 + A_1A_0 \cdot D_3 \\
 &= m_0 \cdot D_0 + m_1 \cdot D_1 + m_2 \cdot D_2 + m_3 \cdot D_3
 \end{aligned}$$



逻辑电路图



3.4 常用的集成组合逻辑电路

四. 数据选择器：集成数据选择器

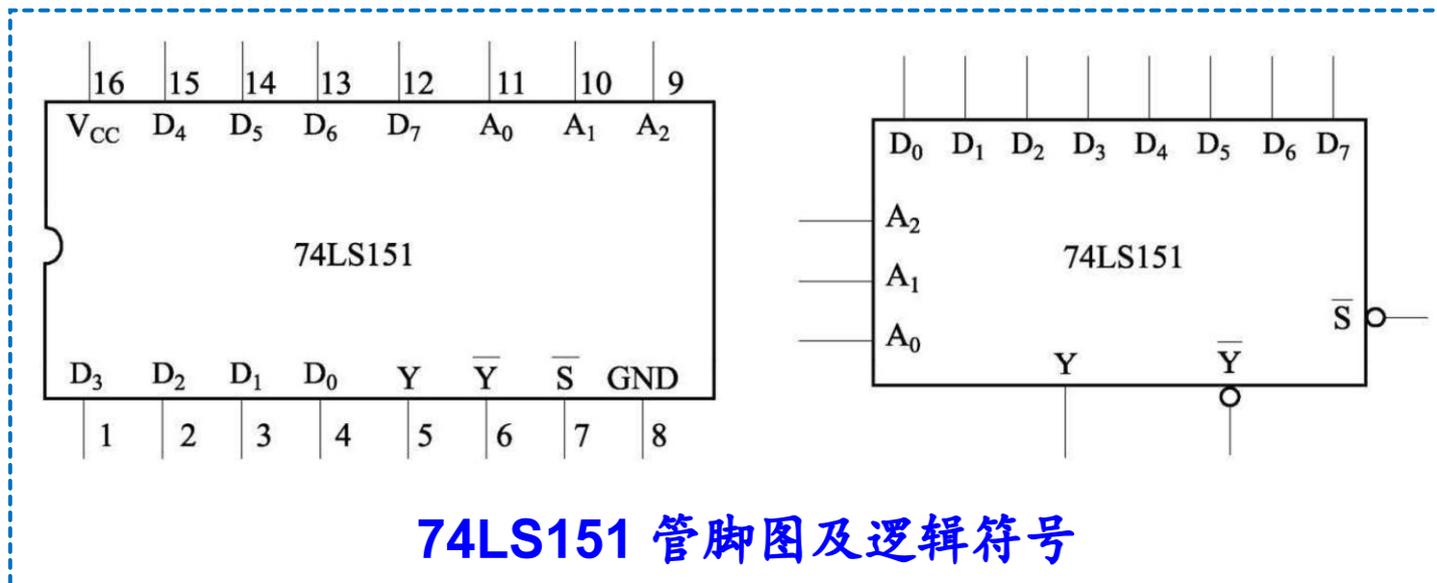
◇ 常见的集成数据选择器

- 4选1数据选择器：74LS153、74HC153、CC14539
- 8选1数据选择器：74LS151、74HC151、CC4512
- 16选1数据选择器：74LS150



3.4 常用的集成组合逻辑电路

四. 数据选择器：集成数据选择器（74LS151）



- 👉 8 个数据输入： $D_0 \sim D_7$
- 👉 3 个地址输入： $A_2 \sim A_0$
- 👉 2 个互补输出： Y 、 \bar{Y}
- 👉 1 使能端： \bar{S}



3.4 常用的集成组合逻辑电路

四. 数据选择器：集成数据选择器（74LS151）

✧ 逻辑功能

输入地址			输入使能	输出数据	
A_2	A_1	A_0	\overline{S}	Y	\overline{Y}
×	×	×	1	0	1
0	0	0	0	D_0	\overline{D}_0
0	0	1	0	D_1	\overline{D}_1
0	1	0	0	D_2	\overline{D}_2
0	1	1	0	D_3	\overline{D}_3
1	0	0	0	D_4	\overline{D}_4
1	0	1	0	D_5	\overline{D}_5
1	1	0	0	D_6	\overline{D}_6
1	1	1	0	D_7	\overline{D}_7

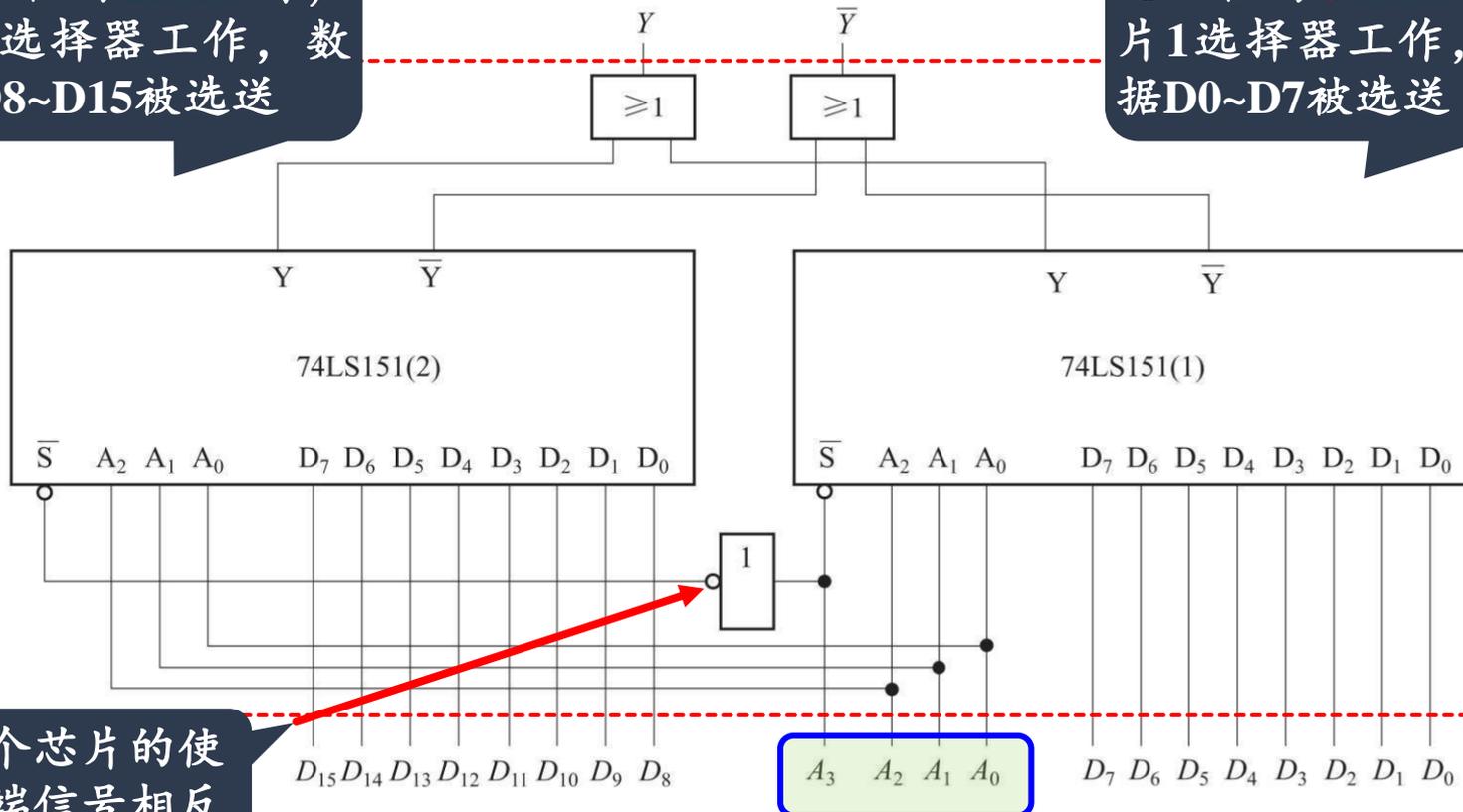
3.4 常用的集成组合逻辑电路

四. 数据选择器：数据选择器的应用（通道扩展）

① 有使能端的数据选择器的扩展（字扩展为16选1）

地址位为**1XXX**时，片2选择器工作，数据D8~D15被选送

地址位为**0XXX**时，片1选择器工作，数据D0~D7被选送

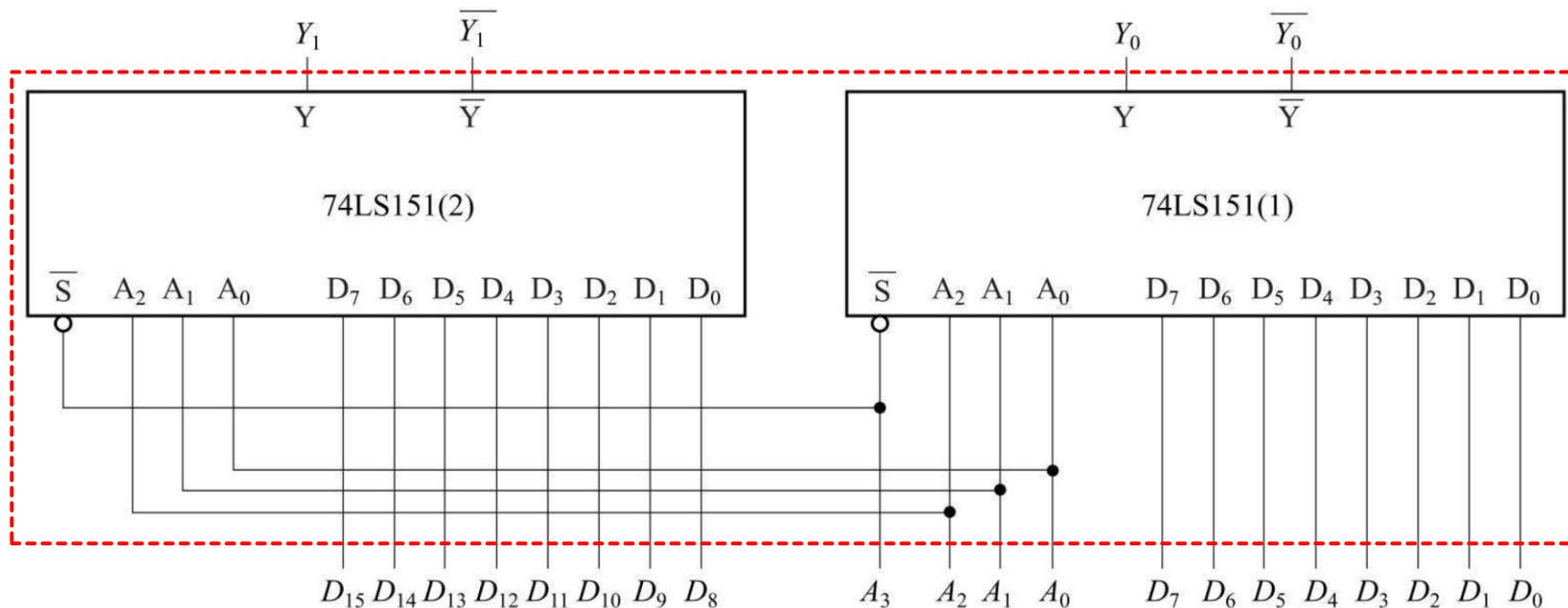


两个芯片的使能端信号相反

3.4 常用的集成组合逻辑电路

四. 数据选择器：数据选择器的应用（通道扩展）

② 有使能端的数据选择器的扩展（位扩展为16选2）



- ① 当 $A_3 = 0$ 时，使能端 \bar{S} 有效，两片 74LS151 均能工作
- ② 当地址位信号确定时，每片的对应数据被送到输出端

3.4 常用的集成组合逻辑电路

四. 数据选择器：数据选择器的应用（通道扩展）

③ 不用（或无）使能端的数据选择器的扩展

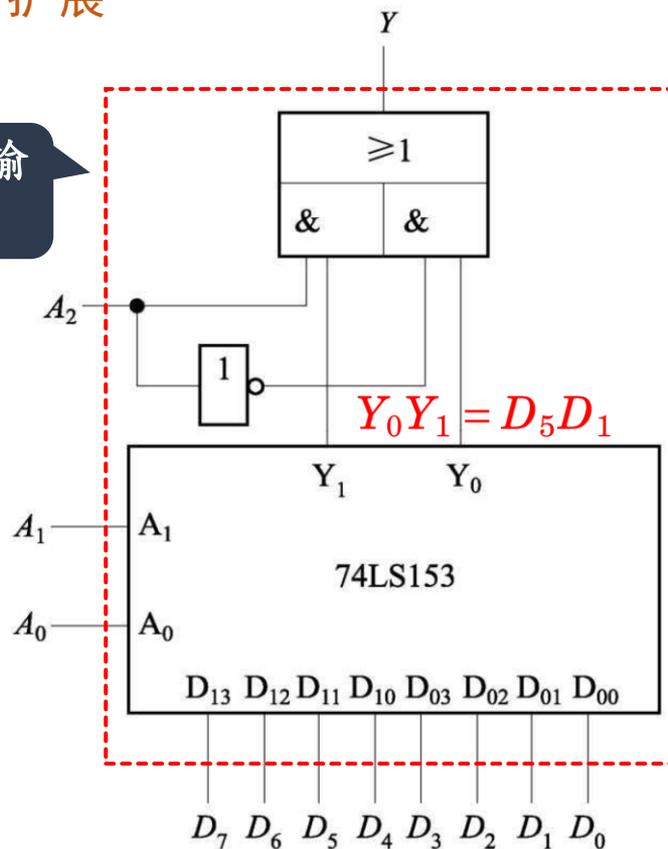
- 一般采用先输出，再选择的方式进行扩展

原理：先由74LS153将低2位地址对应的数据输出到复合门，再由最高位地址进行选择输出

$$A_2A_1A_0 = 101$$

$$A_1A_0 = 01$$

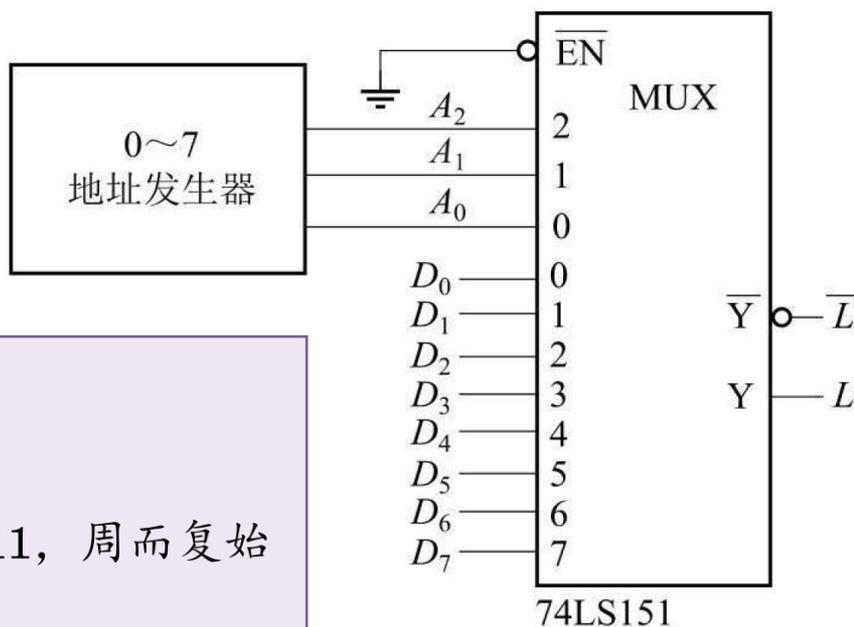
此时由 A_2 的取值决定最终输出，由于 $A_2 = 1$ ， Y_0 的数据被选中，最终输出 D_5 数据



3.4 常用的集成组合逻辑电路

四. 数据选择器：数据选择器的应用

④ 实现数据的并串转换



- ① 并行数据送到 $D_0 \sim D_7$
- ② 地址发生器输出从 000 ~ 111, 周而复始
- ③ L 输出从 $D_0 \sim D_7$ 变化, 得到串行信号



3.4 常用的集成组合逻辑电路

四. 数据选择器：数据选择器的应用

⑤ 实现组合逻辑函数

✧ 依据：

① n 变量函数：部分最小项之和的形式为 $L = \sum_{i=0}^{2^n-1} m_i$

② 数据选择器： $L = \sum_{i=0}^{2^n-1} m_i D_i$

✧ 思路：

① 使函数卡诺图（或最小项表达式）与数据选择器卡诺图完全一致

② 对应函数所含的 m_i 项，使数据选择器的 $D_i = 1$

③ 对应函数所不含的 m_j 项，使数据选择器的 $D_j = 0$



3.4 常用的集成组合逻辑电路

四. 数据选择器：数据选择器的应用

⑤ 实现组合逻辑函数

✧ 方法步骤：

- 适用条件：数据选择器地址位个数与逻辑函数变量个数相等
 - ① 将给定函数化为最小项表达式
 - ② 根据函数的变量数 n ，确定 MUX 的规模
 - ③ 将函数式中已存在的最小项 m_i 相对应的数据输入端 D_i 赋值为 1
 - ④ 将函数式不存在的最小项相对应的数据输入端赋值为 0
 - ⑤ 画出电路图

3.4 常用的集成组合逻辑电路

例3-6

例 用 74LS151 实现全加和函数 $L = A \oplus B \oplus C = \sum m(1, 2, 4, 7)$ 。

方法1

解 将函数写为标准最小项表达式

$$L = \sum m(1, 2, 4, 7) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

对照 74LS151 的逻辑功能输出表达式

$$Y = \bar{A}_2\bar{A}_1\bar{A}_0 \cdot D_0 + \bar{A}_2\bar{A}_1A_0 \cdot D_1 + \bar{A}_2A_1\bar{A}_0 \cdot D_2 + \bar{A}_2A_1A_0 \cdot D_3 + A_2\bar{A}_1\bar{A}_0 \cdot D_4 + A_2\bar{A}_1A_0 \cdot D_5 + A_2A_1\bar{A}_0 \cdot D_6 + A_2A_1A_0 \cdot D_7$$

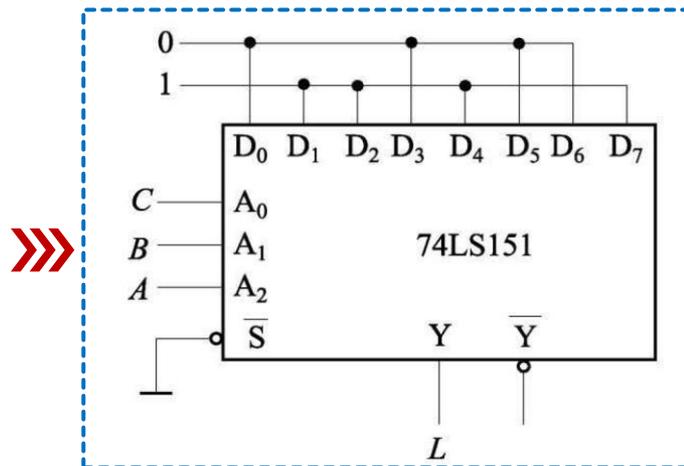
令

$$D_1 = D_2 = D_4 = D_7 = 1$$

$$D_0 = D_3 = D_5 = D_6 = 0$$

$$A = \bar{A}_2, B = \bar{A}_1, C = \bar{A}_0$$

$$L = Y$$



3.4 常用的集成组合逻辑电路

例3-6

例 用 74LS151 实现全加和函数 $L = A \oplus B \oplus C = \sum m(1, 2, 4, 7)$ 。

方法2

解 卡诺图对比

	BC			
	00	01	11	10
A				
0		1		1
1	1		1	

	A_1A_0			
	00	01	11	10
A_2				
0	D_0	D_1	D_3	D_2
1	D_4	D_5	D_7	D_6

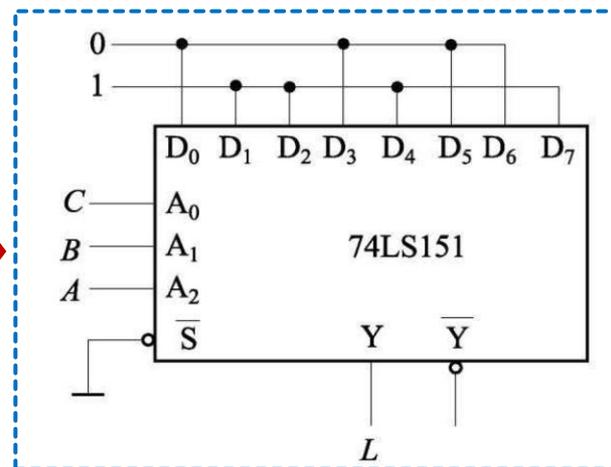
令

$$D_1 = D_2 = D_4 = D_7 = 1$$

$$D_0 = D_3 = D_5 = D_6 = 0$$

$$A = \bar{A}_2, B = \bar{A}_1, C = \bar{A}_0$$

$$L = Y$$





3.4 常用的集成组合逻辑电路

四. 数据选择器：数据选择器的应用

⑤ 实现组合逻辑函数

- 适用条件：数据选择器地址位个数少于逻辑函数输入变量个数

✧ 方法1：扩展法

- 利用数据选择器的通道扩展方式，将数据选择器地址位扩展为与逻辑变量个数相同

✧ 方法2：降维法

- 将逻辑函数最小项表达式（卡诺图）维数降到与数据选择器的一致

3.4 常用的集成组合逻辑电路

例3-7

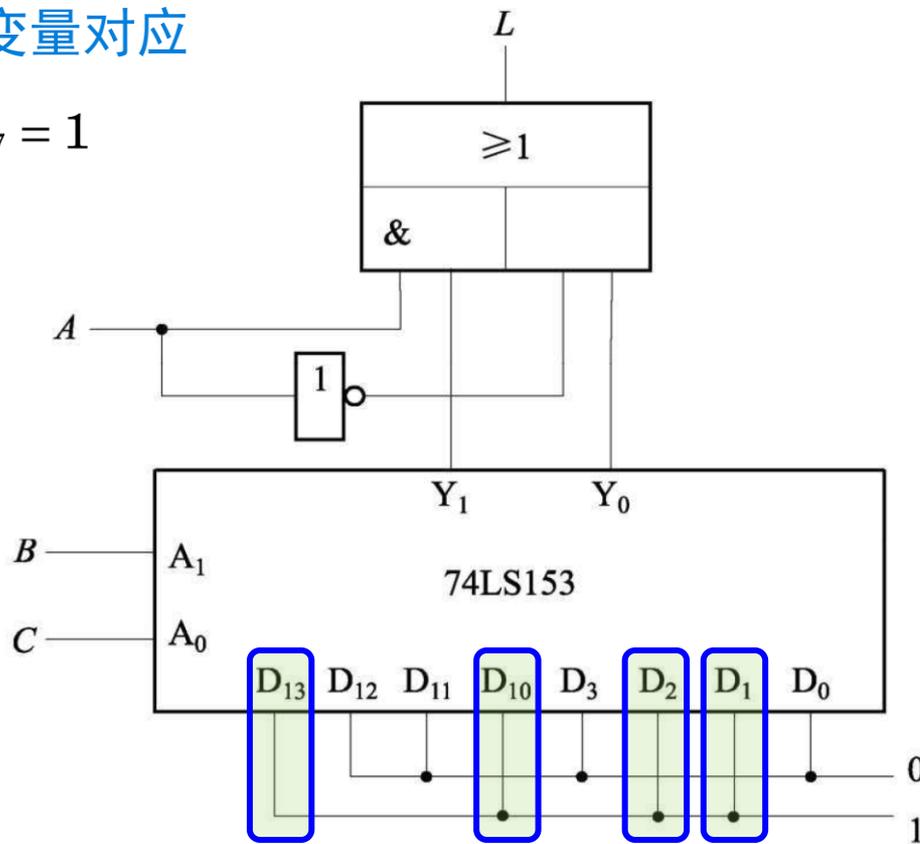
例 用 74LS153 实现全加和函数 $L = A \oplus B \oplus C = \sum m(1, 2, 4, 7)$ 。

方法1

解 首先将 74LS153 附加门电路 构成 8 选 1 数据选择器

然后令地址位与逻辑变量对应

$$D_1 = D_2 = D_4 = D_7 = 1$$



3.4 常用的集成组合逻辑电路

例3-7

例 用 74LS153 实现全加和函数 $L = A \oplus B \oplus C = \sum m(1, 2, 4, 7)$ 。

方法2

解 将函数写为标准最小项表达式

$$L = \sum m(1, 2, 4, 7) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

对照 74LS153 的逻辑功能输出表达式

$$Y = \bar{A}_1\bar{A}_0 \cdot D_0 + \bar{A}_1A_0 \cdot D_1 + A_1\bar{A}_0 \cdot D_2 + A_1A_0 \cdot D_3$$

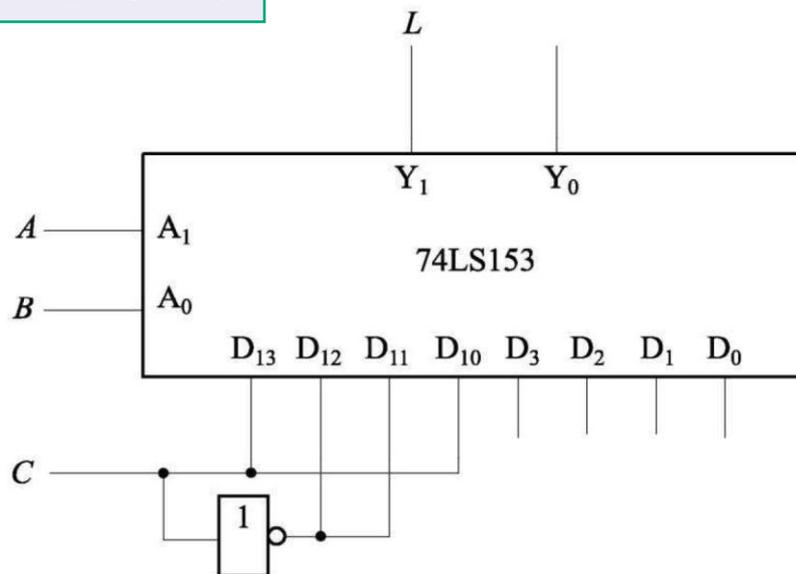
令 $A_1 = A, A_0 = B$, 则

$$Y = \bar{A}\bar{B}D_0 + \bar{A}BD_1 + A\bar{B}D_2 + ABD_3$$

比较 L 和 Y

$$D_0 = D_3 = C$$

$$D_1 = D_2 = \bar{C}$$



3.4 常用的集成组合逻辑电路

例3-7

例 用 74LS153 实现全加和函数 $L = A \oplus B \oplus C = \sum m(1, 2, 4, 7)$ 。

方法2

解 将函数写为标准最小项表达式

$$L = \sum m(1, 2, 4, 7) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

对照 74LS153 的逻辑功能输出表达式

$$Y = \bar{A}_1\bar{A}_0 \cdot D_0 + \bar{A}_1A_0 \cdot D_1 + A_1\bar{A}_0 \cdot D_2 + A_1A_0 \cdot D_3$$

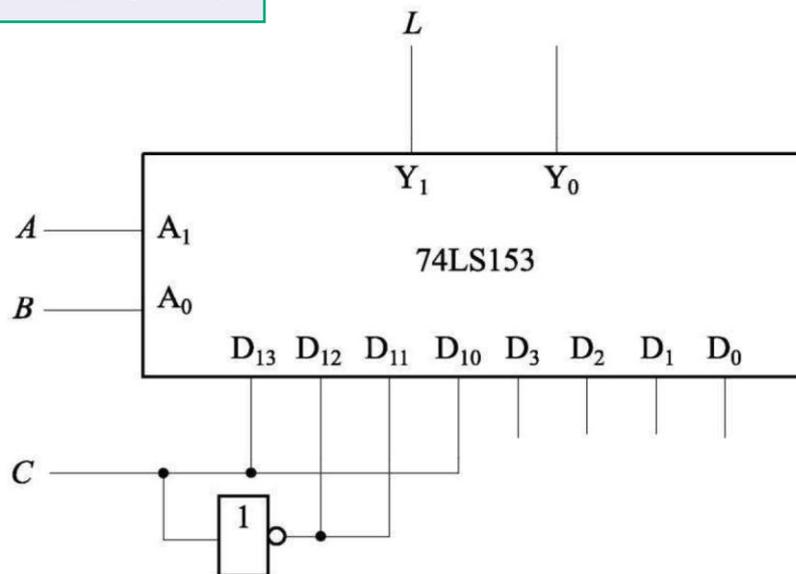
令 $A_1 = A, A_0 = B$, 则

$$Y = \bar{A}\bar{B}D_0 + \bar{A}BD_1 + A\bar{B}D_2 + ABD_3$$

比较 L 和 Y

$$D_0 = D_3 = C$$

$$D_1 = D_2 = \bar{C}$$





3.4 常用的集成组合逻辑电路

四. 数值比较器

◇ 用来比较两个数字大小

- 可以对两组二进制数或者二—十进制数进行比较
- 比较结果有“>” “<” 和“=”

◇ 常见类型

- 74LS85
- CD4063

3.4 常用的集成组合逻辑电路

四. 数值比较器：工作原理

✧ 以设计两个一位二进制数比较器为例

✧ 逻辑抽象

✧ 输入：2 个待比较数据 A 、 B
 ✧ 输出：1 个比较结果 $L_{A>B}$ 、 $L_{A<B}$ 或 $L_{A=B}$

✧ 列真值表

输入		输出		
A	B	$L_{(A>B)}$	$L_{(A<B)}$	$L_{(A=B)}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

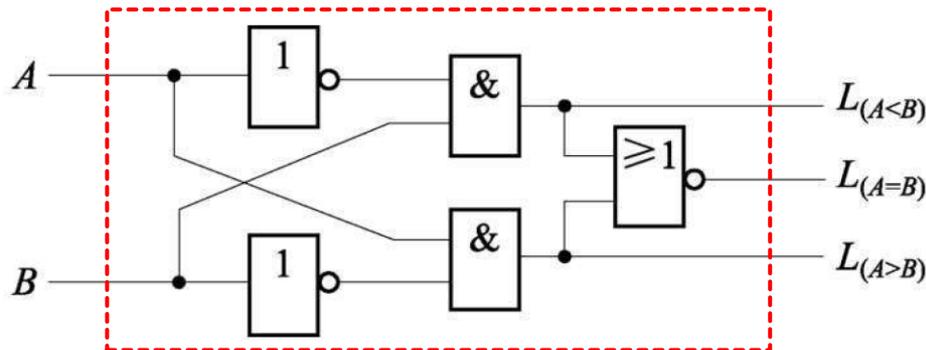
✧ 列逻辑表达式

$$L_{(A>B)} = \bar{A}\bar{B}$$

$$L_{(A<B)} = \bar{A}B$$

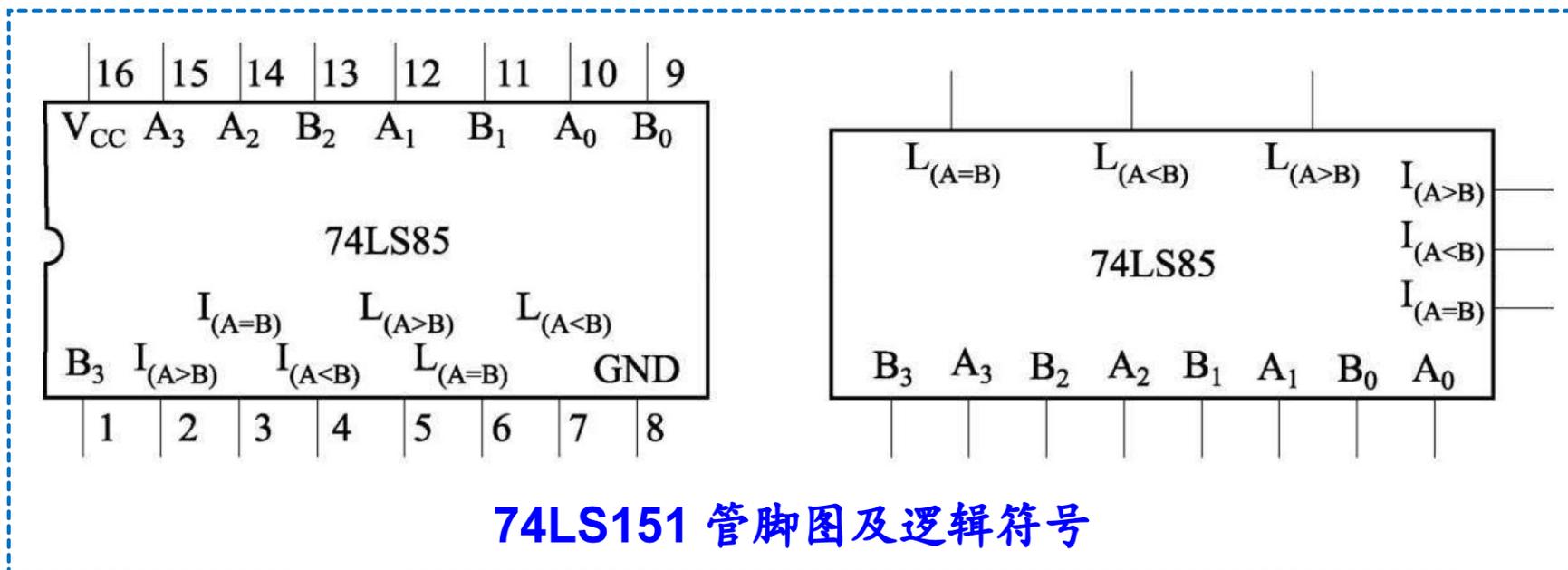
$$L_{(A=B)} = \bar{A}\bar{B} + AB$$

逻辑电路图 \ggg



3.4 常用的集成组合逻辑电路

四. 数值比较器：集成数值比较器（74LS85）



2 个 4 位二进制数输入： $B_3 \sim B_0, A_3 \sim A_0$ （交替排列）

3 个级联信号输入： $I_{(A>B)}, I_{(A<B)}, I_{(A=B)}$ （级联实现多位数据比较）

3 个比较结果输出： $Y_{(A>B)}, Y_{(A<B)}, Y_{(A=B)}$



3.4 常用的集成组合逻辑电路

四. 数值比较器：集成数值比较器（74LS85）

✧ 逻辑功能：高位优先，本位数值相等，则考虑级联信号

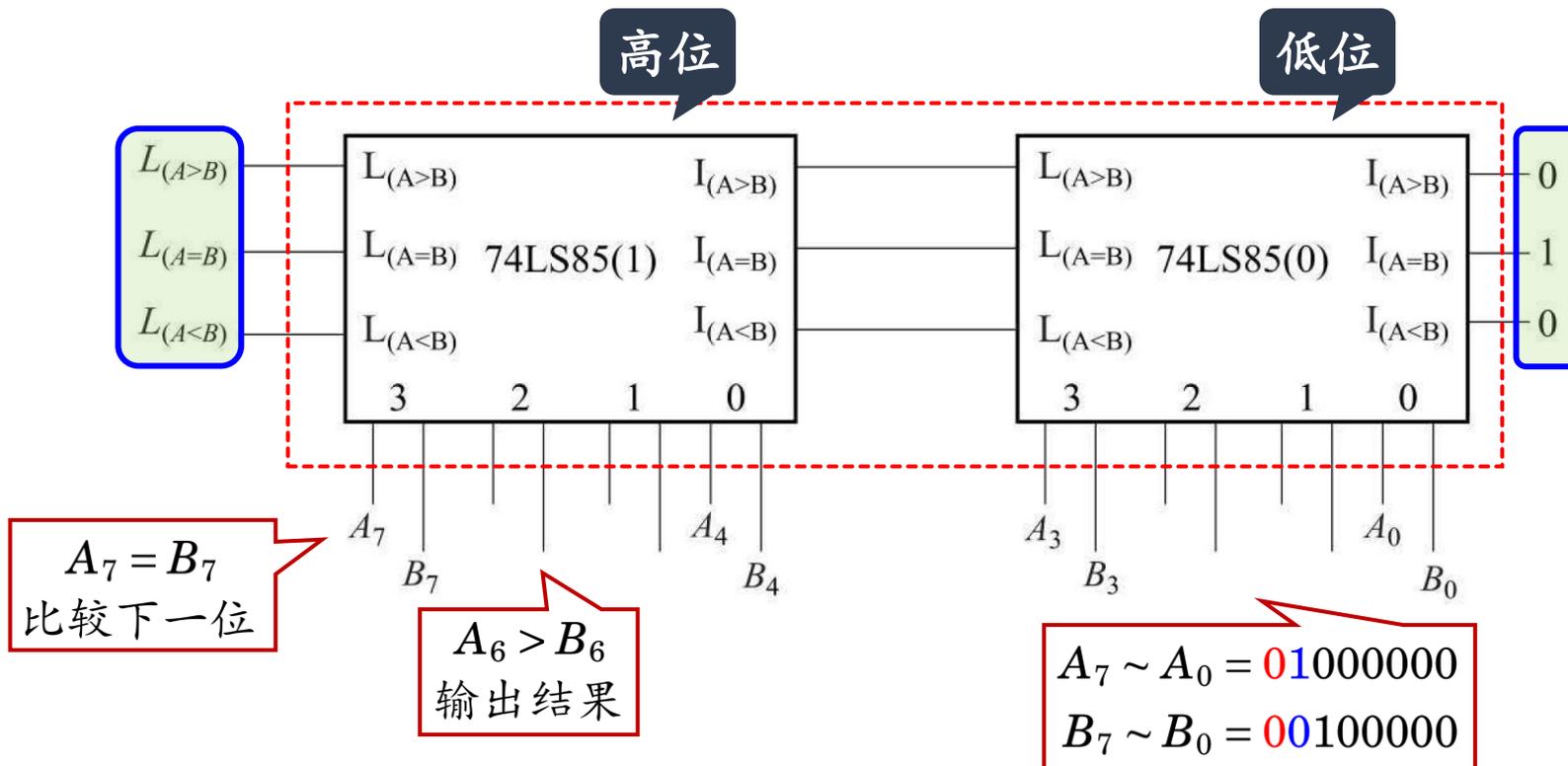
输入				输出					
$A_3 - B_3$	$A_2 - B_2$	$A_1 - B_1$	$A_0 - B_0$	$I_{(A>B)}$	$I_{(A<B)}$	$I_{(A=B)}$	$L_{(A>B)}$	$L_{(A<B)}$	$L_{(A=B)}$
>	×	×	×	×	×	×	1	0	0
<	×	×	×	×	×	×	0	1	0
=	>	×	×	×	×	×	1	0	0
=	<	×	×	×	×	×	0	1	0
=	=	>	×	×	×	×	1	0	0
=	=	<	×	×	×	×	0	1	0
=	=	=	>	×	×	×	1	0	0
=	=	=	<	×	×	×	0	1	0
=	=	=	=	1	0	0	1	0	0
=	=	=	=	0	1	0	0	1	0
=	=	=	=	0	0	1	0	0	1

3.4 常用的集成组合逻辑电路

四. 数值比较器：数值比较器的扩展

① 串联扩展

✧ 先比较高片（1片）的结果，若两组数据相等，再比较低片数据

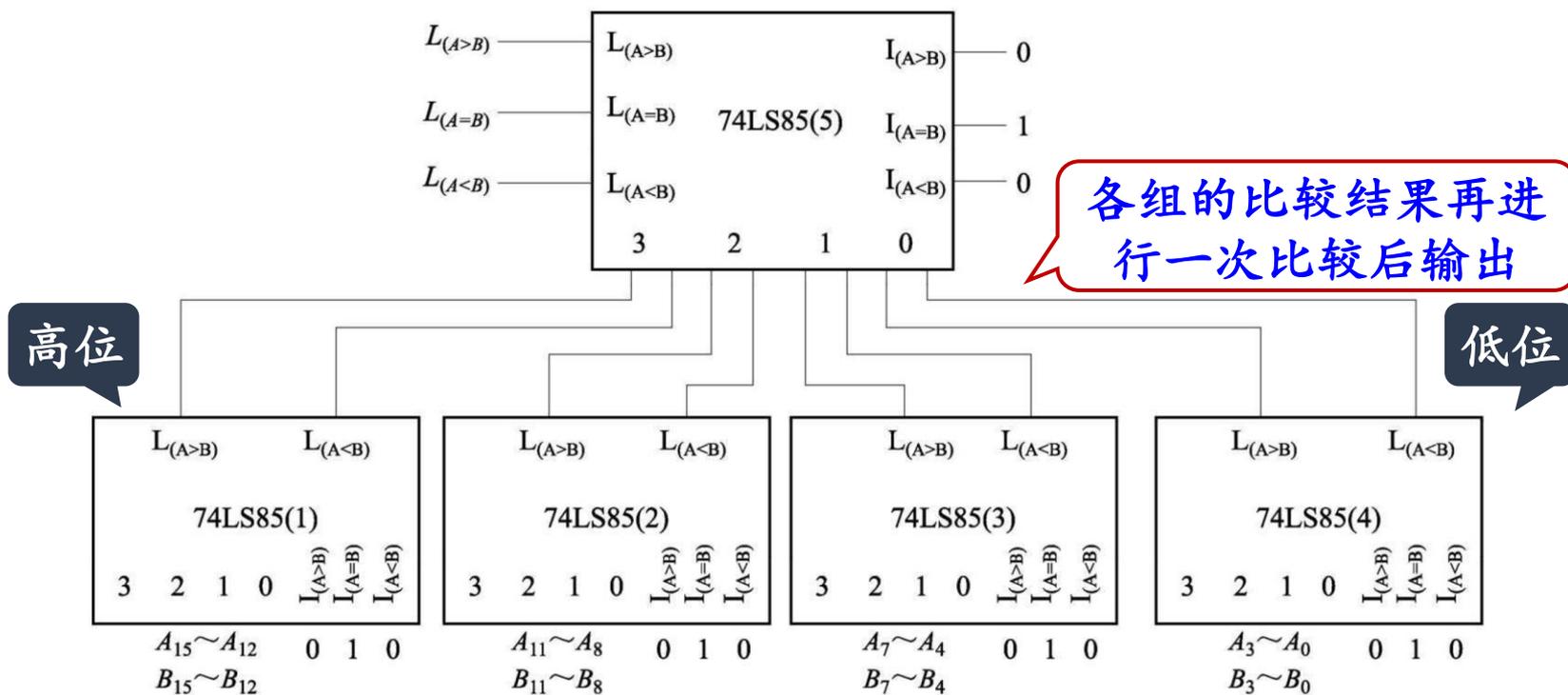


3.4 常用的集成组合逻辑电路

四. 数值比较器：数值比较器的扩展

② 并联扩展

- 主要用于待比较数据较多，且有一定速度要求的情形



将16位按高低位次序分成4组，各组并行比较



本章小结

- 组合逻辑电路的一般分析方法
- 组合逻辑电路的设计方法
- 组合逻辑电路中的竞争-冒险
- 常用的集成组合逻辑电路